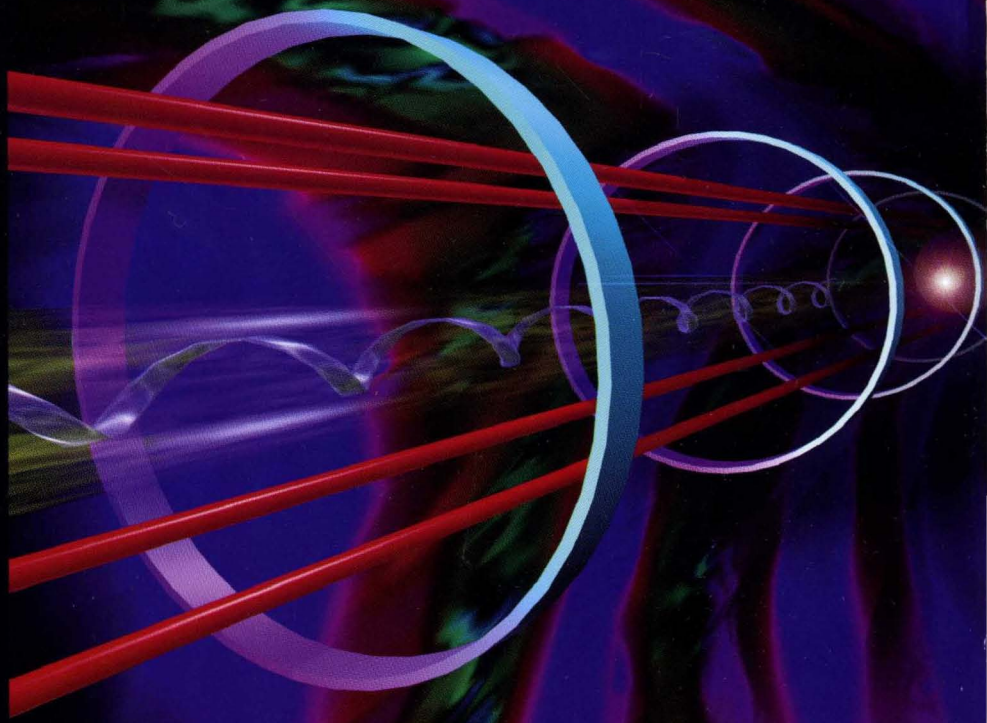
i960® Processors and Related Products

High-Performance 32-Bit RISC Processors for Embedded Applications



intel®

LITERATURE

For additional information on Intel products in the U.S. or Canada, call Intel's Literature Center at (800) 548-4725 or write to:

Intel Literature
P.O. Box 7641
Mt. Prospect, Il 60056-7641

To order literature outside of the U.S. and Canada contact your local international sales office.

CURRENT DATABOOKS

Product line databooks contain datasheets, application notes, article reprints, and other design information. Databooks can be ordered in the U.S. and Canada by calling TAB/McGraw-Hill at 1-800-822-8158; outside of the U.S. and Canada contact your local international sales office.

Title	Intel Order Number	ISBN
Automotive Products	231792	N/A
Embedded Applications (2 vol. set)	270648	1-55512-242-6
Embedded Microcontrollers	270646	1-55512-230-2
Embedded Microprocessors	272396	1-55512-231-0
Flash Memory (2 vol. set)	210830	1-55512-232-9
Intel486™ Microprocessors and Related Products	241731	1-55512-235-3
i960® Processors and Related Products	272084	1-55512-234-5
Military and Special Products	210461	N/A
Networking	297360	1-55512-236-1
OEM Boards, Systems and Software	280407	1-55512-237-X
Packaging	240800	1-55512-238-8
Pentium™ Processors and Related Products	241732	1-55512-239-6
Peripheral Components	296467	1-55512-240-X

A complete set of this information is available on CD-ROM through Intel's Data on Demand program, order number 240897. For information about Intel's Data on Demand ask for item number 240952.

24-HOUR AUTOMATED TECHNICAL SUPPORT*

Intel's Application Bulletin Board System (BBS) and FaxBack System are at your service, 24-hours a day, at no charge, and the information is updated frequently.

FaxBack SYSTEM

Technical and product information are available 24-hours a day! Order documents containing:

- Product Announcements
- Product Literature
- Intel Device Characteristics
- Design/Application Recommendations
- Stepping/Change Notifications
- Quality and Reliability Information

Information on the following subjects is also available:

- Microcontroller and Flash
- OEM Branded Systems
- Multibus/BBS Listing
- Multimedia
- Development Tools
- Quality and Reliability/Change Notification
- Microprocessor/PCI/Peripheral
- Intel Architecture Lab

To use FaxBack for Intel components and systems, dial **(800) 628-2283** or (916) 356-3105 (U.S. and Canada) *or +44{0} 1793-496646 (Europe)* and follow the automated voice-prompt menu. Document orders will be faxed to the fax number you specify. Catalogs are updated twice a month, so call for the latest information!

BULLETIN BOARD SYSTEM

Intel's Application Bulletin Board System (BBS) enables file retrieval 24-hours a day. The following can be located on the BBS:

- Software Drivers
- Tool Information
- Software/Application Utilities
- Product/Technical Documentation
- Firmware Upgrades
- Quality and Reliability Data

To use the Intel Application BBS (components and systems), dial **(916) 356-3600** for download access (U.S. and Canada) *or +44{0} 1793-496340 (Europe).* The BBS will support 1200–19200 baud rate modem. *Typical modem configuration: 9600 baud rate, No Parity, 8 Data Bits, 1 Stop Bit.* A directory listing of BBS files is also available through FaxBack or our 800 BBS (800-897-2536).

Retail Products

Information on Intel's retail products (Coprocessors and wireless, video, personal conferencing and network products) is available through the following services:

Internet : ftp.intel.com (143.185.65.2)
CompuServe : GO INTELFORUM (modem settings: E-7-1, up to 14.4 Kbps)

Country	BBS (N-8-1, up to 14.4 Kbps)	FaxBack
North America	(503) 264-7999	(800) 525-3019 or (503) 264-6835
Europe	+44 1 793-432955	+44 1 793-432509
Australia	+61 2 975-3066	+61 2 975-3922
Taiwan	+886 2 718-6422	+886 2 514-0815
Singapore	+65 256-4776	+65 256-5350
Hong Kong	+852 530-4116	+852 844-4448
Korea	+822 784-3430	+822 767-2594

*Support services provided courtesy of Intel Application Support

intel®

i960® Processors and Related Products

High-Performance 32-Bit RISC Processors for Embedded Applications

1995

Additional copies of this document or other Intel literature may be obtained from:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641

or call 1-800-879-4683

DATASHEET DESIGNATIONS

Intel uses various datasheet markings to designate each phase of the document as it relates to the product. The markings appear in the lower inside corner of each datasheet page. Following are the definitions of each marking:

Datasheet Marking	Description
Product Preview	Contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product becomes available.
Advanced Information	Contains information on products being sampled or in the initial production phase of development.*
Preliminary	Contains preliminary information on new products in production.*
No Marking	Contains information on products in full production.*

* Specifications within these datasheets are subject to change without notice. Verify with your local Intel sales office that you have the latest datasheet before finalizing a design.

intel®

i960® Microprocessor Family 1

Memories and Peripherals 2

Development Support Tools 3

Table of Contents

Alphanumeric Index

intel

1

i960® Microprocessor Family

80960SA
EMBEDDED 32-BIT MICROPROCESSOR WITH 16-BIT BURST DATA BUS

- **High-Performance Embedded Architecture**
  - **— 20 MIPS* Burst Execution at 20 MHz**
  - **— 7.5 MIPS Sustained Execution at 20 MHz**
- **512-Byte On-Chip Instruction Cache**
  - **— Direct Mapped**
  - **— Parallel Load/Decode for Uncached Instructions**
- **Multiple Register Sets**
  - **— Sixteen Global 32-Bit Registers**
  - **— Sixteen Local 32-Bit Registers**
  - **— Four Local Register Sets Stored On-Chip**
  - **— Register Scoreboarding**
- **Pin Compatible with 80960SB**
- **Built-in Interrupt Controller**
  - **— 4 Direct Interrupt Pins**
  - **— 31 Priority Levels, 256 Vectors**
- **Easy to Use, High Bandwidth 16-Bit Bus**
  - **— 32 Mbytes/s Burst**
  - **— Up to 16 Bytes Transferred per Burst**
- **32-Bit Address Space, 4 Gigabytes**
- **80-Lead Quad Flat Pack (EIAJ QFP)**
- **84-Lead Plastic Leaded Chip Carrier (PLCC)**
- **Software Compatible with 80960KA/KB/CA/CF Processors**

1

The 80960SA is a member of Intel's i960® 32-bit processor family, which is designed especially for low cost embedded applications. It includes a 512-byte instruction cache and a built-in interrupt controller. The 80960SA has a large register set, multiple parallel execution units and a 16-bit burst bus. Using advanced RISC technology, this high performance processor is capable of execution rates in excess of 7.5 million instructions per second*. The 80960SA is well-suited for a wide range of cost sensitive embedded applications including non-impact printers, network adapters and I/O controllers.

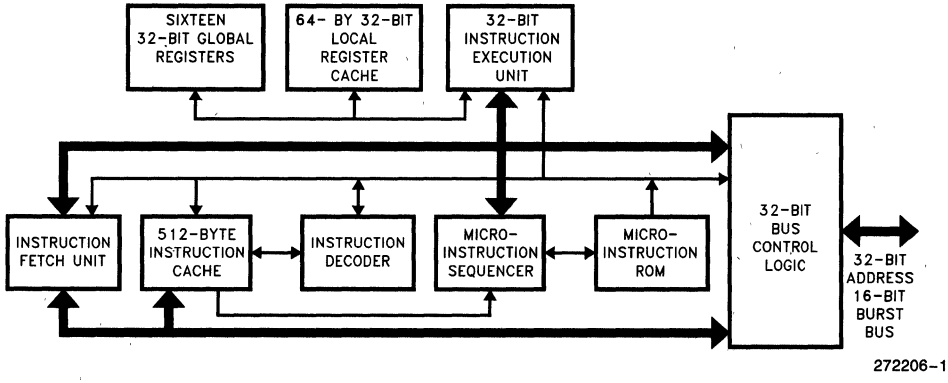


Figure 1. The 80960SA Processor's Highly Parallel Architecture

*Relative to Digital Equipment Corporation's VAX-11/780 at 1 MIPS (VAX-11 is a trademark of Digital Equipment Corporation).

December 1994
Order Number: 272206-002

80960SA
EMBEDDED 32-BIT MICROPROCESSOR WITH 16-BIT BURST DATA BUS

CONTENTS

CONTENTS

PAGE

FIGURES

CONTENTS

PAGE

FIGURES

TABLES

CONTENTS

PAGE

TABLES

1.0 THE i960® PROCESSOR

The 80960SA is a member of the 32-bit architecture from Intel known as the i960 processor family. These microprocessors were especially designed to serve the needs of embedded applications. The embedded market includes applications as diverse as industrial automation, avionics, image processing, graphics and networking. These types of applications require high integration, low power consumption, quick interrupt response times and high performance. Since time to market is critical, embedded microprocessors need to be easy to use in both hardware and software designs.

All members of the i960 processor family share a common core architecture which utilizes RISC technology so that, except for special functions, the family members are object-code compatible. Each new processor in the family adds its own special set of functions to the core to satisfy the needs of a specific application or range of applications in the embedded market.

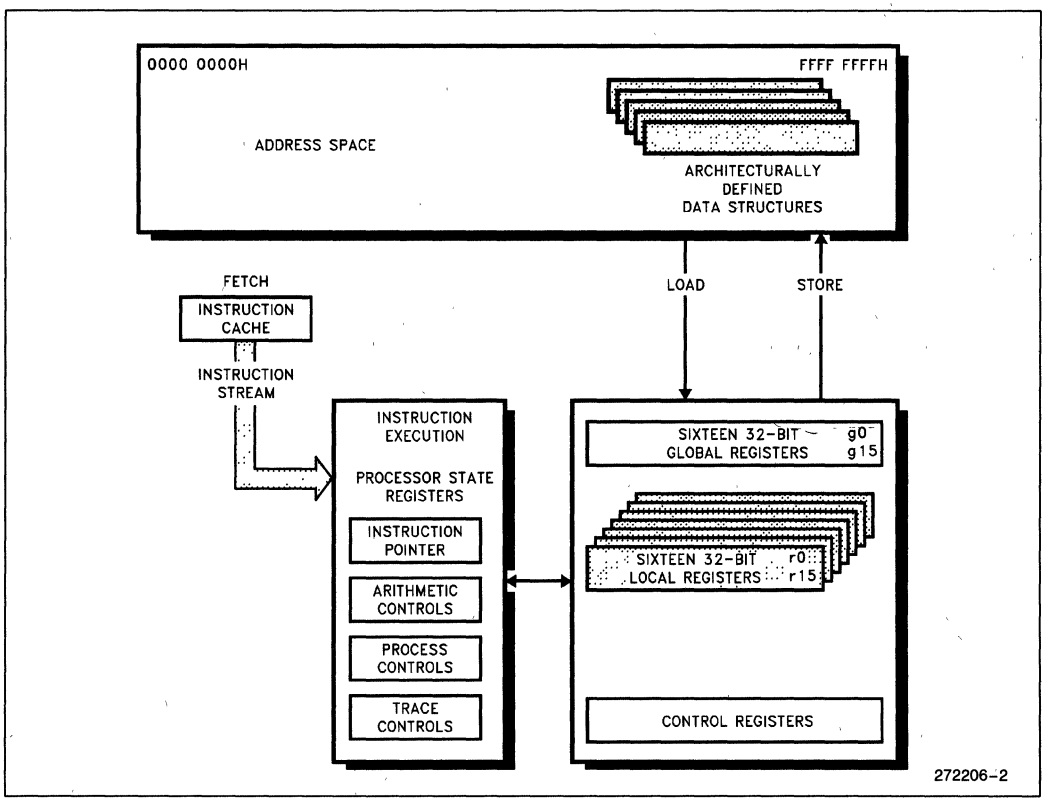


Figure 2. 80960SA Programming Environment

1.1 Key Performance Features

The 80960SA architecture is based on the most recent advances in microprocessor technology and is grounded in Intel's long experience in the design and manufacture of embedded microprocessors. Many features contribute to the 80960SA's exceptional performance:

1. **Large Register Set.** Having a large number of registers reduces the number of times that a processor needs to access memory. Modern compilers can take advantage of this feature to optimize execution speed. For maximum flexibility, the 80960SA provides thirty-two 32-bit registers. (See Figure 2.)
2. **Fast Instruction Execution.** Simple functions make up the bulk of instructions in most programs so that execution speed can be improved by ensuring that these core instructions are executed as quickly as possible. The most frequently executed instructions—such as register-register moves, add/subtract, logical operations and shifts—execute in one to two cycles. (Table 1 contains a list of instructions.)
3. **Load/Store Architecture.** One way to improve execution speed is to reduce the number of times that the processor must access memory to perform an operation. As with other processors based on RISC technology, the 80960SA has a Load/Store architecture. As such, only the LOAD and STORE instructions reference memory; all other instructions operate on registers. This type of architecture simplifies instruction decoding and is used in combination with other techniques to increase parallelism.
4. **Simple Instruction Formats.** All instructions in the 80960SA are 32 bits long and must be aligned on word boundaries. This alignment makes it possible to eliminate the instruction alignment stage in the pipeline. To simplify the instruction decoder, there are only five instruction formats; each instruction uses only one format. (See Figure 3.)
5. **Overlapped Instruction Execution.** Load operations allow execution of subsequent instructions to continue before the data has been returned from memory, so that these instructions can overlap the load. The 80960SA manages this process transparently to software through the use of a register scoreboard. Conditional instructions also make use of a scoreboard so that subsequent unrelated instructions may be executed while the conditional instruction is pending.
6. **Integer Execution Optimization.** When the result of an arithmetic execution is used as an operand in a subsequent calculation, the value is sent immediately to its destination register. At the same time, the value is put on a bypass path to the ALU, thereby saving the time that otherwise would be required to retrieve the value for the next operation.
7. **Bandwidth Optimizations.** The 80960SA gets optimal use of its memory bus bandwidth because the bus is tuned for use with the on-chip instruction cache: instruction cache line size matches the maximum burst size for instruction fetches. The 80960SA automatically fetches four words in a burst and stores them directly in the cache. Due to the size of the cache and the fact that it is continually filled in anticipation of needed instructions in the program flow, the 80960SA is relatively insensitive to memory wait states. The benefit is that the 80960SA delivers outstanding performance even with a low cost memory system.
8. **Cache Bypass.** If a cache miss occurs, the processor fetches the needed instruction then sends it on to the instruction decoder at the same time it updates the cache. Thus, no extra time is spent to load and read the cache.

Table 1. 80960SA Instruction Set

Data Movement	Arithmetic	Logical	Bit and Bit Field
Load Store Move Load Address	Add Subtract Multiply Divide Remainder Modulo Shift Extended Multiply Extended Divide	And Not And And Not Or Exclusive Or Not Or Or Not Nor Exclusive Nor Not Nand Rotate	Set Bit Clear Bit Not Bit Check Bit Alter Bit Scan For Bit Scan Over Bit Extract Modify
Comparison	**Branch**	**Call/Return**	**Fault**
Compare Conditional Compare Compare and Increment Compare and Decrement	Unconditional Branch Conditional Branch Compare and Branch	Call Call Extended Call System Return Branch and Link	Conditional Fault Synchronize Faults
Debug	**Miscellaneous**	**Decimal**	
Modify Trace Controls Mark Force Mark	Atomic Add Atomic Modify Flush Local Registers Modify Arithmetic Controls Scan Byte for Equal Test Condition Code	Move Add and Carry Subtract with Carry	
Synchronous			
Synchronous Load Synchronous Move			

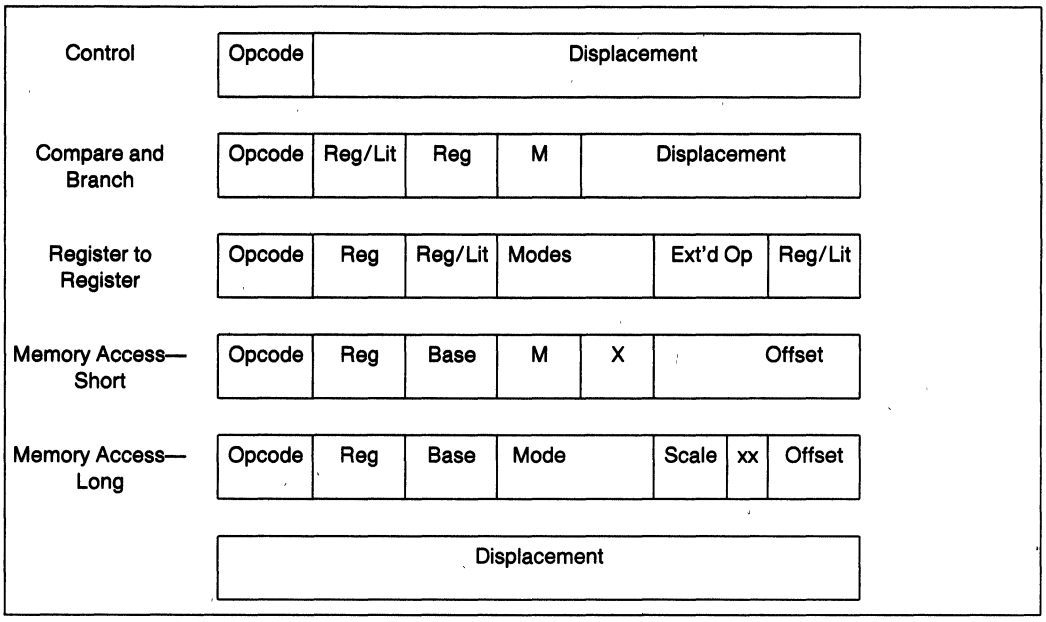


Figure 3. Instruction Formats

1.1.1 MEMORY SPACE AND ADDRESSING MODES

The 80960SA offers a linear programming environment so that all programs running on the processor are contained in a single address space. Maximum address space size is 4 Gigabytes (2^{32} bytes).

For ease of use the 80960SA has a small number of addressing modes, but includes all those necessary to ensure efficient execution of high-level languages such as C. Table 2 lists the memory addressing modes.

Table 2. Memory Addressing Modes

- 12-Bit Offset
- 32-Bit Offset
- Register-Indirect
- Register + 12-Bit Offset
- Register + 32-Bit Offset
- Register + (Index-Register x Scale-Factor)
- Register x Scale Factor + 32-Bit Displacement
- Register + (Index-Register x Scale-Factor) + 32-Bit Displacement

Scale-Factor is 1, 2, 4, 8 or 16

1.1.2 DATA TYPES

The 80960SA recognizes the following data types:

Numeric:

- 8-, 16-, 32- and 64-bit ordinals
- 8-, 16-, 32- and 64-bit integers

Non-Numeric:

- Bit
- Bit Field
- Triple Word (96 bits)
- Quad-Word (128 bits)

1.1.3 LARGE REGISTER SET

The 80960SA programming environment includes a large number of registers. In fact, 32 registers are available at any time. The availability of this many registers greatly reduces the number of memory accesses required to perform algorithms, which leads to greater instruction processing speed.

There are two types of general-purpose register: local and global. The global registers consist of sixteen 32-bit registers (g0 though g15). These registers perform the same function as the general-

purpose registers provided in other popular microprocessors. The term global refers to the fact that these registers retain their contents across procedure calls.

The local registers, on the other hand, are procedure specific. For each procedure call, the 80960SA allocates 16 local registers (r0 through r15). Each local register is 32 bits wide.

1.1.4 MULTIPLE REGISTER SETS

To further increase the efficiency of the register set, multiple sets of local registers are stored on-chip (See Figure 4). This cache holds up to four local register frames, which means that up to three procedure calls can be made without having to access the procedure stack resident in memory.

Although programs may have procedure calls nested many calls deep, a program typically oscillates back and forth between only two to three levels. As a result, with four stack frames in the cache, the probability of having a free frame available on the cache when a call is made is very high. In fact, runs of representative C-language programs show that 80% of the calls are handled without needing to access memory.

If four or more procedures are active and a new procedure is called, the 80960SA moves the oldest local register set in the stack-frame cache to a procedure stack in memory to make room for a new set of registers. Global register g15 is the frame pointer (FP) to the procedure stack.

Global registers are not exchanged on a procedure call, but retain their contents, making them available to all procedures for fast parameter passing.

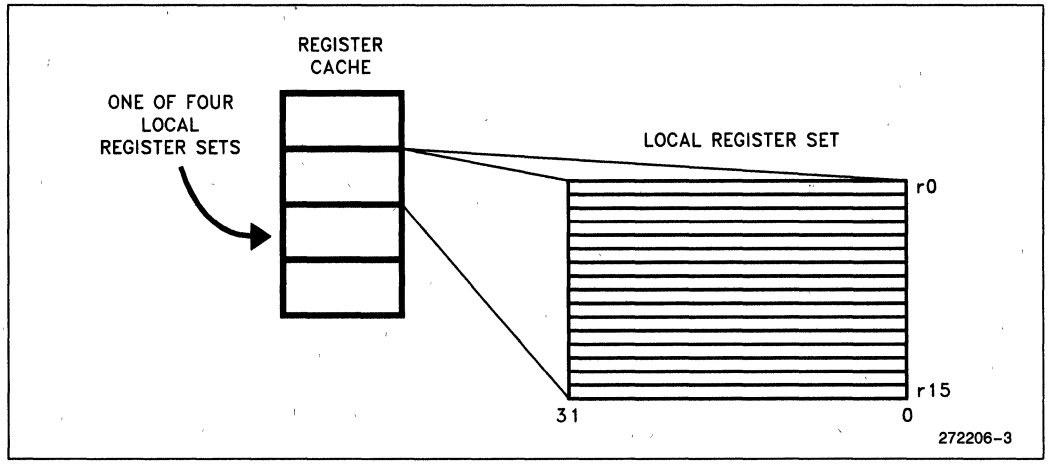


Figure 4. Multiple Register Sets Are Stored On-Chip

1.1.5 INSTRUCTION CACHE

To further reduce memory accesses, the 80960SA includes a 512-byte on-chip instruction cache. The instruction cache is based on the concept of locality of reference; most programs are not usually executed in a steady stream but consist of many branches, loops and procedure calls that lead to jumping back and forth in the same small section of code. Thus, by maintaining a block of instructions in cache, the number of memory references required to read instructions into the processor is greatly reduced.

To load the instruction cache, instructions are fetched in 16-byte blocks; up to four instructions can be fetched at one time. An efficient prefetch algorithm increases the probability that an instruction will already be in the cache when it is needed.

Code for small loops often fits entirely within the cache, leading to a great increase in processing speed since further memory references might not be necessary until the program exits the loop. Similarly, when calling short procedures, the code for the calling procedure is likely to remain in the cache so it will be there on the procedure's return.

1.1.6 REGISTER SCOREBOARDING

The instruction decoder is optimized in several ways. One optimization method is the ability to overlap instructions by using register scoreboarding.

Register scoreboarding occurs when a LOAD moves a variable from memory into a register. When the instruction initiates, a scoreboard bit on the target register is set. Once the register is loaded, the bit is reset. In between, any reference to the register contents is accompanied by a test of the scoreboard bit to ensure that the load has completed before processing continues. Since the processor does not need to wait for the LOAD to complete, it can execute additional instructions placed between the LOAD and the instruction that uses the register contents, as shown in the following example:

```
ld data_2, r4
ld data_2, r5
Unrelated instruction
Unrelated instruction
add r4, r5, r6
```

In essence, the two unrelated instructions between LOAD and ADD are executed "for free" (i.e., take no apparent time to execute) because they are executed while the register is being loaded. Up to three load instructions can be pending at one time with three corresponding scoreboard bits set. By exploiting this feature, system programmers and compiler writers have a useful tool for optimizing execution speed.

1.1.7 HIGH BANDWIDTH BUS

The 80960SA CPU resides on a high-bandwidth address/data bus. The bus provides a direct communication path between the processor and the memory and I/O subsystem interfaces. The processor uses the bus to fetch instructions, manipulate memory and respond to interrupts. Bus features include:

- 16-bit data path multiplexed onto the lower bits of the 32-bit address path
- Eight 16-bit half-word burst capability which allows transfers from 1 to 16 bytes at a time
- High bandwidth reads and writes with 32 Mbytes/s burst (at 20 MHz)

Table 3 defines bus signal names and functions; Table 4 defines other component-support signals such as interrupt lines.

1.1.8 INTERRUPT HANDLING

The 80960SA can be interrupted in one of two ways: by the activation of one of four interrupt pins or by sending a message on the processor's data bus.

The 80960SA is unusual in that it automatically handles interrupts on a priority basis and can keep track of pending interrupts through its on-chip interrupt controller. Two of the interrupt pins can be configured to provide 8259A-style handshaking for expansion beyond four interrupt lines.

1.1.9 DEBUG FEATURES

The 80960SA has built-in debug capabilities. There are two types of breakpoints and six trace modes. Debug features are controlled by two internal 32-bit registers, the Process-Controls Word and the Trace-Controls Word. By setting bits in these control words, a software debug monitor can closely control how the processor responds during program execution.

The 80960SA provides two hardware breakpoint registers on-chip which, by using a special command, can be set to any value. When the instruction pointer matches either breakpoint register value, the breakpoint handling routine is automatically called.

The 80960SA also provides software breakpoints through the use of two instructions: MARK and FMARK. These can be placed at any point in a program and cause the processor to halt execution at that point and call the breakpoint handling routine. The breakpoint mechanism is easy to use and provides a powerful debugging tool.

Tracing is available for instructions (single step execution), calls and returns and branching. Each trace type may be enabled separately by a special debug instruction. In each case, the 80960SA executes the instruction first and then calls a trace handling routine (usually part of a software debug monitor). Further program execution is halted until the routine completes, at which time execution resumes at the next instruction. The 80960SA's tracing mechanisms, implemented completely in hardware, greatly simplify the task of software test and debug.

1.1.10 FAULT DETECTION

The 80960SA has an automatic mechanism to handle faults. Fault types include trace and arithmetic faults. When the processor detects a fault, it automatically calls the appropriate fault handling routine and saves the current instruction pointer and necessary state information to make efficient recovery possible. Like interrupt handling routines, fault handling routines are usually written to meet the needs of specific applications and are often included as part of the operating system or kernel.

For each of the fault types, there are numerous subtypes that provide specific information about a fault. The fault handler can use this specific information to respond correctly to the fault.

1.1.11 BUILT-IN TESTABILITY

Upon reset, the 80960SA automatically conducts an exhaustive internal test of its major blocks of logic. Then, before executing its first instruction, it does a zero check sum on the first eight words in memory to ensure that the memory image was programmed correctly. If a problem is discovered at any point during the self-test, the 80960SA asserts its $\overline{\text{FAIL}}$ pin and will not begin program execution. Self test takes approximately 24,000 cycles to complete.

System manufacturers can use the 80960SA's self-test feature during incoming parts inspection. No special diagnostic programs need to be written. The test is both thorough and fast. The self-test capability helps ensure that defective parts are discovered before systems are shipped and, once in the field, the self-test makes it easier to distinguish between problems caused by processor failure and problems resulting from other causes.

1.1.12 CHMOS

The 80960SA is fabricated using Intel's CHMOS IV (Complementary High Speed Metal Oxide Semiconductor) process. The 80960SA is available at 10 and 16 MHz in the QFP package and at 10, 16 and 20 MHz in the PLCC package.

Table 3. 80960SA Pin Description: Bus Signals

Name	Type	Description
CLK2	I	**SYSTEM CLOCK** provides the fundamental timing for 80960SA systems. It is divided by two inside the 80960SA to generate the internal processor clock.
A31:16	O T.S.	**ADDRESS BUS** carries the upper 16 bits of the 32-bit physical address to memory. It is valid throughout the burst cycle; no latch is required.
AD15:1, D0	I/O T.S.	**ADDRESS/DATA BUS** carries the low order 32-bit addresses and 16-bit data to and from memory. AD15:4 must be latched since the cycle following the address cycle carries data on the bus.
A3:1	O T.S.	**ADDRESS BUS** carries the word addresses of the 32-bit address to memory. These three bits are incremented during a burst access indicating the next word address of the burst access. Note that A3:1 are duplicated with AD3:1 during the address cycle.
ALE	O T.S.	**ADDRESS LATCH ENABLE** indicates the transfer of a physical address. ALE is asserted during a T_a cycle and deasserted before the beginning of the T_d state. It is active high and floats to a high impedance state during a hold cycle (T_h).
$\overline{AS}$	O T.S.	**ADDRESS STATUS** indicates an address state. $\overline{AS}$ is asserted every T_a state and deasserted during the following T_d state. $\overline{AS}$ is driven HIGH during reset.
W/$\overline{R}$	O T.S.	**WRITE/READ** specifies, during a T_a cycle, whether the operation is a write or read. It is latched on-chip and remains valid during T_d cycles.
$\overline{DEN}$	O T.S.	**DATA ENABLE** is asserted during T_d cycles and indicates transfer of data on the AD lines. The AD lines should not be driven by an external source unless $\overline{DEN}$ is asserted. When $\overline{DEN}$ is asserted, outputs from the previous cycle are guaranteed to be three-stated. In addition, $\overline{DEN}$ deasserted indicates inputs have been captured and therefore input hold times can be disregarded. $\overline{DEN}$ is driven HIGH during reset.
DT/$\overline{R}$	O T.S.	**DATA TRANSMIT/RECEIVE** indicates the direction of data transfer to and from the bus. It is low during T_a and T_d cycles for a read or interrupt acknowledgment; it is high during T_a and T_d cycles for a write. DT/$\overline{R}$ never changes state when $\overline{DEN}$ is asserted. DT/$\overline{R}$ is driven HIGH during reset.
$\overline{READY}$	I	**READY** indicates that data on AD lines can be sampled or removed. If $\overline{READY}$ is not asserted during a T_d cycle, the T_d cycle is extended to the next cycle by inserting a wait state (T_w).
$\overline{LOCK}$	I/O O.D.	**BUS LOCK** prevents bus masters from gaining control of the bus during Read/Modify/Write (RMW) cycles. The processor or any bus agent may assert $\overline{LOCK}$. At the start of a RMW operation, the processor examines the $\overline{LOCK}$ pin. If the pin is already asserted, the processor waits until it is not asserted. If the pin is not asserted, the processor asserts $\overline{LOCK}$ during the T_a cycle of the read transaction. The processor deasserts $\overline{LOCK}$ in the T_a cycle of the write transaction. During the time $\overline{LOCK}$ is asserted, a bus agent can perform a normal read or write but not a RMW operation. The processor also asserts $\overline{LOCK}$ during interrupt-acknowledge transactions. Do not leave $\overline{LOCK}$ unconnected. It must be pulled HIGH for the processor to function properly. **ONCE MODE:** The $\overline{LOCK}$ pin is sampled during reset. If it is asserted LOW at the end of reset, all outputs will be three-stated until the part is reset again. ONCE mode is used in conjunction with an in-circuit emulator.

I/O = Input/Output, O = Output, I = Input, O.D. = Open Drain, T.S. = Three-state

Table 3. 80960SA Pin Description: Bus Signals (Continued)

Name	Type	Description
$\overline{BE1:0}$	O T.S.	**BYTE ENABLE LINES** specify which data bytes (up to two) on the bus take part in the current bus cycle. $\overline{BE1}$ corresponds to AD15:8; $\overline{BE0}$ corresponds to AD7:1, D0. The byte enable lines are asserted appropriately during each data cycle. **INITIALIZATION FAILURE** indicates that the processor has failed to initialize correctly. The failure state is indicated by a combination of $\overline{BLAST}$ asserted and $\overline{BE1:0}$ not asserted. This condition occurs after $\overline{RESET}$ is deasserted and before the first bus transaction begins. $\overline{FAIL}$ is asserted while the processor performs a self-test. If the self-test completes successfully, $\overline{FAIL}$ is deasserted. The processor then performs a zero checksum on the first eight words of memory. If it fails, $\overline{FAIL}$ is asserted for a second time and remains asserted; if it passes, system initialization continues and $\overline{FAIL}$ remains deasserted.
HOLD	I	**HOLD:** A request from an external bus master to acquire the bus. When the processor receives HOLD and grants bus control to another master, it floats its three-state bus lines, then asserts HLDA and enters the T_h state. When HOLD is deasserted, the processor deasserts HLDA and enters the T_i or T_a state.
HLDA	O T.S.	**HOLD ACKNOWLEDGE:** Notifies an external bus master that the processor has relinquished control of the bus. This signal is always driven. At reset it is driven LOW.
$\overline{BLAST}/\overline{FAIL}$	O T.S.	**BURST LAST** indicates the last data cycle (T_d) of a burst access. It is asserted low during the last T_d and associated with T_w cycles in a burst access. **INITIALIZATION FAILURE** indicates that the processor has failed to initialize correctly. The failure state is indicated by a combination of $\overline{BLAST}$ asserted and $\overline{BE1:0}$ not asserted. This condition occurs after $\overline{RESET}$ is deasserted and before the first bus transaction begins. FAIL is asserted while the processor performs a self-test. If the self-test completes successfully, $\overline{FAIL}$ is deasserted. The processor then performs a zero checksum on the first eight words of memory. If it fails, $\overline{FAIL}$ is asserted for a second time and remains asserted; if it passes, system initialization continues and $\overline{FAIL}$ remains deasserted.

I/O = Input/Output, O = Output, I = Input, O.D. = Open Drain, T.S. = Three-state

Table 4. 80960SA Pin Description: Support Signals

Name	Type	Description
$\overline{\text{RESET}}$	I	**RESET** clears the processor's internal logic and causes it to reinitialize. During $\overline{\text{RESET}}$ assertion, the input pins are ignored (except for $\overline{\text{INT0}}$, INT1, $\overline{\text{INT3}}$, $\overline{\text{LOCK}}$), the three-state output pins are placed in a HIGH impedance state (except for DT/$\overline{\text{R}}$, $\overline{\text{DEN}}$, and $\overline{\text{AS}}$) and other output pins are placed in their non-asserted states. $\overline{\text{RESET}}$ must be asserted for at least 41 CLK2 cycles for a predictable reset. Optionally, for a synchronous reset, the LOW and HIGH transition of $\overline{\text{RESET}}$ should occur after the rising edge of both CLK2 and the external bus CLK and before the next rising edge of CLK2. The interrupt pins indicate the initialization sequence executed. Typical initialization requires driving only $\overline{\text{INT0}}$ and $\overline{\text{INT3}}$ to a HIGH state. The reset conditions follow:

$\overline{\text{INT0}}$	INT1	$\overline{\text{INT3}}$	$\overline{\text{LOCK}}$	Action Taken
1	x	1	1	Run-self-test (core initialization)
0	0	1	1	Disable self-test
0	1	x	x	Reserved
x	x	0	x	Reserved
x	x	x	0	ONCE mode (see $\overline{\text{LOCK}}$ pin)

$\overline{\text{INT0}}$	I	**INTERRUPT 0** indicates a pending interrupt. To signal an interrupt in a synchronous system, this pin—as well as the other interrupt pins—must be enabled by being deasserted for at least one bus cycle and then asserted for at least one additional bus cycle. In an asynchronous system the pin must remain deasserted for at least two system clock cycles and then asserted for at least two more system clock cycles. The interrupt control register must be programmed with an interrupt vector before using this pin. $\overline{\text{INT0}}$ is sampled during reset to determine if the self-test sequence is to be executed.
INT1	I	**INTERRUPT 1,** like $\overline{\text{INT0}}$, provides direct interrupt signaling. INT1 is sampled during reset to determine if the self-test sequence is to be executed.
INT2/INTR	I	**INTERRUPT2/INTERRUPT REQUEST:** The interrupt control register determines how this pin is interpreted. If INT2, it has the same interpretation as the $\overline{\text{INT0}}$ and INT1 pins. If INTR, it is used to receive an interrupt request from an external interrupt controller.
$\overline{\text{INT3}}$/$\overline{\text{INTA}}$	I/O T.S.	**INTERRUPT3/INTERRUPT ACKNOWLEDGE:** The interrupt control register determines how this pin is interpreted. If $\overline{\text{INT3}}$, it has the same interpretation as the $\overline{\text{INT0}}$ and INT1 pins. If $\overline{\text{INTA}}$, it is used as an output to control interrupt-acknowledge transactions. The $\overline{\text{INTA}}$ output is latched on-chip and remains valid during T_d cycles; as an output, it is open-drain. $\overline{\text{INT3}}$ must be pulled HIGH during reset.
NC	N/A	**NOT CONNECTED** indicates pins should not be connected. Never connect any pin marked NC; these pins may be reserved for factory use.

I/O = Input/Output, O = Output, I = Input, O.D. = Open Drain, T.S. = Three-state

2.0 ELECTRICAL SPECIFICATIONS

2.1 Power and Grounding

The 80960SA is implemented in CHMOS IV technology and therefore has modest power requirements. Its high clock frequency and numerous output buffers (address/data, control, error and arbitration signals) can cause power surges as multiple output buffers simultaneously drive new signal levels. For clean on-chip power distribution, V_{CC} and V_{SS} pins separately feed the device's functional units. Power and ground connections must be made to all 80960SA power and ground pins. On the circuit board, all V_{CC} pins must be strapped closely together, preferably on a power plane; all V_{SS} pins should be strapped together, preferably on a ground plane.

2.2 Power Decoupling Recommendations

Place a liberal amount of decoupling capacitance near the 80960SA. When driving the bus the processor can cause transient power surges, particularly when connected to a large capacitive load.

Low inductance capacitors and interconnects are recommended for best high frequency electrical performance. Inductance is reduced by shortening board traces between the processor and decoupling capacitors as much as possible.

2.3 Connection Recommendations

For reliable operation, always connect unused inputs to an appropriate signal level. In particular, if one or more interrupt lines are not used, they should be pulled up. No inputs should ever be left floating.

The $\overline{LOCK}$ open drain pin requires a pullup resistor whether or not the pin is used as an output. Figure 5 shows the recommended resistor value.

Do not connect external logic to pins marked NC.

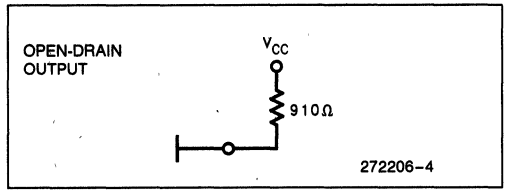


Figure 5. Connection Recommendation for $\overline{LOCK}$

2.4 Characteristic Curves

Figure 6 shows typical supply current requirements over the operating temperature range of the processor at supply voltage (V_{CC}) of 5V. Figure 7 shows the typical power supply current (I_{CC}) that the 80960SA requires at various operating frequencies when measured at three input voltage (V_{CC}) levels.

For a given output current (I_{OL}) the curve in Figure 8 shows the worst case output low voltage (V_{OL}). Figure 9 shows the typical capacitive derating curve for the 80960SA measured from 1.5V on the system clock (CLK) to 0.8V on the falling edge and 2.0V on the rising edge of the bus address/data (AD) signals.

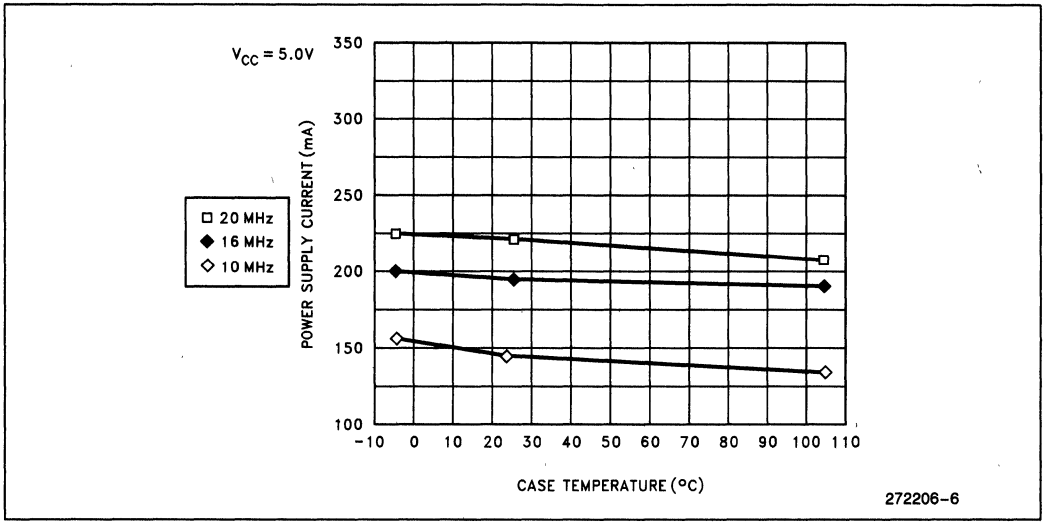


Figure 6. Typical Supply Current vs Case Temperature

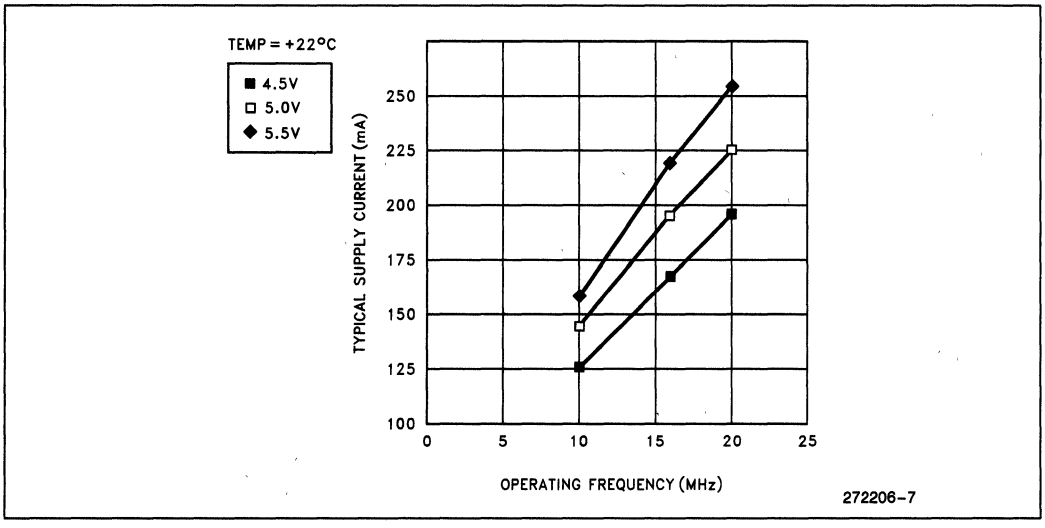


Figure 7. Typical Current vs Frequency (Room Temp)

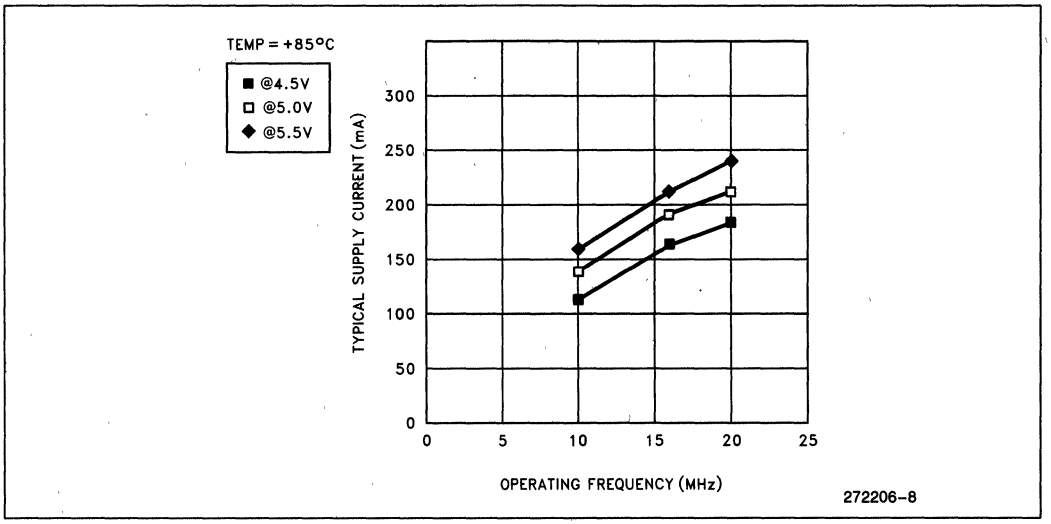


Figure 8. Typical Current vs Frequency (Hot Temp)

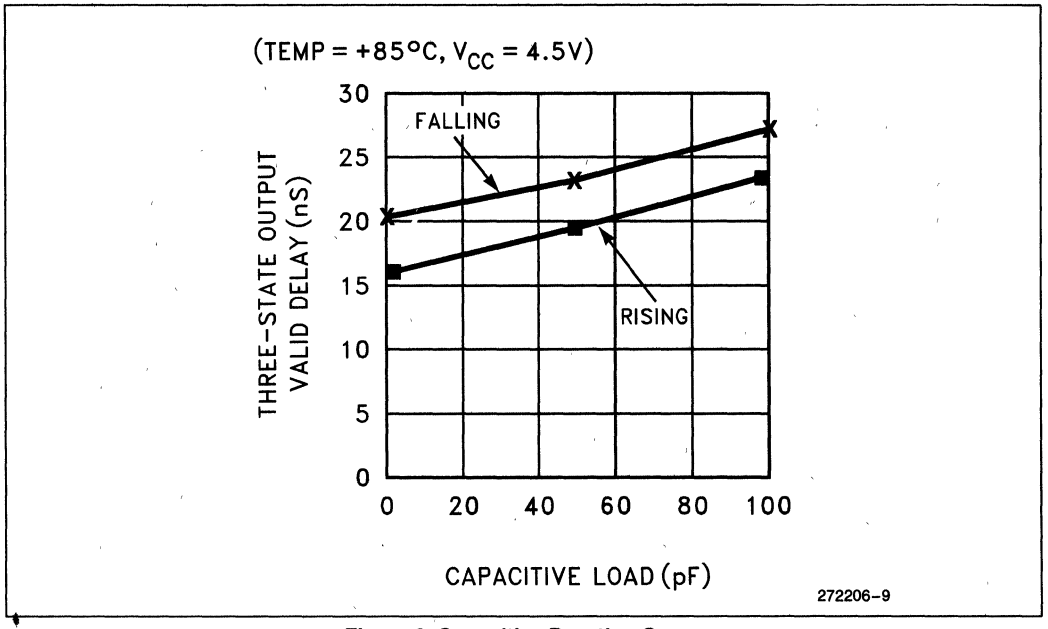


Figure 9. Capacitive Derating Curve

2.5 Test Load Circuit

Figure 10 illustrates the load circuit used to test the 80960SA's output pins.

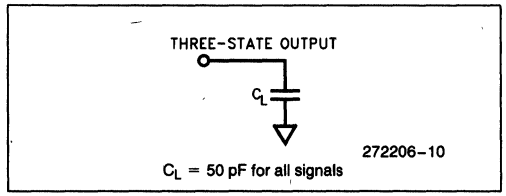


Figure 10. Test Load Circuit for Three-State Output Pins

2.6 Absolute Maximum Ratings*

Operating Temperature (PLCC) . . . 0°C to +85°C Case
Operating Temperature (QFP) . . . 0°C to +100°C Case
Storage Temperature . . . . . −65°C to +150°C
Voltage on Any Pin (PLCC) . . . −0.5V to V_{CC} +0.5V
Voltage on Any Pin (QFP) . . . −0.25V to V_{CC} +0.25V
Power Dissipation . . . . . 1.9W (20 MHz)

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

2.7 DC Characteristics

80960SA (10 and 16 MHz QFP) T_{CASE} = 0°C to +100°C, V_{CC} = 5V ±5%
80960SA (10 and 16 MHz PLCC) T_{CASE} = 0°C to +85°C, V_{CC} = 5V ±10%
80960SA (20 MHz PLCC) T_{CASE} = 0°C to +85°C, V_{CC} = 5V ±5%

Table 5. DC Characteristics

Symbol	Parameter	Min	Max	Units	Notes
V_{IL}	Input Low Voltage	−0.3	+0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} +0.3	V	
V_{CL}	CLK2 Input Low Voltage	−0.3	+0.8	V	
V_{CH}	CLK2 Input High Voltage	0.7 V_{CC}	V_{CC} +0.3	V	
V_{OL}	Output Low Voltage		0.45 0.45 0.60	V V V	I_{OL} = 4.0 mA I_{OL} = 6 mA, $\overline{LOCK}$ Pin I_{OL} = 20 mA, $\overline{LOCK}$ Pin
V_{OH}	Output High Voltage	2.4		V	All TS, −2.5 mA (1)
I_{CC}	Power Supply Current: 10 MHz-QFP 10 MHz-PLCC 16 MHz-PLCC 20 MHz-PLCC		 240 240 300 340	 mA mA mA mA	 T_{CASE} = 0°C T_{CASE} = 0°C T_{CASE} = 0°C T_{CASE} = 0°C
I_{LI1}	Input Leakage Current, Except $\overline{INT0}$, $\overline{LOCK}$		±15	µA	$0 \le V_{IN} \le V_{CC}$
I_{LI2}	Input Leakage Current, $\overline{INT0}$, $\overline{LOCK}$		−300	µA	V_{IN} = 0.45V(2)
I_{OL}	Output Leakage Current		±15	µA	
C_{IN}	Input Capacitance		10	pF	f_C = 1 MHz(3)
C_O	Output Capacitance		12	pF	f_C = 1 MHz(3)
C_{CLK}	Clock Capacitance		10	pF	f_C = 1 MHz(3)

NOTES:
1. Not measured for open-drain output.
2. $\overline{INT0}$ and $\overline{LOCK}$ have internal pullup devices.
3. Input, output and clock capacitance are not tested.

2.8 AC Specifications

This section describes the AC specifications for the 80960SA pins. All input and output timings are specified relative to the 1.5V level of the rising edge of CLK2 and refer to the time at which the signal crosses 1.5V (for output delay and input setup). All AC testing should be done with input voltages of 0.4V and 2.4V, except for the clock (CLK2) which should be tested with input voltages of 0.45V and 0.7 x V_{CC}. See Figure 11 and Tables 6, 7 and 8 for timing relationships for the 80960SA signals.

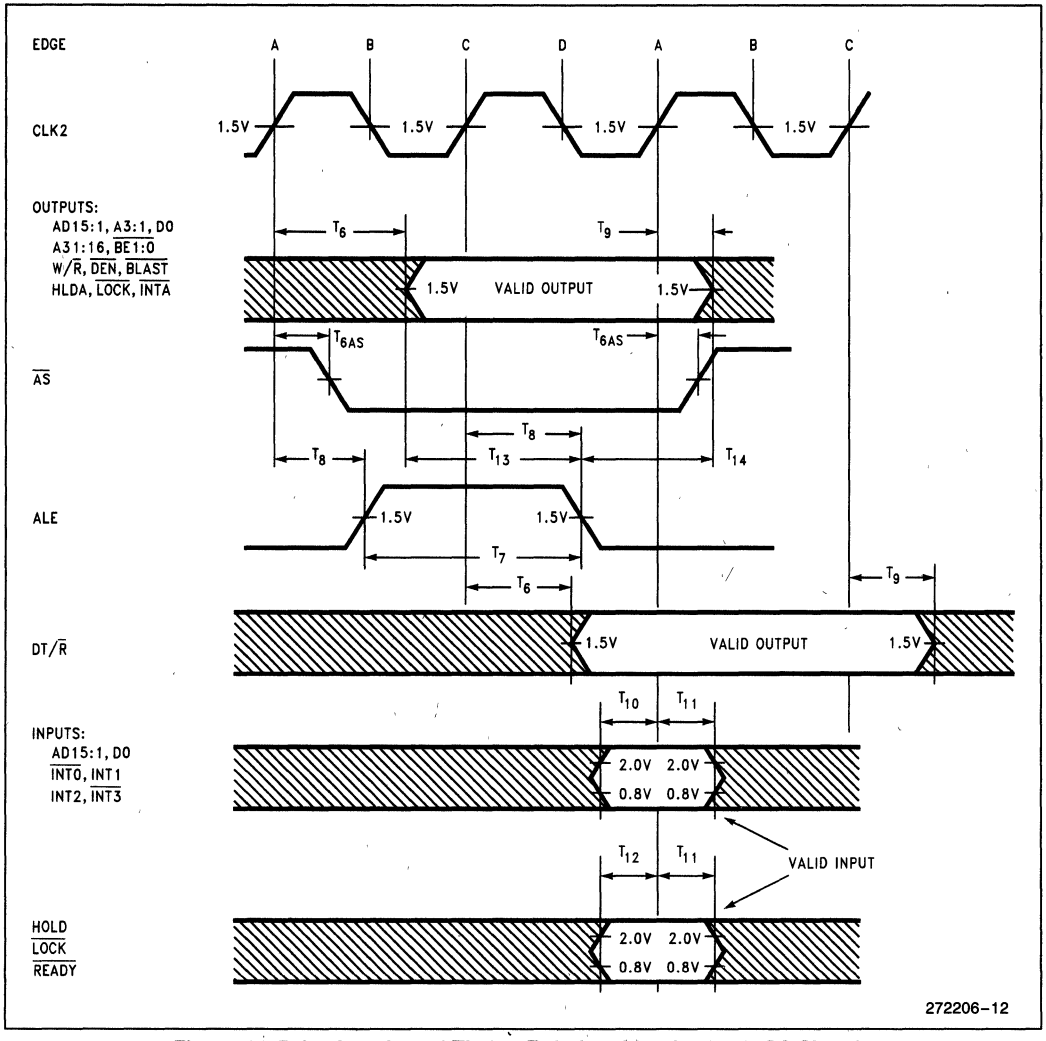


Figure 11. Drive Levels and Timing Relationships for 80960SA Signals

2.8.1 AC SPECIFICATION TABLES

Table 6. 80960SA AC Characteristics (10 MHz)

Symbol	Parameter	Min	Max	Units	Notes
Input Clock					
T_1	Processor Clock Period (CLK2)	50	125	ns	V_{IN} = 1.5V
T_2	Processor Clock Low Time (CLK2)	8		ns	V_T = 10% Point = V_{CL} + (V_{CH} − V_{CL}) x 0.1
T_3	Processor Clock High Time (CLK2)	8		ns	V_T = 90% Point = V_{CL} + (V_{CH} − V_{CL}) x 0.9
T_4	Processor Clock Fall Time (CLK2)		10	ns	V_T = 90% to 10% Point (1)
T_5	Processor Clock Rise Time (CLK2)		10	ns	V_T = 10% to 90% Point (1)
Synchronous Outputs					
T_6	Output Valid Delay	2	31	ns	
T_{6AS}	$\overline{AS}$ Output Valid Delay	2	25	ns	
T_7	ALE Width	T_1−11		ns	
T_8	ALE Output Valid Delay	4	33	ns	
T_9	Output Float Delay	2	20	ns	(2)
Synchronous Inputs					
T_{10}	Input Setup 1	10		ns	
T_{11}	Input Hold	2		ns	
T_{12}	Input Setup 2	13		ns	
T_{13}	Setup to ALE Inactive	10		ns	
T_{14}	Hold After ALE Inactive	8		ns	
T_{15}	$\overline{RESET}$ Hold	3		ns	(3)
T_{16}	$\overline{RESET}$ Setup	5		ns	(3)
T_{17}	$\overline{RESET}$ Width	2050		ns	41 CLK2 Periods Minimum

NOTES:
1. Processor clock (CLK2) rise time and fall time are not tested.
2. A float condition occurs when the maximum output current becomes less than I_{LO}. Float delay is not tested, but should be no longer than the valid delay.
3. Meeting $\overline{RESET}$ setup and hold times is an optional method of synchronizing your clocks. If you decide to use an asynchronous reset, synchronizing the clock can be accomplished by using $\overline{AS}$.

Table 7. 80960SA AC Characteristics (16 MHz)

Symbol	Parameter	Min	Max	Units	Notes
Input Clock					
T_1	Processor Clock Period (CLK2)	31.25	125	ns	V_{IN} = 1.5V
T_2	Processor Clock Low Time (CLK2)	8		ns	V_T = 10% Point = V_{CL} + (V_{CH} − V_{CL}) x 0.1
T_3	Processor Clock High Time (CLK2)	8		ns	V_T = 90% Point = V_{CL} + (V_{CH} − V_{CL}) x 0.9
T_4	Processor Clock Fall Time (CLK2)		10	ns	V_T = 90% to 10% Point (1)
T_5	Processor Clock Rise Time (CLK2)		10	ns	V_T = 10% to 90% Point (1)
Synchronous Outputs					
T_6	Output Valid Delay	2	25	ns	
T_{6AS}	$\overline{AS}$ Output Valid Delay	2	21	ns	
T_7	ALE Width	T_1−11		ns	
T_8	ALE Output Valid Delay	2	22	ns	
T_9	Output Float Delay	2	20	ns	(2)
Synchronous Inputs					
T_{10}	Input Setup 1	10		ns	
T_{11}	Input Hold	2		ns	
T_{12}	Input Setup 2	13		ns	
T_{13}	Setup to ALE Inactive	10		ns	
T_{14}	Hold After ALE Inactive	8		ns	
T_{15}	$\overline{RESET}$ Hold	3		ns	(3)
T_{16}	$\overline{RESET}$ Setup	5		ns	(3)
T_{17}	$\overline{RESET}$ Width	1281		ns	41 CLK2 Periods Minimum

NOTES:
1. Processor clock (CLK2) rise time and fall time are not tested.
2. A float condition occurs when the maximum output current becomes less than I_{LO}. Float delay is not tested, but should be no longer than the valid delay.
3. Meeting $\overline{RESET}$ setup and hold times is an optional method of synchronizing your clocks. If you decide to use an asynchronous reset, synchronizing the clock can be accomplished by using $\overline{AS}$.

Table 8. 80960SA AC Characteristics (20 MHz)

Symbol	Parameter	Min	Max	Units	Notes
Input Clock					
T_1	Processor Clock Period (CLK2)	25	125	ns	V_{IN} = 1.5V
T_2	Processor Clock Low Time (CLK2)	6		ns	V_T = 10% Point = V_{CL} + (V_{CH} − V_{CL}) x 0.1
T_3	Processor Clock High Time (CLK2)	6		ns	V_T = 90% Point = V_{CL} + (V_{CH} − V_{CL}) x 0.9
T_4	Processor Clock Fall Time (CLK2)		10	ns	V_T = 90% to 10% Point (1)
T_5	Processor Clock Rise Time (CLK2)		10	ns	V_T = 10% to 90% Point (1)
Synchronous Outputs					
T_6	Output Valid Delay	2	20	ns	
T_{6AS}	$\overline{AS}$ Output Valid Delay	2	20	ns	
T_7	ALE Width	T_1−11		ns	
T_8	ALE Output Valid Delay	2	18	ns	
T_9	Output Float Delay	2	17	ns	(2)
Synchronous Inputs					
T_{10}	Input Setup 1	7		ns	
T_{11}	Input Hold	2		ns	
T_{12}	Input Setup 2	13		ns	
T_{13}	Setup to ALE Inactive	10		ns	
T_{14}	Hold After ALE Inactive	8		ns	
T_{15}	$\overline{RESET}$ Hold	3		ns	(3)
T_{16}	$\overline{RESET}$ Setup	5		ns	(3)
T_{17}	$\overline{RESET}$ Width	1025		ns	41 CLK2 Periods Minimum

NOTES:
1. Processor clock (CLK2) rise time and fall time are not tested.
2. A float condition occurs when the maximum output current becomes less than I_{LO}. Float delay is not tested, but should be no longer than the valid delay.
3. Meeting $\overline{RESET}$ setup and hold times is an optional method of synchronizing your clocks. If you decide to use an asynchronous reset, synchronizing the clock can be accomplished by using $\overline{AS}$.

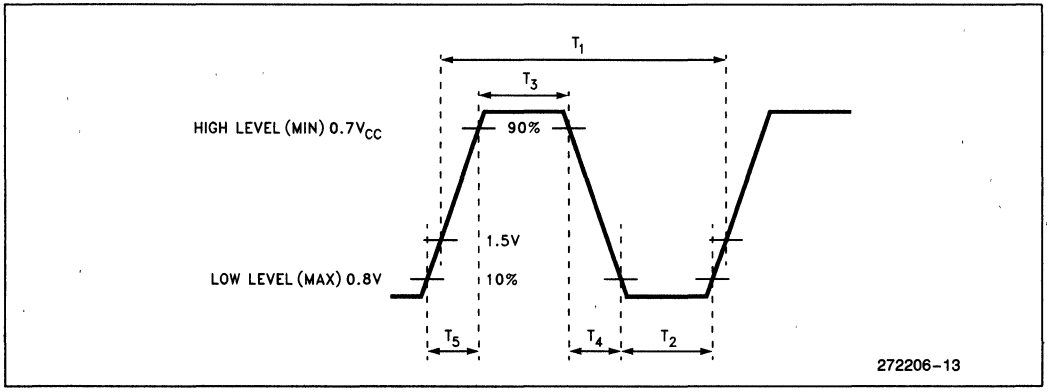


Figure 12. Processor Clock Pulse (CLK2)

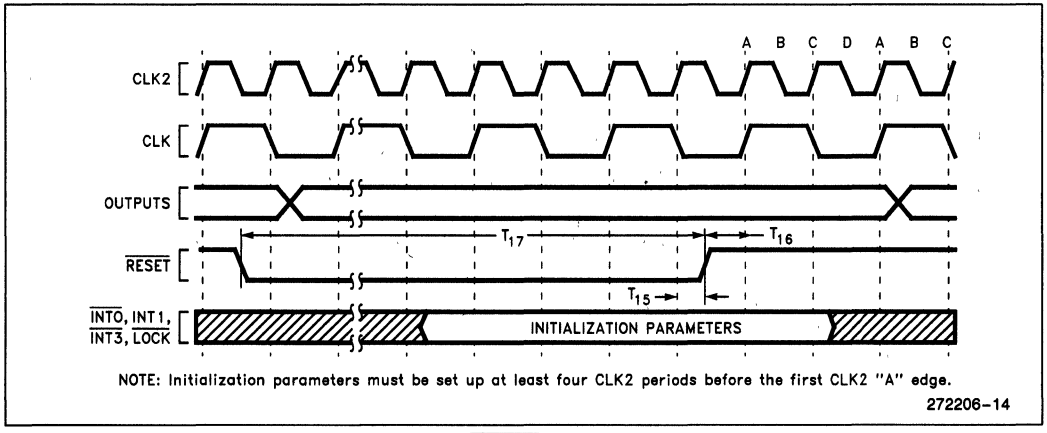


Figure 13. $\overline{\text{RESET}}$ Signal Timing

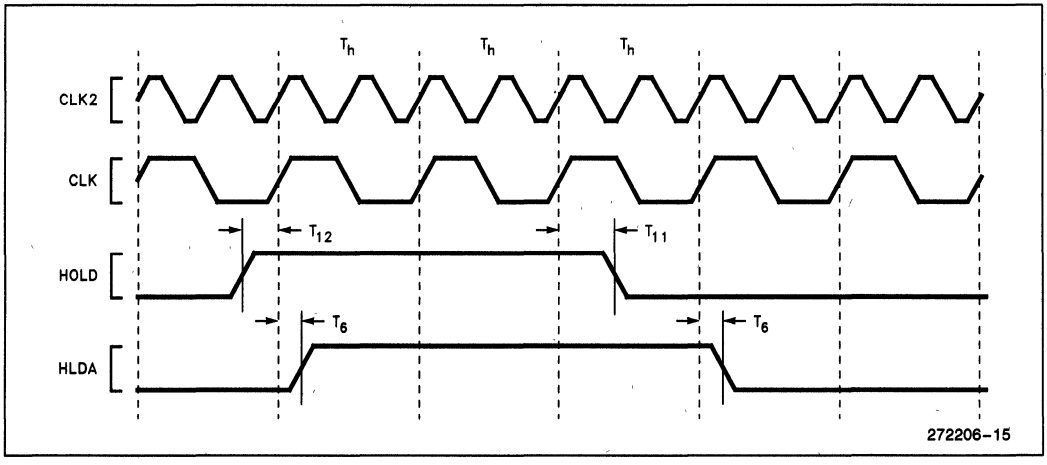


Figure 14. HOLD Timing

3.0 MECHANICAL DATA

3.1 Packaging

The 80960SA is available in two package types:

- 80-lead quad flat pack (EIAJ QFP). Shown in Figure 15.
- 84-lead plastic leaded chip carrier (PLCC). Shown in Figure 16.

Dimensions for both package types are given in the Intel *Packaging* handbook (Order #240800).

3.2 Pin Assignment

The QFP and PLCC have different pin assignments. The QFP pins are numbered in order from 1 to 80 around the package perimeter. The PLCC pins are numbered in order from 1 to 84 around the package perimeter. Table 9 and Table 10 list the function of each QFP pin; Table 11 and Table 12 list the function of each PLCC pin.

V_{CC} and GND connections must be made to multiple V_{CC} and GND pins. Each V_{CC} and GND pin must be connected to the appropriate voltage or ground and externally strapped close to the package. It is recommended that you include separate power and ground planes in your circuit board for power distribution.

Pins identified as N.C. (No Connect) should never be connected.

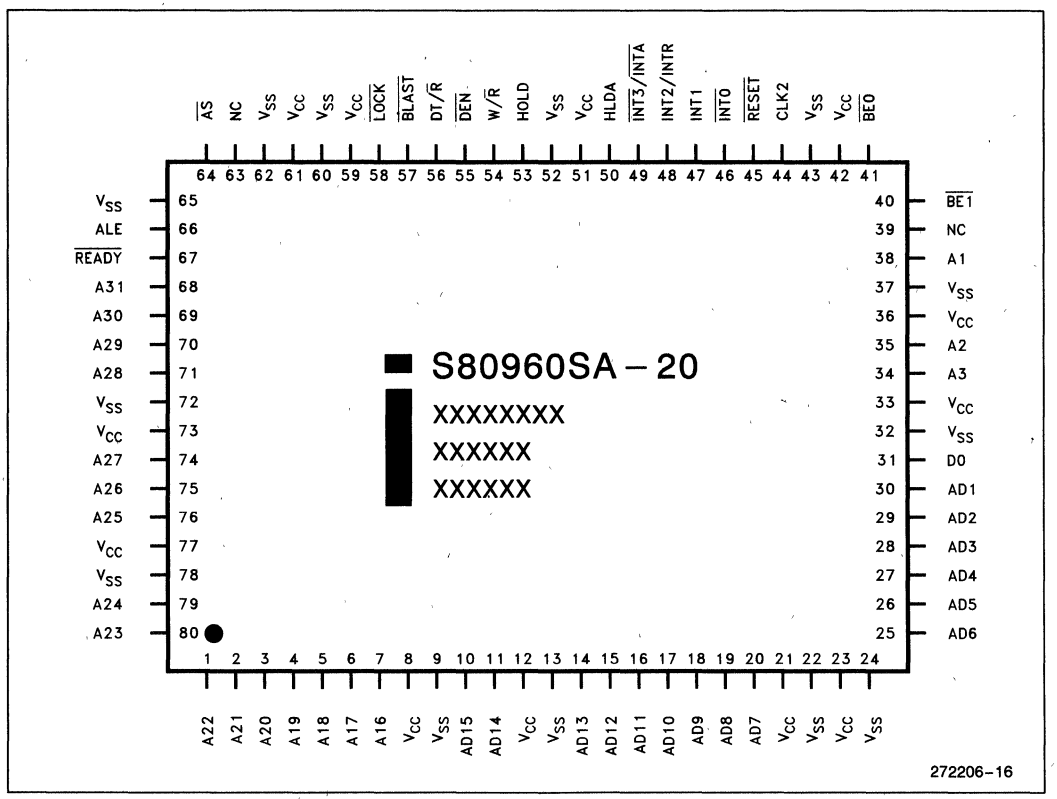


Figure 15. 80-Lead EIAJ Quad Flat Pack (QFP) Package

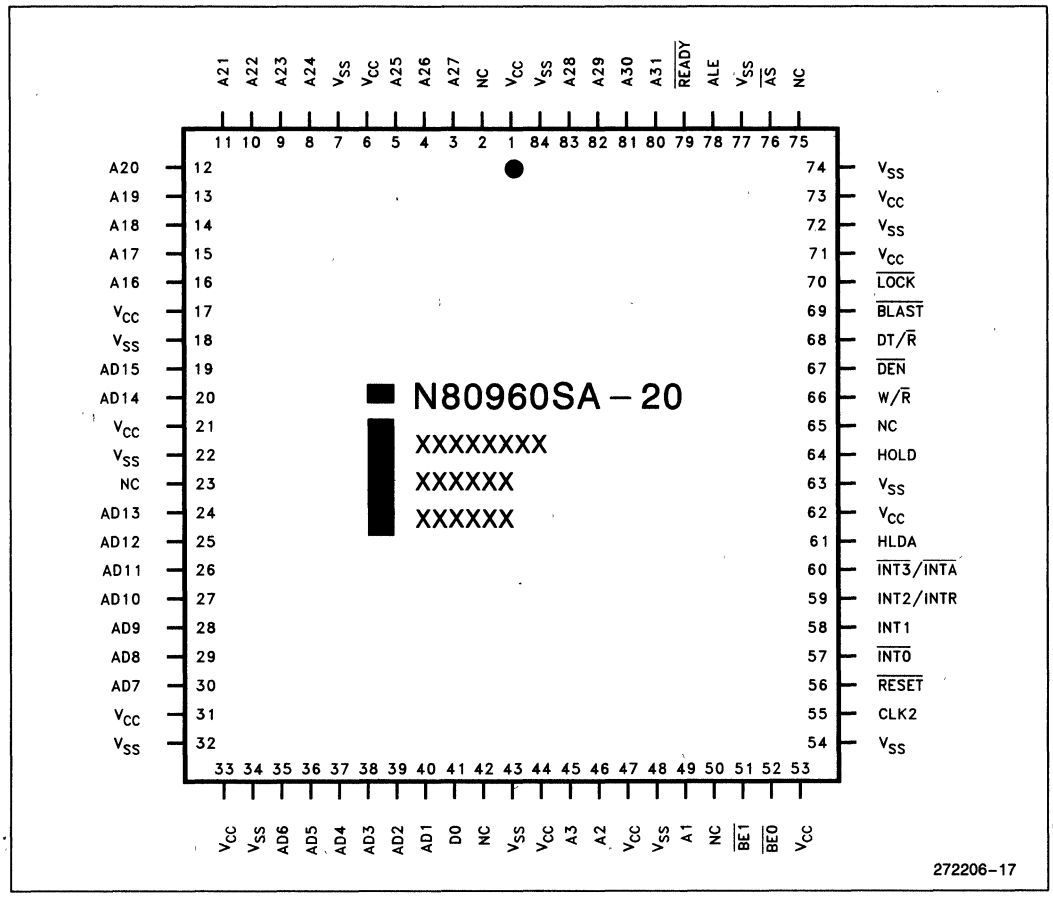


Figure 16. 84-Lead Plastic Leaded Chip Carrier (PLCC) Package

3.3 Pinout

Table 9. 80960SA QFP Pinout—In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	A22	21	V_{CC}	41	$\overline{BE0}$	61	V_{CC}
2	A21	22	V_{SS}	42	V_{CC}	62	V_{SS}
3	A20	23	V_{CC}	43	V_{SS}	63	N.C.
4	A19	24	V_{SS}	44	CLK2	64	$\overline{AS}$
5	A18	25	AD6	45	$\overline{RESET}$	65	V_{SS}
6	A17	26	AD5	46	$\overline{INT0}$	66	ALE
7	A16	27	AD4	47	INT1	67	$\overline{READY}$
8	V_{CC}	28	AD3	48	INT2/INTR	68	A31
9	V_{SS}	29	AD2	49	$\overline{INT3}/\overline{INTA}$	69	A30
10	AD15	30	AD1	50	HLDA	70	A29
11	AD14	31	D0	51	V_{CC}	71	A28
12	V_{CC}	32	V_{SS}	52	V_{SS}	72	V_{SS}
13	V_{SS}	33	V_{CC}	53	HOLD	73	V_{CC}
14	AD13	34	A3	54	W/$\overline{R}$	74	A27
15	AD12	35	A2	55	$\overline{DEN}$	75	A26
16	AD11	36	V_{CC}	56	DT/$\overline{R}$	76	A25
17	AD10	37	V_{SS}	57	$\overline{BLAST}$	77	V_{CC}
18	AD9	38	A1	58	$\overline{LOCK}$	78	V_{SS}
19	AD8	39	N.C.	59	V_{CC}	79	A24
20	AD7	40	$\overline{BE1}$	60	V_{SS}	80	A23

NOTE:
Do not connect any external logic to any pins marked N.C.

Table 10. 80960SA QFP Pinout—In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
A1	38	A18	5	D0	31	V_{CC}	51
A2	35	A19	4	$\overline{DEN}$	55	V_{CC}	59
A3	34	A20	3	DT/$\overline{R}$	56	V_{CC}	61
AD1	30	A21	2	HLDA	50	V_{CC}	73
AD2	29	A22	1	HOLD	53	V_{CC}	77
AD3	28	A23	80	$\overline{INT0}$	46	V_{CC}	8
AD4	27	A24	79	INT1	47	V_{SS}	13
AD5	26	A25	76	INT2/INTR	48	V_{SS}	22
AD6	25	A26	75	$\overline{INT3}$/$\overline{INTA}$	49	V_{SS}	24
AD7	20	A27	74	$\overline{LOCK}$	58	V_{SS}	32
AD8	19	A28	71	N.C.	39	V_{SS}	37
AD9	18	A29	70	N.C.	63	V_{SS}	43
AD10	17	A30	69	$\overline{READY}$	67	V_{SS}	52
AD11	16	A31	68	$\overline{RESET}$	45	V_{SS}	60
AD12	15	ALE	66	V_{CC}	12	V_{SS}	62
AD13	14	$\overline{AS}$	64	V_{CC}	21	V_{SS}	72
AD14	11	$\overline{BE0}$	41	V_{CC}	23	V_{SS}	78
AD15	10	$\overline{BE1}$	40	V_{CC}	33	V_{SS}	9
A16	7	$\overline{BLAST}$	57	V_{CC}	36	V_{SS}	65
A17	6	CLK2	44	V_{CC}	42	W/$\overline{R}$	54

NOTE:
Do not connect any external logic to any pins marked N.C.

Table 11. 80960SA PLCC Pinout—In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	V_{CC}	22	V_{SS}	43	V_{SS}	64	HOLD
2	N.C.	23	N.C.	44	V_{CC}	65	N.C.
3	A27	24	AD13	45	A3	66	W/$\overline{R}$
4	A26	25	AD12	46	A2	67	$\overline{DEN}$
5	A25	26	AD11	47	V_{CC}	68	DT/$\overline{R}$
6	V_{CC}	27	AD10	48	V_{SS}	69	$\overline{BLAST}$
7	V_{SS}	28	AD9	49	A1	70	$\overline{LOCK}$
8	A24	29	AD8	50	N.C.	71	V_{CC}
9	A23	30	AD7	51	$\overline{BE1}$	72	V_{SS}
10	A22	31	V_{CC}	52	$\overline{BE0}$	73	V_{CC}
11	A21	32	V_{SS}	53	V_{CC}	74	V_{SS}
12	A20	33	V_{CC}	54	V_{SS}	75	N.C.
13	A19	34	V_{SS}	55	CLK2	76	AS
14	A18	35	AD6	56	$\overline{RESET}$	77	V_{SS}
15	A17	36	AD5	57	$\overline{INT0}$	78	ALE
16	A16	37	AD4	58	INT1	79	$\overline{READY}$
17	V_{CC}	38	AD3	59	INT2/INTR	80	A31
18	V_{SS}	39	D2	60	$\overline{INT3}$/$\overline{INTA}$	81	A30
19	AD15	40	D1	61	HLDA	82	A29
20	AD14	41	D0	62	V_{CC}	83	A28
21	V_{CC}	42	N.C.	63	V_{SS}	84	V_{SS}

NOTE:
Do not connect any external logic to any pins marked N.C.

Table 12. 80960SA PLCC Pinout—In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
A1	49	A18	14	DT/$\overline{R}$	68	V_{CC}	44
A2	46	A19	13	HLDA	61	V_{CC}	47
A3	45	A20	12	HOLD	64	V_{CC}	53
D0	41	A21	11	$\overline{INT0}$	57	V_{CC}	6
AD1	40	A22	10	INT1	58	V_{CC}	62
AD2	39	A23	9	INT2/INTR	59	V_{CC}	71
AD3	38	A24	8	$\overline{INT3}$/$\overline{INTA}$	60	V_{CC}	73
AD4	37	A25	5	$\overline{LOCK}$	70	V_{SS}	18
AD5	36	A26	4	N.C.	2	V_{SS}	22
AD6	35	A27	3	N.C.	23	V_{SS}	32
AD7	30	A28	83	N.C.	42	V_{SS}	34
AD8	29	A29	82	N.C.	50	V_{SS}	43
AD9	28	A30	81	N.C.	65	V_{SS}	48
AD10	27	A31	80	N.C.	75	V_{SS}	54
AD11	26	ALE	78	$\overline{READY}$	79	V_{SS}	63
AD12	25	$\overline{AS}$	76	$\overline{RESET}$	56	V_{SS}	7
AD13	24	$\overline{BE0}$	52	V_{CC}	1	V_{SS}	72
AD14	20	$\overline{BE1}$	51	V_{CC}	17	V_{SS}	74
AD15	19	$\overline{BLAST}$	69	V_{CC}	21	V_{SS}	77
AD16	16	CLK2	55	V_{CC}	31	V_{SS}	84
A17	15	$\overline{DEN}$	67	V_{CC}	33	W/$\overline{R}$	66

NOTE:
Do not connect any external logic to any pins marked N.C.

3.4 Package Thermal Specification

The 80960SA is specified for operation when case temperature is within the range 0°C to +85°C (PLCC) or 0°C to +100°C (QFP). Measure case temperature at the top center of the package. Ambient temperature can be calculated from:

$T_J = T_C + P*\theta_{JC}$
$T_A = T_J - P*\theta_{JA}$
$T_C = T_A + P*[\theta_{JA} - \theta_{JC}]$

Compute P by multiplying the maximum voltage by the typical current at maximum temperature. Values for θ_{JA} and θ_{JC} for various airflows are given in Table 13 for the QFP package and in Table 14 for the PLCC package. I_{CC} at maximum temperature is typically 80 percent of specified I_{CC} maximum (cold).

Table 13. 80960SA QFP Package Thermal Characteristics

Thermal Resistance—°C/Watt							
Parameter	Airflow—ft./min (m/sec)						
	0	50	100	200	400	600	800
θ Junction-to-Ambient (Case measured in the middle of the top of the package) (No heatsink)	59	57	54	50	44	40	38
θ Junction-to-Case	11	11	11	11	11	11	11

NOTE:
This table applies to 80960SA QFP soldered directly to board.

Table 14. 80960SA PLCC Package Thermal Characteristics

Thermal Resistance—°C/Watt								
Parameter	Airflow—ft./min (m/sec)							
	0	50	100	200	400	600	800	1000
θ Junction-to-Ambient (No heatsink)	34	32	29.5	28	25	23	21	20.5
θ Junction-to-Case	12	12	12	12	12	12	12	12

NOTE:
This table applies to 80960SA QFP soldered directly to board.

3.5 Stepping Register Information

Upon reset, register g0 contains die stepping information. Table 15 shows the relationship between the number in g0 and the current die stepping.

Table 15. Die Stepping Cross Reference

Register g0	Die Stepping
01010101H	C-1

The current numbering pattern in g0 may not be consistent with past or future steppings of this product.

4.0 WAVEFORMS

Figures 17, 18, 19, 20 and 21 show waveforms for various transactions on the 80960SA's bus. Figure 22 shows a cold reset functional waveform.

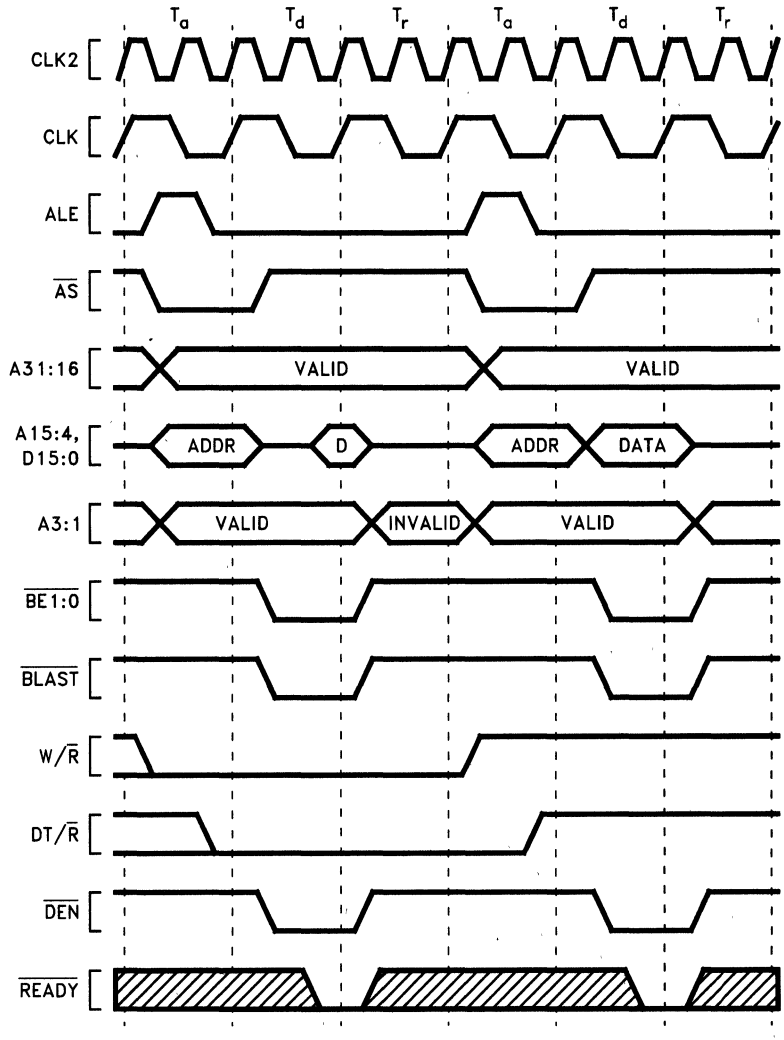


Figure 17. Non-Burst Read and Write Transactions Without Wait States

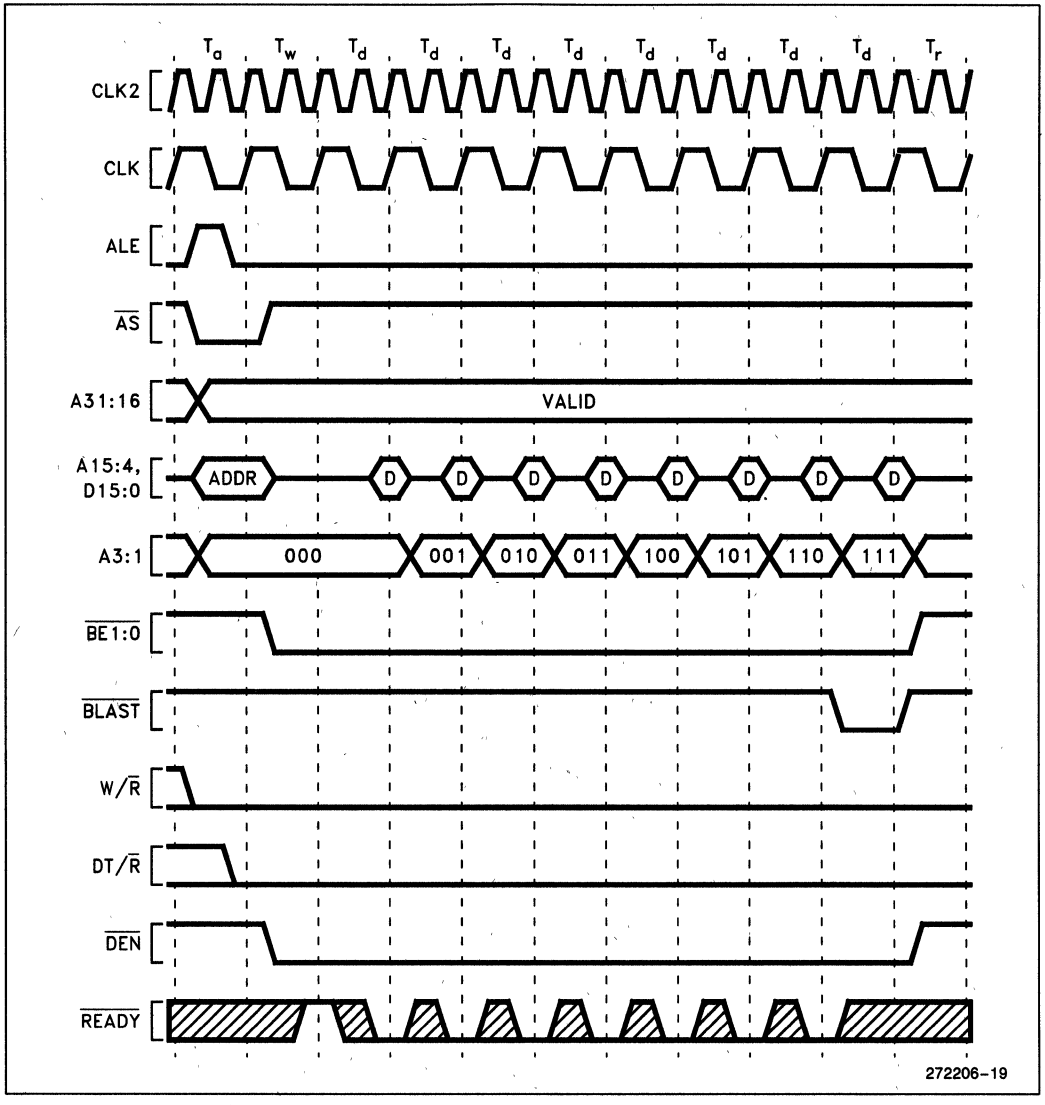


Figure 18. Quad Word Burst Read Transaction with 1, 0, 0, 0, 0, 0, 0, 0 Wait States

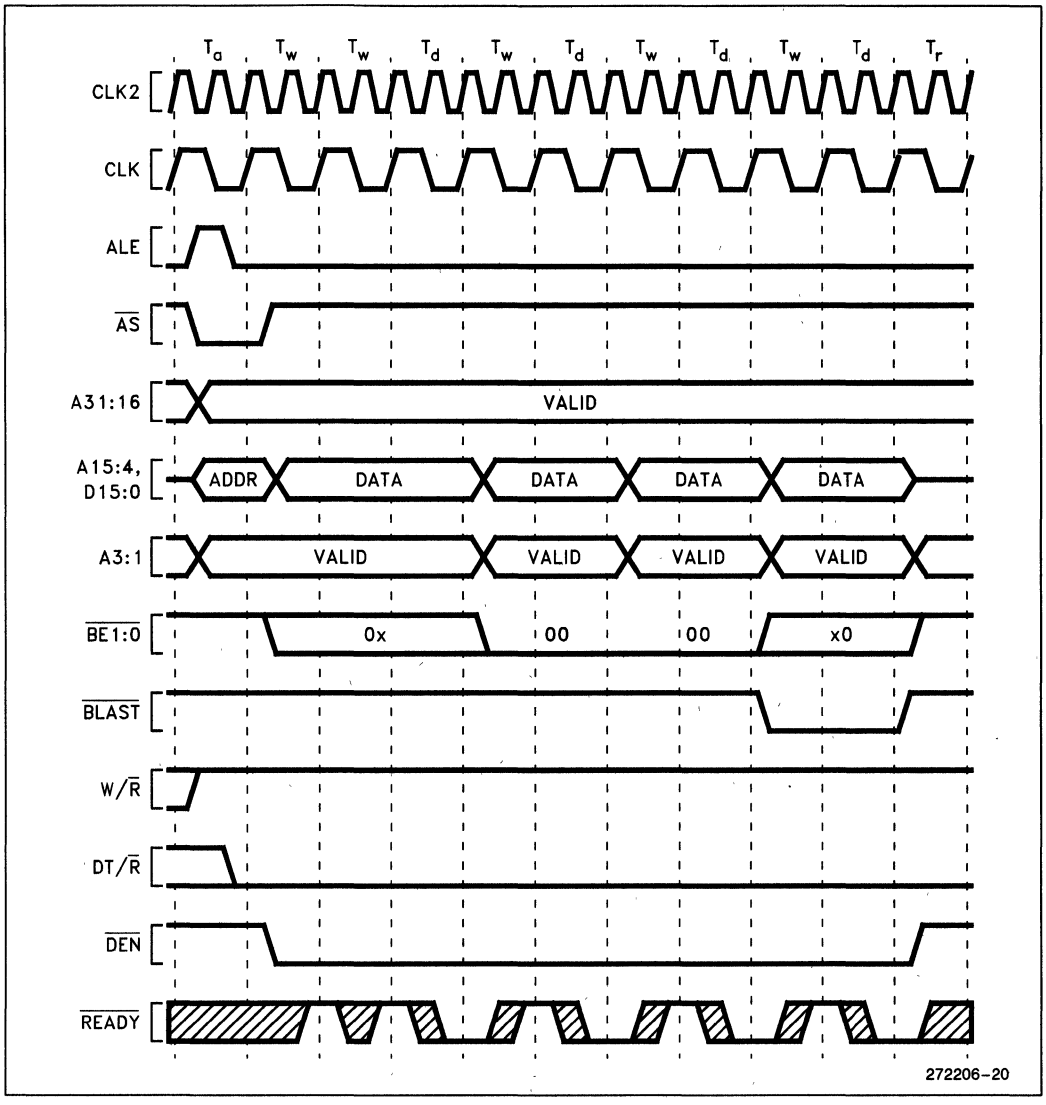


Figure 19. Burst Write Transaction with 2, 1, 1, 1 Wait States (6-8 Bytes Transferred)

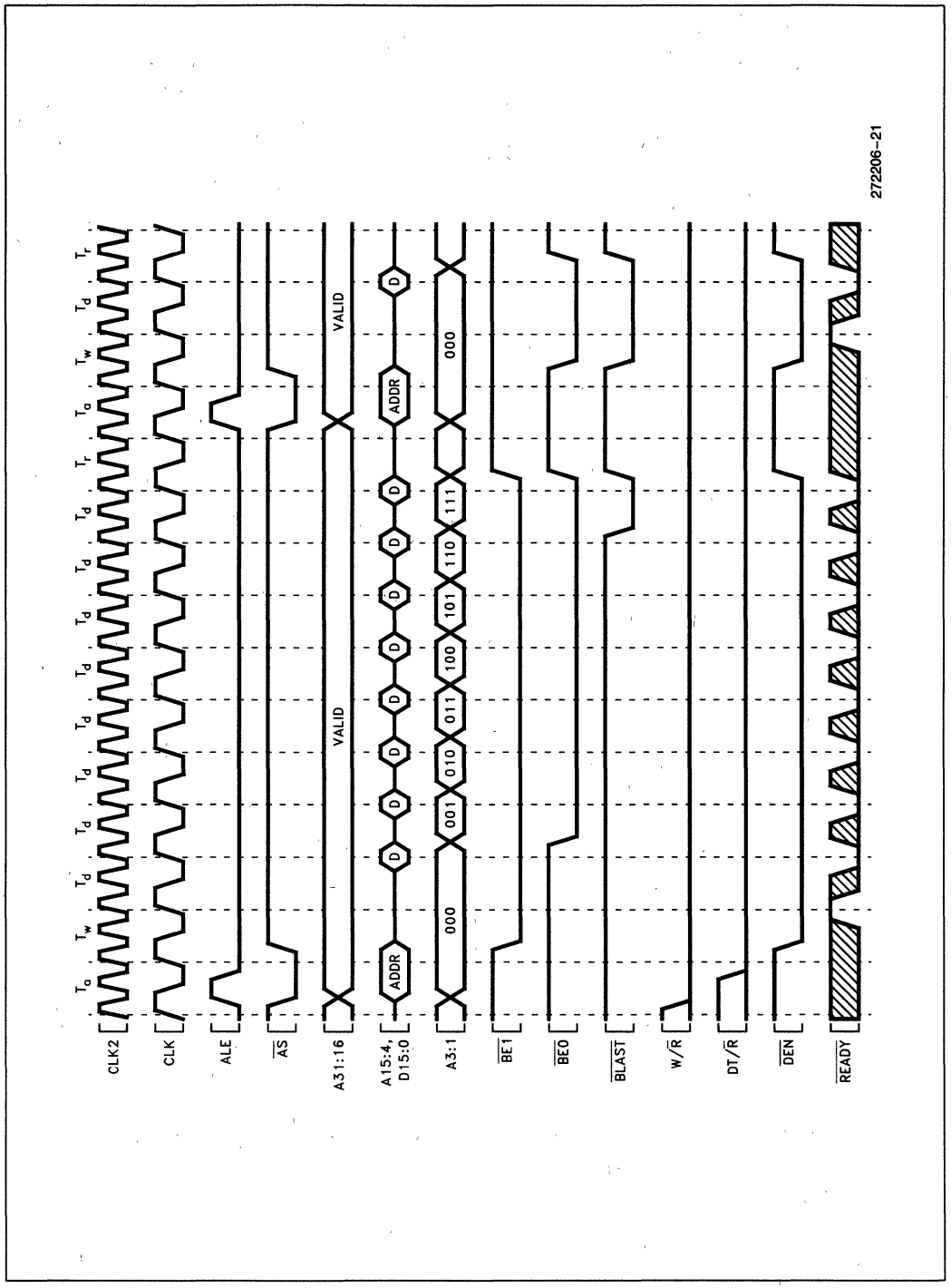


Figure 20. Accesses Generated by Quad Word Read Bus Request, Misaligned One Byte from Quad Word Boundary (1, 0, 0, 0, 0, 0, 0, 0) Wait States

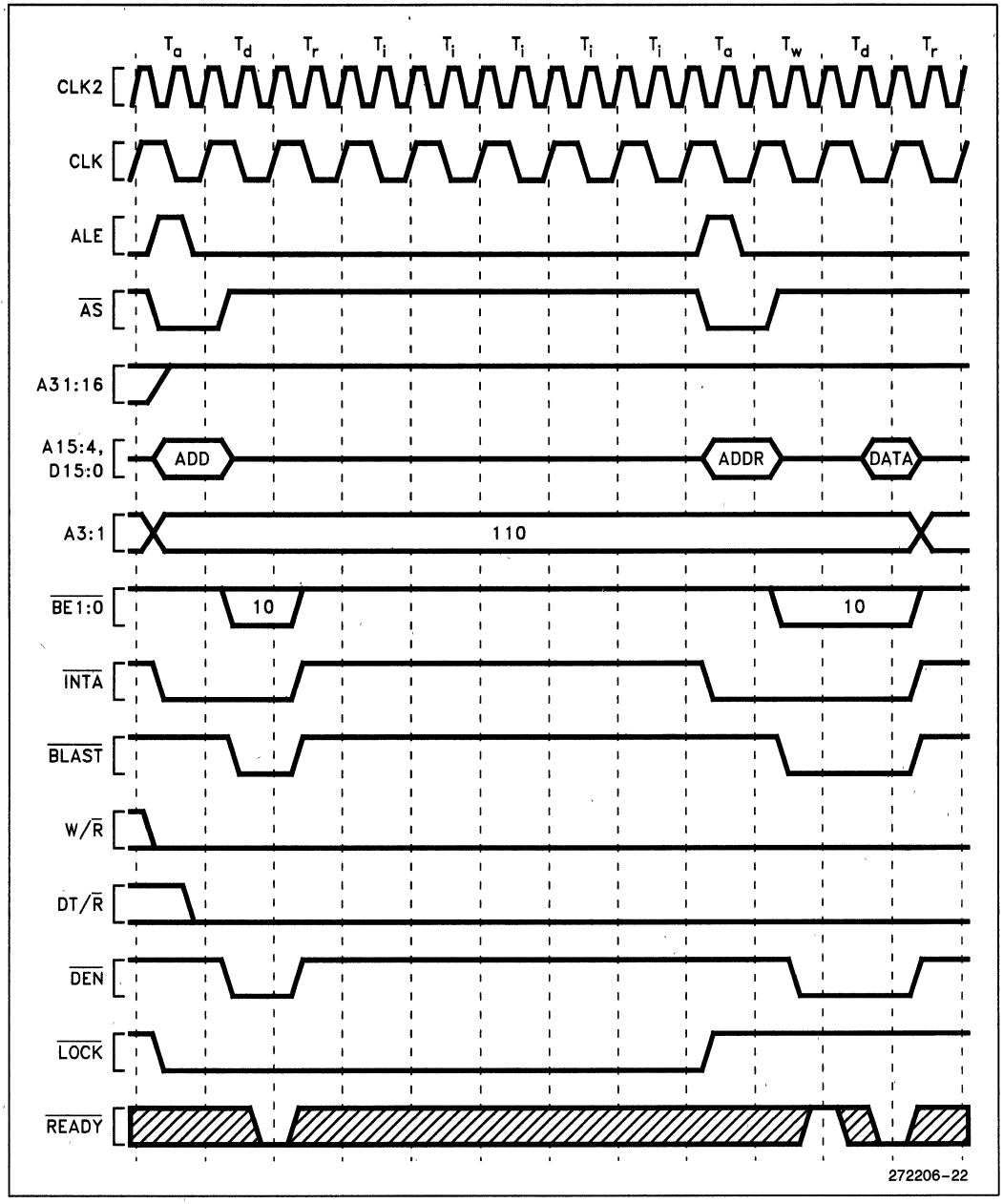


Figure 21. Interrupt Acknowledge Cycle

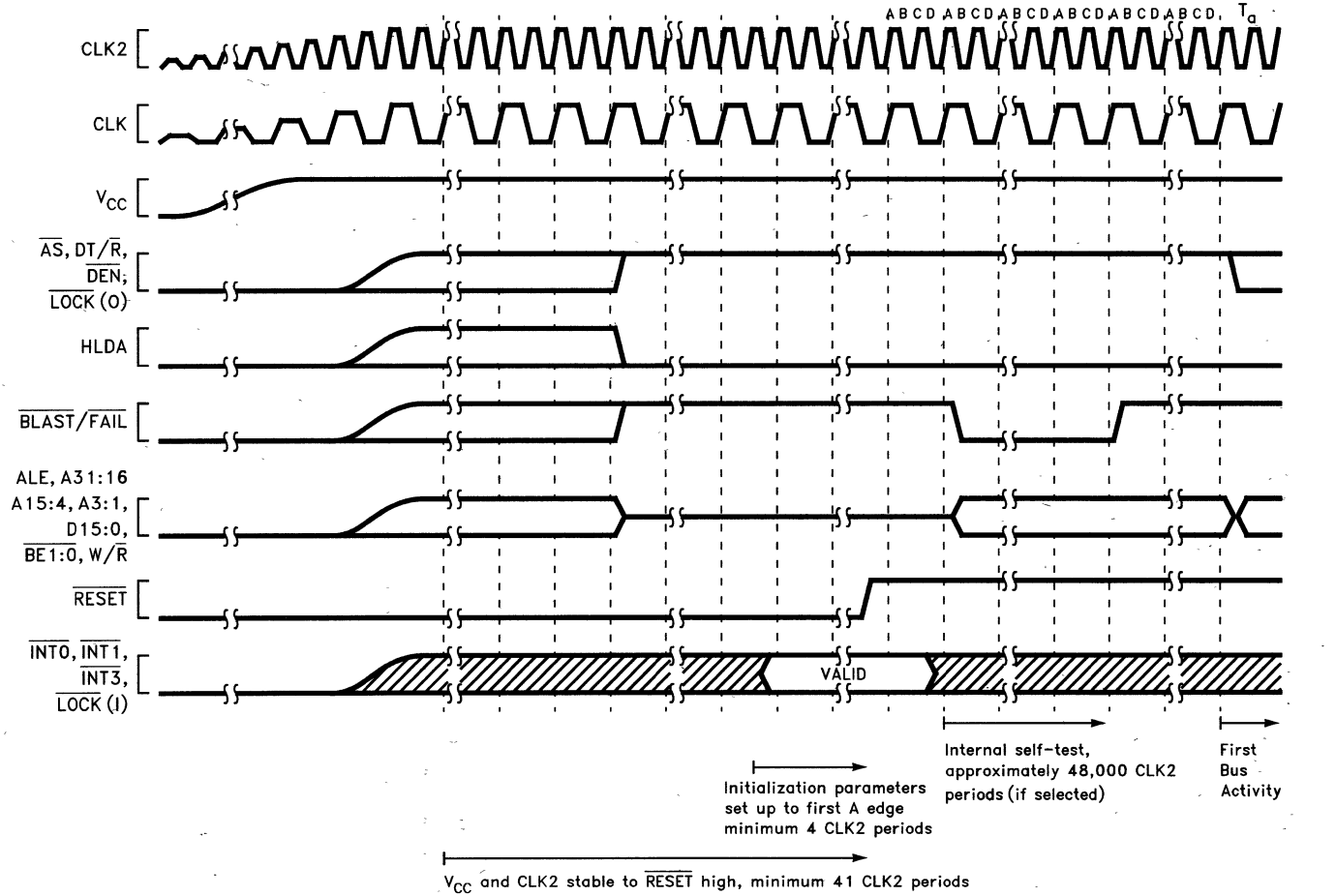


Figure 22. Cold Reset Waveform

5.0 REVISION HISTORY

This data sheet supersedes data sheet 272206-001 and applies only to those devices identified as the current stepping in Section 3.5. The section significantly changed since the previous revision are:

Section	Last Rev.	Description
2.3 Connection Recommendations (pg.11)	−001	Remove two $\overline{LOCK}$ pin Connection Recommendation figures and added Firgure 5 to reflect the new $\overline{LOCK}$ pin connection recommodation of a single 910Ω pullup resistor.
2.5 Test Load Circuit (pg.13)	−001	Obsolete figure (Test Load Circuit for Open-Drain Output Pins) removed to reflect current test conditions.
2.7 DC Characteristics (pg. 14)	−001	I_{OL} value at 0.45V improved. WAS: 2.5 mA IS: 4.0 mA $\overline{LOCK}$ pin I_{OL} value at 0.45V relaxed. WAS: 12 mA IS: 6 mA $\overline{LOCK}$ pin I_{OL} value at 0.60V deleted. 80960SA 16 MHz QFP added to product list.
3.5 Stepping Register Information (pg. 27)	−001	New section added.

This data sheet supersedes data sheet 270917-004, which applied to both the 80960SA and the 80960SB. The 80960SB is now documented in 272207-002. Specification changes in the 80960SA data sheet are a result of design changes. The sections significantly changed since the previous revision are:

Section	Last Rev.	Description
2.3 Connection Recommendations (p. 15)	−004	Deleted corresponding graph of Open Drain Voltage vs. Output Current.
Figure 6. Typical Supply Current vs. Case Temperature (p. 16), Figure 7. Typical Current vs. Frequency (Room Temp) (p. 6) and Figure 8. Typical Current vs. Frequency (Hot Temp) (p. 7)	−004	Regraphed new data in three graphs instead of two.
Table 5. DC Characteristics (p. 19)	−004	Input Leakage Current (I_{L12}) specification added to accurately describe leakage of $\overline{INT0}$ and $\overline{LOCK}$ as inputs. I_{CC} max reduced: Power Supply Current: Was: Is: 10 MHz 280 240 16 MHz 350 300
Table 6. 80960SA AC Characteristics (10 MHz) (p. 21) and Table 7. 80960SA AC Characteristics (16 MHz) (p. 22)	−004	T_7 minimum specification improved: Power Supply Current: Was: Is: 10 MHz 24 ns T_1−11 ns 16 MHz 15 ns T_1−11 ns
Table 8. 80960SA AC Characteristics (20 MHz) (p. 23)	−004	New 20 MHz specification table added for 80960SA C-step.
Table 13. 80960SA QFP Package Thermal Characteristics (p. 32)	−004	θ_{JA} increased to reflect smaller die size and lower I_{CC}.
Table 14. 80960SA PLCC Package Thermal Characteristics (p. 32)	−004	θ_{JA} and θ_{JC} increased to reflect smaller die size and lower I_{CC}.

NOTE:
Page numbers refer to 80960SA data sheet number 272206-001.

The sections significantly changed between revisions -003 and -004 of the 80960SA/SB data sheet were:

Section	Last Rev.	Description
DC Characteristics (p. 19)	−003	Operating temperature for PLCC package changed: Was: T_{CASE} = 0°C to +100°C Is: T_{CASE} = 0°C to +85°C The test program has not changed.
Table 9. 80960SA and 80960SB QFP Pinout—In Pin Order (p. 23)	−003	Signal A12 incorrectly shown as Pin 28; is now correctly shown as Pin 38. Note added to clarify No Connect Pins.

80960SB
EMBEDDED 32-BIT MICROPROCESSOR WITH 16-BIT BURST DATA BUS

- **High-Performance Embedded Architecture**
  - **16 MIPS* Burst Execution at 16 MHz**
  - **5 MIPS Sustained Execution at 16 MHz**
- **512-Byte On-Chip Instruction Cache**
  - **Direct Mapped**
  - **Parallel Load/Decode for Uncached Instructions**
- **Multiple Register Sets**
  - **Sixteen Global 32-Bit Registers**
  - **Sixteen Local 32-Bit Registers**
  - **Four Local Register Sets Stored On-Chip**
  - **Register Scoreboarding**
- **Pin Compatible with 80960SA**
- **Built-In Interrupt Controller**
  - **4 Direct Interrupt Pins**
  - **31 Priority Levels, 256 Vectors**
- **Built-In Floating Point Unit**
  - **Fully IEEE 754 Compatible**
- **Easy to Use, High Bandwidth 16-Bit Bus**
  - **25.6 Mbytes/s Burst**
  - **Up to 16 Bytes Transferred per Burst**
- **32-Bit Address Space, 4 Gigabytes**
- **80-Lead Quad Flat Pack (EIAJ QFP)**
- **84-Lead Plastic Leaded Chip Carrier (PLCC)**
- **Software Compatible with 80960KA/KB/CA/CF Processors**

1

The 80960SB is a member of Intel's i960® 32-bit processor family, which is designed especially for low cost embedded applications. It includes a 512-byte instruction cache, an integrated floating-point unit and a built-in interrupt controller. The 80960SB has a large register set, multiple parallel execution units and a 16-bit burst bus. Using advanced RISC technology, this high performance processor is capable of execution rates in excess of 5 million instructions per second*. The 80960SB is well-suited for a wide range of cost sensitive embedded applications including non-impact printers, network adapters and I/O controllers.

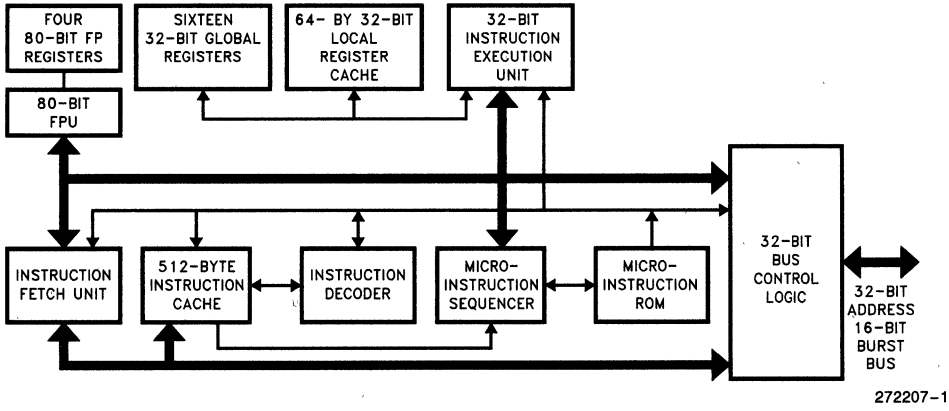


Figure 1. The 80960SB Processor's Highly Parallel Architecture

*Relative to Digital Equipment Corporation's VAX-11/780 at 1 MIPS (VAX-11 is a trademark of Digital Equipment Corporation).

80960SB
Embedded 32-Bit Microprocessor with 16-Bit Burst Data Bus

CONTENTS

CONTENTS

PAGE

FIGURES

CONTENTS

PAGE

FIGURES

CONTENTS

PAGE

TABLES

1.0 THE i960® PROCESSOR

The 80960SB is a member of the 32-bit architecture from Intel known as the i960 processor family. These microprocessors were especially designed to serve the needs of embedded applications. The embedded market includes applications as diverse as industrial automation, avionics, image processing, graphics and networking. These types of applications require high integration, low power consumption, quick interrupt response times and high performance. Since time to market is critical, embedded microprocessors need to be easy to use in both hardware and software designs.

All members of the i960 processor family share a common core architecture which utilizes RISC technology so that, except for special functions, the family members are object-code compatible. Each new processor in the family adds its own special set of functions to the core to satisfy the needs of a specific application or range of applications in the embedded market.

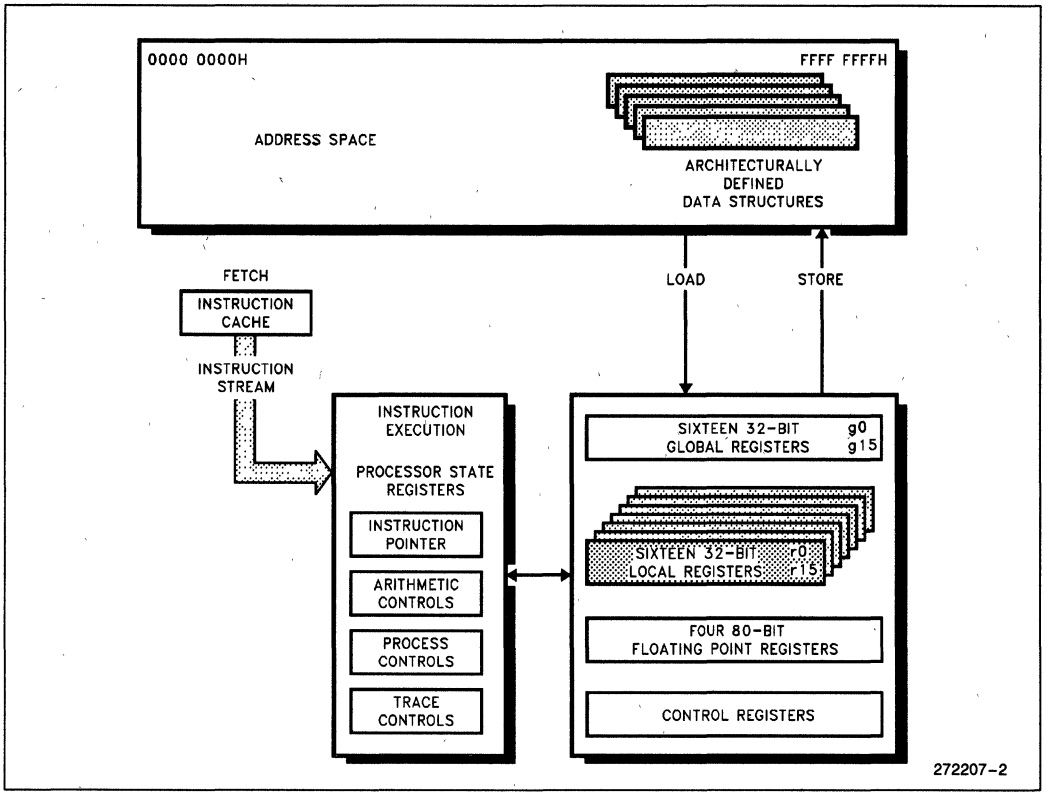


Figure 2. 80960SB Programming Environment

1.1 Key Performance Features

The 80960SB architecture is based on the most recent advances in microprocessor technology and is grounded in Intel's long experience in the design and manufacture of embedded microprocessors. Many features contribute to the 80960SB's exceptional performance:

1. **Large Register Set.** Having a large number of registers reduces the number of times that a processor needs to access memory. Modern compilers can take advantage of this feature to optimize execution speed. For maximum flexibility, the 80960SB provides thirty-two 32-bit registers and four 80-bit floating point registers. (See Figure 2.)
2. **Fast Instruction Execution.** Simple functions make up the bulk of instructions in most programs so that execution speed can be improved by ensuring that these core instructions are executed as quickly as possible. The most frequently executed instructions—such as register-register moves, add/subtract, logical operations and shifts—execute in one to two cycles. (Table 1 contains a list of instructions.)
3. **Load/Store Architecture.** One way to improve execution speed is to reduce the number of times that the processor must access memory to perform an operation. As with other processors based on RISC technology, the 80960SB has a Load/Store architecture. As such, only the LOAD and STORE instructions reference memory; all other instructions operate on registers. This type of architecture simplifies instruction decoding and is used in combination with other techniques to increase parallelism.
4. **Simple Instruction Formats.** All instructions in the 80960SB are 32 bits long and must be aligned on word boundaries. This alignment makes it possible to eliminate the instruction alignment stage in the pipeline. To simplify the instruction decoder, there are only five instruction formats; each instruction uses only one format. (See Figure 3.)
5. **Overlapped Instruction Execution.** Load operations allow execution of subsequent instructions to continue before the data has been returned from memory, so that these instructions can overlap the load. The 80960SB manages this process transparently to software through the use of a register scoreboard. Conditional instructions also make use of a scoreboard so that subsequent unrelated instructions may be executed while the conditional instruction is pending.
6. **Integer Execution Optimization.** When the result of an arithmetic execution is used as an operand in a subsequent calculation, the value is sent immediately to its destination register. At the same time, the value is put on a bypass path to the ALU, thereby saving the time that otherwise would be required to retrieve the value for the next operation.
7. **Bandwidth Optimizations.** The 80960SB gets optimal use of its memory bus bandwidth because the bus is tuned for use with the on-chip instruction cache: instruction cache line size matches the maximum burst size for instruction fetches. The 80960SB automatically fetches four words in a burst and stores them directly in the cache. Due to the size of the cache and the fact that it is continually filled in anticipation of needed instructions in the program flow, the 80960SB is relatively insensitive to memory wait states. The benefit is that the 80960SB delivers outstanding performance even with a low cost memory system.
8. **Cache Bypass.** If a cache miss occurs, the processor fetches the needed instruction then sends it on to the instruction decoder at the same time it updates the cache. Thus, no extra time is spent to load and read the cache.

Table 1. 80960SB Instruction Set

Data Movement	Arithmetic	Logical	Bit and Bit Field
Load Store Move Load Address	Add Subtract Multiply Divide Remainder Modulo Shift Extended Multiply Extended Divide	And Not And And Not Or Exclusive Or Not Or Or Not Nor Exclusive Nor Not Nand Rotate	Set Bit Clear Bit Not Bit Check Bit Alter Bit Scan For Bit Scan Over Bit Extract Modify
Comparison	**Branch**	**Call/Return**	**Fault**
Compare Conditional Compare Compare and Increment Compare and Decrement	Unconditional Branch Conditional Branch Compare and Branch	Call Call Extended Call System Return Branch and Link	Conditional Fault Synchronize Faults
Debug	**Miscellaneous**	**Decimal**	**Floating Point**
Modify Trace Controls Mark Force Mark	Atomic Add Atomic Modify Flush Local Registers Modify Arithmetic Controls Scan Byte for Equal Test Condition Code	Move Add and Carry Subtract with Carry	Move Real Scale Round Square Root Sine Cosine Tangent Arctangent Log Log Binary Log Natural Exponent Classify Copy Real Extended Compare
Synchronous	**Conversion**		
Synchronous Load Synchronous Move	Convert Real to Integer Convert Integer to Real		

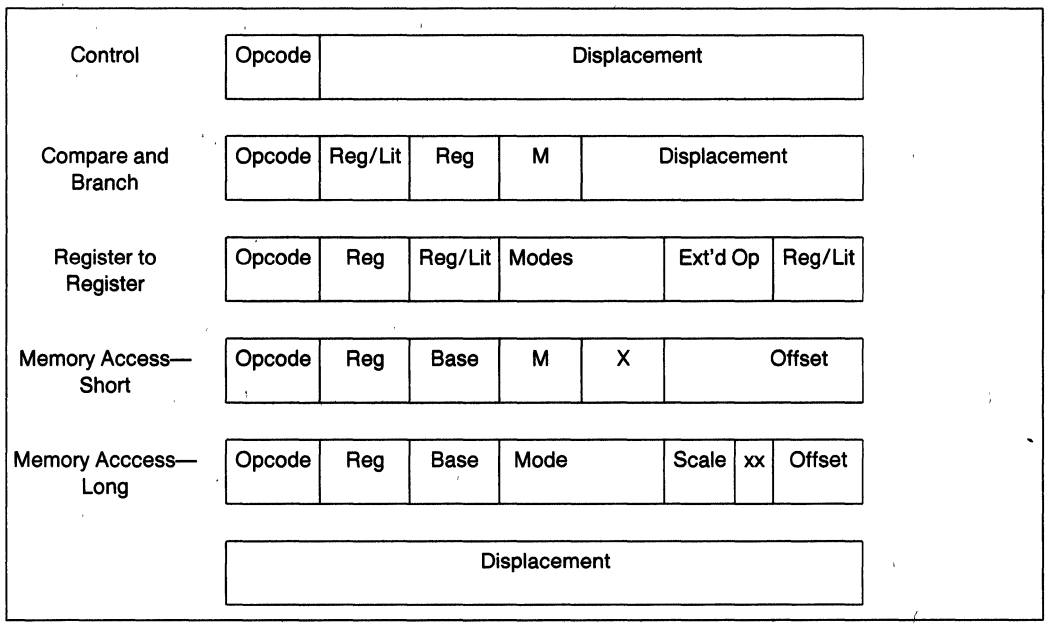


Figure 3. Instruction Formats

1.1.1 MEMORY SPACE AND ADDRESSING MODES

The 80960SB offers a linear programming environment so that all programs running on the processor are contained in a single address space. Maximum address space size is 4 Gigabytes (2^{32} bytes).

For ease of use the 80960SB has a small number of addressing modes, but includes all those necessary to ensure efficient execution of high-level languages such as C. Table 2 lists the memory addressing modes.

Table 2. Memory Addressing Modes

- 12-Bit Offset
- 32-Bit Offset
- Register-Indirect
- Register + 12-Bit Offset
- Register + 32-Bit Offset
- Register + (Index-Register x Scale-Factor)
- Register x Scale Factor + 32-Bit Displacement
- Register + (Index-Register x Scale-Factor) + 32-Bit Displacement

Scale-Factor is 1, 2, 4, 8 or 16

1.1.2 DATA TYPES

The 80960SB recognizes the following data types:

Numeric:

- 8-, 16-, 32- and 64-bit ordinals
- 8-, 16-, 32- and 64-bit integers
- 32-, 64- and 80-bit real numbers

Non-Numeric:

- Bit
- Bit Field
- Triple Word (96 bits)
- Quad-Word (128 bits)

1.1.3 LARGE REGISTER SET

The 80960SB programming environment includes a large number of registers. In fact, 32 registers are available at any time. The availability of this many registers greatly reduces the number of memory accesses required to perform algorithms, which leads to greater instruction processing speed.

There are two types of general-purpose register: local and global. The global registers consist of sixteen 32-bit registers (g0 though g15) and four 80-bit registers (FP0 through FP3). These registers per-

form the same function as the general-purpose registers provided in other popular microprocessors. The term global refers to the fact that these registers retain their contents across procedure calls.

The local registers, on the other hand, are procedure specific. For each procedure call, the 80960SB allocates 16 local registers (r0 through r15). Each local register is 32 bits wide. Any register can also be used for single or double-precision floating-point operations; the 80-bit floating-point registers are provided for extended precision.

1.1.4 MULTIPLE REGISTER SETS

To further increase the efficiency of the register set, multiple sets of local registers are stored on-chip (See Figure 4). This cache holds up to four local register frames, which means that up to three procedure calls can be made without having to access the procedure stack resident in memory.

Although programs may have procedure calls nested many calls deep, a program typically oscillates back and forth between only two to three levels. As a result, with four stack frames in the cache, the probability of having a free frame available on the cache when a call is made is very high. In fact, runs of representative C-language programs show that 80% of the calls are handled without needing to access memory.

If four or more procedures are active and a new procedure is called, the 80960SB moves the oldest local register set in the stack-frame cache to a procedure stack in memory to make room for a new set of registers. Global register g15 is the frame pointer (FP) to the procedure stack.

Global and floating point registers are not exchanged on a procedure call, but retain their contents, making them available to all procedures for fast parameter passing.

1.1.5 INSTRUCTION CACHE

To further reduce memory accesses, the 80960SB includes a 512-byte on-chip instruction cache. The instruction cache is based on the concept of locality of reference; most programs are not usually executed in a steady stream but consist of many branches, loops and procedure calls that lead to jumping back and forth in the same small section of code. Thus, by maintaining a block of instructions in cache, the number of memory references required to read instructions into the processor is greatly reduced.

To load the instruction cache, instructions are fetched in 16-byte blocks; up to four instructions can be fetched at one time. An efficient prefetch algorithm increases the probability that an instruction will already be in the cache when it is needed.

Code for small loops often fits entirely within the cache, leading to a great increase in processing speed since further memory references might not be necessary until the program exits the loop. Similarly, when calling short procedures, the code for the calling procedure is likely to remain in the cache so it will be there on the procedure's return.

1.1.6 REGISTER SCOREBOARDING

The instruction decoder is optimized in several ways. One optimization method is the ability to overlap instructions by using register scoreboarding.

Register scoreboarding occurs when a LOAD moves a variable from memory into a register. When the instruction initiates, a scoreboard bit on the target register is set. Once the register is loaded, the bit is reset. In between, any reference to the register contents is accompanied by a test of the scoreboard bit to ensure that the load has completed before processing continues. Since the processor does not need to wait for the LOAD to complete, it can execute additional instructions placed between the LOAD and the instruction that uses the register contents, as shown in the following example:

```
ld data_2, r4
ld data_2, r5
Unrelated instruction
Unrelated instruction
add r4, r5, r6
```

In essence, the two unrelated instructions between LOAD and ADD are executed "for free" (i.e., take no apparent time to execute) because they are executed while the register is being loaded. Up to three load instructions can be pending at one time with three corresponding scoreboard bits set. By exploiting this feature, system programmers and compiler writers have a useful tool for optimizing execution speed.

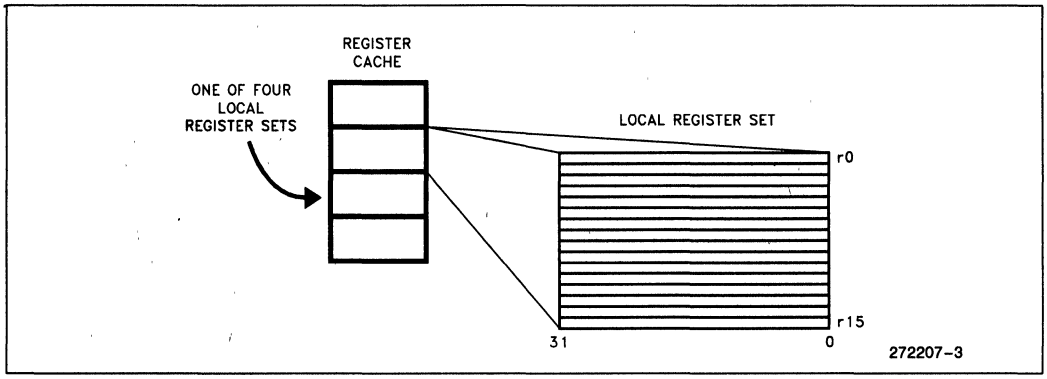


Figure 4. Multiple Register Sets Are Stored On-Chip

1.1.7 FLOATING-POINT ARITHMETIC

In the 80960SB, floating-point arithmetic has been made an integral part of the architecture. Having the floating-point unit integrated on chip provides two advantages. First, it improves the performance of the chip for floating-point applications, since no additional bus overhead is associated with floating-point calculations, thereby leaving more time for other bus operations such as I/O. Second, the cost of using floating-point operations is reduced because a separate coprocessor chip is not required.

The 80960SB floating-point (real-number) data types include single-precision (32-bit), double-precision (64-bit) and extended precision (80-bit) floating-point numbers. Any registers may be used to execute floating-point operations.

The processor provides hardware support for both mandatory and recommended portions of IEEE Standard 754 for floating-point arithmetic, including all arithmetic, exponential, logarithmic and other transcendental functions. Table 3 shows execution times for some representative instructions.

Table 3. Sample Floating-Point Execution Times (μs) at 16 MHz

Function	32-Bit	64-Bit
Add	0.6	0.8
Subtract	0.6	0.8
Multiply	1.1	2.0
Divide	2.0	4.5
Square Root	5.8	6.1
Arctangent	15.8	20.5
Exponent	17.7	19.5
Sine	23.8	25.9
Cosine	23.8	25.9

1.1.8 HIGH BANDWIDTH BUS

The 80960SB CPU resides on a high-bandwidth address/data bus. The bus provides a direct communication path between the processor and the memory and I/O subsystem interfaces. The processor uses the bus to fetch instructions, manipulate memory and respond to interrupts. Bus features include:

- 16-bit data path multiplexed onto the lower bits of the 32-bit address path
- Eight 16-bit half-word burst capability which allows transfers from 1 to 16 bytes at a time
- High bandwidth reads and writes with 25.6 Mbytes/s burst (at 16 MHz)

Table 4 defines bus signal names and functions; Table 5 defines other component-support signals such as interrupt lines.

1.1.9 INTERRUPT HANDLING

The 80960SB can be interrupted in one of two ways: by the activation of one of four interrupt pins or by sending a message on the processor's data bus.

The 80960SB is unusual in that it automatically handles interrupts on a priority basis and can keep track of pending interrupts through its on-chip interrupt controller. Two of the interrupt pins can be configured to provide 8259A-style handshaking for expansion beyond four interrupt lines.

1.1.10 DEBUG FEATURES

The 80960SB has built-in debug capabilities. There are two types of breakpoints and six trace modes. Debug features are controlled by two internal 32-bit registers, the Process-Controls Word and the Trace-Controls Word. By setting bits in these control words, a software debug monitor can closely control how the processor responds during program execution.

The 80960SB provides two hardware breakpoint registers on-chip which, by using a special command, can be set to any value. When the instruction pointer matches either breakpoint register value, the breakpoint handling routine is automatically called.

The 80960SB also provides software breakpoints through the use of two instructions: MARK and FMARK. These can be placed at any point in a program and cause the processor to halt execution at that point and call the breakpoint handling routine. The breakpoint mechanism is easy to use and provides a powerful debugging tool.

Tracing is available for instructions (single step execution), calls and returns and branching. Each trace type may be enabled separately by a special debug instruction. In each case, the 80960SB executes the instruction first and then calls a trace handling routine (usually part of a software debug monitor). Further program execution is halted until the routine completes, at which time execution resumes at the next instruction. The 80960SB's tracing mechanisms, implemented completely in hardware, greatly simplify the task of software test and debug.

1.1.11 FAULT DETECTION

The 80960SB has an automatic mechanism to handle faults. Fault types include floating point, trace and arithmetic faults. When the processor detects a fault, it automatically calls the appropriate fault handling routine and saves the current instruction pointer and necessary state information to make efficient recovery possible. Like interrupt handling routines, fault handling routines are usually written to meet the needs of specific applications and are often included as part of the operating system or kernel.

For each of the fault types, there are numerous subtypes that provide specific information about a fault. For example, a floating point fault may have the subtype set to an Overflow or Zero-Divide fault. The fault handler can use this specific information to respond correctly to the fault.

1.1.12 BUILT-IN TESTABILITY

Upon reset, the 80960SB automatically conducts an exhaustive internal test of its major blocks of logic. Then, before executing its first instruction, it does a zero check sum on the first eight words in memory to ensure that the memory image was programmed correctly. If a problem is discovered at any point during the self-test, the 80960SB asserts its $\overline{\text{FAIL}}$ pin and will not begin program execution. Self test takes approximately 47,000 cycles to complete.

System manufacturers can use the 80960SB's self-test feature during incoming parts inspection. No special diagnostic programs need to be written. The test is both thorough and fast. The self-test capability helps ensure that defective parts are discovered before systems are shipped and, once in the field, the self-test makes it easier to distinguish between problems caused by processor failure and problems resulting from other causes.

1.1.13 CHMOS

The 80960SB is fabricated using Intel's CHMOS IV (Complementary High Speed Metal Oxide Semiconductor) process. The 80960SB is available at 10 MHz in the QFP package and at 10 and 16 MHz in the PLCC package.

Table 4. 80960SB Pin Description: Bus Signals

Name	Type	Description
CLK2	I	**SYSTEM CLOCK** provides the fundamental timing for 80960SB systems. It is divided by two inside the 80960SB to generate the internal processor clock.
A31:16	O T.S.	**ADDRESS BUS** carries the upper 16 bits of the 32-bit physical address to memory. It is valid throughout the burst cycle; no latch is required.
AD15:1, D0	I/O T.S.	**ADDRESS/DATA BUS** carries the low order 32-bit addresses and 16-bit data to and from memory. AD15:4 must be latched since the cycle following the address cycle carries data on the bus.
A3:1	O T.S.	**ADDRESS BUS** carries the word addresses of the 32-bit address to memory. These three bits are incremented during a burst access indicating the next word address of the burst access. Note that A3:1 are duplicated with AD3:1 during the address cycle.
ALE	O T.S.	**ADDRESS LATCH ENABLE** indicates the transfer of a physical address. ALE is asserted during a T_a cycle and deasserted before the beginning of the T_d state. It is active HIGH and floats to a high impedance state during a hold cycle (T_h).
$\overline{AS}$	O T.S.	**ADDRESS STATUS** indicates an address state. $\overline{AS}$ is asserted every T_a state and deasserted during the following T_d state. $\overline{AS}$ is driven HIGH during reset.
W/$\overline{R}$	O T.S.	**WRITE/READ** specifies, during a T_a cycle, whether the operation is a write or read. It is latched on-chip and remains valid during T_d cycles.
$\overline{DEN}$	O T.S.	**DATA ENABLE** is asserted during T_d cycles and indicates transfer of data on the AD lines. The AD lines should not be driven by an external source unless $\overline{DEN}$ is asserted. When $\overline{DEN}$ is asserted, outputs from the previous cycle are guaranteed to be three-stated. In addition, $\overline{DEN}$ deasserted indicates inputs have been captured and therefore input hold times can be disregarded. $\overline{DEN}$ is driven high during reset.
DT/$\overline{R}$	O T.S.	**DATA TRANSMIT/RECEIVE** indicates the direction of data transfer to and from the bus. It is low during T_a and T_d cycles for a read or interrupt acknowledgment; it is high during T_a and T_d cycles for a write. DT/$\overline{R}$ never changes state when $\overline{DEN}$ is asserted. DT/$\overline{R}$ is driven high during reset.
$\overline{READY}$	I	**READY** indicates that data on AD lines can be sampled or removed. If $\overline{READY}$ is not asserted during a T_d cycle, the T_d cycle is extended to the next cycle by inserting a wait state (T_w).
$\overline{LOCK}$	I/O O.D.	**BUS LOCK** prevents bus masters from gaining control of the bus during Read/Modify/Write (RMW) cycles. The processor or any bus agent may assert $\overline{LOCK}$. At the start of a RMW operation, the processor examines the $\overline{LOCK}$ pin. If the pin is already asserted, the processor waits until it is not asserted. If the pin is not asserted, the processor asserts $\overline{LOCK}$ during the T_a cycle of the read transaction. The processor deasserts $\overline{LOCK}$ in the T_a cycle of the write transaction. During the time $\overline{LOCK}$ is asserted, a bus agent can perform a normal read or write but not a RMW operation. The processor also asserts $\overline{LOCK}$ during interrupt-acknowledge transactions. Do not leave $\overline{LOCK}$ unconnected. It must be pulled high for the processor to function properly. **ONCE MODE:** The $\overline{LOCK}$ pin is sampled during reset. If it is asserted LOW at the end of reset, all outputs will be three-stated until the part is reset again. ONCE mode is used in conjunction with an in-circuit emulator.

I/O = Input/Output, O = Output, I = Input, O.D. = Open Drain, T.S. = Three-state

Table 4. 80960SB Pin Description: Bus Signals (Continued)

Name	Type	Description
$\overline{BE1:0}$	O T.S.	**BYTE ENABLE LINES** specify which data bytes (up to two) on the bus take part in the current bus cycle. $\overline{BE1}$ corresponds to AD15:8; $\overline{BE0}$ corresponds to AD7:1, D0. The byte enable lines are asserted appropriately during each data cycle. **INITIALIZATION FAILURE** indicates that the processor has failed to initialize correctly. The failure state is indicated by a combination of $\overline{BLAST}$ asserted and $\overline{BE1:0}$ not asserted. This condition occurs after $\overline{RESET}$ is deasserted and before the first bus transaction begins. $\overline{FAIL}$ is asserted while the processor performs a self-test. If the self-test completes successfully, $\overline{FAIL}$ is deasserted. The processor then performs a zero checksum on the first eight words of memory. If it fails, $\overline{FAIL}$ is asserted for a second time and remains asserted; if it passes, system initialization continues and $\overline{FAIL}$ remains deasserted.
HOLD	I	**HOLD:** A request from an external bus master to acquire the bus. When the processor receives HOLD and grants bus control to another master, it floats its three-state bus lines, then asserts HLDA and enters the T_h state. When HOLD is deasserted, the processor deasserts HLDA and enters the T_i or T_a state.
HLDA	O T.S.	**HOLD ACKNOWLEDGE:** Notifies an external bus master that the processor has relinquished control of the bus. This signal is always driven. At reset it is driven LOW.
$\overline{BLAST}/\overline{FAIL}$	O T.S.	**BURST LAST** indicates the last data cycle (T_d) of a burst access. It is asserted low during the last T_d and associated with T_w cycles in a burst access. **INITIALIZATION FAILURE** indicates that the processor has failed to initialize correctly. The failure state is indicated by a combination of $\overline{BLAST}$ asserted and $\overline{BE1:0}$ not asserted. This condition occurs after $\overline{RESET}$ is deasserted and before the first bus transaction begins. $\overline{FAIL}$ is asserted while the processor performs a self-test. If the self-test completes successfully, $\overline{FAIL}$ is deasserted. The processor then performs a zero checksum on the first eight words of memory. If it fails, $\overline{FAIL}$ is asserted for a second time and remains asserted; if it passes, system initialization continues and $\overline{FAIL}$ remains deasserted.

I/O = Input/Output, O = Output, I = Input, O.D. = Open Drain, T.S. = Three-state

Table 5. 80960SB Pin Description: Support Signals

Name	Type	Description
$\overline{\text{RESET}}$	I	**RESET:** Clears the processor's internal logic and causes it to reinitialize. During $\overline{\text{RESET}}$ assertion, the input pins are ignored (except for $\overline{\text{INT0}}$, INT1, $\overline{\text{INT3}}$, $\overline{\text{LOCK}}$), the three-state output pins are placed in a HIGH impedance state (except for DT/$\overline{\text{R}}$, $\overline{\text{DEN}}$, and $\overline{\text{AS}}$) and other output pins are placed in their non-asserted states. $\overline{\text{RESET}}$ must be asserted for at least 41 CLK2 cycles for a predictable reset. Optionally, for a synchronous reset, the LOW and HIGH transition of $\overline{\text{RESET}}$ should occur after the rising edge of both CLK2 and the external bus CLK and before the next rising edge of CLK2. The interrupt pins indicate the initialization sequence executed. Typical initialization requires driving only $\overline{\text{INT0}}$ and $\overline{\text{INT3}}$ to a HIGH state. The reset conditions follow:

$\overline{\text{INT0}}$	INT1	$\overline{\text{INT3}}$	$\overline{\text{LOCK}}$	Action Taken
1	x	1	1	Run-self-test (core initialization)
0	0	1	1	Disable self-test
0	1	x	x	Reserved
x	x	0	x	Reserved
x	x	x	0	ONCE mode (see $\overline{\text{LOCK}}$ pin)

$\overline{\text{INT0}}$	I	**INTERRUPT 0:** Indicates a pending interrupt. To signal an interrupt in a synchronous system, this pin—as well as the other interrupt pins—must be enabled by being deasserted for at least one bus cycle and then asserted for at least one additional bus cycle. In an asynchronous system the pin must remain deasserted for at least two system clock cycles and then asserted for at least two more system clock cycles. The interrupt control register must be programmed with an interrupt vector before using this pin. $\overline{\text{INT0}}$ is sampled during $\overline{\text{RESET}}$ to determine if the self-test sequence is to be executed.
INT1	I	**INTERRUPT 1:** Like $\overline{\text{INT0}}$, provides direct interrupt signaling. INT1 is sampled during reset to determine if the self-test sequence is to be executed.
INT2/INTR	I	**INTERRUPT2/INTERRUPT REQUEST:** The interrupt control register determines how this pin is interpreted. If INT2, it has the same interpretation as the $\overline{\text{INT0}}$ and INT1 pins. If INTR, it is used to receive an interrupt request from an external interrupt controller.
$\overline{\text{INT3}}$/$\overline{\text{INTA}}$	I/O T.S.	**INTERRUPT3/INTERRUPT ACKNOWLEDGE:** The interrupt control register determines how this pin is interpreted. If $\overline{\text{INT3}}$, it has the same interpretation as the $\overline{\text{INT0}}$ and INT1 pins. If $\overline{\text{INTA}}$, it is used as an output to control interrupt-acknowledge transactions. The $\overline{\text{INTA}}$ output is latched on-chip and remains valid during T_d cycles; as an output, it is open drain. $\overline{\text{INT3}}$ must be pulled HIGH during reset.
NC	N/A	**NOT CONNECTED:** Indicates pins should not be connected. Never connect any pin marked NC; these pins may be reserved for factory use.

I/O = Input/Output, O = Output, I = Input, O.D. = Open Drain, T.S. = Three-state

2.0 ELECTRICAL SPECIFICATIONS

2.1 Power and Grounding

The 80960SB is implemented in CHMOS IV technology and therefore has modest power requirements. Its high clock frequency and numerous output buffers (address/data, control, error and arbitration signals) can cause power surges as multiple output buffers simultaneously drive new signal levels. For clean on-chip power distribution, V_{CC} and V_{SS} pins separately feed the device's functional units. Power and ground connections must be made to all 80960SB power and ground pins. On the circuit board, all V_{CC} pins must be strapped closely together, preferably on a power plane; all V_{SS} pins should be strapped together, preferably on a ground plane.

2.2 Power Decoupling Recommendations

Place a liberal amount of decoupling capacitance near the 80960SB. When driving the bus the processor can cause transient power surges, particularly when connected to a large capacitive load.

Low inductance capacitors and interconnects are recommended for best high frequency electrical performance. Inductance is reduced by shortening board traces between the processor and decoupling capacitors as much as possible.

2.3 Connection Recommendations

For reliable operation, always connect unused inputs to an appropriate signal level. In particular, if one or more interrupt lines are not used, they should be pulled up. No inputs should ever be left floating.

The $\overline{LOCK}$ open drain pin requires a pullup resistor whether or not the pin is used as an output. Figure 5 shows the recommended resistor value.

Do not connect external logic to pins marked NC.

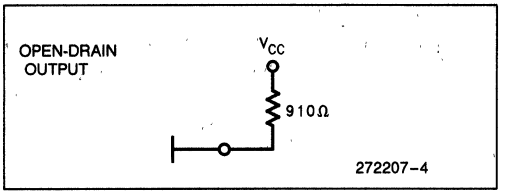


Figure 5. Connection Recommendation for $\overline{LOCK}$

2.4 Characteristic Curves

Figure 6 shows typical supply current requirements over the operating temperature range of the processor at supply voltage (V_{CC}) of 5V. Figure 7 shows the typical power supply current (I_{CC}) that the 80960SB requires at various operating frequencies when measured at three input voltage (V_{CC}) levels.

For a given output current (I_{OL}) the curve in Figure 8 shows the worst case output low voltage (V_{OL}). Figure 9 shows the typical capacitive derating curve for the 80960SB measured from 1.5V on the system clock (CLK) to 0.8V on the falling edge and 2.0V on the rising edge of the bus address/data (AD) signals.

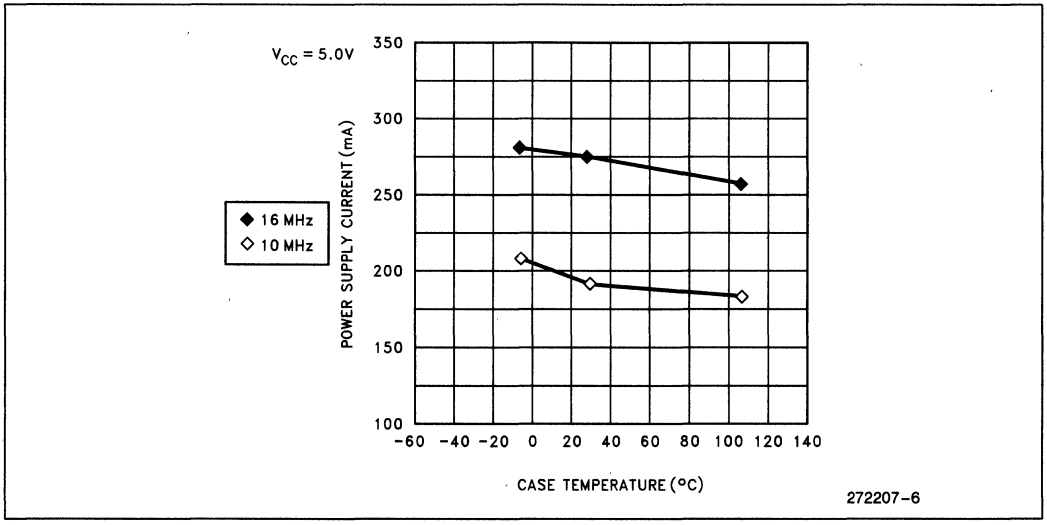


Figure 6. Typical Supply Current vs Case Temperature

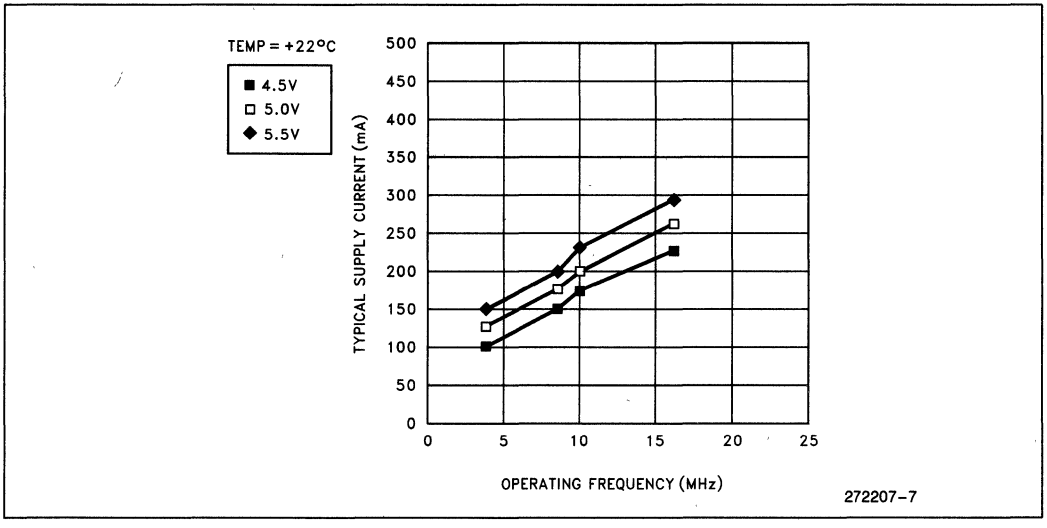


Figure 7. Typical Current vs Frequency (Room Temp)

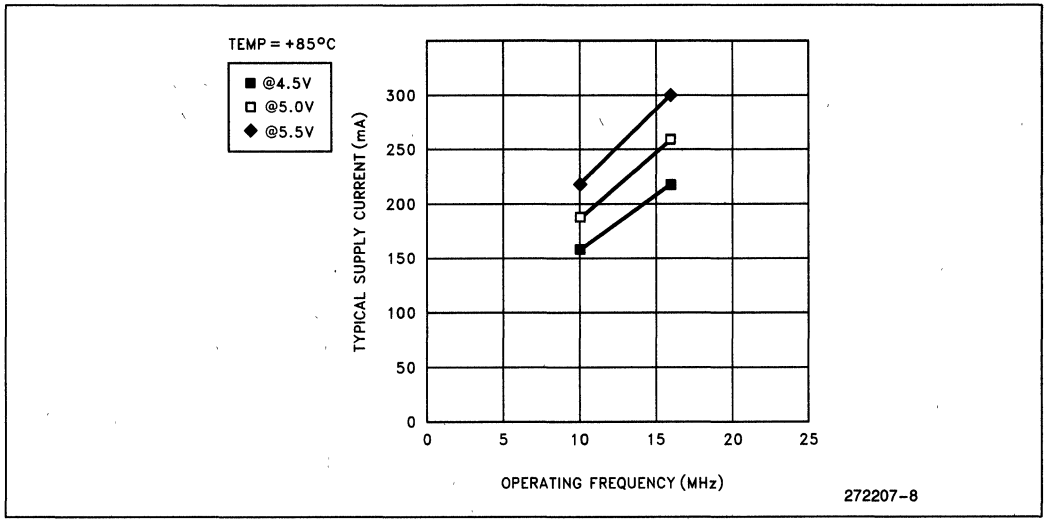


Figure 8. Typical Current vs Frequency (Hot Temp)

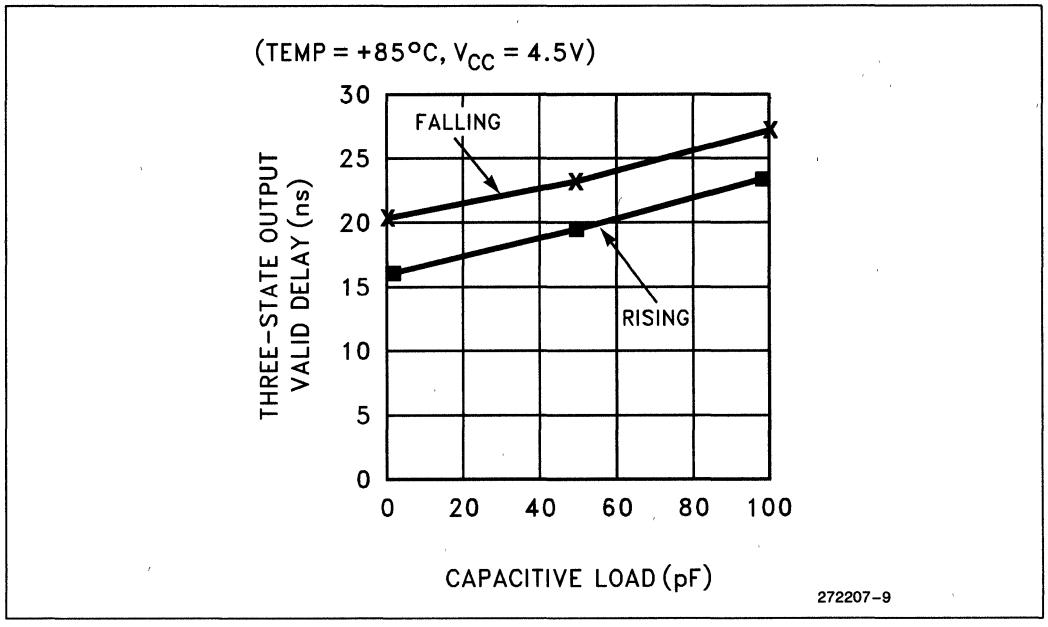


Figure 9. Capacitive Derating Curve

2.5 Test Load Circuit

Figure 10 illustrates the load circuit used to test the 80960SB's output pins.

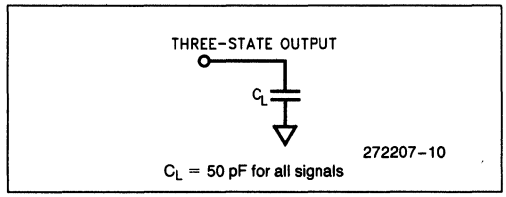


Figure 10. Test Load Circuit for Three-State Output Pins

2.6 ABSOLUTE MAXIMUM RATINGS*

Operating Temperature (PLCC) 0°C to +85°C Case
Operating Temperature (QFP) .0°C to +100°C Case
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin (PLCC) ...−0.5V to V_{CC} +0.5V
Voltage on Any Pin (QFP) ..−0.25V to V_{CC} +0.25V
Power Dissipation .....1.9W (16 MHz)

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

2.7 DC Characteristics

80960SB (10 MHz QFP) T_{CASE} = 0°C to +100°C, V_{CC} = 5V ±5%
80960SB (10 and 16 MHz PLCC) T_{CASE} = 0°C to +85°C, V_{CC} = 5V ±10%

Table 6. DC Characteristics

Symbol	Parameter	Min	Max	Units	Notes
V_{IL}	Input Low Voltage	−0.3	+0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} +0.3	V	
V_{CL}	CLK2 Input Low Voltage	−0.3	+0.8	V	
V_{CH}	CLK2 Input High Voltage	0.7 V_{CC}	V_{CC} +0.3	V	
V_{OL}	Output Low Voltage		0.45 0.45	V V	I_{OL} = 4.0 mA I_{OL} = 6 mA, $\overline{LOCK}$ Pin
V_{OH}	Output High Voltage	2.4		V	All TS, −2.5 mA (1)
I_{CC}	Power Supply Current: 10 MHz-QFP 10 MHz-PLCC 16 MHz-PLCC		 280 280 350	 mA mA mA	 T_{CASE} = 0°C T_{CASE} = 0°C T_{CASE} = 0°C
I_{LI1}	Input Leakage Current, Except $\overline{INT0}$, $\overline{LOCK}$		±15	μA	0 ≤ V_{IN} ≤ V_{CC}
I_{LI2}	Input Leakage Current, $\overline{INT0}$, $\overline{LOCK}$		−300	μA	V_{IN} = 0.45V(2)
I_{LO}	Output Leakage Current		±15	μA	
C_{IN}	Input Capacitance		10	pF	f_C = 1 MHz(3)
C_O	Output Capacitance		12	pF	f_C = 1 MHz(3)
C_{CLK}	Clock Capacitance		10	pF	f_C = 1 MHz(3)

NOTES:
1. Not measured for open-drain output.
2. $\overline{INT0}$ and $\overline{LOCK}$ have internal pullup devices.
3. Input, output and clock capacitance are not tested.

2.8 AC Specifications

This section describes the AC specifications for the 80960SB pins. All input and output timings are specified relative to the 1.5V level of the rising edge of CLK2 and refer to the time at which the signal crosses 1.5V (for output delay and input setup). All AC testing should be done with input voltages of 0.4V and 2.4V, except for the clock (CLK2) which should be tested with input voltages of 0.45V and 0.7 x V_{CC}. See Figure 11 and Tables 7 and 8 for timing relationships for the 80960SB signals.

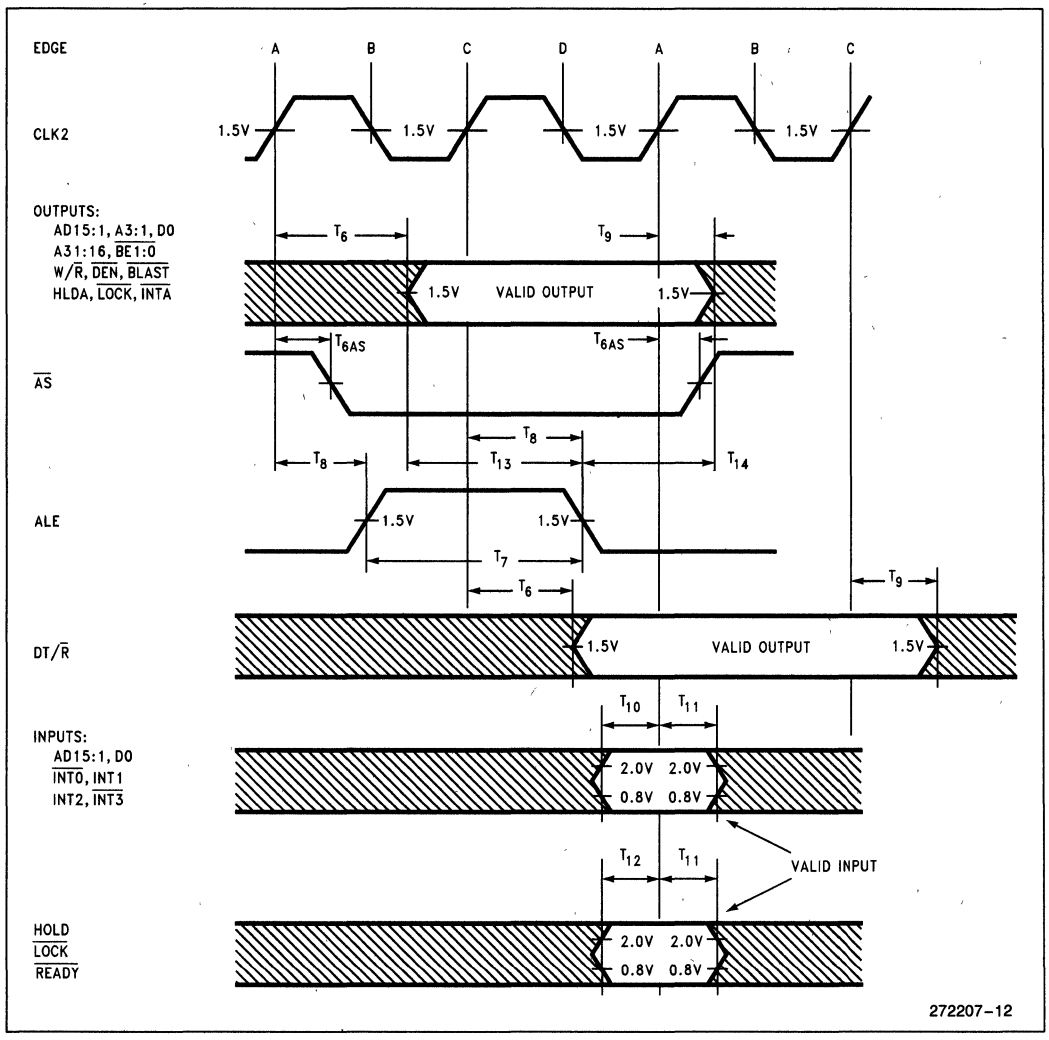


Figure 11. Drive Levels and Timing Relationships for 80960SB Signals

2.8.1 AC SPECIFICATION TABLES

Table 7. 80960SB AC Characteristics (10 MHz)

Symbol	Parameter	Min	Max	Units	Notes
Input Clock					
T_1	Processor Clock Period (CLK2)	50	125	ns	V_{IN} = 1.5V
T_2	Processor Clock Low Time (CLK2)	8		ns	V_T = 10% Point = V_{CL} + (V_{CH} − V_{CL}) x 0.1
T_3	Processor Clock High Time (CLK2)	8		ns	V_T = 90% Point = V_{CL} + (V_{CH} − V_{CL}) x 0.9
T_4	Processor Clock Fall Time (CLK2)		10	ns	V_T = 90% to 10% Point (1)
T_5	Processor Clock Rise Time (CLK2)		10	ns	V_T = 10% to 90% Point (1)
Synchronous Outputs					
T_6	Output Valid Delay	2	31	ns	
T_{6AS}	$\overline{AS}$ Output Valid Delay	2	25	ns	
T_7	ALE Width	T_1−11		ns	
T_8	ALE Output Valid Delay	4	33	ns	
T_9	Output Float Delay	2	20	ns	(2)
Synchronous Inputs					
T_{10}	Input Setup 1	10		ns	
T_{11}	Input Hold	2		ns	
T_{12}	Input Setup 2	13		ns	
T_{13}	Setup to ALE Inactive	10		ns	
T_{14}	Hold After ALE Inactive	8		ns	
T_{15}	$\overline{RESET}$ Hold	3		ns	(3)
T_{16}	$\overline{RESET}$ Setup	5		ns	(3)
T_{17}	$\overline{RESET}$ Width	2050		ns	41 CLK2 Periods Minimum

NOTES:
1. Processor clock (CLK2) rise time and fall time are not tested.
2. A float condition occurs when the maximum output current becomes less than I_{LO}. Float delay is not tested, but should be no longer than the valid delay.
3. Meeting $\overline{RESET}$ setup and hold times is an optional method of synchronizing your clocks. If you decide to use an asynchronous reset, synchronizing the clock can be accomplished by using $\overline{AS}$.

Table 8. 80960SB AC Characteristics (16 MHz)

Symbol	Parameter	Min	Max	Units	Notes
Input Clock					
T_1	Processor Clock Period (CLK2)	31.25	125	ns	V_{IN} = 1.5V
T_2	Processor Clock Low Time (CLK2)	8		ns	V_T = 10% Point = V_{CL} + (V_{CH} − V_{CL}) x 0.1
T_3	Processor Clock High Time (CLK2)	8		ns	V_T = 90% Point = V_{CL} + (V_{CH} − V_{CL}) x 0.9
T_4	Processor Clock Fall Time (CLK2)		10	ns	V_T = 90% to 10% Point (1)
T_5	Processor Clock Rise Time (CLK2)		10	ns	V_T = 10% to 90% Point (1)
Synchronous Outputs					
T_6	Output Valid Delay	2	25	ns	
T_{6AS}	$\overline{AS}$ Output Valid Delay	2	21	ns	
T_7	ALE Width	T_1−11		ns	
T_8	ALE Output Valid Delay	2	22	ns	
T_9	Output Float Delay	2	20	ns	(2)
Synchronous Inputs					
T_{10}	Input Setup 1	10		ns	
T_{11}	Input Hold	2		ns	
T_{12}	Input Setup 2	13		ns	
T_{13}	Setup to ALE Inactive	10		ns	
T_{14}	Hold After ALE Inactive	8		ns	
T_{15}	$\overline{RESET}$ Hold	3		ns	(3)
T_{16}	$\overline{RESET}$ Setup	5		ns	(3)
T_{17}	$\overline{RESET}$ Width	1281		ns	41 CLK2 Periods Minimum

NOTES:
1. Processor clock (CLK2) rise time and fall time are not tested.
2. A float condition occurs when the maximum output current becomes less than I_{LO}. Float delay is not tested, but should be no longer than the valid delay.
3. Meeting $\overline{RESET}$ setup and hold times is an optional method of synchronizing your clocks. If you decide to use an asynchronous reset, synchronizing the clock can be accomplished by using $\overline{AS}$.

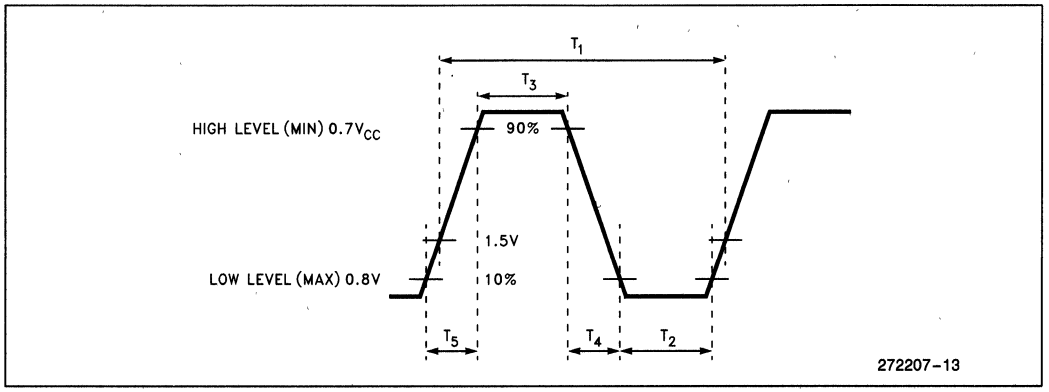


Figure 12. Processor Clock Pulse (CLK2)

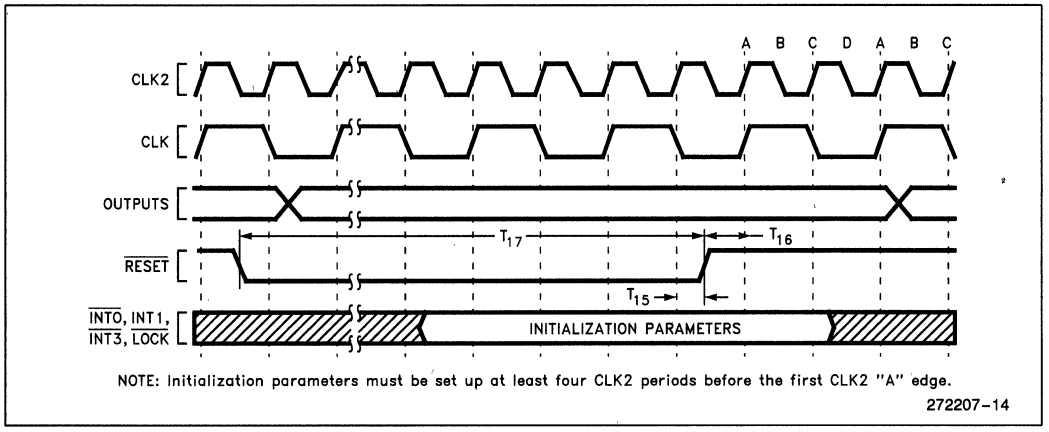


Figure 13. $\overline{\text{RESET}}$ Signal Timing

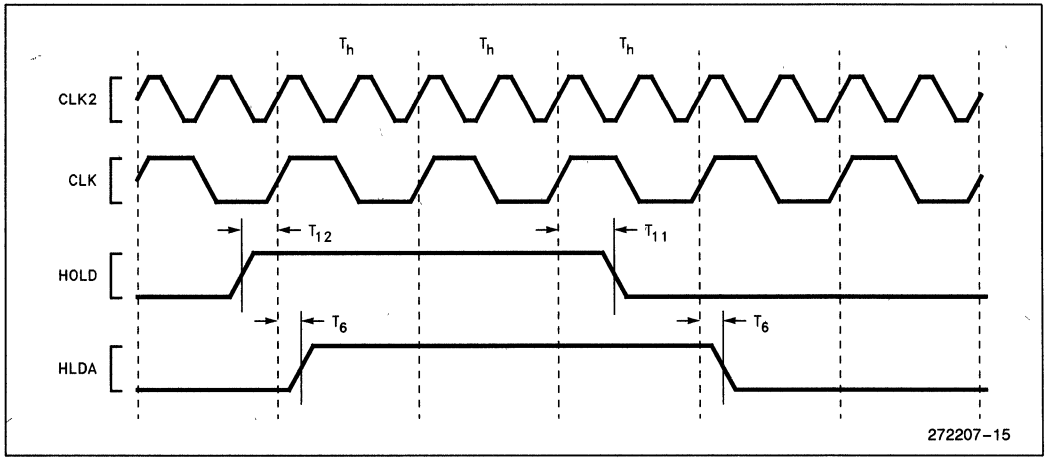


Figure 14. HOLD Timing

3.0 MECHANICAL DATA

3.1 Packaging

The 80960SB is available in two package types:

- 80-lead quad flat pack (EIAJ QFP). Shown in Figure 15.
- 84-lead plastic leaded chip carrier (PLCC). Shown in Figure 16.

Dimensions for both package types are given in the Intel *Packaging* handbook (Order #240800).

3.2 Pin Assignment

The QFP and PLCC have different pin assignments. The QFP pins are numbered in order from 1 to 80 around the package perimeter. The PLCC pins are numbered in order from 1 to 84 around the package perimeter. Table 9 and Table 10 list the function of each QFP pin; Table 11 and Table 12 list the function of each PLCC pin.

V_{CC} and GND connections must be made to multiple V_{CC} and GND pins. Each V_{CC} and GND pin must be connected to the appropriate voltage or ground and externally strapped close to the package. It is recommended that you include separate power and ground planes in your circuit board for power distribution.

Pins identified as NC (No Connect) should never be connected.

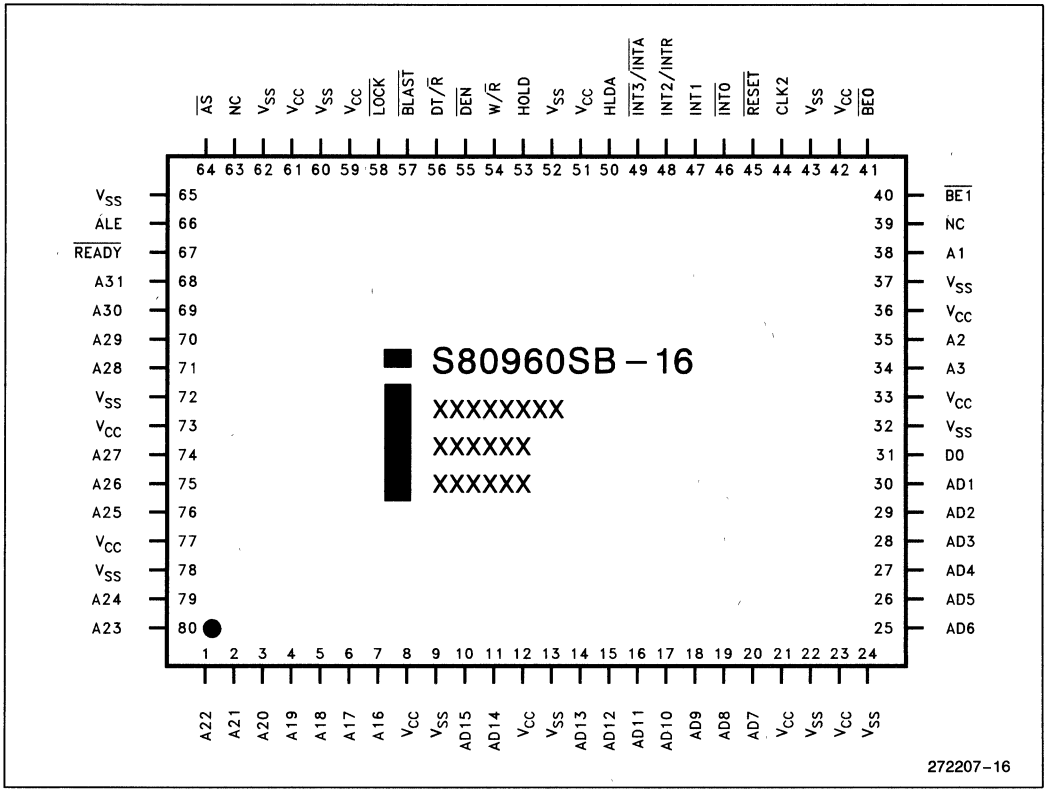


Figure 15. 80-Lead EIAJ Quad Flat Pack (QFP) Package

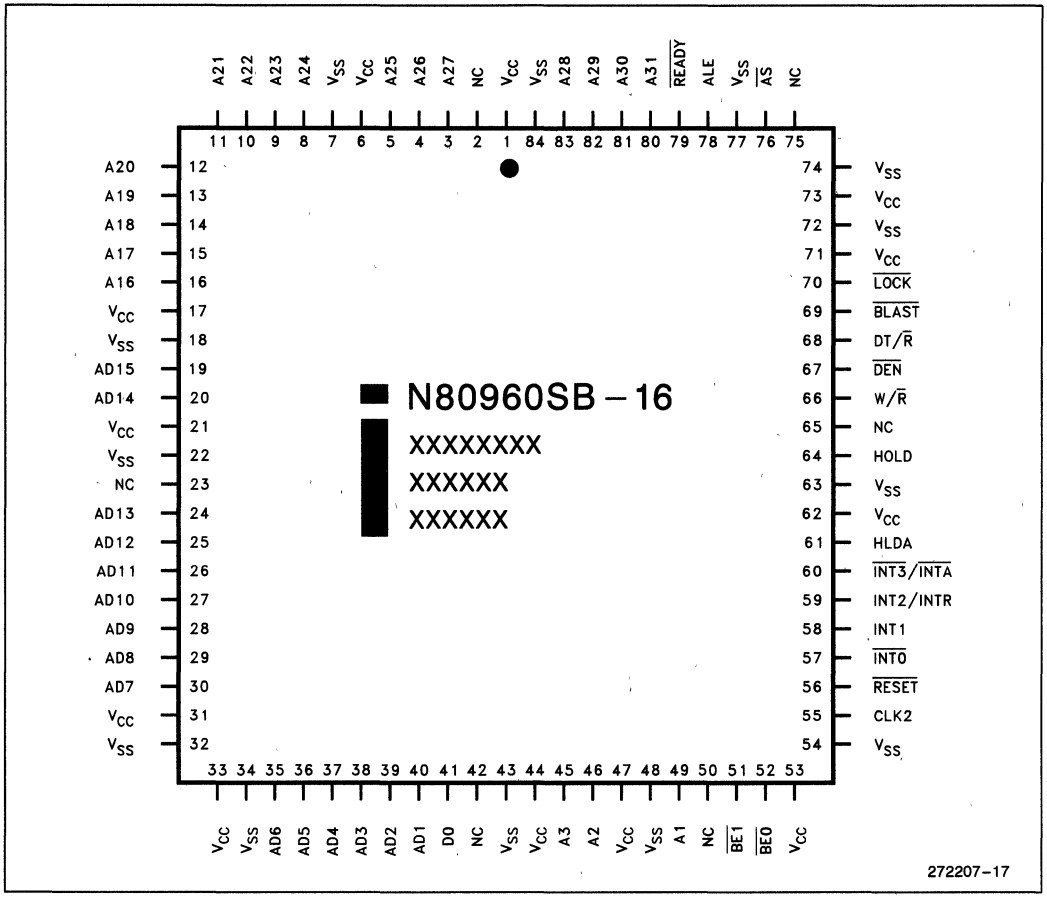


Figure 16. 84-Lead Plastic Leaded Chip Carrier (PLCC) Package

3.3 Pinout

Table 9. 80960SB QFP Pinout—In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	A22	21	V_{CC}	41	$\overline{BE0}$	61	V_{CC}
2	A21	22	V_{SS}	42	V_{CC}	62	V_{SS}
3	A20	23	V_{CC}	43	V_{SS}	63	NC
4	A19	24	V_{SS}	44	CLK2	64	$\overline{AS}$
5	A18	25	AD6	45	$\overline{RESET}$	65	V_{SS}
6	A17	26	AD5	46	$\overline{INT0}$	66	ALE
7	A16	27	AD4	47	INT1	67	$\overline{READY}$
8	V_{CC}	28	AD3	48	INT2/INTR	68	A31
9	V_{SS}	29	AD2	49	$\overline{INT3}/\overline{INTA}$	69	A30
10	AD15	30	AD1	50	HLDA	70	A29
11	AD14	31	D0	51	V_{CC}	71	A28
12	V_{CC}	32	V_{SS}	52	V_{SS}	72	V_{SS}
13	V_{SS}	33	V_{CC}	53	HOLD	73	V_{CC}
14	AD13	34	A3	54	$W/\overline{R}$	74	A27
15	AD12	35	A2	55	$\overline{DEN}$	75	A26
16	AD11	36	V_{CC}	56	$DT/\overline{R}$	76	A25
17	AD10	37	V_{SS}	57	$\overline{BLAST}$	77	V_{CC}
18	AD9	38	A1	58	$\overline{LOCK}$	78	V_{SS}
19	AD8	39	NC	59	V_{CC}	79	A24
20	AD7	40	$\overline{BE1}$	60	V_{SS}	80	A23

NOTE:
Do not connect any external logic to any pins marked NC.

Table 10. 80960SB QFP Pinout—In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
A1	38	A18	5	D0	31	V_{CC}	51
A2	35	A19	4	$\overline{DEN}$	55	V_{CC}	59
A3	34	A20	3	DT/$\overline{R}$	56	V_{CC}	61
AD1	30	A21	2	HLDA	50	V_{CC}	73
AD2	29	A22	1	HOLD	53	V_{CC}	77
AD3	28	A23	80	$\overline{INT0}$	46	V_{CC}	8
AD4	27	A24	79	INT1	47	V_{SS}	13
AD5	26	A25	76	INT2/INTR	48	V_{SS}	22
AD6	25	A26	75	$\overline{INT3}$/$\overline{INTA}$	49	V_{SS}	24
AD7	20	A27	74	$\overline{LOCK}$	58	V_{SS}	32
AD8	19	A28	71	NC	39	V_{SS}	37
AD9	18	A29	70	NC	63	V_{SS}	43
AD10	17	A30	69	$\overline{READY}$	67	V_{SS}	52
AD11	16	A31	68	$\overline{RESET}$	45	V_{SS}	60
AD12	15	ALE	66	V_{CC}	12	V_{SS}	62
AD13	14	$\overline{AS}$	64	V_{CC}	21	V_{SS}	72
AD14	11	$\overline{BE0}$	41	V_{CC}	23	V_{SS}	78
AD15	10	$\overline{BE1}$	40	V_{CC}	33	V_{SS}	9
A16	7	$\overline{BLAST}$	57	V_{CC}	36	V_{SS}	65
A17	6	CLK2	44	V_{CC}	42	W/$\overline{R}$	54

NOTE:
Do not connect any external logic to any pins marked NC.

Table 11. 80960SB PLCC Pinout—In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	V_{CC}	22	V_{SS}	43	V_{SS}	64	HOLD
2	NC	23	NC	44	V_{CC}	65	NC
3	A27	24	AD13	45	A3	66	W/$\overline{R}$
4	A26	25	AD12	46	A2	67	$\overline{DEN}$
5	A25	26	AD11	47	V_{CC}	68	DT/$\overline{R}$
6	V_{CC}	27	AD10	48	V_{SS}	69	$\overline{BLAST}$
7	V_{SS}	28	AD9	49	A1	70	$\overline{LOCK}$
8	A24	29	AD8	50	NC	71	V_{CC}
9	A23	30	AD7	51	$\overline{BE1}$	72	V_{SS}
10	A22	31	V_{CC}	52	$\overline{BE0}$	73	V_{CC}
11	A21	32	V_{SS}	53	V_{CC}	74	V_{SS}
12	A20	33	V_{CC}	54	V_{SS}	75	NC
13	A19	34	V_{SS}	55	CLK2	76	AS
14	A18	35	AD6	56	$\overline{RESET}$	77	V_{SS}
15	A17	36	AD5	57	$\overline{INT0}$	78	ALE
16	A16	37	AD4	58	INT1	79	$\overline{READY}$
17	V_{CC}	38	AD3	59	INT2/INTR	80	A31
18	V_{SS}	39	D2	60	$\overline{INT3}$/$\overline{INTA}$	81	A30
19	AD15	40	D1	61	HLDA	82	A29
20	AD14	41	D0	62	V_{CC}	83	A28
21	V_{CC}	42	NC	63	V_{SS}	84	V_{SS}

NOTE:
Do not connect any external logic to any pins marked NC.

Table 12. 80960SB PLCC Pinout—In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
A1	49	A18	14	DT/$\overline{R}$	68	V_{CC}	44
A2	46	A19	13	HLDA	61	V_{CC}	47
A3	45	A20	12	HOLD	64	V_{CC}	53
D0	41	A21	11	$\overline{INT0}$	57	V_{CC}	6
AD1	40	A22	10	INT1	58	V_{CC}	62
AD2	39	A23	9	INT2/INTR	59	V_{CC}	71
AD3	38	A24	8	$\overline{INT3}$/$\overline{INTA}$	60	V_{CC}	73
AD4	37	A25	5	$\overline{LOCK}$	70	V_{SS}	18
AD5	36	A26	4	NC	2	V_{SS}	22
AD6	35	A27	3	NC	23	V_{SS}	32
AD7	30	A28	83	NC	42	V_{SS}	34
AD8	29	A29	82	NC	50	V_{SS}	43
AD9	28	A30	81	NC	65	V_{SS}	48
AD10	27	A31	80	NC	75	V_{SS}	54
AD11	26	ALE	78	$\overline{READY}$	79	V_{SS}	63
AD12	25	$\overline{AS}$	76	$\overline{RESET}$	56	V_{SS}	7
AD13	24	$\overline{BE0}$	52	V_{CC}	1	V_{SS}	72
AD14	20	$\overline{BE1}$	51	V_{CC}	17	V_{SS}	74
AD15	19	$\overline{BLAST}$	69	V_{CC}	21	V_{SS}	77
AD16	16	CLK2	55	V_{CC}	31	V_{SS}	84
A17	15	$\overline{DEN}$	67	V_{CC}	33	W/$\overline{R}$	66

NOTE:
Do not connect any external logic to any pins marked NC.

3.4 Package Thermal Specification

The 80960SB is specified for operation when case temperature is within the range 0°C to +85°C (PLCC) or 0°C to +100°C (QFP). Measure case temperature at the top center of the package. Ambient temperature can be calculated from:

$T_J = T_C + P^*\theta_{JC}$
$T_A = T_J - P^*\theta_{JA}$
$T_C = T_A + P^*[\theta_{JA} - \theta_{JC}]$

Compute P by multiplying the maximum voltage by the typical current at maximum temperature. Values for θ_{JA} and θ_{JC} for various airflows are given in Table 13 for the QFP package and in Table 14 for the PLCC package. I_{CC} at maximum temperature is typically 80 percent of specified I_{CC} maximum (cold).

Table 13. 80960SB QFP Package Thermal Characteristics

Thermal Resistance—°C/Watt							
Parameter	Airflow—ft./min (m/sec)						
	0	50	100	200	400	600	800
θ Junction-to-Ambient (Case measured in the middle of the top of the package) (No heatsink)	54	52	49	45	39	35	33
θ Junction-to-Case	11	11	11	11	11	11	11

NOTE:
This table applies to 80960SB QFP soldered directly to board.

Table 14. 80960SB PLCC Package Thermal Characteristics

Thermal Resistance—°C/Watt								
Parameter	Airflow—ft./min (m/sec)							
	0	50	100	200	400	600	800	1000
θ Junction-to-Ambient (No heatsink)	33	31	28.5	27	24	22	20	19.5
θ Junction-to-Case	11	11	11	11	11	11	11	11

NOTE:
This table applies to 80960SB PLCC soldered directly to board.

4.0 WAVEFORMS

Figures 17, 18, 19, 20, and 21 show waveforms for various transactions on the 80960SB's bus. Figure 22 shows a cold reset functional waveform.

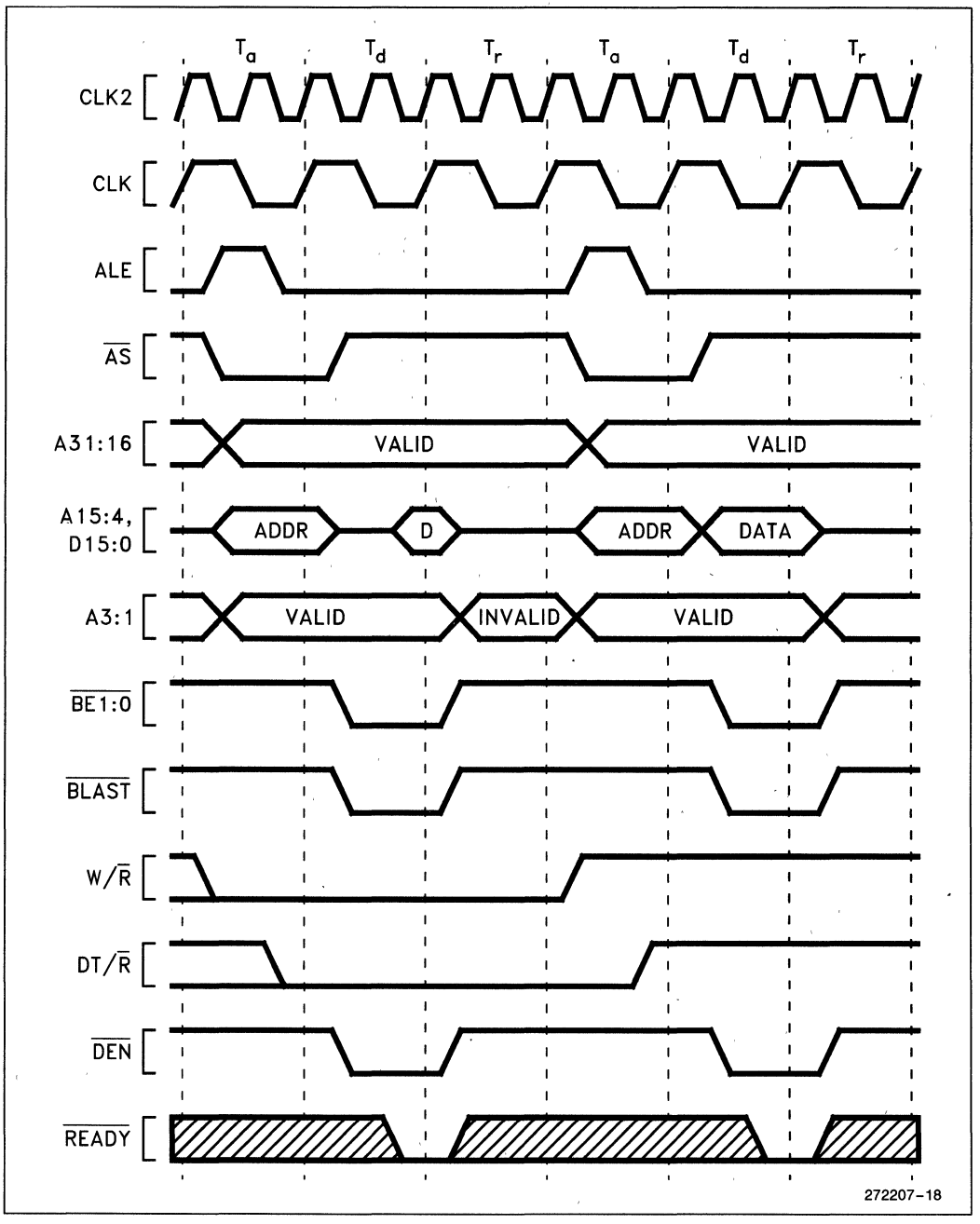


Figure 17. Non-Burst Read and Write Transactions Without Wait States

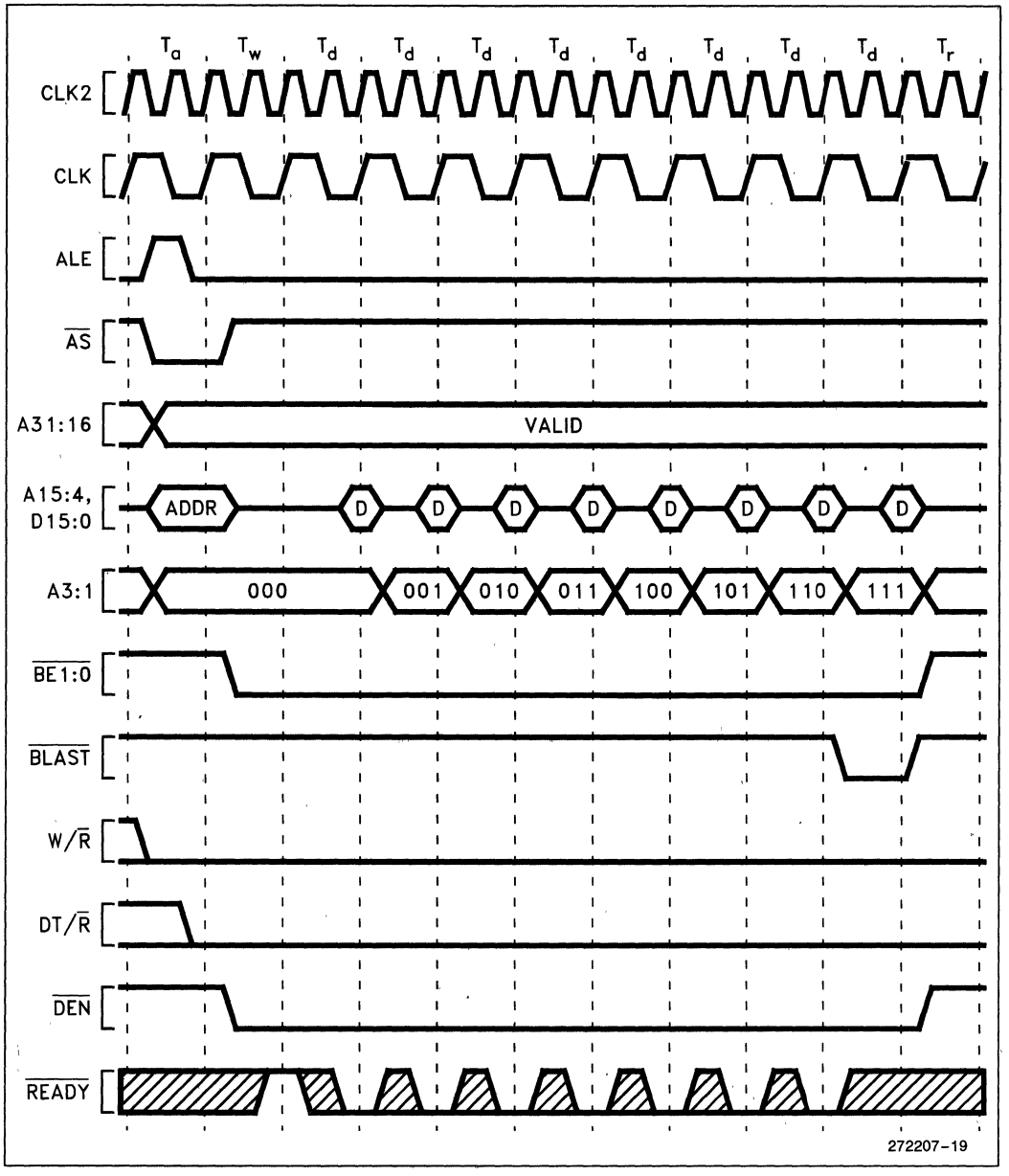


Figure 18. Quad Word Burst Read Transaction with 1, 0, 0, 0, 0, 0, 0, 0 Wait States

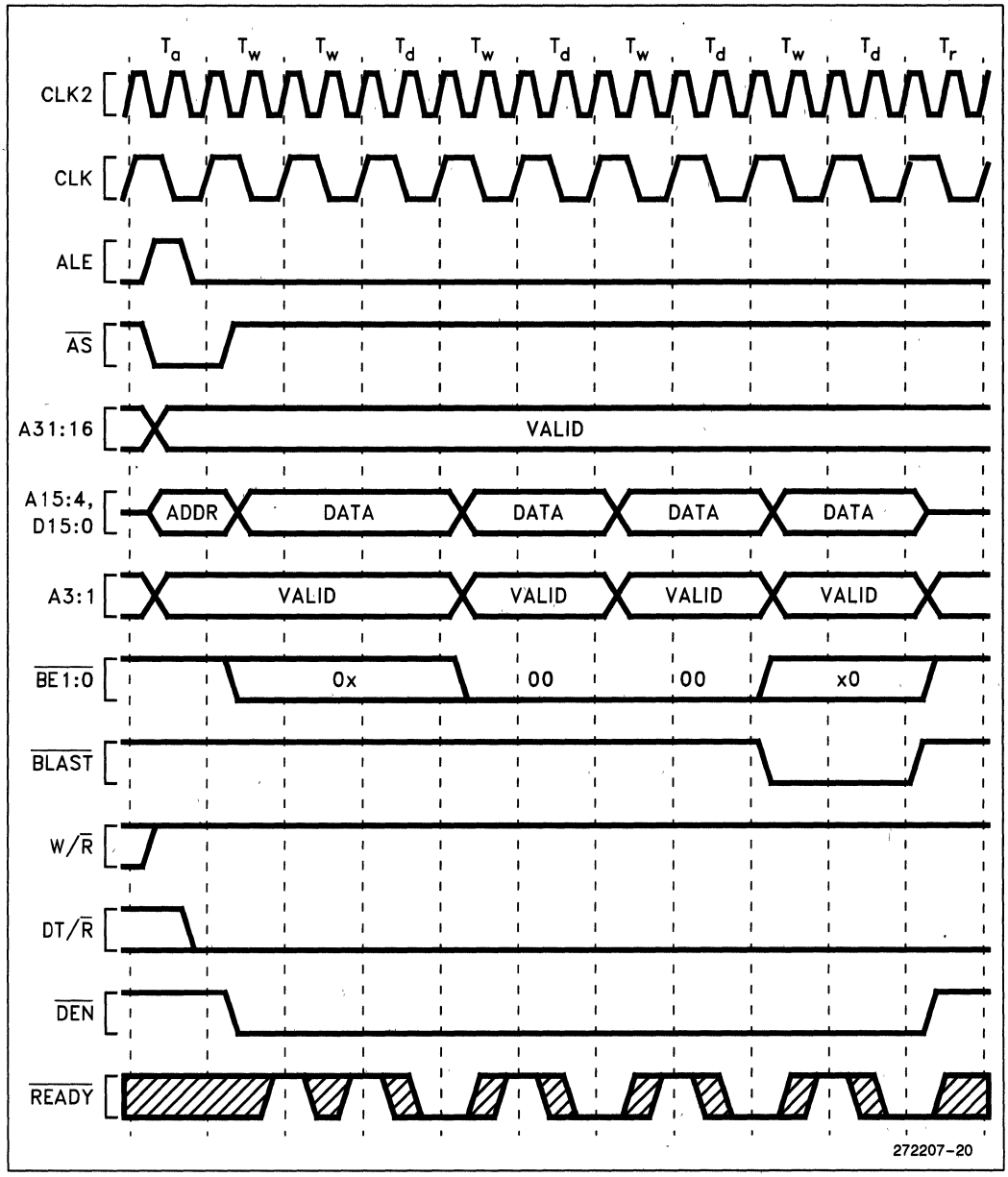


Figure 19. Burst Write Transaction with 2, 1, 1, 1 Wait States (6–8 Bytes Transferred)

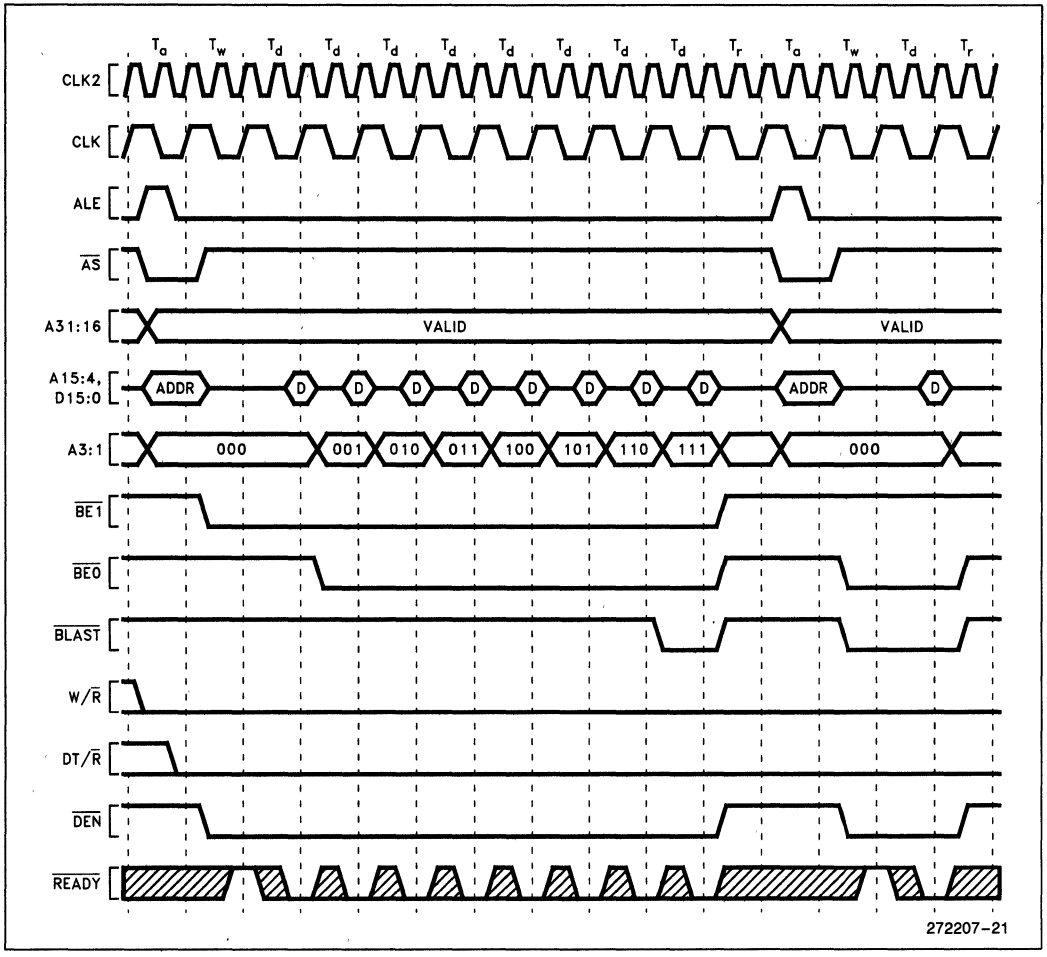


Figure 20. Accesses Generated by Quad Word Read Bus Request, Misaligned One Byte from Quad Word Boundary (1, 0, 0, 0, 0, 0, 0, 0) Wait States

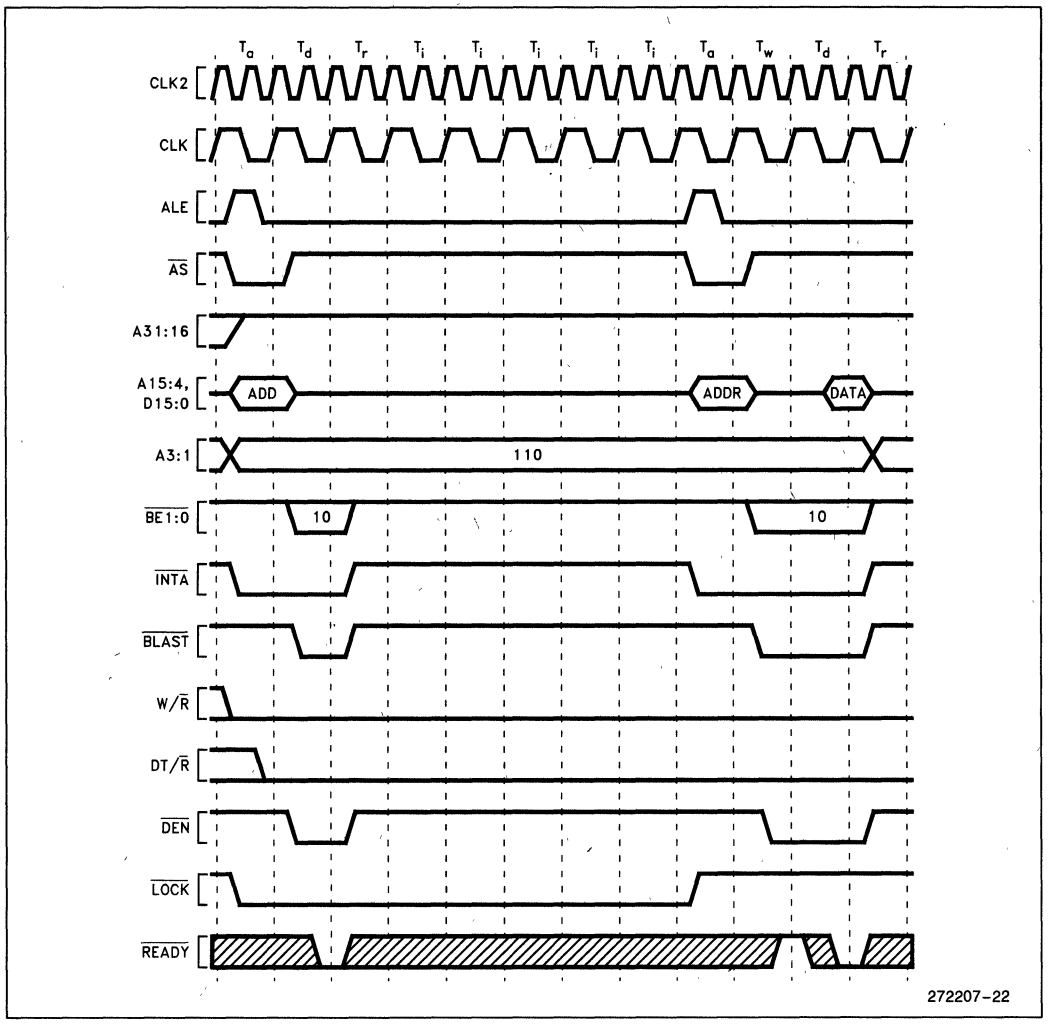


Figure 21. Interrupt Acknowledge Cycle

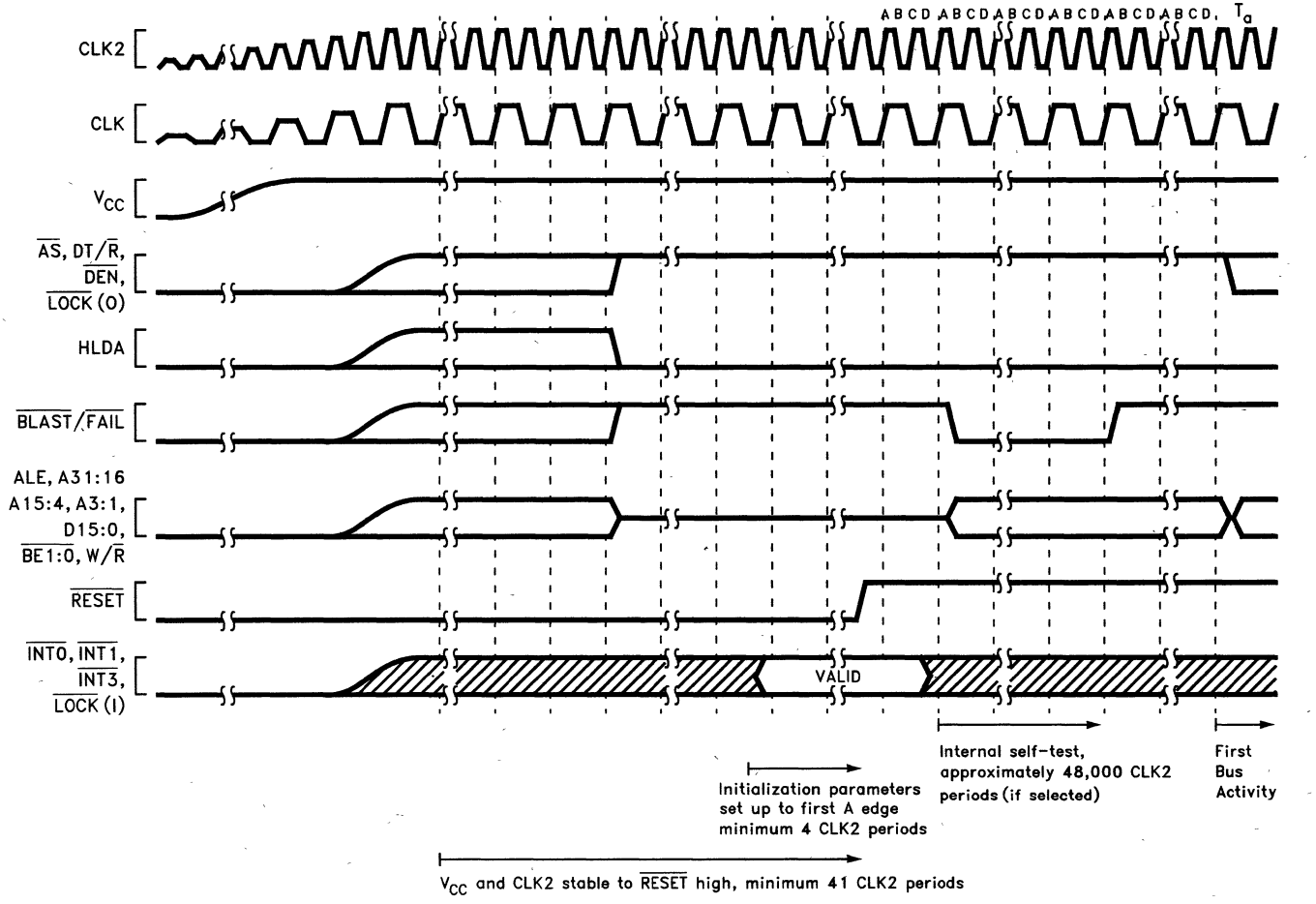


Figure 22. Cold Reset Waveform

5.0 REVISION HISTORY

This data sheet supersedes data sheet 272207-001. The sections significantly changed since the previous revision are:

Section	Last Rev.	Description
2.3 Connection Recommendations (pg.11)	−001	Remove two $\overline{LOCK}$ pin Connection Recommendation figures and added Firgure 5 to reflect the new $\overline{LOCK}$ pin connection recommodation of a single 910Ω pullup resistor.
2.5 Test Load Circuit (pg. 13)	−001	Obsolute figure (Test Load Curcuit for Open-Drain output Pins) removed to relflect current test conditions.
2.7 DC Characteristics (pg. 14)	−001	I_{OL} value improved. WAS: 2.5 mA IS: 4.0 mA $\overline{LOCK}$ pin I_{OL} value at 0.45V relaxed. WAS: 12 mA IS: 6 mA $\overline{LOCK}$ pin I_{OL} value at 0.60V deleted.

This data sheet supersedes data sheet 270917-004, which applied to both the 80960SA and the 80960SB. The 80960SA is now documented in 272206-002.

The sections significantly changed since the previous revision are:

Section	Last Rev.	Description
2.3 Connection Recommendations (pg. 15)	−004	Deleted corresponding graph of Open Drain Voltage vs. Output Current.
Figure 7. Typical Supply Current vs. Case Temperature (pg. 16), Figure 8. Typical Current vs. Frequency (Room Temp) (pg. 16) and Figure 9. Typical Current vs. Frequency (Hot Temp) (pg. 17)	−004	Regraphed data in three graphs instead of two.
Table 6. DC Characteristics (pg. 19)	−004	Input Leakage Current (I_{L12}) specification added to accurately describe leakage of $\overline{INT0}$ and $\overline{LOCK}$ as inputs.
Table 7. 80960SB AC Characteristics (10 MHz) (pg. 21) and Table 8. 80960SB AC Characteristics (16 MHz) (pg. 22)	−004	T_7 minimum specification improved: Power Supply Current: Was: Is: 10 MHz 24 ns T_1−11 ns 16 MHz 15 ns T_1−11 ns

The sections significantly changed between revisions -003 and -004 of the 80960SA/SB data sheet were:

Section	Last Rev.	Description
DC Characteristics (pg. 15).	−003	Operating temperature for PLCC package changed: Was: T_{CASE} = 0°C to +100°C Is: T_{CASE} = 0°C to +85°C The test program has not changed.
Table 7. QFP Package, Thermal Resistance—°C/Watt (pg. 21)	−003	Corrected QFP Package Thermal Resistance values: for θ_{JA} at 0 ft/min airflow: Was: 45.7°/W Is: 54°/W for θ_{JC} at 0 ft/min airflow: Was: 4°/W Is: 11°/W
Table 8. PLCC Package, Thermal Resistance—°C/Watt (pg. 22)	−003	Corrected PLCC Package Thermal Resistance values: for θ_{JA}: at 50 ft/min airflow Was: NA Is: 31 at 100 ft/min airflow Was: NA Is: 28.5 for θ_{JC}: at 0 ft/min airflow Was: 13 Is: 11 at 50 ft–1000 ft/min airflow Was: NA Is: 11
Table 9. 8096SA and 80960SB QFP Pinout — In Pin Order (pg. 23)	−003	Signal A12 incorrectly shown as Pin 28; is now correctly shown as Pin 38. Note added to clarify No Connect Pins.

i960 KA/KB PROCESSOR PRODUCT OVERVIEW

INTRODUCTION

This chapter provides an overview of the Intel i960 KB processor (which is part of the i960 K series of embedded-processor products).

All of the processors in the i960 K series of products are based on the Intel i960 architecture. Most of the information in this overview also applies to the i960 KA processor. The only difference between the i960 KB and i960 KA processors is that the i960 KA processor does not provide on-chip support for floating-point operations or operations on decimal numbers.

OVERVIEW OF THE i960 KB ARCHITECTURE

The i960 KB processor introduced the i960 architecture—a new 32-bit architecture from Intel. This architecture has been designed to meet the needs of embedded applications such as machine control, robotics, process control, avionics and instrumentation.

The i960 architecture can best be characterized as a high-performance computing engine. It features high-speed instruction execution and ease of programming. It is also easily extensible, allowing processors and controllers based on this architecture to be conveniently customized to meet the needs of specific processing and control applications.

The following are some of the important attributes of the i960 architecture:

- full 32-bit registers
- high-speed, pipelined instruction execution
- a convenient program execution environment with 32 general-purpose registers and a versatile set of special-function registers
- a highly optimized procedure call mechanism that features on-chip caching of local variables and parameters
- extensive facilities for handling interrupts and faults
- extensive tracing facilities to support efficient program debugging and monitoring
- register scoreboarding and write buffering to permit efficient operation when used with lower performance memory subsystems

OVERVIEW OF THE SINGLE PROCESSOR SYSTEM ARCHITECTURE

The central processing module, memory module and I/O module form the natural boundaries for the hardware system architecture. The modules are connected together by the high bandwidth 32-bit multiplexed L-bus, which can transfer data at a maximum sustained rate of 53 Mbytes per second for an i960 processor operating at 20 MHz.

Figure 1 shows a simplified block diagram of one possible system configuration. The heart of this system is the i960 KB processor, which fetches instructions, executes code, manipulates stored information and interacts with I/O devices. The high bandwidth L-bus connects the i960 KB processor to memory and I/O modules. The i960 KB processor stores system data, instructions and programs in the memory module. By accessing various peripheral devices in the I/O module, the i960 KB processor supports communication to terminals, modems, printers, disks and other I/O devices.

i960 KB Processor and the L-Bus

The i960 KB processor performs bus operations using multiplexed address and data signals, and provides all the necessary control signals. For example standard control signals, such as Address Latch Enable ($\overline{ALE}$), Address/Data Status ($\overline{ADS}$), Write/Read Command (W/$\overline{R}$), Data Transmit/Receive (DT/$\overline{R}$) and Data Enable ($\overline{DEN}$), are provided by the i960 KB processor. The i960 processor also generates byte enable signals that specify which bytes on the 32-bit data lines are valid for the transfer.

The L-bus supports burst transactions, which access up to four data words at a maximum rate of one word per clock cycle. The i960 KB processor uses the two low-order address lines to indicate how many words are to be transferred. The i960 KB processor performs burst transactions to load the on-chip 512-byte instruction cache to minimize memory accesses for instruction fetches. Burst transactions can also be used for data access.

To transfer control of the bus to an external bus master, the i960 KB provides two arbitration signals: hold request (HOLD) and hold acknowledge (HLDA). After receiving HOLD, the processor grants control of the bus to an external master by asserting HLDA.

June 1991
Order Number: 272030-001

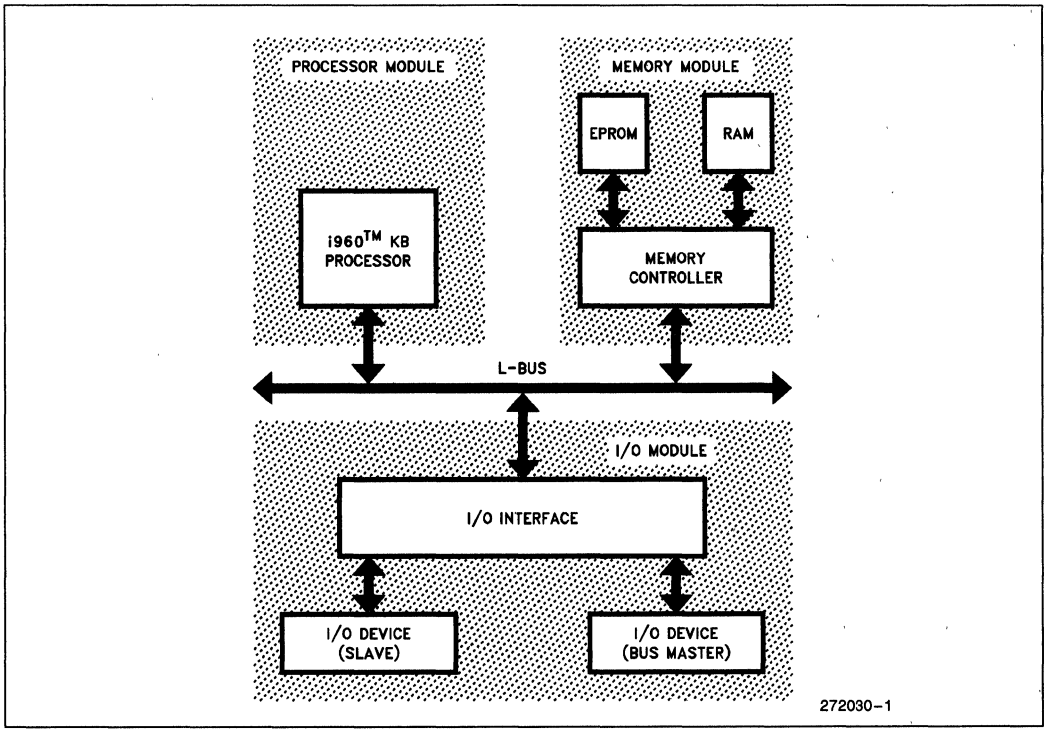


Figure 1. Basic i960 KB System Configuration

The i960 KB processor provides a flexible interrupt structure by using an on-chip interrupt controller, an external interrupt controller or both. The type of interrupt structure is specified by an internal interrupt vector register. For a system with multiple processors, another method is available, called inter-agent communication (IAC) where a processor can interrupt another processor by sending an IAC message.

Memory Module

A memory module can consist of a memory controller, Erasable Programmable Read Only Memory (EPROM), and static or dynamic Random Access Memory (RMA). The memory controller first conditions the L-bus signals for memory operation. It demultiplexes the address and data lines, generates the chip select signals from the address, detects the start of the cycle for burst mode operation and latches the byte enable signals.

The memory controller generates the control signals for EPROM, SRAM and DRAM. Specifically, it provides the control signals, multiplexed row/column address and refresh control for dynamic RAMs. The controller can be designed to accommodate the burst transaction of the i960 KB processor by using the static column mode or nibble mode features of the dynamic RAM. In addition to supplying the operational signals, the controller generates the READY signal to indicate that data can be transferred to or from the i960 KB processor.

The i960 KB processor directly addresses up to 4 Gbytes of physical memory. The processor does not allow burst accesses to cross a 16-byte boundary, to ease the design of the controller. Each address specifies a four-byte data word within the block. Individual data bytes can be accessed by using the four byte-enable signals from the i960 KB processor. Chapter 5 provides design guidelines for the memory controller.

I/O Module

The I/O module consists of the I/O components and the interface circuit. I/O components can be used to allow the i960 KB processor to use most of its clock cycles for computational and system management activities. Time consuming tasks can be off-loaded to specialized slave-type components, such as the 8259A Pro-

grammable Interrupt Controller or the 82530 Serial Communication Controller. Some tasks may require a master-type component, such as the 82586 Local Area Network Control.

The interface circuit performs several functions. It demultiplexes the address and data lines, generates the chip select signals from the address, produces the I/O read or I/O write command from the processor's W/R signal, latches the byte enable signals and generates the READY signals. Since some of these functions are identical to those of the memory controller, the same logic can be used for both interfaces. For master-type peripherals that operate on a 16-bit data bus, the interface circuit translates the 32-bit data bus to a 16-bit data bus.

The i960 KB processor uses memory-mapped addresses to access I/O devices. This allows the CPU to use many of the same instuctions to exchange information for both memory and peripheral devices. Thus, the powerful memory-type instructions can be used to perform 8-, 16- and 32-bit data transfers.

HIGH PERFORMANCE PROGRAM EXECUTION

Much of the design of the i960 architecture has been aimed at maximizing the processor's computational and data processing speed through the use of increased parallelism. The following paragraphs describe several of the mechanisms and techniques used to accomplish this goal.

Load and Store Model

One of the more important features of the i960 architecture is its performance of most operations on operands in registers, rather than in memory. For example, all arithmetic, logic, comparison, branching and bit operations are performed with registers and literals.

This feature provides two benefits. First, it increases program execution speed by minimizing the number of memory accesses necessary to execute a program. Second, it reduces the memory latency encountered when using slower, lower-cost memory parts.

To support this concept, the architecture provides a generous supply of general-purpose registers. For each procedure, 32 registers are available, 28 of which are available for general use. These registers are divided into two types: global and local. Both types of registers can be used for general storage of operands. The only difference is that global registers retain their contents across procedure boundaries, whereas the processor allocates a new set of local registers each time a new procedure is called.

The architecture also provides a set of fast, versatile load and store instructions. These instructions allow burst transfers of 1, 2, 4, 8, 12 or 16 bytes of information between memory and the registers.

On-Chip Caching of Code and Data

To further reduce memory accesses, the architecture offers two mechanisms for caching code and data on chip: an instruction cache and multiple sets of local registers. The instruction cache allows prefetching of blocks of instruction from memory. This helps ensure that the instruction execution pipeline is supplied with a steady stream of instructions. It also reduces the number of memory accesses required when performing iterative operations such as loops. The architecture allows the size of the instruction cache to vary. For the i960 KB processor, it is 512 bytes.

To optimize the architecture's procedure call mehanism, the processor provides multiple sets of local registers. This allows the processor to perform procedure calls without having to write the local registers out to the stack in memory. The number of register sets depends on the processor implementation. The i960 KB processor provides four sets of local registers.

Overlapped Instruction Execution

The i960 architecture also enchances program execution speed by overlapping the execution of some instructions. In the i960 K series of processors, this is accomplished through register scoreboarding.

Register scoreboarding permits instruction execution to continue while data is being fetched from memory. When a load instruction is executed, the processor sets one or more scoreboard bits to indicate the target registers to be loaded. After the target registers are loaded, the scoreboard bits are cleared. While the target registers are being loaded, the processor is allowed to execute other instructions that do not use these registers.

The processor uses the scoreboard bits to ensure that the target registers are not used until the load is complete. (Scoreboard bits are checked transparently from software.) This technique allows code to be executed such that some instructions can be executed in zero clock cycles (that is, executed for free).

Single-Clock Instructions

The i960 architecture is designed to let a processor execute commonly used instructions, such as moves, adds, subtracts, logical operations and branches, in a minimum number of clock cycles (preferably one cycle). The architecture supports this concept in several

ways. For example, the load and store model described earlier eliminates the clock cycles required to perform memory-to-memory operations, by concentrating on register-to-register operations.

In addition, all of the instructions in the i960 architecture are 32 bits long and aligned on 32-bit boundaries. This lets instructions be decoded in one clock cycle, and eliminates the need for an instruction-alignment stage in the pipeline.

The i960 KB processor takes full advantage of these features of the architecture, resulting in more than 50 instructions that can be executed in a single clock cycle.

Efficient Interrupt Model

The i960 architecture provides an efficient mechanism for servicing interrupts from external sources. To handle interrupts, the processor maintains an interrupt table of 248 interrupt vectors, 240 of which are available for general use. When an interrupt is signaled, the processor uses a pointer to the interrupt table to perform an implicit call to an interrupt handler procedure. In performing this call, the processor automatically saves the state of the processor prior to receiving the interrupt, performs the interrupt routine, then restores the state of the processor. A separate interrupt stack is also provided to segregate interrupt handling from application programs.

The interrupt handling facilities also allow interrupts to be evaluated by priority. The processor is then able to store interrupt vectors that are lower in priority than the current processor task in a pending interrupt section of the interrupt table. The processor checks and services the pending interrupts at defined times.

SIMPLIFIED PROGRAMMING ENVIRONMENT

Because of its streamlined execution environment, processors based on the i960 architecture are particularly easy to program. The following paragraphs describe some of the architecture features that simplify programming.

Highly Efficient Procedure Call Mechanism

The procedure call mechanism makes procedure calls and parameter passing between procedures simple and compact. Each time a call instruction is issued, the processor automatically saves the current set of local registers and allocates a new set for the called procedure. Likewise, on a return from a procedure, the current set of local registers is deallocated and the local registers for the procedure being returned to are restored. This means a program never has to explicitly save and restore those local variables that are stored in local registers.

Versatile Instruction Set and Addressing

The selection of instructions and addressing modes also simplifies programming. A full set of load, store, move, arithmetic, comparison and branch instructions are provided, with operations on both integer and ordinal data types. Operations on bits and bit strings are simplified by a complete set of Boolean and bit-field instructions.

The addressing modes are efficient and straightforward, while at the same time providing the necessary indexing and scaling modes required to address complex arrays and record structures. The large 4-gigabyte address space provides ample room to store programs and data. The availability of 32 addressing lines allows some address lines to be memory-mapped to control hardware functions.

Extensive Fault Handling Capability

To aid in program development, the i960 architecture defines a wide range of faults that the processor detects, including, arithmetic, faults, invalid operations, invalid operands and machine faults. When a fault is detected, the processor makes an implicit call to a fault handler routine, in a way similar to the interrupt mechanism described previously. The information collected for each fault allows program developers to quickly correct faulting code, and allows automatic recovery from some faults.

Debugging and Monitoring

To support debugging systems, the i960 architecture provides a mechanism for monitoring processor activity by means of trace events. When the processor detects a trace event, it signals a trace fault and calls a fault handler. Intel provides several tools that use this feature, including an in-circuit emulator (ICE) device.

SUPPORT FOR ARCHITECTURAL EXENSIONS

The i960 architecture provides several features that enable processors based on this architecture to be easily customized to meet the needs of specific embedded applications, such as signal processing, array processing or graphics processing.

The most important of these features is the set of 32 special function registers. These regisers provide a convenient interface to circuitry in the processor or pins that can be connected to external hardware. They can be used to control timers, to perform operations on special data types or to perform I/O functions. The special function registers are similar to the global registers. They can be addressed by all of the register access instructions.

EXTENSIONS INCLUDED IN THE i960 K SERIES PROCESSORS

The i960 K series of processors provides a complete implementation of the i960 architecture, plus several extensions to that architecture. These extensions fall into two categories: floating-point processing and inter-agent communication.

On-Chip Floating Point

The i960 KB processor provides a complete implementation of the IEEE standard for binary floating-point arithmetic (IEEE 754-185). This implementation includes a full set of floating-point operations, including add, subtract, multiply, divide, trigonometric functions and logarithmic functions. These operations are performed on single precision (32-bit), double precision (64-bit) and extended precision (80-bit) real numbers.

One of the benefits of this implementation is that the floating-point handling facilities are integrated into the normal instruction execution environment. Single and double precision floating-point values are stored in the same registers as non-floating point values. Four 80-bit floating-point registers are provided to hold extended-precision values.

Interagent Communication

All of the processors in the i960 K series provide an inter-agent communication (IAC) mechanism, allowing agents connected to the processor's bus to communicate with one another. This mechanism operates similarly to the interrupt mechanism, except that IAC messages are passed through dedicated sections of memory. The sort of tasks handled with IAC messages are processor reinitialization, stopping the processor, purging the instruction cache and forcing the processor to check pending interrupts.

80960KA
EMBEDDED 32-BIT MICROPROCESSOR

- **High-Performance Embedded Architecture**
  - **25 MIPS Burst Execution at 25 MHz**
  - **9.4 MIPS* Sustained Execution at 25 MHz**
- **512-Byte On-Chip Instruction Cache**
  - **Direct Mapped**
  - **Parallel Load/Decode for Uncached Instructions**
- **Multiple Register Sets**
  - **Sixteen Global 32-Bit Registers**
  - **Sixteen Local 32-Bit Registers**
  - **Four Local Register Sets Stored On-Chip**
  - **Register Scoreboarding**
- **4 Gigabyte, Linear Address Space**
- **Pin Compatible with 80960KB**
- **Built-In Interrupt Controller**
  - **31 Priority Levels, 256 Vectors**
  - **3.4 μs Latency @ 25 MHz**
- **Easy to Use, High Bandwidth 32-Bit Bus**
  - **66.7 Mbytes/s Burst**
  - **Up to 16 Bytes Transferred per Burst**
- **132-Lead Packages**
  - **Pin Grid Array (PGA)**
  - **Plastic Quad Flat-Pack (PQFP)**

1

The 80960KA is a member of Intel's i960® 32-bit processor family, which is designed especially for embedded applications. It includes a 512-byte instruction cache and a built-in interrupt controller. The 80960KA has a large register set, multiple parallel execution units and a high-bandwidth burst bus. Using advanced RISC technology, this high performance processor is capable of execution rates in excess of 9.4 million instructions per second. The 80960KA is well-suited for a wide range of applications including non-impact printers, I/O control and specialty instrumentation.

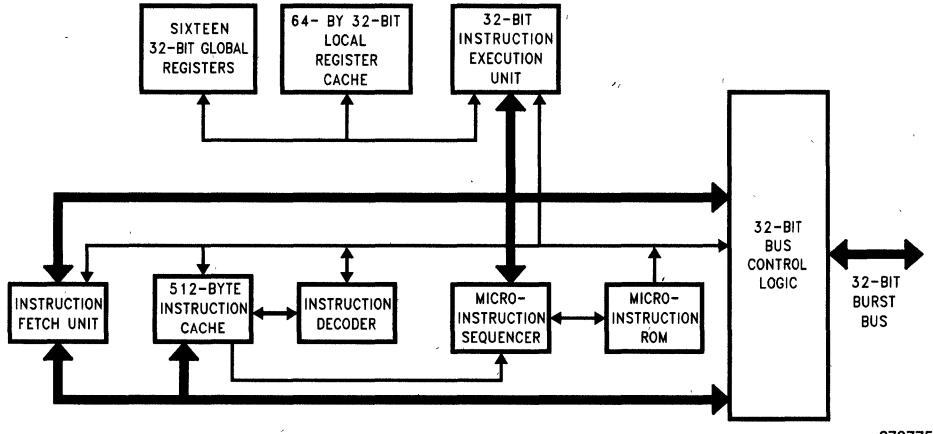


Figure 1. The 80960KA Processor's Highly Parallel Architecture

*Relative to Digital Equipment Corporation's VAX-11/780 at 1 MIPS (VAX-11 is a trademark of Digital Equipment Corporation.)

February 1994
Order Number: 270775-005

1.0 THE i960® PROCESSOR

The 80960KA is a member of the 32-bit architecture from Intel known as the i960 processor family. These were especially designed to serve the needs of embedded applications. The embedded market includes applications as diverse as industrial automation, avionics, image processing, graphics and networking. These types of applications require high integration, low power consumption, quick interrupt response times and high performance. Since time to market is critical, embedded microprocessors need to be easy to use in both hardware and software designs.

All members of the i960 processor family share a common core architecture which utilizes RISC technology so that, except for special functions, the family members are object-code compatible. Each new processor in the family adds its own special set of functions to the core to satisfy the needs of a specific application or range of applications in the embedded market.

Software written for the 80960KA will run without modification on any other member of the 80960 Family. It is also pin-compatible with the 80960KB which includes an integrated floating-point unit and the 80960MC which is a military-grade version that supports multitasking, memory management, multiprocessing and fault tolerance.

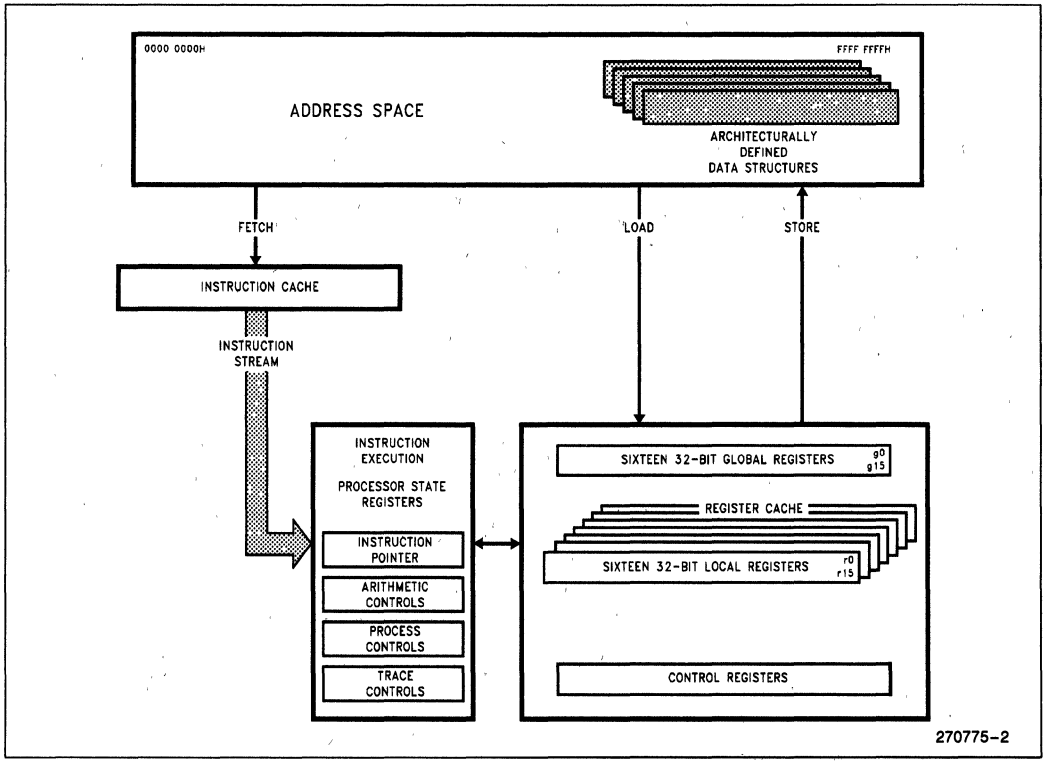


Figure 2. 80960KA Programming Environment

1.1 Key Performance Features

The 80960 architecture is based on the most recent advances in microprocessor technology and is grounded in Intel's long experience in the design and manufacture of embedded microprocessors. Many features contribute to the 80960KA's exceptional performance:

1. **Large Register Set.** Having a large number of registers reduces the number of times that a processor needs to access memory. Modern compilers can take advantage of this feature to optimize execution speed. For maximum flexibility, the 80960KA provides thirty-two 32-bit registers. (See Figure 2.)
2. **Fast Instruction Execution.** Simple functions make up the bulk of instructions in most programs so that execution speed can be improved by ensuring that these core instructions are executed as quickly as possible. The most frequently executed instructions such as register-register moves, add/subtract, logical operations and shifts execute in one to two cycles. (Table 1 contains a list of instructions.)
3. **Load/Store Architecture.** One way to improve execution speed is to reduce the number of times that the processor must access memory to perform an operation. As with other processors based on RISC technology, the 80960KA has a Load/Store architecture. As such, only the LOAD and STORE instructions reference memory; all other instructions operate on registers. This type of architecture simplifies instruction decoding and is used in combination with other techniques to increase parallelism.
4. **Simple Instruction Formats.** All instructions in the 80960KA are 32 bits long and must be aligned on word boundaries. This alignment makes it possible to eliminate the instruction alignment stage in the pipeline. To simplify the instruction decoder, there are only five instruction formats; each instruction uses only one format. (See Figure 3.)
5. **Overlapped Instruction Execution.** Load operations allow execution of subsequent instructions to continue before the data has been retumed from memory, so that these instructions can overlap the load. The 80960KA manages this process transparently to software through the use of a register scoreboard. Conditional instructions also make use of a scoreboard so that subsequent unrelated instructions may be executed while the conditional instruction is pending.
6. **Integer Execution Optimization.** When the result of an arithmetic execution is used as an operand in a subsequent calculation, the value is sent immediately to its destination register. Yet at the same time, the value is put on a bypass path to the ALU, thereby saving the time that otherwise would be required to retrieve the value for the next operation.
7. **Bandwidth Optimizations.** The 80960KA gets optimal use of its memory bus bandwidth because the bus is tuned for use with the on-chip instruction cache: instruction cache line size matches the maximum burst size for instruction fetches. The 80960KB automatically fetches four words in a burst and stores them directly in the cache. Due to the size of the cache and the fact that it is continually filled in anticipation of needed instructions in the program flow, the 80960KA is relatively insensitive to memory wait states. The benefit is that the 80960KA delivers outstanding performance even with a low cost memory system.
8. **Cache Bypass.** If a cache miss occurs, the processor fetches the needed instruction then sends it on to the instruction decoder at the same time it updates the cache. Thus, no extra time is spent to load and read the cache.

Table 1. 80960KA Instruction Set

Data Movement	Arithmetic	Logical	Bit and Bit Field
Load Store Move Load Address	Add Subtract Multiply Divide Remainder Modulo Shift	And Not And And Not Or Exclusive Or Not Or Or Not Exclusive Nor Not Nand Rotate	Set Bit Clear Bit Not Bit Check Bit Alter Bit Scan For Bit Scan Over Bit Extract Modify
Comparison	**Branch**	**Call/Return**	**Fault**
Compare Conditional Compare Compare and Increment Compare and Decrement	Unconditional Branch Conditional Branch Compare and Branch	Call Call Extended Call System Return Branch and Link	Conditional Fault Synchronize Faults
Debug	**Miscellaneous**	**Decimal**	
Modify Trace Controls Mark Force Mark	Atomic Add Atomic Modify Flush Local Registers Modify Arithmetic Controls Scan Byte for Equal Test Condition Code Modify Process Controls	Decimal Move Decimal Add with Carry Decimal Subtract with Carry	
		Synchronous	
		Synchronous Load Synchronous Move	

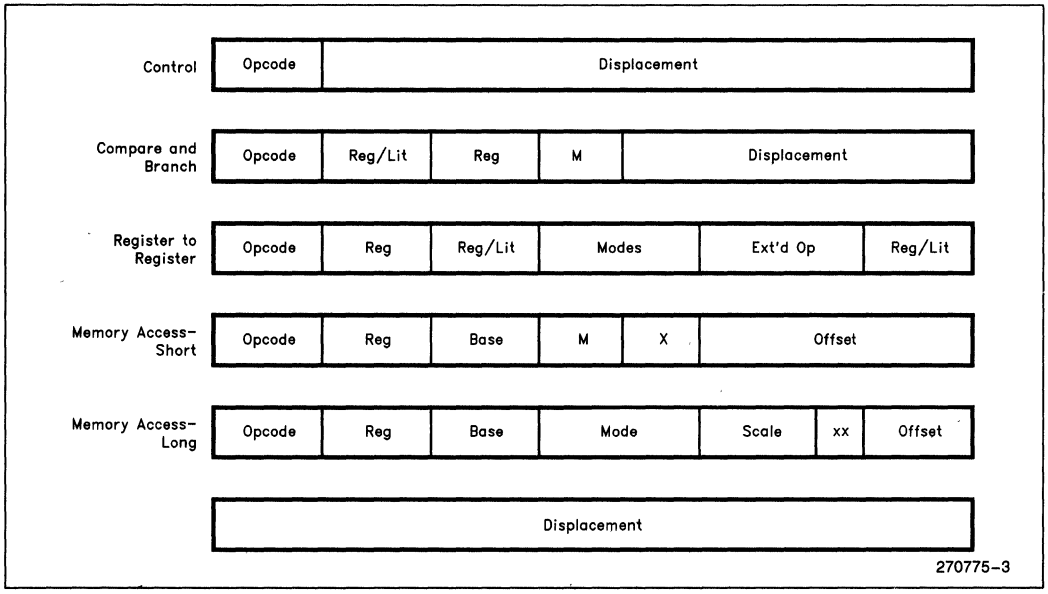


Figure 3. Instruction Formats

1.1.1 MEMORY SPACE AND ADDRESSING MODES

The 80960KA offers a linear programming environment so that all programs running on the processor are contained in a single address space. Maximum address space size is 4 Gigabytes (2^{32} bytes).

For ease of use the 80960KA has a small number of addressing modes, but includes all those necessary to ensure efficient execution of high-level languages such as C. Table 2 lists the modes.

Table 2. Memory Addressing Modes

- 12-Bit Offset
- 32-Bit Offset
- Register-Indirect
- Register + 12-Bit Offset
- Register + 32-Bit Offset
- Register + (Index-Register × Scale-Factor)
- Register × Scale Factor + 32-Bit Displacement
- Register + (Index-Register × Scale-Factor) + 32-Bit Displacement
- Scale-Factor is 1, 2, 4, 8 or 16

1.1.2 DATA TYPES

The 80960KA recognizes the following data types:

Numeric:

- 8-, 16-, 32- and 64-bit ordinals
- 8-, 16-, 32- and 64-bit integers

Non-Numeric:

- Bit
- Bit Field
- Triple Word (96 bits)
- Quad-Word (128 bits)

1.1.3 LARGE REGISTER SET

The 80960KA programming environment includes a large number of registers. In fact, 32 registers are available at any time. The availability of this many registers greatly reduces the number of memory accesses required to perform algorithms, which leads to greater instruction processing speed.

There are two types of general-purpose registers: local and global. The global registers consist of sixteen 32-bit registers (G0 though G15). These registers perform the same function as the general-

purpose registers provided in other popular microprocessors. The term global refers to the fact that these registers retain their contents across procedure calls.

The local registers, on the other hand, are procedure specific. For each procedure call, the 80960KA allocates 16 local registers (R0 through R15). Each local register is 32 bits wide.

1.1.4 MULTIPLE REGISTER SETS

To further increase the efficiency of the register set, multiple sets of local registers are stored on-chip (see Figure 4). This cache holds up to four local register frames, which means that up to three procedure calls can be made without having to access the procedure stack resident in memory.

Although programs may have procedure calls nested many calls deep, a program typically oscillates back and forth between only two to three levels. As a result, with four stack frames in the cache, the probability of having a free frame available on the cache when a call is made is very high. In fact, runs of representative C-language programs show that 80% of the calls are handled without needing to access memory.

If four or more procedures are active and a new procedure is called, the 80960KA moves the oldest local register set in the stack-frame cache to a procedure stack in memory to make room for a new set of registers. Global register G15 is the frame pointer (FP) to the procedure stack.

Global registers are not exchanged on a procedure call, but retain their contents, making them available to all procedures for fast parameter passing.

1.1.5 INSTRUCTION CACHE

To further reduce memory accesses, the 80960KA includes a 512-byte on-chip instruction cache. The instruction cache is based on the concept of locality of reference; most programs are not usually executed in a steady stream but consist of many branches, loops and procedure calls that lead to jumping back and forth in the same small section of code. Thus, by maintaining a block of instructions in cache, the number of memory references required to read instructions into the processor is greatly reduced.

To load the instruction cache, instructions are fetched in 16-byte blocks; up to four instructions can be fetched at one time. An efficient prefetch algorithm increases the probability that an instruction will already be in the cache when it is needed.

Code for small loops often fits entirely within the cache, leading to a great increase in processing speed since further memory references might not be necessary until the program exits the loop. Similarly, when calling short procedures, the code for the calling procedure is likely to remain in the cache so it will be there on the procedure's return.

1.1.6 REGISTER SCOREBOARDING

The instruction decoder is optimized in several ways. One optimization method is the ability to overlap instructions by using register scoreboarding.

Register scoreboarding occurs when a LOAD moves a variable from memory into a register. When the instruction initiates, a scoreboard bit on the target register is set. Once the register is loaded, the bit is reset. In between, any reference to the register contents is accompanied by a test of the scoreboard bit to ensure that the load has completed before processing continues. Since the processor does not need to wait for the LOAD to complete, it can execute additional instructions placed between the LOAD and the instruction that uses the register contents, as shown in the following example:

```
ld data_2, r4
ld data_2, r5
Unrelated instruction
Unrelated instruction
add R4, R5, R6
```

In essence, the two unrelated instructions between LOAD and ADD are executed "for free" (i.e., take no apparent time to execute) because they are executed while the register is being loaded. Up to three load instructions can be pending at one time with three corresponding scoreboard bits set. By exploiting this feature, system programmers and compiler writers have a useful tool for optimizing execution speed.

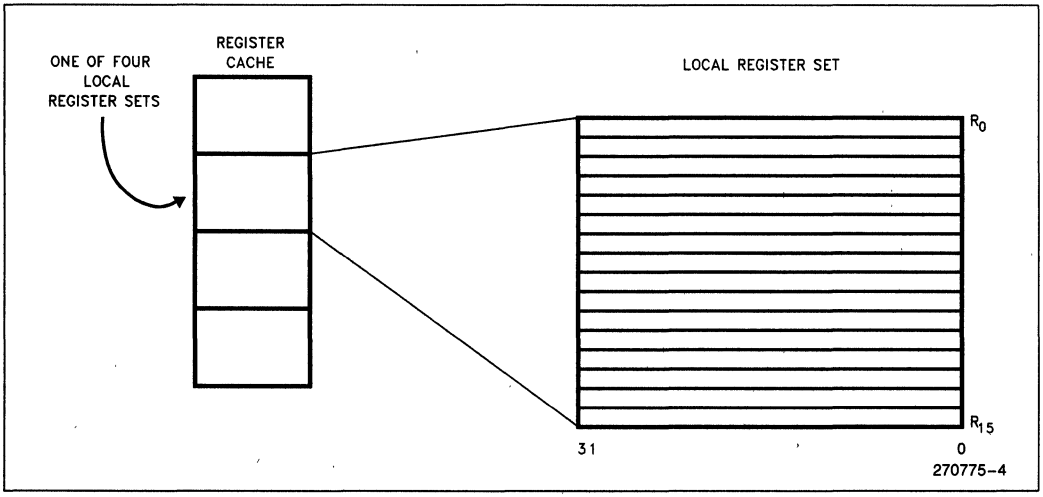


Figure 4. Multiple Register Sets Are Stored On-Chip

1.1.7 HIGH BANDWIDTH LOCAL BUS

The 80960KA CPU resides on a high-bandwidth address/data bus known as the local bus (L-Bus). The L-Bus provides a direct communication path between the processor and the memory and I/O subsystem interfaces. The processor uses the L-Bus to fetch instructions, manipulate memory and respond to interrupts. L-Bus features include:

- 32-bit multiplexed address/data path
- Four-word burst capability which allows transfers from 1 byte to 16 bytes at a time
- High bandwidth reads and writes with 66.7 Mbytes/s burst (at 25 MHz)

Table 3 defines L-bus signal names and functions; Table 4 defines other component-support signals such as interrupt lines.

1.1.8 INTERRUPT HANDLING

The 80960KA can be interrupted in two ways: by the activation of one of four interrupt pins or by sending a message on the processor's data bus.

The 80960KA is unusual in that it automatically handles interrupts on a priority basis and can keep track of pending interrupts through its on-chip interrupt controller. Two of the interrupt pins can be configured to provide 8259A-style handshaking for expansion beyond four interrupt lines.

1.1.9 DEBUG FEATURES

The 80960KA has built-in debug capabilities. There are two types of breakpoints and six trace modes. Debug features are controlled by two internal 32-bit registers: the Process-Controls Word and the Trace-Controls Word. By setting bits in these control words, a software debug monitor can closely control how the processor responds during program execution.

The 80960KA provides two hardware breakpoint registers on-chip which, by using a special command, can be set to any value. When the instruction pointer matches either breakpoint register value, the breakpoint handling routine is automatically called.

The 80960KA also provides software breakpoints through the use of two instructions: MARK and FMARK. These can be placed at any point in a program and cause the processor to halt execution at that point and call the breakpoint handling routine. The breakpoint mechanism is easy to use and provides a powerful debugging tool.

Tracing is available for instructions (single step execution), calls and returns and branching. Each trace type may be enabled separately by a special

debug instruction. In each case, the 80960KA executes the instruction first and then calls a trace handling routine (usually part of a software debug monitor). Further program execution is halted until the routine completes, at which time execution resumes at the next instruction. The 80960KA's tracing mechanisms, implemented completely in hardware, greatly simplify the task of software test and debug.

1.1.10 FAULT DETECTION

The 80960KA has an automatic mechanism to handle faults. Fault types include trace and arithmetic faults. When the processor detects a fault, it automatically calls the appropriate fault handling routine and saves the current instruction pointer and necessary state information to make efficient recovery possible. Like interrupt handling routines, fault handling routines are usually written to meet the needs of specific applications and are often included as part of the operating system or kernel.

For each of the fault types, there are numerous subtypes that provide specific information about a fault. The fault handler can use this specific information to respond correctly to the fault.

1.1.11 BUILT-IN TESTABILITY

Upon reset, the 80960KA automatically conducts an exhaustive internal test of its major blocks of logic. Then, before executing its first instruction, it does a zero check sum on the first eight words in memory to ensure that the memory image was programmed correctly. If a problem is discovered at any point during the self-test, the 80960KA asserts its $\overline{\text{FAILURE}}$ pin and will not begin program execution. Self test takes approximately 47,000 cycles to complete.

System manufacturers can use the 80960KA's self-test feature during incoming parts inspection. No special diagnostic programs need to be written. The test is both thorough and fast. The self-test capability helps ensure that defective parts are discovered before systems are shipped and, once in the field, the self-test makes it easier to distinguish between problems caused by processor failure and problems resulting from other causes.

1.1.12 CHMOS

The 80960KA is fabricated using Intel's CHMOS IV (Complementary High Speed Metal Oxide Semiconductor) process. The 80960KA is currently available in 16, 20 and 25 MHz versions.

Table 3. 80960KA Pin Description: L-Bus Signals

Name	Type	Description
CLK2	I	**SYSTEM CLOCK** provides the fundamental timing for 80960KA systems. It is divided by two inside the 80960KA to generate the internal processor clock.
LAD31:0	I/O T.S.	**LOCAL ADDRESS/DATA BUS** carries 32-bit physical addresses and data to and from memory. During an address (T_a) cycle, bits 2–31 contain a physical word address (bits 0–1 indicate SIZE; see below). During a data (T_d) cycle, bits 0–31 contain read or write data. These pins float to a high impedance state when not active. Bits 0–1 comprise SIZE during a T_a cycle. SIZE specifies burst transfer size in words: **LAD1 LAD0** 0 0 1 Word 0 1 2 Words 1 0 3 Words 1 1 4 Words
$\overline{ALE}$	O T.S.	**ADDRESS LATCH ENABLE** indicates the transfer of a physical address. $\overline{ALE}$ is asserted during a T_a cycle and deasserted before the beginning of the T_d state. It is active LOW and floats to a high impedance state during a hold cycle (T_h).
$\overline{ADS}$	O O.D.	**ADDRESS/DATA STATUS** indicates an address state. $\overline{ADS}$ is asserted every T_a state and deasserted during the following T_d state. For a burst transaction, $\overline{ADS}$ is asserted again every T_d state where $\overline{READY}$ was asserted in the previous cycle.
W/$\overline{R}$	O O.D.	**WRITE/READ** specifies, during a T_a cycle, whether the operation is a write or read. It is latched on-chip and remains valid during T_d cycles.
DT/$\overline{R}$	O O.D.	**DATA TRANSMIT/RECEIVE** indicates the direction of data transfer to and from the L-Bus. It is low during T_a and T_d cycles for a read or interrupt acknowledgment; it is high during T_a and T_d cycles for a write. DT/$\overline{R}$ never changes state when $\overline{DEN}$ is asserted.
$\overline{READY}$	I	**READY** indicates that data on LAD lines can be sampled or removed. If $\overline{READY}$ is not asserted during a T_d cycle, the T_d cycle is extended to the next cycle by inserting a wait state (T_w) and $\overline{ADS}$ is not asserted in the next cycle.
$\overline{LOCK}$	I/O O.D.	**BUS LOCK** prevents bus masters from gaining control of the L-Bus during Read/Modify/Write (RMW) cycles. The processor or any bus agent may assert $\overline{LOCK}$. At the start of a RMW operation, the processor examines the $\overline{LOCK}$ pin. If the pin is already asserted, the processor waits until it is not asserted. If the pin is not asserted, the processor asserts $\overline{LOCK}$ during the T_a cycle of the read transaction. The processor deasserts $\overline{LOCK}$ in the T_a cycle of the write transaction. During the time $\overline{LOCK}$ is asserted, a bus agent can perform a normal read or write but not a RMW operation. The processor also asserts $\overline{LOCK}$ during interrupt-acknowledge transactions. Do not leave $\overline{LOCK}$ unconnected. It must be pulled high for the processor to function properly.

I/O = Input/Output, O = Output, I = Input, O.D. = Open Drain, T.S. = Three-State

Table 3. 80960KA Pin Description: L-Bus Signals (Continued)

Name	Type	Description
$\overline{BE3:0}$	O O.D.	**BYTE ENABLE LINES** specify the data bytes (up to four) on the bus which are used in the current bus cycle. $\overline{BE3}$ corresponds to LAD31:24; $\overline{BE0}$ corresponds to LAD7:0. The byte enables are provided in advance of data: • Byte enables asserted during T_a specify the bytes of the first data word. • Byte enables asserted during T_d specify the bytes of the next data word, if any (the word to be transmitted following the next assertion of $\overline{READY}$). Byte enables that occur during T_d cycles that precede the last assertion of $\overline{READY}$ are undefined. Byte enables are latched on-chip and remain constant from one T_d cycle to the next when $\overline{READY}$ is not asserted. For reads, byte enables specify the byte(s) that the processor will actually use. L-Bus agents are required to assert only adjacent byte enables (e.g., asserting just $\overline{BE0}$ and $\overline{BE2}$ is not permitted) and are required to assert at least one byte enable. Address bits A_0 and A_1 can be decoded externally from the byte enables.
HOLD	I	**HOLD:** A request from an external bus master to acquire the bus. When the processor receives HOLD and grants bus control to another master, it floats its three- state bus lines and open-drain control lines, asserts HLDA and enters the T_h state. When HOLD deasserts, the processor deasserts HLDA and enters the T_i or T_a state.
HLDA	O T.S.	**HOLD ACKNOWLEDGE:** Notifies an external bus master that the processor has relinquished control of the bus.
CACHE	O T.S.	**CACHE** indicates when an access is cacheable during a T_a cycle. It is not asserted during any synchronous access, such as a synchronous load or move instruction used for sending an IAC message. The CACHE signal floats to a high impedance state when the processor is idle.

I/O = Input/Output, O = Output, I = Input, O.D. = Open Drain, T.S. = Three-State

Table 4. 80960KA Pin Description: Support Signals

Name	Type	Description
$\overline{BADAC}$	I	**BAD ACCESS**, if asserted in the cycle following the one in which the last $\overline{READY}$ of a transaction is asserted, indicates an unrecoverable error occurred on the current bus transaction or a synchronous load/store instruction has not been acknowledged. During system reset the $\overline{BADAC}$ signal is interpreted differently. If the signal is high, it indicates that this processor will perform system initialization. If it is low, another processor in the system will perform system initialization instead.
RESET	I	**RESET** clears the processor's internal logic and causes it to reinitialize. During RESET assertion, the input pins are ignored (except for $\overline{BADAC}$ and $\overline{IAC}/\overline{INT_0}$), the three-state output pins are placed in a high impedance state and other output pins are placed in their non-asserted states. RESET must be asserted for at least 41 CLK2 cycles for a predictable RESET. The HIGH to LOW transition of RESET should occur after the rising edge of both CLK2 and the external bus clock and before the next rising edge of CLK2.

I/O = Input/Output, O = Output, I = Input, O.D. = Open Drain, T.S. = Three-State

Table 4. 80960KA Pin Description: Support Signals (Continued)

Name	Type	Description
$\overline{\text{FAILURE}}$	O O.D.	**INITIALIZATION FAILURE** indicates that the processor did not initialize correctly. After RESET deasserts and before the first bus transaction begins, $\overline{\text{FAILURE}}$ asserts while the processor performs a self-test. If the self-test completes successfully, then $\overline{\text{FAILURE}}$ deasserts. The processor then performs a zero checksum on the first eight words of memory. If it fails, $\overline{\text{FAILURE}}$ asserts for a second time and remains asserted. If it passes, system initialization continues and $\overline{\text{FAILURE}}$ remains deasserted.
$\overline{\text{IAC}}/\overline{\text{INT}}_0$	I	**INTERAGENT COMMUNICATION REQUEST/INTERRUPT 0** indicates an IAC message or an interrupt is pending. The bus interrupt control register determines how the signal is interpreted. To signal an interrupt or IAC request in a synchronous system, this pin—as well as the other interrupt pins—must be enabled by being deasserted for at least one bus cycle and then asserted for at least one additional bus cycle. In an asynchronous system the pin must remain deasserted for at least two bus cycles and then asserted for at least two more bus cycles. During system reset, this signal must be in the logic high condition to enable normal processor operation. The logic low condition is reserved.
INT_1	I	**INTERRUPT 1,** like $\overline{\text{INT}}_0$, provides direct interrupt signaling.
INT_2/INTR	I	**INTERRUPT 2/INTERRUPT REQUEST:** The interrupt control register determines how this pin is interpreted. If INT_2, it has the same interpretation as the $\overline{\text{INT}}_0$ and INT_1 pins. If INTR, it is used to receive an interrupt request from an external interrupt controller.
$\overline{\text{INT}}_3/\overline{\text{INTA}}$	I/O O.D.	**INTERRUPT 3/INTERRUPT ACKNOWLEDGE:** The bus interrupt control register determines how this pin is interpreted. If $\overline{\text{INT}}_3$, it has the same interpretation as the $\overline{\text{INT}}_0$, INT_1 and INT_2 pins. If $\overline{\text{INTA}}$, it is used as an output to control interrupt-acknowledge transactions. The $\overline{\text{INTA}}$ output is latched on-chip and remains valid during T_d cycles; as an output, it is open-drain.
N.C.	N/A	**NOT CONNECTED** indicates pins should not be connected. Never connect any pin marked N.C. as these pins may be reserved for factory use.

I/O = Input/Output, O = Output, I = Input, O.D. = Open Drain, T.S. = Three-State

2.0 ELECTRICAL SPECIFICATIONS

2.1 Power and Grounding

The 80960KA is implemented in CHMOS IV technology and therefore has modest power requirements. Its high clock frequency and numerous output buffers (address/data, control, error and arbitration signals) can cause power surges as multiple output buffers simultaneously drive new signal levels. For clean on-chip power distribution, V_{CC} and V_{SS} pins separately feed the device's functional units. Power and ground connections must be made to all 80960KA power and ground pins. On the circuit board, all V_{CC} pins must be strapped closely together, preferably on a power plane; all V_{SS} pins should be strapped together, preferably on a ground plane.

2.2 Power Decoupling Recommendations

Place a liberal amount of decoupling capacitance near the 80960KA. When driving the L-bus the processor can cause transient power surges, particularly when connected to a large capacitive load.

Low inductance capacitors and interconnects are recommended for best high frequency electrical performance. Inductance is reduced by shortening board traces between the processor and decoupling capacitors as much as possible.

2.3 Connection Recommendations

For reliable operation, always connect unused inputs to an appropriate signal level. In particular, if one or more interrupt lines are not used, they should be pulled up. No inputs should ever be left floating.

All open-drain outputs require a pullup device. While in most cases a simple pullup resistor is adequate, a network of pullup and pulldown resistors biased to a valid V_{IH} (>3.0V) and terminated in the characteristic impedance of the circuit board is recommended to limit noise and AC power consumption. Figure 5 and Figure 6 show recommended values for the resistor network for low and high current drive, assuming a characteristic impedance of 100Ω. Terminating output signals in this fashion limits signal swing and reduces AC power consumption.

NOTE:
Do not connect external logic to pins marked N.C.

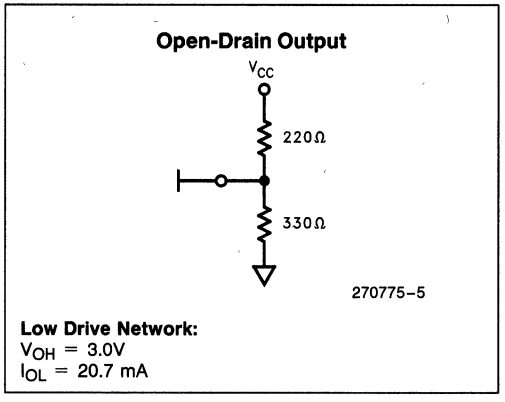


Figure 5. Connection Recommendations for Low Current Drive Network

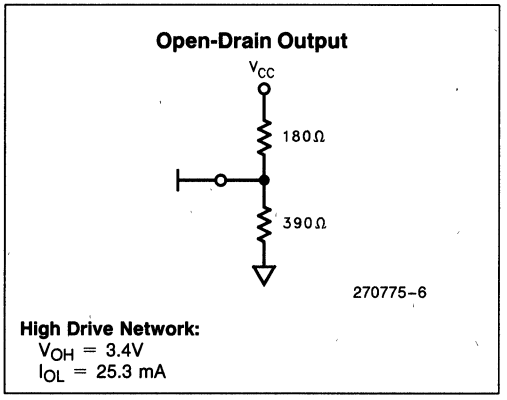


Figure 6. Connection Recommendations for High Current Drive Network

2.4 Characteristic Curves

Figure 7 shows typical supply current requirements over the operating temperature range of the processor at supply voltage (V_{CC}) of 5V. Figure 8 and Figure 9 show the typical power supply current (I_{CC}) that the 80960KA requires at various operating frequencies when measured at three input voltage (V_{CC}) levels and two temperatures.

For a given output current (I_{OL}) the curve in Figure 10 shows the worst case output low voltage (V_{OL}). Figure 11 shows the typical capacitive derating curve for the 80960KA measured from 1.5V on the system clock (CLK) to 1.5V on the falling edge and 1.5V on the rising edge of the L-Bus address/data (LAD) signals.

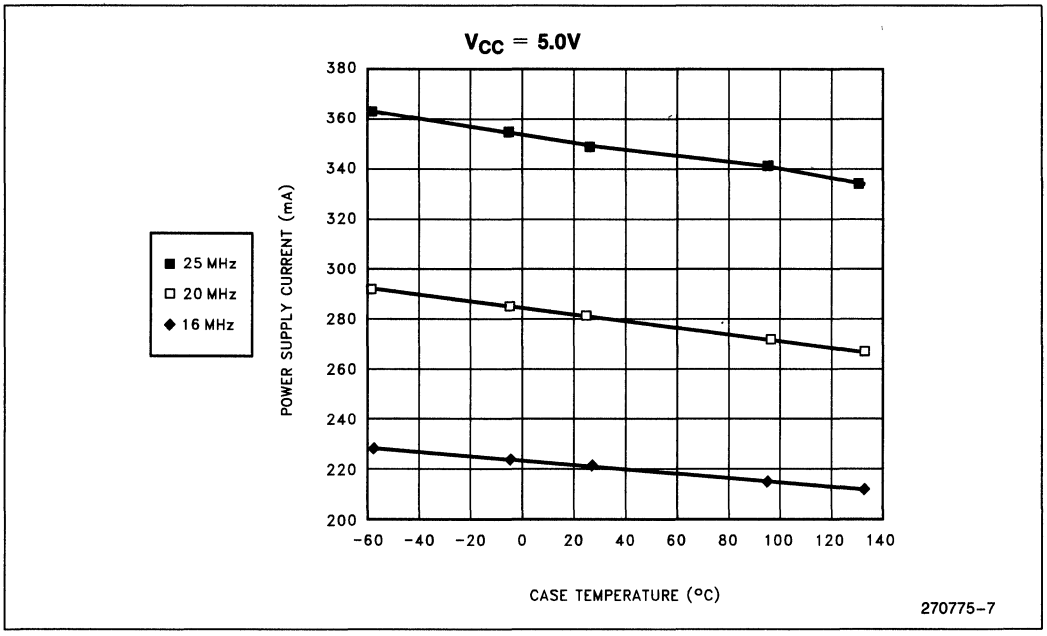


Figure 7. Typical Supply Current vs. Case Temperature

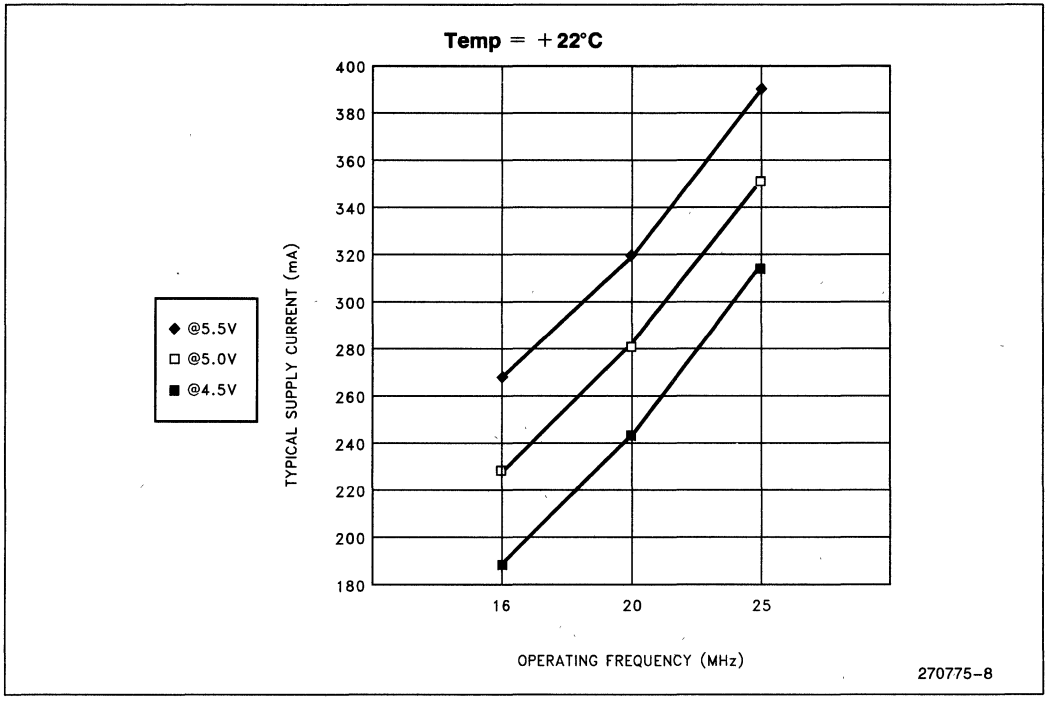


Figure 8. Typical Current vs. Frequency (Room Temp)

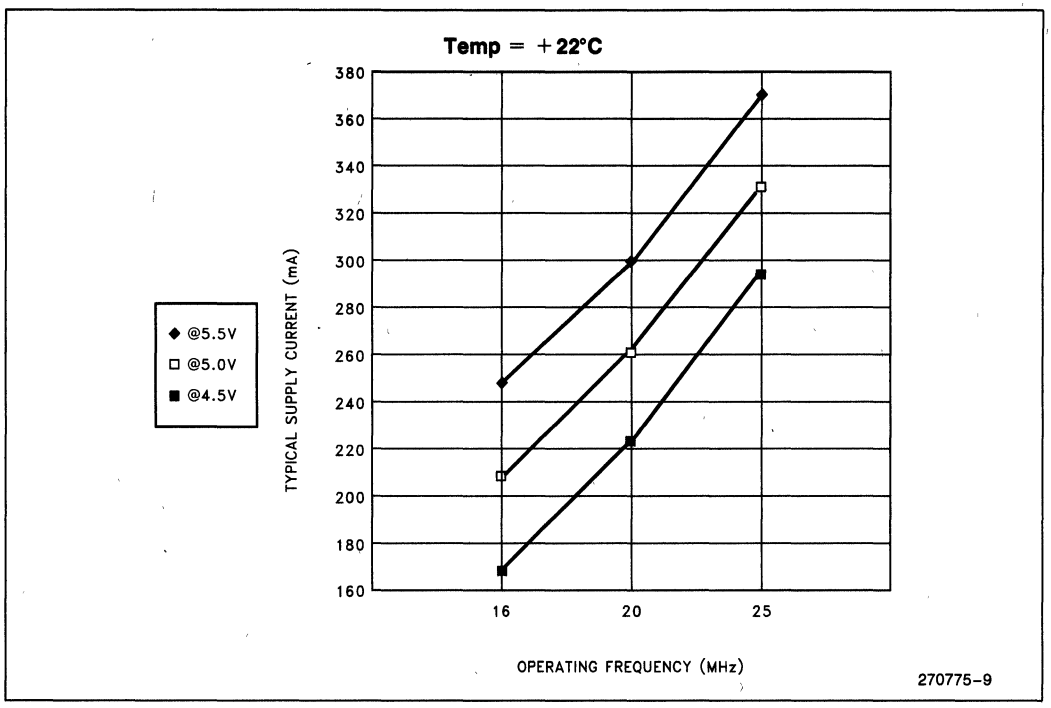


Figure 9. Typical Current vs. Frequency (Hot Temp)

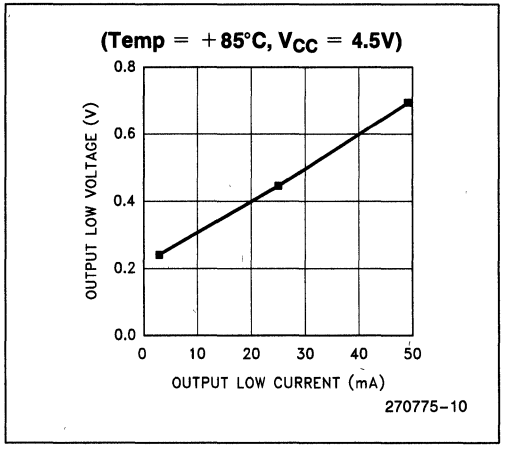


Figure 10. Worst-Case Voltage vs. Output Current on Open-Drain Pins

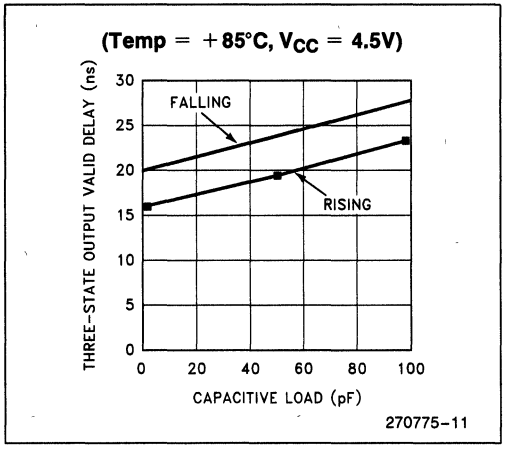
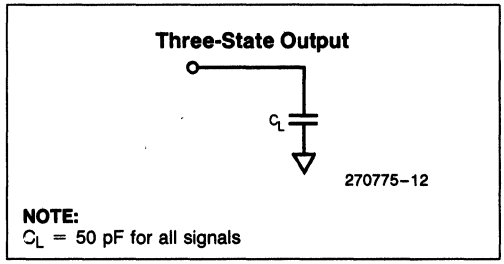


Figure 11. Capacitive Derating Curve

2.5 Test Load Circuit

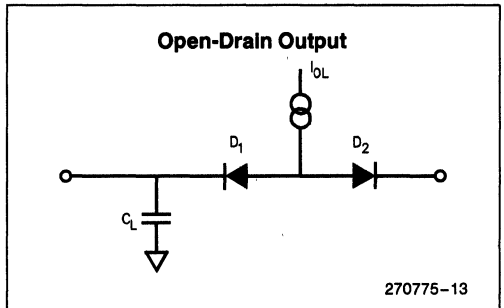
Figure 12 illustrates the load circuit used to test the 80960KA's three-state pins; Figure 13 shows the load circuit used to test the open drain outputs. The open drain test uses an active load circuit in the form of a matched diode bridge. Since the open-drain outputs sink current, only the I_{OL} legs of the bridge are necessary and the I_{OH} legs are not used. When the 80960KA driver under test is turned off, the output pin is pulled up to V_{REF} (i.e., V_{OH}). Diode D_1 is turned off and the I_{OL} current source flows through diode D_2.

When the 80960KA open-drain driver under test is on, diode D_1 is also on and the voltage on the pin being tested drops to V_{OL}. Diode D_2 turns off and I_{OL} flows through diode D_1.



NOTE:
C_L = 50 pF for all signals

Figure 12. Test Load Circuit for Three-State Output Pins



NOTES:
I_{OL} Tested at 25 mA
$V_{REF} = V_{CC}$
D_1 and D_2 are matched
C_L = 50 pF for all signals

Figure 13. Test Load Circuit for Open-Drain Output Pins

2.6 Absolute Maximum Ratings

Operating Temperature
PGA .....0°C to +85°C Case
PQFP .....0°C to +100°C Case
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin.....−0.5V to V_{CC} + 0.5V
Power Dissipation .....2.5W (25 MHz)

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

2.7 DC Characteristics

PGA: 80960KA (16 MHz) T_{CASE} = 0°C to +85°C, V_{CC} = 5V ±10%
80960KA (20 and 25 MHz) T_{CASE} = 0°C to +85°C, V_{CC} = 5V ±5%
PQFP: 80960KA (16 MHz) T_{CASE} = 0°C to +100°C, V_{CC} = 5V ±10%
80960KA (20 and 25 MHz) T_{CASE} = 0°C to +100°C, V_{CC} = 5V ±5%

Table 5. DC Characteristics

Symbol	Parameter	Min	Max	Units	Notes
V_{IL}	Input Low Voltage	−0.3	+0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.3	V	
V_{CL}	CLK2 Input Low Voltage	−0.3	+0.8	V	
V_{CH}	CLK2 Input High Voltage	0.55 V_{CC}	V_{CC} + 0.3	V	
V_{OL}	Output Low Voltage		0.45	V	(1, 2)
V_{OH}	Output High Voltage	2.4		V	(3, 4)
I_{CC}	Power Supply Current: 16 MHz 20 MHz 25 MHz		 315 360 420	 mA mA mA	 (5) (5) (5)
I_{LI}	Input Leakage Current		±15	µA	$0 \le V_{IN} \le V_{CC}$
I_{LO}	Output Leakage Current		±15	µA	$0.45 \le V_O \le V_{CC}$
C_{IN}	Input Capacitance		10	pF	f_C = 1 MHz(6)
C_O	Output Capacitance		12	pF	f_C = 1 MHz(6)
C_{CLK}	Clock Capacitance		10	pF	f_C = 1 MHz(6)

NOTES:
1. For three-state outputs, this parameter is measured at:
   Addresss/Data.....4.0 mA
   Controls.....5.0 mA
2. For open-drain outputs .....25 mA
3. This parameter is measured at:
   Address/Data.....−1.0 mA
   Controls.....−0.9 mA
   $\overline{ALE}$ .....−5.0 mA
4. Not measured on open-drain outputs.
5. Measured at worst case frequency, V_{CC} and temperature, with device operating and outputs loaded to the test conditions in Figures 12 and 13. Figure 7, Figure 8 and Figure 9 indicate typical values.
6. Input, output and clock capacitance are not tested.

2.8 AC Specifications

This section describes the AC specifications for the 80960KA pins. All input and output timings are specified relative to the 1.5V level of the rising edge of CLK2. For output timings the specifications refer to the time it takes the signal to reach 1.5V.

For input timings the specifications refer to the time at which the signal reaches (for input setup) or leaves (for hold time) the TTL levels of LOW (0.8V) or HIGH (2.0V). All AC testing should be done with input voltages of 0.4V and 2.4V, except for the clock (CLK2), which should be tested with input voltages of 0.45V and 0.55 V_{CC}.

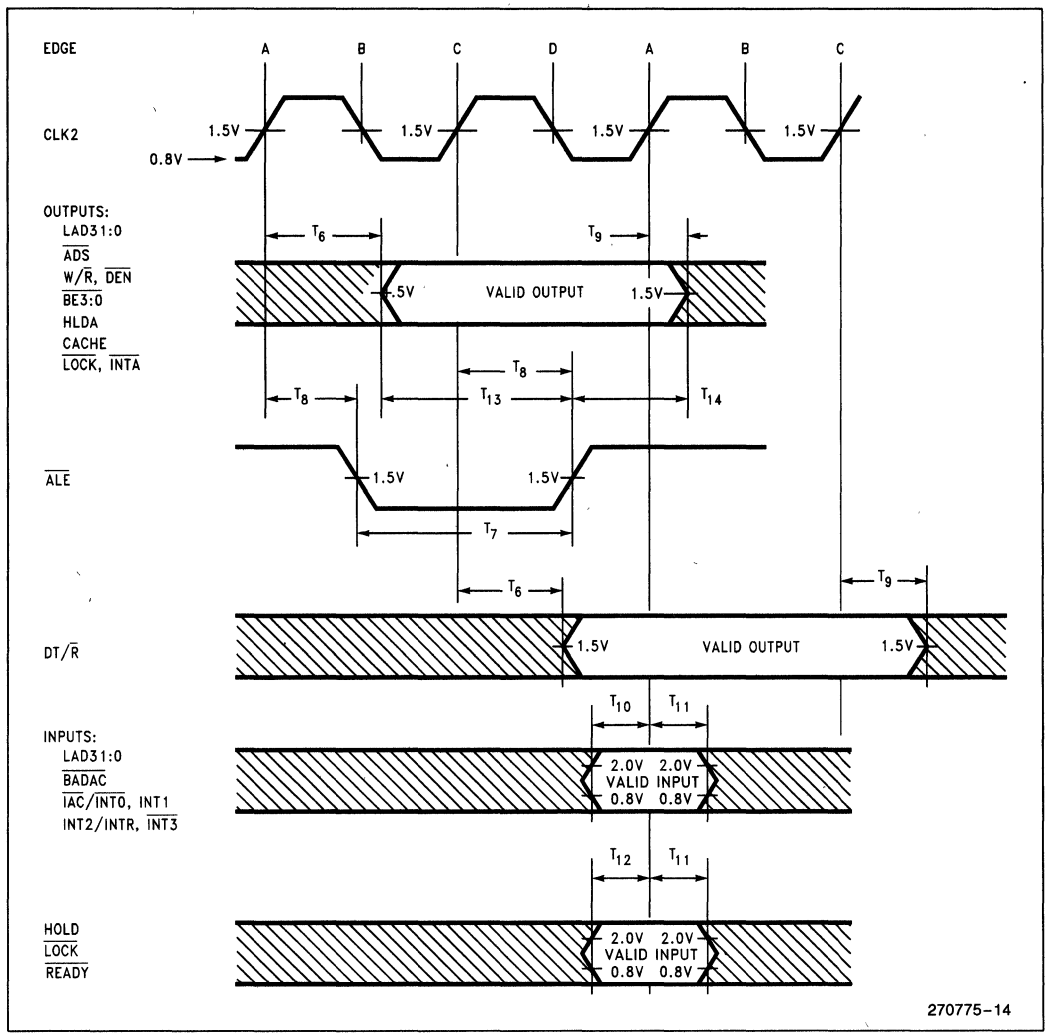


Figure 14. Drive Levels and Timing Relationships for 80960KA Signals

2.8.1 AC SPECIFICATION TABLES

Table 6. 80960KA AC Characteristics (16 MHz)

Symbol	Parameter	Min	Max	Units	Notes
INPUT CLOCK					
T_1	Processor Clock Period (CLK2)	31.25	125	ns	V_{IN} = 1.5V
T_2	Processor Clock Low Time (CLK2)	8		ns	V_{IL} = 10% Point = 1.2V
T_3	Processor Clock High Time (CLK2)	8		ns	V_{IH} = 90% Point = 0.1V + 0.5 V_{CC}
T_4	Processor Clock Fall Time (CLK2)		10	ns	V_{IN} = 90% Point to 10% Point(1)
T_5	Processor Clock Rise Time (CLK2)		10	ns	V_{IN} = 10% Point to 90% Point(1)
SYNCHRONOUS OUTPUTS					
T_6	Output Valid Delay	2	25	ns	
T_{6H}	HLDA Output Valid Delay	4	28	ns	
T_7	$\overline{ALE}$ Width	15		ns	
T_8	$\overline{ALE}$ Output Valid Delay	2	18	ns	
T_9	Output Float Delay	2	20	ns	(2)
T_{9H}	HLDA Output Float Delay	4	20	ns	(2)
SYNCHRONOUS INPUTS					
T_{10}	Input Setup 1	3		ns	(3)
T_{11}	Input Hold	5		ns	(3)
T_{11H}	HOLD Input Hold	4		ns	(3)
T_{12}	Input Setup 2	8		ns	(3)
T_{13}	Setup to $\overline{ALE}$ Inactive	10		ns	
T_{14}	Hold after $\overline{ALE}$ Inactive	8		ns	
T_{15}	Reset Hold	3		ns	(3)
T_{16}	Reset Setup	5		ns	(3)
T_{17}	Reset Width	1281		ns	41 CLK2 Periods Minimum

NOTES:
1. Clock rise and tall times are not tested.
2. A float condition occurs when the maximum output current becomes less than I_{LO}. Float delay is not tested; however, it should not be longer than the valid delay.
3. LAD31:0, $\overline{BADAC}$, HOLD, $\overline{LOCK}$ and $\overline{READY}$ are synchronous inputs. $\overline{IAC}/\overline{INT_0}$, INT_1, INT_2/INT_R and $\overline{INT_3}$ may be synchronous or asynchronous.

Table 7. 80960KA AC Characteristics (20 MHz)

Symbol	Parameter	Min	Max	Units	Notes
INPUT CLOCK					
T_1	Processor Clock Period (CLK2)	25	125	ns	V_{IN} = 1.5V
T_2	Processor Clock Low Time (CLK2)	6		ns	V_{IL} = 10% Point = 1.2V
T_3	Processor Clock High Time (CLK2)	6		ns	V_{IH} = 90% Point = 0.1V + 0.5 V_{CC}
T_4	Processor Clock Fall Time (CLK2)		10	ns	V_{IN} = 90% Point to 10% Point(1)
T_5	Processor Clock Rise Time (CLK2)		10	ns	V_{IN} = 10% Point to 90% Point(1)
SYNCHRONOUS OUTPUTS					
T_6	Output Valid Delay	2	20	ns	
T_{6H}	HLDA Output Valid Delay	4	23	ns	
T_7	$\overline{ALE}$ Width	12		ns	
T_8	$\overline{ALE}$ Output Valid Delay	2	18	ns	
T_9	Output Float Delay	2	20	ns	(2)
T_{9H}	HLDA Output Float Delay	4	20	ns	(2)
SYNCHRONOUS INPUTS					
T_{10}	Input Setup 1	3		ns	(3)
T_{11}	Input Hold	5		ns	(3)
T_{11H}	HOLD Input Hold	4		ns	(3)
T_{12}	Input Setup 2	7		ns	(3)
T_{13}	Setup to $\overline{ALE}$ Inactive	10		ns	
T_{14}	Hold after $\overline{ALE}$ Inactive	8		ns	
T_{15}	Reset Hold	3		ns	
T_{15}	Reset Setup	5		ns	
T_{17}	Reset Width	1025		ns	41 CLK2 Periods Minimum

NOTES:
1. Clock rise and fall times are not tested.
2. A float condition occurs when the maximum output current becomes less than I_{LO}. Float delay is not tested; however, it should not be longer than the valid delay.
3. LAD31:0, $\overline{BADAC}$, HOLD, $\overline{LOCK}$ and $\overline{READY}$ are synchronous inputs. $\overline{IAC}/\overline{INT_0}$, INT_1, INT_2/INT_R and $\overline{INT_3}$ may be synchronous or asynchronous.

Table 8. 80960KA AC Characteristics (25 MHz)

Symbol	Parameter	Min	Max	Units	Notes
INPUT CLOCK					
T_1	Processor Clock Period (CLK2)	20	125	ns	V_{IN} = 1.5V
T_2	Processor Clock Low Time (CLK2)	5		ns	V_{IL} = 10% Point = 1.2V
T_3	Processor Clock High Time (CLK2)	5		ns	V_{IH} = 90% Point = 0.1V + 0.5 V_{CC}
T_4	Processor Clock Fall Time (CLK2)		10	ns	V_{IN} = 90% Point to 10% Point(1)
T_5	Processor Clock Rise Time (CLK2)		10	ns	V_{IN} = 10% Point to 90% Point(1)
SYNCHRONOUS OUTPUTS					
T_6	Output Valid Delay	2	18	ns	
T_{6H}	HLDA Output Valid Delay	4	23	ns	
T_7	$\overline{ALE}$ Width	12		ns	
T_8	$\overline{ALE}$ Output Valid Delay	2	18	ns	
T_9	Output Float Delay	2	18	ns	(2)
T_{9H}	HLDA Output Float Delay	4	20	ns	(2)
SYNCHRONOUS INPUTS					
T_{10}	Input Setup 1	3		ns	(3)
T_{11}	Input Hold	5		ns	(3)
T_{11H}	HOLD Input Hold	4		ns	
T_{12}	Input Setup 2	7		ns	
T_{13}	Setup to $\overline{ALE}$ Inactive	8		ns	
T_{14}	Hold after $\overline{ALE}$ Inactive	8		ns	
T_{15}	Reset Hold	3		ns	
T_{16}	Reset Setup	5		ns	
T_{17}	Reset Width	820		ns	41 CLK2 Periods Minimum

NOTES:
1. Clock rise and tall times are not tested.
2. A float condition occurs when the maximum output current becomes less than I_{LO}. Float delay is not tested; however, it should not be longer than the valid delay.
3. LAD31:0, $\overline{BADAC}$, HOLD, $\overline{LOCK}$ and $\overline{READY}$ are synchronous inputs. $\overline{IAC}/\overline{INT_0}$, INT_1, INT_2/INT_R and $\overline{INT_3}$ may be synchronous or asynchronous.

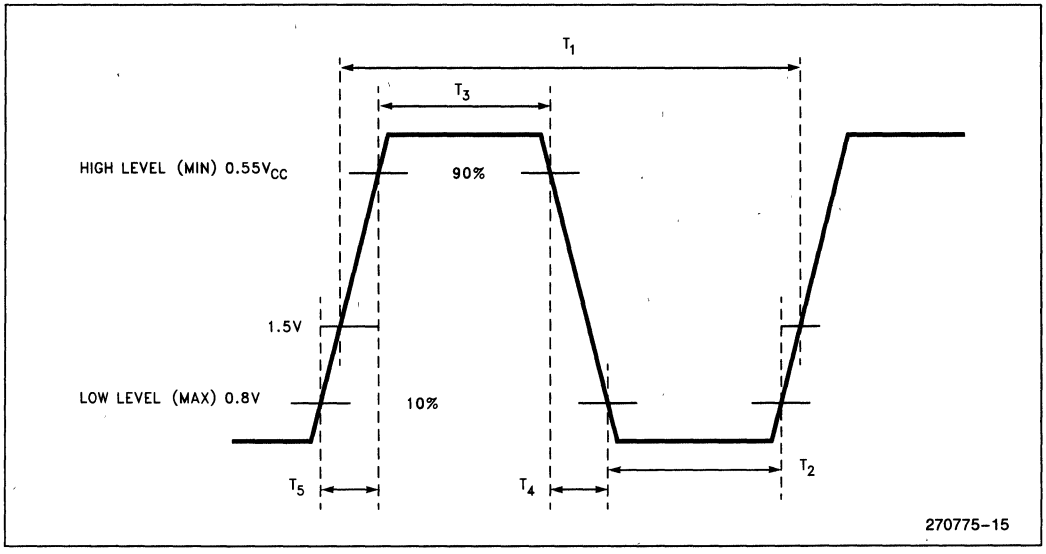
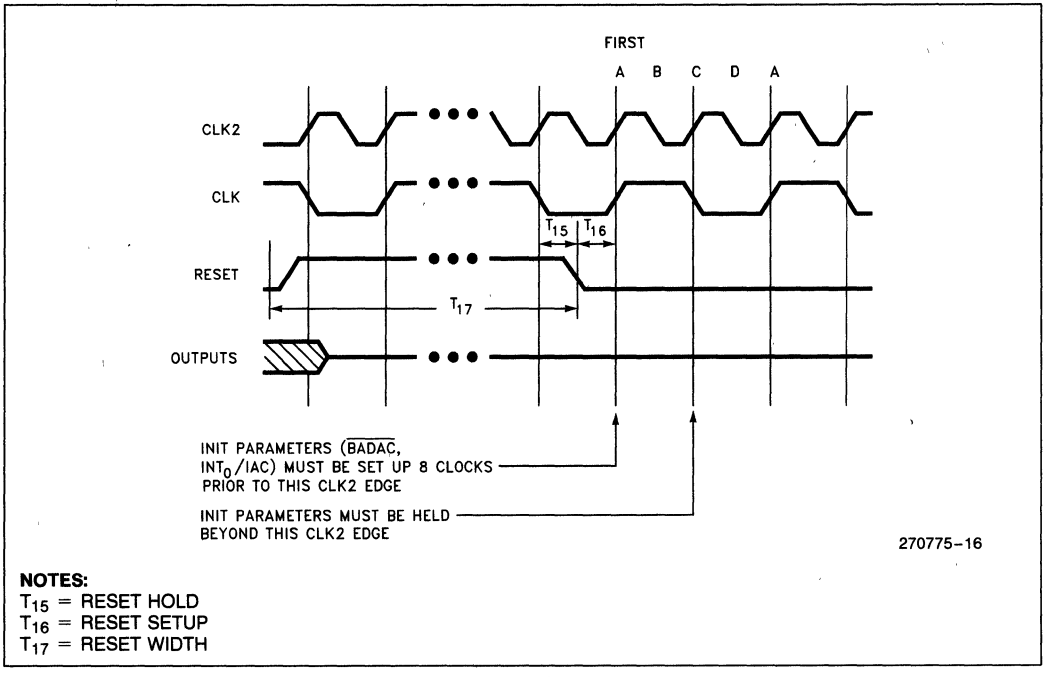


Figure 15. Processor Clock Pulse (CLK2)



NOTES:
T_{15} = RESET HOLD
T_{16} = RESET SETUP
T_{17} = RESET WIDTH

Figure 16. RESET Signal Timing

3.0 MECHANICAL DATA

3.1 Packaging

The 80960KA is available in two package types:

- 132-lead ceramic pin-grid array (PGA). Pins are arranged 0.100 inch (2.54 mm) center-to-center, in a 14 by 14 matrix, three rows around (see Figure 17).
- 132-lead plastic quad flat pack (PQFP). This package uses fine-pitch gull wing leads arranged in a single row along the package perimeter with 0.025 inch (0.64 mm) spacing (see Figure 20).

Dimensions for both package types are given in the Intel *Packaging* handbook (Order #240800).

3.1.1 PIN ASSIGNMENT

The PGA and PQFP have different pin assignments. Figure 18 shows the view from the PGA bottom (pins facing up) and Figure 19 shows a view from the PGA top (pins facing down). Figure 20 shows the PQFP package; Figure 21 shows the PQFP pinout with signal names. Notice that the pins are numbered in order from 1 to 132 around the package perimeter. Table 9 and Table 10 list the function of each PGA pin; Table 11 and Table 12 list the function of each PQFP pin.

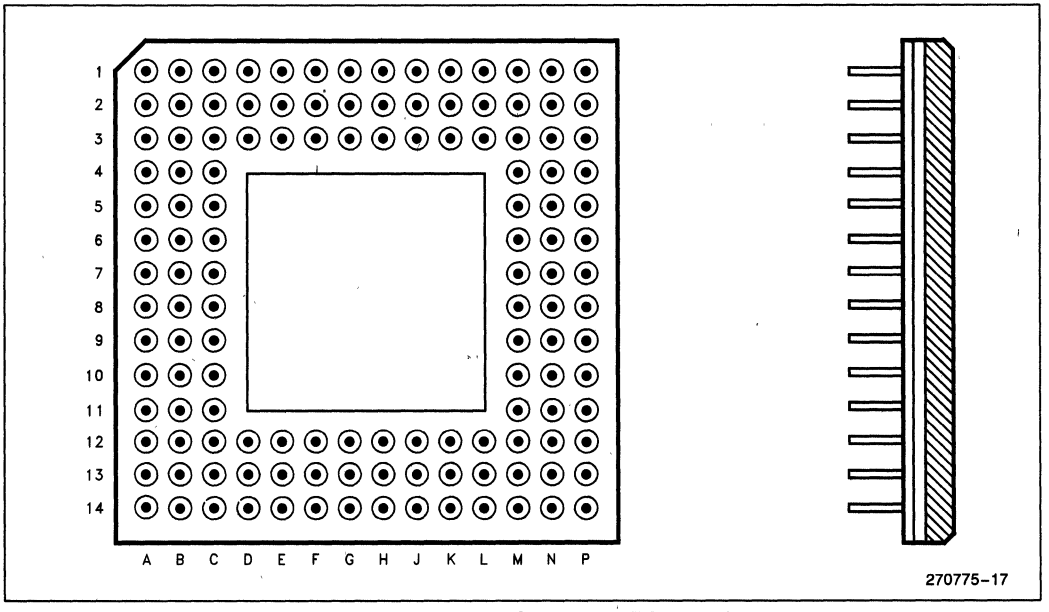


Figure 17. 132-Lead Pin-Grid Array (PGA) Package

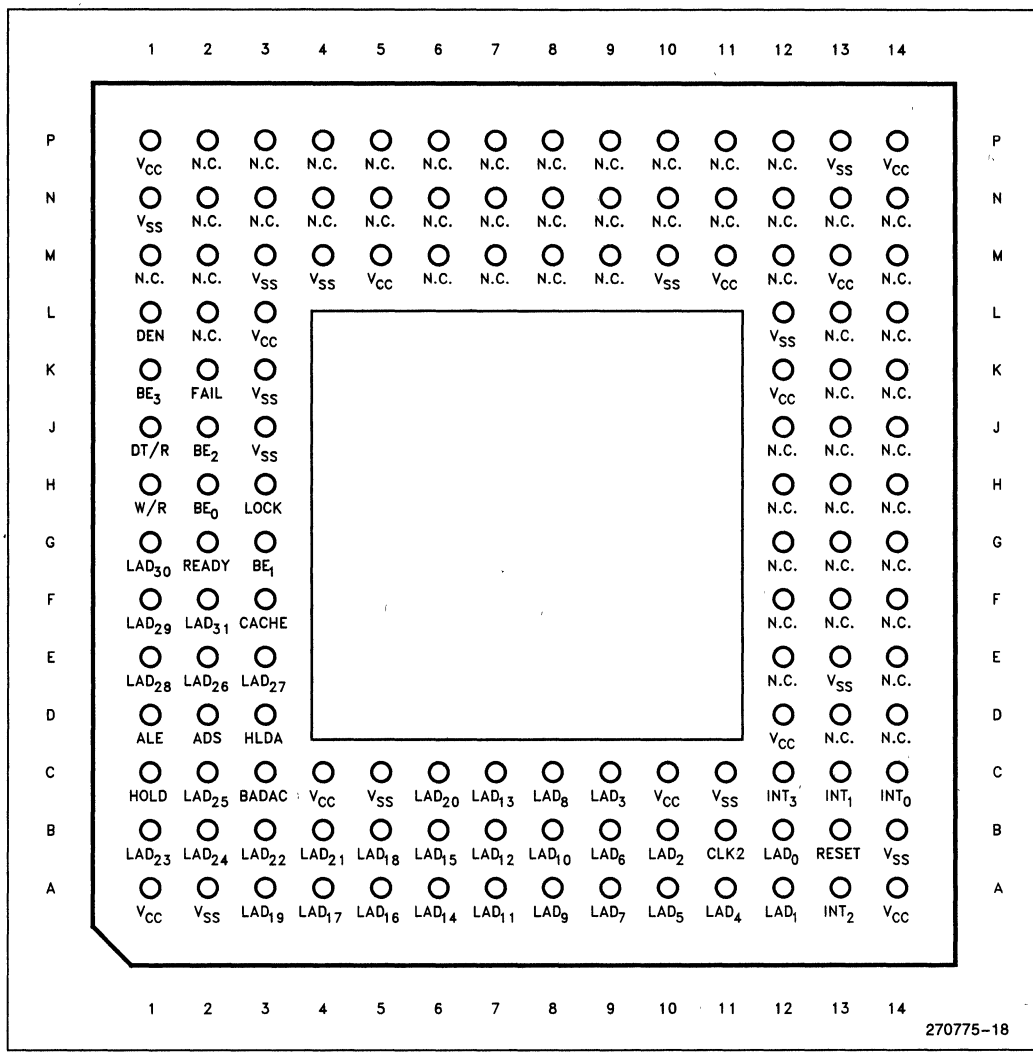


Figure 18. 80960KA PGA Pinout—View from Bottom (Pins Facing Up)

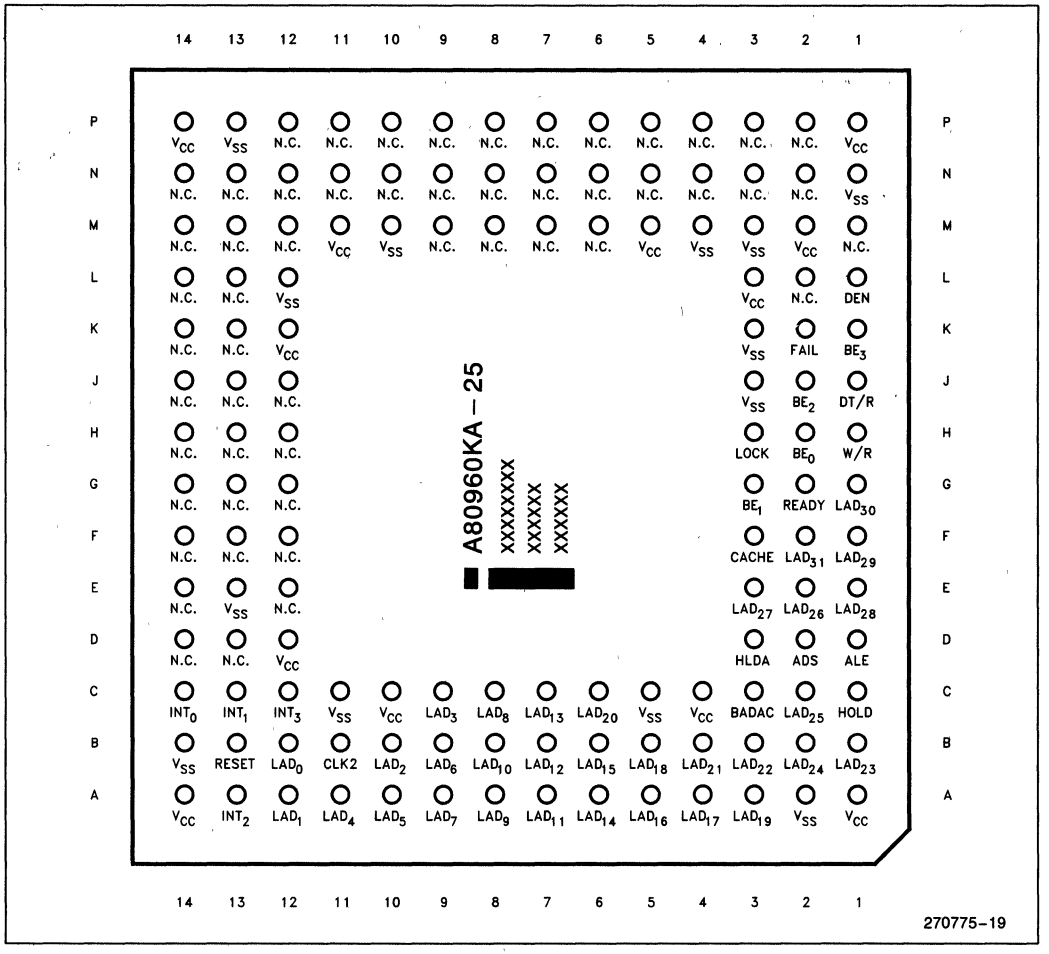


Figure 19. 80960KA PGA Pinout—View from Top (Pins Facing Down)

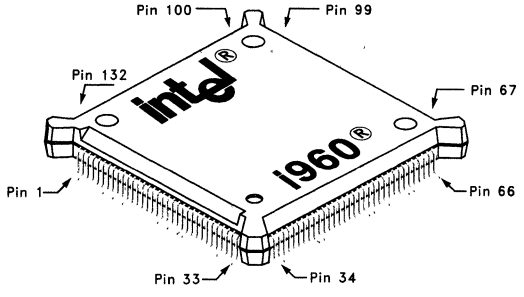


Figure 20. 80960KA 132-Lead Plastic Quad Flat-Pack (PQFP) Package

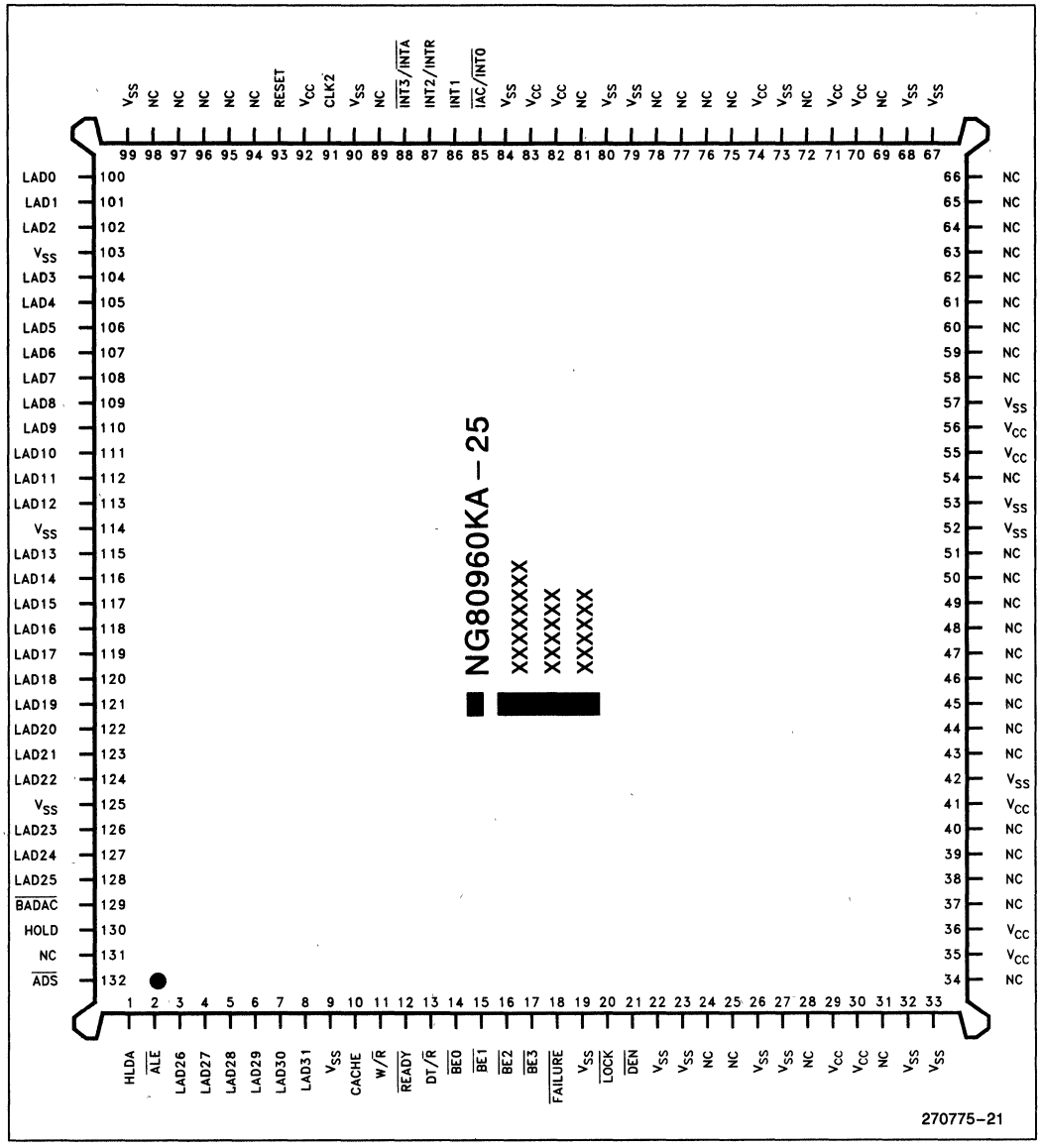


Figure 21. PQFP Pinout—View from Top

3.2 Pinout

Table 9. 80960KA PGA Pinout—In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	V_{CC}	C6	LAD_{20}	H1	$W/\overline{R}$	M10	V_{SS}
A2	V_{SS}	C7	LAD_{13}	H2	$\overline{BE_0}$	M11	V_{CC}
A3	LAD_{19}	C8	LAD_8	H3	$\overline{LOCK}$	M12	N.C.
A4	LAD_{17}	C9	LAD_3	H12	N.C.	M13	N.C.
A5	LAD_{16}	C10	V_{CC}	H13	N.C.	M14	N.C.
A6	LAD_{14}	C11	V_{SS}	H14	N.C.	N1	V_{SS}
A7	LAD_{11}	C12	$\overline{INT_3}/\overline{INTA}$	J1	$DT/\overline{R}$	N2	N.C.
A8	LAD_9	C13	INT_1	J2	$\overline{BE_2}$	N3	N.C.
A9	LAD_7	C14	$\overline{IAC}/\overline{INT_0}$	J3	V_{SS}	N4	N.C.
A10	LAD_5	D1	$\overline{ALE}$	J12	N.C.	N5	N.C.
A11	LAD_4	D2	$\overline{ADS}$	J13	N.C.	N6	N.C.
A12	LAD_1	D3	HLDA	J14	N.C.	N7	N.C.
A13	INT_2/INTR	D12	V_{CC}	K1	$\overline{BE_3}$	N8	N.C.
A14	V_{CC}	D13	N.C.	K2	$\overline{FAILURE}$	N9	N.C.
B1	LAD_{23}	D14	N.C.	K3	V_{SS}	N10	N.C.
B2	LAD_{24}	E1	LAD_{28}	K12	V_{CC}	N11	N.C.
B3	LAD_{22}	E2	LAD_{26}	K13	N.C.	N12	N.C.
B4	LAD_{21}	E3	LAD_{27}	K14	N.C.	N13	N.C.
B5	LAD_{18}	E12	N.C.	L1	$\overline{DEN}$	N14	N.C.
B6	LAD_{15}	E13	V_{SS}	L2	N.C.	P1	V_{CC}
B7	LAD_{12}	E14	N.C.	L3	V_{CC}	P2	N.C.
B8	LAD_{10}	F1	LAD_{29}	L12	V_{SS}	P3	N.C.
B9	LAD_6	F2	LAD_{31}	L13	N.C.	P4	N.C.
B10	LAD_2	F3	CACHE	L14	N.C.	P5	N.C.
B11	CLK2	F12	N.C.	M1	N.C.	P6	N.C.
B12	LAD_0	F13	N.C.	M2	V_{CC}	P7	N.C.
B13	RESET	F14	N.C.	M3	V_{SS}	P8	N.C.
B14	V_{SS}	G1	LAD_{30}	M4	V_{SS}	P9	N.C.
C1	HOLD	G2	$\overline{READY}$	M5	V_{CC}	P10	N.C.
C2	LAD_{25}	G3	$\overline{BE_1}$	M6	N.C.	P11	N.C.
C3	$\overline{BADAC}$	G12	N.C.	M7	N.C.	P12	N.C.
C4	V_{CC}	G13	N.C.	M8	N.C.	P13	V_{SS}
C5	V_{SS}	G14	N.C.	M9	N.C.	P14	V_{CC}

NOTE:
Do not connect any external logic to any pins marked N.C.

Table 10. 80960KA PGA Pinout—In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
$\overline{ADS}$	D2	LAD_{15}	B6	N.C.	J14	N.C.	P9
$\overline{ALE}$	D1	LAD_{16}	A5	N.C.	K13	N.C.	P10
$\overline{BADAC}$	C3	LAD_{17}	A4	N.C.	K14	N.C.	P11
$\overline{BE}_0$	H2	LAD_{18}	B5	N.C.	L13	N.C.	P12
$\overline{BE}_1$	G3	LAD_{19}	A3	N.C.	L14	N.C.	L2
$\overline{BE}_2$	J2	LAD_{20}	C6	N.C.	M1	$\overline{READY}$	G2
$\overline{BE}_3$	K1	LAD_{21}	B4	N.C.	M6	RESET	B13
CACHE	F3	LAD_{22}	B3	N.C.	M7	V_{CC}	A1
CLK2	B11	LAD_{23}	B1	N.C.	M8	V_{CC}	A14
$\overline{DEN}$	L1	LAD_{24}	B2	N.C.	M9	V_{CC}	C4
DT/$\overline{R}$	J1	LAD_{25}	C2	N.C.	M12	V_{CC}	C10
$\overline{FAILURE}$	K2	LAD_{26}	E2	N.C.	M13	V_{CC}	D12
HLDA	D3	LAD_{27}	E3	N.C.	M14	V_{CC}	K12
HOLD	C1	LAD_{28}	E1	N.C.	N2	V_{CC}	L3
$\overline{IAC}/\overline{INT}_0$	C14	LAD_{29}	F1	N.C.	N3	V_{CC}	M2
INT_1	C13	LAD_{30}	G1	N.C.	N4	V_{CC}	M5
INT_2/INTR	A13	LAD_{31}	F2	N.C.	N5	V_{CC}	M11
$\overline{INT}_3/\overline{INTA}$	C12	$\overline{LOCK}$	H3	N.C.	N6	V_{CC}	P1
LAD_0	B12	N.C.	D13	N.C.	N7	V_{CC}	P14
LAD_1	A12	N.C.	D14	N.C.	N8	V_{SS}	A2
LAD_2	B10	N.C.	E12	N.C.	N9	V_{SS}	B14
LAD_3	C9	N.C.	E14	N.C.	N10	V_{SS}	C5
LAD_4	A11	N.C.	F12	N.C.	N11	V_{SS}	C11
LAD_5	A10	N.C.	F13	N.C.	N12	V_{SS}	E11
LAD_6	B9	N.C.	F14	N.C.	N13	V_{SS}	J3
LAD_7	A9	N.C.	G12	N.C.	N14	V_{SS}	K3
LAD_8	C8	N.C.	G13	N.C.	P2	V_{SS}	L12
LAD_9	A8	N.C.	G14	N.C.	P3	V_{SS}	M3
LAD_{10}	B8	N.C.	H12	N.C.	P4	V_{SS}	M4
LAD_{11}	A7	N.C.	H13	N.C.	P5	V_{SS}	M10
LAD_{12}	B7	N.C.	H14	N.C.	P6	V_{SS}	N1
LAD_{13}	C7	N.C.	J12	N.C.	P7	V_{SS}	P13
LAD_{14}	A6	N.C.	J13	N.C.	P8	W/$\overline{R}$	H1

NOTE:
Do not connect any external logic to any pins marked N.C.

Table 11. 80960KA PQFP Pinout—In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	HLDA	34	N.C.	67	V_{SS}	100	LAD_0
2	$\overline{ALE}$	35	V_{CC}	68	V_{SS}	101	LAD_1
3	LAD_{26}	36	V_{CC}	69	N.C.	102	LAD_2
4	LAD_{27}	37	N.C.	70	V_{CC}	103	V_{SS}
5	LAD_{28}	38	N.C.	71	V_{CC}	104	LAD_3
6	LAD_{29}	39	N.C.	72	N.C.	105	LAD_4
7	LAD_{30}	40	N.C.	73	V_{SS}	106	LAD_5
8	LAD_{31}	41	V_{CC}	74	V_{CC}	107	LAD_6
9	V_{SS}	42	V_{SS}	75	N.C.	108	LAD_7
10	CACHE	43	N.C.	76	N.C.	109	LAD_8
11	W/$\overline{R}$	44	N.C.	77	N.C.	110	LAD_9
12	$\overline{READY}$	45	N.C.	78	N.C.	111	LAD_{10}
13	DT/$\overline{R}$	46	N.C.	79	V_{SS}	112	LAD_{11}
14	$\overline{BE_0}$	47	N.C.	80	V_{SS}	113	LAD_{12}
15	$\overline{BE_1}$	48	N.C.	81	N.C.	114	V_{SS}
16	$\overline{BE_2}$	49	N.C.	82	V_{CC}	115	LAD_{13}
17	$\overline{BE_3}$	50	N.C.	83	V_{CC}	116	LAD_{14}
18	$\overline{FAILURE}$	51	N.C.	84	V_{SS}	117	LAD_{15}
19	V_{SS}	52	V_{SS}	85	$\overline{IAC}/\overline{INT_0}$	118	LAD_{16}
20	$\overline{LOCK}$	53	V_{SS}	86	INT_1	119	LAD_{17}
21	$\overline{DEN}$	54	N.C.	87	INT_2/INTR	120	LAD_{18}
22	V_{SS}	55	V_{CC}	88	$\overline{INT_3}/\overline{INTA}$	121	LAD_{19}
23	V_{SS}	56	V_{CC}	89	N.C.	122	LAD_{20}
24	N.C.	57	V_{SS}	90	V_{SS}	123	LAD_{21}
25	N.C.	58	N.C.	91	CLK2	124	LAD_{22}
26	V_{SS}	59	N.C.	92	V_{CC}	125	V_{SS}
27	V_{SS}	60	N.C.	93	RESET	126	LAD_{23}
28	N.C.	61	N.C.	94	N.C.	127	LAD_{24}
29	V_{CC}	62	N.C.	95	N.C.	128	LAD_{25}
30	V_{CC}	63	N.C.	96	N.C.	129	$\overline{BADAC}$
31	N.C.	64	N.C.	97	N.C.	130	HOLD
32	V_{SS}	65	N.C.	98	N.C.	131	N.C.
33	V_{SS}	66	N.C.	99	V_{SS}	132	$\overline{ADS}$

NOTE:
Do not connect any external logic to any pins marked N.C.

Table 12. 80960KA PQFP Pinout—In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
$\overline{ADS}$	132	LAD_{15}	117	N.C.	49	V_{CC}	41
$\overline{ALE}$	2	LAD_{16}	118	N.C.	50	V_{CC}	55
$\overline{BADAC}$	129	LAD_{17}	119	N.C.	51	V_{CC}	56
$\overline{BE}_0$	14	LAD_{18}	120	N.C.	54	V_{CC}	70
$\overline{BE}_1$	15	LAD_{19}	121	N.C.	58	V_{CC}	71
$\overline{BE}_2$	16	LAD_{20}	122	N.C.	59	V_{CC}	74
$\overline{BE}_3$	17	LAD_{21}	123	N.C.	60	V_{CC}	82
CACHE	10	LAD_{22}	124	N.C.	61	V_{CC}	83
CLK2	91	LAD_{23}	126	N.C.	62	V_{CC}	92
$\overline{DEN}$	21	LAD_{24}	127	N.C.	63	V_{SS}	9
DT/$\overline{R}$	13	LAD_{25}	128	N.C.	64	V_{SS}	19
$\overline{FAILURE}$	18	LAD_{26}	3	N.C,	65	V_{SS}	22
HLDA	1	LAD_{27}	4	N.C.	66	V_{SS}	23
HOLD	130	LAD_{28}	5	N.C.	69	V_{SS}	26
$\overline{IAC}/\overline{INT}_0$	85	LAD_{29}	6	N.C.	72	V_{SS}	27
INT_1	86	LAD_{30}	7	N.C.	75	V_{SS}	32
INT_2/INTR	87	LAD_{31}	8	N.C.	76	V_{SS}	33
$\overline{INT}_3/\overline{INTA}$	88	$\overline{LOCK}$	20	N.C.	77	V_{SS}	42
LAD_0	100	N.C.	24	N.C.	78	V_{SS}	52
LAD_1	101	N.C.	25	N.C.	81	V_{SS}	53
LAD_2	102	N.C.	28	N.C.	89	V_{SS}	57
LAD_3	104	N.C.	31	N.C.	94	V_{SS}	67
LAD_4	105	N.C.	34	N.C.	95	V_{SS}	68
LAD_5	106	N.C.	37	N.C.	96	V_{SS}	73
LAD_6	107	N.C.	38	N.C.	97	V_{SS}	79
LAD_7	108	N.C.	39	N.C.	98	V_{SS}	80
LAD_8	109	N.C.	40	N.C.	131	V_{SS}	84
LAD_9	110	N.C.	43	$\overline{READY}$	12	V_{SS}	90
LAD_{10}	111	N.C.	44	RESET	93	V_{SS}	99
LAD_{11}	112	N.C.	45	V_{CC}	29	V_{SS}	103
LAD_{12}	113	N.C.	46	V_{CC}	30	V_{SS}	114
LAD_{13}	115	N.C.	47	V_{CC}	35	V_{SS}	125
LAD_{14}	116	N.C.	48	V_{CC}	36	W/$\overline{R}$	11

1

3.3 Package Thermal Specification

The 80960KA is specified for operation when case temperatures within the range 0°C to 85°C (PGA) or 0°C to 100°C (PQFP). Measure case temperature at the top center of the package. Ambient temperature can be calculated from:

$$T_J = T_C + P * \theta_{JC}$$
$$T_A = T_J + P * \theta_{JA}$$
$$T_C = T_A + P * [\theta_{JA} - \theta_{JC}]$$

Values for θ_{JA} and θ_{JC} for various airflows are given in Table 13 for the PGA package and in Table 14 for the PQFP package. The PGA's θ_{JA} can be reduced by adding a heatsink. For the PQFP, however, a heatsink is not generally used since the device is intended to be surface mounted.

Maximum allowable ambient temperature (T_A) permitted without exceeding T_C is shown by the graphs in Figures 23, 24, 25 and 26. The curves assume the maximum permitted supply current (I_{CC}) at each speed, V_{CC} of +5.0V and a T_{CASE} of +85°C (PGA) or +100°C (PQFP).

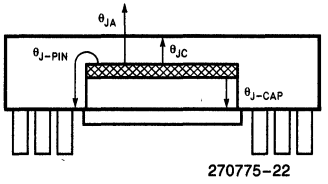
If the 80960KA is to be used in a harsh environment where the ambient temperature may exceed the limits for the normal commercial part, consider using an extended temperature device. These components are designated by the prefix "TA" and are available at 16, 20 and 25 MHz in the ceramic PGA package. Extended operating temperature range is −40°C to +125°C (case).

Figure 26 shows the maximum allowable ambient temperature for the 20 MHz extended temperature TA80960KA at various airflows. The curve assumes an I_{CC} of 420 mA, V_{CC} of 5.0V and a T_{CASE} of +125°C.

Table 13. 80960KA PGA Package Thermal Characteristics

Thermal Resistance—°C/Watt

Parameter	Airflow—ft./min (m/sec)						
	0 (0)	50 (0.25)	100 (0.50)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)
θ Junction-to-Case	2	2	2	2	2	2	2
θ Case-to-Ambient (No Heatsink)	19	18	17	15	12	10	9
θ Case-to-Ambient (Omnidirectional Heatsink)	16	15	14	12	9	7	6
θ Case-to-Ambient (Unidirectional Heatsink)	15	14	13	11	8	6	5



270775-22

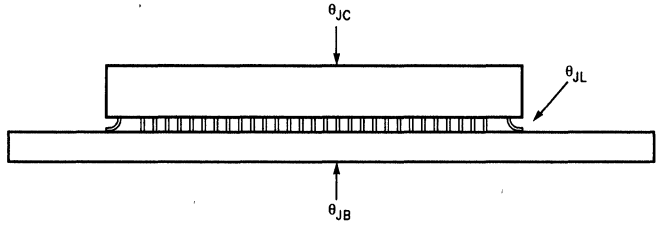
NOTES:
1. This table applies to 80960KA PGA plugged into socket or soldered directly to board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$
3. $\theta_{J\text{-}CAP}$ = 4°C/W approx.)
   $\theta_{J\text{-}PIN}$ = 4°C/W (inner pins) (approx.)
   $\theta_{J\text{-}PIN}$ = 8°C/W (outer pins) (approx.)

Table 14. 80960KA PQFP Package Thermal Characteristics

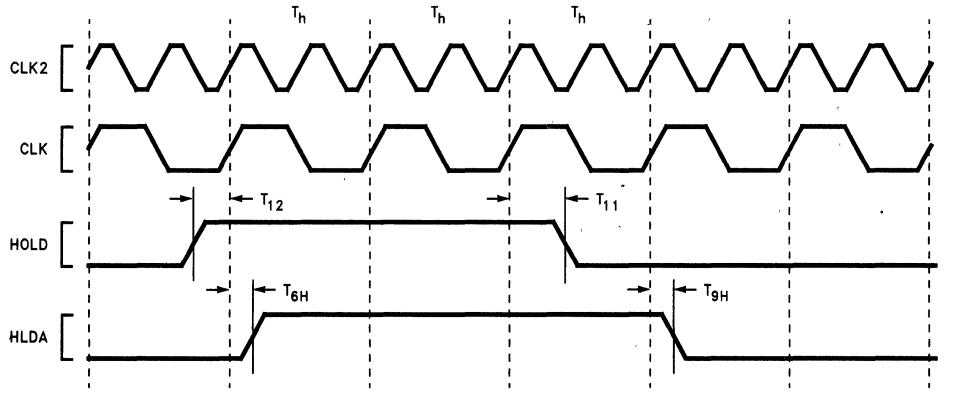
Thermal Resistance—°C/Watt

Parameter	Airflow—ft./min (m/sec)						
	0 (0)	50 (0.25)	100 (0.50)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)
θ Junction-to-Case	9	9	9	9	9	9	9
θ Case-to-Ambient (No Heatsink)	22	19	18	16	11	9	8

NOTES:
1. This table applies to 80960KA PQFP soldered directly to board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$
3. θ_{JL} = 18°C/W (approx.)
   θ_{JB} = 18°C/W (approx.)



270775-23



270775-24

Figure 22. HOLD Timing

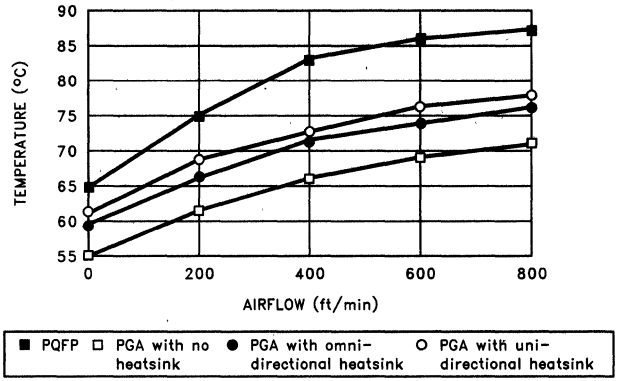


Figure 23. 16 MHz Maximum Allowable Ambient Temperature

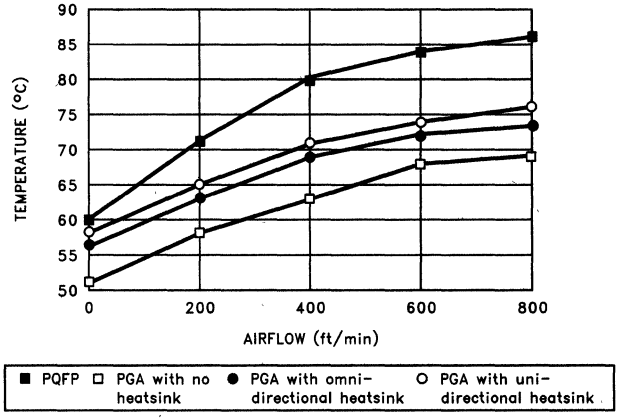


Figure 24. 20 MHz Maximum Allowable Ambient Temperature

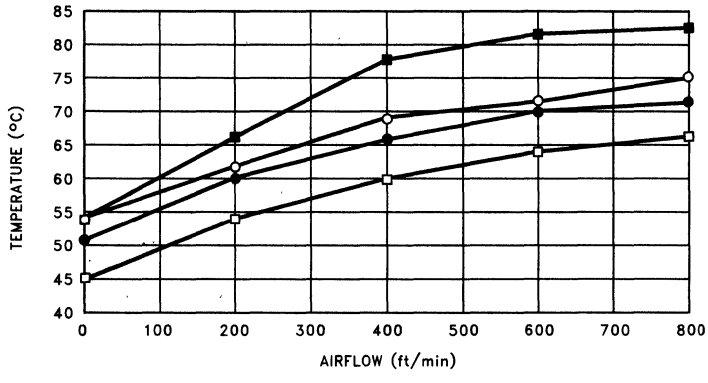


Figure 25. 25 MHz Maximum Allowable Ambient Temperature

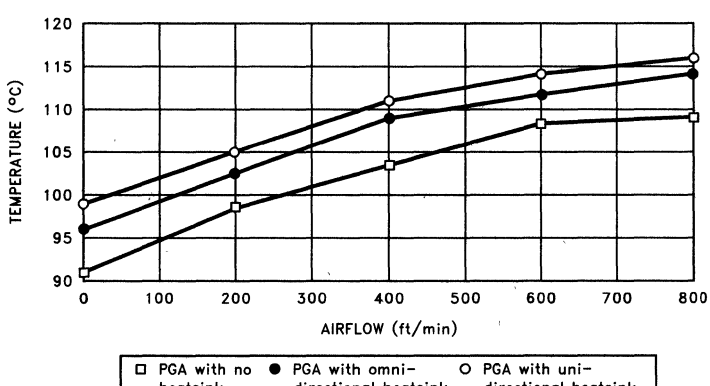


Figure 26. Maximum Allowable Ambient Temperature for the Extended Temperature TA-80960KA 20 MHz in PGA Package

4.0 WAVEFORMS

Figures 27, 28, 29 and 30 show the waveforms for various transactions on the 80960KA's local bus.

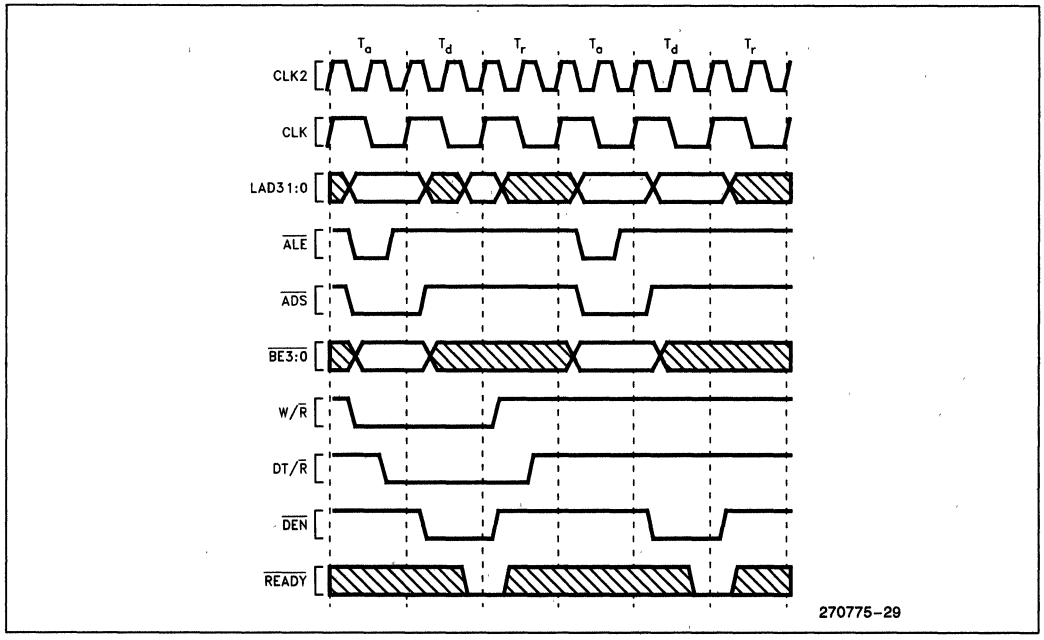


Figure 27. Non-Burst Read and Write Transactions without Wait States

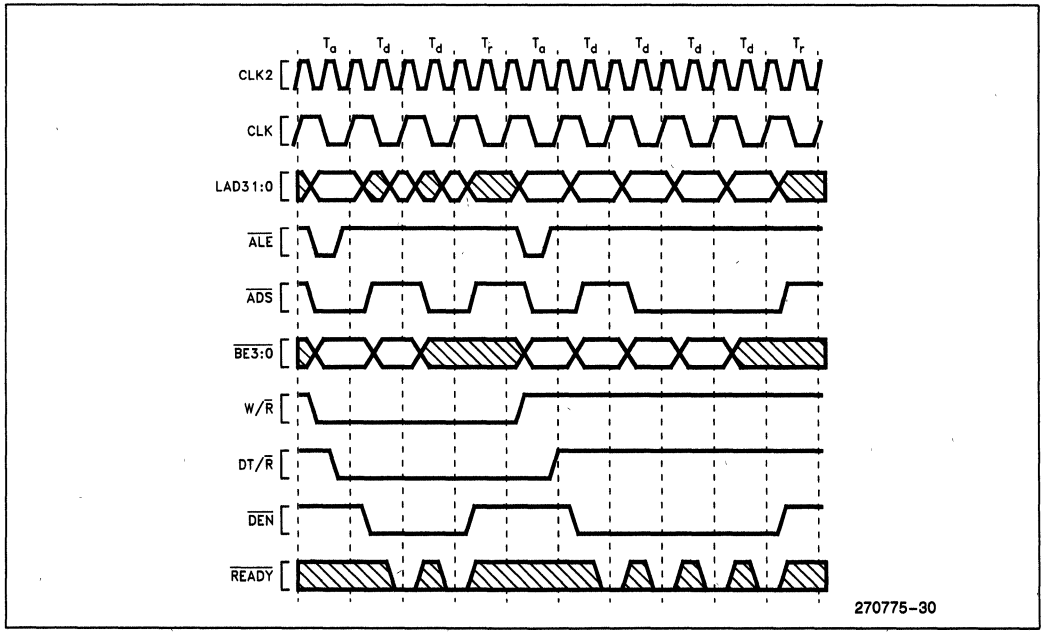


Figure 28. Burst Read and Write Transaction without Wait States

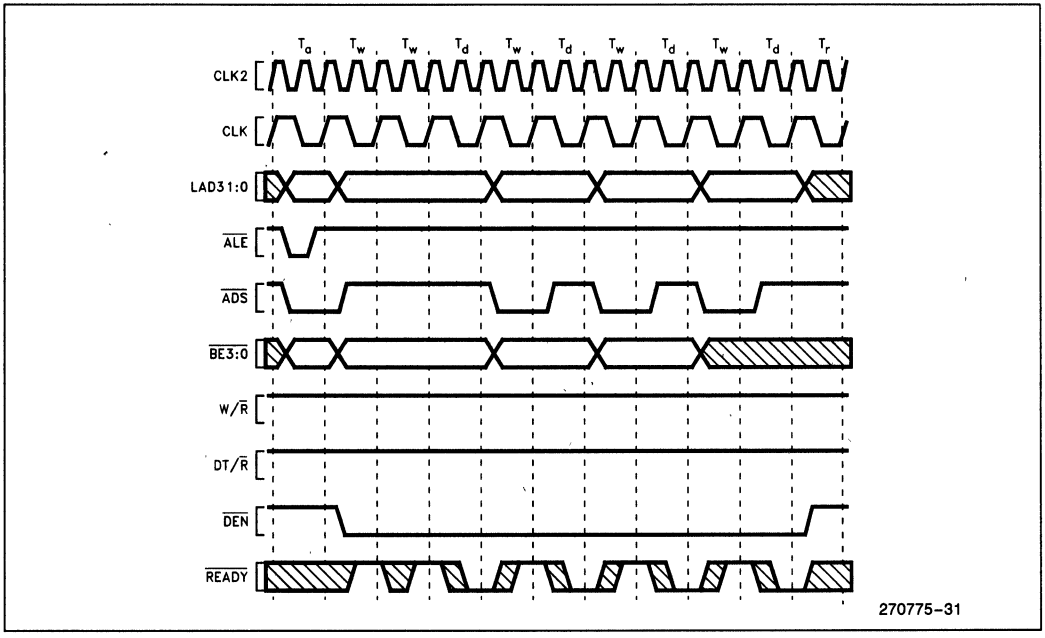


Figure 29. Burst Write Transaction with 2, 1, 1, 1 Wait States

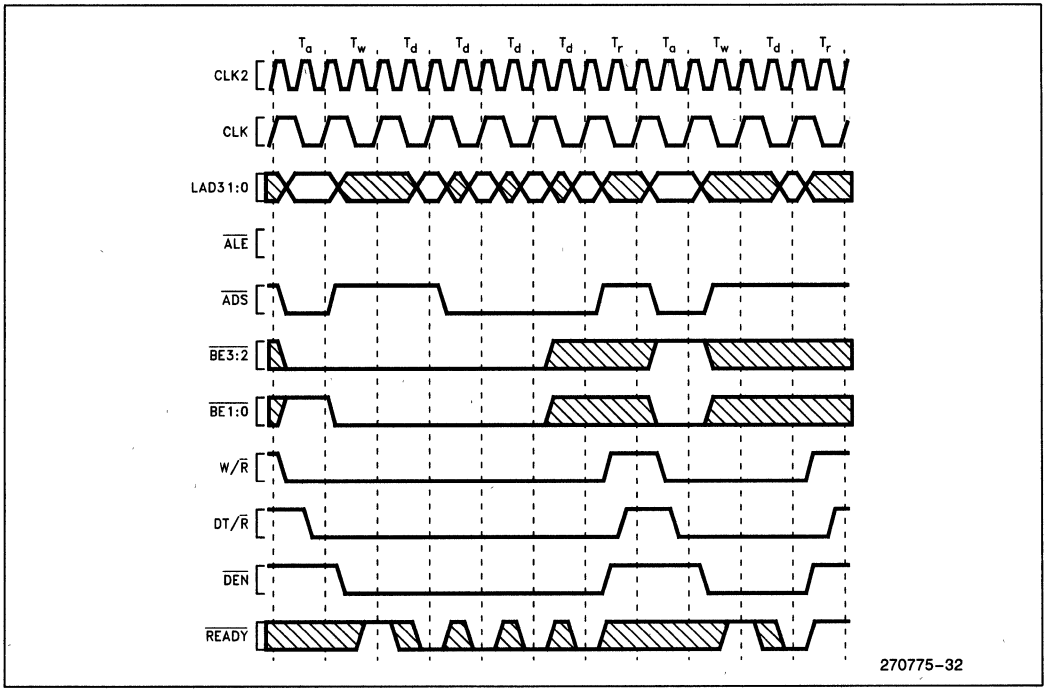
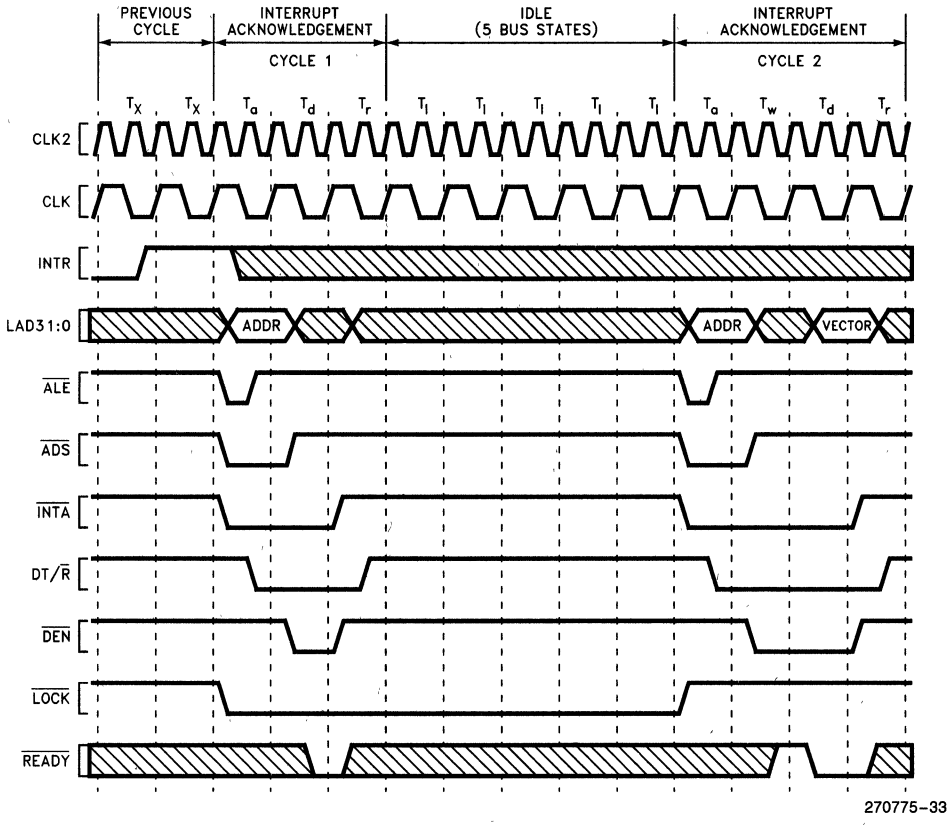


Figure 30. Accesses Generated by Quad Word Read Bus Request, Misaligned Two Bytes from Quad Word Boundary (1, 0, 0, 0 Wait States)



NOTE:
INTR can go low no sooner than the input hold time following the beginning of interrupt acknowledgment cycle 1. For a second interrupt to be acknowledged, INTR must be low for at least three cycles before it can be reasserted.

Figure 31. Interrupt Acknowledge Transaction

5.0 REVISION HISTORY

No revision history was maintained in earlier revisions of this data sheet. All errata that has been identified to date is incorporated into this revision. The sections significantly changed since the previous revision are:

Section	Last Rev.	Description
Table 3. 80960KA Pin Description: L-Bus Signals (pg. 9)	-004	$\overline{LOCK}$ pin description rewritten for clarity.
2.3. Connection Recommendations (pg. 12)	-004	Changed suggested open-drain termination networks to reflect more realistic operating conditions with reduction in DC power consumption.
Figure 9. Typical Current vs. Frequency (Hot Temp) (pg. 14)	-004	Added figure for typical power supply current at hot temperature to aid thermal analysis.
Figure 12. Test Load Circuit for Three-State Output Pins (pg. 15) Figure 13. Test Load Circuit for Open-Drain Output Pins (pg. 15)	-004	All outputs now specified with standard 50 pF test loads to agree with actual test methodology.
2.7. DC Characteristics (pg. 16)	-004	I_{CC} max specification reduced: **WAS: IS: AT:** 375 mA 315 mA 16 MHz 420 mA 360 mA 20 MHz 480 mA 420 mA 25 MHz Figures 7, 8, 9, 23, 24, 25 and 26 have also been changed accordingly.
2.8. AC Specifications (pg. 17)	-004	25 MHz operation extended to product in PQFP package. T_8 min. improved at all frequencies from 0 ns to 2 ns and T_8 max. improved from 20 ns to 18 ns. T_{8H} max improvement: **WAS: IS: AT:** 31 ns 28 ns 16 MHz 26 ns 23 ns 20 MHz 24 ns 23 ns 25 MHz
Functional Waveforms	-004	Redrawn for clarity. CLK signal drawn with more likely phase relationship to CLK2. Open-drain output signals drawn to show correct inactive states.
Various	-004	Deleted all references to 10 MHz. Intel no longer offers a 10 MHz 80960KA device.

80960KB
EMBEDDED 32-BIT MICROPROCESSOR WITH INTEGRATED FLOATING POINT UNIT

- **High-Performance Embedded Architecture**
  - **25 MIPS Burst Execution at 25 MHz**
  - **9.4 MIPS* Sustained Execution at 25 MHz**
- **512-Byte On-Chip Instruction Cache**
  - **Direct Mapped**
  - **Parallel Load/Decode for Uncached Instructions**
- **Multiple Register Sets**
  - **Sixteen Global 32-Bit Registers**
  - **Sixteen Local 32-Bit Registers**
  - **Four Local Register Sets Stored On-Chip**
  - **Register Scoreboarding**
- **4 Gigabyte, Linear Address Space**
- **Pin Compatible with 80960KA**
- **Built-In Interrupt Controller**
  - **31 Priority Levels, 256 Vectors**
  - **3.4 μs Latency @ 25 MHz**
- **Easy to Use, High Bandwidth 32-Bit Bus**
  - **66.7 Mbytes/s Burst**
  - **Up to 16 Bytes Transferred per Burst**
- **132-Lead Packages:**
  - **Pin Grid Array (PGA)**
  - **Plastic Quad Flat-Pack (PQFP)**
- **On-Chip Floating Point Unit**
  - **Supports IEEE 754 Floating Point Standard**
  - **Four 80-Bit Registers**
  - **13.6 Million Whetstones/s (Single Precision) at 25 MHz**

The 80960KB is a member of Intel's i960® 32-bit processor family, which is designed especially for embedded applications. It includes a 512-byte instruction cache, an integrated floating point unit, and a built-in interrupt controller. The 80960KB has a large register set, multiple parallel execution units and a high-bandwidth burst bus. Using advanced RISC technology, this high performance processor is capable of execution rates in excess of 9.4 million instructions per second*. The 80960KB is well-suited for a wide range of applications including non-impact printers, I/O control and specialty instrumentation.

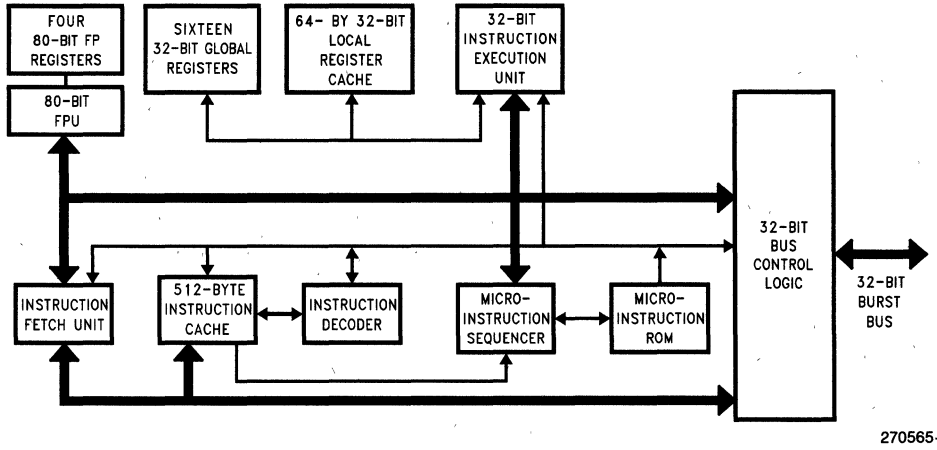


Figure 1. The 80960KB Processor's Highly Parallel Architecture

*Relative to Digital Equipment Corporation's VAX-11/780 at 1 MIPS (VAX-11 is a trademark of Digital Equipment Corporation.)

October 1993
Order Number: 270565-007

80960KB
EMBEDDED 32-BIT MICROPROCESSOR WITH INTEGRATED FLOATING POINT UNIT

CONTENTS

CONTENTS

1.0 THE i960® PROCESSOR

The 80960KB is a member of the 32-bit architecture from Intel known as the i960 processor family. These were especially designed to serve the needs of embedded applications. The embedded market includes applications as diverse as industrial automation, avionics, image processing, graphics and networking. These types of applications require high integration, low power consumption, quick interrupt response times and high performance. Since time to market is critical, embedded microprocessors need to be easy to use in both hardware and software designs.

All members of the i960 processor family share a common core architecture which utilizes RISC technology so that, except for special functions, the family members are object-code compatible. Each new processor in the family adds its own special set of functions to the core to satisfy the needs of a specific application or range of applications in the embedded market.

Software written for the 80960KB will run without modification on any other member of the 80960 Family. It is also pin-compatible with the 80960KA and the 80960MC which is a military-grade version that supports multitasking, memory management, multiprocessing and fault tolerance.

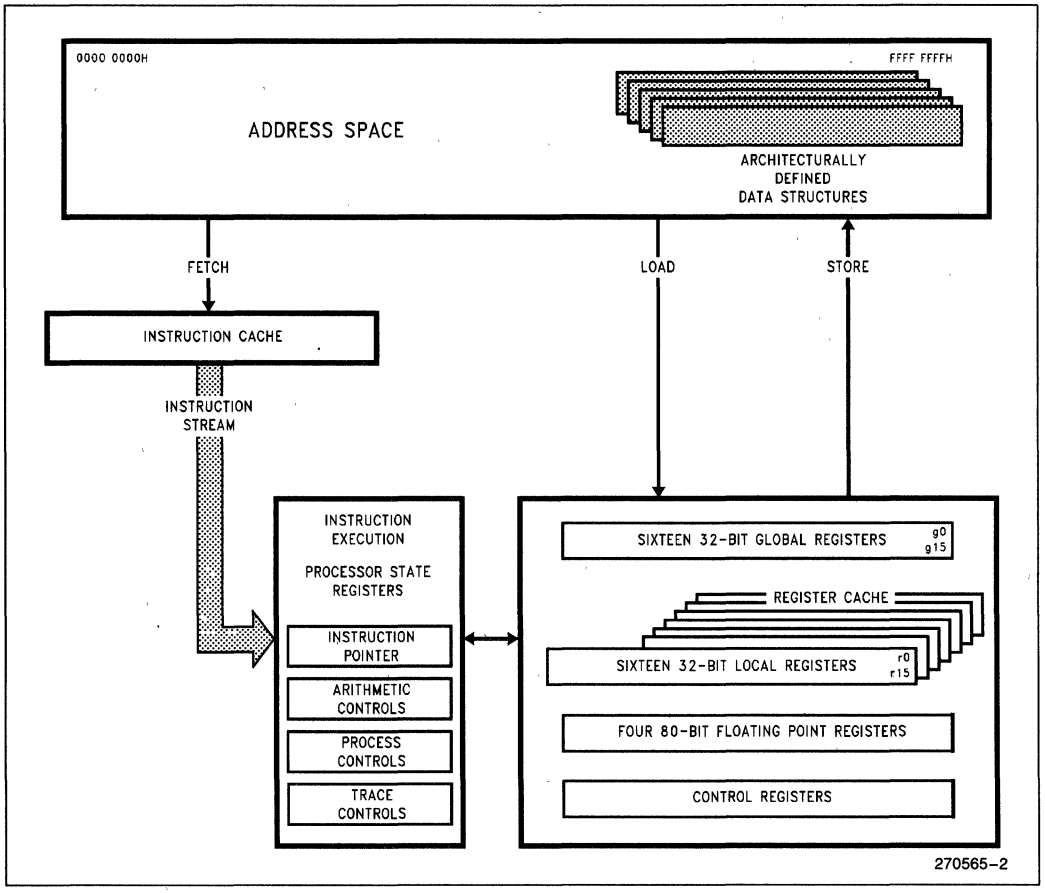


Figure 2. 80960KB Programming Environment

1.1 Key Performance Features

The 80960 architecture is based on the most recent advances in microprocessor technology and is grounded in Intel's long experience in the design and manufacture of embedded microprocessors. Many features contribute to the 80960KB's exceptional performance:

1. **Large Register Set.** Having a large number of registers reduces the number of times that a processor needs to access memory. Modern compilers can take advantage of this feature to optimize execution speed. For maximum flexibility, the 80960KB provides thirty-two 32-bit registers and four 80-bit floating point registers. (See Figure 2.)
2. **Fast Instruction Execution.** Simple functions make up the bulk of instructions in most programs so that execution speed can be improved by ensuring that these core instructions are executed as quickly as possible. The most frequently executed instructions such as register-register moves, add/subtract, logical operations and shifts execute in one to two cycles. (Table 1 contains a list of instructions.)
3. **Load/Store Architecture.** One way to improve execution speed is to reduce the number of times that the processor must access memory to perform an operation. As with other processors based on RISC technology, the 80960KB has a Load/Store architecture. As such, only the LOAD and STORE instructions reference memory; all other instructions operate on registers. This type of architecture simplifies instruction decoding and is used in combination with other techniques to increase parallelism.

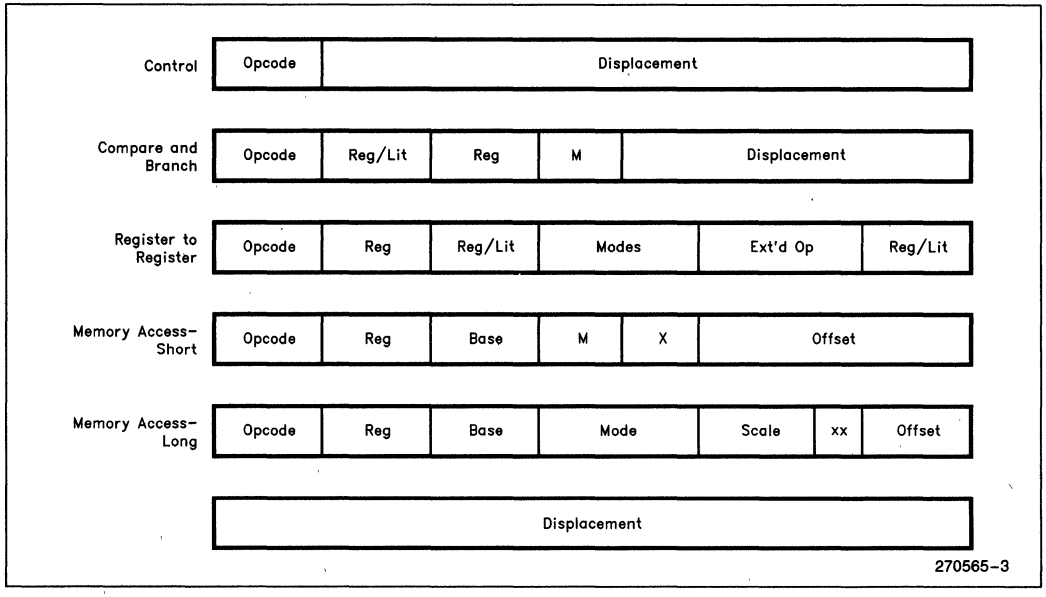


Figure 3. Instruction Formats

Table 1. 80960KB Instruction Set

Data Movement	Arithmetic	Logical	Bit and Bit Field
Load Store Move Load Address	Add Subtract Multiply Divide Remainder Modulo Shift	And Not And And Not Or Exclusive Or Not Or Or Not Exclusive Nor Not Nand Rotate	Set Bit Clear Bit Not Bit Check Bit Alter Bit Scan For Bit Scan Over Bit Extract Modify
Comparison	**Branch**	**Call/Return**	**Fault**
Compare Conditional Compare Compare and Increment Compare and Decrement	Unconditional Branch Conditional Branch Compare and Branch	Call Call Extended Call System Return Branch and Link	Conditional Fault Synchronize Faults
Debug	**Miscellaneous**	**Decimal**	**Floating Point**
Modify Trace Controls Mark Force Mark	Atomic Add Atomic Modify Flush Local Registers Modify Arithmetic Controls Scan Byte for Equal Test Condition Code Modify Process Controls	Decimal Move Decimal Add with Carry Decimal Subtract with Carry	Move Real Add Subtract Multiply Divide Remainder Scale Round Square Root Sine Cosine Tangent Arctangent Log Log Binary Log Natural Exponent Classify Copy Real Extended Compare
		Synchronous	**Conversion**
		Synchronous Load Synchronous Move	Convert Real to Integer Convert Integer to Real

4. **Simple Instruction Formats.** All instructions in the 80960KB are 32 bits long and must be aligned on word boundaries. This alignment makes it possible to eliminate the instruction alignment stage in the pipeline. To simplify the instruction decoder, there are only five instruction formats; each instruction uses only one format. (See Figure 3.)
5. **Overlapped Instruction Execution.** Load operations allow execution of subsequent instructions to continue before the data has been retumed from memory, so that these instructions can overlap the load. The 80960KB manages this process transparently to software through the use of a register scoreboard. Conditional instructions also make use of a scoreboard so that subsequent unrelated instructions may be executed while the conditional instruction is pending.
6. **Integer Execution Optimization.** When the result of an arithmetic execution is used as an operand in a subsequent calculation, the value is sent immediately to its destination register. Yet at the same time, the value is put on a bypass path to the ALU, thereby saving the time that otherwise would be required to retrieve the value for the next operation.
7. **Bandwidth Optimizations.** The 80960KB gets optimal use of its memory bus bandwidth because the bus is tuned for use with the on-chip instruction cache: instruction cache line size matches the maximum burst size for instruction fetches. The 80960KB automatically fetches four words in a burst and stores them directly in the cache. Due to the size of the cache and the fact that it is continually filled in anticipation of needed instructions in the program flow, the 80960KB is relatively insensitive to memory wait states. The benefit is that the 80960KB delivers outstanding performance even with a low cost memory system.
8. **Cache Bypass.** If a cache miss occurs, the processor fetches the needed instruction then sends it on to the instruction decoder at the same time it updates the cache. Thus, no extra time is spent to load and read the cache.

1.1.1 MEMORY SPACE AND ADDRESSING MODES

The 80960KB offers a linear programming environment so that all programs running on the processor are contained in a single address space. Maximum address space size is 4 Gigabytes (2^{32} bytes).

For ease of use the 80960KB has a small number of addressing modes, but includes all those necessary to ensure efficient execution of high-level languages such as C. Table 2 lists the modes.

Table 2. Memory Addressing Modes

- 12-Bit Offset
- 32-Bit Offset
- Register-Indirect
- Register + 12-Bit Offset
- Register + 32-Bit Offset
- Register + (Index-Register × Scale-Factor)
- Register × Scale Factor + 32-Bit Displacement
- Register + (Index-Register × Scale-Factor) + 32-Bit Displacement
- Scale-Factor is 1, 2, 4, 8 or 16

1.1.2 DATA TYPES

The 80960KB recognizes the following data types:

Numeric:

- 8-, 16-, 32- and 64-bit ordinals
- 8-, 16-, 32- and 64-bit integers
- 32-, 64-, and 80-bit real numbers

Non-Numeric:

- Bit
- Bit Field
- Triple Word (96 bits)
- Quad-Word (128 bits)

1.1.3 LARGE REGISTER SET

The 80960KB programming environment includes a large number of registers. In fact, 32 registers are available at any time. The availability of this many registers greatly reduces the number of memory accesses required to perform algorithms, which leads to greater instruction processing speed.

There are two types of general-purpose registers: local and global. The global registers consist of sixteen 32-bit registers (G0 though G15) and four 80-bit registers (FP0 through FP3). These registers perform the same function as the general-purpose registers provided in other popular microprocessors. The term global refers to the fact that these registers retain their contents across procedure calls.

The local registers, on the other hand, are procedure specific. For each procedure call, the 80960KB allocates 16 local registers (R0 through R15). Each local register is 32 bits wide. Any register can also be used for single or double precision floating point operations; the 80-bit floating point registers are provided for extended precision.

1.1.4 MULTIPLE REGISTER SETS

To further increase the efficiency of the register set, multiple sets of local registers are stored on-chip (see Figure 4). This cache holds up to four local register frames, which means that up to three procedure calls can be made without having to access the procedure stack resident in memory.

Although programs may have procedure calls nested many calls deep, a program typically oscillates back and forth between only two to three levels. As a result, with four stack frames in the cache, the probability of having a free frame available on the cache when a call is made is very high. In fact, runs of representative C-language programs show that 80% of the calls are handled without needing to access memory.

If four or more procedures are active and a new procedure is called, the 80960KB moves the oldest local register set in the stack-frame cache to a procedure stack in memory to make room for a new set of registers. Global register G15 is the frame pointer (FP) to the procedure stack.

Global and floating point registers are not exchanged on a procedure call, but retain their contents, making them available to all procedures for fast parameter passing.

1.1.5 INSTRUCTION CACHE

To further reduce memory accesses, the 80960KB includes a 512-byte on-chip instruction cache. The instruction cache is based on the concept of locality of reference; most programs are not usually executed in a steady stream but consist of many branches, loops and procedure calls that lead to jumping back and forth in the same small section of code. Thus, by maintaining a block of instructions in cache, the number of memory references required to read instructions into the processor is greatly reduced.

To load the instruction cache, instructions are fetched in 16-byte blocks; up to four instructions can be fetched at one time. An efficient prefetch algorithm increases the probability that an instruction will already be in the cache when it is needed.

Code for small loops often fits entirely within the cache, leading to a great increase in processing speed since further memory references might not be necessary until the program exits the loop. Similarly, when calling short procedures, the code for the calling procedure is likely to remain in the cache so it will be there on the procedure's return.

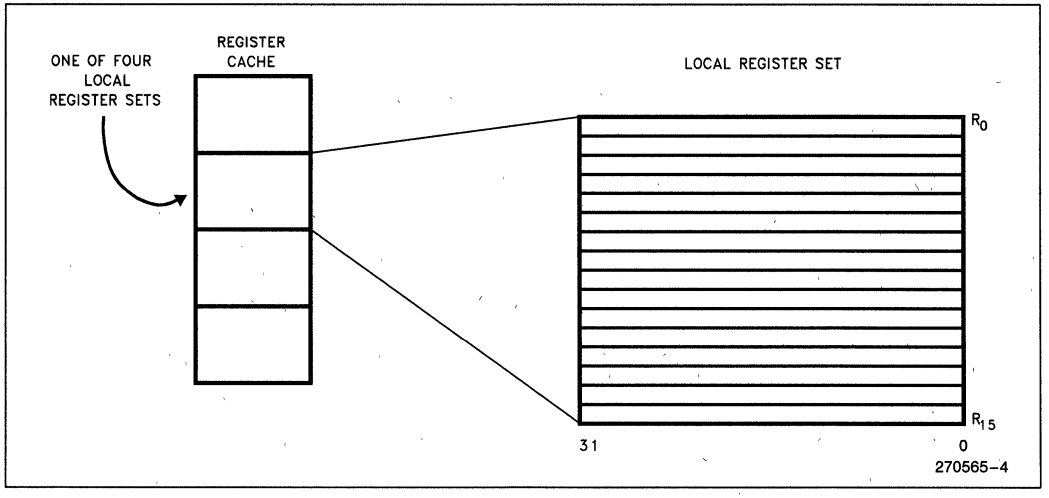


Figure 4. Multiple Register Sets Are Stored On-Chip

1.1.6 REGISTER SCOREBOARDING

The instruction decoder is optimized in several ways. One optimization method is the ability to overlap instructions by using register scoreboarding.

Register scoreboarding occurs when a LOAD moves a variable from memory into a register. When the instruction initiates, a scoreboard bit on the target register is set. Once the register is loaded, the bit is reset. In between, any reference to the register contents is accompanied by a test of the scoreboard bit to ensure that the load has completed before processing continues. Since the processor does not need to wait for the LOAD to complete, it can execute additional instructions placed between the LOAD and the instruction that uses the register contents, as shown in the following example:

```
ld data_2, r4
ld data_2, r5
Unrelated instruction
Unrelated instruction
add R4, R5, R6
```

In essence, the two unrelated instructions between LOAD and ADD are executed "for free" (i.e., take no apparent time to execute) because they are executed while the register is being loaded. Up to three load instructions can be pending at one time with three corresponding scoreboard bits set. By exploiting this feature, system programmers and compiler writers have a useful tool for optimizing execution speed.

1.1.7 FLOATING POINT ARITHMETIC

In the 80960KB, floating point arithmetic has been made an integral part of the architecture. Having the floating point unit integrated on chip provides two advantages. First, it improves the performance of the chip for floating point applications, since no additional bus overhead is associated with floating point calculations, thereby leaving more time for other bus operations such as I/O. Second, the cost of using floating point operations is reduced because a separate coprocessor chip is not required.

The 80960KB floating point (real number) data types include single precision (32-bit), double precision (64-bit) and extended precision (80-bit) floating point numbers. Any registers may be used to execute floating point operations.

The processor provides hardware support for both mandatory and recommended portions of IEEE Standard 754 for floating point arithmetic, including all arithmetic, exponential, logarithmic and other transcendental functions. Table 3 shows execution times for some representative instructions.

Table 3. Sample Floating Point Execution Times (μs) at 25 MHz

Function	32-Bit	64-Bit
Add	0.4	0.5
Subtract	0.4	0.5
Multiply	0.7	1.3
Divide	1.3	2.9
Square Root	3.7	3.9
Arctangent	10.1	13.1
Exponent	11.3	12.5
Sine	15.2	16.6
Cosine	15.2	16.6

1.1.8 HIGH BANDWIDTH LOCAL BUS

The 80960KB CPU resides on a high-bandwidth address/data bus known as the local bus (L-Bus). The L-Bus provides a direct communication path between the processor and the memory and I/O subsystem interfaces. The processor uses the L-Bus to fetch instructions, manipulate memory and respond to interrupts. L-Bus features include:

- 32-bit multiplexed address/data path
- Four-word burst capability which allows transfers from 1 byte to 16 bytes at a time
- High bandwidth reads and writes with 66.7 MBytes/s burst (at 25 MHz)

Table 4 defines L-bus signal names and functions; Table 5 defines other component-support signals such as interrupt lines.

1.1.9 INTERRUPT HANDLING

The 80960KB can be interrupted in two ways: by the activation of one of four interrupt pins or by sending a message on the processor's data bus.

The 80960KB is unusual in that it automatically handles interrupts on a priority basis and can keep track of pending interrupts through its on-chip interrupt controller. Two of the interrupt pins can be configured to provide 8259A-style handshaking for expansion beyond four interrupt lines.

1.1.10 DEBUG FEATURES

The 80960KB has built-in debug capabilities. There are two types of breakpoints and six trace modes. Debug features are controlled by two internal 32-bit registers: the Process-Controls Word and the Trace-Controls Word. By setting bits in these control words, a software debug monitor can closely control how the processor responds during program execution.

The 80960KB provides two hardware breakpoint registers on-chip which, by using a special command, can be set to any value. When the instruction pointer matches either breakpoint register value, the breakpoint handling routine is automatically called.

The 80960KB also provides software breakpoints through the use of two instructions: MARK and FMARK. These can be placed at any point in a program and cause the processor to halt execution at that point and call the breakpoint handling routine. The breakpoint mechanism is easy to use and provides a powerful debugging tool.

Tracing is available for instructions (single step execution), calls and returns and branching. Each trace type may be enabled separately by a special debug instruction. In each case, the 80960KB executes the instruction first and then calls a trace handling routine (usually part of a software debug monitor). Further program execution is halted until the routine completes, at which time execution resumes at the next instruction. The 80960KB's tracing mechanisms, implemented completely in hardware, greatly simplify the task of software test and debug.

1.1.11 FAULT DETECTION

The 80960KB has an automatic mechanism to handle faults. Fault types include floating point, trace and arithmetic faults. When the processor detects a fault, it automatically calls the appropriate fault handling routine and saves the current instruction pointer and necessary state information to make efficient recovery possible. Like interrupt handling routines, fault handling routines are usually written to meet the needs of specific applications and are often included as part of the operating system or kernel.

For each of the fault types, there are numerous subtypes that provide specific information about a fault. For example, a floating point fault may have the subtype set to an Overflow or Zero Divide fault. The fault handler can use this specific information to respond correctly to the fault.

1.1.12 BUILT-IN TESTABILITY

Upon reset, the 80960KB automatically conducts an exhaustive internal test of its major blocks of logic. Then, before executing its first instruction, it does a zero check sum on the first eight words in memory to ensure that the memory image was programmed correctly. If a problem is discovered at any point during the self-test, the 80960KB asserts its $\overline{\text{FAILURE}}$ pin and will not begin program execution. Self test takes approximately 47,000 cycles to complete.

System manufacturers can use the 80960KB's self-test feature during incoming parts inspection. No special diagnostic programs need to be written. The test is both thorough and fast. The self-test capability helps ensure that defective parts are discovered before systems are shipped and, once in the field, the self-test makes it easier to distinguish between problems caused by processor failure and problems resulting from other causes.

1.1.13 CHMOS

The 80960KB is fabricated using Intel's CHMOS IV (Complementary High Speed Metal Oxide Semiconductor) process. The 80960KB is currently available in 16, 20 and 25 MHz versions.

Table 4. 80960KB Pin Description: L-Bus Signals

Name	Type	Description
CLK2	I	**SYSTEM CLOCK** provides the fundamental timing for 80960KB systems. It is divided by two inside the 80960KB and four 80-bit registers (FP0 through FP3) to generate the internal processor clock.
LAD31:0	I/O T.S.	**LOCAL ADDRESS/DATA BUS** carries 32-bit physical addresses and data to and from memory. During an address (T_a) cycle, bits 2–31 contain a physical word address (bits 0–1 indicate SIZE; see below). During a data (T_d) cycle, bits 0–31 contain read or write data. These pins float to a high impedance state when not active. Bits 0–1 comprise SIZE during a T_a cycle. SIZE specifies burst transfer size in words. **LAD1 LAD0** 0 0 1 Word 0 1 2 Words 1 0 3 Words 1 1 4 Words
$\overline{ALE}$	O T.S.	**ADDRESS LATCH ENABLE** indicates the transfer of a physical address. $\overline{ALE}$ is asserted during a T_a cycle and deasserted before the beginning of the T_d state. It is active LOW and floats to a high impedance state during a hold cycle (T_h).
$\overline{ADS}$	O O.D.	**ADDRESS/DATA STATUS** indicates an address state. $\overline{ADS}$ is asserted every T_a state and deasserted during the following T_d state. For a burst transaction, $\overline{ADS}$ is asserted again every T_d state where $\overline{READY}$ was asserted in the previous cycle.
W/$\overline{R}$	O O.D.	**WRITE/READ** specifies, during a T_a cycle, whether the operation is a write or read. It is latched on-chip and remains valid during T_d cycles.
DT/$\overline{R}$	O O.D.	**DATA TRANSMIT/RECEIVE** indicates the direction of data transfer to and from the L-Bus. It is low during T_a and T_d cycles for a read or interrupt acknowledgment; it is high during T_a and T_d cycles for a write. DT/$\overline{R}$ never changes state when $\overline{DEN}$ is asserted.
$\overline{READY}$	I	**READY** indicates that data on LAD lines can be sampled or removed. If $\overline{READY}$ is not asserted during a T_d cycle, the T_d cycle is extended to the next cycle by inserting a wait state (T_w) and $\overline{ADS}$ is not asserted in the next cycle.
$\overline{LOCK}$	I/O O.D.	**BUS LOCK** prevents bus masters from gaining control of the L-Bus during Read/Modify/Write (RMW) cycles. The processor or any bus agent may assert $\overline{LOCK}$. At the start of a RMW operation, the processor examines the $\overline{LOCK}$ pin. If the pin is already asserted, the processor waits until it is not asserted. If the pin is not asserted, the processor asserts $\overline{LOCK}$ during the T_a cycle of the read transaction. The processor deasserts $\overline{LOCK}$ in the T_a cycle of the write transaction. During the time $\overline{LOCK}$ is asserted, a bus agent can perform a normal read or write but not a RMW operation. The processor also asserts $\overline{LOCK}$ during interrupt-acknowledge transactions. Do not leave $\overline{LOCK}$ unconnected. It must be pulled high for the processor to function properly.

I/O = Input/Output, O = Output, I = Input, O.D. = Open Drain, T.S. = Three-State

Table 4. 80960KB Pin Description: L-Bus Signals (Continued)

Name	Type	Description
$\overline{BE3:0}$	O O.D.	**BYTE ENABLE LINES** specify the data bytes (up to four) on the bus which are used in the current bus cycle. $\overline{BE3}$ corresponds to LAD31:24; $\overline{BE0}$ corresponds to LAD7:0. The byte enables are provided in advance of data: • Byte enables asserted during T_a specify the bytes of the first data word. • Byte enables asserted during T_d specify the bytes of the next data word, if any (the word to be transmitted following the next assertion of $\overline{READY}$). Byte enables that occur during T_d cycles that precede the last assertion of $\overline{READY}$ are undefined. Byte enables are latched on-chip and remain constant from one T_d cycle to the next when $\overline{READY}$ is not asserted. For reads, byte enables specify the byte(s) that the processor will actually use. L-Bus agents are required to assert only adjacent byte enables (e.g., asserting just $\overline{BE0}$ and $\overline{BE2}$ is not permitted) and are required to assert at least one byte enable. Address bits A_0 and A_1 can be decoded externally from the byte enables.
HOLD	I	**HOLD:** A request from an external bus master to acquire the bus. When the processor receives HOLD and grants bus control to another master, it floats its three- state bus lines and open-drain control lines, asserts HLDA and enters the T_h state. When HOLD deasserts, the processor deasserts HLDA and enters the T_i or T_a state.
HLDA	O T.S.	**HOLD ACKNOWLEDGE:** Notifies an external bus master that the processor has relinquished control of the bus.
CACHE	O T.S.	**CACHE** indicates when an access is cacheable during a T_a cycle. It is not asserted during any synchronous access, such as a synchronous load or move instruction used for sending an IAC message. The CACHE signal floats to a high impedance state when the processor is idle.

I/O = Input/Output, O = Output, I = Input, O.D. = Open Drain, T.S. = Three-State

Table 5. 80960KB Pin Description: Support Signals

Name	Type	Description
$\overline{BADAC}$	I	**BAD ACCESS,** if asserted in the cycle following the one in which the last $\overline{READY}$ of a transaction is asserted, indicates an unrecoverable error occurred on the current bus transaction or a synchronous load/store instruction has not been acknowledged. During system reset the $\overline{BADAC}$ signal is interpreted differently. If the signal is high, it indicates that this processor will perform system initialization. If it is low, another processor in the system will perform system initialization instead.
RESET	I	**RESET** clears the processor's internal logic and causes it to reinitialize. During RESET assertion, the input pins are ignored (except for $\overline{BADAC}$ and $\overline{IAC}/\overline{INT_0}$), the three-state output pins are placed in a high impedance state and other output pins are placed in their non-asserted states. RESET must be asserted for at least 41 CLK2 cycles for a predictable RESET. The HIGH to LOW transition of RESET should occur after the rising edge of both CLK2 and the external bus clock and before the next rising edge of CLK2.

I/O = Input/Output, O = Output, I = Input, O.D. = Open Drain, T.S. = Three-State

Table 5. 80960KB Pin Description: Support Signals (Continued)

Name	Type	Description
$\overline{\text{FAILURE}}$	O O.D.	**INITIALIZATION FAILURE** indicates that the processor did not initialize correctly. After RESET deasserts and before the first bus transaction begins, $\overline{\text{FAILURE}}$ asserts while the processor performs a self-test. If the self-test completes successfully, then $\overline{\text{FAILURE}}$ deasserts. The processor then performs a zero checksum on the first eight words of memory. If it fails, $\overline{\text{FAILURE}}$ asserts for a second time and remains asserted. If it passes, system initialization continues and $\overline{\text{FAILURE}}$ remains deasserted.
$\overline{\text{IAC}}/\overline{\text{INT}}_0$	I	**INTERAGENT COMMUNICATION REQUEST/INTERRUPT 0** indicates an IAC message or an interrupt is pending. The bus interrupt control register determines how the signal is interpreted. To signal an interrupt or IAC request in a synchronous system, this pin—as well as the other interrupt pins—must be enabled by being deasserted for at least one bus cycle and then asserted for at least one additional bus cycle. In an asynchronous system the pin must remain deasserted for at least two bus cycles and then asserted for at least two more bus cycles. During system reset, this signal must be in the logic high condition to enable normal processor operation. The logic low condition is reserved.
INT_1	I	**INTERRUPT 1,** like $\overline{\text{INT}}_0$, provides direct interrupt signaling.
INT_2/INTR	I	**INTERRUPT 2/INTERRUPT REQUEST:** The interrupt control register determines how this pin is interpreted. If INT_2, it has the same interpretation as the $\overline{\text{INT}}_0$ and INT_1 pins. If INTR, it is used to receive an interrupt request from an external interrupt controller.
$\overline{\text{INT}}_3/\overline{\text{INTA}}$	I/O O.D.	**INTERRUPT 3/INTERRUPT ACKNOWLEDGE:** The bus interrupt control register determines how this pin is interpreted. If $\overline{\text{INT}}_3$, it has the same interpretation as the $\overline{\text{INT}}_0$ INT_1 and INT_2 pins. If $\overline{\text{INTA}}$, it is used as an output to control interrupt-acknowledge transactions. The $\overline{\text{INTA}}$ output is latched on-chip and remains valid during T_d cycles; as an output, it is open-drain.
N.C.	N/A	**NOT CONNECTED** indicates pins should not be connected. Never connect any pin marked N.C. as these pins may be reserved for factory use.

I/O = Input/Output, O = Output, I = Input, O.D. = Open Drain, T.S. = Three-State

2.0 ELECTRICAL SPECIFICATIONS

2.1 Power and Grounding

The 80960KB is implemented in CHMOS IV technology and therefore has modest power requirements. Its high clock frequency and numerous output buffers (address/data, control, error and arbitration signals) can cause power surges as multiple output buffers simultaneously drive new signal levels. For clean on-chip power distribution, V_{CC} and V_{SS} pins separately feed the device's functional units. Power and ground connections must be made to all 80960KB power and ground pins. On the circuit board, all V_{CC} pins must be strapped closely together, preferably on a power plane; all V_{SS} pins should be strapped together, preferably on a ground plane.

2.2 Power Decoupling Recommendations

Place a liberal amount of decoupling capacitance near the 80960KB. When driving the L-bus the processor can cause transient power surges, particularly when connected to a large capacitive load.

Low inductance capacitors and interconnects are recommended for best high frequency electrical performance. Inductance is reduced by shortening board traces between the processor and decoupling capacitors as much as possible.

2.3 Connection Recommendations

For reliable operation, always connect unused inputs to an appropriate signal level. In particular, if one or more interrupt lines are not used, they should be pulled up. No inputs should ever be left floating.

All open-drain outputs require a pullup device. While in most cases a simple pullup resistor is adequate, a network of pullup and pulldown resistors biased to a valid V_{IH} (>3.0V) and terminated in the characteristic impedance of the circuit board is recommended to limit noise and AC power consumption. Figure 5 and Figure 6 show recommended values for the resistor network for low and high current drive, assuming a characteristic impedance of 100Ω. Terminating output signals in this fashion limits signal swing and reduces AC power consumption.

NOTE:
Do not connect external logic to pins marked N.C.

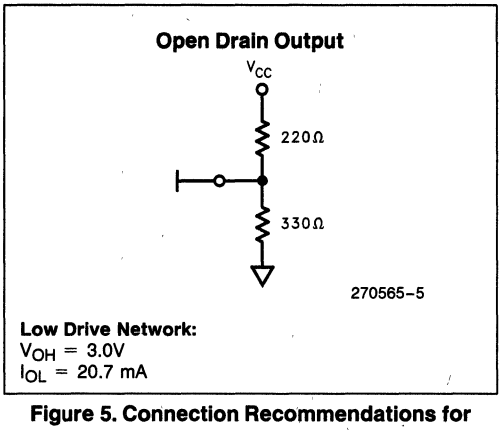


Figure 5. Connection Recommendations for Low Current Drive Network

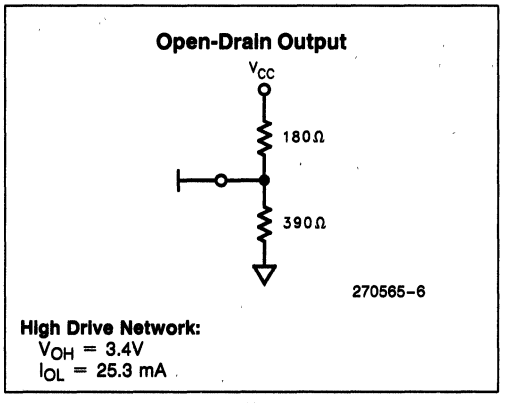


Figure 6. Connection Recommendations for High Current Drive Network

2.4 Characteristic Curves

Figure 7 shows typical supply current requirements over the operating temperature range of the processor at supply voltage (V_{CC}) of 5V. Figure 8 and Figure 9 show the typical power supply current (I_{CC}) that the 80960KB requires at various operating frequencies when measured at three input voltage (V_{CC}) levels and two temperatures.

For a given output current (I_{OL}) the curve in Figure 10 shows the worst case output low voltage (V_{OL}). Figure 11 shows the typical capacitive derating curve for the 80960KB measured from 1.5V on the system clock (CLK) to 1.5V on the falling edge and 1.5V on the rising edge of the L-Bus address/data (LAD) signals.

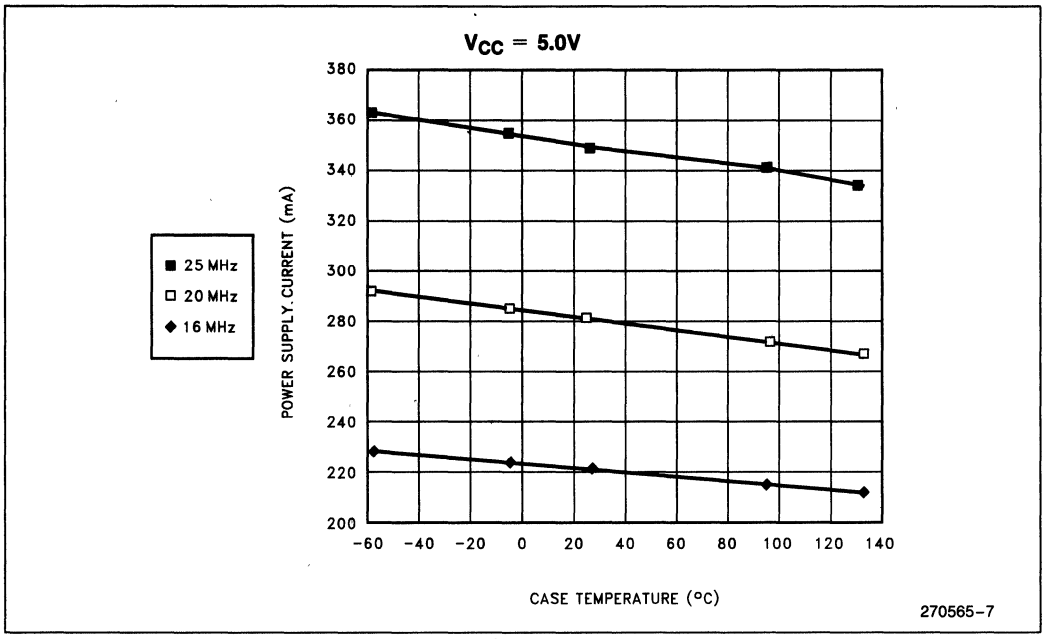


Figure 7. Typical Supply Current vs Case Temperature

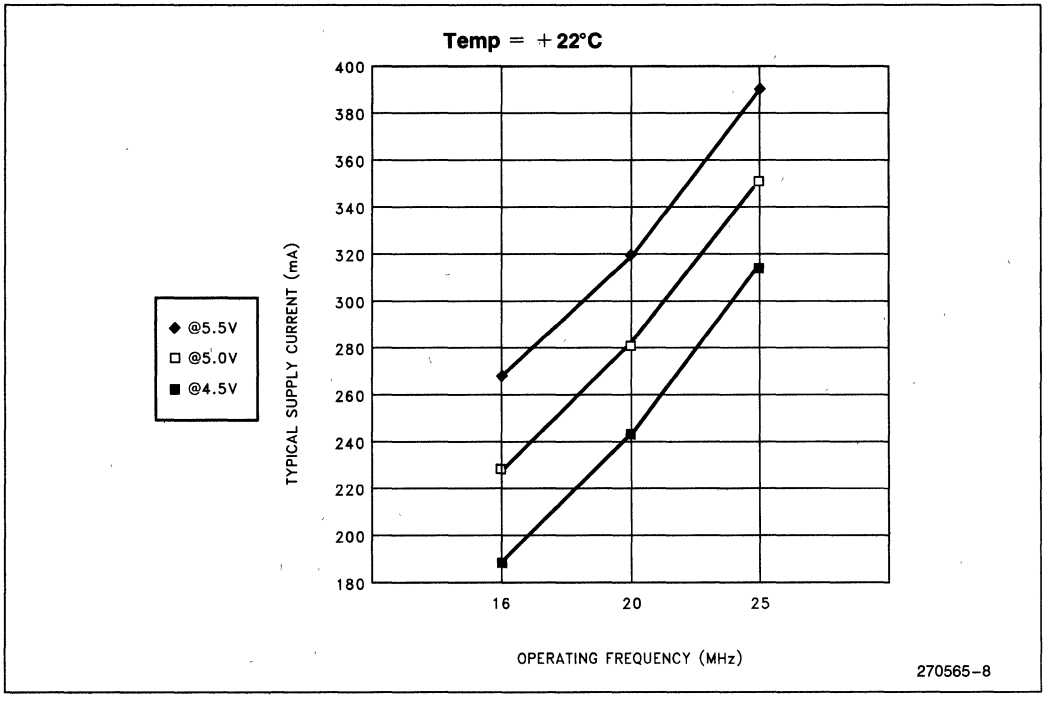


Figure 8. Typical Current vs Frequency (Room Temp)

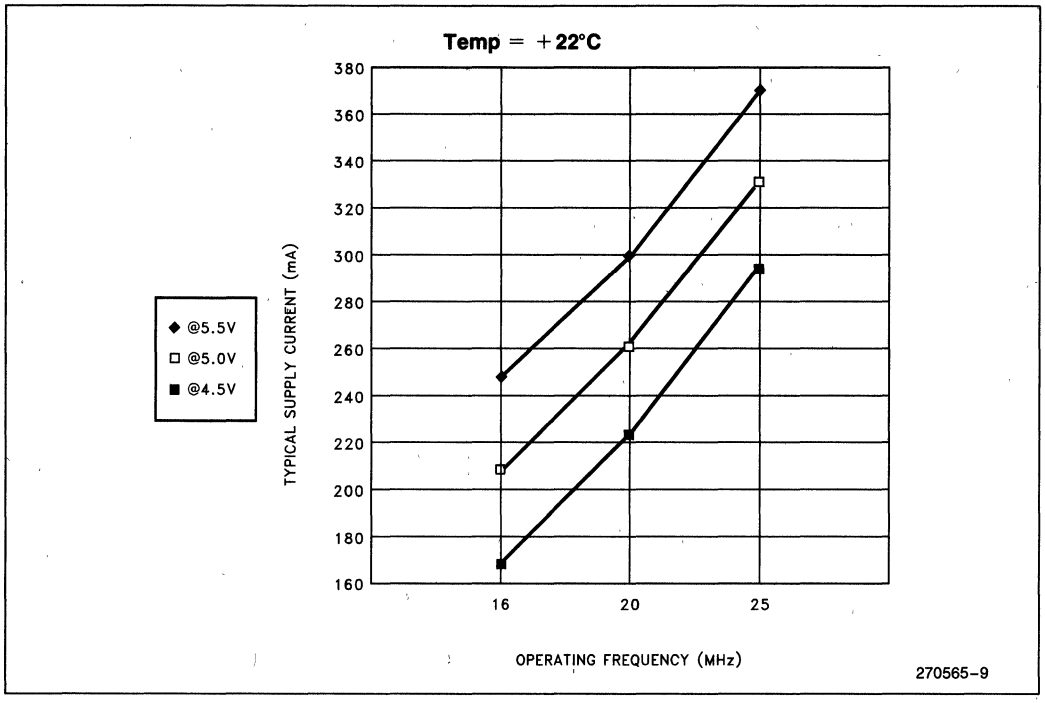


Figure 9. Typical Current vs Frequency (Hot Temp)

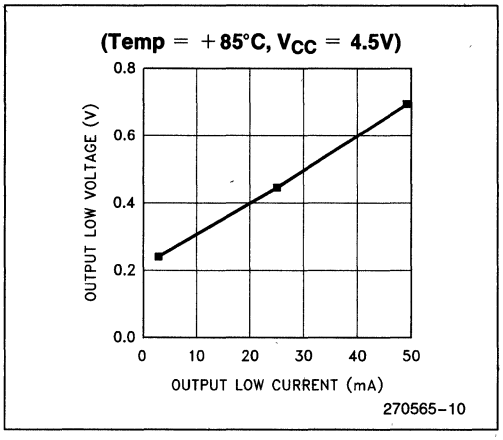


Figure 10. Worst-Case Voltage vs Output Current on Open-Drain Pins

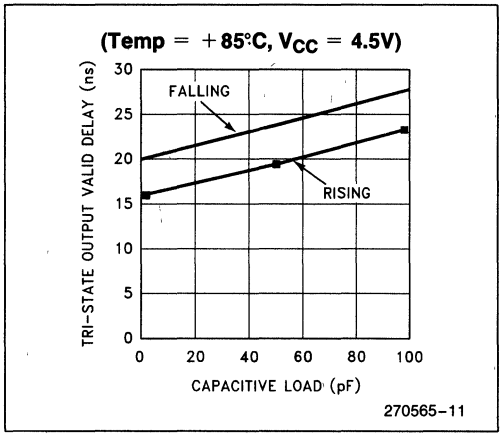
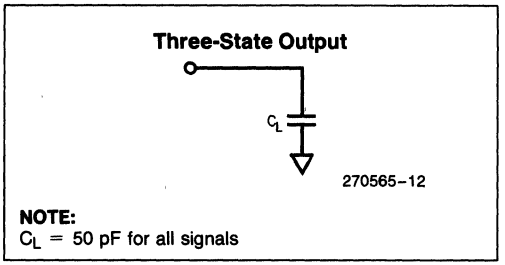


Figure 11. Capacitive Derating Curve

2.5 Test Load Circuit

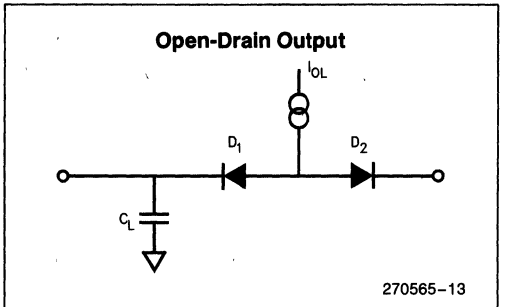
Figure 12 illustrates the load circuit used to test the 80960KB's three-state pins; Figure 13 shows the load circuit used to test the open drain outputs. The open drain test uses an active load circuit in the form of a matched diode bridge. Since the open-drain outputs sink current, only the I_{OL} legs of the bridge are necessary and the I_{OH} legs are not used. When the 80960KB driver under test is turned off, the output pin is pulled-up to V_{REF} (i.e., V_{OH}). Diode D_1 is turned off and the I_{OL} current source flows through diode D_2.

When the 80960KB open-drain driver under test is on, diode D_1 is also on and the voltage on the pin being tested drops to V_{OL}. Diode D_2 turns off and I_{OL} flows through diode D_1.



NOTE:
C_L = 50 pF for all signals

Figure 12. Test Load Circuit for Three-State Output Pins



NOTES:
I_{OL} Tested at 25 mA
$V_{REF} = V_{CC}$
D_1 and D_2 are matched
C_L = 50 pF for all signals

Figure 13. Test Load Circuit for Open-Drain Output Pins

3.0 ABSOLUTE MAXIMUM RATINGS

Operating Temperature
- PGA .....0°C to +85°C Case
- PQFP .....0°C to +100°C Case

Storage Temperature .....−65°C to +150°C

Voltage on Any Pin.....−0.5V to V_{CC} + 0.5V

Power Dissipation .....2.5W (25 MHz)

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

3.1 DC Characteristics

PGA: 80960KB (16 MHz) T_{CASE} = 0°C to +85°C, V_{CC} = 5V ±10%
80960KB (20 and 25 MHz) T_{CASE} = 0°C to +85°C, V_{CC} = 5V ±5%

PQFP: 80960KB (16 MHz) T_{CASE} = 0°C to +100°C, V_{CC} = 5V ±10%
80960KB (20 and 25 MHz) T_{CASE} = 0°C to +100°C, V_{CC} = 5V ±5%

Table 6. DC Characteristics

Symbol	Parameter	Min	Max	Units	Notes
V_{IL}	Input Low Voltage	−0.3	+0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.3	V	
V_{CL}	CLK2 Input Low Voltage	−0.3	+0.8	V	
V_{CH}	CLK2 Input High Voltage	0.55 V_{CC}	V_{CC} + 0.3	V	
V_{OL}	Output Low Voltage		0.45	V	(1, 2)
V_{OH}	Output High Voltage	2.4		V	(3, 4)
I_{CC}	Power Supply Current: 16 MHz 20 MHz 25 MHz		 315 360 420	 mA mA mA	 (5) (5) (5)
I_{LI}	Input Leakage Current		±15	µA	$0 \le V_{IN} \le V_{CC}$
I_{LO}	Output Leakage Current		±15	µA	$0.45 \le V_O \le V_{CC}$
C_{IN}	Input Capacitance		10	pF	f_C = 1 MHz(6)
C_O	Output Capacitance		12	pF	f_C = 1 MHz(6)
C_{CLK}	Clock Capacitance		10	pF	f_C = 1 MHz(6)

NOTES:
1. For three-state outputs, this parameter is measured at:
   - Addresss/Data .....4.0 mA
   - Controls .....5.0 mA
2. For open-drain outputs .....25 mA
3. This parameter is measured at:
   - Address/Data .....−1.0 mA
   - Controls .....−0.9 mA
   - $\overline{ALE}$ .....−5.0 mA
4. Not measured on open-drain outputs.
5. Measured at worst case frequency, V_{CC} and temperature, with device operating and outputs loaded to the test conditions in Figures 12 and 13. Figure 7, Figure 8 and Figure 9 indicate typical values.
6. Input, output and clock capacitance are not tested.

3.2 AC Specifications

This section describes the AC specifications for the 80960KB pins. All input and output timings are specified relative to the 1.5V level of the rising edge of CLK2. For output timings the specifications refer to the time it takes the signal to reach 1.5V.

For input timings the specifications refer to the time at which the signal reaches (for input setup) or leaves (for hold time) the TTL levels of LOW (0.8V) or HIGH (2.0V). All AC testing should be done with input voltages of 0.4V and 2.4V, except for the clock (CLK2), which should be tested with input voltages of 0.45V and 0.55 V_{CC}.

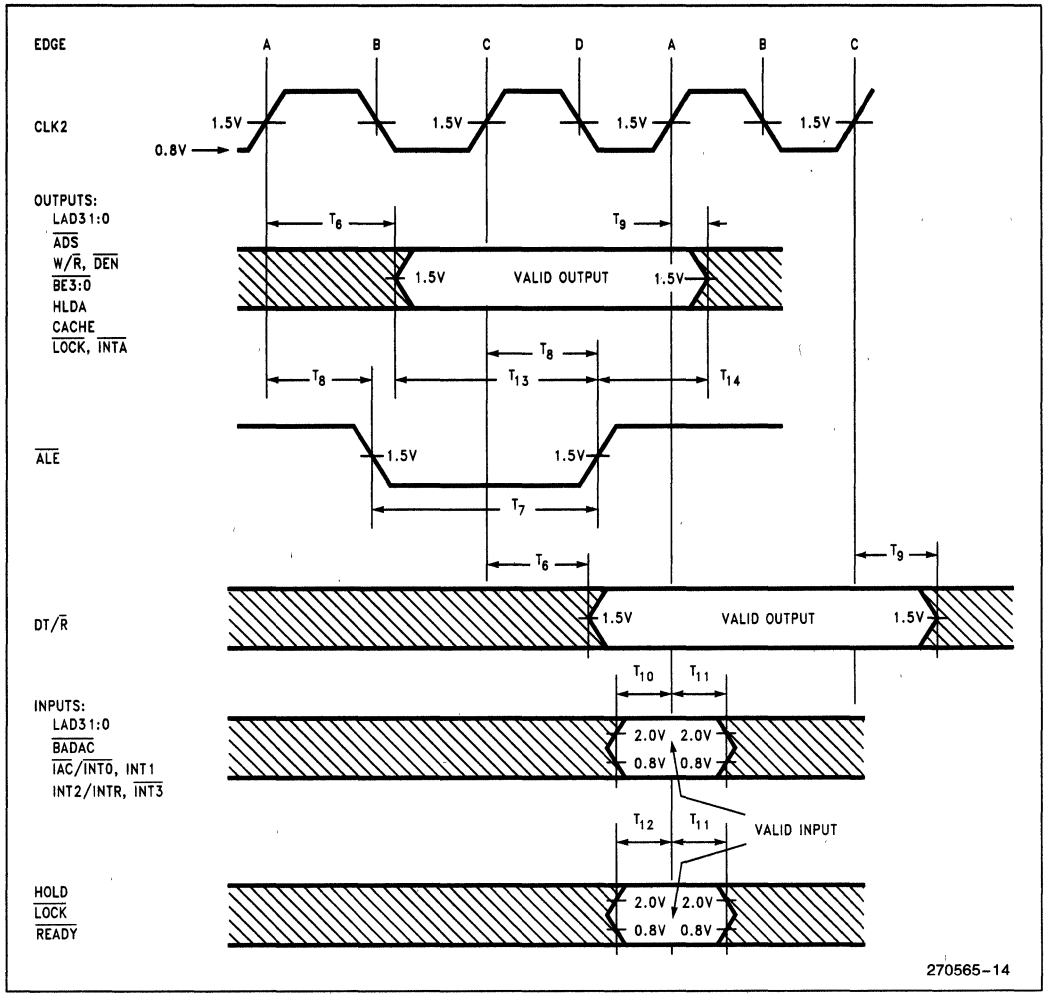


Figure 14. Drive Levels and Timing Relationships for 80960KB Signals

3.2.1 AC SPECIFICATION TABLES

Table 7. 80960KB AC Characteristics (16 MHz)

Symbol	Parameter	Min	Max	Units	Notes
INPUT CLOCK					
T_1	Processor Clock Period (CLK2)	31.25	125	ns	V_{IN} = 1.5V
T_2	Processor Clock Low Time (CLK2)	8		ns	V_{IL} = 10% Point = 1.2V
T_3	Processor Clock High Time (CLK2)	8		ns	V_{IH} = 90% Point = 0.1V + 0.5 V_{CC}
T_4	Processor Clock Fall Time (CLK2)		10	ns	V_{IN} = 90% Point to 10% Point(1)
T_5	Processor Clock Rise Time (CLK2)		10	ns	V_{IN} = 10% Point to 90% Point(1)
SYNCHRONOUS OUTPUTS					
T_6	Output Valid Delay	2	25	ns	
T_{6H}	HLDA Output Valid Delay	4	28	ns	
T_7	$\overline{ALE}$ Width	15		ns	
T_8	$\overline{ALE}$ Output Valid Delay	2	18	ns	
T_9	Output Float Delay	2	20	ns	(2)
T_{9H}	HLDA Output Float Delay	4	20	ns	(2)
SYNCHRONOUS INPUTS					
T_{10}	Input Setup 1	3		ns	(3)
T_{11}	Input Hold	5		ns	(3)
T_{11H}	HOLD Input Hold	4		ns	(3)
T_{12}	Input Setup 2	8		ns	(3)
T_{13}	Setup to $\overline{ALE}$ Inactive	10		ns	
T_{14}	Hold after $\overline{ALE}$ Inactive	8		ns	
T_{15}	Reset Hold	3		ns	(3)
T_{16}	Reset Setup	5		ns	(3)
T_{17}	Reset Width	1281		ns	41 CLK2 Periods Minimum

NOTES:
1. Clock rise and tall times are not tested.
2. A float condition occurs when the maximum output current becomes less than I_{LO}. Float delay is not tested; however, it should not be longer than the valid delay.
3. LAD31:0, $\overline{BADAC}$, HOLD, $\overline{LOCK}$ and $\overline{READY}$ are synchronous inputs. $\overline{IAC}/\overline{INT}_0$, INT_1, INT_2/INT_R and $\overline{INT}_3$ may be synchronous or asynchronous.

Table 8. 80960KB AC Characteristics (20 MHz)

Symbol	Parameter	Min	Max	Units	Notes
INPUT CLOCK					
T_1	Processor Clock Period (CLK2)	25	125	ns	V_{IN} = 1.5V
T_2	Processor Clock Low Time (CLK2)	6		ns	V_{IL} = 10% Point = 1.2V
T_3	Processor Clock High Time (CLK2)	6		ns	V_{IH} = 90% Point = 0.1V + 0.5 V_{CC}
T_4	Processor Clock Fall Time (CLK2)		10	ns	V_{IN} = 90% Point to 10% Point(1)
T_5	Processor Clock Rise Time (CLK2)		10	ns	V_{IN} = 10% Point to 90% Point(1)
SYNCHRONOUS OUTPUTS					
T_6	Output Valid Delay	2	20	ns	
T_{6H}	HLDA Output Valid Delay	4	23	ns	
T_7	$\overline{ALE}$ Width	12		ns	
T_8	$\overline{ALE}$ Output Valid Delay	2	18	ns	
T_9	Output Float Delay	2	20	ns	(2)
T_{9H}	HLDA Output Float Delay	4	20	ns	(2)
SYNCHRONOUS INPUTS					
T_{10}	Input Setup 1	3		ns	(3)
T_{11}	Input Hold	5		ns	(3)
T_{11H}	HOLD Input Hold	4		ns	(3)
T_{12}	Input Setup 2	7		ns	(3)
T_{13}	Setup to $\overline{ALE}$ Inactive	10		ns	
T_{14}	Hold after $\overline{ALE}$ Inactive	8		ns	
T_{15}	Reset Hold	3		ns	
T_{15}	Reset Setup	5		ns	
T_{17}	Reset Width	1025		ns	41 CLK2 Periods Minimum

NOTES:
1. Clock rise and fall times are not tested.
2. A float condition occurs when the maximum output current becomes less than I_{LO}. Float delay is not tested; however, it should not be longer than the valid delay.
3. LAD31:0, $\overline{BADAC}$, HOLD, $\overline{LOCK}$ and $\overline{READY}$ are synchronous inputs. $\overline{IAC}/\overline{INT_0}$, INT_1, INT_2/INT_R and $\overline{INT_3}$ may be synchronous or asynchroncous.

Table 9. 80960KB AC Characteristics (25 MHz)

Symbol	Parameter	Min	Max	Units	Notes
INPUT CLOCK					
T_1	Processor Clock Period (CLK2)	20	125	ns	V_{IN} = 1.5V
T_2	Processor Clock Low Time (CLK2)	5		ns	V_{IL} = 10% Point = 1.2V
T_3	Processor Clock High Time (CLK2)	5		ns	V_{IH} = 90% Point = 0.1V + 0.5 V_{CC}
T_4	Processor Clock Fall Time (CLK2)		10	ns	V_{IN} = 90% Point to 10% Point(1)
T_5	Processor Clock Rise Time (CLK2)		10	ns	V_{IN} = 10% Point to 90% Point(1)
SYNCHRONOUS OUTPUTS					
T_6	Output Valid Delay	2	18	ns	
T_{6H}	HLDA Output Valid Delay	4	23	ns	
T_7	$\overline{ALE}$ Width	12		ns	
T_8	$\overline{ALE}$ Output Valid Delay	2	18	ns	
T_9	Output Float Delay	2	18	ns	(2)
T_{9H}	HLDA Output Float Delay	4	20	ns	(2)
SYNCHRONOUS INPUTS					
T_{10}	Input Setup 1	3		ns	(3)
T_{11}	Input Hold	5		ns	(3)
T_{11H}	HOLD Input Hold	4		ns	
T_{12}	Input Setup 2	7		ns	
T_{13}	Setup to $\overline{ALE}$ Inactive	8		ns	
T_{14}	Hold after $\overline{ALE}$ Inactive	8		ns	
T_{15}	Reset Hold	3		ns	
T_{16}	Reset Setup	5		ns	
T_{17}	Reset Width	820		ns	41 CLK2 Periods Minimum

NOTES:
1. Clock rise and tall times are not tested.
2. A float condition occurs when the maximum output current becomes less than I_{LO}. Float delay is not tested; however, it should not be longer than the valid delay.
3. LAD31:0, $\overline{BADAC}$, HOLD, $\overline{LOCK}$ and $\overline{READY}$ are synchronous inputs. $\overline{IAC}/\overline{INT_0}$, INT_1, INT_2/INT_R and $\overline{INT_3}$ may be synchronous or asynchronous.

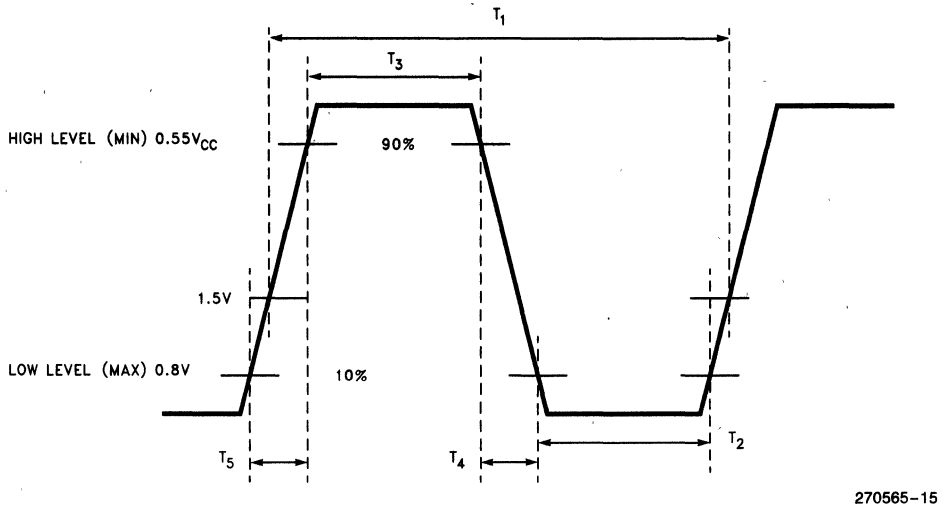
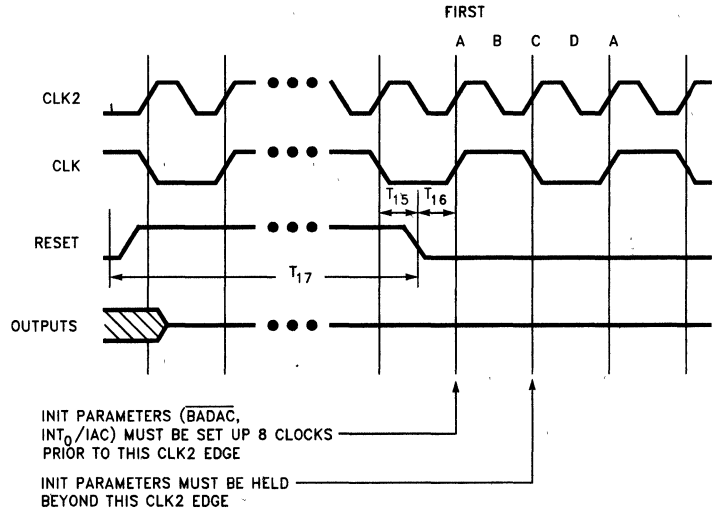


Figure 15. Processor Clock Pulse (CLK2)



NOTES:
T_{15} = RESET HOLD
T_{16} = RESET SETUP
T_{17} = RESET WIDTH

Figure 16. RESET Signal Timing

3.3 Mechanical Data

3.3.1 PACKAGING

The 80960KB is available in two package types:

- 132-lead ceramic pin-grid array (PGA). Pins are arranged 0.100 inch (2.54 mm) center-to-center, in a 14 by 14 matrix, three rows around (see Figure 17).
- 132-lead plastic quad flat pack (PQFP). This package uses fine-pitch gull wing leads arranged in a single row along the package perimeter with 0.025 inch (0.64 mm) spacing (see Figure 20).

Dimensions for both package types are given in the Intel *Packaging* handbook (Order #240800).

3.3.2 PIN ASSIGNMENT

The PGA and PQFP have different pin assignments. Figure 18 shows the view from the PGA bottom (pins facing up) and Figure 19 shows a view from the PGA top (pins facing down). Figure 20 shows the PQFP package; Figure 21 shows the PQFP pinout with signal names. Notice that the pins are numbered in order from 1 to 132 around the package perimeter. Table 10 and Table 11 list the function of each PGA pin; Table 12 and Table 13 list the function of each PQFP pin.

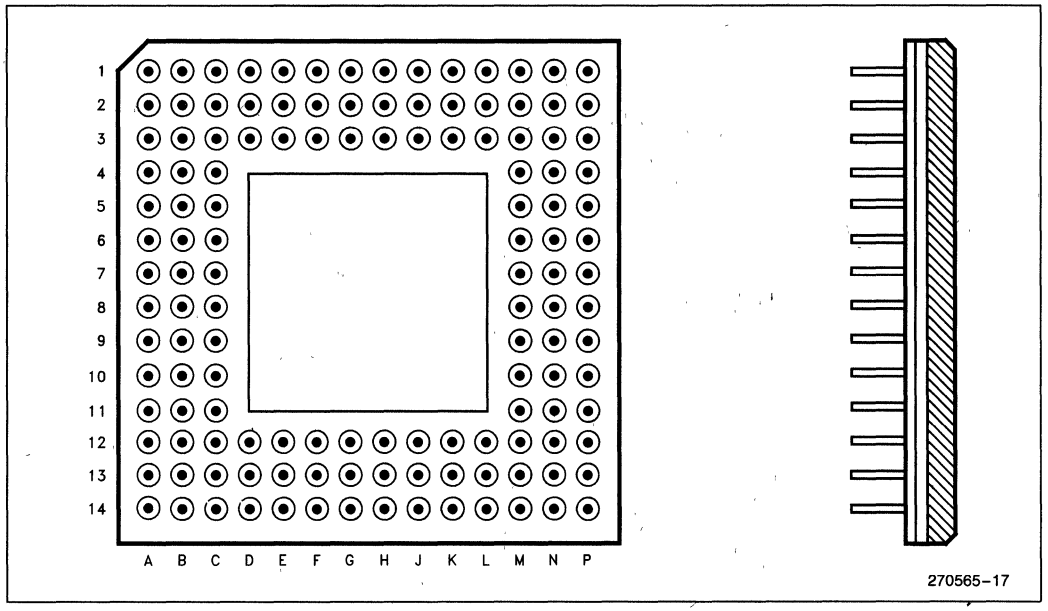


Figure 17. 132-Lead Pin-Grid Array (PGA) Package

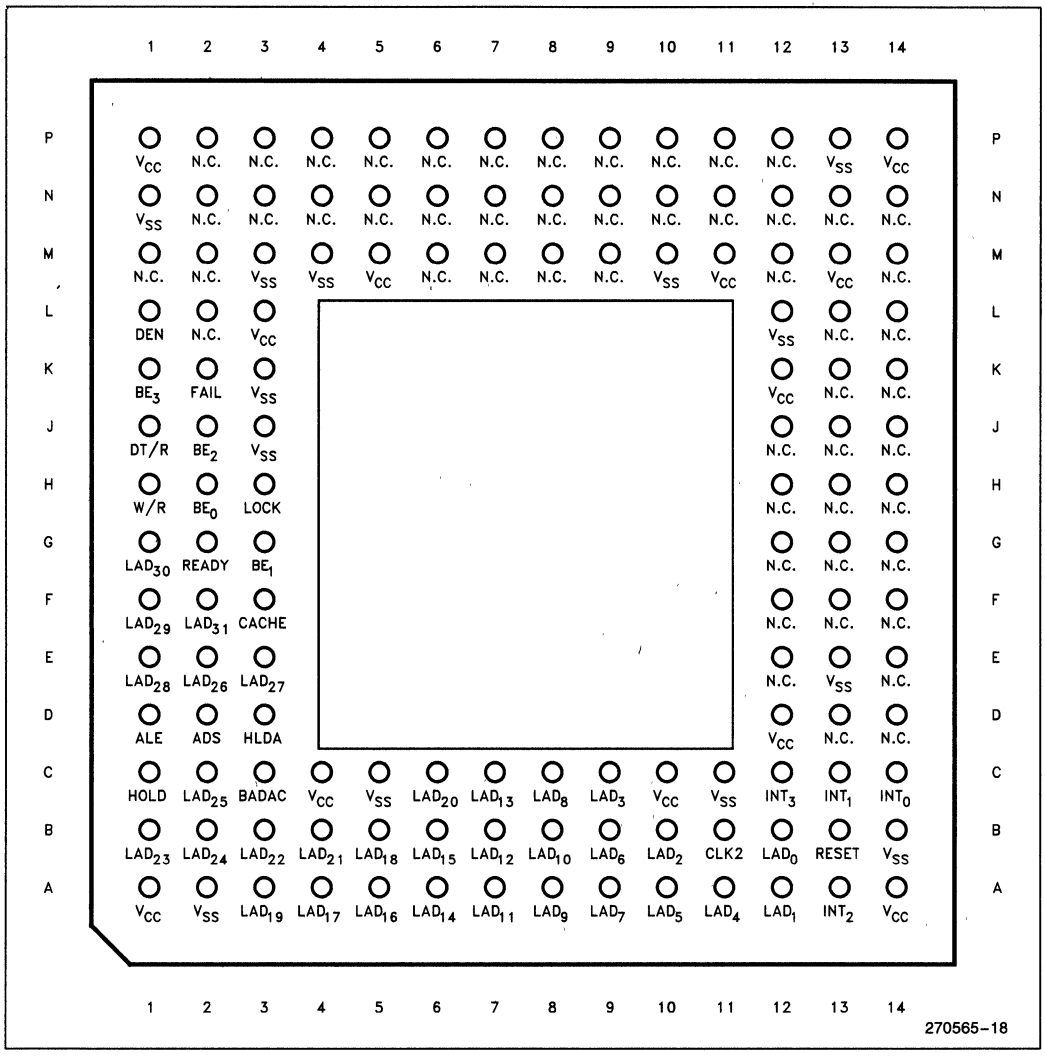


Figure 18. 80960KB PGA Pinout—View from Bottom (Pins Facing Up)

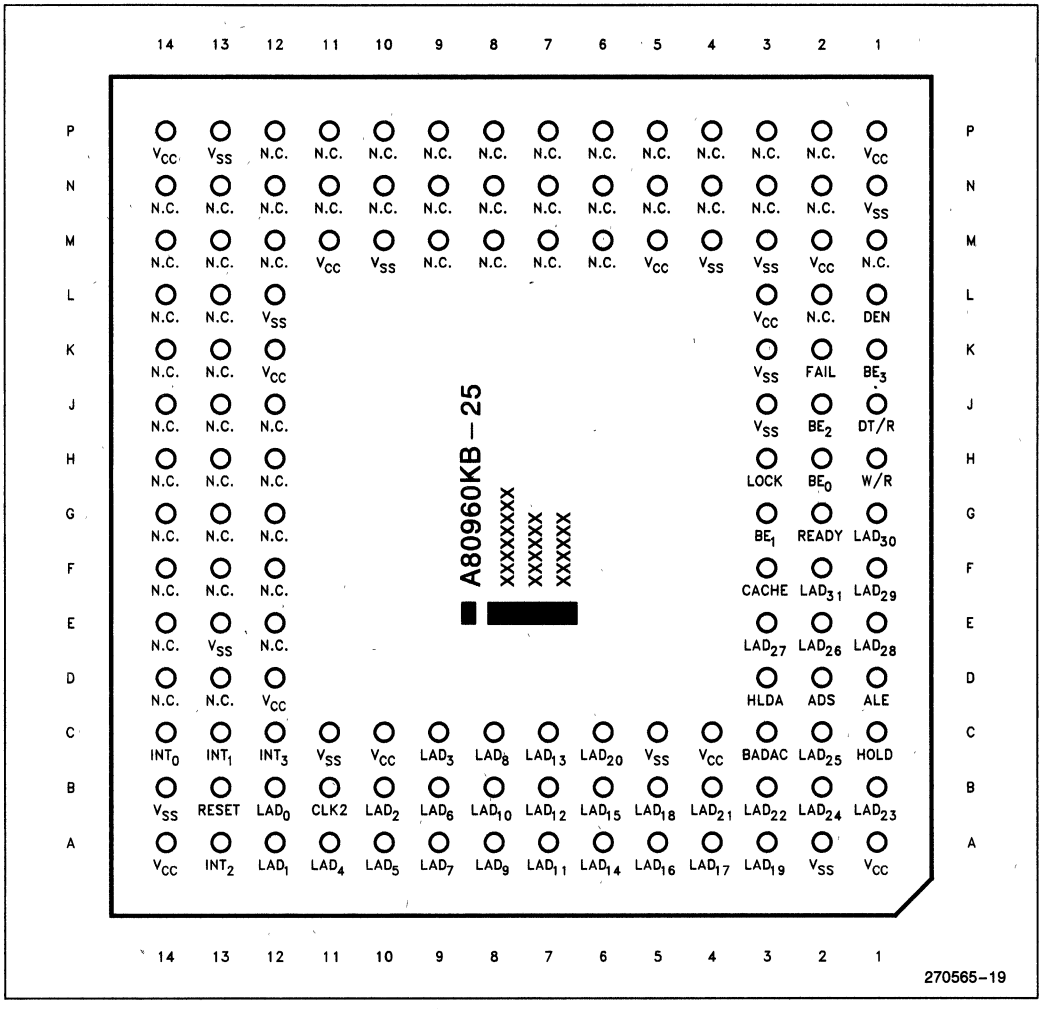


Figure 19. 80960KB PGA Pinout—View from Top (Pins Facing Down)

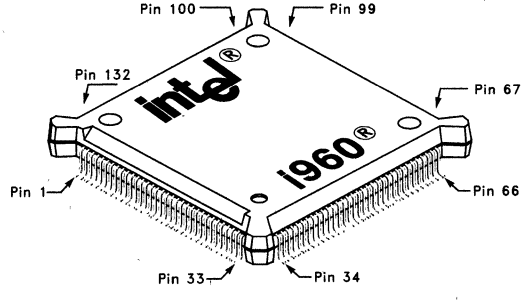


Figure 20. 80960KB 132-Lead Plastic Quad Flat-Pack (PQFP) Package

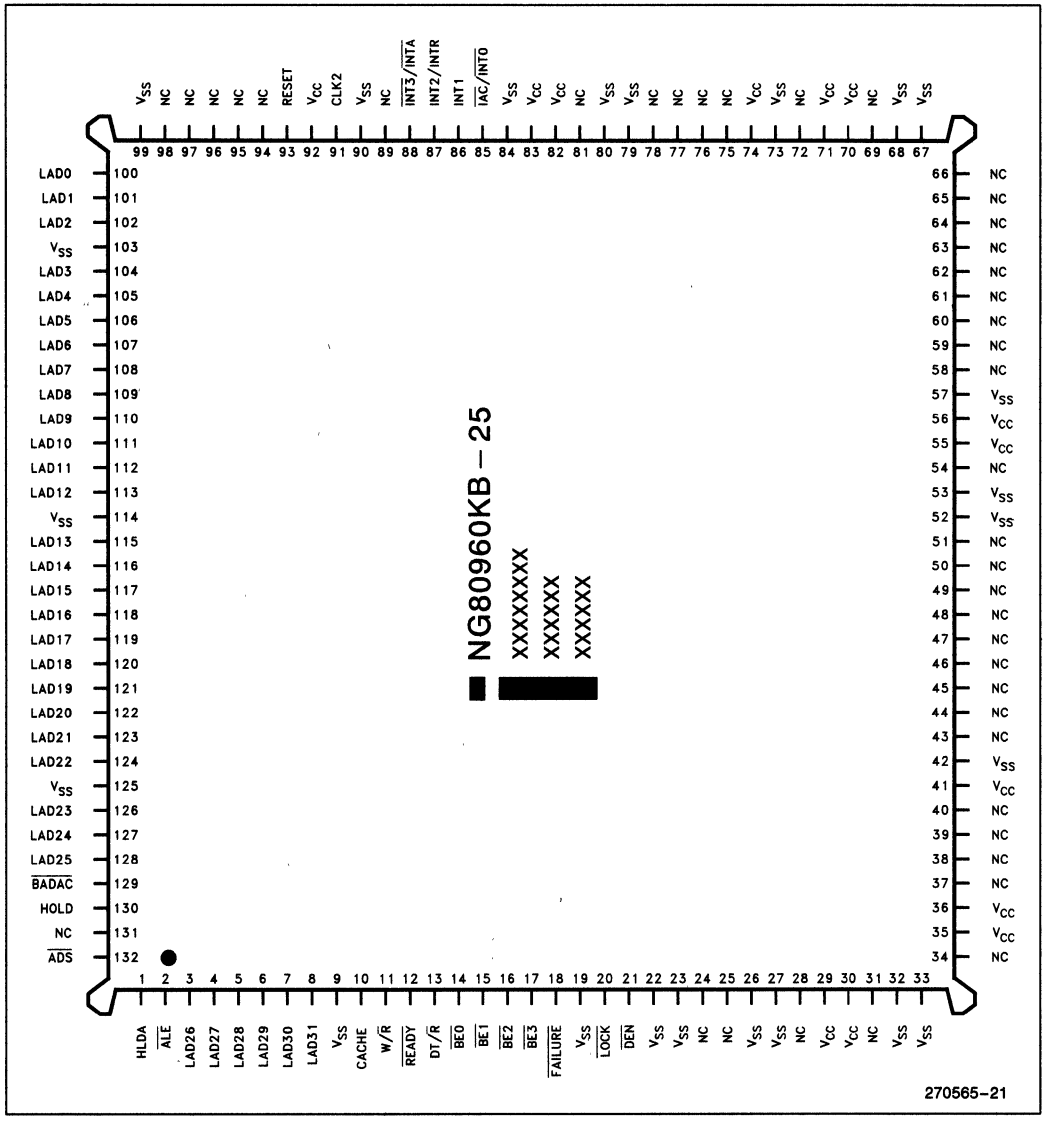


Figure 21. PQFP Pinout—View from Top

3.4 Pinout

Table 10. 80960KB PGA Pinout—In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	V_{CC}	C6	LAD_{20}	H1	W/$\overline{R}$	M10	V_{SS}
A2	V_{SS}	C7	LAD_{13}	H2	$\overline{BE_0}$	M11	V_{CC}
A3	LAD_{19}	C8	LAD_8	H3	$\overline{LOCK}$	M12	N.C.
A4	LAD_{17}	C9	LAD_3	H12	N.C.	M13	N.C.
A5	LAD_{16}	C10	V_{CC}	H13	N.C.	M14	N.C.
A6	LAD_{14}	C11	V_{SS}	H14	N.C.	N1	V_{SS}
A7	LAD_{11}	C12	$\overline{INT_3}$/$\overline{INTA}$	J1	DT/$\overline{R}$	N2	N.C.
A8	LAD_9	C13	INT_1	J2	$\overline{BE_2}$	N3	N.C.
A9	LAD_7	C14	$\overline{IAC}$/$\overline{INT_0}$	J3	V_{SS}	N4	N.C.
A10	LAD_5	D1	$\overline{ALE}$	J12	N.C.	N5	N.C.
A11	LAD_4	D2	$\overline{ADS}$	J13	N.C.	N6	N.C.
A12	LAD_1	D3	HLDA	J14	N.C.	N7	N.C.
A13	INT_2/INTR	D12	V_{CC}	K1	$\overline{BE_3}$	N8	N.C.
A14	V_{CC}	D13	N.C.	K2	$\overline{FAILURE}$	N9	N.C.
B1	LAD_{23}	D14	N.C.	K3	V_{SS}	N10	N.C.
B2	LAD_{24}	E1	LAD_{28}	K12	V_{CC}	N11	N.C.
B3	LAD_{22}	E2	LAD_{26}	K13	N.C.	N12	N.C.
B4	LAD_{21}	E3	LAD_{27}	K14	N.C.	N13	N.C.
B5	LAD_{18}	E12	N.C.	L1	$\overline{DEN}$	N14	N.C.
B6	LAD_{15}	E13	V_{SS}	L2	N.C.	P1	V_{CC}
B7	LAD_{12}	E14	N.C.	L3	V_{CC}	P2	N.C.
B8	LAD_{10}	F1	LAD_{29}	L12	V_{SS}	P3	N.C.
B9	LAD_6	F2	LAD_{31}	L13	N.C.	P4	N.C.
B10	LAD_2	F3	CACHE	L14	N.C.	P5	N.C.
B11	CLK2	F12	N.C.	M1	N.C.	P6	N.C.
B12	LAD_0	F13	N.C.	M2	V_{CC}	P7	N.C.
B13	RESET	F14	N.C.	M3	V_{SS}	P8	N.C.
B14	V_{SS}	G1	LAD_{30}	M4	V_{SS}	P9	N.C.
C1	HOLD	G2	$\overline{READY}$	M5	V_{CC}	P10	N.C.
C2	LAD_{25}	G3	$\overline{BE_1}$	M6	N.C.	P11	N.C.
C3	$\overline{BADAC}$	G12	N.C.	M7	N.C.	P12	N.C.
C4	V_{CC}	G13	N.C.	M8	N.C.	P13	V_{SS}
C5	V_{SS}	G14	N.C.	M9	N.C.	P14	V_{CC}

NOTE:
Do not connect any external logic to any pins marked N.C.

Table 11. 80960KB PGA Pinout—In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
$\overline{ADS}$	D2	LAD_{15}	B6	N.C.	J14	N.C.	P9
$\overline{ALE}$	D1	LAD_{16}	A5	N.C.	K13	N.C.	P10
$\overline{BADAC}$	C3	LAD_{17}	A4	N.C.	K14	N.C.	P11
$\overline{BE}_0$	H2	LAD_{18}	B5	N.C.	L13	N.C.	P12
$\overline{BE}_1$	G3	LAD_{19}	A3	N.C.	L14	N.C.	L2
$\overline{BE}_2$	J2	LAD_{20}	C6	N.C.	M1	$\overline{READY}$	G2
$\overline{BE}_3$	K1	LAD_{21}	B4	N.C.	M6	RESET	B13
CACHE	F3	LAD_{22}	B3	N.C.	M7	V_{CC}	A1
CLK2	B11	LAD_{23}	B1	N.C.	M8	V_{CC}	A14
$\overline{DEN}$	L1	LAD_{24}	B2	N.C.	M9	V_{CC}	C4
DT/$\overline{R}$	J1	LAD_{25}	C2	N.C.	M12	V_{CC}	C10
$\overline{FAILURE}$	K2	LAD_{26}	E2	N.C.	M13	V_{CC}	D12
HLDA	D3	LAD_{27}	E3	N.C.	M14	V_{CC}	K12
HOLD	C1	LAD_{28}	E1	N.C.	N2	V_{CC}	L3
$\overline{IAC}/\overline{INT}_0$	C14	LAD_{29}	F1	N.C.	N3	V_{CC}	M2
INT_1	C13	LAD_{30}	G1	N.C.	N4	V_{CC}	M5
INT_2/INTR	A13	LAD_{31}	F2	N.C.	N5	V_{CC}	M11
$\overline{INT}_3/\overline{INTA}$	C12	$\overline{LOCK}$	H3	N.C.	N6	V_{CC}	P1
LAD_0	B12	N.C.	D13	N.C.	N7	V_{CC}	P14
LAD_1	A12	N.C.	D14	N.C.	N8	V_{SS}	A2
LAD_2	B10	N.C.	E12	N.C.	N9	V_{SS}	B14
LAD_3	C9	N.C.	E14	N.C.	N10	V_{SS}	C5
LAD_4	A11	N.C.	F12	N.C.	N11	V_{SS}	C11
LAD_5	A10	N.C.	F13	N.C.	N12	V_{SS}	E11
LAD_6	B9	N.C.	F14	N.C.	N13	V_{SS}	J3
LAD_7	A9	N.C.	G12	N.C.	N14	V_{SS}	K3
LAD_8	C8	N.C.	G13	N.C.	P2	V_{SS}	L12
LAD_9	A8	N.C.	G14	N.C.	P3	V_{SS}	M3
LAD_{10}	B8	N.C.	H12	N.C.	P4	V_{SS}	M4
LAD_{11}	A7	N.C.	H13	N.C.	P5	V_{SS}	M10
LAD_{12}	B7	N.C.	H14	N.C.	P6	V_{SS}	N1
LAD_{13}	C7	N.C.	J12	N.C.	P7	V_{SS}	P13
LAD_{14}	A6	N.C.	J13	N.C.	P8	W/$\overline{R}$	H1

NOTE:
Do not connect any external logic to any pins marked N.C.

Table 12. 80960KB PQFP Pinout—In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	HLDA	34	N.C.	67	V_{SS}	100	LAD_0
2	$\overline{ALE}$	35	V_{CC}	68	V_{SS}	101	LAD_1
3	LAD_{26}	36	V_{CC}	69	N.C.	102	LAD_2
4	LAD_{27}	37	N.C.	70	V_{CC}	103	V_{SS}
5	LAD_{28}	38	N.C.	71	V_{CC}	104	LAD_3
6	LAD_{29}	39	N.C.	72	N.C.	105	LAD_4
7	LAD_{30}	40	N.C.	73	V_{SS}	106	LAD_5
8	LAD_{31}	41	V_{CC}	74	V_{CC}	107	LAD_6
9	V_{SS}	42	V_{SS}	75	N.C.	108	LAD_7
10	CACHE	43	N.C.	76	N.C.	109	LAD_8
11	W/$\overline{R}$	44	N.C.	77	N.C.	110	LAD_9
12	$\overline{READY}$	45	N.C.	78	N.C.	111	LAD_{10}
13	DT/$\overline{R}$	46	N.C.	79	V_{SS}	112	LAD_{11}
14	$\overline{BE}_0$	47	N.C.	80	V_{SS}	113	LAD_{12}
15	$\overline{BE}_1$	48	N.C.	81	N.C.	114	V_{SS}
16	$\overline{BE}_2$	49	N.C.	82	V_{CC}	115	LAD_{13}
17	$\overline{BE}_3$	50	N.C.	83	V_{CC}	116	LAD_{14}
18	$\overline{FAILURE}$	51	N.C.	84	V_{SS}	117	LAD_{15}
19	V_{SS}	52	V_{SS}	85	$\overline{IAC}/\overline{INT}_0$	118	LAD_{16}
20	$\overline{LOCK}$	53	V_{SS}	86	INT_1	119	LAD_{17}
21	$\overline{DEN}$	54	N.C.	87	INT_2/INTR	120	LAD_{18}
22	V_{SS}	55	V_{CC}	88	$\overline{INT}_3/\overline{INTA}$	121	LAD_{19}
23	V_{SS}	56	V_{CC}	89	N.C.	122	LAD_{20}
24	N.C.	57	V_{SS}	90	V_{SS}	123	LAD_{21}
25	N.C.	58	N.C.	91	CLK2	124	LAD_{22}
26	V_{SS}	59	N.C.	92	V_{CC}	125	V_{SS}
27	V_{SS}	60	N.C.	93	RESET	126	LAD_{23}
28	N.C.	61	N.C.	94	N.C.	127	LAD_{24}
29	V_{CC}	62	N.C.	95	N.C.	128	LAD_{25}
30	V_{CC}	63	N.C.	96	N.C.	129	$\overline{BADAC}$
31	N.C.	64	N.C.	97	N.C.	130	HOLD
32	V_{SS}	65	N.C.	98	N.C.	131	N.C.
33	V_{SS}	66	N.C.	99	V_{SS}	132	$\overline{ADS}$

NOTE:
Do not connect any external logic to any pins marked N.C.

Table 13. 80960KB PQFP Pinout—In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
$\overline{ADS}$	132	LAD_{15}	117	N.C.	49	V_{CC}	41
$\overline{ALE}$	2	LAD_{16}	118	N.C.	50	V_{CC}	55
$\overline{BADAC}$	129	LAD_{17}	119	N.C.	51	V_{CC}	56
$\overline{BE_0}$	14	LAD_{18}	120	N.C.	54	V_{CC}	70
$\overline{BE_1}$	15	LAD_{19}	121	N.C.	58	V_{CC}	71
$\overline{BE_2}$	16	LAD_{20}	122	N.C.	59	V_{CC}	74
$\overline{BE_3}$	17	LAD_{21}	123	N.C.	60	V_{CC}	82
CACHE	10	LAD_{22}	124	N.C.	61	V_{CC}	83
CLK2	91	LAD_{23}	126	N.C.	62	V_{CC}	92
$\overline{DEN}$	21	LAD_{24}	127	N.C.	63	V_{SS}	9
DT/$\overline{R}$	13	LAD_{25}	128	N.C.	64	V_{SS}	19
$\overline{FAILURE}$	18	LAD_{26}	3	N.C,	65	V_{SS}	22
HLDA	1	LAD_{27}	4	N.C.	66	V_{SS}	23
HOLD	130	LAD_{28}	5	N.C.	69	V_{SS}	26
$\overline{IAC}/\overline{INT_0}$	85	LAD_{29}	6	N.C.	72	V_{SS}	27
INT_1	86	LAD_{30}	7	N.C.	75	V_{SS}	32
INT_2/INTR	87	LAD_{31}	8	N.C.	76	V_{SS}	33
$\overline{INT_3}/\overline{INTA}$	88	$\overline{LOCK}$	20	N.C.	77	V_{SS}	42
LAD_0	100	N.C.	24	N.C.	78	V_{SS}	52
LAD_1	101	N.C.	25	N.C.	81	V_{SS}	53
LAD_2	102	N.C.	28	N.C.	89	V_{SS}	57
LAD_3	104	N.C.	31	N.C.	94	V_{SS}	67
LAD_4	105	N.C.	34	N.C.	95	V_{SS}	68
LAD_5	106	N.C.	37	N.C.	96	V_{SS}	73
LAD_6	107	N.C.	38	N.C.	97	V_{SS}	79
LAD_7	108	N.C.	39	N.C.	98	V_{SS}	80
LAD_8	109	N.C.	40	N.C.	131	V_{SS}	84
LAD_9	110	N.C.	43	$\overline{READY}$	12	V_{SS}	90
LAD_{10}	111	N.C.	44	RESET	93	V_{SS}	99
LAD_{11}	112	N.C.	45	V_{CC}	29	V_{SS}	103
LAD_{12}	113	N.C.	46	V_{CC}	30	V_{SS}	114
LAD_{13}	115	N.C.	47	V_{CC}	35	V_{SS}	125
LAD_{14}	116	N.C.	48	V_{CC}	36	W/$\overline{R}$	11

NOTE:
Do not connect any external logic to any pins marked N.C.

3.4.1 PACKAGE THERMAL SPECIFICATION

The 80960KB is specified for operation when case temperatures within the range 0°C to 85°C (PGA) or 0°C to 100°C (PQFP). Measure case temperature at the top center of the package. Ambient temperature can be calculated from:

$$T_J = T_C + P * \theta_{JC}$$
$$T_A = T_J + P * \theta_{JA}$$
$$T_C = T_A + P * [\theta_{JA} - \theta_{JC}]$$

Values for θ_{JA} and θ_{JC} for various airflows are given in Table 14 for the PGA package and in Table 15 for the PQFP package. The PGA's θ_{JA} can be reduced by adding a heatsink. For the PQFP, however, a heatsink is not generally used since the device is intended to be surface mounted.

Maximum allowable ambient temperature (T_A) permitted without exceeding T_C is shown by the graphs in Figures 23, 24, 25 and 26. The curves assume the maximum permitted supply current (I_{CC}) at each speed, V_{CC} of +5.0V and a T_{CASE} of +85°C (PGA) or +100°C (PQFP).

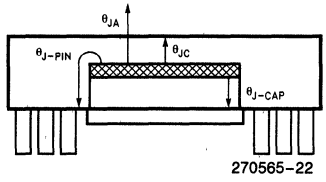
If the 80960KB is to be used in a harsh environment where the ambient temperature may exceed the limits for the normal commercial part, consider using an extended temperature device. These components are designated by the prefix "TA" and are available at 16, 20 and 25 MHz in the ceramic PGA package. Extended operating temperature range is −40°C to +125°C (case).

Figure 26 shows the maximum allowable ambient temperature for the 20 MHz extended temperature TA80960KB at various airflows. The curve assumes an I_{CC} of 420 mA, V_{CC} of 5.0V and a T_{CASE} of +125°C.

Table 14. 80960KB PGA Package Thermal Characteristics

Thermal Resistance—°C/Watt

Parameter	Airflow—ft./min (m/sec)						
	0 (0)	50 (0.25)	100 (0.50)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)
θ Junction-to-Case	2	2	2	2	2	2	2
θ Case-to-Ambient (No Heatsink)	19	18	17	15	12	10	9
θ Case-to-Ambient (Omnidirectional Heatsink)	16	15	14	12	9	7	6
θ Case-to-Ambient (Unidirectional Heatsink)	15	14	13	11	8	6	5



270565-22

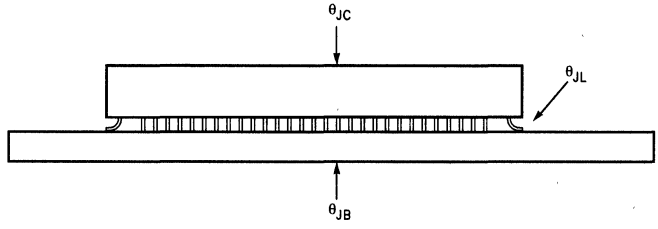
NOTES:
1. This table applies to 80960KB PGA plugged into socket or soldered directly to board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$
3. $\theta_{J\text{-}CAP}$ = 4°C/W (approx.)
   $\theta_{J\text{-}PIN}$ = 4°C/W (inner pins) (approx.)
   $\theta_{J\text{-}PIN}$ = 8°C/W (outer pins) (approx.)

Table 15. 80960KB PQFP Package Thermal Characteristics

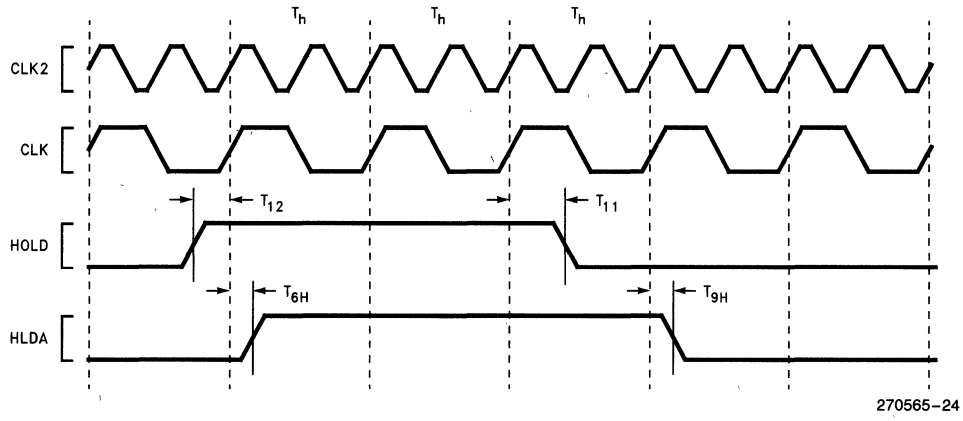
Thermal Resistance—°C/Watt

Parameter	Airflow—ft./min (m/sec)						
	0 (0)	50 (0.25)	100 (0.50)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)
θ Junction-to-Case	9	9	9	9	9	9	9
θ Case-to-Ambient (No Heatsink)	22	19	18	16	11	9	8

NOTES:
1. This table applies to 80960KB PQFP soldered directly to board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$
3. θ_{JL} = 18°C/W (approx.)
   θ_{JB} = 18°C/W (approx.)



270565-23



270565-24

Figure 22. HOLD Timing

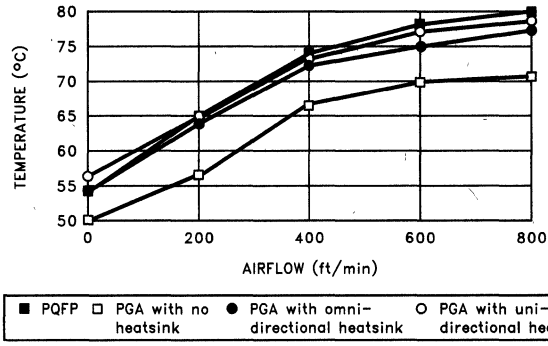


Figure 23. 16 MHz Maximum Allowable Ambient Temperature

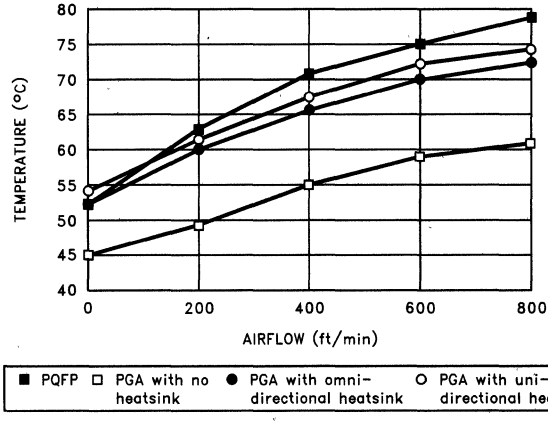


Figure 24. 20 MHz Maximum Allowable Ambient Temperature

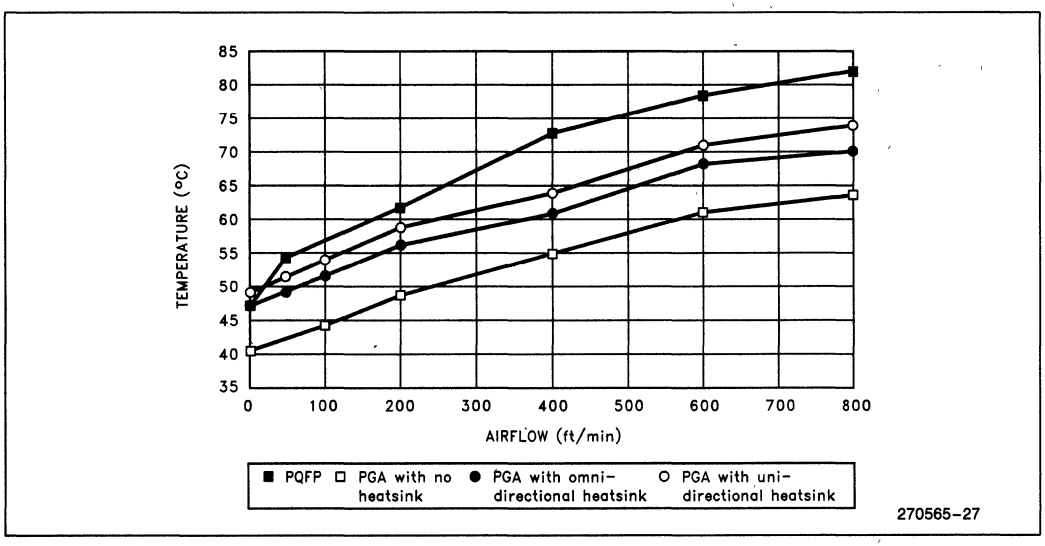


Figure 25. 25 MHz Maximum Allowable Ambient Temperature

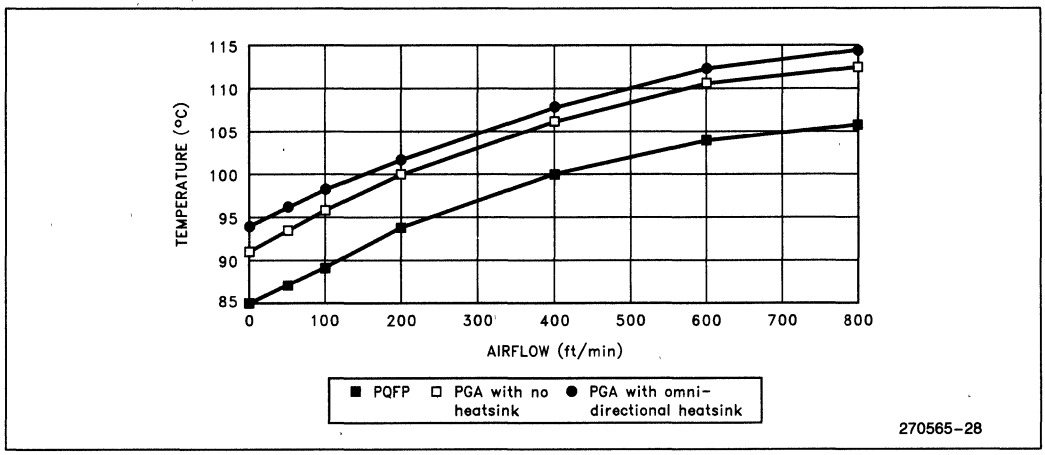


Figure 26. Maximum Allowable Ambient Temperature for the Extended Temperature TA-80960KB 20 MHz in PGA Package

3.5 Waveforms

Figures 27, 28, 29 and 30 show the waveforms for various transactions on the 80960KB's local bus.

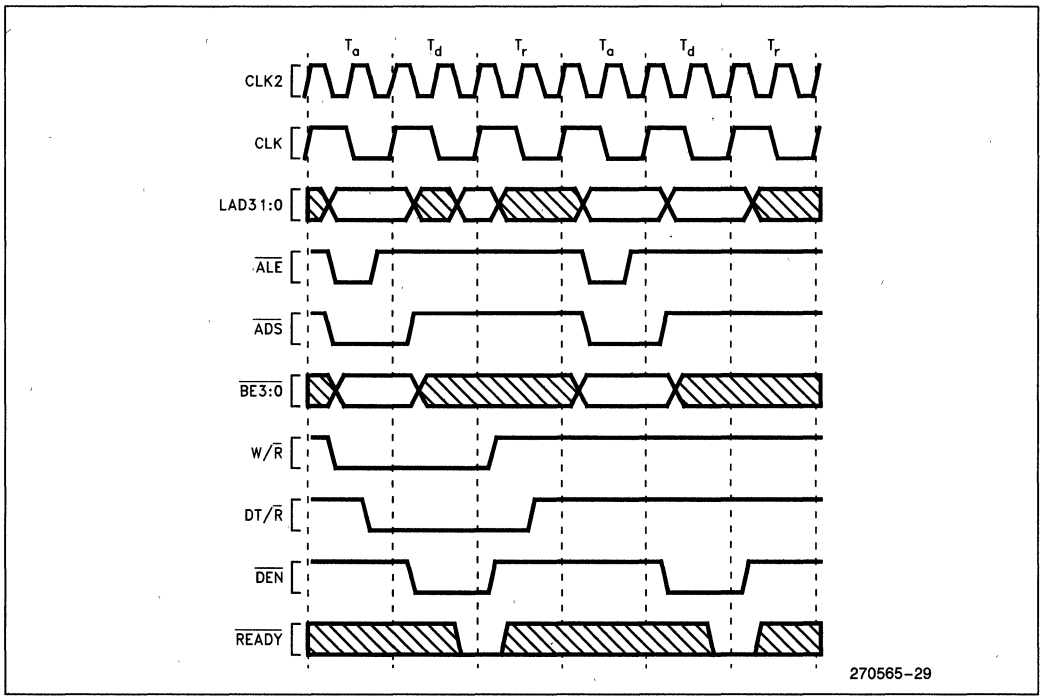


Figure 27. Non-Burst Read and Write Transactions without Wait States

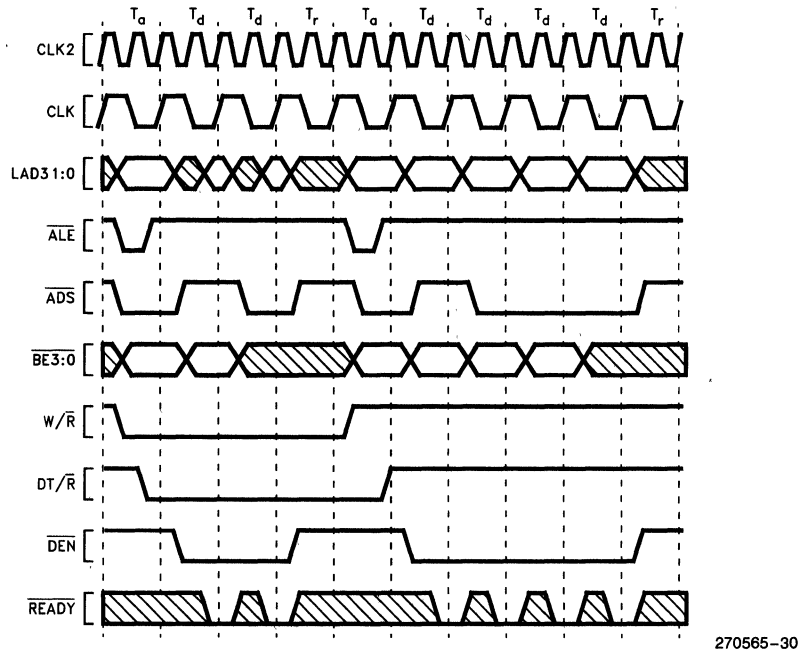


Figure 28. Burst Read and Write Transaction without Wait States

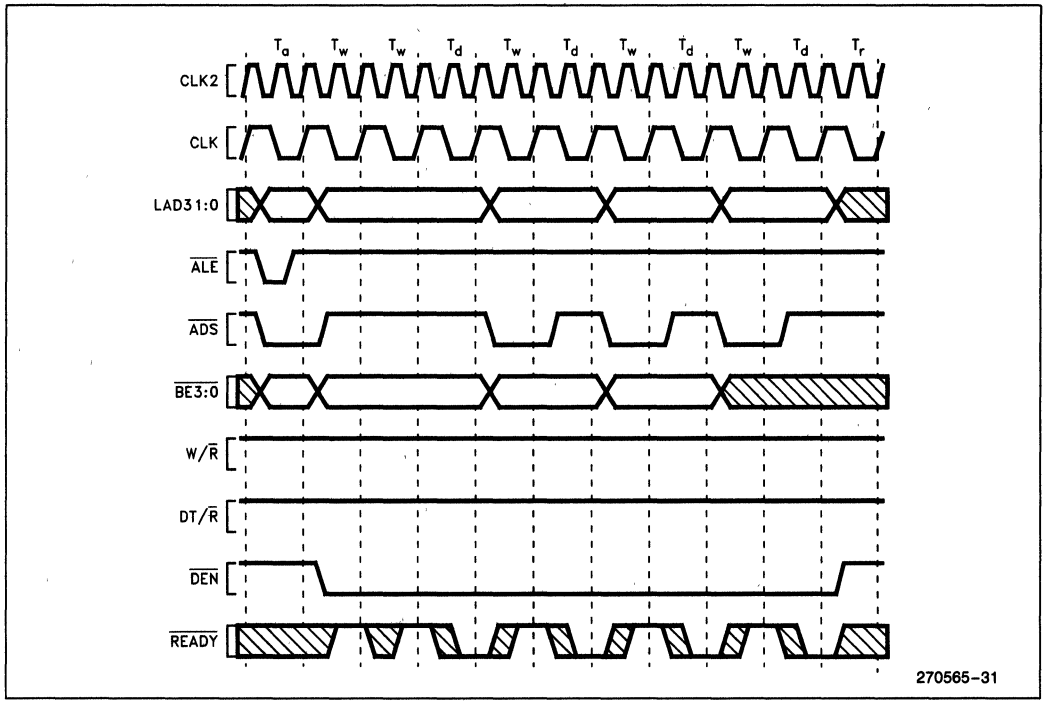


Figure 29. Burst Write Transaction with 2, 1, 1, 1 Wait States

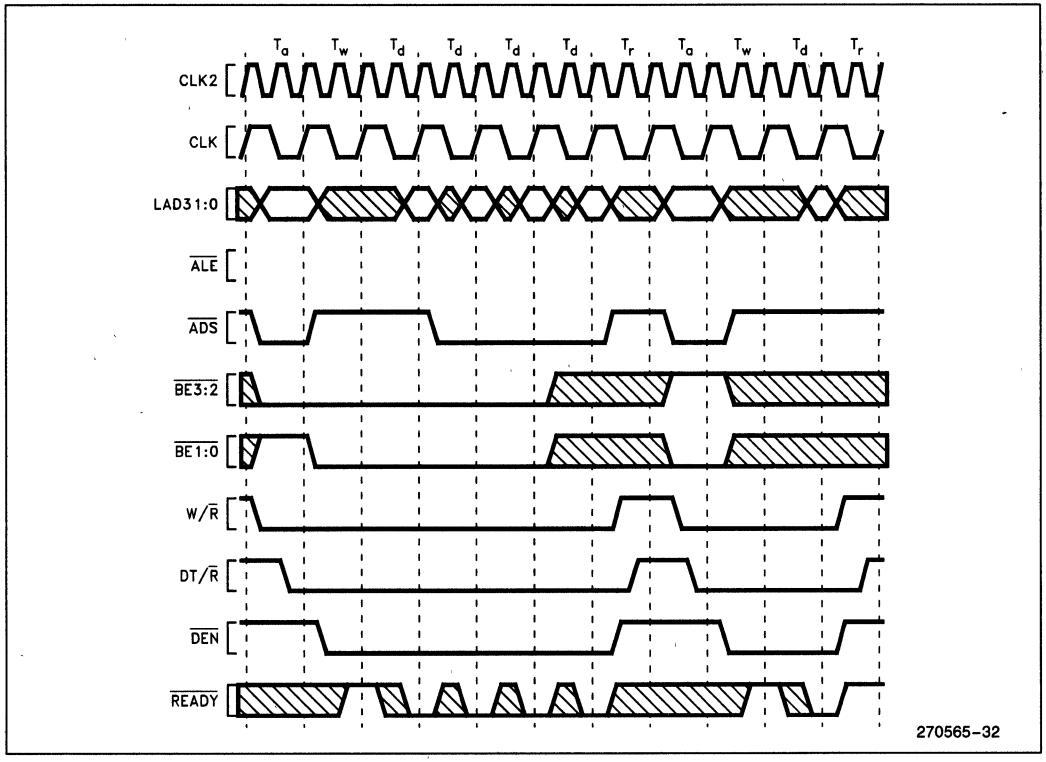
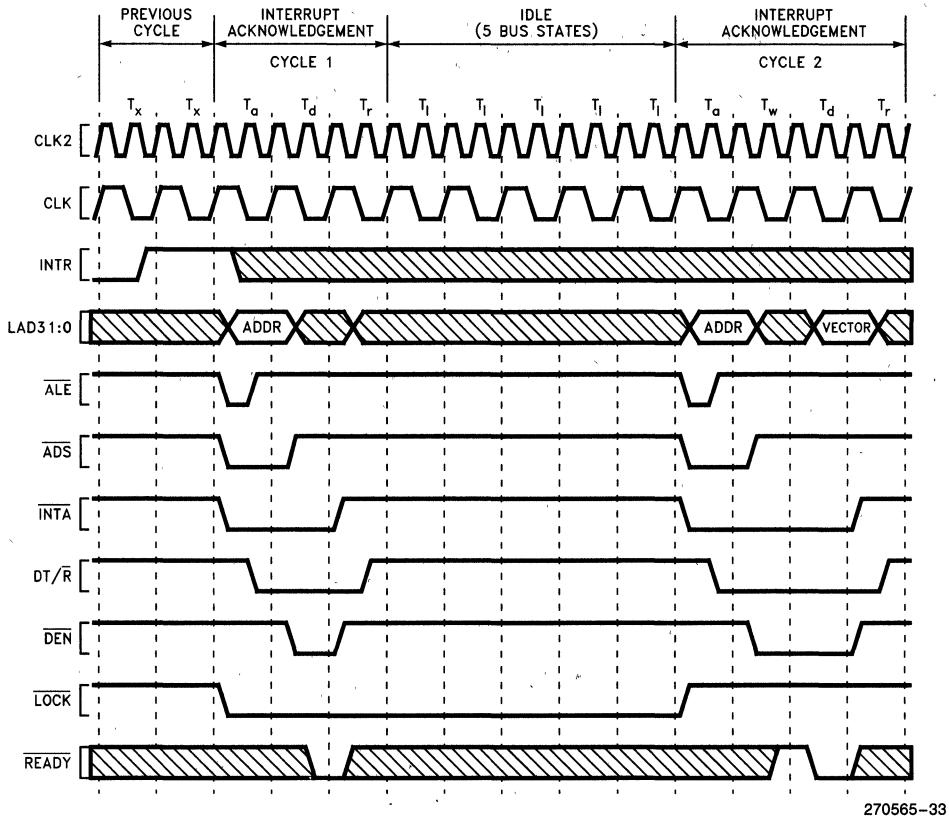


Figure 30. Accesses Generated by Quad Word Read Bus Request, Misaligned Two Bytes from Quad Word Boundary (1, 0, 0, 0 Wait States)



NOTE:
INTR can go low no sooner than the input hold time following the beginning of interrupt acknowledgment cycle 1. For a second interrupt to be acknowledged, INTR must be low for at least three cycles before it can be reasserted.

Figure 31. Interrupt Acknowledge Transaction

3.6 Revision History

No revision history was maintained in earlier revisions of this data sheet. All errata that has been identified to date is incorporated into this revision. The sections significantly changed since the previous revision are:

Section	Last Rev.	Description
Table 4. 80960KB Pin Description: L-Bus Signals (pg. 10)	-006	$\overline{LOCK}$ pin description rewritten for clarity.
2.3. Connection Recommendations (pg. 13)	-006	Changed suggested open-drain termination networks to reflect more realistic operating conditions with reduction in DC power consumption.
Figure 9. Typical Current vs. Frequency (Hot Temp) (pg. 15)	-006	Added figure for typical power supply current at hot temperature to aid thermal analysis.
Figure 12. Test Load Circuit for Three-State Output Pins (pg. 16) Figure 13. Test Load Circuit for Open-Drain Output Pins (pg. 16)	-006	All outputs now specified with standard 50 pF test loads to agree with actual test methodology.
3.1. DC Characteristics (pg. 17)	-006	I_{CC} max specification reduced: **WAS:** 375 mA, 420 mA, 480 mA **IS:** 315 mA, 360 mA, 420 mA **AT:** 16 MHz, 20 MHz, 25 MHz
3.2. AC Specifications (pg. 18)	-006	25 MHz operation extended to product in PQFP package. T_8 min. improved at all frequencies from 0 ns to 2 ns and T_8 max. improved from 20 ns to 18 ns. T_{8H} max improvement: **WAS:** 31 ns, 26 ns, 24 ns **IS:** 28 ns, 23 ns, 23 ns **AT:** 16 MHz, 20 MHz, 25 MHz
Functional Waveforms	-006	Redrawn for clarity. CLK signal drawn with more likely phase relationship to CLK2. Open-drain output signals drawn to show correct inactive states.
Various	-006	Deleted all references to 10 MHz. Intel no longer offers a 10 MHz 80960KB device.

intel®

80960CA
Product Overview

32-Bit High-Performance Embedded Processor with On-Chip DMA Controller, Interrupt Controller, High-Speed Bus Unit, Instruction and Register Caches

September 1989

ADVANCE INFORMATION

Order Number: 270669-001

80960CA PRODUCT OVERVIEW

CONTENTS

80960CA PRODUCT OVERVIEW

1.0 PURPOSE

The *80960CA Product Overview* is a summary of the features and operation of Intel's 80960CA Embedded Processor. The Product Overview is intended for those who are not familiar with the 80960 architecture or the 80960CA, a product built around this architecture. The 80960CA Product Overview provides a programmer or a system designer with a quick, global view of software and hardware design considerations for the 80960CA. For further information, refer to the following reference documents:

— The *80960CA User's Manual* contains detailed technical information and examples for designing embedded systems using the 80960CA.

— The *80960CA Data Sheet* provides electrical specifications for the device, such as the DC and AC parameters, operating conditions, and packaging specifications.

2.0 80960CA 32-BIT EMBEDDED PROCESSOR

The 80960CA (Figure 2-1) is optimized for embedded processing applications. This product features the high-performance C-Series core plus built-in system peripherals, effectively integrating a high-speed CPU and system components onto a single silicon die. The 80960CA is a member of Intel's 80960 embedded processor family. Each member of the 80960 family is based on a common architectural definition referred to as the *core architecture.*

An 80960 family member, such as the 80960CA, is made up of an implementation of the core architecture plus application-specific extensions. These extensions may consist of integrated peripherals, instruction-set extensions, or additional registers and caches beyond those defined by the architecture. The common core architecture provides a basis for code compatibility for all 80960 family products, while application-specific extensions optimize a particular product for a class of applications.

The 80960 architectural target is the execution of multiple instructions per clock (i.e., fractional clocks per instruction). By defining an architecture which supports parallel instruction execution and out-of-order instruction execution, performance advances are not constrained by the system clock.

The 80960CA is capable of launching and executing instructions in parallel. This is accomplished by the use of advanced silicon technology as well as innovative "microarchitectural" constructs. The term microarchitecture refers to the implementation of the instruction set and programming resources. For example, different microarchitectures may have different pipeline construction, internal bus widths, register set porting, degrees of parallelism, and cache parameterization (two-way, four-way, etc.).

A principal objective of the 80960 architecture is to provide the framework to allow microarchitectural advances to translate directly into increased performance without architectural limitations.

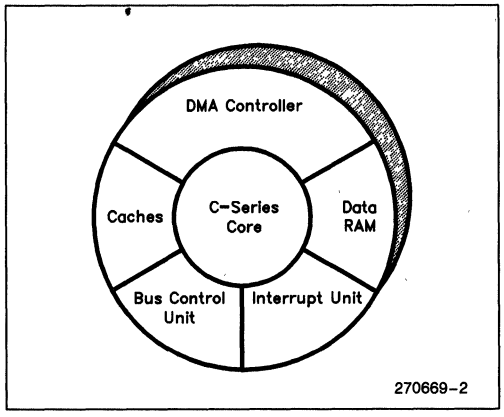


Figure 2-1. 80960CA

2.1 80960 Architecture

Embedded applications are cost sensitive, require a different mix of instructions than reprogrammable applications, have demanding interrupt response requirements, and often use real-time executives rather than full-blown operating systems. The 80960 architecture was developed with these factors in mind. Several key optimizations which are provided by the architecture are explained below.

Instruction Set: Powerful Boolean operations are provided. Frequently executed functions are available as single instructions for greater code density and performance. Call, Return, Compare-and-Branch, Conditional-Compare, Compare-and-Increment or Decrement, and Bit-Field-Extract are each single instructions.

Interrupts: A priority interrupt structure simplifies the management of real-time events. With 31 discrete levels of priority and 248 possible interrupt-handling procedures, this structure provides the low latency and high throughput interrupt handling required in embedded processor applications.

Faults: A generalized fault-handling mechanism simplifies the task of detecting errant arithmetic calculations or other conditions that typically require a significant amount of in-line user code.

Application-Specific Extensions: The core architecture is designed to accept application-specific extensions such as instruction set extensions (e.g., string functions, floating point), special purpose registers, larger caches, on-chip program and data memory, a memory management and protection unit, fault-tolerance support, multiprocessing support, and real-time peripherals (DMA, serial ports, etc.).

2.2 80960 C-Series Core

The C-series core is an implementation of the 80960 core architecture. The core can execute instructions at a sustained speed of 66 MIPS(1) with bursts of performance up to 99 MIPS. To achieve this level of performance, Intel has incorporated state-of-the-art silicon technology and innovative microarchitectural constructs into the C-Series core. Factors which contribute to the core's performance are listed below.

— Parallel instruction decoding allows the 80960CA to start two instructions in every clock, with bursts of three instructions per clock.

— Most instructions execute in a single clock cycle.

— Multiple independent execution units enable overlapping instruction execution.

— Advanced silicon technology allows operation with a 33 MHz internal clock.

— Efficient instruction pipeline is designed to minimize pipeline break losses.

— Register and resource scoreboarding transparently manage parallel execution.

— Branch look-ahead feature enables branches to execute in parallel with other instructions.

— Local register cache is integrated on-chip.

— 1 Kbyte two-way set associative instruction cache is integrated on-chip.

— 1 Kbyte Static Data RAM is integrated on-chip.

These factors combine to make the 80960CA an ultra-high performance computing engine.

NOTE:
1. Single clock instructions at 33 MHz.

2.3 80960CA System Peripherals

The 80960CA features several extensions to the core architecture in the form of integrated peripherals. These peripherals are intended to reduce the external system requirements needed for embedded applications. These peripherals are described below.

Bus Controller Unit: A 32-bit high-performance bus controller interfaces the 80960CA to external memory and peripherals. The bus controller transfers instructions or data at a maximum rate of 132 Mbytes per second.(2) Internally programmable wait states and 16 separately configurable memory regions allow the bus controller to interface with a variety of memory subsystems with minimum system complexity and maximum performance.

DMA Controller: A four channel DMA controller performs high speed data transfers between peripherals and memory. The DMA controller provides advanced features such as data chaining, byte assembly and disassembly, and a fly-by mode capable of transfer speeds of up to 66 Mbytes per second. The DMA controller features a performance and flexibility which is only possible by integrating the DMA controller and the 80960CA core.

Interrupt Controller: A priority interrupt controller manages 8 external interrupt inputs, 4 internal interrupt sources from the DMA controller, and a single non-maskable interrupt input (NMI). A total of 248 external interrupt sources are supported by the interrupt controller by configuring the 8 external interrupt pins as an 8-bit input port. The interrupt controller provides the mechanism for the low latency and high throughput interrupt service featured by the 80960CA. The interrupt latency for the 80960CA is typically less than 1 μs.

3.0 EXECUTION ENVIRONMENT

The *Execution Environment* (Figure 3-1) refers to the resources which are available for executing code on the 80960CA. The following sections describe the elements of the execution environment.

3.1 Registers and Literals

The 80960CA provides four types of working data registers: *Global Registers, Local Registers, Special Function Registers* (SFRs), and *Control Registers.*

Global and local registers are general purpose 32-bit data registers. The SFRs and the control registers provide a programmer's interface to the on-chip peripherals (i.e., the DMA controller, interrupt controller, and bus controller).

NOTE:
2. 33 MHz internal clock, load or instruction fetch on 0 wait state, pipelined burst bus.

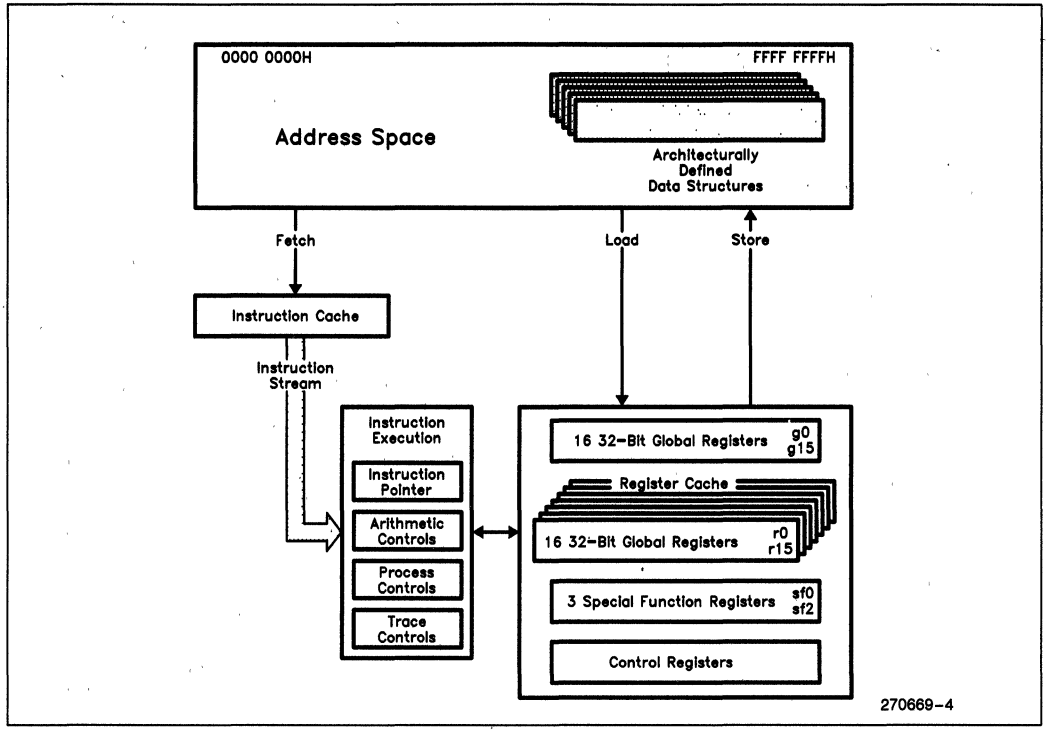


Figure 3-1. Execution Environment

The 80960 architecture is a register-oriented architecture. That is, operands and results of instructions are placed in working data registers rather than in memory. Since the architecture is register oriented, an ample supply of registers is provided. The architecture's working register set consists of 16, 32-bit global registers and 16, 32-bit local registers.

3.1.1 GLOBAL AND LOCAL REGISTERS

The procedure call and return mechanism, which is part of the 80960 architecture, inspires the names given to the local and global registers. When a procedure call or return is executed, the contents of global registers are preserved across procedure boundaries. In other words, the same set of global registers is used for each procedure. A new set of local registers, however, is allocated for each procedure. The 80960's call and return mechanism is explained in Section 3.8.

The 80960CA supplies 16, 32-bit global registers designated **g0** through **g15**. Registers **g0** through **g14** are general purpose global registers. Register **g15** is reserved for the current Frame Pointer. This register is available in assembly language as the **fp** register. The **fp** contains the address of the first byte in the current stack frame. The **fp** register and the stack frame are described in Section 3.8.

The 80960CA supplies 16, 32-bit **Local Registers** designated **r0** through **r15**. Registers **r3** through **r15** are general purpose local registers. Registers **r0, r1,** and **r2** are reserved for special functions as follows: **r0** contains the Previous Frame Pointer, **r1** contains the Stack Pointer, and **r2** is reserved for the Return Instruction Pointer. These registers are available in assembly language as, respectively, the **pfp, sp,** and **rip** registers. The **pfp, sp,** and **rip** registers manage stack frame linkage for the 80960's procedure call and return mechanism. The function of these registers is decribed in Section 3.8.

3.1.2 SPECIAL FUNCTION REGISTERS AND CONTROL REGISTERS

The 80960CA uses 3 Special Function Registers (SFRs) for communicating with on-chip peripherals. These SFR's are an architectural extension specific to the 80960CA. The SFRs on the 80960CA are designated as **sf0, sf1,** and **sf2**. SFRs are accessed as source operands by most of the 80960CA's instructions. The registers serve as part of the programmer's interface to the DMA and interrupt controller.

Control registers, like SFRs are used to communicate with the on-chip peripherals. Configuration information for the peripherals is generally stored in these registers. Control registers can only be accessed by using the system control (**sysctl**) instruction. The **sysctl** instruction is used to load the internal control register from a table in external memory called the control table. In order to simplify the process of peripheral configuration, the control registers are automatically loaded from this table at initialization.

3.1.3 LITERALS

The 80960CA provides *literals* which may be used in the place of source register operands in most instructions. The literals range from 0 to 31 (5 bits). When a literal is used as an operand, the processor expands it to 32 bits by adding leading zeros. If the instruction defines an operand larger than 32 bits, the processor zero extends the literal to the operand size.

3.2 Address Space and Memory

The address space of the 80960CA (Figure 3-2) is considered a subset of the execution environment since the code, data, data structures, and external peripherals for the processor reside here. The 80960 family has an address space which is 2^{32} bytes (4 Gbytes) in size. This address space is linear (unsegmented); therefore, code, data, and peripherals may be placed anywhere in the usable space. For the 80960CA, some memory locations are reserved or are assigned special functions as shown in Figure 3-2.

3.2.1 INTERNAL DATA RAM

The 80960CA provides 1 Kbyte of internal static RAM for fast access of frequently used data. The data RAM allows time critical data storage and retrieval, with no dependence on the performance of the external bus. Any load or store, including quad-word operations,

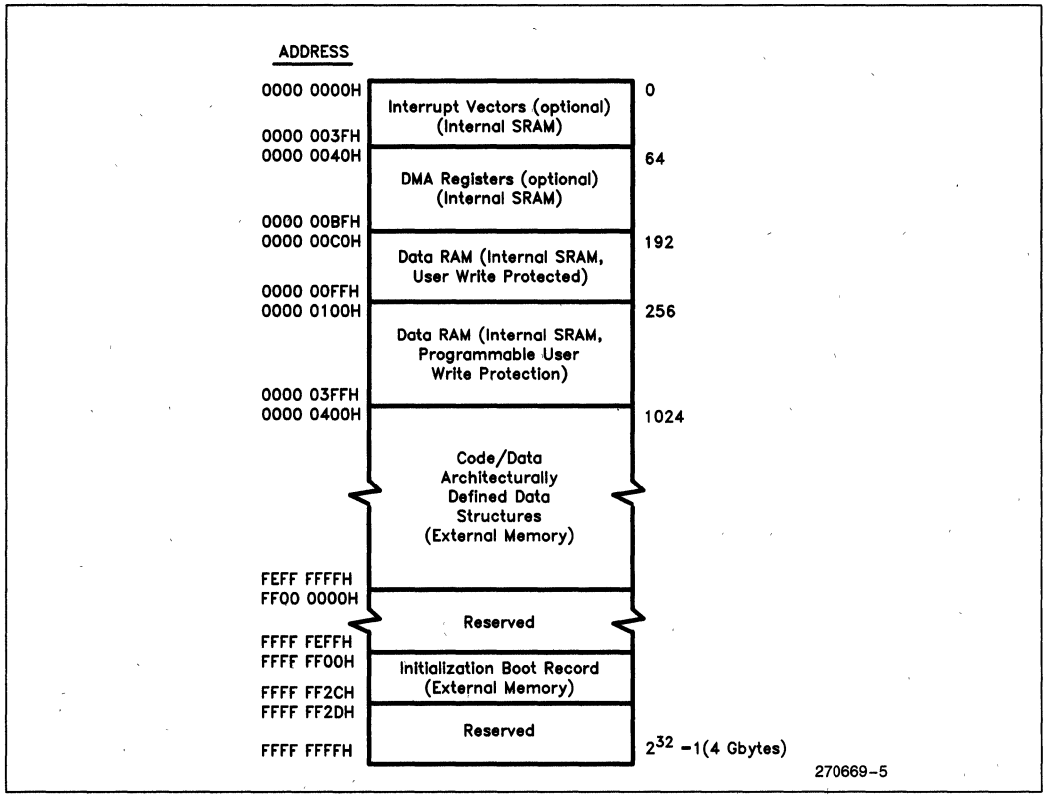


Figure 3-2. Address Space

execute in a single clock cycle when directed to internal data RAM. The data RAM is located at address 00H in the processor's address space. When the DMA controller is in use, 32 bytes of data RAM are reserved for each active DMA channel. Additionally, 64 bytes of data RAM are reserved for 16 interrupt vectors which may be cached internally to reduce interrupt latency. The data RAM reserved for the DMA controller and the interrupt controller can be used for additional data storage when these peripherals are not used.

Two execution modes are possible on the 80960CA, *user mode* or *supervisor mode*. These modes are used to implement a protection model in which system data structures are isolated from user code. As shown in Figure 3-2, the first 256 bytes of data RAM are always write protected when a program is executing in user mode but may always be written when executing in supervisor mode. The remainder of the data RAM can be programmed for this protection feature. The user and supervisor modes are described further in Section 3.7.

3.2.2 RESERVED ADDRESS SPACE

The upper 16 Mbytes of memory (FF000000H–FFFFFFFFH) are reserved for specific functions and extensions to the 80960 architecture. The 12 words in reserved space (FFFFFF00H–FFFFFF2CH) are used to start up the processor when it comes out of reset. These 12 words are called the *initialization boot record*.

3.2.3 ARCHITECTURALLY DEFINED DATA STRUCTURES

To execute a program on the 80960CA, data structures specific to the 80960 architecture must reside in the processor's address space. Architecture-defined data structures include stacks, initialization structures, and various procedure entry tables. These data structures may generally be located anywhere in the address space. Pointers to each data structure are specified when the 80960CA is initialized. The architecture-defined data structures include:

- Interrupt Table
- System-Procedure Table
- Fault Table
- User Stack
- Interrupt Stack
- Supervisor Stack

In addition to the data structure defined by the architecture, the 80960CA requires several implementation-specific data structures which are used for configuring peripherals and initialization. These data structures include:

- Control Table
- Process Control Block
- Initialization Boot Record

Each data structure will be explained in more detail later in this product overview.

3.3 Memory Addressing Modes

The 80960CA offers a variety of modes for memory addressing. The addressing modes available are summarized in Table 3-1.

Absolute addressing is used to reference an address as an offset from address 0 of the processor's address spaee. At the machine level, absolute addressing may be implemented in one of two ways depending on the size of the absolute offset from address 0. Two instruction formats, MEMA and MEMB, are used to provide absolute addressing modes. For the MEMA format, the offset is an ordinal number ranging from 0 to 2048. For the MEMB format, the offset is an integer (called a displacement) ranging from $-2^{31}-1$ to 2^{31}. An assembler will choose the MEMA or MEMB format based on the size of the offset.

Register-indirect addressing modes use a 32-bit ordinal value in a register as the base for the address calculation. Offsets and indexes are added to this address base depending on the particular addressing mode. The *register-indirect-with-index* addressing mode adds a scaled index to the address base. The index is specified as a value in a register. The scale value may be selected as 1, 2, 4, 8, or 16.

The *index-with-displacement* addressing mode uses a scaled index plus an integer displacement. No address base is used in this address calculation.

The *IP-with-displacement* addressing mode is used with load and store instructions to make them IP relative. In this mode, an integer displacement plus a constant of 8 is added to the IP of the instruction to calculate the next address.

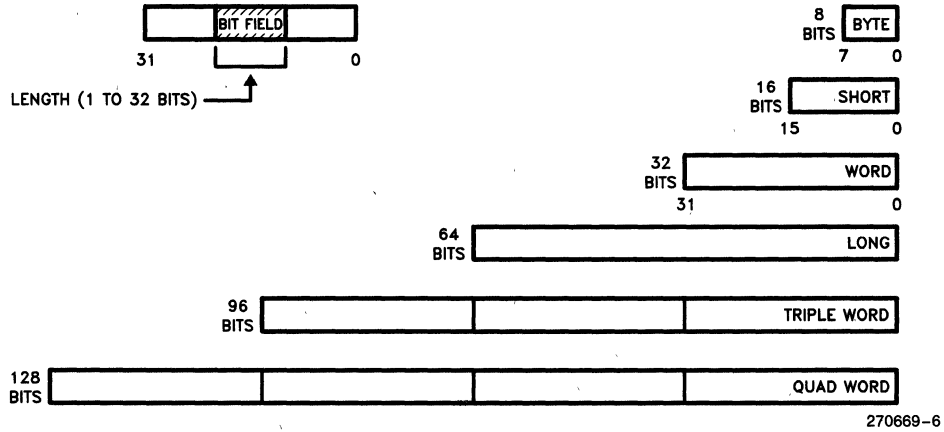
Table 3-1. Memory Addressing Modes

Mode	Description
Absolute Offset	Offset
Absolute Displacement	Displacement
Register Indirect	Abase
Register Indirect with Offset	Abase + Offset
Register Indirect with Index	Abase + (Index*Scale)
Register Indirect with Index and Displacement	Abase + (Index*Scale) + Displacement
Index with Displacement	(Index*Scale) + Displacement
Register Indirect with Displacement	Abase + Displacement
IP with Displacement	IP + Displacement + 8

3.4 Data Types

The 80960CA operates on the following data types (Figure 3-3):

— Integer (8, 16, 32, and 64 bits)
— Ordinal (8, 16, 32, and 64 bits)
— Bit
— Bit Field
— Triple Word (96 bits)
— Quad Word (128 bits)



Class	Data Type	Length	Range
Numeric (Integer)	Byte Integer	8 bits	-2^7 to $2^7 - 1$
	Short Integer	16 bits	-2^{15} to $2^{15} - 1$
	Integer	32 bits	-2^{31} to $2^{31} - 1$
	Long Integer	64 bits	-2^{63} to $2^{63} - 1$
Numeric (Ordinal)	Byte Ordinal	8 bits	0 to $2^8 - 1$
	Short Ordinal	16 bits	0 to $2^{16} - 1$
	Ordinal	32 bits	0 to $2^{32} - 1$
	Long Ordinal	64 bits	0 to $2^{64} - 1$
Non-Numeric	Bit	1-bit	N/A
	Bit Field	1–32 bits	
	Triple Word	96 bits	
	Quad Word	128 bits	

Figure 3-3. Data Types

The following sections describe the data types supported by the 80960CA.

3.4.1 NUMERIC DATA TYPES

Integers and ordinals are considered numeric data types since the processor performs arithmetic operations with this data. The integer data type is a signed binary value in standard 2's complement representation. The ordinal data type is an unsigned binary value.

3.4.2 NON-NUMERIC DATA TYPES

The remaining data types (bit field, triple word, and quad word) represent groupings of bits or bytes that the processor can operate on as a whole, regardless of the nature of the data contained in the group. These data types facilitate the moving of blocks of bits or bytes.

3.5 Instruction Set

The 80960CA features a comprehensive instruction set (Table 3-2). Much of the instruction set is that of a RISC architecture. Unlike pure RISC machines, however, the 80960CA provides an extension to the RISC instruction set with instructions that perform complex functions such as procedure calls and returns, high-speed multiplies, and other complex control, arithmetic, and logical operations. The instruction set allows functionally complex yet highly compact code to be written for embedded control applications where memory is a valuable commodity.

3.5.1 INSTRUCTION GROUPS

The 80960CA instruction set is most easily described if grouped by the functions listed below:

— Data Movement
— Address Computation
— Logical and Arithmetic
— Bit and Bit Field
— Comparison
— Branch
— Call and Return
— Fault
— Debug
— Processor Management

The instructions which make up each of these groups are described in the following sections.

3.5.1.1 Data Movement Instructions

The data movement instructions move data from memory to registers, from registers to memory, and between registers. The load instructions copy bytes, words, or multiple words from memory to a selected register or group of registers. Conversely, the store instructions copy bytes, words, or groups of words from a selected register or group of registers to memory. The move instructions copy data between registers.

Load Instructions

- **ld** load word
- **ldob** load ordinal byte
- **ldos** load ordinal short
- **ldib** load integer byte
- **ldis** load integer short
- **ldl** load long
- **ldt** load triple
- **ldq** load quad

Store Instructions

- **st** store word
- **stob** store ordinal byte
- **stos** store ordinal short
- **stib** store integer byte
- **stis** store integer short
- **stl** store long
- **stt** store triple
- **stq** store quad

Move Instructions

- **mov** move word
- **movl** move long
- **movt** move triple
- **movq** move quad

3.5.1.2 Address Computation Instructions

The load address (**lda**) instruction causes a 32-bit address to be computed and placed in a destination register. The address is computed based on the addressing mode selected. The load and store instructions perform a function identical to that of the **lda** instruction when calculating a source or destination address. The **lda** instruction is useful for loading a 32-bit constant into a register.

3.5.1.3 Logical and Arithmetic Instructions

Logical instructions perform bitwise Boolean operations on operands in registers. Since this group of instructions performs only bitwise manipulations of data, separate logical instructions for integer and ordinal data types do not exist. In the table below, src1 and src2 represent processor registers or literals which are the operands for these instructions.

Table 3-2. Instruction Set Summary

Data Movement	Arithmetic	Logical	Bit and Bit Field
Load Store Move	Add Subtract Multiply Divide Remainder Modulo Shift Extended Shift Extended Multiply Extended Divide Add with Carry Subtract with Carry	And Not And And Not Or Exclusive Or Not Or Or Not Nor Exclusive Nor Not Nand Rotate	Set Bit Clear Bit Not Bit Check Bit Alter Bit Scan for Bit Scan for Byte Span over Bit Extract Modify
Comparison	**Branch**	**Call and Return**	**Fault**
Compare Condition Compare Compare and Increment Compare and Decrement Condition Test	Unconditional Branch Conditional Branch Branch and Link Condition Compare and Conditional Branch	Call Call Extended Call System Return	Conditional Fault Synchronize Faults
Debug	**Processor Management**	**Address Computation**	**Atomic**
Modify Trace Controls Mark Force Mark	Modify Process Controls Modify Arithmetic Controls System Control Update DMA Setup DMA Flush Local Registers	Load Address	Atomic Add Atomic Modify

1

Logical Instructions

- **and** src1 and src2
- **notand** src1 and (not src2)
- **andnot** (not src1) and src2
- **or** src1 or src2
- **notor** src1 or (not src2)
- **ornot** (not src1) or src2
- **xor** src1 xor src2
- **xnor** src1 xnor src2
- **nor** not (src1 or src2)
- **nand** not (src1 and src2)
- **not** not (src1)

Arithmetic instructions perform add, subtract, multiply, divide, and shift operations on integer or ordinal operands in registers.

Arithmetic Instructions

- **addi** add integer
- **addo** add ordinal
- **subi** subtract integer
- **subo** subtract ordinal
- **muli** multiply integer
- **mulo** multiply ordinal
- **divi** divide integer
- **divo** divide ordinal
- **remi** remainder integer
- **remo** remainder ordinal
- **modi** modulo integer
- **rotate** rotate bit left
- **shli** shift left integer
- **shlo** shift left ordinal
- **shri** shift right integer
- **shro** shift right ordinal
- **shrdi** shift right dividing integer

Extended arithmetic instructions facilitate computation on ordinals and integers which are longer than 32 bits. In add with carry and subtract with carry instructions, the carry out from the previous arithmetic instruction is used in the computation. The extended multiply instruction multiplies two ordinal source operands producing a long ordinal result (64 bits). The extended divide instruction divides a long ordinal dividend by an ordinal divisor and produces a 64-bit result. The extended shift right instruction shifts a 64-bit source value and produces the lower order 32 bits of the shifted value.

Extended Arithmetic Instructions

- **addc** add ordinal with carry
- **subc** subtract ordinal with carry
- **emul** extended multiply
- **ediv** extended divide
- **eshro** shift right extended ordinal

The atomic instructions perform read-modify-write operations on operands in memory. They allow a system to insure that when an atomic operation is performed on a specified memory location, the operation will be completed before another agent is allowed to perform an operation on the same memory. These instructions are required to enable synchronization between interrupt handlers and background tasks in any system. They are also particularly useful in systems where several agents (processors, coprocessors, or external logic) have access to the same system memory for communication.

Atomic Instructions

- **atadd** atomic add
- **atmod** atomic modify

3.5.1.4 Bit and Bit Field Instructions

The bit instructions operate on a specified bit in a register.

Bit Instructions

- **setbit** set bit
- **clrbit** clear bit
- **notbit** not bit
- **alterbit** alter bit
- **scanbit** scan for bit
- **spanbit** span over bit

Bit field instructions operate on a specified contiguous group of bits in a register. This group of bits can be from 0 to 32 bits in length.

Bit Field Instructions

- **extract** extract field
- **modify** modify field
- **scanbyte** scan for byte

3.5.1.5 Branch Instructions

The branch instructions allow the direction of program flow to be changed by explicitly modifying the *Instruction Pointer (IP)*. The target IP in a branch instruction is generally specified as a displacement to be added to the current IP. The extended branch instructions allow IP calculation using any addressing mode.

The unconditional branch instructions always alter program flow when executed.

Unconditional Branch Instructions

- **b** branch
- **bx** branch extended

The RISC branch-and-link instructions automatically save a Return Instruction Pointer (RIP) before the

jump is taken. The RIP is the address of the instruction following the branch and link.

Branch and Link Instructions

- **bal** branch and link
- **balx** branch and link extended

Conditional branch instructions alter program flow only if the *condition code flags* in the arithmetic control register match a value specified in the instruction. The condition code flags indicate conditions of equality or inequality between two operands in a previously executed instruction. The arithmetic control register and condition code flags are described in Section 3.6.

Based on a *branch prediction flag* located in the machine level instruction, the 80960CA will assume that an instruction usually takes or does not take a conditional branch. By executing along the predicted path of program flow, delays due to breaks in the instruction stream are often avoided. This feature of the 80960CA is referred to as *branch prediction*. The 80960CA incorporates the branch prediction feature because code using a conditional branch instruction usually favors a single direction of program flow.

The branch prediction flag is specified at the assembly level by appending a *.t* or *.f* to a conditional branch instruction meaning, respectively, "assume branch taken" or "assume branch not taken". For example, the assembler mnemonic **be.t** means that the processor will assume that this branch-if-equal instruction usually branches when encountered. In the following table .p represents the branch prediction flag.

Conditional Branch Instructions

- **be.p** branch if equal
- **bne.p** branch if not equal
- **bl.p** branch if less
- **ble.p** branch if less or equal
- **bg.p** branch if greater
- **bge.p** branch if greater or equal
- **bo.p** branch if ordered
- **bno.p** branch if unordered

Compare and conditional branch instructions compare two operands, then branch according to the immediate results.

Conditional Compare and Conditions Branch Instructions

- **cmpibe.p** compare integer and branch if equal
- **cmpibne.p** compare integer and branch if not equal
- **cmpibl.p** compare integer and branch if less
- **cmpible.p** compare integer and branch if less or equal
- **cmpibg.p** compare integer and branch if greater
- **cmpibge.p** compare integer and branch if greater or equal
- **cmpibo.p** compare integer and branch if ordered
- **cmpibno.p** compare integer and branch if unordered
- **cmpobe.p** compare ordinal and branch if equal
- **cmpobne.p** compare ordinal and branch if not equal
- **cmpobl.p** compare ordinal and branch if less
- **cmpoble.p** compare ordinal and branch if less or equal
- **cmpobg.p** compare ordinal and branch if greater
- **cmpobge.p** compare ordinal and branch if greater or equal
- **bbs.p** check bit and branch if set
- **bbc.p** check bit and branch if clear

3.5.1.6 Compare and Condition Test Instructions

The 80960CA provides several types of instructions that are used to compare two operands. The condition code flags in the arithmetic control register are set to indicate whether one operand is less than, equal to, or greater than the other operand.

Compare Instructions

- **cmpi** compare integer
- **cmpo** compare ordinal
- **chkbit** check bit

Conditional compare instructions test the existing status of the condition code flags before a compare is

performed. These conditional compare instructions are provided to optimize two-sided range comparisons (i.e. to test if a value is less than one number but greater than another).

Conditional Compare Instructions

- concmpi	conditional compare integer
- concmpo	conditional compare ordinal

The compare and increment and compare and decrement instructions set the condition code flags based on a comparison of two register sources, decrements or increments one of the sources, and finally stores this result in a destination register.

- cmpinci	compare and increment integer
- cmpinco	compare and increment ordinal
- cmpdeci	compare and decrement integer
- cmpdeco	compare and decrement ordinal

The condition test instructions allow the state of the condition code flags to be tested. Based on the outcome of the comparison, a true or false code is stored in a destination register. The branch prediction flag is used in this instruction to reduce the execution time of the instruction when the test outcome is predicted correctly. For example **teste.t** (test if equal) will execute in a shorter time if the condition code flags test true for the equal condition. Analogous to the function of the branch prediction flag in the conditional compare and branch instructions, the prediction flag in this case eliminates breaks in the micro-instruction sequence which is used to implement the condition test instructions.

Condition Test Instructions

- teste.p	test if equal
- testne.p	test if not equal
- testl.p	test if less
- testle.p	test if less or equal
- testg.p	test if greater
- testge.p	test if greater or equal
- testo.p	test if ordered
- testno.p	test if not ordered

3.5.1.7 Call and Return Instructions

The 80960CA features an on-chip call and return mechanism for making procedure calls to local and system procedures. The call instructions and the call and return mechanism is described in Section 3.8.

Call and Return Instructions

- call	call
- callx	call extended
- calls	call system
- ret	return

3.5.1.8 Fault Instructions

The 80960CA will fault automatically as the result of certain errant operations which may occur when executing code. Fault procedures are then invoked automatically to handle the various types of faults. In addition, the fault instructions permit a fault to be generated explicitly based on the value of the condition code flags. The branch prediction flag in these instructions is used to reduce the execution time of these instructions when the state of the condition code flags are guessed correctly.

Conditional Fault Instructions

- faulte.p	fault if equal
- faultne.p	fault if not equal
- faultl.p	fault if less
- faultle.p	fault if less or equal
- faultg.p	fault if greater
- faultge.p	fault if greater or equal
- faulto.p	fault if ordered
- faultno.p	fault if unordered

The **syncf** instruction causes the processor to wait for all faults to be generated which are associated with any prior uncompleted instructions.

- syncf	synchronize faults

3.5.1.9 Debug Instructions

The processor supports debugging and monitoring of program activity through the use of trace events. The debug instructions support debugging and monitoring software.

Debug Instructions

- modtc	modify trace controls
- mark	mark
- fmark	force mark

3.5.1.10 Processor Management Instructions

The 80960CA provides several instructions for direct control of processor functions and for configuring the 80960CA's peripherals. A brief description of the processor management instructions is given below.

Processor Management Instructions

- modpc	modify process controls
- modac	modify arithmetic controls
- sysctl	system control instruction
- udma	update DMA SRAM
- sdma	setup DMA
- flushreg	flush local registers

3.6 Arithmetic Controls

The *Arithmetic Control (AC) Register* is a 32-bit on-chip register (Figure 3-4). The AC register is used primarily to monitor and control the execution of 80960CA arithmetic instructions. The processor reads and modifies bits in the AC register when performing many arithmetic operations. The AC register is also used to control the faulting conditions for some instructions. The **modac** instruction allows the user to directly read or modify the AC register.

The processor sets the condition code flags (bits 0–2) to indicate equality or inequality as the result of certain instructions (such as the compare instructions). Other instructions, such as the conditional branch instructions, take action based on the value of the condition code flags. Table 3-3 shows the functional assignment for each condition code flag.

Table 3-3. Arithmetic Condition Codes

Condition Code	Condition
001	Greater Than
010	Equal
100	Less Than

The integer overflow flag (bit 8) and the integer overflow mask (bit 12) are used in conjunction with the arithmetic integer overflow fault. The mask bit masks the integer overflow fault. When the fault is masked, and an integer overflow occurs, the integer overflow flag is set but no fault handling action is taken. If the fault is not masked, and an integer overflow occurs, the integer overflow fault is taken and the integer overflow flag is not set.

The no imprecise faults flag (bit 15) determines if imprecise faults are allowed to occur. Fault handling and precise and imprecise faults in the 80960CA are discussed in Section 3.10.

3.7 Process Management

Process management refers to the monitoring and control of certain properties of an executing process. The following sections describe the mechanisms available on the 80960CA to perform this function.

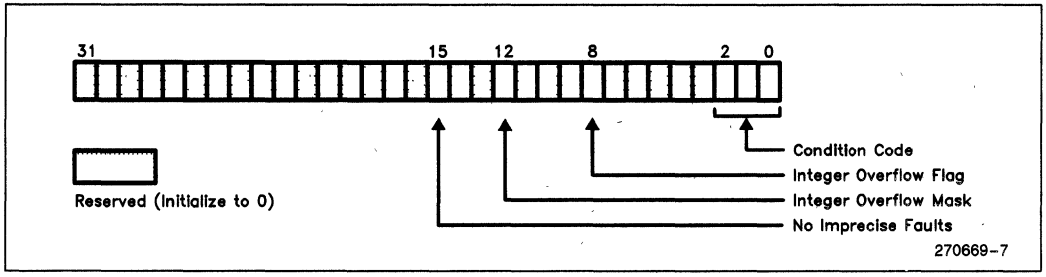


Figure 3-4. Arithmetic Control Register

3.7.1 PROCESS CONTROL REGISTER

The *Process Control (PC) Register* (Figure 3-5) provides access to process state information. The function for the PC register is described below.

Execution Mode Flag—This flag indicates that the processor is executing in user mode (0) or supervisor mode (1).

Priority Field—This 5-bit field indicates the current executing priority of the processor. Priority values range from 0 to 31, with 0 as the lowest and 31 as the highest priority.

State Flag—This flag determines the executing state of the processor. The processor state is either executing state (0) or interrupted state (1).

Trace Enable Bit and Trace Fault Pending Flags—These fields control and monitor trace activity in the processor. The Trace Enable Bit enables fault generation for trace events. The Trace Fault Pending Flag indicates that a trace event has been detected.

The process controls can be modified by software with the modify process controls (**modpc**) instruction. The **modpc** instruction may only write the PC register when the processor is in supervisor mode.

3.7.2 PRIORITIES

The 80960 architecture defines a means to assign priorities to executing programs and interrupts. The current priority of the processor is stored in the priority field of the PC register. This priority is used to determine if an interrupt will be serviced and in which order multiple pending interrupts will be serviced. Setting the priority of an executing program above that of interrupts allows critical code to be prioritized and executed without interruption.

The priority field of the PC register can be modified directly using the **modpc** instruction. The priority field is also modified to reflect the priority of serviced interrupts. On a return from an interrupt routine, the priority of the processor is restored to its priority before the interrupt occurred.

3.7.3 PROCESSOR STATES AND MODES

The 80960CA may execute programs in *user mode* or *supervisor mode*. The user-supervisor protection mechanism allows a system to be designed in which kernel code and data reside in the same address space as user code and data, but access to the kernel procedures and data is only allowed through a tightly controlled interface. This interface is the system call table and the interrupt mechanism. The 80960CA provides a supervisor pin ($\overline{SUP}$) to implement memory systems which protect code and data from possible corruption by programs executing in user mode. Some instructions and functions of the 80960CA are also insulated from code executing in user mode.

The processor has two operating states: executing and interrupted. In executing state, the processor can execute in user or supervisor mode. In the interrupted state, the processor always executes in supervisor mode.

3.8 Call and Return Mechanism

The 80960 architecture features a built-in call and return mechanism. This mechanism is designed to make procedure calls simple and fast, and to provide a flexible method for storing and handling variables that are local to a procedure. A call automatically allocates a new set of local registers and a new stack frame. All linkage information is maintained by the processor, making procedure calls and returns virtually transparent to the user. A system call instruction is provided as a method for calling privileged procedures such as a kernel service. The call and return model supports efficient translation of structured high level code (such as C, or ADA) to 80960 machine language.

The procedure call and return mechanism provides a number of significant benefits which contribute to the performance and ease of use of the 80960CA.

1) The call and return instructions are implemented entirely on-chip, resulting in an extremely high performance implementation of these commonly used functions.

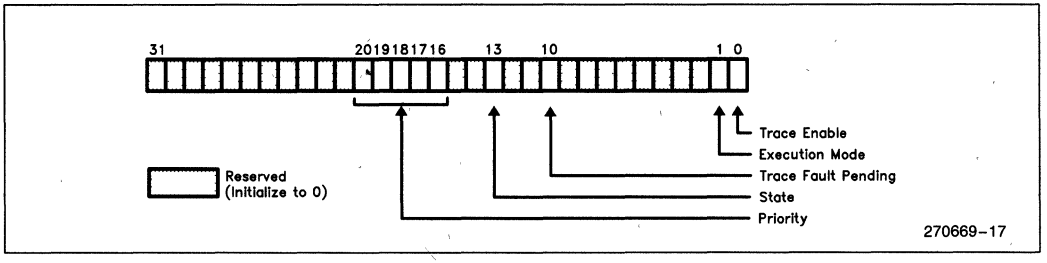


Figure 3-5. Process Control Register

2) A single instruction to implement each call or return operation results in code density improvements compared to processors which require multiple instructions to encode these functions.
3) By implementing the call and return functions as single instructions, the 80960 architecture is open for further optimization of these instructions, while maintaining assembly-level compatibility.
4) A program does not have to explicitly save or restore the variables stored in the local registers when a call or return is executed. The processor does this implicitly on procedure calls and on returns.
5) The call and return mechanism provides a structure for storing a virtually unlimited number of local variables for each procedure: the on-chip local registers provide quick access to often used variables and the stack provides space for additional variables.

3.8.1 LOCAL REGISTERS AND THE STACK FRAME

At any point in a program, the 80960 has access to a local register set and a section of the procedure stack referred to as a stack frame. When a call is executed, a new stack frame is allocated for the called procedure. Additionally, the current local register set is saved by the processor, freeing these registers for use by the newly called procedure. In this way, every procedure has a unique stack and unique set of local registers. When a return is executed, the current local register set and current stack frame are deallocated. The previous local register set and previous stack frame are restored. This call and return mechanism is illustrated in Figure 3-6 where n is procedure depth for the currently executing procedure.

The procedure stack structure is defined by the 80960 architecture. The procedure stack always grows upward (i.e. towards higher addresses) and the stack pointer (SP) always points to the next available byte of the stack frame. The 80960CA requires that each stack frame begins on a 16-byte boundary. Due to this alignment requirement, a padding space of 0 to 15 bytes may exist between adjacent stack frames in memory. When a stack frame is allocated, the first 16 words are always assigned as storage for the local registers; therefore, the SP initially points to the 17th word in the stack frame. It should be noted that although each stack frame is assigned storage space for the local registers, these locations in the stack are not guaranteed to contain the values of the saved local registers. This is because several sets of local registers are cached on-chip rather than written to the stack in external memory. This caching mechanism is described in detail later in this section.

3.8.2 PROCEDURE LINKING

The 80960 architecture automatically manages procedure linkage. One global register and three local registers are reserved for procedure linkage information.

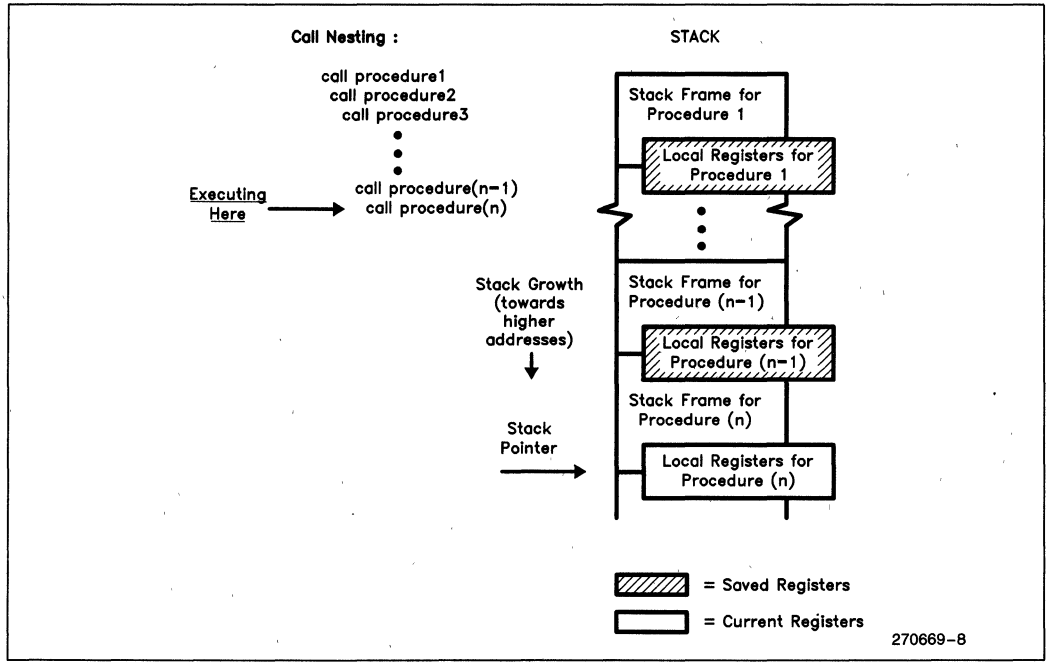


Figure 3-6. Call and Return Mechanism

Figure 3-7 describes the pointer structure used to link frames and to provide a unique SP for each frame. Register **g15** is the *Frame Pointer (FP)*. The FP is the address of the first byte of the current (topmost) stack frame. The FP is always updated to point to the current frame when calls and returns are executed. Register **r0** is the *Previous Frame Pointer (PFP)*. The PFP is the address of the first byte of the stack frame which was created prior to the frame containing this PFP. Register **r1** is the *Stack Pointer (SP)*. The SP points to the next available byte of the stack frame. Register **r2** is reserved for the *Return Instruction Pointer (RIP)*. The RIP is the address of the instruction which follows a call instruction, this is also the target address for the return from that procedure. The RIP is automatically stored in register **r2** of the calling procedure when a call is executed.

3.8.3 PARAMETER PASSING

Parameters may be passed by value or passed by reference between procedures. The global registers, the stack, or predefined data structures in memory may be used to pass these parameters.

The global registers provide the fastest method for passing parameters. The values to be passed into a procedure reside in the global registers of the calling procedure. When a procedure is called, the values in the global registers are preserved. If more parameters are to be passed than will fit in the global registers, additional parameters may be passed in the stack of the calling procedure, or in a data structure which is referenced by a pointer passed in the global registers.

3.8.4 LOCAL REGISTER CACHE

The 80960CA provides an on-chip cache for saving and restoring the local registers on calls and returns. This cache greatly enhances performance of the call and return mechanism on the 80960CA. Movement of data between the local registers and the register cache is typically accomplished in only 4 processor clocks with no external bus traffic. When this cache is filled, the registers associated with the oldest stack frame are moved to the area reserved for those registers on the physical stack (Figure 3-7).

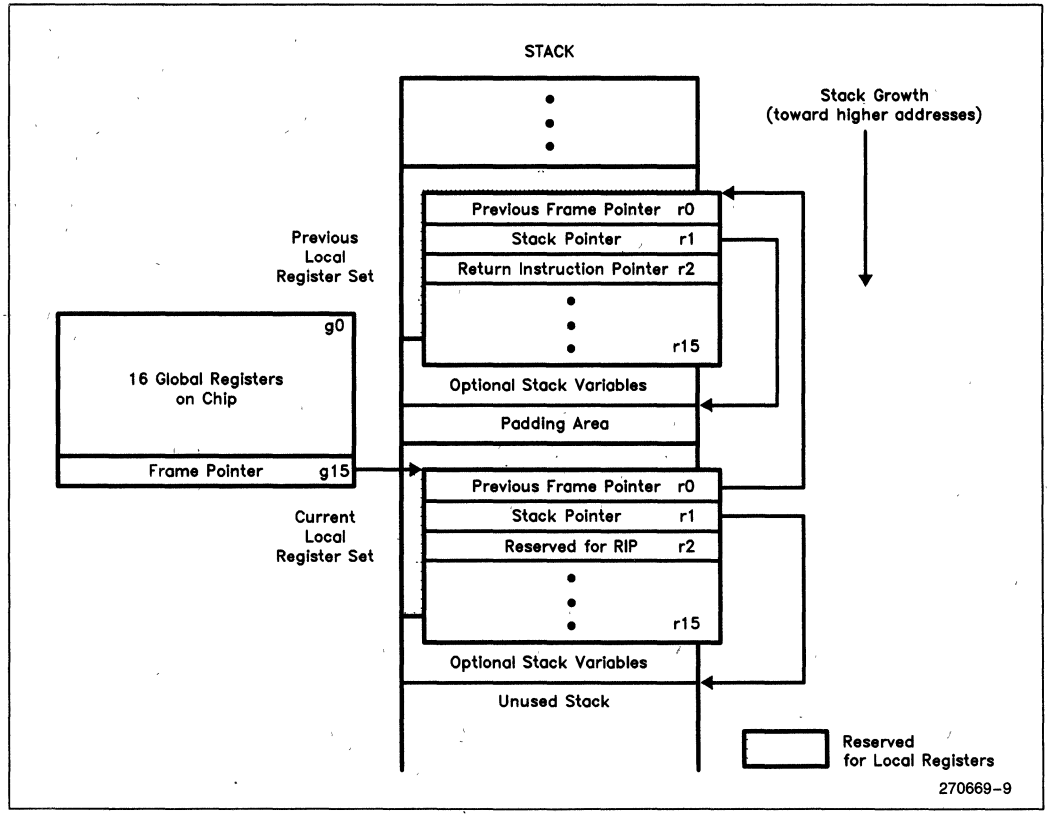


Figure 3-7. Stack Frame Linkage

The local register cache is a physical extension of the internal data RAM. The part of the data RAM used for this cache is not visible to the user and is large enough to hold up to 5 sets of local registers. The register cache may be extended to hold up to 15 sets of local registers. When extended, each new register set consumes 16 words of the user's data RAM, beginning at the highest address and growing downward. The size of the local register cache is selected when the processor is initialized.

In some cases, the contents of the cached local register sets may require examination or modification (e.g. for fault handling). Since the local registers are cached, the **flushreg** instruction is provided to flush the local register cache to the locations reserved for the registers on the stack. This insures that the values in external memory are consistent with the values held in the local register cache.

3.8.5 LOCAL AND SYSTEM CALLS

The 80960CA provides two methods for making procedure calls: local calls and system calls. Local and system calls differ in their operation and use in an application.

The local call instructions initiate a procedure call using the call and return mechanism described earlier. The stack frames for these procedure calls are allocated on the *local procedure stack*. A local call is made using either of two local call instructions: **call** or **callx**. The **call** instruction specifies the address of the called procedure using an IP plus displacement addressing mode with a range of -2^{23} to $2^{23}-4$ bytes from the current IP. The **callx** (call extended) instruction specifies the address of the calling procedure using any of the 80960's addressing modes.

A system call is made using the **calls** instruction. This call is similar to a local call except that the processor gets the IP for the called procedure from a data structure called the *system procedure table*. The **calls** instruction requires a procedure number operand. This procedure number serves as an index into the system procedure table, which contains IP's for specific procedures. The system procedure table is shown in Figure 3-8.

The system call mechanism supports two types of procedure calls: system-local calls and system-supervisor calls (also referred to as supervisor calls). The system-

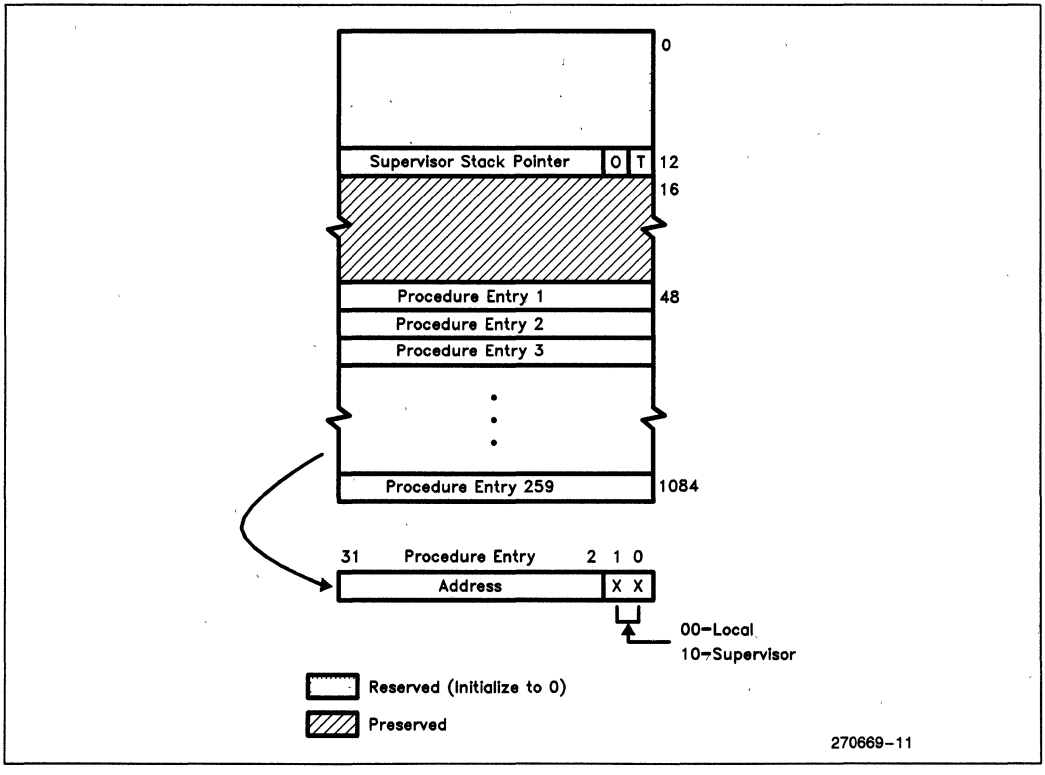


Figure 3-8. System Procedure Table

local call performs the same action as the local call instructions with one exception: the IP target for a system-local call is fetched from the system-procedure table. The supervisor call differs from the local call as follows:

1) A supervisor call causes the processor to switch to another stack (called the supervisor stack).
2) A supervisor call causes the processor to switch to the supervisor execution mode and asserts the 80960CA's supervisor ($\overline{SUP}$) pin for all bus accesses.

The system call mechanism offers several benefits. The system call promotes the portability of application software. System calls are commonly used for kernel services. By calling these services with a procedure number rather than a specific IP, application software does not have to be changed each time the implementation of the kernel service is modified. Additionally, the ability to switch to a different execution mode and stack allows kernel procedures and data to be insulated from application code.

3.8.6 IMPLICIT PROCEDURE CALLS

The call and return mechanism described for procedure calls applies to several classes of call instructions as well as to the context switching initiated by interrupts and faults. When an interrupt or fault condition occurs, an implicit call is performed that saves the current state of the processor before branching to the interrupt or fault handling procedure. When this context switch occurs, the local registers are saved and a new stack frame is allocated. Additionally, the values of the AC register and PC register are saved when the implicit call occurs. These values are restored on the return from the interrupt or fault handler.

3.9 Interrupts

An interrupt is a temporary break in the control stream of a program so that the processor can handle another task. Interrupts may be triggered by the instruction stream or by hardware sources internal and external to the 80960CA. An interrupt request is associated with a vector (i.e. an address) of an interrupt handling procedure. The processor will branch to the handling procedure when an interrupt is serviced. When the handling action is completed, the processor is restored to its state prior to the interrupt.

3.9.1 INTERRUPT VECTORS AND PRIORITY

Interrupt vectors are simply instruction pointers (addresses) to interrupt handling procedures. The 80960 architecture defines 248 interrupt vectors. This means that 248 unique interrupt handling procedures may be used. An 8-bit interrupt vector number is associated with each interrupt vector. This number ranges from 8 to 255. Each interrupt vector has a priority from 1 to 31, which is determined by the 5 most significant bits of the interrupt vector number. Priority 1 is the lowest priority and 31 is the highest. Priority 0 interrupts are not defined.

The 80960CA executes with a unique priority ranging from 0 to 31. When an interrupt is serviced, the processor's priority switches to the priority corresponding to that of the interrupt request. When a return from an interrupt procedure is executed, the process priority is restored to its value prior to servicing the interrupt. This priority switching is handled automatically by the 80960CA.

The 80960CA compares its current priority and the priority of an interrupt request to determine whether to service an interrupt immediately or to delay service. If a requested interrupt priority is greater than the processor's current priority or equal to 31, the processor services the interrupt immediately; otherwise, the processor saves (posts) the interrupt request as a pending interrupt so that it can be serviced later. When the processor's priority falls below the priority of a pending interrupt, the pending interrupt is serviced. With the mechanism described, interrupts with a priority of 0 will never be serviced. For this reason, vectors numbered 0 to 7 are not defined.

3.9.2 INTERRUPT TABLE

The interrupt table (Figure 3-9) is an architecturally defined data structure which holds the interrupt vectors and information on pending interrupts. The first 36 bytes of the table are used to post interrupts. The 31 most significant bits in the 32-bit pending priorities field represent a possible priority (1 to 31) of a pending interrupt. When the processor posts an interrupt in the interrupt table, the bit corresponding to the interrupt's priority is set. For example, if an interrupt with a priority of 10 is posted in the interrupt table, bit 10 is set in the pending priorities field.

The pending interrupts field contains a 256-bit string in which each bit represents an interrupt vector. When the processor posts an interrupt in the interrupt table, the bit corresponding to the vector number of that interrupt is set.

Portions of the interrupt table are cached on-chip in a non-transparent fashion. This caching is implemented to minimized interrupt latency by reducing the number of accesses to the table in external memory when an interrupt is serviced.

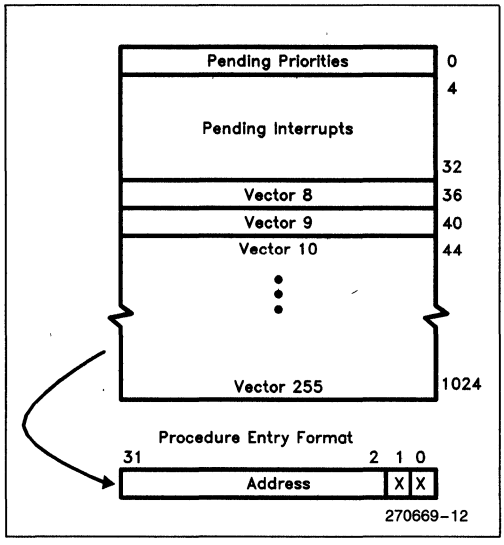


Figure 3-9. Interrupt Table

3.9.3 INTERRUPT STACK

Stack frames for interrupt handling procedures are allocated on a separate *interrupt stack*. The interrupt stack can be located anywhere in the processor's address space. The beginning address of the interrupt stack is specified when the processor is initialized.

3.9.4 INTERRUPT HANDLING ACTION

When an interrupt is serviced, the processor saves the processor state and calls the interrupt procedure. The processor state is restored upon return from the interrupt procedure.

This interrupt service mechanism is handled by an implicit call operation. When the interrupt is serviced, the current local registers are saved. A new local register set and stack frame are allocated on the interrupt stack for the interrupt handler procedure and the processor switches to supervisor execution mode. In addition to the local registers, the current value of the AC and PC registers are saved as an interrupt record on the interrupt stack.

3.9.5 PENDING INTERRUPTS

Any of the 248 interrupts can be requested by software. The system control instruction (**sysctl**) is provided to support this feature. When the system control instruction requests an interrupt, one of two actions may occur depending on the priority of the requested interrupt and the current process priority. 1) The interrupt is serviced immediately, or 2) the interrupt is posted (the pending priorities field and the pending interrupts field are modified to reflect a pending interrupt).

Interrupts may also be requested by hardware sources internal and external to the 80960CA. Managing the hardware sources and posting these interrupts is handled by the interrupt controller. Interrupts requested by hardware are posted in an internal register, not in the interrupt table. A mask register enables or disables interrupts from each hardware source. Requesting and posting hardware interrupts is described in Section 4.4 Interrupt Controller.

3.9.6 INTERRUPT LATENCY

The time required to perform an interrupt task switch is referred to as the ***interrupt latency***. The latency is the time measured between the activation of an interrupt source and the execution of the first instruction for the interrupt-handling procedure for the source.

Interrupt latency for the 80960CA varies depending on conditions such as:

— Complex instructions are executing when the interrupt occurs (e.g. **sysctl, call, ret,** etc.).

— Outstanding loads to a local register are pending, delaying the interrupt context switch.

— Division, multiplication, or other multi-cycle instructions with a local register as destination are executing.

The 80960CA has been designed to optimize latency and throughput for interrupts. Two processor features are designed for this purpose:

First, in the interrupt table, all interrupt vectors with an index whose least significant four bits are 0010_2 can be cached in internal data RAM. The processor will automatically read these vectors from data RAM when the interrupt is serviced. This feature reduces the added latency due to an external access of the interrupt table for that vector. The NMI vector is always cached in data RAM.

Second, an instruction cache locking mechanism allows interrupt procedures or segments of interrupt procedures to be stored in the instruction cache. These routines are always executed from the internal cache, eliminating external code fetches and reducing latency and increasing throughput for the interrupt.

3.10 Fault Handling and Instruction Tracing

The 80960CA is able to detect various conditions in code or in its internal state that could cause the processor to deliver incorrect or inappropriate results or that could cause it to head down an undesirable control path. These conditions are referred to as faults. The 80960 architecture provides fault handling mechanisms to detect and, in most cases, fully recover from a fault.

The 80960CA provides on-chip debug support by triggering trace events and servicing the trace fault. A trace event is activated when a particular instruction or type of instruction is encountered in an instruction stream. The trace event optionally signals a fault. A fault handling procedure for the trace fault can act as a debug monitor and analyze the state of the processor when the trace event occurred.

3.10.1 FAULT TYPES AND SUBTYPES

All of the faults that the processor detects are predefined. These faults are divided into types and subtypes, each of which is given a number. Table 3-4 lists the faults that the processor detects arranged by type and subtype.

Table 3-4. Fault Types and Subtypes

Fault Type	Fault Subtype	Fault Record
Parallel		XX00 00XX
Trace	Instruction Type	XX01 0002
	Branch Trace	XX01 0004
	Call Trace	XX01 0008
	Return Trace	XX01 0010
	Prereturn Trace	XX01 0020
	Supervisor Trace	XX01 0040
	Breakpoint Trace	XX01 0080
Operation	Invalid Opcode	XX02 0001
	Unimplemented	XX02 0002
	Invalid Operand	XX02 0004
Arithmetic	Integer Overflow	XX03 0001
	Arithmetic Zero-Divide	XX03 0002
Constraint	Range	XX05 0001
	Privileged	XX05 0002
Protection	Length	XX07 0001
Type	Mismatch	XX0A 0001

NOTE: X refers to preserved locations in the fault record.

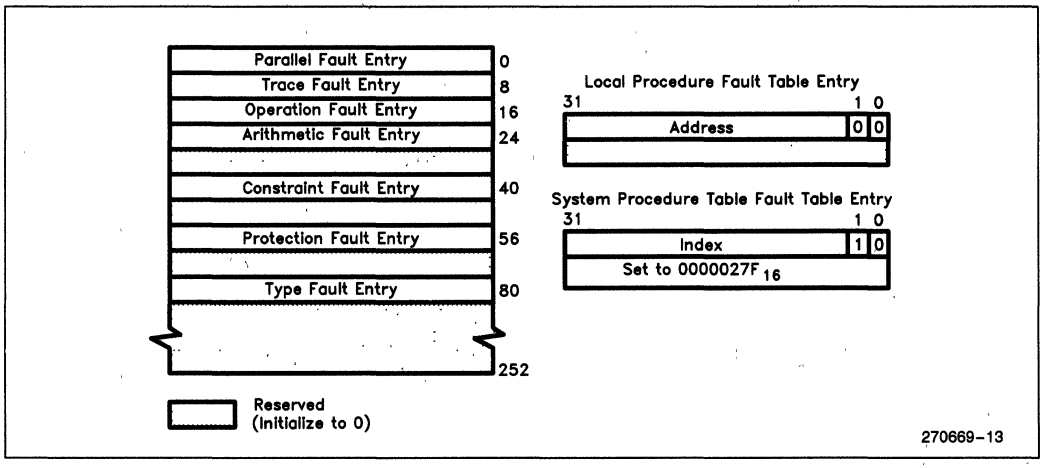


Figure 3-10. Fault Table

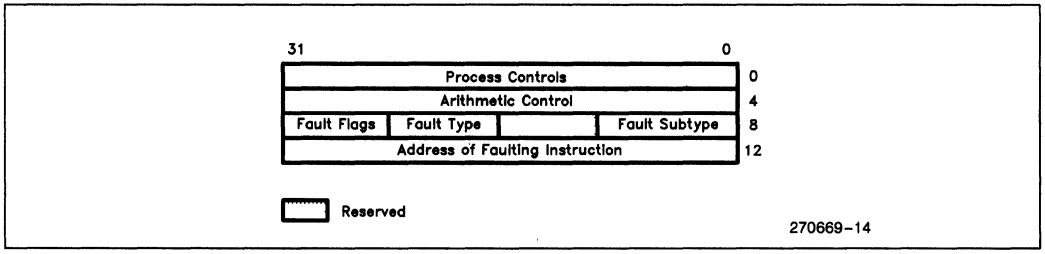


Figure 3-11. Fault Record

3.10.2 FAULT TABLE

The fault table (Figure 3-10) provides the processor with a pathway to fault handling procedures. The fault table is an architecture-defined data structure, which may be located anywhere in the processor's address space. The location of the fault table is specified at initialization. When a fault occurs, an entry in the table is selected based on the type of fault that occurs. The entry in the fault table contains a pointer to a specific fault handler.

The fault table can contain two types of entries (Figure 3-10). The first type of entry is simply a pointer to the address of the fault-handling procedure. The second type of entry is an index into the system-procedure table. Fault-handling procedures accessed through the system-procedure table may be executed in user or supervisor execution mode.

3.10.3 FAULT HANDLING ACTION

When a fault occurs, the processor performs an implicit call operation to the procedure specified in the fault table. In addition to performing the implicit call operation, the processor creates a fault record in its newly allocated stack frame. This fault record contains information on the state of the processor when the fault occurred and the fault type and subtype (Figure 3-11).

Some faults can be recovered from easily. When recovery from a fault is possible, the processor's fault handling mechanism allows the processor to automatically resume work where the fault was signalled. The resumption action is initiated with the **ret** instruction. If simple recovery from a fault is not possible, then the fault handling procedure may call a debug monitor, initiate a reset, or take other actions to recover from the fault.

3.10.4 TRACING AND DEBUG

The 80960CA provides a facility for monitoring the activity of the processor by tracing the instruction stream. A trace event occurs at points in a program where certain types of instructions are encountered or a certain IP or data address is encountered. When a trace event occurs, a trace fault can be generated and a trace-fault handler called which displays or analyzes the state of the processor.

3.10.4.1 Trace Events

The *Trace Control (TC) Register* (Figure 3-12) is used to specify the types of instructions which cause trace events. When a mode bit in the TC register is set, specific instructions will generate trace events. For example, if the branch trace mode bit is enabled and a branch instruction is executed, a branch trace event will be signalled. An event flag is used to record trace events. A single event flag is provided for each mode bit. Any trace event generates a trace fault when the trace enable bit in the process control register is set.

The 80960CA recognizes 7 trace events. These events are described below.

Instruction Trace Event—Signalled each time an instruction is executed. This trace event can be used with a debug monitor to single step the processor.

Branch Trace Event—Signalled each time a branch instruction is executed. For conditional branch instructions, this event is only signalled when the branch is taken. Branch-and-link, call, and return instructions do not signal this trace event.

Call Trace Event—Signalled each time a branch-and-link or call instruction is executed. Implicit calls, such as those used in interrupt or fault handling, signal this event. When a call trace event occurs, the prereturn trace flag (bit 3 in local register **r0**) is set by the processor to indicate a prereturn trace pending.

Pre-Return Trace Event—Signalled just prior to any **ret** instruction. This event is only signalled if the pre-return trace flag in register **r0** is set. Since the pre-return trace flag is set when a call trace event occurs, the call trace mode must be enabled before a pre-return trace event can be signalled.

Return Trace Event—Signalled each time a **ret** instruction is executed.

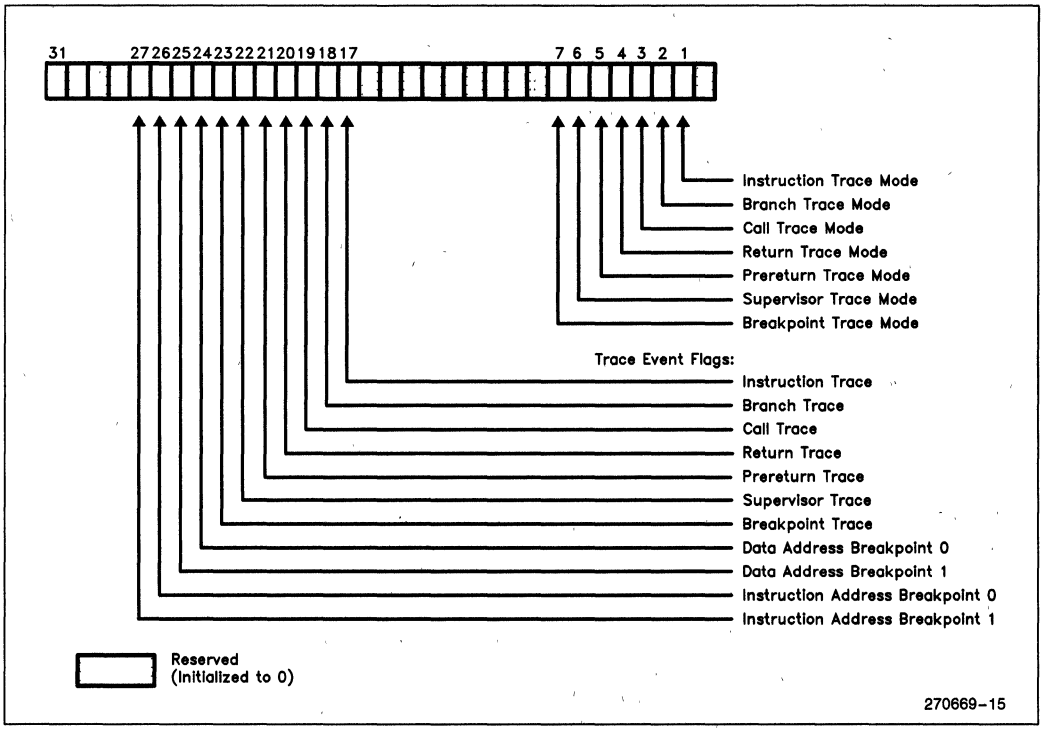


Figure 3-12. Trace Control Register

Supervisor Trace Event—Signalled each time a **calls** instruction is executed where the selected entry type is supervisor, or when a **ret** from supervisor mode is executed.

Breakpoint Trace Events—Signalled each time a **mark** instruction, **fmark** instruction, or specified address is encountered in the instruction stream. The **mark** instruction signals an event when the breakpoint trace mode is enabled, the **fmark** (force mark) instruction will generate a breakpoint trace event regardless of the value of the breakpoint trace mode bit.

Two IP breakpoint registers and two internal data address breakpoint registers are provided on the 80960CA. These breakpoints are loaded with an instruction or data address using the system control (**sysctl**) instruction. When the address is encountered and the breakpoint trace mode bit is set, a breakpoint trace event occurs. A corresponding instruction or data address event flag is set in the TC register when the address is encountered.

3.10.5 PROCESSOR INITIALIZATION

The Initial Memory Image (IMI) are the data structures needed to initialize the 80960CA (Figure 3-13). The initialization boot record, in reserved memory beginning at FFFFFF00H, contains a pointer to the Processor Control Block (PRCB). The PRCB in turn holds pointers to the data structures which are necessary to execute code on the 80960CA. The PRCB also holds several fields which contain information to initially configure the 80960CA.

Processor initialization begins by asserting the $\overline{\text{RESET}}$ pin. At initialization the processor optionally performs an internal self-test. A bus confidence test is also performed by calculating a checksum of 8 words read from external memory. If either of these self-tests fails, the $\overline{\text{FAIL}}$ pin indicates the failure and the processor aborts initialization. If the self-test passes, the 80960CA continues with initialization and branches to the first address of the user's code.

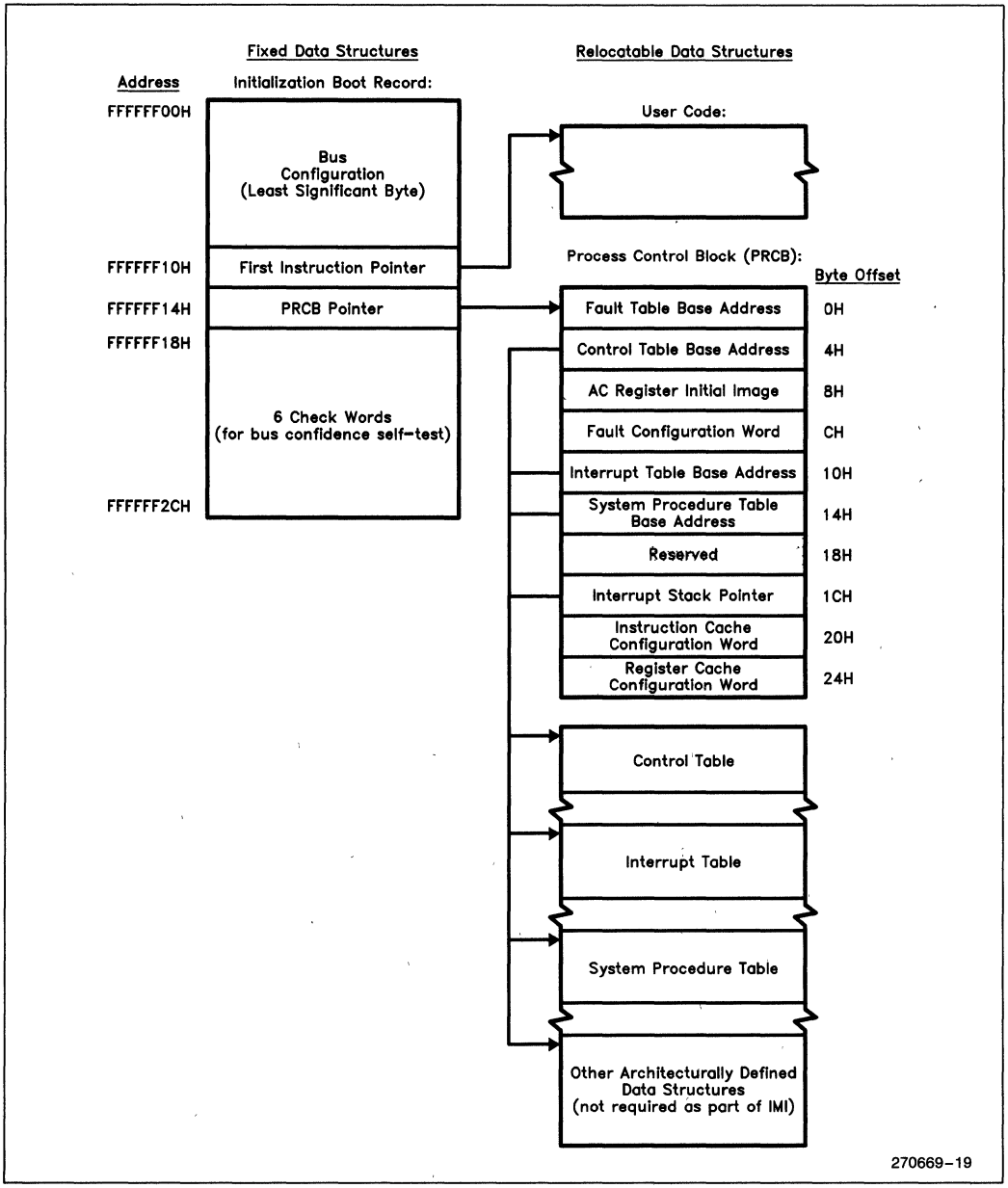


Figure 3-13. Initial Memory Image

4.0 80960CA SYSTEM IMPLEMENTATION

This section is an overview of the peripherals integrated with the 80960CA core. The features and operation of the Bus Controller, DMA Controller, Interrupt Controller, and the interfaces between these peripherals and the core are described.

4.1 Peripheral Interface

A program communicates with the on-chip peripherals by reading or modifying the special function registers (SFRs) or by loading control registers. The SFRs generally serve to transfer status information and data between a peripheral and the core, and the control registers serve to configure the peripherals. SFRs are accessed directly as instruction operands. The control registers are loaded by using the system control (**sysctl**) instruction.

4.2 Bus Controller Unit

The Bus Controller Unit (BCU) manages the data and instruction path between the 80960CA and external memory. Data operations and instruction fetches share a 32-bit data bus. Memory addresses are output on a separate 32-bit address bus. The BCU incorporates several advanced features to simplify the bus interface to external memory. A programmable *memory region configuration table* allows the characteristics of the external bus to be programmed differently for 16 separate regions in memory. The attributes of the external bus which are programmable include wait states and external ready control, data bus width (8, 16, or 32 bits), burst mode, address pipelining, and byte ordering. The region programmable bus options are described in this section.

4.2.1 BUS TRANSFERS, ACCESSES, AND REQUESTS

The distinction between *transfer, bus access,* and *bus request,* as these terms apply to the 80960CA, must be presented before beginning a discussion of the BCU.

Transfer—A *bus transfer* is defined simply as a movement of code or data between a memory system and the 80960CA. A write transfer occurs when the memory system is the destination of a data movement. A read transfer occurs when the 80960CA is the destination for a data or a code fetch from memory.

Bus Access—A *bus access* is defined as an address cycle and one or more transfers. In burst mode, an access can consist of a single address cycle and 1 to 4 transfers.

Bus Request—A *bus request* is issued by the core and directed to the Bus Controller. A bus request is sent to the BCU when a load, store, or an atomic instruction is executed, or when an instruction fetch is needed. Bus requests are also issued by the core to perform DMA transfers. A bus request can consist of one or more bus accesses. For example, an aligned word (32-bit) request to an 8-bit memory region will result in four byte-length accesses.

4.2.2 BUS CONTROL COPROCESSOR

The 80960CA's peripherals are often referred to as coprocessors, since their operation is decoupled from the execution of the instruction stream. As an integrated coprocessor, the BCU receives bus requests and independently carries out the action of moving data or code between the processor and external memory. The BCU uses a three deep queue to store pending bus requests. The queue decouples the core from the BCU, since a series of adjacent requests may be issued faster than the BCU can service each request. Two of the three queue entries store requests from a user's program (loads, stores, fetches, etc.). The third queue entry is used by requests originating from a DMA operation. This queue entry takes user requests when the DMA is turned off. The 80960CA alternates service of requests issued by the user program and requests issued by a DMA operation.

4.2.3 SIGNAL DESCRIPTIONS

The external bus signals consist of 30 address signals, 4 byte enables, 32 data lines, and various control signals.

D31–D0 32-bit Data Bus (bi-directional)—32-, 16-, and 8-bit values are transmitted and received on these lines. The 8- and 16-bit quantities are transferred on the low order data lines when a memory region is configured respectively for an 8- or 16-bit bus.

A31–A2 30-bit Address (outputs)—The 30-bit address bus identifies all external addresses to word (4-byte) boundaries. The byte enable lines indicate the selected byte in each word.

$\overline{BE3}$–$\overline{BE0}$ Byte Enables (outputs)—The byte enables select which of 4 addressed bytes are active in a memory access. When a memory region is configured for an 8-bit bus width, $\overline{BE1}$ and $\overline{BE0}$ act as the lower two bits of the address. For a 16-bit memory region, $\overline{BE1}$, $\overline{BE3}$, and $\overline{BE0}$ are encoded to provide A1, $\overline{BHE}$, and $\overline{BLE}$ respectively.

W/$\overline{R}$ Write or Read (output)—This signal is low for read accesses and high for write accesses.

$\overline{ADS}$ Address Strobe (output)—Indicates valid address and the start of a new bus access.

DT/$\overline{R}$ Data Transmit or Receive (output)—Direction control for data transceivers; similar to W/$\overline{R}$.

$\overline{DEN}$ Data Enable (output)—Low during a bus request after the first address cycle. This signal is used to control data transceivers and to indicate the end of a bus request.

$\overline{WAIT}$ Wait (output)—Indicates that wait states are being inserted by the internal wait state generator.

$\overline{READY}$ Ready (input)—Signals that data is valid for a read transfer or ends data hold for a write transfer. This function can be disabled for a memory region.

$\overline{BTERM}$ Burst Terminate (input)—Terminates a burst access. Another address is generated to complete the request when the signal is deasserted. This function can be disabled for a memory region.

D/C Data or Code (output)—Indicates a data transfer or a code fetch.

$\overline{DMA}$ DMA Access (output)—Indicates that a bus request was initiated by either the user program or the DMA.

$\overline{SUP}$ Supervisor Access (output)—Indicates that a bus access originated from a bus request issued in supervisor mode. This signal can be used to protect system data structures, or peripherals from errant modification by the user code.

$\overline{LOCK}$ Lock (output)—Indicates that an atomic memory operation is in progress. This signal can be used to inhibit external agents from modifying memory which is atomically accessed.

$\overline{BLAST}$ Burst Last (output)—Indicates the last transfer in a burst access.

HOLD Hold (input)—HOLD can be used by a bus requester to request access to the bus. The processor asserts HLDA after the current bus request or locked requests have completed.

HOLDA Hold Acknowledge (output)—Indicates to a bus requester that the processor has relinquished control of the bus.

BREQ Bus Request (output)—Indicates that requests are queued in the bus controller and are waiting to be serviced. BREQ can be used for external bus arbitration logic in conjunction with HOLD and HLDA to regain bus mastership.

Figure 4-1 shows the timing for a simple, non-burst, non-pipelined read and write access. The timing relations for the key control signals are shown in this figure.

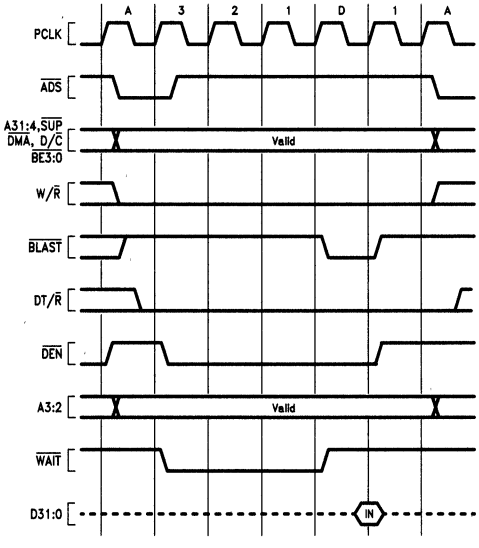


Figure 4-1. Basic Read and Write Request

4.2.4 MEMORY REGION CONFIGURATION TABLE

The BCU can be configured differently for 16 separate sections (referred to as regions) of the address space. The four most significant bits of a memory address define the location of each region in memory. The bus characteristics in a region are specified in the *memory region configuration table.* When a bus request is serviced, the BCU accesses the configuration table entry for the region addressed and services the request based on the bus characteristics programmed for that region. The characteristics programmed for each region are listed below:

- Burst Mode (on/off)
- Wait States (5 parameters)
- Bus Width (8-, 16-, or 32-bit)
- Ready Inputs (on/off)
- Address Pipelining (on/off)
- Byte Ordering (Big/Little Endian)

The flexibility of region programming simplifies the bus interface in applications where a memory system is made up of a variety of sub-systems, such as SRAM, DRAM, ROM, and memory mapped peripherals. Each memory sub-system can be mapped into a different region in memory, and that region can be configured specifically for the requirements of the particular memory sub-system.

The configuration table is made up of 16 on-chip control registers (Figure 4-2). Each register is programmed with the configuration information for a single region. Since the region table is located on-chip, access to region information does not affect the performance of the bus.

4.2.4.1 Burst Accesses

The 80960CA BCU is capable of burst accesses to memory systems which are designed to support this feature. Burst mode is intended to get the most performance from low cost memory systems. A burst access is a single address cycle followed by successive data or instruction transfers. The transfers reference data or instructions at sequential addresses starting at the address which began the burst access (Figure 4-3). In a burst memory system, the upper 28 bits of an address remain fixed while the lower two bits A2 and A3 increment to access subsequent locations.

Wait state timing for the first access of a burst request is controlled independently from the timing for subsequent accesses. A memory sub-system using static column mode or page mode DRAMs, for example, can take advantage of the short column access times for these devices by using burst mode. Interleaved ROM or EPROM systems can also be constructed which simultaneously access several words and then use burst mode to multiplex the multi-word array onto the data bus.

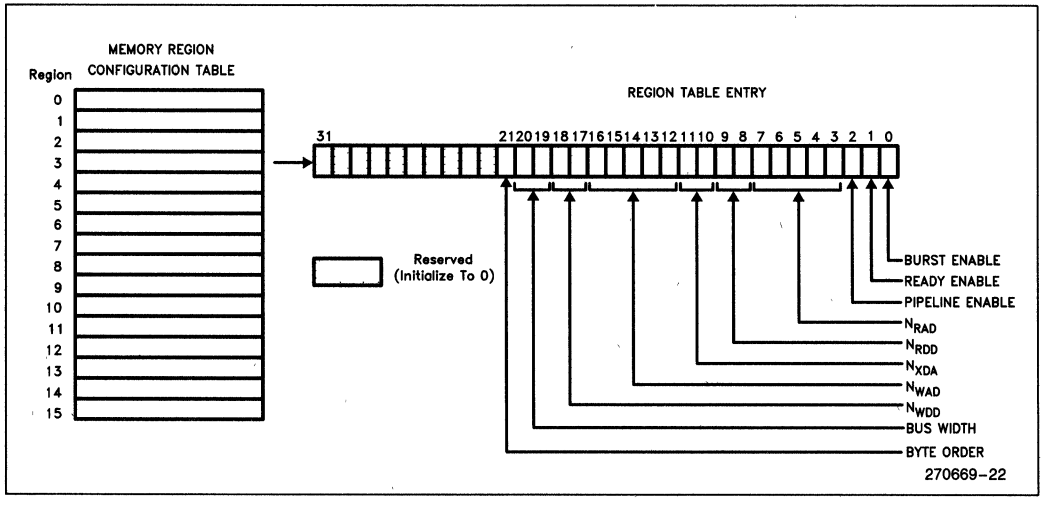


Figure 4-2. Memory Region Configuration Table

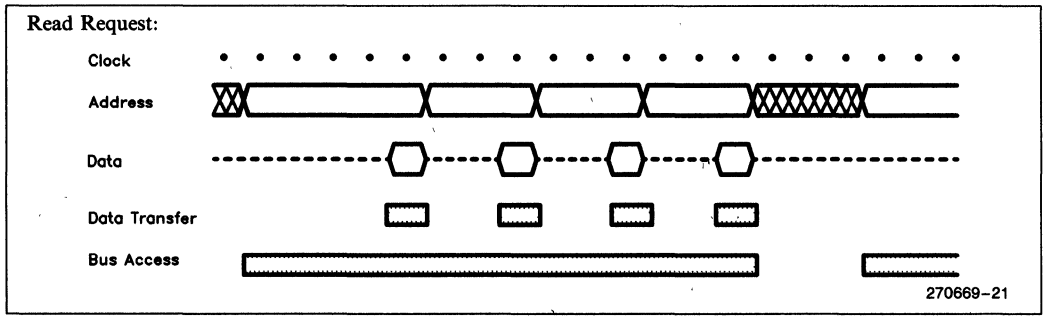


Figure 4-3. Burst Memory Request

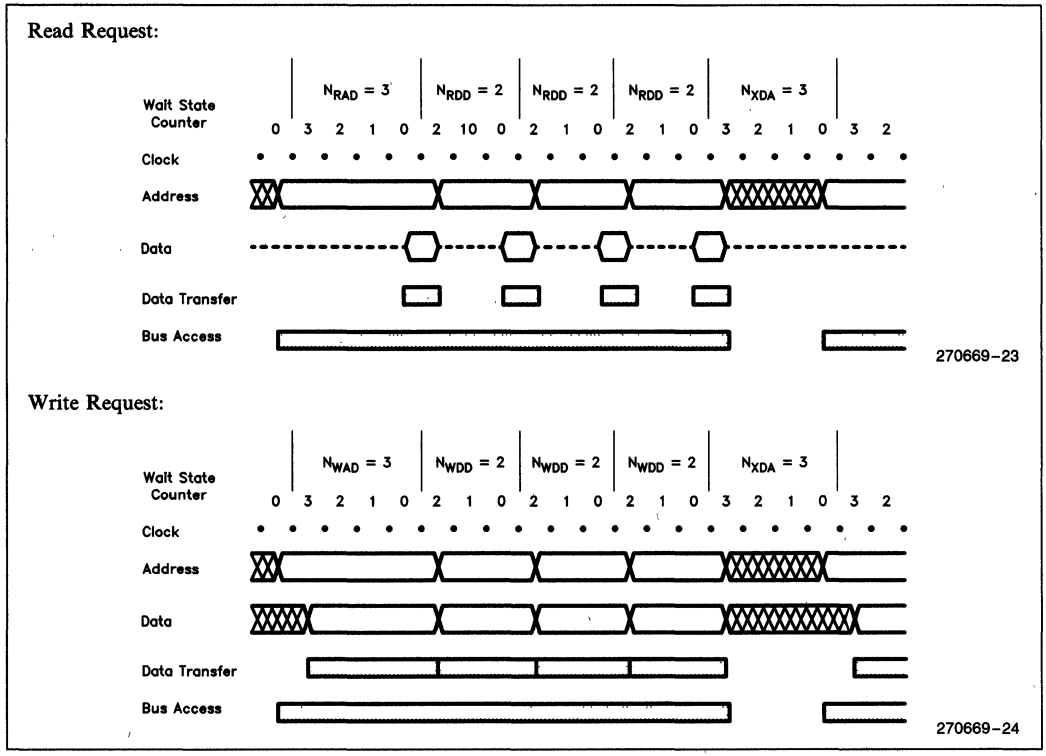


Figure 4-4. Programmable Wait States

4.2.4.2 Programmable Wait State Generation

The 80960CA may be interfaced with a variety of memory sub-systems and peripherals with a minimum system cost and complexity. To achieve this interface flexibility, the 80960CA implements an internal programmable wait state generator. Internally generated wait states eliminate the potential system delays which come from generating wait states with external logic.

Wait states are programmed for each region in the memory region configuration table. The number of wait states is programmable over a range which allows efficient control of memory devices ranging from ultra-fast SRAMs to slow peripherals. An external ready signal is also provided for external wait state control.

The wait states which can be generated by the 80960CA are shown in Figure 4-4. In this table N is the number of wait states inserted. The wait states for read accesses and for write accesses are described by three parameters each. For read accesses, N_{RAD} is the number of states between the address cycle and the first data cycle and N_{RDD} is the number of states between consecutive data cycles in a burst access. For writes, N_{WAD} is the number of states that data is held after an address cycle, and N_{WDD} is the number of states that data is held for consecutive data cycles in a burst write. For both reads and writes, N_{XDA} is the number of dead cycles after the last data cycle and before the next address.

4.2.4.3 READY Control

The memory region configuration table allows the ready input ($\overline{READY}$) to be enabled or disabled for each region. If the ready input is disabled, the external input has no effect on the wait states generated for a memory access; all wait states are generated internally. If the ready input is enabled, it works in conjunction with the programmable wait state generator. In this case, the ready input has no effect until the number of programmed wait states has expired. When the wait state counter reaches 0, the ready input is sampled, and wait states continue or are terminated based on the value of the ready input. In order to gain complete external control over wait states, all wait state parameters for a region can be set to 0.

4.2.4.4 Pipelined Reads

The 80960CA BCU provides an address pipelining mode (Figure 4-5) to optimize the performance of instruction and data fetches from external memory. When the pipelined read mode is enabled, an address cycle overlaps with the last data cycle in each access, effectively reducing the total time needed for each access. Pipelining mode is selected in each region by programming the memory region configuration table.

4.2.4.5 Byte Ordering

One of two configurations for byte ordering, often referred to as little endian or big endian, is selected for each region by programming the memory region configuration table. The byte ordering options make the 80960CA capable of sharing memory with a processor which uses either byte ordering scheme. Byte ordering refers to the way that the 80960CA relates internal data to the way that data is stored or fetched from memory. The little endian configuration orders the bytes in a short-word or word so that the least significant byte of the quantity is positioned at the lowest address and the most significant byte at the highest address in memory. Conversely, for the big endian configuration, the least significant byte is positioned at the highest address, and the most significant byte at the lowest address. For example, for little endian ordering, byte 0 for word data would be found in memory at an address of the form XXXX XXX0H and, for big endian, at address XXXX XXX3H.

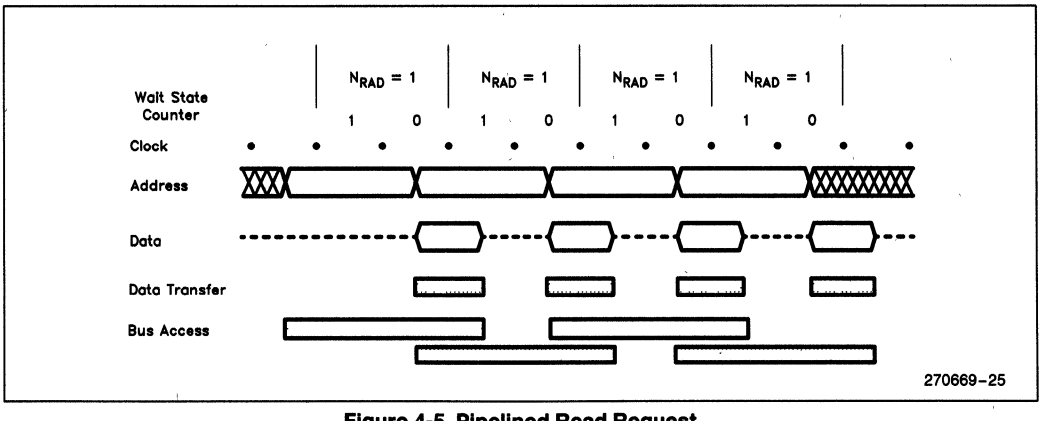


Figure 4-5. Pipelined Read Request

4.2.4.6 Data Alignment

The 80960CA can service any aligned or non-aligned bus request. Aligned requests are directed to their natural boundary in memory. In other words, the addresses for aligned requests are even multiples of the length of the data transferred. Non-aligned requests are not serviced directly by the BCU but are assisted by microcode. Microcode automatically breaks non-aligned requests into multiple aligned requests which are then reissued to the BCU. Depending on the degree of non-alignment and the length of the original request, the resulting requests by microcode will consist of a combination of byte, short-word, and double-word requests. The BCU is able to generate an operation-unaligned fault when a non-aligned bus request is first received. This fault can be selectively masked at initialization.

4.3 DMA Controller

The DMA controller is a high-performance, full-functioned integrated peripheral. The DMA controller can manage 4 channels of DMA transfer concurrent with program execution. Separate external control for each channel is provided. Each channel supports high-performance memory to memory transfers where the source and destination can be any combination of internal data RAM or external memory. The DMA Controller supports various types of transfers such as high-speed fly-by transfers and data chaining with the use of linked descriptor lists in memory.

The 80960CA's DMA controller is implemented using dedicated hardware and microcode. Because of the efficiency of the core, it is possible for the microcode to execute DMA transfers at high speeds. DMA transfers are performed by the core concurrently with execution of the user's program. Internal DMA logic is used for sampling requests, synchronizing transfers with external devices, and handling the service of multiple active channels.

4.3.1 SIGNAL DESCRIPTIONS

Twelve pins are dedicated to the DMA controller. Three pins are associated with each DMA channel. These pins are described below. In this description, the pin number corresponds to the channel number. For example, the $\overline{\text{DREQ0}}$ pin is the request pin for channel 0.

$\overline{\text{DREQ3}}$–$\overline{\text{DREQ0}}$ DMA Request (input)—This input indicates that an external device is requesting a DMA transfer. A DMA transfer refers to the complete transfer of one byte, short-word, word, or quad-word, depending on the transfer data width selected for the channel.

$\overline{\text{DACK3}}$–$\overline{\text{DACK0}}$ DMA Acknowledge (output)—This output becomes active when the requesting device is accessed.

$\overline{\text{EOP3/TC3}}$–$\overline{\text{EOP0/TC0}}$ End of Process (input) or Terminal Count (output)—This pin functions either as an input ($\overline{\text{EOPx}}$) or as an output ($\overline{\text{TCx}}$). When programmed as an output, the pin is driven active for one clock after byte count reaches zero and a DMA terminates. When programmed as an input, an external device can cause the DMA operation to terminate.

4.3.2 DMA TRANSFERS

The 80960CA DMA controller supports a variety of transfer modes and variations of these modes, allowing the DMA to adapt to a number of hardware systems and the performance requirements of these systems.

4.3.2.1 Standard Block and Demand Mode Transfers

A standard DMA transfer is made up of multiple bus requests. Loads from a source address are followed by stores to a destination address. The DMA controller issues the proper combination of these bus requests to execute the DMA transfer. For example, a typical DMA transfer between memory and an 8-bit peripheral could appear as a single byte load request directed to the source memory, followed by a single byte store request directed to the 8-bit peripheral.

The DMA controller has two basic transfer modes: block mode (unsynchronized) and demand mode (synchronized). Any DMA transfer will be serviced by one of these basic transfer modes.

A block mode DMA is initiated by software. Block mode DMAs are generally between memory. Block mode DMA transfers are not synchronized with any type of request from an external device. Once the DMA begins, it will continue until the entire block is complete or until it is suspended. The source and destination addresses for block mode transfers can be incremented or held constant for a DMA.

A demand mode DMA is controlled by an external device. Demand mode DMAs are generally between an external device and memory. In demand mode, each individual DMA transfer can be synchronized with a request. The request is signalled when an external device activates a DMA channel request pin ($\overline{\text{DREQ3}}$–$\overline{\text{DREQ0}}$). The DMA controller acknowledges this request with the DMA acknowledge pin ($\overline{\text{DACK3}}$–$\overline{\text{DACK0}}$) when the requesting device is accessed. A demand mode transfer may be synchronized with either the source or the destination device.

4.3.2.2 Fly-by Transfers

A fly-by transfer mode is provided for the most performance-critical DMA applications. Fly-by mode also makes very efficient use of the external bus during a DMA. Standard DMA transfers involve multiple bus requests: load requests directed to the source and a store request directed to the destination. Fly-by transfers only require a single bus request. For a fly-by transfer, memory sees a load or a store on the bus while the requesting device is selected by the DMA acknowledge pin. The data is never actually read from or written to the 80960CA. For memory to device transfers, the processor issues a load, and, while reading the memory, accesses the external device with the DMA acknowledge pin. The data is then written directly to the destination device with a single bus request. For a device to memory transfer, the reverse operation is performed. The DMA issues a store, and, while writing the memory, accesses the source device with the DMA acknowledge pin. In this case, the processor floats the data bus and the device's data is written directly into memory.

4.3.2.3 Data Chaining

Each DMA channel can be programmed in a data chaining mode. In this mode, all transfer information is taken from a linked-list descriptor in memory (Figure 4-6). Data chaining is started by specifying a pointer to a descriptor in memory. The transfer continues until the number of bytes in the byte count field in the descriptor is transferred. At this time, another linked-list descriptor may be executed. The next descriptor is specified by the next-pointer field in the current description. Data chaining continues until a null pointer is encountered in the next-pointer field. Data chaining can be designated as source chaining, destination chaining, or both.

In data chaining mode, an option exists which allows chaining descriptors to be updated while the DMA is running. When this option is enabled, the DMA sets a bit in the DMA's special function register after loading a descriptor and then checks this bit before loading the next descriptor. If the bit has been cleared by the user, the DMA continues; otherwise, the DMA waits for the next descriptor to be set up and for the user to clear the bit. An interrupt can be generated when each buffer is complete or when the DMA is terminated with a null pointer or the EOP pin.

4.3.3 TRANSFER CHARACTERISTICS

The DMA controller provides the programmer with a number of options for configuring the characteristics of a DMA transfer. Intelligent selection of transfer characteristics works to balance DMA performance and functionality with performance of the user program when the DMA is in progress.

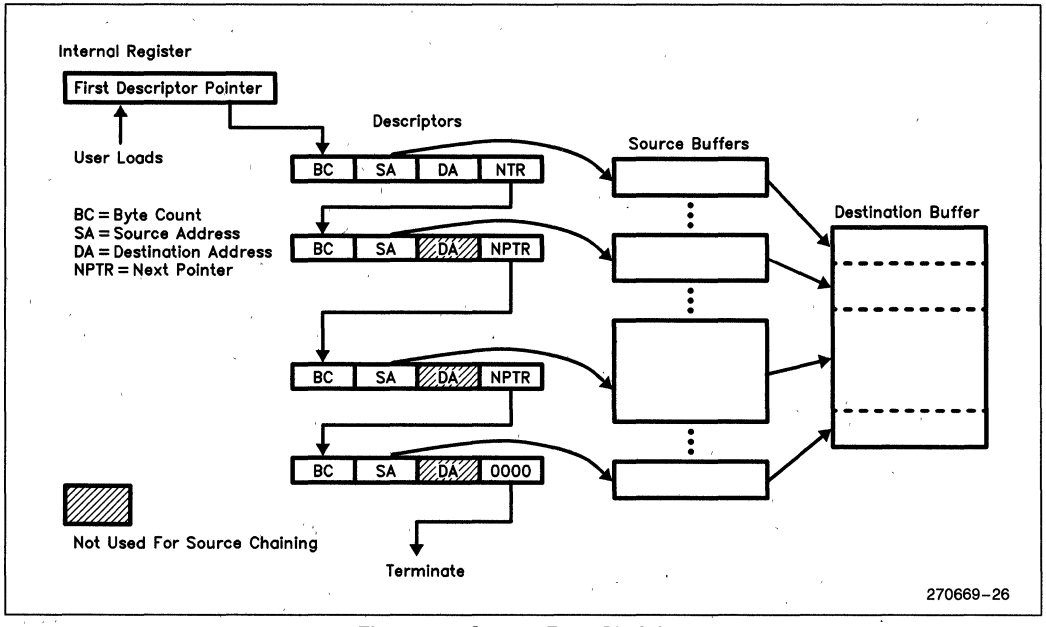


Figure 4-6. Source Data Chaining

The DMA controller provides features to optimize transfers by moving a maximum amount of data for each bus request issued. This is controlled by specifying the width of the source and destination directed bus requests for a DMA transfer, and by on-chip assembly or disassembly of the transfer when source and destination are not of equal widths.

Data alignment is performed automatically by the DMA controller when the source and destination of a transfer are not aligned. The alignment algorithm is optimized for many transfers, providing a performance comparable to the aligned transfer cases.

4.3.3.1 Transfer Data Length

The transfer data length specifies the length of bus requests directed to the source and destination in a standard DMA transfer. Byte, short, word, or quad-word loads and stores are selected for either source or destination when a DMA channel is set up. Assembly and disassembly of data is automatically performed when the source and destination widths are different. This feature provides the most efficient use of the bus when DMA transfers occur between a source and a destination with different external bus widths.

The DMA controller provides the option of using quad word transfers to enhance DMA performance. When quad transfers are specified, the DMA will request a four-word load request and four-word store request for each DMA transfer. The trade-off for the added DMA performance is latency on the external bus, preventing requests by the core, or by another DMA channel from being immediately serviced.

4.3.3.2 Data Alignment

The DMA controller supports transfer of source and destination data aligned to different byte boundaries in memory. The DMA implements microcode algorithms to transfer some non-aligned data with a performance level approaching that for aligned transfers. The DMA accomplishes this by attempting to issue the maximum number of aligned bus requests during a DMA (Figure 4-7). As shown, most of the overhead due to non-aligned DMAs is incurred at the beginning and end of the DMA. DMAs with low byte counts, therefore, do not benefit as much from the data alignment features of the DMA. The alignment feature is optimized for 8-bit to 8-bit, 32-bit to 32-bit and for 8-bit and 32-bit combinations of source and destination lengths.

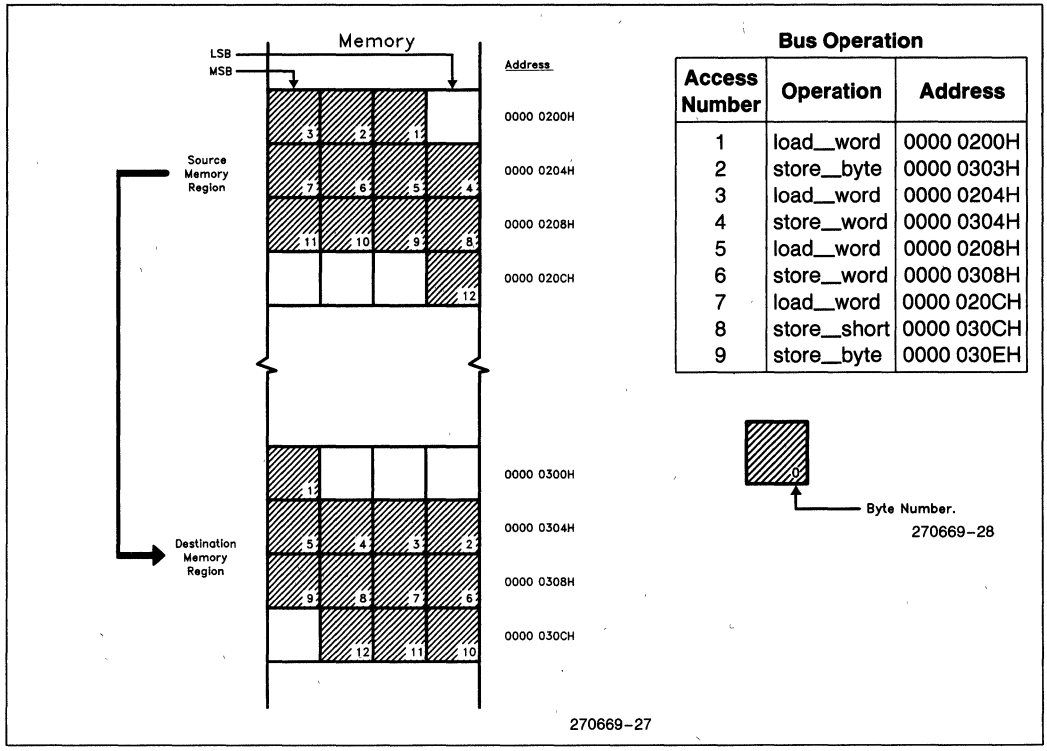


Figure 4-7. DMA Data Alignment

4.3.3.3 Channel Priority

The DMA controller arbitrates the priority of the 4 DMA channels. If multiple DMA channels are enabled, the DMA controller will determine in which order each channel is serviced.

The DMA controller can be configured in one of two priority modes, fixed mode or rotating mode. The fixed mode assumes a fixed priority for each channel with channel 0 having the highest priority, followed by channels 1, 2, and 3, with channel 3 having the lowest priority. The rotating mode updates a channel's priority to the lowest priority after that channel's DMA is made. This insures that a single channel is never locked out by other active channels. The priority sequence is always in the same order, with priority rotating from the low channel numbers to the high channel numbers.

4.3.3.4 Performance and Latency Considerations

DMA operations and the user program share the resources of the core and of the external bus. DMA performance and the performance of the user program are coupled directly to the balance of load sharing between these two processes. The core resources necessary to perform a DMA transfer vary depending on the way a channel has been configured. For example, byte assembly and disassembly requires more processor overhead per byte of transfer than does a transfer in which the source and destination transfer lengths are equal. The performance of a DMA is also tightly coupled to the user program's use of the external bus. If the user program does not make frequent bus requests, the requests by the DMA controller will be serviced with little or no delay.

The user can enhance performance of the DMA with trade-offs in system complexity and flexibility. Aligned transfers eliminate the microcode overhead needed to perform the internal alignments. DMAs between regions of equal transfer widths eliminate overhead for assembly and disassembly. Source or destination memory configured as burst memory will provide the most efficient use of the DMA controller when the quad-transfer feature is enabled. Using the fly-by mode reduces the number of bus requests needed for a DMA since fly-by mode uses only a single load or a single store request for each transfer.

4.3.4 DMA CONTROL AND CONFIGURATION

The DMA Controller uses an SFR register, the DMA command (DMAC) register, and the setup DMA (**sdma**) instruction for configuration and control of a DMA. The **sdma** instruction is used to configure each DMA channel. Transfer widths, byte count, source and destination addresses for a DMA are specified in this instruction.

The DMAC register (Figure 4-8) is described below.

The *channel enable field* enables a DMA once the channel is set up. Clearing these bits will also cause a DMA transfer to be suspended.

The *terminal count field* signals that byte count has reached zero and a DMA has ended.

The *channel active field* indicates that a channel is idle or active. If set, this bit indicates that the channel is active. This implies that the channel is servicing a transfer or has a request pending. The active bits are status information only.

The *channel done field* indicates that a DMA operation is complete. The done bits are status information only.

The *channel wait field* is used for handshaking with a user program in data chaining mode. The DMA sets these bits when a new linked-list descriptor is read. The DMA will not read the next descriptor until this bit is cleared by the user. The user can set up the next descriptor and then clear the channel wait bits to dynamically change descriptors.

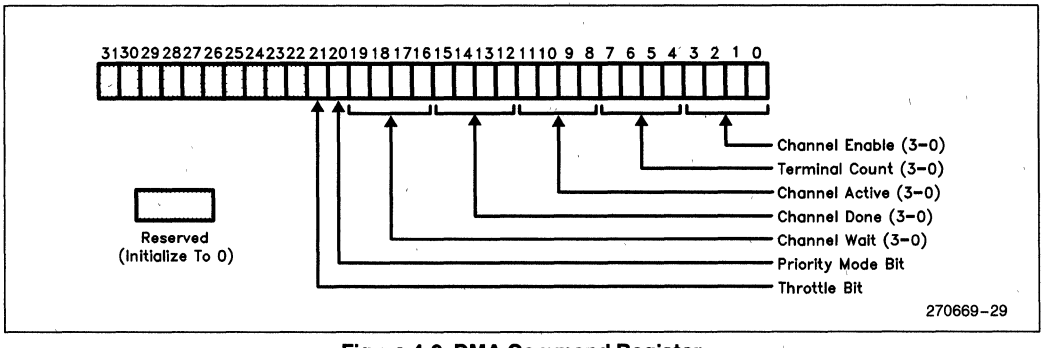


Figure 4-8. DMA Command Register

A *priority mode bit* selects rotating or fixed priority mode.

The *throttle bit* selects the maximum amount of core resources that the DMA microcode will receive in relation to the execution of the user program.

4.3.5 DMA INTERRUPTS

The DMA controller is the source of 4 hardware interrupts in the 80960CA. The DMA Controller can be programmed to request an interrupt when a DMA is complete, or when a buffer transfer is completed in chaining mode. Each channel requests a different interrupt.

4.4 Interrupt Controller

The 80960CA *Interrupt Controller* manages interrupts which are requested by external agents or by the DMA Controller. The interrupt controller manages 4 internal DMA interrupt sources, a single NMI (Non-Maskable Interrupt) pin, and 8 external interrupt pins. Up to 248 external interrupt sources can be supported by the interrupt controller. The interrupt controller handles the prioritization of software interrupts, hardware interrupts, and the process priority, and signals the core when interrupts are to be serviced. The interrupt controller provides the low-latency interrupt service featured on the 80960CA.

4.4.1 EXTERNAL INTERRUPTS

The 80960CA provides 8 interrupt pins and one NMI pin for detecting external requests. The interrupt controller allows the 8 interrupt pins to be configured as dedicated inputs capable of requesting 8 interrupts, or as a vectored input capable of requesting up to 248 interrupts. The NMI pin is always a dedicated input. The interrupt controller pins are described below.

$\overline{\text{XINT7}}$–$\overline{\text{XINT0}}$ External Interrupts (inputs)—These pins can be used as dedicated inputs, or acting together as an 8-bit number, request any interrupt. The inputs are edge or level detected, and are optionally debounced internally.

$\overline{\text{NMI}}$ Non-Maskable Interrupt (input)—NMI requests the highest priority interrupt. NMI is always taken and is not maskable (as the name implies), and not interruptable.

4.4.2 INTERRUPT MODES

The 8 external interrupt pins can be configured in one of three modes: dedicated mode, expanded mode, or mixed mode (Figure 4-9).

4.4.2.1 Dedicated Mode Interrupts

In dedicated mode, each of the 8 interrupt pins acts as a dedicated input. When an external event is detected on an interrupt pin, a unique interrupt is requested for that pin. It is possible to map each dedicated pin to one of a number of possible interrupt vectors. This is accomplished by programming the interrupt map (IMAP) control registers with an interrupt vector number for each pin. (Recall that interrupt vector numbers are 8-bit values which reference the 248 vectors in the interrupt table.)

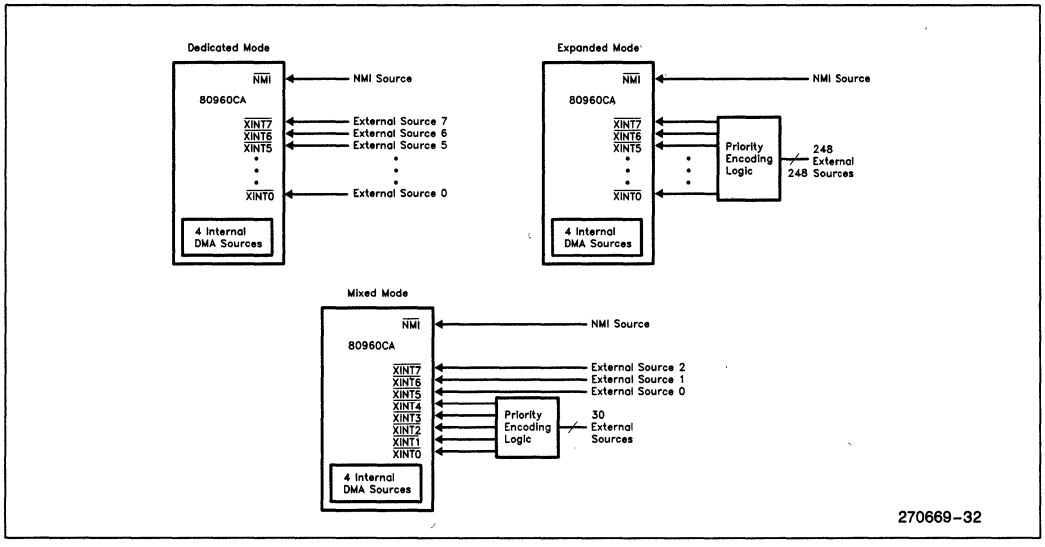


Figure 4-9. Interrupt Modes

Only the upper four bits of the vector number can be programmed for a dedicated mode interrupt. The lower four bits are fixed at the value 0010_2. With four programmable bits, one of 15 interrupt vectors is available for each dedicated pin. These interrupt vectors span the even priority levels from priority 2 to 30. The vector at priority 0 is not defined.

The 15 interrupt vectors available to dedicated sources can be cached in internal data RAM. If this interrupt vector caching feature is selected, the processor will automatically fetch the vector from data RAM, eliminating the latency caused by a bus request for a vector in external memory.

The DMA Controller can request four interrupts to signal the end of a DMA for each of four channels. The four interrupt signals from the DMA are handled by the interrupt controller in the same way as an interrupt pin configured as a dedicated input. Each of the four DMA sources may request one of 15 interrupts by programming the IMAP for that source.

4.4.2.2 Expanded Mode Interrupts

In expanded mode, external hardware considers the interrupt pins ($\overline{XINT0}$–$\overline{XINT7}$) as an 8-bit binary number. This number is used directly as the interrupt vector number. Each of the 248 possible interrupt vectors can be referenced in this way, allowing a separate external source for each vector. External hardware is responsible for recognizing individual hardware sources and then driving the interrupt vector number corresponding to that source onto the interrupt pins.

4.4.2.3 Mixed Mode Interrupts

In mixed mode, the 8 interrupt pins are divided into two functional sets. One set functions in dedicated mode, the other in expanded mode. In mixed mode, three pins are dedicated interrupt pins ($\overline{XINT7}$–$\overline{XINT5}$). A programmable vector number is associated with each of these pins. The remaining five interrupt pins ($\overline{XINT4}$–$\overline{XINT0}$) are treated as the most significant five bits of the expanded mode vector number. The lower order bits are internally forced to 010_2 to form the full 8-bit value for the vector number.

4.4.3 INTERRUPT CONTROLLER SETUP

The interrupt controller uses two special function registers to manage interrupt requests by hardware sources. The hardware interrupt pending register (IPND) and the hardware interrupt mask register (IMSK) are addressed as sf0 and sf1 respectively. A single bit in each register corresponds to each of the 8 possible external sources and 4 DMA sources for hardware interrupts. The IMSK register performs the function of masking hardware interrupts and the IPND register implements posting of interrupts requested by hardware. When configured for expanded or mixed mode interrupts, bit 0 of the IMSK register globally masks the expanded mode interrupts.

4.4.4. NON-MASKABLE INTERRUPT

In addition to the maskable hardware interrupts, a single *Non-Maskable Interrupt (NMI)* is provided. A dedicated NMI pin is used to request this interrupt. NMI is defined as a higher priority than any hardware interrupt, software interrupt, or process priority. The NMI procedure, therefore, can never be interrupted and must execute the return instruction before other procedures can execute. The NMI procedure is entered through vector 248. This vector is cached in internal data RAM at initialization to reduce latency for the NMI.

APPENDIX A
80960CA CORE IMPLEMENTATION

The 80960CA Core is a high-performance implementation of the 80960 Core Architecture. This section briefly describes the microarchitecture of the 80960CA core and the key constructs used to achieve parallel instruction execution.

The 80960CA core can be divided into the 6 main subunits listed below.

— Instruction Sequencer
— Register File
— Execution Unit
— Multiply and Divide Unit
— Address Generation Unit
— Static Data RAM and Local Register Cache

Figure A-1 is a simple block diagram of the 80960CA. The nucleus of the processor is the Instruction Sequencer and Register File. The other subunits of the core, referred to as coprocessors, radiate from these units, connecting to either the register (REG) side or the memory (MEM) side of the processor. The Instruction Sequencer issues directives, via the REG and MEM interfaces, which target a specific coprocessor. That coprocessor then executes an express function virtually decoupled from the IS and the other coprocessors. The REG and MEM data busses shown in Figure A-1 are used to transfer data between the common Register File and the coprocessors.

A.1 Instruction Sequencer

The *Instruction Sequencer (IS)* decodes the instruction stream and drives the decoded instruction stream onto the coprocessor interfaces. In a single clock, the IS decodes up to 4 instruction and issues up to three of these instructions to the on-chip coprocessors or to the IS itself. One register (REG) format, one memory (MEM) format, and one control or control and branch (CTRL or COBR) format instruction can be issued at one time. These instructions are directed respectively to the REG coprocessors, the MEM coprocessors, or to the IS. The ability to issue multiple instructions in parallel can result in the simultaneous execution of many instructions at once. An optimizing compiler or hand optimization of assembly code can easily produce an instruction stream which takes full advantage of the parallel execution of the core.

A technique known as *resource scoreboarding* is used to manage the parallel execution of instructions and the common resources of the processor. A coprocessor, for example, can scoreboard itself, indicating that it cannot

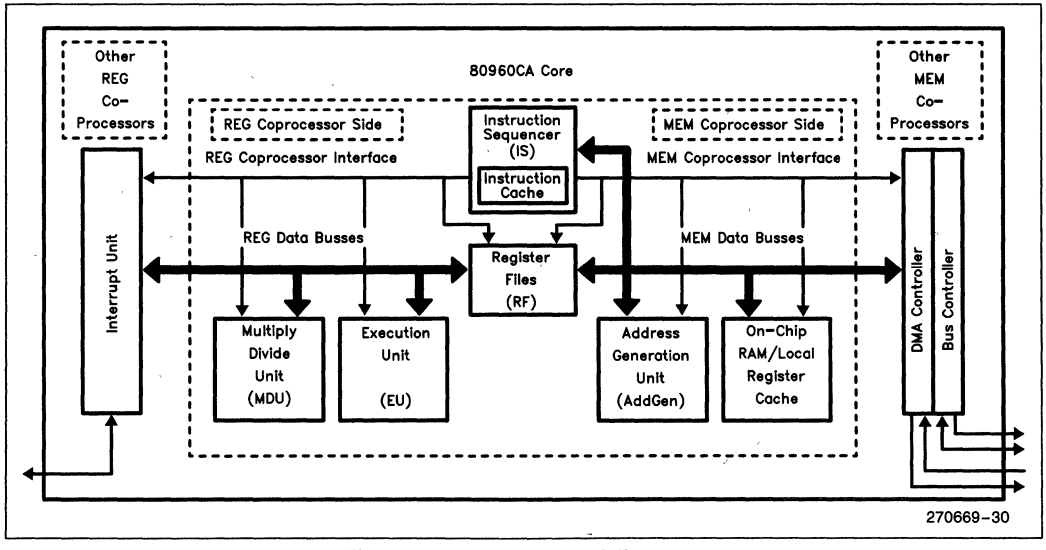


Figure A-1. 80960CA Block Diagram

act on another instruction until an instruction currently executing on that coprocessor is completed. A specific form of resource scoreboarding is referred to as *register scoreboarding*. When the computation stage of an instruction takes more than one clock, the destination register or registers for the result are scoreboarded as busy. A subsequent operation needing that particular register will be delayed until the multi-clock operation is completed. Instructions which do not use the scoreboarded registers can be executed in parallel.

The IS manages a three stage parallel instruction pipeline (Figure A-2). In the first stage of the pipeline (pipe 0), the address of the next instruction is calculated. This address may be the next sequential instruction, the target of a branch, or a location in microcode. In the second stage of the pipeline (pipe 1), the instructions are issued to the rest of the machine. In the third stage (pipe 2), the instruction computation is started, and for single cycle instructions, a result is returned.

Several microarchitectural features of the core are designed to minimize performance loss due to pipeline breaks.

Branch Prediction—To minimize pipeline breaks due to branching, the user can specify the direction that a conditional branch instruction will usually follow. The processor will execute along the specified instruction path with no pipeline break. If the branch direction specified was the direction actually selected by execution of the conditional branch, no pipeline break occurs. The direction of the branch guess is determined by a bit value in the CTRL format instructions.

Register Bypassing—Register bypassing is a feature which forwards the result of an instruction for immediate use as the source of another instruction. This forwarding occurs at the same time that the value is written to its destination register. Bypassing the register file saves the one clock cycle break which would otherwise occur while waiting for the value to be written to the register file and the register scoreboard to be cleared.

On-chip Cache—The on-chip instruction cache and local register cache eliminate many pipeline breaks which will occur if the IS is forced to wait for code or data to be moved between the 80960CA and external memory.

Register File Access—The Register File allows multiple instructions to gain access to the register set simultaneously. This eliminates pipeline breaks which would be caused by a loss of access to the register set by any coprocessor.

A.1.1 INSTRUCTION CACHE

The IS includes a 1 Kbyte two-way set associative instruction cache capable of delivering up to four instructions each clock to the Instruction Sequencer. The cache allows inner loops of code to execute with no external instruction fetches.

A.1.2 MICROCODE ROM

The 80960CA uses *microcode ROM* to implement complex instructions and functions. This includes calls, returns, DMA transfers, and initialization sequences. Microcode provides an inexpensive and simple method for implementing complex instructions in the mostly RISC environment of the 80960CA. When the IS encounters a microcoded instruction, it automatically branches to the microcode routine. The 80960CA performs this microcode branch in 0 clocks.

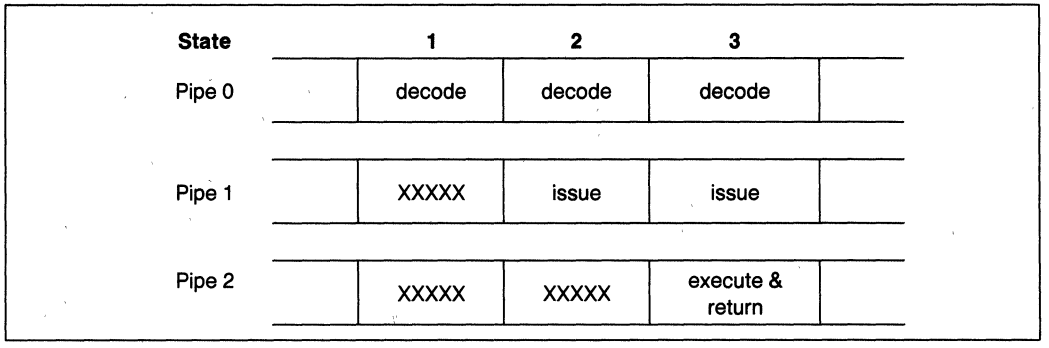


Figure A-2. Instruction Pipeline

A.2 Register File

The *Register File (RF)* contains the 16 local and 16 global registers. The register file has six ports (Figure A-3), allowing parallel access of the register set by several 80960CA coprocessors. This parallel access results in an ability to execute one simple logic or arithmetic instruction, one memory operation (load/store), and one address calculation per clock.

MEM coprocessors interface to the RF with a 128-bit wide load bus and a 128-bit wide store bus. These busses enable movement of up to 4 words per clock to and from the RF. These busses also allow LOAD data from a previous read access and STORE data from a current write access to be processed in the register file simultaneously. An additional 32-bit port allows an address or address reduction operand to be simultaneously fetched by the Address Generation Unit.

REG coprocessors interface to the RF with two 64-bit source busses and a single 64-bit destination bus. With this bus structure, two source operands are simultaneously issued to a REG coprocessor when an instruction is issued. A 64-bit destination bus allows the result from the previous operation to be written to the RF at the same time that the current operation's source operands are issued.

A.3 Execution Unit

The Execution Unit is the 32-bit Arithmetic and Logic Unit of the 80960CA Core. The EU can be viewed as a self-contained REG coprocessor with its own instruction set. As such, the EU is responsible for executing or supporting the execution of all the integer and ordinal arithmetic instructions, the logic and shift instructions, the move instructions, the bit and bit field instructions, and the compare operations. The EU performs any arithmetic or logical instructions in a single clock.

A.4 Multiply Divide Unit

The *Multiply and Divide Unit (MDU)* is a REG coprocessor which performs integer and ordinal multiply, divide, remainder, and modulo operations. The MDU detects integer overflow and divide by zero errors. The MDU is optimized for multiplication, performing 32-bit multiplies in 4 clocks. The MDU performs multiplies and divides in parallel with the main execution unit.

A.5 Address Generation Unit

The *Address Generation Unit (AGU)* is a MEM coprocessor which computes the effective addresses for memory operations. It directly executes the load address instruction (**lda**) and calculates addresses for loads and stores based on the addressing mode specified in these instructions. The address calculations are performed in parallel with the main execution unit (EU).

A.6 Data RAM and Local Register Cache

The *Data RAM and Local Register Cache* is part of a 1.5 Kbyte block of on-chip Static RAM (SRAM). 1 Kbyte of this SRAM is mapped into the 80960CA's address space from location 00000000H to 000003FFH. A portion of the remaining 512 bytes is dedicated to the Local Register Cache. This part of internal SRAM is not directly visible to the user. Loads and Stores, including quad-word accesses, to the internal SRAM are typically performed in only one clock. The complete local register set, therefore, can be moved to the local register cache in only four clocks.

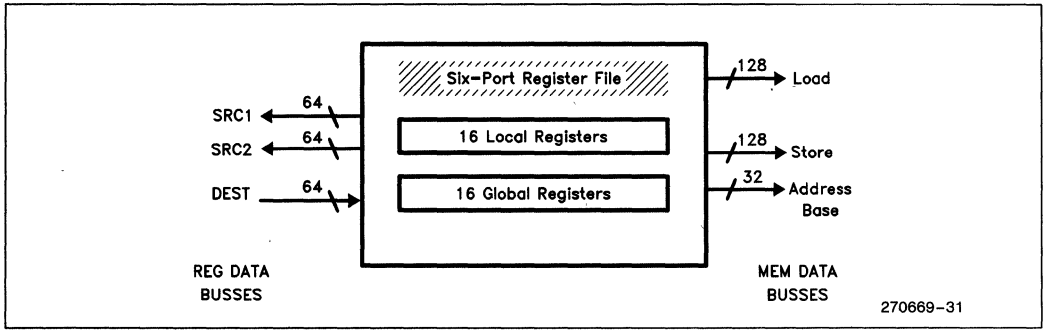


Figure A-3. Six-Port Register File

intel

80960CA-33, -25, -16
32-BIT HIGH-PERFORMANCE EMBEDDED PROCESSOR

• Two Instructions/Clock Sustained Execution
• Four 59 Mbytes/s DMA Channels with Data Chaining
• Demultiplexed 32-bit Burst Bus with Pipelining

- **32-bit Parallel Architecture**
  - **Two Instructions/clock Execution**
  - **Load/Store Architecture**
  - **Sixteen 32-bit Global Registers**
  - **Sixteen 32-bit Local Registers**
  - **Manipulates 64-bit Bit Fields**
  - **11 Addressing Modes**
  - **Full Parallel Fault Model**
  - **Supervisor Protection Model**
- **Fast Procedure Call/Return Model**
  - **Full Procedure Call in 4 Clocks**
- **On-Chip Register Cache**
  - **Caches Registers on Call/Ret**
  - **Minimum of 6 Frames Provided**
  - **Up to 15 Programmable Frames**
- **On-Chip Instruction Cache**
  - **1 Kbyte Two-Way Set Associative**
  - **128-bit Path to Instruction Sequencer**
  - **Cache-Lock Modes**
  - **Cache-Off Mode**
- **High Bandwidth On-Chip Data RAM**
  - **1 Kbyte On-Chip Data RAM**
  - **Sustains 128 bits per Clock Access**
- **Four On-Chip DMA Channels**
  - **59 Mbytes/s Fly-by Transfers**
  - **32 Mbytes/s Two-Cycle Transfers**
  - **Data Chaining**
  - **Data Packing/Unpacking**
  - **Programmable Priority Method**
- **32-Bit Demultiplexed Burst Bus**
  - **128-bit Internal Data Paths to *and* from Registers**
  - **Burst Bus for DRAM Interfacing**
  - **Address Pipelining Option**
  - **Fully Programmable Wait States**
  - **Supports 8-, 16- or 32-bit Bus Widths**
  - **Supports Unaligned Accesses**
  - **Supervisor Protection Pin**
- **Selectable Big or Little Endian Byte Ordering**
- **High-Speed Interrupt Controller**
  - **Up to 248 External Interrupts**
  - **32 Fully Programmable Priorities**
  - **Multi-mode 8-bit Interrupt Port**
  - **Four Internal DMA Interrupts**
  - **Separate, Non-maskable Interrupt Pin**
  - **Context Switch in 750 ns Typical**

January 1994
Order Number: 270727-006

80960CA-33, -25, -16
32-BIT HIGH-PERFORMANCE EMBEDDED PROCESSOR

CONTENTS

CONTENTS

PAGE

LIST OF FIGURES

CONTENTS

PAGE

LIST OF FIGURES (Continued)

LIST OF TABLES

1.0 PURPOSE

This document provides electrical characteristics for the 33, 25 and 16 MHz versions of the 80960CA. For a detailed description of any 80960CA functional topic—other than parametric performance—consult the *80960CA Product Overview* (Order No. 270669) or the *i960® CA Microprocessor User's Manual* (Order No. 270710). To obtain data sheet updates and errata, please call Intel's FaxBACK® data-on-demand system (1-800-628-2283 or 916-356-3105). Other information can be obtained from Intel's technical BBS (916-356-3600).

2.0 80960CA OVERVIEW

The 80960CA is the second-generation member of the 80960 family of embedded processors. The 80960CA is object code compatible with the 32-bit 80960 Core Architecture while including Special Function Register extensions to control on-chip peripherals and instruction set extensions to shift 64-bit operands and configure on-chip hardware. Multiple 128-bit internal buses, on-chip instruction caching and a sophisticated instruction scheduler allow the processor to sustain execution of two instructions every clock and peak at execution of three instructions per clock.

A 32-bit demultiplexed and pipelined burst bus provides a 132 Mbyte/s bandwidth to a system's high-speed external memory sub-system. In addition, the 80960CA's on-chip caching of instructions, procedure context and critical program data substantially decouple system performance from the wait states associated with accesses to the system's slower, cost sensitive, main memory subsystem.

The 80960CA bus controller integrates full wait state and bus width control for highest system performance with minimal system design complexity. Unaligned access and Big Endian byte order support reduces the cost of porting existing applications to the 80960CA.

The processor also integrates four complete data-chaining DMA channels and a high-speed interrupt controller on-chip. DMA channels perform: single-cycle or two-cycle transfers, data packing and unpacking and data chaining. Block transfers—in addition to source or destination synchronized transfers—are provided.

The interrupt controller provides full programmability of 248 interrupt sources into 32 priority levels with a typical interrupt task switch ("latency") time of 750 ns.

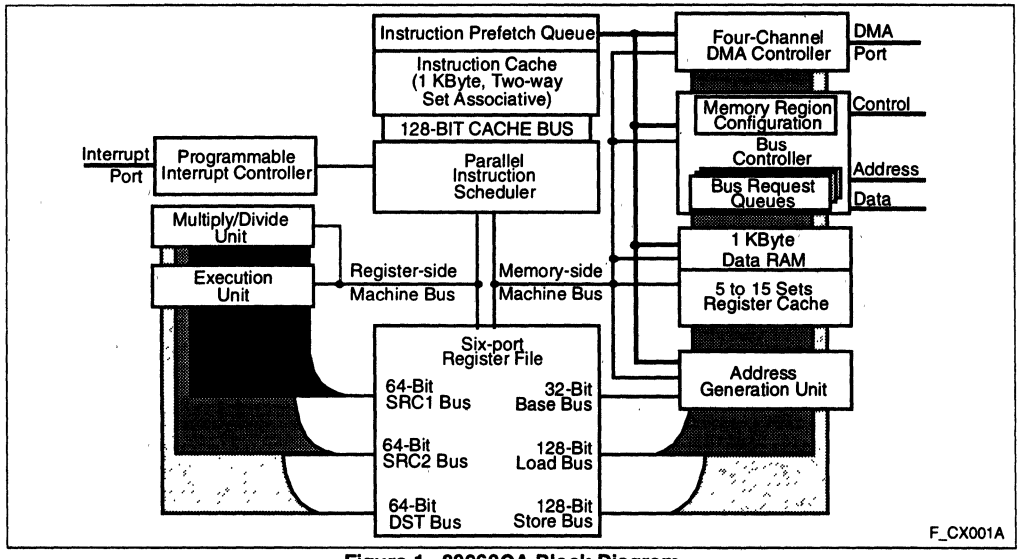


Figure 1. 80960CA Block Diagram

2.1 The C-Series Core

The C-Series core is a very high performance microarchitectural implementation of the 80960 Core Architecture. The C-Series core can sustain execution of two instructions per clock (66 MIPs at 33 MHz). To achieve this level of performance, Intel has incorporated state-of-the-art silicon technology and innovative microarchitectural constructs into the implementation of the C-Series core. Factors that contribute to the core's performance include:

- Parallel instruction decoding allows issuance of up to three instructions per clock
- Single-clock execution of most instructions
- Parallel instruction decode allows sustained, simultaneous execution of two single-clock instructions every clock cycle
- Efficient instruction pipeline minimizes pipeline break losses
- Register and resource scoreboarding allow simultaneous multi-clock instruction execution
- Branch look-ahead and prediction allows many branches to execute with no pipeline break
- Local Register Cache integrated on-chip caches Call/Return context
- Two-way set associative, 1 Kbyte integrated instruction cache
- 1 Kbyte integrated Data RAM sustains a four-word (128-bit) access every clock cycle

2.2 Pipelined, Burst Bus

A 32-bit high performance bus controller interfaces the 80960CA to external memory and peripherals. The Bus Control Unit features a maximum transfer rate of 132 Mbytes per second (at 33 MHz). Internally programmable wait states and 16 separately configurable memory regions allow the processor to interface with a variety of memory subsystems with a minimum of system complexity and a maximum of performance. The Bus Controller's main features include:

- Demultiplexed, Burst Bus to exploit most efficient DRAM access modes
- Address Pipelining to reduce memory cost while maintaining performance
- 32-, 16- and 8-bit modes for I/O interfacing ease
- Full internal wait state generation to reduce system cost
- Little and Big Endian support to ease application development
- Unaligned access support for code portability
- Three-deep request queue to decouple the bus from the core

2.3 Flexible DMA Controller

A four-channel DMA controller provides high speed DMA control for data transfers involving peripherals and memory. The DMA provides advanced features such as data chaining, byte assembly and disassembly and a high performance fly-by mode capable of transfer speeds of up to 59 Mbytes per second at 33 MHz. The DMA controller features a performance and flexibility which is only possible by integrating the DMA controller and the 80960CA core.

2.4 Priority Interrupt Controller

A programmable-priority interrupt controller manages up to 248 external sources through the 8-bit external interrupt port. The Interrupt Unit also handles the four internal sources from the DMA controller and a single non-maskable interrupt input. The 8-bit interrupt port can also be configured to provide individual interrupt sources that are level or edge triggered.

Interrupts in the 80960CA are prioritized and signaled within 270 ns of the request. If the interrupt is of higher priority than the processor priority, the context switch to the interrupt routine typically is complete in another 480 ns. The interrupt unit provides the mechanism for the low latency and high throughput interrupt service which is essential for embedded applications.

2.5 Instruction Set Summary

Table 1 summarizes the 80960CA instruction set by logical groupings. See the *i960® CA Microprocessor User's Manual* for a complete description of the instruction set.

Table 1. 80960CA Instruction Set

Data Movement	Arithmetic	Logical	Bit and Bit Field and Byte
Load	Add	And	Set Bit
Store	Subtract	Not And	Clear Bit
Move	Multiply	And Not	Not Bit
Load Address	Divide	Or	Alter Bit
	Remainder	Exclusive Or	Scan For Bit
	Modulo	Not Or	Span Over Bit
	Shift	Or Not	Extract
	*Extended Shift	Nor	Modify
	Extended Multiply	Exclusive Nor	Scan Byte for Equal
	Extended Divide	Not	
	Add with Carry	Nand	
	Subtract with Carry		
	Rotate		
Comparison	**Branch**	**Call/Return**	**Fault**
Compare	Unconditional Branch	Call	Conditional Fault
Conditional Compare	Conditional Branch	Call Extended	Synchronize Faults
Compare and Increment	Compare and Branch	Call System	
		Return	
Compare and Decrement		Branch and Link	
Test Condition Code			
Check Bit			
Debug	**Processor Management**	**Atomic**	
Modify Trace Controls	Flush Local Registers	Atomic Add	
Mark	Modify Arithmetic Controls	Atomic Modify	
Force Mark	Modify Process Controls		
	*System Control		
	*DMA Control		

NOTES:
Instructions marked by (*) are 80960CA extensions to the 80960 instruction set.

3.0 PACKAGE INFORMATION

3.1 Package Introduction

This section describes the pins, pinouts and thermal characteristics for the 80960CA in the 168-pin Ceramic Pin Grid Array (PGA) package and the 196-pin Plastic Quad Flat Package (PQFP). For complete package specifications and information, see the *Packaging* Handbook (Order No. 240800).

3.2 Pin Descriptions

The 80960CA pins are described in this section. Table 2 presents the legend for interpreting the pin descriptions in the following tables. Pins associated with the 32-bit demultiplexed processor bus are described in Table 3. Pins associated with basic processor configuration and control are described in Table 4. Pins associated with the 80960CA DMA Controller and Interrupt Unit are described in Table 5.

All pins float while the processor is in the ONCE mode.

Table 2. Pin Description Nomenclature

Symbol	Description
I	Input only pin
O	Output only pin
I/O	Pin can be either an input or output
–	Pins "must be" connected as described
S(...)	Synchronous. Inputs must meet setup and hold times relative to PCLK2:1 for proper operation. All outputs are synchronous to PCLK2:1. S(E) Edge sensitive input S(L) Level sensitive input
A(...)	Asynchronous. Inputs may be asynchronous to PCLK2:1. A(E) Edge sensitive input A(L) Level sensitive input
H(...)	While the processor's bus is in the Hold Acknowledge or Bus Backoff state, the pin: H(1) is driven to V_{CC} H(0) is driven to V_{SS} H(Z) floats H(Q) continues to be a valid input
R(...)	While the processor's $\overline{\text{RESET}}$ pin is low, the pin: R(1) is driven to V_{CC} R(0) is driven to V_{SS} R(Z) floats R(Q) continues to be a valid output

Table 3. 80960CA Pin Description — External Bus Signals

Name	Type	Description
A31:2	**O** S H(Z) R(Z)	**ADDRESS BUS** carries the physical address' upper 30 bits. A31 is the most significant address bit; A2 is the least significant. During a bus access, A31:2 identify all external addresses to word (4-byte) boundaries. The byte enable signals indicate the selected byte in each word. During burst accesses, A3:2 increment to indicate successive data cycles.
D31:0	**I/O** S(L) H(Z) R(Z)	**DATA BUS** carries 32-, 16- or 8-bit data quantities depending on bus width configuration. The least significant bit of the data is carried on D0 and the most significant on D31. When the bus is configured for 8-bit data, the lower 8 data lines, D7:0 are used. For 16-bit data bus widths, D15:0 are used. For 32-bit bus widths the full data bus is used.
$\overline{BE3:0}$	**O** S H(Z) R(1)	**BYTE ENABLES** select which of the four bytes addressed by A31:2 are active during an access to a memory region configured for a 32-bit data-bus width. $\overline{BE3}$ applies to D31:24; $\overline{BE2}$ applies to D23:16; $\overline{BE1}$ applies to D15:8 $\overline{BE0}$ applies to D7:0. 32-bit bus: $\overline{BE3}$ –Byte Enable 3 –enable D31:24 $\overline{BE2}$ –Byte Enable 2 –enable D23:16 $\overline{BE1}$ –Byte Enable 1 –enable D15:8 $\overline{BE0}$ –Byte Enable 0 –enable D7:0 For accesses to a memory region configured for a 16-bit data-bus width, the processor uses the $\overline{BE3}$, $\overline{BE1}$ and $\overline{BE0}$ pins as $\overline{BHE}$, A1 and $\overline{BLE}$ respectively. 16-bit bus: $\overline{BE3}$ –Byte High Enable ($\overline{BHE}$) –enable D15:8 $\overline{BE2}$ –Not used (driven high or low) $\overline{BE1}$ –Address Bit 1 (A1) $\overline{BE0}$ –Byte Low Enable ($\overline{BLE}$) –enable D7:0 For accesses to a memory region configured for an 8-bit data-bus width, the processor uses the $\overline{BE1}$ and $\overline{BE0}$ pins as A1 and A0 respectively. 8-bit bus: $\overline{BE3}$ –Not used (driven high or low) $\overline{BE2}$ –Not used (driven high or low) $\overline{BE1}$ –Address Bit 1 (A1) $\overline{BE0}$ –Address Bit 0 (A0)
W/$\overline{R}$	**O** S H(Z) R(0)	**WRITE/READ** is asserted for read requests and deasserted for write requests. The W/$\overline{R}$ signal changes in the same clock cycle as $\overline{ADS}$. It remains valid for the entire access in non-pipelined regions. In pipelined regions, W/$\overline{R}$ is not guaranteed to be valid in the last cycle of a read access.
$\overline{ADS}$	**O** S H(Z) R(1)	**ADDRESS STROBE** indicates a valid address and the start of a new bus access. $\overline{ADS}$ is asserted for the first clock of a bus access.

Table 3. 80960CA Pin Description — External Bus Signals (Continued)

Name	Type	Description
$\overline{\text{READY}}$	I S(L) H(Z) R(Z)	**READY** is an input which signals the termination of a data transfer. $\overline{\text{READY}}$ is used to indicate that read data on the bus is valid or that a write-data transfer has completed. The $\overline{\text{READY}}$ signal works in conjunction with the internally programmed wait-state generator. If $\overline{\text{READY}}$ is enabled in a region, the pin is sampled after the programmed number of wait-states has expired. If the $\overline{\text{READY}}$ pin is deasserted, wait states continue to be inserted until $\overline{\text{READY}}$ becomes asserted. This is true for the N_{RAD}, N_{RDD}, N_{WAD} and N_{WDD} wait states. The N_{XDA} wait states cannot be extended.
$\overline{\text{BTERM}}$	I S(L) H(Z) R(Z)	**BURST TERMINATE** is an input which breaks up a burst access and causes another address cycle to occur. The $\overline{\text{BTERM}}$ signal works in conjunction with the internally programmed wait-state generator. If $\overline{\text{READY}}$ and $\overline{\text{BTERM}}$ are enabled in a region, the $\overline{\text{BTERM}}$ pin is sampled after the programmed number of wait states has expired. When $\overline{\text{BTERM}}$ is asserted, a new ADS signal is generated and the access is completed. The $\overline{\text{READY}}$ input is ignored when $\overline{\text{BTERM}}$ is asserted. $\overline{\text{BTERM}}$ must be externally synchronized to satisfy $\overline{\text{BTERM}}$ setup and hold times.
$\overline{\text{WAIT}}$	O S H(Z) R(1)	**WAIT** indicates internal wait state generator status. $\overline{\text{WAIT}}$ is asserted when wait states are being caused by the internal wait state generator and not by the $\overline{\text{READY}}$ or $\overline{\text{BTERM}}$ inputs. $\overline{\text{WAIT}}$ can be used to derive a write-data strobe. $\overline{\text{WAIT}}$ can also be thought of as a $\overline{\text{READY}}$ output that the processor provides when it is inserting wait states.
$\overline{\text{BLAST}}$	O S H(Z) R(0)	**BURST LAST** indicates the last transfer in a bus access. $\overline{\text{BLAST}}$ is asserted in the last data transfer of burst and non-burst accesses after the wait state counter reaches zero. $\overline{\text{BLAST}}$ remains asserted until the clock following the last cycle of the last data transfer of a bus access. If the $\overline{\text{READY}}$ or $\overline{\text{BTERM}}$ input is used to extend wait states, the $\overline{\text{BLAST}}$ signal remains asserted until $\overline{\text{READY}}$ or $\overline{\text{BTERM}}$ terminates the access.
DT/$\overline{\text{R}}$	O S H(Z) R(0)	**DATA TRANSMIT/RECEIVE** indicates direction for data transceivers. DT/$\overline{\text{R}}$ is used in conjunction with $\overline{\text{DEN}}$ to provide control for data transceivers attached to the external bus. When DT/$\overline{\text{R}}$ is asserted, the signal indicates that the processor receives data. Conversely, when deasserted, the processor sends data. DT/$\overline{\text{R}}$ changes only while $\overline{\text{DEN}}$ is high.
$\overline{\text{DEN}}$	O S H(Z) R(1)	**DATA ENABLE** indicates data cycles in a bus request. $\overline{\text{DEN}}$ is asserted at the start of the bus request first data cycle and is deasserted at the end of the last data cycle. $\overline{\text{DEN}}$ is used in conjunction with DT/$\overline{\text{R}}$ to provide control for data transceivers attached to the external bus. $\overline{\text{DEN}}$ remains asserted for sequential reads from pipelined memory regions. $\overline{\text{DEN}}$ is deasserted when DT/$\overline{\text{R}}$ changes.
$\overline{\text{LOCK}}$	O S H(Z) R(1)	**BUS LOCK** indicates that an atomic read-modify-write operation is in progress. $\overline{\text{LOCK}}$ may be used to prevent external agents from accessing memory which is currently involved in an atomic operation. $\overline{\text{LOCK}}$ is asserted in the first clock of an atomic operation and deasserted in the clock cycle following the last bus access for the atomic operation. To allow the most flexibility for memory system enforcement of locked accesses, the processor acknowledges a bus hold request when $\overline{\text{LOCK}}$ is asserted. The processor performs DMA transfers while $\overline{\text{LOCK}}$ is active.
HOLD	I S(L) H(Z) R(Z)	**HOLD REQUEST** signals that an external agent requests access to the external bus. The processor asserts HOLDA after completing the current bus request. HOLD, HOLDA and BREQ are used together to arbitrate access to the processor's external bus by external bus agents.

Table 3. 80960CA Pin Description — External Bus Signals (Continued)

Name	Type	Description
$\overline{\text{BOFF}}$	I S(L) H(Z) R(Z)	**BUS BACKOFF**, when asserted, suspends the current access and causes the bus pins to float. When $\overline{\text{BOFF}}$ is deasserted, the $\overline{\text{ADS}}$ signal is asserted on the next clock cycle and the access is resumed.
HOLDA	O S H(1) R(Q)	**HOLD ACKNOWLEDGE** indicates to a bus requestor that the processor has relinquished control of the external bus. When HOLDA is asserted, the external address bus, data bus and bus control signals are floated. HOLD, $\overline{\text{BOFF}}$, HOLDA and BREQ are used together to arbitrate access to the processor's external bus by external bus agents. Since the processor grants HOLD requests and enters the Hold Acknowledge state even while $\overline{\text{RESET}}$ is asserted, the state of the HOLDA pin is independent of the $\overline{\text{RESET}}$ pin.
BREQ	O S H(Q) R(0)	**BUS REQUEST** is asserted when the bus controller has a request pending. BREQ can be used by external bus arbitration logic in conjunction with HOLD and HOLDA to determine when to return mastership of the external bus to the processor.
D/$\overline{\text{C}}$	O S H(Z) R(Z)	**DATA OR CODE** is asserted for a data request and deasserted for instruction requests. D/$\overline{\text{C}}$ has the same timing as W/$\overline{\text{R}}$.
$\overline{\text{DMA}}$	O S H(Z) R(Z)	**DMA ACCESS** indicates whether the bus request was initiated by the DMA controller. $\overline{\text{DMA}}$ is asserted for any DMA request. $\overline{\text{DMA}}$ is deasserted for all other requests.
$\overline{\text{SUP}}$	O S H(Z) R(Z)	**SUPERVISOR ACCESS** indicates whether the bus request is issued while in supervisor mode. $\overline{\text{SUP}}$ is asserted when the request has supervisor privileges and is deasserted otherwise. $\overline{\text{SUP}}$ can be used to isolate supervisor code and data structures from non-supervisor requests.

Table 4. 80960CA Pin Description — Processor Control Signals

Name	Type	Description
$\overline{\text{RESET}}$	I A(L) H(Z) R(Z)	**RESET** causes the chip to reset. When $\overline{\text{RESET}}$ is asserted, all external signals return to the reset state. When $\overline{\text{RESET}}$ is deasserted, initialization begins. When the 2-x clock mode is selected, $\overline{\text{RESET}}$ must remain asserted for 32 CLKIN cycles before being deasserted to guarantee correct processor initialization. When the 1-x clock mode is selected, $\overline{\text{RESET}}$ must remain asserted for 10,000 CLKIN cycles before being deasserted to guarantee correct processor initialization. The CLKMODE pin selects 1-x or 2-x input clock division of the CLKIN pin. The processor's Hold Acknowledge bus state functions while the chip is reset. If the processor's bus is in the Hold Acknowledge state when $\overline{\text{RESET}}$ is asserted, the processor will internally reset, but maintains the Hold Acknowledge state on external pins until the Hold request is removed. If a Hold request is made while the processor is in the reset state, the processor bus will grant HOLDA and enter the Hold Acknowledge state.
$\overline{\text{FAIL}}$	O S H(Q) R(0)	**FAIL** indicates failure of the processor's self-test performed at initialization. When $\overline{\text{RESET}}$ is deasserted and the processor begins initialization, the $\overline{\text{FAIL}}$ pin is asserted. An internal self-test is performed as part of the initialization process. If this self-test passes, the $\overline{\text{FAIL}}$ pin is deasserted; otherwise it remains asserted. The $\overline{\text{FAIL}}$ pin is reasserted while the processor performs an external bus self-confidence test. If this self-test passes, the processor deasserts the $\overline{\text{FAIL}}$ pin and branches to the user's initialization routine; otherwise the $\overline{\text{FAIL}}$ pin remains asserted. Internal self-test and the use of the $\overline{\text{FAIL}}$ pin can be disabled with the STEST pin.
STEST	I S(L) H(Z) R(Z)	**SELF TEST** causes the processor's internal self-test feature to be enabled or disabled at initialization. STEST is read on the rising edge of $\overline{\text{RESET}}$. When asserted, the processor's internal self-test and external bus confidence tests are performed during processor initialization. When deasserted, only the bus confidence tests are performed during initialization.
$\overline{\text{ONCE}}$	I A(L) H(Z) R(Z)	**ON CIRCUIT EMULATION**, when asserted, causes all outputs to be floated. $\overline{\text{ONCE}}$ is continuously sampled while $\overline{\text{RESET}}$ is low and is latched on the rising edge of $\overline{\text{RESET}}$. To place the processor in the ONCE state: (1) assert $\overline{\text{RESET}}$ and $\overline{\text{ONCE}}$ (order does not matter) (2) wait for at least 16 CLKIN periods in 2-x mode—or 10,000 CLKIN periods in 1-x mode—after V_{CC} and CLKIN are within operating specifications (3) deassert $\overline{\text{RESET}}$ (4) wait at least 32 CLKIN periods (The processor will now be latched in the ONCE state as long as $\overline{\text{RESET}}$ is high.) To exit the ONCE state, bring V_{CC} and CLKIN to operating conditions, then assert $\overline{\text{RESET}}$ and bring $\overline{\text{ONCE}}$ high prior to deasserting $\overline{\text{RESET}}$. CLKIN must operate within the specified operating conditions of the processor until Step 4 above has been completed. CLKIN may then be changed to DC to achieve the lowest possible ONCE mode leakage current. $\overline{\text{ONCE}}$ can be used by emulator products or for board testers to effectively make an installed processor transparent in the board.

Table 4. 80960CA Pin Description — Processor Control Signals (Continued)

Name	Type	Description
CLKIN	I A(E) H(Z) R(Z)	**CLOCK INPUT** is an input for the external clock needed to run the processor. The external clock is internally divided as prescribed by the CLKMODE pin to produce PCLK2:1.
CLKMODE	I A(L) H(Z) R(Z)	**CLOCK MODE** selects the division factor applied to the external clock input (CLKIN). When CLKMODE is high, CLKIN is divided by one to create PCLK2:1 and the processor's internal clock. When CLKMODE is low, CLKIN is divided by two to create PCLK2:1 and the processor's internal clock. CLKMODE should be tied high or low in a system as the clock mode is not latched by the processor. If left unconnected, the processor will internally pull the CLKMODE pin low, enabling the 2-x clock mode.
PCLK2:1	O S H(Q) R(Q)	**PROCESSOR OUTPUT CLOCKS** provide a timing reference for all processor inputs and outputs. All input and output timings are specified in relation to PCLK2 and PCLK1. PCLK2 and PCLK1 are identical signals. Two output pins are provided to allow flexibility in the system's allocation of capacitive loading on the clock. PCLK2:1 may also be connected at the processor to form a single clock signal.
V_{SS}	–	**GROUND** connections must be connected externally to a V_{SS} board plane.
V_{CC}	–	**POWER** connections must be connected externally to a V_{CC} board plane.
V_{CCPLL}	–	**V_{CCPLL}** is a separate V_{CC} supply pin for the phase lock loop used in 1-x clock mode. Connecting a simple lowpass filter to V_{CCPLL} may help reduce clock jitter (T_{CP}) in noisy environments. Otherwise, V_{CCPLL} should be connected to V_{CC}. This pin is implemented starting with the D-stepping. See Table 13 for die stepping information.
NC	–	**NO CONNECT** pins must not be connected in a system.

Table 5. 80960CA Pin Description — DMA and Interrupt Unit Control Signals

Name	Type	Description
$\overline{\text{DREQ3:0}}$	I A(L) H(Z) R(Z)	**DMA REQUEST** causes a DMA transfer to be requested. Each of the four signals requests a transfer on a single channel. DREQ0 requests channel 0, DREQ1 requests channel 1, etc. When two or more channels are requested simultaneously, the channel with the highest priority is serviced first. The channel priority mode is programmable.
$\overline{\text{DACK3:0}}$	O S H(1) R(1)	**DMA ACKNOWLEDGE** indicates that a DMA transfer is being executed. Each of the four signals acknowledges a transfer for a single channel. $\overline{\text{DACK0}}$ acknowledges channel 0, $\overline{\text{DACK1}}$ acknowledges channel 1, etc. $\overline{\text{DACK3:0}}$ are asserted when the requesting device of a DMA is accessed.
$\overline{\text{EOP}}$/$\overline{\text{TC3:0}}$	I/O A(L) H(Z/Q) R(Z)	**END OF PROCESS/TERMINAL COUNT** can be programmed as either an input ($\overline{\text{EOP3:0}}$) or as an output ($\overline{\text{TC3:0}}$), but not both. Each pin is individually programmable. When programmed as an input, $\overline{\text{EOPx}}$ causes the termination of a current DMA transfer for the channel corresponding to the $\overline{\text{EOPx}}$ pin. $\overline{\text{EOP0}}$ corresponds to channel 0, $\overline{\text{EOP1}}$ corresponds to channel 1, etc. When a channel is configured for source *and* destination chaining, the EOP pin for that channel causes termination of only the current buffer transferred and causes the next buffer to be transferred. $\overline{\text{EOP3:0}}$ are asynchronous inputs. When programmed as an output, the channel's $\overline{\text{TCx}}$ pin indicates that the channel byte count has reached 0 and a DMA has terminated. $\overline{\text{TCx}}$ is driven with the same timing as $\overline{\text{DACKx}}$ during the last DMA transfer for a buffer. If the last bus request is executed as multiple bus accesses, $\overline{\text{TCx}}$ will stay asserted for the entire bus request.
$\overline{\text{XINT7:0}}$	I A(E/L) H(Z) R(Z)	**EXTERNAL INTERRUPT PINS** cause interrupts to be requested. These pins can be configured in three modes: **Dedicated Mode:** each pin is a dedicated external interrupt source. Dedicated inputs can be individually programmed to be level (low) or edge (falling) activated. **Expanded Mode:** the eight pins act together as an 8-bit vectored interrupt source. The interrupt pins in this mode are level activated.Since the interrupt pins are active low, the vector number requested is the one's complement of the positive logic value place on the port. This eliminates glue logic to interface to combinational priority encoders which output negative logic. **Mixed Mode:** $\overline{\text{XINT7:5}}$ are dedicated sources and $\overline{\text{XINT4:0}}$ act as the five most significant bits of an expanded mode vector. The least significant bits are set to 010 internally.
$\overline{\text{NMI}}$	I A(E) H(Z) R(Z)	**NON-MASKABLE INTERRUPT** causes a non-maskable interrupt event to occur. $\overline{\text{NMI}}$ is the highest priority interrupt recognized. $\overline{\text{NMI}}$ is an edge (falling) activated source.

3.3 80960CA Mechanical Data

3.3.1 80960CA PGA Pinout

Tables 6 and 7 list the 80960CA pin names with package location. Figure 2 depicts the complete 80960CA PGA pinout as viewed from the top side of the component (i.e., pins facing down). Figure 3 shows the complete 80960CA PGA pinout as viewed from the pin-side of the package (i.e., pins facing up). See **Section 4.0, ELECTRICAL SPECIFICATIONS** for specifications and recommended connections.

Table 6. 80960CA PGA Pinout — In Signal Order

Address Bus		Data Bus		Bus Control		Processor Control		I/O	
Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
A31	S15	D31	R3	$\overline{BE3}$	S5	$\overline{RESET}$	A16	$\overline{DREQ3}$	A7
A30	Q13	D30	Q5	$\overline{BE2}$	S6			$\overline{DREQ2}$	B6
A29	R14	D29	S2	$\overline{BE1}$	S7	$\overline{FAIL}$	A2	$\overline{DREQ1}$	A6
A28	Q14	D28	Q4	$\overline{BE0}$	R9			$\overline{DREQ0}$	B5
A27	S16	D27	R2			STEST	B2		
A26	R15	D26	Q3	W/$\overline{R}$	S10			$\overline{DACK3}$	A10
A25	S17	D25	S1			$\overline{ONCE}$	C3	$\overline{DACK2}$	A9
A24	Q15	D24	R1	$\overline{ADS}$	R6			$\overline{DACK1}$	A8
A23	R16	D23	Q2			CLKIN	C13	$\overline{DACK0}$	B8
A22	R17	D22	P3	$\overline{READY}$	S3	CLKMODE	C14		
A21	Q16	D21	Q1	$\overline{BTERM}$	R4	PLCK1	B14	$\overline{EOP}$/$\overline{TC3}$	A14
A20	P15	D20	P2			PLCK2	B13	$\overline{EOP}$/$\overline{TC2}$	A13
A19	P16	D19	P1	$\overline{WAIT}$	S12			$\overline{EOP}$/$\overline{TC1}$	A12
A18	Q17	D18	N2	$\overline{BLAST}$	S8	V_{SS}		$\overline{EOP}$/$\overline{TC0}$	A11
A17	P17	D17	N1			*Location*			
A16	N16	D16	M1	DT/$\overline{R}$	S11	C7, C8, C9, C10, C11, C12, F15, G3, G15, H3, H15, J3, J15, K3, K15, L3, L15, M3, M15, Q7, Q8, Q9, Q10, Q11		$\overline{XINT7}$	C17
A15	N17	D15	L1	$\overline{DEN}$	S9			$\overline{XINT6}$	C16
A14	M17	D14	L2					$\overline{XINT5}$	B17
A13	L16	D13	K1	$\overline{LOCK}$	S14			$\overline{XINT4}$	C15
A12	L17	D12	J1					$\overline{XINT3}$	B16
A11	K17	D11	H1			V_{CC}		$\overline{XINT2}$	A17
A10	J17	D10	H2	HOLD	R5	*Location*		$\overline{XINT1}$	A15
A9	H17	D9	G1	HOLDA	S4	B7, B9, B11, B12, C6, E15, F3, F16, G2, H16, J2, J16, K2, K16, M2, M16, N3, N15, Q6, R7, R8, R10, R11		$\overline{XINT0}$	B15
A8	G17	D8	F1	$\overline{BREQ}$	R13				
A7	G16	D7	E1					$\overline{NMI}$	D15
A6	F17	D6	F2	D/$\overline{C}$	S13				
A5	E17	D5	D1	$\overline{DMA}$	R12				
A4	E16	D4	E2	$\overline{SUP}$	Q12	V_{CCPLL}	B10		
A3	D17	D3	C1			**No Connect**			
A2	D16	D2	D2	$\overline{BOFF}$	B1	*Location*			
		D1	C2			A1, A3, A4, A5, B3, B4, C4, C5, D3			
		D0	E3						

Table 7. 80960CA PGA Pinout — In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	NC	C1	D3	G1	D9	M1	D16	R1	D24
A2	$\overline{FAIL}$	C2	D1	G2	V_{CC}	M2	V_{CC}	R2	D27
A3	NC	C3	$\overline{ONCE}$	G3	V_{SS}	M3	V_{SS}	R3	D31
A4	NC	C4	NC	G15	V_{SS}	M15	V_{SS}	R4	$\overline{BTERM}$
A5	NC	C5	NC	G16	A7	M16	V_{CC}	R5	HOLD
A6	$\overline{DREQ1}$	C6	V_{CC}	G17	A8	M17	A14	R6	$\overline{ADS}$
A7	$\overline{DREQ3}$	C7	V_{SS}					R7	V_{CC}
A8	$\overline{DACK1}$	C8	V_{SS}	H1	D11	N1	D17	R8	V_{CC}
A9	$\overline{DACK2}$	C9	V_{SS}	H2	D10	N2	D18	R9	$\overline{BE0}$
A10	$\overline{DACK3}$	C10	V_{SS}	H3	V_{SS}	N3	V_{CC}	R10	V_{CC}
A11	$\overline{EOP/TC0}$	C11	V_{SS}	H15	V_{SS}	N15	V_{CC}	R11	V_{CC}
A12	$\overline{EOP/TC1}$	C12	V_{SS}	H16	V_{CC}	N16	A16	R12	$\overline{DMA}$
A13	$\overline{EOP/TC2}$	C13	CLKIN	H17	A9	N17	A15	R13	BREQ
A14	$\overline{EOP/TC3}$	C14	CLKMODE					R14	A29
A15	$\overline{XINT1}$	C15	$\overline{XINT4}$	J1	D12	P1	D19	R15	A26
A16	$\overline{RESET}$	C16	$\overline{XINT6}$	J2	V_{CC}	P2	D20	R16	A23
A17	$\overline{XINT2}$	C17	$\overline{XINT7}$	J3	V_{SS}	P3	D22	R17	A22
				J15	V_{SS}	P15	A20		
B1	$\overline{BOFF}$	D1	D5	J16	V_{CC}	P16	A19	S1	D25
B2	STEST	D2	D2	J17	A10	P17	A17	S2	D29
B3	NC	D3	NC					S3	$\overline{READY}$
B4	NC	D15	$\overline{NMI}$	K1	D13	Q1	D21	S4	HOLDA
B5	$\overline{DREQ0}$	D16	A2	K2	V_{CC}	Q2	D23	S5	$\overline{BE3}$
B6	$\overline{DREQ2}$	D17	A3	K3	V_{SS}	Q3	D26	S6	$\overline{BE2}$
B7	V_{CC}			K15	V_{SS}	Q4	D28	S7	$\overline{BE1}$
B8	$\overline{DACK0}$	E1	D7	K16	V_{CC}	Q5	D30	S8	$\overline{BLAST}$
B9	V_{CC}	E2	D4	K17	A11	Q6	V_{CC}	S9	$\overline{DEN}$
B10	V_{CCPLL}	E3	D0			Q7	V_{SS}	S10	W/$\overline{R}$
B11	V_{CC}	E15	V_{CC}	L1	D15	Q8	V_{SS}	S11	DT/$\overline{R}$
B12	V_{CC}	E16	A4	L2	D14	Q9	V_{SS}	S12	$\overline{WAIT}$
B13	PCLK2	E17	A5	L3	V_{SS}	Q10	V_{SS}	S13	D/$\overline{C}$
B14	PCLK1			L15	V_{SS}	Q11	V_{SS}	S14	$\overline{LOCK}$
B15	$\overline{XINT0}$	F1	D8	L16	A13	Q12	$\overline{SUP}$	S15	A31
B16	$\overline{XINT3}$	F2	D6	L17	A12	Q13	A30	S16	A27
B17	$\overline{XINT5}$	F3	V_{CC}			Q14	A28	S17	A25
		F15	V_{SS}			Q15	A24		
		F16	V_{CC}			Q16	A21		
		F17	A6			Q17	A18		

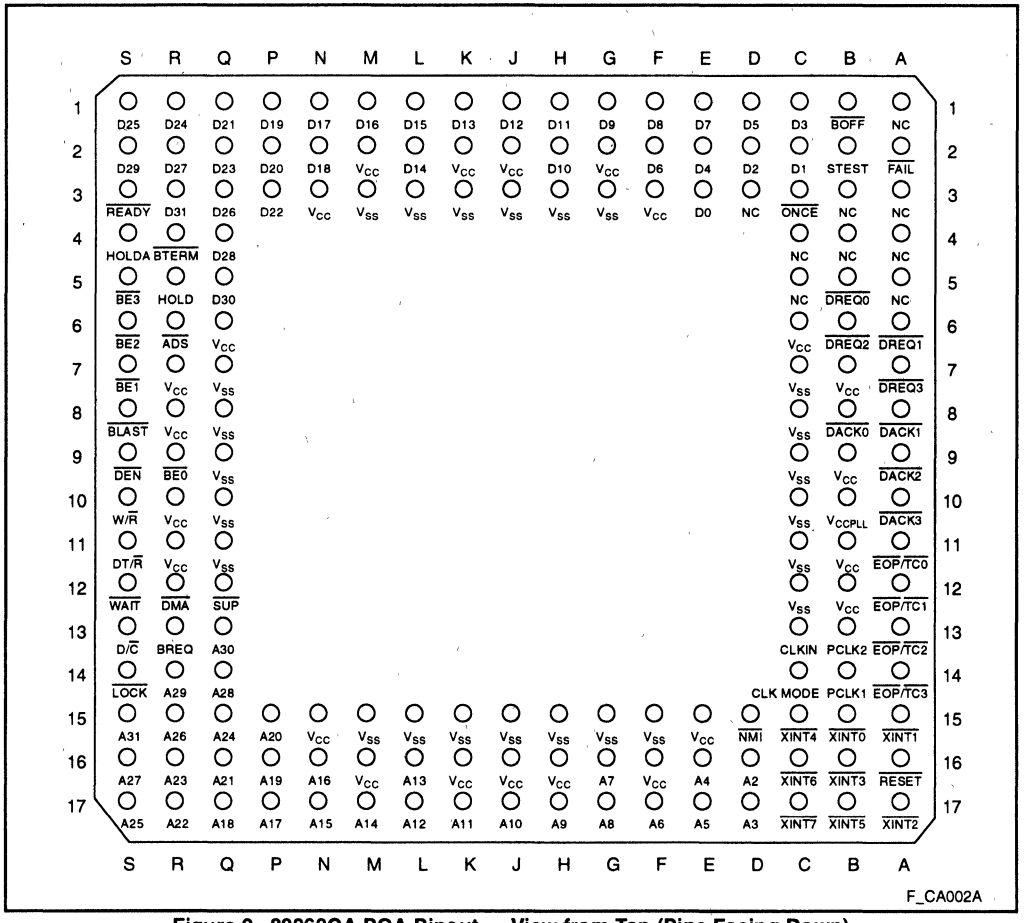


Figure 2. 80960CA PGA Pinout — View from Top (Pins Facing Down)

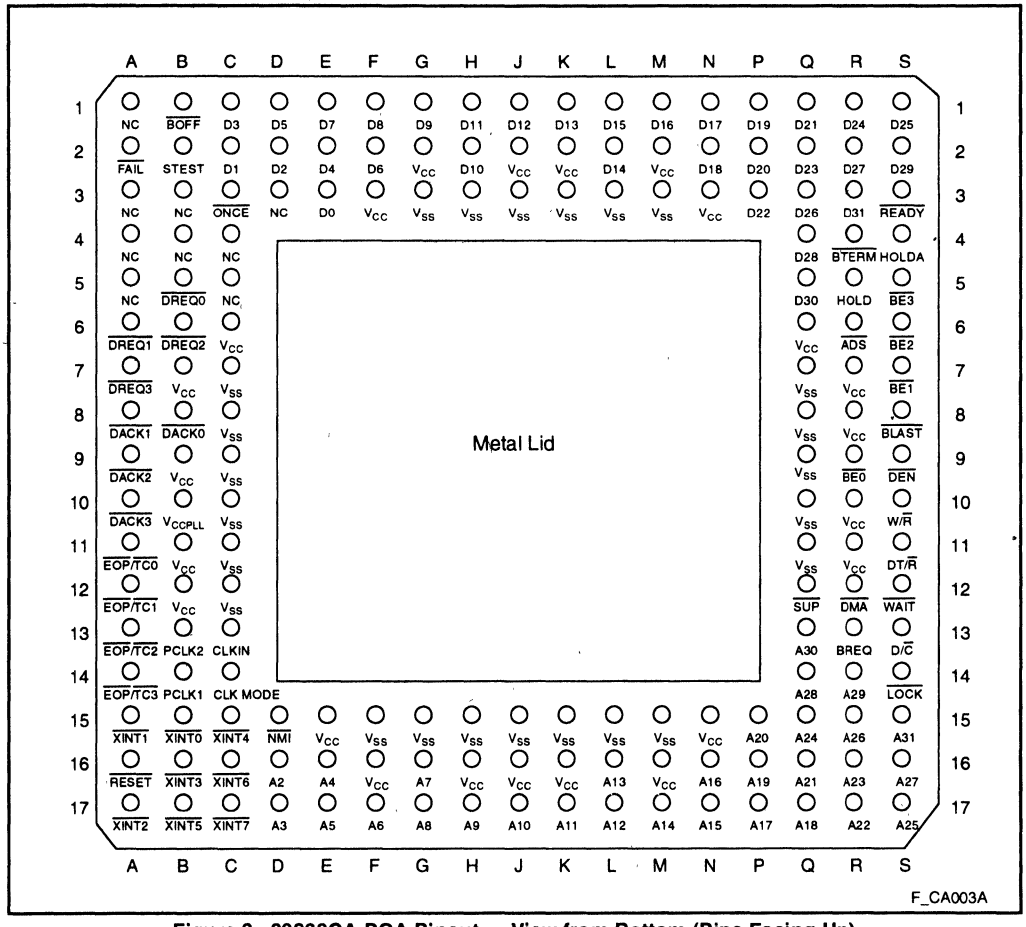


Figure 3. 80960CA PGA Pinout — View from Bottom (Pins Facing Up)

3.3.2 80960CA PQFP Pinout

Tables 8 and 9 list the 80960CA pin names with package location. Figure 4 shows the 80960CA PQFP pinout as viewed from the top side.

See **Section 4.0, ELECTRICAL SPECIFICATIONS** for specifications and recommended connections.

Table 8. 80960CA PQFP Pinout — In Signal Order

Address Bus		Data Bus		Bus Control		Processor Control		I/O	
Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
A31	153	D31	186	$\overline{BE3}$	176	$\overline{RESET}$	91	$\overline{DREQ3}$	60
A30	152	D30	187	$\overline{BE2}$	175	$\overline{FAIL}$	45	$\overline{DREQ2}$	59
A29	151	D29	188	$\overline{BE1}$	172	STEST	46	$\overline{DREQ1}$	58
A28	145	D28	189	$\overline{BE0}$	170	$\overline{ONCE}$	43	$\overline{DREQ0}$	57
A27	144	D27	191			CLKIN	87		
A26	143	D26	192	W/$\overline{R}$	164	CLKMODE	85	$\overline{DACK3}$	65
A25	142	D25	194			PCLK2	74	$\overline{DACK2}$	64
A24	141	D24	195	$\overline{ADS}$	178	PCLK1	78	$\overline{DACK1}$	63
A23	139	D23	3			**V_{SS}**		$\overline{DACK0}$	62
A22	138	D22	4	$\overline{READY}$	182	*Location*			
A21	137	D21	5	$\overline{BTERM}$	184	2, 7, 16, 24, 30, 38, 39, 49, 56, 70, 75, 77, 81, 83, 88, 89, 92, 98, 105, 109, 110, 121, 125, 131, 135, 147, 150, 161, 165, 173, 174, 185, 196		$\overline{EOP/TC3}$	69
A20	136	D20	6					$\overline{EOP/TC2}$	68
A19	134	D19	8	$\overline{WAIT}$	162			$\overline{EOP/TC1}$	67
A18	133	D18	9	$\overline{BLAST}$	169			$\overline{EOP/TC0}$	66
A17	132	D17	10						
A16	130	D16	11	DT/$\overline{R}$	163			$\overline{XINT7}$	107
A15	129	D15	13	$\overline{DEN}$	167	**V_{CC}**		$\overline{XINT6}$	106
A14	128	D14	14			*Location*		$\overline{XINT5}$	102
A13	124	D13	15	$\overline{LOCK}$	156	1, 12, 20, 28, 32, 37, 44, 50, 61, 71, 79, 82, 96, 99, 103, 115, 127, 140, 148, 154, 168, 171, 180, 190		$\overline{XINT4}$	101
A12	123	D12	17					$\overline{XINT3}$	100
A11	122	D11	18	HOLD	181			$\overline{XINT2}$	95
A10	120	D10	19	HOLDA	179			$\overline{XINT1}$	94
A9	119	D9	21	BREQ	155			$\overline{XINT0}$	93
A8	118	D8	22			V_{CCPLL}	72		
A7	117	D7	23	D/$\overline{C}$	159	**No Connect**		$\overline{NMI}$	108
A6	116	D6	25	$\overline{DMA}$	160	*Location*			
A5	114	D5	26	$\overline{SUP}$	158	29, 31, 41, 42, 47, 48, 51, 52, 53, 54, 55, 73, 76, 80, 84, 86, 90, 97, 104, 126, 146, 149, 157, 166, 177, 183, 193			
A4	113	D4	27						
A3	112	D3	33	$\overline{BOFF}$	40				
A2	111	D2	34						
		D1	35						
		D0	36						

Table 9. 80960CA PQFP Pinout — In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	V_{CC}	34	D2	67	$\overline{EOP}/\overline{TC1}$	100	$\overline{XINT3}$	133	A18	166	NC
2	V_{SS}	35	D1	68	$\overline{EOP}/\overline{TC2}$	101	$\overline{XINT4}$	134	A19	167	$\overline{DEN}$
3	D23	36	D0	69	$\overline{EOP}/\overline{TC3}$	102	$\overline{XINT5}$	135	V_{SS}	168	V_{CC}
4	D22	37	V_{CC}	70	V_{SS}	103	V_{CC}	136	A20	169	$\overline{BLAST}$
5	D21	38	V_{SS}	71	V_{CC}	104	NC	137	A21	170	$\overline{BE0}$
6	D20	39	V_{SS}	72	V_{CCPLL}	105	V_{SS}	138	A22	171	V_{CC}
7	V_{SS}	40	$\overline{BOFF}$	73	NC	106	$\overline{XINT6}$	139	A23	172	$\overline{BE1}$
8	D19	41	NC	74	PCLK2	107	$\overline{XINT7}$	140	V_{CC}	173	V_{SS}
9	D18	42	NC	75	V_{SS}	108	$\overline{NMI}$	141	A24	174	V_{SS}
10	D17	43	$\overline{ONCE}$	76	NC	109	V_{SS}	142	A25	175	$\overline{BE2}$
11	D16	44	V_{CC}	77	V_{SS}	110	V_{SS}	143	A26	176	$\overline{BE3}$
12	V_{CC}	45	$\overline{FAIL}$	78	PCLK1	111	A2	144	A27	177	NC
13	D15	46	STEST	79	V_{CC}	112	A3	145	A28	178	$\overline{ADS}$
14	D14	47	NC	80	NC	113	A4	146	NC	179	HOLDA
15	D13	48	NC	81	V_{SS}	114	A5	147	V_{SS}	180	V_{CC}
16	V_{SS}	49	V_{SS}	82	V_{CC}	115	V_{CC}	148	V_{CC}	181	HOLD
17	D12	50	V_{CC}	83	V_{SS}	116	A6	149	NC	182	$\overline{READY}$
18	D11	51	NC	84	NC	117	A7	150	V_{SS}	183	NC
19	D10	52	NC	85	CLKMODE	118	A8	151	A29	184	$\overline{BTERM}$
20	V_{CC}	53	NC	86	NC	119	A9	152	A30	185	V_{SS}
21	D9	54	NC	87	CLKIN	120	A10	153	A31	186	D31
22	D8	55	NC	88	V_{SS}	121	V_{SS}	154	V_{CC}	187	D30
23	D7	56	V_{SS}	89	V_{SS}	122	A11	155	BREQ	188	D29
24	V_{SS}	57	$\overline{DREQ0}$	90	NC	123	A12	156	$\overline{LOCK}$	189	D28
25	D6	58	$\overline{DREQ1}$	91	$\overline{RESET}$	124	A13	157	NC	190	V_{CC}
26	D5	59	$\overline{DREQ2}$	92	V_{SS}	125	V_{SS}	158	$\overline{SUP}$	191	D27
27	D4	60	$\overline{DREQ3}$	93	$\overline{XINT0}$	126	NC	159	D/$\overline{C}$	192	D26
28	V_{CC}	61	V_{CC}	94	$\overline{XINT1}$	127	V_{CC}	160	$\overline{DMA}$	193	NC
29	NC	62	$\overline{DACK0}$	95	$\overline{XINT2}$	128	A14	161	V_{SS}	194	D25
30	V_{SS}	63	$\overline{DACK1}$	96	V_{CC}	129	A15	162	$\overline{WAIT}$	195	D24
31	NC	64	$\overline{DACK2}$	97	NC	130	A16	163	DT/$\overline{R}$	196	V_{SS}
32	V_{CC}	65	$\overline{DACK3}$	98	V_{SS}	131	V_{SS}	164	W/$\overline{R}$		
33	D3	66	$\overline{EOP}/\overline{TC0}$	99	V_{CC}	132	A17	165	V_{SS}		

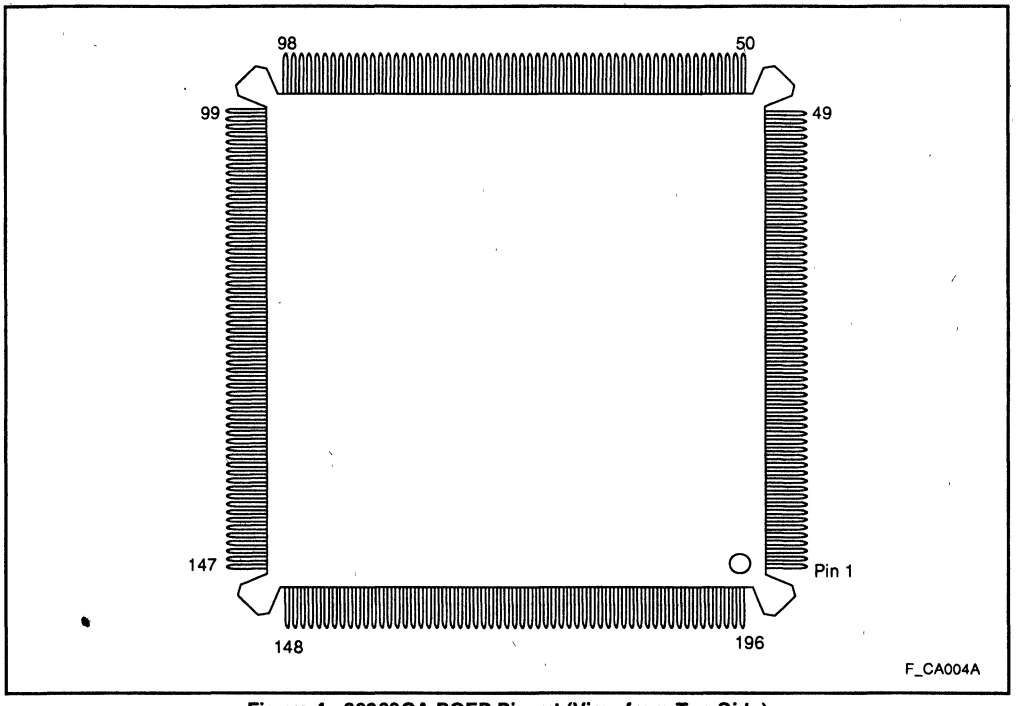


Figure 4. 80960CA PQFP Pinout (View from Top Side)

3.4 Package Thermal Specifications

The 80960CA is specified for operation when T_C (case temperature) is within the range of 0°C–100°C. T_C may be measured in any environment to determine whether the 80960CA is within specified operating range. Case temperature should be measured at the center of the top surface, opposite the pins. Refer to Figure 5.

T_A (ambient temperature) can be calculated from θ_{CA} (thermal resistance from case to ambient) using the following equation:

$$T_A = T_C - P^*\theta_{CA}$$

Table 10 shows the maximum T_A allowable (without exceeding T_C) at various airflows and operating frequencies (f_{PCLK}).

Note that T_A is greatly improved by attaching fins or a heatsink to the package. P (maximum power consumption) is calculated by using the typical I_{CC} as tabulated in **Section 4.4, DC Specifications** and V_{CC} of 5V.

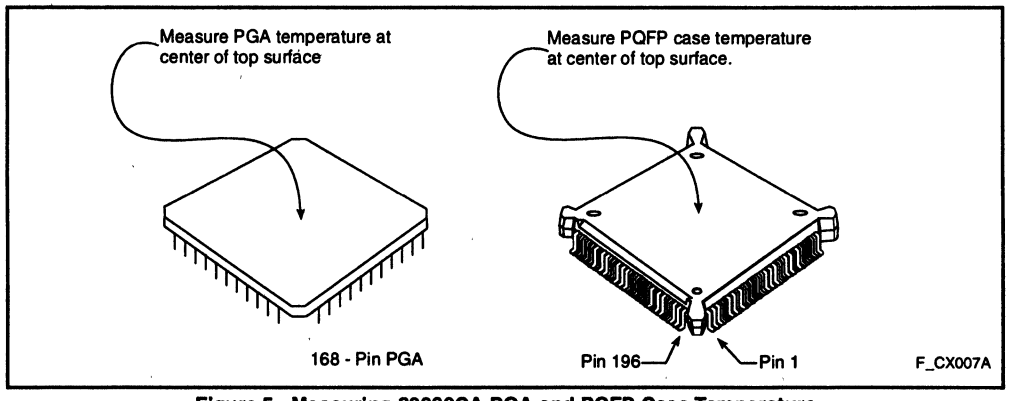


Figure 5. Measuring 80960CA PGA and PQFP Case Temperature

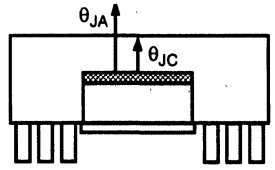
Table 10. Maximum T_A at Various Airflows in °C (PGA Package Only)

		Airflow-ft/min (m/sec)					
	f_{PCLK} (MHz)	0 (0)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)	1000 (5.07)
T_A with Heatsink*	33	51	66	79	81	85	87
	25	61	73	83	85	88	89
	16	74	82	89	90	92	93
T_A without Heatsink*	33	36	47	59	66	73	75
	25	49	58	67	73	78	80
	16	66	72	78	82	86	87

NOTES:
*0.285" high unidirectional heatsink (Al alloy 6061, 50 mil fin width, 150 mil center-to-center fin spacing).

Table 11. 80960CA PGA Package Thermal Characteristics

Parameter	Thermal Resistance — °C/Watt Airflow — ft./min (m/sec)					
	0 (0)	200 (1.01)	400 (2.03)	600 (3.07)	800 (4.06)	1000 (5.07)
θ Junction-to-Case (Case measured as shown in Figure 5)	1.5	1.5	1.5	1.5	1.5	1.5
θ Case-to-Ambient (No Heatsink)	17	14	11	9	7.1	6.6
θ Case-to-Ambient (With Heatsink)*	13	9	5.5	5	3.9	3.4



NOTES:

1. This table applies to 80960CA PGA plugged into socket or soldered directly to board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$

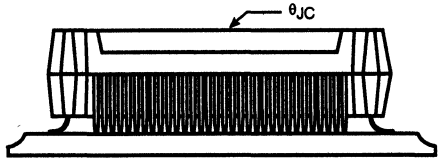
*0.285" high unidirectional heatsink (Al alloy 6061, 50 mil fin width, 150 mil center-to-center fin spacing).

Table 12. 80960CA PQFP Package Thermal Characteristics

Parameter	Thermal Resistance — °C/Watt Airflow — ft./min (m/sec)						
	0 (0)	50 (0.25)	100 (0.50)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)
θ Junction-to-Case (Case Measured as shown in Figure 5)	5	5	5	5	5	5	5
θ Case-to-Ambient (No Heatsink)	19	18	17	15	12	10	9

NOTES:

1. This table applies to 80960CA PQFP soldered directly to board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$



3.5 Stepping Register Information

Upon reset, register g0 contains die stepping information. Figure 6 shows how g0 is configured. The most significant byte contains an ASCII 0. The upper middle byte contains an ASCII C. The lower middle byte contains an ASCII A. The least significant byte contains the stepping number in ASCII. g0 retains this information until it is overwritten by the user program.

ASCII	00	43	41	Stepping Number
DECIMAL	0	C	A	Stepping Number
	MSB			LSB

Figure 6. Register g0

Table 13 contains a cross reference of the number in the least significant byte of register g0 to the die stepping number.

Table 13. Die Stepping Cross Reference

g0 Least Significant Byte	Die Stepping
01	B
02	C-1
03	C-2,C-3
04	D

3.6 Suggested Sources for 80960CA Accessories

The following is a list of suggested sources for 80960CA accessories. This is not an endorsement of any kind, nor is it a warranty of the performance of any of the listed products and/or companies.

Sockets

1. 3M Textool Test and Interconnection
   Products Department
   P.O. Box 2963
   Austin, TX 78769-2963
2. Augat, Inc.
   Interconnection Products Group
   33 Perry Avenue
   P.O. box 779
   Attleboro, MA 02703
   (508) 699-7646
3. Concept Manufacturing, Inc.
   (Decoupling Sockets)
   41484 Christy Street
   Fremont, CA 94538
   (415) 651-3804

Heatsinks/Fins

1. Thermalloy, Inc.
   2021 West Valley View Lane
   Dallas, TX 75234-8993
   (214) 243-4321
   FAX: (214) 241-4656
2. E G & G Division
   60 Audubon Road
   Wakefield, MA 01880
   (617) 245-5900

4.0 ELECTRICAL SPECIFICATIONS

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

4.1 Absolute Maximum Ratings

Parameter	Maximum Rating
Storage Temperature	–65°C to +150°C
Case Temperature Under Bias	–65°C to +110°C
Supply Voltage wrt. V_{SS}	–0.5V to + 6.5V
Voltage on Other Pins wrt. V_{SS}	–0.5V to V_{CC} + 0.5V

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

4.2 Operating Conditions

Table 14. Operating Conditions (80960CA-33, -25, -16)

Symbol	Parameter		Min	Max	Units	Notes
V_{CC}	Supply Voltage	80960CA-33	4.75	5.25	V	
		80960CA-25	4.50	5.50	V	
		80960CA-16	4.50	5.50	V	
f_{CLK2x}	Input Clock Frequency (2-x Mode)	80960CA-33	0	66.66	MHz	
		80960CA-25	0	50	MHz	
		80960CA-16	0	32	MHz	
f_{CLK1x}	Input Clock Frequency (1-x Mode)	80960CA-33	8	33.33	MHz	
		80960CA-25	8	25	MHz	(1)
		80960CA-16	8	16	MHz	
T_C	Case Temperature Under Bias	PGA Package	0	100	°C	
	80960CA-33, -25, -16	196-Pin PQFP	0	100	°C	

NOTES:

1. When in the 1-x input clock mode, CLKIN is an input to an internal phase-locked loop and must maintain a minimum frequency of 8 MHz for proper processor operation. However, in the 1-x mode, CLKIN may still be stopped when the processor either is in a reset condition or is reset. If CLKIN is stopped, the specified RESET low time must be provided once CLKIN restarts and has stabilized.

4.3 Recommended Connections

Power and ground connections must be made to multiple V_{CC} and V_{SS} (GND) pins. Every 80960CA-based circuit board should include power (V_{CC}) and ground (V_{SS}) planes for power distribution. Every V_{CC} pin must be connected to the power plane, and every V_{SS} pin must be connected to the ground plane. Pins identified as "NC" **must not** be connected in the system.

Liberal decoupling capacitance should be placed near the 80960CA. The processor can cause transient power surges when its numerous output buffers transition, particularly when connected to large capacitive loads.

Low inductance capacitors and interconnects are recommended for best high frequency electrical performance. Inductance can be reduced by shortening the board traces between the processor and decoupling capacitors as much as possible. Capacitors specifically designed for PGA packages will offer the lowest possible inductance.

For reliable operation, always connect unused inputs to an appropriate signal level. In particular, any unused interrupt (XINT, NMI) or DMA (DREQ) input should be connected to V_{CC} through a pull-up resistor, as should BTERM if not used. Pull-up resistors should be in the in the range of 20 KΩ for each pin tied high. If READY or HOLD are not used, the unused input should be connected to ground. **N.C. pins must always remain unconnected.** Refer to the *i960® CA Microprocessor User's Manual* (Order Number 270710) for more information.

4.4 DC Specifications

Table 15. DC Characteristics

(80960CA-33, -25, -16 under the conditions described in **Section 4.2, Operating Conditions.**)

Symbol	Parameter	Min	Max	Units	Notes
V_{IL}	Input Low Voltage for all pins except $\overline{RESET}$	−0.3	+0.8	V	
V_{IH}	Input High Voltage for all pins except $\overline{RESET}$	2.0	V_{CC} + 0.3	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 5 mA
V_{OH}	Output High Voltage I_{OH} = −1 mA I_{OH} = −200 µA	2.4 V_{CC} − 0.5		V V	
V_{ILR}	Input Low Voltage for $\overline{RESET}$	−0.3	1.5	V	
V_{IHR}	Input High Voltage for $\overline{RESET}$	3.5	V_{CC} + 0.3	V	
I_{LI1}	Input Leakage Current for each pin *except*: $\overline{BTERM}$, $\overline{ONCE}$, $\overline{DREQ3:0}$, $\overline{STEST}$, $\overline{EOP3:0}$/$\overline{TC3:0}$, NMI, $\overline{XINT7:0}$, $\overline{BOFF}$, $\overline{READY}$, HOLD, CLKMODE		±15	µA	$0 \le V_{IN} \le V_{CC}$ (1)
I_{LI2}	Input Leakage Current for: $\overline{BTERM}$, $\overline{ONCE}$, $\overline{DREQ3:0}$, $\overline{STEST}$, $\overline{EOP3:0}$/$\overline{TC3:0}$, NMI, $\overline{XINT7:0}$, $\overline{BOFF}$	0	−300	µA	V_{IN} = 0.45V (2)
I_{LI3}	Input Leakage Current for: $\overline{READY}$, HOLD, CLKMODE	0	500	µA	V_{IN} = 2.4V (3,7)
I_{LO}	Output Leakage Current		±15	µA	$0.45 \le V_{OUT} \le V_{CC}$
I_{CC}	Supply Current (80960CA-33): I_{CC} Max I_{CC} Typ		 900 750	 mA mA	 (4) (5)
I_{CC}	Supply Current (80960CA-25): I_{CC} Max I_{CC} Typ		 750 600	 mA mA	 (4) (5)
I_{CC}	Supply Current (80960CA-16): I_{CC} Max I_{CC} Typ		 550 400	 mA mA	 (4) (5)
I_{ONCE}	ONCE-mode Supply Current		100	mA	
C_{IN}	Input Capacitance for: CLKIN, $\overline{RESET}$, $\overline{ONCE}$, $\overline{READY}$, HOLD, $\overline{DREQ3:0}$, $\overline{BOFF}$, $\overline{XINT7:0}$, NMI, BTERM, CLKMODE	0	12	pF	F_C = 1 MHz
C_{OUT}	Output Capacitance of each output pin		12	pF	F_C = 1 MHz (6)
$C_{I/O}$	I/O Pin Capacitance		12	pF	F_C = 1 MHz

NOTES:
1. No pullup or pulldown.
2. These pins have internal pullup resistors.
3. These pins have internal pulldown resistors.
4. Measured at worst case frequency, V_{CC} and temperature, with device operating and outputs loaded to the test conditions described in **Section 4.5.1, AC Test Conditions.**
5. I_{CC} Typical is not tested.
6. Output Capacitance is the capacitive load of a floating output.
7. CLKMODE pin has a pulldown resistor only when $\overline{ONCE}$ pin is deasserted.

4.5 AC Specifications

Table 16. 80960CA AC Characteristics (33 MHz)

(80960CA-33 only, under the conditions described in **Section 4.2, Operating Conditions** and **Section 4.5.1, AC Test Conditions.**)

Symbol	Parameter		Min	Max	Units	Notes
Input Clock (1,9)						
T_F	CLKIN Frequency		0	66.66	MHz	
T_C	CLKIN Period	In 1-x Mode (f_{CLK1x})	30	125	ns	(11)
		In 2-x Mode (f_{CLK2x})	15	∞	ns	
T_{CS}	CLKIN Period Stability	In 1-x Mode (f_{CLK1x})		±0.1%	Δ	(12)
T_{CH}	CLKIN High Time	In 1-x Mode (f_{CLK1x})	6	62.5	ns	(11)
		In 2-x Mode (f_{CLK2x})	6	∞	ns	
T_{CL}	CLKIN Low Time	In 1-x Mode (f_{CLK1x})	6	62.5	ns	(11)
		In 2-x Mode (f_{CLK2x})	6	∞	ns	
T_{CR}	CLKIN Rise Time		0	6	ns	
T_{CF}	CLKIN Fall Time		0	6	ns	
Output Clocks (1,8)						
T_{CP}	CLKIN to PCLK2:1 Delay	In 1-x Mode (f_{CLK1x})	−2	2	ns	(3,12)
		In 2-x Mode (f_{CLK2x})	2	25	ns	(3)
T	PCLK2:1 Period	In 1-x Mode (f_{CLK1x})	T_C		ns	(12)
		In 2-x Mode (f_{CLK2x})	$2T_C$		ns	(3)
T_{PH}	PCLK2:1 High Time		(T/2) − 2	T/2	ns	(12)
T_{PL}	PCLK2:1 Low Time		(T/2) − 2	T/2	ns	(12)
T_{PR}	PCLK2:1 Rise Time		1	4	ns	(3)
T_{PF}	PCLK2:1 Fall Time		1	4	ns	(3)
Synchronous Outputs (8)						
T_{OH} T_{OV}	Output Valid Delay, Output Hold					(6,10)
	T_{OH1}, T_{OV1}	A31:2	3	14	ns	
	T_{OH2}, T_{OV2}	BE3:0	3	16	ns	
	T_{OH3}, T_{OV3}	ADS	6	18	ns	
	T_{OH4}, T_{OV4}	W/R	3	18	ns	
	T_{OH5}, T_{OV5}	D/C, SUP, DMA	4	16	ns	
	T_{OH6}, T_{OV6}	BLAST, WAIT	5	16	ns	
	T_{OH7}, T_{OV7}	DEN	3	16	ns	
	T_{OH8}, T_{OV8}	HOLDA, BREQ	4	16	ns	
	T_{OH9}, T_{OV9}	LOCK	4	16	ns	
	T_{OH10}, T_{OV10}	DACK3:0	4	18	ns	
	T_{OH11}, T_{OV11}	D31:0	3	16	ns	
	T_{OH12}, T_{OV12}	DT/R	T/2 + 3	T/2 + 14	ns	
	T_{OH13}, T_{OV13}	FAIL	2	14	ns	
	T_{OH14}, T_{OV14}	EOP3:0/TC3:0	3	18	ns	(6,10)
T_{OF}	Output Float for all outputs		3	22	ns	(6)
Synchronous Inputs (1,9,10)						
T_{IS}	Input Setup					
	T_{IS1}	D31:0	3		ns	
	T_{IS2}	BOFF	17		ns	
	T_{IS3}	BTERM/READY	7		ns	
	T_{IS4}	HOLD	7		ns	
T_{IH}	Input Hold					
	T_{IH1}	D31:0	5		ns	
	T_{IH2}	BOFF	5		ns	
	T_{IH3}	BTERM/READY	2		ns	
	T_{IH4}	HOLD	3		ns	

Table 16. 80960CA AC Characteristics (33 MHz) (Continued)

(80960CA-33 only, under the conditions described in **Section 4.2, Operating Conditions** and **Section 4.5.1, AC Test Conditions.**)

Symbol	Parameter	Min	Max	Units	Notes
Relative Output Timings (1,2,3,8)					
T_{AVSH1}	A31:2 Valid to $\overline{ADS}$ Rising	T – 4	T + 4	ns	
T_{AVSH2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, D/$\overline{C}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{ADS}$ Rising	T – 6	T + 6	ns	
T_{AVEL1}	A31:2 Valid to $\overline{DEN}$ Falling	T – 4	T + 4	ns	
T_{AVEL2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, $\overline{INST}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{DEN}$ Falling	T – 6	T + 6	ns	
T_{NLQV}	$\overline{WAIT}$ Falling to Output Data Valid	± 4		ns	
T_{DVNH}	Output Data Valid to $\overline{WAIT}$ Rising	N*T – 4	N*T + 4	ns	(4)
T_{NLNH}	$\overline{WAIT}$ Falling to $\overline{WAIT}$ Rising	N*T ± 4		ns	(4)
T_{NHQX}	Output Data Hold after $\overline{WAIT}$ Rising	(N+1)*T–8	(N+1)*T+6	ns	(5)
T_{EHTV}	DT/$\overline{R}$ Hold after $\overline{DEN}$ High	T/2 – 7	∞	ns	(6)
T_{TVEL}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Falling	T/2 – 4		ns	
Relative Input Timings (1,2,3)					
T_{IS5}	$\overline{RESET}$ Input Setup (2-x Clock Mode)	6		ns	(13)
T_{IH5}	$\overline{RESET}$ Input Hold (2-x Clock Mode)	5		ns	(13)
T_{IS6}	$\overline{DREQ3:0}$ Input Setup	12		ns	(7)
T_{IH6}	$\overline{DREQ3:0}$ Input Hold	7		ns	(7)
T_{IS7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Setup	7		ns	(15)
T_{IH7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Hold	3		ns	(15)
T_{IS8}	$\overline{RESET}$ Input Setup (1-x Clock Mode)	3		ns	(14)
T_{IH8}	$\overline{RESET}$ Input Hold (1-x Clock Mode)	T/4 + 1		ns	(14)

NOTES:

1. See **Section 4.5.2, AC Timing Waveforms** for waveforms and definitions.
2. See Figure 16 for capacitive derating information for output delays and hold times.
3. See Figure 17 for capacitive derating information for rise and fall times.
4. Where N is the number of N_{RAD}, N_{RDD}, N_{WAD} or N_{WDD} wait states that are programmed in the Bus Controller Region Table. $\overline{WAIT}$ never goes active when there are no wait states in an access.
5. N = Number of wait states inserted with $\overline{READY}$.
6. Output Data and/or DT/$\overline{R}$ may be driven indefinitely following a cycle if there is no subsequent bus activity
7. Since asynchronous inputs are synchronized internally by the 80960CA, they have no required setup or hold times to be recognized and for proper operation However, to guarantee recognition of the input at a particular edge of PCLK2 1, the setup times shown must be met Asynchronous inputs must be active for at least two consecutive PCLK2 1 rising edges to be seen by the processor
8. These specifications are guaranteed by the processor
9. These specifications must be met by the system for proper operation of the processor.
10. This timing is dependent upon the loading of PCLK2·1 Use the derating curves of **Section 4.5.3, Derating Curves** to adjust the timing for PCLK2.1 loading
11. In the 1-x input clock mode, the maximum input clock period is limited to 125 ns while the processor is operating When the processor is in reset, the input clock may stop even in 1-x mode
12. When in the 1-x input clock mode, these specifications assume a stable input clock with a period variation of less than ± 0 1% between adjacent cycles
13. In 2-x clock mode, $\overline{RESET}$ is an asynchronous input which has no required setup and hold time for proper operation However, to guarantee the device exits reset synchronized to a particular clock edge, the $\overline{RESET}$ pin must meet setup and hold times to the falling edge of the CLKIN. (See Figure 22.)
14. In 1-x clock mode, $\overline{RESET}$ is an asynchronous input which has no required setup and hold time for proper operation However, to guarantee the device exits reset synchronized to a particular clock edge, the $\overline{RESET}$ pin must meet setup and hold times to the rising edge of the CLKIN (See Figure 23)
15. The interrupt pins are synchronized internally by the 80960CA. They have no required setup or hold times for proper operation These pins are sampled by the interrupt controller every other clock and must be active for at least three consecutive PCLK2:1 rising edges when asserting them asynchronously To guarantee recognition at a particular clock edge, the setup and hold times shown must be met for two consecutive PCLK2 1 rising edges

Table 17. 80960CA AC Characteristics (25 MHz)

(80960CA-25 only, under conditions described in **Section 4.2, Operating Conditions** and **Section 4.5.1, AC Test Conditions**.)

Symbol	Parameter		Min	Max	Units	Notes
Input Clock (1,9)						
T_F	CLKIN Frequency		0	50	MHz	
T_C	CLKIN Period	In 1-x Mode (f_{CLK1x})	40	125	ns	(11)
		In 2-x Mode (f_{CLK2x})	20	∞	ns	
T_{CS}	CLKIN Period Stability	In 1-x Mode (f_{CLK1x})		±0.1%	Δ	(12)
T_{CH}	CLKIN High Time	In 1-x Mode (f_{CLK1x})	8	62.5	ns	(11)
		In 2-x Mode (f_{CLK2x})	8	∞	ns	
T_{CL}	CLKIN Low Time	In 1-x Mode (f_{CLK1x})	8	62.5	ns	(11)
		In 2-x Mode (f_{CLK2x})	8	∞	ns	
T_{CR}	CLKIN Rise Time		0	6	ns	
T_{CF}	CLKIN Fall Time		0	6	ns	
Output Clocks (1,8)						
T_{CP}	CLKIN to PCLK2:1 Delay	In 1-x Mode (f_{CLK1x})	−2	2	ns	(3,12)
		In 2-x Mode (f_{CLK2x})	2	25	ns	(3)
T	PCLK2:1 Period	In 1-x Mode (f_{CLK1x})	T_C		ns	(12)
		In 2-x Mode (f_{CLK2x})	$2T_C$		ns	(3)
T_{PH}	PCLK2:1 High Time		(T/2) − 3	T/2	ns	(12)
T_{PL}	PCLK2:1 Low Time		(T/2) − 3	T/2	ns	(12)
T_{PR}	PCLK2:1 Rise Time		1	4	ns	(3)
T_{PF}	PCLK2:1 Fall Time		1	4	ns	(3)
Synchronous Outputs (8)						
T_{OH} T_{OV}	Output Valid Delay, Output Hold					(6,10)
	T_{OH1}, T_{OV1}	A31:2	3	16	ns	
	T_{OH2}, T_{OV2}	BE3:0	3	18	ns	
	T_{OH3}, T_{OV3}	ADS	6	20	ns	
	T_{OH4}, T_{OV4}	W/R	3	20	ns	
	T_{OH5}, T_{OV5}	D/C, SUP, DMA	4	18	ns	
	T_{OH6}, T_{OV6}	BLAST, WAIT	5	18	ns	
	T_{OH7}, T_{OV7}	DEN	3	18	ns	
	T_{OH8}, T_{OV8}	HOLDA, BREQ	4	18	ns	
	T_{OH9}, T_{OV9}	LOCK	4	18	ns	
	T_{OH10}, T_{OV10}	DACK3:0	4	20	ns	
	T_{OH11}, T_{OV11}	D31:0	3	18	ns	
	T_{OH12}, T_{OV12}	DT/R	T/2 + 3	T/2 + 16	ns	
	T_{OH13}, T_{OV13}	FAIL	2	16	ns	
	T_{OH14}, T_{OV14}	EOP3:0/TC3:0	3	20	ns	(6,10)
T_{OF}	Output Float for all outputs		3	22	ns	(6)
Synchronous Inputs (1,9,10)						
T_{IS}	Input Setup					
	T_{IS1}	D31:0	5		ns	
	T_{IS2}	BOFF	19		ns	
	T_{IS3}	BTERM/READY	9		ns	
	T_{IS4}	HOLD	9		ns	
T_{IH}	Input Hold					
	T_{IH1}	D31:0	5		ns	
	T_{IH2}	BOFF	7		ns	
	T_{IH3}	BTERM/READY	2		ns	
	T_{IH4}	HOLD	5		ns	

Table 17. 80960CA AC Characteristics (25 MHz) (Continued)

(80960CA-25 only, under conditions described in **Section 4.2, Operating Conditions** and **Section 4.5.1, AC Test Conditions**.)

Symbol	Parameter	Min	Max	Units	Notes
Relative Output Timings (1,2,3,8)					
T_{AVSH1}	A31:2 Valid to $\overline{ADS}$ Rising	T – 4	T + 4	ns	
T_{AVSH2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, D/$\overline{C}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{ADS}$ Rising	T – 6	T + 6	ns	
T_{AVEL1}	A31:2 Valid to $\overline{DEN}$ Falling	T – 4	T + 4	ns	
T_{AVEL2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, $\overline{INST}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{DEN}$ Falling	T – 6	T + 6	ns	
T_{NLQV}	$\overline{WAIT}$ Falling to Output Data Valid	± 4		ns	
T_{DVNH}	Output Data Valid to $\overline{WAIT}$ Rising	N*T – 4	N*T + 4	ns	(4)
T_{NLNH}	$\overline{WAIT}$ Falling to $\overline{WAIT}$ Rising	N*T ± 4		ns	(4)
T_{NHQX}	Output Data Hold after $\overline{WAIT}$ Rising	(N+1)*T–8	(N+1)*T+6	ns	(5)
T_{EHTV}	DT/$\overline{R}$ Hold after $\overline{DEN}$ High	T/2 – 7	∞	ns	(6)
T_{TVEL}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Falling	T/2 – 4		ns	
Relative Input Timings (1,2,3)					
T_{IS5}	$\overline{RESET}$ Input Setup (2-x Clock Mode)	8		ns	(13)
T_{IH5}	$\overline{RESET}$ Input Hold (2-x Clock Mode)	7		ns	(13)
T_{IS6}	$\overline{DREQ3:0}$ Input Setup	14		ns	(7)
T_{IH6}	$\overline{DREQ3:0}$ Input Hold	9		ns	(7)
T_{IS7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Setup	9		ns	(15)
T_{IH7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Hold	5		ns	(15)
T_{IS8}	$\overline{RESET}$ Input Setup (1-x Clock Mode)	3		ns	(14)
T_{IH8}	$\overline{RESET}$ Input Hold (1-x Clock Mode)	T/4 + 1		ns	(14)

NOTES:

1 See **Section 4.5.2, AC Timing Waveforms** for waveforms and definitions.

2 See Figure 16 for capacitive derating information for output delays and hold times.

3 See Figure 17 for capacitive derating information for rise and fall times

4. Where N is the number of N_{RAD}, N_{RDD}, N_{WAD} or N_{WDD} wait states that are programmed in the Bus Controller Region Table. $\overline{WAIT}$ never goes active when there are no wait states in an access.

5 N = Number of wait states inserted with $\overline{READY}$.

6 Output Data and/or DT/$\overline{R}$ may be driven indefinitely following a cycle if there is no subsequent bus activity

7 Since asynchronous inputs are synchronized internally by the 80960CA, they have no required setup or hold times to be recognized and for proper operation However, to guarantee recognition of the input at a particular edge of PCLK2 1, the setup times shown must be met Asynchronous inputs must be active for at least two consecutive PCLK2 1 rising edges to be seen by the processor

8 These specifications are guaranteed by the processor

9 These specifications must be met by the system for proper operation of the processor.

10 This timing is dependent upon the loading of PCLK2 1. Use the derating curves of **Section 4.5.3, Derating Curves** to adjust the timing for PCLK2 1 loading

11 In the 1-x input clock mode, the maximum input clock period is limited to 125 ns while the processor is operating When the processor is in reset, the input clock may stop even in 1-x mode.

12 When in the 1-x input clock mode, these specifications assume a stable input clock with a period variation of less than ± 0 1% between adjacent cycles.

13 In 2-x clock mode, $\overline{RESET}$ is an asynchronous input which has no required setup and hold time for proper operation However, to guarantee the device exits reset synchronized to a particular clock edge, the $\overline{RESET}$ pin must meet setup and hold times to the falling edge of the CLKIN (See Figure 22)

14 In 1-x clock mode, $\overline{RESET}$ is an asynchronous input which has no required setup and hold time for proper operation. However, to guarantee the device exits reset synchronized to a particular clock edge, the $\overline{RESET}$ pin must meet setup and hold times to the rising edge of the CLKIN (See Figure 23.)

15. The interrupt pins are synchronized internally by the 80960CA. They have no required setup or hold times for proper operation. These pins are sampled by the interrupt controller every other clock and must be active for at least three consecutive PCLK2 1 rising edges when asserting them asynchronously To guarantee recognition at a particular clock edge, the setup and hold times shown must be met for two consecutive PCLK2 1 rising edges

Table 18. 80960CA AC Characteristics (16 MHz)

(80960CA-16 only, under conditions described in **Section 4.2, Operating Conditions** and **Section 4.5.1, AC Test Conditions.**)

Symbol	Parameter		Min	Max	Units	Notes
Input Clock (1,9)						
T_F	CLKIN Frequency		0	32	MHz	
T_C	CLKIN Period	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	62.5 31.25	125 ∞	ns ns	(11)
T_{CS}	CLKIN Period Stability	In 1-x Mode (f_{CLK1x})		±0.1%	Δ	(12)
T_{CH}	CLKIN High Time	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	10 10	62.5 ∞	ns ns	(11)
T_{CL}	CLKIN Low Time	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	10 10	62.5 ∞	ns ns	(11)
T_{CR}	CLKIN Rise Time		0	6	ns	
T_{CF}	CLKIN Fall Time		0	6	ns	
Output Clocks (1,8)						
T_{CP}	CLKIN to PCLK2:1 Delay	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	−2 2	2 25	ns ns	(3,12) (3)
T	PCLK2:1 Period	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	T_C $2T_C$		ns ns	(12) (3)
T_{PH}	PCLK2:1 High Time		(T/2) − 4	T/2	ns	(12)
T_{PL}	PCLK2:1 Low Time		(T/2) − 4	T/2	ns	(12)
T_{PR}	PCLK2:1 Rise Time		1	4	ns	(3)
T_{PF}	PCLK2:1 Fall Time		1	4	ns	(3)
Synchronous Outputs (8)						
T_{OH} T_{OV}	Output Valid Delay, Output Hold					(6,10)
	T_{OH1}, T_{OV1}	A31:2	3	18	ns	
	T_{OH2}, T_{OV2}	BE3:0	3	20	ns	
	T_{OH3}, T_{OV3}	ADS	6	22	ns	
	T_{OH4}, T_{OV4}	W/R	3	22	ns	
	T_{OH5}, T_{OV5}	D/C, SUP, DMA	4	20	ns	
	T_{OH6}, T_{OV6}	BLAST, WAIT	5	20	ns	
	T_{OH7}, T_{OV7}	DEN	3	20	ns	
	T_{OH8}, T_{OV8}	HOLDA, BREQ	4	20	ns	
	T_{OH9}, T_{OV9}	LOCK	4	20	ns	
	T_{OH10}, T_{OV10}	DACK3:0	4	22	ns	
	T_{OH11}, T_{OV11}	D31:0	3	20	ns	
	T_{OH12}, T_{OV12}	DT/R	T/2 + 3	T/2 + 18	ns	
	T_{OH13}, T_{OV13}	FAIL	2	18	ns	
	T_{OH14}, T_{OV14}	EOP3:0/TC3:0	3	22	ns	(6,10)
T_{OF}	Output Float for all outputs		3	22	ns	(6)
Synchronous Inputs (1,9,10)						
T_{IS}	Input Setup					
	T_{IS1}	D31:0	5		ns	
	T_{IS2}	BOFF	21		ns	
	T_{IS3}	BTERM/READY	9		ns	
	T_{IS4}	HOLD	9		ns	
T_{IH}	Input Hold					
	T_{IH1}	D31:0	5		ns	
	T_{IH2}	BOFF	7		ns	
	T_{IH3}	BTERM/READY	2		ns	
	T_{IH4}	HOLD	5		ns	

Table 18. 80960CA AC Characteristics (16 MHz) (Continued)

(80960CA-16 only, under conditions described in **Section 4.2, Operating Conditions** and **Section 4.5.1, AC Test Conditions.**)

Symbol	Parameter	Min	Max	Units	Notes
Relative Output Timings (1,2,3,8)					
T_{AVSH1}	A31:2 Valid to $\overline{ADS}$ Rising	T – 4	T + 4	ns	
T_{AVSH2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, D/$\overline{C}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{ADS}$ Rising	T – 6	T + 6	ns	
T_{AVEL1}	A31:2 Valid to $\overline{DEN}$ Falling	T – 6	T + 6	ns	
T_{AVEL2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, $\overline{INST}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{DEN}$ Falling	T – 6	T + 6	ns	
T_{NLQV}	$\overline{WAIT}$ Falling to Output Data Valid	± 4		ns	
T_{DVNH}	Output Data Valid to $\overline{WAIT}$ Rising	N*T – 4	N*T + 4	ns	(4)
T_{NLNH}	$\overline{WAIT}$ Falling to $\overline{WAIT}$ Rising	N*T ± 4		ns	(4)
T_{NHQX}	Output Data Hold after $\overline{WAIT}$ Rising	(N+1)*T–8	(N+1)*T+6	ns	(5)
T_{EHTV}	DT/$\overline{R}$ Hold after $\overline{DEN}$ High	T/2 – 7	∞	ns	(6)
T_{TVEL}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Falling	T/2 – 4		ns	
Relative Input Timings (1,2,3)					
T_{IS5}	$\overline{RESET}$ Input Setup (2-x Clock Mode)	10		ns	(13)
T_{IH5}	$\overline{RESET}$ Input Hold (2-x Clock Mode)	9		ns	(13)
T_{IS6}	$\overline{DREQ3:0}$ Input Setup	16		ns	(7)
T_{IH6}	$\overline{DREQ3:0}$ Input Hold	11		ns	(7)
T_{IS7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Setup	9		ns	(15)
T_{IH7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Hold	5		ns	(15)
T_{IS8}	$\overline{RESET}$ Input Setup (1-x Clock Mode)	3		ns	(14)
T_{IH8}	$\overline{RESET}$ Input Hold (1-x Clock Mode)	T/4 + 1		ns	(14)

NOTES:

1. See **Section 4.5.2, AC Timing Waveforms** for waveforms and definitions.
2. See Figure 16 for capacitive derating information for output delays and hold times.
3. See Figure 17 for capacitive derating information for rise and fall times.
4. Where N is the number of N_{RAD}, N_{RDD}, N_{WAD} or N_{WDD} wait states that are programmed in the Bus Controller Region Table. $\overline{WAIT}$ never goes active when there are no wait states in an access.
5. N = Number of wait states inserted with $\overline{READY}$.
6. Output Data and/or DT/$\overline{R}$ may be driven indefinitely following a cycle if there is no subsequent bus activity.
7. Since asynchronous inputs are synchronized internally by the 80960CA, they have no required setup or hold times to be recognized and for proper operation. However, to guarantee recognition of the input at a particular edge of PCLK2:1, the setup times shown must be met. Asynchronous inputs must be active for at least two consecutive PCLK2:1 rising edges to be seen by the processor.
8. These specifications are guaranteed by the processor.
9. These specifications must be met by the system for proper operation of the processor.
10. This timing is dependent upon the loading of PCLK2:1. Use the derating curves of **Section 4.5.3, Derating Curves** to adjust the timing for PCLK2:1 loading.
11. In the 1-x input clock mode, the maximum input clock period is limited to 125 ns while the processor is operating. When the processor is in reset, the input clock may stop even in 1-x mode.
12. When in the 1-x input clock mode, these specifications assume a stable input clock with a period variation of less than ± 0.1% between adjacent cycles.
13. In 2-x clock mode, $\overline{RESET}$ is an asynchronous input which has no required setup and hold time for proper operation. However, to guarantee the device exits reset synchronized to a particular clock edge, the $\overline{RESET}$ pin must meet setup and hold times to the falling edge of the CLKIN. (See Figure 22.)
14. In 1-x clock mode, $\overline{RESET}$ is an asynchronous input which has no required setup and hold time for proper operation. However, to guarantee the device exits reset synchronized to a particular clock edge, the $\overline{RESET}$ pin must meet setup and hold times to the rising edge of the CLKIN. (See Figure 23.)
15. The interrupt pins are synchronized internally by the 80960CA. They have no required setup or hold times for proper operation. These pins are sampled by the interrupt controller every other clock and must be active for at least three consecutive PCLK2:1 rising edges when asserting them asynchronously. To guarantee recognition at a particular clock edge, the setup and hold times shown must be met for two consecutive PCLK2:1 rising edges.

4.5.1 AC Test Conditions

The AC Specifications in Section 4.5 are tested with the 50 pF load shown in Figure 7. Figure 16 shows how timings vary with load capacitance.

Specifications are measured at the 1.5V crossing point, unless otherwise indicated. Input waveforms are assumed to have a rise and fall time of ≤ 2 ns from 0.8V to 2.0V. See **Section 4.5.2, AC Timing Waveforms** for AC spec definitions, test points and illustrations.

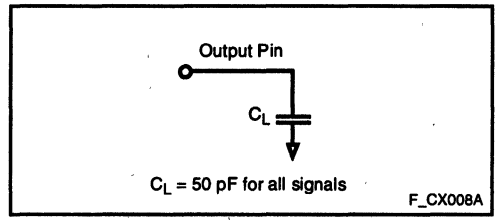


Figure 7. AC Test Load

4.5.2 AC Timing Waveforms

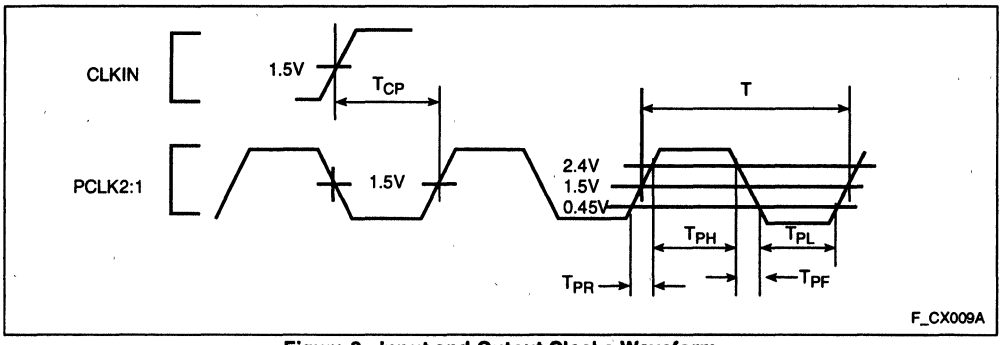


Figure 8. Input and Output Clocks Waveform

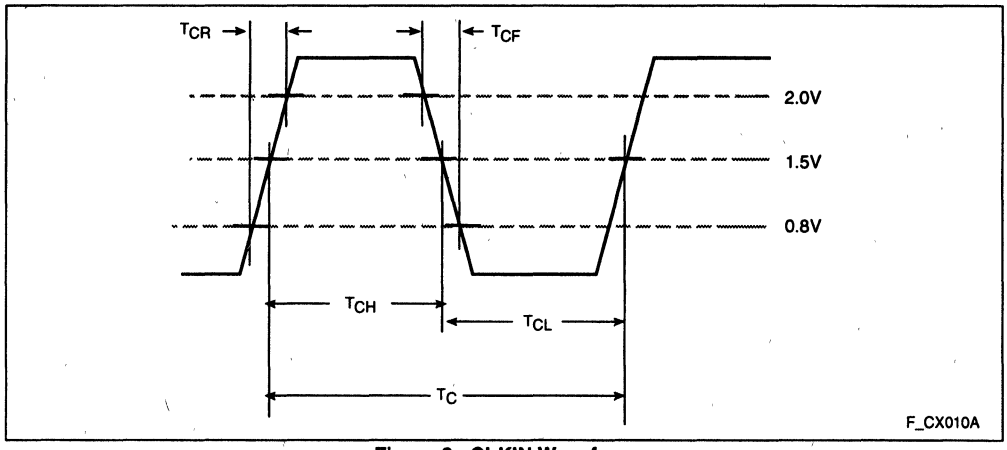


Figure 9. CLKIN Waveform

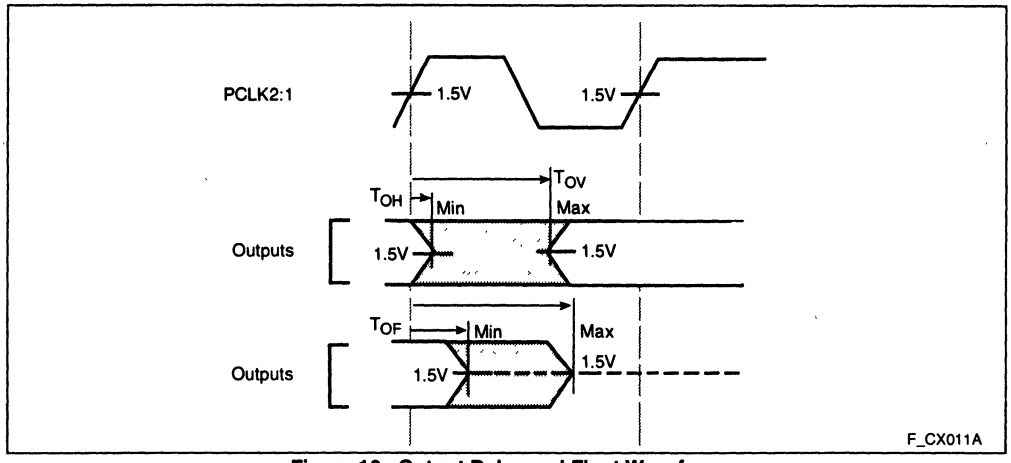


Figure 10. Output Delay and Float Waveform

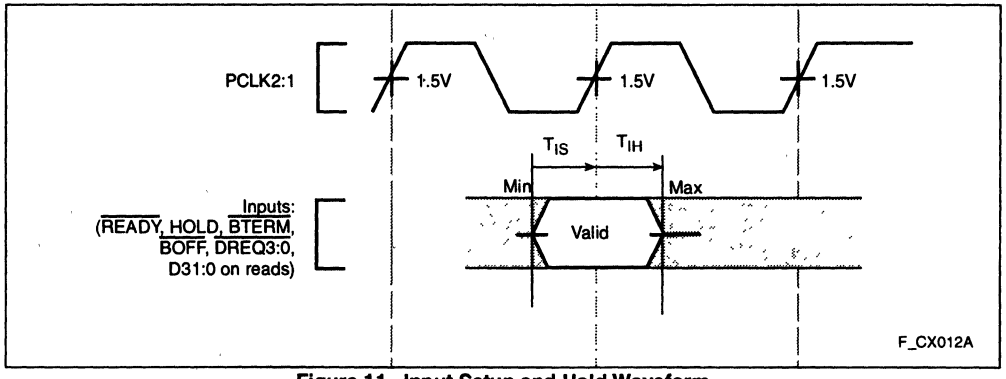


Figure 11. Input Setup and Hold Waveform

T_{OV} T_{OH} - OUTPUT DELAY - The maximum output delay is referred to as the Output Valid Delay (T_{OV}). The minimum output delay is referred to as the Output Hold (T_{OH}).

T_{OF} - OUTPUT FLOAT DELAY - The output float condition occurs when the maximum output current becomes less that I_{LO} in magnitude.

T_{IS} T_{IH} - INPUT SETUP AND HOLD - The input setup and hold requirements specify the sampling window during which synchronous inputs must be stable for correct processor operation.

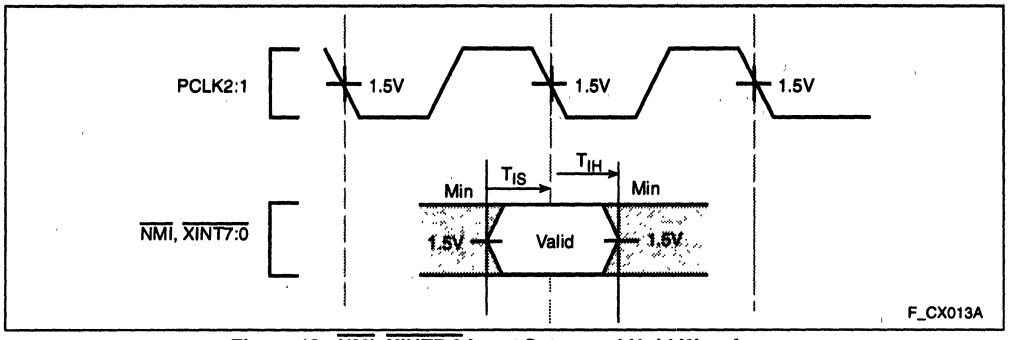
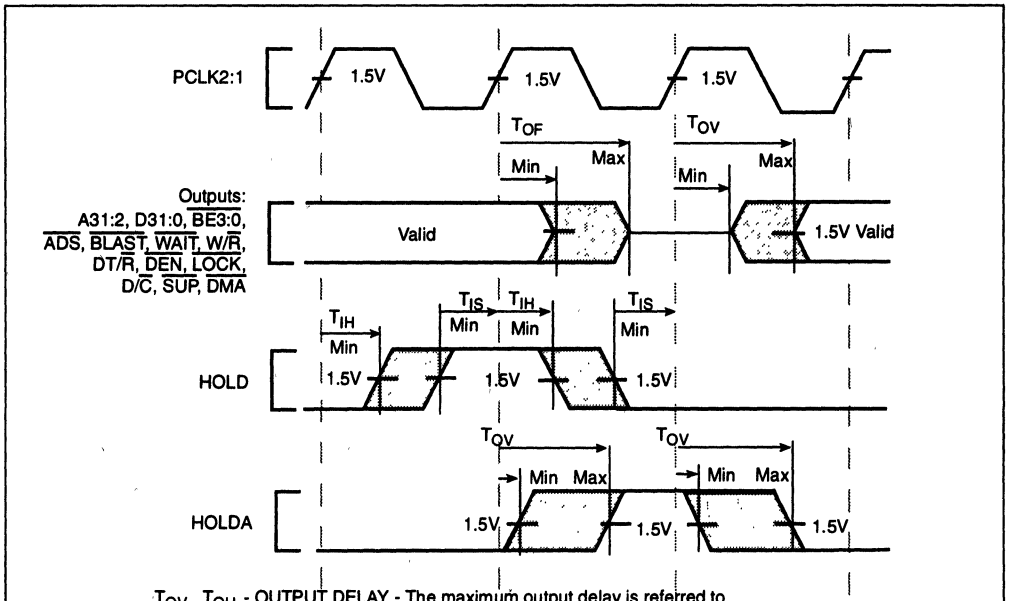


Figure 12. NMI, XINT7:0 Input Setup and Hold Waveform



T_{OV} T_{OH} - OUTPUT DELAY - The maximum output delay is referred to as the Output Valid Delay (T_{OV}). The minimum output delay is referred to as the Output Hold (T_{OH}).

T_{OF} - OUTPUT FLOAT DELAY - The output float condition occurs when the maximum output current becomes less that I_{LO} in magnitude.

T_{IS} T_{IH} - INPUT SETUP AND HOLD - The input setup and hold requirements specify the sampling window during which synchronous inputs must be stable for correct processor operation.

F_CX014A

Figure 13. Hold Acknowledge Timings

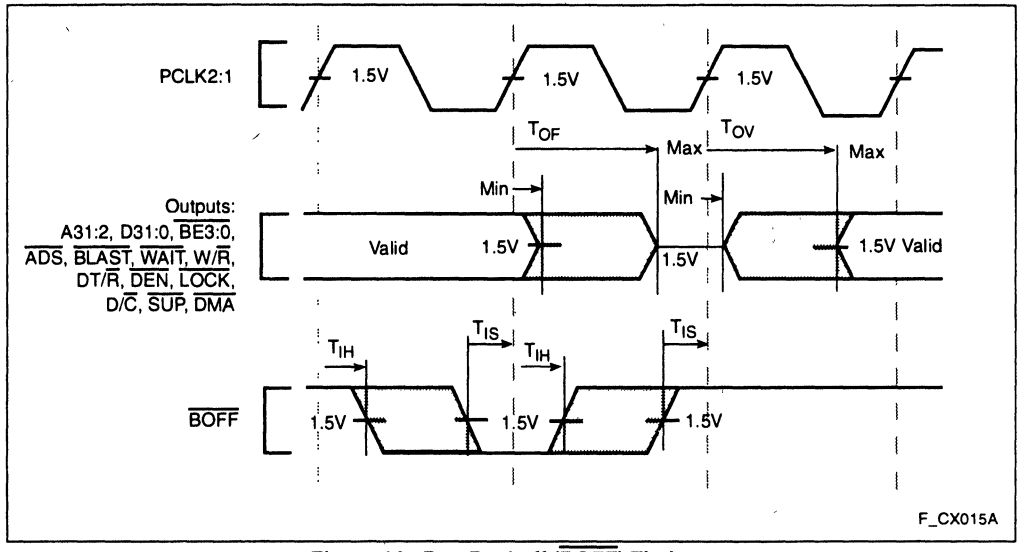


Figure 14. Bus Backoff (BOFF) Timings

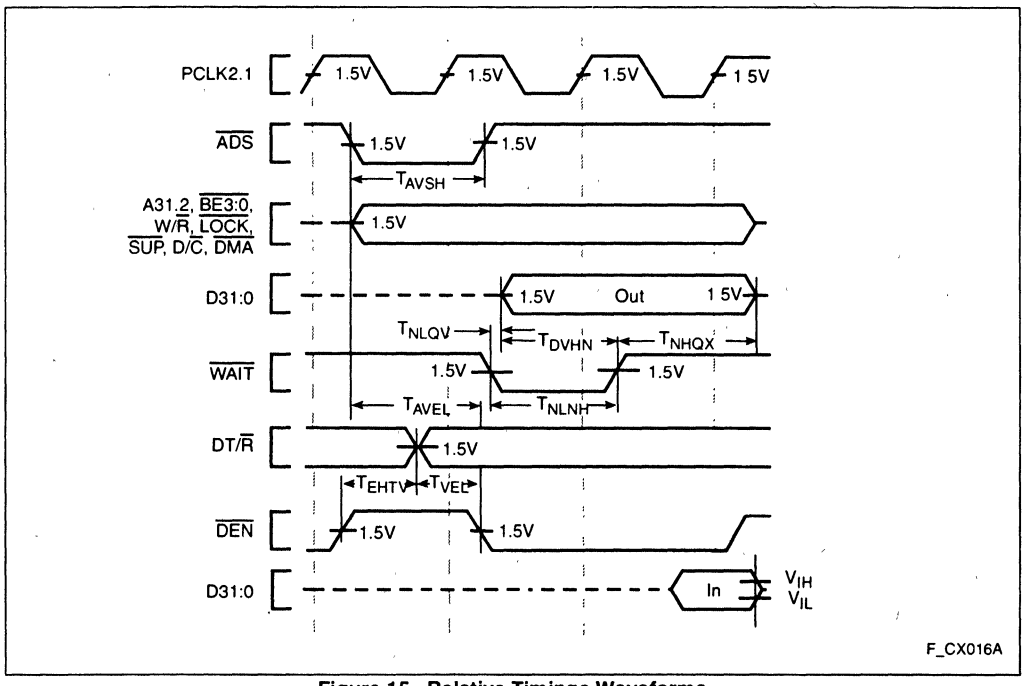


Figure 15. Relative Timings Waveforms

4.5.3 Derating Curves

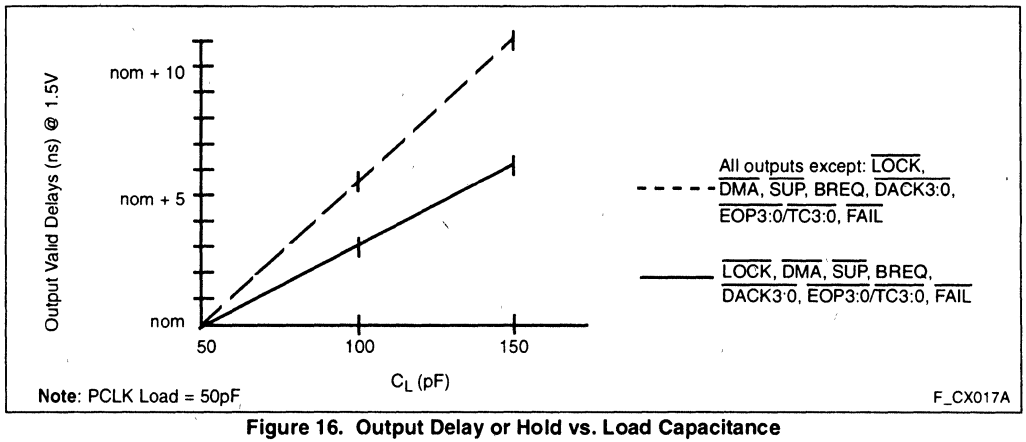


Figure 16. Output Delay or Hold vs. Load Capacitance

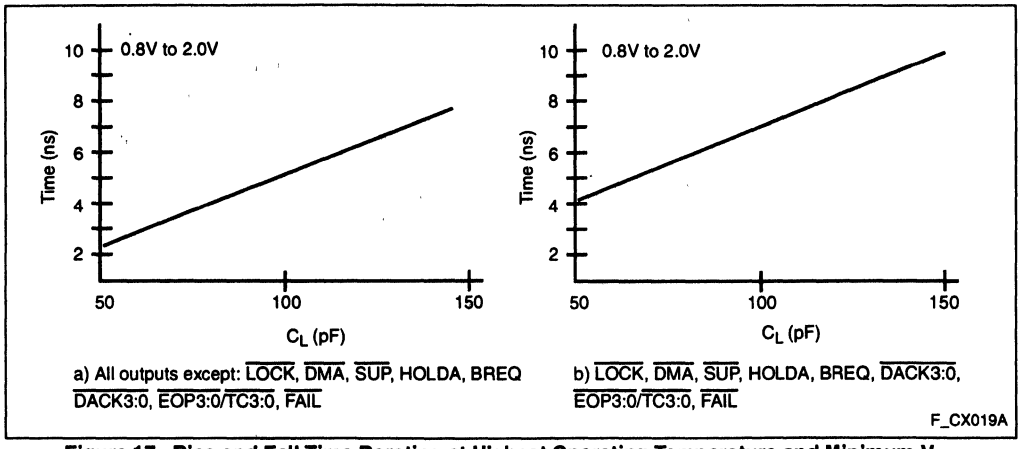


Figure 17. Rise and Fall Time Derating at Highest Operating Temperature and Minimum V_{CC}

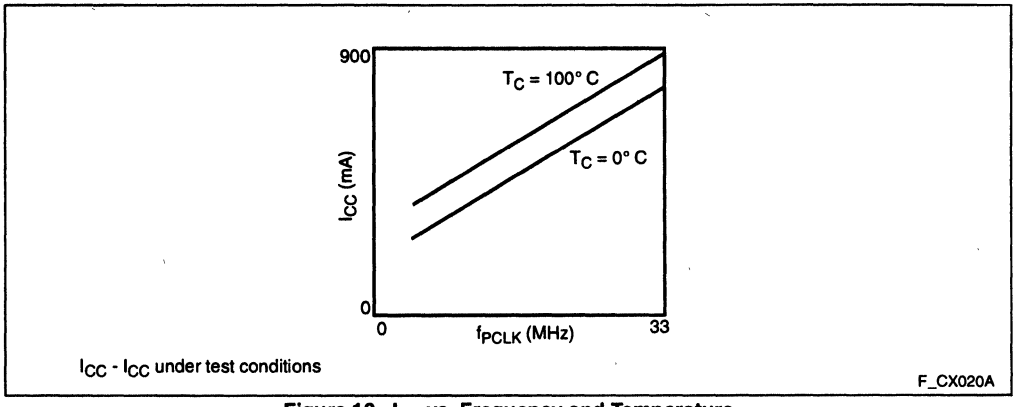


Figure 18. I_{CC} vs. Frequency and Temperature

5.0 RESET, BACKOFF AND HOLD ACKNOWLEDGE

Table 19 lists the condition of each processor output pin while RESET is asserted (low).

Table 19. Reset Conditions

Pins	State During Reset (HOLDA Inactive)[1]
A31:2	Floating
D31:0	Floating
BE3:0	Driven high (Inactive)
W/R	Driven low (Read)
ADS	Driven high (Inactive)
WAIT	Driven high (Inactive)
BLAST	Driven low (Active)
DT/R	Driven low (Receive)
DEN	Driven high (Inactive)
LOCK	Driven high (Inactive)
BREQ	Driven low (Inactive)
D/C	Floating
DMA	Floating
SUP	Floating
FAIL	Driven low (Active)
DACK3:0	Driven high (Inactive)
EOP3:0/TC3:0	Floating (Set to input mode)

NOTES:

1. With regard to bus output pin state only, the Hold Acknowledge state takes precedence over the reset state. Although asserting the RESET pin will internally reset the processor, the processor's bus output pins will not enter the reset state if it has granted Hold Acknowledge to a previous HOLD request (HOLDA is active). Furthermore, the processor will grant new HOLD requests and enter the Hold Acknowledge state even while in reset.
   For example, if HOLDA is inactive and the processor is in the reset state, then HOLD is asserted, the processor's bus pins enter the Hold Acknowledge state and HOLDA is granted. The processor will not be able to perform memory accesses until the HOLD request is removed, even if the RESET pin is brought high. This operation is provided to simplify boot-up synchronization among multiple processors sharing the same bus.

Table 20 lists the condition of each processor output pin while HOLDA is asserted (low).

Table 20. Hold Acknowledge and Backoff Conditions

Pins	State During HOLDA
A31:2	Floating
D31:0	Floating
BE3:0	Floating
W/R	Floating
ADS	Floating
WAIT	Floating
BLAST	Floating
DT/R	Floating
DEN	Floating
LOCK	Floating
BREQ	Driven (High or low)
D/C	Floating
DMA	Floating
SUP	Floating
FAIL	Driven high (Inactive)
DACK3:0	Driven high (Inactive)
EOP3:0/TC3:0	Driven (If output)

6.0 BUS WAVEFORMS

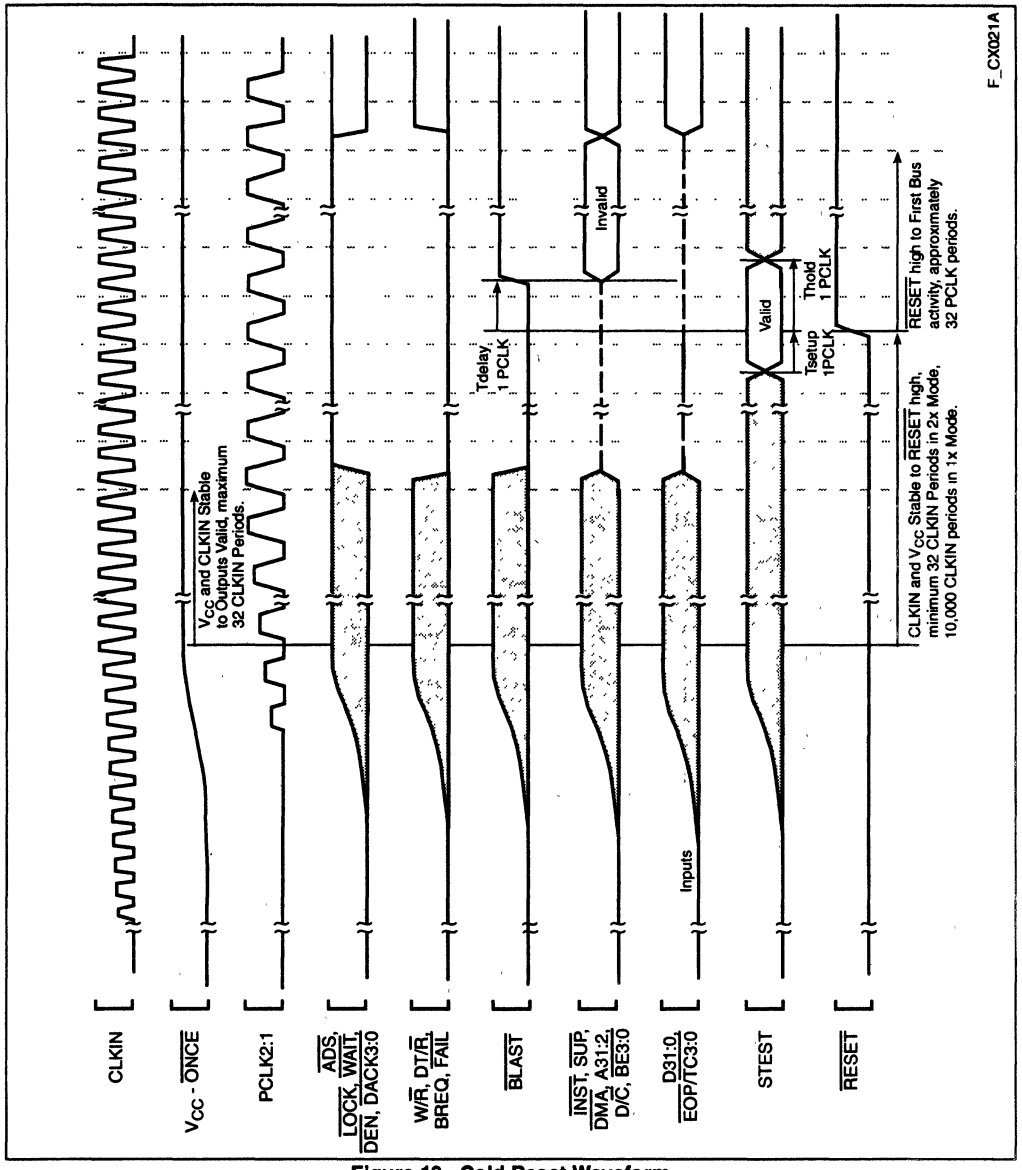


Figure 19. Cold Reset Waveform

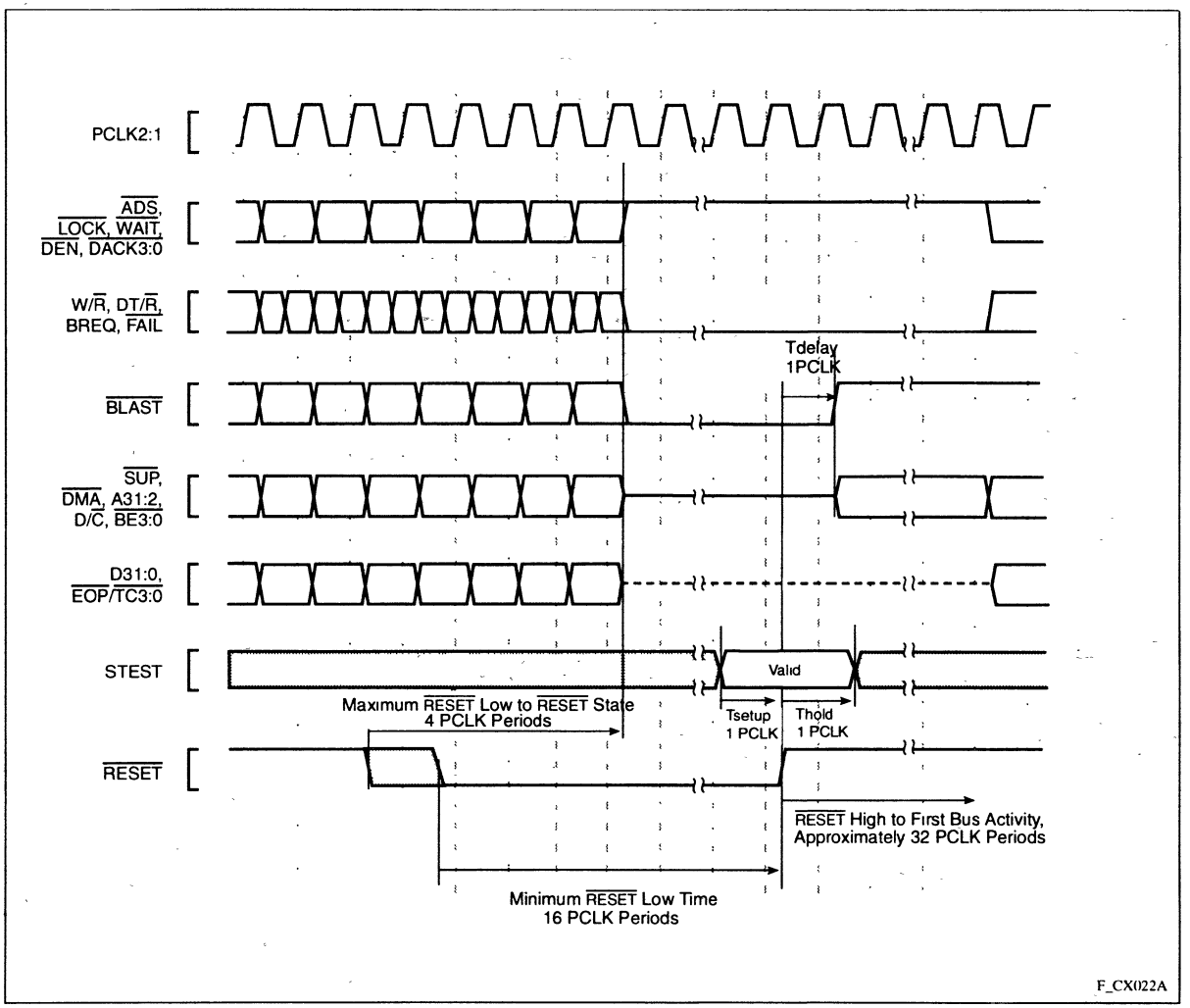


Figure 20. Warm Reset Waveform

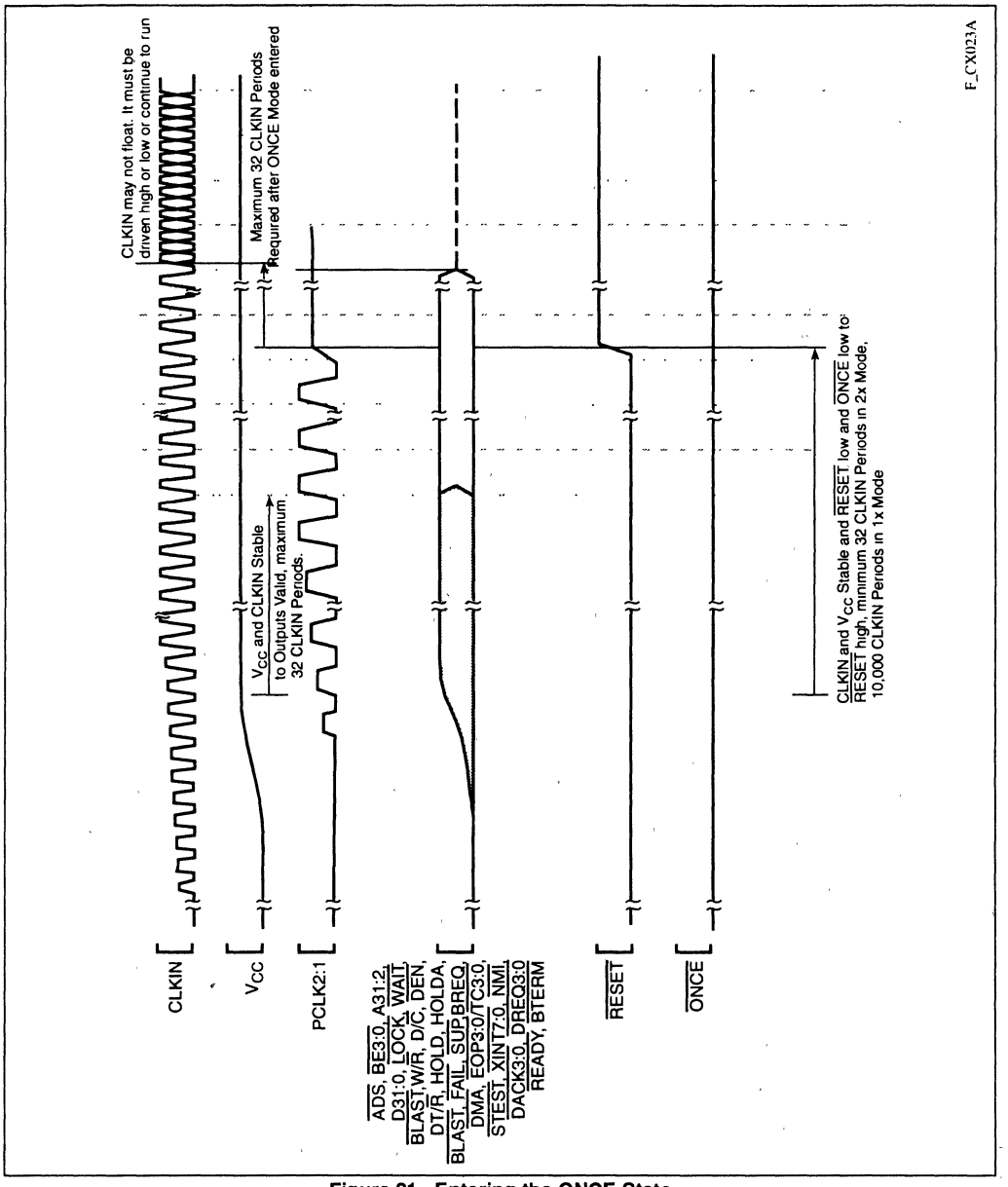
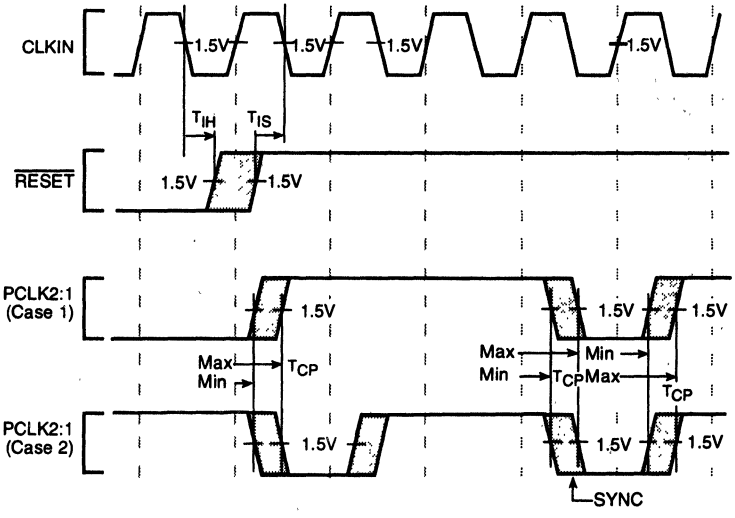


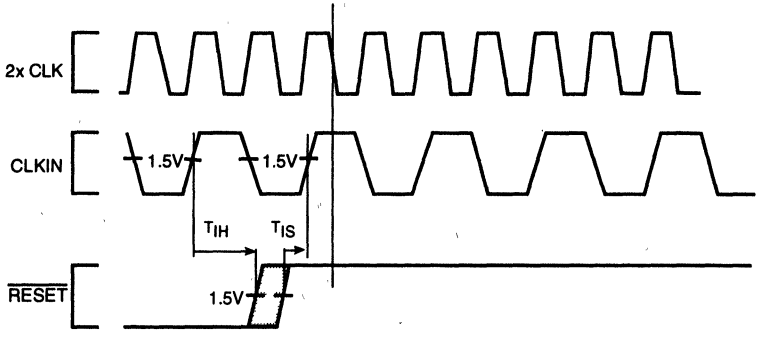
Figure 21. Entering the ONCE State



Note: Case 1 and Case 2 show two possible polarities of PCLK2:1

F_CX024A

Figure 22. Clock Synchronization in the 2-x Clock Mode



Note: In 1x clock mode, the $\overline{\text{RESET}}$ pin is actually sampled on the falling edge of 2xCLK. 2xCLK is an internal signal generated by the PLL and is not available on an external pin. Therefore, $\overline{\text{RESET}}$ is specified relative to the rising edge of CLKIN. The $\overline{\text{RESET}}$ pin is sampled when PCLK is high.

F_CX025A

Figure 23. Clock Synchronization in the 1-x Clock Mode

Function	reserved	Byte Order	reserved	Bus Width	N_{WDD}	N_{WAD}	N_{XDA}	N_{RDD}	N_{RAD}	Pipe-Lining	External Ready Control	Burst
Bit	31-23	22	21	20-19	18-17	16-12	11-10	9-8	7-3	2	1	0
Value	0 0..0	X x	0 0	X xx	X xx	0 00000	0 00	X xx	0 00000	OFF 0	Disabled 0	Disabled 0

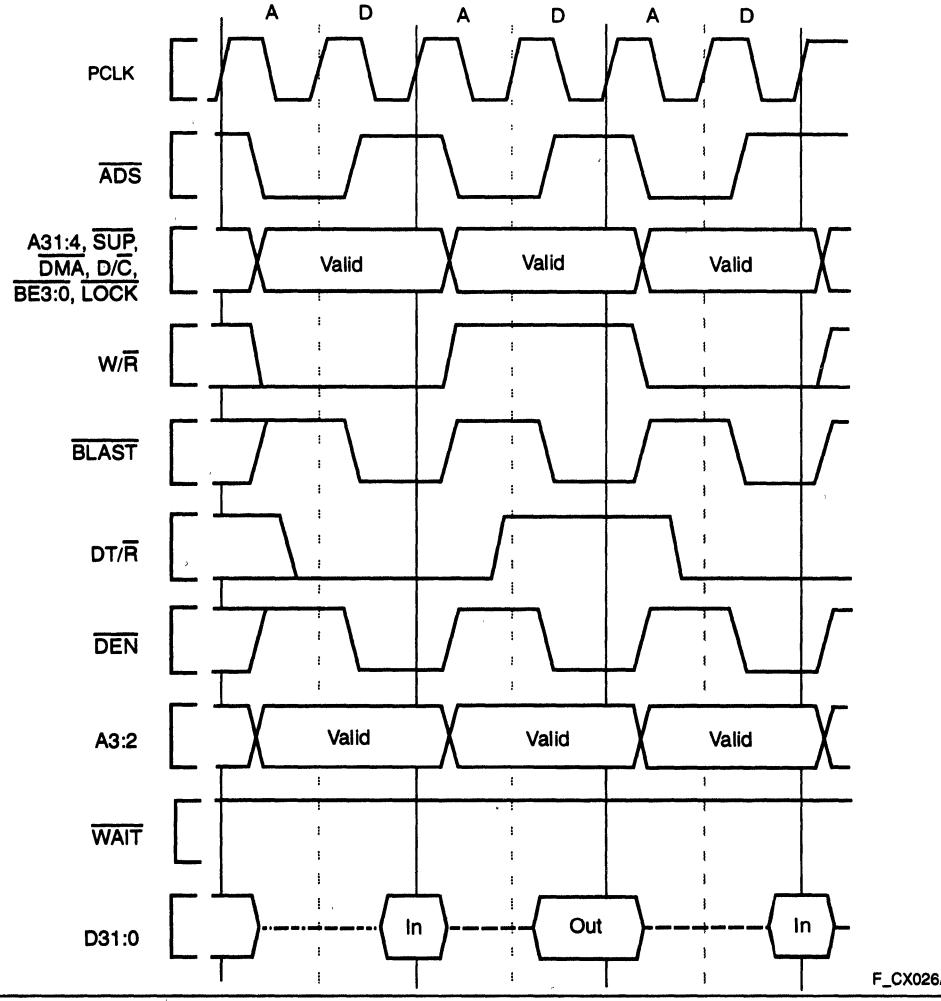


Figure 24. Non-Burst, Non-Pipelined Requests Without Wait States

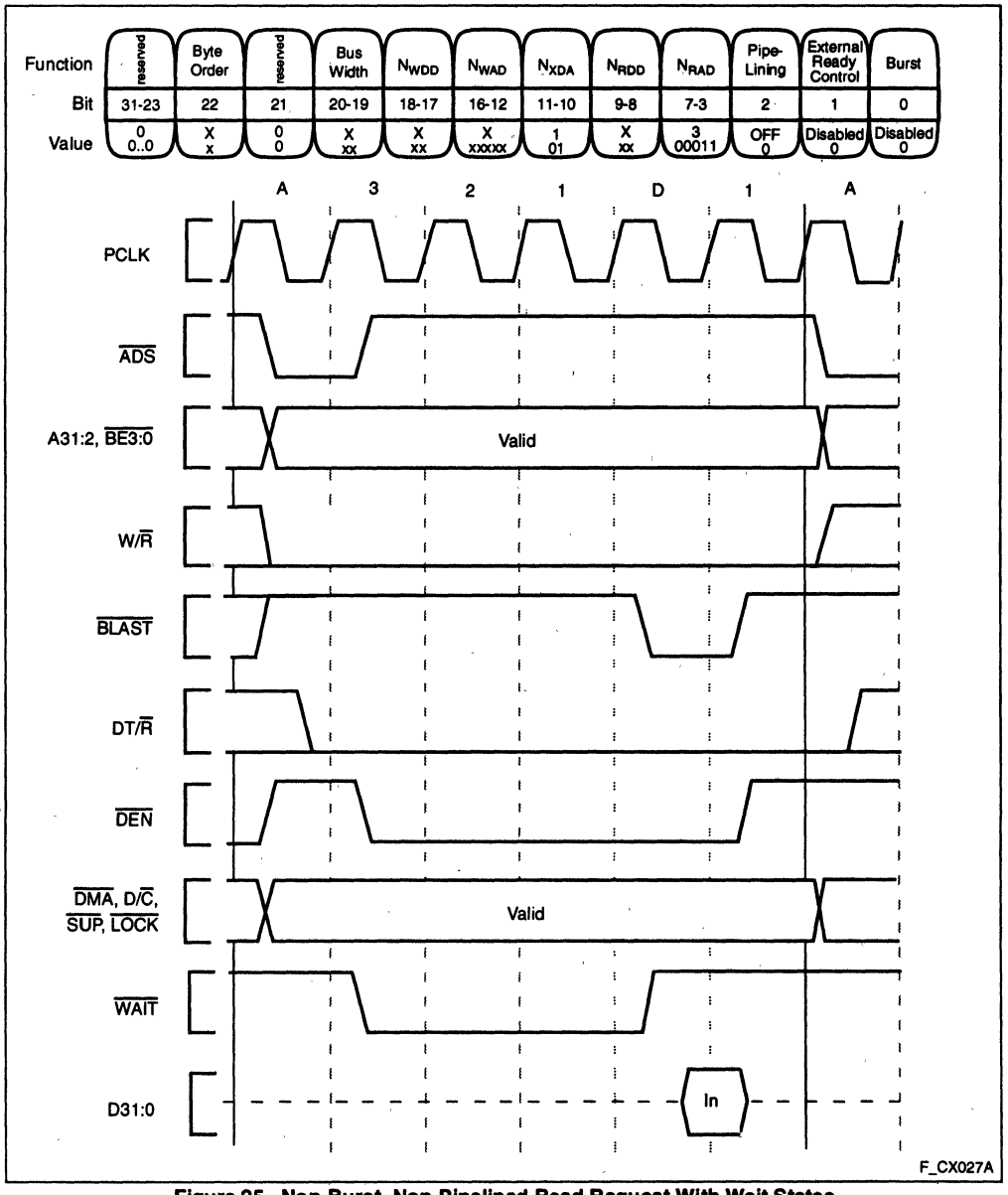


Figure 25. Non-Burst, Non-Pipelined Read Request With Wait States

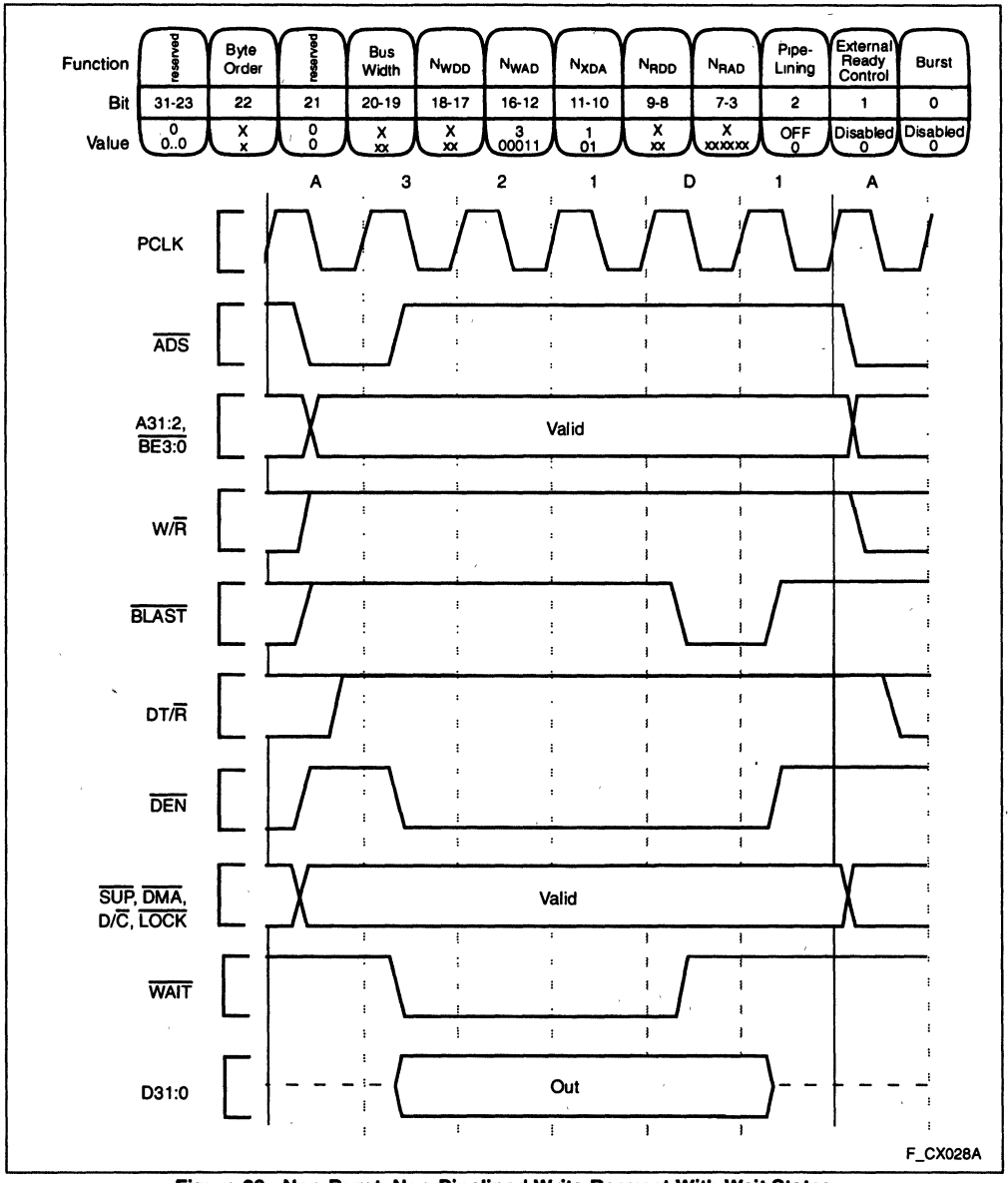


Figure 26. Non-Burst, Non-Pipelined Write Request With Wait States

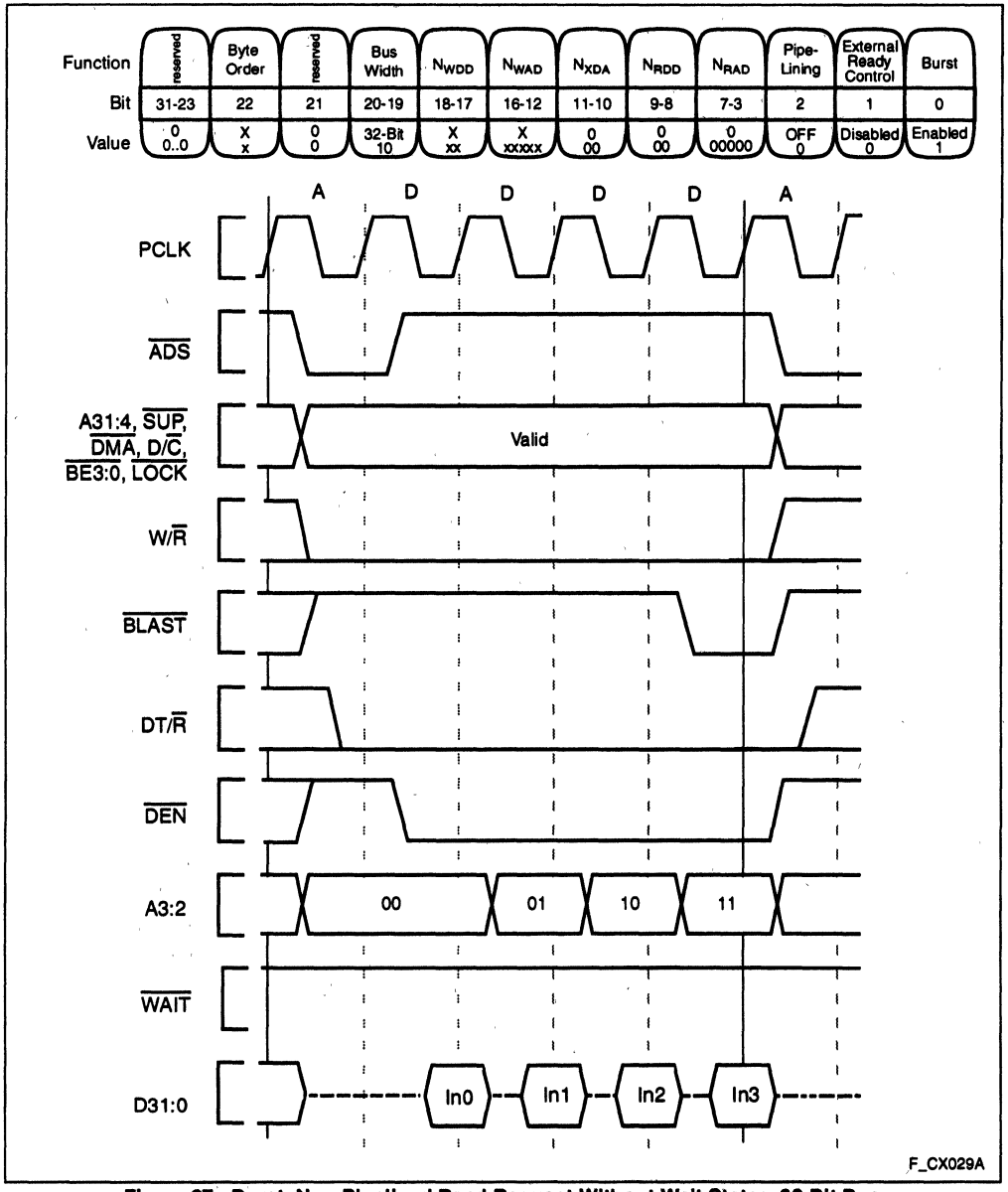


Figure 27. Burst, Non-Pipelined Read Request Without Wait States, 32-Bit Bus

Function	reserved	Byte Order	reserved	Bus Width	N_{WDD}	N_{WAD}	N_{XDA}	N_{RDD}	N_{RAD}	Pipe-Lining	External Ready Control	Burst
Bit	31-23	22	21	20-19	18-17	16-12	11-10	9-8	7-3	2	1	0
Value	0 0..0	X x	0 0	32-bit 10	X xx	X xxxxx	1 01	1 01	2 00010	OFF 0	Disabled 0	Enabled 1

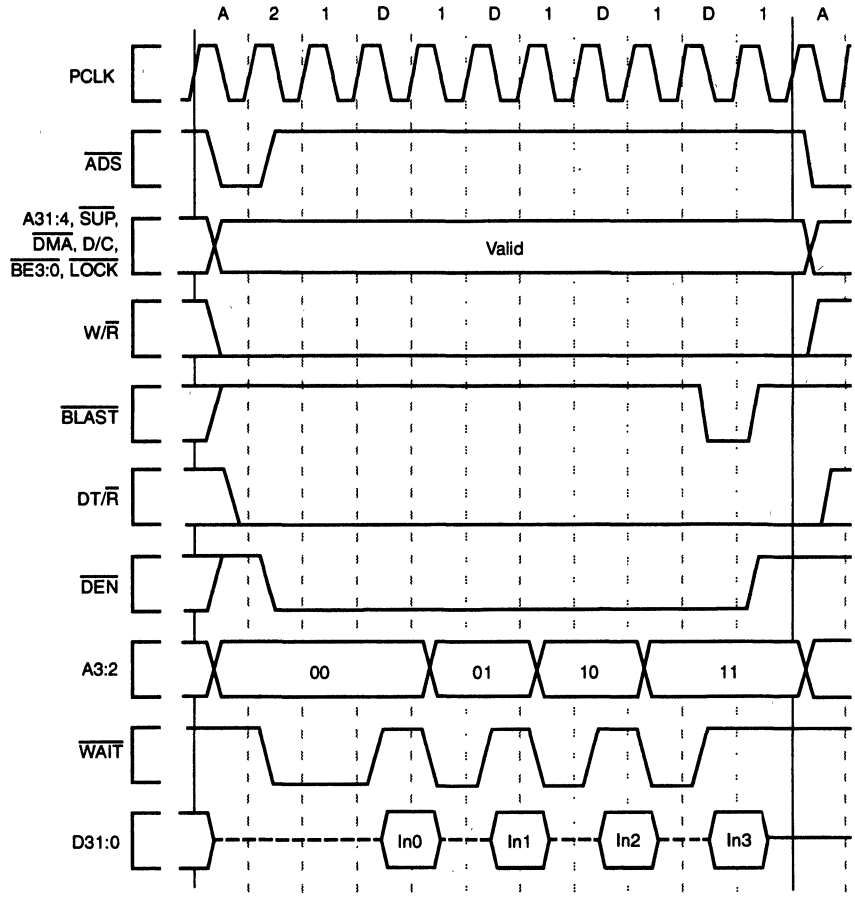


Figure 28. Burst, Non-Pipelined Read Request With Wait States, 32-Bit Bus

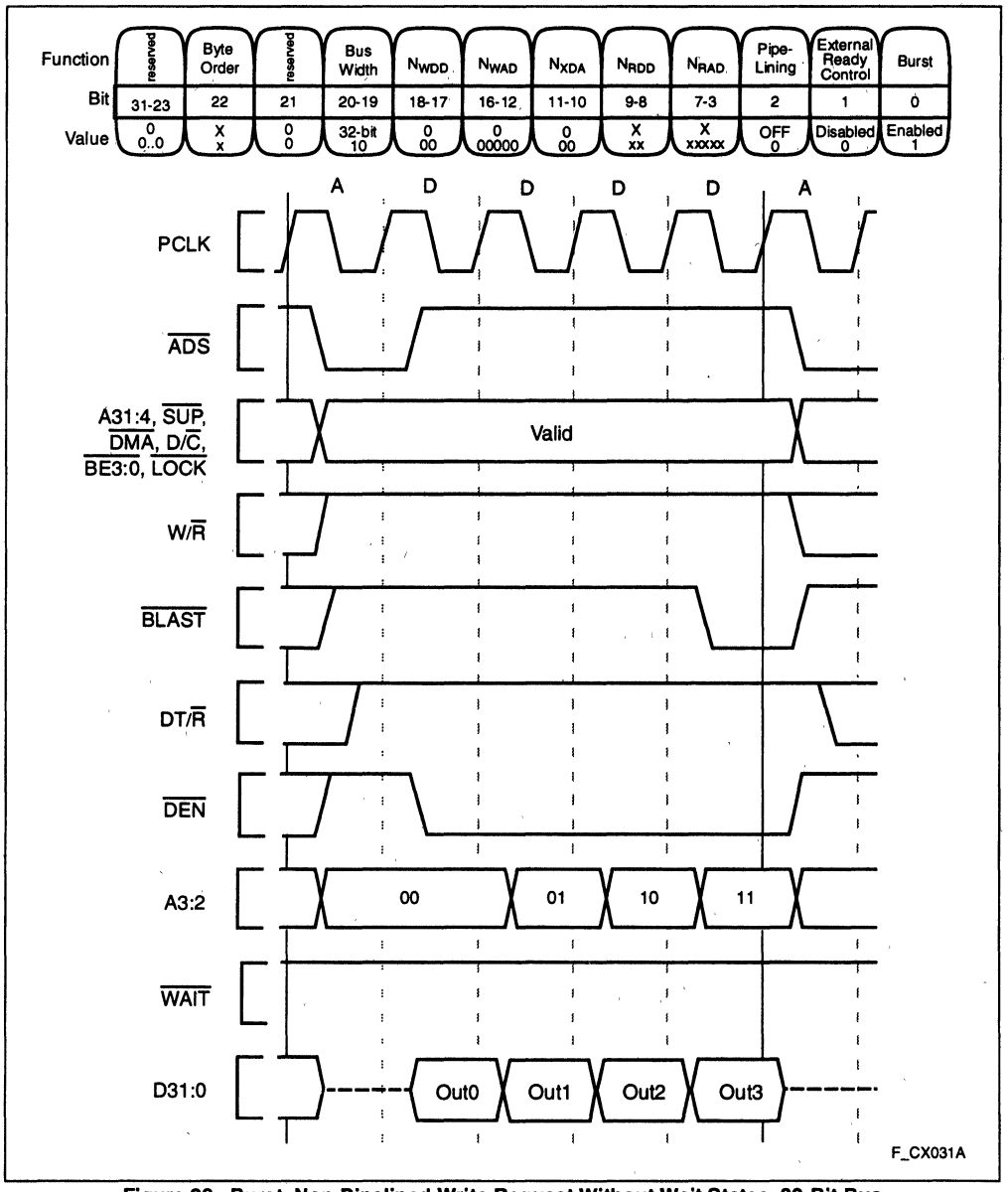


Figure 29. Burst, Non-Pipelined Write Request Without Wait States, 32-Bit Bus

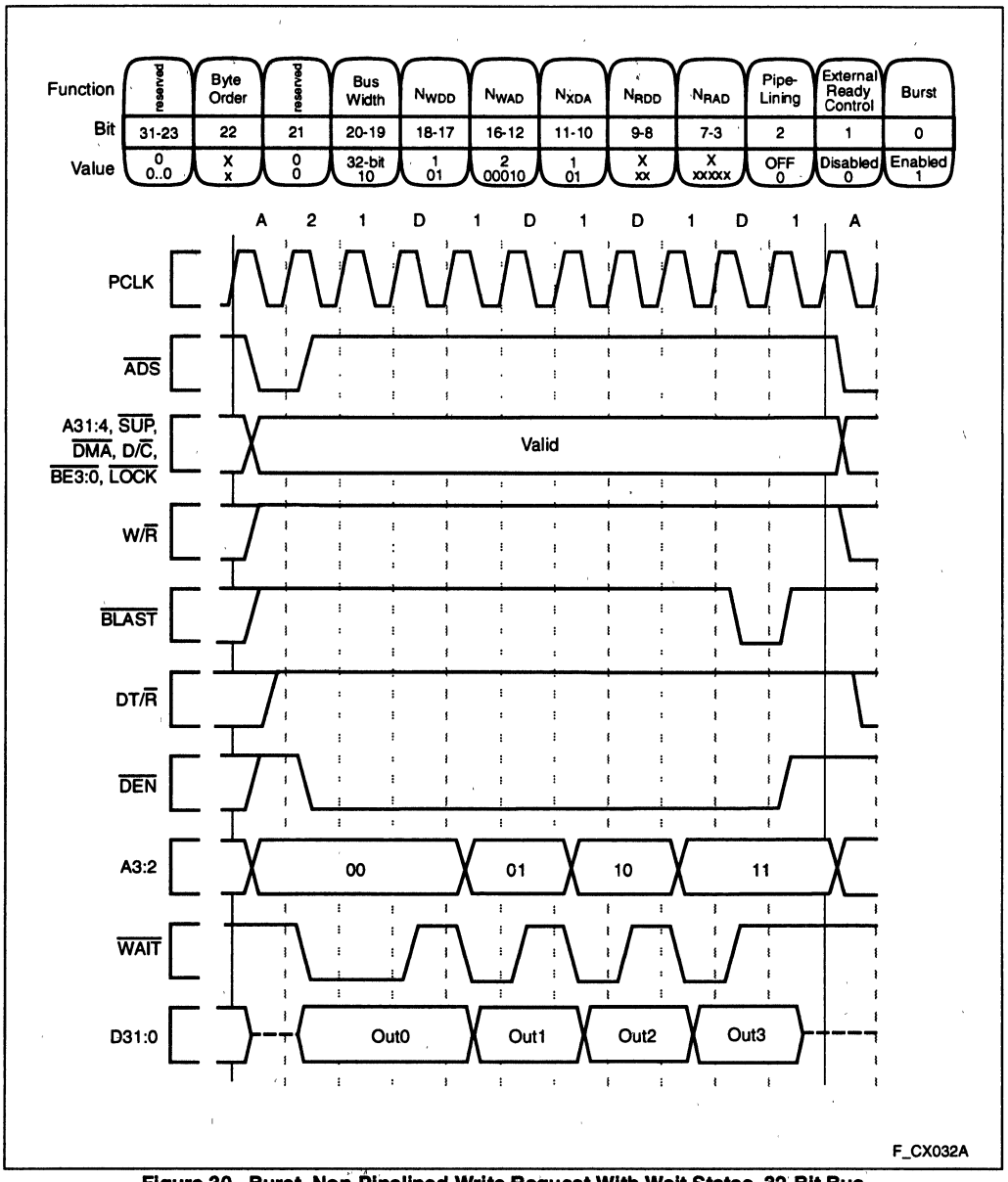


Figure 30. Burst, Non-Pipelined Write Request With Wait States, 32-Bit Bus

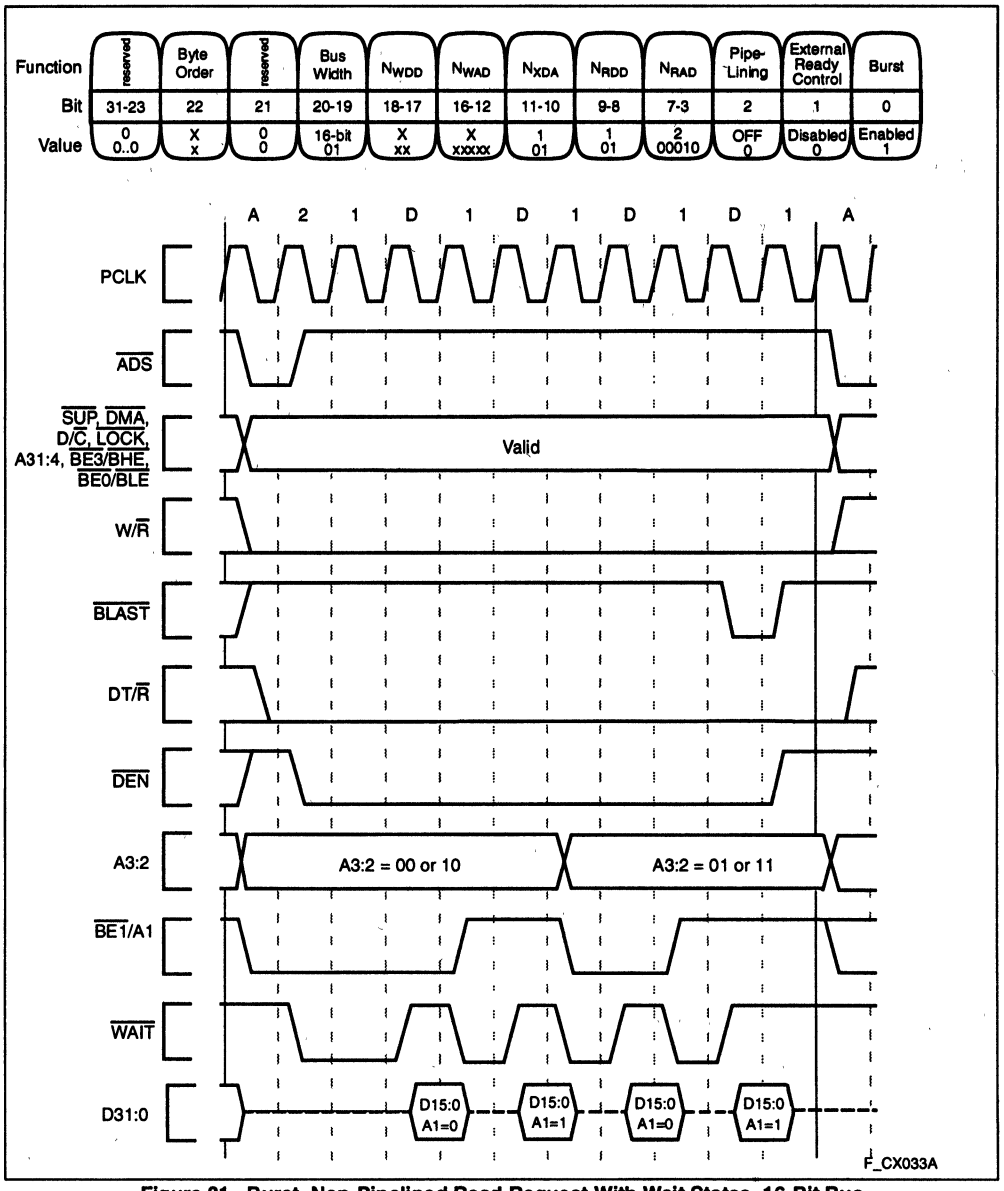


Figure 31. Burst, Non-Pipelined Read Request With Wait States, 16-Bit Bus

Function	reserved	Byte Order	reserved	Bus Width	N_{WDD}	N_{WAD}	N_{XDA}	N_{RDD}	N_{RAD}	Pipe-Lining	External Ready Control	Burst
Bit	31-23	22	21	20-19	18-17	16-12	11-10	9-8	7-3	2	1	0
Value	0 0..0	X x	0 0	8-bit 00	X xx	X xxxxx	1 01	1 01	2 00010	OFF 0	Disabled 0	Enabled 1

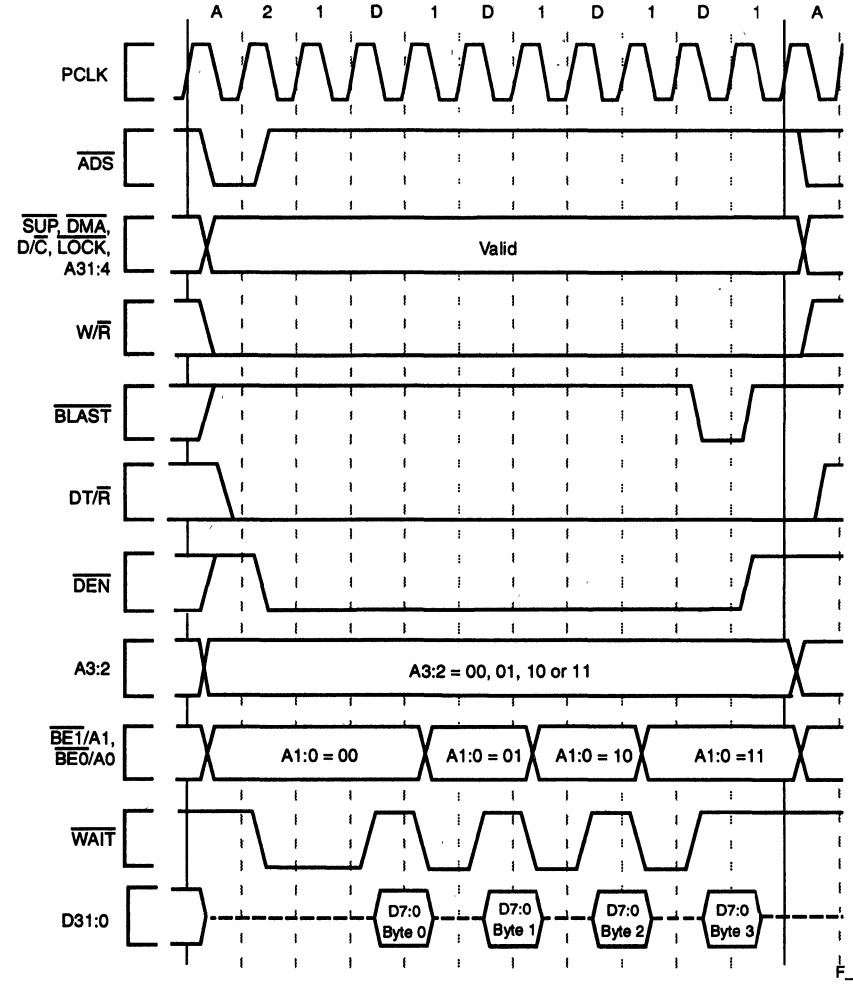


Figure 32. Burst, Non-Pipelined Read Request With Wait States, 8-Bit Bus

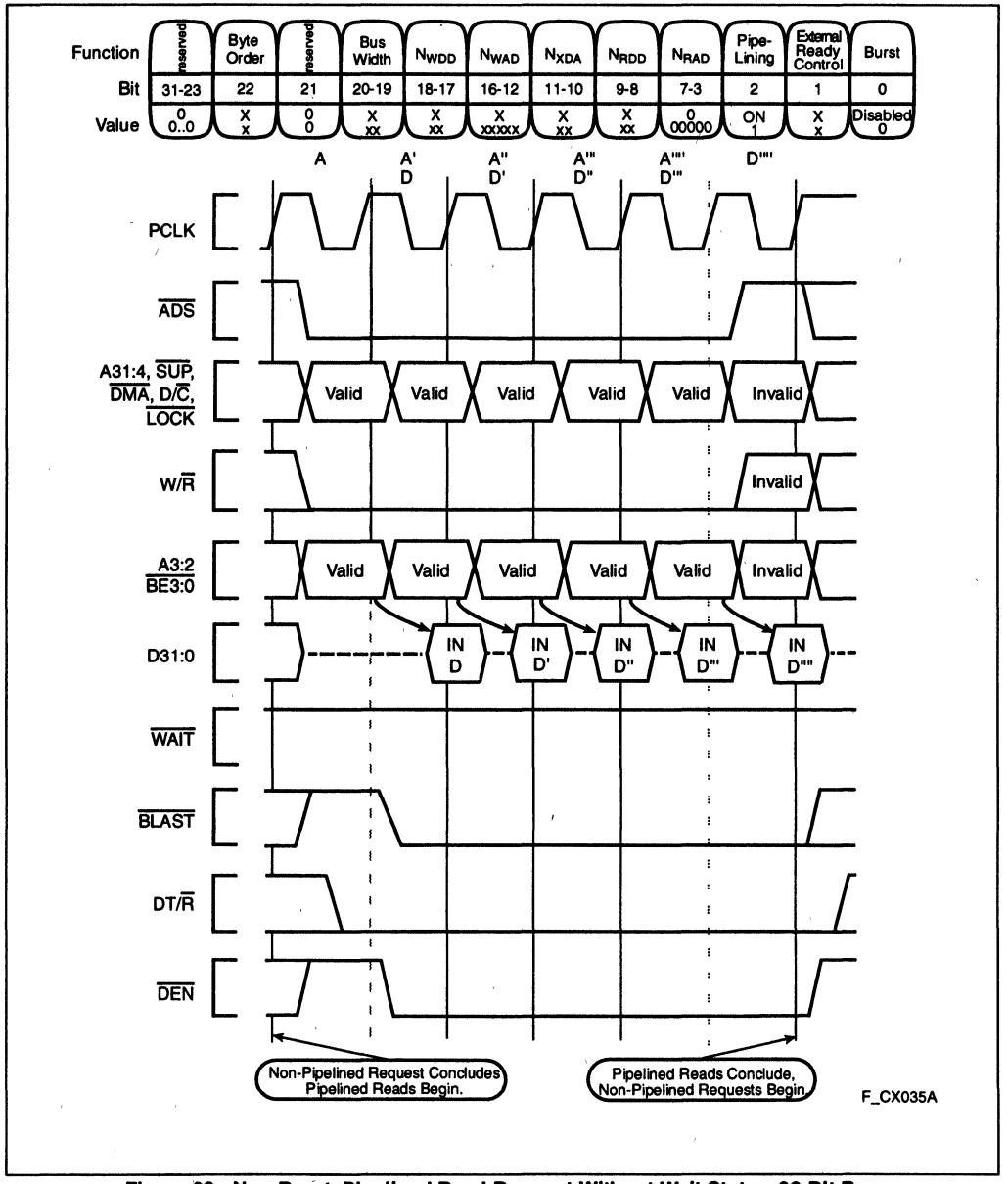


Figure 33. Non-Burst, Pipelined Read Request Without Wait States, 32-Bit Bus

Function	reserved	Byte Order	reserved	Bus Width	N_{WDD}	N_{WAD}	N_{XDA}	N_{RDD}	N_{RAD}	Pipe-Lining	External Ready Control	Burst
Bit	31-23	22	21	20-19	18-17	16-12	11-10	9-8	7-3	2	1	0
Value	0 0..0	X x	0 0	X xx	X xx	X xxxxx	X xx	X xx	1 00001	ON 1	X x	Disabled 1

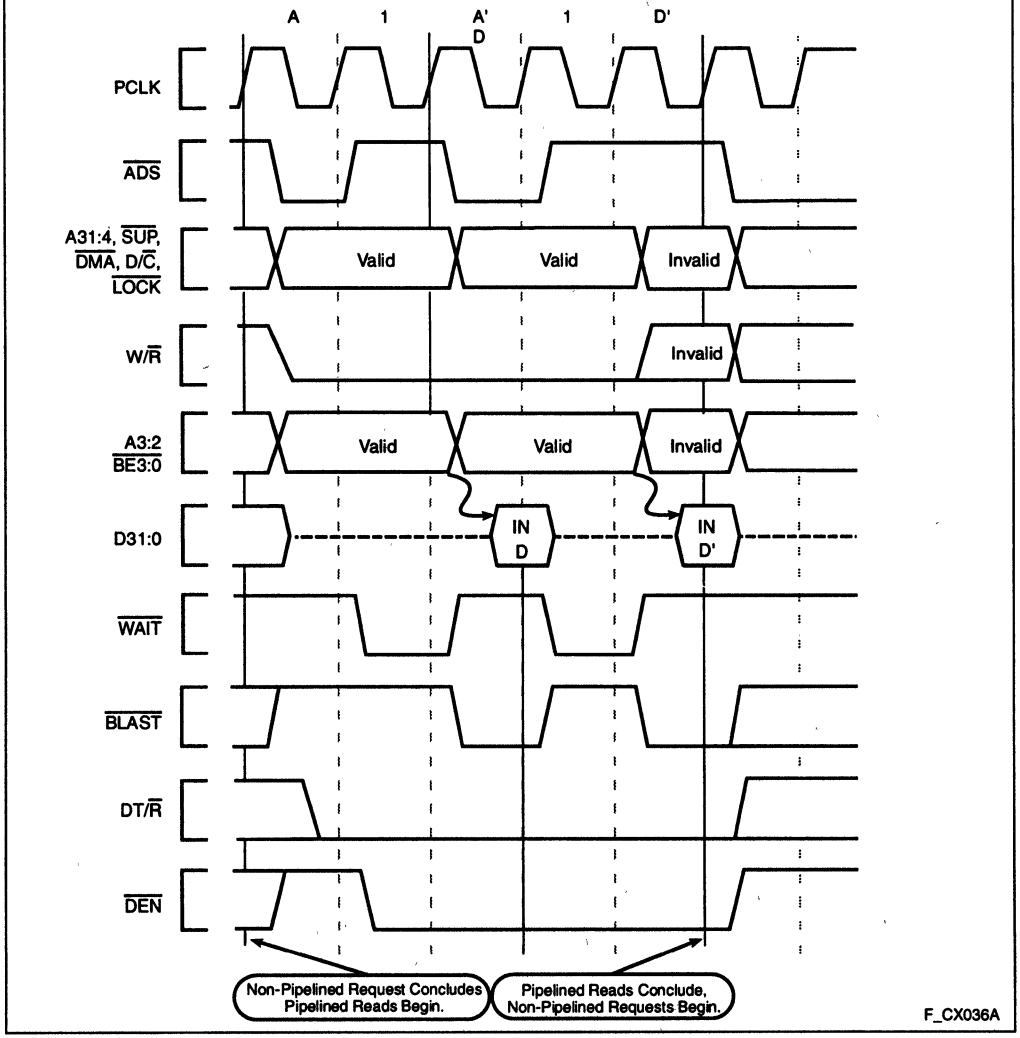


Figure 34. Non-Burst, Pipelined Read Request With Wait States, 32-Bit Bus

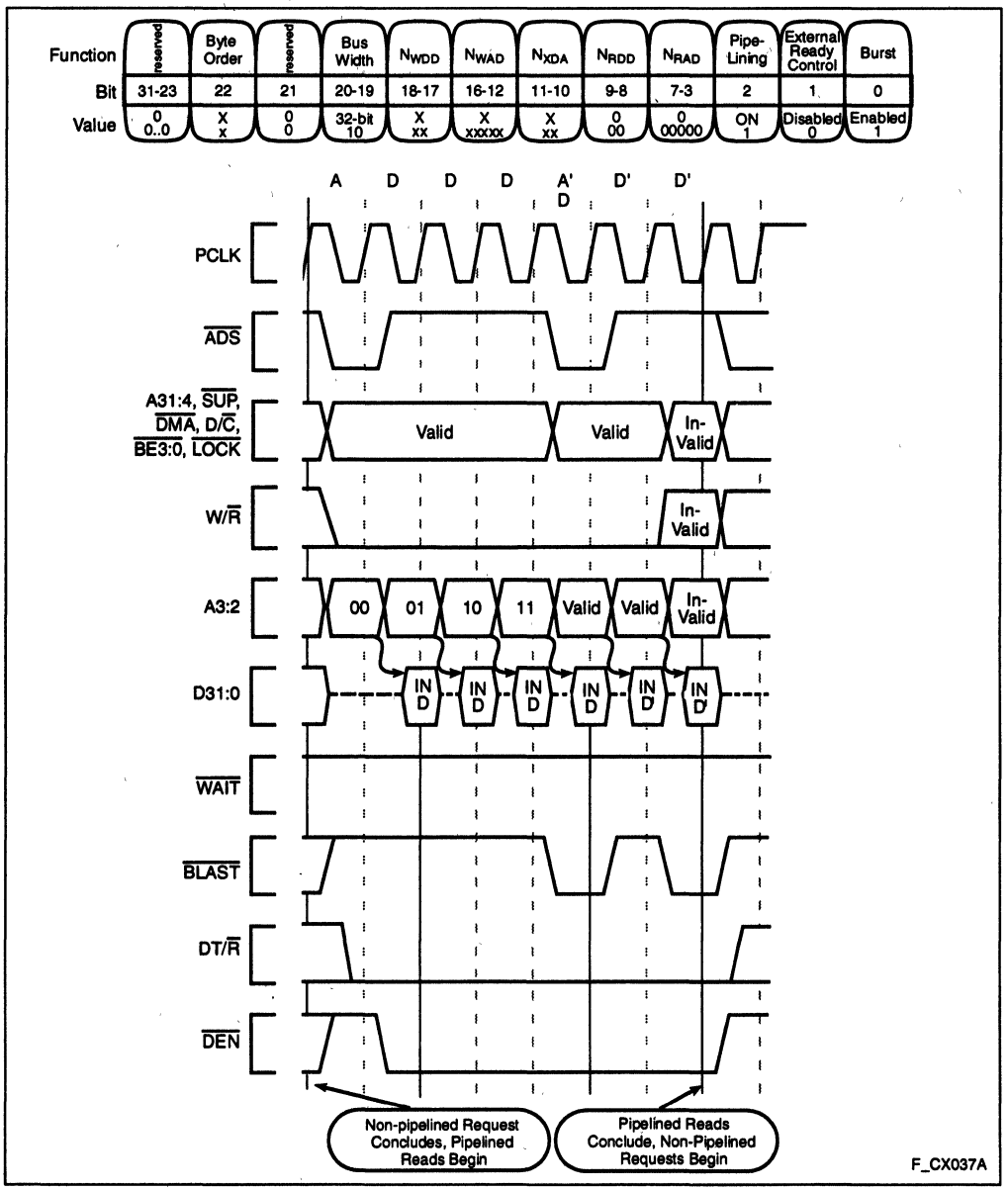


Figure 35. Burst, Pipelined Read Request Without Wait States, 32-Bit Bus

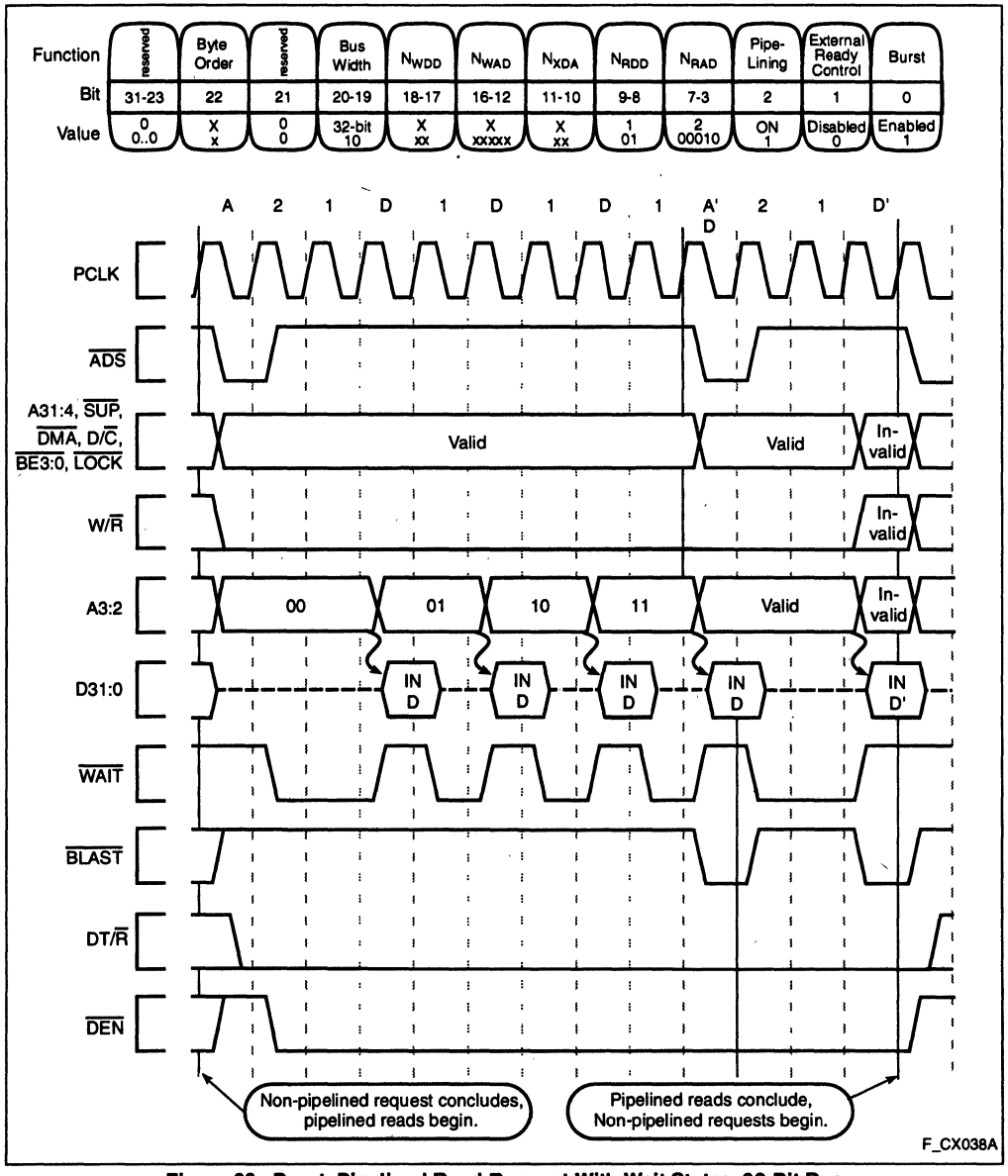


Figure 36. Burst, Pipelined Read Request With Wait States, 32-Bit Bus

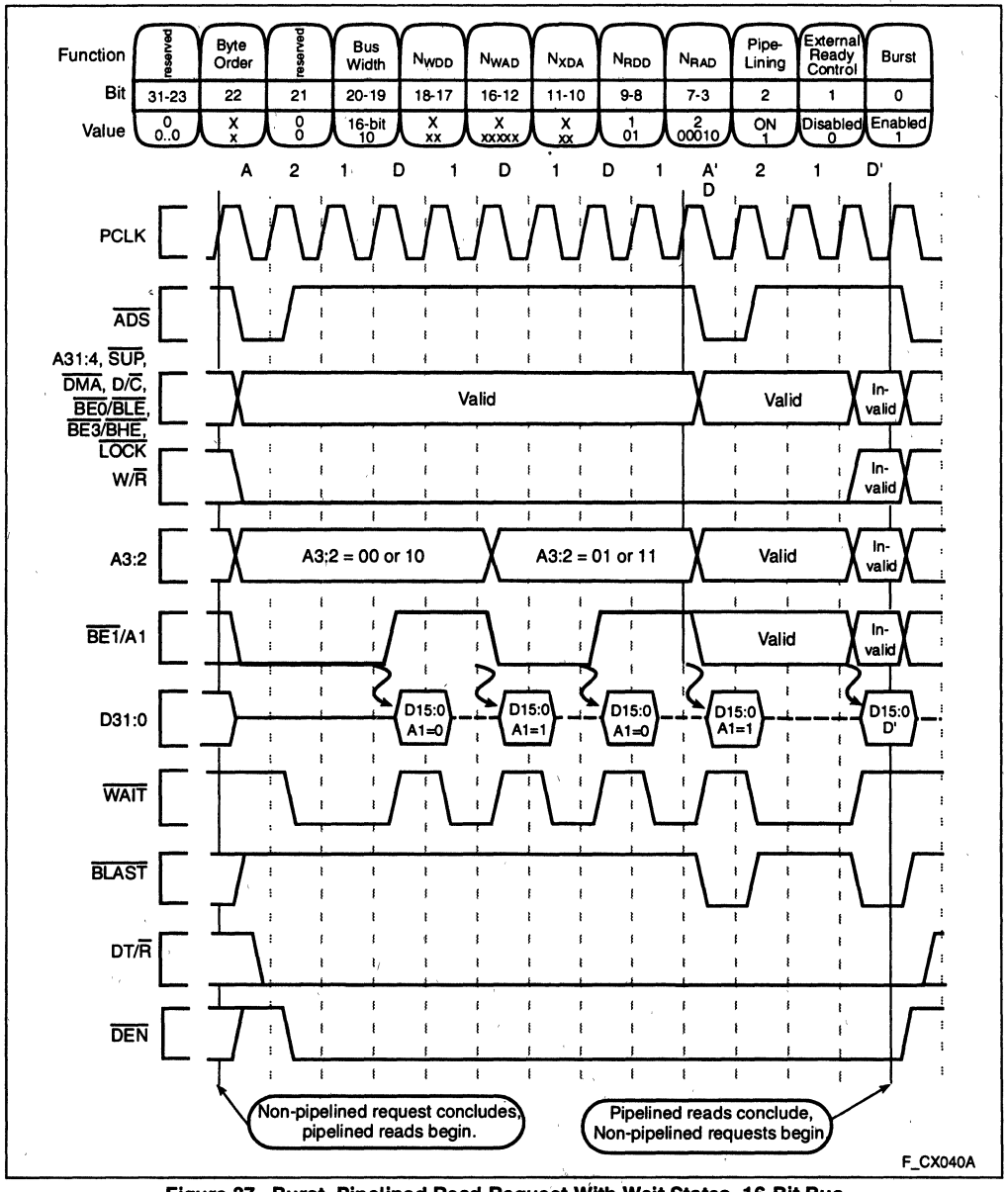


Figure 37. Burst, Pipelined Read Request With Wait States, 16-Bit Bus

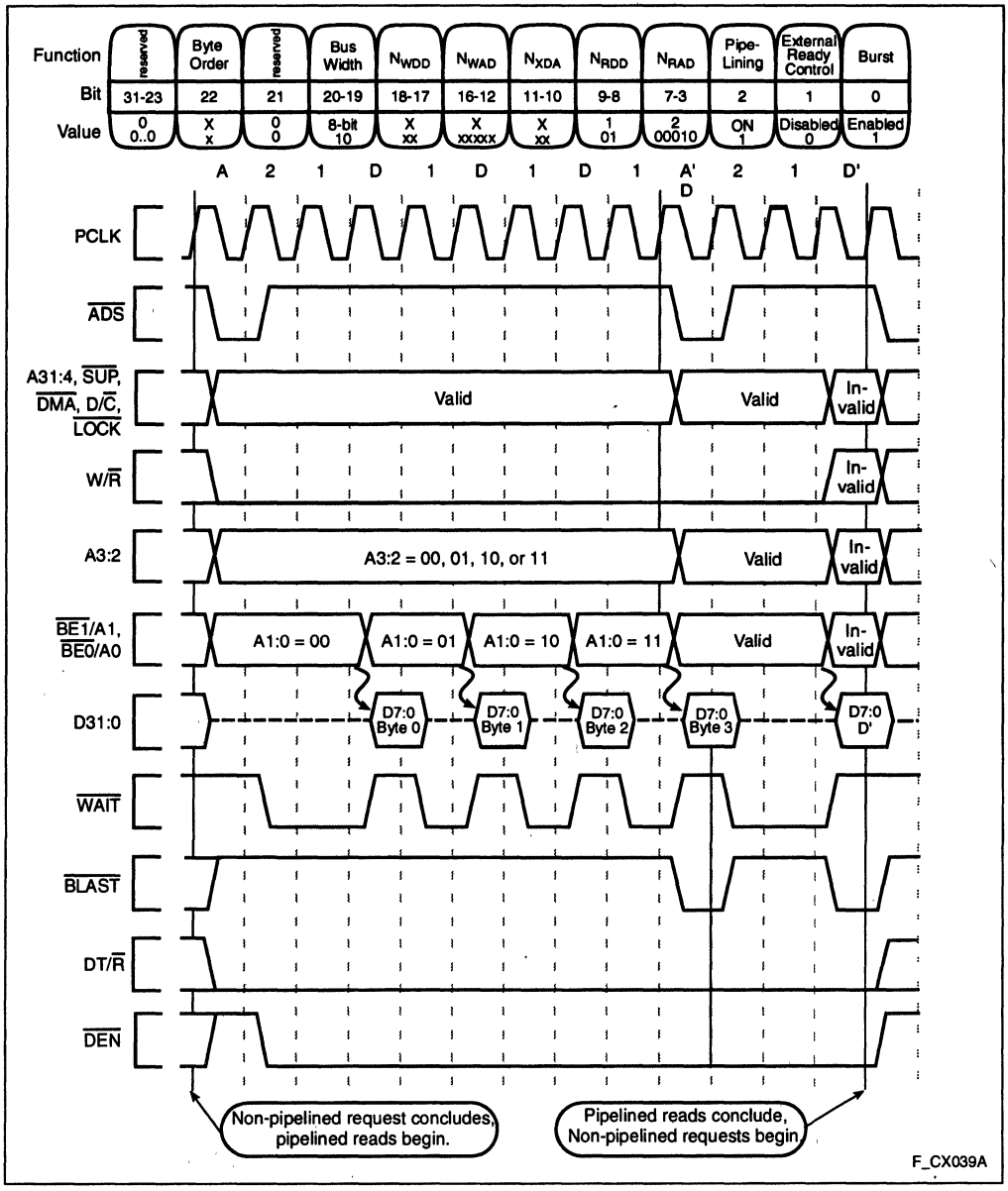


Figure 38. Burst, Pipelined Read Request With Wait States, 8-Bit Bus

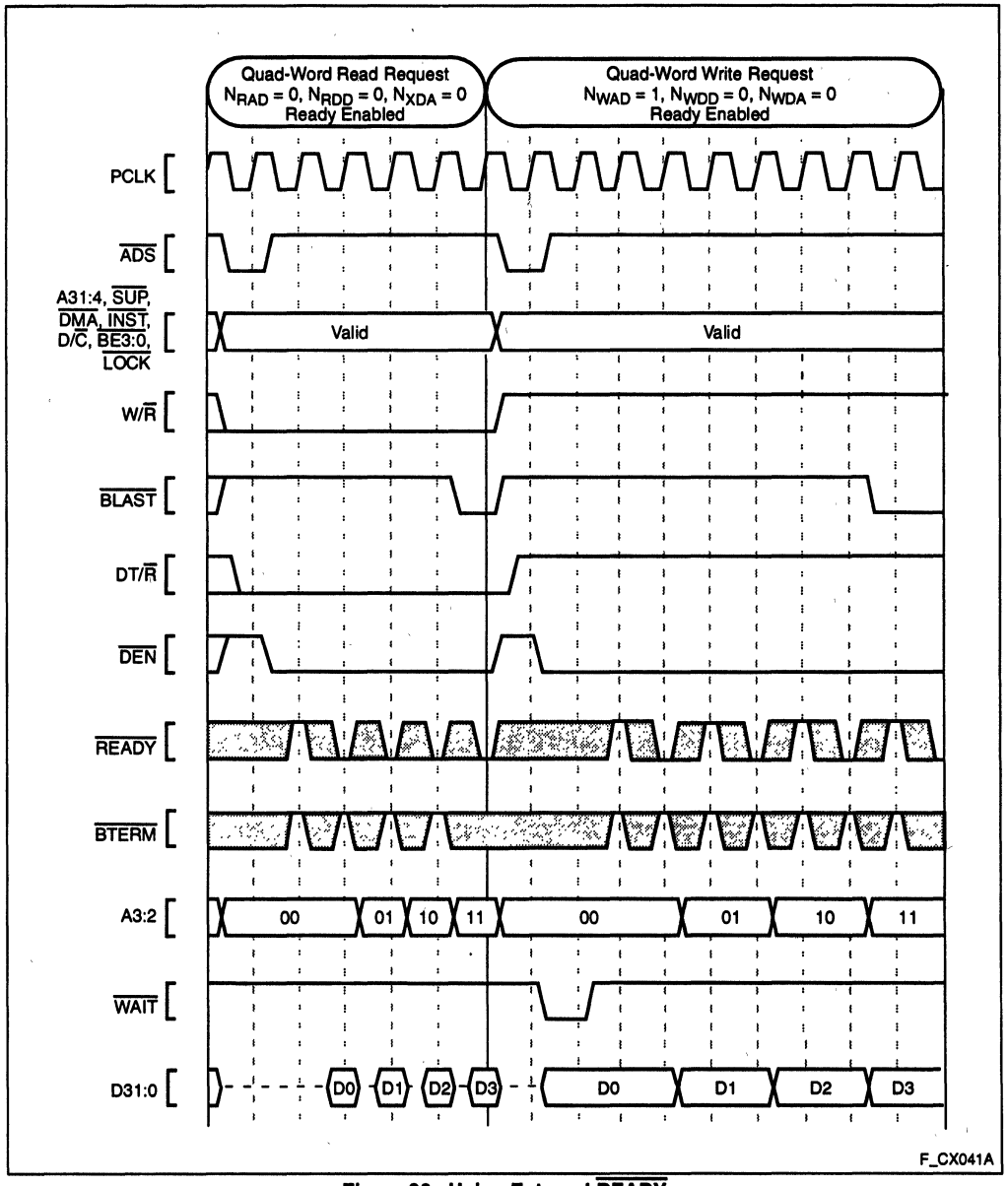
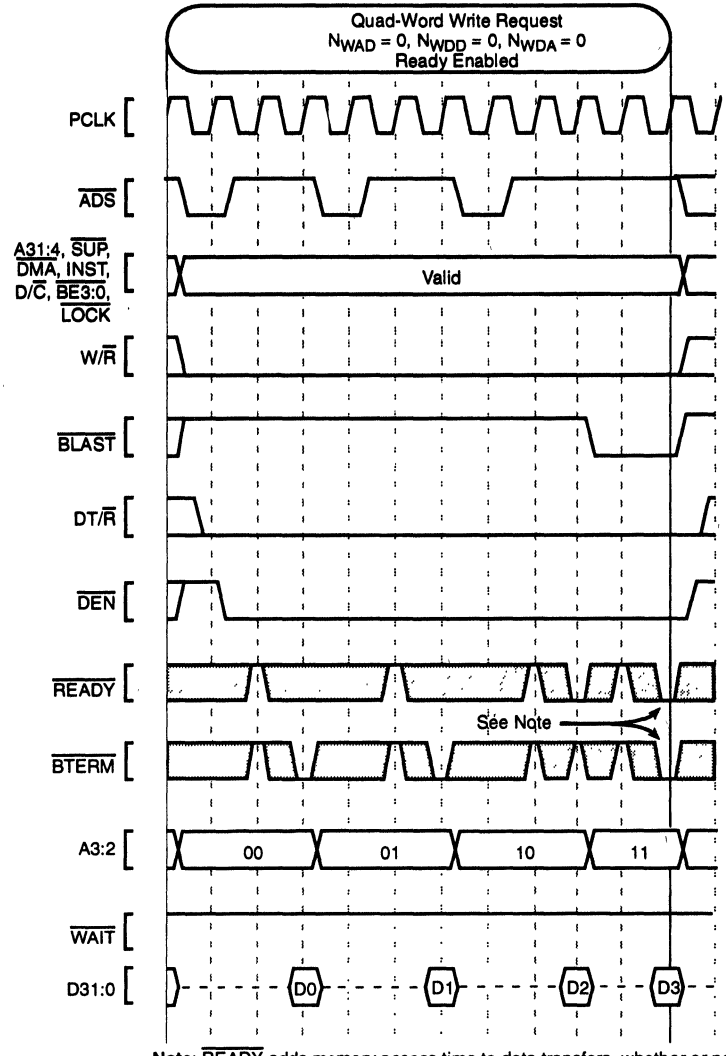


Figure 39. Using External READY



Note: READY adds memory access time to data transfers, whether or not the bus access is a burst access. BTERM interrupts a bus access, whether or not the bus access has more data transfers pending. Either the READY signal or the BTERM signal will terminate a bus access if the signal is asserted during the last (or only) data transfer of the bus access.

F_CX042A

Figure 40. Terminating a Burst with BTERM

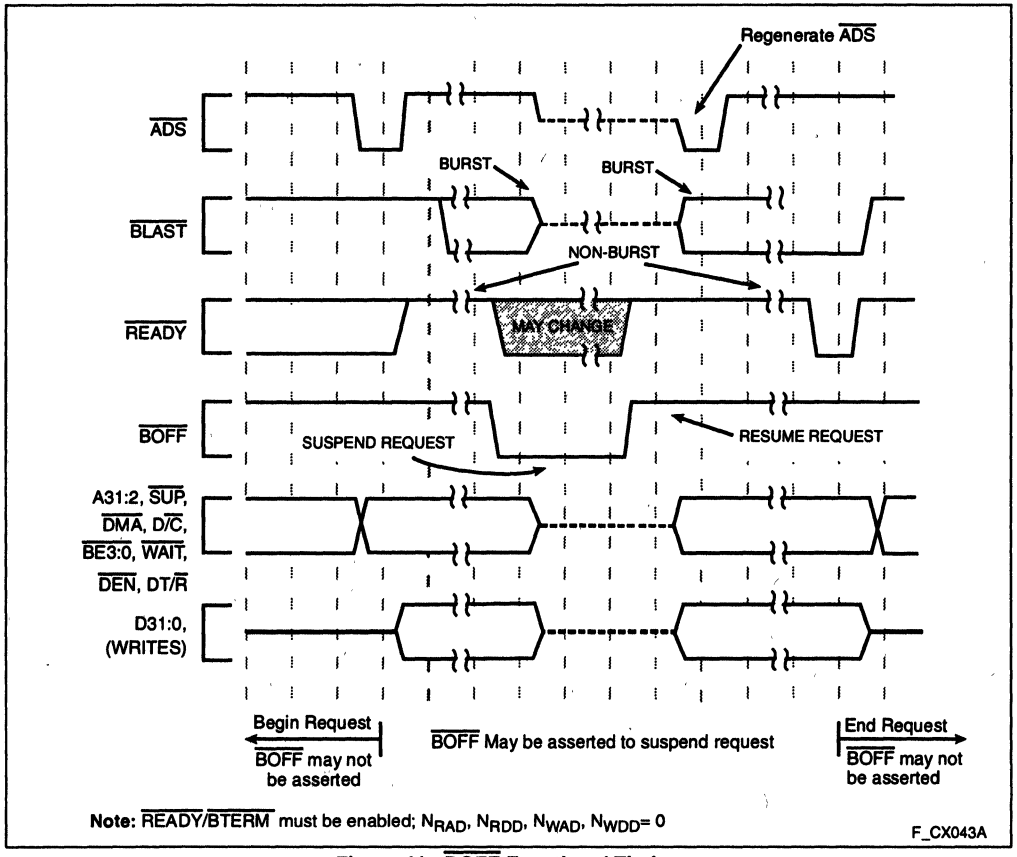


Figure 41. BOFF Functional Timing

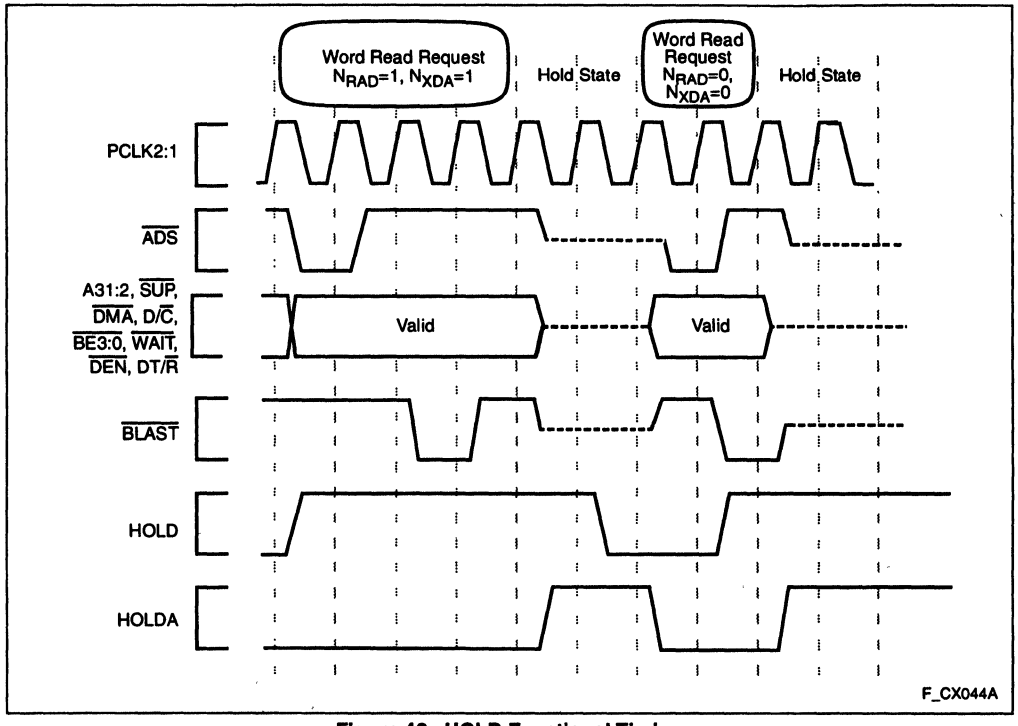
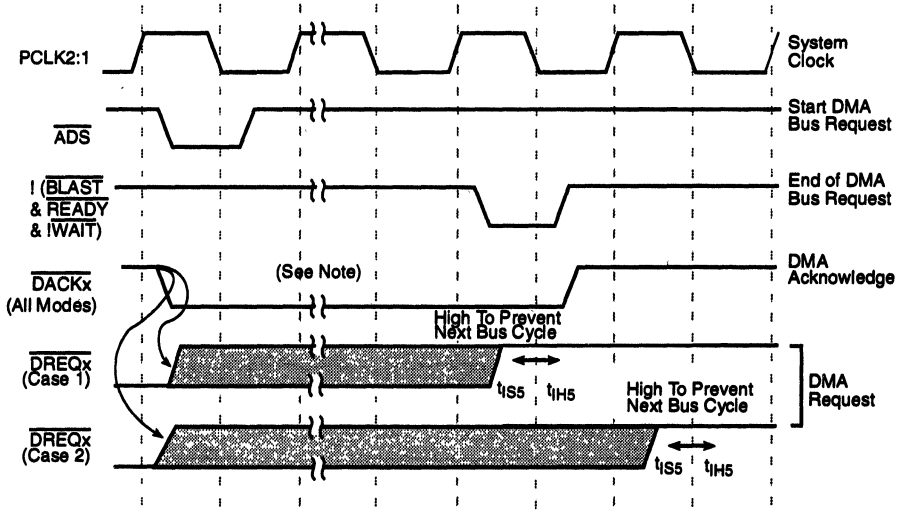


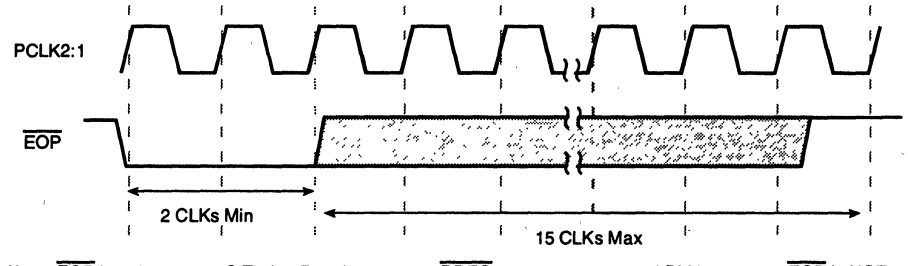
Figure 42. HOLD Functional Timing



Note:
1. Case 1: DREQ must deassert before DACK deasserts. Applications are Fly-By and some packing and unpacking modes in which loads are followed by loads or stores are followed by stores.
2. Case 2: DREQ must be deasserted by the second clock (rising edge) after DACK is driven high. Applications are non Fly-By transfers and adjacent load-stores or store-loads.
3. DACKx is asserted for the duration of a DMA bus request. The request may consist of multiple bus accesses (defined by ADS and BLAST. Refer to *i960® CA Microprocessor User's Manual* for "access", "request" definitions.

F_CX018A

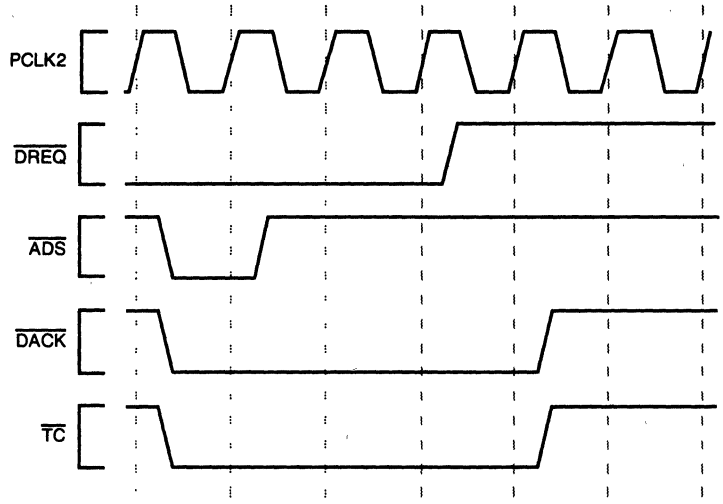
Figure 43. DREQ and DACK Functional Timing



Note: EOP has the same AC Timing Requirements as DREQ to prevent unwanted DMA requests. EOP is NOT edge triggered. EOP must be held for a minimum of 2 clock cycles then deasserted within 15 clock cycles.

F_CX045A

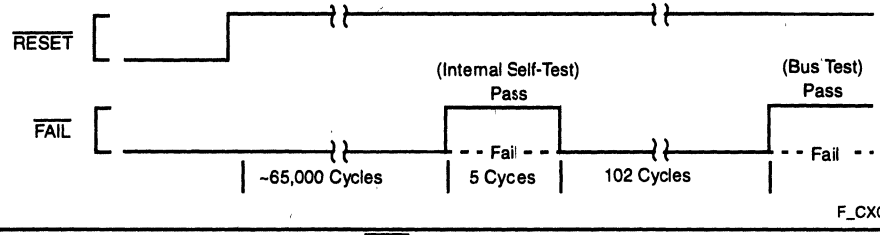
Figure 44. EOP Functional Timing



Note: Terminal Count becomes active during the last bus request of a buffer transfer. If the last LOAD/STORE bus request is executed as multiple bus accesses, the TC will be active for the entire bus request. Refer to the *i960® CA Microprocessor User's Manual* for further information.

F_CX046A

Figure 45. Terminal Count Functional Timing



F_CX047A

Figure 46. FAIL Functional Timing

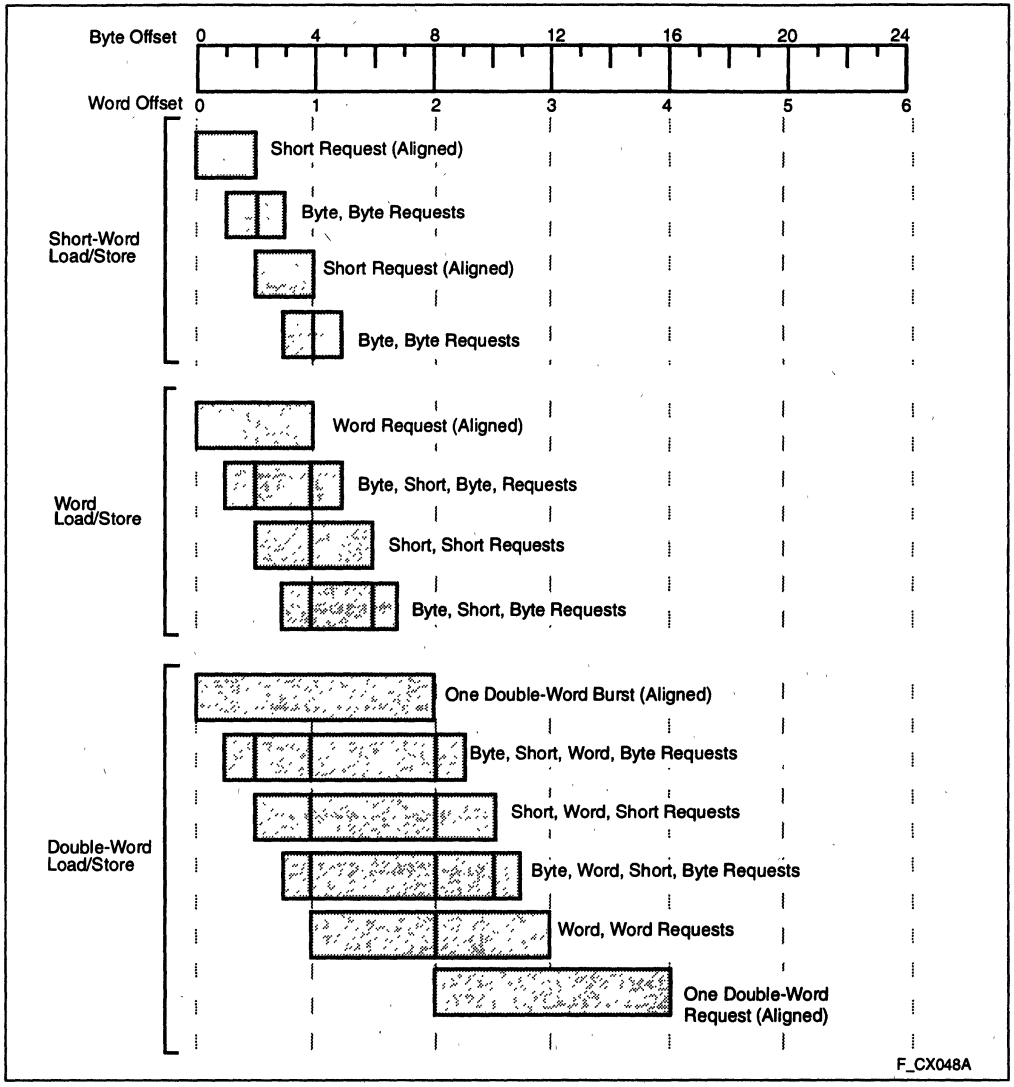


Figure 47. A Summary of Aligned and Unaligned Transfers for Little Endian Regions

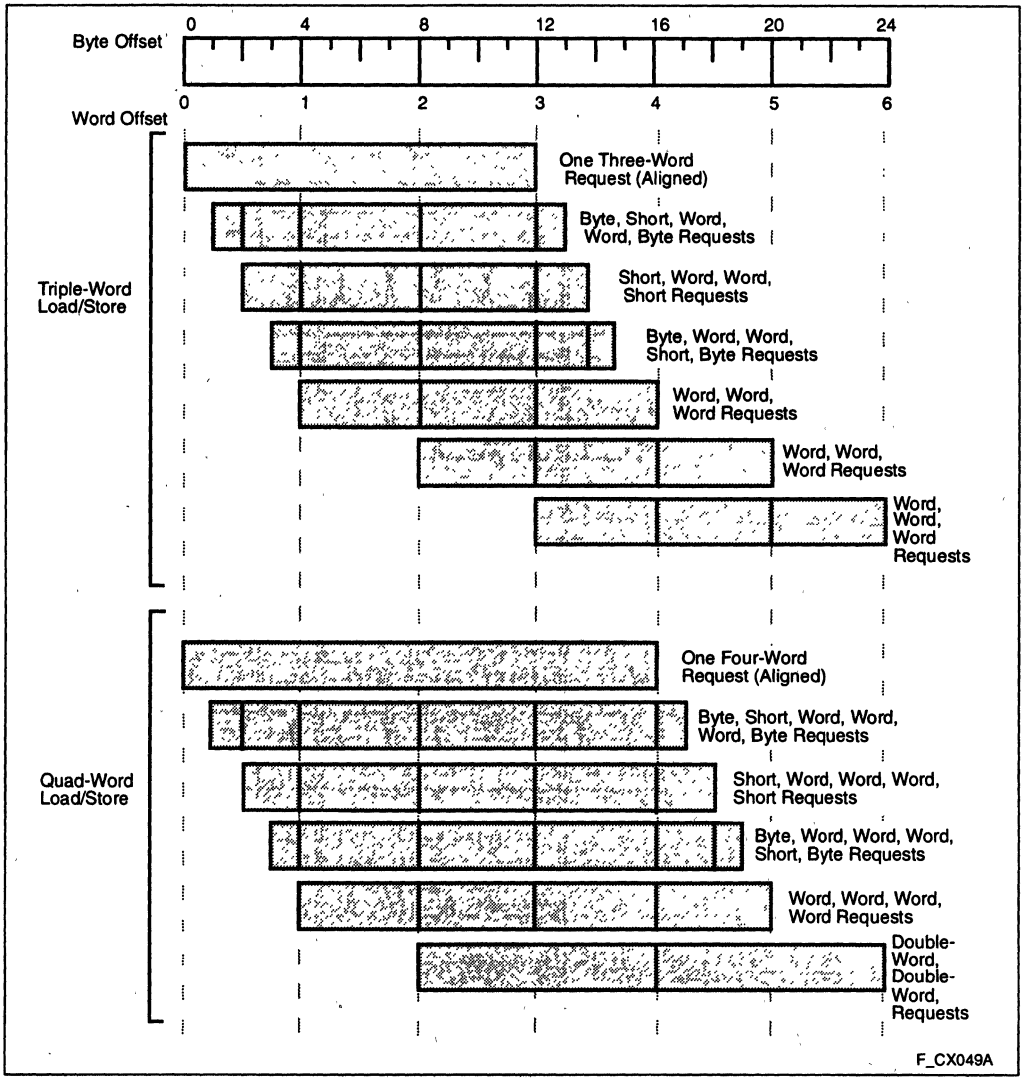


Figure 48. A Summary of Aligned and Unaligned Transfers for Little Endian Regions (Continued)

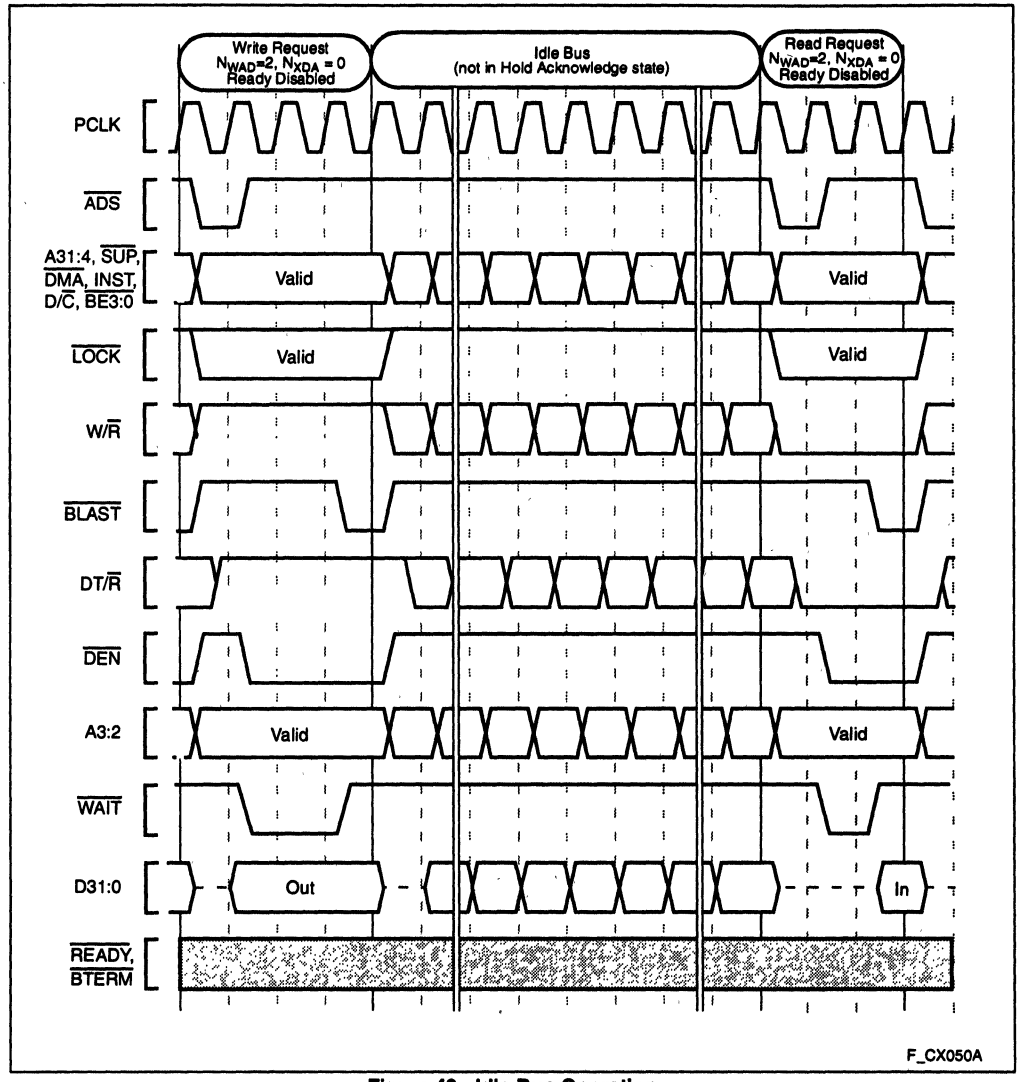


Figure 49. Idle Bus Operation

7.0 REVISION HISTORY

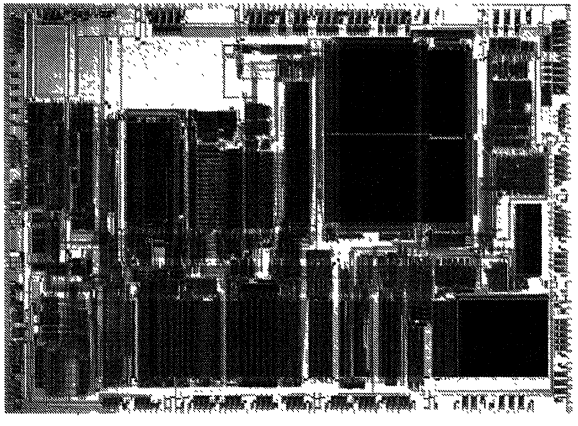
This data sheet supersedes data sheet 270727-005. Specification changes in the 80960CA data sheet are a result of design changes. The sections significantly changed since the previous revision are:

Section	Last Rev.	Description
Table 11. 80960CA PGA Package Thermal Characteristics	-005	Removed references and notes pertaining to $\theta_{J\text{-}CAP}$ and $\theta_{J\text{-}PIN}$.
Table 12. 80960CA PQFP package Thermal Characteristics	-005	Removed references and notes pertaining to θ_{JL} and θ_{JB}.
3.3 80960CA Mechanical Data	-005	Removed section containing information on Package Dimensions. Moved section header to encompass Pinout tables and diagrams.
3.7 Suggested Sources for 80960CA Accessories	-005	Removed entire section containing information about 80960CA accessories.
Tables 16, 17 and18 80960CA AC Characteristics (33-, 25- and 16MHz, respectively)	-005	T_{TVEL} maximum deleted. T_{NHQX} and T_{EHTV} minimums changed: WAS: IS: T_{NHQX} (N+1)*T-6 (N+1)*T-8 T_{EHTV} T/2 - 6 T/2 - 7
All	-005	All timing diagrams and waveforms have been redrawn to conform to consistent format. Data sheet formatting has been changed to conform to corporate standards. Specific formatting changes are not itemized in this revision history.

80960CF-33, -25, -16
32-BIT HIGH PERFORMANCE SUPERSCALAR PROCESSOR

- Socket and Object Code Compatible with 80960CA
- Two Instructions/Clock Sustained Execution
- Four 59 Mbytes/s DMA Channels with Data Chaining
- Demultiplexed 32-bit Burst Bus with Pipelining

- 32-bit Parallel Architecture
  - Two Instructions/clock Execution
  - Load/Store Architecture
  - Sixteen 32-bit Global Registers
  - Sixteen 32-bit Local Registers
  - Manipulate 64-bit Bit Fields
  - 11 Addressing Modes
  - Full Parallel Fault Model
  - Supervisor Protection Model
- Fast Procedure Call/Return Model
  - Full Procedure Call in 4 clocks
- On-Chip Register Cache
  - Caches Registers on Call/Ret
  - Minimum of 6 Frames provided
  - Up to 15 Programmable Frames
- On-Chip Instruction Cache
  - 4 Kbyte Two-Way Set Associative
  - 128-bit Path to Instruction Sequencer
  - Cache-Lock Modes
  - Cache-Off Mode
- On-Chip Data Cache
  - 1 Kbyte Direct-Mapped, Write Through
  - 128 bits per Clock Access on Cache Hit
- High Bandwidth On-Chip Data RAM
  - 1 Kbytes On-Chip RAM for Data
  - Sustain 128 bits per clock access
- Four On-Chip DMA Channels
  - 59 Mbytes/s Fly-by Transfers
  - 32 Mbytes/s Two-Cycle Transfers
  - Data Chaining
  - Data Packing/Unpacking
  - Programmable Priority Method
- 32-Bit Demultiplexed Burst Bus
  - 128-bit Internal Data Paths to *and* from Registers
  - Burst Bus for DRAM Interfacing
  - Address Pipelining Option
  - Fully Programmable Wait States
  - Supports 8, 16 or 32-bit Bus Widths
  - Supports Unaligned Accesses
  - Supervisor Protection Pin
- Selectable Big or Little Endian Byte Ordering
- High-Speed Interrupt Controller
  - Up to 248 External Interrupts
  - 32 Fully Programmable Priorities
  - Multi-mode 8-bit Interrupt Port
  - Four Internal DMA Interrupts
  - Separate, Non-maskable Interrupt Pin
  - Context Switch in 750 ns Typical



272187-59

Figure 1. 80960CF Die Photo

December 1992
Order Number: 272187-002

80960CF-33, -25, -16
32-Bit High Performance Superscalar Processor

CONTENTS

CONTENTS

CONTENTS

CONTENTS

1.0 PURPOSE

This document previews electrical characterizations of Intel's i960 CF embedded microprocessor (available in 33, 25 and 16 MHz). For a detailed description of any i960 CF processor functional topic—other than parametric performance—refer to the latest i960 CA Microprocessor Reference Manual (Order No. 270710) and the *i960 CF Reference Manual Addendum* (Order No. 272188).

2.0 i960 CF PROCESSOR OVERVIEW

Intel's i960 CF microprocessor is the performance follow-on product to the i960 CA processor. The i960 CF product is socket- and object code-compatible with the CA; this makes CA-to-CF design upgrades straightforward. The i960 CF processor's instruction cache is 4 Kbytes (CA device has 1 Kbyte); CF data cache is 1 Kbyte (CA device has no data cache). This extra cache on the CF product adds a significant performance boost over the CA. The 80960CF is object code compatible with the 32-bit 80960 Core Architecture while including Special Function Register extensions to control on-chip peripherals, and instruction set extensions to shift 64-bit operands and configure on-chip hardware. Multiple 128-bit internal busses, on-chip instruction caching and a sophisticated instruction scheduler allow the processor to sustain execution of two instructions every clock, and peak at execution of three instructions per clock.

A 32-bit demultiplexed and pipelined burst bus provides a 132 Mbyte/s bandwidth to a system's high-speed external memory sub-system. In addition, the 80960CF's on-chip caching of instructions, procedure context and critical program data substantially decouples system performance from the wait states associated with accesses to the system's slower, cost sensitive, main memory sub-system.

The 80960CF bus controller also integrates full wait state and bus width control for highest system performance with minimal system design complexity. Unaligned access and Big Endian byte order support reduces the cost of porting existing applications to the 80960CF.

The processor also integrates four complete data-chaining DMA channels and a high-speed interrupt controller on-chip. The DMA channels perform: single-cycle or two-cycle transfers, data packing and unpacking, and data chaining. Block transfers, in addition to source or destination synchronized transfers, are provided.

The interrupt controller provides full programmability of 248 interrupt sources into 32 priority levels with a typical interrupt task switch ("latency") time of 750 ns.

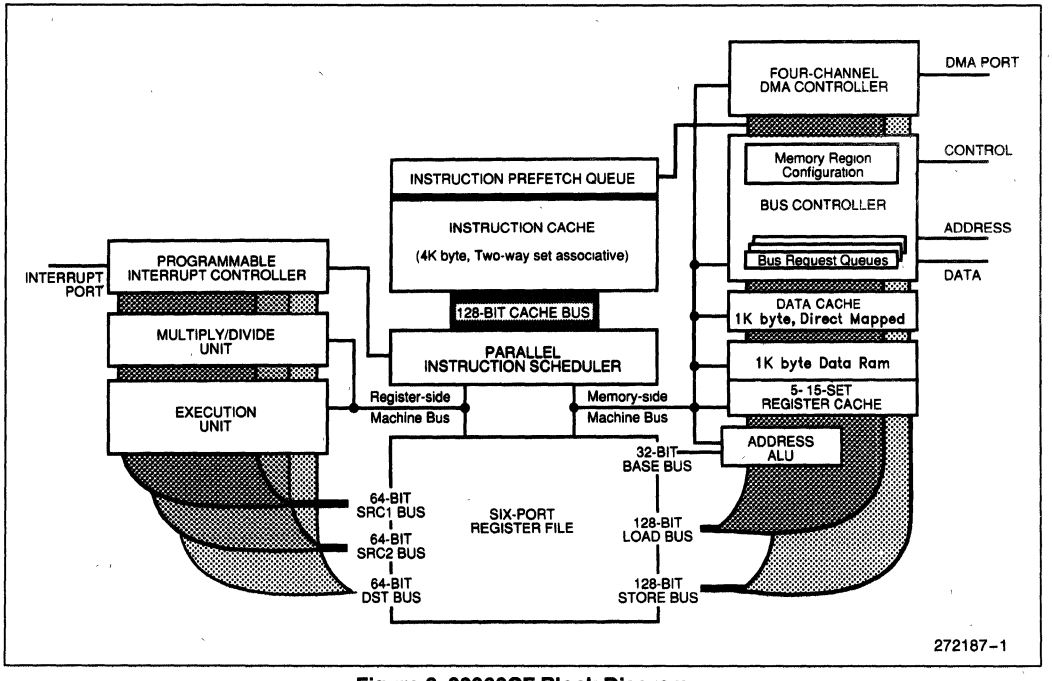


Figure 2. 80960CF Block Diagram

2.1. The C-Series Core

The C-Series core is a very high performance microarchitectural implementation of the 80960 Core Architecture. The C-Series core can sustain execution of two instructions per clock (66 MIPs at 33 MHz). To achieve this level of performance, Intel has incorporated state-of-the-art silicon technology and innovative microarchitectural constructs into the implementation of the C-Series core. Factors that contribute to the core's performance include:

— Parallel instruction decoding allows issue of up to three instructions per clock.

— Most instructions execute in a single clock.

— Parallel instruction decode allows sustained, simultaneous execution of two single-clock instructions every clock cycle.

— Efficient instruction pipeline minimizes pipeline break losses.

— Register and resource scoreboarding allow simultaneous multi-clock instruction execution.

— Branch look-ahead and prediction allows many branches to execute with no pipeline break.

— Local Register Cache integrated on-chip caches Call/Return context.

— Two-way set associative, 4 Kbyte integrated instruction cache.

— Direct mapped, 1 Kbyte data cache, write through, write allocate.

— 1 Kbyte integrated Data RAM sustains a four-word (128-bit) access every clock cycle.

2.2. Pipelined, Burst Bus

A 32-bit high performance bus controller interfaces the 80960CF to external memory and peripherals. The Bus Control Unit features a maximum transfer rate of 132 Mbytes per second (at 33 MHz). Internally programmable wait states and 16 separately configurable memory regions allow the processor to interface with a variety of memory subsystems with a minimum of system complexity and a maximum of performance. The Bus Controller's main features include:

— Demultiplexed, Burst Bus to exploit most efficient DRAM access modes.

— Address Pipelining to reduce memory cost while maintaining performance.

— 32-, 16- and 8-bit modes for I/O interfacing ease.

— Full internal wait state generation to reduce system cost.

— Little and Big Endian support to ease application development.

— Unaligned access support for code portability.

— Three-deep request queue to decouple the bus from the core.

2.3. Flexible DMA Controller

A four channel DMA controller provides high speed DMA control for data transfers involving peripherals and memory. The DMA provides advanced features such as data chaining, byte assembly and disassembly, and a high performance fly-by mode capable of transfer speed of up to 59 Mbytes per second at 33 MHz. The DMA controller features a performance and flexibility which is only possible by integrating the DMA controller and the 80960CF core.

2.4. Priority Interrupt Controller

A programmable-priority interrupt controller manages up to 248 external sources through the 8-bit external interrupt port. The Interrupt Unit also handles the four internal sources from the DMA controller, and a single non-maskable interrupt input. The 8-bit interrupt port can also be configured to provide individual interrupt sources that are level or edge triggered.

Interrupts in the 80960CF are prioritized and signaled within 270 ns of the request. If the interrupt is of higher priority than the processor priority, the context switch to the interrupt routine typically is complete in another 480 ns. The interrupt unit provides the mechanism for the low latency and high throughput interrupt service which is essential for embedded applications.

2.5. Instruction Set Summary

The following table summarizes the 80960CF instruction set by logical groupings. See the *i960 CA Microprocessor Reference Manual* for a complete description of the instruction set.

Data Movement	Arithmetic	Logical	Bit, Bit Field and Byte
Load Store Move Load Address	Add Subtract Multiply Divide Remainder Modulo Shift *Extended Shift Extended Multiply Extended Divide Add with Carry Subtract with Carry Rotate	And Not And And Not Or Exclusive Or Not Or Or Not Nor Exclusive Nor Not Nand	Set Bit Clear Bit Not Bit Alter Bit Scan for Bit Span over Bit Extract Modify Scan Byte for Equal
Comparison	**Branch**	**Call and Return**	**Fault**
Compare Conditional Compare Compare and Increment Compare and Decrement Test Condition Code Check Bit	Unconditional Branch Conditional Branch Compare and Branch	Call Call Extended Call System Return Branch and Link	Conditional Fault Synchronize Faults
Debug	**Processor Management**	**Atomic**	
Modify Trace Controls Mark Force Mark	Modify Process Controls Modify Arithmetic Controls *System Control *DMA Control Flush Local Registers	Atomic Add Atomic Modify	

NOTE:
Instructions marked by (*) are 80960CF extensions to the 80960 instruction set.

3.0 PACKAGE INFORMATION

3.1. Package Introduction

This section describes the pins, pinouts and thermal characteristics for the 80960CF in the 168-pin Ceramic Pin Grid Array (PGA) package and the 196 pin Plastic Quad Flat Package (PQFP). For complete package specifications and information, see the Intel *Packaging Outlines and Dimensions Guide* (Order No. 231369).

3.2. Pin Descriptions

The 80960CF pins are described in this section. Table 1 presents the legend for interpreting the pin descriptions in the following tables.

Pins associated with the 32-bit demultiplexed processor bus are described in Table 2. Pins associated with basic processor configuration and control are described in Table 3. Pins associated with the 80960CF DMA Controller and Interrupt Unit are described in Table 4.

Figure 3 provides an example pin description table entry. "I/O" signifies that data pins are input-output. "S" indicates pins are synchronous to PCLK2:1. "H(Z)" indicates that these pins float while the processor bus is in a Hold Acknowledge state. "R(Z)" indicates that the pins also float while $\overline{\text{RESET}}$ is low.

All pins float while the processor is in the ONCE mode.

Table 1. Pin Description Nomenclature

Symbol	Description
I	Input only pin
O	Output only pin
I/O	Pin can be either an input or output
-	Pins "must be" connected as described
S(. . .)	Synchronous. Inputs must meet setup and hold times relative to PCLK2:1 for proper operation. All outputs are synchronous to PCLK2:1. S(E) Edge sensitive input S(L) Level sensitive input
A(. . .)	Asynchronous. Inputs may be asynchronous to PCLK2:1. A(E) Edge sensitive input A(L) Level sensitive input
H(. . .)	While the processor's bus is in the Hold Acknowledge or Bus Backoff state, the pin: H(1) is driven to V_{CC} H(0) is driven to V_{SS} H(Z) floats H(Q) continues to be a valid output
R(. . .)	While the processor's $\overline{\text{RESET}}$ pin is low, the pin R(1) is driven to V_{CC} R(0) is driven to V_{SS} R(Z) floats R(Q) continues to be a valid output

Name	Type	Description
D31:0	**I/O** S(L) H(Z) R(Z)	**DATA BUS** carries 32-, 16- or 8-bit data quantities depending on bus width configuration. The least significant bit of the data is carried on D0 and the most significant on D31. When the bus is configured for 8-bit data, the lower 8 data lines, D7:0 are used. For 16-bit bus widths, D15:0 are used. For 32-bit bus widths the full data bus is used.

Figure 3. Example Pin Description Entry

Table 2. 80960CF Pin Description—External Bus Signals

Name	Type	Description
A31:2	O S H(Z) R(Z)	**ADDRESS BUS** carries the physical address upper 30 bits. A31 is the most significant address bit and A2 is the least significant. During a bus access, A31:2 identify all external addresses to word (4-byte) boundaries. The byte enable signals indicate the selected byte in each word. During burst accesses, A3 and A2 increment to indicate successive data cycles.
D31:0	I / O S(L) H(Z) R(Z)	**DATA BUS** carries 32-, 16- or 8-bit data quantities depending on bus width configuration. The least significant bit of the data is carried on D0 and the most significant on D31. When the bus is configured for 8-bit data, the lower 8 data lines, D7:0 are used. For 16-bit bus widths, D15:0 are used. For 32-bit bus widths the full data bus is used.
$\overline{BE3}$ **$\overline{BE2}$** **$\overline{BE1}$** **$\overline{BE0}$**	O S H(Z) R(1)	**BYTE ENABLES** select which of the four bytes addressed by A31:2 are active during an access to a memory region configured for a 32-bit data-bus width. $\overline{BE3}$ applies to D31:24; $\overline{BE2}$ applies to D23:16; $\overline{BE1}$ applies to D15:8; and $\overline{BE0}$ applies to D7:0. 32-bit bus: $\overline{BE3}$ –Byte Enable 3 –enable D31:24 $\overline{BE2}$ –Byte Enable 2 –enable D23:16 $\overline{BE1}$ –Byte Enable 1 –enable D15:8 $\overline{BE0}$ –Byte Enable 0 –enable D7:0 For accesses to a memory region configured for a 16-bit data-bus width, the processor directly encodes $\overline{BE3}$, $\overline{BE1}$ and $\overline{BE0}$ to provided $\overline{BHE}$, A1 and $\overline{BLE}$ respectively. 16-bit bus: $\overline{BE3}$ –Byte High Enable ($\overline{BHE}$) –enable D15:8 $\overline{BE2}$ –Not used (is driven high or low) $\overline{BE1}$ –Address Bit 1 (A1) $\overline{BE0}$ –Byte Low Enable ($\overline{BLE}$) –enable D7:0 For accesses to a memory region configured for an 8-bit data bus width, the processor directly encodes $\overline{BE1}$ and $\overline{BE0}$ to provide A1 and A0 respectively. 8-bit bus: $\overline{BE3}$ –Not used (is driven high or low) $\overline{BE2}$ –Not used (is driven high or low) $\overline{BE1}$ –Address Bit 1 (A1) $\overline{BE0}$ –Address Bit 0 (A0)
W/$\overline{R}$	O S H(Z) R(0)	**WRITE/READ** is asserted for read requests and deasserted for write requests. The W/$\overline{R}$ signal changes in the same clock cycle as $\overline{ADS}$. It remains valid for the entire access in non-pipelined regions. In pipelined regions, W/$\overline{R}$ is not guaranteed valid in the last cycle of a read access.
$\overline{ADS}$	O S H(Z) R(1)	**ADDRESS STROBE** indicates valid address and the start of a new bus access. $\overline{ADS}$ is asserted for the first clock of a bus access.
$\overline{READY}$	I S(L) H(Z) R(Z)	**READY** is an input which signals the termination of a data transfer. $\overline{READY}$ is used to indicate that read data on the bus is valid, or that a write-data transfer has completed. The $\overline{READY}$ signal works in conjunction with the internally programmed wait-state generator. If $\overline{READY}$ is enabled in a region, the pin is sampled after the programmed number of wait-states has expired. If the $\overline{READY}$ pin is deasserted, wait states continue to be inserted until $\overline{READY}$ becomes asserted. This is true for the N_{RAD}, N_{RDD}, N_{WAD}, and N_{WDD} wait states. The N_{XDA} wait states cannot be extended.

Table 2. 80960CF Pin Description—External Bus Signals (Continued)

Name	Type	Description
$\overline{\text{BTERM}}$	I S(L) H(Z) R(Z)	**BURST TERMINATE**—The burst terminate signal breaks up a burst access and causes another address cycle to occur. The $\overline{\text{BTERM}}$ signal works in conjunction with the internally programmed wait-state generator. If $\overline{\text{READY}}$ and $\overline{\text{BTERM}}$ are enabled in a region, the $\overline{\text{BTERM}}$ pin is sampled after the programmed number of wait states has expired. When $\overline{\text{BTERM}}$ is asserted, a new $\overline{\text{ADS}}$ signal is generated and the access is completed. The $\overline{\text{READY}}$ input is ignored when $\overline{\text{BTERM}}$ is asserted. $\overline{\text{BTERM}}$ must be externally synchronized to satisfy the $\overline{\text{BTERM}}$ setup and hold times.
$\overline{\text{WAIT}}$	O S H(Z) R(1)	**WAIT** indicates internal wait state generator status. $\overline{\text{WAIT}}$ is asserted when wait states are being caused by the internal wait state generator and not by the $\overline{\text{READY}}$ or $\overline{\text{BTERM}}$ inputs. $\overline{\text{WAIT}}$ can be used to derive a write-data strobe. $\overline{\text{WAIT}}$ can also be thought of as a $\overline{\text{READY}}$ output that the processor provides when it is inserting wait states.
$\overline{\text{BLAST}}$	O S H(Z) R(0)	**BURST LAST** indicates the last transfer in a bus access. $\overline{\text{BLAST}}$ is asserted in the last data transfer of burst and non-burst accesses after the wait state counter reaches zero. $\overline{\text{BLAST}}$ remains asserted until the clock following the last cycle of the last data transfer of a bus access. If the $\overline{\text{READY}}$ or $\overline{\text{BTERM}}$ input is used to extend wait states, the $\overline{\text{BLAST}}$ signal remains asserted until $\overline{\text{READY}}$ or $\overline{\text{BTERM}}$ terminates the access.
DT/$\overline{\text{R}}$	O S H(Z) R(0)	**DATA TRANSMIT/RECEIVE** indicates direction for data transceivers. DT/$\overline{\text{R}}$ is used in conjunction with $\overline{\text{DEN}}$ to provide control for data transceivers attached to the external bus. When DT/$\overline{\text{R}}$ is asserted, the signal indicates that the processor receives data. Conversely, when deasserted, the processor sends data. DT/$\overline{\text{R}}$ changes only while $\overline{\text{DEN}}$ is high.
$\overline{\text{DEN}}$	O S H(Z) R(1)	**DATA ENABLE** indicates data cycles in a bus request. $\overline{\text{DEN}}$ is asserted at the start of the bus request first data cycle and is deasserted at the end of the last data cycle. $\overline{\text{DEN}}$ is used in conjunction with DT/$\overline{\text{R}}$ to provide control for data transceivers attached to the external bus. $\overline{\text{DEN}}$ remains asserted for sequential reads from pipelined memory regions. $\overline{\text{DEN}}$ is deasserted when DT/$\overline{\text{R}}$ changes.
$\overline{\text{LOCK}}$	O S H(Z) R(1)	**BUS LOCK** indicates that an atomic read-modify-write operation is in progress. $\overline{\text{LOCK}}$ may be used to prevent external agents from accessing memory which is currently involved in an atomic operation. $\overline{\text{LOCK}}$ is asserted in the first clock of an atomic operation, and deasserted in the clock cycle following the last bus access for the atomic operation. To allow the most flexibility for a memory system enforcement of locked accesses, the processor acknowledges a bus hold request when $\overline{\text{LOCK}}$ is asserted. The processor performs DMA transfers while $\overline{\text{LOCK}}$ is active.
HOLD	I S(L) H(Z) R(Z)	**HOLD REQUEST** signals that an external agent requests access to the external bus. The processor asserts HOLDA after completing the current bus request. HOLD, HOLDA and BREQ are used together to arbitrate access to the processor's external bus by external bus agents.
$\overline{\text{BOFF}}$	I S(L) H(Z) R(Z)	**BUS BACKOFF** —The backoff pin, when asserted, suspends the current access and causes the bus pins to float. When deasserted, the $\overline{\text{ADS}}$ signal is asserted on the next clock cycle and the access is resumed.

Table 2. 80960CF Pin Description—External Bus Signals (Continued)

Name	Type	Description
HOLDA	O S H(1) R(Q)	**HOLD ACKNOWLEDGE** indicates to a bus requestor that the processor has relinquished control of the external bus. When HOLDA is asserted, the external address bus, data bus and bus control signals are floated. HOLD, $\overline{BOFF}$, HOLDA and BREQ are used together to arbitrate access to the processor's external bus by external bus agents. Since the processor grants HOLD requests and enters the Hold Acknowledge state even while $\overline{RESET}$ is asserted, HOLDA pin state is independent of the $\overline{RESET}$ pin.
BREQ	O S H(Q) R(0)	**BUS REQUEST** is asserted when the bus controller has a request pending. BREQ can be used by external bus arbitration logic in conjunction with HOLD and HOLDA to determine when to return mastership of the external bus to the processor.
D/$\overline{C}$	O S H(Z) R(Z)	**DATA OR CODE** is asserted for a data request and deasserted for instruction requests. D/$\overline{C}$ has the same timing as W/$\overline{R}$.
$\overline{DMA}$	O S H(Z) R(Z)	**DMA ACCESS** indicates whether the bus request was initiated by the DMA controller. $\overline{DMA}$ is asserted for any DMA request. $\overline{DMA}$ is deasserted for all other requests.
$\overline{SUP}$	O S H(Z) R(Z)	**SUPERVISOR ACCESS** indicates whether the bus request is issued while in supervisor mode. $\overline{SUP}$ is asserted when the request has supervisor privileges, and is deasserted otherwise. $\overline{SUP}$ can be used to isolate supervisor code and data structures from non-supervisor requests.

Table 3. 80960CF Pin Description—Processor Control Signals

Name	Type	Description
$\overline{RESET}$	I A(L) H(Z) R(Z) N(Z)	**$\overline{RESET}$** causes the chip to reset. When $\overline{RESET}$ is asserted, all external signals return to the reset state. When $\overline{RESET}$ is deasserted, initialization begins. When the 2-x clock mode is selected, $\overline{RESET}$ must remain asserted for 16 PCLK2:1 cycles before being deasserted in order to guarantee correct processor initialization. When the 1-x clock mode is selected, $\overline{RESET}$ must remain asserted for 10,000 PCLK2:1 cycles before being deasserted in order to guarantee correct initialization. The CLKMODE pin selects 1-x or 2-x input clock division of the CLKIN pin. The processor's Hold Acknowledge bus state functions while the chip is reset. If the processor's bus is in the Hold Acknowledge state when $\overline{RESET}$ is asserted, the processor will internally reset, but maintains the Hold Acknowledge state on external pins until the Hold request is removed. If a hold request is made while the processor is in the reset state, the processor bus grants HOLDA and enters the Hold Acknowledge state.
$\overline{FAIL}$	O S H(Q) R(0)	**FAIL** indicates failure of the processor's self-test performed at initialization. When $\overline{RESET}$ is deasserted and the processor begins initialization, the $\overline{FAIL}$ pin is asserted. An internal self-test is performed as part of the initialization process. If this self-test passes, the $\overline{FAIL}$ pin is deasserted otherwise it remains asserted. The $\overline{FAIL}$ pin is reasserted while the processor performs an external bus self-confidence test. If this self-test passes, the processor deasserts the $\overline{FAIL}$ pin and branches to the user's initialization routine; otherwise the $\overline{FAIL}$ pin remains asserted. Internal self-test and the use of the $\overline{FAIL}$ pin can be disabled with the STEST pin.

Table 3. 80960CF Pin Description—Processor Control Signals (Continued)

Name	Type	Description
STEST	I S(L) H(Z) R(Z)	**SELF TEST** causes the processor's internal self-test feature to be enabled or disabled at initialization. STEST is read on the rising edge of $\overline{\text{RESET}}$. When asserted, the processor's internal self-test and external bus confidence tests are performed during processor initialization. When deasserted, only the external bus confidence tests are performed during initialization.
$\overline{\text{ONCE}}$	I A(L) H(Z) R(Z)	**ON CIRCUIT EMULATION** causes all outputs to be floated when asserted. $\overline{\text{ONCE}}$ is continuously sampled while $\overline{\text{RESET}}$ is low, and is latched on the rising edge of $\overline{\text{RESET}}$. To place the processor in the ONCE state: (1) assert $\overline{\text{RESET}}$ and $\overline{\text{ONCE}}$ (order does not matter) (2) wait for at least 16 CLKIN periods in 2-x mode, or 10,000 CLKIN periods in 1-x mode, after V_{CC} and CLKIN are within operating specifications (3) deassert $\overline{\text{RESET}}$ (4) wait at least 32 CLKIN periods (The processor is now latched in the ONCE state as long as $\overline{\text{RESET}}$ is high.) To exit the ONCE state, bring V_{CC} and CLKIN to operating conditions, then assert $\overline{\text{RESET}}$ and bring $\overline{\text{ONCE}}$ high prior to deasserting $\overline{\text{RESET}}$. CLKIN must operate within the specified operating conditions of the processor until step 4 above is completed. The CLKIN may then be changed to DC to achieve the lowest possible ONCE mode leakage current. $\overline{\text{ONCE}}$ can be used by emulator products or for board testers to effectively make an installed processor transparent in the board.
CLKIN	I A(E) H(Z) R(Z)	**CLOCK INPUT** is an input for the external clock needed to run the processor. The external clock is internally divided as prescribed by the CLKMODE pin to produce PCLK2:1.
CLKMODE	I A(L) H(Z) R(Z)	**CLOCK MODE** selects the division factor applied to the external clock input (CLKIN). When CLKMODE is high, CLKIN is divided by one to create PCLK2:1 and the processor's internal clock. When CLKMODE is low, CLKIN is divided by two to create PCLK2:1 and the processor's internal clock. CLKMODE should be tied high or low in a system, as the clock mode is not latched by the processor. If left unconnected, the processor internally pulls the CLKMODE pin low, enabling the 2-x clock mode.
PCLK2 **PCLK1**	O S H(Q) R(Q)	**PROCESSOR OUTPUT CLOCKS** provide a timing reference for all inputs and outputs of the processor. All inputs and output timings are specified in relation to PCLK2 and PCLK1. PCLK2 and PCLK1 are identical signals. Two output pins are provided to allow flexibility in the system's allocation of capacitive loading on the clock. PCLK2:1 may also be connected at the processor to form a single clock signal.
V_{SS}	—	**GROUND** connections consist of 24 pins which must be connected externally to a V_{SS} board plane.
V_{CC}	—	**POWER** connections consist of 24 pins which must be connected externally to a V_{CC} board plane.
V_{CCPLL}	—	**V_{CCPLL}** is a separate V_{CC} supply pin for the phase lock loop used in 1x clock mode. Connecting a simple low pass filter to V_{CCPLL} may help reduce clock jitter (T_{CP}) in noisy environments. Otherwise, V_{CCPLL} should be connected to V_{CC}.
N/C	—	**NO CONNECT** pins must not be connected in a system.

Table 4. 80960CF Pin Description—DMA and Interrupt Unit Control Signals

Name	Type	Description
DREQ3 DREQ2 DREQ1 DREQ0	I A(L) H(Z) R(Z)	**DMA REQUEST** causes a DMA transfer to be requested. Each of the four signals request a transfer on a single channel. DREQ0 requests channel 0, DREQ1 requests channel 1, etc. When two or more channels are requested simultaneously, the channel with the highest priority is serviced first. Channel priority mode is programmable.
DACK3 DACK2 DACK1 DACK0	O S H(1) R(1)	**DMA ACKNOWLEDGE** indicates that a DMA transfer is being executed. Each of the four signals acknowledge a transfer for a single channel. DACK0 acknowledges channel 0, DACK1 acknowledges channel 1, etc. DACK3:0 are asserted when the requesting device of a DMA is accessed.
EOP3/TC3 EOP2/TC2 EOP1/TC1 EOP0/TC0	I / O A(L) H(Z/Q) R(Z)	**END OF PROCESS/TERMINAL COUNT** can be programmed as either an input (EOP3:0) or as an output (TC3:0), but not both. Each pin is individually programmable. When programmed as an input, EOPx causes the termination of a current DMA transfer for the channel corresponding to the EOPx pin. EOP0 corresponds to channel 0, EOP1 corresponds to channel 1, etc. When a channel is configured for source *and* destination chaining, the EOP pin for that channel causes termination of only the current buffer transferred and causes the next buffer to be transferred. EOP3:0 are asynchronous inputs. When programmed as an output, the channel's TCx pin indicates that the channel byte count has reached 0 and a DMA has terminated. TCx is driven with the same timing as DACKx during the last DMA transfer for a buffer. If the last bus request is executed as multiple bus accesses, TCx remains asserted for the entire bus request.
XINT7 XINT6 XINT5 XINT4 XINT3 XINT2 XINT1 XINT0	I A(E/L) H(Z) R(Z)	**EXTERNAL INTERRUPT PINS** cause interrupts to be requested. These pins can be configured in three modes. In Dedicated Mode, each pin is a dedicated external interrupt source. Dedicated inputs can be individually programmed to be level (low) or edge (falling) activated. In Expanded Mode, the 8 pins act together as an 8-bit vectored interrupt source. The interrupt pins in this mode are level activated. Since the interrupt pins are active low, the vector number requested is the one's complement of the positive logic value place on the port. This eliminates glue logic to interface to combinational priority encoders which output negative logic. In Mixed Mode, XINT7:5 are dedicated sources and XINT4:0 act as the 5 most significant bits of an expanded mode vector. The least significant bits are set to 010 internally.
NMI	I A(E) H(Z) R(Z)	**NON-MASKABLE INTERRUPT** causes a non-maskable interrupt event to occur. NMI is the highest priority interrupt recognized. NMI is an edge (falling) activated source.

1

3.3. 80960CF Pinout

3.3.1 80960CF PGA PINOUT

Tables 5 and 6 list the 80960CF pin names with package location. Figure 4-a depicts the complete 80960CF pinout as viewed from the top side of the component (i.e., pins facing down). Figure 4b shows the complete 80960CF pinout as viewed from the pin-side of the package (i.e., pins facing up). See **Section 4.0, Electrical Specifications** for specifications and recommended connections.

Table 5. PGA Pin Name with Package Location (Signal Order)

Address Bus	Data Bus	Bus Control	Processor Control	I/O
Name . . Location	*Name . . Location*	*Name . . Location*	*Name . . . . Location*	*Name . . Location*
A31 . . . . . S15	D31 . . . . . R03	$\overline{BE3}$ S05	$\overline{RESET}$ A16	$\overline{DREQ3}$ A07
A30 . . . . . Q13	D30 . . . . . Q05	$\overline{BE2}$ S06		$\overline{DREQ2}$ B06
A29 . . . . . R14	D29 . . . . . S02	$\overline{BE1}$ S07	$\overline{FAIL}$ A02	$\overline{DREQ1}$ A06
A28 . . . . . Q14	D28 . . . . . Q04	$\overline{BE0}$ R09		$\overline{DREQ0}$ B05
A27 . . . . . S16	D27 . . . . . R02		STEST B02	
A26 . . . . . R15	D26 . . . . . Q03	W/$\overline{R}$ S10		$\overline{DACK3}$ A10
A25 . . . . . S17	D25 . . . . . S01		$\overline{ONCE}$ C03	$\overline{DACK2}$ A09
A24 . . . . . Q15	D24 . . . . . R01	$\overline{ADS}$ R06		$\overline{DACK1}$ A08
A23 . . . . . R16	D23 . . . . . Q02		CKLIN C13	$\overline{DACK0}$ B08
A22 . . . . . R17	D22 . . . . . P03	$\overline{READY}$ S03	CLKMODE C14	
A21 . . . . . Q16	D21 . . . . . Q01	$\overline{BTERM}$ R04	PCLK1 . . . . . B14	$\overline{EOP}$/$\overline{TC0}$ . . . A11
A20 . . . . . P15	D20 . . . . . P02		PCLK2 . . . . . B13	$\overline{EOP}$/$\overline{TC1}$ . . . A12
A19 . . . . . P16	D19 . . . . . P01	$\overline{WAIT}$ S12		$\overline{EOP}$/$\overline{TC2}$ . . . A13
A18 . . . . . Q17	D18 . . . . . N02	$\overline{BLAST}$ S08	**V_{SS}**	$\overline{EOP}$/$\overline{TC3}$ . . . A14
A17 . . . . . P17	D17 . . . . . N01		*Location*	
A16 . . . . . N16	D16 . . . . . M01	DT/$\overline{R}$ S11	C07, C08, C09, C10, C11, C12, F15, G03, G15, H03, H15, J03, J15, K03, K15, L03, L15, M03, M15, Q07, Q08, Q09, Q10, Q11	$\overline{XINT7}$ C17
A15 . . . . . N17	D15 . . . . . L01	$\overline{DEN}$ S09		$\overline{XINT6}$ C16
A14 . . . . . M17	D14 . . . . . L02			$\overline{XINT5}$ B17
A13 . . . . . L16	D13 . . . . . K01	$\overline{LOCK}$ S14		$\overline{XINT4}$ C15
A12 . . . . . L17	D12 . . . . . J01			$\overline{XINT3}$ B16
A11 . . . . . K17	D11 . . . . . H01	HOLD R05		$\overline{XINT2}$ A17
A10 . . . . . J17	D10 . . . . . H02	HOLDA S04	**V_{CC}**	$\overline{XINT1}$ A15
A9 . . . . . H17	D9 . . . . . G01	BREQ R13	*Location*	$\overline{XINT0}$ B15
A8 . . . . . G17	D8 . . . . . F01		B07, B09, B11, B12, C06, E15, F03, F16, G02, H16, J02, J16, K02, K16, M02, M16, N03, N15, Q06, R07, R08, R10, R11	
A7 . . . . . G16	D7 . . . . . E01	D/$\overline{C}$ S13		$\overline{NMI}$ D15
A6 . . . . . F17	D6 . . . . . F02	$\overline{DMA}$ R12		
A5 . . . . . E17	D5 . . . . . D01	$\overline{SUP}$ Q12		
A4 . . . . . E16	D4 . . . . . E02			
			V_{CCPLL} B10	
A3 . . . . . D17	D3 . . . . . C01	$\overline{BOFF}$ B01	**No Connect**	
A2 . . . . . D16	D2 . . . . . D02		*Location*	
	D1 . . . . . C02		A01, A03, A04, A05, B03, B04, C04, C05, D03	
	D0 . . . . . E03			

Table 6. PGA Pin Name with Package Location (Pin Order)

Address Bus	Data Bus	Bus Control	Processor Control	I/O
Location . . Name	*Location . . Name*	*Location . . Name*	*Location . . . . Name*	*Location . . Name*
A01 . . . . . NC	C01 . . . . . D3	G01 . . . . . D9	M01 . . . . . D16	R01 . . . . . D24
A02 . . . . . $\overline{\text{FAIL}}$	C02 . . . . . D1	G02 . . . . . V_{CC}	M02 . . . . . V_{CC}	R02 . . . . . D27
A03 . . . . . NC	C03 . . . . . $\overline{\text{ONCE}}$	G03 . . . . . V_{SS}	M03 . . . . . V_{SS}	R03 . . . . . D31
A04 . . . . . NC	C04 . . . . . NC	G15 . . . . . V_{SS}	M15 . . . . . V_{SS}	R04 . . . . . $\overline{\text{BTERM}}$
A05 . . . . . NC	C05 . . . . . NC	G16 . . . . . A7	M16 . . . . . V_{CC}	R05 . . . . . HOLD
A06 . . . . . $\overline{\text{DREQ1}}$	C06 . . . . . V_{CC}	G17 . . . . . A8	M17 . . . . . A14	R06 . . . . . $\overline{\text{ADS}}$
A07 . . . . . $\overline{\text{DREQ3}}$	C07 . . . . . V_{SS}			R07 . . . . . V_{CC}
A08 . . . . . $\overline{\text{DACK1}}$	C08 . . . . . V_{SS}	H01 . . . . . D11	N01 . . . . . D17	R08 . . . . . V_{CC}
A09 . . . . . $\overline{\text{DACK2}}$	C09 . . . . . V_{SS}	H02 . . . . . D10	N02 . . . . . D18	R09 . . . . . $\overline{\text{BE0}}$
A10 . . . . . $\overline{\text{DACK3}}$	C10 . . . . . V_{SS}	H03 . . . . . V_{SS}	N03 . . . . . V_{CC}	R10 . . . . . V_{CC}
A11 . . . $\overline{\text{EOP/TC0}}$	C11 . . . . . V_{SS}	H15 . . . . . V_{SS}	N15 . . . . . V_{CC}	R11 . . . . . V_{CC}
A12 . . . $\overline{\text{EOP/TC1}}$	C12 . . . . . V_{SS}	H16 . . . . . V_{CC}	N16 . . . . . A16	R12 . . . . . $\overline{\text{DMA}}$
A13 . . . $\overline{\text{EOP/TC2}}$	C13 . . . . . CLKIN	H17 . . . . . A9	N17 . . . . . A15	R13 . . . . . BREQ
A14 . . . $\overline{\text{EOP/TC3}}$	C14 . . CLKMODE			R14 . . . . . A29
A15 . . . . . $\overline{\text{XINT1}}$	C15 . . . . . $\overline{\text{XINT4}}$	J01 . . . . . D12	P01 . . . . . D19	R15 . . . . . A26
A16 . . . . . $\overline{\text{RESET}}$	C16 . . . . . $\overline{\text{XINT6}}$	J02 . . . . . V_{CC}	P02 . . . . . D20	R16 . . . . . A23
A17 . . . . . $\overline{\text{XINT2}}$	C17 . . . . . $\overline{\text{XINT7}}$	J03 . . . . . V_{SS}	P03 . . . . . D22	R17 . . . . . A22
		J15 . . . . . V_{SS}	P15 . . . . . A20	
B01 . . . . . $\overline{\text{BOFF}}$	D01 . . . . . D5	J16 . . . . . V_{CC}	P16 . . . . . A19	S01 . . . . . D25
B02 . . . . . STEST	D02 . . . . . D2	J17 . . . . . A10	P17 . . . . . A17	S02 . . . . . D29
B03 . . . . . NC	D03 . . . . . NC			S03 . . . . . $\overline{\text{READY}}$
B04 . . . . . NC	D15 . . . . . $\overline{\text{NMI}}$	K01 . . . . . D13	Q01 . . . . . D21	S04 . . . . . HOLDA
B05 . . . . . $\overline{\text{DREQ0}}$	D16 . . . . . A2	K02 . . . . . V_{CC}	Q02 . . . . . D23	S05 . . . . . $\overline{\text{BE3}}$
B06 . . . . . $\overline{\text{DREQ2}}$	D17 . . . . . A3	K03 . . . . . V_{SS}	Q03 . . . . . D26	S06 . . . . . $\overline{\text{BE2}}$
B07 . . . . . V_{CC}		K15 . . . . . V_{SS}	Q04 . . . . . D28	S07 . . . . . $\overline{\text{BE1}}$
B08 . . . . . $\overline{\text{DACK0}}$	E01 . . . . . D7	K16 . . . . . V_{CC}	Q05 . . . . . D30	S08 . . . . . $\overline{\text{BLAST}}$
B09 . . . . . V_{CC}	E02 . . . . . D4	K17 . . . . . A11	Q06 . . . . . V_{CC}	S09 . . . . . $\overline{\text{DEN}}$
B10 . . . . . V_{CCPLL}	E03 . . . . . D0		Q07 . . . . . V_{SS}	S10 . . . . . W/$\overline{\text{R}}$
B11 . . . . . V_{CC}	E15 . . . . . V_{CC}	L01 . . . . . D15	Q08 . . . . . V_{SS}	S11 . . . . . DT/$\overline{\text{R}}$
B12 . . . . . V_{CC}	E16 . . . . . A4	L02 . . . . . D14	Q09 . . . . . V_{SS}	S12 . . . . . $\overline{\text{WAIT}}$
B13 . . . . . PCLK2	E17 . . . . . A5	L03 . . . . . V_{SS}	Q10 . . . . . V_{SS}	S13 . . . . . D/$\overline{\text{C}}$
B14 . . . . . PCLK1		L15 . . . . . V_{SS}	Q11 . . . . . V_{SS}	S14 . . . . . $\overline{\text{LOCK}}$
B15 . . . . . $\overline{\text{XINT0}}$	F01 . . . . . D8	L16 . . . . . A13	Q12 . . . . . $\overline{\text{SUP}}$	S15 . . . . . A31
B16 . . . . . $\overline{\text{XINT3}}$	F02 . . . . . D6	L17 . . . . . A12	Q13 . . . . . A30	S16 . . . . . A27
B17 . . . . . $\overline{\text{XINT5}}$	F03 . . . . . V_{CC}		Q14 . . . . . A28	S17 . . . . . A25
	F15 . . . . . V_{SS}		Q15 . . . . . A24	
	F16 . . . . . V_{CC}		Q16 . . . . . A21	
	F17 . . . . . A6		Q17 . . . . . A18	

1

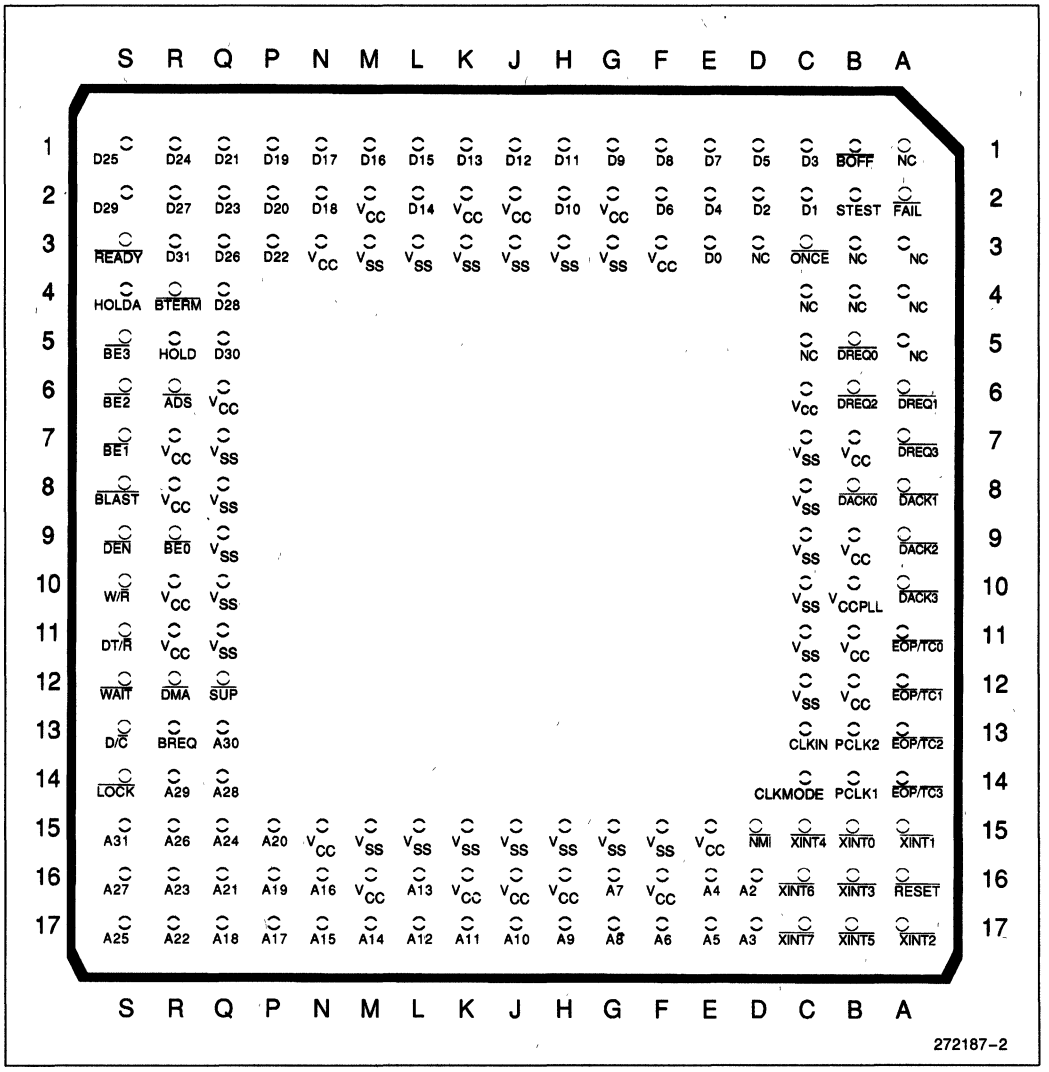


Figure 4a. 80960CF PGA Pinout (View from Top Side)

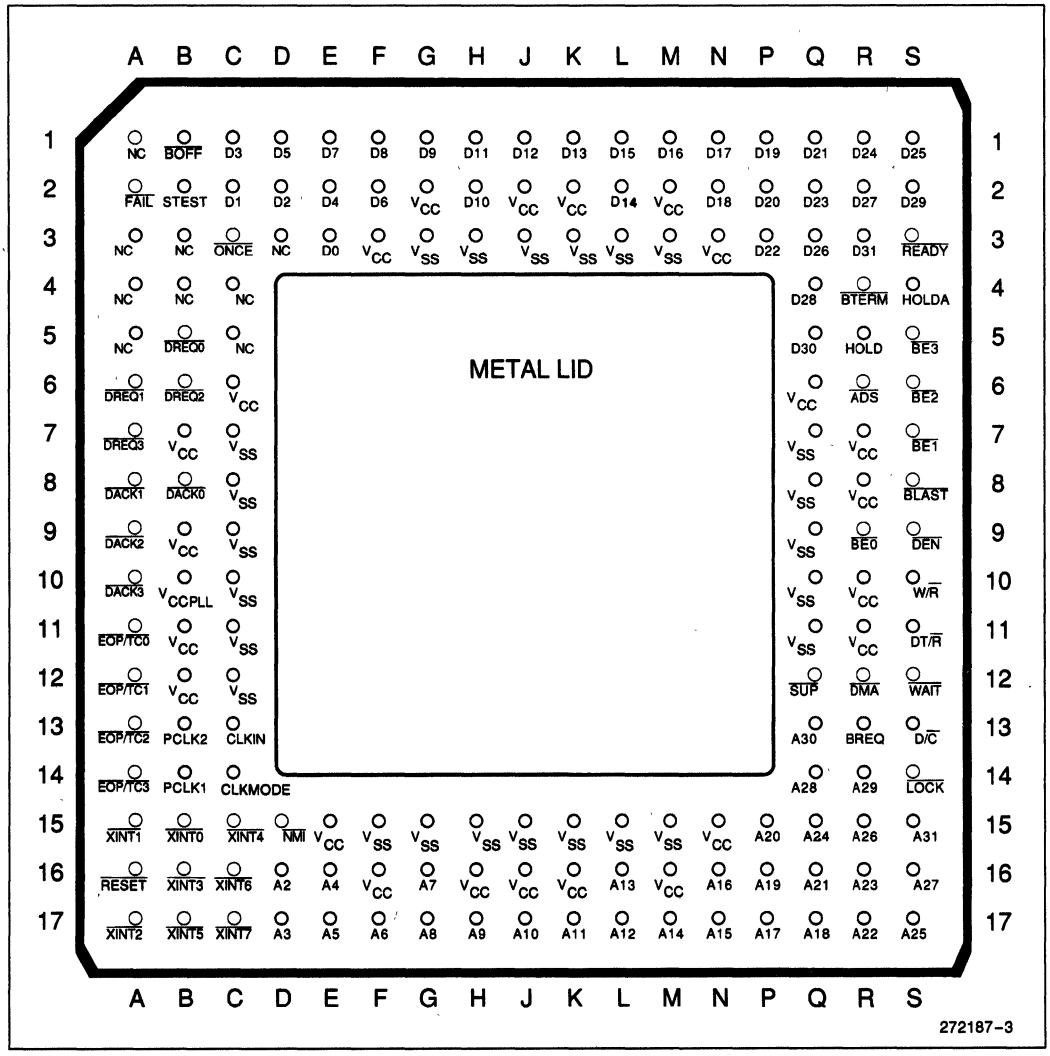


Figure 4b. 80960CF PGA Pinout (View from Bottom Side)

3.3.2 80960CF PQFP Pinout

See **Section 4.0, Electrical Specifications** for specifications and recommended connections.

Tables 7 and 8 list the 80960CF pin names with package location.

Table 7. PQFP Pin Name with Package Location (Signal Order)

Address Bus	Data Bus	Bus Control	Processor Control	I/O
Name . . Location	*Name . . Location*	*Name . . Location*	*Name . . . . . Location*	*Name . . Location*
A31 153	D31 186	$\overline{BE3}$ 176	$\overline{RESET}$ 091	$\overline{DREQ3}$ 060
A30 152	D30 187	$\overline{BE2}$ 175		$\overline{DREQ2}$ 059
A29 151	D29 188	$\overline{BE1}$ 172	$\overline{FAIL}$ 045	$\overline{DREQ1}$ 058
A28 145	D28 189	$\overline{BE0}$ 170		$\overline{DREQ0}$ 057
A27 144	D27 191		STEST 046	
A26 143	D26 192	W/$\overline{R}$ 164		$\overline{DACK3}$ 065
A25 142	D25 194		$\overline{ONCE}$ 043	$\overline{DACK2}$ 064
A24 141	D24 195	$\overline{ADS}$ 178		$\overline{DACK1}$ 063
A23 139	D23 003		CLKIN 087	$\overline{DACK0}$ 062
A22 138	D22 004	$\overline{READY}$ 182	CLKMODE 085	
A21 137	D21 005	$\overline{BTERM}$ 184	PCLK1 078	$\overline{EOP}$/$\overline{TC3}$ . . . 069
A20 136	D20 006		PCLK2 074	$\overline{EOP}$/$\overline{TC2}$ . . . 068
A19 134	D19 008	$\overline{WAIT}$ 162		$\overline{EOP}$/$\overline{TC1}$ . . . 067
A18 133	D18 009	$\overline{BLAST}$ 169	**V_{SS}**	$\overline{EOP}$/$\overline{TC0}$ . . . 066
A17 132	D17 010		*Location*	
A16 130	D16 011	DT/$\overline{R}$ 163	2, 7, 16, 24, 30, 38, 39, 49, 56, 70, 75, 77, 81, 83, 88, 89, 92, 98, 105, 109, 110, 121, 125, 131, 135, 147, 150, 161, 165, 173, 174, 185, 196	$\overline{XINT7}$ 107
A15 129	D15 013	$\overline{DEN}$ 167		$\overline{XINT6}$ 106
A14 128	D14 014			$\overline{XINT5}$ 102
A13 124	D13 015	$\overline{LOCK}$ 156		$\overline{XINT4}$ 101
A12 123	D12 017			$\overline{XINT3}$ 100
A11 122	D11 018	HOLD 181		$\overline{XINT2}$ 095
A10 120	D10 019	HOLDA 179	**V_{CC}**	$\overline{XINT1}$ 094
A9 119	D9 021	BREQ 155	*Location*	$\overline{XINT0}$ 093
A8 118	D8 022		1, 12, 20, 28, 32, 37, 44, 50, 61, 71, 79, 82, 96, 99, 103, 115, 127, 140, 148, 154, 168, 171, 180, 190	
A7 117	D7 023	D/$\overline{C}$ 159		$\overline{NMI}$ 108
A6 116	D6 025	$\overline{DMA}$ 160		
A5 114	D5 026	$\overline{SUP}$ 158		
A4 113	D4 027		V_{CCPLL} 72	
A3 112	D3 033	$\overline{BOFF}$ 040	**No Connect**	
A2 111	D2 034		*Location*	
	D1 035		29, 31, 41, 42, 47, 48, 51, 52, 53, 54, 55, 73, 76, 80, 84, 86, 90, 97, 104, 126, 146, 149, 157, 166, 177, 183, 193	
	D0 036			

Table 8. PQFP Pin Name with Package Location (Pin Order)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	V_{CC}	50	V_{CC}	99	V_{CC}	148	V_{CC}
2	V_{SS}	51	NC	100	$\overline{XINT3}$	149	NC
3	D23	52	NC	101	$\overline{XINT4}$	150	V_{SS}
4	D22	53	NC	102	$\overline{XINT5}$	151	A29
5	D21	54	NC	103	V_{CC}	152	A30
6	D20	55	NC	104	NC	153	A31
7	V_{SS}	56	V_{SS}	105	V_{SS}	154	V_{CC}
8	D19	57	$\overline{DREQ0}$	106	$\overline{XINT6}$	155	BREQ
9	D18	58	$\overline{DREQ1}$	107	$\overline{XINT7}$	156	$\overline{LOCK}$
10	D17	59	$\overline{DREQ2}$	108	$\overline{NMI}$	157	NC
11	D16	60	$\overline{DREQ3}$	109	V_{SS}	158	$\overline{SUP}$
12	V_{CC}	61	V_{CC}	110	V_{SS}	159	D/$\overline{C}$
13	D15	62	$\overline{DACK0}$	111	A2	160	$\overline{DMA}$
14	D14	63	$\overline{DACK1}$	112	A3	161	V_{SS}
15	D13	64	$\overline{DACK2}$	113	A4	162	$\overline{WAIT}$
16	V_{SS}	65	$\overline{DACK3}$	114	A5	163	DT/$\overline{R}$
17	D12	66	$\overline{EOP0}$/$\overline{TC0}$	115	V_{CC}	164	W/$\overline{R}$
18	D11	67	$\overline{EOP1}$/$\overline{TC1}$	116	A6	165	V_{SS}
19	D10	68	$\overline{EOP2}$/$\overline{TC2}$	117	A7	166	NC
20	V_{CC}	69	$\overline{EOP3}$/$\overline{TC3}$	118	A8	167	$\overline{DEN}$
21	D9	70	V_{SS}	119	A9	168	V_{CC}
22	D8	71	V_{CC}	120	A10	169	$\overline{BLAST}$
23	D7	72	V_{CCPLL}	121	V_{SS}	170	$\overline{BE0}$
24	V_{SS}	73	NC	122	A11	171	V_{CC}
25	D6	74	PCLK2	123	A12	172	$\overline{BE1}$
26	D5	75	V_{SS}	124	A13	173	V_{SS}
27	D4	76	NC	125	V_{SS}	174	V_{SS}
28	V_{CC}	77	V_{SS}	126	NC	175	$\overline{BE2}$
29	NC	78	PCLK1	127	V_{CC}	176	$\overline{BE3}$
30	V_{SS}	79	V_{CC}	128	A14	177	NC
31	NC	80	NC	129	A15	178	$\overline{ADS}$
32	V_{CC}	81	V_{SS}	130	A16	179	HOLDA
33	D3	82	V_{CC}	131	V_{SS}	180	V_{CC}
34	D2	83	V_{SS}	132	A17	181	HOLD
35	D1	84	NC	133	A18	182	$\overline{READY}$
36	D0	85	CLKMODE	134	A19	183	NC
37	V_{CC}	86	NC	135	V_{SS}	184	$\overline{BTERM}$
38	V_{SS}	87	CLKIN	136	A20	185	V_{SS}
39	V_{SS}	88	V_{SS}	137	A21	186	D31
40	$\overline{BOFF}$	89	V_{SS}	138	A22	187	D30
41	NC	90	NC	139	A23	188	D29
42	NC	91	$\overline{RESET}$	140	V_{CC}	189	D28
43	$\overline{ONCE}$	92	V_{SS}	141	A24	190	V_{CC}
44	V_{CC}	93	$\overline{XINT0}$	142	A25	191	D27
45	$\overline{FAIL}$	94	$\overline{XINT1}$	143	A26	192	D26
46	STEST	95	$\overline{XINT2}$	144	A27	193	NC
47	NC	96	V_{CC}	145	A28	194	D25
48	NC	97	NC	146	NC	195	D24
49	V_{SS}	98	V_{SS}	147	V_{SS}	196	V_{SS}

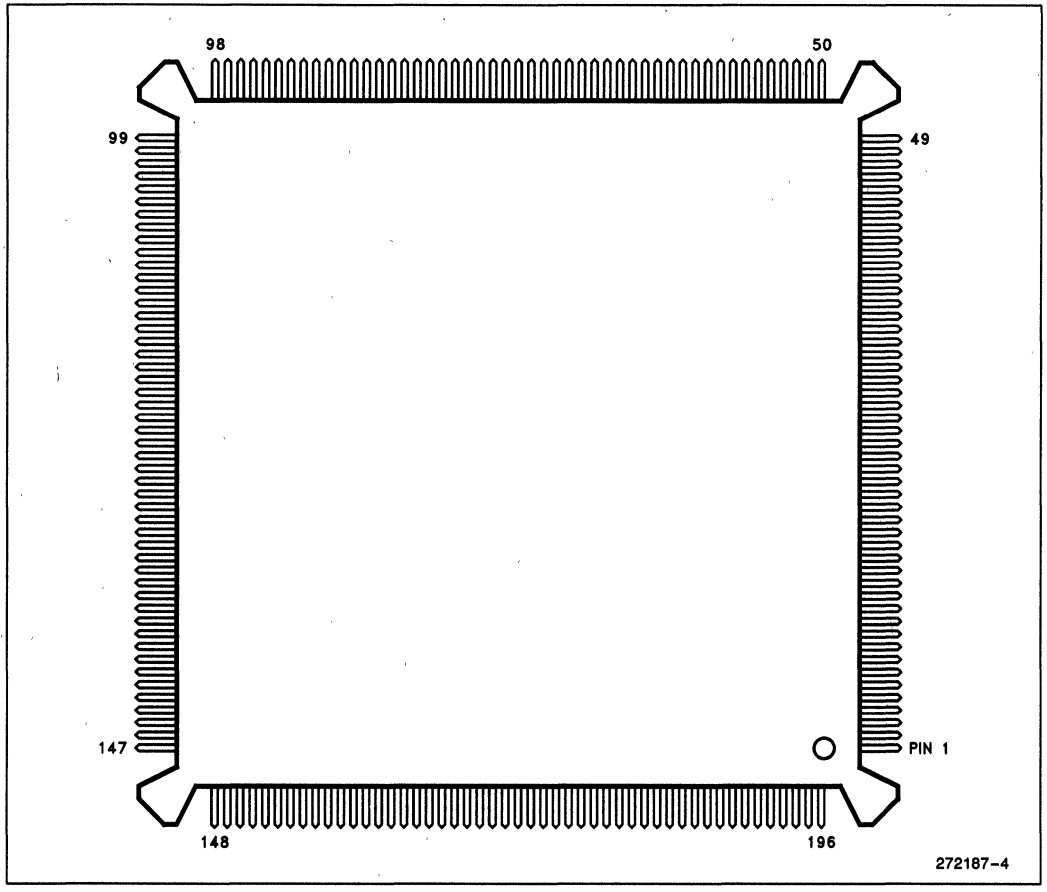
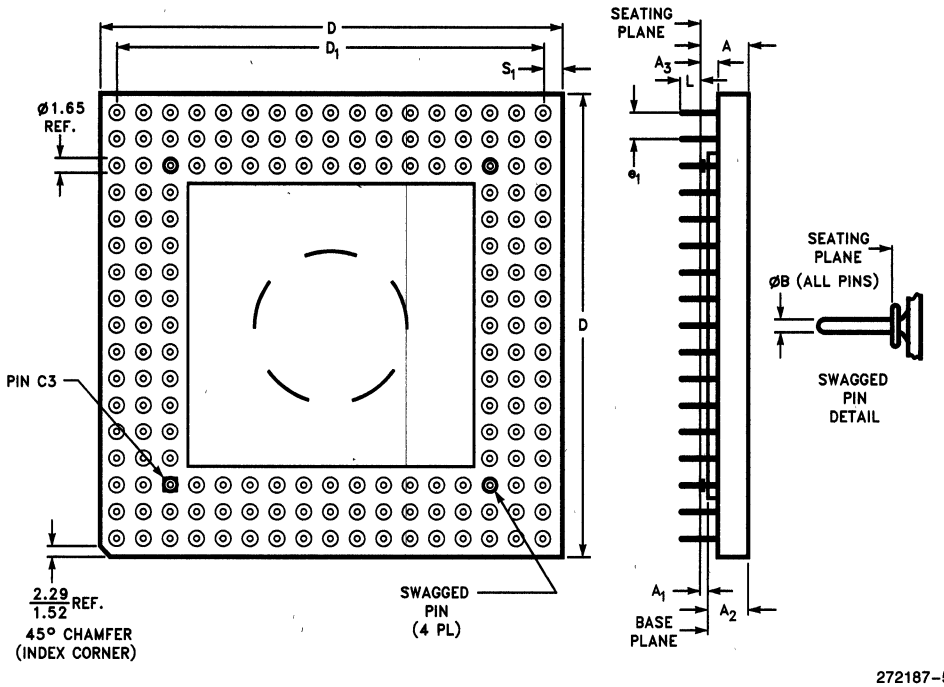


Figure 4c. 80960CF PQFP Pinout (View from Top Side)

3.4. Mechanical Data

3.4.1 CERAMIC PGA PACKAGE



Family: Ceramic Pin Grid Array Package						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
A	3.56	4.57		0.140	0.180	
A_1	0.64	1.14	SOLID LID	0.025	0.045	SOLID LID
A_2	23	0.30	SOLID LID	0.110	0.140	SOLID LID
A_3	1.14	1.40		0.045	0.055	
B	0.43	0.51		0.017	0.020	
D	44.07	44.83		1.735	1.765	
D_1	40.51	40.77		1.595	1.605	
e_1	2.29	2.79		0.090	0.110	
L	2.54	3.30		0.100	0.130	
N	168			168		
S_1	1.52	2.54		0.060	0.100	
ISSUE	IWS REV X 7/15/88					

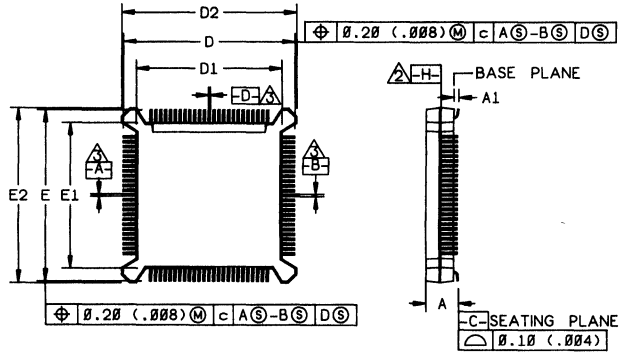
Figure 5. 168-Lead Ceramic PGA Package Dimensions

Table 9. Ceramic PGA Package Dimension Symbols

Letter or Symbol	Description of Dimensions
A	Distance from seating plane to highest point of body
A_1	Distance between seating plane and base plane (lid)
A_2	Distance from base plane to highest point of body
A_3	Distance from seating plane to bottom of body
B	Diameter of terminal lead pin
D	Largest overall package dimension of length
D_1	A body length dimension, outer lead center to outer lead center
e_1	Linear spacing between true lead position centerlines
L	Distance from seating plane to end of lead
S_1	Other body dimension, outer lead center to edge of body

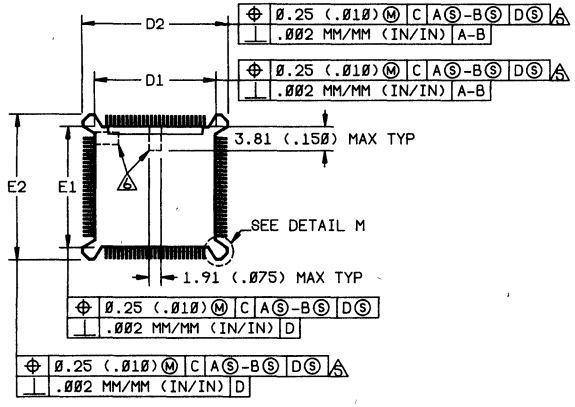
NOTES:
1. Controlling dimension: millimeter.
2. Dimension "e_1" ("e") is non-cumulative.
3. Seating plane (standoff) is defined by P.C. board hole size: 0.0415–0.0430 inch.
4. Dimensions "B", "B_1" and "C" are nominal.
5. Details of Pin 1 identifier are optional.

3.4.2 PLASTIC QUAD FLAT PACKAGE



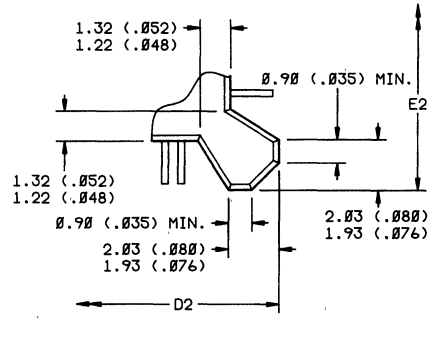
mm (inch)

Figure 6. Principal Dimensions and Data



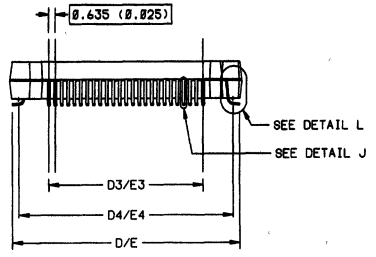
mm (inch)

Figure 7. Molded Details



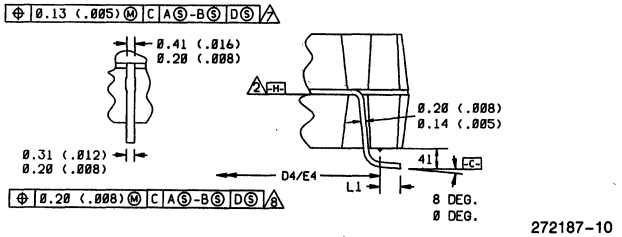
mm (inch)

Figure 8. Detail M



mm (inch)

Figure 9. Terminal Details



mm (inch)

Figure 10. Typical Lead

Table 10. PQFP Package Dimension Symbols

Symbol	Description	Min	Max	Min	Max
N	Leadcount	196		196	
A	Package Height	0.160	0.170	4.06	4.32
A1	Standoff	0.020	0.030	0.51	0.76
D, E	Terminal Dimension	1.475	1.485	37.47	37.72
D1, E1	Package Body	1.347	1.353	34.21	34.37
D2, E2	Bumper Distance	1.497	1.503	38.02	38.18
D3, E3	Lead Dimension	1.200 REF		30.48 REF	
D4, E4	Foot Radius Location	1.423	1.437	36.14	36.49
L1	Foot Length	0.020	0.030	0.51	0.76
Dimension			INCH		mm

NOTES:
1. All dimensions and tolerances conform to ANSI Y14.5M-1982.
2. Datum plane -H- located at the mold parting line and coincident with the bottom of the lead where lead exits plastic body.
3. Datums A-B and -D- to be determined where center leads exit plastic body at datum plane -H-.
4. Controlling Dimension, Inch.
5. Dimensions D1, D2, E1 and E2 are measured at the mold parting line. D1 and E1 do not include an allowable mold protrusion of 0.18 mm (0.007 in) per side. D2 and E2 do not include a total allowable mold protrusion of 0.18 mm (0.007 in) at maximum package size.
6. Pin 1 identifier is located within one of the two zones indicated.
7. Measured at datum plane -H-.
8. Measured at seating plane datum -C-.

3.5. Package Thermal Specifications

The 80960CF is specified for operation when T_C (the case temperature) is within the range of 0°C–100°C. T_C may be measured in any environment to determine whether the 80960CF is within specified operating range. The case temperature is measured at the center of the top surface, opposite the pins. Refer to Figure 13.

T_A (the ambient temperature) can be calculated from θ_{CA} (thermal resistance from case to ambient) with the following equation:

$$T_A = T_C - P*\theta_{CA}$$

Table 11 shows the maximum T_A allowable (without exceeding T_C) at various airflows and operating frequencies (f_{PCLK}).

Note that T_A is greatly improved by attaching fins or a heat sink to the package. P (the maximum power consumption) is calculated by using the typical I_{CC} as tabulated in Section 4.4, **DC Specifications**, and V_{CC} of 5V.

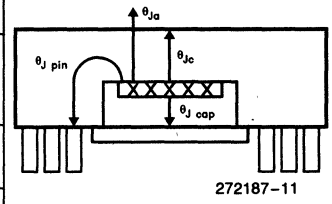
1

Table 11. Maximum T_A at Various Airflows In °C (PGA Package Only)

		Airflow-ft/min (m/sec)					
	f_{PCLK} (MHz)	0 (0)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)	1000 (5.07)
T_A with Heat Sink*	33	38	57	74	76	81	84
	25	50	65	79	81	85	87
	16	63	74	84	86	89	90
T_A without Heat Sink	33	18	33	47	57	66	67
	25	34	46	57	65	72	74
	16	51	60	68	74	80	81

*0.285" high unidirectional heat sink (Al alloy 6061, 50 mil fin width, 150 mil center-to-center fin spacing).

PGA Thermal Resistance—°C/Watt						
Parameter	Airflow—ft./min (m/sec)					
	0 (0)	200 (1.01)	400 (2.03)	600 (3.07)	800 (4.06)	1000 (5.07)
θ Junction-to-Case (Case Measured as shown in Figure 13)	1.5	1.5	1.5	1.5	1.5	1.5
θ Case-to-Ambient (No Heatsink)	17	14	11	9	7.1	6.6
θ Case-to-Ambient (with Unidirectional) Heatsink)*	13	9	5.5	5.0	3.9	3.4



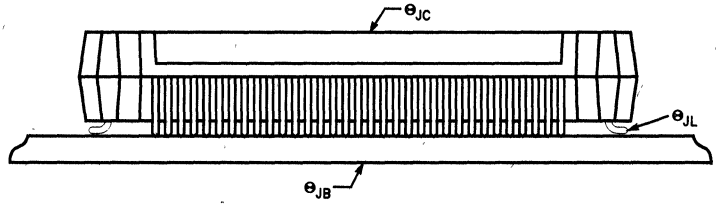
272187-11

NOTES:
1. This table applies to 80960CF PGA plugged into socket or soldered directly into board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$.
3. $\theta_{J\text{-}CAP}$ = 4°C/W (approx.)
   $\theta_{J\text{-}PIN}$ = 4°C/W (inner pins) (approx.)
   $\theta_{J\text{-}PIN}$ = 8°C/W (outer pins) (approx.)
* 0.285" high unidirectional heat sink (Al alloy 6061, 50 mil fin width, 150 mil center-to-center fin spacing).

Figure 11. 80960CF PGA Package Thermal Characteristics

PQFP Thermal Resistance—°C/Watt							
Parameter	Airflow—ft./min (m/sec)						
	0 (0)	50 (0.25)	100 (0.50)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)
θ Junction-to-Case (Case Measured) as shown in Figure 13)	5	5	5	5	5	5	5
θ Case-to-Ambient (No Heatsink)	19	18	17	15	12	10	9

NOTES:
1. This table applies to 80960CF PQFP soldered directly into board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$.
3. $\theta_{JL} = 18°C/Watt$
   $\theta_{JB} = 18°C/Watt$



272187-12

Figure 12. 80960CF PQFP Package Thermal Characteristics

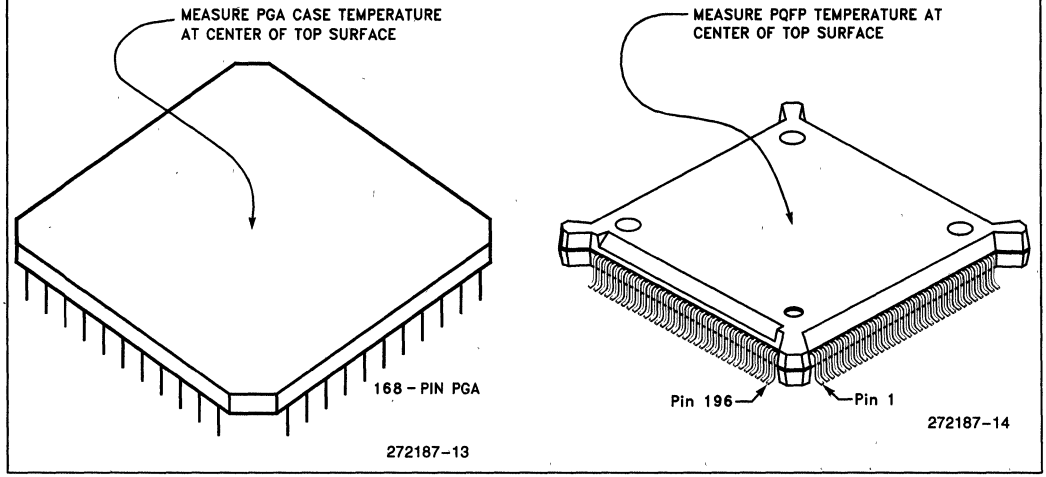


Figure 13. Measuring 80960CF PGA and PQFP Case Temperature

3.6 Stepping Register Information

Upon Reset, Register G0 contains die stepping information. The following figure shows how G0 is configured. The most significant byte contains an ASCII 0. The upper middle byte contains an ASCII C. The lower middle byte contains an ASCII F. The least significant byte contains the stepping number in ASCII. G0 retains this information until it is written over by the user program.

Table 12 contains a cross reference of the number in the least significant byte of register G0 to the die stepping number.

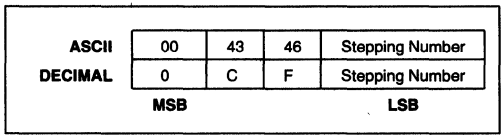


Figure 14. Register G0

Table 12. Die Stepping Cross Reference

G0 Least Significant Byte	Die Stepping
01	A
02	B
03	C

3.7 Suggested Sources for 80960CF Accessories

The following are some suggested sources of accessories for the 80960CF. They are neither an endorsement of any kind, nor a warranty of the performance of any of the listed products and/or companies.

Sockets

1. 3M Textool Test and Interconnection Products Department
   P.O. Box 2963
   Austin, TX 78769-2963

2. Augat, Inc.
   Interconnection Products Group
   33 Perry Avenue
   P.O. Box 779
   Attleboro, MA 02703
   (508) 222-2202

3. Concept Manufacturing Inc.
   (Decoupling Sockets)
   43024 Christy Street
   Fremont, CA 94538
   (415) 651-3804

Heat Sinks/Fins

1. Thermalloy, Inc.
   2021 West Valley View Lane
   Dallas, TX 75381-0839
   (214) 243-4321

2. E G & G Division
   60 Audubon Road
   Wakefield, MA 01880
   (617) 245-5900

4.0 ELECTRICAL SPECIFICATIONS

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. It is valid for the devices indicated in the revision history. The specifications are subject to change without notice.

4.1 Absolute Maximum Ratings

Parameter	Maximum Rating
Storage Temperature	−65 °C to +150 °C
Case Temperature Under Bias	−65 °C to +110 °C
Supply Voltage wrt. V_{SS}	−0.5V to +6.5V
Voltage on Other pins wrt V_{SS}	−0.5V to V_{CC} +0.5V

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

4.2. Operating Conditions

Operating Conditions (80960CF-33, -25, -16)

Symbol	Parameter		Min	Max	Units	Notes
V_{CC}	Supply Voltage	80960CF-33	4.75	5.25	V	
		80960CF-25	4.50	5.50		
		80960CF-16	4.50	5.50		
f_{CLK2x}	Input Clock Frequency (2-x Mode)	80960CF-33	0	66.66	MHz	
		80960CF-25	0	50	MHz	
		80960CF-16	0	32	MHz	
f_{CLK1x}	Input Clock Frequency (1-x Mode)	80960CF-33	8	33.33	MHz	(1)
		80960CF-25	8	25	MHz	
		80960CF-16	8	16	MHz	
T_C	Case Temperature Under Bias 80960CF-33, -25, -16	PGA Package	0	100	°C	
		196-Pin PQFP	0	100		

NOTE:
(1) When in the 1-x input clock mode, CLKIN is an input to an internal phase-locked loop and must maintain a minimum frequency of 8 MHz for proper processor operation. However, in the 1-x Mode, CLKIN may still be stopped when the processor either is in a reset condition or is reset. If CLKIN is stopped, the specified $\overline{RESET}$ low time must be provided once CLKIN restarts and has stabilized.

4.3 Recommended Connections

Power and ground connections must be made to multiple V_{CC} and V_{SS} (GND) pins. Every 80960CF-based circuit board should include power (V_{CC}) and ground (V_{SS}) planes for power distribution. Every V_{CC} pin must be connected to the power plane, and every V_{SS} pin must be connected to the ground plane. Pins identified as "N.C." **must not** be connected in the system.

Liberal decoupling capacitance should be placed near the 80960CF. The processor can cause transient power surges when its numerous output buffers transition, particularly when connected to large capacitive loads.

Low inductance capacitors and interconnects are recommended for best high frequency electrical performance. Inductance can be reduced by shortening board traces between the processor and decoupling capacitors as much as possible. Capacitors specifically designed for PGA packages will offer the lowest possible inductance.

For reliable operation, always connect unused inputs to an appropriate signal level. In particular, any unused interrupt ($\overline{XINT}$, $\overline{NMI}$) or DMA ($\overline{DREQ}$) input should be connected to V_{CC} through a pull-up resistor, as should $\overline{BTERM}$ if not used. Pull-up resistors should be in the range of 20 KΩ for each pin tied high. If $\overline{READY}$ or HOLD are not used, the unused input should be connected to ground. **N.C. pins must always remain unconnected.** Refer to the *i960 CA Microprocessor Reference Manual* for more information.

4.4. DC Specifications

DC Characteristics

(80960CF-33, -25, -16 under the conditions described in **Section 4.2, Operating Conditions.**)

Symbol	Parameter	Min	Max	Units	Notes
V_{IL}	Input Low Voltage for all pins except $\overline{RESET}$	−0.3	0.8	V	
V_{IH}	Input High Voltage for all pins except $\overline{RESET}$	2.0	V_{CC} + 0.3	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 5 mA
V_{OH}	Output High Voltage I_{OH} = −1mA I_{OH} = −200μA	2.4 V_{CC} − 0.5		V V	
V_{ILR}	Input Low Voltage for $\overline{RESET}$	− 0.3	1.5	V	
V_{IHR}	Input High Voltage for $\overline{RESET}$	3.5	V_{CC} + 0.3	V	
I_{LI1}	Input Leakage Current for each pin *except*: $\overline{BTERM}$, $\overline{ONCE}$, $\overline{DREQ3:0}$, STEST, $\overline{EOP3:0}$/$\overline{TC3:0}$, $\overline{NMI}$, $\overline{XINT7:0}$, $\overline{READY}$, HOLD, $\overline{BOFF}$, CLKMODE		±15	μA	0V≤V_{IN}≤V_{CC} (1)
I_{LI2}	Input Leakage Current for: $\overline{BTERM}$, $\overline{ONCE}$, $\overline{DREQ3:0}$, STEST, $\overline{EOP3:0}$/$\overline{TC3:0}$, $\overline{NMI}$, $\overline{XINT7:0}$, $\overline{BOFF}$	0	−300	μA	V_{IN} = 0.45V (2)
I_{LI3}	Input Leakage Current for: $\overline{READY}$, HOLD, CLKMODE	0	500	μA	V_{IN} = 2.4V (3)
I_{LO}	Output Leakage Current		±15	μA	0.45V≤V_{OUT}≤V_{CC}
I_{CC}	Supply Current (80960CF-33) I_{CC} Max I_{CC} Typ		 1150 960	 mA	 (4) (5)
I_{CC}	Supply Current (80960CF-25) I_{CC} Max I_{CC} Typ		 950 775	 mA	 (4) (5)
I_{CC}	Supply Current (80960CF-16) I_{CC} Max I_{CC} Typ		 750 575	 mA	 (4) (5)
I_{ONCE}	ONCE-mode Supply Current		150	mA	
C_{IN}	Input Capacitance for: CLKIN, $\overline{RESET}$, $\overline{ONCE}$, $\overline{READY}$, HOLD, $\overline{DREQ3:0}$, $\overline{BOFF}$ $\overline{XINT7:0}$, $\overline{NMI}$, $\overline{BTERM}$, CLKMODE	0	12	pF	F_C = 1 MHz
C_{OUT}	Output Capacitance of each output pin		12	pF	F_C = 1 MHz, (6)
$C_{I/O}$	I/O Pin Capacitance		12	pF	F_C = 1 MHz

NOTES:
(1) No Pull-up or pull-down.
(2) These pins have internal pullup resistors.
(3) These pins have internal pulldown resistors.
(4) Measured at worst case frequency, V_{CC} and temperature, with device operating and outputs loaded to the test conditions described in **Section 4.5.1, AC Test Conditions.**
(5) I_{CC} Typical is not tested.
(6) Output Capacitance is the capacitive load of a floating output.
(7) CLKMODE pin has a pulldown resistor only when $\overline{ONCE}$ pin is deasserted.

4.5 AC Specifications

AC Characteristics — 80960CF-33

(80960CF-33 only, under the conditions described in **Section 4.2, Operating Conditions** and **Section 4.5.1, AC Test Conditions.**)

Symbol	Parameter		Min	Max	Units	Notes
	INPUT CLOCK(10)					
T_F	CLKIN Frequency		0	66.66	MHz	(1)
T_C	CLKIN Period	In 1-x Mode (f_{CLK1x})	30	125	ns	(1,12)
		In 2-x Mode (f_{CLK2x})	15	∞	ns	(1)
T_{CS}	CLKIN Period Stability	In 1-x Mode (f_{CLK1x})		±0.1%	Δ	(1,13)
T_{CH}	CLKIN High Time	In 1-x Mode (f_{CLK1x})	6	62.5	ns	(1,12)
		In 2-x Mode (f_{CLK2x})	6	∞	ns	(1)
T_{CL}	CLKIN Low Time	In 1-x Mode (f_{CLK1x})	6	62.5	ns	(1,12)
		In 2-x Mode (f_{CLK2x})	6	∞	ns	(1)
T_{CR}	CLKIN Rise Time		0	6	ns	(1)
T_{CF}	CLKIN Fall Time		0	6	ns	(1)
	OUTPUT CLOCKS(9)					
T_{CP}	CLKIN to PCLK2:1 Delay	In 1-x Mode (f_{CLK1x})	−2	2	ns	(1,3,13,14)
		In 2-x Mode (f_{CLK2x})	2	25	ns	(1,3)
T	PCLK2:1 Period	In 1-x Mode (f_{CLK1x})	T_C		ns	(1,13)
		In 2-x Mode (f_{CLK2x})	$2T_C$		ns	(1,3)
T_{PH}	PCLK2:1 High Time		(T/2) − 2	T/2	ns	(1,13)
T_{PL}	PCLK2:1 Low Time		(T/2) − 2	T/2	ns	(1,13)
T_{PR}	PCLK2:1 Rise Time		1	4	ns	(1,3)
T_{PF}	PCLK2:1 Fall Time		1	4	ns	(1,3)
	SYNCHRONOUS OUTPUTS(10)					
T_{OV} T_{OH}	Output Valid Delay, Output Hold					(6, 11)
	T_{OV1}, T_{OH1}	A31:2	3	14	ns	
	T_{OV2}, T_{OH2}	$\overline{BE3:0}$	3	16	ns	
	T_{OV3}, T_{OH3}	$\overline{ADS}$	6	18	ns	
	T_{OV4}, T_{OH4}	W/$\overline{R}$	3	18	ns	
	T_{OV5}, T_{OH5}	D/$\overline{C}$, $\overline{SUP}$, $\overline{DMA}$	4	16	ns	
	T_{OV6}, T_{OH6}	$\overline{BLAST}$, $\overline{WAIT}$	5	16	ns	
	T_{OV7}, T_{OH7}	$\overline{DEN}$	3	16	ns	
	T_{OV8}, T_{OH8}	HOLDA, BREQ	4	16	ns	
	T_{OV9}, T_{OH9}	$\overline{LOCK}$	4	16	ns	
	T_{OV10}, T_{OH10}	$\overline{DACK3:0}$	4	18	ns	
	T_{OV11}, T_{OH11}	D31:0	3	16	ns	
	T_{OV12}, T_{OH12}	DT/$\overline{R}$	T/2 + 3	T/2 + 14	ns	
	T_{OV13}, T_{OH13}	$\overline{FAIL}$	2	14	ns	(6, 11)
	T_{OV14}, T_{OH14}	$\overline{EOP3:0}$/TC3:0	3	18	ns	
T_{OF}	Output Float for all outputs		3	22	ns	(6)
	SYNCHRONOUS INPUTS(10)					
T_{IS}	Input Setup					
	T_{IS1}	D31:0	3		ns	(1,11)
	T_{IS2}	$\overline{BOFF}$	17		ns	(1,11)
	T_{IS3}	$\overline{BTERM}$/READY	7		ns	(1,11)
	T_{IS4}	HOLD	7		ns	(1,11)
T_{IH}	Input Hold					
	T_{IH1}	D31:0	5		ns	(1,11)
	T_{IH2}	$\overline{BOFF}$	5		ns	(1,11)
	T_{IH3}	$\overline{BTERM}$/READY	2		ns	(1,11)
	T_{IH4}	HOLD	3		ns	(1,11)

AC Characteristics — 80960CF-33

80960CF-33 only, under the conditions described in **Section 4.2, Operating Conditions** and **Section 4.5.1, AC Test Conditions.**) (Continued)

Symbol	Parameter	Min	Max	Units	Notes
	RELATIVE OUTPUT TIMINGS(9,7)				
T_{AVSH1}	A31:2 Valid to $\overline{ADS}$ Rising	T − 4	T + 4	ns	
T_{AVSH2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, D/$\overline{C}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{ADS}$ Rising	T − 6	T + 6	ns	
T_{AVEL1}	A31:2 Valid to $\overline{DEN}$ Falling	T − 4	T + 4	ns	
T_{AVEL2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, $\overline{INST}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{DEN}$ Falling	T − 6	T + 6	ns	
T_{NLQV}	$\overline{WAIT}$ Falling to Output Data Valid	±6		ns	
T_{DVNH}	Output Data Valid to $\overline{WAIT}$ Rising	N*T − 6	N*T + 6	ns	(4)
T_{NLNH}	$\overline{WAIT}$ Falling to $\overline{WAIT}$ Rising	N*T ± 4		ns	(4)
T_{NHQX}	Output Data Hold after $\overline{WAIT}$ Rising	(N + 1) * T − 6	(N + 1) * T + 6	ns	(5)
T_{EHTV}	DT/$\overline{R}$ Hold after $\overline{DEN}$ High	T/2 − 6	∞	ns	(6)
T_{TVEL}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Falling	T/2 − 4	T/2 + 4	ns	(7)
	RELATIVE INPUT TIMINGS(7)				
T_{IS5}	$\overline{RESET}$ Input Setup (2x Clock Mode)	6		ns	(14)
T_{IH5}	$\overline{RESET}$ Input Hold (2x Clock Mode)	5		ns	(14)
T_{IS6}	$\overline{DREQ3:0}$ Input Setup	12		ns	(8)
T_{IH6}	$\overline{DREQ3:0}$ Input Hold	7		ns	(8)
T_{IS7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Setup	7		ns	(8)
T_{IH7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Hold	3		ns	(8)
T_{IS8}	$\overline{RESET}$ Input Setup (1x Clock Mode)	3		ns	(15)
T_{IH8}	$\overline{RESET}$ Input Hold (1x Clock Mode)	T/4 + 1		ns	(15)

NOTES:
(1) See **Section 4.5.2, AC Timing Waveforms** for waveforms and definitions.
(2) See Figure 22 for capacitive derating information for output delays and hold times.
(3) See Figure 23 for capacitive derating information for rise and fall times.
(4) Where N is the number of N_{RAD}, N_{RDD}, N_{WAD}, or N_{WDD} wait states that are programmed in the Bus Controller Region Table. When there are no wait states in an access, $\overline{WAIT}$ never goes active.
(5) N = Number of wait states inserted with $\overline{READY}$.
(6) Output Data and/or DT/$\overline{R}$ may be driven indefinitely following a cycle if there is no subsequent bus activity.
(7) See Notes 1, 2 and 3.
(8) Since asynchronous inputs are synchronized internally by the 80960CF they have no required setup or hold times in order to be recognized and for proper operation. However, to guarantee recognition of the input at a particular edge of PCLK2:1 the setup times shown must be met. Asynchronous inputs must be active for at least two consecutive PCLK2:1 rising edges to be seen by the processor.
(9) These specifications are guaranteed by the processor.
(10) These specifications must be met by the system for proper operation of the processor.
(11) This timing is dependent upon the loading of PCLK2:1. Use the derating curves of **Section 4.5.3** to adjust the timing for PCLK2:1 loading.
(12) In the 1-x input clock mode, the maximum input clock period is limited to 125 ns while the processor is operating. When the processor is in reset, the input clock may stop even in 1-x mode.
(13) When in the 1-x input clock mode, these specifications assume a stable input clock with a period variation of less than ±0.1% between adjacent cycles.
(14) In 2x clock mode, $\overline{RESET}$ is an asynchronous input which has no required setup and hold time for proper operation. However, to guarantee the device exits reset synchronized to a particular clock edge, the $\overline{RESET}$ pin must meet setup and hold times to the falling edge of the CLKIN. (See Figure 28a.)
(15) In 1x clock mode, $\overline{RESET}$ is an asynchronous input which has no required setup and hold time for proper operation. However, to guarantee the device exits reset synchronized to a particular clock edge, the $\overline{RESET}$ pin must be deasserted while CLKIN is high and meet setup and hold times to the rising edge of the CLKIN. (See Figure 28b.)

AC Characteristics — 80960CF-25

(80960CF-25 only, under the conditions described in **Section 4.2, Operating Conditions** and **Section 4.5.1, AC Test Conditions.**)

Symbol	Parameter		Min	Max	Units	Notes
INPUT CLOCK(10)						
T_F	CLKIN Frequency		0	50	MHz	(1)
T_C	CLKIN Period	In 1-x Mode (f_{CLK1x})	40	125	ns	(1,12)
		In 2-x Mode (f_{CLK2x})	20	∞	ns	(1)
T_{CS}	CLKIN Period Stability	In 1-x Mode (f_{CLK1x})		±0.1%	Δ	(1,13)
T_{CH}	CLKIN High Time	In 1-x Mode (f_{CLK1x})	8	62.5	ns	(1,12)
		In 2-x Mode (f_{CLK2x})	8	∞	ns	(1)
T_{CL}	CLKIN Low Time	In 1-x Mode (f_{CLK1x})	8	62.5	ns	(1,12)
		In 2-x Mode (f_{CLK2x})	8	∞	ns	(1)
T_{CR}	CLKIN Rise Time		0	6	ns	(1)
T_{CF}	CLKIN Fall Time		0	6	ns	(1)
OUTPUT CLOCKS(9)						
T_{CP}	CLKIN to PCLK2:1 Delay	In 1-x Mode (f_{CLK1x})	−2	2	ns	(1,3,13,14)
		In 2-x Mode (f_{CLK2x})	2	25	ns	(1,3)
T	PCLK2:1 Period	In 1-x Mode (f_{CLK1x})	T_C		ns	(1,13)
		In 2-x Mode (f_{CLK2x})	$2T_C$		ns	(1,3)
T_{PH}	PCLK2:1 High Time		(T/2) − 3	T/2	ns	(1,13)
T_{PL}	PCLK2:1 Low Time		(T/2) − 3	T/2	ns	(1,13)
T_{PR}	PCLK2:1 Rise Time		1	4	ns	(1,3)
T_{PF}	PCLK2:1 Fall Time		1	4	ns	(1,3)
SYNCHRONOUS OUTPUTS(10)						
T_{OV}	Output Valid Delay, Output Hold					(6, 11)
T_{OH}	T_{OV1}, T_{OH1}	A31:2	3	16	ns	
	T_{OV2}, T_{OH2}	$\overline{BE3:0}$	3	18	ns	
	T_{OV3}, T_{OH3}	$\overline{ADS}$	6	20	ns	
	T_{OV4}, T_{OH4}	W/$\overline{R}$	3	20	ns	
	T_{OV5}, T_{OH5}	D/$\overline{C}$,$\overline{SUP}$,$\overline{DMA}$	4	18	ns	
	T_{OV6}, T_{OH6}	$\overline{BLAST}$, $\overline{WAIT}$	5	18	ns	
	T_{OV7}, T_{OH7}	$\overline{DEN}$	3	18	ns	
	T_{OV8}, T_{OH8}	HOLDA, BREQ	4	18	ns	
	T_{OV9}, T_{OH9}	$\overline{LOCK}$	4	18	ns	
	T_{OV10}, T_{OH10}	$\overline{DACK3:0}$	4	20	ns	
	T_{OV11}, T_{OH11}	D31:0	3	18	ns	
	T_{OV12}, T_{OH12}	DT/$\overline{R}$	T/2 + 3	T/2 + 16	ns	
	T_{OV13}, T_{OH13}	$\overline{FAIL}$	2	16	ns	
	T_{OV14}, T_{OH14}	$\overline{EOP3:0}$/$\overline{TC3:0}$	3	20	ns	(6, 11)
T_{OF}	Output Float for all outputs		3	22	ns	(6)
SYNCHRONOUS INPUTS(10)						
T_{IS}	Input Setup					
	T_{IS1}	D31:0	5		ns	(1,11)
	T_{IS2}	$\overline{BOFF}$	19		ns	(1,11)
	T_{IS3}	$\overline{BTERM}$/READY	9		ns	(1,11)
	T_{IS4}	HOLD	9		ns	(1,11)
T_{IH}	Input Hold					
	T_{IH1}	D31:0	5		ns	(1,11)
	T_{IH2}	$\overline{BOFF}$	7		ns	(1,11)
	T_{IH3}	$\overline{BTERM}$/READY	2		ns	(1,11)
	T_{IH4}	HOLD	5		ns	(1,11)

AC Characteristics — 80960CF-25

(80960CF-25 only, under the conditions described in **Section 4.2, Operating Conditions** and **Section 4.5.1, AC Test Conditions.**) (Continued)

Symbol	Parameter	Min	Max	Units	Notes
RELATIVE OUTPUT TIMINGS(9,7)					
T_{AVSH1}	A31:2 Valid to $\overline{ADS}$ Rising	T − 4	T + 4	ns	
T_{AVSH2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, D/$\overline{C}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{ADS}$ Rising	T − 6	T + 6	ns	
T_{AVEL1}	A31:2 Valid to $\overline{DEN}$ Falling	T − 4	T + 4	ns	
T_{AVEL2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, $\overline{INST}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{DEN}$ Falling	T − 6	T + 6	ns	
T_{NLQV}	$\overline{WAIT}$ Falling to Output Data Valid	±6		ns	
T_{DVNH}	Output Data Valid to $\overline{WAIT}$ Rising	N*T − 6	N*T + 6	ns	(4)
T_{NLNH}	$\overline{WAIT}$ Falling to $\overline{WAIT}$ Rising	N*T ± 4		ns	(4)
T_{NHQX}	Output Data Hold after $\overline{WAIT}$ Rising	(N + 1) * T − 6	(N + 1) * T + 6	ns	(5)
T_{EHTV}	DT/$\overline{R}$ Hold after $\overline{DEN}$ High	T/2 − 6	∞	ns	(6)
T_{TVEL}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Falling	T/2 − 4	T/2 + 4	ns	(7)
RELATIVE INPUT TIMINGS(7)					
T_{IS5}	$\overline{RESET}$ Input Setup (2x Clock Mode	8		ns	(14)
T_{IH5}	$\overline{RESET}$ Input Hold (2x Clock Mode)	7		ns	(14)
T_{IS6}	$\overline{DREQ3:0}$ Input Setup	14		ns	(8)
T_{IH6}	$\overline{DREQ3:0}$ Input Hold	9		ns	(8)
T_{IS7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Setup	9		ns	(8)
T_{IH7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Hold	5		ns	(8)
T_{IS8}	$\overline{RESET}$ Input Setup (1x Clock Mode)	3		ns	(15)
T_{IH8}	$\overline{RESET}$ Input Hold (1x Clock Mode)	T/4 + 1		ns	(15)

NOTES:
(1) See **Section 4.5.2, AC Timing Waveforms** for waveforms and definitions.
(2) See Figure 22 for capacitive derating information for output delays and hold times.
(3) See Figure 23 for capacitive derating information for rise and fall times.
(4) Where N is the number of N_{RAD}, N_{RDD}, N_{WAD}, or N_{WDD} wait states that are programmed in the Bus Controller Region Table. When there are no wait states in an access, $\overline{WAIT}$ never goes active.
(5) N = Number of wait states inserted with $\overline{READY}$.
(6) Output Data and/or DT/$\overline{R}$ may be driven indefinitely following a cycle if there is no subsequent bus activity.
(7) See Notes 1, 2 and 3.
(8) Since asynchronous inputs are synchronized internally by the 80960CF they have no required setup or hold times in order to be recognized and for proper operation. However, to guarantee recognition of the input at a particular edge of PCLK2:1 the setup times shown must be met. Asynchronous inputs must be active for at least two consecutive PCLK2:1 rising edges to be seen by the processor.
(9) These specifications are guaranteed by the processor.
(10) These specifications must be met by the system for proper operation of the processor.
(11) This timing is dependent upon the loading of PCLK2:1. Use the derating curves of **Section 4.5.3** to adjust the timing for PCLK2:1 loading.
(12) In the 1-x input clock mode, the maximum input clock period is limited to 125 ns while the processor is operating. When the processor is in reset, the input clock may stop even in 1-x mode.
(13) When in the 1-x input clock mode, these specifications assume a stable input clock with a period variation of less than ±0.1% between adjacent cycles.
(14) In 2x clock mode, $\overline{RESET}$ is an asynchronous input which has no required setup and hold time for proper operation. However, to guarantee the device exits reset synchronized to a particular clock edge, the $\overline{RESET}$ pin must meet setup and hold times to the falling edge of the CLKIN. (See Figure 28a.)
(15) In 1x clock mode, $\overline{RESET}$ is an asynchronous input which has no required setup and hold time for proper operation. However, to guarantee the device exits reset synchronized to a particular clock edge, the $\overline{RESET}$ pin must be deasserted while CLKIN is high and meet setup and hold times to the rising edge of the CLKIN. (See Figure 28b.)

AC Characteristics — 80960CF-16

(80960CF-16 only, under the conditions described in **Section 4.2, Operating Conditions** and **Section 4.5.1, AC Test Conditions.**) (Continued)

Symbol	Parameter		Min	Max	Units	Notes
	INPUT CLOCK(10)					
T_F	CLKIN Frequency		0	32	MHz	(1)
T_C	CLKIN Period	In 1-x Mode (f_{CLK1x})	62.5	125	ns	(1,12)
		In 2-x Mode (f_{CLK2x})	31.25	∞	ns	(1)
T_{CS}	CLKIN Period Stability	In 1-x Mode (f_{CLK1x})		±0.1%	Δ	(1,13)
T_{CH}	CLKIN High Time	In 1-x Mode (f_{CLK1x})	10	62.5	ns	(1,12)
		In 2-x Mode (f_{CLK2x})	10	∞	ns	(1)
T_{CL}	CLKIN Low Time	In 1-x Mode (f_{CLK1x})	10	62.5	ns	(1,12)
		In 2-x Mode (f_{CLK2x})	10	∞	ns	(1)
T_{CR}	CLKIN Rise Time		0	6	ns	(1)
T_{CF}	CLKIN Fall Time		0	6	ns	(1)
	OUTPUT CLOCKS(9)					
T_{CP}	CLKIN to PCLK2:1 Delay	In 1-x Mode (f_{CLK1x})	−2	2	ns	(1,3,13,14)
		In 2-x Mode (f_{CLK2x})	2	25	ns	(1,3)
T	PCLK2:1 Period	In 1-x Mode (f_{CLK1x})	T_C		ns	(1,13)
		In 2-x Mode (f_{CLK2x})	$2T_C$		ns	(1,3)
T_{PH}	PCLK2:1 High Time		(T/2) − 4	T/2	ns	(1,13)
T_{PL}	PCLK2:1 Low Time		(T/2) − 4	T/2	ns	(1,13)
T_{PR}	PCLK2:1 Rise Time		1	4	ns	(1,3)
T_{PF}	PCLK2:1 Fall Time		1	4	ns	(1,3)
	SYNCHRONOUS OUTPUTS(10)					
T_{OV} T_{OH}	Output Valid Delay, Output Hold					(6, 11)
	T_{OV1}, T_{OH1}	A31:2	3	18	ns	
	T_{OV2}, T_{OH2}	$\overline{BE3:0}$	3	20	ns	
	T_{OV3}, T_{OH3}	$\overline{ADS}$	6	22	ns	
	T_{OV4}, T_{OH4}	W/$\overline{R}$	3	22	ns	
	T_{OV5}, T_{OH5}	D/$\overline{C}$, $\overline{SUP}$, $\overline{DMA}$	4	20	ns	
	T_{OV6}, T_{OH6}	$\overline{BLAST}$, $\overline{WAIT}$	5	20	ns	
	T_{OV7}, T_{OH7}	$\overline{DEN}$	3	20	ns	
	T_{OV8}, T_{OH8}	HOLDA, BREQ	4	20	ns	
	T_{OV9}, T_{OH9}	$\overline{LOCK}$	4	20	ns	
	T_{OV10}, T_{OH10}	$\overline{DACK3:0}$	4	22	ns	
	T_{OV11}, T_{OH11}	D31:0	3	20	ns	
	T_{OV12}, T_{OH12}	DT/$\overline{R}$	T/2 + 3	T/2 + 18	ns	
	T_{OV13}, T_{OH13}	$\overline{FAIL}$	2	18	ns	
	T_{OV14}, T_{OH14}	$\overline{EOP3:0}$/$\overline{TC3:0}$	3	22	ns	(6, 11)
T_{OF}	Output Float for all outputs		3	22	ns	(6)
	SYNCHRONOUS INPUTS(10)					
T_{IS}	Input Setup					
	T_{IS1}	D31:0	5		ns	(1,11)
	T_{IS2}	$\overline{BOFF}$	21		ns	(1,11)
	T_{IS3}	$\overline{BTERM}$/$\overline{READY}$	9		ns	(1,11)
	T_{IS4}	HOLD	9		ns	(1,11)
T_{IH}	Input Hold					
	T_{IH1}	D31:0	5		ns	(1,11)
	T_{IH2}	$\overline{BOFF}$	7		ns	(1,11)
	T_{IH3}	$\overline{BTERM}$/$\overline{READY}$	2		ns	(1,11)
	T_{IH4}	HOLD	5		ns	(1,11)

AC Characteristics — 80960CF-16

(80960CF-16 only, under the conditions described in **Section 4.2, Operating Conditions** and **Section 4.5.1, AC Test Conditions.**) (Continued)

Symbol	Parameter	Min	Max	Units	Notes
	RELATIVE OUTPUT TIMINGS(9,7)				
T_{AVSH1}	A31:2 Valid to $\overline{ADS}$ Rising	T − 4	T + 4	ns	
T_{AVSH2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, D/$\overline{C}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{ADS}$ Rising	T − 6	T + 6	ns	
T_{AVEL1}	A31:2 Valid to $\overline{DEN}$ Falling	T − 6	T + 6	ns	
T_{AVEL2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, $\overline{INST}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{DEN}$ Falling	T − 6	T + 6	ns	
T_{NLQV}	$\overline{WAIT}$ Falling to Output Data Valid	±6		ns	
T_{DVNH}	Output Data Valid to $\overline{WAIT}$ Rising	N*T − 6	N*T + 6	ns	(4)
T_{NLNH}	$\overline{WAIT}$ Falling to $\overline{WAIT}$ Rising	N*T ± 4		ns	(4)
T_{NHQX}	Output Data Hold after $\overline{WAIT}$ Rising	(N + 1) * T − 6	(N + 1) * T + 6	ns	(5)
T_{EHTV}	DT/$\overline{R}$ Hold after $\overline{DEN}$ High	T/2 − 6	∞	ns	(6)
T_{TVEL}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Falling	T/2 − 4	T/2 + 4	ns	(7)
	RELATIVE INPUT TIMINGS(7)				
T_{IS5}	$\overline{RESET}$ Input Setup (2x Clock Mode)	10		ns	(14)
T_{IH5}	$\overline{RESET}$ Input Hold (2x Clock Mode)	9		ns	(14)
T_{IS6}	$\overline{DREQ3:0}$ Input Setup	16		ns	(8)
T_{IH6}	$\overline{DREQ3:0}$ Input Hold	11		ns	(8)
T_{IS7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Setup	9		ns	(8)
T_{IH7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Hold	5		ns	(8)
T_{IS8}	$\overline{RESET}$ Input Setup (1x Clock Mode)	3		ns	(15)
T_{IH8}	$\overline{RESET}$ Input Hold (1x Clock Mode)	T/4 + 1		ns	(15)

NOTES:
(1) See **Section 4.5.2, AC Timing Waveforms** for waveforms and definitions.
(2) See Figure 22 for capacitive derating information for output delays and hold times.
(3) See Figure 23 for capacitive derating information for rise and fall times.
(4) Where N is the number of N_{RAD}, N_{RDD}, N_{WAD}, or N_{WDD} wait states that are programmed in the Bus Controller Region Table. When there are no wait states in an access, $\overline{WAIT}$ never goes active.
(5) N = Number of wait state inserted with $\overline{READY}$.
(6) Output Data and/or DT/$\overline{R}$ may be driven indefinitely following a cycle if there is no subsequent bus activity.
(7) See Notes 1, 2 and 3.
(8) Since asynchronous inputs are synchronized internally by the 80960CF they have no required setup or hold times in order to be recognized and for proper operation. However, to guarantee recognition of the input at a particular edge of PCLK2:1 the setup times shown must be met. Asynchronous inputs must be active for at least two consecutive PCLK2:1 rising edges to be seen by the processor.
(9) These specifications are guaranteed by the processor.
(10) These specifications must be met by the system for proper operation of the processor.
(11) This timing is dependent upon the loading of PCLK2:1. Use the derating curves of Figure 22 to adjust the timing for PCLK2:1 loading.
(12) In the 1-x input clock mode, the maximum input clock period is limited to 125 ns while the processor is operating. When the processor is in reset, the input clock may stop even in 1-x mode.
(13) When in the 1-x input clock mode, these specifications assume a stable input clock with a period variation of less than ±0.1% between adjacent cycles.
(14) In 2x clock mode, $\overline{RESET}$ is an asynchronous input which has no required setup and hold time for proper operation. However, to guarantee the device exits reset synchronized to a particular clock edge, the $\overline{RESET}$ pin must meet setup and hold times to the falling edge of the CLKIN. (See Figure 28a.)
(15) In 1x clock mode, $\overline{RESET}$ is an asynchronous input which has no required setup and hold time for proper operation. However, to guarantee the device exits reset synchronized to a particular clock edge, the $\overline{RESET}$ pin must be deasserted while CLKIN is high and meet setup and hold times to the rising edge of the CLKIN. (See Figure 28b.)

4.5.1. AC TEST CONDITIONS

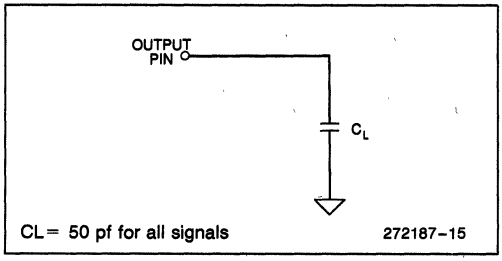


Figure 15. AC Test Load

The AC Specifications in Section 4.5 are tested with the 50 pf load shown in Figure 15. See Figure 22 to see how timings vary with load capacitance.

Specifications are measured at the 1.5V crossing point, unless otherwise indicated. Input waveforms are assumed to have a rise-and-fall time of ≤ 2 ns from 0.8V to 2.0V. See **Section 4.5.2, AC Timing Waveforms** for AC spec definitions, test points and illustrations.

4.5.2. AC TIMING WAVEFORMS

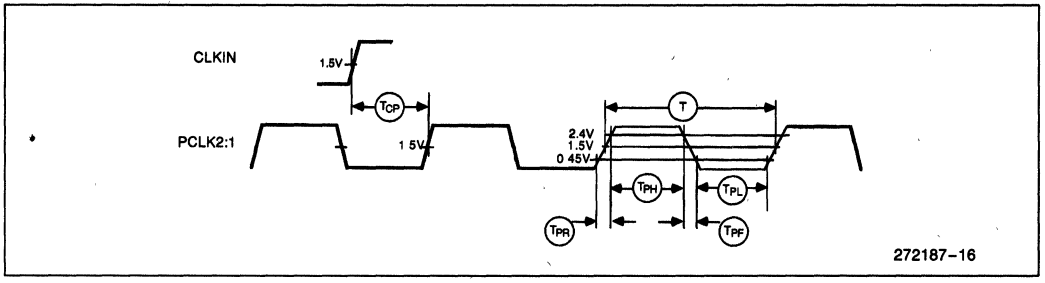


Figure 16a. Input and Output Clocks Waveform

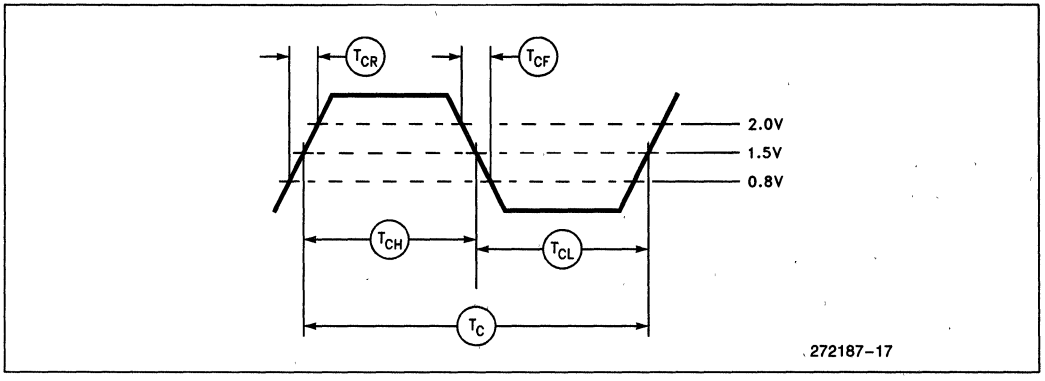


Figure 16b. CLKIN Waveform

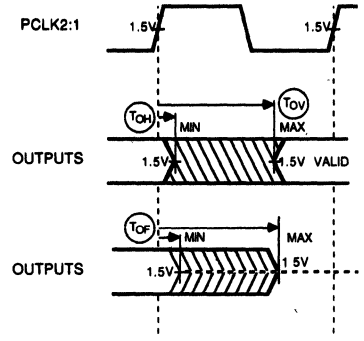


Figure 17. Output Delay and Float Waveform

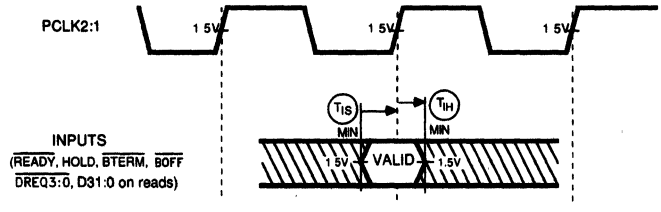


Figure 18a. Input Setup and Hold Waveform

(T_{OV})(T_{OH}) — OUTPUT DELAY — The maximum output delay is referred to as the Output Valid Delay (T_{OV}). The minimum output delay is referred to as the Output Hold (T_{OH}).

(T_{OF}) — OUTPUT FLOAT DELAY — The output float condition occurs when the maximum output current becomes less than I_{LO} in magnitude

(T_{IS})(T_{IH}) — INPUT SETUP AND HOLD — The input setup and hold requirements specify the sampling window during which synchronous inputs must be stable for correct processor operation.

272187-20

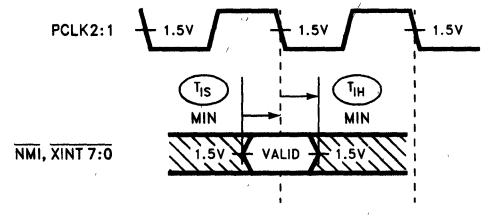


Figure 18b. NMI, XINT7:0 Input Setup and Hold Waveform

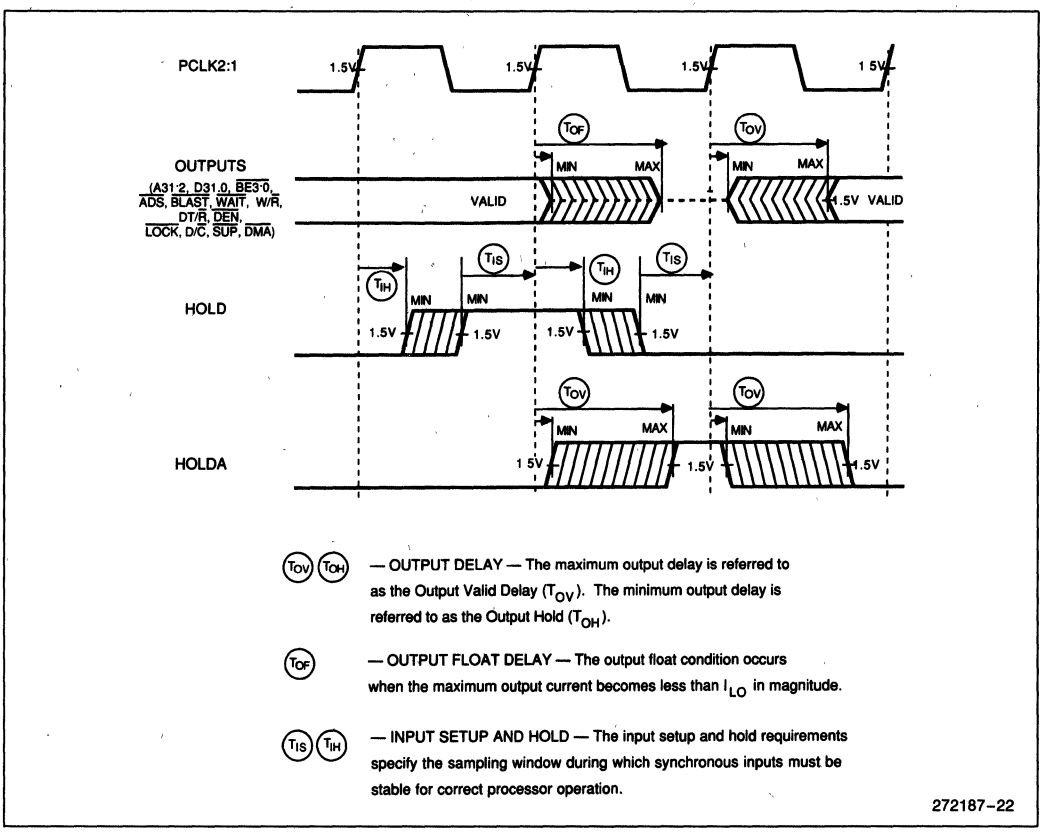


Figure 19. Hold Acknowledge Timings

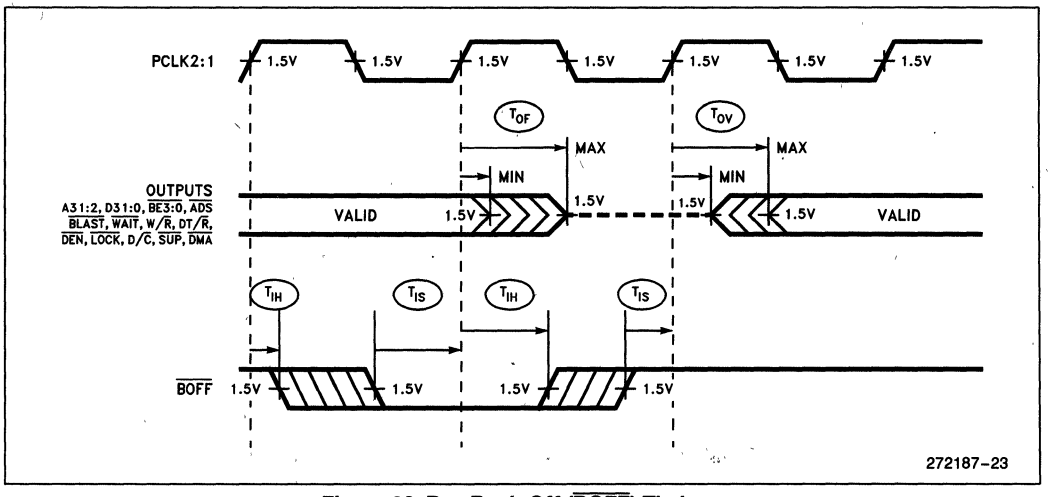


Figure 20. Bus Back-Off ($\overline{BOFF}$) Timings

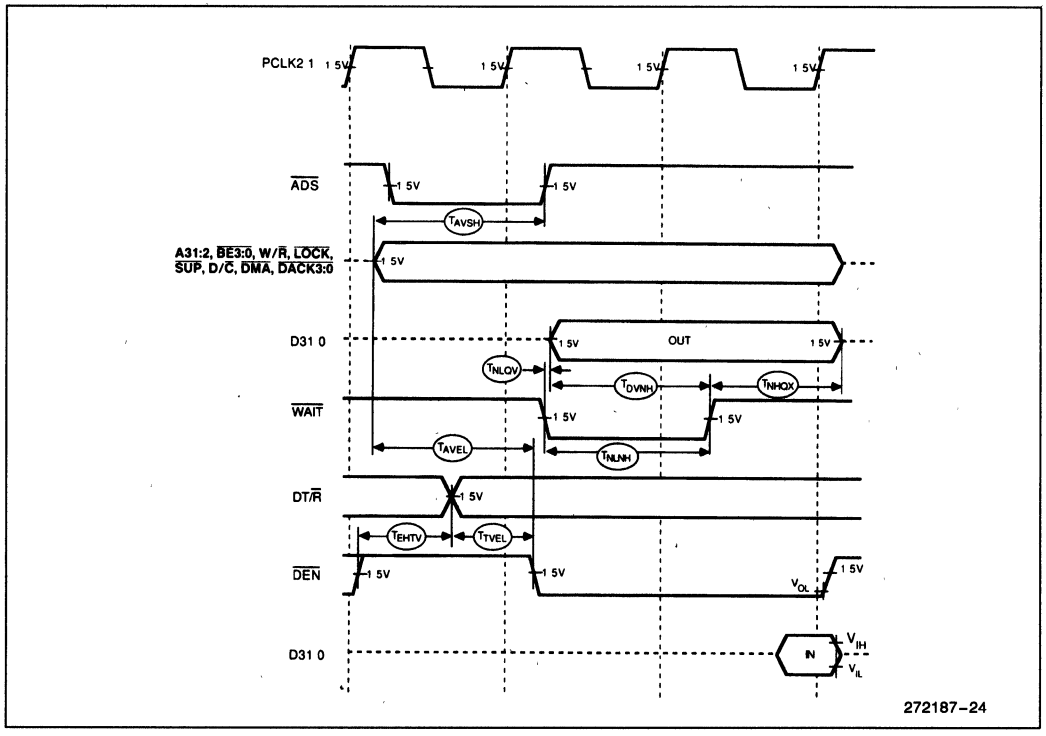
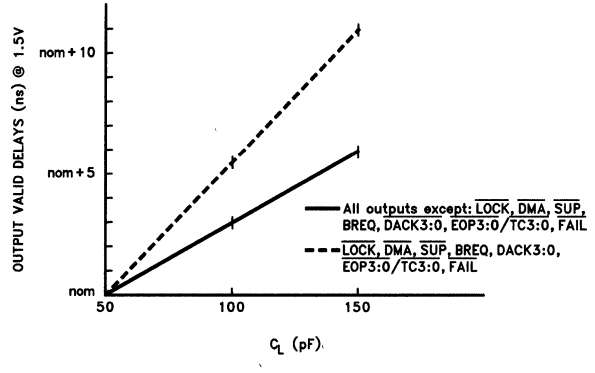


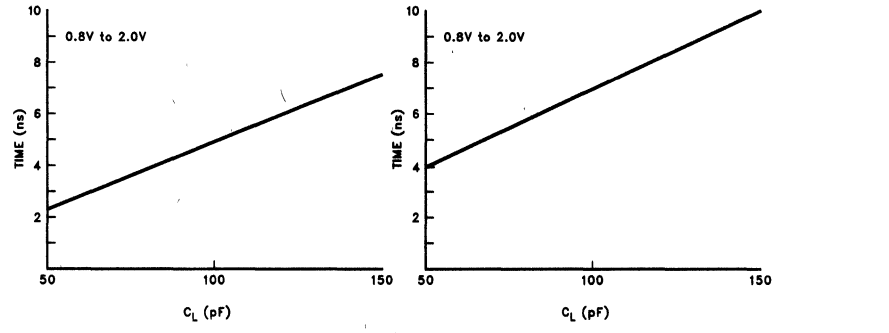
Figure 21. Relative Timings Waveforms

4.5.3 DERATING CURVES



NOTE:
PCLK Load = 50 pF

Figure 22. Output Delay or Hold vs Load Capacitance

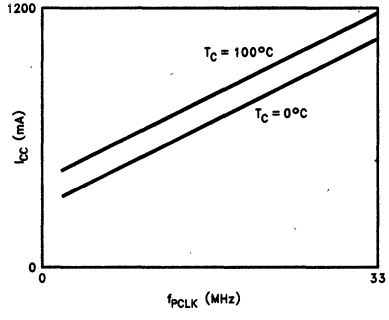


(a) All outputs except: $\overline{LOCK}$, $\overline{DMA}$, $\overline{SUP}$, HOLDA, BREQ, $\overline{DACK3:0}$, $\overline{EOP3:0}$/$\overline{TC3:0}$, $\overline{FAIL}$

(b) $\overline{LOCK}$, $\overline{DMA}$, $\overline{SUP}$, HOLDA, BREQ, $\overline{DACK3:0}$, $\overline{EOP3:0}$/$\overline{TC3:0}$, $\overline{FAIL}$

272187-26

Figure 23. Rise and Fall Time Derating at Highest Operating Temperature and Minimum V_{CC}



272187-27

I_{CC}—I_{CC} under test conditions

Figure 24. I_{CC} vs Frequency and Temperature

5.0 RESET, BACKOFF AND HOLD ACKNOWLEDGE

The following table lists the condition of each processor output pin while RESET is asserted (low).

Table 13. Reset Conditions

Pins	State During Reset (HOLDA inactive)[1]
A31:A2	Floating
D31:D0	Floating
BE3:0	Driven high (Inactive)
W/R	Driven low (Read)
ADS	Driven high (Inactive)
WAIT	Driven high (Inactive)
BLAST	Driven low (Active)
DT/R	Driven low (Receive)
DEN	Driven high (Inactive)
LOCK	Driven high (Inactive)
BREQ	Driven low (Inactive)
D/C	Floating
DMA	Floating
SUP	Floating
FAIL	Driven low (Active)
DACK3	Driven high (Inactive)
DACK2	Driven high (Inactive)
DACK1	Driven high (Inactive)
DACK0	Driven high (Inactive)
EOP/TC3	Floating (set to input mode)
EOP/TC2	Floating (set to input mode)
EOP/TC1	Floating (set to input mode)
EOP/TC0	Floating (set to input mode)

NOTE:
(1) With regard to bus output pin state only, the Hold Acknowledge state takes precedence over the reset state. Although asserting the RESET pin will internally reset the processor, the processor's bus output pins will not enter the reset state if it has granted Hold Acknowledge to a previous HOLD request (HOLDA is active). Furthermore, the processor will grant new HOLD requests and enter the Hold Acknowledge state even while in reset.
For example, if HOLDA is not active and the processor is in the reset state, then HOLD is asserted, the processor's bus pins will enter the Hold Acknowledge state and HOLDA will be granted. The processor will not be able to perform memory accesses until the HOLD request is removed, even if the RESET pin is brought high. This operation is provided to simplify boot-up synchronization among multiple processors sharing the same bus.

The following table lists the condition of each processor output pin while HOLDA is asserted (low).

Table 14. Hold Acknowledge and Backoff Conditions

Pins	State During HOLDA
A31:A2	Floating
D31:D0	Floating
BE3:0	Floating
W/R	Floating
ADS	Floating
WAIT	Floating
BLAST	Floating
DT/R	Floating
DEN	Floating
LOCK	Floating
BREQ	Driven (high or low)
D/C	Floating
DMA	Floating
SUP	Floating
FAIL	Driven high (Inactive)
DACK3	Driven high (Inactive)
DACK2	Driven high (Inactive)
DACK1	Driven high (Inactive)
DACK0	Driven high (Inactive)
EOP/TC3	Driven if output
EOP/TC2	Driven if output
EOP/TC1	Driven if output
EOP/TC0	Driven if output

6.0 BUS WAVEFORMS

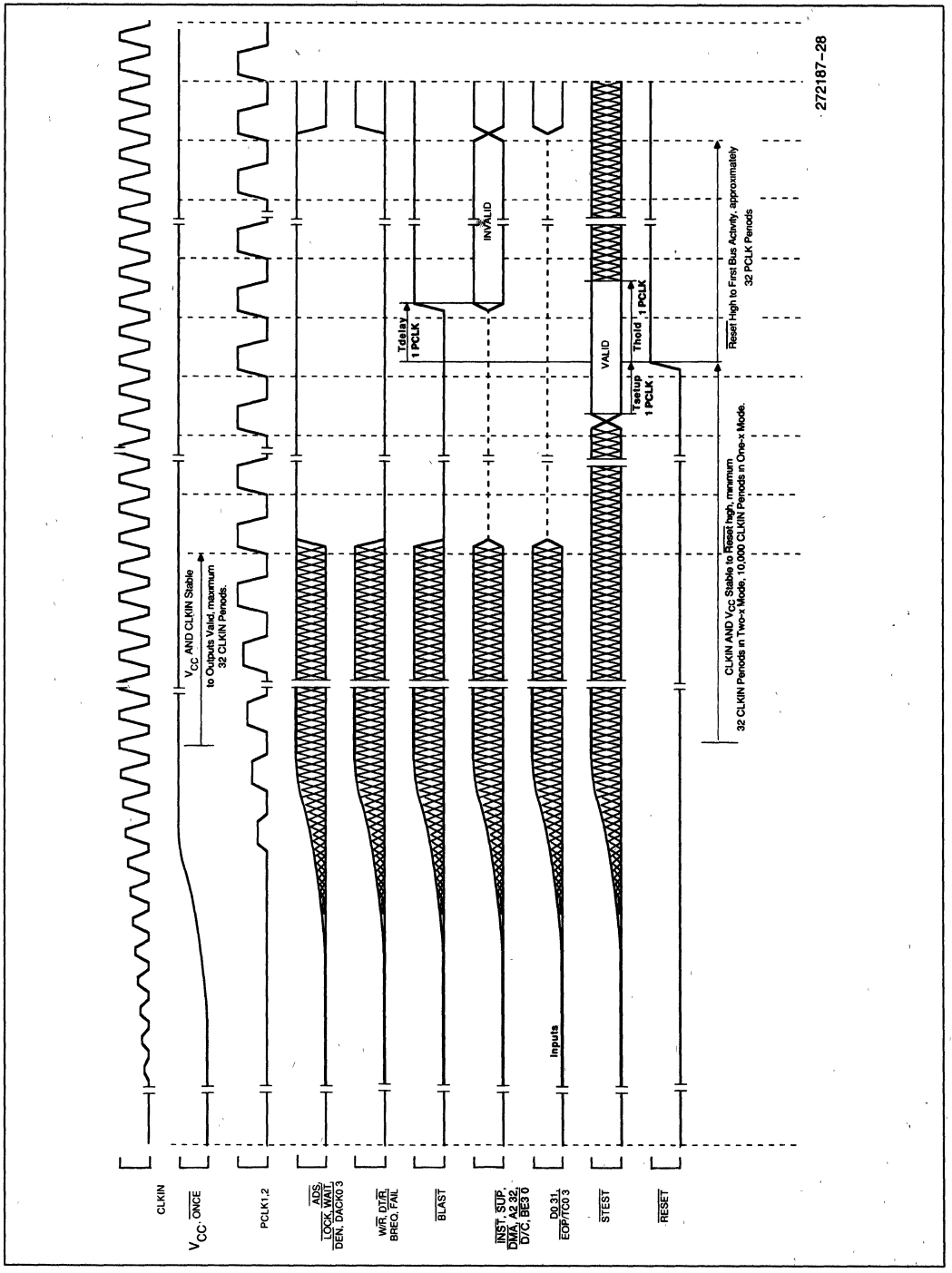


Figure 25. Cold Reset Waveform

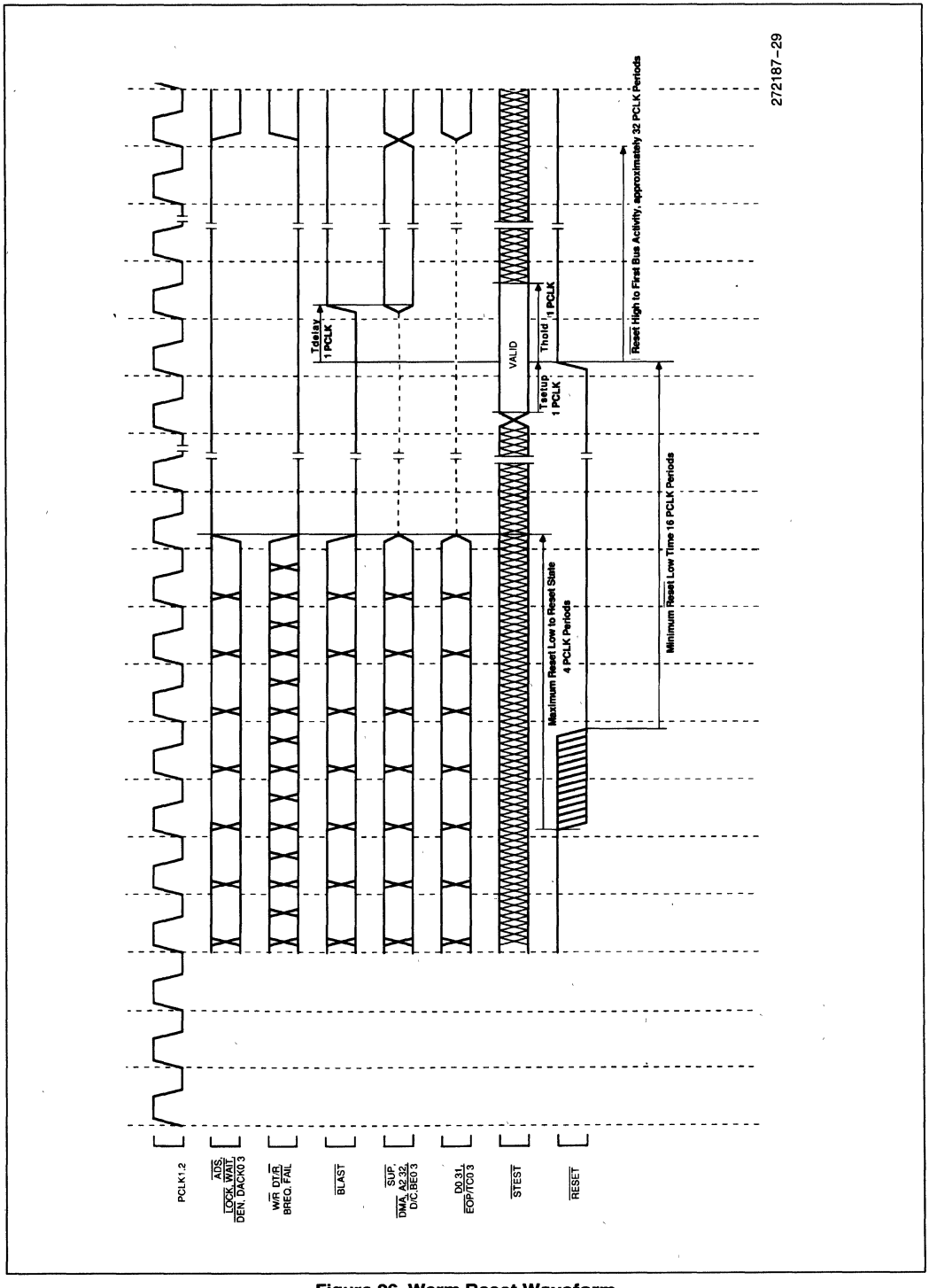


Figure 26. Warm Reset Waveform

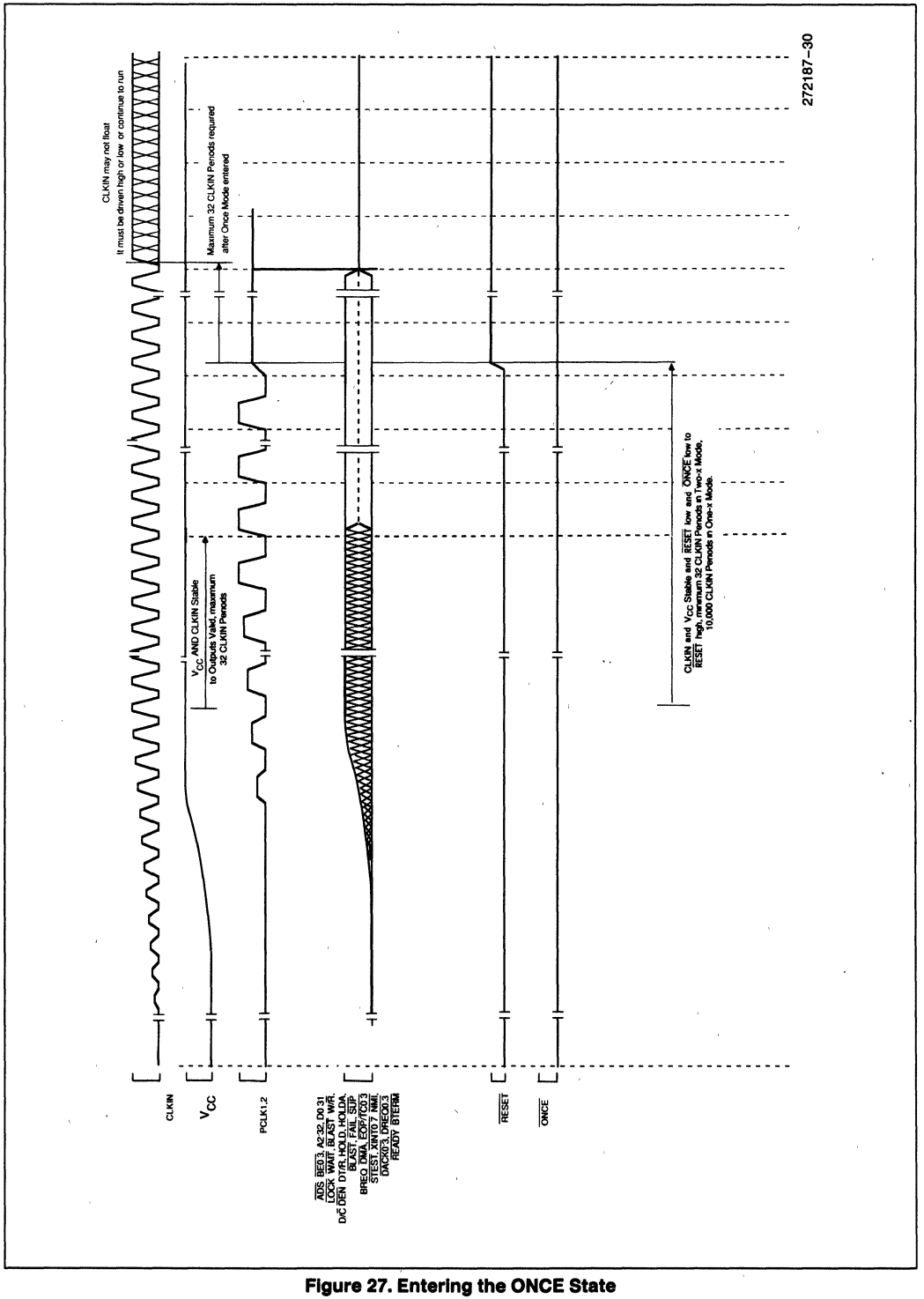
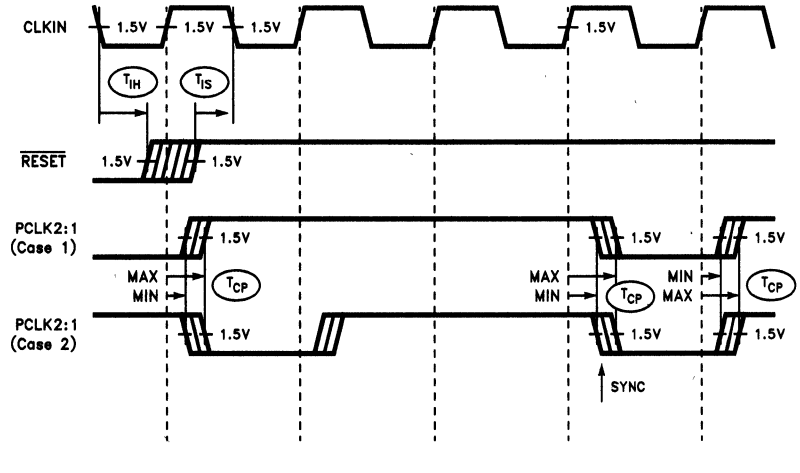
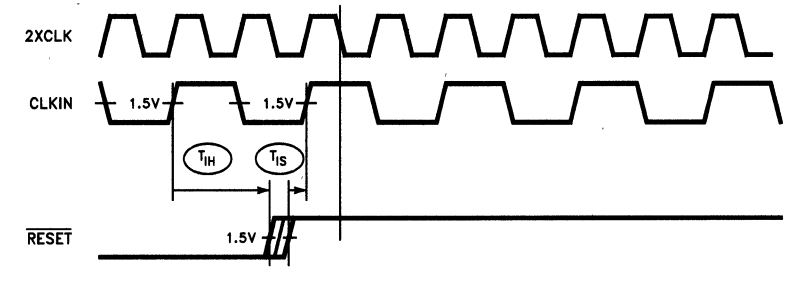


Figure 27. Entering the ONCE State



NOTE:
Case 1 and Case 2 show two possible polarities of PCLK2:1.

Figure 28a. Clock Synchronization in the 2x Clock Mode



NOTE:
In 1x clock mode, the RESET pin is actually sampled on the falling edge of 2XCLK. 2XCLK is an internal signal generated by the PLL and is not available on an external pin. Therefore, RESET is specified relative to the rising edge of CLKIN. The RESET pin is sampled when PCLK is high.

Figure 28b. Clock Synchronization in the 1x Clock Mode

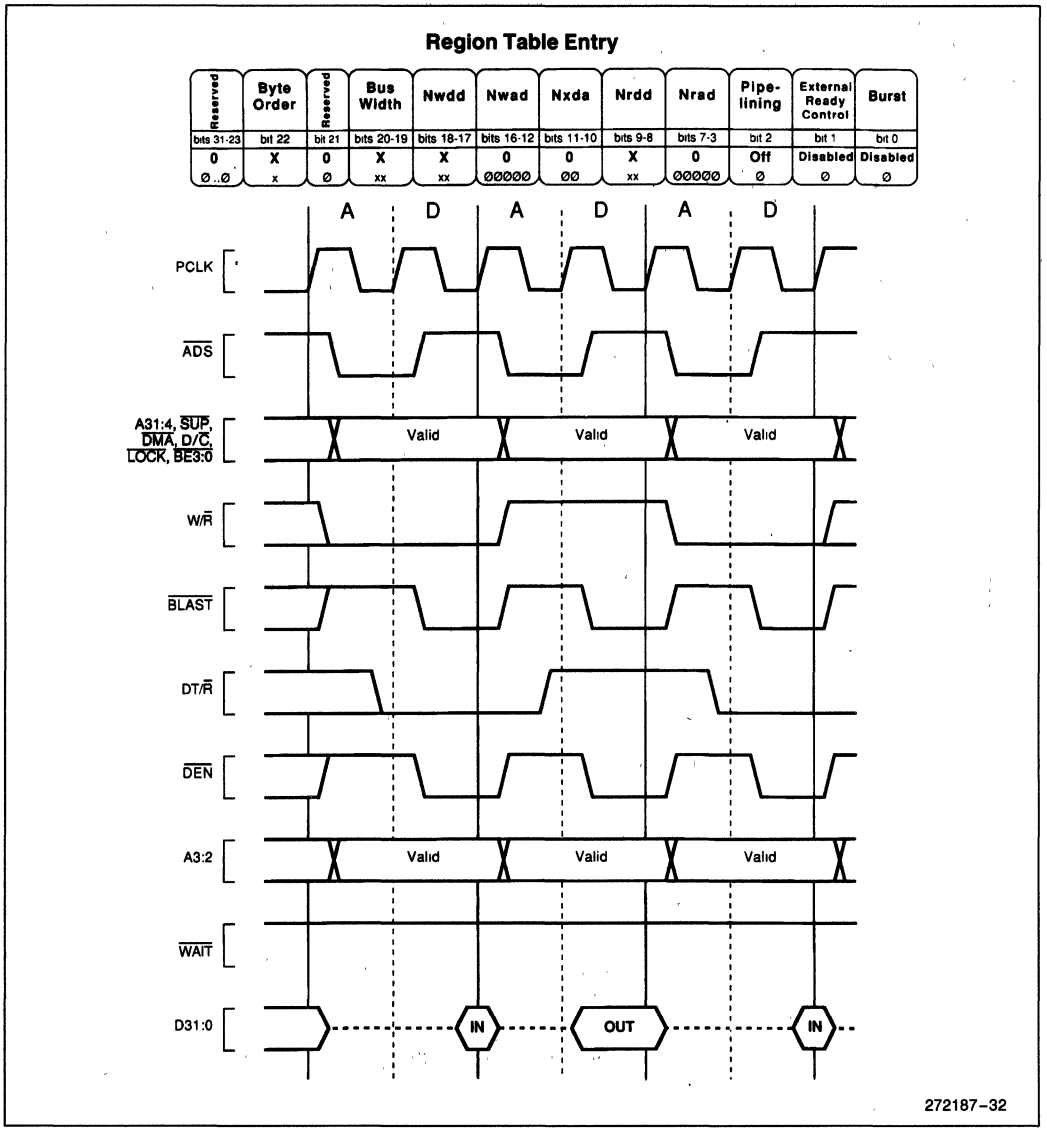


Figure 29. Non-Burst, Non-Pipelined Requests without Wait States

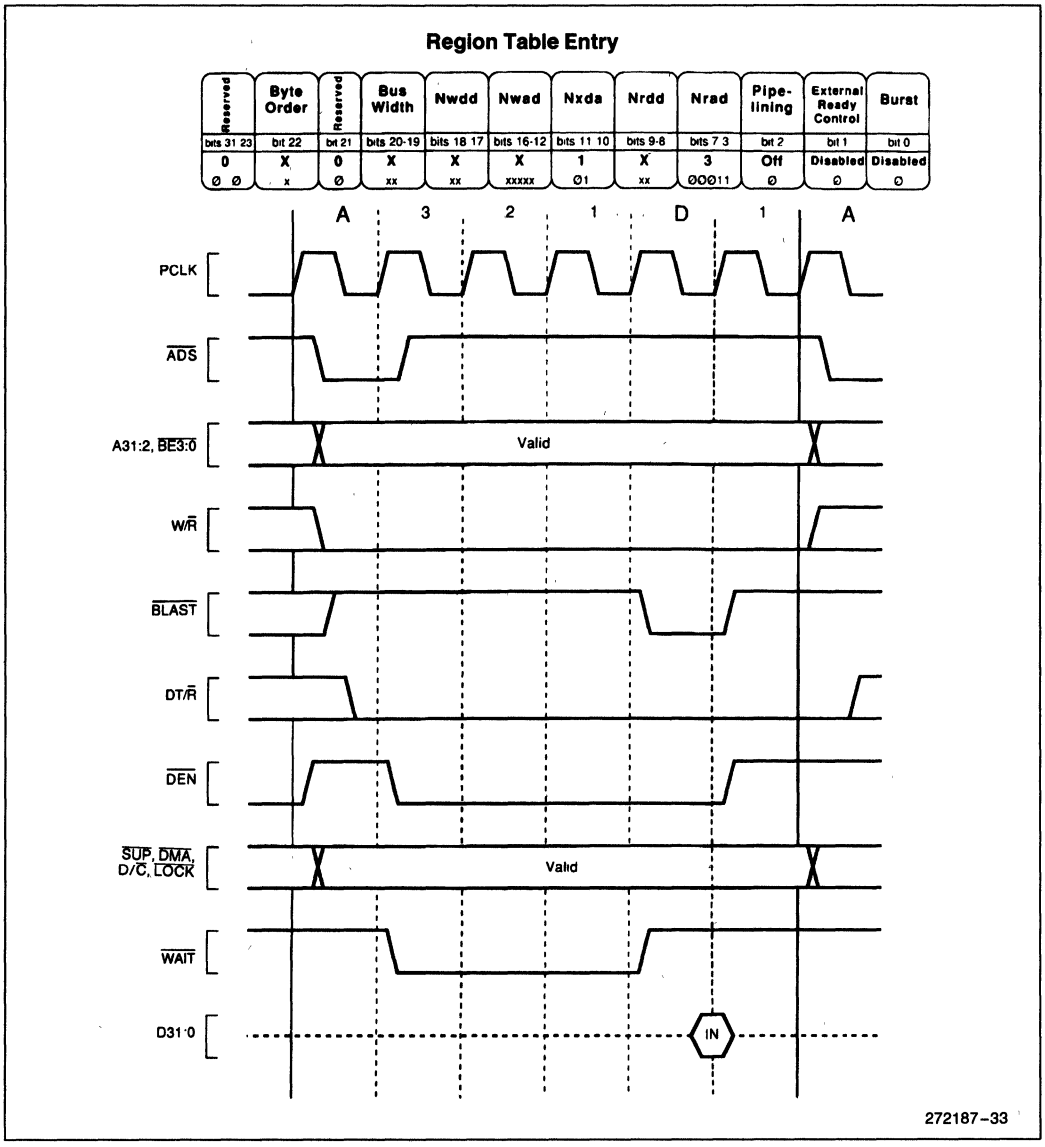


Figure 30. Non-Burst, Non-Pipelined Read Request with Wait States

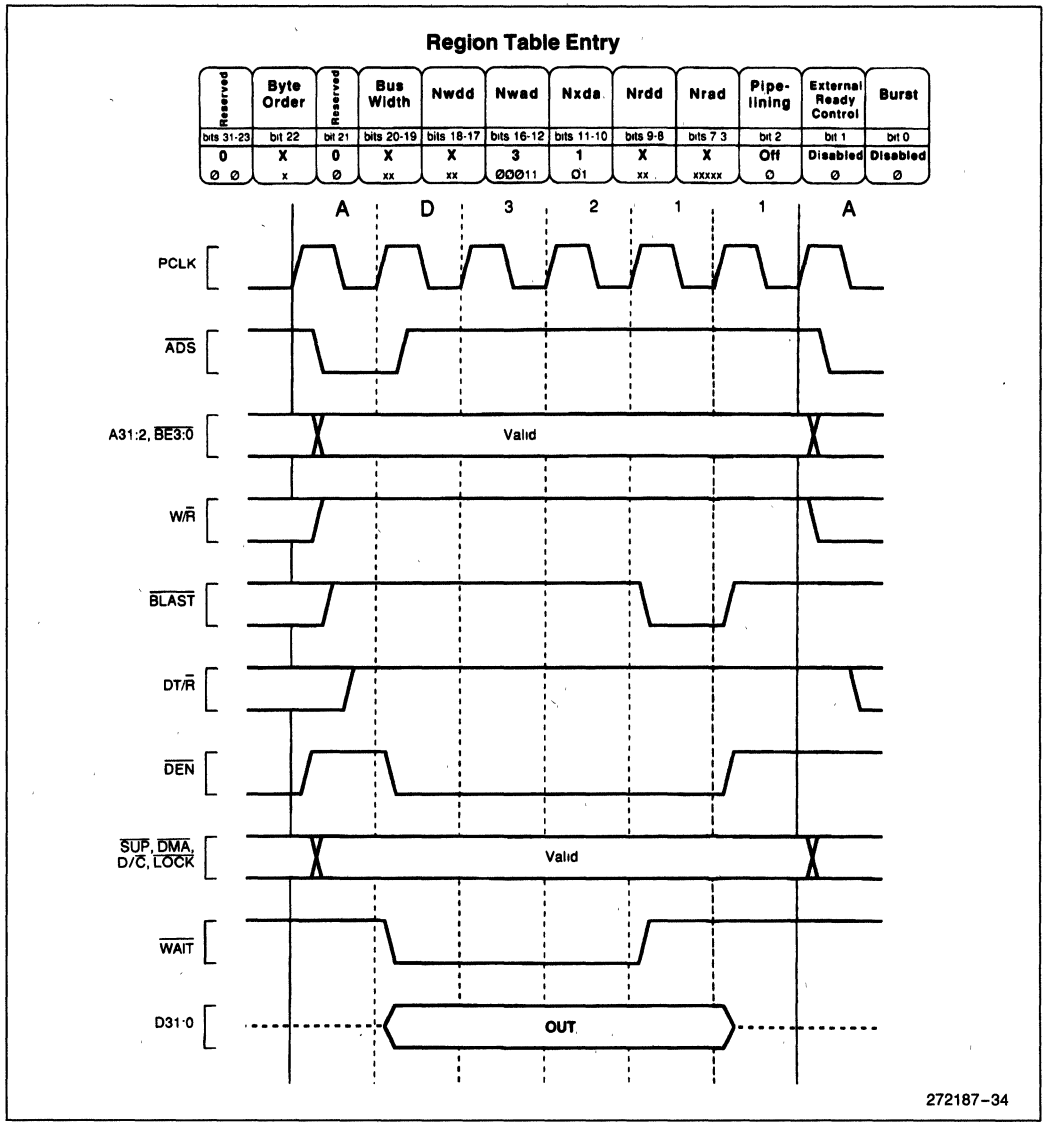


Figure 31. Non-Burst, Non-Pipelined Write Request with Wait States

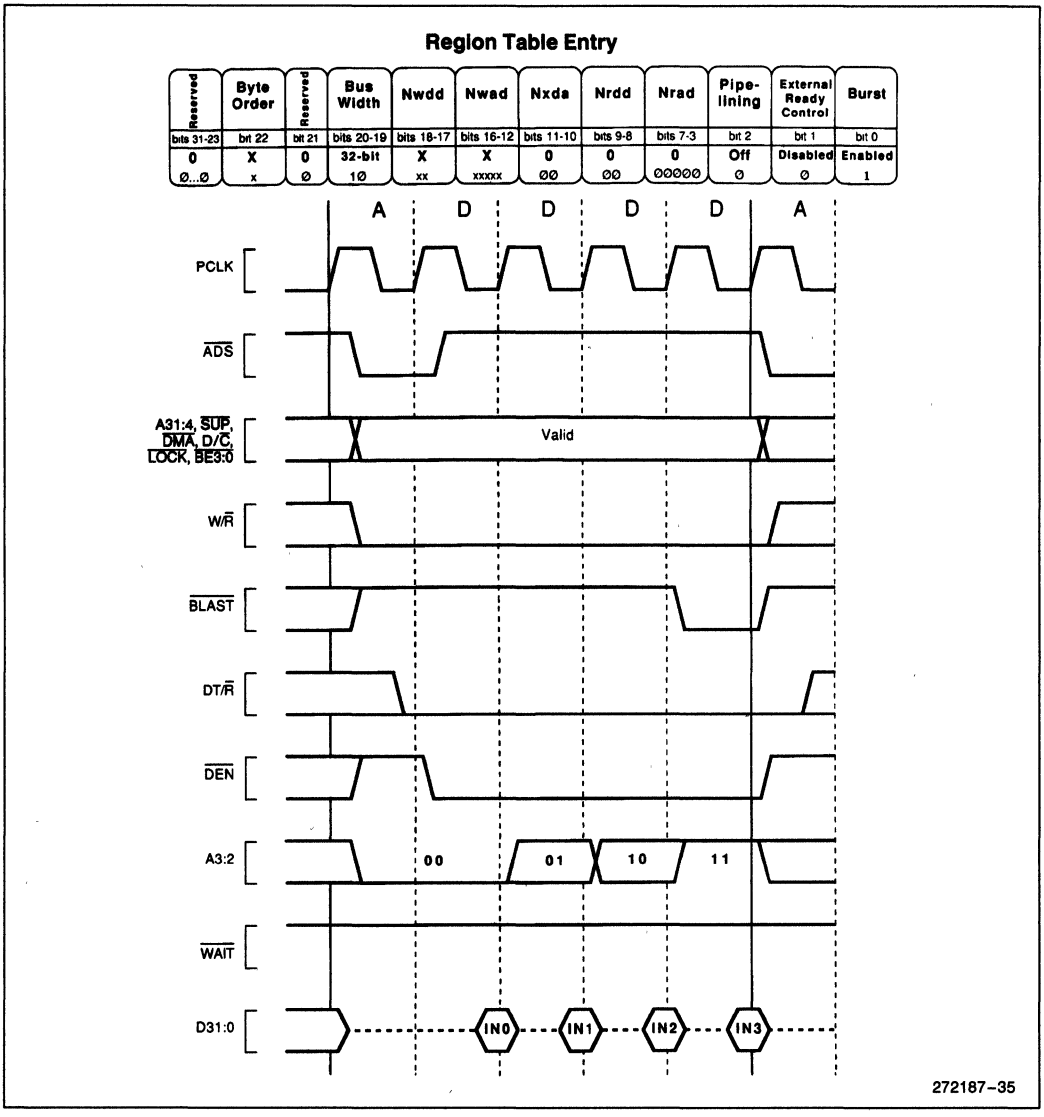


Figure 32. Burst, Non-Pipelined Read Request without Wait States, 32-Bit Bus

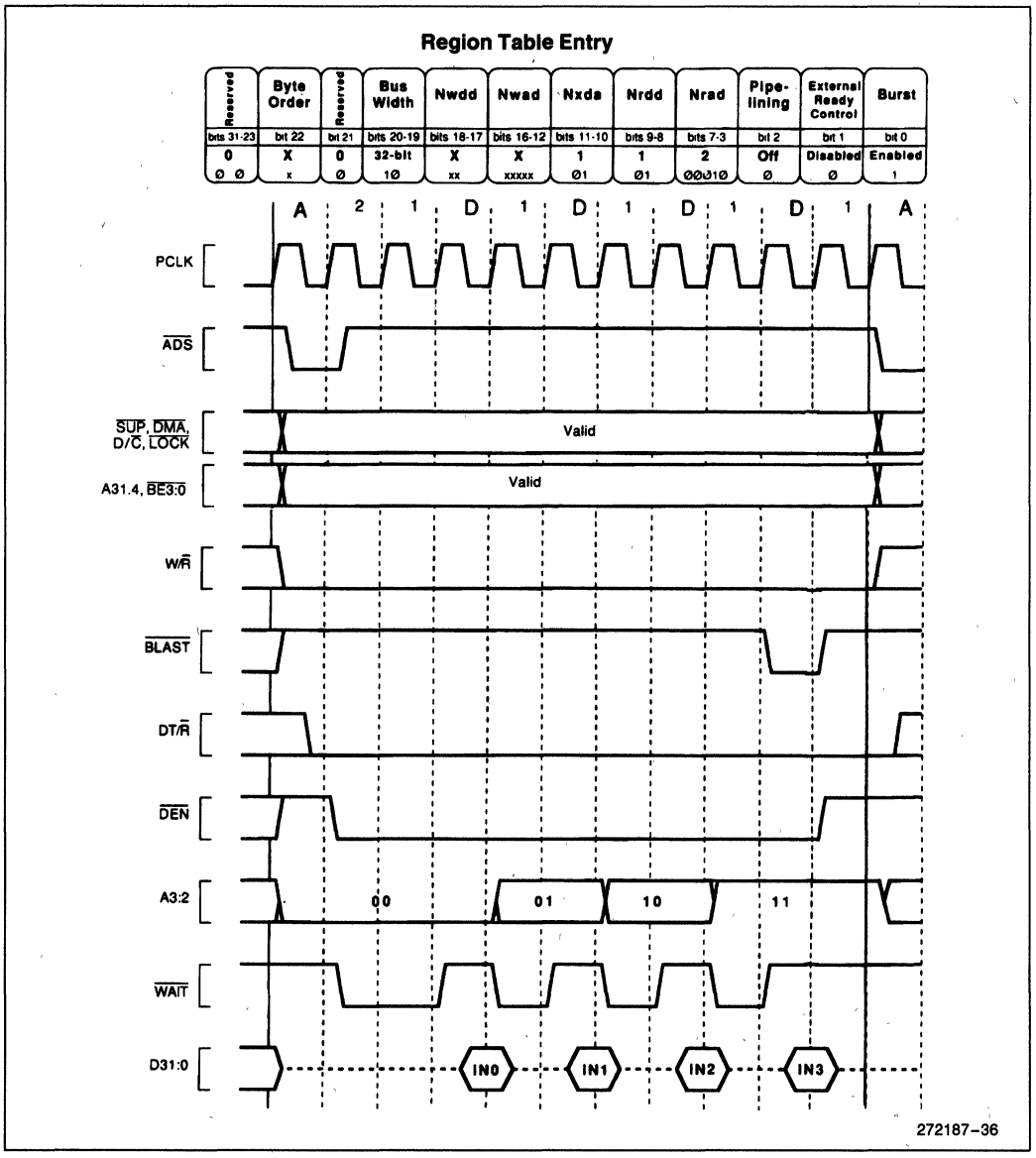


Figure 33. Burst, Non-Pipelined Read Request with Wait States, 32-Bit Bus

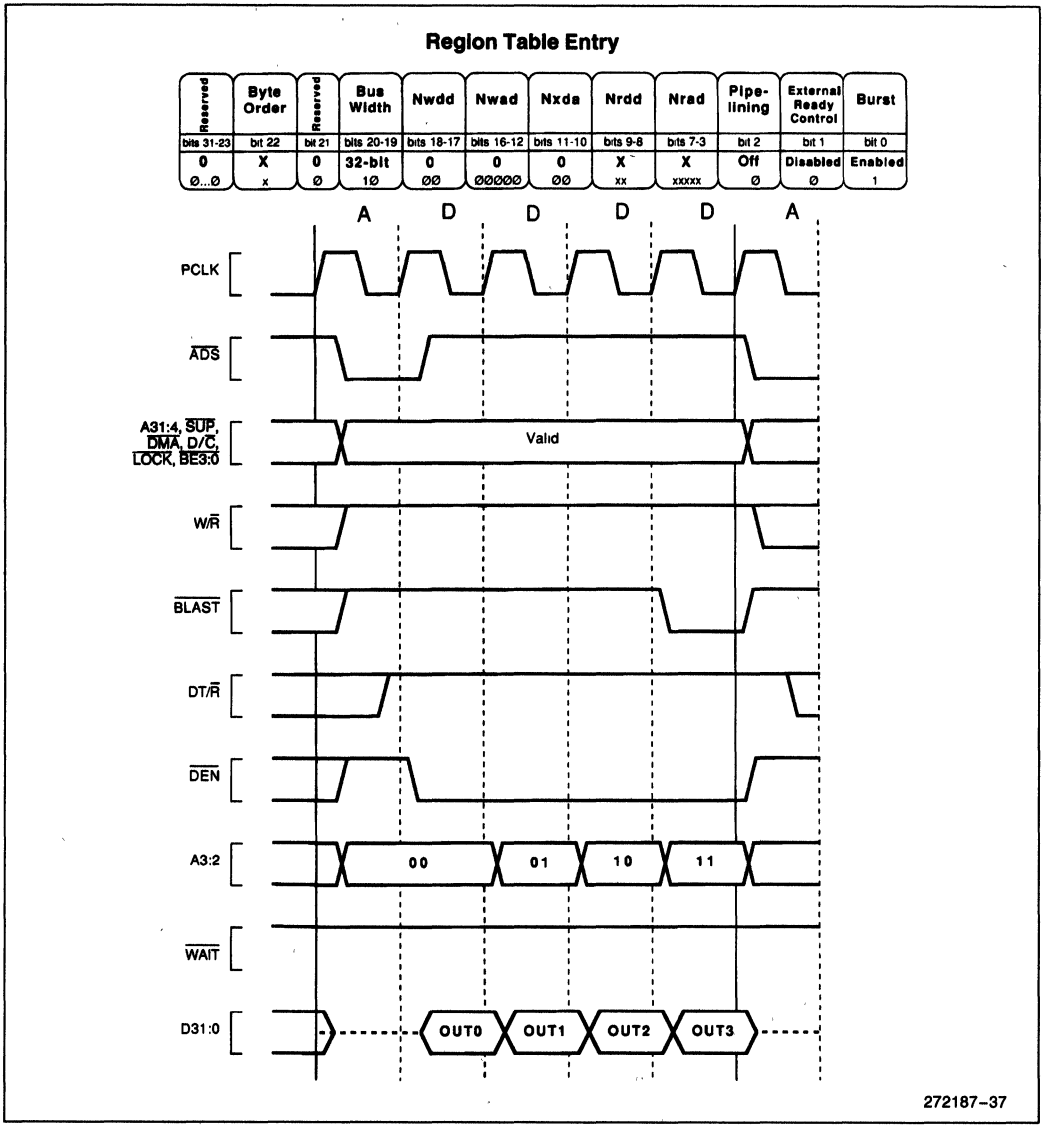
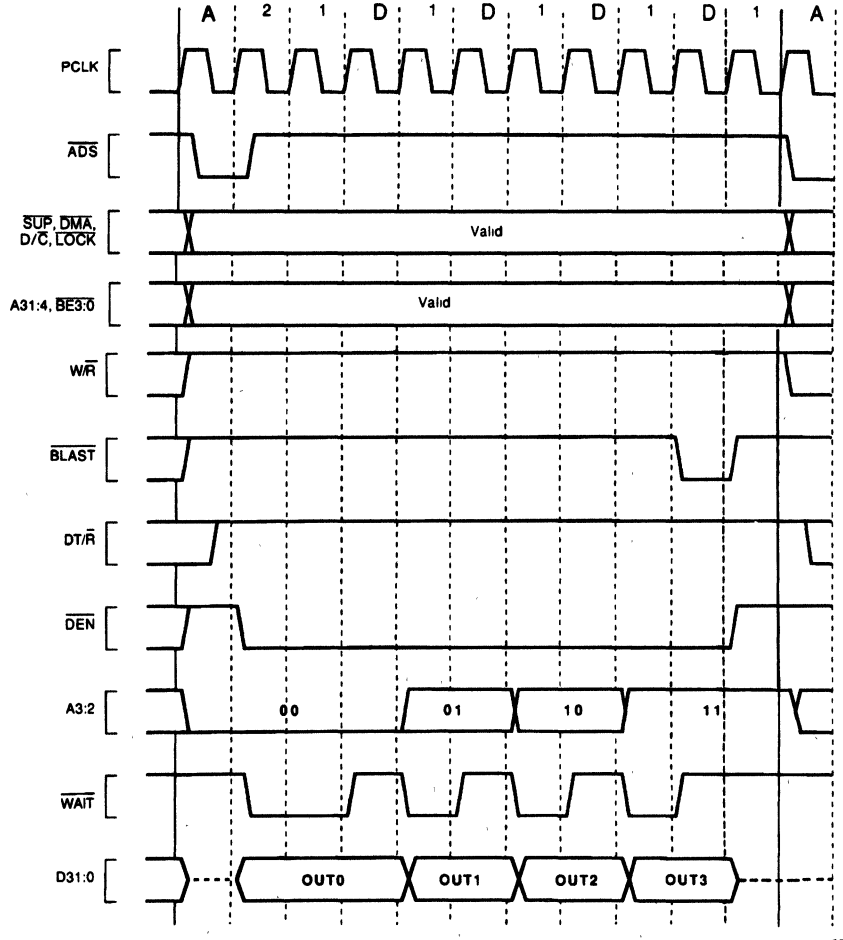


Figure 34. Burst, Non-Pipelined Write Request without Wait States, 32-Bit Bus

Region Table Entry

Reserved	Byte Order	Reserved	Bus Width	Nwdd	Nwad	Nxda	Nrdd	Nrad	Pipe-lining	External Ready Control	Burst
bits 31-23	bit 22	bit 21	bits 20-19	bits 18-17	bits 16-12	bits 11-10	bits 9 8	bits 7 3	bit 2	bit 1	bit 0
0	X	0	32-bit	1	2	1	X	X	Off	Disabled	Enabled
Ø Ø	x	Ø	1Ø	Ø1	ØØ1Ø	Ø1	xx	xxxxx	Ø	Ø	1



272187-38

Figure 35. Burst, Non-Pipelined Write Request with Wait States, 32-Bit Bus

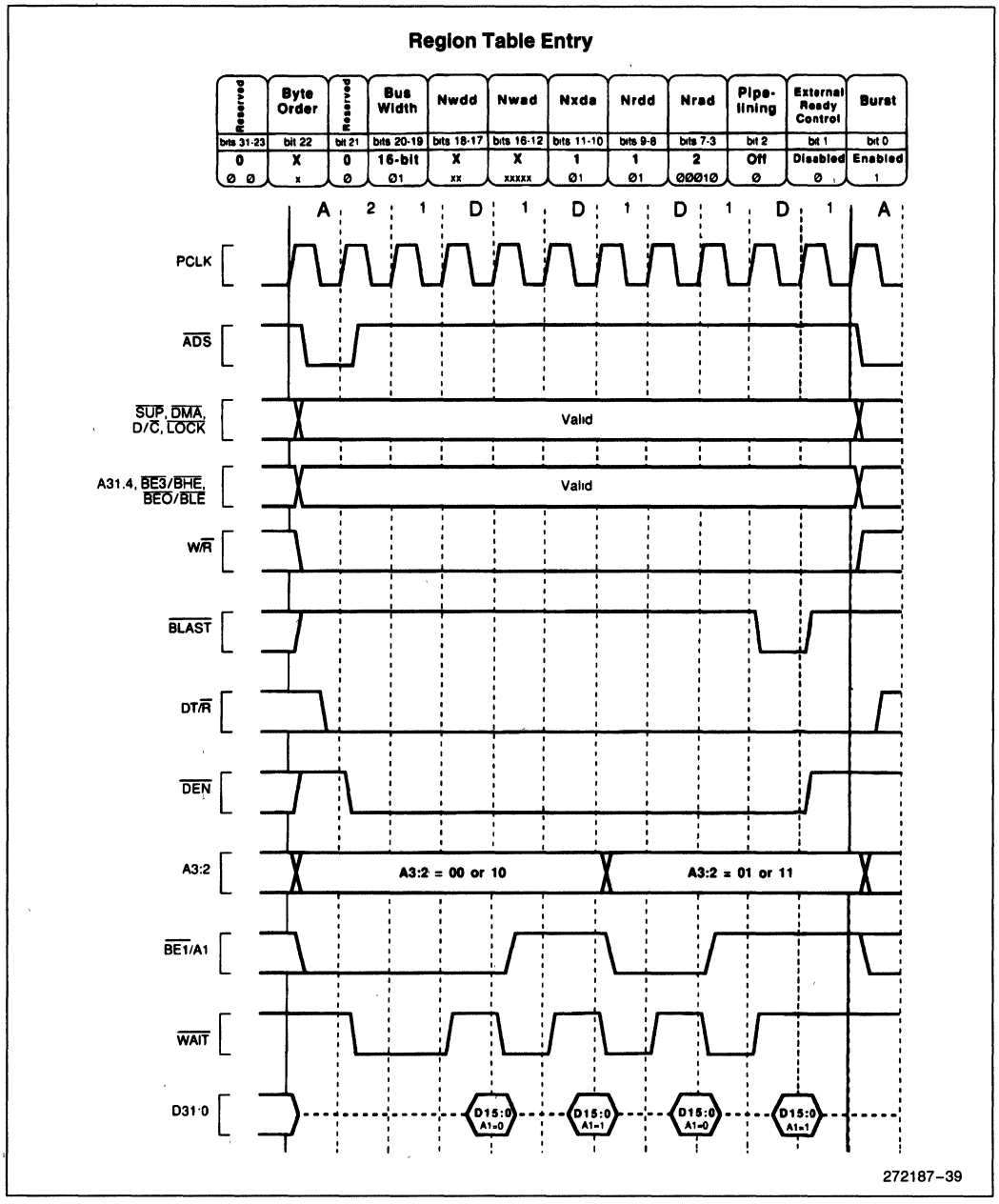


Figure 36. Burst, Non-Pipelined Read Request with Wait States, 16-Bit Bus

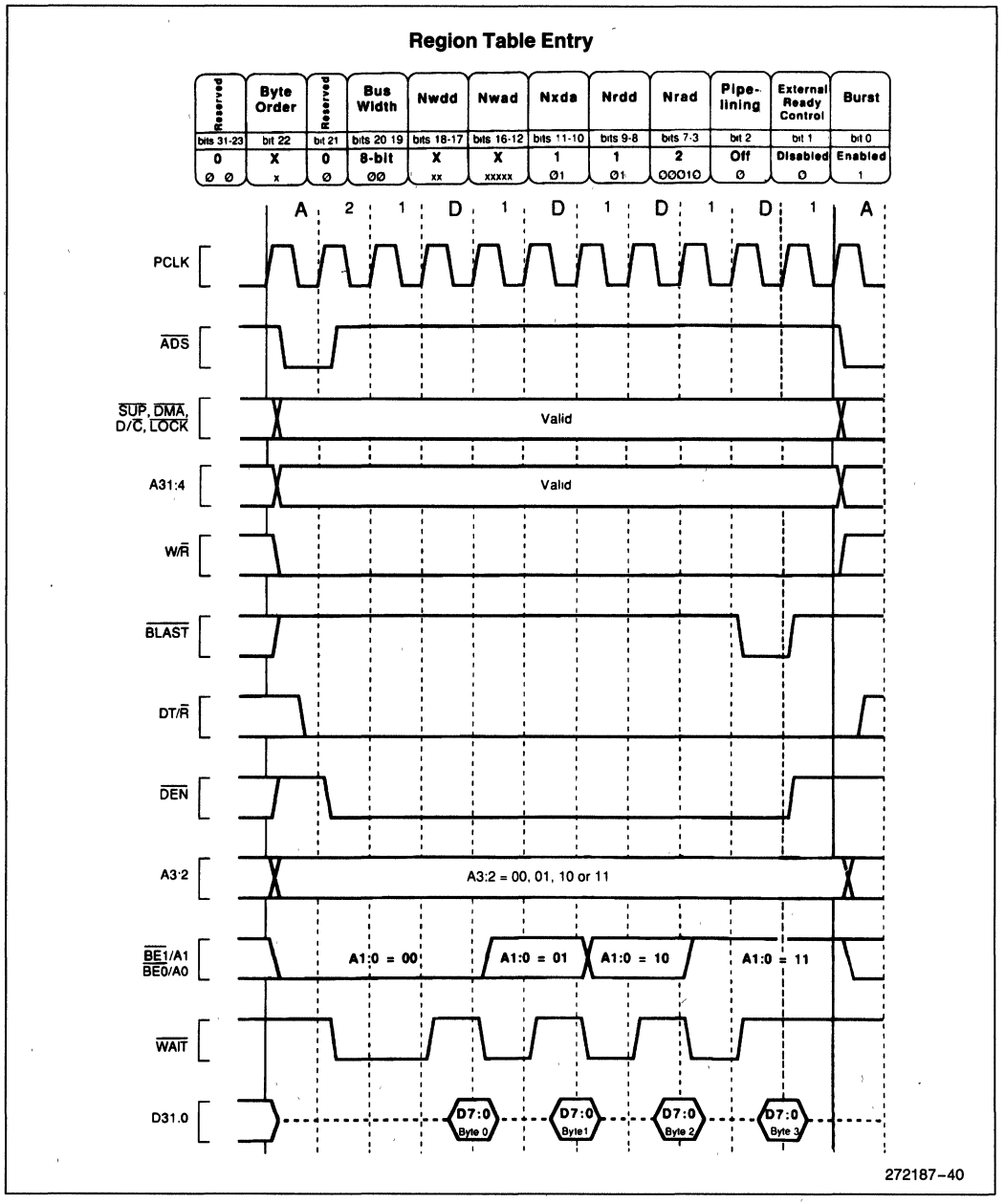


Figure 37. Burst, Non-Pipelined Read Request with Wait States, 8-Bit Bus

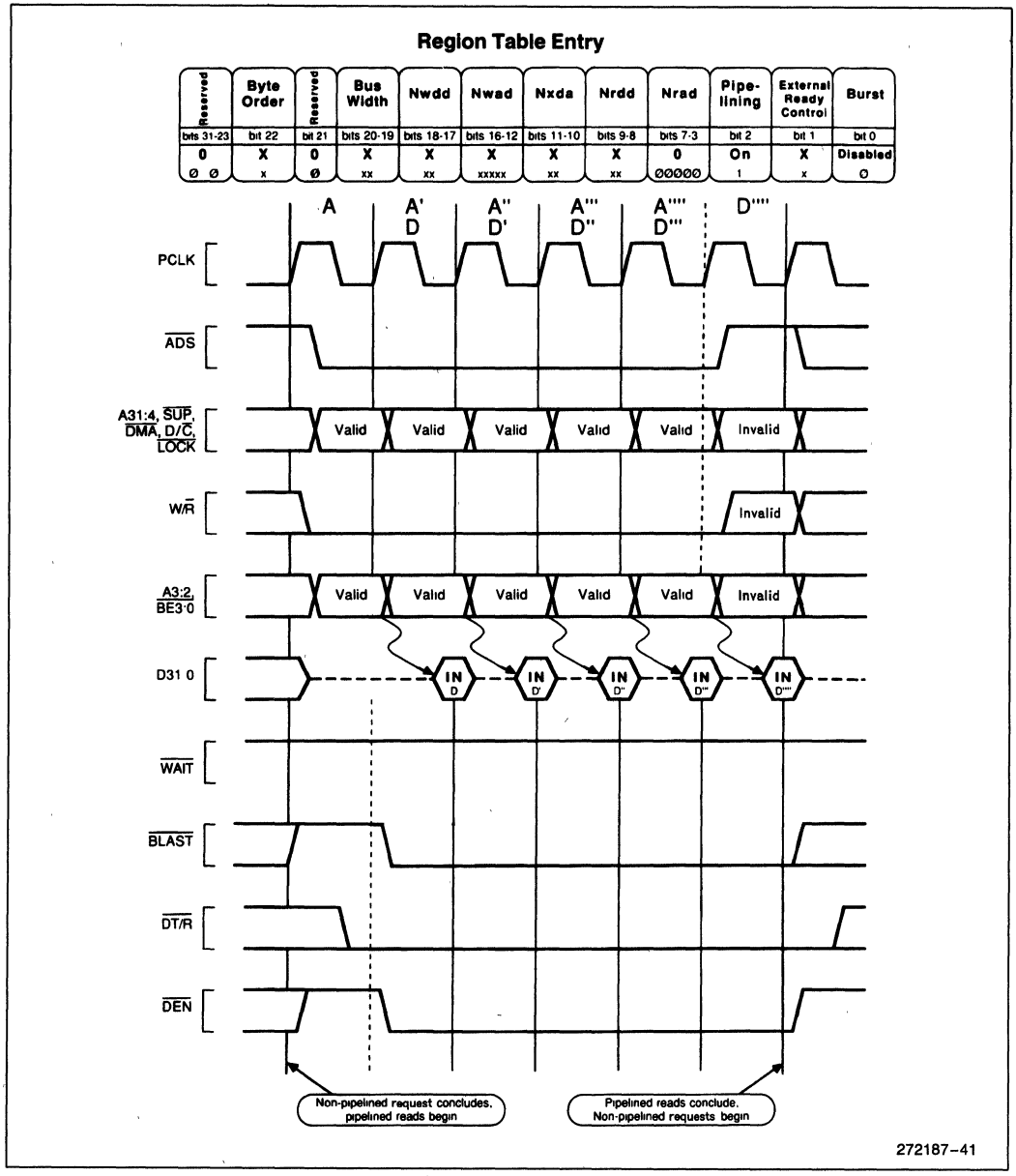


Figure 38. Non-Burst, Pipelined Read Request without Wait States, 32-Bit Bus

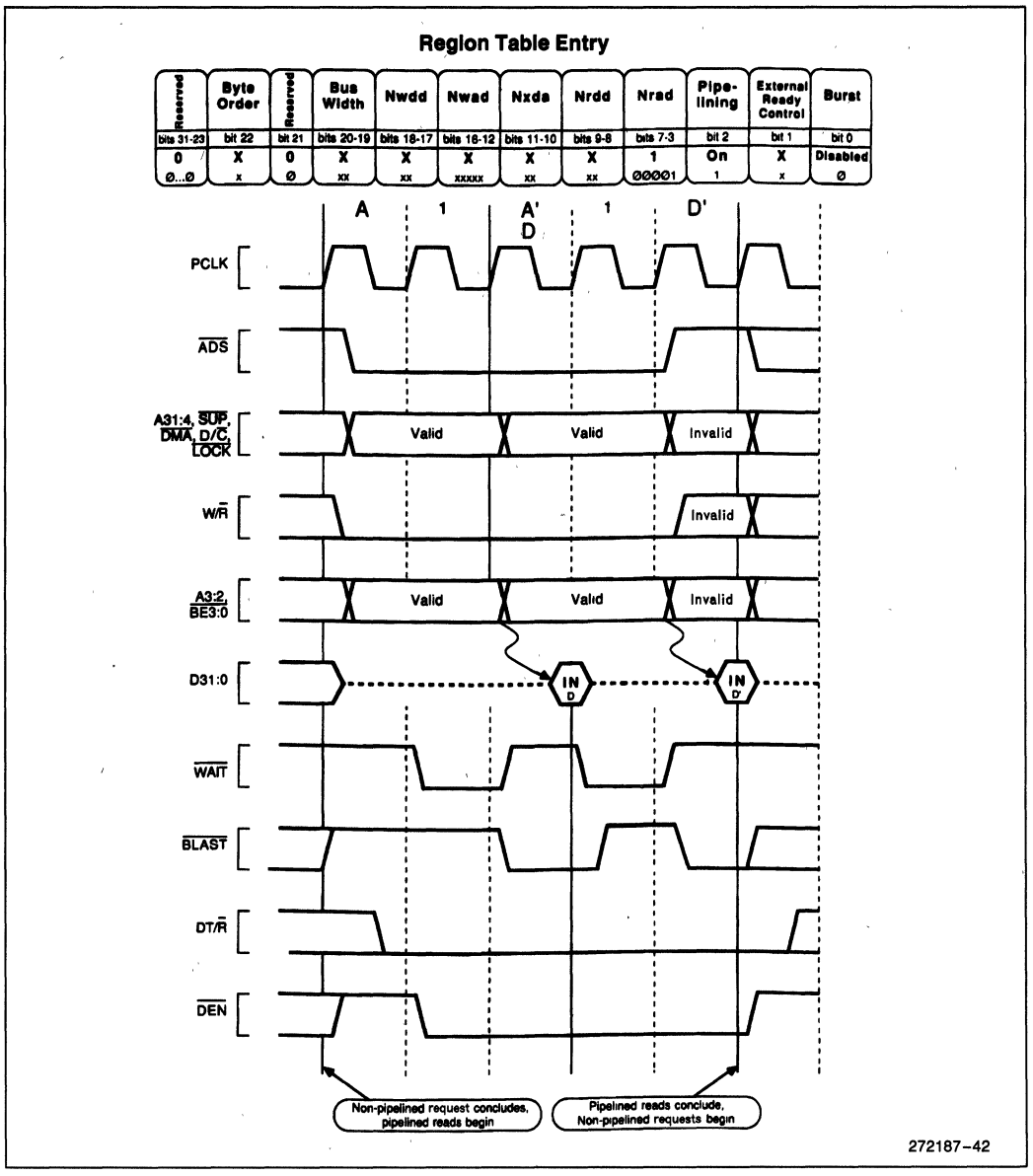


Figure 39. Non-Burst, Pipelined Read Request with Wait States, 32-Bit Bus

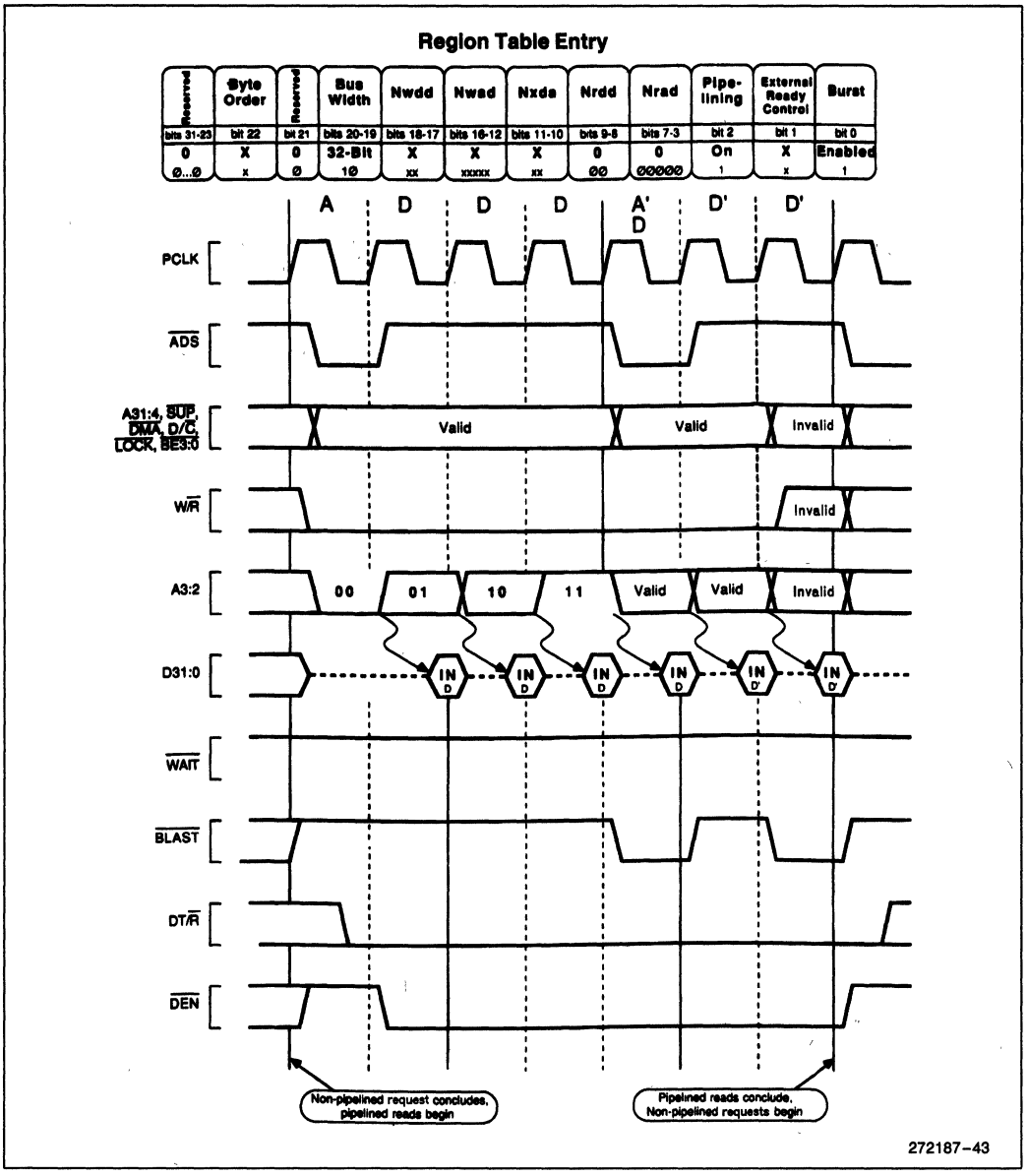


Figure 40. Burst, Pipelined Read Request without Wait States, 32-Bit Bus

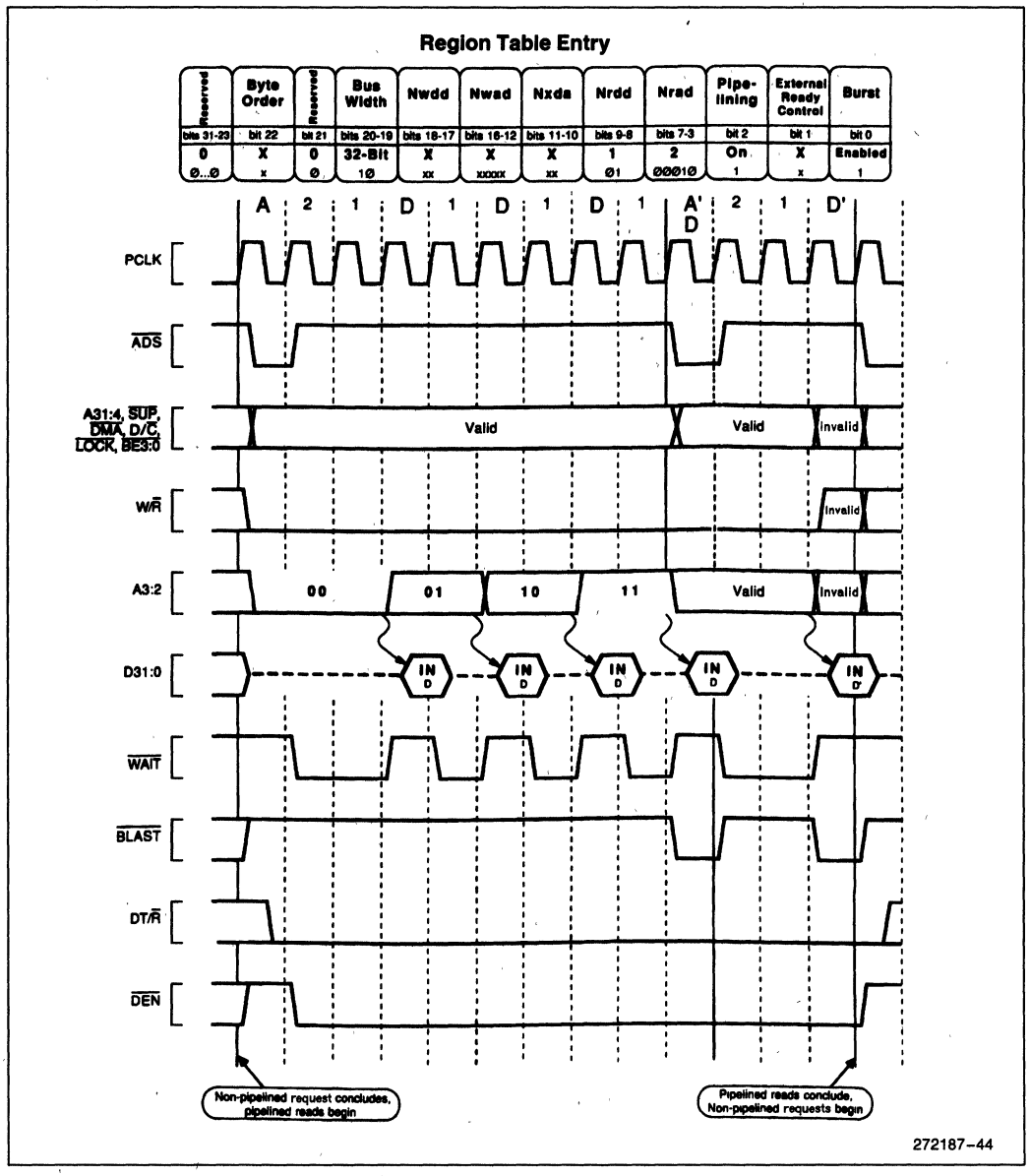


Figure 41. Burst, Pipelined Read Requests with Wait States, 32-Bit Bus

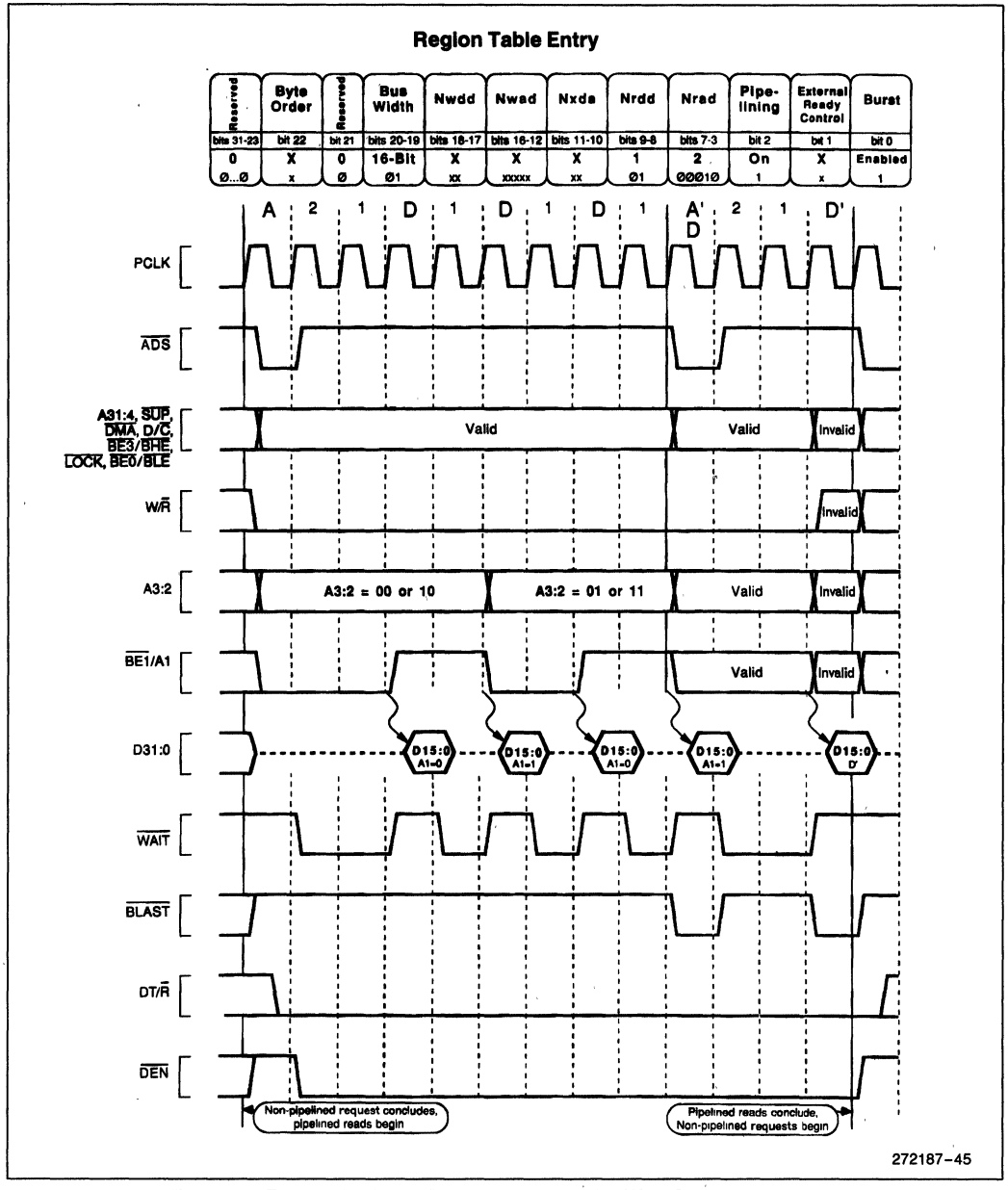


Figure 42. Burst, Pipelined Read Requests with Wait States, 16-Bit Bus

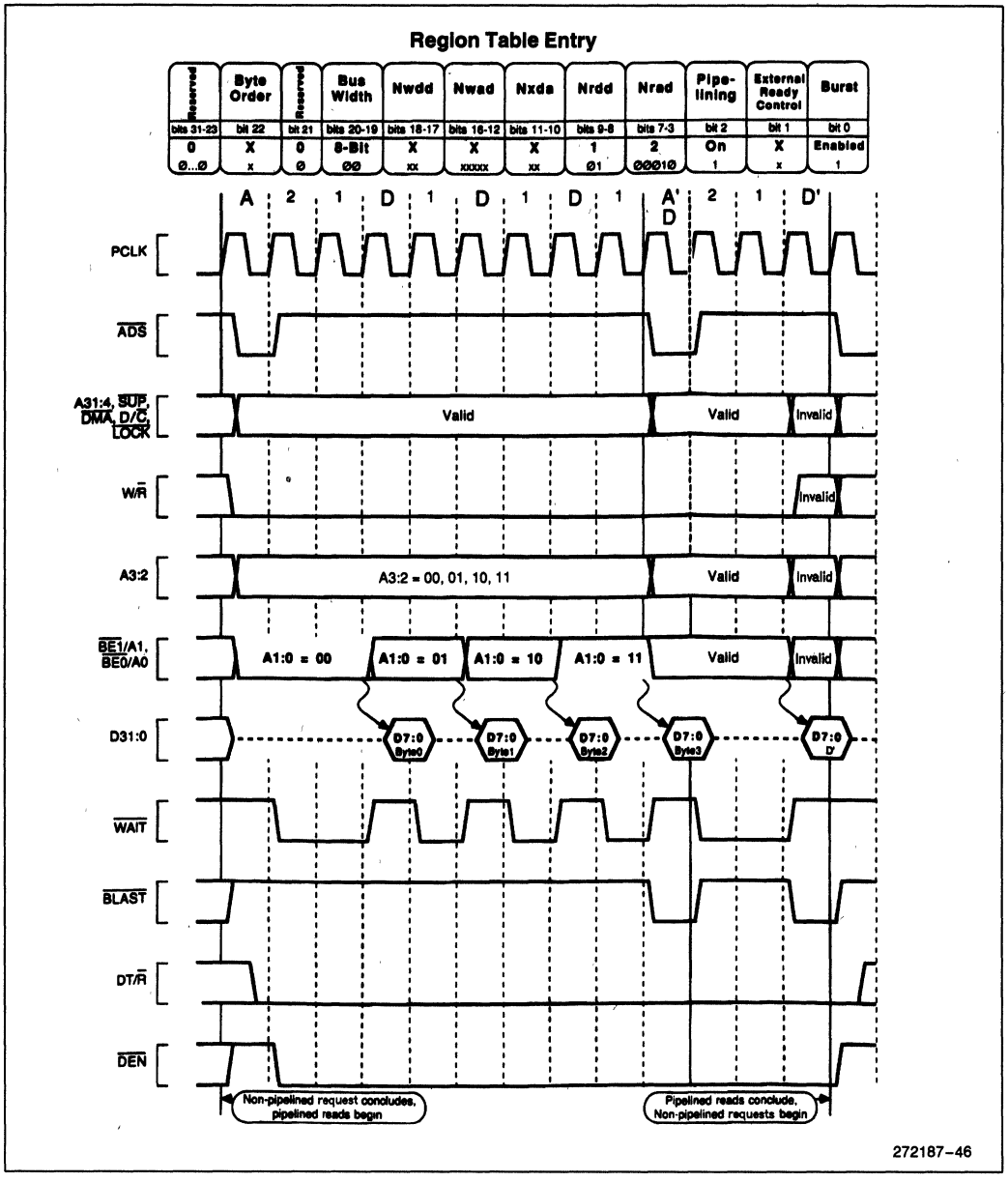


Figure 43. Burst, Pipelined Read Requests with Wait States, 8-Bit Bus

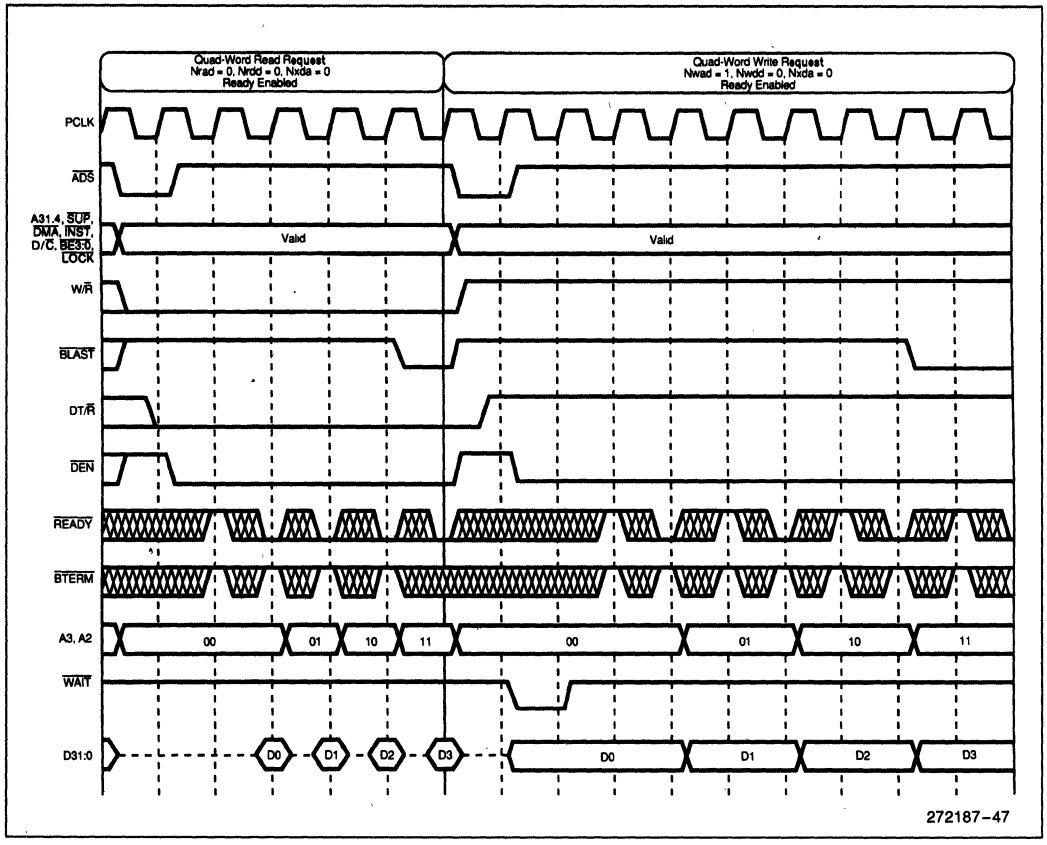
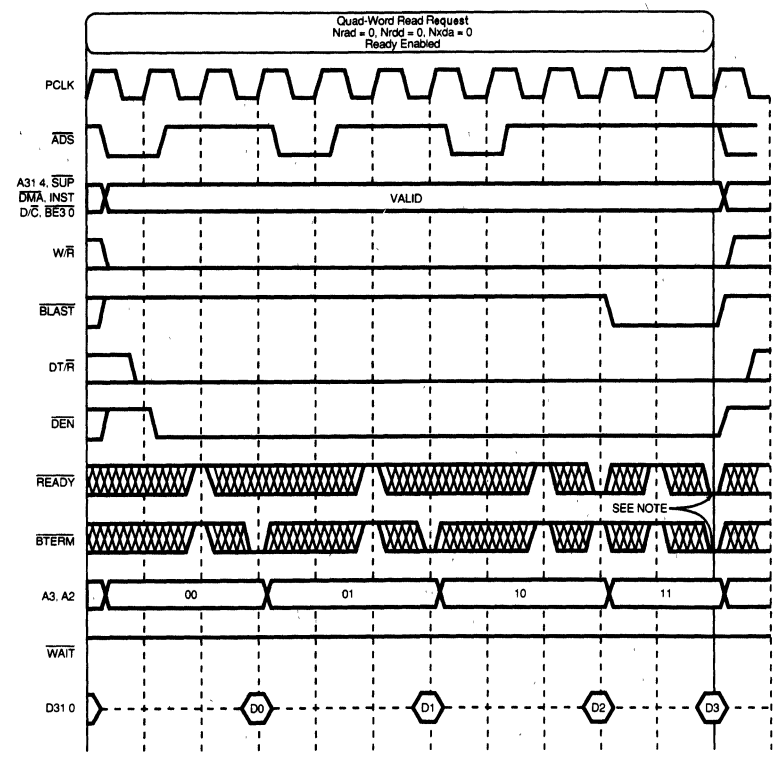


Figure 44. Using External READY



NOTE:
READY adds memory access time to data transfers, whether or not the bus access is a burst access. BTERM interrupts a bus access, whether or not the bus access has more data transfers pending. Either the READY signal or the BTERM signal will terminate a bus access if the signal is asserted during the last (or only) data transfer of the bus access.

Figure 45. Terminating a Burst with BTERM

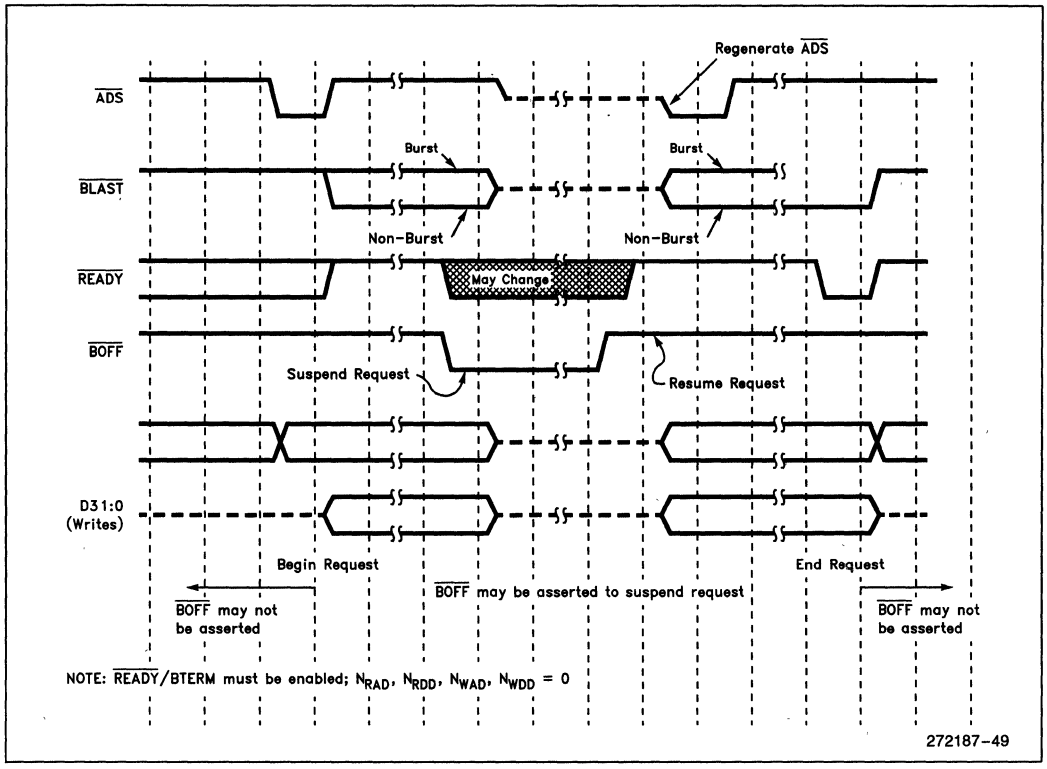


Figure 46. BOFF Functional Timing

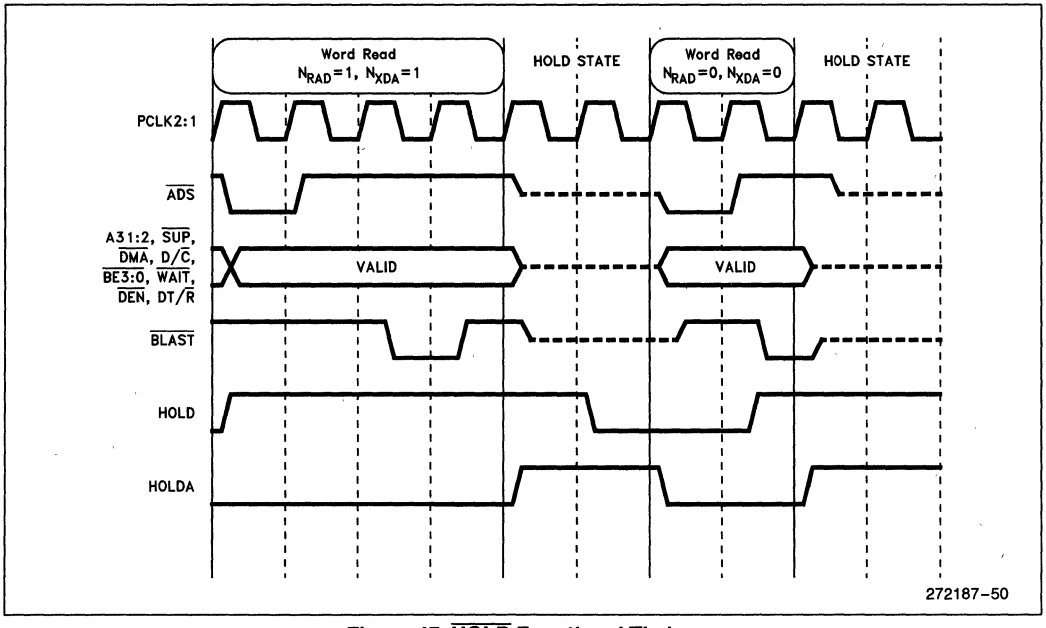
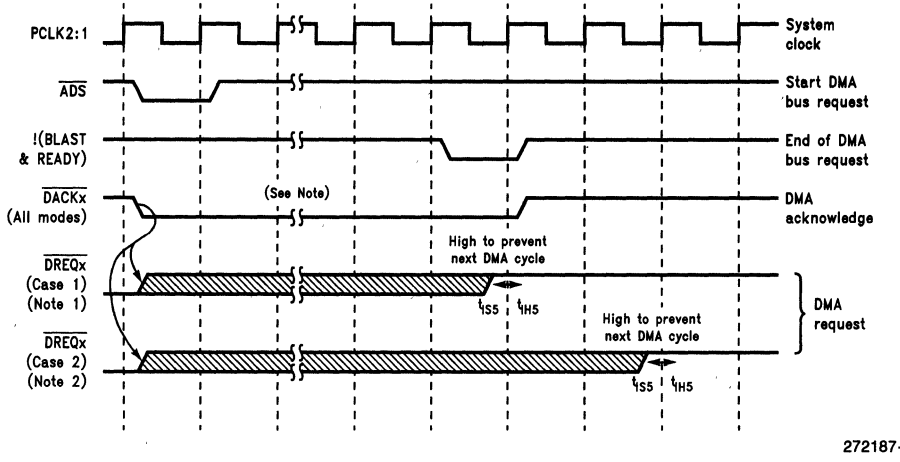
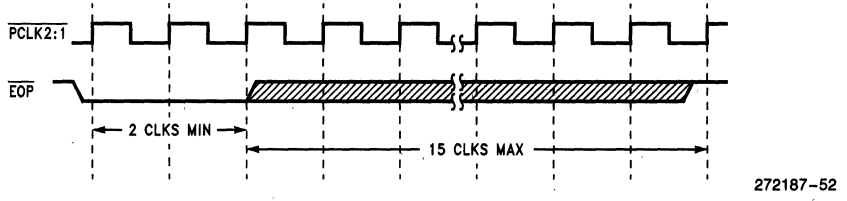


Figure 47. HOLD Functional Timing



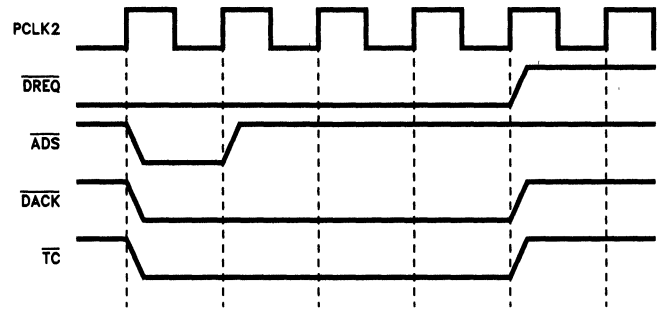
NOTES:
1. Case 1: DREQ must deassert before DACK deasserts. Applications are Fly-by and some packing and unpacking modes, in which loads are followed by loads, or stores are followed by stores.
2. Case 2: DREQ must be deasserted by the second clock (rising edge) after DACK is driven high. Applications are non fly-by transfers and adjacent load-stores or store-loads.
3. DACKx is asserted for the duration of a DMA bus request. The request may consist of multiple bus accesses (defined by ADS and BLAST. Refer to User's Manual for "access", "request" definition.

Figure 48. DREQ and DACK Functional Timing



NOTE:
EOP has the same AC Timing Requirements as DREQ to prevent unwanted DMA requests.
EOP is NOT edge triggered. EOP must be held for a minimum of 2 clock cycles then EOP must be deasserted within 15 clock cycles.

Figure 49. EOP Functional Timing



NOTE:
Terminal Count becomes active during the last bus request of a buffer transfer. If the last LOAD/STORE bus request is executed as multiple bus accesses, the TC will be active for the entire bus request. Refer to the User's Manual for further information.

Figure 50. Terminal Count Functional Timing

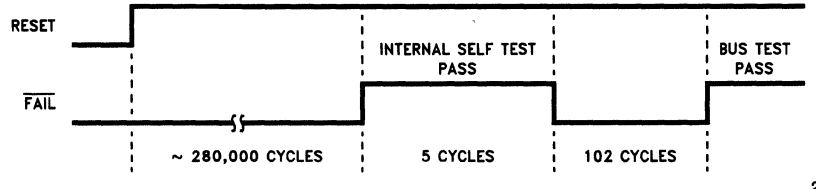


Figure 51. FAIL Functional Timing

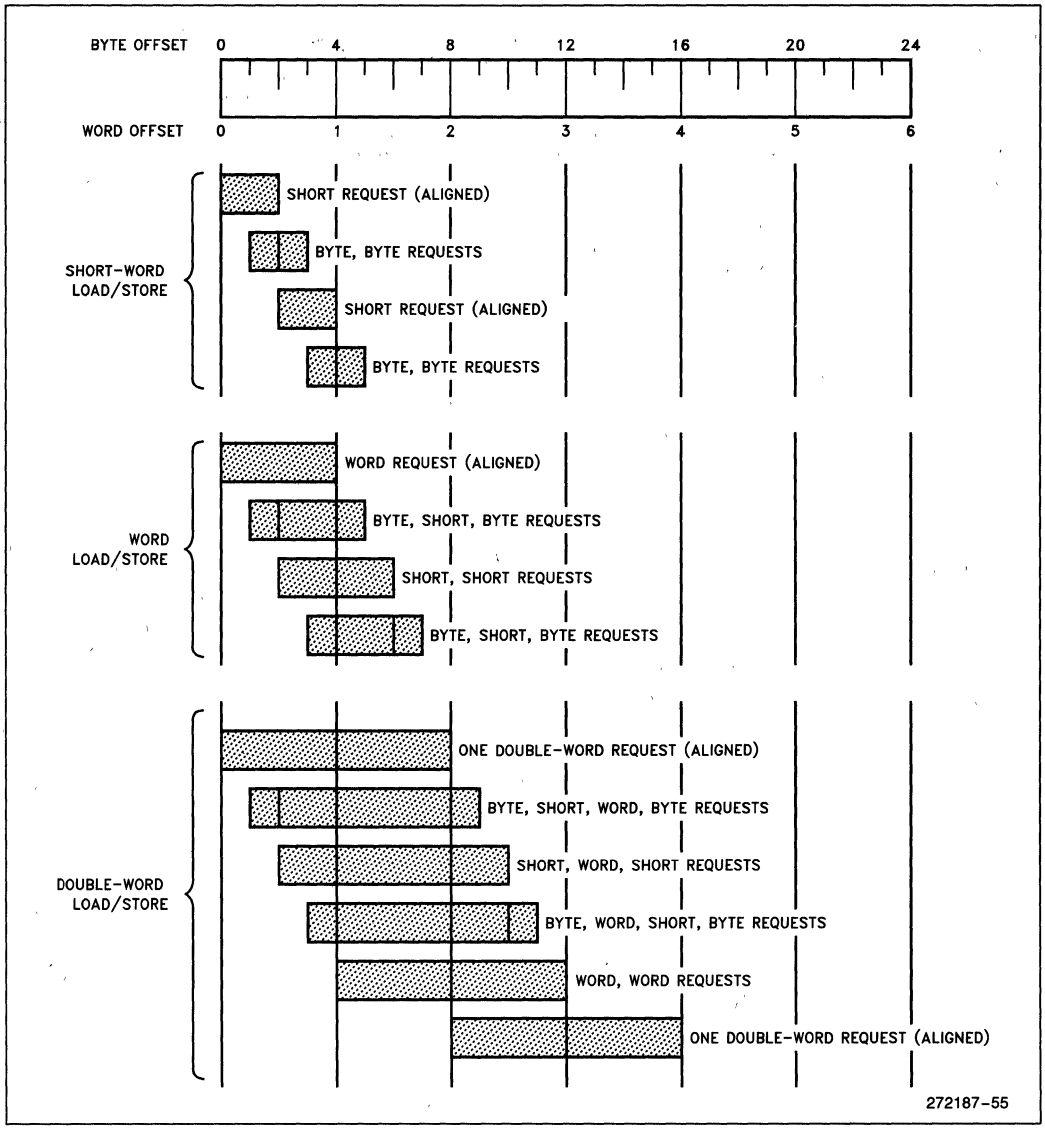


Figure 52. A Summary of Aligned and Unaligned Transfers for Little Endian Regions

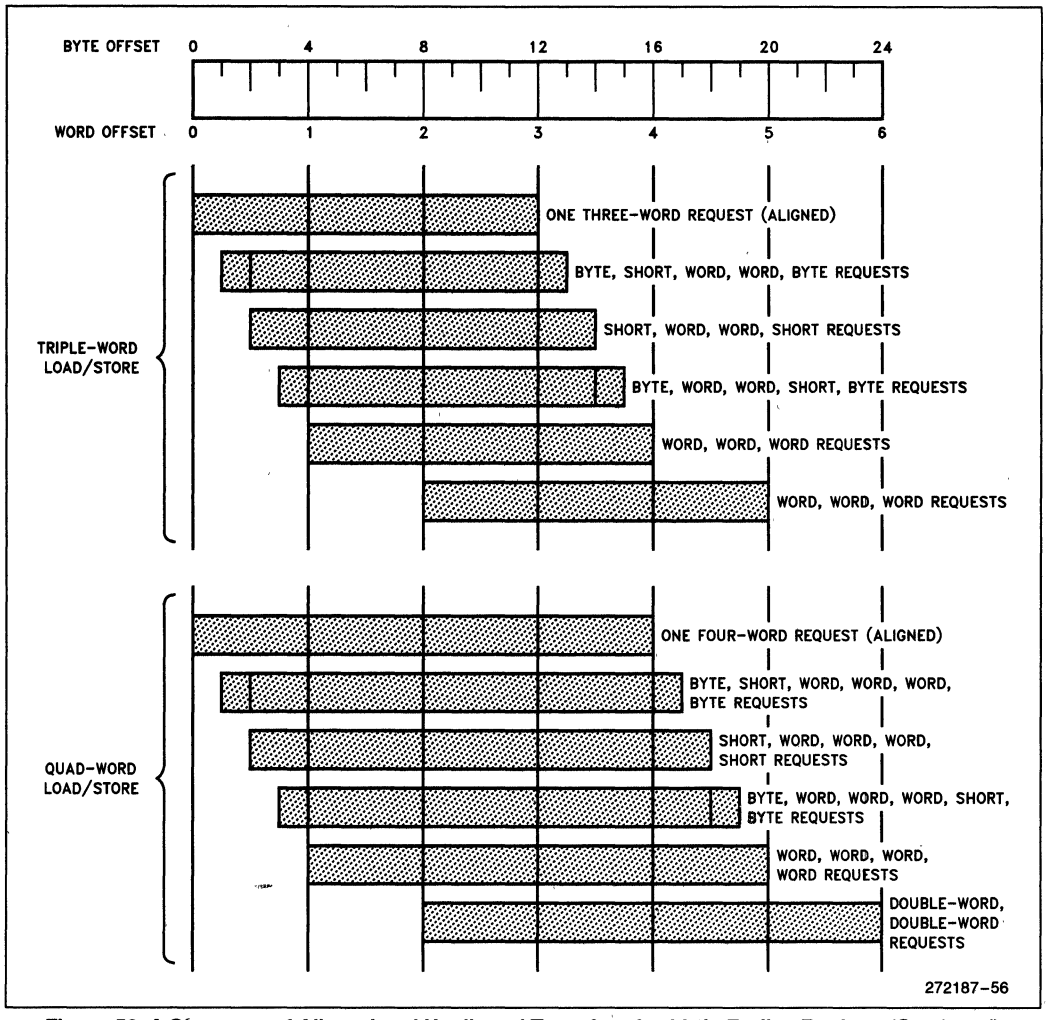


Figure 53. A Summary of Aligned and Unaligned Transfers for Little Endian Regions (Continued)

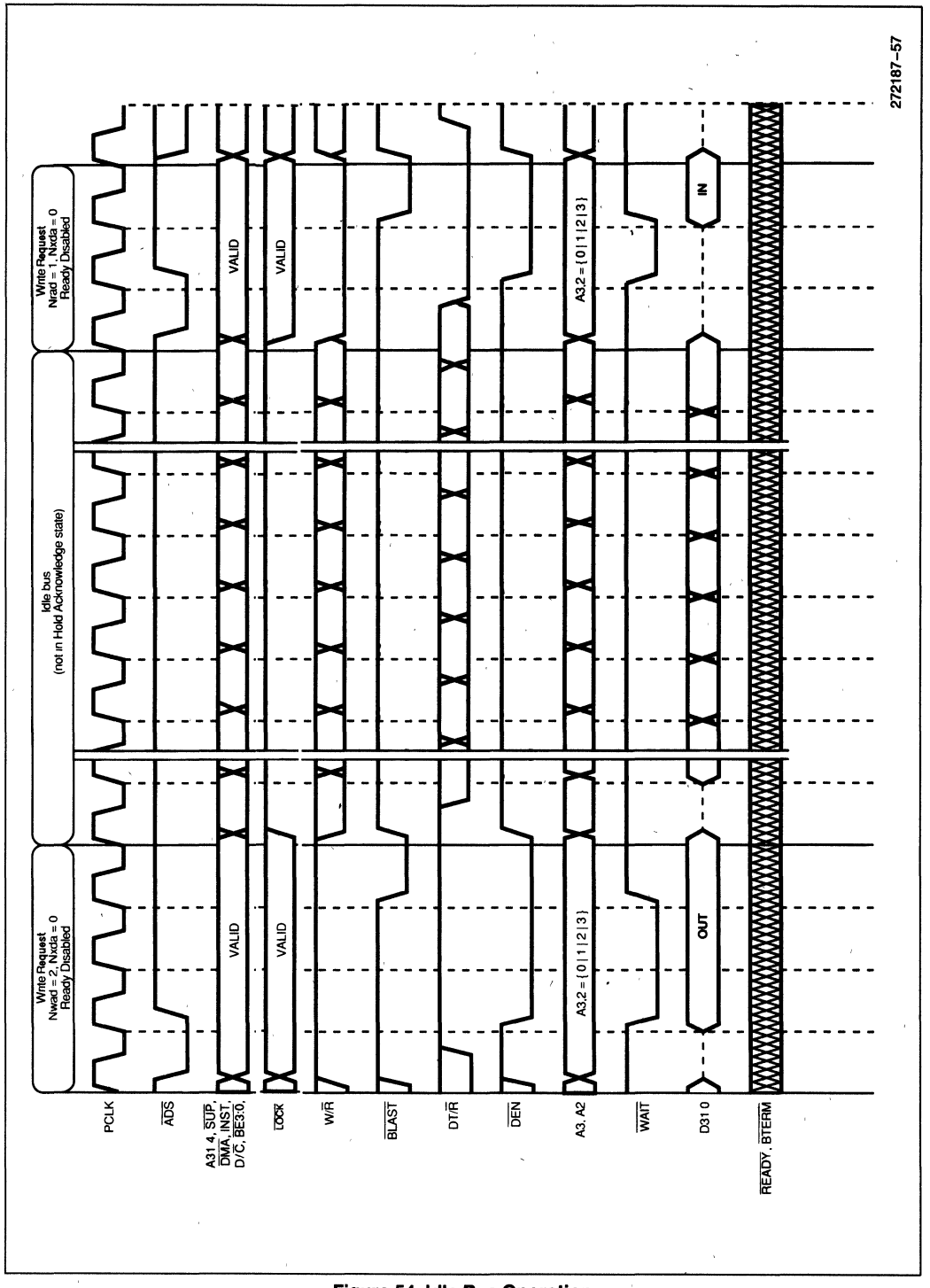


Figure 54. Idle Bus Operation

80960CF-40
32-BIT HIGH-PERFORMANCE SUPERSCALAR PROCESSOR

• Socket and Object Code Compatible with 80960CA
• Two Instructions/Clock Sustained Execution
• Four 71 Mbytes/s DMA Channels with Data Chaining
• Demultiplexed 32-Bit Burst Bus with Pipelining

- **32-Bit Parallel Architecture**
  - **Two Instructions/clock Execution**
  - **Load/Store Architecture**
  - **Sixteen 32-Bit Global Registers**
  - **Sixteen 32-Bit Local Registers**
  - **Manipulates 64-Bit Bit Fields**
  - **11 Addressing Modes**
  - **Full Parallel Fault Model**
  - **Supervisor Protection Model**
- **Fast Procedure Call/Return Model**
  - **Full Procedure Call in 4 Clocks**
- **On-Chip Register Cache**
  - **Caches Registers on Call/Ret**
  - **Minimum of 6 Frames Provided**
  - **Up to 15 Programmable Frames**
- **On-Chip Instruction Cache**
  - **4 Kbyte Two-Way Set Associative**
  - **128-Bit Path to Instruction Sequencer**
  - **Cache-Lock Modes**
  - **Cache-Off Mode**
- **High Bandwidth On-Chip Data RAM**
  - **1 Kbyte On-Chip Data RAM**
  - **Sustains 128 bits per Clock Access**
- **Selectable Big or Little Endian Byte Ordering**
- **Four On-Chip DMA Channels**
  - **71 Mbytes/s Fly-by Transfers**
  - **40 Mbytes/s Two-Cycle Transfers**
  - **Data Chaining**
  - **Data Packing/Unpacking**
  - **Programmable Priority Method**
- **32-Bit Demultiplexed Burst Bus**
  - **128-Bit Internal Data Paths to *and* from Registers**
  - **Burst Bus for DRAM Interfacing**
  - **Address Pipelining Option**
  - **Fully Programmable Wait States**
  - **Supports 8-, 16- or 32-Bit Bus Widths**
  - **Supports Unaligned Accesses**
  - **Supervisor Protection Pin**
- **High-Speed Interrupt Controller**
  - **Up to 248 External Interrupts**
  - **32 Fully Programmable Priorities**
  - **Multi-mode 8-Bit Interrupt Port**
  - **Four Internal DMA Interrupts**
  - **Separate, Non-maskable Interrupt Pin**
  - **Context Switch in 625 ns Typical**
- **On-Chip Data Cache**
  - **1 Kbyte Direct-Mapped, Write Through**
  - **128 bits per Clock Access on Cache Hit**

272493-1

80960CF-40
32-BIT HIGH-PERFORMANCE SUPERSCALAR PROCESSOR

CONTENTS

CONTENTS

PAGE

LIST OF FIGURES

CONTENTS

PAGE

LIST OF FIGURES

LIST OF TABLES

1.0 PURPOSE

This document provides electrical characteristics of Intel's i960® CF embedded 40 MHz microprocessor (also available in 33, 25 and 16 MHz). For descriptions of any 80960CF functional topic — other than parametric performance — consult the *i960® Cx Microprocessor User's Manual* (#270710). To obtain data sheet updates and errata, call Intel's FaxBack data-on-demand system (1-800-628-2283 or 916-356-3105). Other information can be obtained from Intel's technical BBS (916-356-3600).

2.0 80960CF OVERVIEW

Intel's 80960CF is the performance follow-on product to the 80960CA. The 80960CF is socket- and object code-compatible with the CA; this makes CA-to-CF design upgrades straightforward.

As shown in Figure 1, the 80960CF's instruction cache is 4 Kbytes; data cache is 1 Kbyte (80960CA instruction cache is 1 Kbyte; it does not have a data cache.) This extra cache on the CF adds a significant performance boost over the CA.

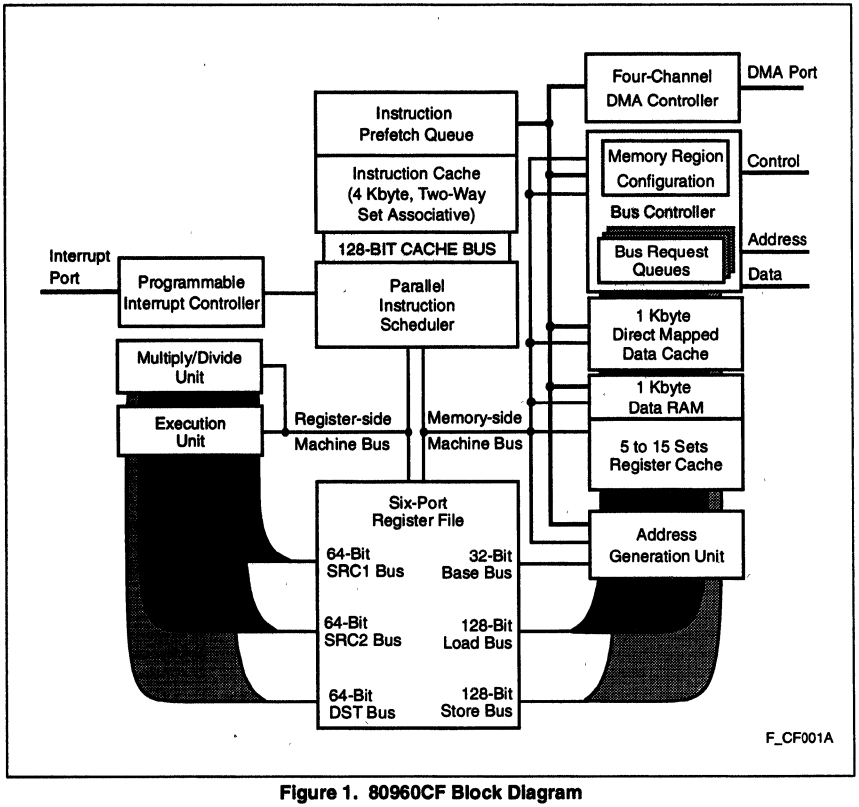


Figure 1. 80960CF Block Diagram

272493-5

The 80960CF is object code compatible with the 32-bit 80960 Core Architecture while including Special Function Register extensions to control on-chip peripherals and instruction set extensions to shift 64-bit operands and configure on-chip hardware. Multiple 128-bit internal buses, on-chip instruction caching and a sophisticated instruction scheduler allow the processor to sustain execution of two instructions every clock and peak at execution of three instructions per clock.

A 32-bit demultiplexed and pipelined burst bus provides a 160 Mbyte/s bandwidth to a system's high-speed external memory subsystem. Also, the 80960CF's on-chip caching of instructions, procedure context and critical program data substantially decouple system performance from the wait states associated with accesses to the system's slower, cost sensitive, main memory subsystem.

The 80960CF bus controller integrates full wait state and bus width control for highest system performance with minimal system design complexity. Unaligned access and Big Endian byte order support reduces the cost of porting existing applications to the 80960CF.

The processor also integrates four complete data-chaining DMA channels and a high-speed interrupt controller on-chip. DMA channels perform single-cycle or two-cycle transfers, data packing and unpacking and data chaining. Block transfers — in addition to source or destination synchronized transfers — are provided.

The interrupt controller provides full programmability of 248 interrupt sources into 32 priority levels with a typical interrupt task switch (latency) time of 625 ns.

2.1 The C-Series Core

The C-Series core is a very high performance microarchitectural implementation of the 80960 Core Architecture. This core can sustain execution of two instructions per clock (80 MIPS at 40 MHz). To achieve this level of performance, Intel has incorporated state-of-the-art silicon technology and innovative microarchitectural constructs into the C-Series core implementation. Factors that contribute to the core's performance include:

- Parallel instruction decoding allows issuance of up to three instructions per clock
- Single-clock execution of most instructions
- Parallel instruction decode allows sustained, simultaneous execution of two single-clock instructions every clock cycle
- Efficient instruction pipeline minimizes pipeline break losses
- Register and resource scoreboarding allow simultaneous multi-clock instruction execution
- Branch look-ahead and prediction allows many branches to execute with no pipeline break
- Local Register Cache integrated on-chip caches Call/Return context
- Two-way set associative, 4 Kbyte integrated instruction cache
- 1 Kbyte integrated Data RAM sustains a four-word (128-bit) access every clock cycle
- Direct mapped, 1 Kbyte data cache, write through, write allocate

2.2 Pipelined, Burst Bus

A 32-bit high performance bus controller interfaces the 80960CF to external memory and peripherals. The Bus Control Unit features a maximum transfer rate of 160 Mbytes per second (at 40 MHz). Internally programmable wait states and 16 separately configurable memory regions allow the processor to interface with a variety of memory subsystems with a minimum of system complexity and a maximum of performance. The Bus Controller's main features include:

- Demultiplexed, Burst Bus to exploit most efficient DRAM access modes
- Address Pipelining to reduce memory cost while maintaining performance
- 32-, 16- and 8-bit modes for I/O interfacing ease
- Full internal wait state generation to reduce system cost
- Little and Big Endian support to ease application development
- Unaligned access support for code portability
- Three-deep request queue to decouple the bus from the core

272493-6

2.3 Instruction Set Summary

Table 1 summarizes the 80960CF instruction set by logical groupings. See the *i960® Cx Microprocessor User's Manual* (#270710) for a complete description of the instruction set.

2.4 Flexible DMA Controller

A four-channel DMA controller provides high speed DMA control for data transfers involving peripherals and memory. The DMA provides advanced features such as data chaining, byte assembly and disassembly and a high performance fly-by mode capable of transfer speeds of up to 71 Mbytes per second at 40 MHz. The DMA controller features a performance and flexibility which is only possible by integrating the DMA controller and the 80960CF core.

2.5 Priority Interrupt Controller

A programmable-priority interrupt controller manages up to 248 external sources through the 8-bit external interrupt port. The Interrupt Unit also handles the four internal sources from the DMA controller and a single non-maskable interrupt input. The 8-bit interrupt port can also be configured to provide individual interrupt sources that are level or edge triggered.

80960CF interrupts are prioritized and signaled within 225 ns of the request. If the interrupt is of higher priority than the processor priority, the context switch to the interrupt routine typically completes in another 400 ns. The interrupt unit provides the mechanism for the low latency and high throughput interrupt service which is essential for embedded applications.

1

Table 1. 80960CF Instruction Set

Data Movement	Arithmetic	Logical	Bit / Bit Field / Byte
Load	Add	And	Set Bit
Store	Subtract	Not And	Clear Bit
Move	Multiply	And Not	Not Bit
Load Address	Divide	Or	Alter Bit
	Remainder	Exclusive Or	Scan For Bit
	Modulo	Not Or	Span Over Bit
	Shift	Or Not	Extract
	*Extended Shift	Nor	Modify
	Extended Multiply	Exclusive Nor	Scan Byte for Equal
	Extended Divide	Not	
	Add with Carry	Nand	
	Subtract with Carry		
	Rotate		
Comparison	**Branch**	**Call/Return**	**Fault**
Compare	Unconditional Branch	Call	Conditional Fault
Conditional Compare	Conditional Branch	Call Extended	Synchronize Faults
Compare and Increment	Compare and Branch	Call System	
Compare and Decrement		Return	
Test Condition Code		Branch and Link	
Check Bit			
Debug	**Processor Mgmt**	**Atomic**	
Modify Trace Controls	Flush Local Registers	Atomic Add	
Mark	Modify Arithmetic Controls	Atomic Modify	
Force Mark	Modify Process Controls		
	*System Control		
	*DMA Control		

NOTES: Instructions marked by (*) are 80960Cx extensions to the 80960 instruction set.

272493-7

3.0 PACKAGE INFORMATION

3.1 Package Introduction

This section describes the pins, pinouts and thermal characteristics for the 80960CF in the 168-pin Ceramic Pin Grid Array (PGA) package. For complete package specifications and information, see the *Packaging* Handbook (# 240800).

3.2 Pin Descriptions

This section defines the 80960CF pins. Table 2 presents the legend for interpreting the pin descriptions in the following tables. Pins associated with the 32-bit demultiplexed processor bus are described in Table 3. Pins associated with basic processor configuration and control are described in Table 4. Pins associated with the 80960CF DMA Controller and Interrupt Unit are described in Table 5.

All pins float while the processor is in the ONCE mode.

Table 2. Pin Description Nomenclature

Symbol	Description
I	Input only pin
O	Output only pin
I/O	Pin can be either an input or output
–	Pins "must be" connected as described
S(...)	Synchronous. Inputs must meet setup and hold times relative to PCLK2:1 for proper operation. Outputs are synchronous to PCLK2:1. S(E) Edge sensitive input S(L) Level sensitive input
A(...)	Asynchronous. Inputs may be asynchronous to PCLK2:1. A(E) Edge sensitive input A(L) Level sensitive input
H(...)	While the bus is in the Hold Acknowledge or Bus Backoff state, the pin: H(1) is driven to V_{CC} H(0) is driven to V_{SS} H(Z) floats H(Q) continues to be a valid input
R(...)	While the processor's $\overline{\text{RESET}}$ pin is low, the pin: R(1) is driven to V_{CC} R(0) is driven to V_{SS} R(Z) floats R(Q) continues to be a valid output

272493-8

Table 3. 80960CF Pin Description — External Bus Signals (Sheet 1 of 3)

Name	Type	Description
A31:2	O S H(Z) R(Z)	**ADDRESS BUS** carries the physical address' upper 30 bits. A31 is the most significant bit; A2 is least significant. During a bus access, A31:2 identify all external addresses to word (4-byte) boundaries. Byte enable signals indicate the selected byte in each word. During burst accesses, A3:2 increment to indicate successive data cycles.
D31:0	I/O S(L) H(Z) R(Z)	**DATA BUS** carries 32-, 16- or 8-bit data quantities depending on bus width configuration. The least significant bit is carried on D0 and the most significant on D31. When the bus is configured for 8-bit data, the lower 8 data lines, D7:0 are used. For 16-bit data bus widths, D15:0 are used. For 32-bit bus widths the full data bus is used.
$\overline{BE3:0}$	O S H(Z) R(1)	**BYTE ENABLES** select which of the four bytes addressed by A31:2 are active during an access to a memory region configured for a 32-bit data-bus width. $\overline{BE3}$ applies to D31:24; $\overline{BE2}$ applies to D23:16; $\overline{BE1}$ applies to D15:8 $\overline{BE0}$ applies to D7:0. 32-bit bus: $\overline{BE3}$ –Byte Enable 3 –enable D31:24 $\overline{BE2}$ –Byte Enable 2 –enable D23:16 $\overline{BE1}$ –Byte Enable 1 –enable D15:8 $\overline{BE0}$ –Byte Enable 0 –enable D7:0 For accesses to a memory region configured for a 16-bit data-bus width, the processor uses the $\overline{BE3}$, $\overline{BE1}$ and $\overline{BE0}$ pins as $\overline{BHE}$, A1 and $\overline{BLE}$ respectively. 16-bit bus: $\overline{BE3}$ –Byte High Enable ($\overline{BHE}$) –enable D15:8 $\overline{BE2}$ –Not used (driven high or low) $\overline{BE1}$ –Address Bit 1 (A1) $\overline{BE0}$ –Byte Low Enable ($\overline{BLE}$) –enable D7:0 For accesses to a memory region configured for an 8-bit data-bus width, the processor uses the $\overline{BE1}$ and $\overline{BE0}$ pins as A1 and A0 respectively. 8-bit bus: $\overline{BE3}$ –Not used (driven high or low) $\overline{BE2}$ –Not used (driven high or low) $\overline{BE1}$ –Address Bit 1 (A1) $\overline{BE0}$ –Address Bit 0 (A0)
W/$\overline{R}$	O S H(Z) R(0)	**WRITE/READ** is asserted for read requests and deasserted for write requests. The W/$\overline{R}$ signal changes in the same clock cycle as $\overline{ADS}$. It remains valid for the entire access in non-pipelined regions. In pipelined regions, W/$\overline{R}$ is not guaranteed to be valid in the last cycle of a read access.
$\overline{ADS}$	O S H(Z) R(1)	**ADDRESS STROBE** indicates a valid address and the start of a new bus access. $\overline{ADS}$ is asserted for the first clock of a bus access.
$\overline{READY}$	I S(L) H(Z) R(Z)	**READY** is an input which signals the termination of a data transfer. $\overline{READY}$ is used to indicate that read data on the bus is valid or that a write-data transfer has completed. The $\overline{READY}$ signal works in conjunction with the internally programmed wait-state generator. If $\overline{READY}$ is enabled in a region, the pin is sampled after the programmed number of wait-states has expired. If the $\overline{READY}$ pin is deasserted, wait states continue to be inserted until $\overline{READY}$ becomes asserted. This is true for the N_{RAD}, N_{RDD}, N_{WAD} and N_{WDD} wait states. The N_{XDA} wait states cannot be extended.

272493–9

Table 3. 80960CF Pin Description — External Bus Signals (Sheet 2 of 3)

Name	Type	Description
BTERM	I S(L) H(Z) R(Z)	**BURST TERMINATE** is an input which breaks up a burst access and causes another address cycle to occur. The BTERM signal works in conjunction with the internally programmed wait-state generator. If READY and BTERM are enabled in a region, the BTERM pin is sampled after the programmed number of wait states has expired. When BTERM is asserted, a new ADS signal is generated and the access is completed. The READY input is ignored when BTERM is asserted. BTERM must be externally synchronized to satisfy BTERM setup and hold times.
WAIT	O S H(Z) R(1)	**WAIT** indicates internal wait state generator status. WAIT is asserted when wait states are being caused by the internal wait state generator and not by the READY or BTERM inputs. WAIT can be used to derive a write-data strobe. WAIT can also be thought of as a READY output that the processor provides when it is inserting wait states.
BLAST	O S H(Z) R(0)	**BURST LAST** indicates the last transfer in a bus access. BLAST is asserted in the last data transfer of burst and non-burst accesses after the wait state counter reaches zero. BLAST remains asserted until the clock following the last cycle of the last data transfer of a bus access. If the READY or BTERM input is used to extend wait states, the BLAST signal remains asserted until READY or BTERM terminates the access.
DT/R	O S H(Z) R(0)	**DATA TRANSMIT/RECEIVE** indicates direction for data transceivers. DT/R is used in conjunction with DEN to provide control for data transceivers attached to the external bus. When DT/R is asserted, the signal indicates that the processor receives data. Conversely, when deasserted, the processor sends data. DT/R changes only while DEN is high.
DEN	O S H(Z) R(1)	**DATA ENABLE** indicates data cycles in a bus request. DEN is asserted at the start of the bus request first data cycle and is deasserted at the end of the last data cycle. DEN is used in conjunction with DT/R to provide control for data transceivers attached to the external bus. DEN remains asserted for sequential reads from pipelined memory regions. DEN is deasserted when DT/R changes.
LOCK	O S H(Z) R(1)	**BUS LOCK** indicates that an atomic read-modify-write operation is in progress. LOCK may be used to prevent external agents from accessing memory which is currently involved in an atomic operation. LOCK is asserted in the first clock of an atomic operation and deasserted in the clock cycle following the last bus access for the atomic operation. To allow the most flexibility for memory system enforcement of locked accesses, the processor acknowledges a bus hold request when LOCK is asserted. The processor performs DMA transfers while LOCK is active.
HOLD	I S(L) H(Z) R(Z)	**HOLD REQUEST** signals that an external agent requests access to the external bus. The processor asserts HOLDA after completing the current bus request. HOLD, HOLDA and BREQ are used together to arbitrate access to the processor's external bus by external bus agents.
BOFF	I S(L) H(Z) R(Z)	**BUS BACKOFF**, when asserted, suspends the current access and causes the bus pins to float. When BOFF is deasserted, the ADS signal is asserted on the next clock cycle and the access is resumed.

272493–10

Table 3. 80960CF Pin Description — External Bus Signals (Sheet 3 of 3)

Name	Type	Description
HOLDA	O S H(1) R(Q)	**HOLD ACKNOWLEDGE** indicates to a bus requestor that the processor has relinquished control of the external bus. When HOLDA is asserted, the external address bus, data bus and bus control signals are floated. HOLD, $\overline{BOFF}$, HOLDA and BREQ are used together to arbitrate access to the processor's external bus by external bus agents. Since the processor grants HOLD requests and enters the Hold Acknowledge state even while $\overline{RESET}$ is asserted, the state of the HOLDA pin is independent of the $\overline{RESET}$ pin.
BREQ	O S H(Q) R(0)	**BUS REQUEST** is asserted when the bus controller has a request pending. BREQ can be used by external bus arbitration logic in conjunction with HOLD and HOLDA to determine when to return mastership of the external bus to the processor.
D/$\overline{C}$	O S H(Z) R(Z)	**DATA OR CODE** is asserted for a data request and deasserted for instruction requests. D/$\overline{C}$ has the same timing as W/$\overline{R}$.
$\overline{DMA}$	O S H(Z) R(Z)	**DMA ACCESS** indicates whether the bus request was initiated by the DMA controller. $\overline{DMA}$ is asserted for any DMA request. $\overline{DMA}$ is deasserted for all other requests.
$\overline{SUP}$	O S H(Z) R(Z)	**SUPERVISOR ACCESS** indicates whether the bus request is issued while in supervisor mode. $\overline{SUP}$ is asserted when the request has supervisor privileges and is deasserted otherwise. $\overline{SUP}$ can be used to isolate supervisor code and data structures from non-supervisor requests.

272493-11

Table 4. 80960CF Pin Description — Processor Control Signals (Sheet 1 of 2)

Name	Type	Description
$\overline{\text{RESET}}$	I A(L) H(Z) R(Z)	**RESET** causes the chip to reset. When $\overline{\text{RESET}}$ is asserted, all external signals return to the reset state. When $\overline{\text{RESET}}$ is deasserted, initialization begins. When the 2-x clock mode is selected, $\overline{\text{RESET}}$ must remain asserted for 32 CLKIN cycles before being deasserted to guarantee correct processor initialization. When the 1-x clock mode is selected, $\overline{\text{RESET}}$ must remain asserted for 10,000 CLKIN cycles before being deasserted to guarantee correct processor initialization. The CLKMODE pin selects 1-x or 2-x input clock division of the CLKIN pin. The Hold Acknowledge bus state functions while the chip is reset. If the bus is in the Hold Acknowledge state when $\overline{\text{RESET}}$ is asserted, the processor internally resets, but maintains the Hold Acknowledge state on external pins until the Hold request is removed. If a Hold request is made while the processor is in the reset state, the processor bus grants HOLDA and enters the Hold Acknowledge state.
$\overline{\text{FAIL}}$	O S H(Q) R(0)	**FAIL** indicates failure of the self-test performed at initialization. When $\overline{\text{RESET}}$ is deasserted and initialization begins, the $\overline{\text{FAIL}}$ pin is asserted. An internal self-test is performed as part of the initialization process. If this self-test passes, the $\overline{\text{FAIL}}$ pin is deasserted; otherwise it remains asserted. The $\overline{\text{FAIL}}$ pin is reasserted while the processor performs an external bus self-confidence test. If this self-test passes, the processor deasserts the $\overline{\text{FAIL}}$ pin and branches to the user's initialization routine; otherwise the $\overline{\text{FAIL}}$ pin remains asserted. Internal self-test and the use of the $\overline{\text{FAIL}}$ pin can be disabled with the STEST pin.
STEST	I S(L) H(Z) R(Z)	**SELF TEST** enables or disables the internal self-test feature at initialization. STEST is read on the rising edge of $\overline{\text{RESET}}$. When asserted, internal self-test and external bus confidence tests are performed during processor initialization. When deasserted, only the bus confidence tests are performed during initialization.
$\overline{\text{ONCE}}$	I A(L) H(Z) R(Z)	**ON CIRCUIT EMULATION**, when asserted, causes all outputs to be floated. $\overline{\text{ONCE}}$ is continuously sampled while $\overline{\text{RESET}}$ is low and is latched on the rising edge of $\overline{\text{RESET}}$. To place the processor in the ONCE state: (1) assert $\overline{\text{RESET}}$ and $\overline{\text{ONCE}}$ (order does not matter) (2) wait for at least 16 CLKIN periods in 2-x mode—or 10,000 CLKIN periods in 1-x mode—after V_{CC} and CLKIN are within operating specifications (3) deassert $\overline{\text{RESET}}$ (4) wait at least 32 CLKIN periods (The processor will now be latched in the ONCE state while $\overline{\text{RESET}}$ is high.) To exit the ONCE state, bring V_{CC} and CLKIN to operating conditions, then assert $\overline{\text{RESET}}$ and bring $\overline{\text{ONCE}}$ high prior to deasserting $\overline{\text{RESET}}$. CLKIN must operate within the specified operating conditions until Step 4 completes. CLKIN may then be changed to DC to achieve the lowest possible ONCE mode leakage current. $\overline{\text{ONCE}}$ can be used by emulator products or board testers to effectively make an installed processor transparent in the board.

272493-12

Table 4. 80960CF Pin Description — Processor Control Signals (Sheet 2 of 2)

Name	Type	Description
CLKIN	I A(E) H(Z) R(Z)	**CLOCK INPUT** is an input for the external clock needed to run the processor. The external clock is internally divided as prescribed by the CLKMODE pin to produce PCLK2:1.
CLKMODE	I A(L) H(Z) R(Z)	**CLOCK MODE** selects the division factor applied to the external clock input (CLKIN). When CLKMODE is high, CLKIN is divided by one to create PCLK2:1 and the processor's internal clock. When CLKMODE is low, CLKIN is divided by two to create PCLK2:1 and the processor's internal clock. CLKMODE should be tied high or low in a system as the clock mode is not latched by the processor. If left unconnected, the processor internally pulls the CLKMODE pin low, enabling the 2-x clock mode.
PCLK2:1	O S H(Q) R(Q)	**PROCESSOR OUTPUT CLOCKS** provide a timing reference for all inputs and outputs. All input and output timings are specified in relation to PCLK2 and PCLK1. PCLK2 and PCLK1 are identical signals. Two output pins are provided to allow flexibility in the system's allocation of capacitive loading on the clock. PCLK2:1 may also be connected at the processor to form a single clock signal.
V_{SS}	–	**GROUND** connections must be connected externally to a V_{SS} board plane.
V_{CC}	–	**POWER** connections must be connected externally to a V_{CC} board plane.
V_{CCPLL}	–	V_{CCPLL} is a separate V_{CC} supply pin for the phase lock loop used in 1-x clock mode. Connecting a simple lowpass filter to V_{CCPLL} may help reduce clock jitter (T_{CP}) in noisy environments. Otherwise, V_{CCPLL} should be connected to V_{CC}.
NC	–	**NO CONNECT** pins must not be connected in a system.

272493-13

Table 5. 80960CF Pin Description — DMA and Interrupt Unit Control Signals

Name	Type	Description
$\overline{\text{DREQ3:0}}$	I A(L) H(Z) R(Z)	**DMA REQUEST** is used to request a DMA transfer. Each of the four signals requests a transfer on a single channel. $\overline{\text{DREQ0}}$ requests channel 0, $\overline{\text{DREQ1}}$ requests channel 1, etc. When two or more channels are requested simultaneously, the channel with the highest priority is serviced first. Channel priority mode is programmable.
$\overline{\text{DACK3:0}}$	O S H(1) R(1)	**DMA ACKNOWLEDGE** indicates that a DMA transfer is being executed. Each of the four signals acknowledges a transfer for a single channel. $\overline{\text{DACK0}}$ acknowledges channel 0, $\overline{\text{DACK1}}$ acknowledges channel 1, etc. $\overline{\text{DACK3:0}}$ are asserted when the requesting device of a DMA is accessed.
$\overline{\text{EOP/TC3:0}}$	I/O A(L) H(Z/Q) R(Z)	**END OF PROCESS/TERMINAL COUNT** can be programmed as either an input ($\overline{\text{EOP3:0}}$) or output ($\overline{\text{TC3:0}}$), but not both. Each pin is individually programmable. When programmed as an input, $\overline{\text{EOPx}}$ causes termination of a current DMA transfer for the channel that corresponds to the $\overline{\text{EOPx}}$ pin. $\overline{\text{EOP0}}$ corresponds to channel 0, $\overline{\text{EOP1}}$ corresponds to channel 1, etc. When a channel is configured for source *and* destination chaining, the EOP pin for that channel causes termination of only the current buffer transferred and causes the next buffer to be transferred. $\overline{\text{EOP3:0}}$ are asynchronous inputs. When programmed as an output, the channel's $\overline{\text{TCx}}$ pin indicates that the channel byte count has reached 0 and a DMA has terminated. $\overline{\text{TCx}}$ is driven with the same timing as $\overline{\text{DACKx}}$ during the last DMA transfer for a buffer. If the last bus request is executed as multiple bus accesses, $\overline{\text{TCx}}$ stays asserted for the entire bus request.
$\overline{\text{XINT7:0}}$	I A(E/L) H(Z) R(Z)	**EXTERNAL INTERRUPT PINS** cause interrupts to be requested. These pins can be configured in three modes: **Dedicated Mode:** each pin is a dedicated external interrupt source. Dedicated inputs can be individually programmed to be level (low) or edge (falling) activated. **Expanded Mode:** the eight pins act together as an 8-bit vectored interrupt source. The interrupt pins in this mode are level activated. Since the interrupt pins are active low, the vector number requested is the 1's complement of the positive logic value place on the port. This eliminates glue logic to interface to combinational priority encoders which output negative logic. **Mixed Mode:** $\overline{\text{XINT7:5}}$ are dedicated sources and $\overline{\text{XINT4:0}}$ act as the five most significant bits of an expanded mode vector. The least significant bits are set to 010 internally.
$\overline{\text{NMI}}$	I A(E) H(Z) R(Z)	**NON-MASKABLE INTERRUPT** causes a non-maskable interrupt event to occur. $\overline{\text{NMI}}$ is the highest priority interrupt recognized. $\overline{\text{NMI}}$ is an edge (falling) activated source.

272493-14

3.3 80960CF Mechanical Data

3.3.1 80960CF PGA PINOUT

Figure 2 depicts the complete 80960CF PGA pinout as viewed from the top side of the component (i.e., pins facing down). Figure 3 shows the complete 80960CF PGA pinout as viewed from the pin-side of the package (i.e., pins facing up).

Table 6 lists the 80960CF pin names and package location in signal order; Table 7 lists the pin names and package location in pin order. See **Section 4.0, ELECTRICAL SPECIFICATIONS** for specifications and recommended connections.

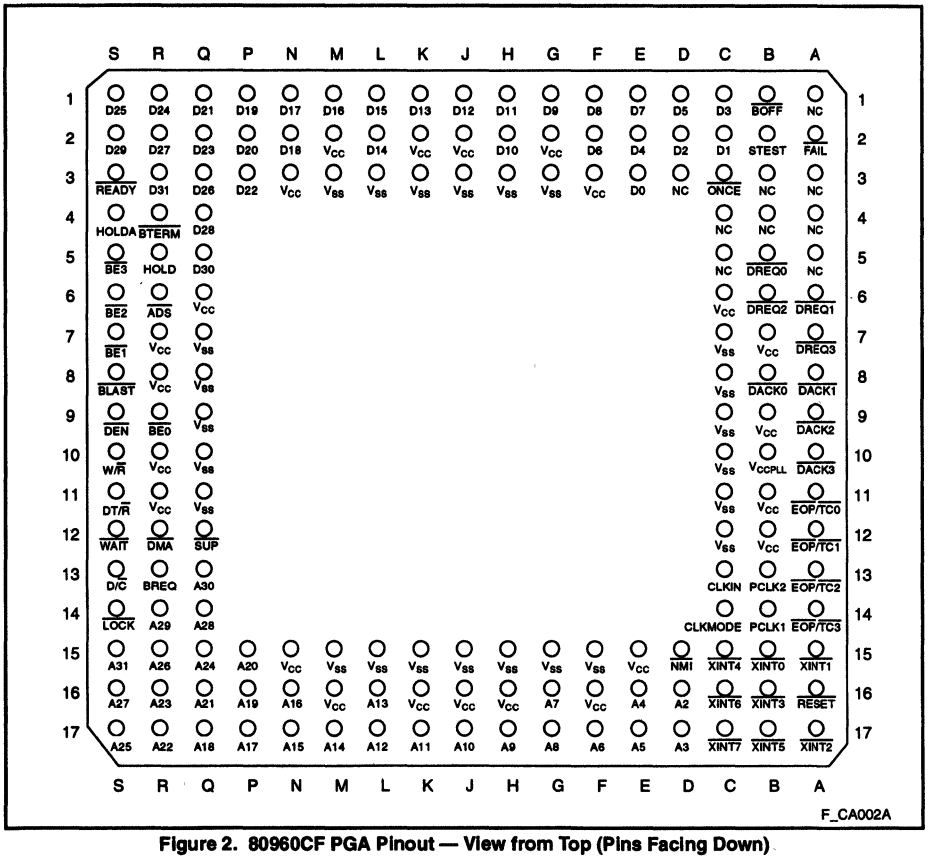


Figure 2. 80960CF PGA Pinout — View from Top (Pins Facing Down)

272493-15

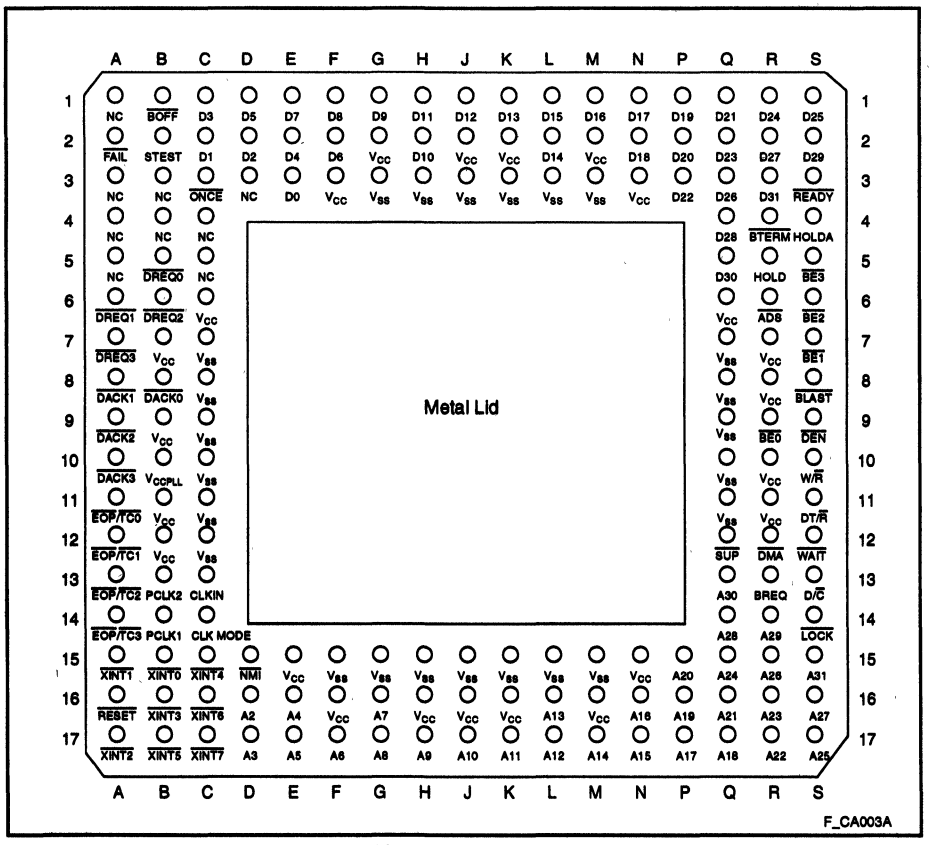


Figure 3. 80960CF PGA Pinout — View from Bottom (Pins Facing Up)

272493-16

Table 6. 80960CF PGA Pinout — In Signal Order

Address Bus		Data Bus		Bus Control		Processor Control		I/O	
Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
A31	S15	D31	R3	$\overline{BE3}$	S5	$\overline{RESET}$	A16	$\overline{DREQ3}$	A7
A30	Q13	D30	Q5	$\overline{BE2}$	S6			$\overline{DREQ2}$	B6
A29	R14	D29	S2	$\overline{BE1}$	S7	$\overline{FAIL}$	A2	$\overline{DREQ1}$	A6
A28	Q14	D28	Q4	$\overline{BE0}$	R9			$\overline{DREQ0}$	B5
A27	S16	D27	R2			STEST	B2		
A26	R15	D26	Q3	W/$\overline{R}$	S10			$\overline{DACK3}$	A10
A25	S17	D25	S1			$\overline{ONCE}$	C3	$\overline{DACK2}$	A9
A24	Q15	D24	R1	$\overline{ADS}$	R6			$\overline{DACK1}$	A8
A23	R16	D23	Q2			CLKIN	C13	$\overline{DACK0}$	B8
A22	R17	D22	P3	$\overline{READY}$	S3	CLKMODE	C14		
A21	Q16	D21	Q1	$\overline{BTERM}$	R4	PLCK1	B14	$\overline{EOP}$/$\overline{TC3}$	A14
A20	P15	D20	P2			PLCK2	B13	$\overline{EOP}$/$\overline{TC2}$	A13
A19	P16	D19	P1	$\overline{WAIT}$	S12			$\overline{EOP}$/$\overline{TC1}$	A12
A18	Q17	D18	N2	$\overline{BLAST}$	S8	**V_{SS}**		$\overline{EOP}$/$\overline{TC0}$	A11
A17	P17	D17	N1			*Location*			
A16	N16	D16	M1	DT/$\overline{R}$	S11	C7, C8, C9, C10, C11, C12, F15, G3, G15, H3, H15, J3, J15, K3, K15, L3, L15, M3, M15, Q7, Q8, Q9, Q10, Q11		$\overline{XINT7}$	C17
A15	N17	D15	L1	$\overline{DEN}$	S9			$\overline{XINT6}$	C16
A14	M17	D14	L2					$\overline{XINT5}$	B17
A13	L16	D13	K1	$\overline{LOCK}$	S14			$\overline{XINT4}$	C15
A12	L17	D12	J1					$\overline{XINT3}$	B16
A11	K17	D11	H1			**V_{CC}**		$\overline{XINT2}$	A17
A10	J17	D10	H2	HOLD	R5	*Location*		$\overline{XINT1}$	A15
A9	H17	D9	G1	HOLDA	S4	B7, B9, B11, B12, C6, E15, F3, F16, G2, H16, J2, J16, K2, K16, M2, M16, N3, N15, Q6, R7, R8, R10, R11		$\overline{XINT0}$	B15
A8	G17	D8	F1	BREQ	R13				
A7	G16	D7	E1					$\overline{NMI}$	D15
A6	F17	D6	F2	D/$\overline{C}$	S13				
A5	E17	D5	D1	$\overline{DMA}$	R12				
A4	E16	D4	E2	$\overline{SUP}$	Q12	V_{CCPLL}	B10		
A3	D17	D3	C1			**No Connect**			
A2	D16	D2	D2	$\overline{BOFF}$	B1	*Location*			
		D1	C2			A1, A3, A4, A5, B3, B4, C4, C5, D3			
		D0	E3						

272493-17

Table 7. 80960CF PGA Pinout — In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	NC	C1	D3	G1	D9	M1	D16	R1	D24
A2	$\overline{FAIL}$	C2	D1	G2	V_{CC}	M2	V_{CC}	R2	D27
A3	NC	C3	$\overline{ONCE}$	G3	V_{SS}	M3	V_{SS}	R3	D31
A4	NC	C4	NC	G15	V_{SS}	M15	V_{SS}	R4	$\overline{BTERM}$
A5	NC	C5	NC	G16	A7	M16	V_{CC}	R5	HOLD
A6	$\overline{DREQ1}$	C6	V_{CC}	G17	A8	M17	A14	R6	$\overline{ADS}$
A7	$\overline{DREQ3}$	C7	V_{SS}					R7	V_{CC}
A8	$\overline{DACK1}$	C8	V_{SS}	H1	D11	N1	D17	R8	V_{CC}
A9	$\overline{DACK2}$	C9	V_{SS}	H2	D10	N2	D18	R9	$\overline{BE0}$
A10	$\overline{DACK3}$	C10	V_{SS}	H3	V_{SS}	N3	V_{CC}	R10	V_{CC}
A11	$\overline{EOP/TC0}$	C11	V_{SS}	H15	V_{SS}	N15	V_{CC}	R11	V_{CC}
A12	$\overline{EOP/TC1}$	C12	V_{SS}	H16	V_{CC}	N16	A16	R12	$\overline{DMA}$
A13	$\overline{EOP/TC2}$	C13	CLKIN	H17	A9	N17	A15	R13	BREQ
A14	$\overline{EOP/TC3}$	C14	CLKMODE					R14	A29
A15	$\overline{XINT1}$	C15	$\overline{XINT4}$	J1	D12	P1	D19	R15	A26
A16	$\overline{RESET}$	C16	$\overline{XINT6}$	J2	V_{CC}	P2	D20	R16	A23
A17	$\overline{XINT2}$	C17	$\overline{XINT7}$	J3	V_{SS}	P3	D22	R17	A22
				J15	V_{SS}	P15	A20		
B1	$\overline{BOFF}$	D1	D5	J16	V_{CC}	P16	A19	S1	D25
B2	STEST	D2	D2	J17	A10	P17	A17	S2	D29
B3	NC	D3	NC					S3	$\overline{READY}$
B4	NC	D15	$\overline{NMI}$	K1	D13	Q1	D21	S4	HOLDA
B5	$\overline{DREQ0}$	D16	A2	K2	V_{CC}	Q2	D23	S5	$\overline{BE3}$
B6	$\overline{DREQ2}$	D17	A3	K3	V_{SS}	Q3	D26	S6	$\overline{BE2}$
B7	V_{CC}			K15	V_{SS}	Q4	D28	S7	$\overline{BE1}$
B8	$\overline{DACK0}$	E1	D7	K16	V_{CC}	Q5	D30	S8	$\overline{BLAST}$
B9	V_{CC}	E2	D4	K17	A11	Q6	V_{CC}	S9	$\overline{DEN}$
B10	V_{CCPLL}	E3	D0			Q7	V_{SS}	S10	$W/\overline{R}$
B11	V_{CC}	E15	V_{CC}	L1	D15	Q8	V_{SS}	S11	$DT/\overline{R}$
B12	V_{CC}	E16	A4	L2	D14	Q9	V_{SS}	S12	$\overline{WAIT}$
B13	PCLK2	E17	A5	L3	V_{SS}	Q10	V_{SS}	S13	$D/\overline{C}$
B14	PCLK1			L15	V_{SS}	Q11	V_{SS}	S14	$\overline{LOCK}$
B15	$\overline{XINT0}$	F1	D8	L16	A13	Q12	$\overline{SUP}$	S15	A31
B16	$\overline{XINT3}$	F2	D6	L17	A12	Q13	A30	S16	A27
B17	$\overline{XINT5}$	F3	V_{CC}			Q14	A28	S17	A25
		F15	V_{SS}			Q15	A24		
		F16	V_{CC}			Q16	A21		
		F17	A6			Q17	A18		

272493-18

3.4 Package Thermal Specifications

The 80960CF is specified for operation when T_C (case temperature) is within the range of 0°C–85°C. T_C may be measured in any environment to determine whether the 80960CF is within specified operating range. Case temperature should be measured at the center of the top surface, opposite the pins. Refer to Figure 4.

T_A (ambient temperature) is calculated from θ_{CA} (thermal resistance from case to ambient) using the equation:

$$T_A = T_C - P^*\theta_{CA}$$

Table 8 shows the maximum T_A allowable (without exceeding T_C) at various airflows and operating frequencies (f_{PCLK}).

Note that T_A is greatly improved by attaching fins or a heatsink to the package. P (maximum power consumption) is calculated by using the typical I_{CC} as tabulated in **Section 4.4, DC Specifications** and V_{CC} of 5V.

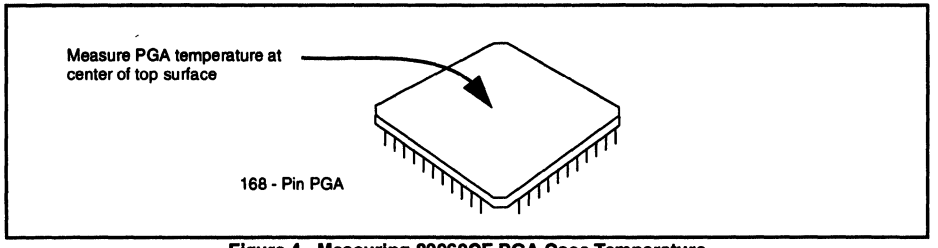


Figure 4. Measuring 80960CF PGA Case Temperature

Table 8. Maximum T_A at Various Airflows in °C (PGA Package Only)

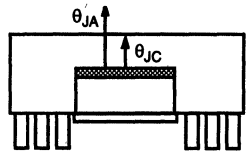
		Airflow-ft/min (m/sec)					
	f_{PCLK} (MHz)	0 (0)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)	1000 (5.07)
T_A with Heatsink*	40	20	40	58	60	66	68
T_A without Heatsink*	40	0	15	30	40	50	52

NOTES:
*0.285" high unidirectional heatsink (Al alloy 6061, 50 mil fin width, 150 mil center-to-center fin spacing).

272493-19

Table 9. 80960CF PGA Package Thermal Characteristics

Parameter	Thermal Resistance — °C/Watt: Airflow — ft./min (m/sec) 0 (0)	200 (1.01)	400 (2.03)	600 (3.07)	800 (4.06)	1000 (5.07)
θ Junction-to-Case (Case measured as shown in Figure 4)	1.5	1.5	1.5	1.5	1.5	1.5
θ Case-to-Ambient (No Heatsink)	17	14	11	9	7.1	6.6
θ Case-to-Ambient (With Heatsink)*	13	9	5.5	5	3.9	3.4



NOTES:

1. This table applies to 80960CF PGA plugged into socket or soldered directly to board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$

*0.285" high unidirectional heatsink (Al alloy 6061, 50 mil fin width, 150 mil center-to-center fin spacing).

3.5 Stepping Register Information

Upon reset, register g0 contains die stepping information (Figure 5). The most significant byte contains ASCII 0; the upper middle byte contains an ASCII C; the lower middle byte contains an ASCII F. The least significant byte contains the stepping number in ASCII. g0 retains this information until it is overwritten by the user program. Table 10 contains a cross reference of the number in the least significant byte of register g0 to the die stepping number.

ASCII	00	43	46	Stepping Number
DECIMAL	0	C	F	Stepping Number
	MSB			LSB

Figure 5. Register g0

Table 10. Die Stepping Cross Reference

g0 Least Significant Byte	Die Stepping
01	A
02	B
03	C
04	D
05	E

3.6 Sources for Accessories

The following is a list of suggested sources for 80960CF accessories. This is neither an endorsement or a warranty of the performance of any of the listed products and/or companies.

Sockets

1. 3M Textool Test and Interconnection Products
   P.O. Box 2963
   Austin, TX 78769-2963
2. Augat, Inc. Interconnection Products Group
   33 Perry Avenue, P.O. box 779
   Attleboro, MA 02703
   (508) 699-7646
3. Concept Mfg, Inc. Decoupling Sockets)
   41484 Christy Street
   Fremont, CA 94538
   (415) 651-3804

Heatsinks/Fins

1. Thermalloy, Inc.
   2021 West Valley View Lane
   Dallas, TX 75234-8993
   (214) 243-4321 FAX: (214) 241-4656
2. E G & G Division
   60 Audubon Road
   Wakefield, MA 01880
   (617) 245-5900

272493-20

4.0 ELECTRICAL SPECIFICATIONS

4.1 Absolute Maximum Ratings

Parameter	Maximum Rating
Storage Temperature	−65°C to +150°C
Case Temperature Under Bias	−65°C to +110°C
Supply Voltage wrt. V_{SS}	−0.5V to + 6.5V
Voltage on Other Pins wrt. V_{SS}	−0.5V to V_{CC} + 0.5V

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. It is valid for the devices indicated in the revision history. The specifications are subject to change without notice.

***WARNING:** Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

4.2 Operating Conditions

Table 11. Operating Conditions (80960CF-40)

Symbol	Parameter		Min	Max	Units	Notes
V_{CC}	Supply Voltage	80960CF-40	4.75	5.25	V	
f_{CLK2x}	Input Clock Frequency (2-x Mode)	80960CF-40	0	80	MHz	
f_{CLK1x}	Input Clock Frequency (1-x Mode)	80960CF-40	8	40	MHz	(1)
T_C	Case Temp Under Bias, PGA Pkg.	80960CF-40	0	85	°C	

NOTES:

1. When in the 1-x input clock mode, CLKIN is an input to an internal phase-locked loop and must maintain a minimum frequency of 8 MHz for proper processor operation. However, in the 1-x mode, CLKIN may still be stopped when the processor is in a reset condition. If CLKIN is stopped, the specified RESET low time must be provided once CLKIN restarts and has stabilized.

4.3 Recommended Connections

Power and ground connections must be made to multiple V_{CC} and V_{SS} (GND) pins. Every 80960CF-based circuit board should include power (V_{CC}) and ground (V_{SS}) planes for power distribution. Every V_{CC} pin must be connected to the power plane, and every V_{SS} pin must be connected to the ground plane. Pins identified as "NC" **must not** be connected in the system.

Liberal decoupling capacitance should be placed near the 80960CF. The processor can cause transient power surges when its numerous output buffers transition, particularly when connected to large capacitive loads.

Low inductance capacitors and interconnects are recommended for best high frequency electrical performance. Inductance can be reduced by shortening the board traces between the processor and decoupling capacitors as much as possible. Capacitors specifically designed for PGA packages will offer the lowest possible inductance.

For reliable operation, always connect unused inputs to an appropriate signal level. In particular, any unused interrupt (XINT, NMI) or DMA (DREQ) input should be connected to V_{CC} through a pull-up resistor, as should BTERM if not used. Pull-up resistors should be in the in the range of 20 KΩ for each pin tied high. If READY or HOLD are not used, the unused input should be connected to ground. **N.C. pins must always remain unconnected.** For additional information refer to the *i960® Cx Microprocessor User's Guide* (#270710).

272493-21

4.4 DC Specifications

Table 12. DC Characteristics

(80960CF-40 under the conditions described in **Section 4.2, Operating Conditions.**)

Symbol	Parameter	Min	Max	Units	Notes
V_{IL}	Input Low Voltage for all pins except $\overline{RESET}$	−0.3	+0.8	V	
V_{IH}	Input High Voltage for all pins except $\overline{RESET}$	2.0	V_{CC} + 0.3	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 5 mA
V_{OH}	Output High Voltage I_{OH} = −1 mA I_{OH} = −200 µA	2.4 V_{CC} − 0.5		V V	
V_{ILR}	Input Low Voltage for $\overline{RESET}$	−0.3	1.5	V	
V_{IHR}	Input High Voltage for $\overline{RESET}$	3.5	V_{CC} + 0.3	V	
I_{LI1}	Input Leakage Current for each pin *except*: $\overline{BTERM}$, $\overline{ONCE}$, $\overline{DREQ3:0}$, STEST, $\overline{EOP3:0}$/$\overline{TC3:0}$, $\overline{NMI}$, $\overline{XINT7:0}$, $\overline{BOFF}$, $\overline{READY}$, HOLD, CLKMODE		±15	µA	0 ≤ V_{IN} ≤ V_{CC} (1)
I_{LI2}	Input Leakage Current for: $\overline{BTERM}$, $\overline{ONCE}$, $\overline{DREQ3:0}$, STEST, $\overline{EOP3:0}$/$\overline{TC3:0}$, $\overline{NMI}$, $\overline{XINT7:0}$, $\overline{BOFF}$	0	−300	µA	V_{IN} = 0.45V (2)
I_{LI3}	Input Leakage Current for: $\overline{READY}$, HOLD, CLKMODE	0	500	µA	V_{IN} = 2.4V (3,7)
I_{LO}	Output Leakage Current		±15	µA	0.45 ≤ V_{OUT} ≤ V_{CC}
I_{CC}	Supply Current (80960CF-40): I_{CC} Max I_{CC} Typ		1150 1000	mA mA	(4) (5)
I_{ONCE}	ONCE-mode Supply Current		200	mA	
C_{IN}	Input Capacitance for: CLKIN, $\overline{RESET}$, $\overline{ONCE}$, $\overline{READY}$, HOLD, $\overline{DREQ3:0}$, $\overline{BOFF}$, $\overline{XINT7:0}$, $\overline{NMI}$, $\overline{BTERM}$, CLKMODE	0	12	pF	F_C = 1 MHz
C_{OUT}	Output Capacitance of each output pin		12	pF	F_C = 1 MHz (6)
$C_{I/O}$	I/O Pin Capacitance		12	pF	F_C = 1 MHz

NOTES:

1. No pullup or pulldown.
2. These pins have internal pullup resistors.
3. These pins have internal pulldown resistors.
4. Measured at worst case frequency, V_{CC} and temperature, with device operating and outputs loaded to the test conditions described in **Section 4.5.1, AC TEST CONDITIONS.**
5. I_{CC} Typical is not tested.
6. Output Capacitance is the capacitive load of a floating output.
7. CLKMODE pin has a pulldown resistor only when $\overline{ONCE}$ pin is deasserted.

272493-22

4.5 AC Specifications

Table 13. 80960CF AC Characteristics (40 MHz) (Sheet 1 of 2)

(80960CF-40 only, per the conditions in **4.2 Operating Conditions** and **4.5.1 AC TEST CONDITIONS**.)

Symbol	Parameter		Min	Max	Units	Notes
Input Clock (1,9)						
T_F	CLKIN Frequency		0	80	MHz	
T_C	CLKIN Period	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	25 12.5	125 ∞	ns ns	(11)
T_{CS}	CLKIN Period Stability	In 1-x Mode (f_{CLK1x})		±0.1%	Δ	(12)
T_{CH}	CLKIN High Time	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	5 5	62.5 ∞	ns ns	(11)
T_{CL}	CLKIN Low Time	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	5 5	62.5 ∞	ns ns	(11)
T_{CR}	CLKIN Rise Time		0	6	ns	
T_{CF}	CLKIN Fall Time		0	6	ns	
Output Clocks (1,8)						
T_{CP}	CLKIN to PCLK2:1 Delay	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	−2 2	2 25	ns ns	(3,12) (3)
T	PCLK2:1 Period	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	T_C $2T_C$		ns ns	(12) (3)
T_{PH}	PCLK2:1 High Time		(T/2) − 2	T/2	ns	(12)
T_{PL}	PCLK2:1 Low Time		(T/2) − 2	T/2	ns	(12)
T_{PR}	PCLK2:1 Rise Time		1	4	ns	(3)
T_{PF}	PCLK2:1 Fall Time		1	4	ns	(3)
Synchronous Outputs (8)						
T_{OH} T_{OV}	Output Valid Delay, Output Hold					(6,10)
	T_{OH1}, T_{OV1}	A31:2	3	14	ns	
	T_{OH2}, T_{OV2}	$\overline{BE3:0}$	3	16	ns	
	T_{OH3}, T_{OV3}	$\overline{ADS}$	6	16	ns	
	T_{OH4}, T_{OV4}	W/$\overline{R}$	3	16	ns	
	T_{OH5}, T_{OV5}	D/$\overline{C}$, $\overline{SUP}$, $\overline{DMA}$	4	16	ns	
	T_{OH6}, T_{OV6}	$\overline{BLAST}$, $\overline{WAIT}$	5	16	ns	
	T_{OH7}, T_{OV7}	$\overline{DEN}$	3	16	ns	
	T_{OH8}, T_{OV8}	HOLDA, $\overline{BREQ}$	4	16	ns	
	T_{OH9}, T_{OV9}	$\overline{LOCK}$	4	16	ns	
	T_{OH10}, T_{OV10}	$\overline{DACK3:0}$	4	16	ns	
	T_{OH11}, T_{OV11}	D31:0	3	16	ns	
	T_{OH12}, T_{OV12}	DT/$\overline{R}$	T/2 + 3	T/2 + 14	ns	
	T_{OH13}, T_{OV13}	$\overline{FAIL}$	2	14	ns	
	T_{OH14}, T_{OV14}	$\overline{EOP3:0}$/$\overline{TC3:0}$	3	16	ns	(6,10)
T_{OF}	Output Float for all outputs		3	22	ns	(6)

NOTES:
See Table 14 (following this table) for all notes related to AC specifications.

272493-23

Table 13. 80960CF AC Characteristics (40 MHz) (Sheet 2 of 2)

(80960CF-40 only, per the conditions in **4.2 Operating Conditions** and **4.5.1 AC TEST CONDITIONS.**)

Symbol	Parameter	Min	Max	Units	Notes
Synchronous Inputs (1,9,10)					
T_{IS}	Input Setup				
T_{IS1}	D31:0	3		ns	
T_{IS2}	$\overline{BOFF}$	15		ns	
T_{IS3}	$\overline{BTERM}$/READY	7		ns	
T_{IS4}	HOLD	5		ns	
T_{IH}	Input Hold				
T_{IH1}	D31:0	5		ns	
T_{IH2}	$\overline{BOFF}$	5		ns	
T_{IH3}	$\overline{BTERM}$/READY	2		ns	
T_{IH4}	HOLD	3		ns	
Relative Output Timings (1,2,3,8)					
T_{AVSH1}	A31:2 Valid to $\overline{ADS}$ Rising	T – 4	T + 4	ns	
T_{AVSH2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, D/$\overline{C}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{ADS}$ Rising	T – 6	T + 6	ns	
T_{AVEL1}	A31:2 Valid to $\overline{DEN}$ Falling	T – 4	T + 4	ns	
T_{AVEL2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, $\overline{INST}$,$\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{DEN}$ Falling	T – 6	T + 6	ns	
T_{NLQV}	$\overline{WAIT}$ Falling to Output Data Valid	± 6		ns	
T_{DVNH}	Output Data Valid to $\overline{WAIT}$ Rising	N*T – 6	N*T + 6	ns	(4)
T_{NLNH}	$\overline{WAIT}$ Falling to $\overline{WAIT}$ Rising	N*T ± 4		ns	(4)
T_{NHQX}	Output Data Hold after $\overline{WAIT}$ Rising	(N+1)*T–8	(N+1)*T+6	ns	(5)
T_{EHTV}	DT/$\overline{R}$ Hold after $\overline{DEN}$ High	T/2 – 7	∞	ns	(6)
T_{TVEL}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Falling	T/2 – 4		ns	
Relative Input Timings (1,2,3)					
T_{IS5}	$\overline{RESET}$ Input Setup (2-x Clock Mode)	6		ns	(13)
T_{IH5}	$\overline{RESET}$ Input Hold (2-x Clock Mode)	5		ns	(13)
T_{IS6}	$\overline{DREQ3:0}$ Input Setup	12		ns	(7)
T_{IH6}	$\overline{DREQ3:0}$ Input Hold	7		ns	(7)
T_{IS7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Setup	7		ns	(15)
T_{IH7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Hold	3		ns	(15)
T_{IS8}	$\overline{RESET}$ Input Setup (1-x Clock Mode)	3		ns	(14)
T_{IH8}	$\overline{RESET}$ Input Hold (1-x Clock Mode)	T/4 + 1		ns	(14)

NOTES:
See Table 14 (following this table) for all notes related to AC specifications.

272493-24

Table 14. AC Characteristics Notes

NOTES:

1. See **Section 4.5.2, AC TIMING WAVEFORMS** for waveforms and definitions.
2. See Figure 15 for capacitive derating information for output delays and hold times.
3. See Figure 16 for capacitive derating information for rise and fall times.
4. Where N is the number of N_{RAD}, N_{RDD}, N_{WAD} or N_{WDD} wait states that are programmed in the Bus Controller Region Table. $\overline{WAIT}$ never goes active when there are no wait states in an access.
5. N = Number of wait states inserted with $\overline{READY}$.
6. Output Data and/or DT/$\overline{R}$ may be driven indefinitely following a cycle if there is no subsequent bus activity.
7. Since asynchronous inputs are synchronized internally by the 80960CF, they have no required setup or hold times to be recognized and for proper operation. However, to guarantee recognition of the input at a particular edge of PCLK2:1, the setup times shown must be met. Asynchronous inputs must be active for at least two consecutive PCLK2:1 rising edges to be seen by the processor.
8. These specifications are guaranteed by the processor.
9. These specifications must be met by the system for proper operation of the processor.
10. This timing is dependent upon the loading of PCLK2:1. Use the derating curves of **Section 4.5.3, DERATING CURVES** to adjust the timing for PCLK2:1 loading.
11. In the 1-x input clock mode, the maximum input clock period is limited to 125 ns while the processor is operating. When the processor is in reset, the input clock may stop even in 1-x mode.
12. When in the 1-x input clock mode, these specifications assume a stable input clock with a period variation of less than ± 0.1% between adjacent cycles.
13. In 2-x clock mode, $\overline{RESET}$ is an asynchronous input which has no required setup and hold time for proper operation. However, to guarantee the device exits reset synchronized to a particular clock edge, the $\overline{RESET}$ pin must meet setup and hold times to the falling edge of the CLKIN. (See Figure 21.)
14. In 1-x clock mode, $\overline{RESET}$ is an asynchronous input which has no required setup and hold time for proper operation. However, to guarantee the device exits reset synchronized to a particular clock edge, the $\overline{RESET}$ pin must meet setup and hold times to the rising edge of the CLKIN. (See Figure 22.)
15. The interrupt pins are synchronized internally by the 80960CF. They have no required setup or hold times for proper operation. These pins are sampled by the interrupt controller every other clock and must be active for at least three consecutive PCLK2:1 periods when asserting them asynchronously. To guarantee recognition at a particular clock edge, the setup and hold times shown must be met for two consecutive PCLK2:1 falling edges.

272493-25

4.5.1 AC TEST CONDITIONS

The AC Specifications in Section 4.5 are tested with the 50 pF load shown in Figure 6. Figure 15 shows how timings vary with load capacitance.

Specifications are measured at the 1.5V crossing point, unless otherwise indicated. Input waveforms are assumed to have a rise and fall time of ≤ 2 ns from 0.8V to 2.0V. See **Section 4.5.2, AC TIMING WAVEFORMS** for AC specification definitions, test points and illustrations.

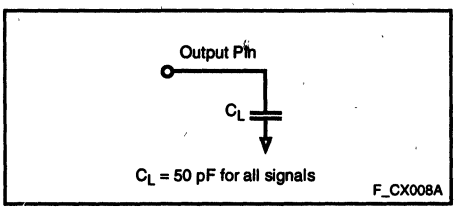


Figure 6. AC Test Load

4.5.2 AC TIMING WAVEFORMS

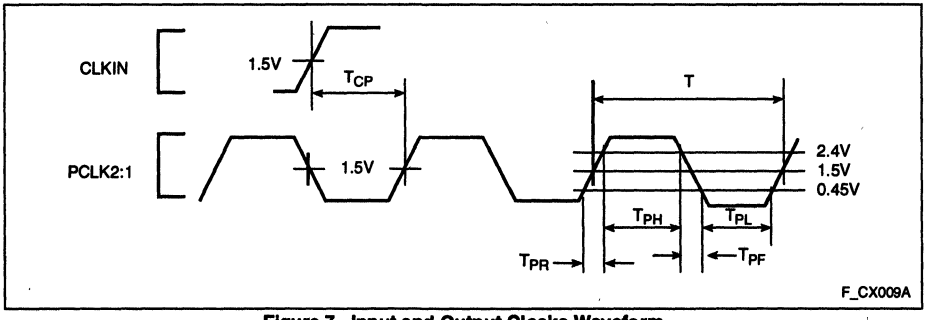


Figure 7. Input and Output Clocks Waveform

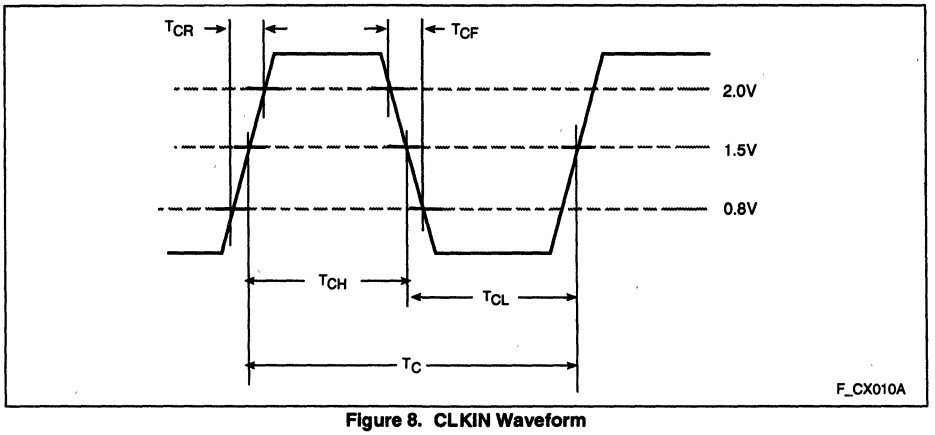
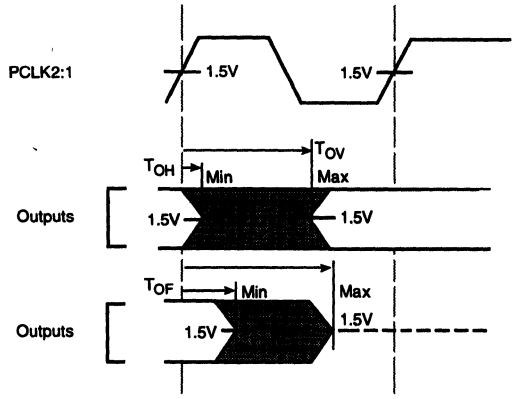


Figure 8. CLKIN Waveform

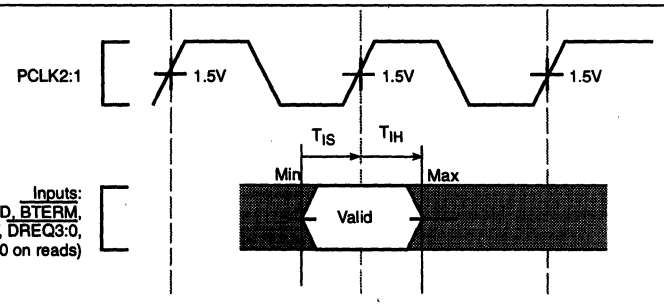
272493-26



NOTES:

1. T_{OV} T_{OH} - OUTPUT DELAY - Maximum output delay is referred to as Output Valid Delay (T_{OV}); minimum output delay is referred to as Output Hold (T_{OH}).
2. T_{OF} - OUTPUT FLOAT DELAY - Output float condition occurs when the maximum output current becomes less that I_{LO} in magnitude.

Figure 9. Output Delay and Float Waveform



NOTES:

1. T_{IS} T_{IH} - INPUT SETUP AND HOLD - The input setup and hold requirements specify the sampling window during which synchronous inputs must be stable for correct processor operation.

Figure 10. Input Setup and Hold Waveform

272493-27

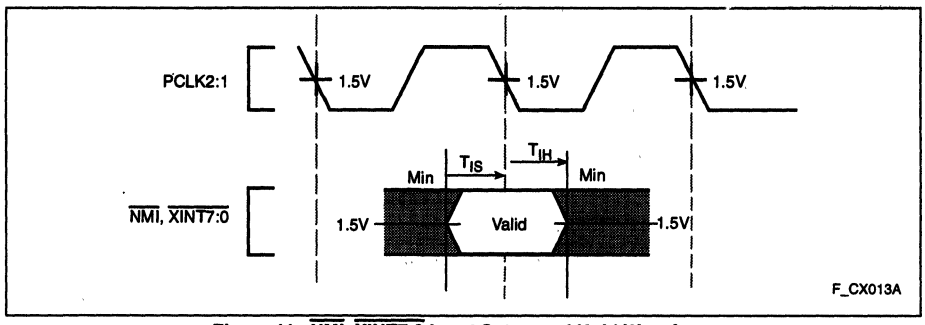
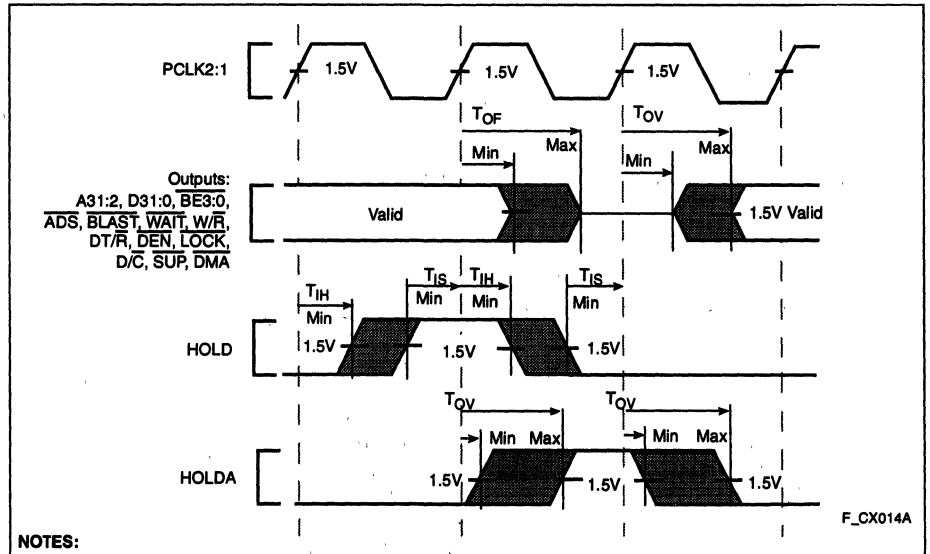


Figure 11. NMI, XINT7:0 Input Setup and Hold Waveform



NOTES:

1. T_{OV} T_{OH} - OUTPUT DELAY - Maximum output delay is referred to as Output Valid Delay (T_{OV}); minimum output delay is referred to as Output Hold (T_{OH}).
2. T_{OF} - OUTPUT FLOAT DELAY - Output float condition occurs when the maximum output current becomes less that I_{LO} in magnitude.
3. T_{IS} T_{IH} - INPUT SETUP AND HOLD - The input setup and hold requirements specify the sampling window during which synchronous inputs must be stable for correct processor operation.

Figure 12. Hold Acknowledge Timings

272493-28

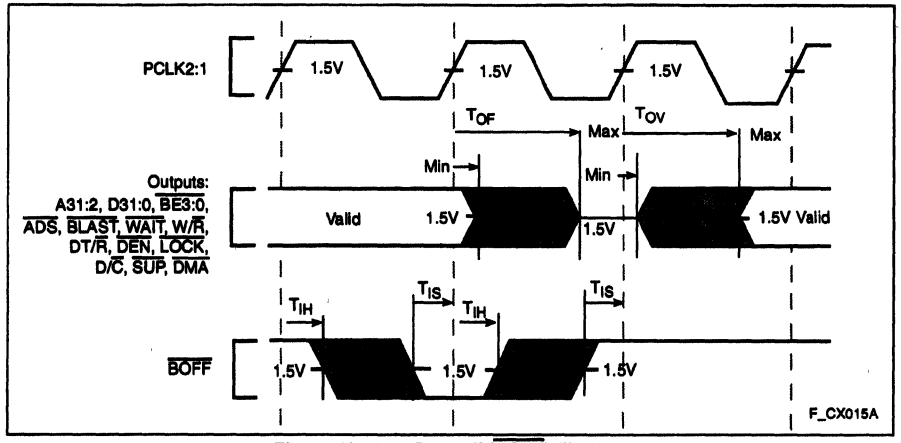


Figure 13. Bus Backoff (BOFF) Timings

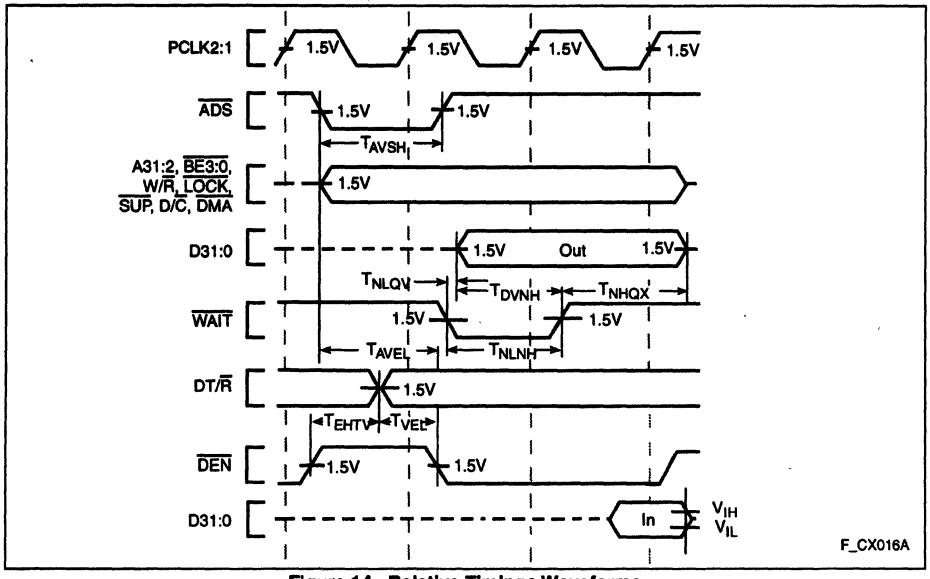


Figure 14. Relative Timings Waveforms

272493-29

4.5.3 DERATING CURVES

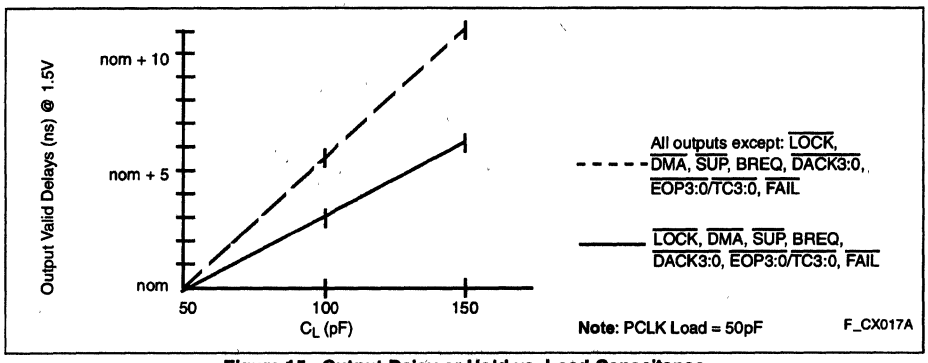


Figure 15. Output Delay or Hold vs. Load Capacitance

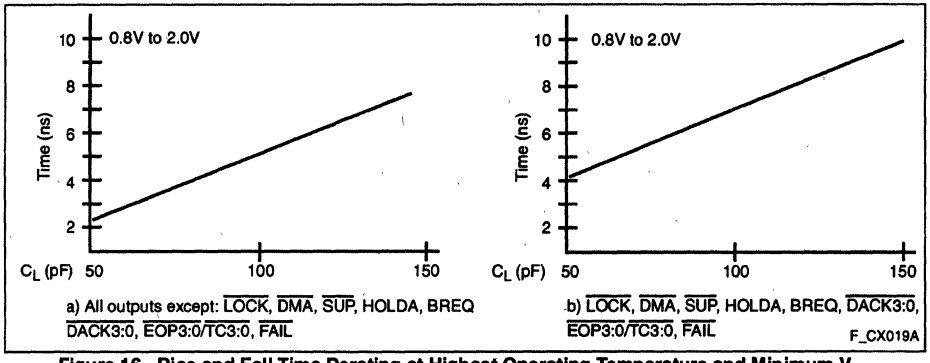


Figure 16. Rise and Fall Time Derating at Highest Operating Temperature and Minimum V_{CC}

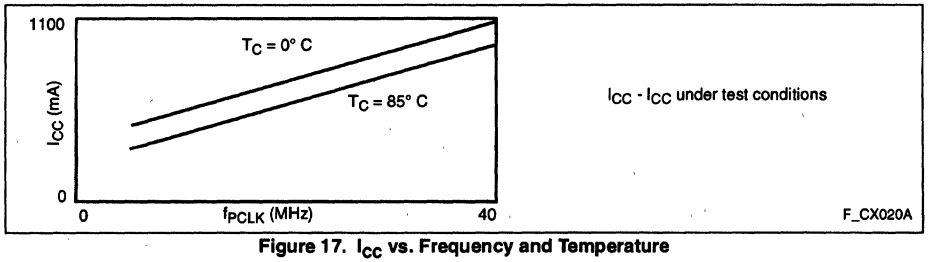


Figure 17. I_{CC} vs. Frequency and Temperature

272493-30

5.0 RESET, BACKOFF AND HOLD ACKNOWLEDGE

Table 15 lists the condition of each processor output pin while RESET is asserted (low). Table 16 lists the condition of each processor output pin while HOLDA is asserted (high).

In Table 15, with regard to bus output pin state only, the Hold Acknowledge state takes precedence over the reset state. Although asserting the RESET pin internally resets the processor, the processor's bus output pins do not enter the reset state if Hold Acknowledge has been granted to a previous HOLD request (HOLDA is active). Furthermore, the processor grants new HOLD requests and enters the Hold Acknowledge state even while in reset.

For example, if HOLD is asserted while HOLDA is inactive and the processor is in the reset state, the processor's bus pins enter the Hold Acknowledge state and HOLDA is granted. The processor is not able to perform memory accesses until the HOLD request is removed, even if the RESET pin is brought high. This operation is provided to simplify boot-up synchronization among multiple processors sharing the same bus.

Table 15. Reset Conditions (Sheet 1 of 2)

Pins	State During Reset (HOLDA Inactive)
A31:2	Floating
D31:0	Floating
$\overline{BE3:0}$	Driven high (Inactive)
W/$\overline{R}$	Driven low (Read)
$\overline{ADS}$	Driven high (Inactive)
$\overline{WAIT}$	Driven high (Inactive)
$\overline{BLAST}$	Driven low (Active)
DT/$\overline{R}$	Driven low (Receive)
$\overline{DEN}$	Driven high (Inactive)
$\overline{LOCK}$	Driven high (Inactive)
BREQ	Driven low (Inactive)
D/$\overline{C}$	Floating
$\overline{DMA}$	Floating

Table 15. Reset Conditions (Sheet 2 of 2)

Pins	State During Reset (HOLDA Inactive)
$\overline{SUP}$	Floating
$\overline{FAIL}$	Driven low (Active)
$\overline{DACK3:0}$	Driven high (Inactive)
$\overline{EOP3:0}$/$\overline{TC3:0}$	Floating (Set to input mode)

Table 16. Hold Acknowledge and Backoff Conditions

Pins	State During HOLDA
A31:2	Floating
D31:0	Floating
$\overline{BE3:0}$	Floating
W/$\overline{R}$	Floating
$\overline{ADS}$	Floating
$\overline{WAIT}$	Floating
$\overline{BLAST}$	Floating
DT/$\overline{R}$	Floating
$\overline{DEN}$	Floating
$\overline{LOCK}$	Floating
BREQ	Driven (High or low)
D/$\overline{C}$	Floating
$\overline{DMA}$	Floating
$\overline{SUP}$	Floating
$\overline{FAIL}$	Driven high (Inactive)
$\overline{DACK3:0}$	Driven high (Inactive)
$\overline{EOP3:0}$/$\overline{TC3:0}$	Driven (If output)

272493-31

6.0 BUS WAVEFORMS

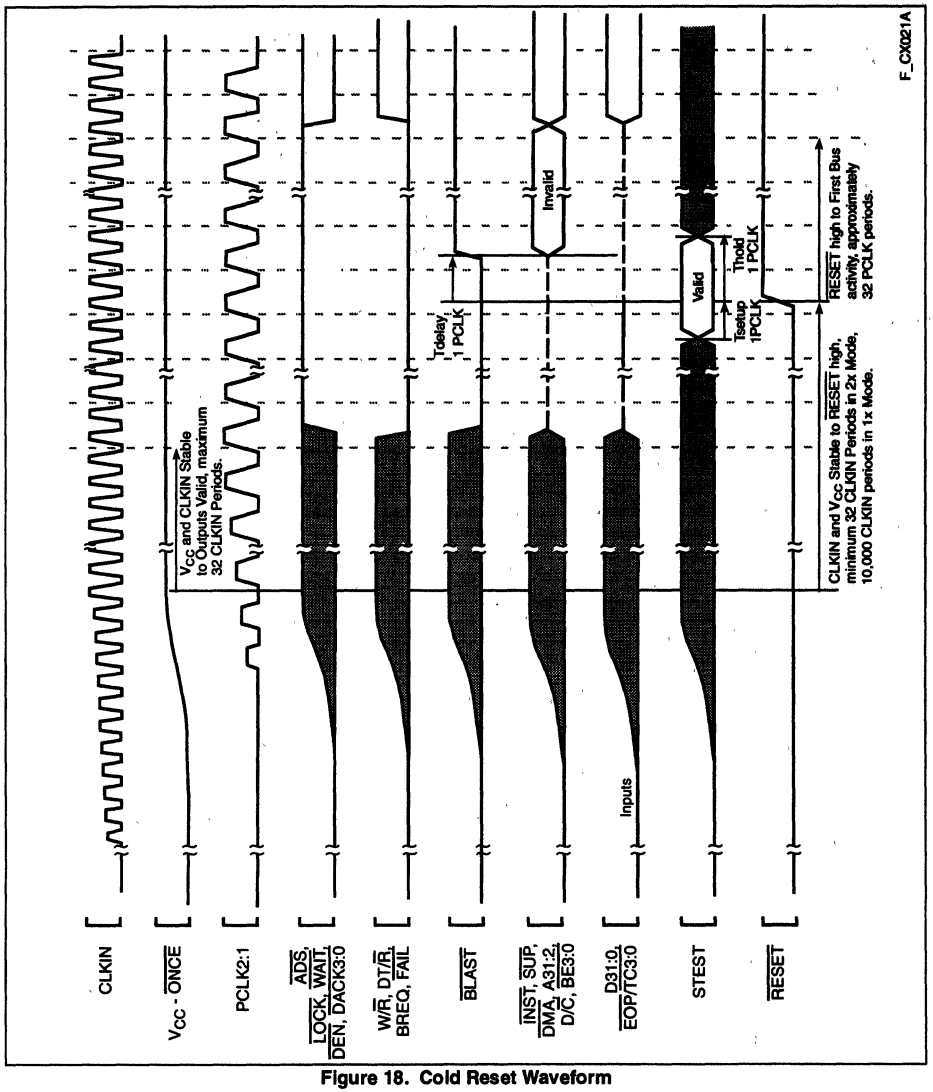


Figure 18. Cold Reset Waveform

272493-32

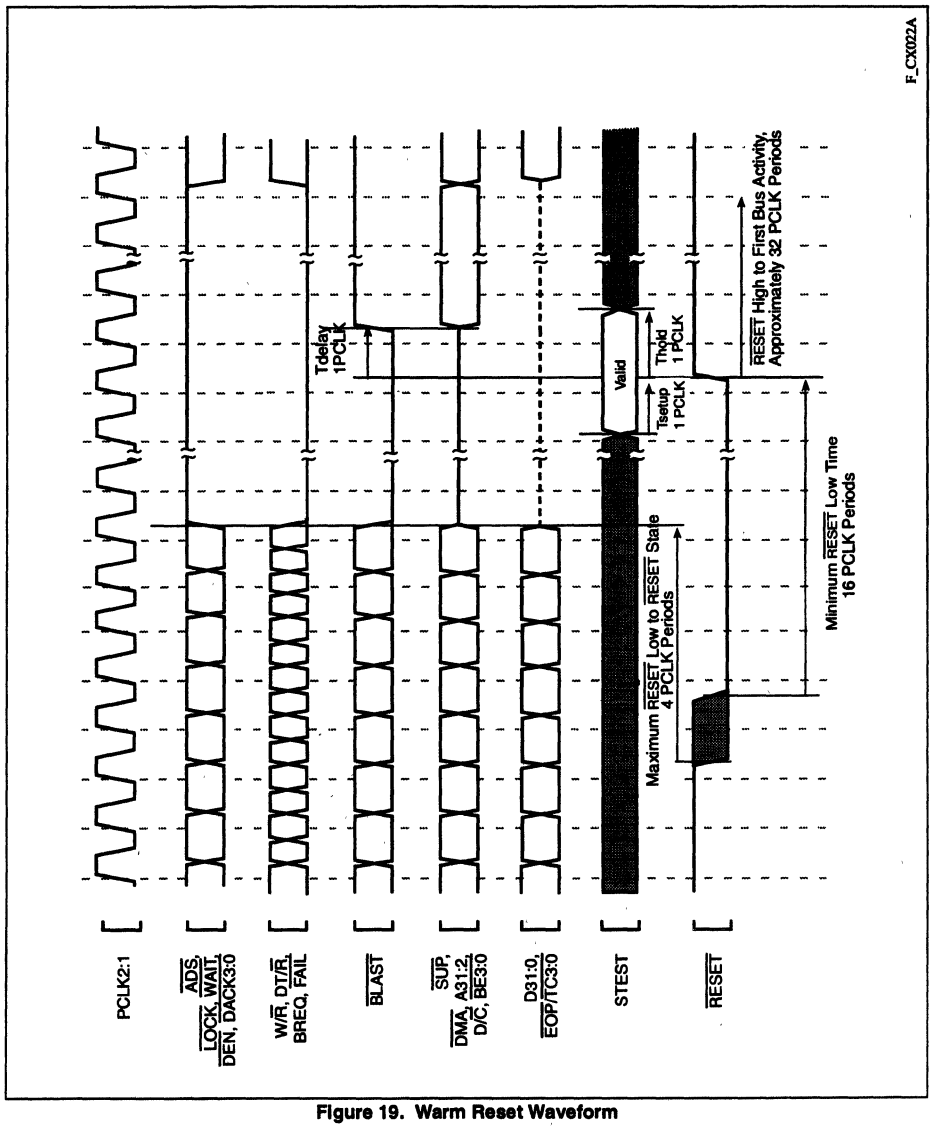


Figure 19. Warm Reset Waveform

272493-33

1

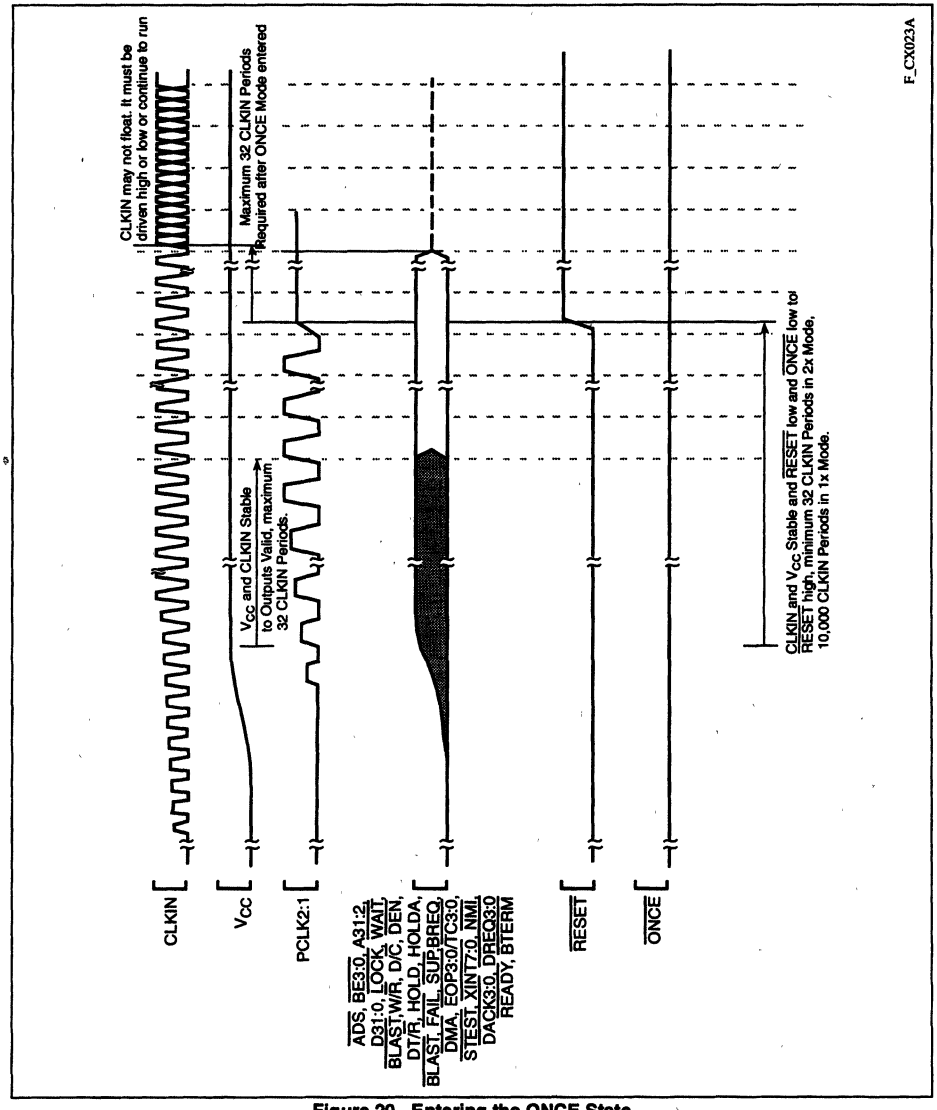
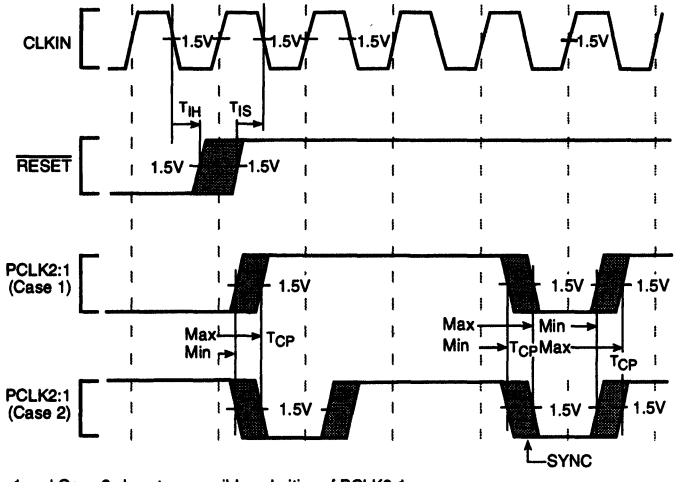


Figure 20. Entering the ONCE State

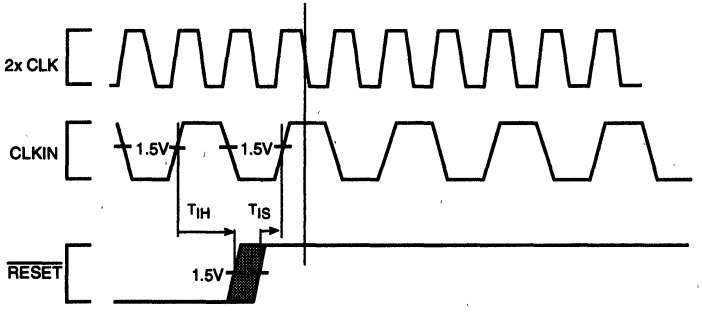
272493-34



Note: Case 1 and Case 2 show two possible polarities of PCLK2:1

F_CX024A

Figure 21. Clock Synchronization in the 2-x Clock Mode



Note: In 1x clock mode, the RESET pin is actually sampled on the falling edge of 2xCLK. 2xCLK is an internal signal generated by the PLL and is not available on an external pin. Therefore, RESET is specified relative to the rising edge of CLKIN. The RESET pin is sampled when PCLK is high.

F_CX025A

Figure 22. Clock Synchronization in the 1-x Clock Mode

272493-35

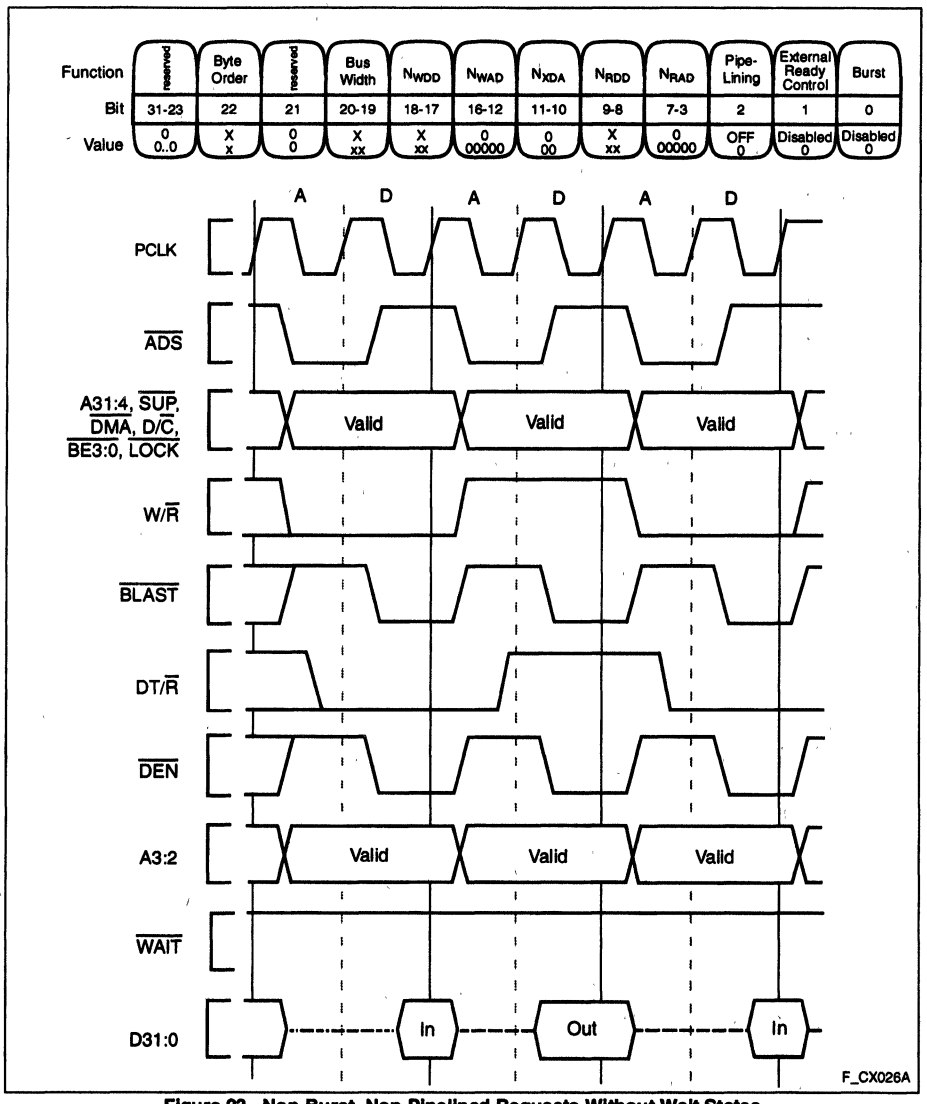


Figure 23. Non-Burst, Non-Pipelined Requests Without Wait States

272493-36

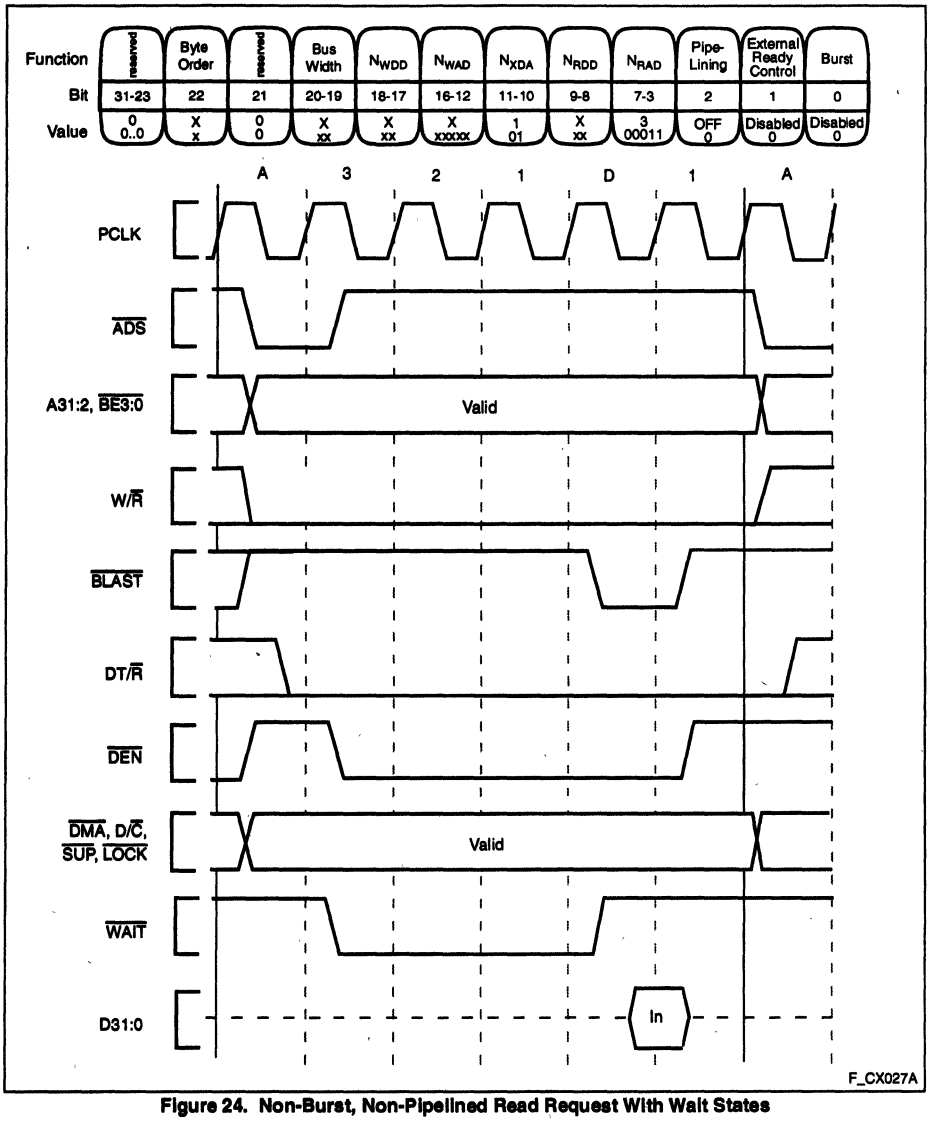


Figure 24. Non-Burst, Non-Pipelined Read Request With Wait States

272493-37

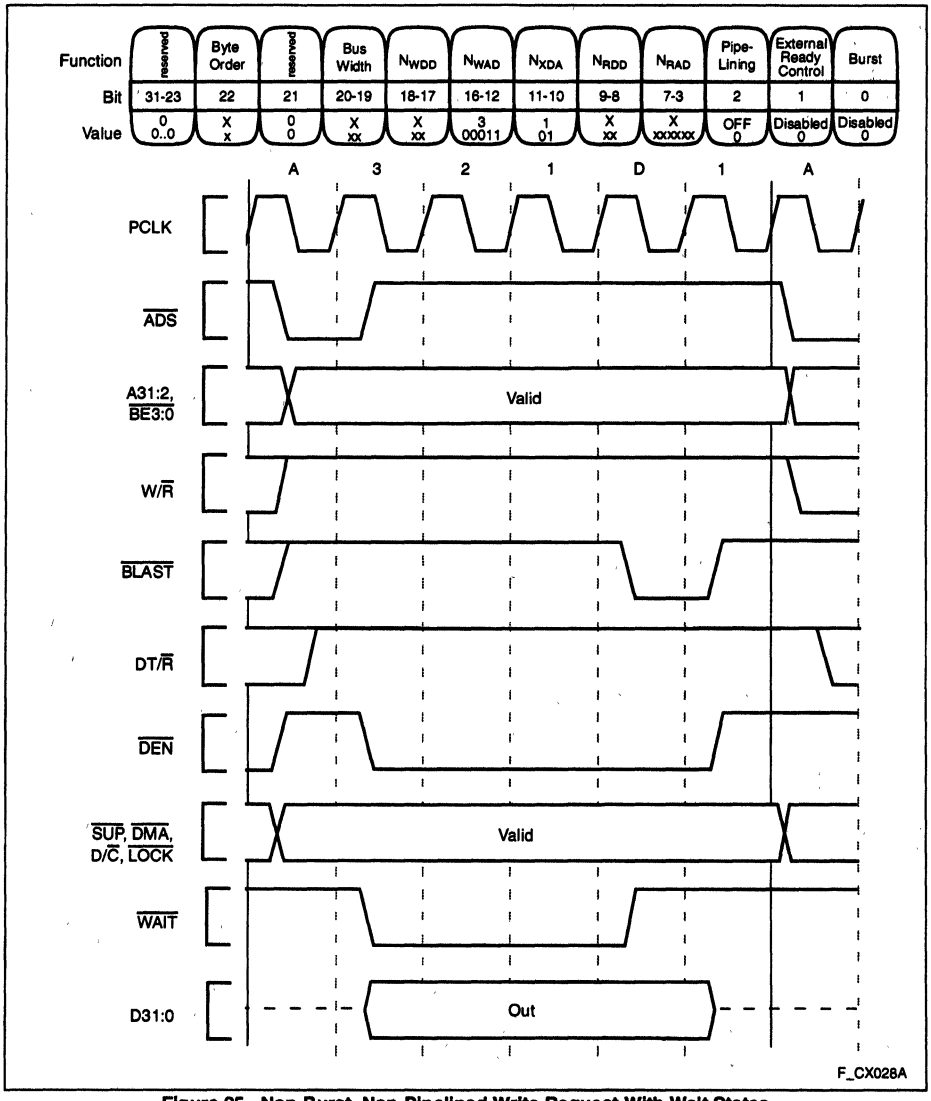


Figure 25. Non-Burst, Non-Pipelined Write Request With Wait States

272493-38

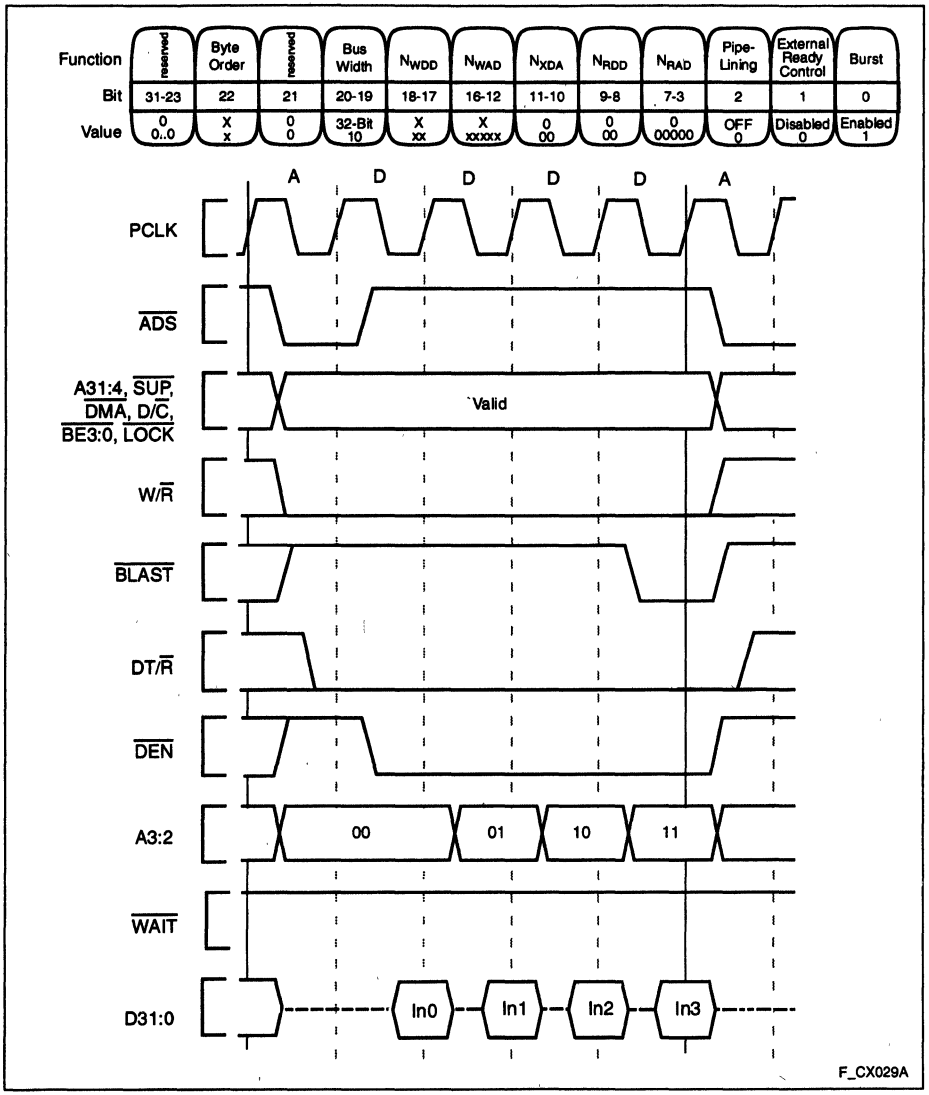


Figure 26. Burst, Non-Pipelined Read Request Without Wait States, 32-Bit Bus

272493-39

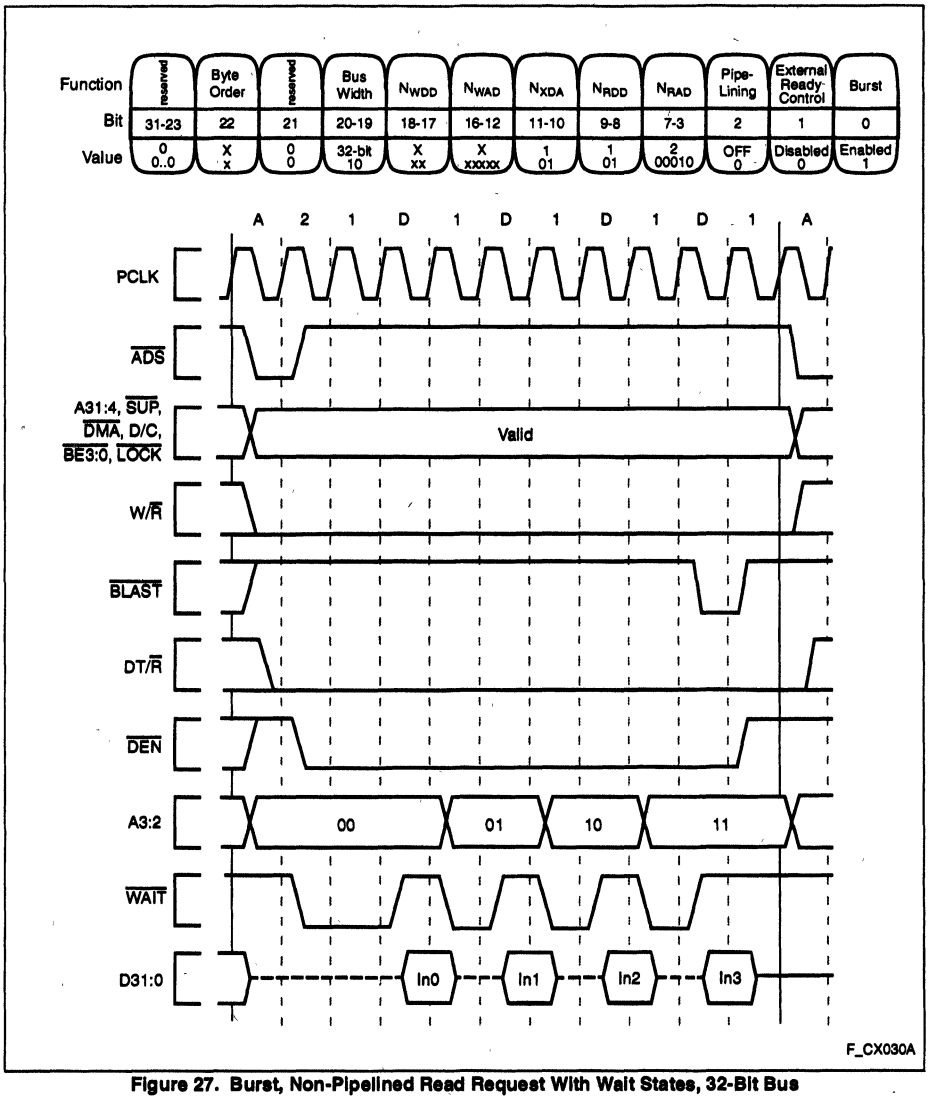


Figure 27. Burst, Non-Pipelined Read Request With Wait States, 32-Bit Bus

272493-40

Function	reserved	Byte Order	reserved	Bus Width	N_{WDD}	N_{WAD}	N_{XDA}	N_{RDD}	N_{RAD}	Pipe-Lining	External Ready Control	Burst
Bit	31-23	22	21	20-19	18-17	16-12	11-10	9-8	7-3	2	1	0
Value	0 0..0	X x	0 0	32-bit 10	0 00	0 00000	0 00	X xx	X xxxxx	OFF 0	Disabled 0	Enabled 1

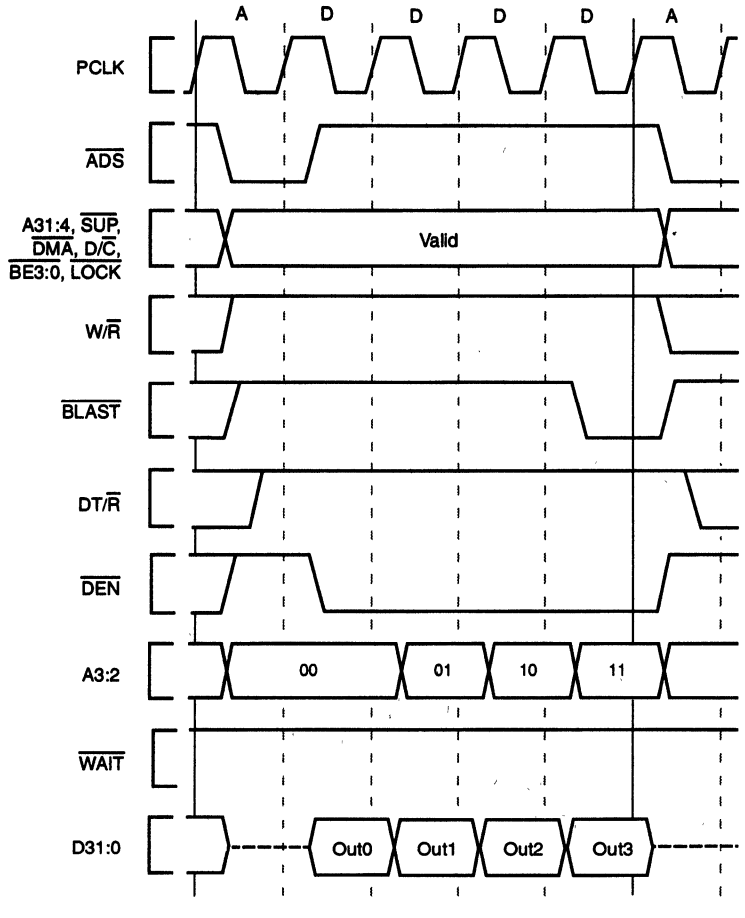


Figure 28. Burst, Non-Pipelined Write Request Without Wait States, 32-Bit Bus

272493-41

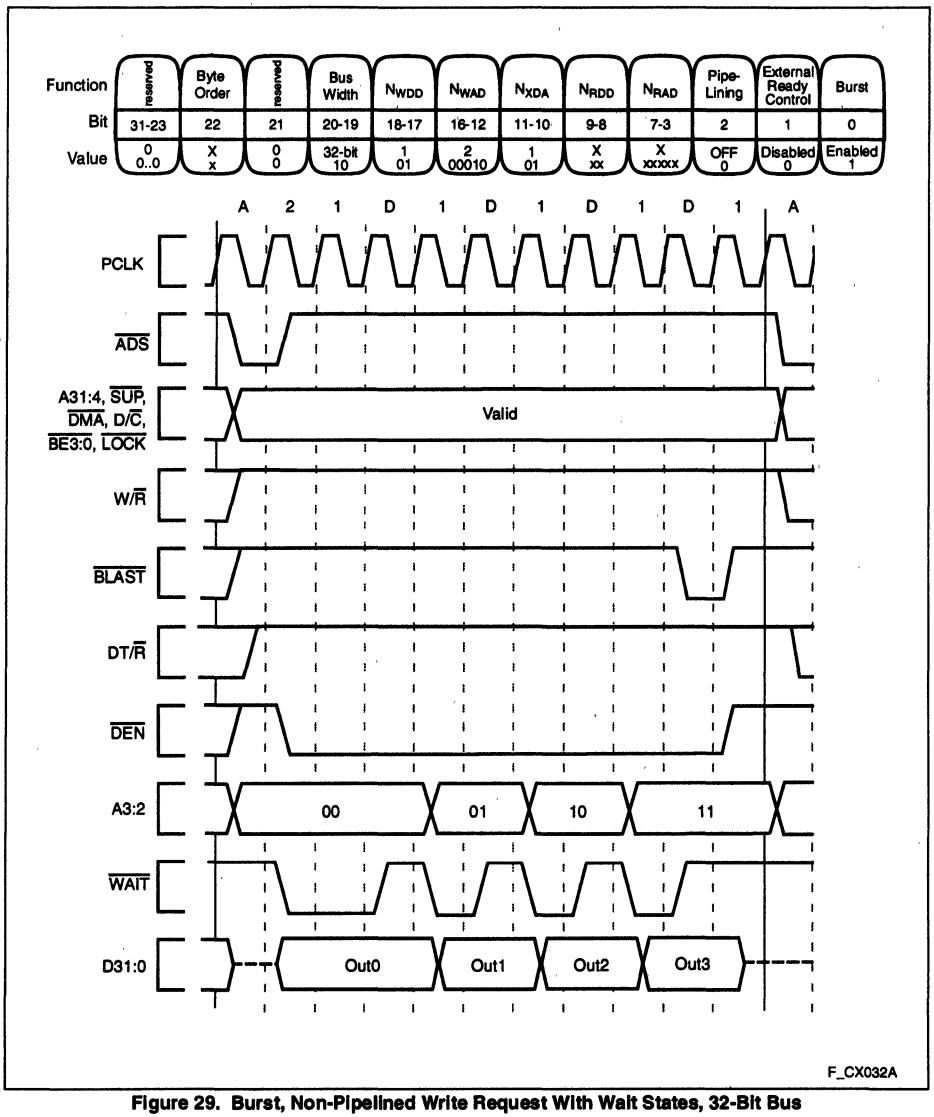


Figure 29. Burst, Non-Pipelined Write Request With Wait States, 32-Bit Bus

272493-42

Function	reserved	Byte Order	reserved	Bus Width	N_{WDD}	N_{WAD}	N_{XDA}	N_{RDD}	N_{RAD}	Pipe-Lining	External Ready Control	Burst
Bit	31-23	22	21	20-19	18-17	16-12	11-10	9-8	7-3	2	1	0
Value	0 0..0	X x	0 0	16-bit 01	X xx	X xxxxx	1 01	1 01	2 00010	OFF 0	Disabled 0	Enabled 1

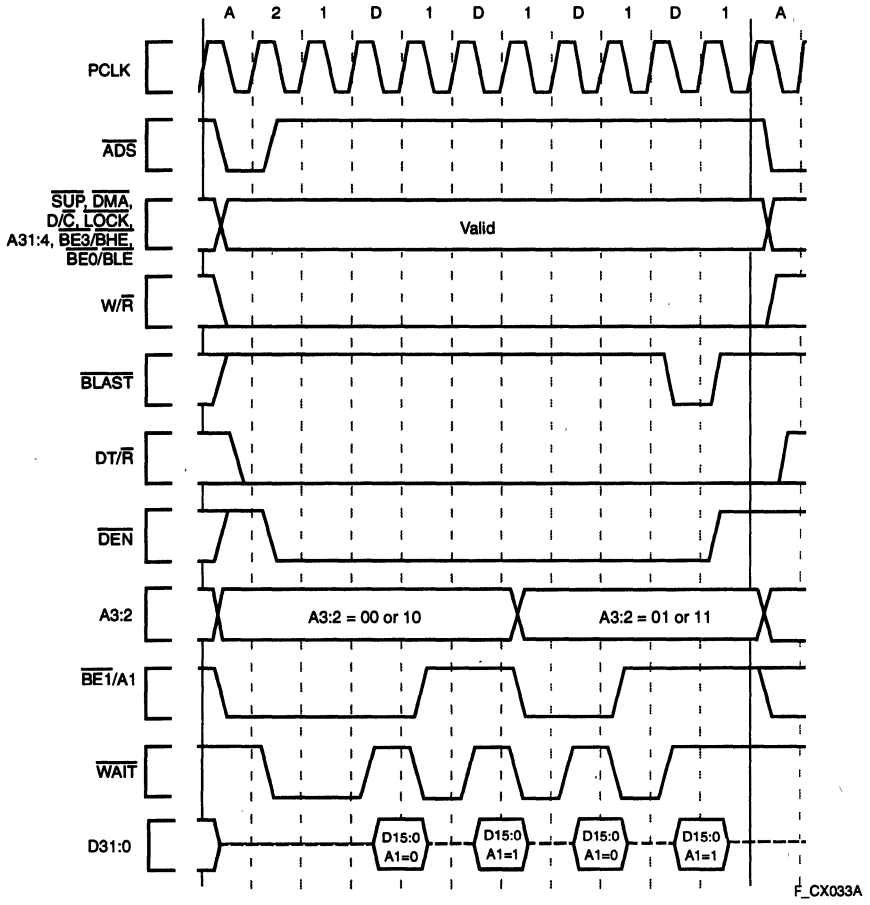


Figure 30. Burst, Non-Pipelined Read Request With Wait States, 16-Bit Bus

272493-43

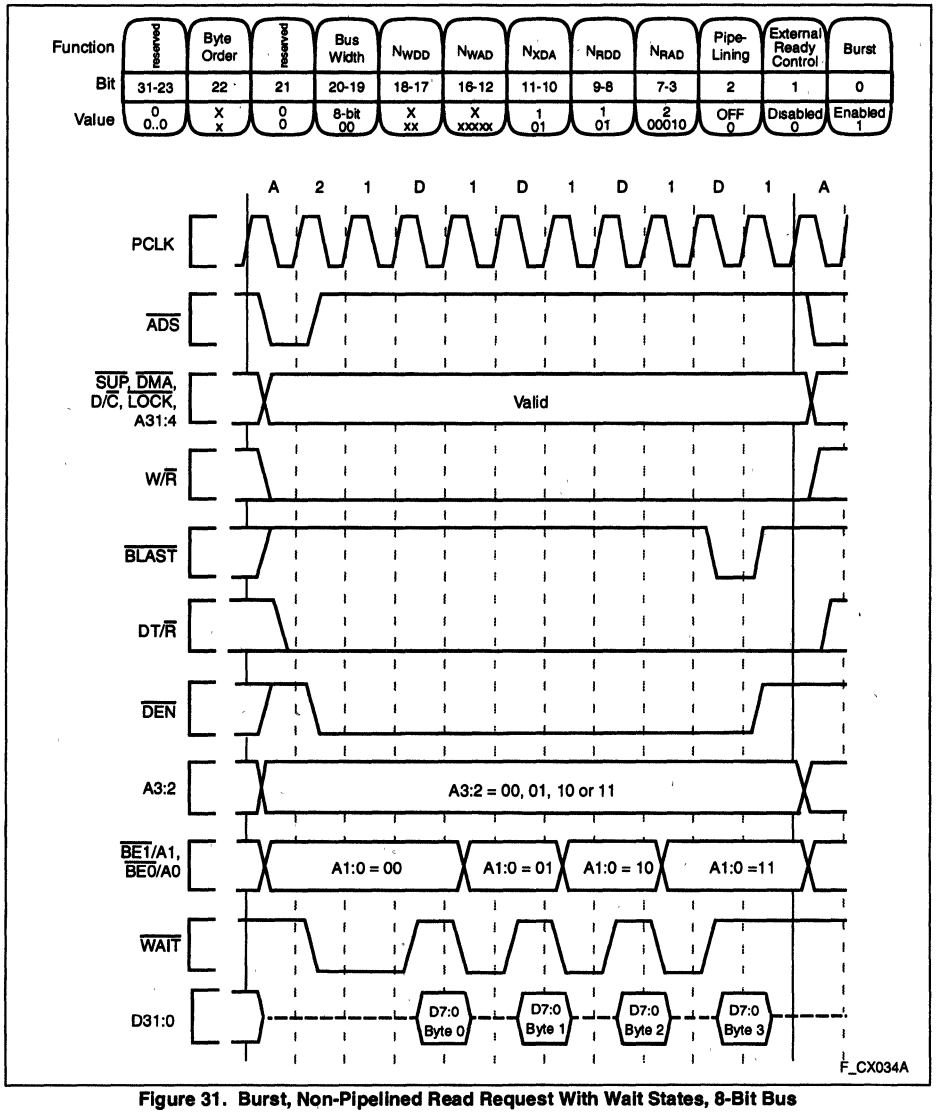


Figure 31. Burst, Non-Pipelined Read Request With Wait States, 8-Bit Bus

272493-44

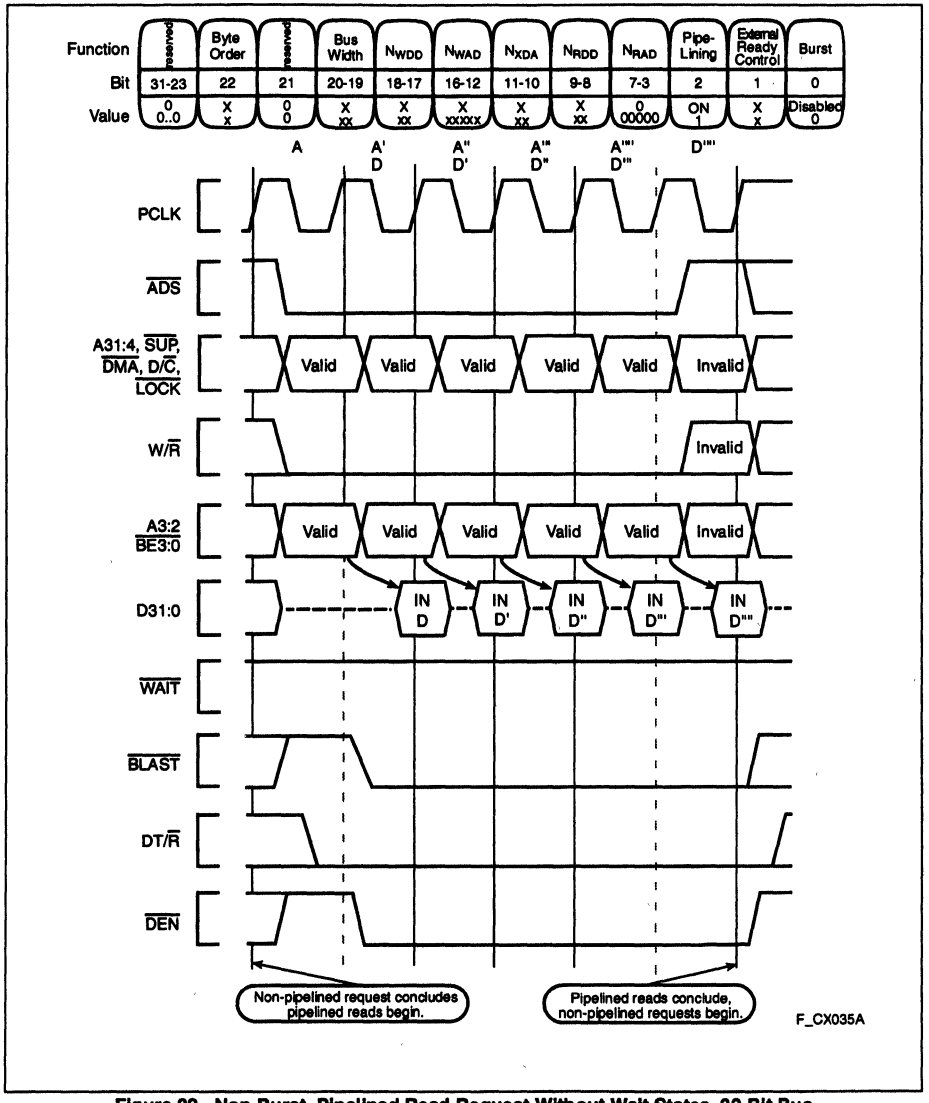


Figure 32. Non-Burst, Pipelined Read Request Without Wait States, 32-Bit Bus

272493-45

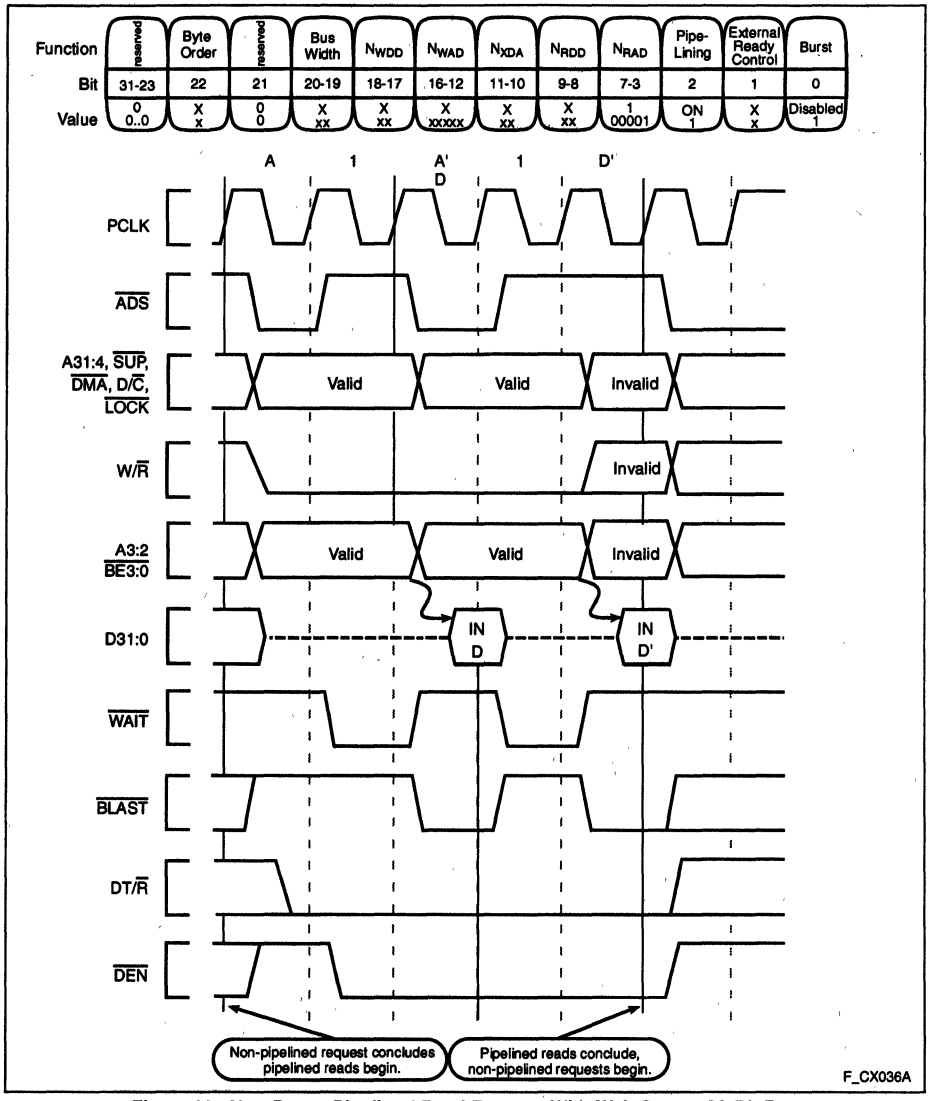


Figure 33. Non-Burst, Pipelined Read Request With Wait States, 32-Bit Bus

272493-46

Function	reserved	Byte Order	reserved	Bus Width	N_{WDD}	N_{WAD}	N_{XDA}	N_{RDD}	N_{RAD}	Pipe-Lining	External Ready Control	Burst
Bit	31-23	22	21	20-19	18-17	16-12	11-10	9-8	7-3	2	1	0
Value	0 0..0	X x	0 0	32-bit 10	X xx	X xxxxx	X xx	0 00	0 00000	ON 1	Disabled 0	Enabled 1

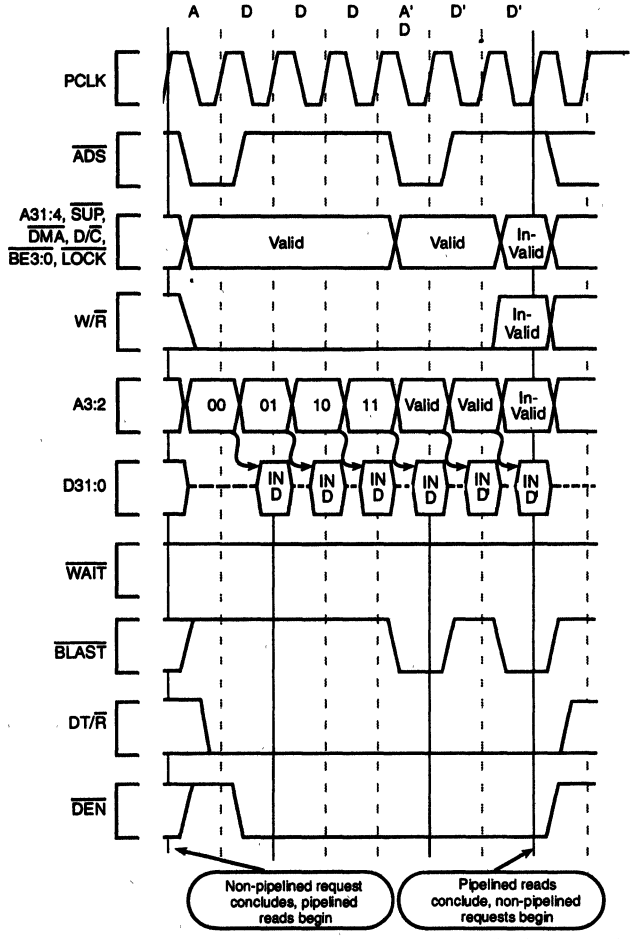


Figure 34. Burst, Pipelined Read Request Without Wait States, 32-Bit Bus

272493-47

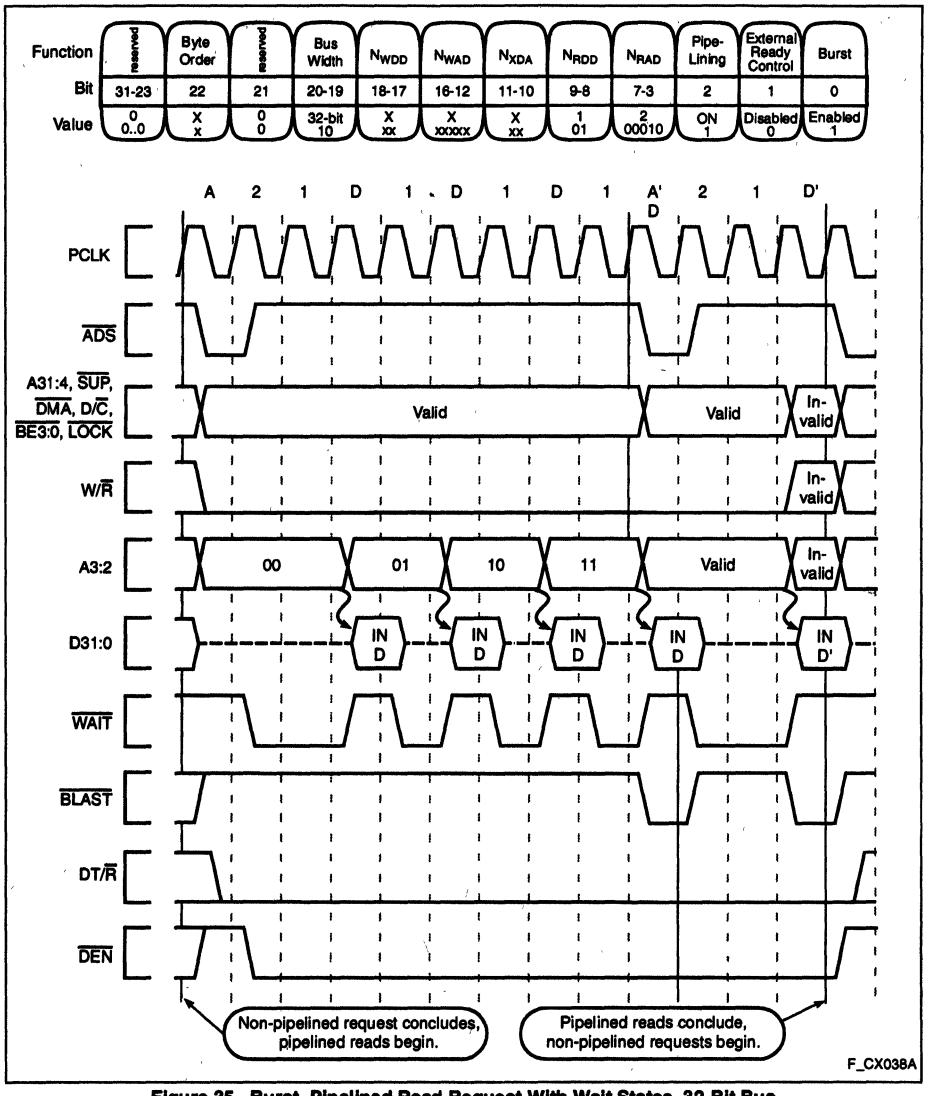


Figure 35. Burst, Pipelined Read Request With Wait States, 32-Bit Bus

272493-48

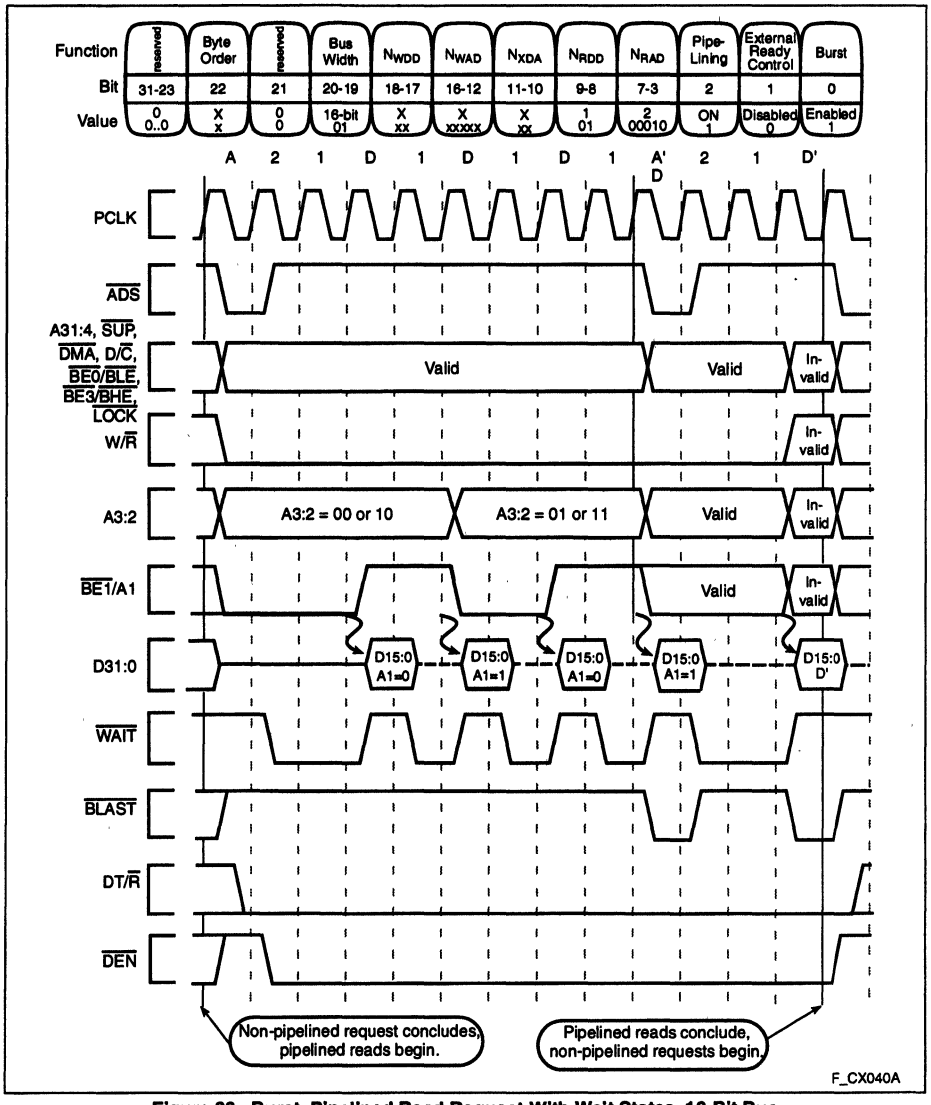


Figure 36. Burst, Pipelined Read Request With Wait States, 16-Bit Bus

272493-49

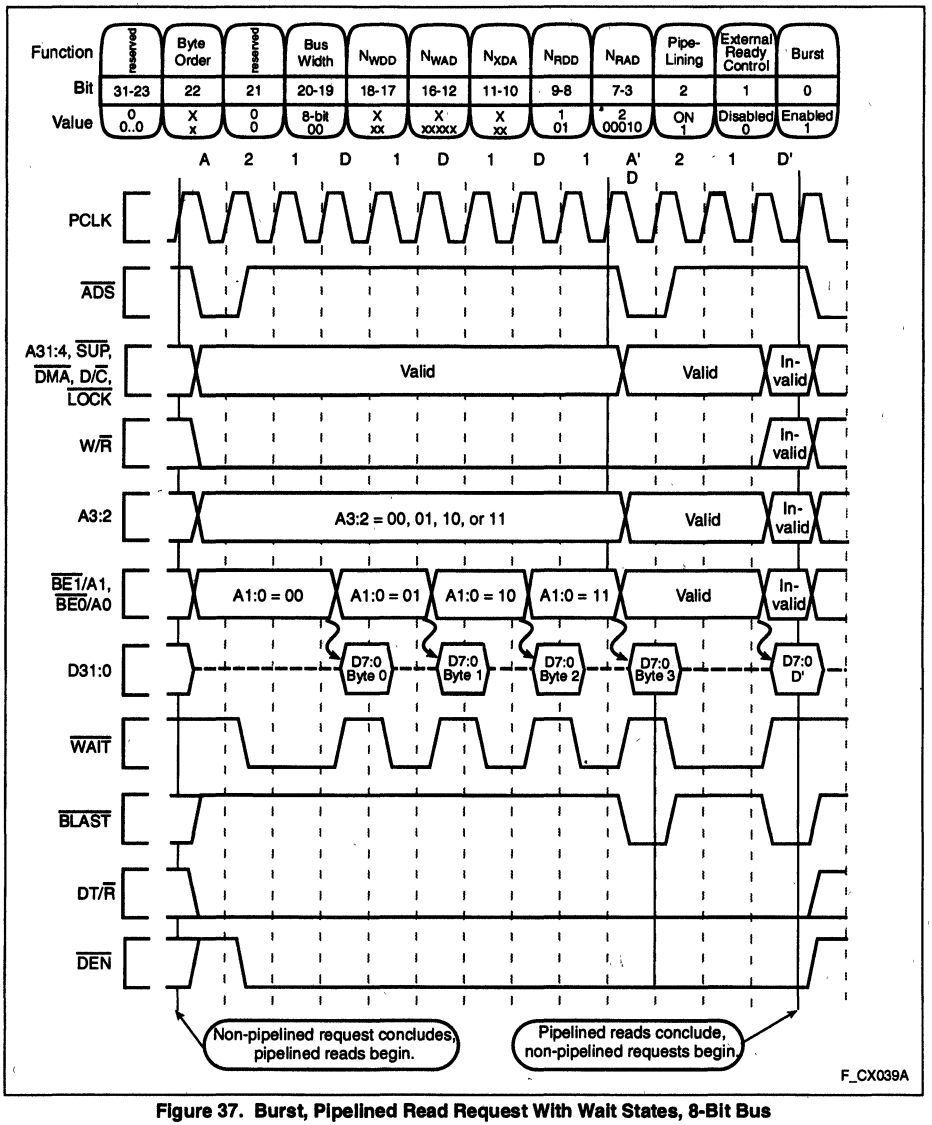


Figure 37. Burst, Pipelined Read Request With Wait States, 8-Bit Bus

272493-50

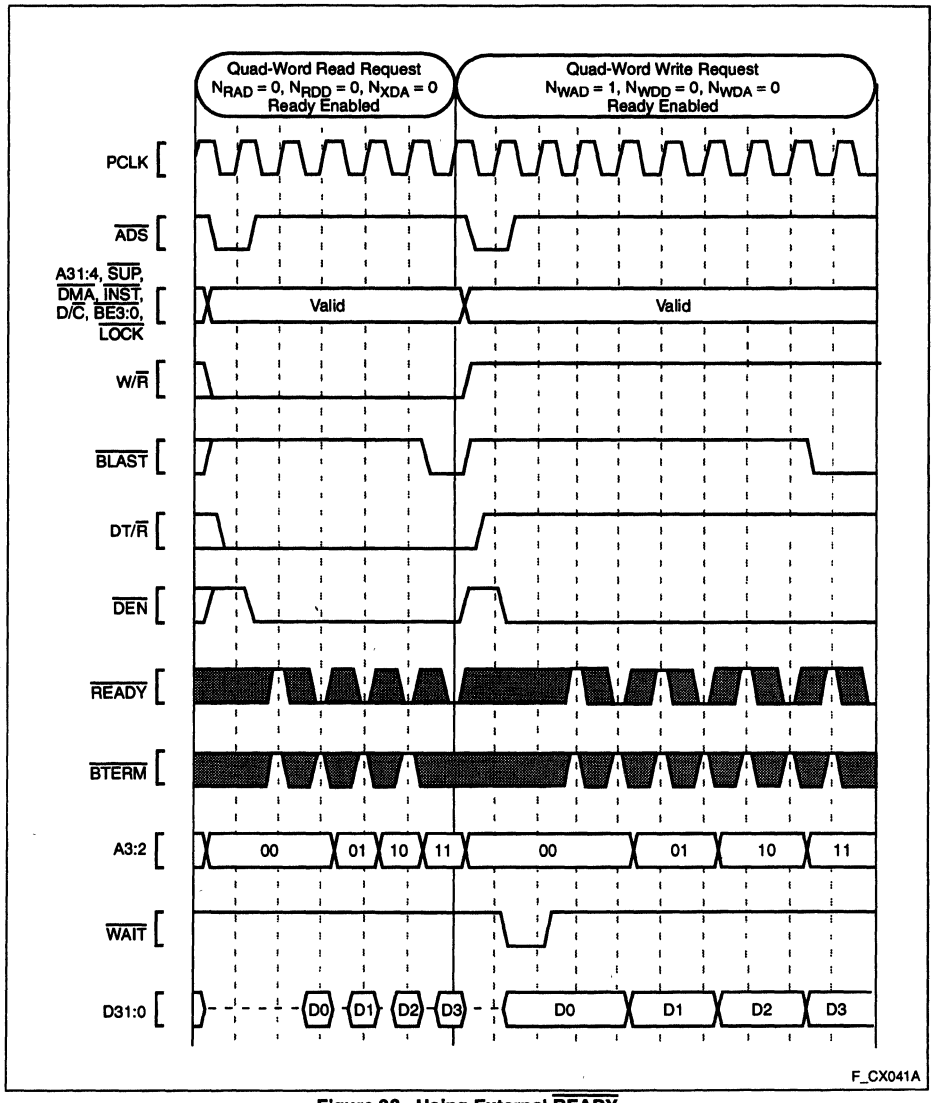


Figure 38. Using External READY

272493-51

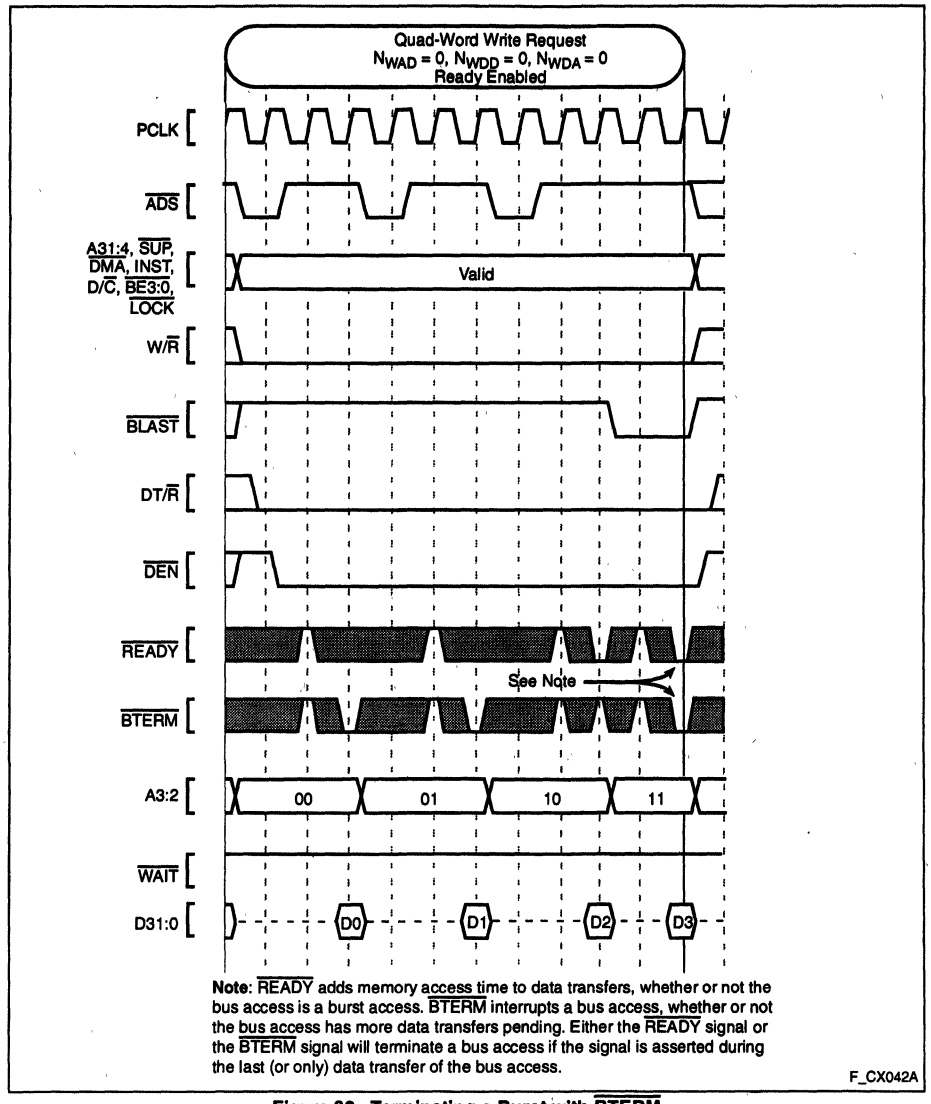


Figure 39. Terminating a Burst with BTERM

272493-52

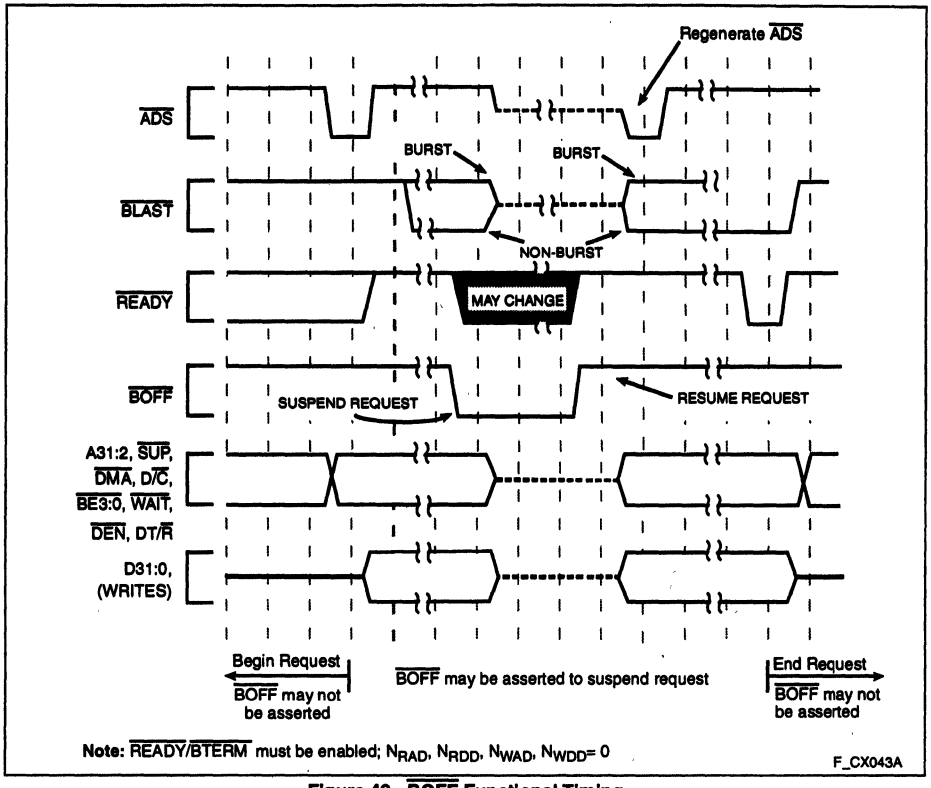


Figure 40. BOFF Functional Timing

272493-53

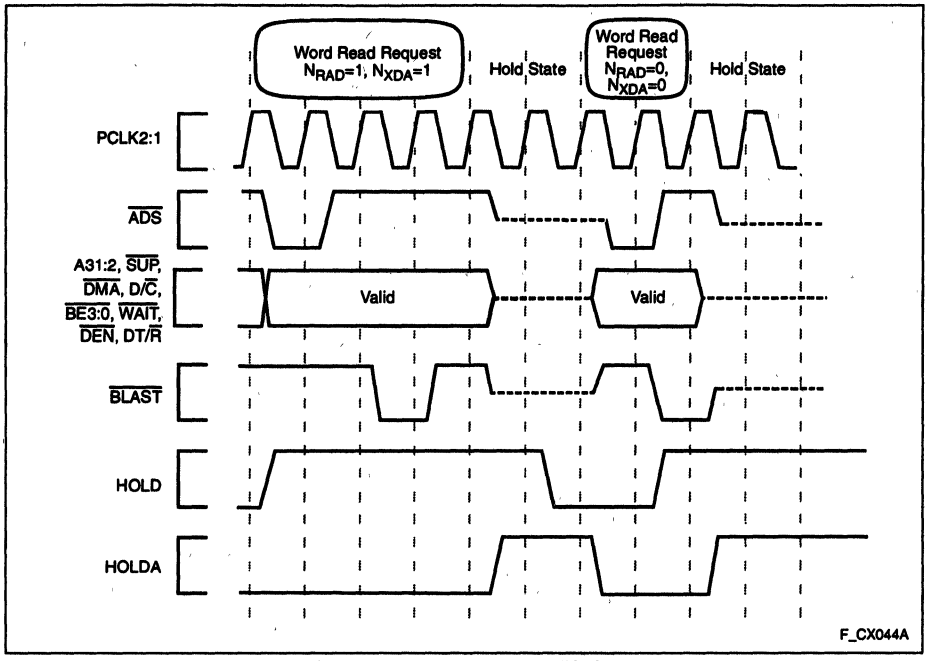
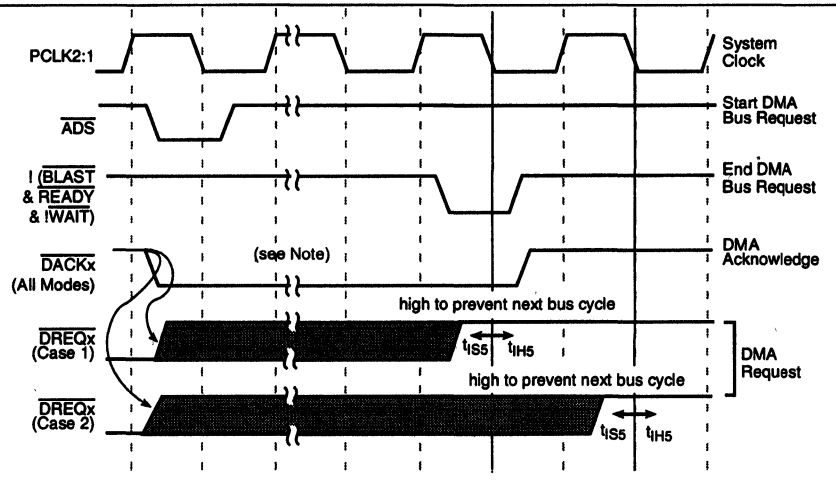


Figure 41. HOLD Functional Timing

272493-54

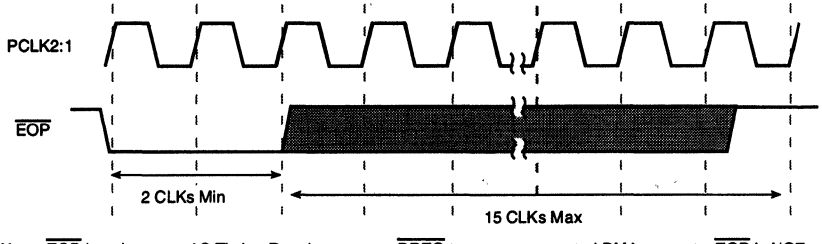


Note:

1. Case 1: $\overline{DREQ}$ must deassert before $\overline{DACK}$ deasserts. This applies to all Fly-By modes: source synchronized packing modes and destination synchronized unpacking modes.
2. Case 2: $\overline{DREQ}$ must be deasserted by the second clock (rising edge) after $\overline{DACK}$ is driven high. This applies to all other DMA transfers.
3. $\overline{DACK}$x is asserted for the duration of a DMA bus request. The request may consist of multiple bus accesses (defined by $\overline{ADS}$ and $\overline{BLAST}$).

F_CX018A

Figure 42. $\overline{DREQ}$ and $\overline{DACK}$ Functional Timing



Note: $\overline{EOP}$ has the same AC Timing Requirements as $\overline{DREQ}$ to prevent unwanted DMA requests. $\overline{EOP}$ is NOT edge triggered. $\overline{EOP}$ must be held for a minimum of 2 clock cycles then deasserted within 15 clock cycles.

F_CX045A

Figure 43. $\overline{EOP}$ Functional Timing

272493-55

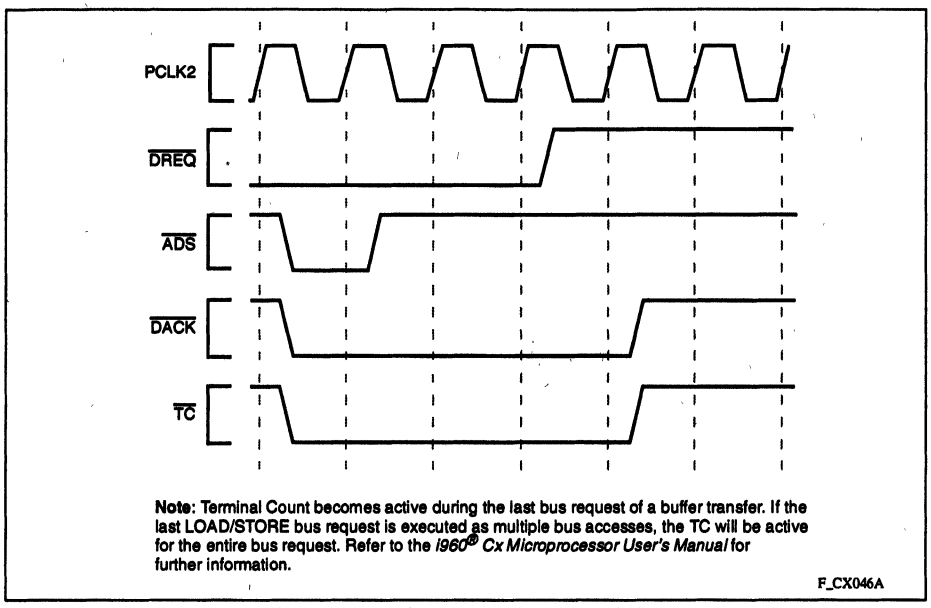


Figure 44. Terminal Count Functional Timing

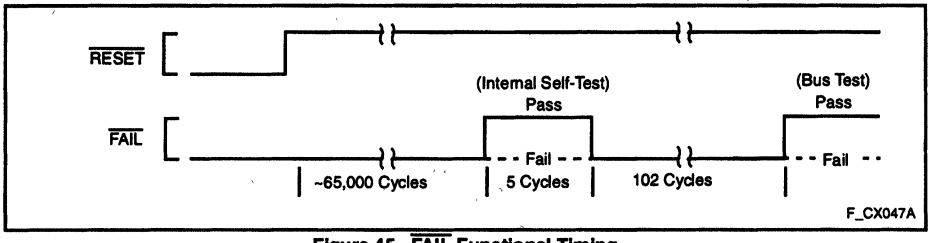


Figure 45. FAIL Functional Timing

272493-56

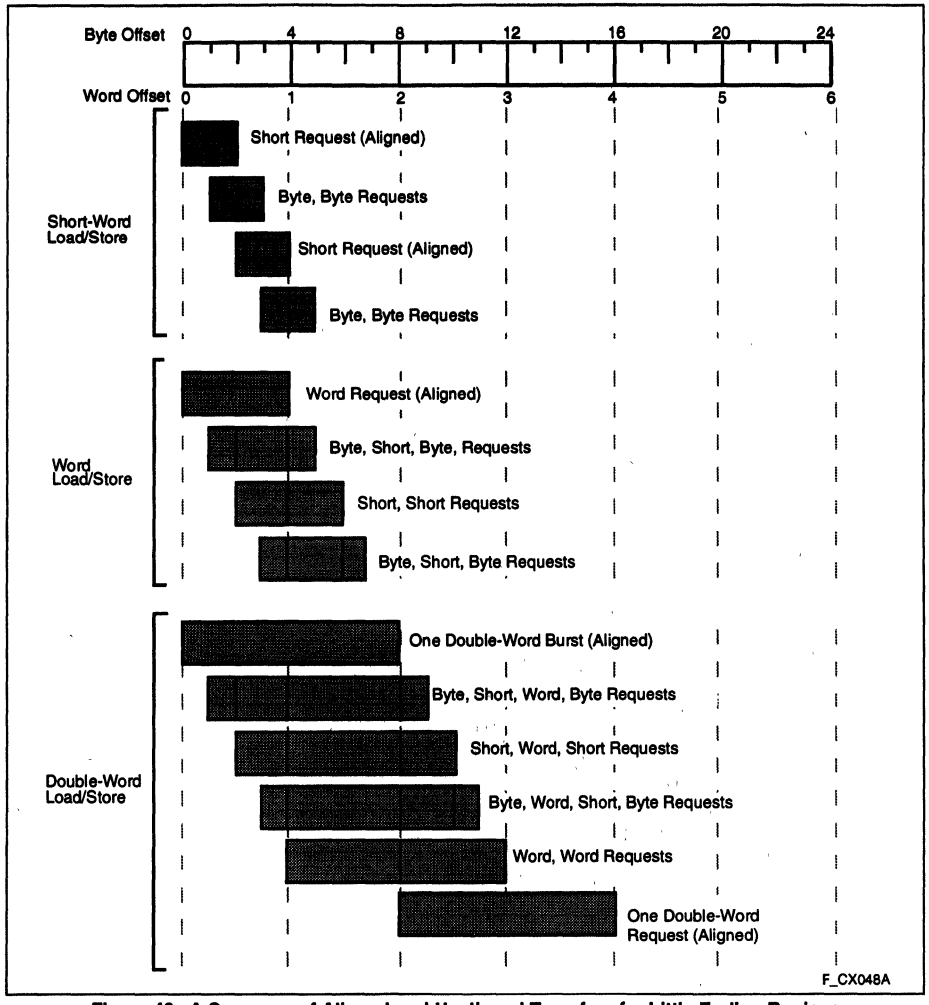


Figure 46. A Summary of Aligned and Unaligned Transfers for Little Endian Regions

272493-57

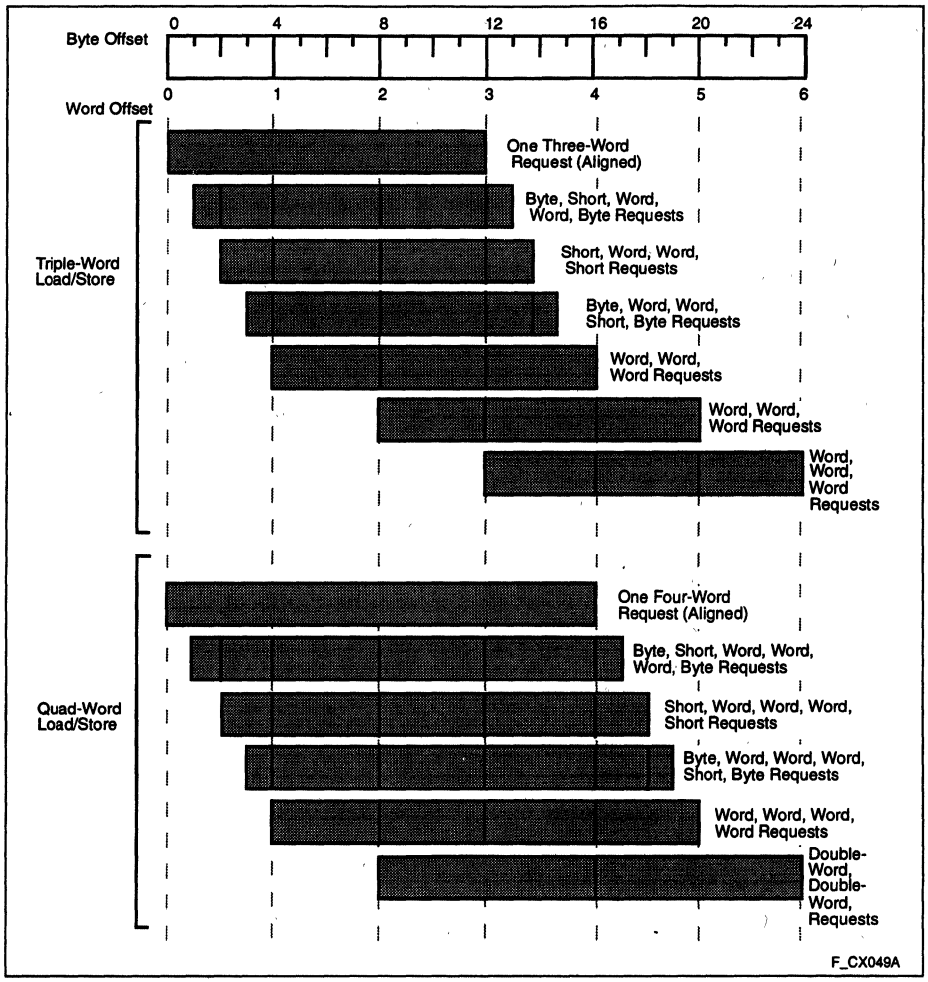


Figure 47. A Summary of Aligned and Unaligned Transfers for Little Endian Regions (Continued)

272493-58

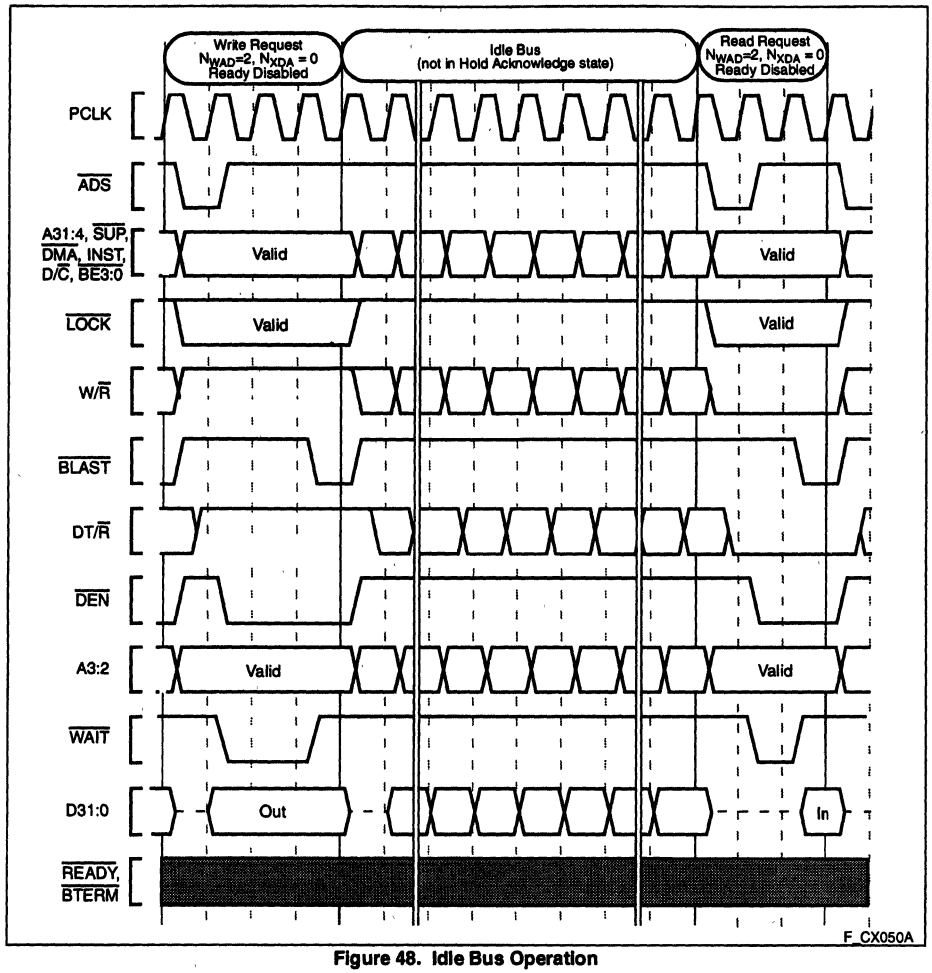


Figure 48. Idle Bus Operation

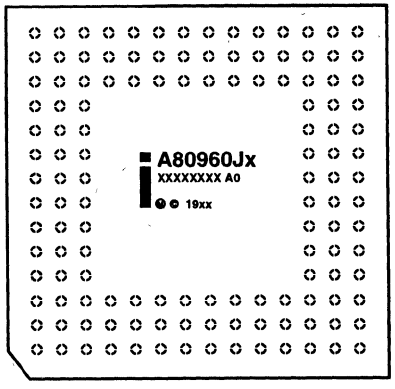
7.0 REVISION HISTORY

This is a new data sheet for the 80960CF-40 product. It is derived from the 80960CF-33, -25, -16 data sheet. Aside from a few minor edits, only the AC Characteristics differ from the 80960CF-33, -25, -16 data sheet.

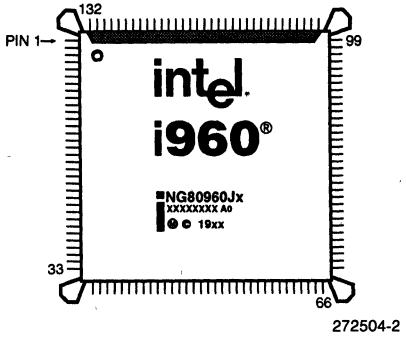
272493-59

80960JA/JF
EMBEDDED 32-BIT MICROPROCESSOR

- High-Performance Embedded Architecture
  - One Instruction/Clock Execution
  - Load/Store Programming Model
  - Sixteen 32-Bit Global Registers
  - Sixteen 32-Bit Local Registers (8 sets)
  - Nine Addressing Modes
  - User/Supervisor Protection Model
- Two-Way Set Associative Instruction Cache
  - 80960JA—2 KByte
  - 80960JF—4 KByte
  - Programmable Cache Locking Mechanism
- Direct Mapped Data Cache
  - 80960JA—1 KByte
  - 80960JF—2 KByte
  - Write Through Operation
- On-Chip Local Register Cache
  - Eight Stack Frames Available
  - Automatic Allocation on Call/Return
  - 0-8 Frames Reserved for Interrupts
- On-Chip Data RAM
  - 1 KByte Critical Variable Storage
  - Single-Cycle Access
- High Bandwidth Burst Bus
  - 32-Bit Multiplexed Address/Data
  - Programmable Memory Configuration
  - Selectable 8-, 16-, 32-Bit Bus Widths
  - Supports Unaligned Accesses
  - Big or Little Endian Byte Ordering
- High-Speed Interrupt Controller
  - 31 Programmable Priorities
  - Eight Maskable Pins plus $\overline{\text{NMI}}$
  - Up to 240 Vectors in Expanded Mode
- Two On-Chip Timers
  - Independent 32-Bit Counting
  - Clock Prescaling by 1, 2, 4 or 8
  - Internal Interrupt Sources
- HALT Mode for Low Power
- IEEE 1149.1 (JTAG) Boundary Scan Compatibility
- Packages
  - 132-Lead Pin Grid Array (PGA)
  - 132-Lead Plastic Quad Flat-Pack (PQFP)



272504-1



272504-2

Figure 1. 80960JA/JF Microprocessors

November 1994
Order Number: 272504-002

80960JA/JF
EMBEDDED 32-BIT MICROPROCESSOR

CONTENTS

CONTENTS

PAGE

FIGURES

CONTENTS

PAGE

TABLES

1.0 PURPOSE

This document provides advance information for the 80960JA/JF microprocessor, including targeted electrical characteristics and package pinout information. Detailed functional descriptions—other than parametric performance—will be published in the *i960® Jx Microprocessor User's Guide* (272483).

2.0 80960JA/JF OVERVIEW

The 80960JA/JF, a new member in the family of embedded i960® processors, provides a new set of essential enhancements for an emerging class of high-performance embedded applications. The 80960JA/JF is object code compatible with the 32-bit 80960 Core Architecture and is capable of sustained execution at the rate of one instruction per clock. This processor's features include a larger instruction cache, data cache and increased data RAM. It also boasts a fast Interrupt mechanism, dual programmable timer units and new instructions.

Memory subsystems for cost-sensitive embedded applications often impose substantial wait state penalties. The 80960JA/JF integrates generous storage resources on-chip to decouple CPU execution from the external bus.

The 80960JF includes a 4 Kbyte instruction cache and a 2 Kbyte data cache. For greater economy, the 80960JA includes a 2 Kbyte instruction cache and a 1 Kbyte data cache. Both processors include a 1 Kbyte data RAM.

An eight-set stack frame cache allows the processor to rapidly allocate and deallocate local registers during context switches.

A 32-bit multiplexed burst bus provides a high-speed interface to system memory and I/O. A full complement of control signals simplifies the connection of the 80960JA/JF to external components. The user programs physical and logical memory attributes through memory-mapped control registers (MMRs)—an extension not found on the i960 Kx, Sx or Cx processors. Physical and logical configuration registers enable the processor to operate with all combinations of bus width and data object alignment. The processor supports a homogeneous byte ordering model.

This processor integrates two important peripherals: a timer unit and an interrupt controller. These and other hardware resources are programmed through memory-mapped control registers, a user-friendly architectural extension.

The timer unit (TU) contains two independent 32-bit timers which are capable of counting at several clock rates and generating interrupts. Each is programmed by use of the Timer Unit registers. These registers are memory-mapped within the 80960JA/JF, addressable on 32-bit boundaries. The timers have a single-shot mode and auto-reload capabilities for continuous operation. Each timer has an independent interrupt request to the 80960JA/JF's interrupt controller. The TU can generate a fault when unauthorized writes from user mode are detected.

The interrupt controller unit (ICU) provides a flexible, low-latency means for requesting interrupts. It provides full programmability of up to 240 interrupt sources into 31 priority levels. The ICU takes advantage of a cached priority table, dedicated local registers and optional routine caching to minimize interrupt latency. Acting independently from the core, the ICU compares the priorities of posted interrupts with the current process priority, off-loading this task from the core. The ICU also supports the integrated timer interrupts.

The 80960JA/JF features a Halt mode designed to support applications where low power consumption is critical. The **halt** instruction lets you shut down the processor, resulting in a power savings of approximately 90 percent.

The 80960JA/JF's testability features, including ONCE (On-Circuit Emulation) and Boundary Scan (JTAG), create a powerful environment for design debug and fault diagnosis.

The *Solutions960®* program features a wide variety of development tools that support the i960 processor family. Many of these tools are developed by partner companies; some are developed by Intel, such as profile-driven optimizing compilers. For more information on these products, contact your local Intel representative.

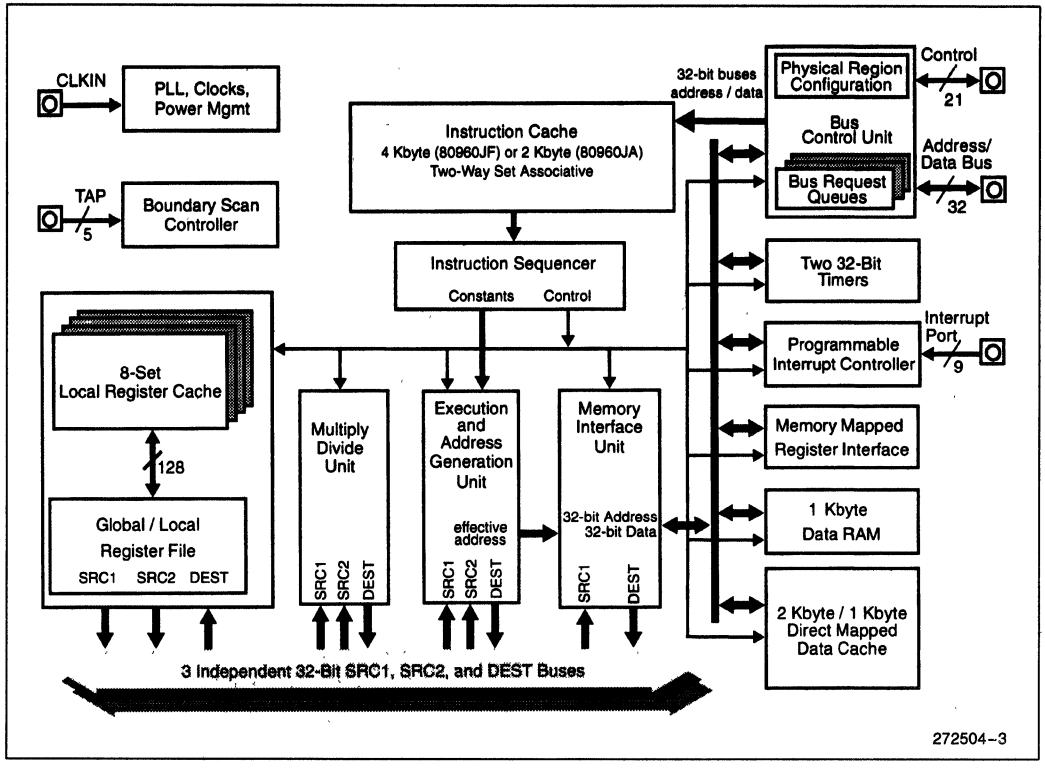


Figure 2. 80960JA/JF Block Diagram

2.1 80960 Processor Core

The 80960JA/JF microprocessor is a new, scalar implementation of the 80960 Core Architecture. Intel designed it to be a very high performance device that is also cost-effective. Factors that contribute to the core's performance include:

- Single-clock execution of most instructions
- Independent Multiply/Divide Unit
- Efficient instruction pipeline minimizes pipeline break latency
- Register and resource scoreboarding allow overlapped instruction execution
- 128-bit register bus speeds local register caching
- Two-way set associative, integrated instruction cache (80960JF is 4 Kbyte; 80960JA is 2 Kbyte)
- Direct-mapped, integrated data cache (80960JF is 2 Kbyte; 80960JA is 1 Kbyte)
- 1 Kbyte integrated data RAM delivers zero wait state program data

2.2 Burst Bus

A 32-bit high-performance bus controller interfaces the 80960JA/JF to external memory and peripherals. The Bus Control Unit fetches instructions and transfers data at the rate of up to four 32-bit words per six clock cycles. The external address/data bus is multiplexed.

Users may configure the memory space for physical and logical characteristics to match an application's requirements. Physical bus width is register programmed for up to eight regions. Byte ordering and data caching are programmed through a group of logical memory templates and a defaults register.

The Bus Control Unit's features include:

- Multiplexed external bus to minimize pin count
- 32-, 16- and 8-bit bus widths to simplify I/O interfaces
- External ready control for address-to-data, data-to-data and data-to-next-address (recovery) wait states
- Support for Big or Little Endian byte ordering to facilitate the porting of existing program code
- Unaligned bus accesses performed transparently
- Three-deep load/store queue to decouple the bus from the core

Upon reset, the 80960JA/JF conducts an internal self test. Then, before executing its first instruction, it performs an external bus confidence test by performing a checksum on the first words of the Initialization Boot Record.

While the processor is running, the user may examine the contents of the caches by issuing special cache control instructions.

2.3 Timer Unit

The timer unit offers two independent 32-bit timers for use as real-time system clocks and general purpose system timing. These operate in either single-shot or auto-reload mode and can generate interrupts to the core. Clock prescaling is supported.

2.4 Priority Interrupt Controller

A programmable interrupt controller manages up to 240 external sources through an 8-bit external interrupt port. Alternatively, the interrupt inputs may be configured for individual edge- or level-triggered inputs. The Interrupt Unit also accepts interrupts from the two on-chip timer channels and a single Non-Maskable Interrupt ($\overline{NMI}$) pin. Interrupts are serviced according to their priority levels relative to the current process priority.

Low interrupt latency is critical to many embedded applications. As part of its highly flexible interrupt mechanism, the 80960JA/JF exploits several techniques to minimize latency:

- Interrupt vectors and interrupt handler routines can be cached on-chip
- Register frames for high-priority interrupt handlers can be cached on-chip
- The interrupt stack can be placed in cacheable memory space

2.5 Instruction Set Summary

The 80960JA/JF adds several new instructions to the i960 core architecture. Table 1 shows all instructions that are available. The new instructions are:

- Conditional Move
- Conditional Add
- Conditional Subtract
- Byte Swap
- Halt
- Cache Control
- Interrupt Control

2.6 Faults and Debugging

The 80960JA/JF employs a comprehensive fault model. The processor responds to faults by making implicit calls to a fault handling routine. Specific information collected for each fault allows the fault handler to diagnose exceptions and recover appropriately.

The processor also has built-in debug capabilities. In software, the 80960JA/JF may be configured to detect as many as seven different trace event types. Alternatively, **mark** and **fmark** instructions can generate trace events explicitly in the instruction stream. Hardware breakpoint registers are also available to trap on execution and data addresses.

2.7 Low Power Operation

Intel fabricates the 80960JA/JF using an advanced sub-micron manufacturing process. The processor's sub-micron topology provides the circuit density for

optimal cache size and high operating speeds while dissipating modest power. The processor also uses dynamic power management to turn off clocks to unused circuits.

Users may program the 80960JA/JF to Halt Mode for maximum power savings. In Halt Mode, the processor core stops completely but the integrated peripherals continue to function, reducing overall power requirements by approximately 90 percent.

Processor execution resumes from internally or externally generated interrupts.

2.8 Test Features

The 80960JA/JF incorporates numerous features which enhance the user's ability to test both the processor and the system to which it is attached. These features include ONCE (On-Circuit Emulation) mode and Boundary Scan (JTAG).

The 80960JA/JF provides testability features compatible with IEEE Standard Test Access Port and Boundary Scan Architecture (IEEE Std. 1149.1).

One of the boundary scan instructions, HIGHZ, forces the processor to float all its output pins (ONCE mode). ONCE mode can also be initiated at reset without using the boundary scan mechanism.

ONCE is useful for board-level testing. This feature allows a mounted 80960JA/JF to electrically "remove" itself from a circuit board. This allows for system-level testing where a remote tester—such as an in-circuit emulator (ICE system)—can exercise the processor system.

The provided test logic does not interfere with component or circuit board behavior and ensures that components function correctly, connections between various components are correct, and various components interact correctly on the printed circuit board.

The JTAG Boundary Scan feature is an attractive alternative to conventional "bed-of-nails" testing. It can examine connections which might otherwise be inaccessible to a test system.

2.9 Memory-Mapped Control Registers

The 80960JA/JF, though compliant with i960 series processor core, has the added advantage of memory-mapped, internal control registers not found on the i960 Kx, Sx or Cx processors. These give software the interface to easily read and modify internal control registers.

Each of these registers is accessed as a memory-mapped, 32-bit register. Access is accomplished through regular memory-format instructions. The processor ensures that these accesses do not generate external bus cycles.

2.10 Data Types and Memory Addressing Modes

The 80960JA/JF instruction set supports several different data types and formats:

- Bit
- Bit fields
- Integer (8-, 16-, 32-, 64-bit)
- Ordinal (8-, 16-, 32-, 64-bit unsigned integers)
- Triple word (96 bits)
- Quad word (128 bits)

The 80960JA/JF provides a full set of addressing modes for C and assembly:

- Two Absolute modes
- Five Register Indirect modes
- Index with displacement
- IP with displacement

Table 1. 80960JA/JF Instruction Set

Data Movement	Arithmetic	Logical	Bit, Bit Field and Byte
Load Store Move *Conditional Select Load Address	Add Subtract Multiply Divide Remainder Modulo Shift Extended Shift Extended Multiply Extended Divide Add with Carry Subtract with Carry *Conditional Add *Conditional Subtract Rotate	And Not And And Not Or Exclusive Or Not Or Or Not Nor Exclusive Nor Not Nand	Set Bit Clear Bit Not Bit Alter Bit Scan For Bit Span Over Bit Extract Modify Scan Byte for Equal *Byte Swap
Comparison	**Branch**	**Call/Return**	**Fault**
Compare Conditional Compare Compare and Increment Compare and Decrement Test Condition Code Check Bit	Unconditional Branch Conditional Branch Compare and Branch	Call Call Extended Call System Return Branch and Link	Conditional Fault Synchronize Faults
Debug	**Processor Management**	**Atomic**	
Modify Trace Controls Mark Force Mark	Flush Local Registers Modify Arithmetic Controls Modify Process Controls *Halt System Control *Cache Control *Interrupt Control	Atomic Add Atomic Modify	

* Denotes new instructions unavailable on 80960CA/CF, 80960KA/KB and 80960SA/SB implementations.

3.0 PACKAGE INFORMATION

The 80960JA/JF will be offered in several speed and package grades. The following 132-pin Pin Grid Array (PGA) devices will be specified for operation at $V_{CC} = 5.0V \pm 5\%$ over a case temperature range of 0°C to 100°C:

- A80960JA/JF-33 (33 MHz)
- A80960JA/JF-25 (25 MHz)
- A80960JA/JF-16 (16 MHz)

The following 132-lead Plastic Quad Flatpack (PQFP) devices will be specified for operation at $V_{CC} = 5.0V \pm 5\%$ over a case temperature range of 0°C to 100°C:

- NG80960JA/JF-33 (33 MHz)
- NG80960JA/JF-25 (25 MHz)
- NG80960JA/JF-16 (16 MHz)

For complete package specifications and information, refer to Intel's *Packaging* Handbook (Order No. 240800).

3.1 Pin Descriptions

This section describes the pins for the 80960JA/JF in the 132-pin ceramic Pin Grid Array (PGA) package and 132-lead Plastic Quad Flatpack Package (PQFP).

Section 3.1.1, Functional Pin Definitions describes pin function; **Section 3.1.2, 80960JA/JF 132-Lead PGA Pinout** and **Section 3.1.3, 80960JA/JF PQFP Pinout** define the signal and pin locations for the supported package types.

3.1.1 FUNCTIONAL PIN DEFINITIONS

Table 2 presents the legend for interpreting the pin descriptions which follow. Pins associated with the bus interface are described in Table 3. Pins associated with basic control and test functions are described in Table 4. Pins associated with the Interrupt Unit are described in Table 5.

Table 2. Pin Description Nomenclature

Symbol	Description
I	Input pin only.
O	Output pin only.
I/O	Pin can be either an input or output.
—	Pin must be connected as described.
S	Synchronous. Inputs must meet setup and hold times relative to CLKIN for proper operation. S(E) Edge sensitive input S(L) Level sensitive input
A(...)	Asynchronous. Inputs may be asynchronous relative to CLKIN. A(E) Edge sensitive input A(L) Level sensitive input
R(...)	While the processor's RESET pin is asserted, the pin: R(1) is driven to V_{CC} R(0) is driven to V_{SS} R(Q) is a valid output R(X) is driven to unknown state R(H) is pulled up to V_{CC}
H(...)	While the processor is in the hold state, the pin: H(1) is driven to V_{CC} H(0) is driven to V_{SS} H(Q) Maintains previous state or continues to be a valid output H(Z) Floats
P(...)	While the processor is halted, the pin: P(1) is driven to V_{CC} P(0) is driven to V_{SS} P(Q) Maintains previous state or continues to be a valid output

1

Table 3. Pin Description—External Bus Signals (Sheet 1 of 3)

NAME	TYPE	DESCRIPTION
AD31:0	**I/O** S(L) R(X) H(Z) P(Q)	**ADDRESS/DATA BUS** carries 32-bit physical addresses and 8-, 16- or 32-bit data to and from memory. During an address (T_a) cycle, bits 2–31 contain a physical word address (bits 0–1 indicate SIZE; see below). During a data (T_d) cycle, read or write data is present on one or more contiguous bytes, comprising AD31:24, AD23:16, AD15:7 and AD7:0. During write operations, unused pins are driven to determinate values. SIZE, which comprises bits 0–1 of the AD lines during a T_a cycle, specifies the number of data transfers during the bus transaction. AD1 AD0 Bus Transfers 0 0 1 Transfer 0 1 2 Transfers 1 0 3 Transfers 1 1 4 Transfers When the processor enters HALT mode, if the previous bus operation was a: • write—AD31:2 are driven with the last data value on the AD bus. • read—AD31:4 are driven with the last address value on the AD bus; AD3:2 are driven with the value of A3:2 from the last data cycle. Typically, AD1:0 reflect the SIZE information of the last bus transaction (either instruction fetch or load/store) that was executed before entering HALT mode.
ALE	**O** R(0) H(Z) P(0)	**ADDRESS LATCH ENABLE** indicates the transfer of a physical address. ALE is asserted during a T_a cycle and deasserted before the beginning of the T_d state. It is active HIGH and floats to a high impedance state during a hold cycle (T_h).
$\overline{\text{ALE}}$	**O** R(1) H(Z) P(1)	**ADDRESS LATCH ENABLE** indicates the transfer of a physical address. $\overline{\text{ALE}}$ is the inverted version of ALE. This signal gives the 80960JA/JF a high degree of compatibility with existing 80960Kx systems.
$\overline{\text{ADS}}$	**O** R(1) H(Z) P(1)	**ADDRESS STROBE** indicates a valid address and the start of a new bus access. The processor asserts $\overline{\text{ADS}}$ for the entire T_a cycle. External bus control logic typically samples $\overline{\text{ADS}}$ at the end of the cycle.
A3:2	**O** R(X) H(Z) P(Q)	**ADDRESS3:2** comprise a partial demultiplexed address bus. 32-bit memory accesses: the processor asserts address bits A3:2 during T_a. The partial word address increments with each assertion of $\overline{\text{RDYRCV}}$ during a burst. 16-bit memory accesses: the processor asserts address bits A3:1 during T_a with A1 driven on the $\overline{\text{BE1}}$ pin. The partial short word address increments with each assertion of $\overline{\text{RDYRCV}}$ during a burst. 8-bit memory accesses: the processor asserts address bits A3:0 during T_a, with A1:0 driven on $\overline{\text{BE1:0}}$. The partial byte address increments with each assertion of $\overline{\text{RDYRCV}}$ during a burst.

Table 3. Pin Description—External Bus Signals (Sheet 2 of 3)

NAME	TYPE	DESCRIPTION
$\overline{BE3:0}$	O R(1) H(Z) P(1)	**BYTE ENABLES** select which of up to four data bytes on the bus participate in the current bus access. Byte enable encoding is dependent on the bus width of the memory region accessed: 32-bit bus: $\overline{BE3}$ enables data on AD31:24 $\overline{BE2}$ enables data on AD23:16 $\overline{BE1}$ enables data on AD15:8 $\overline{BE0}$ enables data on AD7:0 16-bit bus: $\overline{BE3}$ becomes Byte High Enable (enables data on AD15:8) $\overline{BE2}$ is not used (state is high) $\overline{BE1}$ becomes Address Bit 1 (A1) $\overline{BE0}$ becomes Byte Low Enable (enables data on AD7:0) 8-bit bus: $\overline{BE3}$ is not used (state is high) $\overline{BE2}$ is not used (state is high) $\overline{BE1}$ becomes Address Bit 1 (A1) $\overline{BE0}$ becomes Address Bit 0 (A0) The processor asserts byte enables, byte high enable and byte low enable during T_a. Since unaligned bus requests are split into separate bus transactions, these signals do not toggle during a burst. They remain active through the last T_d cycle. For accesses to 8- and 16-bit memory, the processor asserts the address bits in conjunction with A3:2 described above.
WIDTH/ HLTD1:0	O R(X) H(Z) P(1)	**WIDTH/HLTD** signals denote the physical memory attributes for a bus transaction: WIDTH/HLTD1 WIDTH/HLTD0 0 0 8 Bits Wide 0 1 16 Bits Wide 1 0 32 Bits Wide 1 1 Processor Halted The processor floats the WIDTH/HLTD pins whenever it relinquishes the bus in response to a HOLD request, regardless of prior operating state.
D/$\overline{C}$	O R(X) H(Z) P(Q)	**DATA/CODE** indicates that a bus access is a data access (1) or an instruction access (0). D/$\overline{C}$ has the same timing as W/$\overline{R}$.
W/$\overline{R}$	O R(0) H(Z) P(Q)	**WRITE/READ** specifies, during a T_a cycle, whether the operation is a write (1) or read (0). It is latched on-chip and remains valid during T_d cycles.
DT/$\overline{R}$	O R(0) H(Z) P(Q)	**DATA TRANSMIT/RECEIVE** indicates the direction of data transfer to and from the address/data bus. It is low during T_a and T_w/T_d cycles for a read; it is high during T_a and T_w/T_d cycles for a write. DT/$\overline{R}$ never changes state when $\overline{DEN}$ is asserted.

Table 3. Pin Description—External Bus Signals (Sheet 3 of 3)

NAME	TYPE	DESCRIPTION
$\overline{\text{DEN}}$	O R(1) H(Z) P(1)	**DATA ENABLE** indicates data transfer cycles during a bus access. $\overline{\text{DEN}}$ is asserted at the start of the first data cycle in a bus access and deasserted at the end of the last data cycle. $\overline{\text{DEN}}$ is used with DT/$\overline{\text{R}}$ to provide control for data transceivers connected to the data bus.
$\overline{\text{BLAST}}$	O R(1) H(Z) P(1)	**BURST LAST** indicates the last transfer in a bus access. $\overline{\text{BLAST}}$ is asserted in the last data transfer of burst and non-burst accesses. $\overline{\text{BLAST}}$ remains active as long as wait states are inserted via the $\overline{\text{RDYRCV}}$ pin. $\overline{\text{BLAST}}$ becomes inactive after the final data transfer in a bus cycle.
$\overline{\text{RDYRCV}}$	I S(L)	**READY/RECOVER** indicates that data on AD lines can be sampled or removed. If $\overline{\text{RDYRCV}}$ is not asserted during a T_d cycle, the T_d cycle is extended to the next cycle by inserting a wait state (T_w). The $\overline{\text{RDYRCV}}$ pin has an alternate function during the recovery (T_r) state. The processor continues to insert additional recovery states until it samples the pin HIGH. This function allows slow external devices longer to float their buffers before the processor begins to drive address again.
$\overline{\text{LOCK}}$/ $\overline{\text{ONCE}}$	I/O S(L) R(H) H(Z) P(1)	**BUS LOCK** indicates that an atomic read-modify-write operation is in progress. $\overline{\text{LOCK}}$ is asserted in the first clock of an atomic operation and deasserted in the last data transfer of the sequence. The processor does not grant HOLDA while $\overline{\text{LOCK}}$ is asserted. This prevents external agents from accessing memory involved in semaphore operations. The processor samples this pin during reset. If it is asserted LOW at the end of reset, the processor enters ONCE Mode. In ONCE Mode, the processor stops all clocks and floats all output pins. The pin has a weak internal pullup which is active during reset to ensure normal operation if the pin is left unconnected.
HOLD	I S(L)	**HOLD:** A request from an external bus master to acquire the bus. When the processor receives HOLD and grants bus control to another master, it asserts HOLDA, floats the address/data and control lines and enters the T_h state. When HOLD is deasserted, the processor deasserts HOLDA and enters either the T_i or T_a state, resuming control of the address/data and control lines.
HOLDA	O R(Q) H(1) P(Q)	**HOLD ACKNOWLEDGE** indicates to an external bus master that the processor has relinquished control of the bus. The processor can grant HOLD requests and enter the T_h state during reset and while halted as well as during regular operation.
BSTAT	O R(0) H(Q) P(0)	**BUS STATUS** indicates that the processor may soon stall unless it has sufficient access to the bus; see *i960® Jx Microprocessor User's Guide* (272483). Arbitration logic can examine this signal to determine when an external bus master should acquire/relinquish the bus.

Table 4. Pin Description—Processor Control Signals, Test Signals and Power (Sheet 1 of 2)

NAME	TYPE	DESCRIPTION
CLKIN	I	**CLOCK INPUT** provides the processor's fundamental time base; both the processor core and the external bus run at the CLKIN rate. All input and output timings are specified relative to a rising CLKIN edge.
$\overline{\text{RESET}}$	I A(L)	**RESET** initializes the processor and clears its internal logic. During reset, the processor places the address/data bus and control output pins in their idle (inactive) states. During reset, the input pins are ignored with the exception of $\overline{\text{LOCK}}$/$\overline{\text{ONCE}}$, STEST and HOLD. The $\overline{\text{RESET}}$ pin has an internal synchronizer. To ensure predictable processor initialization during power up, $\overline{\text{RESET}}$ must be asserted a minimum of 10,000 CLKIN cycles with V_{CC} and CLKIN stable. On a warm reset, $\overline{\text{RESET}}$ should be asserted for a minimum of 15 cycles.
STEST	I S(L)	**SELF TEST** enables or disables the processor's internal self-test feature at initialization. STEST is examined at the end of reset. If STEST is asserted, the processor performs its internal self-test and the external bus confidence test. If STEST is deasserted, the processor performs only the external bus confidence test.
$\overline{\text{FAIL}}$	O R(0) H(Q) P(1)	**FAIL** indicates a failure of the processor's built-in self-test performed during initialization. $\overline{\text{FAIL}}$ is asserted immediately upon reset and toggles during self-test to indicate the status of individual tests: • If self-test passes, the processor deasserts $\overline{\text{FAIL}}$ and commences operation from user code. • If self-test fails, the processor asserts the $\overline{\text{FAIL}}$ pin and then stops executing.
TCK	I	**TEST CLOCK** is a CPU input which provides the clocking function for IEEE 1149.1 Boundary Scan Testing (JTAG). State information and data are clocked into the component on the rising edge; data is clocked out of the component on the falling edge.
TDI	I S(L)	**TEST DATA INPUT** is the serial input pin for JTAG. TDI is sampled on the rising edge of TCK, during the SHIFT-IR and SHIFT-DR states of the Test Access Port.
TDO	O R(Q) H(Q) P(Q)	**TEST DATA OUTPUT** is the serial output pin for JTAG. TDO is driven on the falling edge of TCK during the SHIFT-IR and SHIFT-DR states of the Test Access Port. At other times, TDO floats. TDO does not float during ONCE Mode.
$\overline{\text{TRST}}$	I A(L)	**TEST RESET** asynchronously resets the Test Access Port (TAP) controller function of IEEE 1149.1 Boundary Scan testing (JTAG). If the Test Access Port will not be used, connect this pin to ground.

Table 4. Pin Description—Processor Control Signals, Test Signals and Power (Sheet 2 of 2)

NAME	TYPE	DESCRIPTION
TMS	I S(L)	**TEST MODE SELECT** is sampled at the rising edge of TCK to select the operation of the test logic for IEEE 1149.1 Boundary Scan testing.
V_{CC}	—	**POWER** leads intended for external connection to a V_{CC} board plane.
V_{CCPLL}	—	**PLL POWER** is a separate V_{CC} supply lead for the phase lock loop clock generator. It is intended for external connection to the V_{CC} board plane. In noisy environments, add a simple bypass filter circuit to reduce noise-induced clock jitter and its effects on timing relationships.
V_{SS}	—	**GROUND** leads intended for external connection to a V_{SS} board plane.
NC	—	**NO CONNECT** leads. Do not make any system connections to these leads.

Table 5. Pin Description—Interrupt Unit Signals

NAME	TYPE	DESCRIPTION
$\overline{XINT7:0}$	I A(E/L)	**EXTERNAL INTERRUPT** pins are used to request interrupt service. The $\overline{XINT7:0}$ pins can be configured in three modes: **Dedicated Mode:** Each pin is assigned a dedicated interrupt level. Dedicated inputs can be programmed to be level (low or high) or edge (rising or falling) sensitive. **Expanded Mode:** All eight pins act as a vectored interrupt source. The interrupt pins are level sensitive in this mode. **Mixed Mode:** The $\overline{XINT7:5}$ pins act as dedicated sources and the $\overline{XINT4:0}$ pins act as the five most significant bits of a vectored source. The least significant bits of the vectored source are set to 010 internally. Unused external interrupt pins should be connected to V_{CC}.
$\overline{NMI}$	I A(E)	**NON-MASKABLE INTERRUPT** causes a non-maskable interrupt event to occur. $\overline{NMI}$ is the highest priority interrupt source and is falling edge-triggered. If $\overline{NMI}$ is unused, it should be connected to V_{CC}.

3.1.2 80960JA/JF 132-LEAD PGA PINOUT

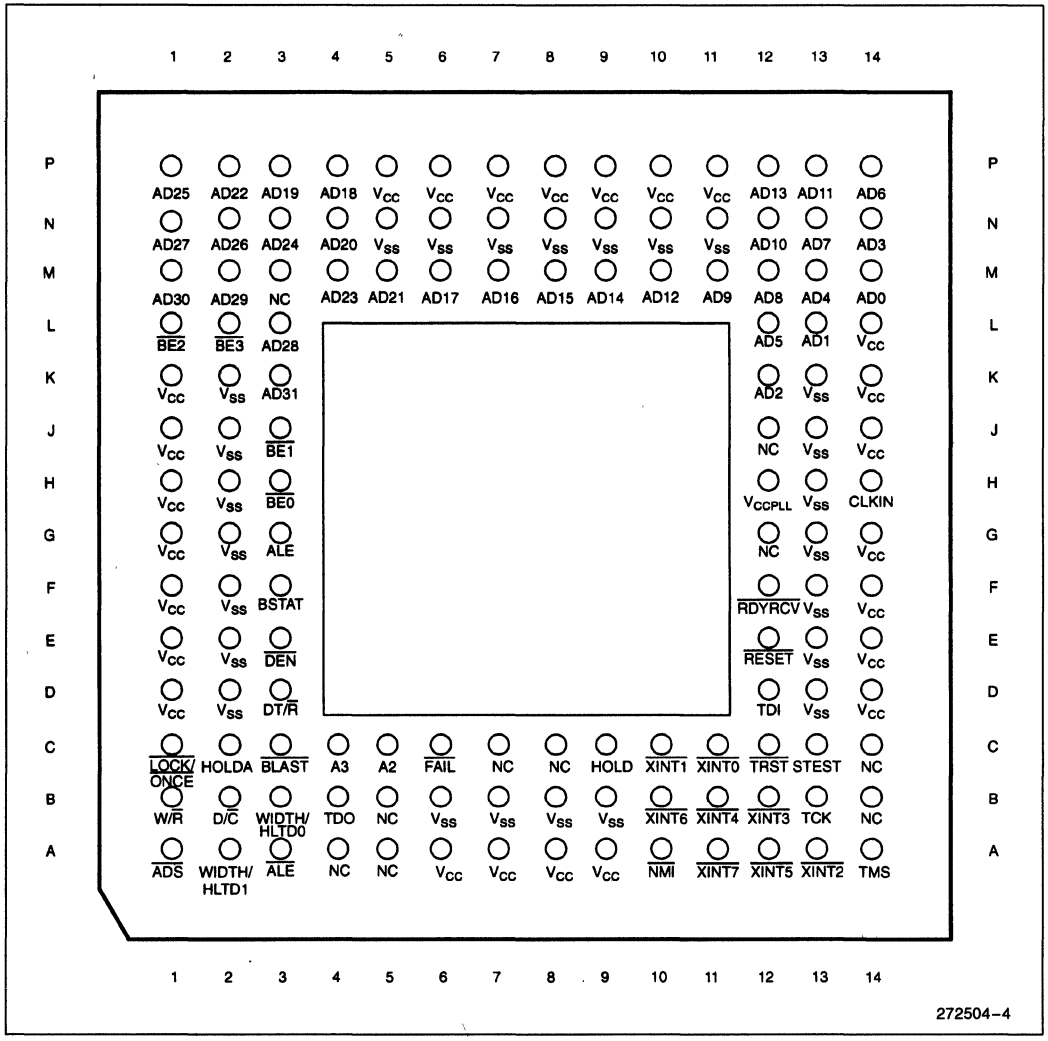


Figure 3. 132-Lead Pin Grid Array Bottom View—Pins Facing Up

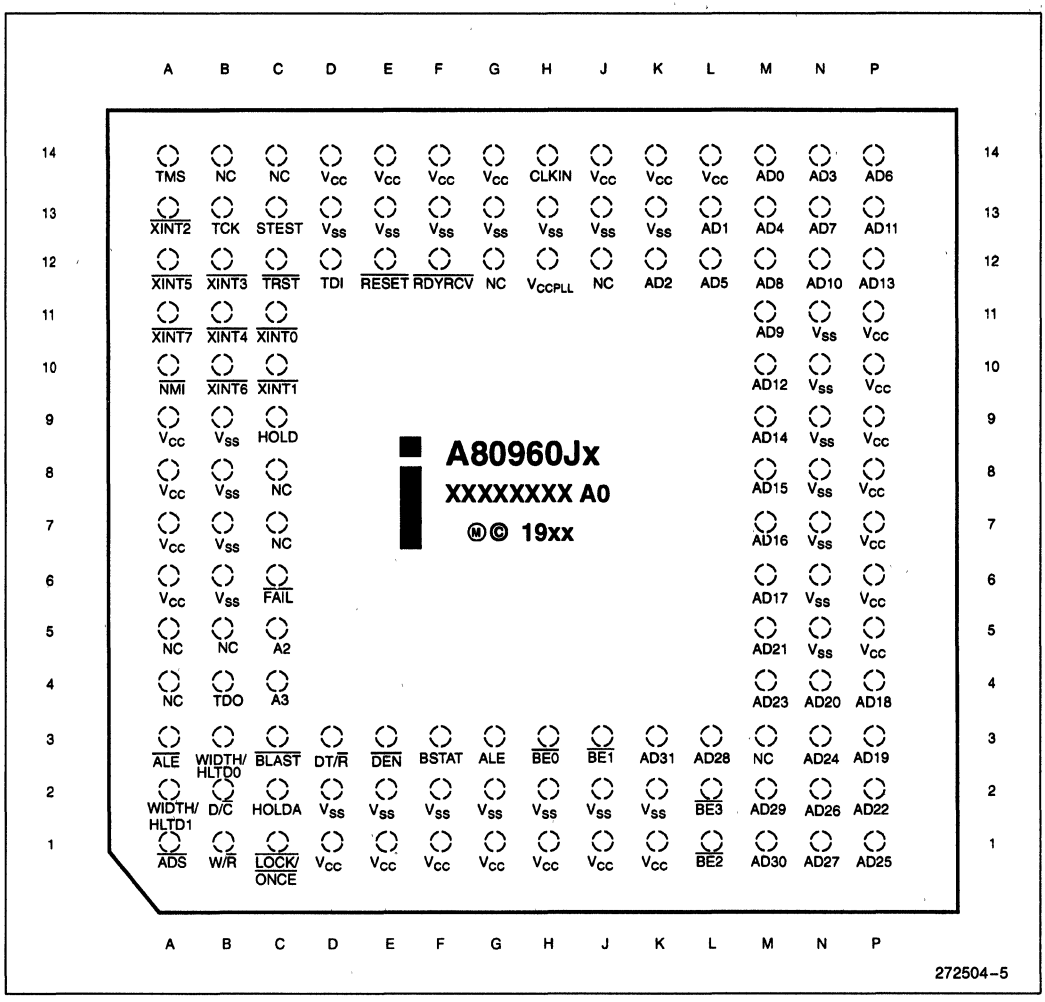


Figure 4. 132-Lead Pin Grid Array Top View—Pins Facing Down

Table 6. 132-Lead PGA Pinout—In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
A2	C5	AD31	K3	TDI	D12	V_{SS}	B9
A3	C4	$\overline{ADS}$	A1	TDO	B4	V_{SS}	D2
AD0	M14	ALE	G3	TMS	A14	V_{SS}	D13
AD1	L13	$\overline{ALE}$	A3	$\overline{TRST}$	C12	V_{SS}	E2
AD2	K12	$\overline{BE0}$	H3	V_{CC}	A6	V_{SS}	E13
AD3	N14	$\overline{BE1}$	J3	V_{CC}	A7	V_{SS}	F2
AD4	M13	$\overline{BE2}$	L1	V_{CC}	A8	V_{SS}	F13
AD5	L12	$\overline{BE3}$	L2	V_{CC}	A9	V_{SS}	G2
AD6	P14	$\overline{BLAST}$	C3	V_{CC}	D1	V_{SS}	G13
AD7	N13	BSTAT	F3	V_{CC}	D14	V_{SS}	H2
AD8	M12	CLKIN	H14	V_{CC}	E1	V_{SS}	H13
AD9	M11	D/$\overline{C}$	B2	V_{CC}	E14	V_{SS}	J2
AD10	N12	$\overline{DEN}$	E3	V_{CC}	F1	V_{SS}	J13
AD11	P13	DT/$\overline{R}$	D3	V_{CC}	F14	V_{SS}	K2
AD12	M10	$\overline{FAIL}$	C6	V_{CC}	G1	V_{SS}	K13
AD13	P12	HOLD	C9	V_{CC}	G14	V_{SS}	N5
AD14	M9	HOLDA	C2	V_{CC}	H1	V_{SS}	N6
AD15	M8	$\overline{LOCK/ONCE}$	C1	V_{CC}	J1	V_{SS}	N7
AD16	M7	NC	A4	V_{CC}	J14	V_{SS}	N8
AD17	M6	NC	A5	V_{CC}	K1	V_{SS}	N9
AD18	P4	NC	B5	V_{CC}	K14	V_{SS}	N10
AD19	P3	NC	B14	V_{CC}	L14	V_{SS}	N11
AD20	N4	NC	C7	V_{CC}	P5	W/$\overline{R}$	B1
AD21	M5	NC	C8	V_{CC}	P6	WIDTH/HLTD0	B3
AD22	P2	NC	C14	V_{CC}	P7	WIDTH/HLTD1	A2
AD23	M4	NC	G12	V_{CC}	P8	$\overline{XINT0}$	C11
AD24	N3	NC	J12	V_{CC}	P9	$\overline{XINT1}$	C10
AD25	P1	NC	M3	V_{CC}	P10	$\overline{XINT2}$	A13
AD26	N2	$\overline{NMI}$	A10	V_{CC}	P11	$\overline{XINT3}$	B12
AD27	N1	$\overline{RDYRCV}$	F12	V_{CCPLL}	H12	$\overline{XINT4}$	B11
AD28	L3	$\overline{RESET}$	E12	V_{SS}	B6	$\overline{XINT5}$	A12
AD29	M2	STEST	C13	V_{SS}	B7	$\overline{XINT6}$	B10
AD30	M1	TCK	B13	V_{SS}	B8	$\overline{XINT7}$	A11

NOTE:
Do not connect any external logic to pins marked NC (No Connect Pins).

Table 7. 132-Lead PGA Pinout—In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	$\overline{ADS}$	C6	$\overline{FAIL}$	H1	V_{CC}	M10	AD12
A2	WIDTH/HLTD1	C7	NC	H2	V_{SS}	M11	AD9
A3	$\overline{ALE}$	C8	NC	H3	$\overline{BE0}$	M12	AD8
A4	NC	C9	HOLD	H12	V_{CCPLL}	M13	AD4
A5	NC	C10	$\overline{XINT1}$	H13	V_{SS}	M14	AD0
A6	V_{CC}	C11	$\overline{XINT0}$	H14	CLKIN	N1	AD27
A7	V_{CC}	C12	$\overline{TRST}$	J1	V_{CC}	N2	AD26
A8	V_{CC}	C13	STEST	J2	V_{SS}	N3	AD24
A9	V_{CC}	C14	NC	J3	$\overline{BE1}$	N4	AD20
A10	$\overline{NMI}$	D1	V_{CC}	J12	NC	N5	V_{SS}
A11	$\overline{XINT7}$	D2	V_{SS}	J13	V_{SS}	N6	V_{SS}
A12	$\overline{XINT5}$	D3	DT/$\overline{R}$	J14	V_{CC}	N7	V_{SS}
A13	$\overline{XINT2}$	D12	TDI	K1	V_{CC}	N8	V_{SS}
A14	TMS	D13	V_{SS}	K2	V_{SS}	N9	V_{SS}
B1	W/$\overline{R}$	D14	V_{CC}	K3	AD31	N10	V_{SS}
B2	D/$\overline{C}$	E1	V_{CC}	K12	AD2	N11	V_{SS}
B3	WIDTH/HLTD0	E2	V_{SS}	K13	V_{SS}	N12	AD10
B4	TDO	E3	$\overline{DEN}$	K14	V_{CC}	N13	AD7
B5	NC	E12	$\overline{RESET}$	L1	$\overline{BE2}$	N14	AD3
B6	V_{SS}	E13	V_{SS}	L2	$\overline{BE3}$	P1	AD25
B7	V_{SS}	E14	V_{CC}	L3	AD28	P2	AD22
B8	V_{SS}	F1	V_{CC}	L12	AD5	P3	AD19
B9	V_{SS}	F2	V_{SS}	L13	AD1	P4	AD18
B10	$\overline{XINT6}$	F3	BSTAT	L14	V_{CC}	P5	V_{CC}
B11	$\overline{XINT4}$	F12	$\overline{RDYRCV}$	M1	AD30	P6	V_{CC}
B12	$\overline{XINT3}$	F13	V_{SS}	M2	AD29	P7	V_{CC}
B13	TCK	F14	V_{CC}	M3	NC	P8	V_{CC}
B14	NC	G1	V_{CC}	M4	AD23	P9	V_{CC}
C1	$\overline{LOCK/ONCE}$	G2	V_{SS}	M5	AD21	P10	V_{CC}
C2	HOLDA	G3	ALE	M6	AD17	P11	V_{CC}
C3	$\overline{BLAST}$	G12	NC	M7	AD16	P12	AD13
C4	A3	G13	V_{SS}	M8	AD15	P13	AD11
C5	A2	G14	V_{CC}	M9	AD14	P14	AD6

NOTE:
Do not connect any external logic to pins marked NC (no connect pins).

3.1.3 80960JA/JF PQFP PINOUT

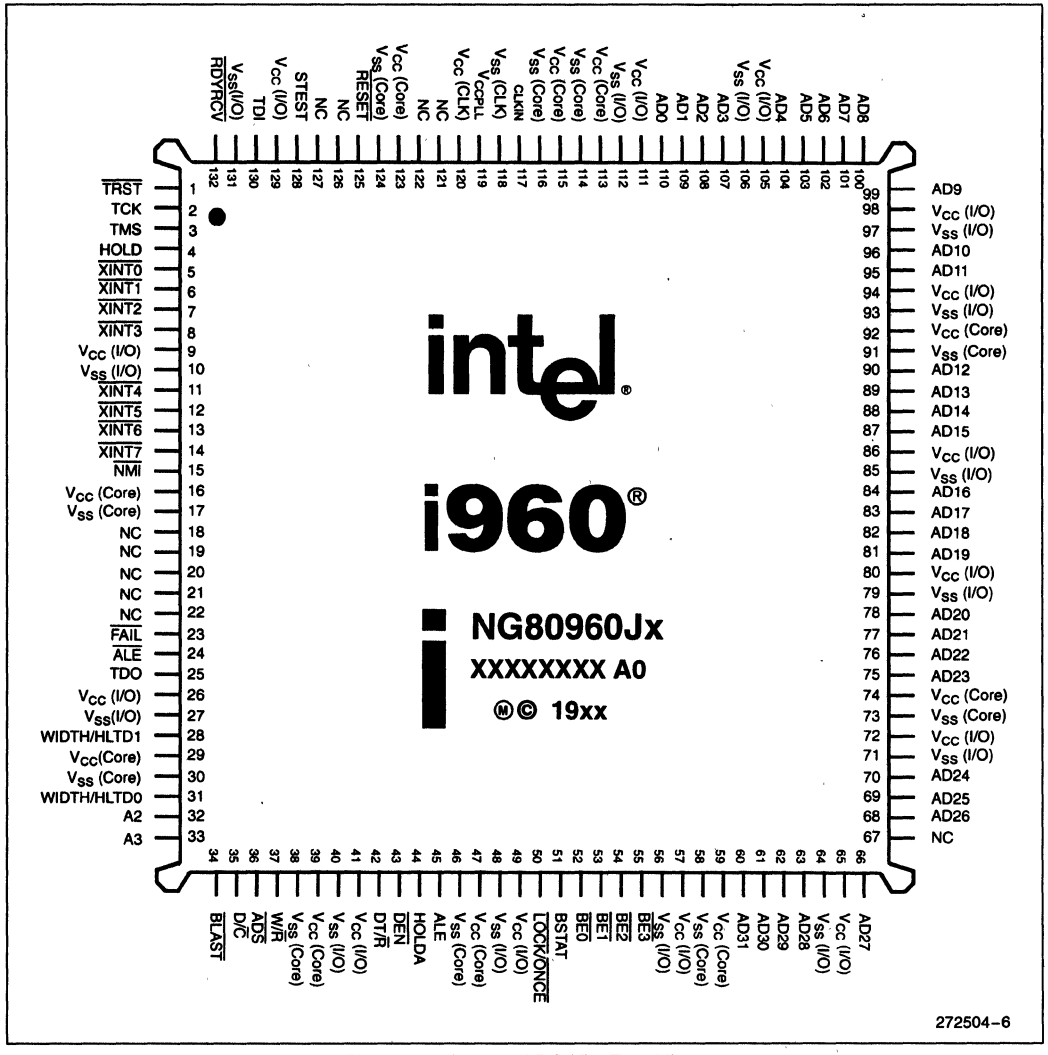


Figure 5. 132-Lead PQFP - Top View

Table 8. 132-Lead PQFP Pinout—In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
AD31	60	$\overline{ALE}$	24	V_{CC} (Core)	47	V_{SS} (I/O)	10
AD30	61	$\overline{ADS}$	36	V_{CC} (Core)	59	V_{SS} (I/O)	27
AD29	62	A3	33	V_{CC} (Core)	74	V_{SS} (I/O)	40
AD28	63	A2	32	V_{CC} (Core)	92	V_{SS} (I/O)	48
AD27	66	$\overline{BE3}$	55	V_{CC} (Core)	113	V_{SS} (I/O)	56
AD26	68	$\overline{BE2}$	54	V_{CC} (Core)	115	V_{SS} (I/O)	64
AD25	69	$\overline{BE1}$	53	V_{CC} (Core)	123	V_{SS} (I/O)	71
AD24	70	$\overline{BE0}$	52	V_{CC} (I/O)	9	V_{SS} (I/O)	79
AD23	75	WIDTH/HLTD1	28	V_{CC} (I/O)	26	V_{SS} (I/O)	85
AD22	76	WIDTH/HLTD0	31	V_{CC} (I/O)	41	V_{SS} (I/O)	93
AD21	77	D/$\overline{C}$	35	V_{CC} (I/O)	49	V_{SS} (I/O)	97
AD20	78	W/$\overline{R}$	37	V_{CC} (I/O)	57	V_{SS} (I/O)	106
AD19	81	DT/$\overline{R}$	42	V_{CC} (I/O)	65	V_{SS} (I/O)	112
AD18	82	$\overline{DEN}$	43	V_{CC} (I/O)	72	V_{SS} (I/O)	131
AD17	83	$\overline{BLAST}$	34	V_{CC} (I/O)	80	NC	18
AD16	84	$\overline{RDYRCV}$	132	V_{CC} (I/O)	86	NC	19
AD15	87	$\overline{LOCK/ONCE}$	50	V_{CC} (I/O)	94	NC	20
AD14	88	HOLD	4	V_{CC} (I/O)	98	NC	21
AD13	89	HOLDA	44	V_{CC} (I/O)	105	NC	22
AD12	90	BSTAT	51	V_{CC} (I/O)	111	NC	67
AD11	95	CLKIN	117	V_{CC} (I/O)	129	NC	121
AD10	96	$\overline{RESET}$	125	V_{CCPLL}	119	NC	122
AD9	99	STEST	128	V_{SS} (CLK)	118	NC	126
AD8	100	$\overline{FAIL}$	23	V_{SS} (Core)	17	NC	127
AD7	101	TCK	2	V_{SS} (Core)	30	$\overline{XINT7}$	14
AD6	102	TDI	130	V_{SS} (Core)	38	$\overline{XINT6}$	13
AD5	103	TDO	25	V_{SS} (Core)	46	$\overline{XINT5}$	12
AD4	104	$\overline{TRST}$	1	V_{SS} (Core)	58	$\overline{XINT4}$	11
AD3	107	TMS	3	V_{SS} (Core)	73	$\overline{XINT3}$	8
AD2	108	V_{CC} (CLK)	120	V_{SS} (Core)	91	$\overline{XINT2}$	7
AD1	109	V_{CC} (Core)	16	V_{SS} (Core)	114	$\overline{XINT1}$	6
AD0	110	V_{CC} (Core)	29	V_{SS} (Core)	116	$\overline{XINT0}$	5
ALE	45	V_{CC} (Core)	39	V_{SS} (Core)	124	$\overline{NMI}$	15

NOTE:
Do not connect any external logic to pins marked NC (no connect pins).

Table 9. 132-Lead PQFP Pinout—In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	$\overline{TRST}$	34	$\overline{BLAST}$	67	NC	100	AD8
2	TCK	35	D/$\overline{C}$	68	AD26	101	AD7
3	TMS	36	$\overline{ADS}$	69	AD25	102	AD6
4	HOLD	37	W/$\overline{R}$	70	AD24	103	AD5
5	$\overline{XINT0}$	38	V_{SS} (Core)	71	V_{SS} (I/O)	104	AD4
6	$\overline{XINT1}$	39	V_{CC} (Core)	72	V_{CC} (I/O)	105	V_{CC} (I/O)
7	$\overline{XINT2}$	40	V_{SS} (I/O)	73	V_{SS} (Core)	106	V_{SS} (I/O)
8	$\overline{XINT3}$	41	V_{CC} (I/O)	74	V_{CC} (Core)	107	AD3
9	V_{CC} (I/O)	42	DT/$\overline{R}$	75	AD23	108	AD2
10	V_{SS} (I/O)	43	$\overline{DEN}$	76	AD22	109	AD1
11	$\overline{XINT4}$	44	HOLDA	77	AD21	110	AD0
12	$\overline{XINT5}$	45	ALE	78	AD20	111	V_{CC} (I/O)
13	$\overline{XINT6}$	46	V_{SS} (Core)	79	V_{SS} (I/O)	112	V_{SS} (I/O)
14	$\overline{XINT7}$	47	V_{CC} (Core)	80	V_{CC} (I/O)	113	V_{CC} (Core)
15	$\overline{NMI}$	48	V_{SS} (I/O)	81	AD19	114	V_{SS} (Core)
16	V_{CC} (Core)	49	V_{CC} (I/O)	82	AD18	115	V_{CC} (Core)
17	V_{SS} (Core)	50	$\overline{LOCK/ONCE}$	83	AD17	116	V_{SS} (Core)
18	NC	51	BSTAT	84	AD16	117	CLKIN
19	NC	52	$\overline{BE0}$	85	V_{SS} (I/O)	118	$V_{SS(CLK)}$
20	NC	53	$\overline{BE1}$	86	V_{CC} (I/O)	119	V_{CCPLL}
21	NC	54	$\overline{BE2}$	87	AD15	120	$V_{CC(CLK)}$
22	NC	55	$\overline{BE3}$	88	AD14	121	NC
23	$\overline{FAIL}$	56	V_{SS} (I/O)	89	AD13	122	NC
24	$\overline{ALE}$	57	V_{CC} (I/O)	90	AD12	123	V_{CC} (Core)
25	TDO	58	V_{SS} (Core)	91	V_{SS} (Core)	124	V_{SS} (Core)
26	V_{CC} (I/O)	59	V_{CC} (Core)	92	V_{CC} (Core)	125	$\overline{RESET}$
27	V_{SS} (I/O)	60	AD31	93	V_{SS} (I/O)	126	NC
28	WIDTH/HLTD1	61	AD30	94	V_{CC} (I/O)	127	NC
29	V_{CC} (Core)	62	AD29	95	AD11	128	STEST
30	V_{SS} (Core)	63	AD28	96	AD10	129	V_{CC} (I/O)
31	WIDTH/HLTD0	64	V_{SS} (I/O)	97	V_{SS} (I/O)	130	TDI
32	A2	66	V_{CC} (I/O)	98	V_{CC} (I/O)	131	V_{SS} (I/O)
33	A3	65	AD27	99	AD9	132	$\overline{RDYRCV}$

NOTE:
Do not connect any external logic to pins marked NC (no connect pins).

3.2 Package Thermal Specifications

The 80960JA/JF is specified for operation when T_C (case temperature) is within the range of 0°C to 100°C. Case temperature may be measured in any environment to determine whether the 80960JA/JF is within specified operating range. The case temperature should be measured at the center of the top surface, opposite the pins.

T_A (ambient temperature) can be calculated from θ_{CA} (thermal resistance from case to ambient) using the following equation:

$$T_A = T_C - P^*\theta_{CA}$$

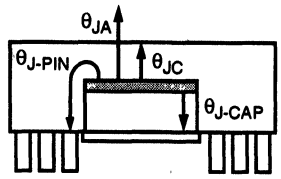
Similarly, T_J (junction temperature) can be calculated from θ_{JC} (thermal resistance from junction to case) using the following equation:

$$Tj = T_C + P^*\theta_{JC}$$

Compute P by multiplying I_{CC} (the typical current) from Table 13 and a V_{CC} of 5 V. Values for θ_{JC} and θ_{CA} are given in Table 10 for the PGA package and Table 11 for the PQFP package. Note that the processor's θ_{JA} for the ceramic PGA package may be significantly reduced by adding a heatsink.

Table 10. 132-Lead PGA Package Thermal Characteristics

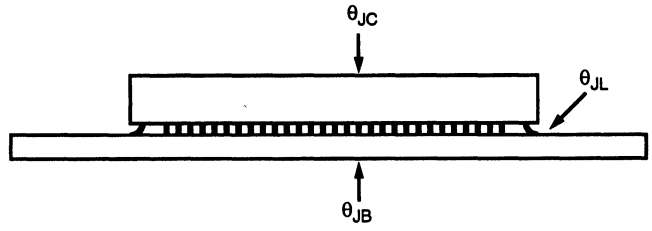
Thermal Resistance—°C/Watt						
Parameter	Airflow—ft./min (m/sec)					
	0 (0)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)	1000 (5.08)
θ_{JC} (Junction-to-Case)	3	3	3	3	3	3
θ_{CA} (Case-to-Ambient) (No Heatsink)	18	15	12	11	11	11
θ_{CA} (Case-to-Ambient) (Omnidirectional Heatsink)	15	12	9	8	8	8
θ_{CA} (Case-to-Ambient) (Unidirectional Heatsink)	14	11	8	7	7	7



NOTES:
1. This table applies to a PGA device plugged into a socket or soldered directly into a board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$
3. $\theta_{J\text{-}CAP}$ = 4°C/W (approx.)
4. $\theta_{J\text{-}PIN}$ = 4°C/W (inner pins) (approx.)
5. $\theta_{J\text{-}PIN}$ = 8°C/W (outer pins) (approx.)

Table 11. 132-Lead PQFP Package Thermal Characteristics

Thermal Resistance—°C/Watt							
	Airflow—ft./min (m/sec)						
Parameter	**0 (0)**	**50 (0.25)**	**100 (0.50)**	**200 (1.01)**	**400 (2.03)**	**600 (3.04)**	**800 (4.06)**
θ_{JC} (Junction-to-Case)	6	7	7	7	7	7	7
θ_{CA} (Case-to-Ambient—No Heatsink)	23	20	18	14	10	9	8



NOTES:
1. This table applies to a PQFP device soldered directly into board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$
3. θ_{JL} = 18°C/W (approx.)
4. θ_{JB} = 18°C/W (approx.)

4.0 ELECTRICAL SPECIFICATIONS

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

4.1 Absolute Maximum Ratings

Parameter	Maximum Rating
Storage Temperature	−65°C to +150°C
Case Temperature Under Bias	−65°C to +110°C
Supply Voltage wrt. V_{SS}	−0.5V to + 6.5V
Voltage on Other Pins wrt. V_{SS}	−0.5V to V_{CC} + 0.5V

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

4.2 Operating Conditions

Table 12. Targeted 80960JA/JF Operating Conditions

Symbol	Parameter	Min	Max	Units	Notes
V_{CC}	Supply Voltage				
	80960JA/JF-33	4.75	5.25	V	
	80960JA/JF-25	4.75	5.25		
	80960JA/JF-16	4.75	5.25		
f_{CLKIN}	Input Clock Frequency				
	80960JA/JF-33	16	33.33	MHz	
	80960JA/JF-25	16	25		
	80960JA/JF-16	16	16.67		
T_C	Operating Case Temperature				
	A80960JA/JF-33 (132 PGA)	0	100	°C	
	NG80960JA/JF-33 (132 PQFP)	0	100		
	A80960JA/JF-25 (132 PGA)	0	100		
	NG80960JA/JF-25 (132 PQFP)	0	100		
	A80960JA/JF-16 (132 PGA)	0	100		
	NG80960JA/JF-16 (132 PQFP)	0	100		

4.3 Connection Recommendations

For clean on-chip power distribution, V_{CC} and V_{SS} pins separately feed the device's functional units. Power and ground connections must be made to all 80960JA/JF power and ground pins. On the circuit board, every V_{CC} pin should connect to a power plane and every V_{SS} pin should connect to a ground plane. Place liberal decoupling capacitance near the 80960JA/JF, since the processor can cause transient power surges. **Pins identified as NC must not be connected in the system.**

4.4 DC Specifications

Table 13. Targeted 80960JA/JF DC Characteristics

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{IL}	Input Low Voltage	−0.3		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC} + 0.3	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 5 mA
V_{OH}	Output High Voltage	2.4 V_{CC} − 0.5			V	I_{OH} = −1 mA I_{OH} = −200 µA
I_{LI}	Input Leakage Current			± 5	µA	0 ≤ V_{IN} ≤ V_{CC}
I_{LO}	Output Leakage Current			± 5	µA	0.4 ≤ V_{OUT} ≤ V_{CC}
I_{CC}	Power Supply Current (80960JA/JF-33) I_{CC} Operating I_{CC} Halt Mode I_{CC} ONCE Mode	320	355 36 TBD	395 40 TBD	mA	(1,2,3,4) (3,4) (3,4)

NOTES:
1. Measured with device operating and outputs loaded to the test condition in Figure 6, AC Test Load (pg. 31).
2. I_{CC} Minimum is measured at minimum V_{CC} and maximum temperature. This parameter is characterized but not tested.
3. I_{CC} Typical is measured at nominal V_{CC} and T_C = 25 °C. This parameter is characterized but not tested.
4. I_{CC} Maximum is measured at maximum V_{CC} and minimum temperature. This parameter is fully tested.
5. Not tested.

Table 13. Targeted 80960JA/JF DC Characteristics (Continued)

Symbol	Parameter	Min	Typ	Max	Units	Notes
I_{CC}	Power Supply Current (80960JA/JF-25)					
	I_{CC} Operating	240	270	300	mA	(1,2,3,4)
	I_{CC} Halt Mode		27	30		(3,4)
	I_{CC} ONCE Mode		TBD	TBD		(3,4)
I_{CC}	Power Supply Current (80960JA/JF-16)					
	I_{CC} Operating	155	175	195	mA	(1,2,3,4)
	I_{CC} Halt Mode		18	20		(3,4)
	I_{CC} ONCE Mode		TBD	TBD		(3,4)
C_{IN}	Input Capacitance					
	PGA			12	pF	$f_{CLKIN} = f_{MIN}$(5)
	PQFP			10		
C_{OUT}	I/O or Output Capacitance					
	PGA			12	pF	$f_{CLKIN} = f_{MIN}$(5)
	PQFP			10		
C_{CLK}	CLKIN Capacitance					
	PGA			12	pF	$f_{CLKIN} = f_{MIN}$(5)
	PQFP			10		

NOTES:
1. Measured with device operating and outputs loaded to the test condition in Figure 6, AC Test Load (pg. 31).
2. I_{CC} Minimum is measured at minimum V_{CC} and maximum temperature. This parameter is characterized but not tested.
3. I_{CC} Typical is measured at nominal V_{CC} and T_C = 25 °C. This parameter is characterized but not tested.
4. I_{CC} Maximum is measured at maximum V_{CC} and minimum temperature. This parameter is fully tested.
5. Not tested.

4.5 AC Specifications

Targeted 80960JA/JF AC timings are based upon design simulation at 33 MHz. Revised information for all frequency grades will be published upon the completion of device characterization. Contact your local Intel representative before finalizing a design.

Table 14. Targeted 80960JA/JF Input Clock Timings

Symbol	Parameter	Min	Max	Units	Notes
T_F	CLKIN Frequency	TBD	33.33	MHz	
T_C	CLKIN Period	30	62.5	ns	
T_{CS}	CLKIN Period Stability		±0.1	%Δ	Adjacent Clocks(1)
T_{CH}	CLKIN High Time	12		ns	Measured at 1.5V(1)
T_{CL}	CLKIN Low Time	12		ns	Measured at 1.5V(1)
T_{CR}	CLKIN Rise Time		4	ns	0.8V to 2.0V(1)
T_{CF}	CLKIN Fall Time		4	ns	2.0V to 0.8V(1)

NOTES:
1. Not tested

Table 15. Targeted 80960JA/JF Synchronous Output Timings

Symbol	Parameter	Min	Max	Units	Notes
T_{OV1}	Output Valid Delay, Except ALE/$\overline{ALE}$ Inactive and DT/$\overline{R}$	2.5	15	ns	
T_{OV2}	Output Valid Delay, DT/$\overline{R}$	0.5 T_C +2.5	0.5 T_C +15	ns	
T_{OF}	Output Float Delay	3	13	ns	(1)

NOTES:
1. A float condition occurs when the output current becomes less than I_{LO}. Float delay is not tested.

Table 16. Targeted 80960JA/JF Synchronous Input Timings

Symbol	Parameter	Min	Max	Units	Notes
T_{IS1}	Input Setup to CLKIN—AD31:0	6		ns	(1)
T_{IH1}	Input Hold from CLKIN—AD31:0	2		ns	(1)
T_{IS2}	Input Setup to CLKIN—$\overline{RDYRCV}$ and HOLD	8		ns	(2)
T_{IH2}	Input Hold from CLKIN—$\overline{RDYRCV}$ and HOLD	2		ns	(2)
T_{IS3}	Input Setup to CLKIN—$\overline{RESET}$	6		ns	(3)
T_{IH3}	Input Hold from CLKIN—$\overline{RESET}$	2		ns	(3)
T_{IS4}	Input Setup to CLKIN—$\overline{NMI}$, $\overline{XINT7:0}$, $\overline{ONCE}$, STEST	7		ns	(1)
T_{IH4}	Input Hold from CLKIN—$\overline{NMI}$, $\overline{XINT7:0}$, $\overline{ONCE}$, STEST	3		ns	(1)

NOTES:
1. AD31:0, $\overline{ONCE}$ and STEST are synchronous inputs. Setup and hold times must be met for proper processor operation. $\overline{NMI}$ and $\overline{XINT7:0}$ may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge. For asynchronous operation, $\overline{NMI}$ and $\overline{XINT7:0}$ must be asserted for a minimum of two CLKIN periods to guarantee recognition.
2. $\overline{RDYRCV}$ and HOLD are synchronous inputs. Setup and hold times must be met for proper processor operation.
3. $\overline{RESET}$ may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge.

Table 17. Targeted 80960JA/JF Relative Output Timings

Symbol	Parameter	Min	Max	Units	Notes
T_{LXL}	ALE/$\overline{ALE}$ Width	$0.5T_C - 3$		ns	(1)
T_{LXA}	Address Hold from ALE/$\overline{ALE}$ Inactive	$0.5T_C - 3$		ns	Equal Loading(1)
T_{DXD}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Active	$0.5T_C - 3$		ns	Equal Loading(1)

NOTES:
1. Guaranteed by design. May not be 100% tested.

Table 18. Targeted 80960JA/JF Boundary Scan Test Signal Timings

Symbol	Parameter	Min	Max	Units	Notes
T_{BSF}	TCK Frequency		8	MHz	
T_{BSC}	TCK Period	125		ns	
T_{BSCH}	TCK High Time	40		ns	Measured at 1.5V (1)
T_{BSCL}	TCK Low Time	40		ns	Measured at 1.5V (1)
T_{BSCR}	TCK Rise Time		8	ns	0.8V to 2.0V (1)
T_{BSCF}	TCK Fall Time		8	ns	2.0V to 0.8V (1)
T_{BSIS1}	Input Setup to TCK—TDI, TMS	8		ns	
T_{BSIH1}	Input Hold from TCK—TDI, TMS	10		ns	
T_{BSOV1}	TDO Valid Delay	3	30	ns	Relative to falling edge of TCK
T_{BSOF1}	TDO Float Delay	3	36	ns	Relative to falling edge of TCK
T_{BSOV2}	All Outputs (Non-Test) Valid Delay	3	30	ns	Relative to falling edge of TCK
T_{BSOF2}	All Outputs (Non-Test) Float Delay	3	36	ns	Relative to falling edge of TCK
T_{BSIS2}	Input Setup to TCK—All Inputs (Non-Test)	8		ns	
T_{BSIH2}	Input Hold from TCK—All Inputs (Non-Test)	10		ns	

NOTES:
1.Not tested.

4.5.1 AC TEST CONDITIONS AND DERATING CURVES

The AC Specifications in Section 4.5, AC Specifications are tested with the 50 pF load indicated in Figure 6. Figure 9 shows how timings vary with load capacitance; Figure 10 shows how output rise and fall times vary with load capacitance.

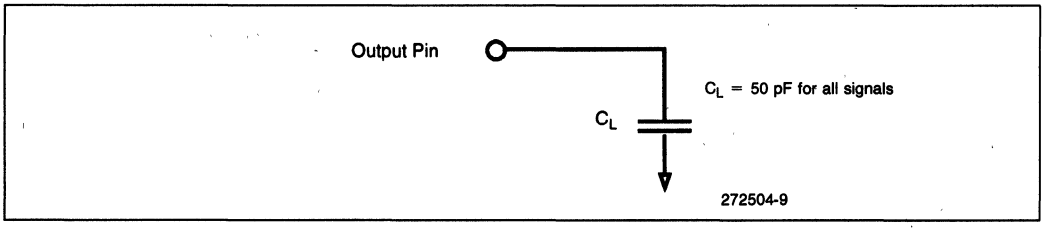


Figure 6. AC Test Load

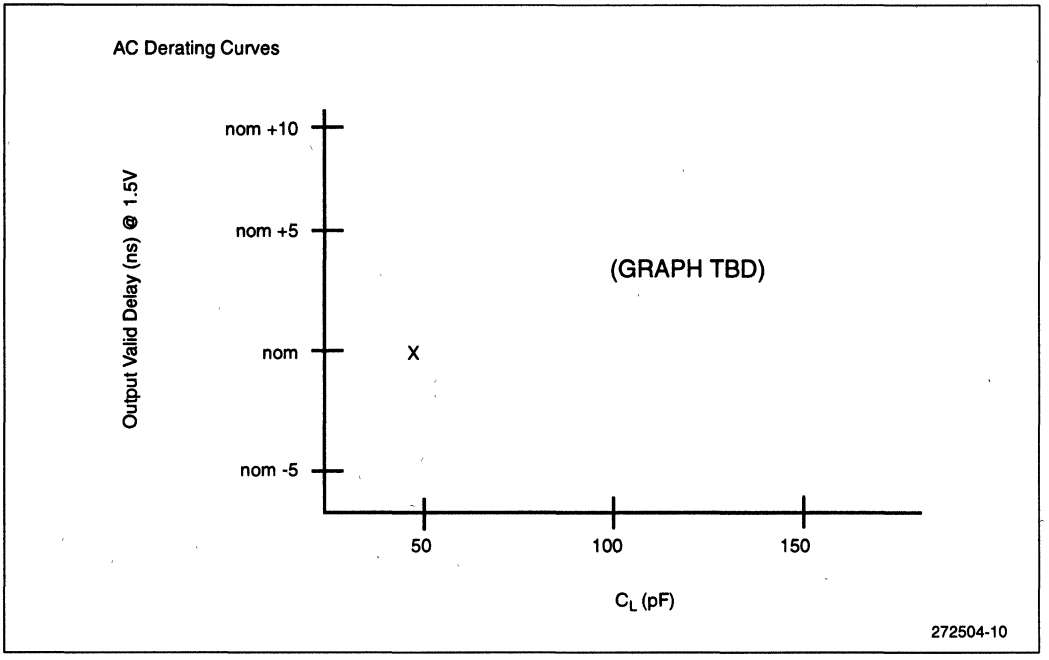


Figure 7. Output Delay or Hold vs. Load Capacitance

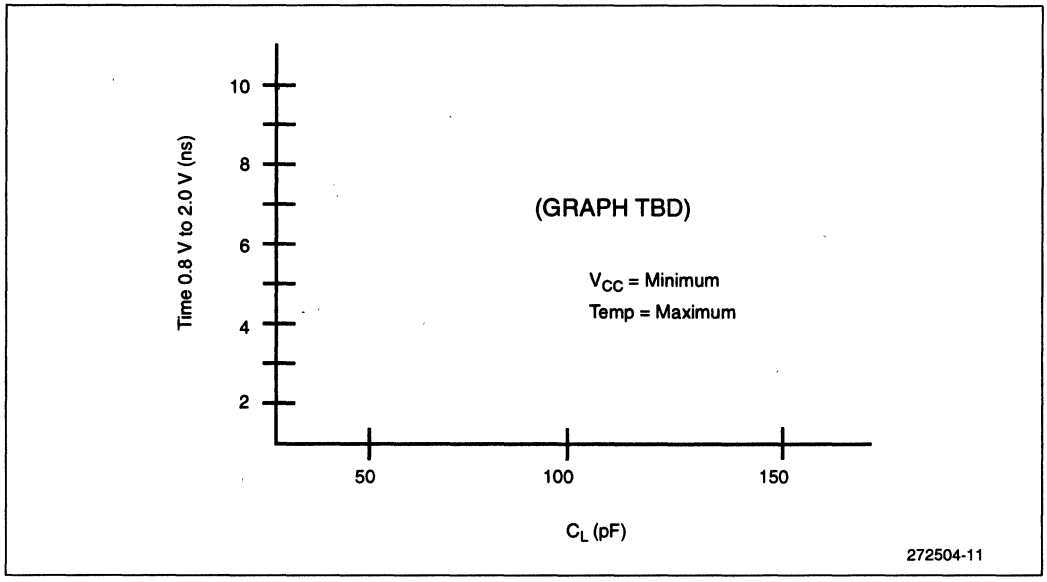


Figure 8. Rise and Fall Time Derating

4.5.2 AC TIMING WAVEFORMS

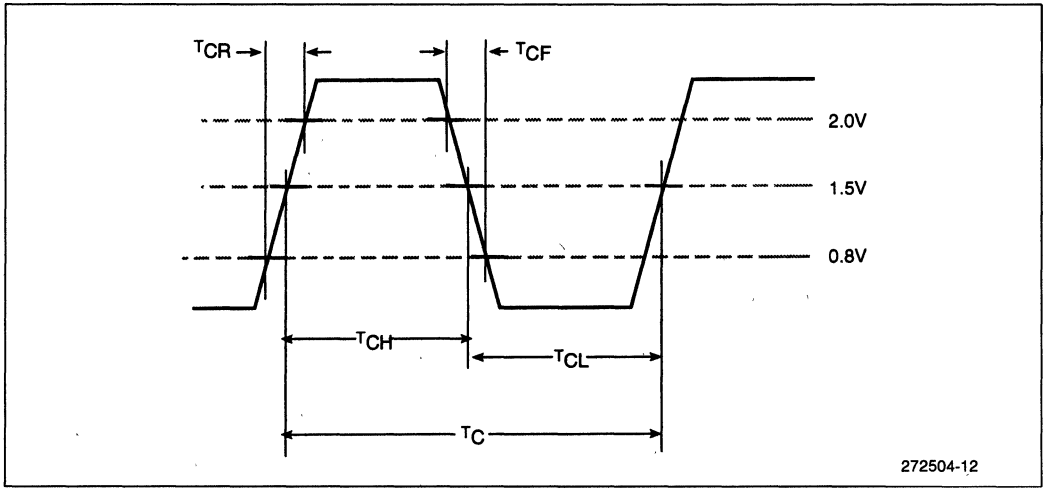


Figure 9. CLKIN Waveform

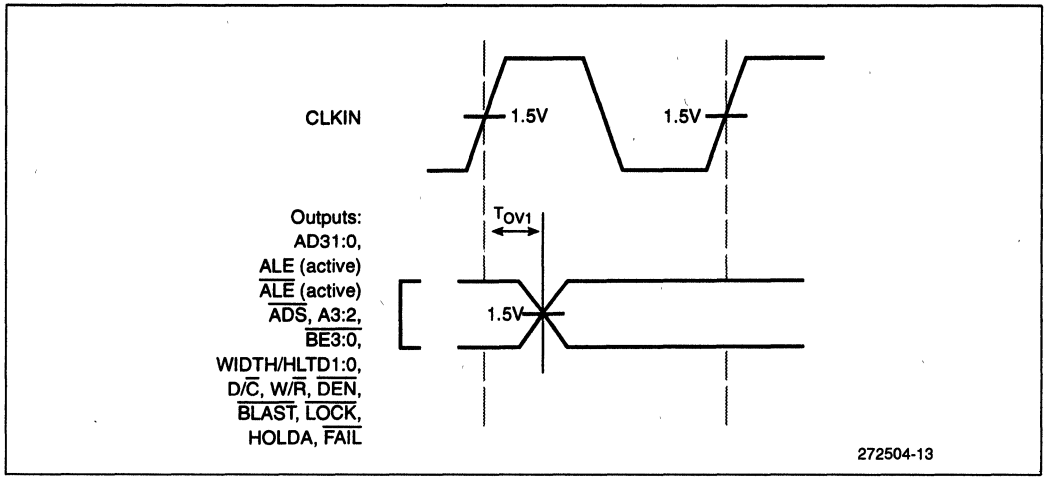


Figure 10. Output Delay Waveform for T_{OV1}

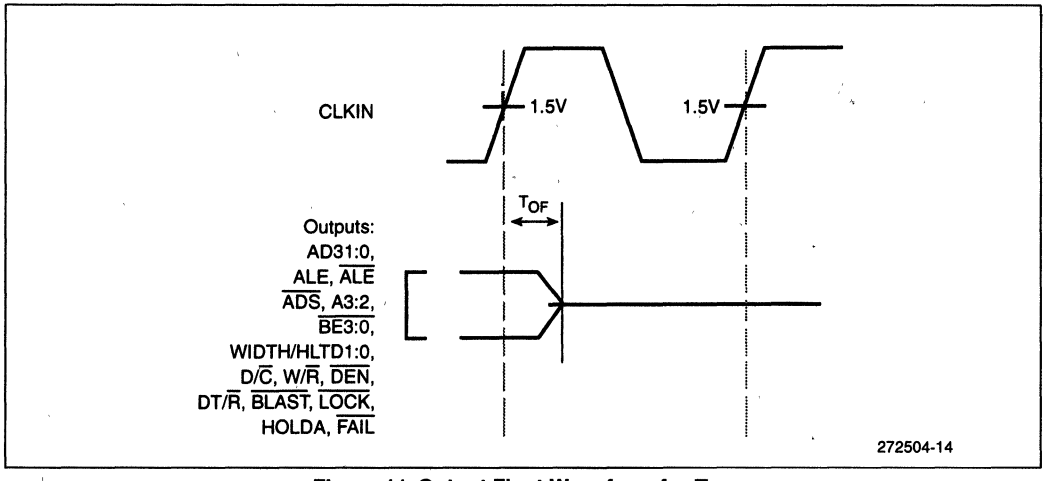


Figure 11. Output Float Waveform for T_{OF}

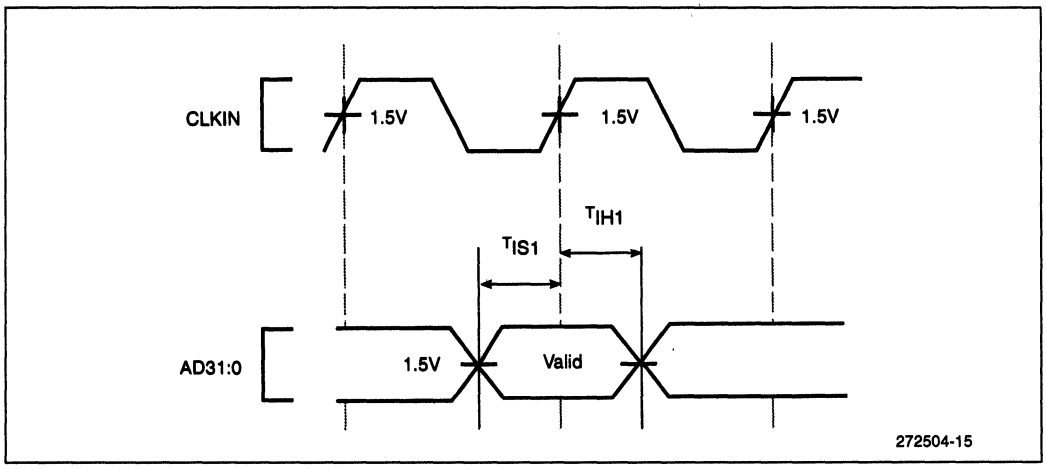


Figure 12. Input Setup and Hold Waveform for T_{IS1} and T_{IH1}

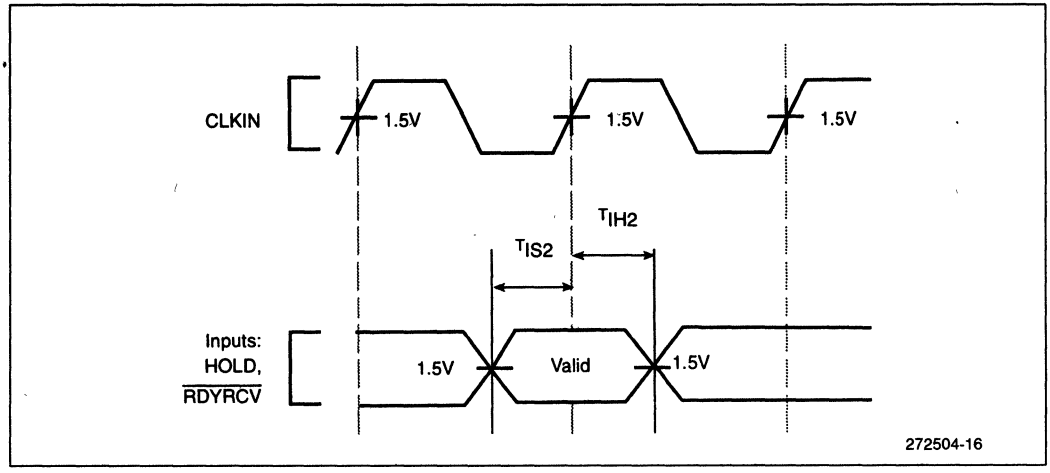


Figure 13. Input Setup and Hold Waveform for T_{IS2} and T_{IH2}

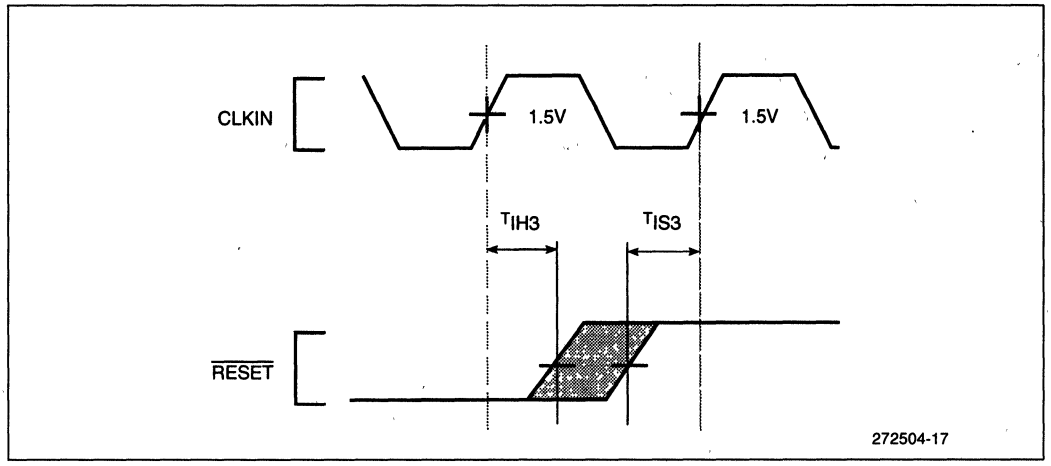


Figure 14. Input Setup and Hold Waveform for T_{IS3} and T_{IH3}

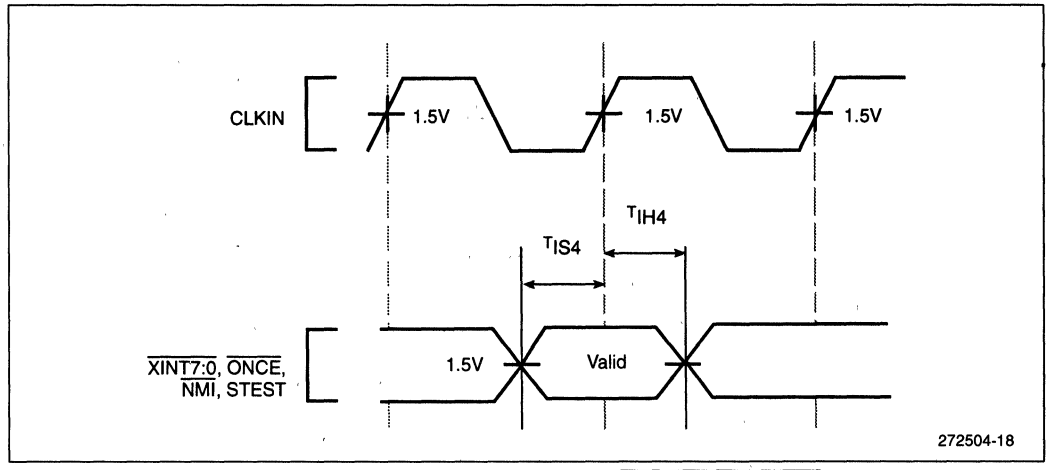


Figure 15. Input Setup and Hold Waveform for NMI, XINT7:0, ONCE, STEST

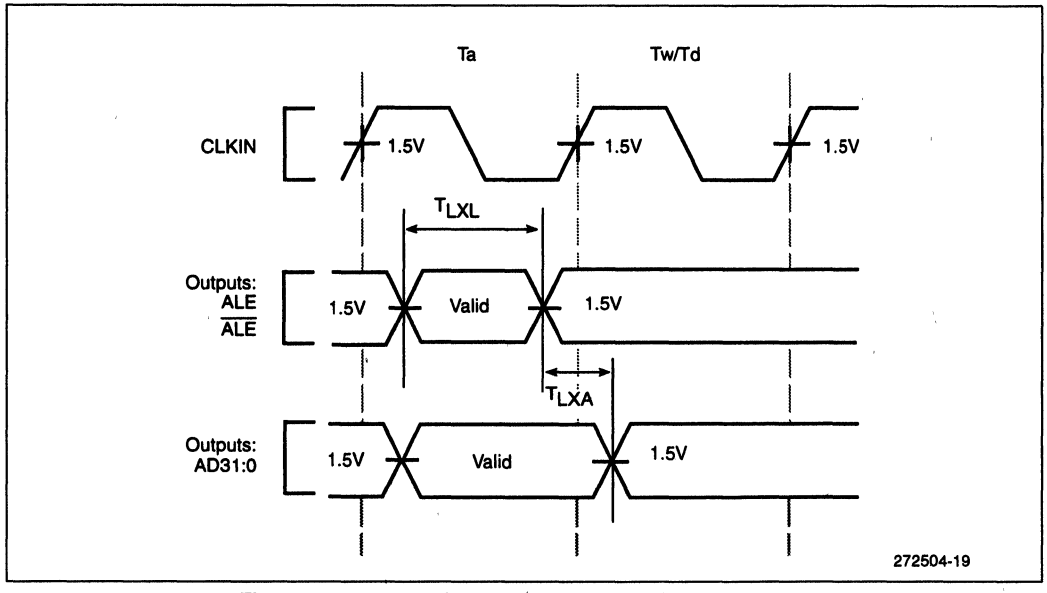


Figure 16. Relative Timings Waveform for T_{LXL} and T_{LXA}

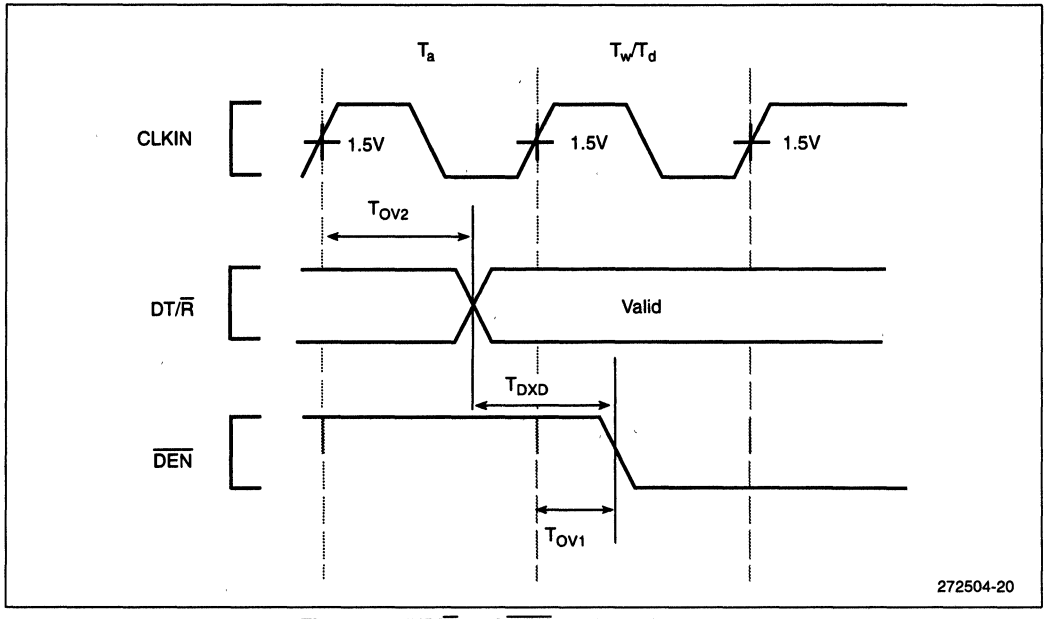


Figure 17. DT/$\overline{R}$ and $\overline{DEN}$ Timings Waveform

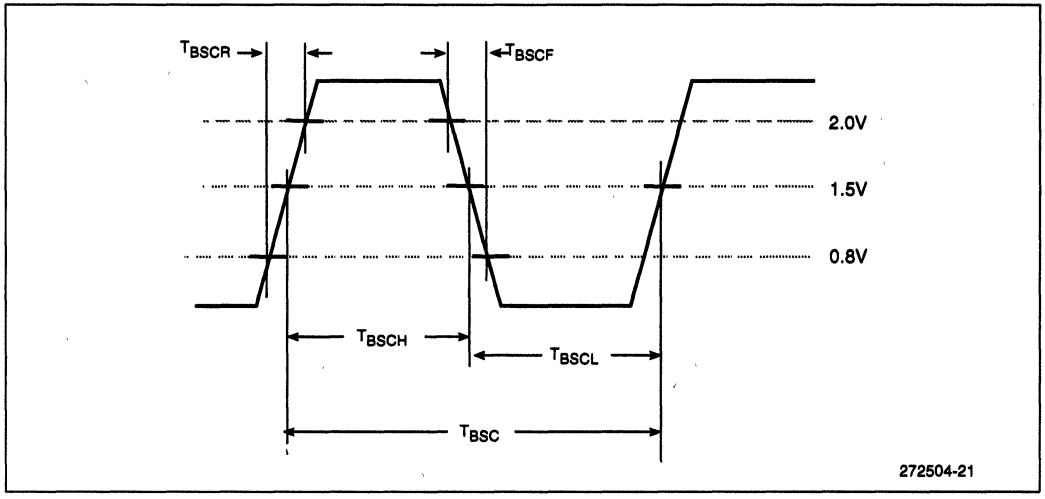


Figure 18. TCK Waveform

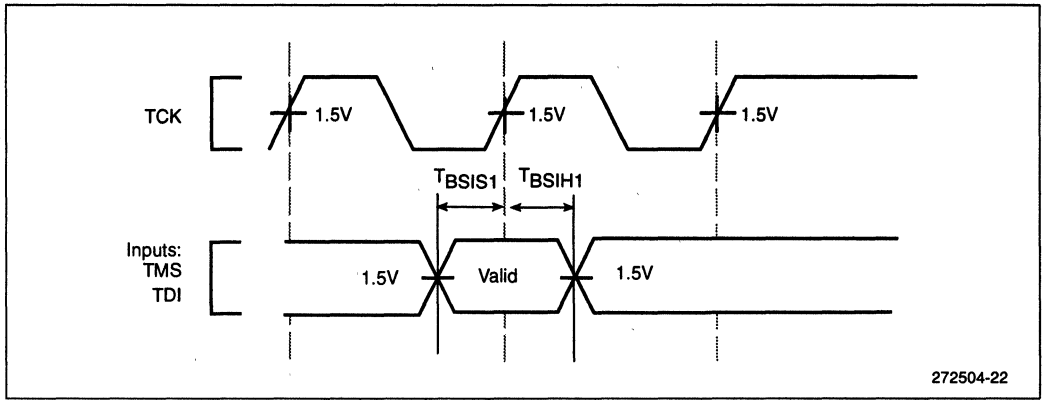


Figure 19. Input Setup and Hold Waveforms for T_{BSIS1} and T_{BSIH1}

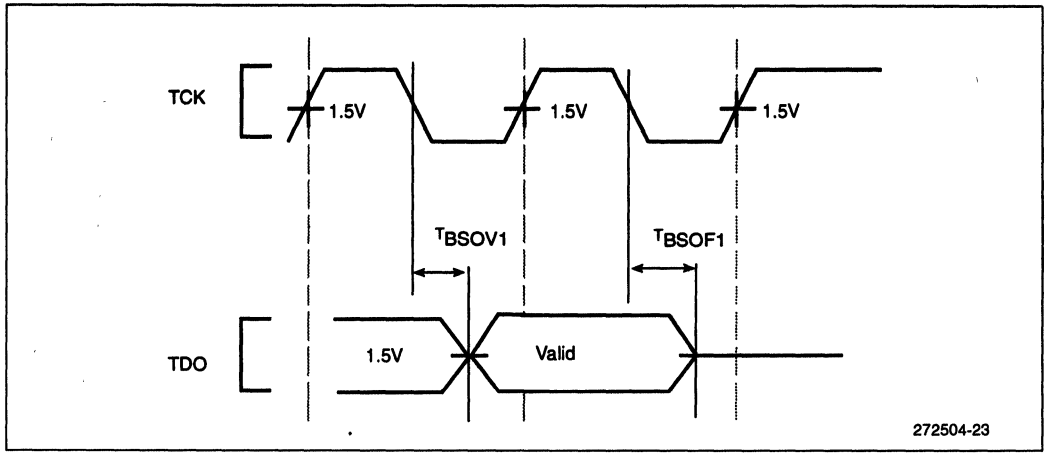


Figure 20. Output Delay and Output Float for T_{BSOV1} and T_{BSOF1}

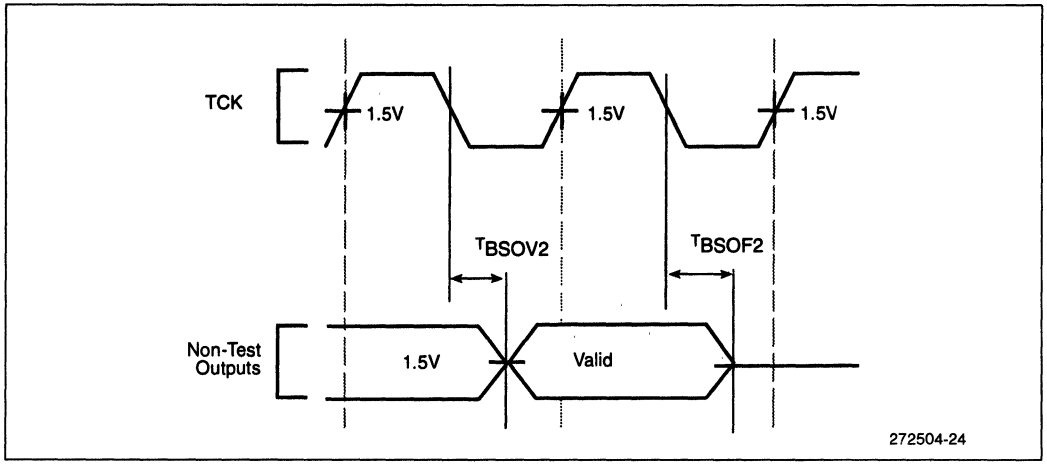


Figure 21. Output Delay and Output Float Waveform for T_{BSOV2} and T_{BSOF2}

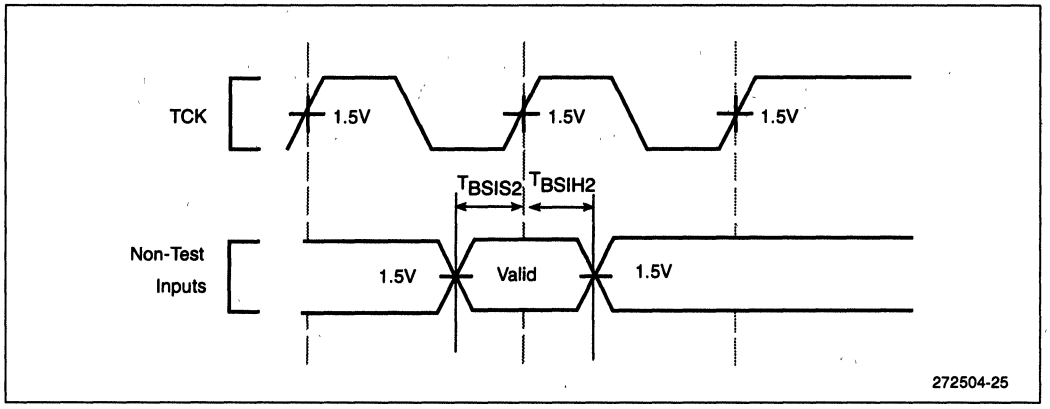


Figure 22. Input Setup and Hold Waveform for T_{BSIS2} and T_{BSIH2}

5.0 BUS FUNCTIONAL WAVEFORMS

Figures 23 through 28 illustrate typical 80960JA/JF bus transactions. Figure 29 depicts the bus arbitration sequence. Tables 19 through 22 summarize all possible combinations of bus accesses across 8-, 16-, and 32-bit buses according to data alignment. Figures 30 and 31 also show accesses on 32-bit buses. Figure 32 illustrates the processor reset sequence from the time power is applied to the device.

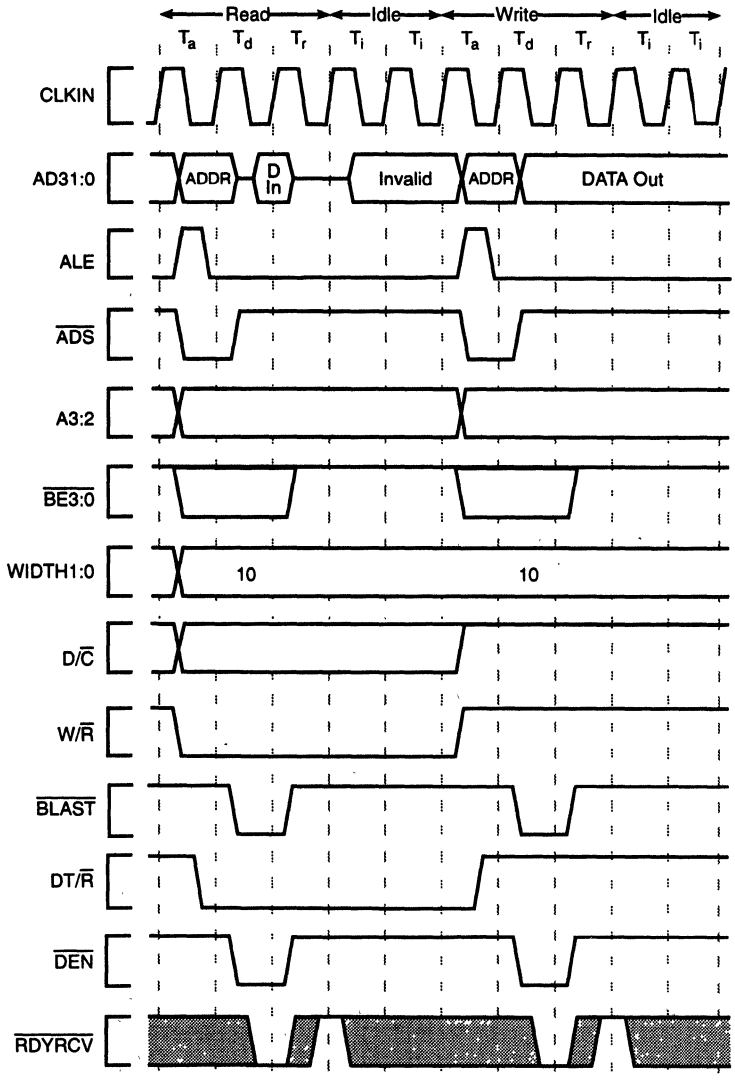


Figure 23. Non-Burst Read and Write Transactions Without Wait States, 32-Bit Bus

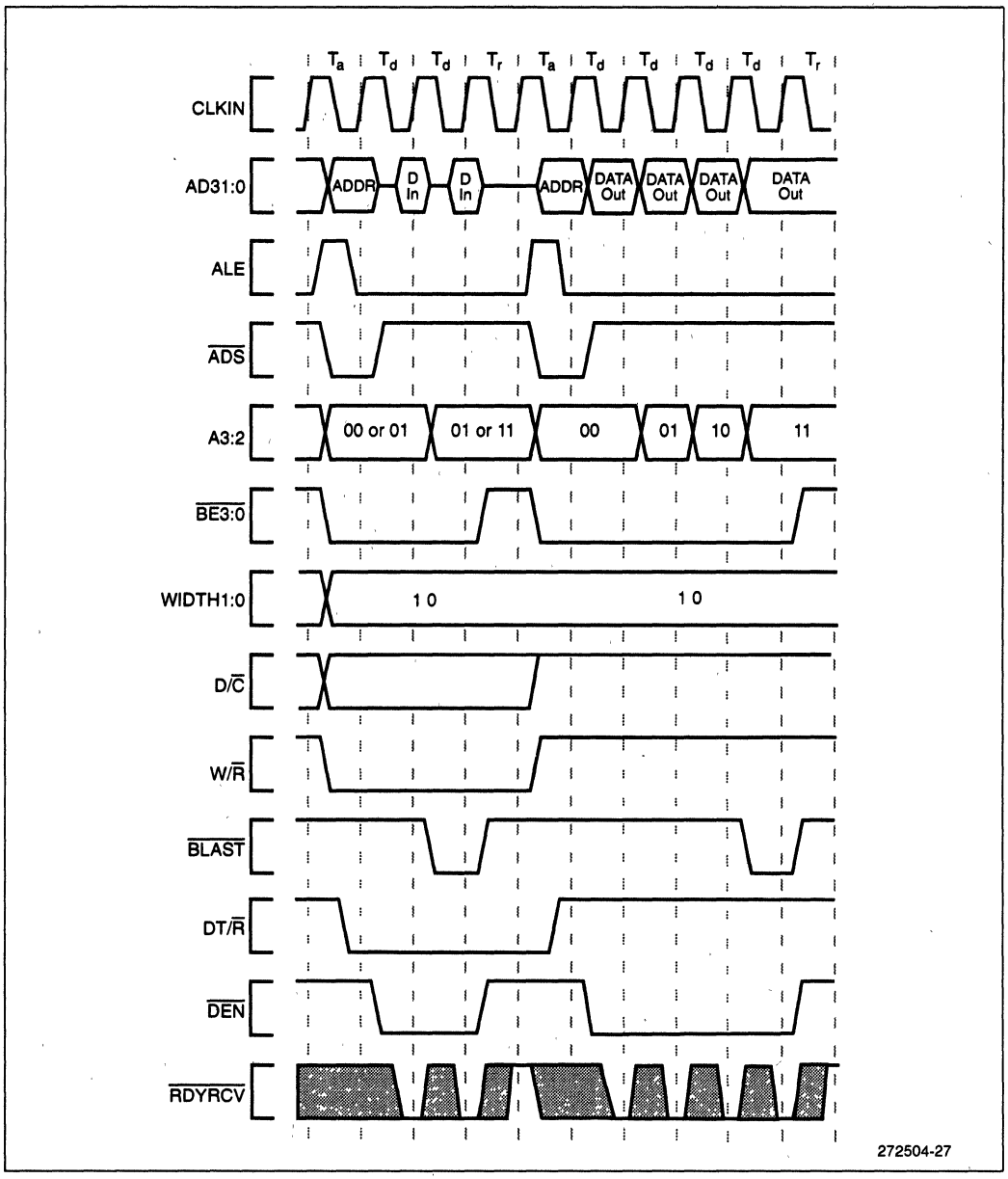


Figure 24. Burst Read and Write Transactions Without Wait States, 32-Bit Bus

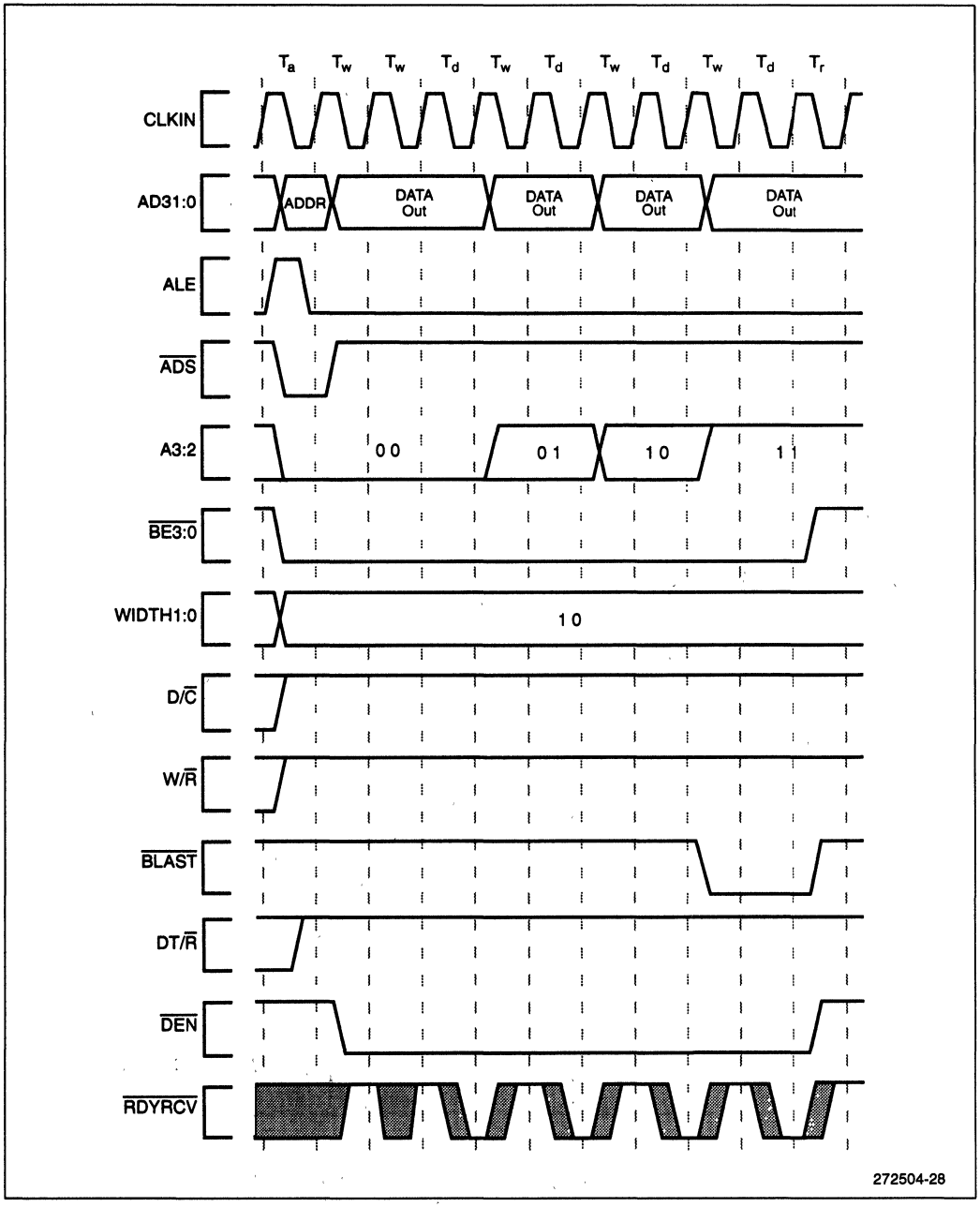


Figure 25. Burst Write Transactions With 2,1,1,1 Wait States, 32-Bit Bus

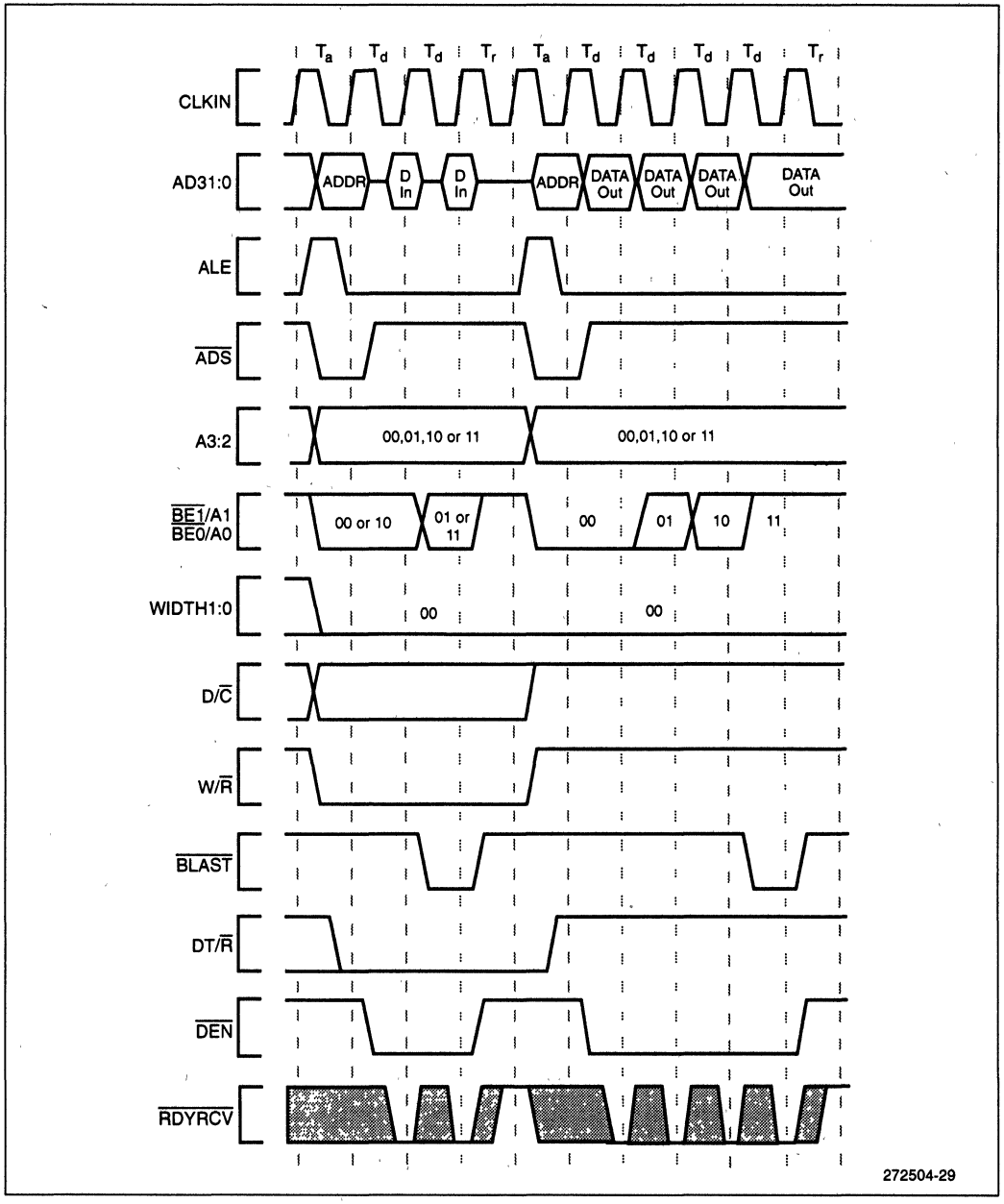


Figure 26. Burst Read and Write Transactions Without Wait States, 8-Bit Bus

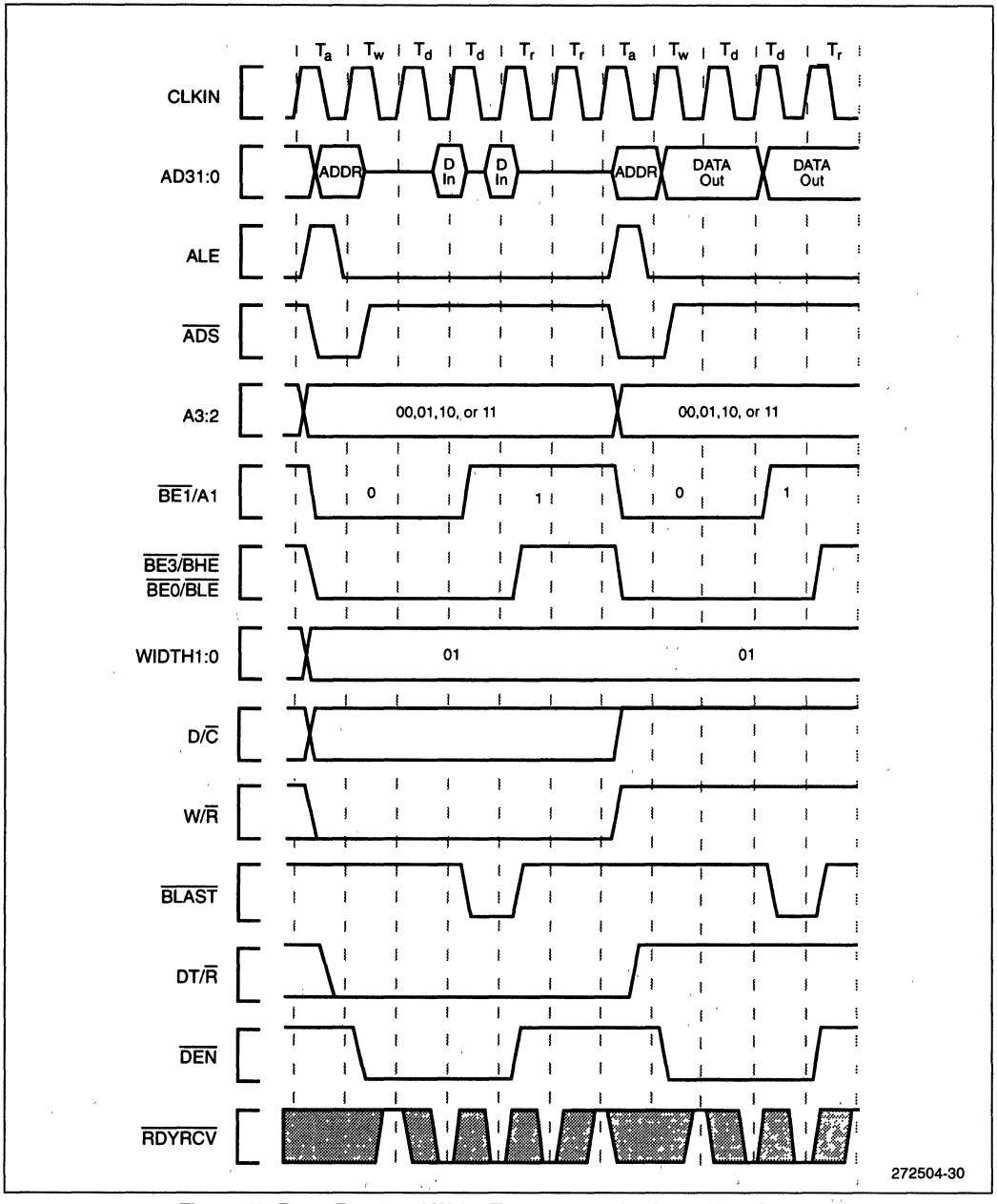


Figure 27. Burst Read and Write Transactions With 1, 0 Wait States and Extra Tr State on Read, 16-Bit Bus

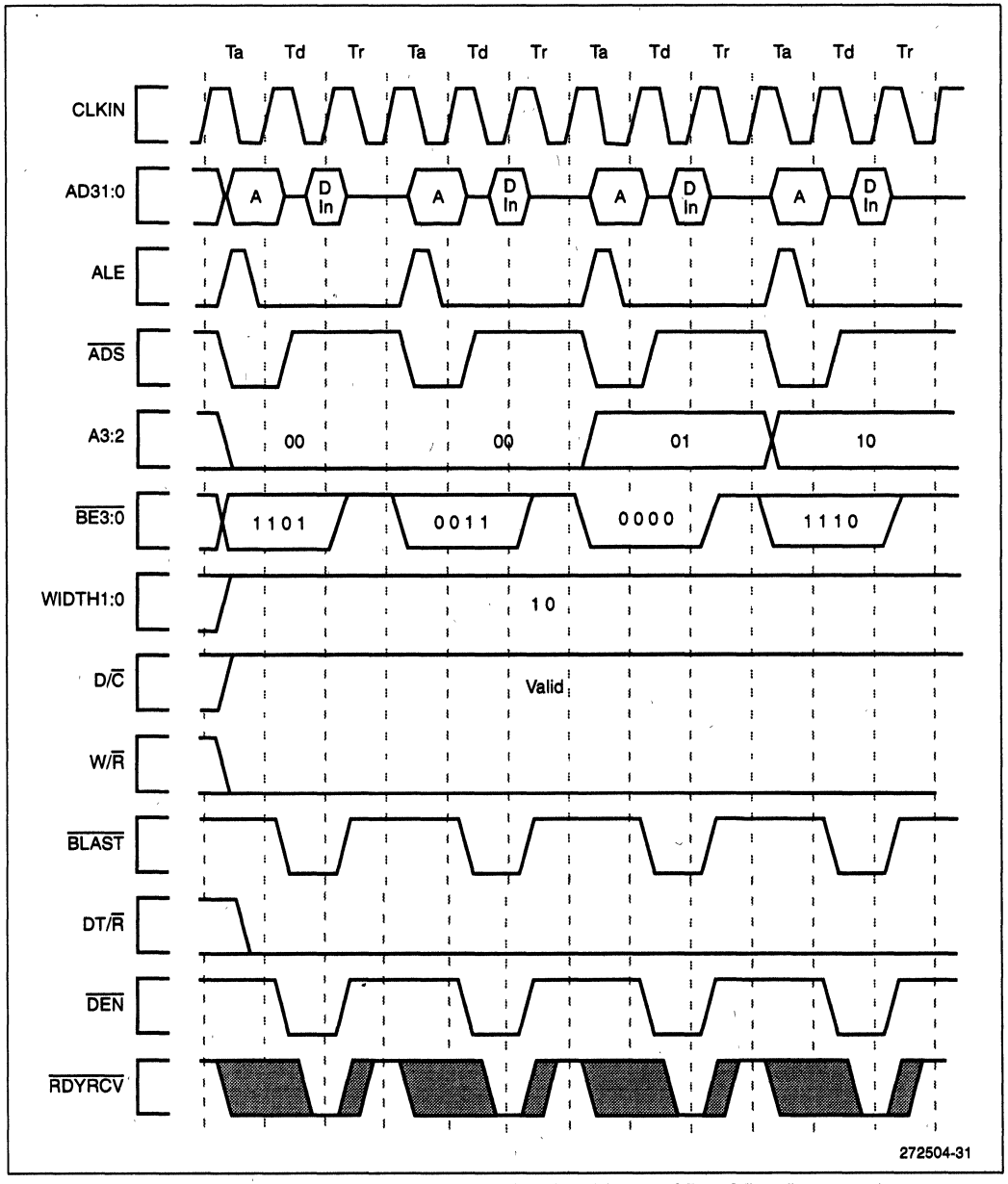
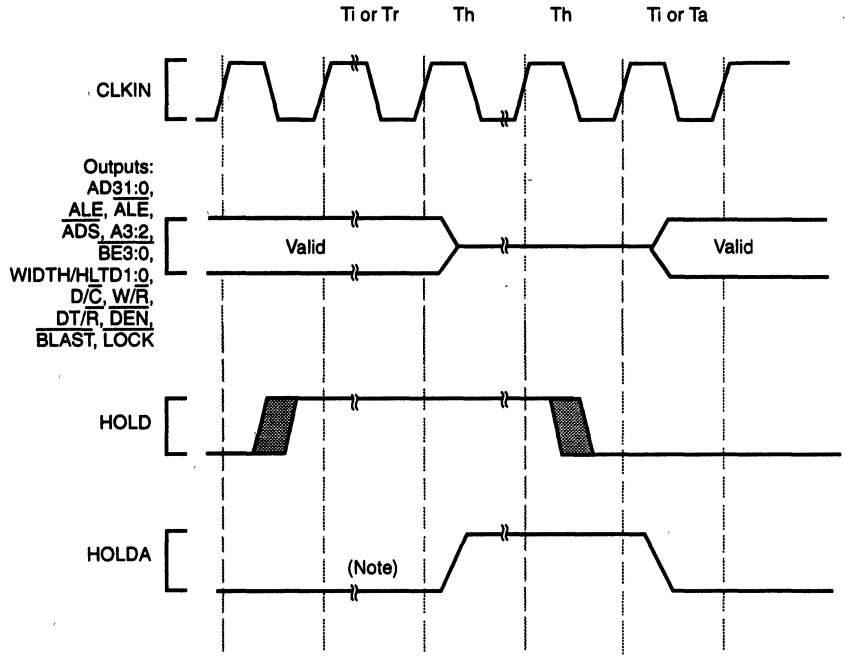


Figure 28. Bus Transactions Generated by Double Word Read Bus Request, Misaligned One Byte From Quad Word Boundary, 32-Bit Bus, Little Endian



NOTE:
HOLD is sampled on the rising edge of CLKIN. The processor asserts HOLDA to grant the bus on the same edge in which it recognizes HOLD if the last state was T_i or the last T_r of a bus transaction. Similarly, the processor deasserts HOLDA on the same edge in which it recognizes the deassertion of HOLD.

Figure 29. HOLD/HOLDA Waveform For Bus Arbitration

Table 19. Natural Boundaries for Load and Store Accesses

Data Width	Natural Boundary (Bytes)
Byte	1
Short Word	2
Word	4
Double Word	8
Triple Word	16
Quad Word	16

Table 20. Summary of Byte Load and Store Accesses

Address Offset from Natural Boundary (in Bytes)	Accesses on 8-Bit Bus (WIDTH1:0 = 00)	Accesses on 16-Bit Bus (WIDTH1:0 = 01)	Accesses on 32-Bit Bus (WIDTH1:0 = 10)
+0 (aligned)	• byte access	• byte access	• byte access

Table 21. Summary of Short Word Load and Store Accesses

Address Offset from Natural Boundary (in Bytes)	Accesses on 8-Bit Bus (WIDTH1:0 = 00)	Accesses on 16-Bit Bus (WIDTH1:0 = 01)	Accesses on 32-Bit Bus (WIDTH1:0 = 10)
+0 (aligned)	• burst of 2 bytes	• short-word access	• short-word access
+1	• 2 byte accesses	• 2 byte accesses	• 2 byte accesses

Table 22. Summary of *n*-Word Load and Store Accesses (n = 1, 2, 3, 4)

Address Offset from Natural Boundary (in Bytes)	Accesses on 8-Bit Bus (WIDTH1:0 = 00)	Accesses on 16-Bit Bus (WIDTH1:0 = 01)	Accesses on 32 Bit Bus (WIDTH1:0 = 10)
+0 (aligned) (*n* = 1, 2, 3, 4)	• *n* burst(s) of 4 bytes	• case *n* = 1: burst of *n* word(s) • case *n* = 2: burst of 4 short words • case *n* = 3: burst of 4 short words burst of 2 short words • case *n* = 4: 2 bursts of 4 short words	• burst of 2 short words
+1 (*n* = 1, 2, 3, 4) +5 (*n* = 2, 3, 4) +9 (*n* = 3, 4) +13 (*n* = 3, 4)	• byte access • burst of 2 bytes • *n*-1 burst(s) of 4 bytes • byte access	• byte access • short-word access • *n*-1 burst(s) of 2 short words • byte access	• byte access • short-word access • *n*-1 word access(es) • byte access
+2 (*n* = 1, 2, 3, 4) +6 (*n* = 2, 3, 4) +10 (*n* = 3, 4) +14 (*n* = 3, 4)	• burst of 2 bytes • *n*-1 burst(s) of 4 bytes • burst of 2 bytes	• short-word access • *n*-1 burst(s) of 2 short words • short-word access	• short-word access • *n*-1 word access(es) • short-word access
+3 (*n* = 1, 2, 3, 4) +7 (*n* = 2, 3, 4) +11 (*n* = 3, 4) +15 (*n* = 3, 4)	• byte access • *n*-1 burst(s) of 4 bytes • burst of 2 bytes • byte access	• byte access • *n*-1 burst(s) of 2 short words • short-word access • byte access	• byte access • *n*-1 word access(es) • short-word access • byte access
+4 (*n* = 2, 3, 4) +8 (*n* = 3, 4) +12 (*n* = 3, 4)	• *n* burst(s) of 4 bytes	• *n* burst(s) of 2 short words	• *n* word access(es)

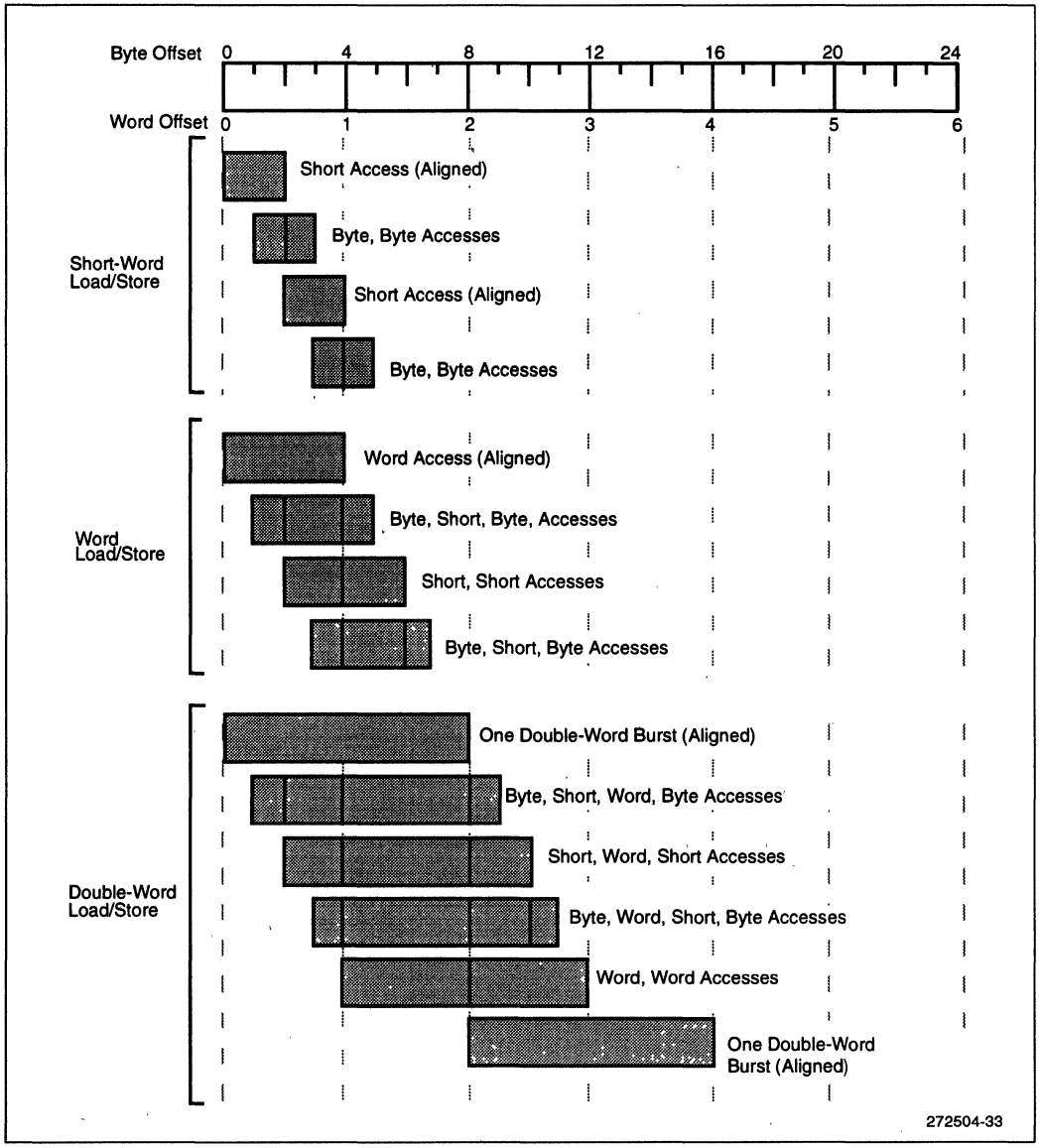


Figure 30. Summary of Aligned and Unaligned Accesses (32-Bit Bus)

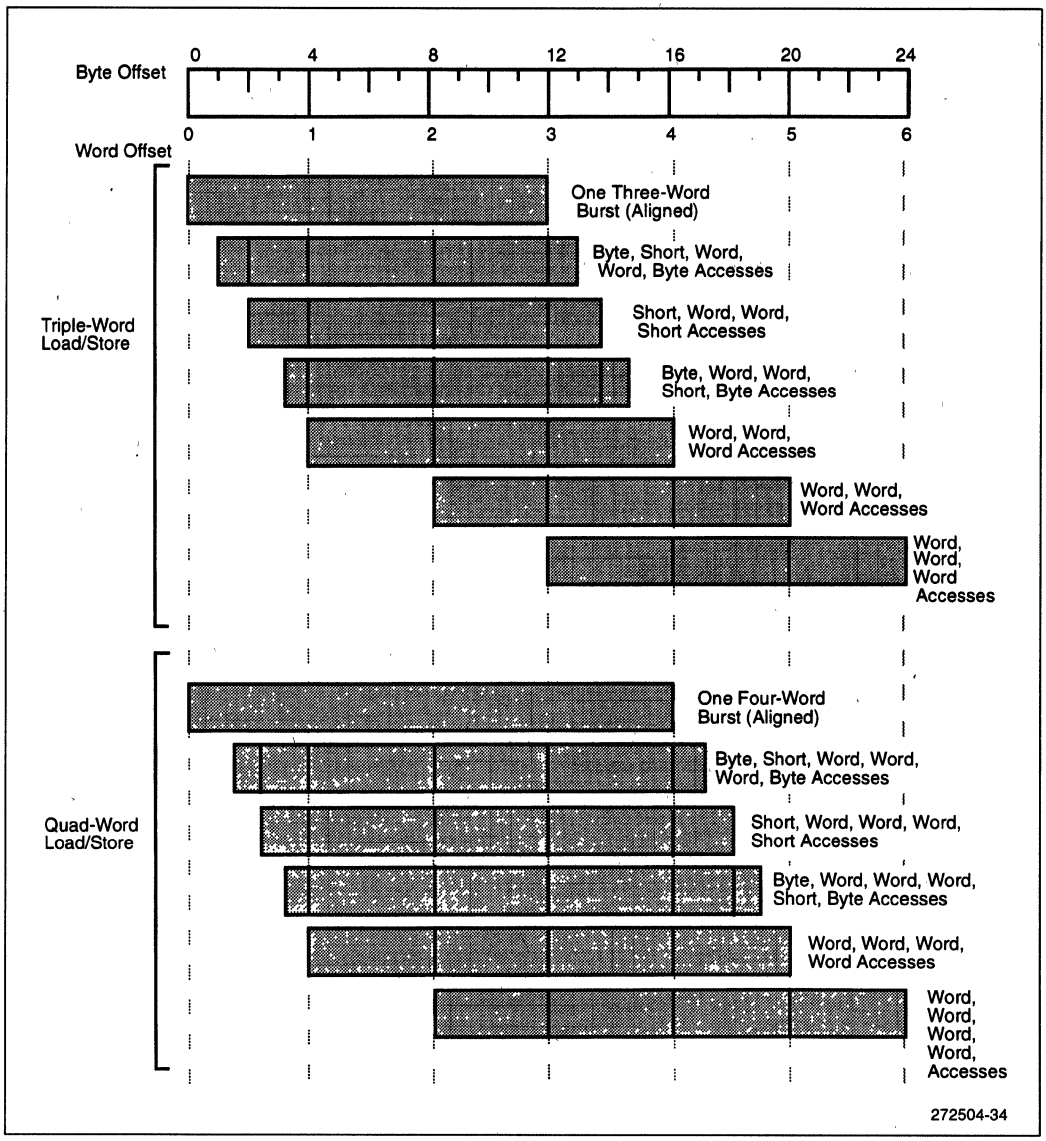
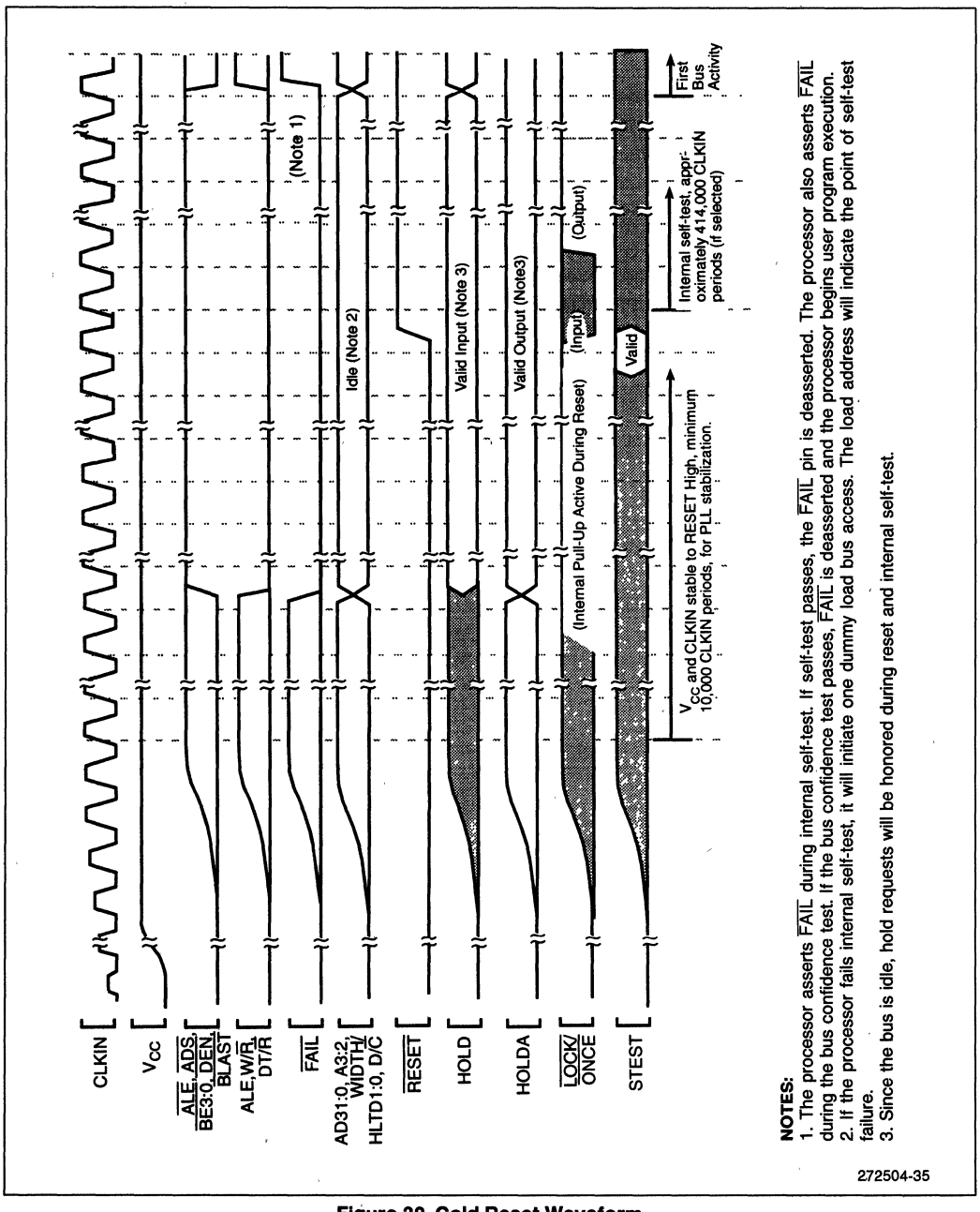


Figure 31. Summary of Aligned and Unaligned Accesses (32-Bit Bus) (Continued)



NOTES:
1. The processor asserts $\overline{FAIL}$ during internal self-test. If self-test passes, the $\overline{FAIL}$ pin is deasserted. The processor also asserts $\overline{FAIL}$ during the bus confidence test. If the bus confidence test passes, $\overline{FAIL}$ is deasserted and the processor begins user program execution.
2. If the processor fails internal self-test, it will initiate one dummy load bus access. The load address will indicate the point of self-test failure.
3. Since the bus is idle, hold requests will be honored during reset and internal self-test.

272504-35

Figure 32. Cold Reset Waveform

6.0 DEVICE IDENTIFICATION

80960JA/JF processors may be identified electrically according to device type and stepping (see Table 23). The 32-bit identifier is accessible in three ways:

- Upon reset, the identifier is placed into the g0 register.
- The identifier may be accessed from supervisor mode at any time by reading the DEVICEID register at address FF008710H.
- The IEEE Standard 1149.1 Test Access Port may select the DEVICE ID register through the IDCODE instruction.

The stepping number is also printed on the top side of the product package.

Table 23. 80960JA/JF Die and Stepping Reference

Device and Stepping	Version Number	Part Number	Manufacturer	X	Complete ID (Hex)
80960JA A0	0000	1000 1000 0010 0001	0000 0001 001	1	08821013
80960JF A0	0000	1000 1000 0010 0000	0000 0001 001	1	08820013

7.0 REVISION HISTORY

This data sheet supersedes revision 272504-001. Table 24 indicates significant changes since the previous revision.

Table 24. Revision History (Sheet 1 of 2)

Section	Last Rev.	Description
Figure 2, 80960JA/JF Block Diagram (pg. 2)	-001	Added 80960JA product version, which includes 2 Kbyte instruction cache and 1 Kbyte data cache.
Table 1, 80960JA/JF Instruction Set (pg. 5)	-001	Renamed Conditional Move instruction to Conditional Select instruction.
Table 3, Pin Description—External Bus Signals (pg. 7)	-001	Clarified description of unused AD31:0 pins during write operations. Corrected description of AD31:0 pins during HALT mode. Renamed $\overline{ADS}$ pin from Address/ Data Status to Address Strobe (pin function unchanged). Corrected description of $\overline{BE3:2}$ pins to indicate high state when pin(s) unused.
Table 4, Pin Description—Processor Control Signals, Test Signals and Power (pg. 10)	-001	Added note that TDO does not float during ONCE Mode. Added note to ground $\overline{TRST}$ if TAP unused.
Figure 3, 132-Lead Pin Grid Array Bottom View—Pins Facing up (pg. 12) Figure 4, 132-Lead Pin Grid Array Top View—Pins Facing down (pg. 13) Table 6, 132-Lead PGA Pinout—In Signal Order (pg. 14) Table 7, 132-Lead PGA Pinout—In Pin Order (pg. 15)	-001	New figures and tables. Pin C8 changed from V_{CC} (previously published elsewhere) to NC. Processor operates correctly in a system with either connection.

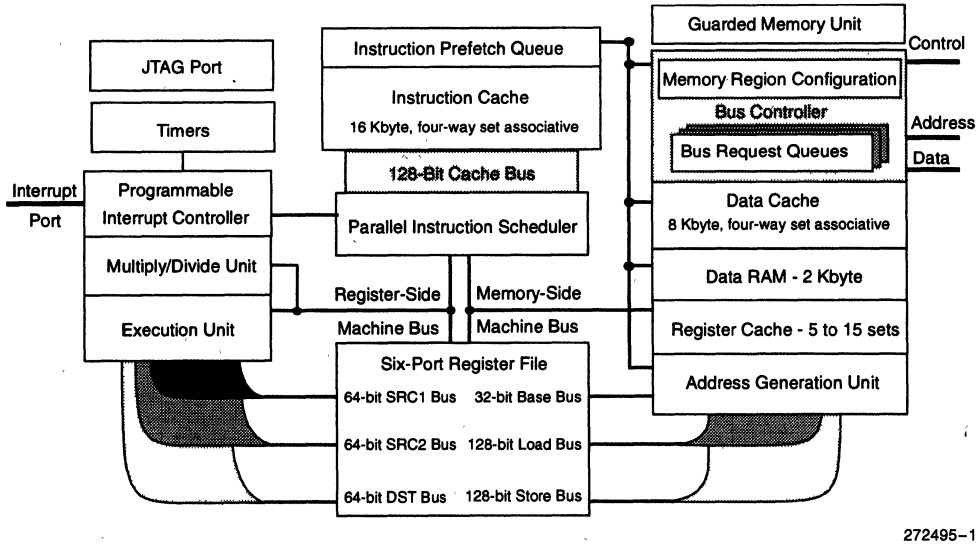
Table 24. Revision History (Sheet 2 of 2) (Continued)

Section	Last Rev.	Description
Figure 5, 132-Lead PQFP—Top View (pg. 16) Table 8, 132-Lead PQFP Pinout—In Signal Order (pg. 18) Table 9, 132-Lead PQFP Pinout—In Pin Order (pg. 18)	-001	New figure and tables. Lead 19 changed from $V_{CC(I/O)}$ (previously published elsewhere) to NC. Processor operates correctly in a system with either connection.
Section 3.2, Package Thermal Specifications (pg. 19)	-001	New section with tables of thermal characteristics.
Section 4.0, ELECTRICAL SPECIFICATIONS (pg. 21)	-001	New section with targeted operating conditions, targeted DC characteristics, targeted AC specifications and AC timing waveforms.
Figure 29, HOLD/HOLDA Waveform For Bus Arbitration (pg. 42)	-001	Improved figure.
Table 19, Natural Boundaries for Load and Store Accesses (pg. 42) Table 20, Summary of Byte Load and Store Accesses (pg. 43) Table 21, Summary of Short Word Load and Store Accesses (pg. 43) Table 22, Summary of n-Word Load and Store Accesses (n = 1, 2, 3, 4) (pg. 43) Figure 30, Summary of Aligned and Unaligned Accesses (32-Bit Bus) (pg. 44)	-001	New tables and figures to explain bus access alignment.
Figure 32, Cold Reset Waveform (pg. 46)	-001	New figure.
Section 6.0, DEVICE IDENTIFICATION (pg. 47)	-001	New section.

80960HA/HD/HT
32-BIT HIGH-PERFORMANCE SUPERSCALAR PROCESSOR

•Binary Compatible with Other 80960 Processors
•Two Instructions/Clock Sustained Execution

- **32-Bit Parallel Architecture**
  - **— Load/Store Architecture**
  - **— Sixteen 32-Bit Global Registers**
  - **— Sixteen 32-Bit Local Registers**
  - **— 1.2 Gbyte/second Internal Bandwidth (75 MHz)**
  - **— On-Chip Register Cache**
- **High-Performance On-Chip Storage**
  - **— 16 Kbyte Four-Way Set Associative Instruction Cache**
  - **— 8 Kbyte Four-Way Set Associative Data Cache**
  - **— 2 Kbyte General Purpose RAM**
  - **— Separate 128 Bit Internal Paths For Instructions/Data**
- **3.3V Supply Voltage**
  - **— 5V Tolerant Inputs**
  - **— TTL Compatible Outputs**
- **Guarded Memory Unit**
  - **— Provides Memory Protection**
  - **— User/Supervisor Read/Write/ Execute**
- **32-Bit Demultiplexed Burst Bus**
  - **— Per Byte Parity Generation and Checking**
  - **— Address Pipelining Option**
  - **— Fully Programmable Wait State Generator**
  - **— Supports 8-, 16- or 32-Bit Bus Widths**
  - **— 160 Mbyte/second External Bandwidth (40 MHz)**
- **High-Speed Interrupt Controller**
  - **— Up to 240 External Interrupts**
  - **— 31 Fully Programmable Priorities**
  - **— Separate, Non-maskable Interrupt Pin**
- **On-Chip 32-Bit Timers**
  - **— Auto Reload Capability and One-shot**
  - **— CLKIN Prescaling, ÷ 1, 2, 4, or 8**
- **JTAG Support-IEEE 1149.1 Compliant**
- **Processor Core Clock**
  - **— 80960HA is 1x bus clock**
  - **— 80960HD is 2x bus clock**
  - **— 80960HT is 3x bus clock**



272495-1

Figure 1. 80960HA Block Diagram

October 1994
Order Number: 272495-002

80960HA/HD/HT
32-BIT HIGH-PERFORMANCE SUPERSCALAR PROCESSOR

CONTENTS

FIGURES

FIGURES

FIGURES

TABLES

1.0 PURPOSE

This document provides a descriptive preview of the Intel's i960r Hx embedded superscalar microprocessors, including (see Table 1):

- 80960HA—executes instructions at the same frequency as the bus (CLKIN)
- 80960HD—clock doubled: executes instructions at twice the bus frequency
- 80960HT—clock tripled: executes instructions at three times the bus frequency

Future revisions of this data sheet will provide targeted electrical characteristics. Detailed descriptions for functional topics—other than parametric performance—will be published in the *i960® Hx Microprocessor User's Guide*.

2.0 80960Hx OVERVIEW

Intel's 80960Hx is the performance follow-on product to the 80960CF microprocessor. The 80960Hx is binary code-compatible with the 80960CF; this allows customers to create CA/CF designs which can use the 80960Hx.

As shown in Figure 1, the 80960Hx instruction cache is 16 Kbytes; data cache is 8 Kbytes and the data RAM expanded to 2 Kbytes.

The 80960Hx is pin and binary code compatible with the 32-bit 80960Cx Core Architecture. It includes Special Function Register extensions to control on-chip peripherals, and instruction set extensions to shift 64-bit operands and configure on-chip hardware. Multiple 128-bit internal buses, on-chip instruction caching and a sophisticated instructio n scheduler allow the processor to sustain execution of two instructions every clock and peak at execution of three instructions per clock.

A 32-bit demultiplexed and pipelined burst bus interface provides a maximum 160 Mbyte/s bandwidth to the external memory subsystem. Also, the 80960Hx's support for on-chip instruction caching, procedure context and critical program data substantially decouple system performance from the wait states associated with accesses to the system's s lower, cost sensitive, main memory subsystem.

The 80960Hx bus controller integrates full wait state and bus width control providing high system performance with minimal system design complexity. Unaligned access and Big Endian byte order support reduces the cost of porting existing applications to the 80960Hx. The Big Endian and unaligned word loads and stores are implemented in hardware.

The interrupt controller provides full programmability of 240 interrupt sources into 31 priority levels.

2.1 The H-Series Core

The H-Series core is a very high performance microarchitectural implementation of the 80960 Core Architecture. This core can sustain execution of two instructions per core clock (150 MIPS at 25 MHz in 3x clock mode). To achieve this level of performance, Intel has incorporated state-of-the-art silicon technology and innovative microa rchitectural constructs into the H-Series core implementation. Factors that contribute to the core's performance include:

- Parallel instruction decoding allows issuance of up to three instructions per clock
- Single-clock execution of most instructions
- Parallel instruction decode allows sustained, simultaneous execution of two instructions every clock cycle
- Efficient instruction pipeline minimizes pipeline break losses
- Register and resource scoreboarding allow simultaneous multi-clock instruction execution
- Branch look-ahead and prediction allows many branches to execute with no pipeline break
- Local Register Cache integrated on-chip caches Call/Return context
- Four-way set associative, 16 Kbyte integrated instruction cache
- 8 Kbyte data cache, write through, write allocate
- 2 Kbyte integrated Data RAM sustains a four-word (128-bit) access every clock cycle
- Processor core clock frequency is a multiple of the bus clock
- Unaligned word and short-word accesses are implemented in hardware
- Big Endian byte order implemented in hardware

Table 1. 80960Hx Product Description

Product	Core	Voltage	Operating Frequency (bus/core)
80960HA	1x	3.3V	25/25, 33/33, 40/40
80960HD	2x	3.3V	16/32, 25/50, 33/66
80960HT	3x	3.3V	20/60, 25/75

2.2 Pipelined, Burst Bus

A 32-bit high performance bus controller interfaces the 80960Hx core to the external memory and peripherals. The Bus Control Unit features a maximum transfer rate of 160 Mbytes per second (at 40 MHz). Internally programmable wait states and 16 separately configurable physical memory regions allow the processor to interface with a va riety of memory subsystems with a minimum of system complexity and a maximum of performance.

The Bus Controller's main features include:

- Demultiplexed, Burst Bus to exploit most efficient DRAM access modes
- Address Pipelining to reduce memory cost while maintaining performance
- 32-, 16- and 8-bit modes for I/O interfacing ease
- Full internal wait state generation to reduce system cost
- Little and Big Endian support to ease application development
- High performance Unaligned access support for code portability
- Three-deep request queue to decouple the bus from the core
- Decoupled physical (wait state profile, bus width, parity) and logical (cacheability, big/little endian) configurations

2.3 Priority Interrupt Controller

A programmable priority interrupt controller manages up to 240 external sources through the 8-bit external interrupt port. The Interrupt Unit also handles the two internal sources from the Timers and a single non-maskable interrupt input. The 8-bit interrupt port can also be configured to provide individual interrupt sources that are level or edge triggered.

80960Hx interrupts are prioritized and signaled within TBD ns of the request. If the interrupt is of higher priority than the processor priority, the context switch to the interrupt routine typically completes in another TBD ns. The interrupt unit provides the mechanism for the low latency and high throughput interrupt service which is essential for embedded applications.

2.4 Instruction Set Summary

Table 2 summarizes the 80960Hx instruction set by logical groupings.

Table 2. 80960Hx Instruction Set

Data Movement	Arithmetic	Logical	Bit/Bit Field/Byte
Load Store Move Load Address Conditional Select[2]	Add Subtract Multiply Divide Remainder Modulo Shift Extended Shift Extended Multiply Extended Divide Add with Carry Subtract with Carry Rotate Conditional Add[2] Conditional Subtract[2]	And Not And And Not Or Exclusive Or Not Or Or Not Nor Exclusive Nor Not Nand	Set Bit Clear Bit Not Bit Alter Bit Scan For Bit Span Over Bit Extract Modify Scan Byte for Equal Byte Swap[2]
Comparison	**Branch**	**Call/Return**	**Fault**
Compare Conditional Compare Compare and Increment Compare and Decrement Compare byte[2] Compare short[2] Test Condition Code Check Bit	Unconditional Branch Conditional Branch Compare and Branch	Call Call Extended Call System Return Branch and Link	Conditional Fault Synchronize Faults
Debug	**Processor Mgmt**	**Atomic**	**Cache Control**
Modify Trace Controls Mark Force Mark	Flush Local Registers Modify Arithmetic Control Modify Process Controls Interrupt Enable/Disable[1,2] System Control[1] HALT[1,2]	Atomic Add Atomic Modify	Instruction Cache Control[1,2] Data Cache Control[1,2]

NOTES:
1. 80960Hx extensions to the 80960 core instruction set.
2. 80960Hx extensions to the 80960Cx instruction set

3.0 PACKAGE INFORMATION

This section describes the pins, pinouts and thermal characteristics for the 80960Hx in the 168-pin Ceramic Pin Grid Array (PGA) package, 208-pin power quad II (SQFP). For complete package specifications and information, see the Intel Packaging Handbook (Order #240800).

3.1 Pin Descriptions

This section defines the 80960Hx pins. Table 3 presents the legend for interpreting the pin descriptions in Table 4. All pins float while the processor is in the ONCE mode, except TDO.

Table 3. Pin Description Nomenclature

Symbol	Description
I	Input only pin O Output only pin
I/O	Pin can be input or output
—	Pin must be connected as indicated for proper device functionality
S(E)	Synchronous edge sensitive input. This input must meet the setup and hold times relative to CLKIN to ensure proper operation of the processor.
S(L)	Synchronous level sensitive input. This input must meet the setup and hold times relative to CLKIN to ensure proper operation of the processor.
A(E)	Asynchronous edge sensitive input.
A(L)	Asynchronous level sensitive input.
H(. . . .)	While the processor bus is in the HOLD state (HLDA asserted), the pin: H(1) is driven to V_{CC} H(0) is driven to V_{SS} H(Z) floats H(Q) continues to be a valid output
B(. . . .)	While the processor is in the bus backoff state ($\overline{BOFF}$ asserted), the pin: B(1) is driven to V_{CC} B(0) is driven to V_{SS} B(Z) floats B(Q) continues to be a valid output
R(. . . .)	While the processor's $\overline{RESET}$ pin is asserted, the pin: R(1) is driven to V_{CC} R(0) is driven to V_{SS} R(Z) floats R(Q) continues to be a valid output
P(. . . .)	While the processor is in HALT mode: P(1) is driven to V_{CC} P(0) is driven to V_{SS} P(Z) floats P(Q) continues to be a valid output

1

Table 4. 80960Hx Processor Family Pin Descriptions

Name	Type	Description
A31:2	O H(Z) B(Z) R(Z) P(Q)	**ADDRESS BUS** carries the upper 30 bits of the physical address. A31 is the most significant address bit and A2 is the least significant. During a bus access, A31:2 identify all external addresses to word (4-byte) boundaries. The byte enable signals indicate the selected byte in each word. During burst accesses, A3 and A2 increment to indicate succe ssive addresses.
D31:0	I/O S(L) H(Z) B(Z) R(Z) P(1)	**DATA BUS** carries 32, 16, or 8-bit data quantities depending on bus width configuration. The least significant bit of the data is carried on D0 and the most significant on D31. The lower 8 data lines (D7:0) are used when the bus is configured for 8-bit data. When configured for 16-bit data, D15:0 are used.
DP3:0	I/O S(L) H(Z) B(Z) R(Z) P(1)	**DATA PARITY** carries parity information for the data bus. Each parity bit is assigned a group of 8 data bus pins as follows: DP3 generates/checks parity for D31:24 DP2 generates/checks parity for D23:16 DP1 generates/checks parity for D15:8 DP0 generates/checks parity for D7:0 Parity information is generated for a processor write cycle and is checked for a processor read cycle. Parity checking and polarity are programmable. Parity generation/checking is only performed for the size of the data accessed.
$\overline{\text{PCHK}}$	O H(Q) B(Q) R(1) P(1)	**PARITY CHECK** indicates the result of parity check operation. When $\overline{\text{PCHK}}$ is asserted, the previous bus read access resulted in a parity check error.
$\overline{\text{BE3:0}}$	O H(Z) B(Z) R(1) P(1)	**BYTE ENABLES** select which of the four bytes addressed by A31:2 are active during a bus access. Byte enable encoding is dependent on the bus width of the memory region accessed: *32-bit bus:* $\overline{\text{BE3}}$ enables D31:24 $\overline{\text{BE2}}$ enables D23:16 $\overline{\text{BE1}}$ enables D15:8 $\overline{\text{BE0}}$ enables D7:0 *16-bit bus:* $\overline{\text{BE3}}$ becomes Byte High Enable (enables D15:8) $\overline{\text{BE2}}$ is not used (state is high) $\overline{\text{BE1}}$ becomes Address Bit 1 (A1) $\overline{\text{BE0}}$ becomes Byte Low Enable (enables D7:0) *8-bit bus:* $\overline{\text{BE3}}$ is not used (state is high) $\overline{\text{BE2}}$ is not used (state is high) $\overline{\text{BE1}}$ Address Bit 1 (A1) $\overline{\text{BE0}}$ Address Bit 0 (A0)

Table 4. 80960Hx Processor Family Pin Descriptions (Continued)

Name	Type	Description
W/$\overline{R}$	**0** H(Z) B(Z) R(0) P(0)	**WRITE/READ** is low for read accesses and high for write accesses. W/$\overline{R}$ becomes valid during the address phase of a bus cycle and remains valid until the end of the cycle for non-pipelined accesses. For pipelined accesses, W/$\overline{R}$ changes state when the next address is presented.
D/$\overline{C}$	O H(Z) B(Z) R(0) P(0)	**DATA/CODE** indicates that a bus access is a data access (1) or an instruction access(0). D/$\overline{C}$ has the same timing as W/$\overline{R}$.
$\overline{SUP}$	**O** H(Z) B(Z) R(1) P(Q)	**SUPERVISOR ACCESS** indicates whether the current bus access originates from a request issued while in supervisor mode (0) or user mode (1). SUP can be used by the memory subsystem to isolate supervisor code and data structures from non-supervisor access.
$\overline{ADS}$	**O** H(Z) B(Z) R(1) P(1)	**ADDRESS STROBE** indicates a valid address and the start of a new bus access. $\overline{ADS}$ is asserted for the first clock of a bus access.
$\overline{READY}$	**I** S(L)	**$\overline{READY}$,** when enabled for a memory region, is asserted by the memory subsystem to indicate the completion of a data transfer. $\overline{READY}$ is used to indicate that read data on the bus is valid, or that a write transfer has completed. $\overline{READY}$ works in conjunction with the internal wait state generator to accommodate various memory speeds. $\overline{READY}$ is sampled af ter any programmed wait states: { •during each data cycle of a burst access •during the data cycle of a non-burst access
$\overline{BTERM}$	**I** S(L)	**BURST TERMINATE,** when enabled for a memory region, is asserted by the memory subsystem to terminate a burst access in progress. When $\overline{BTERM}$ is asserted, the current burst access is terminated and another address cycle occurs.
$\overline{WAIT}$	**O** H(Z) B(Z) R(1) P(1)	**WAIT** indicates the status of the internal wait-state generator. $\overline{WAIT}$ is asserted when the internal wait state generator generates N_{WAD}, N_{RAD}, N_{WDD} and N_{RDD} wait states. $\overline{WAIT}$ can be used to derive a write data strobe.
$\overline{BLAST}$	**O** H(Z) B(Z) R(1) P(1)	**BURST LAST** indicates the last transfer in a bus access. $\overline{BLAST}$ is asserted in the last data transfer of burst and non-burst accesses after the internal wait-state generator reaches zero. $\overline{BLAST}$ remains active as long as wait states are inserted via the $\overline{READY}$ pin. $\overline{BLAST}$ becomes inactive after the final data transfer in a bus cycle.

1

Table 4. 80960Hx Processor Family Pin Descriptions (Continued)

Name	Type	Description
DT/$\overline{R}$	O H(Z) B(Z) R(0) P(0)	**DATA TRANSMIT/RECEIVE** indicates direction for data transceivers. DT/$\overline{R}$ is used with $\overline{DEN}$ to provide control for data transceivers connected to the data bus. DT/$\overline{R}$ is driven low to indicate the processor expects data (a read cycle). DT/$\overline{R}$ is driven high when the processor is "transmitting" data (a store cycle). DT/$\overline{R}$ only changes state when $\overline{DEN}$ is high.
$\overline{DEN}$	O H(Z) B(Z) R(1) P(1)	**DATA ENABLE** indicates data transfer cycles during a bus access. $\overline{DEN}$ is asserted at the start of the first data cycle in a bus access and de-asserted at the end of the last data cycle. $\overline{DEN}$ is used with DT/$\overline{R}$ to provide control for data transceivers connected to the data bus. $\overline{DEN}$ remains asserted for sequential reads from pipelined memory regions.
$\overline{LOCK}$	O H(Z) B(Z) R(1) P(1)	**BUS LOCK** indicates that an atomic read-modify-write operation is in progress. $\overline{LOCK}$ may be used by the memory subsystem to prevent external agents from accessing memory which is currently involved in an atomic operation (e.g. a semaphore). $\overline{LOCK}$ is asserted in the first clock of an atomic operation and de-asserted when $\overline{BLAST}$ is deasserted in the las t bus cycle.
HOLD	I S(L)	**HOLD REQUEST** signals that an external agent requests access to the processor's address, data and control buses. When HOLD is asserted, the processor: •Completes the current bus request. •Asserts HOLDA and floats the address, data, and control buses. When HOLD is deasserted, the HOLDA pin is deasserted and the processor reassumes control of the address, data, and control pins.
HOLDA	O H(1) B(Z) R(Q) P(Q)	**HOLD ACKNOWLEDGE** indicates to an external master that the processor has relinquished control of the bus. The processor grants HOLD requests and enters the HOLDA state while the RESET pin is asserted. HOLDA is never granted while LOCK is asserted.
$\overline{BOFF}$	I S(L)	**BUS BACKOFF** forces the processor to immediately relinquish control of the bus on the next clock cycle. When $\overline{READY}$/$\overline{BTERM}$ is enabled and: • When $\overline{BOFF}$ is asserted, the address, data, and control buses are floated on the next clock cycle and the current access is aborted. • When BOFF is deasserted, the processor resumes by regenerating the aborted bus access. See Figure 39, BOFF Functional Timing (pg. 46) for $\overline{BOFF}$ timing requirements.
BREQ	O H(Q) B(Q) R(0) P(0)	**BUS REQUEST** indicates that a bus request is pending in the bus controller, but the processor is not stalled pending the result of the bus operation. BREQ can be used with BSTALL to indicate to a an external bus arbiter the processor's bus ownership requirements.

Table 4. 80960Hx Processor Family Pin Descriptions (Continued)

Name	Type	Description
BSTALL	O H(Q) B(Q) R(0) P(0)	**BUS STALL** indicates that the processor has stalled pending the result of a request in the bus controller. When BSTALL is asserted, the processor must regain ownership of the bus in order to continue processing (i.e. it can no longer execute strictly out of on chip cache memory).
CT3:0	O H(Z) B(Z) R(Z) P(7)	**CYCLE TYPE** indicates the type of bus cycle currently being started or processor state. CT3:0 Encoding: **Cycle Type** — **ADS** — **CT3:0** Program-initiated access using 8-bit bus — 0 — 0000 Program-initiated access using 16-bit bus — 0 — 0001 Program-initiated access using 32-bit bus — 0 — 0010 Event-initiated access using 8-bit bus — 0 — 0100 Event-initiated access using 16-bit bus — 0 — 0101 Event-initiated access using 32-bit bus — 0 — 0110 Reserved — 0 — 0X11 Reserved for future products — 0 — 1XXX Processor not halted, otherwise reserved — 1 — 0X00 Processor not halted, otherwise reserved — 1 — 0X01 Processor not halted, otherwise reserved — 1 — 0X10 Reserved — 1 — 0011 Processor in HALT mode — 1 — 0111 Reserved for future products — 1 — 1XXX
$\overline{\text{XINT7:0}}$	I A(E) A(L)	**EXTERNAL INTERRUPT** pins are used to request interrupt service. These pins can be configured in three modes: *Dedicated Mode:* Each pin is assigned a dedicated interrupt level. Dedicated inputs can be programmed to be level (low or high) or edge (rising or falling) sensitive. *Expanded Mode:* All eight pins act as a vectored interrupt source. The interrupt pins are level sensitive in this mode. *Mixed Mode:* The $\overline{\text{XINT7:5}}$ pins act as dedicated sources and the $\overline{\text{XINT4:0}}$ pins act as the 5 most significant bits of a vectored source. The least significant bits of the vectored source are set to "010" internally.
$\overline{\text{NMI}}$	I A(E)	**NON-MASKABLE INTERRUPT** causes a non-maskable interrupt event to occur. $\overline{\text{NMI}}$ is the highest priority interrupt source. $\overline{\text{NMI}}$ is falling edge triggered.
CLKIN	I	**CLOCK INPUT** provides the time base for the 80960Hx. All internal circuitry is synchronized to CLKIN. All input and output timings are specified relative to CLKIN. For the 80960HD, the 2x internal clock is derived by multiplying the CLKIN frequency by 2. For the 80960HT, the 3x internal clock is derived by multiplying the CLKIN frequency by 3.
$\overline{\text{RESET}}$	I A(L)	**$\overline{\text{RESET}}$** forces the device into reset. $\overline{\text{RESET}}$ causes all external and internal signals to return to their reset state (if defined). The rising edge of $\overline{\text{RESET}}$ starts the processor boot sequence.

Table 4. 80960Hx Processor Family Pin Descriptions (Continued)

Name	Type	Description
STEST	I S(L)	**SELF TEST,** when asserted during the rising edge of $\overline{\text{RESET}}$, causes the processor to execute its built in self-test.
$\overline{\text{FAIL}}$	O H(Q) B(Q) R(0) P(1)	$\overline{\text{FAIL}}$ indicates a failure of the processor's built-in self-test performed during initialization. $\overline{\text{FAIL}}$ is asserted immediately out of reset and toggles during self-test to indicate the status of individual tests. If self-test passes, FAIL is de-asserted and the processor branches to the user's initialization code. Should self-test fail, the FAIL pin is asserted and the processor ceases execution.
$\overline{\text{ONCE}}$	I	On-Circuit Emulation control: the processor samples this pin during reset. If it is asserted low at the end of reset, the processor enters ONCE mode. In ONCE Mode, the processor stops all clocks and floats all output pins except the TDO pin.
TCK	I	**TEST CLOCK** provides the clocking function for IEEE 1149.1 Boundary Scan testing.
TDI	I	**TEST DATA INPUT** is the serial input pin for IEEE 1149.1 Boundary Scan testing.
TDO	O	**TEST DATA OUTPUT** is the serial output pin for IEEE 1149.1 Boundary Scan testing. THIS OUTPUT IS ACTIVE DURING ONCE.
$\overline{\text{TRST}}$	I	**TEST RESET** asynchronously resets the Test Access Port (TAP) controller.
TMS	I	**TEST MODE SELECT** is sampled at the rising edge of TCK. TCK controls the sequence of TAP controller state changes for IEEE 1149.1 Boundary Scan testing.
VCC5	I	**5V Reference Voltage** input is the reference voltage for the 5V-tolerant I/O buffers. This signal should be connected to +5V ±5% for use with inputs which exceed 3.3V. If all inputs are from 3.3V components, this pin should be connected to 3.3V.
VCCPLL	I	**PLL Voltage** is the +3.3 VDC analog input for the PLL.
VOLDET	O	**Voltage Detect** signal allows external system logic to distinguish between a 5V 80960Cx processor and the 3.3V 80960Hx processor. This signal is active low for a 3.3V 80960Hx (it is high impedance for 5V 80960Cx). This pin is available only on the PGA version.

3.2 80960Hx Mechanical Data

3.2.1 80960Hx PGA PINOUT

Figure 2 depicts the complete 80960Hx PGA pinout as viewed from the top side of the component (i.e., pins facing down). Figure 3 shows the complete 80960Hx PGA pinout as viewed from the pin-side of the package (i.e., pins facing up). Table 6 lists the 80960Hx pin names with package location. See Section 4.2, Recommended Connections (pg. 18) for sp ecifications and recommended connections.

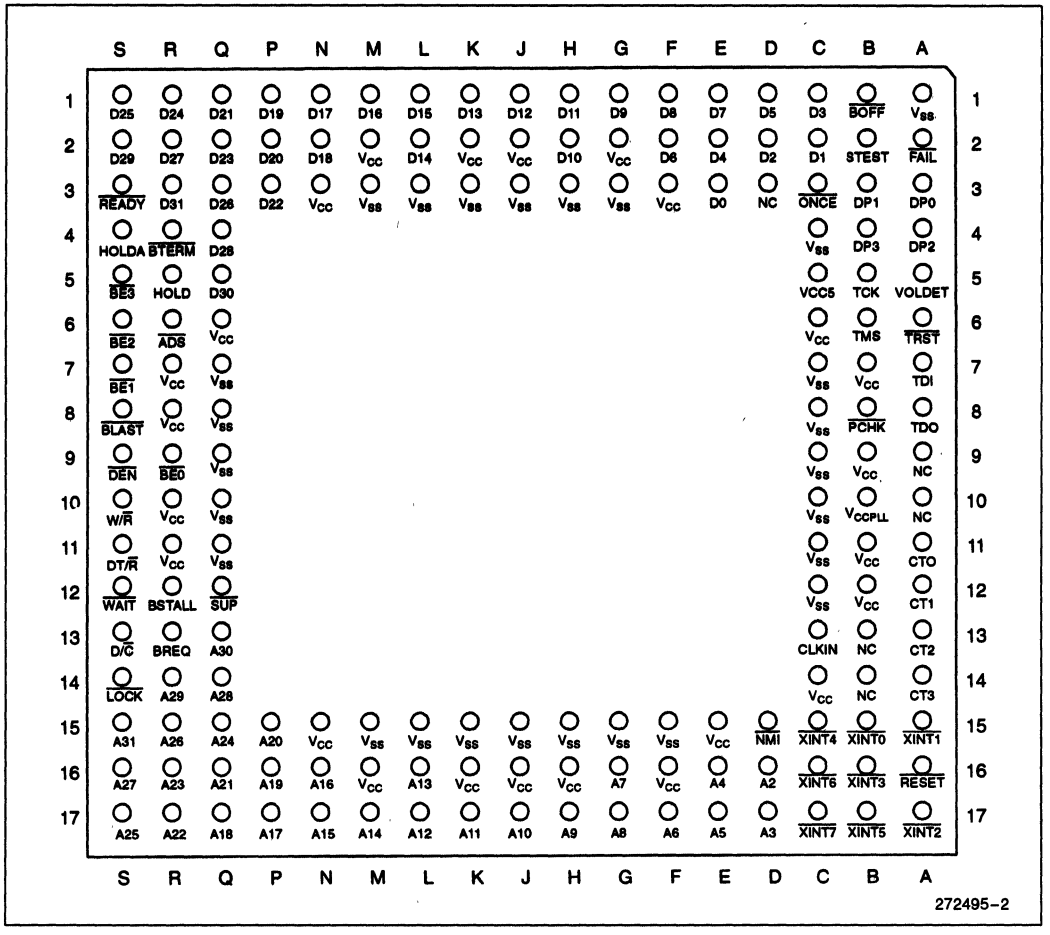


Figure 2. 80960Hx 168-Pin PGA Pinout—View from Top (Pins Facing Down)

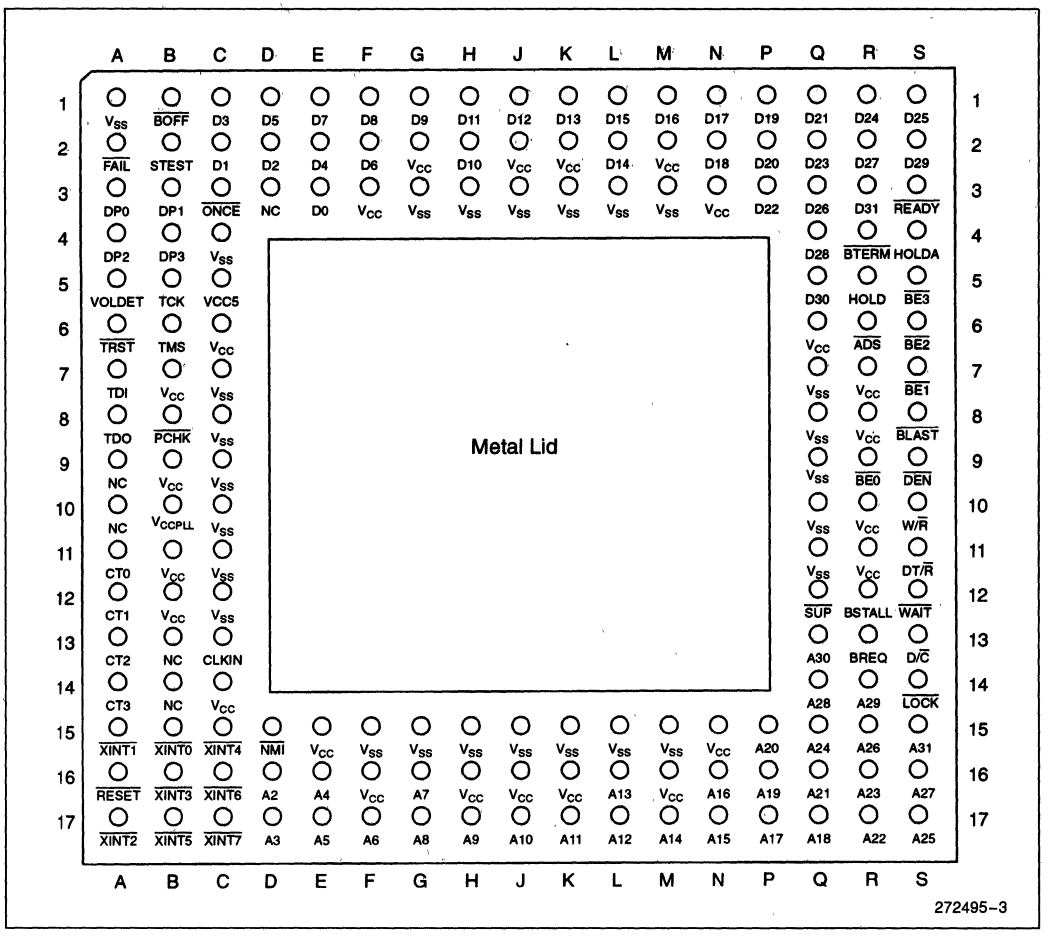


Figure 3. 80960Hx 168-Pin PGA Pinout—View from Bottom (Pins Facing Up)

Table 5. 80960Hx 168-pin PGA Pinout—Pin Number Order

PGA Pin	Signal Name	PGA Pin	Signal Name	PGA Pin	Signal Name	PGA Pin	Signal Name
A1	V_{SS}	C9	V_{SS}	J15	V_{SS}	Q10	V_{SS}
A2	$\overline{FAIL}$	C10	V_{SS}	J16	V_{CC}	Q11	V_{SS}
A3	DP0	C11	V_{SS}	J17	A10	Q12	$\overline{SUP}$
A4	DP2	C12	V_{SS}	K1	D13	Q13	A30
A5	VOLDET	C13	CLKIN	K2	V_{CC}	Q14	A28
A6	TRST	C14	V_{CC}	K3	V_{SS}	Q15	A24
A7	TDI	C15	$\overline{XINT4}$	K15	V_{SS}	Q16	A21
A8	TDO	C16	$\overline{XINT6}$	K16	V_{CC}	Q17	A18
A9	NC	C17	$\overline{XINT7}$	K17	A11	R1	D24
A10	NC	D1	D5	L1	D15	R2	D27
A11	CT0	D2	D2	L2	D14	R3	D31
A12	CT1	D3	NC	L3	V_{SS}	R4	$\overline{BTERM}$
A13	CT2	D15	$\overline{NMI}$	L15	V_{SS}	R5	HOLD
A14	CT3	D16	A2	L16	A13	R6	$\overline{ADS}$
A15	$\overline{XINT1}$	D17	A3	L17	A12	R7	V_{CC}
A16	RESET	E1	D7	M1	D16	R8	V_{CC}
A17	$\overline{XINT2}$	E2	D4	M2	V_{CC}	R9	$\overline{BE0}$
B1	$\overline{BOFF}$	E3	D0	M3	V_{SS}	R10	V_{CC}
B2	STEST	E15	V_{CC}	M15	V_{SS}	R11	V_{CC}
B3	DP1	E16	A4	M16	V_{CC}	R12	BSTALL
B4	DP3	E17	A5	M17	A14	R13	BREQ
B5	TCK	F1	D8	N1	D17	R14	A29
B6	TMS	F2	D6	N2	D18	R15	A26
B7	V_{CC}	F3	V_{CC}	N3	V_{CC}	R16	A23
B8	$\overline{PCHK}$	F15	V_{SS}	N15	V_{CC}	R17	A22
B9	V_{CC}	F16	V_{CC}	N16	A16	S1	D25
B10	VCCPLL	F17	A6	N17	A15	S2	D29
B11	V_{CC}	G1	D9	P1	D19	S3	$\overline{READY}$
B12	V_{CC}	G2	V_{CC}	P2	D20	S4	HOLDA
B13	NC	G3	V_{SS}	P3	D22	S5	$\overline{BE3}$
B14	NC	G15	V_{SS}	P15	A20	S6	$\overline{BE2}$
B15	$\overline{XINT0}$	G16	A7	P16	A19	S7	$\overline{BE1}$
B16	$\overline{XINT3}$	G17	A8	P17	A17	S8	$\overline{BLAST}$
B17	$\overline{XINT5}$	H1	D11	Q1	D21	S9	$\overline{DEN}$
C1	D3	H2	D10	Q2	D23	S10	$\overline{W/R}$
C2	D1	H3	V_{SS}	Q3	D26	S11	DT/R
C3	$\overline{ONCE}$	H15	V_{SS}	Q4	D28	S12	$\overline{WAIT}$
C4	V_{SS}	H16	V_{CC}	Q5	D30	S13	$\overline{D/C}$
C5	VCC5	H17	A9	Q6	V_{CC}	S14	$\overline{LOCK}$
C6	V_{CC}	J1	D12	Q7	V_{SS}	S15	A31
C7	V_{SS}	J2	V_{CC}	Q8	V_{SS}	S16	A27
C8	V_{SS}	J3	V_{SS}	Q9	V_{SS}	S17	A25

1

3.3 Stepping Register Information

The memory mapped register at FF008710 contains the 80960Hx Device ID. The ID is identical to the ID obtained from a JTAG Query. Figure 4 defines the current 80960Hx Device IDs. The value for device identification is compliant with the IEEE 1149.1 specification and Intel standards. Table 6 describes the fields of the device ID.

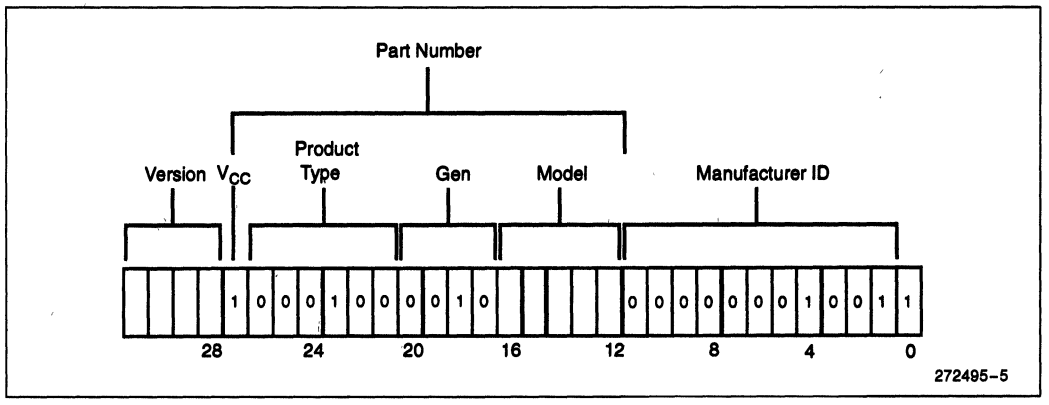


Figure 4. 80960Hx Device Identification Register

Table 6. Fields of 80960Hx Device ID

Field	Value	Definition
Version	0000 = A0 Step	Indicates major stepping changes.
V_{CC}	1 = 3 volt part	Indicates a device is 3V.
Product Type	00 0100 (Indicates i960 CPU)	Designates type of product.
Generation Type	0000 = reserved 0010 = H-series	Indicates the generation (or series) that the product belongs to.
Model	See Table 8	Indicates member within a series and specific model information.
Manufacturer ID	000 0000 1001 (Indicates Intel)	Manufacturer ID assigned by IEEE.

Table 7. HA Device ID Model Types

Device	Version	V_{CC}	Product	Gen.	Model	Manufacturer ID	'1'
80960HA	0000	1	000100	0010	00000	00000001001	1
80960HD	0000	1	000100	0010	00010	00000001001	1
80960HT	0000	1	000100	0010	TBD	00000001001	1

3.4 Sources for Accessories

The following is a list of suggested sources for 80960Hx accessories. This is neither an endorsement or a warranty of the performance of any of the listed products and/or companies.

Sockets

- 3M Textool Test and Interconnection Products
  6801 River Place Blvd. MS 130-3N-29
  Austin, TX 78726-2963
  (800) 328-0411 FAX: (800) 932-9372
- Augat, Inc. Interconnection Products Group
  452 John Dietsch Blvd. P.O. Box 2510
  Attleboro Falls, MA 02763
  (508) 699-7646
- Concept Mfg, Inc. (Decoupling Sockets)
  400 Walnut St. Suite 609
  Redwood City, CA 94063
  (415) 365-1162 FAX: (415) 365-1164

Heatsinks/Fins

- Thermalloy, Inc.
  2021 West Valley View Lane
  Dallas, TX 75234-8993
  (214) 243-4321 FAX: (214) 241-4656
- Wakefield Engineering, Inc.
  60 Audubon Road
  Wakefield, MA 01880
  (617) 245-5900 FAX: (617) 246-0874

4.0 ELECTRICAL SPECIFICATIONS

4.1 Absolute Maximum Ratings

Parameter	Maximum Ratings
Storage Temperature	−65°C to +150°C
Case Temperature Under Bias	−65°C to +110°C
Supply Voltage wrt. V_{SS}	−0.5V to +4.6V
Voltage on VCC5 wrt. V_{SS}	−0.5V to + 6.5V
Voltage on Other Pins wrt. V_{SS}	−0.5V to V_{CC} + 0.5V

NOTICE: This document contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product is available. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

4.2 Recommended Connections

Power and ground connections must be made to multiple V_{CC} and V_{SS} (GND) pins. Every 80960Hx-based circuit board should include power (V_{CC}) and ground (V_{SS}) planes for power distribution. Every V_{CC} pin must be connected to the power plane; every V_{SS} pin must be connected to the ground plane. Pins identified as "NC"—no connect pins—must not be connected in the system.

Liberal decoupling capacitance should be placed near the 80960Hx. The processor can cause transient power surges when its numerous output buffers transition, particularly when connected to large capacitive loads.

Low inductance capacitors and interconnects are recommended for best high frequency electrical performance. Inductance can be reduced by shortening the board traces between the processor and decoupling capacitors as much as possible. Capacitors specifically designed for PGA packages will offer the lowest possible inductance.

For reliable operation, always connect unused inputs to an appropriate signal level. In particular, any unused interrupt ($\overline{XINT}$, $\overline{NMI}$) input should be connected to V_{CC} through a pull-up resistor, as should $\overline{BTERM}$ if not used. Pull-up resistors should be in the in the range of 20 KΩ for each pin tied high. If $\overline{READY}$ or HOLD are not used, the unused input should be connected to ground. **N.C. pins must always remain unconnected.**

4.3 VCC5 Pin Requirements (V_{DIFF})

In mixed voltage systems that drive 80960Hx processor inputs in excess of 3.3V, the V_{CC5} pin must be connected to the system's 5V supply. To limit current flow into the V_{CC5} pin, there is a limit to the voltage differential between the V_{CC5} pin and the other V_{CC} pins. The voltage differential between the 80960Hx V_{CC5} pin and its 3.3V V_{CC} pins should never exceed 2.25V. This limit applies to power up, power down, and steady-state operation. Table 9 outlines this requirement.

Meeting this requirement ensures proper operation and guarantees that the current draw into the V_{CC5} pin does not exceed the I_{CC5} specification.

If the voltage difference requirements cannot be met due to system design limitations, an alternate solution may be employed. As shown in Figure 5, a minimum of 100 ohm series resistor may be used to limit the current into the V_{CC5} pin. This resistor ensures that current drawn by the V_{CC5} pin does not exceed the maximum rating for this pin.

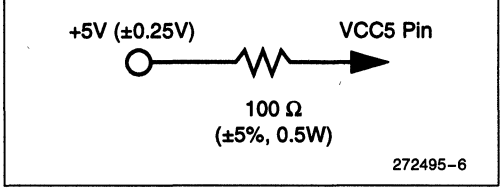


Figure 5. V_{CC5} Current-Limiting Resistor

This resistor is not necessary in systems that can guarantee the V_{DIFF} specification.

In 3.3V-only systems and systems that drive 80960Hx inputs and I/O's from 3.3V logic, connect the V_{CC5} pin directly to the 3.3V V_{CC} plane.

Table 8. V_{DIFF} Specification for Dual Power Supply Requirements (3.3V, 5V)

Symbol	Parameter	Min	Max	Units	Notes
V_{DIFF}	V_{CC5}–V_{CC} Difference		2.25	V	V_{CC5} input should not exceed VCC by more than 2.25V during power-up, power down or during steady-state operation.

4.3.1 AC Test Conditions

AC values are derived using the 50 pF load shown in Figure 6. Figure 7 shows how timings vary with load capacitance. Specifications are measured at the 1.5V crossing point, unless otherwise indicated. Input waveforms (except for CLKIN) are assumed to have a rise and fall time of ≤ 2 ns from 0.8V to 2.0V.

4.3.2 AC Timing Waveforms

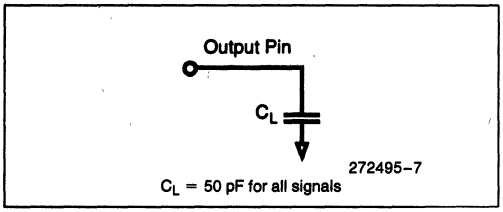


Figure 6. AC Test Load

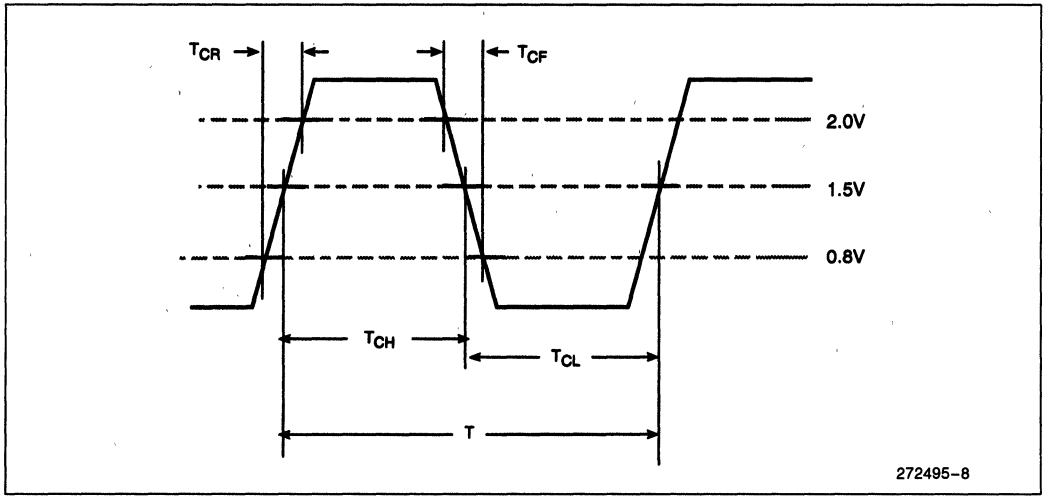


Figure 7. CLKIN Waveform

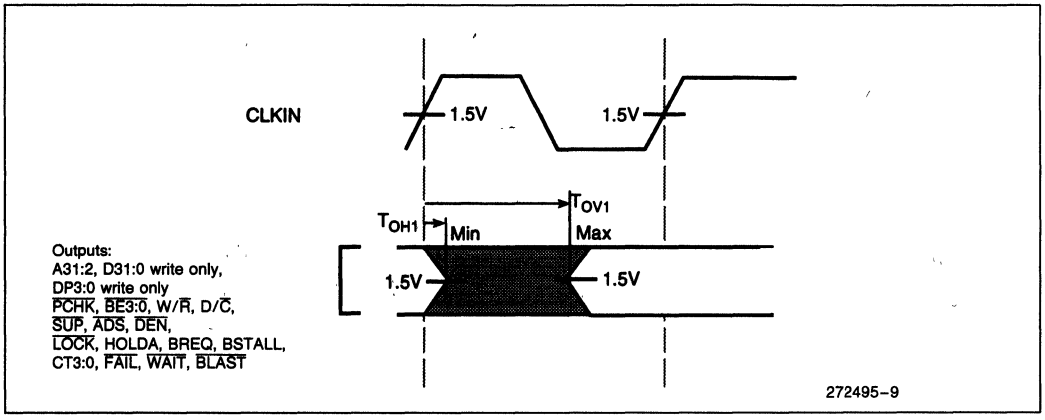


Figure 8. Output Delay Waveform

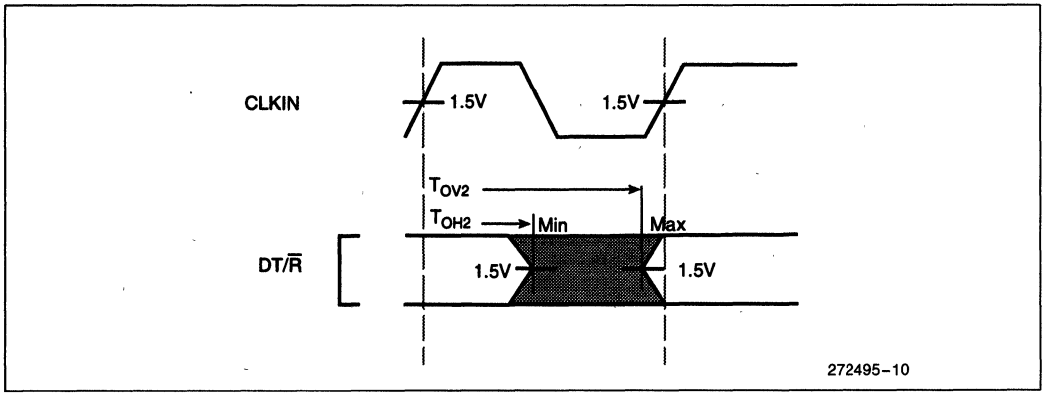


Figure 9. Output Delay Waveform

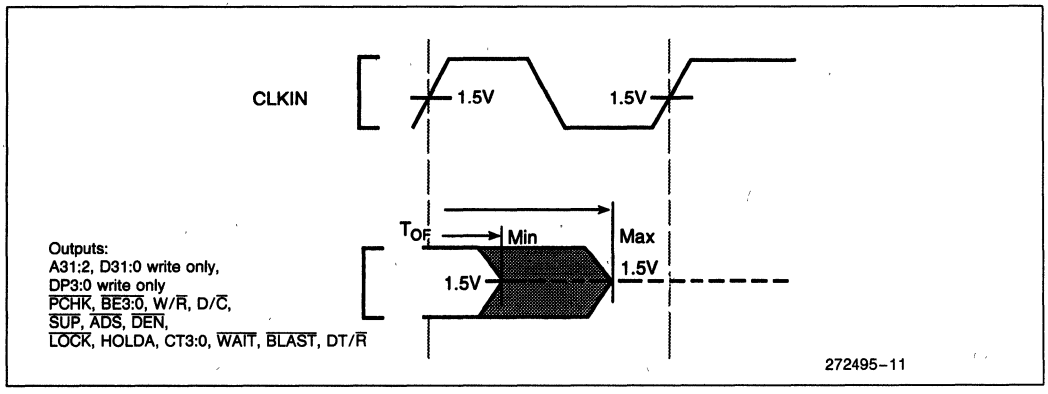


Figure 10. Output Float Waveform

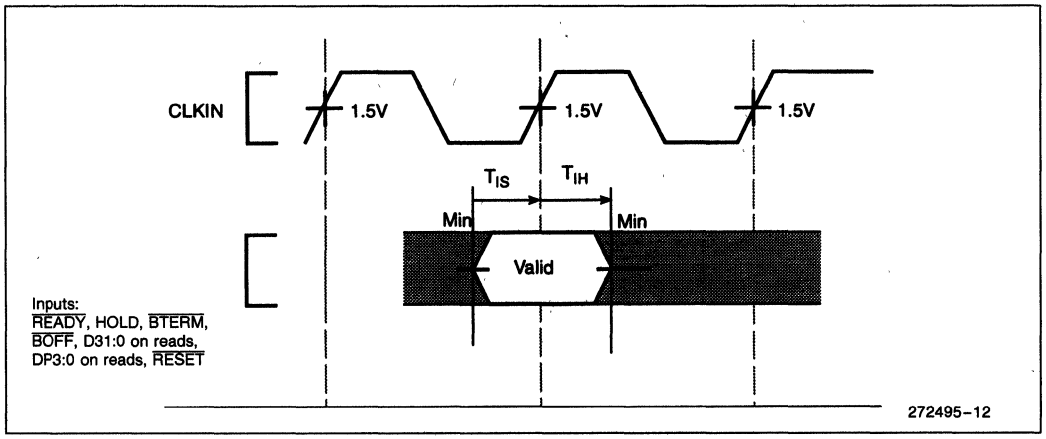
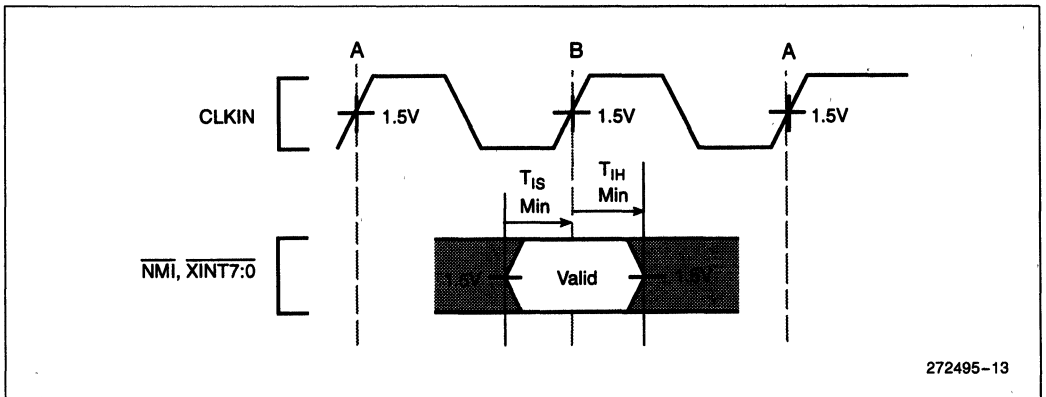


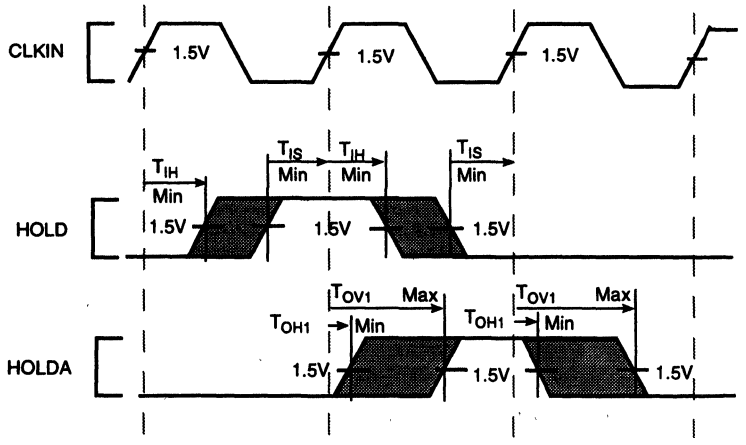
Figure 11. Input Setup and Hold Waveform

T_{OV1} T_{OH1}	OUTPUT DELAY—The maximum output delay is referred to as the Output Valid Delay (T_{OV1}). The minimum output delay is referred to as the Output Hold (T_{OH1}).
T_{OF}	OUTPUT FLOAT DELAY—The output float condition cocurs when the maximum output current becomes less than I_{LO} in magnitude.
T_{IS} T_{IH}	INPUT SETUP AND HOLD—The input setup and hold requirements specify the sampling window during which synchronous inputs must be stable for correct processor operation.



NOTE:
A and B edges are established by de-assertion of RESET. See Figure 19, Cold Reset Waveform (pg. 26).

Figure 12. NMI, XINT7:0 Input Setup and Hold Waveform



T_{OV1} T_{OH1} OUTPUT DELAY—The maximum output delay is referred to as the Output Valid Delay (T_{OV1}). The minimum output delay is referred to as the Output Hold (T_{OH1})

T_{IS} T_{IH} INPUT SETUP AND HOLD—The input setup and hold requirements specify the sampling window during which synchronous inputs must be stable for correct processor operation.

Figure 13. Hold Acknowledge Timings

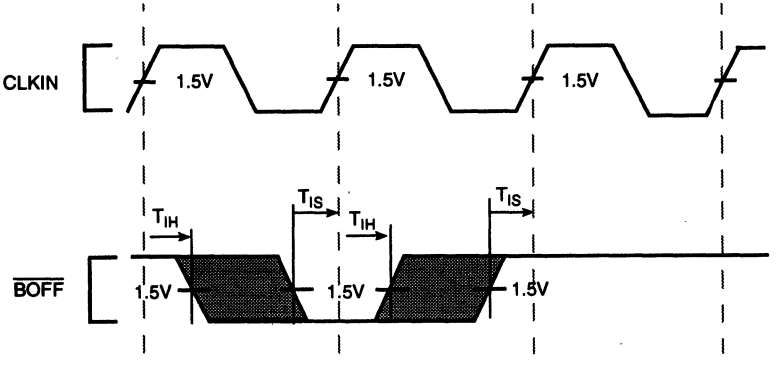


Figure 14. Bus Backoff ($\overline{BOFF}$) Timings

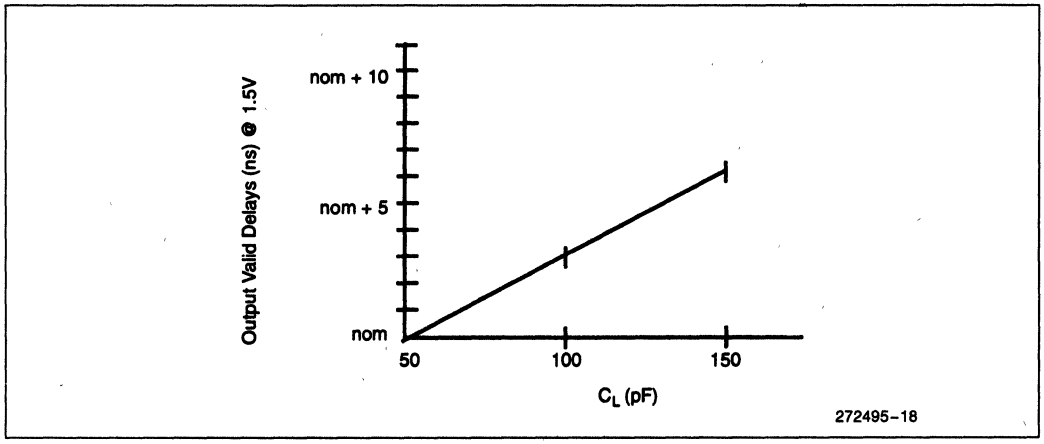
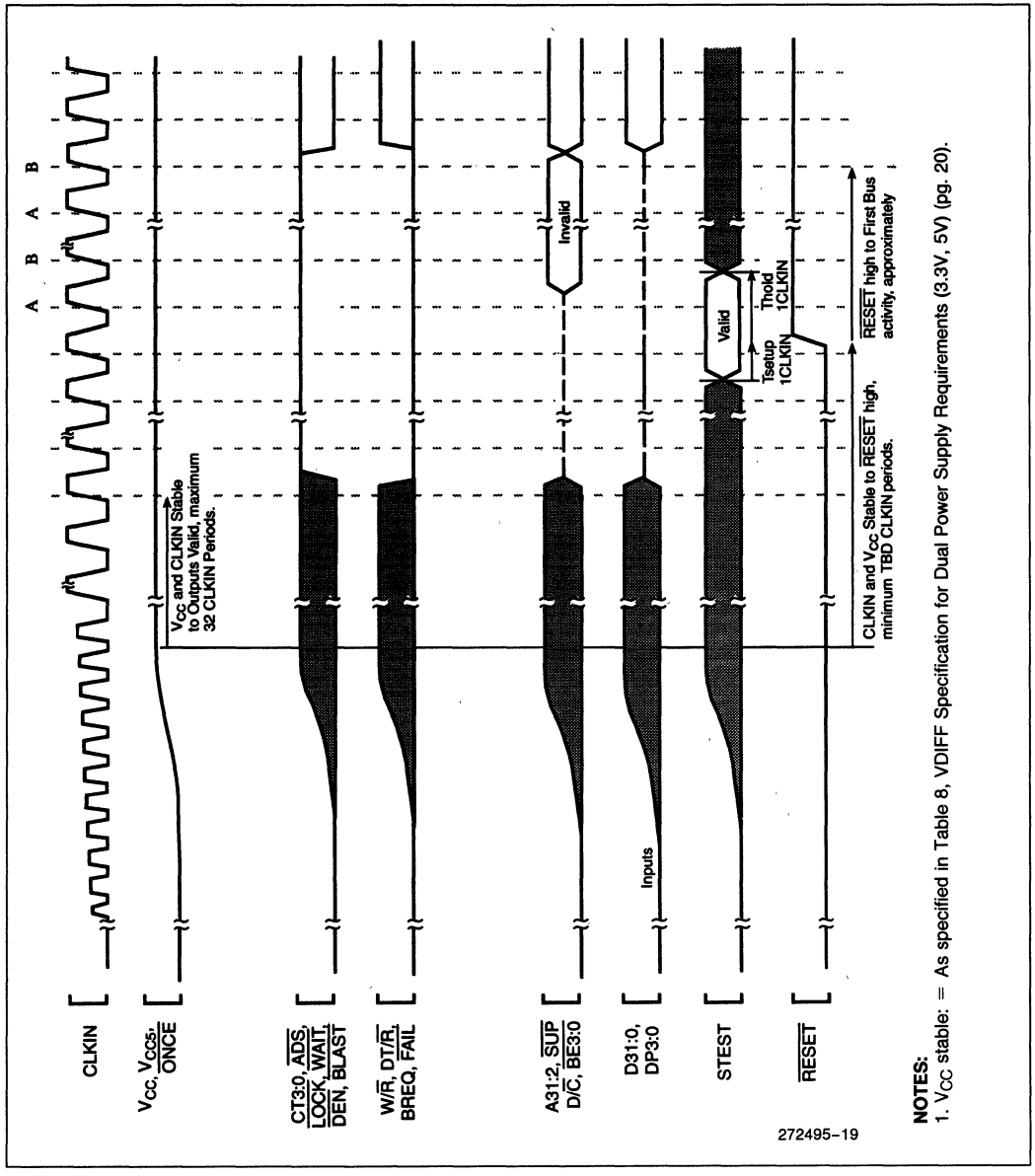


Figure 15. Output Delay or Hold vs. Load Capacitance

5.0 BUS WAVEFORMS



NOTES:
1. V_{CC} stable: = As specified in Table 8, VDIFF Specification for Dual Power Supply Requirements (3.3V, 5V) (pg. 20).

Figure 16. Cold Reset Waveform

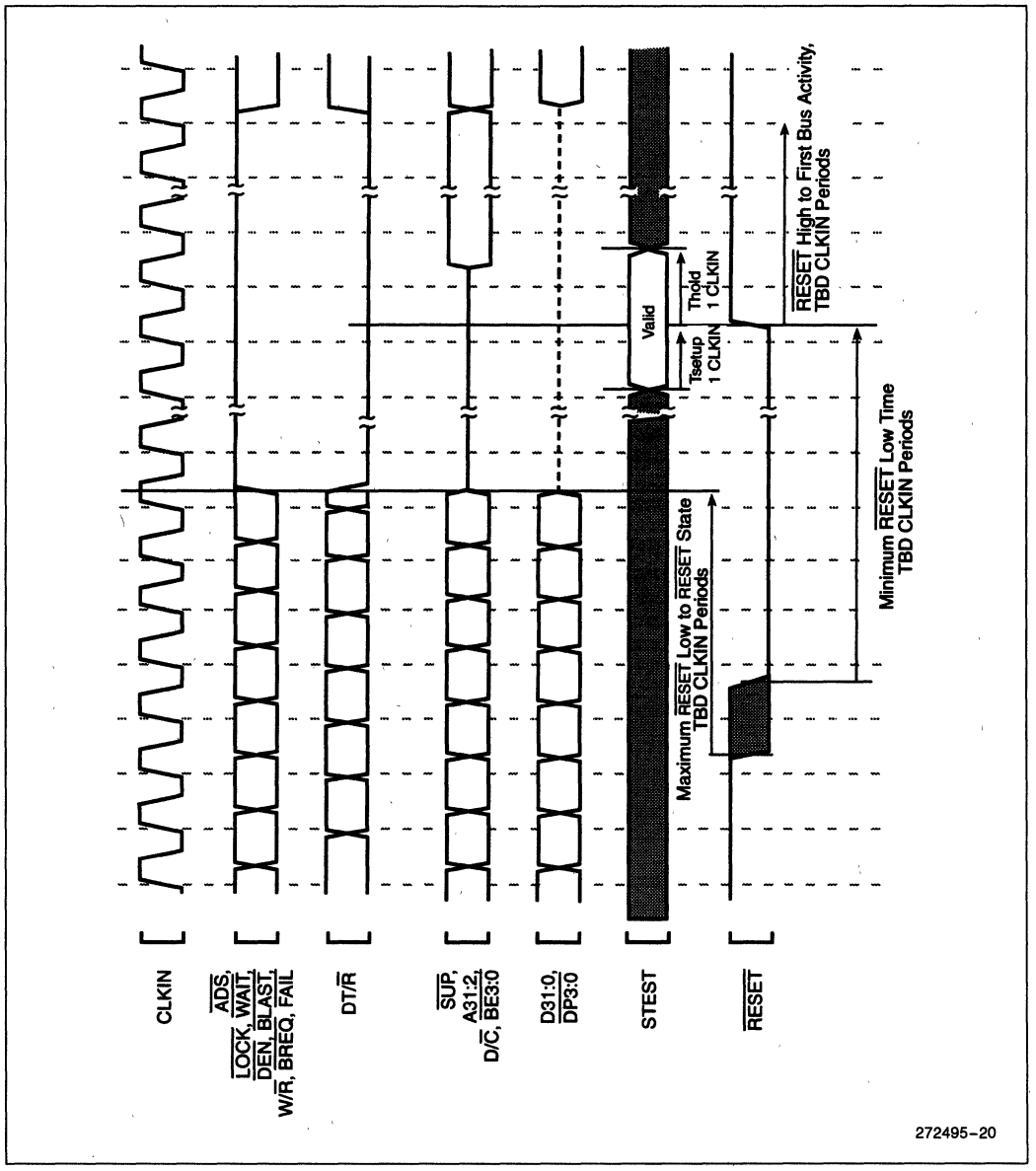


Figure 17. Warm Reset Waveform

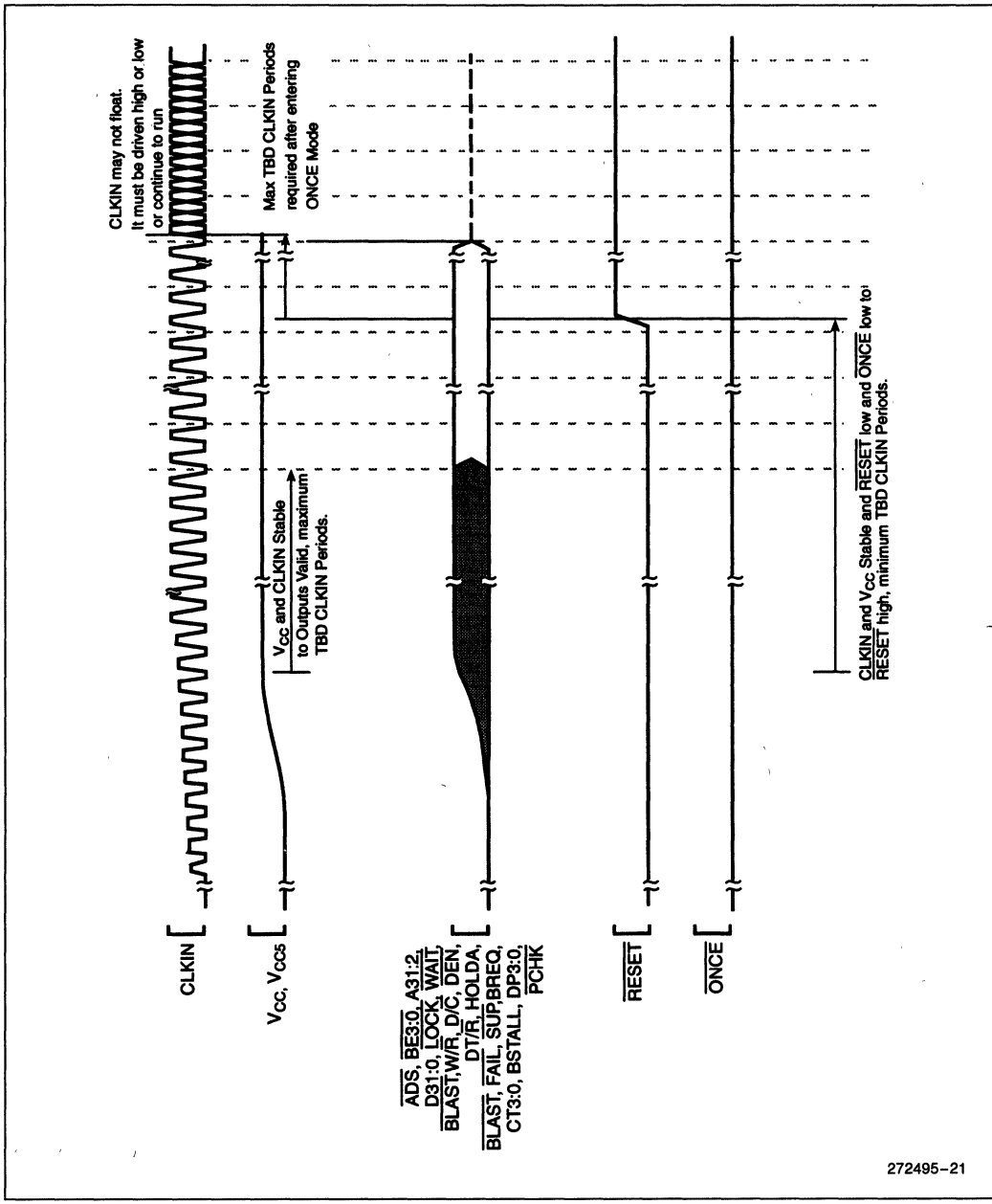


Figure 18. Entering the ONCE State

PMCON Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Disabled 0	OFF 0	X xx	X x	Enabled 1	0 0000	0 00	0 00000	0 00	0 00000

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

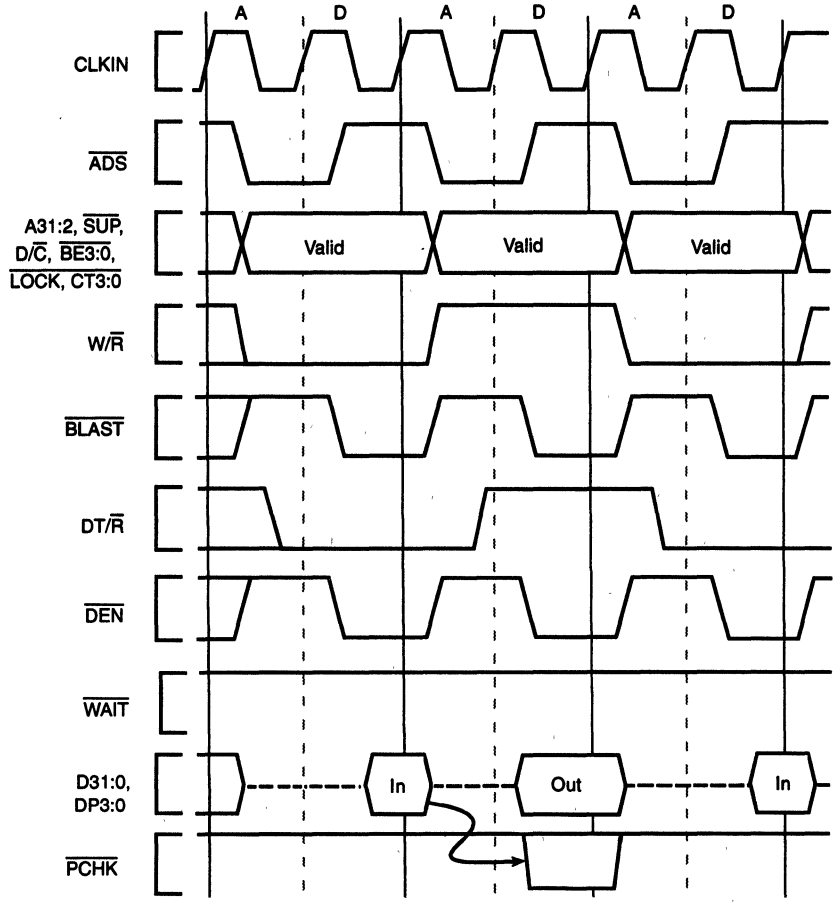


Figure 19. Non-Burst, Non-Pipelined Requests Without Wait States

PMCON	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Disabled 0	OFF 0	X xx	X x	Enabled 1	1 0001	X xx	X xxxxx	X xx	3 00011

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

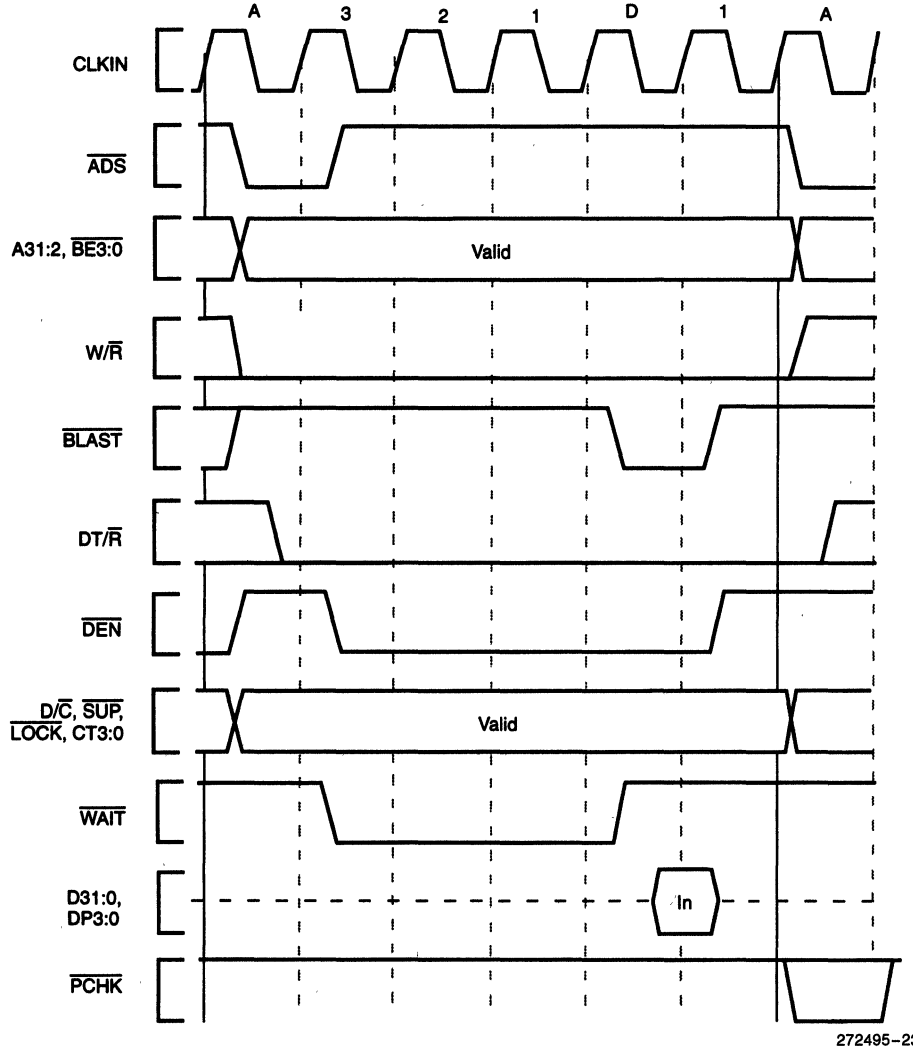


Figure 20. Non-Burst, Non-Pipelined Read Request With Wait States

PMCON

Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Disabled 0	OFF 0	X xx	X x	Enabled 1	1 0001	X xxxxx	3 00011	X xx	X xxxxx

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

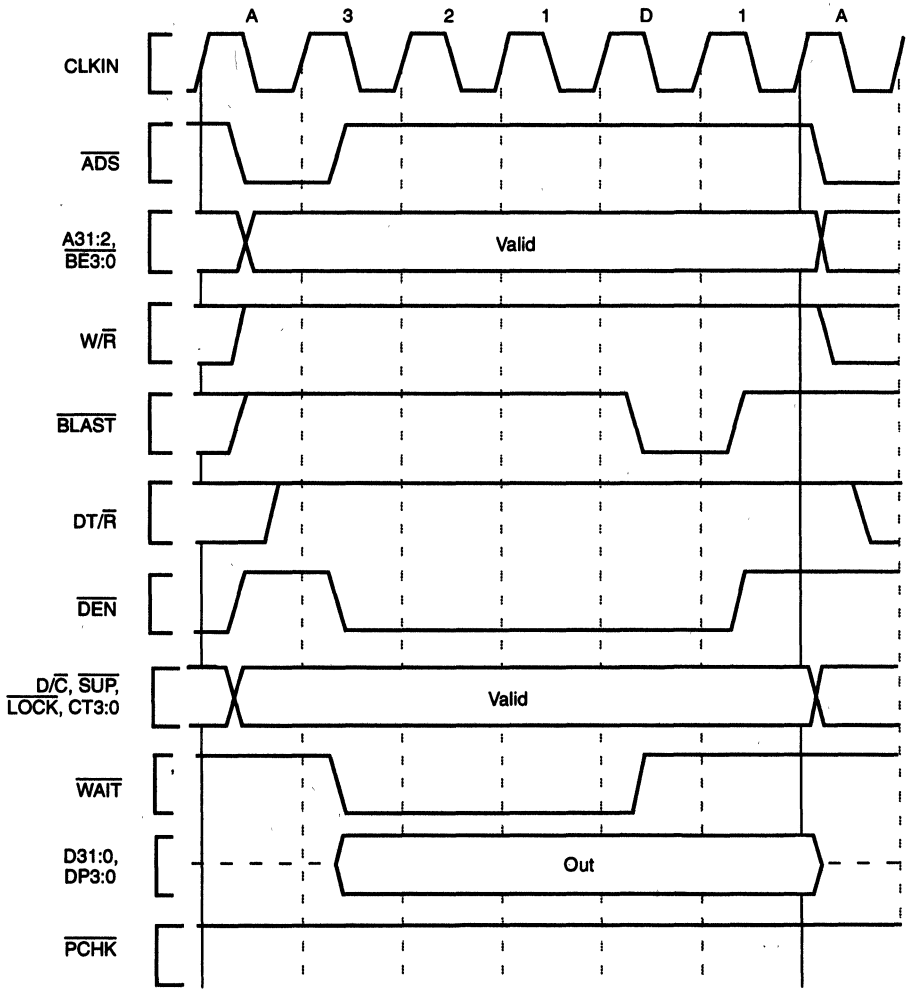


Figure 21. Non-Burst, Non-Pipelined Write Request With Wait States

PMCON

Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Enabled 1	OFF 0	32-Bit 10	X x	Enabled 1	0 0000	X xx	X xxxxx	0 00	0 00000

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

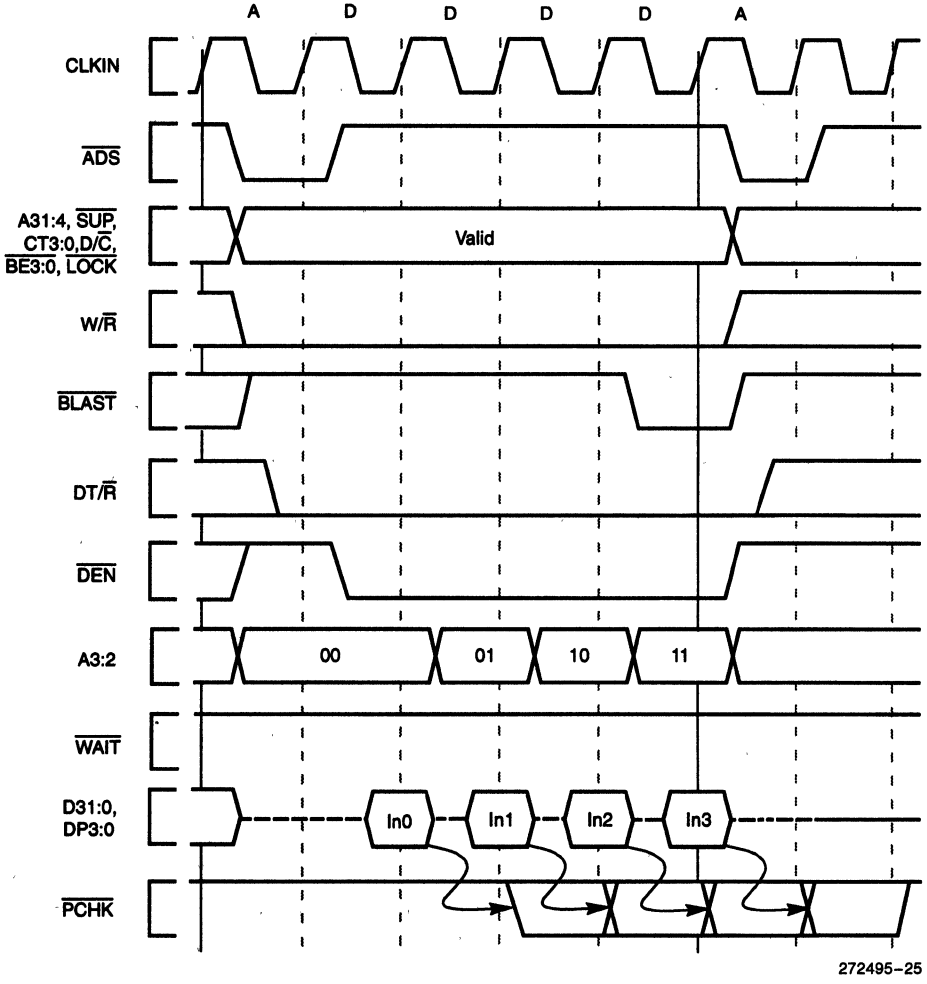


Figure 22. Burst, Non-Pipelined Read Request Without Wait States, 32-Bit Bus

PMCON

Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Enabled 1	OFF 0	32-Bit 10	X x	Enabled 1	1 0001	X xx	X xxxxx	1 01	2 00010

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

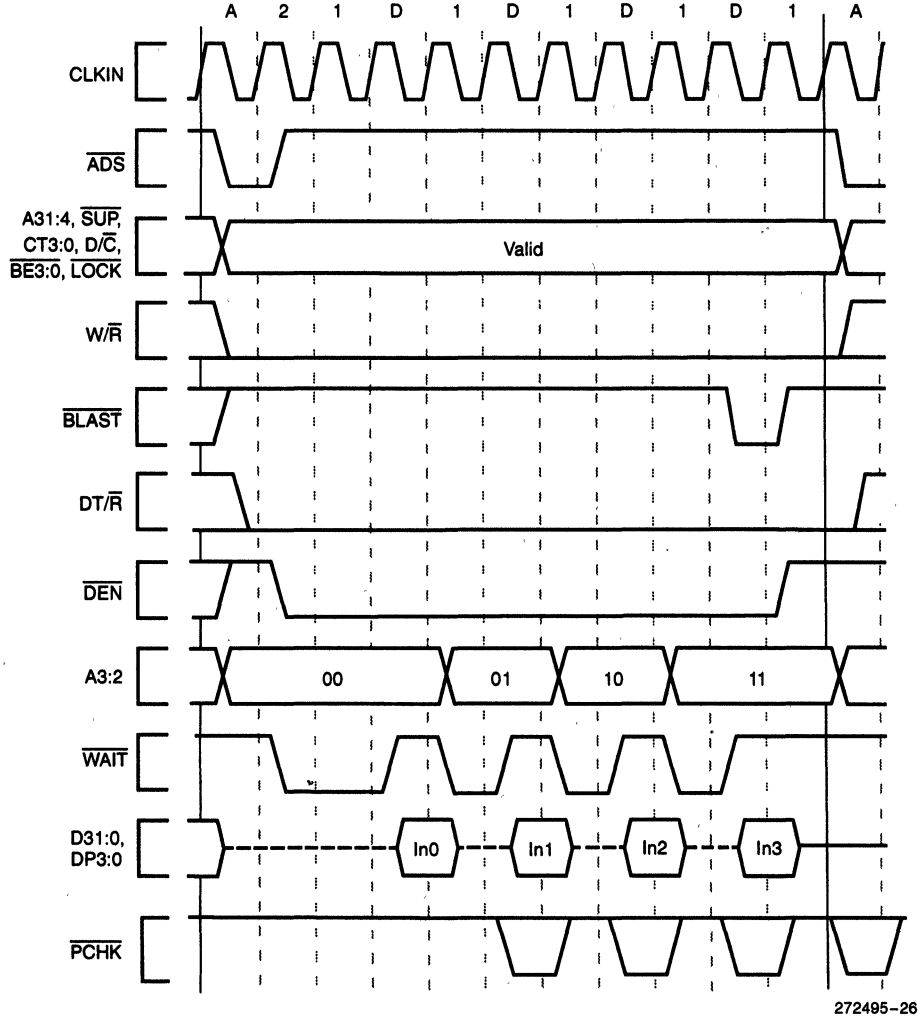


Figure 23. Burst, Non-Pipelined Read Request With Wait States, 32-Bit Bus

PMCON Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Enabled 1	OFF 0	32-Bit 10	X x	Enabled 1	0 0000	0 00	0 00000	X xx	X xxxxx

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

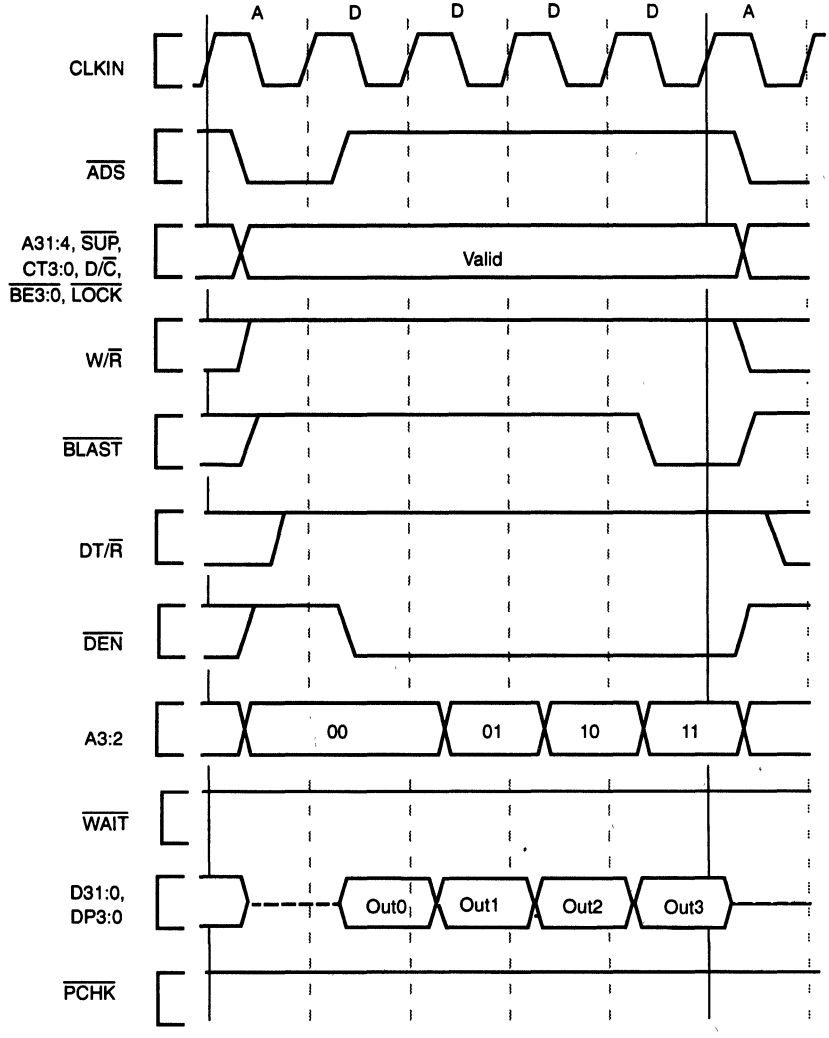


Figure 24. Burst, Non-Pipelined Write Request Without Wait States, 32-Bit Bus

PMCON

Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Enabled 1	OFF 0	32-bit 10	X x	Enabled 1	1 0001	1 01	2 00010	X xx	X xxxxx

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

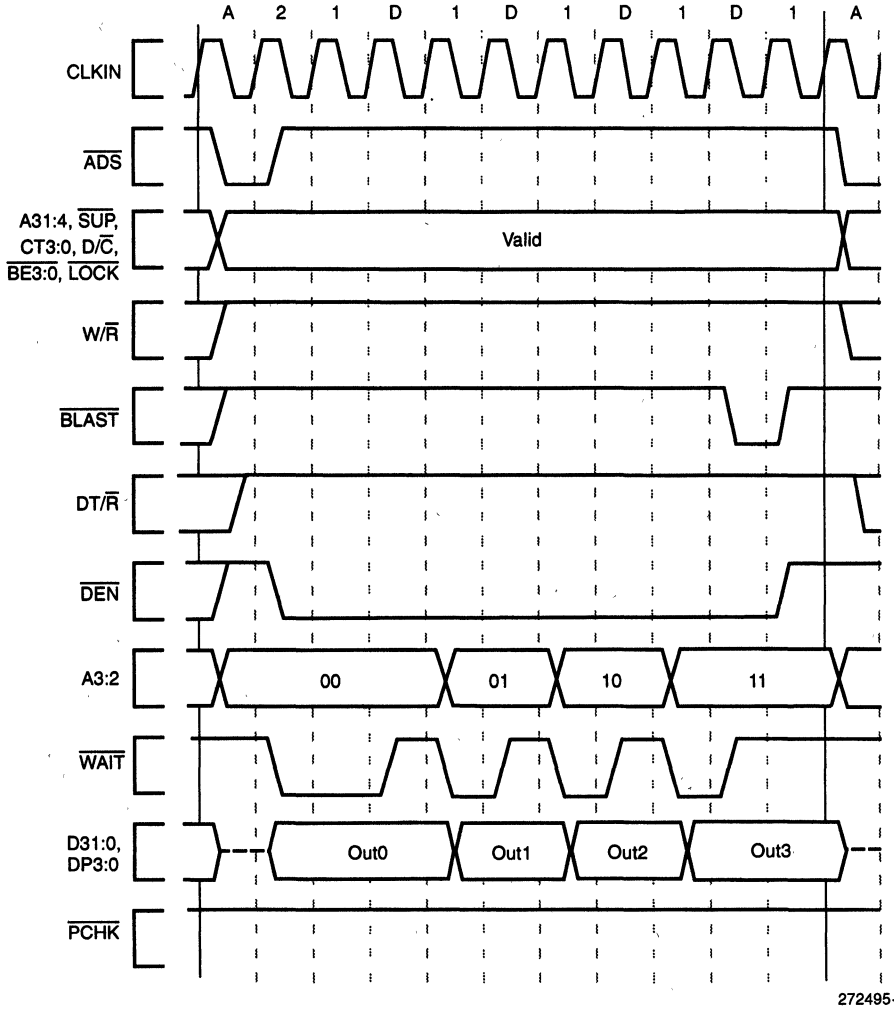


Figure 25. Burst, Non-Pipelined Write Request With Wait States, 32-Bit Bus

PMCON Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Enabled 1	OFF 0	16-Bit 01	X x	Enabled 1	1 0001	X xx	X xxxxx	1 01	2 00010

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

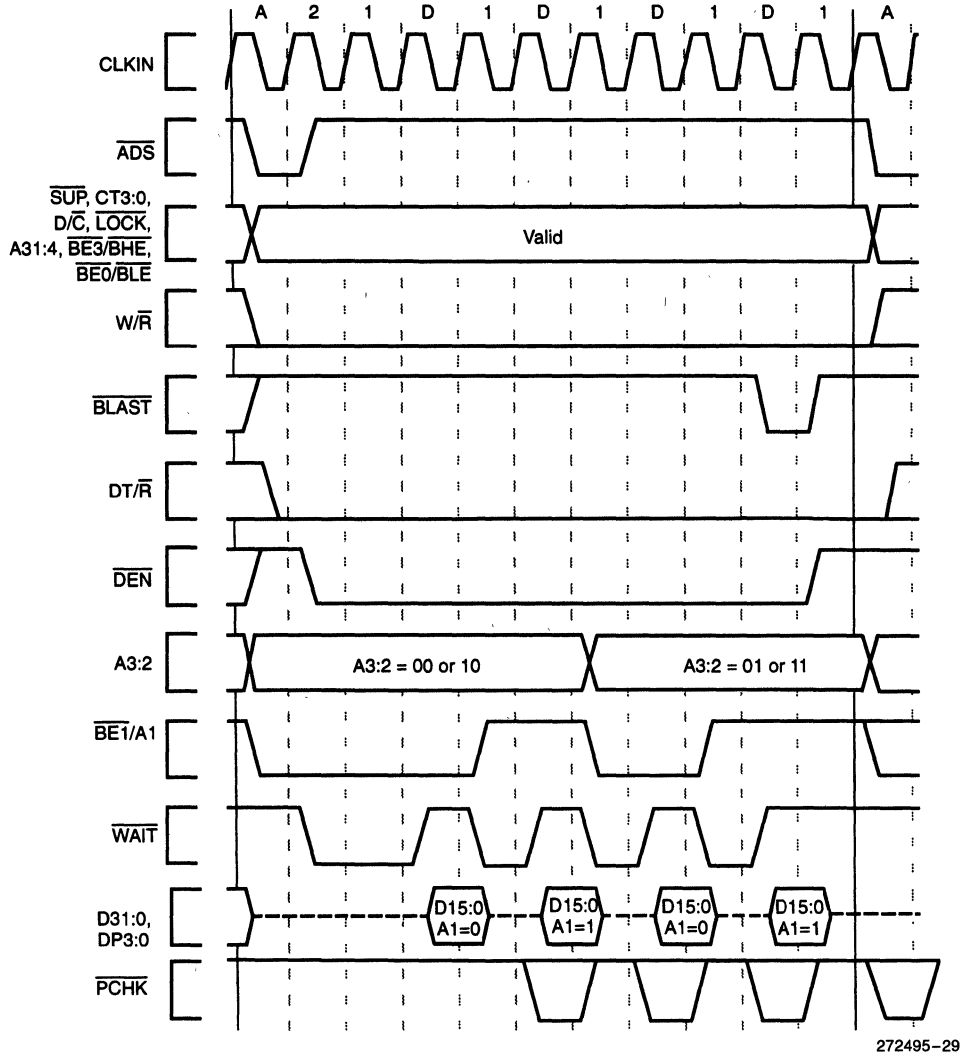


Figure 26. Burst, Non-Pipelined Read Request With Wait States, 16-Bit Bus

PMCON

Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Enabled 1	OFF 0	8-Bit 00	X x	Enabled 1	1 0001	X xx	X xxxxx	1 01	2 00010

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

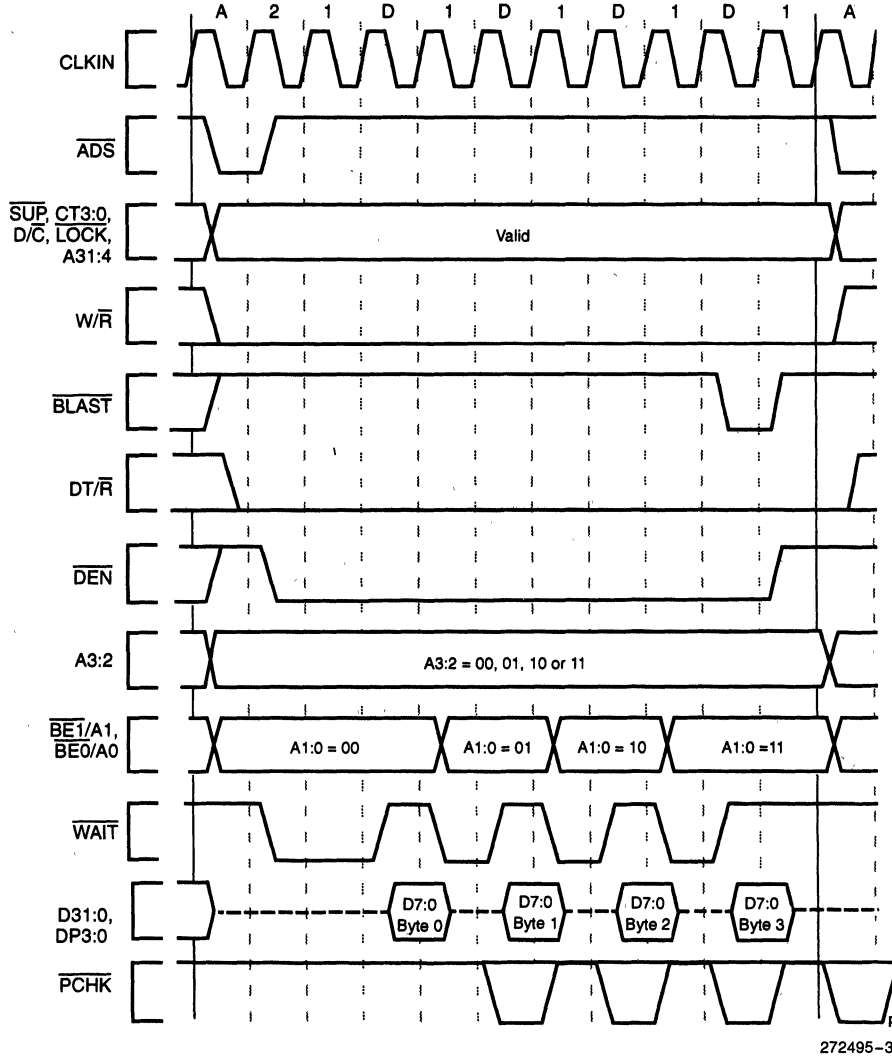
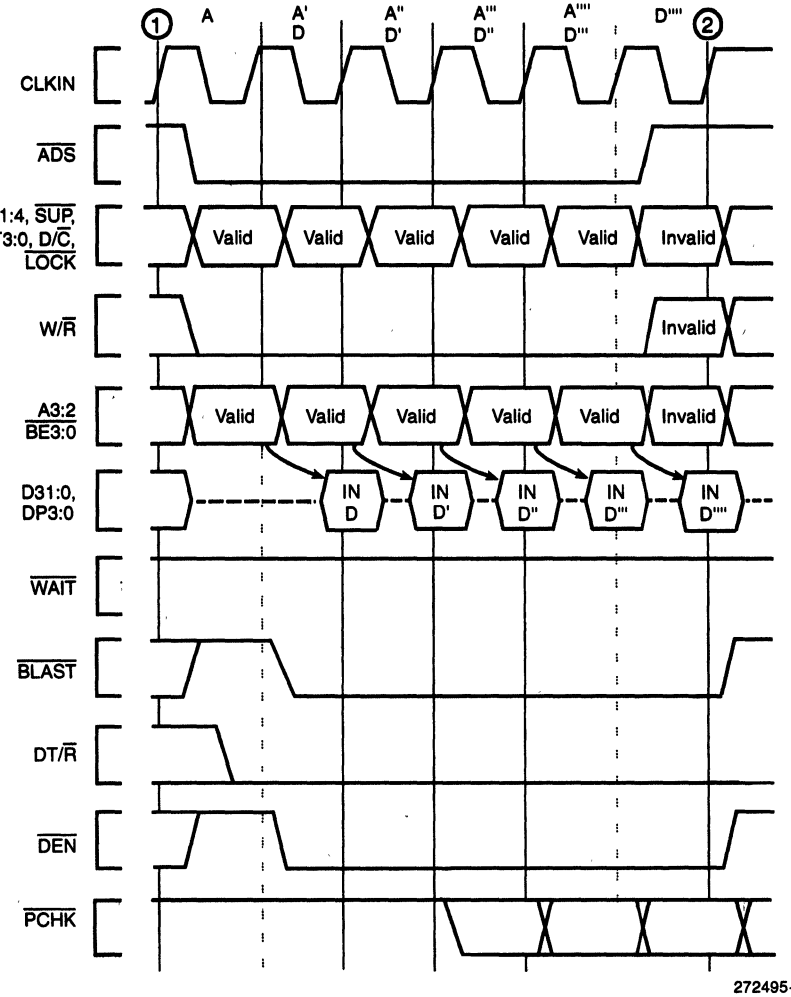


Figure 27. Burst, Non-Pipelined Read Request With Wait States, 8-Bit Bus

PMCON Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	X x	Disabled 0	ON 1	32-Bit 10	X x	Enabled 1	X xxxx	X xx	X xxxxx	X xx	0 00000

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.



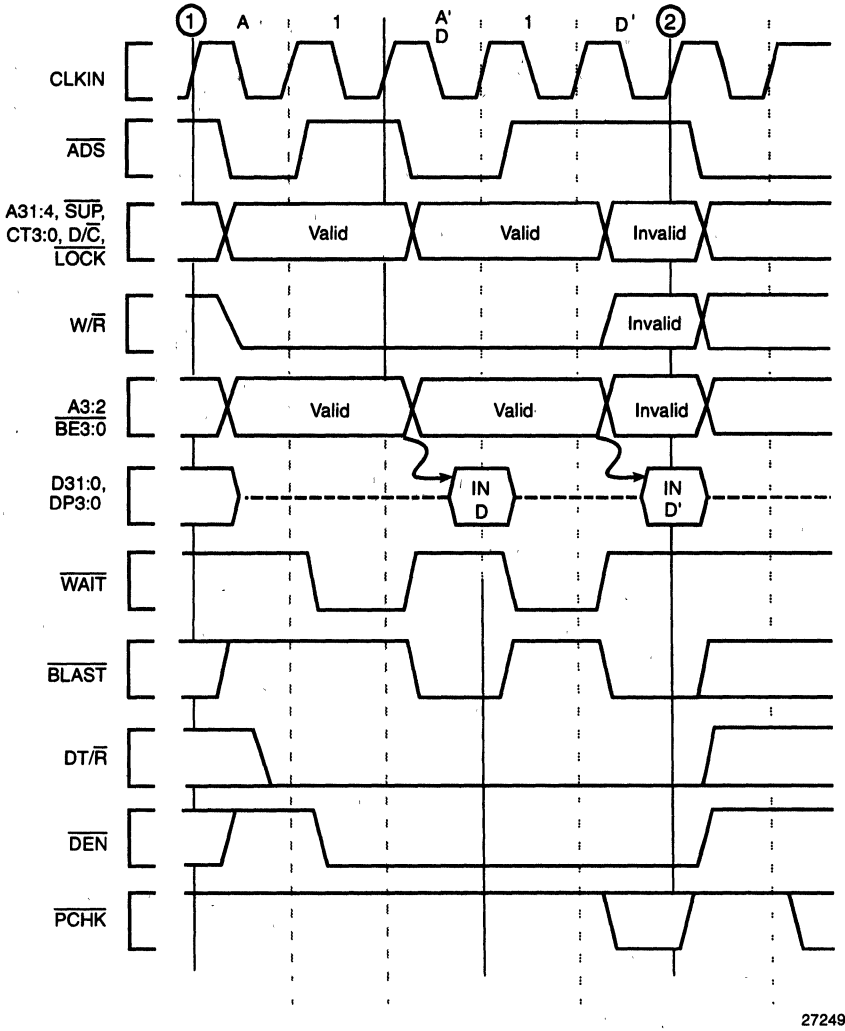
1. Non-pipelined request concludes, pipelined reads begin.
1. Pipelined reads conclude, non-pipelined requests begin.

Figure 28. Non-Burst, Pipelined Read Request Without Wait States, 32-Bit Bus

PMCON

Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	X x	Disabled 0	ON 1	32-Bit 10	X x	Enabled 1	X xxxx	X xx	X xxxxx	X xx	1 00001

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

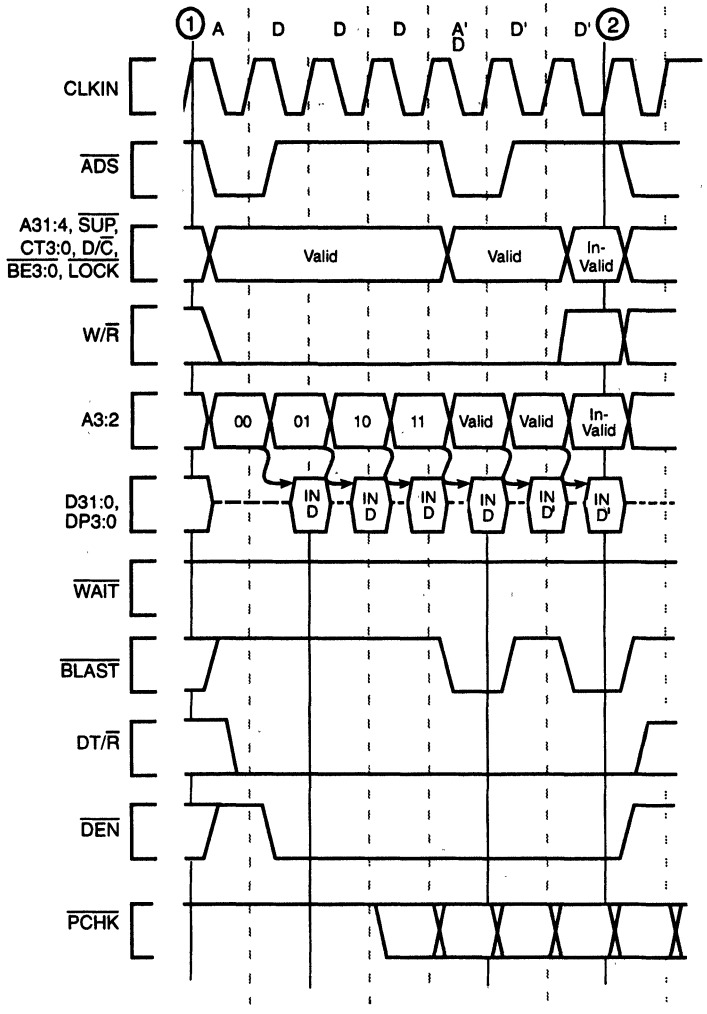


1. Non-pipelined request concludes, pipelined reads begin
2. Pipelined reads conclude, non-pipelined requests begin

Figure 29. Non-Burst, Pipelined Read Request With Wait States, 32-Bit Bus

PMCON Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	X	Enabled 1	ON 1	32-Bit 10	X	Enabled 1	X XXXX	X XX	X XXXXX	0 00	0 00000

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.



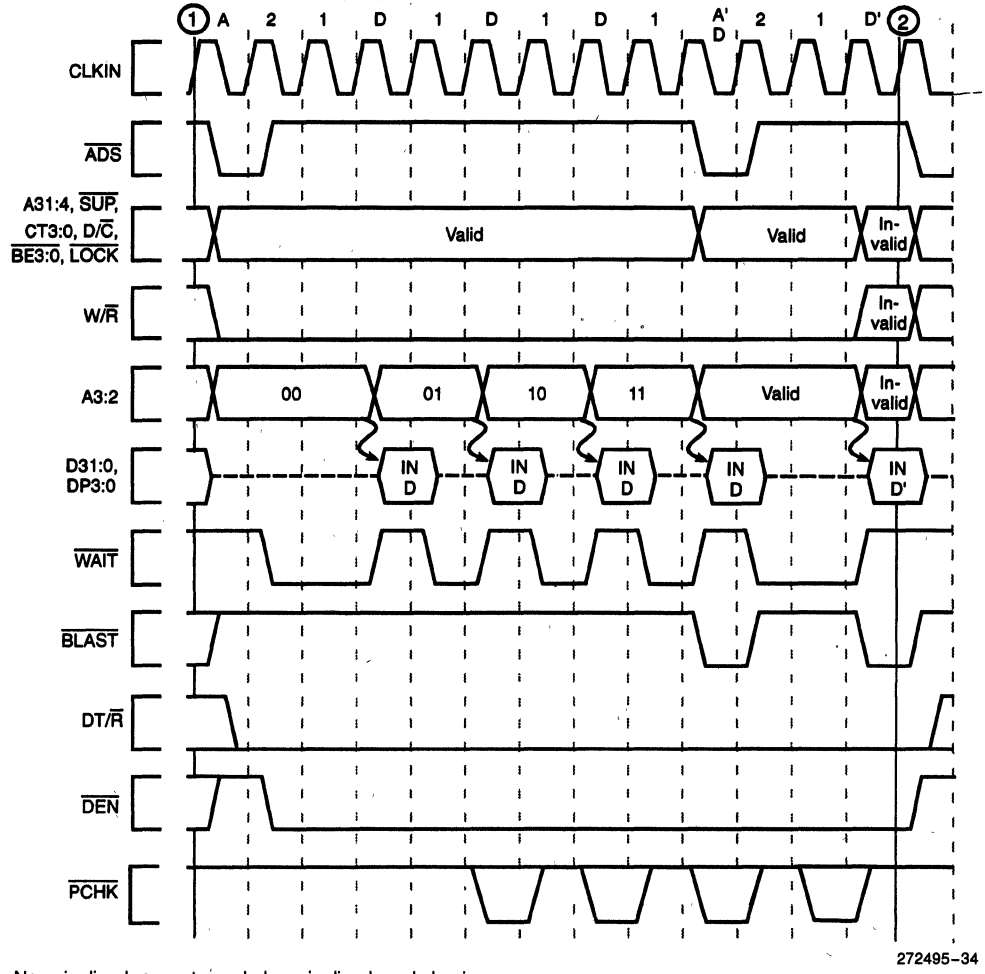
272495-33

1. Non-pipelined request concludes, pipelined reads begin.
2. Pipelined reads conclude, non-pipelined requests begin.

Figure 30. Burst, Pipelined Read Request Without Wait States, 32-Bit Bus

PMCON Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	X x	Enabled 1	ON 1	32-Bit 10	X x	Enabled 1	X xxxx	X xx	X xxxxx	1 01	2 00010

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

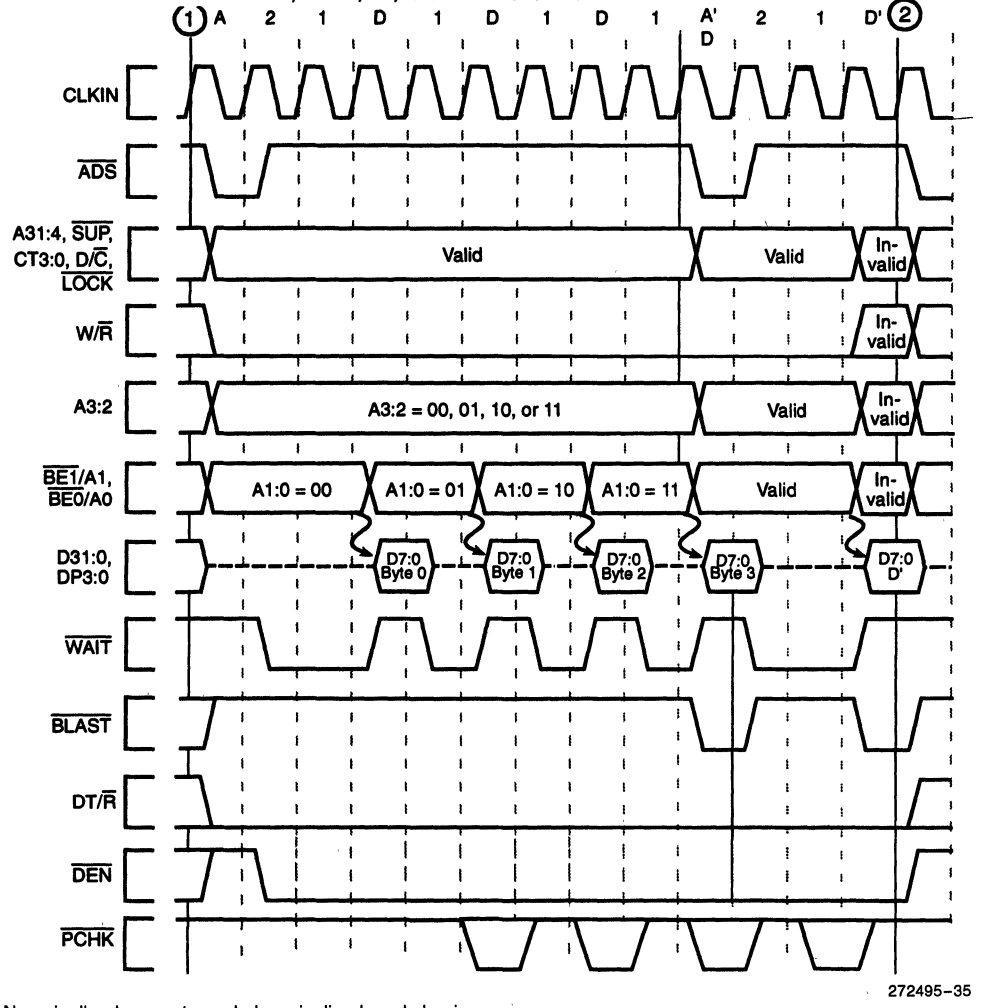


1. Non-pipelined request concludes, pipelined reads begin.
1. Pipelined reads conclude, non-pipelined requests begin.

Figure 31. Burst, Pipelined Read Request With Wait States, 32-Bit Bus

PMCON Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	X x	Enabled 1	ON 1	8-Bit 00	X x	Enabled 1	X xxxx	X xx	X xxxxx	1 01	2 00010

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

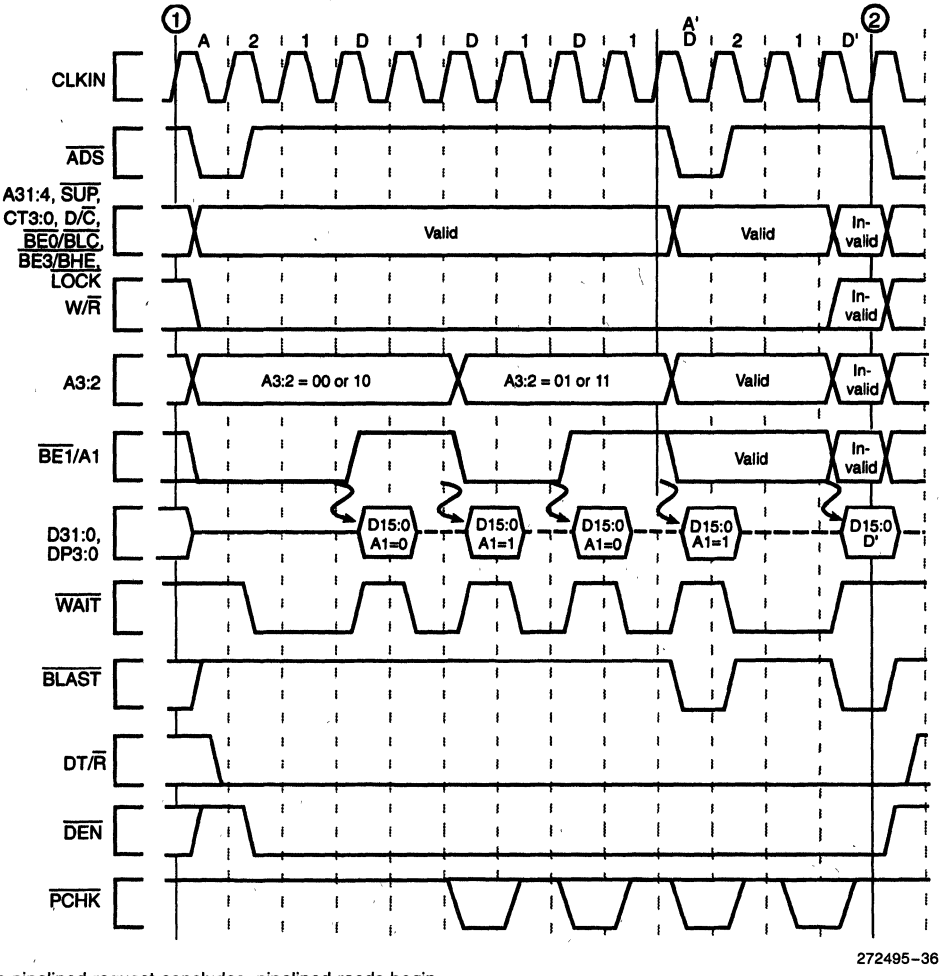


1. Non-pipelined request concludes, pipelined reads begin
2. Pipelined reads conclude, non-pipelined requests begin

Figure 32. Burst, Pipelined Read Request With Wait States, 8-Bit Bus

PMCON Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	X x	Enabled 1	ON 1	16-Bit 01	X x	Enabled 1	X xxxx	X xx	X xxxxx	1 01	2 00010

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.



1. Non-pipelined request concludes, pipelined reads begin
2. Pipelined reads conclude, non-pipelined requests begin

Figure 33. Burst, Pipelined Read Request With Wait States, 16-Bit Bus

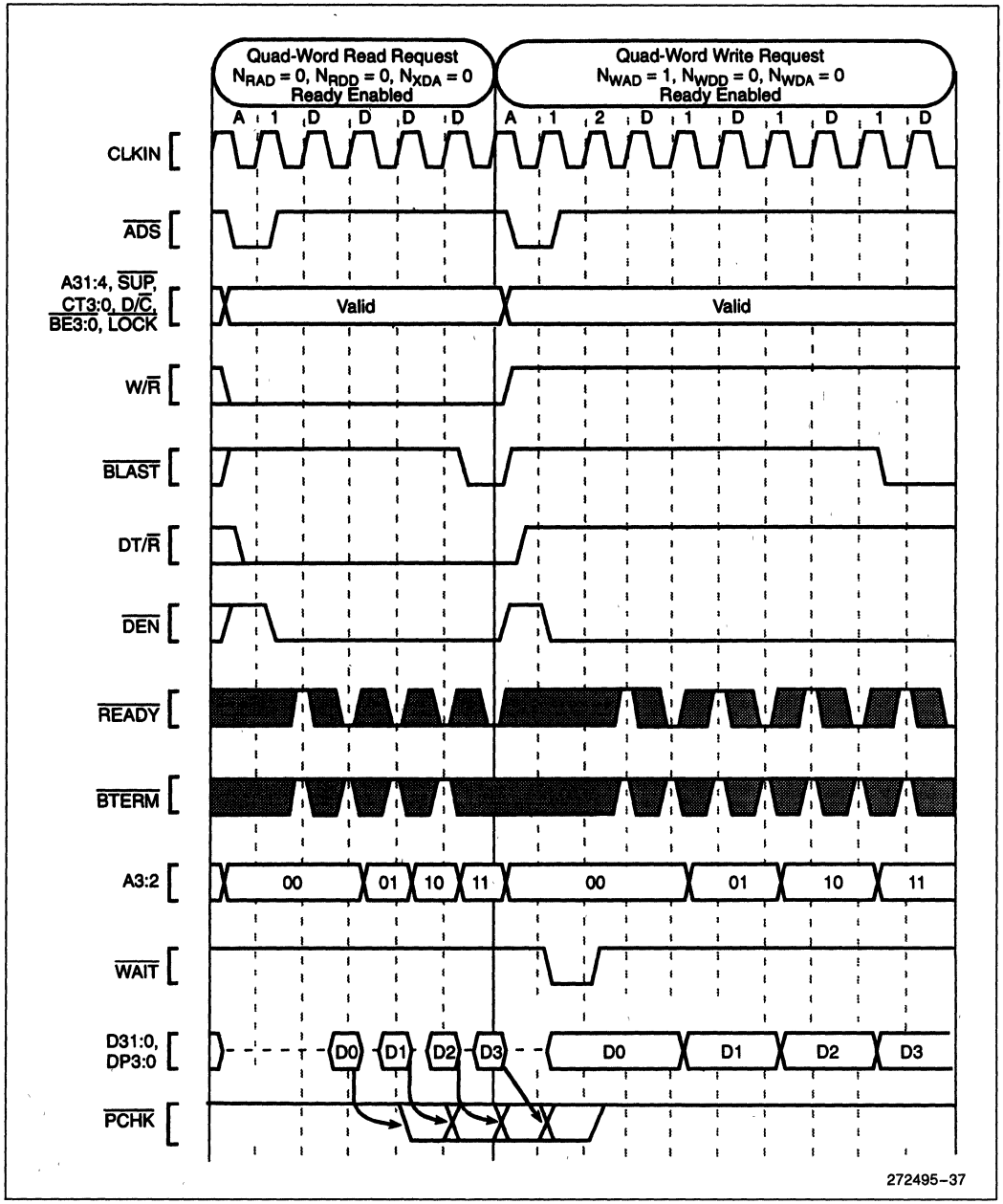
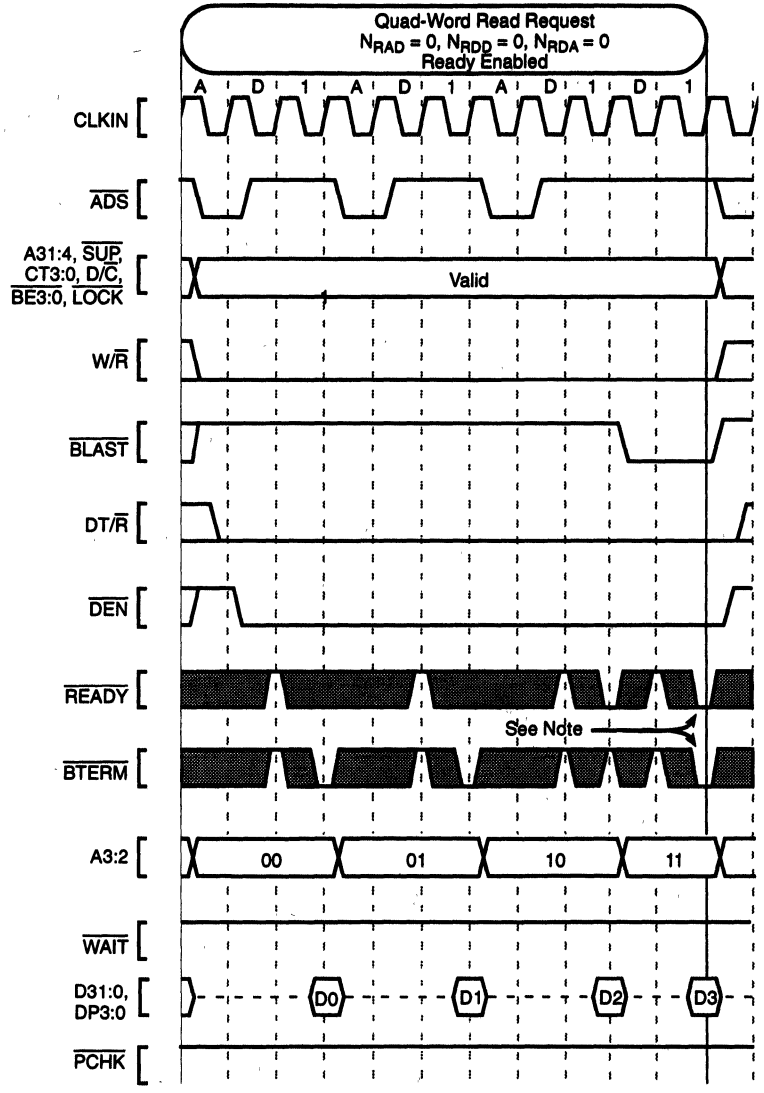
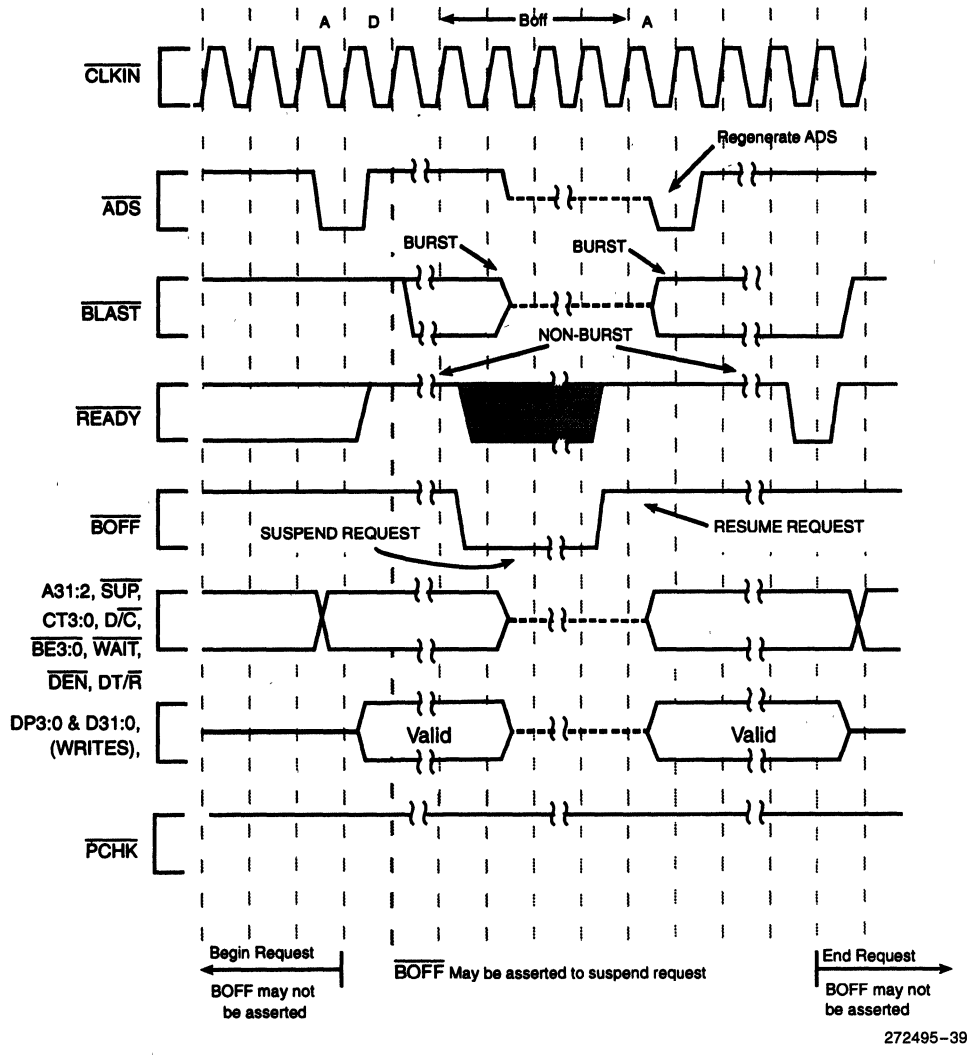


Figure 34. Using External READY



NOTE:
READY adds memory access time to data transfers, whether or not the bus access is a burst access. BTERM interrupts a bus access, whether or not the bus access has more data transfers pending. Either the READY signal or the BTERM signal will terminate a bus access if the signal is asserted during the last (or only) data transfer of the bus access.

Figure 35. Terminating a Burst with BTERM



NOTE:
READY/BTERM must be enabled: N_{RAD}, N_{RDD}, N_{WAD}, N_{WDD} = 0

Figure 36. BOFF Functional Timing

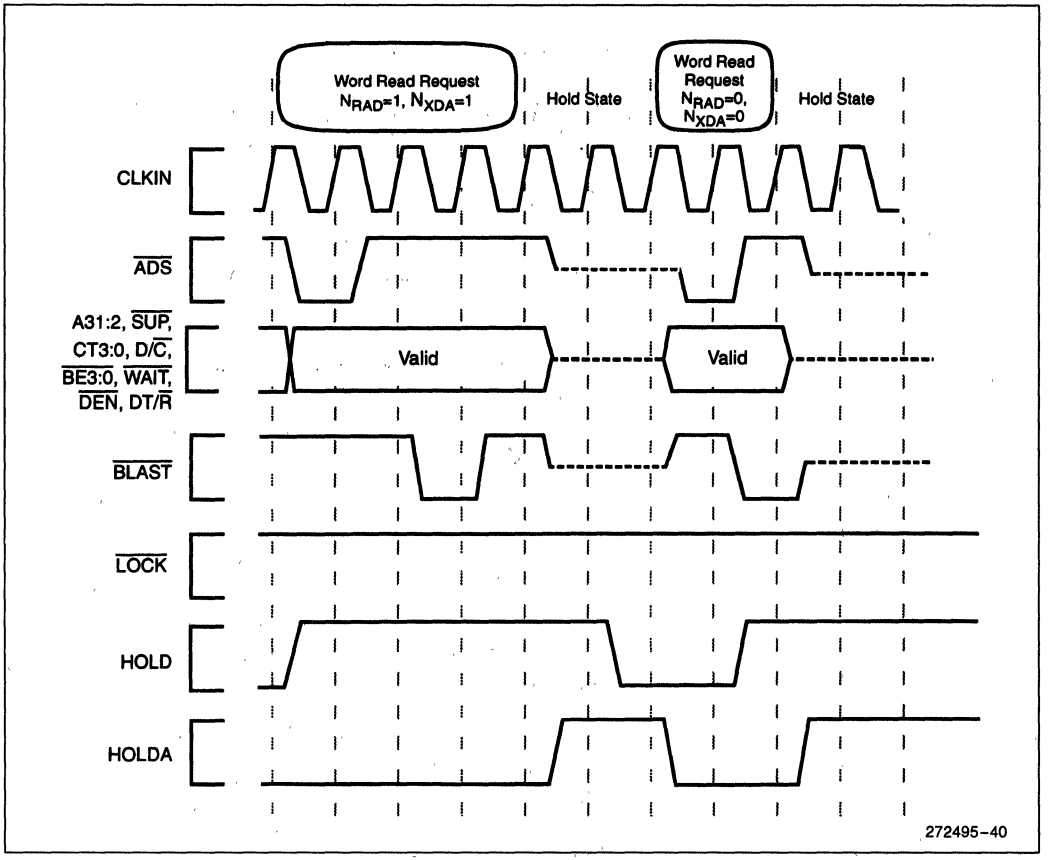


Figure 37. HOLD Functional Timing

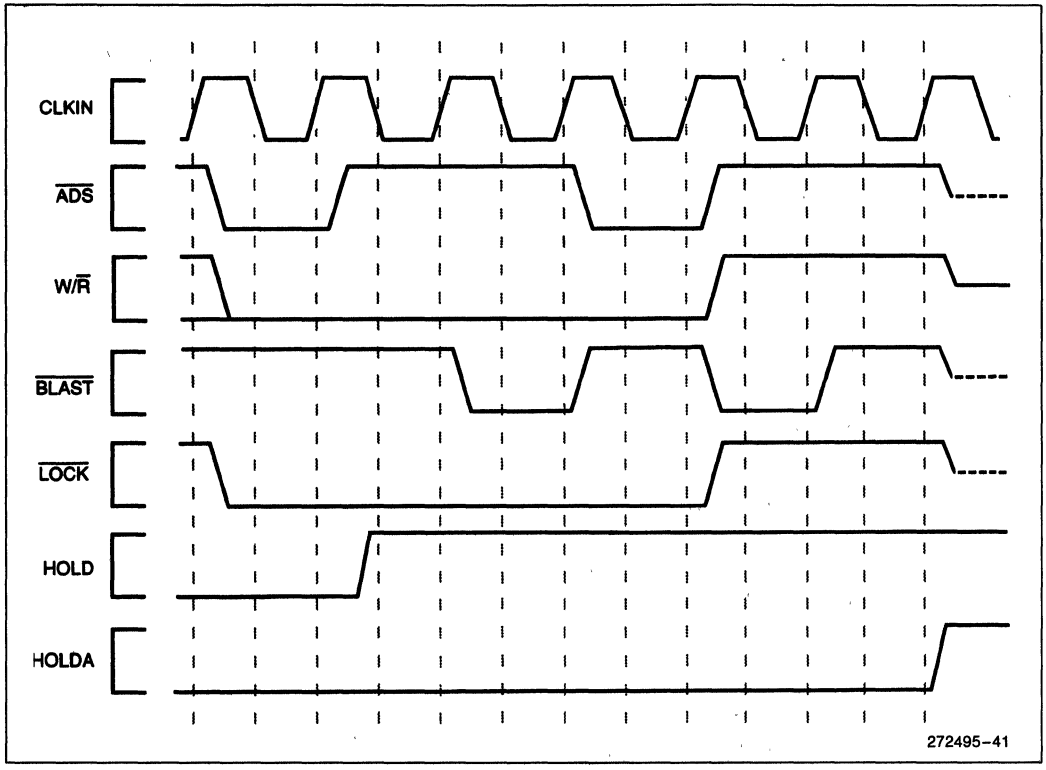


Figure 38. Lock Delays HOLDA Timing

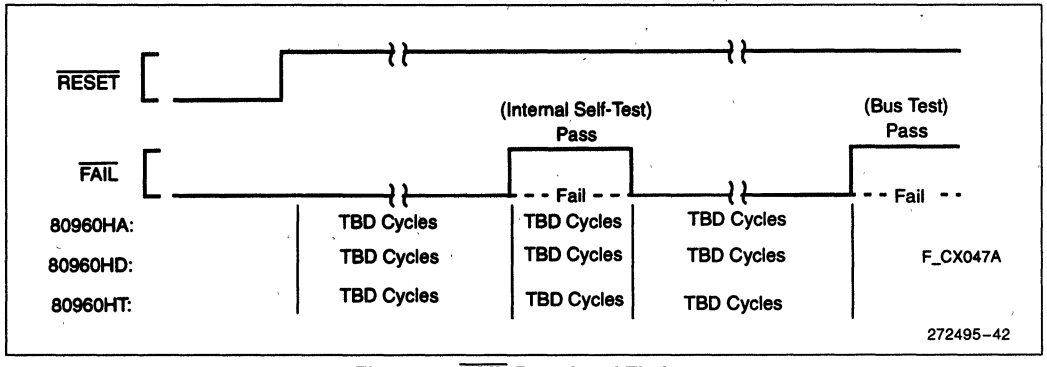
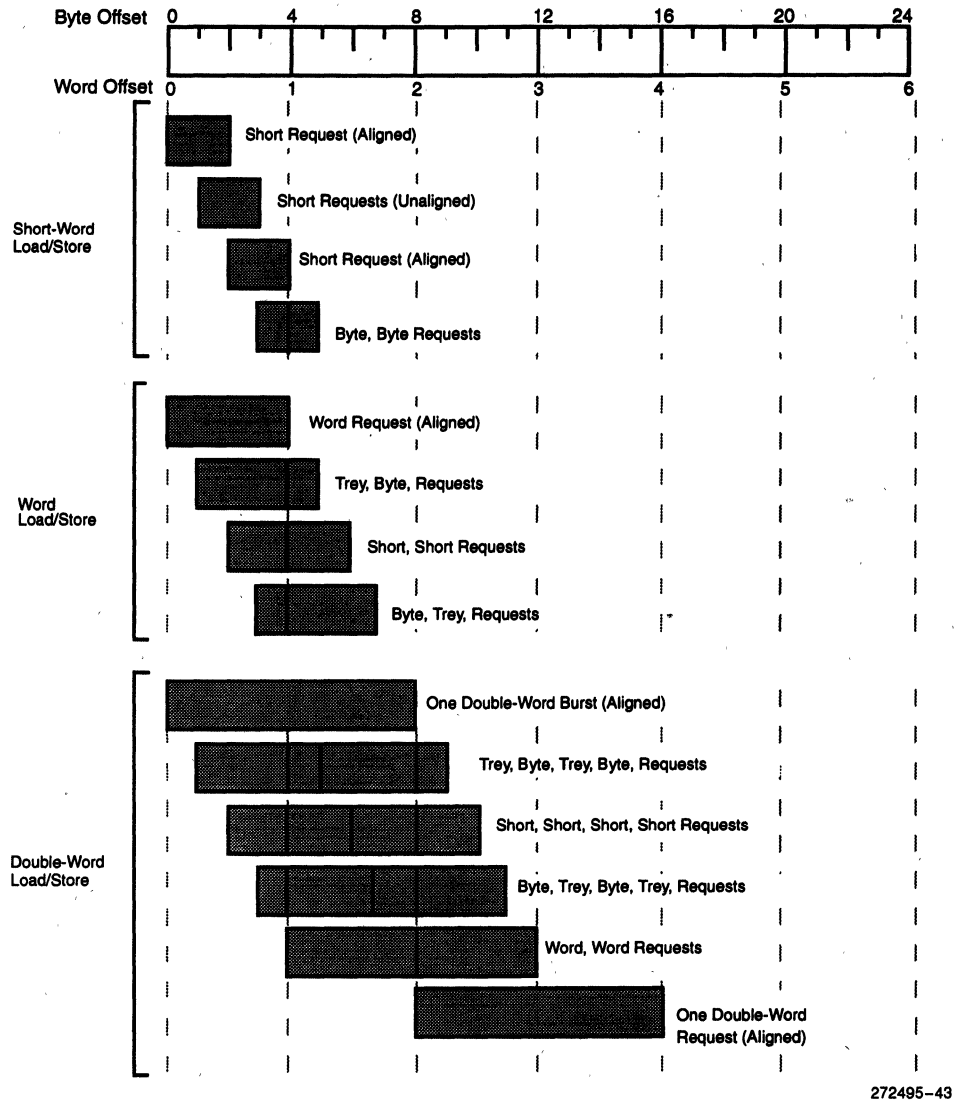
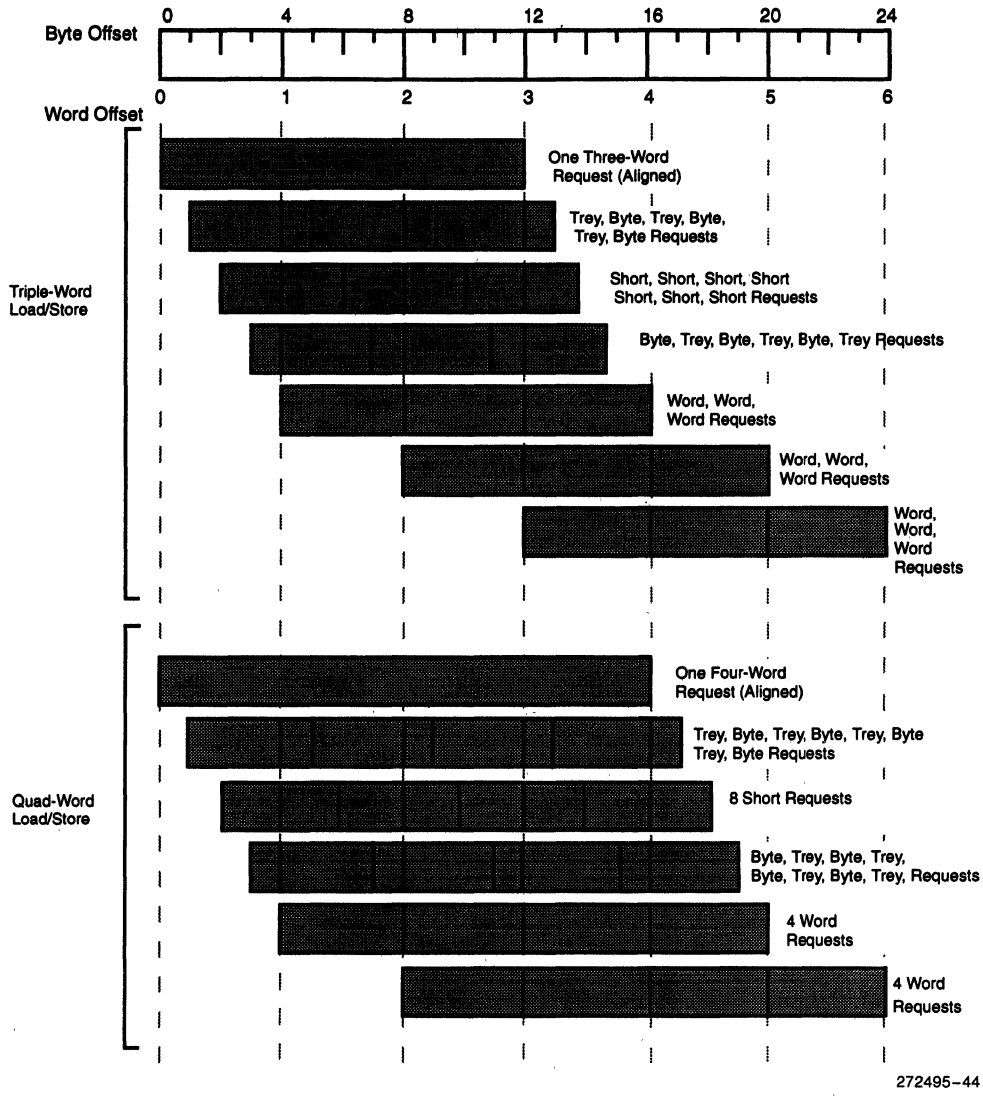


Figure 39. FAIL Functional Timing



NOTE:
All requests which are less than a word in size and are cacheable will be promoted to a word to be cached. This causes adjacent requests to occur for full words to the same address.

Figure 40. A Summary of Aligned and Unaligned Transfers for 32-Bit Regions



NOTE:
All requests which are less than a word in size and are cacheable will be promoted to a word to be cached. This causes adjacent requests to occur for full words to the same address.

Figure 41. A Summary of Aligned and Unaligned Transfers for 32-Bit Regions (Continued)

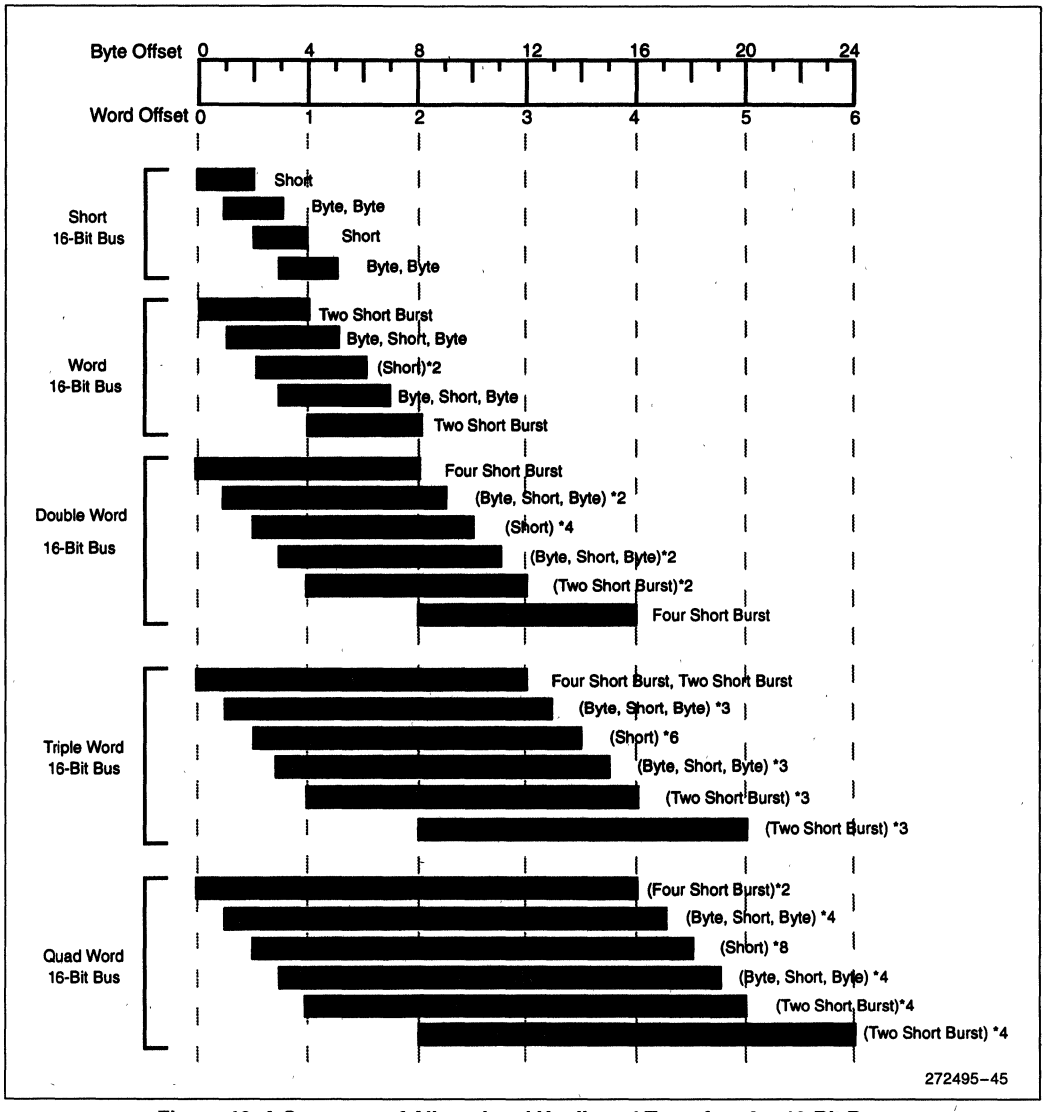


Figure 42. A Summary of Aligned and Unaligned Transfers for 16-Bit Bus

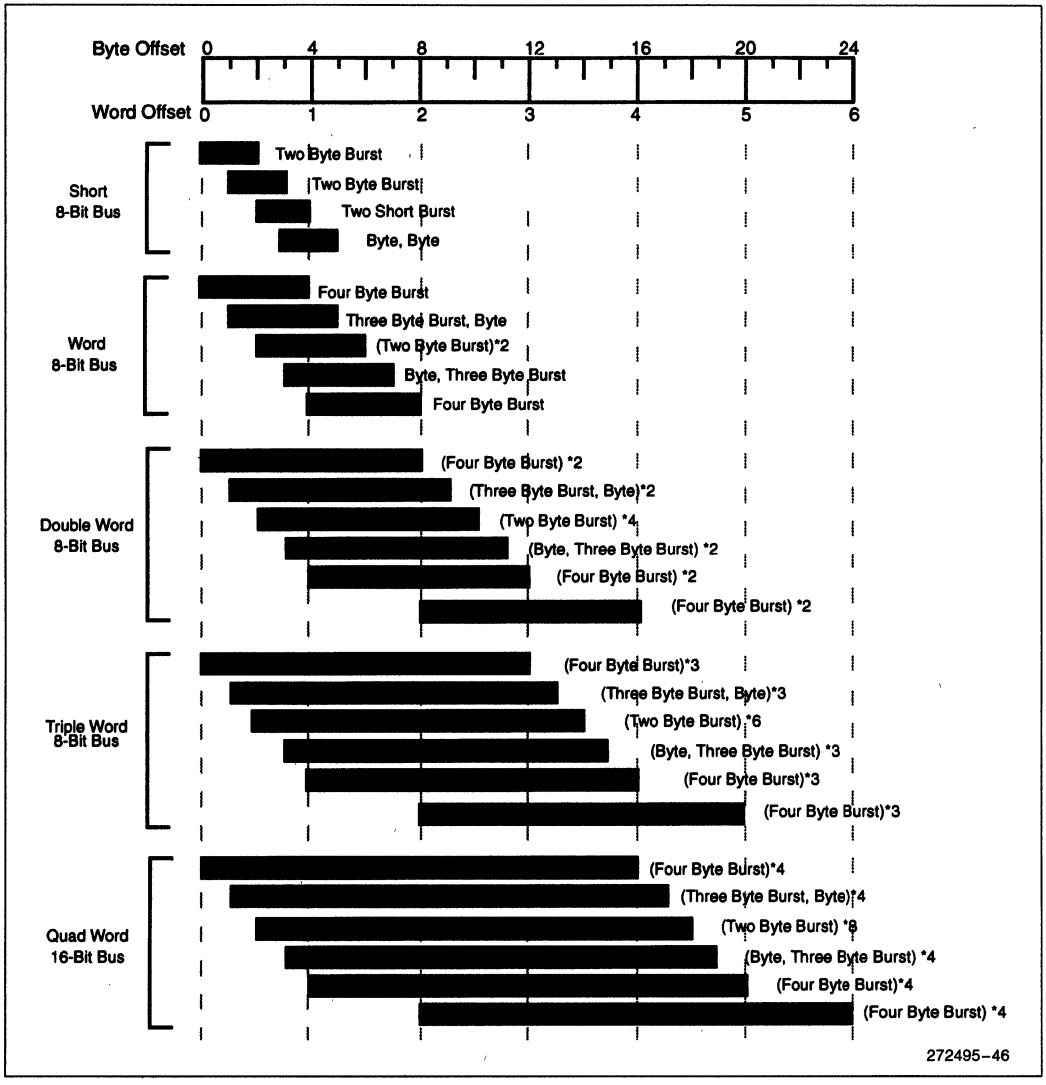


Figure 43. A Summary of Aligned and Unaligned Transfers for 8-Bit Bus

1

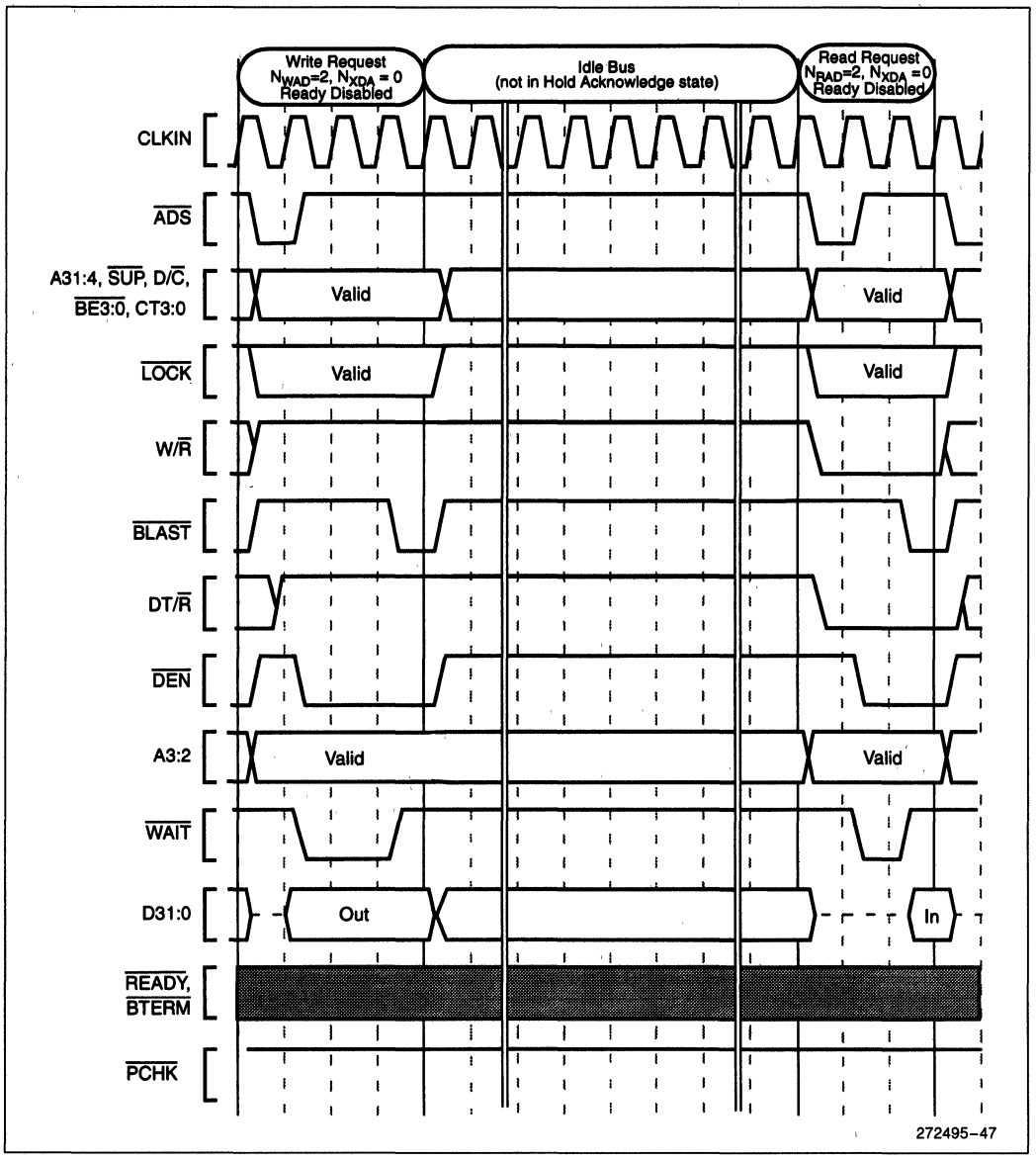


Figure 44. Idle Bus Operation

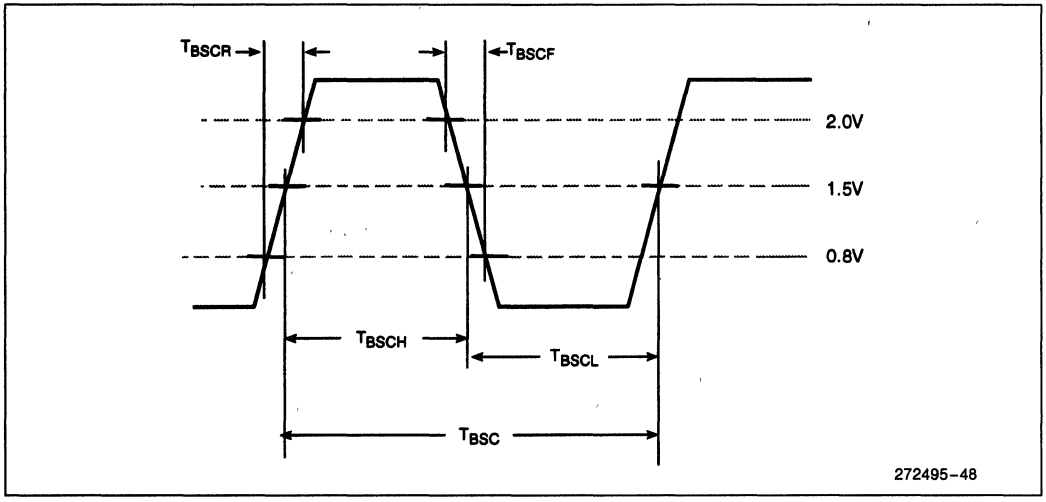


Figure 45. TCK Waveform

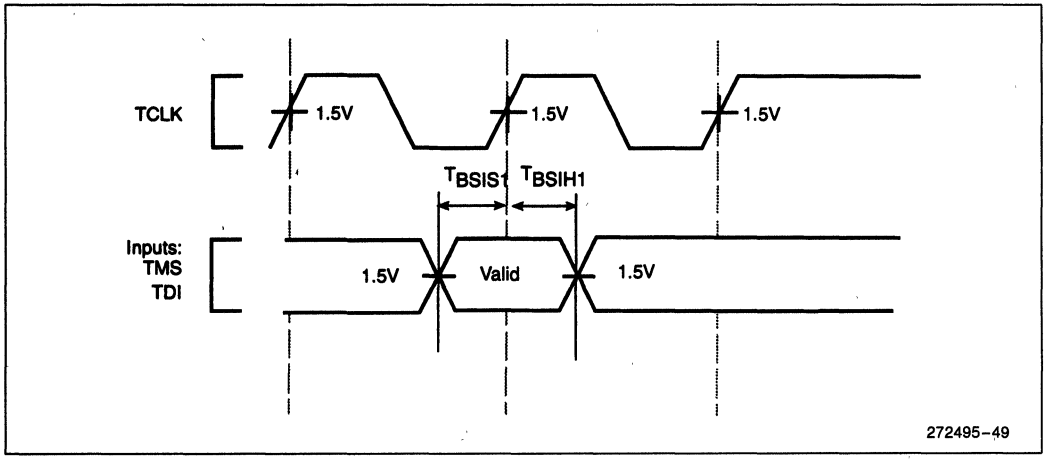


Figure 46. Input Setup and Hold Waveforms for T_{BSIS1} and T_{BSIH1}

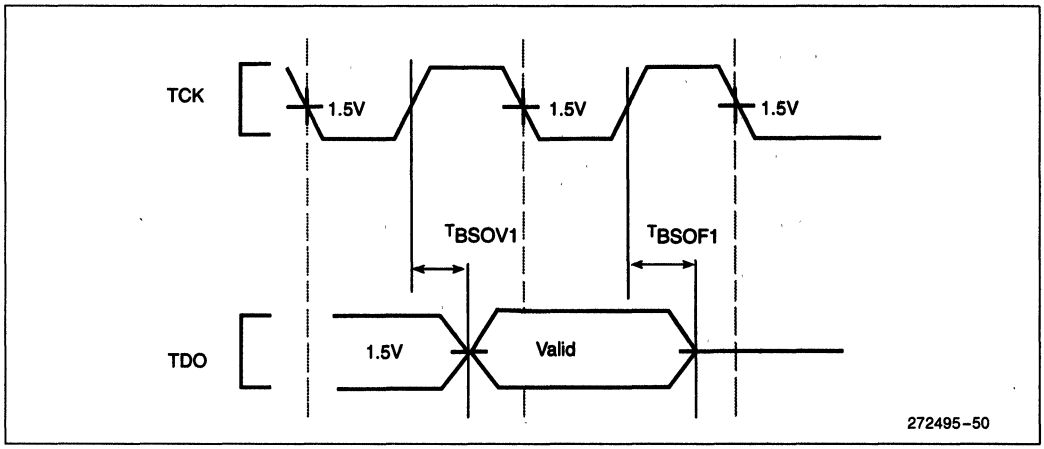


Figure 47. Output Delay and Output Float for T_{BSOV1} AND T_{BSOF1}

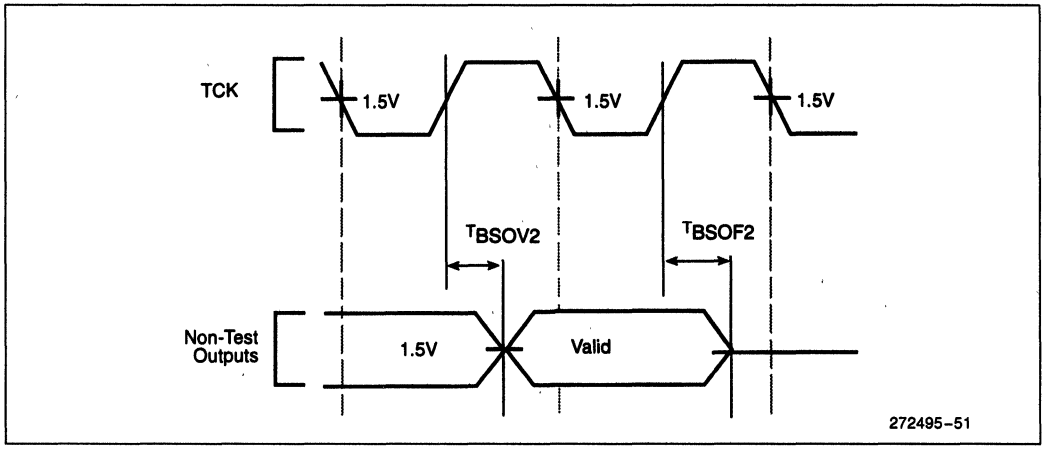


Figure 48. Output Delay and Output Float Waveform for T_{BSOV2} and T_{BSOF2}

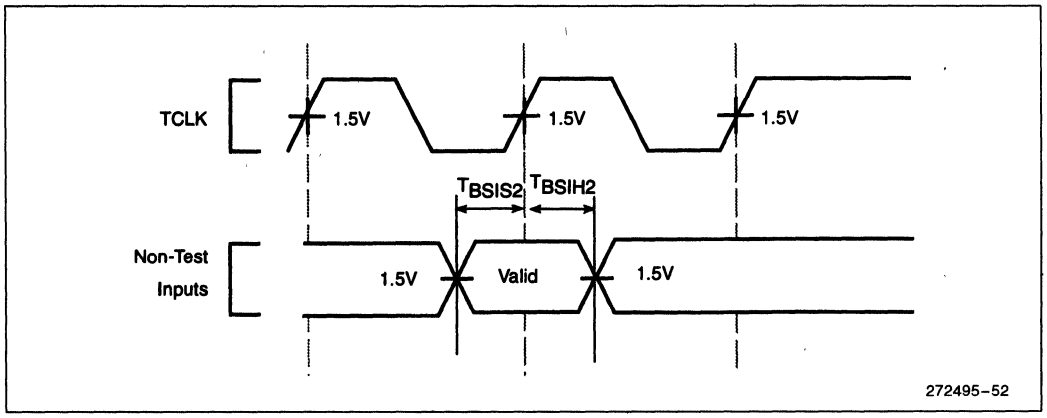
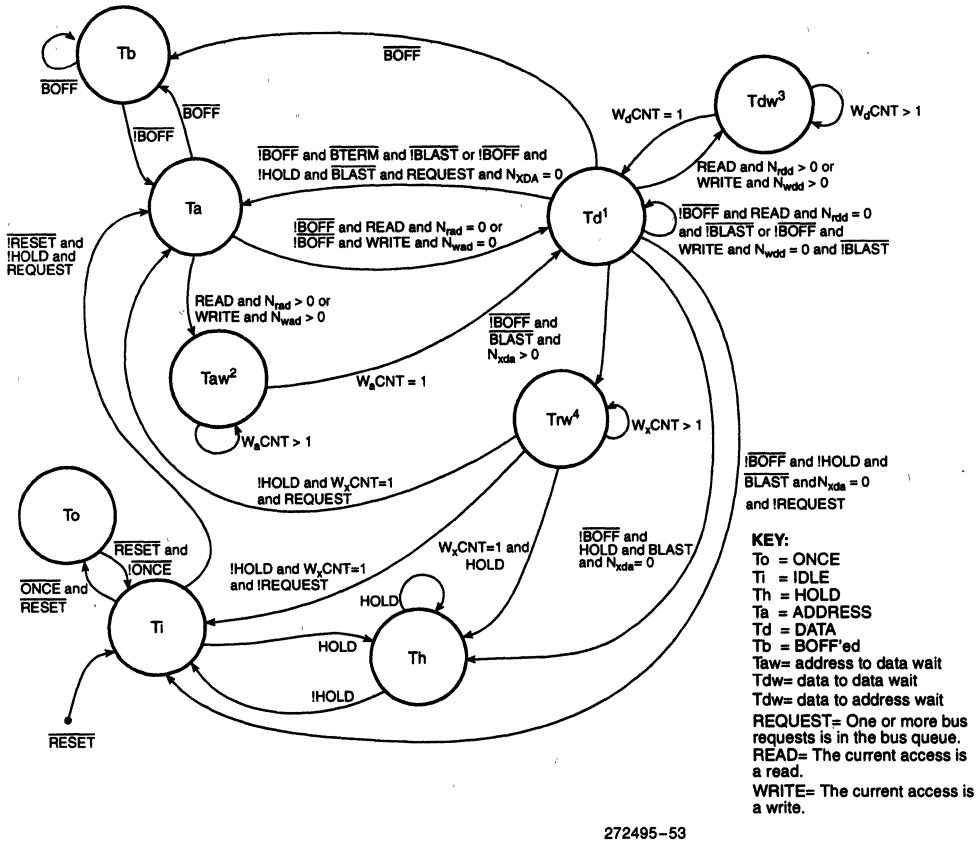


Figure 49. Input Setup and Hold Waveform for T_{BSIS2} and T_{BSIH2}



272495-53

NOTE:

1. When the PMCON for the region has External Ready Control enabled, wait states are inserted as long as READY and BTERM are de-asserted. In the case Read Pipelining is enabled, the Ta state of the subsequent read access is concurrent with the last data cycle of the access. Because External Ready Control is disabled for Read Pipelining, the address cycle occurs during BLAST.
2. W_aCNT is decremented during T_{aw}
3. W_dCNT is decremented during T_{dw}
4. W_xCNT is decremented during T_{rw}

Figure 50. Bus States

82961KD
PRINTER COPROCESSOR

- **High Performance Printer Coprocessor***
  - **— Direct Interface to Intel's i960® KA or KB** 32-Bit Embedded Processors**
- **Direct Generic Printer Engine Interface**
  - **— TEC, Canon, Ricoh, Okidata and Ink Jet Printer Engines**
- **Burst Interface Support for I960 KA/KB Bus**
- **Big Endian or Little Endian Byte Ordering of Font Cartridges**
- **DRAM Burst Accesses up to 256 Words for Graphics Operations**
- **Bit Map Image Compression**
- **Compressed Display List Processing**
- **Programmable EPROM Interface**
  - **— Interfaces to Eight Banks of Interleaved EPROM**
  - **— Individually Programmable Timing Parameters**
- **Programmable DRAM Interface**
  - **— Direct Interface to Four Banks**
  - **— Expandable to Eight Banks**
  - **— Transparent Refresh of DRAM Banks**
  - **— Supports 256 Kbit, 1 Mbit and 4 Mbit DRAMs**
  - **— Individually Programmable Timing Parameters for All Banks of DRAM**
- **Programmable I/O Control**
  - **— Chip Select, Access Time, Recovery Time**
  - **— Wait State Control for Eight External Devices**
- **Automatic Data Conversion from 16-Bit Font Cartridge to 32-Bit I960 Embedded Processor Format**
- **Low-Cost 164-Lead Plastic Quad Flat Pack (PQFP)**
- **Provides System Timer Functions**

1

Intel's 82961KD Printer Coprocessor provides the Intel i960 KA or i960 KB microprocessors with a powerful graphics accelerator and compression processor that dramatically increase system performance and reduce printer coprocessor system cost. This single chip device provides all necessary system control for the i960 KA or i960 KB microprocessors and a direct interface to most laser printer engines. The 82961KD coprocessor contains complete DRAM, I/O and interleaved ROM controllers, font cartridge support and the associated logic required to control most non-impact printer mechanisms, a programmable wait state generator and programmable chip select generation logic.

The 82961KD Printer Coprocessor performs all graphics functions necessary for complex page description language (PDL) or printer control language (PCL) controllers. Image compression is achieved using "Scanline Tables". Memory requirements for storage of bit mapped images—such as character font cache and graphics objects—is significantly reduced using these structures.

The 82961KD coprocessor processes a compressed display list to form the bit mapped image of the page to be printed. The 82961KD coprocessor automatically supports "Band buffered" print operations. The chip's compressed display list, coupled with its fast graphics operations, allow band buffered printing of very complex page description language (PDL) pages such as those PostScript† generates.

---

*The 82961KD i960 Printer Coprocessor is based on the single-chip controller architecture created by Peerless Systems, Corp.

**Throughout this data sheet, 80960Kx refers to the 80960KA and KB processors.

†Other brands and names are the property of their respective owners.

August 1992
Order Number: 272221-001

intel®

AP-506

APPLICATION NOTE

Designing for 80960Cx and 80960Hx Compatibility

Larry Gass
80960 Applications Engineer

With contribution from AP-505
(David Harriman, author)

Intel Corporation
Embedded Processor Division
Mail Stop CH5-233
5000 W. Chandler Blvd.
Chandler, Arizona 85226

November 1994

PRODUCT PREVIEW

Order Number: 272556-001

DESIGNING FOR 80960Cx AND 80960Hx COMPATIBILITY

CONTENTS

CONTENTS

INTRODUCTION

The 80960HA/HD/HT[1] processors, Intel's new superscalar i960® processor, adds new features and performance to the other well-known products in the i960 processor family. The 80960Hx is designed to satisfy the compute-intensive, data throughput performance requirements of both today's applications and those of the future.

This document addresses the important hardware considerations when designing a "80960Hx ready" system[2]. This is a system which is designed to use an i960 Cx processor[3] and can also use the 80960Hx processor (when available). To help simplify this task, pinout for the 80960Hx PGA package is similar to pinout for the i960 Cx processor PGA package. Although the 80960Hx is not drop-in compatible with all Cx designs, systems can be built with a CPU socket footprint which will accept either processor.

A summary of the most important hardware design considerations are:

power supply — V_{CC} for the Cx is 5V; the Hx uses 3.3V. An 80960Hx-ready system's power supply must accommodate these voltage requirements.

DMA controller — Cx processors have a built-in DMA controller; the Hx does not. An 80960Hx-ready system should not use the Cx built-in DMA controller. Cx pins used for DMA control have different function on the Hx.

byte enable signals — The Hx's byte enable encodings are a superset of the Cx byte enable encodings. The 80960Hx-ready system should be designed to accept all combinations of byte enable encodings.

bus arbitration — The Hx does not grant HOLD requests during an atomic operation (assert HOLDA in response to HOLD), but Cx processors will grant HOLD requests after any bus request, including in the middle of atomic accesses. A 80960Hx-ready system must not allow HOLD requests when the external LOCK pin is asserted if semaphore operations are to be performed between bus masters.

The Hx has an additional arbitration signal — BSTALL — which can be used by an external arbiter to indicate the processor has stalled because the bus controller is busy. (The Cx does not have BSTALL.)

external interrupts — Interrupt subsystems must produce asynchronous interrupt inputs. The Hx samples interrupts differently than the Cx processors.

N_{XDA} wait states — A system must not rely on N_{XDA} wait states between each access. Although both the Hx and Cx processors have programmable N_{XDA} wait states, behavior in the Hx is different. The Hx always inserts N_{XDA} wait states between accesses. The Cx only inserts N_{XDA} wait states between bus "requests." Each bus request can cause multiple bus accesses.

An 80960Hx-ready system must NOT accept data on writes during N_{XDA} wait states. During N_{XDA} wait states, the Hx processor drives the D31:0 bus. Cx processors do not drive valid data during N_{XDA} wait states.

parity — The Hx provides built-in byte parity; Cx processors do not. If parity is used when the system contains an Hx processor, pull-up resistors must be provided to ensure that inputs sent to either the processor or to the external parity system do not float.

---

[1] Throughout this document, "Hx" refers to the i960 HA, HD and HT processors. Information that is specific to each is clearly indicated.

[2] "80960Hx-ready" refers to a system designed to use a CA/CF processor that can also use an 80960Hx.

[3] Throughout this document, "Cx" refers to both the i960 CA and CF processors. Information that is specific to each is clearly indicated.

boundary scan The Hx has an IEEE 1149.1 JTAG interface; conversely, the Cx does not support JTAG. If JTAG is used when the system contains a Cx processor, the processor must be externally bypassed in the JTAG chain.

reserved memory Accesses to reserved memory (0xffxxxxxx) do not appear on the Hx bus. The Cx uses 0xffffffxx to fetch the Initial Boot Record. External decoders should map this memory to two different areas in the processor's address space.

AC timing AC specifications differ for Hx and Cx processors. Of course, AC timing analysis must be performed when designing a 80960Hx-ready system. The Cx AC timings are referenced to PCLK2:1; on the Hx, AC timings are referenced to CLKIN. (The Hx does not have PCLK2:1 signals.)

POWER REQUIREMENTS

The Hx requires a V_{CC} of 3.3V while the Cx operate at 5V. A system can be designed with a socket that accepts either processor. The Hx processor may be damaged if plugged into a socket that supplies 5V V_{CC}. Jumpers, switches, programmable power regulators, or other V_{CC} switching must be provided to select the proper V_{CC} for the processor. The 80960Hx's VOLDET pin can be used to accommodate automatic voltage selection circuitry.

An 80960Hx-ready system requires 5V on the VCC5 pin to provide 5V tolerant inputs.

Providing 3.3 V in a 5 V System

In most system board designs, the 5 V system power supply is routed to the components on the board through a dedicated board layer. With the requirement of a new 3.3 V supply for the Hx, it is not necessary to add a completely new power supply layer to the circuit board, as it is possible to create a 3.3 V "island" around the processor in the existing power supply plane.

Figure 1 shows a recommended "island" layout. The Hx processor's 5 V tolerant input buffers and TTL compatible outputs allow the processor to interface with existing TTL compatible external logic without requiring extra components. Thus, the processor can run at 3.3 V while the system logic runs at 5 V.

Other important considerations are:

- The "island" needs to be large enough to include the processor, the required power supply decoupling capacitance, and the necessary connection to the 3.3 V source.
- To minimize signal degradation, the gap between the 3.3 V "island" and the 5 V plane should be kept small. A typical gap size is about 0.02 inches.
- Minimize the number of traces routed across the power plane gap, since each crossing introduces signal degradation due to the impedance discontinuity that occurs at the gap. For traces that must cross the gap, route them on the side of the board next to the ground plane to reduce or eliminate the signal degradation caused by crossing the gap. If this is not possible, route the trace to cross the gap at a right angle (90 degrees).
- Use liberal decoupling capaciatnce between the 5V plane and the 3.3V island. A0.01 μf ceramic capacitor every 0.5 to 1.0 inches along the perimeter of the island will greatly reduce the impedance discontinuity.

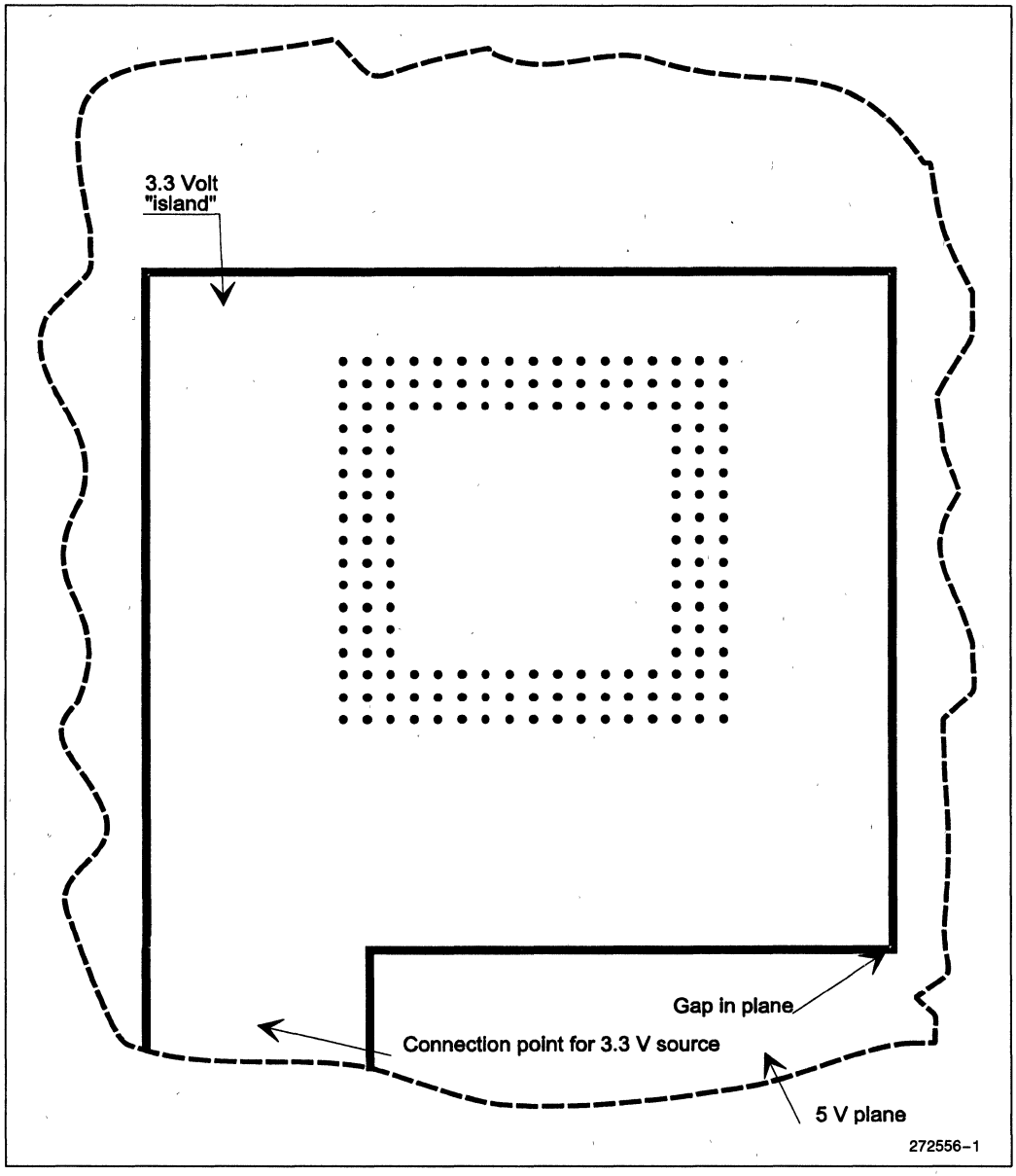


Figure 1. Creating a Power "Island"

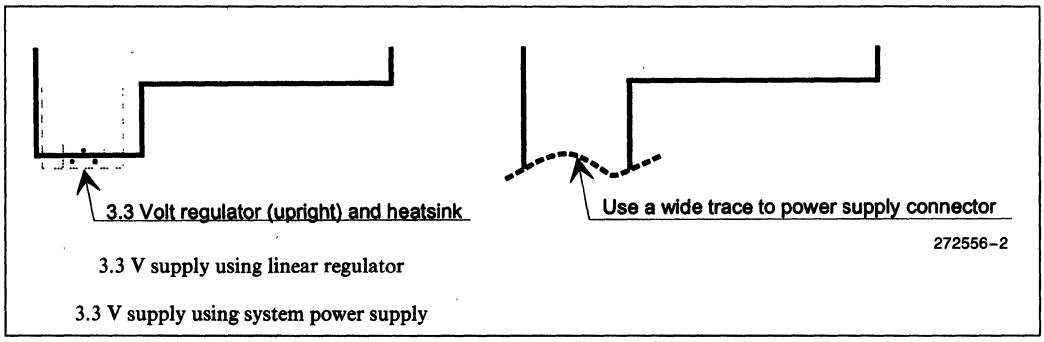


Figure 2. Recommended Power Supply Connection Layout

Choosing a Power Source

The primary concerns which must be addressed when selecting a power source are maximum and minimum load current requirements and response time. The processor power supply must be able to maintain correct voltage regulation at current levels below 10 mA for the Hx in the HALT Mode, and up to the maximum current of 1.5 A.

Executing a HALT instruction causes the Hx to enter the HALT Mode, which causes a significant reduction in the current consumption of the processor in as few as 100 ns. The transition from HALT to the Normal State causes current consumption to return to the normal levels in a similarly short period of time. The processor power supply must be able to maintain correct voltage regulation during these transitions.

There are basically two options for supplying 3.3 V to the processor, either:

- Add a 3.3 V tap to the primary system power supply
- Use on-board secondary regulation to derive 3.3 V from the 5 V system power supply

For on-board secondary regulation, a linear voltage regulator will perform adequately for most designs. If low heat or power dissipation is a design goal, the higher complexity and cost of a switching regulator may be warranted. Switching regulators offer better efficiency, thereby lowering regulator power consumption and heat.

Figure 2 shows recommended layouts for power supply or linear regulator connection to the 3.3 V "island."

Power Supply Selection For Flexible Systems

Using the 80960Hx's voltage detect sense feature, you may design a flexible system which will automatically provide the proper processor voltage for an 80960Hx or Cx processor. It is also possible to make the selection of processor voltage an option during system board assembly.

VOLDET Automatic Voltage Select Circuit Option

By sampling the VOLDET pin at powerup, system boards can automatically select the processor power supply voltage, enabling a design that may use the 3.3V Hx or a 5 V Cx processor without jumpers or assembly time changes. The VOLDET pin is only present in the PGA package version of the Hx. This pin, which is an NC (No Connect) on the Cx processor, is connected internally to V_{SS} on the Hx. This pin should be left unconnected in designs that do not use the voltage detect feature.

Figure 3 shows an example of VOLDET pin usage with a linear regulator circuit to automatically select the correct power supply voltage. If VOLDET is not connected inside the processor, indicating a 5 V part, the gate of MOSFET Q1 is pulled high, which bypasses the 3.3 V regulator, supplying 5 V directly to the processor. Shorting the regulator's input to the output in this way is harmless for most linear regulators, due to regulator feedback circuitry which shuts the regulator off (con-

tact regulator manufacturers for specifics). Note that in this case, most regulators require Q1 to handle all the processor's current requirements, and so should be a high-current, low on-state-resistance MOSFET. If VOLDET is connected to V_{SS}, indicating a 3.3 V part, the Q1 transistor is turned off, allowing the regulator to function normally. Figure 4 shows a suggested placement and layout for MOSFET Q1.

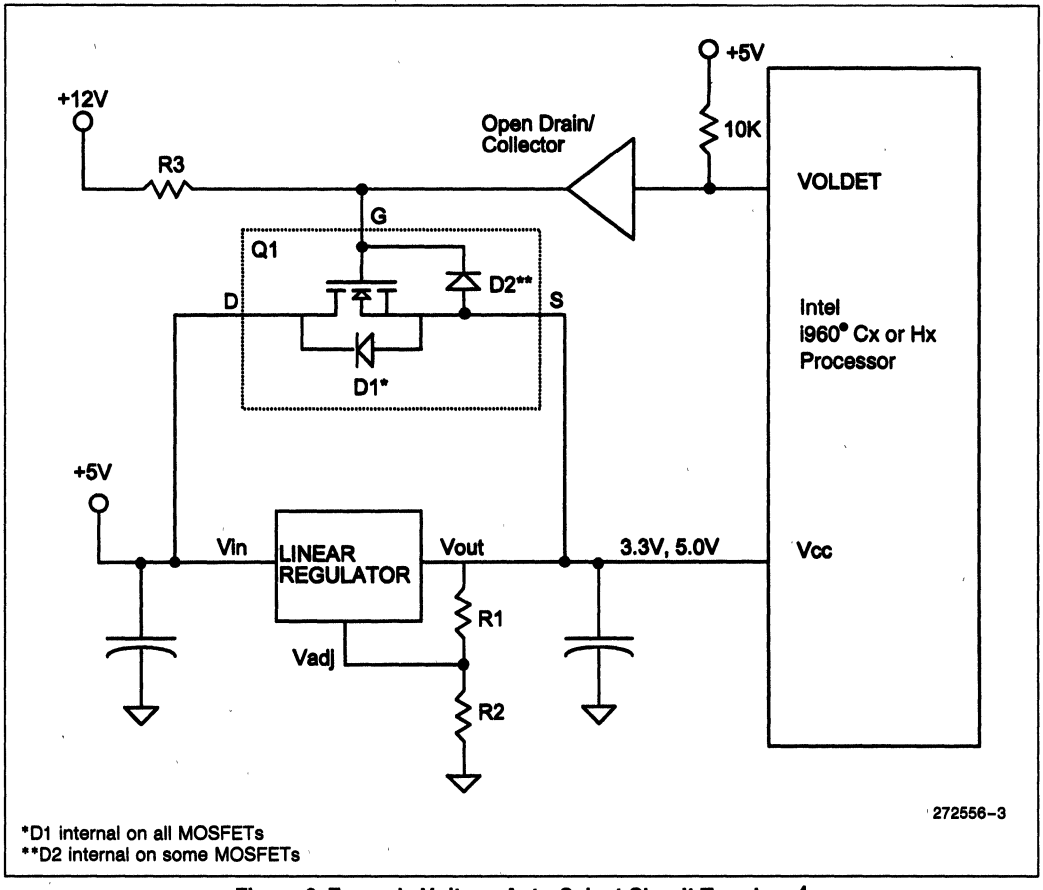


Figure 3. Example Voltage Auto-Select Circuit Topology[4]

[4]Illustration courtesy of Linear Technology Corporation

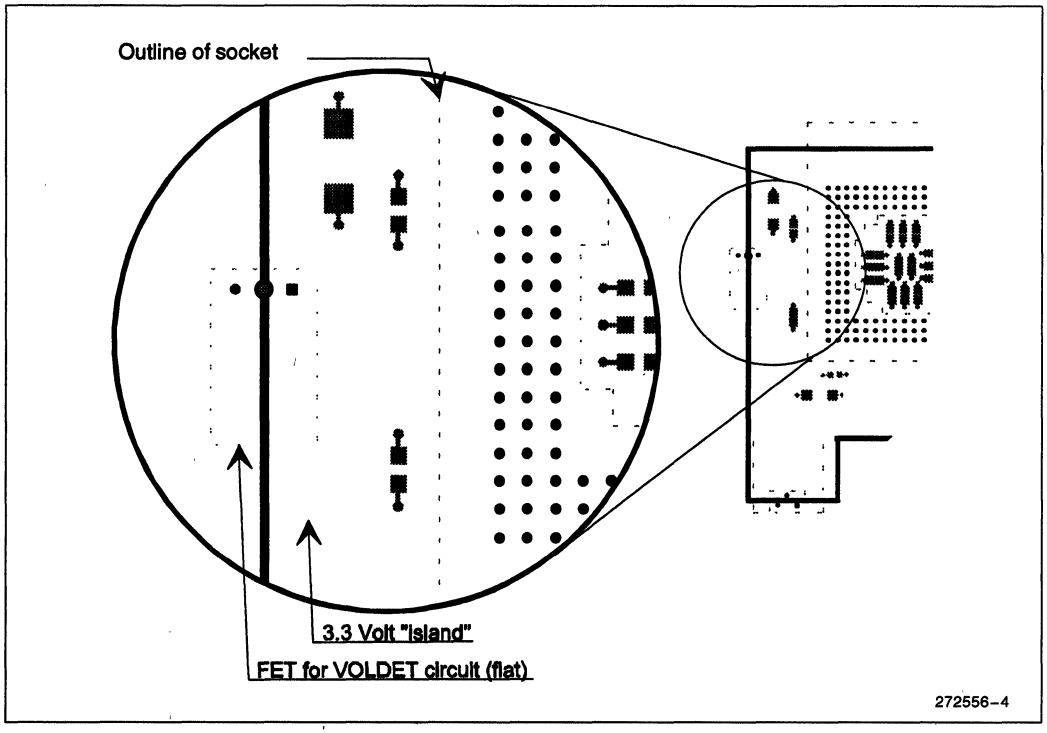


Figure 4. Suggested Placement and Layout for MOSFET Used in Optional Voltage Auto-select Circuit

Other Voltage Selection Options

It is also possible to design a flexible system board where the processor supply voltage is selected by an assembly time option. There are several methods to achieve this; the key requirement being that the design must handle the maximum current of 1.5 A.

VCC5 Pin Requirement

For mixed voltage systems where the processor interfaces with 5 V components, the VCC5 pin must be connected to 5 V for proper 5 V tolerant buffer operation. **The VCC5 input should not exceed V_{CC} by more than 2.25 V during power-up, power-down or during operation.** If this requirement is not met, current flow through the pin may exceed 55 mA which may damage the component. To meet this requirement, one of two things must be done:

- The power supply must be designed to turn on and off such that the difference between the VCC5 and V_{CC} voltages never exceeds 2.25 V, or,
- A 100 Ω resistor must be put in series with the VCC5 pin to limit the current through this path (Figure 5 shows a possible layout for this connection).
- The 100 Ω series resistor is required for power supplies which do not meet the voltage difference specification, and also provides protection in the case of a power supply failure (where the 5 V supply remains on, but the 3.3 V supply goes to zero).

The VCC5 pin corresponds to a NC (no connect) pin on the Cx processor. This pin has no effect on the operation of the Cx, and can be driven.

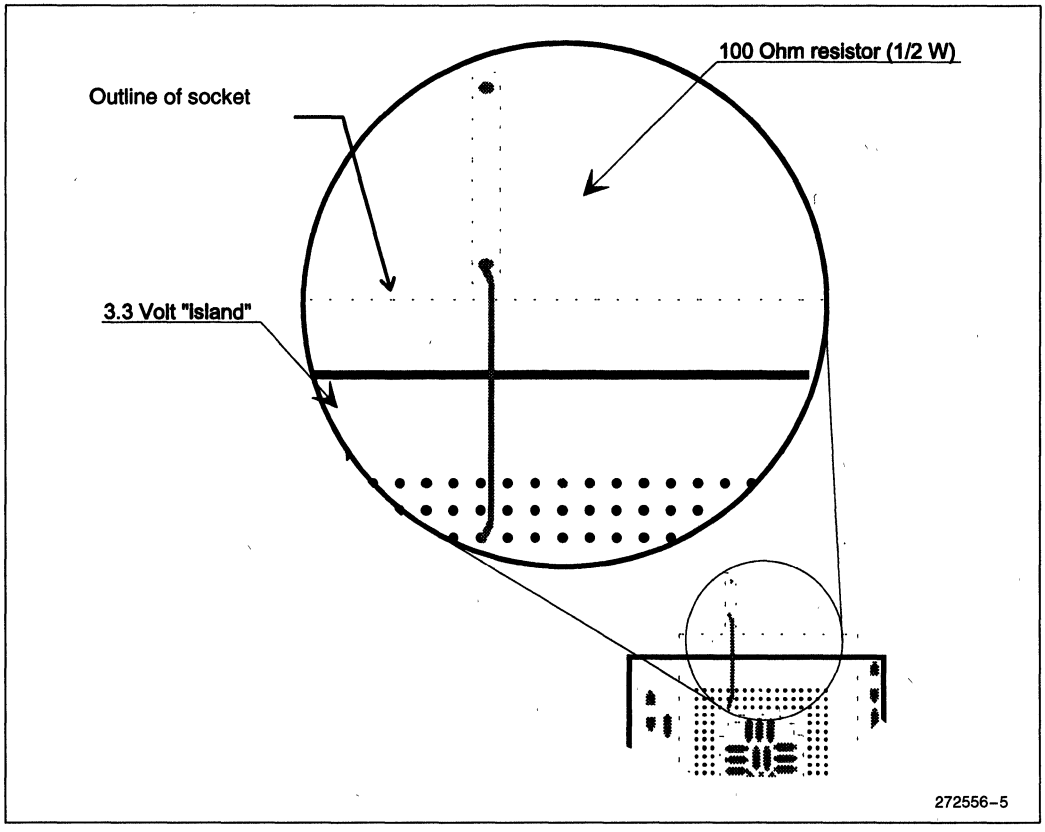


Figure 5. Possible Layout For VCC5 Pin Connection

Processor Power Supply Decoupling

Processor power supply decoupling is critical for reliable operation. With the 80960Hx-ready system, there are two areas of concern, each of which are described in the following subsections:

- High frequency decoupling, necessitated by the processor's high speed operation
- Low frequency decoupling, necessitated by the processor's power saving features

High Frequency Power Supply Decoupling

High frequency decoupling is critical on the Cx processor. It is especially critical on the Hx processor, because of its high speed external bus, and also because of its very fast 66 MHz internal operation.

A reliable design will include a minimum of nine 0.1 μF capacitors and nine 0.01 μF surface mount capacitors between power and ground, evenly distributed, close to the processor. The capacitors must be placed as close to the processor as possible, attached directly to the power and ground planes, or circuit board inductance will significantly reduce their effectiveness.

A typical failure mode caused by inadequate high frequency decoupling is unreliable or inconsistent program behavior. These failures are often intermittent, and are very hard to debug. Figure 6 shows a recommended layout for the high frequency capacitors, with values as shown.

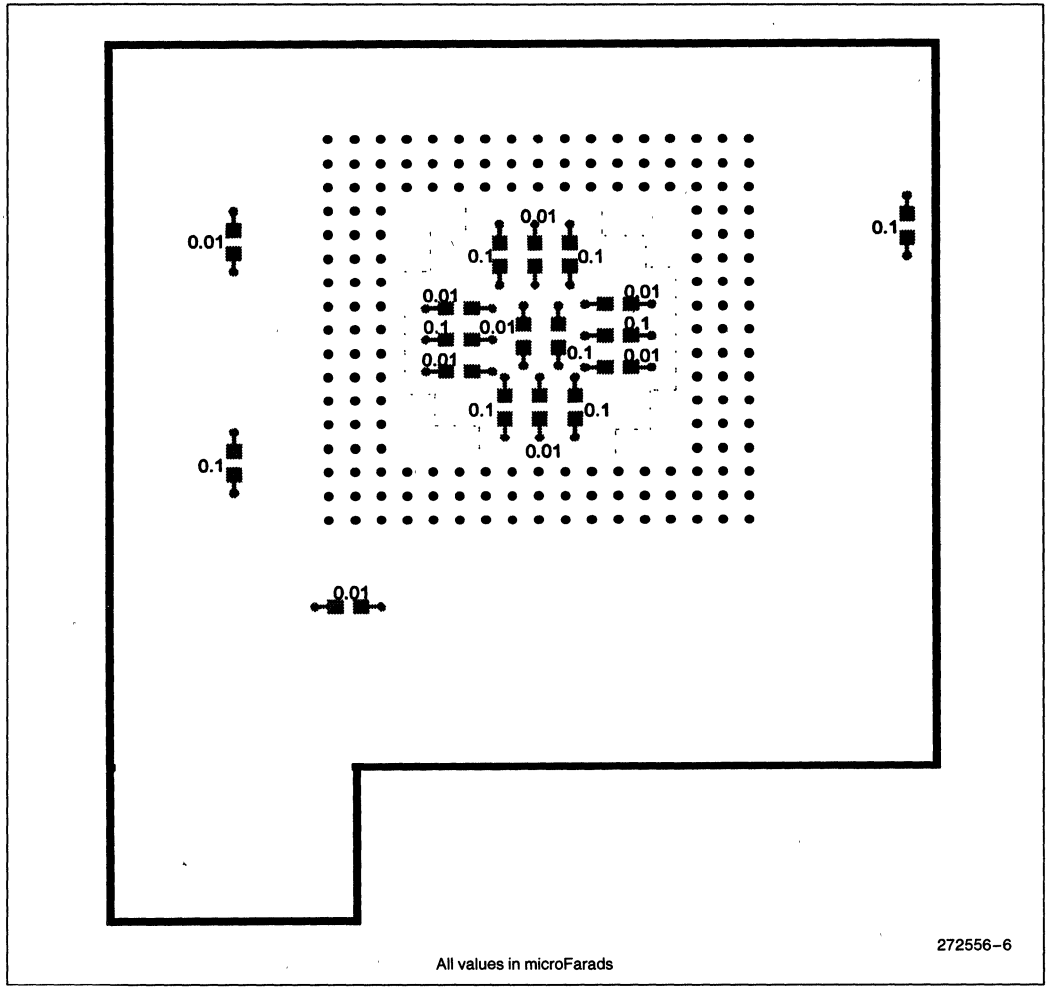


Figure 6. Recommended High-Frequency Capacitor Values and Layout

Bulk Power Supply Decoupling

Bulk, or low frequency, decoupling is needed on all i960 processors, including the Cx and Hx processors, since the Hx processor may switch between normal and low power states very quickly, causing large instantaneous current changes. To properly handle these instantaneous current changes, all designs must have adequate bulk decoupling.

In 5 V only systems, the processor can use the bulk decoupling capacitance all over the system board; however — with the processor on a separate power plane "island" — it is necessary to place adequate bulk capacitance on the processor "island." For bulk decoupling, multiple capacitors each in the range of 10 ΩF to 100 ΩF are typically used in parallel to achieve the required capacitance while maintaining a low effective series resistance (ESR). You can determine the amount of bulk decoupling required with the following formula:

$$C \approx (\Delta I * \Delta T) / \Delta V$$

where ΔI is the maximum change in current, ΔT is the time it takes the power supply to adjust to the current change, ΔV is the allowable voltage change to remain within specification.

The effective series resistance (ESR) must also be taken into account. You can find the maximum allowable ESR with this formula:

$$ESR \approx \Delta V / \Delta I$$

where ΔV and ΔI are the same as in the first equation.

For example, for the Hx processor, the maximum change in current is about 1.5A. The response time of a linear regulator may be around 15 μs (contact regulator manufacturer for precise value). With no guard band, the maximum allowable supply voltage deviation from 3.3 V is 0.3 V, yielding the following:

$$C \approx (1.5\ A * 15\ \mu s) / 0.3\ V = 75\ \mu F$$

with a maximum allowable ESR:

$$ESR \approx 0.3\ V / 1.5\ A = 0.2\ \Omega$$

Placing four 33 μF tantalum surface mount capacitors in parallel, directly between the power and ground planes, will reduce the ESR below this limit and provide adequate capacitance. Figure 8 shows a recommended layout for this example.

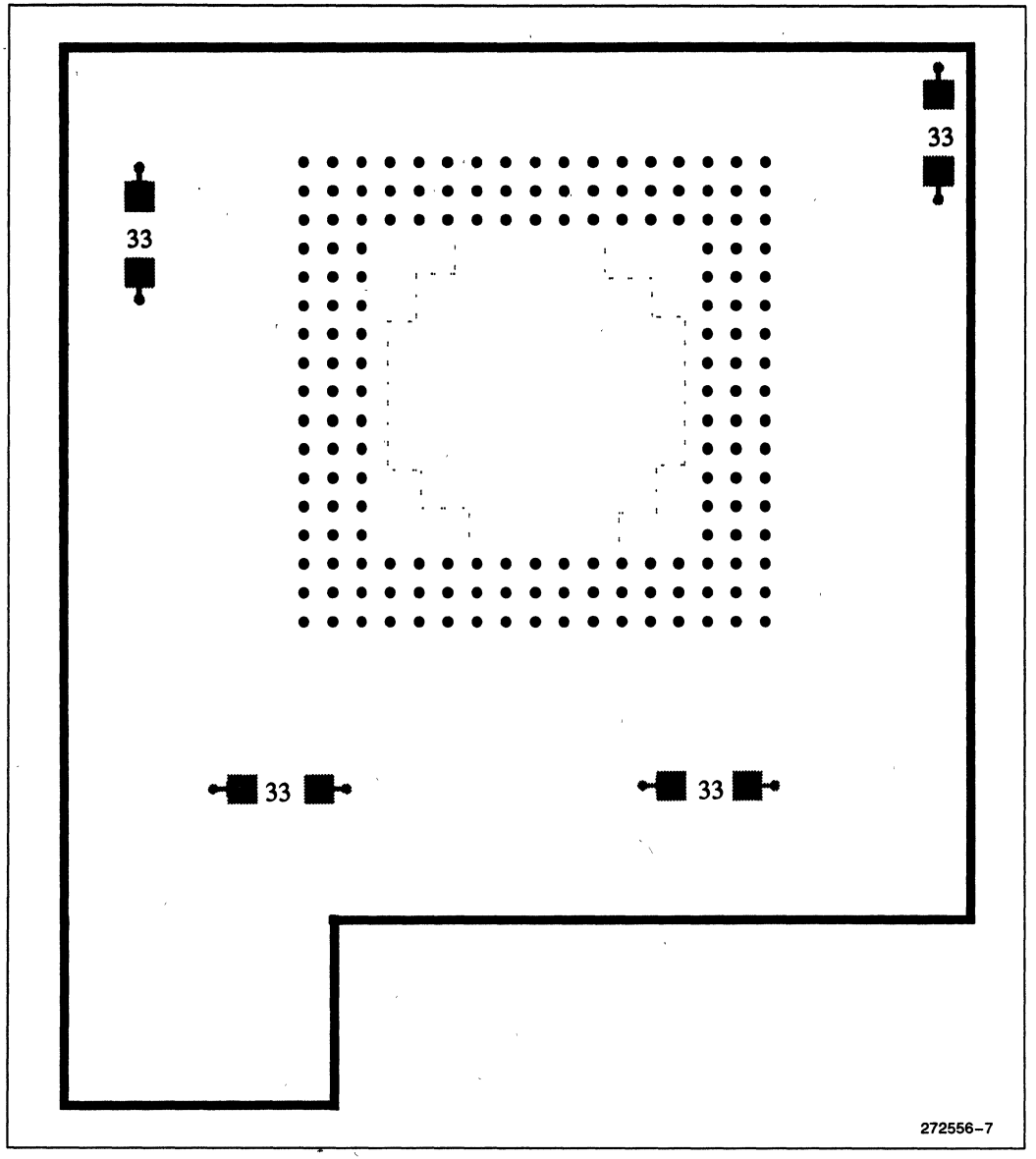


Figure 7. Recommended Bulk Decoupling Capacitor Values and Locations

BYTE ENABLE SIGNALS

The i960 Cx processors always perform aligned accesses on the bus. This means that the byte enable signals are limited to the following combinations.

In addition to the accesses that the Cx performs, the Hx issues three unaligned cases when accessing 32-bit memory regions.

80960Hx-ready systems must be designed to support all encodings. This is accomplished by ensuring that the memory write-enable signals for each byte are dependent on that byte's corresponding BE signal — not on a certain combination of byte enables. When accessing 16- or 8-bit regions, the Hx and Cx processors behave the same.

INTERRUPT SAMPLING

80960Hx-ready systems should be designed to produce asynchronous interrupts to the CPU. Synchronous systems such as lock-step multi-processor systems must meet input setup and hold times on the rising edges of CLKIN for the Hx, and on the falling edges of PCLK2:1 for the Cx. Interrupt pins are sampled on the rising edge of CLKIN for the Hx. Contrarily, on the Cx processors these pins are sampled on the falling edge of CLKIN. The actual sampling of the interrupt pins occurs once every two CLKIN cycles. Improper system behavior occurs if these setup and hold times are not met in a synchronous system. An example of this is the loosing synchronous operation of multiple processors.

PARITY

A 80960Hx-ready system can implement parity when a Hx is in the CPU socket. Parity is disabled while a Cx is in the CPU socket.

Five parity pins are added to the Hx. Four of these pins, labeled DP3:0, provide byte parity for data and possess the same timing as D31:0. The fifth pin is an output labeled PCHK#. It is asserted if a parity error is detected on reads. PCHK# is asserted in the clock, following the data cycle which has incorrect parity. The Hx DP3:0 pins correspond to the CA/CF's "no connect" pins. The Hx PCHK# pin corresponds to the

Table 1. Byte Enable Signal Combinations

Access	BE3#	BE2#	BE1#	BE0#
WORD	0	0	0	0
SHORT	1	1	0	0
SHORT	0	0	1	1
BYTE	1	1	1	0
BYTE	1	1	0	1
BYTE	1	0	1	1
BYTE	0	1	1	1

Table 2. Unaligned Cases When Accessing 32-Bit Memory Regions

Access	BE3#	BE2#	BE1#	BE0#
Unaligned Three-byte	1	0	0	0
Unaligned Three-byte	0	0	0	1
Unaligned SHORT	1	0	0	1

Table 3. Bus Access

Cycle Type	ADS#	CT3:0
Program initiated access using 8-bit bus	0	0000
Program initiated access using 16-bit bus	0	0001
Program initiated access using 32-bit bus	0	0010
Event initiated access using 8-bit bus	0	0100
Event initiated access using 16-bit bus	0	0101
Event initiated access using 32-bit bus	0	0110
Reserved	0	0X11
Reserved	0	1XXX
Processor not halted	1	0X0X0
Processor not halted	1	0X10
Reserved	1	0011
Processor in HALT mode	1	0111
Reserved for future products	1	1XXX

DACK0# pin on the CA/CF. Pull-up resistors are recommended on DP3:0. These resistors are required if parity is not being used to put the Hx parity inputs to a known state. They are also required if parity is being used when a Cx is in the system, in order to provide valid logic levels for the external parity logic. External logic will detect PCHK# high when a Cx processor is in a system. This disables external parity reporting logic.

Parity is only checked on bytes which possess a corresponding active BE signal.

CYCLE TYPE

An 80960Hx-ready system should not use cycle type pins, nor should it use DMA. The Hx uses the pins which correspond to the Cx EOP#/TC# pins for CT3:0. When ADS# is not active, the cycle type is driven to indicate whether it is executing or is in HALT mode. When ADS# is active, CT3:0 indicate the type of bus access currently being started.

BSTALL

The BSTALL signal becomes active when the Hx processor can not continue execution until a pending bus transaction is completed. A load instruction followed by an instruction that uses the result of the load, causes a stall until the load is completed. A store or a load instruction, issued when the bus queues are full, also cause a stall. In this case the Hx is stalled until a bus queue entry becomes available. One of these becomes available as a result of processing a pending bus request. The instruction scheduler can cause BSTALL when the processor fetches instructions from external memory. The processor must fetch these instructions due to instruction cache misses.

The BSTALL pin can be used to provide "on demand" bus arbitration. When a system has an external bus master which is given higher priority than the Hx, it can maintain ownership of the bus until the Hx needs the bus. The Hx will assert BREQ when it has a pending bus request. When BREQ is asserted without BSTALL, the processor can continue operation even in

the presence of a pending bus request. Some systems may choose to ignore this condition. Alternatively, they don't give the bus to the Hx, but instead wait until the processor is stalled. The assertion of BSTALL informs the arbitration logic of this condition.

The Cx processor does not have a BSTALL pin — the corresponding pin on the Cx is the DMA# pin. It will be driven high during normal operation (no DMA). This signals a stall condition to the external logic.

If BSTALL is used for bus arbitration in a 80960Hx-ready system, the recommendation is to logically "and" BSTALL and BREQ to indicate when the microprocessor requires the bus. By qualifying BSTALL with BREQ, the resulting signal can be used interchangeably between the Cx and Hx processors. This resulting signal is equivalent to BSTALL on a 80960Hx system, and equivalent to BREQ on a 80960Cx system.

JTAG

If boundary scan is used in an 80960Hx-ready system, a jumper should be used to connect TDI to the next device in the scan chain when a Cx is installed. The jumper should isolate the pin corresponding to TDO from the scan chain.

The Hx supports IEEE 1149.1 boundary scan. This interface consists of 5 pins: 4 input pins and 1 output pin. The JTAG interface utilizes pins used for DMA on the Cx. The Hx JTAG input pins correspond to CA/CF DREQ3:0# input pins. The JTAG output pin corresponds to a Cx DACK1# output pin.

RESERVED MEMORY

The Hx processor is not able to access external memory in the range 0xff000000 to 0xffffffff. This area is reserved for memory mapped registers. Consequently, an Hx processor cannot access the IBR of a Cx system located at 0xffffff00. The IBR of an Hx processor is located at 0xfefffff30 through 0xfefffff5f. It may be beneficial to use a single memory area mapped to two different areas.

For a system to be capable of using memory for either Hx or Cx boot up, at either 0xfexxxxxx or 0xffxxxxxx, address bit 24 should not be used in the boot area decode logic. Using this methodology, the Cx processor accesses this memory using addresses such as 0xffffff00, while the Hx processor uses addresses like 0xfefffff30.

AC TIMING

The timing of signals on the Hx differs from corresponding timing on the Cx. In general, the Hx is faster than the Cx. This generates some interesting design requirements for systems which accept either processor. Specifications for both implementations must be considered — the worst-case numbers must be used in the design.

The Hx specifications described in this section are estimates, and are subject to change when silicon becomes available.

REFERENCE CLOCK

The Cx AC timings for input and output signals are measured against the transitions of the output PCLK2:1 signals. When operating in 1x clock mode, the Cx processor input and output clocks are synchronized. TCP, the CLKIN to PCLK2:1 delay, is ± 2ns in 1x mode. When operating in 2x mode, the output clock edges are delayed from the input clocks. In 2x mode, T_{CP} is 2 to 25ns at 33 MHz.

The Hx has no output clocks; Hx AC timings are specified according to the input clock.

One of two clocking methods are recommended for a 80960Hx-ready system:

- External logic can be clocked with PCLK2:1 when a Cx is plugged into the socket. It can be clocked with CLKIN when using a Hx processor.
- Always use CLKIN to clock external logic for either a Cx or Hx processor.

For the first of the above recommendations, a method of clock selection must be implemented. Jumpers can be used to select either CLKIN or PCLK2:1 (to route to the synchronous logic within the system).

This is a simple methodology because the clocked logic performs the same function with either processor. One benefit derived from this is that the clocks used by external logic are always the processor's reference clocks.

CLKIN is the reference for the Hx; for the Cx it may be away from the reference (PCLK2:1) by as much as 2ns in 1x clock mode, or 25ns in 2x clock mode. This offset must be considered when analyzing system timing. Due to the wide range of possible delays, it is not practical to use 2x clock mode when using CLKIN for the external logic. The Hx does not support a 2x clock input.

INPUT/OUTPUT TIMING

Input pins specify setup and hold times according to the processor reference clock.

Input signals must be stable between the minimum input setup and hold times. This is the time when the signals are being latched internally within the processor. For minimum input setup and hold values, use the figures with the largest **maximum** values between the two devices.

AC timing parameters for output signals include both a minimum output hold time and a maximum output valid delay. The minimum output hold time specifies the time after a clock during which a signal continues to be valid from the previous state. The maximum output valid delay specifies the maximum time necessary for a signal to switch states.

Output signals switch between the minimum output hold and the maximum output valid times. For minimum output hold, the smallest **minimum** value of the different devices should be used. Maximum output valid delay is the largest **maximum** value of the different devices.

The combined specification for AC timings differs from the sole specification of either a Cx or Hx processor. An application which accepts either a Cx or Hx must operate over this wider range of timing. For example, pins D31:0 are bi-directional and require both input and output timing analysis. The AC timings used in this example are subject to change; refer to current data sheet for actual values.

During a write cycle:

- A CF outputs data within a 13 ns window — between 3 and 16 ns after the corresponding clock edge.
- A Hx outputs data within a 7 ns window — between 1.5 and 8.5 ns.

The combination of these specifications leads to a 14.5 ns window — between 1.5 and 16 ns. Minimum output hold (T_{OH}) analysis must be performed using 1.5 ns. This is the worst case time. Maximum output delay (T_{OV}) analysis must be performed using 16 ns, worst case. Similar "widening" of specifications also occur on input timings.

		D0 (80960CF)	D0 (80960Hx)	Combined
Output timing	TOH (min)	3 ns	1.5 ns	1.5 ns
	TOV (max)	16 ns	8.5 ns	16 ns
Input timing	TIS (min)	3 ns	5 ns	5 ns
	TIH (min)	5	1.5	5

PINOUT

The following table highlights the differences between the Hx and Cx processors. Differences are indicated with a heavier line around the table cell. Table 4 shows the pin differences between the Cx and Hx. The "comments" section describes recommended usage in an 80960Hx-ready system.

PGA Pin	Hx Signal Name	Cx Signal Name	PGA Pin	Hx Signal Name	Cx SignaL Name	PGA Pin	Hx Signal Name	Cx Signal Name	PGA Pin	Hx Signal Name	Cx Signal Name
A1	VSS	NC	C9	VSS	VSS	J15	VSS	VSS	Q10	VSS	VSS
A2	FAIL#	FAIL#	C10	VSS	VSS	J16	VCC	VCC	Q11	VSS	VSS
A3	DP0	NC	C11	VSS	VSS	J17	A10	A10	Q12	SUP#	SUP#
A4	DP2	NC	C12	VSS	VSS	K1	D13	D13	Q13	A30	A30
A5	VOLDET	NC	C13	CLKIN	CLKIN	K2	VCC	VCC	Q14	A28	A28
A6	TRST#	DREQ1#	C14	VCC	CLKMODE	K3	VSS	VSS	Q15	A24	A24
A7	TDI	DREQ3#	C15	XINT4#	XINT4#	K15	VSS	VSS	Q16	A21	A21
A8	TDO	DACK1#	C16	XINT6#	XINT6#	K16	VCC	VCC	Q17	A18	A18
A9	NC	DACK2#	C17	XINT7#	XINT7#	K17	A11	A11	R1	D24	D24
A10	NC	DACK3#	D1	D5	D5	L1	D15	D15	R2	D27	D27
A11	CT0	EOP/TC0#	D2	D2	D2	L2	D14	D14	R3	D31	D31
A12	CT1	EOP/TC1#	D3	NC	NC	L3	VSS	VSS	R4	BTERM#	BTERM#
A13	CT2	EOP/TC2#	D15	NMI#	NMI#	L15	VSS	VSS	R5	HOLD	HOLD
A14	CT3	EOP/TC3#	D16	A2	A2	L16	A13	A13	R6	ADS#	ADS#
A15	XINT1#	XINT1#	D17	A3	A3	L17	A12	A12	R7	VCC	VCC
A16	RESET#	RESET#	E1	D7	D7	M1	D16	D16	R8	VCC	VCC
A17	XINT2#	XINT2#	E2	D4	D4	M2	VCC	VCC	R9	BE0#	BE0#
B1	BOFF#	BOFF#	E3	D0	D0	M3	VSS	VSS	R10	VCC	VCC
B2	STEST	STEST	E15	VCC	VCC	M15	VSS	VSS	R11	VCC	VCC
B3	DP1	NC	E16	A4	A4	M16	VCC	VCC	R12	BSTALL	DMA#
B4	DP3	NC	E17	A5	A5	M17	A14	A14	R13	BREQ	BREQ
B5	TCK	DREQ0#	F1	D8	D8	N1	D17	D17	R14	A29	A29
B6	TMS	DREQ2#	F2	D6	D6	N2	D18	D18	R15	A26	A26
B7	VCC	VCC	F3	VCC	VCC	N3	VCC	VCC	R16	A23	A23
B8	PCHK#	DACK0#	F15	VSS	VSS	N15	VCC	VCC	R17	A22	A22
B9	VCC	VCC	F16	VCC	VCC	N16	A16	A16	S1	D25	D25
B10	VCCPLL	VCCPLL	F17	A6	A6	N17	A15	A15	S2	D29	D29
B11	VCC	VCC	G1	D9	D9	P1	D19	D19	S3	READY#	READY#
B12	VCC	VCC	G2	VCC	VCC	P2	D20	D20	S4	HOLDA	HOLDA
B13	NC	PCLK2	G3	VSS	VSS	P3	D22	D22	S5	BE3#	BE3#
B14	NC	PCLK1	G15	VSS	VSS	P15	A20	A20	S6	BE2#	BE2#
B15	XINT0#	XINT0#	G16	A7	A7	P16	A19	A19	S7	BE1#	BE1#
B16	XINT3#	XINT3#	G17	A8	A8	P17	A17	A17	S8	BLAST#	BLAST#
B17	XINT5#	XINT5#	H1	D11	D11	Q1	D21	D21	S9	DEN#	DEN#
C1	D3	D3	H2	D10	D10	Q2	D23	D23	S10	W/R#	W/R#
C2	D1	D1	H3	VSS	VSS	Q3	D26	D26	S11	DT/R#	DT/R#
C3	ONCE#	ONCE#	H15	VSS	VSS	Q4	D28	D28	S12	WAIT#	WAIT#
C4	VSS	NC	H16	VCC	VCC	Q5	D30	D30	S13	D/C#	D/C#
C5	VCC5	NC	H17	A9	A9	Q6	VCC	VCC	S14	LOCK#	LOCK#
C6	VCC	VCC	J1	D12	D12	Q7	VSS	VSS	S15	A31	A31
C7	VSS	VSS	J2	VCC	VCC	Q8	VSS	VSS	S16	A27	A27
C8	VSS	VSS	J3	VSS	VSS	Q9	VSS	VSS	S17	A25	A25

Table 4. 80960Cx/80960Hx Pin Differences

Pin	CA/CF	Hx	80960Hx-ready System
A1	NC	VSS	Connect to VSS.
A3	NC	DP0	Should be pulled up with a resistor. In a system with parity, connect to the parity bit which corresponds to D7:0.
A4	NC	DP2	Should be pulled up with a resistor. In a system with parity, connect to the parity bit which corresponds to D23:16.
A5	NC	VOLDET	Can be used to detect which processor is in the socket. High impedance - CA/CF. VSS - Hx
A6	DREQ1#	TRST#	When active (low), causes TAP controller (IEEE 1149.1) to go to Test_Logic_Reset state. This pin should be pulled high when not in use.
A7	DREQ3#	TDI	OK to pull-up or drive. If it is driven low, be sure DMA is disabled. If JTAG is used with a Cx in the system, this signal should be connected to TDI of next device in the chain, via a jumper.
A8	DACK1#	TDO	For Cx, this pin will always be high when DMA is not in use. Because this pin is an output, use a jumper or external logic to disconnect this pin when using the Cx.
A9	DACK2#	NC	No connection
A10	DACK3#	NC	No connection
A11–A14	EOP/TC#3:0	CT3:0	Use pull-ups. An output of 1111 indicates a Cx processor is in the system. Indicates the cycle type if ADS# is active. Indicates when the processor is halted if ADS# is not active.
B3	NC	DP1	Should be pulled up with a resistor. In a system with parity, connect to the parity bit which corresponds to D15:8.
B4	NC	DP3	Should be pulled up with a resistor. In a system with parity, connect to the parity bit which corresponds to D31:28.
B5	DREQ0#	TCK	Connect to Test Clock of 1149.1 interface. This pin should be pulled high when not in use.
B6	DREQ2#	TMS	Connect to Test Mode Select of 1149.1 interface. This pin should be pulled high when not in use.
B8	DACK0#	PCHK#	Connect to external parity error recovery/reporting logic. Cx will not generate or check parity.
B13	PCLK2	NC	OK to drive Hx with CLKIN for compatibility.
B14	PCLK1	NC	OK to drive Hx with CLKIN for compatibility.
C4	NC	V_{SS}	Connect to V_{SS}.
C5	NC	VCC5	Connect to 5V through a 100 Ohm resistor if inputs can be driven from 5V logic. Connect directly to 3.3V if inputs are not driven by 5V logic.
C14	CLKMODE	V_{CC}	Connect to processor's V_{CC}.
R12	DMA#	BSTALL	Can use for arbitration.

DESIGN GUIDELINE SUMMARY

A system can be designed which accepts either a Hx or Cx processor. The following items summarize the guidelines discussed in this paper:

- Don't use the DMA controller on the Cx processor
- Isolate V_{CC} for the CPU. Hx = 3.3V; Cx = 5V
- Provide 5V reference voltage for Hx (VCC5)
- Use CLKIN for system timing
- Combine AC specifications for timing analysis
- Accommodate new BE3:0# encodings
- Use pull-up resistors on parity signals
- Connect additional V_{SS} signals
- If using JTAG boundary scan, bypass Cx in the JTAG chain

intel®

2

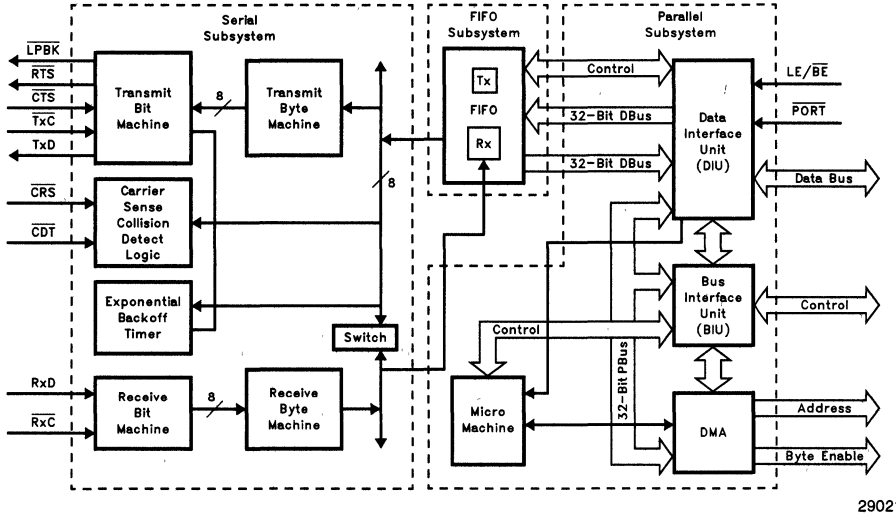
Memories and Peripherals

82596CA
HIGH-PERFORMANCE 32-BIT LOCAL AREA NETWORK COPROCESSOR

- **Performs Complete CSMA/CD Medium Access Control (MAC) Functions—Independently of CPU**
  - **— IEEE 802.3 (EOC) Frame Delimiting**
  - **— HDLC Frame Delimiting**
- **Supports Industry Standard LANs**
  - **— IEEE TYPE 10BASE-T, IEEE TYPE 10BASE5 (Ethernet*), IEEE TYPE 10BASE2 (Cheapernet), IEEE TYPE 1BASE5 (StarLAN), and the Proposed Standard 10BASE-F**
  - **— Proprietary CSMA/CD Networks Up to 20 Mb/s**
- **On-Chip Memory Management**
  - **— Automatic Buffer Chaining**
  - **— Buffer Reclamation after Receipt of Bad Frames; Optional Save Bad Frames**
  - **— 32-Bit Segmented or Linear (Flat) Memory Addressing Formats**
- **Network Management and Diagnostics**
  - **— Monitor Mode**
  - **— 32-Bit Statistical Counters**
- **82586 Software Compatible**
- **Self-Test Diagnostics**
- **Optimized CPU Interface**
  - **— Optimized Bus Interface to Intel's i486™DX, i486™SX, i487™SX and 80960CA Processors**
  - **— 33 MHz, 25 MHz, 20 MHz and 16 MHz Clock Frequencies**
  - **— Supports Big Endian and Little Endian Byte Ordering**
- **32-Bit Bus Master Interface**
  - **— 106 MB/s Bus Bandwidth**
  - **— Burst Bus Transfers**
  - **— Bus Throttle Timers**
  - **— Transfers Data at 100% of Serial Bandwidth**
  - **— 128-Byte Receive FIFO, 64-Byte Transmit FIFO**
- **Configurable Initialization Root for Data Structures**
- **High-Speed, 5V, CHMOS** IV Technology**
- **132-Pin Plastic Quad Flat Pack (PQFP) and PGA Package**

  (See Packaging Spec Order No. 240800-001, Package Type KU and A)

i486 is a trademark of Intel Corporation.
*Ethernet is a registered trademark of Xerox Corporation.
**CHMOS is a patented process of Intel Corporation.



290218-1

Figure 1. 82596CA Block Diagram

82596CA High-Performance 32-Bit Local Area Network Coprocessor

CONTENTS

CONTENTS

INTRODUCTION

The 82596CA is an intelligent, high-performance 32-bit Local Area Network coprocessor. The 82596CA implements the CSMA/CD access method and can be configured to support all existing IEEE 802.3 standards—TYPEs 10BASE-T, 10BASE5, 10BASE2, 1BASE5, and 10BROAD36. It can also be used to implement the proposed standard TYPE 10BASE-F. The 82596CA performs high-level commands, command chaining, and interprocessor communications via shared memory, thus relieving the host CPU of many tasks associated with network control. All time-critical functions are performed independently of the CPU, this increases network performance and efficiency. The 82596CA bus interface is optimized for Intel's i486™SX, i486™DX, i487™SX, 80960CA, and 80960KB processors.

The 82596CA implements all IEEE 802.3 Medium Access Control and channel interface functions, these include framing, preamble generation and stripping, source address generation, destination address checking, short-frame detection, and automatic length-field handling. Data rates up to 20 Mb/s are supported.

The 82596CA provides a powerful host system interface. It manages memory structures automatically, with command chaining and bidirectional data chaining. An on-chip DMA controller manages four channels, this allows autonomous transfer of data blocks (buffers and frames) and relieves the CPU of byte transfer overhead. Buffers containing errored or collided frames can be automatically recovered without CPU intervention. The 82596CA provides an upgrade path for existing 82586 software drivers by providing an 82586-software-compatible mode that supports the current 82586 memory structure. The 82586CA also has a Flexible memory structure and a Simplified memory structure. The 82596CA can address up to 4 gigabytes of memory. The 82596CA supports Little Endian and Big Endian byte ordering.

The 82596CA bus interface can achieve a burst transfer rate of 106 MB/s at 33 MHz. The bus interface employs bus throttle timers to regulate 82596CA bus use. Two large, independent FIFOs—128 bytes for Receive and 64 bytes for Transmit—tolerate long bus latencies and provide programmable thresholds that allow the user to optimize bus overhead for any worst-case bus latency. The high-performance bus is capable of back-to-back transmission and reception during the IEEE 802.3 9.6-μs Interframe Spacing (IFS) period.

The 82596CA provides a wide range of diagnostics and network management functions, these include internal and external loopback, exception condition tallies, channel activity indicators, optional capture of all frames regardless of destination address (promiscuous mode), optional capture of errored or collided frames, and time domain reflectometry for locating fault points on the network cable. The statistical counters, in 32-bit segmented and linear modes, are 32-bits each and include CRC errors, alignment errors, overrun errors, resource errors, short frames, and received collisions. The 82596CA also features a monitor mode for network analysis. In this mode the 82596CA can capture status bytes, and update statistical counters, of frames monitored on the link without transferring the contents of the frames to memory. This can be done concurrently while transmitting and receiving frames destined for that station.

The 82596CA can be used in both baseband and broadband networks. It can be configured for maximum network efficiency (minimum contention overhead) with networks of any length. Its highly flexible CSMA/CD unit supports address field lengths of zero through six bytes—configurable to either IEEE 802.3/Ethernet or HDLC frame delimitation. It also supports 16- or 32-bit cyclic redundancy checks. The CRC can be transferred directly to memory for receive operations, or dynamically inserted for transmit operations. The CSMA/CD unit can also be configured for full duplex operation for high throughput in point-to-point connections.

The 82596 C-step incorporates several new features not found in previous steppings. The following is a summary of the 82596 C-step's new features.

- The 82596 C-step fixes Errata found in the A1 and B steppings.
- The 82596 C-step has improved AC timings over both the A and B steppings.
- The 82596 C-step has a New Enhanced Big Endian Mode where in Linear Addressing Mode, true 32-bit Big Endian functionality is achieved. New Enhanced Big Endian Mode is enabled by setting bit 7 of the SYSBUS byte. This mode is software compatible with the big endian mode of the B-step with one exception—no 32-bit addresses need to be swapped by software in the C-step. In this new mode, the 82596 C-step treats 32-bit address pointers as true 32-bit entities and the SCB absolute address and statistical counters are still treated as two 16-bit big endian entities. Not setting this mode will configure the 82596 C-step to be 100% compatible to the A1-step big endian mode.
- The 82596 C-step is hardware and software compatible to both the A1 and B steppings allowing for easy "drop-in" to current designs. Pinout and control structures remain unchanged.

The 82596CA is fabricated with Intel's reliable, 5-V, CHMOS IV (process 648.8) technology. It is available in a 132-pin PQFP or PGA package.

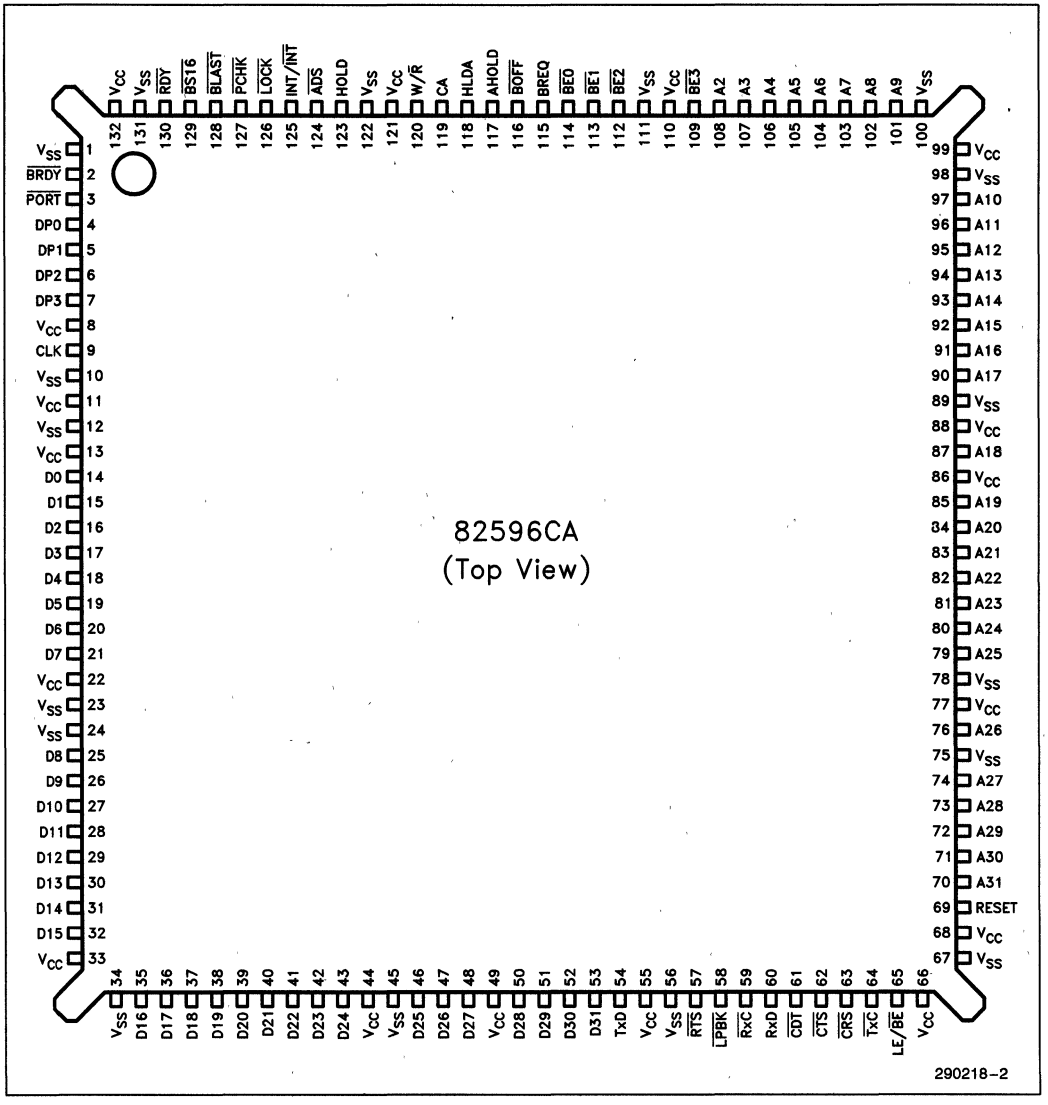


Figure 2. 82596CA PQFP Pin Configuration

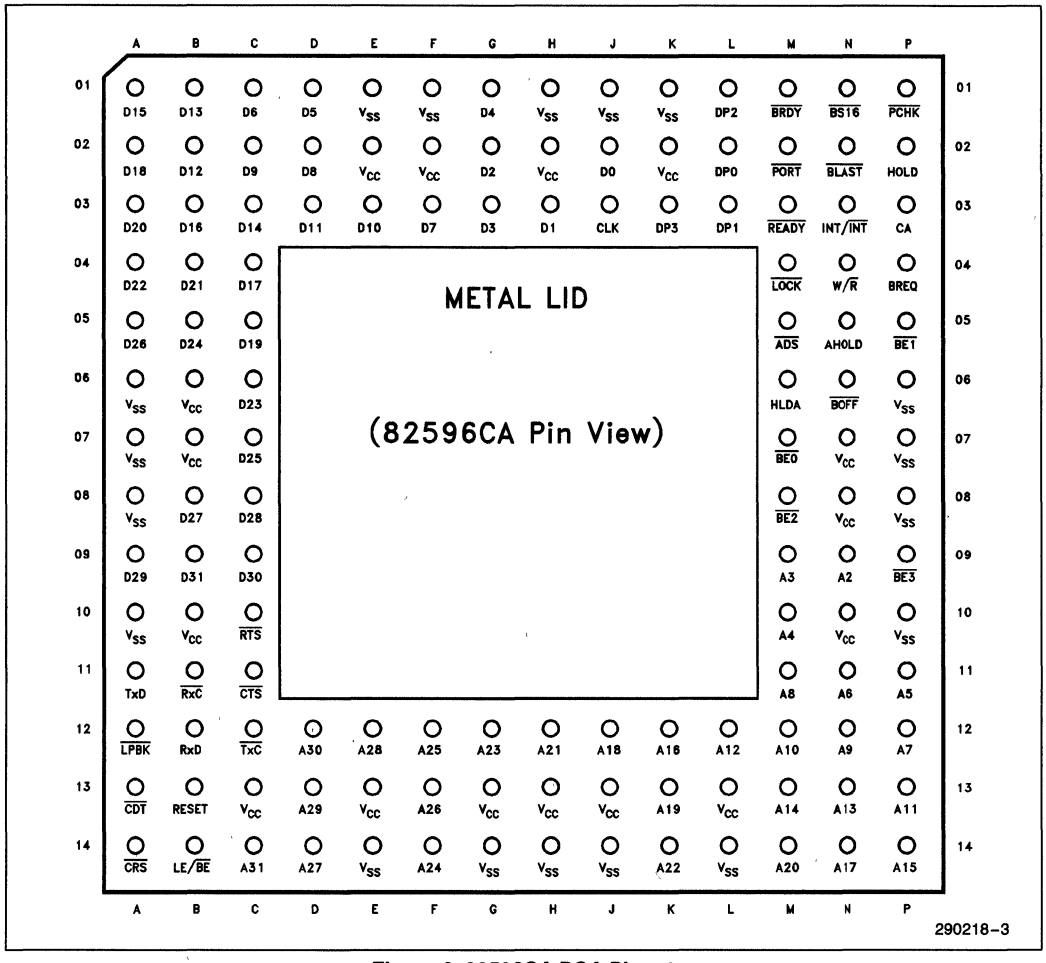


Figure 3. 82596CA PGA Pinout

82596CA PGA Cross Reference by Pin Name

Address		Data		Control		Serial Interface		V_{CC}	V_{SS}
Signal	Pin No.	Signal	Pin No.	Signal	Pin No.	Signal	Pin No.	Pin No.	Pin No.
A2	N9	D0	J2	$\overline{ADS}$	M5	$\overline{CDT}$	A13	B6	A6
A3	M9	D1	H3	AHOLD	N5	$\overline{CRS}$	A14	B7	A7
A4	M10	D2	G2	$\overline{BE0}$	M7	$\overline{CTS}$	C11	B10	A8
A5	P11	D3	G3	$\overline{BE1}$	P5	$\overline{LPBK}$	A12	C13	A10
A6	N11	D4	G1	$\overline{BE2}$	M8	$\overline{RTS}$	C10	E2	E1
A7	P12	D5	D1	$\overline{BE3}$	P9	$\overline{RxC}$	B11	E13	E14
A8	M11	D6	C1	$\overline{BLAST}$	N2	$\overline{RxD}$	B12	F2	F1
A9	N12	D7	F3	$\overline{BOFF}$	N6	$\overline{TxC}$	C12	G13	G14
A10	M12	D8	D2	$\overline{BRDY}$	M1	TxD	A11	H2	H1
A11	P13	D9	C2	BREQ	P4			H13	H14
A12	L12	D10	E3	$\overline{BS16}$	N1			J13	J1
A13	N13	D11	D3	CA	P3			K2	J14
A14	M13	D12	B2	CLK	J3			L13	K1
A15	P14	D13	B1	DP0	L2			N7	L14
A16	K12	D14	C3	DP1	L3			N8	P6
A17	N14	D15	A1	DP2	L1			N10	P7
A18	J12	D16	B3	DP3	K3				P8
A19	K13	D17	C4	HLDA	M6				P10
A20	M14	D18	A2	HOLD	P2				
A21	H12	D19	C5	INT/$\overline{INT}$	N3				
A22	K14	D20	A3	LE/$\overline{BE}$	B14				
A23	G12	D21	B4	$\overline{LOCK}$	M4				
A24	F14	D22	A4	$\overline{PCHK}$	P1				
A25	F12	D23	C6	$\overline{PORT}$	M2				
A26	F13	D24	B5	$\overline{READY}$	M3				
A27	D14	D25	C7	RESET	B13				
A28	E12	D26	A5	W/$\overline{R}$	N4				
A29	D13	D27	B8						
A30	D12	D28	C8						
A31	C14	D29	A9						
		D30	C9						
		D31	B9						

PIN DESCRIPTIONS

Symbol	PQFP Pin No.	Type	Name and Function
CLK	9	I	**CLOCK.** The system clock input provides the fundamental timing for the 82596. It is a 1X CLK input used to generate the 82596 clock and requires TTL levels. All external timing parameters are specified in reference to the rising edge of CLK.
D0–D31	14–53	I/O	**DATA BUS.** The 32 Data Bus lines are bidirectional, tri-state lines that provide the general purpose data path between the 82596 and memory. With the 82596 the bus can be either 16 or 32 bits wide; this is determined by the $\overline{BS16}$ signal. The 82596 always drives all 32 data lines during Write operations, even with a 16-bit bus. D31 – D0 are floated after a Reset or when the bus is not acquired. These lines are inputs during a CPU Port access; in this mode the CPU writes the next address to the 82596 through the data lines. During $\overline{PORT}$ commands (Relocatable SCP, Self-Test, Reset and Dump) the address must be aligned to a 16-byte boundary. This frees the D_3–D_0 lines so they can be used to distinguish the commands. The following is a summary of the decoding data.

D0	D1	D2	D3	D31–D4	Function
0	0	0	0	0000	Reset
0	1	0	0	ADDR	Relocatable SCP
1	0	0	0	ADDR	Self-Test
1	1	0	0	ADDR	Dump Command

DP0–DP3	4–7	I/O	**DATA PARITY.** These are tri-stated data parity pins. There is one parity line for each byte of the data bus. The 82596 drives them with even-parity information during write operations having the same timing as data writes. Likewise, even-parity information, with the same timing as read information, must be driven back to the 82596 over these pins to ensure that the correct parity check status is indicated by the 82596.
$\overline{PCHK}$	127	O	**PARITY CHECK.** This pin is driven high one clock after $\overline{RDY}$ to inform Read operations of the parity status of data sampled at the end of the previous clock cycle. When driven low it indicates that incorrect parity data has been sampled. It only checks the parity status of enabled bytes, which are indicated by the Byte Enable and Bus Size signals. $\overline{PCHK}$ is only valid for one clock time after data read is returned to the 82596; i.e., it is inactive (high) at all other times.
A31-A2	70–108	O	**ADDRESS LINES.** These 30 tri-stated Address lines output the address bits required for memory operation. These lines are floated after a Reset or when the bus is not acquired.
$\overline{BE3}$–$\overline{BE0}$	109–114	O	**BYTE ENABLE.** These tri-stated signals are used to indicate which bytes are involved with the current memory access. The number of Byte Enable signals asserted indicates the physical size of the data being transferred (1, 2, 3, or 4 bytes). • $\overline{BE0}$ indicates D7–D0 • $\overline{BE1}$ indicates D15–D8 • $\overline{BE2}$ indicates D23–D16 • $\overline{BE3}$ indicates D31–D24 These lines are floated after a Reset or when the bus is not acquired.
W/$\overline{R}$	120	O	**WRITE/READ.** This dual function pin is used to distinguish Write and Read cycles. This line is floated after a Reset or when the bus is not acquired.

PIN DESCRIPTIONS (Continued)

Symbol	PQFP Pin No.	Type	Name and Function
$\overline{ADS}$	124	O	**ADDRESS STATUS.** The 82596 uses this tri-state pin to indicate to indicate that a valid bus cycle has begun and that A31–A2, $\overline{BE3}$–$\overline{BE0}$, and W/$\overline{R}$ are being driven. It is asserted during t1 bus states. This line is floated after a Reset or when the bus is not acquired.
$\overline{RDY}$	130	I	**READY.** Active low. This signal is the acknowledgment from addressed memory that the transfer cycle can be completed. When high, it causes wait states to be inserted. It is ignored at the end of the first clock of the bus cycle's data cycle. This active-low signal does not have an internal pull-up resistor. This signal must meet the setup and hold times to operate correctly.
$\overline{BRDY}$	2	I	**BURST READY.** Active low. Burst Ready, like $\overline{RDY}$, indicates that the external system has presented valid data on the data pins in response to a Read, or that the external system has accepted the 82596 data in response to a Write request. Also, like $\overline{RDY}$, this signal is ignored at the end of the first clock in a bus cycle. If the 82596 can still receive data from the previous cycle, ADS will not be asserted in the next clock cycle; however, Address and Byte Enable will change to reflect the next data item expected by the 82596. $\overline{BRDY}$ will be sampled during each succeeding clock and if active, the data on the pins will be strobed to the 82596 or to external memory (read/write). $\overline{BRDY}$ operates exactly like READY during the last data cycle of a burst sequence and during nonburstable cycles.
$\overline{BLAST}$	128	O	**BURST LAST.** A signal (active low) on this tri-state pin indicates that the burst cycle is finished and when $\overline{BRDY}$ is next returned it will be treated as a normal ready; i.e., another set of addresses will be driven with $\overline{ADS}$ or the bus will go idle. $\overline{BLAST}$ is not asserted if the bus is not acquired.
AHOLD	117	I	**ADDRESS HOLD.** This hold signal is active high, it allows another bus master to access the 82596 address bus. In a system where an 82596 and an i486 processor share the local bus, AHOLD allows the cache controller to make a cache invalidation cycle while the 82596 holds the address lines. In response to a signal on this pin, the 82596 immediately (i.e. during the next clock) stops driving the entire address bus (A31–A2); the rest of the bus can remain active. For example, data can be returned for a previously specified bus cycle during Address Hold. The 82596 will not begin another bus cycle while AHOLD is active.
$\overline{BOFF}$	116	I	**BACKOFF.** This signal is active low, it informs the 82596 that another bus master requires access to the bus before the 82596 bus cycle completes. The 82596 immediately (i.e. during the next clock) floats its bus. Any data returned to the 82596 while $\overline{BOFF}$ is asserted is ignored. $\overline{BOFF}$ has higher priority than $\overline{RDY}$ or $\overline{BRDY}$; if two such signals are returned in the same clock period, $\overline{BOFF}$ is given preference. The 82596 remains in Hold until $\overline{BOFF}$ goes high, then the 82596 resumes its bus cycle by driving out the address and status, and asserting $\overline{ADS}$.
$\overline{LOCK}$	126	O	**LOCK.** This tri-state pin is used to distinguish locked and unlocked bus cycles. $\overline{LOCK}$ generates a semaphore handshake to the CPU. $\overline{LOCK}$ can be active for several memory cycles, it goes active during the first locked memory cycle (t1) and goes inactive at the last locked cycle (t2). This line is floated after a Reset or when the bus is not acquired. $\overline{LOCK}$ can be disabled via the sysbus byte in software.

PIN DESCRIPTIONS (Continued)

Symbol	PQFP Pin No.	Type	Name and Function
$\overline{BS16}$	129	I	**BUS SIZE.** This signal allows the 82596CA to work with either 16- or 32-bit bytes. Inserting $\overline{BS16}$ low causes the 82596 to perform two 16-bit memory accesses when transferring 32-bit data. In little endian mode the D15–D0 lines are driven when $\overline{BS16}$ is inserted, in Big Endian mode the D31–D16 lines are driven.
HOLD	123	O	**HOLD.** The HOLD signal is active high, the 82596 uses it to request local bus mastership. In normal operation HOLD goes inactive before HLDA. The 82596 can be forced off the bus by deasserting HLDA or if the bus throttle timers expire.
HLDA	118	I	**HOLD ACKNOWLEDGE.** The HLDA signal is active high, it indicates that bus mastership has been given to the 82596. HLDA is internally synchronized; after HOLD is detected low, the CPU drives HLDA low. **NOTE:** *Do not connect HLDA to V_{CC}—it will cause a deadlock.* A user wanting to give the 82596 permanent access to the bus should connect HLDA to HOLD. If HLDA goes inactive before HOLD, the 82596 will release the bus (by deasserting HOLD) within a maximum of within a specified number of bus cycles as specified in the 82596 User's Manual.
BREQ	115	I	**BUS REQUEST.** This signal, when configured to an externally activated mode, is used to trigger the bus throttle timers.
$\overline{PORT}$	3	I	**PORT.** When this signal is received, the 82596 latches the data on the data bus into an internal 32-bit register. When the CPU is asserting this signal it can write into the 82596 (via the data bus). This pin must be activated twice during all CPU Port access commands.
RESET	69	I	**RESET.** This active high, internally synchronized signal causes the 82596 to terminate current activity. The signal must be high for at least five system clock cycles. After five system clock cycles and four $\overline{TxC}$ clock cycles the 82596 will execute a Reset when it receives a high RESET signal. When RESET returns to low the 82596 waits for the first CA signal and then begins the initialization sequence.
LE/$\overline{BE}$	65	I	**LITTLE ENDIAN/BIG ENDIAN.** This dual-function pin is used to select byte ordering. When LE/$\overline{BE}$ is high, little endian byte ordering is used; when low, big endian byte ordering is used for data in frames (bytes) and for control (SCB, RFD, CBL, etc).
CA	119	I	**CHANNEL ATTENTION.** The CPU uses this pin to force the 82596 to begin executing memory resident Command blocks. The CA signal is internally synchronized. The signal must be high for at least one system clock. It is latched internally on the high to low edge and then detected by the 82596. The first CA after a Reset forces the 82596 into the initialization sequence beginning at location 00FFFFF6h or an SCP address written to the 82596 using CPU Port access. All subsequent CA signals cause the 82596 to begin executing new command sequences from the SCB.
INT/$\overline{INT}$	125	O	**INTERRUPT.** A high signal on this pin notifies the CPU that the 82596 is requesting an interrupt. This signal is an edge triggered interrupt signal, and can be configured to be active high or low.

PIN DESCRIPTIONS (Continued)

Symbol	PQFP Pin No.	Type	Name and Function
V_{CC}	17 Pins		**POWER.** +5 V ±10%.
V_{SS}	17 Pins		**GROUND.** 0 V.
TxD	54	O	**TRANSMIT DATA.** This pin transmits data to the serial link. It is high when not transmitting.
$\overline{TxC}$	64	I	**TRANSMIT CLOCK.** This signal provides the fundamental timing for the serial subsystem. The clock is also used to transmit data synchronously on the TxD pin. For NRZ encoding, data is transferred to the TxD pin on the high to low clock transition. For Manchester encoding, the transmitted bit center is aligned with the low to high transition. Transmit clock must always be running for proper device operation.
$\overline{LPBK}$	58	O	**LOOPBACK.** This TTL-level control signal enables the loopback mode. In this mode serial data on the TxD input is routed through the 82C501 internal circuits and back to the RxD output without driving the transceiver cable. To enable this signal, both internal and external loopback need to be set with the Configure command.
RxD	60	I	**RECEIVE DATA.** This pin receives NRZ serial data only. It must be high when not receiving.
$\overline{RxC}$	59	I	**RECEIVE CLOCK.** This signal provides timing information to the internal shifting logic. For NRZ data the state of the RxD pin is sampled on the high to low transition of the clock.
$\overline{RTS}$	57	O	**REQUEST TO SEND.** When this signal is low the 82596 informs the external interface that it has data to transmit. It is forced high after a Reset or when transmission is stopped.
$\overline{CTS}$	62	I	**CLEAR TO SEND.** An active-low signal that enables the 82596 to send data. It is normally used as an interface handshake to $\overline{RTS}$. Asserting $\overline{CTS}$ high stops transmission. $\overline{CTS}$ is internally synchronized. If $\overline{CTS}$ goes inactive, meeting the setup time to the $\overline{TxC}$ negative edge, the transmission will stop and $\overline{RTS}$ will go inactive within, at most, two $\overline{TxC}$ cycles.
$\overline{CRS}$	63	I	**CARRIER SENSE.** This signal is active low, it is used to notify the 82596 that traffic is on the serial link. It is only used if the 82596 is configured for external Carrier Sense. In this configuration external circuitry is required for detecting traffic on the serial link. $\overline{CRS}$ is internally synchronized. To be accepted, the signal must remain active for at least two serial clock cycles (for CRSF = 0).
$\overline{CDT}$	61	I	**COLLISION DETECT.** This active-low signal informs the 82596 that a collision has occurred. It is only used if the 82596 is configured for external Collision Detect. External circuitry is required for collision detection. $\overline{CDT}$ is internally synchronized. To be accepted, the signal must remain active for at least two serial clock cycles (for CDTF = 0).

82596 AND HOST CPU INTERACTION

The 82596CA and the host CPU communicate through shared memory. Because of its on-chip DMA capability, the 82596 can make data block transfers (buffers and frames) independently of the CPU; this greatly reduces the CPU byte transfer overhead.

The 82596 is a multitasking coprocessor that comprises two independent logical units—the Command Unit (CU) and the Receive Unit (RU). The CU executes commands from shared memory. The RU handles all activities related to frame reception. The independence of the CU and RU enables the 82596 to engage in both activities simultaneously—the CU can fetch and execute commands from memory while the RU is storing received frames in memory. The CPU is only involved with this process after the CU has executed a sequence of commands or the RU has finished storing a sequence of frames.

The CPU and the 82596 use the hardware signals Interrupt (INT) and Channel Attention (CA) to initiate communication with the System Control Block (SCB), see Figure 4. The 82596 uses INT to alert the CPU of a change in the contents of the SCB, the CPU uses CA to alert the 82596.

The 82596 has a CPU Port Access state that allows the CPU to execute certain functions without accessing memory. The 82596 $\overline{\text{PORT}}$ pin and data bus pins are used to enable this feature. The CPU can directly activate four operations when the 82596 is in this state.

- Write an alternative System Configuration Pointer (SCP). This can be used when the 82596 cannot use the default SCP address space.
- Write a different Dump Command Pointer and execute Dump. This can be used for troubleshooting No Response problems.
- The CPU can reset the 82596 via software without disturbing the rest of the system.
- A self-test can be used for board testing; the 82596 will execute a self-test and write the results to memory.

82596 BUS INTERFACE

The 82596CA has bus interface timings and pin definitions that are compatible with Intel's 32-bit i486™SX and i486™DX microprocessors. This eliminates the need for additional bus interface logic. Operating at 33 MHz, the 82596's bus bandwidth can be as high as 106 MB/s. Since Ethernet only requires 1.25 MB/s, this leaves a considerable amount of bandwidth for the CPU. The 82596 also has a bus throttle to regulate its use of the bus. Two timers can be programmed through the SCB: one controls the maximum time the 82596 can remain on the bus, the other controls the time the 82596 must stay off the bus (see Figure 5). The bus throttle can be programmed to trigger internally with HLDA or externally with BREQ. These timers can restrict the 82596 HOLD activation time and improve bus utilization.

2

82596 MEMORY ADDRESSING

The 82596 has a 32-bit memory address range, which allows addressing up to four gigabytes of memory. The 82596 has three memory addressing modes (see Table 1).

- **82586 Mode.** The 82596 has a 24-bit memory address range. The System Control Block, Command List, Receive Descriptor List, and Buffer Descriptors must reside in one 64-KB memory segment. Transmit and Receive buffers can reside in a 24-bit address space.
- **32-Bit Segmented Mode.** The 82596 has a 32-bit memory address range. The System Control Block, Command List, Receive Descriptor List, and Buffer Descriptors must reside in one 64-KB memory segment. Transmit and Receive buffers can reside in a 32-bit address space.
- **Linear Mode.** The 82596 has a 32-bit memory address range. Any memory structure can reside anywhere within the 32-bit memory address range.

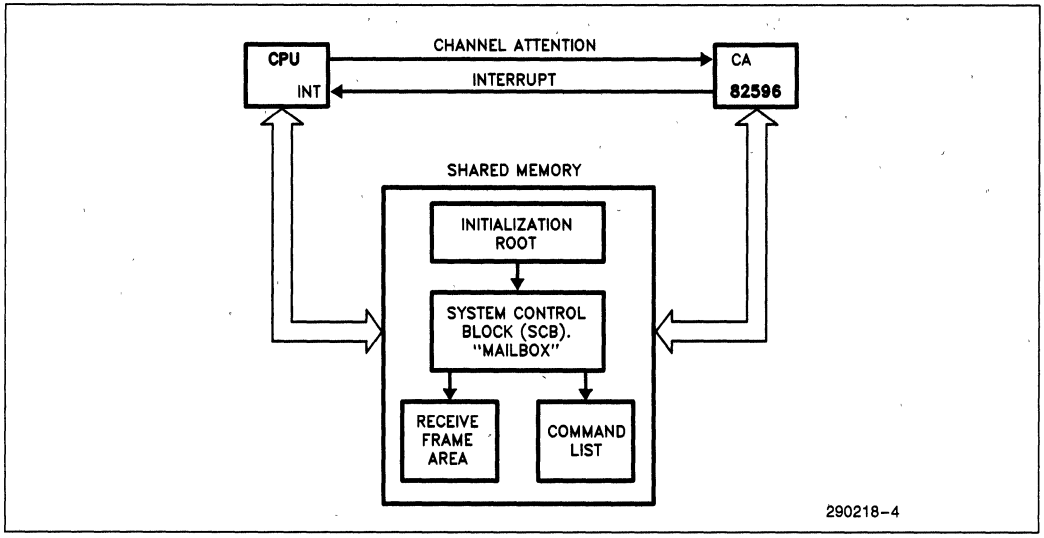


Figure 4. 82596 and Host CPU Intervention

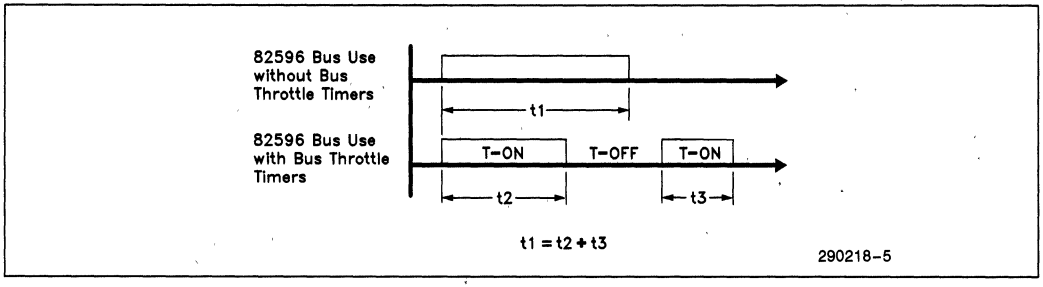


Figure 5. Bus Throttle Timers

Table 1. 82596 Memory Addressing Formats

Pointer or Offset	Operation Mode		
	82586	32-Bit Segmented	Linear
ISCP Address	24-Bit Linear	32-Bit Linear	32-Bit Linear
SCB Address	Base (24) + Offset (16)	Base (32) + Offset (16)	32-Bit Linear
Command Block Pointers	Base (24) + Offset (16)	Base (32) + Offset (16)	32-Bit Linear
Rx Frame Descriptors	Base (24) + Offset (16)	Base (32) + Offset (16)	32-Bit Linear
Tx Frame Descriptors	Base (24) + Offset (16)	Base (32) + Offset (16)	32-Bit Linear
Rx Buffer Descriptors	Base (24) + Offset (16)	Base (32) + Offset (16)	32-Bit Linear
Tx Buffer Descriptors	Base (24) + Offset (16)	Base (32) + Offset (16)	32-Bit Linear
Rx Buffers	24-Bit Linear	32-Bit Linear	32-Bit Linear
Tx Buffers	24-Bit Linear	32-Bit Linear	32-Bit Linear

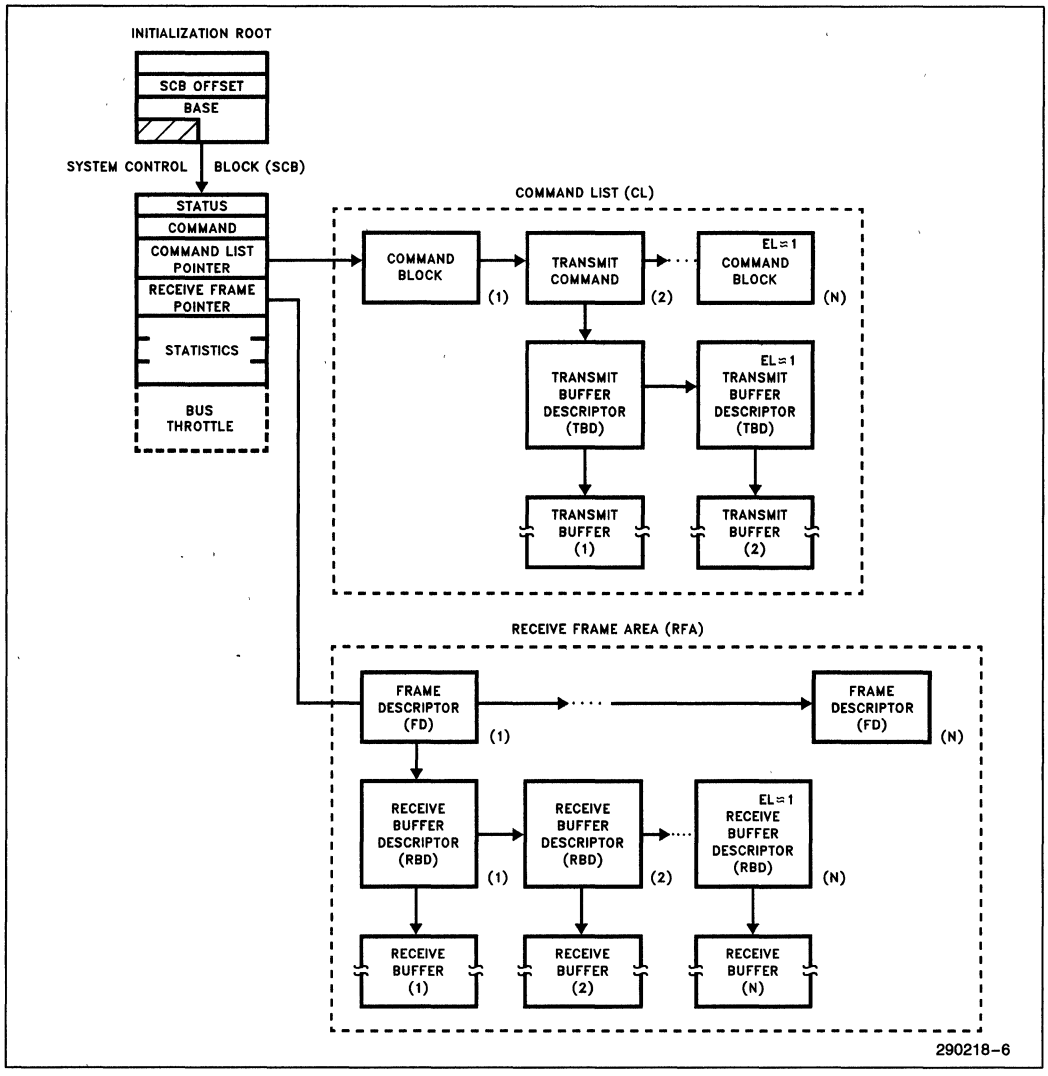


Figure 6. 82596 Shared Memory Structure

82596 SYSTEM MEMORY STRUCTURE

The Shared Memory structure consists of four parts: the Initialization Root, the System Control Block, the Command List, and the Receive Frame Area (see Figure 6).

The Initialization Root is in an established location known to the host CPU and the 82596 (00FFFFF6h). However, the CPU can establish the Initialization Root in another location by using the CPU Port access. This root is accessed during initialization, and points to the System Control Block.

The System Control Block serves as a bidirectional mail drop for the host CPU and the 82596 CU and RU. It is the central point through which the CPU and the 82596 exchange control and status information. The SCB has two areas. The first contains instructions from the CPU to the 82596. These include: control of the CU and RU (Start, Abort, Suspend, and Resume), a pointer to the list of CU commands, a pointer to the Receive Frame Area, a set of Interrupt Acknowledge bits, and the T-ON and T-OFF timers for the bus throttle. The second area contains status information the 82596 is sending to the CPU. Such as, the CU and RU states (Idle, Active

Ready, Suspended, No Receive Resources, etc.), interrupt bits (Command Completed, Frame Received, CU Not Ready, and RU Not Ready), and statistical counters.

The Command List functions as a program for the CU; individual commands are placed in memory units called Command Blocks (CBs). These CBs contain the parameters and status of specific high-level commands called Action Commands; e.g., Transmit or Configure.

Transmit causes the 82596 to transmit a frame. The Transmit CB contains the destination address, the length field, and a pointer to a list of linked buffers holding the frame that is to be constructed from several buffers scattered throughout memory. The Command Unit operates without CPU intervention; the DMA for each buffer, and the prefetching of references to new buffers, is performed in parallel. The CPU is notified only after a transmission is complete.

The Receive Frame Area is a list of Free Frame Descriptors (descriptors not yet used) and a list of user-prepared buffers. Frames arrive at the 82596 unsolicited; the 82596 must always be ready to receive and store them in the Free Frame Area. The Receive Unit fills the buffers when it receives frames, and reformats the Free Buffer List into received-frame structures. The frame structure is, for all practical purposes, identical to the format of the frame to be transmitted. The first Frame descriptor is referenced by the SCB. Unless the 82596 is configured to Save Bad Frames, the frame descriptor, and the associated buffer descriptor, which is wasted when a bad frame is received, are automatically reclaimed and returned to the Free Buffer List.

Receive buffer chaining (storing incoming frames in a linked buffer list) significantly improves memory utilization. Without buffer chaining, the user must allocate consecutive blocks of memory, each capable of containing a maximum frame (for Ethernet, 1518 bytes). Since an average frame is about 200 bytes, this is very inefficient. With buffer chaining, the user can allocate small buffers and the 82596 will only use those that are needed.

Figure 7 A–D illustrates how the 82596 uses the Receive Frame Area. Figure 7A shows an unused Receive Frame Area composed of Free Frame Descriptors and Free Receive Buffers prepared by the user. The SCB points to the first Frame Descriptor of the Frame Descriptor List. Figure 7B shows the same Receive Frame Area after receiving one frame. This first frame occupies two Receive Buffers and one Frame Descriptor—a valid received frame will only occupy one Frame Descriptor. After receiving this frame the 82596 sets the next Free Frame Descriptor RBD pointer to the next Free RBD. Figure 7C shows the RFA after receiving a second frame. In this example the second frame occupies only one Receive Buffer and one RFD. The 82596 again sets the RBD pointer. This process is repeated again in Figure 7D, showing the reception of another frame using one Receive Buffer; in this example there is an extra Frame Descriptor.

TRANSMIT AND RECEIVE MEMORY STRUCTURES

There are three memory structures for reception and transmission. The 82586 memory structure, the Flexible memory structure, and the Simplified memory structure. The 82586 mode is selected by configuring the 82596 during initialization. In this mode all the 82596 memory structures are compatible with the 82586 memory structures.

When the 82596 is not configured to the 82586 mode, the other two memory structures, Simplified and Flexible, are available for transmitting and receiving. These structures are selected by setting the S/F bit in the Transmit Command and/or the Receive Frame Descriptor (see Figures 29, 30, 41, and 42). It is recommended that any linked list of buffers be relegated to a single type—either simplified or flexible. The Simplified memory structure offers a simple structure for ease of programming (see Figure 8). All information about a frame is contained in one structure; for example, during reception the RFD and data field are contained in one structure.

The Flexible memory structure (see Figure 9) has a control field that allows the programmer to specify the amount of receive data the RFD will contain for receive operations and the amount of transmit data the Transmit Command Block will contain for transmit operations. For example, when the control field in the RFD is set to 20 bytes during a reception, the first 20 bytes of the data field are stored in the RFD (6 bytes of destination address, 6 bytes of source address, 2 bytes of length field, and 6 bytes of data) and the remainder of the data field is stored in the Receive Data Buffers. This is useful for capturing frame headers when header information is contained in the data field. The header information can then be automatically stored in the RFD partitioned from the Receive Data Buffer.

The control field can also be used for the Transmit Command when the Flexible memory structure is used. The quantity of data field bytes to be transmitted from the Transmit Command Block is specified by the variable control field.

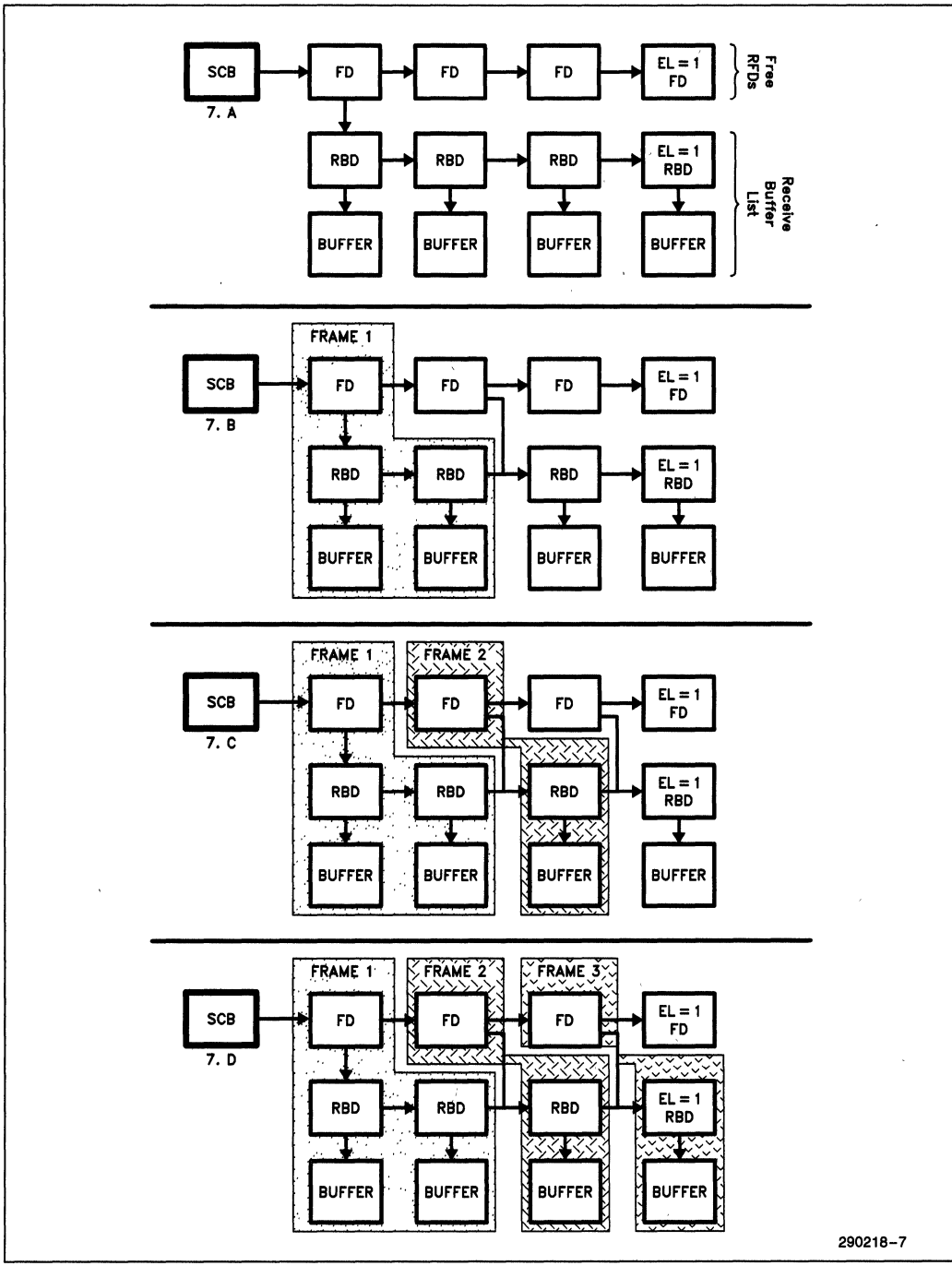


Figure 7. Frame Reception in the RFA

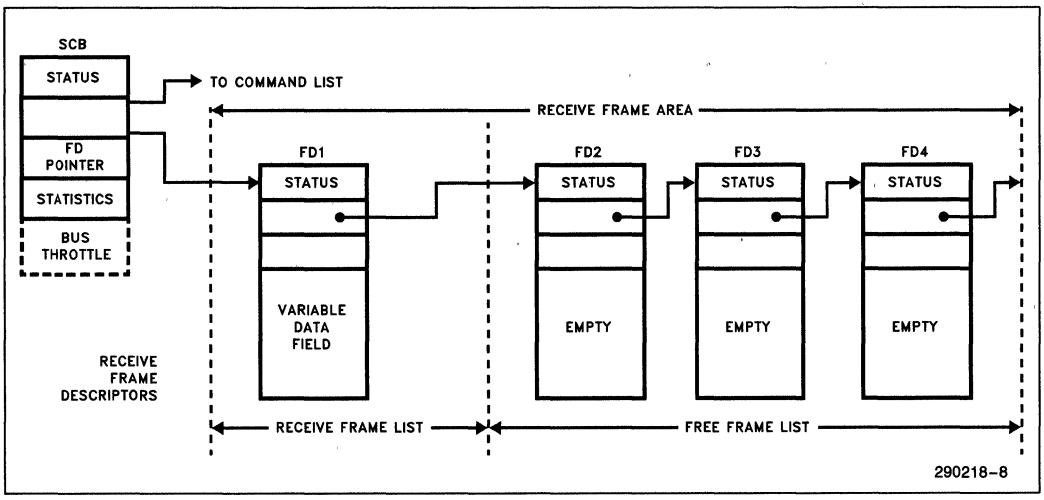


Figure 8. Simplified Memory Structure

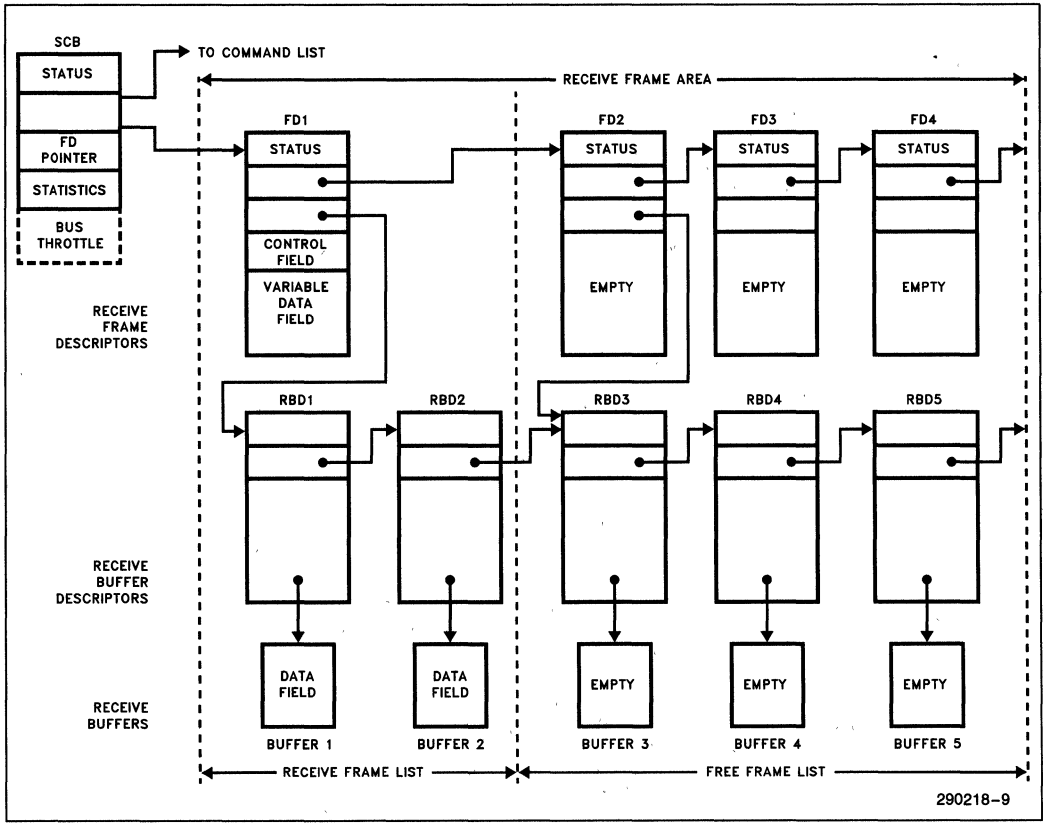


Figure 9. Flexible Memory Structure

TRANSMITTING FRAMES

The 82596 executes high-level Action Commands from the Command List in system memory. Action Commands are fetched and executed in parallel with the host CPU operation, thereby significantly improving system performance. The format of the Action Commands is shown in Figure 10. Figure 28 shows the 82586 mode, and Figures 29 and 30 show the command formats of the Linear and 32-bit Segmented modes.

A single Transmit command contains, as part of the command-specific parameters, the destination address and length field of the transmitted frame and a pointer to buffer area in memory containing the data portion of the frame. The data field is contained in a memory data structure consisting of a buffer descriptor (BD) and a data buffer—or a linked list of buffer descriptors and buffers—as shown in Figure 11.

Multiple data buffers can be chained together using the BDs. Thus, a frame with a long data field can be transmitted using several (shorter) data buffers chained together. This chaining technique allows the system designer to develop efficient buffer management.

The 82596 automatically generates the preamble (alternating 1s and 0s) and start frame delimiter, fetches the destination address and length field from the Transmit command, inserts its unique address as the source address, fetches the data field specified by the Transmit command, and computes and appends the CRC to the end of the frame (see Figure 12). In the Linear and 32-bit Segmented mode the CRC can be optionally inserted on a frame-by-frame basis by setting the NC bit in the Transmit Command Block (see Figures 29 and 30).

The 82596 can be configured to generate two types of start and end frame delimiters—End of Carrier (EOC) or HDLC. In EOC mode the start frame delimiter is 10101011 and the end frame delimiter is indicated by the lack of a signal after the last bit of the frame check sequence field has been transmitted. In EOC mode the 82596 can be configured to extend short frames by adding pad bytes (7Eh) during transmission, according to the length field. In HDLC mode the 82596 will generate the 01111110 flag for the start and end frame delimiters, and do standard bit stuffing and stripping. Furthermore, the 82596 can be configured to pad frames shorter than the specified minimum frame length by appending the appropriate number of flags to the end of the frame.

When a collision occurs, the 82596 manages the jam, random wait, and retry processes, reinitializing DMA pointers without CPU intervention. Multiple frames can be sent by linking the appropriate number of Transmit commands together. This is particularly useful when transmitting a message larger than the maximum frame size (1518 bytes for Ethernet).

2

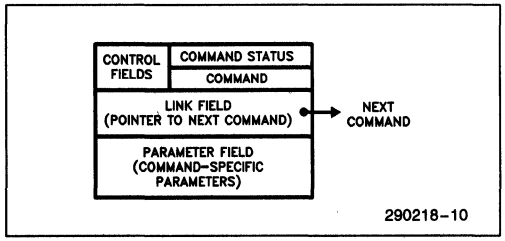


Figure 10. Action Command Format

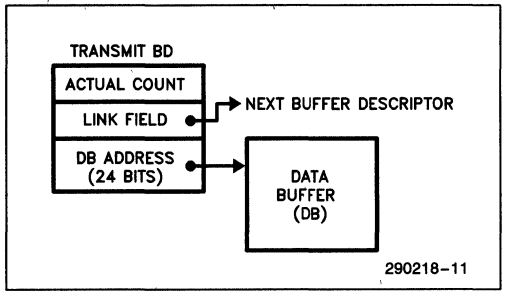


Figure 11. Data Buffer Descriptor and Data Buffer Structure

PREAMBLE	START FRAME DELIMITER	DESTINATION ADDRESS	SOURCE ADDRESS	LENGTH FIELD	DATA FIELD	FRAME CHECK SEQUENCE	END FRAME DELIMITER

Figure 12. Frame Format

RECEIVING FRAMES

To reduce CPU overhead, the 82596 is designed to receive frames without CPU supervision. The host CPU first sets aside an adequate receive buffer space and then enables the 82596 Receive Unit. Once enabled, the RU watches for arriving frames and automatically stores them in the Receive Frame Area (RFA). The RFA contains Receive Frame Descriptors, Receive Buffer Descriptors, and Data Buffers (see Figure 13). The individual Receive Frame Descriptors make up a Receive Descriptor List (RDL) used by the 82596 to store the destination and source addresses, the length field, and the status of each frame received (see Figure 14).

Once enabled, the 82596 checks each passing frame for an address match. The 82596 will recognize its own unique address, one or more multicast addresses, or the broadcast address. If a match is found the 82596 stores the destination and source addresses and the length field in the next available RFD. It then begins filling the next available Data Buffer on the FBL, which is pointed to by the current RFD, with the data portion of the incoming frame. As one Data Buffer is filled, the 82596 automatically fetches the next DB on the FBL until the entire frame is received. This buffer chaining technique is particularly memory efficient because it allows the system designer to set aside buffers to fit frames much shorter than the maximum allowable frame length. If AL-LOC = 1, or if the flexible memory structure is used, the addresses and length field can be placed in the Receive Buffer.

Once the entire frame is received without error, the 82596 does the following housekeeping tasks.

- The actual count field of the last Buffer Descriptor used to hold the frame just received is updated with the number of bytes stored in the associated Data Buffer.
- The next available Receive Frame Descriptor is fetched.
- The address of the next available Buffer Descriptor is written to the next available Receive Frame Descriptor.
- A frame received interrupt status bit is posted in the SCB.
- An interrupt is sent to the CPU.

If a frame error occurs, for example a CRC error, the 82596 automatically reinitializes its DMA pointers and reclaims any data buffers containing the bad frame. The 82596 will continue to receive frames without CPU help as long as Receive Frame Descriptors and Data Buffers are available.

82596 NETWORK MANAGEMENT AND DIAGNOSTICS

The behavior of data communication networks is normally very complex because of their distributed and asynchronous nature. It is particularly difficult to pinpoint a failure when it occurs. The 82596 has extensive diagnostic and network management functions that help improve reliability and testability. The 82596 reports on the following events after each frame is transmitted.

- Transmission successful.
- Transmission unsuccessful. Lost Carrier Sense.
- Transmission unsuccessful. Lost Clear to Send.
- Transmission unsuccessful. A DMA underrun occurred because the system bus did not keep up with the transmission.
- Transmission unsuccessful. The number of collisions exceeded the maximum allowed.
- Number of Collisions. The number of collisions experienced during transmission of the frame.
- Heartbeat Indicator. This indicates the presence of a heartbeat during the last Interframe Spacing (IFS) after transmission.

When configured to Save Bad Frames the 82596 checks each incoming frame and reports the following errors.

- CRC error. Incorrect CRC in a properly aligned frame.
- Alignment error. Incorrect CRC in a misaligned frame.
- Frame too short. The frame is shorter than the value configured for minimum frame length.
- Overrun. Part of the frame was not placed in memory because the system bus did not keep up with incoming data.
- Out of buffer. Part of the frame was discarded because of insufficient memory storage space.
- Receive collision. A collision was detected during reception and the destination address of the incoming frame matches the 82596 individual address. Collisions in the preamble are not counted.
- Length error. A frame not matching the frame length parameter was detected.

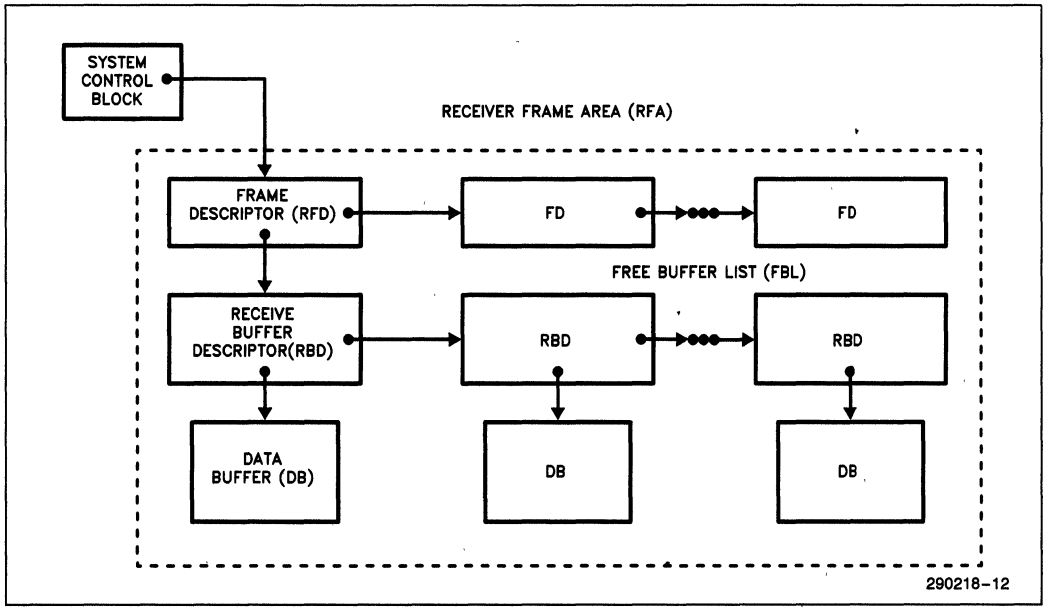


Figure 13. Receive Frame Area Diagram

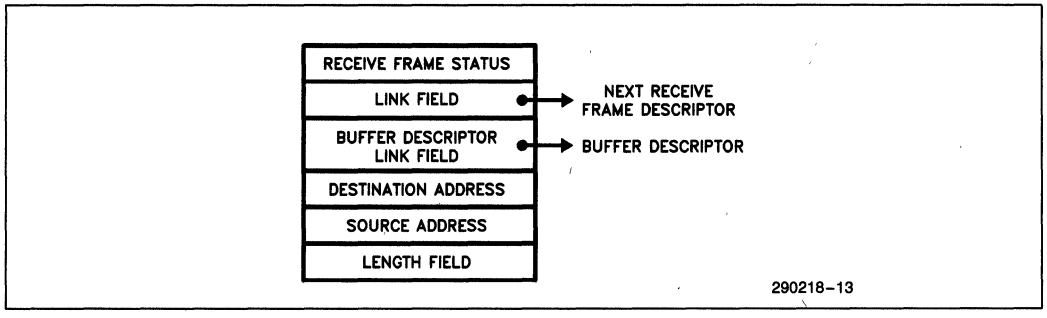


Figure 14. Receive Frame Descriptor

NETWORK PLANNING AND MAINTENANCE

To properly plan, operate, and maintain a communication network, the network management entity must accumulate information on network behavior. The 82596 provides a rich set of network-wide diagnostics that can serve as the basis for a network management entity.

Information on network activity is provided in the status of each frame transmitted. The 82596 reports the following activity indicators after each frame.

- Number of collisions. The number of collisions the 82596 experienced while attempting to transmit the frame.
- Deferred transmission. During the first transmission attempt the 82596 had to defer to traffic on the link.

The 82596 updates its 32-bit statistical counters after each received frame that both passes address filtering and is longer than the Minimum Frame Length configuration parameter. The 82596 reports the following statistics.

- CRC errors. The number of well-aligned frames that experienced a CRC error.
- Alignment errors. The number of misaligned frames that experienced a CRC error.
- No resources. The number of frames that were discarded because of insufficient resources for reception.
- Overrun errors. The number of frames that were not completely stored in memory because the system bus did not keep up with incoming data.
- Receive Collision counter. The number of collisions detected during receive. Collisions occurring before the minimum frame length will be counted as short frames. Collisions in the preamble will not be counted at all.
- Short Frame counter. The number of frames that were discarded because they were shorter than the configured minimum frame length.

Once again, these counters are not updated until the 82596 decodes a destination address match.

The 82596 can be configured to Promiscuous mode. In this mode it captures all frames transmitted on the network without checking the Destination Address. This is useful when implementing a monitoring station to capture all frames for analysis.

A useful method of capturing frame headers is to use the Simplified memory mode, configure the 82596 to Save Bad Frames, and configure the 82596 to Promiscuous mode with space in the RFD allocated for specific number of receive data bytes. The 82596 will receive all frames and put them in the RFD. Frames that exceed the available space in the RFD will be truncated, the status will be updated, and the 82596 will retrieve the next RFD. This allows the user to capture the initial data bytes of each frame (for instance, the header) and discard the remainder of the frame.

The 82596 also has a monitor mode for network analysis. During normal operation the receive function enables the 82596 to receive frames that pass address filtering. These frames must have the Start of Frame Delimiter (SFD) field and must be longer than the absolute minimum frame length of 5 bytes (6 bytes in case of Multicast address filtering). Contents and status of the received frames are transferred to memory. The monitor function enables the 82596 to simply evaluate the incoming frames. The 82596 can monitor the frames that pass or do not pass the address filtering. It can also monitor frames which do not have the SFD fields. The 82596 can be configured to only keep statistical information about monitor frames. Three options are available in the Monitor mode. These options are selected by the two monitor mode configuration bits available in the configuration command.

When the first option is selected, the 82596 receives good frames that pass address filtering and transfers them to memory while monitoring frames that do not pass address filtering or are shorter than the minimum frame size (these frames are not transferred to memory). When this option is used the 82596 updates six counters: CRC errors, alignment errors, no resource errors, overrun errors, short frames and total good frames received.

When the second option is selected, the receive function is completely disabled. The 82596 monitors only those frames that pass address filterings and meet the minimum frame length requirement. When this option is used the 82596 updates six counters: CRC errors, alignment errors, total frames (good and bad), short frames, collisions detected and total good frames.

When the third option is selected, the receive function is completely disabled. The 82596 monitors all frames, including frames that do not have a Start Frame Delimiter. When this option is used the 82596 updates six counters: CRC errors, alignment errors, total frames (good and bad), short frames, collisions detected and total good frames.

STATION DIAGNOSTICS AND SELF-TEST

The 82596 provides a large set of diagnostic and network management functions. These include internal and external loopback and time domain reflectometry for locating fault points in the network cable. The 82596 ensures software reliability by dumping the contents of the 82596 internal registers into system memory. The 82596 has a self-test mode that enables it to run an internal self-test and place the results in system memory.

82586 SOFTWARE COMPATIBILITY

The 82596 has a software-compatible state in which all its memory structures are compatible with the 82586 memory structure. This includes all the Action Commands, the Receive Frame Area (including the RFD, Buffer Descriptors, and Data Buffers), the System Control Block, and the initialization procedures. There are two minor differences between the 82596 in the 82586-Compatible memory structure and the 82586.

- When the internal and external loopback bits in the Configure command are set to 11 the 82596 is in external loopback and the $\overline{LPBK}$ pin is activated; in the 82586 this situation would produce internal loopback.
- During a Dump command both the 82596 and 82586 dump the same number of bytes; however, the data format is different.

INITIALIZING THE 82596

A Reset command is issued to the 82596 to prepare it for normal operation. The 82596 is initialized through two data structures that are addressed by two pointers, the System Configuration Pointer (SCP) and the Intermediate System Configuration Pointer (ISCP). The initialization procedure begins when a Channel Attention signal is asserted after RESET. The 82596 uses the address of the double word that contains the SCP as a default—00FFFFF4h. Before the CA signal is asserted this default address can be changed to any other available address by asserting the $\overline{PORT}$ pin and providing the desired address over the $D_{31}-D_4$ pins of the address bus. Pins D_3-D_0 must be 0010; i.e., any alternative address must be aligned to 16-byte boundaries. All addresses sent to the 82596 must be word aligned, which means that all pointers and memory structures must start on an even address (A_0 = zero).

SYSTEM CONFIGURATION POINTER (SCP)

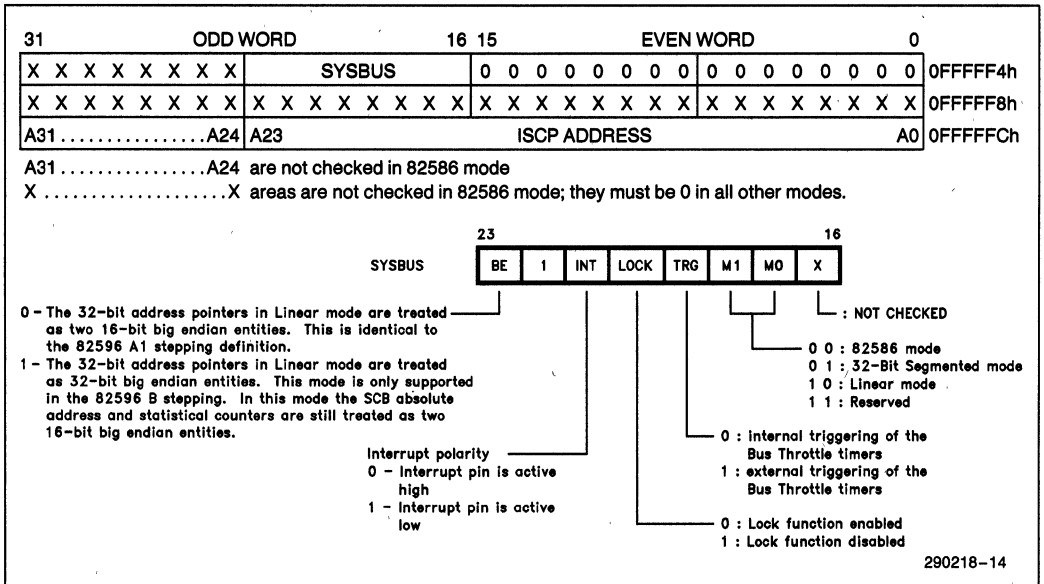
The SCP contains the sysbus byte and the location of the next structure of the initialization process, the ISCP. The following parameters are selected in the SYSBUS.

- The 82596 operation mode.
- The Bus Throttle timer triggering method.
- Lock enabled.
- Interrupt polarity.
- Big Endian 32-bit entity mode.

Byte ordering is determined by the LE/$\overline{BE}$ pin. LE/$\overline{BE}$ = 1 selects Little Endian byte ordering and LE/$\overline{BE}$ = 0 selects Big Endian byte ordering.

NOTE:
In the following, X indicates a bit not checked 82586 mode. This bit must be set to 0 in all other modes.

The following diagram illustrates the format of the SCP.



ISCP ADDRESS— The physical address of the ISCP. In the 82586 mode, bits A31–A24 are considered to be zero.

Figure 15. The System Configuration Pointer

Writing the Sysbus

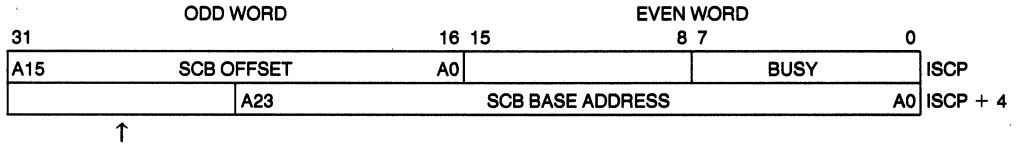
When writing the sysbus byte it is important to pay attention to the byte order.

- When a Little Endian processor is used, the sysbus byte is located at byte address 00FFFFF6h (or address $n+2$ if an alternative SCP address n was programmed).
- When a processor using Big Endian byte ordering is used, the sysbus, alternative SCP, and ISCP addresses will be different.
  - The sysbus byte is located at 00FFFFF5h.
  - If an alternative SCP address is programmed, the sysbus byte should be at byte address $n+1$.

INTERMEDIATE SYSTEM CONFIGURATION POINTER (ISCP)

The ISCP indicates the location of the System Control Block. Often the SCP is in ROM and the ISCP is in RAM. The CPU loads the SCB address (or an equivalent data structure) into the ISCP and asserts CA. This Channel Attention signal causes the 82596 to begin its initialization procedure and to get the SCB address from the ISCP and SCP. In 82586 and 32-bit Segmented modes the SCP base address is also the base address of all Command Blocks, Frame Descriptors, and Buffer Descriptors (but not buffers). All these data structures must reside in one 64-KB segment; however, in Linear mode no such limitation is imposed.

The following diagram illustrates the ISCP format.



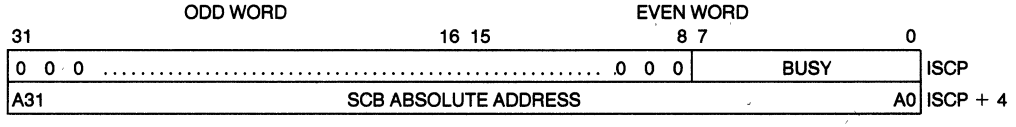
X X X X X X X X — in 82586 mode
A31 .....A24 — in 32-bit segmented mode.

BUSY — Indicates that the 82596 is being initialized. The CPU sets the ISCP to 01h before it gives the first CA to the 82596. The ISCP is cleared by the 82596 after the SCB base and offset are read. Note that the most significant byte of the first word of the ISCP is not modified when BUSY is cleared.

SCB OFFSET— This 16-bit quantity specifies the offset portion of the address of the SCB.

SCB BASE — Specifies the base portion of the address of the SCB. The base of SCB is also the base of all 82596 Command Blocks, Frame Descriptors and Buffer Descriptors. In the 82586 mode, bits A31–A24 are considered to be zero.

Figure 16. The Intermediate System Configuration Pointer—82586 and 32-Bit Segmented Modes



BUSY — Indicates that the 82596 is being initialized. The ISCP is set to 01h by the CPU before its first CA to the 82596. It is cleared by the 82596 after the SCB address is read.

SCB ADDRESS— This 32-bit quantity specifies the physical address of the SCB.

Figure 17. The Intermediate System Configuration Pointer—Linear Mode.

INITIALIZATION PROCESS

The CPU sets up the SCP, ISCP, and the SCB structures, and, if desired, an alternative SCP address. It also sets BUSY to 01h. The 82596 is initialized when a Channel Attention signal follows a Reset signal, causing the 82596 to access the System Configuration Pointer. The sysbus byte, the operational mode, the bus throttle timer triggering method, the interrupt polarity, and the state of $\overline{LOCK}$ are read. After reset the Bus Throttle timers are essentially disabled—the T-ON value is infinite, the T-OFF value is zero. After the SCP is read, the 82596 reads the ISCP and saves the SCB address. In 82586 and 32-bit Segmented modes this address is represented as a base address plus the offset (this base address is also the base address of all the control blocks). In Linear mode the base address is also an absolute address. The 82596 clears BUSY, sets CX and CNR to equal 1 in the SCB, clears the SCB command word, sends an interrupt to the CPU, and awaits another Channel Attention signal. RESET configures the 82596 to its default state before CA is asserted.

CONTROLLING THE 82596CA

The host CPU controls the 82596 with the commands, data structures, and methods described in this section. The CPU and the 82596 communicate through shared memory structures. The 82596 contains two independent units: the Command Unit and the Receive Unit. The Command Unit executes commands from the CPU, and the Receive Unit handles frame reception. These two units are controlled and monitored by the CPU through a shared memory structure called the System Control Block (SCB). The CPU and the 82596 use the CA and INT signals to communicate with the SCB.

82596 CPU ACCESS INTERFACE ($\overline{PORT}$)

The 82596 has a CPU access interface that allows the host CPU to do four things.

- Write an alternative System Configuration Pointer address.
- Write an alternative Dump area pointer and perform Dump.
- Execute a software reset.
- Execute a self-test.

The following events initiate the CPU access state.

- Presence of an address on the $D_{31}-D_4$ data bus pins.
- The D_3-D_0 pins are used to select one of the four functions.
- The $\overline{PORT}$ input pin is asserted, as in a regular write cycle.

NOTE.
The SCP Dump and Self-Test addresses must be 16-byte aligned.

The 82596 requires two 16-bit write cycles for a port command. The first write holds the internal machines and reads the first 16 bits; the second activates the $\overline{PORT}$ command and reads the second 16 bits.

The $\overline{PORT}$ Reset is useful when only the 82596 needs to be reset. The CPU must wait for 10-system and 5-serial clocks before issuing another CA to the 82596; this new CA begins a new initialization process.

The Dump function is useful for troubleshooting No Response problems. If the chip is in a No Response state, the $\overline{PORT}$ Dump operation can be executed and a $\overline{PORT}$ Reset can be used to reinitialize the 82596 without disturbing the rest of the system.

The Self-Test function can be used for board testing; the 82596 will execute a self-test and write the results to memory.

Table 2. $\overline{PORT}$ Function Selection

	D31 ..... D4			..... D0			
Function	**Addresses and Results**			**D3**	**D2**	**D1**	**D0**
Reset	A31	Don't Care	A4	0	0	0	0
Self-Test	A31	Self-Test Results Address	A4	0	0	0	1
SCP	A31	Alternative SCP Address	A4	0	0	1	0
Dump	A31	Dump Area Pointer	A4	0	0	1	1

MEMORY ADDRESSING FORMATS

The 82596 accesses memory by 32-bit addresses. There are two types of 32-bit addresses: linear and segmented. The type of address used depends on the 82596 operating mode and the type of memory structure it is addressing. The 82596 has three operating modes.

- 82586 Mode
  - A Linear address is a single 24-bit entity. Address pins $A_{31}-A_{24}$ are always zero.
  - A Segmented address uses a 24-bit base and a 16-bit offset.
- 32-bit Segmented Mode
  - A Linear address is a single 32-bit entity.
  - A Segmented address uses a 32-bit base and a 16-bit offset.

NOTE:
In the previous two memory addressing modes, each command header (CB, TBD, RFD, RBD, and SCB) must wholly reside within one segment. If the 82596 encounters a memory structure that does not follow this restriction, the 82596 will fetch the next contiguous location in memory (beyond the segment).

- Linear Mode
  - A Linear address is a single 32-bit entity.
  - There are no Segmented addresses.

Linear addresses are primarily used to address transmit and receive data buffers. In the 82586 and 32-bit Segmented modes, segmented addresses (base plus offset) are used for all Command Blocks, Buffer Descriptors, Frame Descriptors, and System Control Blocks. When using Segmented addresses, only the offset portion of the entity being addressed is specified in the block. The base for all offsets is the same—that of the SCB. See Table 1.

LITTLE ENDIAN AND BIG ENDIAN BYTE ORDERING

The 82596 supports both Little Endian and Big Endian byte ordering for its memory structures.

The 82596 A1 stepping supports Big Endian byte ordering for word and byte entities. Dword entities are not supported with 82596 A1 Big Endian byte ordering. This results in slightly different 82596A1 memory structures for Big Endian operation. These structures are defined in the *32-Bit LAN Components User's Manual.*

The 82596 B stepping supports Big Endian byte ordering for Linear mode only. All 82596 B 32-bit address pointers are treated as 32-bit Big Endian entities, however, the SCB absolute address and statistical counters are treated as two 16-bit Big Endian entities. This 32-bit Big Endian entity support is configured through bit 7 in the SYSBUS byte.

The 82596 C-step has a New Enhanced Big Endian Mode where in Linear Addressing mode, true 32-bit Big Endian functionality is achieved. New Enhanced Big Endian Mode is enabled exactly the same as the B-step, by setting bit 7 of the SYSBUS byte. This mode is software compatible with the big endian mode of the B-step with one exception—no 32-bit addresses need to be swapped by software in the C-step. In this new mode, the 82596 C-step treats 32-bit address pointers as true 32-bit entities and the SCB absolute address and statistical counters are still treated as two 16-bit big endian entities. Not setting this mode will configure the 82596 C-step to be 100% compatible to the A1-step big endian mode.

NOTE:
All 82596 memory entities must be word or dword aligned, except the transmit buffers can be byte aligned for the 82596 B or C-steppings.

An example of a dword entity is a frame descriptor command/status dword, whereas the raw data of the frame are byte entities. Both 32- and 16-bit buses are supported. When a 16-bit bus is used with Big Endian memory organization, data lines $D_{15}-D_0$ are used. The 82596 has an internal crossover that handles these swap operations.

COMMAND UNIT (CU)

The Command Unit is the logical unit that executes Action Commands from a list of commands very similar to a CPU program. A Command Block is associated with each Action Command. The CU is modeled as a logical machine that takes, at any given time, one of the following states.

- **Idle.** The CU is not executing a command and is not associated with a CB on the list. This is the initial state.
- **Suspended.** The CU is not executing a command; however, it is associated with a CB on the list.
- **Active.** The CU is executing an Action Command and pointing to its CB.

The CPU can affect CU operation in two ways: by issuing a CU Control Command or by setting bits in the Command word of the Action Command.

When programming the 82596 CU, it is important to consider the asynchronous way the 82596 processes commands. If a command is issued to the 82596 CU, it may be busy processing other commands. In order to avoid asynchronous race conditions, the following guidelines are recommended to the 82596 programmer:

- If the CU is already in the Active state, and another command needs to be executed, it is unwise to immediately issue another CU Start command. If a new command (or list of commands) needs to be started, first issue a CU Suspend command, wait for the CU to become Suspended, then issue the new CU Start. This will insure that all commands are processed correctly.
- In general, it is a good idea to make sure any CU command has been accepted and executed before issuing a new control command to the CU.

RECEIVE UNIT (RU)

The Receive Unit is the logical unit that receives frames and stores them in memory. The RU is modeled as a logical machine that takes, at any given time, one of the following states.

- **Idle.** The RU has no memory resources and is discarding incoming frames. This is the initial state.
- **No Resources.** The RU has no memory resources and is discarding incoming frames. This state differs from Idle in that the RU accumulates statistics on the number of discarded frames.
- **Suspended.** The RU has memory available for storing frames, but is discarding them. The suspend state can only be reached if the CPU forces this through the SCB or sets the suspend bit in the RFD.
- **Ready.** The RU has memory available and is storing incoming frames.

The CPU can affect RU operation in three ways: by issuing an RU Control Command, by setting bits in the Frame Descriptor Command word of the frame being received, or by setting the EL bit of the current buffer's Buffer Descriptor.

When programming the 82596 RU, it is important to consider the asynchronous way the 82596 processes receive frames. If an RU Start is issued to the 82596 RU, it may be busy processing other incoming packets. In order to avoid asynchronous race conditions, the following guidelines are recommended to the 82596 programmer:

- If the RU is already in the Ready state, and a new RFA is required to be started, it is unwise to immediately issue another RU Start command. If the new RFA needs to be started, first issue an RU Suspend command, wait for the RU to become Suspended, then issue the new RU Start. This will insure that all incoming frames are received correctly.
- In general, it is a good idea to make sure any RU command has been accepted and executed before issuing a new control command to the RU.

SYSTEM CONTROL BLOCK (SCB)

The SCB is a memory block that plays a major role in communications between the CPU and the 82596. Such communications include the following.

- Commands issued by the CPU
- Status reported by the 82596

Control commands are sent to the 82596 by writing them into the SCB and then asserting CA. The 82596 examines the command, performs the required action, and then clears the SCB command word. Control commands perform the following types of tasks.

- Operation of the Command Unit (CU). The SCB controls the CU by specifying the address of the Command Block List (CBL) and by starting, suspending, resuming, or aborting execution of CBL commands.
- Operation of the Bus Throttle. The SCB controls the Bus Throttle timers by providing them with new values and sending the Load and Start timer commands. The timers can be operated in both the 32-bit Segmented and Linear modes.
- Reception of frames by the Receive Unit (RU). The SCB controls the RU by specifying the address of the Receive Frame Area and by starting, suspending, resuming, or aborting frame reception.
- Acknowledgment of events that cause interrupts.
- Resetting the chip.

The 82596 sends status reports to the CPU via the System Control Block. The SCB contains four types of status reports.

- The cause of the current interrupts. These interrupts are caused by one or more of the following 82596 events.
  - The Command Unit completes an Action Command that has its I bit set.
  - The Receive Unit receives a frame.
  - The Command Unit becomes inactive.
  - The Receive Unit becomes not ready.
- The status of the Command Unit.
- The status of the Receive Unit.
- Status reports from the 82596 regarding reception of corrupted frames.

Events can be cleared only by CPU acknowledgment. If some events are not acknowledged by the ACK field the Interrupt signal (INT) will be reissued after Channel Attention (CA) is processed. Furthermore, if a new event occurs while an interrupt is set, the interrupt is temporarily cleared to trigger edge-triggered interrupt controllers.

The CPU uses the Channel Attention line to cause the 82596 to examine the SCB. This signal is trailing-edge triggered—the 82596 latches CA on the trailing edge. The latch is cleared by the 82596 before the SCB control command is read.

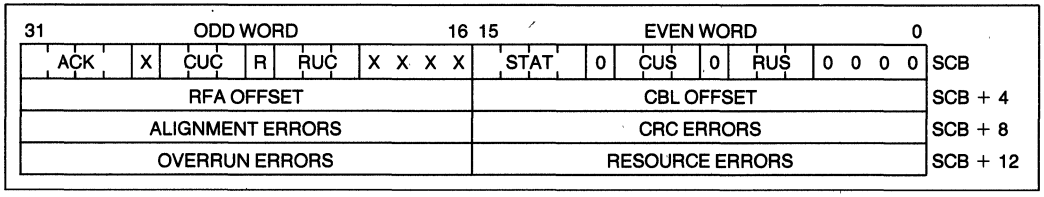


Figure 18. SCB—82586 Mode

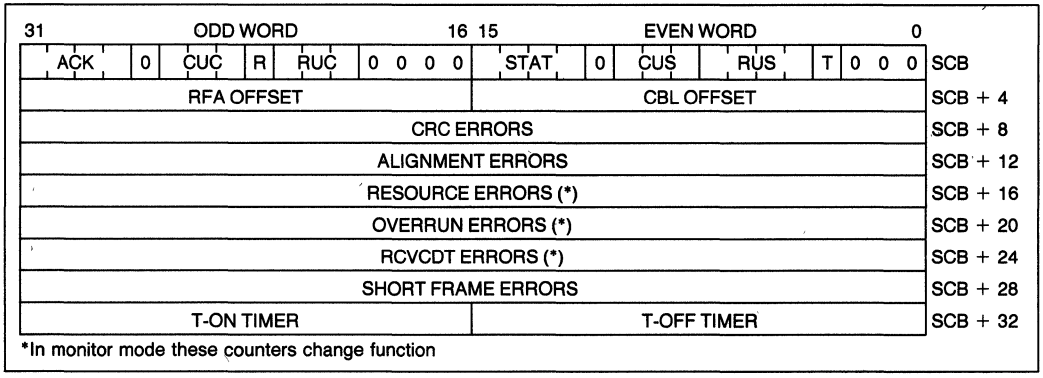


Figure 19. SCB—32-Bit Segmented Mode

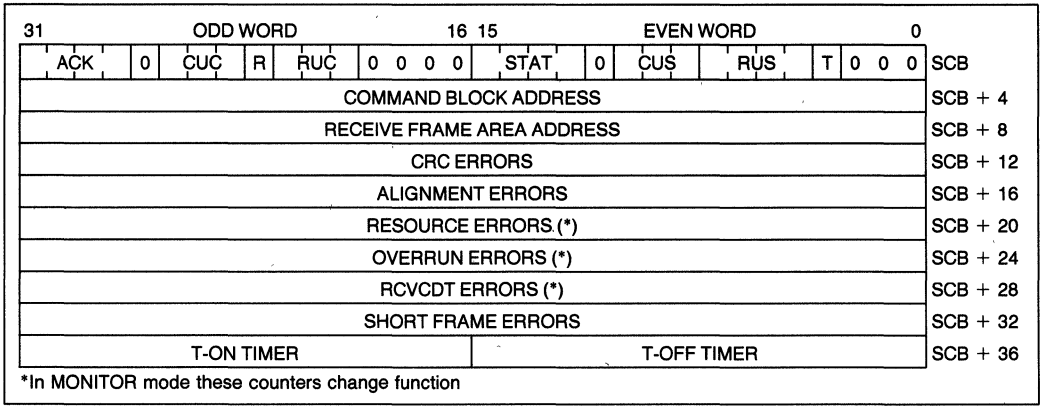
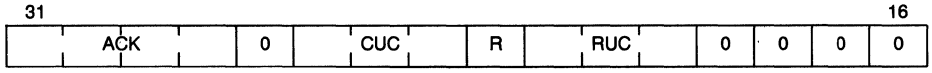


Figure 20. SCB—Linear Mode

Command Word



These bits specify the action to be performed as a result of a CA. This word is set by the CPU and cleared by the 82596. Defined bits are:

Bit 31 ACK-CX — Acknowledges that the CU completed an Action Command.

Bit 30 ACK-FR — Acknowledges that the RU received a frame.

Bit 29 ACK-CNA — Acknowledges that the Command Unit became not active.

Bit 28 ACK-RNR — Acknowledges that the Receive Unit became not ready.

Bits 24–26 CUC — (3 bits) This field contains the command to the Command Unit. Valid values are:

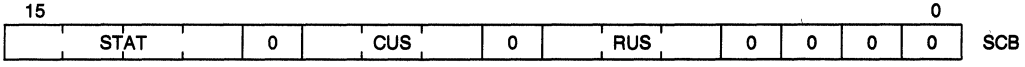
- 0 — NOP (does not affect current state of the unit).
- 1 — Start execution of the first command on the CBL. If a command is executing, complete it before starting the new CBL. The beginning of the CBL is in CBL OFFSET (address).
- 2 — Resume the operation of the Command Unit by executing the next command. This operation assumes that the Command Unit has been previously suspended.
- 3 — Suspend execution of commands on CBL after current command is complete.
- 4 — Abort current command immediately.
- 5 — Loads the Bus Throttle timers so they will be initialized with their new values after the active timer (T-ON or T-OFF) reaches Terminal Count. If no timer is active new values will be loaded immediately. This command is not valid in 82586 mode.
- 6 — Loads and immediately restarts the Bus Throttle timers with their new values. This command is not valid in 82586 mode.
- 7 — Reserved.

Bit 23 RESET — Reset chip (logically the same as hardware RESET).

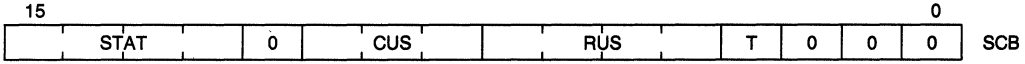
Bits 20–22 RUC — (3 bits) This field contains the command to the Receive Unit. Valid values are:

- 0 — NOP (does not alter current state of unit).
- 1 — Start reception of frames. The beginning of the RFA is contained in the RFA OFFSET (address). If a frame is being received complete reception before starting.
- 2 — Resume frame reception (only when in suspended state).
- 3 — Suspend frame reception. If a frame is being received complete its reception before suspending.
- 4 — Abort receiver operation immediately.
- 5–7 — Reserved.

Status Word



82586 mode



32-Bit Segmented and Linear mode.

Indicates the status of the 82596. This word is modified only by the 82596. Defined bits are:

Bit 15 CX — The CU finished executing a command with its I (interrupt) bit set.

Bit 14 FR — The RU finished receiving a frame.

Bit 13 CNA — The Command Unit left the Active state.

Bit 12 RNR — The Receive Unit left the Ready state.

Bits 8–10 CUS — (3 bits) This field contains the status of the command unit. Valid values are:

- 0 — Idle
- 1 — Suspended
- 2 — Active
- 3–7 — Not used

Bits 4–7 RUS — This field contains the status of the receive unit. Valid values are:

- 0h (0000) — Idle
- 1h (0001) — Suspended
- 2h (0010) — No Resources. This bit indicates both no resources due to lack of RFDs in the RDL and no resources due to lack of RBDs in the FBL.
- 4h (0100) — Ready
- Ah (1010) — No resources due to no more RBDs (not in the 82586 mode).
- Ch (1100) — No more RBDs (not in 82586 mode)

No other combinations are allowed

Bit 3 T — Bus Throttle timers loaded (not in 82586 mode).

SCB OFFSET ADDRESSES

CBL Offset (Address)

In 82586 and 32-bit Segmented modes this 16-bit quantity indicates the offset portion of the address for the first Command Block on the CBL. In Linear mode it is a 32-bit linear address for the first Command Block on the CBL. It is accessed only if CUC equals Start.

RFA Offset (Address)

In 82586 and 32-bit Segmented modes this 16-bit quantity indicates the offset portion of the address for the Receive Frame Area. In Linear mode it is a 32-bit linear address for the Receive Frame Area. It is accessed only if RUC equals Start.

SCB STATISTICAL COUNTERS

Statistical Counter Operation

- The CPU is responsible for clearing all error counters before initializing the 82596. The 82596 updates these counters by reading them, adding 1, and then writing them back to the SCB.
- The counters are wraparound counters. After reaching FFFFFFFFh the counters wrap around to zero.
- The 82596 updates the required counters for each frame. It is possible for more than one counter to be updated; multiple errors will result in all affected counters being updated.
- The 82596 executes the read-counter/increment/write-counter operation without relinquishing the bus (locked operation). This is to ensure that no logical contention exists between the 82596 and the CPU due to both attempting to write to the counters simultaneously. In the dual-port memory configuration the CPU should not execute any write operation to a counter if $\overline{LOCK}$ is asserted.
- The counters are 32-bits wide and their behavior is fully compatible with the IEEE 802.3 standard. The 82596 supports all relevant statistics (mandatory, optional, and desired) through the status of the transmit and receive header and directly through SCB statistics.

CRCERRS

This 32-bit quantity contains the number of aligned frames discarded because of a CRC error. This counter is updated, if needed, regardless of the RU state.

ALNERRS

This 32-bit quantity contains the number of frames that both are misaligned (i.e., where $\overline{CRS}$ deasserts on a nonoctet boundary) and contain a CRC error. The counter is updated, if needed, regardless of the RU state.

SHRTFRM

This 32-bit quantity contains the number of received frames shorter than the minimum frame length.

The last three counters change function in monitor mode.

RSCERRS

This 32-bit quantity contains the number of good frames discarded because there were no resources to contain them. Frames intended for a host whose RU is in the No Receive Resources state, fall into this category. This counter is updated only if the RU is in the No Resources state. When in Monitor mode this counter counts the total number of frames—good and bad.

OVRNERRS

This 32-bit quantity contains the number of frames known to be lost because the local system bus was not available. If the traffic problem lasts longer than the duration of one frame, the frames that follow the first are lost without an indicator, and they are not counted. This counter is updated, if needed, regardless of the RU state.

This 32-bit counter contains the number of collisions detected during frame reception. This counter will only be updated if at least 64 bytes of data are received before the collision occurs. If a collision occurs before 64 bytes of data are received, the frame is counted as a short frame. If the collision occurs in the preamble, no counters are incremented.

ACTION COMMANDS AND OPERATING MODES

This section lists all the Action Commands of the Command Unit Command Block List (CBL). Each command contains the Command field, the Status and Control fields, the link to the next Action Command, and any command-specific parameters. There are three basic types of action commands: 82596 Configuration and Setup, Transmission, and Diagnostics. The following is a list of the actual commands.

- NOP
- Individual Address Setup
- Configure
- MC Setup
- Transmit
- TDR
- Dump
- Diagnose

The 82596 has three addressing modes. In the 82586 mode all the Action Commands look exactly like those of the 82586.

- **82586 Mode.** The 82596 software and memory structure is compatible with the 82586.
- **32-Bit Segmented Mode.** The 82596 can access the entire system memory and use the two new memory structures—Simplified and Flexible—while still using the segmented approach. This does not require any significant changes to existing software.
- **Linear Mode.** The 82596 operates in a flat, linear, 4 gigabyte memory space without segmentation. It can also use the two new memory structures.

In the 32-bit Segmented mode there are some differences between the 82596 and 82586 action commands, mainly in programming and activating new 82596 features. Those bits marked "don't care" in the compatible mode are not checked; however, we strongly recommend that those bits all be zeroes; this will allow future enchancements and extensions.

In the Linear mode all of the address offsets become 32-bit address pointers. All new 82596 features are accessible in this mode, and all bits previously marked "don't care" must be zeroes.

The Action Commands, and all other 82596 memory structures, must begin on even byte boundaries, i.e., they must be word aligned.

NOP

This command results in no action by the 82596 except for those performed in the normal command processing. It is used to manipulate the CBL manipulation. The format of the NOP command is shown in Figure 21.

NOP—82586 and 32-Bit Segmented Modes

31																16 15																0	
ODD WORD																			EVEN WORD														
EL	S	I	X	X	X	X	X	X	X	X	X	0	0	0	C	B	OK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	A15																A0	4
LINK OFFSET																																	

NOP—Linear Mode

31																16 15																0	
ODD WORD																			EVEN WORD														
EL	S	I	0	0	0	0	0	0	0	0	0	0	0	0	0	C	B	OK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A31																																A0	4
LINK ADDRESS																																	

Figure 21

where:

LINK POINTER — In the 82586 or 32-bit Segmented modes this is a 16-bit offset to the next Command Block. In the Linear mode this is the 32-bit address of the next Command Block.

EL — If set, this bit indicates that this command block is the last on the CBL.

S — If set to one, suspend the CU upon completion of this CB.

I — If set to one, the 82596 will generate an interrupt after execution of the command is complete. If I is not set to one, the CX bit will not be set.

CMD (bits 16–18) — The NOP command. Value: 0h.

Bits 19–28 — Reserved (zero in the 32-bit Segmented and Linear modes).

C — This bit indicates the execution status of the command. The CPU initially resets it to zero when the Command Block is placed on the CBL. Following a command Completion, the 82596 will set it to one.

B — This bit indicates that the 82596 is currently executing the NOP command. It is initially reset to zero by the CPU. The 82596 sets it to one when execution begins and to zero when execution is completed. This bit is also set when the 82596 prefetches the command.

NOTE:

The C and B bits are modified in one operation.

OK — Indicates that the command was executed without error. If set to one no error occurred (command executed OK). If zero an error occured.

Individual Address Setup

This command is used to load the 82596 with the Individual Address. This address is used by the 82596 for inserting the Source Address during transmission and recognizing the Destination Address during reception. After RESET, and prior to Individual Address Setup Command execution, the 82596 assumes the Broadcast Address is the Individual Address in all aspects, i.e.:

- This will be the Individual Address Match reference.
- This will be the Source Address of a transmitted frame (for AL-LOC=0 mode only).

The format of the Individual Address Setup command is shown in Figure 22.

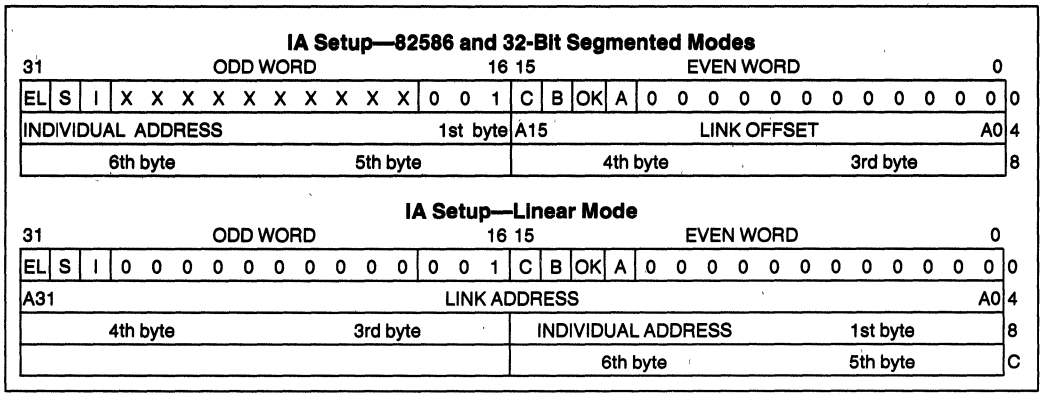


Figure 22

where:

LINK ADDRESS, EL, B, C, I, S — As per standard Command Block (see the NOP command for details)

A — Indicates that the command was abnormally terminated due to CU Abort control command. If one, then the command was aborted, and if necessary it should be repeated. If this bit is zero, the command was not aborted.

Bits 19–28 — Reserved (zero in the 32-bit Segmented and Linear modes).

CMD (bits 16–18) — The Address Setup command. Value: 1h.

INDIVIDUAL ADDRESS — The individual address of the node, 0 to 6 bytes long.

The least significant bit of the Individual Address must be zero for Ethernet (see the Command Structure). However, no enforcement of 0 is provided by the 82596. Thus, an Individual Address with 1 as its least significant bit is a valid Individual Address in all aspects.

The default address length is 6 bytes long, as in 802.3. If a different length is used the IA Setup command should be executed after the Configure command.

Configure

The Configure command loads the 82596 with its operating parameters. It allows changing some of the parameters by specifying a byte count less than the maximum number of configuration bytes (11 in the 82586 mode, 14 in the 32-Bit Segmented and Linear modes). The 82596 configuration depends on its mode of operation. When configuring the 12th byte (Byte 11 undefined) in 82586 mode this byte should be all ones.

- In the 82586 mode the maximum number of configuration bytes is 12. Any number larger than 12 will be reduced to 12 and any number less than 4 will be increased to 4.
- The additional features of the serial side are disabled in the 82586 mode.
- In both the 32-Bit Segmented and Linear modes there are four additional configuration bytes, which hold parameters for additional 82596 features. If these parameters are not accessed, the 82596 will follow their default values.
- For more detailed information refer to the 32-Bit LAN Components User's Manual.

The format of the Configure command is shown in Figure 23, 24 and 25.

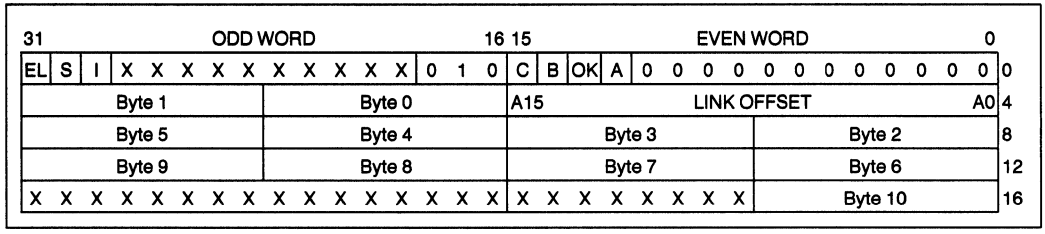


Figure 23. CONFIGURE—82586 Mode

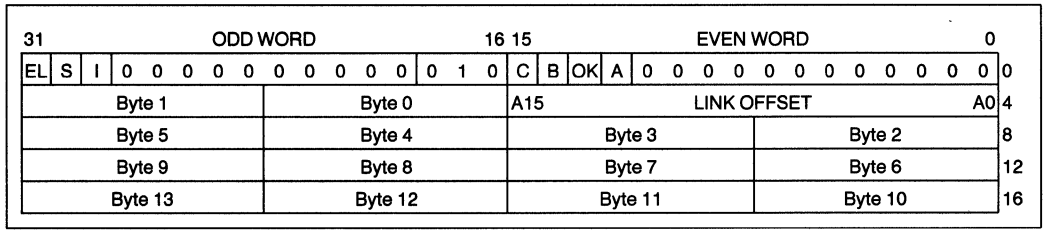


Figure 24. CONFIGURE—32-Bit Segmented Mode

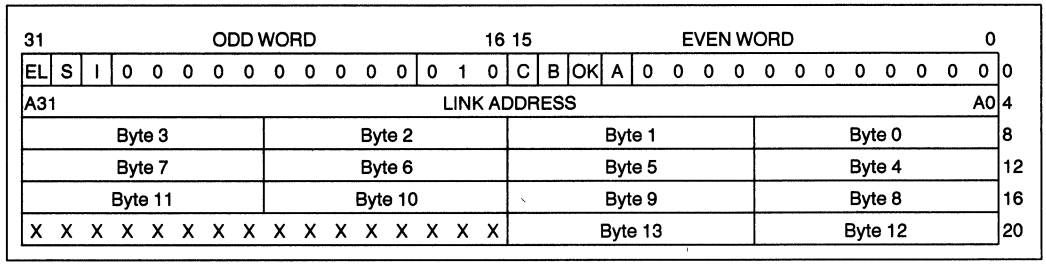


Figure 25. CONFIGURE—Linear Mode

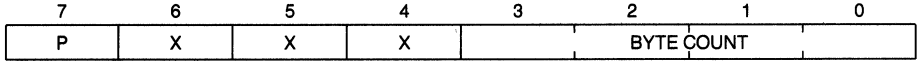
LINK ADDRESS, EL, B, C, I, S — As per standard Command Block (see the NOP command for details)

A — Indicates that the command was abnormally terminated due to a CU Abort control command. If 1, then the command was aborted and if necessary it should be repeated. If this bit is 0, the command was not aborted.

Bits 19–28 — Reserved (zero in the 32-Bit Segmented and Linear Modes)

CMD (bits 16–18) — The CONFIGURE command. Value: 2h.

The interpretation of the fields follows:

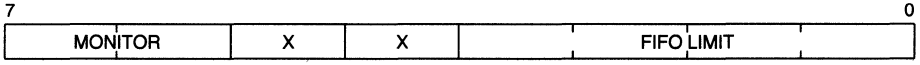


BYTE 0

BYTE CNT (Bits 0–3) — Byte Count. Number of bytes, including this one, that hold parameters to be configured.

PREFETCHED (Bit 7) — Enable the 82596 to write the prefetched bit in all prefetch RBDs.

NOTE:
The P bit is valid only in the new memory structure modes. In 82586 mode this bit is disabled (i.e., no prefetched mark).

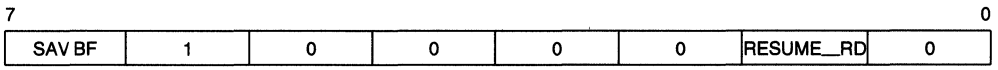


BYTE 1

FIFO Limit (Bits 0–3) — FIFO limit.

MONITOR# (Bits 6–7) — Receive monitor options. If the Byte Count of the configure command is less than 12 bytes then these Monitor bits are ignored.

DEFAULT: C8h

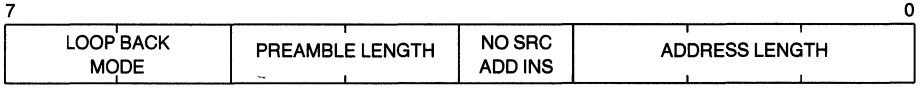


BYTE 2

SAV BF (Bit 7) — 0—Received bad frames are not saved in the memory.
1—Received bad frames are saved in the memory.

DEFAULT: 40h

RESUME_RD (Bit 1) — 0 — The 82596 does not reread the next CB on the list when a CU Resume Control Command is issued.
1 — The 82596 will reread the next CB on the list when a CU Resume Control Command is issued. This is available only on the 82596B stepping.



BYTE 3

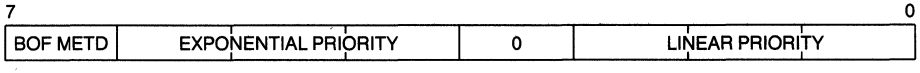
ADR LEN (Bits 0–2) — Address length (any kind).

NO SCR ADD INS (Bit 3) — No Source Address Insertion.
In the 82586 this bit is called AL LOC.

PREAM LEN (Bits 4–5) — Preamble length.

LP BCK MODE (Bits 6–7) — Loopback mode.

DEFAULT: 26h



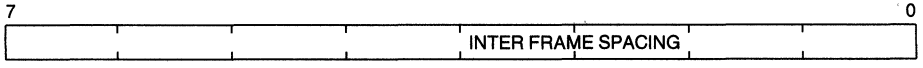
BYTE 4

LIN PRIO (Bits 0–2) — Linear Priority.

EXP PRIO (Bits 4–6) — Exponential Priority.

BOF METD (Bit 7) — Exponential Backoff method.

DEFAULT: 00h



BYTE 5

INTERFRAME SPACING — Interframe spacing.

DEFAULT: 60h

7							0
SLOT TIME - LOW							

BYTE 6

SLOT TIME (L) — Slot time, low byte.

DEFAULT: 00h

7							0
MAXIMUM RETRY NUMBER				0	SLOT TIME - HIGH		

BYTE 7

SLOT TIME (H) (Bits 0–2) — Slot time, high part.

RETRY NUM (Bits 4–7) — Number of transmission retries on collision.

DEFAULT: F2h

7							0
PAD	BIT STUFF	CRC16/ CRC32	NO CRC INSER	TONO CRS	MAN/ NRZ	BC DIS	PRM MODE

BYTE 8

PRM (Bit 0) — Promiscuous mode.

BC DIS (Bit 1) — Broadcast disable.

MANCH/NRZ (Bit 2) — Manchester or NRZ encoding. See specific timing requirements for TXC in Manchester mode.

TONO CRS (Bit 3) — Transmit on no CRS.

NOCRC INS (Bit 4) — No CRC insertion.

CRC-16/CRC-32 (Bit 5) — CRC type.

BIT STF (Bit 6) — Bit stuffing.

PAD (Bit 7) — Padding.

DEFAULT: 00h

7				0
CDT SRC	COLLISION DETECT FILTER	CRS SRC	CARRIER SENSE FILTER	

BYTE 9

CRSF (Bits 0–2) — Carrier Sense filter (length).

CRS SRC (Bit 3) — Carrier Sense source.

CDTF (Bits 4–6) — Collision Detect filter (length).

CDT SRC (Bit 7) — Collision Detect source.

DEFAULT: 00h

7						0
MINIMUM FRAME LENGTH						

BYTE 10

MIN FRAME LEN — Minimum frame length.

DEFAULT: 40h

7						0
MONITOR	MC_ALL	CDBSAC	AUTOTX	CRCINM	LNGFLD	PRECRS

BYTE 11

PRECRS (Bit 0) — Preamble until Carrier Sense

LNGFLD (Bit 1) — Length field. Enables padding at the End-of-Carrier framing (802.3).

CRCINM (Bit 2) — Rx CRC appended to the frame in memory.

AUTOTX (Bit 3) — Auto retransmit when a collision occurs during the preamble.

CDBSAC (Bit 4) — Collision Detect by source address recognition.

MC_ALL (Bit 5) — Enable to receive all MC frames.

MONITOR (Bits 6–7) — Receive monitor options.

DEFAULT: FFH

7							0
0	FDX	0	0	0	0	0	0

BYTE 12

FDX (Bit 6) — Enables Full Duplex operation.

DEFAULT: 00h

7							0
DIS_BOF	MULT_IA	1	1	1	1	1	1

BYTE 13

MULT_IA (Bit 6) — Multiple individual address.

DIS_BOF (Bit 7) — Disable the backoff algorithm.

DEFAULT: 3Fh

A reset (hardware or software) configures the 82596 according to the following defaults.

Table 4. Configuration Defaults

	Parameter	Default Value	Units/Meaning
	ADDRESS LENGTH	**6	Bytes
	A/L FIELD LOCATION	0	Located in FD
*	AUTO RETRANSMIT	1	Auto Retransmit Enable
	BITSTUFFING/EOC	0	EOC
	BROADCAST DISABLE	0	Broadcast Reception Enabled
*	CDBSAC	1	Disabled
	CDT FILTER	0	Bit Times
	CDT SRC	0	External Collision Detection
*	CRC IN MEMORY	1	CRC Not Transferred to Memory
	CRC-16/CRC-32	**0	CRC-32
	CRS FILTER	0	0 Bit Times
	CRS SRC	0	External CRS
*	DISBOF	0	Backoff Enabled
	EXT LOOPBACK	0	Disabled
	EXPONENTIAL PRIORITY	**0	802.3 Algorithm
	EXPONENTIAL BACKOFF METHOD	**0	802.3 Algorithm
*	FULL DUPLEX (FDX)	0	CSMA/CD Protocol (No FDX)
	FIFO THRESHOLD	8	TX: 32 Bytes, RX: 64 Bytes
	INT LOOPBACK	0	Disabled
	INTERFRAME SPACING	**96	Bit Times
	LINEAR PRIORITY	**0	802.3 Algorithm
*	LENGTH FIELD	1	Padding Disabled
	MIN FRAME LENGTH	**64	Bytes
*	MC ALL	1	Disabled
*	MONITOR	11	Disabled
	MANCHESTER/NRZ	0	NRZ
*	MULTI IA	0	Disabled
	NUMBER OF RETRIES	**15	Maximum Number of Retries
	NO CRC INSERTION	0	CRC Appended to Frame
	PREFETCH BIT IN RBD	0	Disabled (Valid Only in New Modes)
	PREAMBLE LENGTH	**7	Bytes
*	Preamble Until CRS	1	Disabled
	PROMISCUOUS MODE	0	Address Filter On
	PADDING	0	No Padding
	SLOT TIME	**512	Bit Times
	SAVE BAD FRAME	0	Discards Bad Frames
	TRANSMIT ON NO CRS	0	Disabled

NOTES:
1. This configuration setup is compatible with the IEEE 802.3 specification.
2. The Asterisk "*" signifies a new configuration parameter not available in the 82586.
3. The default value of the Auto retransmit configuration parameter is enabled(1).
4. Double Asterisk "**" signifies IEEE 802.3 requirements.

Multicast-Setup

This command is used to load the 82596 with the Multicast-IDs that should be accepted. As noted previously, the filtering done on the Multicast-IDs is not perfect and some unwanted frames may be accepted. This command resets the current filter and reloads it with the specified Multicast-IDs. The format of the Multicast-addresses setup command is:

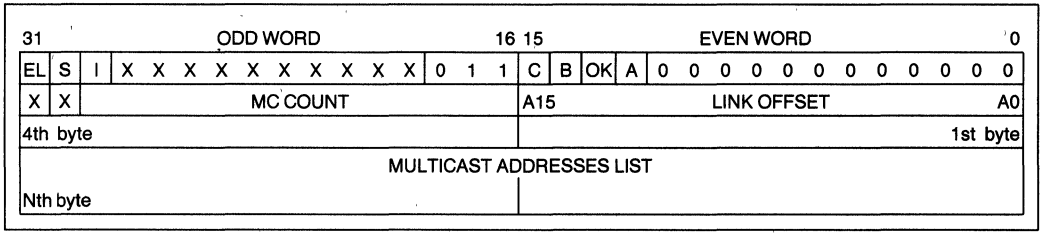


Figure 26. MC Setup—82586 and 32-Bit Segmented Modes

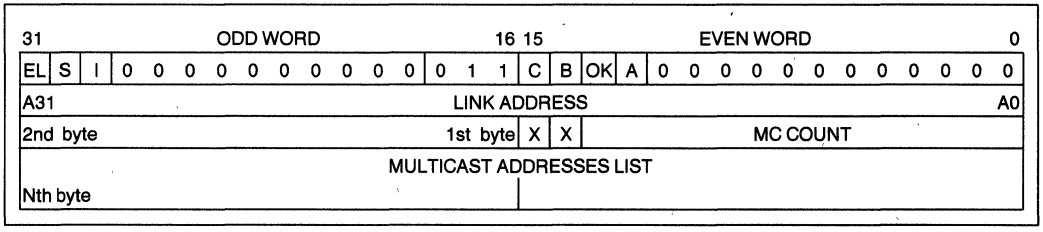


Figure 27. MC Setup—Linear Mode

where:

LINK ADDRESS, EL, B, C, I, S — As per standard Command Block (see the NOP command for details)

A — Indicates that the command was abnormally terminated due to a CU Abort control command. If one, then the command was aborted and if necessary it should be repeated. If this bit is zero, the command was not aborted.

Bits 19–28 — Reserved (0 in both the 32-Bit Segmented and Linear Modes).

CMD (bits 16–18) — The MC SETUP command value: 3h.

MC-CNT — This 14-bit field indicates the number of bytes in the MC LIST field. The MC CNT must be a multiple of the ADDR LEN; otherwise, the 82596 reduces the MC CNT to the nearest ADDR LEN multiple. MC CNT = 0 implies resetting the Hash table which is equivalent to disabling the Multicast filtering mechanism.

MC LIST — A list of Multicast Addresses to be accepted by the 82596. The least significant bit of each MC address must be 1.

NOTE:

The list is sequential; i.e., the most significant byte of an address is immediately followed by the least significant byte of the next address.

— When the 82596 is configured to recognize multiple Individual Address (Multi-IA), the MC-Setup command is also used to set up the Hash table for the individual address.

The least significant bit in the first byte of each IA address must be 0.

Transmit

This command is used to transmit a frame of user data onto the serial link. The format of a Transmit command is as follows.

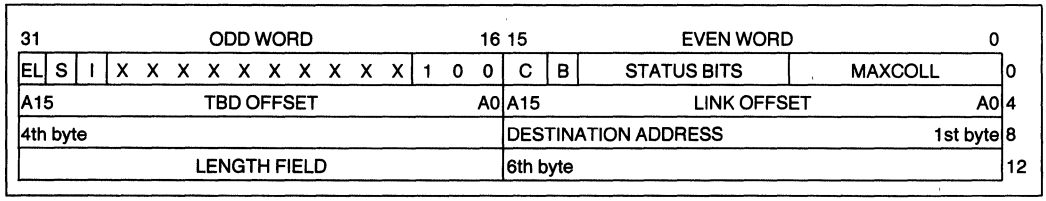


Figure 28. TRANSMIT—82586 Mode

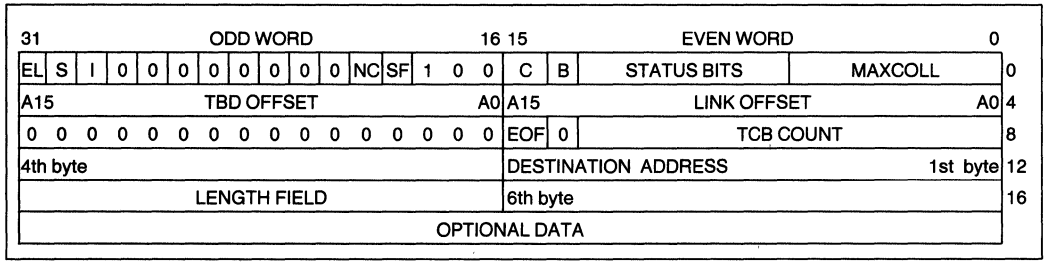


Figure 29. TRANSMIT—32-Bit Segmented Mode

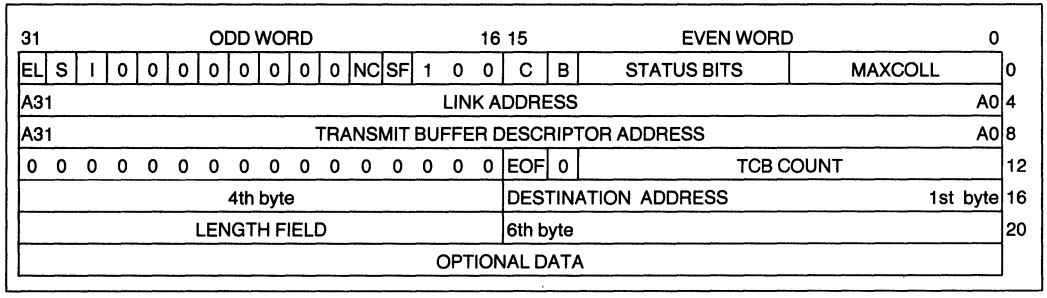
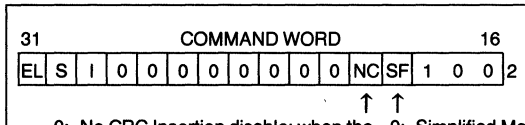


Figure 30. TRANSMIT—Linear Mode



NC:
- 0: No CRC Insertion disable; when the configure command is configured to not insert the CRC during transmission the NC bit has no effect.
- 1: No CRC Insertion enable; when the configure command is configured to insert the CRC during transmission the CRC will not be inserted when NC = 1.

SF:
- 0: Simplified Mode, all the Tx data is in the Transmit Command Block. The Transmit Buffer Descriptor Address field is all 1s.
- 1: Flexible Mode. Data is in the TCB and in a linked list of TBDs.

where:

EL, B, C, I, S	— As per standard Command Block (see the NOP command for details).
OK (Bit 13)	— Error free completion.
A (Bit 12)	— Indicates that the command was abnormally terminated due to CU Abort control command. If 1, then the command was aborted, and if necessary it should be repeated. If this bit is 0, the command was not aborted.
Bits 19–28	— Reserved (0 in the 32-bit Segmented and Linear modes).
CMD (Bits 16–18)	— The transmit command: 4h.
Status Bit 11	— Late collision. A late collision (a collision after the slot time is elapsed) is detected.
Status Bit 10	— No Carrier Sense signal during transmission. Carrier Sense signal is monitored from the end of Preamble transmission until the end of the Frame Check Sequence for TONOCRS = 1 (Transmit On No Carrier Sense mode) it indicates that transmission has been executed despite a lack of CRS. For TONOCRS = 0 (Ethernet mode), this bit also indicates unsuccessful transmission (transmission stopped when lack of Carrier Sense has been detected).
Status Bit 9	— Transmission unsuccessful (stopped) due to Loss of $\overline{\text{CTS}}$.
Status Bit 8	— Transmission unsuccessful (stopped) due to DMA Underrun; i.e., the system did not supply data for transmission.
Status Bit 7	— Transmission Deferred, i.e., transmission was not immediate due to previous link activity.
Status Bit 6	— Heartbeat Indicator, Indicates that after a previously performed transmission, and before the most recently performed transmission, (Interframe Spacing) the CDT signal was monitored as active. This indicates that the Ethernet Transceiver Collision Detect logic is performing properly. The Heartbeat is monitored during the Interframe Spacing period.
Status Bit 5	— Transmission attempt was stopped because the number of collisions exceeded the maximum allowable number of retries.
Status Bit 4	— 0 (Reserved).
MAX-COL (Bits 3–0)	— The number of Collisions experienced during this frame. Max Col = 0 plus S5 = 1 indicates 16 collisions.
LINK OFFSET	— As per standard Command Block (see the NOP Command for details)
TBD POINTER	— In the 82586 and 32-bit Segmented modes this is the offset of the first Tx Buffer Descriptor containing the data to be transmitted. In the Linear mode this is the 32-bit address of the first Tx Buffer Descriptor on the list. If the TBD POINTER is all 1s it indicates that no TBD is used.
DEST ADDRESS	— Contains the Destination Address of the frame. The least significant bit (MC) indicates the address type. MC = 0: Individual Address. MC = 1: Multicast or Broadcast Address. If the Destination Address bits are all 1s this is a Broadcast Address.
LENGTH FIELD	— The contents of this 2-byte field are user defined. In 802.3 it contains the length of the data field. It is placed in memory in the same order it is transmitted; i.e., most significant byte first, least significant byte second.
TCB COUNT	— This 14-bit counter indicates the number of bytes that will be transmitted from the Transmit Command Block, starting from the third byte after the TCB COUNT field (address $n+12$ in the 32-bit Segmented mode, $N+16$ in the Linear mode). The TCB COUNT field can be any number of bytes (including an odd byte), this allows the user to transmit a frame with a header having an odd number of bytes. The TCB COUNT field is not used in the 82586 mode.
EOF Bit	— Indicates that the whole frame is kept in the Transmit Command Block. In the Simplified memory model it must be always asserted.

The interpretation of what is transmitted depends on the No Source Address insertion configuration bit and the memory model being used.

NOTES:

1. The Destination Address and the Length Field are sequential. The Length Field immediately follows the most significant byte of the Destination Address.
2. In case the 82596 is configured with No Source Address insertion bit equal to 0, the 82596 inserts its configured Source Address in the transmitted frame.
   - In the 82586 mode, or when the Simplified memory model is used, the Destination and Length fields of the transmitted frame are taken from the Transmit Command Block.
   - If the FLEXIBLE memory model is used, the Destination and Length fields of the transmitted frame can be found either in the TCB or TBD, depending on the TCB COUNT.
3. If the 82596 is configured with the Address/Length Field Location equal to 1, the 82596 does not insert its configured Source Address in the transmitted frame. The first (2 × Address Length) + 2 bytes of the transmitted frame are interpreted as Destination Address, Source Address, and Length fields respectively. The location of the first transmitted byte depends on the operational mode of the 82596:
   - In the 82586 mode, it is always the first byte of the first Tx Buffer.
   - In both the 32-bit Segmented and Linear modes it depends on the SF bit and TCB COUNT:
     — In the Simplified memory mode the first transmitted byte is always the third byte after the TCB COUNT field.
     — In the Flexible mode, if the TCB COUNT is greater than 0 then it is the third byte after the TCB COUNT field. If TCB COUNT equals 0 then it is first byte of the first Tx Buffer.
   - Transmit frames shorter than six bytes are invalid. The transmission will be aborted (only in 82586 mode) because of a DMA Underrun.
4. Frames which are aborted during transmission are jammed. Such an interruption of transmission can be caused by any reason indicated by any of the status bits 8, 9, 10 and 12.

Jamming Rules

1. Jamming will not start before completion of preamble transmission.
2. Collisions detected during transmission of the last 11 bits will not result in jamming.

The format of a Transmit Buffer Descriptor is:

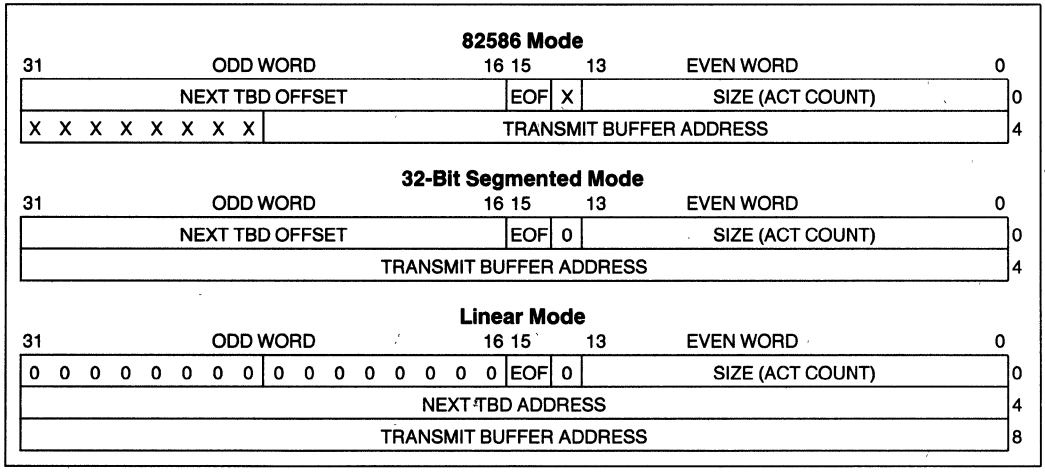


Figure 31

where:

EOF — This bit indicates that this TBD is the last one associated with the frame being transmitted. It is set by the CPU before transmit.

SIZE (ACT COUNT) — This 14-bit quantity specifies the number of bytes that hold information for the current buffer. It is set by the CPU before transmission.

NEXT TBD ADDRESS — In the 82586 and 32-bit Segmented modes, it is the offset of the next TBD on the list. In the Linear mode this is the 32-bit address of the next TBD on the list. It is meaningless if EOF = 1.

BUFFER ADDRESS — The starting address of the memory area that contains the data to be sent. In the 82586 mode, this is a 24-bit address (A31–A24 are considered to be zero). In the 32-bit Segmented and Linear modes this is a 32-bit address. This buffer can be byte aligned for the 82596 B step.

TDR

This operation activates Time Domain Reflectomet, which is a mechanism to detect open or short circuits on the link and their distance from the diagnosing station. The TDR command has no parameters. The TDR transmit sequence was changed, compared to the 82586, to form a regular transmission. The TDR command is designed to be used statically. Make sure that both the CU and RU are idle before attempting a TDR command. The TDR bit stream is as follows.

— Preamble

— Source address

— Another Source address (the TDR frame is transmitted back to the sending station, so DEST ADR = SRC ADR).

— Data field containing 7Eh patterns.

— Jam Pattern, which is the inverse CRC of the transmitted frame.

Maximum length of the TDR frame is 2048 bits. If the 82596 senses collision while transmitting the TDR frame it transmits the jam pattern and stops the transmission. The 82596 then triggers an internal timer (STC); the timer is reset at the beginning of transmission and reset if CRS is returned. The timer measures the time elapsed from the start of transmission until an echo is returned. The echo is indicated by Collision Detect going active or a drop in the Carrier Sense signal. The following table lists the possible cases that the 82596 is able to analyze.

Conditions of TDR as Interpreted by the 82596

Transceiver Type / Condition	Ethernet	Non Ethernet
Carrier Sense was inactive for 2048-bit-time periods	Short or Open on the Transceiver Cable	NA
Carrier Sense signal dropped	Short on the Ethernet cable	NA
Collision Detect went active	Open on the Ethernet cable	Open on the Serial Link
The Carrier Sense Signal did not drop or the Collision Detect did not go active within 2048-bit time period	No Problem	No Problem

An Ethernet transceiver is defined as one that returns transmitted data on the receive pair and activates the Carrier Sense Signal while transmitting. A Non-Ethernet Transceiver is defined as one that does not do so.

The format of the Time Domain Reflectometer command is:

82586 and 32-Bit Segmented Modes

31															16	15														0
ODD WORD																EVEN WORD														
EL	S	I	X	X	X	X	X	X	X	X	X	1	0	1		C	B	OK	0	0	0	0	0	0	0	0	0	0	0	0
LNK OK	XVR PRB	ET OPN	ET SRT	X	TIME (11 bits)											A15	LINK OFFSET													A0

Linear Mode

31															16	15														0
ODD WORD																EVEN WORD														
EL	S	I	0	0	0	0	0	0	0	0	0	1	0	1		C	B	OK	0	0	0	0	0	0	0	0	0	0	0	0
A31	LINK ADDRESS																													A0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LNK OK	XVR PRB	ET OPN	ET SRT	X	TIME (11 bits)									

Figure 32. TDR

where:

LINK ADDRESS, EL, B, C, I, S — As per standard Command Block (see the NOP command for details).

A — Indicates that the command was abnormally terminated due to CU Abort control command. If one, then the command was aborted, and if necessary it should be repeated. If this bit is zero, the command was not aborted.

Bits 19–28 — Reserved (0 in the 32-bit Segmented and Linear Modes).

CMD (Bits 16–18) — The TDR command. Value: 5h.

TIME — An 11-bit field that specifies the number of TxC cycles that elapsed before an echo was observed. No echo is indicated by a reception consisting of "1s" only. Because the network contains various elements such as transceiver links, transceivers, Ethernet, repeaters etc., the TIME is not exactly proportional to the problems distance.

LNK OK (Bit 15) — No link problem identified. TIME = 7FFh.

XCVR PRB (Bit 14) — Indicates a Transceiver problem. Carrier Sense was inactive for 2048-bit time period. LNK OK = 0. TIME = 7FFh.

ET OPN (Bit 13) — The transmission line is not properly terminated. Collision Detect went active and LNK OK = 0.

ET SRT (Bit 12) — There is a short circuit on the transmission line. Carrier Sense Signal dropped and LNK OK = 0.

DUMP

This command causes the contents of various 82596 registers to be placed in a memory area specified by the user. It is supplied as a 82596 self-diagnostic tool, and to provide registers of interest to the user. The format of the DUMP command is:

82586 and 32-Bit Segmented Modes

31			ODD WORD										16	15				EVEN WORD											0
EL	S	I	X	X	X	X	X	X	X	X	X	X	1	1	0	C	B	OK	0	0	0	0	0	0	0	0	0	0	0	0
A15			BUFFER OFFSET										A0			A15			LINK OFFSET										A0

Linear Mode

31			ODD WORD										16	15				EVEN WORD											0
EL	S	I	X	X	X	X	X	X	X	X	X	X	1	1	0	C	B	OK	0	0	0	0	0	0	0	0	0	0	0	0
A31			LINK ADDRESS																											A0
A31			BUFFER ADDRESS																											A0

Figure 33. Dump

where:

LINK ADDRESS, EL, B, C, I, S — As per standard Command Block (see the NOP command for details).

OK — Indicates error free completion.

Bits 19–28 — Reserved (0 in the 32-bit Segmented and Linear Modes).

CMD (Bits 16–18) — The Dump command. Value: 6h.

BUFFER POINTER — In the 82586 and 32-bit Segmented modes this is the 16-bit-offset portion of the dump area address. In the Linear mode this is the 32-bit linear address of the dump area.

Dump Area Information Format

- The 82596 is not Dump compatible with the 82586 because of the 32-bit internal architecture. In 82586 mode the 82596 will dump the same number of bytes as the 82586. The compatible data will be marked with an asterisk.
- In 82586 mode the dump area is 170 bytes.
- The DUMP area format of the 32-bit Segmented and Linear modes is described in Figure 35.
- The size of the dump area of the 32-bit Segmented and Linear modes is 304 bytes.
- When the Dump is executed by the Port command an extra word will be appended to the Dump Area. The extra word is a copy of the Dump Area status word (containing the C, B, and OK Bits). The C and OK Bits are set when the 82596 has completed the Port Dump command.

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
DMA CONTROL REGISTER	00
CONFIGURE BYTES* 3, 2	02
CONFIGURE BYTES* 5, 4	04
CONFIGURE BYTES* 7, 6	06
CONFIGURE BYTES* 9, 8	08
CONFIGURE BYTES* 10	0A
I.A. BYTES 1, 0*	0C
I.A. BYTES 3, 2*	0E
I.A. BYTES 5, 4*	10
LAST T.X. STATUS*	12
T.X. CRC BYTES 1, 0*	14
T.X. CRC BYTES 3, 2*	16
R.X. CRC BYTES 1, 0*	18
R.X. CRC BYTES 3, 2*	1A
R.X. TEMP MEMORY 1, 0*	1C
R.X. TEMP MEMORY 3, 2*	1E
R.X. TEMP MEMORY 5, 4*	20
LAST RECEIVED STATUS*	22
HASH REGISTER BYTES 1, 0*	24
HASH REGISTER BYTES 3, 2*	26
HASH REGISTER BYTES 5, 4*	28
HASH REGISTER BYTES 7, 6*	2A
SLOT TIME COUNTER*	2C
WAIT TIME COUNTER*	2E
MICRO MACHINE**	30
REGISTER FILE	.
60 BYTES	6A
MICRO MACHINE LFSR**	6C
MICRO MACHINE**	6E
FLAG ARRAY	.
14 BYTES	7A
QUEUE MEMORY**	7C
CU PORT	.
8 BYTES	82
MICRO MACHINE ALU**	84
RESERVED**	86
M.M. TEMP A ROTATE R**	88
M.M. TEMP A**	8A
T.X. DMA BYTE COUNT**	8C
M.M. INPUT PORT ADDRESS**	8E
T.X. DMA ADDRESS	90
M.M. OUTPUT PORT**	92
R.X. DMA BYTE COUNT**	94
M.M. OUTPUT PORT ADDRESS REGISTER**	96
R. DMA ADDRESS**	98
RESERVED**	9A
BUS THROTTLE TIMERS	9C
DIU CONTROL REGISTER**	9E
RESERVED**	A0
DMA CONTROL REGISTER**	A2
BIU CONTROL REGISTER**	A4
M.M. DISPATCHER REG.**	A6
M.M. STATUS REGISTER**	A8

*The 82596 is not Dump compatible with the 82586 because of the 32-bit internal architecture. In 82586 mode the 82596 will dump the same number of bytes as the 82586.
**These bytes are not user defined, results may vary from Dump command to Dump command.

Figure 34. Dump Area Format—82586 Mode

31	0	
CONFIGURE BYTES 5, 4, 3, 2		00
CONFIGURE BYTES 9, 8, 7, 6		04
CONFIGURE BYTES 13, 12, 11, 10		08
I.A. BYTES 1, 0	X X X X X X X X	0C
I.A. BYTES 5, 2		10
TX CRC BYTES 0, 1	LAST T.X. STATUS	14
RX CRC BYTES 0, 1	TX CRC BYTES 3, 2	18
RX TEMP MEMORY 1, 0	RX CRC BYTES 3, 2	1C
R.X. TEMP MEMORY 5, 2		20
HASH REGISTERS 1, 0	LAST R.X. STATUS	24
HASH REGISTER BYTES 5, 2		28
SLOT TIME COUNTER	HASH REGISTERS 7, 6	2C
RECEIVE FRAME LENGTH	WAIT-TIME COUNTER	30
MICRO MACHINE** REGISTER FILE 128 BYTES		34 . . . B0
MICRO MACHINE LFSR**		B4
MICRO MACHINE** FLAG ARRAY 28 BYTES		B8 . . . D0
M.M. INPUT PORT** 16 BYTES		D4 . . E0
MICRO MACHINE ALU**		E4
RESERVED**		E8
M.M. TEMP A ROTATE R.**		EC
M.M. TEMP A**		F0
T.X. DMA BYTE COUNT**		F4
M.M. INPUT PORT ADDRESS REGISTER**		F8
T.X. DMA ADDRESS**		FC
M.M. OUTPUT PORT REGISTER**		100
R.X. DMA BYTE COUNT**		104
M.M. OUTPUT PORT ADDRESS REGISTER**		108
R.X. DMA ADDRESS REGISTER**		10C
RESERVED**		110
BUS THROTTLE TIMERS		114
DIU CONTROL REGISTER**		118
RESERVED**		11C
DMA CONTROL REGISTER**		120
BIU CONTROL REGISTER**		124
M.M. DISPATCHER REG.**		128
M.M. STATUS REGISTER**		12C

The 82596 is not Dump compatible with the 82586 because of the 32-bit internal architecture. In 82586 mode the 82596 will dump the same number of bytes as the 82586.
**These bytes are not user defined, results may vary from Dump command to Dump command.

Figure 35. Dump Area Format—Linear and 32-Bit Segmented Mode

Diagnose

The Diagnose Command triggers an internal self-test procedure that checks internal 82596 hardware, which includes:

- Exponential Backoff Random Number Generator (Linear Feedback Shift Register).
- Exponential Backoff Timeout Counter.
- Slot Time Period Counter.
- Collision Number Counter.
- Exponential Backoff Shift Register.
- Exponential Backoff Mask Logic.
- Timer Trigger Logic.

This procedure checks the operation of the Backoff block, which resides in the serial side and is not easily controlled. The Diagnose command is performed in two phases.

The format of the 82596 Diagnose command is:

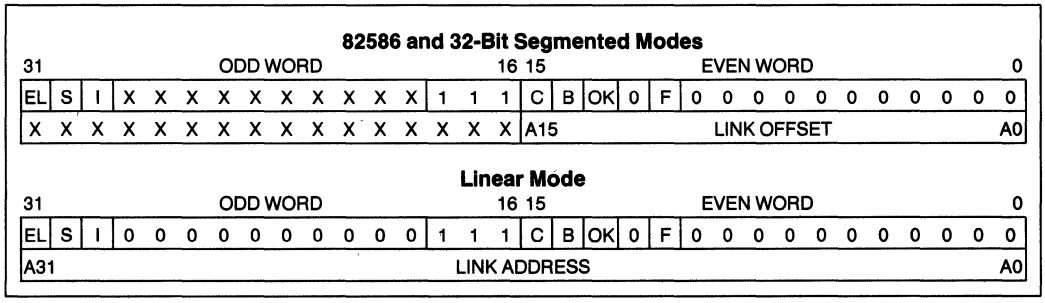


Figure 36. Diagnose

where:

LINK ADDRESS, EL, B, C, I, S	— As per standard Command Block (see the NOP command for details).
Bits 19–28	— Reserved (0 in the 32-bit Segmented and Linear Modes).
CMD (bits 16–18)	— The Diagnose command. Value: 7h.
OK (bit 13)	— Indicates error free completion.
F (bit 11)	— Indicates that the self-test procedure has failed.

RECEIVE FRAME DESCRIPTOR

Each received frame is described by one Receive Frame Descriptor (see Figure 37). Two new memory structures are available for the received frames. The structures are available only in the Linear and 32-bit Segmented modes.

Simplified Memory Structure

The first is the Simplified memory structure, the data section of the received frame is part of the RFD and is located immediately after the Length Field. Receive Buffer Descriptors are not used with the Simplified structure, it is primarily used to make programming easier. If the length of the data area described in the Size Field is smaller than the incoming frame, the following happens.

1. The received frame is truncated.
2. The No Resource error counter is updated.
3. If the 82596 is configured to Save Bad Frames the RFD is not reused; otherwise, the same RFD is used to hold the next received frame, and the only action taken regarding the truncated frame is to update the counter.
4. The 82596 continues to receive the next frame in the next RFD.

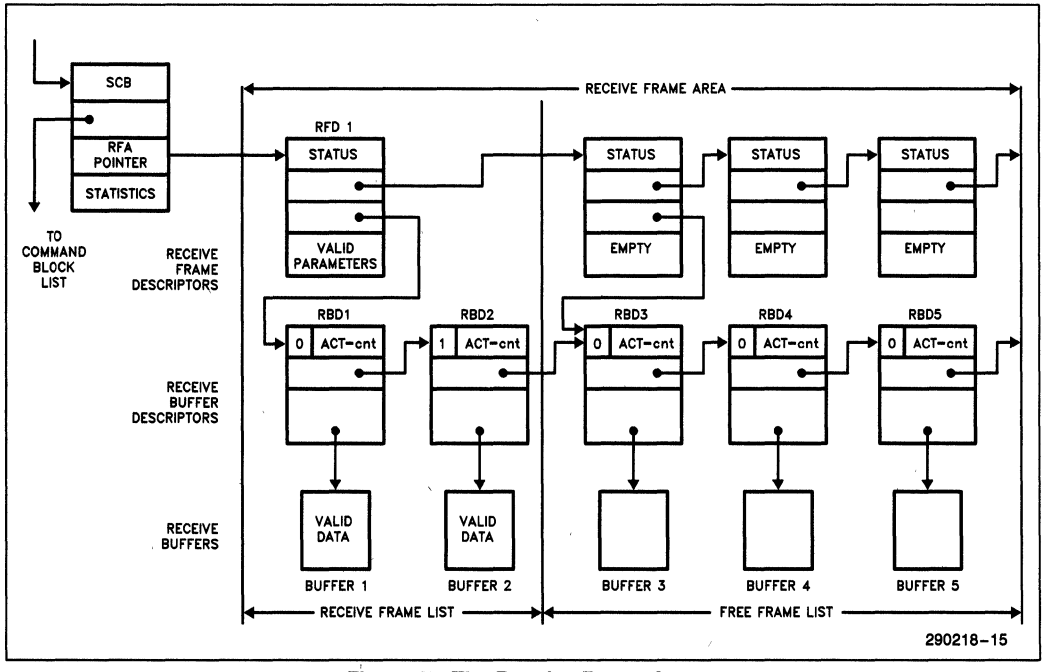


Figure 37. The Receive Frame Area

Note that this sequence is very useful for monitoring. If the 82596 is configured to Save Bad Frames, to receive in Promiscuous mode, and to use the Simplified memory structure, any programmed length of received data can be saved in memory.

The Simplified memory structure is shown in Figure 38.

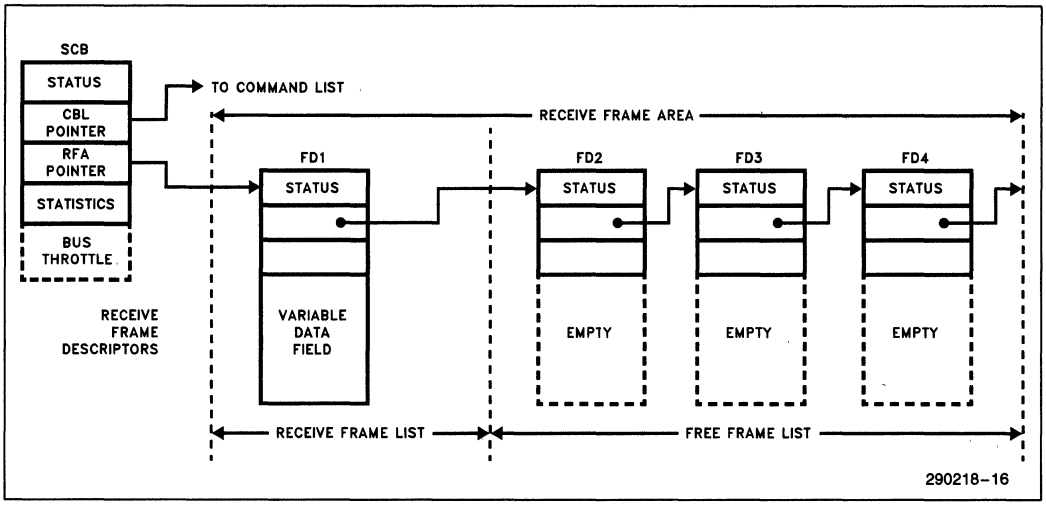


Figure 38. RFA Simplified Memory Structure

Flexible Memory Structure

The second structure is the Flexible memory structure, the data structure of the received frame is stored in both the RFD and in a linked list of Receive Buffers—Receive Buffer Descriptors. The received frame is placed in the RFD as configured in the Size field. Any remaining data is placed in a linked list of RBDs.

The Flexible memory structure is shown in Figure 39.

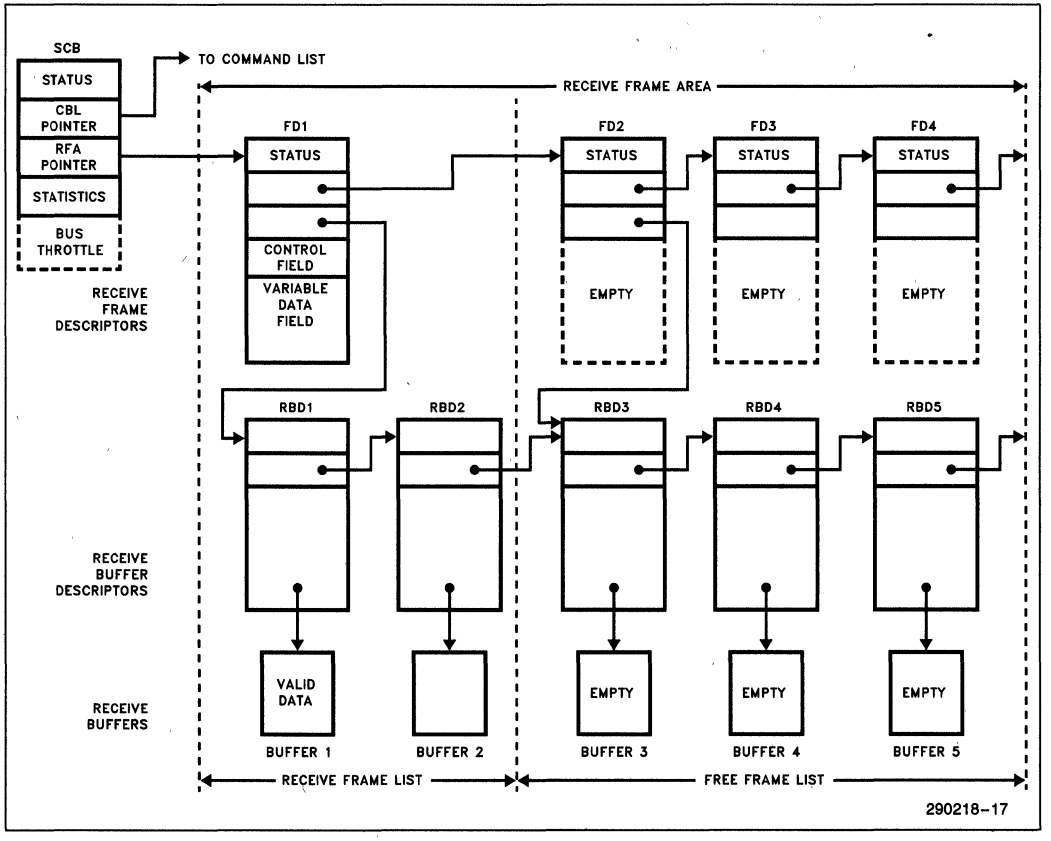


Figure 39. RFA Flexible Memory Structure

Buffers on the receive side can be different lengths. The 82596 will not place more bytes into a buffer than indicated in the associated RBD. The 82596 will fetch the next RBD before it is needed. The 82596 will attempt to receive frames as long as the FBL is not exhausted. If there are no more buffers, the 82596 Receive Unit will enter the No Resources state. Before starting the RU, the CPU must place the FBL pointer in the RBD pointer field of the first RFD. All remaining RBD pointer fields for subsequent RFDs should be "1s." If the Receive Frame Descriptor and the associated Receive Buffers are not reused (e.g., the frame is properly received or the 82596 is configured to Save Bad Frames), the 82596 writes the address of the next free RBD to the RBD pointer field of the next RFD.

Receive Buffer Descriptor (RBD)

The RBDs are used to store received data in a flexible set of linked buffers. The portion of the frame's data field that is outside the RFD is placed in a set of buffers chained by a sequence of RBDs. The RFD points to the first RBD, and the last RBD is flagged with an EOF bit set to 1. Each buffer in the linked list of buffers related to a particular frame can be any size up to 2^{14} bytes but must be word aligned (begin on an even numbered byte). This ensures optimum use of the memory resources while maintaining low overhead. All buffers in a frame are filled with the received data except for the last, in which the actual count can be smaller than the allocated buffer space.

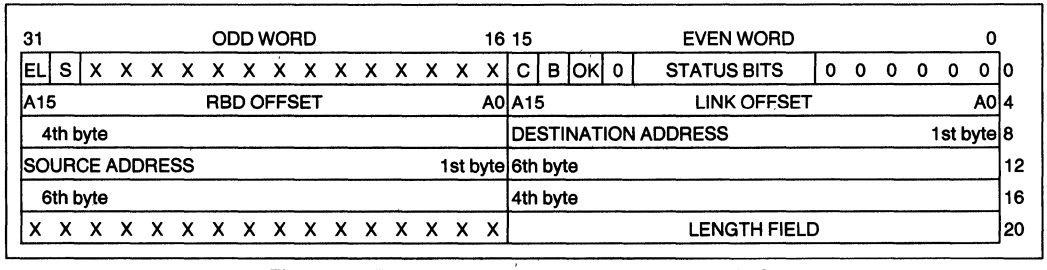


Figure 40. Receive Frame Descriptor—82586 Mode

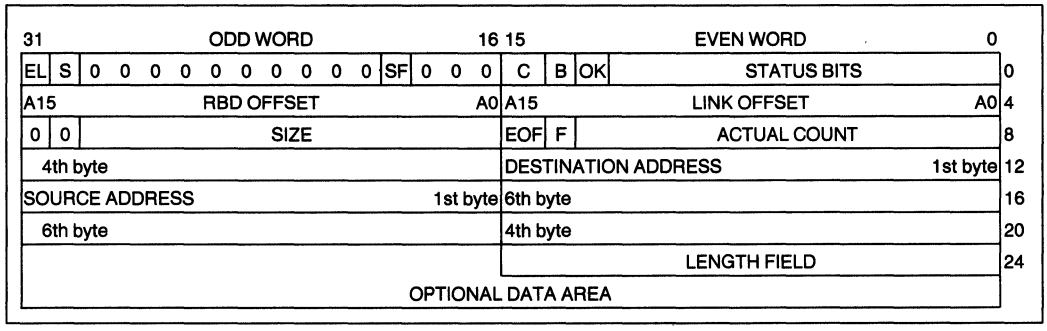


Figure 41. Receive Frame Descriptor—32-Bit Segmented Mode

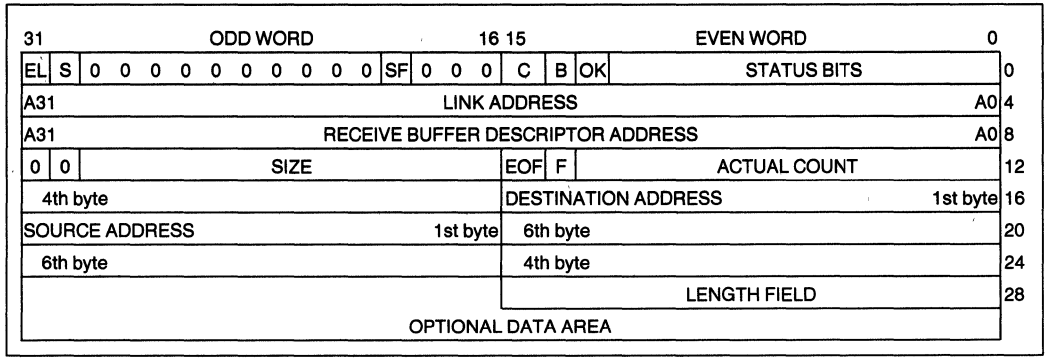


Figure 42. Receive Frame Descriptor—Linear Mode

where:

EL — When set, this bit indicates that this RFD is the last one on the RDL.

S — When set, this bit suspends the RU after receiving the frame.

SF — This bit selects between the Simplified or the Flexible mode.

0 — Simplified mode, all the RX data is in the RFD. RBD ADDRESS field is all "1s."

1 — Flexible mode. Data is in the RFD and in a linked list of Receive Buffer Descriptors.

C — This bit indicates the completion of frame reception. It is set by the 82596.

B — This bit indicates that the 82596 is currently receiving this frame, or that the 82596 is ready to receive the frame. It is initially set to 0 by the CPU. The 82596 sets it to 1 when reception set up begins, and to 0 upon completion. The C and B bits are set during the same operation.

OK (bit 13) — Frame received successfully, without errors. RFDs with bit 13 equal to 0 are possible only if the save bad frames, configuration option is selected. Otherwise all frames with errors will be discarded, although statistics will be collected on them.

STATUS — The results of the Receive operation. Defined bits are,

Bit 12: Length error if configured to check length

Bit 11: CRC error in an aligned frame

Bit 10: Alignment error (CRC error in misaligned frame)

Bit 9: Ran out of buffer space—no resources

Bit 8: DMA Overrun failure to acquire the system bus.

Bit 7: Frame too short.

Bit 6: No EOP flag (for Bit stuffing only)

Bit 5: When the SF bit equals zero, and the 82596 is configured to save bad frames, this bit signals that the receive frame was truncated. Otherwise it is zero.

Bits 2–4: Zeros

Bit 1: When it is zero, the destination address of the received frame matches the IA address. When it is a 1, the destination address of the received frame did not match the individual address. For example, a multicast address or broadcast address will set this bit to a 1.

Bit 0: Receive collision. A collision is detected during reception and the collision occurred after the destination address was received.

LINK ADDRESS — A 16-bit offset (32-bit address in the Linear mode) to the next Receive Frame Descriptor. The Link Address of the last frame can be used to form a cyclical list.

RBD POINTER — The offset (address in the Linear mode) of the first RBD containing the received frame data. An RBD pointer of all ones indicates no RBD.

EOF
F
SIZE
ACT COUNT — These fields are for the Simplified and Flexible memory models. They are exactly the same as the respective fields in the Receive Buffer Descriptor. See the next section for detailed explanation of their functions.

MC — Multicast bit.

DESTINATION ADDRESS — The contents of the destination address of the receive frame. The field is 0 to 6 bytes long.

SOURCE ADDRESS — The contents of the Source Address field of the received frame. It is 0 to 6 bytes long.

LENGTH FIELD — The contents of this 2-byte field are user defined. In 802.3 it contains the length of the data field. It is placed in memory in the same order it is received, i.e., most significant byte first, least significant byte second.

NOTES

1. The Destination address, Source address and Length fields are packed, i.e., one field immediately follows the next.
2. The affect of Address/Length Location (No Source Address Insertion) configuration parameter while receiving is as follows:

— 82586 Mode: The Destination address, Source address and Length field are not used, they are placed in the RX data buffers.

— 32-Bit Segmented and Linear Modes: when the Simplified memory model is used, the Destination address, Source address and Length fields reside in their respective fields in the RFD. When the Flexible memory strucrture is used the Destination address, Source address, and Length field locations depend on the SIZE field of the RFD. They can be placed in the RFD, in the RX data buffers, or partially in the RFD and the rest in the RX data buffers, depending on the SIZE field value.

82586 Mode

31 ODD WORD 16	15 EVEN WORD 0	
A15 NEXT RBD OFFSET A0	EOF F ACTUAL COUNT	0
X X X X X X X X A23 RECEIVE BUFFER ADDRESS A0		4
X X X X X X X X X X X X X X X X	EL X SIZE	8

32-Bit Segmented Mode

31 ODD WORD 16	15 EVEN WORD 0	
A15 NEXT RBD OFFSET A0	EOF F ACTUAL COUNT	0
A31 RECEIVE BUFFER ADDRESS A0		4
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	EL P SIZE	8

Linear Mode

31 ODD WORD 16	15 EVEN WORD 0	
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	EOF F ACTUAL COUNT	0
A31 NEXT RBD ADDRESS A0		4
A31 RECEIVE BUFFER ADDRESS A0		8
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	EL P SIZE	

Figure 43. Receive Buffer Descriptor

where:

EOF — Indicates that this is the last buffer related to the frame. It is cleared by the CPU before starting the RU, and is written by the 82596 at the end of reception of the frame.

F — Indicates that this buffer has already been used. The Actual Count has no meaning unless the F bit equals one. This bit is cleared by the CPU before starting the RU, and is set by the 82596 after the associated buffer has been. This bit has the same meaning as the Complete bit in the RFD and CB.

ACT COUNT — This 14-bit quantity indicates the number of meaningful bytes in the buffer. It is cleared by the CPU before starting the RU, and is written by the 82596 after the associated buffer has already been used. In general, after the buffer is full, the Actual Count value equals the size field of the same buffer. For the last buffer of the frame, Actual Count can be less than the buffer size.

NEXT BD ADDRESS — The offset (absolute address in the Linear mode) of the next RBD on the list. It is meaningless if EL = 1.

BUFFER ADDRESS — The starting address of the memory area that contains the received data. In the 82586 mode, this is a 24-bit address (with pins A24–A31 = 0). In the 32-bit Segmented and Linear modes this is a 32-bit address.

EL — Indicates that the buffer associated with this RBD is last in the FBL.

P — This bit indicates that the 82596 has already prefetched the RBDs and any change in the RBD data will be ignored. This bit is valid only in the new 82596 memory modes, and if this feature has been enabled during configure command. The 82596 Prefetches the RBDs in locked cycles; after prefetching the RBD the 82596 performs a write cycle where the P bit is set to one and the rest of the data remains unchanged. The CPU is responsible for resetting it in all RBDs. The 82596 will not check this bit before setting it.

SIZE — This 14-bit quantity indicates the size, in bytes, of the associated buffer. This quantity must be an even number.

PGA PACKAGE THERMAL SPECIFICATION

Parameter	Thermal Resistance
θ_{JC}	3°C/W
θ_{JA}	24°C/W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

ELECTRICAL AND TIMING CHARACTERISTICS

Absolute Maximum Ratings

- Storage Temperature ..... −65°C to +150°C
- Case Temperature under Bias −65°C to +110°C
- Supply Voltage with Respect to V_{SS} ..... −0.5V to +6.5V
- Voltage on Other Pins .... −0.5V to V_{CC} + 0.5V

2

DC Characteristics

T_C = 0°C–85°C, V_{CC} = 5V ±10% LE/$\overline{BE}$ have MOS levels (see V_{MIL}, V_{MIH}).
All other signals have TTL levels (see V_{IL}, V_{IH}, V_{OL}, V_{OH}).

Symbol	Parameter	Min	Max	Units	Notes
V_{IL}	Input Low Voltage (TTL)	−0.3	+0.8	V	
V_{IH}	Input High Voltage (TTL)	2.0	V_{CC} + 0.3	V	
V_{MIL}	Input Low Voltage (MOS)	−0.3	+0.8	V	
V_{MIH}	Input High Voltage (MOS)	3.7	V_{CC} + 0.3	V	
V_{OL}	Output Low Voltage (TTL)		0.45	V	I_{OL} = 4.0 mA
V_{CIL}	$\overline{RXC}$, $\overline{TXC}$ Input Low Voltage	−0.5	0.6	V	
V_{CIH}	$\overline{RXC}$, $\overline{TXC}$ Input High Voltage	3.3	V_{CC}+0.5	V	
V_{OH}	Output High Voltage (TTL)	2.4		V	I_{OH} = 0.9 mA–1 mA
I_{LI}	Input Leakage Current		±15	μA	$0 \le V_{IN} \le V_{CC}$
I_{LO}	Output Leakage Current		±15	μA	$0.45 < V_{OUT} < V_{CC}$
C_{IN}	Capacitance of Input Buffer		10	pF	FC = 1 MHz
C_{OUT}	Capacitance of Input/Output Buffer		12	pF	FC = 1 MHz
C_{CLK}	CLK Capacitance		20	pF	FC = 1 MHz
I_{CC}	Power Supply		200	mA	At 25 MHz I_{CC} Typical = 100 mA
I_{CC}	Power Supply		300	mA	At 33 MHz I_{CC} Typical = 150 mA

AC Characteristics

82596CA C-STEP INPUT/OUTPUT SYSTEM TIMINGS

T_C = 0°C–+85°C, V_{CC} = 5V ±10%. These timing assume the C_L on all outputs is 50 pF unless otherwise specified. C_L can be 20 pF to 120 pF however timings must be derated. All timing requirements are given in nanoseconds.

Symbol	Parameter	16 MHz		Notes
		Min	Max	
	Operating Frequency	12.5 MHz	16 MHz	1X CLK Input
T1	CLK Period	62.5	80	
T1a	CLK Period Stability		0.1%	Adjacent CLK Δ
T2	CLK High	20		2.0V
T3	CLK Low	20		0.8V
T4	CLK Rise Time		8	0.8V to 2.0V
T5	CLK Fall Time		8	2.0V to 0.8V
T6	$\overline{BE}n$, $\overline{LOCK}$, and A2–A31 Valid Delay	3	23	
T6a	$\overline{BLAST}$, $\overline{PCHK}$ Valid Delay	3	32	
T7	$\overline{BE}n$, $\overline{LOCK}$, $\overline{BLAST}$, A2–A31 Float Delay	3	39	
T8	W/$\overline{R}$ and $\overline{ADS}$ Valid Delay	3	23	
T9	W/$\overline{R}$ and $\overline{ADS}$ Float Delay	3	39	
T10	D0–D31, DPn Write Data Valid Delay	3	27	
T11	D0–D31, DPn Write Data Float Delay	3	39	
T12	HOLD Valid Delay	2	30	
T13	CA and BREQ Setup Time	11		1, 2
T14	CA and BREQ Hold Time	6		1, 2
T15	$\overline{BS16}$ Setup Time	12		2
T16	$\overline{BS16}$ Hold Time	5		2
T17	$\overline{BRDY}$, $\overline{RDY}$ Setup Time	12		2
T18	$\overline{BRDY}$, $\overline{RDY}$ Hold Time	5		2
T19	D0–D31, DPn READ Setup Time	10		2
T20	D0–D31, DPn READ Hold Time	6		2
T21	AHOLD and HLDA Setup Time	15		1, 2
T22	AHOLD Hold Time	5		1, 2
T22a	HLDA Hold Time	5		1, 2
T23	RESET Setup Time	14		1, 2
T24	RESET Hold Time	5		1, 2
T25	INT/$\overline{INT}$ Valid Delay	1	23	
T26	CA and BREQ, $\overline{PORT}$ Pulse Width	2 T1		1, 2, 3
T27	D0–D31 CPU $\overline{PORT}$ Access Setup Time	10		2
T28	D0–D31 CPU $\overline{PORT}$ Access Hold Time	6		2
T29	$\overline{PORT}$ Setup Time	11		2
T30	$\overline{PORT}$ Hold Time	5		2
T31	$\overline{BOFF}$ Setup Time	12		2
T32	$\overline{BOFF}$ Hold Time	5		2

*Timings shown are for the 82596CA C-Stepping. For information regarding timings for the 82596CA A1 or B-Step, contact your local Intel representative.

AC Characteristics (Continued)

82596CA C-STEP INPUT/OUTPUT SYSTEM TIMINGS

T_C = 0°C–+85°C, V_{CC} = 5V ±10%. These timing assume the C_L on all outputs is 50 pF unless otherwise specified. C_L can be 20 pF to 120 pF however timings must be derated. All timing requirements are given in nanoseconds.

Symbol	Parameter	20 MHz		Notes
		Min	Max	
	Operating Frequency	12.5 MHz	20 MHz	1X CLK Input
T1	CLK Period	50	80	
T1a	CLK Period Stability		0.1%	Adjacent CLK Δ
T2	CLK High	16		2.0V
T3	CLK Low	16		0.8V
T4	CLK Rise Time		6	0.8V to 2.0V
T5	CLK Fall Time		6	2.0V to 0.8V
T6	$\overline{BEn}$, $\overline{LOCK}$, and A2–A31 Valid Delay	3	20	
T6a	$\overline{BLAST}$, $\overline{PCHK}$ Valid Delay	3	25	
T7	$\overline{BEn}$, $\overline{LOCK}$, $\overline{BLAST}$, A2–A31 Float Delay	3	34	
T8	W/$\overline{R}$ and $\overline{ADS}$ Valid Delay	3	20	
T9	W/$\overline{R}$ and $\overline{ADS}$ Float Delay	3	34	
T10	D0–D31, DPn Write Data Valid Delay	3	23	
T11	D0–D31, DPn Write Data Float Delay	3	34	
T12	HOLD Valid Delay	2	25	
T13	CA and BREQ Setup Time	10		1, 2
T14	CA and BREQ Hold Time	6		1, 2
T15	$\overline{BS16}$ Setup Time	12		2
T16	$\overline{BS16}$ Hold Time	4		2
T17	$\overline{BRDY}$, $\overline{RDY}$ Setup Time	12		2
T18	$\overline{BRDY}$, $\overline{RDY}$ Hold Time	4		2
T19	D0–D31, DPn READ Setup Time	6		2
T20	D0–D31, DPn READ Hold Time	5		2
T21	AHOLD and HLDA Setup Time	15		1, 2
T22	AHOLD Hold Time	4		1, 2
T22a	HLDA Hold Time	5		1, 2
T23	RESET Setup Time	12		1, 2
T24	RESET Hold Time	4		1, 2
T25	INT/$\overline{INT}$ Valid Delay	1	23	
T26	CA and BREQ, $\overline{PORT}$ Pulse Width	2 T1		1, 2, 3
T27	D0–D31 CPU $\overline{PORT}$ Access Setup Time	6		2
T28	D0–D31 CPU $\overline{PORT}$ Access Hold Time	5		2
T29	$\overline{PORT}$ Setup Time	10		2
T30	$\overline{PORT}$ Hold Time	5		2
T31	$\overline{BOFF}$ Setup Time	12		2
T32	$\overline{BOFF}$ Hold Time	4		2

*Timings shown are for the 82596CA C-Stepping. For information regarding timings for the 82596CA A1 or B-Step, contact your local Intel representative.

AC Characteristics (Continued)

82596CA C-STEP INPUT/OUTPUT SYSTEM TIMINGS

T_C = 0°C–+85°C, V_{CC} = 5V ±10%. These timing assume the C_L on all outputs is 50 pF unless otherwise specified. C_L can be 20 pF to 120 pF however timings must be derated. All timing requirements are given in nanoseconds.

Symbol	Parameter	25 MHz		Notes
		Min	Max	
	Operating Frequency	12.5 MHz	25 MHz	1X CLK Input
T1	CLK Period	40	80	
T1a	CLK Period Stability		0.1%	Adjacent CLK Δ
T2	CLK High	14		2.0V
T3	CLK Low	14		0.8V
T4	CLK Rise Time		4	0.8V to 2.0V
T5	CLK Fall Time		4	2.0V to 0.8V
T6	$\overline{BEn}$ Valid Delay	3	17	
T6a	$\overline{BLAST}$ Valid Delay	3	20	
T6b	$\overline{LOCK}$ Valid Delay	3	18	
T6c	A2–A31 Valid Delay	3	18	
T6d	$\overline{PCHK}$ Valid Delay	3	24	
T7	$\overline{BEn}$, $\overline{LOCK}$, $\overline{BLAST}$, A2–A31 Float Delay	3	30	
T8	W/$\overline{R}$ and $\overline{ADS}$ Valid Delay	3	19	
T9	W/$\overline{R}$ and $\overline{ADS}$ Float Delay	3	30	
T10	D0–D31, DPn Write Data Valid Delay	3	20	
T11	D0–D31, DPn Write Data Float Delay	3	30	
T12	HOLD Valid Delay	3	19	
T13	CA and BREQ Setup Time	7		1, 2
T14	CA and BREQ Hold Time	3		1, 2
T15	$\overline{BS16}$ Setup Time	8		2
T16	$\overline{BS16}$ Hold Time	3		2
T17	$\overline{BRDY}$ Setup Time	9		2
T17a	$\overline{RDY}$ Setup Time	8		2
T18	$\overline{BRDY}$, $\overline{RDY}$ Hold Time	3		2
T19	D0–D31, DPn READ Setup Time	6		2
T20	D0–D31, DPn READ Hold Time	4.5		2
T21	AHOLD and HLDA Setup Time	10		1, 2
T22	AHOLD Hold Time	3		1, 2
T22a	HLDA Hold Time	3		1, 2
T23	RESET Setup Time	10		1, 2
T24	RESET Hold Time	3		1, 2
T25	INT/$\overline{INT}$ Valid Delay	1	20	

*Timings shown are for the 82596CA C-Stepping. For information regarding timings for the 82596CA A1 or B-Step, contact your local Intel representative.

AC Characteristics (Continued)

82596CA C-STEP INPUT/OUTPUT SYSTEM TIMINGS

T_C = 0°C–+85°C, V_{CC} = 5V ±10%. These timing assume the C_L on all outputs is 50 pF unless otherwise specified. C_L can be 20 pF to 120 pF however timings must be derated. All timing requirements are given in nanoseconds.

Symbol	Parameter	25 MHz		Notes
		Min	Max	
T26	CA and BREQ, $\overline{PORT}$ Pulse Width	2 T1		1, 2, 3
T27	D0–D31 CPU $\overline{PORT}$ Access Setup Time	6		2
T28	D0–D31 CPU $\overline{PORT}$ Access Hold Time	4.5		2
T29	$\overline{PORT}$ Setup Time	7		2
T30	$\overline{PORT}$ Hold Time	3		2
T31	$\overline{BOFF}$ Setup Time	10		2
T32	$\overline{BOFF}$ Hold Time	3		2

*Timings shown are for the 82596CA C-Stepping. For information regarding timings for the 82596CA A1 or B-Step, contact your local Intel representative.

2

AC Characteristics (Continued)

82596CA C-STEP INPUT/OUTPUT SYSTEM TIMINGS

T_C = 0°C to +85°C, V_{CC} = 5V ±5%. These timing assume the C_L on all outputs is 50 pF unless otherwise specified. C_L can be 20 pF to 120 pF, however timings must be derated. All timing requirements are given in nanoseconds.

Symbol	Parameter	33 MHz		Notes
		Min	Max	
	Operating Frequency	12.5 MHz	33 MHz	1X CLK Input
T1	CLK Period	30	80	
T1a	CLK Period Stability		0.1%	Adjacent CLK Δ
T2	CLK High	11		2.0V
T3	CLK Low	11		0.8V
T4	CLK Rise Time		3	0.8V to 2.0V
T5	CLK Fall Time		3	2.0V to 0.8V
T6	$\overline{BEn}$ Valid Delay	3	17	
T6a	$\overline{BLAST}$ Valid Delay	3	20	
T6b	$\overline{LOCK}$ Valid Delay	3	16	
T6c	A2–A31 Valid Delay	3	18	
T6d	$\overline{PCHK}$ Valid Delay	3	23	
T7	$\overline{BEn}$, $\overline{LOCK}$, $\overline{BLAST}$, A2–A31 Float Delay	3	20	
T8	W/$\overline{R}$ and $\overline{ADS}$ Valid Delay	3	16	
T9	W/$\overline{R}$ and $\overline{ADS}$ Float Delay	3	20	
T10	D0–D31, DPn Write Data Valid Delay	3	19	
T11	D0–D31, DPn Write Data Float Delay	3	20	
T12	HOLD Valid Delay	3	19	
T13	CA and BREQ Setup Time	7		1, 2
T14	CA and BREQ Hold Time	3		1, 2
T15	$\overline{BS16}$ Setup Time	7		2
T16	$\overline{BS16}$ Hold Time	3		2
T17	$\overline{BRDY}$ Setup Time	9		2
T17a	$\overline{RDY}$ Setup Time	8		2
T18	$\overline{BRDY}$, $\overline{RDY}$ Hold Time	3		2
T19	D0–D31, DPn READ Setup Time	6		2
T20	D0–D31, DPn READ Hold Time	4.5		2
T21	AHOLD Setup Time	10		1, 2
T21a	HLDA Setup Time	8		1, 2
T22	AHOLD Hold Time	3		1, 2

*Timings shown are for the 82596CA C-Stepping. For information regarding timings for the 82596CA A1 or B-Step, contact your local Intel representative.

AC Characteristics (Continued)

82596CA C-STEP INPUT/OUTPUT SYSTEM TIMINGS

C_L on all outputs is 50 pF unless otherwise specified.
All timing requirements are given in nanoseconds.

Symbol	Parameter	33 MHz		Notes
		Min	Max	
T22a	HLDA Hold Time	3		1, 2
T23	RESET Setup Time	9		1, 2
T24	RESET Hold Time	3		1, 2
T25	INT/$\overline{INT}$ Valid Delay	1	20	
T26	CA and BREQ, $\overline{PORT}$ Pulse Width	2T1		1, 2, 3
T27	D0–D31 CPU $\overline{PORT}$ Access Setup Time	6		2
T28	D0–D31 CPU $\overline{PORT}$ Access Hold Time	4.5		2
T29	$\overline{PORT}$ Setup Time	7		2
T30	$\overline{PORT}$ Hold Time	3		2
T31	$\overline{BOFF}$ Setup Time	10		2
T32	$\overline{BOFF}$ Hold Time	3		2

NOTES:
*Timings shown are for the 82596CA C-stepping. For information regarding timings for the 82596CA A1 or B-step, contact your local Intel representative.
1. RESET, HLDA, and CA are internally synchronized. This timing is to guarantee recognition at next clock for RESET, HLDA and CA.
2. All set-up, hold and delay timings are at maximum frequency specification Fmax, and must be derated according to the following equation for operation at lower frequencies:
   Tderated = (Fmax/Fopr) × T
   where:
   Tderate = Specifies the value to derate the specification.
   Fmax = Maximum operating frequency.
   Fopr = Actual operating frequency.
   T = Specification at maximum frequency.
   This calculation only provides a rough estimate for derating the frequency. For more detailed information, contact your Intel Sales Office for the data sheet supplement.
3. CA pulse width need only be 1 T1 wide if the set up and hold times are met; BREQ must meet setup and hold times and need only be 1 T1 wide.

TRANSMIT/RECEIVE CLOCK PARAMETERS

Symbol	Parameter	20 MHz		Notes
		Min	Max	
T36	$\overline{TxC}$ Cycle	50		1, 3
T38	$\overline{TxC}$ Rise Time		5	1
T39	$\overline{TxC}$ Fall Time		5	1
T40	$\overline{TxC}$ High Time	19		1, 3
T41	$\overline{TxC}$ Low Time	18		1, 3
T42	TxD Rise Time		10	4
T43	TxD Fall Time		10	4
T44	TxD Transition	20		2, 4
T45	$\overline{TxC}$ Low to TxD Valid		25	4, 6
T46	$\overline{TxC}$ Low to TxD Transition		25	2, 4
T47	$\overline{TxC}$ High to TxD Transition		25	2, 4
T48	$\overline{TxC}$ Low to TxD High (At End of Transition)		25	4

TRANSMIT/RECEIVE CLOCK PARAMETERS (Continued)

Symbol	Parameter	20 MHz		Notes
		Min	Max	
$\overline{\text{RTS}}$ AND $\overline{\text{CTS}}$ PARAMETERS				
T49	$\overline{\text{TxC}}$ Low to $\overline{\text{RTS}}$ Low, Time to Activate $\overline{\text{RTS}}$		25	5
T50	$\overline{\text{CTS}}$ Low to $\overline{\text{TxC}}$ Low, $\overline{\text{CTS}}$ Setup Time		20	
T51	$\overline{\text{TxC}}$ Low to $\overline{\text{CTS}}$ Invalid, $\overline{\text{CTS}}$ Hold Time	10		7
T52	$\overline{\text{TxC}}$ Low to $\overline{\text{RTS}}$ High		25	5
RECEIVE CLOCK PARAMETERS				
T53	$\overline{\text{RXC}}$ Cycle	50		1, 3
T54	$\overline{\text{RXC}}$ Rise Time		5	1
T55	$\overline{\text{RXC}}$ Fall Time		5	1
T56	$\overline{\text{RXC}}$ High Time	19		1
T57	$\overline{\text{RXC}}$ Low Time	18		1
RECEIVED DATA PARAMETERS				
T58	RXD Setup Time	20		6
T59	RXD Hold Time	10		6
T60	RXD Rise Time		10	
T61	RXD Fall Time		10	
$\overline{\text{CRS}}$ AND $\overline{\text{CDT}}$ PARAMETERS				
T62	$\overline{\text{CDT}}$ Low to $\overline{\text{TXC}}$ HIGH External Collision Detect Setup Time	20		
T63	$\overline{\text{TXC}}$ High to $\overline{\text{CDT}}$ Inactive, $\overline{\text{CDT}}$ Hold Time	10		
T64	$\overline{\text{CDT}}$ Low to Jam Start			10
T65	$\overline{\text{CRS}}$ Low to $\overline{\text{TXC}}$ High, Carrier Sense Setup Time	20		
T66	$\overline{\text{TXC}}$ High to $\overline{\text{CRS}}$ Inactive, $\overline{\text{CRS}}$ Hold Time (Internal Collision Detect)	10		
T67	$\overline{\text{CRS}}$ High to Jamming Start,			12
T68	Jamming Period			11
T69	$\overline{\text{CRS}}$ High to $\overline{\text{RXC}}$ High, $\overline{\text{CRS}}$ Inactive Setup Time	30		
T70	$\overline{\text{RXC}}$ High to $\overline{\text{CRS}}$ High, $\overline{\text{CRS}}$ Inactive Hold Time	10		

TRANSMIT/RECEIVE CLOCK PARAMETERS (Continued)

Symbol	Parameter	20 MHz		Notes
		Min	Max	
INTERFRAME SPACING PARAMETERS				
T71	Interframe Delay			9
EXTERNAL LOOPBACK-PIN PARAMETERS				
T72	$\overline{TXC}$ Low to $\overline{LPBK}$ Low		T36	4
T73	$\overline{TXC}$ Low to $\overline{LPBK}$ High		T36	4

NOTES:
1. Special MOS levels. V_{CIL} = 0.9V and V_{CIH} = 3.0V.
2. Manchester only.
3. Manchester. Needs 50% duty cycle.
4. 1 TTL load + 50 pF.
5. 1 TTL load + 100 pF.
6. NRZ only.
7. Abnormal end of transmission—CTS expires before RTS.
8. Normal end to transmission.
9. Programmable value:
   T71 = N_{IFS} • T36
   where: N_{IFS} = the IFS configuration value
   (if N_{IFS} is less than 12 then N_{IFS} is forced to 12).
10. Programmable value:
    T64 = (N_{CDF} • T36) + x • T36
    (If the collision occurs after the preamble)
    where:
    N_{CDF} = the collision detect filter configuration value,
    and
    x = 12, 13, 14, or 15
11. T68 = 32 • T36
12. Programmable value:
    T67 = (N_{CSF} • T36) + x • T36
    where: N_{CSF} = the Carrier Sense Filter configuration value, and
    x = 12, 13, 14, or 15
13. To guarantee recognition on the next clock.

82596CA BUS OPERATION

The following figures show the 82596CA basic bus cycle and basic burst cycle.

Please refer to the *32-Bit LAN Component User's Manual.*

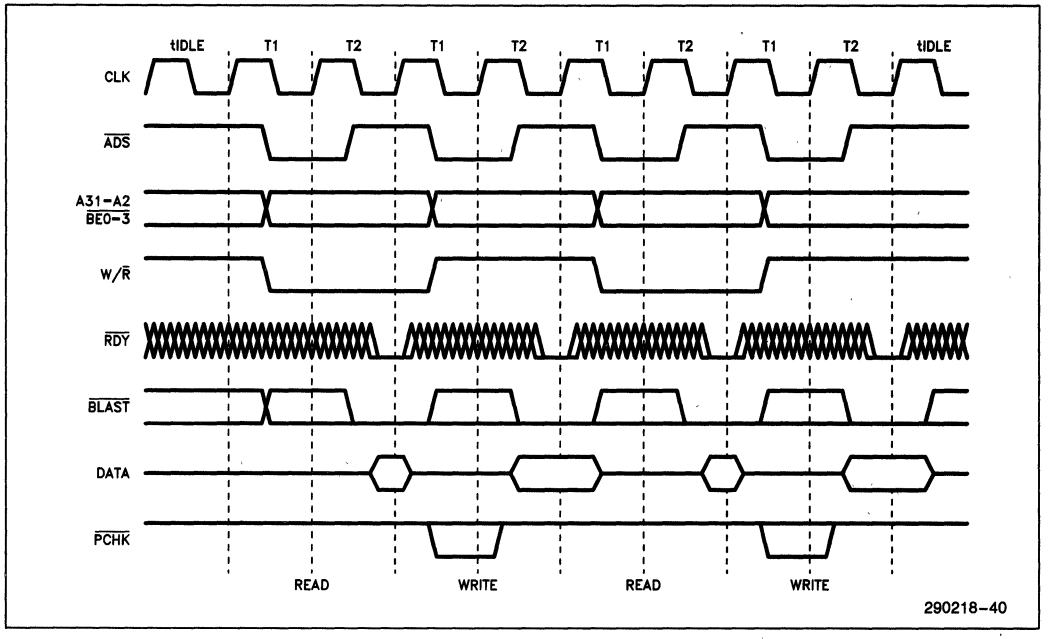


Figure 44. Basic 82596CA Bus Cycle

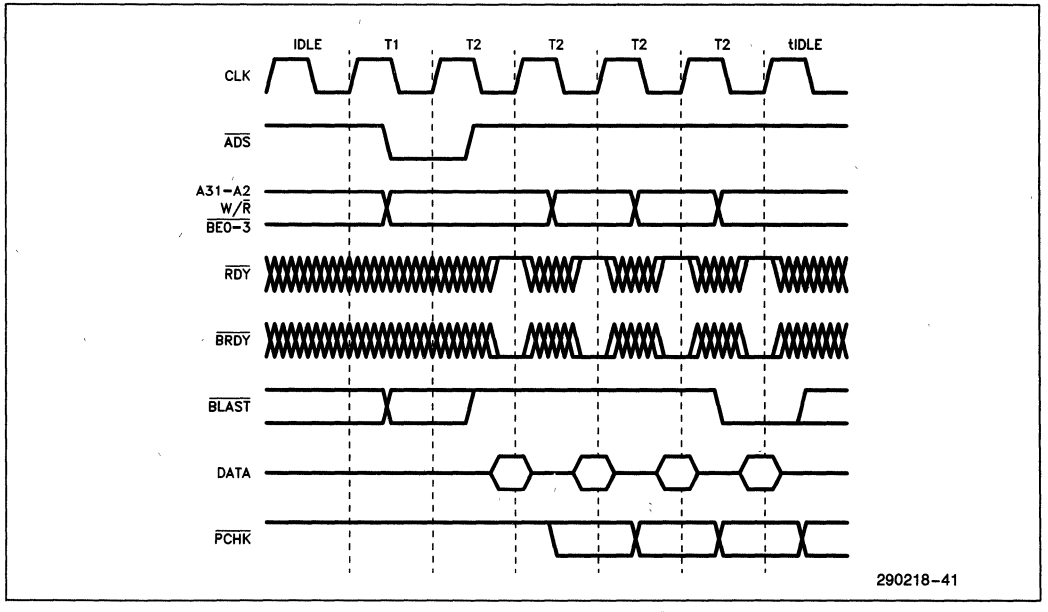
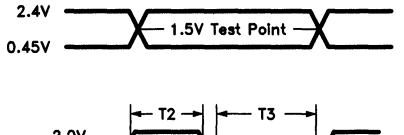


Figure 45. Basic 82596CA Burst Cycle

SYSTEM INTERFACE A.C. TIMING CHARACTERISTICS

The measurements should be done at:

- T_C = 0°C to +85°C, V_{CC} = 5V ±10%, C = 50 pF unless otherwise specified.
- A.C. testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0".
- Timing measurements are made at 1.5V for both logic "1" and "0".
- Rise and Fall time of inputs and outputs signals are measured between 0.8V and 2.0V respectively unless otherwise specified.
- All timings are relative to CLK crossing the 1.5V level.
- All A.C. parameters are valid only after 100 μs from power up.



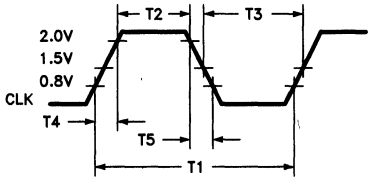
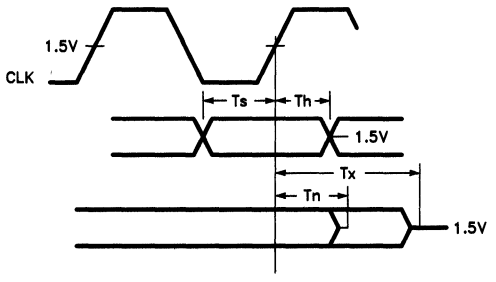


Figure 46. CLK Timings

Two types of timing specifications are presented below:

1. Input Timing—minimum setup and hold times.
2. Output Timings—output delays and float times from CLK rising edge.

Figure 47 defines how the measurements should be done:



LEGEND:
Ts = Input Setup Time
Th = Input Hold Time
Tn = Minimum output delay or Mininum float delay
Tx = Maximum output delay or Maximum float delay

Figure 47. Drive Levels and Measurements Points for A.C. Specifications

Ts = T13, T15, T17, T19, T21, T23, T27, T29, T31
Th = T14, T16, T18, T20, T22, T22a, T24, T28, T30, T32
Tn = T6, T6a, T7, T8, T9, T10, T11, T12, T25
Tx = T6, T6a, T7, T8, T9, T10, T11, T12, T25

INPUT WAVEFORMS

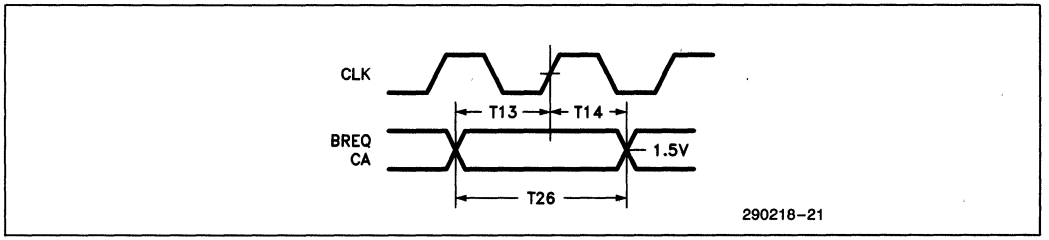


Figure 48. CA and BREQ Input Timing

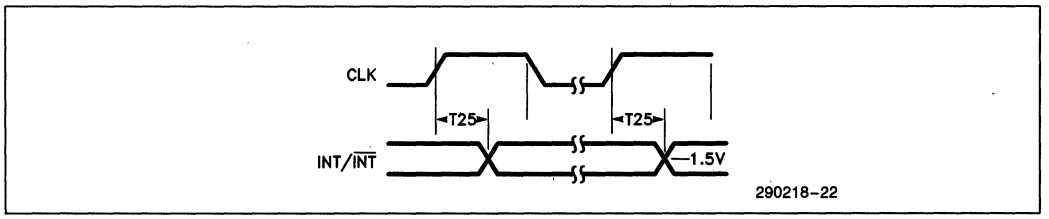


Figure 49. INT/INT Output Timing

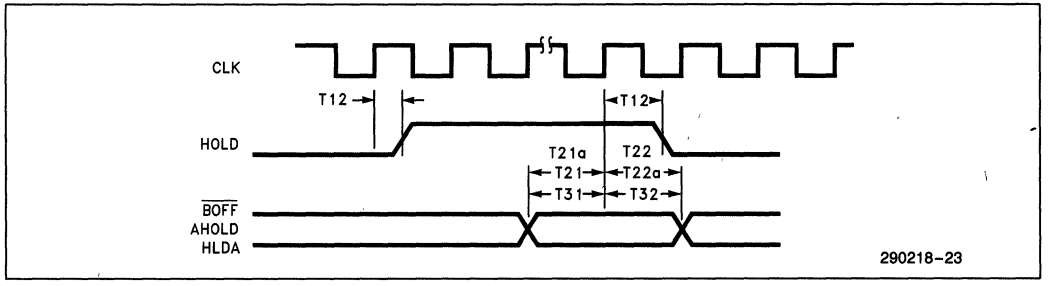


Figure 50. HOLD/HLDA Timings

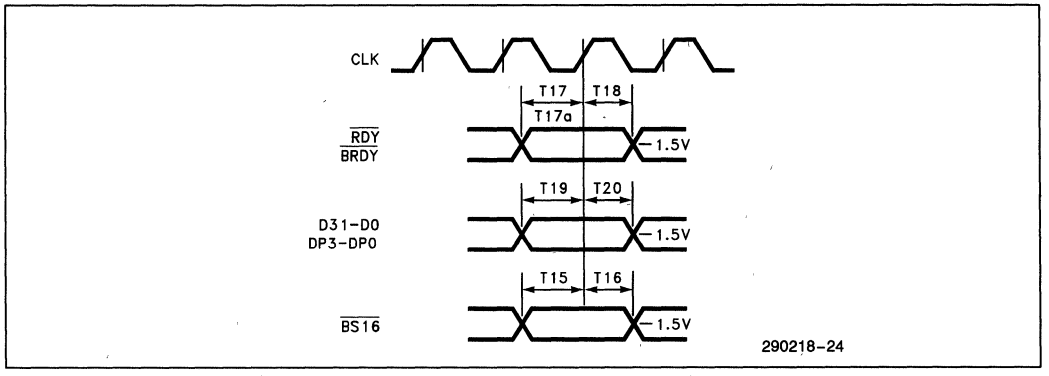


Figure 51. Input Setup and Hold Time

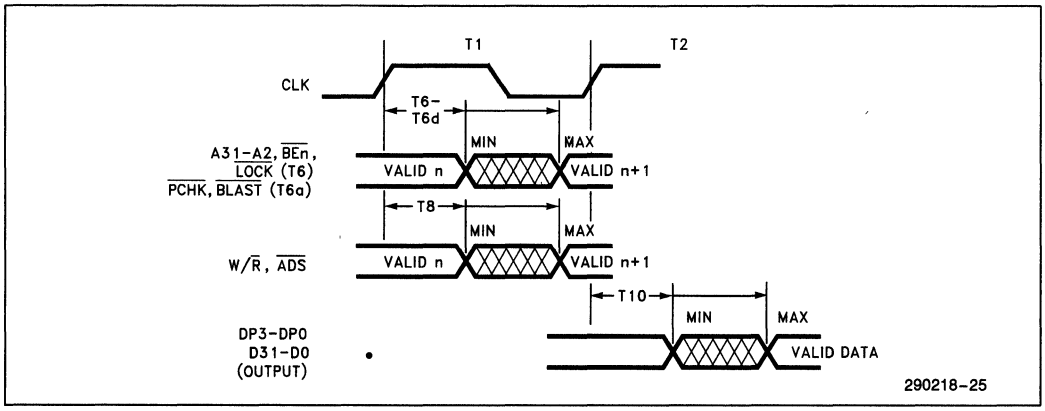


Figure 52. Output Valid Delay Timing

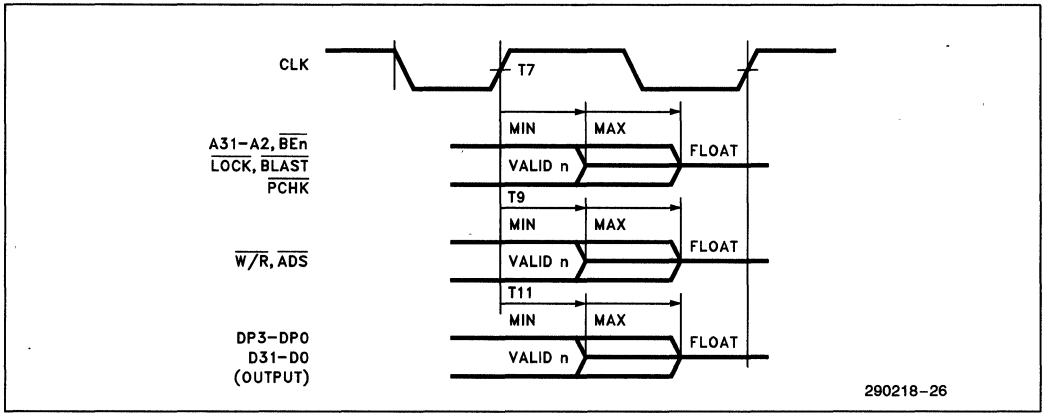


Figure 53. Output Float Delay Timing

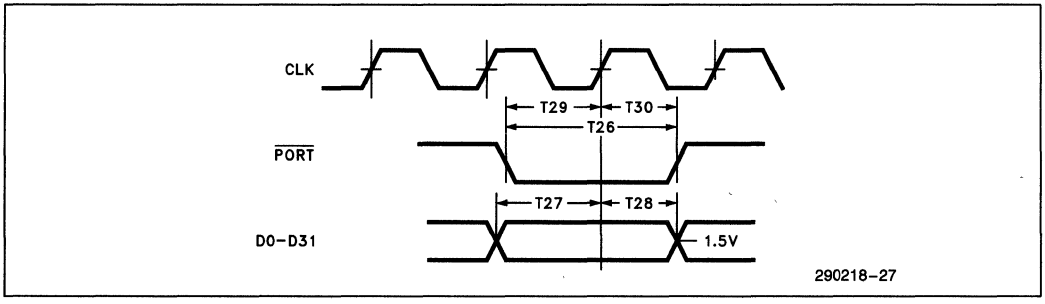


Figure 54. PORT Setup and Hold Time

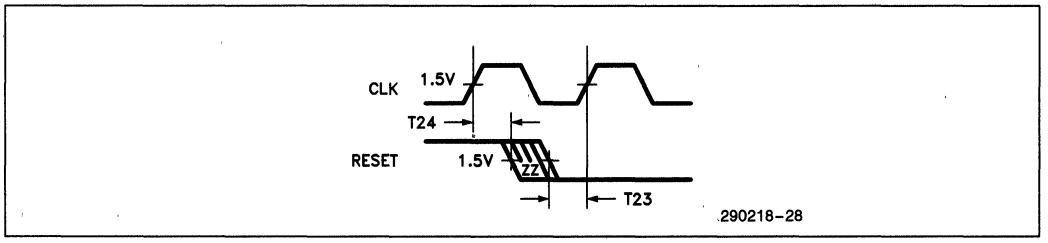


Figure 55. RESET Input Timing

SERIAL AC TIMING CHARACTERISTICS

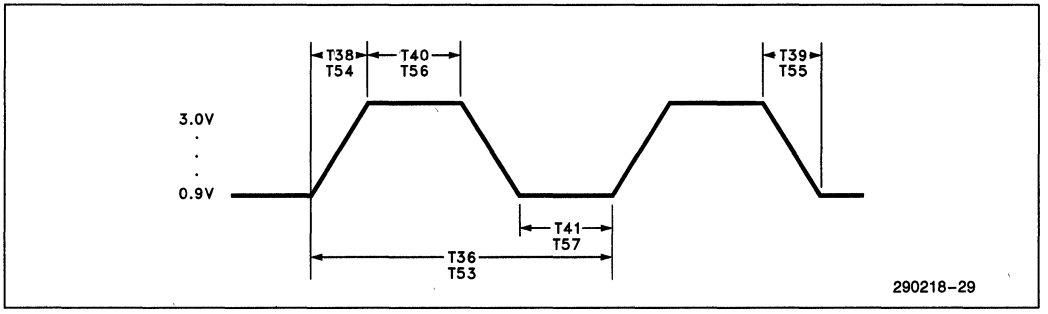


Figure 56. Serial Input Clock Timing

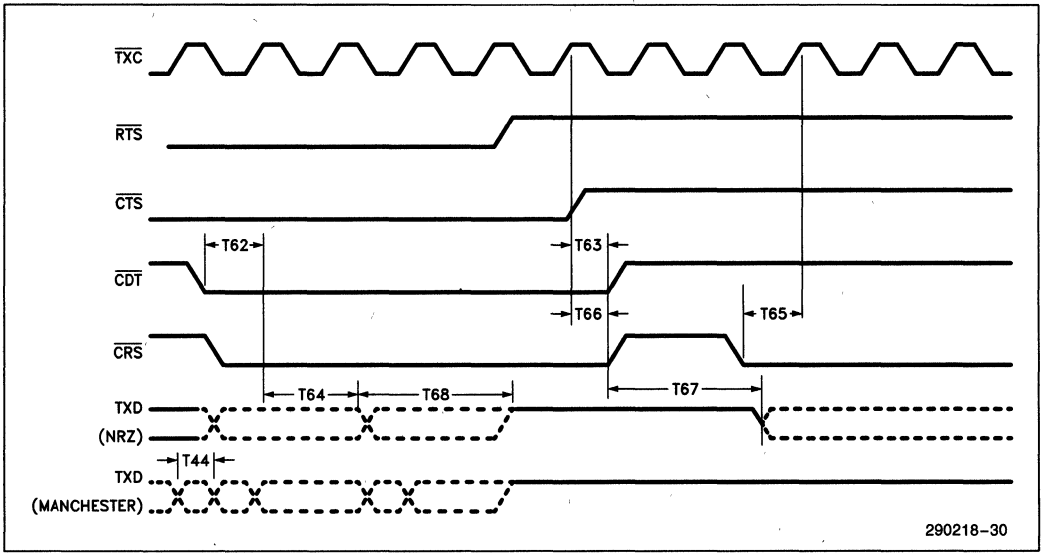


Figure 57. Transmit Data Waveforms

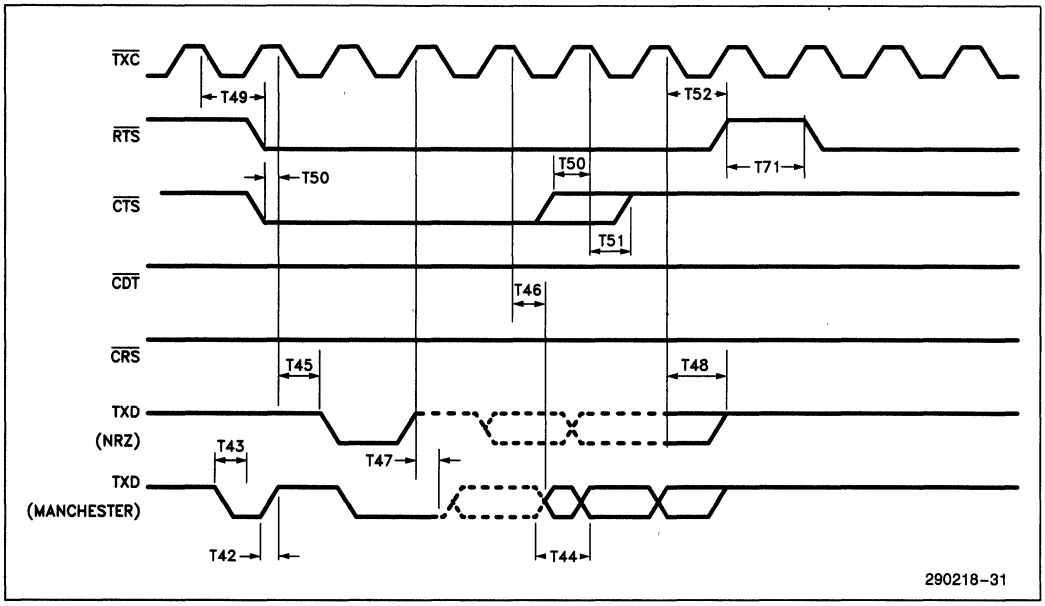


Figure 58. Transmit Data Waveforms

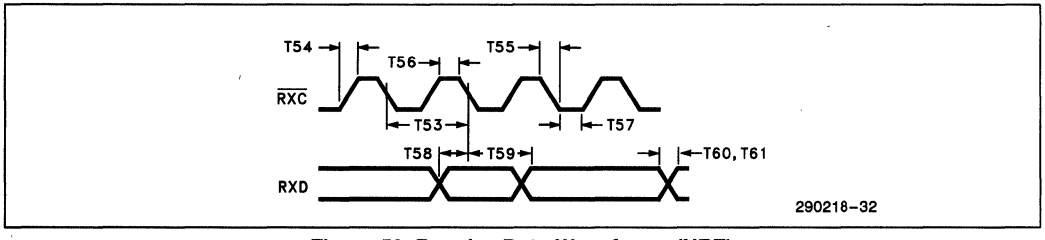


Figure 59. Receive Data Waveforms (NRZ)

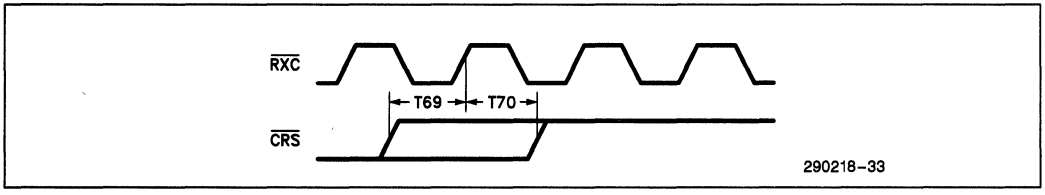
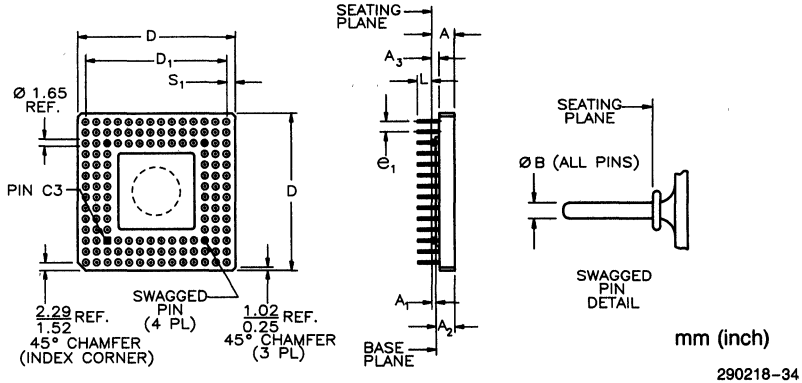


Figure 60. Receive Data Waveforms (CRS)

OUTLINE DIAGRAMS

132 LEAD CERAMIC PIN GRID ARRAY PACKAGE INTEL TYPE A



Family: Ceramic Pin Grid Array Package						
Symbol	**Millimeters**			**Inches**		
	Min	**Max**	**Notes**	**Min**	**Max**	**Notes**
A	3.56	4.57		0.140	0.180	
A_1	0.76	1.27	Solid Lid	0.030	0.050	Solid Lid
A_2	2.67	3.43	Solid Lid	0.105	0.135	Solid Lid
A_3	1.14	1.40		0.045	0.055	
B	0.43	0.51		0.017	0.020	
D	36.45	37.21		1.435	1.465	
D_1	32.89	33.15		1.295	1.305	
e_1	2.29	2.79		0.090	0.110	
L	2.54	3.30		0.100	0.130	
N	132			132		
S_1	1.27	2.54		0.050	0.100	
ISSUE	IWS 10/12/88					

Intel Case Outline Drawings
Plastic Quad Flat Pack (PQFP)
0.025 Inch (0.635mm) Pitch

Symbol	Description	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
N	Leadcount	68		84		100		132		164		196	
A	Package Height	0.160	0.170	0.160	0.170	0.160	0.170	0.160	0.170	0.160	0.170	0.160	0.170
A1	Standoff	0.020	0.030	0.020	0.030	0.020	0.030	0.020	0.030	0.020	0.030	0.020	0.030
D, E	Terminal Dimension	0.675	0.685	0.775	0.785	0.875	0.885	1.075	1.085	1.275	1.285	1.475	1.485
D1, E1	Package Body	0.547	0.553	0.647	0.653	0.747	0.753	0.947	0.953	1.147	1.153	1.347	1.353
D2, E2	Bumper Distance	0.697	0.703	0.797	0.803	0.897	0.903	1.097	1.103	1.297	1.303	1.497	1.503
D3, E3	Lead Dimension	0.400 REF		0.500 REF		0.600 REF		0.800 REF		1.000 REF		1.200 REF	
D4, E4	Foot Radius Location	0.623	0.637	0.723	0.737	0.823	0.837	1.023	1.037	1.223	1.237	1.423	1.437
L1	Foot Length	0.020	0.030	0.020	0.030	0.020	0.030	0.020	0.030	0.020	0.030	0.020	0.030
Issue	IWS Preliminary 12/12/88												INCH

Symbol	Description	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
N	Leadcount	68		84		100		132		164		196	
A	Package Height	4.06	4.32	4.06	4.32	4.06	4.32	4.06	4.32	4.06	4.32	4.06	4.32
A1	Standoff	0.51	0.76	0.51	0.76	0.51	0.76	0.51	0.76	0.51	0.76	0.51	0.76
D, E	Terminal Dimension	17.15	17.40	19.69	19.94	22.23	22.48	27.31	27.56	32.39	32.64	37.47	37.72
D1, E1	Package Body	13.89	14.05	16.43	16.59	18.97	19.13	24.05	24.21	29.13	29.29	34.21	34.37
D2, E2	Bumper Distance	17.70	17.85	20.24	20.39	22.78	22.93	27.86	28.01	32.94	33.09	38.02	38.18
D3, E3	Lead Dimension	10.16 REF		12.70 REF		15.24 REF		20.32 REF		25.40 REF		30.48 REF	
D4, E4	Foot Radius Location	15.82	16.17	18.36	18.71	21.25	21.25	25.89	26.33	31.06	31.41	36.14	36.49
L1	Foot Length	0.51	0.76	0.51	0.76	0.51	0.76	0.51	0.76	0.51	0.76	0.51	0.76
Issue	IWS Preliminary 12/12/88												mm

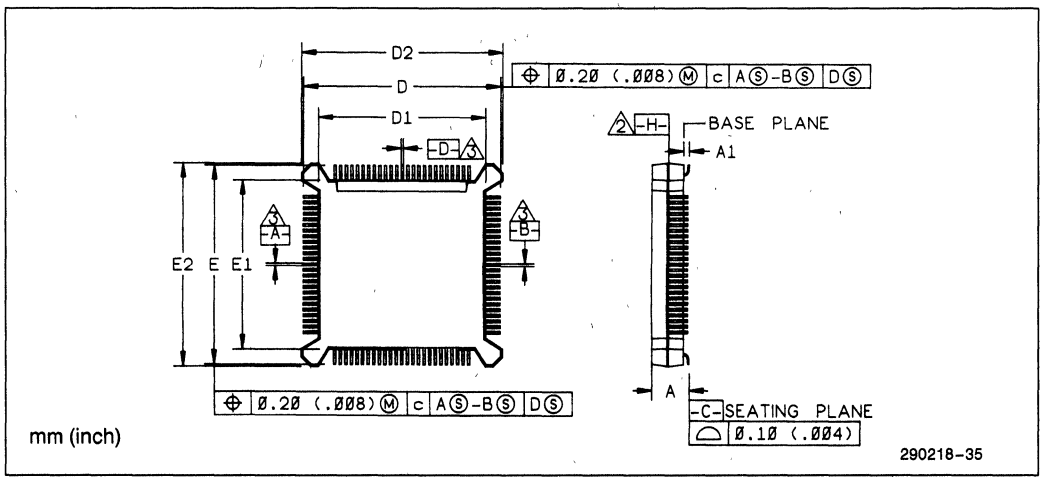


Figure 61. Principal Dimensions and Datums

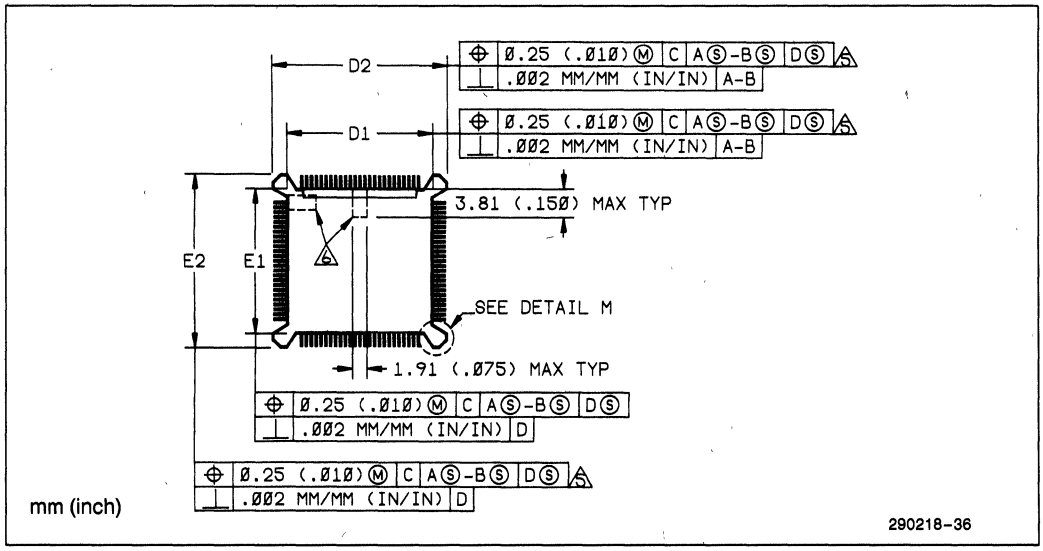


Figure 62. Molded Details

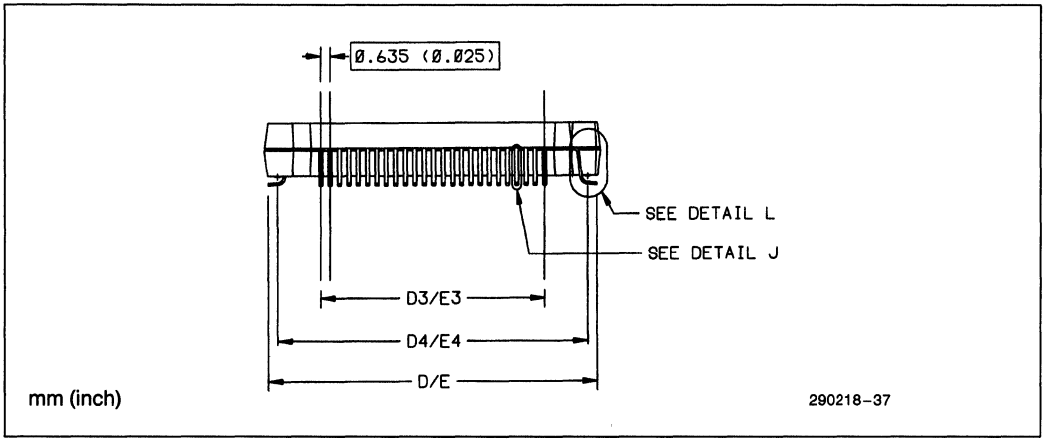


Figure 63. Terminal Details

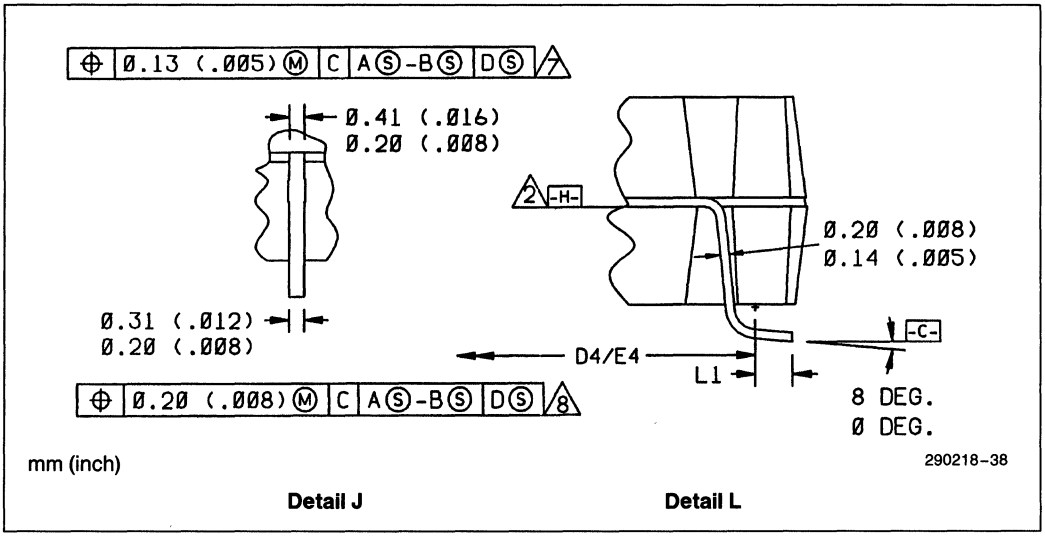
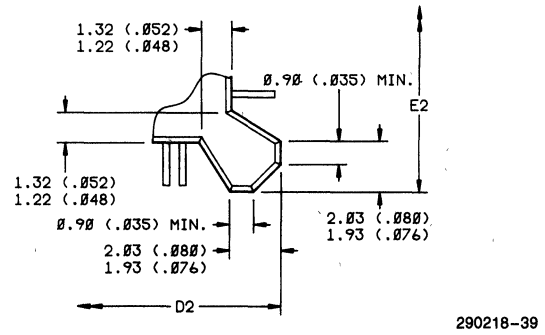


Figure 64. Typical Lead



mm (inch)

Figure 65. Detail M

REVISION SUMMARY

The following represents the key differences between version 004 and version 005 of the 82596CA Data Sheet.

1.. Timings added for -16 MHz and -20 MHz specfications.

The following represents the key differences between version 005 and version 006 of the 82596CA Data Sheet.

1. A description of the 82596CA C-stepping enhancements was added and the 82596CA B-step information was removed.
2. Description of BOFF pin changed. BOFF may be asserted in T1 in the 82596 C-step.
3. Recommendation to use only one type of buffer (either Simplified or Flexible) in any given linked list.
4. Added detailed description regarding operation or RCVCDT counter.
5. Added New Enhanced Big Endian Mode section. The New Enhanced Big Endian Mode applies only to the 82596 C-stepping.
6. Added programming recommendations regarding RU and CU Start commands. These warn against Starting the CU while it is Active and Starting the RU while it is Ready.
7. Emphasized that the TDR command is a static command and should not be used in an active network.
8. Improved 82596CA C-step timings were added for all speeds.

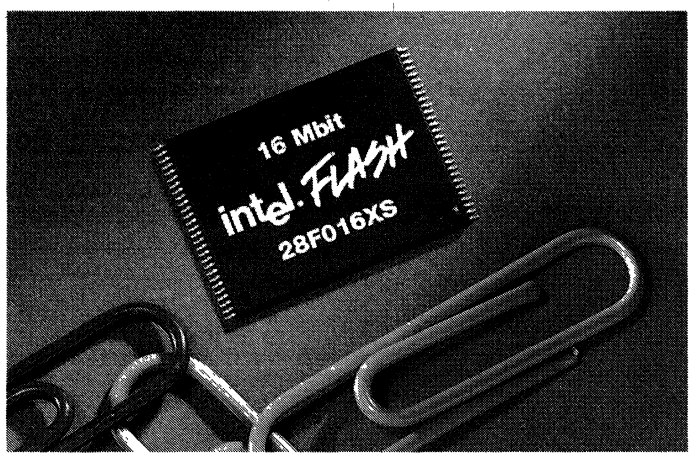
28F016XS
16-MBIT (1-MBIT x 16, 2-MBIT x 8) SYNCHRONOUS FLASH MEMORY

- **Effective 0-Wait-State Performance up to 33 MHz**
- **SmartVoltage Technology**
  - **— User-Selectable 3.3V or 5V V_{CC}**
  - **— User-Selectable 5V or 12V V_{PP}**
- **30.8 MB/sec Burst Write Transfer Rate**
- **0.48 MB/sec Sustainable Write Transfer Rate**
- **Configurable x8 or x16 Operation**
- **16 Separately-Erasable/Lockable 128-KByte Blocks**
- **56-Lead TSOP Type I Package**
- **Backward Compatible with 28F016SA/SV, 28F008SA Command-Set**
- **Revolutionary Architecture**
  - **— Synchronous Pipelined Reads**
  - **— Multiple Command Execution**
  - **— Write During Erase**
  - **— Page Buffer Write**
- **2 μA Typical Deep Power-Down**
- **1 mA Typical Active I_{CC} Current in Static Mode**
- **1 Million Erase Cycles per Block**
- **State-of-the-Art 0.6 μm ETOX™ IV Flash Technology**

2

Intel's 28F016XS 16-Mbit Flash memory is a revolutionary architecture which is the ideal choice for designing truly revolutionary high-performance products. Combining very high read performance with the intrinsic non-volatility of flash memory, the 28F016XS eliminates the traditional redundant memory paradigm of shadowing code from a slow nonvolatile storage source to a faster execution memory, such as DRAM, for improved system performance. The innovative capabilities of the 28F016XS enable the design of direct-execute code and mass storage data/file flash memory systems.

The 28F016XS is the highest performance high density nonvolatile read/write flash memory solution available today. Its synchronous pipelined read interface, flexible V_{CC} and V_{PP} voltages, extended cycling, fast write and read performance, symmetrically blocked architecture, and selective block locking provide a highly flexible memory component suitable for resident flash component arrays on the system board or SIMMs. The synchronous pipelined interface and x8/x16 architecture of the 28F016XS allow easy interface with minimal glue logic to a wide range of processors/buses, providing effective 0-wait-state read performance up to 33 MHz. The 28F016XS's dual read voltage allows the same component to operate at either 3.3V or 5.0V V_{CC}. Programming voltage at 5V V_{PP} minimizes external circuitry in minimal-chip, space critical designs, while the 12V V_{PP} option maximizes write/erase performance. Its high read performance combined with flexible block locking enable both storage and execution of operating systems/application software and fast access to large data tables. The 28F016XS is manufactured on Intel's 0.6 μm ETOX™ IV process technology.



290532-21

November 1994
Order Number: 290532-001

28F016XS
16-MBIT (1-MBIT x 16, 2-MBIT X 8) SYNCHRONOUS FLASH MEMORY

CONTENTS

CONTENTS

1.0 INTRODUCTION

The documentation of the Intel 28F016XS Flash memory device includes this datasheet, a detailed user's manual, a number of application notes and design tools, all of which are referenced at the end of this datasheet.

The datasheet is intended to give an overview of the chip feature-set and of the operating AC/DC specifications. The 16-Mbit Flash Product Family User's Manual provides complete descriptions of the user modes, system interface examples and detailed descriptions of all principles of operation. It also contains the full list of software algorithm flowcharts, and a brief section on compatibility with the Intel 28F008SA.

1.1 Product Overview

The 28F016XS is a high-performance, 16-Mbit (16,777,216-bit) block erasable nonvolatile random access memory organized as either 1-MWord x 16 or 2-MByte x 8, subdivided into even and odd banks. Address A_1 makes the bank selection. The 28F016XS includes sixteen 128-KByte (131,072 byte) blocks or sixteen 64-KWord (65,536 word) blocks. Chip memory maps for x8 and x16 modes are shown in Figures 3 and 4.

The implementation of a new architecture, with many enhanced features, will improve the device operating characteristics and result in greater product reliability and ease-of-use as compared to other flash memories. Significant features of the 28F016XS as compared to previous asynchronous flash memories include:

- Synchronous Pipelined Read Interface
- Significantly Improved Read and Write Performance
- SmartVoltage Technology
  - — User-Selectable 5.0V or 12.0 V_{PP}
- Internal 3.3V/5.0V V_{CC} Detection Circuitry
- Block Write/Erase Protection

The 28F016XS's synchronous pipelined interface dramatically raises read performance far beyond previously attainable levels. Addresses are synchronously latched and data is read from a 28F016XS bank every 30 ns. This capability translates to 0-wait-state reads at clock rates up to 33 MHz at 5V V_{CC}, after an initial address pipeline fill delay and assuming even and odd banks within the flash memory are alternately accessed. Data is latched and driven valid 20 ns (t_{CHQV}) after a rising CLK edge. The 28F016XS is capable of operating up to 66 MHz (5V V_{CC}), and the programmable SFI Configuration enables system design flexibility optimizing the 28F016XS to a specific system clock frequency. See Section 4.9, SFI Configuration Table, for specific SFI Configurations for given operating frequencies.

The SFI Configuration optimizes the 28F016XS for a wide range of system operating frequencies. The default SFI Configuration is 4, which allows system boot from the 28F016XS at any frequency up to 66 MHz at 5V V_{CC}. After initiating an access, data is latched and will begin driving on the data outputs after a CLK count corresponding to the SFI Configuration has elapsed. The 28F016XS will hold data valid until CE# or OE# is deactivated or a CLK count corresponding to the SFI Configuration for a subsequent access has elapsed.

The CLK and ADV# inputs, new to the 28F016XS in comparison to previous flash memories, control address latching and device synchronization during read operations. CLK input controls the device latencies, times out the SFI Configuration and synchronizes data outputs. ADV# indicates the presence of a valid address on the 28F016XS address inputs. During read operations, addresses are latched and accesses are initiated on a rising CLK edge in conjunction with ADV# low. Both CLK and ADV# are ignored by the 28F016XS during write operations.

The 28F016XS incorporates SmartVoltage technology, providing V_{CC} operation at both 3.3V and 5.0V and program and erase capability at V_{PP} = 12.0V or 5.0V. Operating at V_{CC} = 3.3V, the 28F016XS consumes less than one half the power consumption at 5.0V V_{CC}, while 5.0V V_{CC} provides highest read performance capability. V_{PP} operation at 5.0V eliminates the need for a separate 12.0V converter, while the V_{PP} = 12.0V option maximizes write/erase performance. In addition to the flexible program and erase voltages, the dedicated V_{PP} gives complete code protection with $V_{PP} \leq V_{PPLK}$.

Internal 3.3V or 5.0V V_{CC} detection automatically configures the device internally for optimized 3.3V or 5.0V Read/Write operation. Hence, the 28F016SA's 3/5# pin is not required and is a no-connect (NC) on the 28F016XS maintaining backwards-compatibility between components.

A Command User Interface (CUI) serves as the system interface between the microprocessor or microcontroller and the internal memory operation.

Internal Algorithm Automation allows Byte/Word Writes and Block Erase operations to be executed using a Two-Write command sequence to the CUI in the same way as the 28F008SA 8-Mbit FlashFile™ memory.

A super-set of commands has been added to the basic 28F008SA command-set to achieve higher write performance and provide additional capabilities. These new commands and features include:

- Page Buffer Writes to Flash
- Command Queuing Capability
- Automatic Data Writes during Erase
- Software Locking of Memory Blocks
- Two-Byte Successive Writes in 8-bit Systems
- Erase All Unlocked Blocks

Writing of memory data is performed in either byte or word increments, typically within 6 µsec at 12.0V V_{PP}, which is a 33% improvement over the 28F008SA. A Block Erase operation erases one of the 16 blocks in typically 1.2 sec, independent of the other blocks.

Each block can be written and erased a minimum of 100,000 cycles. Systems can achieve one million Block Erase Cycles by providing wear-leveling algorithms and graceful block retirement. These techniques have already been employed in many flash file systems and hard disk drive designs.

The 28F016XS incorporates two Page Buffers of 256 bytes (128 words) each to allow page data writes. This feature can improve a system write performance by up to 4.8 times over previous flash memory devices, which have no Page Buffers.

All operations are started by a sequence of Write commands to the device. Three Status Registers (described in detail later in this datasheet) and a RY/BY# output pin provide information on the progress of the requested operation.

While the 28F008SA requires an operation to complete before the next operation can be requested, the 28F016XS allows queuing of the next operation while the memory executes the current operation. This eliminates system overhead when writing several bytes in a row to the array or erasing several blocks at the same time. The 28F016XS can also perform Write operations to one block of memory while performing Erase of another block.

The 28F016XS provides selectable block locking to protect code or data such as direct-executable operating systems or application code. Each block has an associated nonvolatile lock-bit which determines the lock status of the block. In addition, the 28F016XS has a master Write Protect pin (WP#) which prevents any modifications to memory blocks whose lock-bits are set.

The 28F016XS contains three types of Status Registers to accomplish various functions:

- A Compatible Status Register (CSR) which is 100% compatible with the 28F008SA FlashFile memory Status Register. The CSR, when used alone, provides a straightforward upgrade capability to the 28F016XS from a 28F008SA-based design.
- A Global Status Register (GSR) which informs the system of command Queue status, Page Buffer status, and overall Write State Machine (WSM) status.
- 16 Block Status Registers (BSRs) which provide block-specific status information such as the block lock-bit status.

The GSR and BSR memory maps for Byte-Wide and Word-Wide modes are shown in Figures 5 and 6.

The 28F016XS incorporates an open drain RY/BY# output pin. This feature allows the user to OR-tie many RY/BY# pins together in a multiple memory configuration such as a Resident Flash Array. The RY/BY# output pin employs five distinct configurations, which are enabled via special CUI commands and are described in detail in the 16-Mbit Flash Product Family User's Manual.

The 28F016XS also incorporates a dual chip-enable function with two input pins, CE_0# and CE_1#. These pins have exactly the same functionality as the regular chip-enable pin, CE#, on the 28F008SA. For minimum chip designs, CE_1# may be tied to ground and system logic may use CE_0# as the chip enable input. The 28F016XS uses the logical combination of these two signals to enable or disable the entire chip. Both CE_0# and CE_1# must be active low to enable the device. If either one becomes inactive, the chip will be disabled. This feature, along with the open drain RY/BY# pin, allows the system designer to reduce the number of control pins used in a large array of 16-Mbit devices.

The BYTE# pin allows either x8 or x16 read/writes to the 28F016XS. BYTE# at logic low selects 8-bit mode with address A_0 selecting between low byte and high byte. On the other hand, BYTE# at logic high enables 16-bit operation with address A_1 becoming the lowest order address and address A_0 is not used (don't care). A device block diagram is shown in Figure 1.

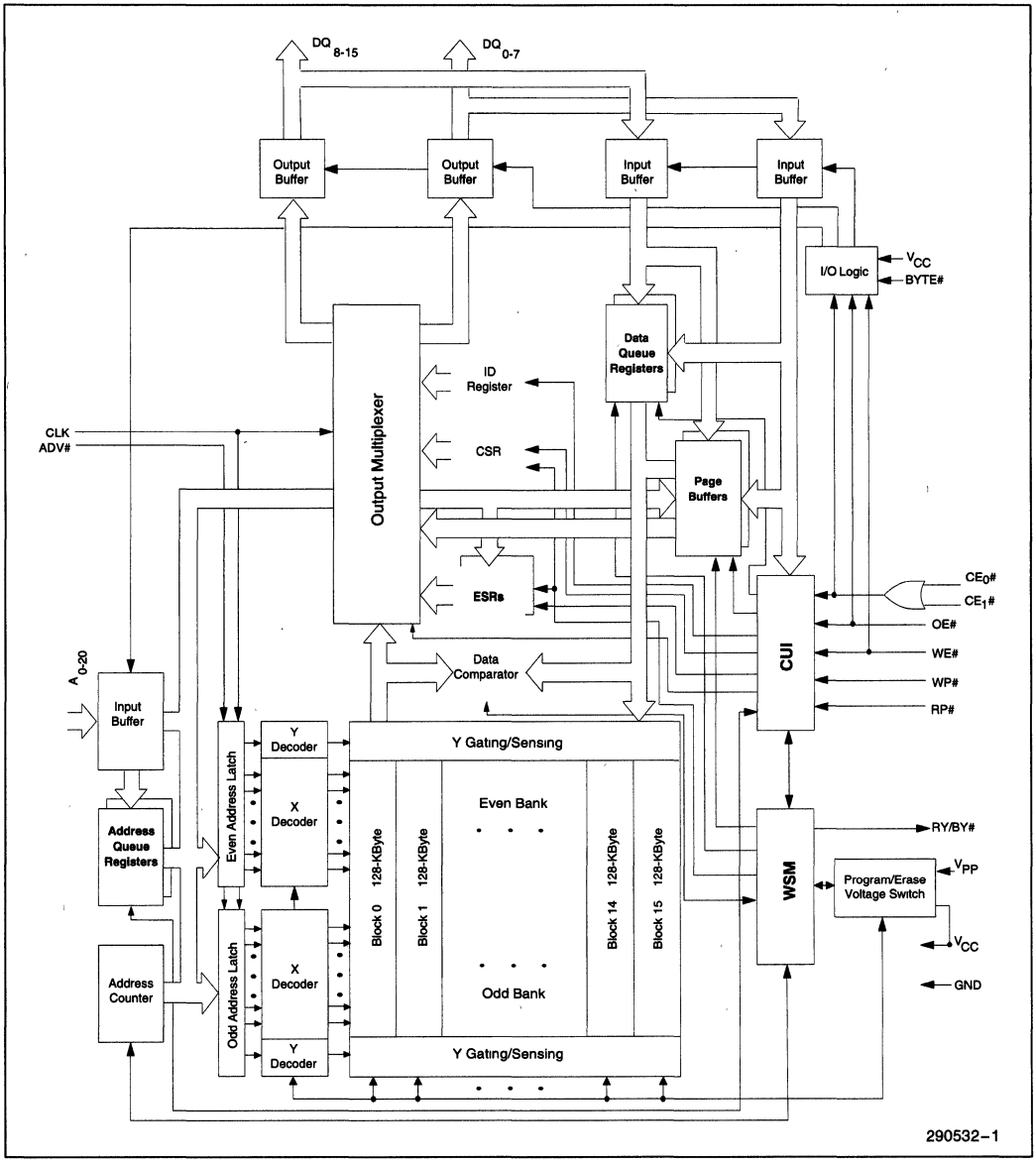


Figure 1. 28F016XS Block Diagram
Architectural Evolution Includes Synchronous Pipelined Read Interface, SmartVoltage Technology, Page Buffers, Queue Registers and Extended Status Registers

The 28F016XS incorporates an Automatic Power Saving (APS) feature, which substantially reduces the active current when the device is in static mode of operation (addresses not switching). In APS mode, the typical I_{CC} current is 1 mA at 5.0V (3 mA at 3.3V).

A deep power-down mode of operation is invoked when the RP# (called PWD# on the 28F008SA) pin transitions low. This mode brings the device power consumption to less than 2.0 μA, typically, and provides additional write protection by acting as a device reset pin during power transitions. A reset time of 300 ns is required from RP# switching high before latching an address into the 28F016XS. In the Deep Power-Down state, the WSM is reset (any current operation will abort) and the CSR, GSR and BSR registers are cleared.

A CMOS standby mode of operation is enabled when either CE_0# or CE_1# transitions high and RP# stays high with all input control pins at CMOS levels. In this mode, the device typically draws an I_{CC} standby current of 70 μA at 5V V_{CC}.

The 28F016XS will be available in a 56-Lead, 1.2mm thick, 14mm x 20mm TSOP Type I package. The package's form factor and pinout allows for very high board layout densities.

2.0 DEVICE PINOUT

The 28F016XS is pinout compatible with the 28F016SA/SV 16-Mbit FlashFile™ memory component, providing an performance upgrade path to the 28F016XS. The 28F016XS 56-Lead TSOP pinout configuration is shown in Figure 2.

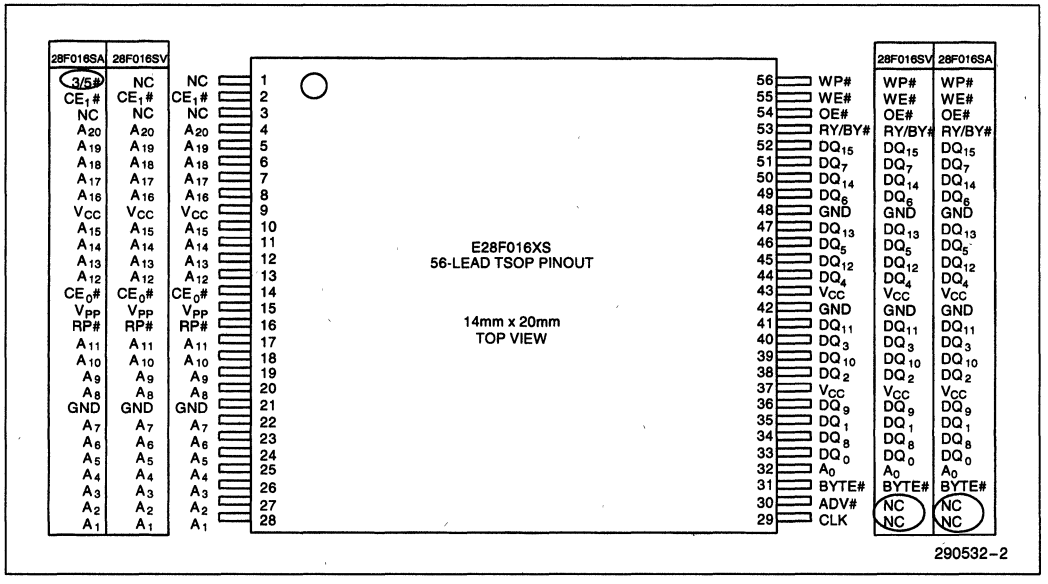


Figure 2. 28F016XS 56-Lead TSOP Pinout Configuration Shows Compatibility with the 28F016SA/SV, Allowing for Easy Performance Upgrades from Existing 16-Mbit Designs

2.1 Lead Descriptions

Symbol	Type	Name and Function
A_0	Input	**BYTE-SELECT ADDRESS:** Selects between high and low byte when device is in x8 mode. This address is latched in x8 Data Writes and ignored in x16 mode (i.e., the A_0 input buffer is turned off when BYTE# is high).
A_1	Input	**BANK-SELECT ADDRESS:** Selects an even or odd bank in a selected block. A 128-KByte block is subdivided into an even and odd bank. A_1 = 0 selects the even bank and A_1 = 1 selects the odd bank, in both byte-wide mode and word-wide mode device configurations.
A_2–A_{16}	Input	**WORD-SELECT ADDRESSES:** Select a word within one 128-KByte block. Address A_1 and A_{7-16} select 1 of 2048 rows, and A_{2-6} selects 16 of 512 columns. These addresses are latched during both data reads and writes.
A_{17}–A_{20}	Input	**BLOCK-SELECT ADDRESSES:** Select 1 of 16 Erase blocks. These addresses are latched during Data Writes, Erase and Lock-Block operations.
DQ_0–DQ_7	Input Output	**LOW-BYTE DATA BUS:** Inputs data and commands during CUI write cycles. Outputs array, buffer, identifier or status data in the appropriate read mode. Floated when the chip is de-selected or the outputs are disabled.
DQ_8–DQ_{15}	Input Output	**HIGH-BYTE DATA BUS:** Inputs data during x16 Data-Write operations. Outputs array, buffer or identifier data in the appropriate read mode; not used for Status Register reads. Outputs floated when the chip is de-selected, the outputs are disabled (OE# = V_{IH}) or BYTE# is driven active.
CE_0#, CE_1#	Input	**CHIP ENABLE INPUTS:** Activate the device's control logic, input buffers, decoders and sense amplifiers. With either CE_0# or CE_1# high, the device is de-selected and power consumption reduces to standby levels upon completion of any current Data-Write or Erase operations. Both CE_0# and CE_1# must be low to select the device. All timing specifications are the same for both signals. Device Selection occurs with the latter falling edge of CE_0# or CE_1#. The first rising edge of CE_0# or CE_1# disables the device.
RP#	Input	**RESET/POWER-DOWN:** RP# low places the device in a Deep Power-Down state. All circuits that consume static power, even those circuits enabled in standby mode, are turned off. When returning from Deep Power-Down, a recovery time of t_{PHCH} is required to allow these circuits to power-up. When RP# goes low, any current or pending WSM operation(s) are terminated, and the device is reset. All Status Registers return to ready, clearing all status flags. Exit from Deep Power-Down places the device in read array mode.
OE#	Input	**OUTPUT ENABLE:** Drives device data through the output buffers when low. The outputs float to tri-state off when OE# is high. CE_x# overrides OE#, and OE# overrides WE#.
WE#	Input	**WRITE ENABLE:** Controls access to the CUI, Page Buffers, Data Queue Registers and Address Queue Latches. WE# is active low, and latches both address and data (command or array) on its rising edge. Page Buffer addresses are latched on the falling edge of WE#.

2.1 Lead Descriptions (Continued)

Symbol	Type	Name and Function
CLK	Input	**CLOCK:** Provides the fundamental timing and internal operating frequency. CLK latches input addresses in conjunction with ADV#, times out the desired output SFI Configuration as a function of the CLK period, and synchronizes device outputs. CLK can be slowed or stopped with no loss of data or synchronization. CLK is ignored during write operations.
ADV#	Input	**ADDRESS VALID:** Indicates that a valid address is present on the address inputs. ADV# low at the rising edge of CLK latches the address on the address inputs into the flash memory and initiates a read access to the even or odd bank depending on the state of address A_1. ADV# is ignored during write operations.
RY/BY#	Open Drain Output	**READY/BUSY:** Indicates status of the internal WSM. When low, it indicates that the WSM is busy performing an operation. RY/BY# high indicates that the WSM is ready for new operations (or WSM has completed all pending operations), or Erase is Suspended, or the device is in deep power-down mode. This output is always active (i.e., not floated to tri-state off when OE# or CE_0#, CE_1# are high), except if a RY/BY# Pin Disable command is issued.
WP#	Input	**WRITE PROTECT:** Erase blocks can be locked by writing a nonvolatile lock-bit for each block. When WP# is low, those locked blocks as reflected by the Block-Lock Status bits (BSR.6), are protected from inadvertent Data Writes or Erases. When WP# is high, all blocks can be written or erased regardless of the state of the lock-bits. The WP# input buffer is disabled when RP# transitions low (deep power-down mode).
BYTE#	Input	**BYTE ENABLE:** BYTE# low places device in x8 mode. All data is then input or output on DQ_{0-7}, and DQ_{8-15} float. Address A_0selects between the high and low byte. BYTE# high places the device in x16 mode, and turns off the A_0 input buffer. Address A_1 then becomes the lowest order address.
V_{PP}	Supply	**WRITE/ERASE POWER SUPPLY (12.0V ±0.6V, 5.0V ±0.5V):** For erasing memory array blocks or writing words/bytes/pages into the flash array. V_{PP} = 5.0V ±0.5V eliminates the need for a 12V converter, while the 12.0V ±0.6V option maximizes Write/Erase Performance. Write and Erase attempts are inhibited with V_{PP} at or below 1.5V. Write and Erase attempts with V_{PP} between 1,5V and 4.5V, between 5.5V and 11.4V, and above 12.6V produce spurious results and should not be attempted.
V_{CC}	Supply	**DEVICE POWER SUPPLY (3.3V ±0.3V, 5.0V ±0.5V):** Internal detection configures the device for 3.3V or 5.0V operation. To switch 3.3V to 5.0V (or vice versa), first ramp V_{CC} down to GND, and then power to the new V_{CC} voltage. Do not leave any power pins floating.
GND	Supply	**GROUND FOR ALL INTERNAL CIRCUITRY:** Do not leave any ground pins floating.
NC		**NO CONNECT:** No internal connection to die, lead may be driven or left floating.

3.0 MEMORY MAPS

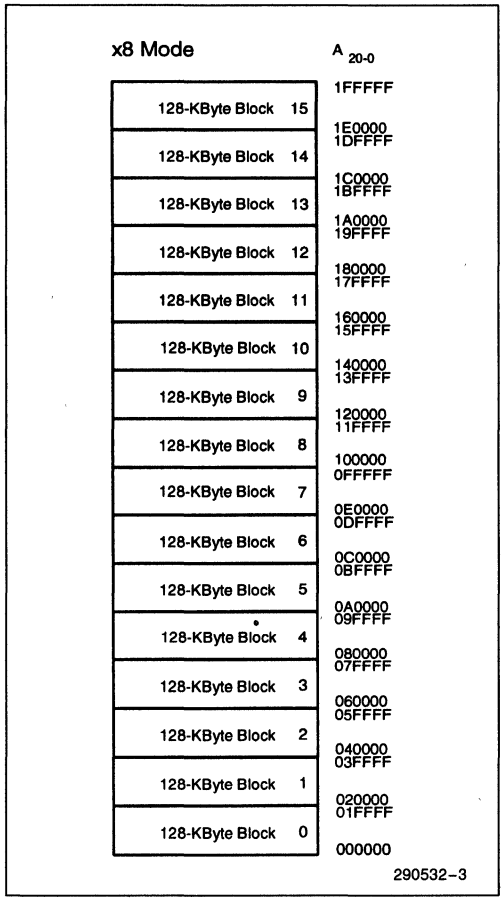


Figure 3. 28F016XS Memory Map (Byte-Wide Mode)

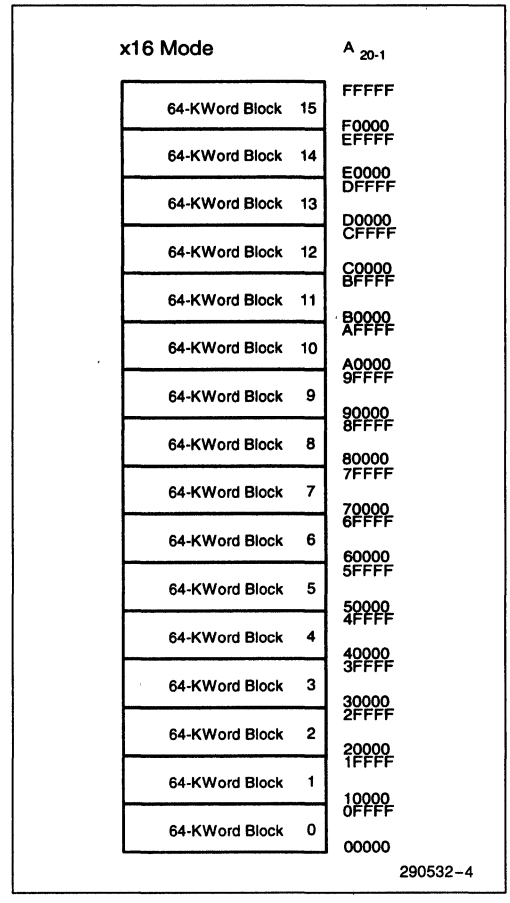


Figure 4. 28F016XS Memory Map (Word-Wide Mode)

2

3.1 Extended Status Register Memory Map

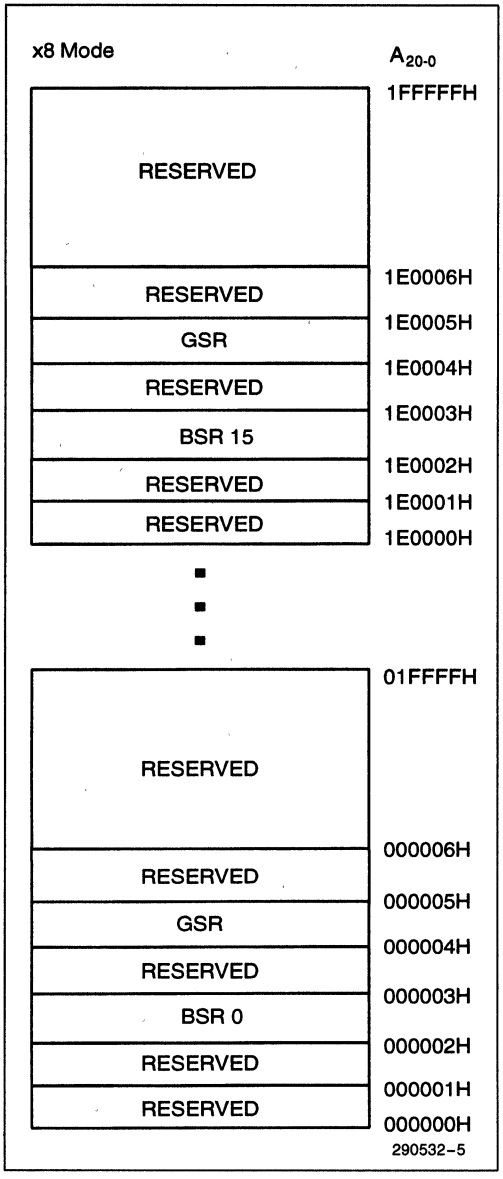


Figure 5. Extended Status Register Memory Map (Byte-Wide Mode)

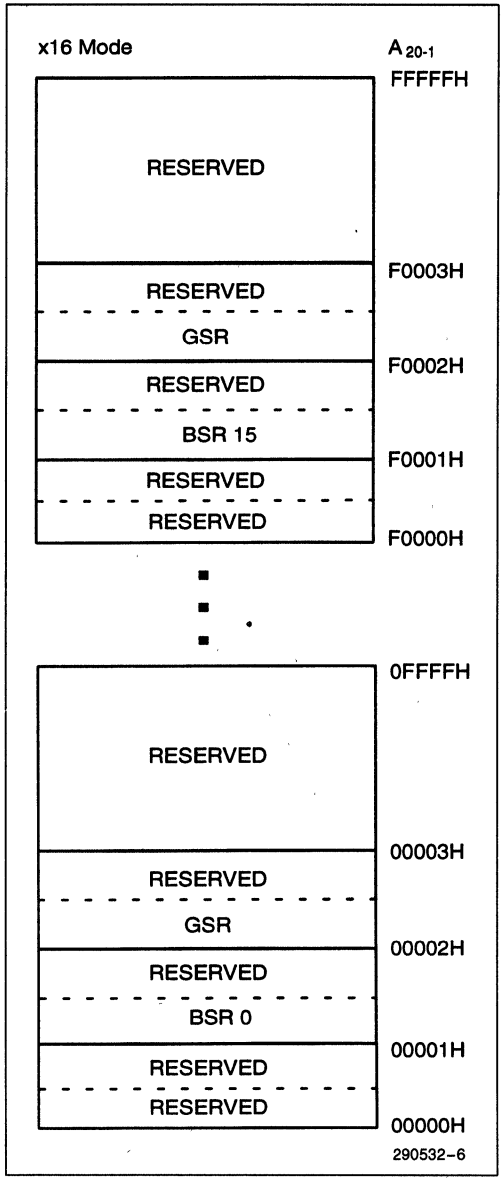


Figure 6. Extended Status Register Memory Map (Word-Wide Mode)

4.0 BUS OPERATIONS, COMMANDS AND STATUS REGISTER DEFINITIONS

4.1 Bus Operations for Word-Wide Mode (BYTE# = V_{IH})

Mode	Notes	RP#	CE_{0-1}#	OE#	WE#	ADV#	CLK	A_1	DQ_{0-15}	RY/BY#
Latch Read Address	1,9,10	V_{IH}	V_{IL}	X	V_{IH}	V_{IL}	↑	X	X	X
Inhibit Latching Read Address	1,9	V_{IH}	V_{IL}	X	V_{IH}	V_{IH}	↑	X	X	X
Read	1,2,7,9	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	↑	X	D_{OUT}	X
Output Disable	1,6,7,9	V_{IH}	V_{IL}	V_{IH}	V_{IH}	X	X	X	High Z	X
Standby	1,6,7,9	V_{IH}	V_{IL}	X	X	X	X	X	High Z	X
Deep Power-Down	1,3	V_{IL}	X	X	X	X	X	X	High Z	V_{OH}
Manufacturer ID	1,4,9	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	↑	V_{IL}	0089H	V_{OH}
Device ID	1,4,8,9	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	↑	V_{IH}	66A8H	V_{OH}
Write	1,5,6,9	V_{IH}	V_{IL}	V_{IH}	V_{IL}	X	X	X	D_{IN}	X

NOTES:
1. X can be V_{IH} or V_{IL} for address or control pins except for RY/BY#, which is either V_{OL} or V_{OH}, or High Z or D_{OUT} for data pins depending on whether or not OE# is active.
2. RY/BY# output is open drain. When the WSM is ready, Erase is suspended or the device is in deep power-down mode. RY/BY# will be at V_{OH} if it is tied to V_{CC} through a resistor. RY/BY# at V_{OH} is independent of OE# while a WSM operation is in progress.
3. RP# at GND ±0.2V ensures the lowest deep power-down current.
4. A_0 and A_1 at V_{IL} provide device manufacturer codes in x8 and x16 modes respectively. A_0 and A_1 at V_{IH} provide device ID codes in x8 and x16 modes respectively. All other addresses are set to zero.
5. Commands for Erase, Data Write, or Lock-Block operations can only be completed successfully when $V_{PP} = V_{PPH1}$ or $V_{PP} = V_{PPH2}$.
6. While the WSM is running, RY/BY# in level-mode (default) stays at V_{OL} until all operations are complete. RY/BY# goes to V_{OH} when the WSM is not busy or in erase suspend mode.
7. RY/BY# may be at V_{OL} while the WSM is busy performing various operations. For example, a Status Register read during a Write operation.
8. The 28F016XS shares an identical device identifier with other Intel Flash memories. Reading this identifier in conjunction with the unique Device Proliferation Code (read from the Page Buffer after writing the Upload Device Configuration command), the 28F016XS can be identified by system software.
9. CE_{0-1}# at V_{IL} is defined as both CE_0# and CE_1# low, and CE_{0-1}# at V_{IH} is defined as either CE_0# or CE_1# high.
10. Addresses are latched on the rising edge of CLK in conjunction with ADV# low. Address $A_1 = 0$ selects the even bank and $A_1 = 1$ selects the odd bank, in both byte-wide mode and word-wide mode device configurations.

4.2 Bus Operations for Byte-Wide Mode (BYTE# = V_{IL})

Mode	Notes	RP#	CE_{0-1}#	OE#	WE#	ADV#	CLK	A_0	DQ_{0-7}	RY/BY#
Latch Read Address	1,9,10	V_{IH}	V_{IL}	X	V_{IH}	V_{IL}	↑	X	X	X
Inhibit Latching Read Address	1,9	V_{IH}	V_{IL}	X	V_{IH}	V_{IH}	↑	X	X	X
Read	1,2,7,9	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	↑	X	D_{OUT}	X
Output Disable	1,6,7,9	V_{IH}	V_{IL}	V_{IH}	V_{IH}	X	X	X	High Z	X
Standby	1,6,7,9	V_{IH}	V_{IH}	X	X	X	X	X	High Z	X
Deep Power-Down	1,3	V_{IL}	X	X	X	X	X	X	High Z	V_{OH}
Manufacturer ID	1,4,9	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	↑	V_{IL}	89H	V_{OH}
Device ID	1,4,8,9	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	↑	V_{IH}	A8H	V_{OH}
Write	1,5,6,9	V_{IH}	V_{IL}	V_{IH}	V_{IL}	X	X	X	D_{IN}	X

NOTES:
1. X can be V_{IH} or V_{IL} for address or control pins except for RY/BY#, which is either V_{OL} or V_{OH}, or High Z or D_{OUT} for data pins depending on whether or not OE# is active.
2. RY/BY# output is open drain. When the WSM is ready, Erase is suspended or the device is in deep power-down mode. RY/BY# will be at V_{OH} if it is tied to V_{CC} through a resistor. RY/BY# at V_{OH} is independent of OE# while a WSM operation is in progress.
3. RP# at GND ±0.2V ensures the lowest deep power-down current.
4. A_0 and A_1 at V_{IL} provide device manufacturer codes in x8 and x16 modes respectively. A_0 and A_1 at V_{IH} provide device ID codes in x8 and x16 modes respectively. All other addresses are set to zero.
5. Commands for Erase, Data Write, or Lock-Block operations can only be completed successfully when $V_{PP} = V_{PPH1}$ or $V_{PP} = V_{PPH2}$.
6. While the WSM is running, RY/BY# in level-mode (default) stays at V_{OL} until all operations are complete. RY/BY# goes to V_{OH} when the WSM is not busy or in erase suspend mode.
7. RY/BY# may be at V_{OL} while the WSM is busy performing various operations. For example, a Status Register read during a Write operation.
8. The 28F016XS shares an identical device identifier with other Intel Flash memories. Reading this identifier in conjunction with the unique Device Proliferation Code (read from the Page Buffer after writing the Upload Device Configuration command), the 28F016XS can be identified by system software.
9. CE_{0-1}# at V_{IL} is defined as both CE_0# and CE_1# low, and CE_{0-1}# at V_{IH} is defined as either CE_0# or CE_1# high.
10. Addresses are latched on the rising edge of CLK in conjunction with ADV# low. Address $A_1 = 0$ selects the even bank and $A_1 = 1$ selects the odd bank, in both byte-wide mode and word-wide mode device configurations.

4.3 28F008SA—Compatible Mode Command Bus Definitions

Command	Notes	First Bus Cycle			Second Bus Cycle		
		Oper	Addr	Data	Oper	Addr	Data
Read Array		Write	X	FFH	Read	AA	AD
Intelligent Identifier	1	Write	X	90H	Read	IA	ID
Read Compatible Status Register	2	Write	X	70H	Read	X	CSRD
Clear Status Register	3	Write	X	50H			
Word/Byte Write		Write	X	40H	Write	WA	WD
Alternate Word/Byte Write		Write	X	10H	Write	WA	WD
Block Erase/Confirm		Write	X	20H	Write	BA	D0H
Erase Suspend/Resume		Write	X	B0H	Write	X	D0H

ADDRESS
AA = Array Address
BA = Block Address
IA = Identifier Address
WA = Write Address
X = Don't Care

DATA
AD = Array Data
CSRD = CSR Data
ID = Identifier Data
WD = Write Data

NOTES:
1. Following the Intelligent Identifier command, two Read operations access the manufacturer and device signature codes.
2. The CSR is automatically available after device enters Data Write, Erase, or Suspend operations.
3. Clears CSR.3, CSR.4 and CSR.5. Also clears GSR.5 and all BSR.5, BSR.4 and BSR.2 bits. See Status Register definitions.

4.4 28F016XS—Performance Enhancement Command Bus Definitions

Command	Mode	Notes	First Bus Cycle			Second Bus Cycle			Third Bus Cycle		
			Oper	Addr	Data	Oper	Addr	Data	Oper	Addr	Data
Read Extended Status Register		1	Write	X	71H	Read	RA	GSRD BSRD			
Page Buffer Swap		7	Write	X	72H						
Read Page Buffer		11	Write	X	75H	Read	PA	PD			
Single Load to Page Buffer			Write	X	74H	Write	PA	PD			
Sequential Load to Page Buffer	x8	4,6,10	Write	X	E0H	Write	X	BCL	Write	X	BCH
	x16	4,5,6,10	Write	X	E0H	Write	X	WCL	Write	X	WCH
Page Buffer Write to Flash	x8	3,4,9,10	Write	X	0CH	Write	A_0	BC(L,H)	Write	WA	BC(H,L)
	x16	4,5,10	Write	X	0CH	Write	X	WCL	Write	WA	WCH
Two-Byte Write	x8	3	Write	X	FBH	Write	A_0	WD(L,H)	Write	WA	WD(H,L)
Lock Block/Confirm			Write	X	77H	Write	BA	D0H			
Upload Status Bits/Confirm		2	Write	X	97H	Write	X	D0H			
Upload Device Information/Confirm		12	Write	X	99H	Write	X	D0H			
Erase All Unlocked Blocks/Confirm			Write	X	A7H	Write	X	D0H			
Device Configuration		8	Write	X	96H	Write	X	DCCD			
Sleep			Write	X	F0H						
Abort			Write	X	80H						

ADDRESS
BA = Block Address
PA = Page Buffer Address
RA = Extended Register Address
WA = Write Address
X = Don't Care

DATA
AD = Array Data
PD = Page Buffer Data
BSRD = BSR Data
GSRD = GSR Data

WC (L,H) = Word Count (Low, High)
BC (L,H) = Byte Count (Low, High)
WD (L,H) = Write Data (Low, High)
DCCD = Device Configuration Code Data

NOTES:
1. RA can be the GSR address or any BSR address. See Figures 5 and 6 for Extended Status Register memory maps.
2. Upon device power-up, all BSR lock-bits come up locked. The Upload Status Bits command must be written to reflect the actual lock-bit status.
3. A_0 is automatically complemented to load second byte of data. BYTE# must be at V_{IL}. A_0 value determines which WD/BC is supplied first: A_0 = 0 looks at the WDL/BCL, A_0 = 1 looks at the WDH/BCH.
4. BCH/WCH must be at 00H for this product because of the 256-byte (128-word) Page Buffer size, and to avoid writing the Page Buffer contents to more than one 256-byte segment within an array block. They are simply shown for future Page Buffer expandability.
5. In x16 mode, only the lower byte DQ_{0-7} is used for WCL and WCH. The upper byte DQ_{8-15} is a don't care.
6. PA and PD (whose count is given in cycles 2 and 3) are supplied starting in the fourth cycle, which is not shown.
7. This command allows the user to swap between available Page Buffers (0 or 1).
8. This command reconfigures RY/BY# output and SFI Configuration.
9. Write address, WA, is the Destination address in the flash array which must match the Source address in the Page Buffer. Refer to the 16-Mbit Flash Product Family User's Manual.
10. BCL = 00H corresponds to a byte count of 1. Similarly, WCL = 00H corresponds to a word count of 1.
11. Page buffer reads are valid at any frequency up to the corresponding SFI Configuration setting of 2. Page buffer reads above this frequency may produce invalid results and should not be attempted. See Section 4.9 for SFI Configuration frequency settings.
12. After writing the Upload Device Information command and the Confirm command, the following information is output at Page Buffer addresses specified below:

Address	Information
06H, 07H (Byte Mode)	Device Revision Number
03H (Word Mode)	Device Revision Number
1EH (Byte Mode)	Device Configuration Code
0FH (DQ_{0_7})(Word Mode)	Device Configuration Code
1FH (Byte Mode)	Device Proliferation Code (03H)
0FH (DQ_{8_15})(Word Mode)	Device Proliferation Code (03H)

A page buffer swap followed by a page buffer read sequence is necessary to access this information. The contents of all other Page Buffer locations, after the Upload Device Information command is written, are reserved for future implementation by Intel Corporation. See Section 4.8 for a description of the Device Configuration Code. This code also corresponds to data written to the 28F016XS after writing the Device Configuration command.

4.5 Compatible Status Register

WSMS	ESS	ES	DWS	VPPS	R	R	R
7	6	5	4	3	2	1	0

CSR.7 = WRITE STATE MACHINE STATUS
- 1 = Ready
- 0 = Busy

CSR.6 = ERASE-SUSPEND STATUS
- 1 = Erase Suspended
- 0 = Erase In Progress/Completed

CSR.5 = ERASE STATUS
- 1 = Error In Block Erasure
- 0 = Successful Block Erase

CSR.4 = DATA-WRITE STATUS
- 1 = Error in Data Write
- 0 = Data Write Successful

CSR.3 = V_{PP} STATUS
- 1 = V_{PP} Error Detect, Operation Abort
- 0 = V_{PP} OK

NOTES:

RY/BY# output or WSMS bit must be checked to determine completion of an operation (Erase, Erase Suspend, or Data Write) before the appropriate Status bit (ESS, ES or DWS) is checked for success.

If DWS and ES are set to "1" during an erase attempt, an improper command sequence was entered. Clear the CSR and attempt the operation again.

The VPPS bit, unlike an A/D converter, does not provide continuous indication of V_{PP}level. The WSM interrogates V_{PP}'s level only after the Data Write or Erase command sequences have been entered, and informs the system if V_{PP} has not been switched on. VPPS is not guaranteed to report accurate feedback between V_{PPLK}(max) and V_{PPH1}(min), between V_{PPH1}(max) and V_{PPH2}(min), and above V_{PPH2}(max).

CSR.2-0 = RESERVED FOR FUTURE ENHANCEMENTS
These bits are reserved for future use; mask them out when polling the CSR.

4.6 Global Status Register

WSMS	OSS	DOS	DSS	QS	PBAS	PBS	PBSS
7	6	5	4	3	2	1	0

GSR.7 = WRITE STATE MACHINE STATUS
- 1 = Ready
- 0 = Busy

GSR.6 = OPERATION SUSPEND STATUS
- 1 = Operation Suspended
- 0 = Operation in Progress/Completed

GSR.5 = DEVICE OPERATION STATUS
- 1 = Operation Unsuccessful
- 0 = Operation Successful or Currently Running

GSR.4 = DEVICE SLEEP STATUS
- 1 = Device in Sleep
- 0 = Device Not in Sleep

MATRIX 5/4

0 0 = Operation Successful or Currently Running	If operation currently running, then GSR.7 = 0. If device pending sleep, then GSR.7 = 0.
0 1 = Device in Sleep mode or Pending Sleep	
1 0 = Operation Unsuccessful	Operation aborted: Unsuccessful due to Abort command.
1 1 = Operation Unsuccessful or Aborted	

GSR.3 = QUEUE STATUS
- 1 = Queue Full
- 0 = Queue Available

GSR.2 = PAGE BUFFER AVAILABLE STATUS — The device contains two Page Buffers.
- 1 = One or Two Page Buffers Available
- 0 = No Page Buffer Available

GSR.1 = PAGE BUFFER STATUS — Selected Page Buffer is currently busy with WSM operation
- 1 = Selected Page Buffer Ready
- 0 = Selected Page Buffer Busy

GSR.0 = PAGE BUFFER SELECT STATUS
- 1 = Page Buffer 1 Selected
- 0 = Page Buffer 0 Selected

NOTES:
[1] RY/BY# output or WSMS bit must be checked to determine completion of an operation (Block Lock, Suspend, any RY/BY# reconfiguration, Upload Status Bits, Erase or Data Write) before the appropriate Status bit (OSS or DOS) is checked for success.

NOTE:
1. When multiple operations are queued, checking BSR.7 only provides indication of completion for that particular block. GSR.7 provides indication when all queued operations are completed.

4.7 Block Status Register

BS	BLS	BOS	BOAS	QS	VPPS	VPPL	R
7	6	5	4	3	2	1	0

BSR.7 = BLOCK STATUS
- 1 = Ready
- 0 = Busy

BSR.6 = BLOCK LOCK STATUS
- 1 = Block Unlocked for Write/Erase
- 0 = Block Locked for Write/Erase

BSR.5 = BLOCK OPERATION STATUS
- 1 = Operation Unsuccessful
- 0 = Operation Successful or Currently Running

BSR.4 = BLOCK OPERATION ABORT STATUS
- 1 = Operation Aborted
- 0 = Operation Not Aborted

The BOAS bit will not be set until BSR.7 = 1.

MATRIX 5/4
- 0 0 = Operation Successful or Currently Running
- 0 1 = Not a Valid Combination
- 1 0 = Operation Unsuccessful
- 1 1 = Operation Aborted

Operation halted via Abort command.

BSR.3 = QUEUE STATUS
- 1 = Queue Full
- 0 = Queue Available

BSR.2 = V_{PP} STATUS
- 1 = V_{PP} Error Detect, Operation Abort
- 0 = V_{PP} OK

BSR.1 = V_{PP} LEVEL
- 1 = V_{PP} Detected at 5.0V ±10%
- 0 = V_{PP} Detected at 12.0V ±5%

BSR.1 is not guaranteed to report accurate feedback between the V_{PPH1} and V_{PPH2} voltage ranges. Writes and erases with V_{PP} between V_{PPLK}(max) and V_{PPH1} (min), between V_{PPH1}(max) and V_{PPH2}(min), and above V_{PPH2}(max) produce spurious results and should not be attempted.

BSR.0 = RESERVED FOR FUTURE ENHANCEMENTS
This bits is reserved for future use; mask it out when polling the BSRs.

NOTES:
[1] RY/BY# output or BS bit must be checked to determine completion of an operation (Block Lock, Suspend, Erase or Data Write) before the appropriate Status bits (BOS, BLS) is checked for success.

NOTE:
1. When multiple operations are queued, checking BSR.7 only provides indication of completion for that particular block. GSR.7 provides indication when all queued operations are completed.

4.8 Device Configuration Code

R	R	SFI2	SFI1	SFI0	RB2	RB1	RB0
7	6	5	4	3	2	1	0

DCC.5-DCC.3 = SFI CONFIGURATION (SFI2-SFI0)
- 001 = SFI Configuration 1
- 010 = SFI Configuration 2
- 011 = SFI Configuration 3
- 100 = SFI Configuration 4 (Default)

DCC.2-DCC.0 = RY/BY# CONFIGURATION (RB2-RB0)
- 001 = Level Mode (Default)
- 010 = Pulse-On-Write
- 011 = Pulse-On-Erase
- 100 = RY/BY# Disabled
- 101 = Pulse-On-Write/Erase

DCC.7-DCC.6 = RESERVED FOR FUTURE ENHANCEMENTS
These bits are reserved for future use; mask them out when reading the Device Configuration Code. Set these bits to "0" when writing the desired RY/BY# configuration to the device.

NOTES:
Default SFI Configuration on powerup or return from deep powerdown mode is 4, allowing system boot from the 28F016XS at any frequency up to the device's maximum frequency. Undocumented combinations of SFI2-SFI0 are reserved by Intel Corporation for future implementations and should not be used.

Undocumented combinations of RB2-RB0 are reserved by Intel Corporation for future implementations and should not be used.

4.9 SFI Configuration Table

SFI Configuration	Notes	28F016XS-15 Frequency (MHz)	28F016XS-20 Frequency (MHz)	28F016XS-25 Frequency (MHz)
4	1,2	66 (and below)	50 (and below)	40 (and below)
3	2	50 (and below)	37.5 (and below)	30 (and below)
2	2	33 (and below)	25 (and below)	20 (and below)
1	2	16.7 (and below)	12.5 (and below)	10 (and below)

NOTE:
1. Default SFI Configuration after powerup or return from deep power-down mode via RP# low.
2. SFI Configuration is retained if put in sleep mode via a Sleep or Abort Command.

5.0 ELECTRICAL SPECIFICATIONS

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

5.1 Absolute Maximum Ratings*

Temperature Under Bias .....0°C to +80°C
Storage Temperature .....65°C to +125°C

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

V_{CC} = 3.3V ±0.3V Systems(5)

Symbol	Parameter	Notes	Min	Max	Units	Test Conditions
T_A	Operating Temperature, Commercial	1	0	70	°C	Ambient Temperature
V_{CC}	V_{CC} with Respect to GND	2	−0.2	7.0	V	
V_{PP}	V_{PP} Supply Voltage with Respect to GND	2,3	−0.2	14.0	V	
V	Voltage on any Pin (except V_{CC}, V_{PP}) with Respect to GND	2	−0.5	V_{CC} + 0.5	V	
I	Current into any Non-Supply Pin			30	mA	
I_{OUT}	Output Short Circuit Current	4		100	mA	

V_{CC} = 5.0V ±0.5V Systems(5)

Symbol	Parameter	Notes	Min	Max	Units	Test Conditions
T_A	Operating Temperature, Commercial	1	0	70	°C	Ambient Temperature
V_{CC}	V_{CC} with Respect to GND	2	−0.2	7.0	V	
V_{PP}	V_{PP} Supply Voltage with Respect to GND	2,3	−0.2	14.0	V	
V	Voltage on any Pin (except V_{CC}, V_{PP}) with Respect to GND	2	−2.0	7.0	V	
I	Current into any Non-Supply Pin			30	mA	
I_{OUT}	Output Short Circuit Current	4		100	mA	

NOTES:
1. Operating temperature is for commercial product defined by this specification.
2. Minimum DC voltage is −0.5V on input/output pins. During transitions, this level may undershoot to −2.0V for periods <20 ns. Maximum DC voltage on input/output pins is V_{CC}+ 0.5V which may overshoot to V_{CC} + 2.0V for periods <20 ns.
3. Maximum DC voltage on V_{PP} may overshoot to +14.0V for periods <20 ns.
4. Output shorted for no more than one second. No more than one output shorted at a time.
5. AC specifications are valid at both voltage ranges. See DC Characteristics tables for voltage range-specific specifications.

5.2 Capacitance

For a 3.3V ±0.3V System:

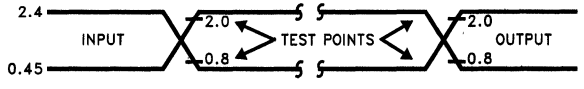
Symbol	Parameter	Notes	Typ	Max	Units	Test Conditions
C_{IN}	Capacitance Looking into an Address/Control Pin	1	6	8	pF	T_A = 25°C, f = 1.0 MHz
C_{OUT}	Capacitance Looking into an Output Pin	1	8	12	pF	T_A = 25°C, f = 1.0 MHz
C_{LOAD}	Load Capacitance Driven by Outputs for Timing Specifications	1, 2		50	pF	For the 28F016XS-20 and 28F016XS-25

For 5.0V ±0.5V System:

Symbol	Parameter	Notes	Typ	Max	Units	Test Conditions
C_{IN}	Capacitance Looking into an Address/Control Pin	1	6	8	pF	T_A= 25°C, f = 1.0 MHz
C_{OUT}	Capacitance Looking into an Output Pin	1	8	12	pF	T_A= 25°C, f = 1.0 MHz
C_{LOAD}	Load Capacitance Driven by Outputs for Timing Specifications	1, 2		100	pF	For the 28F016XS-20
				30	pF	For the 28F016XS-15

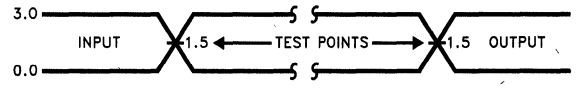
NOTE:
1. Sampled, not 100% tested. Guaranteed by design.
2. Intel is currently developing more accurate models for the Transient Equivalent Testing Load Circuits. For more information or to obtain iBIS models, please contact your local Intel/Distribution Sales Office.

5.3 Transient Input/Output Reference Waveforms



AC test inputs are driven at V_{OH} (2.4 VTTL) for a Logic "1" and V_{OL} (0.45 VTTL) for a Logic "0." Input timing begins at V_{IH} (2.0 VTTL) and V_{IL} (0.8 VTTL). Output timing ends at V_{IH} and V_{IL}. Input rise and fall times (10% to 90%) <10 ns.

Figure 7. Transient Input/Output Reference Waveform (V_{CC} = 5.0V ±0.5V) for Standard Testing Configuration(1)



AC test inputs are driven at 3.0V for a Logic "1" and 0.0V for a Logic "0." Input timing begins, and output timing ends, at 1.5V. Input rise and fall times (10% to 90%) <10 ns.

Figure 8. Transient Input/Output Reference Waveform (V_{CC} = 3.3V ±0.3V) High Speed Reference Waveform(2) (V_{CC} = 5.0V ±0.5V)

NOTES:
1. Testing characteristics for 28F016XS-20 at 5V V_{CC}.
2. Testing characteristics for 28F016XS-15 at 5V V_{CC} and 28F016XS-20/28F016XS-25 at 3.3V V_{CC}.

5.4 DC Characteristics

V_{CC} = 3.3V ±0.3V, T_A = 0°C to +70°C

Symbol	Parameter	Notes	Min	Typ	Max	Units	Test Conditions
I_{LI}	Input Load Current	1			±1	µA	V_{CC} = V_{CC} Max, V_{IN} = V_{CC} or GND
I_{LO}	Output Leakage Current	1			±10	µA	V_{CC} = V_{CC} Max, V_{OUT} = V_{CC} or GND
I_{CCS}	V_{CC} Standby Current	1,5		70	130	µA	V_{CC} = V_{CC}Max, CE_0#, CE_1#, RP# = V_{CC} ±0.2V BYTE#, WP# = V_{CC} ±0.2V or GND ±0.2V
				1	4	mA	V_{CC} = V_{CC} Max, CE_0#, CE_1#, RP# = V_{IH} BYTE#, WP# = V_{IH} or V_{IL}
I_{CCD}	V_{CC} Deep Power-Down Current	1		2	5	µA	RP# = GND ± 0.2V BYTE# = V_{CC} ±0.2V or GND ±0.2V
I_{CCR1}	V_{CC} Word/Byte Read Current	1,4,5		65	85	mA	V_{CC} = V_{CC} Max CMOS: CE_0#, CE_1# = GND ±0.2V BYTE# = GND ±0.2V or V_{CC} ±0.2V Inputs = GND ±0.2V or V_{CC} ±0.2V 4-Location Access Sequence: 3-1-1-1 (clocks) f = 25 MHz, I_{OUT} = 0 mA
I_{CCR2}	V_{CC} Word/Byte Read Current	1,4, 5,6		60	75	mA	V_{CC} = V_{CC} Max CMOS: CE_0#, CE_1# = GND ±0.2V BYTE# = GND ±0.2V or V_{CC} ±0.2V Inputs = GND ±0.2V or V_{CC} ±0.2V 4-Location Access Sequence: 3-1-1-1 (clocks) f = 16 MHz, I_{OUT} = 0 mA
I_{CCW}	V_{CC} Write Current	1,6		8	12	mA	Word/Byte Write in Progress V_{PP} = 12.0V ±5%
				8	17	mA	Word/Byte Write in Progress V_{PP} = 5.0V ±10%
I_{CCE}	V_{CC} Block Erase Current	1,6		6	12	mA	Block Erase in Progress V_{PP} = 12.0V ±5%
				9	17	mA	Block Erase in Progress V_{PP} = 5.0V ±10%
I_{CCES}	V_{CC} Erase Suspend Current	1,2		3	6	mA	CE_0#, CE_1# = V_{IH} Block Erase Suspended

2

5.4 DC Characteristics (Continued)

V_{CC} = 3.3V ±0.3V, T_A = 0°C to +70°C

Symbol	Parameter	Notes	Min	Typ	Max	Units	Test Conditions
I_{PPS} I_{PPR}	V_{PP} Standby/Read Current	1		±1	±10	µA	$V_{PP} \leq V_{CC}$
				30	50	µA	$V_{PP} > V_{CC}$
I_{PPD}	V_{PP} Deep Power-Down Current	1		0.2	5	µA	RP# = GND ±0.2V
I_{PPW}	V_{PP} Write Current	1,6		10	15	mA	V_{PP} = 12.0V ±5% Word/Byte Write in Progress
				15	25	mA	V_{PP} = 5.0V ±10% Word/Byte Write in Progress
I_{PPE}	V_{PP} Erase Current	1,6		4	10	mA	V_{PP} = 12.0V ±5% Block Erase in Progress
				14	20	mA	V_{PP} = 5.0V ±10% Block Erase in Progress
I_{PPES}	V_{PP} Erase Suspend Current	1		30	50	µA	V_{PP} = V_{PPH1} or V_{PPH2}, Block Erase Suspended
V_{IL}	Input Low Voltage	6	−0.3		0.8	V	
V_{IH}	Input High Voltage	6	2.0		V_{CC} +0.3	V	
V_{OL}	Output Low Voltage	6			0.4	V	V_{CC} = V_{CC} Min and I_{OL} = 4 mA
V_{OH1}	Output High Voltage	6	2.4			V	I_{OH} = −2.0 mA V_{CC} = V_{CC} Min
V_{OH2}			V_{CC} −0.2				I_{OH} = −100 µA V_{CC} = V_{CC} Min
V_{PPLK}	V_{PP} Erase/Write Lock Voltage	3,6	0.0		1.5	V	
V_{PPH1}	V_{PP} during Write/Erase Operations	3	4.5	5.0	5.5	V	
V_{PPH2}	V_{PP} during Write/Erase Operations	3	11.4	12.0	12.6	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage		2.0			V	

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 3.3V, V_{PP} = 12.0V or 5.0V, T = 25°C. These currents are valid for all product versions (package and speeds).
2. I_{CCES} is specified with the device de-selected. If the device is read while in erase suspend mode, current draw is the sum of I_{CCES} and I_{CCR}.
3. Block Erases, Word/Byte Writes and Lock Block operations are inhibited when V_{PP} \ V_{PPLK} and not guaranteed in the ranges between V_{PPLK}(max) and V_{PPH1}(min), between V_{PPH1}(max) and V_{PPH2}(min) and above V_{PPH2}(max).
4. Automatic Power Savings (APS) reduces I_{CCR} to 3 mA typical in static operation.
5. CMOS Inputs are either V_{CC} ±0.2V or GND ±0.2V. TTL Inputs are either V_{IL} or V_{IH}.
6. Sampled, but not 100% tested. Guaranteed by design.

5.5 DC Characteristics

V_{CC} = 5.0V ±0.5V, T_A = 0°C to +70°C

Symbol	Parameter	Notes	Min	Typ	Max	Units	Test Conditions
I_{LI}	Input Load Current	1			±1	µA	V_{CC} = V_{CC} Max V_{IN} = V_{CC} or GND
I_{LO}	Output Leakage Current	1			±10	µA	V_{CC} = V_{CC} Max V_{OUT} = V_{CC} or GND
I_{CCS}	V_{CC} Standby Current	1,5		70	130	µA	V_{CC} = V_{CC} Max CE_0#, CE_1#, RP# = V_{CC} ± 0.2V BYTE#, WP# = V_{CC} ±0.2V or GND ±0.2V
				2	4	mA	V_{CC} = V_{CC} Max CE_0#, CE_1#, RP# = V_{IH} BYTE#, WP# = V_{IH} or V_{IL}
I_{CCD}	V_{CC} Deep Power-Down Current	1		2	5	µA	RP# = GND ±0.2V BYTE# = V_{CC} ±0.2V or GND ±0.2V
I_{CCR1}	V_{CC} Read Current	1,4,5		120	175	mA	V_{CC} = V_{CC} Max, CMOS: CE_0#, CE_1# = GND ±0.2V BYTE# = GND ±0.2V or V_{CC} ±0.2V Inputs = GND ±0.2V or V_{CC} ±0.2V 4-Location Access Sequence: 3-1-1-1 (clocks) f = 33 MHz, I_{OUT} = 0 mA
I_{CCR2}	V_{CC} Read Current	1,4, 5,6		105	150	mA	V_{CC} = V_{CC} Max, CMOS: CE_0#, CE_1# = GND ±0.2V BYTE# = GND ±0.2V or V_{CC} ±0.2V Inputs = GND ±0.2V or V_{CC} ±0.2V 4-Location Access Sequence: 3-1-1-1 (clocks) f = 20 MHz, I_{OUT} = 0 mA
I_{CCW}	V_{CC} Write Current	1,6		25	35	mA	Word/Byte in Progress V_{PP} = 12.0V ±5%
				25	40	mA	Word/Byte in Progress V_{PP} = 5.0V ±10%
I_{CCE}	V_{CC} Erase Suspend Current	1,6		18	25	mA	Block Erase in Progress V_{PP} = 12.0V ±5%
				20	30	mA	Block Erase in Progress V_{PP} = 5.0V ±10%
I_{CCES}	V_{CC} Block Erase Current	1,2		5	10	mA	CE_0#, CE_1# = V_{IH} Block Erase Suspended

2

5.5 DC Characteristics (Continued)

V_{CC} = 5.0V ±0.5V, T_A = 0°C to +70°C

Symbol	Parameter	Notes	Min	Typ	Max	Units	Test Conditions
I_{PPS} I_{PPR}	V_{PP} Standby/Read Current	1		±1	± 10	µA	$V_{PP} \leq V_{CC}$
				30	50	µA	$V_{PP} > V_{CC}$
I_{PPD}	V_{PP} Deep Power-Down Current	1		0.2	5	µA	RP# = GND ±0.2V
I_{PPW}	V_{PP} Write Current	1,6		7	12	mA	V_{PP} = 12.0V ±5% Word/Byte Write in Progress
				17	22	mA	V_{PP} = 5.0V ±10% Word/Byte Write in Progress
I_{PPE}	V_{PP} Block Erase Current	1,6		5	10	mA	V_{PP} = 12.0V ±5% Block Erase in Progress
				16	20	mA	V_{PP} = 5.0V ±10% Block Erase in Progress
I_{PPES}	V_{PP} Erase Suspend Current	1		30	50	µA	V_{PP} = V_{PPH1} or V_{PPH2}, Block Erase Suspended
V_{IL}	Input Low Voltage	6	−0.5		0.8	V	
V_{IH}	Input High Voltage	6	2.0		V_{CC} +0.5	V	
V_{OL}	Output Low Voltage	6			0.45	V	V_{CC} = V_{CC} Min I_{OL} = 5.8 mA
V_{OH1}	Output High Voltage	6	0.85 V_{CC}			V	I_{OH} = −2.5 mA V_{CC} = V_{CC} Min
V_{OH2}			V_{CC} −0.4				I_{OH} = −100 µA V_{CC} = V_{CC} Min
V_{PPLK}	V_{PP} Write/Erase Lock Voltage	3,6	0.0		1.5	V	
V_{PPH1}	V_{PP} during Write/Erase Operations		4.5	5.0	5.5	V	
V_{PPH2}	V_{PP} during Write/Erase Operations		11.4	12.0	12.6	V	
V_{LKO}	V_{CC} Write/Erase Lock Voltage		2.0			V	

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V or 5.0V, T = 25°C. These currents are valid for all product versions (package and speeds).
2. I_{CCES} is specified with the device de-selected. If the device is read while in erase suspend mode, current draw is the sum of I_{CCES} and I_{CCR}.
3. Block Erases, Word/Byte Writes and Lock Block operations are inhibited when $V_{PP} \leq V_{PPLK}$ and not guaranteed in the ranges between V_{PPLK}(max) and V_{PPH1}(min), between V_{PPH1}(max) and V_{PPH2}(min) and above V_{PPH2}(max).
4. Automatic Power Saving (APS) reduces I_{CCR} to 1 mA typical in Static operation.
5. CMOS Inputs are either V_{CC} ±0.2V or GND ±0.2V. TTL Inputs are either V_{IL} or V_{IH}.
6. Sampled, but not 100% tested. Guaranteed by design.

5.6 Timing Nomenclature

All 3.3V system timings are measured from where signals cross 1.5V.

For 5.0V systems, use the standard JEDEC cross point definitions (standard testing) or from where signals cross 1.5V (high speed testing).

Each timing parameter consists of 5 characters. Some common examples are defined below:

t_{ELCH} time(t) from CE# (E) going low (L) to CLK (C) going high (H)

t_{AVCH} time(t) from address (A) valid (V) to CLK (C) going high (H)

t_{WHDX} time(t) from WE# (W) going high (H) to when the data (D) can become undefined (X)

	Pin Characters		Pin States
A	Address Inputs	H	High
C	CLK (Clock)	L	Low
D	Data Inputs	V	Valid
Q	Data Outputs	X	Driven, but Not Necessarily Valid
E	CE# (Chip Enable)	Z	High Impedance
F	BYTE# (Byte Enable)	L	Latched
G	OE# (Output Enable)		
W	WE# (Write Enable)		
P	RP# (Deep Power-Down Pin)		
R	RY/BY# (Ready Busy)		
V	ADV# (Address Valid)		
5V	V_{CC} at 4.5V Minimum		
3V	V_{CC} at 3.0V Minimum		

5.7 AC Characteristics—Read Only Operations[1]

V_{CC} = 3.3V ±0.3V, T_A = 0°C to +70°C

Versions[3]			28F016XS-20		28F016XS-25		Units
Symbol	**Parameter**	**Notes**	**Min**	**Max**	**Min**	**Max**	
f_{CLK}	CLK Frequency	7		50		40	MHz
t_{CLK}	CLK Period		20		25		ns
t_{CH}	CLK High Time		6		8.5		ns
t_{CL}	CLK Low Time		6		8.5		ns
t_{CLCH}	CLK Rise Time			4		4	ns
t_{CHCL}	CLK Fall Time			4		4	ns
t_{ELCH}	CE_X# Setup to CLK	6	25		35		ns
t_{VLCH}	ADV# Setup to CLK		20		25		ns
t_{AVCH}	Address Valid to CLK		20		25		ns
t_{CHAX}	Address Hold from CLK		0		0		ns
t_{CHVH}	ADV# Hold from CLK		0		0		ns
t_{GLCH}	OE# Setup to CLK		20		25		ns
t_{CHQV}	CLK to Data Delay			30		35	ns
t_{PHCH}	RP# High to CLK		480		480		ns
t_{CHQX}	Output Hold from CLK	2	6		6		ns
t_{ELQX}	CE_X# to Output Low Z	2,6	0		0		ns
t_{EHQZ}	CE_X# High to Output High Z	2,6		30		30	ns
t_{GLQX}	OE# to Output Low Z	2	0		0		ns
t_{GHQZ}	OE# High to Output High Z	2		30		30	ns
t_{OH}	Output Hold from CE_X# or OE# Change, Whichever Occurs First	6	0		0		ns

5.7 AC Characteristics—Read Only Operations[1] (Continued)

V_{CC} = 5.0V ±0.5V, T_A = 0°C to +70°C

Versions[3]			28F016XS-15[4]		28F016XS-20[5]		Units
Symbol	Parameter	Notes	Min	Max	Min	Max	
f_{CLK}	CLK Frequency	7		66		50	MHz
t_{CLK}	CLK Period		15		20		ns
t_{CH}	CLK High Time		3.5		6		ns
t_{CL}	CLK Low Time		3.5		6		ns
t_{CLCH}	CLK Rise Time			4		4	ns
t_{CHCL}	CLK Fall Time			4		4	ns
t_{ELCH}	CE_X# Setup to CLK	6	25		30		ns
t_{VLCH}	ADV# Setup to CLK		15		20		ns
t_{AVCH}	Address Valid to CLK		15		20		ns
t_{CHAX}	Address Hold from CLK		0		0		ns
t_{CHVH}	ADV# Hold from CLK		0		0		ns
t_{GLCH}	OE# Setup to CLK		15		20		ns
t_{CHQV}	CLK to Data Delay			20		30	ns
t_{PHCH}	RP# High to CLK		300		300		ns
t_{CHQX}	Output Hold from CLK	2	5		5		ns
t_{ELQX}	CE_X# to Output Low Z	2,6	0		0		ns
t_{EHQZ}	CE_X# High to Output High Z	2,6		30		30	ns
t_{GLQX}	OE# to Output Low Z	2	0		0		ns
t_{GHQZ}	OE# High to Output High Z	2		30		30	ns
t_{OH}	Output Hold from CE_X# or OE# Change, Whichever Occurs First	6	0		0		ns

NOTES:
1. See AC Input/Output Reference Waveforms for timing measurements.
2. Sampled, not 100% tested. Guaranteed by design.
3. Device speeds are defined as:
   15 ns at V_{CC} = 5.0V equivalent to
   20 ns at V_{CC} = 3.3V
   20 ns at V_{CC} = 5.0V equivalent to
   25 ns at V_{CC} = 3.3V
4. See the high speed AC Input/Output Reference Waveforms.
5. See the standard AC Input/Output Reference Waveforms.
6. CE_X# is defined as the latter of CE_0# or CE_1# going low, or the first of CE_0# or CE_1# going high.
7. Page buffer reads are valid at any frequency up to the corresponding SFI Configuration setting of 2. Page buffer reads above this frequency may produce invalid results and should not be attempted. See Section 4.9 for SFI Configuration frequency settings.

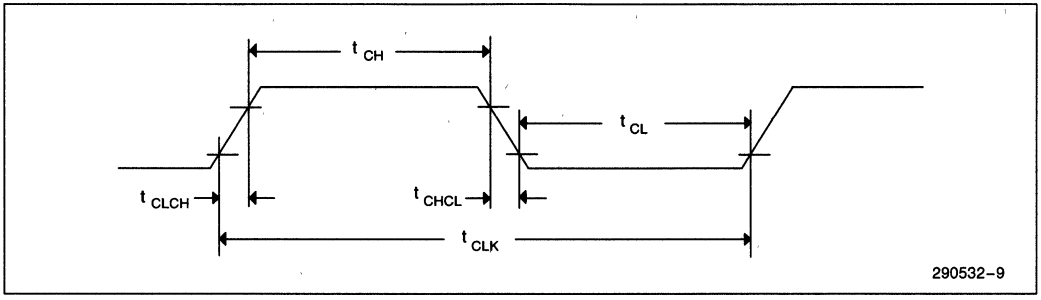
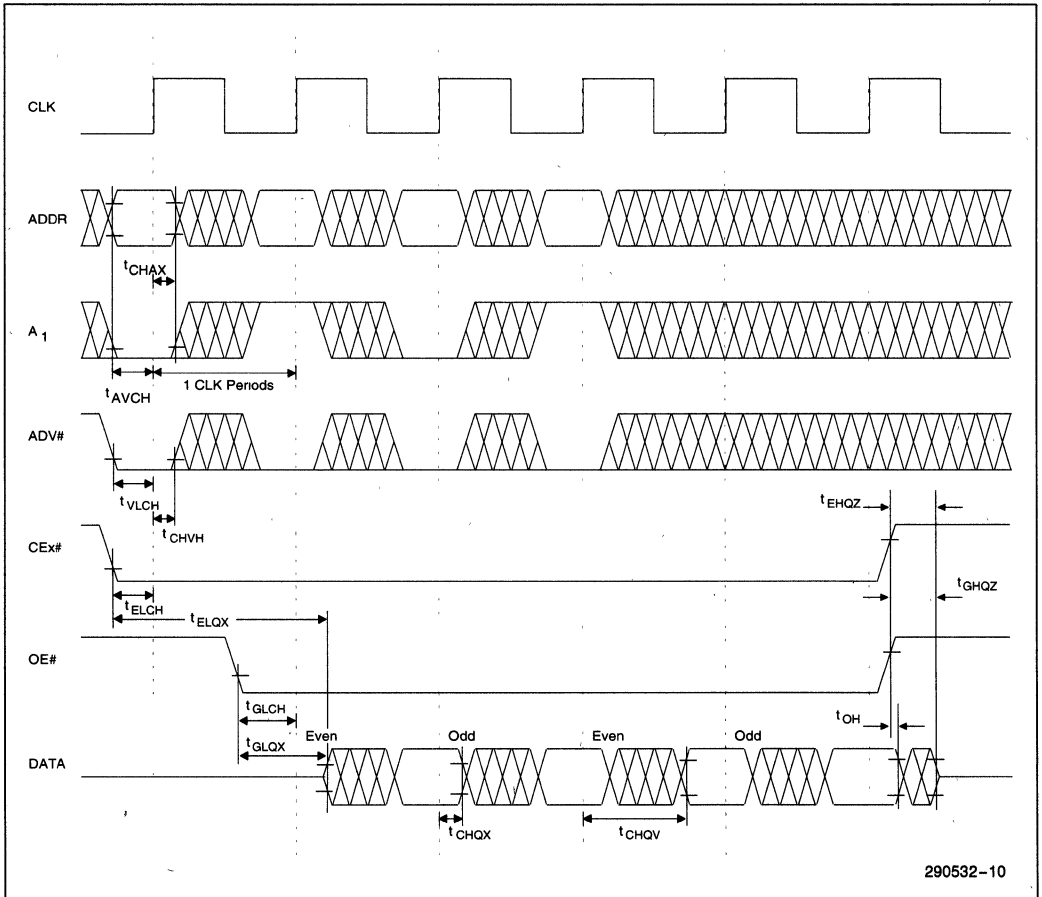
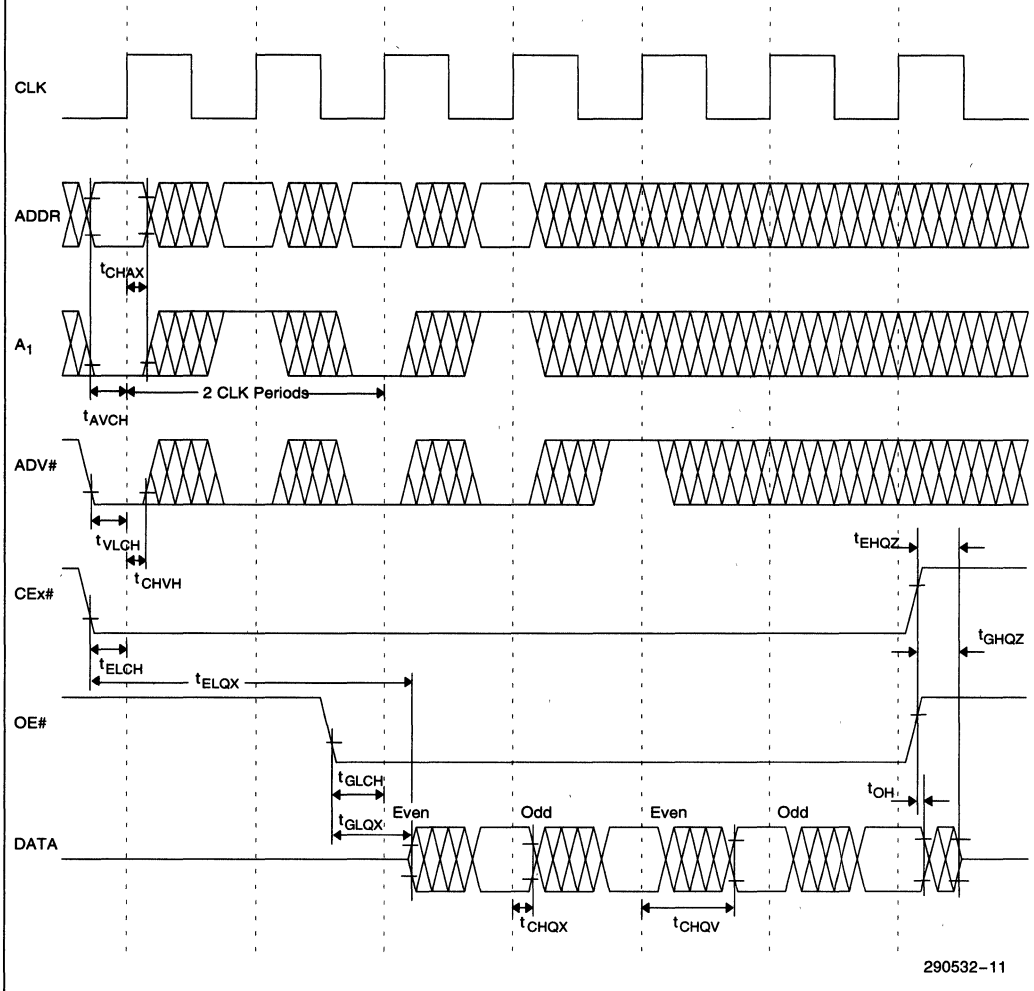


Figure 9. CLK Waveform



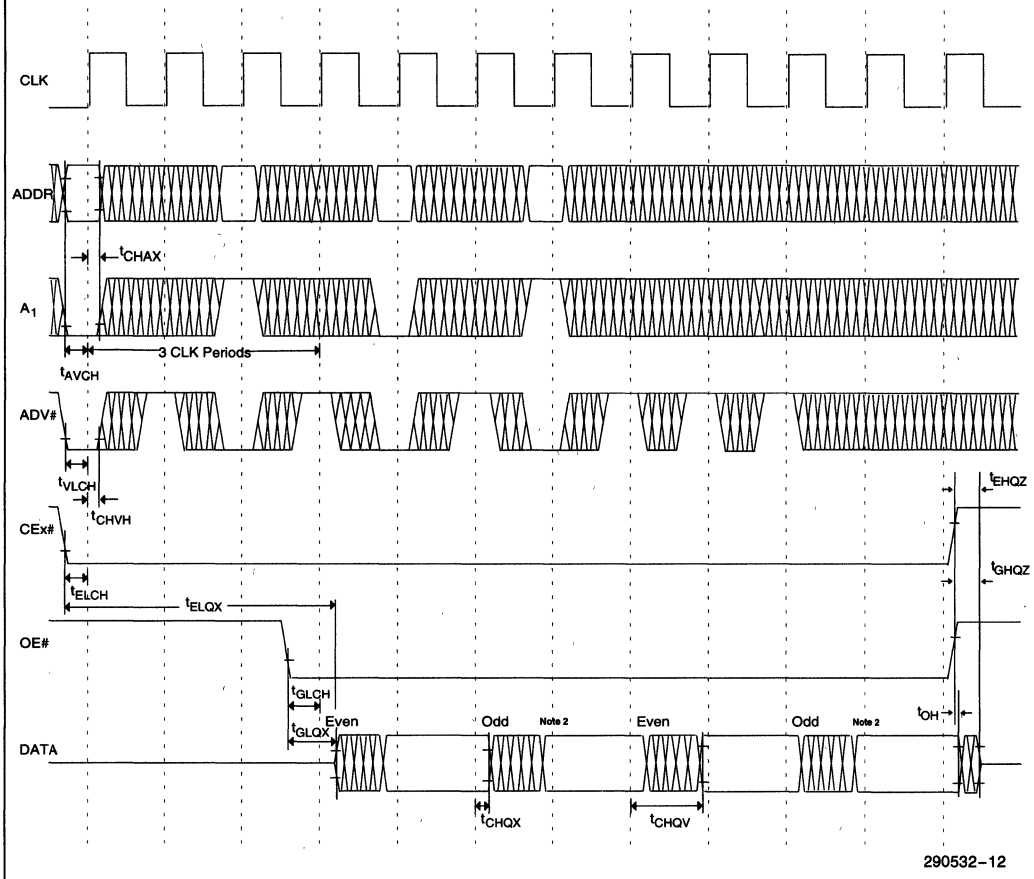
NOTE:
1. The 28F016XS can sustain an endless burst access assuming alternating bank accesses; the length of the burst access is dictated by the control CPU or bus architecture.

Figure 10. Read Timing Waveform(1) (SFI Configuration = 1, Alternate-Bank Accesses)



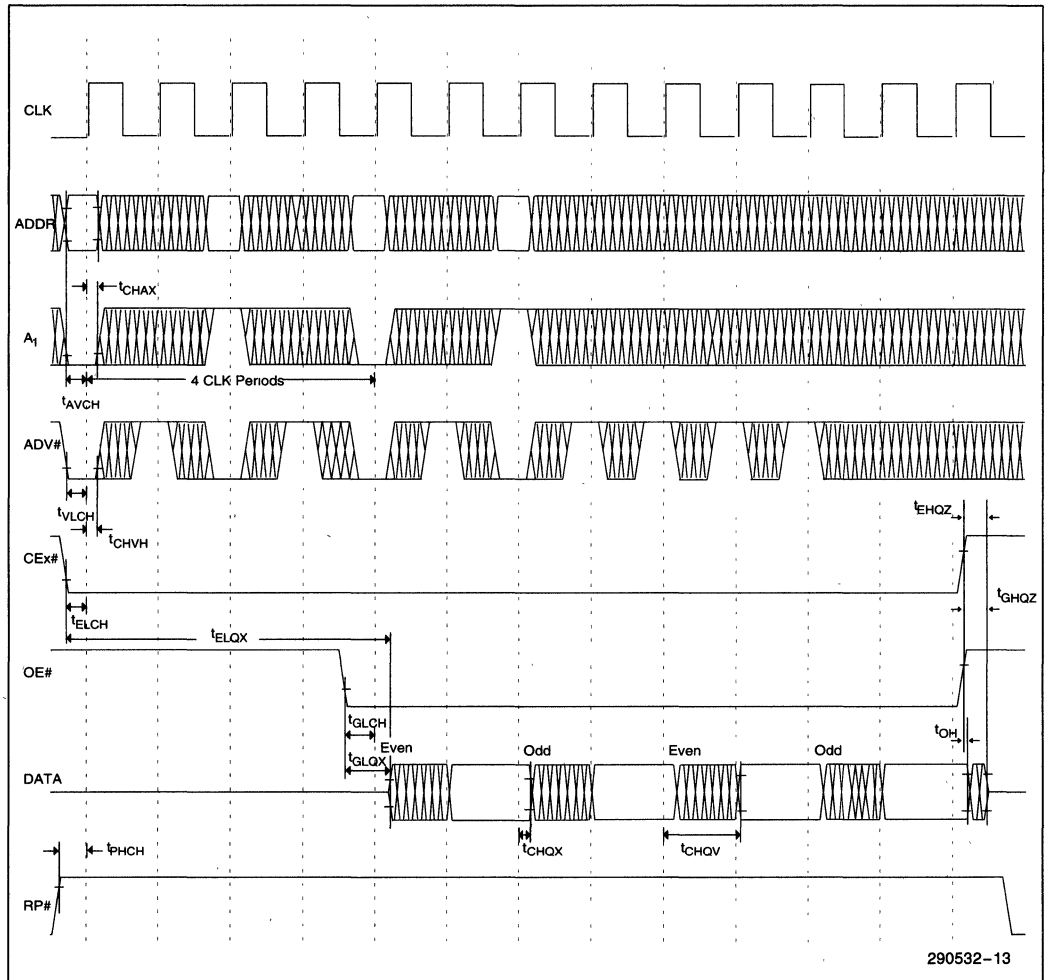
NOTE:
1. The 28F016XS can sustain an endless burst access assuming alternating bank accesses; the length of the burst access is dictated by the control CPU or bus architecture.

Figure 11. Read Timing Waveform(1) (SFI Configuration = 2, Alternate-Bank Accesses)



NOTES:
1. The 28F016XS can sustain an endless burst access assuming alternating bank accesses; the length of the burst access is dictated by the control CPU or bus architecture.
2. Depending on the actual operation frequency, a consecutive alternating bank access can be initiated one clock period earlier. See AP-398 for further information.

Figure 12. Read Timing Waveform(1) (SFI Configuration = 3, Alternate-Bank Accesses)



NOTE:
1. The 28F016XS can sustain an endless burst access assuming alternating bank accesses; the length of the burst access is dictated by the control CPU or bus architecture.

Figure 13. Read Timing Waveform(1) (SFI Configuration = 4, Alternating Bank Accesses)

5.8 AC Characteristics for WE#-Controlled Write Operations[1]

V_{CC} = 3.3V ±0.3V, T_A = 0°C to +70°C

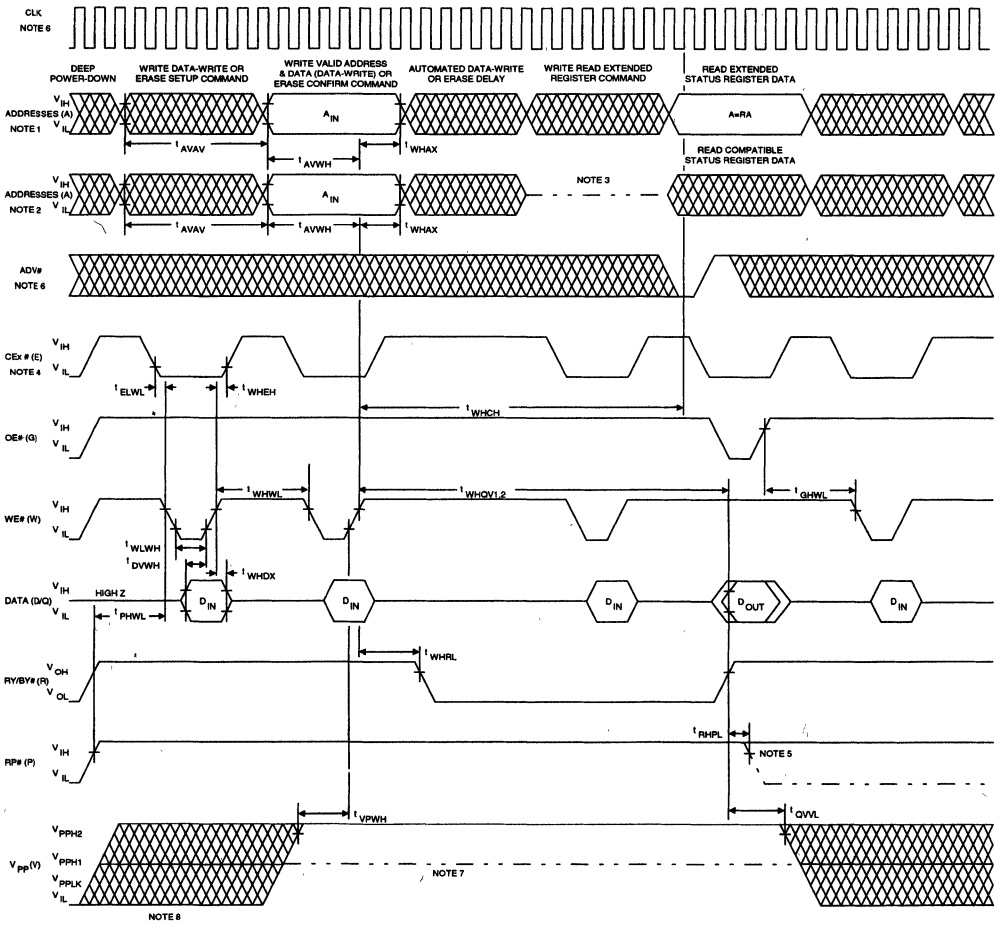
Versions			28F016XS-20			28F016XS-25			Unit
Symbol	Parameter	Notes	Min	Typ	Max	Min	Typ	Max	
t_{AVAV}	Write Cycle Time		75			75			ns
t_{VPWH}	V_{PP} Setup to WE# Going High	3	100			100			ns
t_{PHEL}	RP# Setup to CE_X# Going Low	3,7	480			480			ns
t_{ELWL}	CE_X# Setup to WE# Going Low	3,7	0			0			ns
t_{AVWH}	Address Setup to WE# Going High	2,6	60			60			ns
t_{DVWH}	Data Setup to WE# Going High	2,6	60			60			ns
t_{WLWH}	WE# Pulse Width		60			60			ns
t_{WHDX}	Data Hold from WE# High	2	5			5			ns
t_{WHAX}	Address Hold from WE# High	2	5			5			ns
t_{WHEH}	CE_X# hold from WE# High	3,7	5			5			ns
t_{WHWL}	WE# Pulse Width High		15			15			ns
t_{GHWL}	Read Recovery before Write	3	0			0			ns
t_{WHRL}	WE# High to RY/BY# Going Low	3			100			100	ns
t_{RHPL}	RP# Hold from Valid Status Register (CSR, GSR, BSR) data and RY/BY# High	3	0			0			ns
t_{PHWL}	RP# High Recovery to WE# Going Low	3	480			480			ns
t_{WHCH}	Write Recovery before Read	3	20			20			ns
t_{QVVL}	V_{PP} Hold from Valid Status Register (CSR, GSR, BSR) Data and RY/BY# High	3	0			0			µs
t_{WHQV1}	Duration of Word/Byte Write Operation	3,4,5	5	9	TBD	5	9	TBD	µs
t_{WHQV2}	Duration of Block Erase Operation	3,4	0.6	1.6	20	0.6	1.6	20	sec

5.8 AC Characteristics for WE#—Controlled Write Operations[1] (Continued)

V_{CC} = 5.0V ±0.5V, T_A = 0°C to +70°C

Versions			28F016XS-15			28F016XS-20			Unit
Symbol	Parameter	Notes	Min	Typ	Max	Min	Typ	Max	
t_{AVAV}	Write Cycle Time		65			65			ns
t_{VPWH}	V_{PP} Setup to WE# Going High	3	100			100			ns
t_{PHEL}	RP# Setup to CE_X# Going Low	3,7	300			300			ns
t_{ELWL}	CE_X# Setup to WE# Going Low	3,7	0			0			ns
t_{AVWH}	Address Setup to WE# Going High	2,6	50			50			ns
t_{DVWH}	Data Setup to WE# Going High	2,6	50			50			ns
t_{WLWH}	WE# Pulse Width		50			50			ns
t_{WHDX}	Data Hold from WE# High	2	0			0			ns
t_{WHAX}	Address Hold from WE# High	2	5			5			ns
t_{WHEH}	CE_X# hold from WE# High	3,7	5			5			ns
t_{WHWL}	WE# Pulse Width High		15			15			ns
t_{GHWL}	Read Recovery before Write	3	0			0			ns
t_{WHRL}	WE# High to RY/BY# Going Low	3			100			100	ns
t_{RHPL}	RP# Hold from Valid Status Register (CSR, GSR, BSR) data and RY/BY# High	3	0			0			ns
t_{PHWL}	RP# High Recovery to WE# Going Low	3	300			300			ns
t_{WHCH}	Write Recovery before Read	3	20			20			ns
t_{QVVL}	V_{PP} Hold from Valid Status Register (CSR, GSR, BSR) Data and RY/BY# High	3	0			0			µs
$t_{WHQV}1$	Duration of Word/Byte Write Operation	3,4,5	4.5	6	TBD	4.5	6	TBD	µs
$t_{WHQV}2$	Duration of Block Erase Operation	3,4	0.6	1.2	20	0.6	1.2	20	sec

NOTES:
1. Read timings during Write and Erase are the same as for normal read.
2. Refer to command definition tables for valid address and data values.
3. Sampled, but not 100% tested. Guaranteed by design.
4. Write/Erase durations are measured to valid Status Register (CSR) Data.
5. Word/Byte Write operations are typically performed with 1 Programming Pulse.
6. Address and Data are latched on the rising edge of WE# for all Command Write operations.
7. CE_X# is defined as the latter of CE_0# or CE_1# going low, or the first of CE_0# or CE_1# going high.



290532-14

NOTES:
1. This address string depicts Data Write/Erase cycles with corresponding verification via ESRD.
2. This address string depicts Data Write/Erase cycles with corresponding verification via CSRD.
3. This cycle is invalid when using CSRD for verification during Data Write/Erase operations.
4. CE_X# is defined as the latter of CE_0# or CE_1# going low or the first of CE_0# or CE_1# going high.
5. RP# low transition is only to show t_{RHPL}; not valid for above Read and Write cycles.
6. Data Write/Erase cycles are asynchronous; CLK and ADV# are ignored.
7. V_{PP} voltage during Data Write/Erase operations valid at both 12.0V and 5.0V.
8. V_{PP} voltage equal to or below V_{PPLK} provides complete flash memory array protection.

Figure 14. AC Waveforms for WE# Command Write Operations, Illustrating a Two Command Write Sequence Followed by a Extended Status Register Read

5.9 AC Characteristics for CE_X#—Controlled Write Operations(1)

V_{CC} = 3.3V ±0.3V, T_A = 0°C to +70°C

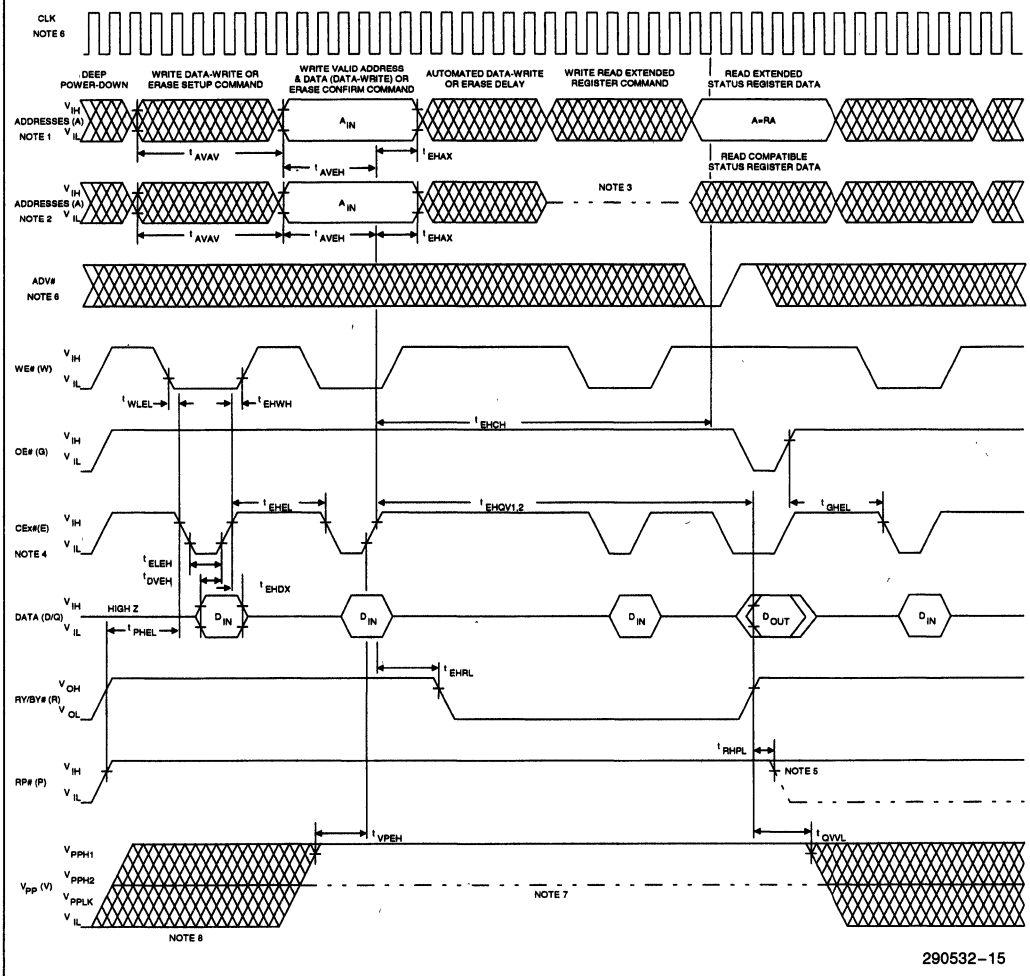
Versions			28F016XS-20			28F016XS-25			Unit
Symbol	Parameter	Notes	Min	Typ	Max	Min	Typ	Max	
t_{AVAV}	Write Cycle Time		75			75			ns
t_{VPEH}	V_{PP} Setup to CE_X# Going High	3,7	100			100			ns
t_{PHWL}	RP# Setup to WE# Going Low	3	480			480			ns
t_{WLEL}	WE# Setup to CE_X# Going Low	3,7	0			0			ns
t_{AVEH}	Address Setup to CE_X# Going High	2,6,7	60			60			ns
t_{DVEH}	Data Setup to CE_X# Going High	2,6,7	60			60			ns
t_{ELEH}	CE_X# Pulse Width	7	60			60			ns
t_{EHDX}	Data Hold from CE_X# High	2,7	10			10			ns
t_{EHAX}	Address Hold from CE_X# High	2,7	10			10			ns
t_{EHWH}	WE hold from CE_X# High	3,7	5			5			ns
t_{EHEL}	CE_X# Pulse Width High	7	15			15			ns
t_{GHEL}	Read Recovery before Write	3	0			0			ns
t_{EHRL}	CE_X# High to RY/BY# Going Low	3,7			100			100	ns
t_{RHPL}	RP# Hold from Valid Status Register (CSR, GSR, BSR) Data and RY/BY# High	3	0			0			ns
t_{PHEL}	RP# High Recovery to CE_X# Going Low	3,7	480			480			ns
t_{EHCH}	Write Recovery before Read	3	20			20			ns
t_{QVVL}	V_{PP} Hold from Valid Status Register (CSR, GSR, BSR) Data and RY/BY# High	3	0			0			µs
t_{EHQV1}	Duration of Word/Byte Write Operation	3,4,5	5	9	TBD	5	9	TBD	µs
t_{EHQV2}	Duration of Block Erase Operation	3,4	0.6	1.6	20	0.6	1.6	20	sec

5.9 AC Characteristics for CE_X# Controlled Write Operations[1] (Continued)

V_{CC} = 5.0V ±0.5V, T_A = 0°C to +70°C

Versions			28F016XS-15			28F016XS-20			Unit
Symbol	Parameter	Notes	Min	Typ	Max	Min	Typ	Max	
t_{AVAV}	Write Cycle Time		60			60			ns
t_{PHWL}	RP# Setup to WE# Going Low	3	300			300			ns
t_{VPEH}	V_{PP} Setup to CE_X# Going Low Going High	3,7	100			100			ns
t_{WLEL}	WE# Setup to CE_X# Going Low	3,7	0			0			ns
t_{AVEH}	Address Setup to CE_X# Going High	2,6,7	45			45			ns
t_{DVEH}	Data Setup to CE_X# Going High	2,6,7	45			45			ns
t_{ELEH}	CE_X# Pulse Width	7	45			45			ns
t_{EHDX}	Data Hold from Going HighCE_X# High	2,7	0			0			ns
t_{EHAX}	Address Hold from CE_X# High	2,7	5			5			ns
t_{EHWH}	WE hold from CE_X# High	3,7	5			5			ns
t_{EHEL}	CE_X# Pulse Width High	7	15			15			ns
t_{GHEL}	Read Recovery before Write	3	0			0			ns
t_{EHRL}	CE_X# High to RY/BY# Going Low	3,7			100			100	ns
t_{RHPL}	RP# Hold from Valid Status Register (CSR, GSR, BSR) Data and RY/BY# High	3	0			0			ns
t_{PHEL}	RP# High Recovery to CE_X# Going Low	3,7	300			300			ns
t_{EHCH}	Write Recovery before Read	3	20			20			ns
t_{QVVL}	V_{PP} Hold from Valid Status Register (CSR, GSR, BSR) Data and RY/BY# High	3	0			0			µs
$t_{EHQV}1$	Duration of Word/Byte Write Operation	3,4,5	4.5	6	TBD	4.5	6	TBD	µs
$t_{EHQV}2$	Duration of Block Erase Operation	3,4	0.6	1.2	20	0.6	1.2	20	sec

NOTES:
1. Read timings during Write and Erase are the same as for normal read.
2. Refer to command definition tables for valid address and data values.
3. Sampled, but not 100% tested. Guaranteed by design.
4. Write/Erase durations are measured to valid Status Register (CSR) Data.
5. Word/Byte Write operations are typically performed with 1 Programming Pulse.
6. Address and Data are latched on the rising edge of WE# for all Command Write operations.
7. CE_X# is defined as the latter of CE_0# or CE_1# going low, or the first of CE_0# or CE_1# going high.



NOTES:
1. This address string depicts Data Write/Erase cycles with corresponding verification via ESRD.
2. This address string depicts Data Write/Erase cycles with corresponding verification via CSRD.
3. This cycle is invalid when using CSRD for verification during Data Write/Erase operations.
4. CE_X# is defined as the latter of CE_0# or CE_1# going low or the first of CE_0# or CE_1# going high.
5. RP# low transition is only to show t_{RHPL}; not valid for above Read and Write cycles.
6. Data Write/Erase cycles are asynchronous; CLK and ADV# are ignored.
7. V_{PP} voltage during Data Write/Erase operations valid at both 12.0V and 5.0V.
8. V_{PP} voltage equal to or below V_{PPLK} provides complete flash memory array protection.

Figure 15. AC Waveforms for CE_X#—Controlled Write Operations, Illustrating a Two Command Write Sequence Followed by a Extended Status Register Read

5.10 AC Characteristics for WE#—Controlled Page Buffer Write Operations(1)

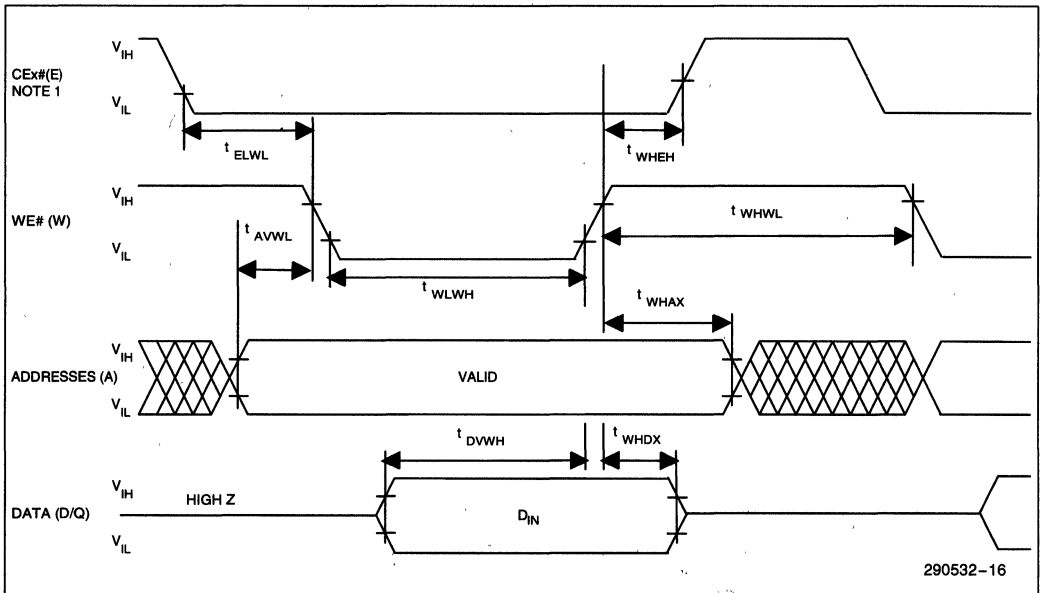
V_{CC} = 3.3V ±0.3V, T_A = 0°C to +70°C

Versions			28F016XS-20			28F016XS-25			Unit
Symbol	Parameter	Notes	Min	Typ	Max	Min	Typ	Max	
t_{AVWL}	Address Setup to WE# Going Low	2	0			0			ns

V_{CC} = 5.0V ±0.5V, T_A = 0°C to +70°C

Versions			28F016XS-15			28F016XS-20			Unit
Symbol	Parameter	Notes	Min	Typ	Max	Min	Typ	Max	
t_{AVWL}	Address Setup to WE# Going Low	2	0			0			ns

NOTES:
1. All other specifications for WE# Controlled Page Buffer Write Operations see Section 5.8.
2. Address must be valid during the entire WE# low pulse.



NOTE:
1. CE_X# is defined as the latter of CE_0# or CE_1# going low, or the first of CE_0# or CE_1# going high.

Figure 16. WE#—Controlled Page Buffer Write Timing Waveforms (Loading Data to the Page Buffer)

5.11 AC Characteristics for CE_X#—Controlled Page Buffer Write Operations

V_{CC} = 3.3V ±0.3V, T_A = 0°C to +70°C

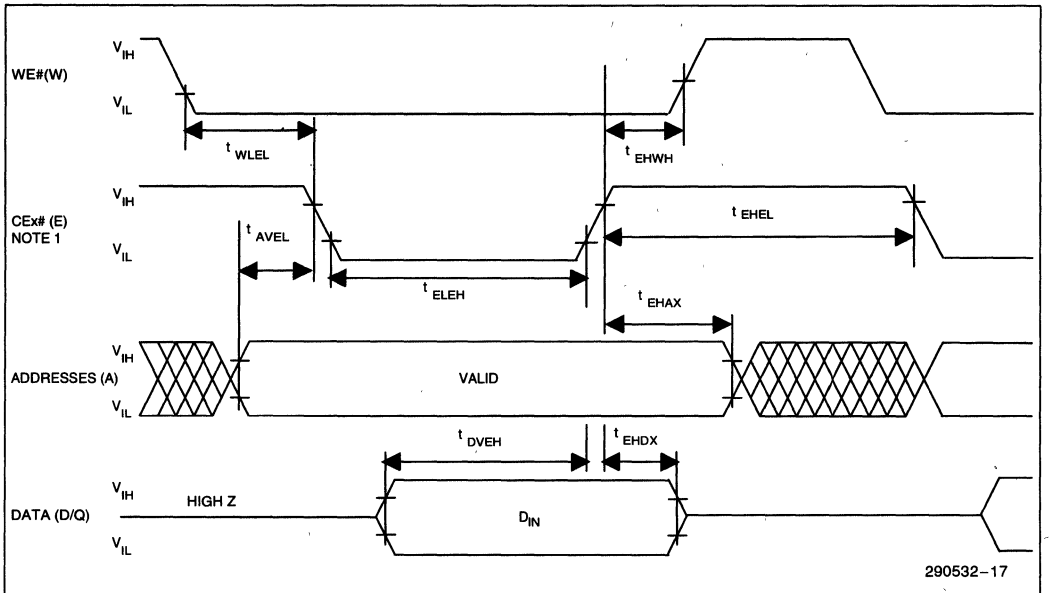
Versions			28F016XS-20			28F016XS-25			Unit
Symbol	Parameter	Notes	Min	Typ	Max	Min	Typ	Max	
t_{AVEL}	Address Setup to CE_X# Going Low	2,3	0			0			ns

V_{CC} = 5.0V ±0.5V, T_A = 0°C to +70°C

Versions			28F016XS-15			28F016XS-20			Unit
Symbol	Parameter	Notes	Min	Typ	Max	Min	Typ	Max	
t_{AVEL}	Address Setup to CE_X# Going Low	2,3	0			0			ns

2

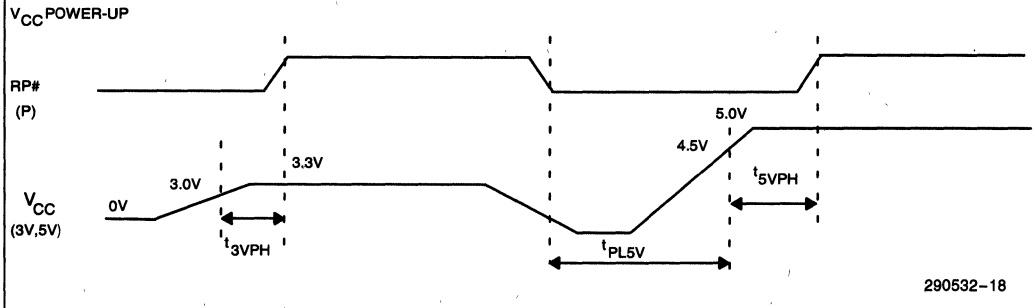
NOTES:
1. All other specifications for CE_X# Controlled Page Buffer Write Operations see Section 5.9.
2. Address must be valid during the entire CE# low pulse.
3. CE_X# is defined as the latter of CE_0# or CE_1# going low, or the first of CE_0# or CE_1# going high.



NOTE:
1. CE_X# is defined as the latter of CE_0# or CE_1# going low, or the first of CE_0# or CE_1# going high.

Figure 17. CE_X#—Controlled Page Buffer Write Timing Waveforms (Loading Data to the Page Buffer)

5.12 Power-Up and Reset Timings



NOTE:
For read timings following reset see Section 5.7.

Figure 18. V_{CC} Power-Up and RP# Reset Waveforms

Symbol	Parameter	Notes	Min	Max	Unit
t_{PL5V}	RP# Low to V_{CC} at 4.5V (Minimum)	2	0		μs
t_{PL3V}	RP# Low to V_{CC}at 3.0V (Minimum)	2	0		μs
t_{5VPH}	V_{CC} at 4.5V Minimum) to RP# High	1	2		μs
t_{3VPH}	V_{CC} at 3.0V (Minimum) to RP# High	1	2		μs

NOTES:
1. The t_{5VPH} and/or t_{3VPH} times must be strictly followed to guarantee all other read and write specifications for the 28F016XS.
2. The power supply may start to switch concurrently with RP# going low.

5.13 Erase and Word/Byte Write Performance(3,5)

V_{CC} = 3.3V ±0.3V, V_{PP} = 5.0V ±0.5V, T_A = 0°C to +70°C

Symbol	Parameter	Notes	Min	Typ(1)	Max	Units	Test Conditions
	Page Buffer Byte Write Time	2	TBD	6.0	TBD	µs	
	Page Buffer Word Write Time	2	TBD	12.1	TBD	µs	
t_{WHRH1A}	Byte Write Time	2	TBD	16.5	TBD	µs	
t_{WHRH1B}	Word Write Time	2	TBD	24.0	TBD	µs	
t_{WHRH2}	Block Write Time	2	TBD	2.2	TBD	sec	Byte Write Mode
t_{WHRH3}	Block Write Time	2	TBD	1.6	TBD	sec	Word Write Mode
	Block Erase Time	2	TBD	2.8	TBD	sec	
	Full Chip Erase Time	2	TBD	44.8	TBD	sec	
	Time From Erase Suspend Command to WSM Ready	4	TBD	10	TBD	µs	

2

V_{CC} = 3.3V ±0.3V, V_{PP} = 12.0V ±0.6V, T_A = 0°C to +70°C

Symbol	Parameter	Notes	Min	Typ(1)	Max	Units	Test Conditions
	Page Buffer Byte Write Time	2	TBD	2.2	TBD	µs	
	Page Buffer Word Write Time	2	TBD	4.4	TBD	µs	
t_{WHRH1}	Word/Byte Write Time	2	5	9	TBD	µs	
t_{WHRH2}	Block Write Time	2	TBD	1.2	4.2	sec	Byte Write Mode
t_{WHRH3}	Block Write Time	2	TBD	0.6	2.0	sec	Word Write Mode
	Block Erase Time	2	0.6	1.6	20	sec	
	Full Chip Erase Time	2	TBD	25.6	TBD	sec	
	Time From Erase Suspend Command to WSM Ready	4	TBD	10	TBD	µs	

5.13 Erase and Word/Byte Write Performance[3,5] (Continued)

V_{CC} = 5.0V ±0.5V, V_{PP} = 5.0V ±0.5V, T_A = 0°C to +70°C

Symbol	Parameter	Notes	Min	Typ[1]	Max	Units	Test Conditions
	Page Buffer Byte Write Time	2	TBD	6.0	TBD	µs	
	Page Buffer Word	2	TBD	12.1	TBD	µs	Write Time
t_{WHRH1A}	Byte Write Time	2	TBD	11	TBD	µs	
t_{WHRH1B}	Word Write Time	2	TBD	16	TBD	µs	
t_{WHRH2}	Block Write Time	2	TBD	1.6	TBD	sec	Byte Write Mode
t_{WHRH3}	Block Write Time	2	TBD	1.2	TBD	sec	Word Write Mode
	Block Erase Time	2	TBD	2.0	TBD	sec	
	Full Chip Erase Time	2	TBD	32.0	TBD	sec	
	Time From Erase Suspend Command to WSM Ready	4	TBD	10	TBD	µs	

V_{CC} = 5.0V ±0.5V, V_{PP} = 12.0V ±0.6V, T_A = 0°C to +70°C

Symbol	Parameter	Notes	Min	Typ[1]	Max	Units	Test Conditions
	Page Buffer Byte Write Time	2	TBD	2.1	TBD	µs	
	Page Buffer Word Write Time	2	TBD	4.1	TBD	µs	
t_{WHRH1}	Word/Byte Write Time	2	4.5	6	TBD	µs	
t_{WHRH2}	Block Write Time	2	TBD	0.8	4.2	sec	Byte Write Mode
t_{WHRH3}	Block Write Time	2	TBD	0.4	2.0	sec	Word Write Mode
	Block Erase Time	2	0.6	1.2	20	sec	
	Full Chip Erase Time	2	TBD	19.2	TBD	sec	
	Time From Erase Suspend Command to WSM Ready	4	TBD	10	TBD	µs	

NOTES:
1. 25°C, and nominal voltages.
2. Excludes system-level overhead.
3. These performance numbers are valid for all speed versions.
4. Specification applies to interrupt latency for Single Block Erase. Suspend latency for Erase All Unlocked Block operation typically extends erase suspend latency time to 140 µs.
5. Sampled, but not 100% tested. Guaranteed by design.

6.0 MECHANICAL SPECIFICATIONS

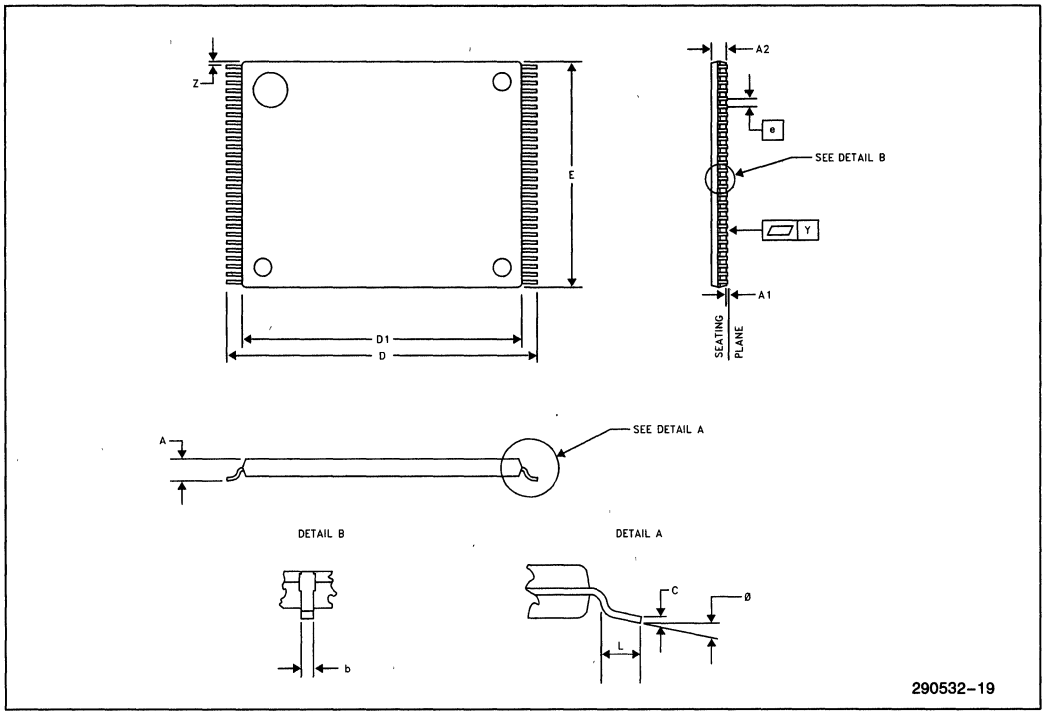
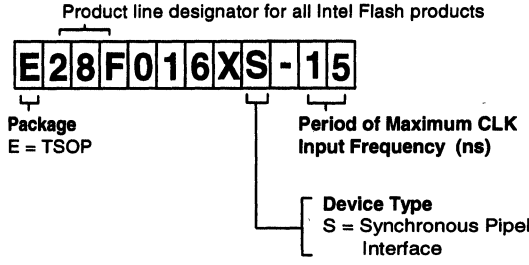


Figure 19. Mechanical Specifications of the 28F016XS 56-Lead TSOP Type I Package

Family: Thin Small Out-Line Package				
Symbol	Millimeters			Notes
	Minimum	Nominal	Maximum	
A			1.20	
A_1	0.50			
A_2	0.965	0.995	1.025	
b	0.100	0.150	0.200	
c	0.115	0.125	0.135	
D_1	18.20	18.40	18.60	
E	13.80	14.00	14.20	
e		0.50		
D	19.80	20.00	20.20	
L	0.500	0.600	0.700	
N		56		
∅	0°	3°	5°	
Y			0.100	
Z	0.150	0.250	0.350	

DEVICE NOMENCLATURE AND ORDERING INFORMATION



Option	Order Code	Valid Combinations		
		V_{CC} = 3.3V ±0.3V, 50 pF load, 1.5V I/O Levels(1)	V_{CC} = 5.0V ± 10%, 100 pF load TTL I/O Levels(1)	V_{CC} = 5.0V ±10%, 30 pF load 1.5V I/O Levels(1)
1	E28F016XS15	28F016XS-20		28F016XS-15
2	E28F016XS20	28F016XS-25	28F016XS-20	

ADDITIONAL INFORMATION

Order Number	Document/Tool
297372	16-Mbit Flash Product Family User's Manual, 28F016SA/28F016SV/28F016XS/28F016XD
292147	AP-398, "Designing with the 28F016XS"
297500	"Interfacing the 28F016XS to the i960® Microprocessor Family"
297504	"Interfacing the 28F016XS to the Intel486™ Microprocessor Family"
292146	AP-600, "Performance Benefits and Power/Energy Savings of 28F016XS Based System Designs"
297508	FlashBuilder Utility
Contact Intel/Distribution Sales Office	28F016XS Benchmark Utility FlashBuilder
Contact Intel/Distribution Sales Office	28F016XS iBIS/VHDL Models
Contact Intel/Distribution Sales Office	28F016XS Orcad/Viewlogic Schematic Symbols
292126	AP 377, "16-Mbit Flash Product Family Software Drivers 28F016SA/28F016SV/28F016XS/28F016XD"
294016	ER 33, "ETOX™ Flash Memory TechnologyInsight to Intel's Fourth Generation Process Innovation"

DATASHEET REVISION HISTORY

Number	Description
001	Original Version

intel®

TECHNICAL PAPER

Interfacing the 28F016XS to the i960® Microprocessor Family

KEN MC KEE
TECHNICAL MARKETING ENGINEER

TIM KELLY
ENGINEER

RANNA PRAJAPATI
ENGINEER

November 1994

ADVANCE INFORMATION

Order Number: 297500-001

INTERFACING THE 28F016XS TO THE i960® MICROPROCESSOR FAMILY

CONTENTS

CONTENTS

1.0 INTRODUCTION

This technical paper describes several designs interfacing the high-performance 28F016XS Flash memory to the i960® microprocessor family. All designs are based on preliminary 28F016XS specifications. These designs have been fully simulated but no yet taken to lab prototype. Please contact your Intel or distribution sales office for up-to-date information. Do not finalize a design based on the specifications in this document.

The 28F016XS is a 16-Mbit flash memory with a high-performance synchronous pipelined read interface. The 28F016XS combines ROM-like non-volatility, DRAM-like read performance and in-system update ability in one memory technology. These characteristics enable code execution directly from the 28F016XS memory space, replacing the costly practice of shadowing code from HDD or ROM to DRAM for increased performance. The 28F016XS improves system performance, ruggedness and cost of any burst microprocessor, such as the i960 microprocessor, base design. The i960 microprocessor family sees widespread use in various applications, including imaging and data communications.

The 28F016XS performs synchronous pipelined reads. Up to three accesses can be initiated before reading data output from the initial cycle. This pipelined structure is ideal for use with the i960 microprocessor's burst transfer mechanism. The 28F016XS brings significant system performance enhancements to an i960 microprocessor-based environment. This technical paper describes processor-to-memory interfaces that exploit these capabilities to achieve maximum system performance. Figures 1 and 2 illustrate relative system performance enhancements that the 28F016XS brings to an i960 microprocessor-based environment, compared to other memory technologies. The benchmark parameters are documented in Appendix B.

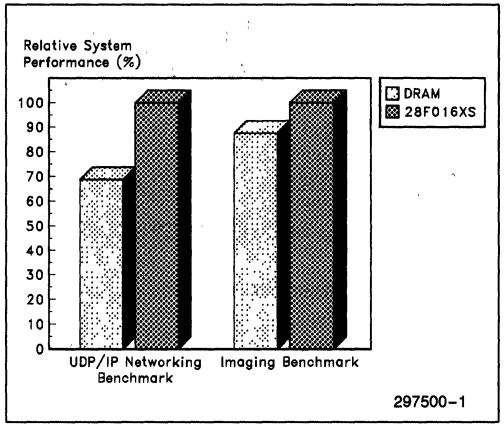


Figure 1. Relative System Performance Enhancement of the 28F016XS Compared to Other Memory Technologies in an i960® KB-25 Microprocessor-Based Design

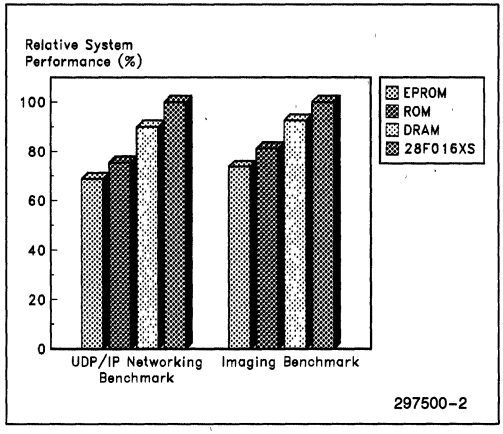


Figure 2. Relative System Performance Enhancement of the 28F016XS Compared to Other Memory Technologies in an i960® CA-33 Processor-Based Design

2.0 I960® CA-33 MICROPROCESSOR INTERFACE

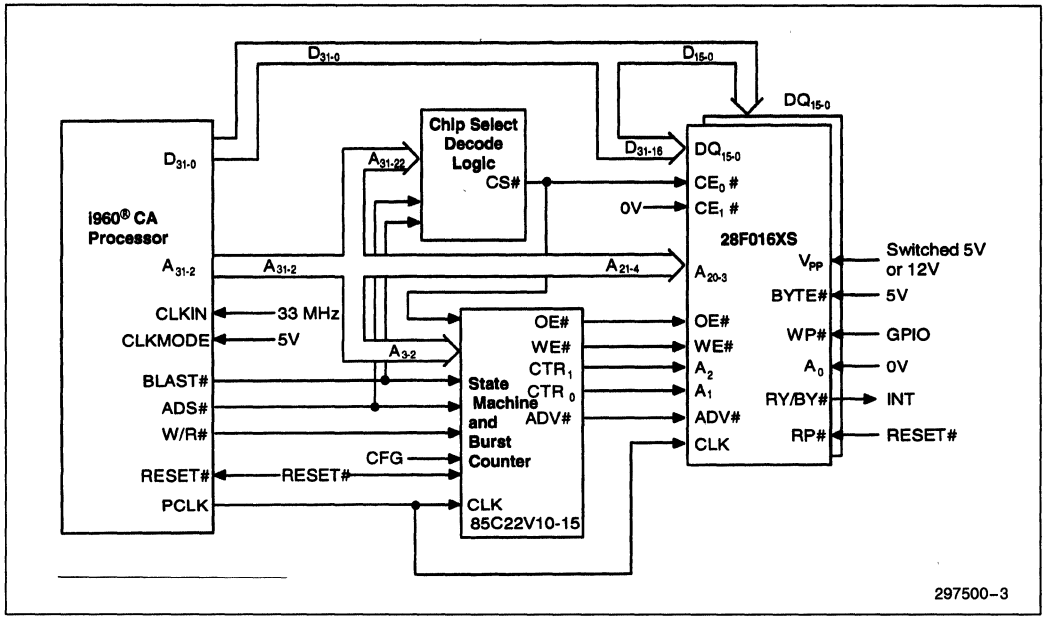


Figure 3. Minimal Logic Required Interfacing the 28F016XS-15 to the i960 CA-33 Microprocessor

Using this interface, the 28F016XS-15/i960 CA-33 microprocessor system achieves 3-0-0-0-2-0-0-0 . . . wait-state read performance, supporting burst transfers and address pipelining.

2.1 Circuit Description

This section describes the 28F016XS-15/i960 CA-33 microprocessor interface block diagram in Figure 3.

Memory Configuration

This design uses two 28F016XS-15s, in x16 mode, to match the i960 CA microprocessor's 32-bit data bus, providing 4 Mbytes of flash memory. Signals A_{21-4} from the i960 CA microprocessor and CTR_{1-0} from the PLD select locations within the 28F016XS memory space, arranged as 1-Meg double words. The two-bit counter implemented in the PLD loads addresses A_{3-2} at the beginning of each memory cycle and generates the lower two bits of the burst addresses on its outputs CTR_{1-0}. The counter feeds burst addresses to the 28F016XS, so that they do not stall waiting for the processor to supply the next address.

Chip Select Logic

Chip select decode logic may use A_{31-22} to generate an active low chip select signal, CS#, for the 28F016XS-15 memory space and other system peripherals. The chip select drives CE_0# on each 28F016XS-15 and a control input to the PLD. The 28F016XS-15's CE_1# input is grounded.

In support of address pipelining, the chip select logic latches CS#, holding it active throughout the duration of the memory access. This will prevent potential CS# problems caused by using combinatorial logic when utilizing the address pipelining capability of i960 CA microprocessor.

If address pipelining functionality is not implemented, simple combinatorial logic can be utilized in generating

the system CS# for the 28F016XS memory space, and the chip select logic shown in Figure 3 does not examine BLAST# and ADS#. For many systems using the upper address bits in a linear selection scheme may provide a sufficient number of chip select signals, thus eliminating system chip select decode logic. (See Figure 4 for an example of using linear selection for chip selects.) When using a linear chip select scheme however, the software must avoid using addresses that may select more than one device, which could result in bus contention.

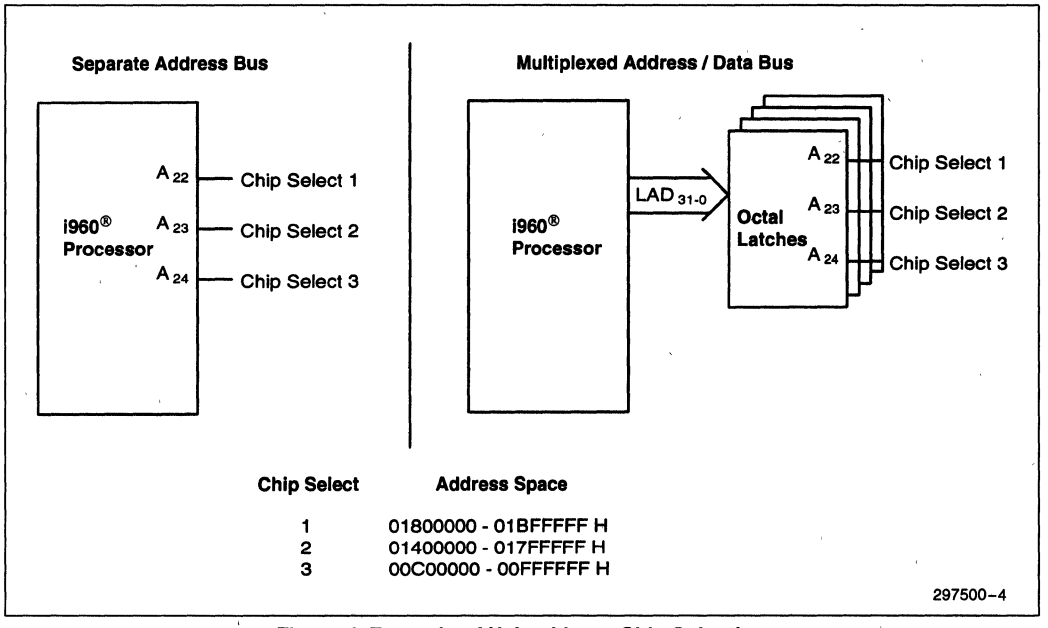


Figure 4. Example of Using Linear Chip Selection

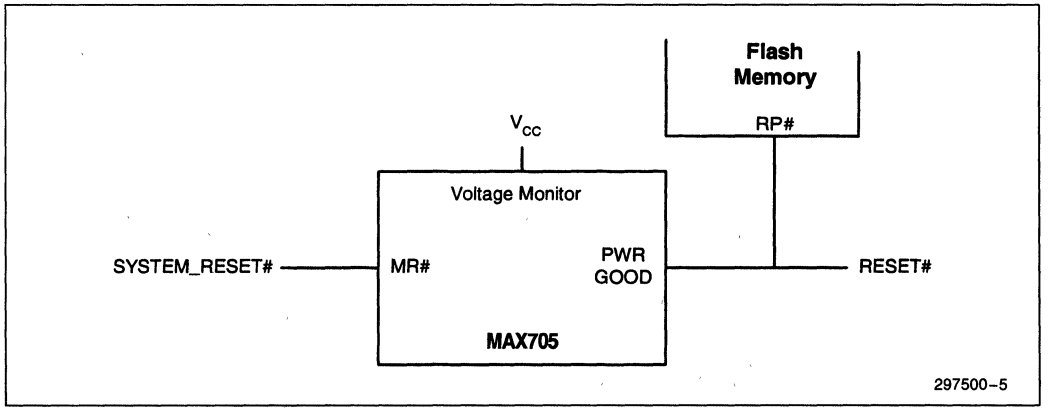


Figure 5. Example RESET Generation Circuitry

CLK Option

A 33 MHz clock signal drives the i960 CA microprocessor CLKIN input. Driving CLKMODE to "1" configures the i960 CA microprocessor for a x1 CLK input. The i960 CA microprocessor outputs an internally-referenced 33 MHz clock on its PCLK1 and PCLK2 pins (the signals on PCLK1 and PCLK2 are identical), which drives the CLK inputs of the PLD and the 28F016XS-15s.

Reset

An active-low reset signal, RESET#, connects to the RESET# inputs of the i960 CA microprocessor, and the PLD, and to the RP# input of the 28F016XS-15s. Figure 5 illustrates a suggested logic configuration for generating RESET#.

Interface Control Signals

The i960 CA-33 microprocessor external bus signals, BLAST#, ADS# and W/R#, serve as inputs to the state machine, which controls the two-bit counter and generates OE#, WE# and ADV#. The counter is loaded at the beginning of the memory access, generating the burst addresses to the 28F016XS-15s. ADV# indicates that a valid address is available to the 28F016XS-15. Addresses are latched and a read cycle is initiated on a rising CLK edge. WE# controls writes to the 28F016XS-15, latching data into the 28F016XS-15 on its rising edge if the applicable timing requirements are satisfied. (Data is latched on the falling edge of WE# during page buffer writes.)

Configuration Signal

A general purpose input/output (GPIO) generates the configuration signal input to the state machine. The configuration signal must reset to logic "0" on power-up and system reset to ensure that the operation of the state machine matches the initial configurations of the 28F016XS-15s and the i960 CA microprocessor. After optimizing the 28F016XS-15s and i960 CA microprocessor, the configuration signal must switch to logic "1."

Additional 28F016XS Control Signals

The BYTE# input to the 28F016XS-15s is tied to 5.0V to configure the 28F016XS-15s for x16 mode, and A_0 is tied to GND (A_0 is only used for byte addressing). A GPIO controls the write protect input, WP#, to the 28F016XS-15s. As shown in Figure 3, the 28F016XS-15 is compatible with either a 5.0V or a 12.0V V_{PP} voltage and is completely write protected by switching V_{PP} to GND. When V_{PP} voltage drops below V_{PPLK}, the 28F016XS-15 will not successfully complete Program and Erase operations. Figure 3 also illustrates the 28F016XS-15 RY/BY# output connected to a system interrupt for background erase operation. RY/BY#, WP#, and V_{PP} implementation are application dependent. See the Additional Information section of this technical paper for documentation that cover these topics in more detail.

2.2 Software Interface Considerations

Boot-up Capability / Configuration

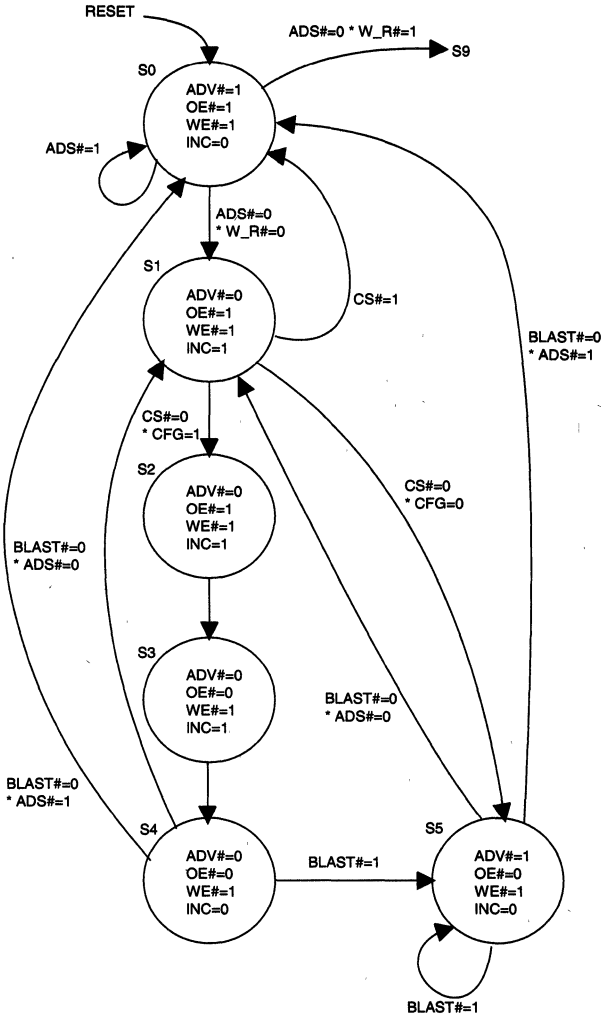
This interface supports processor boot from the 28F016XS memory space after power-up or reset. However, the boot code must follow some restrictions until it has properly configured the 28F016XS-15s, the i960 CA microprocessor and the CFG state machine input valve. In the default configuration state, the i960 CA microprocessor supports only non-burst reads and writes. Program control should jump to an area of RAM to execute the configuration sequence. The code must configure the 28F016XS-15s and all necessary i960 CA microprocessor programmable attributes before setting the CFG input to logic "1." Table 1 illustrates the required configuration settings for both the 28F016XS-15s and the i960 CA-33 microprocessor.

Table 1. Configuration Settings for the 28F016XS-15 and i960 CA-33 Microprocessor Employing Address Pipelining at 33 MHz

Part	Parameter	Setting
28F016XS-15 (5V V_{CC})	SFI Configuration	2
i960 CA-33 Microprocessor	Ready Inputs	OFF
	Byte Ordering	LITTLE ENDIAN
	Bus Width	32-BIT
	Wait States: Nrad	3
	Nrdd	0
	Nwad	2
	Nwdd	2
	Nxda	0
	Address Pipelining	ON
	Burst mode	ON

2.3 Single and Burst Read Cycle Description at 33 MHz

Refer to the read cycle timing diagrams (Figures 7 and 8) and the state diagram (Figure 6) for the following read cycle discussion.



NOTE:
OE# and WE# are clocked on the inverted CLK edge

Figure 6. Read State Diagram of Single and Address Pipelined Burst Control Interface Shown in Figure 3

Initial Configuration

Figure 7 illustrates a read cycle with the 28F016XS-15s and i960 CA microprocessor in a reset/power-up configuration state. The initial configuration permits only non-burst transfers. The i960 CA microprocessor initiates a read cycle by asserting ADS# with W/R# = "0," presenting the valid address and control signals. At N = 1, the two-bit counter loads the values on address bits A_{3-2}. The state machine asserts ADV# for the next clock edge (N = 2), where the 28F016XS-15 will clock in the address if CS# is asserted. If CS# is not asserted, the state machine returns to inactive state at N = 2. The state machine asserts ADV# for only one clock edge before entering a hold state to await the assertion of BLAST# by the i960 CA microprocessor. The state machine asserts OE# (to meet timing requirements OE# is falling-edge triggered) on the falling edge between N = 2 and N = 3 to enable the 28F016XS-15 data output buffers. With SFI Configuration = 4, the data will be valid at the N = 7. The 28F016XS-15s will hold data on the bus until the i960 CA microprocessor asserts BLAST#. During the clock period following N = y, the state machine returns to its inactive state, de-asserting OE# to tri-state the 28F016XS-15 data outputs.

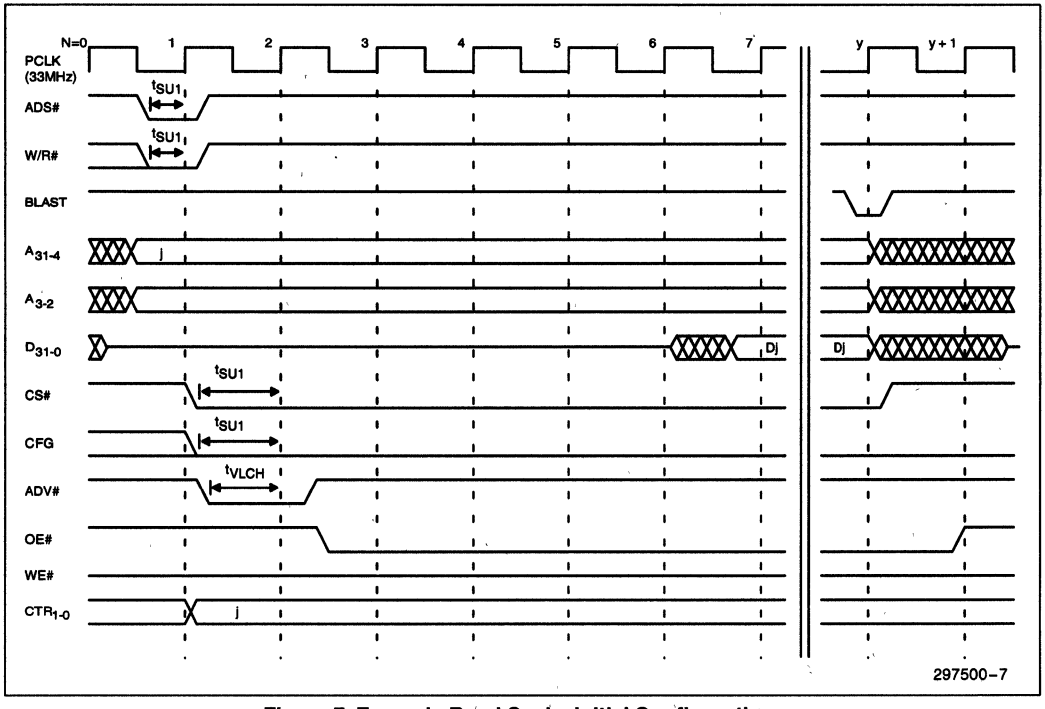


Figure 7. Example Read Cycle, Initial Configuration, Showing Key Specifications Requiring Consideration

Optimized Configuration

Figure 8 illustrates a two double-word burst read followed by a four double-word burst read with the 28F016XS-15s, i960 CA microprocessor and state machine configured for optimum read performance. With CFG = 1, the counter increments the two lower bits of the address at N = 2, N = 3 and N = 4, and ADV# remains asserted so that the 28F016XS-15 latches in four successive addresses at N = 2 through 5. With SFI Configuration = 2, the first data will be valid at N = 5. If a second read cycle follows the current read cycle, the i960 CA microprocessor will assert ADS# one clock after asserting BLAST#. The state machine will respond by immediately re-entering the read cycle. After detecting the assertion of BLAST#, the state machine will return to its inactive state waiting for a new access targeting the 28F016XS memory space.

When implementing the i960 CA microprocessor address pipelining capability, the state machine controlling CS# monitors the upper address lines, ADS# and BLAST#. CS# is held active upon detecting an access targeting the flash memory space until BLAST# is asserted with ADS# de-asserted. When BLAST# and ADS# are active at the same time, a pipelined read access is in progress. The CS# state machine examines the upper address lines to determine whether or not the current pipelined access is aimed at the 28F016XS memory space.

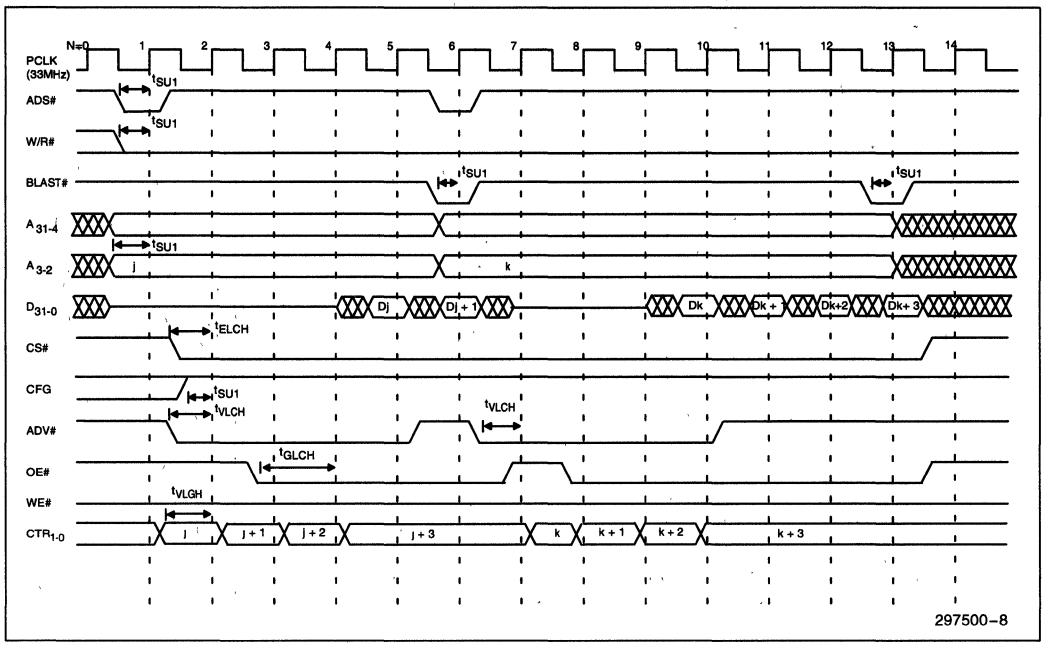


Figure 8. Example Two Double-Word Burst Read Followed by Pipelined Four Double Word Burst Read Showing Key Specifications Requiring Consideration

Critical Timings

Table 2 describes the critical timings illustrated in Figures 7 and 8. One particularly critical timing in this designs, is the data hold time. The i960 CA-33 microprocessor requires a 5 ns hold time after the clock edge. The 28F016XS-15 guarantees a 5 ns data hold after clock, meeting the processor's hold requirement with 0 ns of margin.

This design provides 7 ns of margin in meeting the 3 ns setup time of the i960 CA-33 microprocessor data inputs, outputting data t_{CHQV} after a rising CLK edge.

Another critical area concerns CS# during pipelined read accesses. Since the 28F016XS-15 specifies zero data hold from CE# going high, the chip select state machine must hold CS# active for 5 ns to satisfy the i960 CA-33 microprocessor data input hold specification of 5 ns. Hence, the chip select state machine holds CS# active for an additional clock period after detecting BLAST# active.

The i960 CA-33 microprocessor control outputs ADS# and W/R# have 3 ns of margin and BLAST# has 5 ns of margin to meeting the 85C22V10-15 input setup requirement.

Consult the appropriate datasheets for full timing information.

Table 2. Example Read Cycle Timing Specifications at 5V V_{CC}

Part	Symbol	Parameter	Minimum Specified Value (ns)
85C22V10-15	t_{SU1}	Input Setup Time to CLK	9
i960 CA-33 Microprocessor	T_{IS1}	Input Setup D_{31-0}	3
	T_{IH1}	Input Hold D_{31-0}	5
28F016XS-15	t_{ELCH}	CE# Setup to CLK	25
	t_{VLCH}	ADV# Setup to CLK	15
	t_{AVCH}	Address Setup to CLK	15
	t_{GLCH}	OE# Setup to CLK	15

NOTE:
Consult appropriate datasheets for up-to-date specifications.

2.4 Single Burst Write Cycle Description at 33 MHz

Refer to the write cycle timing diagrams and the state diagram (Figure 9) for the following write cycle discussion.

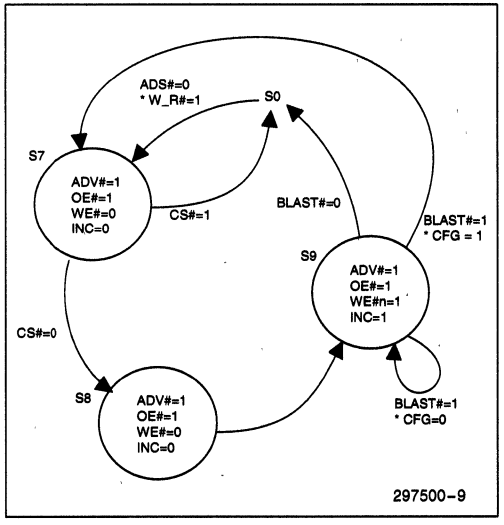


Figure 9. Write State Diagram of Control Interface Shown in Figure 3

Initial Configuration

Figure 10 illustrates a write cycle. In the reset/power-up configuration state, the interface supports only non-burst writes. The i960 CA microprocessor initiates a write cycle by asserting ADS# with W/R# = "1," presenting a valid address and control signals. At N = 1, the two-bit counter loads the values on address bits A_{3-2}. The state machine asserts WE# (to meet timing requirements, WE# is falling-edge triggered) on the falling edge between N = 1 and N = 2. WE# remains asserted for two clock periods, in order to meet the 28F016XS-15 timing requirements. The state machine then enters a holding state until the processor asserts BLAST#, after which time the interface state machine will return to S0.

Optimized Configuration

Figure 11 illustrates a two double-word burst write with CFG = "1." When the first data write is complete at N = 4, the counter increments the two lower address bits, and the state machine asserts WE# on the next falling clock edge to begin the next the 28F016XS-15 data write. The i960 CA microprocessor must provide the next data during the clock period following N = 4. The data writes continue to the next consecutive addresses until the i960 CA microprocessor asserts BLAST#, indicating the end of the burst write cycle.

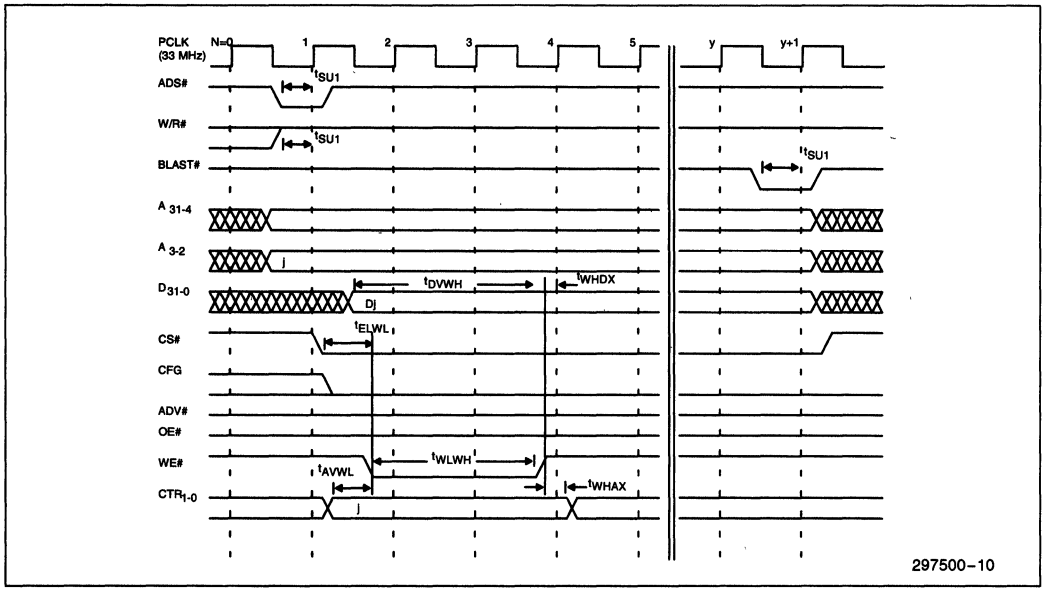


Figure 10. Example Write Cycle, Initial Configuration, Showing Key Specifications Requiring Consideration

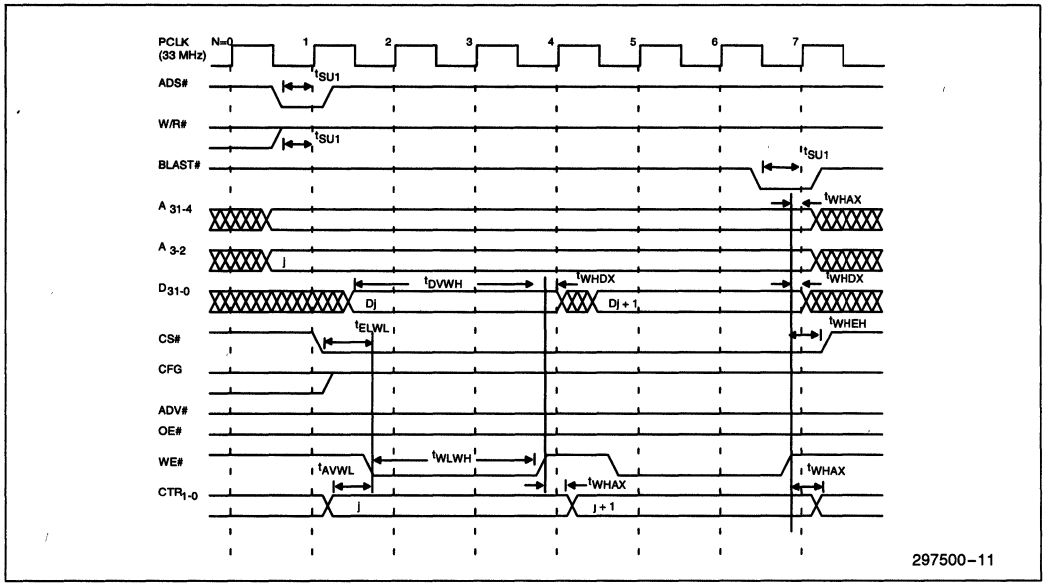


Figure 11. Example Two Double-Word Burst Write Illustrating Key Specifications Requiring Considerations

Critical Timings

Table 3 describes the critical timings illustrated in Figures 10 and 11.

One critical hold time to notice is t_{WHAX}. WE# is guaranteed to transition within 8 ns from the falling clock edge. Therefore, the t_{WHAX} requirement has 2 ns of margin on CTR_{1-0}, and 5 ns of margin on A_{31-4}.

Also notice that CTR_{1-0} must be valid before WE# is asserted. CTR_{1-0} are guaranteed valid 8 ns after the rising clock edge, providing 9 ns of margin.

Consult the appropriate datasheets for full timing information.

Table 3. Example Write Cycle Timing Parameters at 5V V_{CC}

Part	Symbol	Parameter	Minimum Specified Value (ns)
85C22V10-15	t_{SU1}	Input Setup Time to CLK	9
28F016XS-15	t_{ELWL}	CE# Setup to WE# Going Low	0
	t_{AVWL}	Address Setup to WE# Going Low	0
	t_{WLWH}	WE# Pulse Width	50
	t_{DVWH}	Data Setup to WE# Going High	50
	t_{WHDX}	Data Hold from WE# High	0
	t_{WHAX}	Address Hold from WE# High	5
	t_{WHEH}	CE# Hold from WE# High	5

NOTE:
Consult appropriate datasheets for up-to-date specifications.

3.0 i960® JF-33 MICROPROCESSOR INTERFACE

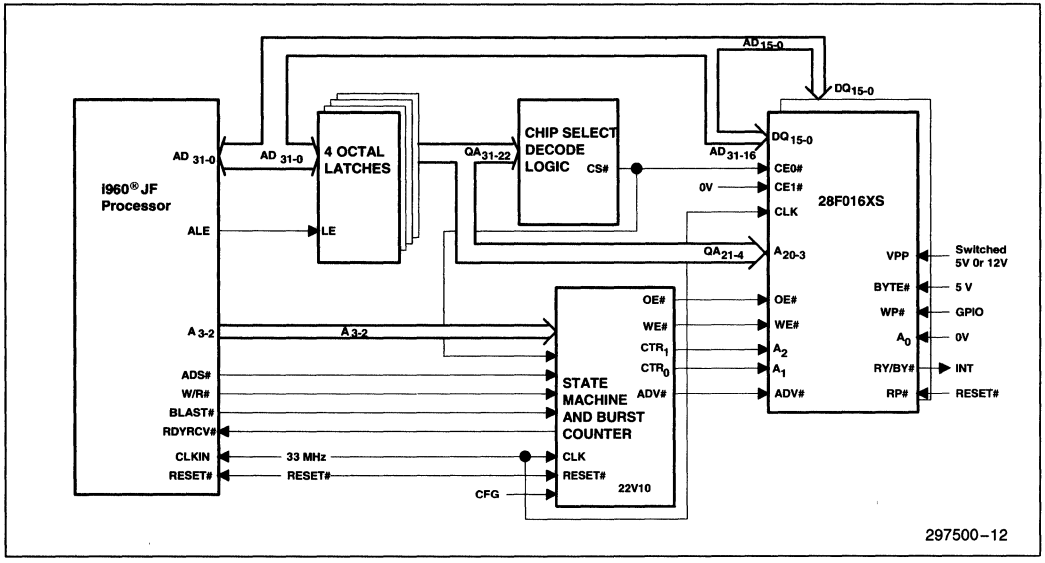


Figure 12. Minimal Interface Logic Required in Interfacing the 28F016XS-15 to the i960 JF-33 Microprocessor

Using this interface, the 28F016XS-15/i960 JF-33 microprocessor system can achieve 3-0-0-0 wait state read performance, supporting burst transfers.

3.1 Circuit Description

This design (Figure 12) uses two 28F016XS-15s to match the 32-bit data bus of the i960 JF microprocessor, providing 4 Mbytes of flash memory. Four octal latches, enabled by the ALE signal, de-multiplex the 32-bit address from the AD bus. The latched address bits QAD_{21-4} and the counter outputs CTR_{1-0} from the PLD select locations within the 28F016XS memory space. The two-bit counter implemented in the PLD loads the address bits on A_{3-2}, at the beginning of each memory cycle, and generates the lower two bits of the burst addresses on its outputs CTR_{1-0} to the 28F016XS-15.

CLK Option

A 33 MHz clock signal drives the i960 JF microprocessor CLKIN input and the PLD and 28F016XS-15s CLK input.

Reset

An active-low reset signal, RESET#, connects to the RESET# inputs of the i960 JF microprocessor and PLD and to the RP# input of the 28F016XS-15. Figure 5 illustrates a suggested logic configuration for generating RESET#.

Interface Control Signals

ADS# and W/R# i960 JF microprocessor signals, just as in the i960 CA microprocessor design, serve as inputs to the state machine, which controls the two-bit counter and generates the OE#, WE# and ADV# signals for the 28F016XS-15s. The state machine also generates the RDYRCV# signal for the i960 JF microprocessor to control the insertion of wait states during data transfers.

Configuration Signal

A general purpose input/output (GPIO) generates the configuration signal for input to the state machine. The configuration signal must be reset to logic "0" on power-up and system reset to ensure that the operation of the state machine matches the 28F016XS-15s. After optimizing the 28F016XS-15s, the reconfiguration signal must switch to a logic "1" to take advantage of the new configuration.

Additional Control Signals

For information regarding BYTE#, WP#, RY/BY# and Vpp, see Section 2.1.

3.2 Software Interface Considerations

Boot-up Capability

This interface supports processor boot-up from the 28F016XS-15 memory space after power-up or reset. Burst reads and writes may commence with no configuration. However, read wait state performance will be 5-1-1-1 until the SFI Configuration is set to 2 and the CFG input is set to logic "1." Program control should jump to an area of RAM to execute the configuration sequence. A pseudocode flow for this configuration sequence is shown below.

Execute Device Configuration command sequence
Activate CFG signal
End

The SFI Configuration must be set to 2 before the CFG input is set to logic "1." Thereafter, burst read wait state performance will improve to 3-0-0-0.

3.3 Burst Read Cycle Description at 33 MHz

Refer to the read cycle timing diagrams and state diagram (Figure 13) for the following discussions of the read cycle.

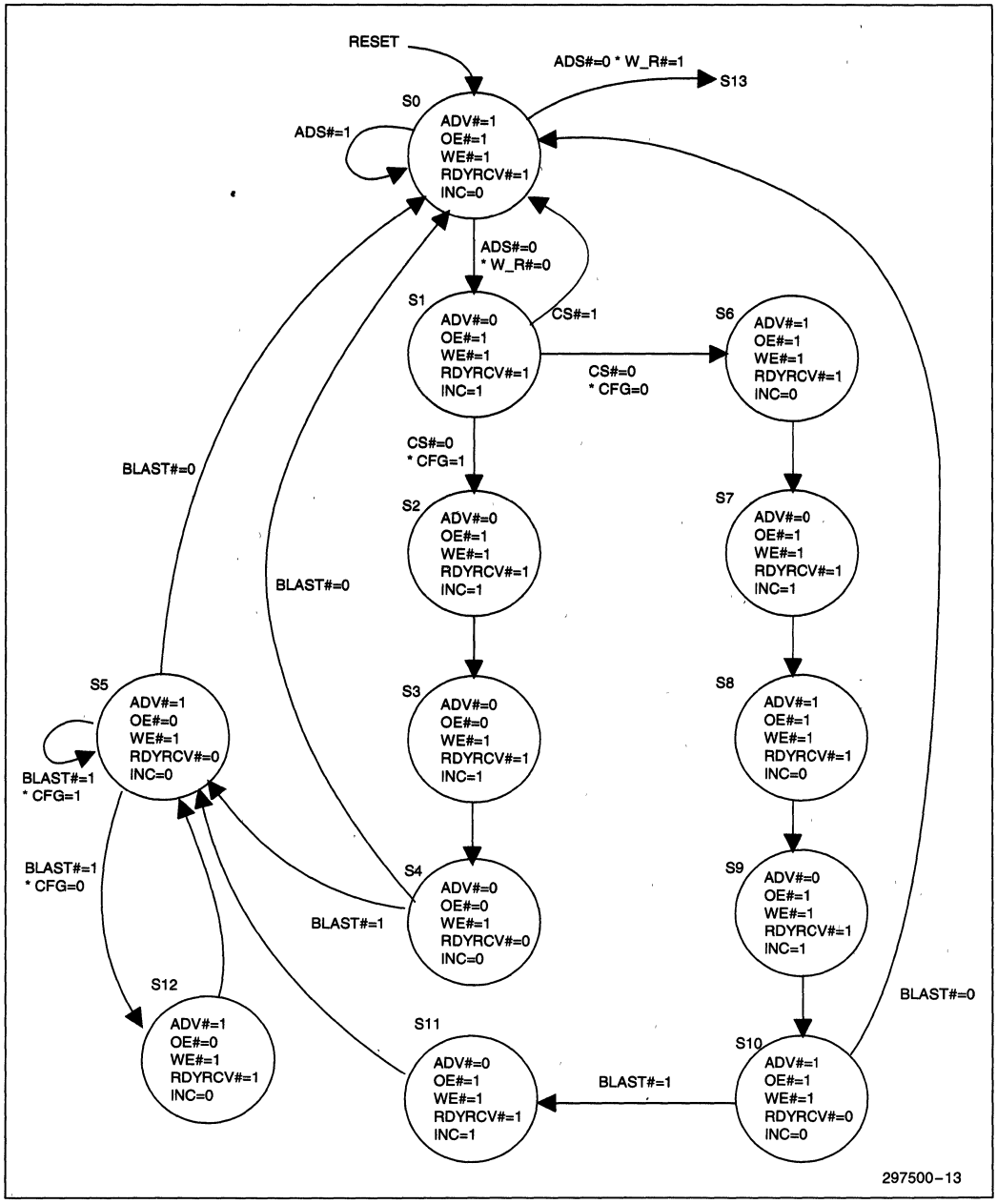


Figure 13. Read State Diagram of Burst Control Interface Shown in Figure 12

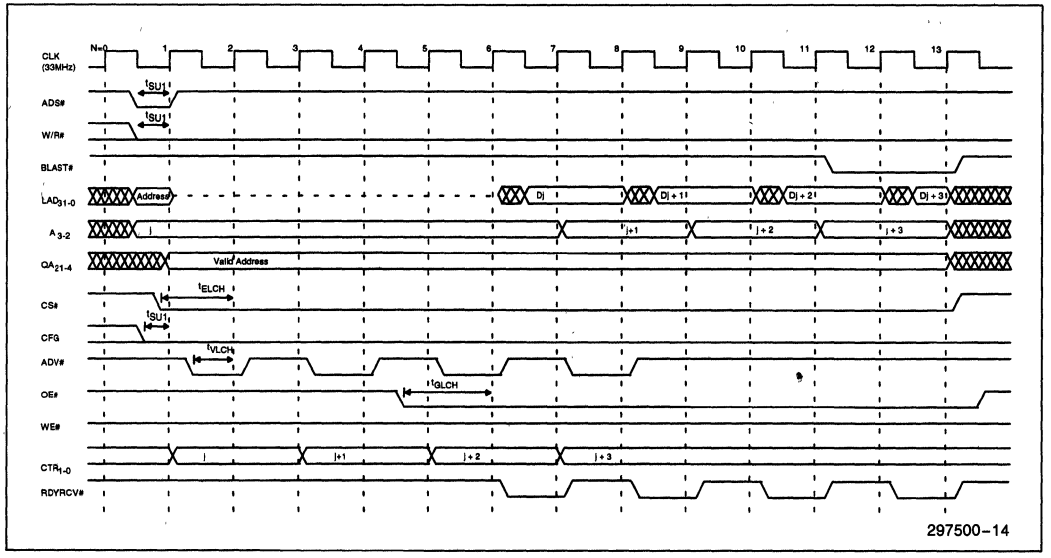


Figure 14. Example Four Double-Word Burst Read in Initial Configuration Showing Key Specifications Requiring Consideration

Initial Configuration

Figure 14 illustrates a four double-word burst read cycle with the 28F016XS-15s and the state machine in the reset/power-up configuration state. The i960 JF microprocessor initiates a read cycle by asserting ADS# with W/R# = "0", presenting a valid address and control signals. At N = 1 with ADS# = "0", the two-bit counter loads the values on the address bits A_{3-2}.

The state machine asserts ADV# after clock edge N = 1 where the 28F016XS-15s will clock in the first address at the next rising clock edge (N = 2), if CS# is asserted. If CS# is not asserted, the state machine will return to its inactive state at N = 2.

The state machine deactivates ADV# at N = 2. The state machine then asserts ADV# at N = 3 to load the next read address into the 28F016XS-15s. De-asserting ADV# for one clock cycle (at N = 2, 4, 6 and 8) between accesses forces the 28F016XS-15s to hold data output for two clock cycles (access stretching), which allows time for the data to stabilize and meet the timing requirements of the i960 JF microprocessor bus.

The counter increments the two lower bits of the address at N = 3, 5 and 7 to provide the four successive burst addresses. The state machine asserts OE# (to meet timing requirements OE# is falling-edge triggered) on the falling edge between N = 4 and N = 5 to enable the 28F016XS-15 data output buffers. With the SFI Configuration = 4, the data will be valid at the i960 JF microprocessor data inputs at N = 7.

The state machine asserts RDYRCV# to inform the i960 JF microprocessor that the data is valid. RDYRCV# is returned active at N = 7, 9, 11 and 12. The interface will follow this methodology until the processor asserts BLAST#, which identifies the end of the burst transaction. BLAST# is examined at N = 7, 9, 11 and 13. The interface will transition to its inactive state, S0, after the assertion of BLAST#.

Optimized Configuration

Figure 15 illustrates a four double-word burst read with the 28F016XS-15s and state machine configured for optimum read performance. With the SFI Configuration = 2, ADV# is held active and the counter increments at N = 2, 3 and 4, supplying the 28F016XS-15s with four consecutive accesses. Data from the initial access will be valid for transfer at N = 5. Subsequent data will be valid at N = 6, 7 and 8. This interface and 28F016XS-15 configuration improve read wait-state performance to 3-0-0-0. All other signal monitoring and generation are identical to the reset/power-up configuration read cycle documented in the preceding section.

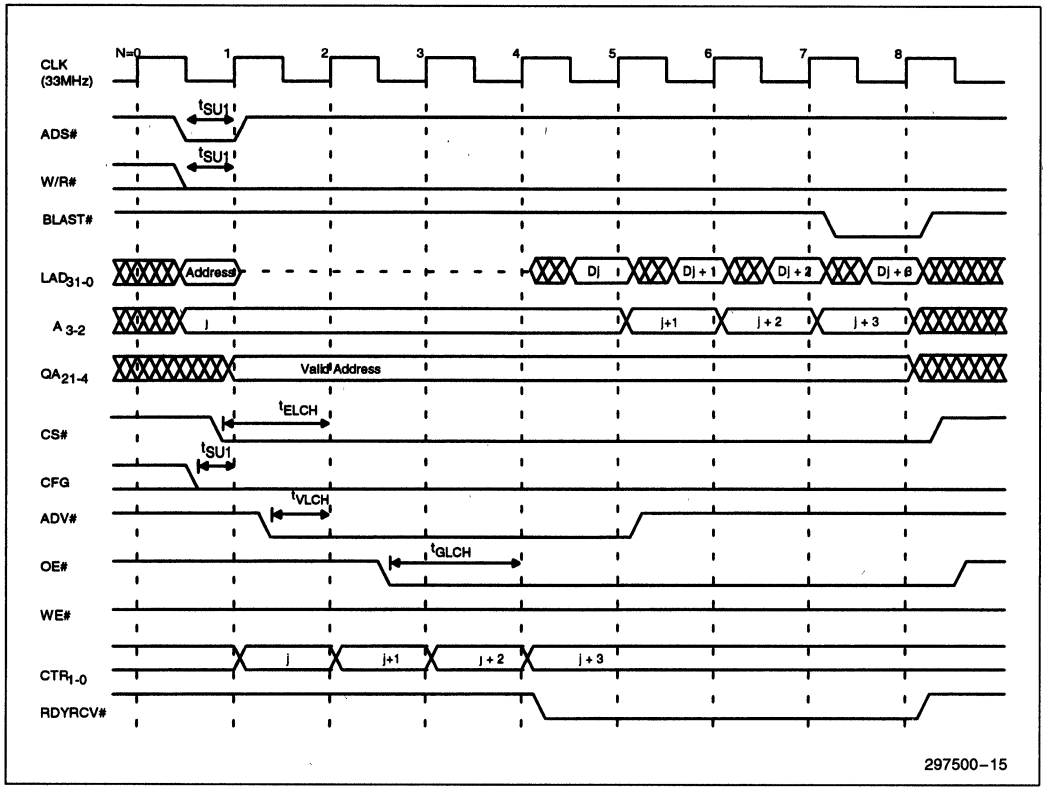


Figure 15. Example Four Double-Word Burst Read Illustrating Important Timing Parameters Requiring Consideration

Critical Timings

Table 4 describes the critical timings illustrated in Figures 14 and 15. One particularly critical timing in this design is the data hold time, which the 28F016XS-15 meets with 0 ns margin. The 28F016XS holds data for 5 ns after a rising clock.

The 28F016XS-15 will provide data 10 ns before the rising edge of the system clock, which satisfies the i960 JF-33 microprocessor's data input requirement.

ADV# and CTR_{1-0} are guaranteed valid 8 ns after the rising clock edge. Setup times for these inputs to 28F016XS-15 are each 15 ns. Since the clock period is 30 ns, this allows 7 ns margin for these timings.

RDYRCV# is guaranteed valid 8 ns after the rising clock edge to met the microprocessor's setup time to rising clock edge.

Consult the appropriate datasheets for full timing information.

Table 4. Example Write Cycle Timing Parameters at 5V V_{CC}

Part	Symbol	Parameter	Minimum Specified Value (ns)
85C22V10-15	t_{SU1}	Input Setup Time to CLK	9
28F016XS-15	t_{ELCH}	CE# Setup to CLK	25
	t_{VLCH}	ADV# Setup to CLK	15
	t_{AVCH}	Address Setup to CLK	15
	t_{GLCH}	OE# Setup to CLK	15

NOTE:
Consult appropriate datasheets for up-to-date specifications.

3.4 Burst Write Cycle Description at 33 MHz

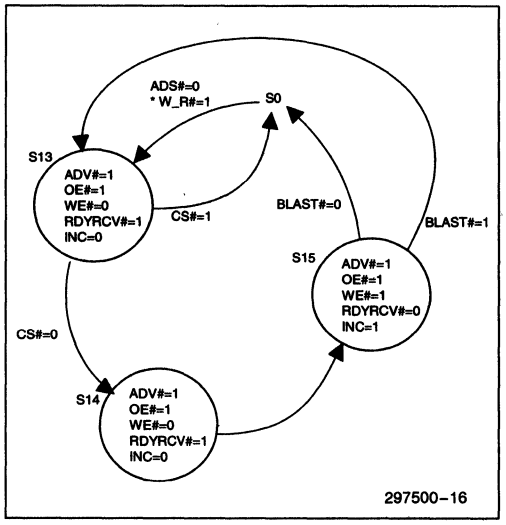


Figure 16. Write State Diagram of Burst Control Interface Shown in Figure 12

Write Configuration

Figure 17 illustrates a two double-word burst write cycle. The i960 JF microprocessor initiates a write cycle by asserting ADS# with W/R# = 1 and presenting a valid address and control signals. At N = 1 with ADS# = 0, the two-bit counter loads the values on the address bits A_{3-2}. The state machine asserts WE# (to meet timing requirements, WE# is falling-edge triggered) on the falling edge between N = 1 and N = 2. WE# remains asserted for four clock periods, in order to meet 28F016XS-15 timing requirements. The state machine asserts RDYRCV# for N = 4 to inform the i960 JF microprocessor to supply the next data. At N = 4, the counter increments the two lower address bits, and the state machine asserts WE# on the next falling clock edge to begin the next data write to the 28F016XS-15s. The data writes continue until the processor asserts BLAST#, noting the end of the current write transaction. The SFI Configuration has no effect on the write cycle.

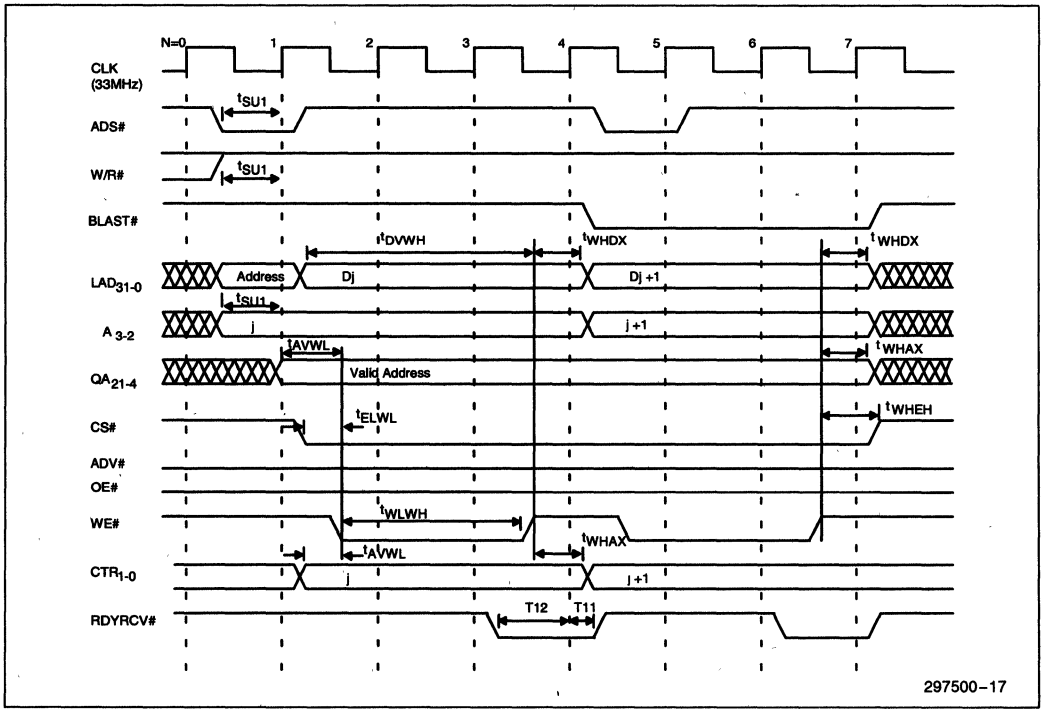


Figure 17. Two Double-Word Burst Write

Critical Timings

Table 5 describes the critical timings illustrated in Figure 17.

Notice that CTR_{1-0}, and CS# must be valid before WE# is asserted. CTR_{1-0} are guaranteed valid 8 ns after the rising clock edge, providing 12 ns of margin.

Consult the appropriate datasheets for full timing information.

Table 5. Example Write Cycle Timing Parameters at 5V V_{CC}

Part	Symbol	Parameter	Minimum Specified Value (ns)
85C22V10-15	t_{SU1}	Input Setup Time to CLK	9
28F016XS-15	t_{ELWL}	CE# Setup to WE# Going Low	0
	t_{AVWL}	Address Setup to WE# Going Low	0
	t_{WLWH}	WE# Pulse Width	50
	t_{DVWH}	Data Setup to WE# Going High	50
	t_{WHDX}	Data Hold from WE# High	0
	t_{WHAX}	Address Hold from WE# High	5
	t_{WHEH}	CE# Hold from WE# High	5

NOTES:
Consult appropriate datasheets for up-to-date specifications.

4.0 I960 KB-25 MICROPROCESSOR INTERFACE

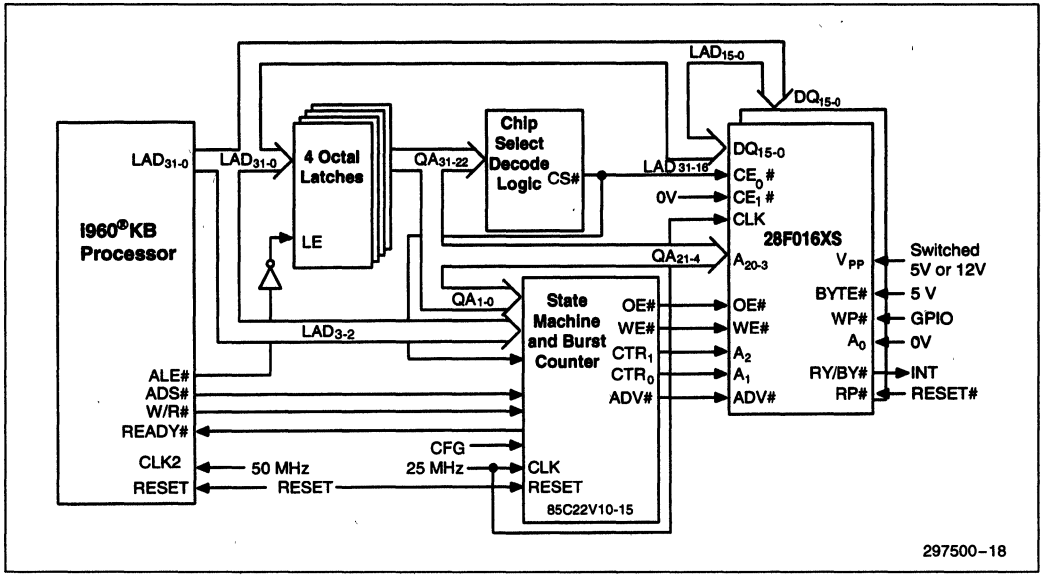


Figure 18. Minimal Logic Required in Interfacing 28F016XS-20 to the i960 KB-25 Microprocessor

Using this interface, the 28F016XS-20/i960 KB-25 microprocessor system can achieve 3-0-0-0 wait state read performance (5-1-1-1 read performance) in terms of the CPU's internal 25 MHz CLK, supporting burst cycles. This design operates the logic and 28F016XS-20s at 25 MHz.

4.1 Circuit Description

This interface uses two 28F016XS-20s to match the 32-bit data bus of the i960 KB microprocessor, providing access to 4 Mbytes of flash memory. Like the 28F016XS-20 interface to the i960 JF microprocessor, four octal latches de-multiples the 32-bit address from the LAD bus. The latches are enabled by an inverted ALE# signal from the microprocessor. The latched address and two-bit counter integrated into the PLD select locations with the flash memory space. The two-bit counter loads the address bits on LAD_{3-2} at the beginning of each memory cycle, and generates the lower two bits of the burst addresses on its outputs CTR_{1-0}.

CLK Option

In this interface, a 50 MHz clock signal drives the i960 KB microprocessor CLK2 input, and an external 25 MHz clock signal, synchronized to the i960 KB microprocessor internal 25 MHz clock, drives the CLK inputs for the 28F016XS-20s and the PLD. The reduced clocking frequency places a less stringent demand on the PLD to meet setup times, thereby allowing the usage of a slower PLD.

External 25 MHz CLK generation and synchronization can be accomplished using simple flip-flop logic. The i960 KB microprocessor synchronizes its internal 25 MHz clock on the falling edge of RESET.

Two different methods for generating a 25 MHz CLK are illustrated in Figure 19, and the PLD equations are located in the Appendix A.

The two 1X clock generation methods in Figure 19 will produce a different amount of clock skew between the 2x and 1x CLK. Clock generation method (A) will have a maximum skew of T_{CO1}, while method (B) will have little to no clock skew because the 2x and 1x CLKs are exposed to the same PLD delay (see Figure 20).

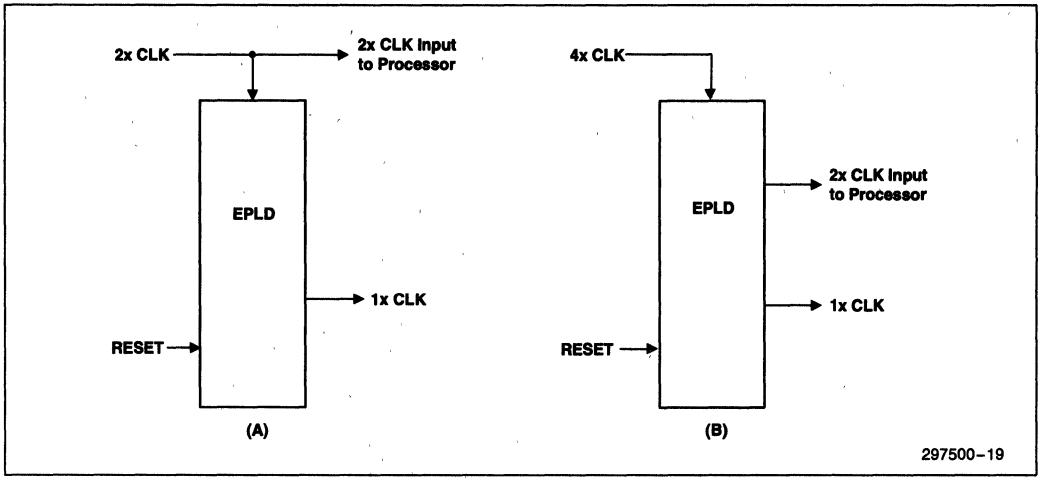


Figure 19. Example 1x CLK Generation and Synchronization Circuitry

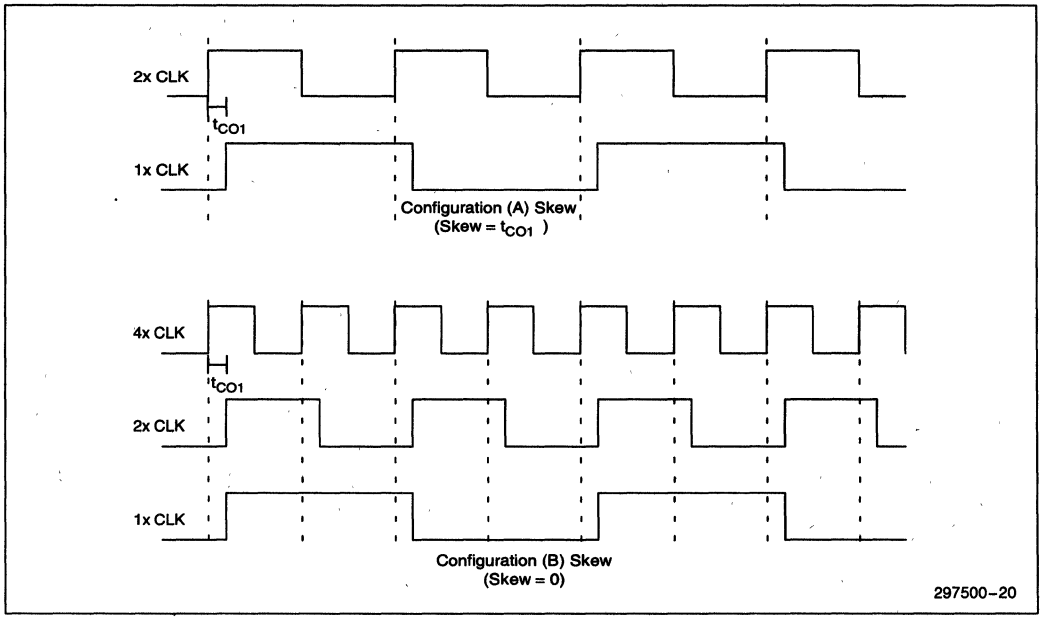


Figure 20. CLK Skew Produced by the Two Different CLK Generation Methods Illustrated in Figure 19

4.2 Software Interface Considerations

Boot-up Capability

This interface supports processor boot-up from the 28F016XS-20 memory space after power-up or reset. Burst reads and writes may commence with no configuration. However, read wait-state performance will be 5-0-2-0 until the SFI Configuration is set to 2 and the CFG input is set to logic "1." Thereafter, burst transfers will improve to 3-0-0-0 wait state performance. Program control should jump to an area of RAM to execute the configuration sequence. The code must set the SFI Configuration to 2 before setting the CFG input to logic "1."

4.3 Read Burst Cycle Description at 25 MHz

Refer to the read cycle timing diagrams and the state diagram (Figure 21) for the following discussion of the read cycle.

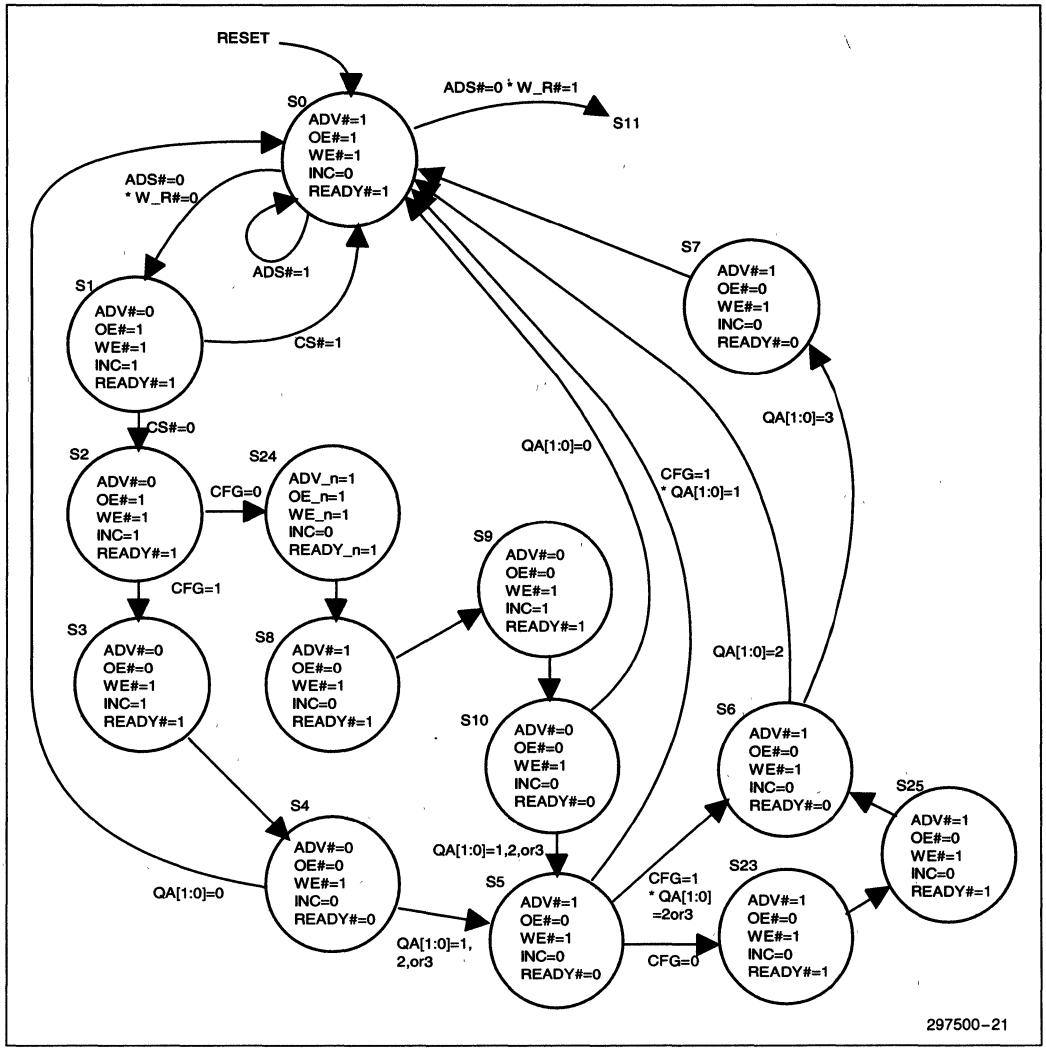


Figure 21. Read State Diagram of Burst Control Interface Logic Shown in Figure 18

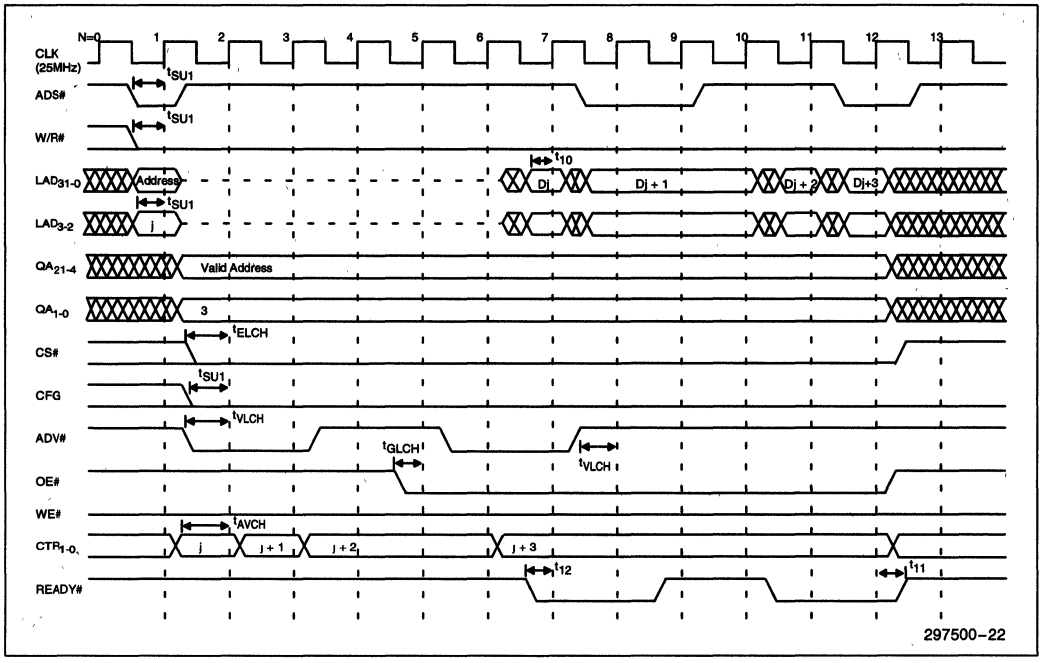


Figure 22. Example Four Double-Word Burst Read in Initial Configuration Showing Key Specifications Requiring Consideration

Initial Configuration

Figure 22 illustrates a four double-word burst read cycle with the 28F016XS-20s and state machine in the reset/power-up configuration state. The i960 KB microprocessor initiates a read cycle by asserting ADS# with W/R# = "0," presenting valid address and control signals.

At N = 1, the two-bit counter loads the values on address bits A_{3-2}. The state machine asserts ADV# at the next clock edge (N = 2), where 28F016XS-20 will latch in the address if CS# is asserted. If CS# is not asserted, the state machine will return to its inactive state at N = 2. The counter increments the two lower bits of the address at N = 2 and ADV# remains asserted so that the 28F016XS-20 clocks in the first two burst addresses at N = 2 and 3.

With the SFI Configuration = 4, the state machine must wait 4 clock periods between loading two even or two odd addresses into 28F016XS-20. Therefore, the state machine asserts ADV# at N=6 and N=7 to latch the third and fourth addresses, respectively, into 28F016XS-20. The state machine asserts OE# (to meet timing requirements OE# is falling-edge triggered) on the falling edge between N = 4 and N = 5 to enable the 28F016XS-20 data output buffers. Data will be valid at the i960 KB microprocessor data inputs at N = 7.

The state machine asserts READY# to inform the i960 KB microprocessor that the data is valid. To compensate for the delayed third and fourth burst addresses, the state machine must de-assert READY# for N third and fourth addresses, respectively, N=9 and N=10 before again asserting READY# for N=11 and N=12, when the third and fourth data, respectively, become valid.

The i960 KB microprocessor encodes the length of the burst transfer onto the two lowest bits of the initial address. Therefore, the state machine decodes QAD_{1-0} to determine the end of the burst transfer. At this point, the state machine will return to its inactive state, S0, de-asserting OE# to tri-state the 28F016XS-20's data outputs.

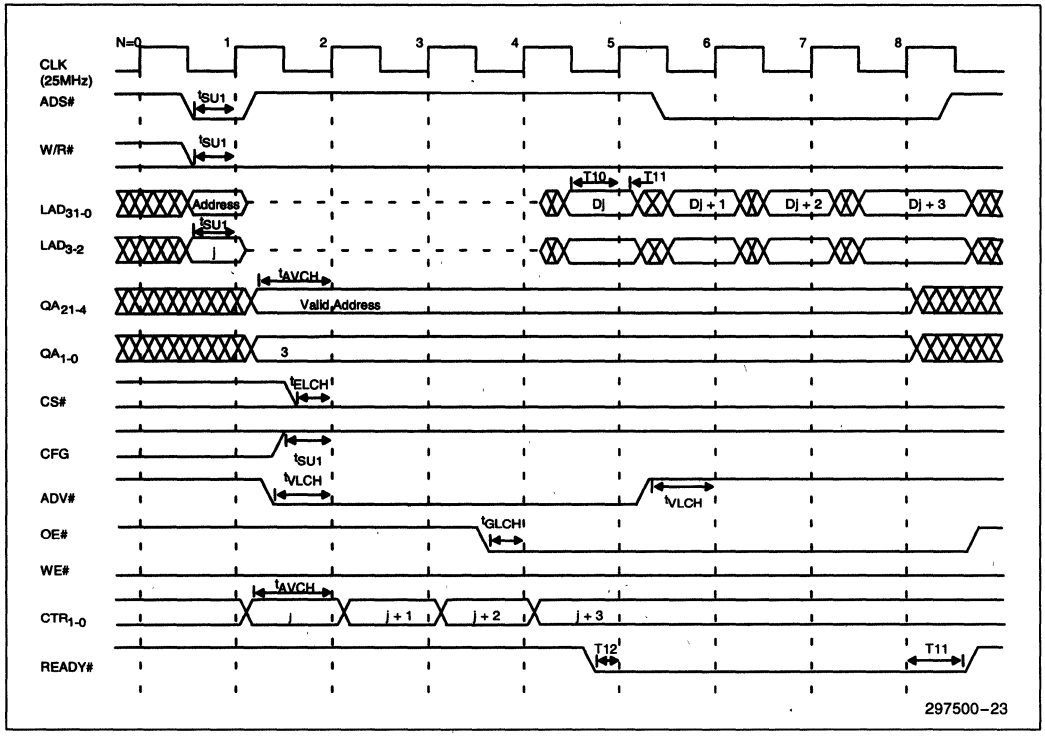


Figure 23. Example Four Double-Word Burst Read Illustrating Key Specifications Requiring Consideration

Optimized Configuration

Figure 23 illustrates a four double-word burst read with the 28F016XS-20s and state machine configured for optimum read performance. With the SFI Configuration = 2, 28F016XS-20 can accept addresses with only two clock periods between each even address and two clock periods between each odd address. Therefore, the four burst addresses flow into SFI on successive rising clock edges and the four data become valid for transfer on successive rising clock edges with the first data valid by N = 5. Otherwise, the transaction is similar to the reset/power-up configuration read cycle.

Critical Timings

Table 6 describes the critical timings illustrated in Figures 22 and 23. One particularly critical timing in this design is the data hold time. The i960 KB-25 microprocessor requires a 5 ns hold time after the clock edge. The 28F016XS-20 guarantees a 5 ns data hold after clock, meeting the setup requirement with 0 ns of margin. This design provides:

$$1/25\ \text{MHz} - t_{10} = 7\ \text{ns}$$

of margin to meeting the 3 ns setup time of the i960 KB-25 microprocessor data inputs.

Another critical area concerns CS# and QA_{21-4} setup time to 28F016XS-20. This design allows two clock periods (80 ns) for these signals to stabilize and meet the 28F016XS-20 setup time. Since the i960 KB-25 microprocessor guarantees ALE# and LAD_{31-0} 18 ns after the rising clock edge and the 28F016XS-20 setup time is 30 ns, this leaves:

$$2*1/25\ \text{MHz} - t_6 - t_{ELCH} = 32\ \text{ns}$$

for the propagation delays of the inverter plus the latch plus the chip select decode logic (if applicable, for CS#).

READY# is guaranteed valid 8 ns after the falling clock edge, providing 5 ns of margin on the 7 ns setup required by the i960 KB-25 microprocessor.

OE# is also guaranteed valid 8 ns after the falling clock edge, providing 0 ns of margin on the 12 ns setup requirement.

Consult the appropriate datasheets for full timing information.

Table 6. Example Read Burst Cycle Timing Parameters at 5V V_{CC}

Part	Symbol	Parameter	Minimum Specified Value (ns)
85C22V10-15	t_{SU1}	Input Setup Time to CLK	9
i960 KB-25 Microprocessor	T_{10}	Input Setup 1	3
	T_{11}	Input Hold	5
	T_{12}	Input Setup 2	7
28F016XS-20	t_{ELCH}	CE# Setup to CLK	30
	t_{VLCH}	ADV# Setup to CLK	20
	t_{AVCH}	Address Setup to CLK	20
	t_{GLCH}	OE# Setup to CLK	20

4.4 Write Burst Cycle Configuration at 25 MHz

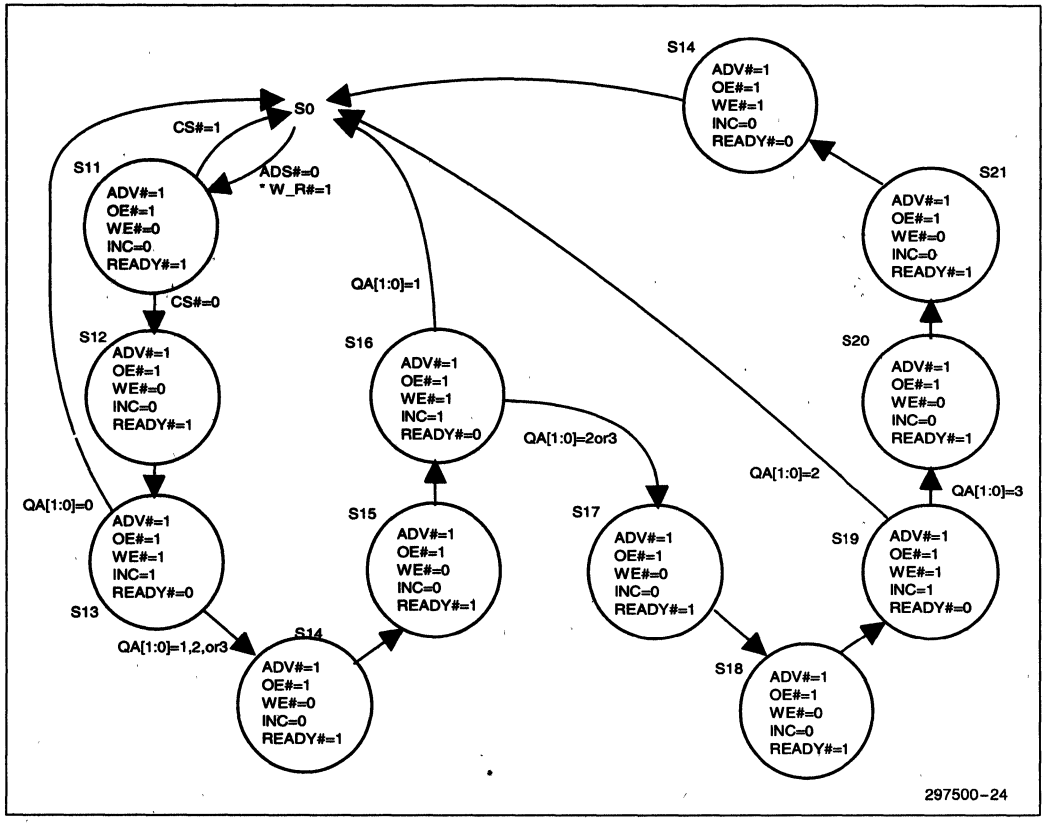


Figure 24. Write State Diagram of Burst Control Interface Logic in Figure 18

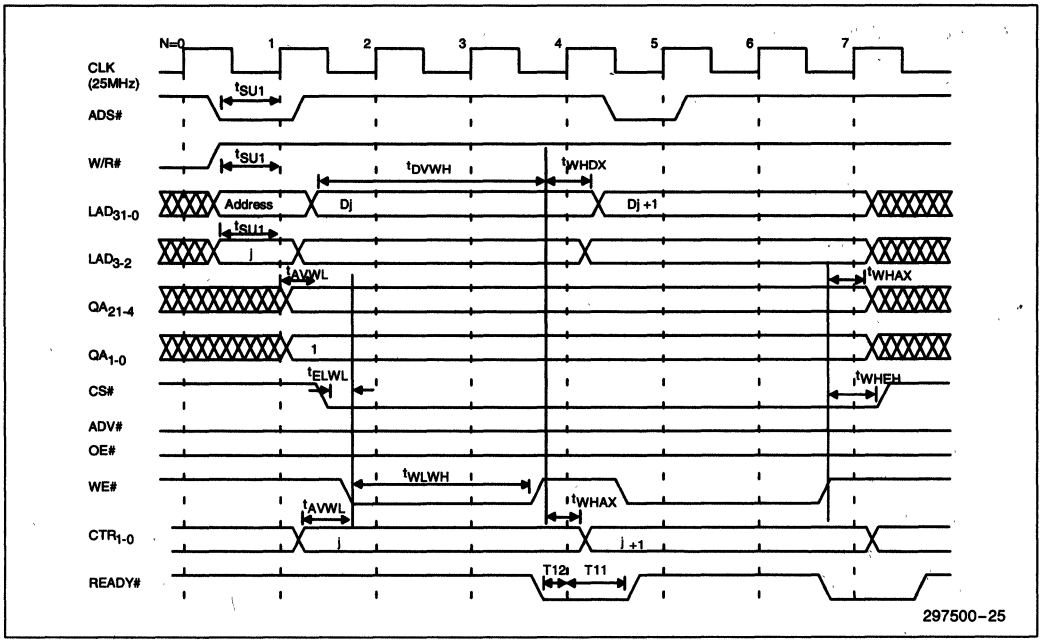


Figure 25. Example Two Double-Word Burst Write Showing Key Timing Parameters Requiring Consideration

Write Considerations

The i960 KB microprocessor initiates a write cycle by asserting ADS# with W/R# = 1 and presenting a valid address and control signals. At N = 1 with ADS# = 0, the two-bit counter loads the values on the address bits LAD_{3-2}. The state machine asserts WE# (to meet timing requirements, WE# is falling-edge triggered) on the falling edge between N = 1 and N = 2. WE# remains asserted for two clock periods, in order to meet the 28F016XS-20 timing requirements. The state machine asserts READY# for N = 4 to inform the i960 KB microprocessor to supply the next data. At N = 4, the counter increments the two lower address bits, and the state machine asserts WE# on the next falling clock edge to begin the next data write to the 28F016XS-20s. The data writes continue until the burst cycle is complete. The state machine determines the length of the burst cycle by decoding QAD_{1-0}. The 28F016XS-20 Configuration has no effect on the write cycle.

Critical Timings

Table 7 describes the critical timings illustrated in Figure 25.

One critical hold time to notice is t_{WHAX}. WE# is guaranteed to transition with 8 ns from the falling clock edge. Therefore, the t_{WHAX} requirement has 7 ns of margin on CTR_{1-0} and 9 ns of margin on A_{31-4}.

Also notice that CTR_{1-0} must be valid before WE# is asserted. CTR_{1-0} are guaranteed valid 8 ns after the rising clock edge, providing 12 ns of margin.

Consult the appropriate datasheets for full timing information.

Table 7. Example Write Burst Cycle Timing Parameters at 5V V_{CC}

Part	Symbol	Parameter	Minimum Specified Value (ns)
85C22V10-15	t_{SU1}	Input Setup Time to CLK	9
i960 KB-25 Microprocessor	T_{11}	Input Hold	5
	T_{12}	Input Setup 2	7
28F016XS-20	t_{ELWL}	CE# Setup to WE# Going Low	0
	t_{AVWL}	Address Setup to WE# Going Low	0
	t_{WLWH}	WE# Pulse Width	50
	t_{DVWH}	Data Setup to WE# Going High	50
	t_{WHDX}	Data Hold from WE# High	0
	t_{WHAX}	Address Hold from WE# High	5
	t_{WHEH}	CE# Hold from WE# High	5

5.0 INTERFACING TO OTHER i960 MICROPROCESSORS

i960®CF-16, i960®CF-25 and i960®CF-33 Microprocessors

The i960 CF microprocessor bus interface is completely compatible with the i960 CA microprocessor bus interface. Therefore, the 28F016XS-15 interfaces described above for the i960 CA-33 microprocessor work equally well with the i960 CF-25 and 33 MHz microprocessors.

At 16 MHz, the interface requires a slight modification because the SFI Configuration value at 16 MHz equals 1. The 28F016XS-15 will begin driving the data pin 1 CLK period after initiating a read access. The interface returns READY# to the i960 CF-16 microprocessor, 1 CLK cycle earlier. Therefore, the 28F016XS-15 interface to the i960 CF-16 microprocessor will deliver 3-0-0-0 wait-state read performance.

i960® SA Microprocessor, i960® SB Microprocessor

The 28F016XS's interface to the i960® Sx microprocessor series will be similar to the i960 KB microprocessor interfaces, with the following differences:

- The i960 Sx microprocessor series has a 16-bit data bus multiplexed with the lower 16 of 32 address bits. Therefore, a single 28F016XS will match the width of the data bus.
- Two octal latches will de-multiplex the address/data bus, and the ALE signal, without inversion, will properly enable the latches.
- The i960 Sx microprocessor series supports eight double-word burst transfers. Therefore, the 28F016XS interface will require a three-bit counter to generate the lower three bits of the burst addresses.
- The interface state machine must use the i960 Sx microprocessor BLAST# signal to recognize the end of a burst cycle (see the i960 CA microprocessor state diagrams).

6.0 CONCLUSION

This technical paper has described the interface between the 28F016XS 16-Mbit Flash memory component and the i960 microprocessor family. This simple design has been implemented with a minimal number of components and achieves exceptional read performance. The 28F016XS provides the microprocessor with the non-volatility and update ability of flash memory and the performance of DRAM. For further information about 28F016XS, consult reference documentation for a more comprehensive understanding of device capabilities and design techniques. Please contact your local Intel or distribution sales office for more information on Intel's flash memory products.

ADDITIONAL INFORMATION

Order Number	Document/Tools
290532	28F016XS Datasheet
297500	"Interfacing 28F016XS to the Intel486™ Microprocessor Family"
292147	AP-398, "Designing with the 28F016XS"
292146	AP-600, "Performance Benefits and Power/Energy Savings of 28F016XS Based System Designs"
297372	16-Mbit Flash Product Family User's Manual, 28F016SA/28F016SV/28F016XS/28F016XD
297508	FlashBuilder Utility
Contact Intel/Distribution Sales Office	28F016XS Benchmark Utility
Contact Intel/Distribution Sales Office	28F016XS iBIS Models
Contact Intel/Distribution Sales Office	28F016XS VHDL/Verilog Models
Contact Intel/Distribution Sales Office	28F016XS Timing Designer Library Files
Contact Intel/Distribution Sales Office	28F016XS Orcad and ViewLogic Schematic Symbols

REVISION HISTORY

Number	Description
001	Original Version

APPENDIX A
PLD FILES

PLD File for the 28F016xs Interface to the i960 CA-33 Microprocessor

```
Title          28F016XS/ i960 CA Microprocessor Interface State Machine
Pattern        PDS
Revision       1
Authors        Example
Company        Intel Corporation - Folsom, California
Date           1-25-94

CHIP     STATEMACHINE     85C22V10

; inputs
PIN  1         CLK
PIN            ADS_n          ; address status - i960 CA microprocessor
PIN            W_R_n          ; W/R# - i960 CA microprocessor
PIN            BLAST_n        ; burst last - i960 CA microprocessor
PIN            CS_n           ; chip select - 28F016XS
PIN            CFG            ; 28F016XS/i960 CA microprocessor config status set input
PIN            A2             ; LAD bit 2
PIN            A3             ; LAD bit 3
PIN            RESET          ; resets all FFs in device
PIN  25        GLOBAL         ; virtual pin to implement reset

; outputs
PIN            CTR0           ; burst counter out - 28F016XS-A1
PIN            CTR1           ; burst counter out - 28F016XS-A2
PIN            /WE            ; write enable - 28F016XS
PIN            /OE            ; output enable - 28F016XS
PIN            Q0             ; state variables
PIN            Q1
PIN            /ADV           ; state variable and address valid - 28F016XS
PIN            Q3

 ; burst counter control signals
STRING LD '(/ADS_n)'          ; load
STRING INC '(ADV * /Q1 * /Q0
            + Q3 * ADV * /Q1
            + Q3 * /Q1 * /Q0)'  ; increment

STATE
MOORE_MACHINE
DEFAULT_BRANCH S0
```

297500-26

```
; state assignments
            S0 =  /Q3 * /ADV * /Q1 * /Q0
            S1 =  /Q3 *  ADV * /Q1 * /Q0
            S2 =   Q3 *  ADV * /Q1 * /Q0
            S3 =   Q3 *  ADV * /Q1 *  Q0
            S4 =  /Q3 *  ADV * /Q1 *  Q0
            S5 =  /Q3 * /ADV * /Q1 *  Q0
            S6 =  /Q3 * /ADV *  Q1 * /Q0
            S7 =   Q3 * /ADV *  Q1 * /Q0
            S8 =   Q3 * /ADV * /Q1 * /Q0

; state transitions
S0 :=   (/ADS_n * /W_R_n)    -> S1      ; READ cycle
   +    (/ADS_n * W_R_n)     -> S6      ; WRITE cycle
                         +-> S0         ; else, stay
S1 :=   (/CS_n * /CFG)       -> S5      ; 28F016XS selected, initial configurations
   +    (/CS_n * CFG)        -> S2      ; 28F016XS selected, 28F016XS and i960 CA microprocessor configured
                         +-> S0         ; else, return to idle state
S2 :=   VCC                  -> S3
S3 :=   VCC                  -> S4      ; 28F016XS is configured to wait 4 clocks
S4 :=   (/BLAST_n * ADS_n)   -> S0      ; 1 double word  read
   +    (/BLAST_n * /ADS_n)  -> S1      ; pipelined read
                         +-> S5         ; else, continue
S5 :=   (/BLAST_n * ADS_n)   -> S0      ; burst read finished
  +     (/BLAST_n * /ADS_n)  -> S1      ; pipelined read
                         +-> S5         ; else, continue
S6 :=   /CS_n                -> S7      ; 28F016XS selected, continue
                         +-> S0     ; else, return to idle state
S7 :=   VCC                  -> S8
S8 :=   (BLAST_n * CFG)      -> S6      ; continue burst
   +    (BLAST_n * /CFG)     -> S8      ; pre-config write
                         +-> S0         ; write is finished

; transition outputs
S0.OUTF :=     /OE * /WE
S1.OUTF :=     /OE * /WE
S2.OUTF :=      OE * /WE
S3.OUTF :=      OE * /WE
S4.OUTF :=      OE * /WE
S5.OUTF :=      OE * /WE
S6.OUTF :=      OE * /WE
S7.OUTF :=     /OE *  WE
S8.OUTF :=     /OE *  WE
S9.OUTF :=     /OE * /WE

EQUATIONS
  ; implement RESET
      GLOBAL.RSTF = /RESET

  ; implement 2-bit burst counter -  registered counter equations
      CTR1 := (LD * A3) + (/LD * INC * CTR0 * /CTR1)
            + (/LD * INC * /CTR0 * CTR1) + (/LD * /INC * CTR1)
      CTR0 := (LD * A2) + (/LD * INC * /CTR0) + (/LD * /INC * CTR0)

  ; flop OE and WE on falling edge
      OE.CLKF = /CLK
      WE.CLKF = /CLK
```

297500-27

PLD File for the 28F016XS Interface to the i960 JF-33 Microprocessor

```
Title          28F016XS/i960 JF Microprocessor Interface State Machine
Pattern        PDS
Revision       1
Authors        Example
Company        Intel Corporation - Folsom, California
Date           8-16-94

CHIP      STATEMACHINE      85C22V10

; inputs
PIN  1         CLK
PIN            ADS            ; address status - i960 FJ microprocessor
PIN            W_R            ; W/R# - i960 FJ microprocessor
PIN            BLAST          ; burst last - i960 JF microprocessor
PIN            CS             ; chip setect
PIN            CFG            ; 28F016XS/i960 JF microprocessor config status set input
PIN            A2             ; A bit 2
PIN            A3             ; A bit 3
PIN            RESET          ; resets all FFs in device
PIN  25        GLOBAL         ; virtual pin to implement reset

; outputs
PIN            CTR0           ; burst counter out - 28F016XS-A1
PIN            CTR1           ; burst counter out - 28F016XS-A2
PIN            /WE            ; write enable - 28F016XS
PIN            /OE            ; output enable - 28F016XS
PIN            /RDYRCV        ; wait-state control
PIN            Q0             ; state variables
PIN            Q1
PIN            /ADV           ; state variable and address valid - 28F016XS
PIN            Q3

 ; burst counter control signals
STRING LD '(/ADS)'            ; load
STRING INC '(ADV
           + RDYRCV * Q3 * Q1 * Q0)'   ; increment

STATE
MOORE_MACHINE
DEFAULT_BRANCH S0
```

297500-28

```
; state assignments
            S0 = /RDYRCV * /Q3 * /ADV * /Q1 * /Q0
            S1 = /RDYRCV * /Q3 *  ADV * /Q1 * /Q0
            S2 = /RDYRCV * /Q3 *  ADV * /Q1 *  Q0
            S3 = /RDYRCV * /Q3 *  ADV *  Q1 * /Q0
            S4 =  RDYRCV * /Q3 *  ADV *  Q1 *  Q0
            S5 =  RDYRCV * /Q3 * /ADV * /Q1 *  Q0
            S6 = /RDYRCV * /Q3 * /ADV *  Q1 * /Q0
            S7 = /RDYRCV *  Q3 *  ADV * /Q1 * /Q0
            S8 = /RDYRCV * /Q3 * /ADV *  Q1 *  Q0
            S9 = /RDYRCV *  Q3 *  ADV * /Q1 *  Q0
           S10 =  RDYRCV *  Q3 * /ADV * /Q1 * /Q0
           S11 = /RDYRCV *  Q3 *  ADV *  Q1 * /Q0
           S12 = /RDYRCV *  Q3 * /ADV * /Q1 *  Q0
           S13 = /RDYRCV *  Q3 * /ADV *  Q1 * /Q0
           S14 = /RDYRCV *  Q3 * /ADV *  Q1 *  Q0
           S15 =  RDYRCV *  Q3 * /ADV *  Q1 *  Q0

; state transitions
S0 :=   (/ADS * /W_R)          -> S1     ; READ cycle
  +     (/ADS *  W_R)          -> S13    ; WRITE cycle
                            +-> S0       ; else, stay
S1 :=   (/CS * /CFG)           -> S6     ; 28F016XS selected, init configurations
  +     (/CS * CFG)            -> S2     ; 28F016XS selected, optimized configured
                            +-> S0       ; else, return to idle state
S2 :=   VCC                    -> S3
S3 :=   VCC                    -> S4     ; 28F016XS is configured to wait 4 clocks
S4 :=   (/BLAST * ADS)         -> S0     ; 1 double word read
                            +-> S5       ; else, continue
S5 :=   /BLAST                 -> S0     ; burst read finished
  +     (BLAST *  CFG)         -> S5     ; continue, optimized configuration
  +     (BLAST * /CFG)         -> S12    ; continue, initial configuration
S6 :=   VCC                    -> S7
S7 :=   VCC                    -> S8
S8 :=   VCC                    -> S9
S9 :=   VCC                    -> S10
S10:=   /BLAST                 -> S0     ; BLAST -  end of the burst read transaction
  +     BLAST                  -> S11
S11:=   VCC                    -> S5
S12:=   VCC                    -> S5
S13:=   CS                     -> S0     ; write cycle control
  +     /CS                    -> S14
S14:=   VCC                    -> S15
S15:=   BLAST                  -> S13    ; BLAST - end of burst write transaction
  +     /BLAST                 -> S0

; transition outputs
S0.OUTF :=    /OE * /WE
S1.OUTF :=    /OE * /WE
S2.OUTF :=     OE * /WE
S3.OUTF :=     OE * /WE
S4.OUTF :=     OE * /WE
S5.OUTF :=     OE * /WE
S6.OUTF :=    /OE * /WE
S7.OUTF :=    /OE * /WE
S8.OUTF :=     OE * /WE
S9.OUTF :=     OE * /WE
```

297500-29

```
S10.OUTF :=    OE * /WE
S11.OUTF :=    OE * /WE
S12.OUTF :=    OE * /WE
S13.OUTF :=   /OE *  WE
S14.OUTF :=   /OE *  WE
S15.OUTF :=   /OE * /WE

EQUATIONS
  ; implement RESET
      GLOBAL.RSTF = /RESET

  ; implement 2-bit burst counter -  registered counter equations
      CTR1 := (LD * A3) + (/LD * INC * CTR0 * /CTR1)
            + (/LD * INC * /CTR0 * CTR1) + (/LD * /INC * CTR1)
      CTR0 := (LD * A2) + (/LD * INC * /CTR0) + (/LD * /INC * CTR0)

  ; flop OE and WE on falling edge
      OE.CLKF = /CLK
      WE.CLKF = /CLK
```

297500-30

2

PLD File for the 28F016XS Interface to the i960 KB-25 Microprocessor

```
Title          28F016XS/i960 KB Microprocessor Interface State Machine - 25MHz Version
Pattern        PDS
Revision       1
Authors        Example
Company        Intel Corporation - Folsom, California
Date           1-18-94

CHIP     STATEMACHINE     85C22V10

; inputs
PIN   1        CLK
PIN            ADS              ; address status - i960 KB microprocessor
PIN            QA0              ; latched LAD bit 0
PIN            QA1              ; latched LAD bit 1
PIN            W_R              ; W/R# - i960 KB microprocessor
PIN            CS               ; chip select - 28F016XS
PIN            CFG              ; 28F016XS config status input
                                ;  CFG=0 => 28F016XS config=4 (initial config)
                                ;  CFG=1 => 28F016XS config=2
PIN            A2               ; LAD bit 2
PIN            A3               ; LAD bit 3
PIN            RESET            ; resets all FFs in device
PIN   25       GLOBAL           ; virtual pin to implement reset

; outputs
PIN            CTR0             ; burst counter out - 28F016XS-A1
PIN            CTR1             ; burst counter out - 28F016XS-A2
PIN            /OE              ; output enable - 28F016XS
PIN            /WE              ; write enable - 28F016XS
PIN            /READY           ; i960 KB microprocessor
PIN            /ADV             ; state register and address valid - 28F016XS
PIN            Q0               ; state registers
PIN            Q1
PIN            Q2
PIN            Q3

 ; burst counter control signals
STRING LD '(S0 * /ADS)'         ; load
STRING INC '(S1 + S2 + S3
          + S9 + S13 + S16 + S19)'  ; increment

STATE
MOORE_MACHINE
DEFAULT_BRANCH S0

; state assignments
          S0  = /ADV * /Q3 * /Q2 * /Q1 * /Q0
          S1  =  ADV * /Q3 * /Q2 * /Q1 * /Q0
          S2  =  ADV * /Q3 * /Q2 * /Q1 *  Q0
          S3  =  ADV * /Q3 * /Q2 *  Q1 *  Q0
          S4  =  ADV * /Q3 * /Q2 *  Q1 * /Q0
          S5  = /ADV * /Q3 * /Q2 *  Q1 * /Q0
          S6  =  ADV *  Q3 * /Q2 *  Q1 * /Q0
          S7  = /ADV * /Q3 * /Q2 * /Q1 *  Q0
          S8  = /ADV * /Q3 *  Q2 *  Q1 *  Q0
```

297500-31

```
S9  = ADV * /Q3 * Q2 * Q1 * Q0
S10 = ADV * /Q3 * Q2 * Q1 * /Q0
S11 = /ADV * Q3 * /Q2 * /Q1 * /Q0
S12 = /ADV * Q3 * /Q2 * /Q1 * Q0
S13 = /ADV * Q3 * /Q2 * Q1 * Q0
S14 = /ADV * Q3 * /Q2 * Q1 * /Q0
S15 = /ADV * Q3 * Q2 * Q1 * /Q0
S16 = /ADV * Q3 * Q2 * Q1 * Q0
S17 = /ADV * Q3 * Q2 * /Q1 * Q0
S18 = ADV * Q3 * Q2 * /Q1 * Q0
S19 = /ADV * Q3 * Q2 * /Q1 * /Q0
S20 = /ADV * /Q3 * Q2 * /Q1 * /Q0
S21 = ADV * /Q3 * Q2 * /Q1 * /Q0
S22 = /ADV * /Q3 * Q2 * /Q1 * Q0
S23 = /ADV * /Q3 * Q2 * Q1 * /Q0
S24 = /ADV * /Q3 * /Q2 * Q1 * Q0
S25 = ADV * Q3 * Q2 * Q1 * /Q0

; state transitions
S0 :=   (/ADS * /W_R_n)      -> S1     ; READ cycle
   +    (/ADS * W_R_n)       -> S11    ; WRITE cycle
                          +-> S0       ; else, stay
S1 :=   /CS                  -> S2     ; continue if 28F016XS is selected
                          +-> S0       ; else, return to idle state
S2 :=   /CFG                 -> S24    ; 28F016XS is configured to wait 4 clocks
                          +-> S3       ; else, continue
S3 :=   VCC                  -> S4     ; else, continue
S4 :=   (/QA1 * /QA0)        -> S0     ; 1 double word read
                          +-> S5       ; else, continue
S5 :=   (/QA1 * QA0)         -> S0     ; 2 double word burst read
   +    (QA1 * /CFG)         -> S23    ; SFI Configuration = 4
                          +-> S6       ; else, continue
S6 :=   (QA1 * /QA0)         -> S0     ; 3 double word burst read
                          +-> S7       ; else, continue
S7 :=   VCC                  -> S0     ; 4 double word burst read
S8 :=   VCC                  -> S9
S9 :=   VCC                  +-> S5    ; else, continue
S11 :=  /CS                  -> S12    ; continue if 28F016XS is selected
                          +-> S0       ; else, return to idle state
S12 :=  VCC                  -> S13
S13 :=  (/QA1 * /QA0)        -> S0     ; 1 double word write
                          +-> S14      ; else, continue
S14 :=  VCC                  -> S15
S15 :=  VCC                  -> S16
S16 :=  (/QA1 * QA0)         -> S0     ; 2 double word burst write
                          +-> S17      ; else, continue
S17 :=  VCC                  -> S18
S18 :=  VCC                  -> S19
S19 :=  (QA1 * /QA0)         -> S0     ; 3 double word burst write
                          +-> S20      ; else, continue
S20 :=  VCC                  -> S21
S21 :=  VCC                  -> S22
S22 :=  VCC                  -> S0     ; 4 double word burst write
S23 :=  VCC                  -> S25
S24 :=  VCC                  -> S8
S25 :=  VCC                  -> S6
```

297500-32

2

```
; transition outputs
S0.OUTF  :=    /WE * /OE * /READY
S1.OUTF  :=    /WE * /OE * /READY
S2.OUTF  :=    /WE * /OE * /READY
S3.OUTF  :=    /WE *  OE * /READY
S4.OUTF  :=    /WE *  OE *  READY
S5.OUTF  :=    /WE *  OE *  READY
S6.OUTF  :=    /WE *  OE *  READY
S7.OUTF  :=    /WE *  OE *  READY
S8.OUTF  :=    /WE *  OE * /READY
S9.OUTF  :=    /WE *  OE * /READY
S10.OUTF :=    /WE *  OE *  READY
S11.OUTF :=     WE * /OE * /READY
S12.OUTF :=     WE * /OE * /READY
S13.OUTF :=    /WE * /OE *  READY
S14.OUTF :=     WE * /OE * /READY
S15.OUTF :=     WE * /OE * /READY
S16.OUTF :=    /WE * /OE *  READY
S17.OUTF :=     WE * /OE * /READY
S18.OUTF :=     WE * /OE * /READY
S19.OUTF :=    /WE * /OE *  READY
S20.OUTF :=     WE * /OE * /READY
S21.OUTF :=     WE * /OE * /READY
S22.OUTF :=    /WE * /OE *  READY
S23.OUTF :=    /WE *  OE * /READY
S24.OUTF :=    /WE * /OE * /READY
S25.OUTF :=    /WE *  OE * /READY

EQUATIONS
  ; implement RESET
      GLOBAL.RSTF = RESET

  ; implement 2-bit burst counter - registered counter equations
      CTR1 := (LD * A3) + (/LD * INC * CTR0 * /CTR1)
            + (/LD * INC * /CTR0 * CTR1) + (/LD * /INC * CTR1)
      CTR0 := (LD * A2) + (/LD * INC * /CTR0) + (/LD * /INC * CTR0)

  ; flop WE, OE, and READY on falling edge
      WE.CLKF = /CLK
      OE.CLKF = /CLK
      READY.CLKF = /CLK
```

297500-33

PLD File for the 1x CLK Generation and Synchronization

```
Title           1x Clock Generations & Sychronization
Pattern         PDS
Revision        1
Author          Example
Company Name    Intel
Date            5/26/94

CHIP  Clock_Generation_Circuit  85C220

; input pins
PIN             CLK     ; clk frequency 33MHz
PIN             RESET

; output pins
PIN             SYSCLK

EQUATIONS
      SYSCLK := /SYSCLK * /RESET
      SYSCLK.CLKF = CLK
```

297500-34

PLD File for th1 1x and 2x CLK Generation and Synchronization

```
Title           1x and 2x Clock Generation and Synchronization
Pattern         PDS
Revision        1
Author          Example
Company Name    Intel
Date            5/26/94

CHIP  Clock_Generation_Circuit  85C220

; input pins
PIN             CLK             ; clk frequency 4x
PIN             RESET           ; System RESET

; output pins
PIN             CLK2            ; 2x CLK output
PIN             SYSCLK          ; Sychronized 1x CLK output

EQUATIONS
      CLK2 := /CLK2
      CLK2.CLKF = CLK

      SYSCLK :=  CLK2 *  SYSCLK * /RESET
               + /CLK2 * /SYSCLK * /RESET
      SYSCLK.CLKF = CLK
```

297500-35

APPENDIX B
BENCHMARK PERFORMANCE ANALYSIS

The following section provides detailed memory technology information used in the performance analysis (UDP/IP Networking and Imaging Benchmarks) contained in the introduction. The performance analysis was based on actual memory component performance in an i960 processor-based environment. System interface delay between microprocessor and memory was not included in the analysis. The two benchmarks illustrate relative system memory performance.

A. 28F016XS Flash Memory

28F016XS is capable of 3-1-1-1-1-1-1-1 . . . read performance at 5.0V V_{CC} and 33 MHz or 25 MHz (2-0-0-0-0-0-0-0 . . . in terms of wait states). The benchmarking analysis is shown below:

	UDP/IP Networking Benchmark	Imaging Benchmark
	Time (sec)	*Time (sec)*
i960 KB-25 Microprocessor	1.30	1.64
i960 CA-33 Microprocessor	.89	.89
i960 CF-33 Microprocessor	.53	.59

B. 16-Mbit DRAM

16-Mbit DRAMs were, at the time this technical paper was published, only beginning to ramp into production. Only advance information for the wider x16, 16-Mbit DRAMs was available for use in the calculations that follow.

Sequential reads allow use of the DRAM fast page mode. Assumed DRAM specifications are shown below:

- 80 ns t_{RAC}, 40 ns t_{AA} (5.0V V_{CC})
- 256 word (512 byte) page buffer

Therefore, 16-Mbit DRAMs are capable of 3-2-2-2-2-2-2-2 . . . read performance at 33 MHz and 25 MHz (2-1-1-1-1-1-1-1 in terms of wait states . . .). The benchmarking analysis is shown below:

	UDP/IP Networking Benchmark	Imaging Benchmark
	Time (sec)	*Time (sec)*
i960 KB-25 Microprocessor	1.88	1.89
i960 CA-33 Microprocessor	1.06	1.03
i960 CF-33 Microprocessor	.59	.64

C. 4-Mbit EPROM

Calculations that follow used the x16 version of the 4-Mbit EPROM (Intel 27C400 or equivalent).

The assumed 5.0V V_{CC} 4-Mbit EPROM random access time is 150 ns. Therefore, 4-Mbit EPROMs are capable of 5-5-5-5-5-5-5-5 . . . read performance at 5.0V V_{CC} and 33 MHz (4-4-4-4-4-4-4-4 . . . in terms of wait states). The benchmarking analysis is shown below:

	UDP/IP Networking Benchmark	Imaging Benchmark
	Time (sec)	*Time (sec)*
i960 KB-25 Microprocessor	NA	NA
i960 CA-33 Microprocessor	1.56	1.49
i960 CF-33 Microprocessor	.78	.81

2

D. 16-Mbit PAGED MASK ROM

Calculations that follow used the x16 version of the 16-Mbit paged mask ROM, which is not yet widely available from multiple vendors. The x8, 16-Mbit paged mask ROM is the more common version today.

Sequential reads allow use of the mask ROM page mode. The assumed 5.0V V_{CC} 16-Mbit mask ROM random access time is 150 ns, with 75 ns accesses in page mode (4-word page). Therefore, 16-Mbit mask ROMs are capable of 5-3-3-3-5-3-3-3 . . . read performance at 5.0V V_{CC} and 33 MHz (4-2-2-2-4-2-2-2 . . . in terms of wait states). The benchmarking analysis is shown below:

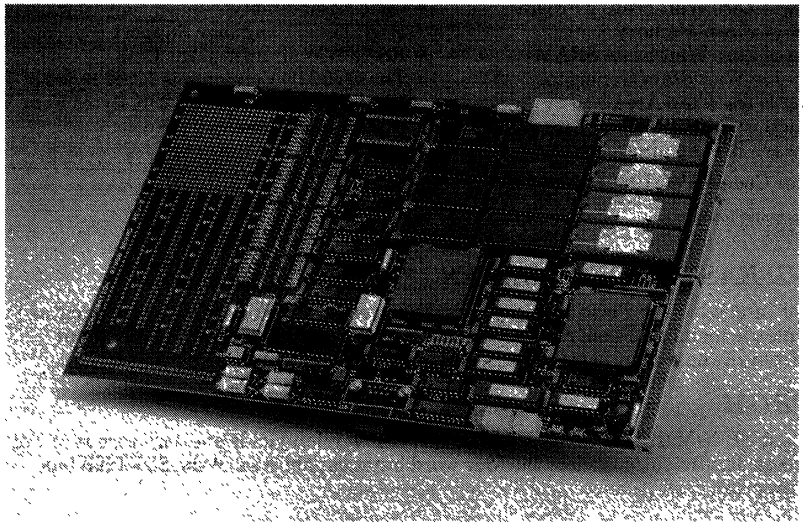
	UDP/IP Networking Benchmark	Imaging Benchmark
	Time (sec)	*Time (sec)*
i960 KB-25 Microprocessor	NA	NA
i960 CA-33 Microprocessor	1.35	1.30
i960 CF-33 Microprocessor	.71	.73

intel®

3

Development Support Tools

QT960 EVALUATION AND PROTOTYPING BOARD



270743-1

3

LOW COST EVALUATION TOOL

The QT960 products give you a 32-bit starter kit to begin software evaluation and hardware design at a low cost. The boards feature the 20 MHz 80960KB 32-bit embedded processor. The 80960KB has integrated floating point, instruction and register caches, and an on-chip interrupt controller. The 80960K-series are the first in a new architectural family of embedded processors from Intel built using Intel's CHMOS IV† process. These boards provide you with full access to the features of the 80960KB processor. A wire wrap prototyping area offers you easy access to board features to test your designs. Interleaved EPROM means fast execution of your code taking advantage of the 80960KB's burst bus. A programmable wait state generator simulates different memory environments useful in evaluating the performance of your code. These features make the QT960 boards useful low cost tools for the 32-bit embedded designer.

Once written, you can debug your program with NINDY, an EPROM resident debug monitor. NINDY enables you to download code, set seven different trace modes, display and modify memory or registers, and disassemble problem code sequences.

Available separately from Intel are the ASM-960 (assembly language) and iC-960 (high-level language) products which provide you with the code development environment for the QT960 boards.

The starter kit comes in two versions: the QT960F version has fast SRAM, high speed EPROM and Flash memory; the QT960E version has lower cost SRAM, Flash memory and no high speed EPROM. Each version has NINDY in either EPROM (QT960F) or Flash memory (QT960E), power supply cable, and the QT960 User Manual. Both versions also include the parts list, source code of the debug monitor, and the board data base (schematics) all on diskette. Armed with this starter kit you now have a system to evaluate and prototype your product ideas quickly and at low cost.

November 1990
Order Number. 270743-002

intel

FEATURES

QT960 FEATURES

- 20 MHz Execution Speed
- 128K Bytes to Zero Wait State EPROM‡
- 128K Bytes of Flash Memory
- 128K Bytes of Zero Wait State SRAM‡
- Programmable Wait State Generator
- Prototyping Wire Wrap Area
- Five Instruction Traces
- Two Hardware Breakpoints
- Display/Modify Memory and Registers
- Code Disassembly
- High Level Language Support
- RS-232 Communications Link
- The QT960E Version has 128K Bytes of Two Wait State SRAM and 128K Bytes Four Wait State Flash Memory

Product Order Codes: EVQT960F20 and EVQT960E20

†CHMOS IV is a patented Intel process.
‡QT960F Version only.

FAST AND EASY CODE UPDATES

128K Bytes of Intel's 28F256 Flash memory provides an easy and quick method of changing your code in nonvolatile memory. Flash memory may be conveniently reprogrammed without removing it from the board while software is under development.

FAST EPROM

Interleaved fast EPROM (Intel's 27C202) on the QT960F version yields one-zero-zero-zero wait state code access. It efficiently utilizes the four word burst capabilities of the 80960KB bus maximizing program performance.

PROTOTYPING SUPPORT

A prototyping wire wrap area is provided on board with access to the system's signals and buses. This area gives you access to the board's features and allows you to easily test design ideas. A system bus connector is also provided for off board prototyping.

PROGRAMMABLE WAIT STATE GENERATOR

A software programmable wait state generator enables you to quickly model various memory speeds. Under software control you can set over 16 different wait state combinations and evaluate the performance of your target system.

DMA

The board offers you eight DMA channels accessed through a NINDY library function using Intel's 82380. In addition, off board connectors provide DMA I/O capabilities.

FIVE INSTRUCTION TRACES AND TWO HARDWARE BREAKPOINTS

NINDY utilizes the built-in trace capabilities of the 80960KB to provide you with single step, supervisor, call, return, and branch instruction tracing offering you extensive debug capabilities for software examination and modification. Two hardware breakpoints enable you to break on and examine EPROM resident code.

FEATURES

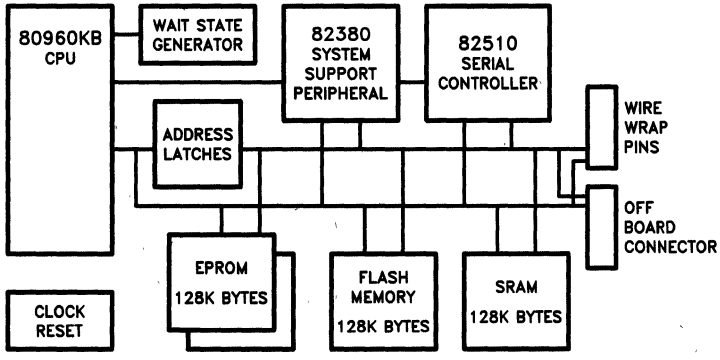
HIGH LEVEL LANGUAGE SUPPORT

NINDY is capable of downloading absolute object code generated by ASM-960 or iC-960. ASM-960 and iC-960 may be purchased separately from Intel.

COMMUNICATION AND SOFTWARE REQUIREMENTS

The QT960 boards communicate with the host through the RS-232 link using an Intel 82510 UART provided on board. The boards support five baud rates: 1200, 2400, 9600, 19200, and 38400. The default is 9600 baud. To communicate with the QT960 boards you must meet the following minimum software requirements:

- Terminal Emulator
- XMODEM Download Capabilities

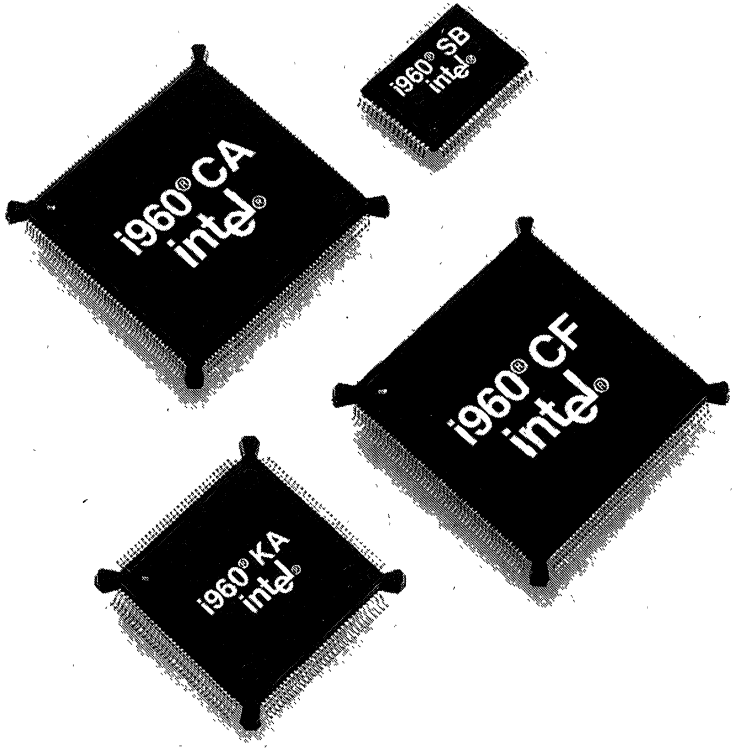


270743-2

Block Diagram of the QT960 Board

For information or the number of your nearest sales office call 800-548-4752 (U.S. and Canada).
Intel Corporaton, Literature Department, 3065 Bowers Avenue, Santa Clara CA 95051, United States. Tel: 408-987-8080.

C PROGRAMMING TOOLS FOR THE i960® MICROPROCESSOR FAMILY



281434-1

PRODUCT OVERVIEW

In recent years, embedded designs have grown in sophistication, encompassing many years of development and thousands of lines of code. Due to the size, complexity and constantly evolving desire to stay competitive, designers have turned to high-level languages for productivity and RISC-based processors for performance.

Advanced processor technology such as RISC and superscalar has promised enhanced performance through the use of pipelining, multiple execution units, and generally, more efficient code execution. Achievement of this performance, however, has been constrained by the lack of compiler technology to take advantage of it.

Intel's continuous investment in compiler technology allows user applications to deliver maximum performance from the i960 processor family. CTOOLS960 exploits the i960 processor features and implements the most advanced optimization technique—profile driven optimizations, where code optimizations are based on application runtime behavior, thus achieving superior performance compared with code generated by conventional global optimizations.

November 1993
Order Number: 281434-002

C PROGRAMMING TOOLS FOR THE i960® MICROPROCESSOR FAMILY

PRODUCT HIGHLIGHTS

- CTOOLS960 consists of a high-performance profiling compiler, profiling tools, an assembler/linker and utility tools
- Fully conforms to ANSI standard, passes Plum Hall* and Perennial* tests
- Produces superior quality, highly optimized code for i960 microprocessor family
- Tightly integrated with DB960 source-level debugger and in-circuit emulators
- Operates on a variety of host operating systems
- Supports in-line assembly code in C source
- Big-endian support (i960 CA/i960 CF processors)
- Faster compile time and smaller code with Revision 4.0

PROFILE-DRIVEN COMPILER OPTIMIZATIONS

Profile-driven compiler optimization is a technique wherein the compiler reads a runtime profile of the source code being compiled, and makes code optimization decisions based on that profile. The runtime profile is gathered by instrumentation code inserted by the compiler during a preliminary compilation. The instrumented program is executed using typical data and the resultant program profile is saved. In a subsequent compilation, the runtime profile is correlated to the source code and an optimized program generated.

Code optimizations based on a program profile are more effective than traditional global optimizations. Conventional optimization strategies are limited by static knowledge of the program: the text of the program is extrapolated to its runtime behavior. Since the compiler has no idea which parts of the program influence execution time, all parts of the program receive equal attention.

In contrast, profile-driven optimizations are based on known runtime characteristics of the program. They go beyond traditional local and global optimizations, to become application-specific. The profile information is used by the compiler in a number of ways:

- Additional resources (e.g., registers) can be allocated to parts of the program that execute more often.
- Loops may be transformed to separate frequently executed paths from rarely executed ones, reducing interference between code segments and opening up more code motion opportunities. (Superblock Formation)
- Frequently taken paths through the program code may be placed in sequence, increasing instruction cache hit rates. (Basic Block Rearrangement)
- Program-level function inlining reduces call overhead and creates more optimization opportunities. (Inter-module Inlining)
- Heavily used global variables may be placed in faster memory or in on-chip data RAM.

All of these optimizations have been implemented in the Intel CTOOLS960 compiler tool set.

i960® ARCHITECTURE-SPECIFIC OPTIMIZATIONS

These optimizations are implemented to ensure the application can benefit from all features of the architecture.

- **Specialized instruction selection.** The compiler makes use of the i960 architecture's specialized instructions such as clrbit, setbit, nand, and ornot instructions.
- **Intelligent register manager.** It performs the assignment of all register operands to the available general purpose and floating point registers.
- **Code scheduling.** The compiler modifies the ordering of instructions to increase the amount of parallel execution available on the specific i960 processor.
- **Use of on-chip data RAM for spill registers.** When the i960 processor's physical registers are insufficient to meet the demands of a function, some registers must be "spilled" to some slower storage. For the i960 CA/i960 CF processor, you may "spill" into the on-chip data RAM instead of the slower off-chip memory.
- **Complex addressing modes (reducing instruction fetches and code space).** By default, the compiler uses complex addressing modes that generate denser code,

generating fewer instructions and fetches from memory. Users can instruct the compiler to retain the RISC form of addressing when code compaction is not desirable.

- **Branch prediction.** The compiler uses profile data to set the branch prediction bit for the i960 CA/i960 CF processors to maximize the chances of a correct prediction.
- **Identification of leaf procedures.** The iC960 compiler identifies procedures that contain no further calls. The linker optimizes the call to such leaf procedures to use the branch-and-link mechanism that does not allocate a new stack frame or register frame and executes faster than the call mechanism.

ARCHITECTURE INDEPENDENT TRADITIONAL OPTIMIZATIONS

The i960 processor compiler performs the following local and global optimizations.

- **Constant expression evaluation.** The iC960 compiler directly evaluates simple arithmetic expressions containing constants and tracks constant values through all the computations performed.
- **Collapsing of arithmetic and bitwise Boolean identities.** The compiler recognizes instances of arithmetic and bitwise Boolean operations in which one of the operands is an identity constant. The unnecessary operation is eliminated.
- **Common subexpression elimination.** The compiler detects multiple occurrences of the same expression and avoids redundant computations by using the result left in the register.
- **Register subsumption or register coalescing.** The compiler coalesces multiple registers containing the same value, thus eliminating a large number of register move instructions.
- **Local variable promotions.** The compiler promotes to a register location, a variable of automatic or register storage class.
- **Tail-call elimination.** When the last statement in a function is a call, execution time can be saved by replacing the call with an unconditional branch.
- **Automatic procedure inlining.** The compiler automatically selects functions for inlining. Pragmas are also available for user control over inlining.
- **Branch optimizations.** The compiler rearranges branch instructions to minimize the number of branches executed.
- **Dead code elimination.** The compiler eliminates code that computes values never used and code that cannot be reached.
- **Loop invariant code motion.** The compiler identifies computations that do not change within the body of a loop and moves those computations to a point before the loop entry.
- **Induction variable elimination.** FOR loop array subscripts can be simplified by the compiler thus minimizing address calculation overhead.

ASSEMBLER AND LINKER

The ASM960 assembler tool is based on the Free Software Foundation Tools (Note: Applications code generated by ASM960 is not subject to copyleft. Copyleft applies to application code incorporating ASM960 source code.). The assembler processes assembly code produced by the compiler. The i960 processor linker links together separately compiled modules, performing additional optimizations such as replacing calls by branch-and-link sequences.

The ASM960 toolset offers other useful utilities such as:

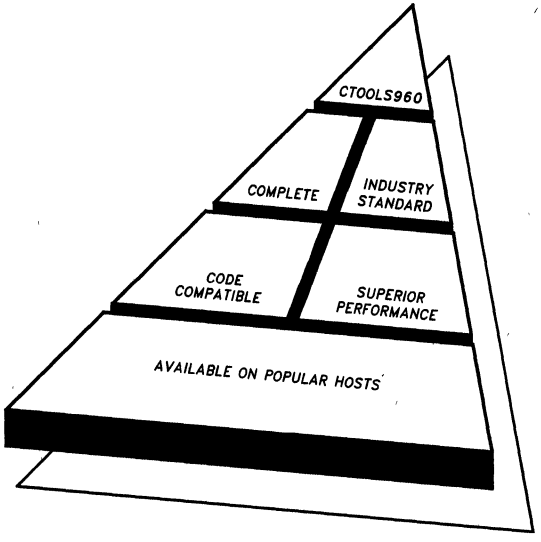
- **Debugging aids:** COFF dumper and mapper.
- **An archiver** to build libraries in COFL format.
- **A COFF stripper** to eliminate debug records from the object module.
- **A big-endian to little-endian converter** (on big endian hosts).
- **A ROM builder** to produce ROMable code.
- **A COFF to IEEE-695 converter.**

LIBRARY SUPPORT

There are three types of libraries supported by CTOOLS960:

- **High level libraries**—contain over 200 ANSI conforming and ANSI superset functions.

C PROGRAMMING TOOLS FOR THE i960® MICROPROCESSOR FAMILY



- **Floating-point support**—the Accelerated Floating Point library provides highly optimized, basic floating-point operations for i960 architectures without a floating-point unit (KA, SA, CA, CF).
- **Low-level libraries**—provide low-level support for execution on Intel-supported i960 processor execution vehicles.

WORLDWIDE SERVICE AND SUPPORT

To augment its development tools, Intel offers field application engineering expertise and hotline technical support.

Intel also offers software support which includes technical software information, automatic distributions of software and documentation updates, *iCOMMENTS* publication, remote diagnostic software, and a development tools troubleshooting guide.

Intel Development Tools also offers a 30-day, money-back guarantee to customers who are not satisfied after purchasing any Intel development tool.

ORDERING INFORMATION

CTOOLS960 includes C compiler, assembler (ASM960), linker, utilities, C and floating point libraries. Note ASM960 source code is also included.

Code	Description
CTOOLS960D	Intel386™ or Intel486™ CPU DOS-based i960 processor C programming tools.
CTOOLS960H	HP9000/300 or 700, HP-UX based i960 processor C programming tools.
CTOOLS960R	IBM RS6000/AIX-based i960 processor C programming tools.
CTOOLS960W	Sun-4 based i960 processor programming tools.

For more product information call 1-800-628-8686 (U.S. and Canada).

For literature call 1-800-548-4725 (U.S. and Canada)

GNU/960 Software Toolset

Intel's GNU/960 cross-development toolset offers a complete C language development environment for the entire family of i960® RISC microprocessors. GNU/960 is available in binary form for ten UNIX* hosts and DOS/X. Source code is provided for all tools, allowing users to easily port the toolset to a wide variety of other host environments. The GNU/960 tools include gcc960, a profile-driven optimizing compiler designed to take full advantage of the i960 processor's advanced RISC features, as well as an assembler, a linker, a symbolic debugger and many other tools to improve developer productivity.

The GNU/960 tools support both COFF (Common Object File Format) and b.out (GNU/960 specific) object file formats. This allows users to mix and match tools with Intel's CTOOLS960, protecting their software and hardware investment. GNU/960 is based on the Free Software Foundation tools and has been optimized by Intel for the i960 architecture. GNU/960 includes run-time libraries; software developed using the GNU960 tools is fully copyrightable. GNU/960 offers users ongoing maintenance, with frequent releases and two software support options.

The GNU/960 Tools: gcc960

GNU/960 features gcc960, an ANSI-Compliant C compiler. It emits standard Intel assembly language for assembly by either Intel's asm960 or by the GNU/960 toolset's gas960 assembler. gcc960 contains many sophisticated optimizations, including a number of Intel additions to the standard GNU C compiler.

Profile-Driven Optimization

The best optimization decisions require execution information about the whole program. To obtain program-wide information, gcc960 employs a profile-driven compilation model. Several of gcc960's optimization techniques are profile-driven:

- global function inlining
- allocating global variables to on-chip SRAM
- refining alias information for global variables

Other gcc960 optimizations include:

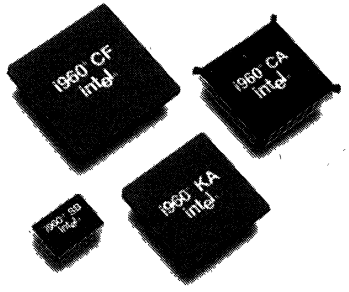
- shadowing memory with registers
- i960 processor instruction scheduling
- constant propagation
- adjacent memory access coalescing
- supports S, K and C-series processors

gss960, gld960, gar960

The GNU/960 assembler, linker, librarian and other tools form a complete software development toolset, allowing users to get the most out of the powerful gcc960 compiler. The GNU/960 tools are designed to be interoperable with Intel's CTOOLS960. This lets users choose the best combination of tools for their individual i960 processor development environment.

gdb960

The GNU/960 debugger is a fully symbolic debugger, and operates with the MON960 monitor.



September 1994
Order Number: 272178-002

Debug Monitor

MON960 monitor is the new debug monitor for the i960 processor family. It replaces the NINDY monitor. New features are: interrupt-driven capability, big Endian and flash support, and a well defined, host debugger interface. It is shipped complete with source for easy retargeting, can be integrated as part of a customer's application, and includes ROM-able hex files for all the standard Intel i960 processor evaluation boards.

Hosts Supported

Sun 3
Sun 4
Sun 386i
i386/i486™ CPU-based UNIX System V R3.2
i386/i486 CPU-based DOS/X 5.0 PC
HP9000/300/700
IBM RS/6000
VAX/ULTRIX
DECStation 3100
Apollo 400

Software Support

Intel offers two different annual maintenance contracts for GNU/960:

Full Supported Contract (GNU960 SSC):

- Unlimited 1-800 technical hotline assistance
- Guaranteed 48 hour bug fix or workaround to all critical problems once they can be duplicated at Intel's development facilities
- Free release upgrades
- Inter-release fixes

Software Assistance Contract (GNU960 SAC):

- Unlimited 1-800 technical hotline assistance
- Free release upgrades

Ordering Information

ORDER CODE	DESCRIPTION
GNU960T	The complete GNU960 toolset, including source, on a QIC-24 format UNIX tar cartridge tape.
GNU960H	The complete GNU/960 toolset, including source, on a HP9000 16-track UNIX tar format cartridge tape.
GNU960M	The complete GNU/960 toolset for DOS/X 5.0 hosts, on 3.5" and 5.25" floppies.**
GNU960DT	The complete GNU/960 toolset, including source, for DOS/X 5.0 hosts, on a QIC-24 format Sytos Plus cartridge tape.
GNU960SSC	Annual full software support contract for GNU/960.
GNU960SAC	Annual software assistance contract for GNU/960.

Contact your nearest Intel Sales Office for information about on-site application development or consulting services or call 1-800-628-8686 (U.S. and Canada) for more product information.
For more literature call 1-800-548-4725 (U.S. and Canada).

United States
Intel Corporation
2200 Mission College Blvd.
P.O. Box 58119
Santa Clara, CA 95052-8119

Japan
Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26

France
Intel Corporation S.A.R.L.
1, Rue Edison, BP 303
78054 Saint-Quentin-en-Yvelines Cedex

United Kingdom
Intel Corporation (U.K.) Ltd.
Pipers Way
Swindon
Wiltshire, England SN3 1RJ

Germany
Intel GmbH
Dornacher Strasse 1
85622 Feldkirchen/Muenchen

Hong Kong
Intel Semiconductor Ltd.
32/F Two Pacific Place
88 Queensway
Central

Canada
Intel Semiconductor of Canada, Ltd.
190 Attwell Drive, Suite 500
Rexdale, Ontario M9W 6H8

*Other brands and names are property of their respective owners.

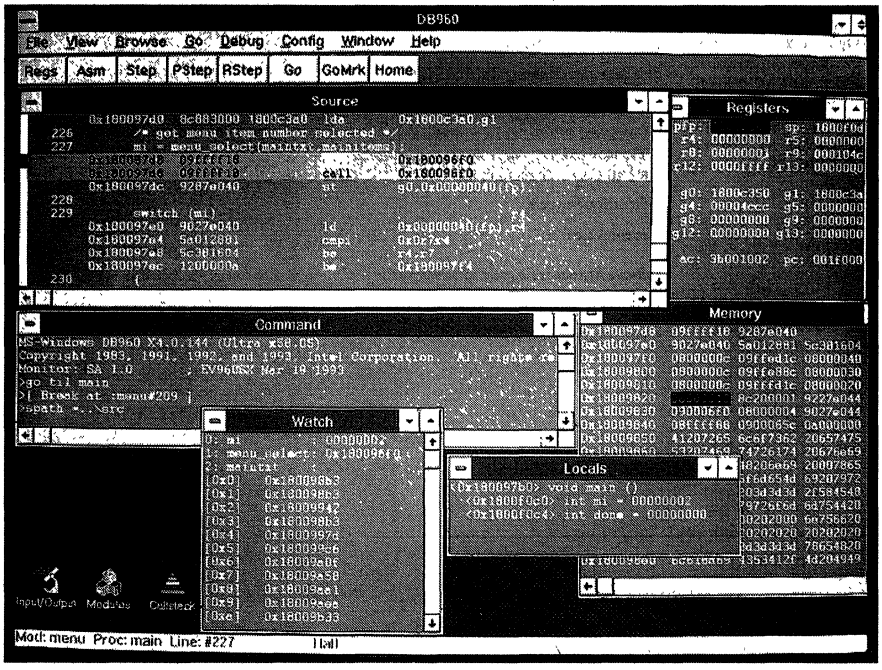
** Source for DOS/X 5.0 release is available only on the cartridge tape distribution, GNU960DT.

Order Number: [illegible]

intel.

3

DB960 SOURCE-LEVEL RETARGETABLE DEBUGGER



281433-1

PRODUCT OVERVIEW

DB960 is a source-level symbolic debugger supporting all members of the i960® microprocessor family. It allows the users to debug application code in their own targets. DB960D's new user interface is based on the Microsoft Windows 3.1 environment providing greater debug productivity. DB960 shares the same debugger interface as Release 4.0 of Intel's i960 KB/SB processor emulators so users can easily transition debug work between the different tools. It operates on a DOS-based Microsoft Windows* 3.1 environment, communicating with the user's target system via a serial port. A retargetable monitor, MON960, is included with DB960D. The complete monitor source, example code and documentation are shipped so users can easily customize the monitor for the target system. There is no incorporation or royalty fee incurred if the monitor is shipped with the customers' products.

FEATURES

- Supports CTOOL960 R4.0 and GNU960 R2.2 compiler development environment
- Microsoft Windows 3.1 human interface with all the convenience of the windows environment, plus source display window, register window, watch window, memory window and local variable window to boost debugging productivity
- Extensive breakpoint modes including source breakpoints, passpoints, temporary breakpoints and event-action breakpoints which trigger actions after the breakpoint is reached
- Breakpoints can be defined interactively, in the source window or symbolically using module names, procedure names and line numbers

*Windows is a trademark of Microsoft Corporation.

August 1993
Order Number: 281433-002

DB960 Source-Level Retargetable Debugger

- Single step program executions based on an assembler instruction, a C language statement or function
- Memory and registers can be displayed and modified
- Watch expressions can be defined and observed in the watch window as the program executes
- Low level run-time library allows programs to access the host file system or to perform I/O operations
- Symbols are demand loaded
- DB960 R4.0 comes with a retargetable monitor (MON960)

DEBUGGING FEATURES

High-level source or disassembled code can be displayed in the source window. Users can scroll through the source, browse from module to module in a program, scope to any executable point in the source, or instantaneously relocate from a symbol name to the location where it was defined (hyperscope operation). Symbol names in the source can be highlighted to inspect the current run-time value of program variables. Call stacks can be examined to trace execution flow.

A variety of breakpoints can be specified including source breakpoints, watch points, passpoints, or event-action breakpoints. Breakpoints can be defined symbolically using module names, procedure names and line numbers. Watch points allow users to observe a variable as it changes during program execution. Passpoints display a message when a specified instruction is executed, giving the user a non-realtime way to track execution of key code sequences without halting instruction flow. The event-action form allows complex breakpoint conditions to be set up, including data breakpoints (when supported by on-chip registers).

Users can step through program execution via a single assembly language instruction, a high-level language statement or a high-level function, function return statement or branch taken. Memory can be displayed or modified as common data types and all processor registers and system tables can be examined or changed. Expressions involving symbol names, memory references, or both, can be defined as watch expressions whose values are monitored in a Watch window as a program executes.

RETARGETABLE MONITOR

A retargetable monitor, MON960, is shipped with DB960 for users to customize and incorporate into their target systems. Complete source code is provided with comprehensive retargeting instructions. Example code for porting to Intel evaluation boards, Intel 82510 UART serial controller chip and the 82C54 counter/timer are provided. MON960 has a well defined host debugger interface allowing users to develop a custom host debugger for their product. It supports big Endian, flash and interrupt-driven capabilities. MON960 also supports GDB960 R2.2 (debugger shipped with GNU960) and is available as a stand-alone product.

HARDWARE DEBUG

DB960 takes advantage of on-chip debug registers like those found on the i960 CA processor to provide hardware execution address and data address breakpoints. By using the available hardware breakpoint registers, the debugger can be used to find difficult bugs like stack overruns and invalid pointer accesses. Once the monitor has been retargeted to the target system, hardware designers can download initialization code, read/write to registers, and examine memory or register contents.

HIGH-SPEED SERIAL LINK

Communications between the debugger host and target system is supported via RS-232 and RS-422 communication links. RS-232 allows access to industry standard serial protocols while the RS-422 interface provides higher speed communication (up to 115K baud) for faster code and data download. PC-AT bus-compatible RS-422 communication boards are available from various third party vendors.

WORLDWIDE SERVICE AND SUPPORT

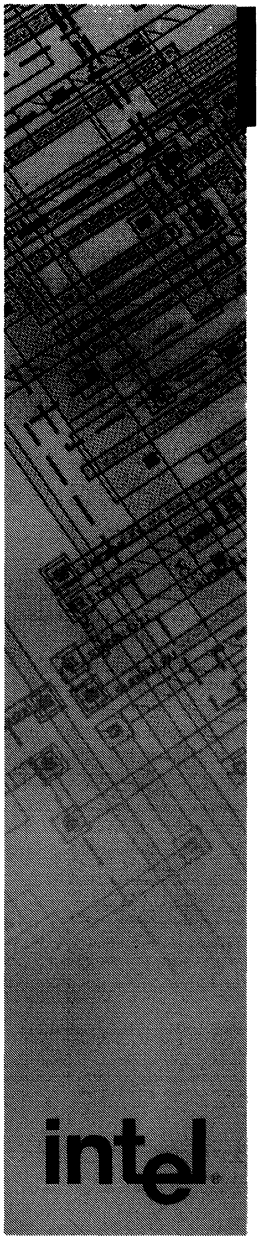
Intel offers software support which includes technical software information, a technical hotline, automatic distributions of software and documentation updates, iCOMMENTS publication, remote diagnostic software, and a development tools troubleshooting guide. Intel Development Tools also offers a 30-day, money-back guarantee to customers who are not satisfied after purchasing any Intel development tool.

ORDERING INFORMATION

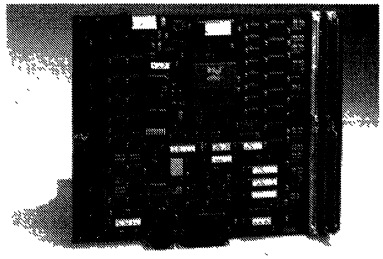
Order Code	Description
DB960D	DOS-based, retargetable software debugger for the i960 KA, i960 KB, i960 SA, i960 SB, i960 CA and i960 CF microprocessors. Includes host debug software, MON960 retargetable monitor, host I/O libraries and documentation.
MON960	System debug monitor, source code product.

For more product information call 1-800-628-8686 (U.S. and Canada).

For more literature call 1-800-548-4725 (U.S. and Canada).



EP80960Cx Evaluation Platform



Low-Cost Processor Evaluation Tool

Intel's EP80960Cx Evaluation Platform provides a low-cost hardware environment for code execution and software debug. The board features the i960® CA processor and is easily upgradeable to the i960 CF processor: the newest and highest performance member of Intel's family of 32-bit embedded microprocessors.

The EP80960Cx includes a high-performance interleaved DRAM subsystem, operating at 2-0-0-0 wait state burst reads and zero wait state (0-0-0-0) posted write. The 2 Mbyte subsystem, expandable up to 16 MBytes, employs standard 70 ns DRAM SIMMs and runs at frequencies up to 33 MHz. The EP80960Cx also features an I/O subsystem and an advanced set of peripheral devices for benchmarking and debugging application code written for the i960 CA/CF embedded processors. The EP80960Cx expansion bus (X-Bus) supports expansion cards and external devices; the X-Bus provides direct access to the i960 CA/CF processor's bus and control signals.

Popular features such a single line assembler/disassembler, single-step program execution, and software breakpoints are standard on the EP80960Cx's on-board monitor. Available separately, Intel offers a complete code development environment with tools such as Intel's iC-960 or GNU/960 compiler.

The EP80960Cx Evaluation Platform package features MON960 Monitor in EPROM, a power supply, a 9-pin PC /AT serial (RS-232) connector, a 25-pin parallel cable for parallel download, the EP80960Cx User's Manual, and diskettes containing MON960 host software, design and library files and an example program. The EP80960Cx User' Manual includes board schematics, a parts list, programmable logic (PLD) equations, and step-by-step instructions on how to compile, assemble, link, download, and execute the example program.

EP80960Cx Features

- 33MHz Execution Speed
- 32 Kbytes of EPROM for 80960CA MON960 Target Operating Firmware
- 2 Mbytes of Fast Page Mode DRAM* expandable to 16 Mbytes
- Concurrent Interrogation of Memory and Registers
- 384 Software Breakpoints
- Code Disassembly
- High Level Language Support
- One RS-232 port for Host and User Communication up to 115.2 Kbaud
- One Centronics Parallel Port for Parallel Downloads
- An Expansion Bus (X-Bus) for Peripheral-to-CPU Interface
- Memory Wait State Control for Memory Subsystem Simulation

* The DRAM subsystem provides zero wait-state posted writes (0-0-0-0) and (2-0-0-0) wait-state read.

Fast Page-Mode DRAM SIMM Modules

The EP80960Cx Evaluation Platform's memory design takes advantage of the i960 CA/CF processor's burst mode bus for interfacing with fast page-mode DRAM. The high-performance DRAM design utilizes three techniques: bank interleaving, write posting and RAS interleaving.

Concurrent Interrogation of Memory and Registers

The on-board MON960 Monitor allows the user to read and modify internal registers and external memory while running the user's program.

September 1994
Order Number: 272505-001

High Speed Downloads

The parallel port on the EP80960Cx is a full implementation of a Centronics-compatible receive only port. This port allows fast downloads of code or data to the EP80960Cx. Intel's software tools support parallel downloads.

Prototype and Expansion Bus

The X-Bus allows the user to quickly prototype simple applications. Two rows of header pins on the EP80960Cx provide buffered versions of address, data and control signals as well as other signals which ease the task of interfacing simple peripherals and I/O devices. The header pins also double as logic analyzer access points.

Communication Link

The EP80960Cx board communicates with the host computer via the RS-232 link. This platform supports baud rates from 300 to 115200 baud.

Power Requirements

The EP80960Cx Evaluation Platform requires 5 volts at 2500 mA, which the included power supply provides. The EP80960Cx also supports a power supply that provides +5VDC / +12VDC for FLASH programming.

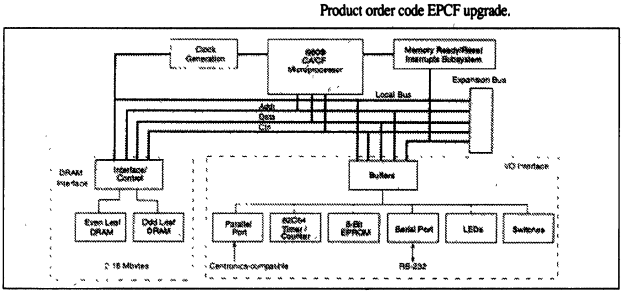
Host System Requirements

The EP80960Cx Evaluation Platform interfaces to either a UNIX- or DOS-based system. An Intel486-based PC is recommended. A DOS-based host system must meet the following minimum requirements:

- PC-DOS 3.2 or Later
- 512 Kbytes of Memory
- One 1.4 Mbyte Floppy Disk Drive
- Serial Port (COM1 or COM2)
- Parallel Port (LPT1) (parallel port is optional)

CF Upgrade

Available for the EP80960CX is a kit designed to quickly and easily upgrade your platform to take advantage of the 80960CF microprocessor performance. This kit supplies the 80960CF component and all documentation needed to make the upgrade. Product order code EPCF upgrade.



EP80960Cx Functional (Block) Diagram

Support Information

Product and Sales Information	Intel product information, sales office and distributor phone numbers.	**800-628-8686**
Literature Department	To order Intel product literature.	**800-548-4725**
Development Tools Hotline	Technical applications support for Intel development tools.	**800-843-4481**
FaxBack Service	Automated response system that faxes Intel documents to your fax machine at no cost. For a listing of all i960 microprocessor documents, order FaxBack service document 2068.	**800-628-2283**
Applications Bulletin Board Service	24-hour access via modem to publicly available technical information. For a complete listing of all files available on BBS, call the FaxBack service and request catalog 6 (BBS Master File Listing).	**916-356-3600**

Phone numbers listed above are for U.S. and Canada only.

UNITED STATES
Intel Corporation
2200 Mission College Boulevard
P.O. Box 58119
Santa Clara, CA 95052-8119

JAPAN
Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26

FRANCE
Intel Corporation S.A.R.L.
1, Rue Edison, BP 303
78054 Saint-Quentin-en-Yvelines
Cedex

UNITED KINGDOM
Intel Corporation (U.K.) Ltd.
Pipers Way
Swindon
Wiltshire, England SN3 1RJ

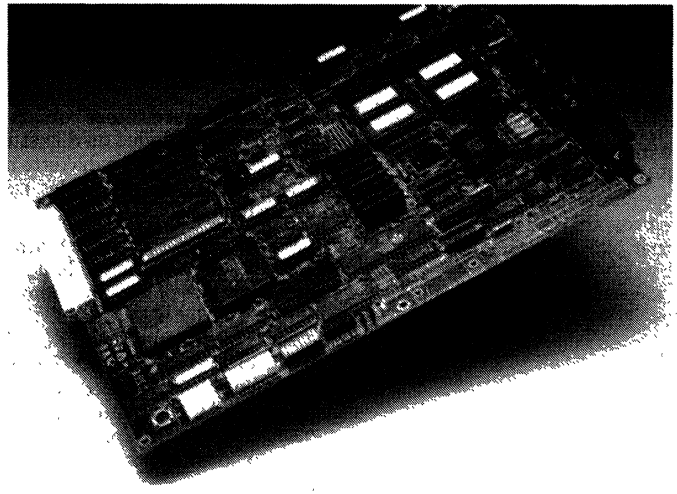
GERMANY
Intel GmbH
Dornacher Strasse 1
85622 Feldkirchen/ Muenchen

HONG KONG
Intel Semiconductor Ltd.
32/F Two Pacific Place
88 Queensway
Central

CANADA
Intel Semiconductor of
Canada, Ltd.
190 Attwell Drive, Suite 500
Rexdale, Ontario M9W 6H8

March 1994
Order Number 272505-001
Printed in U.S.A. [illegible]
Printed on Recycled Paper

i960 SA/SB EVALUATION BOARD



272033-1

3

i960 SA/SB EVALUATION BOARD

The EV80960SX board is a general purpose evaluation tool for the i960 SA/SB embedded processors. This evaluation board provides a high-performance DRAM subsystem, an interleaved EPROM subsystem, and a robust set of peripheral devices for benchmarking and debugging application code written for the i960 SA/SB embedded processors.

The EV80960SX is a great starter kit for your 32-bit application. The EV80960SX, NINDY debug environment, along with assembler and C-compiler (not provided) provide a seamless environment for developing code and evaluating the i960 SA/SB processors. The NINDY monitor provides code download capabilities from a number of popular development systems, including DOS-based PC's. Single step, breakpoints, register and memory display are among the full set of features provided by NINDY.

The board is provided with the following features:

- DRAM Subsystem operates at 1-0-0-0-0-0-0-0 wait states for read and write cycles in the burst mode. The DRAM subsystem runs at the maximum processor frequency of 16 MHz, using 100 ns fast page mode DRAMs. The DRAM subsystem can accommodate from 512 Kbytes to 4 Mbytes, using 4 or 8 ZIP-packaged DRAMs.
- Interleaved EPROM Subsystem executes burst program fetches with a 2-0-1-0-2-0-1-0 wait state performance. The EPROM subsystem accommodates four, 32-pin or 28-pin 8-bit wide EPROMs with up to 150 ns access times.
- Flash EPROM Subsystem reads and writes two 8-bit wide Flash EPROMS.
- 8259A Interrupt Controller provides expanded interrupt capabilities using the i960 SA/SB's interrupt controller interface.
- Parallel Port Input allows fast downloads of code or data to the EV80960SX board. The parallel port provides auto-busy and interrupt capabilities, and is a full implementation of the Centronics standard.

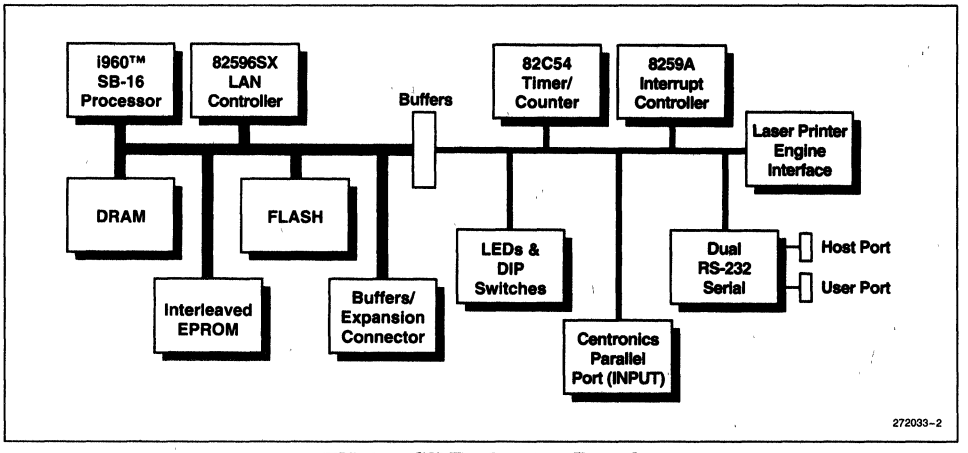
June 1991
Order Number: 272033-001

i960 SA/SB EVALUATION BOARD

- Two serial ports provide queued and interrupt driven serial transfer at up to 128000 baud.
- 82C54 Timer/Counter provides a 32-bit counter and 16-bit counter, each with dedicated interrupts.
- Expansion/Prototype Bus (XBUS) allows expansion cards and prototype hardware direct access to the i960 SA/SB's bus and control signals. Optionally, a configurable wait state scheme provides a no glue interface to most peripherals attached to the XBUS.
- LEDs and Switches are user programmable. One 10-segment bar LED, a 7-segment LED and an 8-position switch are under program control.
- Local Area Networking (LAN) is implemented using an 82596SX LAN coprocessor.
- Laser Printer Control provides interfaces to TEC or Canon compatible laser engines.
- Monitor and Self-test diagnostics are provided for the EV80960SX in the EPROMs installed in the board.

The evaluation board comes complete with a design database included on diskette, the NINDY debug monitor on diskette and in EPROM, power and serial cables, schematics and user's manual.

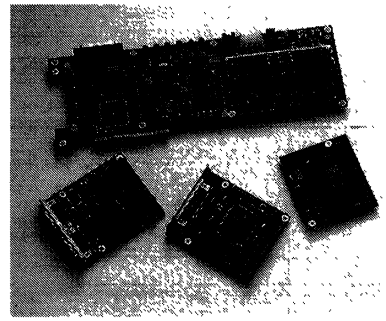
The EV80960SX is a public domain design. The hardware is fully documented and provides working examples of popular memory and peripheral interfaces to the i960 SA/SB processor. The schematic and PLD database are provided with each board. The EV80960SX designs are easily duplicated and can be used directly as the building blocks for custom designs. Custom hardware can be prototyped using the expansion bus (XBUS) connector.



EV80960SX Evaluation Board

i960® Microprocessor Evaluation Platform

Cyclone EP



By tailoring these low cost modules to be interchangeable, designers are freed from the burdensome and time consuming task of supporting multiple boards during an architectural evaluation.

Flexibility Plus

Intel's Cyclone Evaluation Platform is designed to provide the developer with a flexible, low cost environment for code execution and software debug. Configured as a stand-alone base board with interchangeable modules, the Cyclone EP is capable of supporting the entire i960® processor family: from the low cost i960 SA processor to the newest and highest performance member of the i960 family, the i960 HT processor.

The standard interface incorporated on the Cyclone EP allows a designer to interchange application specific modules ranging from Ethernet to SCSI-3. Published specifications for the interface allow designers to build their own module or have Cyclone Microsystems custom develop it for them.

Processor Modules

Designed for the Cyclone EP are CPU modules featuring a specific i960 processor. By tailoring these four low cost modules to be interchangeable, the designer is freed from the burdensome and time consuming task of supporting multiple boards during an architectural evaluation. The CPU modules each contain a boot flash ROM with the MON960 monitor, appropriate glue logic and configuration switches. Using these switches the designer has the ability to further tune the design.

DRAM Memory

The Cyclone EP is shipped with 2 Mbytes of interleaved DRAM memory upgradable to 8 or 32 Mbytes. The DRAM controller automatically adjusts the wait-states based on processor type, clock frequency, and memory speed. The controller supports burst transfers using the interleaved banks to maximize performance. The default memory configuration comes standard with 70ns memory. For processor frequencies of 25 and 40 MHz, performance can be significantly increased by using 60ns memory. The board requires no shunts or switches to adjust for memory changes. The Presence Detect Signals and the controller automatically adjusts to minimum wait-states for the processor frequency and memory speeds. In addition the initialization code automatically sizes the memory so user software can take advantage of larger memory modules.

Software Development Support

The Cyclone EP supports many software development tools. The installation instructions presented in the manual were verified using Intel's GNU/960 and IC960. These advanced C-language compilers for the i960 processor family are available for DOS-based systems and a variety of UNIX workstation hosts. Both products provide execution profiling and instruction scheduling optimizations for tuned code generation. In addition to the Intel tools, the Cyclone EP has been validated for use with many of the Solutions960® program development tools.

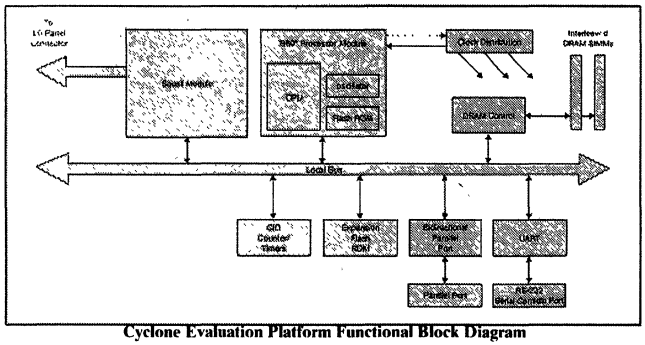
Console Serial Port/Parallel Port

The Cyclone EP has a single console port with an RS-232 line interface as well as a Centronix PC compatible input parallel port. Cables are supplied for both communication ports.

Cyclone Evaluation Platform Features

- Interchangeable i960 processor modules
- DIP switch selectable processor clock frequency
- 2 Mbytes of DRAM, expandable to 32 Mbytes
- DRAM controller automatically optimizes wait- states to processor frequency and memory
- Flash ROM sockets
- Parallel download port
- Three sixteen bit counter/timers
- Squall II module I/O expansion interface

October 1994
Order Number: 272598-001



Cyclone Evaluation Platform Functional Block Diagram

Squall II Module Interface

The Cyclone EP has a single expansion location (Squall II) and connector. While Cyclone Microsystems has many off the shelf modules available, users are also encouraged to build their own modules or contact Cyclone regarding custom modules and boards. Boards currently available through both Intel and Cyclone Microsystems are:

- Ethernet
- SCSI-2
- SCSI-3
- High Speed Serial

The Cyclone EP Functional Overview

I/O subsystem provides data buffers and simplified control:

- The Centronics compatible parallel port allows fast download of code or data to the Cyclone EP The asynchronous serial (RS-232) port provides transfers up to 115.2 Kbaud.
- A Z8536 timer/counter provides three 16-bit counters, with interrupts.

A single expansion bus (Squall II module) allows expansion cards and external devices direct access to the i960 processor's bus and control signals.

Product Ordering Information

EP80960BB	Base Board without CPU Module
CPU80960HX[1]	HX Microprocessor CPU Module
CPU80960CX	CX Microprocessor CPU Module
CPU80960SX	SX Microprocessor CPU Module
CPU80960KX	KX Microprocessor CPU Module
CPU80960JX	JX Microprocessor CPU Module

Available Q2 '95

I/O MODULES

SQETHERNET	ETHERNET MODULE
SQHSS[1]	HIGH SPEED SERIAL
SQSCSI-2[2]	SCSI-2
SQSCSI-3[2]	SCSI-3 (FAST AND WIDE)

[1] Not available on the CPU80960KX OR CPU80960SX

[2] Not available on the CPU80960SX

Support Information		
Product and Sales Information	Intel product information, sales office and distributor phone numbers	800-628-8686
Literature Department	To order Intel product literature	800-548-4725
Development Tools Hotline	Technical applications support for Intel development tools	800-873-4483
FaxBack Service	Automated response system that faxes Intel documents to your fax machine at no cost. For a listing of all i960 microprocessor documents, order FaxBack service document 4040.	800-628-2283
Applications Bulletin Board Service	24-hour access via modem to publicly available technical information. For a complete listing of all files available on BBS, call the FaxBack service and request category 6 (BBS Master File Listing).	916-356-3600

Phone numbers listed above are for U.S. and Canada only.

UNITED STATES
Intel Corporation
2200 Mission College Boulevard
P.O. Box 58119
Santa Clara, CA 95052-8119

JAPAN
Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26

FRANCE
Intel Corporation S.A.R.L.
1, Rue Edison, BP 303
78054 Saint-Quentin-en-Yvelines Cedex

UNITED KINGDOM
Intel Corporation (U.K.) Ltd.
Pipers Way
Swindon
Wiltshire, England SN3 1RJ

GERMANY
Intel GmbH
Dornacher Strasse 1
8016 Feldkirchen bei Muenchen

HONG KONG
Intel Semiconductor Ltd.
32/F Two Pacific Place
88 Queensway
Central

CANADA
Intel Semiconductor of Canada, Ltd.
190 Attwell Drive, Suite 500
Rexdale, Ontario M9W 6H8

Cyclone Microsystems, Inc.
25 Science Park
New Haven, CT
Phone: 203-786-5536
FAX: 203-786-5025
email: info@cyclone.com

Order Number: 272508-001
Printed in the U.S.A./1094/15k/JI KG

NORTH AMERICAN SALES OFFICES

ALABAMA

Intel Corp.
4024 Medford Drive
Huntsville 35802
Tel: (205) 883-6137
FAX: (205) 883-4826

ARIZONA

†Intel Corp.
410 North 44th Street
Suite 470
Phoenix 85008
Tel (800) 628-8686
FAX: (602) 244-0446

CALIFORNIA

Intel Corp.
3550 Watt Avenue
Suite 140
Sacramento 95821
Tel: (800) 628-8686
FAX (916) 488-1473

†Intel Corp.
9655 Granite Ridge Drive
3rd Floor, Suite 4A
San Diego 92123
Tel (800) 628-8686
FAX (619) 467-2460

Intel Corp
1781 Fox Drive
San Jose 95131
Tel. (800) 628-8686
FAX (408) 441-9540

*†Intel Corp.
1551 N Tustin Avenue
Suite 800
Santa Ana 92701
Tel: (800) 628-8686
TWX. 910-595-1114
FAX (714) 541-9157

†Intel Corp.
15260 Ventura Boulevard
Suite 360
Sherman Oaks 91403
Tel (800) 628-8686
FAX (818) 995-6624

Intel Corp
120 Birmingham
Suite 110-114
Cardiff, CA 92007
Tel: (619) 942-8938
FAX. (619) 942-2849

Intel Corp
300 N Continental Blvd
Suite 100
El Segundo 90245
Tel (800) 628-8686
FAX (310) 640-7133

COLORADO

*†Intel Corp
600 S. Cherry St
Suite 700
Denver 80222
Tel. (800) 628-8686
TWX 910-931-2289
FAX. (303) 322-8670

CONNECTICUT

†Intel Corp
40 Old Ridgebury Road
Suite 311
Danbury 06811
Tel. (800) 628-8686
FAX: (203) 778-2168

FLORIDA

†Intel Corp
800 Fairway Drive
Suite 160
Deerfield Beach 33441
Tel (800) 628-8686
FAX (305) 421-244

Intel Corp.
2250 Lucien Way
Suite 100, Room 8
Maitland 32751
Tel: (800) 628-8686
FAX: (407) 660-1283GEORGIA

†Intel Corp.
20 Technology Park
Suite 150
Norcross 30092
Tel. (800) 628-8686
FAX. (404) 448-0875

IDAHO

Intel Corp.
9456 Fairview Ave , Suite C
Boise 83704
Tel: (800) 628-8686
FAX (208) 377-1052

ILLINOIS

*†Intel Corp.
Woodfield Corp Center III
300 N. Martingale Road
Suite 400
Schaumburg 60173
Tell (800) 628-8686
FAX (708) 605-9762

INDIANA

†Intel Corp.
8041 Knue Road
Indianapolis 46250
Tel (800) 628-8686
FAX: (317) 577-4939

MARYLAND

*†Intel Corp
131 National Business Parkway
Suite 200
Annapolis Junction 20701
Tel. (800) 628-8686
FAX: (301) 206-3678

MASSACHUSETTS

*†Intel Corp.
Westford Corp. Center
5 Carlisle Road
2nd Floor
Westford 01886
Tel: (800) 628-8686
TWX. 710-343-6333
FAX (508) 692-7867

MICHIGAN

†Intel Corp
7071 Orchard Lake Road
Suite 100
West Bloomfield 48322
Tel. (800) 628-8686
FAX (313) 851-8770

Intel Corp
32255 N Western Hwy.
Suite 212, Tri Atria
Farmington Hills 48334
Tel (800) 628-8686
FAX. (313) 851-8770

MINNESOTA

†Intel Corp
3500 W. 80th St
Suite 360
Bloomington 55431
Tel (800) 628-8686
TWX. 910-576-2867
FAX: (612) 831-6497

NEW JERSEY

Intel Corp
2001 Route 46, Suite 310
Parsippany 07054-1315
Tel: (800) 628-8686
FAX. (201) 402-4893

*†Intel Corp.
Lincroft Center
125 Half Mile Road
Red Bank 07701
Tel: (800) 628-8686
FAX: (908) 747-0983NEW YORK

*Intel Corp.
850 Cross Keys Office Park
Fairport 14450
Tel: (800) 628-8686
TWX. 510-253-7391
FAX: (716) 223-2561

*†Intel Corp
2950 Express Dr. South
Suite 130
Islandia 11722
Tel (800) 628-8686
TWX. 510-227-6236
FAX: (516) 348-7939

OHIO

*Intel Corp.
56 Milford Dr , Suite 205
Hudson 44236
Tel: (800) 628-8686
FAX (216) 528-1026

*†Intel Corp
3401 Park Center Drive
Suite 220
Dayton 45414
Tel (800) 628-8686
TWX 810-450-2528
FAX (513) 890-8658

OKLAHOMA

Intel Corp.
6801 N Broadway
Suite 115
Oklahoma City 73162
Tel. (800) 628-8686
FAX (405) 840-9819

OREGON

†Intel Corp.
15254 NW Greenbrier Pkwy
Building B
Beaverton 97006
Tel: (800) 628-8686
TWX 910-467-8741
FAX (503) 645-8181

PENNSYLVANIA

*†Intel Corp
925 Harvest Drive
Suite 200
Blue Bell 19422
Tel. (800) 628-8686
FAX (215) 641-0785

SOUTH CAROLINA

Intel Corp.
7403 Parklane Rd , Suite 4
Columbia 29223
Tel (800) 628-8686
FAX (803) 788-7999

Intel Corp.
100 Executive Center Drive
Suite 109, B183
Greenville 29615
Tel. (800) 628-8686
FAX (803) 297-3401

TEXAS

†Intel Corp
8911 N Capital of Texas Hwy
Suite 4230
Austin 78759
Tel (800) 628-8686
FAX. (512) 338-9335

*†Intel Corp.
5000 Quorum Drive
Suite 750
Dallas 75240
Tel (800) 628-8686
FAX: (214) 233-1325

*†Intel Corp
20515 SH 249
Suite 401
Houston 77070
Tel. (800) 628-8686
TWX. 910-881-2490
FAX: (713) 376-2891

UTAH

†Intel Corp
428 East 6400 South
Suite 135
Murray 84107
Tel: (800) 628-8686
FAX (801) 268-1457

Intel Corp.
2581 E Cobblestone Way
Sandy, UT 84093
Tel: (801) 942-8820
FAX: (801) 942-8815

WASHINGTON

†Intel Corp
2800 156th Avenue SE
Suite 105
Bellevue 98007
Tel (800) 628-8686
FAX (206) 746-4495

WISCONSIN

Intel Corp.
400 N. Executive Dr
Suite 401
Brookfield 53005
Tel: (800) 628-8686
FAX (414) 789-2746

CANADA

BRITISH COLUMBIA

Intel Semiconductor of
Canada, Ltd
999 Canada Place
Suite 404, #11
Vancouver V6C 3E2
Tel. (800) 628-8686
FAX (604) 844-2813

ONTARIO

†Intel Semiconductor of
Canada, Ltd
2650 Queensview Drive
Suite 250
Ottawa K2B 8H6
Tel (800) 628-8686
FAX (613) 820-5936

†Intel Semiconductor of
Canada, Ltd
190 Attwell Drive
Suite 500
Rexdale M9W 6H8
Tel (800) 628-8686
FAX (416) 675-2438

QUEBEC

†Intel Semiconductor of
Canada, Ltd
1 Rue Holiday, Tour West
Suite 320
Pt. Claire H9R 5N3
Tel (800) 628-8686
FAX 514-694-0064

†Sales and Service Office
*Field Application Location

intel®

NORTH AMERICAN SERVICE OFFICES

PrimeService

Intel Corporation's North American Preferred Service Provider

Central Dispatch: 1-800-876-SERV (1-800-876-7378)

ALABAMA
Birmingham
Huntsville

ALASKA
Anchorage

ARIZONA
Phoenix*
Tucson

ARKANSAS
Little Rock

CALIFORNIA
Bakersfield
Brea
Carson*
Fresno
Livermore
Mar Del Rey
Ontario*
Orange
Sacramento*
San Diego*
San Francisco*
Santa Clara*
Ventura
Sunnyvale
Walnut Creek*
Woodland Hills*

COLORADO
Colorado Springs
Denver
Englewood*

CONNECTICUT
Glastonbury*

DELAWARE
New Castle

FLORIDA
Ft Lauderdale
Heathrow
Jacksonville
Melbourne
Pensacola
Tampa
West Palm Beach

GEORGIA
Atlanta*
Savannah
West Robbins

HAWAII
Honolulu

ILLINOIS
Buffalo*
Calumet City
Chicago
Lansing
Oak Brook

INDIANA
Carmel*
Ft Wayne

KANSAS
Overland Park*
Wichita

KENTUCKY
Lexington
Louisville
Madisonville

LOUISIANA
Baton Rouge
Metarie

MAINE
Brunswick

MARYLAND
Frederick
Linthicum*
Rockville*

MASSACHUSETTS
Boston*
Natick*
Norton*
Springfield

MICHIGAN
Ann Harbor
Benton Harbor
Flint
Grand Rapids*
Leslie
Livonia*
St Joseph
Troy*

MINNESOTA
Bloomington*
Duluth

MISSOURI
Springfield
St. Louis* NEVADA
Minden
Las Vegas
Reno

NEW HAMPSHIRE
Manchester*

NEW JERSEY
Edison*
Hamton Town*
Parsippany*

NEW MEXICO
Albuquerque

NEW YORK
Albany*
Amherst*
Dewitt*
Fairport*
Farmingdale*
New York City*

NORTH CAROLINA
Brevard
Charlotte
Greensboro
Haveluch
Raleigh
Wilmington

NORTH DAKOTA
Bismark

OHIO
Cincinnati*
Columbus
Dayton
Independence*
Middle Heights*
Toledo*

OREGON
Beaverton*

PENNSYLVANIA
Bala Cynwyd*
Camp Hill*
East Erie
Pittsburgh*
Wayne*

SOUTH CAROLINA
Charleston
Cherry Point
Columbia
Fountain Inn SOUTH DAKOTA
Sioux Falls

TENNESSEE
Bartlett
Chattanooga
Knoxville
Nashville

TEXAS
Austin
Bay City
Beaumont
Canyon
College Station
Houston*
Irving*
San Antonio
Tyler

UTAH
Salt Lake City*

VIRGINIA
Charlottesville
Glen Allen
Maclean*
Norfolk
Virginia Beach

WASHINGTON
Bellevue*
Olympia
Renton
Richland
Spokane
Verdale

WASHINGTON D.C.*

St. John WEST VIRGINIA
St Albans

WISCONSIN
Brookfield*
Green Bay
Madison
Wausau

CANADA
Calgary*
Edmonton
Halifax
London*
Montreal*
Ottawa
Toronto*
Vancouver, BC*
Winnipeg
Regina

NORTH AMERICAN CUSTOMER TRAINING CENTERS

ARIZONA
Computervision Customer Education
2401 W Behrend Dr , Suite 17
Phoenix 85027
Tel 1-800-234-8806

ILLINOIS
Computervision Customer Education
1 Oakbrook Terrace
Suite 600
Oakbrook 60181
Tel 1-800-234-8806

MASSACHUSETTS
Computervision Customer Education
11 Oak Park Drive
Bedford 01730
Tel 1-800-234-8806

SYSTEMS ENGINEERING OFFICES

MINNESOTA
3500 W 80th Street
Suite 360
Bloomington 55431
Tel (612) 835-6722

NEW YORK
2950 Expressway Dr , South
Islandia 11722
Tel (506) 231-3300

*Carry-in locations

i960® Processors and Related Products

The i960® embedded processor is the world's best selling 32-bit RISC processor and is used in a wide range of embedded applications including imaging, networking, communications and intelligent I/O. Whatever your application, there is an i960 processor to fit your cost and performance goals. This book contains detailed product specifications on the 14 different versions of the processor family that are now available. We've also included information on other related Intel products that can be used to enhance your design or speed your development.

intel

Order Number: 272084-004
Printed in USA/0195/12K/RRD/DS
Memory Products

Printed on Recycled Material