National
Semiconductor
Corporation

400036

Advanced Peripheral Processing Solutions

APPS:™ Handbook 1

Mass Storage

Advanced Peripheral Processing Solutions

MASS STORAGE

TRADEMARKS

Following is the most current list of National Semiconductor Corporation's trademarks and registered trademarks.

Abuseable™
Anadig™
ANS-R-TRAN™
APPS™
Auto-Chem Deflasher™
BI-FET™
BI-FET II™
BI-LINE™
BIPLAN™
BLC™
BLX™
Brite-Lite™
BTL™
CIM™
CIMBUS™
Clock✓Chek™
COMBO™
COPS™ microcontrollers
DATACHECKER®
DENSPAK™
DIB™
Digitalker®
DISCERN™
DISTILL™
DNR™
DPVM™
ELSTAR™
E-Z-LINK™
GENIX™
HEX 3000™
INFOCHEX™
Integral ISE™
Intelisplay™
ISE™
ISE/06™
ISE/08™
ISE/16™
ISE32™
LMCMOS™
Macrobus™
Macrocomponent™
Meat✓Chek™
Microbus™ data bus
(adjective)
MICRO-DAC™
μtalker™
Microtalker™
MICROWIRE™
MICROWIRE/PLUS™
MOLE™
MST™
National®
NAX 800™
Nitride Plus™
Nitride Plus Oxide™
NML™
NOBUS™
NSC800™
NSX-16™
NS-XC-16™
NURAM™
OXISS™
Perfect Watch™
Pharma✓Chek™
PLAN™
Polycraft™
POSitalker™
QUAD3000™
RAT™
RTX16™
Script✓Chek™
SCX™
Shelf-Chek™
SERIES/800™
Series 32000®
SPIRE™
STAR™
Starlink™
STARPLEX™
STARPLEX II™
SuperChip™
SYS32™
TAPE-PAK™
TDS™
TeleGate™
The National Anthem®
Time✓Chek™
TLC™
Trapezoidal™
TRI-CODE™
TRI-POLY™
TRI-SAFE™
TRI-STATE®
XMOS™
XPU™
Z STAR™
883B/RETS™
883S/RETS™

PAL® and PALASM™ are trademarks of and are used under license from Monolithic Memories, Inc.
DEC™, VAX™ and Unibus™ are trademarks of Digital Equipment Corp.
Intellec®, MULTIBUS® and MULTIMODULE™ are trademarks of Intel Corp.
Z80® is a trademark of Zilog Corp.
UNIX™ is a trademark of AT&T Bell Laboratories.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

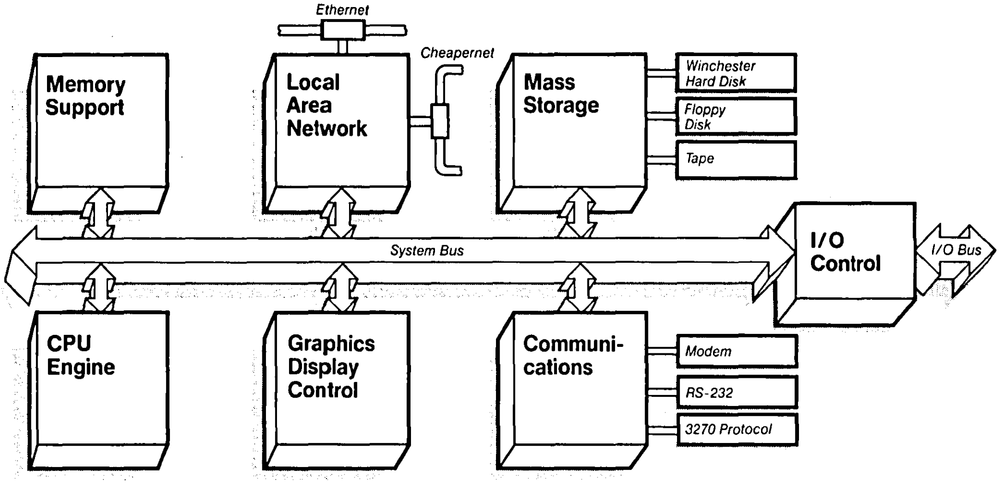
National Semiconductor Corporation 2900 Semiconductor Drive, P.O. Box 58090, Santa Clara, California 95052-8090 (408) 721-5000
TWX (910) 339-9240

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied, and National reserves the right, at any time without notice, to change said circuitry or specifications.

Introduction

Advanced
Peripheral
Processing
Solutions
APPS™

0009-1



0009-2

National Semiconductor APPS products include complex VLSI peripheral circuits designed to serve a variety of applications. The APPS products are especially well suited for microcomputer and microprocessor systems such as graphics workstations, personal computers, and many others. National Semiconductor APPS devices are fully described in a series of databooks and handbooks.

Among the APPS books are the following titles:

MASS STORAGE

The National Semiconductor family of mass storage interface products offers the industry's highest performance and broadest range of products for Winchester hard disks and floppy disks. The Mass Storage Handbook includes complete product information and datasheets as well as a comprehensive design guide for disk controller systems.

MEMORY SUPPORT

Today's large Dynamic Random Access Memory (DRAM) arrays require sophisticated high performance devices to provide timing and control. National Semiconductor offers the broadest range of DRAM controllers with the highest performance available on the market. Controllers are available for DRAMs from 64k bit through 1M bit devices, supporting memory arrays up to 8 Mbyte in size. For critical applications, National Semiconductor has developed several Error Checking and Correction (ECC) devices to provide maximum data integrity. The Memory Support Handbook contains complete product information and several application notes detailing complete memory system design.

LOCAL AREA NETWORKS AND DATA COMMUNICATIONS

Today's computer systems have created a huge demand for data communications and Local Area Networks (LANs). National Semiconductor supplies a broad range of products to fill the needs. The IEEE 802.3 Standard for Ethernet/Cheapernet LANs is one of the most popular solutions. National Semiconductor provides a complete three-chip solution for an entire 802.3 design. For mainframe communication the IBM 3270 and other coax protocols are another offering from National Semiconductor. To drive the communications lines, National Semiconductor has drivers and receivers designed to meet all the major standards such as RS-232, RS-422, and RS-485. Datasheets and applications information for all these products are in the LAN/DATA COMM Handbook.

GRAPHICS

Sophisticated human interface is a mark of the newest computer systems designs. Today's personal computer may have better graphics display capability than engineering workstations of a few years ago. National Semiconductor has developed a new family of Advanced Graphics products to provide extremely high performance, high resolution color graphics displays. The graphics chip set is designed to provide the highest level of performance with minimum demands and loading on the system CPU. The graphics system may be expanded to any number of color planes with virtually unlimited resolution. The Graphics Databook lays it all out and makes the display system design easy.

Table of Contents

Alpha-Numeric Index

Alpha-Numeric Index (Continued)

Alpha-Numeric Index (Continued)

Alpha-Numeric Index (Continued)

Alpha-Numeric Index (Continued)

Alpha-Numeric Index (Continued)

Alpha-Numeric Index (Continued)

Alpha-Numeric Index (Continued)

Alpha-Numerical Index (Continued)

Section 1

Disk Interface Design Guide and User Manual

Section Contents

Disk Interface Design Guide and Users Manual

National Semiconductor
Application Note 413
James Cecil
Ramachandran Gopalan
William Llewellyn
Shakeel Masood
Pat Tucci
Larry Wakeman

CHAPTER 1 DISK DRIVE TECHNOLOGY—OVERVIEW

1.0 INTRODUCTION—WINCHESTER DRIVES

From the start, digital computers have required some form of data storage as an adjunct to their relatively sparse main-storage facilities. Some of the early forms of storage were punched cards, paper tape and the magnetic tape storage. This was the principal storage medium, until faster-transfer, higher-capacity media became available and a direct link was established between the computer's main memory and the mass storage device. This link was the rotating memories, commonly referred to as disks.

Disk technology started a quarter-century ago, with the introduction of a large cumbersome fixed disk unit with 50 rotating surfaces 24″ in diameter, a single read/write head assembly, 600 ms seek time and a modest capacity of 5 megabytes. Half a decade later, capacities had increased by tenfold. Multiple head assemblies, one for each surface, introduced the concept of a "cylinder", providing simultaneous access to multiple tracks, one above the other, with a single head movement. Packing densities increased, resulting in increased storage capacity up to 100 megabytes. Head designs became more sophisticated; bits per inch increased by an order of 10; tracks per inch doubled.

Contamination-free Winchester technology was introduced by IBM in 1973. In addition to a controlled environment that eliminated dust collection on the disk surface, Winchester innovations included lightly loaded heads, an oriented iron-oxide coating to support higher flux reversal densities, and a silicone or wax coating that permitted heads to slide directly on the surface during "takeoff" and "landing"—eliminating the need for complex head loading mechanisms. The Winchester technology offers a number of advantages; device reliability, data integrity, faster transfer rates and a broader range of capacities. By the early 80's, fixed disk 14″ Winchester capacities were approaching 600 megabytes. Drives with capacities of 3 to 6 gigabytes are now on the immediate horizon. Winchester innovations also served as the springboard for miniaturized rigid disk systems. First came compact single or double-platter, non-removeable 14″ units with capacities down to 10 megabytes. Then around 1975 the 8″ Winchesters appeared, closely followed by the 5¼″ units, suitable for smaller desktop computers. Today the market boasts of a continuous spectrum of small to medium Winchester sizes: 3½, 5¼, 8, 10½ and 14 inches. Capacities begin at 5 Mbytes to 900 Mbytes.

The disk drive consists of one or more platters and heads, and the control mechanism with its associated electronics. The disk is essentially a platter made of aluminum or other base material, coated with iron-oxide or other magnetizable material. Each side of the disk consists of a number of thin annular regions called *tracks*. Each track is divided into blocks referred to as *sectors*. Data and other identification information is stored in the sectors. There are two types of sectoring: hard sectored discs and soft sectored discs. The hard sectored discs have sectors demarcated by the manufacturer and are identified by a sector pulse at the start of each sector while the soft sectored discs have only an index pulse signifying the start of a track.

The more recent hard disk drives have a number of platters on the same spindle, with one head per surface. In such cases similar track position on each platter constitutes a cylinder, e.g. cylinder 0 is the cylinder corresponding to track 0 on both sides of all the platters. The reading or writing of data is accomplished by the read/write head. This head is positioned on the required track by the drives positioning control system. This process is commonly referred to as seeking and is usually less than 17 ms. The quantity of data that can be stored on a disk depends on how much of its surface area is magnetized for the storage of a bit. On a typical low cost Winchester disk track densities are around 400 tracks per inch, while flux densities range around 9000 flux transitions per inch (implying recording densities of 9000 bits per inch). The rate at which data is written on the disk or read from it is termed as *transfer rate* and ranges from 5 Mbits/sec to 24 Mbits/sec and greater. The speed at which a particular sector is found for the writing or reading of data is gauged by the access time. First the head must be positioned over the proper track referred to as seek time. Then the proper sector of the track must come under the head which is referred to as the latency time. These are some of the common terms associated with the disk drive system.

The disk selection process is a function of several factors like storage capacity, upward mobility, transfer rate, etc. Data capacity is, perhaps, the most difficult decision to make in the selection process. All questions, present and future, must be considered in the context of the application. A fail, safe option, of course, would be to select a drive design with enough potential capacity to meet any future storage requirements. Disk technology has been striving to increase capacity, with future increases taking the form of increased data densities. There is considerable room for growth. Better head and disk material techniques are being used to raise track densities. Higher track densities have resulted in replacing the head positioning stepper motors by solenoid type "voice coil" actuators with theoretically infinite track following resolution. Developments in disk technology can also influence the transfer rate. The transfer rate directly affects system throughput. It is the average transfer rate that counts, and again this is a function of the application.

If write/read accesses are scattered because of varied reasons, track-seeking and sector-searching delays will reduce the effective transfer rate to a fraction of the theoretical value determined from data density and rotational speed. A series of application-dependent cost-performance tradeoffs must be individually evaluated. Higher rotational speeds reduce the latency time as the system waits for a desired sector to pass under the write/read heads. Multiple heads reduce both the number of head repositions and the distance that must be travelled. Lower cost stepper motor actuators are normally open loop—moving the heads from track to track at a constant, relatively slow rate. Voice coil actuators are more expensive but inherently faster, accelerating and decelerating in response to feedback signals from a closed loop servo system.

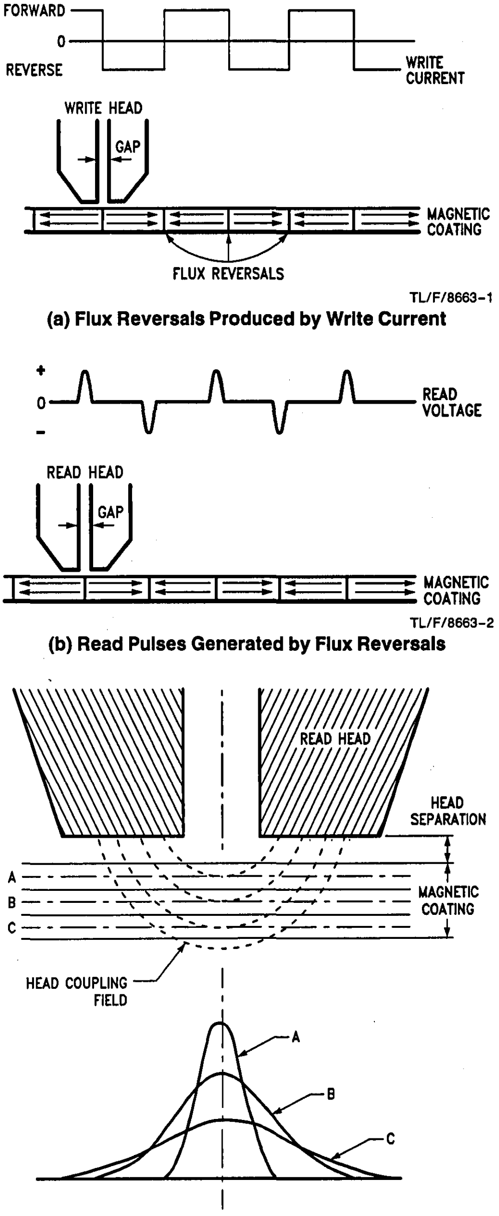
1.1 DISK STORAGE BASICS

Magnetic writing—the recording of data in a magnetic medium, is based on the principle that if a current flows in a coil of wire, it produces a magnetic field. The field is largely confined in a ring-shaped core of magnetic material, around which the wire is wound. A narrow slot is cut in the magnetic material and the field in the vicinity of the slot magnetizes the magnetic medium on the disk surface. Thus it creates alternating north-south magnets in the coated surface of the rotating disk. Thereby data is written, refer to *Figure 1.1(a).*

The head that writes the data can also be used to read it. This is done based on the principle of induction wherein a voltage is induced in an open circuit (like a loop of wire) by the presence of a changing magnetic field. In the case of a head positioned above a spinning magnetic disk on which data has been written, the magnetic fields emanate from the magnetized regions on the disk. During the time the head is over a single magnetized region, the field is more or less uniform. Hence no voltage develops across the coil that is part of the head. When a region passes under the head in which the magnetization of the medium reverses from one state to the other, i.e. a flux reversal, there is a rapid change in the field, developing a voltage pulse, refer to *Figure 1.1(b).* In this way the digital data are read as an analog signal, which can be readily converted back to digital form. The shape of this pulse and its ability to be recovered depends on various spacings. *Figure 1.1(c)* shows the spread of the coupling effect as a function of the width of the read-head gap and, equally important, the distance from the gap. The latter is, in turn, a function of both the head-surface separation and the depth of the flux reversal within the magnetic coating.

The quality of writing and reading of data depends of course on the magnetic properties of both the medium in which the data are stored and the head that writes and reads them. The common method of disk manufacture is to coat an aluminum disk with a slurry containing the gamma form of iron oxide. The iron atoms in the needle-like particles have their own minute magnetic fields and act like bar magnets with a dipole. The overall magnetization in any given region of the disk is the sum of the fields of these particles within it.

The core of most read/write heads is a ceramic consisting of spherical ferrite particles. The design of the head must conform to the design of the disk. In the case of the floppy disk(or flexible disket), which is a thin sheet of mylar plastic on which the gamma form of iron oxide is coated, the head makes contact with the surface, resulting in higher error rates and greater wear of the medium. In high performance disk drives, the magnetic medium is the coating on a rigid aluminum disk, and the head is kept from touching the medium by the so-called air-cushion effect. Consider a head that is nearly in contact with the surface of a hard disk spinning at 3600 revolutions per minute. If the length of the head along the direction of relative motion is two orders of magnitude longer than the separation between the head and the medium, the flow of air between the head and the medium provides support for a head weighing up to several grams. Therefore because of the high cost of producing a hard disc along with its large storage capacity, it is used even if it has a bad track or sector while a floppy could be discarded. The bad sector is detected by using error checking and correction codes.



TL/F/8663-1

(a) Flux Reversals Produced by Write Current

TL/F/8663-2

(b) Read Pulses Generated by Flux Reversals

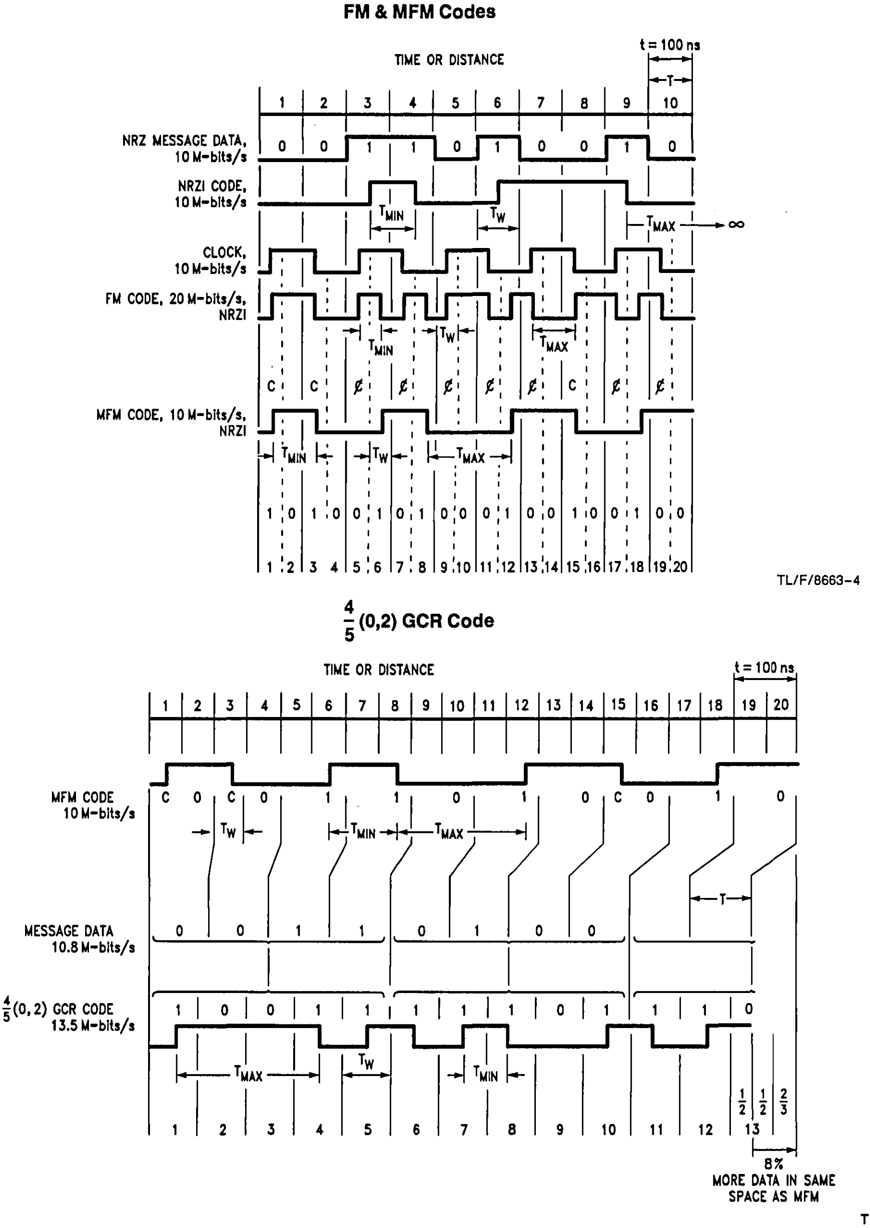
TL/F/8663-3

(c) Output as a Function of Saturation

FIGURE 1.1 Flux Reversals as They Relate to "Writing To" and "Reading From" the Platter

Optical Disk Technology

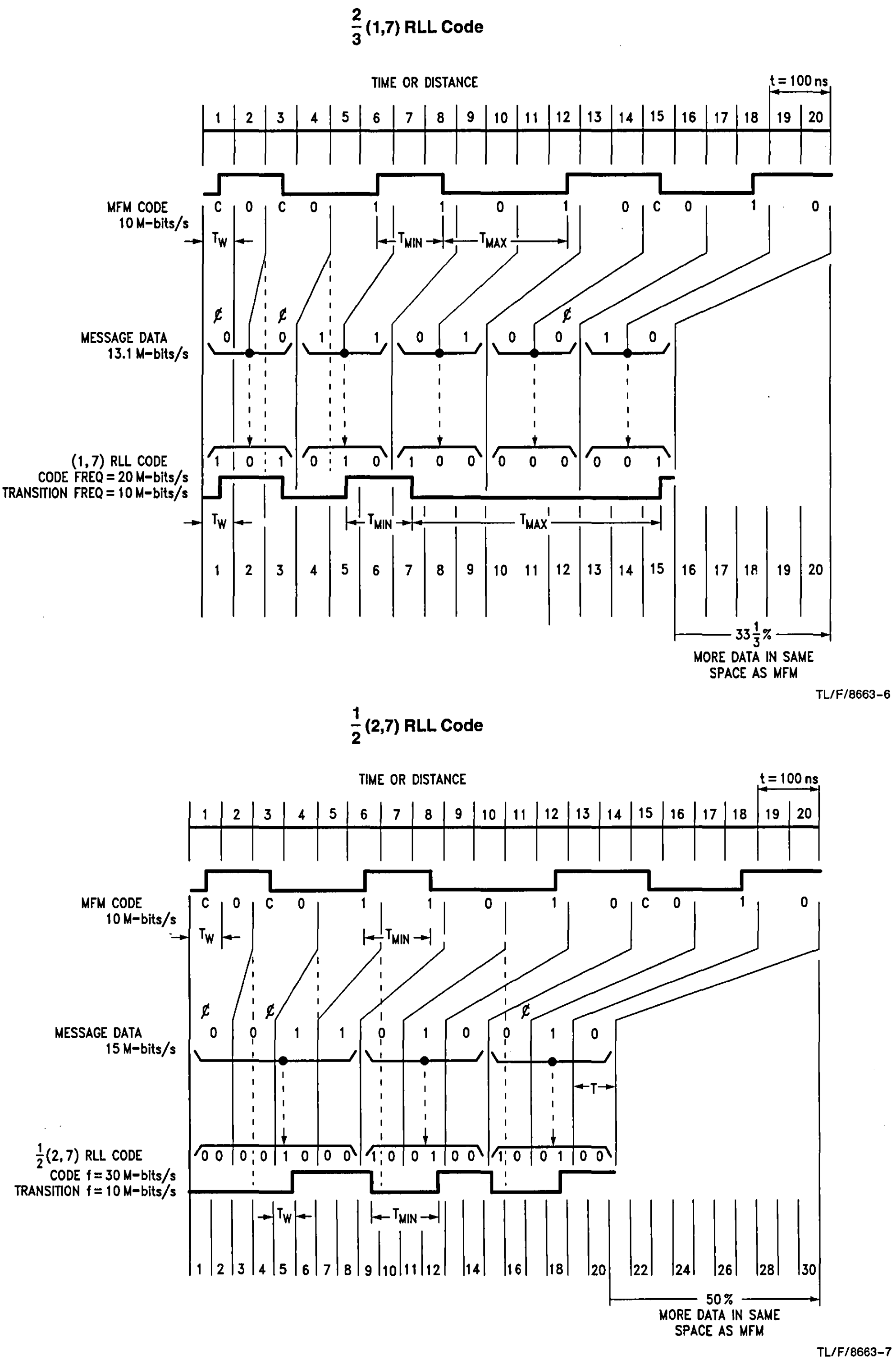
Disk drive improvements have resulted in faster data rates caused by increasing the density of the magnetic particles for greater storage. In the case of rotating magnetic memories, the strength of the signal depends on the strength of the medium's remnant magnetization. Recent advances in laser technology have resulted in digital optical disks becoming the last word in data storage and retrieval. Here, the laser beam itself provides the energy, hence the head is not in contact with the medium and it is protected, resulting in reduced errors and minimum medium wear. The advantages offered by optical disk technology are increased storage capacity, long data life, low cost per bit, noncontact read/write and easy physical mass replication. The optical disks initially developed could be written to only once. Read/write optical technology is being developed. Applications for optical disks are many and varied. On the interface level it is no different from Winchester drives and SCSI seems to be one of the most suitable of several possible choices. Another magnetic disk technology, "vertical" recording, is done with north-south magnetic poles perpendicular to the disk surface instead of end-to-end along the track.



FM & MFM Codes

$\frac{4}{5}$ (0,2) GCR Code

FIGURE 1.2 (a). FM, MFM and GCR Encoding



$\frac{2}{3}$ (1,7) RLL Code

TL/F/8663-6

$\frac{1}{2}$ (2,7) RLL Code

TL/F/8663-7

FIGURE 1.2 (b). RLL Encoding Schemes

1.2 DATA ENCODING/DECODING

Disk Data Encoding is the specific technique by which data is written to the disk, whereas decoding of data is necessary while reading from the disk. Data encoding removes the need of having clock information added to the track. Encoding also assists the controller in resynchronizing the data to the correct byte alignment, by allowing code violations for special data and address marks on the track. Considering the demand for ever-increasing data densities, it is understandable that the selection of a particular code is based largely on the efficiency with which flux reversals are converted into binary information, ZEROs and ONEs. In the ideal case, there should be the fewest flux reversals relative to the number of data bits they represent. Ideally, too, the code itself should provide its own "clock" for identifying the bit-cell intervals. Lacking this feature, a separate clock track may be required—or an extremely accurate oscillator must be provided to maintain the bit-cell divisions during intervals without flux reversals. The two requirements tend to be contradictory. An efficient code in terms of flux reversals will not be self-clocking. A self-clocking code will be wasteful of flux reversals. Nearly all of the widely used codes represent a compromise between these two extremes. *Figure 1.2(a)* shows details of FM, MFM and GCR encoding schemes, while *Figure 1.2(b)* shows details of some RLL encoding schemes. The commonly used encoding methods are discussed below in brief.

NRZ (NON-RETURN TO ZERO)

This is a telecommunication code and by far the most efficient. "Zero" refers to the transmission signal level. Instead of discrete pulses for each data bit, the signal rises or falls only when a ZERO bit is followed by a ONE bit or a ONE by a ZERO. NRZ coding reduces signal bandwidth by at least half. It also requires precise synchronization between source and destination in order to maintain bit cell divisions during the transmission of long strings of ZEROs or ONEs. NRZ could be used to transmit serial data to or from a magnetic recording device, disk or tape. But the extended intervals which can occur between flux reversals limit its usefulness as a recording technique.

NRZI (NRZ CHANGE ON ONEs)

This is the next most efficient code. It is widely used for tape recording and, to an increasing degree, disk recording. All ONEs are clocked, but special steps must be taken to compensate for the absence of flux reversals during strings of ZEROs. In the case of parallel-bit recording (tape), parity-bit ONEs serve as clock when all other bits in the byte are ZERO. In the case of serial-bit recording, data can be converted to RLL code (discussed below) which restricts the number of successive, unclocked ZEROs.

PE (PHASE ENCODED)

This is the least efficient of the coding methods but is completely self-clocking. The direction of a flux reversal at the middle of each cell indicates whether the bit is a ZERO or a ONE. Either one or two flux reversals occur, therefore, during each bit cell interval. The effect is to shift the "phase" of the signal by 180 degrees each time there is an NRZ type transition between ZEROs and ONEs.

FM (FREQUENCY MODULATION)

The FM method of encoding is equivalent to the PE technique and was the first choice for early disk-recording systems. It is generally only used for older floppy drives. Every bit cell interval is clocked by a flux reversal at the start of the cell. ONEs are marked by an additional flux reversal at the middle of the cell, doubling (modulating) the frequency of flux reversals for a series of ONEs compared to a series of ZEROs. A constant bit cell reference, provided by the clock bit, simplifies encoding and decoding with this scheme.

MFM (MODIFIED FREQUENCY MODULATION ENCODING)

With available head and media technology, MFM encoding is the most easily implemented encoding scheme and by far the most popular for floppy drives. It is used in the IBM System/34 and in available double-density LSI controller chips. MFM encoding doubles the data capacity over FM by eliminating the clock transitions (used in FM encoding) with data bits, refer to *Figure 1.2(b)*. Clock bits are still used, but are written only when data bits are not present in both the preceding and the current bit cell. As a result there is a maximum of one flux change per bit cell. Clock bits are written at the beginning of the bit cell, while data bits are written in the middle of the bit cell.

To decode data bits in MFM encoding, a data separator must generate a data window and a data window complement for a clock window. Because not every bit cell has a clock pulse, the data/clock windows cannot be timed from the clock pulse. Instead, the data separator must continuously analyze the bit position inside the windows so that the data/clock windows remain synchronous with the data/clock bits. Ideally, the clock transitions should appear at the center of the window. However, clock edges data bits can shift due to bit-shift effects. Present LSI controller chips can handle the drive interface, double density encoding function, and bit-shift pattern detection and compensation. National's DP8466 takes care of all these functions and needs only the data separator DP8465. Despite these constraints, disc controller design for MFM is simpler than that for either of the following encoding schemes.

M^{2}FM RECORDING SCHEME (MODIFIED-MODIFIED FREQUENCY MODULATION ENCODING SCHEME)

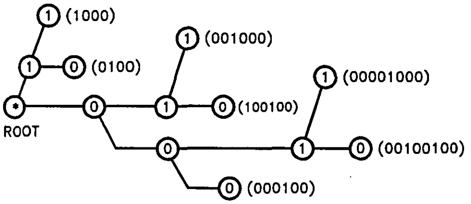
Until recently, M^{2}FM has been used as a double density encoding scheme, because the resolution of the medium and the read/write head was not adequate for the sizes of data window used in MFM. In M^{2}FM, a clock is written only if no data or clock bit is present in the preceding bit cell, and no data bit occurs in the current cell. Because clock pulses are relatively isolated on the medium, the effect of bit shift on clock pulses is minimal. Therefore, a narrower clock window can be used to decode the clock pulse. The width of the data window can thus be increased by 20%, which allows more margin for shifted data bits. Today's ceramic-based read/write heads have much better resolution than those used in the past. This head design reduces the effects of bit shift, and makes the window margin provided by M^{2}FM unnecessary. Additionally, M^{2}FM is subject to a droop problem, which occurs in the read amplifier circuit when a low frequency pattern is read.

GCR (GROUP CODED RECORDING) ENCODING SCHEME

GCR encoding evolved from methods used in magnetic tape recorders. This method translates four data bits into a 5-bit code during a write. During a read, the 5-bit code is retranslated to four data bits; no clock bits are generated. Using data rates specified by drive manufacturers, this scheme is less dense than MFM. This method requires more circuits to code and decode, requiring necessary look-up tables, and costs more than either of the other two encoding schemes. For example; 1101 is encoded into a serial bit stream 01101 according to GCR encoding rules. To decode, a data window is generated around the expected position of each bit. The result is serial read data of 01101, which must be decoded to 1101 by lookup tables.

RLL (RUN LENGTH LIMITED) ENCODING SCHEMES

These recently popular encoding schemes are used in big 14″ drives from IBM, CDC and DEC, and are starting to make an appearance in the small 5¼″ drive market. The RLL encoding schemes have an excellent encoding efficiency, up to 50% higher than MFM. It is, however, considerably more complex to generate, requires a much better data separation unit to recover recorded data and is more suceptible to wider error bursts. The encoding rules for RLL depend on the RLL scheme chosen. The most common one is the 2,7 RLL code which refers to the maximum number of consecutive 0s, refer to *Figure 1.2(b)*. A standard encoding tree is defined and the data is encoded on the basis of those rules, as shown in *Figure 1.3*. The data bit stream is taken and the encoding tree is traversed, starting at the root, where the nodes traversed are the data bit stream in the sequence they arrive. On reaching the leaf of the tree, the code there is then the 2,7 RLL code for that data stream, e.g. if there is a data stream 100011010, then on traversing the tree, a data bit stream of 10 has a code of 0100, while the next bits 0011 are encoded as 00001000.



TL/F/8663-8

FIGURE 1.3. Encoding Tree—2,7 RLL Code

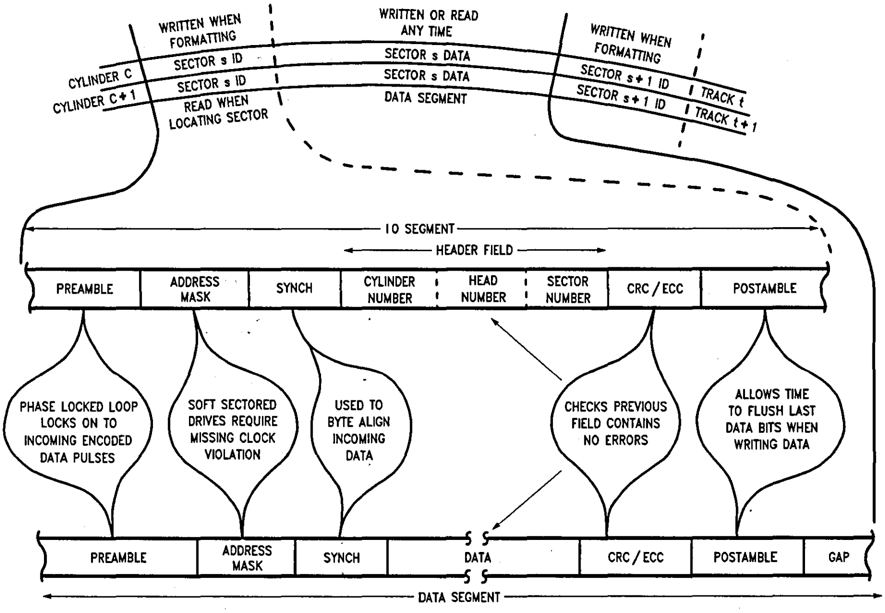
1.3 MEDIA FORMATTING

Media formatting provides the user with a reliable means of data retrieval using the magnetic recording surface of the track. There are many different formats but most of them are variations of the same basic structure. The formatting process is different for hard sectored and soft sectored disks. In hard sectored disks the sectors are defined by the manufacturers because the start of each sector is identified by the sector pulse generated by the drive. In soft sectored disks the drive issues only an Index pulse at the beginning of the track and the user can define all details of how information will be stored on the track, allowing more flexibility. *Figure 1.4* shows the basic format used. It consists of two segments—the ID segment and the data segment. The ID segment contains unique header information for the sector and the data segment contains the actual data. When the system requests a particular sector on a disk, the head must be positioned over the selected track, and the desired sector on that track must be found. This requires electronics to lock on to the data stream and then decode it. The beginning of a track is indicated by the Index pulse while the beginning of the sector is indicated by the sector pulse. This is followed by gap before the start of the sector on the track, which is referred to as the **Post Index/Sector Gap**. The explanation of the various fields are given below:

1.3.1 ID Segment

PREAMBLE OR PLL SYNCH FIELD

This is a field of repetitive clocked data bits usually 10 to 13 bytes long. The preamble normally will be all zeroes of NRZ data (encoded as 1010... in MFM). During the ID preamble, the signal Read Gate will go active, indicating that the incoming data pattern has to be locked on to.



TL/F/8663-9

FIGURE 1.4. Typical Sector Format Showing the Various Fields Within the ID and Data Segments

ADDRESS MARK FIELD

Address Mark (AM) is required on soft sectored drives to indicate the beginning of a sector, because this type of drive does not have a sector pulse at the start of each sector. This address mark byte contains a missing clock code violation, typically in MFM. The violation is detected by circuitry to indicate the start of a sector. The first decoded byte that does not contain all 0s after the preamble will be the address mark. The first 1 to be received is then used to byte align after the all zeroes preamble. Some formats have one ID address mark byte, while others have several.

ID SYNCH FIELD

For a hard sectored disk, byte alignment begins with the synch field that follows the preamble. The Synch bytes constitute a bit pattern that enables control circuitry to determine the byte boundaries of the incoming data, bit synchronization. Synch field usually follows the address mark on soft sectored drives and the AM is used for byte alignment also. Some formats use two synch fields: synch #1 and synch #2.

HEADER FIELD

The Header Field format varies between drive types, but typically has two cyclinder number bytes, a sector number byte, and a head number byte. It is generally 3 to 6 bytes long and one of the bytes may contain bits for bad sector or bad track recognition.

HEADER CRC/ECC FIELD

CRC (Cyclic Redundancy Checking) code or ECC (Error Checking and Correcting) code is appended to the header field. If CRC is used it consists of two bytes of the standard CRC-CCITT polynomial. The code detects errors in the header field. If ECC code is used, it is normally the same ECC polynomial that is used for the data field. This appendage is basically a protection field to make sure that the ID field contains valid information.

POSTAMBLE

This field may be used to give the disk controller time to interpret the data found in the ID field and to act upon it. It provides slack for write splicing that occurs between the ID and Data segment. A Write splice occurs when the read/write head starts writing the data field. A splice is created each time a sector's data segment is written to. The slight variations in the rotational speeds cause the first flux change to occur in different positions for each write operation. It also allows time in a write disk operation for the read/write circuitry to be switched from read to write mode. Finally it allows time for the PLL circuit to re-lock on to a fixed reference clock before it returns to synchronize to the preamble of the data field.

1.3.2 Data Segment

PREAMBLE FIELD

The Data Preamble field is necessary when reading a sector's data. It ensures that the PLL circuit locks on to the Data segment data rate. Initially, the ID segment and the data segment of every sector will be written when formatting the disk, but the Data segment will be written over later. Due to drive motor speed variations within the tolerance specified, the ID and Data segments will have slightly different data rates because they are written at different times. This implies that the PLL must adjust its frequency and phase in order to lock on to the data rate of the Data segment before the incoming preamble field has finished. Hence the need for a second preamble field in the sector.

DATA ADDRESS MARK FIELD AND DATA SYNCH FIELD

Following the Data Preamble will be the Data Address Mark for soft sectored drives, and Data synch, both similar to the ID segment equivalents.

DATA FIELD

The Data field is transferred to or from external memory. It is usually from 128 bytes to 64 kbytes per sector.

DATA CRC/ECC

A CRC/ECC appendage usually follows the Data field. CRC/ECC generating (when writing to the disk) and checking (when reading from the disk) are performed on the Data field. Errors may therefore be detected, and, depending on the type of error and if an ECC polynomial is used, they may also be corrected.

DATA POSTAMBLE FIELD

This has the same function as the ID Postamble field.

GAP FIELD

This is sometimes referred to as Gap 3, and is the final field of the sector. It allows slack between neighboring sectors. Without this gap, whenever a data segment is written to a sector, any reduction in drive motor speed at the instance of writing to the disk would cause an overlap of the data segment and the succeeding ID segment of the next sector. This field is only written when formatting the disk.

A final gap field is added from the end of the last sector until the INDEX pulse occurs and this gap is often termed Gap 4. It takes up the slack from the end of the last sector to the Index pulse.

1.4 THE DISK SYSTEM—DRIVE AND CONTROLLER

The Disk system essentially consists of two main paths: 1)the Disk Data Path, 2) the Disk Control Path, refer to *Figure 1.5*. The disk control path is responsible for controlling the disk drive with respect to positioning the head at the desired track and control the associated control signals (these are a function of the disk interface). The various disk interfaces are discussed later. The other component of the Disk System is the Disk Data Path which is responsible for data transfer from and to the disk.

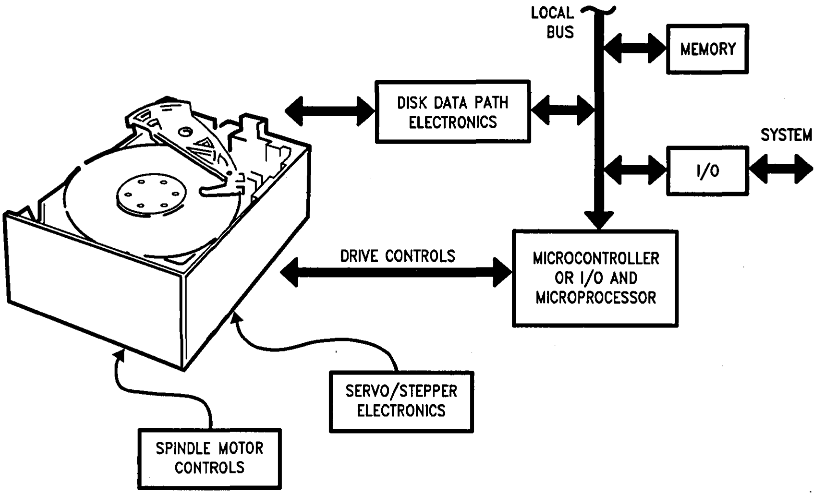
1.4.1 Reading Data from the Disk

Reading data from the disk to the system memory is a complex process and involves a number of operations enroute, as shown in *Figure 1.6(a)*. To initiate a read operation—a command is sent to the disk drive indicating the track and sector from which data is to be read. The seek operation moves the head to the desired track on the disk. Eventually the desired sector is identified by the header ID segment and the various fields are checked depending on the formatting rules used. The flux reversals are recorded by the head and are of the order of 500 microvolts. These pulses are then amplified by the read/write amplifier to about 10 mV. The signal from the read/write amplifier when reading a disk is therefore a series of pulses with alternating polarity. These pulses are passed through a Pulse Detector, like the DP8464. Electrically, these peaks correspond to flux reversals on the magnetic medium. The Disk Pulse Detector accurately replicates the time position of these peaks. The Disk Pulse Detector utilizes analog and digital circuitry to detect amplitude peaks of the signal received from the read/write amplifier associated with the heads of disk drives. A TTL compatible output is produced which on the positive leading edge indicates a signal peak.

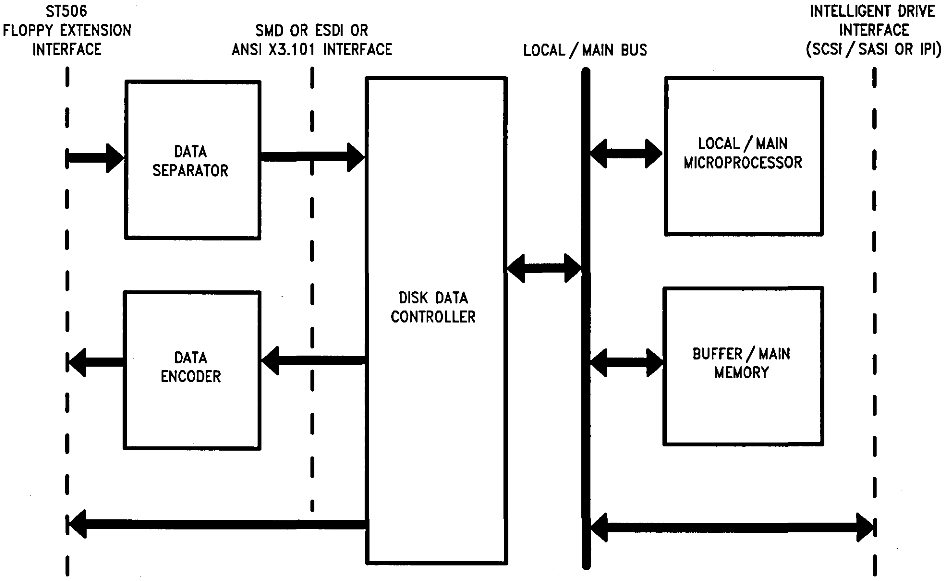
The raw data coming from the pulse detector consists of composite clock and data bits depending on the encoding scheme used. This encoded data has to be synchronized and decoded. These functions are performed by the Data Separator, like the DP8465. Due to bit shifting and distortion of the read pulses, the Pulse Detector issues non-synchronous pulses. For reliable decoding this jittery bit stream must be synchronized. The data separator has a Phase Locked Loop which attempts to lock on to the bit stream and synchronize it.

In hard sectored drives, the sector pulse indicates the beginning of the sector. Normally the preamble pattern does not begin immediately, because gap bytes from the preceding sector usually extend just beyond the sector pulse. Allowing two bytes to pass after the sector pulse helps ensure that the PLL will begin locking on to the preamble and will not be chasing non-symmetrical gap bits. For soft sectored drives, the controller normally will not wait for the Index pulse before it attempts lock-on. Chances are the head will not be over a preamble field and therefore there is no need to wait two bytes before attempting lock-on.

Disk Data Controller in a Disk System

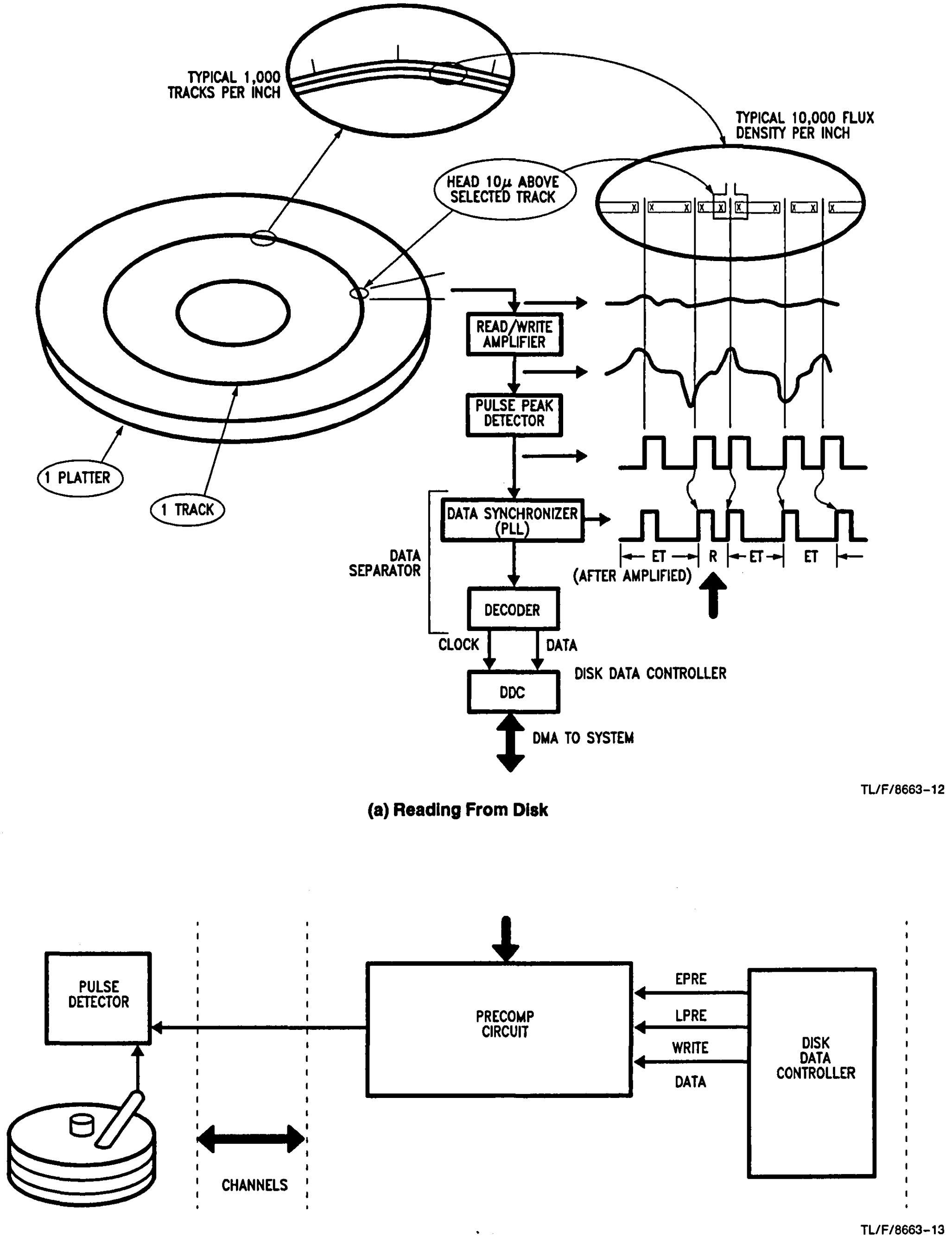

LOCAL BUS
MEMORY
DISK DATA PATH ELECTRONICS
SYSTEM
I/O
DRIVE CONTROLS
MICROCONTROLLER OR I/O AND MICROPROCESSOR
SERVO/STEPPER ELECTRONICS
SPINDLE MOTOR CONTROLS

TL/F/8663-10


ST506 FLOPPY EXTENSION INTERFACE
SMD OR ESDI OR ANSI X3.101 INTERFACE
LOCAL / MAIN BUS
INTELLIGENT DRIVE INTERFACE (SCSI / SASI OR IPI)
DATA SEPARATOR
DATA ENCODER
DISK DATA CONTROLLER
LOCAL / MAIN MICROPROCESSOR
BUFFER / MAIN MEMORY

TL/F/8663-11

FIGURE 1.5. Disk System—Data and Control Paths



TL/F/8663–12

(a) Reading From Disk

TL/F/8663–13

(b) Writing to Disk (MFM Encoding)

FIGURE 1.6

Having locked-on to the bit stream the data synchronizer circuit must first determine the nominal position of clock and data bits and then generate an appropriate clock and data window that is centered around the bit positions. However there are many causes for bits to shift from the position where they are written. Erroneous data could be issued if bit jitter is beyond the tolerance computed. Therefore, special design considerations must be given to the type and resolution of the Data Separator used in reading data bits from the disk. The more accurately the bit position can be determined and the tighter the resolution of the data window, the lower is the soft error rate of the disk. Essentially the Data Separator's Phase Locked Loop locks on to the basic frequency of data bits read from the disk, and determines nominal bit positions for data and clock bits by sampling every bit (clock and data). It uses the phase relationship between a bit and its window to vary the position of the window. By sampling each bit, the phase-lock loop determines the phase error between a bit and the frequency being generated. To determine the nominal bit position around which to center the window, the data separator must track data bit frequency changes, yet ignore jitter. In this manner, even if an unpredictable bit shift occurs, the data separator can adjust the window's position to compensate for the change. Otherwise the shifted bit could be positioned outside the window. To remain within the typical error rate specified by the system, not more than 1 in 10^{10} bits can appear outside the window. With the present media technology, only a data separator based on an analog phase-lock loop technique can provide the necessary reliability.

Once the bit stream read from the disk has been synchronized and decoded to NRZ data, it is directly sent to the Disk Data Controller block, DDC, like the DP8466. In the DDC, the serial data is converted to parallel data (in terms of bytes), by the deserializer block. The main task is to recognize the byte boundaries accurately. In soft sectored drives this can be done by detecting a "missing clock" signal, which provides a fixed reference in the bit stream to set the byte boundary. Upon receipt of this signal the divide-by-eight circuit is set, to allow subsequent stages of the controller to acquire the bytes correctly. Hard sectored drives use a preset bit pattern in the synch field to determine byte alignment. Once the data is in parallel form it is stored in a temporary register in the controller. Transfer of data from this register to the system memory is achieved by DMA (Direct Memory Access) transfer. In this fashion data are read from the disk and transferred to the system.

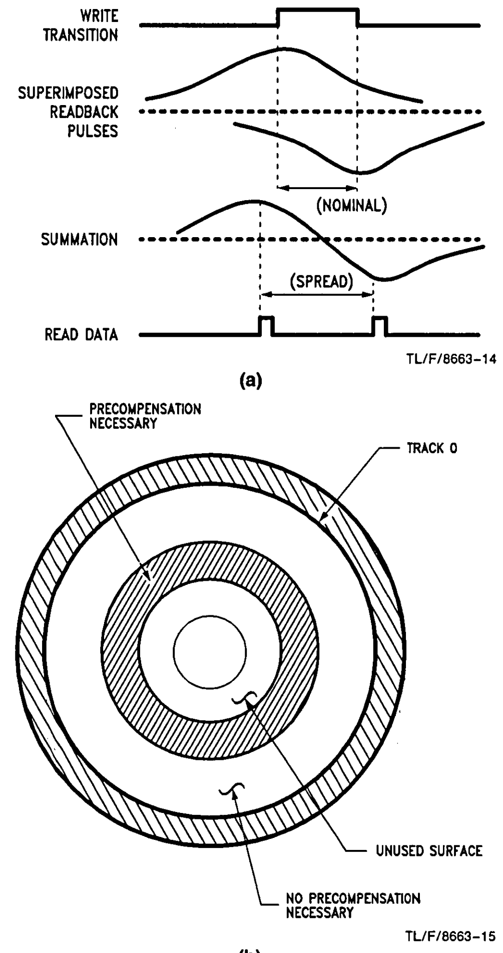
1.4.2 Writing Data to the Disk

The process of writing data to the disk is similar to the read operation in the reverse direction, with some changes. The write operation is initiated after the appropriate Seek command has been issued to the drive and the head is positioned over the desired track/sector. *Figure 1.6(b)* shows the basic write path blocks. Data is transferred from the system to the Controller using the DMA. The parallel data is converted to serial data by the serializer in the controller. This operation is conceptually easier to do, as the controller already has the right byte boundaries in the data and knows exactly where to insert the address mark. Most disk Controllers, like National's DP8466, provide either NRZ encoded data or MFM encoded data.

As mentioned in the previous section, predictable bit shift effects result from normal read/write head operation. Data are written when the read/write head generates a flux change in the media. In reading, a current is induced into the read/write head when a flux transition on the medium is encountered. The current change is not instantaneous, since it takes a finite time to build up to the peak and then to return to zero, refer to *Figure 1.7(a)*. If flux transitions are close together, the signal buildup after one flux transition declines, but it does not reach zero before a second transition begins. So when the flux changes are detected by the read/write head the peaks are shifted. A negative flux change, for example, may appear late because it has been added to the remnant of a positive transition. Narrower spacing between bits results in greater bit shift on the inner tracks. Hence compensation is needed on the inner tracks to minimize bit shift while no compensation is required on the outer tracks as bit shift is negligible, *Figure 1.7(b)*. Two methods currently being used are *precompensation* and *postcompensation*.

With precompensation, bits are deliberately shifted in the direction opposite to that of the expected shift. As data are being written, the controller detects bit patterns. From these bit patterns, the controller calculates which bit will shift in which direction. For example, a 4-bit pattern of 0110 on an inner track would cause the third bit to appear a few nanoseconds later than its nominal position. The controller chip, after detecting this late bit shift pattern, would generate an early signal, indicating that the third bit should be written earlier to make it appear closer to its nominal position when read. Conversely, if the third bit were going to appear early, a late signal would be generated so that the bit could be written later. How early or late the bit should be written is a function of its position in the data pattern, track position, and media, among other factors. Most Controllers provide signals to indicate what type of compensation is necessary. External circuitry is used to provide the actual delay as shown in *Figure 1.7(b)*.

The encoded precompensated data is then sent to the read/write amplifier where the stream of pulses is recorded on the disk as magnetic flux reversals. Postcompensation can be used when reading, usually as filter components around the pulse detector.



(a)

(b)

FIGURE 1.7. Bit Shifting

1.4.3 DMA (Direct Memory Access) Transfer/Data Buffering

The DMA block is responsible for the transfer of data between the host system and the disk controller. This is done because it is inefficient to dedicate a special communication channel to the task of transferring data between the disk controller and the system. The DMA system takes control of the control lines associated with a system's address and data buses, and exercises them in such a way as to transfer data in an appropriate direction from one device to another. It is also generally optimally efficient in using the available bus bandwidth whenever it is on the bus. The DMA capability is built-in for some disk controllers while in some an external one is required. National's DP8466 supports a single or dual channel DMA with capability of using an external DMA instead, if desired.

Data buffering is the temporary storage of some or all of the data to be transferred between the disk and the system memory. Any centrally intelligent system benefits from minimizing the bus occupancy. This is because the system has a lot of other tasks to perform, and if the bus is too heavily used, the system will miss performing some timely tasks. Therefore to prevent this, the data from the disk is transferred to a FIFO (First In First Out buffer). In a dual-DMA system, the local channel transfers the data from the FIFO to a local buffer memory while the remote DMA channel optimally transfers data from the local to the remote main memory over the system bus. This minimizes bus bandwidth use by the Disk I/O channel. The size of the FIFO is a function of different factors like: 1) the rate at which the system picks up the blocks of data, 2) the data rate from the disk and, 3) the burst transfer rate of the DMA. The way this is incorporated may differ in disk controllers. The buffer memory optimizes bus bandwidth. A system utilizing a single channel DMA would transfer data from the FIFO directly to the Host.

1.4.4 Error Detection/Error Correction

There are a number of factors which contribute to disk errors, viz. electrical noise, crosstalk, inadequately erased signals from previous recording, offtrack error in positioner, pin holes, inclusions, media thinning, and pattern induced errors. Of these, media defects are permanent errors. In general, ECC (error checking and correcting code), ensures reliable data storage and recovery. Generation of the ECC polynomial involves a detailed understanding of the mathematics of coding theory, and a cookbook approach to designing ECC logic. The basic idea behind ECC is the concept of irreducible polynomials. Take an irreducible polynomial (prime) and multiply it by the data pattern. Store the resulting remainder on the disk after all the data has been sent through the polynomial, *Figure 1.8(a)*. When reading the data back, divide the data coming off the disk into the polynomial. The reciprocal of the result should be equal to the check bytes on the disk, *Figure 1.8(b)*. If not, there is an error. The way error correction works is shown in *Figure 1.8(c)*. If there were no errors, then the sequence follows the straight path and the shift register contains all zeroes. If an error occurred, then at the point of occurrence the sequence vectors off, and at the end the shift register contains the pattern which caused the error. This helps in tracing back to the point of occurrence. The correction span is the number of contiguous bits in error which could be corrected. The probability of miscorrection is given by:

$$P_{mc} = (2^{(C-1)}) \times S/2^A$$

where C = correction span in bits
S = no. of bits in the sector
A = no. of bits in ECC appendage

Some codes have a higher miscorrection probability due to pattern sensitivities. A 48-bit ECC with an 11-bit correction span is recommended for 1,7 or 2,7 Run Length Limited encoded disks, while for MFM a 32-bit ECC with a 5-bit correction span results in low miscorrection probabilities. There are different types of codes:

FIRE CODES

Used in older systems and some current chips on the market. Fire codes have a high miscorrection probability related to double-burst errors (errors at two locations separated by more than the detection span) and are not recommended by National. Some examples of fire codes are:

32-bit FIRE CODE	$(x^{21}+1)*(x^{11}+x^{2}+1)$
48-bit FIRE CODE	$(x^{13}+1)*(x^{35}+x^{23}+x^{8}+x^{2}+1)$
56-bit FIRE CODE	$(x^{22}+1)*(x^{11}+x^{7}+x^{6}+x+1)*$ $(x^{12}+x^{11}+\ldots+x^{2}+x+1)*$ $(x^{11}+x^{9}+x^{7}+x^{6}+x^{5}+x+1)$

COMPUTER GENERATED CODES

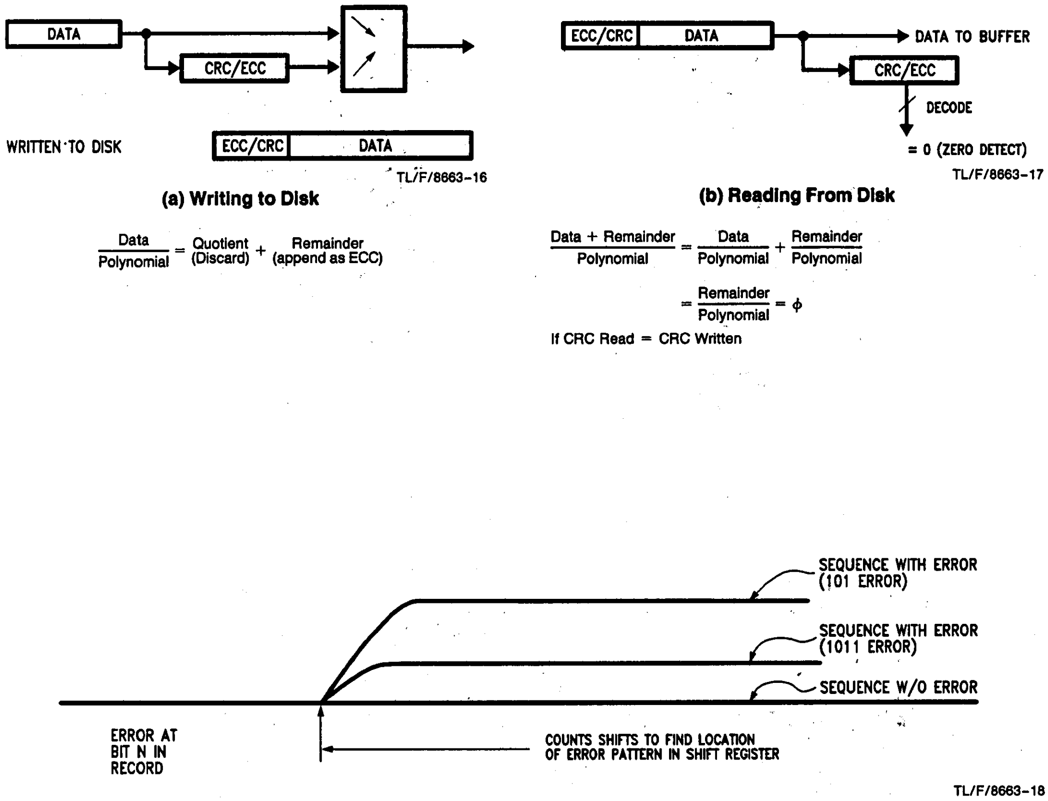
These have a very good reliability and are specifically chosen to guarantee not miscorrecting a specified worst case double burst error. The reliability can be calculated by the equation for miscorrection probability. National recommends the use of these codes for disk systems.

DOUBLE BURST REED SOLOMON CODES

Reed Solomon codes can handle longer bursts, multiple burst error (two burst error within a sector) correction capability and would be necessary for use with some optical media because of the high error rates. Typically RS codes for optical media are as long as a quarter of the sector. An example of the Reed Solomon code is given below.

$$(x + a^{5})*(x + a^{6})*(x + a^{7})*(x + a^{8})*(x + a^{9})$$

How Error Detection Works



TL/F/8663-16

(a) Writing to Disk

$$\frac{\text{Data}}{\text{Polynomial}} = \underset{\text{(Discard)}}{\text{Quotient}} + \underset{\text{(append as ECC)}}{\text{Remainder}}$$

TL/F/8663-17

(b) Reading From Disk

$$\frac{\text{Data + Remainder}}{\text{Polynomial}} = \frac{\text{Data}}{\text{Polynomial}} + \frac{\text{Remainder}}{\text{Polynomial}}$$

$$= \frac{\text{Remainder}}{\text{Polynomial}} = \phi$$

If CRC Read = CRC Written

TL/F/8663-18

(c) How Correction Works

FIGURE 1.8. ECC/CRC

CYCLIC REDUNDANCY CODE

These can only detect errors and will not correct. They are generally used for header appendage and in floppy drives. The most widely used code is the CRC-CCITT code given below.

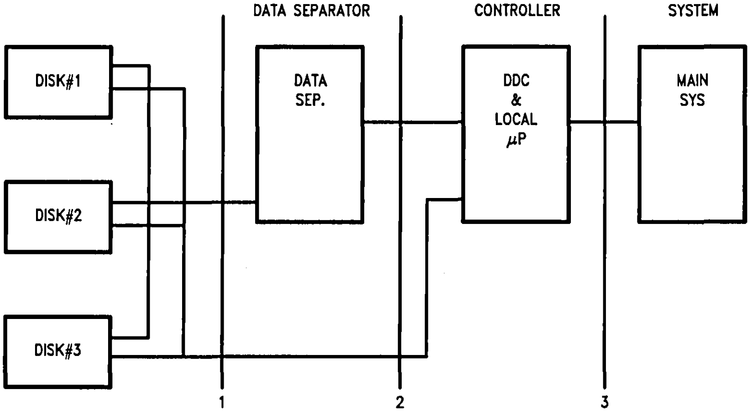
CRC-CCITT 16 bit $x^{16} + x^{12} + x^5 + 1$

Selecting the correct CRC/ECC is a function of the parameters being evaluated.

Detection Ability: The ability of the CRC/ECC to detect errors in the data transferred, measured as the number of bits affected and number of distinct bursts, the important measure being the guaranteed value.

Correction Ability: The ability of the ECC to restore erroneous data to its original state. Again, like the detection ability, this is measured as number of bits and number of bursts, the important measure being the guaranteed value.

Operating Environment: This involves factors like encoding scheme, data rate, data block size, technology on disk, product environment, and compatibility.



TL/F/8663-19

(a) CONCEPTUAL REPRESENTATION OF THE DISK SYSTEM WITH POTENTIAL INTERFACE POINTS. 1 = ST506/ST412; 2 = ESDI, SMD; 3 = SCSI, IPI

Interface	Year	Data Rate	Connectors	Drives	Status
SMD*	1975	≤15 Mb/s	60-Pin, 26-Pin	Hi Perf 8″, 14″	Upgrading Now
SA1000	1978	4.3 Mb/s	34-Pin, 20-Pin	Low Cost 8″	Limited Future
ST506	1980	5 Mb/s	34-Pin, 20-Pin	Most 5¼″, 3½″	Still Popular
ST412HP	1983	10 Mb/s	34-Pin, 20-Pin	Low Cost 8″	
ESDI*	1983	10–15 Mb/s	34-Pin, 20-Pin	Mid-Hi Perf 5¼″	New Standard
Future	1985-6	24 Mb/s	34-Pin, 20-Pin	Hi-Perf 5¼″, 8″	

*Data separator on the drive.

(b) POPULAR HARD DISK DRIVE INTERFACES

Interface	Year	Data Rate	Data Bus	CTL Bus	System	Status
SASI	1981	≤1–2 Mb/s	8-Bit + P	9-Bit	Low-End	Superseded by SCSI
SCSI Asynchronous	1982	≤1–2 Mb/s	8-Bit + P	9-Bit	Low/Mid-End	ANSI Standard
SCSI Synchronous	1984	≤4 Mb/s	8-Bit + P	9-Bit	Mid-End	Recent ANSI Standard
IPI-3	1984	≤10 Mb/s	8-Bit + P 16-Bit + 2P	6-Bit	High-End	Almost ANSI Standard

(c) POPULAR INTELLIGENT DISK SYSTEM INTERFACES

FIGURE 1.9. Drive Interface Standards

1.5 THE DISK DRIVE CONTROL PATH

The disk drive control path essentially consists of the various control signals defined by the drive interface for drive control and data path control. The control path in a disk system has a number of potential interface points, *Figure 1.9(a)*. The popular hard disk drive interfaces which define the physical connections of the controller with the drive are given in *Figure 1.9(b)*. These are at the interface points 1 and 2. At these points data is still in the serial format. With the advent of sophisticated controllers many Intelligent disk system interfaces have come into being, *Figure 1.9(c)*. These essentially incorporate the complete controller on the drive and interface to the outside world, through an 8- or 16-bit standard bus, interface specific. The physical interface with the disk is usually one of the standard hard disk drive interfaces mentioned in *Figure 1.9(b)*.

INTERFACE STANDARDS

Interface standards are the definition of the connection between parts of the disk unit, controller, and system. Interfaces can be defined on several levels, viz.

- Electrical specification of signal levels.
- Timing relationships between signal lines.
- Physical specification of cabling, connectors etc.
- Functional specifications of tasks the standard performs.
- Command descriptor specification of the standard.

Some interface standards require only a subset of the above definition categories. For example, there is no necessity for a command descriptor segment to the ST506 defacto standard, as no provisions are made for command communication other than the simple control lines described in the functional specifications.

Standards are important as they allow numerous manufacturers to cater to the same market segment, thus creating healthy competition. For example, the ST506 interface is an industry standard simply because it is being used by a lot of drive manufacturers. The factors which affect the choice of the standard are: data rate, flexibility, popularity, performance, and cost.

1.5.1 Popular Hard Disk Drive Interfaces

There are a number of disk interface standards. It is important to realize the implications of the various contenders for the interface point. For example, if the data separator is placed on the drive, the cost of the drive increases, the cost of the controller board decreases, the speed of the interface can increase if desired, but the system has to cope with this increase. Some of the most commonly used standards (disk interfaces) are discussed briefly in the following sections.

FLOPPY DISK INTERFACE

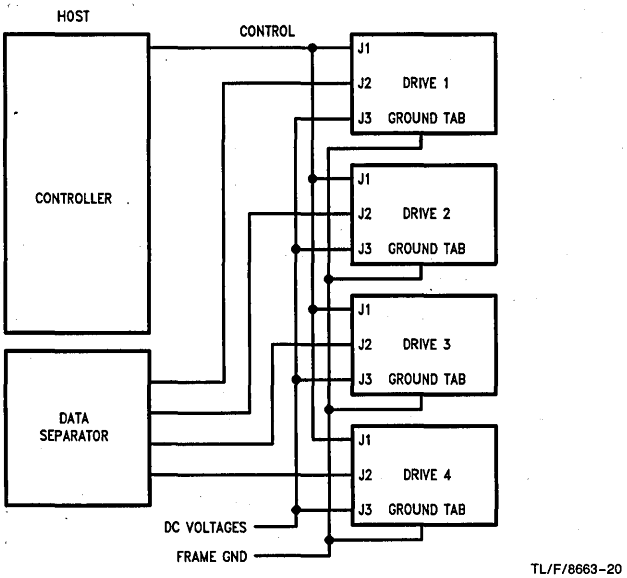
This is a relatively "dumb", single connector, serial data and control interface. There are two lines which carry the read/write data, and several control signals. This interface is positioned at point 1 in *Figure 1.9(a)*. The data rate for such interfaces is comparatively slow, around 100 to 500 kBits per second and the data capacity of floppy disks is not very large. The head is positioned by issuing step pulses to the drive. Read and write operations are initiated by asserting signals called Read Gate or Write Gate.

INTERFACE SIGNALS

Head load, Index, Sector (hard sectored drives only), Ready, Drive Select (usually 4), Step, Direction, Write Gate, Track 0, Write protect.

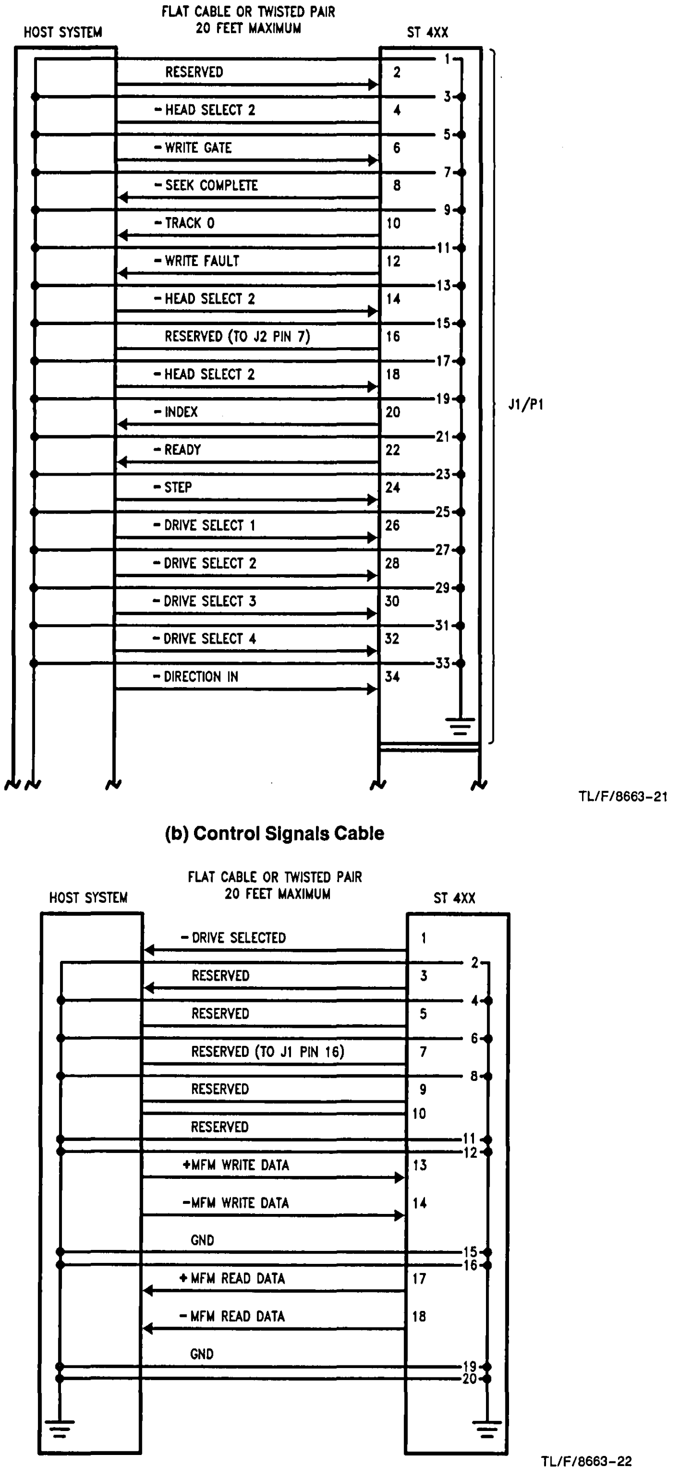
ST506/ST412 DISK INTERFACE STANDARD

This is also sometimes referred to as the floppy extension interface and is one of the most commonly used interface standards. The data rate is defined to be 5 Mbits per second, and the code is MFM. The interface is divided into two cables—a 34-pin control cable and a 20-pin data cable. The control cable allows for a daisy chain connection of up to four drives with only the last drive being terminated, *Figure 1.10*. The data cable must be attached in a radial configuration. This interface is at point 1 and, hence, the data separator is a part of the controller.



(a) Typical Connection, 4 Drive System

FIGURE 1.10. ST506/412 Configurations



(b) Control Signals Cable

(c) Data Signals Cable

FIGURE 1.10. ST506/412 Configurations (Continued)

Functional Operations

DRIVE SELECTION

Drive selection occurs when one of the DRIVE SELECT lines is activated. Only the selected drive will respond to the input signals, and only that drive's output signals are then gated to the controller interface.

TRACK ACCESSING

Read/Write head positioning is accomplished by:

a) Deactivating WRITE GATE line.

b) Activating the appropriate DRIVE SELECT line.

c) Being in the READY condition with SEEK COMPLETE true.

d) Selecting the appropriate direction.

e) Pulsing the STEP line.

Each step pulse will cause the head to move either one track in or one track out depending on the level of the direction line. A low level on the DIRECTION line will cause a seek inward toward the spindle, a high, outward toward track 0. Some drives have buffered seeks where the drive stores the pulses until the last one is received, then executes the seek as one continuous movement.

HEAD SELECTION

Any of the heads can be selected by placing the head's binary address on the Head Select lines.

READ OPERATION

Reading data from the disk is accomplished by:

a) Deactivating the WRITE GATE line.

b) Activating the appropriate DRIVE SELECT line.

c) Assuring the drive is READY.

d) Selecting the appropriate head.

WRITE OPERATION

Writing data onto the disk is accomplished by:

a) Activating the appropriate DRIVE SELECT line.

b) Assuring the drive is READY.

c) Selecting the proper head.

d) Insuring no WRITE FAULT conditions exist.

e) Activating WRITE GATE and placing data on WRITE DATA line.

Electrical Interface

The interface to the ST506/ST412 family can be separated into three categories, each of which is physically separated.

1. Control Signals.
2. Data Signals.
3. DC Power.

All control lines are single ended and digital in nature (open collector TTL) and either provide signals to the drive (input) or signals to the controller (output) via interface connection J1/P1. The data transfer signals are differential in nature and provide data either to (write) or from (read) the drive via J2/P2. *Figure 1.10* shows the connector pin assignments for this interface.

Since the data separator is on the controller, the ST506/ST412 drive must have a transfer rate of 5 M bit/sec. The bit density cannot be increased as the data rate and disc rotational speed are fixed. The only way to increase drive capacity is to increase the number of tracks, which does not allow large increases of capacity. Despite this limitation, it has a strong future as it moves into lower cost systems and smaller 3½″ drives.

ST412 HP INTERFACE

This standard was designed to provide an upgrade path from the ST506 and is very similar. This interface is also at point 1. The main differences from the ST506 family are:

- One additional control line in the daisy chain . . . Recovery mode.
- Reduced write current is not part of the interface.
- The data rate is 10 Mbits/sec.
- The encoding scheme is not tightly specified, but suggested to be MFM.
- The maximum repetition rate of step pulses has been increased.

The major benefit of this interface is the higher data rate compared to ST506 drives, however, a much more careful design is needed to keep the bit error rate the same and for this reason may not be popular. Since the data separator is located on the controller, the data transfer rate must be exactly the 10 Mbits/sec rate and still be MFM encoded.

Recovery Mode

Recovery mode has been added in response to higher track density. It is asserted by the controller in response to bad data. In this mode the controller issues up to eight step pulses, and the drive steps through its own micropositioning algorithm. After each pulse, the controller tries to reread data and, if it fails again, after the eighth try it abandons the procedure. This drive interface emerged as higher data rate embellishment to the ST412.

ESDI (ENHANCED SMALL DEVICE INTERFACE)

This interface is at point 2 on *Figure 1.9*. This standard was a proposal by Maxtor Corporation, subsequently modified by an experienced working committee, and is finding growing acceptance largely because it is a sensible proposal. It has control and data cables like the ST506/412 interface but adds a driver and receiver on the data cable for the clock information, as shown in *Figure 1.11*. The implication is that the data separator resides on the drive, which means fewer design problems for drive users, and that certain status and command information is transmitted in serial, which means more control circuitry on both sides of the interface. The data rate is allowed to be several frequencies, dependent upon options, with the maximum rate probably reaching 24 Mbits/sec.

Features

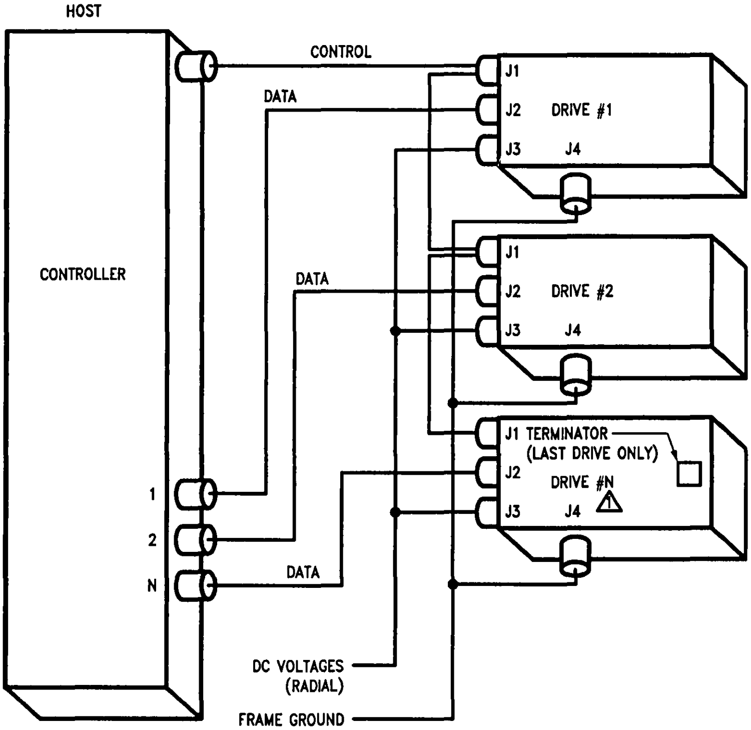
- Low cost, high performance interface suitable for smaller, high performance drives.
- Two protocols supported: serial and step mode.
- Supports up to 7 drives in the serial mode and 3 drives in the step mode.
- Maximum data rate of 24 Mbits/sec.
- Supports cable lengths of up to 3 meters.
- Serial mode of operation utilizes NRZ data transfer along with serial commands and serial configuration and status reporting across the command cable.

- Step mode implementation utilizes the same NRZ data transfer; however, the step and direction lines are used to cause actuator motion. Hence, with this mode configuration and status reporting are unavailable over the interface.

The ESDI interface puts the data separator on the drive and its output is NRZ data with a synchronous clock. This results in the data rate, and therefore the bit density, not being rigidly defined. The controller speed is governed by the synchronous clock coming from the drive, not from a data separator as in the ST412/ST506 interfaces. The drive is code independent as the data across the interface is always NRZ (or decoded) format. This enables the use of codes like RLL which put more data on the disk for the same bit density (flux reversals per inch). Moreover the use of NRZ encoding results in decreased errors due to electrical transients on the interface cable. This lowers practical bit error rates and allows the use of higher speeds.

Step Mode

The ESDI step mode is essentially similar to the step mode in the ST506/412 family of drives, except for the NRZ data transfer. Only two of the seventeen signals change function in the control cable between ESDI step and ST412HP. READ GATE being added is the important change which enables the data separator on the drive to the controller. The data cable is considerably different. Differential drivers and receivers are used for signals like Write Clock and Read Clock and a few single ended lines are added like Cartridge Changed (for tapes) and other lines like Seek Complete, Index etc. The step mode pulse timings are comparable to that of the ST412HP. This enables switching between the two interfaces under software control, in the controller design.

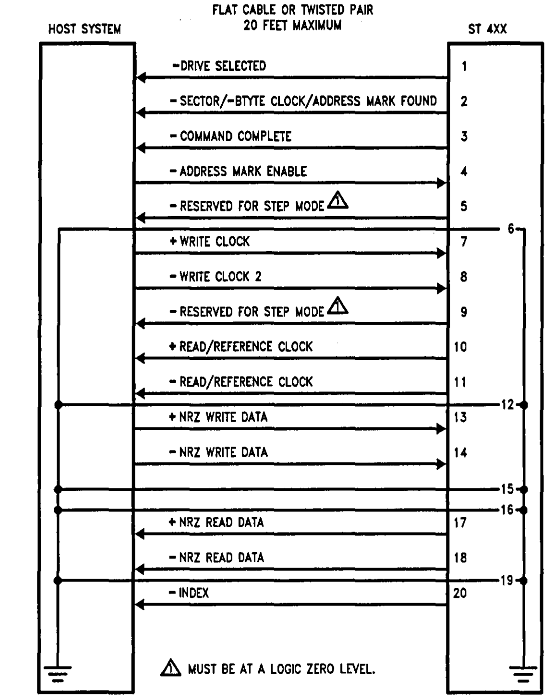


△1 NOTE: IN STEP MODE, MAXIMUM NUMBER OF DRIVES = 3
IN SERIAL MODE, MAXIMUM NUMBER OF DRIVES = 7

TL/F/8663-24

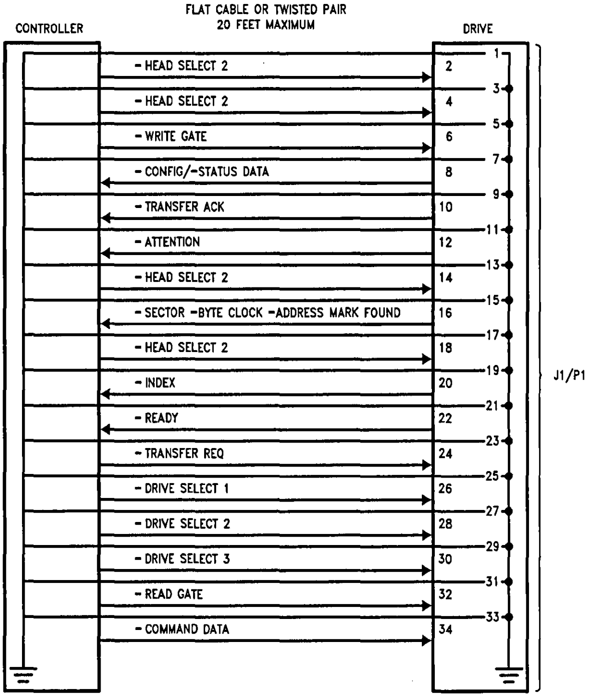
(a) Typical Connection, Multiple Drive System

FIGURE 1.11. ESDI (Enhanced Small Device Interface)



TL/F/8663-23

(b) Data Cable (J2/P2) Signals (Disk Implementation—Serial Mode)



TL/F/8663-25

(c) Control Cable (J1/P1) Signals (Disk Implementation—Serial Mode)

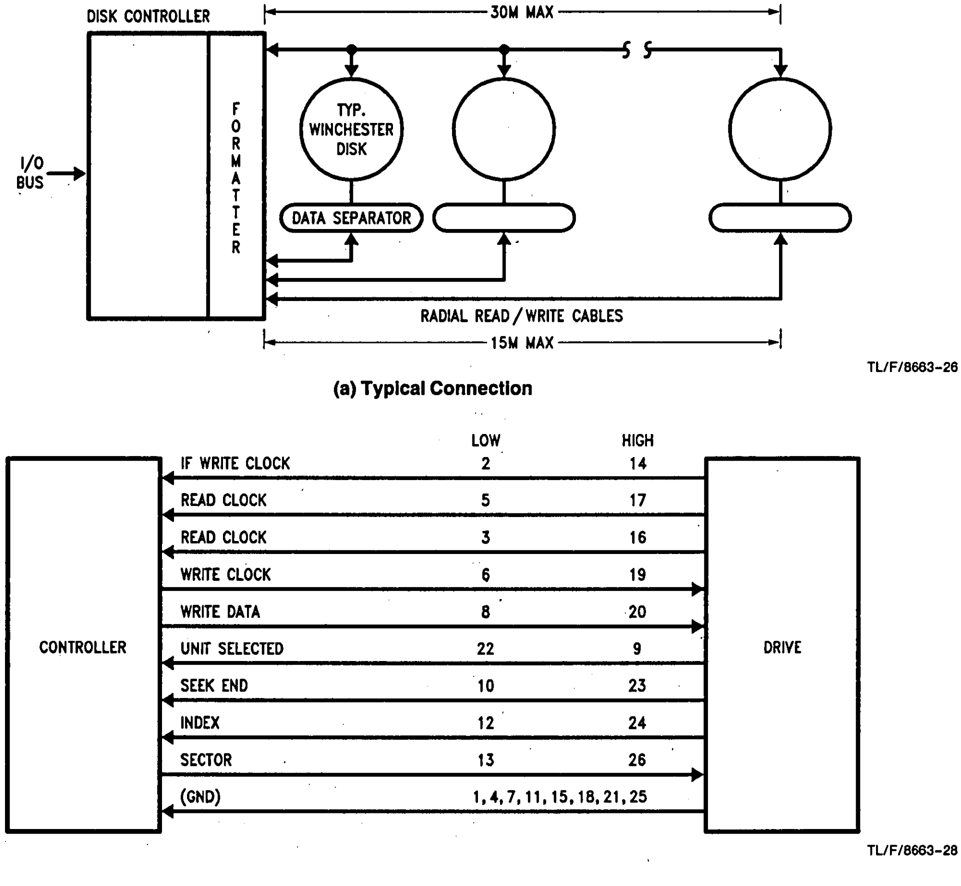
FIGURE 1.11. ESDI (Enhanced Small Device Interface) (Continued)

Serial Mode

Serial mode ESDI is a definite improvement over the interfaces discussed. As the name implies, communication from the controller to the drive takes place on the COMMAND DATA line of J1/P1 in conjunction with the handshake signals TRANSFER REQUEST and TRANSFER ACKNOWLEDGE. Communication from the drive to the controller takes place on the CONFIG-STATUS line of J1/P1 in conjunction with the handshake signals. Each bit of the 16-bit command or status word is handshaked across the interface. The hardware changes between EDSI serial and step modes, have several control lines redefined. The disk drive's microprocessor interprets commands like SEEK (seek to a cylinder), RECALIBRATE (seek to track 0), REQUEST STATUS and REQUEST CONFIGURATION, which provide the controller with standard status and configuration information of the drive like the number of heads, number of tracks, sectors per track, bytes per track, command data parity fault, write fault etc. Hence the controller can configure itself to the drive connected to it and can send the data to the host if desired. Thus ESDI serial mode offers big benefits and is rapidly gaining popularity in higher performance hard disk drives.

STORAGE MODULE (SMD) INTERFACE (ANS X3.91M 1982)

The Storage Module Interface was originated by Control Data Corporation around 1972. It has been extremely popular with 8″–14″ drives. However, as it is expensive and hardware intensive and because of competition due to ESDI and SCSI, it is not very popular with 5¼″ drives. *Figure 1.12* gives the data and control cable assignment.



TL/F/8663-26

(a) Typical Connection

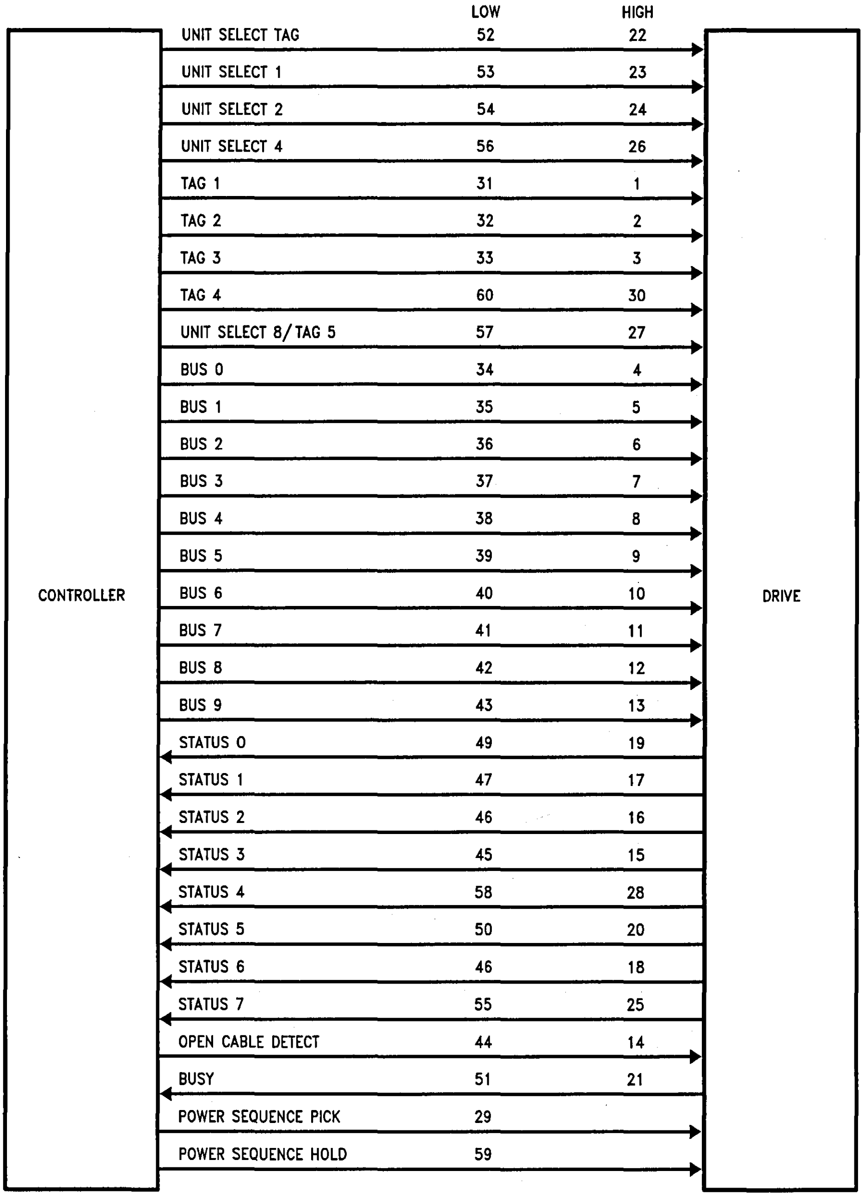
	LOW	HIGH
IF WRITE CLOCK	2	14
READ CLOCK	5	17
READ CLOCK	3	16
WRITE CLOCK	6	19
WRITE DATA	8	20
UNIT SELECTED	22	9
SEEK END	10	23
INDEX	12	24
SECTOR	13	26
(GND)	1, 4, 7, 11, 15, 18, 21, 25	

TL/F/8663-28

B = Cable

(b) Data Cable

FIGURE 1.12. Storage Module (SMD) Interface (ANS X3.91M 1982)



	LOW	HIGH
UNIT SELECT TAG	52	22
UNIT SELECT 1	53	23
UNIT SELECT 2	54	24
UNIT SELECT 4	56	26
TAG 1	31	1
TAG 2	32	2
TAG 3	33	3
TAG 4	60	30
UNIT SELECT 8/TAG 5	57	27
BUS 0	34	4
BUS 1	35	5
BUS 2	36	6
BUS 3	37	7
BUS 4	38	8
BUS 5	39	9
BUS 6	40	10
BUS 7	41	11
BUS 8	42	12
BUS 9	43	13
STATUS 0	49	19
STATUS 1	47	17
STATUS 2	46	16
STATUS 3	45	15
STATUS 4	58	28
STATUS 5	50	20
STATUS 6	46	18
STATUS 7	55	25
OPEN CABLE DETECT	44	14
BUSY	51	21
POWER SEQUENCE PICK	29	
POWER SEQUENCE HOLD	59	

TL/F/8663-27

A = Cable

(c) Control Cable

Storage Module (SMD) Interface (ANS X3.91M 1982) (Continued)

Features

- Bit serial digital data transfer (Data Separator in drive)
- Relatively high transfer rate (9.67 Mbits/s is common in older 14″ drives and newer 8″ drives, new 10.5″ and 14″ drives are typically about 15 Mbits/s)
- Dominant de-facto standard for 14″ OEM disk drives; virtual basis of OEM disk controller industry. Widely used by minicomputer system manufacturers.
- Differential signals
- 23 required plus 8 optional control bus signals, 7 required plus 2 optional read/write cables
- Parallel control bus, but radial read/write cables, one per drive
- Incorporates error recovery facilities
- Includes power sequencing for multiple units
- Approved ANS X3.91 1982

1.5.2 Intelligent Disk System Interfaces

These are high level interfaces which result in the complete disk controller being situated on the drive and the interface to the host is through a special bus. Their chief advantage is nearly complete device transparency to system hardware and software, also lower system overhead for disk control and higher speeds. A well defined protocol is used for communication with the host system. Some of the popular Intelligent disk system interfaces are discussed below in brief.

SHUGART ASSOCIATES SYSTEM INTERFACE (SASI)

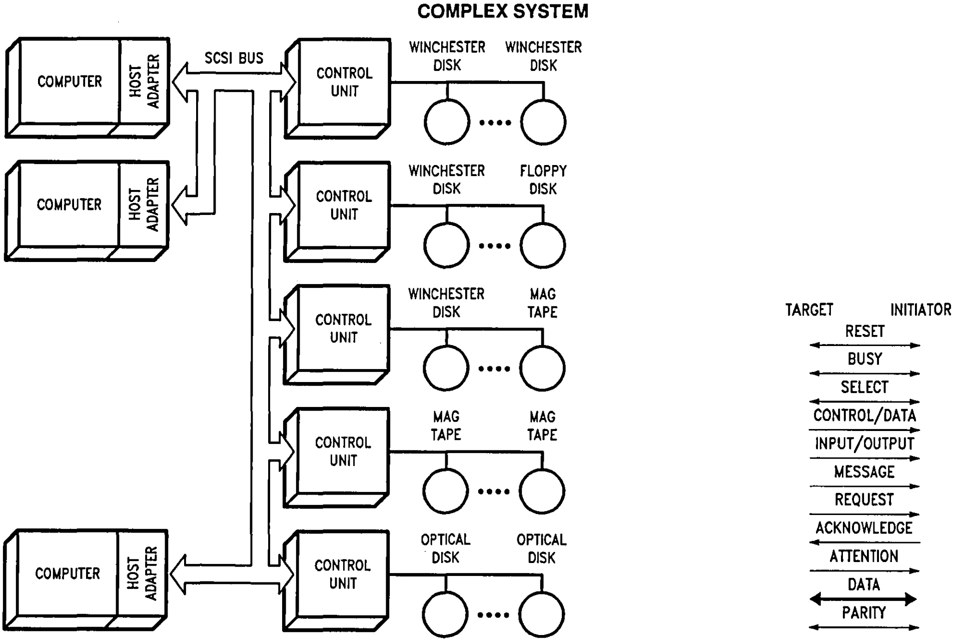
SASI was introduced by Shugart around 1980. The overall objective was to make it easier for computer systems to talk to disk drives. SASI defines the logical level and all lower interface levels, down to the 50-pin connectors and ribbon cable. Eighteen lines are used for signals, nine for data and nine for control. The data lines consist of a single octet with an odd parity bit. The control lines include a two-wire handshake and various lines to put the bus in different transfer modes or phases. The interface is always in one of the five phases:

Bus Free, Arbitration, Selection, Reselection, Data

Eight devices are allowed but only one host or "initiator" is allowed. So a maximum SASI system will consist of an Initiator and seven target devices. All signals in the interface are open collector driven. The ANSI standard version of this interface is the SCSI (Small Computer System Interface).

SMALL COMPUTER SYSTEM INTERFACE (SCSI)

The Small Computer System Interface (SCSI) was formed from the SASI framework and ANSI has standardized it under X3T9.2. The interface consists of a single cable that is daisy-chained to other SCSI units. It will accommodate not only disk drives but also tapes, printers and other devices and is potentially a universal peripheral port for small systems. The SCSI system could potentially be a single initiator - single target system or a single initiator - multiple target system or a multiple initiator - multiple target system as shown in *Figure 1.13(a)*. The SCSI bus signals are shown in *Figure 1.13(b)*. The cable consists of transfer handshaking and status signals in addition to an 8-bit data bus. Information is exchanged on the bus via a set of higher level commands sent by the host.



TL/F/8663-29

TL/F/8663-30

Up to 8 SCSI DEVICES can be supported by the SCSI bus. They can be any combination of host CPUs and intelligent controllers.

(a) Multiple Target-Multiple Initiator

(b) SCSI Cable

FIGURE 1.13. Small-Computer System Interface

Features

- Connects up to 8 computers and peripheral controllers
- Maximum rate up to 1.2 Mbyte/sec asynchronous, 4 Mbytes/sec synchronous: suitable for floppy disks, all 5.25″ and 8″ Winchester disks, medium performance 8″ and 14″ disks, and tape drives
- Relatively high level peripheral command set
- Single ended version: 50-conductor flat ribbon cable, up to 6 meters, 48 mA drivers
- Differential version: 50-conductor flat or twisted pair cable, up to 25 meters, EIA RS-485
- Distributed bus arbitration
- Includes command sets for common peripherals
- Products now widely available include: disk drives with integral controllers, SCSI to ST506, SMD, Floppy, SCSI to S-100, Multibus®, IBM PC™, VME™, Unibus™, TRS-80™, and Q-BUS™ Adapters, and VLSI bus protocol and disk controller chips

SCSI makes no hardware changes compared to the SASI but adds several features which are discussed below.

Arbitration

This allows multiple Initiators to talk to multiple targets in any order, to a maximum of eight nodes only.

Reselection

This allows a target to disconnect from the Bus while it is getting data and reconnect to the proper Initiator when it has found it. This results in efficient utilization of the bus because other nodes can use it during the relatively long seek time of the disk drive or search time of a tape drive.

Synchronous Mode Transfer

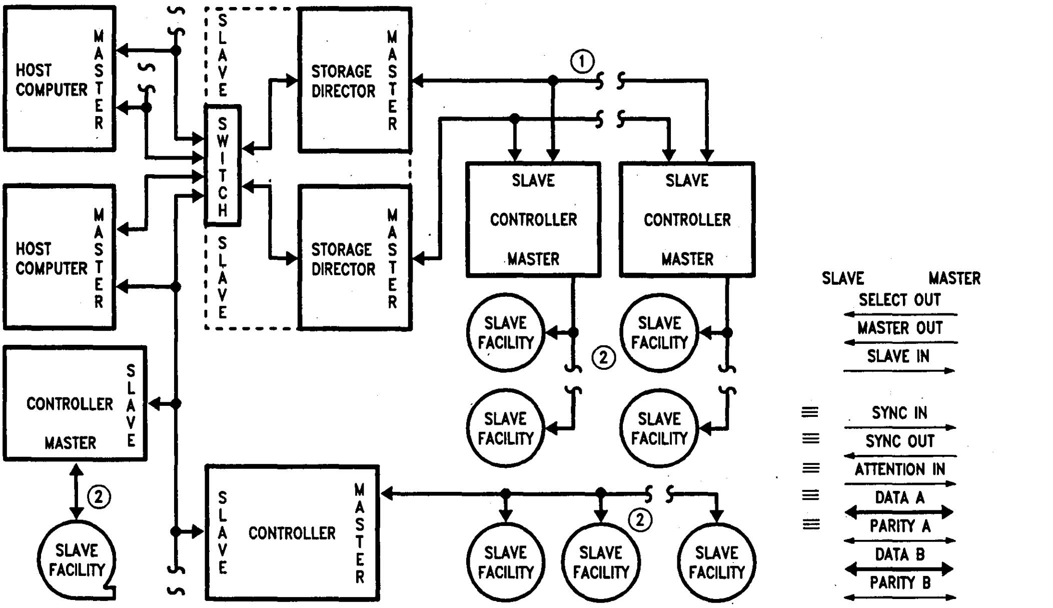
This speeds data transfer to a maximum of 4 Mbytes/sec (from an asynchronous maximum of approximately 1.2 Mbytes/sec).

Differential Transceivers

These boost the maximum length of the interface from 6m to 25m.

Extended Command Set

This set includes expanded large block addressing (from 2^{21} to 2^{32} blocks) and "Inquiry" type commands for self configuring controllers. It also handles tape drives, printers, processors, optical disks and read-only optical disks. There is also room for a "vendor unique" command set, i.e. commands unique to the device.



TL/F/8663-31

(a) IPI Configuration

TL/F/8663-32

(b) IPI Cable Signals

FIGURE 1.14. Intelligent Peripheral Interface

INTELLIGENT PERIPHERAL INTERFACE(IPI)

This is the ANSI standard X3T9.3 and is an additional peripheral bus, having higher performance than SCSI. *Figure 1.14(a)* shows the orientation of the IPI system and the IPI port signals are shown in *Figure 1.14(b).*

Features

- Connects master to a maximum of 8 slaves
- Can be used at various levels in the system as shown
- Two 8-bit buses (in and out) for commands and status speed protocol and status presentation for fast path switching
- 8- or 16-bit parallel transfers
- 24 signals
- Several electrical options; fastest allows 10 Mbytes/sec through a 75 meter cable
- Offers both "intelligent" and "device level" command definitions, command and data handshaking
- 50-pin cable ground increases noise immunity

The IPI interface comprises four levels. Level 0 consists of cables, connectors and drivers/receivers. Level 1 consists of state machine and bus protocol. Features of Level 2 are device specific commands, timing critical, physical addressing, physical volumes, command parameters and bus control commands. Features of Level 3 are device generic commands, timing independent, buffered, command stacking, queing, limited specific commands, logical addressing and physical volumes. Messages are transmitted in packets.

IPI derives its higher performance from a faster handshake and a wider data bus. Two octets, each with a parity line, make up the data interface. Six control lines fill out the interface of 24 signals. There is one master allowed and up to eight slaves on a daisy-chained cable. This master to slave interface is a parallel one and hence IPI 3 could be used, (point 1) in *Figure 1.14(a).* Each Slave can address up to 16 Facilities, like disk drives. The Slave-to-Facility interface may be IPI 2, (point 2) in *Figure 1.13(a)*, or a lower level interface such as ESDI. Data can be moved at 5 Mbytes/sec in asynchronous mode, 10 Mbytes/sec in synchronous mode. The interface supports various driver options with maximum cable lengths ranging from 5 meters to 125 meters.

1.5.3 Other Disk Interfaces

There are many other ANSI standardized interfaces which were the outcome of the interface standards discussed above. Some of these are disk level while some are intelligent interfaces. A brief discussion follows, also refer to *Figure 1.15.* Detailed descriptions can be found in the appropriate ANSI document.

FLEXIBLE DISK INTERFACE (ANS X3.80)

Features

- American National Standard Interface between flexible disk cartridge drives and their host controllers
- 50-wire flat ribbon cable (8″ disk) or 34-wire (5¼″ disk)
- Bit serial encoded FM or MFM data transfer from/to Read/Write electronics (data separator in controller)
- Modest data transfer rate (100 kbyte/s)
- Very widely used by the industry; based on a de facto standard. Supported by most 8″ and 5¼″ drives; also used by some, but not all micros (less than 4″) floppy drives.
- Seeks track by track, one step per pulse
- Single ended signals
- Change has been submitted to identify high density 5¼″ drives
- Approved ANS X3.80 - 1981

RIGID DISK INTERFACE (ANS X3.101—1983)

With the emergence of the 8″ rigid-disk drive, there was a strong industry push for a new interface standard with broader applicability than the SMD, which would allow for self-applicability than the SMD, which would allow for self-reconfiguring controllers, as different devices were attached. As a result the ANSI Rigid Disk Interface came into existence.

Features

- Optimized for relatively high performance small winchester disks
- Bit serial digital data transfer (data separator in drive)
- Relatively high transfer rate possible (up to 10 Mbits/s with low cost option, up to 16 Mbits/s with high performance option)
- 50 conductor ribbon cable
- Class A: 24 and 40 mA single-ended control bus signals, 20 mA differential serial data and clock lines
- Class B: 100 mA single-ended control bus signals, 40 mA differential serial data and clock lines
- 22 single-ended plus 4 differential signals
- Byte parallel command bus
- Relatively high level command set
- Approved ANS X3.101—1983

Peripheral Bus Interface

- Connects computer to peripheral different controllers
- Block transfer rather than word transfer orientation
- No provision for memory address on bus
- Longer distances than backplane
- Interface hides many device characteristics from software
- Peer to peer multi-master protocol may be called a "system bus"

Device Interface

- Specific to particular device type
- Between controller and device
- Often serial data transfer
- User device interchangeability not always certain
- Very widely used by industry

This interface has not gained acceptance and has been superseded by ESDI

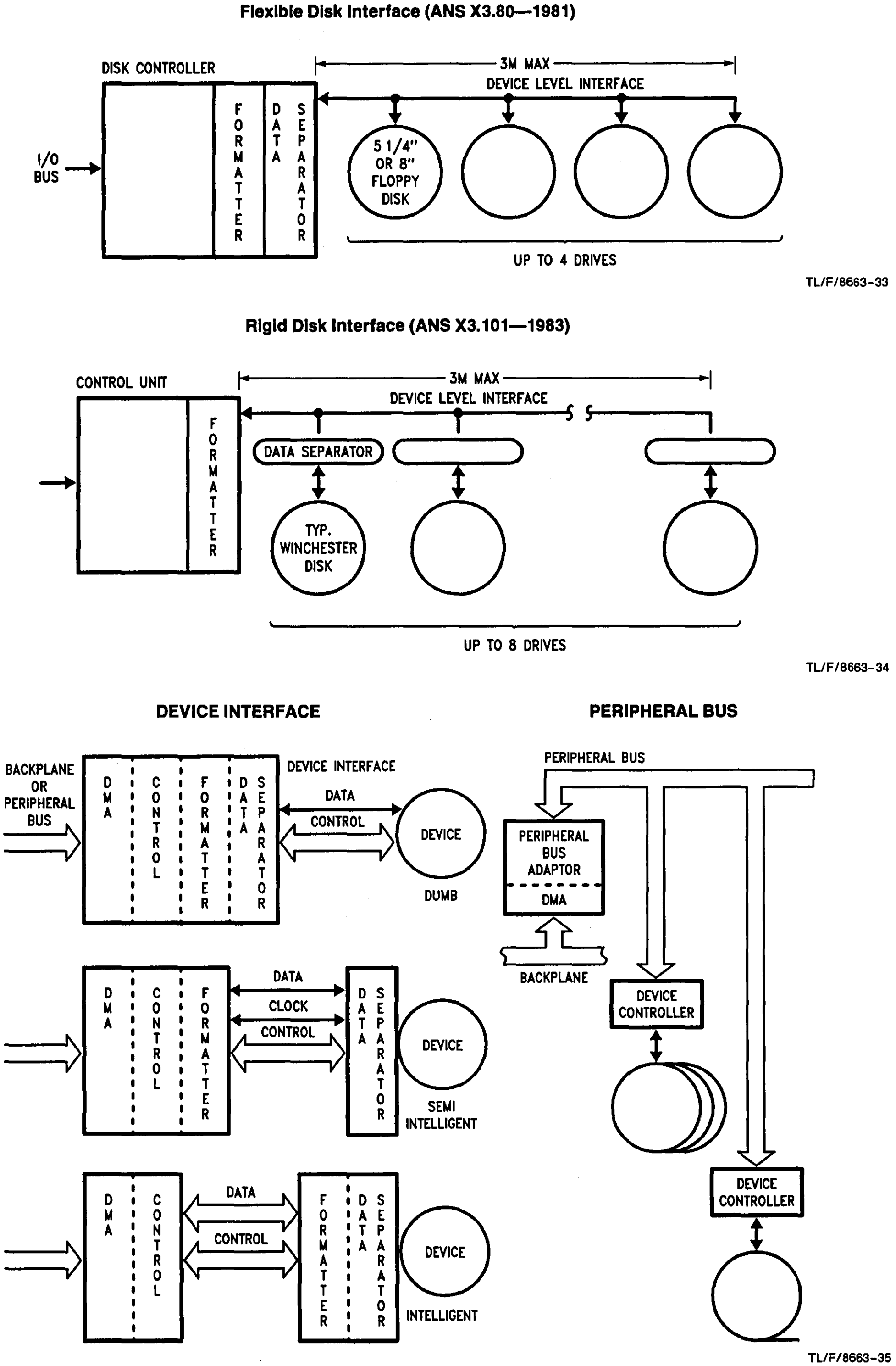


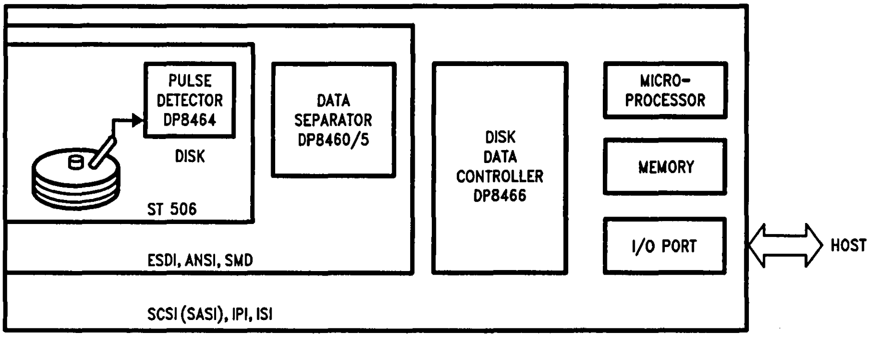
FIGURE 1.15. Other Drive Interfaces—Typical Connections

1.5.4 Universal Plug

Although the concept of a standard interface is appealing, the desire to gain industry recognition and lock in customers, coupled with valid technical improvements, will continue to spawn new interfaces at regular intervals. In systems where a maximum of one or two drives are required or where configuration flexibility justifies the higher cost of standard attachments, high-level interfaces will attach to a "universal" system plug. Here controllers will be integrated into the devices themselves, while in cost effective multiple drive systems, the connection of device level interfaces to system specific controllers is expected to continue.

1.6 ELEMENTS OF DISK CONTROLLING ELECTRONICS

Disk controller chips in the market today are complex VLSI chips and perform a multitude of functions. In fact they take care of most of the tasks besides the task of data separation. National's Disk Data Controller DP8466 is one such chip. It takes care of serialization, deserialization, data encoding, DMA transfer, error detection and correction, and pattern recognition to determine type of compensation required. The Disk Data Controller DP8466 is discussed in detail in the following sections. National's Data Separator chip DP8465, together with the Disk Pulse Detector DP8464, comprise the chip set for the disk controlling electronics. If RLL encoding is used, then National's DP8463 2,7 ENDEC could be integrated into the system. *Figure 1.16* shows the place of these chips in the disk data and control path. It also shows the separation lines of the components on the drive and controller for the various interface standards.



TL/F/8663-36

FIGURE 1.16. Typical Disk Controller System Configuration Showing the Interface Points with Respect to the Various Standards

CHAPTER 2 DP8464B HARD DISK PULSE DETECTOR

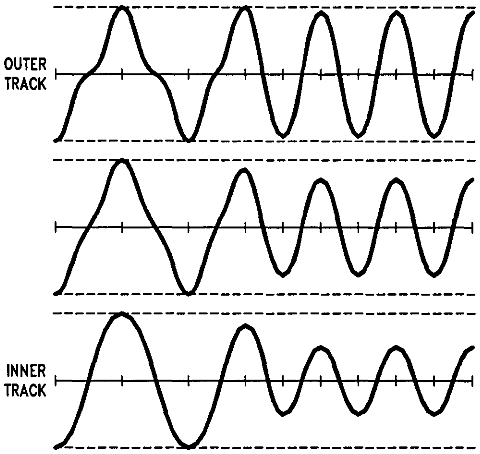
2.0 INTRODUCTION

The standard Winchester disk drive available today uses a magnetic film platter, a ferrite head and MFM coding. This combination produces relatively wide pulses off the disk which can be detected using a simple time-domain filter technique. However, as disk manufacturers strive for higher density and data rates, they must turn to new technologies such as plated media, thin film heads and run length limited codes (such as the 2,7 code). Unfortunately, these technologies produce more complex pulses off the disk which require more sophisticated pulse detection techniques. The DP8464B utilizes a separate time and gate channel which can detect the peaks in these complex waveforms.

2.1 BACKGROUND OF PULSE WAVEFORM DETECTION

Data on the disk is stored as a series of magnetic domains recorded on concentric circular tracks. To read the data, the head arm assembly brings the head directly above the track on the rotating disk. As previously recorded flux reversals pass under the head, a small signal will be induced. The signal from the disk is therefore a series of pulses, each of which are caused by flux reversals on the magnetic medium. The pulse detector must accurately replicate the time position of the peaks of these pulses. This task is complicated by variable pulse amplitudes depending on the media type, head position, head type and the gain of the Read/Write amplifier. Pulse amplitudes may vary on any one track if the distance between the head and the media varies as the disk rotates. Additionally, as the bit density on the disk increases, significant bit interaction occurs resulting in decreased amplitude, pulse distortion and peak shift.

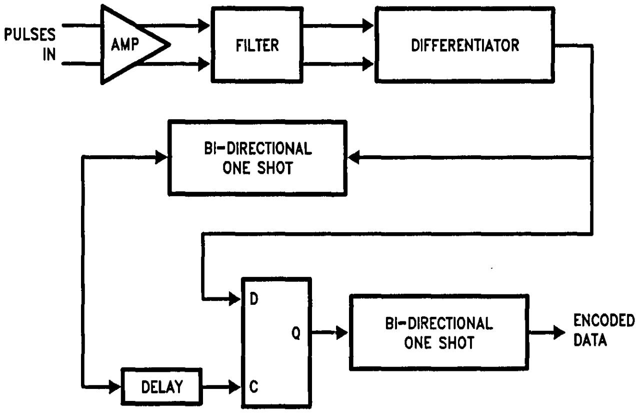
Traditionally MFM code has been used to encode digital information on the disk surface. MFM code uses the limited frequency range of F to 2F as illustrated in *Figure 2.1.* Such a system can use a special self gating circuit for the pulse detector as shown in *Figure 2.2.* Pulses from the inner track (the bottom waveform in *Figure 2.1*) are almost sinusoidal so the peak detector can simply differentiate the waveform to determine the peak positions. On the outer track however, the pulses have a small amount of shouldering as shown in the top waveform in *Figure 2.1.* The problem is that any noise occurring during these very narrow shoulders can be incorrectly interpreted as signal peaks.



TL/F/8663-37

Typical MFM waveforms on oxide media with ferrite heads. Since there is only slight shouldering on the outer track, the traditional self gating (de-snaker) pulse detector can be used.

FIGURE 2.1



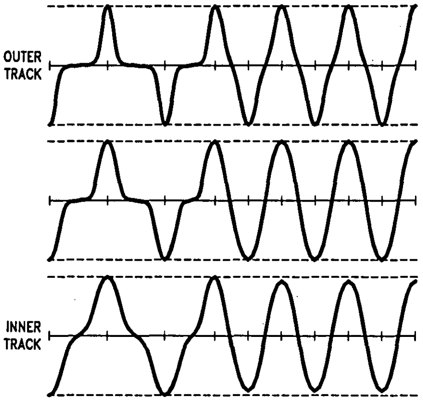
TL/F/8663-38

Self gating circuit (de-snaker) traditionally used for pulse detection of MFM code on oxide media with ferrite heads. The amount of delay in the clock line must exceed the maximum amount of shouldering.

FIGURE 2.2

The self gating circuit places a fixed delay and bi-directional one shot between the output of the differentiator and the clock input. The amount of this delay is selected to be longer than the worst case shouldering. When shoulder-induced noise occurs, the D input to the flip-flop will change states. However, by the time the clock occurs, the noise will no longer be present and the D input will have returned to its previous state. The flip-flop will therefore "clock" in the previous data. The output Q will not change states due to these narrow shoulder-induced noise pulses. This fixed delay in the clock line is called a time domain filter. This circuit is also known as a "de-snaker", named after the snake appearance of the waveform which exhibits slight shouldering. This "de-snaker" circuit works very well with waveforms which exhibit only slight shouldering. Unfortunately, the new methods used to increase the disk capacity do not produce such simple pulse patterns.

There are several methods of increasing the disk capacity. This includes plated media, thin film heads, and run length limited codes. All of these techniques result in narrow pulses with increased shouldering. An example of these slimmer pulses is shown in *Figure 2.3*. Instead of the slight shouldering present in the MFM example, the signal returns to the baseline between pulses. Since the shouldering is so extensive, the "de-snaker" technique simply will not work here. If a long delay were used to correct the shouldering present in the top waveform, it would not capture the pulses at the highest frequency.



TL/F/8663-39

Pulse waveforms for high resolution technologies. The large shouldering on the outer tracks precludes the use of the de-snaker pulse detector.

FIGURE 2.3

Detecting pulse peaks of waveforms of such variable characteristics requires a means of separating both noise and shouldering-induced errors from the true peaks. The old self gating circuits (such as the "de-snaker") will not work with the new techniques to increase the disk capacity. Hence the need for a circuit that includes a peak sensing circuit with an amplitude sensitive gating channel in parallel. Such a circuit is a key feature of the DP8464B Pulse Detector.

2.2 DP8464B FEATURES

Certainly a key feature of the DP8464B is the combination of a peak sensing circuit with an amplitude sensitive gating channel in parallel which allows the DP8464B to accurately detect the peak of waveforms that preclude the use of the traditional "de-snaker" circuit. The DP8464B, however, has many other features that make it ideal for the disk drive read channel.

Another key feature of the DP8464B is a wide bandwidth automatic gain controlled (AGC) amplifier. The automatic gain control removes the signal level variations of the read signal. The amplifier's wide bandwidth (20 MHz) insures that timing errors will not be introduced by the amplifier's pole.

The DP8464B offers considerable flexibility to the user, allowing him to tailor various operating characteristics to his specific needs. In particular, the user can set the frequency response of the differentiator, the width of the pulses on the encoded data output, the amount of hysteresis in the gating channel, the signal amplitude at the output of the gain controlled amplifier and the overall frequency response of the system and AGC. This kind of flexibility is provided by strategically placing pinouts at key points throughout the circuit. Differential signal paths were utilized whenever possible to minimize effects of power supply noise and external noise pickup. The IC can be effectively disabled when the disk drive is in a write mode, thus preventing saturation of the input amplifier and preventing disturbance of the AGC level.

The IC is powered from a single +12V supply (which is standard in most drives) and has an internal regulator and separate analog and digital grounds in order to properly isolate the sensitive analog circuitry. It is presently offered in a 24-pin DIP but will soon be made available in a 28-pin PCC surface mount package.

The DP8464B is fabricated on an advanced low power Schottky process which allows the part to handle data rates up to 15 Megabits/sec., 2, 7 Code.

2.3 THE DP8464B HARD DISK PULSE DETECTOR OPERATION

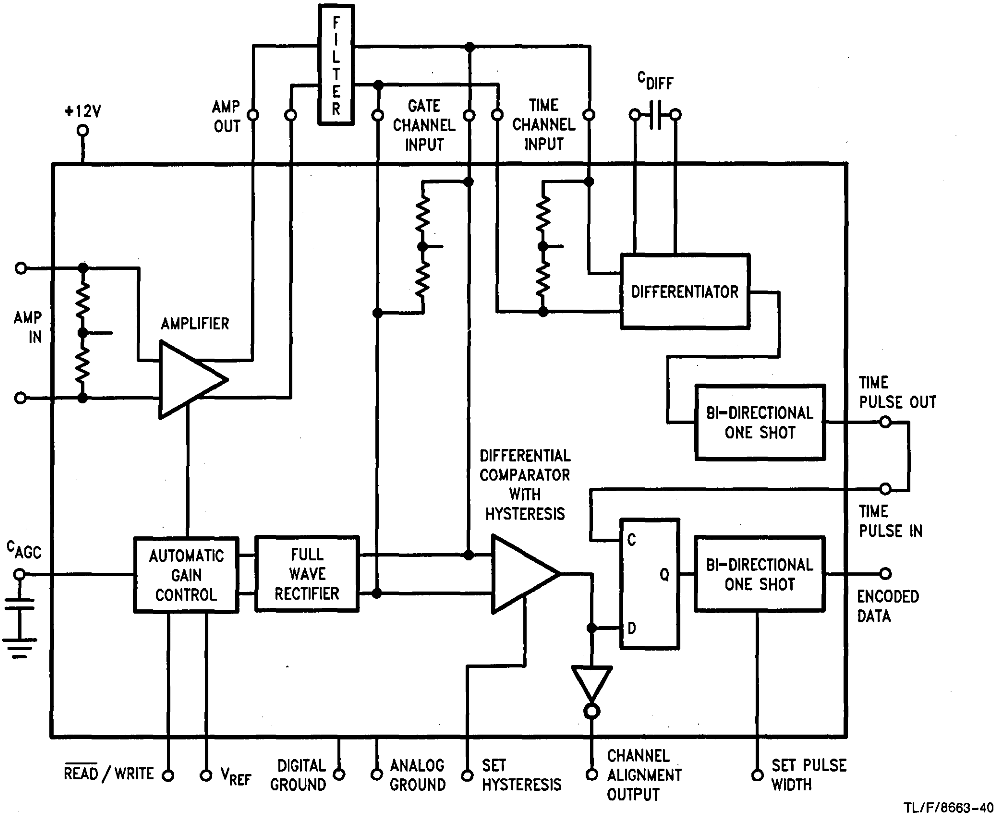
The main circuit blocks of the DP8464B are shown in *Figure 2.4*. The circuit consists of three main sections: the Amplifier, the Time Channel and the Gate Channel. The Amplifier section consists of a wide bandwidth amplifier, a full wave rectifier and the Automatic Gain Control (AGC). The Time Channel is the differentiator and its associated bi-directional one shot, while the Gate Channel is made from the Differential Comparator with Hysteresis, the D flip-flop and its following bi-directional one shot. Also, there is special circuitry for the Write mode. To better understand the circuit operation, let's discuss each section separately.

2.3.1 Gain Controlled Amplifier

The purpose of the Amplifier is to increase the differential input signal to a fixed amplitude while maintaining the exact shape of the input waveform. The Amplifier is designed to accept input signals from 20 mV_{pp} to 660 mV_{pp} differential and amplify the signal to 4 V_{pp} differential. The gain is therefore from 6 to 200 and is controlled by the Automatic Gain Control (AGC) loop.

2.3.2 Time Channel

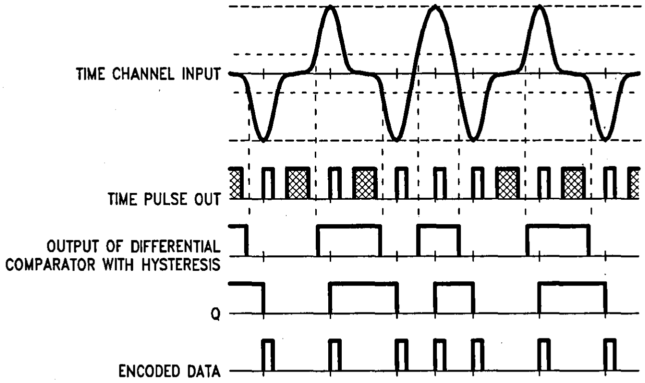
The peak detection is performed by feeding the output of the Amplifier through an external filter to the Differentiator. The Differentiator output changes state when the input pulse changes direction, generally this will be at the peaks. The Differentiator can also respond to noise near the baseline, in which case the Gating Channel will inhibit the output pulse (as discussed in the Gate Channel section). The purpose of the external filter is to bandwidth limit the incoming signal for noise considerations. Care must be used in the design of this filter to ensure the filter delay is not a function of frequency. The output of the Differentiator drives a bi-directional one shot which creates the Time Pulse Out.



TL/F/8663-40

Main circuit blocks of DP8464B. The three main sections are the Amplifier (amplifier and AGC), the Gating Channel (comparator with hysteresis and D flip-flop), and the Time Channel (differentiator and bi-directional one shot).

FIGURE 2.4



TL/F/8663-41

These signal waveforms illustrate the operation of the DP8464B. The noise in the Time Pulse Out (which occurs during the shouldering) simply clocks in old data present at the output of the Differential Comparator with Hysteresis. A bi-directional one shot at the Encoded Data output provides a rising edge representing the relative time position of the peaks at the Time Channel Input.

FIGURE 2.5

2.3.3 Gate Channel

While the actual peak detection is done in the Time Channel with the Differentiator, there is the problem of preventing the output data from being contaminated when the Differentiator responds to noise at the baseline. To prevent this, the signal is also passed through a Gate Channel which prevents any output pulse before the input signal has crossed an established level. This Gate Channel comprises a Differential Comparator with Hysteresis and a D flip-flop. The hysteresis for this comparator is set externally via the Set Hysteresis pin.

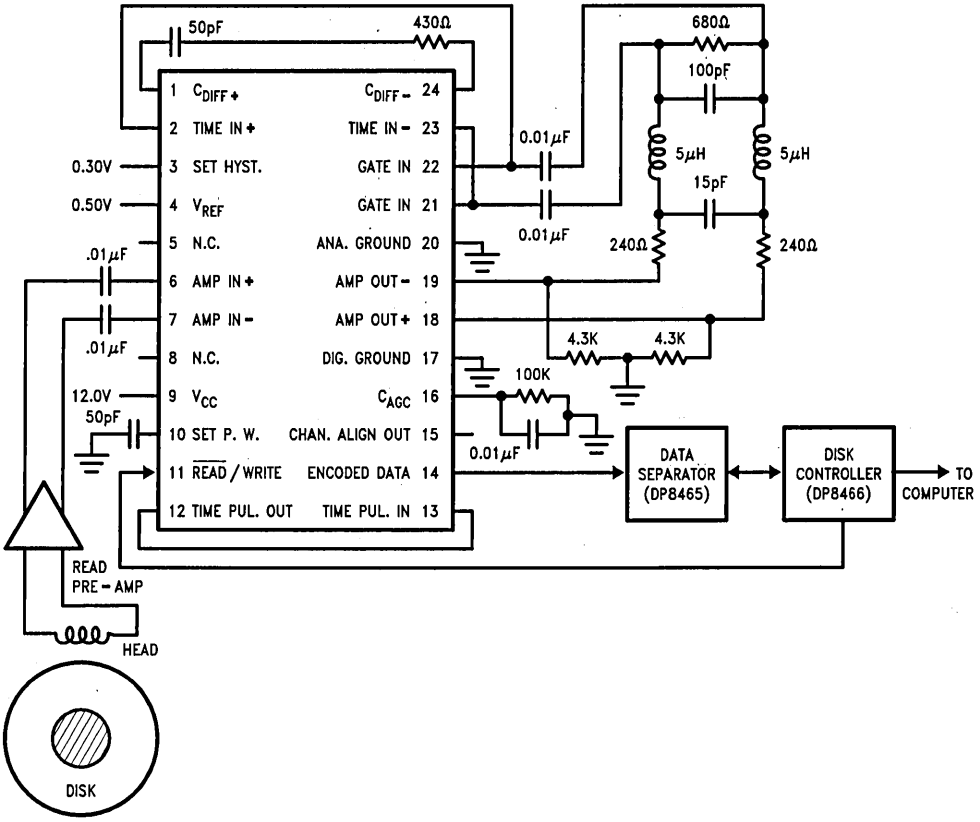
The operation of this Gate Channel is shown in *Figure 2.5*. At the top is a typical waveform which exhibits shouldering at the lowest frequency, and is almost sinusoidal at the highest frequency. This waveform is fed to both the Time and the Gate Channel. The hysteresis level (of abut 30%) has been drawn on this waveform. The second waveform is Time Pulse Out. While there is a positive edge pulse at each peak, there is also noise at the shoulders. Since Time Pulse Out is externally connected to the Time Pulse In, this output is therefore the clock for the D flip-flop.

The third waveform is the output of the Comparator with Hysteresis which goes to the D input of the flip-flop. The true peaks are the first positive edges of the Time Pulse Out which occur after the output of the comparator has changed states. The D flip-flop will "clock" in these valid peaks to the output bi-directional one shot. Therefore, the noise pulses due to the Differentiator responding to noise at the baseline just "clock" in the old data through the flip-flop so there is no noise pulse on Encoded Data.

The Q output of the flip-flop drives the bi-directional one shot which generates the pulses on Encoded Data. The positive edges on Encoded Data correspond to the signal peaks. The width of the data pulses can be controlled by an external capacitor from the Set Pulse Width pin to ground. This pulse width can be adjusted from 20 ns to 1/2 the period of the highest frequency.

Design Example

A typical system implementation of the DP8464B is shown in *Figure 2.6*. The DP8464B is driven from the Read/Write amplifier which is generally located very close to the actual Read/Write head. This amplifier is fixed gain and pre-amplifies the weak readback signal picked up by the head from the surface of the disk.



TL/F/8663-42

Shown here is a typical 10 Mbit/sec MFM disk drive application. Signals are picked off the disk by the Read/Write head and amplified by the pre-amplifier. The DP8464B further amplifies the read signal and accurately represents the time position of the peaks by the rising edge of the signal at the Encoded Data output pin. The Data Separator locks up to the signal at the Encoded Data output and provides decoded MFM data to the Disk Data Controller. The Disk Data Controller handles the interface between the disk drive and the computer.

FIGURE 2.6

The output of the DP8464B drives a data separator (such as the DP8465) which extracts the digital data from the encoded data supplied by the DP8464B. The data separator utilizes a phase-locked loop that locks onto the leading edge of the encoded data signal from the DP8464B.

In addition to extracting the digital data from the encoded data, the data separator synchronizes the digital data thereby removing any timing jitter present in the encoded data signal. The data separator implements this last function by opening up timing windows that bracket the encoded data signal. The encoded data signal need only appear within the window to be detected. For 10 Mbit MFM, the window is only 50 ns wide.

In order to guarantee that the drives have error rates on the order of 1 per 10 to the 10th power, bits read (industry standard), the leading edge of the encoded data must fall well within the 50 ns window. Because of this stringent criteria, there is little room for error with regard to the accuracy that the DP8464B can extract the relative time position of the peaks of the read back signal.

One form of timing error is jitter of time position of the leading edge of the encoded data signal. This jitter can be a result of noise output from the differentiator or noise pickup in other portions of the read channel. These timing errors will significantly affect the Encoded Data signal causing an increase in the error rate.

The filter that drives the differentiator is important in reducing the noise input to the differentiator. For this reason, a high order Bessel filter with its constant group delay characteristic can be used in this application. The constant group delay characteristic insures that the filter does not introduce any timing errors by distorting the signal and moving the position of the peaks. Often, this filter must be specifically designed to correct phase errors introduced by the non-ideal characteristics of the input read head. The typical −3 dB point for this filter is around 1.5 times the highest recorded frequency. Reducing the noise input to the differentiator will ultimately reduce the amount of noise jitter on the encoded data output.

Another way to reduce noise jitter is to limit the bandwidth of the differentiator with a series combination of resistor, capacitor and inductor in the external differentiator network and to use as large a differentiator capacitor as possible, thereby maximizing the differentiator gain. In order to prevent saturation of the differentiator, Schottky diode clamps were added to the differentiator output thus allowing the use of a larger differentiator capacitor.

An automatic gain control (AGC) circuit is used to maintain a constant input level to the gating channel (which is typically tied directly to the input of the differentiator). By maintaining a constant signal level at this point, we insure not only a large input level to the differentiator but also a constant level of hysteresis of the signal to the gating channel. Gain control is also necessary because the amplitude of the input signal will vary with track location, variations in the magnetic film, and differences in the actual recording amplitude. The peak-to-peak differential amplitude on the Gate Channel Input is four times the voltage set by the user on the V_{ref} pin.

The actual dynamics of the AGC loop are very important to the system operation. The AGC must be fast enough to respond to the expected variations in the input amplitude, but not so fast as to distort the actual data. A simplified circuit of the AGC block is shown in *Figure 2.7*. When the full wave rectified signal from the Amplifier is greater than V_{ref}, the voltage on the collector of transistor T1 will increase and charge up the external capacitor C_{agc} through T2. The maximum available charging current is 3 mA. Conversely, if this input is less than V_{ref}, transistor T2 will be off, so the capacitor, C_{agc}, will be discharged by the base current going into the Darlington T3 and T4. This discharge current is approximately 1 μA. The voltage on the emitter of T4 controls the gain of the Amplifier.

If the AGC circuit has not received an input signal for a long time, the base current of the Darlington will discharge the external C_{agc}. The Amplifier will now be at its highest gain. If a large signal comes in, the external C_{agc} will be charged by the 3 mA from T2, thereby reducing the gain of the Amplifier. The formula, $I = C^*(dV/dt)$ can be used to calculate the time required for the Amplifier to go from a gain of 200 to a gain of 6. For instance, if $C_{agc} = 0.05\ \mu F$, the charging current I is 3 mA, and the dV required for the Amplifier to go through its gain range is 1V, then

$$dt = (0.05\ \mu F^*\ 1V)/(3\ mA) \text{ or } 17\ \mu s.$$

By using the same argument, the time required to increase the Amplifier gain after the input has been suddenly reduced can be calculated. This time, the discharging current is only 1 μA, so

$$dt = (0.05\ \mu F^*\ 1V)/(1\ \mu A) \text{ or } 50\ ms.$$

This time can be decreased by placing an external resistor across C_{agc}.

2.4 READ/WRITE

In the normal read mode, the signal from the read/write head amplifier is in the range of 20 mV_{pp} to 660 mV_{pp}. However, when data is being written to the disk, the signal coming into the analog input of the pulse detector will be on the order of 600 mV. Such a large signal will disturb the AGC level and would probably saturate the amplifier. In addition, if a different read/write amplifier is selected, there will be a transient introduced because the offset of the preamplifiers are not matched.

A TTL-compatible $\overline{\text{READ}}$/WRITE input pin has been provided to minimize these effects to the pulse detector. When the $\overline{\text{READ}}$/WRITE pin is taken high, three things happen. First, the 1k resistors across the AMP IN pins are shunted by 300Ω resistors. Next, the amplifier is squelched so there is no signal on the Amp Output. Finally, the previous AGC level is held. This AGC hold function is accomplished by not allowing any current to charge up the external C_{agc}. The voltage across this capacitor will slowly decrease due to the bias current into the Darlington (see *Figure 2.7*) or through any resistor placed in parallel with C_{agc}. Therefore, the gain of the amplifier will slowly increase. All of these three events happen simultaneously.

When the $\overline{\text{READ}}$/WRITE input is returned low, the pulse detector will go back to the read mode in a specific sequence. First, the input impedance at the Amp In is returned to 1k. Then, after approximately 1.2 μs, the Amplifier is taken out of the squelch mode, and finally approximately 1.2 μs after that, the AGC circuit is turned back on. This return to the read mode is designed to minimize analog transients in order to provide stable operation after 2.4 μs.

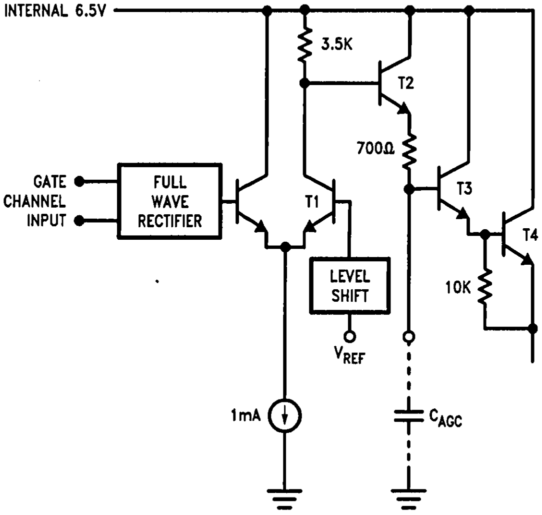
2.5 CONCLUSION

The push to higher disk capacities in increasingly smaller drives has forced drive manufactures to utilize different media, heads and encoding. Each of these changes render the traditional de-snaker pulse detector unusable. The DP8464B pulse detector utilizes a new detection technique that overcomes the limitations of the de-snaker. Furthermore, the DP8464B provides both gain controlled amplification of the pre-amplified readback signal and the ability to disable the circuit during write operations. The DP8464B is easily adapted to a wide variety of applications through selection of external components.

References

1. I. H. Graham, *"Data Detection Methods vs. Head Resolution in Digital Recording,"* IEEE Transactions on Magnetics Vol. MAG-14, No. 4 (July 1978).

2. I. H. Graham, *"Digital Magnetic Recording Circuits,"* available through the University of Santa Clara, (California), bookstore (408) 554-4491.



TL/F/8663-44

The AGC Circuit senses the level of the signal at the Gate Channel Input and compares it to an externally set reference voltage. The signal that results from this comparison (at the collector of T1) charges C_{AGC}. The voltage across C_{AGC}, when buffered by T3 and T4, provides the gain control voltage to the input amplifier.

FIGURE 2.7

CHAPTER 3 DISK DATA SEPARATOR OVERVIEW (DP8460/61/62/65 AND DP8451/55)

3.0 INTRODUCTION—THE DATA SEPARATOR

As was discussed in the chapter on Disk Drive Technology (overview), the disk information which is recovered during a read operation ordinarily would have no defined phase relationship with respect to the timing within the host system. In order to establish a method of reconstructing a clock waveform with which the disk data may be entered into a shift register for deserialization and decoding, clock information is imbedded into the recorded bit pattern in any of a number of different ways by the various encoding schemes discussed in Chapter 1. The schemes vary in their efficiency of use of disk surface (bit density) and ease of recovery (challenge to the data separator), but they all are employed to achieve a mixture of clock and data within the same serial bit stream. It is then the function of the data separator to accurately extract this clock information from the bit stream and reconstruct a stabilized replica of the data, while at the same time remaining essentially immune to the random displacement of individual bits due to noise, media defects, pulse crowding and anomalies in the data channel.

From a "black box" standpoint, the data separator is fed a logic-level digital signal from a pulse detector (DP8464) within the disk head electronics (with positive transitions representing flux reversals on the media) and a read gate signal from the controller, and produces a reconstructed clock waveform along with a re-synchronized data output derived from the incoming disk pulse stream (see *Figure 3.1*). The regenerated clock and data signals have fixed timing relationships with respect to one another for use by subsequent shift register circuitry.

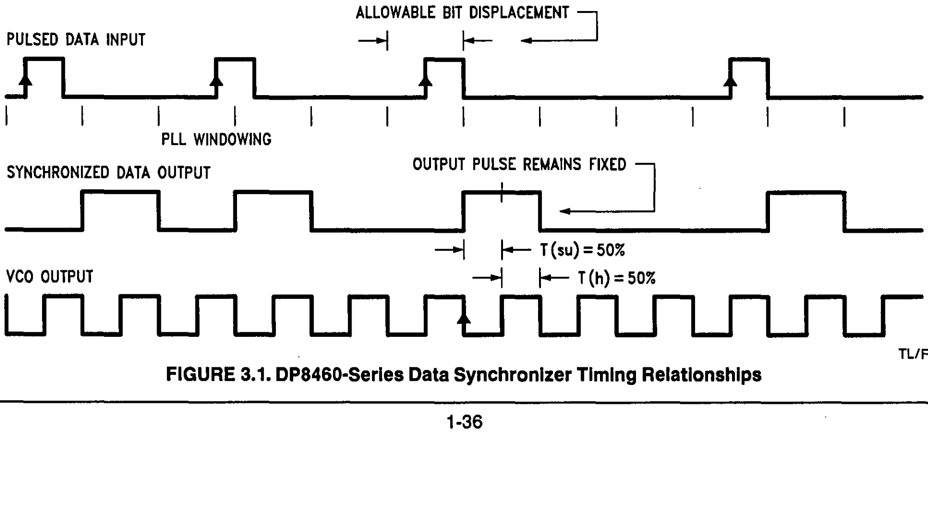
3.0.1 Separators and Synchronizers

The term "data separator" actually applies to a device which both regenerates a clock waveform from the bit stream as well as decodes (separates) the original NRZ data from the encoded disk data. This would include National's DP8461 and DP8465, both of which perform data synchronization with MFM-to-NRZ data separation, while including slight functional variations between devices. (The DP8460 initially released device is being replaced by the fully pin-for-pin compatible DP8465. The DP8461 pinout matches the DP8465, but is intended for use with hard and pseudo-hard sectoring only. Further details will be discussed later.) A device which performs clock regeneration and data synchronization without the separation function is simply called a data synchronizer. This would include the DP8451, DP8455, and the DP8462; again, there are functional differences between the devices, which will be discussed later. Additionally, the DP8461 and DP8465 also have outputs available which allow them to serve as data synchronizers, if desired (See Table 3.1).

The complete data separator circuit eliminates the need for external decoding circuitry but is dedicated to only a single code type. The data synchronizer requires an external decoding network but has the capacity to be used with any coding scheme. Since the MFM environment is being addressed in this design guide, the discussion in the remainder of this chapter will deal primarily with the integrated, MFM-type data separator. The PLL fundamentals being presented apply to all of the circuits.

3.0.2 Window

The data separator must establish what is called a "window" around the expected position of bits within the disk data stream. Windows are laid end-to-end in time by the data separator at a repetition rate (equal to the separator's VCO frequency) known as the disk code rate. Each window is an allotment of time within which a disk bit, if detected, will be captured and interpreted as if it had occurred exactly at the window center. This allows for a random displacement (jitter) of individual bits within the boundaries of the window with no apparent effect on the accuracy of the data recovery (error rate). Bits are displaced from a nominal position in a fashion which could be represented by a bell-shaped probability curve (see *Figure 3.2*), and it could be easily seen that for optimum performance, the window must be accurately centered about the mean of this curve. This is traditionally a difficult goal to achieve, and is essentially the primary responsibility of the data separator. Although various techniques have been employed to attain this goal, the phase-locked loop (incorporated within all National data separator/synchronizers) has proven to date to yield the most reliable and satisfactory results among all the synchronization methods, and its use is the standard approach taken within the disk industry.

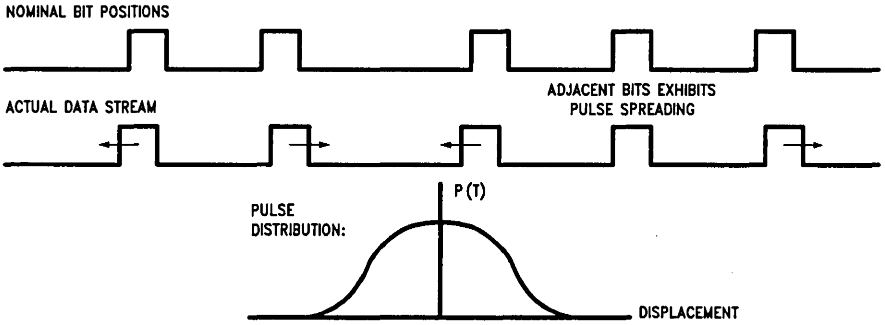


TL/F/8663-45

FIGURE 3.1. DP8460-Series Data Synchronizer Timing Relationships

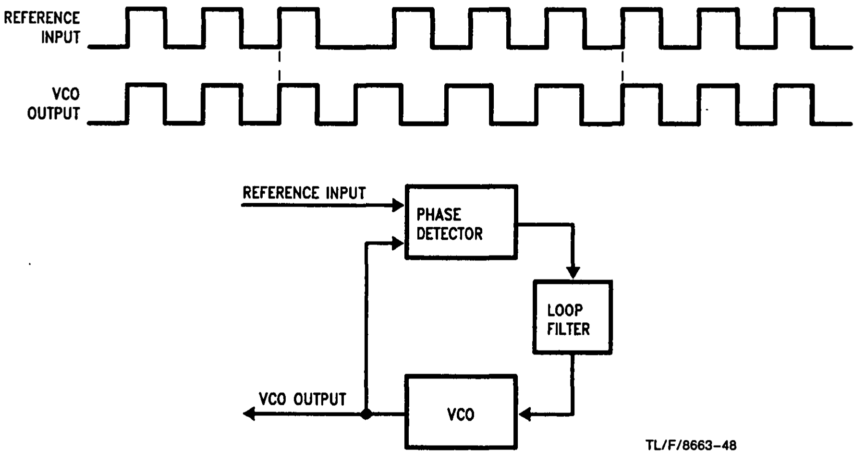
3.1 PHASE-LOCKED LOOP OVERVIEW

A phase-locked loop is a closed-loop control system which forces the phase from the output of a controlled oscillator to track the phase of an external reference signal. It consists of three essential elemental blocks; a phase detector, a loop filter, and a voltage controlled oscillator (VCO) (see *Figure 3.3*). The phase detector compares the phase of the reference input with that of the VCO output and generates an "error" signal at its output which is proportional to the sensed phase difference. This error signal is filtered by the loop filter (low-pass) to suppress any unwanted high-frequency components within the error signal, and is then fed to the VCO control voltage input. If the phase of the reference signal leads that of the VCO signal, the phase detector develops a positive error voltage across the loop filter, and the VCO responds by increasing its frequency (advancing phase). This continues until the phase error is eliminated and the control voltage returns to a quiescent value. When the reference signal lags that of the VCO, the adjustment occurs in the opposite direction until equilibrium (phase lock) is again obtained.



TL/F/8663-46

FIGURE 3.2. Data Jitter



TL/F/8663-47

TL/F/8663-48

A phase-locked loop is a system which forces the phase of the output from a voltage controlled oscillator (VCO) to track the phase of a reference signal.

FIGURE 3.3

3.1.1 PLL Dynamics

Much of the performance of the PLL, given adequate design in the major functional blocks, is determined by the loop filter. This includes (1) the ability of the PLL to rapidly (or slowly) track phase or frequency changes in the reference signal, (2) the ability of the loop to re-aquire lock after encountering a large frequency step at the reference input, and (3) the ability of the loop to exhibit stable behavior during operation.

TRACKING

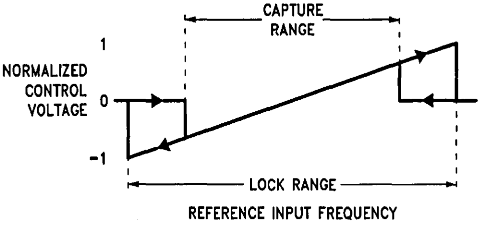
In many PLL applications, such as FM demodulation and frequency shift keying demodulation, rapid PLL tracking (high bandwidth) is a necessity. However, in the disk drive application, where (1) frequency changes (disk rotational speed variations) are gradual with respect to the data rate and (2) it is desirable to suppress response to instantaneous bit shift (jitter), a very slow tracking rate (low bandwidth) is necessary.

CAPTURE RANGE

The classical PLL is well able to maintain relative phase lock to a reference signal, but is unable to pass true difference-frequency information through its phase detector (this is true for the standard analog four-quadrant multiplier technique as well as for the gated-VCO technique employed in disk drive data separation applications). With this being the case, the PLL has a limited ability to re-establish lock when an instantaneous input frequency change occurs. The new frequency must lie inside a relatively narrow band on either side of the current VCO frequency, or re-lock will not occur (see *Figure 3.4*). This band is known as the capture range, and is a direct function of the passband of the loop filter, or more accurately, of the bandwidth of the PLL as a whole. A digital frequency discrimination technique, however, is employed in National Semiconductor's disk drive PLL's which provides an extended capture range and guarantees successful lock to the reference clock input and, as a chip dependent option, to the data as well. Consequences of having a limited capture range are discussed in National Semiconductor Application Notes AN414, AN415, AN416 (Also see "Frequency Lock" in section 3.2 of this chapter).

STABILITY

Mathematical analysis of the functional blocks of the PLL show a 1/s factor in the VCO; i.e., it behaves as a integrator, adding a pole to the transfer function. The loop filter adds at least one additional pole, resulting in a system which is, at minimum, second order in nature. Since the PLL is then a closed loop, minimum second order system, it has the potential for instability if improperly implemented. All of the dynamic characteristics of the PLL, however, can be controlled by loop filter selection, including bandwidth and capture range along with stability; thus great care must be taken in the selection of the loop filter in order to achieve the desired performance within the specific application.



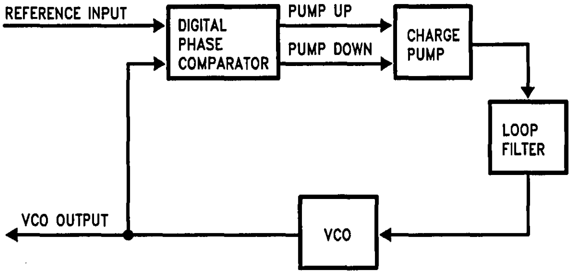
TL/F/8663-49

Capture range in a PLL is determined by the loop filter bandwidth.

FIGURE 3.4. Capture and Lock Range

3.2 THE PLL WITHIN A DISK DRIVE SYSTEM

Many methods have been employed in implementing each of the individual blocks within the phase locked loop (PLL), with techniques being customer-tailored to specific applications. Design methods include analog and digital configurations or a combination of both. For the application of a PLL within a disk drive data separator (and specifically regarding National's family of data separator/synchronizers), a combination of digital and analog block design has been found to provide the most efficient and reliable solution (see *Figure 3.5*). Here, since the waveforms to be compared are digital signals and the phase relationships are indicated by logical transitions (positive edges), the phase detector is comprised of a simple set of cross-coupled latches which produce "pump-up" (reference leads VCO) and "pump-down" (reference lags VCO) digital outputs. Since the filtering, however, is most easily and flexibly performed with passive analog components, the pump-up and pump-down signals are converted into gated sourcing and sinking currents, respectively, via an analog "charge pump" circuit, which is used to develop an error voltage across a capacitive loop filter (see *Figure 3.6*). This error voltage is used as a control potential for a variable rate relaxation oscillator (emitter coupled multivibrator VCO), whose oscillation is converted to digital signal levels again for use both at the phase detector input and as the regenerated clock waveform.

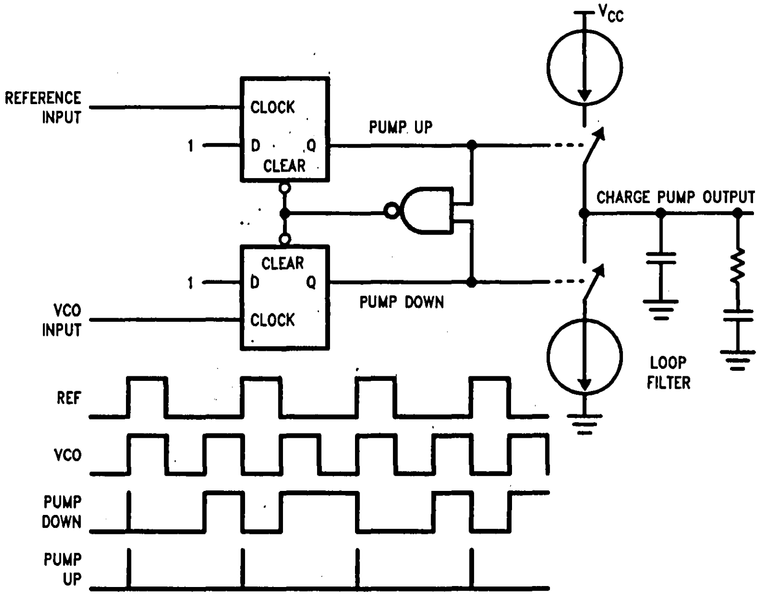


TL/F/8663-50

Goals:

1. Edge sensitive detector which eliminates dependence on waveform shape.
2. Unlimited capture range to ensure phase and frequency lock.
3. Zero phase difference when in lock to improve lock range.

FIGURE 3.5. Basic Disk System PLL



TL/F/8663-51

FIGURE 3.6. Digital Phase Detector

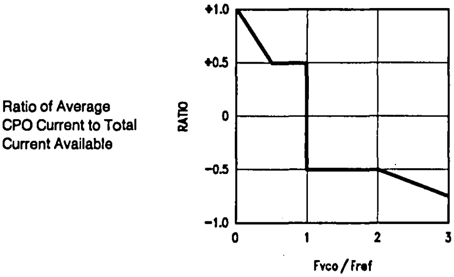
PHASE DETECTOR

The digital phase detector employed within National's data separator/synchronizer circuits is actually a true phase/frequency discriminator block capable of allowing a theoretically infinite capture range for the PLL (see *Figures 3.7–3.10*). Essentially, the capture range is limited only by the design constraints placed on the VCO's frequency excursion. In all of National's current disk PLL circuits (excluding the DP8460/50), this extended lock capability is employed while the circuit is in the non-read mode and the PLL is locked to a constant reference signal, guaranteeing proper lock recovery from any given mislock which may occur during a read operation.

PULSE GATE

The data returning from the disk is not a periodic waveform, but instead has the possibility of bits either appearing or not appearing within assigned positions (windows) in the data stream (see *Figure 3.1*). The PLL is required to achieve and maintain lock to this pseudo-random pattern, despite the missing bits. This is analogous to the placement of teeth in a gear (data separator) which are ready to mesh with another gear (data stream), regardless of whether or not some teeth on the second gear (data) are occasionally missing (random data patterns).

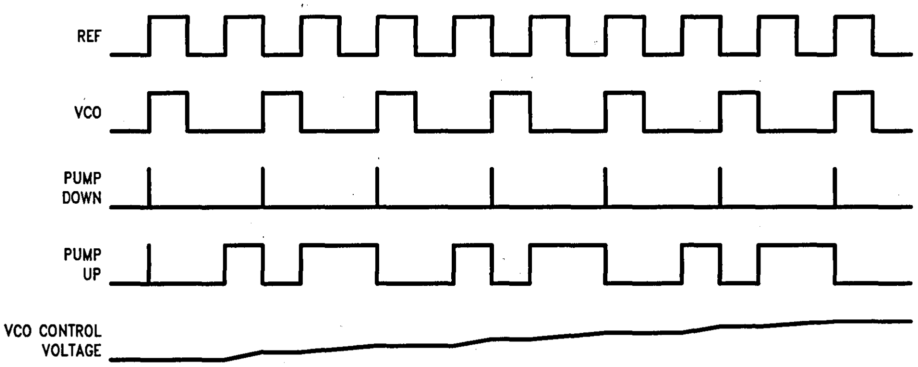
In order to allow for the missing bits, the PLL employs a Pulse Gate circuit, which functions as follows: a data bit arriving at the PLL is sent to its corresponding input on the phase detector, and at the same time trips a gate which allows the next occurring VCO edge into the phase detector; the gate then closes following transmission of the VCO edge. If no data bit arrives, no comparison occurs and the VCO holds its frequency. Essentially, the PLL is attempting to align each data bit with the nearest occurring VCO edge, thus maintaining phase lock while frequency discrimination is suppressed (see *Figures 3.11* and *3.12*).



TL/F/8663-52

Ensures unlimited capture range. The digital phase detector ensures frequency acquisition by forcing the charge pump to always pump in the direction needed to make F_{VCO} equal to F_{REF}.

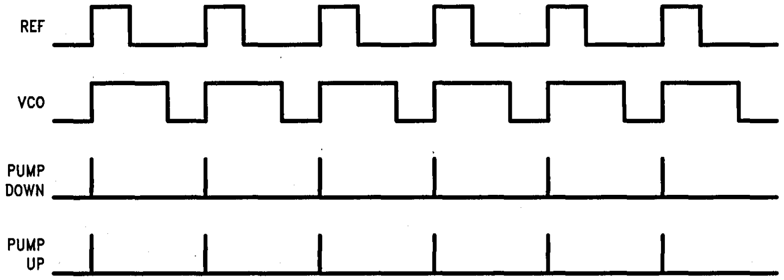
FIGURE 3.7. Digital Phase Detector



TL/F/8663-53

When the digital phase-lock loop is out of lock, the output of the phase comparator has a duty cycle which varies between 0 and 100%. The charge pump is active more than 50% of the average time but it only pumps current in the direction necessary to lock the VCO phase.

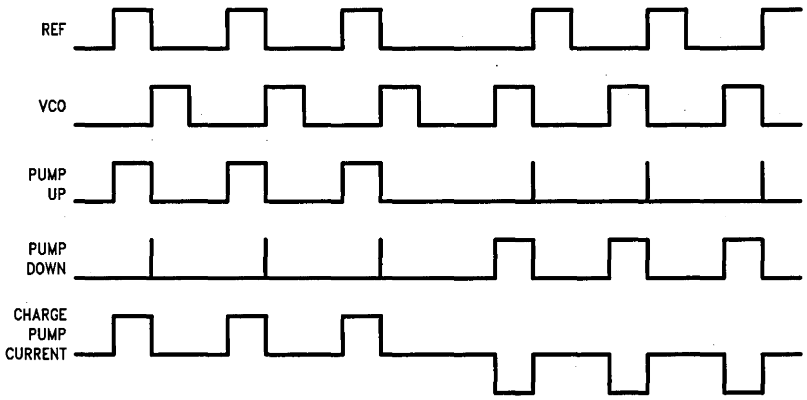
FIGURE 3.8. Digital Phase Detector



TL/F/8663-54

Once the VCO is locked to the reference signal, the only phase difference which occurs will be that required to pump enough charge to compensate for any leakage current in the charge pump, loop filter or bias current in the VCO control input.

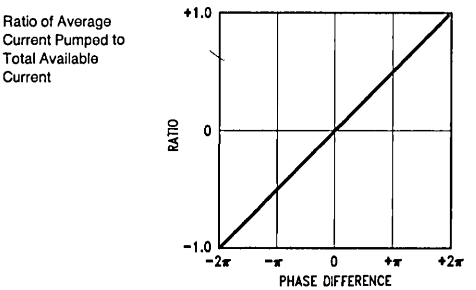
FIGURE 3.9. Digital Phase Detector



TL/F/8663-55

Effect of phase difference on digital phase detector when the VCO and reference frequencies are equal. The net charge pumped during each period is equal to the product of the charge pump current and the time difference between the phase comparator inputs.

FIGURE 3.10. Digital Phase Detector



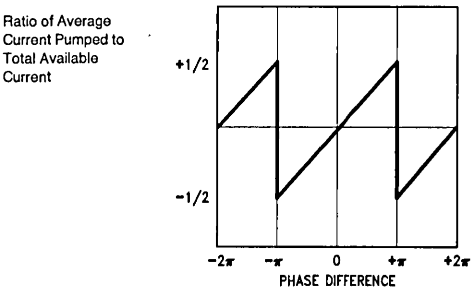
TL/F/8663-56

Phase detector gain when in lock

When the loop is in lock the net charge (Q) pumped during each period is equal to the product of the charge pump current and the time difference between the phase comparator inputs.

$$Q = I_{cpo} (T_{up} - T_{down})$$

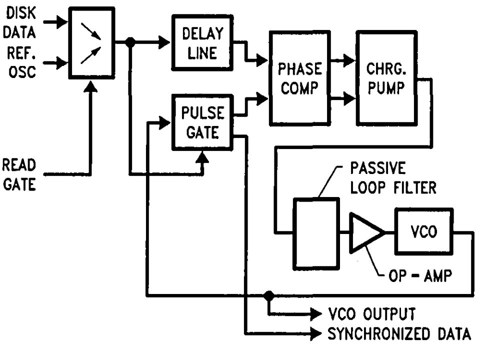
FIGURE 3.11. Digital Phase Detector Gain without Pulse Gate



TL/F/8663-57

Each vertical transition represents a window boundary.

FIGURE 3.12. Digital Phase Detector Gain with Pulse Gate



TL/F/8663-58

FIGURE 3.13. Disk Data PLL

DELAY LINE

In the generation of the symmetrical bit-detection window mentioned previously, a delay line is employed as shown in *Figure 3.13.* The incoming data is allowed to trip the VCO pulse gate immediately upon arrival, but is allowed into the phase detector only after it traverses the delay line. To understand the purpose for the delay line, first consider the case where the delay line is not present; given proper PLL lock, the VCO would align its edge to occur exactly at the same time as the arrival of each data bit. Any bits shifted early would cause a small VCO phase-advancing correction within the loop, which would be desirable. However, any bits shifted even a very slight amount late would arrive after the current VCO edge had passed and been suppressed by the pulse gate. The bit would then have to be compared to the subsequent VCO edge instead of to the current edge, producing an erroneous phase correction. With the delay line in place and set to delay the data bit by one-half of the VCO period, the data bit would first trip the VCO gate and then spend one-half of the VCO period traversing the delay line before it reached the phase detector. Given the loop is in lock, both the delayed bit and the VCO edge arrive at the phase detector at exactly the same time. If the bit were early up to one-half VCO cycle, it would still gate the appropriate VCO edge through, and produce an appropriate phase correction at the phase detector. Also, if the bit were late up to one-half of the VCO cycle, it would again still gate the appropriate VCO edge through to the phase detector, as well as produce the appropriate phase correction. Thus, the net effect of the delay line is to allow the incoming data bit to shift either one-half cycle early or one-half cycle late while yet maintaining a proper comparison to the appropriate VCO edge.

WINDOW ACCURACY

As mentioned previously, the integrity of the window alignment is crucial in maintaining an acceptable system error rate. It can be easily seen that accuracy in the delay line is critical in achieving this alignment. This has traditionally made the implementation of a delay line a costly design challenge. Within National's data separator/synchronizer circuits, a proprietary technique has been employed which extracts precise timing information from the 2F CLOCK input waveform, and uses this information to regulate the timing within an on-chip silicon delay line. This in itself has unique advantages since the 2F source is either (1) a highly accurate crystal oscillator source, or (2) derived from the disk servo clock which tracks the data rate and consequently allows the delay line to adjust its delay accordingly. Completely dependent on this 2F signal for timing information, the delay remains independent of variations in its associated external components, power supply, temperature, and silicon processing. It requires no adjustment (although fine tuning is optional on the DP8462), and its accuracy is guaranteed within the window tolerance specification for the device.

FREQUENCY LOCK

The frequency discriminating capability of the phase detector within National's data PLL circuits can be employed to great advantage if used appropriately. It is brought into play simply by the internal bypassing of the pulse gate circuit. The advantages achieved are (1) the avoidance of mis-lock to the reference clock input, (2) rapid and guaranteed lock recovery from an aborted read operation, and (3) avoidance of mis-lock within the disk PLL synchronization field (preamble).

Items #1 and #2 above are easily attained by pulse gate bypassing when the PLL is locked to the reference signal in the non-read mode. Pulse gate bypassing allows the digital phase detector to perform unrestricted frequency comparison and thus guarantees lock. Both items are employed within all of National's currently released data separator/synchronizer circuits (see Table 3.1).

Incorporation of item #3 (employed within the DP8461 and DP8451, and optional within the DP8462) is highly dependent on preamble type and places specific requirements on the controller's sector search algorithm. First, there are several common preamble types currently in use on disk drives; (1) the MFM and 1,N type, (2) the 2,7 high frequency preamble, the (3) the 2,7 low frequency preamble:

Code Type	VCO Cycles (Code Positions or Windows) Per Recorded Preamble Bit
GCR*	1
MFM	2
1,7	2
1,8	2
2,7	3
2,7	4

*Note: GCR (Group Code Recording) is used almost exclusively in tape drive systems; it is mentioned here for comparative purposes only.

Since each preamble is recorded at a different frequency with respect to the VCO operating frequency, the VCO must be internally divided down to equal the preamble frequency for the particular code in use before being fed into the phase detector along with the data pattern. (This function is performed internally within specific National disk PLLs listed in Table 3.1.) It is then the responsibility of the PLL to detect the occurrence of frequency lock and revert back to the pulse gate mode prior to leaving the preamble and encountering random data patterns. Second, while in the frequency acquisition mode, the controller must allow a read operation to begin (initial PLL lock to the data) only during the presence of the appropriate field, i.e., the system must employ a hard-sectored or pseudo hard-sectored PLL control algorithm which will guarantee the PLL read gate will only be asserted at the start of the preamble on the disk, otherwise serious PLL mislock problems will result.

DP846X EXPOSITION

Because of the varied requirements and applications which exist for the data separator/synchronizer, National provides an assortment of disk PLL circuits, including versions which provide frequency lock for specific preamble types, as mentioned above.

TABLE 3.1. Data Separator/Synchronizer Reference List

Device	Synchronized Codes	Separated Codes	Frequency Lock	Delay Trim
DP8461*	MFM; 1,N	MFM	Reference & Data	None
DP8462*	2,7 1,N MFM	None	Reference (Optional for Data)	Optional
DP8465*	All	MFM	Reference	None
DP8451	MFM; 1,N	None	Reference & Data	None
DP8455	All	None	Reference	None

Note 1: "All" code synchronization does not include GCR.

Note 2: DP846X devices are in the 24-pin, 300 mil. package; DP845X devices are in the 20-pin, 300 mil. package.

Note 3: *Also available in 28-lead plastic chip carrier.

Note 4: DP8461 and DP8451 pinouts match the DP8465 and DP8455, respectively; for use with hard and pseudo-hard sectoring only.

Note 5: DP8451 and DP8455 are also available in 20-pin plastic chip carrier.

3.3 SYSTEM DYNAMICS—LOOP FILTER DESIGN

The key element contained within the PLL system for governing the loop dynamics and overall performance is the loop filter. It is at this point that the user has the greatest flexibility and control regarding the behavior of the PLL. As previously mentioned, there are several requirements placed on the dynamics of the loop, some of which tend to conflict with others. Table 3.2 lists some of the issues at hand, and where they lie with respect to one another.

TABLE 3.2

	High Bandwidth	Low Bandwidth	High Damping Factor	Low Damping Factor
Lock Time	Good	Poor	Good	Poor
Jitter Rejection	Poor	Good	Poor	Good
Capture Range	Good	Poor	Good	Poor
Noise Immunity	Poor	Good	Poor	Good
Stability	(No Relationship)		Good	Poor

Although the designations of "good" and "poor" are very general in nature, they apply fairly well here for comparative purposes. An ideal PLL would be able to lock to any frequency and/or phase step in a very short time with no possibility of missed or false lock, settle quickly to a highly stable state, and track any frequency variations encountered in the data stream while at the same time rejecting all bit jitter and extraneous noise. While all this is not possible with any single loop filter, acceptable performance can be achieved via careful compromise, with the design biased to accommodate the more critical parameters.

3.4 OVERVIEW OF FILTER DESIGN OBJECTIVES

The first design objective to be discussed is the minimization of acquisition time. This includes both acquisition of phase lock to the data stream as well as acquisition of phase lock to the crystal frequency (or servo track). Both of these acquisition times impact the length of the preamble field which precedes the address mark. Since longer preambles result in more overhead per data sector, a decrease in formatted disk capacity may result from excessively long acquisition times. Acquisition times are directly controlled by the phase locked loop filter.

The second design objective is the maximization of data margin. Data margin measures the ability of the data separator to allow the data bit to move from its expected time position without a resulting data error. This movement of the data bit away from its expected time position is caused by noise, read channel asymmetry, magnetic domain interference, and other factors in the head, media, and channel portion of the drive system. The data separator generates a window around the expected data position; however, the window accuracy is affected by some factors in the data separator. These factors include delay inaccuracy, VCO jitter, phase detector inaccuracy, and phase locked loop response to bit movement which occurred in preceding windows. The last of these factors, loop response, is controlled by the phase locked loop filter.

The final objective to be checked is the tracking of disk data. The rate of change of phase between the VCO and the read data is modulated by various mechanical phenomena in the drive. Instantaneous variations in disk speed as well as head vibration contribute to this modulation. The maximum frequency of these mechanical resonances tends to be in the 10 kHz or 64 Krads/sec range. Phase-locked loop bandwidth must be wide enough to allow this modulation to be tracked. This objective tends to be encompassed by the acquisition time objective. However, it is conceivable that a system which allows relatively long acquisition times may come up against this barrier.

The loop filter design process may start with any one of these objectives. If the disk format has been established, or a certain disk capacity is desired, the acquisition performance may dictate the loop filter design. If data reliability and error rates are of primary importance, the design may start with margin loss considerations. In any case, all aspects of the loop performance must be checked and the final design is usually a tradeoff between the desired performance and the achievable performance.

3.5 ACQUISITION PERFORMANCE

The read acquisition time is the time between the assertion of READ GATE and the reading of the address mark. Also of concern Is the time required for the loop to acquire lock to the crystal frequency. Many application-specific system parameters impact this portion of the loop design. Some of these parameters which will be discussed include sector search algorithms in soft sectored systems and frequency differences between the crystal and the data in removable media systems.

Before the READ GATE is asserted, the VCO is locked to the crystal. When READ GATE is asserted, the phase difference between the VCO and the read data is random. The first portion of the acquisition is where the loop captures phase alignment. In the worst case, the initial phase alignment is such that the data bit is positioned at the edge of the window which gives the proper polarity of error signal, however, the loop cannot keep the bit in the window since it started so close to the edge. One of the results is that the incoming data will appear to be different than what is actually being read. Any system which desires to immediately monitor the read data must wait for this initial cycle slip to occur before reading. The second result is that a series of error signals of the wrong polarity occurs after this initial cycle slip while the phase aligns to the window center. The duration of this slip and phase acquisition is approximately $1/\omega n$ for damping factors between 0.7 and 1.0. See Section 3.7 for acquisition plots. Note that ωn is the loop bandwidth or the natural frequency of the loop and that the phase error is zero at $\omega nt = 1$.

The next period of the acquisition is where the loop begins to capture frequency. There is also some overshoot from the phase acquisition during this period. This analysis assumes that the frequency acquisition begins where $\omega nt = 1$ and is superimposed upon the phase acquisition for $t > 1/\omega n$.

In a fixed media system the difference between the crystal frequency and the read data frequency can be about 1%. In a removable media system the data may be written in one drive and read in another. The total difference between the read data frequency and the crystal frequency can be twice the rotational speed difference of the drive. For example, if the rotational speed variation of the drive is +1% when the disk is written, and then −1% when the disk is read, the read back data frequency will be 2% slower than the crystal frequency. Since a frequency difference becomes a phase ramp, the phase error will initially grow during read acquisition while the loop attempts to hold the phase error to zero (see Section 3.7). If the peak phase error which results exceeds the window tolerance, the loop will not capture within the desired acquisition time. The data will slip out of its proper window into the neighboring window and a series of error signals of the wrong polarity will result.

When the data rate is 5 Mbit/sec, the window period is 100 ns and the peak phase error allowed is ideally 50 ns. For a damping factor of 0.7, the peak phase error is given by:

$$\text{peak error} = (0.45)\ (\Delta\omega/\omega n)\ (200\ \text{ns}/2\pi) + 0.21(\Delta\theta)$$

where $\Delta\theta$ is the worst case initial phase (50 ns). The 200 ns/2π term converts radians to seconds. $\Delta\omega$ is the worst case initial frequency difference which is determined by the rotational speed variation of the drive. For a removable media drive with 1% variation:

$$\Delta\omega = (2\pi/200\text{ ns})\,(0.02) = 628\text{ Krads/sec}$$

The results are as follows

ωn = 600 Krads/sec → peak phase error = 25.5 ns
ωn = 500 Krads/sec → peak phase error = 28.5 ns
ωn = 400 Krads/sec → peak phase error = 33.0 ns
ωn = 300 Krads/sec → peak phase error = 40.5 ns

For ωn = 300 Krads/sec, with a damping factor of 0.7, the loop will just barely capture. If a removable media system were to use a 300 Krads/sec loop bandwidth, the capture range analysis should be performed very carefully since the numbers given here are very approximate. Capture performance improves for larger damping factors, however, total acquisition time increases.

There is no precise definition of phase lock. At the end of the preamble, there will be some residual phase error. The loop is locked if this phase error contributes an acceptable amount of margin loss during the reading of the address mark. It is recommended that the residual error be about 2 ns in a 5 Mbit/sec data rate system:

$$2.0\text{ ns }(2\pi/200\text{ ns}) = 0.062\text{ radians}$$

If the damping factor is 0.7, the following results

ωn = 600 Krads/sec → error = 0.062 at t = 7 μs
ωn = 500 Krads/sec → error = 0.062 at t = 9 μs
ωn = 400 Krads/sec → error = 0.062 at t = 12 μs
ωn = 300 Krads/sec → error = 0.062 at t = 17 μs

The total read acquisition time includes the initial phase acquisition as follows:

ωn = 600 Krads/sec → 7 μs + 1.7 μs
= 9 μs = 5.5 bytes
ωn = 500 Krads/sec → 9 μs + 2.0 μs
= 11 μs = 6.5 bytes
ωn = 400 Krads/sec → 12 μs + 2.5 μs
= 15 μs = 9.0 bytes
ωn = 300 Krads/sec → 17 μs + 3.3 μs
= 20 μs = 12.5 bytes

3.5.1. Crystal Acquisition

Analysis of the crystal acquisition time is similar to the read acquisition time. In the case of the National DP8465 data separator, however, a high bandwidth mode is provided to decrease the acquisition time. The high bandwidth is activated when $\overline{\text{SET PLL LOCK}}$ is deasserted. This increases the phase detector gain (increases the charge pump current). When the phase detector gain increases, both the loop bandwidth and the damping factor are increased. Loop performance is poor if the damping factor gets much larger than 1.0 and therefore the increase in loop bandwidth should be limited to the point where the damping factor is 1.0. This means that:

$$\omega n(\text{high track}) = \omega n(\text{low track})\,(1.0/0.7)$$

Repeating the acquisition analysis for the four bandwidths used before:

ωn = 600 Krads/s (1/0.7) = 857 Krads/s →
acq = 5 bytes
ωn = 500 Krads/s (1/0.7) = 714 Krads/s →
acq = 6 bytes
ωn = 400 Krads/s (1/0.7) = 571 Krads/s →
acq = 7 bytes
ωn = 300 Krads/s (1/0.7) = 429 Krads/s →
acq = 9 bytes

3.5.2. Margin Loss Due to PLL Response

Fast acquisition is desirable to minimize preamble lengths. However, the wider the loop bandwidth, the larger is the loop response to shifted data. Loop response to shifted data results in margin loss. The data is shifted by noise and other factors which contain no information about data frequency changes. When the loop responds to this bit shift it moves the windows for subsequent data bits thereby reducing the amount of shift allowed for these bits.

For a 5 Mbit/sec system with a maximum shifted early bit, the following formula gives the loop response:

$$\text{loss} = 40[1 - (\cos\sqrt{1-\zeta^2}\,\omega nt - \zeta/\sqrt{1-\zeta^2}\,\sin\sqrt{1-\zeta^2}\,\omega nt)\exp(-\zeta\omega nt)] \qquad \text{Note 1.}$$

where 40 is the phase step (in ns) due to the early bit, ωn is the loop bandwidth, and ζ is the damping factor. The phase detector output is active for 40 ns and the amount of loss is determined by setting t = 240 ns which is the time to the far edge of the next window where a bit may appear.

The ωn and ζ for this calculation will be the same as used in the data acquisition analysis as long as t does not exceed 2 VCO cycles. If t is much greater than 2 cycles, the effective phase detector gain is reduced and the ωn is reduced also (see design example of ωn formula). This calculation also assumes that the second pole in the filter is well outside the loop bandwidth.

ωn(max freq data) = 600 Krads/s → loss = 7.7 ns
ωn(max freq data) = 500 Krads/s → loss = 6.4 ns
ωn(max freq data) = 400 Krads/s → loss = 5.2 ns
ωn(max freq data) = 300 Krads/s → loss = 3.9 ns

There are techniques for reducing this margin loss without heavily impacting the acquisition performance. See Section 3.7 for details.

DESIGN EXAMPLE FOR 5 Mbit/s DATA RATE

Although there is no real standard, most of the track formats for small Winchesters are using about 12 bytes of preamble. There is a formated gap after the data ECC field but it cannot be assumed that any of this gap is available for PLL acquisition. Some of this gap will be lost when the sector is updated due to rotational speed variation and the remainder is required for write-to-read recovery of the read channel.

Note 1: Phaselock Techniques; Floyd M. Gardner, Second Edition, John Wiley & Sons; pg. 48.

As discussed, in a soft sectored disk drive, acquisition to the crystal, as well as acquisition to the read data, must occur within the preamble. (In a hard sectored drive some of the preamble may be lost to uncertainty in sector pulse detection but usually most of the preamble is available for data acquisition.) For this reason, the loop bandwidth during acquisition must be in the 600 Krads/sec range for a soft sectored drive. The 600 Krads/sec bandwidth gives 5 byte crystal acquisition and 5.5 byte data acquisition. The margin loss can be held to about 6.0 ns + 7.7 ns = 13.7 ns with the DP8465-3. (This margin loss can be reduced if a longer preamble/lower bandwidth were used or by using some of the techniques discussed in the 10 Mbit/sec design example.)

For the DP8465 with $\overline{\text{SET PLL LOCK}}$ asserted:

$$\omega n = \sqrt{(2.5)\,(F_{VCO})/(N)\,(C1)\,(R_{rate})}$$

Rrate should be set at 820Ω since the current in this resistor does have a small effect on the VCO stability and 820Ω has been determined to be the optimum value. F_{VCO} is the center frequency of the VCO in hertz. F_{VCO} is 10 MHz for 5 Mbit/sec data rates. N is the number of VCO cycles per data bit. MFM preamble data has 2 cycles per data bit. This gives:

$$\begin{aligned} C1 &= (2.5)\,(F_{vco})/(N)\,(Rate)\,(\omega n^2) \\ &= (2.5)\,(10E6)/(2)\,(820)\,(3.6E11) \\ &= 0.042E\text{-}6 \text{ (use } 0.039\ \mu F) \end{aligned}$$

C1 should be an ultra-stable monolithic ceramic capacitor or equivalent timing quality capacitor.

In the data field, the MFM data frequency can be half the preamble frequency. This means that N = 4 in the bandwidth equation. This reduces the bandwidth by $1/\sqrt{2}$:

$$\omega n\,(\min) = (1/\sqrt{2})\,(6.25.1 \text{ Krads/s}) = 442.1 \text{ Krads/s}$$

where 625.2 Krads/sec is the computed bandwidth in the preamble with C1 = 0.039 μF. Since there should be no mechanical resonances anywhere near this frequency, the loop will be able to track the data.

The damping factor should be about 0.5 when ωn is minimum. Response to bit shift is minimized when the damping factor is small; however, if the damping factor drops much below 0.5 the system tends to be oscillatory (underdamped):

$$\zeta = (\omega n)\,(R1)\,(C1)/2 \rightarrow R1 = (2)\,(\zeta)/(\omega n)\,(C1)$$

$$R1 = 1/(442.1E3)\,(0.039E\text{-}6) = 58 \text{ (use } 56\Omega)$$

The actual damping during acquisition is then:

$$\zeta(acq) = (625.2)\,(56)\,(0.039E\text{-}6)/2 = 0.68$$

The linear approximation used to predict loop performance assumes that the phase detector output is constantly proportional to the input phase difference. In reality, the phase detector output is a pulse applied for a period of time equal to the phase difference. The function of C2 is to smooth the phase detector output over the cycle. C2 adds a second pole to the filter transfer function as discussed in section 3.7. This pole should be far enough outside the loop bandwidth that its phase and amplitude contribution is negligible to the loop bandwidth. If:

$$C2 = C1/50 = 789 \text{ pF (use 820 pF)}$$

the acquisition performance and the margin loss are not significantly changed from the predictions. If a larger C2 is used, the margin loss can be reduced at the expense of the acquisition. This may be desirable for some systems. See section 3.7 for discussion of the function of C2.

The final loop component is R_{boost}. When $\overline{\text{SET PLL LOCK}}$ is deasserted, R_{boost} is in parallel with R_{rate} to set the charge pump current. If the parallel combination of R_{rate} and R_{boost} is called R_p:

$$\omega n\,(\text{high}) = \sqrt{(2.5)\,(F_{VCO})/(N)\,(C1)\,(R_p)}$$

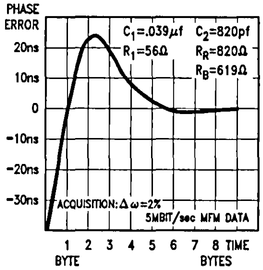
Since the total set current, I_{set}, into the charge pump may not exceed 2 mA, the parallel resistance R_{boost} and R_{rate} (R_p) should also be restricted to:

$$R_p \geq V_{be}/I_{set} = 0.7V/2\text{ mA} = 350\Omega$$

Solving for R_{boost}:

$$\begin{aligned} R_{boost} &= (R_p)\,(R_{rate})/(R_{rate} - R_p) \\ &= (350)\,(820)/(820 - 350) = 610.9 \text{ (use } 619\Omega) \end{aligned}$$

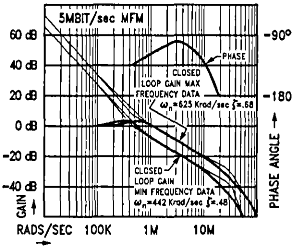
Remember, R_{boost} is switched-in whenever $\overline{\text{SET PLL LOCK}}$ is deasserted. This design example assumes that $\overline{\text{SET PLL LOCK}}$ is deasserted whenever READ GATE is deasserted.



TL/F/8663-59

ωn (acquisition) = 625 Krads/s ζ = 0.68
Lock time ≈ 5 bytes.
ωn (write mode) = 957 Krads/s ζ = 1.04
Lock time ≈ 4 bytes.
ωn (min data) = 442 Krads/s ζ = 0.48

FIGURE 3.14a. 5 Mbit/sec Design Example



TL/F/8663-60

FIGURE 3.14b. 5 Mbit/sec Design Example

DESIGN EXAMPLE FOR 10 MBIT/SEC MFM DATA RATE

At 10 Mbit/sec the window period is reduced to 50 ns and margin loss becomes a much more important design parameter than at 5 Mbit/sec. The total bit shift allowed is half the window (25 ns). If the National DP8465-3 is used, there will be a maximum static loss (independent of the loop filter components) of 6 ns. This part could be used as the starting point for a 10 Mbit/sec design.

The margin loss due to loop response should be kept as low as possible. The maximum allowed bit shift with the DP8465-3 is (25 ns − 6 ns) or about 19 ns. The loss approximation formula is then:

$$\begin{aligned} \text{loss} = 19[1 - (&\cos\sqrt{1-\zeta^2}\,\omega nt \\ &- \zeta/\sqrt{1-\zeta^2}\,\sin\sqrt{1-\zeta^2}\,\omega nt)\exp(-\zeta\omega nt)] \end{aligned} \qquad \text{Note 1.}$$

Note 1: Ibid.

where 19 is the phase step (in ns) due to an early bit. Evaluation should be done with t = 120 ns:

$$\omega n = 600 \text{ Krads/s and } \zeta = 0.66 \longrightarrow \text{loss} = 1.9 \text{ ns}$$

This 1.9 ns loss should be acceptable for most designs. The total loss would then be 6 ns + 1.9 ns = 7.9 ns for a total window of 34.2 ns with the DP8465-3. As discussed in Section 3.7, the actual margin loss due to loop response will be less than 1.9 ns due to the roll off of the second pole in the loop filter.

The acquisition performance of a 600 Krad/s loop was analyzed in the 5 Mbit/sec design example. With a 2% frequency difference between the read data and the crystal, the peak phase error was about 24 ns. At 10 Mbit/sec $\Delta\omega$ is doubled and the conversion term becomes 100 ns/2π. The initial phase misalignment is 25 ns. The peak phase error is then just over 20 ns. This is pushing the capture range of the DP8465-3 since its allowed shift is only 19 ns. If the system is fixed media with 1% or better rotational speed variation there is no problem since the peak phase error is reduced by a factor of two in this case. If capture is needed at 2%, $\overline{\text{SET PLL LOCK}}$ can be deasserted during read acquisition as discussed later.

Notice that the 5 Mbit/sec lock time was given in bytes (i.e. 5.5 bytes). At 10 Mbit/sec the read acquisition time will remain about the same but that means that the number of bytes is doubled to 11 bytes. This read acquisition time can be reduced without increasing the margin loss if the disk controller will hold $\overline{\text{SET PLL LOCK}}$ deasserted (high) during read acquisition. This increases the damping factor to 1.04 and the bandwidth to 940 Krads/sec. The result is a read acquisition time of 8 bytes. The crystal acquisition time is also 8 bytes so the total preamble length is 16 bytes in a soft sectored disk drive. (This assumes zero monitoring overhead. If x bytes are required to determine that the read data is not preamble, the total preamble length would be 16 + x bytes.) Of course the price paid for this reduction in acquisition time is that the controller must now control the $\overline{\text{SET PLL LOCK}}$ input during acquisition.

The calculation of the loop components is similar to the 5 Mbit/sec data rate example:

$$C1 = (2.5)\,(F_{VCO})/(N)\,(R_{rate})\,(\omega n^2)$$
$$= (2.5)\,(20E6)/(2)\,(820)\,(3.6E11)$$
$$= 0.085E\text{-}6 \text{ (use } 0.082\ \mu F \text{ (5\%))}$$

C1 should be an ultra-stable monolithic ceramic capacitor or equivalent timing quality capacitor. Evaluation of the bandwidth formula with C1 = 0.082 μF gives:

$$\omega n = \sqrt{(2.5)\,(20E6)/(2)\,(820)\,(0.082E\text{-}6)} = 609.8 \text{ Krads/s}$$

The target damping factor at this bandwidth is 0.66:

$$R1 = (2)\,(\zeta)/(\omega n)\,(C1)$$
$$= (2)\,(0.66)/(609.8E3)\,(0.082E\text{-}6) = 26.4 \text{ (use } 27\Omega)$$

The total set current into the charge pump through the parallel combination of R_{boost} and R_{rate}, (R_p), must not exceed 2 mA. For $I_{set} \leq Vbe/R_p$, then:

$$R_p > Vbe/I_{set} = 0.7V/\ 2\text{ mA} = 350\Omega$$

Solving for R_{boost}:

$$R_{boost} = (RP)\,(R_{rate})/(R_{rate} - R_p)$$
$$= (350)\,(820)/(820-350) = 610.6 \text{ (use } 619\Omega)$$

This design example assumes that R_{boost} is switched in during crystal acquisition and data acquisition.

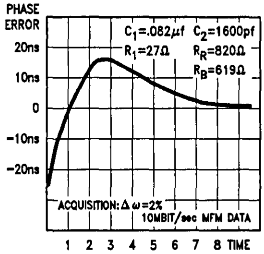
If the margin loss due to loop response of 1.9 ns is acceptable and acquisition performance is to remain unaffected:

$$C2 \ll C1/50 = 0.082\ \mu F/50 = 1640 \text{ pF (use 1600 pF)}$$

A smaller value of C2 is chosen for this example since the 1.9 ns of margin loss seems very acceptable whereas it is undesirable to impact the capture range or the acquisition time. Larger values of C2 may be used, however, the additional pole in the filter may begin to affect frequencies inside the loop bandwidth. The major impact of large C2 is on the capture range although some degradation in read acquisition time can be seen for larger values of C2. See section 3.7 for further discussion.

3.6 COMMENTS ON OTHER CODES

MFM is a 1,3 RLL code. This means that a minimum of one empty window will occur between two windows which each contain a disk data bit, and that a maximum of three empty windows will occur between windows which each contain a disk data bit. The most popular of the newer RLL codes are the 2,7 codes, in which there are a minimum of 2 and a maximum of 7 windows between windows containing a disk data bit. Although the ratio of encoded disk data bit positions (windows) on the disk to non-encoded data (NRZ) bits for both MFM and 2,7 code is 2:1, the two codes differ in the actual number of recorded pulses required to store a given number of NRZ bits (their NRZ-bit versus disk-bit ratio, or efficiency). The 2,7 code requires fewer recorded bits than MFM on average for disk encoding of the same amount of information and has a 50% larger minimum bit spacing than MFM. These allow, on a given disk, a theoretical increase in data storage of 50% when 2,7 encoding is chosen over MFM while the minimum flux transition spacing is kept constant.



TL/F/8663-61

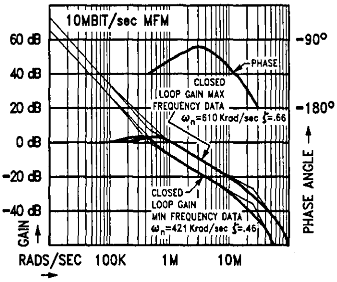
Controller must keep $\overline{\text{SET PLL LOCK}}$ deasserted for 8 bytes after READ GATE is asserted.

ωn (acquisition) = 933 Krads/s ζ = 1.03

ωn (min data) = 431 Krads/s ζ = 0.48

ωn (max data) = 610 Krads/s ζ = 0.66

FIGURE 3.15a. 10 Mbit/sec Design Example

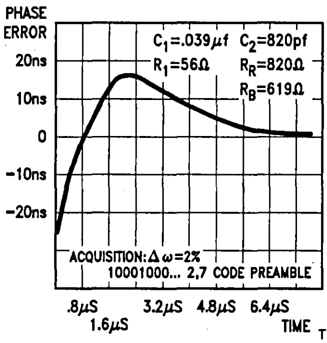


TL/F/8663-62

FIGURE 3.15b. 10 Mbit/sec Design Example

The impact of a 2,7 code on the data separator is significant. Loop bandwidth is dependent upon the sample rate into the phase detector. With MFM it has been seen that the bandwidth in minimum frequency data is $1/\sqrt{2}$ times the bandwidth in maximum frequency data. This is a ratio of 1:0.707. In a 2,7 code the ratio is $\sqrt{3/8}$ or 1:0.612. The damping factor follows the bandwidth so a 2,7 code system must be more carefully designed to avoid underdamped or overdamped response.

Another complexity of 2,7 codes is that most systems are not using the maximum frequency data in the preamble. The 100100 .... encoded data does not decode to a data pattern with byte alignment. The 10001000 ... encoded data pattern decodes to all ones byte aligned data and is being used more often. The lower frequency data increases the read acquisition time and increases the probability of harmonic lock. Channel induced pulse pairing (from channel asymmetry), coupled with an initial phase alignment which puts the data bit at the extreme window edge, may allow the loop to stabilize out of phase. There are two techniques which are used to eliminate this problem. The first is to start the VCO in phase with the read data (zero phase start-up). The second technique is to perform phase and frequency comparison (i.e. do not window the data) during read acquisition. The second technique is used on the National DP8462 data separator which is specifically designed for 2,7 codes.



TL/F/8663-63

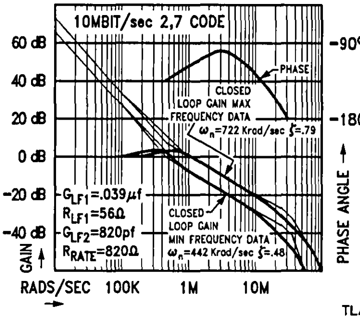
Controller must keep $\overline{\text{SET PLL LOCK}}$ deasserted for 6 µs after READ GATE is asserted.

ωn (acquisition) = 956 Krads/s ζ = 1.04

ωn (min data) = 442 Krads/s ζ = 0.48

ωn (max data) = 722 Krads/s ζ = 0.79

FIGURE 3.16a. 10 Mbit/sec 2,7 Code Example



TL/F/8663-64

FIGURE 3.16b. 10 Mbit/sec 2,7 Code Example

Note 1. Ibid.

In conclusion, loop filter design for 2,7 codes is more difficult than for MFM. In hard sectored drives, or pseudo hard sectored drives with dc erased gaps, the problem is simply the wide range of damping factors and can be easily solved.

3.7 LOOP FILTER DETAILS

Time domain response for a second order feedback control system is well known. The response to a phase step and a frequency step is shown in the graphs in *Figure. 3.17.* The phase locked loop system is normally designed to have a damping factor between 0.7 and 1.0 during acquisition so these curves show the performance boundaries. The equations for the phase step response are:

$$\zeta < 1 \longrightarrow \theta(t) = \Delta\theta(\cos\sqrt{1-\zeta^2}\,\omega nt - \zeta/\sqrt{1-\zeta^2}\sin\sqrt{1-\zeta^2}\,\omega nt)\exp(-\zeta\,\omega nt)$$

$$\zeta = 1 \longrightarrow \theta(t) = \Delta\theta(1 - \omega nt)\exp(-\omega nt) \qquad \text{Note 1}$$

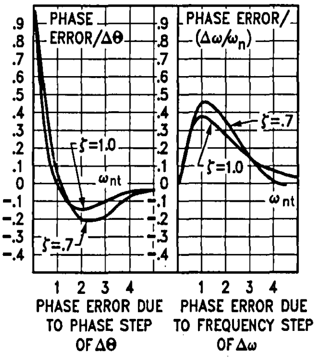
The equations for the frequency step response are:

$$\zeta < 1 \longrightarrow \theta(t) = (\Delta\omega/\omega n)\,(1/\sqrt{1-\zeta^2}\sin\sqrt{1-\zeta^2}\,\omega nt)\exp(-\zeta\,\omega nt)$$

$$\zeta = 1 \longrightarrow \theta(t) = (\Delta\omega/\omega n)\,(\omega nt)\exp(-\omega nt) \qquad \text{Note 1.}$$

These equations were used to derive the margin loss due to bit jitter (i.e. phase steps) as well as the acquisition performance in the previous design examples. The second order time domain response is the usual starting point for loop filter design.

Disk data separation is complicated by the fact that the input data stream is not a single frequency. Missing data bits must not be allowed to generate error signals to the loop since the loop bandwidth would have to be set unacceptably low to filter out this erroneous information. As a result, most phase detectors for disk data separators do not generate a continuous voltage dependent upon the input phase difference. The discontinuity in the input data stream is dealt with by only generating a phase detector output when a data bit has arrived and only during the period of time corresponding to the input phase difference.

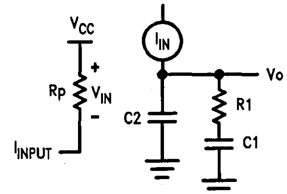


TL/F/8663-65

FIGURE 3.17. Phase Error as a Function of Damping and Phase Step

There are some problems which result from a pulsed phase detector output. First, oscillator response can be affected by the phase relationship between these pulses and its internal charging and discharging cycle. Second, the response of the oscillator occurs in a quantum jump when the error pulse is generated, reducing margin for later bits. These problems are reduced by adding one or more poles to the loop filter. These poles smooth the phase detector output and reduce the loop response to bit jitter. As long as the added poles do not add significant gain loss and phase shift within the loop bandwidth, the system time domain response will not differ appreciably from a pure second order feedback control system.

The voltage output of the National DP8465 phase detector is converted to a current which is sourced into the filter:



TL/F/8663-66

FIGURE 3.18 Charge Pump and Loop Filter Equivalent Circuitry.

If Zf is the input impedance of the loop filter:

$$Zf = (1/sC2) // (1/sC1 + R1) =$$

$$(1 + sC1R1)/sC1(sC2R1 + C2/C1 + 1)$$

$$Vo = (Iin)(Zf) =$$

$$(Vin/R_p)(1 + sC1R1)/sC1(sC2R1 + C2/C1 + 1)$$

$$Vo/Vin = (sC1R1 + 1)/sC1R_p(sC2R1 + C2/C1 + 1)$$

The effect of C2 is to introduce a pole into the transfer function of the loop filter. The pole location is where $s = (1 + C2/C1)/C2R1 = 1/C2R1$ if $C2 << C1$.

The open loop gain is the product of the phase detector, the filter, and the oscillator transfer functions. The phase detector transfer function is simply a constant, K_{pd}, in volts/radian. The oscillator transfer function is K_o/s where the 1/s term represents phase as the integral of frequency and K_o is in units of radians/(volt x sec). The open loop transfer function is then:

$$G(s) = (K_oK_{pd})\,[st2 + 1]/[st1(st3 + 1)s]$$

where $t1 = C1R_p$, $t2 = C1R1$, and $t3 = C2R1$. The transfer function has a zero at 1/t2, a pole at 1/t3, and two poles at the origin.

Note: For the National DP8465 the K_oK_{pd} product is (2.5) $(F_{VCO})/N$ where F_{VCO} is the oscillator frequency in Hertz and N is the number of oscillator cycles between data bits.

In the frequency domain, the open loop gain falls at 40 dB/decade until equal to $K_oK_{pd}/t1$ at $j\omega = 1$. The 40 dB/decade slope continues until the zero at $j\omega = 1/t2$. At the zero the slope changes to 20 dB/decade. The slope returns to 40 dB/decade when the pole breaks at $j\omega = 1/t3$. The phase shift begins at −180 degrees and asymptotically approaches −90 degrees. The phase is equal to −135 degrees at $j\omega = 1/t2$. The phase plot turns around and starts back toward −180 degrees as $j\omega$ approaches 1/t3 such that at $j\omega = 1/t3$ the phase equals −135 degrees.

If a second pole were in the filter around $j\omega = 1/t3$, the phase would equal −225 degrees at $j\omega = 1/t3$ and stability would require that the gain be below 0 dB before the phase reached −180 degrees. This sometimes limits how closely the poles can be moved to the loop bandwidth. When the gain is 0 dB, the difference between the actual phase shift and 180 degrees is referred to as the phase margin. The open loop phase margin is related to the damping factor of the second order system. Note that there is a pole in the buffer amplifier of the DP8465 between the filter and the VCO. This pole is at 5 MHz or about 31.4 Mrads/s and could affect the phase margin in a very wide band loop.

The additional pole in the loop filter helps to improve read margin because it lowers the loop gain at the frequency of the bit jitter. The fundamental frequency content of the bit jitter is slightly below the bit frequency since the pump up error begins before the end of the window and the pump down error ends after the end of the window. At 5 Mbit/sec, the bit jitter frequency is 1/(200 ns + 80 ns) = 22.4 Mrads/sec for 40 ns bit shifts. The 615 Krads/sec loop designed earlier for 5 Mbit/sec data would have a gain of −28 dB at $j\omega = 22.4$ Mrads/s if C2 were not in the loop. With C2 = 820 pF, the gain is reduced to about −31 dB at the bit jitter frequency. If the additional pole were added at $j\omega = 6.15$ Mrads/s (10 ωn) the gain would be −38 dB at the bit jitter frequency for an extra 10 dB of noise rejection. C2 would be 0.0027 μF in this case.

It is difficult to analytically predict the effect of C2 on the acquisition performance. Since C2 is moving close to the loop bandwidth, the system behavior is not purely second order. There are some computer programs which allow time domain response to be predicted for third order systems but normally it is not necessary to use these tools. It is usually sufficient to start with a second order analysis and experimentally measure the system performance as the third pole (or poles) is brought closer to the loop bandwidth.

CHAPTER 4 DP8466 Disk Data Controller Overview

4.0 INTRODUCTION

National's Disk Data Controller (DDC) chip, DP8466, performs many of the functions in the disk data electronics path of either disk controllers or intelligent disk drives. The primary function of the chip is to correctly identify the selected sector on disk and then to transfer that sector's data to or from memory.

The DDC performs serialization and deserialization of disk data, CRC/ECC generation, checking and correction, data buffering with a 16-word (32-byte) FIFO, and single or dual channel DMA addressing. It can write NRZ or MFM encoded data to the disk. The data separation required in the disk data path electronics can be obtained by using one of National's Data Separator chips, DP8460 or DP8461/5. If 2,7 is used instead of MFM, 2,7 ENDEC chip could be used in conjunction with the DP8462 2,7 Data Synchronizer. The DDC is fabricated using the dual layer metal 2μ microCMOS process, which allows complex functions to be implemented with high operating speeds and modest power consumption. Internal gate delays of less than 2 ns allow the DDC to function with disk data rates up to 25 Megabits/sec. This enables the DDC to be used not only with 3½-inch, 5¼-inch and 8-inch drives, either Winchester or Floppy (or both), but also with high-end drives such as 14-inch Winchester drives, vertically recorded drives, and optical drives.

The DDC interfaces with drives compatible with the ST506, ST412HP, ESDI, SMD and other interfaces. Also the DDC may instead be part of an intelligent disk drive that has a SCSI (SASI), or an IPI type interface. Refer to chapter one where the block level boundaries of the various disk interface standards are shown.

4.1 THE DDC ARCHITECTURE AND BASIC OPERATION

An architectural block diagram of the DDC is shown in *Figure 4.1*. The 64 internal registers consist of control, command, pattern and count registers. These registers are initially preloaded with information such as header or synch bytes, ECC polynomial bytes, preamble or postamble patterns, or address marks (for soft sectored drives) etc. Some of the registers will be programmed each time the DDC starts an operation, for example the command register.

The DP8466 has a range of commands that enable reading and writing of both data, and header fields, checking for header fields, formatting with either hard or soft sectored formats, and aborting. Each of these operations can be performed in various modes and an abundance of formats. Most operations can be performed as single or multi-sector operations.

In a typical disk read or write operation, the desired sector (where the data information is to be read from or written to) is first located by comparison of the header bytes. To achieve this comparison, the incoming serial data from the external data separator is deserialized into byte-wide data that is fed both to a comparator and FIFO. The comparator checks the address mark (if present) and the synch bytes to align the incoming bytes. Once the incoming data stream has been byte aligned, header comparison for the desired sector then begins. As each header byte is deserialized it is compared with the next preloaded header byte. If any of the header bytes do not match, the desired sector has not been located, but the DP8466 still performs a CRC/ECC check on the header and waits for the ID segment of the next

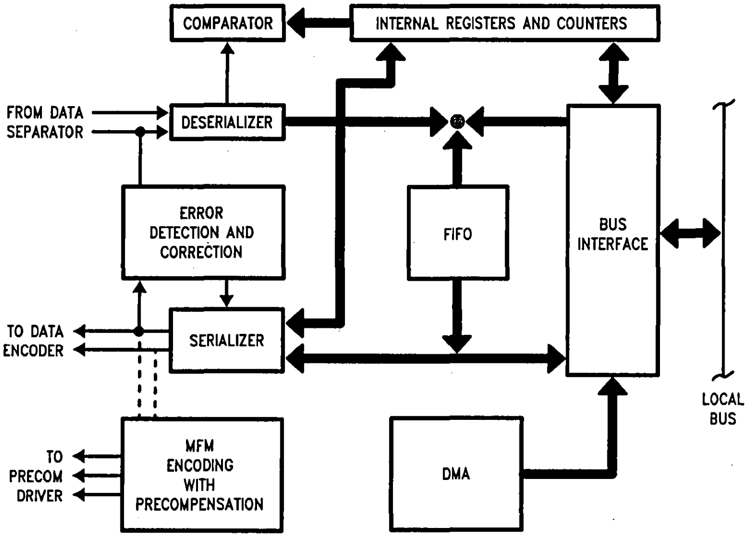


FIGURE 4.1. The DDC Data Path Architecture

sector. If after two disk revolutions the header is not found the DDC will abort the operation. Once a header match is detected, the DDC prepares to transfer data to or from the data segment of the sector.

1. When writing a data segment to the disk, the DDC first inserts the preamble pattern field, address mark (for soft sectored drives) and synch fields, each byte repeated a specified number of times. These fields are followed by the data field bytes that are provided sequentially through the FIFO and external memory. Internal CRC or ECC (or external ECC) check bits are generated from the bits in the data field and subsequently appended to it. The write operation ends with the postamble. As each byte is serially transmitted, the next byte becomes available to be serialized and output. The serial output may be either NRZ data with the associated write clock or MFM encoded data.

2. When reading a data segment from the disk, the DDC deserializes the incoming data and byte aligns with the address mark (if present) and synch fields using the comparator. It then transfers the data field bytes into FIFO. At the same time it checks incoming serial data using an internal CRC code or ECC code.

For both read or write operations, disk data goes to or from the internal FIFO. Once the FIFO has filled or emptied to the selected threshold level, data may be transferred to or from the external buffer memory in the selected burst length by means of a DMA channel. A second DMA channel is available to transfer data to or from buffer memory to the system (for systems that utilize a buffer memory).

If the operation terminates properly an interrupt is issued, and the user may check status. If an error results during the operation the DDC will also interrupt the microprocessor, and the user must determine the appropriate action.

The DDC can be configured in three different modes; peripheral, master and slave. In the peripheral mode, the microprocessor accesses internal register to read or write data. The DDC acts like a peripheral when it is being configured, and when the microprocessor issues a command. During the execution of a command and when the on-chip DMA has been granted access to the bus for local and remote transfers, the DDC goes into its master mode, and becomes bus master. If during the command an external DMA controls data transfer the DDC will go into a slave mode.

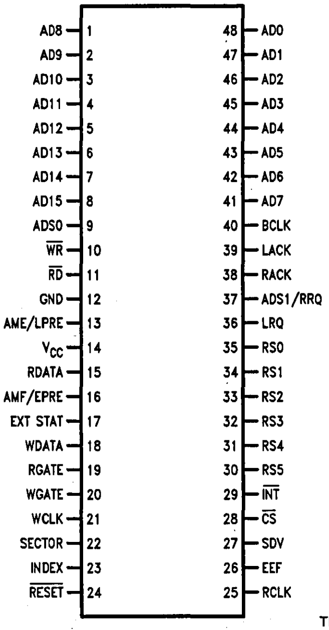
4.2 PIN ASSIGNMENT AND DESCRIPTION

In this section a complete pin description is presented. The pin assignment diagrams are shown in *Figure 4.2(a)* and *4.2(b)*. Specific timing information for these signals can be found in the DP8466's Datasheet.

4.2.1 Bus Interface

Chip Select (CS): When the DDC is in the peripheral mode the chip select signal must be asserted low to access enable microprocessor access. In the peripheral mode pins RS0–5 are address inputs and pins AD0–7 are set for 8-bit transfer of data between the DDC and microprocessor. CS has no effect if on-chip or external DMA is performing a transfer. (DDC in slave or master modes.)

Bus Clock (BCLK): The DDC uses BUS CLOCK input as the reference clock when the DDC is bus master. It is used only during RESET and DMA operations and is independent of the disk data rate. BCLK may be the microprocessor clock and must be at least ¼ the rate of READ CLOCK, RCLK.



TL/F/8663-68

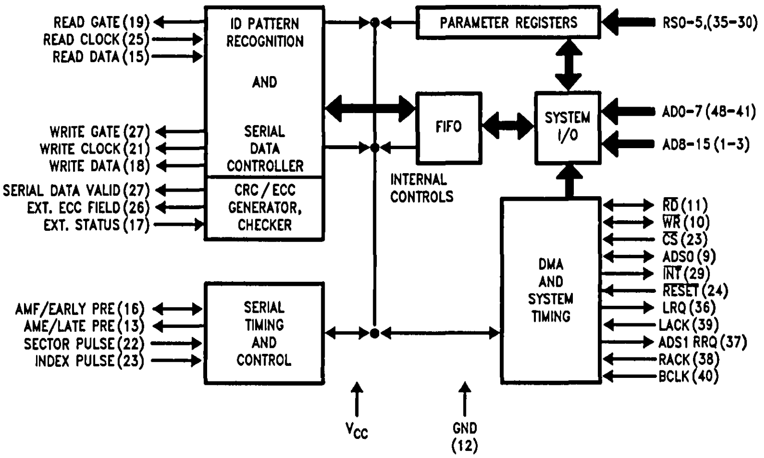
FIGURE 4.2(a). The DDC (DP8466) Connection Diagram

Address/Data (AD0–7): This 8-bit data/address bus port has one microprocessor associated function and four memory transfer associated functions. When the DDC is in the peripheral mode and CS is set low, this port transfers data between the internal sections of the DDC and the microprocessor. When external DMA is active (i.e. the DDC is in slave mode) with LACK (local acknowledge) set low, data bits D0–7 are transferred between the FIFO and memory.

When the DDC is controlling the bus (i.e. when on-chip DMA is active), the AD0–7 bus is multiplexed between DMA address and FIFO data bits. Using the single DMA mode, A0–7 are issued on this port as are A16–23. When using dual DMA, these lines are used to transfer both the local and remote DMA address bits, A0–7. In either dual or single channel mode, the data bits D0–7 are transferred between FIFO and external memory through this port.

Address/Data (AD8–15): This 8-bit I/O port has four memory transfer functions (in the peripheral mode with CS low, these pins remain indeterminant low impedance). In the slave (external DMA active) mode with LACK set low, data bits D8–15 are transferred between the FIFO and memory when 16-bit transfers are enabled. When the DDC is controlling the bus, (master mode) it issues address bits A8–15 on this port and can also issue address A24–31 if it is in single channel DMA mode.

Register Select (RS0–5): In the peripheral mode, these 6 inputs are used to select the internal registers to be accessed by the microprocessor. These inputs feed "fall through" latches that are controlled by the ADS0 input. The RS0–5 inputs fall through and are decoded by the DDC when the input level on ADS0 pin is high. The RS0–5 inputs are stored on the falling edge of ADS0. This enables easy connection to either multiplexed or non-multiplexed buses.



TL/F/8663-69

FIGURE 4.2(b). The DDC Block Diagram with Pin Assignment

Address/Data Strobe 0 (ADS0): This I/O strobe has two functions. In the peripheral mode, ADS0 becomes an input and may be used as a microprocessor address strobe input. In this mode when ADS0 is high, the address bits on RS0-5 enter the Register Select Latch and are latched on low going transition.

With the bus under DDC control, ADS0 becomes an output that issues the address strobe to external memory at the start of the DMA transfer cycle. The low going transition of ADS0 coincides with the DMA address bits A0-15 being valid on pins AD0-15. (Note: ADS0 when an output will still latch "data" into the RS0-5 latches. Normally this is random data and is of no consequence since $\overline{CS}$ is high. However when the system wants to access the DDC after a DMA, the proper address must be loaded into the latches or ADS0 must be high prior to $\overline{CS}$ going low.)

Address/Data Strobe 1 / Remote Request (ADS1/RRQ): This output pin can be configured to have one of two functions. If the DDC has been configured for 32-bit address single channel DMA, the pin becomes ADS1. This address strobe is issued either at the start of the very first memory transfer cycle of a new disk operation or when the lower 16 address bits have just rolled over. In either case the address on pins AD0-15 is A15-31 at the instance of low going transition of ADS1.

If the DDC is configured to perform remote DMA transfers in its dual channel mode, the pin becomes RRQ, or Remote DMA Request. The DDC will assert the RRQ high whenever it is ready to transfer data between buffer memory and the system I/O port. The RRQ will be reset at the end of the selected burst transfer length.

Remote DMA Acknowledge (RACK): This input pin must be asserted high after RRQ has been set high, when the external I/O device is ready to transfer data. Data will be transferred between external buffer memory and the remote I/O device until the DDC sets RRQ low, or until RACK is set low externally. If RACK is removed during a transfer any cycle in progress at this time will complete.

Local DMA Request (LRQ): This output pin is low when no data transfer is required between FIFO and the external buffer memory. The DDC asserts LRQ high when FIFO requires data to be transferred to or from the external buffer memory for both dual and single DMA mode. LRQ will return low during the last cycle of the burst transfer or on emptying or filling up the FIFO.

Local DMA Acknowledge (LACK): Once the DDC has set LRQ high, and the external circuitry subsequently sets the LACK input high the DDC becomes bus master. Data transfers between the on-chip FIFO and external memory will now proceed until the DDC sets LRQ low, or until LACK is set low externally, in which case any cycle in progress at this time will complete.

$\overline{\text{Read}}$ ($\overline{\text{RD}}$): The $\overline{RD}$ strobe I/O pin has two functions. In the peripheral mode $\overline{RD}$ is an input that when low causes data from a DDC register (selected through pins RS0-5) to be output on pins AD0-7 when CS is low. Pins AD0-7 will be high impedance before and after the READ strobe.

When the DDC has bus control, with either LACK or RACK active, the $\overline{RD}$ strobe is an active low output from the DDC to be used by external memory. It enables data to be transferred from the selected memory location to either the on-chip FIFO for local transfers or to an external I/O device for remote transfers.

$\overline{\text{Write}}$ ($\overline{\text{WR}}$): The $\overline{WR}$ strobe I/O pin has two functions. In the peripheral mode WR is an input that when set low will cause data present on pins AD0-7 to be loaded into a DDC register (selected through pins RS0-5, and when $\overline{CS}$ is low).

When the DDC has bus control, with either LACK or RACK active, the $\overline{WR}$ strobe is an active low output used to write data to the selected memory location from either the on-chip FIFO for local transfers or from an external device for remote transfers.

Interrupt (INT): The INT output is set active low whenever the DDC wishes the controlling microprocessor to check status. It is set high when the microprocessor services the interrupt by setting the $\overline{CS}$ input low and reading the Status Register.

Reset: The RESET pin is an input that is normally set high. When RESET is set low, the DDC goes into the internal reset mode. It clears the FIFO contents, the Status and Error registers and also deactivates LRQ, RRQ, WRITE GATE and READ GATE.

4.2.2 Disk Interface

Write Gate (WGATE): When the DDC writes data to the disk during either a Write Data, Write Header or Format operation, it asserts the WRITE GATE output pin active high at the start of the operation. The transition coincides with the first WRITE DATA bit being issued. WRITE GATE remains high until either the last data bit of the sequence to be written has ended, or when the DDC aborts or resets.

Write Data (WDATA): During a write disk operation, this pin outputs serial disk data. The DDC can be configured to output either NRZ or MFM encoded data on the WRITE DATA pin. If NRZ data is selected its bit rate has the same period as the WRITE CLOCK output. When the DDC has been configured to issue MFM encoded data to the disk, data pulses will be output on WRITE DATA as determined by the MFM encoding rules. In either configuration, when WRITE GATE is inactive low, so is WRITE DATA.

Write Clock (WCLK): When the DDC is configured to write NRZ data, a synchronous clock is provided at the WRITE CLOCK output. The WRITE CLOCK frequency is the same as READ CLOCK. When the DDC is configured to output MFM encoded data, clock information is not needed however this output will still toggle.

Address Mark Found/Early Precompensation (AMF/EPRE): This pin has two modes. When the DDC is configured to read from a soft sectored disk, this pin is ADDRESS MARK FOUND, an active high input. Normally this input will be low but whenever external circuitry detects an Address Mark (such as a missing clock, or blank information) AMF should go high for at least one period of READ CLOCK. This will indicate to the DP8466 that a valid address mark has been located.

When the DDC is configured to write MFM encoded data, this pin becomes the output EARLY PRECOMPENSATION. When it is high, the MFM pulse appearing on the WRITE DATA output requires early precompensation. When low, the MFM pulse does not require early precompensation.

If both functions are being used in the system, WRITE GATE is used to determine the function of this pin. When WRITE GATE is active high, this pin is an LPRE output, otherwise it is an AMF input. External demultiplexing circuitry can be used.

Address Mark Enable/Late Precompensation (AMF/LPRE): This pin has two modes of operation depending on whether NRZ or MFM data is written to the disk. When the DDC is configured to write NRZ data on a soft sectored disk, this output pin is ADDRESS MARK ENABLE. AME is normally low and will remain low when the DDC is configured for a hard sectored disk (bit HSS in disk format register is set). On the other hand, for the soft sector configuration, the DDC will set AME active high during the time any Address Mark byte is serially output on WRITE DATA pin.

When the DDC is configured to write MFM encoded data, this pin becomes LATE PRECOMPENSATION output. In this configuraton if LATE PRECOMPENSATION is high, then late precompensation is required on the MFM pulse being output on WRITE DATA. If LATE PRECOMPENSATION is low, late precompensation is not required. If both EARLY PRECOMPENSATION and LATE PRECOMPENSATION output pins are set low, no precompensation is required.

Read Gate (RGATE): When the DDC is set to read the disk, such as in a Read Header, Compare Header, Ignore Header, Read Data or Ignore Data operation, READ GATE will go active high. This informs external data separator that it can begin locking on to incoming disk data. If the data separator fails to achieve locking, the DDC will set READ GATE inactive for 18 bit times before another locking attempt is made. READ GATE is also set inactive either at the end of the specified operation or if the DDC aborts or resets.

Read Data (RDATA): Once READ GATE has been set active high and external circuitry has locked on to the incoming encoded disk data, the encoded data must be separated into clock and NRZ data. The NRZ data connects to the READ DATA input of the DDC, and is clocked into the DDC on the positive edge of READ CLOCK. When READ GATE is set low, the READ DATA input will be ignored.

Read Clock (RCLK): READ CLOCK is a clock input that may have slightly differing frequencies, depending on the READ GATE control pin. When READ GATE is inactive, this clock should be derived from either a servo clock or a crystal clock to produce a clock with a period close to the bit rate of the disk data. After READ GATE has been set active, and external circuitry has locked on to the incoming encoded disk data, the READ CLOCK input must switch frequency (without any short pulses or glitches) to a period identical to the READ DATA signal.

Sector Pulse (SPULSE): In hard sectored drives the SECTOR PULSE input goes high as the start of each sector passes under the drive head. Once the DDC detects this high signal (for at least one period of the READ CLOCK input), it interprets this to indicate a sector operation can begin. In soft sectored drive there is no sector pulse and the start of each sector must be indicated by an Address Mark byte or bytes, this pin should be tied to ground.

Index Pulse (IPULSE): All drives have an index pulse output that goes active high as the beginning of any track passes under the drive head. Once the DDC detects this high signal (for at least one period of the READ CLOCK input), it assumes an INDEX PULSE has occurred. The DDC uses the INDEX PULSE input to begin various operations.

Serial Data Valid (SDV): This output pin goes high whenever the DDC is issuing or receiving either header field bytes and internal header CRC or ECC bytes, or data field bytes and internal data CRC or ECC bytes. It is set high synchronous with the first header or data bit appearing on the WRITE DATA output pin, or the READ DATA input pin. If the encapsulation mode is set then SCV is set high synchronously with the first sync byte (address mark). If the start with address mark bit is set and encapsulation is enabled SDV will be set high at the first sync #2 byte. (See Chapter 5 and 6.2.) It is set low synchronous with the last bit of internal CRC or ECC field ending on the WRITE DATA output pin or the READ DATA input pin. If internal CRC or ECC is not selected, it will be set low synchronous with the last bit of the header field or data field. The SERIAL DATA VALID pin may be used to select external ECC circuitry or for diagnostics in checking the lengths of the fields.

External ECC Field (EEF): This output pin is normally low, but will go high at specified times if external ECC has been selected. This will be during the time the external ECC field check bits need to be generated (with WRITE GATE high) or checked (with READ GATE high). It will be deasserted (synchronous with SERIAL DATA VALID output going low) after the last bit of the external ECC field has ended.

External Status (EXT STAT): The EXTERNAL STATUS input pin has three possible functions: Enabling wait states for the DMA, or Supplying external synchronization information and/or ECC information to the DDC. The user selects either the first alternative, or the other two. In other words, generating wait states is mutually exclusive with external synch and ECC. If the wait state alternative is selected, the use of this status pin is limited to only supplying wait states to the DMA bus cycle. If the latter two alternatives are selected, input signals on EXTERNAL STATUS may provide synchronization at the start of a header or data field, and external ECC error status at the end of the external ECC field.

4.2.3 Power Supply

V_{CC}, GND: The supply pins require a standard +5V ±10% regulated supply. As with any high speed controller that must connect to high speed buses, output switching transients can cause supply noise glitches which can affect other circuitry within the IC. Thus, a good ceramic decoupling capacitor is recommended to be connected across these pins. This capacitor should be ≥0.1 μf and should be located in close proximity to the V_{CC} and ground pins. Good GND and VCC planes are also recommended. Both of these precautions are to minimize the effects of current switchings on the chip affecting sensitive sections of the chip. With inadequate decoupling or GND and VCC planes, inexplicable behavior of the chip may result.

4.3 DDC FUNCTIONAL DESCRIPTION

This section is intended to provide a block level functional overview of the DDC. The detailed operational information is given in Chapter 7. A block diagram of the DDC is shown in *Figure 4.1*. The DDC is composed of a bus interface unit which communicates with the microprocessor and memory. It also is composed of a serializer and a deserializer that communicates with the disk. A single/dual channel DMA block provides intelligent on-chip data transfer. This DMA controller transfers data to and from the internal multi-mode FIFO block. A 32/48 bit ECC or 16-bit CRC correction block is included for error generation and checking of disk data. The functional description of each block follows.

4.3.1 Bus Interface

This block of the DDC provides an interface between the DDC and system bus through its two input/output data bus ports (AD0–7, AD8–15) and one input port (RS0–5). In the peripheral mode, the internal registers of the DDC are selected through pins RS0–5 and data is transferred between microprocessor and the internal registers through the I/O port AD0–7.

When the DDC is controlling the bus, two I/O ports (AD0–7, AD8–15) provide 16-bit address both for the local and remote DMA data transfers. In single channel DMA mode, an address up to 32 bits could be obtained to access memory up to 4 Gigabytes (see the DMA block description section). Multiplexed along with the DMA address information on these two ports is data information. In the 8-bit transfer mode only AD0–7 is used, and the interface logic contains a multiplexer to convert 16 internal data bits to 8 bits. For 16-bit transfers, AD0–7 transfers the lower 8 bits and AD8–15 transfers the upper 8 bits between the FIFO and the system bus.

4.3.2 Internal Registers

The DDC has 64 internal registers including parameter, pattern and count registers. Some of these registers are read-only, some write-only and the remainder read/write. These registers can be classified in four categories:

1) Command and Control Registers

2) ECC/CRC Registers

3) Format Registers

4) DMA Registers

Each of the above mentioned classes is described in the following paragraphs. A list of the DDC's internal registers with their hexadecimal addresses is given in Table 4.1.

COMMAND AND CONTROL REGISTERS

The Command and Control registers are the key registers of the DDC. They control basic functions and operations of the chip. The registers which can be included in this category are Drive Command (address 10H), Operation Command (address 11H), Status (address 00H), Error (address 01H), Disk Format (address 35H), Sector Counter (address 12H), Number of Sector Operations (address 13H), Header Byte Count/Interlock (address 0FH), and Header Diagnostic Readback (address 36H). Table 4.2 lists these registers along with a short description.

The Drive Command register basically determines the operations to be performed on the disk data. Also it can be used to set the DDC to format drives and to abort any operation in progress. The operations determined by the drive command register can then be controlled through the Operation Command register. The Operation Command register enables the DDC to issue certain interrupt and acknowledge signals during different operations. The Status register gives the status of the operation while the Error register indicates errors which may occur during these operations.

The DDC is adapted to the selected drive format through the Format register which determines the format of the information to be written to the disk. The Start Sector, Number of Operations, and Header Byte count registers, in conjunction with the Drive Command register, allow the DDC to perform multisector operations. The Header Diagnostic Readback register on the other hand enables the DDC to perform a readback operation on the header bytes present in the FIFO.

TABLE 4.1. The DDC Internal Registers In Numerical Order

Hex Address	Name
00	Status Register
01	Error Register
02	ECC Shift Register Out0/Polynomial Preset Byte0
03	ECC Shift Register Out1/Polynomial Preset Byte1
04	ECC Shift Register Out2/Polynomial Preset Byte2
05	ECC Shift Register Out3/Polynomial Preset Byte3
06	ECC Shift Register Out4/Polynomial Preset Byte4
07	ECC Shift Register Out5/Polynomial Preset Byte5
08	Polynomial Tap Byte0
09	Polynomial Tap Byte1
0A	Polynomial Tap Byte2
0B	Polynomial Tap Byte3
0C	Polynomial Tap Byte4
0D	Polynomial Tap Byte5
0E	ECC Control
0F	Header Byte Count/Interlock
10	Drive Command Register
11	Operation Command Register
12	Start Sector Number
13	Number of Sector Operations
14	Header Byte0 Pattern
15	Header Byte1 Pattern
16	Header Byte2 Pattern
17	Header Byte3 Pattern
18	Header Byte4 Pattern
19	Header Byte5 Pattern
1A	Local Data Byte Count L
1B	Remote Data Byte Count H
1C	DMA Address Byte0
1D	DMA Address Byte1
1E	DMA Address Byte2
1F	DMA Address Byte3
20	Data Postamble Byte Count
21	ID Preamble Byte Count
22	ID Address Mark Byte Count
23	ID Synch Byte Count
24	Header Byte0 Control Register
25	Header Byte1 Control Register
26	Header Byte2 Control Register
27	Header Byte3 Control Register
28	Header Byte4 Control Register
29	Header Byte5 Control Register
2A	Data External ECC Byte Count
2B	ID External ECC Byte Count
2C	ID Postamble Byte Count
2D	Data Preamble Byte Count
2E	Data Address Mark Byte Count
2F	Data Synch Byte Count
30	Data Postamble Pattern
31	ID Preamble Pattern
32	ID Address Mark Pattern
33	ID Synch Pattern
34	Gap Byte Count
35	Disk Format Register
36	Local Transfer Reg/Header Diagnostic Readback
37	Remote Transfer Register
38	Sector Byte Count L
39	Sector Byte Count H
3A	Gap Pattern
3B	Data Format Pattern
3C	ID Postamble Pattern
3D	Data Preamble Pattern
3E	Data Address Mark Pattern
3F	Data Synch Pattern

TABLE 4.2. Summary of Control Registers

Address	Register Name	General Operations
00H	Status Register	Disk Operation, DMA Status (Read)
01H	Error Register	Error Determination of Operation
0FH	Header Byte Count	
10H	Drive Command	Start Disk Operation
11H	Operation Command	Reset, Remote DMA, INTR Operation
12H	Start Sector	Can Contain Sector Number
13H	No. of Sector Operations	Used in Multi-Sector Operation
35H	Disk Format	MFM, Hard Sector, External ECC
36H	Header Diagnostic	

THE ECC/CRC REGISTERS

The ECC/CRC registers (addresses 02H to 0EH) are used to set up the DDC for desired error detection and correction configuration. Registers 02H to 07H are read-write and contain the preset pattern for the internal ECC. The preset pattern is the data that the ECC shift register is initialized to prior to an operation. Typically this is all ones. During a correcton cycle, reading these registers provides the syndrome bytes to correct the erroneous data. Registers 08H to 0DH are write-only and are used to load in the internal ECC polynomial required for the drive format selected. The ECC Control register (address 0EH) is used for selecting the internal ECC Correction Span, inversion of input or output check bits to the ECC register and ECC encapsulation (the mode that includes sync bytes (address marks) in the check bit calculation). These registers are listed in Table 4.3.

THE FORMAT REGISTERS

The Format registers, Table 4.4, (addresses 20H to 2FH, 30H to 35H and 38H to 3FH) determine and control the format of the fixed fields for the selected drive type according to the selected format. (*Figure 4.3* shows the Sector Format fields incorporated in the DDC). Registers 20H to 2FH contain the byte count of the fixed fields along with the 6 Header Control registers while Registers 30H to 35H and 38H to 3FH contain the patterns of the fixed fields along with the Inter Sector Gap Count and the Sector Byte Count registers. These Registers are shown in Table 4.4. Since almost every pattern register has an associated count register (which controls the repetition number of its field) or a control register, Table 4.4 is organized to show both together.

THE DMA REGISTERS

The DMA registers consist of the Local Transfer register (address 36H), Remote Transfer register (address 37H), and count and address registers, 1AH to 1FH. The Local Transfer register controls the data transfers between the DDC and buffer memory by controlling data and address bus lengths, byte ordering, memory cycle and the burst length. The Remote Transfer register, on the other hand, controls the data transfers between the buffer memory and the system I/O port in dual bus architectures. In addition to the data and address bus lengths, the memory cycle length, and determining the burst lengths, it also controls the transfers in the dual channel DMA mode.

Registers 1AH and 1BH determine the byte count required in a remote data transfer while registers 1CH to 1FH are DMA Address bytes 0 to 3 for local and remote transfers.

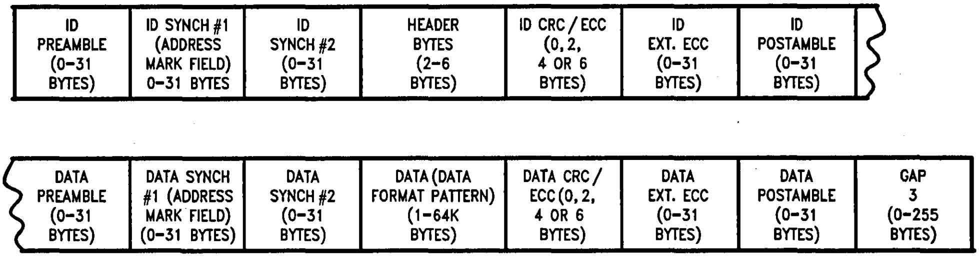
Table 4.5 lists the DMA control and address registers, and their function when the DDC is used in either dual channel or single channel mode.

TABLE 4.3. ECC Control Registers

Shift Reg/ Polynomial Preset	Polynomial Tap	Register Description
02	08	ECC Shift Reg/Poly Preset and Tap 0
03	09	ECC Shift Reg/Poly Preset and Tap 1
04	0A	ECC Shift Reg/Poly Preset and Tap 2
05	0B	ECC Shift Reg/Poly Preset and Tap 3
06	0C	ECC Shift Reg/Poly Preset and Tap 4
07	0D	ECC Shift Reg/Poly Preset and Tap 5
0EH		ECC Control
08H, 09H		Data Byte Count

TABLE 4.4. Format Count, Control and Pattern Registers

Count/ Control Reg	Pattern Reg	Header/ID Field Register Descriptions	Count Reg	Pattern Reg	Data Field/ECC Format Register Description
20	30	Data Postamble	2A	—	Data External ECC
21	31	ID Preamble	2B	—	ID External ECC
22	32	ID Synch Field 1	2C	3C	ID Postamble
23	33	ID Synch Field 2	2D	3D	Data Preamble
			2E	3E	Data Address Mark
24	14	Header Byte 0	2F	3F	Data Synch
25	15	Header Byte 1	34	3A	Post Sector Gap
26	16	Header Byte 2			
27	17	Header Byte 3	—	3B	Data Format
28	18	Header Byte 4	38	—	Sector Byte (LSB)
29	19	Header Byte 5	39	—	Sector Byte (MSB)


ID PREAMBLE (0-31 BYTES)
ID SYNCH #1 (ADDRESS MARK FIELD) 0-31 BYTES
ID SYNCH #2 (0-31 BYTES)
HEADER BYTES (2-6 BYTES)
ID CRC / ECC (0, 2, 4 OR 6 BYTES)
ID EXT. ECC (0-31 BYTES)
ID POSTAMBLE (0-31 BYTES)
DATA PREAMBLE (0-31 BYTES)
DATA SYNCH #1 (ADDRESS MARK FIELD) (0-31 BYTES)
DATA SYNCH #2 (0-31 BYTES)
DATA (DATA FORMAT PATTERN) (1-64K BYTES)
DATA CRC / ECC (0, 2, 4 OR 6 BYTES)
DATA EXT. ECC (0-31 BYTES)
DATA POSTAMBLE (0-31 BYTES)
GAP 3 (0-255 BYTES)

TL/F/8663-70

FIGURE 4.3. Sector Format Fields Incorporated in the DDC

TABLE 4.5. DMA Registers and Functions in Dual or Single Channel Mode

Addrs	Register Name	Single Channel	Dual Channel
1AH	Remote Count	NA	LSB Data Xfer Count
1BH	Remote Count	NA	MSB Data Xfer Count
1CH	DMA Addr 0	A0–A7	A0–A7 Local DMA
1DH	DMA Addr 1	A8–A15	A8–A15 Local DMA
1EH	DMA Addr 2	A16–A23	A0–A7 Remote DMA
1FH	DMA Addr 3	A24–A32	A8–A15 Remote DMA
36H	Local Transfer	Configures Local or Single Channel	
37H	Remote Transfer	Configures Remote (Dual Channel Mode)	
37H	DMA Sector Count	DMA Sector Counting (Dual Channel)	

4.3.3 The FIFO

The primary function of the DDC is to transfer data between disk and the system. The DDC has been configured so that during a disk data transfer operation, it does not occupy the bus for the whole disk transfer. Instead, it allows burst transfers so that the bus is free between the bursts for normal system usage. Systems with a main microprocessor and main memory will interface directly to the DDC. In this type of application, burst data transfers will require occupancy of the main system bus, so use of the bus must be granted at the discretion of the system. For example, if the system is performing a higher priority operation, it must not relinquish the bus for disk data transfer.

Once the bus has been relinquished, the system is held from performing other operations and a burst of data is then transferred. For the DDC, these requirements mean first, that some degree of data buffering is necessary to store the continuous arrival or removal of disk data, and second, that when the bus is granted, transfer must be fast. The amount of data buffering will be dependent on the system, but the majority of low-end systems should be able to respond to a data transfer request from the DDC within 50 μs. Most disk drives in this kind of application run at a data rate of 5 Mbits/sec, or one bit every 200 ns. Typically then, the data buffer must be able to store around 250 bits. A 32-byte FIFO has been included on the DDC enabling it to operate with most bus systems.

The data is transferred between the FIFO and the local memory (dual channel mode) or system memory in different burst thresholds, 1, 4, 8 or 12 words for 16-bit wide word transfers or 2, 8, 16 or 24 bytes for byte wide transfers. In a disk read operation when the FIFO fills to the selected threshold level with disk data, the DMA controller issues a data transfer request. The FIFO continues to fill. Whenever the DMA gets access to the system bus, it transfers data in selected bursts. These bursts may be of fixed length or until the FIFO empties. If the DMA request is not acknowledged, and the disk data fills the FIFO before it reaches to its maximum 32-bytes capacity, the FIFO Data Lost error occurs and the operation is aborted.

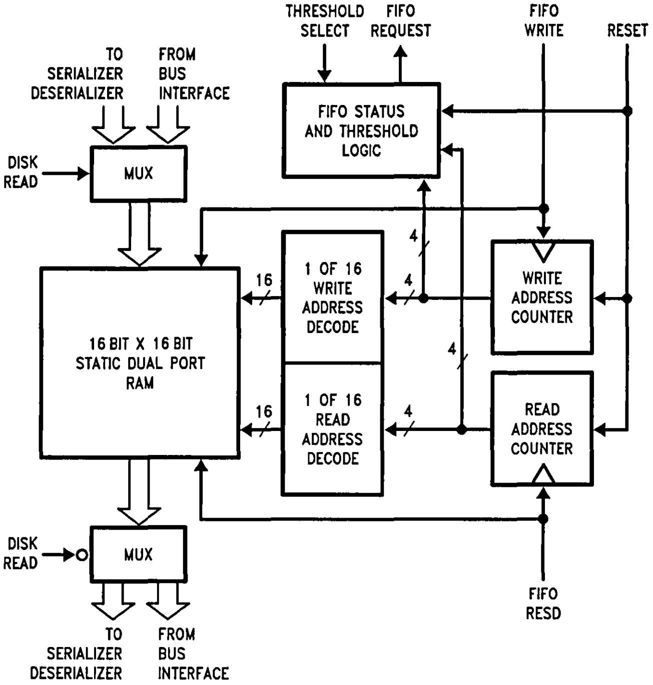
Conversely, in a disk write operation, the FIFO is first filled. It then requests a new data burst when it empties to below the selected threshold level. Depending on burst mode, the DMA will then request a fixed number of bytes or fill the FIFO. If the FIFO completely empties during the operation, a FIFO Data Lost error is again generated.

Figure 4.4 shows the basic blocks that compose the FIFO. It consists of a 16 x 16 bit dual port RAM array, which is addressed by a read counter and a write counter. These counters are decoded to address the array and also to feed the status logic which takes the difference between these pointers to generate the threshold signals. The input and output data ports of the FIFO can be connected to the serializer/deserializer or bus interface block. The direction of the FIFO is determined by the disk operation, read or write. An 8-bit bus interface is supported in 8-bit transfer mode by treating the FIFO as two interleaved banks of 8 bits.

4.3.4 The DMA

The DDC has an important feature that helps both in saving external ICs and in increasing data throughput, namely powerful DMA capability. With on-chip DMA capability, there is no need to dedicate a channel of a DMA controller for disk transfers. This offers two advantages: first, it may alleviate the need for a DMA controller chip, and second, memory transfer time will be faster because DMA controllers are relatively slow, usually around 2 Mbytes/sec maximum throughput. The DDC can transfer data much faster than this, especially when selected to transfer 16 bits each cycle. This faster transfer rate offers a much lower bus occupancy time, freeing the bus sooner for other usage.

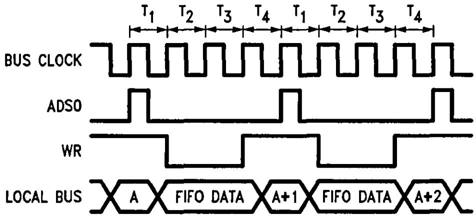
When using the DMA capability, the DDC becomes bus master during the data transfer operation. In bus master mode the DDC issues incrementing address information at the start of each memory cycle. Each read or write memory cycle takes four clock periods, using a similar sequence to a four clock cycle microprocessor with multiplexed address and data bus. In some cases a five clock cycle sequence is used when two address words must be multiplexed to form the DMA address. *Figure 4.5* shows a typical read or write cycle of 4.



THRESHOLD SELECT
FIFO REQUEST
FIFO WRITE
RESET
TO SERIALIZER DESERIALIZER
FROM BUS INTERFACE
FIFO STATUS AND THRESHOLD LOGIC
DISK READ
MUX
4
16
1 OF 16 WRITE ADDRESS DECODE
4
WRITE ADDRESS COUNTER
16 BIT X 16 BIT STATIC DUAL PORT RAM
4
16
1 OF 16 READ ADDRESS DECODE
4
READ ADDRESS COUNTER
DISK READ
MUX
FIFO RESD
TO SERIALIZER DESERIALIZER
FROM BUS INTERFACE

TL/F/8663-71

FIGURE 4.4. Simplified Block Diagram of DDC's FIFO



TL/F/8663-72

FIGURE 4.5. DDC-to-Memory Word Transfers (16-Bit Address)

The block diagram on the DMA section is shown in *Figure 4.6*. The heart of the DMA is a sequencer/PLA that uses inputs from the FIFO (FIFO Request) and disk control logic (DC Ready) for local DMA. It also generates signals for the remote channel to determine if a transfer is necessary. It then issues the requests, and once the acknowledge is received, the PLA sequences through the DMA cycle by placing the address onto the data/address bus (via the bus interface block) and manipulating ADS0, ADS1, read, and write strobes. The various counters are then incremented. In addition to the address counters, there are counters to keep track of bytes per sector (local and remote channel), bytes per header (local and remote), number of total sector operations, and remote counter burst length. The DMA sector counter is used in the tracking mode (described later), and enables the destination DMA whenever it is not zero. The remote transfer counter is decremented after each remote transfer and is used to set the total length of the transfer. When it reaches zero the sequencer halts remote DMA operations.

The DDC can be configured into two DMA modes, Single Channel and Dual Channel. The Dual Channel Mode has two sub-modes: Tracking and Non-Tracking. The general operation of these modes is described below.

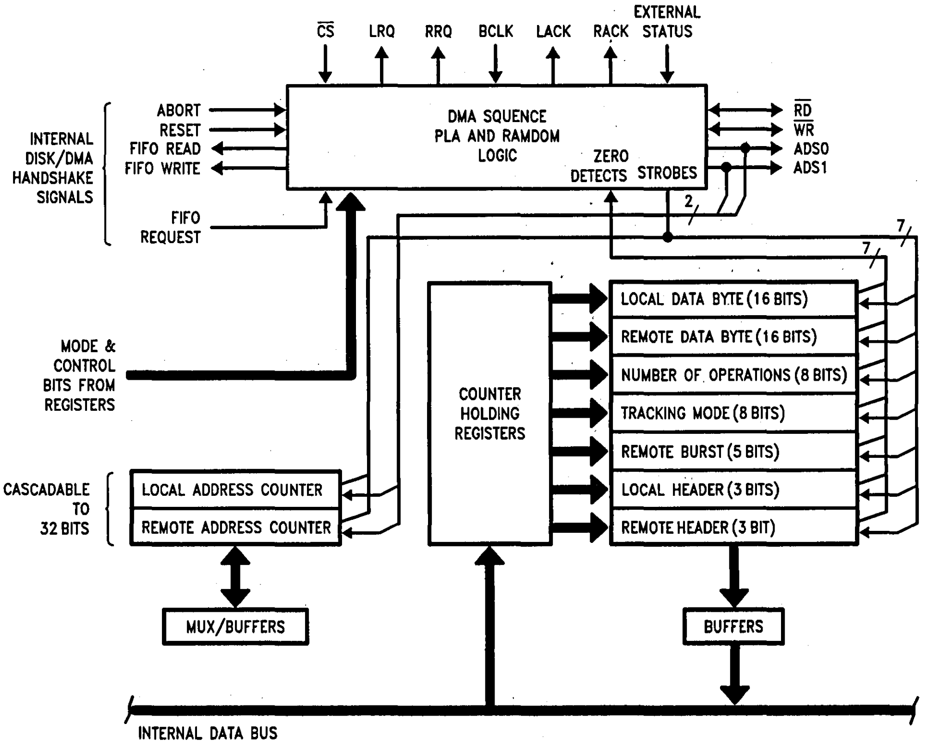
SINGLE CHANNEL

In single channel DMA mode the DDC interfaces directly to main memory having 32 address bits available to access up to 4 Gigabytes of memory. *Figure 4.7(a)* shows the DDC in single channel DMA configuration. The lower 16 address bits are normally issued at the start of each memory cycle so that most memory cycles comprise four clocks. The upper 16 bits are issued at the start of an operation or if the lower 16 bits rollover. In these cases, the memory cycle becomes 5 clock periods, the upper 16 bits are issued during first clock period and the cycle then completes the next four clock periods as in normal read or write operation. The upper two bytes of address information should be latched during the first clock period.

DUAL CHANNEL

Some systems may require the DDC to interface to a local bus with its own dedicated buffer memory before it interfaces to the main system. Such an application would be in intelligent disk drives that comply with the SCSI (SASI) or IPI interfaces. Intelligent drives may receive or transmit data whenever the controlling unit is ready. Another application could be in higher end systems, where the main memory is hooked onto a main bus such as the Q-BUS™, MULTIBUS®, VME or Future Bus. These buses are usually very busy and often impose high latency times while the main processor is performing important tasks. Once the bus is free, it is advantageous to be able to transfer all the disk information in as short a time as possible to minimize bus occupancy. For these types of applications, the dual channel capability of the DDC is ideal.

In the dual channel mode, the DMA generates a 16-bit address for both the local and remote transfers. The local channel controls the data transfers between the FIFO and the local buffer memory. The remote channel, on the other hand, controls data transfers between the addressed local buffer memory and the main system bus through an I/O port. A DMA channel in the system DMA controller could



TL/F/8663-73

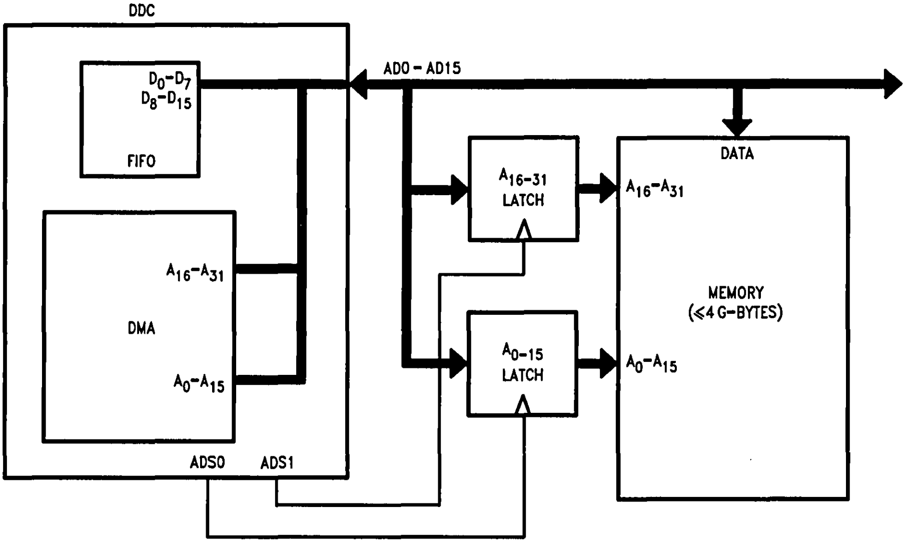
FIGURE 4.6. Block Diagram for Dual/Single Channel DMA Controller

then transfer the data from the port to system memory. Refer to *Figure 4.7(b)*. The local request and remote request are issued when the DDC requires a transfer, the microprocessor and bus arbiter respond by acknowledging the requests. If both channels are requesting, the system should arbitrate the acknowledge, however the local DMA has a higher priority if both requests are acknowledged.

The dual channel mode can be further divided into two modes, Tracking (non-overlapping) and Non-Tracking (overlapping). These dual channel modes are described below.

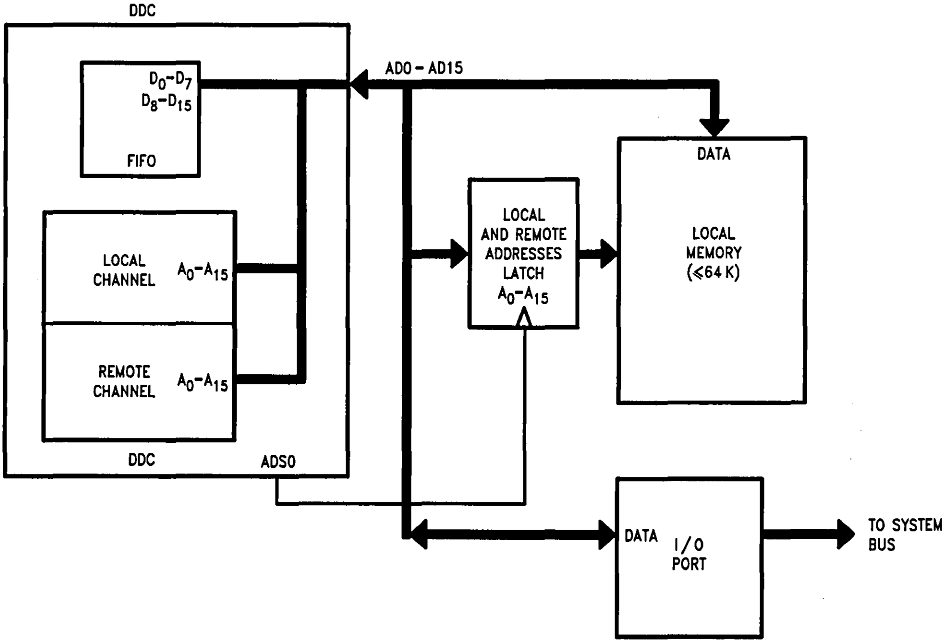
TRACKING OR NON-OVERLAPPING MODE

In Tracking or Non-Overlapping Dual DMA mode the DMA controls the local and remote transfers in such a way that the local buffer memory appears to the system as a large



TL/F/8663-74

(a). The DDC In Single Channel DMA Mode



TL/F/8663-75

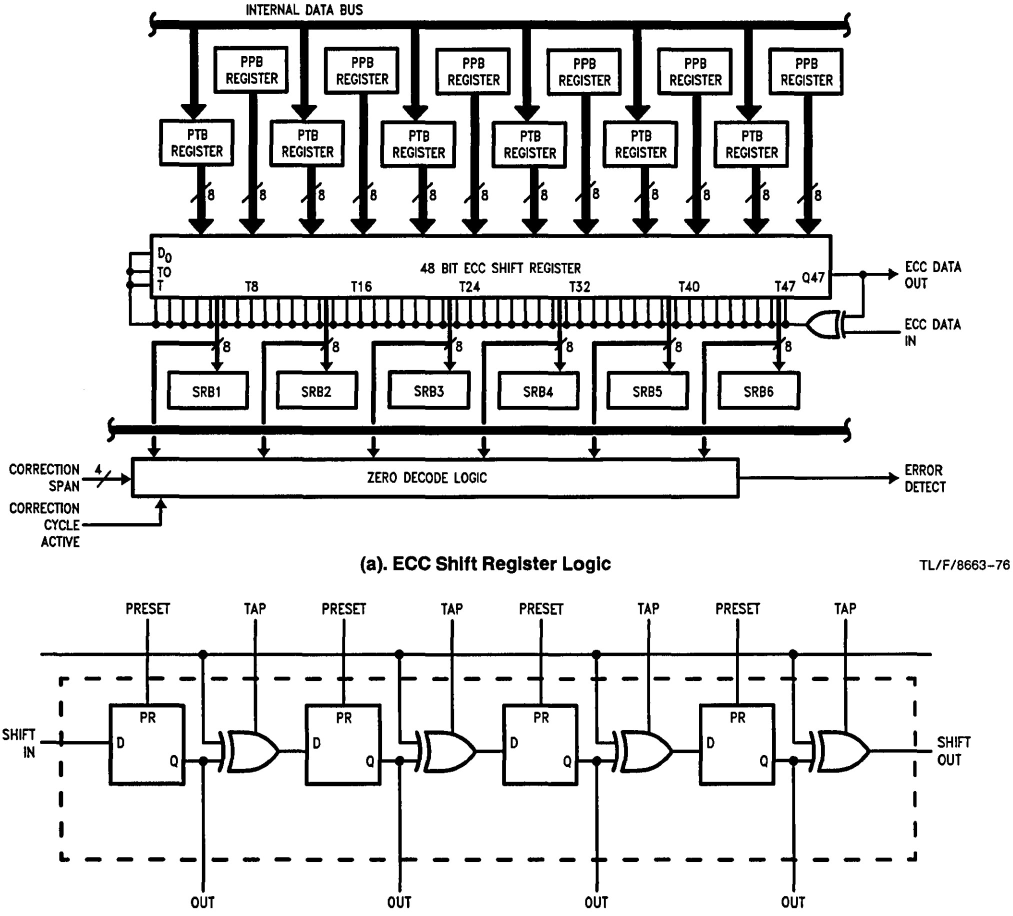
(b). The DDC In Dual Channel DMA Mode
FIGURE 4.7

FIFO. This allows the system to transfer data to or from the local buffer memory whenever the system is ready, but with protection against overlapping of disk data. Basically this mode of operation is more applicable to multi-sector operations where the DDC efficiently interleaves bursts of data into and out of the buffer memory using both channels. Each channel has 16 address bits, allowing up to 65k of buffer memory to be used by the two channels. While doing a remote transfer in a disk read operation, data is read (bursted out) from the same local memory area where it was written to (bursted in) during the local transfer. Similarly, in a disk write operation, data is read from the same local memory area where it was written to during the remote transfer. In both cases, buffer memory addresses for local and remote transfers are issued such that data is never overlapped. If the two channels track very closely, then large amounts of contiguous data can be transferred, making the buffer memory appear to be a multi-megabyte FIFO.

The protection against overlapping of disk data is enabled by use of the DMA Sector Transfer Counter in the DDC. The counter is initially reset at the start of the operation. It is then incremented each time the source has transferred a sector of data into the buffer memory, and is decremented each time the destination has transferred a sector of data from the buffer memory. Whenever the count is zero, destination transfers are inhibited, so preventing the destination from catching up and overtaking the source transfers.

NON-TRACKING OR OVERLAPPING MODE

Some systems require that the two DMA channel are completely independent of each other. For this type of application the DDC can be configured to set up both DMA channels independently. The Local and Remote operations may be from different areas of memory or common areas. The local DMA may already be performing an operation when the Remote DMA is instructed to begin an operation. Likewise a Remote operation can be in progress when the Local operation is initialized. One operation can be for reading memory and the other for writing. This puts the burden on the user to protect from overwriting the buffer memory. In other words, the controlling microprocessor has the responsibility of ensuring that no memory overwriting occurs when both local and remote transfers are in progress. This mode



(a). ECC Shift Register Logic

TL/F/8663-76

TL/F/8663-77

(b). Detail of 4-Bit Section of Shift Register
FIGURE 4.8

PPB = Polynomial Preset Byte
PTB = Polynomial Tap Byte
SRB = Shift Register Buffer

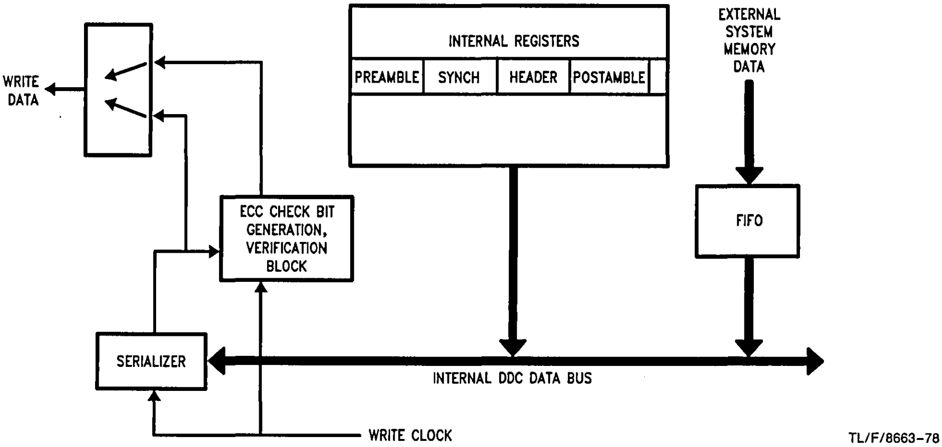
also gives the user the freedom to use the remote DMA controller with no restrictions, even for general non-disk transfers, such as for high level commands or status tranfers.

4.3.5 Error Detection and Correction

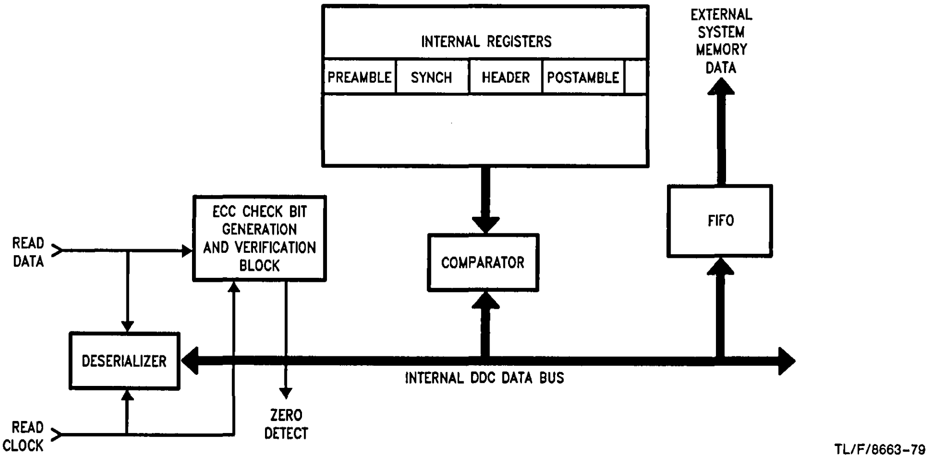
A fixed CRC code for detecting errors only, and a programmable ECC code for detecting and correcting errors can be used via the on-chip Error Detection and Correction block. The DDC has full polynomial programmability for 32 bits or 48 bits of ECC appendage, along with a programmable correction span from 3 to 15 bits and a programmable preset. The DDC can also be easily interfaced to external ECC circuitry if desired.

The Error Detection and Correction block mainly consists of a 48-bit shift register with XOR taps. It generates and then appends the check bits to header and data fields. The CRC uses the standard CCITT polynomial that provides 16 generated check bits. The CRC-CCITT code is hardware implemented in the DDC. The ECC code may be a Fire code or a computer generated code with either 32 or 48 generated check bits. The selected 32- or 48-bit ECC polynomial can be implemented by means of the 48-bit shift register and the Polynomial Tap and Preset Byte registers (addresses 02H thru 0DH).

The ECC Shift register logic is shown in *Figure 4.8*. In this figure the internal data bus connects to the Polynomial Preset Byte Registers (PPB). These register bits feed into the shift register latches. Any bit set in the PPBs will preset a corresponding flip-flop before an ECC operation begins, while all others will not be set. The Polynomial Tap Byte Registers feed the XOR gates in the ECC Shift Register. When a PTB bit is reset, the associated XOR gate is enabled for a particular ECC register bit. This effectively creates the ECC polynomial tap. The outputs of each shift register flip-flop bit input to a set of output buffers which drive the internal DDC data bus. This enables reading of the ECC registers. The ECC outputs also go to a combinational logic block that decodes the contents of the ECC shift register and the correction span. If at the end of a detection cycle the ECC shift register contains zero then no error was detected.



(a) Disk Write Operation



(b) Disk Read Operation
FIGURE 4.9

INTERNAL CHECK BIT GENERATION AND CHECKING

When writing to the disk, the CRC/ECC shift register is preset from the Preset ECC registers. At the same time that the DDC is outputting either the Header field or Data field bytes as a serial data stream through the Serializer, it is feeding them serially into the CRC/ECC shift register as shown in *Figure 4.9(a)*. When the last bit of the Header or Data field has been transmitted out of the DDC, the DDC begins shifting out the generated check bits from the CRC/ECC shift register starting with the MSB and ending with the bit 0. After the specified number of check bits have been appended the DDC internally switches to the next field.

When reading from the disk, the shift register is first preset from the Preset ECC registers before the read data operation begins. The incoming Header or Data field is serially fed into the CRC/ECC shift register as shown in *Figure 4.9(b)*. When all the Header or Data field bits and all the generated check bits have entered the CRC/ECC shift register, the status of the bits in the CRC/ECC shift register is checked for the all zeroes condition. If this condition is met, it signifies the field contains no errors. If any of the CRC/ECC shift register bits are high, the field contains an error. The Header and the Data field errors are indicated by the Status and Error registers respectively.

INTERNAL ERROR CORRECTION

The DDC is capable of correcting from 3 to 15 contiguous bits in error for selected 32- or 48-bit polynomials. The value desired is set in the ECC Control Register, in other words, it can correct a span of the selected amount. The DDC can be put in the Correction Mode through the Operation Command register. The CRC/ECC shift register contains a nonzero 32- or 48-bit pattern which is used to determine the location of the bytes in error and the error pattern. The most significant 3 to 15 bits of the 32 or 48 bits are selected as the Syndrome bits, while the rest are checked for a zero detect. During the correction mode, the CRC/ECC shift register is reverse shifted. Also, reverse shifting guarantees that the correction cycle will be completed within the time it takes to read one sector of the disk.

When the reverse shifting of the shift register begins, the Data Byte Counter register begins decrementing from its preloaded value of the number of data and ECC bytes in the sector. Another 3-bit counter is used to keep track of byte boundaries in the serial bit stream of the whole sector. Reverse shifting continues until all zeroes are detected in the (32-C) or (48-C) bits of the CRC/ECC shift register (where C is the correction span selected). When this occurs and the 3-bit counter contains all zeroes, the clock is stopped. At this point the C syndrome bits contain the bit error pattern of the byte indicated by the Data Byte Counter register. If the 3-bit counter count was not zero when the zero-detect occurred, then the CRC/ECC shift register has to undergo further reverse shifts to byte align the right byte in error. If the Data Byte counter register count goes to zero and the zero-detect is not obtained, then the error is non-correctable. If either the zeroes condition is determined or the Data Byte counter decrements to zero, an INTERRUPT is issued to indicate to the microprocessor that the correction cycle has finished.

The results of the correction cycle are indicated by the Status register. In the case of a correctable error, the error must be in either the Data field or the check bits of the ECC field or overlapping both fields. If the error is only in the ECC field then the memory data is correct and no further action is needed to complete the correction. But if the error is in the Data field then it can be corrected by XORing the C syndrome bits in the CRC/ECC shift register with the contents of the relevant memory location determined from the final Data Byte Counter register count.

EXTERNAL ECC

Some users may wish to use an ECC polynomial code with a different number of check bits. Some encoding schemes require a wider error correction span, or some users may prefer some high integrity ECC codes such as Reed-Solomon code. For these reasons DDC has been configured to interface easily to external ECC circuitry.

When the DDC is configured to utilize an external circuit for ECC code, the external ECC code may use any polynomial that generates from 1 to 31 bytes of check bits. The external circuitry is informed by the DDC when data is valid and when to generate check bits (for writing) or detect (when reading) through SDV, EEF and EXT STAT pins. Refer to Section 4.2 for pin description. The external ECC may be used to encapsulate internal CRC/ECC field as a confirmation of error detection.

4.3.6 The Serializer-Deserializer

This section of the DDC interfaces to the disk. The Serializer takes byte wide data either from the internal registers or the FIFO into a shift register and serially outputs the bits in a continuous bit stream, starting with the most significant bit. This serial data is then fed into the Error Detection and Correction block for check bit generation. When the CRC/ECC appendage is about to begin, the serializer stops shifting out and the Error Detection and Correction block begins shifting out the check bits, again most significant bit first. At the end of the appendage the Serializer starts shifting out further information to finish the segment. The serial data passes through the MFM Encoding and Precompensation block, if selected, or is output to the external encoder.

During a read operation, the incoming serial NRZ data feeds into the Deserializer and the CRC/ECC block, most significant bit first. The Comparator continually checks the incoming data for a synchronizing pattern that matches the pattern loaded into the internal pattern registers. Once a match occurs, the DDC then knows the byte boundary such that all further bytes from the deserializer are byte synchronized to the boundary first established. CRC/ECC fields, postamble, and preamble fields are not required to be deserialized, and do not enter the deserializer. Once the address mark and/or synch byte have aligned, the header bytes preloaded into the internal registers are sequentially output to the Comparator as each incoming byte is ready. The Comparator checks all the header bytes in turn for a match. If a full match is detected, the DDC checks the CRC/ECC appendage and prepares for the following data field. Finally, the data field is read, and serial data bytes are converted to parallel. They then enter the FIFO from the Deserializer to be transferred by the DMA.

The basic blocks associated with the Serializer/Deserializer are shown in *Figure 4.10*. For Serialization, data from either the Pattern Registers, Sector Counter, or FIFO is multiplexed to a holding latch. The holding latch will load the Serializer/Deserializer shift register at the appropriate time (determined by the disk controller's PLA). This data is shifted through the EEC/Data MUX and the MFM or NRZ logic to the Serial Data output pin. When serializing data the write clock feeds the shift register.

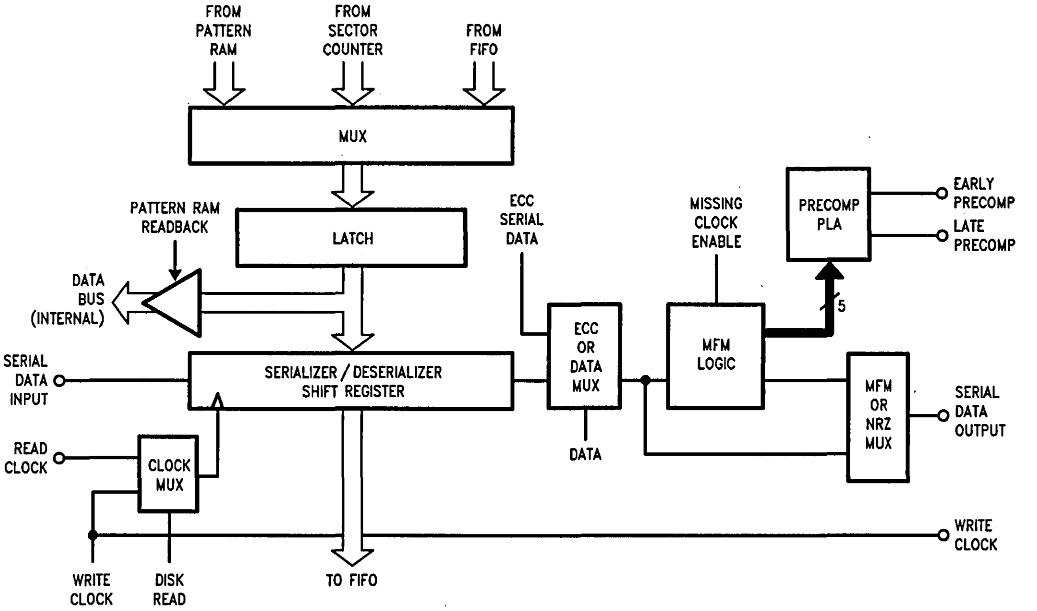
For Deserialization of data, Read Data and Read Clock feeds the internal data bus. Once byte alignment is determined, a byte clock controls loading data into the FIFO.

MFM ENCODING AND PRECOMPENSATION

The DDC can be set to output MFM encoded data and Precompensation information in a disk write operation. The MFM Encoding and Precompensation block of the DDC consists of a 5-bit shift register, and MFM and Precompensation encode logic. The NRZ data coming out of the Serializer passes through the shift register and then is fed into the MFM and Precomp encode logic. The MFM Encode logic converts this NRZ data into MFM and also inserts the missing clocks when Address Mark fields are required to be written to the disk.

The DDC can be programmed to output two control signals, EARLY PRECOMP or LATE PRECOMP, if precompensation is desired. The information on these output pins is then used by the external Precomp Circuitry (MUX, Delay logic, etc.) to perform the actual Precompensation. These pins perform an algorithm that compensates for the bit shifting that occurs when the magnetic flux transitions are recorded on the disk.

The MFM Encoder and precompensation block are shown in *Figure 4.11* also. After serialization, logic performs the MFM encoding. 5 bits from the encoder monitor previous and subsequent data to determine whether early, late, or no precompensation is needed.



TL/F/8663-80

FIGURE 4.10. Serializer/Deserializer Block Diagram

CHAPTER 5 The DDC Registers and Commands

5.0 INTRODUCTION

In this Chapter, the DDC's (DP8466) registers and its command operation will be described in more detail. Initially, a general overview of the registers is given. This section can be used to supplement the data sheet register descriptions. Then details on the various methods of formulating commands and operations are described. Understanding the wide variety of operating modes presented in this section will enable a better understanding of which modes are suitable for which applications.

5.1 INTERNAL REGISTERS AND COUNTERS

In this section a detailed description of internal Command, ECC/CRC, Format and DMA pattern, control and count registers, and various counters is provided. A summary description of the registers is given below, as well as a description of important command bits. Additional details of the DDC's registers are given in the DP8466's Data Sheet.

5.1.1 Command Registers and Counters

The Command registers and counters are listed in *Figure 5.1* and explained in the following paragraphs. These registers include the drive and Operations Command registers, Disk Format register, Error and Status registers, Sector Counter and Sector Operations Counter registers.

Name	Hex Address
Drive Command Register	10H
Operation Command Register	11H
Disk Format Register	35H
Status Register	00H
Error Register	01H
Sector Counter	12H
Number Of Sector Operations	13H
Header Byte Count Register	0FH
Interlock Complete Signal	
Header Diagnostic Readback	36H

FIGURE 5.1. Command Registers and Counters

DRIVE COMMAND (DC) REGISTER (Address = 10H)

The drive command register is shown in *Figure 5.2.* This register is loaded when the DDC is required to perform a command. Disk operations are started after loading this write-only register. This register can be loaded to start a new command when the Next Disk Command bit is set in the Status register. The bit descriptions are given below.

Re-Enable (RED)

When loading a disk command a zero should be written to this bit to permit normal operation. A one should be written to the bit to re-enable the DP8466 after either a reset via the RESET pin or RES bit in the Operation Command Register.

Start Operation (SAIS)

Bit SAIS determines when the operation for the command being written to the drive command register shall begin. If SAIS is high, the operation will begin when the DDC detects either an INDEX or SECTOR PULSE for hard sectored drives, or immediately for soft sectored drives. If SAIS is low the operation only begins when the DDC detects an INDEX PULSE.

Single/Multi Sector Operation (MSO)

Bit MSO of the drive command register indicates whether the operation is for just one sector or a number of sectors. If MSO is set high then the DDC can perform a multi-sector operation. Multi-sector operations usually are handled on logically (by sector address) contiguous rather than physically contiguous sectors. The DDC can perform both types of multi-sector operations.

D0	RED	RE-ENABLE	0 = No Action 1 = Re-Enabled
D1	SAIS	START OPERATION	0 = Start on Index Pulse 1 = Start on Index/Sector Pulse or Immediately
D2	MSO	SINGLE/MULTI-SECTOR OPERATION	0 = Single 1 = Multi
D3	FMT	FORMAT MODE	0 = Normal 1 = Format
D4 D5	HO1 HO2	HEADER OPERATIONS	00 = Ignore Header 01 = Compare Header 10 = Write Header 11 = Read Header
D6 D7	DO1 DO2	DATA OPERATIONS	00 = Invalid* 01 = Check Data 10 = Write Data 11 = Read Data

*Unless used with an Ignore Header operation. When HO1, HO2, DO1 and DO2 are written as 0 then no operation is performed. This is useful when Re-Enabling the DDC by setting the RED bit, for example.

FIGURE 5.2. Drive Command Register Bit Assignments

Format Mode (FMT)

The DDC can be set to format a disk by setting bit FMT high. The DO2, DO1, HO2, and HO1 bits must be set for a write-header/write-data operation, 1010. (For details please refer to the disk formatting section).

HEADER OPERATIONS (HO2, HO1)

Bits HO2 and HO1 determine the operation to be performed on the ID fields. The details are given below.

Ignore Header (HO2 = 0, HO1 = 0)

In an Ignore Header operation, after the byte alignment of Address Mark and/or Synch fields, the header bytes comparison and ECC/CRC checks are not performed. This results in reading or writing the associated data with respect to any sector encountered.

Compare Header (HO2 = 1, HO1 = 1)

Compare Header operation is the normal mode of header operation for locating the selected sector. In this operation the header bytes are compared with the corresponding values in the pattern registers and CRC/ECC checks are also carried out.

Write Header (HO2 = 1, HO1 = 0)

Write Header operation is normally performed during disk formatting. The Header bytes are written to the disk either from the pattern register or the buffer memory through FIFO depending upon the FIFO Table Format (FTF) bit of the Disk Format register.

Read Header (HO2 = 1, HO1 = 1)

Read Header operation performs CRC/ECC checks and transfers the header bytes into buffer memory through the FIFO for diagnostic purposes. If during this operation, a header containing a CRC/ECC error is encountered, the operation is aborted immediately and the header fault bit (Status Register) is set. An interrupt is generated, but no Error register bits are set.

DATA OPERATIONS (DO2, DO1)

Bits DO2 and DO1 determine operations to be performed on data fields.

Check Data (DO2 = 0, DO1 = 1)

After the preceding header operation and the byte alignment, the Check Data operation performs the CRC/ECC checks on the data fields. It does not transfer the data field to the FIFO and hence no data is transferred to memory via the DMA channels.

Read Data (DO2 = 1, DO1 = 0)

Read Data operation, on the other hand, transfers data to external memory via FIFO after performing CRC/ECC checks.

Write Data (DO2 = 1, DO1 = 1)

In Write Data operation, after the associated header operation, data bytes are written to the selected sector from the FIFO using the DMA channel.

OPERATION COMMAND (OC) REGISTER
(Address = 11H)

Operation command register, shown in *Figure 5.3,* is a write-only register and may be updated before each disk operation. This register controls some of the basic DDC operating modes, such as interrupts, starting remote DMA, starting a correction, and precompensation.

Reset (RES)

When RES is set high, the DDC enters the standby mode and remains in this mode until this bit is reset and a one is written to the RED bit in the Drive Command register. To properly reset the DDC this bit must remain set for 32 read clock periods and 4 bus clock periods (with both clocks applied). This bit has the same effect as setting the RESET input pin low. Read and Write Gate are deasserted, the FIFO is cleared, DMA requests are removed, the Error register is cleared and status register is cleared except the Next Disk Command is set, and the abort bit is reset. The parameter registers, sector counter and number of sectors registers are not affected.

Enable Interrupts (EI)

Bit EI, when set high, enables the DDC to issue interrupts when certain conditions are met, such as upon successful completion of a command.

Enable Header Complete Interrupts (EHI)

Setting bit EHI high will enable the DDC to issue an interrupt at the completion of a header operation so that it can be loaded with new information before the next ID operation begins. New commands can be loaded or some pattern information could be updated during the data operation of a sector.

D0	RES	RESET	0 = Normal Operation 1 = Reset DDC
D1	EI	ENABLE INTERRUPTS	0 = Disabled 1 = Enabled
D2	EHI	ENABLE HEADER COMPLETE INTERRUPTS	0 = Disabled 1 = Enabled
D3	SRI	START REMOTE INPUT	0 = No Remote Read 1 = Start Remote Read
D4	SRO	START REMOTE OUTPUT	0 = No Remote Write 1 = Start Remote Write
D5	EP	ENABLE PRECOMPENSATION	0 = Disabled 1 = Enabled
D6	SCC	START CORRECTION CYCLE	0 = No Cycle 1 = Start Cycle
D7	IR	INTERLOCK MODE	0 = No Interlock 1 = Interlock Mode

FIGURE 5.3. Operations Command Register

Start Remote Input (SRI)

The DDC initiates the transfer of data from the system to local memory when SRI is set high. This enables the start of remote transfers (non-tracking dual DMA mode) to the local buffer.

Start Remote Output (SRO)

The DDC initiates the transfer of data from local memory to system when bit SRO is set high. This starts remote transfers from the local buffer.

Enable Precompensation (EP)

The DDC will allow precompensation if bit EP of the operation command register is set high. This bit is valid only if the AMF/EARLY PRECOMP and AME/LATE PRECOMP pins are configured as write precompensation control pins.

Start Correction Cycle (SCC)

The internal correction cycle is initiated by setting bit SCC high (see ECC section).

Interlock Mode (IR)

In some situations, if it is desired to update the header bytes (during disk formatting) or issue a new drive command before the next command, the DDC must be put in Interlock mode by setting bit IR high.

In interlock mode, after every header operation, the DDC issues an interrupt after Header Match Complete flag, (HMC), in the status register goes high, indicating that the information can be updated before the beginning of the next header operation (or during the current data operation). Within this time, microprocessor has to update the information (header bytes or drive command register) and then write the header byte count to the Header Byte count register to indicate completion of update.

DISK FORMAT (DF) REGISTER (Address 35H)

Disk Format register, shown in *Figure 5.4,* is a write-only register and is usually updated when a different drive type is selected. This register controls some of the major format features of a disk, such as MFM, type of ECC/CRC, and configuring address marks.

D0	MFM	NRZ/MFM ENCODE	0 = NRZ 1 = MFM
D1	SAM	START WITH ADDRESS MARK	0 = Start with Preamble 1 = Start with AM
D2	HSS	HARD OR SOFT SECTOR	0 = Soft 1 = Hard
D3	FTF	FIFO TABLE FORMAT	0 = Use Registers 1 = Use FIFO
D4 D5	IH1 IH2	INTERNAL HEADER APPENDAGE	00 = None 01 = 16-bit CRC 10 = 32-bit ECC 11 = 48-bit ECC
D6 D7	ID1 ID2	INTERNAL DATA APPENDAGE	00 = None 01 = 16-bit CRC 10 = 32-bit ECC 11 = 48-bit ECC

FIGURE 5.4. Disk Format Register

MFM/NRZ Encode (MFM)

When writing to the disk, the DDC can output either MFM encoded data if MFM is high, or NRZ data if MFM is low.

Start with Address Mark (SAM)

If SAM is low, the format begins with the Preamble field followed by the Address Mark field. If SAM is high, the first field is Address Mark followed by the Preamble field. This supports ESDI or SMD drive formats.

Hard or Soft Sector (HSS)

The DDC can be configured for soft or hard sectored drives by setting bit HSS low or high, respectively.

FIFO Table Format (FTF)

If bit PTF is low, the header bytes are taken from the internal pattern registers during the disk formatting. If FTF is high, these bytes will be written from the FIFO through local DMA channel. This bit is used only during disk formatting.

Internal Header and Data Appendages (IH1, IH2, ID1, ID2)

The Internal Header Appendage (IH1 and IH2) and the Internal Data Appendage (ID1 and ID2) bits of the Disk Format register control CRC/ECC appendages for header and data fields. The appendage options which could be selected are no appendage (00), 16 bit CRC CCITT polynomial (01), and 32- and 48-bit programmable ECC codes (10, 11 respectively). If none of these internal ECC or CRC codes is selected, then an external header ECC code must be appended. Also, even if the internal codes are being appended, an external ECC code of up to 31 bytes may be added to encapsulate header, data and internal CRC/ECC fields.

STATUS (S) REGISTER (Address = 00H)

Figure 5.5 shows the flags in the Status register, which is a read-only register. This register provides status on current operation of the DDC. This includes DMA local and remote status, correction cycle status, operation error, and ready for next command's status. The flags are set or reset by conditions detected by the DDC. The flags are also reset when either the RESET input pin or RES bit in the operation command register are set. The flags in the Status register either provide the status of different operations in progress or the results of these operations.

D0	HF	HEADER FAULT
D1	NDC	NEXT DISK COMMAND
D2	HMC	HEADER MATCH COMPLETE
D3	LRG	LOCAL REQUEST
D4	RCB	REMOTE COMMAND BUSY
D5	LCB	LOCAL COMMAND BUSY
D6	CCA	CORRECTION CYCLE ACTIVE
D7	ED	ERROR DETECTED

FIGURE 5.5. Status Register

Header Fault (HF)

The Header Fault flag (HF) is set when a header field error is detected after a Compare Header operation. This is set when an ECC or CRC error is detected in any header field read. This may or may not be on the header that the DDC was looking for. During a disk operation if a header error was detected, and subsequently the correct sector was found, this bit will be reset. If the correct sector was not found, the DDC will timeout with the HF bit set. It is reset when the DDC is reset or when a new command is issued.

The HF will abort the operation immediately if the operation is a read header, and any header read has a CRC/ECC error. In this case no Error register bit is set.

Next Disk Command (NDC)

The Next Disk Command flag, when set, shows that the DDC is ready to receive a header byte update and another disk command. It is reset when a new disk command is issued to the DDC.

Header Match Completed (HMC)

In a Compare Header operation, after a header match, the Header Match Completed flag is set. This bit is reset at the end of the data operation. This flag is automatically set in Ignore and Write header operations or when any header field is encountered after a Read Header operation.

Local Request (LRQ)

The Local Request flag follows the LRQ exactly. It is set coincident with the LRQ output when the FIFO first requires a data transfer. The flag is reset whenever the LRQ pin is deasserted.

Remote Command Busy (RCB)

The Remote Command Busy flag is set at the start of a remote transfer operation and is reset at the completion of the last memory transfer. This can be used to determine if the remote DMA channel is in operation.

Local Command Busy (LCB)

The Local Command Busy flag remains set through the entire period the local DMA channel is busy in transferring data between the FIFO and buffer memory. This is the same function as the RCB except local DMA.

Correction Cycle Active (CCA)

The Correction Cycle Active flag is set at the beginning of a Correction Cycle (when the Start Correction Cycle bit is set in the Operation Command Register) and is reset at the end of the cycle whether the error is located or not.

Error Detected (ED)

Error Detected flag is set if any of the error flags in the Error register is set. This is the logical ORing of all the Error register bits.

ERROR (E) REGISTER (Address = 01H)

Error register, shown in *Figure 5.6,* is also a read-only register. The flags of this register are set by conditions within the DDC and reset by the next new command to the Drive Command register. The flags are also set low when either the RESET input pin is set low, or the Reset bit (RES) in the Operation register is set.

D0	HFASM	HEADER FAILED ALTHOUGH SECTOR MATCHED
D1	DFE	DATA FIELD ERROR
D2	SNF	SECTOR NOT FOUND
D3	SO	SECTOR OVERRUN
D4	NDS	NO DATA SYNCH
D5	DL	FIFO DATA LOST
D6	CF	CORRECTION FAILED
D7	LI	LATE INTERLOCK

FIGURE 5.6. Error Register

Header Failed Although Sector Matched (HFASM)

The HFASM (Header Failed Although Sector Matched) flag, when set, indicates that the Sector byte(s) of the header field match correctly but there is an error in other header byte(s). This flag can only be set if the Enable HFASM Function (EFH) bit of at least one of the Header Control registers is set high during a Compare Header operation. This bit will be set if any one of the header byte(s) with its EHF bit set matches but any other header bytes don't match. For example, assume a 6 byte header with the first two bytes having their EHF bit set. If during a compare header operation the first byte matched, but any of the 2nd through 6th bytes don't match this HFASM bit is set.

When executing a Compare Header-Check Data command, and this flag is set, the operation is aborted allowing the header bytes to be read from the FIFO for disk diagnostics. If this bit is set during a Compare Header-Read (or Write) Data, the command is aborted, but the header is not stored in the FIFO.

Data Field Error (DFE)

After a successful header match, if an internal CRC/ECC or external ECC error is detected during a Read Data or Check Data operation, the Data Field Error flag will be set.

Sector Not Found Error (SNF)

If the Header Match Completed flag of the Status register is not set for two consecutive index pulses in a Compare Header operation (i.e., the correct header was not found), then the Sector Not Found bit is set to indicate that the desired sector cannot be found. The operation is aborted and an interrupt is issued.

Sector Overrun (SO)

If during the time when data is being transferred between disk and the FIFO, either the SECTOR PULSE or INDEX PULSE inputs go active, then the sector is assumed to have overrun and the Sector Overrun flag (SO) is set. Operation is aborted. RGATE or WGATE are deactivated, and an interrupt is generated.

No Data Synch (NDS)

If an INDEX PULSE (hard or soft sectored drives) or a SECTOR PULSE (hard sectored only) is encountered while the DDC is looking to byte align on the first data synch byte (synch 1 or 2), this bit is set. Also if the DDC recognizes the first synch byte but not subsequent synch bytes then this bit is also set.

FIFO Data Lost (FDL)

This bit is set if the FIFO overflows during a Read Data operation. This normally would occur when the host does not allow the DMA to empty the FIFO faster than the Disk Data is being read. FIFO Data Lost is also set during a Write Data operation when the DDC empties the FIFO writing to the disk, and attempts to read the empty FIFO again. In either case the operation will be aborted.

Correction Failed (CF)

If by the end of a Correction Cycle (Data Byte Counter decrements to zero) the error has not been located, then the error is not correctable and the Correction Failed flag is set.

Late Interlock (LI)

If the Interlock Complete register is not written to by the time the next header field arrives, and the DDC is in Interlock mode, then the Late Interlock flag will be set.

START SECTOR (SC) REGISTER (Address = 12H)

The Start sector (and Number of Sector Operations Counter) facilitates multi-sector operations. This counter can be programmed to replace the header byte designated by the user to be the sector number. Thus, in a multi-sector operation, the Sector Counter is initialized with the sector number to start on. As a header is compared, and its data field is read or written, the sector counter is incremented at the end of that sector's header operation. This enables immediate operation on the next logical sector. Operation continues until the Number of Sector Operations Counter decrements to zero. The sector counter is enabled if bit substitute sector counter bit of any Header Control register is set high, and the contents of Sector Counter will be substituted for the corresponding Header Byte.

NUMBER OF SECTOR OPERATIONS COUNTER (NSO) REGISTER (Address = 13H)

In a multi-sector operation, the Sector Operations Counter is preset to the logical number of sectors to be consecutively operated on. It is decremented after every sector's header operation and when decremented to zero, terminates the active command.

HEADER BYTE COUNT (HBC)/INTERLOCK REGISTER (Address = 0FH)

This 4-bit read-write register, normally used during formatting, is loaded with the number of header bytes to be written to (or read from) disk. The allowable number of header bytes is from 2 to 6. On read-back, only the three least significant bits of this register are valid.

Another important function of this register is when the DDC is in the Interlock mode (explained in formatting section). During a multi-sector operation, if it is desired to update any header byte (for example in the case of disk formatting) or if the next drive command has to be changed, then this register must be written with the actual header byte count value after updating the header bytes. This will basically strobe the internal hardware to recognize that interlock (update) has occurred.

FIFO HEADER DIAGNOSTIC READBACK (HDR) REGISTER (Address = 36H)

This is a read-only register and allows the FIFO contents to be read one byte at a time. Normally, data or header bytes may be read for diagnostic purposes through this register (described later). There is no way to write to the FIFO except under DMA control. In order to read the header bytes in the same order as they are read from the disk, the Reverse Byte Ordering bit in the local transfer register must be reset.

5.1.2 Error Correction/Cyclic Redundancy (ECC/CRC) Registers and Counters

The ECC/CRC registers and counters are listed in *Figure 5.7.* These registers enable programming of various modes of ECC, the ECC pattern, and access to the ECC shift register for performing correction cycles. They are explained in the following paragraphs.

Name	Hex Address
ECC Shift Register Out0 Register	02H
ECC Shift Register Out1 Register	03H
ECC Shift Register Out2 Register	04H
ECC Shift Register Out3 Register	05H
ECC Shift Register Out4 Register	06H
ECC Shift Register Out5 Register	07H
Polynomial Preset (Byte0) Register	02H
Polynomial Preset (Byte1) Register	03H
Polynomial Preset (Byte2) Register	04H
Polynomial Preset (Byte3) Register	05H
Polynomial Preset (Byte4) Register	06H
Polynomial Preset (Byte5) Register	07H
Polynomial Tap (Byte0) Register	08H
Polynomial Tap (Byte1) Register	09H
Polynomial Tap (Byte2) Register	0AH
Polynomial Tap (Byte3) Register	0BH
Polynomial Tap (Byte4) Register	0CH
Polynomial Tap (Byte5) Register	0DH
ECC Control Register	0EH
Data Byte Count (LS) Register	08H
Data Byte Count (MS) Register	09H

FIGURE 5.7 ECC/CRC Registers and Counters

ECC SHIFT REGISTER OUT0-OUT5 REGISTERS (Address = 02-07H)

The 48-bit long CRC/ECC shift register of the DDC can be read through these 6 read-only registers at any time. If 32 byte ECC is used then only registers 0, 1, 4, and 5 are used. After a correction cycle has occurred, these registers contain the ECC syndrome bits. The memory address of the sector in error and the data bit in error are calculated by the μP, and then the bits of these registers are XORed with the data in order to correct the error (assuming the error is correctable).

POLYNOMIAL PRESET (PPB) 0-5 REGISTERS (Address = 02-07)

The selected ECC polynomial preset pattern is loaded into the ECC/CRC shift register from these six Polynomial Preset registers. These are write only registers. The preset bit pattern could be all ones, all zeroes or a combination. This is the value the ECC shift regsiter is loaded with prior to shifting in the ECC pattern. The most significant bit of PPB5 is the most significant polynomial bit, X^{47}, and the least significant shift register tap is the least significant bit of PTB0. For 32-bit ECC, PPB2 and PPB3 are set to all zeroes, and are not used.

POLYNOMIAL TAP BYTE (PTB) 0-5 REGISTERS (Address = 08-0DH)

The ECC shift register is tapped at every bit by an XOR element. Wherever an exclusive-OR tap is required into the 32-bit or 48-bit shift register a zero should be set in the corresponding PTB0-5 bits. All polynomial elements not in the equation (and hence not tapped) must be disabled by setting all these bits to a one. The tap X^{32} (or X^{48} for 48-bit polynomial) is always present and is not programmable. The taps X^{31} (or X^{47}) to X^{0} are fully programmable. The MSB of PTB5 corresponds to the most significant tap, X^{47}, and the LSB of PTB0 is the least significant tap, X^{0}. For 32-bit ECC, PTB2 and PTB3 are not used and must be set to all ones. In this case PTB5's MSB becomes X^{31}.

ECC CONTROL (EC) REGISTER (Address = 0EH)

The ECC Control register, shown in *Figure 5.8,* is a write-only register. This register works in conjunction with the Disk Format register to set the ECC modes. The bit description is given below.

D0	CS0	CORRECTION SPAN SELECT	
D1	CS1	0011	3 bit span
D2	CS2	thru	thru
D3	CS3	1111	15 bit span
D4	HE	ENCAPSULATION HEADER	0 = Encapsulated 1 = Not Encapsulated
D5	IEO	INVERT ECC OUT	0 = Normal 1 = Inverted
D6	IDI	INVERT DATA IN	0 = Normal 1 = Inverted
D7	DEN	DATA ENCAPSULATION	0 = Encapsulated 1 = Not Encapsulated

FIGURE 5.8. ECC Control Register

Correction Span Select Bits (CS0-CS3)

The number of bits which the ECC circuit attempts to correct, generally known as the correction span, is determined by CS0-CS3. Errors longer than the correction span will be treated as non correctable errors. The allowable correction span for 32-bit ECC is 3 to 15 bits and for 48-bit ECC it is 3–15 bits. Setting the CS bits to any correction span that is outside the maximum allowable range of 3–15 bits causes the CRC/ECC to default to a 3 bit correction span.

Header Encapsulation (HEN)

When this bit is reset, the bit patterns of the Synch and/or Address Mark fields are included in ECC/CRC calculations. Some disk formats want these bytes included in check bit calculations (i.e., IBM 3740 floppy format). When this bit is set the Synch and/or Address Mark fields are excluded from the CRC/ECC calculation, and only the header field bytes are included.

Invert ECC Data Out (IEO)

When the shift register data out bit is set high, all the data and check bits coming out of the ECC shift register are inverted in a disk write operation. Otherwise the ECC data is not inverted.

Invert Data In (IDI)

This bit controls data and check bits when they enter the ECC shift register during a disk read operation. When this bit is set high, both data and check bits will be inverted. When low true data is input.

Data Encapsulation (DEN)

The DEN bit performs the same function for the data field as the HEN bit does for the Header field. When reset DEN will include the Synch and Address Mark fields in the CRC or ECC calculations. If set these fields are not included in the check bit calculations.

DATA BYTE COUNTER REGISTERS
(Address = 08H, 09H)

The Data Byte Counter registers are used during a correction cycle, and are preset by the µP prior to starting a correction cycle. They are set to the sum of the number of bytes in the data and ECC fields of the sector just read. During the correction cycle, the data byte count is decremented after shifting by 8 bits in the ECC shift register each byte. At the completion of the correction cycle, and if the error is correctable, the contents of the data byte counters are added to the starting address of the sector in error to determine the location of the memory byte or bytes in error. Details on this, are provided later.

5.1.3 Format Pattern and Count Registers

The Pattern, Count and Control registers used during disk formatting are listed in *Figure 5.9* and explained in the following paragraphs.

PATTERN REGISTERS
(Address = 30–33H, 3A–3FH, 14–19H)

The pattern registers hold byte information for the various fields of a formatted disk. These registers are written to the disk during a format operation. The Synch or Address Mark, Header, and ID postable pattern are read and compared to the pattern registers during a Compare Header operation. The Data Address Mark, and Data Synch are compared when doing a data field read, and are written to the disk during a disk data write or format. Associated with each pattern register (except the header pattern registers) is a byte repetition counter register that sets the field length, described below.

All the pattern registers listed in *Figure 5.9* are preloaded with the value of their respective fields such as ID and Data fields. The fields which are allowed in the DDC pattern registers are ID and Data Preamble, Address Mark, Synch, Postamble and Gap. Up to six header byte patterns can be programmed, thus enabling a header field of six bytes (excluding synch and preamble).

Name	Hex Address
ID Preamble Register	31H
ID Preamble Byte Count Register	21H
ID Synch #1 (AM) Pattern Register	32H
ID Synch #1 (AM) Byte Count Register	22H
ID Synch #2 Pattern Register	33H
ID Synch #2 Pattern Register	23H
Header (Byte0) Pattern Register	14H
Header (Byte0) Control Register	24H
Header (Byte1) Pattern Register	15H
Header (Byte1) Control Register	25H
Header (Byte2) Pattern Register	16H
Header (Byte2) Control Register	26H
Header (Byte3) Pattern Register	17H
Header (Byte3) Control Register	27H
Header (Byte4) Pattern Register	18H
Header (Byte4) Control Register	28H
Header (Byte5) Pattern Register	19H
Header (Byte5) Control Register	29H
ID External ECC Byte Count Register	2BH
ID Postamble Pattern Register	3CH
ID Postamble Byte Count Register	2CH
Data Preamble Pattern Register	3DH
Data Preamble Byte Count Register	2DH
Data Address Mark Pattern Register	3EH
Data Address Mark Byte Count Register	2EH
Data Synch Pattern Register	3FH
Data Synch Byte Count Register	2FH
Data Format Pattern Register	3BH
Sector Byte Count (L) Register	38H
Sector Byte Count (H) Register	39H
Data External ECC Byte Count Register	2AH
Data Postamble Pattern Register	30H
Data Postamble Byte Count Register	20H
Gap Pattern Register	3AH
Gap Byte Count Register	34H

FIGURE 5.9. Format Registers and Counters

During an operation these registers must not be read, as this will interfere with the DDC's internal access to these registers. This could cause internal PLA's to misinterpret these registers, and lead to sporadic misbehavior of the DDC. These registers may be written to any time. If written to during an operation they will take effect immediately.

One data byte pattern register is provided. This pattern is used during a format operation as the data field byte. It is repeated for the length of the data field in the sector.

BYTE COUNT REGISTERS
(Address = 20–23H, 2A–2FH)

The Byte Count registers determine the number of times each field's pattern can be repeated. All of ID and Data Preamble, Address Mark, Synch, Postamble, External ECC, and Gap patterns can be repeated for maximum 31 times. The Gap pattern, on the other hand, can be repeated for 255 times. As mentioned earlier, there can only be six Header bytes and 64K data bytes in any format for the selected drive. The length of Header and Data bytes is controlled by the Header Control and Sector Byte Count registers, respectively (described below).

During an operation these registers must not be read, as this will interfere with the DDC's internal access to these registers. This could cause internal PLA's to misinterpret these registers, and lead to sporadic misbehavior of the DDC. These registers may be written to any time. If written to during an operation they will take effect immediately.

HEADER BYTE CONTROL REGISTER
(Address = 24–29H)

These six read/write registers control the associated six header bytes. Each of the six registers is 4-bits long and performs the same functions. One of these is shown in *Figure 5.10* and the functional description of each bit is given in the following.

D0	HBA	HEADER BYTE ACTIVE	0 = Header Byte Disabled 1 = Header Byte Enabled
D1	SSC	SUBSTITUTE SECTOR NUMBER	0 = Header Byte Used 1 = Sector Counter Used
D2	EHF	ENABLE HFASM FUNCTION	0 = Header Used Normally 1 = Header Interpreted as Sector Number
D3	NCP	NOT COMPARE	0 = Header Used for Compare 1 = Header Disabled

FIGURE 5.10. Header Byte Control Register (One of Six)

Header Byte Active (HBA)

This bit determines whether the corresponding Header byte is to be included in the Header field or not. If set low, the corresponding header byte will be omitted from the header field and setting it high will include the corresponding byte in the header field. Only 4 out of 6 header bytes can be disabled. Also, only two consecutive header bytes can be disabled.

Note: All the other bits in this register must also be set to zero if the header byte is to be disabled.

Substitute Sector Counter (SSC)

This bit when set high, enables the DDC to substitute the Sector Counter register's contents in the header byte pattern register instead of the actual header byte during a write operation, or to compare the Sector Counter register's contents with the corresonding header byte during a read operation. This is normally done in a multi-sector operation to enable automatically incrementing the sector number.

Enable Header Failed Although Sector Matched Function (EHF)

If bit EHF of any header control register is set high, then the associated header byte is designated as that byte that must match in order to enable generation of an HFASM. In this mode, if this header byte matches but any of the other header bytes don't, then an HFASM error and an interrupt is generated. In a Compare Header-Data operation, the header bytes are loaded into the FIFO and can be examined by the host by reading the FIFO Diagnostic Register (Address = 36H). This can also be used during a Compare Header-Read Data, but the FIFO will not store the header bytes, see Error Register description, HFASM.

When this bit is reset the corresponding header byte is compared normally.

Not Compare (NCP)

When this bit is set low, the Header byte will be written and compared normally. On the other hand, if this bit is set high, the corresponding Header byte will always be declared matched regardless of the actual comparison results. In other words the comparison is disabled. This can be used to read a group of sectors.

5.1.4 DMA Registers and Counters

The DMA registers and counters enable programming of the DMA start address (in single or dual channel mode), transfer length, and the various modes of operation. The Operation Command register controls actual starting of the Remote DMA operation. The DMA registers are listed in *Figure 5.11* and explained in the following paragraphs.

Name	Hex Address
DMA Sector Counter	37H
Local Transfer Register	36H
Remote Transfer Register	37H
Remote Data Byte Counter (L)	1AH
Remote Data Byte Counter (H)	1BH
DMA Address (Byte 0) Counter	1CH
DMA Address (Byte 1) Counter	1DH
DMA Address (Byte 2) Counter	1EH
DMA Address (Byte 3) Counter	1FH

FIGURE 5.11. The DMA Registers and Counters

DMA SECTOR COUNTER
(Address = 37H)

This read only register is used only when the DDC is configured in the dual channel tracking mode. This counter keeps track of the number of sectors transferred by the remote DMA channel (local RAM to/from system), and the local DMA channel (DDC to/from local RAM). When this register is 0, remote channel transfers are inhibited. This counter ensures that the source channel will not overtake the remote channel. This eliminates the chances of overwriting while transferring data to or from the local memory, or transferring non-data.

LOCAL TRANSFER (LT) REGISTER
(Address = 36H)

This write-only register, shown in *Figure 5.12*, controls the data transfers between the DDC and buffer memory, using the local DMA channel or single channel mode. It is configured at the time of initialization and normally need not be written to again. The Local Transfer register is not affected by reset or abort operations.

Local DMA Enable (SLD)

This bit when set high, enables the local DMA channel, Tracking or Non-Tracking. If this bit is not set high, the on-chip DMA will not transfer data. This bit is used to enable control of starting/stopping a DMA operation. If it is permanently disabled, external DMA circuitry can be used.

Local Word Data Transfer (LWDT)

This bit determines the length of the data word to be transferred between the DDC and buffer memory. When set to 0, single 8-bit bytes are transferred each DMA cycle and the address increments by 1. If this bit is set, 16-bit words are transferred each DMA cycle and the address increments by 2.

Reverse Byte Ordering (RBO)

This bit controls the order of bytes in a 16-bit word transfer. When RBO is set low, the first byte to be read from the disk will be placed in the least significant half of the word (AD0–AD7) and when RBO is set high, the LS byte will be mapped to AD8–AD15. This is only valid for data entering the FIFO. This byte should be reset for 8 bit transfers.

Local Slow/Fast Read and Write (LSRW)

This bit can add one wait state cycle to the DMA transfers. When this bit is reset the Read and Write cycle is 4 clock periods, but if this bit is set, one wait state is added and the read or write cycle is 5 clock periods. This extends the RD and WR strobes by one bus clock period, and allows the DDC to access slower memories.

Long Address (LA)

This bit determines the Local DMA address bus width and is only valid if Local DMA is enabled and the Remote channel is disabled. When Long Address is low, 16 address bits are issued and strobed by ADS0 pin. When Long Address is high, 32 address bits are issued, the lower 16 bits are strobed by ADS0 pin, the upper 16 address bits are strobed by ADS1/RRQ pin. The most significant 16 address bits are only issued when a rollover from the least significant 16 address lines occurs or on the first DMA cycle of a multi-byte transfer. When the most significant 16 address bits are issued, that DMA cycle is 5 clock periods long if no internal or external wait states are used.

Dump FIFO/Exact Burst (LTEB)

This bit controls how the data is burst to/from the FIFO. If this bit is reset the FIFO will fill or empty completely. When the disk is being read, the FIFO will wait until the FIFO is filled to the programmed threshold. At this time the DDC will completely empty the FIFO to buffer RAM even if more data entered the FIFO during the burst. During a disk write, when the FIFO is emptied to the programmed threshold, the DMA will fill the FIFO.

When this bit is set, the DMA will only transfer a fixed number of bytes to/from the system. For reading the disk, when the FIFO fills to the programmed threshold, the DMA will burst the exact number of bytes that are in the FIFO. Any bytes entering the FIFO after the burst begins will be transferred at the next burst. For a disk write, when the FIFO empties to the selected threshold, an exact number of bytes will be DMAed to the FIFO, whether the FIFO has emptied more or not.

Local Burst Length Select (LBL1, LBL2)

Bits LBL1 and LBL2 offer different burst lengths and the thresholds according to when data will be transferred to or from the FIFO. These burst lengths/thresholds could be 2, 8, 16 or 24 bytes, if these bits are programmed, 00, 01, 10, or 11, respectively.

REMOTE TRANSFER REGISTER (Address = 37H)

This write-only register, shown in *Figure 5.13*, determines the mode of transfers between the local buffer and the system I/O port (remote DMA channel) if the DDC is in dual channel mode. The register should be loaded at initialization and normally need not be written to again. It is not affected by reset or abort.

Remote DMA Enable (SRD)

This bit, when set high, configures the DDC in dual channel DMA mode, and enables the remote DMA channel.

Remote Word Data Transfer (RWDT)

The data bus may be configured to be either 8 or 16 bits during remote transfers between buffer memory and main system. If RWDT is high, transfers are 16 bits wide, and the remote address information is incremented by 2 each memory cycle and bit A0 remains low. If RWDT is low transfers are 8 bits wide, and the remote address increments by 1.

Enable External Wait (EEW)

If EEW is high, the EXTERNAL STATUS pin of the DDC will be enabled to supply wait states in both the local and remote DMA bus cycles. A side effect using this feature is that external synchronization and external ECC cannot be used. When EEW is low, no external wait states can be inserted, and external synchronization and external ECC may be used.

Remote Slow Read/Write (RSRW)

This bit allows for slower memory or slower system cycle time and is only valid if the Remote DMA Enable bit is high. In this case, if RSRW is set, each remote DMA cycle becomes five clock periods rather than four, and both the $\overline{RD}$ and $\overline{WR}$ strobes are widened by one clock period.

D0	SLD	SELECT LOCAL DMA	0 = Disabled 1 = Enabled
D1	LWDT	LOCAL WORD DATA TRANSFER	0 = 8-Bit 1 = 16-Bit
D2	RBO	REVERSE BYTE ORDER	0 = AD0–7 = LSB, AD8–15 = MSB 1 = AD0–7 = MSB, AD8–15 = MSB
D3	LSRW	LOCAL SLOW READ AND WRITE	0 = 4 Clock Transfer 1 = 5 Clock Transfer
D4	LA	LONG ADDRESS	0 = 16-Bit 1 = 32-Bit
D5	LTEB	LOCAL TRANSFER EXACT BURST	0 = Transfer Until FIFO Empty 1 = Transfer Exact Burst
D6 D7	LBL1 LBL2	LOCAL BURST LENGTH (FIFO THRESHOLD)	00 = 1 Word/2 Bytes 01 = 4 Words/8 Bytes 10 = 8 Words/16 Bytes 11 = 12 Words/24 Bytes

FIGURE 5.12. Local Transfer Control Register

Tracking Mode (TM)

This bit configures the DDC in Tracking or Non-Tracking DMA modes when it is set high or low, respectively. In Non-Tracking mode, the DMA channels are independent and addresses are allowed to overlap, while in Tracking mode, channel addresses are maintained at least one sector apart.

Remote Transfer Exact Bursts (RTEB)

When the DDC is performing a remote transfer, the condition of this bit determines when RRQ is de-asserted after the exact number of words or bytes, specified by RBL1 and RBL2, have been transferred. If RTEB is low, RRQ wil remain asserted until the whole count, specified by the Remote Data Byte Counter, has been transferred.

Remote Burst Length (RBL1, RBL2)

These bits select the burst transfer lengths during a remote transfer operation between buffer memory and a system I/O port if RTEB is set high. These length could be 2-byte (1-word), 8-byte (4-word), 16-byte (8-word) and 32-byte (16-word).

D0	SRD DMA	SELECT REMOTE	0 = Disabled 1 = Enabled
D1	RWDT	REMOTE WORD DATA TRANSFER	0 = 8-Bit 1 = 16-Bit
D2	EEW	ENABLE EXTERNAL WAIT STATE	0 = Disabled 1 = Enabled
D3	RSRW	REMOTE SLOW READ AND WRITE	0 = 4 Clock Transfer 1 = 5 Clock Transfer
D4	TM	TRACKING MODE	0 = Non-Tracking 1 = Tracking
D5	RTEB	REMOTE TRANSFER EXACT BURST	0 = Transfer Whole Count 1 = Transfer Exact Burst
D6 D7	RBL1 RBL2	REMOTE BURST LENGTH	00 = 1 Word/2 Bytes 01 = 4 Words/8 Bytes 10 = 8 Words/16 Bytes 11 = 12 Words/24 Bytes

FIGURE 5.13. Remote Transfer Control Register

REMOTE DATA BYTE COUNT REGISTERS

(Address = 1AH (LSB), 1BH(MSB))

These registers determine the byte count required in a remote data transfer. They are preloaded with a maximum count equal to the desired total DMA transfer, and are decremented by the DDC after each transfer until a count of zero is reached. This counter is 16 bits wide, therefore up to a total of 65,536 bytes can be transferred. Presetting this register to all zeroes transfers 65,536 bytes. The count can be read at any time during the transfer.

This register is also used in tracking mode DMA to keep track of whether a sector is ready to be transferred by the remote channel.

DMA ADDRESS (BYTE0-3) REGISTERS

(Address = 1CH-1FH)

The DMA address registers issue dual channel local and remote addresses during the local and remote transfers or single channel addresses. When using the dual channel DMA mode, registers 0 (LSB) and 1 (MSB) are used for holding and incrementing the local DMA channel address. Registers 2 (LSB) and 3 (MSB) contain the address information for remote DMA transfers. These registers can be preset to any address (within 64k) and can be read at any time.

In single channel DMA mode, the 4 registers are concatenated to form a 32-bit address, thus the single DMA channel can address up to 4 Gigabytes.

5.2 DDC COMMANDS

The DDC can be configured to perform various disk and related operations. These include disk read and write operations, error correction operation, formatting operations, and DMA operations. DMA and error correction commands are considered separately later. To understand the operation better, it is useful to break up the DDC's execution of a command into three phases:

1. Command Entry — μP loads bytes and command word prior to execution.
2. Command Execution — Once loaded the DDC performs the operation.
3. Result — After execution is terminated the μP reads the DDC to determine whether an error terminated the operation.

Since the DDC is primarily reading and writing to the disk drive, it has a rich set of disk read and write functions and each of these can be used is several modes. This creates a large array of commands that provides versatility. However there are about 10-15 commands/modes that would normally be used. This section will present the basic command operations first, then common commands are shown as combinations of the operations and finally a simplifying list is created.

A typical read/write command is composed of two operations; ID field operations and disk data field operations. Bits DO2, DO1, HO2, HO1; FMT in the Drive Command Register, determine the command to be executed. The two least significant bits enable some specific options to the commands. A list of all the possible commands is given in *Figure 5.13.* These commands are executed by setting up all the registers with the desired modes and header information, and as the last step the Drive Command register is loaded with the command. Once loaded the DDC will execute the command.

The header and data operations are described individually below. They repeat some of the information given in the register bit description in Chapter 4. Following the header and data operations, the more useful combinations of these are described.

5.2.1 Header Operations

Ignore Header

The DDC will use the preamble for PLO locking, and will check for the ID field's byte synch fields. The DDC will ignore the actual header contents, and treat any values as a match, and a data field operation will proceed on the subsequent data field.

DO2	DO1	HO2	HO1	FMT	
0	0	0	0	0	No Operation, No Format (No Operation)
0	1	0	0	0	Ignore Header, Check Data, No Format
1	0	0	0	0	Ignore Header, Write Data, No Format
1	1	0	0	0	Ignore Header, Read Data, No Format (Recover Data)
0	1	0	1	0	Compare Header, Check Data, No Format
1	0	0	1	0	Compare Header, Write Data, No Format (Normal Write)
1	1	0	1	0	Compare Header, Read Data, No Format (Normal Read)
0	1	1	0	0	Write Header, Check Data, No Format
1	0	1	0	0	Write Header, Write Data, No Format
1	0	1	0	1	Write Header, Write Data, Format (Normal Format)
0	1	1	1	0	Read Header, Check Data, No Format (Get Header Info)
1	1	1	1	0	Read Header, Read Data, No Format
0	0	0	1	X	Compare Header, No Data Operation *** ILLEGAL COMMAND ***
0	0	1	0	X	Write Header, No Data Operation *** ILLEGAL COMMAND ***
1	1	1	0	X	Write Header, Read Data *** ILLEGAL COMMAND ***
0	0	1	1	X	Read Header, No Data Operation *** ILLEGAL COMMAND ***
1	0	1	1	X	Read Header, Write Data *** ILLEGAL COMMAND***

FIGURE 5.14. The DDC Commands

Compare Header

This operation usually preceeds a normal disk read or write in which a particular sector is operated on. After preamble and synch fields are compared, the DDC compares every byte in the header to the pattern registers. Only if a complete match of the header bytes and no CRC/ECC error occurs, then the data field operation is executed.

Write Header

The DDC will write header information typically when formatting the disk, but operations allow the DDC to write individual headers in a hard sectored disk as a method of correcting a header. In this operation the entire ID field is written, preamble, both byte synch fields, header bytes, and CRC/ECC bytes.

Read Header

This is used when the host desires to know what a header is, usually when trying to determine where a disk drive head is located. This operation will compare the synch fields and then read the header bytes into the FIFO. CRC/ECC is checked.

5.2.2 General Data Operations

No Data Operation

This bit combination is valid only when the header operation is ignore header. Using this with other header operations will cause "unpredictable" results. Using this with the Ignore Header function is a NOP command and can be used when the user wishes to change non-command bits in the command register without executing a disk command.

Check Data

This is essentially a data NOP. If executed the DDC will read the data field just like a read data operation, but no data will be transferred to the system. The data field CRC/ECC is checked at the end of the operation.

Read Data

To fetch data from the disk drive this operation is used. The DDC first checks the data synch field to byte align and then it reads the data from the disk drive and sends it to the FIFO for transfer to external memory. After all bytes have been transferred, the data field's CRC/ECC bytes are checked.

1

Write Data

This is the inverse of the read data command. The DDC will begin by writing the data preamble field and synch fields. Data is input from the external memory into the FIFO, and this data is written to the disk. During the writing of data, CRC/ECC is being generated, and is appended to the data field.

5.2.3 When a Command Starts

As a command is loaded, the microprocessor can decide when the DDC should start the operation. There are two choices, which are programmed by loading bit SAIS in the Drive Command Register, Start at Index or Sector.

If this bit is set, the operation will be started at the beginning of the next sector (if hard sector) or immediately (if soft sectored). This is normally used for a normal read or write operation. Since the operation starts asynchronously to where the head is located over the track, starting immediately means that the sector will be read/written within one disk revolution worst case. If the command is started on an index pulse, it is likely that the head will pass the sector before it gets to the index to start the command, wasting a full revolution.

If this bit is reset when a command is loaded, the DDC will wait for the disk to revolve until the index pulse is seen. This mode would usually be used for track oriented operations. For example, a format and/or multi-sector operation would be most useful if started at the beginning of a track rather than the middle.

It is possible to execute any valid command starting either at the next sector or at the index pulse, thus many specialized command combinations are possible.

5.2.4 Multi-Sector Versus Single Sector Operations

All of the disk command op codes listed below have a multi-sector bit that determines whether the operation is a multi-sector or single sector operation. The DDC can be programmed to read one specific sector, several sectors, or an entire track in one operation. In a multi-sector operation the DDC will read a group of logically or physcially contiguous sectors. Logically contiguous means the DDC reads sectors with sector numbers that are in numerical order, but need not appear physically in order on the drive's track. For example, sectors are numbered physically on the drive as 5, 4, 6, 1, 3, 2, will be read 1, 2, 3, 4, 5, 6. This enables interleaving sectors. In order to do this the multi-sector bit must be set and several other registers and modes must be determined.

Usually a single sector operation can be considered the default operation, and additional steps must be taken to execute a multi-sector command. To review, a single sector operation will first execute a header operation followed by a data operation and then terminate. For a normal read or write, the header bytes are compared to the header pattern registers, and when the desired header is found the data field is operated on.

In a multi-sector operation, the DDC will re-execute the same command over again. The number of times the command is executed depends on the value programmed into the Number of Sector Operations Register (Address = 13H) with a maximum of 255. For most read or write commands, multi-sector operations will also need to select and program the Sector Counter (although some commands won't). For a given disk ID format one of the header bytes is the sector number. The header byte that has been designated the sector number must have its control register programmed to substitute the sector counter for the header pattern register. The sector counter is then programmed with the number of the first sector to be read/written.

After the command starts, the first sector is operated on. Then the Sector Counter increments and the Number of Sector Operations Counter decrements. If this has not reached zero the command is re-executed until the Number of Sector Operations Counter does reach zero. Programming a 1 into the Number of Operations Counter will cause the command to execute once.

The preceding describes how a multi-sector read or write would normally be executed. An example of when the Sector Counter may not need to be enabled might be a track dump command which is a read header-read data. Since sector numbers are not compared all the header fields the data fields can be sequentially read.

5.2.5 Interlock Mode Operation

The Interlock mode can be used to enable a microprocessor to update commands or parameters on the fly just after the previous command finishes with the header operation. This enables the μP to execute a series of commands on contiguous sectors. These commands may be the same one repeated, a format for example, or a different one executed sequentially. Interlock is enabled by setting the Interlock bit, and enabling the header interrupt in the Operation Command register, or polling the Next Disk Command status bit.

In the Interlock mode, and when a command is first issued, an interrupt at the end of the header operation informs the μP that the DDC is ready to accept a new command. The μP updates the header, or command registers, and lastly writes to the Header Byte Counter Register. This update must take place before the end of the CRC/ECC field of the present sector. If the Header Byte Counter is not written to before the header of the following sector, the DDC will set the Late Interlock bit in the Error Register, and abort the operation.

The Interlock mode can be used either in single sector or multi-sector operations. In a single sector operation, the μP updates all the header and control registers. It then writes the new command into the Disk Command register, and finally writes to the Header Byte Counter. This enables different commands to act on physically sequential sectors, i.e. read to one, write from the next.

In a multi-sector operation, the μP sets the Number of Sector Operations to the number of sectors to be read or written. The μP writes the command to the Disk Command register once at the beginning of an operation. Then the μP updates the header and mode registers after each interrupt. Finally, before the end of each data ECC field the Header Byte Counter is loaded. The DDC will automatically repeat the command until the Number of Sector Operations counter reaches zero. This command mode is useful for operating on physically sequential sectors that are not logically sequential. For example, formatting a track with interleaved sector numbers.

While executing in interlock mode, any pattern or count registers may be written to. However, it is recommended to not write to data pattern or count registers, as timing of the μP write relative actual disk data being operated on will determine whether the write will effect the present or next sector. During the operation, the pattern or count registers **must not** be read, as this will cause spurious operation.

Actually, the Interlock mode is somewhat misleading. For any command being executed, the DDC will be ready for the next command after it has successfully completed a header operation. The only action the Interlock mode takes is enabling an Error bit that tells the μP when it didn't update the DDC in time.

5.2.6 Command Termination, Resetting, and Re-enabling

Once a command starts execution, it will perform its desired task, or an error will be encountered that will prevent the command from executing. These errors could result from reading the disk, losing data while not transferring it fast enough, or not finding the header it is looking for after two index pulses have occurred. In these cases, the DDC will terminate the operation, set an error flag, and set itself into an error state from which it must be reset before the next command can be executed.

The errors that the DDC recognizes are listed in the Error register. Errors caused by corrupted disk data or format are:

Data Field Error
No Data Synch
Sector Not Found
Sector Overrun

Errors caused by not transferring data or parameters to the disk fast enough are:

FIFO Data Lost
Late Interlock

Errors that occur because the disk controller is looking for a sector header that is non-existent, or the disk head is on the wrong track are:

Sector Not Found
Header Failed Although Sector Matched

Additionally, during a correction cycle an error will be indicated if the correction failed and a Correction Failed error is set.

When an error occurs the DDC will terminate the command, and will issue an interrupt (if enabled). Once the error is flagged, the CPU must read the error register and then reset the DDC.

To reset the DDC, the CPU first must set the Reset bit in the Operation Command register. Then it must reset the Reset bit. This has reset the DDC into the default state. Now the DDC can be re-enabled by setting the Re-Enable bit high, and the DDC is then ready to receive the next command.

5.2.7 Summarizing Most Useful DDC Commands

As one can see there are a multitude of possible commands that the DDC can implement, and the header-data operations with the various modes tend to be very cryptic. To try to simplify the commands, *Figure 5.15* lists most of the common commands, a mnemonic, and the command op code. These commands assume the disk format is fairly standard, with the header at least containing one byte for the sector number. These are by no means all of the commands. Some specialized ones may be desirable, and can be assembled from the previous descriptions. Or, if the user decides to be creative with the header format other commands or modes may be useful, and maybe encryption/decryption of the header for data security could be implemented.

These commands are header-data operations with the multi-sector and start on sector or index bit configured to their most common way. For example, Read Sector (SRD) starts on a sector pulse (or immediately) and is not multi-sector. Multi-Sector Read Track is multi-sector and normally would start on the index pulse (but doesn't have to) so that the entire track's data can be read starting at the physical beginning of the track.

The read, write, format track commands assume that the Number of Sector Operations register is loaded with the number of sectors per track.

The logical multi-sector read, write commands assume that either the Sector Counter is enabled, or the Interlock mode is used.

Command Name		Op Code	
Read Single Sector	RDSS	11010010	D2H
Read Sector ID	RDID	01110010	72H
Read Multi-Sector Logical	RDMS	11010110	D6H
Read Track	RDTK	11010100	D4H*
Read Track Blind	RDTB	11000100	C4H*
Read ID Multi-Sector	RDIM	01110100	74H
Read Track Data/ID	RDDI	11110100	F4H*
Write Single Sector	WRSS	10010010	92H
Write Multi-Sector Logical	WRMS	10010110	96H
Write Track	WRTK	10000100	84H*
Format Track	FMTK	10101100	ACH
Format Track No Gap	FMNG	10100100	A4H
Find ID	FNID	01010010	52H
Find ID Multi-Sector	FNMS	01010110	54H
Recover Header	RCID	01100010	62H
Re-Enable Controller	RENB	00000001	01H
No Operation	NOP	00000000	00H

***Note:** For an entire track operation, the Number of Sector Operations Counter should be set to the number of sectors per track.

FIGURE 5.15. Common Configurations of the Command Bits

SINGLE SECTOR READ (Compare Header-Read Data)
Op Code = 11010010

This command is used to perform a normal disk read operation by disabling the multi-sector operation and starting immediately/sector pulse. The header bytes loaded into the DDC are compared to header information read off the disk drive. The DDC continues to scan the drive until a match is found. Once a header has matched, data in the subsequent data field is read from the disk and transferred to the system via the internal FIFO and DMA operations.

READ SECTOR ID (Read Header-Check Data)
Op Code = 01110010

This is a single sector command, which starts immediately. It will read the first sector header that the drive head passes over, and transfers it to the external memory. This is useful if the system gets lost and would like to know where the drive head is without recalibrating the drive to track zero.

READ MULTI-SECTOR LOGICAL
(Compare Header-Read Data)
Op Code = 11010110

This is a multi-sector command that starts immediately. There are two modes that can be used for this command. One is to use the sector counter to sequentially read logical sectors. The second is to use the Interlock mode. If the sector counter is used then the logical sectors are read sequentially by sector number. In the Interlock mode the logical sectors can be read in any logical sequence, depending on µP update of header bytes.

This command can be modified to do a single sector read in multi-sector mode, by setting the Number of Operations to one and, preferrably, changing the Start bit from start on index to start immediately.

READ TRACK (Compare Header-Read Data)
Op Code = 11010100
The Read Track command has the same op code as the Multi-Sector Logical Read, except that the command is started on an index pulse. This will cause a read of all sectors ensuring that all other header bytes are compared and header CRC/ECC is checked. Of course the Number of Sector Operations counter must be set to the number of sectors per track. If the track is to be read in a logical order, then the sector number header byte can be compared to the Start Sector register. For a physically contiguous read the sector number header byte can be set to not compare.

READ TRACK BLIND (Ignore Header-Read Data)
Op Code = 11000010
This command will not compare the header field for a match, but will read the first data field that the DDC encounters and all subsequent data fields no matter what the header contains. The DDC will read in the data from the drive and DMA it to external memory.

READ ID MULTI-SECTOR (Read Header-Check Data)
Op Code = 01110100
This is a multi-sector command that starts at an index pulse and reads every header on the track (assuming the Number of Sector Operations equals the number of sectors on the track). This can tell the system what the entire track's header format.

READ TRACK ID AND DATA (Read Header-Read Data)
Op Code = 11110100
The read track command uses the Read Header-Read Data command in multi-sector mode, starting on the index pulse, and setting the Number of Sector Operations counter to the number of sectors on a track. This will read both the header and data fields of all sectors on a track. This can be used as a diagnostic tool to dump the entire contents of the disk drive's track.

WRITE SINGLE SECTOR (Compare Header-Write Data)
Op Code = 10010010
This command is used to perform a normal disk write operation to an individual sector. The multi-sector bit is reset and the command is started on a sector pulse or immediately. The header bytes are compared as in a disk read. Once the header matches, data is written to the associated data field by the DDC.

WRITE MULTI-SECTOR LOGICAL
(Compare Header-Write Data)
Op Code = 10010110
This is the same as the Multi-Sector Logical Read command. This one starts immediately, and can use the Interlock mode or Sector counter to change the header information. The sector counter enables logically sequential sector writing, and Interlock mode enables µP to update sectors on the fly.

As in the read command, by programming the Number of Sector operations counter to 1 results in a multi-sector mode single sector write. Also the start operation bit should be set to start on a sector pulse.

WRITE TRACK (Compare Header-Write Data)
Op Code = 10010100
The Track Write is similar to the Multi-Sector Logical Write command, but several modes are set up differently. First this command starts on an index pulse, and second the header byte corresponding to the sector number is programmed to not compare. This causes every sector to be written to in a physically sequential order.

FORMAT TRACK (Write Header-Write Data)
Op Code = 10101100
This command is commonly used when disk formatting is to be performed. The format bit in the disk command register normally should be set to execute this command. The entire ID Field and Data Field, one or more sectors, are written to the disk either from the header byte registers or from the FIFO, depending on which mode of disk formatting is selected (refer to chapter on Disk Formatting). The remaining header and data fields are written from the respective pattern registers.

FORMAT TRACK NO GAP (Write Header-Write Data)
Op Code = 10100100
There is a special format option which can be used in hard sectored drives by not setting the format bit in the Disk Command register and using a multi-sector operation. This enables writing of a format without intersector gaps. This may be useful if the drive puts servo information in the gaps. This would be overwritten with the normal format.

FIND HEADER/ID (Compare Header-Check Data)
Op Code = 01010010
This command is normally used to perform a diagnostic operation, operating almost like the Compare Header-Read Data command. It will first scan and compare the header information. The data field is read but not transferred.

FIND ID MULTI-SECTOR (Compare Header-Check Data)
Op Code = 01010100
This command is a multi-sector version of the Find Header, and can be used to verify all the headers on a track are valid.

RECOVER HEADER (Write Header-Check Data)
Op Code = 01100010
This command will start immediately, and rewrite the header field for the first sector encountered. This can be used as a method of recovering a damaged or unreadable ID field. To do so actually requires the interlock mode executing this command following a Find Header command. To recover the sector in error, the sector physically preceeding is found and the Find header is executed followed immediately by this command. The new header is then written over the damaged one.

RE-ENABLE (Re-enable bit set)
Op Code = 00000001
This command is used as part of the Reset/Re-Enable operation which is normally used to recover from an error condition. First the Reset bit in the Operation Command register is set, then reset. To finally enable the DDC for another command, the Re-Enable bit is set (see previous section).

NO OPERATION (Ignore Header-No Data Operation)
Op Code = 00000000
As mentioned previously, this is a NOP command. The DDC will not perform any operation.

ILLEGAL COMMANDS

Figure 5.14 lists several commands as illegal commands. They are not implemented by the internal PLA, and executing these commands will cause erroneous results.

CHAPTER 6 System and Disk Interfacing with DDC

6.0 INTRODUCTION

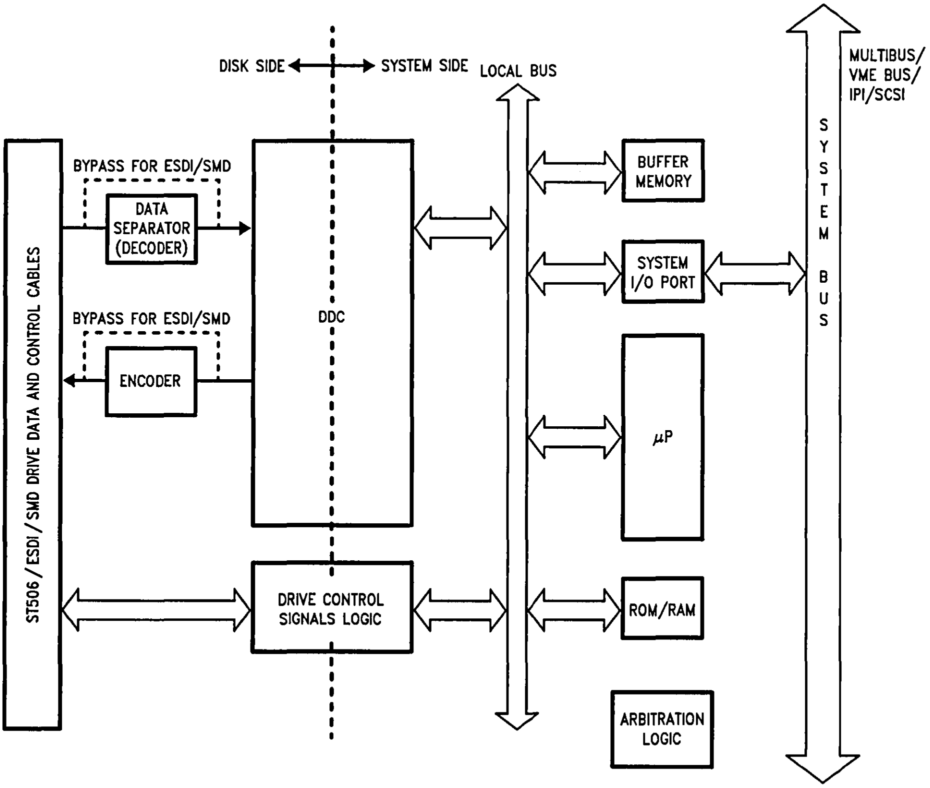
In a typical disk controller design, the DDC interfaces to the controlling microprocessor, memory and/or main system bus (such as MULTIBUS®, VME bus or IPI and SCSI) on the system side, and to the disk drives (via various disk interfaces such as ST506, ESDI and SMD) on the disk side. *Figure 6.1* shows the DDC in a typical disk controller design. As mentioned in previous chapters, the DDC (disk data controller) controls only the disk data path. It takes 8- or 16-bit wide data from the system bus, serializes it and then writes it to the disk in a disk write operation. While reading the disk data from the disk, the DDC deserializes the data and puts it back to the system bus. Before any disk operation can be performed, a track seek operation must be performed. The drive control signals required to perform a seek operation, can be generated using additional simple circuitry.

In this chapter, various DDC hardware interfaces to the host system and disk drive will be discussed in detail. DDC programming algorithms for carrying out disk operations are discussed in detail in chapter 7. The hardware interfacing is divided into two parts, system side and disk side.

6.1 SYSTEM SIDE INTERFACE

The DDC goes through three configuration modes when it performs a disk operation. When a microprocessor accesses the DDC to initialize it for a particular disk operation, it becomes a peripheral to the microprocessor (peripheral mode). While performing a disk operation and whenever the FIFO requires a data transfer to or from the external memory, the DDC becomes the bus master (master mode) and uses its on-chip DMA channels to perform the desired data transfers. If the desired data transfers are carried out by an external DMA controller, the DDC becomes a slave to the external DMA (slave mode controller).

In the following sub-sections the DDC's hardware interfacing with the microprocessor (peripheral mode), memory (master mode), and external DMA (slave mode) are discussed. The logic needed for arbitration of bus control between the DDC and microprocessor, and, to provide drive control signals are also discussed. In addition to the hardware connections, typical timing for various signals is also discussed.



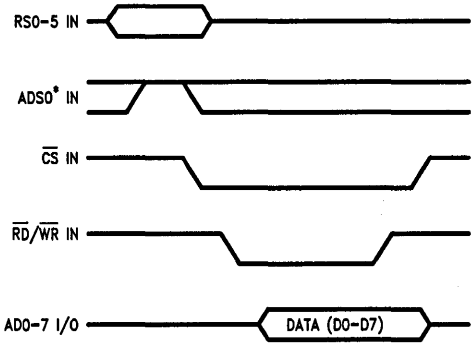
TL/F/8663-81

FIGURE 6.1. A Typical DDC-Based Disk Controller

6.1.1 Microprocessor-DDC Interface

The DDC must be initialized by a microprocessor (or microcontroller) in order to perform various disk operations. *Figure 6.2(a)* shows a basic microprocessor-DDC interface. When the microprocessor accesses the DDC, all 64 internal registers appear to it as unique memory or I/O locations. Each register can be randomly accessed and operated on. Only eight bits of data can be transferred to or from these registers using pins AD0–7. All the registers can be individually selected using six register select lines, RS0–5. Using these dedicated lines with an address strobe input, ADS0, the chip can be used in both multiplexed and demultiplexed address bus environments. Basically, the ADS0 and RS0–5 together act like a flow through latch.

With multiplexed address and data lines, a positive strobe pulse on ADS0 will latch the address. The ADS0 line may be derived from a microprocessor address strobe line such as ALE. In systems with a dedicated address bus (demultiplexed), ADS0 may be pulled high to allow address information to flow through the latch. Finally, by applying $\overline{\text{CS}}$ and $\overline{\text{RD}}$ or $\overline{\text{WR}}$ strobes, the selected register is accessed. *Figure 6.2(b)* shows a typical timing for a multiplexed and demultiplexed system bus when the DDC is in Peripheral mode.

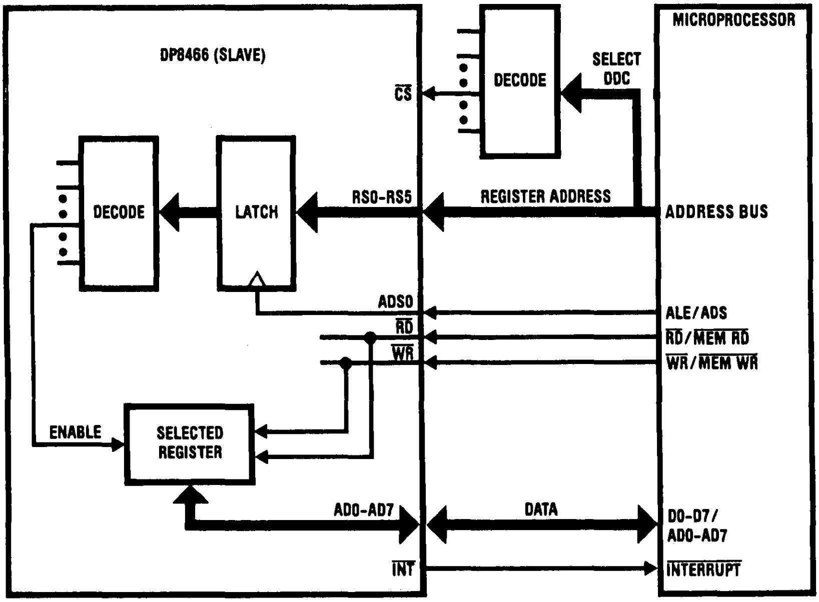

RS0-5 IN
ADS0* IN
$\overline{\text{CS}}$ IN
$\overline{\text{RD}}/\overline{\text{WR}}$ IN
AD0-7 I/O
DATA (D0-D7)

TL/F/8663-83

FIGURE 6.2(b). DDC Register Read/Write in Peripheral Mode

*Latched Register Select: ADS0 = Active

Non-Latched Register Select: ADS0 = Tied High



TL/F/8663-82

FIGURE 6.2(a). A Simplified Microprocessor Interface in Peripheral Mode

6.1.2 Bus Arbitration Logic

When the FIFO fills up (during a read operation) or empties (during a write operation) to the progammed threshold level, the DDC issues a local request (LRQ) to carry out local buffer memory transfers. Similarly, the DDC issues a remote request (RRQ) when local buffer memory needs a data transfer on the remote channel. Upon receiving the request (LRQ or RRQ), the current bus master should generate an acknowledge (LACK or RACK) to transfer bus control to the DDC. The request essentially puts the microprocessor (current bus master) on hold. The DDC keeps LRQ or RRQ asserted for the entire selected data burst transfer. This enables the DDC to remain bus master during this time while the microprocessor is on hold. See *Figure 6.3(a)* for typical RRQ-RACK and LRQ-LACK timing.

Generally, the LRQ-LACK or RRQ-RACK pins of the DDC are connected to the HOLD-HOLDA type (or BUSREQ-BUSACK type) of pins on the microprocessor through some additional combinational logic. This additional logic would be responsible for providing a smooth bus arbitration between the DDC and other possible bus users (like microprocessor). It must remove the possibility of any bus contention and may also prioritize DDC's DMA requests with other requests which may be present in the system. See *Figure 6.3(b)* for typical bus arbitration logic connections.

The Address strobes (ADS0 and ADS1) from the DDC and similar address strobe signals from the microprocessor may also be used in the bus arbitration logic, to generate a common set of address strobes for the system. This is useful if the demultiplexing address latches are to be shared between the DDC and the μP.

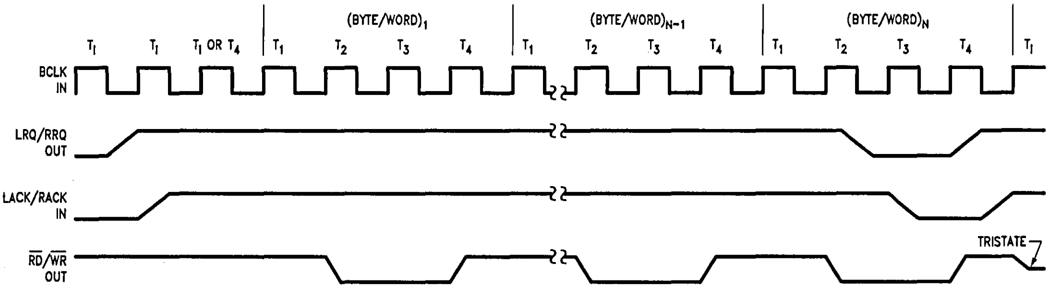
The DDC samples the RACK and LACK inputs during T_i(idle) or T_4 for each memory transfer. In general, the arbitration logic should leave LACK/RACK high for the duration of the burst. It may also toggle LACK/RACK so long as LACK/RACK are high during T_4 when the local channel has transferred one sector (in tracking mode-dual DMA), or when the remote DMA is enabled (in non-tracking mode).

6.1.3 The DDC-Memory Interface

After becoming bus master, the DDC communicates with the buffer or system memory if its on-chip DMA is chosen for data transfers. This communication takes place via a 16-bit multiplexed address/data bus, AD0–15, and supported by $\overline{\text{RD}}$, $\overline{\text{WR}}$, ADS0 and ADS1 strobes. The DDC is programmed in one of the three basic DMA modes; single channel, dual channel non-tracking and dual channel tracking. (See chapter 7 for detailed explanation on DMA programming). The hardware aspects of using the DDC in one of these modes are discussed below.

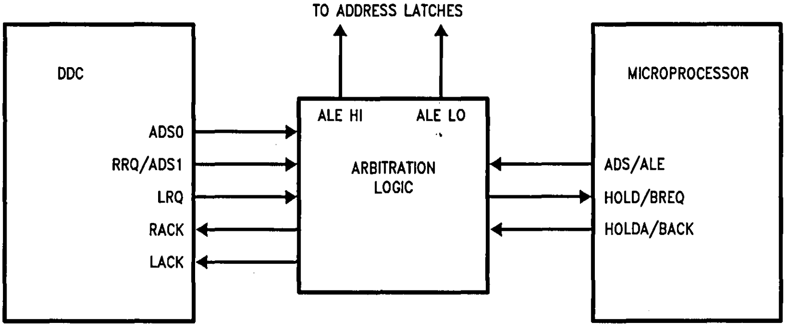
SINGLE CHANNEL DMA

In single channel DMA mode, the DDC takes care of data transfers between the FIFO and external memory using its local DMA channel. In this mode, the DDC can be set to generate either 16-bit or 32-bit of address. This gives system architects a great deal of flexibility. With 32-bit single channel DMA, up to 4 GBytes of memory can be accessed which is ideal in a system where the buffer memory is located within the main system memory. Factors like sector length, number of sectors per track and DMA burst length should be considered in determining the appropriate buffer memory size.



TL/F/8663-84

FIGURE 6.3(a). A Typical LRQ/RRQ-LACK/RACK Timing for a Data Burst Transfer



TL/F/8663-85

FIGURE 6.3(b). Arbitration Logic Connections

DUAL CHANNEL DMA

For systems where disk data is first buffered in a local memory before being transferred to the host memory via a host I/O port, the DDC offers two channels of DMA to handle the necessary data transfers. The "local" DMA channel controls data transfer between the FIFO and local buffer memory, while the "remote" DMA channel controls data transfer between the local buffer memory and the host I/O port. The two DMA channels can be operated independently of each other or the remote channel can be made to track the local channel. In either case, both channels can address a 64k address space. In the rare cases where a dual bus architecture and more than 64k bytes of local buffer memory are required, the DDC should be programmed for single channel, 32-bit addressing mode to handle FIFO-to-buffer transfers, leaving the buffer-to-host transfers to an external DMA controller.

For the purpose of the following discussion, it is helpful to introduce the concept of "source" and "destination" channels. In a disk read operation, the source channel is the local DMA channel, and the destination channel is the remote DMA channel. Conversely, in a disk write operation, the source channel is the remote DMA channel and the destination channel is the local DMA channel. In both cases, the source channel controls data transfers from the data source to buffer memory, while the destination channel controls data transfers from buffer memory to data destination. As an example, in a disk read operation, the source channel is the local DMA channel and controls data transfers from the FIFO to the buffer memory, the remote channel becomes the destination channel and controls transfers from buffer memory to the host.

Non-Tracking Dual Channel DMA Mode

In this mode, the local and remote channels operate as two independent DMA channels. This allows the microprocessor to tightly control data movement between the FIFO, buffer memory and the host I/O port. It also allows the microprocessor to process disk data in the buffer memory before transferring them to the destination, for example. This extra degree of control imposes an extra burden on the microprocessor. It is now responsible for preventing any destination channel transfers that may result in a data overrun situation. This can occur since the destination channel will continue to transfer data out of the buffer memory even though that section of memory has not been written to by the source channel.

Tracking Dual Channel DMA Mode

In this mode, the DDC keeps track of data transfers occurring over the two DMA channels. It forces the two DMA channels to track each other appropriately to prevent data overruns.

The tracking mechanism is implemented through a DMA sector counter, or DSC, which keeps track of the difference between the number of sectors transferred via the source and destination channels. The DSC is incremented every time a sector of data has been transferred into the buffer memory via the source channel, and decremented each time a sector of data has been transferred out of the buffer memory via the destination channel. The destination channel will not initiate the transfer of a new sector of data unless the DSC is non-zero. Thus the destination channel will not initiate the transfer of a new sector until it has been completely transferred into memory by the source channel. With appropriate choice of DMA data burst lengths, the local buffer memory then appears to the DDC as an extension of the internal FIFO.

DATA TRANSFER TIMING

Memory Read/Write Cycles

A standard DMA memory cycle consists of 4 BCLK periods, with the exception of the extended DMA memory cycle associated with the 32-bit addressing single channel DMA mode.

Referring to *Figure 6.4(a)*, a standard memory cycle consists of four bus states, T_1 to T_4, each lasting for one BCLK period. The cycle begins with the rising edge of BCLK in T_1, at which time the 16-bit memory address is put out on the address/data bus. ADS0 is also asserted during T_1 and can be used by the memory to strobe in the address on its negative edge. The low order address bits (A0–A7) are put out on port AD0–7, and the high order address bits (A8–A15) on port AD8–15. The address remains on these ports for the remainder of T_1. At the start of T_2 the address is removed and, depending on whether a read or write access is required, either the Read Strobe ($\overline{RD}$) or the Write Strobe ($\overline{WR}$) is asserted. These strobes stay asserted until the end of T_3. If a write access to buffer memory is required, the DDC will put its FIFO data on the address/data ports at the beginning of T_2 until the end of T_3. If a read access to buffer memory is required, then the DDC expects valid memory data to be placed on these ports during T_2 to T_3 so that the required data setup time referenced to the positive edge of $\overline{RD}$ is met. At the beginning of T_4, the appropriate acknowledge input (LACK/RACK) is sampled. If the sampling occurred when the acknowledge input is high, then the next memory cycle will be permitted to start at the end of T_4. Otherwise the DDC will relinquish control of the bus. If the acknowledge input is de-asserted prior to the rising edge of BCLK in T_4, the current memory cycle will be completed before the DDC frees up the bus.

The extended DMA memory cycle is used only in the 32-bit addressing single channel DMA mode. Since there are only 16 address/data lines available, the 32-bit addresses must be split into two groups of 16 bits and multiplexed onto the address/data ports. The low order 16-bit address is handled exactly as for the 16-bit addressing DMA modes. Additionally, a separate address strobe (ADS1) is provided so that the high order address bits can be stored in an external address latch.

As shown in *Figure 6.4(b)*, the extended DMA memory cycle is essentially the standard DMA memory cycle with an additional state T_0 inserted prior to T_1. At the start of T_0, the high order address bits are placed on the address/data ports for the duration of T_0. The address strobe ADS1 is also asserted and can be used by the memory system to latch in the high order address on its negative edge. Events occurring from T_1 to T_4 are exactly the same as in the case of the 16-bit addressing DMA modes.

Once initialized at the beginning of a DMA block transfer, the external high order address latch only needs to be updated whenever the lower order 16-bit address rolls over. Thus the extended DMA memory cycle is only required at the beginning of a block transfer, and each time the lower order address rolls over. Consequently, even when 32-bit addressing is required, the vast majority of DMA memory cycles will be the standard 4-clock cycle.

Prior to commencing any DMA transfers, the DDC must request and be granted control of the address/data bus. Whenever a DMA channel (local or remote) is ready for a transfer, the corresponding request output (LRQ or RRQ) is asserted to request control of the address/data bus. When such control is granted by the system via the appropriate acknowledge input (LACK or RACK), the DMA cycles will commence. *Figures 6.5(a)* and *6.5(b)* illustrate the signal timings for a burst DMA transfer over the local and remote DMA channels.

Referring to *Figure 6.5(a),* a local transfer is requested by the DDC by asserting LRQ high when the FIFO threshold is reached. The DDC samples the LACK input at each positive edge of BCLK thereafter until a logic high is detected on the LACK input. At which point the DDC assumes control of the bus and starts a burst transfer. The burst spans over N memory cycles (where N is equal to the FIFO threshold in byte mode or half the FIFO threshold in word mode), or until the FIFO is full or empty. In the last transfer cycle of a burst, LRQ is de-asserted at the start of T_2 and the DDC relinquishes control of the bus at the end of T_4.

The remote transfer operation can be programmed to transfer data in bursts or to transfer the entire block in one stream. If burst mode is selected, the remote channel will transfer data in bursts with the user-programmed number of bytes per burst as illustrated in *Figure 6.5(b).* As in the case of local transfers, the DDC first requests bus control by asserting remote request (RRQ) high. The remote acknowledge input (RACK) is then sampled at each positive edge of BCLK until it (RACK) goes high. The DDC then starts the first of a series of N transfer cycles, where N is the programmed number of bytes or words per burst. In the last cycle of the burst, RRQ is de-asserted at the beginning of T_2 and the DDC gives up control of the bus at the end of T_4 if the entire block has been transferred. Otherwise, RRQ is re-asserted at the start of T_4 to request bus control for the next burst. This allows other peripherals with equal or higher priority to gain access to the bus between bursts. The bus arbitration logic must de-assert RACK prior to T_4 to ensure that the DDC will not initiate another memory cycle. It should also hold off RACK to the DDC until such peripherals have completed their transfers.

If the remote channel is programmed to transfer an entire block in a continuous stream, then RRQ will not be de-asserted until T_2 of the last transfer cycle. If the system cannot tolerate such prolonged bus usage by the DDC, then the bus arbitration logic may de-assert RACK to force the DDC to relinquish bus control. If RACK is de-asserted prior to T_4, the DDC will complete the current transfer cycle and relinquish bus control. Otherwise the following transfer cycle will be completed before the DDC will free up the bus. When this technique is used to gain access to the bus, the arbitration logic must delay the granting of bus access to another device for time T_d from the de-assertion of RACK, where T_d is given by:

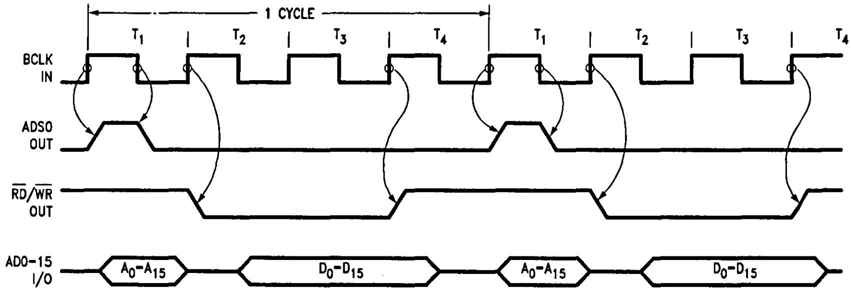
$$T_d = (\text{Period of BCLK}) + (\text{Duration of one DMA memory cycle}).$$

Note that the duration of a DMA memory cycle may vary depending on the number of wait states that may be inserted, as explained in a later section.

In both tracking and non-tracking modes, the local channel has priority over the remote channel for bus access.

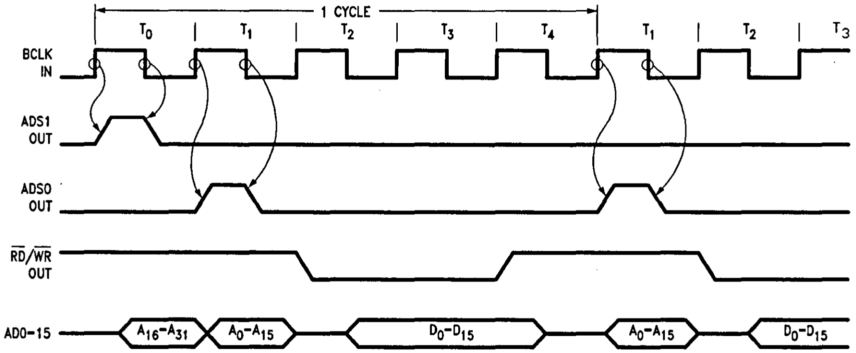
Bus Latency

The DDC can be operated at a 20 MHz Bus Clock (BCLK). With this frequency data can be transferred between the FIFO, local memory and system I/O port at the rate of 5 Mega-Transfers/sec (or 10 MBytes/sec) assuming a 4 clock periods memory cycle. With 15 Mbits/sec disk data rate, the 32-byte FIFO can be filled in approximately 17 μs.



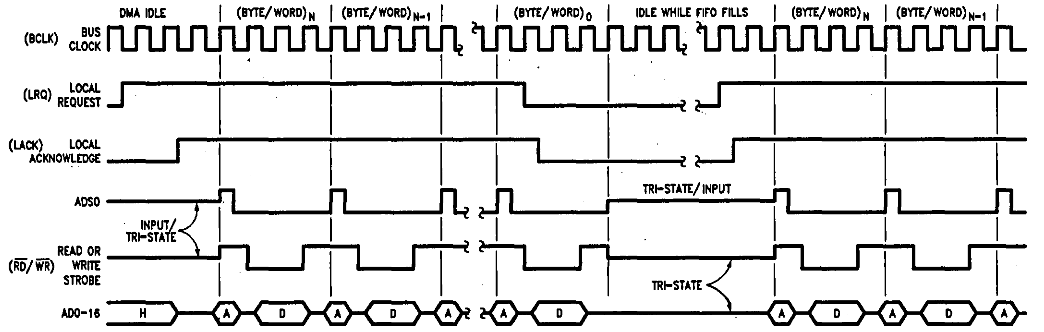
TL/F/8663-86

FIGURE 6.4(a). 4-Clock DMA Memory Cycle (DDC in 16-Bit Single and Dual Channel DMA Modes)



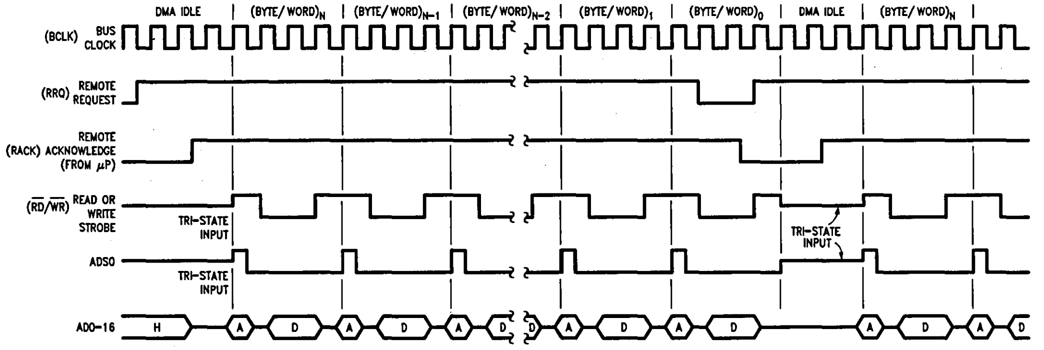
TL/F/8663-87

FIGURE 6.4(b). 5-Clock Extended DMA Memory Cycle (DDC in 32-Bit Single Channel DMA Mode)



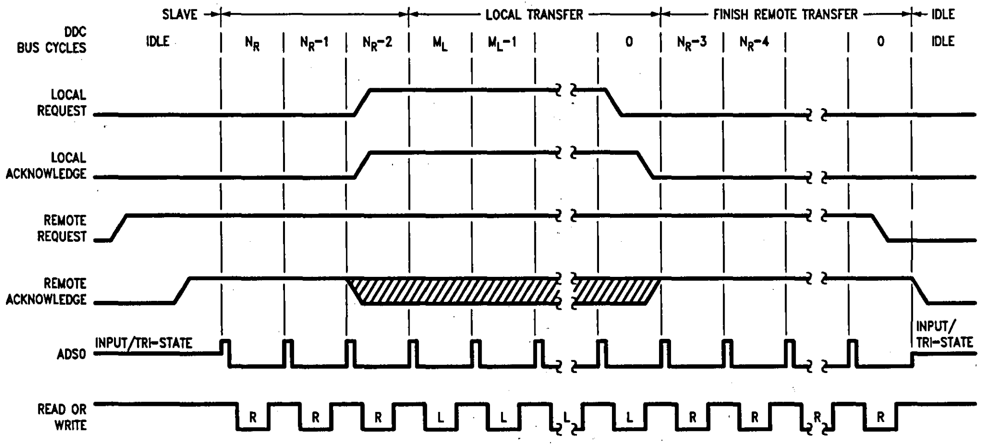
TL/F/8663-89

FIGURE 6.5 (a). Typical Local DMA Burst Transfer Timing (H = Host, A = DDC Address, D = DDC Data)



TL/F/8663-88

FIGURE 6.5 (b). Typical Remote DMA Burst Transfer Timing (H = Host, A = DDC Address, D = DDC Data)



TL/F/8663-90

FIGURE 6.5 (c). Interleaved Remote and Local Transfer Timing

Note N_R = Remote DMA Transfer Cycle Number N

M_L = Local DMA Transfer Cycle Number M

At 20 MHz bus clock (i.e. clock period of 50 ns and memory cycle of 200 ns) and assuming word-wide DMA transfers, the DDC will take 16 × 200 ns = 3.2 μs to empty the FIFO. This essentially shows that microprocessor bus is free for 17 − 3.2 = 13.8 μs (81.2% of the time). In other words, the DDC will need bus control after every 17 μs.

The selection of the FIFO threshold level should be based on the maximum amount of time the system takes to respond to a DMA request, often called bus latency. If the system has a longer latency, then a smaller threshold should be programmed. This will allow greater time for the system to respond without overflowing the FIFO. The disadvantage to programming lower FIFO thresholds is that more requests are made, tying up the system bus more often. For example, with an 8 byte threshold a request would be made about every 4.5 μs.

A second consideration is whether to have an exact burst, or burst until the FIFO empties. If the system has significant latencies then the FIFO should be emptied. If the system must not relinquish the bus for too long then a fixed burst size must be chosen.

The bus latency should be calculated for a given disk and DMA transfer rate to determine appropriate FIFO threshold. Programming the DMA for various burst transfer options is discussed in depth in chapter 7.

Local and Remote Interleave Timing

In dual channel DMA mode, the local and remote transfers can be interleaved using the DDC in tracking mode (see chapter 7 for details). A typical local and remote data transfer interleave timing is shown in *Figure 6.5(c)*. In this case a remote transfer is interrupted by the higher priority local transfer. Taking the same example given in the previous sub-section, the remote transfers can be carried out during the 81.2% of time when the local channel is not using the bus.

Slow Read/Write

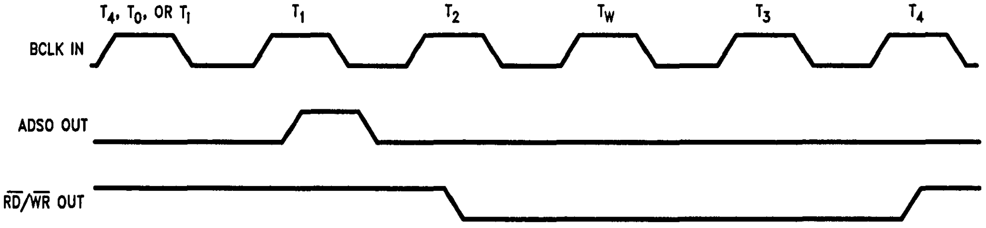
The Read or Write strobes ($\overline{RD}$ or $\overline{WR}$) can be extended by adding extra wait state(s). It can be done internally by setting the LSRW bit in Local Transfer register (and) or the RSRW bit in Remote Transfer register. This will generate an extra cycle (T_W) between the T_2 and T_3 clock cycles as shown in *Figure 6.6(a)*. These strobes can also be extended by setting the EEW bit in the Remote Transfer register in conjunction with driving the EXT STAT input high. This will add extra wait states between clock periods T_3 and T_4, as shown in *Figure 6.6(b)*. The DDC samples the EXT STAT input at the positive edge of T_3 during each DMA (local or remote) memory cycle. If EXT STAT is sensed high and the EEW bit of the Remote Transfer Register is set, then a wait state (T_W) of a bus BCLK period will be inserted after T_3. During such wait states, the DDC continues to sample EXT STAT at the positive edges of BCLK. If EXT STAT is sampled high, a new wait state will be inserted at the end of the current one. If EXT STAT is sampled low, then the next state will be T_4.

6.1.4 DDC-External DMA Interface

If the on-chip DMA is not to be used, an external DMA controller must be used to service the FIFO. A typical DDC-external DMA interface is shown in *Figure 6.7(a)*. The LRQ asserted by the DDC is acknowledged by the external DMA and the DDC becomes a slave to the DMA controller. The DMA controller carries out the local transfers and deasserts LACK after LRQ is deasserted by the DDC. A typical FIFO Read/Write sequence by the external DMA is shown in *Figure 6.7(b)*.

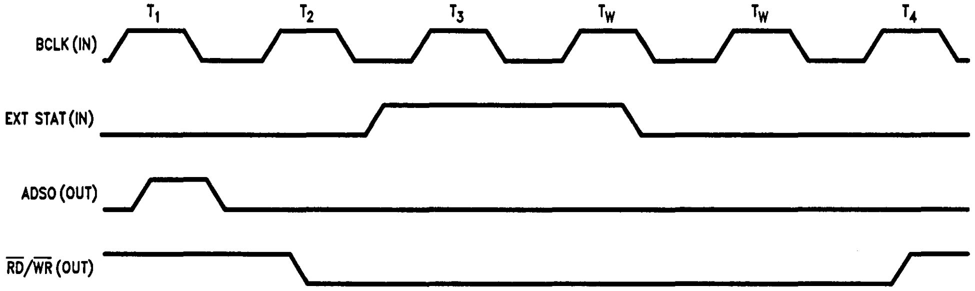
6.1.5 Drive Control Signals Logic

The drive control signals generation is not incorporated on the DDC which makes it interfaceable with any type of disk



TL/F/8663-92

FIGURE 6.6 (a). DMA with Internal Wait States



TL/F/8663-91

FIGURE 6.6 (b). DMA with External Wait States

interface. These signals can be generated simply with an I/O port associated with the controlling microprocessor. The microprocessor prepares the control signals for a particular drive interface and writes it to the I/O port. In high performance multi-drive, multi-interface systems, a dedicated microcontroller (COPS™, 8048 or HPC) or microprocessor (NSC800™, Series 32000® etc.) may be used to provide control signals for various drives. In this case, the controlling microprocessor (or host) instructs the microcontroller to perform a complete disk seek operation.

6.1.6 DDC in Typical System Configurations

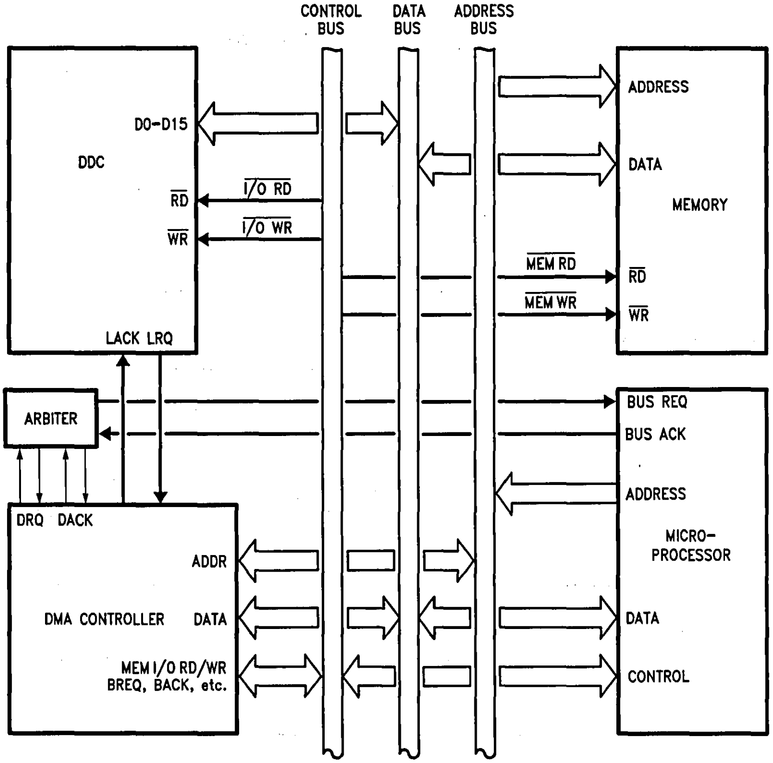
The DDC can typically be used in two types of system architectures, i.e. a single bus system or a dual bus system.

SINGLE BUS SYSTEM

Single bus systems usually are standalone systems controlled by a microprocessor. In such system, each component of the system communicates with the rest of the system through a single bus. For example, in a single board microcomputer, all the functional components like memory, I/O ports etc. communicate with the controlling microprocessor via main system bus.

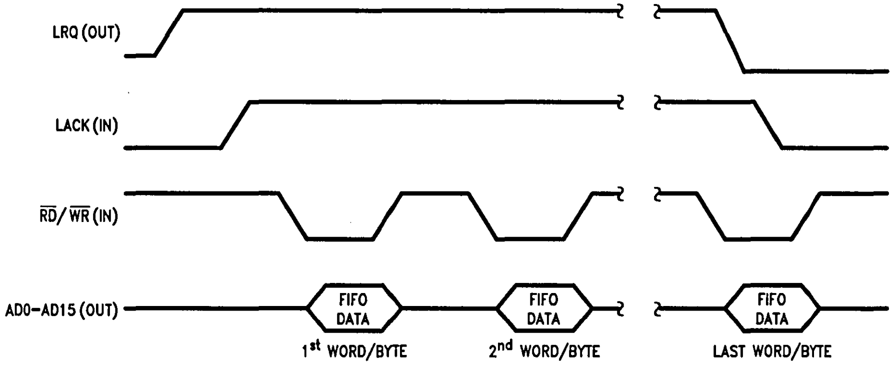
The DDC when used in such a system, becomes another component that communicates with the controlling microprocessor via the main system bus. It also communicates directly with the system memory using its on-chip DMA capability (usually the local channel). A single bus system generally requires a large system memory, so using the DDC in its single channel DMA mode becomes very worthwhile as it can access up to 4 GBytes of memory in this mode. The DDC in a typical signal bus system is shown in *Figure 6.8(a)*. The controlling microprocessor sets up the drive signal control logic to perform track seek operations and the DDC to perform disk operations.

In high performance single bus systems, it may be desirable to use a dedicated microcontroller to control the DDC and to generate the drive control signals instead of involving the main system microprocessor in such tasks. *Figure 6.9* shows the disk controller design in a 32-bit single bus system. Here an HPC or other single chip microcontroller is used to program the DDC for various disk operations and to generate drive control signals compatible with the desired interface. As mentioned above, the DDC is typically configured in single channel DMA mode when used in a single bus system environment. There is no restriction on using the DDC in dual channel DMA mode except total address capability. *Figure 6.10* shows a single bus system with the DDC in dual channel DMA mode, transferring data between the FIFO and 64k-pages of 16 MBytes system memory. External latches were added to extend the address range to 24 bits. To use them the host μP would load the most significant 8 bits with the 64k page to transfer data to/from. In this design the local and remote DMA channels are restricted to operating in the same page. To overcome this an additional latch and some logic gating the LACK and RACK signals could be used to enable operation of each channel in different pages.



TL/F/8663–93

FIGURE 6.7 (a). A Typical External DMA-DDC Interface



TL/F/8663–94

FIGURE 6.7 (b). DDC FIFO Read/Write by External DMA

DUAL BUS SYSTEM

Systems usually have two or more buses associated to them. One set of buses for local communication and another main system bus for communication with the host. These systems contain sub-systems (or modules) of the main system (the host). The purpose of the local bus architecture is to allow peripheral processors to control specific tasks, offloading the host CPU. Thus the local bus has in addition to peripherals, such as the DDC, a local microprocessor, memory and an interface to the main system. All the sub-systems access the main system through the I/O channels connected to the main system bus (such as MULTIBUS®, VME etc.).

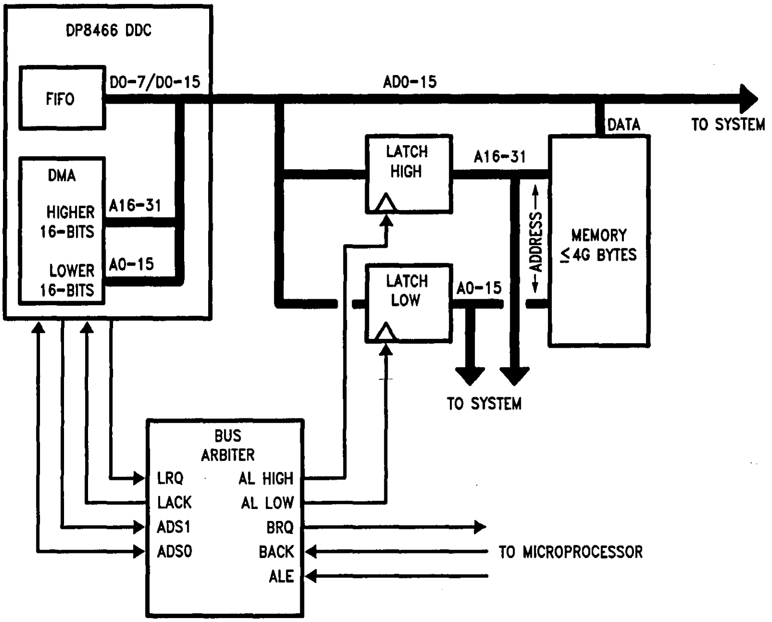
The DDC easily fits into a dual bus system when configured in the dual channel DMA mode. The local DMA channel communicates with local memory and the remote DMA channel is used to transfer data between local memory and a system I/O port. The main system's DMA controller then takes the data to/from the I/O port from/to main memory. The DDC in a typical dual bus system is shown in *Figure 6.8(b)*. The local microprocessor receives commands from the host using DDC's remote channel and then sets up the drive control logic and the DDC for different disk operations. The arbitration block arbitrates the bus between the DDC (while in bus master mode) and the local microprocessor. *Figure 6.11* shows a dual bus system with local microprocessor using 48 kbytes of local memory space as ROM and RAM, and allowing only 16 kbytes of local buffer.

Figure 6.11 shows a possible local bus implementation. Here a local CPU such as the NSC800, 32008, 80188, 64180, etc. controls both the DDC and the disk control port. The local CPU is also in charge of receiving commands from the main system, caching disk sectors, and performing any logic to physical sector address translations.

In this design we have optionally segmented a special 16k for sector data buffers. This is not necessary, but does ensure the DDC will not access the local CPU's data memory inadvertently (this could be done in software as well). The HC646 and some read/write logic form the pass thru port for the commands/data from the main system. The DMA external controller, which may reside on the CPU (80188, 64180) or in the main system, controls transfers from main memory to/from the port. The DDC's remote DMA controls data transfers between local memory and the pass thru port.

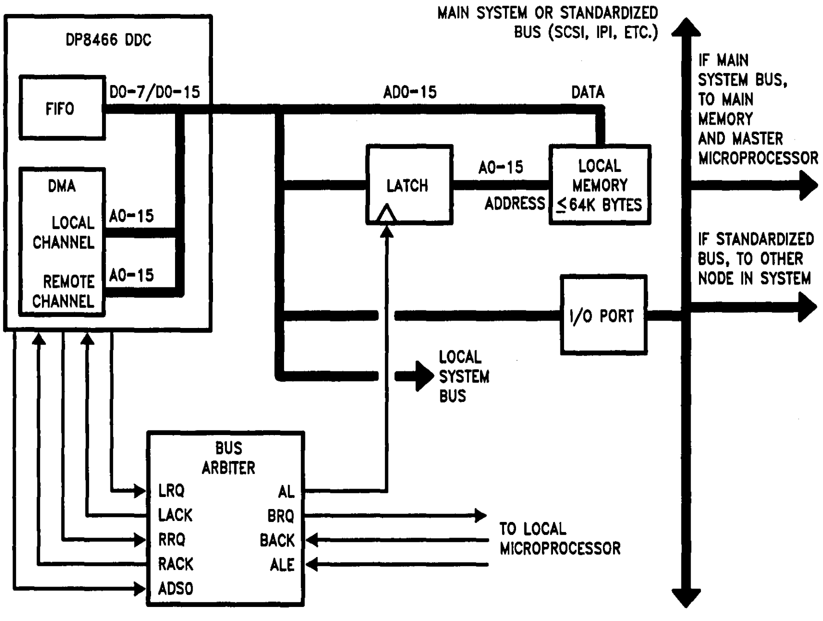
INTELLIGENT PERIPHERAL BUS

A special case of the Dual Bus architecture are intelligent peripheral buses, such as SCSI and IPI. In this case the local bus is the intelligent disk drives bus, and the second bus is the SCSI or IPI bus which will connect to a main system through a host adapter. The dual channel DMA mode of the DDC is ideally suited to this application. The local DMA channel can manage transfers between the DDC's FIFO and the drives buffer RAM, while the remote DMA is passing information to/from the SCSI or IPI bus. The design of the system interface is very similar to *Figure 6.11*, except that the read/write interface logic block and the 'HC646 is replaced by a SCSI or IPI interface, and the systems remote DMA controller is located at the system end of the bus and not on the drive. With the DDC's two channel capability eliminates the need for an external DMA controller on the SCSI or IPI drive.



TL/F/8663-95

FIGURE 6.8(a). The DDC in a Single Bus System



TL/F/8663-96

FIGURE 6.8(b). The DDC in a Dual Bus System

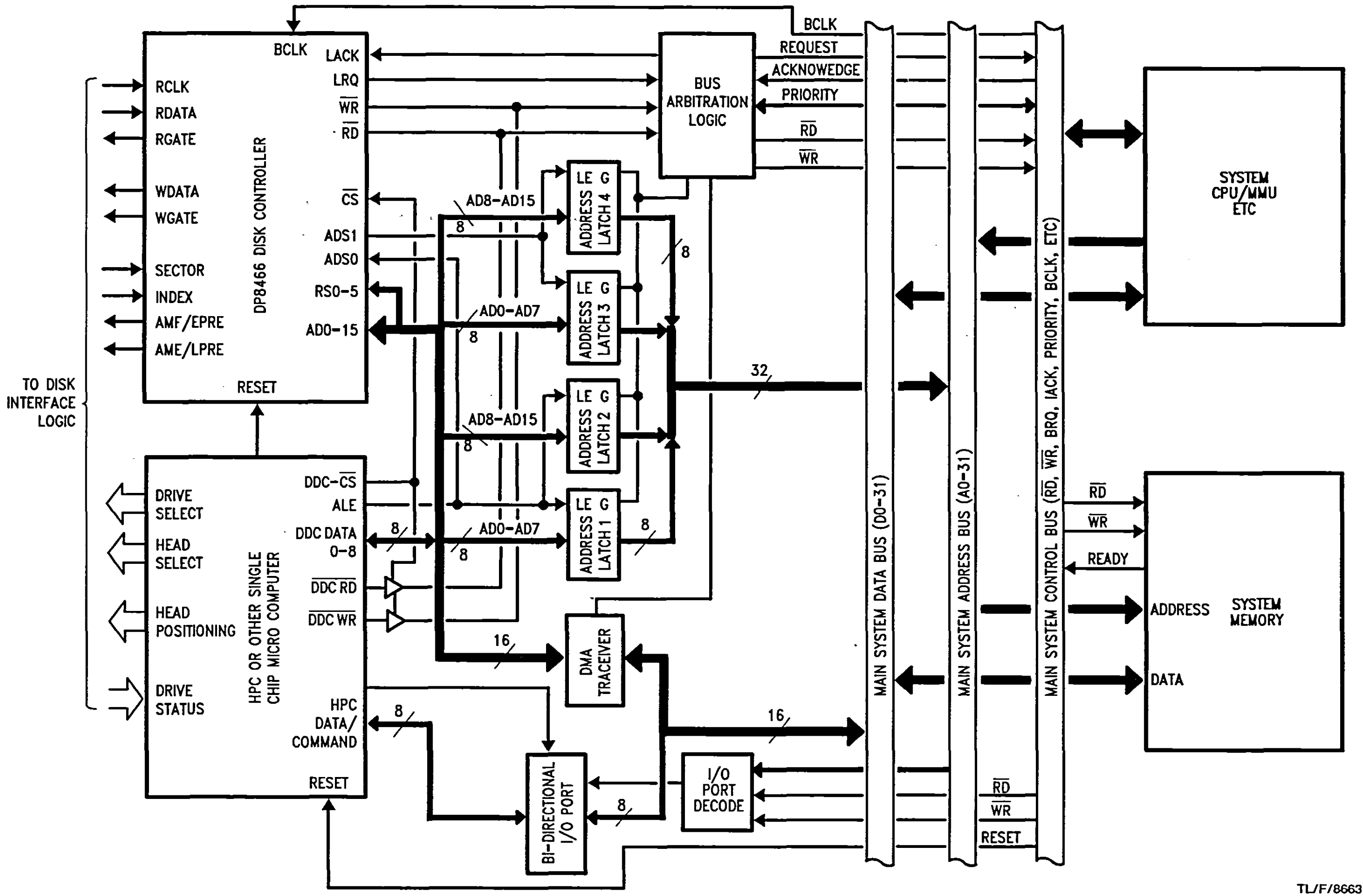
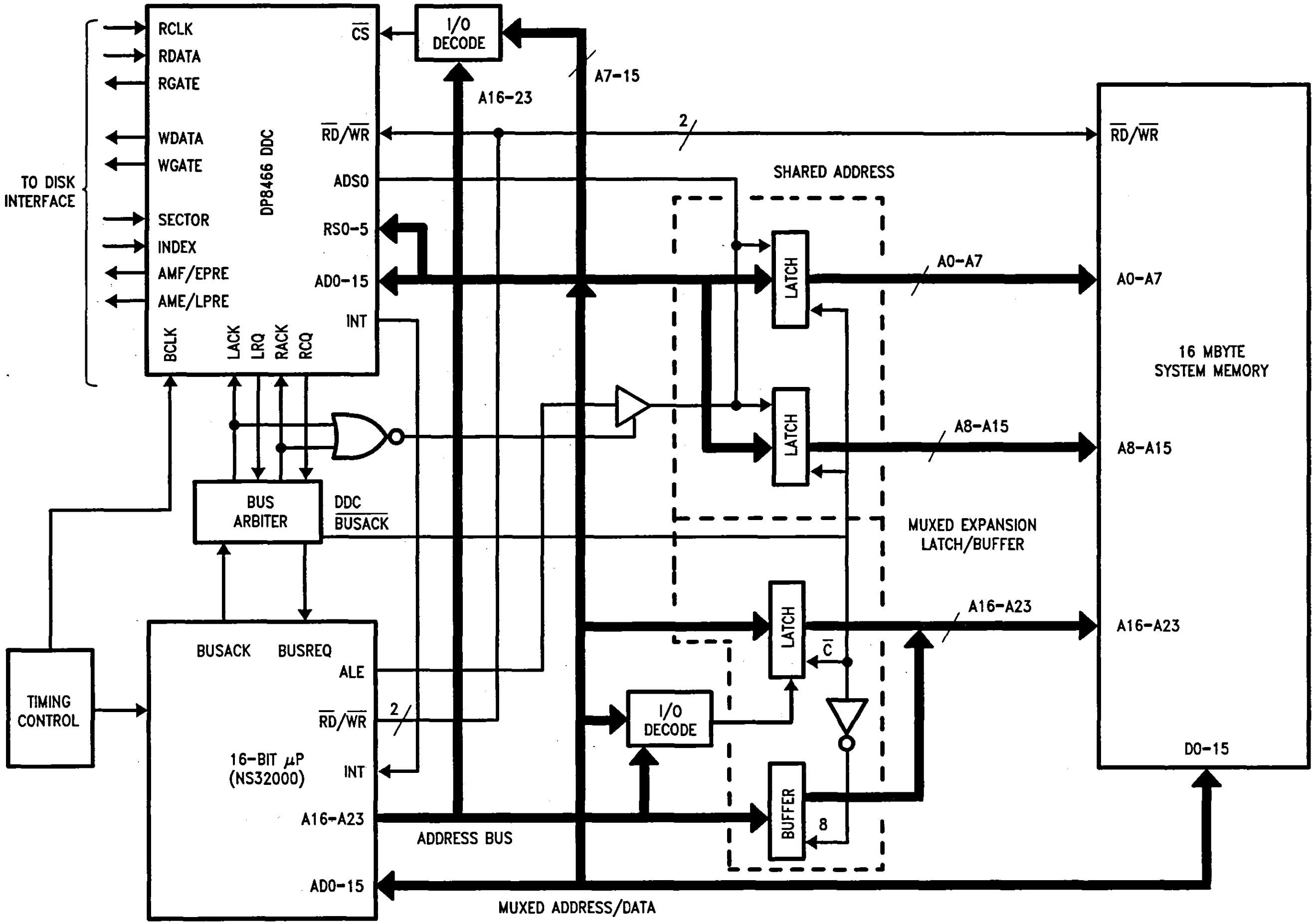
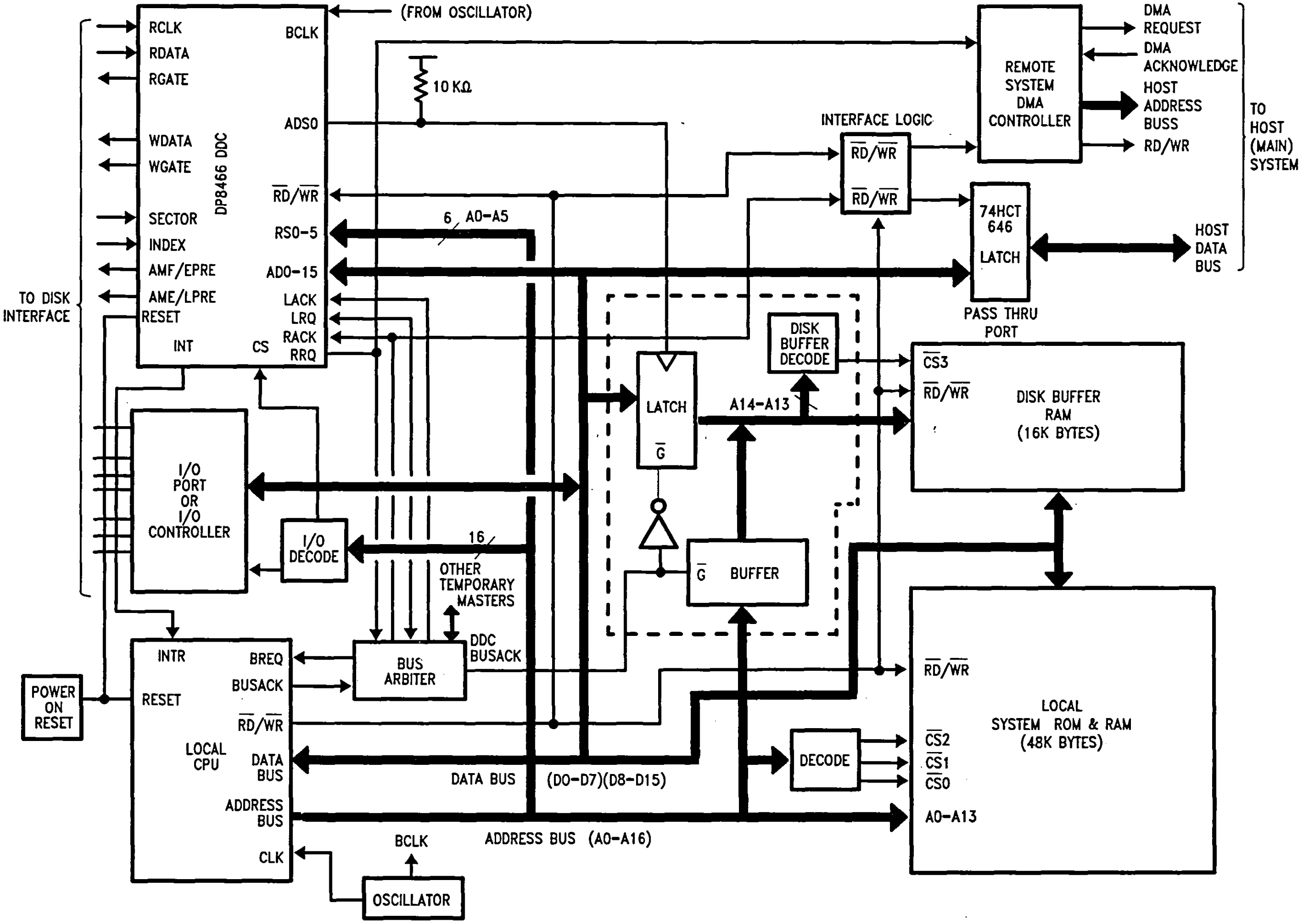


FIGURE 6.9. Conceptual Design of DP8466 Using Single Channel DMA and Controlled by a Microcontroller That Controls DDC and Drive Control Signals



TL/F/8663-98

FIGURE 6.10. Simplified design showing DP8466 with expanded dual direct memory addressing and 16-Bit multiplexed address/data CPU. Lower address latches are shared while the upper address lines are multiplexed.



TL/F/8663-99

FIGURE 6.11. Simplified design of DP8460 with a local 8-bit μP. Latches or microcontroller for disk control signals and limited disk buffer is included.

6.2 DISK SIDE INTERFACE

The disk side hardware connections/orientation is a function of the disk interface standard used, the encoding/decoding scheme and the local intelligence used for disk control signals. The Disk System Controller essentially consists of a data separator (DP8465), the Disk Data Controller (DP8466) and some local microprocessor for disk control like the NSC800 or a microcontroller like the HPC. The Disk Pulse Detector (DP8464) is situated on the drive for all the interface standards, while the data separator is on the drive for the ESDI and SMD interfaces. *Figure 1.9(a)* in chapter 1 also shows the interface points for the various standards. If MFM encoding is used it can be programmed to be part of the DP8466, details can be found in chapter 7; whereas in case of 2,7 RLL code, the DP8463 (2,7 ENDEC) could be used in conjunction with the DP8462 (2,7 data synchronizer).

6.2.1 Generalized Disk Interface with DP846X Chip Set

The DP8464 pulse detector receives signals from the disk's read amplifier, and converts these signals to a digital pulse train. The DP8465 Data Separator receives digital pulses from a pulse detector circuit (such as the DP8464). After locking on to the frequency of these input pulses, the DP8465 separates them into synchronized data and clock signals. If the input pulses are MFM encoded data, the data is made available as decoded NRZ data to be deserialized directly by the disk data controller DP8466. If the RLL code is used, the synchronized data output is available to allow external circuitry to perform the data decoding function. All the digital input and output signals are TTL compatible. The raw MFM from the pulse detector in the drive is connected to the ENCODED DATA input of the DP8465. The DELAY DISABLE input determines whether attempting lock-on will begin immediately after READ GATE is set or after two bytes. Typically in a hard sectored drive, READ GATE is set active as the sector pulse appears, meaning a new sector is about to pass under the head. Normally the preamble pattern does not begin immediately, because gap bytes from the preceding sector usually extend just beyond the sector pulse. Allowing two bytes to pass after the sector pulse helps ensure that the PLL will begin locking on to the preamble, and will not be chasing non-symmetrical gap bits. Attempting to lock-on to a fixed preamble pattern speeds up lock-on, and after another two bytes the PLL will nominally have locked-on. Thus DELAY DISABLE should be set low for this kind of disk drive. For soft sectored drives, the controller normally will not wait for the index pulse before it attempts lock-on, so that READ GATE may go active at any time. Chances are the head will not be over a preamble field and therefore there is no need to wait two bytes before attempting lock-on. DELAY DISABLE can therefore be set high. If a non preamble field is passing by as READ GATE goes active. The DP8465 will not indicate lock, and so no data decoding will occur nor will MISSING CLOCK DETECTED go active. Normally, if lock-on has not been achieved after a certain time limit, the controller should de-activate READ GATE and then try again. For MFM encoded disk drives, the LOCK DETECTED output will be connected back to the SET PLL LOCK input. As the PLL achieves lock-on, the DP8465 will automatically switch to the slower tracking rate and decoded data will appear at the NRZ READ DATA output. Also the READ CLOCK output will switch from half the 2f clock frequency to the disk data rate frequency. If a delay is required before the changeover occurs, a time delay may be inserted between the two pins. The ZEROES/ONES PREAMBLE input selects which preamble the chip is to lock-on to. *Figure 6.12* gives schematics of the data separator in a disk system—when in the controller and when on the drive. For more specific details on the DP8465 refer chapter 3.

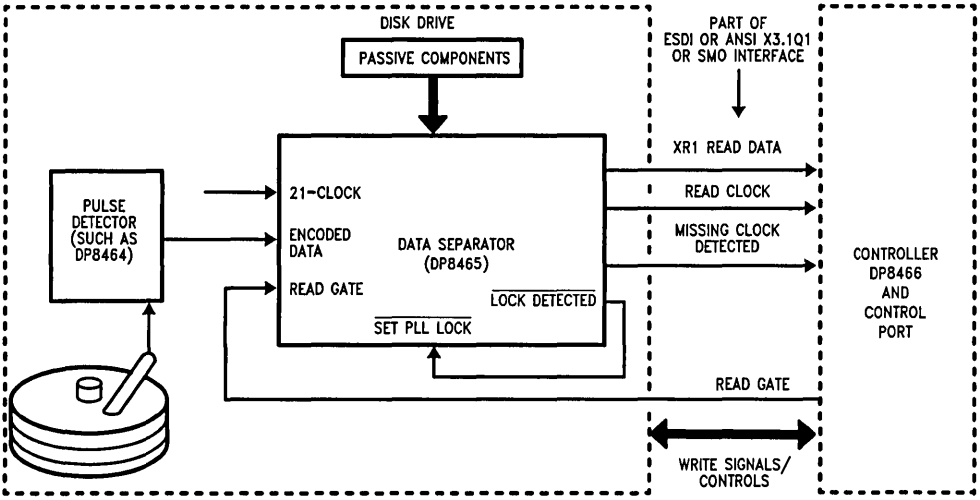
If the drive uses the RRL code such as "2,7", instead of MFM, the PLL function of the DP4862 may be used in conjunction with the 2,7 ENDEC (DP8463), as shown in *Figure 6.13*. The DP8463 performs encoding of NRZ data to RLL encoded data, and RLL encoded data back to NRZ data. It uses the SYNCHRONIZED DATA output of the DP8462 along with VCO CLOCK to lock-on to the preamble and then decode data. For more specific details on the DP8463 refer to it's data sheet.

The most important component in the disk side is the physical disk interface itself. We shall discuss the interface details for some major interface standards viz., STxxx, ESDI etc. Higher level interfaces, like SCSI, incorporate the whole controller board on the drive and interface with the host through a host adapter on to the system bus. However the DDC as such interfaces to the host microprocessor for commands and status.

6.2.2 Interfacing the DDC to the ST506/ST412HP Standard

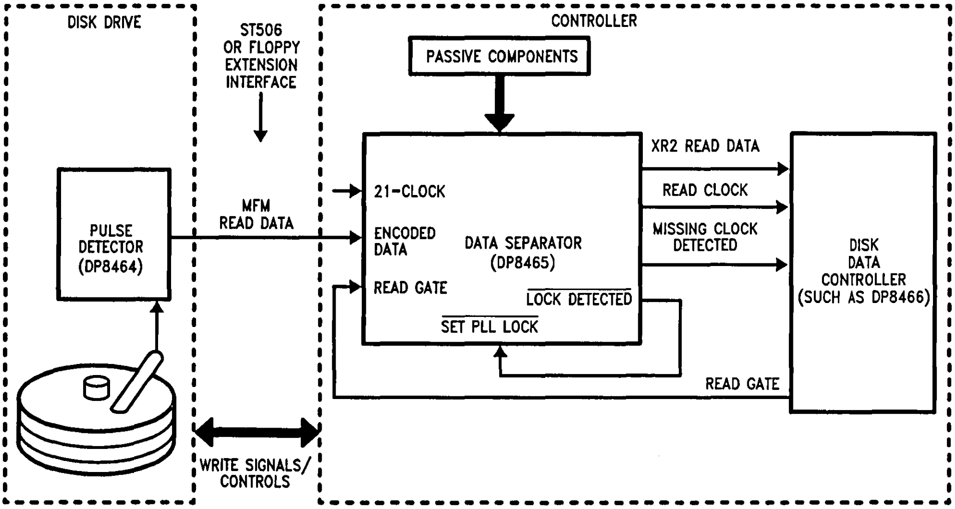
The schematic in *Figure 6.14* shows the interfacing of the DDC to a ST506/ST412HP disk interface. The ST506/ST412HP interface standard has a control cable which is daisy chained to the drives (assuming a multiple drive system). On selecting the drive all the signals on the control cable are associated with the drive selected. The data cable is radial in nature and is multiplexed at the controller using the drive select output on the data cable. The ST506/ST412HP interface standard supports MFM encoding and soft-sectored formatting only. Since the DDC does not take care of the disk control signals, this is done using some local intelligence, indicated as the Disk Signals Controller block, DSC. The drivers are open collecter drivers as per the requirements of the interface standard. The DDC requires NRZ data, hence the MFM encoded data from the drive is sent to the data separator, which synchronizes the data and decodes it to NRZ. The missing clock detect output of the data separator is used to trigger the address mark found (AMF) input of the DDC. While writing data to the disk, the DDC provides MFM encoded data, and the necessary precompensation is provided using an external delay line in conjunction with the early and late precompensation outputs (EPRE/LPRE) of the DDC. The DDC and the DSC could interface to the system bus of the host system or could interface to the host system through an intelligent interface like SCSI or IPI, as mentioned in the previous section.

The basic operation of the interface is as follows. The DSC first selects the drive select lines. Then the head selected by the head select lines is positioned over the desired track by issuing step pulses in conjunction with the direction line. Once the head is positioned, indicated by the seek complete line, the DDC is ready to initiate a read/write operation. This interface's read path is through the data separator and hence a lot depends on the selection of the data separator. The data separator then feeds the controller. The write path consists of write data out of the DDC going to a precompensation block. The output of the precompensation block goes over the ST506 interface to the drive.



TL/F/8663-A0

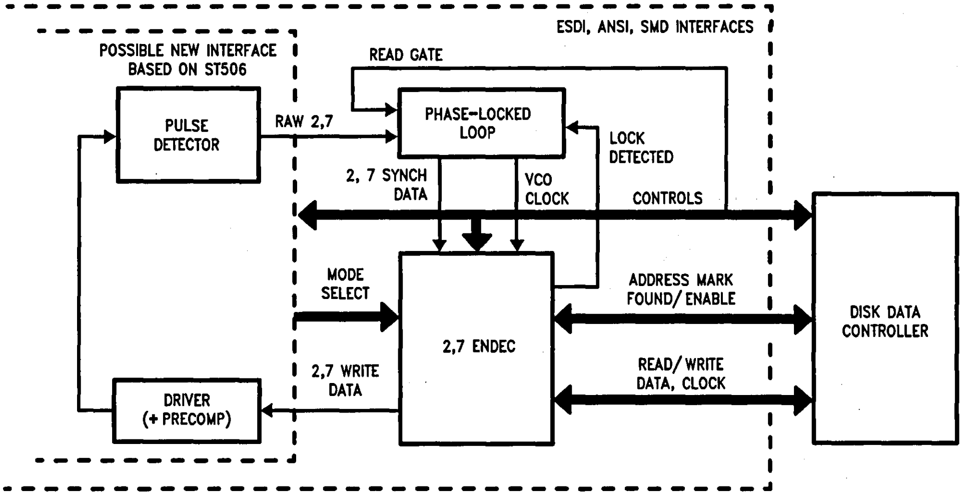
FIGURE 6.12 (a). Data Separator Residing in the Controller



TL/F/8663-A1

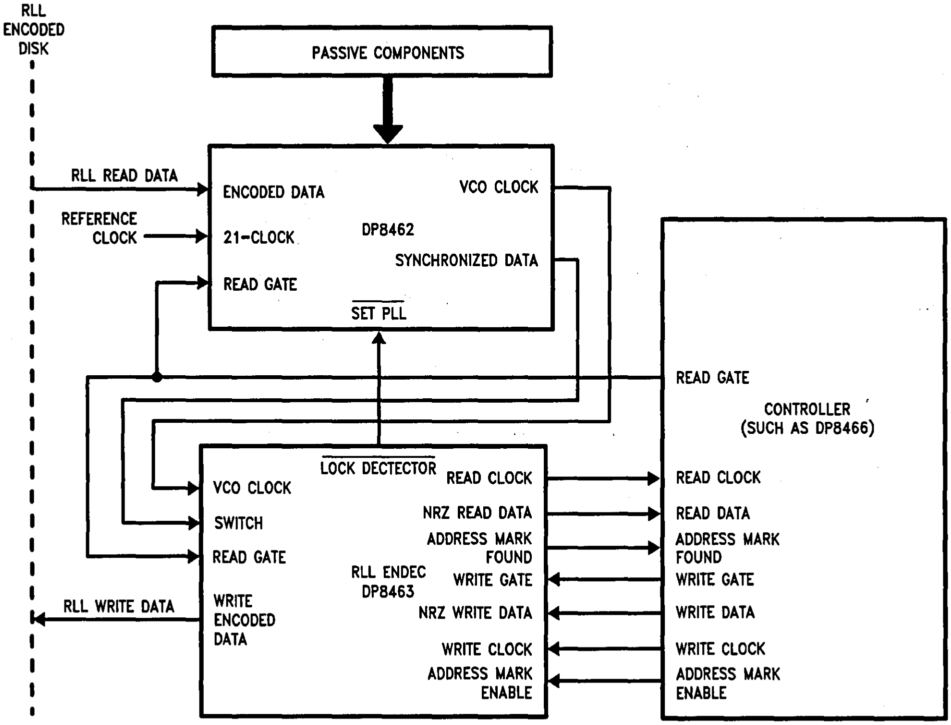
FIGURE 6.12 (b). Data Separator Residing in the Disk Drive

Interfacing the DP8463 2,7 ENDEC in a Disk System



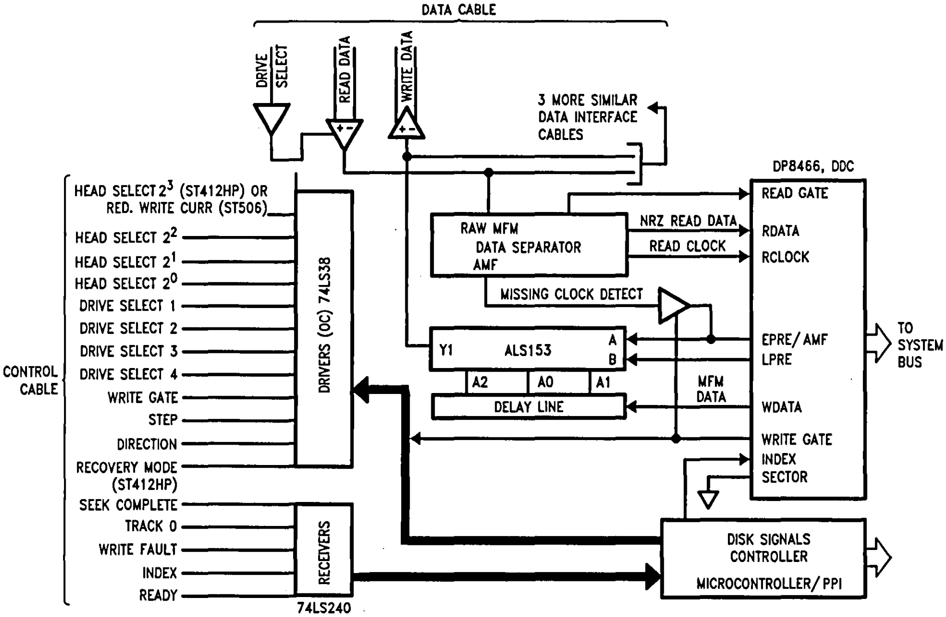
TL/F/8663-A2

FIGURE 6.13. DDC and DP8463-2,7 ENDEC (a) Generalized Disk System Block Diagram



TL/F/8663-A3

FIGURE 6.13. DDC and DP8463-2,7 ENDEC (b) Specific DP846X Solution



TL/F/8663-A4

FIGURE 6.14. Disk Data and Control Paths (ST506/ST412HP)

The necessary sequence of events (with associated timing restrictions) for proper read/write operation of the STxxx drive are shown in *Figure 6.15*. The DDC specifications are in compliance with the above requirements. The DDC has an on board encoder for MFM encoding and provides the precompensation outputs EPRE/LPRE, which are used by some external logic to generate the delays, as shown in *Figure 6.16(a)* and *(b)*

WRITE DATA PATH—PRECOMPENSATION

This consists of a differential pair that defines the transitions to be written on the track. The transition of the +MFM WRITE DATA line going more positive than the −MFM WRITE DATA line will cause a flux reversal on the track provided WRITE GATE is active. To ensure data integrity at the error rate specified for the interface, the write data presented by the DDC must be precompensated on the inner tracks. The optimum amount of precompensation is drive dependent, but usually is around 12 ns for both early and late written bits. In the DP8466 precompensation will be indicated on the EPRE and LPRE pins. Precompensation is issued for the middle bit of a 5-bit field. In the DDC, early and late precompensation will be enacted for all the combinations as shown below. All other patterns will not require precompensation. The center bit column is the present bit being output. The left two bit column is the bits previously shifted out, and the right two bit column is the two bits that will be shifted out next. In the following table a "bit" is a clock or data bit.

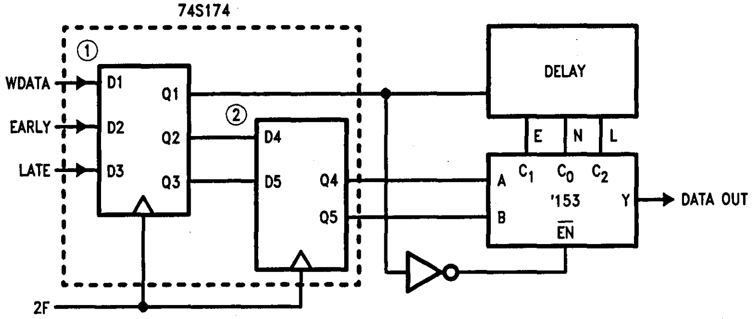
EPRE Patterns	LPRE Patterns
00 1 10	00 1 10
00 0 11	00 1 11
01 1 00	10 0 00
01 1 01	10 0 01
11 1 00	10 1 10
11 1 01	10 1 11

READ DATA PATH—READ GATE

For the read data path the data separator is the critical block. Its design was discussed in chapter 3. In addition how the controller cycles the read gate will affect performance. In case of conventional soft sectored drives the Read Gate cycling by the DDC is different for different conditions. In the case when the address mark is not detected after lock has occurred (abort address mark function—internal to the chip), the DDC deasserts Read Gate 19 RCLKS after getting a non-zero bit on the Read Data line, where it has been receiving an all zeroes data. It will then reassert the Read Gate 17.5 RCLKS later. In the situation where there is a sync failure, Read Gate is deasserted 10 RCLKS after the failed sync word, and reasserts it 17.5 RCLKS later. In case of a Header failure or a CRC failure, Read Gate is deasserted 2 RCLKS after the last check byte and is reasserted 25.5 RCLKS later. *Figure 6.15(c)* shows the Read Gate timing details.

HANDLING THE READ GATE IN SOFT SECTORED ST506 DRIVES

When reading soft-sectored drives, it is difficult to predict when Read Gate will be asserted. Data patterns can appear as preamble and the PLL will lock to these patterns. The controller is able to determine that the field is not a preamble by the address mark signal not being asserted because no missing clock violation was detected and deasserts Read Gate. However, Read Gate might be asserted over a write splice, which may result in the Data Separator failing to lock properly. It is usually up to the data separator design to ensure proper lock. The DDC does not implement a read gate on/off cycling algorithm. If the data separator can be thrown out of lock, the data separator should incorporate this algorithm. If not, this may lead to failure when accessing a sector in certain soft sectored drives. This is only a problem for drives using the conventional soft sectored format



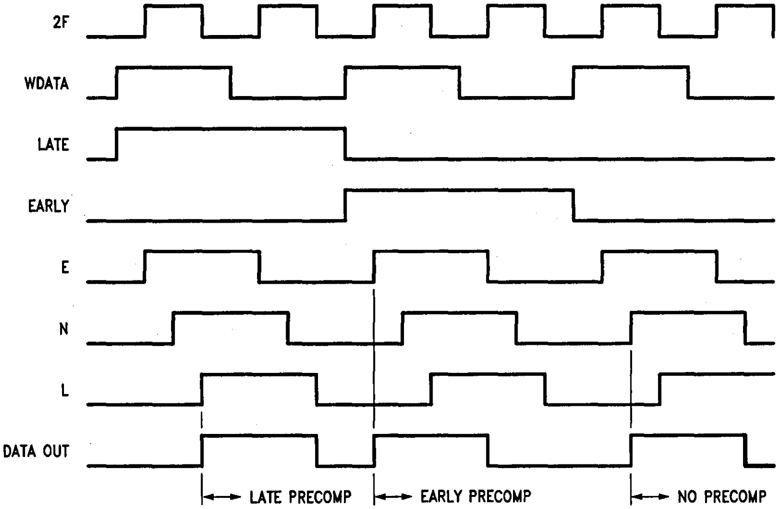
TL/F/8663-I2

FIGURE 6.16 (a). Precompensation Circuitry (MFM Encoding)

Note: Latch ② essentially ensures that the mux is enabled before the early and late signals arrive.

The early, normal, and late compensated data are only relative to each other.

Precomp time is drive dependent, usually 12 ns.



TL/F/8663-I3

FIGURE 6.16 (b). Timing

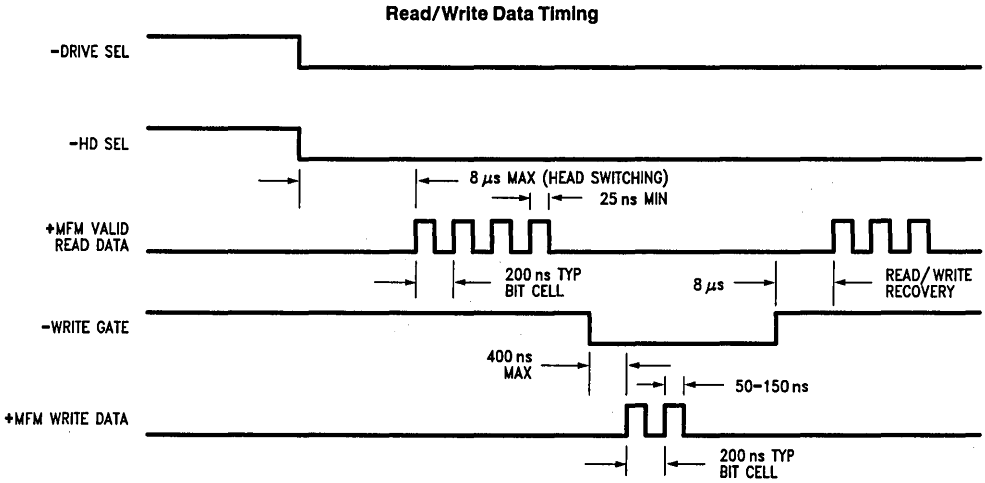
(preamble, address mark, sync for both the ID and DATA segment), which is typical of lower performance, low capacity drives, operating at data rates of 5Mbits/sec and below, like the STxxx family. Newer standards replace the A1 (Hex) missing clock address mark currently used in soft sectored drives with an address mark which is a gap of no transitions at the beginning of the sector. Such a format is referred to as "start with address mark" format. The SMD and ESDI interfaces currently specify this type of operation and should be typical of those drives operating at 10 Mbits/s or greater.

Operation of the DDC Read Gate In Soft Sectored Drives

In a soft sectored drive, the DDC asserts Read Gate at an arbitrary point over the track. This initiates a lock sequence on the Data Separator and the DDC begins to monitor the pattern entering its shift register. To avoid issuing garbage to the controller while it is locking up, the data separator usually will issue zeroes to the controller as it is looking for the first bit of non-zero data signifying lock. If using the 8465, when SET PLL LOCK is asserted, the Data Separator will begin to issue decoded data. In most applications, the LOCK DETECT output is connected to the SET PLL LOCK input, so that when the Data Separator has locked, it will issue data. However, if the Data Separator were to fail to lock and in some manner gets hung up where it will never be able to detect a preamble pattern, the system will be deadlocked since the DDC still is being given all zeroes from the Data Separator. The read operation will be aborted only after 2 revolutions. This type of failure can occur if Read Gate is asserted over write splice areas. The problem is really twofold, the Data separator is not issuing non-zero data to the DDC so that it can deassert Read Gate (the DDC thinks that the Data separator is still looking for the preamble) and the DDC is not placing a timeout on a response from the Data Separator.

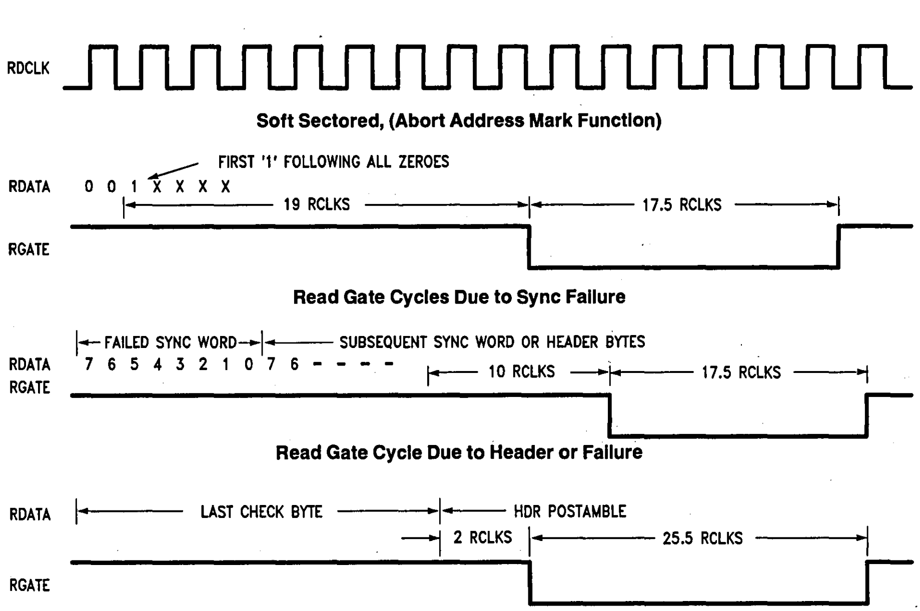
Suggested External Logic to Timeout the Read Gate

If necessary a simple external circuit can perform the time out function. When Read Gate is asserted a counter is started. After 3 to 4 bytes (allowing time for the Data Separator to acquire lock), the SET PLL Lock on the Data Separator is driven low. This enables data to be sent from the Data Separator to the DDC. The DDC monitors the data and if a '1' propagates through the deserializer without matching the first pattern programmed for the format, Read Gate is deasserted. The DDC deasserts Read Gate for approximately 19 bit times, *Figure 6.17(a)*, hence the Data separator should be able to resynchronize to the 2f clock within this time. Hence even in the worst case, the amount of preamble lost by this scheme is 6–7 bytes, (1 byte for the DDC to deassert Read Gate after receiving a '1', pattern not matched, 2 bytes of Read Gate deasserted time and 3–4 bytes of timeout, essentially the lock time of the Data Separator). The suggested circuit is shown in *Figure 6.17(b)*. This circuitry could also be implemented in a PAL.



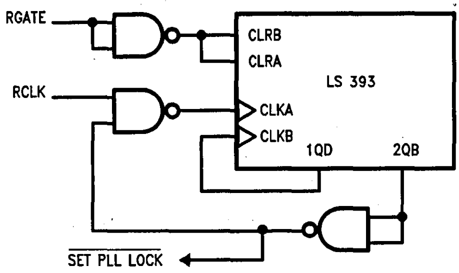
TL/F/8663-A5

FIGURE 6.15. ST506 Read/Write Data Timing


RDCLK
Soft Sectored, (Abort Address Mark Function)
FIRST '1' FOLLOWING ALL ZEROES
RDATA
0 0 1 X X X X
19 RCLKS
17.5 RCLKS
RGATE
Read Gate Cycles Due to Sync Failure
FAILED SYNC WORD
SUBSEQUENT SYNC WORD OR HEADER BYTES
RDATA
7 6 5 4 3 2 1 0 7 6 - - - -
RGATE
10 RCLKS
17.5 RCLKS
Read Gate Cycle Due to Header or Failure
RDATA
LAST CHECK BYTE
HDR POSTAMBLE
2 RCLKS
25.5 RCLKS
RGATE

TL/F/8663-A7

FIGURE 6.17 (a). Read Gate Timing (Cycling)



Counter timer using dual 4 bit binary counters cascaded. Count is stopped by decoding bit 2QB which is reached after 32 RCLKS, counter is reset when RGATE is deasserted.

TL/F/8663-A8

FIGURE 6.17 (b). Read Gate Timeout Circuit

6.2.3 Interfacing the DDC to the ESDI Standard (Serial Mode)

The Enhanced Small Device Interface (ESDI), is a low cost, high performance interface suitable for the smaller memory devices currently on the market. It supports higher data transfer rate, 10–15 Mbits/s and provides for additional performance features desirable in higher performance systems. Two modes of implementation are possible: Serial mode of operation, utilizing NRZ data transfer along with serial commands and serial configuration/status reporting across the command cable (J1). Step mode implementation utilizes the same NRZ data transfer; however, the STEP and DIRECTION lines are used to cause actuator motion. This is similar to the ST506 type of interface as far as interfacing is concerned.

The ESDI interface consists of a 34-pin control cable and a 20-pin data cable. The control cable is attached in a daisy chain configuration while the data cable must be attached in a radial configuration. Hence all the control signals are associated with the drive selected. All the control lines are digital in nature (open collector TTL) and either provide signals to the drive (input) or signals to the host (output). The control signals and the serial commands are handled by the Disk Signals Controller, which is some local intelligence, as shown in *Figure 6.18.*

The ESDI interface can support both Hard and Soft sector drives. The interfacing of the, DDC is slightly different for hard versus soft sectored drives. *Figure 6.18* shows the interfacing of the DDC to a Hard sectored ESDI drive in the serial mode. In this configuration the sector pulse from the drive is used to identify the start of a sector. All lines associated with the transfer of data between the drive and the host system are differential in nature and may not be multiplexed. These lines are provided at the J2/P2 connectors on all drives. Four pairs of balanced signals are used for the transfer of data and clock.

NRZ Write Data: This defines the data to be written on the disk and is clocked by the WRITE CLOCK signal. This definition is compatible with that of the DDC. The Write data and Write Clock timings are shown in *Figure 6.19*.

NRZ Read Data: The data recovered by reading previously written information is transmitted to the host system via the differential pair of NRZ Read Data lines. This data is clocked by the READ CLOCK signal. These lines must be held at a zero level until PLL synch has been obtained and data is valid. This is compatible with what the DDC expects. One note of caution, when the PLL is locking on to the preamble, the DDC is receiving 0's and is looking for a non-zero synch byte. If the PLL goes off to harmonic lock or never locks and continuously outputs NRZ 0's, the DDC will assume it is forever looking at the preamble and will finally abort the command after two index pulses, flagging a sector not found error. It is up to the drive designer to prevent the PLL from going into harmonic lock. The DP8465, DP8461 and DP8462 are designed to remove harmonic lock. *Figure 6.19* shows the timing requirements for the read data.

Read/Reference Clock: *Figure 6.19* depicts the necessary sequence of events (with associated timing restrictions) for proper read/write operation of the ESDI drive. The Reference Clock signal from the drive will determine the data transfer rate. The transitions from Reference Clock to Read Clock must be performed without glitches. Read Clock and Read Data are valid within the number of PLL sync field bytes specified by the drive configuration after read enable and a PLL sync field is encountered. The interface Read/Reference Clock line may contain no transitions for up to two Reference Clock periods for transitions between reference and read clocks. The transition period will also be one-half of a Reference Clock period minimum with no shortened pulse widths. This is compatible with the specifications of the DDC, which has setup and hold times typically of the order of 15 ns. Reference Clock is valid when Read Gate is inactive while Read Clock is valid when Read Gate is active & PLL sync has been established.

Write Clock: Write Clock is provided by the DDC and must be at the bit data rate. This clock frequency is dictated by the Read/Reference clock during write operations. The DDC complies with the ESDI standard which requires the Write Clock to be active when Write Gate is active. See *Figure 6.19* for timing.

READ AND WRITE TIMING

Write Gate

The active state of this signal enables write data to be written on the disk. The low to high transition of this signal often creates a write splice and then initiates the writing of the data PLL sync field by the drive. The timing restrictions on Write Gate in the ESDI specification are as follows:

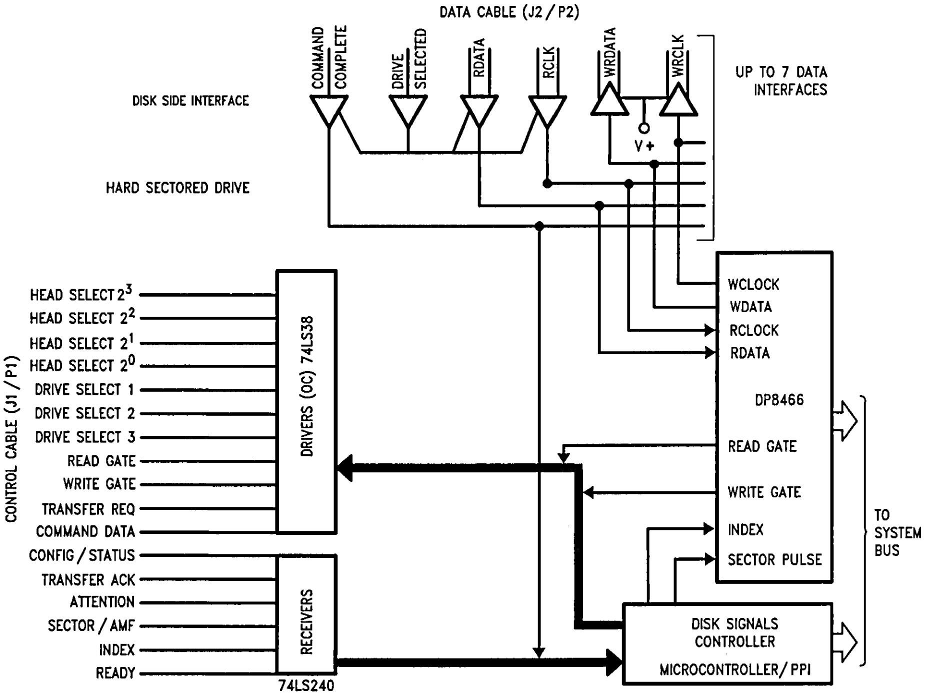
1) When formatting, Write Gate should be deactivated for 2 bit times minimum between address area and the data area to identify to the drive the beginning of the data PLL sync field.

2) It should be asserted at least two and a half reference clock periods after Write Clock.

3) The time lapse from deactivating Read Gate to activating Write Gate should be a minimum of 5 reference/read clock periods.

4) To account for data-encoding delays, Write Gate must be held on for at least two byte times after the last bit of information to be recorded.

5) It should be deactivated at least 1 μs before a head change and may not be activated until 15 μs after a head change.



TL/F/8663-A9

FIGURE 6.18. Data and Control Paths (ESDI-Serial Interface) Hard Sectored Drive

Read Gate

The active state of this signal, or low level, enables data to be read from the disk. This signal should become active only during a PLO sync field and at least the number of bytes defined by the drive prior to the ID or Data Sync bytes. The timing restrictions on Read Gate for the ESDI specification are as follows:

1) Read Gate must be false when passing over a write splice area. It must be deactivated 1 bit time min. before a Write Splice area and may be enabled 1 bit time min. after a Write Splice area.

2) The time lapse before Read Gate can be activated after deactivating the Write Gate is 10 µs.

The Read/Write Gate timings for format, write and read operations are discussed below with reference to the DDC, demonstrating its compatibility with the ESDI specifications.

Format Sector

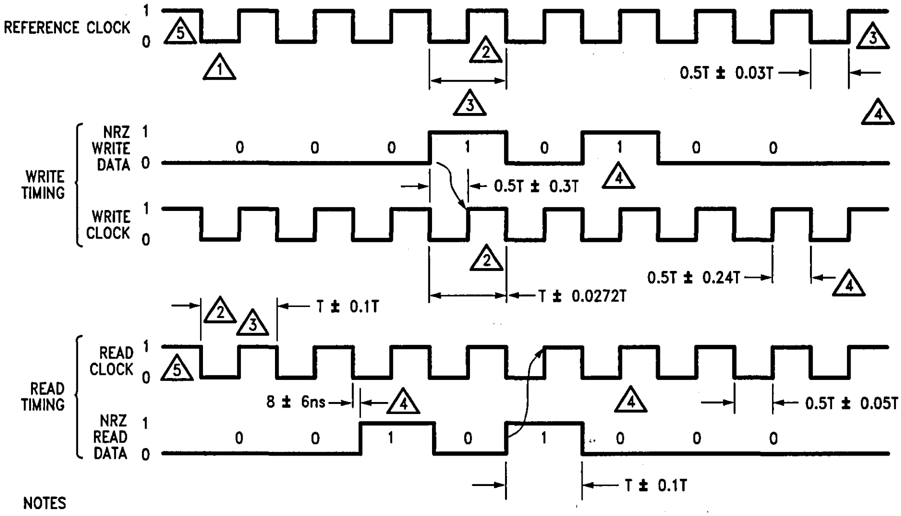
In the case of the DDC, during a format operation, it will do a continuous operation on the whole track. Thus after the index hole is sensed the DDC will assert WRITE GATE within 3.5 bit times. Write Gate will remain asserted until the index hole is sensed again. When each sector pulse is sensed (index for sector 0) the DDC will immediately start writing the various fields. After the data postamble is written the DDC will fill the rest of the sector with the gap until the next sector pulse is sensed. *Figure 6.20* shows the basic timing and the ESDI recommended format.

Write Sector

In case of a compare header-write data operation, the DDC will assert Read Gate within 3.5 Read Clocks from the rising edge of the Index or sector pulse. Read Gate is de-asserted 2 Read Clock periods after the ID check field, (plus a small propagation delay). Read Gate will remain de-asserted for the entire postamble. Due to internal delays, Write Gate is asserted 3 bit times into the data preamble field. Hence this would meet the 5 bit time minimum spec. between the read gate de-assertion and write gate assertion assuming at least one ID postamble byte is programmed. At the end of the write operation, Write Gate is removed 0 bit times after the data postamble. Hence a 3 bit time 'pad' is created between header postamble and data preamble updating data. This pad will contain data preamble as written during format operation. Refer to *Figure 6.20* for basic timing. It should be noted that a write splice of 8 bits is associated with the assertion of Write Gate. Hence if a write header operation is involved, then the read gate should not be asserted in the splice area.

Read Sector

In case of a compare header-read data operation, the DDC will assert Read Gate within 3.5 Read Clock cycles from the rising edge of the Index or sector pulse. Read gate is de-asserted 2 Read Clock cycles after the ID check field. Read Gate is re-asserted 11.5 bit times from the data preamble. This is 8.5 bit times from the point where Write Gate is as-



NOTES

△1 ALL TIMES IN ns MEASURED AT I/O CONNECTOR OF THE DRIVE T IS THE PERIOD OF THE CLOCK SIGNALS AND IS THE INVERSE OF THE REFERENCE OR READ CLOCK FREQUENCY.

△2 SIMILAR PERIOD SYMMETRY SHALL BE IN ± 4ns BETWEEN ANY TWO ADJACENT CYCLES DURING READING OR WRITING.

△3 EXCEPT DURING A HEAD CHANGE OR PLO SYNCHRONIZATION THE CLOCK VARIANCES FOR SPINOLE SPEED AND CIRCUIT TOLERANCES SHALL NOT VARY MORE THAN −5.5% TO +5.0% PHASE RELATIONSHIP BETWEEN REFERENCE CLOCK AND NRZ WRITE DATA OR WRITE CLOCK IS NOT DEFINED.

△4 TIMING APPLICABLE DURING READING OR WRITING.

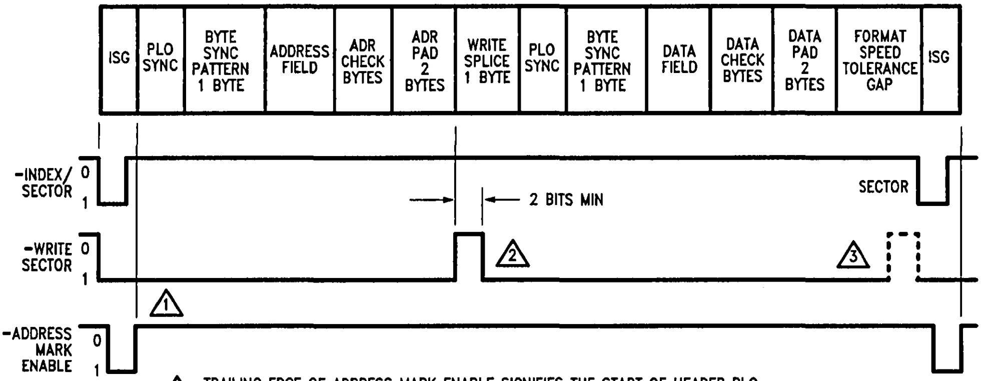
△5 REFERENCE CLOCK IS VALID WHEN READ GATE IS INACTIVE READ CLOCK IS VALID WHEN READ GATE IS ACTIVE AND PLO SYNCHRONIZATION HAS BEEN ESTABLISHED.

TL/F/8663-B0

FIGURE 6.19. NRZ Read/Write Data Timings

serted. This accommodates the 8 bit time write splice generated due to write driver turn on time as required in the ESDI specification. Read Gate is de-asserted 2 bit times into the data postamble. Of particular importance is that the read operation avoid reading the write splice. As can be seen from the write sector discussion, the write splice will occur 3 bits into the preamble, and the read will occur after 11.5 bits. Thus Read Gate is disabled during the actual write splice. When the read and write operations use the same format parameters, Write Gate will cause a splice 8.5 bits before Read Gate is asserted assuming insignificant delay in the read path from the media. However this provides a maximum of 8.5 bit times the write data out could be delayed by the write data encoder, in the drive, prior to being written on the media. *Figure 6.20* gives the basic timing. The user may have to account for additional delays specific to the drive.

Write Gate in ESDI with Index/Sector, AME



1 TRAILING EDGE OF ADDRESS MARK ENABLE SIGNIFIES THE START OF HEADER PLO SYNC FIELD DRIVE WILL NOT WRITE AN ADDRESS MARK ON THE DISK MEDIA.

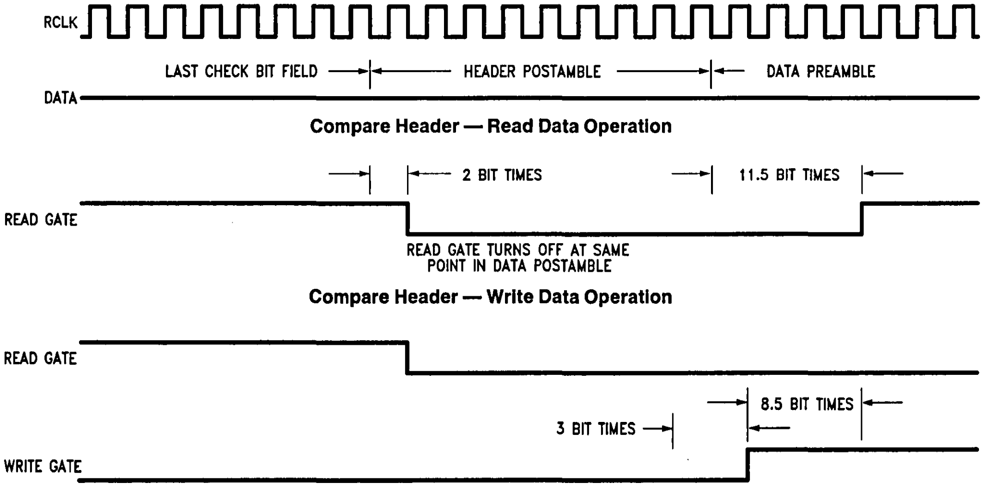
2 TRANSITION REQUIRED ONLY IF THE DISK IS READ AFTER A FORMAT AND PRIOR TO A DATA FIELD WRITE UPDATE.

3 CONTROLLER MUST REINITIALIZE TIMING WITH EACH SECTOR PULSE (NEED NOT DEACTIVATE WRITE GATE).

TL/F/8663-B1

FIGURE 6.20 (a). Read/Write Gate Timing for DDC

Read/Write Gate Timing at End of ID; Beginning of Data Field



TL/F/8663-B2

FIGURE 6.20 (b). Read/Write Gate Timing for DDC

6.2.4 Special Consideration for ESDI Drives

SOFT SECTORED

Interfacing the DDC, to a soft sectored ESDI drive is slightly different. There are no sector pulses demarcating sectors. An Address Mark pattern of no flux transitions is used to identify the start of the sector. The ideal soft sectored format (as supported by the DDC) consists of the Preamble field followed by the Address Mark (AM) and Sync fields. However in the case of ESDI the soft sectored format specification supports an AM at the beginning of the sector followed by the Preamble and Sync fields. This is a special field of no flux transitions which is essentially used to mark the start of the sector. Address Mark detection and generation is done using a handshake protocol between AME and AMF on the drive cable. Since the AMF on the DDC is geared to look for a missing clock violation in the address mark pattern, it cannot be used in the ESDI handshake. Also the DDC generates AME only during the format operation. Hence to achieve successful operation of the ESDI soft sectored drive, the DDC formats the drive as a soft sectored drive and while reading, the AMF signal from the drive is used to generate a sector pulse, as the DDC is configured to operate in the hard sectored mode. It also requires manipulation of the format parameters as shown in *Figure 6.22*. *Figure 6.21* shows the schematic of the interface at a block level.

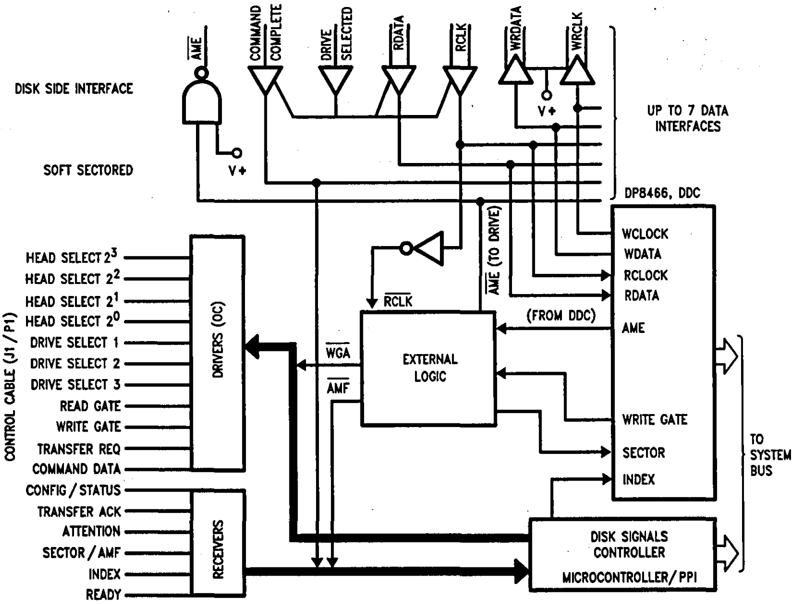
Let us first consider manipulation of the format parameters to achieve proper operation. *Figure 6.22* gives the sequence of events. Consider the first three fields of the format. The DDC views them as Preamble, AM and Sync, *Figure 6.22(a)*. The DDC format is programmed as AM, Preamble and Sync for the first three fields, *Figure 6.22(b)*. If the format operation is initiated with SAM (start on address mark) bit set in the Disk Format register, the recording on the media is shown in *Figure 6.22(c)*. The AME output on the DDC is asserted during the Address Mark (field 1). During a Compare header-read/write operation the first field of the format parameter RAM in the DDC is programmed to be the Preamble. The count for the second field is set to zero, so that it is skipped, while the third field is progammed to be the Sync field. When data is read, the field 1 of no transitions (AM) results in AMF becoming active which generates the sector pulse for the DDC, *Figure 6.22(d)*.

In the ESDI specification, AME (address mark enable) line, when active with Write Gate causes an Address Mark to be written on the media. When AME is active without Write Gate or Read Gate, it causes a search for Address Marks. On detection of the end of Address Mark, AMF (address mark found) responds. The trailing edge of AME with Write Gate true initiates the writing of the header PLO Sync field.

To incorporate the AME/AMF handshake, external logic is required with the DDC. Since the DDC generates AME only during a format operation, the circuit would have to generate AME to the drive during a read operation and incorporate the handshake with AMF from the drive. This AMF is used as a sector pulse input to the DDC. This technique then results in a sector pulse corresponding to each address mark pattern demarcating each sector. However at sector 0 this poses a problem. The index pulse is followed by the post index gap and then the address mark field of sector 0 which would generate a sector pulse. As per ESDI requirements this circuit would have a delay and present the index pulse over the sector 0 pulse and block out the sector 0 pulse to the DDC. The other constraints to be maintained in order to comply with ESDI spec requirements are:

1) AME should be asserted min 100 ns after min write gate is asserted and be deasserted 100 ns before write gate is deasserted.

2) AME can be asserted again at least 10 μs after write gate deassertion.



TL/F/8663-B3

FIGURE 6.21. Data and Control Paths (ESDI-Serial) Soft Sectored Drive

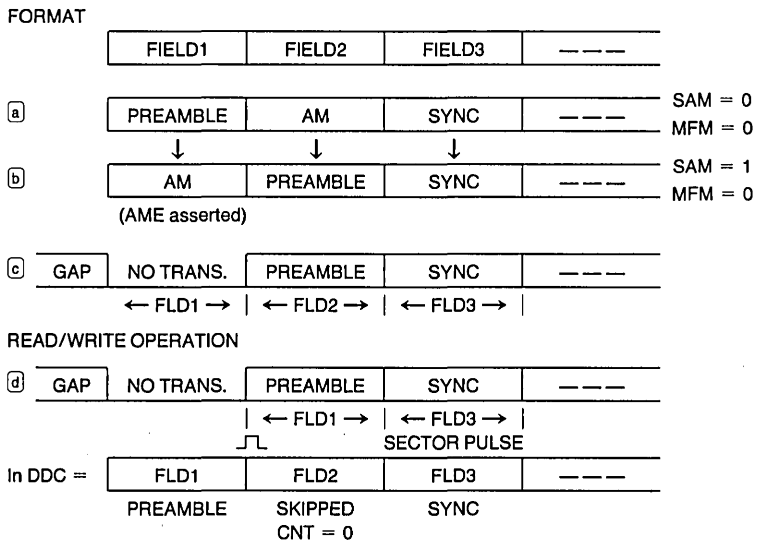


FIGURE 6.22. Manipulation of Format Parameters in DDC for ESDI Soft Sectored Operation

HARD SECTORED ESDI

For an ESDI drive which is hard sectored, the ESDI specification calls for an Inter Sector Gap (ISG) which is to precede and follow the index/sector pulses. The gap is needed to provide the drive with an area for the embedded servo (if used), and gives the controller time to assert read gate. While formatting the drive, the end of the ISG is indicated by the removal of the address mark enable signal (AME). This signal is needed by the drive to indicate the beginning of the PLL field, necessary when the disk encodes the PLL field with a non-standard pattern (as with 2,7 encoding with 3t preambles). The DDC is capable of generating the AME signal with the necessary timing. The DDC needs to be in the hard sectored mode, and have the Start with Address Mark bit (SAM bit of the DISK FORMAT register) enabled. The DDC ID Preamble field now becomes the ISG following the index/sector pulse, and the ID Sync 1 field becomes the PLL preamble field. While not formating, this feature is not needed, and should be disabled.

When the header field of an ESDI drive is read (or compared), the read gate to the drive needs to be delayed until after the ISG. The DDC generates read gate only after receiving a sector or index pulse, so by delaying the sector and index signals to the DDC the read gate will be delayed.

COMBINED SOLUTION

A solution to the above problems can be provided by 1 PAL device and something to provide a delay (possibly another PAL device) *Figure 6.23(b).* The interface solutions can be grouped into two main areas: Address marks and Index/Sector.

The address mark control needs to provide the following:

(1) Direct connection of AME to the drive and AMF to DDC sector input when formating a hard sectored drive.

(2) Delay the leading edge of the AME by the width of post index ISG when formating a soft sectored drive.

(3) Provide AME/AMF handshaking when not writing the disk, and properly change from reading to writing and back with soft sectored drives.

The index/sector control needs to provide the following:

(1) Delay the index and sector pulses from a hard sectored drive when not formating.

(2) Generate index and sector pulses to the DDC from AMF and Index when using a soft sectored drive while not formating.

The Address Mark Machine

The address mark machine consists of a pair of multiplexers which feed the AME input to the disk drive and the AMF input to the DDC. A state machine ensures the proper sequence of events, while a timer provides delay. The address mark machine works in the following way:

When in the soft sectored mode and not formating, it will assert AME to the drive. When AMF is detected, AME is removed until AMF is no longer detected. This uses states 0 and 3 of the address mark state machine, as shown in the diagrams, *Figure 6.23(a).*

When write gate is detected, AME is removed, and write gate to the drive is generated a short time later. When the write operation is ended, write gate is removed from the drive, and AME is enabled a short time later.

In all other modes of operation, the state machine is deactivated and write gate is delayed to the drive by one bit clock time (circuit convenience). The multiplexer continues to provide the drive and DDC with proper signals.

The Index/Sector Machine

The index/sector machine consists of a pair of multiplexers which feed the index and sector inputs of the DDC. While formating, the multiplexers connect the index and sector signals from the drive to the DDC.

When not formating a hard sectored drive, the state machine waits for either a sector or index pulse. When this is detected, the machine waits for a delay and then generates a sector or index pulse. A flip flop, borrowed from the address mark machine, is used to record whether an index or sector pulse should be generated (the address mark machine is disabled when in the hard sectored mode). The delayed index or sector pulse is generated for another delay period of time.

With a soft sectored drive, the state machine waits for either an index or AMF signal. If index is detected, the machine will wait until an AMF is detected. An index pulse will then be sent to the DDC without a sector pulse. If a AMF pulse is detected without index, the state machine will generate a sector pulse.

6.2.5 Interfacing the DDC to the SMD Interface Standard

The Storage Module Device (SMD) interface is a high performance interface, extremely popular with 8″–14″ drives. It provides features similar to the ESDI interface, with some differences. It supports higher data rates from 10 Mbits/s to 24 Mbits/s and utilizes NRZ data transfer along with parallel command and status reporting across the command cable. The SMD interface consists of a 60-pin control (A) cable and a 26-pin data (B) cable. The control cable is attached in a daisy chain configuration while the data cable must be attached in a radial configuration. The control signals are handled by a separate Disk Signals Controller block, which is some local intelligence, as shown in *Figure 6.24*. Only the interfacing of the data path signals are discussed as they are of relevance to the DDC. The control signals could be easily done by a local microprocessor. All lines associated with the transfer of data between the drive and the host system are differential in nature and may not be multiplexed. These lines are provided on the B cable of all drives and are briefly discussed below:

Write Data

This line carries NRZ data, to be written on the disk surface and must be synchronized with Write Clock. This definition is compatible with that of the DDC.

Write Clock

This is a retransmitted clock signal of the servo clock (IF Write Clock) issued by the controller.

Servo Clock

This signal is used by the control unit in the drive to synchronize Write Data with the Clock. Servo Clock may be available during unit ready status except during read operations, or at all times.

Read Clock

This line transmits Read Clock. The Read Data is synchronized with 1F Read Clock. This line may be valid only during a read operation, or may be multiplexed with the Servo Clock signal at other times.

Read Data

This line transmits the recovered data in the form of NRZ data synchronized with 1F Read Clock.

Figure 6.25 shows the basic timing requirements for the above signals for the Fujitsu M2311 micro disk drive hereafter referred to as MDD. In general "SMD" type drives have similar timing requirements.

READ AND WRITE GATE

Read and Write Gate are the two signals which are used to read/write data from/to the specified track/sector. In the SMD interface, they are present on the Bus Out, (bit 1—read gate, bit 0—write gate), when enabled by tag 3. Write Gate enables the write operation and is validated only when Unit Ready, On Cylinder and Seek End are true and Seek Error, Fault, File Protect, Offset are false. If Write Gate is turned on in cases other than the above conditions, fault occurs and writing is inhibited. At this juncture it would be appropriate to mention that there are certain drive dependent constraints which must be taken care of while interfacing to the SMD drives. These are drive dependent, and representative values observed in most of the drives are given below:

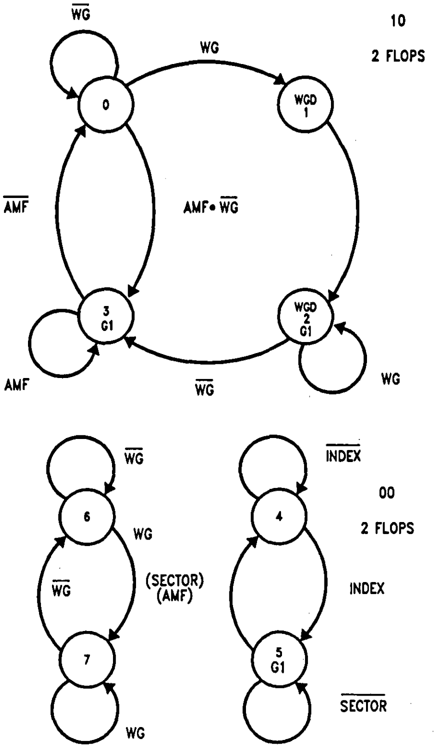
Write circuit turn-on delay: approximately 8 bit times.

Head select transient: A 5 μs delay minimum must be provided between head select and initiating read gate. Normally this is provided by selecting the appropriate length for the gap after the data postamble and the gap after the index/sector pulse.

Read-after-Write transients: A minimum delay of 10 μs must be provided between the trailing edge of write gate and the leading edge of read gate. This could also be done by adjusting the lengths of the gap after the data postamble and the gap after the index/sector pulse.

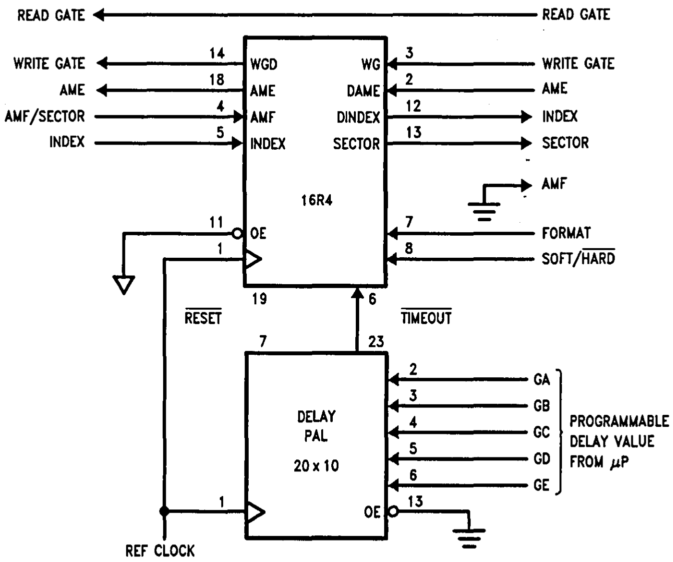
Read/Write Encoding/Decoding delays: Through encoding and decoding circuitry, a read data signal will be delayed by approximately one byte against a write data.

Write-after Read transient: A minimum delay of 0.3 μs must be provided between the trailing edge of read gate and the leading edge of write gate. This is accomplished by having at least one byte of header postamble.



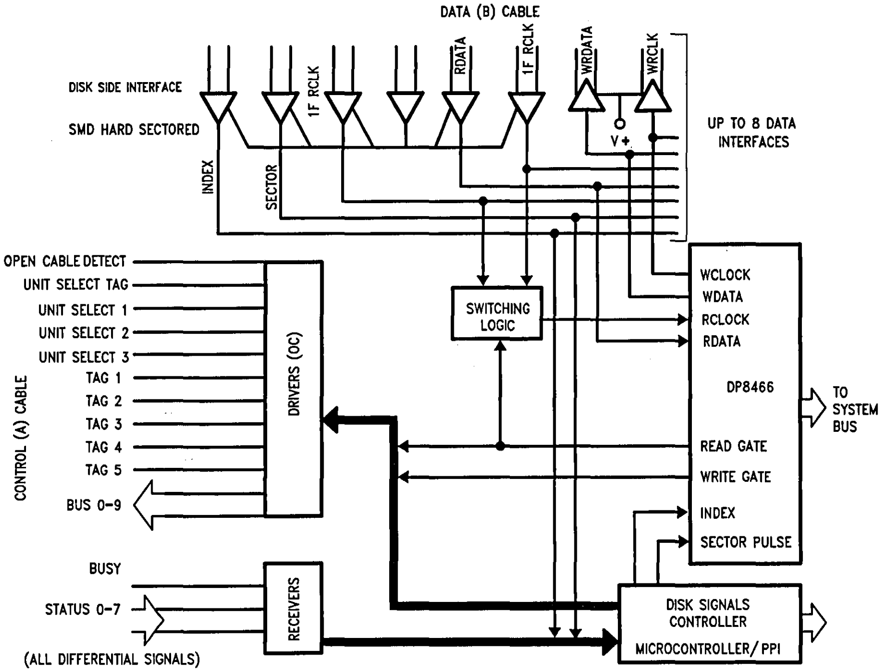
TL/F/8663-B4

FIGURE 6.23 (a). AME/AMF/Index/Sector Logic State Diagrams



TL/F/8663-B5

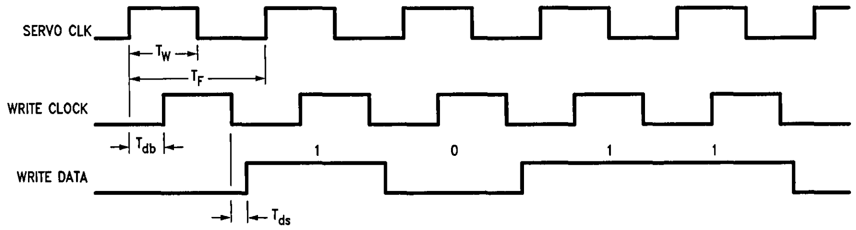
FIGURE 6.23 (b). ESDI AME/AMF Handshake Logic and State Diagram



TL/F/8663-B6

FIGURE 6.24. Data and Control Paths — SMD (Hard Sectored Drive)

1F Write Clock, Write Data/Write Clock Timing



T_W = TF/2

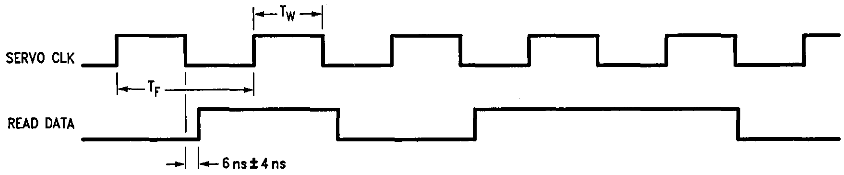
TF = (Function of Transfer Rate)

Tdb = Continuous delay within 2 bits

Tds = 0 ±10 ns

TL/F/8663-B7

FIGURE 6.25 (a). Write Clock/Write Data Timing



T_W = TF/2

TL/F/8663-B8

FIGURE 6.25 (b). Read Clock/Read Data Timing

Preamble length: The synchronization time required to allow the PLL to synchronize is 11 bytes before the sync pattern of address and data fields.

READ/WRITE TIMING

Representative read/write timing for format write, data read and data write operations are shown in *Figure 6.26*. The corresponding timings for the DDC are given in *Figure 6.20*.

FORMAT WRITE

During a format operation or write header operation, the drive requires that the Write Gate be asserted by 600 ns (max.), from the index/sector pulse and Write Gate must be de-asserted at least 1 byte after the check byte field in the header, refer *Figure 6.26(b)*. In the case of DDC, Write Gate is asserted 3.5 bit times after the Index pulse in a format operation or after the index/sector pulse in a write header operation (hard sectored drive). Write Gate is de-asserted at the end of header postamble field. Hence if the header postamble is kept at least one byte long, the DDC should satisfy the requirements of the drive. *Figure 6.26(a)* shows the recommended format used by "SMD" drive manufacturers like CDC, etc.

DATA WRITE

During a data write operation (essentially a compare header-write data operation) read gate is asserted by the DDC 3.5 bit times from the rising edge of the index/sector pulse. The recommended format for the MDD has a post index/sector gap and requires the Read Gate to be asserted 6 μs (approx. 8 bytes) from the index/sector pulse. This can be done in a similar fashion as outlined for the ESDI spec. (refer section—Handling the Post Index/Sector Gap in the ESDI format). Read Gate is de-asserted 2 bit times after the ID check bits field, which satisfies the requirement of 8 bit times maximum. Write Gate is then asserted 3 bit times after the header postamble, which implies that the header postamble can be a maximum of 3 bytes long in order to maintain the requirement of 4 bytes max. by which Write Gate must be asserted after the header check field. There is also a condition that Write Gate must be asserted at least 300 ns after Read Gate is de-asserted. This is satisfied by the DDC as seen from *Figure 6.20(b)*, the minimum time of de-assertion being 9 bit times assuming at least a 1 byte postamble. Write Gate is de-asserted at the end of the data postamble, hence, the data postamble must be at least 4 bytes long. *Figure 6.26(c)* gives the timing requirements for a data write operation.

DATA READ

During a read operation (essentially a compare header-read data operation), DDC asserts the Read Gate 3.5 bit times after the index/sector pulse. Hence the post index/sector gap has to be handled in a similar fashion as discussed in the Data Write section. Read Gate is de-asserted by the DDC 2 bit times after the ID check bit field, which is well within the MDD requirement of 8 bits max. If the continuing operation is Read data, then Read Gate is re-asserted by the DDC 11.5 bit times from the data preamble, *Figure 6.20(b)*, which certainly satisfies the requirement of a 1 byte minimum before re-assertion of the Read Gate. Read Gate is de-asserted by the DDC 2 bit times after the data check bit field. *Figure 6.26(d)* gives the timing requirements of the data read operation.

SPECIAL CONSIDERATIONS FOR SMD SOFT SECTORED DRIVES

For a soft sectored "SMD" drive, the recommended format is similar to the one in ESDI. The sector starts with the address mark field (three bytes of no flux transitions). The AME signal is available on the bit 5 of the bus with Tag 3 active while the AMF signal is available on bit 5 of the Status bus with both Tag 4 and Tag 5 inactive. The AME/AMF handshake is handled in a similar fashion as discussed in section 6.2.4.

HANDLING THE SEPARATE CLOCKS FOR READ AND WRITE OPERATIONS IN THE SMD DATA CABLE

The SMD interface has two reference clocks, one for write (1F WCLK/SERVO CLOCK) and one for read (1F RCLK/READ CLOCK). The DDC requires that the Read/reference clock be provided on the same line (RCLK) and that the switching between the reference clock and the PLL locked frequency clock be such that there are no glitches on the line going to the DDC. To accomplish this the two clock signals, 1F Read Clock/READ CLOCK and 1F Write Clock/Servo Clock need to be multiplexed during read and write operations to switch the appropriate clock signal going to the DDC. It should also be made sure that there are no glitches or short pulses in the process of switching. Since the PLL has a certain finite time for lock, the actual switch of the clocks occurs after a finite lock time from Read Gate assertion. Hence the Read Gate used to mux the two clocks must be delayed by the lock time (worst case) before it is sent to the switching logic. This is shown as the box 'switching logic' in *Figure 6.24* and given in detail in *Figure 6.27*. Besides these other timing requirements are compatible with those of the DDC. This circuitry has also been realized in a PAL.

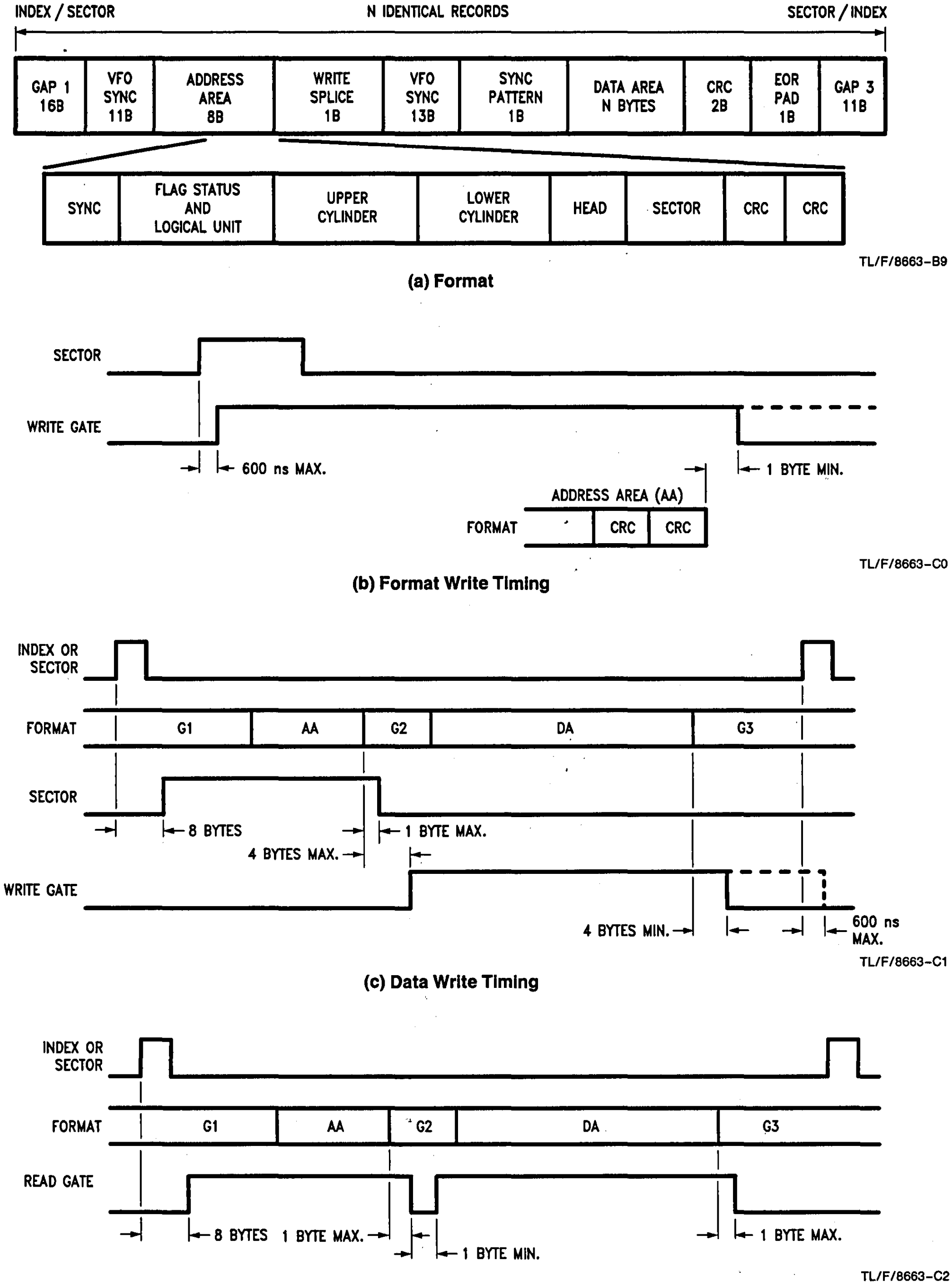
6.2.6 Miscellaneous ESDI/SMD Considerations

As with most standards, actual devices that follow the standards have their own idiosyncracies. ESDI and SMD standards are no exception. In addition some standards define design constraints which do not necessarily exist with actual devices. Some of these have been discussed earlier (for example de-assertion of Write Gate between ID and Data fields in ESDI). The following subsections describe some additional SMD & ESDI considerations that may be necessary depending on actual drive implementation.

HANDLING THE OPTION OF DE-ASSERTION OF WRITE GATE BETWEEN THE ID AND THE DATA FIELDS, IN THE ESDI FORMAT SPECIFICATION

The option of de-asserting Write Gate between the ID and the Data field is not directly supported by the DDC. The purpose of this may be indication to the encoder, of the start of the data preamble field in case of RLL encoding, so that the encoder can send actual data rather than the encoded data for the preamble pattern. To support this, external logic can be used. This logic would be gated from the trailing edge of the DDC's Serial Data Valid signal, count until the 2 byte pad, (header postamble) has been sent (at the end of the ID segment), then force Write Gate low for the desired time, and then re-enable it. This problem can also be effectively circumvented by first doing a two pass format operation. This first pass does a format operation as above, then a second pass-write data operation is done on all of the sectors. If the Write Gate pulse is used to initiate a drive generated preamble, then by performing a compare header-write data operation the correct data preamble will then be written, and the data field can then be properly read, refer *Figure 6.18* for timing with respect to the DDC.

1



(a) Format

(b) Format Write Timing

(c) Data Write Timing

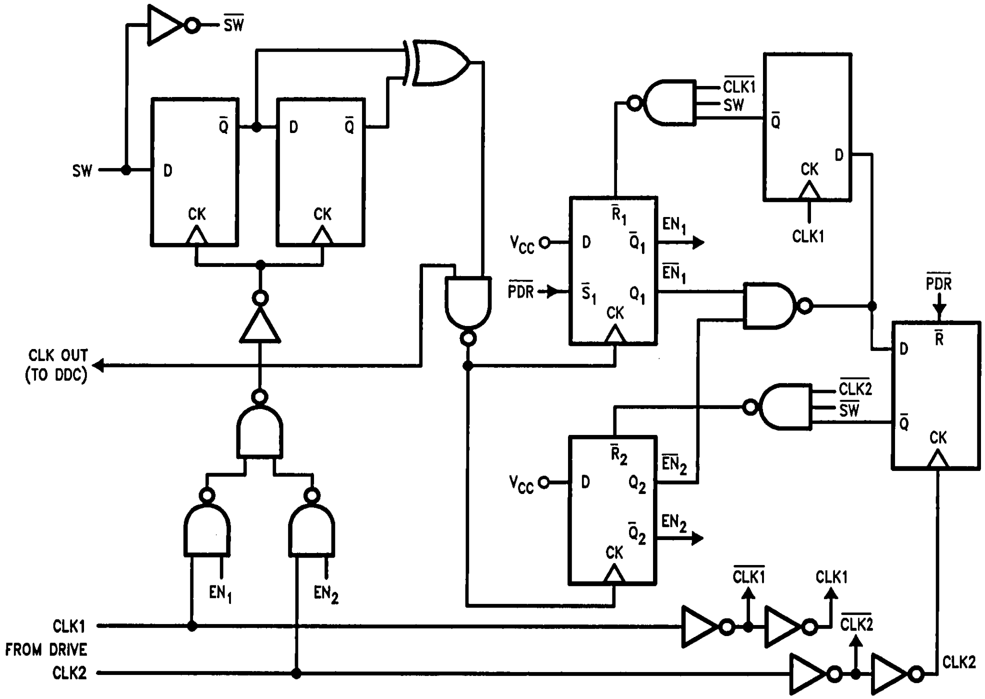
(d) Data Read Timing

FIGURE 6.26. Read/Write Timing (SMD)

A note of caution to be observed for ESDI hard sectored drives during format operation for RLL encoding. Since the leading edge of the write gate may be used by the encoder to trigger the generation of the preamble, this would be a problem as Write Gate is asserted 3.5 bit times after the index pulse and there would be no de-assertion of Write Gate with each sector pulse. The two pass format operation is a good software solution. In hardware the inverted sector pulse could be ANDed with Write Gate to generate the Write Gate to the encoder. Generally after a format write, then read is necessary to determine defective sectors, and initiate some sector substitutions.

HANDLING THE WRITE SPLICE FIELD BETWEEN THE ID AND DATA SEGMENTS IN THE ESDI/SMD FORMAT

The ESDI/SMD format specification recommends a two byte header postamble and a one byte write splice. The DDC format parameters indirectly support a write splice field between the ID and data segments. Consider normal operation of the DDC. The format is programmed to have a 2 byte header postamble and a data preamble one byte longer than the desired length. This byte is taken as the write splice, (a floating byte). During a write operation this floating byte is considered as part of the data preamble, so write gate is asserted 3 bit times into the data preamble i.e., the 'write splice' and data would be written on the media after taking into effect the write propagation delay. In case of a read operaion this byte is taken to be part of the header postamble. Hence as Read gate is asserted after the header postamble, it would never be asserted in the write splice. Normal operation could be achieved with the DDC without physically having a write splice field between the ID and the Data segment, rather creating one by adjusting other field lengths.



TL/F/8663-C3

FIGURE 6.27 (a). MUX and Deglitcher Circuitry to Switch between Read Clock and Reference Clock

Note 1: SW is low at start up, (L selects CLK 2, H selects CLK 1)

Note 2: POR = Power on reset (EN2 → 1, EN$_1$ → 0)

Note 3: Worst case latency (SW to actual switching) = 1.5 periods of clock switching from +2 periods of clock switching.

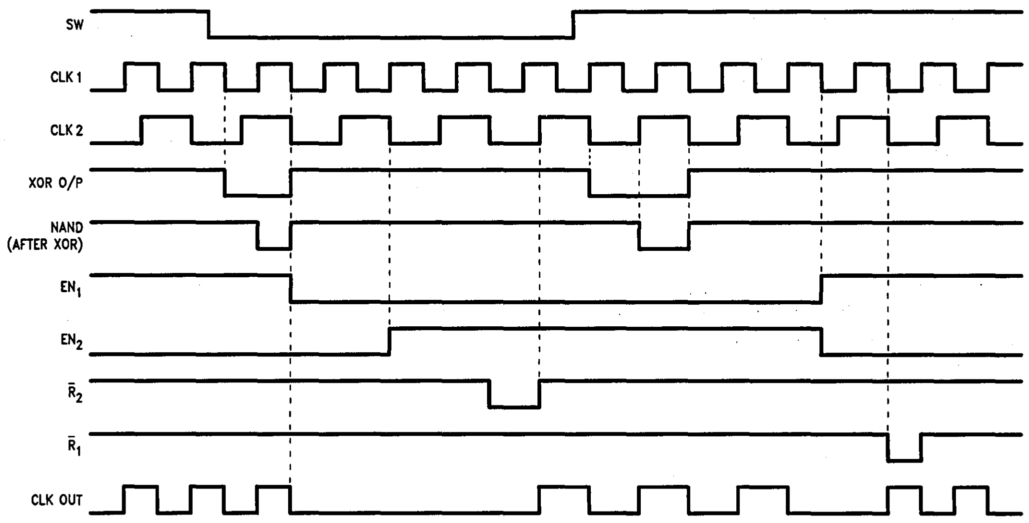


FIGURE 6.27 (b). Timing Diagram for Switching Logic

HANDLING THE POST INDEX/SECTOR GAP FIELD IN THE ESDI/SMD FORMAT SPECIFICATION

In the recommended format of the ESDI/SMD specification there is a gap after the index/sector pulse, referred to as the post index/sector gap. This is necessary mainly to accommodate head transients, read-to-write transients, write-to-read transients, etc. In the DDC, there is no format parameter to implement this field. Hence to implement this, external logic is needed, whereby the index/sector pulse to the DDC is delayed from the index/sector pulse from the drive by the desired gap count using counters. This is done for all format, read and write operations. The gap pattern for the intersector gap is then written for this post index/sector gap field also. Refer *Figure 6.28*.

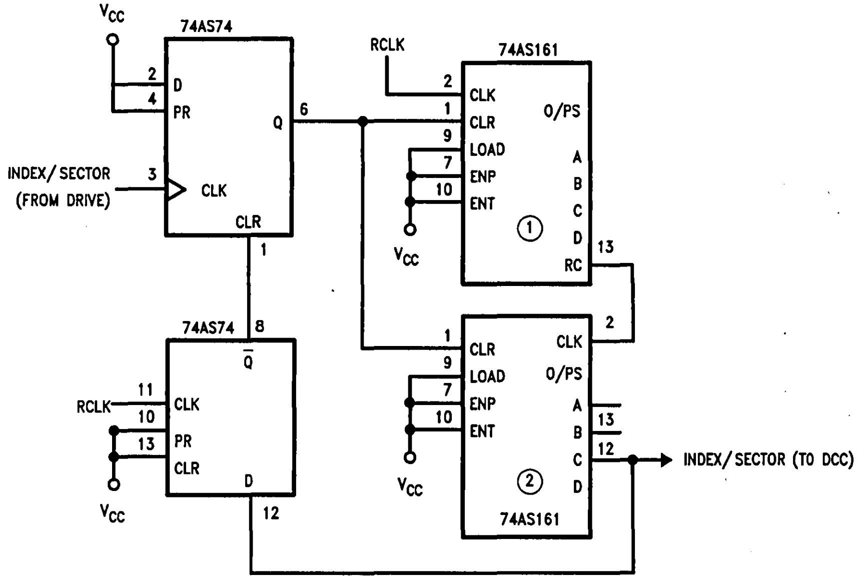
READ GATE DELAY

As discussed earlier, the separation between assertion of Read Gate and Write Gate at the beginning of the sector is 0.5 to −0.5 bit times. This may not accommodate the write splice associated with the Write Gate due to write driver turn on time etc., which is generally about 8 bit times from write gate assertion. Hence with the existing timing Read Gate may get asserted in the write splice area while reading the header. This might be a problem during the Format operation, in the very first sector of the track, because Write Gate is asserted after the Index pulse and remains asserted through the track till the Index pulse is encountered again after one revolution of the disk. Hence for the very first sector, Read Gate would have to be delayed to avoid the write splice. It would not affect other sectors.

An important point to note here is that if a Write Header operation is done on any sector, then the Read Gate may have to be delayed for that sector also.

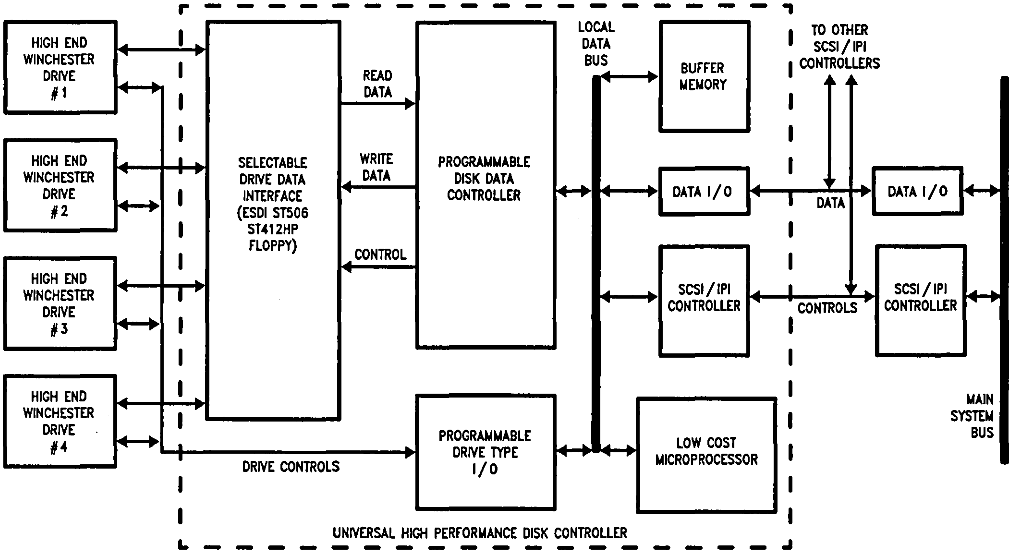
6.2.7 Intelligent Disk Interfaces

The overall objective of interfaces in this category, (like SCSI, IPI), was to make it easier for computer systems to talk to disk drives while ensuring minimum overhead for the host system. The Controller would incorporate a drive level interface on the disk side and a well defined interface to the host bus. The controller board has a local microprocessor which essentially controls the disk controller (data path), disk control signals (control path) and communication with the host system. Actually the DDC interfaces directly to the local bus. The microprocessor essentially controls the transfer of data using the DDC's DMA capability, from the local memory to the system memory, etc. *Figure 6.29* shows a block diagram of a high performance mass storage system, incorporating a SCSI peripheral bus.



TL/F/8663-A6

FIGURE 6.28. Logic to Implement Post Index/Sector Gap Field



HIGH END WINCHESTER DRIVE #1
HIGH END WINCHESTER DRIVE #2
HIGH END WINCHESTER DRIVE #3
HIGH END WINCHESTER DRIVE #4
SELECTABLE DRIVE DATA INTERFACE (ESDI ST506 ST412HP FLOPPY)
READ DATA
WRITE DATA
CONTROL
PROGRAMMABLE DISK DATA CONTROLLER
DRIVE CONTROLS
PROGRAMMABLE DRIVE TYPE I/O
LOCAL DATA BUS
BUFFER MEMORY
DATA I/O
SCSI/IPI CONTROLLER
LOW COST MICROPROCESSOR
TO OTHER SCSI/IPI CONTROLLERS
DATA
CONTROLS
DATA I/O
SCSI/IPI CONTROLLER
MAIN SYSTEM BUS
UNIVERSAL HIGH PERFORMANCE DISK CONTROLLER

TL/F/8663-C4

FIGURE 6.29. High Performance Mass Storage System

6.3 CIRCUIT BOARD LAYOUT AND SUPPLY ROUTING

There are several considerations to PC board design that should be followed. These guidelines serve to minimize problems that can occur with any high speed digital device. Since the DDC can operate at speeds approaching 30 MHz on the disk side, and up to 20 MHz on the system side, typical high speed design techniques should be employed to reduce noise, transmission line, and crosstalk effects. These are described below.

6.3.1 General Layout Considerations

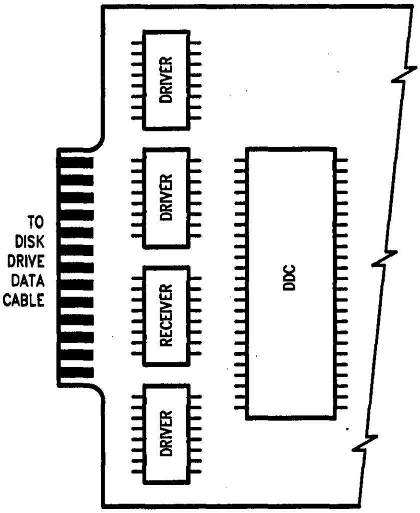
The DP8466 has two design areas of routing and loading on its signal lines, namely: disk interface and bus interface signals. For the disk signals, Read Data, Read Clock, Write Data, and Write Clock can be very high speed signals. It is recommended that when interfacing these signals to the interface line drivers/receivers, these lines be kept short as possible. Also the data cable interface devices should be located close to the data connector. It is especially important to minimize noise, propagation delay skews and jitter on these lines when in MFM mode because excessive skew, noise and jitter can lead to increased error rates. A generalized PCB chip layout is shown in *Figure 6.30*.

The bus signals generally should be treated the same as any VLSI's bus interface. The Address/Data bus provides 16 lines capable of 2 mA DC drive, and can drive variable loads up to about 100–120 pF at 20 MHz (larger loads can be driven although not at full speed). If more than 150 pF or 2 mA load needs to be driven, the data bus should be buffered, as shown in *Figure 6.31*. The one "trick" is to always connect the address de-multiplexing latches directly on the DDC, this ensures maximum address latch strobe setup time. The buffers can then drive the rest of the system. As with any high speed bus good layout practices should be followed to minimize crosstalk and reflections.

6.3.2 Decoupling, and V_{CC} and Ground Routing Guidelines

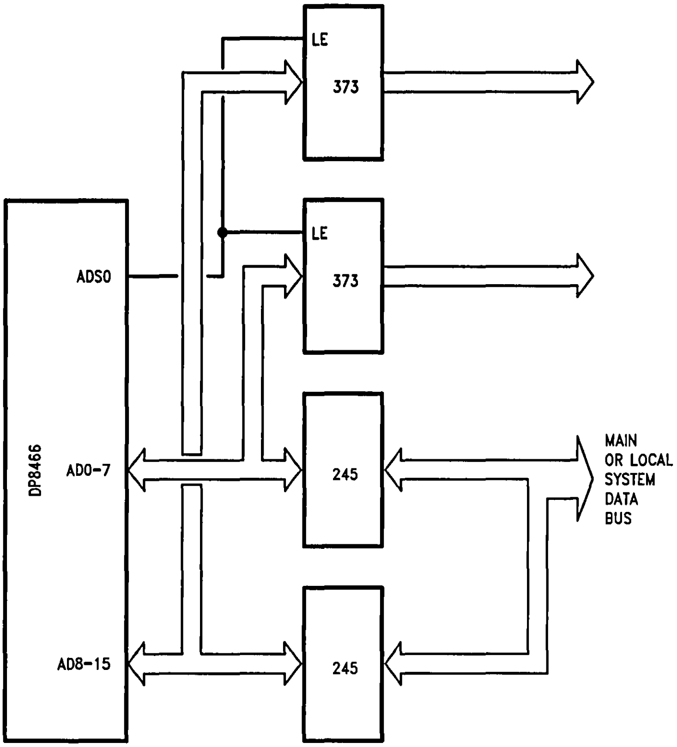
Due to the combined high speed and wide data bus of the DDC it is important to follow good power supply layout practices. Any noise generated by toggling of outputs can cause noise on V_{CC} and this in turn can reflect noise back into other inputs or outputs. The result is noise and glitches appearing on other inputs or outputs. This can be especially true when the address/data bus is toggling many lines simultaneously. In this case it is not unusual to have peak current spikes up to 300 mA being generated when toggling 16 or more outputs.

V_{CC} and ground noise problems can be easily minimized by following some simple rules when laying out the PCB. In general, multi-layer printed circuit layouts should include V_{CC} and ground planes. If a simple two-sided board, then wide V_{CC} and ground traces should be used, and an effort to layout V_{CC} and ground planes on the foil and component sides should be made. Most importantly, the DDC should have a decoupling capacitor placed as close to its V_{CC} and ground pins as possible, as shown in *Figure 6.32*. This capacitor should be a low series inductance type ceramic capacitor. Additionally, it may be desirable to add a 3–10 µF tantalum capacitor in parallel with the ceramic to offer further decoupling.



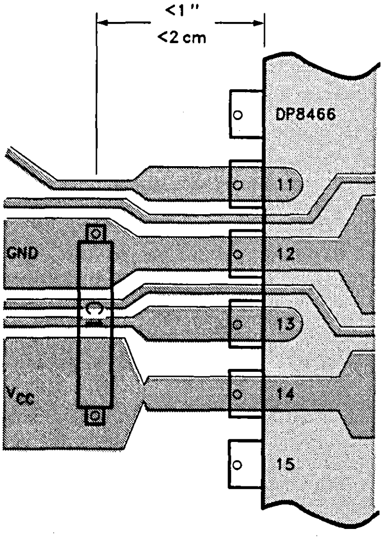
TL/F/8663-18

FIGURE 6.30. Conceptual Component Placement to Ensure Short PCB Traces between Connector and DDC



TL/F/8663-I5

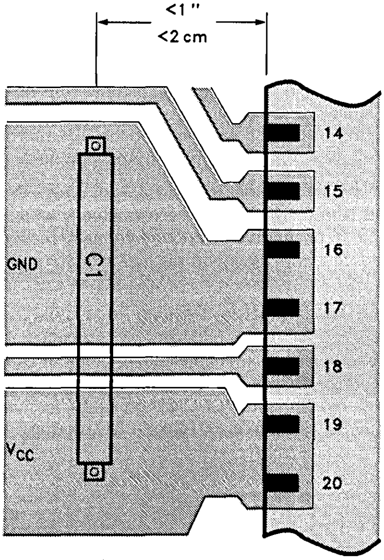
FIGURE 6.31. Proper Buffering of DDC's Address/Data Bus by Latching Address In



C1 = 0.1 μF Ceramic

TL/F/8663-I6

(a) Dual-In-Line Package



C1 = 0.1 μF Ceramic

TL/F/8663-I7

(b) Plastic Chip Carrier

FIGURE 6.32. Typical PC Board Layout for Decoupling the Power Supply (X-Ray View of Foil Side from Component Side)

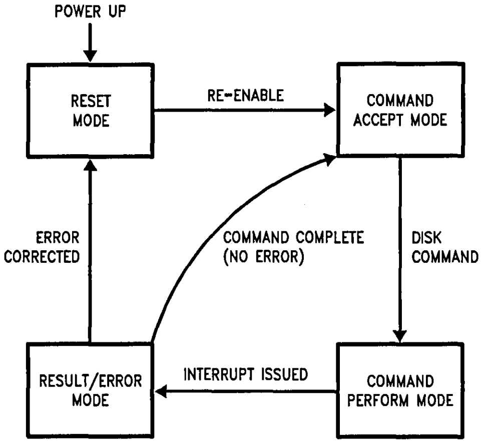
Chapter 7 DDC Functional Operations

7.0 INTRODUCTION

In this chapter, all the main DDC functional operations such as Disk Formatting, DMA Data transfers, Error detection and Correction, and basic Disk Read/Write are discussed in detail from software point of view. Also, the power-up, chip initialization procedure, and interrupt servicing is described.

7.1 OPERATING MODES OF THE DDC

The DDC can be thought of as operating in one of the four modes; Reset, Command Accept, Command Perform or Error. *Figure 7.1* shows a flow chart for these modes.



TL/F/8663-C5

FIGURE 7.1. DDC Operating Modes

RESET MODE:

DDC is put in the reset mode after power-up or prior to starting a new operation if the previous operation was aborted. After the DDC has been reset and the DDC has been reenabled, it moves to the Cornmand Accept mode.

COMMAND ACCEPT MODE:

The DDC is free and ready to receive a command. Various command and pattern registers and counters may be loaded to perform a disk operation, such as format, read or write etc.

COMMAND PERFORM MODE:

In this mode, the DDC executes the disk command that was loaded into it in the Command Accept mode. It carries out DMA operations. On a successful or unsuccessful completion of the operation, the DDC will generate an interrupt (if the interrupts were enabled, EI bit in OC register), and enters the Result/Error mode.

Note: If interrupts were not enabled, then the Status register should be polled in order to find out the result.

RESULT/ERROR MODE:

In this mode, the Status and Error registers should be read to find out the result of the operation or the type of error that occurred. If the operation was completed successfully, the DDC will go back to Command Accept mode. If an error occurred during the execution of the command, the DDC will abort the operation and the Error register will indicate the type of error that occurred. To perform the operation again or to correct the error, the DDC must be reset and loaded for a particular operation, hence it goes back to the reset mode.

7.2 INITIALIZATION

After the DDC is hooked up in the system, it can be powered up and initialized to perform the desired disk operation. The chip power-up reset, operation initialization, and register programming are discussed in the following paragraphs. A flow chart shown in *Figure 7.2* describes a basic algorithm for a power-up, reset and initialization procedure.

7.2.1 Power-up and Reset

After the chip power-up, the DDC must be held reset for a duration of at least 4 BCLK and 32 RCLK periods (with these clocks active) before it could be assigned a disk format or it could be set for any disk operation. The DDC can be reset by asserting the RESET pin low or by setting the internal RES bit in the OC register high. After the DDC has been reset (for the time indicated above), The external RESET pin must be deasserted and internal RES bit in OC register must be cleared. When the system is powered, the DDC should be reset immediately, even if it will not be used right away. This is because its internal sequencers may be randomly powered on into a state that could draw some excessive I_{CC} currents.

7.2.2 Disk Operation Initialization

After a reset, the DDC must be re-enabled by setting RED bit high in the Drive Command (DC) register before other registers are loaded for a particular operation. Once the DDC is reenabled, it is ready to perform a disk operation such as read, write, or format. Various parameter and command registers and counters can then be loaded depending on the type of operation. In most of the operations, the Drive Command (DC) register is loaded the last except when the DDC is configured in a Non-Tracking DMA mode.

7.2.3 Register Programming

In order to perform an operation, related registers can be loaded in any order keeping the following restrictions in mind.

- The Drive Command (DC) register must be the last register to be loaded for any DDC operation. There is one exception to this. In non-tracking DMA mode, a remote DMA operation may be initiated by loading the operation command (OC) register after a disk command has been started.
- If the on-chip DMA is being used, or if the Remote Data Byte Count registers will be read back, the Local and Remote Transfer registers must be loaded before the Sector Byte Count and Remote Data Byte Count registers are loaded.
- The Number of Sector Operations (NSO) counter must be loaded after an external RESET or internal RESET (in OC register) are both inactive. Other registers can be loaded while reset is active.

- During the execution of an operation (format, read or write), the pattern and count registers **must not** be read. Reading these registers will interfere with the DDC's operation, and could cause some bizarre results. These registers may be written to at anytime. When data is written it takes effect immediately. Thus the only caution is to not change a pattern or count register randomly as the system may not know whether the change will apply to the current sector or the next. The Interlock mode with the Header Complete interrupt should be used to synchronize register updates.

7.3 DISK FORMATTING

The DDC can be programmed to format a disk with any type of sector format. A versatile and flexible sector format with various formatting techniques are incorporated in the DDC. Various formatting features, methods and the DDC sector format options are discussed below.

7.3.1 Key Features

MFM/NRZ Data:

The DDC can be programmed to output MFM or NRZ data while writing to the disk (bit MFM in the Disk Format register). In case of MFM data, some external circuitry will be required, as indicated in section 6.2.

Hard/Soft Sectored Drives:

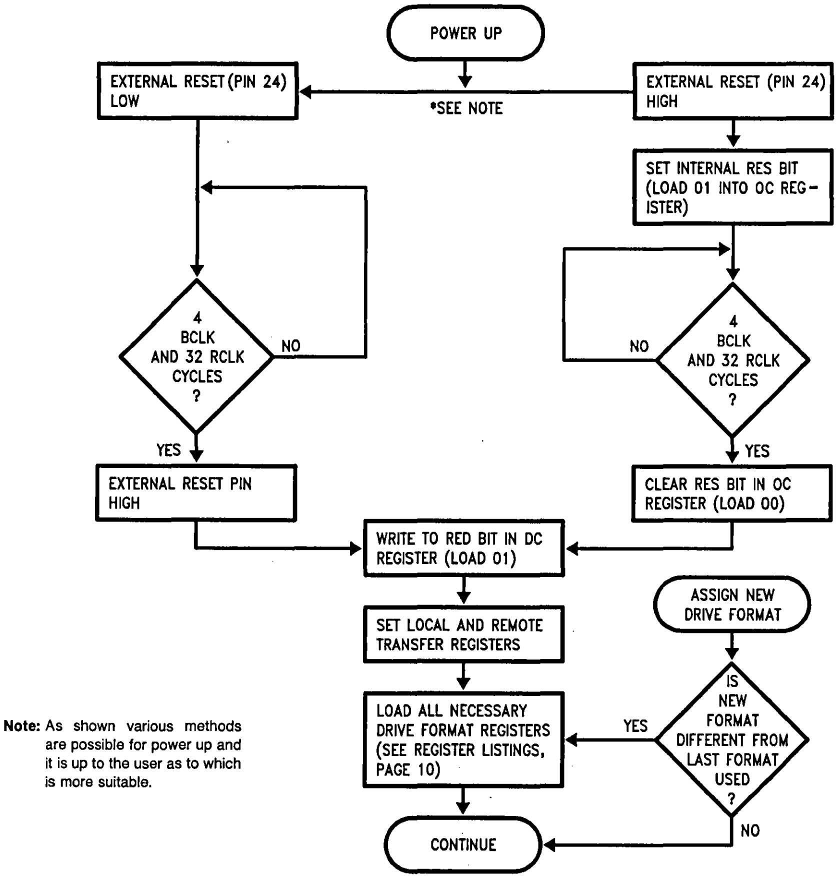
The DDC can be interfaced to both hard and soft sectored drives. This is done through HSS bit in Disk Format register. See section 6.2 and 7.3.3 for implementation of various hard and soft sector formats with the DDC.

7.3.2 Sector Format Options

The DDC offers a versatile sector format which can accommodate most of the currently used disk formats. The DDC sector format options are shown in *Figure 7.3* and the associated registers to program various format fields are shown in *Figure 7.4*. Various ID and Data fields are described below. Discussion on implementation of the popular disk formats using the DDC is given in the next section.

ID And Data Preamble:

There are two fields provided for ID and Data Preambles, which are required in hard and soft sector formats. The ID Preamble field can also be used as an Address Mark (field of no transitions) in case of formats with sector mark (ESDI or SMD type formats). Up to 31 bytes each for ID and Data preambles are allowed. These fields can be programmed using ID and Data Preamble Pattern and Byte Count registers (addresses 31H and 21H for ID, and addresses 3DH and 2DH for the Data).


POWER UP
*SEE NOTE
EXTERNAL RESET (PIN 24) LOW
EXTERNAL RESET (PIN 24) HIGH
SET INTERNAL RES BIT (LOAD 01 INTO 0C REGISTER)
4 BCLK AND 32 RCLK CYCLES ?
NO
YES
4 BCLK AND 32 RCLK CYCLES ?
NO
YES
EXTERNAL RESET PIN HIGH
CLEAR RES BIT IN 0C REGISTER (LOAD 00)
WRITE TO RED BIT IN DC REGISTER (LOAD 01)
SET LOCAL AND REMOTE TRANSFER REGISTERS
ASSIGN NEW DRIVE FORMAT
IS NEW FORMAT DIFFERENT FROM LAST FORMAT USED ?
YES
NO
LOAD ALL NECESSARY DRIVE FORMAT REGISTERS (SEE REGISTER LISTINGS, PAGE 10)
CONTINUE

Note: As shown various methods are possible for power up and it is up to the user as to which is more suitable.

TL/F/8663-C6

FIGURE 7.2. Power-up and Initialization Algorithm

ID And Data Synch #1:

This field will normally be used for writing an ID and Data Address Mark in case of soft sector drives. For hard sector drives this field may either be skipped or be used to extend the ID Preamble or ID Synch #2 fields. Up to 31 bytes can be written in this field for both ID and Data using ID Synch #1 Pattern and Byte Count registers, addresses 32H and 22H for ID, and addresses 3EH and 2EH for data.

ID And Data Synch #2:

The ID and Data Synch #2 fields, used by the DDC for byte alignment, can also have up to 31 bytes each for ID and Data. These fields can be programmed using ID and Data Synch #2 Pattern and Byte Count registers (addresses 33H and 23H for ID and addresses 3FH and 2FH for data).

HEADER BYTES: Header bytes are used to specify the header information of a particular sector such as sector number, cylinder number, track number etc. At least 2 and maximum 6 bytes can be written using 6 Header Byte Pattern and associated control registers (addresses 14H, 15H, 16H, 17H, 18H, 19H and 24H, 25H, 26H, 27H, 28H, 29H respectively). See description of Header Byte Control register in Chapter 5.

ID and Data CRC/ECC:

The DDC can be programmed for a 2 bytes of internal CRC or up to 6 bytes of internal ECC appendage using the Disk Format register, for both ID and Data fields. The CRC appendage is internal to the DDC and no pattern or count

ID FIELD

ID PREAMBLE	ID SYNCH #1 (AM)	ID SYNCH #2	HEADER BYTES	ID CRC/ECC	ID EXT ECC	ID POSTAMBLE
0–31 Bytes	0–31 Bytes	0–31 Bytes	2–6 Bytes	0, 2, 4 or 6 Bytes	0–31 Bytes	0–31 Bytes

DATA FIELD

DATA PREAMBLE	DATA SYNCH #1 (AM)	DATA SYNCH #2	DATA FORMAT PATTERN	DATA CRC/ECC	DATA EXT ECC	DATA POSTAMBLE	GAP 3
0–31 Bytes	0–31 Bytes	0–31 Bytes	1–64K Bytes	0, 2, 4 or 6 Bytes	0–31 Bytes	0–31 Bytes	0–255 Bytes

FIGURE 7.3. Sector Format Options

Pattern Register	Hex Addr	Pattern Source	Control Function	Hex Addr	Control Register
ID Preamble	31	Internal	Repeat 0-31 Bytes	21	ID Preamble Byte Count
ID Synch #1 (AM)	32			22	ID Synch #1 (AM) Byte Count
ID Synch #2	33			23	ID Synch #2 Byte Count
Header Byte 0	14		Define/Control	24	Header Byte 0 Control
Header Byte 1	15			25	Header Byte 1 Control
Header Byte 2	16			26	Header Byte 2 Control
Header Byte 3	17			27	Header Byte 3 Control
Header Byte 4	18			28	Header Byte 4 Control
Header Byte 5	19			29	Header Byte 5 Control
ID CRC/ECC	**		16-BIT CRC/ 32-BIT 48-BIT ECC	35	Disk Format (DF) PPB0-5 PTB0-5
ID External ECC	*	External	0-31 Bytes	2B	ID External ECC Counter
ID Postamble	3C	Internal	Repeat 0-31 Bytes	2C	ID Postamble Byte Count
Data Preamble	3D			2D	Data Preamble Byte Count
Data Synch #1 (AM)	3E			2E	Data Synch #1 (AM) Byte Count
Data Synch #2	3F			2F	Data Synch #2 Byte Count
Data Format	3B		Field Size 1-6k Bytes	38 39	Sector Byte Count 0 Sector Byte Count 1
Data CRC/ECC	**		16-BIT/ 32-BIT 48-BIT ECC	35	Disk Format
Data External ECC	*	External	0-31 Bytes	2A	Data External ECC Counter
Data Postamble	30	Internal	Repeat 0-31 Bytes	20	Data Postamble Byte Count
Gap	3A		Repeat 0-255 Bytes	34	Gap Byte Count

*These are not pattern registers.

**CRC polynomial is built into the DDC and does not require any registers. ECC requires PPB0-5 and PTB0-5 registers.

FIGURE 7.4. Pattern Registers/Counters for Sector Format Fields

register is used besides the disk format register. In case of an ECC appendage, the Polynomial Preset and Tap Byte (0–5) registers (addresses 2H–DH) must be used. Also in case of ID CRC/ECC appendage, if an internal CRC/ECC appendage is not to be used then an external ECC must be used. An external ECC is not necessary if an internal CRC/ECC was not used for a Data CRC/ECC appendage.

ID and Data External ECC:

An external ECC may be appended to encapsulate and internal CRC or ECC, for diagnostic purposes, using an external ECC circuitry. Up to 31 bytes can be appended using ID and Data Ext. Byte Count registers (addresses 2BH and 2AH).

ID and Data Postamble:

Up to 31 bytes can be used for ID and Data postamble using ID and Data Postamble Pattern and Byte Count registers, addresses 3CH and 2CH (for ID) and addresses 30H and 20H (for data).

Data Format Pattern:

Data format is programmed by the Data Pattern register (address 3B) and then can be repeated up to 64K times through the Sector Byte Count registers (addresses 38H and 39H). The number of times data format is repeated depends on the sector size.

Gap 3:

Up to 255 bytes can be written using Gap Pattern and Byte Count registers (address 3AH and 34H). In soft sectored drive operation, the Gap 3 bytes are written for each sector (determined by Gap 3 Byte Count register) except the last sector. For the last sector, gap bytes will be written until an Index pulse is received. In case of hard sectored drives, the Gap 3 Byte Count register is only used while formatting the disk. In normal disk write operation, the DDC writes gap bytes until a Sector pulse is received, ignoring the contents of the Gap byte count register.

Considerations for Pattern and Count Register Programming for Format Operations

- If any Byte Count register is loaded with zero, that field will be excluded and no pattern for the corresponding Pattern register needs to be loaded. Similarly if any of six Header Byte Control registers is set with all bits equal to zero, no pattern for that byte needs to be loaded.
- Maximum two consecutive fields can be excluded from a sector format. This includes the six header bytes which may be thought of as six fields.
- Format operations always start with an index pulse and end with the next index pulse, thus making one track. The DDC can only be programmed to format one track at a time.

7.3.3 Implementing Some Popular Sector Formats

There are three general sector formats which are commonly used in various disk systems; soft sector format (Floppy, ST506 type formats), hard sector format (hard or fixed formats used in ESDI/SMD type drives), and format with sector mark (soft or variable formats used in ESDI/SMD type drives). These three types are shown in *Figure 7.5* and their implementation using the DDC is discussed below.

SOFT SECTOR (FLOPPY/ST506 TYPE) FORMATS

The field shown in *Figure 7.5 (a)* is the most commonly used sector format used in soft sectored drives such as Floppy and ST506/412/419 type winchester drives. A double density floppy format recommended by the IBM and a Seagate's ST506/412/419 type format are shown in *Figure 7.6* as examples. It can be seen from *Figure 7.5* that the fields used in these formats are in direct correspondance with the ones supported by the DDC except the Post Index Gap in floppy format or Gap 1 in ST506 format. The Post Index Gap field is not supported by the DDC and may be eliminated as part of the disk format. If it must be generated external hardware may be added as discussed in section 6.2. The implementation of rest of the format is very straightforward and can be achieved using the registers shown in *Figure 7.4*. Implementation of the ST506/412/419 type format is shown in *Figure 7.7* and Table 7.1.

TABLE 7.1. Implementation of ST506 Formats

ST506			DDC		
Field	Pattern	Byte Count	Field	Pattern Register Address	Byte Count Register Address
Gap 1	4E	16	*	*	*
SYNC	00	13	ID Preamble	31H	21H
ID AM	A1, FE	2	ID Synch #1	32H	22H
CYL	#	2	Header Byte	14H–5H	24H–5H
HD	#	1	Header Byte	7H	7H
SEC	#	1	Header Byte	19H	29H
CRC	*	2	ID CRC/ECC	**	**
Gap 2	00	3	ID Postamble	3CH	2CH
Gap 2	00	13	Data Preamble	3DH	2DH
Data AM	A1, F8	2	Data Synch #1	3EH	2EH
Data	——	256	Data Format	3BH	38H, 39H
CRC	****	2	Data CRC/ECC	****	****
Gap 3	00	3	Data Postamble	30H	20H
Gap 3	4E	15	Gap	3AH	34H
Gap 4	4E	352	Gap **	3AH	34H

*Gap 1, Post Index Gap, is not supported by the DDC. See section 6.2 and the hard sector format section in this chapter.

**The CRC is internal to the DDC and Disk Format register is used to select it.

***The DDC only allows 256 bytes for the Gap field. During a format the gap for the last sector of the track will be written until the pulse is received.

TABLE 7.2. Implementation of ESDI Hard Sector Format

ESDI FORMAT			DDC REGISTERS		
Field	Pattern	Byte Count	Field	Pattern Register Address	Count Register Address
	(See Note†)				
Inter-Sector Gap	00	‡	*	*	*
Address PLO Sync	00	‡	ID Preamble	31H	21H
(No Field)		0	ID Sync #1	32H	22H
Byte Sync Pattern	—	≥ 1	ID Sync #2	33H	23H
Cylinder	xxxx	2	Header #1/2	14/15H	24/25H
Head	xx	1	Header #3	16H	26H
Sector	xx	1	Header #4	17H	27H
Flag/Status	xx	1	Header #5	18H	28H
(No Field)		0	Header #6	19H	29H
Address Check Bytes	**	**	ID CRC/ECC	**	**
Address Pad	00	—	ID Postamble	3CH	2CH
Write Splice	00	1	***	***	***
Data PLO Sync	00	—	Data Preamble	3DH	2DH
(No Field)		0	Data Sync #1	3EH	2FH
Byte Sync Pattern	—	≥ 1	Data Sync #2	3FH	2FH
Data	xx	—	Data Pattern@	3BH	38/39H
Data Check Bytes	**	**	Data CRC/ECC	**	**
Data Pad	00	≥ 2	Data Postamble	30H	20H
Format Tol.	00	—	Gap	3AH	34H
Inter Sector Gap	00	‡	Gap	*	*

†Where dashed entries appear in these columns the ESDI standard does not specifically define these fields but leaves this up to the user.

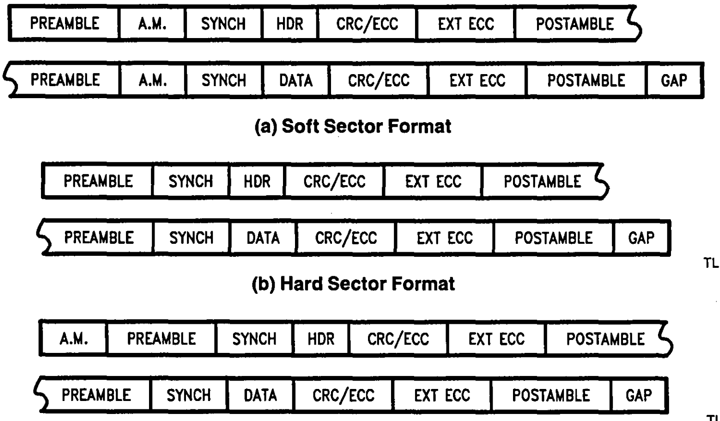
‡Defined by the ESDI specification using a specific formula.

*The standard Inter Sector Gap field is not supported by the DDC. See Hard sector section in this chapter, and section 6.2 for details in generating this field. Chapter 6's hardware actually uses the DDC's gap field to generate the ISG field for the next sector.

**The 16 bit CRC, 32 bit or 48 bit ECC is programmed internally see section 7.6 for details.

***The write splice field is not supported by the DDC. It should be included as part of the Data preamble field for format and write operations, and part of the ID postamble for read operations.

@Used only to format the data field.



TL/F/8663-C7

(a) Soft Sector Format

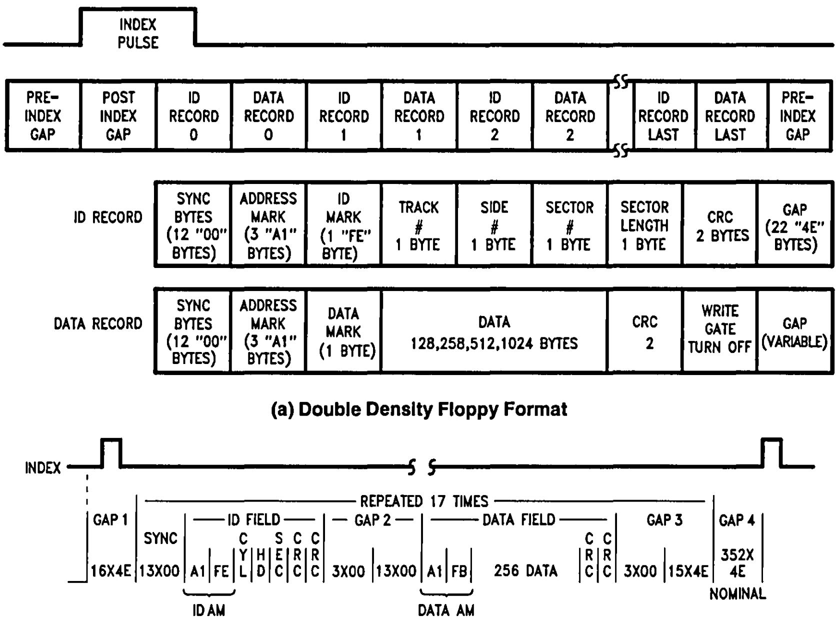
TL/F/8663-C8

(b) Hard Sector Format

TL/F/8663-C9

(c) Format with Sector Mark

FIGURE 7.5. Generalized Common Sector Formats



TL/F/8663-D0

(a) Double Density Floppy Format

TL/F/8663-D1

(b) ST506/412/419-type Sector Format Recommended by Seagate

FIGURE 7.6. Soft Sector Formats

HARD SECTOR FORMATS (Setting up Pattern and Count registers for Read/Write and Formatting)

The second sector format shown in *Figure 7.5* is the general hard sector format. In this format the beginning of each sector is marked by a sector pulse. This format is typically used in ESDI and SMD type drives. The format for these types of drives is fairly straight forward, and Table 7.3 shows the DDC registers, patterns and count lengths to perform and ESDI format. Table 7.4 shows Control Data Corp. recommended format for an SMD specification.

In both formats the Inter Sector Gap of ESDI and the Head Scatter Gap, commonly referred to as the Post Index or Post Sector Gap in SMD are not directly supported by the DDC. Additional counters and logic as shown in Section 6.2 previously are needed if these are to be supported. This solution would be appropriate for ESDI or SMD hard sectored drives. The philosophy behind it being that the index/sector pulse from the drive is presented to the DDC delayed by the external logic by the length of the post index/sector gap.

Another field in both SMD and ESDI that is not directly supported is the write splice field. This field is intended for turning off and on the read/write head without interfering with the data preamble. This field is easily generated by including it in the data preamble for formatting. Then during read operations including it in the ID postamble, and for write operations including it in the Data preamble.

For ESDI drives during the format operation the standard calls out for optionally deasserting Write Gate for two bit times during the write splice. This option is useful if the drive is performing some data encoding (such as 2–7). Pulsing Write Gate for two bit times informs the encoder/decoder on the drive to start the data preamble field. The DDC does not directly support this deassertion of Write Gate, but can format the drive easily anyway, by performing a two pass format operation. For the first pass the DDC will be set up to format the whole track, however only the ID fields will actually be formatted due to the encoder. The second pass should be a multi-sector write data operation, which will format the data fields.

FORMATS WITH SECTOR MARKS (Setting up Pattern and Count registers for Read/Write and Formatting)

The third method of formatting a disk is with sector marks. A simplified typical format is shown in *Figure 7.5(c)*. This format is most common with soft sectored ESDI, and a few SMD drives. Basically this format uses a field of no flux transitions and some special drive hardware to enable the drive to generate a sector pulse-like signal called Address Mark. The Address Mark signal thus signifies the start of a new sector.

To implement this using the DDC, requires some manipulation of the format parameter RAM in conjunction with the use of the ESDI control PAL hardware described in section 6.2. Other external hardware designs may require altering the format parameters, however, much of the discussion is still applicable. While formatting, the ID preamble field of the DDC is set with the pattern of the Post index gap. The count length of this field is set to the length of the gap plus length of address mark (usually 3 bytes). The ID synch #1 field contains the pattern of the preamble while its length is increased by one to accommodate the Address Mark pad field. This ID Sync #2 contains the byte synch pattern.

The SAM bit in the Disk Format register is set. Hence when formatting is initiated, the DDC generates AME and the start of its preamble. This is taken by the PAL and external hardware, and delayed by the length of the gap (which is being written), to when the address mark (3 bytes of no transitions) is written. This is then followed normally by the preamble and synch fields.

TABLE 7.3. Implementation of SMD Hard Sector Format

SMD FORMAT			DDC REGISTERS		
Field	Pattern	Byte Count	Field	Pattern Register Address	Count Register Address
Head Scatter Bytes	00	16	*	*	*
Address PLO Sync	00	11	ID Preamble	31H	21H
(No Field)		0	ID Sync #1	32H	22H
Byte Sync Pattern	—	1	ID Sync #2	33H	23H
Flag/Status	—	1	Header #5	14H	24H
Cylinder	—	2	Header #2/3	15/16H	25/26H
Head	—	1	Header #3	17H	27H
Sector	—	1	Header #4	18H	28H
(No Field)		0	Header #6	19H	29H
Address Check Bytes	CRC/ECC	2-6	ID CRC/ECC	**	**
(No Field)		0	ID Postamble	3CH	2CH
Write Splice	00	1	***	***	***
Data PLO Sync	00	11	Data Preamble	3DH	2DH
(No Field)		0	Data Sync #1	3EH	2FH
Byte Sync Pattern	—	1	Data Sync #2	3FH	2FH
Data	xx	—	Data Pattern@	3BH	38/39H
Data Check Bytes	CRC/ECC	2-6	Data CRC/ECC	**	**
EOR Pad	—	1	Data Postamble	30H	20H
End of Sector	—	10	Gap	3AH	34H

†Where dashed entries appear in these columns the SMD standard does not specifically define these fields but leaves this up to the user.

‡The Data Pattern is used during format operations only.

*The standard Inter Sector Gap (Head Scatter Bytes) field is not supported by the DDC. See Hard sector section in this chapter, and section 6.2 for details in generating this field. The hardware in chapter 6 uses the DDC's gap filed to generate the Head Scatter Bytes for the next sector.

**The 16 bit CRC, 32 bit or 48 bit ECC is programmed internally see section 7.6 for details.

***The write splice field is not supported by the DDC. It should be included as part of the Data Preamble field for format and write operations, and part of the ID postamble for read operations.

@Used only to format the data field.

When reading from the disk the DDC looks on the drive like a hard sectored one. The Address Mark is detected, and the drive asserts AMF which is decoded by the PAL into a sector pulse for the DDC. Hence the DDC format parameter RAM should be changed to the ID preamble field containing the pattern of the actual preamble and count length. The ID synch #1 field count is set to zero so that this field is skipped and the ID synch #2 contains the byte synch. (The read gate is delayed by 8 bits to accommodate the 1 byte write splice field). This ensures a successful read operation. With the DDC working in this mode it would see an index and a sector pulse from the sector 0 AMF separated by the Post Index Gap length. The DDC takes the index to be the sector 0 pulse also. The ESDI control PAL of chapter 6 takes care of delaying the index pulse over the sector 0 AMF pulse and suppresses the sector pulse to the DDC from the sector 0 AMF.

If a Write Header operation is desired at any time then it would be possible to do so only if a regular format operation had been done earlier. The AMF from the drive generates a sector pulse to the DDC to start writing the header. Also the Read Gate assertion is delayed externally by a byte to accommodate the Write Splice associated with Write Gate assertion. The ESDI field names byte values and lengths along with the DDC's equivalent registers are shown in Table 7.5.

The DDC does not directly support the ESDI write splice field as before, however, this is easily remedied by including this byte as part of the postamble during formats and write operations as part of the data preamble during read operations. This ensures that reading of the write splice is avoided which is the purpose of this field.

As in the hard sector format, ESDI has an optional deassertion of the Write Gate in between the ID and Data fields. The DDC does not support this option, but by performing a two pass format as described above a complete format can be accomplished.

MODIFICATIONS TO SECTOR FORMATS

While the previously discussed "standard" formats are a useful starting point, they are generally not strictly adhered to when actually formatting a disk. This is due usually to the users desire to optimize the drive for various parameters. These include optimizing data integrity, data separator per-

TABLE 7.4. Implementation of ESDI Sector Mark Format

ESDI FORMAT			DDC REGISTERS		
Field	Pattern	Byte Count	Field	Pattern Register Address	Count Register Address
	(See Note†)				
Inter-Sector Gap	00	‡	*	*	*
Pad	00	1	ID Preamble	31H	21H
Address PLO Sync	00	—			
Address Mark	†††	3	ID Sync #1	32H	22H
Byte Sync Pattern	—	≥ 1	ID Sync #2	33H	23H
Cylinder	xxxx	2	Header #1/2	14/15H	24/25H
Head	xx	1	Header #3	16H	26H
Sector	xx	1	Header #4	17H	27H
Flag/Status	xx	1	Header #5	18H	28H
(No Field)		0	Header #6	19H	29H
Address Check Bytes	**	**	ID CRC/ECC	**	**
Address Pad	00	≥ 2	ID Postamble	3CH	2CH
Write Splice	00	1	***	***	***
Data PLO Sync	00	—	Data Preamble	3DH	2DH
(No Field)		0	Data Sync #1	3EH	2FH
Byte Sync Pattern	—	≥ 1	Data Sync #2	3FH	2FH
Data	xx	—	Data Pattern@	3BH	38/39H
Data Check Bytes	**	**	Data CRC/ECC	**	**
Format Tol.	00	‡	Gap	*	*
ISG	00	‡	Gap	*	*

†Where dashed entries appear in these columns the ESDI standard does not specifically define these fields but leaves this up to the user.

‡Defined by the ESDI specification using a specific formula.

†††For the Address mark the pattern does not matter, as an area of no flux transitions is recorded.

@The Data Pattern register is used only for format operations.

*The standard Inter Sector Gap field is not supported by the DDC. See ESDI control PAL hardware solution in chapter 6.2 for details in generating this field.

**The 16 bit CRC, 32 bit or 48 bit ECC is programmed internally see section 7.6 for details.

***The write splice field is not supported by the DDC. It should be included as part of the data preamble field for format and write operations, and part of the ID postamble for read operations.

formance, access speed, sector defect sparing algorithms, and total storage capacity. Some of these considerations are discussed below.

Error Detection/Correction And Data Field Length

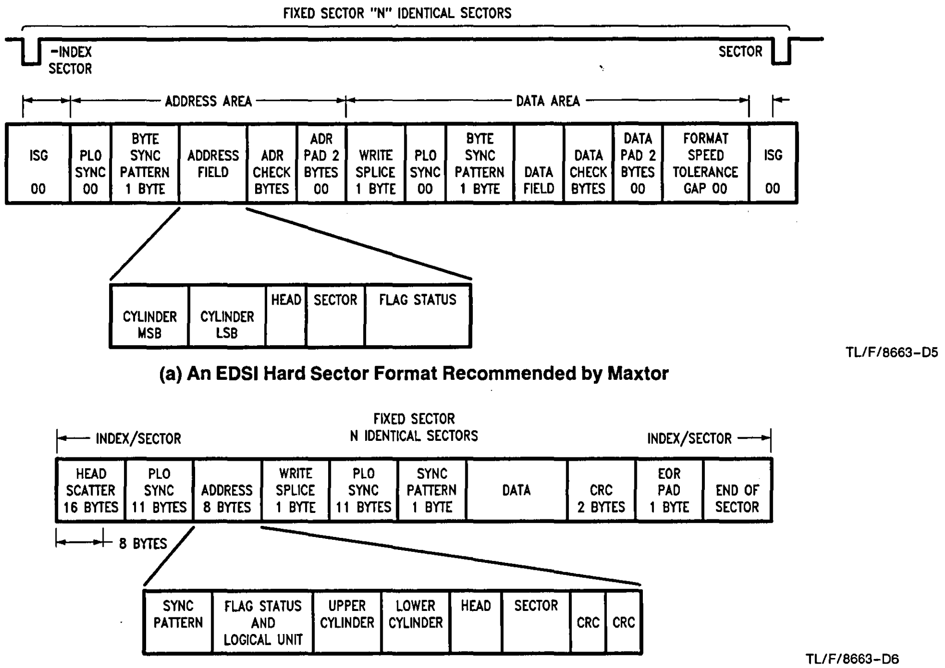
Generally, ST506 type drive recommended formats utilize a 16 bit CRC, however, this generally does not offer the type of data integrity that is needed in the data field. Thus generally ST506 type drives should utilize a 32 ECC for data. The ID field usually can be a CRC check field since the header is so short, and since recovery of the header address can be achieved by reading the previous sector header.

In general the type of ECC field to choose for the data field depends on the length of the data field, the defects on the disk media itself and the tolerance the designer places on his system for detecting errors, and the probability limits for miscorrection.

The choice for the length of the data field is determined in part by the type of ECC chosen (or vice versa), but also by system performance criteria. For a given media, a longer data field, and fewer sectors per track can maximize total storage capacity, but it requires better ECC. In many cases where the disk is to store many small files that leave many partially filled sectors, a larger sector size will waste disk space. However, if a few very large files are stored when large data field maximize storage. Thus usually a good tradeoff is to have sector sizes between 256 to 1024 bytes.

ID And Data Preamble

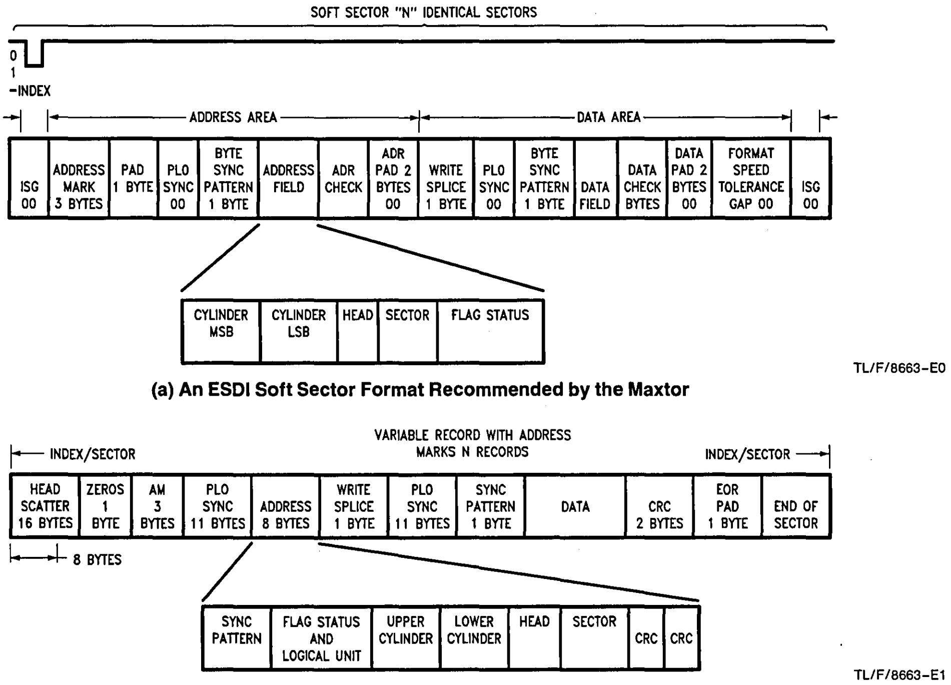
These field lengths are used to lock the data separator to incoming data during a read operation. The lengths of these fields determine the lock time requirements of the data separator. For individual ESDI and SMD drives, where the data separator is on the drive, the preamble should follow the manufacturers recommendations. For ST506 drives, and drives with imbedded controllers, the data separator is part of the controller and the preamble length should be set based on the separators performance goals.



(a) An EDSI Hard Sector Format Recommended by Maxtor

(b) An SMD Hard Sector Format Recommended by CDC

FIGURE 7.7 Hard Sector Formats



(a) An ESDI Soft Sector Format Recommended by the Maxtor

(b) An SMD Soft Sector Format Recommended by the CDC

FIGURE 7.8. Formats with Sector Mark

7.3.4 Formatting Methods

The disk formatting can be carried out using one of the three methods supported by the DDC; Internal Sequential, FIFO Table and Interlock Mode. These three disk formatting methods are explained in the following paragraphs and also summarized in a flow chart in *Figure 7.9.*

1. INTERNAL SEQUENTIAL METHOD

This method of disk formatting is adopted when sectors are to be physically contiguous. The DDC can be set for a multisector operation to format a whole track of sequential sectors. The steps required to perform the Internal Sequential method are explained in the following paragraphs. See *Figure 7.9.*

The DDC in Command Accept Mode

Step 1. All the Pattern and Byte Count registers for various sector format fields are loaded. All the Header byte's Pattern and Control registers (such as the one for cylinder number, head number etc.) are loaded except the one containing sector number information. The Pattern register for this header byte need not be loaded with anything, but the Control register should be loaded with 3H (i.e. SSC = 1 and HB = 1). With SSC = 1, the sector number for each sector will be loaded into this Header Byte Pattern register, automatically, from the Sector Counter.

Step 2. The Sector Counter is loaded with the first sector to be formatted. The contents of Sector Counter are loaded into the Header Byte Pattern register reserved for sector number, written to the disk, and then incremented for the next sector. The Number of Sector Operation (NSO) Counter is loaded with the number of sectors per track.

Step 3. The Disk Format register is loaded with FTF = 0, desired internal CRC/ECC appendage and other information (such as Hard/soft sector, NRZ/MFM data, etc.). The ECC/CRC control register (address 0EH) should also be loaded for desired options, such as inverting the serial data. The ECC polynomial and tap registers should be programmed if ECC is chosen.

Step 4. The Operation Command register is loaded to enable interrupts. Finally the Drive Command register is loaded with the Format Track command (i.e. ACH). Refer to Table 5.6 for various DDC commands.

The DDC in Command Perform Mode

Step 5. The DDC will start the operation when it receives an index pulse indicating the start of a track and will end the operation when it encounters another index pulse.

The DDC in Result/Error Mode

Step 6. An interrupt will be generated after a successful or unsuccessful execution of Format Track command, if the interrupts were enabled. The Status and (or) Error registers will indicate the result or the type of error occurred. In case of successful completion of formatting, the DDC could be initialized to format the next track and steps 1 thru 5 will be repeated. If an error has occurred, the interrupt should be serviced properly and the DDC should be reset. Refer to section 7.7 for interrupt servicing, and section 7.2.2 for resetting.

2. FIFO TABLE METHOD

This method is ideal if sector interleaving is required. The sectors may be written to the disk in any order using this method which offers the minimum amount of microprocessor involvement during the format operation. In this method the header bytes are written on the disk from the memory (via FIFO) instead of the Header Byte registers. This essentially eliminates the need of the microprocessor to update header bytes for each sector as could be done in the interlock mode. All other format pattern and count registers are loaded once by the microprocessor remain valid for the entire operation. The header bytes for each sector (with or without interleaving) are set up contiguously in sets as a table in the memory and then read by the DDC, one set for each sector, from the memory using the local DMA channel.

The steps required to perform the FIFO table method are explained below. Also see *Figure 7.9.*

Step 1. Sets of header bytes (one for each sector) for the entire track are stored contiguously in a memory area accessible to the local DMA. Each header byte set must contain an even number of bytes and start on an even byte boundary. If the header byte set contains an odd number of bytes, an extra dummy byte must be inserted at the end of each set so that each header byte set will start on an even byte boundry, for DMA considerations.

The DDC in Command Accept Mode

Step 2. Address of the first byte of the first header byte set is loaded in the DMA Address Byte (0, 1) registers.

Step 3. Sector Counter (SC), Number of Sector Operations (NSO) counter and Header Byte Count registers are loaded with initial sector number, number of sectors to be operated on, and number of header bytes (2–6 bytes), respectively.

Step 4. All other Format patterns and count registers (such as Preamble, Postamble, ECC/CRC etc.) according to the selected sector format are loaded with appropriate information. See *Figure 7.4.*

Step 5. The Desk Format (DF) register is loaded with the FTF bit set. The DF register is also loaded with MFM/NRZ, hard sector/soft sector, and ID, Data CRC/ECC appendage etc.

Step 6. The Operation Command (OC) register is loaded to enable interrupts. Finally, the Drive Command register is loaded with Format track command (ACH). See Table 5.6 for the DDC commands.

The DDC in Command Perform Mode

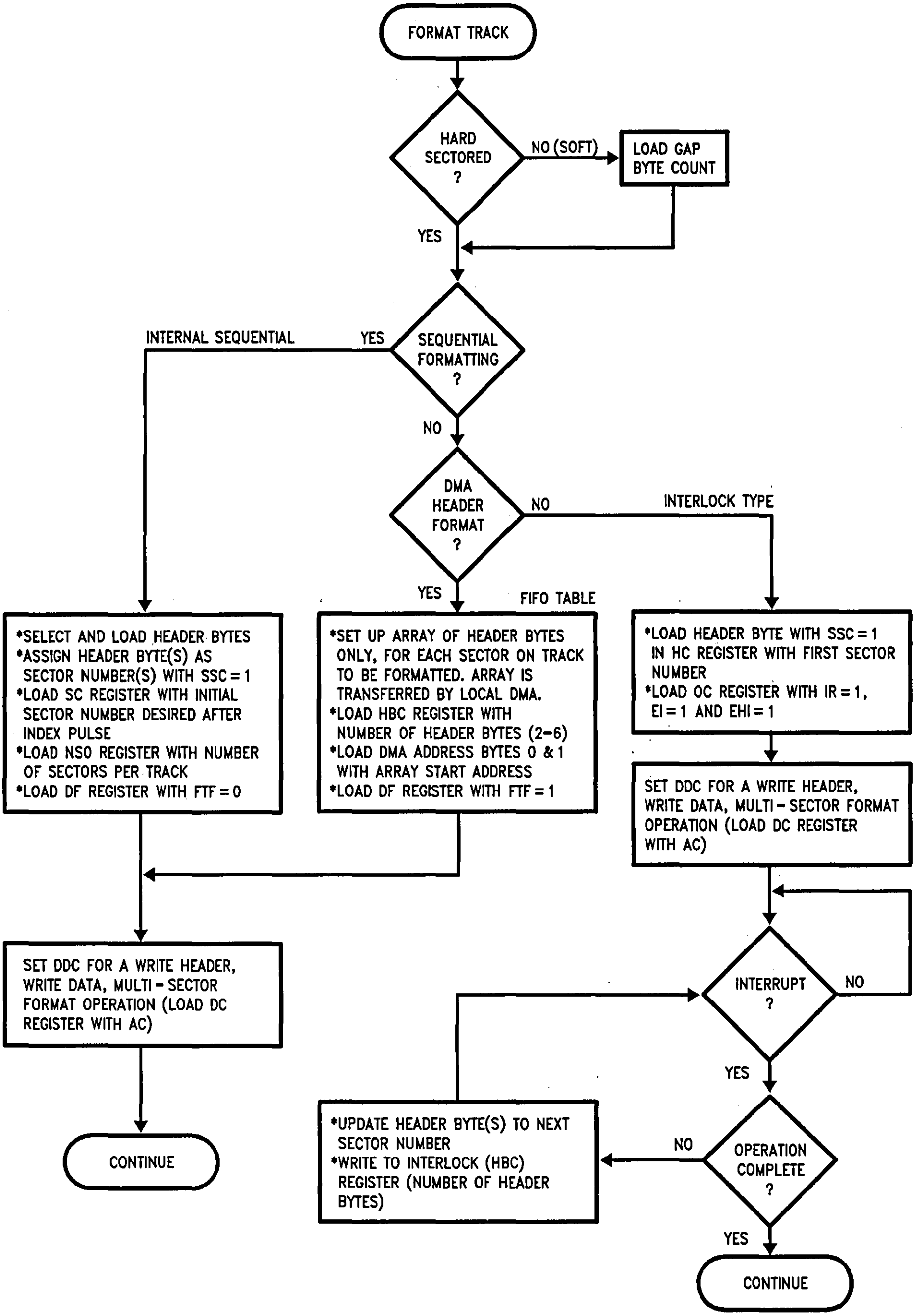
Step 7. The DDC starts the operation when it receives the index pulse and ends it on the occurrence of next index pulse. As the header bytes are needed they are DMA'd from memory. After a successful or unsuccessful completion of the operation, the DDC will generate an interrupt.

The DDC in Result/Error Mode

Step 8. The Status and (or) Error registers are read to find out the cause of interrupt. In case of a successful completion of the operation, steps 1 thru 7 may be repeated or the DDC may be initialized for another disk operation. In case of an error interrupt, the interrupt is serviced properly. See section 7.2.2 and 7.7.

3. INTERLOCK METHOD

This method is the most versatile of the three disk formatting methods. But it requires fast microprocessor involvement. This method may be used to format a whole track of interleaved sectors or a single sector. Using this method to reformat a single sector is ideal. Formatting single sectors is useful for remapping bad sectors. This method can also be used for creating tracks with varying sector field lengths.


FORMAT TRACK
HARD SECTORED ?
NO (SOFT)
LOAD GAP BYTE COUNT
YES
INTERNAL SEQUENTIAL
YES
SEQUENTIAL FORMATTING ?
NO
DMA HEADER FORMAT ?
NO
INTERLOCK TYPE
YES
FIFO TABLE
•SELECT AND LOAD HEADER BYTES
•ASSIGN HEADER BYTE(S) AS SECTOR NUMBER(S) WITH SSC = 1
•LOAD SC REGISTER WITH INITIAL SECTOR NUMBER DESIRED AFTER INDEX PULSE
•LOAD NSO REGISTER WITH NUMBER OF SECTORS PER TRACK
•LOAD DF REGISTER WITH FTF = 0
•SET UP ARRAY OF HEADER BYTES ONLY, FOR EACH SECTOR ON TRACK TO BE FORMATTED. ARRAY IS TRANSFERRED BY LOCAL DMA.
•LOAD HBC REGISTER WITH NUMBER OF HEADER BYTES (2-6)
•LOAD DMA ADDRESS BYTES 0 & 1 WITH ARRAY START ADDRESS
•LOAD DF REGISTER WITH FTF = 1
•LOAD HEADER BYTE WITH SSC = 1 IN HC REGISTER WITH FIRST SECTOR NUMBER
•LOAD OC REGISTER WITH IR = 1, EI = 1 AND EHI = 1
SET DDC FOR A WRITE HEADER, WRITE DATA, MULTI - SECTOR FORMAT OPERATION (LOAD DC REGISTER WITH AC)
SET DDC FOR A WRITE HEADER, WRITE DATA, MULTI - SECTOR FORMAT OPERATION (LOAD DC REGISTER WITH AC)
INTERRUPT ?
NO
YES
CONTINUE
•UPDATE HEADER BYTE(S) TO NEXT SECTOR NUMBER
•WRITE TO INTERLOCK (HBC) REGISTER (NUMBER OF HEADER BYTES)
NO
OPERATION COMPLETE ?
YES
CONTINUE

TL/F/8663-E6

FIGURE 7.9. Track Formatting Methods

The Interlock type disk formatting uses the interlock mode and header complete interrupt to enable the microprocessor to directly update any format parameter bytes. In a write header-write data operation, after the header bytes are written to the disk, the DDC issues the header complete interrupt. With interlock mode set and header complete interrupt issued, the controlling microprocessor has the time (until the preamble field of the next sector) to read status, load the next sector's header bytes, and write the interlock (HBC) register. Writing to the interlock (HBC) register confirms that microprocessor has updated all the required parameters. This must be done following each header match complete interrupt for every sector, including the last sector of the operation. If this is not done, the DDC assumes that the microprocessor did not properly update the parameter registers in time for the next sector and therefore generates an interrupt indicating the Late Interlock error when a subsequent command is loaded in the DC register. A 'step by step' procedure of the Interlock type formatting is explained below. Also see *Figure 7.9*.

DDC In Command Accept Mode

Step 1. All the format registers (shown in *Figure 7.4*) are loaded with respective pattern and count values. The header byte control register for the sector number is loaded with SSC = 1. This is done only for the first sector. Later on, depending upon the interleave factor, the header byte reserved byte for sector number may be loaded with a new value and the associated control register with SSC=0.

Step 2. The Sector Counter is loaded with the sector number to start with and NSO (number or sector operations) Counter with number of sectors to be formatted.

Step 3. In Disk Format register, FTF is set to zero and other relevent bits such as HSS, MFM, IH's and ID's are also set or reset.

Step 4. The DDC is set to be in Interlock Mode by setting IR = 1 in the Operation Command register. Also header complete and other interrupts are enabled by setting EHI = 1 and EI = 1 in the same register.

Step 5. Finally, the Drive Command register is loaded with the Format Track command, ACH. See Table 5.6 for various DDC commands.

DDC In Command Perform Mode

Step 6. The format operation starts when the DDC receives an index pulse. An interrupt is expected at the completion of the Write Header operation. When this interrupt is received it is tested for Command Complete Error or Header Complete Status Bits set. If it's a Header Complete, then parameter, count and control registers are updated. The Interlock register is written to. This is repeated until the Command Complete or Error Interrupt occurs.

DDC In Result/Error Mode

Step 7. On the occurrence of an interrupt, the Status register is read. If the interrupt was an operation complete interrupt then steps 1 through 5 are repeated for the rest of the sectors to be formatted without changing the contents of the NSO counter. In case of an error, the interrupt is serviced properly. See section 7.7 on interrupts.

7.4 READ AND WRITE OPERATIONS

Once the disk has been formatted, various disk read and write operations can be performed. These commands are listed in Table 7.5 and are discussed briefly in section 5.2.7.

Generally, the read operation is taken as reading data from the disk and can therefore be performed by executing the Read Sector (single or multi-sector) and Read Track commands. Similarly, write operation is considered as writing data to the disk and hence could be achieved by executing the Write Sector (single or multi-sector) and Write Track commands. Other read and write commands imply reading or writing ID with or without data from (to) the disk. The DDC programming procedure for all the read and write commands basically is the same.

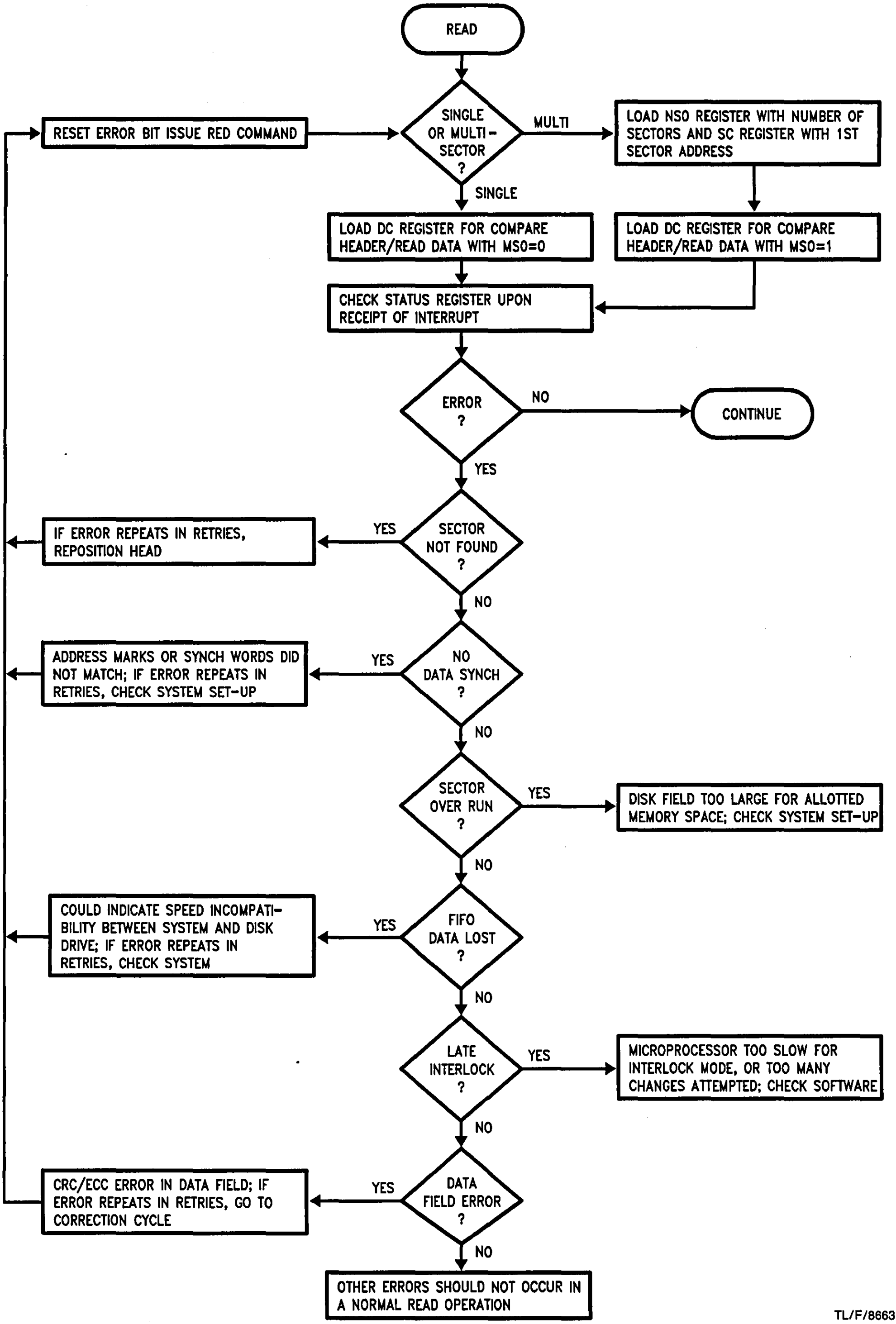
A general register programming procedure to perform read and write operations is given below.

Figure 7.10 shows a generalized flow chart for performing a read operation. To generalize both multi-sector and single sector operations, the continue block would go to the next sector operation if multi-sector, and would go to other μP tasks if single sector. Refer also to *Table 7.5* which shows the command codes for the various operations.

Command Name		Op Code	
Read Single Sector	RDSS	11010010	D2H
Read Sector ID	RDID	01110010	72H
Read Multi-Sector Logical	RDMS	11010110	D6H
Read Track	RDTK	11010100	D4H*
Read Track Blind	RDTB	11000100	C4H*
Read ID Multi-Sector	RDIM	01110100	74H
Read Track Data/ID	RDDI	11110100	F4H*
Write Single Sector	WRSS	10010010	92H
Write Multi-Sector Logical	WRMS	10010110	96H
Write Track	WRTK	10000100	84H*
Format Track	FMTK	10101100	ACH
Format Track No Gap	FMNG	10100100	A4H
Find ID	FNID	01010010	52H
Find ID Multi-Sector	FNMS	01010110	54H
Recover Header	RCID	01100010	62H
Re-Enable Controller	RENB	00000001	01H
No Operation	NOP	00000000	00H

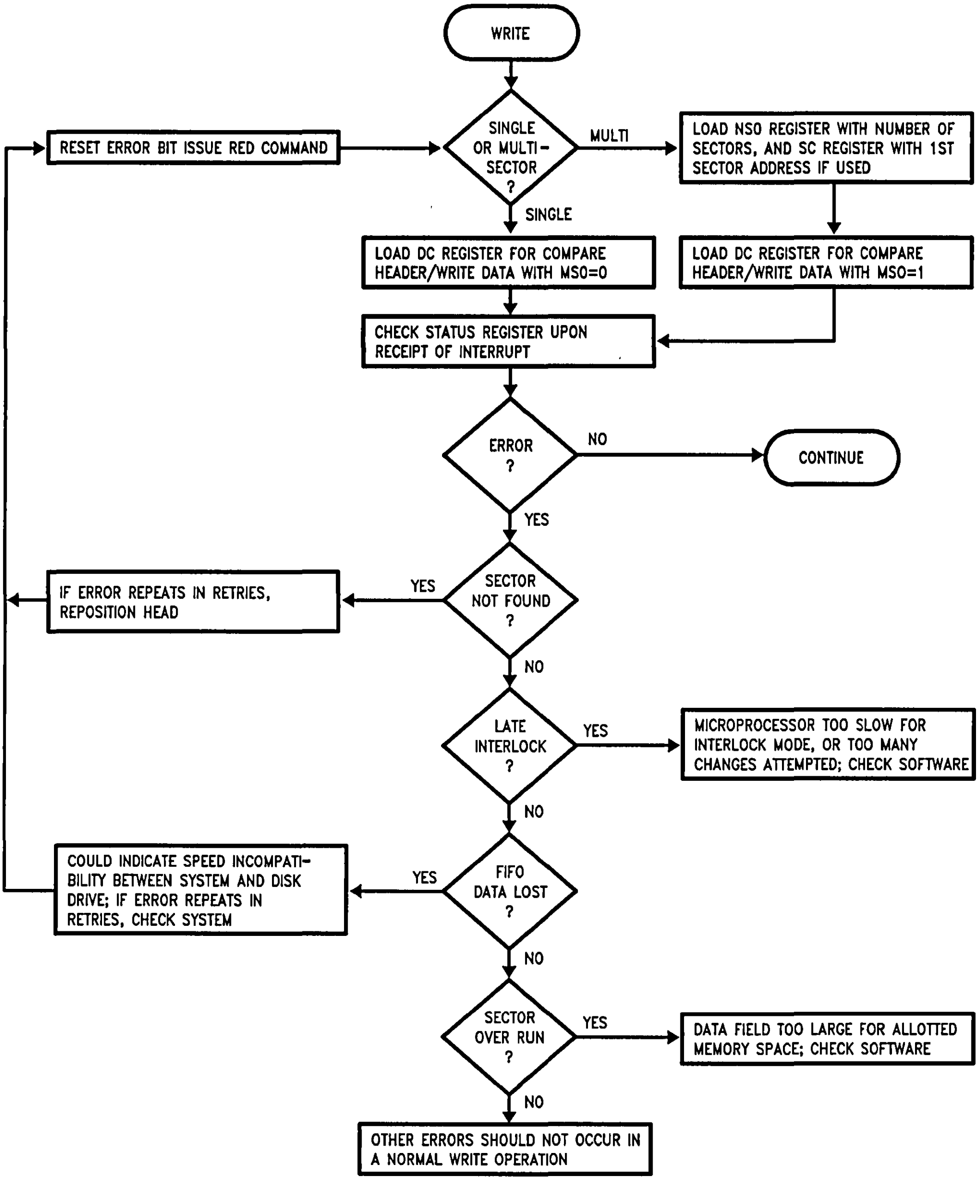
*Note: For an entire track operation, the Number of Sector Operations Counter should be set to the number of sectors per track.

Table 7.5. Common Configurations of the Command Bits


READ
SINGLE OR MULTI-SECTOR ?
MULTI
LOAD NSO REGISTER WITH NUMBER OF SECTORS AND SC REGISTER WITH 1ST SECTOR ADDRESS
RESET ERROR BIT ISSUE RED COMMAND
SINGLE
LOAD DC REGISTER FOR COMPARE HEADER/READ DATA WITH MSO=0
LOAD DC REGISTER FOR COMPARE HEADER/READ DATA WITH MSO=1
CHECK STATUS REGISTER UPON RECEIPT OF INTERRUPT
ERROR ?
NO
CONTINUE
YES
SECTOR NOT FOUND ?
YES
IF ERROR REPEATS IN RETRIES, REPOSITION HEAD
NO
NO DATA SYNCH ?
YES
ADDRESS MARKS OR SYNCH WORDS DID NOT MATCH; IF ERROR REPEATS IN RETRIES, CHECK SYSTEM SET-UP
NO
SECTOR OVER RUN ?
YES
DISK FIELD TOO LARGE FOR ALLOTTED MEMORY SPACE; CHECK SYSTEM SET-UP
NO
FIFO DATA LOST ?
YES
COULD INDICATE SPEED INCOMPATIBILITY BETWEEN SYSTEM AND DISK DRIVE; IF ERROR REPEATS IN RETRIES, CHECK SYSTEM
NO
LATE INTERLOCK ?
YES
MICROPROCESSOR TOO SLOW FOR INTERLOCK MODE, OR TOO MANY CHANGES ATTEMPTED; CHECK SOFTWARE
NO
DATA FIELD ERROR ?
YES
CRC/ECC ERROR IN DATA FIELD; IF ERROR REPEATS IN RETRIES, GO TO CORRECTION CYCLE
NO
OTHER ERRORS SHOULD NOT OCCUR IN A NORMAL READ OPERATION

TL/F/8663–E7

FIGURE 7.10. A Flow Chart for Microprocessor Initiation and Servicing of a Read Operation


WRITE
SINGLE OR MULTI-SECTOR ?
MULTI
SINGLE
RESET ERROR BIT ISSUE RED COMMAND
LOAD NSO REGISTER WITH NUMBER OF SECTORS, AND SC REGISTER WITH 1ST SECTOR ADDRESS IF USED
LOAD DC REGISTER FOR COMPARE HEADER/WRITE DATA WITH MSO=0
LOAD DC REGISTER FOR COMPARE HEADER/WRITE DATA WITH MSO=1
CHECK STATUS REGISTER UPON RECEIPT OF INTERRUPT
ERROR ?
NO
CONTINUE
YES
SECTOR NOT FOUND ?
YES
IF ERROR REPEATS IN RETRIES, REPOSITION HEAD
NO
LATE INTERLOCK ?
YES
MICROPROCESSOR TOO SLOW FOR INTERLOCK MODE, OR TOO MANY CHANGES ATTEMPTED; CHECK SOFTWARE
NO
FIFO DATA LOST ?
YES
COULD INDICATE SPEED INCOMPATIBILITY BETWEEN SYSTEM AND DISK DRIVE; IF ERROR REPEATS IN RETRIES, CHECK SYSTEM
NO
SECTOR OVER RUN ?
YES
DATA FIELD TOO LARGE FOR ALLOTTED MEMORY SPACE; CHECK SOFTWARE
NO
OTHER ERRORS SHOULD NOT OCCUR IN A NORMAL WRITE OPERATION

TL/F/8663-E8

FIGURE 7.11. A Flow Chart for Microprocessor Initiation and Servicing of a Write Operation

SINGLE SECTOR READ AND WRITE OPERATIONS

This operation involves setting the Disk Command register to perform a Compare Head/Read Data Operation. Only one sector is transferred.

DDC In Command Accept Mode

Step 1. The Parameter and Count registers must be loaded with the proper format information (if not already loaded with the correct information). The Header byte pattern registers must be loaded with the exact header information for the sector to be transferred.

Step 2. The Local DMA channel must be initialized. The Local transfer register should be configured to the desired DMA mode. The DMA address registers are loaded with the data transfer address. If the Remote DMA is used in coordination with the local transfer, especially in tracking mode, the Remote Transfer register should be initialized, as should its address, and length information. Interrupts should be enabled, in order to enable determination of operation completion.

Step 3. Finally the Drive Command register is loaded with the desired Read or Write Command. If a remote DMA operation is to be performed, the Operation Command register should be loaded to start the remote DMA.

DDC In Command Perform Mode

Step 4. The DDC will perform the operation by looking for the correct sector header, then acquiring the system bus, and transferring the appropriate data.

DDC In Result/Error Mode

Step 5. At the end of the command an interrupt is generated. If the interrupt is the Operation Complete interrupt the command terminated properly, and the μP can read this from the Status and Error registers and can proceed to the next command. If an error occurred the microprocessor should service this by either retrying the operation, or correcting the error condition.

MULTI SECTOR LOGICAL READ AND WRITE OPERATIONS

Multi sector logical operations transfer sectors sequentially based on their sector number, whether these sectors are scattered around the disk or not. Multi sector operations use the Number of Sector Operations Counter to determine the number of sectors to transfer. Logical operations are most typically done by using the Sector Counter.

DDC In Command Accept Mode

Step 1. The Parameter and Count registers must be loaded with the proper format information (if not already loaded with the correct information). The Header byte pattern registers must be loaded with the exact header information for the sector to be transferred. The header byte that contains the sector number must have the Sector Counter substituted for it. The Sector Counter is set to the number of the first sector to be transferred. The Number of Sector Operations Counter should be loaded with the number of sectors to be transferred.

Step 2. The Local DMA channel must be initialized. The Local transfer register should be configured to the desired DMA mode. The DMA address registers are loaded with the data transfer address. If the Remote DMA is used in coordination with the local transfer, especially in tracking mode, the Remote Transfer register should be initialized, as should its address, and length information. Interrupts should be enabled, in order to enable determination of operation completion.

Step 3. Finally the Drive Command register is loaded with the desired Logical Read or Write Command. If a remote DMA operation is to be performed, the Operation Command register should be loaded to start the remote DMA.

DDC In Command Perform Mode

Step 4. The DDC will perform the operation locating the first sector, acquiring the system bus, and transferring that sector's data. The Number of Sector Operations Counter is decremented, and the Sector Counter is incremented. Step 4 is repeated until the Number of Sector Operations Counter reaches zero. The command may be terminated early if the DDC could not find one of the correct sectors being sought.

DDC In Result/Error Mode

Step 5. At the end of the command when the NSO counter equals zero, an interrupt is generated. If the interrupt is the Operation Complete interrupt the command terminated properly, and the μP can read this from the Status and Error registers and can proceed to the next command. If an error occurred the microprocessor should service this by either retrying the operation, or correcting it.

MULTI SECTOR PHYSICAL READ AND WRITE OPERATIONS

These operations are very similar to the logical read and write commands, except that rather than looking for the sectors in numerical order, they are read from the disk in the exact order that they pass under the read/write head. Only the differences between these and the logical commands are described.

DDC In Command Accept Mode

The only difference in setting up for the command is that since the basic command will ignore the header bytes for comparison (Ignore Header-Read/Write Data) the Header pattern registers do not need to be updated. If doing a track operation, the command should start on an index pulse by setting the SAIS bit, and the Number of Sector Operations Counter should be loaded with the number of sectors on the track.

DDC In Command Perform Mode

This is the same as previous operation.

DDC In Result/Error Mode

This is the same as the previous operation.

OTHER MULTI-SECTOR PHYSICAL OPERATIONS—INTERLOCK MODE

The host microprocessor can perform many related but different operations on physically consecutive sectors, by using the Interlock mode. Several possible command sequences are briefly outlined:

1. Header Re-writing—If an ID is detected to have an error in its header bytes a sequence of commands can be executed to re-write the header. The rewritten ID may attempt to fix the error or mark the sector as bad. This involves doing two commands in sequence:

a) First find the header of the sector located physically prior to the Bad ID field.

b) Second, do a write header operation to re-write the ID field.

2. Recovery of Data—This is like one above except no attempt is made to correct the ID field, just obtain the data field information.

a) First find the header of the sector located physically prior to the defective sector.

b) Second, do an ignore header operation to The data field is to be recovered, so a read data operation should be executed, and the data can be read to memory.

3. Sparing a defective sector—This is a more complicated version of 1 and 2 above, and is performed with the following commands. Steps a and b and steps c and d should be performed in the interlock mode, whereas between steps b and c interlock mode may not be desirable since ECC correction may be necessary.

a) First find the header of the sector located physically prior to the defective sector.

b) Second, do a write header operation to re-write the ID field, indicating a bad sector. If the data field is to be recovered the write header operation could be accompanied by a read data operation, and the data can be read to memory.

c) Third the DDC does a compare header operation to find the sector prior to the spare sector (if located on the same track). (If the spare has a known header a Compare Header is all that is necessary.)

d) Finally the spare sector ID is rewritten and the old sector's data is written to the spare sector.

In general, Interlocked mode operation is required only when unique operations on physically adjacent sectors is necessary. General multi-sector operations not performed on adjacent sectors can be cascaded without using the interlock mode.

DDC In Command Accept Mode

Step 1. The Parameter and Count registers must be loaded with the proper format information (if not already loaded with the correct information). The Header byte pattern registers must be loaded with the exact header information for the sector to be that will be operated on. Many Interlock mode command sequences may not need the Number of Sector Operations Counter loaded, but if needed it should be loaded with the number of sectors to be transferred.

Step 2. The Local DMA channel must be initialized. The Local transfer register should be configured to the desired DMA mode. The DMA address registers are loaded with the data transfer address. If the Remote DMA is used in coordination with the local transfer, especially in tracking mode, the Remote Transfer register should be initialized, as should its address, and length information. The header complete and command complete interrupts should be enabled, in order to enable determination of when to load the subsequent sector's information and command. This is done by setting the EI, EIH, IR bits in the Operation Command Register.

Step 3. Finally the Drive Command register is loaded with the desired first Command. If a remote DMA oeration is to be performed, the Operation Command register should be loaded to start the remote DMA.

DDC In Command Perform Mode

Step 4. The DDC will perform the operation locating the first sector, acquiring the system bus, and transferring that sector's data. As soon as the first sector's header has been located, then the DDC issues a header complete interrupt, and the host must update all desired registers, and finally loading the Drive Command register with the new information. (Note: if the interlocked operation was a multi-sector (NSO not equal 0) operation then the Drive command is not updated, since the operation is a continuation of the multi-sector operation.) Finally the Interlock Register is written to, and step 4 is repeated until the microprocessor is done writing commands. During the last operation, the Interlock register must be written to avoid a late interlock error.

The command may be terminated early if the DDC could not find one of the correct sectors being sought, or the Interlock register is not written to prior to the beginning of the next sector.

DDC In Result/Error Mode

Step 5. At the end of each header field an interrupt is issued. The operation has completed when the Header Interrupt and Operation Complete is received after the last command. If the command terminated properly, and the μP can read this from the Status and Error registers and can proceed to the next command. If an error occurred the microprocessor should service this by either retrying the operation, or correcting the problem.

GENERAL CHAINING OF DISK COMMANDS

Various commands can be executed one right after the other, like the interlock mode described above, except without the Interlock timing constraints. This is done by enabling the Header Complete Interrupt executing a disk command, and when the interrupt is received by the CPU, it checks to make sure it is the Header Complete interrupt, and that the Next Disk Command bit is set. If it is set the CPU can then execute a new operation, while the old operation is completing. In this way fast access to the data on the disk track can be achieved.

7.5 DMA OPERATIONS

In this section, the DDC's data transfer operations using on-chip or external DMA will be discussed in depth. Discussion of DMA as part of the above disk operations has been omitted in favor of a separate discussion. It is important for the designer to keep in mind that while the type of DMA operation won't affect the individual commands, it can be a very important factor in overall system through-put, and bus utilization. In general, once a disk sector buffering scheme and method of transferring data to/from the system has been designed, the DMA mode selection is obvious, and usually remains fixed. The DDC-system interface connections for different system applications are discussed in chapter 6.

7.5.1 Data Transfer Features

All DMA operations are supported by the following four features. These features are valid for all types of DMA modes described in section 7.5.2 including the Slave mode (external DMA).

PROGRAMMABLE BURST LENGTHS

The data transfer from/to the DDC to/from the system is fully programmable. In single bus systems, the data from/to the FIFO to/from the memory, can either be transferred in 32-byte bursts or in smaller bursts of 2, 8, 16 or 24 bytes (or 1, 4, 8 or 12 words). In dual bus systems, data can be transferred either up to 64 Kbytes in a single operation or in smaller operations. The programmable burst lengths feature accommodates the variations in bus latency time usually present in all systems (see Chapter 6).

The DDC is programmed for the desired data transfer mode through LTEB, LBL1, and LBL2 bits in the Local Transfer Register and the RTEB, RBL1, RBL2 bits in the Remote Transfer Register. For Remote transfers, the DMA Byte Count registers are also used.

8-BIT OR 16-BIT WIDE TRANSFERS

Data can be transferred either byte wide or word wide. This is achieved through LWDT and RWDT bits in the Local and Remote Transfer Registers, respectively. The DMA address counters are incremented by one for byte wide and by two for word wide transfers.

SLOW READ/WRITE

For slow memory or other devices, the normal DMA memory read/write cycle of four periods can be extended to five cycles for all DMA modes (including external DMA), using bit LSRW and bit RSRW in the Local and Remote Transfer Registers respectively. The read/write cycles can also be extended to an infinite length by using the External Status input (pin 17) of the DDC in conjunction with EEW bit in the Remote Transfer Register.

REVERSE BYTE ORDER

This option is only valid for 16-bit wide transfers using the Local DMA channel. It enables the two bytes being transferred to be mapped with the high order byte to AD0–7 and the low order byte to AD8–15, or vice-versa. This could be achieved through RBO bit in the Local Transfer Register.

(Note: This option is still functional in 8 bit mode, however it performs no useful function. When reading, the first byte DMA'd was the second byte read, the second DMA'd byte the first read, the third DMA'd byte the fourth, the fourth DMA'd byte the third, and so on. Similar order occurs for a write.)

7.5.2 DMA Modes

Various data transfers are carried out by configuring the DDC in one of the three main DMA modes: single channel, dual channel or external DMA. Some of this has been discussed in Chapter 6.

SINGLE CHANNEL (LOCAL DMA) MODE

In the local DMA mode, only three DMA registers/counters are used; the Local Transfer Register, DMA Address Byte 0 and 1. The Sector Byte Count 0 and 1 registers determines the sector size. A local transfer operation can be carried out following the steps below:

DDC in Command Accept Mode

Step 1. The DMA Address Bytes (0 and 1) Counters are initialized with local (or main) memory address to/from where the data is to be transferred from/to the FIFO.

Step 2. The Local Transfer Register is set for enabling the local DMA channel (bit SLD), 8- or 16-bit transfer (bit LDWT), reverse-byte order (optional in 16-bit data transfer mode, using bit RB0). Slow Read/Write cycles (bit LSRW), Long Address (bit LA), and the burst length (bits LTEB, LBL1, LBL2).

Step 3. The Operation Command Register is loaded for enabling interrupts, if desired (bit EI).

Step 4. Finally, the Drive Command (DC) Register is loaded with the desired DDC command. See Table 7.5. The DDC enters the command perform mode immediately after the DC register is loaded. (This step is the same as Step 3 in previous read/write operations discussions.)

DDC in Command Perform Mode

Step 5. The DDC should be granted bus control on the occurrence of an LRQ. The DDC will generate an interrupt after the completion of the operation or on the occurrence of an error. (Same as previous read/write operations Step 4.)

DDC In Result/Error Mode

Step 6. On the occurrence of an interrupt, the Status Register is read. In case of an operation complete (the NDC bit), steps 1 through 6 may be repeated, or the DDC may be initialized for a new operation. If an error was occurred, appropriate actions should be taken, as discussed in section 7.7.

DUAL CHANNEL TRACKING (LOCAL AND REMOTE) MODE

In the dual DMA mode, all the DMA registers are used (see Chapter 5). The DDC can further be set for either a Tracking Dual DMA or a NON-Tracking Dual DMA mode.

In Tracking mode, data is transferred from the on-chip FIFO to the system I/O port through the local buffer memory, and vice versa. The entire DMA operation is controlled by the DDC and external arbitration logic that synchronizes the DDC's remote operation with the external DMA for the system, after initialized by the microprocessor. Basically, local and remote transfers are dependent on each other and the DDC keeps track of both transfers in order to avoid any possible data overlapping in the local buffer memory.

This mode effectively turns the buffer memory into a large FIFO. This is accomplished through the use of DMA Sector Counter (DSC), which keeps track of the difference between sectors read/written from/to the disk and sectors transferred to/from the host system. Each time the source transfers a sector or data into buffer memory (length is determined by the Sector Byte Count Register pair), the DSC register is incremented. It is decremented each time the destination has transferred a sector of data. Whenever the DSC register contents become zero, destination transfers are inhibited. Note that in a Disk Read operation, local DMA (or the FIFO) is the source and remote DMA (or system I/O port) is the destination. Similarly, in a Disk Write operation, the remote DMA (or the system I/O port) is the source and local DMA (or the FIFO) is the destination. A detailed step-by-step disk read/write operation, in the tracking dual DMA mode, is given below and the flow chart is given in *Figure 7.12.*

DDC In Command Accept Mode

Step 1. The DMA Address Bytes (0, 1, 2, 3) registers are loaded with the same local and remote start address in the local buffer memory.

Step 2. Bits SLD and SRD in the Local and Remote Transfer Registers are set to enable both local and remote DMA channels. The DDC is configured for Tracking Mode by setting the TM bit in the Remote Transfer Register. Other options such as 8-/16-bit transfer, slow read/write cycles, and burst lengths for both FIFO and local buffer memory are selected through LWDT, LSRW, LTEB, LBL1 and LBL2 bits in the Local Transfer Register and RWDT, RSRW or EEW, RTEB, RBL1 and RBL2 bits in the Remote Transfer Register. Bit LA in the Local Transfer Register must be reset for 16-bit address mode.

Step 3. The Number of Sector Operations (NSO) counter and Sector Counter (SC) are loaded for multi-sector operation. Only SC should be loaded for a single sector operation.

Step 4. Interrupts are enabled using EI bit in the Operation Command (OC) Register.

Step 5. Finally, the desired Read/Write command (see Table 7.6). (This is the same as Step 3 in the previous Read/Write command descriptions.)

DDC In Command Perform Mode

Step 6. The DDC will start performing the desired operation after the DC register is loaded. An LRQ, RRQ or Interrupt should be expected. The DDC should be given the bus control when LRQ or RRQ occurs. (Same as Step 4 in previous Read/Write command descriptions.)

DDC In Result/Error Mode

Step 7. If an interrupt is generated, it should be serviced properly (see section 7.7 for interrupt servicing). If the operation was completed successfully, steps 1 through 6 may be repeated for a new operation. (Same as Step 5 in previous Read/Write command descriptions.)

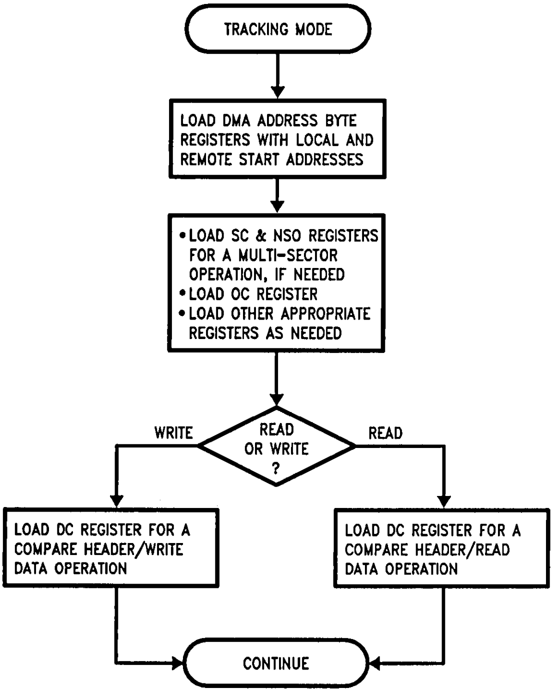
DUAL DMA NON-TRACKING (LOCAL AND REMOTE) MODE

In the non-tracking dual channel DMA mode, the Local and Remote transfers are independent of each other. The controlling microprocessor has to keep track of both transfers to avoid any possible data overlapping in the local buffer memory. The DMA Address (bytes 0–3) Registers are set up independently for Local and Remote transfers. All the necessary steps needed to perform a data transfer between the FIFO and system I/O port with the DDC in this mode are explained below and also shown in a flow chart in *Figure 7.16*.

DDC In Command Accept Mode

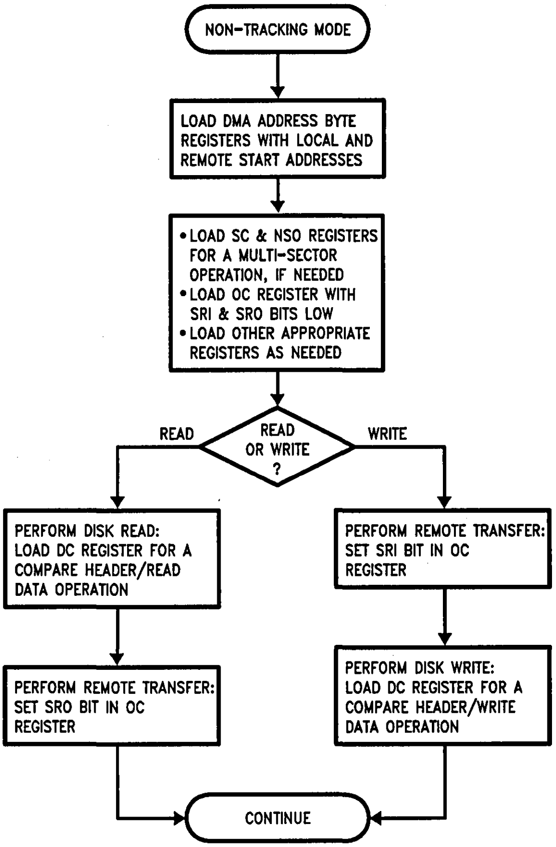
Step 1. The DMA Address (byte 0–3) are set up for the desired Local and Remote addresses. Any address within 64k memory space could be loaded.

Step 2. The DDC is configured in the non-tracking mode, first by enabling Local and Remote DMA channels through SLD and SRD bits in the Local and Remote transfer Registers. The LA bit is also set to zero for 16-bit address for both DMA channels. Other DMA options such as 8-/16-bit data transfer slow read/write, reverse byte ordering, and burst length, are selected through LWDT and RWDT; LSRW, RSRW and EEW; RBO; and LTEB, LBL1, LBL2, RTEB, RBL1, and RBL2 bits in the Local and Remote Transfer Registers.


TRACKING MODE
LOAD DMA ADDRESS BYTE REGISTERS WITH LOCAL AND REMOTE START ADDRESSES
• LOAD SC & NSO REGISTERS FOR A MULTI-SECTOR OPERATION, IF NEEDED
• LOAD OC REGISTER
• LOAD OTHER APPROPRIATE REGISTERS AS NEEDED
READ OR WRITE ?
WRITE
READ
LOAD DC REGISTER FOR A COMPARE HEADER/WRITE DATA OPERATION
LOAD DC REGISTER FOR A COMPARE HEADER/READ DATA OPERATION
CONTINUE

TL/F/8663-E9

FIGURE 7.12. Dual DMA Tracking Mode μP Programming Flow Chart


NON-TRACKING MODE
LOAD DMA ADDRESS BYTE REGISTERS WITH LOCAL AND REMOTE START ADDRESSES
• LOAD SC & NSO REGISTERS FOR A MULTI-SECTOR OPERATION, IF NEEDED
• LOAD OC REGISTER WITH SRI & SRO BITS LOW
• LOAD OTHER APPROPRIATE REGISTERS AS NEEDED
READ
READ OR WRITE ?
WRITE
PERFORM DISK READ: LOAD DC REGISTER FOR A COMPARE HEADER/READ DATA OPERATION
PERFORM REMOTE TRANSFER: SET SRI BIT IN OC REGISTER
PERFORM REMOTE TRANSFER: SET SRO BIT IN OC REGISTER
PERFORM DISK WRITE: LOAD DC REGISTER FOR A COMPARE HEADER/WRITE DATA OPERATION
CONTINUE

TL/F/8663-F0

FIGURE 7.13. Dual DMA Non-tracking Mode μP Programming Flow Chart

Step 3. The Number of Sector Operation (NSO) Counter and Sector Counter (SC) are loaded for multi-sector operation. Only SC is loaded for a single sector operation.

Step 4. Write Operation: Before having the DDC to perform a write operation, data is transferred from the system I/O port to the buffer memory by enabling the Remote channel through SRI bit in the Operation Command Register. Interrupts are also enabled using the EI bit.

Read Operation: The Drive Command (DC) Register is loaded with the desired Read command (refer to Table 5.6). If a multisector operation is selected, the SAIS bit may be set to zero for the operation to start at the Index pulse.

DDC Performs the Remote Transfer

Write Operation: The DDC will transfer the remote data to local memory and will issue an operation complete interrupt. Remote transfer operations could be repeated to fill the local memory before performing a disk operation. The DDC now should be initialized for the actual disk write operation.

Read Operation: The DDC will complete the disk read just like a normal operation. See Step 7. Now the data may be transferred to the system I/O or any remote locations.

Step 5. Finally, the Operation Command Register is loaded to enable data transfer from the local buffer memory to the system I/O, with SRO bit set. The interrupts may also be enabled with the EI bit, if required.

The DDC will start the operation when the OC Register is loaded. The RRQ must be acknowledged.

Step 6. The Drive Command register is loaded with the desired write command. If a multi-sector operation is desired, the SAIS bit may be reset for the operation to start at the Index pulse. (This is the same as Step 3 for Read/Write operations discussed previously.)

DDC In Command Perform Mode

Step 7. The DDC will start performing the desired write operation immediately and will issue a local request, LRQ. Upon receiving an LACK, it completes the write operation. (This is the same as Step 4 for previously discussed Read/Write operations.)

DDC In Result/Error Mode

Step 8. The DDC will issue an interrupt which should be serviced properly (refer section 7.7 for interrupt servicing). In case of an operation complete interrupt, steps 1 through 5 may be repeated for a new operation.

IMPORTANT NOTES

1. By setting both SRI and SRO simultaneously, any non-tracking DMA operation will stop. The current remote address and remote data byte count will be retained, and the local DMA will be unaffected. Loading the original OC instruction (input or output) will restart the original instruction from the last remote DMA address.

2. In either Tracking or Non-tracking mode, if either channel is loaded with an odd byte transfer count, the DDC will transfer the next higher even byte. For example, if 511 was loaded in Remote Data Byte Count Registers, 512 bytes would be transferred, with the valid data only in the first 511 bytes.

3. In the Tracking mode the DDC keeps track of the data in the buffer memory. The Remote Transfer follows the Local transfer by a sector length and the DDC makes sure that the correct data is transferred to the system memory. However in the Non-tracking mode the remote channel is independent of the disk operation and hence the remote channel can follow the local channel as closely as possible. The microprocessor is responsible of preventing overlap of data.

4. Even though normally the remote channel would be used for transfers from system to buffer memory (and vice versa) and the local channel for transfers from the buffer memory to the FIFO (and vice versa), the remote channel could also be used for some other purpose that is independent of the DDC's other operations.

EXTERNAL DMA

In external DMA mode, the data transfer between the on-chip FIFO and external memory (local buffer or system) is controlled by the external DMA. The DDC is programmed to perform a disk read/write operation without the internal DMA. Whenever the FIFO needs any data transfer, the DDC asserts LRQ. At this point, external DMA takes control and completes that particular data transfer. The following steps illustrate the necessary actions to perform a disk read/write operation using external DMA i.e. DDC in the slave mode.

DDC In Command Accept Mode

Step 1. The registers are initialized.

Step 2. Interrupts are enabled using EI bit in the Operation Command Register.

Step 3. Finally, the Drive Command Register is loaded for the desired Read/Write operation.

DDC In Command Perform Mode

Step 4. The DDC will start performing the operation and LRQ will be asserted when the FIFO requires the data transfer. The LRQ must be acknowledged by the external DMA in order to complete the operation.

DDC In Result/Error Mode

Step 5. If an interrupt is issued, it must be serviced, (refer section 7.7). In case of an operation complete interrupt, steps 1 through 3 may be repeated for a new operation.

7.6 ERROR DETECTION AND CORRECTION

The Disk Data Controller, DDC has comprehensive and versatile error detecting and correcting capabilities. It features a fully programmable ECC;

- Programmable Preset Pattern
- Programmable Polynomial Taps
- Programmable Correction Spans
- Programmable Assignment of CRC/ECC on Header or Data

There are essentially two internal codes available; a fixed Cyclic Redundancy Checking (CRC) code for detecting errors only, which uses a CRC-CCITT polynomial that provides 16 generated check bits for appending to the Header fields and/or Data fields. The other type is the ECC code which may be a Fire code or a Computer generated code with 32 or 48 generated check bits that may be appended to the Header field and Data field. National Semiconductor recommends a computer generated polynomial called the Glover 140A0443 code with a correction span of 5-bits for MFM encoded drives. The designation represents the hexadecimal equivalent of the forward polynomial and it requires a preset of all 1's. The code has two polynomials; the forward one for checkbit generation and checking and the reverse one for error location. The error detection span is 32-bits while the correction span is 5-bits. The number of bytes in the sector determines the integrity of the code. The maximum sector length the code can handle is 1024 bytes of data and 4 bytes of ECC, which is within the limits of most disk formats. The completely programmable feature of the DDC with respect to ECC offers a lot of flexibility to the user. In case a user prefers to use his own high integrity code, the DDC can be configured to interface easily with external ECC circuitry and the DDC can be programmed to operate in the external ECC mode, as discussed in chapter 4. There are essentially three kinds of operations associated with the ECC circuitry: 1) checkbit generation, 2) checkbit verification, and 3) error location.

7.6.1 Error Detection

Internal Checkbit Verification—Write

This operation occurs when the controller is performing a write operation. The ECC shift register is downloaded with the preset value stored in the Preset Register. The code length is selected independently for Header and Data appendage. Bits being shifted out of the SERDES (serializer/deserializer) to the disk, are also shifted into the ECC Shift Register. When the last bit of the Header or Data field has been transmitted out of the DDC, the generated check bits in the ECC shift register are directly shifted out and onto the disk, starting with the MSB and ending with bit 0. After the ECC bits have been appended, the DDC switches to the next field.

Internal Checkbit Verification—Read

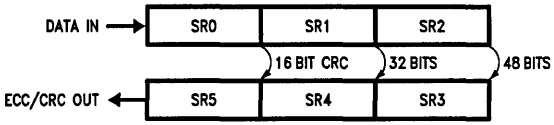
This function will occur concurrently with a read data operation from the disk. The ECC Shift Register is first preset from the Preset Registers. The incoming Header or Data field is serially fed into the same ECC Shift Register that is used to generate checkbits. When all the Header or Data field bits and all the generated checkbits have entered the ECC Shift Register, the status of the bits in it is checked for an all zeroes condition. If it is true then the field contains no errors, else if any of the ECC Shift Register's bits are high, the field contains an error. In the case of a Header field error, the Header Fault bit (SO), in the Status Register is set, while in case of a Data field error, the Data Field Error bit (E1), of the Error Register is set.

System Alternatives on Error Detection

Once an error has been detected by the ECC logic, the Re-enable (REN) bit must be reset via the Drive Command Register, before proceeding. If a Header field error is detected, the DDC will react differently depending on the Header operation involved. The various options are discussed under the Drive Command Register description. If an error is detected in the Data field, a re-read is initiated (by the system), to overrule a soft error. If the data is still not corrected after several re-reads, it implies the detection of a hard error. By re-reading the sector in question and comparing the syndromes to previous retries, a certain level of confidence can be reached, that the error is media induced and ECC correction can be attempted. The syndrome bytes in the ECC Shift Register will contain the bit error information, although the bytes in error have been transferred to memory.

7.6.2 Error Correction

The DDC has a maximum correction span of 15 bits, i.e. it can correct up to 15 contiguous bits in error, or a span of errors 15 bits or less. Of course correction can only be attempted if internal ECC checkbits were appended to the data field when written to the disk. The first step in the correction process is to load the Data Count Register with the data count, (sector byte count) plus the number of bytes of checkbits, i.e. sector byte count registers must be initialized to sector length plus 4 or 6 for 32 bit mode or 48 bit mode ECC respectively. Then the correction cycle is initiated by setting the Start Correction Cycle bit of the Operation Command Register. This should be done before any further Drive Command operation is issued to the DDC. This prevents the destruction of the stored syndromes in the ECC Shift Register. Also while the correction cycle is in progress,



TL/F/8663-F1

FIGURE 7.14(a). Hardware Configuration of ECC Shift Register

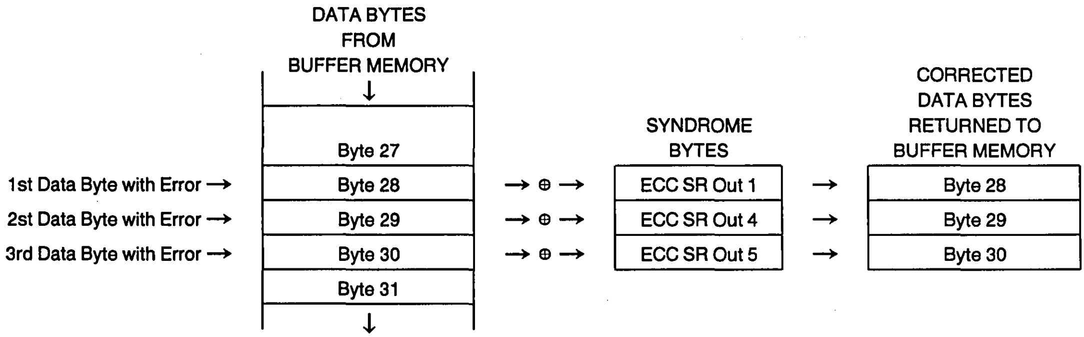

DATA BYTES FROM BUFFER MEMORY
Byte 27
1st Data Byte with Error →
Byte 28
2st Data Byte with Error →
Byte 29
3rd Data Byte with Error →
Byte 30
Byte 31
→ ⊕ →
SYNDROME BYTES
ECC SR Out 1
ECC SR Out 4
ECC SR Out 5
→
CORRECTED DATA BYTES RETURNED TO BUFFER MEMORY
Byte 28
Byte 29
Byte 30

FIGURE 7.14(b). 32-Bit ECC Correction Process

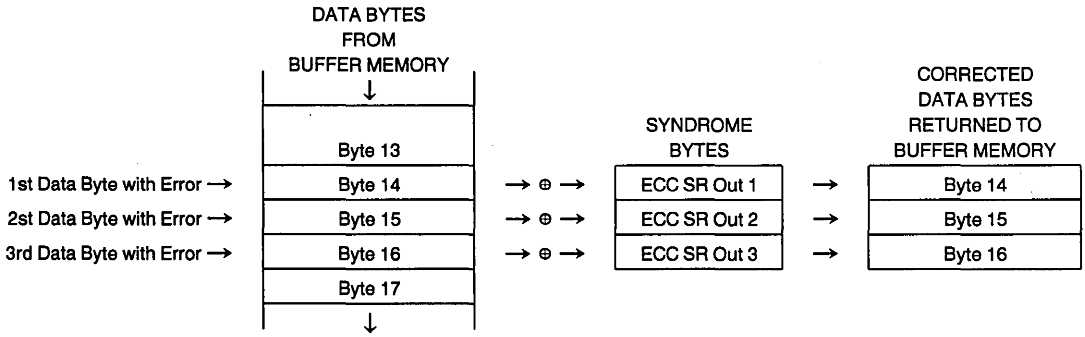
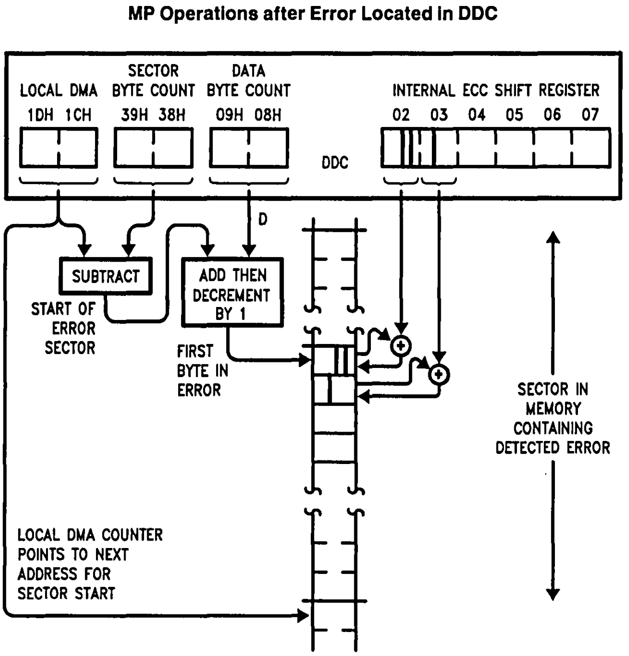

DATA BYTES FROM BUFFER MEMORY
Byte 13
1st Data Byte with Error →
Byte 14
2st Data Byte with Error →
Byte 15
3rd Data Byte with Error →
Byte 16
Byte 17
→ ⊕ →
SYNDROME BYTES
ECC SR Out 1
ECC SR Out 2
ECC SR Out 3
→
CORRECTED DATA BYTES RETURNED TO BUFFER MEMORY
Byte 14
Byte 15
Byte 16

FIGURE 7.14(c). 48-Bit ECC Correction Process


MP Operations after Error Located In DDC
LOCAL DMA
1DH 1CH
SECTOR BYTE COUNT
39H 38H
DATA BYTE COUNT
09H 08H
DDC
INTERNAL ECC SHIFT REGISTER
02 03 04 05 06 07
D
SUBTRACT
START OF ERROR SECTOR
ADD THEN DECREMENT BY 1
FIRST BYTE IN ERROR
SECTOR IN MEMORY CONTAINING DETECTED ERROR
LOCAL DMA COUNTER POINTS TO NEXT ADDRESS FOR SECTOR START

TL/F/8663-F2

FIGURE 7.15. Location of Bytes in Error (in Memory) for the Correction Process

the DDC ignores any drive command loaded into the Drive Command Register. On initiation of the Correction Cycle, the Correction Cycle Active flag, (bit 6 of the Status register), will go high.

The ECC Shift Register contains encoded information with regards to both the location of the bytes in error and the error pattern. The ECC Shift Register's contents are transposed which sets up a reverse shift without actually reversing the direction of shift in the shift register. The advantage of reverse shifting is that a non-correctable error is determined much quicker than if forward shifting is used. It also guarantees the completion of the correction cycle within the time it takes to read one sector of the disk. The ECC logic begins shifting, looking for a zero detect, i.e. detection of all zeroes in the upper (32-C) or (48-C) bits of the ECC Shift Register, where C is the correction span selected. After 8 shifts, the Data Count Register begins decrementing, with one down count for every 8 shifts of the ECC Shift Register. When the zero detect condition occurs, the control logic will stop decrementing the Data Count Register and its state indicates the byte that is in error. If the Data Byte Counter decrements to zero before the selected most significant bits of the ECC Shift Register are all zeroes, the error is non-correctable. In case of this condition or the zero detect condition of the ECC Shift Register, an interrupt is issued to indicate to the host microprocessor that the correction cycle has finished, indicated by the CCA flag (bit 6 of the Status Register being reset).

During the correction cycle other operations like completion of remote DMA etc., may issue an interrupt which should be serviced to enable recognition of the interrupt on completion of the correction cycle. The Error Register bit CF is examined, which if set signifies a non correctable error. If the bit is not set, then the error is correctable and must be either in the data field or the checkbits of the ECC field or overlapping both fields. If there is an error in the data field, the Data Field Error bit DFE of the Error Register is set. So if CF and DFE are not set, the error is in the ECC field, the memory data is correct and no further microprocessor response is needed to complete correction. If CF is not set, but DFE is set, the error in the data field can be corrected.

At the instant when the 'zero detect' condition occurs in the ECC Shift Register, the status of the Data Count Register indicates the byte in error. For example—if the data count register shows 515, then the 515th byte of the data field is in error. If there were only 512 bytes in the data field, then 515 means that the 3rd byte of the checkbit field is in error. The syndrome bytes in the ECC Shift register should be aligned so that the Most Significant Bit of the syndrome field align with the Most Significant Bit of the byte 515. However, if the syndrome spans a field of two bytes, then it will align with byte 515 and 516. When the data byte in error is located, the ECC logic makes sure that the syndrome bits are aligned properly on a bit by bit basis with that byte in error. Therefore, it will continue to shift until this has happened. To facilitate the speed restraints of the process, the syndrome will get shifted one full byte beyond where one would expect to see it in the ECC Shift Register. The syndrome bits will be located starting at the second byte of the ECC Shift Register. *Figure 7.14 (a)* shows the orientation of the ECC Shift Register for various sizes of polynomials selected.

Errors are corrected by XOR'ing syndrome bytes (ECC SR 0–5) with the bytes in the data record in memory that contain the error. The address of the first byte of the data field, in error is computed as follows: [current value of DMA address bytes 0 and 1] – [sector byte count] + [data byte count] – 1. For performing a correction with 32 bit ECC, ECC SR1, ECC SR4, and ECC SR5 contain the syndrome pattern in that sequence. ECC SR2 and ECC SR3 are not used in 32-bit mode and will contain 0's if read. ECC SR0 will contain all 0's if the error is correctable, and may contain some set bits if not. The bytes in error (in the memory) are located as shown in *Figure 7.15* while the correction process is shown in *Figure 7.14 (b)* and *(c)*. To perform a 48 bit correction ECC SR1, ECC SR2 and ECC SR3 should be read sequentially for the syndrome bits. ECC SR0, ECC SR4 and ECC SR5 are not used and will contain 0's for a correctable error. *Figure 7.16* shows an example of the correction process. *Figure 7.17* gives a flow chart of the correction cycle operations.

EXAMPLE OF A 32-BIT CORRECTION

Shown in *Figure 7.18* is a record with several bits read in error from the disk. Bits D4, D11, D13 and D14, now located in memory were read incorrectly and need to be corrected. As can be seen, the correction pattern provided in ECC SR1 and ECC SR2 can be used to correct bits D4, D11, D13 and D14. The CPU reads the Data Byte Count and computes the address of the first data byte in error, read from the disk. This byte is XOR'ed with ECC SR1 and is written back to memory. The second byte read form the disk is XOR'ed with ECC SR4 and then written back. ECC SR5 need not be used since it contains all 0's.

7.6.3 Programming the ECC

There are two sets of six registers used to program the ECC. One set of six is used to program the polynomial taps, while the other set is used to establish a preset pattern. Bits contained in the ECC Control Register are used to control the correction span. The Data Format Register contains bits for choosing the desired type of appendage: either 32 or 48 bit programmable ECC polynomials, or the 16 bit CCITT CRC polynomial.

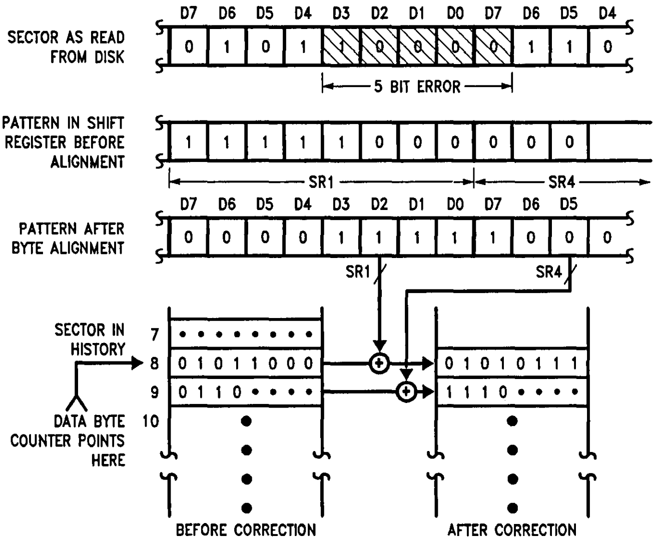
PROGRAMMING POLYNOMIAL TAPS

To program a polynomial into the shift register, each tap position used in the code must be set to 0, and all unused taps should be set to 1. The bit assignment for these registers in 48 and 32-bit modes is shown in the tables, *Figure 7.21 (a)*. It is important to note that for 32-bit codes, PTB2 and PTB3 must be set to all 1's. Failure to do so will result in improper operation. Also x^{48} for 48-bit and x^{32} for 32-bit ECC are implied and so is x^0, even though this bit is accessible.

PROGRAMMING PRESET PATTERN

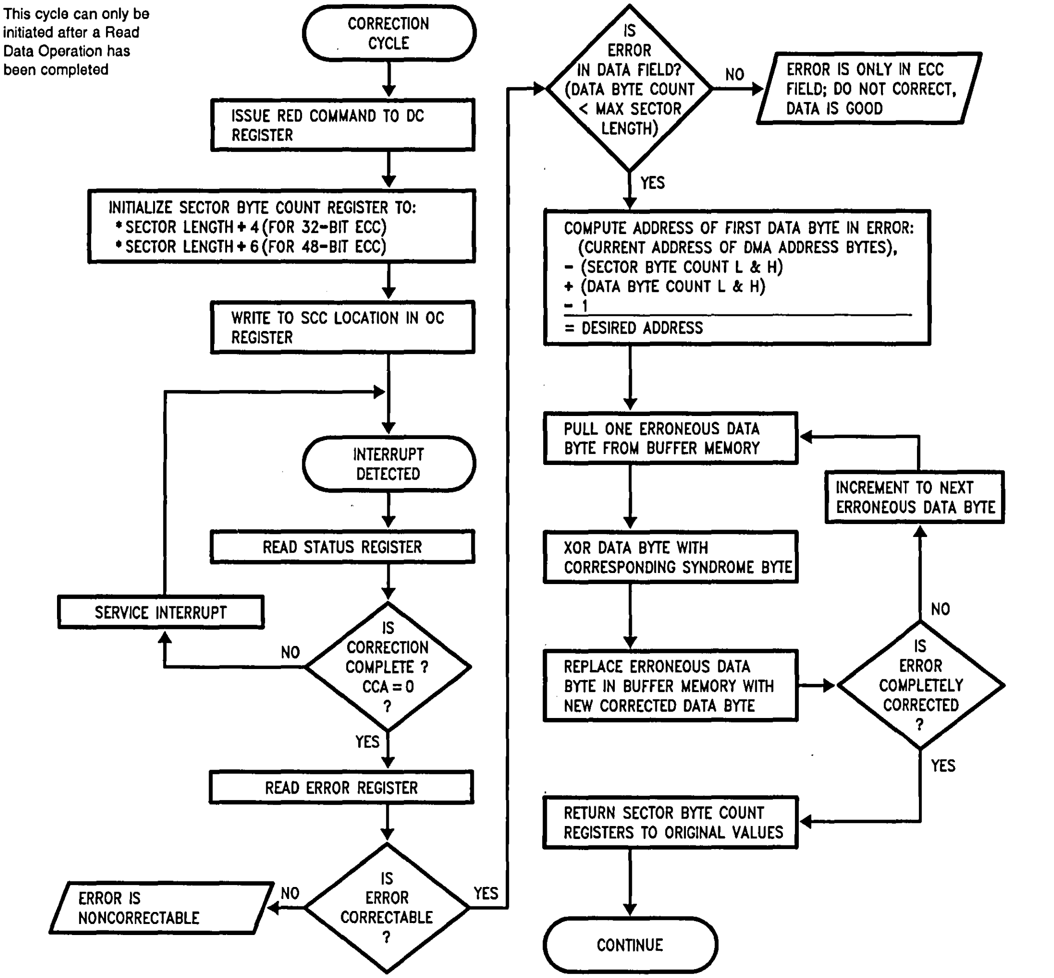
PPB0–PPB5 must be initialized to program the preset pattern that the shift registers will be preset to. As in the polynomial taps, x^{48}, x^{32}, and x^0 are implied. The assignment of the bits for 48 and 32-bit modes is shown in *Figure 7.21 (b)*. The value programmed into each register will be the preset pattern for the eight bits of the corresponding shift register. For typical operation, these will be programmed to all 1's. All unused presets should be set to 0. In 32-bit mode, PPB2 and PPB3 must be set to all 0's. Failure to do so will result in improper operation.

An Example of ECC In Action



TL/F/8663–F3

FIGURE 7.16. How Error Correction Works


This cycle can only be initiated after a Read Data Operation has been completed
CORRECTION CYCLE
ISSUE RED COMMAND TO DC REGISTER
INITIALIZE SECTOR BYTE COUNT REGISTER TO:
• SECTOR LENGTH + 4 (FOR 32-BIT ECC)
• SECTOR LENGTH + 6 (FOR 48-BIT ECC)
WRITE TO SCC LOCATION IN OC REGISTER
INTERRUPT DETECTED
READ STATUS REGISTER
SERVICE INTERRUPT
NO
IS CORRECTION COMPLETE ? CCA = 0 ?
YES
READ ERROR REGISTER
ERROR IS NONCORRECTABLE
NO
IS ERROR CORRECTABLE ?
YES
IS ERROR IN DATA FIELD? (DATA BYTE COUNT < MAX SECTOR LENGTH)
NO
ERROR IS ONLY IN ECC FIELD; DO NOT CORRECT, DATA IS GOOD
YES
COMPUTE ADDRESS OF FIRST DATA BYTE IN ERROR:
(CURRENT ADDRESS OF DMA ADDRESS BYTES),
- (SECTOR BYTE COUNT L & H)
+ (DATA BYTE COUNT L & H)
- 1
= DESIRED ADDRESS
PULL ONE ERRONEOUS DATA BYTE FROM BUFFER MEMORY
INCREMENT TO NEXT ERRONEOUS DATA BYTE
XOR DATA BYTE WITH CORRESPONDING SYNDROME BYTE
NO
REPLACE ERRONEOUS DATA BYTE IN BUFFER MEMORY WITH NEW CORRECTED DATA BYTE
IS ERROR COMPLETELY CORRECTED ?
YES
RETURN SECTOR BYTE COUNT REGISTERS TO ORIGINAL VALUES
CONTINUE

TL/F/8663-F4

FIGURE 7.17. Flow Chart of the Correction Cycle Operation

ECC CONTROL REGISTER

The ECC Control Register controls a number of functions. The correction span can be programmed using four bits of this register. Errors longer than the correction span are treated as non-correctable. The allowable correction span is 3–15 bits. If a span outside this range is loaded, then the DDC defaults to a span of three bits. There is a bit (HEN) to indicate whether Header address mark and/or synch fields are encapsulated in the CRC/ECC calculation. There is also a bit (DEN) for indicating whether data address mark and/or synch fields are encapsulated in the CRC/ECC calculation. Facility for inverting data entering and leaving the ECC Shift Register is also provided. For selecting the internal 16-bit CRC polynomial, the appropriate bits in the Disk Format register are set, the ECC Control Register is programmed as desired.

Syndrome Pattern

Register		Bit Number							
		7	6	5	4	3	2	1	0
ECC	SR1	0	0	0	1	0	0	0	0
ECC	SR4	0	1	1	0	1	0	0	0
ECC	SR5	0	0	0	0	0	0	0	0

Buffer Memory

Corresponding Buffer Data Bit Pattern							
D7	D6	D5	*	D3	D2	D1	D0
D15	*	*	D12	*	D10	D9	D8
D23	D22	D21	D20	D19	D18	D17	D16

* = location of bits in error

Figure 7.18 Example of Correction Syndrome Bits relating to Data Bit Patterns

Tap Assignment 32-Bit Mode

REG#	ADDR	Bit Number							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PTB0	08	X^7	X^6	X^5	X^4	X^3	X^2	X^1	X^0
PTB1	09	X^{15}	X^{14}	X^{13}	X^{12}	X^{11}	X^{10}	X^9	X^8
PTB2	0A	1	1	1	1	1	1	1	1
PTB3	0B	1	1	1	1	1	1	1	1
PTB4	0C	X^{23}	X^{22}	X^{21}	X^{20}	X^{19}	X^{18}	X^{17}	X^{16}
PTB5	0D	X^{31}	X^{30}	X^{29}	X^{28}	X^{27}	X^{26}	X^{25}	X^{24}

Preset Bit Assignment 32-Bit Mode

REG#	ADDR	Bit Number							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PPB0	02	X^7	X^6	X^5	X^4	X^3	X^2	X^1	X^0
PPB1	03	X^{15}	X^{14}	X^{13}	X^{12}	X^{11}	X^{10}	X^9	X^8
PPB2	04	0	0	0	0	0	0	0	0
PPB3	05	0	0	0	0	0	0	0	0
PPB4	06	X^{23}	X^{22}	X^{21}	X^{20}	X^{19}	X^{18}	X^{17}	X^{16}
PPB5	07	X^{31}	X^{30}	X^{29}	X^{28}	X^{27}	X^{26}	X^{25}	X^{24}

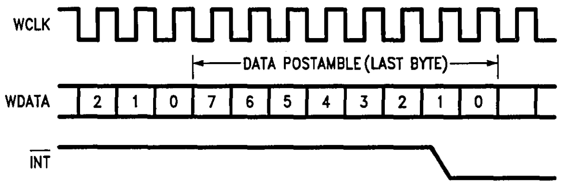
Tap Assignment 48-Bit Mode

REG#	ADDR	Bit Number							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PTB0	08	X^7	X^6	X^5	X^4	X^3	X^2	X^1	X^0
PTB1	09	X^{15}	X^{14}	X^{13}	X^{12}	X^{11}	X^{10}	X^9	X^8
PTB2	0A	X^{23}	X^{22}	X^{21}	X^{20}	X^{19}	X^{18}	X^{17}	X^{16}
PTB3	0B	X^{31}	X^{30}	X^{29}	X^{28}	X^{27}	X^{26}	X^{25}	X^{24}
PTB4	0C	X^{39}	X^{38}	X^{37}	X^{36}	X^{35}	X^{34}	X^{33}	X^{32}
PTB5	0D	X^{47}	X^{46}	X^{45}	X^{44}	X^{43}	X^{42}	X^{41}	X^{40}

Preset Bit Assignment 48-Bit Mode

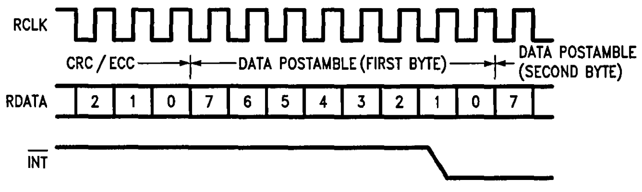
REG#	ADDR	Bit Number							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PPB0	02	X^7	X^6	X^5	X^4	X^3	X^2	X^1	X^0
PPB1	03	X^{15}	X^{14}	X^{13}	X^{12}	X^{11}	X^{10}	X^9	X^8
PPB2	04	X^{23}	X^{22}	X^{21}	X^{20}	X^{19}	X^{18}	X^{17}	X^{16}
PPB3	05	X^{31}	X^{30}	X^{29}	X^{28}	X^{27}	X^{26}	X^{25}	X^{24}
PPB4	06	X^{39}	X^{38}	X^{37}	X^{36}	X^{35}	X^{34}	X^{33}	X^{32}
PPB5	07	X^{47}	X^{46}	X^{45}	X^{44}	X^{43}	X^{42}	X^{41}	X^{40}

FIGURE 7-19. Programming the Presets of Taps; the Tap and Preset Register Configurations



TL/F/8663-F5

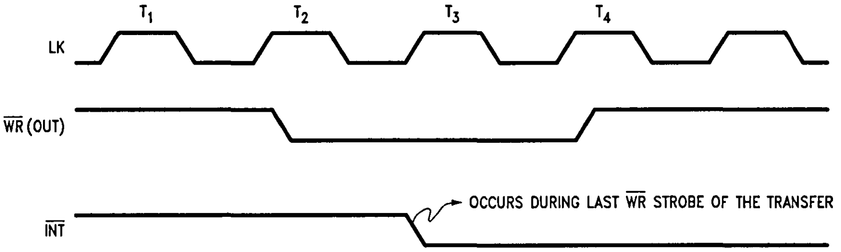
a) Write Data Operation



TL/F/8663-F6

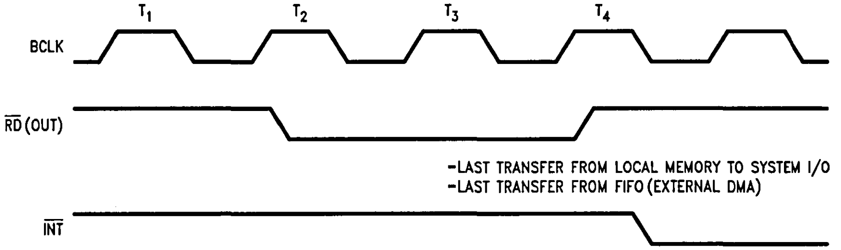
b) Check Data Operation

FIGURE 7.20. Operation Complete Interrupt Timing



TL/F/8663-F7

c) Read Data, In Single or Dual Channel (Non-tracking) Move


T1
T2
T3
T4
BCLK
RD (OUT)
-LAST TRANSFER FROM LOCAL MEMORY TO SYSTEM I/O
-LAST TRANSFER FROM FIFO (EXTERNAL DMA)
INT

TL/F/8663-F8

d) Read Data, in Dual Channel (Tracking) or in an External DMA Mode

FIGURE 7.20. Operation Complete Interrupt (Continued)

Example of programming the ECC registers

Objective: To program the 32-bit polynomial of the form; (This is National Semiconductor's recommended polynomial)

$x^{32} + x^{28} + x^{26} + x^{19} + x^{17} + x^{10} + x^{6} + x^{2} + x^{0}$

with a preset of all 1's, a correction span of 5-bits with no header/data encapsulation. The registers would be programmed as given below. Note that as defined earlier, PTB2 and PTB3 must be all 1's and PPB2 and PPB3 must be all 0's.

Polynomial Taps Registers

REG#	7	6	5	4	3	2	1	0
PTB0	1	0	1	1	1	0	1	0
PTB1	1	1	1	1	1	0	1	1
PTB2	1	1	1	1	1	1	1	1
PTB3	1	1	1	1	1	1	1	1
PTB4	1	1	1	1	0	1	0	1
PTB5	1	1	1	0	1	0	1	1

Polynomial Preset Registers

REG#	7	6	5	4	3	2	1	0
PPB0	1	1	1	1	1	1	1	1
PPB1	1	1	1	1	1	1	1	1
PPB2	0	0	0	0	0	0	0	0
PPB3	0	0	0	0	0	0	0	0
PPB4	1	1	1	1	1	1	1	1
PPB5	1	1	1	1	1	1	1	1

ECC Control Register

BIT#	D7	D6	D5	D4	D3	D2	D1	D0
	1	0	0	1	0	1	0	1

7.6.4 Internal ECC Diagnostics

The DDC has a diagnostic capability for validating the internal ECC function. By loading the Data Byte Count Register with the number of bytes in the sector plus the number of bytes of ECC appendage for the Data field. The internal CRC/ECC appendage for the Data field is set to zero so that no CRC/ECC will append the data field. Next the microprocessor sets up a data pattern in memory of all zeroes for the nominal sector length, except for bit positions where simulated errors are desired. Also, the microprocessor appends to this data the ECC appendage for an all zeroes data field by setting the Drive Command Register to perform a Compare Header-Write Data operation. In this way the DDC executes a diagnostic write function. In this mode, the data field from memory is written as in a normal write operation to the data field of the selected sector. Then the 32-bits or 48-bits of ECC check are also issued, where these check bits are falsely generated as if from an all zeroes data field. The selected sector now contains an all zeroes data field with simulated error bits followed by an ECC appendage representing checkbits generated from an all zeroes data field. The Data Byte Count is now re-loaded with the normal sector length and the correct ECC appendage length selected. A subsequent Read Data operation should produce an error indication. A correction cycle can then be implemented and the syndromes can be examined along with the Data Byte Counter contents. The microprocessor can then compare these syndromes with the positions of the simulated error bits previously written in the data field. This offers the user a diagnostics capability that simulates errors easily, merely by writing the data field with all zeroes except where the simulated error locations are desired.

7.6.5 Encapsulation of Internal ECC with External ECC

The external ECC field may be used to encapsulate the internal ECC/CRC field as a confirmation of error detection. The advantages of this scheme are that both external and internal ECC must agree on 1) the existence of the error, 2) location of the error, and 3) the error pattern. If an error is detected either internally or externally, the DDC will operate as if an internal error were detected.

7.7 INTERRUPTS

The DDC will interrupt the microprocessor only if the Interrupt Enable bit (EI) in the Operation Command register is set high. If it is not set, the INTERRUPT output is always forced high.

7.7.1 Types of Interrupt

The interrupts generated by the DDC can be divided into four categories:

1) Operation Complete Interrupt
2) Header Complete Interrupt
3) Error Interrupt
4) Correction Cycle Complete Interrupt

Each of the above mentioned types is explained in the following.

OPERATION COMPLETE INTERRUPT

The DDC will interrupt the microprocessor when it completes any one of the legal header-data disk operations listed in Table 5.13. The interrupt will also indicate that the DDC is ready to execute a new command. Some interrupt generation situations are explained below.

1) An interrupt will occur when the remote transfer is completed during a disk read operation in Tracking mode.

2) An interrupt will occur when the local transfer is completed during a disk read operation in Non-tracking mode.

3) In Non-tracking mode, if remote DMA channel is enabled, an interrupt will occur after the remote transfer is completed independent of the disk operation or the local transfer.

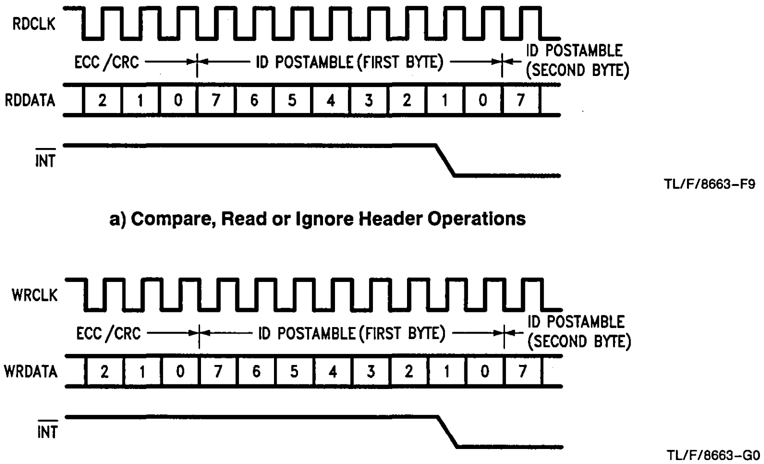
4) If the operation was a multi-sector operation, an interrupt will only occur on the completion of the last operation.

The Operation Complete interrupt generation for various Header and Data operation is shown in *Figure 7.20*. In disk write operations, the operation complete interrupt is generated when last byte of data postamble is being output by the DDC. In Header-Check data operation, operation complete interrupt occurs when first byte of data postamble enters the DDC. In disk read operations when the DDC is using only its local DMA channel (single channel DMA and non-tracking DMA modes), the operation complete interrupt is generated when the last byte (or word) is transferred to the memory. Basically it is coincident with the last WR\ strobe. When the DDC is in dual channel DMA mode, the operation complete interrupt is issued during the last RD\ strobe i.e. when last byte (or word) is transferred from local memory to system I/O. Similarly, when an external DMA is used, the operation complete interrupt is generated during the last RD\ strobe i.e. when last byte (or word) is transferred from the DDC to external memory.

HEADER COMPLETE INTERRUPT

In all legal DDC operations listed in Table 5.13, an interrupt will be generated after a header operation only if the Enable Header Complete Interrupt bit (EHI) in the operation command register is set high. In case of multi-sector operation, this interrupt will be generated after each header of a sector has been operated on. The header complete interrupt feature is commonly used when the DDC is in Interlock Mode (Refer to section 5.2.5). On interrupt, the ID and Data fields for the next sector can be changed, if desired, before the next sector operation starts.

The header complete interrupt is coincident with the Next Disk Command bit (NDC) being set in the Status register. Thus, the controlling microprocessor can be notified to load the DDC with the next disk command. In other words, the DDC could be run continuously for any length of time by loading a new disk command whenever next disk command flag is set. The generation of header complete interrupt is shown in *Figure 7.21*. In Compare, Read and Ignore Header



a) Compare, Read or Ignore Header Operations

b) Write Header Operation

FIGURE 7.21. Header Complete Interrupt Generation

operations, the interrupt is generated when first byte of ID Postamble is being read by the DDC whereas in Write Header operation, interrupt is generated when first byte of ID Postamble is being written to the disk by the DDC.

ERROR INTERRUPT

An interrupt will be generated if any bit in the Error register is set, which in turn sets the Error Detected bit (ED) in the Status register. Refer to description of Error and Status registers in section 5.1.1. Also an error interrupt will be issued.

CORRECTION CYCLE COMPLETE INTERRUPT

An interrupt will occur at the end of an internal correction cycle independent of the result of the correction cycle. If the error was not correctable, another interrupt will not be generated, only the Correction Failed flag (CF) in Error register will be set.

7.7.2 Interrupt Servicing

As explained earlier, the DDC issues an interrupt on, an operation complete, the header operation complete, the occurrence of an error and the completion of a correction cycle. Whenever an interrupt is generated, the Status and Error register should be read in order to find out which one of the four situations has happened. In the status register, flag NDC indicates the completion of an operation, flag HMC indicates the completion of a header operation, and flag ED indicates the occurrence of an error. Only when the ED flag is set, the Error register is read to find out the type of error that caused an interrupt. Also, the CF flag in the Error register indicates result of the correction cycle. The interrupt servicing for various interrupts is described below. *Figure 7.22* shows a flow chart for servicing interrupts.

Operation Complete Interrupt: In case of operation complete interrupt, the NDC flag in the Status register gets set indicating that the DDC is ready for next command. The DDC is brought to the Command Accept mode and the desired command is loaded with all other related registers initialized. Refer to sections 7.1 through 7.6.

Header Complete Interrupt: The HMC flag in the Status register, gets set in case of header complete interrupt. This basically indicates that the header operation (ignore, compare, read or write) has completed. The information in the DDC's registers can be changed before the start of next header operation i.e. during the time when the data operation for the current sector is in progress. If the DDC is in Interlock mode, the HBC/interlock register is also written to during this time. See sections 5.2.5 and 7.2 (interlock format method).

Error Interrupt: In case of an Error interrupt, the ED flag in the Status register gets set. The Error register should be read next, to find out the error that caused the interrupt. For the description of various error flags, refer to Error register description in chapter 5. The HFASM function is explained in detail in section 7.8. Also see description of Header Byte control register in chapter 5. The DFE (data field error) is caused by an ECC/CRC error in the data field. Generally, retrying the operation takes care of this error. If this error does repeat on retries, a correction cycle should be performed. See section 7.6. The SNF (sector not found) error could also be resolved by retrying the operation. If it does repeat on retries, then the head should be repositioned. The SO (sector over run) occurs while reading or writing more data than what has been allotted on the disk and could be taken care of by checking system software. The NDS (no data synch) occurs because of a mismatch in address mark or synch fields and could be resolved by retries or system check-up. The FDL (FIFO data lost) could occur due to speed incompatibility between system and the disk drive and could be resolved by retrying or checking the system. The CF (correction cycle failed) can be taken care of by retrying the correction cycle again, if still not resolved, then that means the error is not correctable. See section 7.7 on ECC/CRC. The LI (late interlock) could also be resolved by retrying. See section 5.2.5 for details on interlock operation.

Correction Cycle Complete: The Error register is read. If CF flag indicates that the correction cycle failed then it can be performed again. After retry if it still fails, then the error is not correctable. See section 7.6 for details on correction cycle.

7.7.3 Interrupt Clearing

The INT pin will be forced inactive high any time the status register is read. If an interrupt condition arises during a status read, an interrupt will be generated as soon as the status read is finished. INT pin will also be deactivated by setting the internal Reset bit (RES) or asserting the external RESET pin. Clearing the RED bit in Drive Command register will not deactivate an interrupt.

7.8 ADDITIONAL OPERATIONS

7.8.1 Data Recovery Using the Interlock Feature

The potential use of the interlock feature is in recovering data from a sector with an unreadable header or ID field. It is assumed that the number of the sector physically preceding the bad sector on the disk is known. A single-sector operation will be performed on these sectors, and the Drive Command Register will be changed in between them. The following steps will recover the data.

Step 1. The header bytes of the physical sector preceding the desired sector are loaded into the relevant header byte pattern registers.

Step 2. The OC Register must be loaded with the EI, EHI, IR bits set. This enables the Header Complete interrupt as well as the interlock feature.

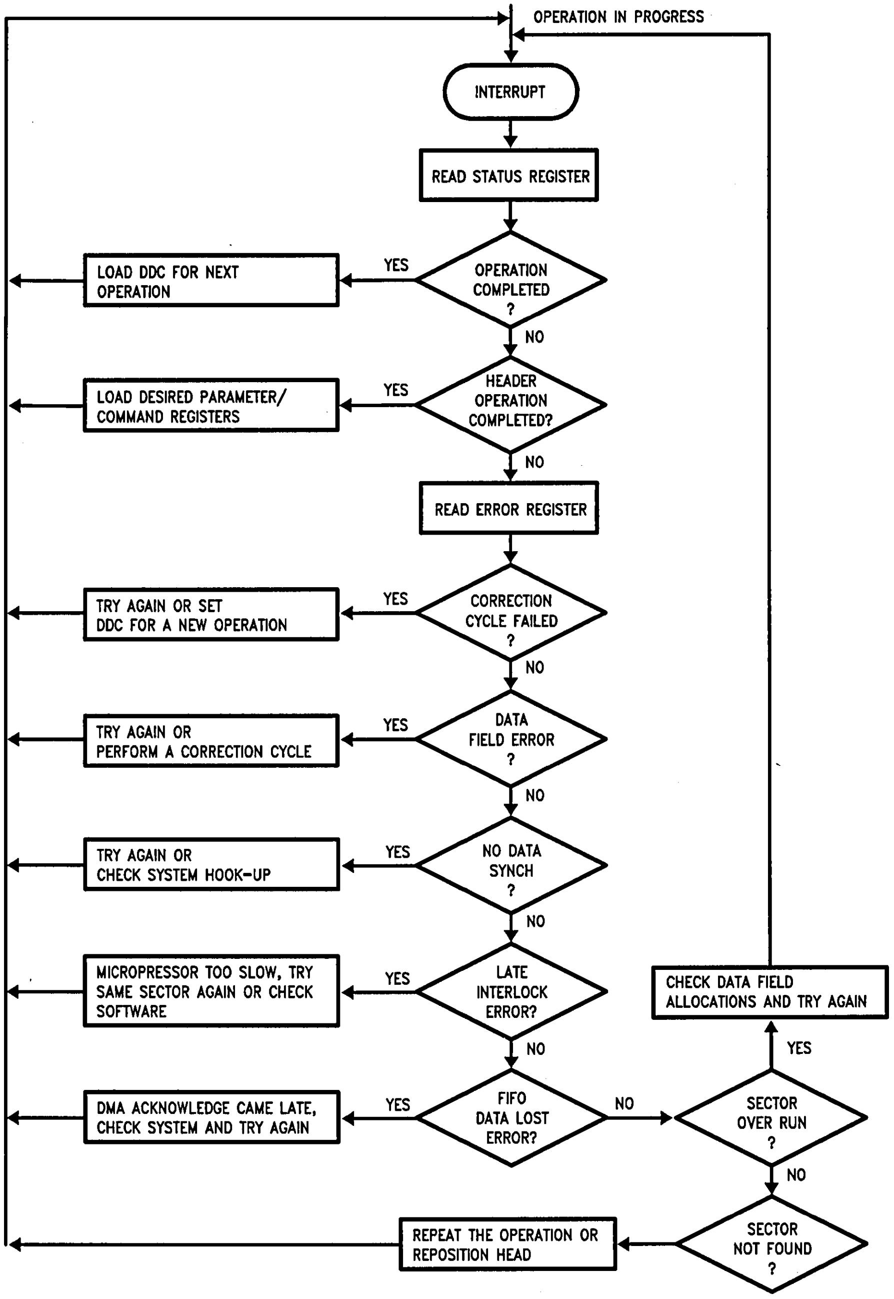
Step 3. The DC register is loaded for a single-sector, compare header/check data operation.

Step 4. After the header complete interrupt, the DC register must be loaded with an Ignore Header/Read data operation, and the Interlock Register (HBC) written to. If the controlling microprocessor fails to write to the HBC register before the end of the data field of the first sector, a Late Interlock error (LI bit in the Error Register) will be flagged, and the operation will be terminated with an interrupt.

Step 5. When the HMC interrupt occurs on the second sector, the Interlock (HBC) Register must be written to again to avoid a LI error.

Step 6. The operation will terminate normally when the data from the badly labeled sector has been read.

Figure 7.23 shows the data recovery algorithm.


OPERATION IN PROGRESS
INTERRUPT
READ STATUS REGISTER
OPERATION COMPLETED ?
YES
LOAD DDC FOR NEXT OPERATION
NO
HEADER OPERATION COMPLETED?
YES
LOAD DESIRED PARAMETER/ COMMAND REGISTERS
NO
READ ERROR REGISTER
CORRECTION CYCLE FAILED ?
YES
TRY AGAIN OR SET DDC FOR A NEW OPERATION
NO
DATA FIELD ERROR ?
YES
TRY AGAIN OR PERFORM A CORRECTION CYCLE
NO
NO DATA SYNCH ?
YES
TRY AGAIN OR CHECK SYSTEM HOOK-UP
NO
LATE INTERLOCK ERROR?
YES
MICROPRESSOR TOO SLOW, TRY SAME SECTOR AGAIN OR CHECK SOFTWARE
NO
FIFO DATA LOST ERROR?
YES
DMA ACKNOWLEDGE CAME LATE, CHECK SYSTEM AND TRY AGAIN
NO
SECTOR OVER RUN ?
YES
CHECK DATA FIELD ALLOCATIONS AND TRY AGAIN
NO
SECTOR NOT FOUND ?
REPEAT THE OPERATION OR REPOSITION HEAD

TL/F/8663-G1

FIGURE 7.22 Interrupt Servicing

7.8.2 The Header Failed Although Sector Matched (HFASM) Function

The Header Failed Although Sector Matched, HFASM function can be used to perform some disk maintenance and diagnostic functions. The HFASM function (pronounced H-fazzzm) has been described very basically in section 5.1.1 and 5.1.3. In this section, an attempt is made to provide a more detailed description and some example uses.

The HFASM function is essentially an Error that can be enabled by setting the EHF bit in any one or more Header Control Registers. When this bit for any header byte is enabled, and a Compare Header Operation is performed, the HFASM Error will be generated if certain conditions are met. The Error is generated if a header byte pattern register matches with its disk header byte, and that header byte had its EHF bit, and any other byte in the header fails to match. If multiple header bytes have been enabled only one need to match, while *any* other header byte does not match in order to generate an HFASM error. The other header bytes may or may not have their EHF bit set. Thus, this error can tell the system when a particular type of header has been found, even though the exact header did not match.

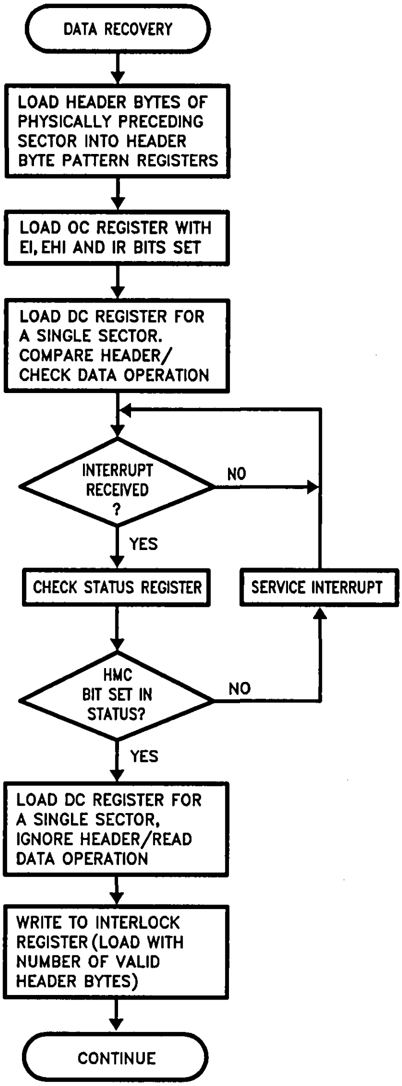
The HFASM error is generated only when execute a command that has a Compare header operation. Write Header, Read Header and Ignore Header operations will not generate an HFASM error. If a Compare Header Operation, and a Check Data Operation form the command is executed and an HFASM error is generated, no data is transferred to the system, but the DDC will load the Header into the FIFO. If a multi-sector operation was in progress the HFASM Error being set will terminate the operation. If the HASM Error is set, but the sector has a CRC error as well, the DDC will terminate the command with both bits set.

This command can be used for various tasks. For example, if the sector's sector number byte has the EHF bit set for all read and write operations, no HFASM error should occur. If one does occur then the system knows that the correct sector number was found, and the reason the correct sector was not found was because of a seek error (head on wrong track); the header was marked bad (and the DDC is looking only for good headers); the wrong head was selected; or some other header parameter was incorrect.

Another example suppose that the header byte that is designated as the sector's sector number has its EHF bit set, as before. The system wants to find sector one, but does not know the other header information. If a Compare Header-Check Data Command is executed, an HFASM error the FIFO will be loaded with the actual header that has a sector number of one. The system can then determine what the status/flag information is (is it a bad sector or a good one etc?) or which cylinder the head is on.

A third possibility, would be to find specific sectors that might have their flag header bytes indicating a bad sector. The EHF bit of the header byte designated for the flag should have its EHF bit set. In this case if a Compare header-Check data is performed using the interlock mode and starting on the sector, all the bad sectors can be identified.

In general the HFASM function is a subtle but powerful tool to enable some diagnostics, and provides to a limited degree the ability for a user definable error condition.


DATA RECOVERY
LOAD HEADER BYTES OF PHYSICALLY PRECEDING SECTOR INTO HEADER BYTE PATTERN REGISTERS
LOAD OC REGISTER WITH EI, EHI AND IR BITS SET
LOAD DC REGISTER FOR A SINGLE SECTOR. COMPARE HEADER/ CHECK DATA OPERATION
INTERRUPT RECEIVED ?
NO
YES
CHECK STATUS REGISTER
SERVICE INTERRUPT
HMC BIT SET IN STATUS?
NO
YES
LOAD DC REGISTER FOR A SINGLE SECTOR, IGNORE HEADER/READ DATA OPERATION
WRITE TO INTERLOCK REGISTER (LOAD WITH NUMBER OF VALID HEADER BYTES)
CONTINUE

TL/F/8663-G2

FIGURE 7.23. Data Recovery Using Interlock Feature

APPENDIX: DISK DATA CONTROLLER COMMAND FLOW CHARTS

The design guide has covered the general ways that the DDC can execute the various operations. However, it is impossible to account for all possible design situations in the guide. To attempt to provide information which the designer can utilize to determine how the controller might behave in various situations, the figures in this section outline the command flow for the disk operations. All header and data operations are covered, and DMA operations are excluded except to the extent that information must be loaded/unloaded from the FIFO to ensure no overflow or underflow occurs. Other than this DMA operation occurs independently (and concurrently) to the actual disk operations.

The command flow that the DDC executes is divided into header and data operations. In addition to this there are several operations that are performed concurrently, these are shown in the flow charts as special sections and are labelled as concurrent operations. *Figures A.1* to *A.13* cover the header operation and *Figures A.14* to *A.18* cover the data operations. These are described below.

A.1 START OF COMMAND

After a reset, the DDC is in the standby mode waiting for a command. In this mode the DDC is waiting for a command, and once a command is loaded the controller starts its operation. The first major task is to start the DMA transfer to the FIFO if a write data operation is desired. This will start a concurrent task for the DMA controller that will be executed throughout the write data operation. Note that this is shown in *Figure A.2* as a separate dotted outlined block. Otherwise the rest of *Figure A.1* sets up for the header operation, including when to start this operation.

A.2 WRITE HEADER OPERATION

In *Figure A.3* a Write Header operation is assumed. The first test is whether the data output is to be MFM or NRZ. If MFM, the precompensation may be enabled if NRZ, *Figure A.4* is executed. The first steps taken by the DDC are to write the preamble, address mark bytes, and the sync bytes if desired. Also, if the CRC/ECC is to encapsulate both header and sync fields than it is turned on, otherwise it will be turned on later.

Figure A.4, A.5, and *A.6* complete the write header operation. First the CRC/ECC is enabled, then the header bytes are written. There are various options on how to do this and they are shown in this figure. Finally the CRC/ECC fields are written to the disk. This includes the internal one, and/or an external field if desired. The last task is to write the header postamble field.

At entry point D, a standard end to the header operation is shown. This is used by other header operations. Here the header match bit is set and if interrupts are enabled, the Header completion interrupt is issued. If the operation is a multi-sector operation, then the Number of Sector Operations counter and the Start Sector register are updated. If the last operation then the DDC is ready for the next command and will start the data operation.

A.3 NON-WRITE HEADER OPERATION

For a read or ignore header operation, the entry point in *Figure A.7* is H. First the operation waits to start based on the internally programmed mode and the receipt of an index or sector pulse. Then Read Gate is asserted, and the DDC searches for the sync or address mark fields. Once all the sync and address mark fields match, the DDC can perform the Header operation. *Figure A.5.* If encapsulation of the CRC/ECC was enabled then CRC/ECC calculation will begin at the address mark field.

If not already active, the CRC/ECC is activated, and the DDC starts operating on the header information. If the data operation is a check data or the header operation is a read header (the later is shown in *Figure A.11*), then the header bytes are loaded into the FIFO. If the operation is a compare header, the header bytes are compared to the registers or start sector register. Next the CRC/ECC is checked. If it is in error then the header fault flag is set, however the operation is not aborted. If there is a CRC/ECC error or the header did not match, the DDC then looks for the next sector. If the correct header is found then, the DDC will deassert read gate to avoid the write splice, and will proceed to execute a data operation.

A.4 WRITE DATA OPERATION

For a write data operation, the write gate is asserted, and an algorithm similar to the write header operation is executed. MFM or NRZ data output is configured, and the start with address mark mode is checked. Address mark, preamble and sync fields are written *Figure A.14.* If needed the CRC/ECC generator is enabled when the Address Mark is written.

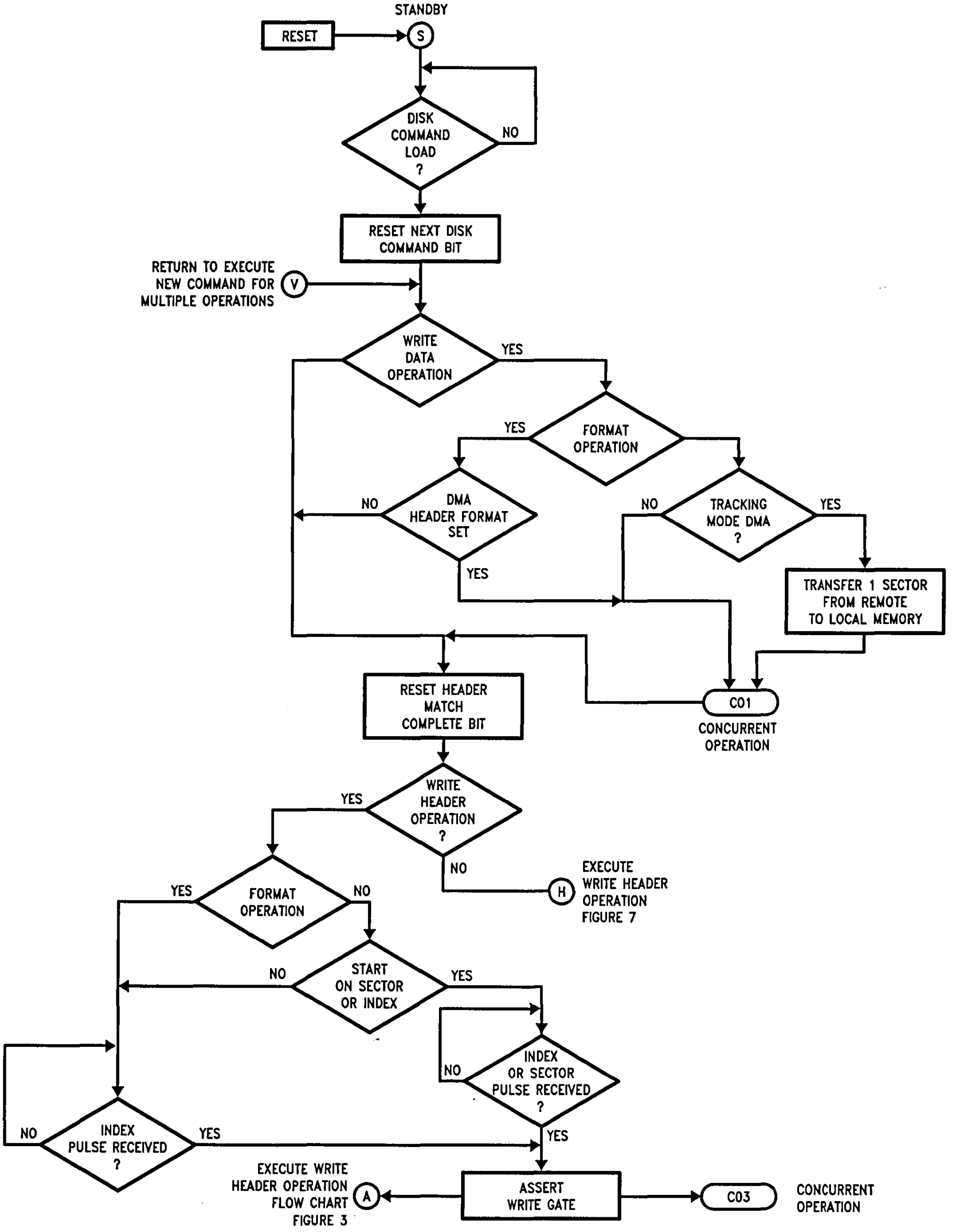
Just prior to writing data, the CRC/ECC will be enabled, if it is not already. Then data is written to the disk from the FIFO or the Format pattern register. Once the data is written, the CRC/ECC is written, followed by the data postamble. If the operation is not a format operation then write gate is deasserted, and interrupts may be generated. If the operation is a format operation, the post sector gap is written, as shown in *Figure A.16.*

The operation then is completed and the DDC will check to see if the command was a multisector one or if a new command has been entered. If so then a new command is started immediately, and this is shown by jumping to entry point V in *Figure A.1.*

A.5 NON-WRITE DATA OPERATION

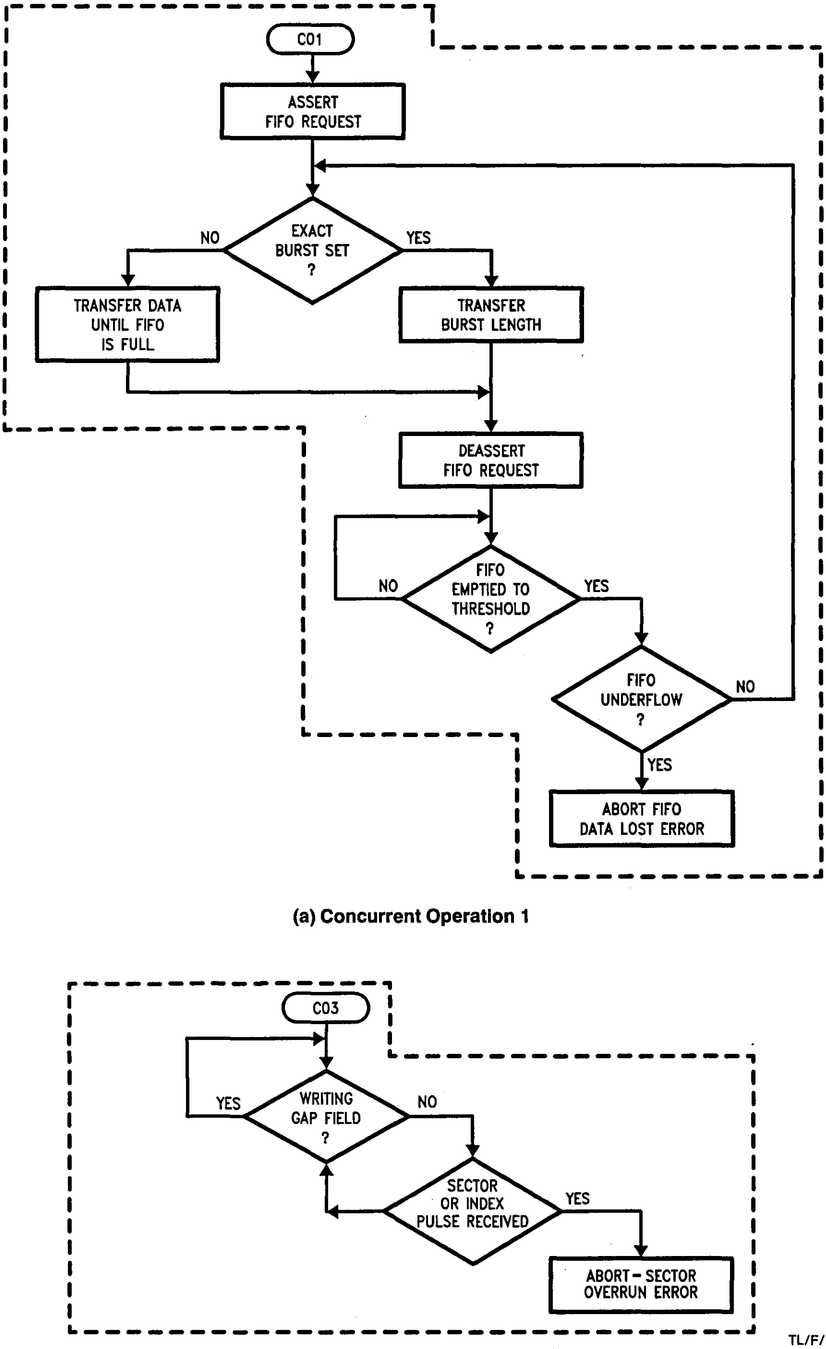
The flow chart for a read data or check data operation is shown in *Figure A.17.* After the header postamble, read gate is asserted. The DDC first configures when the CRC/ECC calculation should begin. It then begins checking the address mark and sync fields. If an error occurs the operation is aborted. Otherwise if the operation is a read data, the data is sent to the FIFO, and the CRC/ECC is checked.

If the operation is a check data command, the data field is not transferred or checked, and the DDC just counts this field and checks the CRC/ECC. At exit point U on *Figure A.18* the end of the data operation jumps to *Figure A.16,* where the interrupts may be generated if enabled and the DDC checks for another command.



TL/F/8663-G3

FIGURE A.1. Flow Chart for Start of a Command


C01
ASSERT FIFO REQUEST
EXACT BURST SET ?
NO
YES
TRANSFER DATA UNTIL FIFO IS FULL
TRANSFER BURST LENGTH
DEASSERT FIFO REQUEST
FIFO EMPTIED TO THRESHOLD ?
NO
YES
FIFO UNDERFLOW ?
NO
YES
ABORT FIFO DATA LOST ERROR
C03
WRITING GAP FIELD ?
YES
NO
SECTOR OR INDEX PULSE RECEIVED
YES
ABORT - SECTOR OVERRUN ERROR

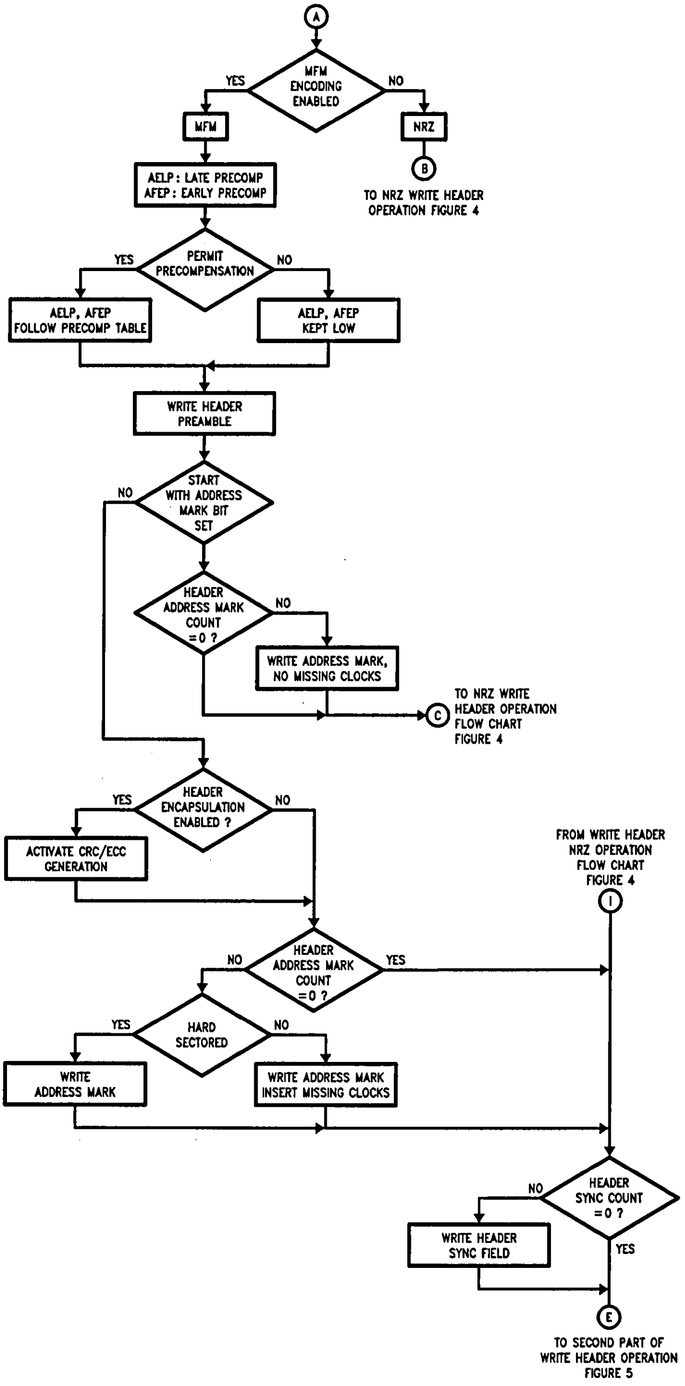
TL/F/8663-G4

(a) Concurrent Operation 1

TL/F/8663-G5

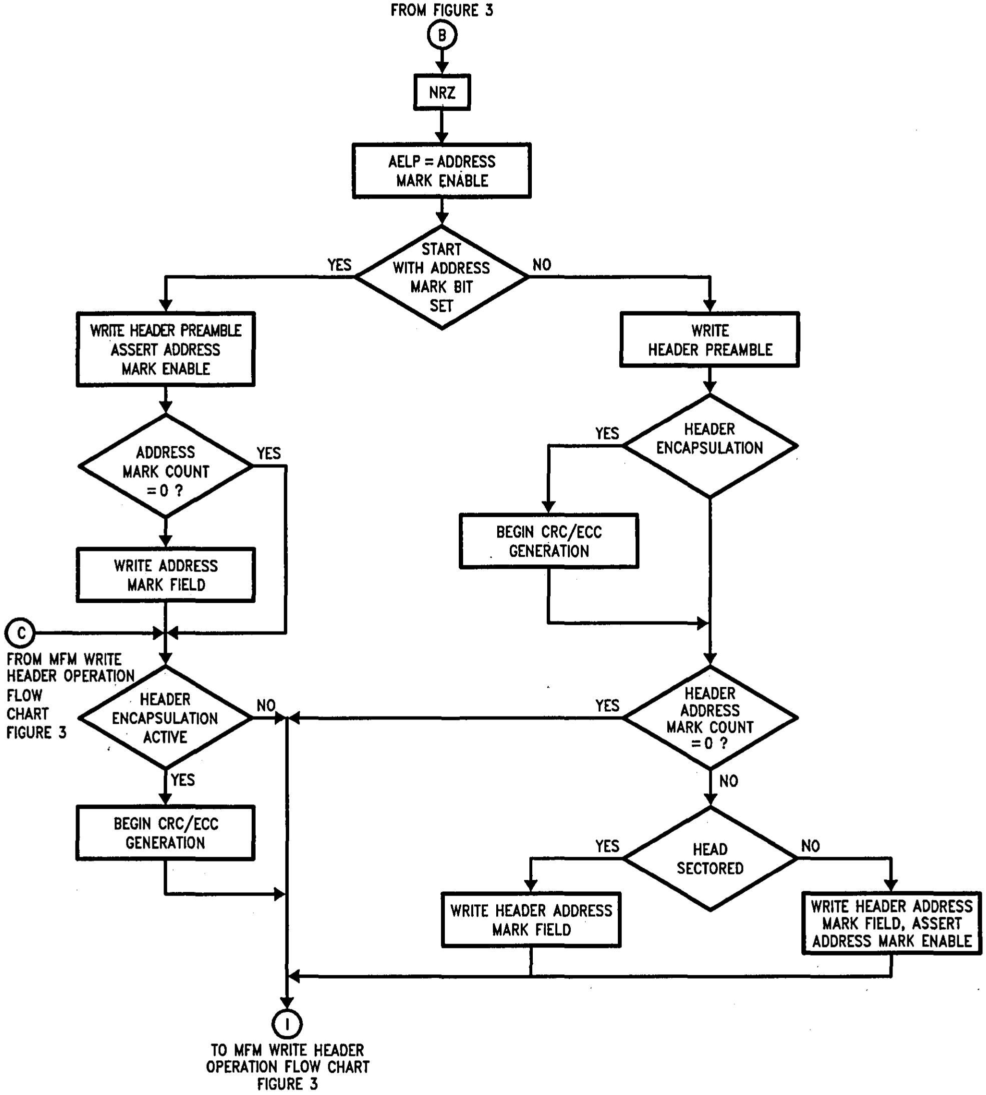
(b) Concurrent Operation 3

FIGURE A.2. Concurrent Operations for the Start of a Command


A
MFM ENCODING ENABLED
YES
NO
MFM
NRZ
B
TO NRZ WRITE HEADER OPERATION FIGURE 4
AELP: LATE PRECOMP
AFEP: EARLY PRECOMP
PERMIT PRECOMPENSATION
YES
NO
AELP, AFEP FOLLOW PRECOMP TABLE
AELP, AFEP KEPT LOW
WRITE HEADER PREAMBLE
START WITH ADDRESS MARK BIT SET
NO
HEADER ADDRESS MARK COUNT =0 ?
NO
WRITE ADDRESS MARK, NO MISSING CLOCKS
C
TO NRZ WRITE HEADER OPERATION FLOW CHART FIGURE 4
HEADER ENCAPSULATION ENABLED ?
YES
NO
ACTIVATE CRC/ECC GENERATION
FROM WRITE HEADER NRZ OPERATION FLOW CHART FIGURE 4
I
HEADER ADDRESS MARK COUNT =0 ?
NO
YES
HARD SECTORED
YES
NO
WRITE ADDRESS MARK
WRITE ADDRESS MARK INSERT MISSING CLOCKS
HEADER SYNC COUNT =0 ?
NO
YES
WRITE HEADER SYNC FIELD
E
TO SECOND PART OF WRITE HEADER OPERATION FIGURE 5

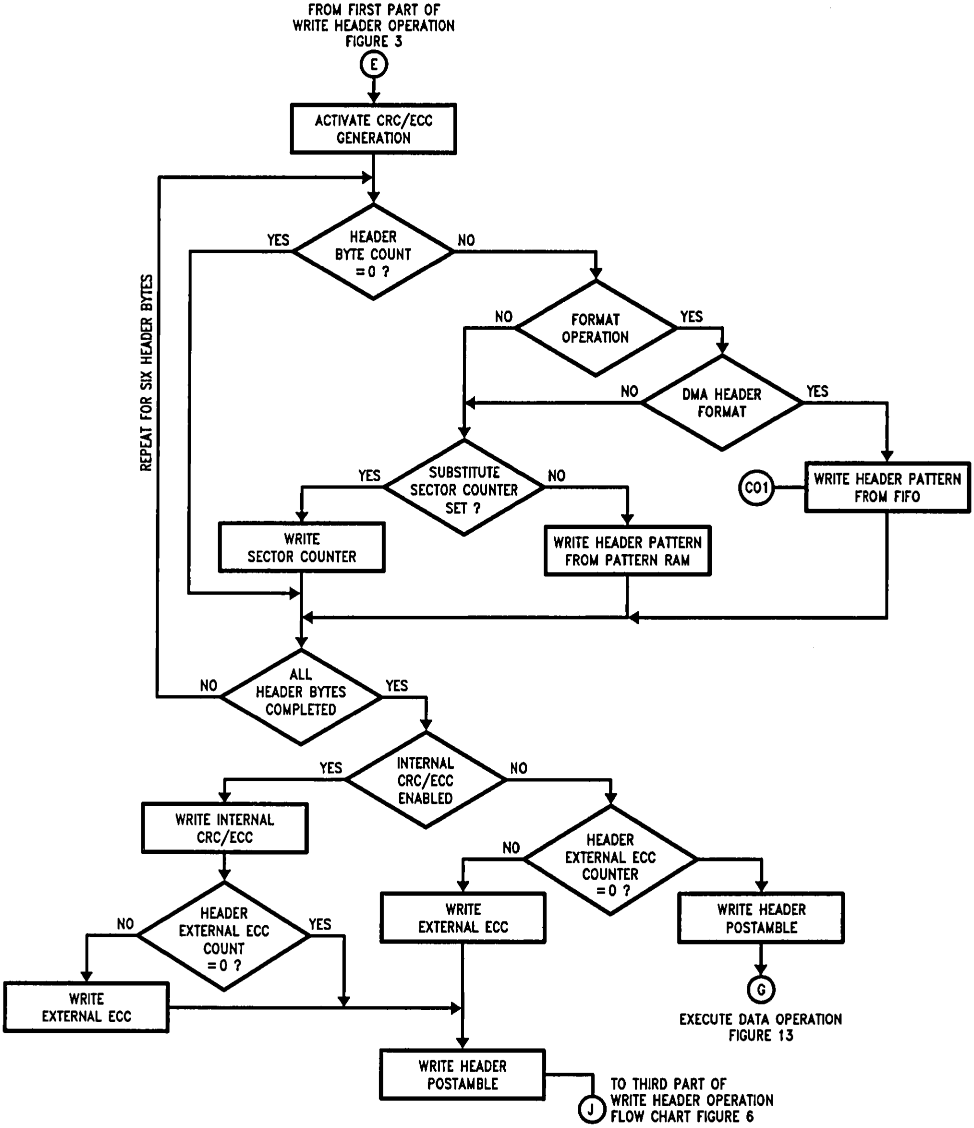
TL/F/8663-G6

FIGURE A.3. Flow Chart for First Half of Write Header Operation


FROM FIGURE 3
B
NRZ
AELP = ADDRESS MARK ENABLE
START WITH ADDRESS MARK BIT SET
YES
NO
WRITE HEADER PREAMBLE ASSERT ADDRESS MARK ENABLE
WRITE HEADER PREAMBLE
ADDRESS MARK COUNT = 0 ?
YES
HEADER ENCAPSULATION
YES
BEGIN CRC/ECC GENERATION
WRITE ADDRESS MARK FIELD
C
FROM MFM WRITE HEADER OPERATION FLOW CHART FIGURE 3
HEADER ENCAPSULATION ACTIVE
NO
YES
HEADER ADDRESS MARK COUNT = 0 ?
YES
NO
BEGIN CRC/ECC GENERATION
HEAD SECTORED
YES
NO
WRITE HEADER ADDRESS MARK FIELD
WRITE HEADER ADDRESS MARK FIELD, ASSERT ADDRESS MARK ENABLE
I
TO MFM WRITE HEADER OPERATION FLOW CHART FIGURE 3

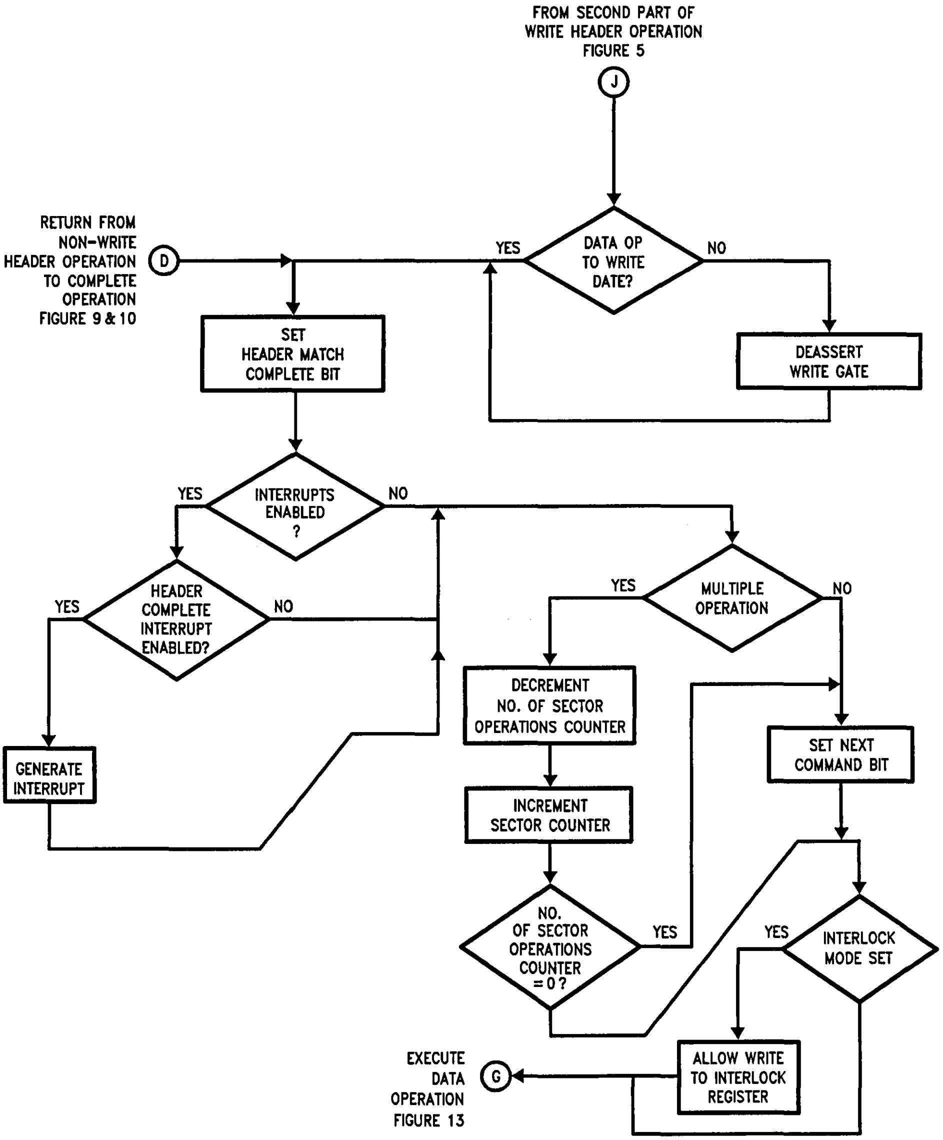
TL/F/8663-G7

FIGURE A.4. Flow Chart for First Half of Write Header NRZ Operation



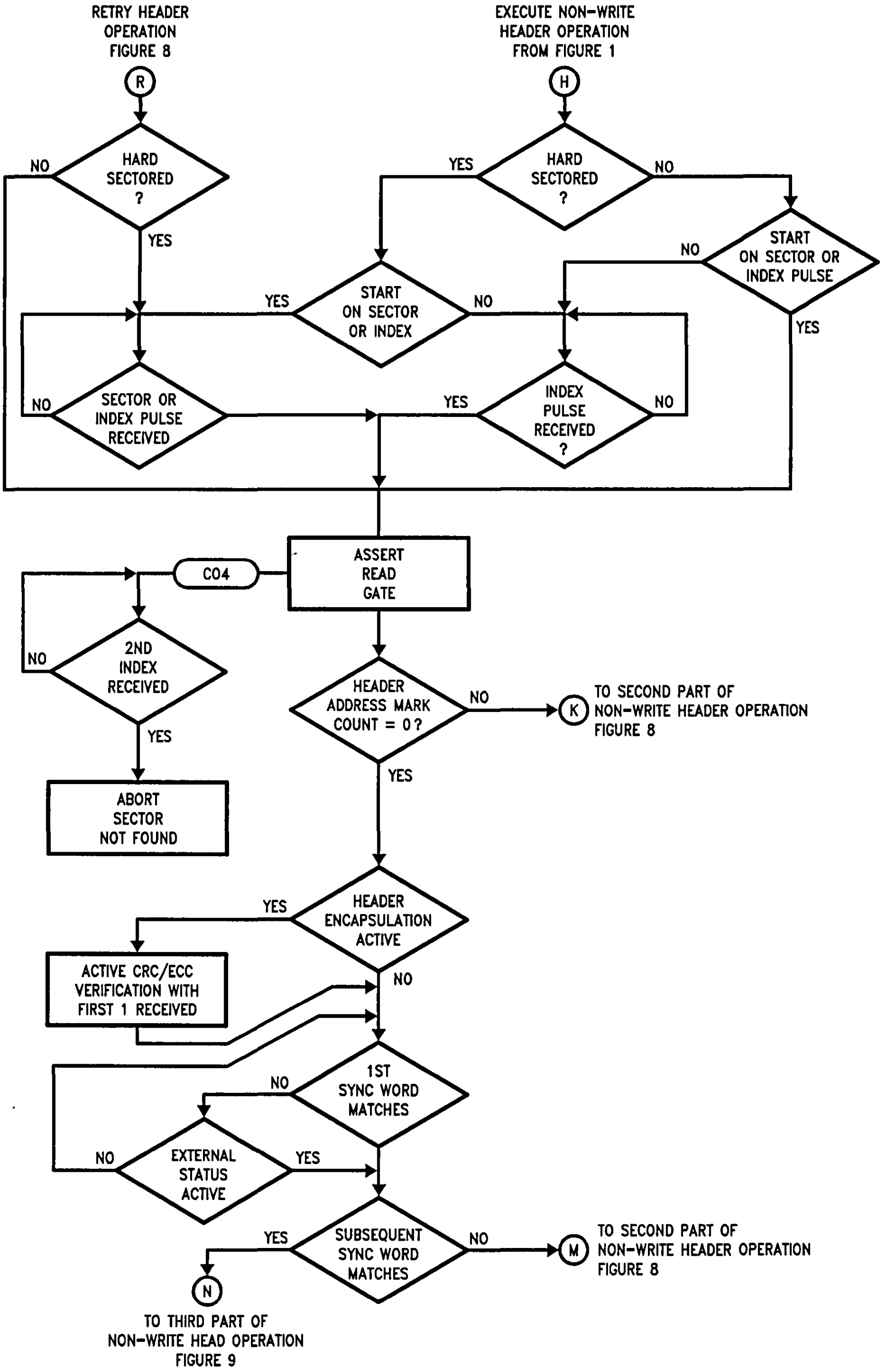
TL/F/8663-G8

FIGURE A.5. Flow Chart for Second Part of a Write Header Operation



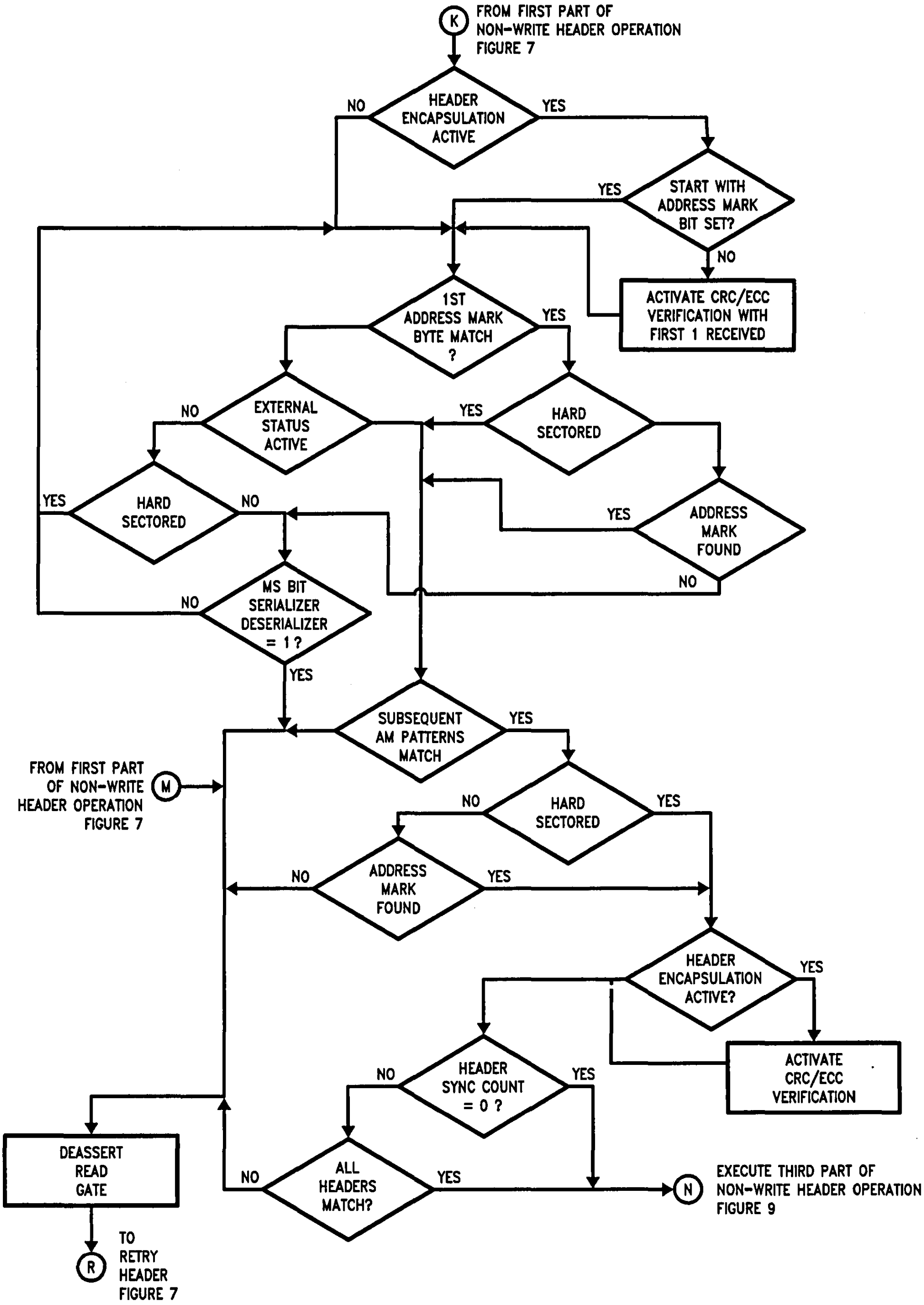
TL/F/8663-G9

FIGURE A.6. Flow Chart for Third Part of a Write Header Operation and Ending Sequence for Other Header Operations



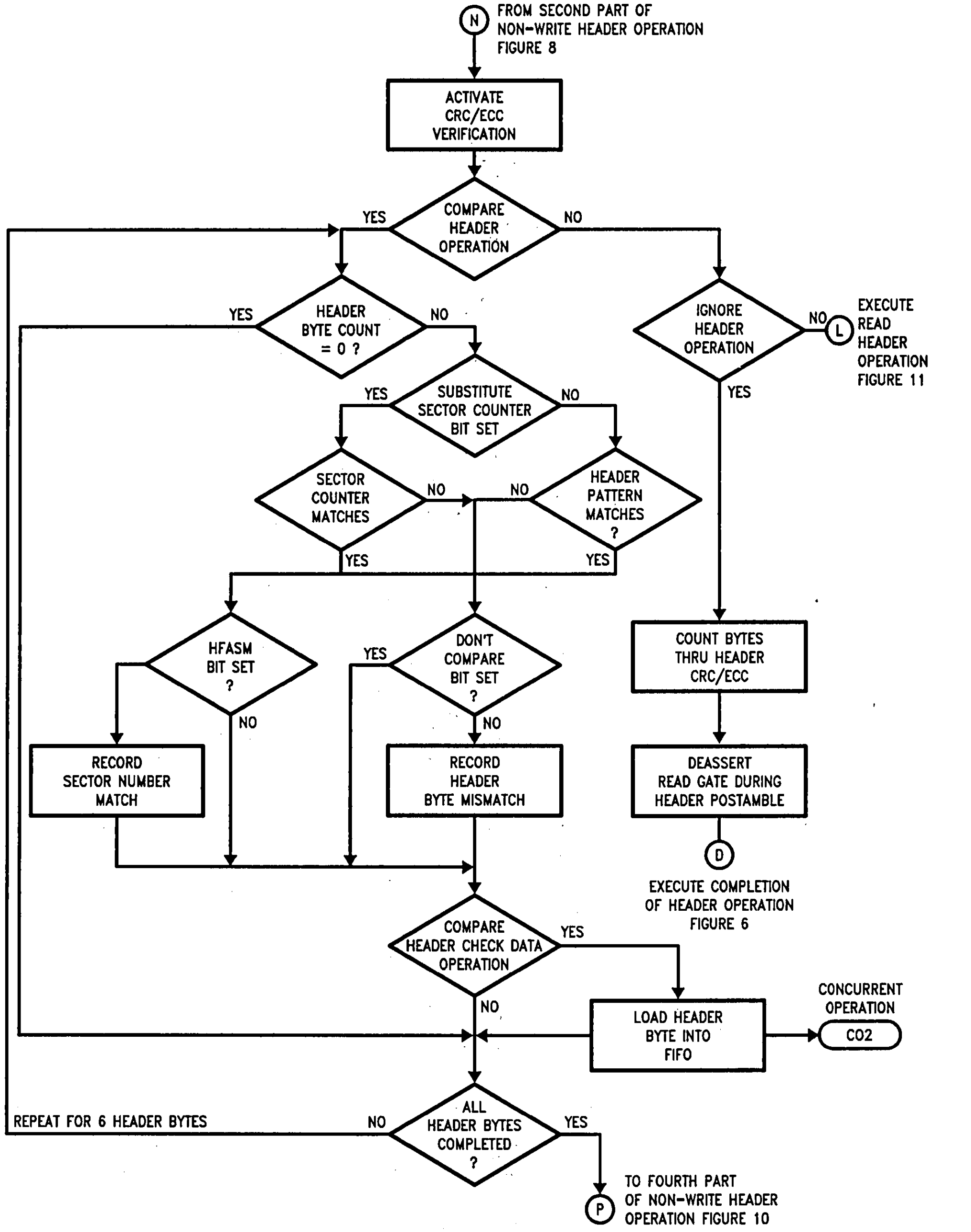
TL/F/8663-H0

FIGURE A.7. Flow Chart for First Part of Non-Write Header Operation and Retry Header


K
FROM FIRST PART OF NON-WRITE HEADER OPERATION FIGURE 7
HEADER ENCAPSULATION ACTIVE
NO
YES
START WITH ADDRESS MARK BIT SET?
YES
NO
ACTIVATE CRC/ECC VERIFICATION WITH FIRST 1 RECEIVED
1ST ADDRESS MARK BYTE MATCH ?
YES
EXTERNAL STATUS ACTIVE
NO
YES
HARD SECTORED
HARD SECTORED
YES
NO
ADDRESS MARK FOUND
YES
NO
MS BIT SERIALIZER DESERIALIZER = 1 ?
NO
YES
SUBSEQUENT AM PATTERNS MATCH
YES
FROM FIRST PART OF NON-WRITE HEADER OPERATION FIGURE 7
M
HARD SECTORED
NO
YES
ADDRESS MARK FOUND
NO
YES
HEADER ENCAPSULATION ACTIVE?
YES
ACTIVATE CRC/ECC VERIFICATION
HEADER SYNC COUNT = 0 ?
NO
YES
ALL HEADERS MATCH?
NO
YES
N
EXECUTE THIRD PART OF NON-WRITE HEADER OPERATION FIGURE 9
DEASSERT READ GATE
R
TO RETRY HEADER FIGURE 7

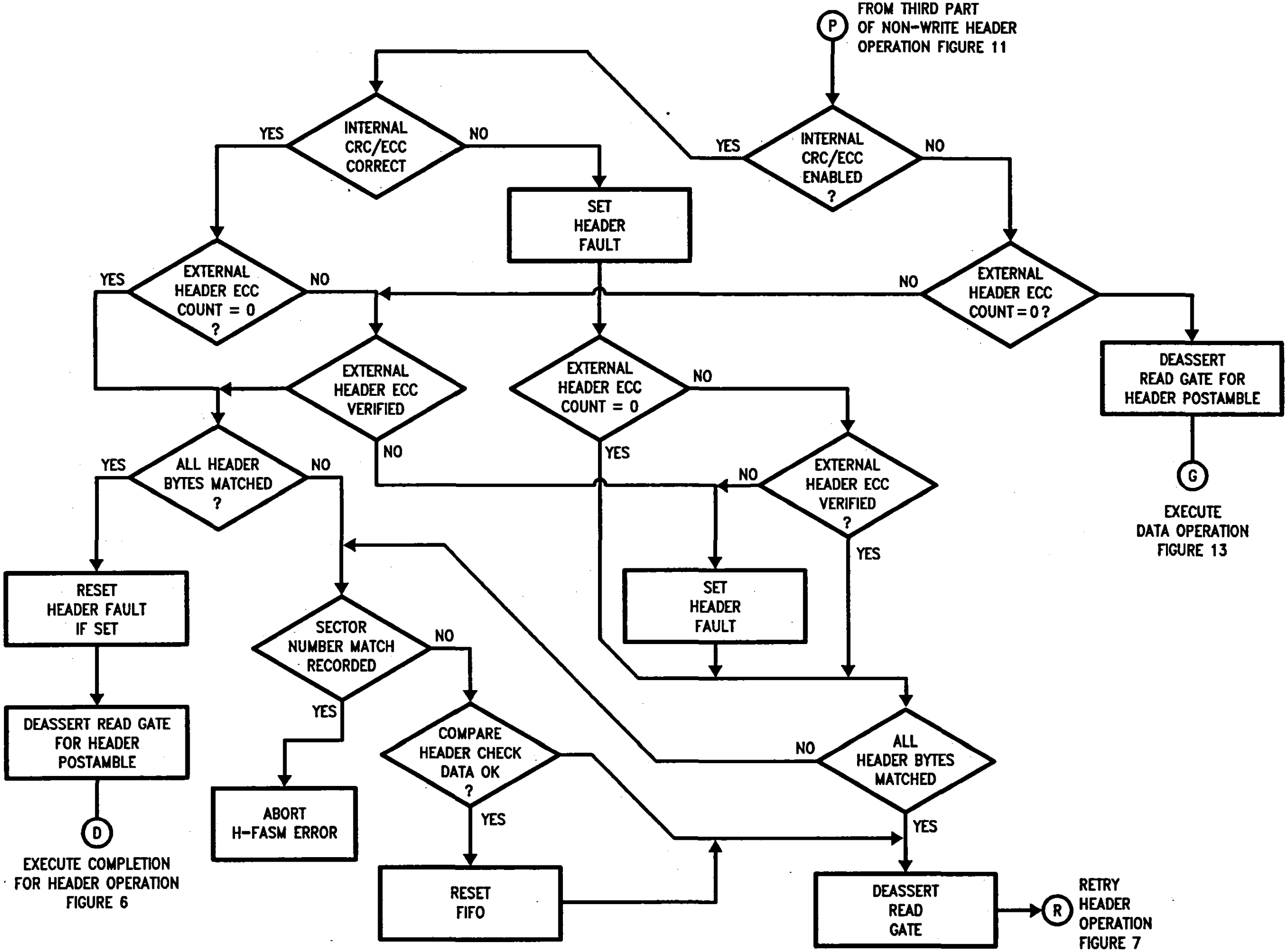
TL/F/8663-H1

FIGURE A.8. Flow Chart for Second Part of Non-Write Header Operation


N
FROM SECOND PART OF NON-WRITE HEADER OPERATION FIGURE 8
ACTIVATE CRC/ECC VERIFICATION
COMPARE HEADER OPERATION
YES
NO
HEADER BYTE COUNT = 0 ?
YES
NO
IGNORE HEADER OPERATION
NO
L
EXECUTE READ HEADER OPERATION FIGURE 11
YES
SUBSTITUTE SECTOR COUNTER BIT SET
YES
NO
SECTOR COUNTER MATCHES
NO
NO
HEADER PATTERN MATCHES ?
YES
YES
COUNT BYTES THRU HEADER CRC/ECC
HFASM BIT SET ?
NO
DON'T COMPARE BIT SET ?
YES
NO
RECORD SECTOR NUMBER MATCH
RECORD HEADER BYTE MISMATCH
DEASSERT READ GATE DURING HEADER POSTAMBLE
D
EXECUTE COMPLETION OF HEADER OPERATION FIGURE 6
COMPARE HEADER CHECK DATA OPERATION
YES
NO
LOAD HEADER BYTE INTO FIFO
CONCURRENT OPERATION
C02
REPEAT FOR 6 HEADER BYTES
NO
ALL HEADER BYTES COMPLETED ?
YES
P
TO FOURTH PART OF NON-WRITE HEADER OPERATION FIGURE 10

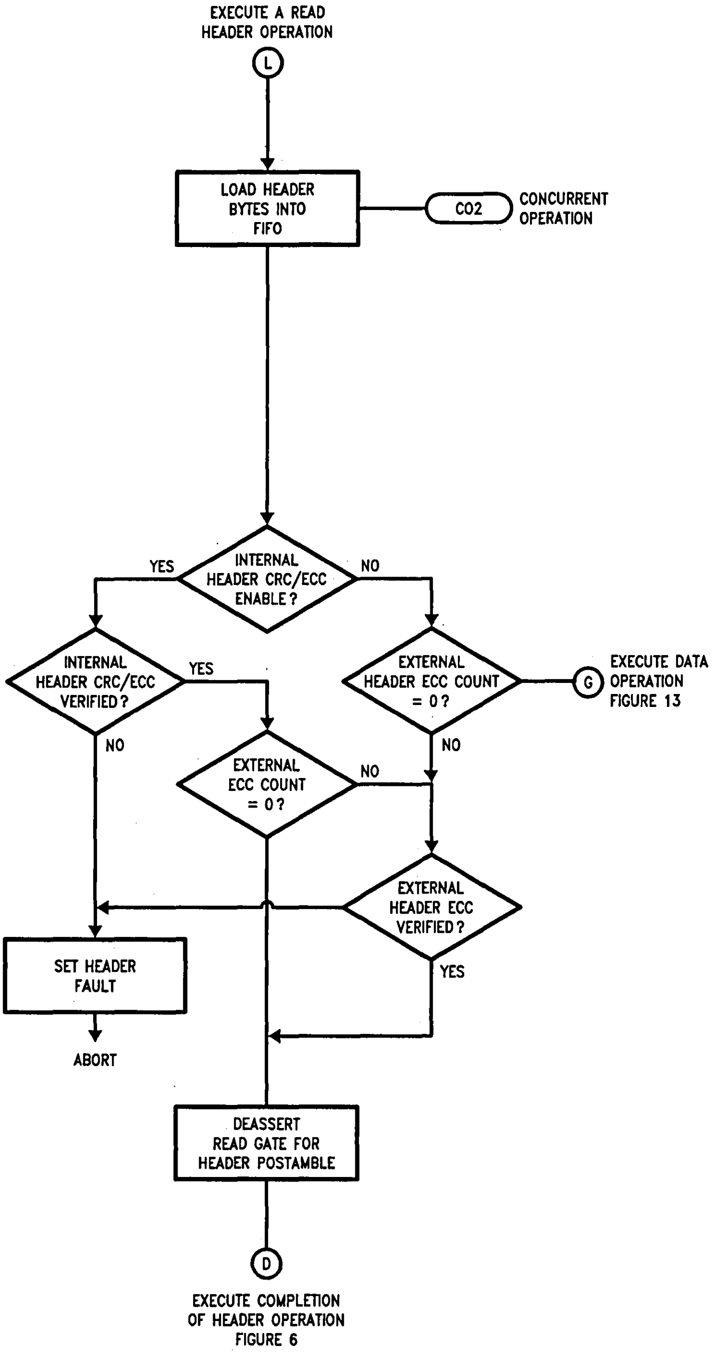
TL/F/8663-H2

FIGURE A.9. Flow Chart for Third Part of Non-Write Header Operation (Except Compare Header)


P
FROM THIRD PART OF NON-WRITE HEADER OPERATION FIGURE 11
INTERNAL CRC/ECC ENABLED ?
YES
NO
INTERNAL CRC/ECC CORRECT
YES
NO
SET HEADER FAULT
EXTERNAL HEADER ECC COUNT = 0 ?
YES
NO
EXTERNAL HEADER ECC COUNT = 0 ?
NO
DEASSERT READ GATE FOR HEADER POSTAMBLE
G
EXECUTE DATA OPERATION FIGURE 13
EXTERNAL HEADER ECC VERIFIED
NO
EXTERNAL HEADER ECC COUNT = 0
NO
YES
EXTERNAL HEADER ECC VERIFIED ?
NO
YES
ALL HEADER BYTES MATCHED ?
YES
NO
SET HEADER FAULT
RESET HEADER FAULT IF SET
DEASSERT READ GATE FOR HEADER POSTAMBLE
D
EXECUTE COMPLETION FOR HEADER OPERATION FIGURE 6
SECTOR NUMBER MATCH RECORDED
NO
YES
ABORT H-FASM ERROR
COMPARE HEADER CHECK DATA OK ?
YES
RESET FIFO
ALL HEADER BYTES MATCHED
NO
YES
DEASSERT READ GATE
R
RETRY HEADER OPERATION FIGURE 7

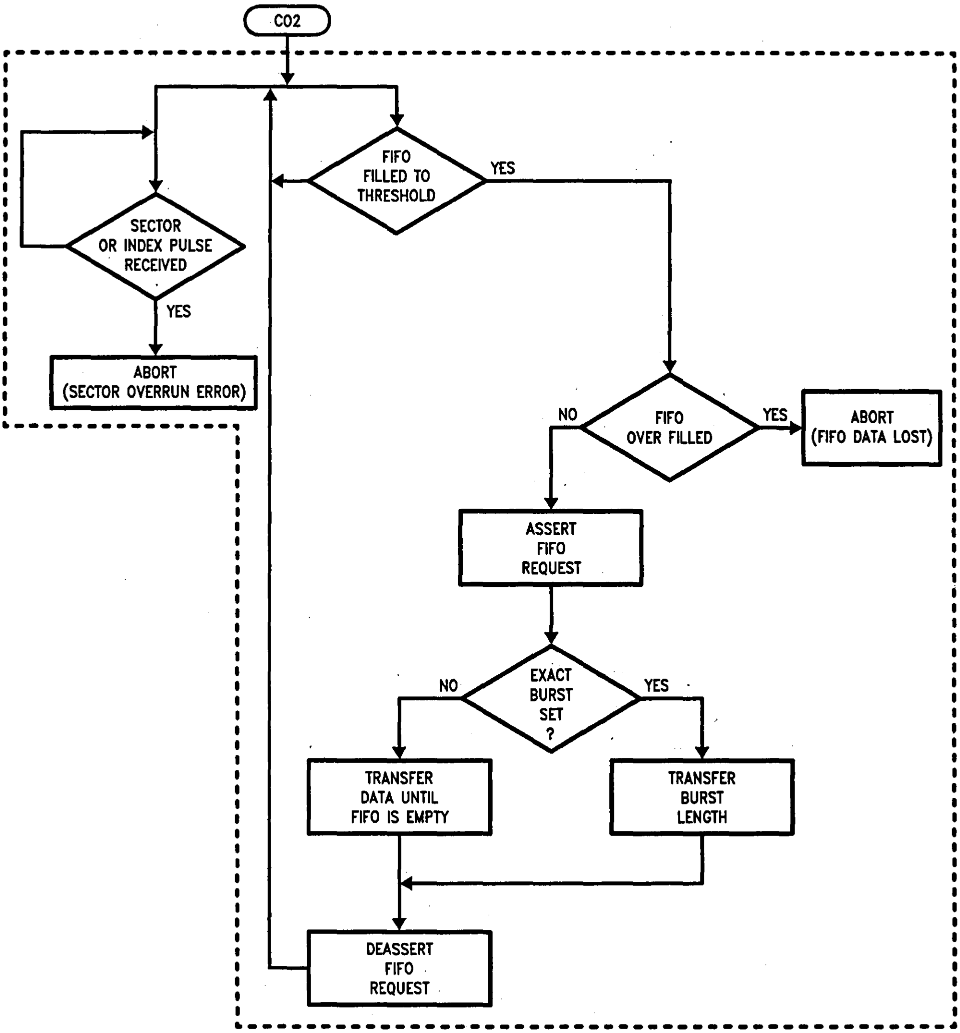
TL/F/8663-H3

FIGURE A.10. Flow Chart for Fourth Part of Non-Write Header Operation (Except Compare Header)


EXECUTE A READ HEADER OPERATION
L
LOAD HEADER BYTES INTO FIFO
C02
CONCURRENT OPERATION
INTERNAL HEADER CRC/ECC ENABLE?
YES
NO
INTERNAL HEADER CRC/ECC VERIFIED?
YES
NO
EXTERNAL HEADER ECC COUNT = 0?
G
EXECUTE DATA OPERATION FIGURE 13
NO
EXTERNAL ECC COUNT = 0?
NO
EXTERNAL HEADER ECC VERIFIED?
YES
SET HEADER FAULT
ABORT
DEASSERT READ GATE FOR HEADER POSTAMBLE
D
EXECUTE COMPLETION OF HEADER OPERATION FIGURE 6

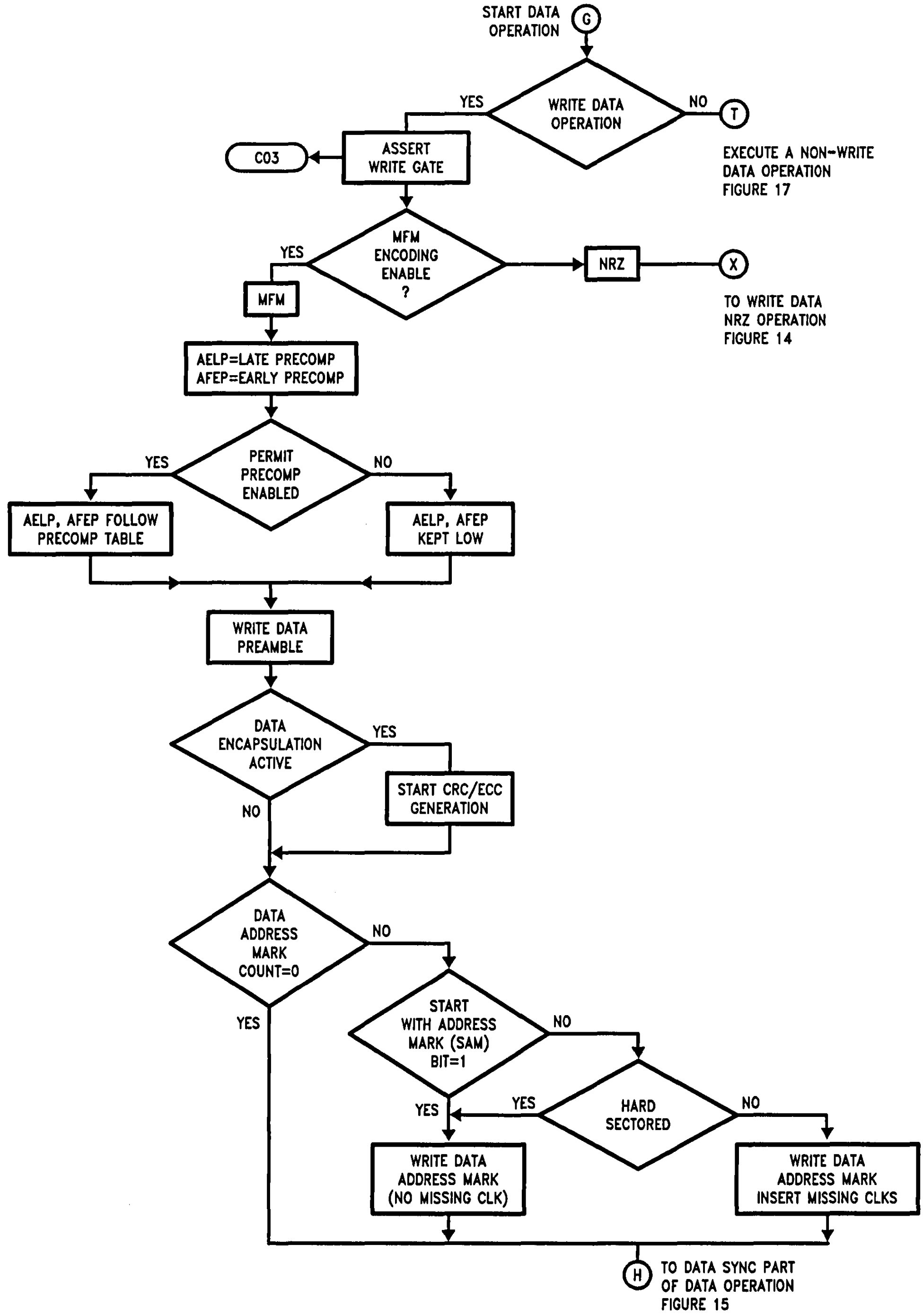
TL/F/8663-H4

FIGURE A.11. Flow Chart for Compare Header Operation



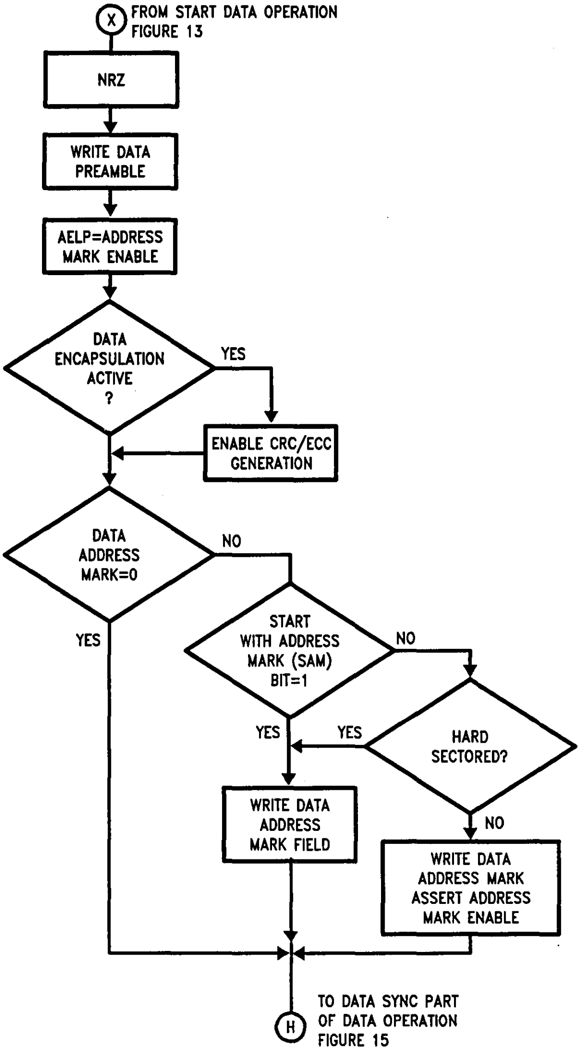
TL/F/8663-H5

FIGURE A.12. Flow Chart for Compare Header Concurrent Operation


START DATA OPERATION
G
WRITE DATA OPERATION
YES
NO
T
EXECUTE A NON-WRITE DATA OPERATION FIGURE 17
C03
ASSERT WRITE GATE
MFM ENCODING ENABLE ?
YES
MFM
NRZ
X
TO WRITE DATA NRZ OPERATION FIGURE 14
AELP=LATE PRECOMP
AFEP=EARLY PRECOMP
PERMIT PRECOMP ENABLED
YES
NO
AELP, AFEP FOLLOW PRECOMP TABLE
AELP, AFEP KEPT LOW
WRITE DATA PREAMBLE
DATA ENCAPSULATION ACTIVE
YES
NO
START CRC/ECC GENERATION
DATA ADDRESS MARK COUNT=0
NO
YES
START WITH ADDRESS MARK (SAM) BIT=1
NO
YES
HARD SECTORED
YES
NO
WRITE DATA ADDRESS MARK (NO MISSING CLK)
WRITE DATA ADDRESS MARK INSERT MISSING CLKS
H
TO DATA SYNC PART OF DATA OPERATION FIGURE 15

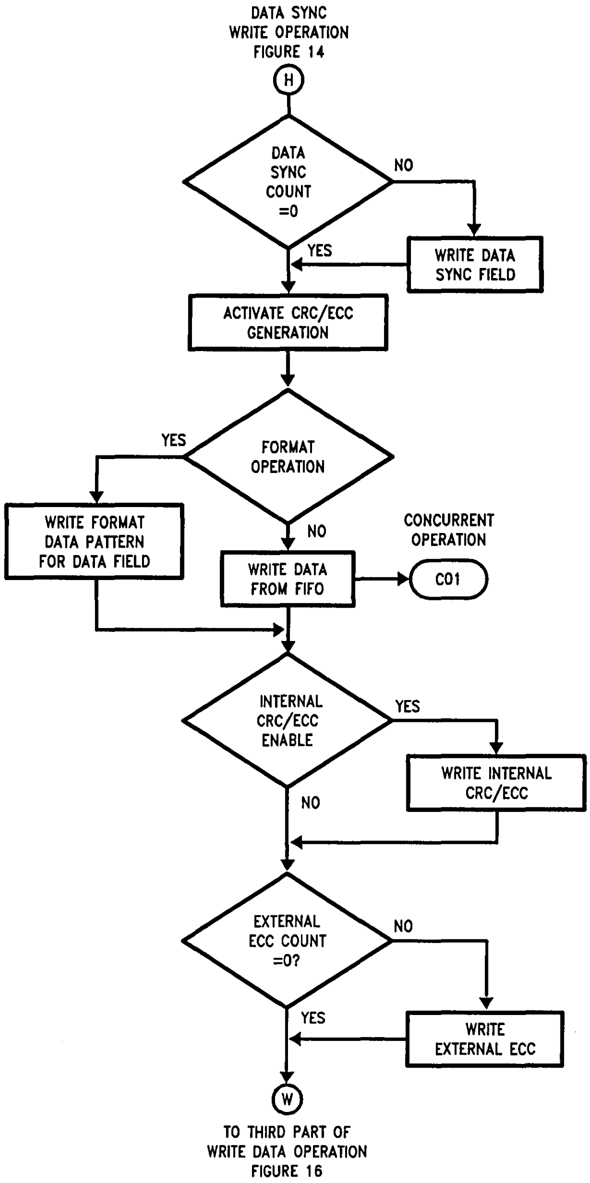
TL/F/8663-H6

FIGURE A.13. First Part of Data Operation Flow Chart, MFM Mode (Up to Data Sync Field)


X
FROM START DATA OPERATION FIGURE 13
NRZ
WRITE DATA PREAMBLE
AELP=ADDRESS MARK ENABLE
DATA ENCAPSULATION ACTIVE ?
YES
ENABLE CRC/ECC GENERATION
DATA ADDRESS MARK=0
NO
YES
START WITH ADDRESS MARK (SAM) BIT=1
NO
YES
YES
HARD SECTORED?
NO
WRITE DATA ADDRESS MARK FIELD
WRITE DATA ADDRESS MARK ASSERT ADDRESS MARK ENABLE
H
TO DATA SYNC PART OF DATA OPERATION FIGURE 15

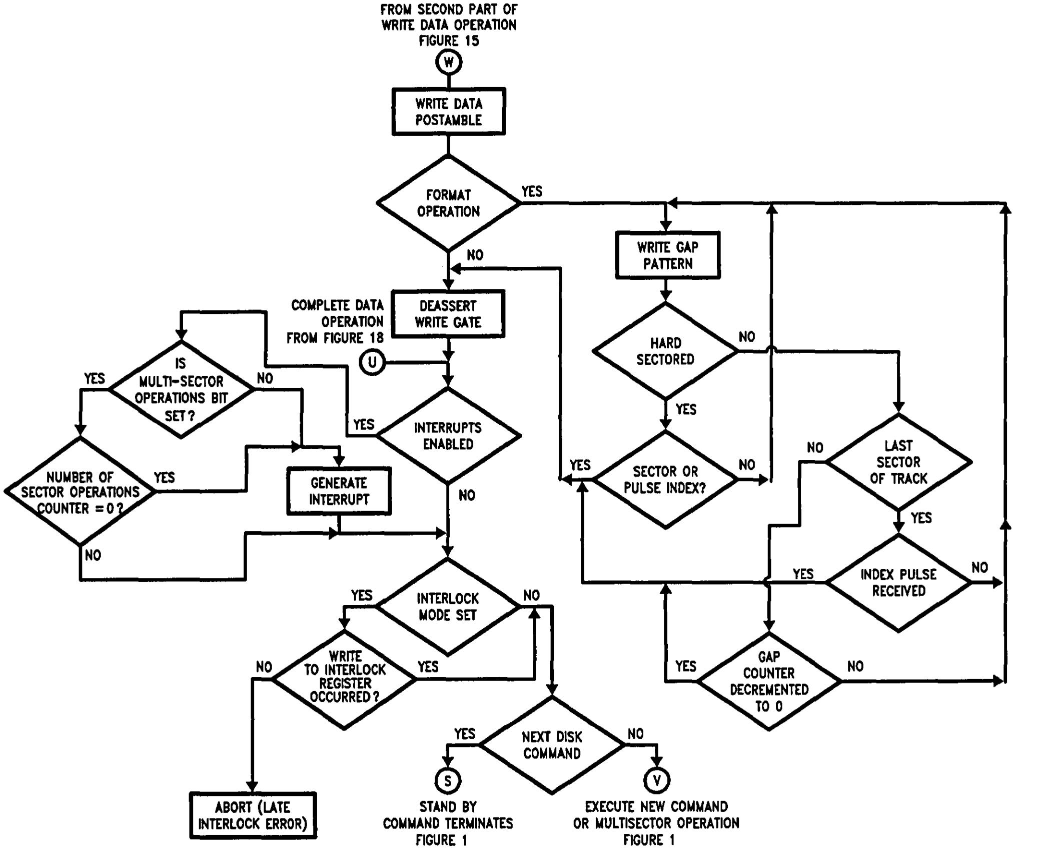
TL/F/8663-H7

FIGURE A.14. First Part of Data Operation Flowchart, NRZ Mode (Up to Data Sync Field)



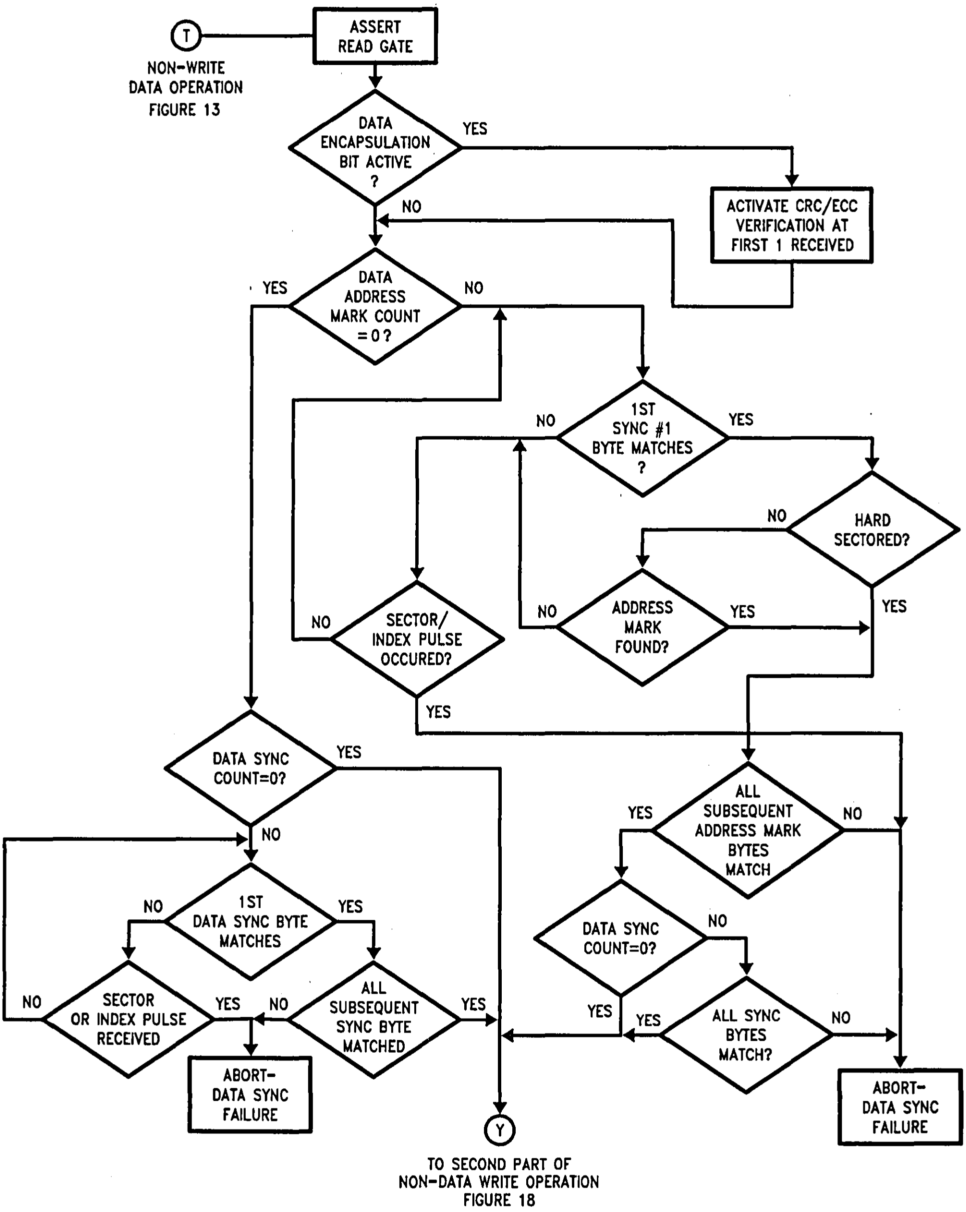
TL/F/8663-H8

FIGURE A.15. Second Part of Data Operation (for Write Data Operation)



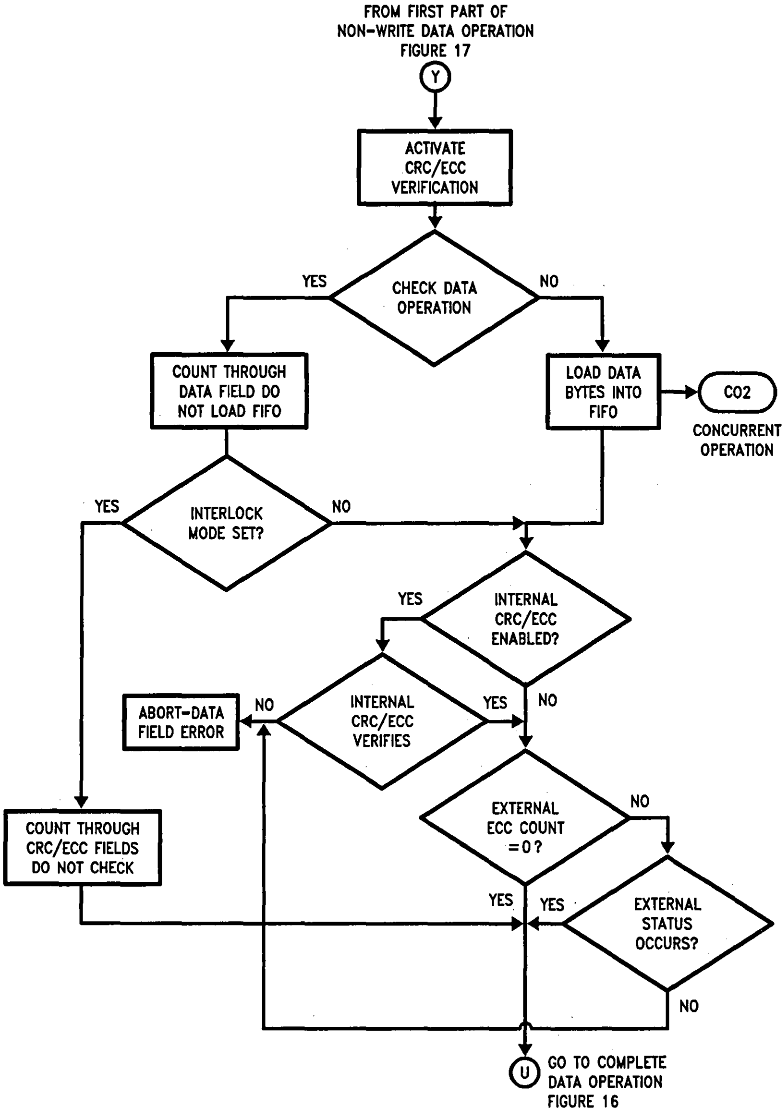
TL/F/8663-H9

FIGURE A.16. Flow Chart for Third Part of Write Data Operation



TL/F/8663-10

FIGURE A.17. First Part of Data Operation Flow Chart for Non-Write Operation


FROM FIRST PART OF
NON-WRITE DATA OPERATION
FIGURE 17
Y
ACTIVATE CRC/ECC VERIFICATION
CHECK DATA OPERATION
YES
NO
COUNT THROUGH DATA FIELD DO NOT LOAD FIFO
LOAD DATA BYTES INTO FIFO
C02
CONCURRENT OPERATION
INTERLOCK MODE SET?
YES
NO
INTERNAL CRC/ECC ENABLED?
YES
NO
ABORT-DATA FIELD ERROR
INTERNAL CRC/ECC VERIFIES
NO
YES
EXTERNAL ECC COUNT =0?
NO
YES
COUNT THROUGH CRC/ECC FIELDS DO NOT CHECK
EXTERNAL STATUS OCCURS?
YES
NO
U
GO TO COMPLETE DATA OPERATION FIGURE 16

TL/F/8663-11

FIGURE A.18. Second Part of Data Operation Flowchart for Non-Write Operation

Section 2
Winchester Disk Support

Section Contents

National Semiconductor Corporation

DP8460 Data Separator/DP8450 Data Synchronizer

General Description

The DP8460 Data Separator is designed for application in disk drive memory systems, and depending on system requirements, may be located either in the drive or in the controller. It receives digital pulses from a pulse detector circuit (such as the DP8464 Disk Pulse Detector), if the DP8460 is situated in the drive, or from an ST506 type interface if it is situated in the controller. After locking on to the frequency of these input pulses, it separates them into synchronized data and clock signals. If the input pulses are MFM encoded data, the data is made available as decoded NRZ data to be deserialized directly by a controller (such as the DP8466 Disk Data Controller). If a run-length-limited code is used, the synchronized data output is available to allow external circuitry to perform the data decoding function. All of the digital input and output signals are TTL compatible and only a single +5V supply is required. The chip is housed in a standard narrow 24-pin dual-in-line package (DIP) and is fabricated using Advanced Schottky bipolar analog and digital circuitry. This high speed I.C. process allows the chip to work with data rates up to 15Mbit/sec.

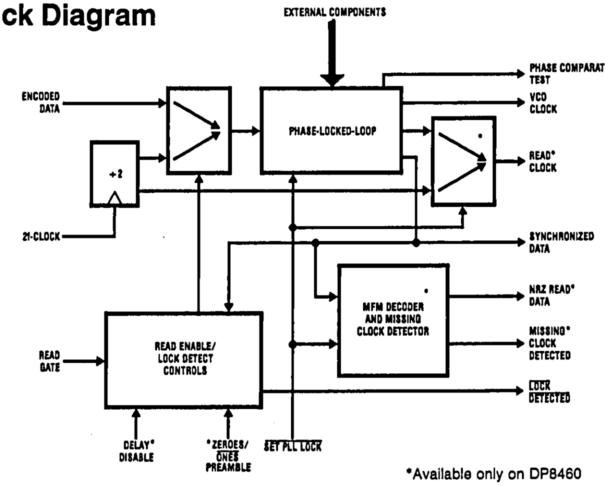
The DP8460 features a phase-lock-loop (PLL) consisting of a pulse gate, phase comparator, charge pump, buffering amplifier, and voltage-controlled-oscillator (VCO). Pins are provided for the user to select the values of the external filtering components required for the pulse gate and amplifier, the frequency setting components required for the VCO, and two current setting resistors for the charge pump. The DP8460 has been designed to indicate lock on to the incoming preamble data pattern once two bytes of synchronized preamble has been detected using a high rate of charge pump current. Once lock-on has been achieved, the charge pump switches to a lower rate (both rates being determined by the external resistors) to maintain stability for the remainder of the read operation. At this time the READ CLOCK output switches, without glitching, from half the 2f-CLOCK frequency to half the VCO CLOCK frequency. After lock-on, with soft sectored disks, the MISSING CLOCK DETECTED output indicates when a missing MFM clock bit in an address mark field occurs so the controller can align byte boundaries to begin deserialization of the incoming data.

The DP8450 incorporates only the data synchronization function of the DP8460. The READ CLOCK generating circuitry, MFM Decoder, Missing Clock Detector, and Read Enable Delay are not included in the DP8450 which is packaged in a 20 pin DIP. Users who do not need these functions and only require the SYNCHRONIZED DATA OUTPUT along with the VCO OUTPUT may employ the DP8450 as an alternative to the DP8460.

Features

- Operates at data rates up to 15Mbit/sec
- Separates MFM data into read clock and serial NRZ data
- 4 byte preamble-lock indication capability
- Preamble recognition of MFM encoded "0"s or "1"s
- User-determined PLL loop filter network
- PLL charge pump has two user-determined tracking rates
- External control of track rate switchover
- No glitch on READ CLOCK at switchover
- Synchronized data provided as an output (for external data separation)
- ORed phase comparator outputs for monitoring bit-shift
- Missing clock detected for soft sectored disks
- Less then ½W power consumption
- Standard narrow 24-pin DIP
- Single +5V supply

Simplified Block Diagram

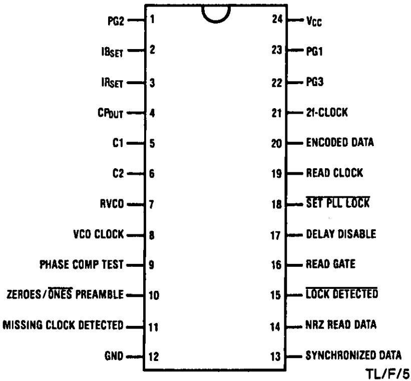

EXTERNAL COMPONENTS
ENCODED DATA
÷2
2f-CLOCK
PHASE-LOCKED-LOOP
PHASE COMPARATOR TEST
VCO CLOCK
READ* CLOCK
SYNCHRONIZED DATA
MFM DECODER AND MISSING CLOCK DETECTOR
NRZ READ* DATA
MISSING* CLOCK DETECTED
READ ENABLE/ LOCK DETECT CONTROLS
READ GATE
$\overline{\text{LOCK DETECTED}}$
DELAY* DISABLE
*ZEROES/ $\overline{\text{ONES}}$ PREAMBLE
$\overline{\text{SET PLL LOCK}}$

*Available only on DP8460

TL/F/5182-1

Connection Diagrams

DP8460 Dual-In-Line Package



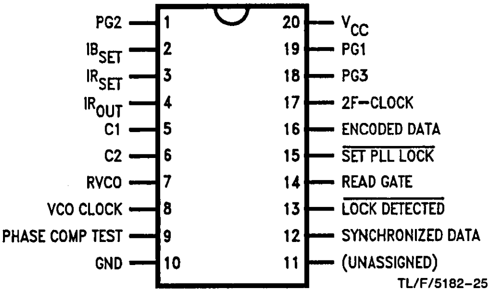
TL/F/5182-2

Top View

Order Number DP8460N-4

See NS Package N24C

DP8450 Dual-In-Line Package



TL/F/5182-25

Top View

Order Number DP8450N-4

See NS Package N20A

Pin Descriptions*

Power Supply

24 V_{CC} +5V ±5%

12 Ground

TTL Level Logic Inputs

16 READ GATE: This is an active high input signal that sets the DP8460 Data Separator into the Read Mode.

17 DELAY DISABLE: This input determines the delay from READ GATE going high to the time the DP8460 enters the Read Mode. If DELAY DISABLE is set high, this delay is within one cycle of the 2f-CLOCK signal. If DELAY DISABLE is set low, the delay is thirty two-cycles of the 2f-CLOCK, as shown in *Figure 1*.

18 $\overline{\text{SET PLL LOCK}}$: This input allows the user to determine when the on-chip PLL will go into the low track rate. A high level at this input results in the PLL being in the high track rate. If this input is connected to the $\overline{\text{LOCK DETECTED}}$ output, the PLL will go into the low track rate mode immediately after lock is detected.

10 ZEROES/$\overline{\text{ONES}}$ PREAMBLE: A high level on this input enables the circuit to recognize an All Zeros MFM data preamble. A low level results in the recognition of an All Ones MFM data preamble.

20 ENCODED DATA: This input is connected to the output of the head amplifier/pulse-detecting network located in the disk drive. Each positive edge of the ENCODED DATA waveform identifies a change of flux on the disk. In the case of MFM encoded data, the input will be raw MFM.

21 2f-CLOCK: This is a system clock input, which is either a signal generated from the servo track (for systems utilizing servo tracks), or a signal buffered from a crystal. 2f CLOCK MUST ALWAYS BE APPLIED TO THIS INPUT FOR PROPER OPERATION.

TTL Level Logic Outputs

8 VCO CLOCK: This is the output of the on-chip VCO, transmitted from an Advanced Schottky-TTL buffer. It is synchronized to the SYNCHRONIZED DATA output, for use with external data separation circuitry.

15 $\overline{\text{LOCK DETECTED}}$: This output goes active low only after both PLL Lock has occurred and 16 pulses of the preamble pattern have been recognized. It remains low until READ GATE goes inactive.

14 NRZ READ DATA: This is the NRZ decoded data output, whose leading edges coincide with the trailing edge of READ CLOCK.

13 SYNCHRONIZED DATA: This output is the same encoded data that is input to the chip, but is synchronous with the negative edge of the VCO CLOCK.

11 MISSING CLOCK DETECTED: When an MFM missing clock is detected, this output will be a single pulse (of width equal to one cycle of READ CLOCK) occurring as shown in *Figure 2*.

19 READ CLOCK: This is half VCO CLOCK frequency when $\overline{\text{SET PLL LOCK}}$ is low; it is half 2f-CLOCK frequency at all other times. A deglitcher is utilized to ensure that no short clock periods occur during either switchover.

9 PHASE COMP TEST: This output is the logical "OR" of the Phase Comparator outputs, and may be used for the testing of the disk media or for data separator performance analysis.

Analog Signals

23, 22, PG1, PG3: The external capacitors and resistor of the Pulse Gate filter are connected to these pins. PG1 should be connected directly to the ground pin, pin 12.

1 PG2: This is the Pulse Gate current supply.

3 IRSET: The current into the rate set pin (V_{BE}/R_{Rate}) is half the charge pump output current for the low tracking rate.

2 IBSET: The current into the boost set pin (V_{BE}/R_{Boost}) is half the amount by which the charge pump current is increased for the high tracking rate. ($I_{HIRATE} = I_{RATE}$ Set + I_{BOOST} Set).

4 CPOUT: CHARGE PUMP OUT/BUFFER AMP IN is available for connection of external filter components, for the phase-lock-loop. In addition to being the charge pump output node, this pin is also the noninverting input to the opamp of the Buffer Amplifier.

7 RVCO: The current into this pin determines the operating currents within the VCO.

5, 6 VCO C1, C2: An external capacitor connected across these pins sets the nominal VCO frequency.

* Pin number designations apply only to the DP8460. See Connection Diagram for DP8450.

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage	7V
TTL Inputs	7V
Output Voltages	7V
Input Current (CPOUT, IRSET, IBSET, RVCO)	2mA
Storage Temperature	−65°C to 150°C

Operating Conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{CC}	Supply Voltage		4.75	5.00	5.25	V
T_A	Ambient Temperature		0	25	70	°C
I_{OH}	High Logic Level Output Current	V_{CO} Clock Others			−2000 −400	μA
I_{OL}	Low Logic Level Output Current	V_{CO} Clock Others			20 8	mA
f_{DATA}	Input Data Rate		2.0		15	Mbit/sec
t_{WCK}	Width of 2f-CLOCK, High or Low		10			ns
t_{WPD}	Width of ENCODED DATA Pulse (Note 2)	High	5 ns + 0.10t			ns
		Low	0.4t			
V_{IH}	High Logic Level Input Voltage		2			V
V_{IL}	Low Logic Level Input Voltage				0.8	V
t_{SETUP} (Read Gate)	Minimum Amount of Time Which a Positive Edge of Read Gate Must Precede a Negative Edge of V_{CO} (Pin 8)		20			ns
t_{HOLD} (Read Gate)	Minimum Time Required for a Positive Edge of Read Gate to Be Held after a Negative edge of V_{CO} (Pin 8)		10			ns

DC Electrical Characteristics Over Recommended Operating Temperature Range

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IC}	Input Clamp Voltage	V_{CC} = Min., $I_I = -18$ mA			−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min., I_{OH} = Max.	$V_{CC}-2V$	$V_{CC}-1.6V$		V
V_{OL}	Low Level Output Voltage	V_{CC} = Min., I_{OL} = Max.			0.5	V
I_{IH}	High Level Input Current	V_{CC} = Max., $V_I = 2.7V$			20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max., $V_I = 0.4V$			−200	μA
I_O	Output Drive Current	V_{CC} = Max., $V_O = 2.125V$ (Note 1)	−20		−110	mA
I_{CC}	Supply Current	V_{CC} = Max.			100	mA
I_{OUT}	Charge Pump Output Current	$I_{RSET} = V_{BE}/R_{RATE}$ $I_{BSET} = V_{BE}/R_{BOOST}$	−10% −10%	$1.7 \times I_{RSET}$ $1.8 \times (I_{RSET}+I_{BSET})$	+10% +10%	mA

Note 1. This value has been chosen to produce a current that closely approximates one-half of the true short-circuit output current. I_{OS}.

Note 2. t is defined as the period of the encoded data $t = \frac{2}{F_{VCO}}$.

AC Electrical Characteristics (Over Recommended V_{CC} and Operating Temperature Range.)

(All Parts unless stated otherwise) ($t_R = t_F = 2.0$ ns, $V_{IH} = 3.0V$, $V_{IL} = 0V$) (Note 1)

Symbol	Parameter	Min	Typ	Max	Units
t_{READ}	Positive READ CLOCK transitions from READ GATE set active until PLL Lock sequence begins (DELAY DISABLE low)		16	17	—
t_{READ}	Positive READ CLOCK transitions from READ GATE set active until PLL Lock sequence begins (DELAY DISABLE high)		1	1	—
$t_{DECODE\ NRZ}$	Number of READ CLOCK cycles required to output each decoded MFM data bit (Note 4)	—	2	3	T-clock
$t_{TRANSMIT\ MFM}$	Positive READ CLOCK transitions required to transmit input MFM to output	1	2	3	—
$t_{READ\ ABORT}$	Number of READ CLOCK cycles after READ GATE set low to read operation abort			2	T-clock
t_{WINDOW}	Variance of center of decode window from nominal (Note 7, 8) DP8460-4			10	ns
$\phi_{LINEARITY}$	Phase range for charge pump output linearity (Note 2)	$-\pi$		$+\pi$	Radians
K_1	Phase Comparator — Charge Pump gain constant (Note 5) ($N = f_{VCO}/f_{INPUT\ DATA}$, $2 \leq N \leq 4$ for MFM)		$\frac{1.78V_{BE}}{N2\pi R}$		Amps/rad
$V_{CONTROL}$	Charge pump output voltage swing from nominal		±100		mV
K_{VCO} $(=A \times K_2)$	VCO gain constant (ω_{VCO} = VCO center frequency in rad/s) (Note 6)	$\frac{1.20\omega_C}{V_{BE}}$	$\frac{1.40\omega_C}{V_{BE}}$	$\frac{1.60\omega_C}{V_{BE}}$	rad/sec. V
f_{VCO}	VCO center frequency variation over temperature and V_{CC}	−5		+5	%
$f_{MAX\ VCO}$	VCO maximum frequency	50			MHz
t_{HOLD}	Time READ CLOCK is held low during changeover after lock detection has occurred (Note 3)			1½	T-clock
t_{PHL}	Prop. delay. V_{CO} negative edge to synchronized data negative edge		15	30	ns
t_{PLH}	Prop. delay. V_{CO} negative edge to synchronized data positive edge		10	25	ns
$t_{2F/RC}$	Delay from 2f positive transition to READ CLOCK positive or negative transition ($\overline{\text{SET PLL LOCK}}$ high)	10		35	ns

Note 1. A sample calculation of frequency variation vs. control voltage: $V_{IN} = \pm 0.1V$;

$$K_{VCO} = \frac{\omega OUT}{V_{IN}} = \frac{0.4\omega_C}{0.2V} = \frac{2.0\omega_C}{V} \frac{(rad/sec)}{(volt)}$$

Note 2. $-\pi$ to $+\pi$ with respect to 2f VCO CLOCK

Note 3. T-clock is defined as the time required for one period of the READ CLOCK to occur.

Note 4. This number remains fixed after PLL Lock occurs.

Note 5. With respect to VCO CLOCK; $I_{PUMP\ OUT} = 1.7\ I_{SET}$

$$I_{SET} = \frac{V_{BE}}{R_{SET}}$$

Note 6. Although specified as the VCO gain constant, this is the gain from the Buffer Amplifier input to the VCO output.

Note 7. τ is defined as the period of the incoming data stream.

Note 8. This specification is guaranteed only for the conditions under which the parts were tested. However, significant variation from the formula is not expected for other data rates and filters. The filter values below were chosen for operation in an automatic test system (static window) environment. Different criteria may apply for choosing filter values in a disk system. See Loop Filter section for sample calculations of other filter values:

		Filter				
Part Type	**Data Rate Tested**	**C_1**	**C_2**	**R_1**	**R_{RATE}**	**R_{BOOST}**
DP8450-4	5 MBit/sec	0.02 µF	150 pF	200Ω	750Ω	1.6K
DP8460-4	5 MBit/sec	0.02 µF	150 pF	200Ω	750Ω	1.6K

Note: For further information refer to Application Note AN-414.

External Component Selection (All Parts) (Note 1)

Symbol	Component	Min	Typ	Max	Unit
R_{VCO}	VCO Frequency Setting Resistor (Note 2)	990		1010	Ω
C_{VCO}	VCO Frequency Setting Capacitor (Note 3,4)	28		245	pF
R_{RATE}	Charge Pump I_{RATE} Set Resistor (Note 6)	0.4		4.0	kΩ
R_{BOOST}	Charge Pump (High Rate) I_{BOOST} Resistor (Note 6)	0.5		∞	kΩ
C_R	I_{RATE} Bypass Capacitor (Note 5)	.01			μF
C_B	I_{BOOST} Bypass Capacitor (Note 5)	.01			μF

Note 1. External component values for the Loop Filter and Pulse Gate are given in their respective sections, following.

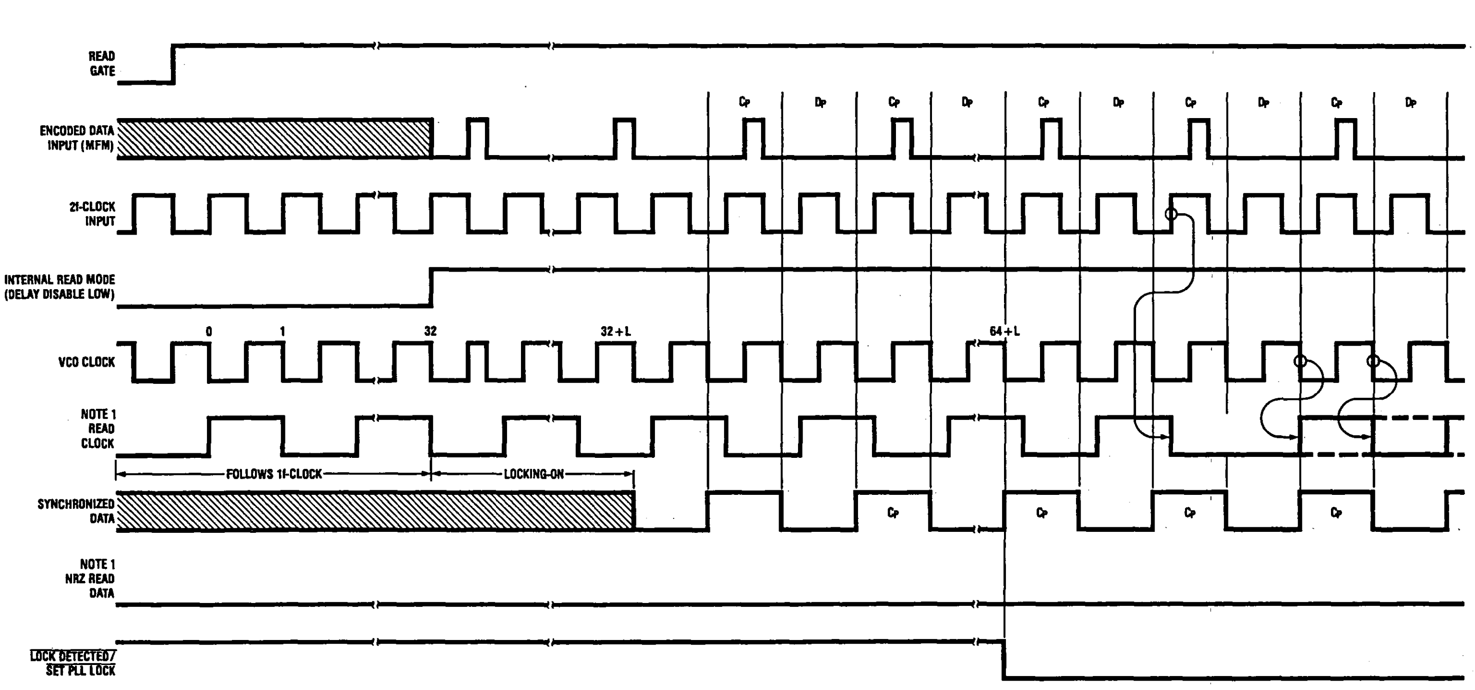
Note 2. A 1% Component Tolerance is Required.

Note 3. These MIN and MAX values correspond to the MAX and MIN data rates respectively.

Note 4. The Component Tolerance is system dependent on how much center frequency deviation can be tolerated.

Note 5. Component Tolerance 15%.

Note 6. The minimum value of the parallel combination of R_{RATE} and R_{BOOST} is 350Ω.



TL/F/5182-3

Note 1. Not included on the DP8450.

C_P, D_P = preamble clock and preamble data bits respectively.

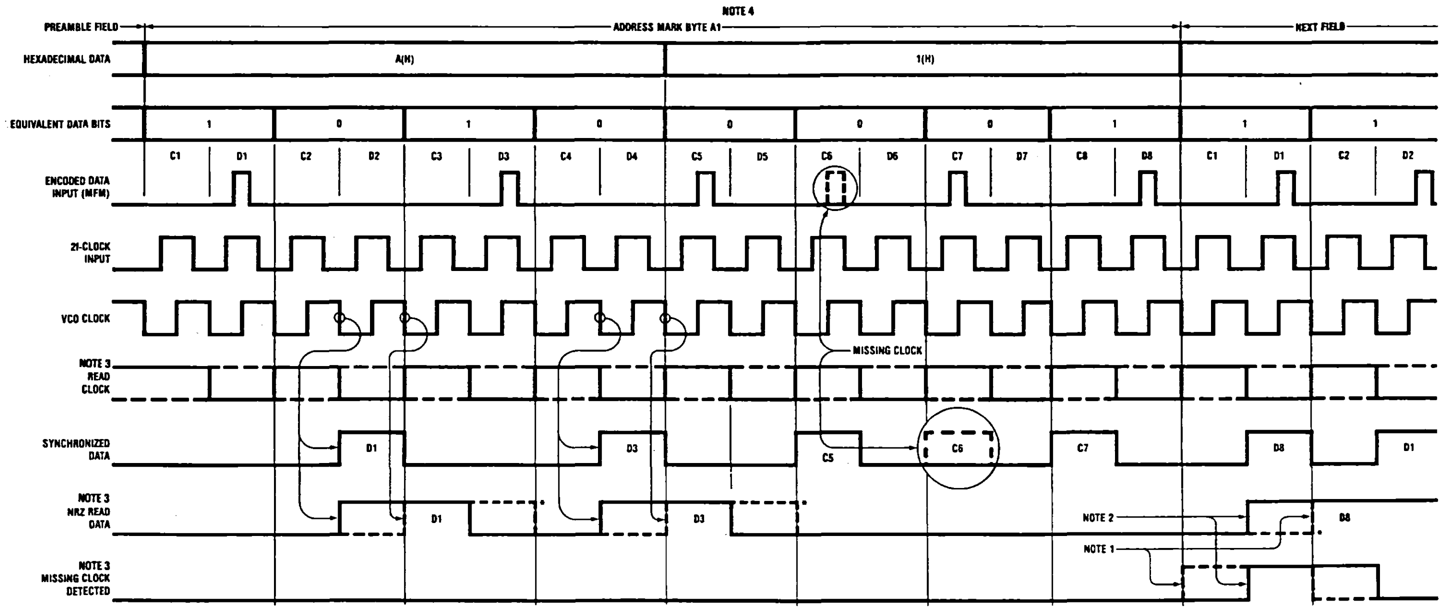
L = Number of 2f-clock cycles required for VCO to lock (typically ≈ 20 2f-clock cycles), but determined by external component values.

At 32 + L, VCO has just locked.

At 64 + L, circuit has confirmed lock (has been in lock for 16 MFM clock bits). This sequence shows the MFM all-zeros preamble pattern.

For DP8450 DELAY DISABLE does not exist and part functions as if this input is always high.

FIGURE 1. Lock-on Sequence Waveform Diagram



TL/F/5182–4

* READ CLOCK and NRZ READ DATA may be delayed by one VCO clock period depending on the phase of the internal clock at activation of READ GATE input.

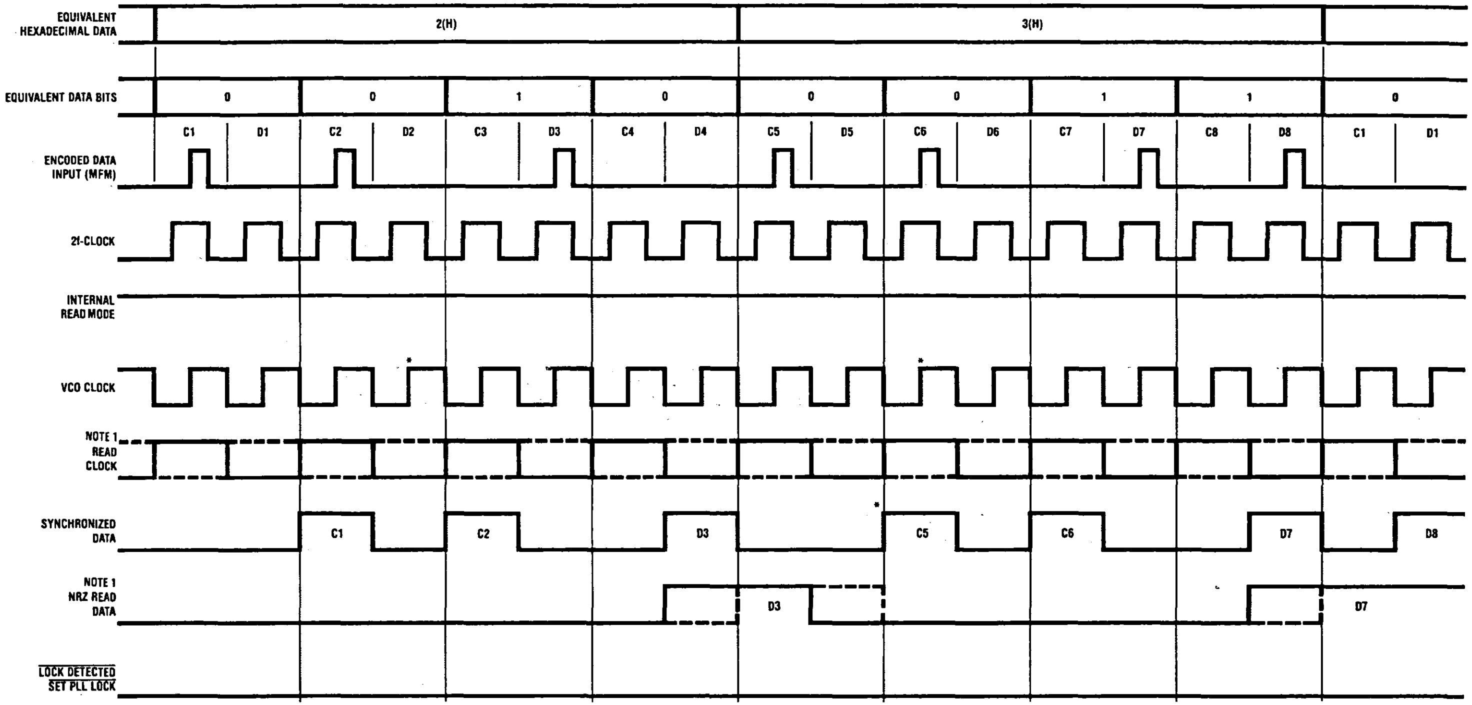
① MISSING CLOCK DETECTED is one READ CLOCK period ahead of the chip issuing D8 on the NRZ READ DATA output when READ CLOCK is delayed by one VCO clock period

② MISSING CLOCK DETECTED is synchronous with the chip issuing D8 on the NRZ READ DATA Output when READ CLOCK is not delayed

③ Not included on the DP8450

④ The A_1 byte is shown as an example only. Any missing clock bit between two adjacent clock bits will be detected.

FIGURE 2. Missing Clock Detection Waveform Diagram

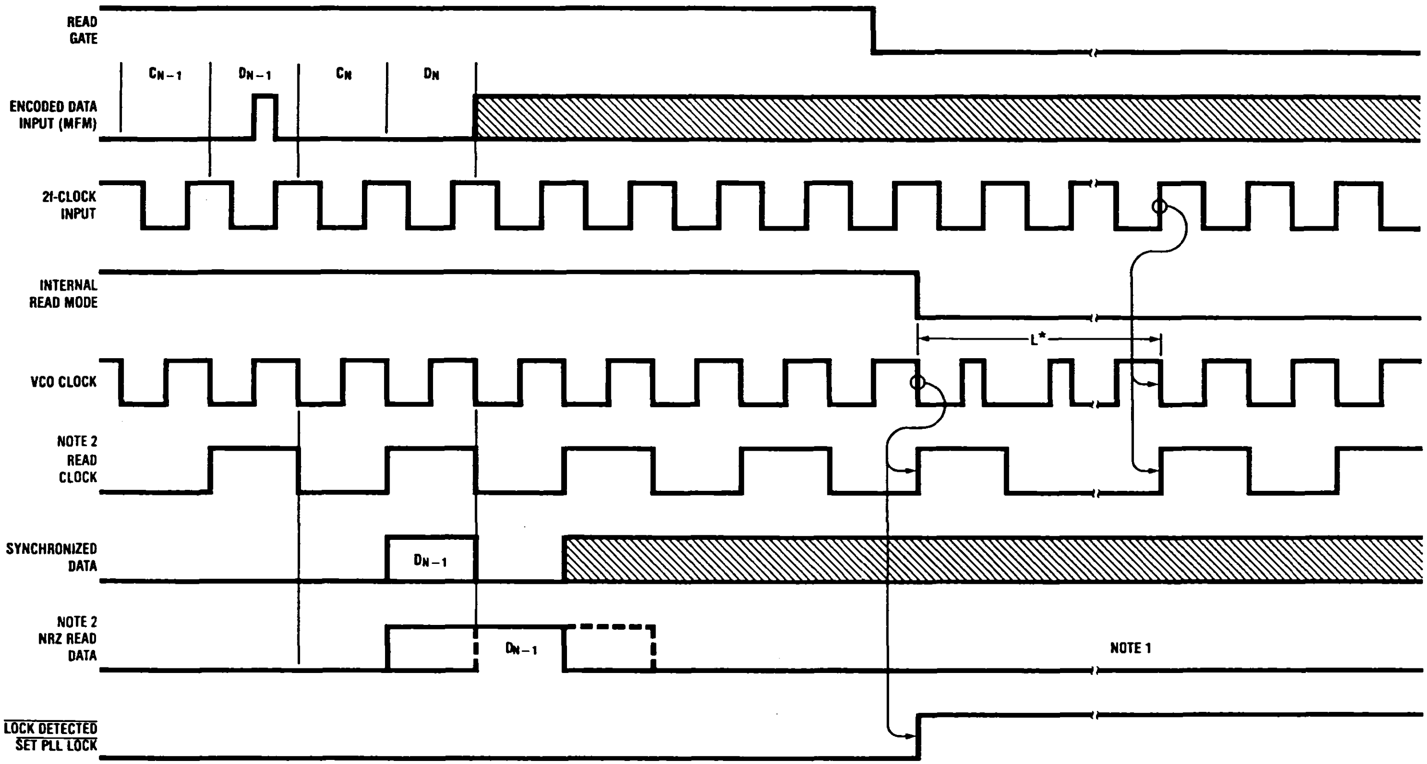


TL/F/5182–5

*** READ CLOCK and NRZ READ DATA may be delayed by one VCO clock period with respect to Synchronized Data depending on the phase of the internal clock at activation of READ GATE input.**

Note 1. Not included on the DP8450.

FIGURE 3. Locked-on Waveform Diagram



TL/F/5182-6

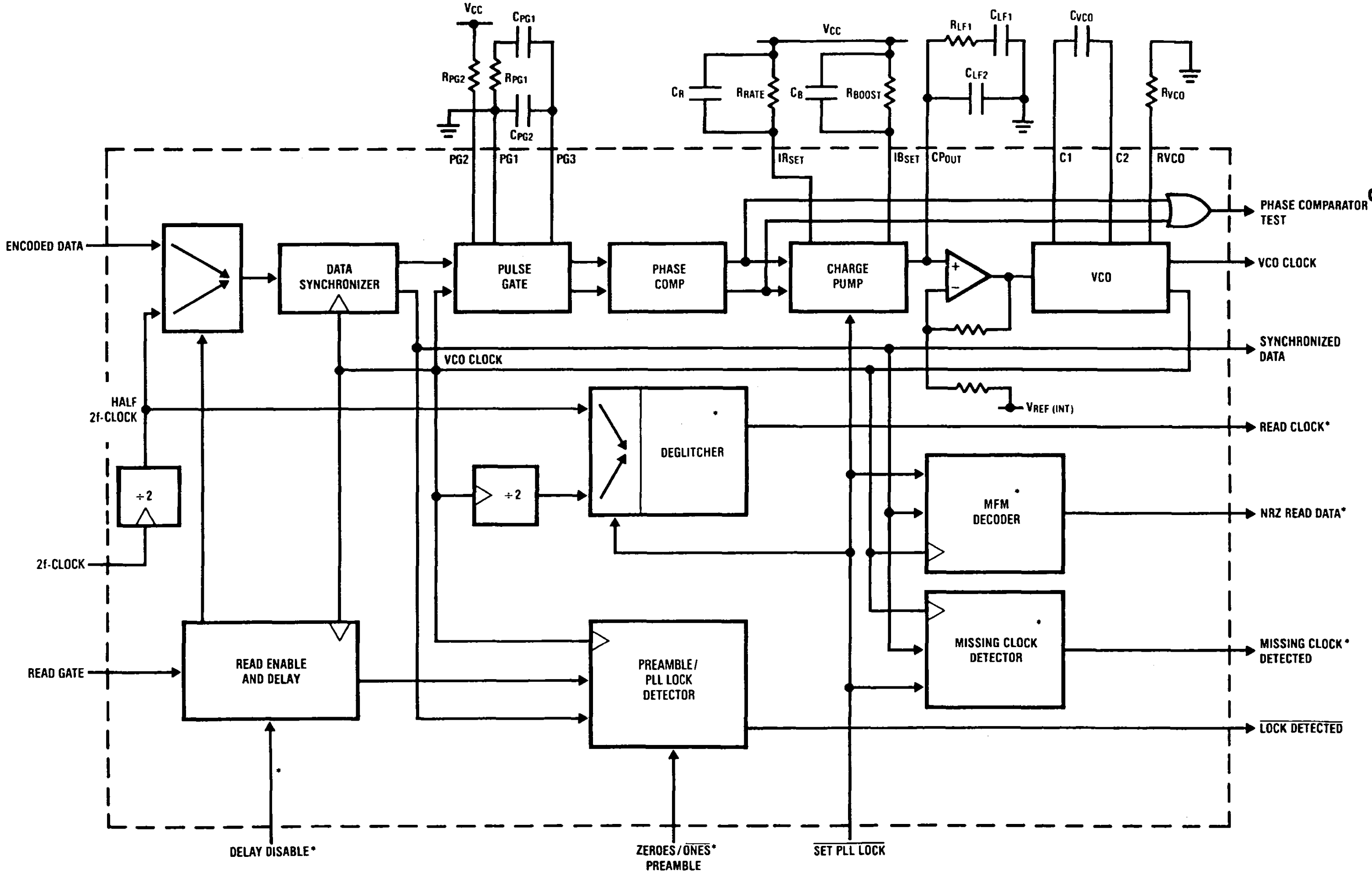
*** L indicates the number of cycles required for the VCO to lock to the 2f-CLOCK**

Note 1: READ GATE going low will always result in NRZ READ DATA going low regardless of the state of the last bit

Note 2: Not included on the DP8450

FIGURE 4. Lock-Ending Sequence Waveform Diagram

Detailed Block Diagram



TL/F/5182-7

*Not included on the DP8450

Circuit Operation

When the READ GATE input goes high, the DP8460 Data Separator enters the read mode after a period determined by the state of the DELAY DISABLE pin. This may be either one or thirty two 2f-CLOCK cycles. Referring to *Figure 1*, once in the read mode, the phase-locked-loop reference signal is switched from 2f-CLOCK input to the ENCODED DATA input. The PLL, initially in the high-tracking rate mode, then attempts to lock onto the incoming encoded data stream. By careful selection of the loop filter components, this can be within 2 bytes. Preamble pattern recognition then can begin. As soon as two bytes of the selected preamble are detected, (the selection is determined by the ZEROES/ONES PREAMBLE pin) the LOCK DETECTED output goes low. In a typical MFM disk drive application, the LOCK DETECTED output is directly connected to the SET PLL LOCK input. With this connection, track rate selection, clock output switchover, and data output enabling will occur at this time.

A low level on the SET PLL LOCK causes the PLL Charge Pump to switch from the high to low tracking rate. At the same time the source of the READ CLOCK signal is switched from half the frequency of the 2f-CLOCK to half the VCO clock. The MFM decoder also becomes enabled and begins to output decoded NRZ data. If the preamble is being decoded, and it is a zeroes data preamble, the NRZ READ DATA output will remain low until the end of the preamble. It will then output NRZ data some 2f-CLOCK periods after the preamble field has ended, as shown in *Figures 2* and *3*.

Figure 4 shows the sequence when READ GATE goes low, signifying the end of a read operation. The PLL reference signal is switched back to half the 2f-CLOCK and the LOCK DETECTED output (and therefore the SET PLL LOCK input) goes high. The PLL then returns to the high tracking rate, and the output signals return to their initial conditions.

NOTE: 2f CLOCK must always be applied to the DP8460 for proper operation.

CIRCUIT DESCRIPTION

1. Read Enable and Delay: If the DELAY DISABLE input is connected low, then thirty two VCO cycles after READ GATE goes active, the DP8460 will go into the read mode. If the DELAY DISABLE input is connected high, the chip will go into the read mode one VCO cycle after READ GATE goes active. This feature provides the user with a technique for write-splice avoidance (hard-sectored). This option is not available in the DP8450; it will perform as if the DELAY DISABLE input is HIGH.

2. Pulse Gate, including Input Multiplexer and Data Synchronizer: The Input Multiplexer selects the input to the phase-lock-loop. When the Read Gate is low, the Input Multiplexer feeds the 2f-CLOCK divided by two into the Pulse Gate. The Pulse Gate allows a reference signal from the VCO into the Phase Comparator only when a signal from the 2f-CLOCK divided by two occurs. In the read mode, the Input Multiplexer switches to the ENCODED DATA signal. The VCO CLOCK then begins to synchronize with the ENCODED DATA signal. The Pulse Gate allows a reference signal from the VCO into the Phase Comparator only when a ENCODED DATA bit is valid. The Pulse Gate utilizes a scheme which delays the incoming data by one-half the period of the 2f-CLOCK. This optimizes the position of the decode window and allows input jitter up to ± half the 2f-CLOCK period, assuming no error in the decode window position. The decode window error can be determined from the specification in the Electrical Characteristics Table.

3. Phase Comparator: The Phase Comparator receives its inputs from the Pulse Gate, and is edge-triggered from these inputs to provide charge-up and charge-down outputs.

4. Charge Pump: The high speed charge pump consists of a switchable constant current source and sink. The charge pump constant current is set by connecting external resistors to V_{CC} from the charge current rate set (IRSET) and current boost set (IBSET) pins. Before lock is indicated, the PLL is in the fast tracking rate and both resistors determine the current. In the slow tracking rate after lock-on, only the IRSET resistor determines the charge pump current. The output of the charge pump feeds into external filter components and the Buffer Amplifier.

5. Buffer Amplifier: The input of the Buffer Amplifier is internally connected to the charge pump's constant current source/sink output as well as the external Loop Filter components. The Buffer Amplifier is configured as a high input impedance amplifier which allows for the connection of external PLL filter components to the Charge Pump output pin CPOUT. The output of the Buffer Amplifier is internally connected to the VCO control input.

6. VCO: The Voltage-Controlled-Oscillator requires a resistor from the RVCO pin to ground and a capacitor between pins C1 and C2, to set the center frequency. The VCO frequency can be varied from nominal by approximately ±20%, as determined by its control input voltage.

7. PLL Lock-on/Preamble Pattern Detector: To recognize preamble, the preamble pattern from the disk must consist exclusively of either data bit zeroes (encoded into ..10.. MFM clock pulses) when the ZEROES/ONES PREAMBLE pin is set high, or data bit ones (encoded into ..01.. MFM clock pulses) when set low. The preamble pattern must be long enough to allow the PLL to lock, and subsequently for the Preamble Pattern Detector circuit to detect two complete bytes.

Once the chip is in the read mode, the VCO proceeds to lock on to the incoming data stream. The Preamble Pattern Detector then searches for a continuous pattern of 10101010101010101010101010101010 (16 consecutive pulses at the data rate) to indicate lock has been achieved. The LOCK DETECTED output then goes low.

Any deviation from the above-mentioned one-zero pattern at any time before PLL Lock is detected will reset the PLL Lock Detector. The lock detection procedure will then start again.

8. MFM Decoder: The MFM Decoder receives synchronized MFM data from the Pulse Gate and converts it to NRZ READ DATA. For run-length-limited codes the MFM Decoder and Missing Clock Detector will not be used.

9. Missing Clock Detector: This block is only required for soft-sectored drives, and is used to detect a missing clock violation of the MFM pattern. The missing clock is inserted when writing to soft-sectored disks to indicate the location of the Address Mark in both the ID and the Data fields of each sector. Once PLL Lock has been indicated, the Missing Clock Detector circuit is enabled. MISSING CLOCK DETECTED will go active any time the incoming data pattern contains one suppressed clock bit framed by two adjacent clock bits. It does not search for a specific byte, such as "A1." The output signal goes high for one cycle of READ CLOCK.

Circuit Operation (Continued)

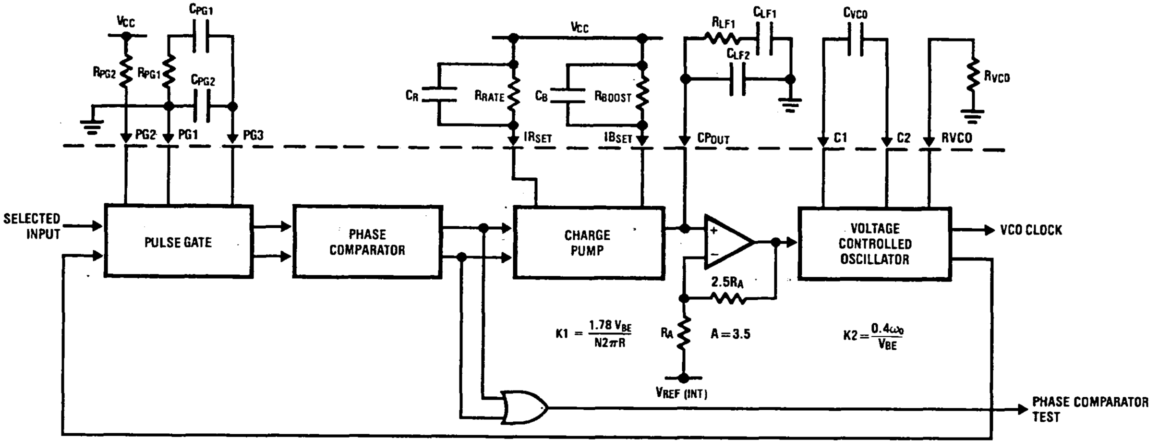
10. Clock Multiplexer and Deglitcher: When the $\overline{\text{SET PLL LOCK}}$ input changes state this circuit switches the source of the READ CLOCK signal between the half 2f-CLOCK frequency and the half VCO CLOCK frequency. A deglitcher circuit is utilized to ensure that no short clock periods occur during either switchover.

BIT JITTER TOLERANCE

The DP8460 −4 part will be used in most 2–5 Mbit/sec data rate applications. As an example, at the 5Mbit/sec data rate the chip therefore contributes up to 10 ns of window error, out of the total allowable error of 50 ns (half the 2f-clock period of 100 ns). This allows the disk drive to have a margin of 40 ns of jitter on the transition position before an error will occur.

ANALOG CONNECTIONS TO THE DP8460

External passive components are required for the Pulse Gate, Charge Pump, Loop Filter and VCO as shown in *Figure 5*. The information provided here is for guidelines only. The user should select values according to his own system requirements. Phase-Locked Loops are complex circuits that require detailed knowledge of the specific system. Factors such as loop gain, stability, response to change of signal, lock-on time, etc are all determined by the external components. In many disk systems these factors are critical, and National Semiconductor recommends the designer be knowledgeable of phase-locked-loops, or seek the advice of an expert. Inaccurate design will probably result in excessive disk error rates. The phase-locked-loop in the DP8460 has many advantages over all but the most sophisticated discrete designs, and if the component values are selected correctly, it will offer significant performance advantages. This should result in a reduction of disk error rates over equivalent discrete designs. Please refer to National Semiconductor Application Note AN414, Precautions for Disk Data Separator Designs, as well as to the Disk Interface Design Guide and User's Manual, Ch. 1.



TL/F/5182-8

FIGURE 5. Phase-Locked-Loop Section

Circuit Operation (Continued)

Pulse Gate

There are four external components connected to the Pulse Gate as shown in *Figure 6* with the associated internal components. The values of R_{PG1}, R_{PG2}, C_{PG1}, and C_{PG2} are dependent on the data rate. R_{PG1} is proportional to the data rate, while R_{PG2}, C_{PG1} and C_{PG2} are inversely proportional. Table I shows component values for the data rates given. Component values are calculated by selecting R_{PG2} from Table I. Next calculate

$$C_{PG1} = \left(\frac{2.12 \times 10^5}{890 + R_{PG2}}\right)\left(\frac{1}{100 \times R_S}\right)^2$$

$$C_{PG2} = \frac{1}{10} C_{PG1}, \text{ and } R_{PG1} = \left(\frac{890 + R_{PG2}}{2.38 \times 10^5}\right)(100 \times R_S).$$

In the above equations R_S is the rotational speed and, for 3600 RPM, R_S = 60Hz. A rotational speed of 3600 RPM was assumed for the calculations in Table I. For data rates not listed, R_{PG2} may be approximated as (30 kΩ/f_{DATA}) −1.20 kΩ = R_{PG2} where f_{DATA} is the data rate in Megabits/second.

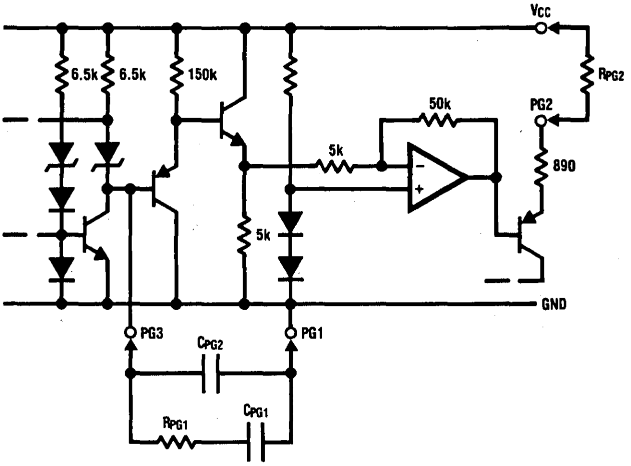
TABLE I. Pulse Gate Component Selection Chart
Components with 10% tolerance will suffice

Data Rate	R_{PG2}	R_{PG1}	C_{PG1}	C_{PG2}
2Mbit/sec	15 kΩ	430Ω	.39 μF	.039 μF
5Mbit/sec	4.7 kΩ	150Ω	1 μF	.1 μF
10Mbit/sec	1.8 kΩ	68Ω	2.2 μF	.22 μF
15Mbit/sec	750Ω	39Ω	3.9 μF	.39 μF

Charge Pump

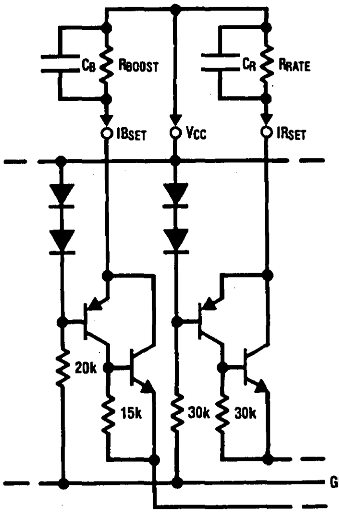
Resistors R_{RATE} and R_{BOOST} determine the charge pump current. The Charge Pump bidirectional output current is approximately (within ±10%) 1.7 × the input current. In the high tracking rate with $\overline{\text{SET PLL LOCK}}$ high, the input current is I_{BSET} + I_{RSET}, ie, the sum of the currents through R_{BOOST} and R_{RATE} from V_{CC}. In the low tracking rate, with $\overline{\text{SET PLL LOCK}}$ low, this input current is I_{RSET} only.

A recommended approach would be to select R_{RATE} first. The External Component Limits table allows R_{RATE} to be 0.4k to 4 kΩ, so for simplicity select R_{RATE} = 820Ω. A typical loop gain change of 2:1 for high to low tracking rate would require R_{BOOST} = R_{RATE} or 820Ω. Referring to *Figure 7* the input current is effectively V_{BE} / R_{RATE} in the low tracking rate, where V_{BE} is an internal voltage. This means that the current into or out of the loop filter is approximately 1.7 V_{BE} / R_{RATE}, or in this example approximately 1.4 mA. Note that although it would seem the overall gain is dependent on V_{BE}, this is not the case. The VCO gain is altered internally by an amount inversely proportional to V_{BE}, as detailed in the section on the Loop Filter. This means that as V_{BE} varies with temperature or device spread, the gain will remain constant for particular fixed values of R_{RATE} and R_{BOOST}. This alleviates the need for potentiometers to select values for each device. The tolerance required for these two resistors will depend on the total loop gain tolerance allowed, but 5% would be typical. Also V_{CC} by-pass capacitors are required for these two resistors. A value of .01 μF is suitable for each.



TL/F/5182-9

FIGURE 6. Pulse Gate Controls



TL/F/5182-10

FIGURE 7. I_{RATE} Set and I_{BOOST} Set

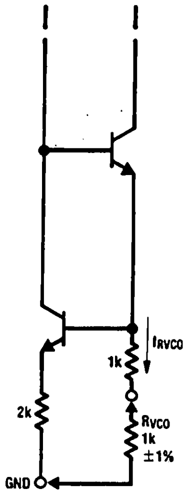
Circuit Operation (Continued)

VCO

The value of R_{VCO} is fixed at 1 kΩ ±1% in the External Component Limits table. *Figure 8* shows how R_{VCO} is connected to the internal components of the chip. This value was fixed at 1 kΩ to set the VCO operating current such that optimum performance of the VCO is obtained for production device spreads. This means fixed value components will be adequate to set the VCO center frequency for production runs. The value of C_{VCO} can therefore be determined from the VCO frequency f_{VCO}, using the equation: $C_{VCO} = [1/(R_{VCO})(f_{VCO})] - 5$ pF where f_{VCO} is twice the input data rate. As an example, for a 5Mbit/sec data rate, $f_{VCO} = 10$ Mhz, requiring that $C_{VCO} = 95$ pF. Note that the additional parasitic capacitance of the PC board will require that the calculated value of C_{VCO} be adjusted further **downward** to achieve nominal frequency centering. The amount of tolerance a particular design can afford on the center frequency will determine the capacitor tolerance. The capacitor is connected to internal circuitry of the chip as shown in *Figure 9.*

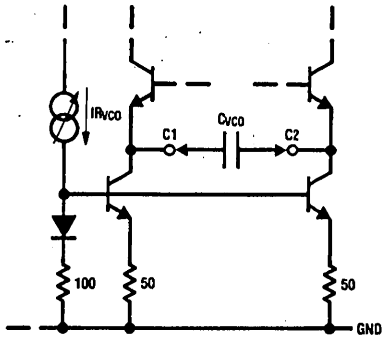
As the data rate increases and C_{VCO} gets smaller, the effects of unwanted parasitic capacitances influence the frequency. As a guide the graph of *Figure 10* shows approximately the value of C_{VCO} for a given data rate.

The center frequency may be checked by averaging the VCO frequencies obtained when the charge pump output (pin 4) is held at 0 volts and then at 3 volts.



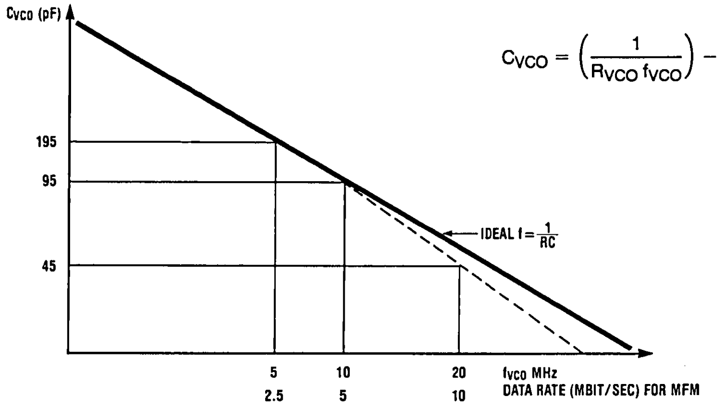
TL/F/5182-11

FIGURE 8. VCO Current Setting Resistor



TL/F/5182-12

FIGURE 9. VCO Capacitor



TL/F/5182-13

FIGURE 10. VCO Capacitor Value for Disk Data Rates

Circuit Operation (Continued)

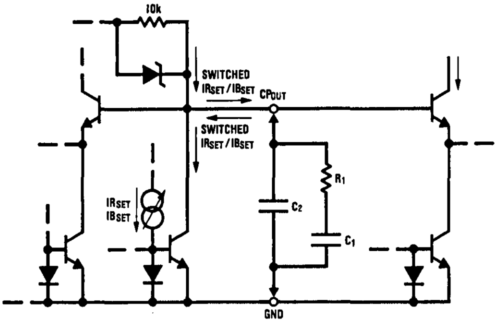
Loop Filter

The input current into the Buffer Amplifier is offset by a matched current out of the Charge Pump, and even so is much less than the switching current in or out of the Charge Pump. It can therefore be assumed that all the Charge Pump switching current goes into the Loop Filter components R_1 and C_1 and C_2. The tolerance of these components should be the same as R_{RATE} and R_{BOOST}, and will determine the overall loop gain variation. The three components connected to the Charge Pump output are shown in *Figure 11*. Note the return current goes to analog GND, which should be electrically very close to the GND pin itself.

The value of capacitor C_1 determines loop bandwidth the larger the value the longer the loop takes to respond to an input change. If C_1 is too small, the loop will track any jitter on the ENCODED DATA input and the VCO output will follow this jitter, which is undesirable. The value of C_1 should therefore be large enough so that the PLL is fairly immune to phase jitter but not large enough that the loop won't respond to longer term data rate changes that occur on the disk drive.

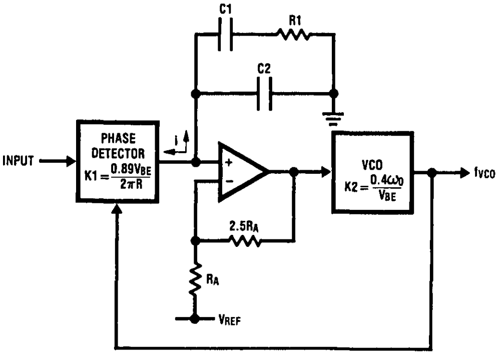
The damping resistor R_1 is required to regulate the second-order behavior (overshoot) of the closed-loop system. A value of R_1 that would give a phase margin of around 45 degrees would be a reasonable starting point.

The main function of the capacitor C_2 is to integrate the effects of the VCO frequency on the VCO input voltage. Typically its value will be less than one tenth of C_1.



TL/F/5182-14

FIGURE 11. Charge Pump Out



TL/F/5182-15

FIGURE 12. Loop Response Components

Figure 12 shows the relevant phase-locked-loop blocks that determine system response, namely the Phase Detector, Filter/Buffer Amplifier, and VCO. The Phase Detector (Phase Comparator and Charge Pump) produces an aggregate output current i which is proportional to the phase difference between the input signal and the VCO signal. The constant

$$(K_1) \text{ is } \frac{.89\ V_{BE}}{2\pi R} \text{ amps per radian.}$$

R is either R_{RATE} or $R_{RATE} \parallel R_{BOOST}$. This aggregate current feeds into or out of the filter impedance (Z), producing a voltage to the VCO that regulates the VCO frequency. The VCO gain constant is $0.4\ \omega_{VCO}/V_{BE}$ radians per second per volt. Under steady state conditions, i will be zero and there will be no phase difference between the input signal and the VCO. Any change of input signal will produce a change in VCO frequency that is determined by the loop gain equation. This equation is determined from the gain constants K_1, A and K_2 and the filter v/i response.

The impedance Z of the filter is:

$$\frac{1}{sC_2} \parallel \left(\frac{1}{sC_1} + R_1\right) = \frac{1 + sC_1R_1}{sC_1(1 + \frac{C_2}{C_1} + sC_2R_1)}$$

If $C_2 \ll C_1$ then the impedance Z approximates to:

$$\frac{1 + sC_1R_1}{sC_1\ (1 + sC_2R_1)}$$

The overall loop gain is then $G(s) = \frac{K_1AK_2}{s} \times \frac{1 + sC_1R_1}{sC_1(1 + sC_2R_1)}$

Let $G_{(K)} = K_1\,A\,K_2$

$$F(s) = \frac{1 + SC_1R_1}{SC_1\ (1 + SC_2R_1)}$$

The Overall Closed Loop Gain is:

$$\frac{\phi_{OUT}}{\phi_{IN}} = \frac{G_{(K)}\ F(s)}{s + G_{(K)}\ F(s)}$$

Substituting, We Get

$$\frac{\phi_{OUT}}{\phi_{IN}} = \frac{G_{(K)}\ (SC_1R_1 + 1)}{S^3\,R_1\,C_1\,C_2 + S^2\,C_1 + GK\,(SC_1R_1 + 1)}$$

$$= \frac{(G_{(K)}/C_1)(SR_1C_1 + 1)}{S^3R_1C_2 + S^2 + SG_{(K)}R_1 + G_{(K)}/C_1}$$

If $C_2 << C_1$, we can ignore the 3rd Order Component introduced by C_2 then:

$$\frac{\phi_{OUT}}{\phi_{IN}} = \frac{(G_{(K)}/C_1)(SR_1C_1 + 1)}{S^2 + SG_{(K)}R_1 + G_{(K)}/C_1}$$

This is a second Order Loop and can be solved as follows:

$$S^2 + SG_{(K)}R_1 + G_{(K)}/C_1 = S^2 + 2\zeta\,\omega_nS + \omega_n^2$$

$$\therefore\ C_1 = \frac{G_{(K)}}{\omega_n^2}$$

$$R_1 = \frac{2\zeta\omega_n}{G_{(K)}}$$

$\zeta = 1.0$ For Critically Damped Response

Circuit Operation (Continued)

From the above equations:

$$\omega = \sqrt{\frac{G(K)}{C1}}$$

$$G(K) = K1 \times A \times K2 =$$

$$\frac{0.89V_{BE}}{2\pi R} \times \frac{0.4\omega V_{CO}}{V_{BE}} \times 3.5$$

MFM encoded data has a two to one frequency range within the data field. The expression

$$K = \frac{0.89V_{BE}}{2\pi R}$$

is valid when the MFM data pattern is at its maximum frequency. In order to make this equation more general, it may be written as follows:

$$K = \frac{1.78V_{BE}}{2\pi RN}$$

where N is defined as the V_{CO} frequency divided by the encoded data frequency, or, N is equal to

$$\frac{F_{VCO}}{F_{DATA}}$$

(N = 2 for maximum data rate and N = 4 for minimum data rate). Now G(K) can be written as follows:

$$G(K) = \frac{1.78V_{BE}}{2\pi RN} \times \frac{0.4\omega_{VCO}}{V_{BE}} \times 3.5 = \frac{2.5F_{VCO}}{RN}$$

$$\omega n = \sqrt{\frac{2.5F_{VCO}}{C1RN}}$$

$R = R_{RATE}$ in the low track rate

$R = R_{RATE} // R_{BOOST}$ in the high track rate

From the above equations:

$$\omega n = \frac{R1 \times G(K)}{2\zeta}$$

$$G(K) = C1(\omega n^2)$$

$$\zeta = \text{(damping factor)} = \frac{R1 \times \omega n \times C1}{2}$$

The damping factor should approach, but not fall below, 0.5 when ωn is minimum. Response to bit shift is minimized when the damping factor is small; however, if the damping factor drops much below 0.5, the system tends to be oscillatory (underdamped).

Additionally, loop performance is poor (excessive phase acquisition times) if the damping factor becomes significantly greater than 1.0. Any increase in loop bandwidth (due to R decreasing in the high track rate) produces a proportional increase in the damping factor, and this should be limited to the point where the maximum damping factor does not significantly exceed 1.0. With the damping factor range established, loop design can now proceed.

A 1550 Krads/sec bandwidth in the non read mode results in a wide capture range; a 4% frequency difference between the crystal and recorded data would not cause an acquisition problem. (This bandwidth may seem excessive to some and if the user does not think it is necessary, he may design his filter with a more desirable bandwidth. For an in-depth discussion of this point, it is suggested that the reader refer to the Disk Interface Design Guide and User's Manual, chapter 1, sections 1.3 thorugh 1.7.)

This design example assumes that the SET PLL LOCK pin is tied to the PLL LOCK DETECTED pin. This results in the track rate being switched from high to low after two bytes of preamble are detected. As an alternative, the SET PLL LOCK pin may be tied to an inverted READ GATE signal, resulting in the track rate switching immediately to low when READ GATE is asserted. This is discussed further in the above mentioned reference material.

In the non read mode or high track rate.

$$\omega n = \sqrt{\frac{2.5F_{VCO}}{C1RN}}$$

Choose $R = R_{RATE} // R_{BOOST} = 410$

In the non-read mode N = 2

$$1550 \text{ Krads/sec} = \sqrt{\frac{2.5 \times 10^7}{C1 \times 410 \times 2}}$$

$$C1 = 0.012\,\mu F$$

In the preamble, after two bytes are detected and PLL LOCK DETECT goes low

$$\omega n = \sqrt{\frac{2.5F_{VCO}}{C1RN}}$$

$R = R_{RATE} = 820$

$N = 2$

ωn = 1127 Krads/sec

Again, in the data field, the minimum data frequency is equal to one half the preamble frequency. This means that N = 4 in the bandwidth equation. This reduces the bandwidth to:

$$\omega n(\min) = \frac{1}{\sqrt{2}} \times 1107 \text{ Krads/sec} = 797 \text{ Krads/sec}$$

Before, we stated that the minimum value of ζ should be 0.5; knowing ωn(min) we can now solve for R1

$$\zeta = \omega n \times R1 \times C1 \times 1/2$$

choose ζ(min) = 0.55

$$R1 = \frac{2\zeta}{\omega n \times C1}$$

R1 = 115 (choose 120)

The maximum damping value occurs in the high track rate;

$$\zeta(\max) = \omega n(\max) \times R1 \times \frac{C1}{2}$$

$$= 1550 \text{ Krads/sec} \times 120 \times \frac{0.012\,\mu F}{2}$$

ζ(max) = 1.12

The maximum damping value in the read mode is as follows:

$$\zeta(\text{max-read}) = 1127 \text{ Krads/sec} \times 120 \times \frac{0.012\,\mu F}{2}$$

ζ(max-read) = 0.81

The continuous behavior (non-quantized) approximation used to predict loop performance assumes that the phase detector output is constantly proportional to the input phase difference. In reality, the phase detector output is a pulse applied for a period of time equal to the phase difference. The function of C2 is to smooth the phase detector output (VCO control voltage) over each cycle. C2 also adds a second pole to the filter transfer function. This pole should be far enough outside the loop bandwidth (at least one order of magnitude) that its phase and amplitude contribution is negligible in the loop bandwidth. If:

$$C2 = \frac{C1}{50} = 240 \text{ pF (choose 300 pF)}$$

The final loop component is R_{BOOST}. Since R_{RATE} and the parallel combination of R_{RATE} and R_{BOOST} are known, we can calculate R_{BOOST}.

Circuit Operation (Continued)

Data Rate	Non-Read		Read		Charge Pump[1]		Loop Filter[2]		
(NRZ)	ωn(max)	ζ	ωn(min)	ζ	R_{RATE}	R_{BOOST}	R1	C1	C2
	rads/sec		rads/sec		ohms	ohms	ohms	μF	pF
5 Mbit/sec	1550k	1.12	797k	0.55	820	820	120	0.012	300
5 Mbit/sec	903k	0.99	435k	0.48	1500	1300	100	0.022	390
5 Mbit/sec	659k	1.55	248k	0.52	1500	590	69	0.068	1500

[1] Component tolerances are system dependent; they depend on how much loop gain deviation can be tolerated.

[2] Component tolerances are typically 5% but they depend on the amount of Loop Bandwidth tolerance that can be accepted.

$$R_{BOOST} = \frac{(Rp)\,(R_{RATE})}{(R_{RATE} - Rp)} = 820$$

The above filter values and those for other bandwidths are listed on following page.

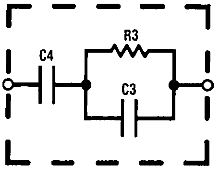
The calculated values are only a guide, the user should then empirically test the loop and determine stability, lock-on time, jitter tolerance, etc.

The desired Bode plot of gain and phase is shown in *Figure 13*, with 20-dB/decade slope at ω_0 for stability at unity gain.

As C_2 is increased in value response to jitter will decrease, but the closer the −40dB/decade slope gets to ω_0 on the Bode plot the more unstable the loop will be. Thus if C_2 is made too large the loop will oscillate.

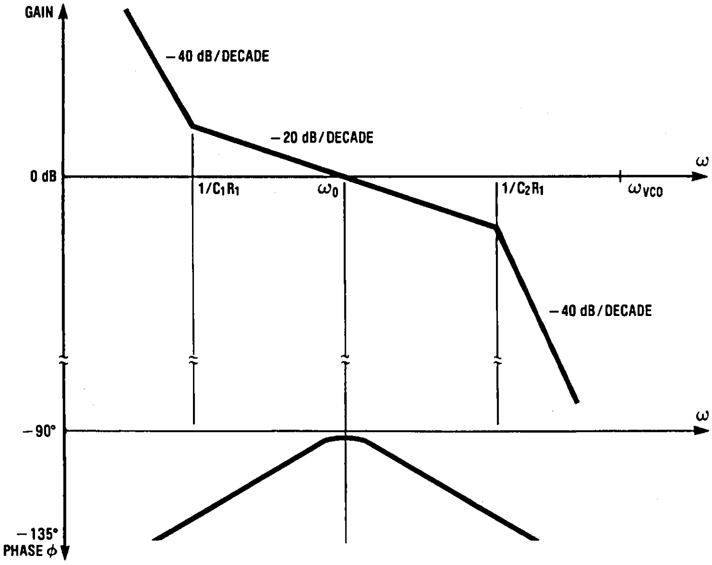
Resistor R_1 determines where the low-frequency end −40 dB/decade slope changes into the −20 dB/decade slope. The wider the −20 dB/decade slope is around unity gain, the more stable the loop becomes. If R_1 is too large it will reduce the impact of C_1, while too small a value will increase instability. The capacitor C_1 strongly effects the response of the loop. Too high a value will slow down the response time, but make the PLL less prone to jitter or frequency shift whereas too low a value will improve response time while tending to increase the PLL's reaction to jitter.

Other filter combinations may be used, other than R_1 in series with C_1, all in parallel with C_2. For example the filter shown in *Figure 14* will also perform similarly, and in fact for some systems it will yield superior performance.



TL/F/5182-17

FIGURE 14. Alternate Loop Filter Configuration



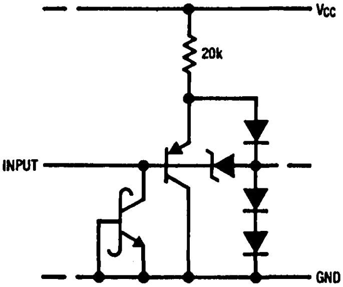
TL/F/5182-16

FIGURE 13. Bode Plot of Loop Response

Circuit Operation (Continued)

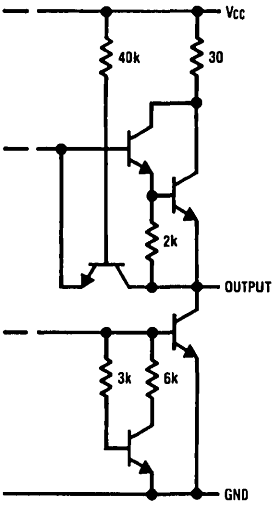
DIGITAL CONNECTIONS TO THE DP8460

Figure 17 shows a connection diagram for the DP8460 in a typical application. All logic inputs and outputs are TTL compatible as shown in *Figure 15* and *16*. The VCO CLOCK output is 74AS compatible and can theoretically drive up to 40 74AS (or 74F) inputs, or 10 74S inputs, or 100 74ALS inputs, or 50 of 74LS inputs. However, it is recommended that the load seen at this output be **minimized** in order to reduce any transient noise local to the PLL. All other outputs are 74ALS compatible and so will drive up to 16 74AS inputs, or 4 74S inputs, or 40 74ALS inputs or 20 74LS inputs. All inputs are 74ALS compatible and therefore can be driven easily from any 74 series devices. The raw MFM from the pulse detector in the drive is connected to the ENCODED DATA input.



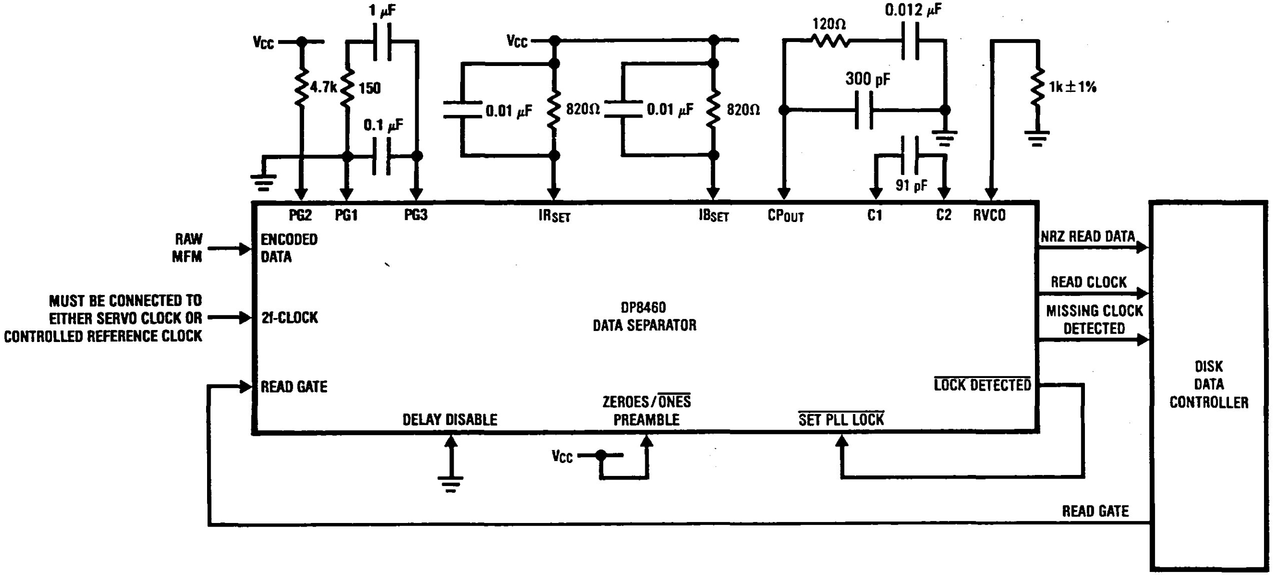
TL/F/5182-18

FIGURE 15. Logic Inputs



TL/F/5182-19

FIGURE 16. Logic Outputs



TL/F/5182–20

1) MFM Data Input, 5Mbit/sec Data Rate
2) 32 Bit Delay to Enable
3) All Zeroes (NRZ) Preamble

FIGURE 17. Typical Connection to DP8460 for:

Circuit Operation (Continued)

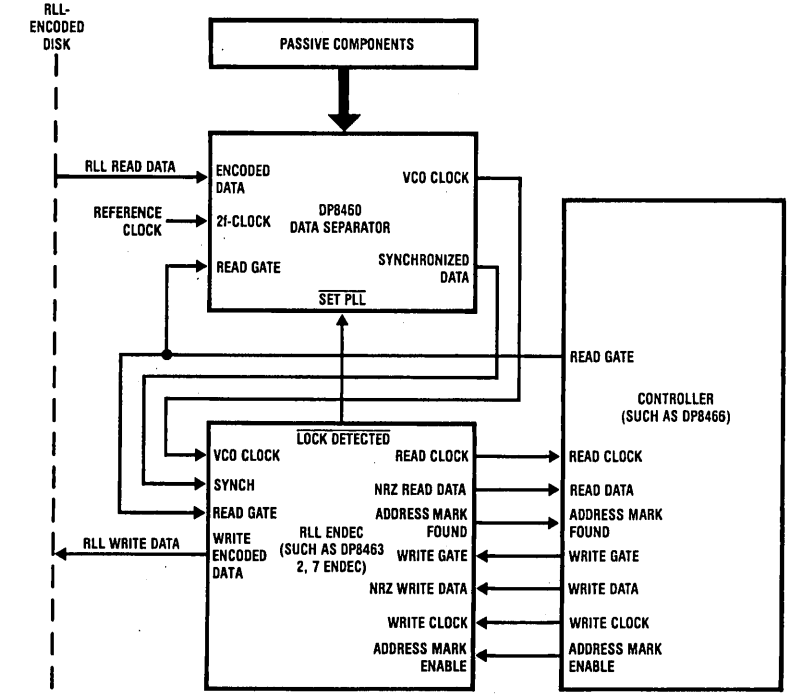
The DELAY DISABLE input determines whether attempting lock-on will begin immediately after READ GATE is set or after 2 bytes. Typically in a hard-sectored drive, READ GATE is set active as the sector pulse appears, meaning a new sector is about to pass under the head. Normally the preamble pattern does not begin immediately, because gap bytes from the preceding sector usually extend just beyond the sector pulse. Allowing 2 bytes to pass after the sector pulse helps ensure that the PLL will begin locking on to preamble, and will not be chasing non-symmetrical gap bits. Attempting to lock-on to a fixed . . . .1010. . . . preamble pattern speeds up lock-on, and after another two bytes the PLL will nominally have locked-on. Thus DELAY DISABLE should be set low for this kind of disk drive.

For soft sectored drives, the controller normally will not wait for the index pulse before it attempts lock-on, so that READ GATE may go active at any time. Chances are the head will not be over a preamble field and therefore there is no need to wait 2 bytes before attempting lock-on. DELAY DISABLE can therefore be set high. If a non-preamble field is passing by as READ GATE goes active, the DP8460 will not indicate lock, and no data decoding will occur nor will MISSING CLOCK DETECTED go active. Normally, if lock-on has not been achieved after a certain time limit, the controller will de-activate READ GATE and then try again.

For MFM encoded disk drives, the $\overline{\text{LOCK DETECTED}}$ output will be connected back to the $\overline{\text{SET PLL LOCK}}$ input. As the PLL achieves lock-on, the DP8460 will automatically switch to the slower tracking rate and decoded data will appear at the NRZ READ DATA output. Also the READ CLOCK output will switch from half the 2f-clock frequency to the disk data rate frequency. If a delay is required before the changeover occurs, a time delay may be inserted between the two pins.

Some drives have an all-ONES data preamble instead of all-ZEROES and the DP8460 must be set to the type being used before it can properly decode data. The ZEROES/$\overline{\text{ONES}}$ PREAMBLE input selects which preamble type the chip is to lock-on to.

If the drive uses a run-length-limited (RLL) code such as 2, 7, instead of MFM, the phase-locked-loop function of the DP8460 may still be used. *Figure 18* shows how the DP8460 may be connected to a RLL ENDEC circuit. The RLL ENDEC performs encoding of NRZ data to RLL encoded data, and RLL encoded data back to NRZ data. The RLL ENDEC can use the SYNCHRONIZED DATA output of the DP8460 along with VCO CLOCK to lock-on to the preamble and then decode data. Once lock-on has been detected, the RLL ENDEC can set the $\overline{\text{SET PLL LOCK}}$ input of the DP8460 low so that the tracking rate can be changed.



TL/F/5182-21

FIGURE 18. DP8460 with Run-Length-Limited (RLL) Codes

Applications of the DP8460 Data Separator

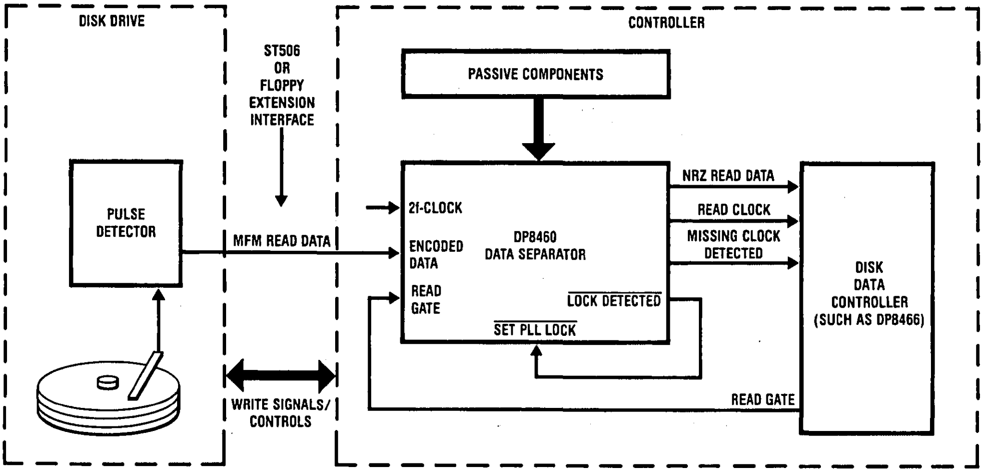
The DP8460 is the first integrated circuit to place on one chip a PLL with features that offer the improved speed and reliability required by the disk industry. Not only does the chip simplify disk system design, but also provides fast lock-on to the incoming preamble. Once locked on, the loop is set into a more stable mode. This inherent loop stability allows for a sizeable amount of jitter on the data stream, such as is encountered in many disk systems. Once in the stable tracking rate, the SYNCHRONIZED DATA output represents the incoming ENCODED DATA and is synchronous with VCO CLOCK. If the disk is MFM encoded, then the chip can decode the synchronized data into NRZ READ DATA and READ CLOCK. These are available as outputs from the chip allowing the NRZ READ DATA to be deserialized using the READ CLOCK.

The DP8460 is capable of operating at up to 15Mbits/sec data rates and so is compatible with a wide assortment of disk drives. Some 8-inch and 14-inch disk drives incorporate the functions of the DP8460, but use many discrete ICs. In these cases, replacing these components with the DP8460 will offer reduced P.C. board area, lower cost, and improved performance while simplifying circuit testing.

Most 5¼-inch and many 8-inch and 14-inch disk drives manufactured at present do not incorporate any of the functions of the DP8460. This is so primarily because the PLL function is difficult to design and implement and requires circuitry which covers a large area of the printed circuit card. This is undesirable both from the drive size aspect and from the cost aspect (the cost includes soldering, testing, and adjusting the components). Consequently, most smaller disk drives output MFM encoded data so that the phase-locked-loop and data separation have to be performed by the controller. The DP8460 will therefore replace these functions in controller designs, as shown in *Figure 19.*

System design criteria may now change because the DP8460 is a one-chip solution, requiring only a few external passive components with fixed values. It operates from a +5V supply, consumes about 0.5W, and is housed in a narrow 24-pin package. The circuitry has been designed so that the external resistors and capacitors need not be adjustable; the user chooses the values according to the disk drive requirements. Once selected, they will be fixed for that particular drive type. These features make it possible to transfer these functions to the disk drive, as shown in *Figure 20.* Apart from a slight increase in board area, the advantages outweigh the disadvantages. First, the components selected are fixed for each type of drive and this facilitates the problem of interchangeability of drives. At present, controllers are adjusted to function with each specific drive; with the DP8460 in the drive, component adjustment will no longer be required. Second there is often a problem of reliability of data transfer. Because the MFM data is clock encoded, this signal is susceptible to noise, bit shift, etc. Soft errors will sometimes occur when the incoming disk data bit position is outside the Pulse Gate window as it is being synchronized to the VCO clock in the phase-locked-loop. Obviously, the nearer the PLL is to the MFM source, the less chance there is that errors will occur. Thus placing the DP8460 in the drive will increase the reliability of data transfer within the system.

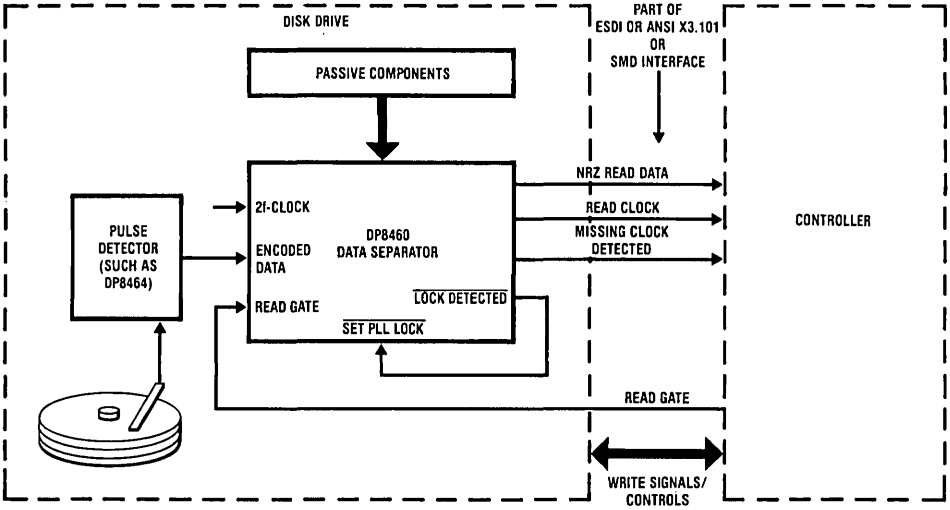
A third advantage is data rate upgrading. Most 5¼-inch drives have 5Mbit/sec data rate because the early drives were made with this data rate. This meant the controllers had to be designed with PLLs which operate at this data rate. It is therefore difficult for drive manufacturers to introduce new drives that are not compatible with existing controllers. Since no new standard data rate has emerged, they must continue to produce drives at this data rate to be compatible with the controllers on the market. With the DP8460 in the drive, and its associated components set for the drive's data rate, it no longer becomes a problem to increase the data rate, assuming the controllers digital circuitry can accommodate the change. This will allow the manufactures to increase the bit density and therefore the capacity of their drives.



TL/F/5182-22

FIGURE 19. DP8460 in the Controller

Applications of the DP8460 Data Separator (Continued)



TL/F/5182-23

FIGURE 20. DP8460 in the Disk Drive

PRECAUTIONS IN BREADBOARDING AND PCB LAYOUT

The DP8460 contains a high performance analog PLL and certain precautions must be taken when breadboarding or designing a PCB layout. The following guidelines should be adhered to when working with the DP8460:

1) Do not wire wrap.
2) Keep component lead lengths short, place components as close to pins as possible.This applies to R1, C1, R2, C_{VCO}, R_{RATE}, R_{BOOST}, C_{RATE}, C_{BOOST}, RPG1, RPG2, and CPG1.
3) Provide a good ground plane and use a liberal amount of supply bypassing. The quieter a PLL's environment, the happier it is.
4) Avoid routing any digital leads within the vicinity of the analog leads and components.

We have used a PC board approach to breadboarding the DP8460 that gives us an excellent ground plane and keeps component lead lengths very short. With this setup we have found very stable and reliable operation. Illustrations of component layout is shown in *Figure 21.* Note that the board layout is a recommendation not a requirement.

ADDITIONAL NOTES

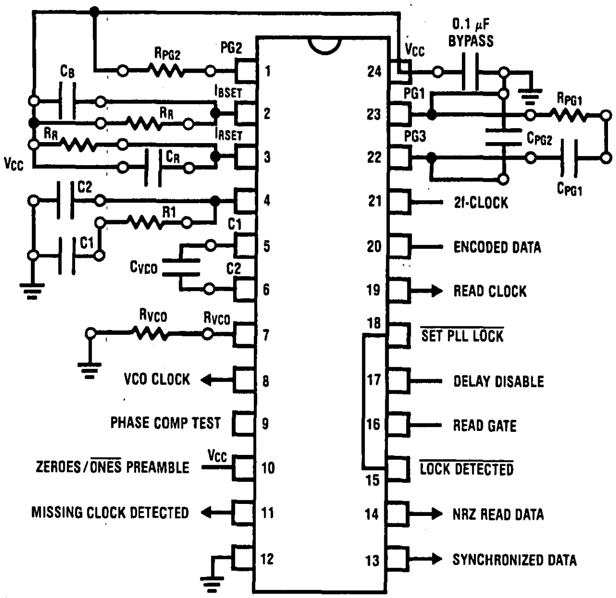
1. PG1 should be grounded to improve noise immunity.
2. 2F clock must be applied at all times; without the 2F clock, the pulse gate circuitry will not operate properly making it impossible to lock onto the incoming data stream.
3. The programming capacitor for the V_{CO} can be calculated as:

   $C_{VCO} = 1/(f_{VCO} {}^{*} R_{VCO}) - 5$ pF

The 5 pF value is due to parasitic and pin to pin capacitance. Typically, an additional 5 pF is also subtracted from the C_{VCO} value to compensate for PC board capacitance.

4. Care must be taken in final PC board layout to minimize pin to pin capacitance, particularly in multi-layer printed circuit boards.
5. Additional design information is available in the application note Precautions for Disk Data Separator Designs, A/N 414. It provides simple solutions to potential application problems.

Connection Diagram



TL/F/5182-24

FIGURE 21. Recommended Component Layout

National
Semiconductor
Corporation

DP8461/65 Data Separator
DP8451/55 Data Synchronizer

General Description

DP8461/65

The DP8461/65 Data Separators are designed for applications in disk drive memory systems, and depending on system requirements, may be located either in the drive or in the controller. They receive digital pulses from a pulse detector circuit (such as the DP8464 Disk Pulse Detector) if situated in the drive, or from an ST506 type interface if situated in the controller. After locking on to the frequency of these input pulses, they separate them into synchronized data and clock signals. While in the non-read mode, both of these circuits employ a phase-frequency comparator to keep the VCO locked to the 2F input (this signal may be derived from a crystal or a servo track). The DP8465 switches to a phase only comparator when the read mode is entered. The DP8461 continues to use a phase-frequency comparator until the preamble detection circuit has detected two bytes of preamble. This feature thus restricts the DP8461 to use with codes employing the 1010 ... preamble. MFM, and certain RLL Codes such as 1,7 and 1,8 employ such a preamble. If a Run Length Limited code is used or if the user wishes to do his own data separation, the synchronized data output is available to allow external circuitry to perform the data decoding function.

All of the digital input and output signals are TTL compatible and only a single +5V supply is required. The chip is housed in a narrow 24-pin dual-in-line package (DIP) and is fabricated using Advanced Schottky bipolar analog and digital circuitry. This high speed I.C. process allows the chip to work with data rates up to 20 Mbit/sec. There are two versions of the chip, each having a different decode window error specification. These two versions (−3, −4) are designed to operate from 2 to 20 Mbit/sec and are tested for their respective window tolerances, as specified in the Electrical Characteristics Table.

The DP8461/65 feature a phase-lock-loop (PLL) consisting of a phase-frequency comparator, pulse gate (to allow for phase-only operation in the read mode), charge pump, buffering amplifier, and voltage-controlled-oscillator (VCO). Pins are provided for the user to select the values of the external filtering components required for the VCO, and two current setting resistors for the charge pump. The DP8461/65 have been designed to be capable of locking onto the incoming preamble data pattern within the first two bytes, using an available high rate of charge pump current. Once lock-on has been achieved, the charge pump can be switched to a lower rate (both rates being determined by the external resistors) to improve bit-jitter immunity for the remainder of the read operation. At this time the READ CLOCK OUTPUT switches, without glitching, from half the 2F-CLOCK frequency to half the VCO CLOCK frequency. After lock-on, with soft sectored disks, the MISSING CLOCK DETECTED output indicates when a missing clock occurs so the controller can align byte boundaries to begin deserialization of the incoming data.

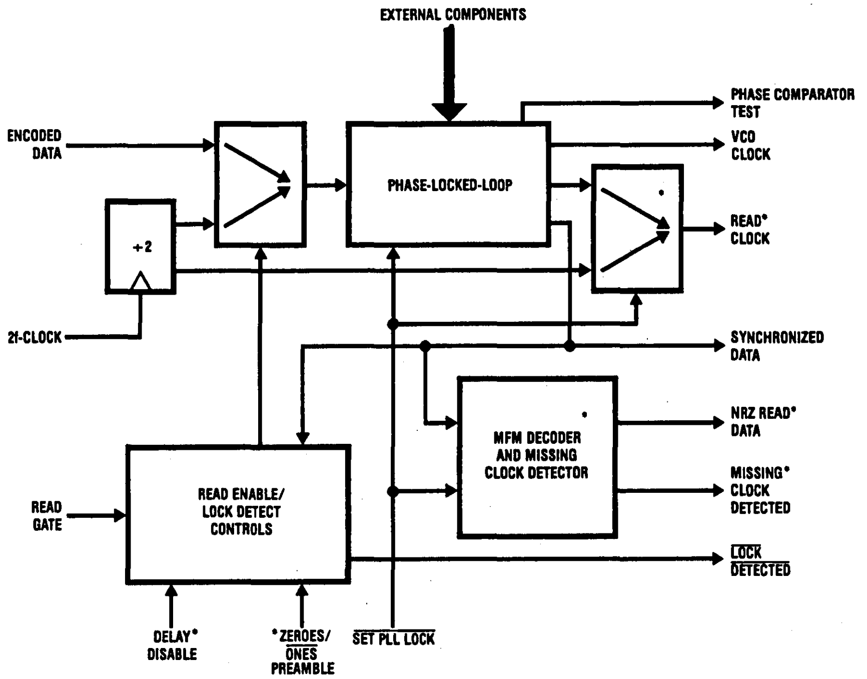
DP8451/55

The DP8451/55 perform the same data synchronization function of the DP8461/65 with no MFM related circuitry. As with the DP8461, the DP8451 continues in the phase-frequency comparison mode until two bytes of preamble are detected. The DP8451/55, which are packaged in 20 pin DIPs or 20-pin PCC's, exclude the READ CLOCK generating circuitry along with the MFM Decoder, Missing Clock Detector, and Read Enable Delay. Users who require only the SYNCHRONIZED DATA OUTPUT and VCO CLOCK OUTPUT can use the DP8451/55 as alternatives to the DP8461/65.

Features

- Operates at data rates up to 20 Mbit/sec
- Phase-Frequency comparison in non-read mode
- Phase-Frequency comparison in preamble—DP8461/51
- Separates MFM data into read clock and serial NRZ data (DP8461/65)
- 4 byte preamble-lock indication capability
- Preamble recognition of MFM encoded "0"s or "1"s
- User-determined PLL loop filter network
- PLL charge pump has two user-determined tracking rates
- External control of track rate switchover
- No glitch on READ CLOCK at switchover (DP8461/65)
- Synchronized data provided as an output (for RLL codes) (all four devices)
- ORed phase comparator outputs for monitoring bit-shift
- Missing clock detected for soft sectored disks
- Less then ½W power consumption
- Standard narrow 24-pin DIP or 28 pin Plastic Chip Carrier Package
- Single +5V supply

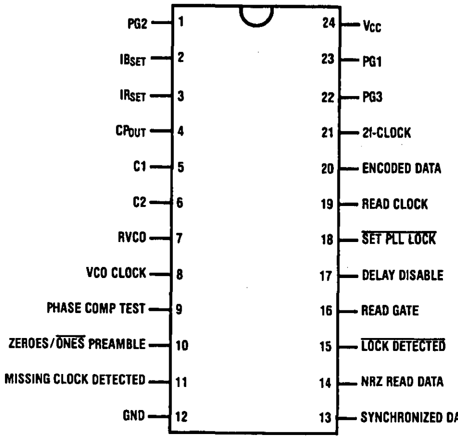
Simplified Block and Connection Diagrams



*Available only on DP8461/65

TL/F/8445-1

DP8461/65
Dual-In-Line Package

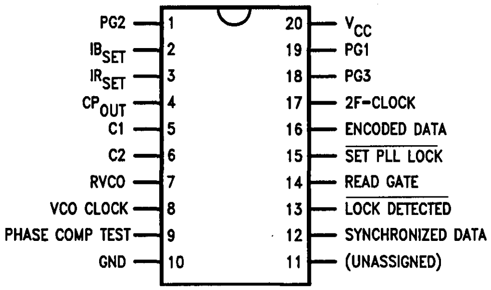


Top View

TL/F/8445-2

Order Number DP8461/65J or N
See NS Package Number J24F or N24C

DP8451/55
Dual-In-Line Package



Top View

TL/F/8445-3

Order Number DP8451/55N
See NS Package Number N20A

Pin Descriptions*

Power Supply

24 V_{CC} +5V ±5%

12 Ground

TTL Level Logic Inputs

16 READ GATE: This is an active high input signal that sets the DP8461/65 Data Separator into the Read Mode.

17 DELAY DISABLE: This input determines the delay from READ GATE going high to the time the DP8461/65 enters the Read Mode. If DELAY DISABLE is set high, this delay is within one cycle of the V_{CO}-CLOCK signal. If DELAY DISABLE is set low, the delay is thirty two-cycles of the VCO CLOCK, as shown in *Figure 1*.

18 $\overline{\text{SET PLL LOCK}}$: This input allows the user to control the on-chip PLL track rate. A high level at this input results in the PLL being in the high track rate. If this input is connected to the $\overline{\text{LOCK DETECTED}}$ output, the PLL will go into the low track rate mode immediately after lock is detected. A low level on this pin is also used to enable the MFM Decoder, the Missing Clock Detector, and to switch the Read Clock Multiplexer from half-2F-CLOCK to half-VCO.

10 ZEROES/$\overline{\text{ONES}}$ PREAMBLE: A high level on this input enables the MFM Decoder circuit to recognize an All Zeros data preamble. A low level results in the recognition of an All Ones data preamble.

20 ENCODED DATA: This input is connected to the output of the head amplifier/pulse-detecting network located in the disk drive. Each positive edge of the ENCODED DATA waveform identifies a change of flux on the disk. In the case of MFM encoded data, the input will be raw MFM.

21 2f-CLOCK: This is a system clock input, which is either a signal generated from the servo track (for systems utilizing servo tracks), or a signal buffered from a crystal. 2f CLOCK MUST ALWAYS BE APPLIED TO THIS INPUT FOR PROPER OPERATION.

TTL Level Logic Outputs

8 VCO CLOCK: This is the output of the on-chip VCO, transmitted from an Advanced Schottky-TTL buffer. It is synchronized to the MFM data output.

15 $\overline{\text{LOCK DETECTED}}$: This output goes active low only after both PLL Lock has occurred and 16 pulses of the preamble pattern have been recognized. It remains low until READ GATE goes inactive.

14 NRZ READ DATA: This is the NRZ (decoded MFM) data output, whose leading edges coincide with the trailing edge of READ CLOCK.

13 SYNCHRONIZED DATA: This output is the same encoded data that is input to the chip, but is synchronous with the negative edge of the VCO CLOCK.

11 MISSING CLOCK DETECTED: When an MFM missing clock is detected, this output will be a single pulse (of width equal to one cycle of READ CLOCK) occurring as shown in *Figure 2*.

19 READ CLOCK: This is half VCO CLOCK frequency when $\overline{\text{SET PLL LOCK}}$ is low; it is half 2f-CLOCK frequency at all other times. A deglitcher is utilized to ensure that no short clock periods occur during either switchover.

9 PHASE COMP TEST: This output is the logical "OR" of the Phase Comparator outputs, and may be used as a bit-shift indicator on for PLL analysis purpose.

Analog Signals

23, 22, PG1, PG3: The external capacitors and resistor of the Pulse Gate filter are connected to these pins. PG1 should be connected directly to the ground pin, pin 12.

1 PG2: This is the Pulse Gate current supply.

3 IRSET: The current into the rate set pin (V_{BE}/R_{RATE}) is used to set the charge pump output current for the low tracking rate.

2 IBSET: The current into the boost set pin (V_{BE}/R_{BOOST}) is used to set the amount by which the charge pump current is increased for the high tracking rate. ($I_{INPUT} = I_{RATE}$ Set + I_{BOOST} Set).

4 CPOUT: CHARGE PUMP OUT/BUFFER AMP IN is available for connection of external filter components for the phase-lock-loop. In addition to being the charge pump output node, this pin is also the noninverting input to the Buffer Amplifier.

7 RVCO: The current at this pin determines the operating currents within the VCO.

5, 6 VCO C1, C2: An external capacitor connected between these pins sets the nominal VCO frequency.

*Pin Number Designations apply only to the DP8461/65. See Connection Diagram for DP8451/55.

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage	7V
TTL Inputs	7V
Output Voltages	7V
Input Current	
(CPOUT, IRSET, IBSET, RVCO)	2 mA
Storage Temperature	−65°C to 150°C

Operating Conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{CC}	Supply Voltage		4.75	5.00	5.25	V
T_A	Ambient Temperature		0	25	70	°C
I_{OH}	High Logic Level Output Current	VCO Clock Others			−2000 −400	μA
I_{OL}	Low Logic Level Output Current	VCO Clock Others			20 8	mA

Operating Conditions (Continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_{DATA}	Input Data Rate		2.0		20	Mbit/sec
t_{WCK}	Width of 2f-CLOCK, High or Low		10			ns
t_{WPD}	Width of ENCODED DATA Pulse, (Note 2)	HIGH	5 ns + 0.10t			ns
		LOW	0.4t			
V_{IH}	High Logic Level Input Voltage		2			V
V_{IL}	Low Logic Level Input Voltage				0.8	V
t_{SETUP} (READ Gate)	Min. Amount of Time Which a Positive Edge of READ Gate Must Precede a Negative Edge of a VCO (Pin 8)		20			ns
t_{HOLD} (READ Gate)	Min. Time Required for a Positive Edge of a READ Gate to be Held after a Negative Edge of a VCO (Pin 8)		10			ns

DC Electrical Characteristics Over Recommended Operating Temperature Range

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IC}	Input Clamp Voltage	V_{CC} = Min., $I_I = -18$ mA			−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min., I_{OH} = Max.	$V_{CC}-2V$	$V_{CC}-1.6V$		V
V_{OL}	Low Level Output Voltage	V_{CC} = Min., I_{OL} = Max.			0.5	V
I_{IH}	High Level Input Current	V_{CC} = Max., V_I = 2.7V			20	µA
I_{IL}	Low Level Input Current	V_{CC} = Max., V_I = 0.4V			−200	µA
I_O	Output Drive Current (Note 1)	V_{CC} = Max., V_O = 2.125V	−12		−110	mA
I_{CC}	Supply Current	V_{CC} = Max.			100	mA
I_{OUT}	Charge Pump Output Current	$500\ \mu A \leq I_{RSET} + I_{BSET} \leq 2000\ \mu A$ $200\ \mu A \leq I_{RSET} + I_{BSET} < 500\ \mu A$	$I_{TYP} - .18\ (I_R + I_B) - 30\ \mu A$ $I_{TYP} - .08\ (I_R + I_B) - 80\ \mu A$	$1.95\ (I_{RSET} + I_{BSET}) - 70\ \mu A$ $1.95\ (I_{RSET} + I_{BSET}) - 70\ \mu A$	$I_{TYP} + .18\ (I_R + I_B) + 30\ \mu A$ $I_{TYP} + .08\ (I_R + I_B) + 80\ \mu A$	µA

Note 1: This value has been chosen to produce a current that closely approximates one-half of the true short-circuit output current. I_{OS}.

Note 2: t is defined as the period of the encoded MFM data, or two times the VCO period.

AC Electrical Characteristics Over Recommended V_{CC} and Operating Temperature Range.

(All Parts unless stated otherwise) ($t_R = t_F = 2.0$ ns, V_{IH} = 3.0V, V_{IL} = 0V)

Symbol	Parameter	Min	Typ	Max	Units
t_{READ}	Positive READ CLOCK transitions from READ GATE set active until PLL Lock sequence begins (DELAY DISABLE low)		16	17	—
t_{READ}	Positive READ CLOCK transitions from READ GATE set active until PLL Lock sequence begins (DELAY DISABLE high)		1	1	—
$t_{DECODE\ NRZ}$	Number of READ CLOCK cycles required to output each decoded MFM data bit (Note 3, 4)	—	2	3	T-clock
$t_{TRANSMIT\ MFM}$	Positive READ CLOCK transitions required to transmit input MFM to output	1	2	3	—

Note: For Further Information Refer to Application Notes AN-414, AN-415, and AN-416.

AC Electrical Characteristics Over Recommended V_{CC} and Operating Temperature Range. (Continued)

(All Parts unless stated otherwise) ($t_R = t_F = 2.0$ ns, $V_{IH} = 3.0V$, $V_{IL} = 0V$)

Symbol	Parameter	Min	Typ	Max	Units
$t_{READ\ ABORT}$	Number of READ CLOCK cycles after READ GATE set low to read operation abort			2	T-clock
t_{WINDOW}	Variance of center of decode window from nominal DP84XX-3 (Note 7) DP84XX-4			6 10	ns
$\phi_{LINEARITY}$	Phase range for charge pump output linearity (Note 2)	$-\pi$		$+\pi$	Radians
K_1	Phase Comparator—Charge Pump gain constant (Note 5) ($N = f_{VCO}/f_{INPUT\ DATA}$, $2 \le N \le 4$ for MFM)		$\frac{1.78V_{BE}}{N2\pi R}$		Amps/rad
$V_{CONTROL}$	Charge pump output voltage swing from nominal		±100		mV
K_{VCO} (=A×K_2)	VCO gain constant (ω_{VCO} = VCO center frequency in rad/s) (Note 1, 6)	$\frac{1.20\omega_C}{V_{BE}}$	$\frac{1.40\omega_C}{V_{BE}}$	$\frac{1.60\omega_C}{V_{BE}}$	rad/sec. V
f_{VCO}	VCO center frequency variation over temperature and V_{CC}	−5		+5	%
$f_{MAX\ VCO}$	VCO maximum frequency		60		MHz
t_{HOLD}	Time READ CLOCK is held low during changeover after lock detection has occurred (Note 3)			1½	T-clock
t_{PHL}	Prop. Delay. VCO Neg. Edge to Synchronized Data Neg. Edge		15	30	ns
t_{PLH}	Prop. Delay. VCO Negative Edge to Synchronized Data Positive Edge		10	25	ns
$t_{2F/RC}$	Delay from 2F positive edge to READ CLOCK positive on negative edge ($\overline{\text{SET PLL LOCK}}$ high)	10		35	ns

Note 1: A sample calculation of frequency variation vs. control voltage: $V_{IN} = \pm 0.1V$;

$$K_{VCO} = \frac{\omega OUT}{V_{IN}} = \frac{0.4\omega_C}{0.2V} = \frac{2.0\omega_C}{V} \frac{(rad/sec)}{(volt)}$$

Note 2: $-\pi$ to $+\pi$ with respect to 2f VCO CLOCK

Note 3: T-clock is defined as the time required for one period of the READ CLOCK to occur.

Note 4: This number remains fixed after PLL Lock occurs.

Note 5: With respect to VCO CLOCK; $I_{PUMP\ OUT} = 1.9\ I_{SET}$

$$I_{SET} = \frac{V_{BE}}{R_{SET}}$$

Note 6: Although specified as the VCO gain constant, this is the gain from the Buffer Amplifier input to the VCO output.

Note 7: This specification is guaranteed only for the conditions under which the parts were tested. However, significant variation from the formula is not expected for other data rates and filters. The filter values below were chosen for operation in an automatic test system (static window) environment. Different criteria may apply for choosing filter values in a disk system. See Loop Filter section for sample calculations of other filter values.

Static Window Margin Test Loop Filter Component Values

Part Type	Data Rate Tested	C_1	C_2	R_1	R_{RATE}	R_{BOOST}
DP8451/55/61/65-4	5 Mbit/Sec	0.02 µF	150 pF	200Ω	750Ω	1.6 kΩ
DP8451/55/61/65-3	10 Mbit/Sec	.082 µF	1600 pF	27Ω	820Ω	619Ω

External Component Selection (All Parts) (Note 1)

Symbol	Component	Min	Typ	Max	Units
R_{VCO}	VCO Frequency Setting Resistor (Note 2)	990		1010	Ω
C_{VCO}	VCO Frequency Setting Capacitor (Note 3, 4)	20		245	pF
R_{RATE}	Charge Pump I_{RATE} Set Resistor (Note 6)	0.4		4.0	kΩ
R_{BOOST}	Charge Pump (High Rate) I_{BOOST} Resistor (Note 6)	0.5		∞	kΩ
C_R	I_{RATE} Bypass Capacitor (Note 5)	.01			µF
C_B	I_{BOOST} Bypass Capacitor (Note 5)	.01			µF

Note 1: External component values for the Loop Filter and Pulse Gate are shown in tables 1 & 2.

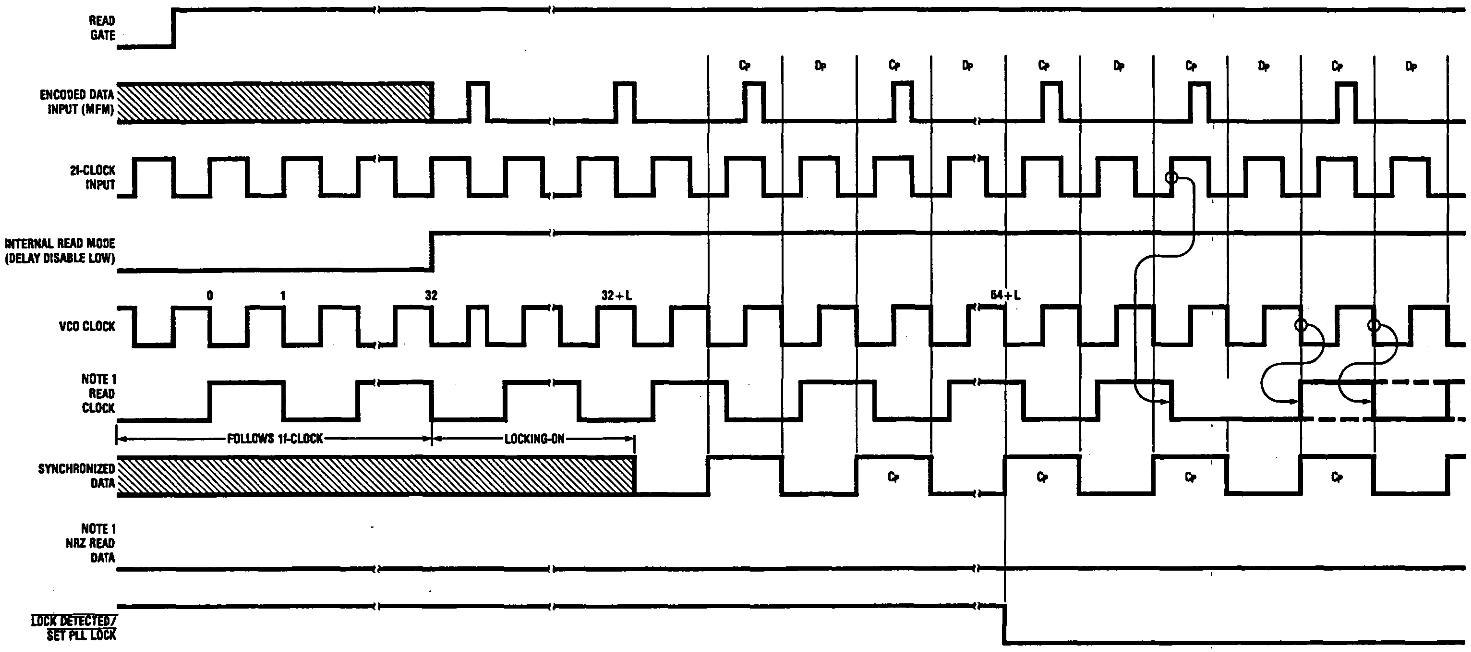
Note 2: A 1% Component Tolerance is Required.

Note 3: These MIN and MAX values correspond to the MAX and MIN data rates respectively.

Note 4: The Component Tolerance is system dependent on how much center frequency deviation can be tolerated.

Note 5: Component Tolerance 15%.

Note 6: The minimum value of the parallel combination of R_{RATE} and R_{BOOST} is 350Ω.



TL/F/8445-4

Note 1: Not included on the DP8451/55.

C_P, D_P = preamble clock and preamble data bits respectively.

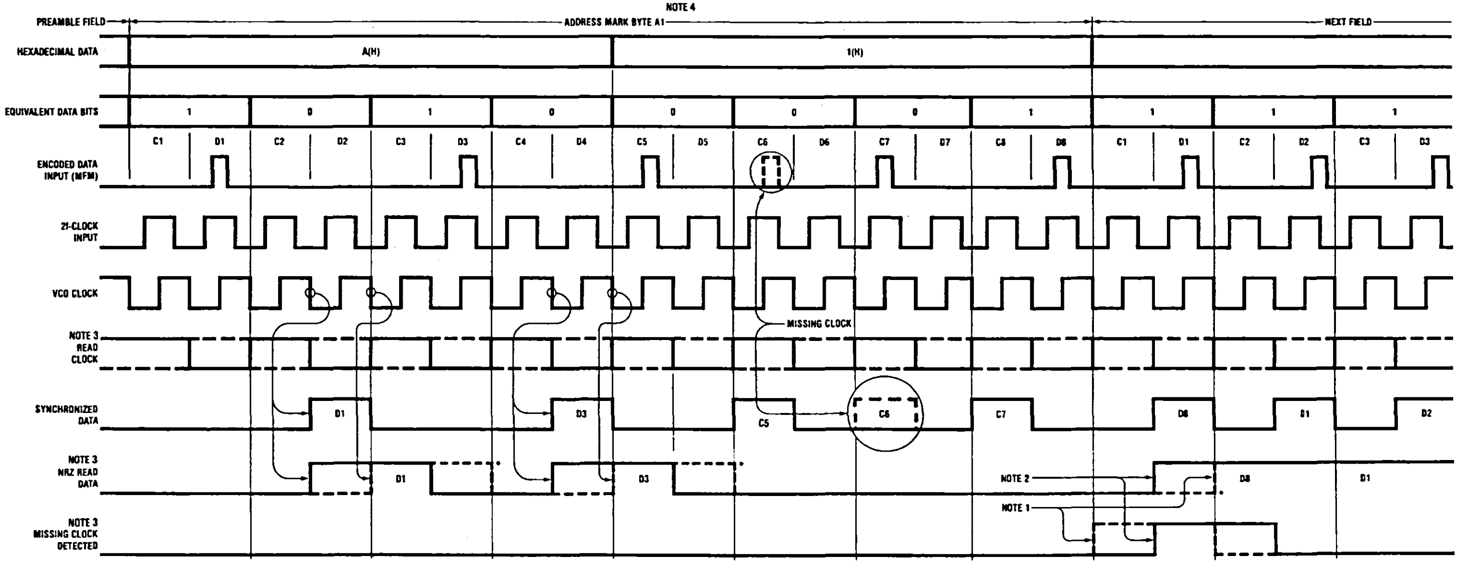
L = Number of 2f-clock cycles required for VCO to lock, determined by external loop filter component values

At 32 + L, VCO has just locked.

At 64 + L, circuit has confirmed lock (has been in lock for 16 MFM clock bits). This sequence shows the MFM all-zeros preamble pattern.

For DP8451/55 delay disable does not exist and part functions as if this input is always high.

FIGURE 1. Lock-on Sequence Waveform Diagram



TL/F/8445-5

* READ CLOCK and NRZ READ DATA may be delayed by one VCO clock period depending on the phase of the internal clock at activation of READ GATE input.

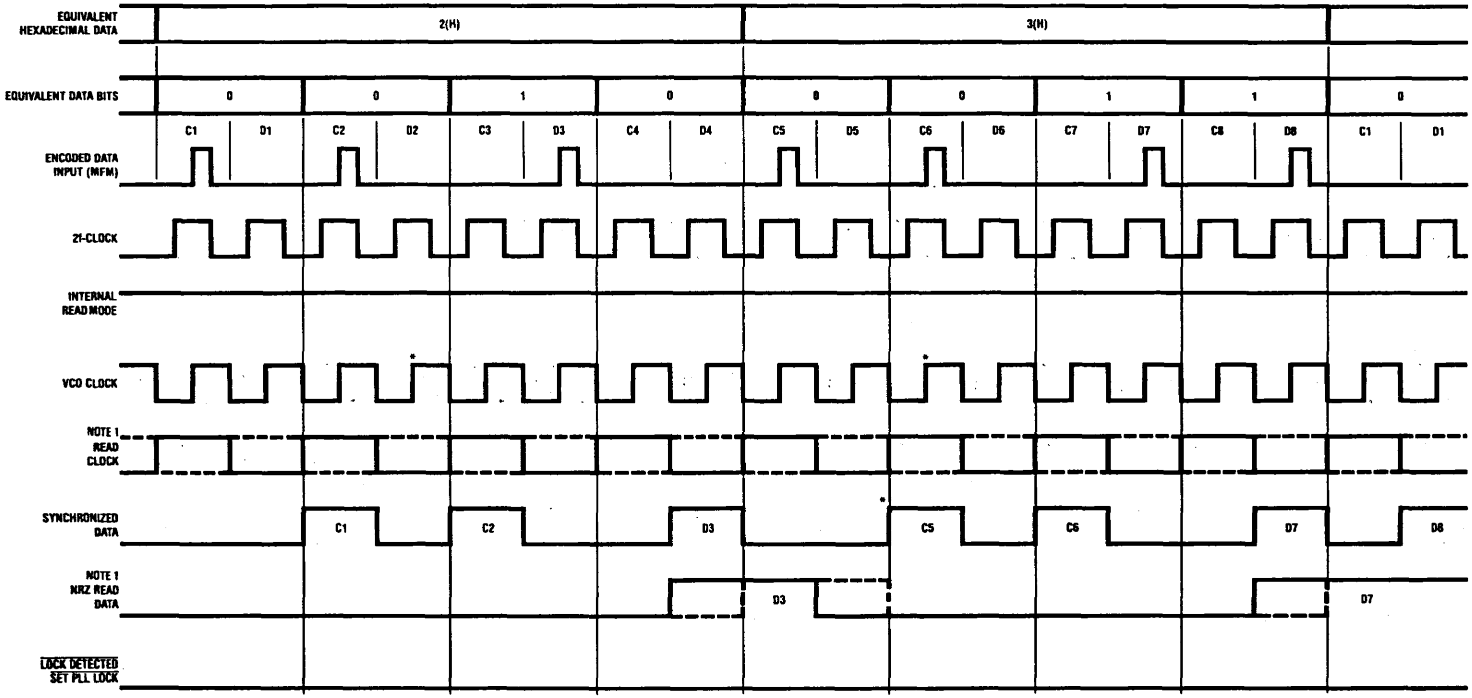
① MISSING CLOCK DETECTED is one READ CLOCK period ahead of the chip issuing D8 on the NRZ READ DATA output when READ CLOCK is delayed by one VCO clock period.

② MISSING CLOCK DETECTED is synchronous with the chip issuing D8 on the NRZ READ DATA Output when READ CLOCK is not delayed.

③ Not included on the DP8451/55.

④ The AI byte is shown only as an example address mark byte. *Any* missing clock bit which is framed by two existing clock bits will produce a missing clock detected pulse.

FIGURE 2. Missing Clock Detection Waveform Diagram

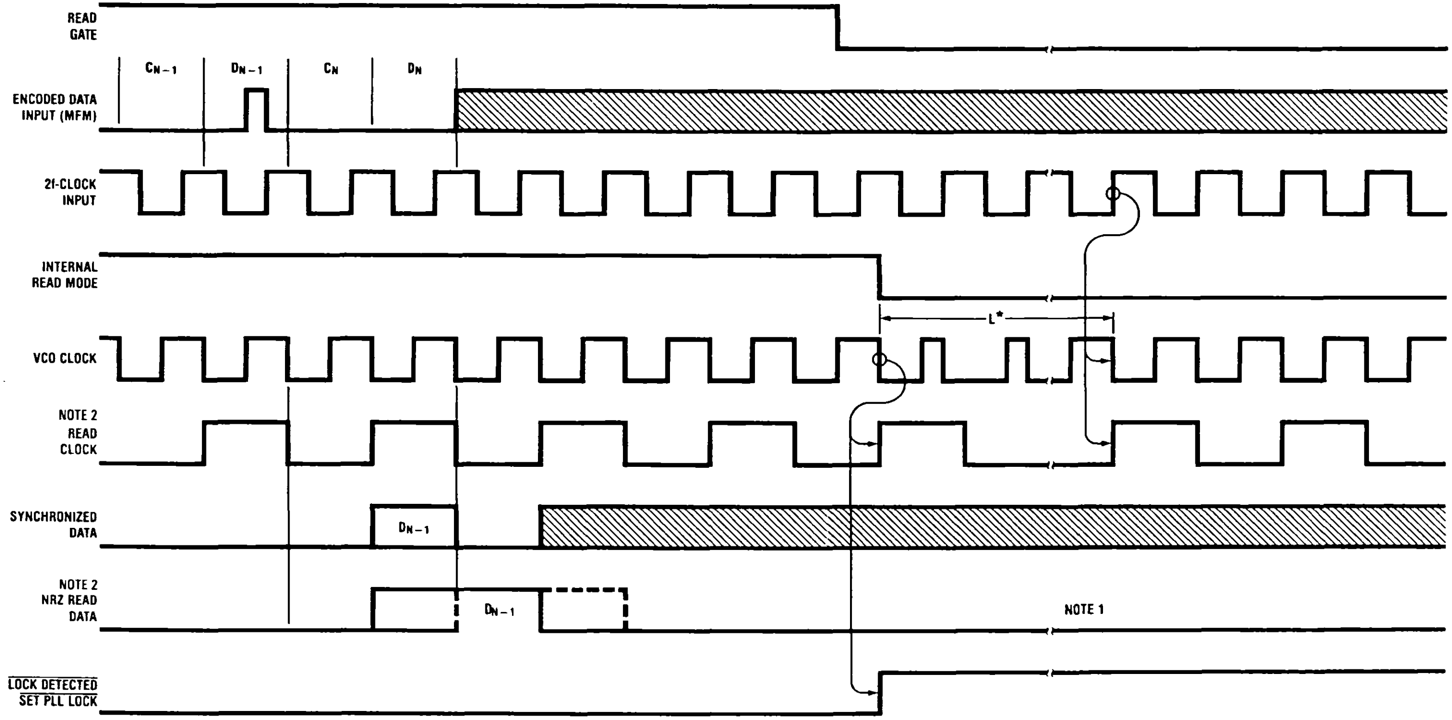


TL/F/8445–6

* **READ CLOCK and NRZ READ DATA may be delayed by one VCO clock period with respect to Synchronized Data depending on the phase of the internal clock at activation of READ GATE input.**

Note 1: Not included on the DP8451/55.

FIGURE 3. Locked-On Waveform Diagram



TL/F/8445-7

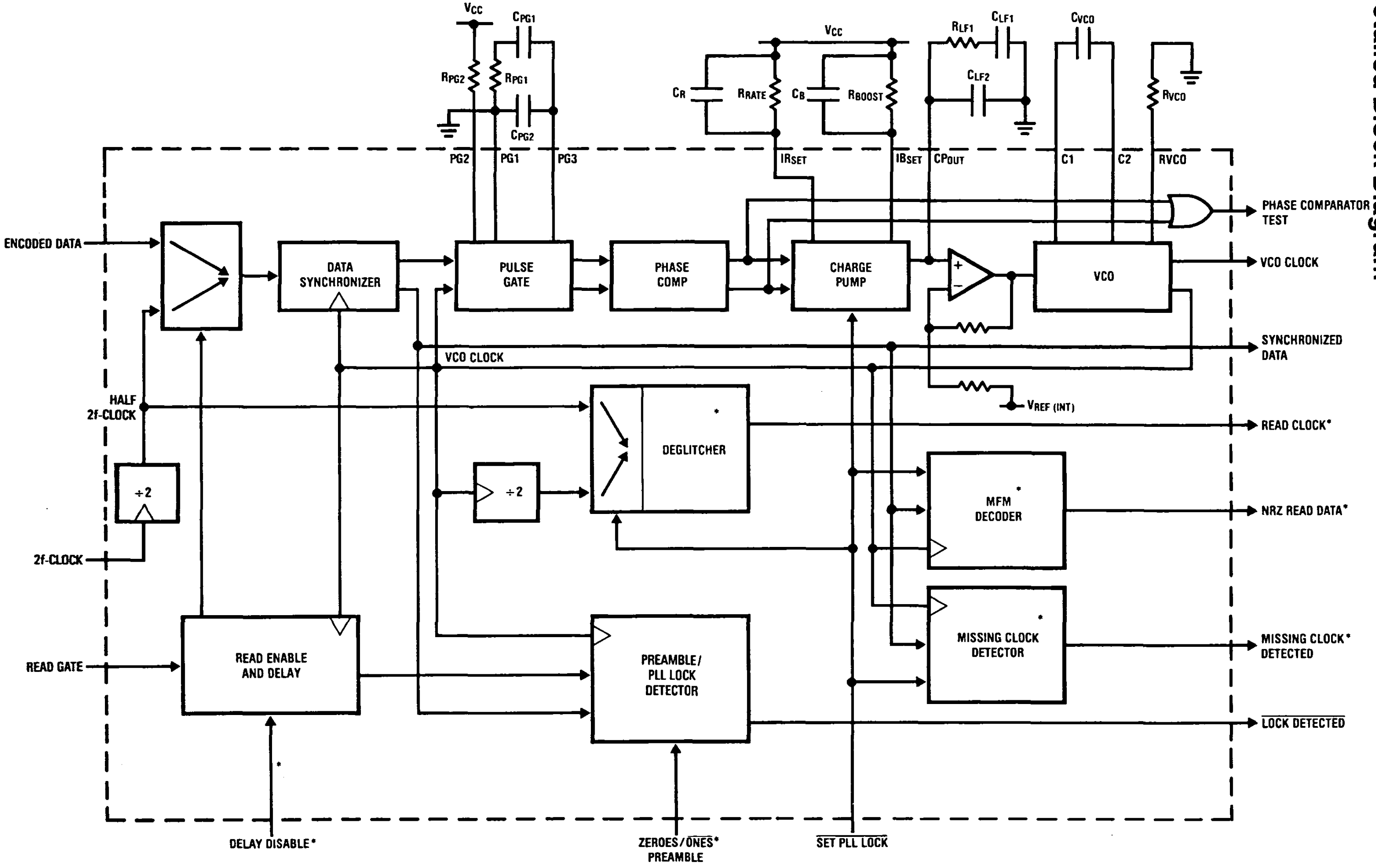
* **L indicates the number of cycles required for the VCO to lock to the 2f-CLOCK.**

Note 1: READ GATE going low will always result in NRZ READ DATA going low regardless of the state of the last bit.

Note 2: Not included on the DP8451/55.

FIGURE 4. Lock-Ending Sequence Waveform Diagram

Detailed Block Diagram



TL/F/8445-8

*Not included on the DP8451/55

Circuit Operation

When the READ GATE input goes high, the DP8461/65 will enter the read mode after a period determined by the state of the DELAY DISABLE pin. This may be either one or thirty two VCO CLOCK cycles. Once in the read mode the DP8465 switches from using a phase-frequency comparator to a phase-only comparator, i.e. the pulse gate is activitated. At this time, however, the DP8461 continues to use a phase-frequency comparator. Referring to *Figure 1*, as the read mode is entered, the phase-locked-loop reference signal is switched from 2F-CLOCK INPUT to the ENCODED DATA. The PLL, initially in the high-tracking rate mode, then attempts to lock onto the incoming encoded data stream. As soon as two bytes of the selected preamble are detected, (the selection is determined by the ZEROES/$\overline{\text{ONES}}$ PREAMBLE pin) the $\overline{\text{LOCK DETECTED}}$ OUTPUT goes low. At this time the DP8461 switches from using a phase-frequency comparator to using the pulse gate, thus beginning phase only comparisons. In a typical MFM disk drive application, the $\overline{\text{LOCK DETECTED}}$ OUTPUT is directly connected to the $\overline{\text{SET PLL LOCK}}$ INPUT. With this connection, track rate selection, clock output switchover, and data output enabling will occur after two consecutive preamble bytes have been detected by the chip. Typically it takes less than one byte time for the VCO to lock to the data sufficiently for preamble detection to begin following the start of the Read operation.

A low level on the $\overline{\text{SET PLL LOCK}}$ causes the PLL Charge Pump to switch from the high to low tracking rate. At the same time, the source of the READ CLOCK signal is switched from half the frequency of the 2F-CLOCK to half the VCO CLOCK. The MFM decoder also becomes enabled and begins to output decoded NRZ data. If a zeroes data preamble is present, the NRZ READ DATA OUTPUT will remain low until the end of the preamble. It will then output whatever NRZ data is present after the preamble field has ended, as shown in *Figures 2* and *3*.

When the READ GATE goes low, signifying the end of a read operation, the DP8461/65 will return to phase-frequency comparator operation. *Figure 4* shows the sequence when READ GATE goes low. The PLL reference signal is switched back to half the 2F-CLOCK and the $\overline{\text{LOCK DETECTED}}$ OUTPUT (and therefore the $\overline{\text{SET PLL LOCK}}$ INPUT) goes high. The PLL then returns to the high track rate, and the output signals return to their initial conditions. The 2F-CLOCK MUST BE APPLIED AT ALL TIMES to the DP8461/65 and DP8451/55 for proper operation.

Since the DP8461/51 employs a phase-frequency comparator until two bytes of the preamble (actually any 16 pulses within a 1010 ... pattern) have been detected, care must be taken to ensure that when using this circuit the READ GATE is applied only within a field containing the 1010 ... pattern. In soft sectored drives the head may be positioned anywhere on the track when initiating a read operation. Therefore, either a controller which only issues READ GATE when a high frequency synchronization field is present, or a simple external circuit between the controller and DP8461/51 to qualify the READ GATE, must be used.

CIRCUIT DESCRIPTION

1. Read Enable and Delay (DP8461/65 only): If the DELAY DISABLE input is connected low, then thirty two VCO CLOCK cycles after READ GATE goes active, the DP8461/65 will go into the read mode. If the DELAY DISABLE input is connected high, the chip will go into the read mode one VCO CLOCK cycle after READ GATE goes active. (The 32 cycle delay is permanently disabled in the DP8451/55).

2. Pulse Gate, including Multiplexer and Data Synchronizer: The Input Multiplexer selects the input to the phase-lock-loop. While the chip is in the bypassed (non-read) mode, the VCO frequency is phase and frequency locked to the 2F-CLOCK INPUT frequency. In the read mode the Input Multiplexer switches to the ENCODED DATA signal and the VCO CLOCK then begins to synchronize with the ENCODED DATA signal. Also, as soon as the read mode is entered, the DP8455/65 cease phase and frequency comparisons by employing the Pulse Gate.

In the DP8461/51 option, switchover from the phase-frequency comparator to the pulse gate (phase-only comparator) occurs after two bytes of the 1010 ... pattern have been detected by the preamble pattern detector.

The Pulse Gate allows a reference pulse from the VCO into the Phase Comparator only after an ENCODED DATA bit has arrived. It utilizes a scheme which delays the incoming data by one-half the period of the 2F-CLOCK. This optimizes the position of the decode window and allows input jitter of approximately half the 2F-CLOCK period. The decode window error can be determined from the specification in the Electrical Characteristics Table.

3. Phase Comparator: The Phase Comparator receives its inputs from the Pulse Gate, and is edge-triggered from these inputs to provide charge-up and charge-down outputs.

4. Charge Pump: The high speed charge pump consists of a switchable constant current source and sink. The charge pump constant current is set by connecting external resistors to V_{CC} from the charge current rate set (IRSET) and current boost set (IBSET) pins. Before lock is indicated, the PLL is in the high tracking rate and the parallel combination of the resistors determines the current. In the low tracking rate after lock-on, only the IRSET resistor determines the charge pump current. The output of the charge pump subsequently feeds into external filter components and the Buffer Amplifier.

5. Buffer Amplifier: The input of the Buffer Amplifier is connected to the charge pump's constant current source/sink output as well as the external Loop Filter components. The Buffer Amplifier is configured as a high input impedance amplifier which allows for the connection of external PLL filter components to the Charge Pump output pin CPOUT. The output of the Buffer Amplifier is internally connected to the VCO control input.

6. VCO: The Voltage-Controlled-Oscillator requires a resistor from the RVCO pin to ground and a capacitor between pins C1 and C2, to set the center frequency. The VCO frequency can be varied from nominal by approximately ±20%, as determined by its control input voltage.

7. PLL Lock-on/Preamble Pattern Detector: To recognize preamble, the preamble pattern from the disk must consist exclusively of either MFM data bit zeroes (encoded into ..10.. MFM clock pulses) when the ZEROES/$\overline{\text{ONES}}$ PREAMBLE pin is set high, or MFM data bit ones (encoded into ..01.. MFM pulses) when set low (DP8461/65 only). The preamble pattern must be long enough to allow the PLL to lock, and subsequently for the Preamble Pattern Detector circuit to detect two complete bytes.

Once the chip is in the read mode, the VCO proceeds to lock on to the incoming data stream. The Preamble Pattern

Circuit Operation (Continued)

Detector then searches for a continuous pattern of 16 consecutive pulses at one-half the VCO frequency to indicate lock has been achieved.

The $\overline{\text{LOCK DETECTED}}$ output then goes low. At this time, in the DP8461/51 option, the PLL switches from using a phase-frequency comparator to employing a pulse gate and thus doing only phase comparisons. Any deviation from the above-mentioned one-zero pattern at any time before PLL Lock is detected will reset the PLL Lock Detector. The lock detection procedure will then start again.

8. MFM Decoder (DP8461/65 only): The MFM Decoder receives synchronized MFM data from the Pulse Gate and converts it to NRZ READ DATA. For run-length-limited codes the MFM Decoder and Missing Clock Detector will not be used.

9. Missing Clock Detector (DP8461/65 only): This block is only required for soft-sectored drives, and is used to detect a missing clock violation of the MFM pattern. The missing clock is inserted when writing to soft-sectored disks to indicate the location of the Address Mark in both the ID and the Data fields of each sector. Once PLL Lock has been indicated, the Missing Clock Detector circuit is enabled. MISSING CLOCK DETECTED will go active if at any time the incoming data pattern contains one suppressed clock bit framed by two adjacent clock bits. (This condition is not constrained to any particular byte pattern such as "A1.") The output signal goes high for one cycle of READ CLOCK.

10. Clock Multiplexer and Deglitcher (DP8461/65 only): When the $\overline{\text{SET PLL}}$ $\overline{\text{LOCK}}$ input changes state this circuit switches the source of the READ CLOCK signal between the half 2f-CLOCK frequency and the half VCO CLOCK frequency. A deglitcher circuit is utilized to ensure that no short clock periods occur during either switchover.

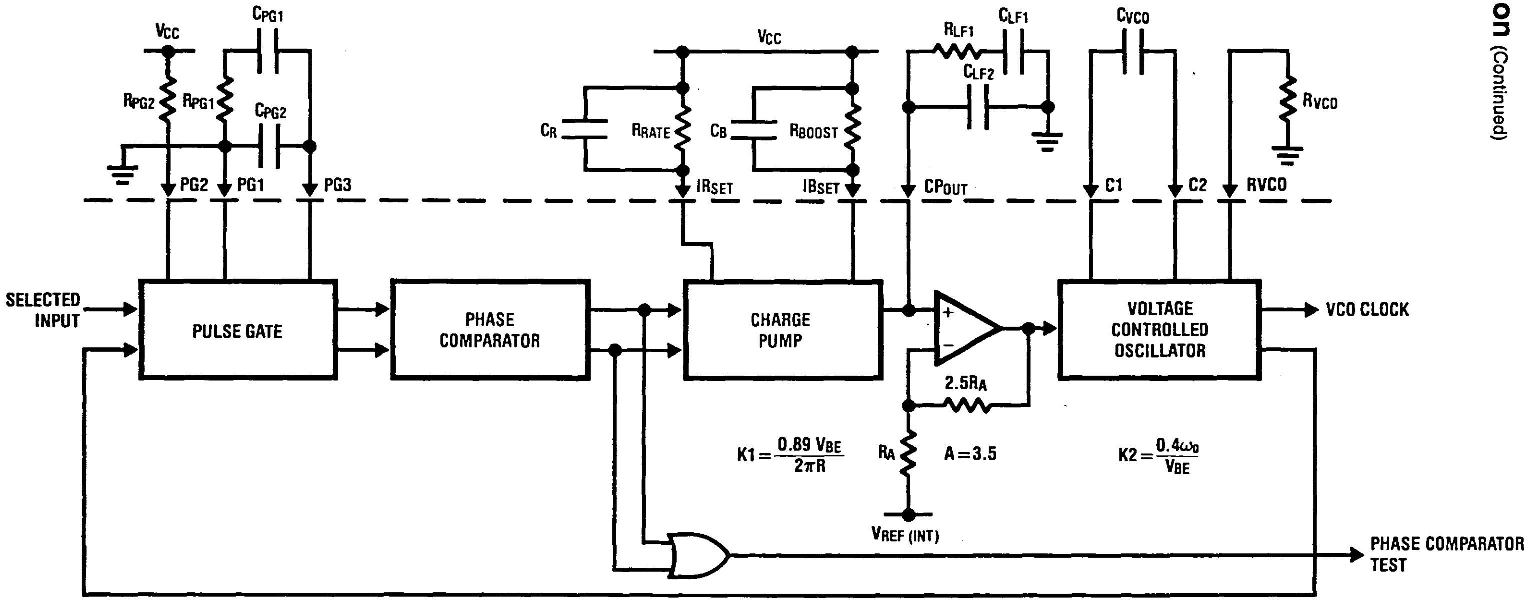
BIT JITTER TOLERANCE

The spec, t-window, as defined in the AC Electrical Characteristics table, describes the distance from the optimum window boundary a single shifted data bit may be placed (following complete PLL lock and stabilization) before it risks being interpreted as residing in the adjacent synchronization window. This is known as the **static window measurement**, which combines all contributing factors of window jitter and displacement within the data separator into a single specification.

The two options of the DP8451/55/61/65, the −4 and −3 offer decreasing static window errors (respectively) so that the parts may be selected for different data rates (up to 20 Mbit/sec). The −4 part will be used in most low data rate applications. As an example, at the 5 Mbit/sec MFM data rate of most 5¼ inch drives, the chip contributes up to ±10 ns of window error, out of the total available window of 100 ns. This allows the disk drive to have a margin of 40 ns of jitter from nominal bit position before an error will occur.

ANALOG CONNECTIONS

External passive components are required for the Pulse Gate, Charge Pump, Loop Filter and VCO as shown in *Figure 5*. The information provided here is for guidelines only. The user should select values according to his own system requirements. Phase-Locked Loops are complex circuits that require detailed knowledge of the specific system. Factors such as loop gain, stability, response to change of signal, lock-on time, etc are all determined by the external components. In many disk systems these factors are critical, and National Semiconductor recommends the designer be knowledgeable of phase-locked-loops, or seek the advice of an expert. Inaccurate design will probably result in excessive disk error rates. The phase-locked-loop in the DP8461/65 has many advantages over all but the most sophisticated discrete designs, and if the component values are selected correctly, it will offer significant performance advantages. This should result in a reduction of disk error rates over equivalent discrete designs. Please refer to the National Semiconductor Application Note AN-414, Precautions for Disk Data Separator Designs, AN-415, Designing with the DP8461, AN-416, Designing with the DP8465, and to the Disk Interface Design Guide and User's Manual, Chapter 1.



TL/F/8445-9

FIGURE 5. Phase-Locked-Loop Section

Circuit Operation (Continued)

Pulse Gate

There are four external components connected to the Pulse Gate as shown in *Figure 6* with the associated internal components. The values of R_{PG1}, R_{PG2}, C_{PG1}, and C_{PG2} are dependent on the data rate. C_{PG1} and C_{PG2} are proportional to the data rate, while R_{PG1} and R_{PG2} are inversely proportional. Table I shows component values for the data rates given. Component values are calculated by selecting R_{PG2} from Table I. Next calculate

$$C_{PG1} = \left(\frac{2.12 \times 10^5}{890 + R_{PG2}}\right)\left(\frac{1}{100 \times R_S}\right)^2$$

$$C_{PG2} = \frac{1}{10} C_{PG1}, \text{ and } R_{PG1} = \left(\frac{890+R_{PG2}}{2.38\times10^5}\right)(100\times R_S).$$

In the above equations R_S is the rotational speed and, for 3600 RPM, $R_S = 60$ Hz. A rotational speed of 3600 RPM was assumed for the calculations in Table I. For data rates not listed, R_{PG2} may be approximated as (30 kΩ/f_{DATA}) − 1.20 kΩ = R_{PG2} where f_{DATA} is the data rate in Megabits/second.

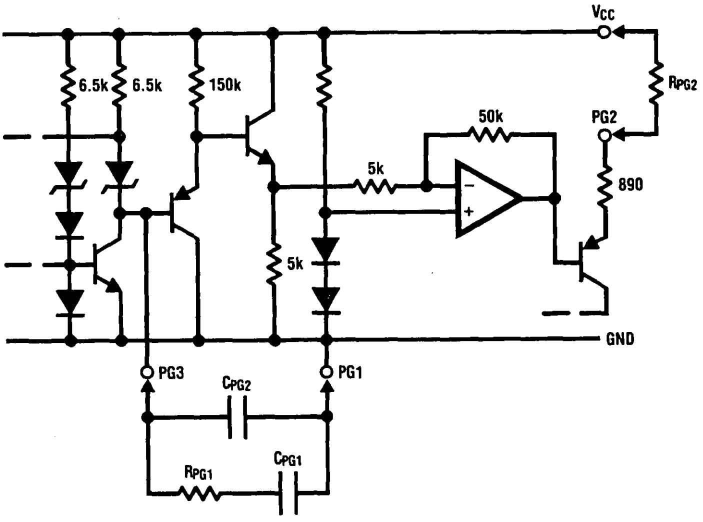
TABLE I. Pulse Gate Component Selection Chart
Components with 10% tolerance will suffice

Data Rate	R_{PG2}	R_{PG1}	C_{PG1}	C_{PG2}
2 Mbit/sec	15 kΩ	430Ω	.39 μF	.039 μF
5 Mbit/sec	4.7 kΩ	150Ω	1 μF	0.1 μF
10 Mbit/sec	1.8 kΩ	68Ω	2.2 μF	.22 μF
15 Mbit/sec	750Ω	39Ω	3.9 μF	.39 μF

Charge Pump

Resistors R_{RATE} and R_{BOOST} determine the charge pump current. The Charge Pump bidirectional output current is approximately 1.9 × the input current (See DC Electrical Characteristics for exact relationship). In the high tracking rate with $\overline{\text{SET PLL LOCK}}$ high, the input current is $I_{BSET} + I_{RSET}$, i.e., the sum of the currents through R_{BOOST} and R_{RATE} from V_{CC}. In the low tracking rate, with $\overline{\text{SET PLL LOCK}}$ low, this input current is I_{RSET} only.

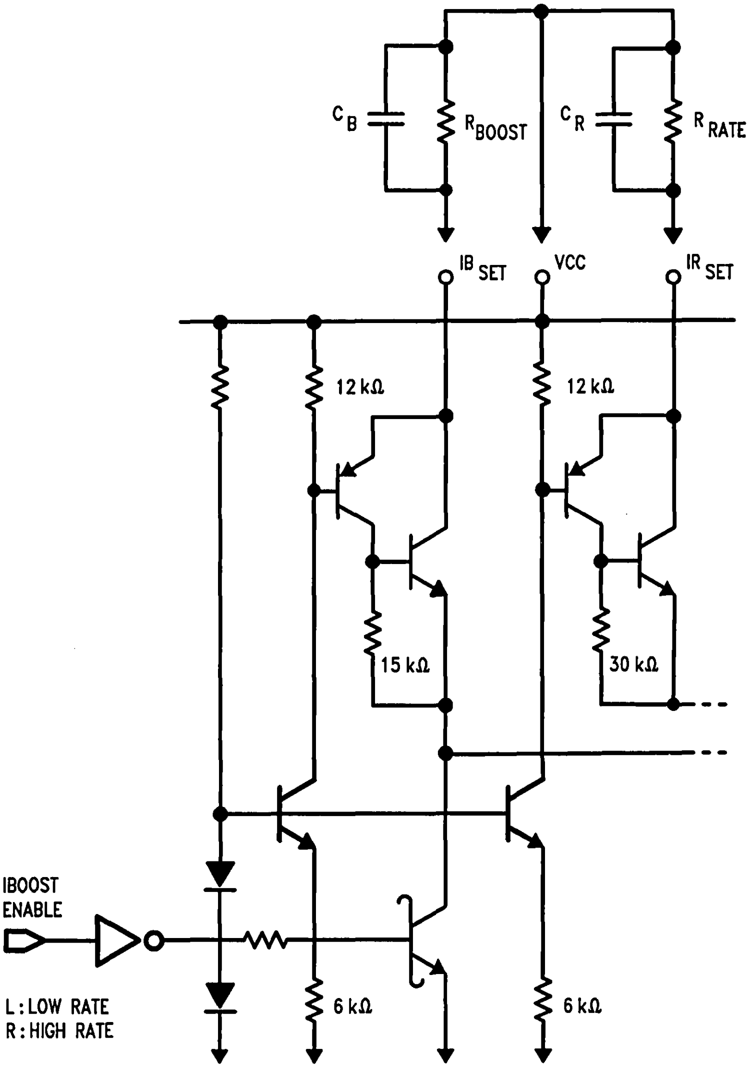
A recommended approach for selecting values for R_{RATE} and R_{BOOST} is described in the design example in the Loop Filter Section. A typical loop gain change of 2:1 for high to low tracking rate would require $R_{BOOST} = R_{RATE}$. Selecting R_{RATE} to be 820Ω would then result in R_{BOOST} equaling 820Ω. Referring to *Figure 7,* the input current is effectively V_{BE}/R_{RATE} in the low tracking rate, where V_{BE} is an internal voltage. This means that the current into or out of the loop filter is approximately (1.95 × V_{BE}/820) − 70 μA = 1.72 mA. Note that although it would seem the overall gain is dependant on V_{BE}, this is not the case. The VCO gain is altered internally by an amount inversely proportional to V_{BE}, as detailed in the section on the Loop Filter. This means that as V_{BE} varies with temperature or device spread, the gain will remain constant for a particular fixed set of values of R_{RATE} and R_{BOOST}. This alleviates the need for potentiometers to select values for each device. The tolerance required for these two resistors will depend on the total loop gain tolerance allowed, but 5% would be typical. Also V_{CC} bypass capacitors are required for these two resistors. A value of .01 μF is suitable for each.



TL/F/8445-10

FIGURE 6. Pulse Gate Controls

Circuit Operation (Continued)



TL/F/8445-11

FIGURE 7. I_{RATE} Set and I_{BOOST} Set

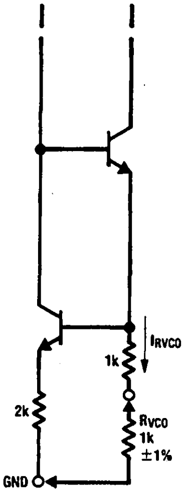
Circuit Operation (Continued)

VCO

The value of R_{VCO} is fixed at 1 kΩ ±1% in the External Component Limits table. *Figure 8* shows how R_{VCO} is connected to the internal components of the chip. This value was fixed at 1 kΩ to set the VCO operating current such that optimum performance of the VCO is obtained for production device spreads. This means fixed value components will be adequate to set the VCO center frequency for production runs. The value of C_{VCO} can therefore be determined from the VCO frequency f_{VCO}, using the equation: $C_{VCO} = [1/(R_{VCO})(f_{VCO})] - 5$ pF where f_{VCO} is twice the input data rate. As an example, for a 5 Mbit/sec data rate, $f_{VCO} = 10$ MHz, requiring that $C_{VCO} = 95$ pF. This does not take into account any inter-lead capacitance on the printed circuit board; the user **must** account for this. The amount of toler ance a particular design can afford on the center frequency will determine the capacitor tolerance. The capacitor is connected to internal circuitry of the chip as shown in *Figure 9*.

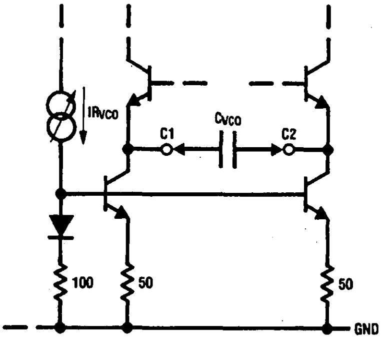
As the data rate increases and C_{VCO} gets smaller, the effects of unwanted internal parasitic capacitances influence the frequency. As a guide the graph of *Figure 10* shows approximately the value of C_{VCO} for a given data rate.

The VCO control input operational range (pin 4) lies at approximately 1.4 volts with a control swing of ±100 millivolts. The VCO itself is constrained to swing a maximum of approximately ±20% of its center frequency, and will remain clamped if the voltage at pin 4 exceeds its operational limit. The VCO center frequency may then be determined by: 1) holding pin 4 at ground potential and measuring the VCO frequency (−20% value); 2) holding pin 4 at approximately 3 volts and measuring the VCO frequency (+20% value); 3) averaging the two measured frequencies for the equivalent center frequency.



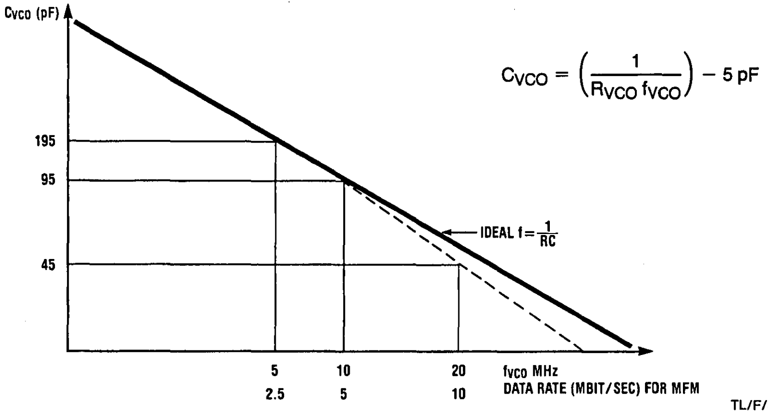
TL/F/8445-12

FIGURE 8. VCO Current Setting Resistor



TL/F/8445-13

FIGURE 9. VCO Capacitor



$$C_{VCO} = \left(\frac{1}{R_{VCO}\,f_{VCO}}\right) - 5\text{ pF}$$

TL/F/8445-14

FIGURE 10. VCO Capacitor Value for Disk Data Rates

Circuit Operation (Continued)

Loop Filter

The input current into the Buffer Amplifier is offset by a matched current out of the Charge Pump, and even so is much less than the switching current in or out of the Charge Pump. It can therefore be assumed that all the Charge Pump switching current goes into the Loop Filter components R_1 and C_1 and C_2. The tolerance of these components should be the same as R_{RATE} and R_{BOOST}, and will determine the overall loop gain variation. The three components connected to the Charge Pump output are shown in *Figure 11*. Note the return current goes to analog GND, which should be electrically very close to the GND pin itself.

The value of capacitor C_1 determines loop bandwidth ... the larger the value the longer the loop takes to respond to an input change. If C_1 is too small, the loop will track any jitter on the ENCODED DATA input and the VCO output will follow this jitter, which is undesirable. The value of C_1 should therefore be large enough so that the PLL is fairly immune to phase jitter but not large enough that the loop won't respond to longer term data rate changes that occur on the disk drive.

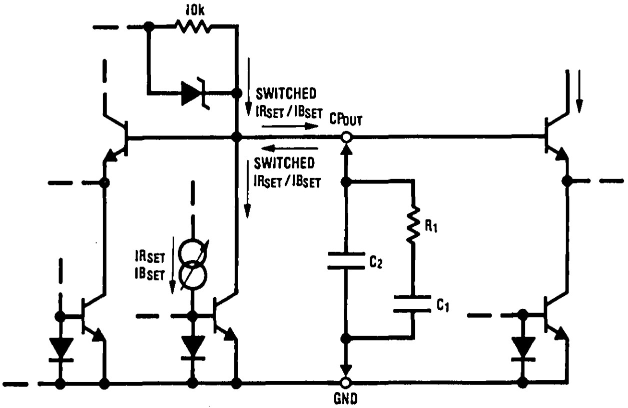
The damping resistor R_1 is required to regulate the second-order behavior of the closed-loop system (overshoot). A value of R_1 that would give a phase margin of around 45 degrees would be a reasonable starting point.

The main function of the capacitor C_2 is to smooth the action of the charge pump at the VCO input. Typically its value will be less than one tenth of C_1. Further effects of C_2 will be discussed later.

Figure 12 shows the relevant phase-locked-loop blocks that determine system response, namely the Phase Detector, Filter/Buffer Amplifier, and VCO. The Phase Detector (Phase Comparator and Charge Pump) produces an aggregate output current i which is proportional to the phase difference between the input signal and the VCO signal. The constant (K_1) is

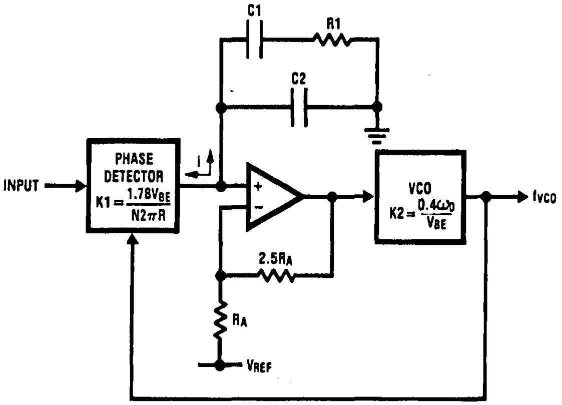
$$\frac{1.78\,V_{BE}}{N2\pi R}\text{ amps per radian, where } N = \frac{f_{VCO}}{f_{DATA}}.$$

R is either R_{RATE} or $R_{RATE} \parallel R_{BOOST}$. The amplified aggregate current feeds into or out of the filter impedance (Z), producing a voltage to the VCO that regulates the VCO frequency. The VCO gain constant is 0.4 ω_{VCO}/V_{BE} radians per second per volt. Under steady state conditions, i will be zero and there will be no phase difference between the input signal and the VCO. Any change of input signal will pro-



TL/F/8445-15

FIGURE 11. Charge Pump Out



TL/F/8445-16

FIGURE 12. Loop Response Components

Circuit Operation (Continued)

duce a change in VCO frequency that is determined by the loop gain equation. This equation is determined from the gain constants K_1, A and K_2 and the filter v/i response.

The impedance Z of the filter is:

$$\frac{1}{sC_2} \,\Big\|\, \left(\frac{1}{sC_1} + R_1\right) = \frac{1 + sC_1R_1}{sC_1\left(1 + \frac{C_2}{C_1} + sC_2R_1\right)}$$

If $C_2 \ll C_1$ then the impedance Z approximates to:

$$\frac{1 + sC_1R_1}{sC_1\,(1 + sC_2R_1)}$$

The overall loop gain is then

$$G(s) = \frac{K_1AK_2}{s} \times \frac{1 + sC_1R_1}{sC_1(1 + sC_2R_1)}$$

Let $G_{(K)} = K_1\,A\,K_2$

$$F(s) = \frac{1 + SC_1R_1}{SC_1\,(1 + SC_2R_1)}$$

The Overall Closed Loop Gain is:

$$\frac{\phi_{OUT}}{\phi_{IN}} = \frac{G_{(K)}\,F(s)}{s + G_{(K)}\,F(s)}$$

Substituting, We Get

$$\frac{\phi_{OUT}}{\phi_{IN}} = \frac{G_{(K)}\,(SC_1R_1 + 1)}{S^3\,R_1\,C_1\,C_2 + S^2\,C_1 + GK\,(SC_1R_1 + 1)}$$

$$= \frac{(G_{(K)}/C_1)\,(SR_1C_1 + 1)}{S^3R_1C_2 + S^2 + SG_{(K)}R_1 + G_{(K)}/C_1}$$

If $C_2 \ll C_1$, we can ignore the 3rd Order Component introduced by C_2 then:

$$\frac{\phi_{OUT}}{\phi_{IN}} = \frac{(G_{(K)}/C_1)\,(SR_1C_1 + 1)}{S^2 + SG_{(K)}R_1 + G_{(K)}/C_1}$$

This is a second Order Loop and can be solved as follows:

$$S^2 + SG_{(K)}R_1 + G_{(K)}/C_1 = S^2 + 2\zeta\,\omega_nS + \omega_n{}^2$$

$$\therefore C_1 = \frac{G_{(K)}}{\omega_n{}^2}$$

$$R_1 = \frac{2\zeta\omega_n}{G_{(K)}}$$

$\zeta = 1.0$ For Critically Damped Response

From the above equations:

$$\omega = \sqrt{\frac{G_{(K)}}{C_1}}$$

$$G_{(K)} = K_1\,A\,K_2 = \frac{0.89 \times V_{BE}}{2\pi R} \times \frac{0.4 \times \omega_{VCO}}{V_{BE}} \times 3.5$$

MFM encoded data has a two to one frequency range within the data field. The expression $K = (0.89 \times V_{BE}\,/\,2\pi R)$ is valid when the MFM data pattern is at its maximum frequency. In order to make this equation more general, it may be written as follows: $K = (1.78 \times V_{BE}\,/\,2\pi RN)$ where N is defined as the V_{CO} frequency divided by the encoded data frequency, or, N is equal to F_{VCO}/F_{DATA} (N = 2 for maximum data rate i.e., MFM = 101010 ... and N = 4 for minimum data rate) i.e., MFM = 100010001 .... Now $G_{(K)}$ can be written as follows:

$$G_{(K)} = \frac{1.78 \times V_{BE}}{2\pi RN} \times \frac{0.4 \times \omega_{VCO}}{V_{BE}} \times 3.5$$

$$= \frac{2.5 \times F_{VCO}}{RN}$$

$$\omega_n = \sqrt{\frac{2.5 \times F_{VCO}}{C_1RN}}$$

$R = R_{RATE}$ in the low track rate

$R = R_{RATE}\ //\ R_{BOOST}$ in the high track rate

From the above equations:

$$\omega_n = \frac{R_1\,G_{(K)}}{2\zeta}$$

$$G_{(K)} = C_1\,\omega_n{}^2$$

$$\zeta = \text{(damping factor)} = \frac{R_1\,\omega_n\,C_1}{2}$$

The damping factor should approach, but not fall below, 0.5 when ω_n is minimum. Response to bit shift is minimized when the damping factor is small; however, if the damping factor drops much below 0.5, the system tends to be oscillatory (underdamped).

Additionally, loop performance is poor (excessive phase acquisition times) if the damping factor becomes significantly greater than 1.0. Any increase in loop bandwidth (due to R decreasing in the high track rate) produces a proportional increase in the damping factor, and this should be limited to the point where the maximum damping factor does not significantly exceed 1.0. With the damping factor range established, loop design can now proceed. The following design example is for a 5 Mbit/sec MFM system.

A 1550 Krads/sec bandwidth in the non read mode results in a wide capture range; a 4% frequency difference between the crystal and recorded data would not cause an acquisition problem. (This bandwidth may seem excessive to some and if the user does not think it is necessary, he may design his filter with a more desirable bandwidth. For an in-depth discussion of this point, it is suggested that the reader refer to the Disk Interface Design Guide and User's Manual, chapter 1, sections 1.3 through 1.7.

This design example assumes that the $\overline{\text{SET PLL LOCK}}$ pin is tied to the $\overline{\text{PLL LOCK DETECTED}}$ pin. This results in the track rate being switched from high to low after two bytes of preamble are detected. As an alternative, the $\overline{\text{SET PLL LOCK}}$ pin may be tied to an inverted READ GATE signal, resulting in the track rate switching immediately to low when READ GATE is asserted. This is discussed further in the above mentioned reference material.

TABLE II.

Data Rate (NRZ)	Non-Read		Read		Charge Pump		Loop Filter		
	$\omega_{n(MAX)}$ rads/sec	ζ	$\omega_{n(MIN)}$ Rads/sec	ζ	R_{RATE} Ω	R_{BOOST} Ω	R_1 Ω	C_1 µF	C_2 pF
5 Mbit/sec	1550K	1.12	797K	0.55	820	820	120	0.012	300
5 Mbit/sec	903K	0.99	435K	0.48	1500	1300	100	0.022	390
5 Mbit/sec	659K	1.55	248K	0.52	1500	590	69	0.068	1500

Circuit Operation (Continued)

In the non read mode or high track rate.

$$\omega_n = \sqrt{\frac{2.5 \times F_{VCO}}{C_1 RN}}$$

Choose R = R_{RATE} // R_{BOOST} = 410

In the non-read mode N = 2

$$1550 \text{ Krads/sec} = \sqrt{\frac{2.5 \times 10^7}{C_1 \times 410 \times 2}}$$

$C_1 = 0.012\ \mu F$

In the preamble, after two bytes are detected and $\overline{\text{PLL LOCK DETECT}}$ goes low

$$\omega_n = \sqrt{\frac{2.5 \times F_{VCO}}{C_1 RN}}$$

$R = R_{RATE} = 820$

$N = 2$

$\omega_n = 1127$ Krads/sec

Again, in the data field, the minimum data frequency is equal to one half the preamble frequency. This means that N = 4 in the bandwidth equation. This reduces the bandwidth to:

$$\omega_{n(min)} = \frac{1}{\sqrt{2}} \times 1107 \text{ Krads/sec} = 797 \text{ Krads/sec}$$

Before, we stated that the minimum value of ζ should be 0.5; knowing $\omega_{n(min)}$ we can now solve for R_1

$$\zeta = \frac{\omega_n R_1 C_1}{2}$$

Choose $\zeta_{(min)} = 0.55$

$$R_1 = \frac{2\zeta}{\omega_n C_1}$$

$R_1 = 115\Omega$ (choose 120Ω)

The maximum damping value occurs in the high track rate;

$\zeta_{(max)} = \omega_{n(max)} R_2 C_1/2$

$= 1550 \text{ Krads/sec} \times 120 \times 0.012\ \mu F/2$

$\zeta_{(max)} = 1.12$

The maximum damping value in the read mode is as follows:

$\zeta_{(max-read)} = 1127 \text{ Krads/sec} \times 120 \times 0.012\ \mu F/2$

$\zeta_{(max-read)} = 0.81$

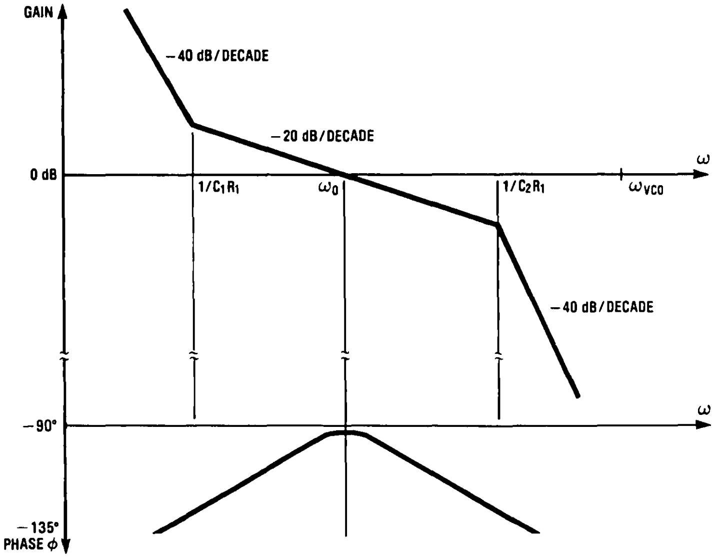
The continuous behavior (non-quantized) approximation used to predict loop performance assumes that the phase detector output is constantly proportional to the input phase difference. In reality, the phase detector output is a pulse applied for a period of time equal to the phase difference. The function of C_2 is to smooth the phase detector output (VCO control voltage) over each cycle. C_2 also adds a second pole to the filter transfer function. This pole should be far enough outside the loop bandwidth (at least one order of magnitude) that its phase and amplitude contribution is negligible in the loop bandwidth. If:

$C_2 = C_1/50 = 240$ pF (choose 300 pF)

The final loop component is R_{BOOST}. Since R_{RATE} and the parallel combination of R_{RATE} and R_{BOOST} are known, we can calculate R_{BOOST}.

$R_{BOOST} = (R_P)(R_{RATE}) / (R_{RATE} - R_P) = 820\Omega$

The above filter values and those for other bandwidths are listed on preceding page.



TL/F/8445-17

FIGURE 13. Bode Plot of Loop Response

Circuit Operation (Continued)

The calculated values are only a guide, the user should then empirically test the loop and determine stability, lock-on time, jitter tolerance, etc.

The desired Bode plot of gain and phase is shown in *Figure 13*, with 20 dB/decade slope at ω_0 for stability at unity gain.

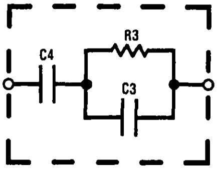
Capacitor C_2 governs the PLL's ability to reject instantaneous bit jitter. As C_2 increases in value, the effective jitter rejection will also increase. However, as the frequency of the pole R_1 and C_2 produce (while increasing C_2) decreases, loop stability will decrease, and the second-order approximation used to analyze the circuit becomes inaccurate. Thus, it is recommended that C_2 remain one tenth (or less) the value of C_1.

The value of resistor R_1 inversely effects the break frequencies on the Bode plot, and directly effects the loop's damping ratio (overshoot response). The capacitor C_1 governs the bandwith of the loop. Too high a value will slow down the response time, but make the PLL less prone to jitter or frequency shift whereas too low a value will improve response time while tending to increase the PLL's reaction to jitter.

Other filter combinations may be used, other than R_1 in series with C_1, all in parallel with C_2. For example the filter shown in *Figure 14* will also perform similarly, and in fact for some systems it will yield superior performance.

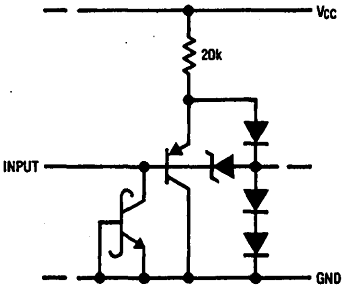
DIGITAL CONNECTIONS TO THE DP8461/65

Figure 17 shows a connection diagram for the DP8461/65 in a typical application. All logic inputs and outputs are TTL compatible as shown in *Figure 15* and *16*. The VCO CLOCK output is 74AS compatible. All other outputs are 74ALS compatible. All inputs are 74ALS compatible and therefore can be driven easily from any 74 series devices. The raw MFM from the pulse detector in the drive is connected to the ENCODED DATA input. The DELAY DISABLE input determines whether attempting lock-on will begin immediately after READ GATE is set or after 2 bytes. Typically in a hard-sectored drive, READ GATE is set active as the sector pulse appears, meaning a new sector is about to pass under the head. Normally the preamble pattern does not begin immediately, because gap bytes from the preceding sector usually extend just beyond the sector pulse. Allowing 2 bytes to pass after the sector pulse helps ensure that the PLL will begin locking on to preamble, and will not be chasing non-symmetrical gap bits. Thus DELAY DISABLE should be set low for this kind of disk drive.



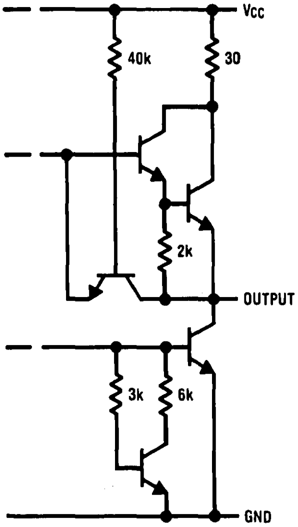
TL/F/8445-18

FIGURE 14. Alternate Loop Filter Configuration



TL/F/8445-19

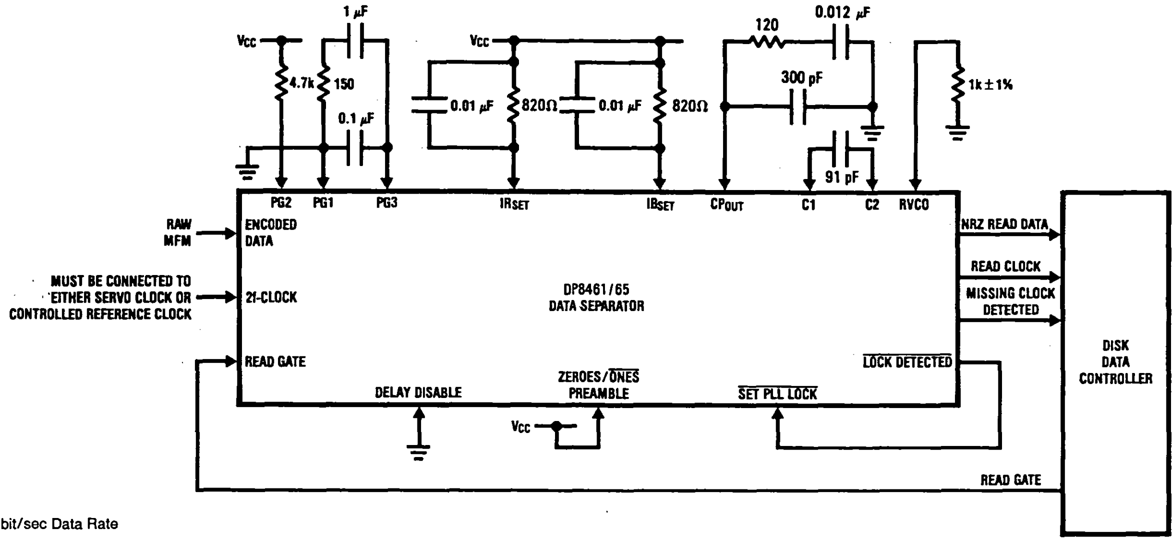
FIGURE 15. Logic Inputs



TL/F/8445-20

FIGURE 16. Logic Outputs

Connection Diagram



1) MFM Data Input, 5 Mbit/sec Data Rate

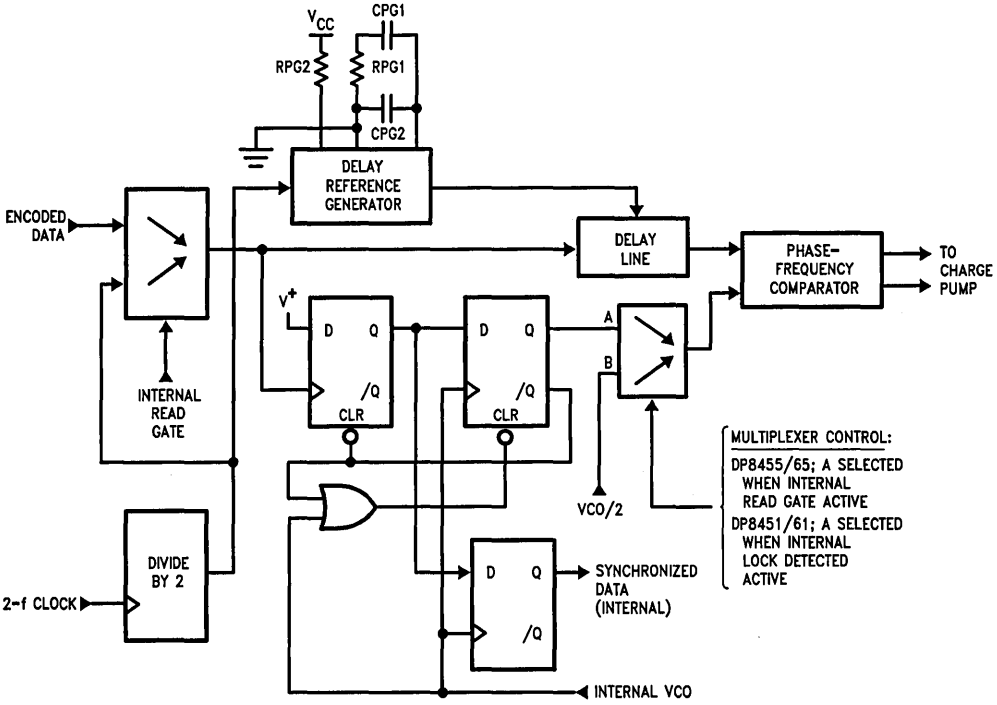
2) 32 Bit Delay to Enable

3) All Zeroes (NRZ) Preamble

TL/F/8445-21

FIGURE 17. Typical Connection to DP8461/65

Block Diagram


V_CC
CPG1
RPG2
RPG1
CPG2
DELAY REFERENCE GENERATOR
ENCODED DATA
DELAY LINE
PHASE-FREQUENCY COMPARATOR
TO CHARGE PUMP
V+
D Q
/Q
CLR
A
B
INTERNAL READ GATE
MULTIPLEXER CONTROL:
DP8455/65; A SELECTED WHEN INTERNAL READ GATE ACTIVE
DP8451/61; A SELECTED WHEN INTERNAL LOCK DETECTED ACTIVE
VCO/2
DIVIDE BY 2
2-f CLOCK
SYNCHRONIZED DATA (INTERNAL)
INTERNAL VCO

TL/F/8445-25

Circuit Operation (Continued)

For soft sectored drives, the controller normally will not wait for the index pulse before it attempts lock-on, so that READ GATE may go active at any time. Chances are the head will not be over a preamble field and therefore there is no need to wait 2 bytes before attempting lock-on. DELAY DISABLE can therefore be set high. If a non-preamble field is passing by as READ GATE goes active, the DP8461/65 will not indicate lock, and no data decoding will occur nor will MISSING CLOCK DETECTED go active. Normally, if lock-on has not been achieved after a certain time limit, the controller will de-activate READ GATE and then try again.

For MFM encoded disk drives, the $\overline{\text{LOCK DETECTED}}$ output will be connected back to the $\overline{\text{SET PLL LOCK}}$ input. As the PLL achieves lock-on, the DP8461/65 will automatically switch to the lower tracking rate and decoded data will appear at the NRZ READ DATA output. Also the READ CLOCK output will switch from half the 2F-CLOCK frequency to the disk data rate frequency. If a delay is required before the changeover occurs, a time delay may be inserted between the two pins.

Some drives have an all-ONES data preamble instead of all-ZEROES and the DP8461/65 must be set to the type being used before it can properly decode data. The ZEROES/$\overline{\text{ONES}}$ PREAMBLE input selects which preamble type the chip is to base its decoding phase on.

USE WITH RUN-LENGTH-LIMITED CODES (RLL)

If the drive uses a Run-Length-Limited Code (RLL) such as 1,7 or 1,8 instead of MFM, the user might choose to use the DP8451/55. These circuits contain the PLL portion of the DP8461/65 and thus perform the data synchronization function. RAW DATA is input to pin 16 and the 2F-CLOCK is applied to pin 17. Instead of supplying NRZ DATA, SYNCHRONIZED DATA OUTPUT is issued at pin 12. The VCO CLOCK, pin 8, is used to clock this data into external decoding circuitry. As long as the high frequency pattern of ... 1010 ... is used for the preamble, the user may choose the DP8451 if he desires to have the circuit perform phase and frequency comparisons until two bytes of preamble are detected by the on chip preamble pattern detector.

If a 2,7 code is being used the DP8465/55 may be used. Again, since the DP8465 MFM decoding function will not be used, the user may choose to use the DP8455. However, the National Semiconductor DP8462 is designed specifically for the 2,7 code. It is recommended that the user reviews the DP8462 specification for the added advantages the circuit offers with the 2,7 format.

Applications of the DP8461/65 Data Separator

The DP8461/65 are the first integrated circuits to place on one chip a PLL with features that offer the improved speed and reliability required by the disk industry. Not only does each chip simplify disk system design, but also provides fast lock-on to the incoming preamble. Once locked on, the loop is set into a more stable mode. This inherent loop stability allows for a sizeable amount of jitter on the data stream, such as is encountered in many disk systems. Once in the stable tracking rate, the SYNCHRONIZED DATA output represents the incoming ENCODED DATA and is synchronous with VCO CLOCK. If the disk is MFM encoded, then the chip can decode the synchronized data into NRZ READ DATA and READ CLOCK. These are available as outputs from the chip allowing the NRZ READ DATA to be deserialized using the READ CLOCK.

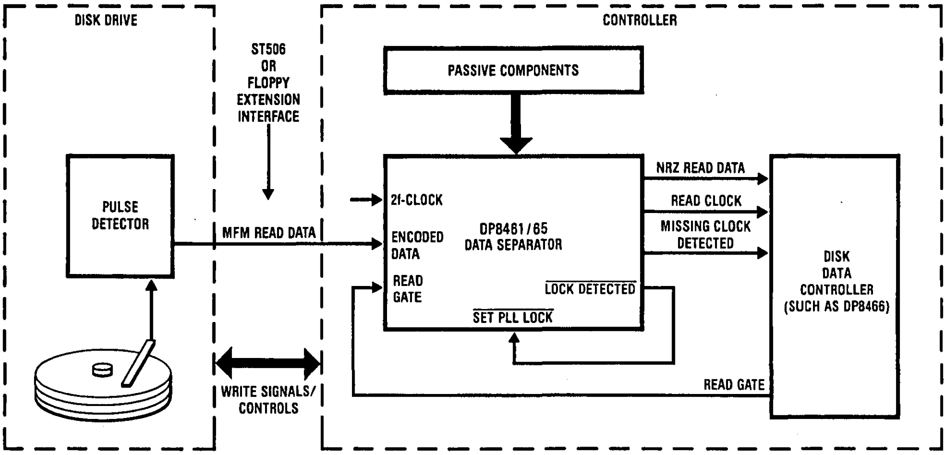
The DP8461/65 are capable of operating at up to 20 Mbits/sec data rates and so are compatible with a wide assortment of disk drives. The faster data rates of the 8-inch and 14-inch disk drives will mandate the selection of the DP8461/65-3 parts with their narrower window margins on the incoming data stream. This will also be the case when 5¼-inch drives achieve higher data rates. Some 8-inch and 14-inch disk drives incorporate the functions of the DP8461/65, but use many discrete ICs. In these cases, replacing these components with the DP8461/65 will offer reduced P.C. board area, lower cost, and improved performance while simplifying circuit testing.

Most 5¼-inch and many 8-inch and 14-inch disk drives manufactured at present do not incorporate any of the functions of the DP8461/65. This is so primarily because the PLL function is difficult to design and implement and requires circuitry which covers a large area of the printed circuit card. This is undesirable both from the drive size aspect and from the cost aspect (the cost includes soldering, testing, and adjusting the components). Consequently, most smaller disk drives output MFM encoded data so that the phase-locked-loop and data separation have to be performed by the controller. The DP8461/65 will therefore replace these functions in controller designs, as shown in *Figure 18.*

System design criteria has become more flexible because the DP8461/65 provide a one-chip solution, requiring only a few external passive components with fixed values. Each operates from a +5V supply, typically consumes about 0.3W, and is housed in a narrow 24-pin package. The circuitry has been designed so that the external resistors and capacitors need not be adjustable; the user chooses the values according to the disk drive requirements. Once selected, they will be fixed for that particular drive type. These features make it possible to transfer these functions to the disk drive, as shown in *Figure 19.* Apart from a slight increase in board area, the advantages outweigh the disadvantages. First, the components selected are fixed for each type of drive and this facilitates the problem of interchangeability of drives. At present, controllers are adjusted to function with each specific drive; with the DP8461/65 in the drive, component adjustment will no longer be required. Second there is often a problem of reliability of data transfer. The data returning from the disk drive is susceptible to noise, bit shift, etc. Soft errors will occur when the incoming disk data bit position is outside the Pulse Gate window as it is being synchronized to the VCO clock in the phase-locked-loop. Obviously, the nearer the PLL is to the data source, the less chance there is that extraneous noise or transmission line imbalances will cause errors to occur. Thus placing the DP8461/65 in the drive will increase the reliability of data transfer within the system.

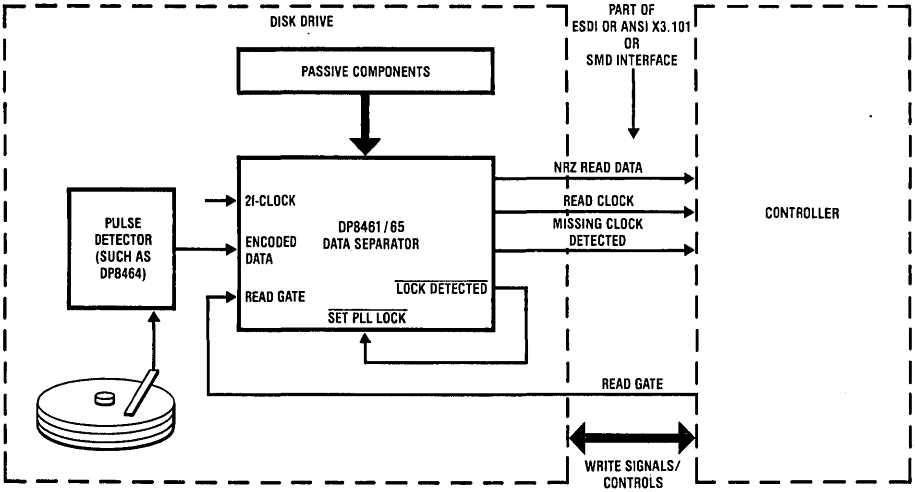
A third advantage is data rate upgrading. Most 5¼-inch drives have 5 Mbit/sec data rate because the early drives were made with this data rate. This meant the controllers had to be designed with PLLs which operate at this data rate. It is therefore difficult for drive manufacturers to introduce new drives that are not compatible with existing controllers. Since no new standard data rate has emerged, they must continue to produce drives at this data rate to be compatible with the controllers on the market. With the DP8461/65 in the drive, and associated components set for the drive's data rate, it no longer becomes a problem to increase the data rate, assuming the controllers digital circuitry can accommodate the change. This will allow the manufactures to increase the bit density and therefore the capacity of their drives.

Applications of the DP8461/65 Data Separator (Continued)



TL/F/8445-22

FIGURE 18. DP8461/65 in the Controller



TL/F/8445-23

FIGURE 19. DP8461/65 in the Disk Drive

PRECAUTIONS IN BREADBOARDING AND PCB LAYOUT

The DP8461/65 contains a high performance analog PLL and certain precautions must be taken when breadboarding or designing a PCB layout. The following guidelines should be adhered to when working with the DP8461/65:

1) Do not wire wrap.
2) Keep component lead lengths short, place components as close to pins as possible.This applies to R1, C1, R2, C_{VCO}, R_{RATE}, R_{BOOST}, C_{RATE}, C_{BOOST}, RPG1, RPG2, and CPG1.
3) Provide a good ground plane and use a liberal amount of supply bypassing. The quieter a PLL's environment, the happier it is.
4) Avoid routing any digital leads within the vicinity of the analog leads and components.
5) Keep inter-pin capacitance to a minimum; i.e., avoid running traces or planes between pins.
6) Minimize digital output pin capacitive loading to reduce current transients.

NSC has used a PC board approach to breadboarding the DP8461/65 that gives an excellent ground plane and keeps component lead lengths very short. With this setup very stable and reliable operation has been observed. Illustration of component layout is shown in *Figure 20*.

Applications of the DP8461/65 Data Separator (Continued)

ADDITIONAL NOTES

1. PG1 should be grounded to improve noise immunity.
2. 2F clock must be applied at all times; without the 2F clock, the pulse gate circuitry will not operate properly making it impossible to lock onto the incoming data stream.
3. The programming capacitor for the V_{CO} can be calculated as:

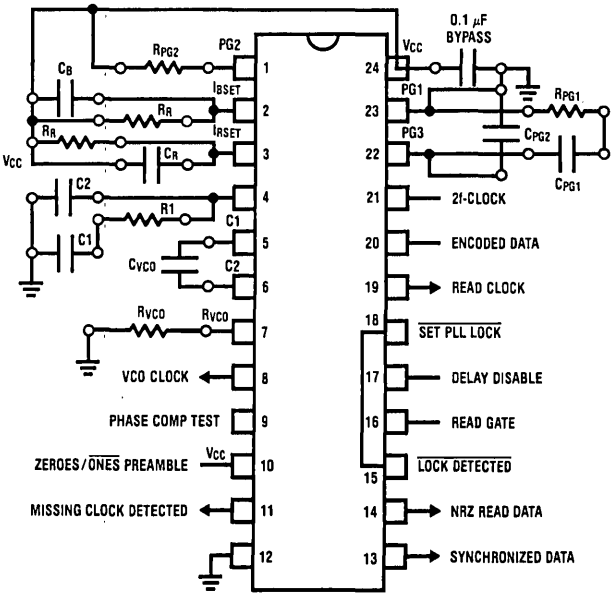
$$C_{VCO} = 1/(f_{VCO} \times R_{VCO}) - 5\text{ pF}$$

The 5 pF value is due to parasitic and pin to pin capacitance. An additional accomodation must also be made for PC board capacitance.

4. Care must be taken in final PC board layout to minimize pin to pin capacitance, particularly in multi-layer printed circuit boards.
5. Please refer also to Precautions for Disk Data Separator Designs, NSC Application Note AN-414.

Connection Diagrams

DP8461/65

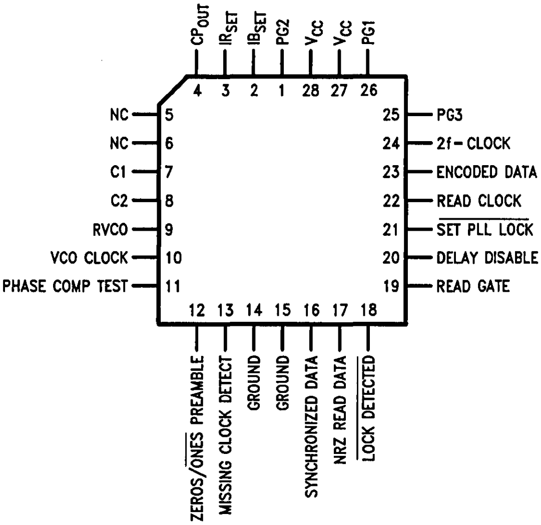


TL/F/8445-24

FIGURE 20. Recommended Component Layout

Connection Diagrams (Continued)

Plastic Chip Carrier



TL/F/8445-26

Order Number DP8461V or DP8465V
See NS Package Number V28A

National Semiconductor Corporation

DP8462 2,7 Code Data Synchronizer

General Description

The DP8462 Data Synchronizer is designed for application in disk drive memory systems employing Run Length Limited Codes using 1-0-0 or 1-0-0-0 preamble patterns, and depending on system requirements, may˝be located either in the drive or in the controller. It receives digital pulses from a pulse detector circuit (such as the DP8464 Disk Pulse Detector) if the DP8462 is situated in the drive, or from an interface if it is situated in the controller. In the read mode, the circuit locks onto and detects either a 100 or 1000 preamble pattern depending on the state of the pattern select input pin. The synchronized data and clock are then available for decoding and deserialization by a decoder circuit. All of the digital input and output signals are TTL compatible and only a single +5V supply is required. Although separate Analog and Digital V_{CC} and Ground pins are provided, they are expected to be tied together by the user. The chip is housed in a standard narrow 24-pin dual-in-line package (DIP) and is fabricated using Advanced Schottky bipolar analog and digital circuitry. This high speed I.C. process allows the chip to work with data rates up to 20 Mbits/sec. There are two versions of the chip, each having a different decode window error specification. These two versions (-3 and -4) will operate from 4 to 20 Mbits/sec, with respectively increasing window errors, as specified in the Electrical Characteristics Table.

The DP8462 features a phase-locked-loop (PLL) consisting of a pulse gate, phase comparator, charge pump, buffering amplifier, and voltage-controlled-oscillator (VCO). Pins are provided for the user to select the values of the external filtering components required for the pulse gate and PLL, the frequency setting components required for the VCO, two current setting resistors for the charge pump, and current setting resistors for the pulse gate that control the delay line.

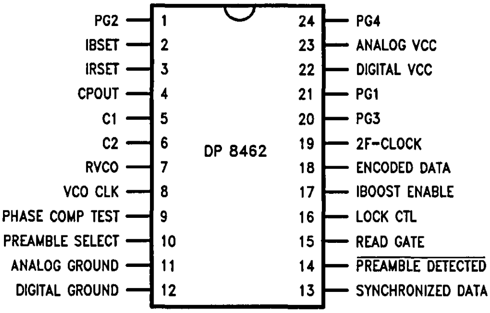
The on-board PLL's phase comparator has two modes of operation: phase and frequency comparison or phase only comparison. In the non-read mode, the comparator performs phase and frequency comparison, but once in the read mode, it switches to phase only comparison. The user selects whether this mode change occurs as soon as read mode is entered or after the preamble pattern is detected. The charge pump also has two modes of operation: high track rate—intended to be used in the non-read mode and in the read mode while acquiring lock, and low track rate—intended to be used in the read mode to retain lock. Both track rates are selected by the user with external components; the user is given control over when the track rate switch takes place.

Features

- Phase-Frequency PLL in non-read mode and during preamble if desired
- Operates at data rates up to 20 Mbit/sec
- Detects either 1-0-0 or 1-0-0-0 preamble patterns
- User determined PLL loop filter network
- PLL charge pump has two user-determined tracking rates
- External control of track-rate switchover
- External control of phase comparator switchover
- Delay line may be externally adjusted if desired
- ORed phase comparator outputs for monitoring bit-shift
- Standard narrow 24-pin DIP or 28-pin Plastic Chip Carrier package
- Less than ½W power consumption
- Single +5V supply

Connection Diagrams

Dual-In-Line Package

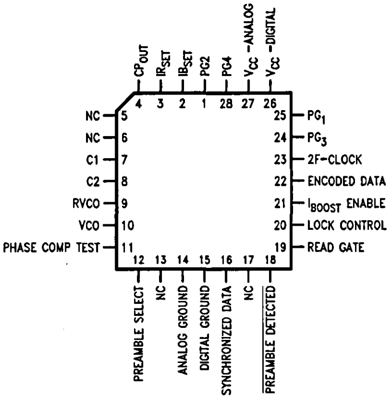


TL/F/8418-2

Top View

Order Number DP8462N
See NS Package N24C

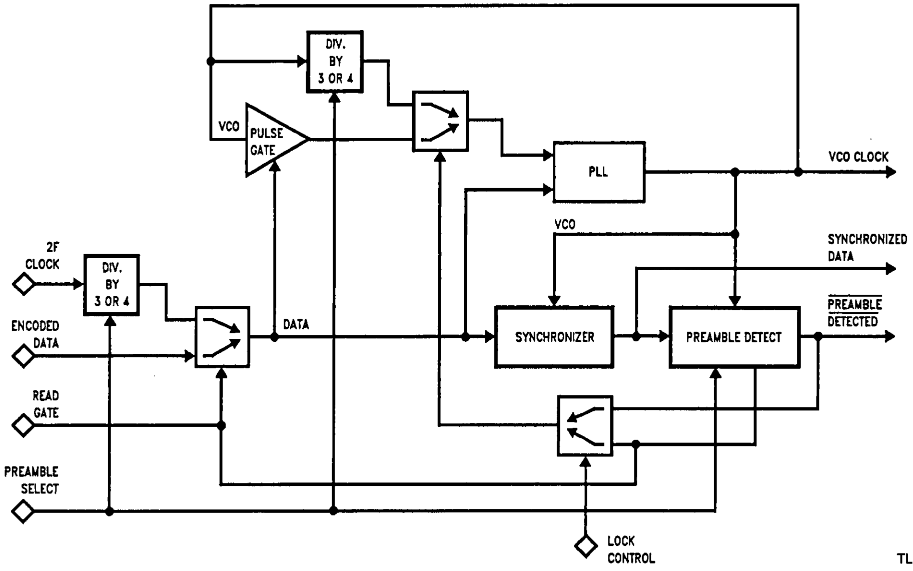
Plastic Chip Carrier



TL/F/8418-24

Order Number DP8462V
See NS Package V28A

Block Diagram



TL/F/8418-1

Pin Descriptions*

POWER SUPPLY

22,23 Digital and Analog V_{CC} = +5V ±5%
Should be tied together and bypassed by user.

11,12 Analog and Digital Ground
Should be tied together by user.

TTL Level Logic Inputs

15 READ GATE: When asserted, this signal sets the DP8462 into the Read Mode. The PLL then begins to lock onto the encoded data.

10 PREAMBLE SELECT: A high level on this input enables the circuit to recognize a 1-0-0-0 pattern while a low level results in the recognition of a 1-0-0 pattern. Also, in the non-read mode, if 1-0-0 is selected VCO/3 will lock onto 2F/3 while if 1-0-0-0 is selected VCO/4 will lock onto 2F/4.

16 LOCK CTL: This input allows the user to determine when the circuit will switch from Phase-Frequency comparison to Phase only comparison once in the Read Mode. A low level on this pin causes the circuit to switch from the phase-frequency comparison mode as soon as READ GATE is asserted while a high level means that the circuit will switch after 4 bytes of preamble have been detected and PREAMBLE DETECTED output has been asserted. (See the Truth Table at the end of this Section.)

17 IBOOST ENABLE: This input allows the user to control the PLL's track rate by turning Iboost current on and off. A high level at this input causes Iboost to be added to Irate—placing the PLL in the high track rate. In a typical system IBOOST ENABLE may be tied to READ GATE or PREAMBLE DETECTED.

18 ENCODED DATA: This input is for the incoming encoded data from the output of the head amplifier/pulse-detecting network located on the disk drive. Each positive edge of the ENCODED DATA waveform identifies a flux reversal on the disk.

19 2F-CLOCK: This is a system clock input, which is either a signal generated from the servo track, or a signal buffered from a crystal. It operates at twice the NRZ DATA rate. 2F-CLOCK MUST ALWAYS BE APPLIED TO THIS INPUT FOR PROPER OPERATION.

TTL Level Logic Outputs

8 VCO CLOCK: This is the output of the on-chip VCO, transmitted from an Advanced Schottky TTL buffer. It is synchronized to the SYNCHRONIZED DATA output so that it can be used by the encoder/decoder circuitry.

13 SYNCHRONIZED DATA: This is the same encoded data that is input to the chip, but is synchronous with the VCO CLOCK.

14 PREAMBLE DETECTED: After READ GATE is asserted, this output goes low after detecting approximately 4 bytes of preamble and remains low until READ GATE goes inactive.

9 PHASE COMP TEST: This output is the logical "OR" of the Phase Comparator outputs, and may be used for the testing of the disk media.

Analog Signals

21,20 PG1, PG3: The external capacitors and resistor of the Pulse Gate filter are connected to these pins. PG1 should be tied directly to ground.

1 PG2: This is the Pulse Gate delay reference pin. The delay reference generator establishes a voltage at this pin; thereby producing the bias current for the Pulse Gate delay section in the resistor tied between this pin and V_{CC}.

*Pin Number Designations apply for the 24 Pin DIP. See Connection Diagram for the Plastic Chip Carrier Pin Designations.

Pin Descriptions (Continued)

24 PG4: This is the Pulse Gate delay control pin. This pin can be tied to the PG2 pin if the user desires to adhere to the chip's standard synchronization window specification; otherwise, it can be tied to PG2 through a "current splitting" network (see *Figure 6*) —thereby shifting the synchronization window early or late.

3 IRSET: The current into the rate set pin (Vbe/Rrate) is approximately half the charge pump output current for the low tracking rate.

2 IBSET: The current into the boost set pin (Vbe/Rboost) is approximately half the amount by which the charge pump current is increased for the high tracking rate.
(I_{hirate} = I_{rate} Set + I_{boost} Set).

4 CPOUT: This pin is the output node of the charge pump and also the noninverting input of the Buffer Amplifier. It is made available for connection of external filter components for the phase-locked-loop.

5,6 VCO C1, C2: An external capacitor connected across these pins sets the nominal frequency.

7 RVCO: The current into this pin determines the operating currents within the VCO.

Note: ANALOG and DIGITAL V_{CC} pins must be tied together by the user.
ANALOG and DIGITAL GND pins must also be tied together by the user.

Truth Table of Pulse-Gate's Modes

LOCK CTL (Pin 16)	READ GATE (Pin 15)	$\overline{\text{PREAMBLE DETECTED}}$ (Pin 14)	Pulse-Gate Comparison Mode	Comments
LO	LO	LO	N/A	N/A
LO	LO	HI	Phase and Frequency	Non-Read Mode
LO	HI	HI	Phase only	Read Mode
LO	HI	LO	Phase only	Read Mode
HI	LO	LO	N/A	N/A
HI	LO	HI	Phase and Frequency	Non-Read Mode
HI	HI	HI	Phase and Frequency	Read Mode
HI	HI	LO	Phase Only	Read Mode

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage	7V
TTL Inputs	7V
Output Voltages	7V
Input Current (CPOUT, IRSET, IBSET, RVCO)	2 mA
Storage Temperature	−65°C to +150°C
Operating Temperature Range	0°C to +70°C

Operating Conditions

Symbol	Parameter		Conditions	Min	Typ	Max	Units
V_{CC}	Supply Voltage			4.75	5.00	5.25	V
T_A	Ambient Temperature			0	25	70	°C
I_{OH}	High Logic Level Output Current		V_{CO} Clock Others			−2000 −400	µA
I_{OL}	Low Logic Level Output Current		V_{CO} Clock Others			20 8	mA
f_{DATA}	Input Data Rate			4.0		20	Mbit/sec
t_{WCK}	Width of 2f-CLOCK, High or Low			10			ns
t_{WPD}	Width of ENCODED DATA Pulse (Note 1)	High		18			ns
		Low		0.4t			ns
V_{IH}	High Logic Level Input Voltage			2			V
V_{IL}	Low Logic Level Input Voltage					0.8	V
t_{SU} Read Gate	Min Time Required for a Positive Edge of Read Gate to Occur Before a Negative Edge of VCO			20			ns
t_{HOLD} Read Gate	Min Time Required for a High Level on Read Gate to be Held After a Negative Edge of VCO			10			ns

Note 1: t is defined as the period of the NRZ data (t = 2/F_{VCO}).

DC Electrical Characteristics Over Recommended Operating Temperature Range

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max	V_{CC} −2V	V_{CC} −1.6V		V
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max			0.5	V
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V			20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V			−200	μA
I_O	Output Drive Current (Note 1)	V_{CC} = Max, V_O = 2.125V	−12		−110	mA
I_{CC}	Supply Current	V_{CC} = Max			85	mA
I_{OUT}	Charge Pump Output Current	$200 \leq I_{RATE}+I_{BOOST} \leq 2000$	0.9 I_{TYP} −25	2.0 (I_{RATE} + I_{BOOST})	1.1 I_{TYP} +25	μA

Note 1: This value has been chosen to produce a current that closely approximates one-half of the true short-circuit output current, I_{OS}.

AC Electrical Characteristics Over Recommended V_{CC} and Operating Temperature Range

(All Parts unless stated otherwise) ($t_R = t_F$ = 2.0 ns, V_{IH} = 3.0V, V_{IL} = 0V) (Note 1)

Symbol	Parameter	Min	Typ	Max	Units
t_{READ}	Positive VCO CLOCK transitions from READ GATE set active until PLL Lock sequence begins (DELAY DISABLE high)		1	1	—
$t_{TRANSMIT}$	Positive VCO CLOCK transitions required to transmit input encoded data to output	1	2	3	—
$t_{READ\ ABORT}$	Number of VCO CLOCK cycles after READ GATE set low to read operation abort (Note 3)			2	T-clock
t_{WINDOW}	Variance of center of decode window from nominal (Note 6) DP8462-3 DP8462-4			6 10	ns
ϕLINEARITY	Phase range for charge pump output linearity (Note 2)	$-\pi$		$+\pi$	Radians
K_1	Phase comparator—Charge Pump gain constant ($N = f_{VCO}/f$ input data) (Note 4)		$\frac{1.78\ V_{BE}}{N2\pi R}$		Amps/rad
$V_{CONTROL}$	Charge pump output voltage swing from nominal		±100		mV
K_{VCO} (= A × K_2)	VCO gain constant (ω_{VCO} = VCO center frequency in rad/s) (Note 5)	$\frac{1.20\ \omega_C}{V_{BE}}$	$\frac{1.40\ \omega_C}{V_{BE}}$	$\frac{1.60\ \omega_C}{V_{BE}}$	rad/sec V
f_{VCO}	VCO center frequency variation over temperature and V_{CC}	−2		+2	%
$f_{MAX\ VCO}$	VCO maximum frequency		60		MHz
t_{PHL}	Propagation delay from VCO negative edge to synchronous DATA negative edge	2		18	ns
t_{PLH}	Propagation delay from VCO negative edge to synchronous DATA positive edge	4		20	ns

Note 1: A sample calculation of frequency variation vs. control voltage: V_{IN} = ±0.1V;

$$K_{VCO} = \frac{\omega OUT}{V_{IN}} = \frac{0.4\ \omega_C}{0.2V} = \frac{2.0\ \omega_C}{V}\frac{(\text{rad/sec})}{(\text{volt})}.$$

Note 2: $-\pi$ to $+\pi$ with respect to 2f VCO CLOCK.

Note 3: T-clock is defined as the time required for one period of the VCO CLOCK to occur.

Note 4: With respect to VCO CLOCK; $I_{PUMP\ OUT}$ = 1.9 I_{SET}

$$I_{SET} = \frac{V_{BE}}{R_{SET}}$$

Note 5: Although specified as the VCO gain constant, this is the gain from the Buffer Amplifier input to the VCO output.

Note 6: This specification is guaranteed only for the conditions under which the parts were tested. However, significant variation from formula is not expected for other data rates and filters. This specification is for the condition when PG2 and PG4 are tied together. External adjustment can be used to optimize t_{WINDOW} as described in the pulse gate section. The filter values below were chosen for operation in an automatic test system (static window) environment. Different criteria may apply for choosing filter values in a disk system. See Loop Filter Section for sample calculations of other filter values.

Part Type	Data Rate Tested	Filter				
		C_1	C_2	R_1	R_{RATE}	R_{BOOST}
DP8462-4 DP8462-3	5 Mbit/sec 10 Mbit/sec	0.03 μF 0.022 μF	600 pF 510 pF	100Ω 81Ω	820Ω 800 kΩ	1.5 kΩ 1.8 kΩ

Note: For further informtion refer to Application Note AN-414

External Component Selection (All Parts) (Note 1)

Symbol	Component	Min	Typ	Max	Units
R_{VCO}	VCO Frequency Setting Resistor (Note 2)	990		1010	Ω
C_{VCO}	VCO Frequency Setting Capacitor (Notes 3,4)	20		120	pF
R_{RATE}	Charge Pump I_{RATE} Set Resistor (Note 6)	0.4		4.0	kΩ
R_{BOOST}	Charge Pump (High Rate) I_{BOOST} Resistor (Note 6)	0.5		∞	kΩ
C_R	I_{RATE} Bypass Capacitor (Note 5)	0.01			μF
C_B	I_{BOOST} Bypass Capacitor (Note 5)	0.01			μF

Note 1: External component values for the Loop Filter and Pulse Gate are given in Table II and Table I respectively.

Note 2: A 1% Component Tolerance is Required.

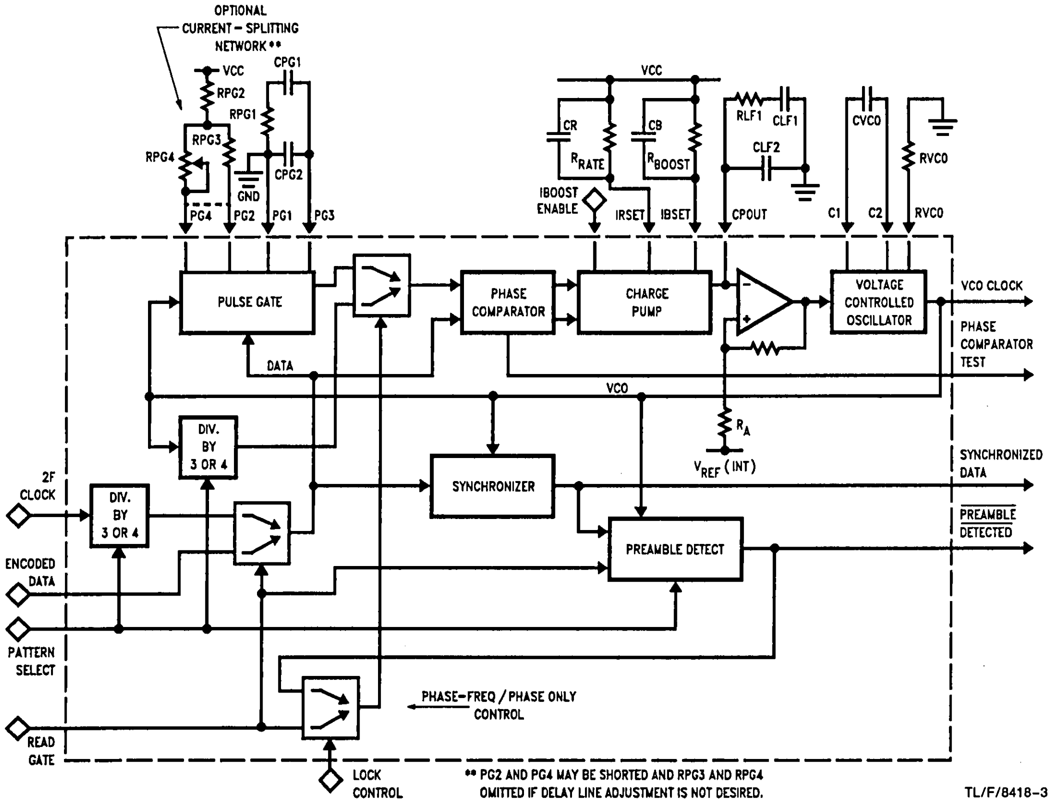
Note 3: These MIN and MAX values correspond to the MAX and MIN data rates respectively.

Note 4: The Component Tolerance is system dependent on how much center frequency deviation can be tolerated.

Note 5: Component Tolerance 15%.

Note 6: The minimum value of the parallel combination of R_{RATE} and R_{BOOST} is 350Ω.

Detailed Block Diagram



** PG2 AND PG4 MAY BE SHORTED AND RPG3 AND RPG4 OMITTED IF DELAY LINE ADJUSTMENT IS NOT DESIRED.

TL/F/8418-3

Circuit Operation

In the non-read mode, the DP8462 Data Separator remains locked to the 2f CLOCK signal divided by 3 or 4 (depending upon the preamble used) in anticipation of a preamble when read mode is entered. When the READ GATE input goes high, the DP8462 enters the read mode after 1 VCO CLOCK cycle. Referring to *Figure 1*, once in the read mode, the PLL reference signal is switched from the 2f-divided-by-3-or-4 signal to the ENCODED DATA input. The PLL is at this point in the high-tracking rate mode and also in the Phase and Frequency Comparison mode. The PLL then attempts to

Circuit Operation (Continued)

quickly lock onto the incoming ENCODED DATA stream and starts looking for 16 consecutive preamble pulses—chosen by the user to be either 100 (PATTERN SELECT: LO) or 1000 (PATTERN SELECT: HI). If the user has chosen to switch to Phase Only Comparison as soon as read operation begins (LOCK CTL: LO), then the Phase Comparator will start to compare ENCODED DATA with VCO-gated-by-DATA immediately (see *Figure 2*); otherwise, it will keep comparing ENCODED DATA with VCO divided by 3 or 4—i.e., remain in Phase and Frequency Comparison mode until after 16 consecutive preamble pulses have been detected. At this time, PREAMBLE DETECTED output goes low and the circuit now starts to compare ENCODED DATA with VCO-gated-by-DATA (see *Figure 1*).

The user is given control over when to switch the charge-pump current rate through the use of the IBOOST ENABLE input. One way the user can accomplish this is by tying PREAMBLE DETECTED output to the IBOOST ENABLE input directly. Thus, once PREAMBLE DETECTED is asserted, the circuit will go into low track rate and Phase Only Comparison mode (if LOCK CTL: HI) so that a more stable lock can be retained. The incoming ENCODED DATA stream is now synchronized with the VCO CLOCK and appears at the SYNCHRONIZED DATA output (see *Figure 3*). (If the user wishes to switch to low track rate as soon as the circuit enters the read mode, then the READ GATE signal should be inverted and applied to the IBOOST ENABLE input).

Figure 4 shows the sequence when READ GATE goes low, signifying the end of a read operation. The PLL reference signal is switched back to 2f divided by 3 or 4 input and the PREAMBLE DETECTED output goes high (causing the charge-pump to go to the high tracking rate, if PREAMBLE DETECTED is tied to the IBOOST ENABLE input). Also, the Phase Comparator goes back to Phase and Frequency Comparison mode and the circuit attempts to lock onto the 2f divided by 3 or 4 signal, thus returning to the initial conditions.

CIRCUIT DESCRIPTION

1. Divide by 3 or 4: Depending on the preamble pattern being used, these circuits divide 2f CLOCK and internal VCO CLOCK signals by 3 or 4. During the non-read mode, the VCO remains phase and frequency locked to these divided signals so that when read mode is entered, the PLL can quickly acquire lock because the data stream that consists of the preamble pattern is very close in frequency to the VCO divided by 3 or 4.

2. Pulse Gate: Once in the read mode, the PLL has to lock the VCO CLOCK to the ENCODED DATA stream; outside of the preamble, however, the data signal is not cyclic like the VCO CLOCK and therefore cannot be frequency compared to the VCO. It is for this reason that the Pulse Gate is used to allow a reference signal from the VCO into the Phase Comparator only when an ENCODED DATA bit is valid. The Pulse Gate also utilizes a scheme which delays the incoming data by one-half the period of the 2f-CLOCK. This optimizes the position of the decode window and allows input jitter up to ± half the 2f-CLOCK period, assuming no error in the decode window position. The decode window error can be determined from the specification in the Electrical Characteristics Table.

3. Multiplexers at the Phase Comparator's inputs: These multiplexers are used to determine which signals the Phase Comparator will compare during different modes of operation. Either 2F divided by 3 or 4 or ENCODED DATA is compared with either VCO divided by 3 or 4 (Phase and Frequency Lock) or with VCO gated by DATA (Phase Only Lock).

4. Phase Comparator: The Phase Comparator receives its inputs from the Multiplexers mentioned above, and is edge-triggered from these inputs to provide charge-up and charge-down outputs.

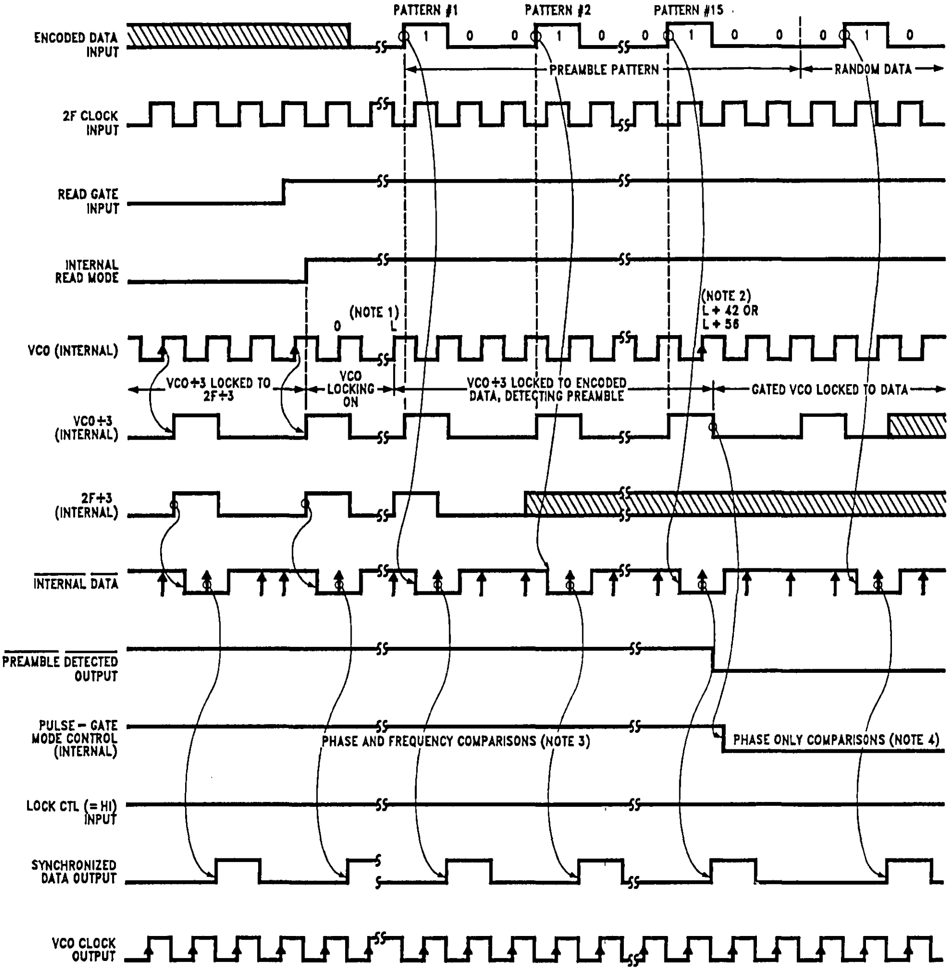
5. Charge Pump: The high speed charge pump consists of a switchable constant current source and sink. The charge pump constant current is set by connecting external resistors to V_{CC} from IRSET and IBSET pins. With IBOOST ENABLE HIGH, the PLL is in the high tracking rate and both resistors determine the current. With IBOOST ENABLE LOW, the PLL is in the low tracking rate and only the IRSET resistor determines the charge pump current. The output of the charge pump feeds into external filter components and the Buffer Amplifier. Thus, through the use of the IBOOST ENABLE pin, the user can determine when the circuit switches track rates.

6. Buffer Amplifier: The Buffer Amplifier is configured as a high input impedance amplifier which is inserted between the charge pump and the VCO, thus allowing connection of external PLL filter components to the charge pump output pin CPOUT. The output of the Buffer Amplifier is internally connected to the VCO control input.

7. VCO: The Voltage-Controlled-Oscillator requires a resistor from the RVCO pin to ground and a capacitor between pins C1 and C2, to set the center frequency. The VCO frequency can be varied from nominal by approximately ±20%, as determined by its control input voltage (CPOUT).

8. Preamble Pattern Detector: Two types of preamble patterns are commonly used in RLL 2,7 code disk systems—1-0-0 and 1-0-0-0. The user selects the preamble pattern to be used by setting PREAMBLE SELECT input either HI for the 1-0-0-0 pattern or LO for the 1-0-0 pattern. The DP8462 divides 2F Clock and VCO Clock signals by 3 or 4 depending upon whether 1-0-0 or 1-0-0-0 pattern is selected, respectively, and remains locked to this divided pattern in anticipation of a preamble. Once the chip is in the read mode, the VCO proceeds to lock onto the incoming data stream. The Preamble Pattern Detector then searches for 16 consecutive patterns (i.e., 100100100... or 100010001000...) to indicate lock has been achieved. The PREAMBLE DETECTED output then goes low. Any deviation from the above-mentioned continuous stream of patterns before 16 of these are detected will reset the Pattern Detector and the procedure will then start over again.

Circuit Operations (Continued)



TL/F/8418-4

Note 1: L = Number of VCO cycles required for VCO to lock—typically 20 but determined by external component value.

Note 2: At L + 42 (Pattern = 1-0-0) or L + 56 (Pattern = 1-0-0-0), 15 patterns have been detected.

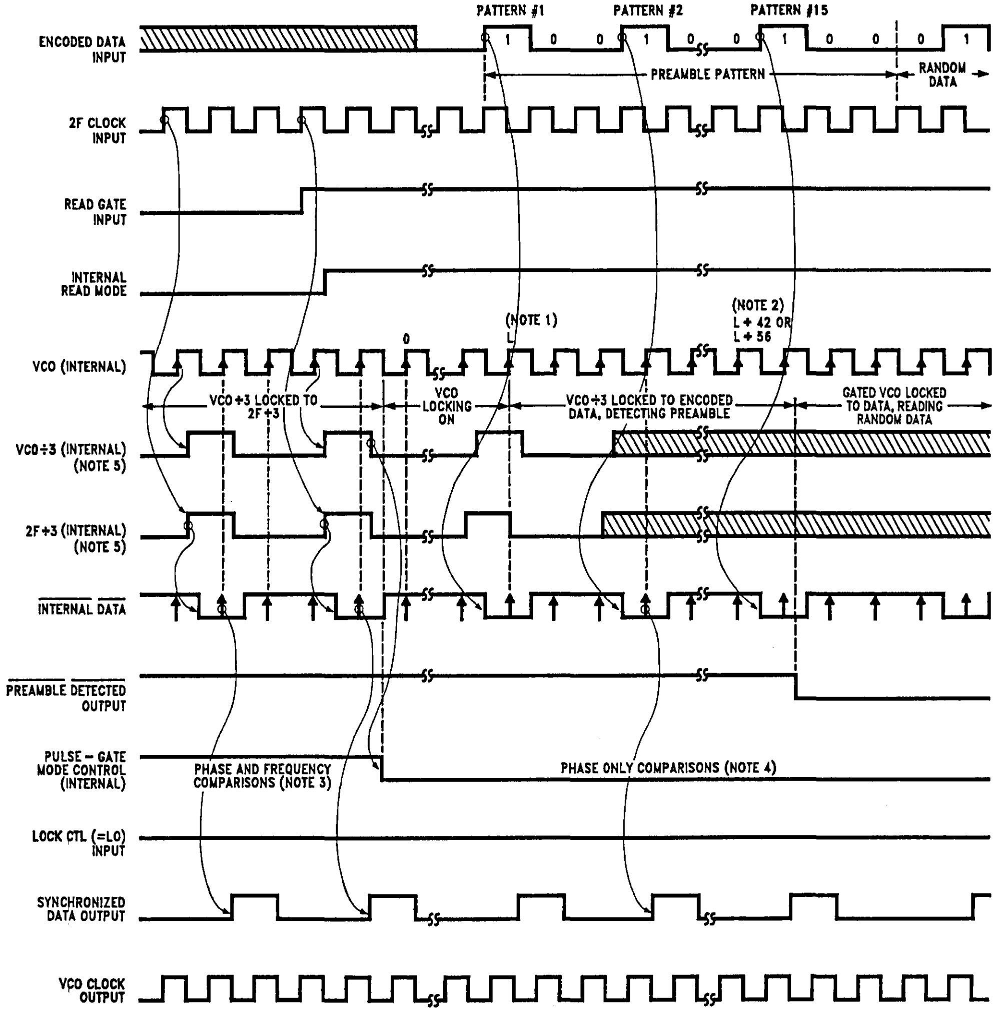
Note 3: VCO ÷ 3 (or 4) being compared with 2F ÷ 3 (or 4) in the non-read mode and Preamble in the Read Mode.

Note 4: VCO GATED BY DATA being compared with ENCODED DATA.

Note 5: PREAMBLE SELECT = LO; 100 pattern selected—so 2F & VCO are being divided by 3.

FIGURE 1. Lock-On Sequence Waveform Diagram—Pulse Gate Mode Switches after Preamble Detection

Circuit Operation (Continued)



TL/F/8418-5

Note 1: L = Number of VCO cycles required for VCO to lock—typically 20 but determined by external component value.

Note 2: At L + 42 (Pattern = 1-0-0) or L + 56 (Pattern = 1-0-0-0), 15 patterns have been detected.

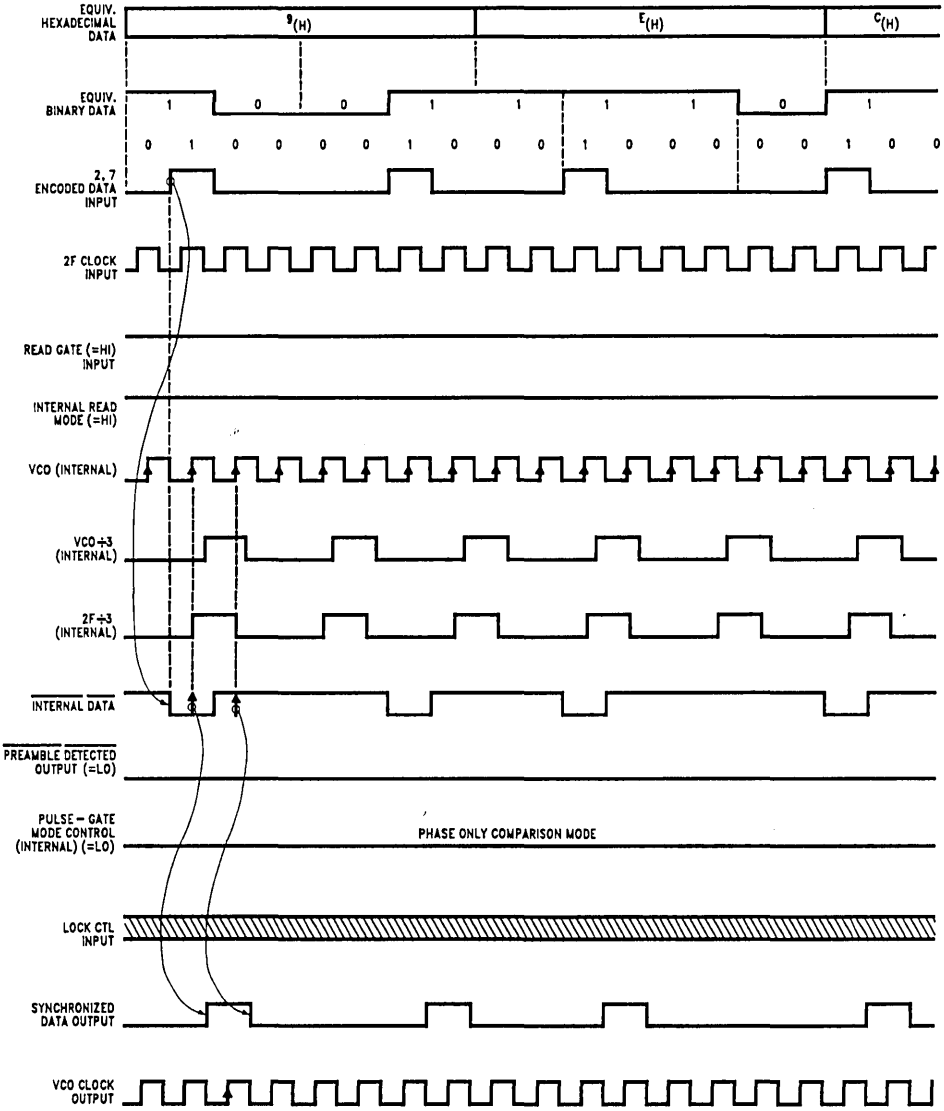
Note 3: VCO ÷ 3 (or 4) being compared with 2F ÷ 3 (or 4) in the non-read mode.

Note 4: VCO gated by DATA being compared with ENCODED DATA.

Note 5: PREAMBLE SELECT = LO; 1-0-0 pattern selected—so 2F & VCO are being divided by 3.

FIGURE 2. Lock-On Sequence Waveform Diagram—Pulse Gate Mode Switches Immediately After READ GATE Is Asserted

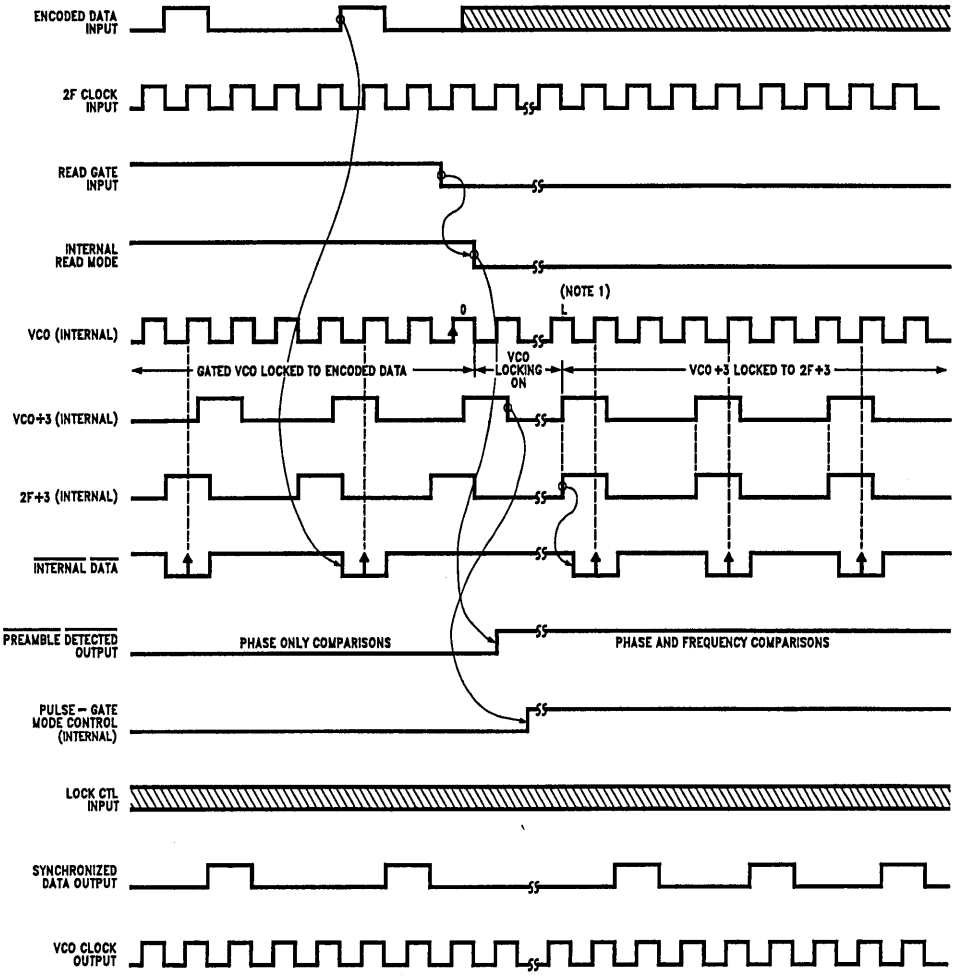
Circuit Operation (Continued)



TL/F/8418-6

FIGURE 3. Locked-On Waveform Diagram

Circuit Operation (Continued)



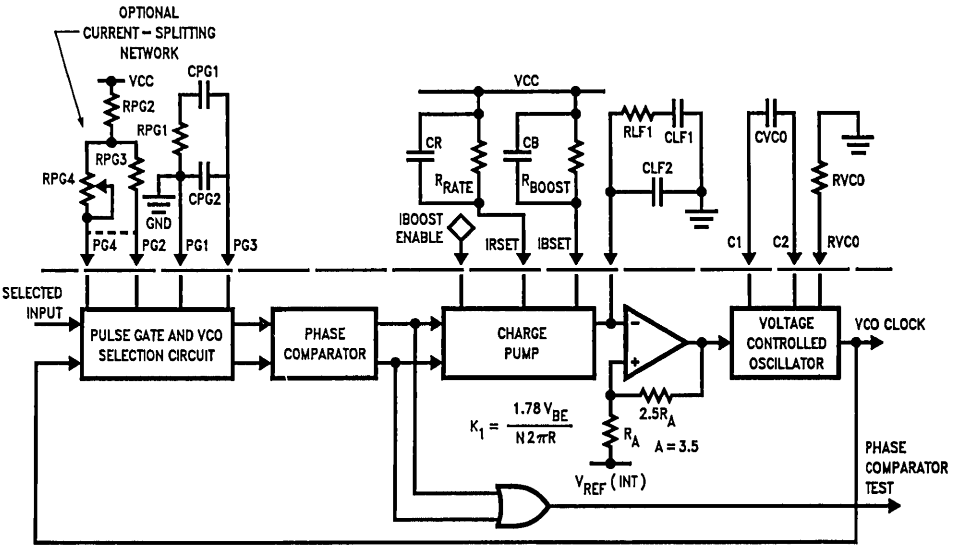
TL/F/8418-7

Note 1: L indicates the number of cycles required for the VCO to lock to the 2f-Clock.

Note 2: PREAMBLE SELECT = LO; 1-0-0 Pattern selected—so 2F & VCO being divided by 3.

FIGURE 4. Lock-Ending Sequence Waveform Diagram

Circuit Operation (Continued)



TL/F/8418-8

FIGURE 5. Phase-Locked-Loop Section

BIT JITTER TOLERANCE

The two options of the DP8462, the -4 and -3 offer decreasing window errors (respectively) so that the parts may be selected for different data rates (up to 20 Mbit/sec). The -4 part will be used in most low data rate applications. As an example, at the 5 Mbit/sec data rate of most 5¼ inch drives, T = 200 ns so that from the Electrical Characteristics Table, t_{WINDOW} = 10 ns. The chip therefore contributes up to 10 ns of window error, out of the total allowable error of 50 ns (half the 2f-clock period of 100 ns). This allows the disk drive to have a margin of 40 ns of jitter on the transition position before an error will occur. The bit jitter tolerance can be improved by adjusting the window center using PG2 and PG4. A current splitting network consisting of RPG3 and RPG4 can be used to adjust the delay line. This adjustment is internally compensated for V_{CC} and temperature variation.

ANALOG CONNECTIONS TO THE DP8462

External passive components are required for the Pulse Gate, Charge Pump, Loop Filter and VCO as shown in *Figure 5*. The information provided here is for guidelines only. The user should select values according to his own system requirements. Phase-Locked Loops are complex circuits that require detailed knowledge of the specific system. Factors such as loop gain, stability, response to change of signal, lock-on time, etc. are all determined by the external components. In many disk systems these factors are critical, and National Semiconductor recommends the designer be knowledgeable of phase-locked-loops, or seek the advice of an expert. Inaccurate design will probably result in excessive disk error rates. The phase-locked-loop in the DP8462 has many advantages over all but the most sophisticated discrete designs, and if the component values are selected correctly, it will offer significant performance advantages. This should result in a reduction of disk error rates over equivalent discrete designs.

PULSE GATE

There are 6 external components connected to the Pulse Gate as shown in *Figure 6* with the associated internal components. Of these, RPG3 and RPG4 are optional and may be omitted if adjustment of the delay line is not desired. The values of RPG1, RPG2, RPG3, RPG4, CPG1, and CPG2 are dependent on the data rate. RPG1 and RPG2 are inversely proportional to the data rate, while CPG1 and CPG2 are proportional. Table I shows component values for the data rates given. Component values are calculated by selecting RPG2 from Table I [RPG2' = RPG2 + (RPG3//RPG4)]. Next calculate

$$\text{CPG1} = \left(\frac{2.12 \times 10^5}{890 + \text{RPG2'}}\right)\left(\frac{1}{100 \times \text{Rs}}\right)^2$$

$$\text{CPG2} = \frac{1}{10}\,\text{CPG1, and}$$

$$\text{RPG1} = \left(\frac{890 + \text{RPG2'}}{2.38 \times 10^5}\right)(100 \times \text{Rs}).$$

In the above equation Rs is the rotational speed and, for 3600 RPM, Rs = 60 Hz. A rotational speed of 3600 RPM was assumed for the calculations in Table I. For data rates not listed, RPG2 may be approximated as (30 kΩ/fDATA) − 1.2 kΩ = RPG2 where fDATA is the data rate in Megabits per second. RPG3 and RPG4, in conjuction with RPG2, form a "current-splitting-network" that can be used to adjust the delay line; thus adjusting the decode window early or late. RPG2 should be made large with respect to RPG3 and RPG4 and a potentiometer can be used for RPG4—with its value centered around that of RPG3. For example, at Data Rate = 5 Mbits/sec., Table I dictates that RPG2' should be 4.7k. If the delay line is to be made adjustable, then one could pick RPG2 = 4.3k and RPG3 = 800Ω. Now, using a 1.6 kΩ potentiometer for RPG4, RPG4 = 800Ω would give RPG2' = 4.7 kΩ and would provide standard window synchronization; varying RPG4 high or low, however, would

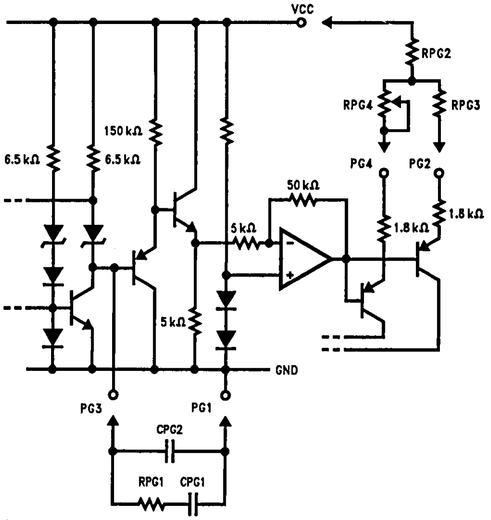
Circuit Operation (Continued)

shift the window late or early, respectively. If no adjustment is desired, then PG2 and PG4 should be tied together and only RPG2 should be used. Components with 5% tolerance will suffice.

TABLE I. Pulse Gate Component Selection Chart

Data Rate	RPG2'	RPG1	CPG1	CPG2
5 Mbit/sec	4.7 kΩ	150Ω	1 µF	0.1 µF
10 Mbit/sec	1.8 kΩ	68Ω	2.2 µF	0.22 µF
15 Mbit/sec	750Ω	39Ω	3.9 µF	0.39 µF

Where [RPG2' = RPG2 + RPG3//RPG4]



TL/F/8418-9

FIGURE 6. Pulse-Gate Controls

CHARGE PUMP

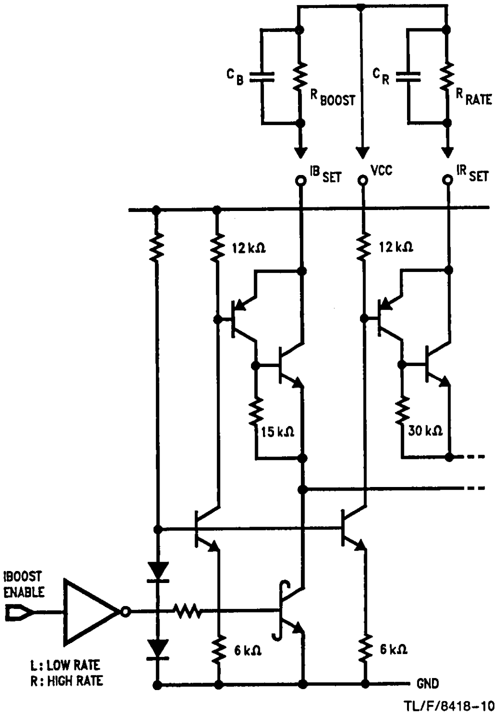
Resistors R_{RATE} and R_{BOOST} determine the charge pump current. The Charge Pump bidirectional output current is related to the input current according to the relationship specified in the DC Electrical Characteristics Table. In the high tracking rate with IBOOST ENABLE high, the input current is I_{BSET} + I_{RSET}, i.e., the sum of the currents through R_{BOOST} and R_{RATE} from V_{CC}. In the low tracking rate, with IBOOST ENABLE low, this input current is I_{RSET} only.

A recommended approach would be to select R_{RATE} first. The External Component Limits table allows R_{RATE} to be 0.4 kΩ to 4.0 kΩ, so for simplicity select R_{RATE} = 820Ω. A typical loop gain change of 2:1 for high to low tracking rate would require R_{BOOST} = R_{RATE} or 820Ω. Referring to *Figure 7* the input current is effectively V_{BE}/R_{RATE} in the low tracking rate, where V_{BE} is an internal voltage. This means that the current into or out of the loop filter is approximately 2.0 V_{BE}/R_{RATE}, or in this example approximately 1.8 mA. Note that although it would seem the overall gain is dependent on V_{BE}, this is not the case. The VCO gain is altered internally by an amount inversely proportional to V_{BE}, as detailed in the section on the Loop Filter. This means that as V_{BE} varies with temperature or device spread, the gain will remain constant for a particular fixed set of values of R_{RATE} and R_{BOOST}. This alleviates the need for potentiometers to select values for each device. The tolerance required for these two resistors will depend on the total loop gain tolerance allowed, but 5% would be typical. Also V_{CC} by-pass capacitors are required for these two resistors. A value of 0.01 µF is suitable for each.

VCO

The value of R_{VCO} is fixed at 1 kΩ ± 1% in the External Component Limits table. *Figure 8* shows how R_{VCO} is connected to the internal components of the chip. This value was fixed at 1 kΩ to set the VCO operating current such that optimum performance of the VCO is obtained for production device spreads. This means fixed value components will be adequate to set the VCO center frequency for production runs. The value of C_{VCO} can therefore be determined from the VCO frequency f_{VCO}, using the equation: $C_{VCO} = [1/(R_{VCO})(f_{VCO})] - 5\ pF$ where f_{VCO} is twice the input data rate. As an example, for a 5 Mbit/sec data rate, f_{VCO} = 10 MHz, requiring that C_{VCO} = 95 pF. This does not take into account any lead capacitance on the printed circuit board; the user must account for this. The amount of tolerance a particular design can afford on the center frequency will determine the capacitor tolerance. The capacitor is connected to the internal circuitry of the chip as shown in *Figure 9.*

As the data rate increases and C_{VCO} gets smaller, the effects of unwanted parasitic capacitances influence the fre-



TL/F/8418-10

FIGURE 7. I_{RATE} Set and I_{BOOST} Set

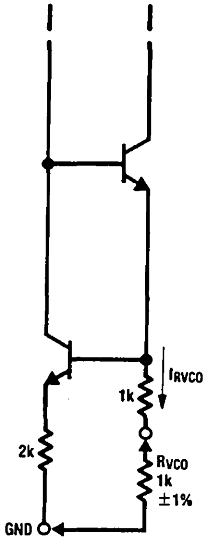
Circuit Operation (Continued)

quency. As a guide the graph of *Figure 10* shows approximately the value of C_{VCO} for a given data rate.

The VCO center frequency may be determined by: 1) holding pin 4 at ground potential and measuring the VCO frequency (−20% value); 2) holding pin 4 at approximately 3 volts and measuring the VCO frequency (+20% value); 3) averaging the two measured frequencies for the equivalent center frequency.

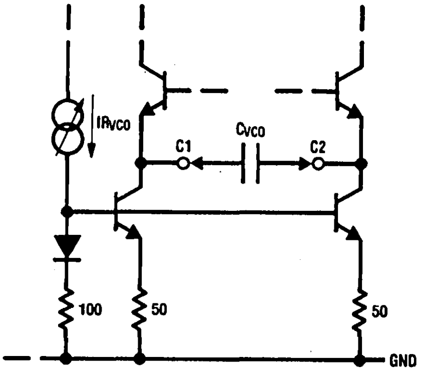
LOOP FILTER

The input current into the Buffer Amplifier is offset by a matched current out of the Charge Pump, and even so is much less than the switching current in or out of the Charge Pump. It can therefore be assumed that all the Charge Pump switching current goes into the Loop Filter components R_1 and C_1 and C_2. The tolerance of these compo-



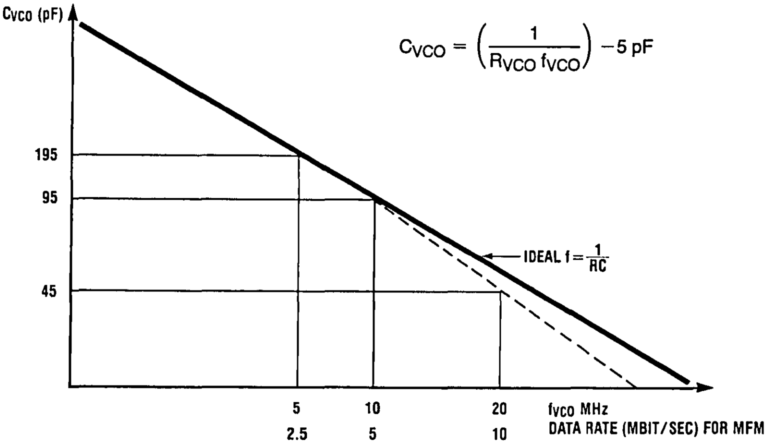
TL/F/8418-11

FIGURE 8. VCO Current Setting Resistor



TL/F/8418-12

FIGURE 9. VCO Capacitor



TL/F/8418-13

FIGURE 10. VCO Capacitor Value for Disk Data Rates

Circuit Operation (Continued)

nents should be the same as R_{RATE} and R_{BOOST}, and will determine the overall loop gain variation. The three components connected to the Charge Pump output are shown in *Figure 11.* Note the return current goes to analog GND, which should be electrically very close to the GND pin itself.

The value of capacitor C_1 determines loop bandwidth—the larger the value the longer the loop takes to respond to an input change. If C_1 is too small, the loop will track any jitter on the ENCODED DATA input and the VCO output will follow this jitter, which is undesirable. The value of C_1 should therefore be large enough so that the PLL is fairly immune to phase jitter but not large enough that the loop won't respond to longer term data rate changes that occur on the disk drive.

The damping resistor R_1 is required to damp any oscillation on the VCO input that would otherwise occur due to step function changes on the input. A value of R_1 that would give a phase margin of around 45 degrees would be a reasonable starting point.

The main function of the capacitor C_2 is to "smooth" the VCO input voltage. Typically its value will be less than one tenth of C_1.

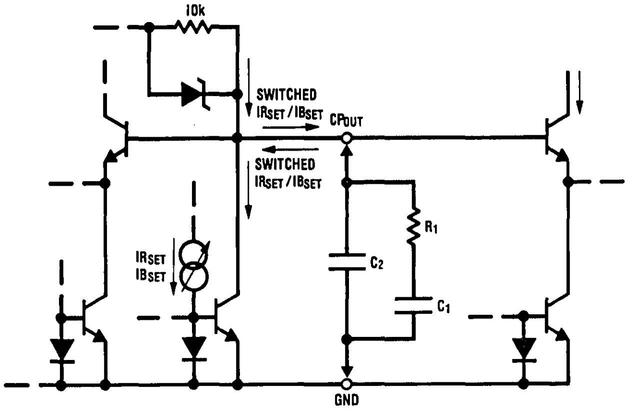
Figure 12 shows the relevant phase-locked-loop blocks that determine system response, namely the Phase Detector, Filter/Buffer Amplifier, and VCO. The Phase Detector (Phase Comparator and Charge Pump) produces an aggregate output current i which is proportional to the phase difference between the input signal and the VCO signal. The constant (K_1) is $\frac{1.78\,V_{BE}}{2\pi RN}$ amps per radian where $N = \frac{f_{VCO}}{f_{DATA}}$. R is either R_{RATE} or $R_{RATE} \parallel R_{BOOST}$. This aggregate current feeds into or out of the filter impedance (Z), producing a voltage to the VCO that regulates the VCO frequency. The VCO gain constant is 0.4 ω_{VCO}/V_{BE} radians per second per volt. Under steady state conditions, i will be zero and there will be no phase difference between the input signal and the VCO. Any change of input signal will produce a change in VCO frequency that is determined by the loop gain equation. This equation is determined from the gain constants K_1, A and K_2 and the filter v/i response.

The impedance Z of the filter is:

$$\frac{1}{sC_2} \parallel \left(\frac{1}{sC_1} + R_1\right) = \frac{1 + sC_1R_1}{sC_1(1 + \frac{C_2}{C_1} + sC_2R_1)}$$

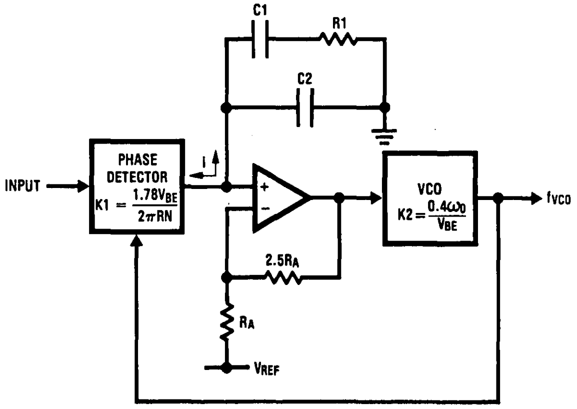
If $C_2 < C_1$ then the impedance Z approximates to:

$$\frac{1 + sC_1R_1}{sC_1\,(1 + sC_2R_1)}$$



TL/F/8418-14

FIGURE 11. Charge Pump Out



TL/F/8418-15

FIGURE 12. Loop Response Components

Circuit Operation (Continued)

The overall loop gain is then

$$G(s) = \frac{K_1AK_2}{s} \times \frac{1 + sC_1R_1}{sC_1(1 + sC_2R_1)}$$

Let $G_{(K)} = K_1\ A\ K_2$

$$F(s) = \frac{1 + SC_1R_1}{SC_1\ (1 + SC_2R_1)}$$

The Overall Closed Loop Gain is:

$$\frac{\phi OUT}{\phi IN} = \frac{G_{(K)}\ F(s)}{s + G_{(K)}\ F(s)}$$

Substituting, We Get

$$\frac{\phi_{OUT}}{\phi_{IN}} = \frac{G_{(K)}\ (SC_1R_1 + 1)}{S^3\ R_1\ C_1\ C_2 + S^2\ C_1 + GK\ (SC_1R_1 + 1)}$$

$$= \frac{(G_{(K)}/C_1)(SR_1C_1 + 1)}{S^3\ R_1C_2 + S^2 + SG_{(K)}R_1 + G_{(K)}/C_1}$$

If $C_2 \lessdot C_1$, we can ignore the 3rd Order Component introduced by C_2 then:

$$\frac{\phi_{OUT}}{\phi_{IN}} = \frac{(G_{(K)}/C_1)(SR_1C_1 + 1)}{S^2 + SG_{(K)}R_1 + G_{(K)}/C_1}$$

This is a second Order Loop and can be solved as follows:

$$S^2 + SG_{(K)}R_1 + G_{(K)}/C_1 = S^2 + 2\zeta\ \omega_nS + \omega_n{}^2$$

$$\therefore C_1 = \frac{G_{(K)}}{\omega_n{}^2}$$

$$R_1 = \frac{2\zeta\omega_n}{G_{(K)}}$$

From the above equations:

$$\omega = (G(K)/C1)^{1/2}$$

$$G(K) = K1 \times A \times K2 =$$

$$[(0.89 \times V_{BE})/(2 \times Pi \times R)] \times [(0.4 \times W_{VCO})/V_{BE}] \times [3.5]$$

2,7 coded data has a 2.67 to 1.0 frequency range within the data field. The expression $K = (0.89 \times V_{BE}/2 \times Pi \times R)$ is valid when the VCO frequency is twice the ENCODED DATA frequency. In order to make this equation more general, it may be written as follows: $K = (1.78 \times V_{BE})/(2 \times Pi \times R \times N)$ where N is defined as the VCO frequency divided by the encoded data pulse frequency, or $N = F_{VCO}/F_{DATA}$ ($N = 3$ for maximum data rate and $N = 8$ for minimum data rate). Now G(K) can be written as follows:

$$G(K) = [(1.78 \times V_{BE})/(2 \times Pi \times R \times N)] \times$$

$$[(0.4 \times \omega_{VCO})/V_{BE}] \times [3.5]$$

$$= (2.5 \times F_{VCO})/(R \times N)$$

$$\omega n = [(2.5 \times F_{VCO})/(C1 \times R \times N)]^{1/2}$$

where,

$R = R_{RATE}$ in the low track rate;

$R = (R_{RATE}//R_{BOOST})$ in the high track rate.

From the above equations:

$$\omega_N = (R1 \times G(K))/(2\zeta)$$

$$G(K) = C1 \times (\omega_N{}^2)$$

$$\zeta = (\text{damping factor}) = (R1 \times \omega_N \times C1)/2$$

The damping factor should be approximately 0.5 when ω_N is minimum. Response to bit shift is minimized when the damping factor is small; however, if the damping factor drops much below 0.5, the system tends to be oscillatory (underdamped). Additionally, loop performance is poor (excessive phase-acquisition times) if the damping factor becomes much larger than 1.0. Any increase in loop bandwidth (due to R decreasing in the high track rate) produces a proportional increase in the damping factor, and should be limited to the point where the maximum damping factor is 1.0. With the damping factor range established, loop design can proceed.

From the Disk Interface Design Guide And User's Manual Chapter 1, Section 1.3–1.7, it is shown that a 946 krads/sec loop bandwidth during acquisition results in a 7 byte crystal reference clock acquisition and data frequency acquisition (VCO settled to within 2 ns of window center). We recommend that these design guide sections be reviewed in conjunction with the DP8462 data sheet in order to obtain a more detailed explanation of the loop bandwidth selection used here; as well as for disk system PLLs in general.

This design example is for a 10 MBit/sec data rate and assumes that the IBOOST ENABLE pin is tied to the PREAMBLE DETECTED pin. This results in the track rate being switched from high to low after four bytes of preamble are detected. As an alternative, the IBOOST ENABLE pin may be tied to an inverted READ GATE signal, resulting in the track rate switching immediately to low when READ GATE is asserted. This is discussed further in the Design Guide.

We will assume a 1-0-0-0 . . . preamble. During acquisition we are in the high track rate and thus ω_N is at maximum value. In the read mode the highest frequency pattern we can encounter is 1-0-0 . . . ; however, ω_N will be lower since we will be in the low track rate.

$$\omega_N = [(2.5 \times F_{VCO})/(C1 \times R \times N)]^{1/2}$$

Choose $R_P = R_{RATE}\ //\ R_{BOOST} = 575\Omega$*

$$946 \times 10^3 = [(2.5 \times 20 \times 10^6)/(C1 \times 575 \times 4)]^{1/2}$$

$C1 = 0.028\ \mu F$ Choose $C1 = 0.022\ \mu F$

We don't want ζ to exceed 1.0. Therefore,

$$\zeta = \frac{\omega_N \times R1 \times C1}{2}$$

$$1.0 = \frac{(946 \times 10^3 \times R1) \times 0.022 \times 10^{-6}}{2}$$

$R1 = 96\Omega$

Choose $R1 = 100\Omega$

***Note:** Designing a PLL is an iterative procedure. For the DP8462, design values for R_{RATE} and R_{BOOST} typically range from 700Ω to 1.5 kΩ. The application note provides a more thorough discussion for choosing these values.

The continuous-behavior (non-quantized) approximation used to predict loop performance assumes that the phase detector output is constantly proportional to the input phase difference. In reality, the phase detector output is a pulse applied for a period of time equal to the phase difference. The function of C2 is to "smooth" the phase detector output (VCO control voltage) over each cycle. C2 also adds a second pole to the filter transfer function. This pole should be far enough outside the loop bandwidth (at least one order of magnitude) that its phase and amplitude contribution is negligable in the loop bandwidth. If

$$C2 = C1/50 = 390\ pF$$

the acquisition performance and the margin loss are not significantly changed from the predictions. If a larger C2 is used, the margin loss can be reduced at the expense of the acquisition time. This may be desirable for some systems. Please see the Disk Interface Design Guide And User's Manual Chapter 1, Sections 1.3–1.7 for a discussion of the function of C2.

Circuit Operation (Continued)

As soon as the $\overline{\text{PREAMBLE DETECTED}}$ output goes low we switch to the low track rate. To maintain stability we must ensure that $\zeta_{min} \geq 0.5$.

ζ_{min} occurs when ω_N is minimum; i.e., when we have seven consecutive zeroes (N = 8).

$$\zeta_{min} = \frac{(\omega_{Nmin} \times R1 \times C1)}{2}$$

$$0.5 = \frac{(\omega_{Nmin} \times 100 \times 0.022 \times 10^{-6})}{2}$$

$$\omega_{Nmin} = 454.5 \text{ krads/sec}$$

We can now calculate R_{RATE}

$$\omega_{Nmin} = [(2.5 \times F_{VCO})/(C1 \times R_{RATE} \times N)]^{1/2}$$

$$454.5 \times 10^3 = [(2.5 \times 20 \times 10^6)/(0.022 \times 10^{-6} \times R_{RATE} \times 8)]^{1/2}$$

Therefore, $R_{RATE} = 1.375\ \text{k}\Omega$

Choose, $R_{RATE} = 1.2\ \text{k}\Omega$

Now we calculate ω_{Nmax} and ζ_{max} in the low track rate

$$\omega_{Nmax} = [(2.5 \times 20 \times 10^6)/(0.022 \times 10^{-6} \times R_{RATE} \times 3)]^{1/2}$$

$$\omega_{Nmax} = 794 \text{ krads/sec}$$

$$\zeta_{max} = \frac{(\omega_{Nmax} \times R1 \times C1)}{2}$$

$$\zeta_{max} = 0.87$$

The final component to be determined is R_{BOOST}

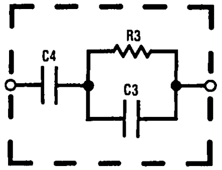
$$\text{Since, } Rp = \frac{R_{BOOST} \times R_{RATE}}{R_{BOOST} + R_{RATE}}$$

$$575 = \frac{R_{BOOST} \times 1.2 \times 10^3}{R_{BOOST} + 1.2 \times 10^3}$$

Therefore, $R_{BOOST} = 1.1\ \text{k}\Omega$

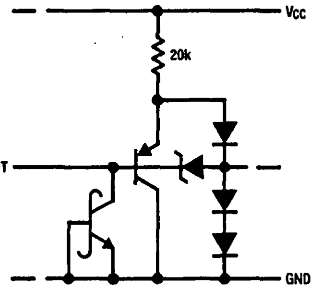
DIGITAL CONNECTIONS TO THE DP8462

Figure 16 shows a connection diagram for the DP8462 in a typical application. All logic inputs and outputs are TTL compatible as shown in *Figure 14* and *15*. The VCO Clock output is 74AS compatible. All other outputs are 74ALS compatible. All inputs are 74ALS compatible and therefore can be driven easily from any 74 series devices.



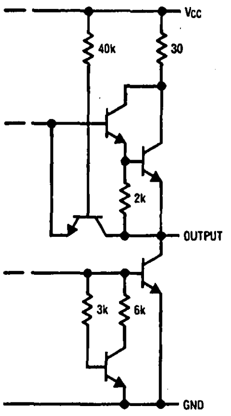
TL/F/8418-17

FIGURE 13. Alternate Loop Filter Configuration



TL/F/8418-16

FIGURE 14. Logic Inputs



TL/F/8418-18

FIGURE 15. Logic Outputs

TABLE II. Loop Filter External Component Values

Data Rate (NRZ)	Pulse Gate Components (Note 3)				Charge Pump (Note 1)		Loop Filter (Note 2)		
	R_{PG2}	R_{PG1}	C_{PG1}	C_{PG2}	R_{RATE}	R_{BOOST}	R_1	C_1	C_2
5 Mbit/sec	4.7k	150Ω	1.0 μF	0.1 μF	820Ω	1.5 kΩ	100Ω	0.03 μF	600 pF
10 Mbit/sec	1.8k	68Ω	2.2 μF	0.22 μF	1.2 kΩ	1.1 kΩ	100Ω	0.022 μF	390 pF
15 Mbit/sec	0.75k	39Ω	3.9 μF	0.39 μF	820Ω	2.7 kΩ	33Ω	0.082 μF	1600 pF

Note 1: Component tolerances are system dependent, they depend on how much loop gain deviation can be tolerated.

Note 2: Component tolerances are typically 5% but they depend on the amount of Loop Bandwidth tolerance that can be accepted.

These values have been altered from calculated values based on empirical tests of the loop.

Note 3: Component tolerances typically 10%, not critical.

Circuit Operation (Continued)

The incoming data from the pulse detector in the drive is connected to the ENCODED DATA input. PREAMBLE SELECT input is tied high or low depending on whether the user's system is employing 1000 or 100 preamble pattern. The LOCK CTL input is to be tied high or low depending on whether or not it is desired to keep the PLL in Phase and Frequency comparison mode while detecting preamble. Phase and Frequency comparison lock while detecting preamble will eliminate the chances of the PLL locking onto a harmonic of the preamble frequency when Read mode is first entered. (Susceptibility to a harmonic lock is increased when using the 1000 preamble). Since a high level on IBOOST ENABLE input puts the PLL in high track rate, it should be held high during Non-Read (standby) mode so that a quick lock is achieved upon entering Read mode. Once the PLL is locked onto the incoming data, however, this input should be taken low. Although the user is free to do this anytime, one possible method is to tie this input to the $\overline{\text{PREAMBLE DETECTED}}$ output of the chip—as shown in *Figure 16*. The READ GATE input is used to place the chip in and out of Read mode and therefore should be tied to the controller and/or a 2, 7 code Encoder/Decoder).

As for the outputs, SYNCHRONIZED DATA and VCO CLOCK may be tied to the Encoder/Decoder—which in turn would deseriallize and decode the data before sending it to the controller. $\overline{\text{PREAMBLE DETECTED}}$ output can be tied to the controller and/or the Encoder/Decoder to provide an indication when 4 consecutive bytes of preamble pattern have been detected. The only output that is not shown in *Figure 16* is the PHASE COMPARATOR TEST output. This output is the logical OR of the Phase Comparator's outputs (Charge-Up and Charge-Down inputs of the Charge-Pump). As such, pulses generated at this output provide information about the loop filter's behavior in that the envelope of the pulses generated at this output is a waveform that represents the loop filter's response to any phase difference detected by the Phase Comparator.

Finally, to improve noise immunity, Digital and Analog VCC pins should be tied together and also the Digital and Analog Ground pins should be tied together. PG1 pin should also be grounded.

Applications of the DP8462 Data Synchronizer

The DP8462 is part of National Semiconductor's DP8460 Series Disk Chip Set and therefore, is designed to work in conjunction with other members of this family; such as DP8464—the pulse detector, and DP8466—the disk data controller. A typical system application employing these components is shown in *Figure 17*. The DP8462 is based upon the proven circuitry of the DP8465 (Data synchronizer and separator for the MFM code)—the first integrated circuit to place on one chip a PLL with features that offer the improved speed and reliability required by the disk industry. Not only does the chip simplify disk system design, but also provides fast lock-on to the incoming preamble. Once locked on, the loop is set into a lower bandwidth mode. This inherent loop stability allows for a sizable amount of jitter on the data stream, such as is encountered in many disk systems. Once in the stable tracking rate, the SYNCHRONIZED DATA output represents the incoming ENCODED DATA and is synchronous with VCO CLOCK. This synchronized data is then deserialized by the ENDEC using the VCO CLOCK.

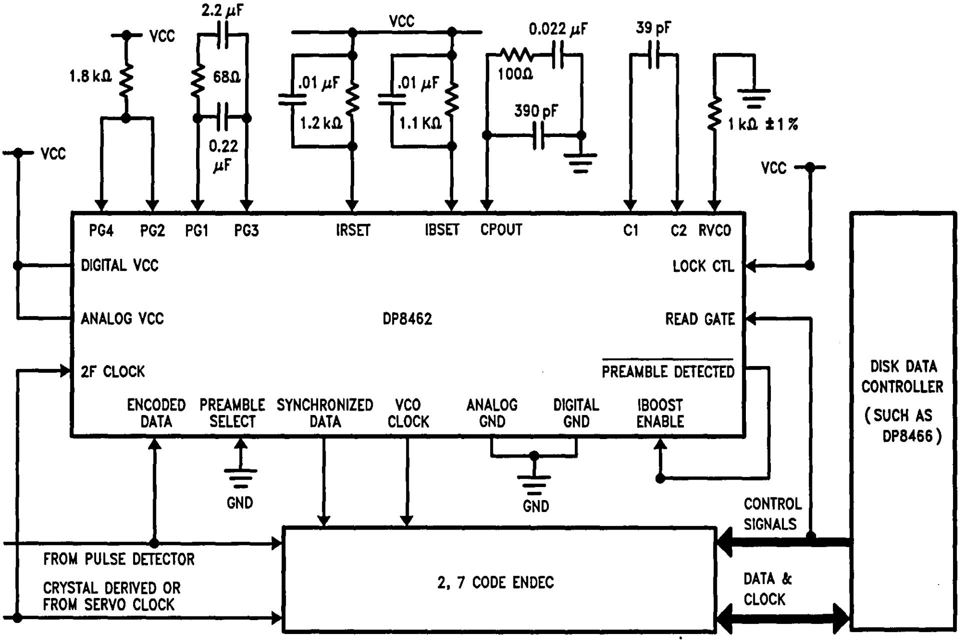
The DP8462 is capable of operating at up to a 20 Mbits/sec data rate and so is compatible with a wide assortment of disk drives. The faster data rates of the 8-inch and 14-inch disk drives will mandate the selection of the DP8462-3 parts with narrower window margin on the incoming data stream. This will also be the case when 5¼-inch drives achieve higher data rates. Some 8-inch and 14-inch disk drives incorporate the functions of the DP8462, but use many discrete ICs. In these cases, replacing these components with the DP8462 will offer reduced P.C. board area, lower cost, and improved performance while simplifying circuit testing.

Most 5¼-inch and many 8-inch and 14-inch disk drives manufactured at present do not incorporate any of the functions of the DP8462. This is so primarily because the PLL function is difficult to design and implement and requires circuitry which covers a large area of the printed circuit card. This is undesirable both from the drive size aspect and from the cost aspect (the cost includes soldering, testing, and adjusting the components). Consequently, most smaller disk drives output RLL encoded data so that the phase-locked-loop and data separation have to be performed by the controller. The DP8462 will therefore replace these functions in controller designs, as shown in *Figure 18a*.

System design criteria may now change because the DP8462 is a one-chip solution, requiring only a few external passive components with fixed values. It operates from a +5V supply, consumes about 0.5W, and is housed in a narrow 24-pin package. The circuitry has been designed so that the external resistors and capacitors need not be adjustable; the user chooses the values according to the disk drive requirements. Once selected, they will be fixed for that particular drive type. These features make it possible to transfer these functions to the disk drive, as shown in *Figure 18b*. Apart from a slight increase in board area, the advantages outweigh the disadvantages. First, the components selected are fixed for each type of drive and this facilitates the problem of interchangeability of drives. At present, components in the controller are adjusted to function with each specific drive; with the DP8462 in the drive, component adjustment will no longer be required. Second, there is often a problem of reliability of data transfer. The incoming data signal is susceptible to noise, bit shift, etc. Soft errors will occur when the incoming disk data bit position is outside the Pulse Gate window as it is being synchronized to the VCO clock in the phase-locked-loop. Obviously, the nearer the PLL is to the data source, the less chance there is that errors will occur. Thus placing the DP8462 in the drive will increase the reliability of data transfer within the system.

Applications of the DP8462 Data Synchronizer (Continued)

A third advantage is data rate upgrading. Most 5¼-inch drives have 5 Mbit/sec data rate because the early drives were made with this data rate. This meant the controllers had to be designed with PLLs which operate at this data rate. It is therefore difficult for drive manufacturers to introduce new drives that are not compatible with existing controllers. Since no new standard data rate has emerged, they must continue to produce drives at this data rate to be compatible with the controllers on the market. With the DP8462 in the drive, and its associated components set for the drive's data rate, it no longer becomes a problem to increase the data rate, assuming the controller's digital circuitry can accommodate the change. This will allow the manufacturers to increase the bit density and therefore the capacity of their drives.



TL/F/8418-19

1) RLL (2,7 Code) Data Input, 10 Mbit/sec Data Rate

2) 1-0-0 Preamble Pattern

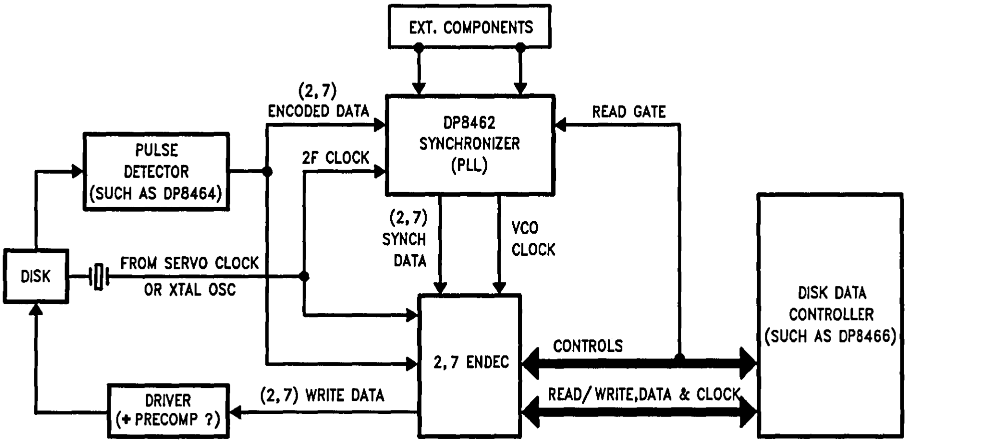
3) PLL to stay in Phase-Frequency Comparison mode until 4 bytes of Preamble Detected

4) PLL to stay in high Track Rate until $\overline{\text{PREAMBLE DETECTED}}$ asserted

5) Delay line left unadjusted (PG2 & PG4 shorted together)

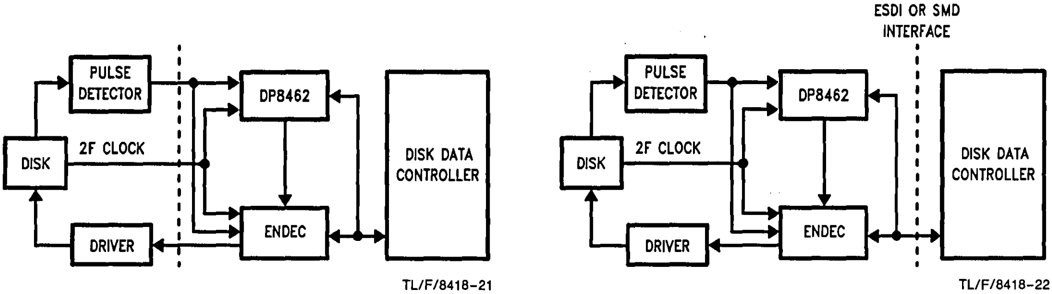
FIGURE 16. Typical Connection to DP8462 For:

Applications of the DP8462 Data Synchronizer (Continued)



TL/F/8418–20

FIGURE 17. Typical Application of DP8462 in a System Employing RLL (2,7) Code



a) DP8462 in the Controller

b) DP8462 in the Drive

FIGURE 18. Two Different Methods of Utilizing DP8462

PRECAUTIONS IN BREADBOARDING AND PCB LAYOUT

The DP8462 contains a high performance analog PLL and certain precautions must be taken when breadboarding or designing a PCB layout. The following guidelines should be adhered to when working with the DP8462:

1) Do not wire wrap.
2) Keep component lead lengths short, place components as close to pins as possible. This applies to R1, C1, R2, C_{VCO}, R_{RATE}, R_{BOOST}, C_{RATE}, C_{BOOST}, RPG1, RPG2, and CPG1.
3) Provide a good ground plane and use a liberal amount of supply bypassing. The quieter a PLL's environment, the happier it is.
4) Avoid routing any digital leads within the vicinity of the analog leads and components.

We have used a PC board approach to breadboarding the DP8462 that gives us an excellent ground plane and keeps component lead lengths very short. With this setup we have found very stable and reliable operation. Illustrations of component layout is shown in *Figure 19*. Note that the board layout is a recommendation not a requirement.

ADDITIONAL NOTES

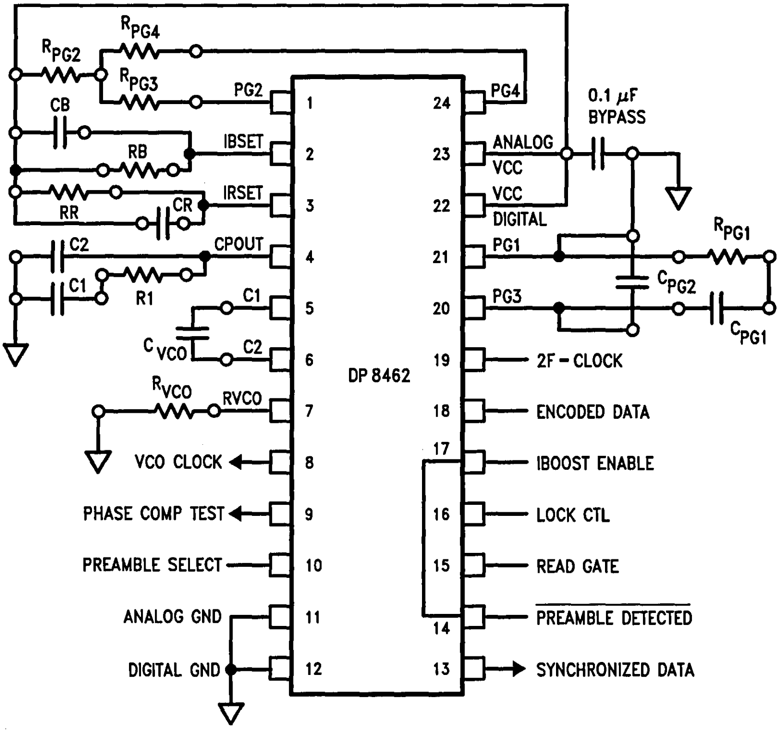
1) PG1 should be grounded to improve noise immunity.
2) 2F clock must be applied at all times; without the 2F clock, the pulse gate circuitry will not operate properly making it impossible to lock onto the incoming data stream.
3) The programming capacitor for the V_{CO} can be calculated as:

$$C_{VCO} = 1/(f_{VCO} * R_{VCO}) - 5\text{ pF}$$

The 5 pF value is due to parasitic internal device capacitance.

4) Care must be taken in final PC board layout to minimize pin to pin capacitance, particularly in multi-layer printed circuit boards.
5) Please refer also to Precaution For Disk Data Separator Designs, NSC Application Note AN-414.

Connection Diagram



TL/F/8418-23

FIGURE 19. Recommended Component Layout

National Semiconductor Corporation

PRELIMINARY

DP8463B (2, 7) ENDEC

General Description

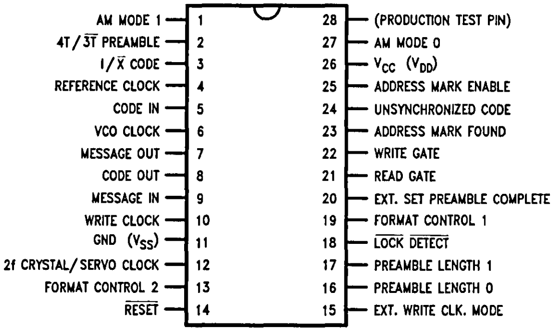
The DP8463B (2, 7) ENDEC performs the encoding and decoding necessary to use either of two different Run-Length Limited (RLL) Codes for disk drive memory systems. Either code gives a disk system the ability to record up to 50% more message data in the same media space without any increase in the Flux Changes per Inch (FCI), when compared to a system using Modified Frequency Modulation (MFM) coding. The DP8463B also performs several other functions to enable many disk controllers, designed for MFM or other codes, to use RLL codes. These added functions are: the writing and reading of an Address Mark for soft-sectored disk formats, and the writing and reading of a Preamble pattern (Phase Locked Loop (PLL) Sync Field) that is compatible with the (2, 7) RLL Code. There are three different Address Marks and two Preamble patterns that may be programmed along with two possible encode/decode tables in seven possible formats for soft sectored disks or three possible formats for hard sectored disks. The user can also select two possible lengths of Preamble to count before issuing a "Lock Detect" signal or an external, user provided, signal may be accepted so any Preamble length may be counted before issuing a Lock Detect signal. The term "Message" is used to designate unencoded data, and the term "Code" is used to designate the encoded data.

Features

- Up to 50% increase in recorded data density over MFM
- Encodes and decodes IBM (2, 7) Code
- Encodes and decodes Xerox (2, 7) Code
- Soft-sector Address Mark generation and detection
- Preamble generation and detection (maximum frequency "3T" = 100 . . . or "4T" = 1000 . . .)
- Programmable formats:
  - Hard sector
  - Soft sector with Address Mark preceding Preamble
  - Soft sector with Address Mark following Preamble
- Programmable Address Mark:
  - SMD 3-Byte gap with no transitions, preceding Preamble
  - IBM 2-Byte gap with two transitions, preceding Preamble
  - Address Mark not violating (2, 7) constraints following Preamble
- Programmable Preamble length counted before "lock-detect" issued:
  - Externally determined
  - 6 Bytes
  - 8 Bytes
- Code output is synchronized to 2f crystal/servo clock
- Glitchless multiplexer switching between VCO clock for reading and 2f clock for writing
- Message Data Rate to 25M bits per second (Code Rate = 50 Mbps)
- TTL Compatible Inputs and Outputs
- Compatible with Phase-Locked-Loop of DP8462 Data Separator
- Compatible with DP8466 Disk Data Controller
- 28-Pin wide Dual-In-Line Package & 28 pin Plastic Chip Carrier
- 2-Micron, Dual Metal CMOS
- Single +5V Supply

Connection Diagrams

Dual-In-Line Package

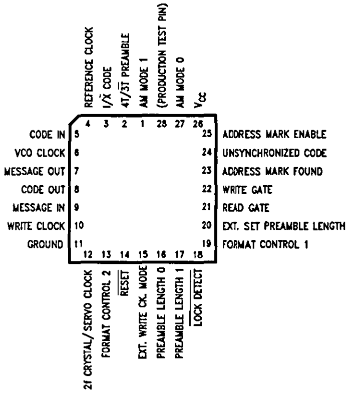


TL/F/8433-1

Top View

28 Pin Package
Order Number DP8463BN
See NS Package N28B

Plastic Chip Carrier



TL/F/8433-2

Top View

28 Pin Package
Order Number DP8463BV
See NS Package V28A

National
Semiconductor
Corporation

DP8464B Disk Pulse Detector

General Description

The DP8464B Disk Pulse Detector utilizes analog and digital circuitry to detect amplitude peaks of the signal received from the read/write amplifier fitted with the heads of disk drives. The DP8464B produces a TTL compatible output which, on the positive leading edge, indicates a signal peak. Electrically, these peaks correspond to flux reversals on the magnetic medium. The signal from the read/write amplifier when reading a disk is therefore a series of pulses with alternating polarity. The Disk Pulse Detector accurately replicates the time position of these peaks.

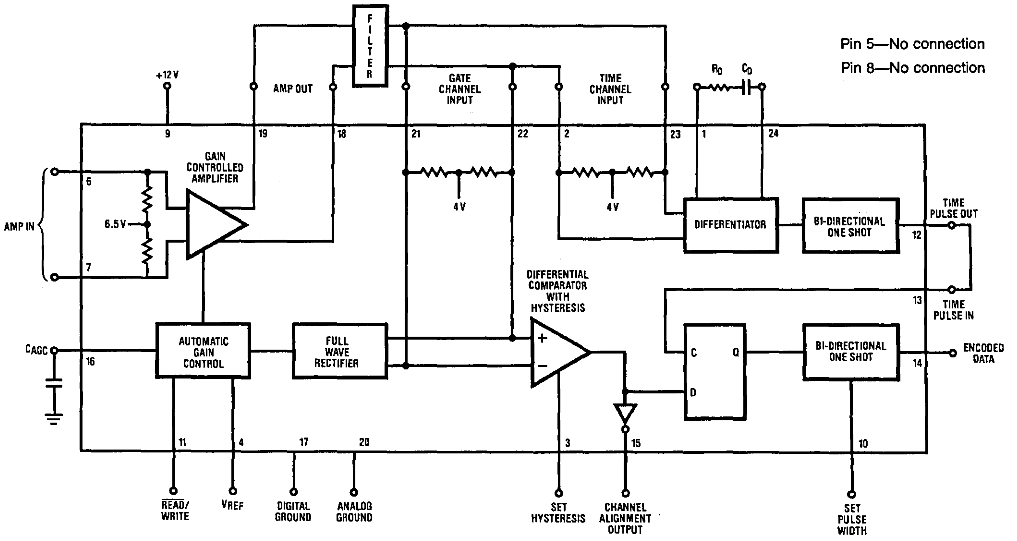
The DP8464B Disk Pulse Detector has three main sections: the Amplifier, the time channel and the gate channel. The Amplifier section consists of a wide bandwidth amplifier, a full wave rectifier and Automatic Gain Control (AGC). The time channel is made from the differentiator and its following bi-directional one shot, while the gate channel is made from the differential comparator with hysteresis, the D flip-flop and its following bi-directional one shot.

The Disk Pulse Detector is fabricated using an advanced oxide isolated Schottky process, and has been designed to function with data rates up to 15 Megabits/second. The DP8464B is available in either a 300 mil wide 24-pin dual-in-line package or a surface mount 28-pin plastic chip carrier package. Normally, it will be fitted in the disk drive, and its output may be directly connected to the DP8461 or the DP8465 Data Separator.

Features

- Wide input signal amplitude range—from 20 mVpp to 660 mVpp differential
- Data rates up to 15 Megabits/sec 2,7 code
- On-chip differential gain controlled amplifier, differentiator, comparator gating circuitry, and output pulse generator
- Input capacitively coupled directly from the disk head read/write amplifier
- Adjustable comparator hysteresis
- AGC and differentiator time constants set by external components
- TTL compatible digital Inputs and Outputs
- Encoded Data Output may connect directly to the DP8461 or DP8465 Data Separator
- Standard drive supply: 12V ± 10%
- Available in 300 mil wide 24-pin dual-in-line package or a surface mount 28-pin plastic chip carrier package

Block Diagram



Pin 5—No connection
Pin 8—No connection

TL/F/5283-7

Note: All pin numbers in this data sheet refer to the 24-pin dual-in-line package.

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

	Pins	Limit
Supply Voltage	9	14V
TTL Input Voltage	11,13	5.5V
TTL Output Voltage	12,14,15	5.5V
Input Voltage	3,4	5.5V
Minimum Input Voltage	3,4	−0.5V
Differential Input Voltage	6-7, 21-22, 2-23	3V or −3V

Storage Temperature	−65°C to +150°C
Lead Temp. (Soldering, 10 seconds)	300°C
Maximum Power Dissipation at 25°C	
Molded DIP Package (derate 15.6 mW/°C above 25°C)	1950 mW
Plastic Chip Carrier Package (derate 12.5 mW/°C above 25°C)	1560 mW

Operating Conditions

Symbol	Parameter	Min	Typ	Max	Units
V_{CC}	Supply Voltage	10.8	12.0	13.2	V
T_A	Ambient Temperature	0		70	°C

DC Electrical Characteristics

Over Recommended Operating Temperature and Supply Range V_{REF} = 0.5V, Set Hysteresis = 0.3V. $\overline{\text{Read}}$/Write = 0.3V unless otherwise noted. All Pin Numbers Refer to 24 Pin Dual-In-Line Package.

Symbol	Pins	Parameter	Conditions	Min	Typ	Max	Units
AMPLIFIER							
$Z_{IN_{AI}}$	6,7	Amp In Impedance	T_A = 25°C (Note 1)	0.8	1.0	1.2	kΩ
A_{VMIN}	18,19	Min Voltage Gain	AC Output 4 Vpp Differential			6.0	V/V
A_{VMAX}	18,19	Max Voltage Gain	AC Output 4 Vpp Differential	200			V/V
$V_{C_{AGC}}$	16	Voltage on C_{AGC}	A_V = 6.0 A_V = 200	 2.8	4.5 3.7	5.5 	V V
GATE CHANNEL							
$Z_{IN_{GCI}}$	21,22	Gate Channel Input Impedance	T_A = 25°C (Note 1)	1.75	2.5	3.25	kΩ
$I_{C_{AGC}-}$	16	Pin 16 Current which Charges C_{AGC}	$V_{PIN\ 16}$ = 3.9V $\lvert V_{PIN\ 21} - V_{PIN\ 22}\rvert$ = 2.6V	−1.5	−2.5	−3.5	mA
$I_{C_{AGC}+}$	16	Pin 16 Current which Discharges C_{AGC}	$V_{PIN\ 16}$ = 5V $\lvert V_{PIN\ 21} - V_{PIN\ 22}\rvert$ = 1.4V		1	5	µA
$I_{V_{REF}}$	4	V_{REF} Input Bias Current			−20	−100	µA
$V_{TH_{AGC}}$	22,21 4,16	AGC Threshold	(Note 2) $V_{PIN\ 16}$ = 4.2V	0.88	1.0	1.12	V
I_{SH}	3	Set Hysteresis Input Bias Current			−60	−100	µA
$V_{TH_{SH}}$	22,21 3,15	Set Hysteresis Threshold	(Note 3)	0.48	0.6	0.72	V
TIME CHANNEL							
$Z_{IN_{TC}}$	2,23	Time Channel Input Impedance	T_A = 25°C (Note 1)	3.5	5.0	6.5	kΩ
I_{C_d}	24	Current into Pin 1 and 24 that Discharges C_d		1.4	1.8	2.2	mA

DC Electrical Characteristics

Over Recommended Operating Temperature and Supply Range V_{REF} = 0.5V, Set Hysteresis = 0.3V. $\overline{Read}$/Write = 0.3V unless otherwise noted. All pin numbers refer to the 24 pin dual-in-line package. (Continued)

Symbol	Pins	Parameter	Conditions	Min	Typ	Max	Units
WRITE MODE							
$Z_{IN_{AI}}$	6,7	Amp In Impedance in Write Mode	$V_{PIN\ 11}$ = 2.0V	100		500	Ω
I_{CAGC-}	16	Pin 16 Current in Write Mode	$V_{PIN\ 11}$ = 2.0V $V_{PIN\ 16}$ = 3.9V $\lvert V_{PIN\ 21} - V_{PIN\ 22} \rvert$ = 1.3V		1	5	μA
DIGITAL PINS							
V_{IH}	11,13	High Level Input Voltage		2			V
V_{IL}	11,13	Low Level Input Voltage				0.8	V
V_I	11,13	Input Clamp Voltage	V_{CC} = Min I_I = −18 mA			−1.5	V
I_{IH}	11,13	High Level Input Current	V_{CC} = Max V_I = 2.7V			20	μA
I_I	11,13	Input Current at Maximum Input Voltage	V_{CC} = Max V_I = 5.5V			1	mA
I_{IL}	11,13	Low Level Input Current	V_{CC} = Max V_I = 0.5V			−200	μA
V_{OH}	12,14, 15	High Level Output Voltage	V_{CC} = Min I_{OH} = −40 μA (Note 4)	2.7			V
V_{OL}	12,14, 15	Low Level Output Voltage	V_{CC} = Min I_{OL} = 800 μA (Note 4)			0.5	V
I_{OS}	12,14, 15	Output Short Circuit Current	V_{CC} = Max V_O = OV			−100	mA
I_{CC}	9	Supply Current	V_{CC} = Max		54	75	mA

AC Electrical Characteristics

Over Recommended Operating Temperature and Supply Range

Symbol	Pins	Parameter	Conditions	Typ	Max	Units
DP8464B-2 t_{pp}	14	Pulse Pairing	f = 2.5 MHz V_{IN} = 40 mVpp differential (Note 5)	±1.5	±3	ns
DP8464B-3 t_{pp}	14	Pulse Pairing	f = 2.5 MHz V_{IN} = 40 mVpp differential (Note 5)	±2	±5	ns

Note 1: The temperature coefficient of the input impedance is typically 0.05% per degree C.

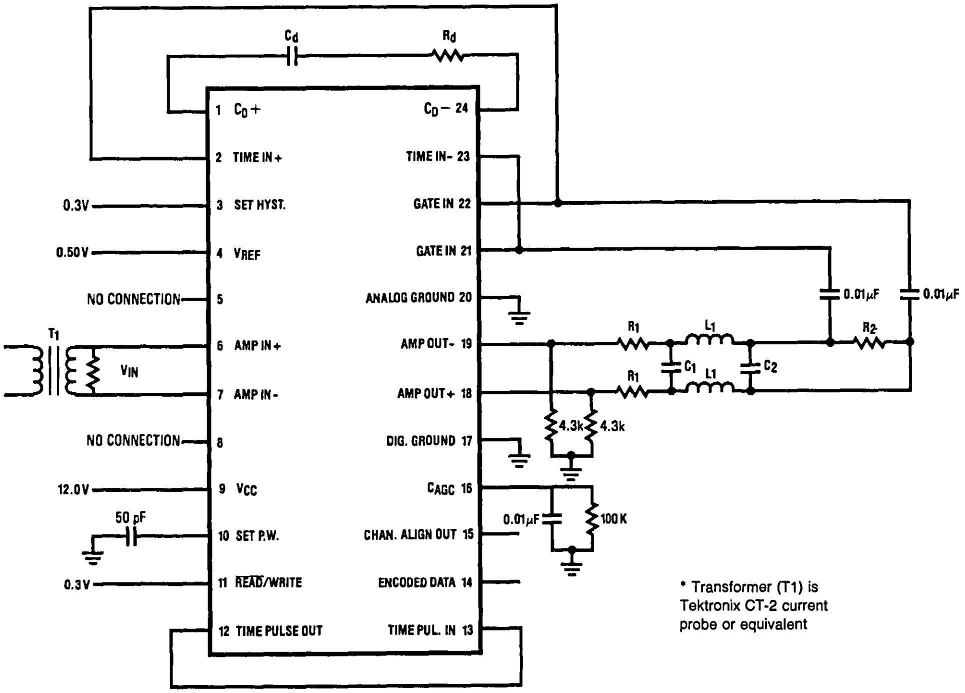
Note 2: The AGC Threshold is defined as the voltage across the Gate Channel Input (pins 21 and 22) when the voltage on C_{AGC} (pin 16) is 4.2V.

Note 3: The Set Hysteresis Threshold is defined as the minimum differential AC signal across the Gate Channel Input (pins 21 and 22) which causes the voltage on the Channel Alignment Output (pin 15) to change state.

Note 4: To prevent inductive coupling from the digital outputs to Amp In, the TTL outputs should not drive more than one ALS TTL load each.

Note 5: The filter and the differentiator network are described in the Pulse Pairing Set Up.

Pulse Pairing Set Up



* Transformer (T1) is Tektronix CT-2 current probe or equivalent

TL/F/5283-3

Differentiator network

Cd = 50 pF   Rd = 430Ω

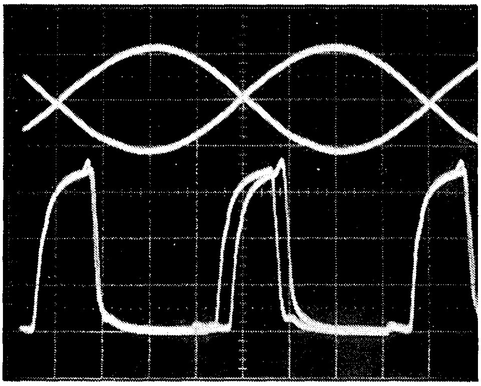
Filter

R1 = 240Ω   R2 = 680Ω

C1 = 15 pF   C2 = 100 pF

L1 = 4.7 μH

This is a 3 pole Bessel with the corner frequency at 7.5 MHz which is similar to filters used in 10 Mbits/sec MFM drives.



TL/F/5283-4

Pulse Pairing Measurement

Connect a scope probe to pin 14 (Encoded Data Out) and trigger off its positive edge. Adjust the trigger holdoff so the scope first triggers off the pulse associated with the positive peak and then off the pulse associated with the negative peak (as shown in the scope photo below). Pulse pairing is displayed on the second pair of pulses on the display. If the second pulses are separated by 10 ns, then the pulse pairing for this part is ±5 ns.

Circuit Operation

The output from the read/write amplifier is AC coupled to the Amp Input of the DP8464B. The amplifier's output voltage is fed back via an external filter to an internal fullwave rectifier and compared against the external voltage on the V$_{REF}$ pin. The AGC circuit adjusts the gain of the amplifier to make the peak to peak differential voltage on the Gate Channel Input four times the DC voltage on V$_{REF}$. Typically the signal on Amp Out will be set for 4 Vpp differential. Since the filter usually has a 6 dB loss, the signal on the Gate Channel Input will be 2 Vpp differential. The user should therefore set 0.5V on V$_{REF}$ which can be done with a simple voltage divider from the +12V supply.

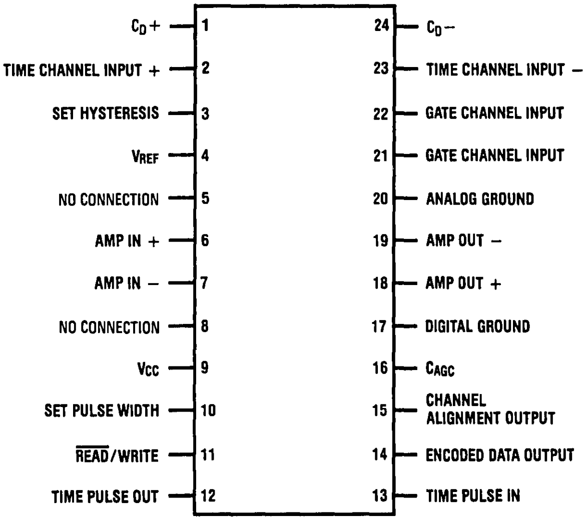
The peak detection is performed by feeding the output of the Amplifier through an external filter to the differentiator. The differentiator output changes state when the input pulse changes direction, generally this will be at the peaks. However, if the signal exhibits shouldering (the tendency to return to the baseline), the differentiator will also respond to noise near the baseline. To avoid this problem, the signal is also fed to a gating channel which is used to define a level either side of the baseline. This gating channel is comprised

Circuit Operation (Continued)

of a differential comparator with hysteresis and a D flip-flop. The hysteresis for this comparator is externally set via the Set Hysteresis pin. In order to have data out, the input amplitude must first cross the hysteresis level which will change the logic level on the D input of the flip-flop. The peak of the input signal will generate a pulse out of the differentiator and bi-directional one shot. This pulse will clock the new data at the D input through to the output. In this way, when the differentiator is responding to noise at the baseline, the output of the D flop is not changing since the logic level into the D input has not changed. The comparator circuitry is therefore a gating channel which prevents any noise near the baseline from contaminating the data. The amount of hysteresis is twice the DC voltage on the Set Hysteresis pin. For instance, if the voltage on the Set Hysteresis pin is 0.3V, the differential AC signal across the Gate Channel Input must be larger than 0.6V before the output of the comparator will change states. In this case, the hysteresis is 30% of a 2V peak to peak differential signal at the gate channel input.

Connection Diagrams

Dual-In-Line (DIP) Package

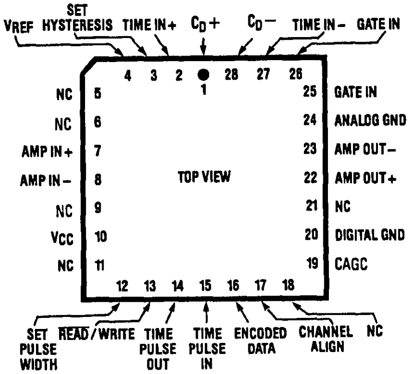


Top View

TL/F/5283-2

Order Number DP8464BN-3 or DP8464BN-4
See NS Package N24C

Plastic Chip Carrier (PCC) Package



TL/F/5283-30

Order Number DP8464BV-3 or DP8464BV-4
See NS Package V28A

Pin Definitions

(All pin numbers refer to the 24 pin dual-in-line package)

Pin#	Name	Function
Power Supply		
9	V_{CC}	The supply is +12V ± 10%.
17	Digital Ground	Digital signals should be referenced to this pin.
20	Analog Ground	Analog signals should be referenced to this pin.
Analog Signals		
6 7	Amp In+ Amp In−	These are the differential inputs to the Amplifier. The output of the read/write head amplifier should be capacitively coupled to these pins.
18 19	Amp Out+ Amp Out−	These are the differential outputs of the Amplifier. These outputs should be capacitively coupled to the gating channel filter (if required) and to the time channel filter.
22 21	Gate Channel Inputs	These are the differential inputs to the AGC block and the gating channel. These inputs must be capacitively coupled from the Amp Out.
2 23	Time Channel Input+ Time Channel Input−	These are the differential inputs to the differentiator in the time channel. In most applications, a filter between the Amp Out (pins 18 and 19) and these inputs is required to band limit the noise and to correct for any phase distortion introduced by the read circuitry. In all cases this input must be capacitively coupled to prevent disturbing the DC input level.
1 24	C_d+ C_d−	The external differentiator network is connected between these two pins.
3	Set Hysteresis	The DC voltage on this pin sets the amount of hysteresis on the differential comparator. Typically this voltage can be established by a simple resistive divider from the positive supply.
4	V_{REF}	The AGC circuit adjusts the gain of the amplifier to make the differential peak to peak voltage on the Gate Channel Input equal to four times the DC voltage on this pin. This voltage can be established by a simple resistive divider from the positive supply.
5 8	No connection No connection	
16	C_{AGC}	The external capacitor for the AGC is connected between this pin and Analog Ground.
Digital Signals		
10	Set Pulse Width	An external capacitor to control the pulse width of the Encoded Data Out is connected between this pin and Digital Ground.
11	$\overline{\text{Read}}$/Write	If this pin is low, the Pulse Detector is in the read mode and the chip is active. When this pin goes high, the pulse detector is forced into a standby mode. This is a standard TTL input.
12	Time Pulse Out	This is the TTL output from the bi-directional one shot following the differentiator. In most applications this can be connected directly to the Time Pulse In.
13	Time Pulse In	This is the TTL input to the clock of the D flip-flop. Usually this is connected directly to the Time Pulse Out pin.
15	Channel Alignment	This is the buffered output of the differential comparator with hysteresis. This is usually used in the initial system design and is not used in production.
14	Encoded Data Out	This is the standard TTL output whose leading edge, indicates the time position of the peaks.

Application Information

GENERAL DESCRIPTION

All pin numbers refer to 24 pin dual-in-line package.

The DP8464B Disk Pulse Detector utilizes analog and digital circuitry to detect amplitude peaks of the signal received from the Read/Write Amplifier. The analog signal from a disk is a series of pulses, the peaks of which correspond to 1's or flux reversals on the magnetic medium. The pulse detector must accurately determine the time position of these peaks. The peaks are indicated by the positive leading edge of a TTL compatible output pulse. This task is complicated by variable pulse amplitudes depending on the media type, head position, head type and read/write amplifier circuit gain. Additionally, as the bit density on the disk increases, the amplitude decreases and significant bit interaction occurs resulting in pulse distortion and shifting of the peaks.

The graph in *Figure 1* shows how the pulse amplitude varies with the number of flux reversals per inch (or recording density) for a given head disk system. The predominant disk applications are associated with the first two regions on this graph, Regions 1 and 2. Typical waveforms received by the pulse detector for these regions are shown next to the graph.

Application Information (Continued)

Region 1 is the high resolution area characterized by a large spread between flux reversals and a definite return to baseline (no signal) between these peaks. Pulses of this type are predominantly found in drives which use either thin film heads or plated media, or in drives which utilize run length limited codes (like the 2,7 code) which spread the distance between flux reversals.

A Region 2 waveform will vary from a tendency to return to the baseline (called shouldering) to almost sinusoidal at the higher frequencies. These pulses come from drives which use limited frequency codes (such as MFM). The pulses may contain shouldering on the outer tracks of the disk and be nearly sinusoidal on the inner tracks since the flux density increases towards the inner track.

Detecting pulse peaks of waveforms of such variable characteristics requires a means of separating both noise and shouldering-caused errors from the true peaks. In the past, mild shoulder-caused errors were blocked by self-gating circuits (such as the "de-snaker"). These circuits fail when shouldering is extensive, hence the need for the DP8464B which includes a peak sensing circuit and an amplitude sensitive gating channel in parallel.

The main circuit blocks of the DP8464B are shown in *Figure 2*. The output from the read/write amplifier is fed directly to the Amp Input of the DP8464B. This is the input of a Gain Controlled Amplifier. The amplifier's output voltage is fed back via an external filter to an internal fullwave rectifier and compared against the external voltage on the V_{REF} pin. The AGC circuit adjusts the gain of the amplifier to make the peak-to-peak differential Gate Channel input voltage four times the DC voltage on V_{REF}.

The peak detection is performed by feeding the output of the Gain Controlled Amplifier through an external filter to the differentiator. The differentiator output changes state when the input pulse changes direction, generally this will be at the peaks. However, if the signal exhibits shouldering (the tendency to return to the baseline) as seen in Region 1 and the upper part of Region 2, the differentiator will also respond to noise near the baseline. To avoid this, the signal is also fed to a gating channel which is used to define a level either side of the baseline. This gating channel comprises a differential comparator with hysteresis and a D flip-flop. The hysteresis for this comparator is externally set via the Set Hysteresis pin. In order to have valid data out, the input amplitude must first cross the hysteresis level. This will change the logic level on the D input of the flip-flop. The peak of the input signal will generate a pulse out of the differentiator and bi-directional one shot. This pulse will clock in the new data on the D input, which will appear at the Q output. In this way, when the differentiator is responding to noise at the baseline, the output of the D flop is not changing since the logic level into the D input has not yet changed. The comparator circuitry is therefore a gating channel to prevent any noise near the baseline from contaminating the data.

The amount of hysteresis is twice the DC voltage on the Set Hysteresis pin. For instance, if the voltage on the Set Hysteresis pin is 0.3V, the differential Gate Channel Input must be larger than 0.6V (±0.3V) before the output of the comparator will change states. The Time Pulse Out, Encoded Data, and Channel Alignment Output are designed to drive 1 standard TTL gate.

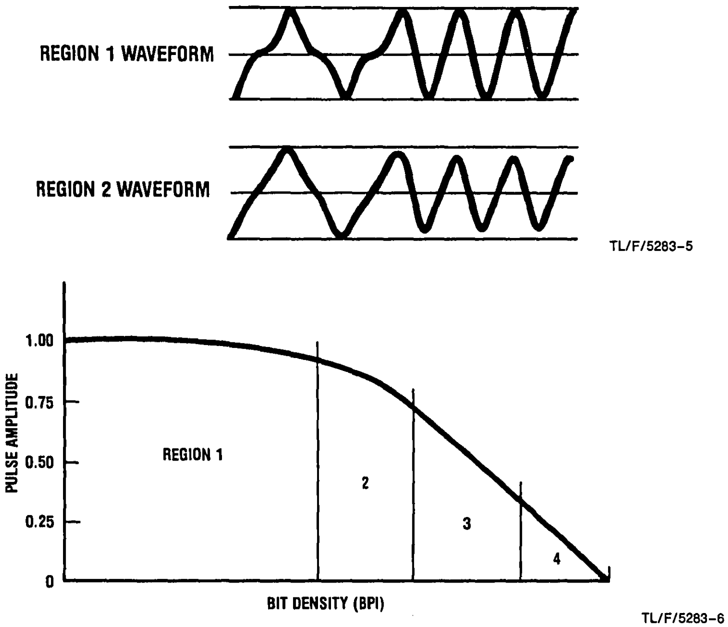
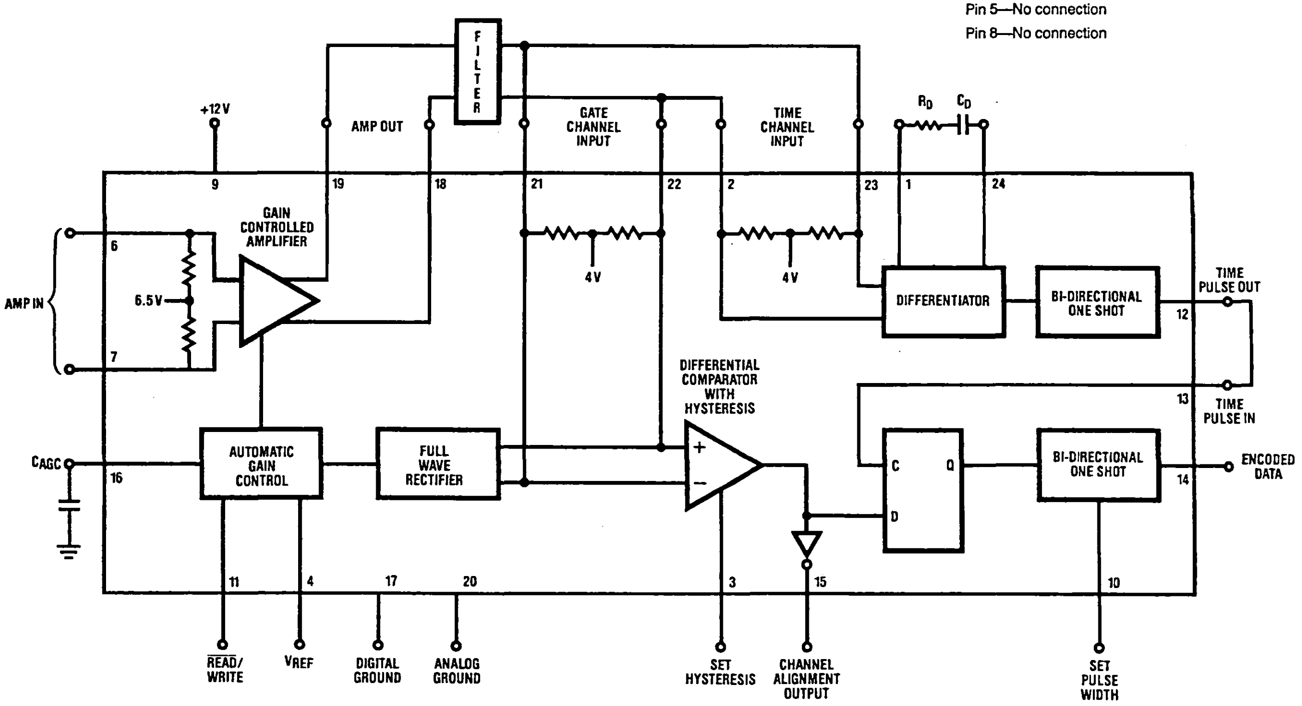


FIGURE 1. Pulse Amplitude vs. Bit Density with Typical Waveforms



Pin 5—No connection

Pin 8—No connection

TL/F/5283-7

FIGURE 2. DP8464B Block Diagram, Region 1 Connection

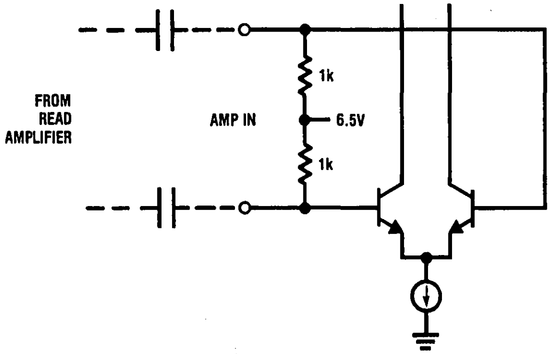
Application Information (Continued)

GAIN CONTROLLED AMPLIFIER

The purpose of the Gain Controlled Amplifier is to increase the differential input signal to a fixed amplitude while maintaining the exact shape of the input waveform. The Gain Controlled Amplifier is designed to accept input signals from 20 mVpp to 660 mVpp differential and amplify that signal to 4 Vpp differential. The gain is therefore from 6 to 200 and is controlled by the automatic gain control (AGC) loop. The amplifier output is actually capable of delivering typically 5 Vpp differential output but the parts are only tested and guaranteed to 4 Vpp.

The input to the Gain Controlled Amplifier is shown in *Figure 3*. The value of the input capacitors should be selected so that the pole formed by the coupling capacitor and the 1k bias resistor is a factor of 10 lower than the lowest signal frequency. These input bias resistors have a ±20% tolerance and a temperature coefficient of 0.05% per degree C. When the pulse detector is in the write mode, these bias resistors are automatically shunted by 425Ω resistors. This allows the input circuit to recover quickly from the large transients encountered during a write to read transition. The input impedance to the amplifier is therefore 1k during read operations and 300Ω during write operations.

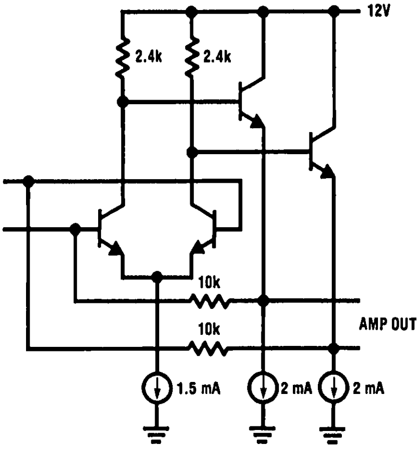
The output of the Gain Controlled Amplifier is shown in *Figure 4*. The outputs are biased at (12V − (0.75 mA × 2.4k) − 0.75V) or 9.5V. Since each output will swing ±1V (4 Vpp differential), each output pin will swing from 8.5V to 10.5V. If the total differential load placed on the output is 1k, (see *Figure 5*) then the circuit must supply 2V/1k or 2 mA. Since the output is class A, external resistors to ground must be used to provide the sink current. In this case, in order to sink 2 mA at the lowest voltage, then (8.5V/2 mA) or an external 4.3k resistor from each output to ground is required. Note that the circuit has additional margin since the internal 2 mA current sources were not included in the calculation. Typically the output impedance of the Gain Controlled Amplifier is 17Ω, and the −3 dB bandwidth is greater than 20 MHz.



TL/F/5283-8

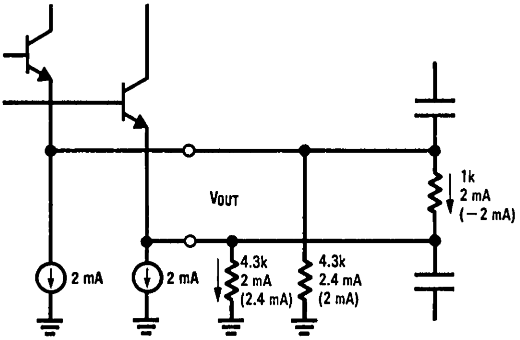
FIGURE 3. Input to Gain Controlled Amplifier

Application Information (Continued)



TL/F/5283-9

FIGURE 4. Output of Gain Controlled Amplifier



TL/F/5283-10

FIGURE 5. Output Stage with 1k Differential Load

Application Information (Continued)

AUTOMATIC GAIN CONTROL (AGC)

The Automatic Gain Control holds the signal level at the Gate Channel Input at a constant level by controlling the gain of the Gain Controlled Amplifier. This is necessary because the amplitude of the input signal will vary with track location, variations in the magnetic film, and differences in the actual recording amplitude. The Gain Controlled Amplifier is designed for a maximum 4 Vpp differential output. To prevent the Gain Controlled Amplifier from saturating, the V_{REF} level must be set so the maximum amplifier output voltage is 4 Vpp. The AGC will force the differential peak-to-peak signal on the Gate Channel Input to be four times the voltage applied to the V_{REF} pin. Normally some kind of filter is connected between the Gain Controlled Amplifier's output and the Gate Channel Input. Typically this filter has a 6 dB insertion loss in its pass band. Since the AGC holds the amplitude at the Gate Channel Input constant, this 6 dB loss through the Gate Channel filter will cause the Gain Controlled Amplifier's output to be 6 dB larger than the Gate Channel Input.

The AGC loop starts out in the high gain mode. When the input signal is larger than expected, the AGC loop will quickly reduce the amplifier gain so the peak-to-peak differential voltage on the Gate Channel Input remains four times the voltage on V_{REF}. If the input amplitude suddenly drops, the AGC loop will slowly increase the amplifier gain until the differential peak-to-peak Gate Channel Input voltage again reaches four times V_{REF}. The AGC loop requires several peaks to react to an increased input signal. In order to recover the exact peak timing during this transition, the V_{OUT} level must be set somewhat lower than the maximum of 4 Vpp. For instance, if the V_{REF} is 0.5V, and if the loss in the gate channel filter is 6 dB, then the Amp Output is 4 Vpp. If the Amp Input suddenly increases 30%, the amplifier may saturate and the timing for a few peaks may be disturbed until the AGC reduces the amplifier gain. If the peak detection is critical during this time, the system may fail. The proper operation, for this example, is to set the V_{REF} at 0.35V so the amplifier will not saturate if the input suddenly increases 30%.

A simplified circuit of the AGC block is shown in *Figure 6*. When the full wave rectified signal from the Gate Channel Input is greater than V_{REF}, the voltage on the collector of transistor T1 will increase and charge up the external capacitor C_{AGC} through T2. The typical available charging current is 2.5 mA. Conversely, if this input is less than V_{REF}, transistor T2 will be off, so the capacitor C_{AGC} will be discharged by the base current going into the Darlington T3 and T4. This discharge current is approximately 1 μA. The voltage across C_{AGC} controls the gain of the Gain Controlled Amplifier. This voltage will vary from typically 3.4V at the highest gain to 4.5V at the lowest gain.

When the AGC circuit has not received an input signal for a long time, the base current of the Darlington will discharge the external C_{AGC} to 3.4V. The amplifier will now be at its highest gain. When a large signal comes in, the external C_{AGC} will be charged up with the 2.4 mA from T2 thereby reducing the gain of the amplifier. The formula, $I = C \times (dV/dt)$ can be used to calculate the time required for the amplifier to go from a gain of 200 to a gain of 6. For instance, if $C_{AGC} = 0.01$ μf, the charging current I is 2.4 mA, and the dV required for the amplifier to go through its gain range is 1.1V, then

$$dt = (0.01\ \mu F \times 1.1V)/(2.4\ mA) \text{ or } 4.6\ \mu s.$$

In reality, the gain does not change this quickly since the C_{AGC} would only be charging during a portion of the input waveform.

By using the same argument, the time required to increase the amplifier gain after the input has been suddenly reduced can be calculated. This time, the discharging current is only 1 μa so

$$dt = (0.01\ \mu F \times 1.1V)/1\ \mu A) \text{ or } 11\ ms.$$

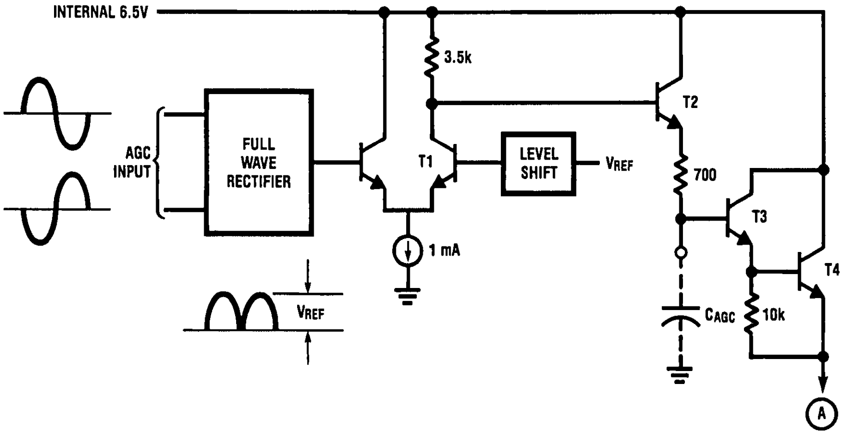


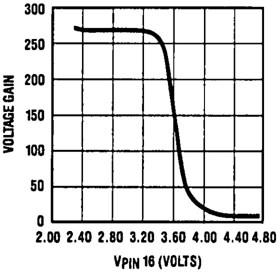
FIGURE 6. Simplified AGC Circuit

Application Information (Continued)

This time can be decreased by placing an external resistor across the C_{AGC}. For instance, if a 100k resistor is placed in parallel with C_{AGC}, then the discharge current is 40 μA. The time required to increase the amplifier gain is now 40 times faster or 275 μs. If this external resistor is made even smaller, say 10k, then the discharge time will go to 27.5 μs. Now however, there is another problem introduced. The response time of the AGC is so fast that it distorts the signal at the output of the Gain Controlled Amplifier. Distortion of the signal at the Amplifier Output can affect the time position of the peaks of this signal. Be sure to check this distortion over the range of input levels you expect to encounter, when choosing the external R and C values for the AGC.

If the value of the bleed resistor across the C_{AGC} is decreased (in order to equalize the AGC attack and decay times) the value of C_{AGC} must be increased in order to maintain an AGC response that does not distort the signal. There is a second order effect on the amplitude that results from this attack and decay time equalization. Referring to *Figure 2*, notice that the AGC is driven from a full wave rectified version of the Gate Channel Input signal. When the AGC is operated normally (ie. fast attack and slow decay) the voltage that appears across C_{AGC} is the peak detected value of this full wave rectified waveform. However, if you equalize the AGC attack and decay times the voltage across C_{AGC} is the RMS voltage (0.707 times the peak) of the full wave rectified waveform. Thus, the voltage across C_{AGC} is less and the amplitude out of the Gain Controlled Amplifier will consequently be 1.4 times larger.

It is possible to externally drive the C_{AGC} pin to control the gain of the amplifier. It must be noted that the gain of the amplifier is not always exactly 200 when the voltage on C_{AGC} is 3.4V. The transfer curve between the gain of the amplifier and the voltage on C_{AGC} is only approximate. This transfer curve will vary between parts and with temperature. Care should be taken to prevent the voltage on the C_{AGC} pin from going below ground or above 5.5V. *Figure 7* shows a typical curve of the Gain Controlled Amplifier Gain vs. the voltage across C_{AGC} (Vpin 16.)

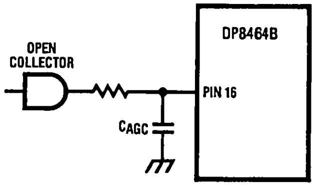


TL/F/5283-12

FIGURE 7. Gain Controlled Amplifier Gain vs. Vpin 16

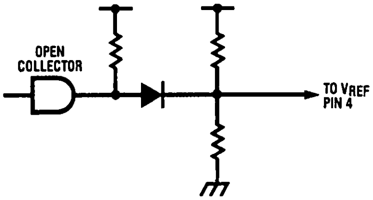
It is possible to change the time constant of the AGC circuit by switching in different external components at the desired times. For instance, as shown in *Figure 8*, an external open collector TTL gate and resistor can be added in parallel with C_{AGC} to decrease the AGC response time. Similarly, an external capacitor could be switched in to increase the response time. Since in the absence of an external resistor the discharge time of C_{AGC} is much longer than the attack time there may be some applications where it is desirable to switch in a parallel resistor to quickly discharge C_{AGC} then switch it out to force a quick attack. Because of the quick attack time, the AGC obtains the proper level quicker than it would had C_{AGC} simply been allowed to discharge to the new level.

There are some applications where it is desirable to hold the AGC level for a period of time. This can be done by raising the $\overline{\text{READ}}$/WRITE pin. This will shut off the input circuitry, and it will take time (about 2.5 μs) for the circuit to recover when going back into the read mode. *Figure 9* shows a method to hold the AGC level while remaining in the read mode (which could be used in embedded servo applications). If the voltage on V_{REF} is raised to 3V, then the amplifier output voltage cannot get large enough to turn on the circuitry to charge up C_{AGC}. For this to work properly, there can not be a large discharge current path (resistor in parallel with C_{AGC}) across C_{AGC}. The AGC block can be bypassed altogether by connecting V_{REF} to 3V. In this way, the user can use his own AGC circuit to drive the C_{AGC} pin directly.



TL/F/5283-13

FIGURE 8. Circuit to Decrease AGC Response Time



TL/F/5283-14

FIGURE 9. Circuit for AGC Hold

$\overline{\text{READ}}$/WRITE

In the normal read mode, the signal from the read/write head amplifier is in the range of 20 mVpp to 660 mVpp. However, when data is being written to the disk, the signal coming into the analog input of the pulse detector will be on the order of 600 mV. Such a large signal will disturb the AGC level and would probably saturate the amplifier. In addition, if a different read/write amplifier is selected, there will be a transient introduced because the offset of the preamplifiers are not matched. A $\overline{\text{READ}}$/WRITE input pin has been provided to minimize these effects to the pulse detector. This is a standard TTL input.

When the $\overline{\text{READ}}$/WRITE pin is low, the pulse detector is in the read mode. When the $\overline{\text{READ}}$/WRITE pin is taken high, three things happen. First, the 1k resistors across the AMP IN pins are shunted by 300Ω resistors, as described previously in the Gain Controlled Amplifier section. Next, the amplifier is squelched so there is no signal on the Amp Output.

Application Information (Continued)

Finally, the previous AGC level is held. This AGC hold function is accomplished by not allowing any current to charge up the external C_{AGC}. The voltage across this capacitor will slowly reduce due to the bias current into the Darlington (see *Figure 6*) or through any resistor placed in parallel with C_{AGC}. Therefore, as described in the Automatic Gain Control section, the gain of the amplifier will slowly increase. All of these three events happen simultaneously.

When the $\overline{\text{READ}}$/WRITE input is returned low, the pulse detector will go back to the read mode in a specific sequence. First of all, the input impedance at the Amp In is returned to 1k. Then, after approximately 1 μs, the Gain Controlled Amplifier is taken out of the squelch mode, and finally approximately 1 μs after that, the AGC circuit is turned back on. This return to the read mode is designed to minimize analog transients in order to provide stable operation after 2.5 μs. It is very important that the analog input be stable before the chip is returned to the read mode. It is recommended that other than when writing, the Pulse Detector be in the read mode at all times in order to prevent the 2.5 μs delay from slowing up the system. The $\overline{\text{READ}}$/WRITE pin may be connected to the Write Gate output of a controller (such as the DP8466 Disk Data Controller).

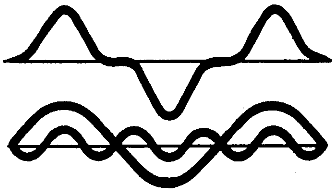
TIME CHANNEL FILTER

The peak detection is performed by feeding the output of the Gain Controlled Amplifier through an external filter to the differentiator. The differentiator output changes state when the input pulse changes direction, generally this will be at the peaks. The differentiator can also respond to noise near the baseline, in which case the comparator gating channel will inhibit the output. The purpose of the external filter is to bandwidth limit the incoming signal for noise considerations. Care must be used in the design of this filter to ensure the delay is not a function of frequency. For this reason, a high order Bessel filter with its constant group delay characteristics can be used in this application. Often, this filter must be specifically designed to correct errors introduced by the non-ideal phase characteristics of the input read head. The typcial −3 dB point for this filter is around 1.5 times the highest recorded frequency. The design of this filter is complex and will not be discussed here. However, the following discussion does give a feel for some of the considerations involved in the filter design. The reader is referred to reference #3 listed at the end of the Applications Notes for further filter design information.

Figure 10 shows a typical Region 1 waveform where there is no bit interaction. This waveform is primarily the sum of the fundamental frequency and its 3rd harmonic (higher odd harmonics are present when there is more shouldering).

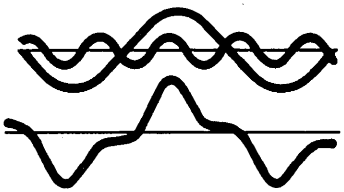
If the filter is to preserve this wave shape (this would be the case if no read/write head phase compensation were necessary) then the phase relationship between the fundamental frequency and its harmonics must not be altered. *Figure 11* shows the output when the 3rd harmonic has the proper magnitude, but the phase relationship is not maintained. The result is that the output waveform is not the same shape as the input (in a severe case it may be almost unrecognizable) and the time position of the peaks has been altered.

One electrical parameter which describes how well a filter will preserve a wave shape is called group delay. Group delay is defined as the change in phase divided by the change in frequency. If the group delay is constant over the frequencies of interest, then the wave shape will be maintained. An MFM coded signal will contain three basic frequency components for the various digital patterns of data. For instance, a 10 Megabit/sec MFM signal will consist of analog frequencies of 2.5 MHz, 3.33 MHz and 5 MHz. On the outer track the bit density is the lowest and the 5 and 3.33 MHz signals will look sinusoidal while the 2.5 MHz signal will have a tendency to return to the baseline. This returning to the baseline is called shouldering and is illustrated in *Figure 10*. Since this shouldering is rich in 3rd harmonic—the 2.5 MHz signal will have a strong 7.5 MHz component. The 10 Megabit/sec MFM signal will therefore have 2.5 MHz, 3.33 MHz, 5 MHz, and 7.5 MHz components which must be filtered with constant group delay in order to reproduce the original waveform. For example, if the phase shift through the filter at 2.5 MHz is 33.3°, then at 3.33 MHz the phase shift must be 44.3°, at 5 MHz—66.6°, and at 7.5 MHz—99.9°. The group delay $\frac{d\theta}{dt}$ for this case is 13.32°/MHz. This can be better interpreted as a time delay. 33.3° of a 2.5 MHz signal is equivalent to (33.3/360) × (1/2.5 MHz) or 37 ns. Similarly, 66.6° on a 5 MHz signal is (66.6/360) × (1/5 MHz) = 37 ns.



TL/F/5283-15

FIGURE 10. Typical Region 1 Waveform



TL/F/5283-16

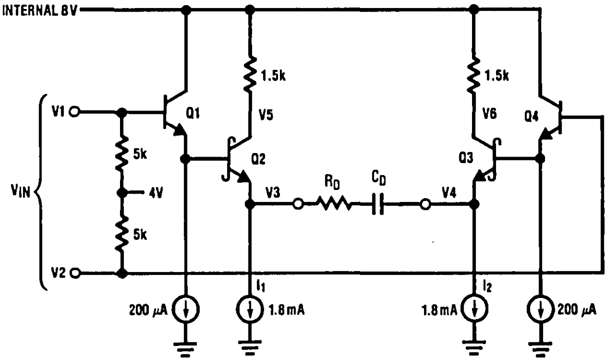
FIGURE 11. Region 1 Waveform with the Incorrect Phase Relationship

The third order Bessel Filter as shown in the 10 Mbit/sec. pulse pairing measurement board on the data sheet is designed for a constant group delay and a −3 dB point of 7.5 MHz. At this frequency the delay through the filter is 35 ns. The Gain Controlled Amplifier of the DP8464B is designed for a group delay of a 7.8 ns ±0.5 ns for frequencies up to 7.5 MHz. The 7.8 ns delay in the Gain Controlled Amplifier and the 37 ns delay in the Bessel Filter do not introduce any timing error, only a delay of 44.3 ns from the Amp Input to the output of the filter.

DIFFERENTIATOR

A simplified circuit of the first stage of the differentiator is shown in *Figure 12*. The voltages at V3 and V4 are simply two diodes down from V1 and V2. Therefore the voltage

Application Information (Continued)



TL/F/5283-17

FIGURE 12. Simplified Differentiator First Stage

across the external differentiator network (C_d in series with R_d) is the differential input voltage V1 − V2. When R_d is zero, the current through C_d is I = C × (dV/dt) or C_d × (dV_{IN}/dt). The Q2 collector current is the sum of the 1.8 mA current source plus the current through C_d or

$$1.8\text{ mA} + C_d \times (dV_{IN}/dt).$$

Similarly, the Q3 collector current is

$$1.8\text{ mA} - C_d \times (dV_{IN}/dt).$$

Therefore, the differentiator output voltage, V5 − V6, is

$$1.5\text{k} \times 2 \times C_d \times (dV_{IN}/dt).$$

The input is at a peak when V5 − V6 = 0V.

The differentiator network (C_d and R_d) should be selected so the maximum current into the differentiator network is not greater than the minimum current of I1 and I2 over temperature. In the electrical specifications, the minimum current is specified for 1.4 mA (I_{Cd} Current into Pin 1 and 24 that discharges C_d). For example, the highest analog frequency in a 10 Megabit/sec, MFM signal is 5 MHz. Since the AGC loop has forced the input to the differentiator to 2 V_{PP} (which includes the 6 dB loss of the filter), then the voltage across the capacitor (assuming R_d is 0) is:

$$V_{IN} = 1 \times \sin(2 \times \pi \times 5E6 \times t)$$

and

$$dV_{IN}/dt = 1 \times 2 \times \pi \times 5E6 \times \cos(2 \times \pi \times 5E6 \times t)$$

and the maximum slope is

$$(dV_{IN}/dt)\text{max} = 1 \times 2 \times \pi \times 5E6 = 314E5\text{ V/sec.}$$

For this example, C_d can now be calculated. Since I = C × (dV/dt), then for I = 1.4 mA, dV/dt = 314E5, then the maximum C_d must equal 45 pF. From this example, a following simple design equation for the value of C_d can be derived.

$$C_d = 445/(V_{IN} \times f_{max})$$

where

C_d is the maximum external differentiator capacitor in pF

V_{IN} is the peak to peak differential Time Channel input voltage

f_{max} is the maximum analog frequency in MHz

Note that this is the maximum value for the capacitor when the series resistor R_d is zero. The value of the capacitor can be increased if a series resistor is used, but the maximum current through the differentiator network must not exceed 1.4 mA. If too large a value for C_d is used, the delay through the differentiator will become dependent on frequency. This will not show up in a single frequency test such as a test for pulse pairing.

For the MFM code, the maximum analog frequency is ½ the data rate. For the ½(2,7) code, the maximum analog frequency is ⅓ the data rate. The above sinusoidal analysis is valid as long as the highest frequency on the outer track is nearly sinusoidal. If, however, there is significant shouldering of this signal then the value of C_d should be reduced accordingly.

The following table summarizes the value of C_d to use for a 2 V_{pp} differential signal to the time channel input.

Data Rate	Code	Maximum Frequency	C_d
5 mbits/sec	MFM	2.5 MHz	90 pF
5 mbits/sec	2,7	1.6 MHz	140 pF
10 mbits/sec	MFM	5.0 MHz	45 pF
10 mbits/sec	2,7	3.3 MHz	67 pF

As noted above, the value of the capacitor can be increased if a series resistor is used, but the maximum current through the differentiator network must not exceed 1.4 mA. For example, the components used in the Pulse Pairing Setup (see AC Electrical Specifications) are for a typical 10 Mbits/sec MFM drive. The combination of the C_d of 50 pF and the R_d of 430Ω gives a combined impedance of 768Ω at the highest frequency of 5 MHz. This gives a maximum current of 1.3 mA—well below the 1.4 mA limit.

A resistor is placed in series with C_d in order to bandlimit the differentiator response. This resistor also has an effect on the phase linearity of the differentiator. An ideal differentiator produces an output that is 90 degree phase shifted from the input regardless of the input frequency. The presence of the series resistor produces an output phase shift that is less than 90 degrees and changes with the input frequency. This resistor can be used to correct for frequency related phase problems encountered elsewhere in the read path.

Application Information (Continued)

To properly decode the information on the disk, the read channel must determine if there is a peak (or a "1") during a period of time called a detection window. The detection window for MFM and the (2,7) code is

1/(2 × data bit rate).

This detection window must accommodate errors in many parts of the system including filters, data separator, and peak shift variations in the data pattern. The pulse pairing of the DP8464B should be included in the error budget calculation.

Unequal delays through the bi-directional one shots will contribute to pulse pairing. To minimize this effect, pin 2 should be connected to 22 and pin 23 should be connected to 21. If connected this way, the delays tend to cancel. For the PCC Package, Pin 26 to Pin 2, and Pin 25 to Pin 27.

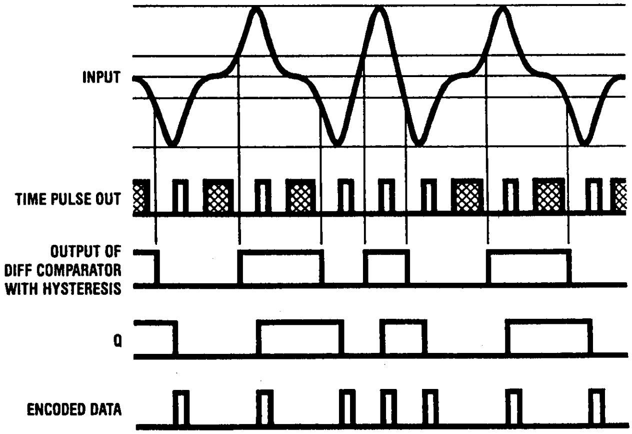
DIFFERENTIAL COMPARATOR WITH HYSTERESIS

The actual peak detection is done in the time channel with the differentiator. Unfortunately, the differentiator not only responds to signal peaks but also responds to noise at the baseline. In order to prevent this noise from generating false data, the signal at the output of the Gain Controlled Amplifier is also passed through a gating channel which prevents any output change before the input signal has crossed an established level. This gating channel comprises a differential comparator with hysteresis and a D flip-flop. The hysteresis for this comparator is set externally via the Set Hysteresis pin. The amount of hysteresis is twice the voltage on the Set Hysteresis pin. For instance, if the voltage on the Set Hysteresis pin is 0.3V, the differential input signal must be larger than 0.6V(±0.3V) before the output of the comparator will change states. The 0.6V hysteresis represents 30% of a typical 2V differential input signal level to the gating channel. The hysteresis level is usually set between 15% to 40% of the differential input signal.

The operation of the gating channel is shown in *Figure 13.* At the top is a typical Region 1 waveform which exhibits shouldering on the lowest frequency and is almost sinusoidal on the highest frequency. In this example, this waveform is fed to both the timing and the gating channel. The hysteresis level (of about 25%) has been drawn on this waveform. The second waveform is the output of the differentiator and its bi-directional one shot. This is the waveform on the Time Pulse Out pin. While there is a positive edge pulse at each peak, there is also noise at the shoulders. In this example, the Time Pulse Out is connected directly to the Time Pulse In without any external delay. This output is therefore the clock for the D flip-flop.

The third waveform in *Figure 13* is the output of the Comparator with Hysteresis which goes to the D input of the flip-flop. The true peaks are the first positive edges of the Time Pulse Out which occur after the output of the comparator has changed states. The D flip-flop will "clock" in these valid peaks to the output bi-directional one shot. Therefore, the noise pulses (due to the differentiator responding to noise at the baseline) just "clock" in the old data through the flip-flop and the output does not change.

The Q output of the flip-flop drives the output bidirectional one-shot which generates the positive edges corresponding to the peaks. The width of the data pulses can be controlled by an external capacitor from the Set Pulse Width pin to ground. This pulse width can be adjusted from 20 ns to ½ the period of the highest frequency. Typical values for this capacitor are 20 pF for a 25 ns pulse width to 100 pF for a 100 ns pulse.



TL/F/5283-19

FIGURE 13. Time and Gate Channel Operation for Region 1 Signals

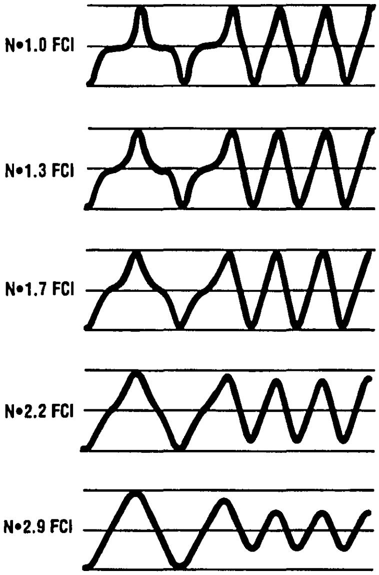
Application Information (Continued)

PULSE DETECTOR OPERATION IN REGIONS 1 AND 2

Figure 14 shows the input waveform for the lowest frequency followed by the highest frequency for an MFM code. In MFM the highest frequency is twice the lowest frequency. The outer track has the least flux changes per inch (FCI) and is illustrated in the waveforms at the top. There is so much room between the pulses that the signal returns to the baseline for the lowest frequency while there is shouldering at the highest frequency. As you go towards the inner track, the pulses become more crowded and bit interaction occurs. At the third curve down (N × 1.7 FCI), there is shouldering at the lowest frequency while the highest frequency is almost sinusoidal. At higher bit densities, the lowest frequency looks sinusoidal, while the highest frequency is decreasing in amplitude. In *Figure 14*, the first three waveforms are examples of Region 1 operation (very little change in amplitude with frequency). The last two waveforms are examples of Region 2 operation.

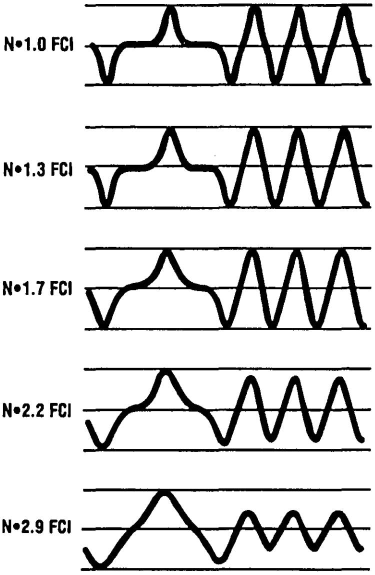
In a disk system, the bit density changes about a factor of 1.7 between the inner and the outer track. For instance, if the input waveform for the F-2F signal on the inner track of a system looks similar to waveform #4 in *Figure 14* (N × 2.2 FCI), then the outer track will have a bit density that is approximately N × 2.2/1.7 or N × 1.3 FCI. This is shown in the second waveform. Tracks half the way in will have a bit density of the average between the inner and outer tracks, in this case N × 1.7 FCI which is illustrated in the third waveform. Note that the analog waveforms change considerably with track location. Self-gating circuits ("desnakers") can be used in MFM systems which operate in these last three curves (from N × 1.7 FCI to N × 2.9 FCI). If the FCI becomes much less, the shouldering on the lowest frequency will let in too much noise. If the FCI is increased, the peak resolution gets very poor. Now we can compare these waveforms to longer run length limited codes.

Figure 15 shows the analog waveform for the lowest frequency followed by the highest frequency for a 2,7 code. In the 2,7 code, the frequency range is from F to 2.66 × F. Unlike the MFM code, there is no region where the self-gating "desnaker" will work on both the inner and outer tracks.



TL/F/5283-20

FIGURE 14. MFM F-2F Pulse Waveforms for Various Flux Changes per Inch



TL/F/5283-21

FIGURE 15. F-2.66 × F Pulse Waveforms for Various Flux Changes per Inch

Application Information (Continued)

The simplest operation is for systems operating entirely in Region 1, that is, no amplitude reduction between the highest and the lowest frequency at the inner track. The inner track is specified because the pulse interaction is most severe on the inner track. For Region 1 operation, only the Time Channel filter is required, so the Gate Channel Input is connected to the Time Channel Input. Since no external time delay is required to align the time and gate channels, the Time Pulse Out is connected directly to the Time Pulse In. The Region 1 connection is shown in *Figure 2.* The internal timing for this operation is shown in *Figure 13.*

If there is significant amplitude reduction at the highest frequency, the peak detection becomes more complex. If the worst case waveform is like the fourth waveform on *Figure 14*, then the Region 1 connection might still work satisfactorily. However, if the input begins to approach the fifth waveform, this system configuration will completely fail. One problem is that the AGC will respond to the frequency dependent amplitude modulation and distort the waveform.

Figure 16 illustrates this problem which is encountered in systems operating in Region 2. If the input digital pattern suddenly shifts from a high frequency to a low frequency, the bit density may shift from the 70% level on the BPI curve of *Figure 1* to a point at 90% on the BPI curve. As shown, the AGC loop is correcting for this frequency-induced change in amplitude by quickly decreasing the amplifier gain. The situation gets worse if the input digital pattern shifts back to a high frequency. The AGC loop now cannot quickly increase the amplifier gain, so the output waveform will very slowly increase. The AGC response to frequency related amplitude change is not desirable since the AGC is now distorting the input waveform. This can be prevented by inserting a lead network between the Gain Controlled Amplifier's output and the AGC input, as shown in *Figure 17.* This will increase the amplitude of the higher frequency into the AGC, thereby preventing the AGC from changing gain.

Another problem encountered in Region 2 operation is that the amplitude of the highest frequency may be so low that it may not trip the hysteresis level. If this happens, these peaks would not be gated on to the output. This problem can also be corrected by placing a separate filter to the gating channel which will make the amplitude of the highest frequency equal the amplitude of the lowest frequency. This is illustrated in the following example.

Consider a disk system which uses the 2,7 code and has an input at the inner track which looks like the fifth waveform in *Figure 15.* Since the flux density on the outer track is 1/1.7 times the flux density of the inner track, the outer track waveform will look like the third waveform. One filter cannot perfectly compensate both these extremes, so we design to compensate a waveform between these two. The track which is ⅔ of the way in towards the inner track is a good compromise. The filter in this example is a single zero placed such that the lowest frequency followed by the highest frequency have the same amplitude on the track ⅔ of the way in. *Figure 18* shows the operation of the inner track of this example. While the gating channel filter has made the amplitudes of the two frequencies nearly the same, the time relationship to the Time Channel Input has not been preserved. The proper operation is to have the positive edge of the signal at the Time Pulse In pin, which corresponds to a peak, be the first positive edge after the output of the comparator has changed states. This can be accomplished either of two ways. One way is to insert an external delay between the Time Pulse Out and the Time Pulse In as shown in *Figure 18.* The required delay can be determined by comparing the Time Pulse Out to the Channel Alignment Output with both external filters in the circuit. Another way is to design the Time Channel Filter with more group delay. This will probably require additional poles.

Figure 19 shows the outer track operation of our example. Notice how the system has taken care of the shoulder-induced-noise on the Time Pulse Out. The external delay has shifted the Time Pulse In so the noise is not clocking in new data to the flip-flop. It is important to select this delay such that the positive edge corresponding to a signal peak is always the first positive edge after the output of the comparator has changed states.

While the gating filter has equalized the amplitudes between the highest and the lowest frequency, the amplitude between the inner and the outer track has not been held constant. This can be seen by comparing the Gate Channel Input between *Figure 18* and *Figure 19.* In order to avoid saturating the Gain Controlled Amplifier, the voltage on the V_{REF} pin must be set so that the voltage out of the Gain Controlled Amplifier is 4 Vpp or less for all tracks. The low frequency signal on the inner track contains far more fundamental frequency than the low frequency signal on the outer track. Consequently, the low frequency inner track signal will experience more attenuation than the low frequency outer track signal in passing through the gating channel filter which, for this example, has been optimized to pass higher frequencies. The AGC tends to hold the input to the gating channel constant for a fixed V_{REF} level. Therefore the largest output from the Gain Controlled Amplifier is for the low frequency inner track signal. The voltage on V_{REF} should be adjusted so that the differential output swing of the Gain Controlled Amplifier is 4 Vpp maximum for this signal. This means that the output voltage on the outer track will be less than 4 Vpp.

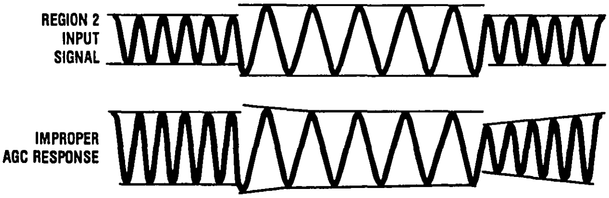
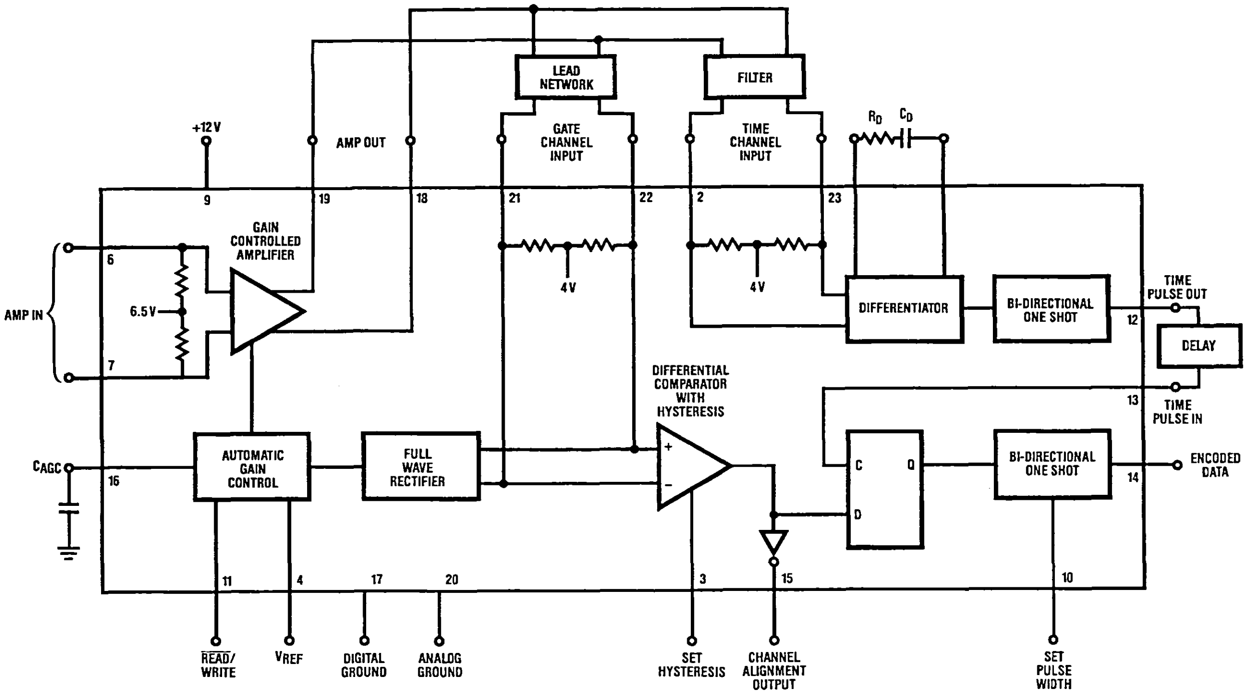


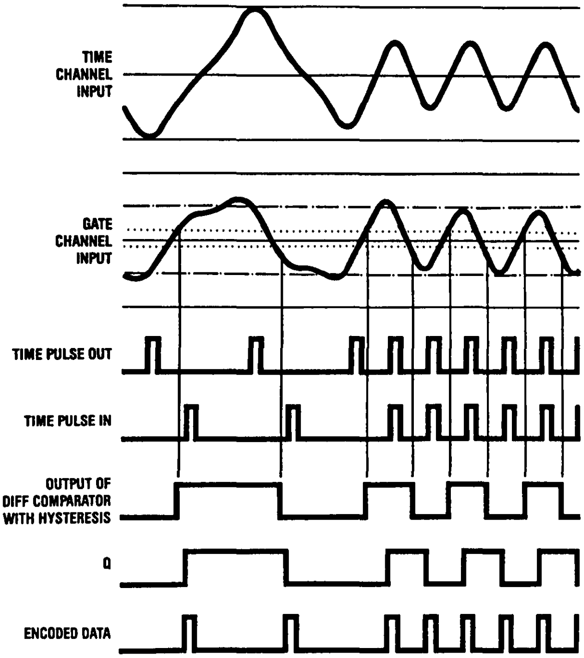
FIGURE 16. Improper AGC Response to Region 2 Signal



TL/F/5283-23

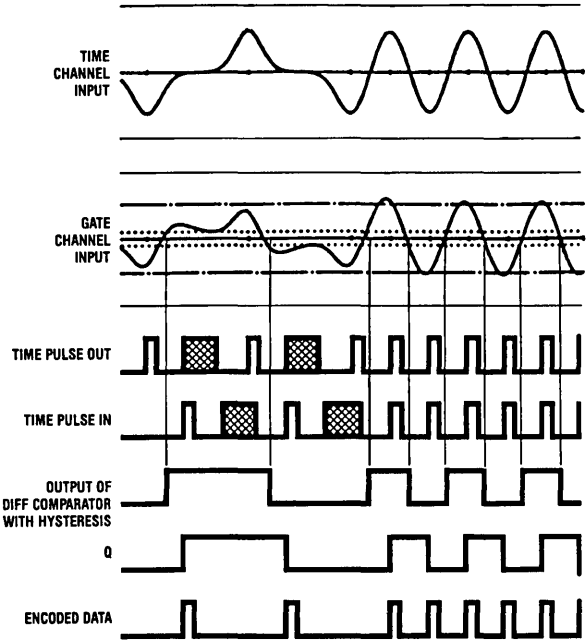
FIGURE 17. Circuit Connection for Region 2 Operation

Application Information (Continued)



TL/F/5283-24

FIGURE 18. Region 2 Inner Track Operation



TL/F/5283-25

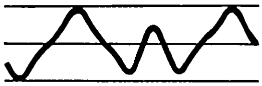
FIGURE 19. Region 2 Outer Track Operation

Application Information (Continued)

Another troublesome input pattern which should be investigated is a high frequency triplet surrounded by the lowest frequency as shown in *Figure 20*. Since the center bit of the triplet does not rise very much above the baseline, there is the possibility it will not trip the hysteresis level. This pattern should be checked to ensure the gating channel filter raises this center bit enough for the proper operation of the gating channel. The operation of the triplet in the previous example is shown in *Figure 21*.

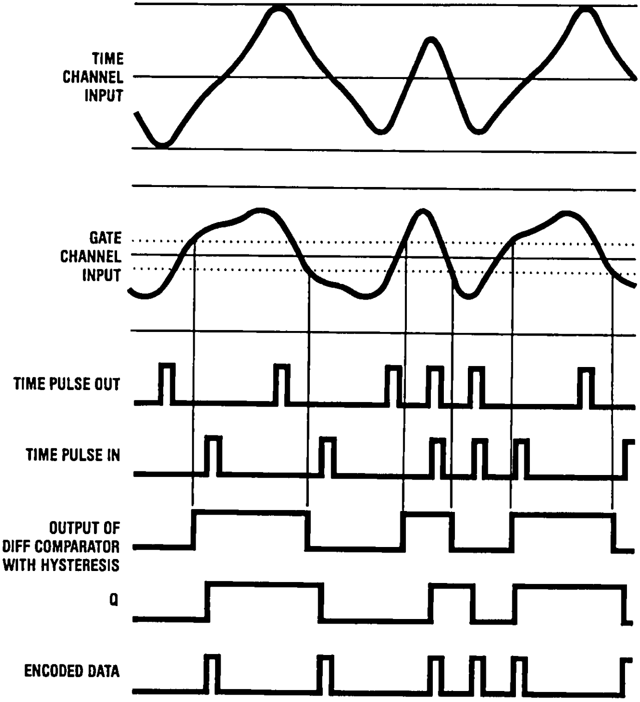
LAYOUT CONSIDERATIONS

Figure 22 is a top view of the component layout for the DP8464B application board whose schematic is shown in *Figure 23*. Care must be exercised in the board layout in order to isolate all digital signals from analog signals. The layout shown in *Figure 22* is a good example of what is required in this regard. In particular the Amp. In pins (pins 6 and 7) and the C_{DIFF} pins (pins 1 and 24) must be isolated from all digital signals. An analog ground plane will greatly aid in this isolation as will separate digital and analog grounds. The V_{CC} (pin 9) should have a 0.1 μf bypass capacitor to analog ground located close to the DP8464B. The component list is provided as an example. These components will need to be optimized for a specific read channel.



TL/F/5283-26

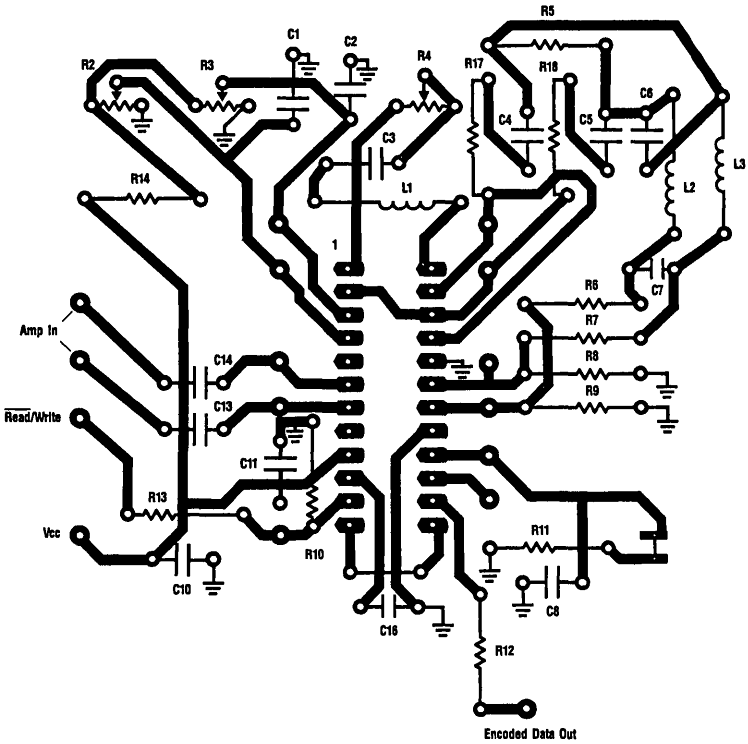
FIGURE 20. (2,7) Triplet



TL/F/5283-27

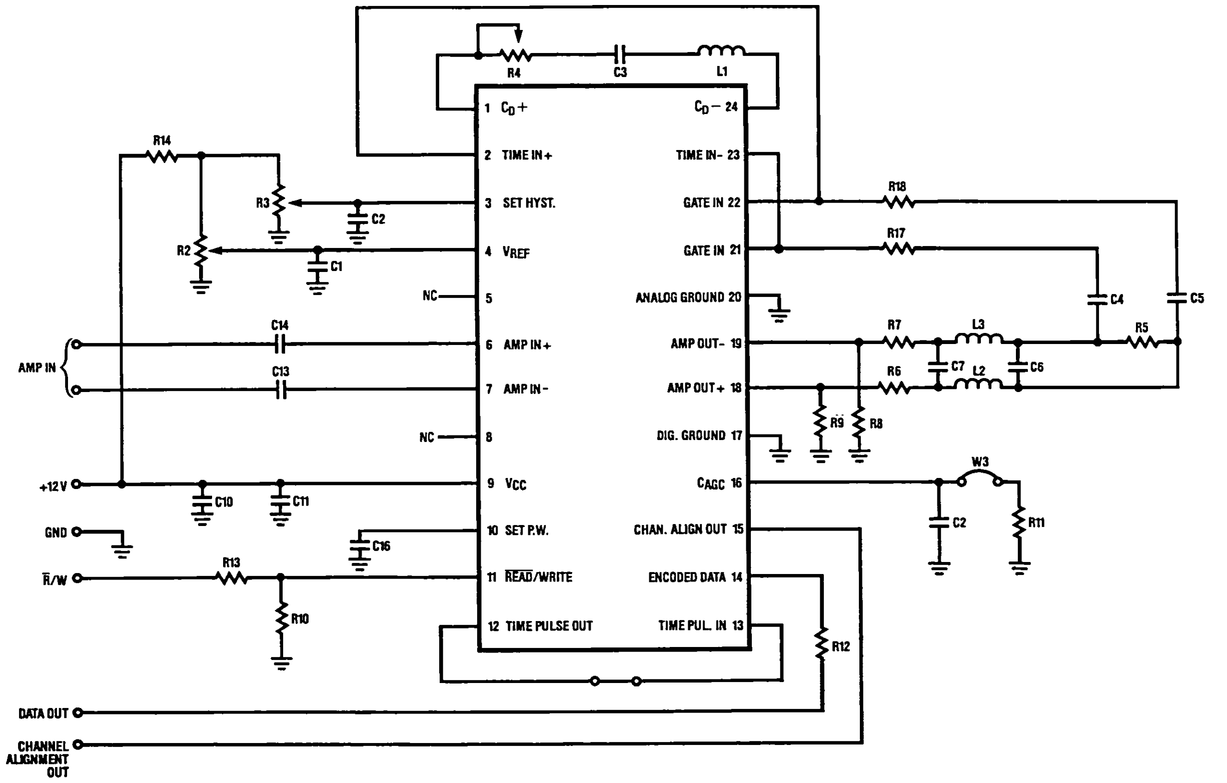
FIGURE 21. Region 2 Triplet Operation

Application Information (Continued)



TL/F/5283-28

FIGURE 22. DP8464B Component Layout—Top View



TL/F/5283-29

FIGURE 23. DP8464B Application Board Schematic

Application Information (Continued)

PARTS LIST FOR DP8464B BOARD

Component Name	Note #	Function	Value	Value for 5 Mbits/sec	Value for 10 Mbits/sec
R2	3	Adjustment for V_{REF} (AGC amplitude)	1k pot		
R3	3	Adjustment for Set Hyst. (threshold)	1k pot		
R4	2	Adjustment for differentiator network Q	5k pot		
R5	1	Low pass filter resistor	560Ω		
R6	1	Low pass filter resistor	240Ω		
R7	1	Low pass filter resistor	240Ω		
R8		Amp Out emitter bias resistor	4.3k		
R9		Amp Out emitter bias resistor	4.3k		
R10		Pull down resistor for Read/Write Pin	5.1k		
R11		Resistor in parallel with C_{AGC}	100k		
R12		Encoded Data Out damping resistor	51Ω		
R13		Read/Write damping resistor	51Ω		
R14		Divider network for Set Hyst. and V_{REF}	2.4k		
R17	6	Series resistor for Time Channel Input	Not required on DP8464B		
R18	6	Series resistor for Time Channel Input	Not required on DP8464B		
C1		V_{REF} cap	0.1 μF		
C2		Set Hyst. cap	0.1 μF		
C3	2	Differentiator cap		100 pF	50 pF
C4		Time and Gate Channel In coupling cap	0.01 μF		
C5		Time and Gate Channel In coupling cap	0.01 μF		
C6	1	Low pass filter cap		200 pF	100 pF
C7	1	Low pass filter cap		30 pF	15 pF
C8	4	C_{AGC} cap	0.01 μF		
C10		V_{CC} cap	1.0 μF		
C11		V_{CC} cap	0.1 μF		
C13	5	Amp In coupling cap	2200 pF		
C14	5	Amp In coupling cap	2200 pF		
C16		Set Pulse Width cap		100 pF	50 pF
L1	2	Differentiator inductor		3.6 μH	1.6 μH
L2	1	Low pass filter inductor		10 μH	4.7 μH
L3	1	Low pass filter inductor		10 μH	4.7 μH

BREADBOARD OPERATION NOTES

1. The low pass filter is a 3 pole Bessel with the corner frequency at 3.75 MHz for the 5Mbits/sec board (7.5 MHz for the 10 Mbits/sec board).
2. The differentiator is a simple RLC filter with the break frequency at 8.5 MHz for the 5 Mbits/sec board (17 MHz for the 10 Mbits/sec board). The resistor can be adjusted to correct for phase distortion in the channel.
3. The V_{REF} should be set at 0.5V. Since the low pass filter has a 6 dB loss, the signal on AMP OUT is 4 Vpp differential while the amplitude into the gate channel is 2 Vpp differential. The Set Hyst. should be nominally set at 0.3V.
4. The AGC attack time (the response to an increased input amplitude) is about 2 μs. To increase this time, increase the value of C8 (the AGC capacitor). The AGC decay time (the response to a decrease in amplitude) is about 10 ms. To increase this time, increase the value of R11. Care must be taken to not allow the response of the AGC loop to become too fast, otherwise loop instability may occur.
5. The input pole is set at 72 kHz (1k input impedance and a 2200 pF input coupling capacitor).
6. Pulse pairing (described in the differentiator section of this data sheet) can be caused by unequal delays through the Bi-directional one shots. To minimize this effect, pin 2 should be connected to pin 22, and pin 23 should be connected to pin 21. If connected this way, the delays tend to cancel.

REFERENCES

1. I. H. Graham, "Data Detection Methods vs. Head Resolution in Digital Recording," IEEE Transactions on Magnetics Vol. MAG-14, No. 4 (July 1978)
2. I. H. Graham, "Digital Magnetic Recording Circuits," to be published.
3. Anatol I., Zverev, Handbook of Filter Synthesis, John Wiley & Sons publisher, 1967.

Precautions for Disk Data Separator (PLL) Designs—How to Avoid Typical Problems

National Semiconductor
Application Note 414
William Llewellyn

The disk data separator/synchronizer PLL is subject to a unique set of concerns, all of which can be accommodated when adequate precautions are taken in system design.

FRACTIONAL HARMONIC LOCK

The frequency discrimination capacity of the digital phase detector within the data separator/synchronizer is suppressed whenever a pulse gate technique is employed. Although this pulse gating technique is a standard in disk drive applications and is necessary in order to allow the PLL to remain phase locked to randomly spaced disk data bits, it essentially causes the phase detector to behave as would an analog quadrature multiplier, i.e., the capture range of the loop takes on the finite value related to the loop bandwidth. Under ordinary circumstances, this is quite acceptable; however, it does permit the PLL to become susceptible to a form of quasi-stable false lock to fractional harmonics of the input frequency. (For example, a typical lock null for this phenomenon would be where the VCO stabilizes at 5/6 or 6/5 of its nominal frequency.) The conditions for occurrence of this are:

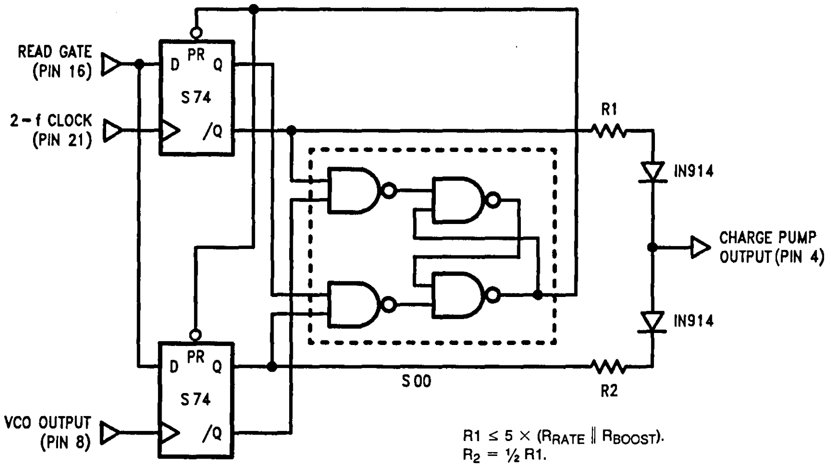
1) Pulse gate in use;
2) Periodic pattern is present (i.e., preamble);
3) Perturbation occurs either during or just prior to the periodic pattern, causing the VCO to swing outside of the dynamic capture range of the loop.

Since the capture range in a typical disk PLL configuration is on the order of ±2% of the data rate, it can be seen that harmonic lock could easily occur given an adequate perturbation of the loop. Typical causes of perturbations would be media defects and spurious noise pulses, among others, but the most commonly seen occurrence within soft-sectored systems is where an attempt is made to "read" through the write splice region on the disk (zone where the head write current is either switched on or off) during a sector search operation. Typical system-level symptoms of fractional harmonic lock are "sector not found" and "address mark not found" errors. Data (CRC or ECC) errors rarely are seen here because the phenomenon occurs primarily during the sector search routine.

Recovery from harmonic lock will occur readily when the read operation is terminated if:

1) frequency discrimination is re-introduced as the PLL is re-locked to the reference clock, or
2) the PLL bandwidth is raised to a higher value (capture range is extended) as the PLL is re-locked to the reference clock, or
3) the phase transient experienced by the PLL as its input is switched back to the reference clock is enough simply to jar the PLL back to the correct frequency.

Item #1 is incorporated within all of National's current hard disk data separator/synchronizer circuits (the DP8460/50 are excepted, being replaced by the DP8465/55). Item #2 (user optional) is incorporated within all of National's hard disk PLL's. Systems which incorporate the frequency lock function (#1) along with a suitable sector search algorithm will rarely, if ever, encounter difficulty in this area. If the



TL/F/8598-1

FIGURE 1. External Phase-Frequency Comparator Circuit for the DP8460

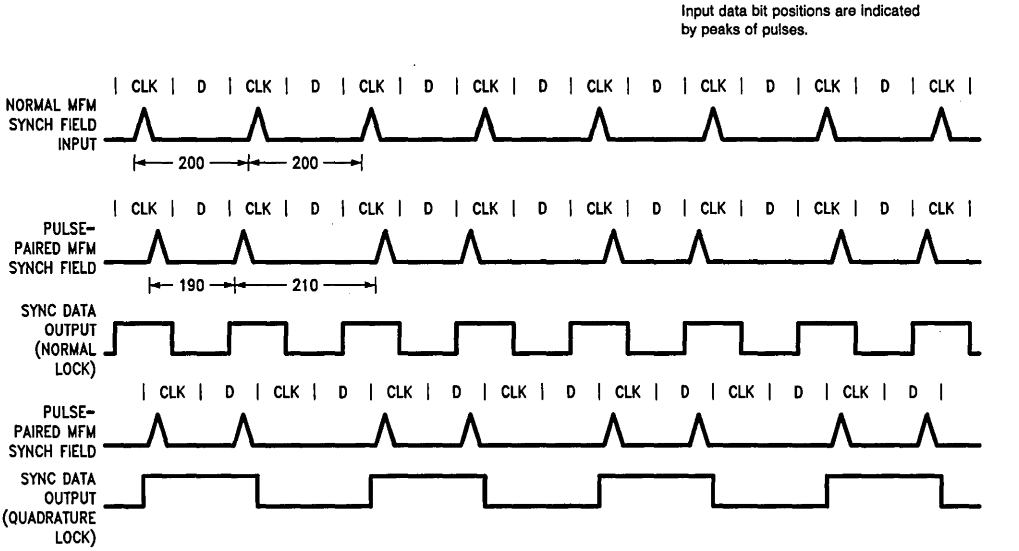

Input data bit positions are indicated by peaks of pulses.
| CLK | D | CLK | D | CLK | D | CLK | D | CLK | D | CLK | D | CLK | D | CLK |
NORMAL MFM SYNCH FIELD INPUT
200
200
PULSE-PAIRED MFM SYNCH FIELD
190
210
SYNC DATA OUTPUT (NORMAL LOCK)
| CLK | D | CLK | D | CLK | D | CLK | D | CLK | D | CLK | D | CLK | D |
PULSE-PAIRED MFM SYNCH FIELD
SYNC DATA OUTPUT (QUADRATURE LOCK)
TL/F/8598-2

FIGURE 2. Timing Diagram of PLL Quadrature Lock Within a Symmetrically Pulse-Paired Synch Field

system employs a PLL which does not incorporate frequency acquisition when locked to the reference signal (such as the DP8460/50 predecessor of the DP8465/55), either a simple external circuit may be added if desired to achieve the function (see *Figure 1*), or the PLL can be updated by inclusion of the DP8465 or DP8455. The DP8461 or DP8451 would provide the most reliable solution (frequency aquisition of both preamble and reference clock), but may be used only within hard or psuedo-hard sectored systems. (Note that the resistor values given in *Figure 1* are initial recommendations only; values may need to be adjusted to optimize system performance.)

Within systems where it becomes evident that the reading of write splices is consistently producing sector-not-found errors, while at the same time it is not possible to either modify the sector search algorithm (in order to avoid the splices) or to incorporate the lock support circuitry of *Figure 1*, the PLL can be made less sensitive to the write splice disturbance by the lowering of the loop bandwidth. This is recommended only as an interim solution until firmware or hardware accommodations can be made.

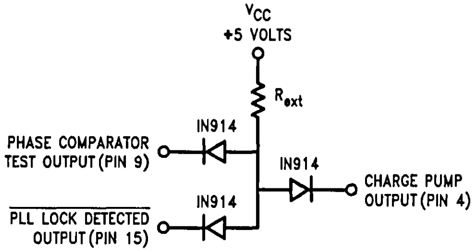
QUADRATURE LOCK

Another form of false lock may also occur (pulse gate in use) within a periodic disk pattern (preamble) given one additional condition; the periodic disk pattern being presented to the PLL exhibits a pulse-pairing phenomenon (typically introduced by the data channel electronics); see *Figure 2*. Within this particular pattern, PLL has the potential to lock to the correct frequency while remaining caught on a phase null 90 degrees from nominal. In this case, each pair of bits is interpreted by the PLL as residing in two directly adjacent windows (actually a violation of all standard disk codes) with the two subsequent windows empty. Although the bits appear to be greatly shifted within these windows, the phase corrections produced complement each other and average to a filtered DC value of zero. This repeating pattern is thus self-sustaining.

Quadrature lock is unique in that it is more likely to occur within a relatively well designed, noise-free system environment. The reason for this is that the randomizing effect noise ordinarily has on the data stream has been minimized, preserving the purity of the pulse paired pattern and thus increasing the probability of this form of lock. Again, this form of lock is generally only seen within the preamble, and may occur within either soft or hard sectored systems. The most typical symptoms are "address mark not found" or "ID error", with "sector not found" occurring, but less frequently. Easily recognized waveform patterns seen at the separator/synchronizer outputs would be (1) the Synchronized Data Output exhibits a 110011001100 .. pattern instead of the standard 1010101010 ... preamble pattern; (2) the Phase Comparator Test output pulse width consistently remains at approximately half of the VCO period (nominal width should be 7–12 nanoseconds); (3) -Lock Detected does not become active (low).

The most robust solution to this phenomenon (as well as to harmonic false lock, as mentioned above) is to incorporate a hard or pseudo hard-sectored search algorithm in conjunction with a data separator/synchronizer which employs frequency acquisition within the preamble. The frequency acquisition mode allows no residual phase or frequency error within the PLL when locked, and thus the possibility of both quadrature and harmonic lock is eliminated.

Although the modified sector search algorithm of the first solution may be possible, certain system constraints may not allow it to be practical. A second, highly effective solution to quadrature lock involves the inclusion of four passive



TL/F/8598-3

Recommended value for R_{ext}:
$10[R_{rate} \| R_{boost}] \leq R_x \leq 20\ [R_{rate} \| R_{boost}]$

FIGURE 3. External Circuitry Used to Insure Correct PLL Lock to a Pulse-Paired Synchronization Field

elements external to the National disk PLL (see *Figure 3*) which will deliberately force the window to shift away from the 90 degree phase null when (and only when) quadrature lock occurs. The passive network is automatically disabled once the PLL detects preamble lock. Although a recommended value is given for the resistor in this support circuit, some experimenting may be required in determining an optimum value for use within any particular system.

VCO JITTER

The inherent purity of the VCO's operating frequency is a key element in the accuracy of the data separator/synchronizer window generation. Any "jitter" present in the VCO frequency (any modulation of the period of the waveform by noise or any other source) will degrade the performance of the PLL. Within National's initially released DP8460/50 data separators-synchronizers, it has been found that maintaining a value of R_{rate} at or below 820Ω has a stabilizing effect on the jitter performance of the VCO circuitry. Although this is primarily a characteristic of these two devices, we are recommending the following guidelines be followed in the selecting of charge pump resistors and loop filter components for all of the hard disk data separator/synchronizer circuits (see table I):

1) An 820Ω value resistor should be substituted for the originally recommended value of 1.5 kΩ.

2) Although this new R_{rate} value is below the original DP8460 specification limit, a substitute requirement has been placed on both R_{rate} and R_{boost} to maintain proper circuit operation:

$$R_{rate} \parallel R_{boost} \geq 350\Omega$$

(i.e., the parallel value of R_{rate} and R_{boost} should not fall below 350Ω.)

3) If the inclusion of an 820Ω value for R_{rate} means a component change within an existing system (i.e., the user had been employing some higher value), all other component values associated with the loop filter must also be modified in order to maintain the original PLL response characteristics within the disk data field:

Define: $M = R_{rate}(\text{old})/820$ [eg., (1500/820)]. Then,

$$CLF1' = CLF1 * M$$
$$CLF2' = CLF2 * M$$
$$RLF1' = RLF1 / M$$

4) Additionally, in the cases where the external Phase-Frequency circuitry and/or the Quadrature lock circuitry are in use:

$$R1' = R1 / M$$
$$R2' = R2 / M$$
$$R_{ext}' = R_{ext} / M$$

Table I. Data Separator/Synchronizer Reference List

Device	Synchronized Codes	Separated Codes	Frequency Lock	Delay Trim
DP8461*	MFM; 1, N	MFM	Reference & Data	None
DP8462*	2, 7 MFM;1, N	None	Reference & Data (optional)	Optional
DP8465*	All	MFM	Reference	None
DP8451	MFM; 1, N	None	Reference & Data	None
DP8455	All	None	Reference	None

Note 1: "All" code synchronization does not include GCR.

Note 2: DP846X devices are in the 24-pin, 300 mil. package; DP845X devices are in the 20-pin, 300 mil. package.

Note 3: * Also available in 28-lead plastic chip carrier.

Note 4: DP8461 and DP8451 pinouts match the DP8465 and DP8455, respectively; for use with hard and psuedo-hard sectoring only.

Note 5: DP8462 incorporates optional frequency acquisition for 2, 7 synchronization fields, but may be used as a data synchronizer for any disk code.

Note 6: DP8451 and DP8455 also available in PCC package (20 pin).

PRINTED CIRCUIT BOARD LAYOUT RECOMMENDATIONS

The phase locked loop is inherently a sensitive device, and thus the environment in which it is operated should be optimized wherever possible to improve reliability. The following list applies for National's family of hard disk data separator/synchronizer circuits:

1) Establish a local V_{cc} island or net, separate from the main V_{CC} plane, to which the device and its associated passive components can be connected. V_{CC} supply filtering should be liberal and in very close proximity to the chip. The electrical lead length of the filter capacitance between the V_{CC} and ground pins themselves should be as short as possible (minimizing lead inductance). Inclusion of a quality high-frequency capacitor, such as a 1000 pF silver-mica capacitor, in parallel with a ceramic 0.1 μF capacitor, is recommended. (Note: the chip is particularly sensitive to inadequately filtered switching supply noise.)
2) Effective capacitive bypassing of the R_{boost} and R_{rate} pins (#2 and #3) directly to the V_{CC} pin is very important. Again, use quality, high-frequency capacitors and maintain the shortest possible electrical lead length.
3) Use the main digital ground plane for all grounding associated with the device. The ground pin and the PG1 pin should tie directly to this plane.
4) Do not locate the chip in a region of the PC board where large ground plane currents are expected.
5) Locate all passive components associated with the chip as close to their respective device pins as possible.
6) Orient the chip's external passive components so as to minimize the length of the ground-return path between each component's ground plane tie point and the chip's ground pin. (Ground noise at the loop filter components, RLF1, CLF1 and CLF2, which is not identically present at the ground pin (common mode), is coupled through the filter components into the VCO control voltage pin.)
7) Include no planing whatsoever (V_{CC} or ground) directly between adjacent pins. This will minimize parasitic capacitance at each pin. Planing between the two pin rows, however, is recommended (directly beneath the package).
8) Avoid running signal traces between pins.
9) Run no digital signal lines between or adjacent to the analog pins or signal traces (pins 1 through 7 and PG3) in order to avoid capacitive coupling of digital transients.
10) Minimize the total lead length of the C_{VCO} capacitor. Inductance in this path degrades VCO performance, as does parasitic pin capacitance.
11) Do not place any bypass filtering at the R_{VCO} pin (minor coupling of the VCO waveform into this pin is normal and acceptable).
12) Eliminate negative-going voltage transients (undershoot) at the digital input pins (pre-termination of driving lines may be necessary) to avoid drawing transient input-clamp-diode current from the device pins.
13) Minimize digital output loading; i.e., if outputs must drive large loads or long lines, employ buffers.
14) Allow unused digital output pins to float, unconnected to any net.
15) Avoid locating the chip within strong electromagnetic fields. If possible, choose the "quietest" region of the board.
16) If chip socketing is desired, use a low-profile, low mutual capacitance, low resistance, forced-insertion type (socket-strips are recommended). Avoid the use of "ZIP-DIP's".
17) Do not use wire-wrap interconnect, even in an evaluation set-up.
18) Make allowance for pin-to-pin capacitance when determining C_{VCO} (Typically 4–5 pF) from data sheet formula.

Designing with the DP8461

National Semiconductor
Application Note 415
Kern Wong

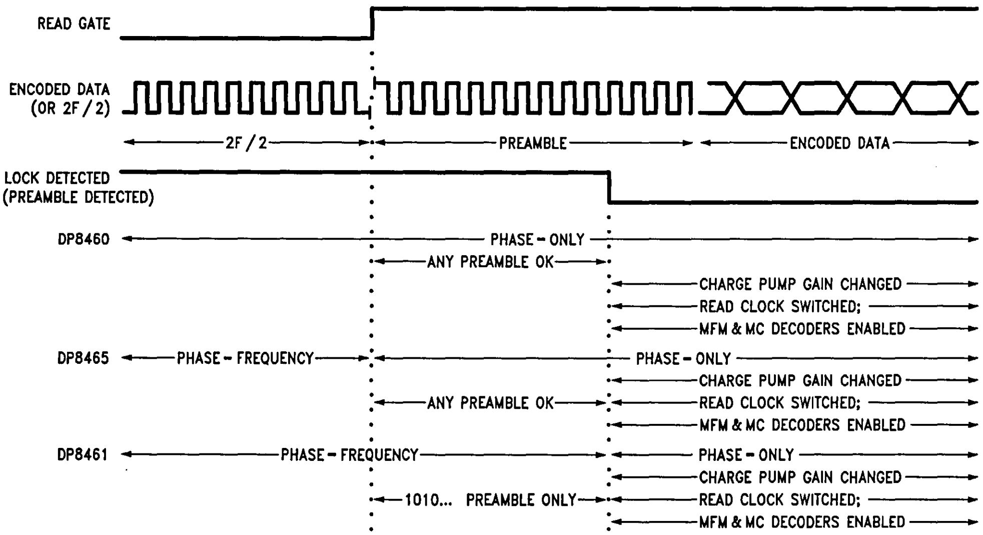
GENERAL DESCRIPTION

The DP8461 is one of the second generation data separator/synchronizer products introduced following the highly successful DP8460 single chip PLL circuit for applications in rotational memory storage systems. The DP8461 consists of the same basic functional blocks as the first generation device (DP8460). It has a proprietary pulse-gate which features an accurate silicon delay line, an edge-triggered digital phase comparator, a high speed matched charge pump, high impedance buffer amplifier, and a temperature compensated stable voltage-controlled-oscillator (VCO). It also contains MFM decoder, missing clock detector, and lock-detect control circuitry for maximum design flexibility.

Like the DP8465, the DP8461 performs PHASE-FREQUENCY COMPARISONS in the non-read mode (READ GATE is "Low"). This enhancement eliminates the possibility of false lock to the reference signal during a power-up sequence or when returning from a read operation. Furthermore, the DP8461 has been designed to allow PHASE-FREQUENCY COMPARISONS to continue into the preamble field during read mode (READ GATE is "High"). This feature eliminates the possibility of a Quadrature Lock or Harmonic Lock problem occurring in the PLL synchronization field. In order to take advantage of phase-frequency comparison during preamble detection, the DP8461 requires a "Qualified Read-Gate" (that is the READ GATE shall be asserted only within the preamble or maximum frequency field span). Since the DP8461 looks for a 1010. . . encoded data pattern while doing PHASE and FREQUENCY COMPARISONS in the read mode, it must be used only with a code employing this preamble such as MFM, (1,7) or (1,8). The DP8461 is pin-for-pin compatible with the DP8460 and DP8465 parts; and is also functionally equivalent to them with the exceptions of extended Phase-Frequency Comparison during preamble and a "Qualified Read-Gate" requirement. The DP8461 can be used as a synchronizer for MFM, (1,7), and (1,8) codes or as a data separator for MFM only. *Figure 1* shows a diagrammatic comparison of the key functional features of the three part types mentioned above.

CIRCUIT OPERATION

The DP8461 is in the non-read mode whenever the READ GATE is deasserted (Low). The 2F REFERENCE CLOCK INPUT is divided by two and transmitted to the READ CLOCK OUTPUT via a multiplexer. In this mode the VCO DIVIDED BY TWO is locked onto the 2F CLOCK DIVIDED BY TWO, keeping the VCO close to the data frequency in anticipation of locking onto the actual data stream. While in the non-read mode, PHASE-FREQUENCY COMPARISONS are always employed to eliminate any possibility of false lock.



TL/F/8599-1

FIGURE 1. DP8460/65/61 Modes of Operation Comparison Diagram

Since the DP8461 continues to make PHASE-FREQUENCY COMPARISONS when the read mode is entered, the assertion of READ GATE should only occur over a PREAMBLE or 1010... pattern. The DP8461 enters the read mode after a selectable delay time which may be either one or thirty-two VCO cycles. The 2-byte (32 VCO cycles) delay is useful in hard-sectored drives for allowing a gap pattern to pass before the PLL locks onto the data pattern. Soft-sectored drives do not need this delay. Once in the read mode, the PLL reference input is switched from the 2F CLOCK source to the ENCODED DATA INPUT. The PLL remains in the high track rate mode and continues to perform PHASE-FREQUENCY COMPARISONS to quickly lock onto the repetitive encoded preamble.

By careful selection of the loop filter components, it takes less than one byte time for the VCO to lock onto the data stream sufficiently for preamble detection to begin. As soon as 2 bytes (16 consecutive pulses) of the selected (ones or zeroes pattern) preamble are detected, the LOCK DETECTED OUTPUT goes low and this causes the PLL circuit to switch from PHASE-FREQUENCY comparisons to PHASE-ONLY comparisons. In a typical disk drive application, the LOCK DETECTED OUTPUT may be directly connected to the SET PLL LOCK INPUT. When a low level is present on the SET PLL LOCK INPUT, the CHARGE PUMP changes from a high to low tracking rate, the source of the READ CLOCK signal switches from the 2F CLOCK INPUT to the VCO CLOCK, and the MFM decoder becomes enabled and begins to output decoded NRZ data. If the DP8461 is employed as a data separator for MFM encoded data, the READ CLOCK OUTPUT and the NRZ READ DATA OUTPUT (which is synchronized to the READ CLOCK) should be used. These TTL compatible signals can be connected directly to a Disk Data Controller such as the DP8466 which controls Winchester or floppy disk drives. The MISSING CLOCK DETECTED OUTPUT can also be utilized for MFM-encoded data for soft-sectored disk drives. It should be noted, however, the circuit is designed only to recognize a missing MFM clock-bit which is framed by two existing clock bits. In order to insure the detection of an address mark, simultaneous monitoring of the NRZ output for an "A1" hexidecimal code and the MISSING CLOCK DETECTED OUTPUT for a single pulse within the same byte time is necessary.

When the READ GATE goes low, signifying the end of a read operation, the PLL reference signal is switched back to the 2F CLOCK and the LOCK DETECTED OUTPUT goes high, causing the VCO gating circuitry within the PULSE GATE to be bypassed thus allowing PHASE and FREQUENCY comparisons to occur. The PLL also returns to the high tracking rate and the output signals return to their initial conditions.

If the chip is used as a data-synchronizer (on-chip decoding not necessary) for MFM or other popular RLL codes employing a 1010... preamble, the SYNCHRONIZED DATA OUTPUT and the VCO CLOCK OUTPUT should be used. External decoding can be accomplished either in commercially available controller chips or encoder/decoder circuits, or by the customer's proprietary design.

PHASE ONLY VS. PHASE-FREQUENCY COMPARISON OPERATION

As stated earlier, the function of the PLL is to maintain phase and frequency lock between the reference signal (2F CLOCK or ENCODED DATA) and the feedback signal (VCO). A comparator that performs only phase comparison is mandatory during read-mode in the data field in order to handle the non-periodic nature of various coding schemes. With this type of detector, the phase-locked-loop functions as a feedback loop in which it responds only to the phase differences between the input and the feedback waveforms. As long as the reference and VCO singals have their edges aligned (are in phase lock), the PLL is insensitive to their frequency relationship.

During the non-read mode the PLL is required to lock onto the 2F CLOCK, a specific frequency reference that is close to the data rate. If a disturbance is somehow introduced in the system which results in cycle slipping or prolonged transient behavior of the reference clock, false lock may occur if a PHASE-ONLY comparator is being used. Similarly, in the preamble field during read mode, the PLL tries to lock onto a periodic pattern. If pulse-pairing occurs in this PLL synchronization field due to asymmetry in disk drive electronics, quadrature lock may result if a PHASE-ONLY comparator is being used. Under these circumstances PHASE comparison alone may be inadequate, since it discriminates only phase and not frequency information. A PHASE-FREQUENCY-COMPARATOR, therefore, is recommended during these modes of operation whenever false lock presents a potential problem. The DP8461 implements such a comparator. It performs identically to the PHASE-COMPARATOR in the case when both inputs to the comparator have the same frequency; however, if the inputs exhibit the slightest frequency offset, the PHASE-FREQUENCY-COMPARATOR also provides a frequency-sensitive error correction signal to ensure frequency acquisition.

As mentioned in the device description, the DP8461 requires a "Qualified Read-Gate" for proper operation in soft-sectored environments. This is necessary to accommodate phase-frequency comparison into the preamble field. If the READ GATE is allowed to be asserted randomly, it might be asserted in the data field or in the write-splice area. With the DP8461, prior to preamble detected, the PLL is operating in the phase-frequency mode. If it encounters a low frequency pattern in the data field, the VCO will try to lock onto it and thus shift its frequency. Similarly, if READ GATE becomes active in a write-splice area, the PLL may be pushed to either of its limiting frequency excursions. By employing a "Qualified Read-Gate" with the DP8461, the READ GATE will always be asserted over a repetitive 1010... pattern and thus avoid any of these problems.

PULSE GATE

The PULSE GATE has two important functions. It ensures a continuous PLL lock in the presence of random patterns encountered on the media and in the bit stream. It also provides a precise time delay (independent of process and external component variations) necessary to align the incoming data with the center of the decoding window. The delay is exactly one-half the period of the 2F CLOCK and the delay generator is referenced to the 2F CLOCK. This allows input bit jitter up to ±one-half the 2F CLOCK period.

The PULSE GATE incorporated in the DP8461 has two multiplexers which allow the circuit to switch from PHASE-FREQUENCY COMPARISON to PHASE-ONLY COMPARISON when the LOCK DETECTED signal becomes active (Low). *Figure 2* is a block diagram of the PULSE GATE which details how this is accomplished. When both the INTERNAL READ GATE and the INTERNAL LOCK DETECTED are inactive (non-read mode) the 2F CLOCK DIVIDED BY TWO and the VCO DIVIDED BY TWO signals are selected by MULTIPLEXER-1 and MULTIPLEXER-2 respectively. In this configuration phase and frequency comparisons are made between them and the possibility for a false lock occurrence is eliminated. When the INTERNAL READ GATE is active while the INTERNAL LOCK DETECTED remains inactive (read-mode, preamble detection) the ENCODED DATA and the VCO DIVIDED BY TWO signals are selected by the multiplexers. Again, PHASE-FREQUENCY COMPARISONS continue to be performed to ensure the PLL locks exactly to the data rate frequency. After sixteen pulses of consecutive preamble pattern are detected, the INTERNAL LOCK DETECTED line becomes active. MULTIPLEXER-2, under the control of the INTERNAL LOCK DETECTED singal, then switches from the VCO DIVIDED BY TWO to the GATED VCO signal. Through the circuit configured by the D-type flip-flops and the OR gate, the comparator effectively performs PHASE-ONLY comparisons (an INTERNAL VCO pulse is allowed to the input of MUX-2 only when an ENCODED DATA pulse is sensed). Thus, the DP8461 guarantees proper frequency lock of the VCO to the 2F REFERENCE CLOCK during the non-read mode and to the preamble synchronization pattern during the read mode. The circuit performs the necessary phase-only comparison in the data field during read mode operation.

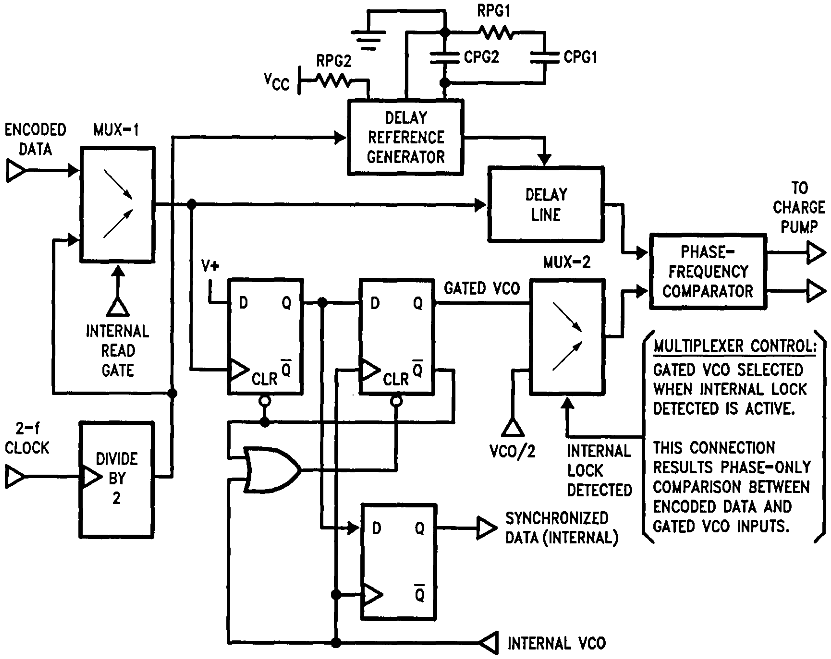
DATA SEPARATOR APPLICATION PROBLEMS

Following are some common application problems for many data separator circuit designs. The purpose of this application note is to identify these problems and to propose simple solutions thus enabling our DP8461 users to avoid these potential application problems.

FORMS OF FALSE LOCK

Two types of pseudo-lock can typically occur in a PLL within a disk drive system. A periodic input waveform must be present, such as a disk synchronization field, in conjunction with the suppression of frequency information (pulse-gate type PLL) in order for either to occur.

Pulse Gate Block Diagram


RPG1
RPG2
CPG2
CPG1
VCC
ENCODED DATA
MUX-1
DELAY REFERENCE GENERATOR
DELAY LINE
TO CHARGE PUMP
PHASE-FREQUENCY COMPARATOR
V+
MUX-2
GATED VCO
D
Q
CLR
Q̄
INTERNAL READ GATE
2-f CLOCK
DIVIDE BY 2
VCO/2
INTERNAL LOCK DETECTED
SYNCHRONIZED DATA (INTERNAL)
INTERNAL VCO
MULTIPLEXER CONTROL:
GATED VCO SELECTED WHEN INTERNAL LOCK DETECTED IS ACTIVE.
THIS CONNECTION RESULTS PHASE-ONLY COMPARISON BETWEEN ENCODED DATA AND GATED VCO INPUTS.

TL/F/8599-2

FIGURE 2

I. The first is herein termed simply "false lock", or more accurately, **fractional harmonic** lock. This occurs when the PLL is disturbed, forcing the VCO outside of the capture range (determined by the loop bandwidth; typically ±2% of the data rate) of the PLL for a period of time. The PLL is then able to acheive a pseudo-lock if (1) the ratio of the VCO frequency vs. the input frequency is an integer fraction, such as 5:6 or 6:5, and (2) the difference frequency between the VCO and the input is greater than the PLL capture range. Ideally, the error signal generated at the VCO control input, which is at the "beat" (difference) frequency of the two signals, would correct the false lock. However, this error signal is suppressed because it lies outside the frequency range of the low pass loop filter. The loop will, however, produce a self-sustaining error signal and thus will remain on the false lock null.

II. The second form of pseudo-lock is called **quadrature lock**. In this case, the PLL is able to lock to the correct frequency, but is caught on a narrow phase null which is positioned 90 degrees (w. r. t. the NRZ data period) from nominal. This phase null can occur only when there exists a pairing of periodic disk data pulses which originates in the disk drive itself (see *Figure 3*). The quadrature lock is perpetuated because the net error signal generated at the VCO input by the displaced, complimentary pulses lies well outside of the loop bandwidth and is averaged to a self-sustaining correction signal.

LOSS OF LOCK DURING READ MODE

In some systems the controller asserts the READ GATE randomly along a formatted track. If the READ GATE is asserted over a write splice, which usually contains unintelligible information, the PLL might false lock to some harmonic of the data or it might be pushed to either extreme of its allowed frequency swing. Similarly, when the READ GATE is asserted over a data field the PLL might lock to a harmonic of the data.

This problem can be completely avoided with the DP8461, which is used in conjunction with a "Qualified Read-Gate" technique. As an example, a good PLL controller algorithm that only allows assertion of the READ GATE over a preamble or similar high frequency pattern is listed below.

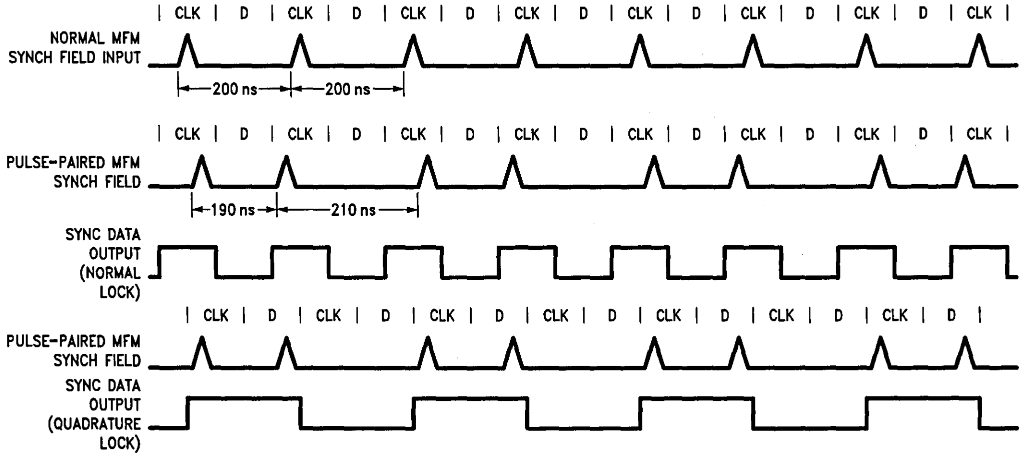
1) Deassert READ GATE—allow a 4 byte time minimum for the PLL to lock to the 2F-REFERENCE CLOCK.
2) Wait for 2.5 bytes of valid preamble pattern.
3) Assert READ GATE.
4) If valid preamble continues for 5 or more bytes then go to 5; otherwise go to 1.
5) "LOCK DETECTED" becomes active, AM search begins.
6) If AM is found, then continue the read routine; otherwise go to 1.

FALSE LOCK IN THE NON-READ MODE

The DP8461 **has been specifically designed to eliminate the possibility of false lock during the non-read mode.** This is accomplished by the use of a phase-frequency comparator in the non-read mode as was described in the PULSE GATE section.

False lock during the non-read mode can occur by two means in systems using phase-only comparisons in the non-read mode. When the power supply of the PLL circuit is switched on for the first time, the VCO ramps toward the reference frequency. The acquisition process may lock the VCO to some harmonic of the 2F REFERENCE CLOCK if the bandwidth (capture range) is not high enough. False lock can also occur in the non-read mode after an aborted read operation as described above. If the VCO has either lost lock or has been driven far from its center frequency while trying to read, false lock might occur during relock to the crystal if the capture range is narrow.

Example shows 5 Mbit/sec MFM synchronization field.
Input data bit positions are indicated by peaks of pulses.


CLK
D
NORMAL MFM SYNCH FIELD INPUT
200 ns
200 ns
PULSE-PAIRED MFM SYNCH FIELD
190 ns
210 ns
SYNC DATA OUTPUT (NORMAL LOCK)
PULSE-PAIRED MFM SYNCH FIELD
SYNC DATA OUTPUT (QUADRATURE LOCK)

TL/F/8599-3

FIGURE 3. Timing Diagram of PLL Quadrature Lock within a Symetrically Pulse-Paired Synch Field

QUADRATURE LOCK

The DP8461 **has been specifically designed to eliminate the possibility of both quadrature lock and harmonic lock in the preamble field during read mode** by extending the phase-frequency comparison technique during preamble detection in the read mode until LOCK DETECTED occurs.

Within the standard synchronization field which precedes the data field, bits are recorded at a constant frequency for a time sufficient to allow the PLL to acquire lock. With normal recording and read-circuit behavior, this synchronization information reaches the PLL as a continuous, periodic data stream. In some disk drives, if an offset has somehow been induced into the recorded information, or if read-channel asymmetry exists within the drive electronics which skews the flux reversal zero-crossing point, the synchronization field waveform which reaches the PLL may appear in the form of periodic pulse-pairs. This condition only arises when a repetitive pattern is present, giving rise to the possibility of quadrature lock. Note that quadrature lock is actually more prone to occur within systems where a low-noise design has minimized the randomizing effect which noise has on bit position.

MINIMUM PULSE WIDTH REQUIREMENT

The DP8461, as with other members of the DP8460 family of data synchronizers, has a minimum pulse width requirement on the ENCODED DATA input for proper operation. As there is no uniform pulse width specification on "Raw Read Data" outputs from disk and tape drive manufacturers, it has been found that certain drive systems output too narrow a pulse width for the DP8460 family of circuits to accept. Our recommended minimum positive pulse width is 6 ns and the minimum negative pulse width is 80% of the VCO period; this allows a maximum positive pulse width of 120% of the VCO period. Some drives utilize a bidirectional one-shot to shape the read data output pulse. The output pulse width from such drives can be readily readjusted from an RC timing network to attain acceptable minimum pulse width requirements for the PLL circuits.

Note: The chip is particularly sensitive to inadequately filtered switching supply noise.

SUMMARY

The DP8461 is another one of National's second generation single chip high performance PLL circuits for application in disk memory systems. This device features a comparator with both phase-frequency and phase-only comparison capabilities. Additionally, the PHASE-FREQUENCY COMPARISON circuit in the DP8461 has been designed to allow its operation in the preamble field during read mode so that employing it with a "Qualified Read-Gate" will eliminate all potential false and quadrature lock problems associated with soft-sectored systems. The DP8461 offers significant savings of cost and time in production, test, and maintenance since only a few fixed passive components are required for operation. The need to trim any external components has been eliminated and, since no external components determine window accuracy, the performance will not be sensitive to external variations. The chip requires a single +5 volt supply and it is housed in a narrow 24-pin dual-in-line package (also available in 28 pin PCC package). The DP8461 has the same pinout as the DP8460 and the DP8465, and thus, can be used in their designed applications provided the READ GATE has been qualified. The DP8461 can be used as a data synchronizer for MFM or any of the existing RLL codes employing a 1010... preamble, or as a data separator for MFM.

For further information, the reader should also refer to the National Semiconductor Application Note 414, Precautions for Disk Data Separator (PLL) Designs.

Designing with the DP8465

National Semiconductor
Application Note 416
Kern Wong

GENERAL DESCRIPTION

The DP8465 is a second generation of the successful DP8460 high performance PLL integrated circuit family of data separators/synchronizers. Like its predecessor, the DP8460, it consists of a proprietary pulse-gate which features an accurate silicon delay line, an edge-triggered digital phase comparator, a high speed matched charge pump, high impedance buffer amplifier, and a temperature compensated stable voltage controlled oscillator (VCO). The DP8465 also contains MFM decoder, missing clock detector, and lock-detect control circuitry for added flexibility to the system designer. There is one difference between the DP8460 and the DP8465. The DP8465 has been designed to perform PHASE FREQUENCY COMPARISONS during the non-read mode and switches to phase comparisons only when in the read mode, whereas the DP8460 employs only phase comparisons in both the read and non-read modes. This enhancement eliminates the possibility of false lock to the reference signal during a power-up sequence or when returning from a read operation. The DP8465 is 100% pin-for-pin and function-for-function compatible with the DP8460. It is a direct replacement for the DP8460 part type.

CIRCUIT OPERATION

The DP8465 is in the non-read mode whenever the READ GATE is deasserted. The 2F REFERENCE CLOCK input is divided by two and transmitted to the READ CLOCK output via a multiplexer. In this mode the VCO is locked onto the 2F CLOCK, keeping the VCO close to the data frequency in anticipation of locking onto the actual data stream. During the non-read mode PHASE-FREQUENCY COMPARISONS are employed, thus eliminating any possibility of false lock.

When the READ GATE input goes high, the DP8465 enters the read mode after a selectable delay time. This may be either one or thirty-two VCO clock cycles. The 2-byte delay is useful in hard-sectored drives for allowing a gap pattern to pass before the PLL locks onto the data. Soft-sectored drives do not need this delay. Once in the read mode, the PLL reference input is switched from the 2F CLOCK source to the ENCODED DATA input. The PULSE GATE allows a reference signal from the VCO into the PHASE COMPARATOR only when an ENCODED DATA bit is valid, thus PHASE-ONLY comparisons are made. The PLL, initially in the high-tracking mode, then attempts to quickly lock onto the repetitive encoded preamble.

By careful selection of the loop filter components, it takes less than one byte time for the VCO to lock onto the data stream sufficiently for preamble detection to begin. As soon as 2 bytes of the selected (ones or zeroes pattern) preamble are detected, the $\overline{\text{LOCK DETECTED}}$ output goes low. In a typical disk drive application, the $\overline{\text{LOCK DETECTED}}$ output may be directly connected to the $\overline{\text{SET PLL LOCK}}$ input. A low level on the $\overline{\text{SET PLL LOCK}}$ input causes the PLL CHARGE PUMP to switch from a high to low tracking-rate. At the same time the source of the READ CLOCK signal is switched from the 2F CLOCK input to the VCO clock. The MFM decoder also becomes enabled and begins to output decoded NRZ data. If the DP8465 is employed as a data-separator for MFM encoded data, the READ CLOCK output and the NRZ READ DATA output (which is synchronized to the READ CLOCK) may be used. These signals can be connected directly to a Disk Data Controller such as the DP8466 which controls Winchester or floppy disk drives. The MISSING CLOCK DETECTED output can also be utilized for MFM-encoded data in soft-sectored disk drives. It should be noted, however, the circuit is designed only to recognize a missing MFM clock-bit which is framed by two existing clock bits. In order to insure the detection of an address mark, simultaneous monitoring of the NRZ output for an "A1" hexidecimal code and the MISSING CLOCK DETECTED output for a single pulse within the same byte time is necessary.

When the READ GATE goes low, signifying the end of a read operation, the PLL reference signal is switched back to the 2F CLOCK, the $\overline{\text{LOCK DETECTED}}$ output goes high, and the VCO gating circuitry within the PULSE GATE is bypassed thus allowing PHASE and FREQUENCY comparisons to occur. The PLL then returns to the high tracking rate and the output signals return to their initial conditions.

If the chip is used as a data-synchronizer for MFM (on-chip data decoding not necessary) or other popular RLL codes, the SYNCHRONIZED DATA OUTPUT and the VCO CLOCK OUTPUT should be used. External decoding can be accomplished either in commercially available controller chips or via an encoder-decoder circuit, or by the customer's proprietary design.

PHASE ONLY VS. PHASE-FREQUENCY COMPARISON OPERATION

As mentioned above, the function of the PLL is to maintain phase and frequency lock between the reference signal (2F CLOCK or ENCODED DATA) and the feedback signal (VCO). A comparator that performs only phase comparison is mandatory during read-mode in order to handle the non-periodic nature of various coding schemes. With this type of detector, the phase-locked-loop functions as a feedback loop in which it responds only to the phase differences between the input and the feedback waveforms. As long as the reference and VCO signals have their edges aligned (are in phase lock) the PLL is insensitive to their frequency relationship.

During the nonread mode the PLL is required to lock onto the 2F CLOCK, a specific frequency reference that is close to the data rate. If a disturbance is somehow introduced in the system which results in cycle slipping or prolonged transient behavior of the reference clock, false lock may occur if a PHASE-ONLY comparator is being used. Under these circumstances PHASE comparison alone may be inadequate, since it discriminates only phase and not frequency information. A PHASE-FREQUENCY-COMPARATOR, therefore is employed within the DP8465 during this mode of operation. This comparator performs identically to the PHASE-COMPARATOR in the case when both inputs to the comparator have the same frequency; however, if the inputs exhibit the slightest frequency offset, the PHASE-FREQUENCY-COMPARATOR also provides a frequency-sensitive error correction signal to ensure frequency aquisition.

The PULSE GATE has two important functions. It ensures a continuous PLL lock in the presence of bit gaps encountered on the media and in the bit stream. It also provides a precise time delay (independent of process and external component variations) necessary to align the incoming data with the center of the decoding window. The delay is exactly one half the period of the 2F CLOCK and the delay generator is referenced to the 2F CLOCK. This allows input bit jitter up to ± half the 2F CLOCK period.

The PULSE GATE incorporated in the DP8465 has two multiplexers which allow the circuit to switch from PHASE-FREQUENCY comparison to PHASE-ONLY comparison as the circuit switches from non-read mode to read mode. *Figure 1* is a block diagram of the PULSE GATE and details how this is accomplished. The delayed output of MUX-1 is shown to be compared with either the GATED VCO or the VCO DIVIDED BY TWO. The two VCO signals are multiplexed, with the INTERNAL READ GATE as the control signal, and the output is connected to the PHASE-FREQUENCY-COMPARATOR. When INTERNAL READ GATE is inactive (non-read mode) the 2F CLOCK DIVIDED BY TWO and the VCO DIVIDED BY TWO signals are selected by MULTIPLEXER-1 and MULTIPLEXER-2, respectively. In this configuration, phase and frequency comparisons are made between them and the possibility for a false lock occurence is eliminated. When the INTERNAL READ GATE is active (read mode), however, the ENCODED DATA and the GATED VCO signals are selected by the multiplexers. Through the circuit configured by the D-type flip-flops and the OR gate, the comparator effectively performs PHASE-ONLY comparisons (an INTERNAL VCO pulse is allowed to reach the input of MUX-2 only when an ENCODED DATA pulse is sensed). Thus, the DP8465 chip guarantees proper frequency lock of the VCO to the 2F REFERENCE CLOCK during the non-read mode, and it performs the necessary phase-only comparison during the read mode.

DATA SEPARATOR APPLICATION PROBLEMS

Following are some common application problems for many data separator circuit designs. The purpose of this application note is to identify these problems and to propose simple solutions. Thus, our DP8465 users will be able to avoid these potential application problems.

A) Loss of lock during read mode

In some systems the controller asserts the READ GATE randomly along a formatted track. If the READ GATE is asserted over a write splice, which usually contains unintelligible information, the PLL might false lock to some harmonic of the data, or it might be pushed to either extreme of its allowed frequency swing. Similarly, when the READ GATE is asserted over a data field, the PLL might lock to a harmonic of the data.

To recover from this problem a recovery routine must be implemented by the disk controller. This routine should toggle READ GATE so that the PLL can lock back to the 2F REFERENCE CLOCK and, after waiting a sufficient amount of time (to frequency lock to the crystal), activate READ GATE to retry the read operation.

A superior controller PLL algorithm only allows assertion of the READ GATE over a preamble or similar high frequency pattern. An example of such an algorithm is as follows:

1) Deassert READ GATE—allow a 4 byte time minimum for the PLL to lock to the 2F-REFERENCE CLOCK.
2) Wait for 2.5 bytes of valid preamble pattern.
3) Assert READ GATE
4) If valid preamble continues for 5 or more bytes then go to 5; otherwise go to 1.
5) "LOCK DETECTED" becomes active, AM search begins.
6) If AM is found, then continue the read routine; otherwise go to 1.

B) False lock in the non-read mode

The DP8465 **has been specifically designed to eliminate the possibility of false lock during the non-read mode.** This is accomplished by the use of a phase-frequency comparator in the non-read mode as was described in the PULSE GATE section.

False lock during the non-read mode can occur by two means in systems using phase only comparisons in the non read mode. When the power supply of the PLL circuit is switched on for the first time, the VCO ramps toward the reference frequency. The aquisition process may lock the VCO to some harmonic of the 2F REFERENCE CLOCK if the bandwidth (capture range) is not high enough. False lock can also occur in the non-read mode after an aborted read operation as described above. If the VCO has either lost lock or has been driven far from its center frequency while trying to read, then while relocking to the crystal, if the capture range is not wide enough, false lock might occur.

C) Quadrature Lock

Quadrature lock is a phenomenon which may occur when the periodic pulses in the PLL synchronization field become distorted such that they appear as periodic pulse-pairs as shown in *Figure 3.* This phenomenon is usually caused by the read channel electronics or recording components in the disk drive and may give rise to a false lock condition in the PLL known as quadrature lock.

Within the standard synchronization field which precedes the data field, bits are recorded at a constant frequency for a time sufficient to allow the PLL to acquire lock. With normal recording and read-circuit behavior, this synchronization information reaches the PLL as a continuous, periodic data stream. In some disk drives, if an offset has somehow been induced into the recorded information, or if a read-channel asymmetry exists within the drive electronics which skews the flux reversal zero-crossing point, the synchronization field waveform which reaches the PLL may appear in the form of periodic pulse-pairs. This condition only arises when a repetitive pattern is present, and gives rise to the occurance of quadrature lock. Note that quadrature lock is actually more prone to occur within systems where a low-noise design has minimized the randomizing effect which noise has on bit position.

Optional External Quadrature Lock Circuitry

To eliminate the possibility of a quadrature lock condition, a simple circuit (4 passive components) solution may be employed to prevent its occurrence. The circuit shown in *Figure 2* has the effect of forcing a misalignment of the data synchronization window with respect to the input pulse pattern should the quadrature condition occur. This circuit does not affect PLL operation once proper lock has occurred, and it is disabled once PLL LOCK has been detected by the DP8465. Although a recommended value is given for the resistor in the support circuit, some experimentation may be required in determining an optimum value for use within any particular system. *Figure 3* shows a diagrammatic representation of the quadrature lock waveforms.

D) VCO Jitter

The recommended starting value for the charge pump current setting resistor, R_{RATE}, was initially 1.5 kΩ. It has been found that maintaining a value of R_{RATE} at or below 820Ω has a stabilizing effect on the jitter performance of the VCO circuitry. Thus, we recommend that this 820Ω value be substituted for the originally recommended value of 1.5 kΩ.

As shown in the DP8465 data sheet, the minimum value of R_{RATE} is 400Ω. When choosing values for R_{RATE} and R_{BOOST}, the only requirement is that the total charge pump input current is less than or equal to 2 mA. This requirement can be met by adhering to the following requirement on the parallel combination of R_{RATE} and R_{BOOST}.

$$R_{RATE} \| R_{BOOST} \geq 350\Omega$$

(i.e., the parallel value of R_{RATE} and R_{BOOST} should not fall below 350Ω.)

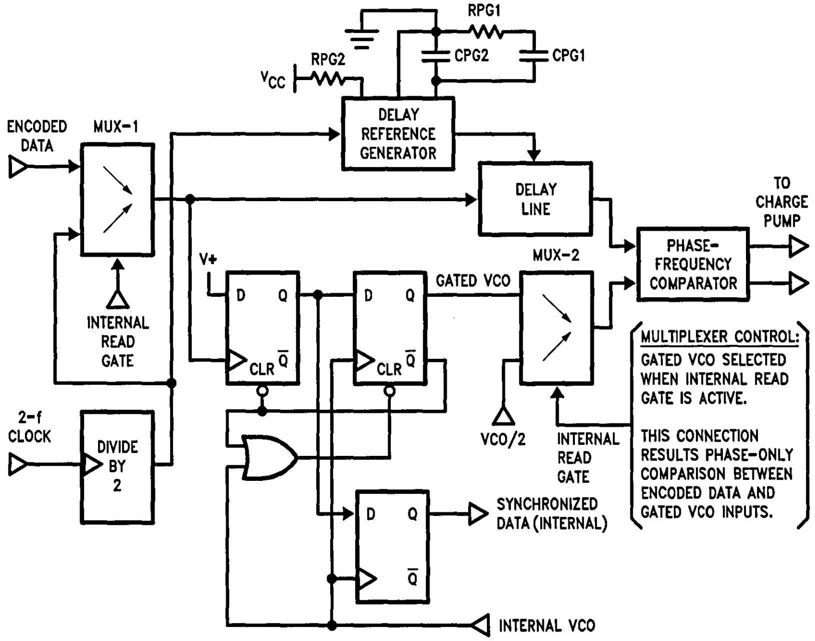
When the R_{RATE} value adjustment is implemented, all other component values associated with the loop filter must also be modified in order to maintain the original PLL response characteristics within the disk data field. The DP8465 Data Sheet shows a sample filter calculation and also several sets of loop filter component values for different values of R_{RATE}.

SUMMARY

The DP8465 is one of National's second generation single-chip high performance PLL circuits for application in disk memory systems. It features a comparator with both phase-frequency and phase-only comparison capabilities. The DP8465 offers significant savings of cost and time in production, test, and maintenance since only a few fixed passive components are required for operation. The need to trim any external components has been eliminated and since no external components determine window accuracy, the performance will not be sensitive to external variations. The chip requires a single +5V supply and it is housed in a narrow 24-pin dual-in line package (also available in 28-pin PCC package). The DP8465 is a direct replacement for the DP8460 and it may be used either as a data synchronizer for MFM or any of the existing Run-Length-Limited codes, or as a data separator for MFM.

For further information, the reader should also refer to the National Semiconductor Application Note 414, Precautions for Disk Data Separator (PLL) Designs.

Pulse Gate Block Diagram



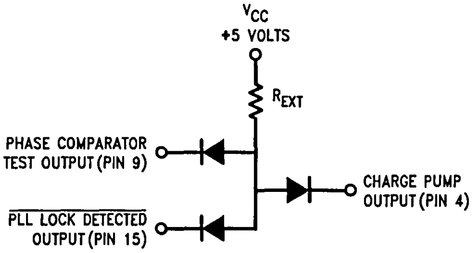
MULTIPLEXER CONTROL:

GATED VCO SELECTED WHEN INTERNAL READ GATE IS ACTIVE.

THIS CONNECTION RESULTS PHASE-ONLY COMPARISON BETWEEN ENCODED DATA AND GATED VCO INPUTS.

TL/F/8600-1

FIGURE 1.



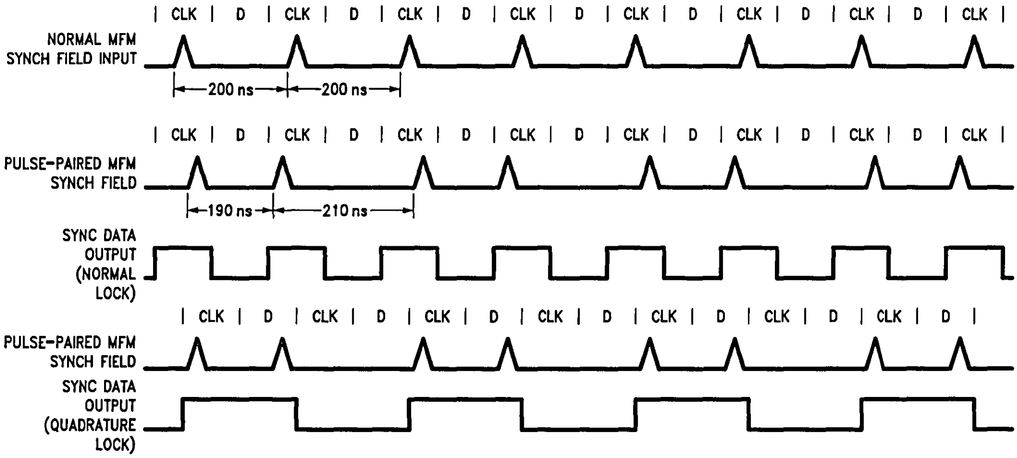
TL/F/8600-2

Recommended value for R_{EXT}: 10 [$R_{RATE} \| R_{BOOST}$] ≤ R_{EXT} ≤ 20[$R_{RATE} \| R_{BOOST}$].

Diodes must be carefully chosen for minimal zero-bias capacitance and reverse leakage current (2 pF and 100 nA or better are recommended values, respectively).

Recommended diode types: 1N4448
1N4148
1N914

FIGURE 2. External Circuitry Used to Insure Correct PLL Lock to a Pulse-Paired Synchronization Field



TL/F/8600-3

FIGURE 3. Timing Diagram of PLL Quadrature Lock Within a Symmetrically Pulse-Paired Synch Field

Section 3

Winchester Disk Data Controller

3

Section Contents

National Semiconductor Corporation

PRELIMINARY

DP8466 Disk Data Controller

General Description

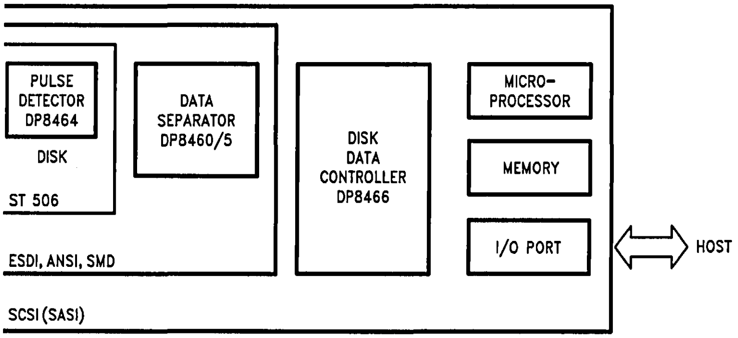
The DP8466 Disk Data Controller (DDC) is an intelligent peripheral which interfaces Winchester or Floppy disk drives to microprocessor based systems. It transfers data between a buffer memory or host system and the serial bit data stream with disk rates up to 25M-bits per second. High speed system data transfer is possible with full on-chip DMA control of buffer or main memory. The 16-bit system I/O interface allows use with any popular 8-bit, 16-bit or 32-bit microprocessor. Programmable track format enables reconfiguration of the DDC for different drive types in a multiple drive environment. Using other National DP8460 series disk data path chips, the DP8466 conforms to ST506, SMD and ESDI standard drive interfaces, as well as to intelligent standard interfaces such as SCSI (SASI) and IPI.

The DP8466 is available in three performance versions DP8466N-12, DP8466N-20 and DP8466N-25.

Part Number	Max Disk Data Rate	Max DMA Transfer Rate
DP8466N-25	25 Mbit/sec	10 Mbyte/sec
DP8466N-20	20 Mbit/sec	8 Mbyte/sec
DP8466N-12	12 Mbit/sec	6 Mbyte/sec

Features

- Easily conforms to any standard drive interface
- Compatible with floppy, hard and optical disk drives
- Compatible with 8, 16 or 32-bit microprocessor systems
- Programmable disk format
- Sector lengths up to 64k bytes, with up to 255 sectors per track
- Programmable 32 or 48-bit ECC polynomial
- Internal ECC correction in less than a sector time
- Disk data rate to 25M bits per second
- Multiple sector transfer capability
- 32 byte internal FIFO data buffer with interleavable burst capability
- 8 or 16-bit wide data transfers
- Single 32-bit or dual 16-bit DMA channel addresses
- Up to 10M bytes per second DMA transfer rate
- +5V supply, 48 pin DIP, microCMOS process



TL/F/5282-1

FIGURE 1. Typical System Configuration

Table of Contents

1.0 Introduction

National's DP8466 Disk Data Controller (DDC) chip is designed to concentrate only on the data aspects of a disk system, leaving the control signals to either a low cost single chip controller or an I/O port from a microprocessor. For this reason, the DDC will work with any standard drive interface.

The DP8466 is an advanced VLSI chip, fabricated in National's latest 2 μ CMOS technology, that allows for operation with disk data rates from the slowest floppy to the fast Winchester and Optical data rates of 25 megabits per second.

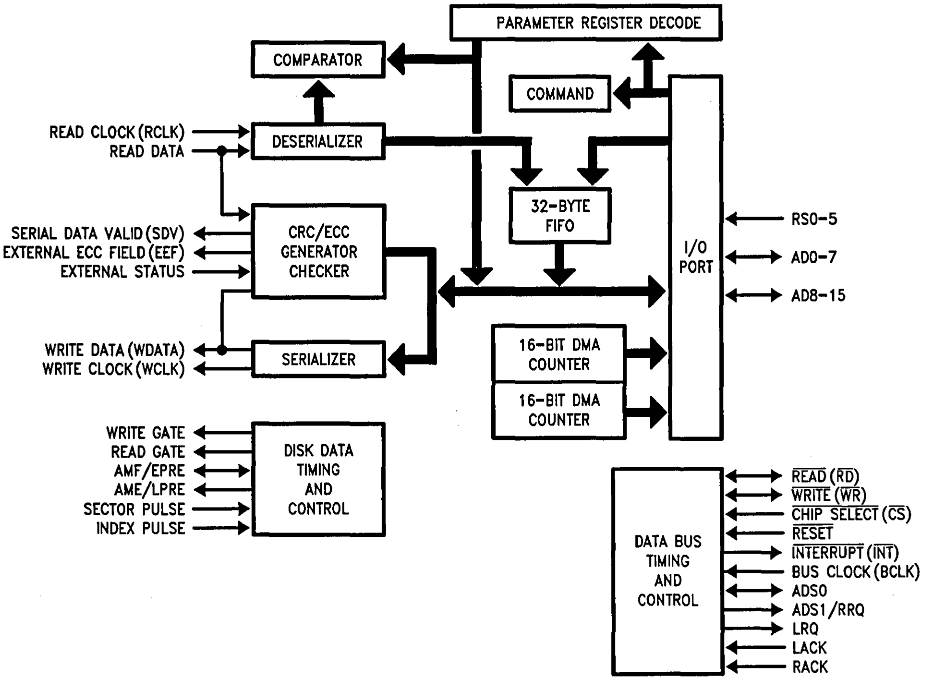
The CMOS design significantly helps the system designer because of reduced power consumption. The chip typically consumes 100 mW.

The DDC is designed for maximum programmability that not only allows the user to select any drive type he wishes, but also allows for different types of drives to be used on the same system. The chip contains 64 registers that can be loaded at any time by a microprocessor connected to the chip's bus. These registers determine the number of bytes in each field of the format, and the byte pattern that each of these fields will repeat. The number of data bytes per sector is selectable from 1 byte to 64k bytes. Finally, both the header field and the data field can each be appended with either a Cyclic Redundancy Check (CRC) field (the 16-bit code used on floppies) or a programmable Error Check and Correct (ECC) field.

The DDC allows the user to load in any 32 or 48-bit ECC polynomial from the microprocessor along with the format parameters. Once an error has been detected, the microprocessor decides whether to re-read the sector during the next revolution of the disk, or to attempt a correction. The DDC can correct errors in a time shorter than that required to read the next sector.

Key blocks in the DDC include a 32-byte FIFO and two 16-bit DMA channels that give the chip a 10 megabyte per second memory transfer capability. This high system data throughout is needed for the high speed drives now becomming available. The small FIFO allows for bursts of data to take place on the bus, thereby leaving the bus free for useful periods of time. The threshold for FIFO data storage is selectable to allow for some degree of system latency. The DDC allows for bursts of 2, 8, 16 or 24 bytes of data to be transferred between the FIFO and memory. The width of the data bus is selectable for either 8 or 16-bit transfers. The system designer selects the threshold so that when the FIFO contains the selected amount of data, the DDC will issue a request. The CPU can continue its operation and then stop to acknowledge the DDC, which then bursts the data between FIFO and memory, before the FIFO has time to overflow or underflow. With a 10 megabit per second disk data rate and a 10 megabyte per second memory transfer cycle, the bus will only be occupied for one-eighth of the time transferring data between FIFO and memory. This leaves the bus free for microprocessor usage for over 80% of the time.

Block Diagram

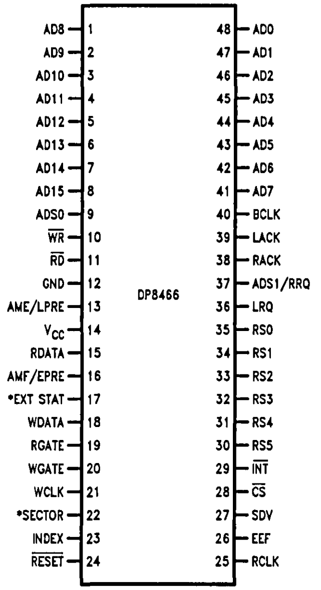


TL/F/5282-2

FIGURE 2. DDC

Connection Diagrams

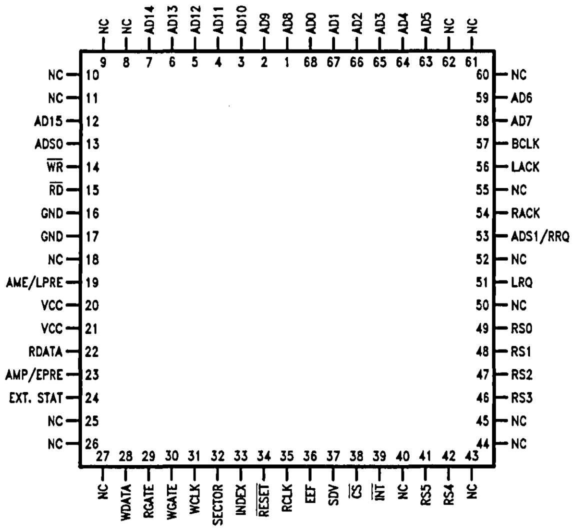
Dual-In-Line Package



TL/F/5282-3

Top View

*This pin must be grounded if not used.



TL/F/5282-76

Top View

Order Number DP8466D or DP8466N
See NS Package Number D48A or N48A

Order Number DP8466V
See NS Package Number V68A

FIGURE 3

2.0 Pin Descriptions

2.1 BUS INTERFACE PINS

Symbol	DIP Pin No.	PCC Pin No.	Type	Function
$\overline{\text{CS}}$	28	38	I	**CHIP SELECT:** Sets DDC as a standard I/O port for reading and writing registers. Configures $\overline{\text{RD}}$ and $\overline{\text{WR}}$ pins as inputs when DMA is inactive. This pin is ignored if on-chip DMA is enabled and performing a transfer.
$\overline{\text{INT}}$	29	39	O	**INTERRUPT:** An interrupt can be generated on any error, or after completion of a command, a correction cycle or any header operation.
$\overline{\text{RESET}}$	24	34	I	**RESET:** Clears FIFO, Status and Error registers. Halts DMA immediately. Halts disk read and write immediately. Does not affect parameter and most count and command registers. On power-up, must be held low for at least 32 RCLK cycles and 4 BCLK cycles. Note that both RCLK and BCLK must be active for the reset cycle to complete.
$\overline{\text{RD}}$	11	15	I/O	**READ:** • MICROPROCESSOR ACCESS MODE, with $\overline{\text{CS}}$ pin low and DMA inactive (RACK AND LACK low): Places data from FIFO or register as selected by pins RS0–5 onto the AD0–7 bus. • SLAVE MODE, with LACK pin high: Places data from FIFO onto the AD0–7/AD0–15 bus. • MASTER MODE: When DMA is active, $\overline{\text{RD}}$ pin enables data from the addressed device onto the address/data bus.
$\overline{\text{WR}}$	10	14	I/O	**WRITE:** • MICROPROCESSOR ACCESS MODE, with $\overline{\text{CS}}$ low and DMA inactive (RACK and LACK low): Latches data from AD0–7 bus to internal registers selected by RS0–5. • SLAVE MODE, with LACK pin high: Latches data from AD0–7/AD0–15 bus to FIFO. • MASTER MODE: When DMA is active, $\overline{\text{WR}}$ pin enables data from the address/data bus to the addressed device.

2.0 Pin Descriptions (Continued)

2.1 BUS INTERFACE PINS (Continued)

Symbol	DIP Pin No.	PCC Pin No.	Type	Function
BCLK	40	57	I	**BUS CLOCK:** Used as a reference clock when DDC is bus master. Used only during reset and DMA operations. Maximum ratio of RCLK/BCLK is 4 for Word Mode, and 2 for Byte Mode.
RACK	38	54	I	**REMOTE DMA ACKNOWLEDGE:** System input granting use of the bus for a remote DMA bus cycle. If RACK is de-asserted during a transfer, the current transfer cycle will complete.
LACK	39	56	I	**LOCAL DMA ACKNOWLEDGE:** System input granting use of bus for a local DMA bus cycle. If LACK is deasserted during a transfer, the current transfer cycle will complete. LACK has priority over RACK.
RS0-5	35-30	41, 42 46-49	I	**REGISTER SELECT:** Used as address inputs to select internal registers when $\overline{\text{CS}}$ pin is low.
AD0-7	48-41	58, 59 63-68	I/O	**ADDRESS/DATA 0-7:** These pins float if $\overline{\text{CS}}$ pin = 1 and DMA is inactive. • STANDARD I/O PORT, With DMA inactive and $\overline{\text{CS}}$ pin low: Command, Parameter, Count and Status register data is transferred. • SLAVE MODE, with external DMA controller active and LACK pin high: D0-7 are transferred between FIFO and memory. • MASTER MODE, with internal DMA active, and LACK pin high: A16–23, A0-7 and D0-7 are transferred depending on DMA mode and bus phase.
LRQ	36	51	O	**LOCAL DMA REQUEST:** Requests are automatically generated when the FIFO needs to have data transferred.
AD8-15	1-8	1-7 12	I/O	**ADDRESS/DATA 8-15:** • STANDARD I/O PORT, with DMA inactive and $\overline{\text{CS}}$ pin low: These pins become indeterminate, but are driven (low impedance). • SLAVE MODE, with external DMA active and LACK pin high: D8–15 are transferred between FIFO and memory. • MASTER MODE, with internal DMA active and LACK pin high: A24–31, A8-15 and D8-15 are transferred, depending on DMA mode and bus phase.
ADS0	9	13	I/O	**ADDRESS STROBE 0:** • INPUT with DMA inactive: ADS0 latches RS0–5 inputs when low. When high, data present on RS0–5 will flow through to internal register decoder. • OUTPUT: ADS0 latches low order address bits (A0–15) to external memory during DMA transfers.
ADS1/RRQ	37	53	O	**ADDRESS STROBE 1/REMOTE REQUEST:** In 32-bit DMA Mode, ADS1 latches high order address bits (A16-31) to external memory. For remote DMA modes, RRQ pin is active high when SRI or SRO bits in the OC register are set in non-tracking mode, or during a remote transfer in tracking mode. (See RT register description in DMA REGISTERS Section.)

2.2 DISK INTERFACE PINS

Symbol	DIP Pin No.	PCC Pin No.	Type	Function
RCLK	25	35	I	**READ CLOCK:** Disk data rate clock. When RGATE is high, RCLK input will be the recovered/separated clock from the recorded data and is used to strobe data into the DDC. When RGATE is low, this input should become the reference clock which will be delayed and used as WCLK to strobe data to the drive. The transition between the recovered/separated clock and reference clock must be made with no short pulses. If RCLK is inactive for longer than the maximum RCLK cycle time (see timing diagrams), the DDC will need to be reset upon RCLK becoming active. Maximum ratio of RCLK/BCLK is 4 for Word Mode, and 2 for Byte Mode.

2.0 Pin Descriptions (Continued)

2.2 DISK INTERFACE PINS (Continued)

Symbol	DIP Pin No.	PCC Pin No.	Type	Function
RGATE	19	29	O	**READ GATE:** Set active high during any disk read operation. This pin commands data separator to acquire lock. Enables RDATA input pin.
RDATA	15	22	I	**READ DATA:** Accepts NRZ disk data from the data separator/decoder.
WCLK	21	31	O	**WRITE CLOCK:** Used when NRZ data is on WDATA pin. Also active when MFM data is used, but normally not utilized. WCLK frequency follows RCLK pin.
WGATE	20	30	O	**WRITE GATE:** When writing data onto a disk, WGATE is asserted high with the first bit of data and deasserted low after the last bit of data. WGATE is also de-asserted on reset or on detection of an error.
WDATA	18	28	O	**WRITE DATA:** During any write operation, MFM or NRZ encoded data is output to disk, dependent upon MFM bit status in the DF register. This pin is inactive low when WGATE is low.
AMF/EPRE	16	23	I/O	**ADDRESS MARK FOUND/EARLY PRECOMPENSATION:** Address mark input is monitored if the HSS bit in the DF register is low (for soft sectoring). If the MFM bit in the DF register and the EP bit in the OC register are both set, then this pin becomes the EPRE control. If both functions are used, WGATE pin determines the function as follows: • WGATE asserted: EPRE output. • WGATE de-asserted: AMF input.
AME/LPRE	13	19	O	**ADDRESS MARK ENABLE/LATE PRECOMPENSATION:** If the MFM bit in the DF register is low, AME will indicate that an address mark byte(s) is being output on WDATA pin. If the MFM bit in the DF register and the EP bit in the OC register are both set, LPRE control is output (if internal MFM encoding is used).
SECTOR	22	32	I	**SECTOR PULSE:** In hard sectored drives, this signal comes from the start of a sector. In a soft sectored drive this pin must be tied low.
INDEX	23	33	I	**INDEX PULSE:** This signal comes from the disk drive, indicating the start of a track.
SDV	27	37	O	**SERIAL DATA VALID:** Asserted when the DDC is either issuing or receiving header field, internal header CRC/ECC, data field, or internal data CRC/ECC information. Mainly used for external ECC and diagnostics.
EEF	26	36	O	**EXTERNAL ECC FIELD:** Only used if the External ECC Byte Count register(s) are non-zero. Asserted when external ECC check bits are being generated (WGATE high) and checked (RGATE high).
EXT STAT	17	24	I	**EXTERNAL STATUS:** *IMPORTANT NOTE: This pin MUST be tied low if it is not to be used.* This pin has three functions: • 1: If EEW bit in the RT register is set, the read and write strobes are extended for both remote and local transfers as long as this pin is high. This is the External Wait State function. • 2: If the EEW bit in the RT register is low, this pin will accept a pulse granting valid byte alignment on the last bit of the synch byte before header or data bytes. This is an OR function with the internal synch detect. • 3: External ECC Check. Only used if External ECC Byte Count register(s) are non-zero, and EEW bit in the RT register is low. After the last byte of external ECC, this pin will accept a pulse confirming that there has been no error. A CRC/ECC error will be flagged if this pulse is not received.
V_{CC} GND	14, 12	20, 21 16, 17		**POWER, GROUND:** +5V DC is required. It is suggested that a decoupling capacitor be connected between these pins. It is essential to provide a path to ground for the GND pin with the lowest possible impedance. Otherwise any voltage spikes resulting from transient switching currents will be reflected in the logic levels of the output pins.

3.0 Internal Registers of the DDC

The numerous registers within the DDC are presented below, grouped according to their function. A key is given as an aid for the use of each register. The key data is only suggested for common operation, and should not be considered as an absolute requirement. Following this listing is a description of each register, in the order of which they are listed below. The HA column at the left of this listing gives the Hex Address of each register.

KEY

D May be updated when a different drive type is selected

C May be updated before each command

R May be read at any time

F Used during formatting

I Used during initialization

NO Operation is not possible

COMMAND

HA	Register	Bits	Write	Read
10	Drive Command Register (DC)	8	C	NO
11	Operation Command Register (OC)	8	C	NO
35	Disk Format Register (DF)	8	D	NO
00	Status Register (S)	8	NO	R
01	Error Register (E)	8	NO	R
12	Sector Counter (SC)	8	C	R
13	Number of Sector Operations Counter (NSO)	8	C	R
0F	Header Byte Count (HBC)/Interlock	3	F	R
36	Header Diagnostic Readback (HDR)	8	NO	R

DMA

HA	Register	Bits	Write	Read
37	DMA Sector Counter (DSC)	8	NO	R
37	Remote Transfer Register (RT)	8	I	NO
36	Local Transfer Register (LT)	8	I	NO
1A	Remote Data Byte Count (L)	8	C	R
1B	Remote Data Byte Count (H)	8	C	R
1C	DMA Address Byte 0	8	C	R
1D	DMA Address Byte 1	8	C	R
1E	DMA Address Byte 2	8	C	R
1F	DMA Address Byte 3	8	C	R

FORMAT

HA	Register	Bits	Write	Read
21	ID Preamble Byte Count	5	D	R
31	ID Preamble Pattern	8	D	R
22	ID Synch #1 (AM) Byte Count	5	D	R
32	ID Synch #1 (AM) Pattern	8	D	R
23	ID Synch #2 Byte Count	5	D	R
33	ID Synch #2 Pattern	8	D	R
24	Header Byte 0 Control Register (HC0)	5	D	R
14	Header Byte 0 Pattern	8	D	R
25	Header Byte 1 Control Register (HC1)	5	D	R
15	Header Byte 1 Pattern	8	D	R
26	Header Byte 2 Control Register (HC2)	5	D	R
16	Header Byte 2 Pattern	8	D	R
27	Header Byte 3 Control Register (HC3)	5	D	R
17	Header Byte 3 Pattern	8	D	R
28	Header Byte 4 Control Register (HC4)	5	D	R
18	Header Byte 4 Pattern	8	D	R
29	Header Byte 5 Control Register (HC5)	5	D	R
19	Header Byte 5 Pattern	8	D	R
2B	ID External ECC Byte Count	5	D	R
2C	ID Postamble Byte Count	5	D	R
3C	ID Postamble Pattern	8	D	R
2D	Data Preamble Byte Count	5	D	R
3D	Data Postamble Pattern	8	D	R
2E	Data Synch #1 (AM) Byte Count	5	D	R
3E	Data Synch #1 (AM) Pattern	8	D	R
2F	Data Synch #2 Byte Count	5	D	R
3F	Data Synch #2 Pattern	8	D	R
3B	Data Format Pattern	8	F	R
38	Sector Byte Count L	8	D	R
39	Sector Byte Count H	8	D	R
2A	Data External ECC Byte Count	5	D	R
20	Data Postamble Byte Count	5	D	R
30	Data Postamble Pattern	8	D	R
34	Gap Byte Count	8	F	R
3A	Gap Pattern	8	F	R

CRC/ECC

HA	Register	Bits	Write	Read
02	ECC SR Out 0	8	NO	R
03	ECC SR Out 1	8	NO	R
04	ECC SR Out 2	8	NO	R
05	ECC SR Out 3	8	NO	R
06	ECC SR Out 4	8	NO	R
07	ECC SR Out 5	8	NO	R
02	Polynomial Preset Byte 0 (PPB0)	8	D	NO
03	Polynomial Preset Byte 1 (PPB1)	8	D	NO
04	Polynomial Preset Byte 2 (PPB2)	8	D	NO
05	Polynomial Preset Byte 3 (PPB3)	8	D	NO
06	Polynomial Preset Byte 4 (PPB4)	8	D	NO
07	Polynomial Preset Byte 5 (PPB5)	8	D	NO
08	Polynomial Tap Byte 0 (PTB0)	8	D	NO
09	Polynomial Tap Byte 1 (PTB1)	8	D	NO
0A	Polynomial Tap Byte 2 (PTB2)	8	D	NO
0B	Polynomial Tap Byte 3 (PTB3)	8	D	NO
0C	Polynomial Tap Byte 4 (PTB4)	8	D	NO
0D	Polynomial Tap Byte 5 (PTB5)	8	D	NO
0E	ECC/CRC Control (EC)	8	D	NO
08	Data Byte Count L	8	NO	R
09	Data Byte Count H	8	NO	R

3.0 Internal Registers of the DDC (Continued)

DUAL-PURPOSE REGISTERS

Some of the above listed registers have dual functions depending on whether they are being written to or read from. These registers are repeated below to help clarify their operation.

HA	Register	Bits	Write	Read
02	ECC SR Out 0	8	NO	R
02	Polynomial Preset Byte 0 (PPB0)	8	D	NO
03	ECC SR Out 1	8	NO	R
03	Polynomial Preset Byte 1 (PPB1)	8	D	NO
04	ECC SR Out 2	8	NO	R
04	Polynomial Preset Byte 2 (PPB2)	8	D	NO
05	ECC SR Out 3	8	NO	R
05	Polynomial Preset Byte 3 (PPB3)	8	D	NO
06	ECC SR Out 4	8	NO	R
06	Polynomial Preset Byte 4 (PPB4)	8	D	NO
07	ECC SR Out 5	8	NO	R
07	Polynomial Preset Byte 5 (PPB5)	8	D	NO
08	Polynomial Tap Byte 0 (PTB0)	8	D	NO
08	Data Byte Count (0)	8	NO	R
09	Polynomial Tap Byte 1 (PTB1)	8	D	NO
09	Data Byte Count (1)	8	NO	R
36	Header Diagnostic Readback (HDR)	8	NO	R
36	Local Transfer Register (LT)	8	I	NO
37	DMA Sector Counter (DSC)	8	NO	R
37	Remote Transfer Register (RT)	8	I	NO

3.1 COMMAND REGISTERS

DRIVE COMMAND (DC) Hex Address (10) Write Only

The locations within this register, when written to, initiate disk commands and chip functions. For a disk operation, after the DDC has been configured, this register is loaded to initiate command execution.

DO2	DO1	H02	H01	FMT	MSO	SAIS	RED
7	6	5	4	3	2	1	0

RED: Re-enable DDC

A 1 should be written into this location during the power up initialization process (see POWER UP AND INITIALIZATION Section), or after an error has been encountered in order to re-enable the DDC to accept commands. (NOTE: If the RES bit in the OC register has been set, a 0 should be written to that location before this operation is performed.) If no error has been encountered, and a command is being issued, a zero should be written to this bit.

SAIS: Start at Index or Sector

0 Operation begins only upon receipt of an index pulse.

1 Operation begins on either an index pulse or sector pulse for hard sector drives or immediately for soft sector drives.

MSO: Multi-sector Operation

0 Single-sector operation.

1 Multi-sector operation using NSO register.

FMT: Format Mode

0 No Format Operation.

1 When set, along with other DC register bits, will initiate disk formatting upon receipt of an index pulse.

HO1, 2: Header Operation Bits:

H02 H01

0 0 *IGNORE HEADER:* associated data transfer operation will take place with any valid sector encountered.

0 1 *COMPARE HEADER:* Normal mode used to find a specific sector. The Header Pattern registers contain the comparison pattern.

1 0 *WRITE HEADER (Write ID):* Normally used only during Format mode to write ID patterns to disk.

1 1 *READ HEADER (Read ID):* Reads header information from disk for diagnostic purposes.

DO1, 2: Data Operation Bits:

D02 D01

0 0 *NO OPERATION:* Can be used only with an Ignore Header command. No disk operation is performed with this combination, and it can be used along with the RED command to re-enable the DDC (see OPERATING MODES).

0 1 *CHECK DATA:* No DMA action and no data movement between disk and FIFO. CRC/ECC checks are calculated and interrupts, if enabled, are asserted on proper conditions. DFE bit in Error register will be set if a data CRC/ECC error occurs unless in Interlock Mode.

1 0 *WRITE DATA:* Initiates local DMA action to fill the FIFO. Writes data to disk with the proper pre and post appendages in the data field. FIFO is replenished by local DMA.

1 1 *READ DATA:* Data enters FIFO from disk, and local DMA transfer is initiated when the FIFO contains the number of bytes specified by the Burst Length in the LT register.

The following table shows a list of valid commands combining the H01, H02, D01, D02, FMT bits from the DC register and the FTF bit in the DF register. No other DC register combinations are allowed.

3.0 Internal Registers of the DDC (Continued)

Valid DDC Commands

DC Register					DF Reg	Operation
D02	D01	H02	H01	FMT	FTF	
0	0	0	0	0	X	No Operation
0	1	0	1	0	X	Check Data, Compare Header
0	1	1	0	0	X	Check Data, Write Header
0	1	1	1	0	X	Check Data, Read Header
1	0	0	0	0	X	Write Data, Ignore Header
1	0	0	1	0	X	Write Data, Compare Header, (normal write)
1	0	1	0	0	X	Write Data, Write Header
1	0	1	0	1	0	Write Data,Write Header, Format with No FIFO Table
1	0	1	0	1	1	Write Data, Write Header, FIFO Table Format
1	1	0	0	0	X	Read Data, Ignore Header, (recover data)
1	1	0	1	0	X	Read Data, Compare Header, (normal read)
1	1	1	1	0	X	Read Data, Read Header

OPERATION COMMAND (OC)
Hex Address (11) Write Only

The fields within this register enable on-chip operations. In non-tracking mode, a remote DMA operation will be initiated by loading the SRO or SRI bits in this register.

IR	SCC	EP	SRO	SRI	EHI	EI	RES
7	6	5	4	3	2	1	0

RES: Reset DDC

0 Clears a previously set RES function. Allows normal operation.

1 DDC immediately enters a stand-by mode. The FIFO is reset, Status and Error registers are cleared and all operations in progress are stopped. DDC is placed in the Reset mode (see OPERATING MODES). RGATE and WGATE pins are de-asserted if active. All DMA counters are cleared. Format Parameter, DMA Address and ECC registers are unaffected.

EI: Enable Interrupts

0 Disabled, INT pin remains inactive high.

1 Enables interrupts generated by the following:

- Correction cycle complete.
- Error which sets ED bit in Status register.
- Command successfully completed (including independent remote DMA transfer).

EHI: Enable Header Interrupt

EI bit must be set if this bit is set.

0 Disabled.

1 Interrupt issued at start of ID postamble field when:

- Header matches in Compare Header operation.
- Header finished in Read, Write or Ignore Header operation.

SRI, SRO: Start Remote Input, Start Remote Output

These bits are only operational in non-tracking mode. The Remote Start Address and Remote Data Byte Count registers must be loaded first.

SRI	SRO	
0	0	Remote DMA operation unchanged.
0	1	*START REMOTE OUTPUT*: Asserts RRQ pin and RCB flag in Status register, to begin a remote DMA operation from memory to I/O Port.
1	0	*START REMOTE INPUT*: Asserts RRQ pin and RCB flag in Status register, to begin a remote DMA operation from I/O Port to local memory.
1	1	*STOP CURRENT REMOTE OPERATION*: RRQ pin is de-asserted and RCB flag is reset in Status register.

EP: Enable Precompensation

0 Early and late precompensation signals are forced low during a disk write operation.

1 Permits precompensation signals to be output to external precompensation circuitry (see MFM ENCODED DATA). This bit is only valid if the MFM bit is set in the DF register.

SCC: Start Correction Cycle

0 No correction is attempted.

1 Setting this command will begin the internal correction cycle. The CCA flag in the Status register is set and drive commands should not be issued during this time. At the completion of the cycle, an interrupt is issued.

IR: Interlock Required (Interlock Mode)

0 No interlock function.

1 The interlock (HBC) register must be written to after the header operation has completed and before the DDC encounters the data postamble field. This allows updating of header bytes during a Format operation or changing of drive commands during a multi-sector operation. Normally used with the header interrupt enabled.

3.0 Internal Registers of the DDC (Continued)

DISK FORMAT (DF) Hex Address (35) Write Only

ID2	1D1	IH2	1H1	FTF	HSS	SAM	MFM
7	6	5	4	3	2	1	0

MFM: MFM Encode

(See MFM Encoded Data section.)

0 NRZ data is output on the WDATA pin when WGATE is active.

1 MFM data is output on the WDATA pin when WGATE is active. Also configures AMF/EPRE and AME/LPRE pins as EPRE and LPRE outputs when Write Gate is active. Precompensated outputs are enabled by the EP bit in the OC register.

SAM: Start with Address Mark

(See Formatting section)

0 Address Marks will be generated in the synch #1 fields if MFM bit = 1, or AME will be generated if MFM bit = 0.

1 Address Mark Enable will be generated in ID preamble if MFM bit = 0.

HSS: Hard or Soft Sectored

(See Hard Sector vs. Soft Sector Operation).

0 Sets DDC for soft sectored operation.

1 Sets DDC for hard sectored operation.

FTF: FIFO Table Format

0 Formatting is done without the use of DMA.

1 The local DMA channel loads the correct number of header bytes (HBC register) per sector into the FIFO from local memory. This data is then substituted for the header bytes during a format operation.

IH1, 2: Internal Header Appendage

IH2 IH1

0 0 No CRC/ECC is internally appended, but external ECC must be attached.

0 1 16-bit CRC CCITT polynomial is appended.

1 0 32-bit programmable ECC code is appended.

1 1 48-bit programmable ECC code is appended.

External ECC may be used with any internal CRC/ECC selection. 1 to 31 bytes of external ECC may be added.

ID1, 2: Internal Data Appendage

ID2, ID1

0 0 No CRC/ECC internally appended.

0 1 16-bit CRC CCITT polynomial is appended.

1 0 32-bit programmable ECC code is appended.

1 1 48-bit programmable ECC code is appended.

External ECC can be appended to any of the four cases dependent upon the Data External ECC Byte Count register.

STATUS (S) Hex Address (00) Read Only

The RESET pin and the RES bit in the OC register reset all of the bits in this register.

ED	CCA	LCB	RCB	LRQ	HMC	NDC	HF
7	6	5	4	3	2	1	0

HF: Header Fault

This bit is valid after a Compare Header or Read Header operation.

SET CRC/ECC error detected in a header field.

RESET This bit is reset when the DDC begins the next disk operation after a new disk command has been issued.

All ID fields entering the DDC during the operation are checked. The HF bit will be set if an error is detected in any header field encountered. However, if the header being sought is found and has no CRC/ECC error, the HF bit is reset. This bit does not produce an error that will stop operation, assert an interrupt, or set the ED bit in the Status register in a compare header operation, but will in a read header operation.

This bit could provide useful diagnostic information if a Sector Not Found error occurs (see Error Register in this section).

NDC: Next Disk Command

SET DDC will accept a new command into the DC register. The header operation is completing the last sector being operated on.

RESET On receipt of a new disk command.

HMC: Header Match Completed

For each of the following, this bit is set and the interrupt is generated at the start of the header postamble field.

Compare Header Operation:

SET Header field correctly matched with no CRC/ECC error.

RESET At beginning of subsequent header operation.

Read Header Operation:

SET Header field has been read with no CRC/ECC error.

RESET At beginning of subsequent header operation.

Ignore Header or Write Header Operation:

SET Always set at end of header field.

RESET At beginning of subsequent header operation.

LRQ: Local Request

This bit follows the LRQ pin, and allows application of the DDC in a polled mode.

SET LRQ pin is asserted.

RESET LRQ pin is not asserted.

RCB: Remote Command Busy

Non-Tracking Mode:

SET When OC register is loaded with a DMA instruction.

RESET Upon completion of the instruction or upon internal or external reset.

Tracking Mode:

SET When RRQ pin is first asserted in a disk write mode, or when the Drive Command register is loaded in a disk read mode.

RESET Upon completion of the instruction or upon internal or external reset.

LCB: Local Command Busy

SET When command requiring local DMA is loaded.

3.0 Internal Registers of the DDC (Continued)

RESET Upon completion of the last local or remote DMA transfer (in tracking mode) or upon internal or external reset.

CCA: Correction Cycle Active

SET On asserting SCC bit in the OC register.

RESET At the end of the correction cycle, simultaneously with the INT pin, if enabled.

ED: Error Detected

SET On assertion of one or more bits in the Error register.

RESET Upon internal or external reset.

ERROR(E) Hex Address (01) Read Only

Any bit set in this register generates an interrupt (if EI bit in the OC register is set) and stops the current operation. The RESET pin and the RES bit in the OC register reset all of the bits in this register.

LI	CF	FDL	NDS	SO	SNF	DFE	HFASM
7	6	5	4	3	2	1	0

HFASM: Header Failed Although Sector Number Matched

(See HFASM description in ADDITIONAL FEATURES)

SET The header bytes(s) marked with the EHF bit in the corresponding HC register(s) matched correctly, but other header bytes were in error.

RESET Upon internal or external reset.

DFE: Data Field Error

SET On detection of a data field CRC/ECC error in a Read Data or Check Data operation. This bit may be set when another error occurs; especially an error occurring during a Write operation. These errors would be Sector Overrun or FIFO Data Lost.

RESET Upon internal or external reset.

The RED command must be loaded into the DC register if error correction is to be attempted.

SNF: Sector Not Found

SET When header cannot be matched for two consecutive index pulses in any Compare Header operation.

RESET Upon internal or external reset.

SO: Sector Overrun

SET If RGATE is active and FIFO is being written to when a sector or index pulse is received. If WGATE is active, this bit is set when a sector or index pulse is received.

RESET Upon internal or external reset.

An SO error will not occur during a Format operation.

NDS: No Data Synch

SET If a sector or index pulse occurs while the DDC is waiting to byte align on the first data synch field (synch #1 or synch #2), or if the DDC byte aligns to the first synch word of the data field but does not match to subsequent bytes (synch #1 or synch #2).

RESET Upon internal or external reset.

FDL: FIFO Data Lost

SET During a disk read operation if the FIFO overflows, or during a disk write operation if the FIFO is read when it is empty.

RESET Upon internal or external reset.

CF: Correction Failed

SET If correction is attempted (SCC bit set in OC register) and correction failed.

RESET Upon internal or external reset.

LI: Late Interlock

Will only occur if IR bit in OC register is set.

SET Controlling logic has failed to write to the Interlock (HBC) register before the end of the data field of the present sector.

RESET Upon internal or external reset.

SECTOR COUNTER (SC)

Allowable Value 0–255 **Hex Address (12) Read/Write**

In a multi-sector operation, the SC register is first loaded with the starting sector number. It is incremented after each header operation is completed. The contents of the SC register will replce any header Byte if the SSC bit is set in the corresponding HC register.

NUMBER OF SECTOR OPERATIONS COUNTER (NSO)

Allowable Value 0–255 **Hex Address (13) Read/Write**

In a multisector operation, the NSO register is loaded with the number of sectors to be operated on. It is decremented after every header operation. When zero, the command is finished. This counter must be reloaded after a reset of the DDC.

HEADER BYTE COUNT (HBC)/INTERLOCK

Allowable Value 2–6 **Hex Address (0F) Read/Write**

This register loads the DMA with the number of header bytes to expect in a Read Header, or a Format operation where FIFO table formatting is used. This register is also used in interlock mode to signal completion of update. The upper five bits of this register are pulled low when read.

HEADER DIAGNOSTIC READBACK (HDR)

Hex Address (36) Read Only

If a Compare Header/Check Data operation is performed and an HFASM error occurs, the header bytes for that sector will have been loaded into the FIFO. By consecutively reading this address, the header bytes are read from the FIFO to the microprocessor. Data will be valid for only the number of header bytes specified in the parameter RAM. (NOTE: This is a dual function register, sharing operation with the Local Transfer register, see DMA REGISTER.)

SECTOR BYTE COUNT REGISTER (L, H)

Allowable Value 1–64k **Hex Address (38, 39) Read/Write**

The two bytes (most and least significant) that comprise this register are loaded during initialization, and define the data field size for each sector. The number of bytes transferred with local DMA is always equal to what has been loaded into this register. Loading *both* with zero is not allowed.

3.0 Internal Registers of the DDC (Continued)

3.2 DMA REGISTERS

LOCAL TRANSFER (LT) Hex Address (36) Write Only

This is a dual function register, sharing operation with the Header Diagnostic Readback (HDR) register (see COMMAND REGISTERS). If any internal DMA is being used, or if the Remote Data Byte Count registers will be read by the processor, the LT (and RT) register must be loaded before the Sector Byte Count and Remote Data Byte Count register pairs.

LBL2	LBL1	LTEB	LA	LSRW	RBO	LWDT	SLD
7	6	5	4	3	2	1	0

SLD: Select Local DMA Mode

0 *SLAVE MODE*: External DMA must be used in place of on-chip DMA.

1 *NON-TRACKING MODE*: Local DMA is enabled. Whenever local transfers are needed, the DDC becomes the bus master.

TRACKING MODE: Local and remote DMA are enabled. DMA transfers are interleaved (see DMA in DATA TRANSFER section).

LWDT: Local Word Data Transfer

0 Address increments by 1, 8 bit wide transfers.

1 Address increments by 2, 16 bit wide transfers. Address, A0, remains unchanged as it was set by the DMA address.

RBO: Reverse Byte Order

Valid if LWDT bit is set.

0 First byte to/from FIFO is mapped onto the AD0–7 bus.

1 First byte to/from FIFO is mapped onto AD8–15 bus (e.g. 68000).

LSRW: Local Slow Read And Write

0 DMA cycles are four clock periods.

1 DMA cycles are five clock periods. RD and WR strobes are widened by one clock period.

LA: Long Address

Valid only if SLD = 1, and SRD = 0 in Remote Transfer register.

0 16 address bits are issued and strobed by the ADS0 pin. ADS1/RRQ is available for use by the remote DMA.

1 32 address bits are issued, the lower 16 are strobed by ADS0 pin. The most significant 16 address lines are only issued when a rollover from the least significant 16 address lines occurs, or after loading the upper half of the 32-bit address. When the upper 16 address lines are issued, that DMA cycle is five clock cycles long if no internal or external wait states are used.

LTEB: Local Transfer Exact Burst

0 When DMA tranfer is needed, the FIFO will be filled when writing to disk or emptied when reading from disk.

1 When DMA tranfer is needed, the FIFO will receive (when writing) or deliver (when reading) an exact burst of data.

LBL1, 2: Local Burst Length

LBL2	LBL1	
0	0	1 word (2 byte)
0	1	4 word (8 byte)
1	0	8 word (16 byte)
1	1	12 word (24 byte)

When reading from disk, these bits select the number of bytes needed in the FIFO in order to generate an LRQ signal. When writing, these bits select the number of bytes that need to be removed from a full FIFO in order to generate an LRQ. In either case, if the LTEB bit is set, this bit pair indicate how many data transfers will be allowed before LRQ is removed.

REMOTE TRANSFER (RT) Hex Address (37) Write Only

This is a dual function register, sharing operation with the DMA Sector Counter (DSC) (see DSC at the end of this section). If any internal DMA is being used, or if Remote Data Byte Count registers will be read by the processor, the RT (and LT) register must be loaded before the Sector Byte Count and Remote Data Byte Count register pairs.

RBL2	RBL1	RTEB	TM	RSRW	EEW	RWDT	SRD
7	6	5	4	3	2	1	0

SRD: Select Remote DMA

0 Remote DMA inhibited, ADS1/RRQ pin is configured as ADS1.

1 Remote DMA enabled. This is necessary but not sufficient to start remote transfer.

RWDT: Remote Word Data Transfer

0 Remote address increments by 1.

1 Remote address increments by 2. Address A0 remains unchanged as it was set by the starting DMA address.

EEW: Enable External Wait

0 No external wait states acknowledged. Functions 2 and 3 of EXT STAT pin are enabled (see PIN DESCRIPTIONS).

1 The EXT STAT pin will lengthen RD and WR strobes during DMA transfers as long as it is maintained at a high level.

RSRW: Remote Slow READ/WRITE

0 Remote DMA cycles are four clock periods long.

1 Remote DMA cycles are five clock periods long, if external wait states are not asserted.

TM: Tracking Mode

See Tracking Mode description in DATA TRANSFER Section.

0 DMA channels are independent and addresses are allowed to overlap.

1 DMA channel addresses are not allowed to overlap.

RTEB: Remote Transfer Exact Burst

0 If a remote transfer has been initiated, the RRQ pin will remain asserted until the number of bytes specified by the Remote Data Byte Count registers has been transferred, or until the oper-

3.0 Internal Registers of the DDC (Continued)

ation is reset or SRI and SRO bits in the OC register are both set when in non-tracking mode, or when DMA sector counter reaches zero when in tracking mode.

1 If a remote transfer has been initiated, the RRQ pin will remain asserted until the exact number of bytes specified by RBL1 and RBL2 has been transferred, or if any of the conditions described in the previous paragraph occur.

RBL1, 2: Remote Burst Length

LBL2	LBL1	
0	0	1 word (2 byte)
0	1	4 word (8 byte)
1	0	8 word (16 byte)
1	1	12 word (24 byte)

REMOTE DATA BYTE COUNT (L, H)

Allowable Value 0–64k Hex Address (1A, 1B) READ/WRITE

This pair of registers specifies the number of bytes in one remote transfer using the 16-bit address of the remote DMA channel. In the non-tracking mode, the remote DMA can transfer 1–64k bytes independent of the local DMA. Loading both registers with zero will be interpreted as a 64k byte count. These registers are ignored in tracking mode.

DMA ADDRESS BYTE 0–3

Allowable Value 0–255 Hex Address (1C–1F) READ/WRITE

These address bytes are configured dependent on the current DMA mode. In *32-bit mode*, all four bytes form the physical address with 1F containing the most significant byte. In *16-bit mode*, bytes 0 and 1 form the low and high bytes of the local DMA channel, and bytes 2 and 3 form the low and high of the remote DMA channel, if enabled.

DMA SECTOR COUNTER (DSC)

Hex Address (37) Read Only

This counter is only valid during tracking mode and holds the difference between the number of sectors transferred by the local and remote DMA channels. In tracking mode, when DSC = 0, remote transfer is disabled in a disk read operation so invalid data is not exchanged between local and host memory. This is a dual function register, sharing operation with the Remote Transfer (RT) register described earlier in this section.

3.3 FORMAT REGISTERS

The disk format is defined by using the format pattern and control registers. Generally, these registers are set up in pairs. In each pair, one register is loaded with an appropriate 8-bit pattern that will be written to the disk during a Format or Write command, or will be used during a Read or Compare command for byte alignment or a comparison in locating a sector. Refer to *Figure 4*, below, for a listing of the format registers, and the manner in which they are paired. The FORMAT, READ AND WRITE Section contains a listing and description of each of the format fields.

The other register in the pair is used to control the use of the corresponding pattern register. These Byte Count registers are loaded with a 5-bit binary number indicating the number of times the associated pattern will be repeated, therefore defining the size of that particular field (0–31 bytes). The Gap Byte Count register is the only one with 8 bits, allowing a field of up to 255 bytes in length.

The External ECC Count registers do not perform any pattern repetition. The external ECC appendage is provided from outside the DDC, and must be fit into the field whose length is defined by these registers (0–31 bytes). If any field is to be excluded from the disk format, the Byte Count register associated with that field must be loaded with zero. This is particularly important with the External ECC Byte Count registers. If these are non-zero, the EXT STAT pin will expect a pulse for each external ECC field during a Read operation. If these pulses are not supplied, the operation will be aborted in an error condition. Also, no more than two consecutive format fields may be deleted at one time.

The Header Byte Control registers also do not perform any pattern repetition, nor do they define field size. They are provided for controlling the function of each corresponding header byte.

HEADER CONTROL (HC0–5)

Hex Address (24–29) Read/Write

There is one HC register for each of six Header Byte pattern registers.

NU	NCP	EHF	SSC	HBA
4	3	2	1	0

HBA: Header Byte Active

0 The corresponding Header Byte is not included in the header byte field and will not be used in the ID operation. All other bits in each HC register in which this bit is set to zero must also be set to zero. A minimum of two Header Bytes must be enabled out of six, with no more than two disabled consecutively.

1 The corresponding Header Byte contains valid data and will be used in the ID operation.

SSC: Substitute Sector Counter

0 The corresponding Header Byte as stored in the pattern register is directly written to the disk for a Write Header command, and will be compared for Compare Header command.

1 The contents of the Sector Counter (SC) are substituted for this Header Byte during a Write Header command and compared during a Compare Header command. This is normally used in multisector operations.

EHF: Enable HFASM Function

See HFASM function description in ADDITIONAL FEATURES.

0 HFASM function is disabled.

1 HFASM function is enabled. The corresponding Header Byte is designated as that byte that must match in order to generate an HFASM error, typically the sector number.

NCP: Not Compare

0 The corresponding Header Byte will be compared normally.

1 A valid comparison will always be assumed, regardless of the true outcome.

NU: Not Used

This bit must be set to zero. If set to 1 unspecified operations may occur.

3.0 Internal Registers of the DDC (Continued)

Pattern Register	Hex Addr	Pattern Source	Control Function	Hex Addr	Control Register
ID Preamble	31	*Internal*	*Repeat 0–31x*	21	ID Preamble Byte Count
ID Synch #1 (AM)	32			22	ID Synch #1 (AM) Byte Count
ID Synch #2	33			23	ID Synch #2 Byte Count
Header Byte 0	14		*Define/Control*	24	Header Byte 0 Control
Header Byte 1	15			25	Header Byte 1 Control
Header Byte 2	16			26	Header Byte 2 Control
Header Byte 3	17			27	Header Byte 3 Control
Header Byte 4	18			28	Header Byte 4 Control
Header Byte 5	19			29	Header Byte 5 Control
ID External ECC	*	*External*	*0–31 Bytes*	2B	ID External ECC Byte Count
ID Postamble	3C	*Internal*	*Repeat 0–31x*	2C	ID Postamble Byte Count
Data Preamble	3D			2D	Data Preamble Byte Count
Data Synch #1 (AM)	3E			2E	Data Synch #1 (AM) Byte Count
Data Synch #2	3F			2F	Data Synch #2 Byte Count
Data Format	3B		*Field Size*	38	Sector Byte Count L
			1–64k Bytes	39	Sector Byte Count H
Data External ECC	*	*External*	*0–31 Bytes*	2A	Data External ECC Byte Count
Data Postamble	30	*Internal*	*Repeat 0–31x*	20	Data Postamble Byte Count
Gap	3A		*Repeat 0–255x*	34	Gap Byte Count

*These are not pattern registers.

FIGURE 4. Format Registers

3.4 CRC/ECC REGISTERS

The following registers are for programming and controlling the CRC/ECC functions of the DDC. Many of these registers have dual functions, depending on whether they are being written to or read from. Take care in noting which these are, to avoid confusion later. Only a basic functional description of these are provided here. Detailed instructions on their use can be found in the CRC/ECC section.

ECC SR OUT 0–5 Hex Address (02–07) Read Only

The syndrome bytes for performing a correction are available from these registers, and are externally XOR'ed with the errored data bytes. These are dual function registers, sharing operation with the Polynomial Preset Bytes.

POLYNOMIAL PRESET BYTES 0–5 (PPB0–5)

Hex Address (02–07) Write Only

The ECC shift registers can be preset by loading a bit pattern into these registers. These are dual function registers, sharing operation with the ECC SR Out registers.

POLYNOMIAL TAP BYTES (PTB0–5)

Hex Address (08–0D) Write Only

These registers are used for programming the taps for the internal 32 or 48-bit ECC polynomial. PTB0 and PTB1 are dual function registers, sharing operation with the Data Byte Counters.

DATA BYTE COUNTER 0, 1 (LS, MS)

Hex Address (08, 09) Read Only

The Data Byte Counters indicate the location of the byte in error after an ECC cycle. These are dual function registers, sharing operation with the Polynomial Tap Bytes 0 & 1. The Sector Byte Count Register must be reloaded with the sector length plus the number of ECC bytes before the start of a correction cycle. If the CF bit in the Error register is reset after a correction, the Data Byte Counter will contain an offset pointing to the first byte in error.

ECC/CRC Control (EC)

Hex Address (OE) Write Only

DNE	IDI	IEO	HNE	CS3	CS2	CS1	CS0
7	6	5	4	3	2	1	0

CS0–CS3: Correction Span Selection Bits

These four bits program the number of bits that the ECC circuit will attempt to correct. Errors longer than the correction span will be treated as non-correctable. The allowable correction span is 3–15 bits. If a span outside this range is loaded, the DDC will automatically default to a span of three bits.

For example, a five bit correction span would load as:

CS3	CS2	CS1	CS0
0	1	0	1

HNE: Header Non-Encapsulation

0 Header address mark and/or synch fields are encapsulated in the CRC/ECC calculation.

1 Header address mark and/or synch fields are not encapsulated in the CRC/ECC calculation.

NOTE: The SAM bit in the DF register must be reset when performing a Compare or Read Header operation, and the HNE bit is active low. If this is not done, the CRC/ECC calculation will begin at the synch word of the header, resulting in a Header Fault that will abort a Read operation or a Sector Not Found error for a Compare Header operation.

IEO: Invert ECC Out

See note under IDI bit, below.

0 Checkbits exiting ECC/CRC shift register are unaltered.

1 Checkbits exiting ECC/CRC shift register are inverted.

3.0 Internal Registers of the DDC (Continued)

IDI: Invert Data In

0 Data and checkbits entering the ECC/CRC shift register are unaltered.

1 Data and checkbits entering the ECC/CRC shift register are inverted.

NOTE: This inversion option has been included for compatability with a few systems that require ECC input and/or output inversion.

DNE: Data Non-Encapsulation

0 Data address mark and/or synch fields are encapsulated in the CRC/ECC calculation.

1 Data address mark and/or synch fields are not encapsulated in the CRC/ECC calculation.

4.0 DDC Operation

4.1 MICROPROCESSOR ACCESS

The DDC requires microprocessor control to initiate operations and commands, and to check chip status. All registers in the DDC appear as unique memory or I/O locations. Each can be randomly accessed and operated on. When the DMA is not performing a memory transfer, the chip can be accessed as a memory location or standard I/O port. Only eight bits of data may be transferred at this time, using pins AD0–7 (the upper 8 bits of a 16 bit microprocessor are not used). Six dedicated address pins (RS0–5) individually select all of the DDC's internal registers. By using these dedicated lines with an address strobe input (ADS0), the chip can be used in both multiplexed and demultiplexed address bus environments. The ADS0 and RS0–5 pins operate as a fall through type latch. By asserting CS active low, the DDC recognizes it has to be a slave and allows RD and WR to effect the internal registers. With multiplexed address and data lines, a positive strobe pulse on ADS0 will latch the address. The ADS0 line may be derived from a microprocessor address strobe such as ALE. In systems with a dedicated address bus (demultiplexed), ADS0 may be pulled high to allow address information to flow through the latch. Finally, by applying CS and a RD or WR strobe, any of the 64 internal locations can be accessed. It is important to note that most registers are read or write only. Some registers, however, change function dependent on whether they are being read from or written to (see Dual Function register list in INTERNAL REGISTERS).

4.2 OPERATING MODES

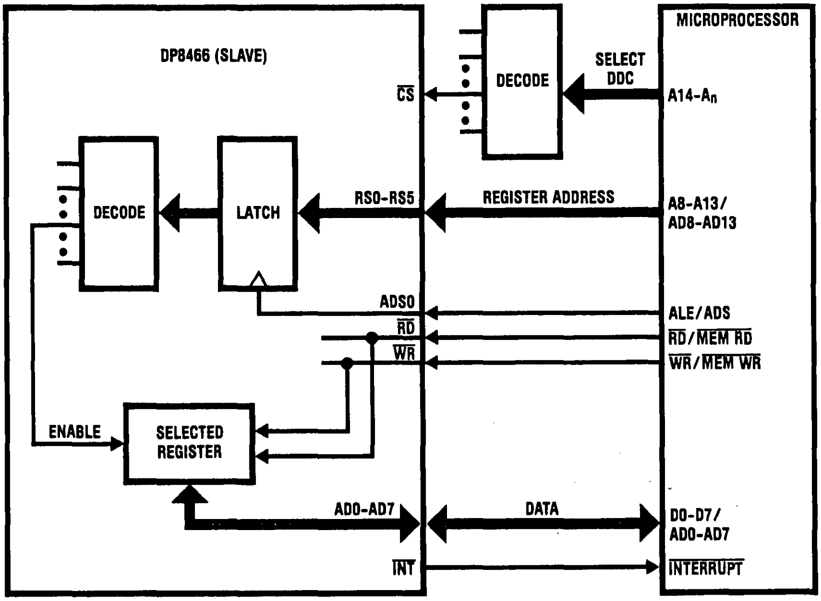
The DDC can be thought of as operating in four modes: *RESET, COMMAND ACCEPT, COMMAND PERFORM* and *ERROR*. These modes are given here in order to provide a functional operating description of the DDC, particularly when an error has been encountered.

Mode 1 *RESET*: All functions are stopped, and no command can be issued. During power up and before initalization, the DDC is held in this mode. To leave this mode, pin 24 (RESET) must be high, a 0 must be written to the RES location in the OC register, and a RED command loaded into the DC register. This places the DDC into MODE 2.

Mode 2 *COMMAND ACCEPT*: The DDC is free and ready to receive the next command (NDC bit set in Status register). Upon receipt of a command, the DDC will enter MODE 3.

Mode 3 *COMMAND PERFORM*: The directed operation is performed. If no error is encountered, the DDC will return to MODE 2. An error will put the DDC into MODE 4.

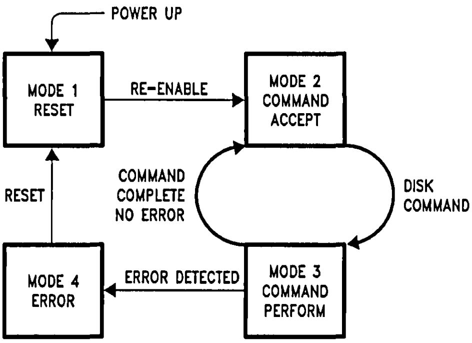
Mode 4 *ERROR*: The error needs to be serviced, and then the DDC can be reset by MODE 1.



TL/F/5282-4

FIGURE 5. Microprocessor Access to DP8466

4.0 DDC Operation (Continued)

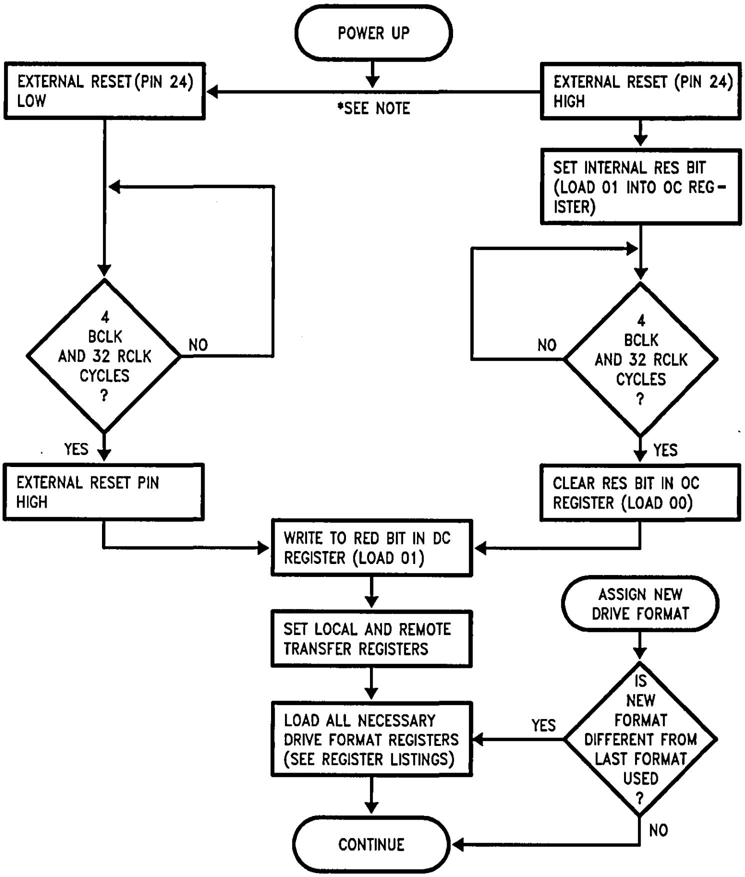


TL/F/5282-5

FIGURE 6. DDC Operating Modes

4.3 POWER UP AND INITIALIZATION

In powering up the DDC, the counters and registers must be initialized before a drive can be assigned and the appropriate information loaded. This can be done by either holding pin 24 (RESET) low, or by setting the internal RES bit in the OC register. Both require that the DDC be held in the reset condition for a minimum of 32 RCLK periods and 4 BCLK periods before the reset condition can be cleared. *Figure 7* shows a general algorithm for both methods. After power up, and whenever a new drive is assigned, the appropriate drive format registers need to be loaded before any drive operation is performed.


POWER UP
EXTERNAL RESET (PIN 24) LOW
*SEE NOTE
EXTERNAL RESET (PIN 24) HIGH
SET INTERNAL RES BIT (LOAD 01 INTO OC REGISTER)
4 BCLK AND 32 RCLK CYCLES ?
NO
YES
4 BCLK AND 32 RCLK CYCLES ?
NO
YES
EXTERNAL RESET PIN HIGH
CLEAR RES BIT IN OC REGISTER (LOAD 00)
WRITE TO RED BIT IN DC REGISTER (LOAD 01)
SET LOCAL AND REMOTE TRANSFER REGISTERS
ASSIGN NEW DRIVE FORMAT
IS NEW FORMAT DIFFERENT FROM LAST FORMAT USED ?
YES
LOAD ALL NECESSARY DRIVE FORMAT REGISTERS (SEE REGISTER LISTINGS)
NO
CONTINUE

TL/F/5282-6

Note: As shown various methods are possible for power up, and it is up to the user as to which is more suitable.
The DDC should be reset and RCLK and BCLK should be applied after power up, otherwise it may draw an excessive amount of current, and may cause bus contention.

FIGURE 7. Power Up and Initialization Algorithm

5.0 Format, Read and Write

5.1 DISK FORMATTING

The formatting process is carried out through the format parameter and pattern registers (see FORMAT REGISTERS). These registers should be loaded during the initialization process for the particular drive in use. The pattern registers are loaded with the specific 8-bit pattern to be written to the disk. The count registers specify the number of times each 8-bit pattern is to be written. In loading these registers, several things need to be kept in mind:

- If any byte count register is loaded with zero, that field will be excluded, and no pattern for the corresponding pattern register need be loaded.
- At least two header bytes must be used, with no more than two consecutive unused header bytes. This also applies to all the fields in the format, where no more than two consecutive fields may be deleted. The one exception is the internal header ECC and external header ECC field. At least one of these fields must be present.
- If the disk is hard sectored, no gap byte count needs to be loaded. See Hard Vs. Soft Sector Operation in the FORMAT, READ AND WRITE Section.

The sector format options that are provided with the DDC are shown in *Figure 8*. The fields common to the ID and data fields, such as the preamble, Synch, CRC/ECC and postamble fields, perform similar functions, and are briefly discussed below.

PREAMBLE: Allows the PLL in the data separator to achieve phase lock.

SYNCH #1 and 2: Synch #1 contains the missing clock address mark for use with soft sectored disks. Generally, this field is not used in hard sectored disks. The synch #1 field can be used to extend the preamble or the synch fields in hard sectored mode. Synch #1 and #2 fields allow for byte alignment of the DDC.

HEADER BYTES: Used to uniquely identify each sector. Examples are sector number, cylinder number, track number, etc.

DATA: Information to be stored.

CRC/ECC: This field is generated and checked internally.

EXT.ECC: Used with external ECC circuitry. Provides space for externally generated ECC bytes.

POSTAMBLE: Allows read gate turn off time for the PLL to unlock. Provides a pad so that the write splice does not occur at the end of the CRC.

GAP 3: Provides protection against speed variation. In soft sectored mode, its length is determined by the Gap Byte Count register. In hard sectored mode, this gap will continue until the next sector pulse.

Format operations always start with an index pulse, and end with the next index pulse, thus making one track. The DDC has three approaches for formating disks:

Internal Sequential
FIFO Table
Interlock Type

INTERNAL SEQUENTIAL

This mode is used where the sector number is incremented for each physically adjacent sector, that is, for an interleave of one. This mode may be used on a multi-sector operation to format a whole track of sequential sectors. The header bytes other than the sector number, such as cylinder number and head number, are loaded. The Sector Counter (SC) is loaded with the first sector number desired on the track and the HC register with SSC=1. The Number of Sector Operations (NSO) counter is loaded with the number of sectors per track. Finally, the FMT bit is set in the DC register in addition to bits for a Write Header/Write Data, multi-sector operation. Formatting begins upon loading the DC register. The last sector number written will therefore be [SC] + [NSO] − 1.

FIFO TABLE

This approach is ideal for sector interleaving and offers the minimum of microprocessor intervention during the format operation. The microprocessor sets up the header bytes of each sector, contiguously in memory. The local DMA channel or external DMA is used to transfer the header byte sets into the FIFO. Each set transferred is used once for each header field. The local DMA transfers a new set for each sector. The number of sectors transferred is determined by the NSO register.

The format operation follows the sequence below:

(1) Before the format operation, a full track of header byte sets is loaded into a memory area accessible to the local DMA channel. Each header byte set must contain an even number of bytes. If it contains an odd number of bytes, an extra "dummy" byte must be inserted so that each header byte set will be contained in an even byte boundary.

(2) The DMA address is loaded with the location of the first byte of the first header byte set.

ID FIELD

ID PREAMBLE	ID SYNCH #1 (AM)	ID SYNCH #2	HEADER BYTES	ID CRC/ECC*	ID EXT ECC*	ID POSTAMBLE
0–31 Bytes	*0–31 Bytes*	*0–31 Bytes*	*2–6 Bytes*	*0, 2, 4 or 6 Bytes*	*0–31 Bytes*	*0–31 Bytes*

DATA FIELD

DATA PREAMBLE	DATA SYNCH #1 (AM)	DATA SYNCH #2	DATA FORMAT PATTERN	DATA CRC/ECC	DATA EXT ECC	DATA POSTAMBLE	GAP 3
0–31 Bytes	*0–31 Bytes*	*0–31 Bytes*	*1–64k Bytes*	*0, 2, 4 or 6 Bytes*	*0–31 Bytes*	*0–31 Bytes*	*0–255 Bytes*

*Note the ID CRC/ECC field and the ID EXT ECC field must not be set to zero simultaneously.

FIGURE 8. Sector Format Fields

5.0 Format, Read & Write (Continued)

(3) The Header Byte Count (HBC) is loaded with the number of header bytes in each sector (2–6 bytes).

(4) The Disk Format (DF) register is loaded with the FTF bit set.

(5) The Drive Command (DC) register is loaded for a Write Header/Write Data, multi-sector, format operation.

INTERLOCK TYPE

This approach offers the most versatility, but requires fast microprocessor intervention. It may be used to format a whole track of interleaved sectors. It can also be used for creating files of varying sector length, but this can be very tricky. The DDC can format sectors with data lengths from 1 to 64k bytes with single byte resolution.

Interlock type formatting uses the interlock mode and the header complete interrupt to enable the microprocessor to directly update any format parameter bytes. The Operation Command (OC) register is loaded wth IR (Interlock Mode), EHI and EI bits set. The Disk Format (DF) register should be loaded with the FTF bit reset. The header byte pattern for each selected header byte must be loaded into the relevant register. The NSO register is loaded with the number of sectors to be formatted. The DC register is then loaded for a Write Header/Write Data, multi-sector, format operation.

After the header field is written in the first sector, the DDC issues the header complete interrupt. With interlock mode set, the controlling microprocessor has the block of time until the preamble field of the next sector to read status, load the next sector's header bytes into the DDC registers and confirm this had been accomplished by writing to the Interlock (HBC) register. This must be done after the HMC interrupt for every sector, including the last sector of the operation. If this is not done, a Late Interlock error will occur when a subsequent command is loaded in the DC register.

In a non-format operation, the user has only until the end of the data field to write to the HBC register (see Data Recovery Using The Interlock Feature in ADDITIONAL FEATURES). This operation is repeated until the NSO register decrements to zero. An interrupt will then be issued indicating that the operation has completed.

5.2 READ AND WRITE

For initiating Read/Write operations, the necessary format registers need to be loaded with the appropriate information to enable the DDC to identify the desired sector. Multi-sector operations will also require the Number of Sector Operations (NSO) counter and the Sector Counter (SC). Algorithms outlining the read/write operations are shown in *Figures 10* and *11*. For each of these, it is assumed that the parameters for the desired sector(s) have been loaded, and that the head is positioned over the proper track.

READ

During a read operation, header data passing under the disk head is compared to the header bytes in the DDC parameter RAM. If a match is found after a read command is issued, the data field of the identified sector will start filling the FIFO. Once the selected threshold data level (burst length) is reached, the Local DMA Request (LRQ) pin will be asserted, signaling that a transfer is required. When the LACK pin grants the bus, either the exact burst length or the entire FIFO contents are transferred to memory. The FIFO continues filling, and this process repeats until the entire data field has been transferred to memory.

WRITE

A similar process occurs in reverse for a write operation. The DMA fills the FIFO, and when the correct sector is found, this data begins to be written to disk. When the data in the FIFO falls by an amount equal to the burst length, a transfer request is issued on LRQ. When LACK is granted, the DMA either fills the FIFO or transfers the exact number of bytes specified in the burst length. This process continues until a number of bytes specified by the Sector Byte Count register has been written to the disk.

Multi-sector operations follow the same procedure, but the operation is repeated on the number of sectors specified in the Number of Sector Operations (NSO) counter, with an interrupt being generated on completion of the last sector.

5.3 HARD SECTOR vs. SOFT SECTOR OPERATION

The choice between hard and soft sectored operation is made through the use of the HSS bit in the Drive Format register. This bit, in conjunction with other control bits can set the DDC to perform a number of functions depending on whether a read, write or format operation is to be enacted. HSS = 0 sets the DDC for soft sectored operation, and HSS = 1 sets the DDC for hard sectored operation.

FORMAT

In hard sectored operation, the DDC assumes that sector pulses are present, and will ignore the gap count. Gap bytes will be written until a pulse is detected on the SECTOR pin.

In soft sectored operation, the gap count will be used for every sector except the last. The Gap Byte Count register determines the Gap 3 length. For the last sector, gap bytes will be written until an index pulse is received.

READ

When reading, the need for the AMF input pulse is determined by the HSS bit. For soft sectoring, the AMF input is required for at least one bit time within the Synch #1 fields in both the ID and Data sections of the sector. For hard sectoring, the AMF input is not required.

The HSS bit in the DF register, and the SAIS command in the DC register define when RGATE is asserted for various sector formats. This is outlined below.

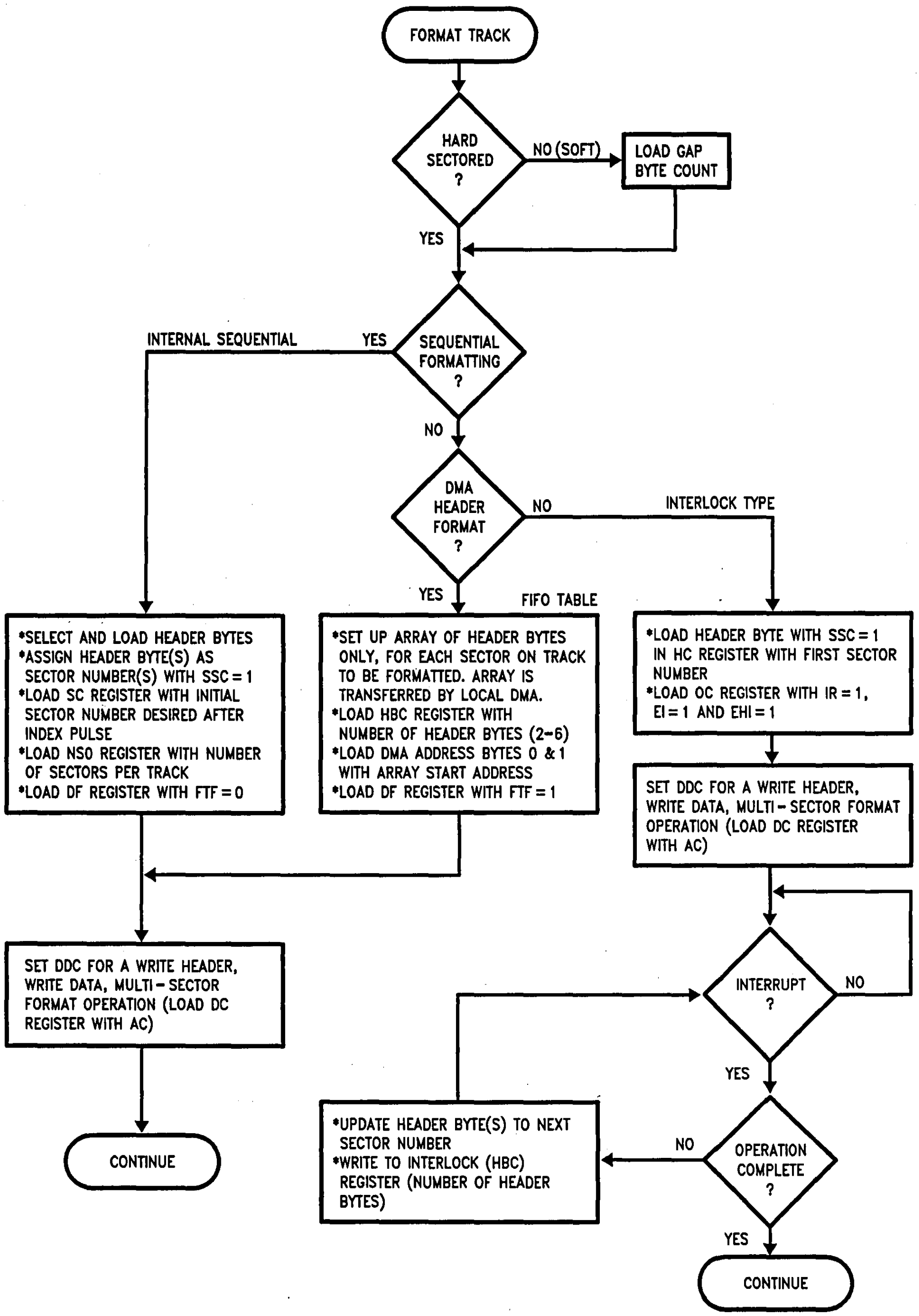
HSS	SAIS	RGATE ASSERTED:
0	0	On index pulse
0	1	On receipt of instruction
1	0	On index pulse
1	1	On index or sector pulse

WRITE

The HSS, MFM and SAM bits in the DF register determine the use of the address mark and the AME pin as follows:

HSS	MFM	SAM	FUNCTION
0	0	0	AME pin activated during ID and data synch #1 fields.
X	0	1	AME pin activated during ID preamble.
0	1	X	Missing clocks inserted in ID and data synch #1 fields. AME pin indicates LPRE (if enabled).
1	0	0	AME pin disabled.
1	1	X	Synch #1 fields written without missing clock pulse.

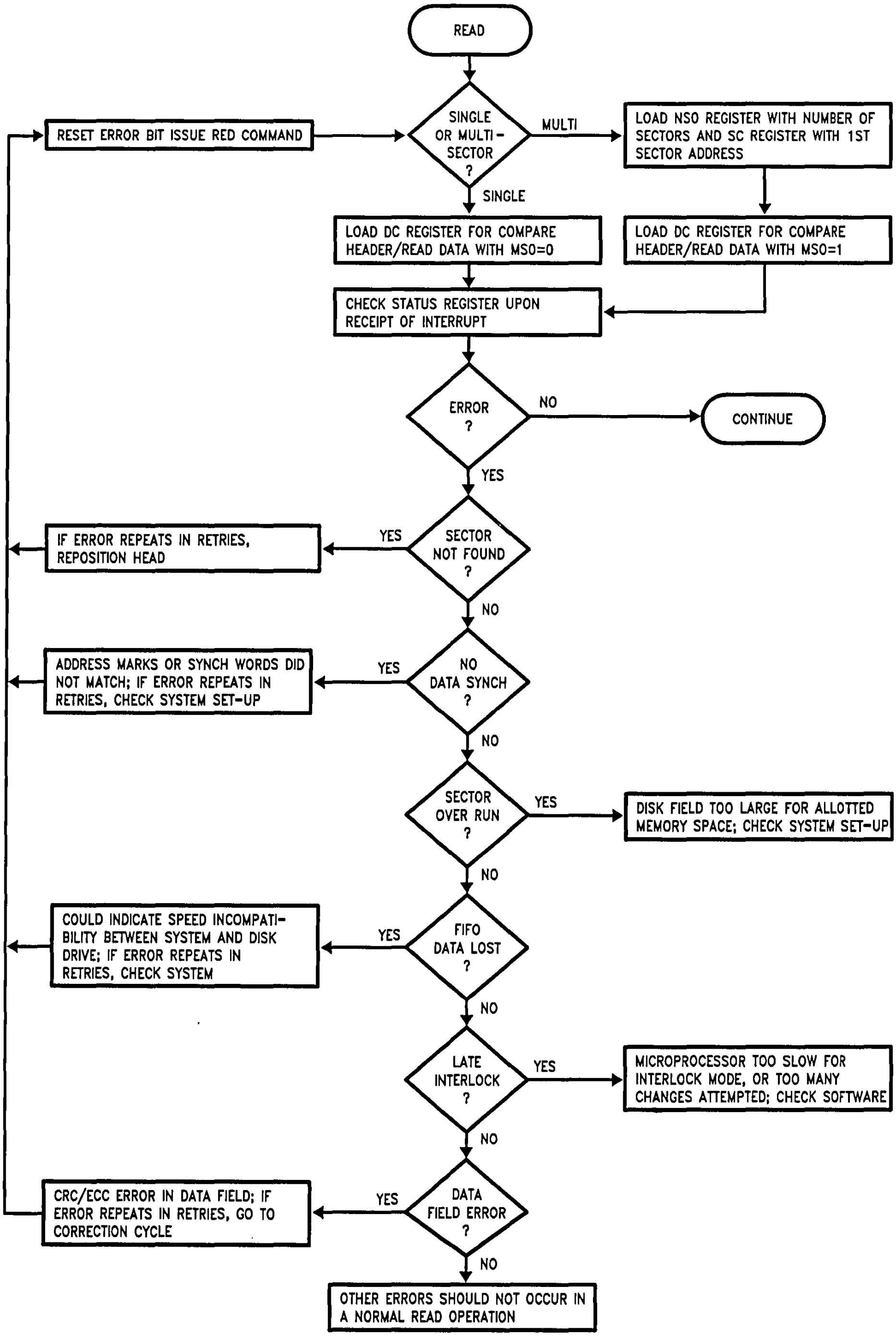
5.0 Format, Read & Write (Continued)


FORMAT TRACK
HARD SECTORED ?
NO (SOFT)
LOAD GAP BYTE COUNT
YES
SEQUENTIAL FORMATTING ?
INTERNAL SEQUENTIAL
YES
NO
DMA HEADER FORMAT ?
NO
INTERLOCK TYPE
YES
FIFO TABLE
•SELECT AND LOAD HEADER BYTES
•ASSIGN HEADER BYTE(S) AS SECTOR NUMBER(S) WITH SSC = 1
•LOAD SC REGISTER WITH INITIAL SECTOR NUMBER DESIRED AFTER INDEX PULSE
•LOAD NSO REGISTER WITH NUMBER OF SECTORS PER TRACK
•LOAD DF REGISTER WITH FTF = 0
•SET UP ARRAY OF HEADER BYTES ONLY, FOR EACH SECTOR ON TRACK TO BE FORMATTED. ARRAY IS TRANSFERRED BY LOCAL DMA.
•LOAD HBC REGISTER WITH NUMBER OF HEADER BYTES (2–6)
•LOAD DMA ADDRESS BYTES 0 & 1 WITH ARRAY START ADDRESS
•LOAD DF REGISTER WITH FTF = 1
•LOAD HEADER BYTE WITH SSC = 1 IN HC REGISTER WITH FIRST SECTOR NUMBER
•LOAD OC REGISTER WITH IR = 1, EI = 1 AND EHI = 1
SET DDC FOR A WRITE HEADER, WRITE DATA, MULTI - SECTOR FORMAT OPERATION (LOAD DC REGISTER WITH AC)
SET DDC FOR A WRITE HEADER, WRITE DATA, MULTI - SECTOR FORMAT OPERATION (LOAD DC REGISTER WITH AC)
INTERRUPT ?
NO
YES
CONTINUE
•UPDATE HEADER BYTE(S) TO NEXT SECTOR NUMBER
•WRITE TO INTERLOCK (HBC) REGISTER (NUMBER OF HEADER BYTES)
NO
OPERATION COMPLETE ?
YES
CONTINUE

TL/F/5282-7

FIGURE 9. Format Track Algorithm

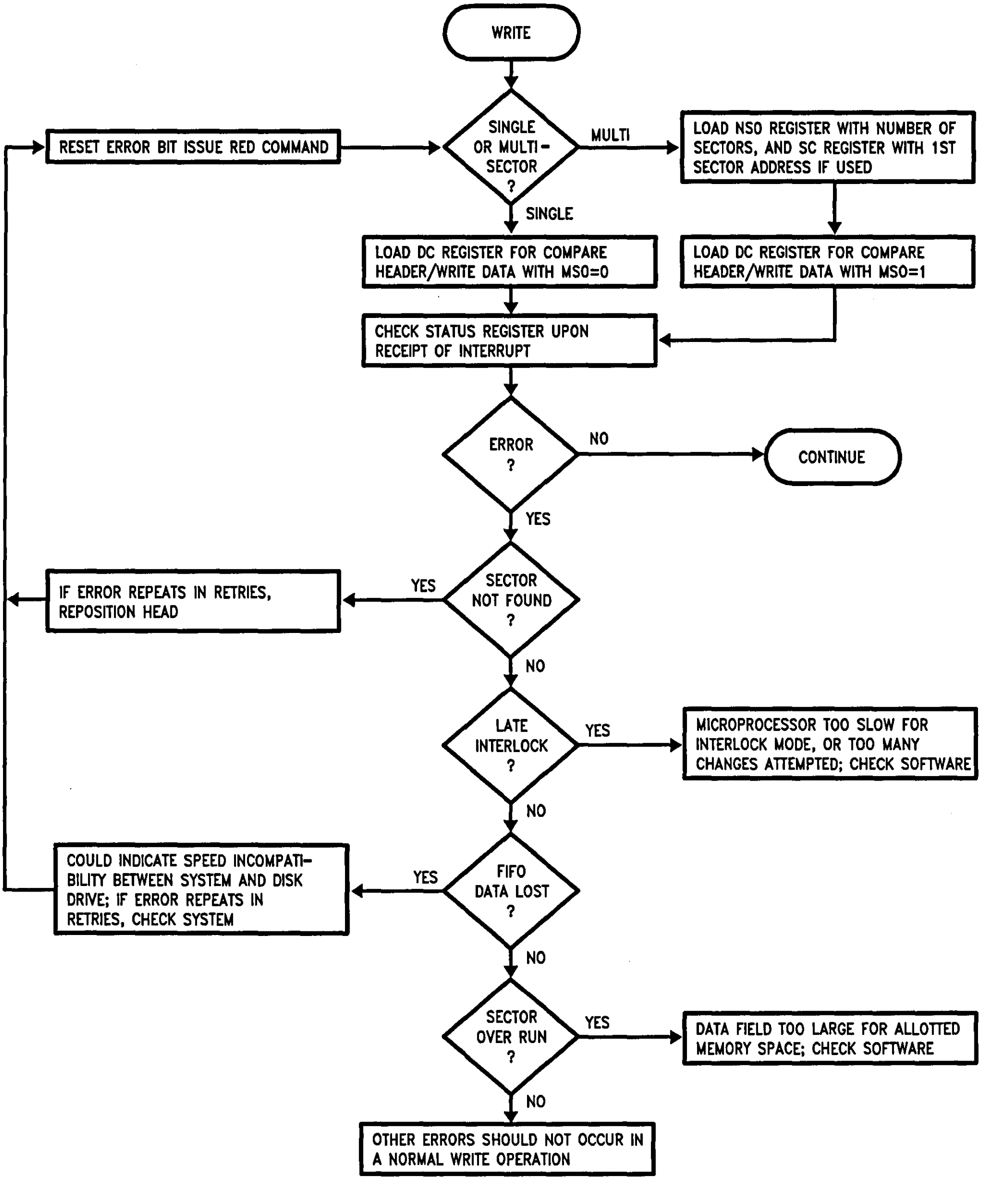
5.0 Format, Read & Write (Continued)


READ
SINGLE OR MULTI-SECTOR ?
MULTI
LOAD NSO REGISTER WITH NUMBER OF SECTORS AND SC REGISTER WITH 1ST SECTOR ADDRESS
RESET ERROR BIT ISSUE RED COMMAND
SINGLE
LOAD DC REGISTER FOR COMPARE HEADER/READ DATA WITH MSO=0
LOAD DC REGISTER FOR COMPARE HEADER/READ DATA WITH MSO=1
CHECK STATUS REGISTER UPON RECEIPT OF INTERRUPT
ERROR ?
NO
CONTINUE
YES
SECTOR NOT FOUND ?
YES
IF ERROR REPEATS IN RETRIES, REPOSITION HEAD
NO
NO DATA SYNCH ?
YES
ADDRESS MARKS OR SYNCH WORDS DID NOT MATCH; IF ERROR REPEATS IN RETRIES, CHECK SYSTEM SET-UP
NO
SECTOR OVER RUN ?
YES
DISK FIELD TOO LARGE FOR ALLOTTED MEMORY SPACE; CHECK SYSTEM SET-UP
NO
FIFO DATA LOST ?
YES
COULD INDICATE SPEED INCOMPATIBILITY BETWEEN SYSTEM AND DISK DRIVE; IF ERROR REPEATS IN RETRIES, CHECK SYSTEM
NO
LATE INTERLOCK ?
YES
MICROPROCESSOR TOO SLOW FOR INTERLOCK MODE, OR TOO MANY CHANGES ATTEMPTED; CHECK SOFTWARE
NO
DATA FIELD ERROR ?
YES
CRC/ECC ERROR IN DATA FIELD; IF ERROR REPEATS IN RETRIES, GO TO CORRECTION CYCLE
NO
OTHER ERRORS SHOULD NOT OCCUR IN A NORMAL READ OPERATION

TL/F/5282-8

FIGURE 10. Simple Read Operation

5.0 Format, Read & Write (Continued)


WRITE
SINGLE OR MULTI-SECTOR ?
MULTI
LOAD NSO REGISTER WITH NUMBER OF SECTORS, AND SC REGISTER WITH 1ST SECTOR ADDRESS IF USED
RESET ERROR BIT ISSUE RED COMMAND
SINGLE
LOAD DC REGISTER FOR COMPARE HEADER/WRITE DATA WITH MSO=0
LOAD DC REGISTER FOR COMPARE HEADER/WRITE DATA WITH MSO=1
CHECK STATUS REGISTER UPON RECEIPT OF INTERRUPT
ERROR ?
NO
CONTINUE
YES
SECTOR NOT FOUND ?
YES
IF ERROR REPEATS IN RETRIES, REPOSITION HEAD
NO
LATE INTERLOCK ?
YES
MICROPROCESSOR TOO SLOW FOR INTERLOCK MODE, OR TOO MANY CHANGES ATTEMPTED; CHECK SOFTWARE
NO
FIFO DATA LOST ?
YES
COULD INDICATE SPEED INCOMPATIBILITY BETWEEN SYSTEM AND DISK DRIVE; IF ERROR REPEATS IN RETRIES, CHECK SYSTEM
NO
SECTOR OVER RUN ?
YES
DATA FIELD TOO LARGE FOR ALLOTTED MEMORY SPACE; CHECK SOFTWARE
NO
OTHER ERRORS SHOULD NOT OCCUR IN A NORMAL WRITE OPERATION

TL/F/5282–9

FIGURE 11. Simple Write Operation

5.0 Format, Read & Write (Continued)

5.4 MFM ENCODED DATA

MFM encoding of write data is controlled by the MFM bit in the DF register MFM = 1 sets the DDC to write MFM data to the disk. MFM = 0 sets the DDC to write NRZ data to the disk.

PRECOMPENSATION OF MFM ENCODED DATA

When the MFM bit in the DF register and the EP bit in the OC register are set, precompensation will be indicated on the EPRE and LPRE pins. Precompensation is issued for the middle bit of a 5-bit field. In the DP8466, early and late precompensation will be enacted for all of the combinations as shown below. All other patterns will not require precompensation. Precompensation can be disabled by setting the EP bit in the OC register inactive low.

EPRE NRZ PATTERNS	LPRE NRZ PATTERNS
00 0 10	00 1 10
00 0 11	00 1 11
01 1 00	10 0 00
01 1 01	10 0 01
11 1 00	10 1 10
11 1 01	10 1 11

Precompensation outputs are aligned to provide symmetrical set-up and hold times relative to the rising edge of the WDATA outputs. This gives a half period of RCLK set-up time on precompensation outputs. This is shown in *Figure 12.* Two bits of zero precede the preamble fields at the leading edge of the write gate when writing MFM data due to MFM encoded delays.

5.5 ADDRESS MARK PATTERNS, MISSING CLOCK

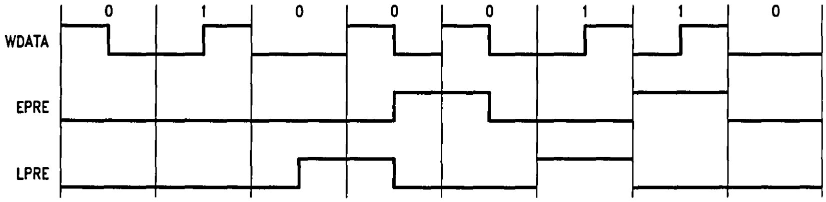
During writing and formatting a sector with MFM encoding enabled, a clock violation, or missing clock pulse, will be inserted in the synch #1 field. This indicates the address mark. For an example of this, refer to *Figure 13.*

When writing MFM encoded data with precompensation enabled, only the following hex values are allowed to be loaded into the synch #1 pattern registers:

A1, C2, C3, E1, 84, 85, 86, 87

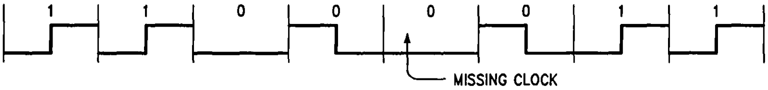
With no precompensation, any pattern containing 100001 is valid.

During a soft sectored read operation, an AMF pulse will be expected on the AMF/EPRE pin during each byte of the synch #1 field.



TL/F/5282-10

FIGURE 12. Example of EPRE and LRPE Outputs



TL/F/5282-11

FIGURE 13. Missing Clock Example

6.0 CRC/ECC

6.1 PROGRAMMING CRC

The DDC is set for internal CRC by programming the disk Format (DF) and ECC/CRC Control (EC) registers. The CRC-CCITT polynomial used by the DDC for the CRC code is given below:

$$P(x) = x^{16} + x^{12} + x^5 + 1$$

The DDC uses the pattern preset to all 1's for the CRC calculation. *Note:* If no CRC/ECC is used for the ID fields, an external ECC must be used.

6.2 PROGRAMMING ECC

There are two sets of six registers used to program the ECC. One set of six is used to program the polynomial taps, while the other set is used to establish a preset pattern (typically all 1's). Bits contained in the ECC Control (EC) register are used to control the correction span. The DF register contains bits for choosing the desired type of appendage: Either 32 or 48-bit programmable ECC polynomials, or the 16-bit CCITT CRC polynomial is possible.

PROGRAMMING POLYNOMIAL TAPS

To program a polynomial into the shift register, each tap position used in the code must be set to 0, and all unused taps should be set to 1. The bit assignment for these registers in 48 and 32-bit modes is shown in the tables that follow. It is important that for 32-bit codes, PTB2 and PTB3 all be set to 1's. Failure to do so will result in improper operation. Also, x^{48} and x^{32} are implied, i.e., a 32-bit ECC will always contain the x^{32} term and a 48-bit ECC will always contain the x^{48} term. For both ECC's, the term x^0 (or 1) is also implied, even though this bit is accessible.

Tap Assignment 48-Bit Mode

REG #	ADDR	*BIT NUMBER*							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PTB0	*08*	x^7	x^6	x^5	x^4	x^3	x^2	x^1	x^0
PTB1	*09*	x^{15}	x^{14}	x^{13}	x^{12}	x^{11}	x^{10}	x^9	x^8
PTB2	*0A*	x^{23}	x^{22}	x^{21}	x^{20}	x^{19}	x^{18}	x^{17}	x^{16}
PTB3	*0B*	x^{31}	x^{30}	x^{29}	x^{28}	x^{27}	x^{26}	x^{25}	x^{24}
PTB4	*0C*	x^{39}	x^{38}	x^{37}	x^{36}	x^{35}	x^{34}	x^{33}	x^{32}
PTB5	*0D*	x^{47}	x^{46}	x^{45}	x^{44}	x^{43}	x^{42}	x^{41}	x^{40}

Tap Assignment 32-Bit Mode

REG #	ADDR	*BIT NUMBER*							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PTB0	*08*	x^7	x^6	x^5	x^4	x^3	x^2	x^1	x^0
PTB1	*09*	x^{15}	x^{14}	x^{13}	x^{12}	x^{11}	x^{10}	x^9	x^8
PTB2	*0A*	1	1	1	1	1	1	1	1
PTB3	*0B*	1	1	1	1	1	1	1	1
PTB4	*0C*	x^{23}	x^{22}	x^{21}	x^{20}	x^{19}	x^{18}	x^{17}	x^{16}
PTB5	*0D*	x^{31}	x^{30}	x^{29}	x^{28}	x^{27}	x^{26}	x^{25}	x^{24}

PROGRAMMING PRESET PATTERN

To program the preset pattern that the shift registers will be preset to, PPB0–PPB5 must be initialized. As in the polynomial taps, x^{48}, x^{32}, and x^0 are implied. The assignment of the bits for 48 and 32 bit modes is shown in the tables on the following pages.

The value programmed into each register will be the preset pattern for the eight bits of the corresponding shift register. For typical operation, these will be programmed to all 1's. All unused presets must be set to 0. In 32-bit mode, PPB2 and PPB3 must be set to all 0's. Failure to do so will result in improper operation.

Preset Bit Assignment 48-Bit Mode

REG #	ADDR	*BIT NUMBER*							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PPB0	*02*	x^7	x^6	x^5	x^4	x^3	x^2	x^1	x^0
PPB1	*03*	x^{15}	x^{14}	x^{13}	x^{12}	x^{11}	x^{10}	x^9	x^8
PPB2	*04*	x^{23}	x^{22}	x^{21}	x^{20}	x^{19}	x^{18}	x^{17}	x^{16}
PPB3	*05*	x^{31}	x^{30}	x^{29}	x^{28}	x^{27}	x^{26}	x^{25}	x^{24}
PPB4	*06*	x^{39}	x^{38}	x^{37}	x^{36}	x^{35}	x^{34}	x^{33}	x^{32}
PPB5	*07*	x^{47}	x^{46}	x^{45}	x^{44}	x^{43}	x^{42}	x^{41}	x^{40}

Preset Bit Assignment 32-Bit Mode

REG #	ADDR	*BIT NUMBER*							
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
PPB0	*02*	x^7	x^6	x^5	x^4	x^3	x^2	x^1	x^0
PPB1	*03*	x^{15}	x^{14}	x^{13}	x^{12}	x^{11}	x^{10}	x^9	x^8
PPB2	*04*	0	0	0	0	0	0	0	0
PPB3	*05*	0	0	0	0	0	0	0	0
PPB4	*06*	x^{23}	x^{22}	x^{21}	x^{20}	x^{19}	x^{18}	x^{17}	x^{16}
PPB5	*07*	x^{31}	x^{30}	x^{29}	x^{28}	x^{27}	x^{26}	x^{25}	x^{24}

RECOMMENDED POLYNOMIAL AS AN EXAMPLE

To program the 32-bit polynomial of the form:

$$x^{32} + x^{28} + x^{26} + x^{19} + x^{17} + x^{10} + x^6 + x^2 + 1$$

with a preset of all 1's, a correction span of 5-bits with no header/data encapsulation, the following registers would be programmed as shown. Note that PTB2 and PTB3 must be all 1's and PPB2 and PPB3 must be all 0's in 32-bit mode.

6.0 CRC/ECC (Continued)

Polynomial Taps

REG#	*BIT NUMBER* 7	6	5	4	3	2	1	0
PTB0	1	0	1	1	1	0	1	0
PTB1	1	1	1	1	1	0	1	1
PTB2	1	1	1	1	1	1	1	1
PTB3	1	1	1	1	1	1	1	1
PTB4	1	1	1	1	0	1	0	1
PTB5	1	1	1	0	1	0	1	1

Preset Pattern

REG#	*BIT NUMBER* 7	6	5	4	3	2	1	0
PTB0	1	1	1	1	1	1	1	1
PTB1	1	1	1	1	1	1	1	1
PTB2	0	0	0	0	0	0	0	0
PTB3	0	0	0	0	0	0	0	0
PTB4	1	1	1	1	1	1	1	1
PTB5	1	1	1	1	1	1	1	1

ECC Control Register

BIT#	7	6	5	4	3	2	1	0
SET	1	0	0	1	0	1	0	1

6.3 OPERATION DURING CORRECTION

The DDC can be set to correct an error any time one has been detected and before another operation has begun. The user decides when to initiate the correction. The sector in question can be re-read several times to insure that the error is repeatable. If so, the error can be considered a hard error on the disk and a correction can be attempted. Since the DDC does not contain drive control circuitry, it is the user's responsibility to provide the programming for the execution of any re-read operations and the associated decision making.

The syndrome bytes in the ECC shift register will contain the bit error information. The bytes in error will already have been transferred to memory. Once initiated, the correction is performed internal to the DDC, leaving the bus free for other operations. An interrupt will be issued within the time it takes to read a sector, indicating whether the error was corrected or not. During this time, the erroneous sector in memory will remain unchanged.

Error correction time is determined by the error's location in the sector. The nearer to the start of the sector, the longer the DDC takes to locate the error. This time can be determined using the formula shown at right. It should be noted that this is internal correction time only; more time is required for the microprocessor to perform additional operations.

Before initiating a correction operation, the DDC needs to be reset, and re-enabled (see Operating Modes in DDC OPERATION). The Sector Byte Count registers must be initialized to *[sector length]* + 4 for 32-bit mode or *[sector length]* + 6 for 48-bit mode. The correction command should be issued when the counter has been updated.

The DDC will issue an interrupt after the correction cycle is complete. Other activities (such as completion of remote DMA) may issue interrupts before this happens. These interrupts should be serviced to allow the Correction Cycle Complete interrupt to be issued. The CCA bit in the Status register will be high during the entire correction cycle. It will be reset when the cycle has completed. The ED bit in the Status register will remain active throughout the correction cycle.

If after an interrupt, the Status register is read and the CCA bit is low, the Error register is read to see if the correction was successful. If the CF bit is set, this signifies that the error was non-correctable. This usually means that two errors have occurred with extremities exceeding the selected correction span. Failure to correct an error is serious and the system should be notified that the data from that sector is erroneous.

If the CF bit was not set, the error was corrected. The microprocessor then computes the address of the first byte in the data field that contains the error. That address is: *[current value of DMA Address Bytes 0 & 1] - [Sector Byte Count L & H]* + *[Data Byte Count L & H]* − 1.

Errors are corrected by XOR'ing syndrome bytes (ECC SR Out 0–5) with the bytes in the data record in memory that contain the error. The Data Byte Count can be used to determine whether the error is in the ECC or data field. If the Data Byte Count is greater than the maximum sector length, the error is in the ECC field and no correction should be attempted. If the Data Byte Count is less than the sector length, the error is in the data field (or it may straddle the data and ECC fields) and may be corrected.

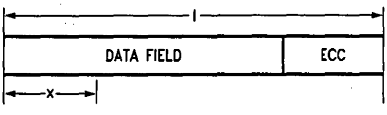
For performing a correction with 32-bit ECC, the following shift registers should be read sequentially to obtain the syndrome byte pattern:

ECC SR Out 1, ECC SR Out 4, ECC SR Out 5

ECC SR Out 2 and 3 are not used in 32-bit mode and will contain 0's if read. ECC SR Out 0 will contain all 0's if the error is correctable, and may contain some set bits if it is not.

ECC SR Out 1 will always contain the first bits in error. The succeeding bits will be contained in ECC SR Out 4 and 5. If the maximum span of 15 bits is used, all three registers may be needed, depending on where the first bit occurs.

To correct the error, the syndrome bits in these registers are XOR'ed with the data bits contained in buffer memory. The corrected data is then written back to the buffer memory, replacing the data in error. The address of the first byte in error is computed by the microprocessor as described above.



TL/F/5282-12

Approximate Correction Time = $(I - x)/f$

I = Entire length of data field and ECC appendage (in bits)

x = Distance from least significant bit to first error location (in bits)

f = read clock frequency (in hertz)

FIGURE 14. Calculating Correction Time

6.0 CRC/ECC (Continued)

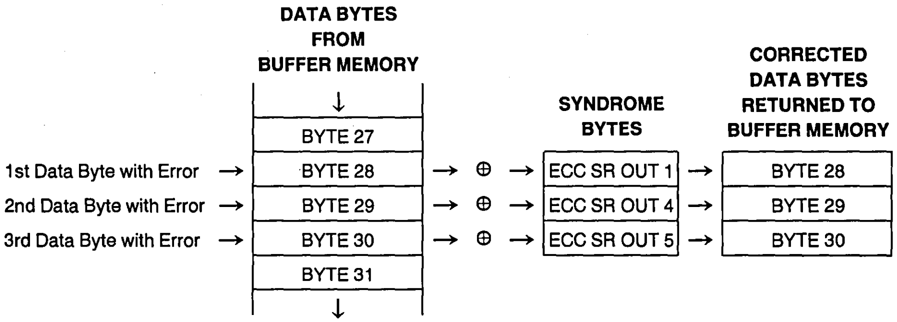

DATA BYTES FROM BUFFER MEMORY
BYTE 27
1st Data Byte with Error → BYTE 28 → ⊕ → ECC SR OUT 1 → BYTE 28
2nd Data Byte with Error → BYTE 29 → ⊕ → ECC SR OUT 4 → BYTE 29
3rd Data Byte with Error → BYTE 30 → ⊕ → ECC SR OUT 5 → BYTE 30
BYTE 31
SYNDROME BYTES
CORRECTED DATA BYTES RETURNED TO BUFFER MEMORY

FIGURE 15. 32-Bit ECC Correction Process

To perform a 48-bit ECC correction, the following registers should be read sequentially:

ECC SR Out 1, ECC SR Out 2, ECC SR Out 3

ECC SR Out 0, 4 and 5 are not used for outputting syndrome bits for correction in 48-bit mode and will contain 0's for a correctable error. If the error is non-correctable, these registers may contain some set bits. Syndrome bit location and error correction is performed as in 32-bit mode.

EXAMPLE OF A 32-BIT CORRECTION

Shown in *Figure 17*, is a record with several bits read in error from disk. Bits D4, D11, D13 and D14, now located in memory, were incorrectly and need to be corrected. As can be seen, the correction pattern provided in ECC SR Out 1 and 2 can be used to correct bits D4, D11, D13 and D14. The CPU reads the Data Byte Count and computes that it points to the first byte read from disk. This byte is XOR'ed with ECC SR Out 1 and is written back to memory. The second byte read from the disk is XOR'ed with ECC SR Out 4 and then written back. ECC SR Out 5 need not be used since it contains all 0's.

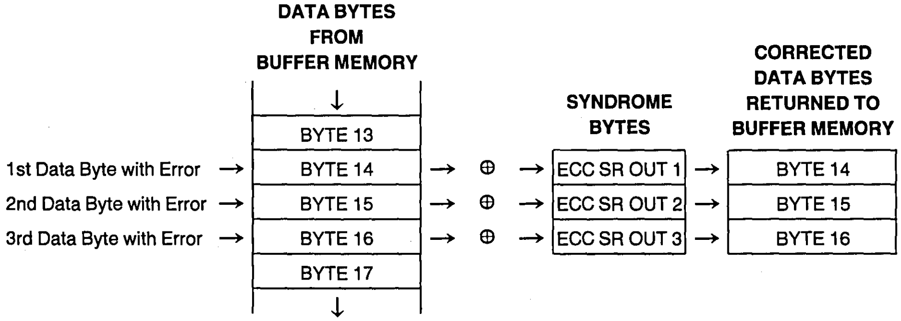

DATA BYTES FROM BUFFER MEMORY
BYTE 13
1st Data Byte with Error → BYTE 14 → ⊕ → ECC SR OUT 1 → BYTE 14
2nd Data Byte with Error → BYTE 15 → ⊕ → ECC SR OUT 2 → BYTE 15
3rd Data Byte with Error → BYTE 16 → ⊕ → ECC SR OUT 3 → BYTE 16
BYTE 17
SYNDROME BYTES
CORRECTED DATA BYTES RETURNED TO BUFFER MEMORY

FIGURE 16. 48-Bit ECC Correction Process

Syndrome Pattern

REGISTER	*BIT NUMBER*							
	7	6	5	4	3	2	1	0
ECC SR OUT 1	0	0	0	1	0	0	0	0
ECC SR OUT 4	0	1	1	0	1	0	0	0
ECC SR OUT 5	0	0	0	0	0	0	0	0

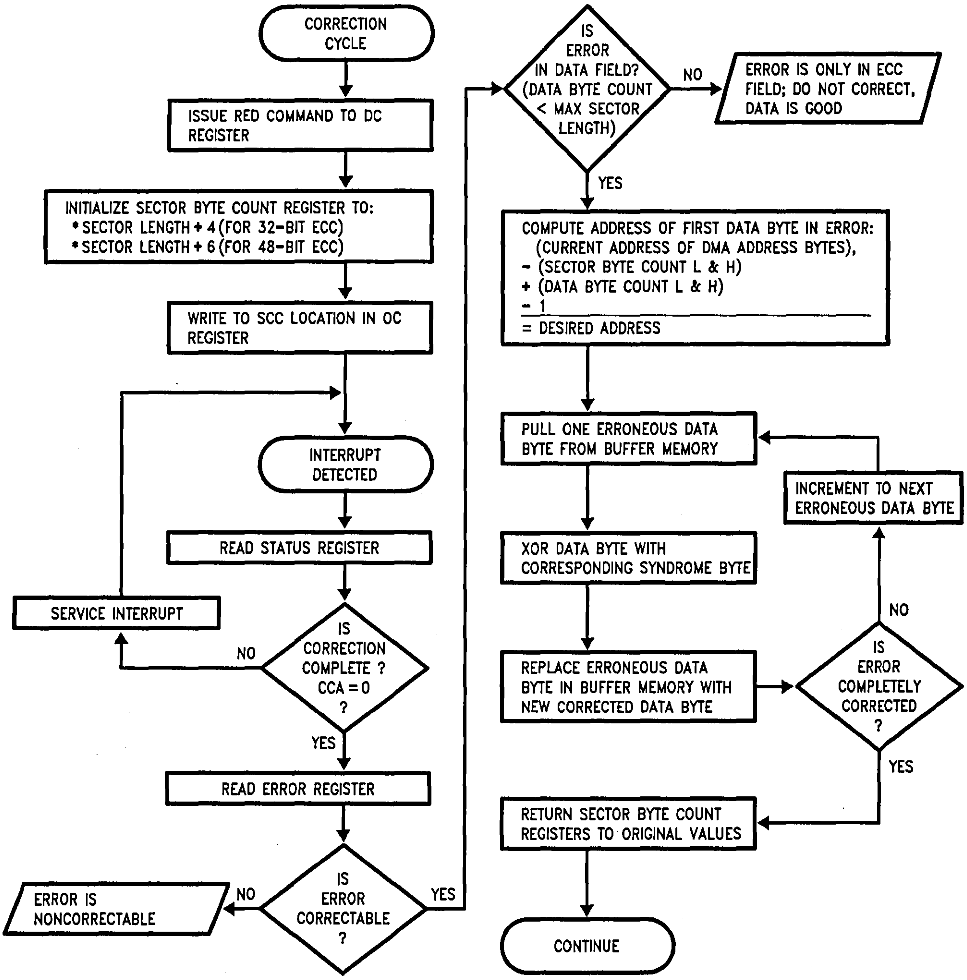
Buffer Memory

CORRESPONDING BUFFER DATA BIT PATTERN							
D7	D6	D5	*	D3	D2	D1	D0
D15	*	*	D12	*	D10	D9	D8
D23	D22	D21	D20	D19	D18	D17	D16

* = location of bits in error

FIGURE 17. Example of a 32-Bit Correction

6.0 CRC/ECC (Continued)


CORRECTION CYCLE
ISSUE RED COMMAND TO DC REGISTER
INITIALIZE SECTOR BYTE COUNT REGISTER TO:
• SECTOR LENGTH + 4 (FOR 32-BIT ECC)
• SECTOR LENGTH + 6 (FOR 48-BIT ECC)
WRITE TO SCC LOCATION IN OC REGISTER
INTERRUPT DETECTED
READ STATUS REGISTER
SERVICE INTERRUPT
NO
IS CORRECTION COMPLETE ? CCA = 0 ?
YES
READ ERROR REGISTER
ERROR IS NONCORRECTABLE
NO
IS ERROR CORRECTABLE ?
YES
IS ERROR IN DATA FIELD? (DATA BYTE COUNT < MAX SECTOR LENGTH)
NO
ERROR IS ONLY IN ECC FIELD; DO NOT CORRECT, DATA IS GOOD
YES
COMPUTE ADDRESS OF FIRST DATA BYTE IN ERROR:
(CURRENT ADDRESS OF DMA ADDRESS BYTES),
- (SECTOR BYTE COUNT L & H)
+ (DATA BYTE COUNT L & H)
- 1
= DESIRED ADDRESS
PULL ONE ERRONEOUS DATA BYTE FROM BUFFER MEMORY
INCREMENT TO NEXT ERRONEOUS DATA BYTE
XOR DATA BYTE WITH CORRESPONDING SYNDROME BYTE
NO
REPLACE ERRONEOUS DATA BYTE IN BUFFER MEMORY WITH NEW CORRECTED DATA BYTE
IS ERROR COMPLETELY CORRECTED ?
YES
RETURN SECTOR BYTE COUNT REGISTERS TO ORIGINAL VALUES
CONTINUE

TL/F/5282-13

FIGURE 18. Correction Cycle Algorithm

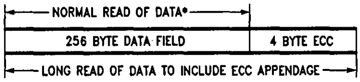
THIS CYCLE CAN ONLY BE INITIATED AFTER A READ DATA OPERATION HAS BEEN COMPLETED

6.0 CRC/ECC (Continued)

A note of caution: If the DDC is in the tracking DMA mode when a data error occurs, the remote DMA channel will transfer the sector in error to its destination in the system. The DDC will still interrupt to indicate that it has detected an error. It is then up to the system to get the DDC to correct the error in buffer memory and retransfer the corrected data to the system.

6.4 ECC CHECK USING LONG READ AND LONG WRITE

During a normal read or write operation, the size of the data field is specified by the Sector Byte Count register pair. If the data field is extended during a readback, the ECC appendage can be read in as data and analyzed outside the DDC. This is what is known as a *long read*.



TL/F/5282-14

*Read length defined by Sector Byte Count register pair.

FIGURE 19. Example of a Long Read

Likewise, an externally generated ECC appendage can be added to the data and written to the disk as data with or without the onboard CRC/ECC generator enabled. This is known as a *long write*.

By using long reads and long writes in conjunction with external software used to produce data fields and external CRC/ECC appendages, various diagnostic programs can be devised to test the DDC's internal correction functions and ECC generation circuitry. These tests could be incorporated in the initialization algorithm to test the chip each time it is powered up.

7.0 Data Transfer

7.1 DIRECT MEMORY ACCESS (DMA)

The DDC is designed to work efficiently in two major system configurations:

(1) A single system bus with shared data buffer/system memory (see *Figure 20*).

(2) A dual bus environment with a local microprocessor, buffer memory and DP8466 on a local bus interfacing the host system bus through an I/O port (see *Figure 21*).

All DMA activity is supported by the following three features:

PROGRAMMABLE BURST LENGTH (THRESHOLD)

Here, the transfer of data between the 32-byte FIFO on the DDC and the external memory (local or main) involves the use of internal or external local DMA channel. While writing to the disk, the DDC will initiate a transfer when the FIFO has been depleted by the burst length. It will also initiate a transfer while reading from the disk when the FIFO fills to the burst length. This length is selectable from 2, 8, 16 or 24 bytes, allowing for the variations in bus latency time encountered in most systems.

At the start of a write operation, the FIFO will be filled up in a series of bursts of the programmed length.

If the exact burst option is not selected, the FIFO will be completely filled (if writing to disk) or emptied (if reading from disk) in one DMA operation. The burst length is always the threshold at which the transfer will be requested and is independent of the DMA mode, including slave.

8-BIT/16-BIT WIDE TRANSFERS

Byte or word wide data transfer can be selected for both local and remote DMA channels. Word wide tranfers with local DMA use the AD0-15 pins, and byte wide use the AD0-7 pins. Both the local and the remote DMA addresses are incremented by 2 for word wide transfers, and 1 for byte wide transfers. Commands and DDC parameter registers are loaded and read only 8-bits at a time, using AD0-7.

REVERSE BYTE ORDER

This option is only valid for 16-bit wide transfers using the local DMA channel. This should not be used for 8 bit wide transfers. It enables the two bytes being transferred to be mapped with the high order byte to AD0-7 and the low order byte to AD8-15, or vice-versa.

The DDC has provisions to accommodate five DMA modes. These are as follows:

EXTERNAL DMA:	1. Slave Mode
INTERNAL DMA, Single Bus:	2. 16-Bit Local Mode
	3. 32-Bit Local Mode
Multiple Bus:	4. Non-Tracking Mode
	5. Tracking Mode

All five modes accommodate the three configurations just described. All DMA modes, except external slave, use an incrementing address. Local channel transfers always have priority over remote channel transfers unless externally reprioritized. If the local channel is used, its transfer length is always automatically loaded from the Sector Byte Count register pair.

7.2 EXTERNAL DMA

SLAVE MODE

In this mode, no on-chip DMA control is used. LRQ and LACK pins are connected to an external DMA controller. After LACK has been granted, I/O RD and I/O WR from the DMA controller are used to strobe data between the internal FIFO and the DDC I/O port. 8-bit and 16-bit wide data transfers are possible. Throughout this data sheet, reference has been made to the use of on-chip DMA for the transfer of data. It is important to note here that external DMA can be used in place of this if so desired.

7.3 INTERNAL DMA

The following four modes all use on-chip DMA control with at least the local channel serving as bus master for data transfers between the internal FIFO and memory.

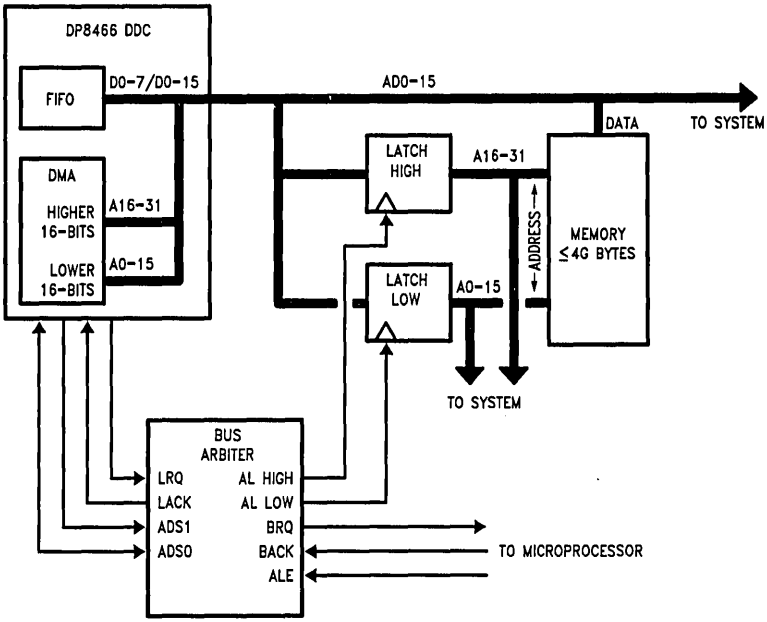
SINGLE BUS SYSTEMS

The following two modes support a single bus and a single shared buffer/system memory. Bus access should be guaranteed before the FIFO overflows or empties during a disk transfer operation. A FIFO Data Lost error (FDL bit in Error register) will be flagged and the operation aborted if this fails to happen. Different system latency times can be accommodated by the selectable burst length.

16-BIT LOCAL MODE

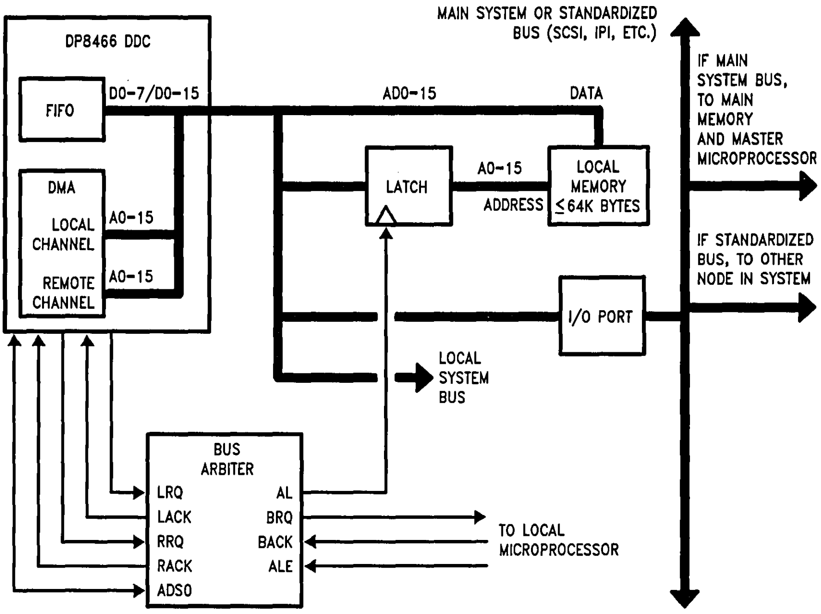
SLD bit is set and LA bit is reset in the LT register. Only the 16-bit local DMA channel is enabled. 64k bytes are directly addressable by the DDC. Address data is presented on AD0-15 and latched with ADS0. Transfers always take 4 BCLK cycles if no wait states are issued.

7.0 Data Transfer (Continued)



TL/F/5282-16

FIGURE 20. Single System Bus, 32-Bit Address DMA



TL/F/5282-15

FIGURE 21. Dual System Bus, 16-Bit Address DMA

7.0 Data Transfer (Continued)

32-BIT LOCAL MODE

SLD bit and LA bit are both set in the LT register. SRD bit in the RT register must be reset. The local DMA channel is now set to issue 32-bit addresses using the remote DMA channel as the upper 16-bit address register. 4 G bytes are addressable by the DDC. During the first DMA cycle of a newly programmed address, or after a roll-over of the lower 16-bit address counter occurs, ADS1 strobes a new high order word (A16-31) into the external address latches. Each time this happens, the DMA cycle is 5 BCLK periods long. When a new high order address is not needed, the DMA cycle is 4 BCLK periods long. ADS0 is used as an output to latch the low order word (A0–15) from the AD0–15 pins into the address latch.

MULTIPLE BUS SYSTEMS

The following two modes support a dual bus environment, where a local microprocessor, buffer memory and the DP8466 interface to the host through an I/O port. The difference between tracking and non-tracking mode is whether the DDC or the controlling microprocessor ensures that an attempt to read data from buffer memory does not occur before data has been written there. Basic algorithms for both are shown in *Figures 22* and *23*.

TRACKING MODE

SLD bit set and LA bit reset in the LT register. SRD bit and TM bit set in the RT register. The DDC ensures that data is not overwritten by data transferred from the FIFO.

This mode effectively turns the buffer memory into a large FIFO. This is accomplished through the use of the DMA Sector Counter (DSC), which keeps track of the difference between sectors read/written to the disk and the sectors transferred to/from the host system. Each time the source transfers a sector of data into buffer memory (length determined by the Sector Byte Count register pair), the DSC register is incremented. It is decremented each time the destination has transferred a sector of data. Whenever the DSC register contents become zero, destination transfers are inhibited. This mode facilitates multi-sector operations.

Example: Tracking Mode, Disk Read

- Source is local DMA
- Destination is remote DMA
- DSC register is reset automatically upon start of operation
- Local and remote start address, SC, NSO, OC and finally DC registers are loaded. Other registers may need to be updated, but this is a minimum set.

A sector is read from the disk and is transferred in bursts from the FIFO to the buffer memory by local DMA. The DSC register then increments and the remote channel can begin transferring the first sector from the buffer memory to the host system. Burst transfers can be interleaved with local DMA, remote DMA and microprocessor all sharing the bus. The local channel bursts have priority over remote bursts. If the remote channel manages to transfer a sector before the local channel has completed the next sector, the DSC register will decrement to zero. Further remote transfers are inhibited until the local channel completes another sector and increments the DSC. In other words, each time a local sector has been transferred, the DSC is incremented and each time a remote sector completes, the DSC is decremented. Therefore, the DDC prevents further buffer memory contents that have not been previously loaded with valid data by the local DMA from being transferred to the host system. The remote channel continues operation until the last byte from the buffer memory has been transferred. An interrupt is issued upon completion of the operation.

NON-TRACKING MODE

SLD bit set and LA bit reset in the LT register. SRD bit set and TM bit reset in the RT register. The remote and local channel addresses are completely independent. The controlling microprocessor must insure that the data to be transferred by the remote channel is not over-written by the local channel and vice-versa. DMA address and count registers are set up independently. Remote start address (DMA Address Bytes 2 and 3) and Remote Data Byte Count registers must be loaded before SRI or SRO bits are set in the OC register. Local or remote transfers may already be in progress when the other channel is started. The local channel has priority over the remote channel. Local bus utilization is then interleaved between the local channel, the remote channel and the controlling microprocessor.

By setting both SRI and SRO simultaneously, any non-tracking remote DMA operation will stop. The present remote address and remote data byte count will be retained and the local DMA will be unaffected. Loading the original OC instruction (input or output) will restart the original instruction from the last remote DMA address.

DMA Mode Select Table

DMA Mode	LT Register		RT Register	
	SLD	LA	SRD	TM
SLAVE	0	0	0	0
16-BIT LOCAL	1	0	0	0
32-BIT LOCAL	1	1	0	0
TRACKING	1	0	1	1
NON-TRACKING	1	0	1	0

NOTE: In either tracking or non-tracking mode, if either channel is loaded with an odd byte transfer count, the DDC will transfer the next higher even number of bytes. For example, if 511 was loaded into the Remote Data Byte Count registers, 512 bytes would be transferred, with valid data only in the first 511 bytes.

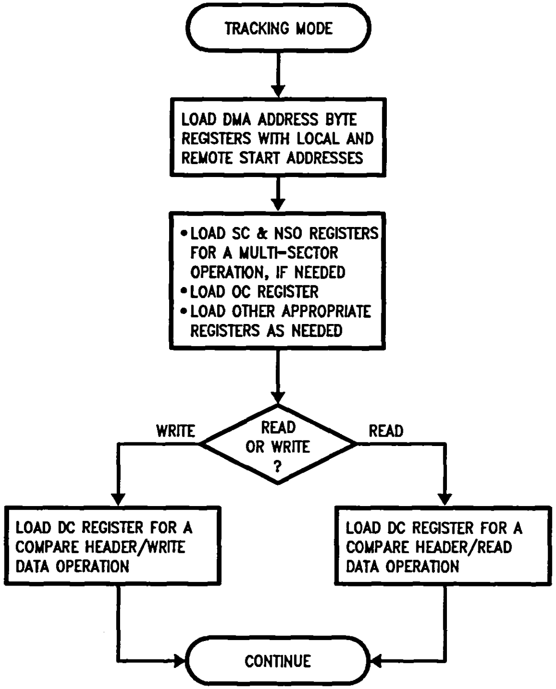
DMA WAIT STATES

INTERNAL

Both DMA channels can independently be set to lengthen the RD and WR strobes by one clock cycle (LSRW bit in the LT register and RSRW bit in the RT register). This lengthens each transfer from 4 cycles to 5 cycles of the BCLK.

EXTERNAL

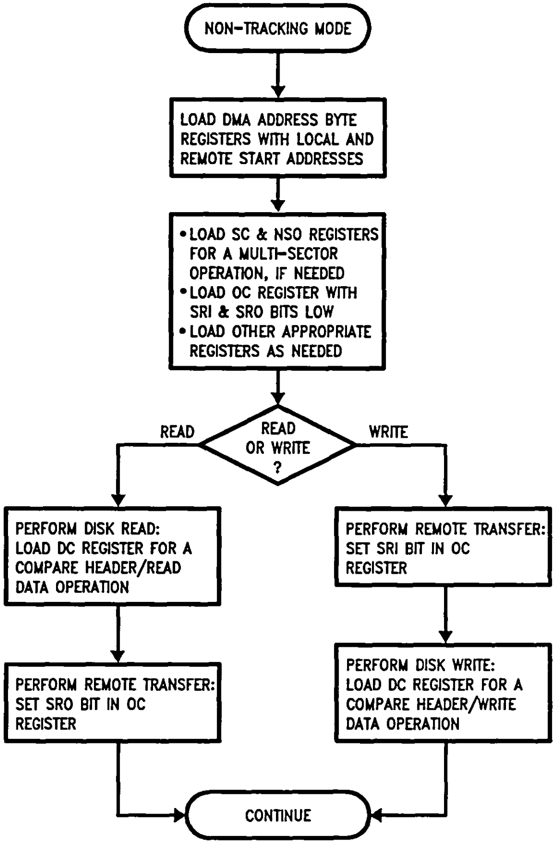
By enabling the external wait states (in the RT register), the EXT STAT pin is configured to insert wait states in each RD and WR pulse as long as this input is high. This is valid for both the local and remote DMA channels.


TRACKING MODE
LOAD DMA ADDRESS BYTE REGISTERS WITH LOCAL AND REMOTE START ADDRESSES
• LOAD SC & NSO REGISTERS FOR A MULTI-SECTOR OPERATION, IF NEEDED
• LOAD OC REGISTER
• LOAD OTHER APPROPRIATE REGISTERS AS NEEDED
READ OR WRITE ?
WRITE
READ
LOAD DC REGISTER FOR A COMPARE HEADER/WRITE DATA OPERATION
LOAD DC REGISTER FOR A COMPARE HEADER/READ DATA OPERATION
CONTINUE

TL/F/5282–17

NOTE: DMA operation is completely automatic for the duration of the command. For example, when reading disk, local DMA empties/fills the FIFO and remote DMA transfers data at least one sector behind the local channel to an I/O port. For disk write, local channel will be at least one sector behind the remote channel.

FIGURE 22. Tracking Mode for Normal Disk Read/Write


NON-TRACKING MODE
LOAD DMA ADDRESS BYTE REGISTERS WITH LOCAL AND REMOTE START ADDRESSES
• LOAD SC & NSO REGISTERS FOR A MULTI-SECTOR OPERATION, IF NEEDED
• LOAD OC REGISTER WITH SRI & SRO BITS LOW
• LOAD OTHER APPROPRIATE REGISTERS AS NEEDED
READ OR WRITE ?
READ
WRITE
PERFORM DISK READ: LOAD DC REGISTER FOR A COMPARE HEADER/READ DATA OPERATION
PERFORM REMOTE TRANSFER: SET SRI BIT IN OC REGISTER
PERFORM REMOTE TRANSFER: SET SRO BIT IN OC REGISTER
PERFORM DISK WRITE: LOAD DC REGISTER FOR A COMPARE HEADER/WRITE DATA OPERATION
CONTINUE

TL/F/5282–18

NOTE: This is the most basic of non-tracking mode operations, and unlimited, more versatile algorithms can be built up from this.

FIGURE 23. Non-Tracking Mode for Normal Disk Read/Write

8.0 Interrupts

Interrupts can only occur if the EI bit in the OC register is set. If it is not set, the INT pin is always de-asserted high. 16 RCLK periods (3.2 μs at 5 Mbit/sec data rate) must pass before servicing an interrupt (i.e. reading Status). Failure to do this will result in servicing the same interrupt twice. There are four general conditions that may cause an interrupt to occur:

Operation Complete
Header Complete
Error
Correction Cycle Complete

OPERATION COMPLETE

This interrupt indicates that the current DDC operation has completed and the DDC is ready to execute a new command. Commands can be loaded sooner by setting EHI bit in the OC register. The Next Disk Command (NDC) bit in the Status register is set coincident with the Header Complete interrupt. New disk commands can be loaded before DMA operation is finished if NDC is set. If the command is a multi-sector operation, the end of operation interrupt will occur only after the operation is completed in the last sector of operation. The INT pin is asserted low when:

- Disk operation is completed for any command that is not a disk read operation.
- A read operation in the tracking DMA mode after the remote transfer is complete.
- A read operation in the non-tracking DMA mode after the local transfer is complete.
- A non-tracking mode remote DMA transfer is completed. This is independent of the disk operation or the local DMA.

HEADER COMPLETE:

If the EHI and EI bits are set in the OC register, an interrupt will occur when any header operation is complete. Multi-sector operations will generate an interrupt after each header in each sector has been operated on. It is asserted two bit times into the ID postamble. This function allows the changing of header bytes (and parameter RAM in general) *on the fly*. The Header Complete interrupt can be used in conjunction with the Interlock Required (IR) bit in the OC register set to insure that changes have been completed before the next sector is encountered (see Interlock Type formatting). Another normal mode of use would be to notify the controlling microprocessor when the next disk command can be loaded. This interrupt is coincident with the Next Disk Command (NDC) bit being set in the Status register.

ERROR

Any bit set in the Error register sets the ED bit in the Status register and causes an interrupt.

CORRECTION CYCLE COMPLETE

An interrupt will occur at the end of an internal correction cycle, regardless of whether the error was corrected or not. If the error was non-correctable, the CF bit will be set in the Error register. This will not generate two interrupts.

CLEARING INTERRUPTS

The INT pin will be forced inactive high any time the Status register is being read. If an interrupt condition arises during a status read, this condition will assert INT as soon as the status read is finished.

Interrupts can also be cleared by setting the internal RES bit, or by asserting the external RESET pin.

9.0 Additional Features

9.1 DATA RECOVERY USING THE INTERLOCK FEATURE

The potential use of the interlock feature is in recovering data from a sector with an unreadable header field. It is assumed that the number of the sector physically preceding the bad sector on the disk is known. A single-sector operation will be performed on these sectors, and the Drive Command register will be changed in between them. The following steps will recover the data:

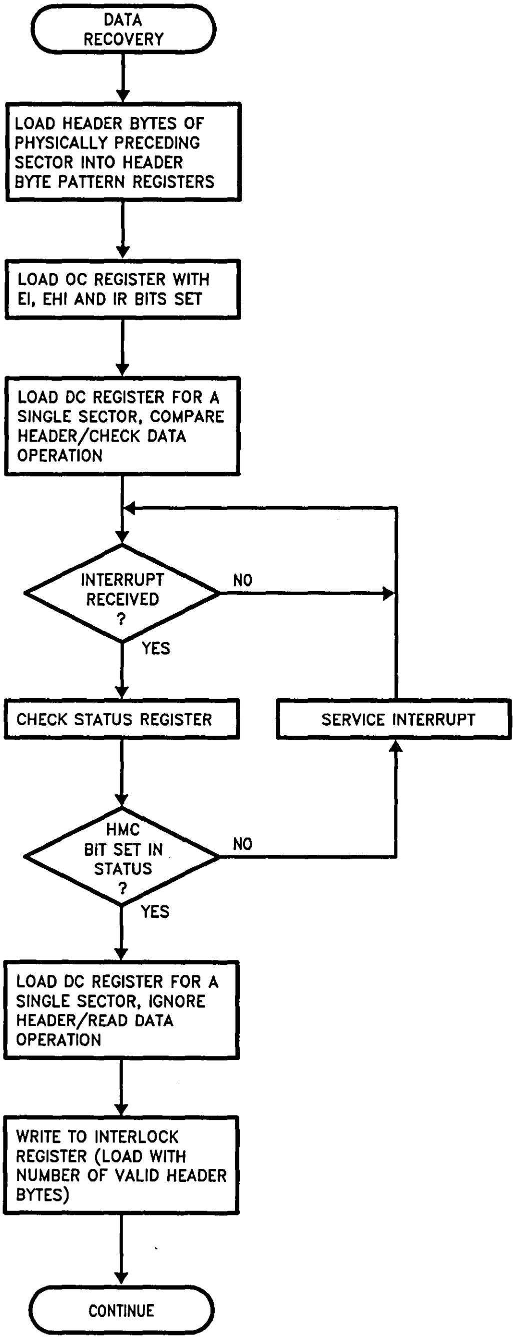
- The header bytes of the physical sector preceding the desired sector are loaded into the relevant byte pattern registers.
- The OC register must be loaded with the EI, EHI and IR bits set. This enables the Header Complete interrupt as well as the interlock feature.
- The DC register is loaded for a single-sector, Compare Header/Check Data operation.
- After the Header Complete interrupt, the DC register must be loaded with an Ignore Header/Read Data operation, and the Interlock (HBC) register written to. If the controlling microprocessor fails to write to the HBC register before the end of the data field of the first sector, a Late Interlock error (LI bit in Error register) will be flagged, and the operation will be terminated with an interrupt.
- When the HMC interrupt occurs on the second sector, the Interlock (HBC) register must be written to again in order to avoid LI error.
- The operation will terminate normally when the data from the badly labeled sector has been read.

9.2 HFASM FUNCTION

The Header Failed Although Sector number Matched (HFASM) function on the DDC can be used to perform maintainance and diagnostic functions, both of which will be briefly outlined here.

The HFASM function is enabled by setting the EHF bit in at least one of the Header Control registers, with a Compare Header command loaded into the DC register. More than one header byte may have its EHF bit set. If any one of the header byte(s) with it's EHF bit set matched, but any other header byte(s) (regardless of the state of their EHF bit) don't match, an HFASM error will occur.

9.0 Additional Features (Continued)


DATA RECOVERY
LOAD HEADER BYTES OF PHYSICALLY PRECEDING SECTOR INTO HEADER BYTE PATTERN REGISTERS
LOAD OC REGISTER WITH EI, EHI AND IR BITS SET
LOAD DC REGISTER FOR A SINGLE SECTOR, COMPARE HEADER/CHECK DATA OPERATION
INTERRUPT RECEIVED ?
NO
YES
CHECK STATUS REGISTER
SERVICE INTERRUPT
HMC BIT SET IN STATUS ?
NO
YES
LOAD DC REGISTER FOR A SINGLE SECTOR, IGNORE HEADER/READ DATA OPERATION
WRITE TO INTERLOCK REGISTER (LOAD WITH NUMBER OF VALID HEADER BYTES)
CONTINUE

TL/F/5282–19

FIGURE 24. Data Recovery Algorithm

9.0 Additional Features (Continued)

In this way, the HFASM function performs a maintenance type function, and can often indicate that the head is positioned over the wrong track. It is independent of whether or not a CRC failure has occurred. An HFASM failure will not stop operation until the header CRC bytes have been compared and the CRC check is completed.

To perform a diagnostic function, the header can be read and analyzed. This can be done only during a Compare Header/Check Data operation with HFASM enabled. This causes the header patterns coming from the disk to be written into the FIFO. We must assume that the FIFO is empty (or has been reset before the operation) in order for this operation not to interfere with data transfers. If an HFASM error occurs during a Header Compare, the FIFO will be left intact and the header with the error can be read out of the FIFO from the Header Diagnostic Readback (HDR) register. (Note: LWDT of the local transfer register must be set to match the bus width of the accessing MP for this function.)

If an HFASM error did not occur, the FIFO will be cleared and the header patterns that were stored there will be lost. This process can only be enabled for one disk command. The Compare Header/Check Data command will enable this function. Any other command will disable it.

10.0 Typical System Configurations

10.1 LOW COST SYSTEM

In a single bus system, the DDC can directly address 4G bytes of main memory. The 16-bit I/O port (AD0–15) is externally demultiplexed and buffered with the octal latches and drivers. The main microprocessor, through a separate disk drive control I/O block, is responsible for commands like Head Select, Seek, TRK 000, Drive select, etc. Bus access must be guaranteed before the FIFO overflows or empties. A short burst length (LT and RT registers) accommodates longer bus latency times and helps to insure this. The burst capability allows for other bus operations to be interleaved while the FIFO is filling (during a read) or emptying (during a write). If long, important CPU operations are required, the next configuration must be used.

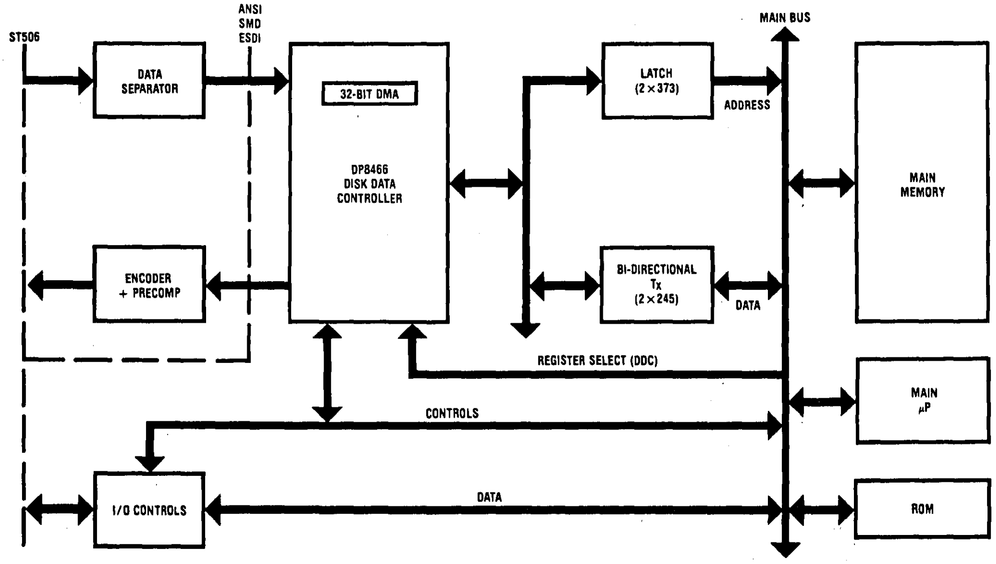


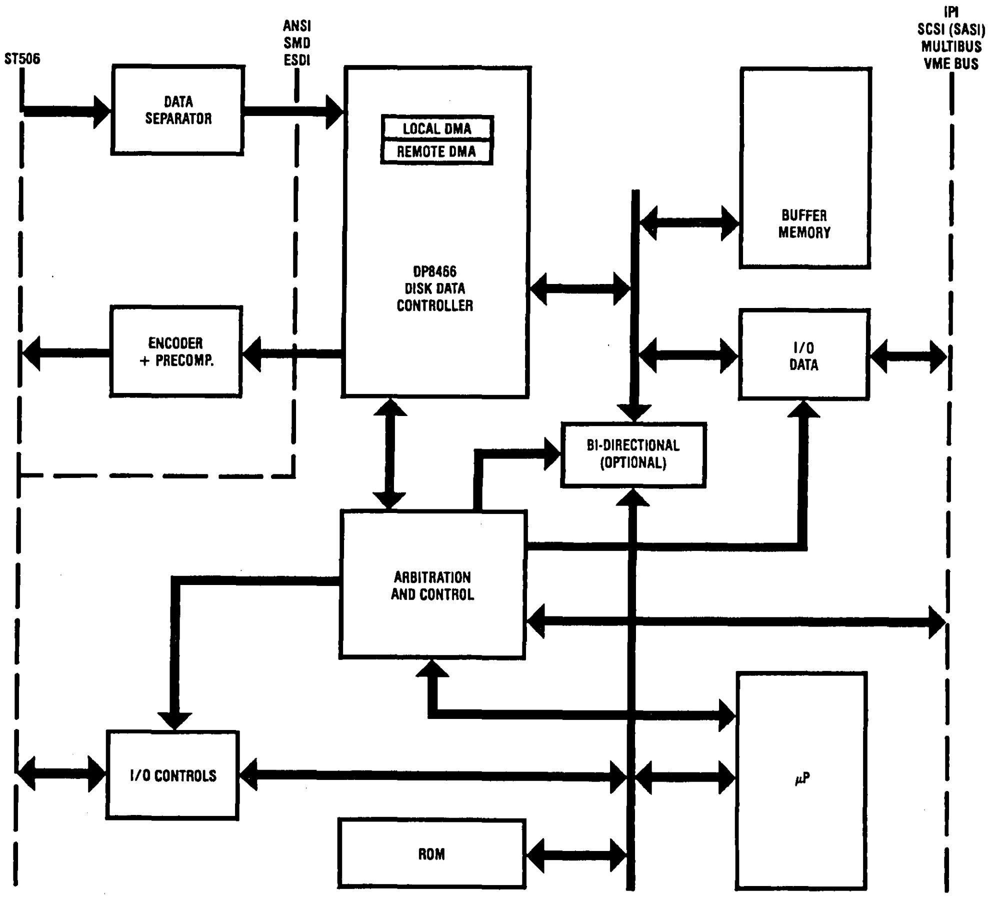
FIGURE 25. Low Cost System Configuration

TL/F/5282-20

10.0 Typical System Configuration (Continued)

10.2 HIGH PERFORMANCE SYSTEM

This configuration provides a local bus for the DDC to share with the local microprocessor and a buffer memory. Here, whole blocks of data can be transferred between the DDC and buffer memory without interfering with the system bus. This leaves the main CPU to perform important operations and to allow data transfers when it is ready. This configuration is also used in intelligent drives or systems that comply to SCSI or IPI specifications. A local bus, dedicated microprocessor and buffer memory are main characteristics of an intelligent disk interface. The buffer memory can be used as a cache for track or file buffering and command lists can be down-loaded for execution by the microprocessor. The two DMA channels can both directly address 64k bytes of buffer memory. The local DMA channel transfers data between the buffer memory and the internal FIFO. The remote DMA channel transfers data between the buffer memory and the host I/O port. With the addition of a bi-directional buffer isolating the DDC from the microprocessor, simultaneous drive operations can be accomplished. While the DDC is transferring data via DMA with the buffer memory, the local microprocessor can issue drive control commands.



TL/F/5282-21

FIGURE 26. High Performance System

INDEX

11.0 Absolute Maximum Ratings*

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage (V_{CC})	−0.5 to +7.0V
DC Input Voltage (V_{IN})	−0.5 to V_{CC}+0.5V
DC Output Voltage (V_{OUT})	−0.5 to V_{CC}+0.5V
Storage Temperature Range (TSTG)	−65°C to +150°C
Power Dissipation (PD)	500 mW
Lead Temperature (TL) (Soldering 10 sec.)	260°C

*Absolute Maximum Ratings are those values beyond which damage to the device may occur.

12.0 DC Electrical Characteristics (V_{CC} = 5V ±10%, unless otherwise specified) T_A = 0°C to +70°C

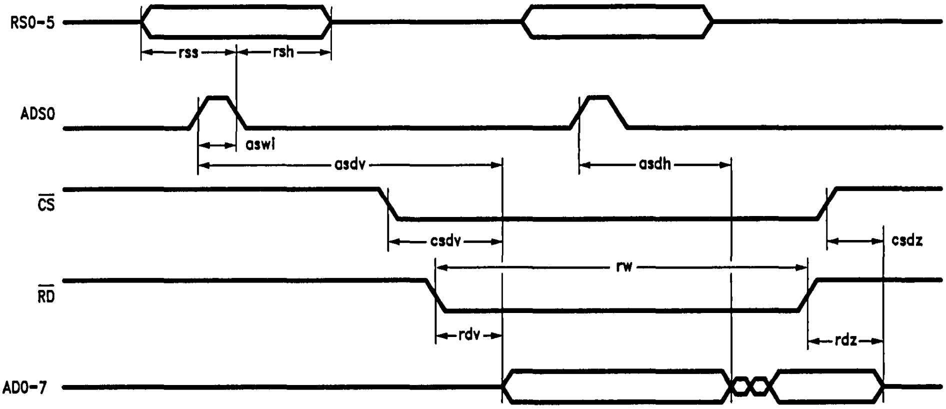
Symbol	Parameter	Conditions	Typ	Limit	Units
V_{IH}	Minimum High Level Input Voltage			2.0	V
V_{IL}	Maximum Low Level Input Voltage			0.8	V
V_{OH1}	Minimum High Level Output Voltage	$\lvert I_{OUT}\rvert$ = 20 μA		V_{CC} − 0.1	V
V_{OH2}		ADS0, ADS1 $\lvert I_{OUT}\rvert$ = 4.0 mA $\lvert I_{OUT}\rvert$ = 2.0 mA		3.5	V
V_{OL1}	Minimum Low Level Output Voltage	$\lvert I_{OUT}\rvert$ = 20 μA		0.1	V
V_{OL2}		ADS0, ADS1 $\lvert I_{OUT}\rvert$ = 4.0 mA $\lvert I_{OUT}\rvert$ = 2.0 mA		0.4	V
I_{IN}	Maximum Input Current	V_{IN} = V_{CC} or GND		±1	μA
I_{OZ}	Maximum TRI-STATE Output Leakage Current	V_{OUT} = V_{CC} or GND		±10	μA
I_{CC}	Supply Current DP8466N-12	V_{IN} = V_{CC} or GND BCLK = RCLK = 12 MHz I_{OUT} = 0 μA	12	25	mA
	Supply Current DP8466N-20	V_{IN} = V_{CC} or GND RCLK = 20 MHz I_{OUT} = 0 μA BCLK = 16 MHz	20	40	mA
	Supply Current DP8466N-25	V_{IN} = V_{CC} or GND BCLK = 20 MHz RCLK = 25 MHz I_{OUT} = 0 μA	25	45	mA

13.0 AC Electrical Characteristics & Timing Diagrams

NATIONAL SEMICONDUCTOR PRELIMINARY TIMING FOR THE DP8466

Note: Refer to 11.4 for AC Timing Test Conditions.
Refer to 11.5.6 for derating factor.

13.1 REGISTER READ (Latched Register Select: ADS0 Active)



TL/F/5282-22

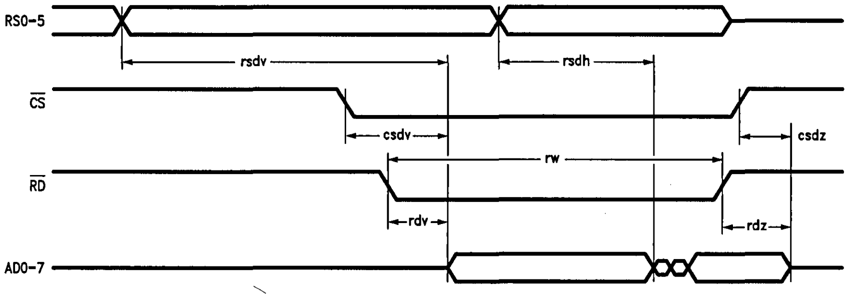
Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
rss	Register Select Setup to ADS0 Low	10		15		ns
rsh	Register Select Hold to ADS0 Low	10		15		ns
aswi	Address Strobe Width In	20		30		ns
asdv	Address Strobe to Data Valid (Note 1)		150		200	ns
csdv	Chip Select to Data Valid		125		150	ns
rdv	Read Strobe to Data Valid		125		150	ns
rw	Read Strobe Width		10		10	µs
csdz	Chip Select to Data TRI-STATE (Note 2)	20	80	20	90	ns
rdz	Read Strobe for Data to TRI-STATE (Note 2)	20	80	20	90	ns
asdh	Data Hold from ADS0 (Note 1)	20		20		ns

Note 1: asdv and asdh timing assumes that RS0-5 is set up before the leading edge of ADS0.

Note 2: TRI-STATE note: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns, enabling other devices to drive this line with no contention.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.2 REGISTER READ (Non-Latched Register Select: ADS0 = 1)



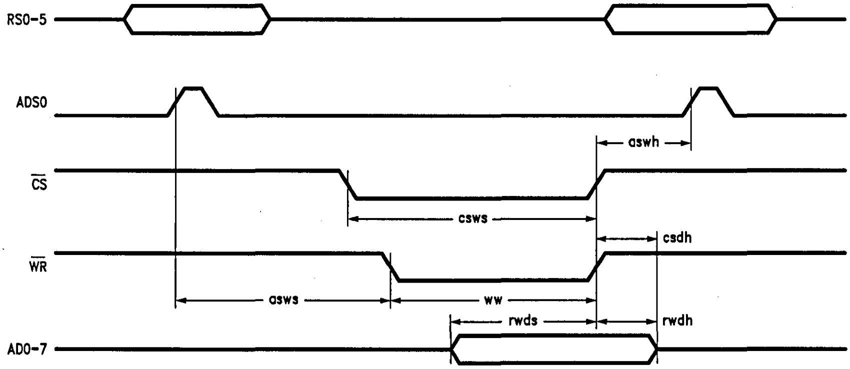
TL/F/5282-23

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
rsdv	Register Select to Data Valid (ADS0 = 1) (Note 1)		150		200	ns
csdv	Chip Select to Data Valid		125		150	ns
rdv	Read Strobe to Data Valid		125		150	ns
rw	Read Strobe Width		10		10	μs
csdz	Chip Select to Data TRI-STATE (Note 2)	20	80	20	90	ns
rdz	Read Strobe for Data to TRI-STATE (Note 2)	20	80	20	90	ns
rsdh	Data Hold from Register Select Change (Note 1)	20		20		ns

Note 1: rsdv and rsdh timing assumes that ADS0 is true when RS0-5 changes.

Note 2: TRI-STATE note: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns, enabling other devices to drive this line with no contention.

13.3 REGISTER WRITE (Latched Register Select: ADS0 Active)



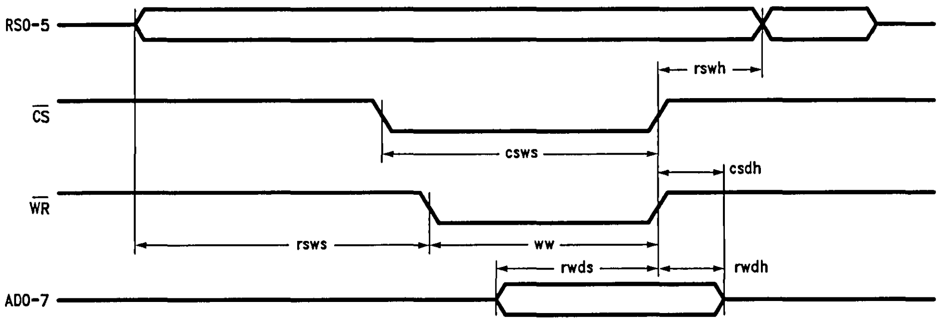
TL/F/5282-24

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
asws	Address Strobe to Write Setup (Note 1)	15		20		ns
csws	Chip Select to Write Setup	50		70		ns
csdh	Chip Select Data Hold	7		10		ns
rwds	Register Write Data Setup	40		50		ns
rwdh	Register Write Data Hold	3		5		ns
ww	Write Strobe Width	50		70		ns
aswh	ADS0 Hold from Write (Note 1)	10		15		ns

Note 1: asws and aswh assume that RS0-5 is set up before the leading edge of ADS0, and held until after the trailing edge of ADS0.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.4 REGISTER WRITE (Non-Latched Register Select: ADS0 = 1)

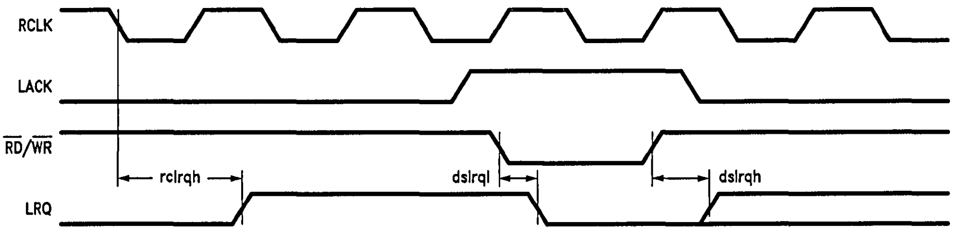


TL/F/5282-25

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
rsws	Register Select to Write Setup (Note 1)	10		15		ns
csws	Chip Select to Write Setup	50		70		ns
csdh	Chip Select to Data Hold	7		10		ns
rwds	Register Write Data Setup	40		50		ns
rwdh	Register Write Data Hold	3		5		ns
ww	Write Strobe Width	50		70		ns
rswh	Register Select Hold from Write (Note 1)	15		20		ns

Note 1: rsws and rswh assume that ADS0 is true when RS0-5 changes.

13.5 LRQ TIMING WITH EXTERNAL DMA



TL/F/5282-26

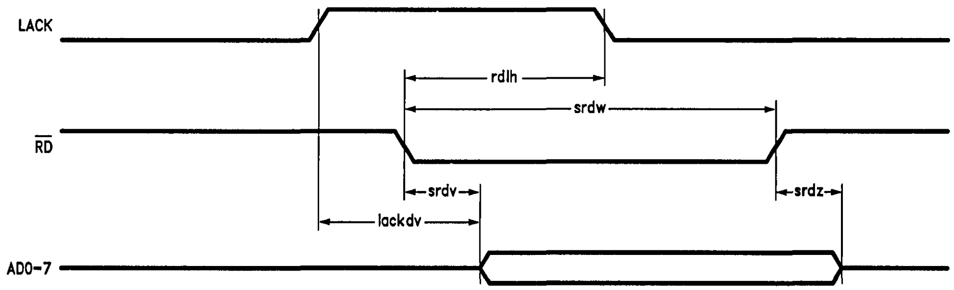
Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
dslrql	Data Strobe to LRQ Low (Note 1)		70		100	ns
dslrqh	Data Strobe to LRQ High (Note 2)		70		100	ns
rclrqh	Read Clock to LRQ High		70		100	ns

Note 1: LRQ is removed at the end of a burst or an operation by the assertion of the RD or WR strobe.

Note 2: LRQ is reissued at the end of a burst by the removal of the RD or WR strobe. This only occurs if sufficient data/space is in the FIFO for another burst and the DDC is operating in the exact burst mode.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.6 READING FIFO DATA IN DMA SLAVE MODE



TL/F/5282-27

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
lackdv	LACK to Data Valid		80		90	ns
rdlh	Read Strobe to LACK Hold (Note 1)	10		15		ns
srdv	Slave Read Strobe to Data Valid		60		70	ns
srdz	Slave Read Strobe to Data TRI-STATE (Note 3)	20	80	20	90	ns
srdw	Slave Read Strobe Width		8rcyc − 100		8rcyc − 100	ns

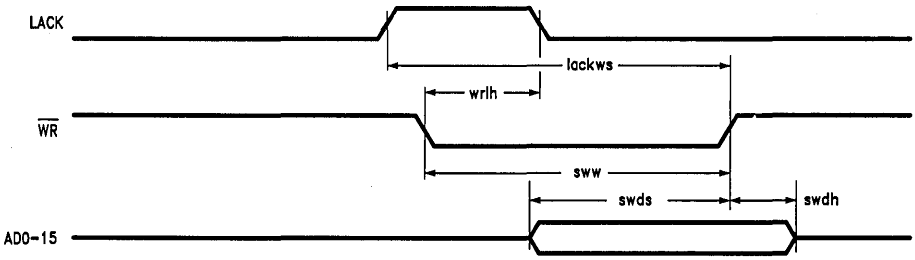
Conditions: Disk read operation, DMA disabled, LRQ output true.

Note 1: The read cycle begins when LACK and $\overline{RD}$ are true. From this point, LACK must be held true for rclh.

Note 2: The maximum rate of FIFO transfers is limited to 1 transfer per 2 rcyc + 75 ns while data is being transferred between the disk and the FIFO.

Note 3: TRI-STATE note: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns, enabling other devices to drive this line with no contention.

13.7 WRITING FIFO DATA IN DMA SLAVE MODE



TL/F/5282-28

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
lackws	Local Acknowledge to Write Setup	50		60		ns
wrlh	Write to LACK Hold (Note 1)	10		15		ns
swds	Slave Write Data Setup	5		10		ns
swdh	Slave Write Data Hold	20		28		ns
sww	Slave Write Strobe Width (Note 2)	30	8rcyc − 100	40	8rcyc − 100	ns

Conditions: Disk write operation, DMA disabled, LRQ output true.

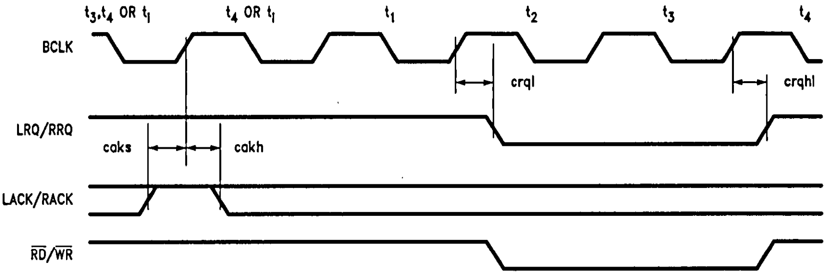
Note 1: The write cycle begins when LACK and $\overline{WR}$ are true. From this point, LACK must be held true for wrlh.

Note 2: The write cycle begins when LACK and $\overline{WR}$ are true. From this point, $\overline{WR}$ must remain true for sww.

Note 3: The maximum rate of FIFO transfers is limited to 1 transfer per 2 rcyc + 75 ns while data is being transferred between the disk and the FIFO.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.8 LOCAL AND REMOTE DMA ACKNOWLEDGE



TL/F/5282-29

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
crql	t2 Clock to Request Low		80		100	ns
crqhl	t4 Clock to Request High (Exact Burst Length Limited)		80		100	ns
caks	Acknowledge Setup to Clock	15		20		ns
cakh	Acknowledge Hold from Clock	10		15		ns

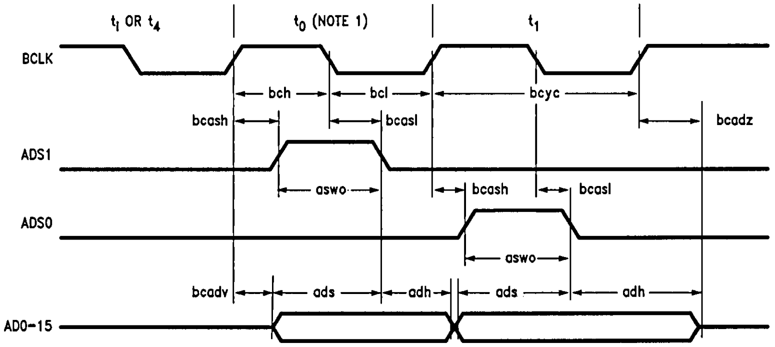
Note 1: The Local and Remote Acknowledges are sampled at the beginning of bus cycles t4 and t1.

Note 2: Local Acknowledge has internal priority over Remote Acknowledge.

Note 3: Local and Remote Acknowledge are ignored if their respective Request output is false.

Note 4: Above timing is for 16 bit address updates. For 32 bit Local address mode, cycle t0 occurs on the first transfer of an operation or when the lower 16 bits of the address roll over.

13.9 DMA ADDRESS GENERATION



TL/F/5282-30

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
bcyc	Bus Clock Cycle Time (Notes 2, 3)	50	10,000	80	10,000	ns
bch	Bus Clock High Time (Note 3)	22.5	10,000	32	10,000	ns
bcl	Bus Clock Low Time (Note 3)	22.5	10,000	32	10,000	ns
bcash	Bus Clock to Address Strobe High		45		55	ns
bcasl	Bus Clock to Address Strobe Low		50		60	ns
aswo	Address Strobe Width In	bch		bch		
bcadv	Bus Clock to Address Valid		60		70	ns
bcadz	Bus Clock to Address TRI-STATE (Note 4)	20		20		ns
ads	Address Setup to ADS0/1 Low	bch − 20		bch − 30		ns
adh	Address Hold from ADS0/1 Low	bcl − 5		bcl − 10		ns

Note 1: Cycle t0 occurs only on the first transfer of an operation or when the lower 16 bits of the address rolls over.

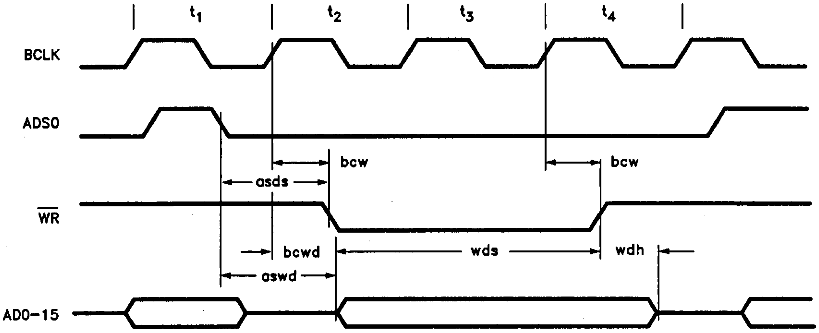
Note 2: The rate of bus clock must be high enough that data will be transferred to and from the FIFO faster than the data being transferred to and from the disk.

Note 3: For DP8466-20, minimum bcyc = 60 ns minimum bch = bcl = 28 ns.

Note 4: **TRI-STATE note:** These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns, enabling other devices to drive this line with no contention.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.10 DMA MEMORY WRITE



TL/F/5282-31

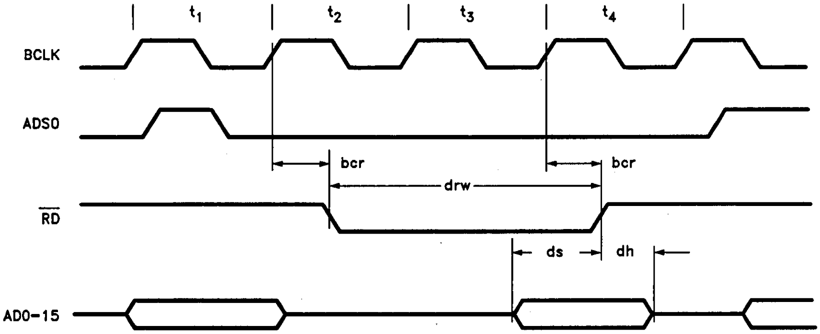
Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
bcw	Bus Clock to Write Strobe		50		60	ns
wds	Data Setup to $\overline{\text{WR}}$ High (Note 1)	2bcyc − 35		2bcyc − 45		ns
wdh	Data Hold from $\overline{\text{WR}}$ high (Note 1)	10	50	10	60	ns
bcwd	Data Valid from t2 Clock (Note 1)		70		90	ns
asds	Address Strobe to Data Strobe (Note 2)	bcl	bcl + 10	bcl	bcl + 20	ns
aswd	Address Strobe to Write Data Valid		bcl + 40		bcl + 60	ns

Conditions: DMA write, Local or Remote transfer, internal DMA.

Note 1: Data is enabled on AD0–15 only in local DMA transfers.

Note 2: Data strobe is either $\overline{\text{RD}}$ or $\overline{\text{WR}}$ out.

13.11 DMA MEMORY READ



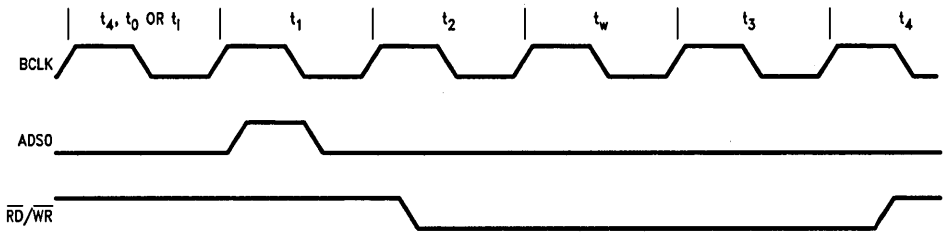
TL/F/5282-32

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
bcr	Bus Clock to Read Strobe		50		60	ns
ds	Data Setup to Read Strobe High	25		30		ns
dh	Data Hold from Read Strobe High	0		0		ns
drw	DMA Read Strobe Width Out	bcyc − 10		bcyc − 15		ns

Note 1: ds and dh timing are for Local transfers only.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.12 DMA WITH INTERNAL WAIT STATES



TL/F/5282-33

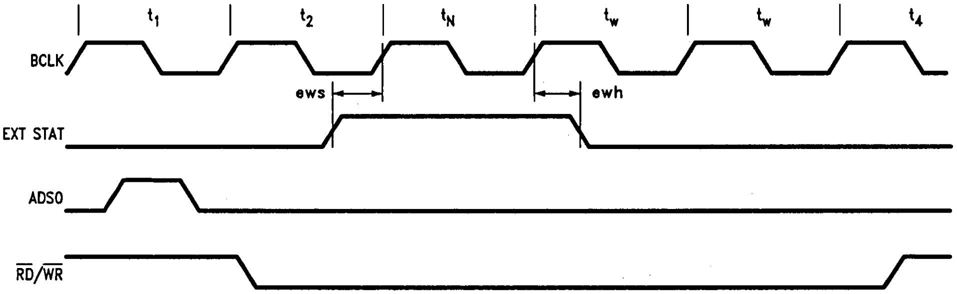
Conditions: Local or Remote DMA transfer, read or write, internal DMA.

Note 1: Addition of an internal wait state will lengthen RD/WR strobes by an additional bus clock cycle.

Note 2: Internal wait states are enabled by setting the Slow Read/Write bits in the Local and Remote Transfer registers.

Note 3: If used, external wait states will be added between cycles t3 and t4.

13.13 DMA WITH EXTERNAL WAIT STATES



TL/F/5282-34

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
ews	External Wait Setup to t3 Clock	15		20		ns
ewh	External Wait Hold after tw Clock	10		15		ns

Conditions: Read or write, internal DMA mode. Local or Remote transfer.

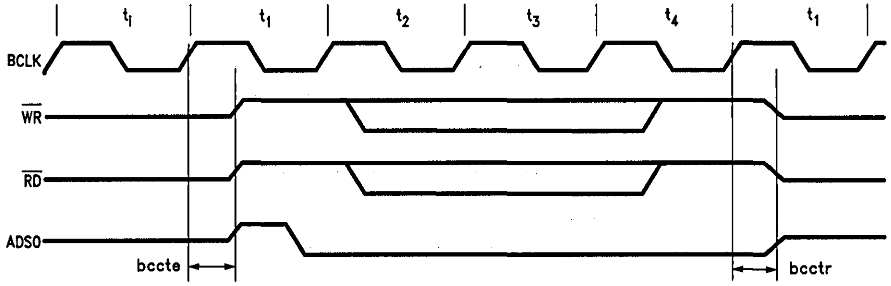
Note 1: Addition of external wait states will extend RD/WR strobes by an integral number of bus clock cycles.

Note 2: If enabled, an internal wait state is added between cycles t2 and t3.

Note 3: EXT STAT is sampled upon entering states t3 and tw, and adds wait states one bus clock cycle later.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.14 DMA CONTROL SIGNALS

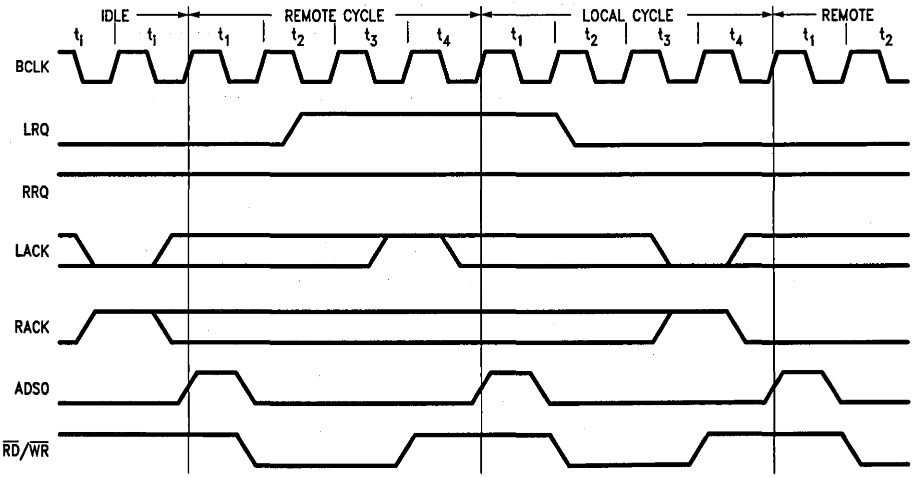


TL/F/5282-35

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
bccte	Bus Clock to Control Enable ($\overline{WR}$, $\overline{RD}$, ADS0)		55		70	ns
bcctr	Bus Clock to Control Release ($\overline{WR}$, $\overline{RD}$, ADS0) (Note 1)		60		70	ns

Note 1: TRI-STATE note: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns, enabling other devices to drive this line with no contention.

13.15 LOCAL AND REMOTE DMA INTERLEAVING



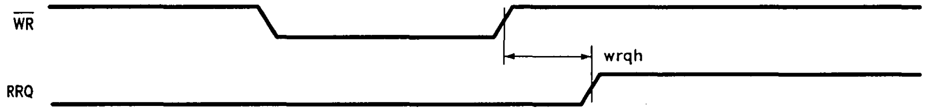
TL/F/5282-36

Note 1: Timing of the acknowledge pulses are used for illustration. Acknowledges need only to be set up with respect to t4 and t1 clock cycle.

Note 2: If both LACK and RACK are asserted with both LRQ and RRQ pending, a local DMA transfer will be performed.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.16 RRQ ASSERTION AFTER WRITING TO OC REGISTER FOR REMOTE TRANSFER

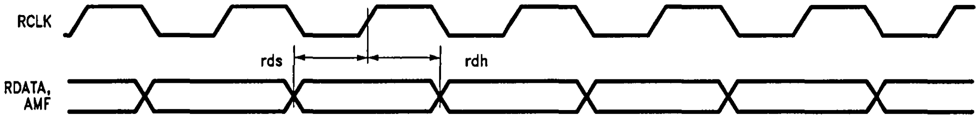


TL/F/5282-37

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
wrqh	Write Strobe to Remote Request High		100		150	ns

Conditions: Non-tracking mode, writing "Start Remote Input/Output" to the Operation Command register.

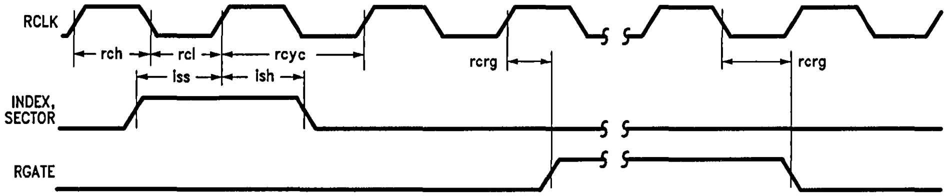
13.17 READ DATA TIMING



TL/F/5282-38

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
rds	Read Data/AMF Setup to Read Clock	10		15		ns
rdh	Read Data/AMF Hold to Read Clock	10		15		ns

13.18 RGATE ASSERTION FROM INDEX OR SECTOR PULSE INPUT



TL/F/5282-39

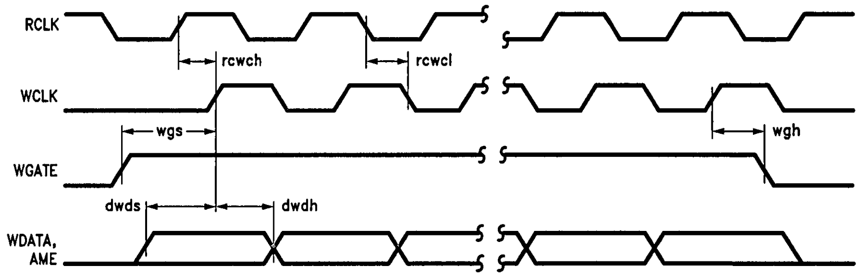
Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
rcyc	Read Clock Cycle Time (Note 2)	40	10,000	80	10,000	ns
rch	Read Clock High Time (Note 2)	16	10,000	32	10,000	ns
rcl	Read Clock Low Time (Note 2)	16	10,000	32	10,000	ns
iss	Index/Sector Setup to Read Clock	10		15		ns
ish	Index/Sector Pulse Hold	10		15		ns
rcrg	Read Clock to Read Gate		45		60	ns

Note 1: INDEX/SECTOR low must meet iss/ish timing for proper INDEX/SECTOR pulse detection.

Note 2: For DP8466-20, minimum rcyc = 50 ns, minimum rch and rcl = 20 ns.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.19 WRITE DATA TIMING FOR NRZ TYPE DATA

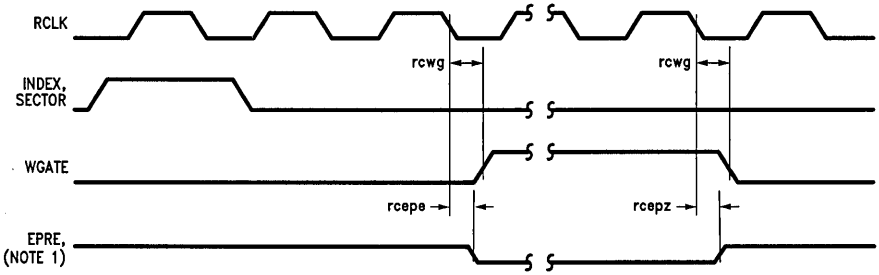


TL/F/5282-40

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
rcwch	Read Clock to Write Clock High Delay		25		40	ns
rcwcl	Read Clock to Write Clock Low Delay		25		40	ns
rcwcs	Absolute Value of (rcwcl — rcwch)		6		7	ns
dwds	Drive Write Data Setup to Write Clock	rcl − 10		rcl − 15		ns
dwdh	Drive Write Data Hold to Write Clock	rch − 5		rch − 8		ns
wgs	Write Gate Setup to Write Clock	rcl − 10		rcl − 15		ns
wgh	Write Gate Hold to Write Clock	rch		rcl		ns

Note 1: rcl and rch are described in Timing Diagram 13.18.

13.20 WGATE ASSERTION FROM INDEX OR SECTOR PULSE INPUT



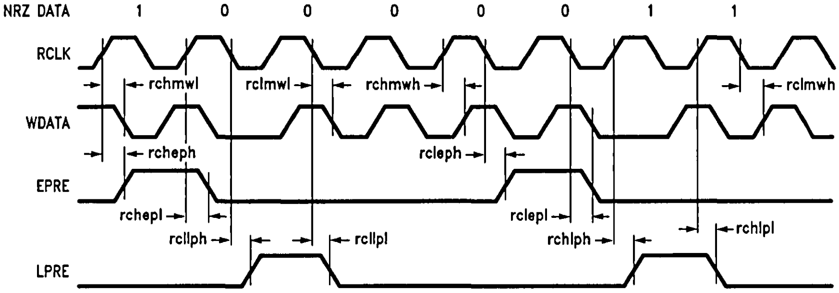
TL/F/5282-41

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
rcwg	Read Clock to Write Gate		40		50	ns
rcepe	Read Clock to Early Precomp Enabled		50		60	ns
rcepz	Read Clock to Early Precomp TRI-STATE		50		60	ns

Note 1: Early Precompensation (EPRE) is used as an output only when writing MFM data.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.21 WRITE DATA TIMING FOR MFM TYPE DATA



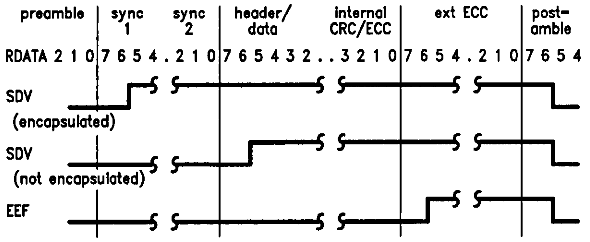
TL/F/5282-42

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
rchmwh	RCLK High to MFM WDATA High		40		50	ns
rchmwl	RCLK High to MFM WDATA Low		40		50	ns
rclmwh	RCLK Low to MFM WDATA High		40		50	ns
rclmwl	RCLK Low to MFM WDATA Low		40		50	ns
rcheph	RCLK High to EPRE High		40		50	ns
rchepl	RCLK High to EPRE Low		40		50	ns
rcleph	RCLK Low to EPRE High		40		50	ns
rclepl	RCLK Low to EPRE Low		40		50	ns
rchlph	RCLK High to LPRE High		40		50	ns
rchlpl	RCLK High to LPRE Low		40		50	ns
rcllph	RCLK Low to LPRE High		40		50	ns
rcllpl	RCLK Low to LPRE Low		40		50	ns

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.22 POSITIONAL TIMING FOR SDV AND EEF

Read operation (Compare Header, Read Header, Compare Data or Read Data)

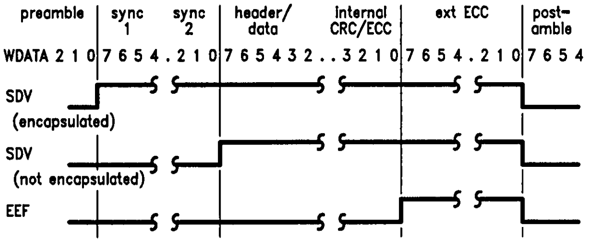


TL/F/5282-43

Note 1: Data should be delayed 2 bit times before entering external ECC circuitry in order for it to properly align correctly with SDC and EEF.

Note 2: Encapsulation is controlled by the HEN and DEN bits in the EC register, and causes the sync patterns to be included in the CRC/ECC calculation.

Write operation (Write Header, Write Data or Format Track)

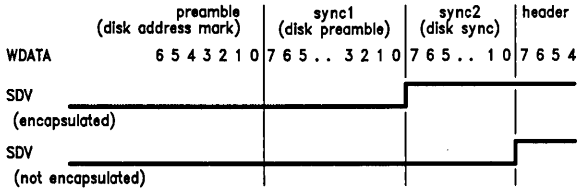


TL/F/5282-44

Note 1: Write operation shown is for NRZ data. For MFM encoding, Write data is delayed two bit times relative to NRZ data.

Note 2: Encapsulation is controlled by the HEN and DEN bits in the EC register, and causes the sync patterns to be included in CRC/ECC calculation.

Write header operation (Start with Address Mark)



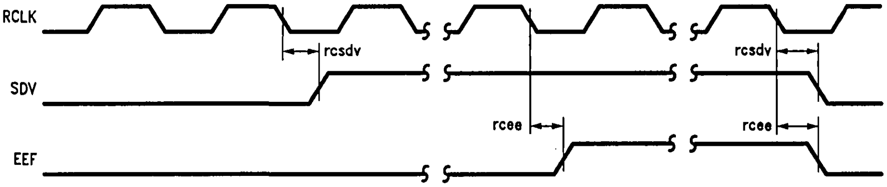
TL/F/5282-45

Note 1: Field names within parenthesis are the names of the fields on disk.

Note 2: Encapsulation is controlled by the HEN bit in the EC register, and causes the sync patterns to be included in CRC/ECC calculation.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.23 FIELD ENVELOPE TIMING



TL/F/5282-46

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
rcsdv	Read Clock to Serial Data Valid		35		50	ns
rcee	Read Clock to External ECC		35		50	ns

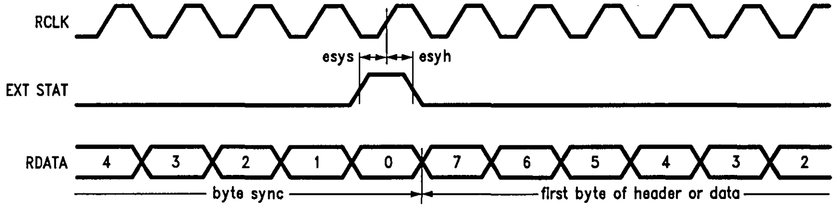
Note 1: SDV is asserted after sync fields, and is deasserted at the start of the postamble field. If sync field encapsulation is enabled, SDV is asserted at the start of the sync fields.

Note 2: EEF is asserted at the start of the external ECC field, and is deasserted at the start of the postamble field.

Note 3: When the DDC is receiving data from the disk, the SDV and EEF are delayed by two bit times from incoming read data due to internal delays.

Note 4: If the external ECC count is set to zero, no EEF output will be generated.

13.24 EXTERNAL STATUS TIMING WHEN USING EXTERNAL BYTE SYNC



TL/F/5282-47

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
esys	External Byte Sync Setup to Rising Edge of Bit Clock 0 of Byte Sync	15		20		ns
esyh	External Byte Sync Hold to Rising Edge of Bit Clock 0 of Byte Sync	10		15		ns

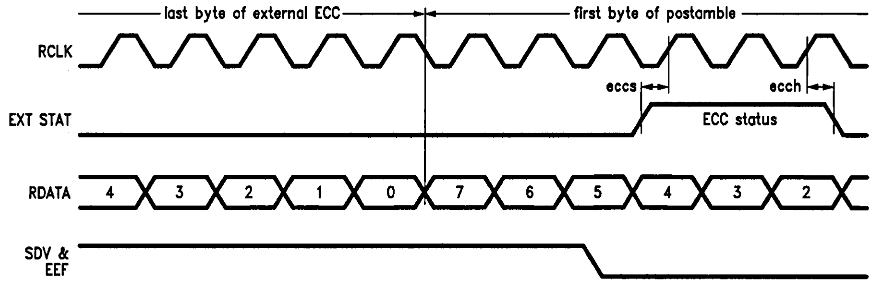
Note 1: The external sync feature can only be used if the Enable External Wait states (EEW) bit of the Remote Transfer (RT) register is not set.

Note 2: External circuitry is needed to feed the DDC with NRZ zeros until the external sync signal has been generated to prevent the DDC from trying to detect sync.

Note 3: If External Sync and External Wait states are not being used, the EXT. STAT. pin must be false during preamble and sync fields.

13.0 AC Electrical Characteristics & Timing Diagrams (Continued)

13.25 EXTERNAL STATUS TIMING WHEN USED FOR EXTERNAL ECC



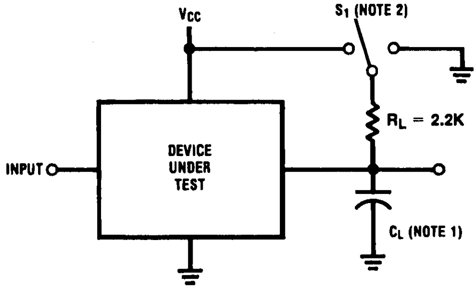
TL/F/5282-48

Symbol	Parameter	DP8466-25/20		DP8466-12		Units
		Min	Max	Min	Max	
eccs	External ECC Status Setup to Rising Edge of Bit Clock 4 of Postamble	15		20		ns
ecch	External ECC Status Hold to Rising Edge of Bit Clock 2 of Postamble	15		15		ns

Note 1: The external ECC error detection feature can only be used if the Enable External Wait states (EEW) bit of the Remote Transfer register (RT) is zero.

14.0 AC Timing Test Conditions

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	6 ns
Input and Output Reference Levels	1.3V
TRI-STATE Reference Levels	Float (ΔV) ±0.5V
Output Load (See Figure 27)	



TL/F/5282-77

FIGURE 27

Note 1: C_L = 50 pF, includes scope and jig capacitance

Note 2: S1 = Open for Push Pull Outputs

S1 = V_{CC} for High Impedance to active low and active low to High Impedance measurements.

S1 = GND for High Impedance to active high and active high to High Impedance measurements.

Capacitance (T_A = 25°C, f = 1MHz)

Parameter	Description	Typ	Max	Unit
C_{IN}	Input Capacitance	7	12	pF
C_{OUT}	Output Capacitance	7	12	pF

Note: This parameter is sampled and not 100% tested.

15.0 Miscellaneous Timing Information

15.1 STATUS REGISTER TIMING

HEADER FAULT: This bit is set at the start of the Header Postamble field of a header with a CRC/ECC error. It is reset at the start of the Header Postamble of the header requested, or upon receipt of a new disk command. No interrupt is generated.

NEXT DISK COMMAND: This bit is set at the start of the Header Postamble of the last sector of an operation, and is reset upon loading the Drive Command register. No interrupt is generated.

HEADER MATCH COMPLETED: This bit is set at the start of the Header Postamble field of the header of interest. This bit is reset when the DDC begins the next header operation. An interrupt is generated if enabled.

LOCAL REQUEST: This bit has the same timing as the Local Request pin. When the FIFO requires servicing, this bit is set. When service is no longer required, this bit is cleared. No interrupt is generated.

REMOTE COMMAND BUSY: In the tracking mode, this bit is set 3–5 RCLK's after receipt of a drive command. In the non-tracking mode, this bit is set when either a Start Remote Input or Start Remote Output command is received in the Operation Command register. This bit is reset and interrupt is generated upon completion of the initiating operation.

LOCAL COMMAND BUSY: This bit is set 3-5 RCLK's after receipt of a drive command which requires the use of the local channel. It is reset after the last transfer of the local channel if in the non-tracking mode or writing the disk, or after the last transfer of the remote channel if in the tracking mode and reading disk. Interrupt is generated upon completion of the initiating operation.

CORRECTION CYCLE ACTIVE: This bit is set upon receipt of the Start Correction Cycle in the Operation Command register, and is reset at the end of the correction operation. An interrupt is generated at the end of the correction cycle.

ERROR DETECTED: This bit is a logical OR function of all the bits in the Error register. An interrupt is generated when an error is detected.

15.2 ERROR REGISTER TIMING

HFASM ERROR: If while in the HFASM mode the sector address matches and another header byte does not, this bit will be set at the start of the Header Postamble field.

DATA FIELD ERROR: If the Data field contains a CRC/ECC error, this bit will be set at the start of the Data Postamble field.

SECTOR NOT FOUND: If the header of the desired sector is not located before two index pulses are received, this bit will be set upon receipt of the second index pulse.

SECTOR OVERRUN: If an index or sector pulse is detected while reading the Header or Data field, or while writing and not in the Gap field, this bit will be set upon receipt of the sector/index pulse.

NO DATA SYNC: If an index or sector pulse is received before data sync is detected, this bit is set upon receipt of the sector/index pulse. If there is a data sync error after the first sync byte has been detected, this bit will be set during the byte following the byte in error.

FIFO DATA LOST: If a transfer between the disk and FIFO causes the FIFO to underrun or overrun, this bit will be set within the next byte time creating a write splice if write gate was on. This is reflected as an ECC error and can be removed if sector is rewritten.

CORRECTION FAILED: This bit is set at the end of the correction cycle if the error is non-correctable.

LATE INTERLOCK: This bit is set at the start of Data Postamble field for Read operations and at the end of the postamble field for non-format Write operations. While formatting, this bit is set at the end of the Gap field.

15.3 GENERAL TIMING FOR READ GATE

Whenever the DDC is reading, comparing, or in some cases, ignoring information, RGATE is asserted. The use of RGATE can be separated into three groups: Header search (soft sectored mode), header examination, and data examination.

SEARCHING FOR HEADERS

When the DDC is searching for a header in the soft-sectored mode, RGATE is asserted in a somewhat random location in the format. After being asserted, if the DDC does not recognize the address mark pattern within eight bit times of detecting a one, RGATE will be de-asserted in 18½ RCLK's. RGATE will then remain low for 17½ RCLK's before another search attempt is made.

In modes where the DDC starts a Read, Compare or Ignore Header operation at an index or sector pulse, RGATE will be asserted 3–4 RCLK cycles from detection of the index or sector pulse.

DATA OPERATIONS

After the header operation has completed, RGATE will be removed two bits after the start of the Header Postamble. If a Read or Check Data operation is to follow, RGATE will be reasserted 11½ bits after the Header Postamble.

At the end of the Data field, RGATE will be removed two bits into the start of the Data Postamble.

15.4 WRITE GATE TIMING

Whenever the DDC is writing information, WGATE is asserted. WGATE can be separated into three uses: Writing header, writing data or track formatting.

WRITING HEADERS

When the DDC writes the header, the write operation does not begin until the receipt of an index or sector pulse. After the pulse is detected, WGATE will be asserted 2½–3½ RCLKs from the detection of the pulse. WGATE will stay true until the end of the Header Postamble, unless the Data field is to be written. If the Data field is to be written, WGATE will not be de-asserted between the Header and Data fields.

WRITING DATA

After a header operation has properly completed, WGATE will be asserted 3 bit times into the Data Preamble. The WGATE will remain active until the end of the Data Postamble. Because of internal delays within the DDC, the Write Data operation is delayed three bit times from the header patterns.

15.0 Miscellaneous Timing Information (Continued)

FORMAT TRACK

In a format track operation, WGATE is asserted 2½–3½ RCLK's from the detection of the index pulse. WGATE will remain active until the next index pulse is detected, and will then be removed.

Note: Detection of an index or sector pulse is defined as the rising edge of the RCLK where index/sector input has met the setup time.

15.5 NORMAL INTERRUPTS

Interrupts are generated by the DDC for a variety of reasons, but they all fall into one of three categories: Either they signal normal completion, a synchronization point, or an error condition. If an interrupt is generated because of an error, the interrupt will have timing as described in the Error register timing section.

The Header Operation Complete interrupt is used for synchronization, and is enabled with the Enable Header Interrupt bit of the Operation Command register. This interrupt will occur when the DDC finishes the header operation, and starts the data operation. For Read, Compare, Write, or Ignore Header operations, the interrupt will be generated at the start of the Header Postamble field.

The normal Operation Complete interrupt is dependent on the operation being performed. If the operation is to Check Data, the interrupt is generated at the start of the Data Postamble field. For Write Data operations, an interrupt will be generated at the end of the Data Postamble. When the DDC is formatting, the interrupt will be delayed by the length of the Header Preamble after the format has finished. The fourth event is further defined by the DMA mode used. For all local channel operations except for tracking mode disk read, the interrupt will be generated during the last transfer of data from the FIFO. In the configuration, tracking mode disk read, the interrupt will be delayed until the last transfer is made by the remote DMA. For all non-tracking remote DMA operations, the interrupt will be generated during the last transfer of the remote DMA.

When a correction operation is being performed, an interrupt is generated at the end of the correction cycle, regardless of the outcome.

15.6 DERATING FACTOR

Output timings are measured with a purely capacitive load for 50 pF. The following correction factor can be used for other loads:

DP8466-25/20	$C_L \geq 50$ pF:	+.13 ns/pF (ADS0, ADS1)
		+.20 ns/pF (all other outputs)
DP846-12	$C_L \geq 50$ pF:	+.18 ns/pF (ADS0, ADS1)
		+.25 ns/pF (all other outputs)

16.0 Appendix

16.1 DDC REGISTERS, INDEX BY HEX ADDRESS

The following is a repeat of what can be found in the DDC INTERNAL REGISTERS Section. This listing is arranged numerically by hex address, and is provided as a quick reference. The section numbers provided indicate where the best description for the particular register can be located. For an explanation of the information contained in the WR and RD columns, refer to the key in the INTERNAL REGISTERS Section.

COLUMN KEY:

HA: *Hex Address* **#B**: *Number of bits* **WR**: *Write* **RD**: *Read* **SC**: *Section*

HA	REGISTER	#B	WR	RD	SC
00	Status Register (S)	8	NO	R	3.1
01	Error Register (E)	8	NO	R	3.1
02	ECC SR Out 0	8	NO	R	3.4
02	Polynomial Preset Byte 0 (PPB0)	8	D	NO	3.4
03	ECC SR Out 1	8	NO	R	3.4
03	Polynomial Preset Byte 1 (PPB1)	8	D	NO	3.4
04	ECC SR Out 2	8	NO	R	3.4
04	Polynomial Preset Byte 2 (PPB2)	8	D	NO	3.4
05	ECC SR Out 3	8	NO	R	3.4
05	Polynomial Preset Byte 3 (PPB3)	8	D	NO	3.4
06	ECC SR Out 4	8	NO	R	3.4
06	Polynomial Preset Byte 4 (PPB4)	8	D	NO	3.4
07	ECC SR Out 5	8	NO	R	3.4
07	Polynomial Preset Byte 5 (PPB5)	8	D	NO	3.4
08	Data Byte Count (0)	8	NO	R	3.4
08	Polynomial Tap Byte 0 (PTB0)	8	D	NO	3.4
09	Data Byte Count (1)	8	NO	R	3.4
09	Polynomial Tap Byte 1 (PTB1)	8	D	NO	3.4
0A	Polynomial Tap Byte 2 (PTB2)	8	D	NO	3.4
0B	Polynomial Tap Byte 3 (PTB3)	8	D	NO	3.4
0C	Polynomial Tap Byte 4 (PTB4)	8	D	NO	3.4
0D	Polynomial Tap Byte 5 (PTB5)	8	D	NO	3.4
0E	ECC CONTROL (EC)	8	D	NO	3.4
0F	Header Byte Count (HBC)/Interlock	3	F	R	3.1
10	Drive Command Register (DC)	8	C	NO	3.1
11	Operation Command Register (OC)	8	C	NO	3.1
12	Sector Counter (SC)	8	C	R	3.1
13	Number of Sector Operations Counter (NSO)	8	C	R	3.1
14	Header Byte 0 Pattern	8	C	R	3.3
15	Header Byte 1 Pattern	8	C	R	3.3
16	Header Byte 2 Pattern	8	C	R	3.3
17	Header Byte 3 Pattern	8	C	R	3.3
18	Header Byte 4 Pattern	8	C	R	3.3
19	Header Byte 5 Pattern	8	C	R	3.3
1A	Remote Data Byte Byte Count (L)	8	C	R	3.2
1B	Remote Data Byte Byte Count (H)	8	C	R	3.2
1C	DMA Address Byte 0	8	C	R	3.2
1D	DMA Address Byte 1	8	C	R	3.2
1E	DMA Address Byte 2	8	C	R	3.2
1F	DMA Address Byte 3	8	C	R	3.2
20	Data Postamble Byte Count	5	D	R	3.3
21	ID Preamble Byte Count	5	C	R	3.3
22	ID Sync #1 (AM) Byte Count	5	D	R	3.3
23	ID Sync #Byte 2 Count	5	D	R	3.3
24	Header Byte 0 Control	5	D	R	3.3
25	Header Byte 1 Control	5	D	R	3.3
26	Header Byte 2 Control	5	D	R	3.3
27	Header Byte 3 Control	5	D	R	3.3
28	Header Byte 4 Control	5	D	R	3.3
29	Header Byte 5 Control	5	D	R	3.3
2A	Data External ECC Byte Count	5	D	R	3.3
2B	ID External ECC Byte Count	5	D	R	3.3
2C	ID Postamble Byte Count	5	D	R	3.3
2D	Data Preamble Byte Count	5	D	R	3.3
2E	Data Sync #1 (AM) Byte Count	5	D	R	3.3
2F	Data Sync #2 Byte Count	5	D	R	3.3
30	Data Postamble Pattern	8	D	R	3.3
31	ID Preamble Pattern	8	D	R	3.3
32	ID Sync #1 (AM) Pattern	8	D	R	3.3
33	ID Sync #2 Pattern	8	D	R	3.3
34	Gap Byte Count	8	F	R	3.3
35	Disk Format Register (DF)	8	D	NO	3.1
36	Header Diagnostic Readback (HDR)	8	NO	R	3.1
36	Local Transfer Register	8	I	NO	3.2
37	DMA Sector Counter (DSC)	8	NO	R	3.2
37	Remote Transfer Register	8	I	NO	3.2
38	Sector Byte Count 0	8	D	R	3.2
39	Sector Byte Count 1	8	D	R	3.2
3A	Gap Pattern	8	F	R	3.3
3B	Data Format Pattern	8	F	R	3.3
3C	ID Postamble Pattern	8	D	R	3.3
3D	Data Preamble Pattern	8	D	R	3.3
3E	Data Sync #2 (AM) Pattern	8	D	R	3.3
3F	Data Sync #2 Pattern	8	D	R	3.3

16.0 Appendix (Continued)

16.2 ALPHABETICAL MNEMONIC GLOSSARY AND INDEX

Listed on the following pages are the majority of the abbreviations used within this data sheet as mnemonics to describe portions or functions of the DDC. The section numbers referenced indicate where the terms are first defined. Mnemonics from the specifications section are not included here.

MNEMONIC	DESCRIPTION	SECTION
AD0-7	Address/Data 0-7 (pins 41-48)	2.0
AD8-15	Address/Data 8-15 (pins 1-8)	2.0
ADS0	Address Strobe 0 (pin 9)	2.0
ADS1	Address Strobe 1 (attached to RRQ, pin 37)	2.0
AME	Address Mark Enable (attached to LPRE, pin 13)	2.0
AMF	Address Mark Found (attached to EPRE, pin 16)	2.0
BCLK	Bus Clock (pin 40)	2.0
CCA	Correction Cycle Active (bit in Status register)	3.1
CF	Correction Failed (bit in Error register)	3.1
$\overline{CS}$	Chip Select (pin 28)	2.0
CS0-3	Correction Span Selection (bits in EC register)	3.4
DC	Drive Command register	3.1
DNE	Data Non-Encapsulation (bit in EC register)	3.4
DF	Disk Format register	3.2
DFE	Data Field Error (bit in Error register)	3.1
D01, 2	Data Operation bits (command in DC register)	3.1
DSC	DMA Sector Counter	3.2
E	Error register	3.1
EC	ECC Control register	3.4
ED	Error Detected (bit in Status register)	3.1
EEF	External ECC Field (pin 26)	2.0
EEW	Enable External Wait (bit in RT register)	3.2
EHF	Enable HFASM Function (bit in HC0-5 registers)	3.3
EHI	Enable Header Interrupts (command in OC register)	3.1
FTF	FIFO Table Format (bit in DF register)	3.1
HBA	Header Byte Active (bit in HC0-5 registers)	3.3
HBC	Header Byte Count register	3.1
HC0-5	Header Byte 0-5 Control registers	3.3
HDR	Header Diagnostic Readback register	3.1
HNE	Header Non-Encapsulation (bit in EC register)	3.4
HF	Header Fault (bit in Status register)	3.1
HFASM	Header Failed Although Sector number Matched (bit in Error register)	2.0
HMC	Header Match Completed (bit in Status register)	3.1
H01, 2	Header Operation bits (command in DC register)	3.1
HSS	Hard or Soft Sectored (bit in DF register)	3.1
ID1, 2	Internal Data Appendage (bits in DF register)	3.1
IDI	Invert Data In (bit in EC register)	3.4
IH1, 2	Internal Header Appendage (bits in DF register)	3.1
$\overline{INT}$	Interrupt (pin 29)	2.0
LA	Long Address (bit in LT register)	3.2
LACK	Local DMA Acknowledge (pin 39)	2.0
LBL1, 2	Local Burst Length (bits in LT register)	3.2
LCB	Local Command Busy (bit in Status register)	3.1
LI	Late Interlock (bit in Error register)	3.1
LPRE	Late Precompensation (attached to AME, pin 13)	2.0
LRQ	Local DMA Request (pin 36)	2.0
LRQ	Local Request (bit in Status register)	3.1
LSRW	Local Slow Read/Write (bit in LT register)	3.2
LT	Local Transfer register	3.2
LTEB	Local Transfer Exact Burst (bit in LT register)	3.2
LWDT	Local Word Data Transfer (bit in LT register)	3.2
MFM	MFM Encode (bit in DF register)	3.1
MSO	Multi-Sector Operation (command in DC register)	3.1
NCP	Not Compare (bit in HC0-5 registers)	3.3
NDC	Next Disk Command (bit in Status register)	3.1
NDS	No Data Synch (bit in Error register)	3.1
NSO	Number of Sector Operations counter	3.1
OC	Operation Command register	3.1
PPB0-5	Polynomial Preset Byte 0-5	3.4
PTB0-5	Polynomial Tap Byte 0-5	3.4
RACK	Remote DMA Acknowledge (pin 38)	2.0
RBL1, 2	Remote Burst Length (bits in RT register)	3.2
RBO	Reverse Byte Order (bit in LT register)	3.2
RCB	Remote Command Busy (bit in Status register)	3.1
RCLK	Read Clock (pin 25)	2.0
$\overline{RD}$	Read (pin 11)	2.0
RDATA	Read Data (pin 15)	2.0
RED	Re-Enable DDC (command in DC register)	3.1
RES	Reset DDC (bit OC register)	3.2
RGATE	Read Gate (pin 19)	2.0
RRQ	Remote Request (attached to ADS1, pin 37)	2.0
RS0-5	Register Select 0-5 (pins 30-35)	2.0
RSRW	Remote Slow Read/Write (bit in RT register)	3.2
RT	Remote Transfer register	3.2
RTEB	Remote Transfer Exact Burst (bit in RT register)	3.2
RWDT	Remote Word Data Transfer (bit in RT register)	3.2
S	Status register	3.1
SAIS	Start At Index or Sector (command in DC register)	3.1
SAM	Start at Address Mark (bit in DF register)	3.1
SC	Sector Counter	3.1
SCC	Start Correction Cycle (command in OC register)	3.1
SDV	Serial Data Valid (pin 27)	2.0
SLD	Select Local DMA (bit in LT register)	3.2
SNF	Sector Not Found (bit in Error register)	3.1
SO	Sector Overrun (bit in Error register)	3.1
SRD	Select Remote DMA (bit in RT register)	3.2
SRI	Start Remote Input (command in OC register)	3.1
SRO	Start Remote Output (command in OC register)	3.1
SSC	Substitute Sector Counter (bit in HC0-5 registers)	3.3
TM	Tracking Mode (bit in RT register)	3.2
WCLK	Write Clock (pin 21)	2.0
WDATA	Write Data (pin 18)	2.0
WGATE	Write Gate (pin 20)	2.0
$\overline{WR}$	Write (pin 10)	2.0

DISK SYSTEM ICs COMBAT DATA ERRORS

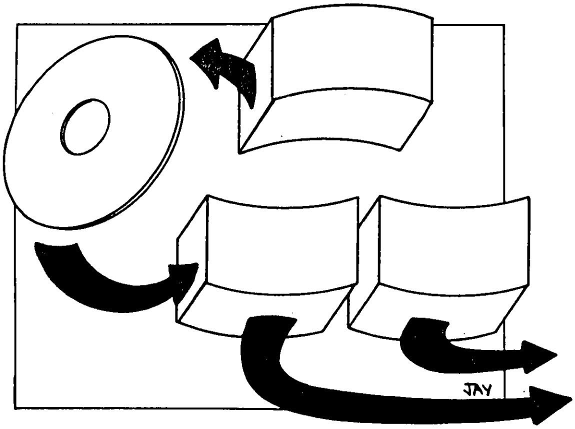
Disk drive controller ICs that implement advanced and flexible error correction techniques require little development work by the designer, yet provide powerful methods of coping with hard errors in disk drives.

by Herb Schneider

While storage capacity is first on the list of factors determining the cost/performance figure for a disk drive, the overall error rate of the drive is a close second. These two factors are closely related, for along with an increase in data density comes an increased risk in data transfer errors. To reduce error rates, manufacturers of systems and disk drive controllers have turned to error checking and correcting codes.

A disk drive controller that normally handles such tasks as formatting and serializing/deserializing data can also spot errors coming off the disk subsystem (the mechanical drive and drive electronics). When an error is detected, the controller can try to recover the data using simple error recovery techniques (rereading) or attempt to correct the data using error checking and correcting circuitry or software.

The controller and its error correction capability determine how much the host is involved when the controller finds an error. Some systems will crash as a result of an error; others will correct the error automatically or with manual aid. If backup data were on tape or floppy, for example, the information could be retrieved. While this might be acceptable in some cases, error correction codes (ECCs) allow the controller to recover from most errors without host intervention. Until recently, system designers concerned with data integrity had to develop the hardware needed to handle errors. Now, more sophisticated controller ICs that can implement ECCs, such as the National Semiconductor DP8466 family, have eased this burden.



Herb Schneider is an applications engineer at National Semiconductor (Santa Clara, Calif). He holds a BS in electrical engineering from the University of California at Davis.

Reprinted with permission from *Computer Design*, March 1985.

Failures in ECC systems

Error detection and correction codes (EDCs) are not perfect, however. The EDC may fail to detect an error or may miscorrect an error. In the first instance, the probability that an error will reach the host depends on two factors: the error rate from the disk subsystem (or P_e, the probability of an error) and the number of errors not detected by the controller (or P_{udt}, the probability of an undetected error). The total probability that an error reaches the host, P_{fail}, is simply P_e multiplied by P_{udt}. In the second instance, the miscorrection probability (P_{mc}) for the controller subsystem is P_e multiplied by the probability of miscorrection (P_{mcc}). Both of these cases are dangerous since the host system is being informed that it has received good data.

Today's drives exhibit error rates on the order of 10^{-10}. ECCs (32 bits) have nondetection probabilities on the order of 10^{-10}. This makes the probability of not detecting an error extremely low—about 10^{-20}. In correction mode, the miscorrection probability is on the order of 10^{-5}, making the probability of an error reaching the host about 10^{-15}. By including error recovery techniques, most systems can recover data without attempting correction.

CRC versus ECC

Floppy disk drive systems use error checking codes to detect errors. Among these, cyclic redundancy codes (CRCs) are typical. Since correction is not possible with this approach, the drive rereads the sector to recover from the error. Assuming that the error is a soft error, a subsequent reread should be successful in recovering the data. However, if the error is a hard error, subsequent rereads will fail to correct the error and the data may be lost. In the case of floppy disks, the faulty disk can be discarded after being replaced by a backup diskette.

Because Winchester disks are not removable, their situation is more serious. To combat such failures, hard disk controllers go a step farther and use ECCs. These codes not only detect an error, but determine the pattern and location of the error. In some applications, ECCs can improve disk performance by avoiding rereads.

Many factors determine the type of error correction required for a drive. Codes, density, phase locked loop design, and precompensation all affect the choice of an ECC.

The purpose of coding in disk storage systems is to combine clock and data information efficiently. The clock information must be encoded on the disk to synchronize the clock with the flux transitions when reading the disk. This requires a data separator circuit (such as the DP8460) and a pulse detector (such as the DP8464). The data separator uses a PLL to synchronize flux transitions with the clock information and allow decoding of the information.

Disk manufacturers use various coding techniques to minimize the number of flux transitions required to represent data on a disk. This is accomplished by deleting unnecessary clock information. By minimizing the number of flux transitions required, the disk data density can be increased without increasing the actual recording density on the medium itself. These codes are collectively known as run-length limited codes. Widely used RLL codes include modified frequency modulated (MFM), and the 2,7 and 1,7 RLL codes.

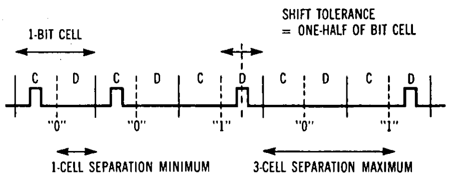
An MFM bit cell consists of two windows per data bit—a clock cell and a data cell. If a 1 is recorded, the data cell contains a pulse. If a 0 follows a 1 in the clock bit, the clock bit in the next bit cell will be deleted. MFM is known as a 1,3 RLL code because it guarantees the minimum distance between pulses to be one cell and the maximum distance between pulses to be three cells. An error will occur if a data pulse enters a clock window or vice versa. This shifting may occur as a result of noise or intersymbol interference (the tendency of crowded bits to interact). If there is a single-bit error, the decoder recovers after 1 bit. This means one physical error results in a single logical error in the decoder (ie, the coding does not propagate the error).

Newer codes such as 2,7 (minimum distance between pulses is 2, maximum distance is 7) are pushing densities even higher. This is accomplished by converting the data sequence into a code sequence that statistically reduces the number of flux changes required to record a data sequence on the media. Thus, a particular data sequence is represented by a code sequence on the drive. If a single-bit error occurs on the drive, the decoder will interpret it as a code sequence that is different from the one originally recorded. And, it will be unable to convert the code sequence back to the correct data sequence. As a result, the length of the physical error will be propagated by a logical error in the decoder. The length of this error is dependent on the number of bits required for the decoder to recover.

With typical 2,7 codes, the time required to recover from a single-bit error can be as high as 4 bits, (5 bits if the original bit in error is included). To determine the correction capabilities of the ECC, system designers must add this length to the maximum burst length that is to be corrected. A typical 32-bit ECC will not suffice for 2,7. By increasing the ECC to 48 bits, however, larger burst errors—typical of those found in 2,7—can be handled.

Density and ECCs

Bit density affects the number of bits involved in a read error that is caused by noise events. It also tends to degrade the signal-to-noise ratio of the



Modified frequency modulation is a 1,3 run-length limited code. All bits must remain in their cells or an error will be generated.

channel. A noise event of the same duration for a 10-Mbit/s drive will involve twice the number of bits as a 5-Mbit/s drive. Doubling density thus doubles the burst length for errors generated by noise effects. As drive manufacturers push bit rates to 24 Mbits/s, burst error lengths will increase correspondingly.

Many drive errors are attributed to such noise in the read channel as pulse detector offsets, peak shifts caused by bit crowding, or channel asymmetry. The effect of all these is to make the pulse appear displaced from the position in which it was originally recorded.

A PLL's ability to tolerate bit shifts is a factor in overall error rates on drives. Bit shift tolerance in a PLL depends on two parameters: bandwidth and window error. Bandwidth affects the time it takes the loop to lock (acquisition time); a higher bandwidth yields a faster acquisition time, but also increases the PLL's tendency to track individual bit shifts (bit shift following). As a result, a trade-off must be made between acquisition time and bit shift following. As a general rule, the lower the bandwidth, the better the immunity to shifted bits.

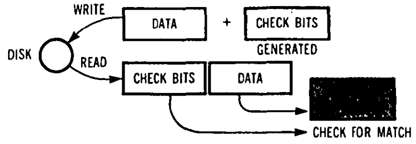
Using delay lines

A common design technique used in PLLs is to delay the data, allowing alignment of the voltage control oscillator edge in the center of a bit cell. The accuracy of this delay determines how far a bit may be shifted before a decoding error occurs. Typical designs employ one shots, which have temperature and voltage variations that limit their performance. Other designs use delay lines that tend to be quite expensive. Integrated PLLs, such as those used in the DP8460, are more tolerant of bit shifts. The DP8460 features an accurate silicon delay line that improves the bit shift tolerance of disk systems. Decoding badly shifted bits greatly reduces the overall error rate of the disk drive, possibly eliminating many errors that would otherwise occur.

A more serious problem with PLLs occurs when they lose sync with the data, and lock on harmonics. As a result, the remaining data in the sector will be read with errors. It is important to avoid these types of read errors because most codes are unable to detect them.

As the head steps toward the inner tracks, bit density increases and bits begin interacting. To compensate for this, bits can be shifted to counteract their interaction on the media. These techniques, known as precompensation, improve the error margin on the PLL. Precompensation, however, must be chosen carefully because a poorly designed scheme can degrade rather than improve performance.

Error detection on serial bit streams has been available for some time. Two commonly used codes for communications are the CRC-CCITT (used in synchronous data link control and high level data link control) and the Autodin II polynomial (used in Ethernet). In communication applications, the basic idea behind error detection codes is to allow a transmitting station to generate a check field that provides information for a receiving station. This information enables the receiver to detect errors in the transmission.



During a write cycle, the disk drive generates and adds a check-bit field to the data field. During a read, the computing check bit is compared to the previously recorded bit.

In a disk drive, the transmitter is the write electronics and the receiver is the read electronics. Writing to the disk involves generating and then appending a check-bit field to the data or ID field. When the sector is subsequently read, the read channel reads the data and computes a check-bit field. This check-bit field is compared to the check field read from disk. If the two fields match, the sector has been read with no errors. If the two fields do not match, the sector has been read with one or more errors.

When writing to disk, check bits are generated by dividing the data stream by a polynomial. The check bit field is the remainder of this division, and is complemented and shifted out following the data—most significant bit first. When reading from disk, the remainder term will again be generated and will cancel the received remainder term if they match. Mathematically this can be expressed as:

$$\frac{X^m \cdot D(x)}{G(x)} = Q(x) + R(x)/G(x)$$

where D(x) is data, P(x) is the polynomial, Q(x) is the quotient, R(x) is the remainder (one's complement final state of shift register), and X^m is the premultiplication factor. When reading a sector, both D(x) and R(x) are processed. If the received R(x) matches the current state of the shift register, the polynomial will divide evenly into D(x) + R(x).

Polynomial division is implemented with a linear feedback shift register (LFSR). As an example, the polynomial $X^4 + X^1 + X^0$ can be implemented with an XOR tap placed at the input to each term of the polynomial. If the LFSR is initialized to all 0s, and encodes a data sequence of 001, the sequence of shifts would be:

Shift No.	Data in	X^0	X^1	X^2	X^3
1	0	0	0	0	0
2	0	0	0	0	0
3	1	1	1	0	0

The check bit sequence appended to this data stream would be 0011 and the sector would be written as 0010011. When reading this sector, the shift register would first be initialized to all 0s, then shifted with the data stream and check bits, resulting in the following shift sequence:

Shift No.	Data in	X^0	X^1	X^2	X^3
1	0	0	0	0	0
2	0	0	0	0	0
3	1	1	1	0	0
4	0	0	1	1	0
5	0	0	0	1	1
6	1	0	0	0	1
7	1	0	0	0	0

Since there has been no error in reading the data, the check-bit field will return the shift register back to its initialized state of all 0s. After the third shift, bits X^0 to X^4 contain the same sequence as the next 4 bits to be read from the disk. Thus, the shift register acts as a serial comparator and cancellation of the two patterns results in all 0s (ie, D(x) + R(x) divided evenly by polynomial).

If a read error occurs, the final state of the shift register before the check-bit field arrives will not match the check-bit field sequence and cancellation will not occur. For example, if the 001 data stream is corrupted and 000 is read, the following sequence will occur:

Shift No.	Data in	X^0	X^1	X^2	X^3	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	<Beginning
5	0	0	0	0	0	of check-
6	1	1	1	0	0	bit field
7	1	1	0	1	0	<Last state

The last state of the shift register after reading the check-bit field is not the same as the initialized value. The last state of the shift register, called the syndrome, can be used to find the location of the error by using some unique properties of ECC polynomials. ECC polynomials have a linear sequence of shifts that can be counted and decoded to determine the location and pattern of a read error. The polynomial $P(x) = X^4 + X^1 + X^0$ has two possible sequences. If the LFSR is preset to all 0s, the LFSR will not leave the all 0s state. If the LFSR is preset to 0001, a repeating 16-symbol sequence will be generated.

The number of symbols a code can generate is referred to as the natural period of the polynomial. The natural period of the code determines how many data symbols may be encoded by the polynomial. Error correction involves determining the distance in the sequence between the syndrome and some reference state. In the polynomial above, a read error was simulated with a resultant syndrome of 1010. When the shift register decodes state S1 (0001), the number of shifts required to reach S1 from the syndrome indicates where the error is located. (In this polynomial, the error location is calculated as 16 − NSHIFTS − 3.) In the case of the 1010 syndrome, 10 shifts are required to reach state S1, indicating that the error is in the third bit of the data stream.

Forward shifting and alternatives

This method of error correction is called forward shifting. Its disadvantage is that it requires many shifts to locate the error. This is related to the natural period of the code. As a result, a 32-bit code could require 43,000 shifts to locate the error at the beginning of a 512-byte sector. This is intolerable in many disk drive systems. A second disadvantage of forward shifting for correction occurs if an error is noncorrectable. This is because the correction circuitry or software must step through the entire period of the polynomial to determine noncorrectability.

A more efficient way to find the error is to step through the polynomial sequence in reverse, using a reciprocal polynomial that generates a sequence in reverse order to the forward polynomial. In the example, the reciprocal polynomial would be $P_r(x) = X^4 + X^3 + X^0$. An LFSR shifted in the opposite direction to the forward polynomial will generate the reverse sequence of the forward polynomial. Shifting the syndrome using $P_r(x)$ will reduce then number of shifts required to locate the error. It will also guarantee finding noncorrectable errors in no more than S + A shifts where S is the number of data bits in the sector and A is the number of check bits.

The time required to correct an error can be a critical ECC performance parameter. In the example, only five shifts would be required to locate the error using the reciprocal polynomial and reverse shifting. These techniques can be extended to include multiple bit correction. While the generator in the example was preset to all 0s, it is better to preset to all 1s to avoid sync slippage errors.

Specifications for measuring the performance of ECCs include detection spans, correction spans, misdetection probability, and miscorrection probability.

The detection span of any polynomial is the length of the largest error burst guaranteed to be detected. The detection span is fixed by the length of the LFSR. The correction span refers to the number of contiguous bits that are guaranteed to be corrected. (A typical 32-bit ECC has a correction span of approximately 6 bits.)

At first glance, it would seem that selecting a high correction span for a particular polynomial would result in higher system integrity. Unfortunately, the opposite is true. All ECCs exhibit some miscorrection probability for the set of all possible errors. Miscorrection results when the span of the error exceeds the correction span and the ECC interprets the error as correctable. The ECC will then add to the problem by miscorrecting another region in the sector and indicating that the error is corrected—a dangerous situation in most systems.

The relationship between miscorrection probability, the number of check bits, and the number of data bits can be expressed as:

$$\text{Miscorrection} = \frac{S \cdot 2^{(C-1)}}{2^{A}}$$

where S is the number of data bits, C is the correction span, and A is the number of check bits. Increasing the correction span by a single bit doubles the miscorrection probability. The only way to compensate miscorrection is to reduce the number of data bits encoded by the polynomial, decrease the correcting span, or increase the number of check bits.

Certain ECC polynomials, known as Fire codes, contain pattern sensitivities that radically increase miscorrection probabilities when encountering certain types of errors. Fire codes that were used in

Error correction in the disk drive controller

Early system implementation of error checking and correcting codes required the actual correction cycle to be performed in software. This approach had the disadvantage of requiring the system designer to develop and test software for correction. Software implementations also tended to be much slower than hardware implementations. National Semiconductor's DP8466 disk data controller includes circuitry that has been designed to locate and correct errors at maximum speed. The logic generates the reciprocal polynomial and reverse shifts to reduce the number of shifts for correction. Generation of the reciprocal polynomial also removes the programming tasks that the CPU must perform to begin a correction cycle.

The DP8466 does not require that the data be passed through the ECC during the correction cycle as did some early ECC designs. As a result, the DP8466 can locate and generate a correction pattern in less than the time to read a sector. If an interleave of one sector is performed, the error can be corrected before the next sector is read. At the end of a correction cycle, the disk data controller provides a byte-aligned correction pattern and a displacement pointer to the first byte in error in the sector. By performing a simple XOR, the CPU can correct the data pattern. If the error is not correctable, ECC circuitry indicates correction has failed.

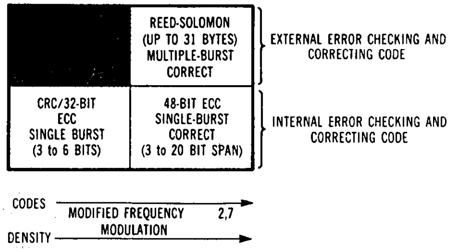
Of prime concern to a system designer is the miscorrection probability. The miscorrection probability is directly proportional to the number of bits to be corrected and is inversely proportional to the number of check bits. A programmable correction span allows the programmer to select a span no larger than the guaranteed correction span for that code. A 5-bit register on the disk data controller is provided to select the correction span.

The DP8466 also features flexible assignment of either cyclic redundancy check, internal ECC, or external ECC to the ID field or the data field. In a typical application, a 16-bit CRC may be used for error detection on the ID field and a 32-bit ECC might be assigned to the data field. It is also possible to assign the 32-bit ECC to both the ID and data field.

There is no single polynomial that manufacturers have agreed upon. But, to ensure flexibility and compatibility with existing designs (especially crucial with removable media), the DP8466 disk data controller has fully programmable polynomial taps. A series of six registers selects the desired polynomial for either 32- or 48-bit operation. The taps reconfigure the linear feedback shift register (LFSR) for any 32- or 48-bit code. A single preprogrammed CRC is implemented—this CRC is the CRC-CCITT most commonly used in disk drives.

There are two types of ECC circuits implemented in existing drives: LFSRs implemented with internal XORs and LFSRs implemented with external XORs. External XOR implementations use parity generators and produce a check-bit sequence different from an equivalently programmed internal XOR ECC. By programming the preset pattern of an internal XOR implementation, a check-bit sequence that is the same as an internal XOR can be generated. The six preset registers on the DP8466 allow full hardware compatibility with existing ECC designs.

Encapsulation provides an interesting application of both internally generated ECC and externally generated ECC. A typical use of two ECC fields is to encapsulate the data field and an ECC field with a second ECC field. When the first ECC detects a read error and correction is performed, the second ECC can determine whether the corrected data is correct. This type of scheme provides an additional level of protection when using error correction. Any miscorrection of the data field by the second ECC will be detected by the first ECC before erroneous data is passed to the host.



Selection of an error checking and correcting code is dependent on media encoding and correcting requirements. A selected polynomial can be 32 bits or 48 bits or longer with a correction span that minimizes the miscorrection probability.

earlier controllers and many VLSI controllers exhibit pattern sensitivities to multiple burst errors. This results in miscorrection probabilities as high as 10 percent. To overcome this system defect, a second generation of codes, developed by a computer search and known as computer-generated codes, have been chosen to provide more reliable correction.

Computer-generated codes meet certain detection and correction spans. They are then tested against a set of requirements to avoid pattern sensitivities. The Fire- and computer-generated codes are known as single-burst ECCs. (These codes detect multiple-burst errors but cannot correct them.) A burst or multiple burst of error with extremities exceeding the correction cannot be corrected—even though the system is informed of the error.

Error recovery techniques

A third class of codes can correct multiple-burst errors or a single long-burst error. These codes are block interleaved Reed-Solomon codes.

When an error is encountered, it will be detected at the end of the sector. The controller must determine whether to attempt to reread the sector or to attempt to correct it. It is quite common to try to reread the sector in order to avoid correction of soft errors. Rereads reduce the number of errors that the ECC must handle, and decreases the overall miscorrection probability since an attempt at correction may result in a miscorrection.

Several recovery methods are used when rereading a sector. If the head is not properly centered over the track, the controller will make small adjustments to step the head over the track. The ST412HP interface standard includes a new step line to allow adjustment of the head correctly over the track. Each incremental step is followed by a reread to determine if the error disappears. Another technique commonly used in storage module devices is the strobe early/strobe late used in system measuring devices to shift the PLL's decode window to read in the presence of phasing errors. After these reread techniques have been exhausted and the syndrome after each reread appears to be consistent, the error is considered a hard error and correction is initiated by the controller.

When a hard error is encountered repeatedly in the same sector, the sector containing the error will be mapped out of the available storage space by the controller. This inhibits access to the sector in future write operations. To avoid writing faulty sectors, one of the manufacturing steps that takes place before shipping disk drives is to identify and map those sectors containing defects that would result in hard errors. The controller will avoid writing these sectors. Sectors containing defects are often identified in a defect map on sector 0. As a result of these mapping strategies, the controller is usually exposed only to soft errors that are recoverable with a reread.

During rereads, a performance penalty occurs since a latency of one revolution is imposed. Although this may be acceptable in single-user applications, the latency may not be acceptable for a multi-user access disk drive or a disk drive that has many read errors. Many systems attempt correction on a sector immediately after an error is detected. Invoking correction on any error is possible, but many techniques require correction in software. This means a reread may be a faster method of recovery than actual correction.

To select an ECC, the designer must examine the types of expected errors (single-burst, double-burst), type of media encoding, and data rate (large-bursts, signal-to-noise degraded). In addition, media defects, available error recovery techniques, PLL margins, and write channel and read channel signal-to-noise margins must be considered. Based on these issues, the user can select either a 32-bit, 48-bit, or longer polynomial and a correction span to minimize the miscorrection probability.

Selection of an ECC is also dependent on media encoding and correction requirements. Current MFM encoded drives at 5 Mbits/s require about 32 bits of ECC with a 5- to 6-bit correction span. The 2,7 RLL drives require a greater correction span and typically will use a 48-bit ECC with an 11-bit correction span.

Section 4
Floppy Disk Support

Section Contents

National Semiconductor Corporation

PRELIMINARY

DP8470 Floppy Disk Support Chip Data Separator & Write Precompensation

General Description

This part is a general purpose data separator which can be used to generate a read clock for FM or MFM encoded data. This read clock can be used with many existing floppy disk controllers including the μPD765A, 8272A, and WD179x. It can also be used with National Semiconductor's Hard Disk Controller, DP8466, for a combination hard disk/floppy disk system. The data separator can be used for data rates ranging from 125 kbits/sec up to 1.25 Mbits/sec.

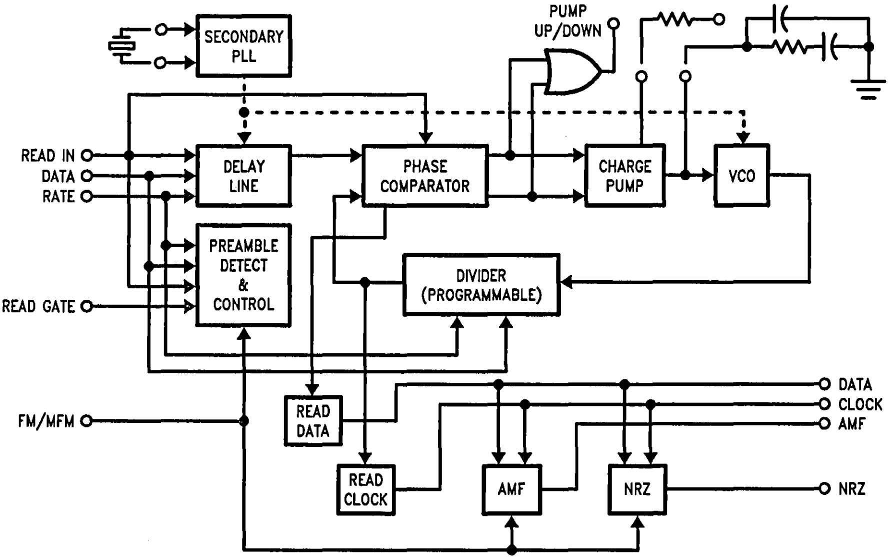
This part also contains a write precompensation circuit. Normally a disk controller will determine whether a bit of data needs to be shifted early, late, or not at all. The controller does not do the actual shifting however. This disk support chip will do the actual shifting that is requested by the controller.

A few other miscellaneous circuits have been included to enable this part to be interfaced to a hard disk controller such as the DP8466. The hard disk controller requires that the data read off from the disk be converted to an NRZ format rather than MFM encoded. Also, the controller needs to know when a valid address mark has been read from the data stream. This disk support chip does both of these functions.

Features

- Analog dual-gain PLL data separator
- Write precompensation (0–393 ns)
- Requires no external trimmable components
- Supports FM/MFM 125 kbits–1.25 Mbits/sec
- Interface to all popular floppy disk controllers.
- Interface to DP8466 hard disk controller
  - — Address Mark Found output
  - — NRZ output
- Pump up/down output for testing
- Low power CMOS
- 24-pin narrow package or 28-pin PCC

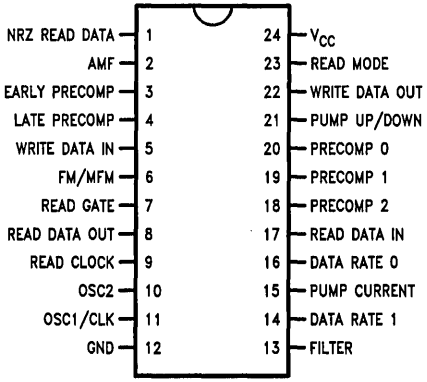
Block Diagram



TL/F/8593-1

Connection Diagram

Dual-In-Line Package

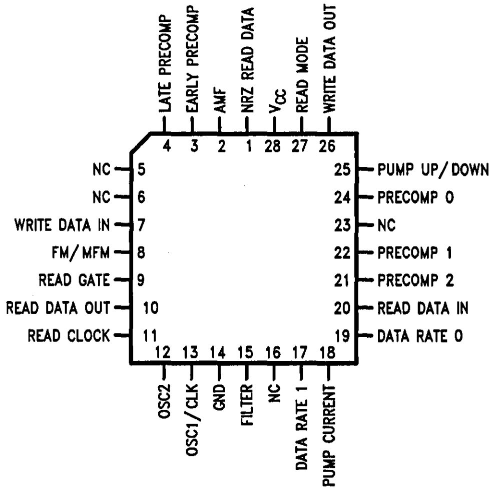


TL/F/8593-2

Top View

Order Number DP8470N or DP8470J
See NS Package Number N24C or J24F

Plastic Chip Carrier



TL/F/8593-9

Top View

Order Number DP8470V
See NS Package Number V28A

Note: Make no corrections to NC pins (No connection).

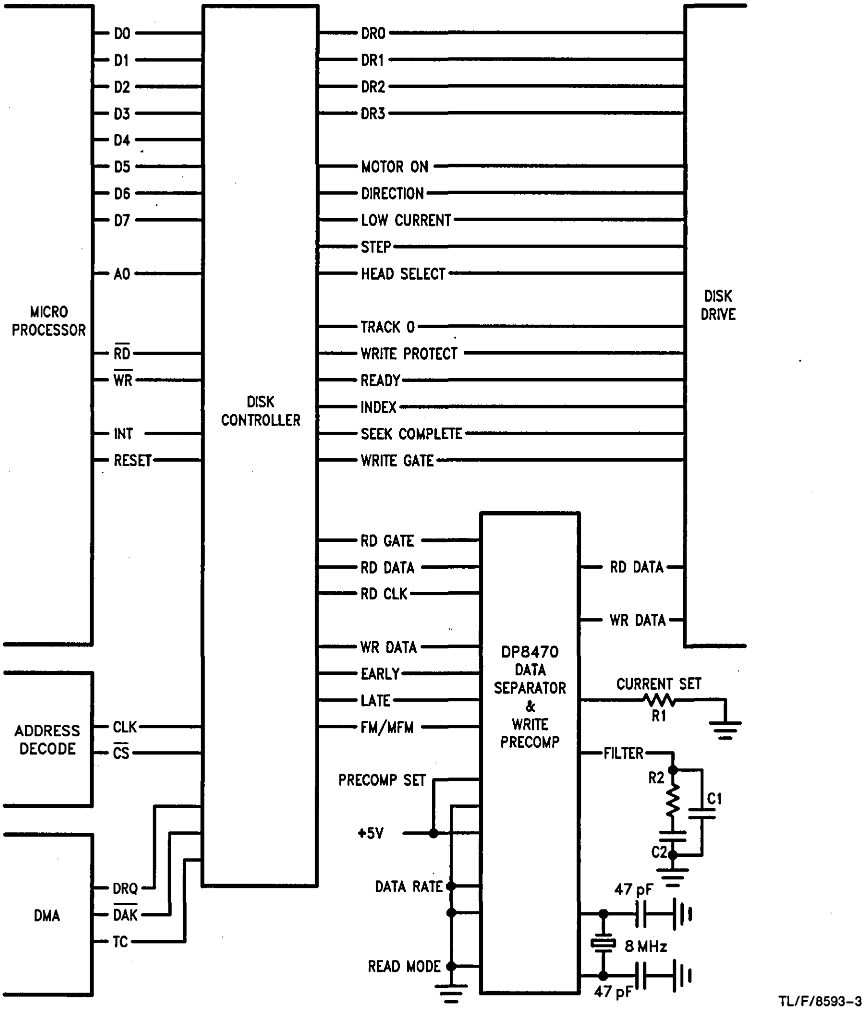
Pin Descriptions

Symbol	DIP Pin No.	PCC Pin No.	Function
NRZ READ DATA	1	1	This output will present data read from the disk in NRZ format based on the read clock. This output is set to a high impedance state when Read Gate is not asserted.
AMF	2	2	This output could be used with any controller that needs an indication of a valid address mark. This pin goes high for one bit period when an address mark is detected. This output is set to a high impedance state when Read Gate is not asserted.
EARLY/PRECOMP LATE/PRECOMP	3 4	3 4	These two active-high inputs determine whether the incoming write-data pulse should be shifted early or late in time.
WRITE DATA IN	5	7	Active-high data input from the disk controller.
FM/MFM	6	8	0 = FM, 1 = MFM. This pin also affects the data rate.
READ GATE	7	9	While this input is low, the PLL will lock to its center frequency. While this input is high, the PLL will lock to signal on the Read Data In pin.
READ DATA OUT	8	10	Active-high data output to the floppy controller. The Read Data Out is synchronized to the Read Clock Output.
READ CLOCK	9	11	This is a clock output that has the same frequency as the data rate. This output is always derived from the output of the VCO. While reading data, the VCO is tracking the data rate. When not reading data, the VCO is locked to its reference frequency.
OSC 1,2	10,11	12,13	These two pins enable the connection of a crystal to form the reference oscillator. Optionally an external clock can be used instead. The clock would drive Osc 1 while Osc 2 would be left open.
FILTER	13	15	This pin is the output of the dual-gain charge pump and is also the input to the VCO. A simple filter is attached to this pin.
DATA RATE 0 DATA RATE 1	16 14	19 17	These two pins select the data rate that this chip will sync to: 00 = 125FM/250MFM 01 = 250FM/500MFM 10 = 500FM/1000MFM 11 = Test Mode
PUMP CURRENT	15	18	A resistor is attached to this pin to set the charge pump current.

Pin Descriptions (Continued)

Symbol	DIP Pin No.	PCC Pin No.	Function
READ DATA IN	17	20	Active-high data input from the disk drive.
PRECOMP 0 PRECOMP 1 PRECOMP 2	20 19 18	24 22 21	These three input pins select the amount of write precompensation.
PUMP UP/DOWN	21	25	This active-high output is the logical OR of Pump Up and Pump Down. This is used for diagnostic purposes.
WRITE DATA OUT	22	26	Active-high data output to the floppy drive. This is the same data as is input on the Write Data In pin, except it has been write-precompensated and delayed.
READ MODE	23	27	This input determines what read algorithm is used to select between the low and the high gain mode. (Low = 4-state algorithm, high = 2-state algorithm.)
V_{CC} GROUND	24 12	28 14	These pins are the power supply pins for both the digital circuitry and the analog circuitry.

Typical Floppy Disk Drive Application



TL/F/8593-3

Functional Description

The data separator consists of a dual gain analog PLL (Phase Locked Loop). This PLL synchronizes a VCO (Voltage Controlled Oscillator) to the raw data signal read from a disk drive. The Read Clock pin is derived from the VCO. The Read Data Out pin mirrors the Read Data In pin except that it is centered with respect to the Read Clock. In addition, NRZ encoded data is available at the Read Clock. In addition, NRZ encoded data is available at the NRZ Read Data pin.

The PLL consists of three main components, a phase comparator, a filter, and a voltage controlled oscillator (VCO), as shown in the Block Diagram. The basic operation of a PLL is fairly straightforward. The phase comparator detects the difference between the phase of the VCO output and the phase of the raw data being read from the disk. This phase difference is converted to a current which either charges or discharges a filter. The resulting voltage of the filter changes the frequency of the VCO in an attempt to reduce the phase difference between the two signals. A PLL is "locked" when the frequency of the VCO is exactly the same as the average frequency of the read data. This is somewhat of a simplified view because it ignores such topics as loop stability, acquisition time, and filter values.

The external filter simply consists of two capacitors and a resistor as shown in the typical application diagram. Another resistor is used to set the charge pump current.

The quarter period delay line is used to determine the center of a bit cell. It is important that this delay line be as accurate as possible. A typical data separator would normally require an external trim to adjust the delay. An external trim is not required for the DP8472/74 however. A secondary PLL is used to automatically calibrate the delay line. The secondary PLL also calibrates the center frequency of the VCO.

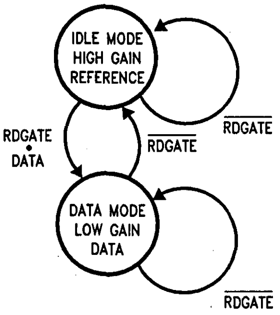
The Preamble Detect circuit is used with the four-state algorithm to determine when the PLL switches from the high gain mode to its low gain mode. This circuit scans the incoming data for the frequency corresponding to a preamble plus or minus 15 percent.

Circuit Operation

READ MODE

There are two read modes to choose from. The Read Mode is selected with the Read Mode pin. The state of this pin should not change during an actual read operation.

Two-State Diagram



Data = first incoming pulse received. TL/F/8593-4

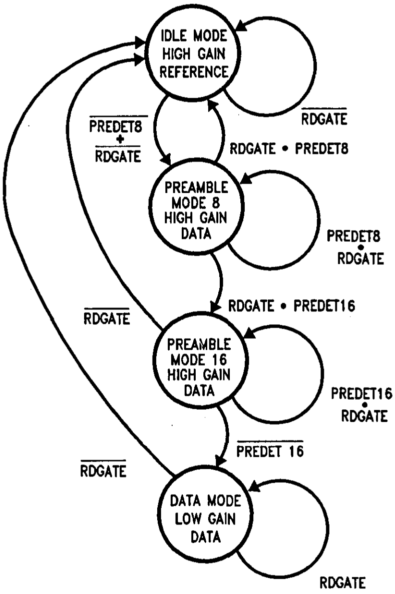
MODE ONE (TWO-STATE DIAGRAM)

When Read Gate is not asserted, the PLL is locked to the crystal frequency in its high-gain mode with a phase/frequency comparator. When Read Gate is asserted, the PLL will remain locked to the crystal until the first data bit arrives.

It will then lock to the incoming data in its low-gain mode with a phase-only comparator. It will stay in this mode until Read Gate is deasserted.

The NRZ Data Output will remain low until 8 bits have been read. This is to guarantee that the clock pulses from the preamble have become stable. The Read Data Out is enabled as soon as the first data bit arrives.

Four-State Diagram



PreDet8 = 8 consecutive bits of preamble frequency. TL/F/8593-5
PreDet16 = 16 consecutive bits of preamble frequency.

MODE TWO (FOUR-STATE DIAGRAM)

When Read Gate is not asserted, the PLL is locked to the crystal frequency in its high-gain mode with a phase/frequency comparator. When Read Gate is asserted, a preamble-detect circuit is enabled. This circuit looks for consecutive bits of the correct preamble frequency. The PLL will stay locked to the crystal frequency until 8 consecutive preamble bits are read. At this point the PLL will lock on to the incoming data (preamble) in its high-gain mode with a phase-only comparator. When the preamble-detect circuit finds 16 consecutive bits of the preamble frequency, the Data Output and the NRZ Data Output logic will be enabled. If at any time before the 16 bits are counted the preamble-detect circuit goes false, the PLL will return to the Idle Mode locked to the crystal. As soon as the preamble-detect circuit goes false after the 16 bits are counted (such as when beginning of the address mark is read) the PLL will switch to its low-gain mode. It will stay in this mode until Read Gate is deasserted. The timing is such that the comparison of the first bit of the Address Mark is done in the low-gain mode.

Between the time in which Read Gate is asserted and the Read Data Output is enabled (states 1 and 2 of the 4-state diagram), the data pattern 4E(hex) or FF(hex) will be output

Circuit Operation (Continued)

for MFM and FM respectively on the Read Data Output pin. The NRZ Data Output will remain low during this time.

WRITE MODE

When writing data, the rising edge of the signal presented to the Write Data Input is delayed before it appears on the Write Data Output. The number of delays is shown in Table I.

The actual value of the delay is determined by the PreComp Set pins. There is also a base delay as specified in the AC timing characteristics.

TABLE I

Early	Late	# of Delays
0	0	1
0	1	2
1	0	0
1	1	illegal

Design Considerations

The operating characteristics of this part are totally pin programmable. The designer needs to set three parameters by tying pins either high or low. These three parameters are the data rate, the amount of write precompensation, and the read mode algorithm.

DATA RATE, FM/MFM

The data rate is determined by three pins (Data Rate 0, Data Rate 1, FM/MFM) and also the clock frequency. The normal clock frequency is 8 MHz. The selectable data rates based on an 8 MHz clock are shown in Table II. If a data rate is needed that is not shown in the 8 MHz column, it may be produced by varying the clock frequency. See the AC Electrical Characteristics for the acceptable range of clock frequencies.

If either of these parameters (data rate or FM/MFM) are subject to change then these pins could be connected to switches, an output port, or through some logic from the controller's drive select output.

TABLE II

Data 1	Rate 0	FM/ MFM	Actual Data Rate (f = 8 MHz)	Actual Data Rate (Variable f)
0	0	0	125 kbits/sec	f/64
0	0	1	250 kbits/sec	f/32
0	1	0	250 kbits/sec	f/32
0	1	1	500 kbits/sec	f/16
1	0	0	500 kbits/sec	f/16
1	0	1	1.00 Mbits/sec	f/8
1	1	0	test mode	
1	1	1	test mode	

f = clock frequency.

The test mode is used by National for testing purposes. It should not normally be used for anything else.

WRITE PRECOMPENSATION

Another parameter to set is the amount of write precompensation needed for the disk drive being used. This value is generally specified by the drive manufacturer. The amount of precompensation used is based on the Precomp Set pins and the Data Rate as shown in Table III.

If the amount of write precompensation is subject to change, then these pins could be connected to switches, an output port, or through some logic from the controller's drive select output.

It is sometimes desirable to enable write precompensation for the inner tracks of a disk only. Some controllers have an output signal that indicates when the head is over a track that needs write precompensation. The easiest way to implement this signal is to choose the amount of write precompensation needed, look up in the table which pins need to be tied high and which need to be tied low. Connect the low pins to ground. Connect the high pins to the controller's write precompensation enable output pin.

CRYSTAL

Normally an 8 MHz crystal is attached in parallel across the two oscillator pins. There should also be a separate 47 pF capacitor attached to each pin with the other side of each capacitor attached to ground. If the system already has an 8 MHz source, this may be used to drive the Osc 1 pin while leaving the Osc 2 pin floating. The frequency at this pin is used to set the center frequency of the VCO and the initial delay of the quarter period delay line. It is also used for the write precompensation circuit timing. See the AC Electrical Characteristics for the acceptable range of the crystal. Varying the frequency will affect many operating parameters as specified in the appropriate sections.

FILTER

The filter is used for the main PLL. The values recommended for the two resistors and the capacitor are given in Table IV based on the data rate needed. If more than one data rate will be used, there are two alternatives. The values can be used that are shown in the table for the multiple data rates. These values are a trade off of PLL characteristics that are not ideal for either data rate. Another alternative is to actually have two separate filters with the capability of switching in one or the other either with a manual switch or an analog switch which can be software controlled.

TABLE III

Precomp Set			Amount of Precompensation					
2	1	0	Data Rt = 00		Data Rt = 01		Data Rt = 10	
			f = 8 MHz	Variable f	f = 8 MHz	Variable f	f = 8 MHz	Variable f
0	0	0	0 ns	0X	0 ns	0X	0 ns	0X
0	0	1	107 ns	3X	36 ns	1X	36 ns	1X
0	1	0	143 ns	4X	71 ns	2X	71 ns	2X
0	1	1	179 ns	5X	107 ns	3X	107 ns	3X
1	0	0	214 ns	6X	143 ns	4X	143 ns	4X
1	0	1	250 ns	7X	179 ns	5X	179 ns	5X
1	1	0	321 ns	9X	214 ns	6X	illegal	
1	1	1	393 ns	11X	250 ns	7X	illegal	

X = 2/7f ns, where f = clock frequency.

TABLE IV

Data Rate	R1	R2	C1	C2
125 kbits/sec FM	kΩ	Ω	nF	μF
250 kbits/sec FM	kΩ	Ω	nF	μF
500 kbits/sec FM	kΩ	Ω	nF	μF
250 kbits/sec MFM	10.0 kΩ	100 Ω	4.7 nF	0.047 μF
500 kbits/sec MFM	kΩ	Ω	nF	μF
1 Mbit/sec MFM	kΩ	Ω	nF	μF
1.25 Mbits/sec MFM	kΩ	Ω	nF	μF
125 FM/250 MFM kbits/sec	kΩ	Ω	nF	μF
250 FM/500 MFM kbits/sec	kΩ	Ω	nF	μF
500 FM/1000 MFM kbits/sec	kΩ	Ω	nF	μF

Interfacing

DISK DRIVE INTERFACE

The connection between the Support Chip and the Disk Drive is very simple. The disk drive's Write Data line connects to the support chip's Write Data Out pin. The disk drive's Read Data line connects to the support chip's Read Data In pin.

FLOPPY CONTROLLER

Simply connect the Write Data and Read Data pins of the controller to the Write Data In and Read Data Out pins of the Support Chip. Connect the Early and Late Precomp outputs from the controller to the Early and Late Precomp inputs of the disk support chip.

The Read Gate input pin of the disk support chip must be connected to the pin of the controller that indicates when the controller is trying to read valid data. On the μPD765A, 8272A this is the VCO pin. On the WD179x this is the VFOE pin.

The Read Clock output pin of the disk support chip must be connected to the pin of the controller that requires a data window (or data clock). This is a window that defines whether an MFM encoded pulse is a data pulse or a clock pulse. The polarity of this pulse is indeterminant. On the μPD765A, 8272A this is the DW pin. On the WD179x this is the RCLK pin.

HARD DISK CONTROLLER

This floppy support chip has been designed to interface directly to the DP8466 Hard Disk Controller. Connect the Write Data lines exactly the same way as for the floppy controller. Connect the Write Precomp lines the same way also.

The hard disk's Read Data line should be connected to the support chip's Read Data In pin. Also, the controller's Read Gate output is connected to the Read Gate input of the support chip. The controller does not use the support chip's Read Data Out pin. The controller needs the data read from the disk to be in NRZ format rather than MFM encoded format. Simply connect the NRZ Read Data pin of the support chip to the Read Data pin of the controller. Connect the Read Clock output of the support chip to the Read Clock input of the controller.

The Address Mark Found output of the support chip gets connected to the Address Mark Found input of the controller.

Note: If write precompensation is used as well as AMF with the DP8466, the signal to indicate early precomp and the signal to indicate AMF must be multiplexed by the Write Gate output of the controller since the DP8466 combines these two functions on the same pin.

PLL Performance

The information in this section is not needed to use this part. It is included for completeness. The performance of the PLL is determined from the following factors:

K_{VCO} — Change in the frequency of the VCO due to a voltage change at the VCO input.

$$K_{VCO} \approx 10 \text{ MRad/s/volt.}$$

I_{CP} — Charge pump current. Set by the external resistor R_1 across the reference voltage set by the chip. $I_{CP} = 1.2\text{ V}/R_1$. This current can be set anywhere between 50 μA and 350 μA. While in the high gain mode, the current is doubled.

C_2 — Filter capacitor.

R_2 — Filter resistor. Determines the PLL damping factor.

C_1 — This filter capacitor improves the performance of the PLL although it has only a secondary effect.

Using second order PLL formulas (i.e. ignoring the effect of C_1) the filter components can be chosen to obtain the required performance.

The bit jitter tolerance of the PLL is given by,

$$\omega_n = (K_{VCO}/2N \times I_{CP}/2\pi \times 1/C_2)^{1/2}$$

where N is the number of VCO cycles between two phase comparisons (N = 2 during the preamble).

The acquisition time (time to lock to the correct phase and frequency) is given by,

$$t_{lk} \approx 6/\omega_n.$$

The trade off, when choosing filter components, is between acquisition time while the PLL is locking and jitter immunity while reading data.

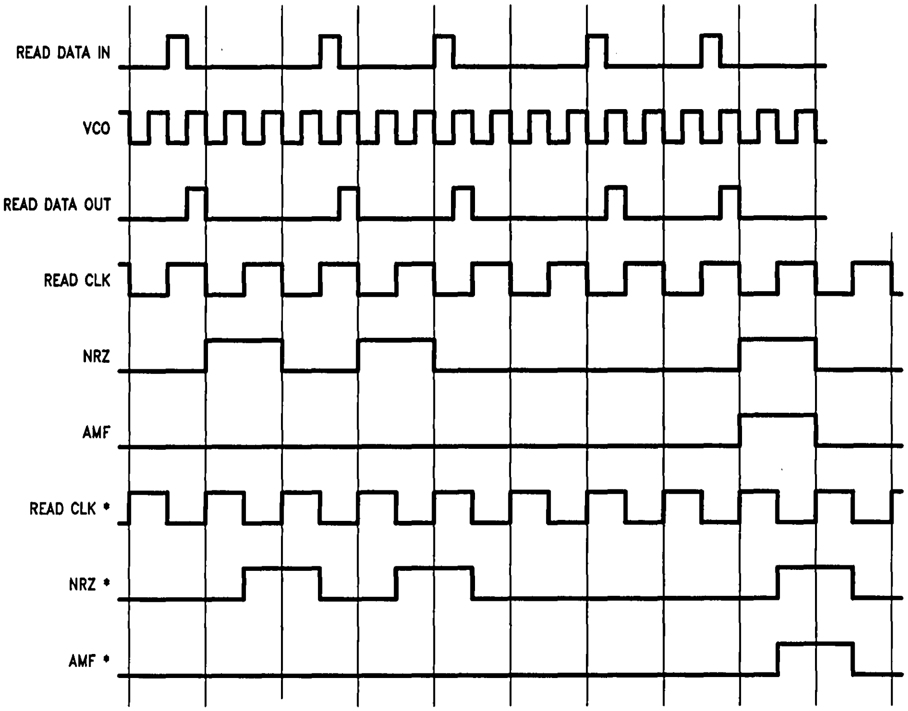
The damping factor is given by,

$$\zeta = \omega_n \times (R_2 \times C_2)/2$$

and is usually set at about 0.7.

Functional Waveform

Read Data Timing



TL/F/8593-6

* = If Read Clock starts out of phase.

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage (V_{CC})	−0.5 to +7.0V
DC Input Voltage (V_{IN})	−1.5 to V_{CC} +1.5V
DC Output Voltage (V_{OUT})	−0.5 to V_{CC} +0.5V
Clamp Diode Current	±20 mA

Operating Conditions

	Min	Max	Units
Supply Voltage (V_{CC})	4.5	5.5	V
DC Input or Output Voltage	0	V_{CC}	V
Operating Temperature Range (T_A):	0	+70	°C

DC Electrical Characteristics V_{CC} = 5V ±10% unless otherwise specified

Symbol	Parameter	Conditions	T_A = 0°C to +70°C Limits	Units
V_{IH}	Minimum High Level Input Voltage		2.0	V
V_{IL}	Maximum Low Level Input Voltage		0.8	V
V_{OH}	Minimum High Level Out	$V_{IN} = V_{IH}$ or V_{IL} $\lvert I_{OUT}\rvert$ = 2 mA	3.7	V
V_{OL}	Maximum Low Level Out	$V_{IN} = V_{IH}$ or V_{IL} $\lvert I_{OUT}\rvert$ = 2 mA	0.4	V
I_{IN}	Maximum Input Current	$V_{IN} = V_{CC}$ or GND	±1.0	μA
I_{OZ}	Maximum TRI-STATE® Leakage Current	$V_{OUT} = V_{CC}$ or GND	±10.0	μA
I_{CC}	Maximum Supply Current	$V_{IN} = V_{CC}$ or GND F_{IN} = 8 MHz	3.0	mA
I_{CC}	Maximum Supply Current	V_{IN} = 2.4V or 0.5V F_{IN} = 8 Mhz	20.0	mA

AC Electrical Characteristics V_{CC} = 5V ±10%, C = 150 pF, unless otherwise specified

Symbol	Parameter	T_A = 0°C to 70°C				Units
		f = 8 MHz		f = variable		
		Min	Max	Min	Max	
f	Crystal Frequency	4	10			MHz
DR	Data Rate	125	1250			Kbit/s
READ TIMING						
t_{DRS}	Data Rate Setup to Data In	Depends on Filter Used				
t_{RMS}	Read Mode Setup to Read Gate	100				ns
t_{FMS}	FM/MFM Setup to Data In	Depends on Filter Used				
t_{RGS}	Read Gate Setup to Data In	600		$100 + \frac{4 \times 10^9}{f}$		ns
t_{DRH}	Data Rate Hold from Read Gate	2 Bit Windows				
t_{RMH}	Read Mode Hold from Read Gate	2 Bit Windows				

AC Electrical Characteristics V_{CC} = 5V ±10%, C = 150 pF, unless otherwise specified (Continued)

Symbol	Parameter	T_A = 0°C to 70°C				Units
		f = 8 MHz		f = variable		
		Min	Max	Min	Max	
READ TIMING (Continued)						
t_{FMH}	FM/MFM Hold from Read Gate	2 Bit Windows				
t_{RGH}	Last Data In to Read Gate Disable	2 Bit Windows				
t_{RGF}	Read Gate Off Time between Reads	600				ns
t_{RDO}	Read Data Offset from Center of Read Clock		34			ns
t_{NRZ}	NRZ & AMF Data Offset from Read Clock Edge		20			ns
t_{IN}	Pulse Width of Data In	50				ns
t_{OUT}	Pulse Width of Data Out	110	200	$\frac{8.8 \times 10^8}{f}$	$\frac{1.6 \times 10^9}{f}$	ns
WRITE TIMING						
t_{DRWS}	Data Rate Setup to Write Data In	125				ns
t_{PSS}	Precomp. Setup to Write Data In	125				ns
t_{RGS}	Read Gate Setup to Write Data In	600		$100 + \frac{4 \times 10^9}{f}$		ns
t_{PS}	Early/Late Setup to Write Data In	−160 (Note 1)				ns
t_{PH}	Early/Late Hold from Write Data In	200				ns
t_{DRWH}	Data Rate Hold from Write Data In	1.0				μs
t_{PSH}	Precomp. Hold from Write Data In	1.0				μs
t_{RGH}	Read Gate Hold from Write Data In	1.0				μs
t_{WI}	Write In Pulse Width	20				ns
t_{WO}	Write Out Pulse Width	220	350	$\frac{1.76 \times 10^9}{f}$	$\frac{2.8 \times 10^9}{f}$	ns
t_{IO}	Write In to Write Out	280 Typ. (Early Precomp.)		$30 + \frac{2 \times 10^9}{f}$		ns
e_{WP}	Error of Write Precomp.		±10			%

Note 1: The Early and Late pins do not need to be valid until 160 ns after the rising edge of the write data in signal. This is to accommodate interfacing to the μPD765A.

PLL Characteristics

Symbol	Parameter	Value	
$K_{\phi High}$	Phase Comparator & Charge Pump Gain Constant. High Gain Mode. (Note 1)	$\frac{5\,V_{REF}}{2\pi R}$	Typ.
V_{REF}	Voltage at Set Pump Current Pin	1.2V	Typ.
$K_{\phi Low}$	Low Gain Mode (Note 1)	$\frac{2.5\,V_{REF}}{2\pi R}$	Typ.
K_{VCO}	Gain of VCO (Note 2)	5/N MRad/S/V	Typ.
f_{VCO}	Center Frequency of VCO	f/2	Typ.
t_{JITTER}	Maximum Tolerance of Bit Jitter (Note 3)	$\frac{(0.95)}{4 \times DR}$	Typ.
$t_{POWER\ ON}$	Time from Full V_{CC} Power to Guaranteed Functionality	50 ms	Max

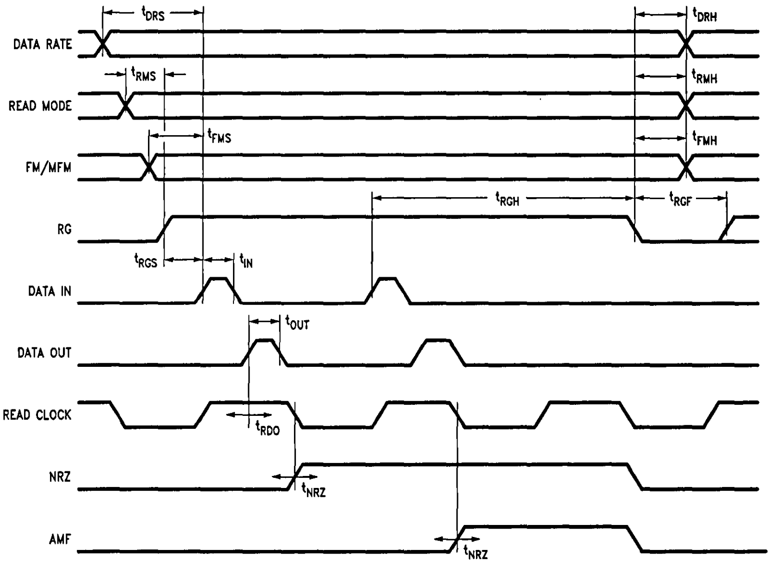
Note 1: R = pump current set resistor (8k–20k).

Note 2: N = # of VCO cycles per bit.

Note 3: DR = Data Rate.

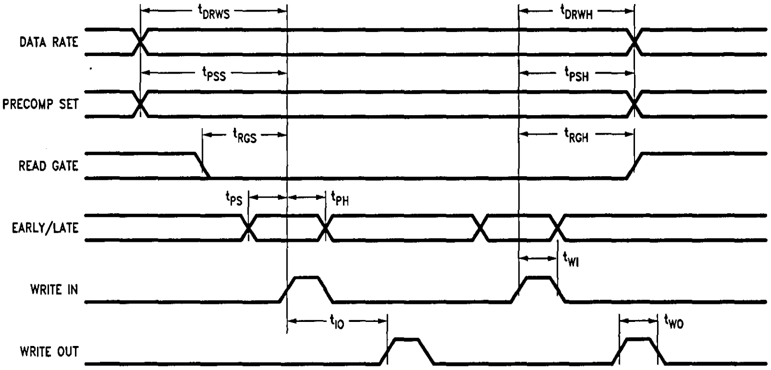
Timing Diagrams

Read Timing



TL/F/8593-7

Write Timing



TL/F/8593-8

National Semiconductor Corporation

PRELIMINARY

DP8472, DP8474 Floppy Disk Controller Plus

General Description

This is a full featured Floppy Disk Drive Controller. It is software compatible with the μPD765A but also has many enhancements over the μPD765A. This includes an internal data separator, internal write precompensation, a motor on/off control, internal line drivers, low power mode, and some software enhancements that simplify programming the DP8472, 74.

The internal data separator uses a combination of digital and analog circuits. The analog PLL requires only fixed value external components, no trims are needed.

The internal Write Precompensation can be programmed to shift the outgoing data early or late anywhere between 0 and 464 ns. It uses a standard single level shifting algorithm.

The Head Load and Head Unload timers can be redefined as a Motor On and Motor Off time. This redefinition allows the longer times needed for a disk drive motor to come up to speed (up to four seconds).

The low power feature allows the crystal of the controller to be turned off by software control or it may be programmed to turn off automatically when all drive motors are off. This reduces the power consumption of the controller to less than 100 μA.

The output buffers of the signals to the disk drive can sink up to 8 mA. They can also be active high or active low on the DP8474. If the length of the cable connecting the disk drive to the controller is relatively small, the drive can be connected directly to the controller without any buffers.

Other enhancements that may be enabled or disabled include implied seeks that will automatically move the drive head to the correct position in many commands. This elimi-

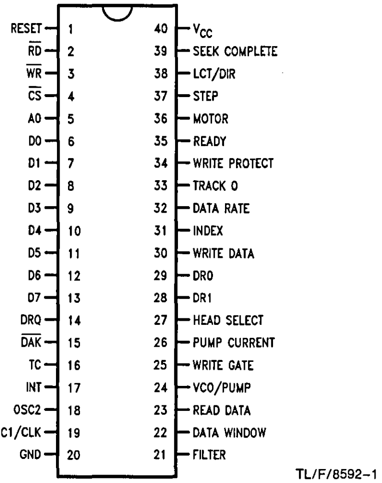
(continued on next page)

Features

- Internal dual gain, analog data separator
- Internal write precompensation (variable starting track #)
- Software compatible with the μPD765A/8272A
- CMOS
- Software selectable data rate (125 kbits–2.5 Mbits)
- No trimmable passive components needed
- Compatible with external data separator
- Low Power mode (I_{CC} < 100 μA)
- Implied seeks on read and write commands
- Disable polling mode
- Can redefine timers for motor on/off
- Seek Complete input for buffered seeks
- Demultiplexed drive select outputs (DP8474)
- Extended track range (up to 4096 tracks)
- Format with or w/o Index Address Mark (μFloppy compatible)

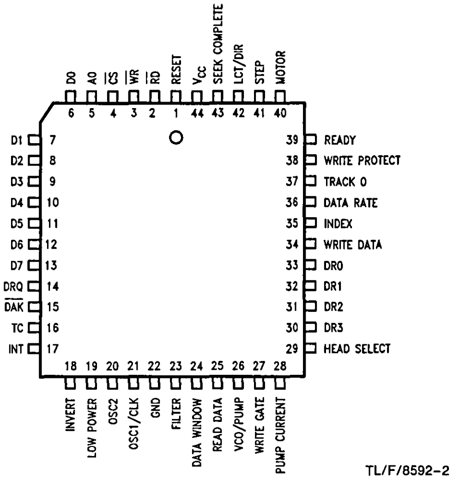
Connection Diagrams

DP8472 Dual-In-Line Package



Top View
Order Number DP8472J or DP8472N
See NS Package Numbers N40A, J40A

DP8474 Plastic Chip Carrier Package



Top View
Order Number DP8474V
See NS Package Number V44A

General Description (Continued)

nates the need for issuing the Seek command and the Sense Interrupt command. Another enhancement is the ability to disable the polling mode. Also, the motor multiplexing scheme may be programmed so that the controller knows whether all the drive motors are on at the same time or if only one is on at a time.

If the command used to control these new features is not accessed, the controller will use default modes that are compatible with software written for use with the μPD765A.

There are two different package types. The DP8472 is a 40 pin DIP package. The DP8474 is a 44 pin PCC package.

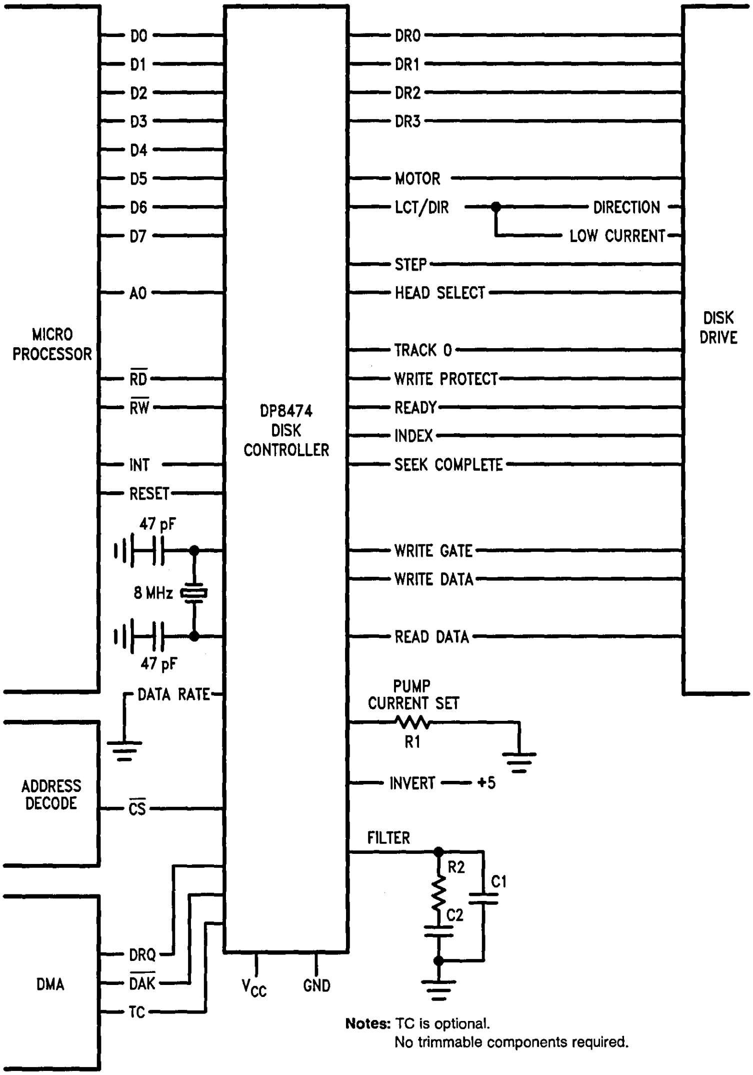
Pin Descriptions

Symbol	DP8472 DIP Pin No.	DP8474 PCC Pin No.	Function
RESET	1	1	Active high input that resets the controller to the idle state and resets all output lines to the disk drive to their disabled state. Does not clear any internal registers except the registers defined in the Mode command. These registers will be set to their default values.
$\overline{RD}$	2	2	Active low input to signal a read from the controller to the microprocessor.
$\overline{WR}$	3	3	Active low input to signal a write from the microprocessor to the controller.
$\overline{CS}$	4	4	Active low input to enable the $\overline{RD}$ and $\overline{WR}$ inputs.
A0	5	5	Address line from the microprocessor. This determines which register the microprocessor is talking to: Data or Status Register.
D0–D7	6–13	6–13	Bi-directional data lines to the microprocessor.
DRQ	14	14	Active high output to signal the DMA controller that a data transfer is needed.
$\overline{DAK}$	15	15	Active low input to acknowledge the DMA request and enable a read or a write.
TC	16	16	Active high input to indicate the termination of a DMA transfer.
INT	17	17	Active high output to signal that an operation requires the attention of the microprocessor. The action required depends on the current function of the controller.
INVERT		18	(DP8474 only) This input determines the polarity of the disk drive interface lines (input & output). Low indicates active high push-pull signals. High indicates active low open drain signals.
LOW POWER		19	(DP8474 only) This active high output indicates when the controller is in its low power mode. This can be connected to a drive that has a low power input signal. Affected by INVERT.
OSC2	18	20	An external crystal is attached here or it is left open if an external clock is used.
CLK/OSC1	19	21	An external crystal or the output of an external clock is attached here. (Usually 8 MHz)
FILTER	21	23	This pin is the output of the dual gain charge pump and is also the input to the VCO. A simple filter is attached to this pin.
DATA WINDOW	22	24	If the internal data separator is disabled, the signal that indicates that the incoming raw data is in the data phase or the clock phase is connected here. This pin is not used if the internal data separator is enabled and should be tied low or high.
READ DATA	23	25	If the internal data separator is enabled, the raw data read from the disk is connected here. If the internal data separator is disabled, the synchronized data signal from an external data separator is connected here. Affected by INVERT.

Pin Descriptions (Continued)

Symbol	DP8472 DIP Pin No.	DP8474 PCC Pin No.	Function
VCO/PUMP	24	26	This active high output enables an external data separator to synchronize to the disk data instead of its center frequency. If the internal data separator is used, this output is the OR of the internal Pump Up and Pump Down signal. This can be used for diagnostic purposes.
WRITE GATE	25	27	This active high output enables the write circuitry of the selected disk drive. Affected by INVERT.
PUMP CURRENT	26	28	A resistor is attached to this pin to set the charge pump current.
HEAD SELECT	27	29	This output determines which disk drive head is active. (low = head 0, high = head 1). Affected by INVERT.
DR0 DR1	29 28		(DP8472) Active high outputs that indicates in binary form which disk drive is active. Affected by INVERT.
DR0 DR1 DR2 DR3		33 32 31 30	(DP8474) Active high outputs to select which disk drive is active. These pins are the demultiplexed DR0, DR1 pins described above. In addition, if no drive is currently selected, no signal will be active. Affected by INVERT.
WRITE DATA	30	34	This is the write precompensated serial data to be written onto the selected disk drive. Affected by INVERT.
INDEX	31	35	This active high input signals the beginning of a track. Affected by INVERT.
DATA RATE	32	36	This input selects the data rate used if the Mode command is not accessed. High indicates 500 kbits/sec (MFM), Low indicates 250 kbits/sec (based on 8 MHz clock). The data rate may be overridden in software through the Mode command. This pin also effects the times programmed with the Specify command. The times are doubled if this pin is low. This doubling effect is not overridden by the Mode Command.
TRACK 0	33	37	This active high input tells the controller that the head is at track zero of the selected disk drive. Affected by INVERT.
WRITE PROTECT	34	38	This active high input tells the controller that the disk is write protected. Any command that writes to the disk drive is not permitted when a disk is write protected. Affected by INVERT.
READY	35	39	This active high input tells the controller that the selected disk drive is ready (i.e. the drive door is closed, may also indicate that the disk is spinning). Affected by INVERT.
MOTOR	36	40	Active high output to turn on the disk drive motor for 5.25″ drives. May also be used to load the head of an 8″ drive. Affected by INVERT.
STEP	37	41	This active high output will produce a pulse at a software programmable rate to move the head during a seek. Affected by INVERT.
LCT/DIR	38	42	When in the seek mode this output will determine the direction of the head movement (high = step in, low = step out). When in the write or read mode this output will go high when the controller detects that the current track number is greater than or equal to a software programmable track number. Affected by INVERT.
SEEK COMPLETE	39	43	This active high input from the disk drive indicates that a buffered seek operation is complete. This input may be tied high if drive does not support buffered seeks. Affected by INVERT.
V_{CC} GROUND	40 20	44 22	These pins are the power supply pins for both the digital circuitry and the analog circuitry.

Typical Application



Notes: TC is optional.
No trimmable components required.

TL/F/8592-3

Functional Description

There are only two registers to access in the floppy disk controller. The read only Main Status Register is used to detect the current status of the controller. The Data Register is used for several purposes. Commands and command parameters are passed to the Data Register. While reading or writing to a disk, the data is transferred through the Data Register. Finally, the result of a command after it is finished is read from the Data Register.

COMMAND SEQUENCE

The disk controller can perform many commands. There are three phases for every command that is executed.

COMMAND PHASE: The μP writes a series of bytes to the Data Register. These bytes indicate the command desired and the particular parameters required for the command.

EXECUTION PHASE: The disk controller performs the desired command. Some commands require the μP to read or write data to or from the Data Register during this time. Reading data from a disk is an example of this.

RESULT PHASE: The μP reads a series of bytes from the Data Register. These bytes indicate whether the command executed properly and other pertinent information.

Each command requires a set of bytes to be written to the disk controller in the Command Phase. All the bytes must be written in the order specified in the Command Description Table. The Execution Phase starts immediately after the last byte in the Command Phase is written.

After the end of the Execution Phase, the Result Phase bytes may be read from the disk controller. The bytes are

Functional Description (Continued)

read in the order specified in the Command Description Table. All the result bytes need not be read, although it is recommended to read them all.

A new command may be initiated by writing the Command Phase bytes after the last bytes needed from the Result Phase have been read.

MAIN STATUS REGISTER (A0 = 0)

The read only Main Status Register indicates the current status of the disk controller. The Main Status Register is always available to be read. One of its functions is to control the flow of data to and from the Data Register. The Main Status Register indicates when the disk controller is ready to send or receive data. It should be read before any byte is transferred to or from the Data Register.

Main Status Register

D7 **Request for Master:** Indicates that the Data Register is ready to send or receive data from the μP. This bit is cleared immediately after a byte transfer and will become set again as soon as the disk controller is ready for the next byte.

D6 **Data Direction:** Indicates whether the controller is expecting a byte to be written to (0) or read from (1) the Data Register. Used in combination with Request for Master (D7).

D5 **Non-DMA Execution:** Bit is set only during the Execution Phase of a command if it is in the non-DMA mode. In other words, if this bit is set, the multiple byte data transfer (in the Execution Phase) must be monitored by the μP either through interrupts, or software polling as described below.

D4 **Command In Progress:** This bit is set after the first byte of the Command Phase is written. This bit is cleared after the last byte of the Result Phase is read. If there is no result phase in a command, the bit is cleared after the last byte of the Command Phase is written.

D3 **Drive 3 Seeking:** Set after the last byte is written in the Command Phase of a Seek or Recalibrate command for drive 3. Cleared after reading the first byte in the Result Phase of the Sense Interrupt Command.

D2 **Drive 2 Seeking:** Same as above for drive 2.

D1 **Drive 1 Seeking:** Same as above for drive 1.

D0 **Drive 0 Seeking:** Same as above for drive 0.

PROCESSOR INTERFACE

Bytes are transferred to and from the disk controller in different ways for the different phases in a command. During the Command Phase and the Result Phase, bytes are transferred using the Main Status Register (A0 = 0) to control the timing and direction of transfer. Bit 6 of the Main Status Register must be clear and bit 7 must be set before a byte can be written to the Data Register (A0 = 1) during the Command Phase. Bits 6 and 7 of the Main Status Register must both be set before a byte can be read from the Data Register during the Result Phase.

If there is information to be transferred during the Execution Phase, there are three methods that can be used. The DMA mode is used if the system has a DMA controller. This allows the μP to do other things during the Execution Phase data transfer. If DMA is not used, an interrupt can be issued for each byte transferred during the Execution Phase. If interrupts are not used, the Main Status Register can be polled to indicate when a byte transfer is required.

DMA MODE

If the DMA mode is selected, a DMA request will be generated in the Execution Phase when each byte is ready to be transferred. The DMA controller should respond to the DMA request with a DMA acknowledge and the $\overline{RD}$ or the $\overline{WR}$ signal. The DMA request will be cleared by the active edge of the DMA acknowledge. After the last byte is transferred, an interrupt is generated. This indicates the beginning of the Result Phase. The interrupt will be cleared by reading the first byte in the Result Phase.

INTERRUPT MODE

If the non-DMA mode is selected, an interrupt will be generated in the Execution Phase when each byte is ready to be transferred. The Main Status Register should be read to verify that the interrupt is for a data transfer. Bits 5 and 7 of the Main Status Register will be set. When the μP reads the data byte from the Data Register, the interrupt will be cleared. The μP should read the byte within the time allotted by the following Time to Service Interrupt Table. If the byte is not transferred within the time allotted, an Overrun Error will be indicated in the Result Phase when the command terminates. An additional interrupt will be generated after the last byte is transferred. This indicates the beginning of the Result Phase. Bits 7 and 6 of the Main Status Register will be set and bit 5 will be clear. This interrupt will be cleared by reading the first byte in the Result Phase.

Time to Service Interrupt

Data Rate	Clk Frequency	Time to Service INT
125 kbits/sec	8 MHz	62.0 μs
250 kbits/sec	8 MHz	30.0 μs
500 kbits/sec	8 MHz	14.0 μs
1000 kbits/sec	8 MHz	6.0 μs
1250 kbits/sec	10 MHz	4.4 μs
2500 kbits/sec	20 MHz	2.2 μs

Time = (8/DR) − (16/f), where DR = Data Rate, f = clock frequency.

SOFTWARE POLLING

If the non-DMA mode is selected and interrupts are not suitable, the μP can poll the Main Status Register during the Execution Phase to determine when a byte is ready to be transferred. During the Execution Phase, in the non-DMA mode, bit 7 of the Main Status Register mirrors the state of the interrupt pin. Otherwise, the data transfer is similar to the Interrupt Mode described above.

DATA RATE

The data rate is determined by three factors; the clock frequency, the data rate pin, and the data rate programmed via the Mode command. Normally an 8 MHz crystal is used as the clock. If this is the case, the data rate pin can be used to select between 250 kbits/sec or 500 kbits/sec (MFM). If a different value clock frequency is used, the data rate is given by the formulas:

data rate = 2×10^{12} / f bits/sec (data rate pin low)

data rate = 4×10^{12} / f bits/sec (data rate pin high)

The Mode command can be used to select the data rate via software. This method gives better flexibility than the data rate pin. With an 8 MHz clock, the following data rates can be selected: 250, 500, or 1000 kbits/sec. With a different clock frequency, the data rate can be calculated by the formulas in the table with the Mode command description.

Functional Description (Continued)

All of the data rates specified in the previous two paragraphs are for MFM encoded data. If FM is used, the data rates are halved.

It is important to note that the internal data separator will not function correctly above a data rate of 1.5 Mbits/sec. The write precompensation logic will also fail above this data rate. Therefore, an external data separator and write precompensation circuit must be used at data rates above 1.5 Mbits/sec.

DATA SEPARATOR

The internal data separator consists of a dual gain analog PLL. This PLL synchronizes the raw data signal read from a disk drive. The synchronized signal is then used to separate the clock and data pulses from the raw signal. The data pulses are grouped into bytes and then sent to the μP.

The PLL consists of three main components, a phase comparator, a filter, and a voltage controlled oscillator (VCO). The basic operation of a PLL is fairly straightforward. The phase comparator detects the difference between the phase of the VCO output and the phase of the raw data being read from the disk. This phase difference is converted to a current which is either charges or discharges a filter. The resulting voltage of the filter changes the frequency of the VCO in an attempt to reduce the phase difference between the two signals. A PLL is "locked" when the frequency of the VCO is exactly the same as the average frequency of the read data. This is somewhat of a simplified view because it ignores such topics as loop stability, acquisition time, and filter values.

The external filter simply consists of two capacitors and a resistor as shown in the typical application diagram. Another resistor is used to set the charge pump current.

The quarter period delay line is used to determine the center of a bit cell. It is important that this delay line be as accurate as possible. A typical data separator would normally require an external trim to adjust the delay. An external trim is not required for the DP8472/74 however. A secondary PLL is used to automatically calibrate the delay line. The secondary PLL also calibrates the center frequency of the VCO.

The quarter period delay line can be programmed to always be set at the ideal delay. It can also be programmed to follow the actual data rate. It does this by following the frequency of the main VCO.

The Preamble Detect circuit is used with an intelligent algorithm to determine when the PLL switches from the high gain mode to its low gain mode. This circuit scans the incoming data for the frequency corresponding to a preamble plus or minus 15 percent.

PLL PERFORMANCE

The information in this section is not needed to use this part. It is included for completeness. The performance of the PLL is determined from the following factors:

K_{VCO}— Change in the frequency of the VCO due to a voltage change at the VCO input.

$$K_{VCO} \approx 10 \text{ MRad/s/volt.}$$

I_{CP}— Charge pump current. Set by the external resistor R_1 across the reference voltage set by the chip. $I_{CP} = 1.2\ V/R_1$. This current can be set anywhere between 50 μA and 350 μA. While in the high gain mode, the current is doubled.

C_2— Filter capacitor.

R_2— Filter resistor. Determines the PLL damping factor.

C_1— This filter capacitor improves the performance of the PLL although it has only a secondary effect.

Using second order PLL formulas (i.e., ignoring the effect of C_1) the filter components can be chosen to obtain the required performance.

The bit jitter tolerance of the PLL is given by,

$$\omega_n = (K_{VCO}/2N \times I_{CP}/2\pi \times 1/C_2)^{1/2}$$

where N is the number of VCO cycles between two phase comparisons (N = 2 during the preamble).

The acquisition time (time to lock to the correct phase and frequency) is given by,

$$t_{lk} \approx 6/\omega_n.$$

The trade off, when choosing filter components is between acquisition time while the PLL is locking and jitter immunity while reading data.

The damping factor is given by,

$$\zeta = \omega_n \times (R_2 \times C_2)/2$$

and is usually set at about 0.7.

Other Features

DRIVE POLLING

If the Polling Mode is enabled, the disk controller will poll the drives continuously while it is in the idle state. The idle state is after the last byte of the Result Phase has been read and before the first byte of the Command Phase has been written. The disk controller will select each drive and check to see if the ready signal has changed states since the last time it checked. If a drive has changed its ready state, an interrupt is generated by the controller. The Sense Interrupt command should be used to identify and clear this interrupt. The Polling Mode can be enabled and disabled through the Mode Command.

LOW POWER MODE

In the Low Power Mode the crystal oscillator is turned off. When the oscillator is turned off the controller will draw less than 100 μA. Also, the internal circuitry is disabled from doing anything when the oscillator is off, since the internal circuitry is driven from this oscillator. The oscillator will turn back on automatically after it detects $\overline{CS}$ & $\overline{RD}$ or $\overline{CS}$ & $\overline{WR}$ being activated. It may take a few milliseconds for the oscillator to return to full frequency, and the μP will be prevented from trying to access the Data Register during this time through the normal Main Status Register protocol. The Controller will go back to low power mode any time it is idle for more than 500 ms (based on 8 MHz clock).

There are two ways to go into the low power mode. One is to command the controller to switch to low power immediately through a software command. The other method is to set the controller to automatically go into the low power mode whenever all the disk drive motors are off (after the Motor Off time expires). This would be invisible to the software. The low power mode is programmed through the Mode Command.

SEEK COMPLETE

The seek complete signal is available on any disk drive that supports buffered seeks. If the drive used does not have this signal available, simply tie this input high.

Result Phase Status Registers

The Result Phase of a command usually contains bytes that hold status information. The format of these bytes are the same for each command and are described below. Do not confuse these bytes with the Main Status Register which is a read only register that is always available. The Result Phase status registers are read from the Data Register only during the Result Phase.

STATUS REGISTER 0 (ST0)

D7 Interrupt Code:

D6 00 = Normal termination of command. Command was completed and properly executed.

01 = Abnormal termination of command. Execution of command was started, but was not successfully completed.

10 = Invalid command issue. Command issued was not recognized as a valid command.

11 = Ready changed state during the polling mode.

D5 Seek End: Seek or Recalibrate command completed by the controller. (Used during Sense Interrupt command.)

D4 Equipment Check: After a Recalibrate command, Track 0 signal failed to occur. (Used during Sense Interrupt command.)

D3 Not Ready: Drive is not ready by 5 disk revolutions after a Read or Write Command. Inverse of current state of ready signal during Sense Interrupt command.

D2 Head Address: (at end of Execution Phase).

D1 Unit Select: (at end of Execution Phase).

D0 00 = Drive 0 selected.
01 = Drive 1 selected.
10 = Drive 2 selected.
11 = Drive 3 selected.

STATUS REGISTER 1 (ST1)

D7 **End of Track:** Controller attempted to access a sector number greater than that programmed by the End of Track (EOT) byte in Command Phase.

D6 **Not Used:** 0.

D5 **CRC Error:** Controller detected a CRC error in the Address Field or the Data Field, depending on the state of bit 5 of ST2.

D4 **Over Run:** Controller was not serviced by the μP soon enough during a data transfer in the Execution Phase.

D3 **Not Used:** 0.

D2 **No Data:** Three possible problems: 1) Controller cannot find the sector specified in the Command Phase during the execution of a Read, Write, or Scan command. An address mark was found however, so it is not a blank disk. 2) Controller cannot read any Address Fields without a CRC error during Read ID command. 3) Controller cannot find starting sector during execution of Read A Track command.

D1 **Not Writable:** Controller detected a write protect signal from the drive during execution of Write Data, Write Deleted Data, or Format A Track commands.

D0 **Missing Address Mark:** If bit 0 of ST2 is clear, then the disk controller cannot detect any Address Field Address Mark after encountering the index hole twice. If bit 0 of ST2 is set, then the disk controller cannot detect the Data Address Mark or Deleted Data Address Mark of the Data Field.

STATUS REGISTER 2 (ST2)

D7 **Not Used:** 0.

D6 **Control Mark:** Controller tried to read a sector which contained a deleted data address mark during execution of Read Data or Scan commands. Or, if a Read Deleted Data command was executed, a regular address mark was detected.

D5 **CRC Error in Data Field:** Controller detected a CRC error in the Data Field. Bit 5 of ST1 is also set.

D4 **Wrong Track:** Only set if desired sector not found. The track number recorded on any sector on the track is different from that stored in the Track Register.

D3 **Scan Equal Hit:** "Equal" condition satisfied during any Scan Command.

D2 **Scan Not Satisfied:** Controller cannot find a sector on the track which meets the desired condition during Scan Command.

D1 **Bad Track:** Only set if the desired sector is not found. The track number recorded on any sector on the track is different from that stored in the Track Register and the recorded track number is FF.

D0 **Missing Address Mark in Data Field:** Controller cannot find a Data Address Mark during a read command. Bit 0 of ST1 is also set.

STATUS REGISTER 3 (ST3)

D7 **Not used:** 0.

D6 **Write Protect Signal.**

D5 **Ready.**

D4 **Track 0**

D3 **Not used:** 0.

D2 **Head Address.**

D1 00 = Drive 0 selected.
D0 01 = Drive 1 selected.
10 = Drive 2 selected.
11 = Drive 3 selected.

COMMAND DESCRIPTION TABLE

READ DATA

Command Phase:

MT	MFM	SK	0	0	1	1	0
IPS	0	0	0	0	HD	DR1	DR0
Track Number							
Head Number							
Sector Number							
Number Data Bytes/Sector							
End of Track							
Gap Length							
Data Length							

Result Phase:

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Number Data Bytes/Sector

READ DELETED DATA

Command Phase:

MT	MFM	SK	0	1	1	0	0
IPS	0	0	0	0	HD	DR1	DR0
Track Number							
Head Number							
Sector Number							
Number Data Bytes/Sector							
End of Track							
Gap Length							
Data Length							

Result Phase:

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Number Data Bytes/Sector

READ A TRACK

Command Phase:

0	MFM	SK	0	0	0	1	0
IPS	0	0	0	0	HD	DR1	DR0
Track Number							
Head Number							
Sector Number							
Number Data Bytes/Sector							
End of Track							
Gap Length							
Data Length							

Result Phase:

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Number Data Bytes/Sector

WRITE DATA

Command Phase:

MT	MFM	0	0	0	1	0	1
IPS	0	0	0	0	HD	DR1	DR0
Track Number							
Head Number							
Sector Number							
Number Data Bytes/Sector							
End of Track							
Gap Length							
Data Length							

Result Phase:

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Number Data Bytes/Sector

WRITE DELETED DATA

Command Phase:

MT	MFM	0	0	1	0	0	1
IPS	0	0	0	0	HD	DR1	DR0
Track Number							
Head Number							
Sector Number							
Number Data Bytes/Sector							
End of Track							
Gap Length							
Data Length							

Result Phase:

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Number Data Bytes/Sector

READ ID

Command Phase:

0	MFM	0	0	1	0	1	0
0	0	0	0	0	HD	DR1	DR0

Result Phase:

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Number Data Bytes/Sector

Notes: MT = Multi-Track
SK = Skip
IPS = Implied Seek
HD = Head #
DR = Drive Select

COMMAND DESCRIPTION TABLE (Continued)

SCAN EQUAL

Command Phase:

MT	MFM	SK	1	0	0	0	1
IPS	0	0	0	0	HD	DR1	DR0
Track Number							
Head Number							
Sector Number							
Number Data Bytes/Sector							
End of Track							
Gap Length							
Sector Step Process							

Result Phase:

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Number Data Bytes/Sector

FORMAT A TRACK

Command Phase:

0	MFM	0	0	1	1	0	1
0	0	0	0	0	HD	DR1	DR0
Number Data Bytes/Sector							
Sectors per Track							
Gap Length							
Data Pattern							

Result Phase:

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Number Data Bytes/Sector

Notes: *—This byte only written or read if the extended track range mode is enabled. (12 bit head #)

TMR = Timer mode
IDX = No index address mark
IPS = Implied Seek
MMX = All motors assumed on if any on
POL = Polling mode
HDR = Extended track range
DRE = Software data rate enable
ANR = Abort Not Ready
EL = Early/Late output enable
WLD = No Wildcard in scan
DTS = Internal data separator
¼ = ¼ period delay tracks data

SCAN LOW OR EQUAL

Command Phase:

MT	MFM	SK	1	1	0	0	1
IPS	0	0	0	0	HD	DR1	DR0
Track Number							
Head Number							
Sector Number							
Number Data Bytes/Sector							
End of Track							
Gap Length							
Sector Step Process							

Result Phase:

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Number Data Bytes/Sector

SEEK

Command Phase:

0	0	0	0	1	1	1	1
0	0	0	0	0	HD	DR1	DR0
New Track Number (NTN)							
*MSB of NTN				0	0	0	0

No Result Phase

SENSE INTERRUPT STATUS

Command Phase:

0	0	0	0	1	0	0	0

Result Phase:

Status Register 0							
Present Track Number (PTN)							
*MSB of PTN				0	0	0	0

SPECIFY

Command Phase:

0	0	0	0	0	0	1	1
Step Rate Time				Motor Off Time			
Motor On Time							DMA

No Result Phase

SET TRACK

Command Phase:

0	$\overline{R}$/W	1	0	0	0	0	1
Internal Register #						DR1	DR0
New Value (or dummy)							

Result Phase:

Value

SCAN HIGH OR EQUAL

Command Phase:

MT	MFM	SK	1	1	1	0	1
IPS	0	0	0	0	HD	DR1	DR0
Track Number							
Head Number							
Sector Number							
Number Data Bytes/Sector							
End of Track							
Gap Length							
Sector Step Process							

Result Phase:

Status Register 0
Status Register 1
Status Register 2
Track Number
Head Number
Sector Number
Number Data Bytes/Sector

RECALIBRATE

Command Phase:

0	0	0	0	0	1	1	1
0	0	0	0	0	0	DR1	DR0

No Result Phase

SENSE DRIVE STATUS

Command Phase:

0	0	0	0	0	1	0	0
0	0	0	0	0	HD	DR1	DR0

Result Phase:

Status Register 3

MODE

Command Phase:

0	0	0	0	0	0	0	1
TMR	IDX	IPS	MMX	LOW PWR		POL	HDR
Low Current & Precomp Track #							
DRE	ANR	EL	WLD	Head Settle			
DTS	1/4	Data Rt		Write PreComp			

No Result Phase

INVALID COMMAND

Command Phase:

Invalid Codes

Result Phase:

Status Register 0

Command Descriptions

READ DATA

The Read Data op-code is written to the data register followed by 8 bytes as specified in the Command Description table. After the last byte is written, the controller turns on the correct drive and starts looking for the sector specified. Once the sector is found the controller sends the data to the μP. After one sector is finished, the Sector Number is incremented by one and this new sector is searched for. If MT (Multi-Track) is set, both sides of one track can be read. Starting on side zero, the sectors are read until the sector number specified by End of Track is reached. Then, side one is read starting with sector number one.

The Read Data command continues to read until the TC pin is set. This means that the DMA controller should be programmed to transfer the correct number of bytes. TC could be controlled by the μP and be asserted when enough bytes are received. An alternative to these methods of stopping the Read Data command is to program the End of Track to be the last sector number that needs to be read. The controller will stop reading the disk with an error indicating that it tried to access a sector number beyond the end of the track.

The Number of Data Bytes per Sector parameter is defined in Table I. If this is set to zero, the Data Length parameter determines the number of bytes that the controller transfers to the μP. If the data length specified is smaller than 128, the controller still reads the entire 128 byte sector and checks the CRC though only the number of bytes specified by the Data Length parameter are transferred to the μP. If the Number of Bytes per Sector parameter is not zero, the Data Length parameter has no meaning and should be set to FF(hex).

TABLE I

# Bytes/Sector Code	Actual # Bytes
0	128
1	256
2	512
3	1024
4	2048
5	4096
6	8192

After the last byte of the Command Phase is written by the μP, the controller will turn on the correct drive. If the drive was previously off, the Motor On Timer will be enabled.

If the Implied Seek Mode is enabled by both the Mode command and the IPS bit in this command, a Seek is performed to the track number specified in the Command Phase.

The controller then waits for all four of the following conditions to occur. 1) The Motor On time must expire, 2) the Ready line must be true, 3) the Seek Complete input must be true and, 4) if an Implied Seek was performed, the Head Settle Time programmed in the Mode Command must expire. The controller will wait up to 5 disk revolutions (counted by Index Pulses) after the Motor On Timer times out for the Ready and the Seek Complete to become true. If 5 revolutions pass, bit 3 of ST0 (Not Ready) is set and an abnormal termination is indicated (bit 7 and 6 of ST0 is 0 and 1 respectively). If a byte is written by the μP during this waiting time, the command is aborted and an abnormal termination is indicated. This way, if the index signal is not active (no disk in drive), the controller will not hang up forever.

After all these conditions are true, the controller searches for the specified sector by comparing the track #, head #, sector #, and number of bytes/sector given in the Command Phase with the appropriate bytes read off the disk in the Address Fields.

If the correct sector is found, but there is a CRC error in the Address Field, bit 5 of ST1 (CRC Error) is set and an abnormal termination is indicated. If the correct sector is not found, bit 2 of ST1 (No Data) is set and an abnormal termination is indicated. In addition to this, if any Address Field track # is FF, bit 1 of ST2 (Bad Track) is set or if any Address Field track # is different from that specified in the Command Phase, bit 4 of ST2 (Wrong Track) is set.

After finding the correct sector, the controller reads that Data Field. If a Deleted Data Mark is found and the SK bit is set, the sector is not read, bit 6 of ST2 (Control Mark) is set, and the next sector is searched for. If a deleted data mark is found and the SK bit is not set, the sector is read, bit 6 of ST2 (Control Mark) is set, and the read terminates with a normal termination. If a CRC error is detected in the Data Field, bit 5 is set in both ST1 and ST2 (CRC Error) and an abnormal termination is indicated.

If no problems occur in the read command, the read will continue from one sector to the next in logical order (not physical order) until either TC is set or an error occurs.

An interrupt will be generated when the Execution Phase of the Read Data command terminates. The values that will be read back in the Result Phase are shown in Table II. If an error occurs, the result bytes will indicate the sector being read when the error occurred.

Command Descriptions (Continued)

TABLE II

MT	HD	Final Sector Xfered to μP	ID Information at Result Phase			
			Track	Head	Sector	Bytes
0	0	< EOT	NC	NC	S+1	NC
		= EOT	T+1	NC	1	NC
	1	< EOT	NC	NC	S+1	NC
		= EOT	T+1	NC	1	NC
1	0	< EOT	NC	NC	S+1	NC
		= EOT	NC	LSB	1	NC
	1	< EOT	NC	NC	S+1	NC
		= EOT	T+1	LSB	1	NC

EOT = End of Track
NC = No Change
LSB = Least Significant Bit = 1
T = Track # Programmed
S = Last Sector # Read

READ DELETED DATA

This command is the same as the Read Data command except for its treatment of a Deleted Data Mark. If a Deleted Data Mark is read, the sector is read normally. If a regular Data Mark is found and the SK bit is set, the sector is not read, bit 6 of ST2 (Control Mark) is set, and the next sector is searched for. If a regular Data Mark is found and the SK bit is not set, the sector is read, bit 6 of ST2 (Control Mark) is set, and the read terminates with a normal termination.

WRITE DATA

The Write Data command is very similar to the Read Data command except that data is transferred from the μP to the disk rather than the other way around. If the controller detects the Write Protect signal, bit 1 of ST1 (Not Writable) is set and an abnormal termination is indicated.

WRITE DELETED DATA

This command is the same as the Write Data command except a Deleted Data Mark is written at the beginning of the Data Field instead of the normal Data Mark.

READ A TRACK

This command is similar to the Read Data command except for the following. The controller starts at the index hole and reads the Data Fields in their physical order, not their logical order. The controller still does a comparison of the Address Field information with the data programmed in the Command Phase and will set bit 2 of ST1 (No Data) if the comparison fails. If there is a CRC error in the Address Field or the Data Field, the read will continue.

The command will terminate when it has read the number of sectors programmed in the EOT parameter.

READ ID

This command will cause the controller to read the first Address Field that it finds. The Result Phase will contain the header bytes that are read. There is no data transfer during the Execution Phase of this command.

FORMAT A TRACK

This command will format one track on the disk. After the index hole is detected, data patterns are written on the disk including all gaps, address marks, Address Fields, and Data Fields. The exact details of the number of bytes for each field is controlled by the parameters given in the Command Phase. The Data Field consists of the Fill Byte specified in the Command Phase repeated to fill the entire sector.

To allow for flexible formatting, the μP must supply the four Address Field bytes (track, head, sector, size) for each sector formatted during the Execution Phase. In other words, as the controller formats each sector, it will request four bytes through either DMA requests or interrupts. This allows for non-sequential sector interleaving. Some typical values for the Address Field bytes are shown in Table III.

The Format command terminates when the index hole is detected a second time.

Command Descriptions (Continued)

TABLE III

Mode	Actual Sector Size	# Bytes per Sector Code	EOT (Hex)	Gap (Hex)	Format Gap (Hex)
		8″ DRIVES			
FM	128 bytes/sec	0	1A	07	1B
	256	1	0F	0E	2A
	512	2	08	1B	3A
	1024	3	04	47	8A
	2048	4	02	C8	FF
	4096	5	01	C8	FF
MFM	256	1	1A	0E	36
	512	2	0F	1B	54
	1024	3	08	35	74
	2048	4	04	99	FF
	4096	5	02	C8	FF
	8192	6	01	C8	FF
		5.25″ DRIVES			
FM	128	0	12	07	09
	128	0	10	10	19
	256	1	08	18	30
	512	2	04	46	87
	1024	3	02	C8	FF
	2048	4	01	C8	FF
MFM	256	1	12	0A	0C
	256	1	10	20	32
	512	2	08	2A	50
	1024	3	04	80	F0
	2048	4	02	C8	FF
	4096	5	01	C8	FF
		3.5″ DRIVES			
FM	128	0	0F	07	1B
	256	1	09	0E	2A
	512	2	05	1B	3A
MFM	256	1	0F	0E	36
	512	2	09	1B	54
	1024	3	05	35	74

Note: Format Gap is the gap length used only for the Format Command.

SCAN COMMANDS

The Scan commands allow data read from the disk to be compared against data sent from the μP. There are three conditions to choose from: Equal, Less than or Equal, Greater than or Equal. An FF(hex) from either the disk or the μP is used as a don't care byte that will always match true. After each sector is read, if the desired condition has not been met, the next sector is read. The next sector is defined as the current logical sector number plus Sector Step Process (SSP). The Scan command will continue until the scan condition has been met, or if the End of Track has been reached, or if TC is asserted.

It is important to program the End of Track to be a multiple of the Sector Step Process. Otherwise, the end of the track will not be detected.

The result of the command is shown in Table IV.

TABLE IV

Status Reg. 2			
Command	**Bit 2**	**Bit 3**	**Conditions**
Scan Equal	0	1	Disk = μP
	1	0	Disk ≠ μP
Scan Low or Equal	0	1	Disk = μP
	0	0	Disk < μP
	1	0	Disk > μP
Scan High or Equal	0	1	Disk = μP
	0	0	Disk > μP
	1	0	Disk < μP

SEEK

There are two ways to move the disk drive head to the desired track number. Method One is to enable the Implied Seek Mode. This way each individual Read or Write command will automatically move the head to the track specified in the command.

Method Two is using the Seek command. During the Execution Phase of the Seek command, the track number to seek to is compared with the present track number and a step pulse is produced to move the head one track closer to the desired track number. This is repeated at the rate specified by the Specify command until the head reaches the correct track. At this point, an interrupt is generated and a Sense Interrupt command is required to clear the interrupt.

During the Execution Phase of the Seek command the only indication via software that a Seek command is in progress is bits 0–3 (Drive Busy) of the Main Status register. Bit 4 of the Main Status register (Controller Busy) is not set. This allows a Seek command to be issued for another drive even while the first drive is still seeking. This is called a Multiple Seek. All four drives may be seeking at the same time. No other command except the Seek command or the Sense Interrupt command should be issued while a Seek command is in progress.

RECALIBRATE

The Recalibrate command is very similar to the Seek command. It is used to step a drive head out to track zero. Step pulses will be produced until the track zero signal from the drive becomes true. If the track zero signal does not go true before 256 step pulses are issued, an error is generated. If the extended track range mode is enabled, an error is not generated until 4096 pulses are issued.

Multiple recalibrations may be issued just like the Seek command for more than one drive. No other command except the Recalibrate command or the Sense Interrupt command should be issued while a Recalibrate Command is in progress.

Command Descriptions (Continued)

SENSE INTERRUPT STATUS

An interrupt is generated by the controller when any of the following conditions occur:

1. Upon entering the Result Phase of:
   a. Read Data command
   b. Read Deleted Data command
   c. Write Data command
   d. Write Deleted Data command
   e. Read a Track command
   f. Read ID command
   g. Format command
   h. Scan commands
2. During data transfers in the Execution Phase while in the Non-DMA mode
3. Ready Line from a drive changes state
4. Seek or Recalibrate command termination

An interrupt generated for reasons 1 or 2 above occurs during normal command operations and is easily discernible by the μP. During an execution phase in Non-DMA Mode, bit 5 (Execution Mode) in the Main Status register is set to 1. Upon entering Result Phase this bit is set to 0. Reasons 1 and 2 do not require the Sense Interrupt Status command. The interrupt is cleared by reading or writing data to the Controller.

Interrupts caused by reasons 3 and 4 are identified with the aid of the Sense Interrupt Status command. This command resets the interrupt when the command byte is written. Use bits 5, 6, and 7 of ST0 to identify the cause of the interrupt as shown in Table V.

TABLE V

Status Register 0			Cause
Seek End	Interrupt Code		
Bit 5	Bit 6	Bit 7	
0	1	1	Ready Line Changed State
1	0	0	Normal Seek Termination
1	1	0	Abnormal Seek Termination

Issuing a Sense Interrupt Status command without an interrupt pending is treated as an invalid command.

If the extended track range mode is enabled, a third byte should be read in the Result Phase which will indicate the four most significant bits of the Present Track Number. Otherwise, only two bytes should be read.

SPECIFY

The Specify command sets the initial values for three internal timers. The timers have two modes as shown in Table VI. The timer modes are programmed from the Mode command. Mode One should be used if the controller is being interfaced to 8″ drives. This mode could be interpreted as defining the timers to be Head Load and Head Unload timers. Mode Two should be used if a drive motor is being turned off and on as in a 5¼″ drive. The Motor On Time defines the time between when the Motor On signal going high and the start of the Read/Write operation starts. The Motor Off Time defines the time from the end of the Execution Phase of one of the Read/Write commands to the Motor Off state. The Step Rate Time defines the time interval between adjacent step pulses during a Seek, Implied Seek, or Recalibrate command.

The times stated in the table are affected by the Data Rate pin. If the pin is high, the table is correct. If the pin is low, the times in the table should be doubled.

The choice of DMA or Non-DMA operation is made by the Non-DMA bit. When this bit = 1 then Non-DMA mode is selected, and when this bit = 0, the DMA mode is selected.

TABLE VI

Timer	Mode 1		Mode 2	
	Value	Min/Max	Value	Min/Max
Step Rate Time	(16-N) ms	1–16 ms	(16-N) ms	1–16 ms
Motor Off Time	N × 16 ms	0–240 ms	N × 512 ms	0–7.68 Sec
Motor On Time	N × 2 ms	0–254 ms	N × 32 ms	0–4.064 Sec

Note 1: Double all times if Data Rate pin is low

Note 2: Based on 8 MHz clock

SENSE DRIVE STATUS

This two-byte command obtains the status of the Drives. Status Register 3 is returned in the result phase and contains the drive status.

MODE

This command is used to select the special features of the controller. The "*" indicates the default which is used after any reset. This reset default has been chosen to be compatible with the μPD765A.

*** TMR = 0** Timers for motor on and motor off are defined for Mode 1 (see Specify command).

TMR = 1 Timers for motor on and motor off are defined for Mode 2 (see Specify command).

*** IDX = 0** The controller will format tracks with the Index Address Mark included. (Exact IBM standard.)

IDX = 1 The controller will format tracks without including the Index Address Mark. This may increase the storage capability of each track. (Sony standard.)

*** IPS = 0** The implied seek bit in the commands is ignored.

IPS = 1 The implied seek bit in the commands is enabled so that if the bit is set in the command, the Seek will be performed automatically.

*** MMX = 0** Only the motor of the drive selected is assumed on. This means that whenever a new drive is selected, the Motor On time is enabled.

MMX = 1 All drive motors are assumed to be on when any motor is turned on. This eliminates the Motor On time when switching from one drive to another before the Motor Off time expires.

LOW = 00* Completely disables the low power mode.

01 Go into low power mode automatically after all drive motors are off.

10 Not used.

11 Go into low power mode now.

Command Descriptions (Continued)

POL=0 Disable polling mode.

*** POL=1** Enable polling mode.

*** HDR=0** The standard header format is used with 8 bits for the track number.

HDR=1 Header format is the same as above but there are 12 bits of track number. The most significant bits of the track number are in the upper four bits of the head number byte.

Low Current and Precomp Track #

Track number to enable the low current output pin and to enable write precompensation. When the controller is writing to track numbers with a value which is less than this value, write precompensation is disabled and the low current output pin is disabled. Default is track zero.

*** DRE=0** The data rate is determined by the Data Rate pin.

DRE=1 The data rate is determined by the bits set in the DATA RT positions.

ANR Abort Not Ready. The state of this bit, in conjunction with the POL bit, determines how the controller treats the drive ready signal. Table VII describes how the controller responds to the ready pin depending on the state of POL and ANR. Default is ANR=1.

*** EL=0** Early/Late. The drive select signals are demultiplexed onto four output signals.

EL=1 The drive select signals are multiplexed onto DR0 and DR1, while precomp Early appears on DR2, and precomp Late appears on DR3. This is only needed if the clock frequency is greater than 10 MHz which is above the range of the internal write precompensation circuit. This is only used in the DP8474 and should always be set low in the DP8472.

*** WLD=0** Wildcard character. An FF(hex) from either the μP or the disk is interpreted as a wildcard character that will always match true.

WLD=1 The Scan commands do not recognize FF(hex) as a wildcard character.

Head Settle Time allowed for head to settle after an Implied Seek. Time = N × 16 ms, (0–240 ms). Default is 64 ms. (Based on 8 MHz clock).

DTS=0 Disable internal Data Separator. Decoded clock signal goes to the Data Window input. Decoded data goes to the Read Data input.

*** DTS=1** Enable internal Data Separator. The encoded data read from the disk goes to the Read Data input.

*** 1/4=0** Quarter period delay line is always set to the reference period through the secondary PLL.

1/4=1 Quarter period delay line follows the changes in frequency of the data by following the same bias voltage as the Main VCO.

DATA RT Data Rate. After a Reset, the data rate is determined by the Data Rate pin. If the DRE bit is set, the data rate is determined by these two bits as shown in Table VIII.

Write Pre-Comp The value of these four bits determines the amount of write precompensation used when writing to the disk drive as shown in Table IX. The default value is based on the data rate used and can be found in Table VIII.

TABLE VII. Ready Polling Mode Table

POL	ANR	Comments
0	0	Do not check ready ever. All commands execute whether ready is true or not.
0	1	Wait up to 5 revs for ready. If ready is not true, wait up to 5 disk revolutions and if still not ready, abort the command.
1	0	Poll, but do not abort. All commands execute whether ready is true or not, but polling continues and ready change INTs are still issued.
1	1	Poll, and abort immediately command if drive not ready (default).

TABLE VIII. Data Rate Table

DT RT	FM		MFM		Default Precomp	
	f = 8 MHz	Variable f	f = 8 MHz	Variable f	f = 8 MHz	Variable f
00	125 kbits/sec	1X	250 kbits/sec	2X	250 ns	0111
01	250 kbits/sec	2X	500 kbits/sec	4X	143 ns	0100
10	500 kbits/sec	4X	1000 kbits/sec	8X	71 ns	0010

Note: X = 10^{12} / f bits/sec, where f = clock frequency. (See Write Precompensation Table for default precomp with variable f.)

Command Descriptions (Continued)

TABLE IX. Write Precompensation Table

Write Pre-Comp	Amount of Pre-Compensation	
	f = 8 MHz	Variable f
0000	none	0X
0001	36 ns	1X
0010	71 ns	2X
0011	107 ns	3X
0100	143 ns	4X
0101	179 ns	5X
0110	214 ns	6X
0111	250 ns	7X
1000	286 ns	8X
1001	321 ns	9X
1010	357 ns	10X
1011	393 ns	11X
1100	429 ns	12X
1101	464 ns	13X
1110	Illegal	
1111	Default	

Note: X = 2/7f ns, where f = clock frequency.

SET TRACK

This command can be used to read or write any value to or from any internal register. For the typical application this command could be used to inspect or change the value of the internal Present Track Register. This could be useful for disk mistracking errors, where the real current track could be read through the Read ID command and then the Set Track Command can set the internal present track register to the correct value. The internal register # for the least significant byte of the Present Track Register is 0C (hex). If more than 8 bits are being used for the track counter, the upper four bits can be accessed through internal register # 0D (hex).

INVALID COMMAND

If an invalid command (i.e., a command not defined) is received by the controller, then the controller terminates the command. The controller does not generate an interrupt during this condition. Bits 6 and 7 in the Main Status Register are both set to 1's indicating to the processor that the Controller is in the Result Phase and the contents of ST0 must be read. When the system reads ST0 it will find a hex 80 indicating an invalid command was received.

In some applications the user may use this command as a No-Op command to place the controller in a standby or no operation state. Simply issue an illegal command and delay reading the result phase until the controller is needed for another command. During this time, the Controller will not poll the drives.

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage (V_{CC})	−0.5V to +7.0V
DC Input Voltage (V_{IN})	−1.5V to V_{CC} +1.5V
DC Output voltage (V_{OUT})	−0.5V to V_{CC} +0.5V
Clamp Diode Current	±20 mA

Operating Conditions

	Min	Max	
Supply Voltage (V_{CC})	4.5	5.5	V
DC Input or Output Voltage	0	V_{CC}	V
Operating Temperature Range (T_A)	−40	+85	°C

DC Electrical Characteristics V_{CC} = 5V ±10%, unless otherwise specified.

Symbol	Parameter	Conditions	T_A = −40 to +85°C Limits	Units
V_{IH}	Minimum High Level Input Voltage		2.0	V
V_{IL}	Maximum Low Level Input Voltage		0.8	V
I_{OH}	Max Leakage Current Disk Interface	$V_{IN} = V_{IH}$ or V_{IL} Invert = 1	10	μA
V_{OH}	Min High Level Out Disk Interface	$V_{IN} = V_{IH}$ or V_{IL} $\lvert I_{OUT} \rvert$ = 2 mA Invert = 0	3.7	V
V_{OL}	Max Low Level Out Disk Interface	$V_{IN} = V_{IH}$ or V_{IL} $\lvert I_{OUT} \rvert$ = 8 mA	0.4	V
V_{OH}	Min High Level Out All Other	$V_{IN} = V_{IH}$ or V_{IL} $\lvert I_{OUT} \rvert$ = 2 mA	3.7	V
V_{OL}	Max Low Level Out All Other	$V_{IN} = V_{IH}$ or V_{IL} $\lvert I_{OUT} \rvert$ = 2 mA	0.4	V
I_{IN}	Maximum Input Current	$V_{IN} = V_{CC}$ or GND	±1.0	μA
I_{OZ}	Maximum TRI-STATE® Leakage Current	$V_{OUT} = V_{CC}$ or GND	±5.0	μA
I_{CC}	Maximum Supply Current	V_{IN} = 3.4 or 0.8V F_{IN} = 8 MHz	40	mA
I_{CC}	Maximum Supply Current	$V_{IN} = V_{CC}$ or GND F_{IN} = 8 MHz	20	mA
I_{CC}	Maximum Supply Current in Low Power Mode	$V_{IN} = V_{CC}$ or GND	100	μA

AC Characteristics V_{CC} = 5V ±10%, C = 150 pF, unless otherwise specified.

Symbol	Parameter	T = −40° to +85°C		Units
		Min	Max	
f	Crystal Frequency	4	10	MHz
f	External Clock Frequency	1	20	MHz
D Rate	Data Rate	125	2500	Kbit/s
μP READ TIMING				
t_{AR}	Address to Read Strobe	0		ns
t_{RA}	Address Hold from Read Strobe	0		ns
t_{RR}	Read Strobe Width	130		ns
t_{RD}	Read Strobe to Data		130	ns
t_{DF}	Data Hold from Read Strobe	5	45	ns
t_{RI}	Clear INT from Read Strobe		130	ns

AC Characteristics V_{CC} = 5V ±10%, C = 150 pF, unless otherwise specified. (Continued)

Symbol	Parameter	T = −40° to +85°C		Units
		Min	Max	
μP WRITE TIMING				
t_{AW}	Address to Write Strobe	0		ns
t_{WA}	Address Hold from Write Strobe	0		ns
t_{WW}	Write Strobe Width	80		ns
t_{DW}	Data Setup to End of Write Strobe	65		ns
t_{WD}	Data Hold from Write Strobe	5		ns
t_{WI}	Clear INT from Write Strobe		130	ns
DMA TIMING				
t_{AM}	End of DRQ from DAK		75	ns
t_{MA}	DAK from DRQ	0		ns
t_{AA}	DAK Pulse Width	130		ns
t_{MRW}	DRQ to End of Read Strobe	130		ns
t_{MW}	DRQ to Read Strobe	0		ns
t_{MR}	DRQ to Write Strobe	0		ns
DRIVE SEEK TIMING				
t_{DIR}	Direction from Drive Select	6		μs
t_{DRH}	Drive Select Hold from End of Step	3		μs
t_{DST}	Step from Direction	6		μs
t_{DH}	Direction Hold from End of Step	3		μs
t_{STR}	Step Rate	Programmable		
t_{STP}	Step Pulse Width	4		μs

AC Characteristics V_{CC} = 5V ±10%, C = 150 pF, unless otherwise specified. (Continued)

Symbol	Parameter	T = −40° to +85°C		Units
		Min	Max	
DISK READ TIMING				
t_{DRS}	Drive Select from Motor On	15		µs
t_{SR}	Read Data from Last Step Pulse	75		µs
t_{RD}	Read Data Width	20		ns
t_{RDS}	Data Window Setup Time	10		ns
t_{RDH}	Data Window Hold Time			ns
DISK WRITE TIMING				
t_{WGS}	Write Data from Write Gate	200 ns	2 bit times	
t_{WGH}	Write Gate Hold from Write Data	1 bit time	2 bit times	

PLL Characteristics

Symbol	Parameter	Valve	
$K\phi_{High}$	Phase Comparator & Charge Pump Gain Constant. High Gain Mode. (Note 1)	$\frac{5\ V_{REF}}{2\pi R}$	Typ
V_{REF}	Voltage at Set Pump Current Pin	1.2V	Typ
$K\phi_{Low}$	Low Gain Mode (Note 1)	$\frac{2.5\ V_{REF}}{2\pi R}$	Typ
K_{VCO}	Gain of VCO (Note 2)	5/N MRad/S/V	Typ
f_{VCO}	Center Frequency of VCO	f/2	Typ
t_{JITTER}	Maximum Tolerance of Bit Jitter (Note 3)	$\frac{(0.95)}{4 \times DR}$	Typ
$t_{POWER\ ON}$	Time from Full V_{CC} Power to Guaranteed Functionally	50 ms	Max

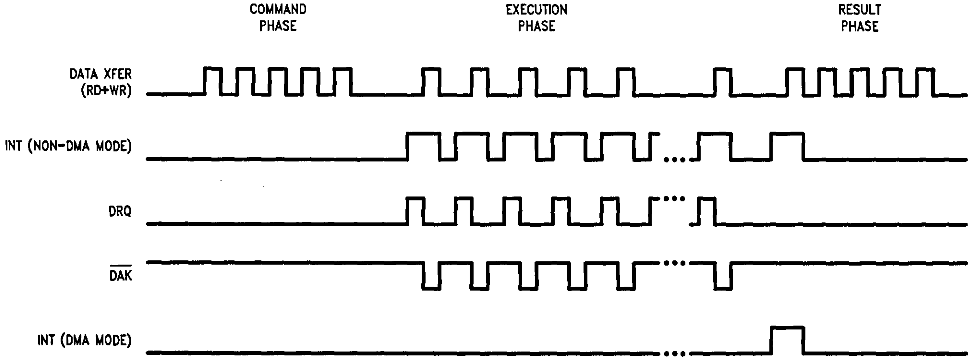
Note 1: R = Pump current set resistor (8k–20k).

Note 2: N = # of VCO cycles per bit.

Note 3: DR = Data Rate.

Timing Waveforms

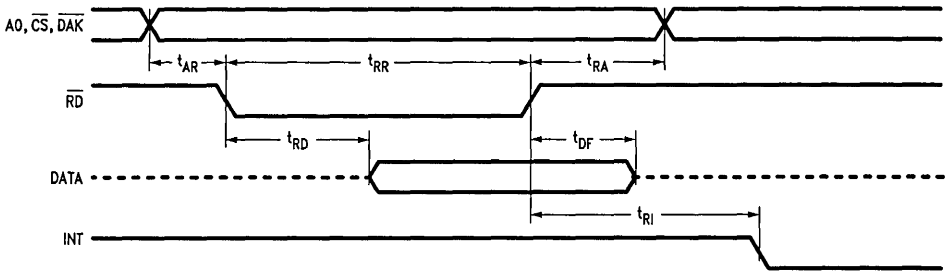
Interrupt and DMA Timing



TL/F/8592-4

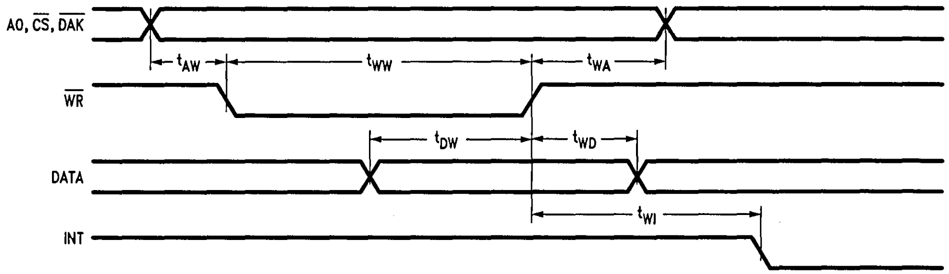
Timing Waveforms (Continued)

μP Read Timing



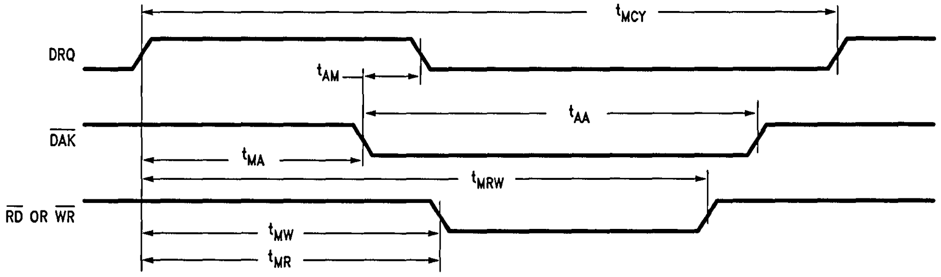
TL/F/8592-5

μP Write Timing



TL/F/8592-6

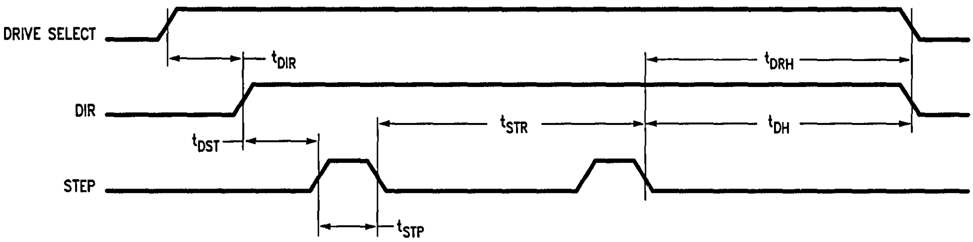
DMA Timing



TL/F/8592-7

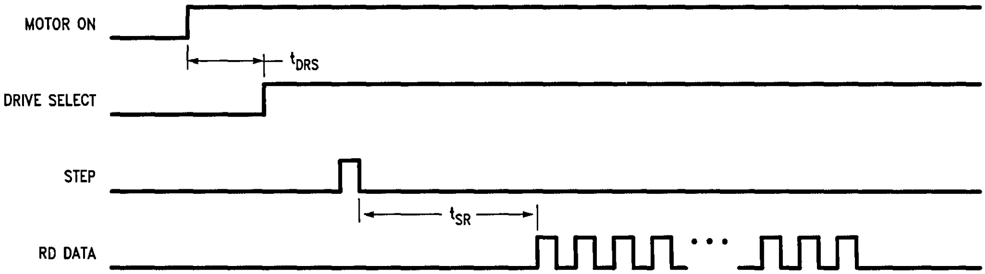
Timing Waveforms (Continued)

Drive Seek



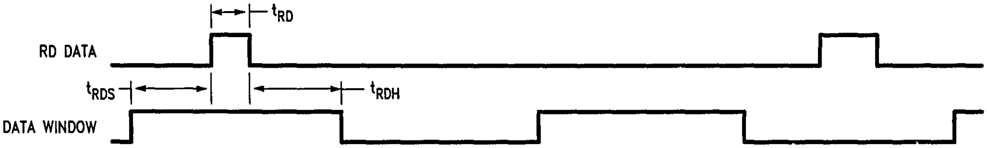
TL/F/8592-8

Disk Read



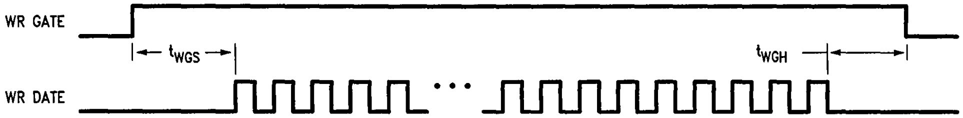
TL/F/8592-9

Read Data



TL/F/8592-11

Disk Write



TL/F/8592-10

Section 5

Drive Interface Support Circuits

Section Contents

National Semiconductor Corporation

PRELIMINARY

DS8921A Differential Line Driver and Receiver Pair

General Description

The DS8921A is a Differential Line Driver and Receiver pair designed specifically for applications meeting the ST506, ST412HP and ESDI Disk Drive Standards. In addition, this device meets the requirements of the EIA Standard RS-422.

The DS8921A receiver offers an input sensitivity of 200 mV over a ±7V common mode operating range. Hysteresis is incorporated (typically 70 mV) to improve noise margin for slowly changing input waveforms. An input fail-safe circuit is provided such that if the receiver inputs are open the output assumes the logical one state.

The DS8921A driver is designed to provide unipolar differential drive to twisted pair or parallel wire transmission lines. Complementary outputs are logically ANDed and provide an output skew of 0.5 ns with propagation delays of 12 ns.

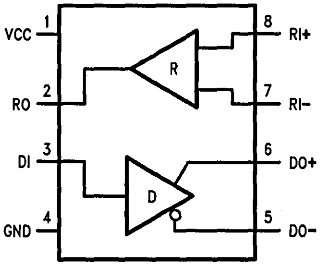
Power up/down circuitry is featured which TRI-STATE® the outputs and prevents erroneous glitches on the transmission lines during system power up or power down operation.

The DS8921A is designed to be compatible with TTL and CMOS.

Features

- 12 ns typical propagation delay
- Output skew - 0.5 ns typical
- Meet the requirements of EIA Standard RS-422
- Complementary Driver Outputs
- High differential or common-mode input voltage ranges of ±7V
- ±0.2V receiver sensitivity over the input voltage range
- Receiver input fail-safe circuitry
- Receiver input hysteresis-70 mV typical
- Glitch free power up/down

Connection Diagram and Truth Table



TL/F/8512-1

Order Number DS8921AJ or N
See NS Package J08A or N08E

Receiver		Driver		
Input	V_{OUT}	Input	V_{OUT}	$\overline{V_{OUT}}$
$V_{ID} \geq V_{TH}$ (MAX)	1	1	1	0
$V_{ID} \leq V_{TH}$ (MIN)	0	0	0	1

Absolute Maximum Ratings (Note 1)

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage	7V
Driver Input Voltage	−0.5V to +7V
Output Voltage	5.5V
Receiver Output Sink Current	50 mA
Receiver Input Voltage	±10V
Differential Input Voltage	±12V
Storage Temperature Range	−65°C to +165°C
Lead Temperature (Soldering, 10 sec.)	300°C

Recommended Operating Conditions

	Min	Max	Units
Supply Voltage	4.5	5.5	V
Temperature (T_A)	0	70	°C

DS8921A Electrical Characteristics (Notes 2, 3 and 4)

Symbol	Conditions	Min	Typ	Max	Units
RECEIVER					
V_{TH}	$-7V \le V_{CM} \le +7V$	−200	±35	+200	mV
V_{HYST}	$-7V \le V_{CM} \le +7V$	25	70		mV
R_{IN}	$-7V \le V_{CM} \le +7V$	4.0	6.0		kΩ
I_{IN}	$V_{IN} = 10V$			3.25	mA
	$V_{IN} = -10V$			−3.25	mA
V_{OH}	$I_{OH} = -400\ \mu A$	2.5			V
V_{OL}	$I_{OL} = 8\ mA$			0.5	V
I_{SC}	$V_{CC} = MAX\ V_{OUT} = 0V$	−15		−100	mA
DRIVER					
V_{IH}		2.0			V
V_{IL}				0.8	V
I_{IL}	$V_{CC} = MAX\ V_{IN} = 0.4V$		−40	−200	µA
I_{IH}	$V_{CC} = MAX\ V_{IN} = 2.7V$			20	µA
I_I	$V_{CC} = MAX\ V_{IN} = 7.0V$			100	µA
V_{CL}	$V_{CC} = MIN\ I_{IN} = -18\ mA$			−1.5	V
V_{OH}	$V_{CC} = MIN\ I_{OH} = -20\ mA$	2.5			V
V_{OL}	$V_{CC} = MIN\ I_{OL} = +48\ mA$			0.5	V
I_{OFF}	$V_{CC} = 0V, V_{OUT} = 5.5V$			100	µA
$\lvert V_T \rvert - \lvert \overline{V_T} \rvert$				0.4	V
V_T		2.0			V
$\lvert V_{OS} - \overline{V_{OS}} \rvert$				0.4	V
I_{SC}	$V_{CC} = MAX\ V_{OUT} = 0V$	−30		−150	mA
DRIVER and RECEIVER					
I_{CC}	$V_{CC} = MAX\ V_{OUT} = Logic\ 0$			35	mA

Receiver Switching Characteristics (*Figures 1* and *2*)

Symbol	Conditions	Min	Typ	Max	Units
T_{pLH}	$C_L = 30$ pF		12		ns
T_{pHL}	$C_L = 30$ pF		12		ns
$\lvert T_{pLH} - T_{pHL} \rvert$	$C_L = 30$ pF		0.5		ns

Driver Switching Characteristics (*Figures 3* and *4*)

SINGLE ENDED CHARACTERISTICS

Symbol	Conditions	Min	Typ	Max	Units
T_{pLH}	$C_L = 30$ pF		12		ns
T_{pHL}	$C_L = 30$ pF		12		ns
T_{TLH}	$C_L = 30$ pF		5		ns
T_{THL}	$C_L = 30$ pF		5		ns
Skew	CL = 30 pF (Note 5)		0.5		ns

Driver Switching Characteristics (*Figures 3* and *5*)

DIFFERENTIAL CHARACTERISTICS (Note 6)

Symbol	Conditions	Min	Typ	Max	Units
T_{pLH}	$C_L = 30$ pF		12		ns
T_{pHL}	$C_L = 30$ pF		12		ns
$\lvert T_{pLH} - T_{pHL} \rvert$	$C_L = 30$ pF		0.5		ns

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The Table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: All currents into device pins are shown as positive values; all currents out of the device are shown as negative; all voltages are referenced to ground unless otherwise specified. All values shown as max or min are classified on absolute value basis.

Note 3: All typical values are $V_{CC} = 5V$, $T_A = 25°C$.

Note 4: Only one output at a time should be shorted.

Note 5: Difference between complementary outputs at the 50% point.

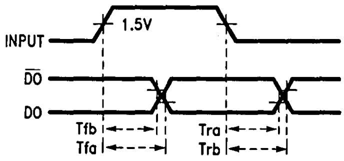
Note 6: Differential Delays are defined as calculated results from single ended rise and fall time measurements. This approach in establishing AC performance specifications has been taken due to limitations of available Automatic Test Equipment (ATE).

The calculated ATE results assume a linear transition between measurement points and are a result of the following equations:

$$T_{cr} = \frac{(T_{fb} \times T_{rb}) - (T_{ra} \times T_{fa})}{T_{rb} - T_{ra} - T_{fa} + T_{fb}}$$

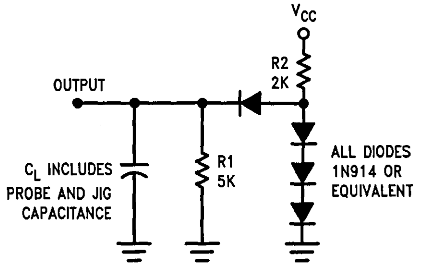
Where: T_{cr} = Crossing Point

T_{ra}, T_{rb}, T_{fa} and T_{fb} are time measurements with respect to the input.



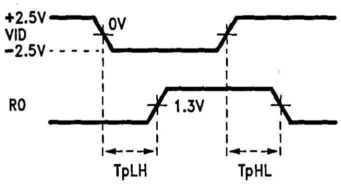
TL/F/8512-2

AC Test Circuits and Switching Diagrams



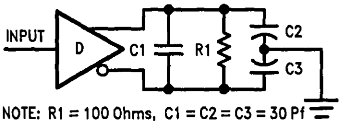
TL/F/8512-3

FIGURE 1



TL/F/8512-4

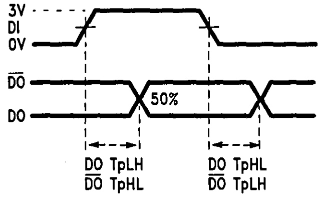
FIGURE 2



NOTE: R1 = 100 Ohms, C1 = C2 = C3 = 30 Pf

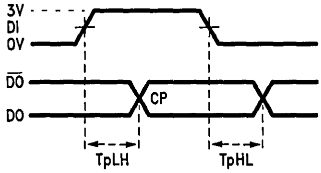
TL/F/8512-5

FIGURE 3



TL/F/8512-6

FIGURE 4

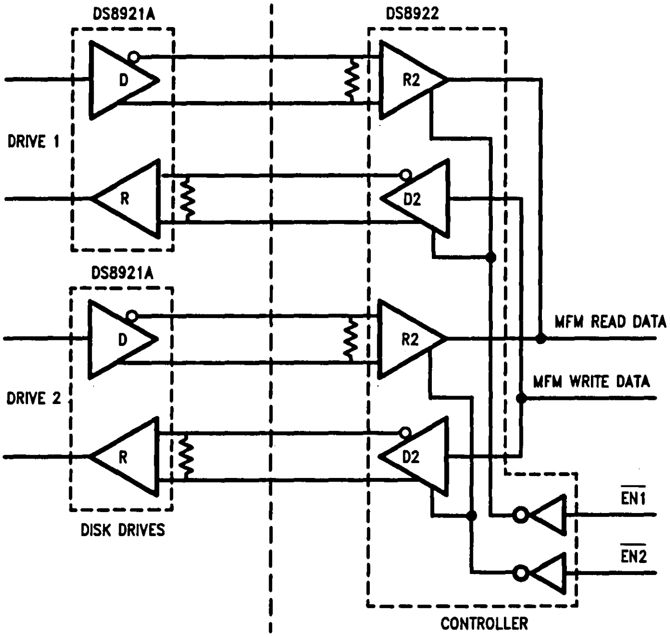


TL/F/8512-7

FIGURE 5

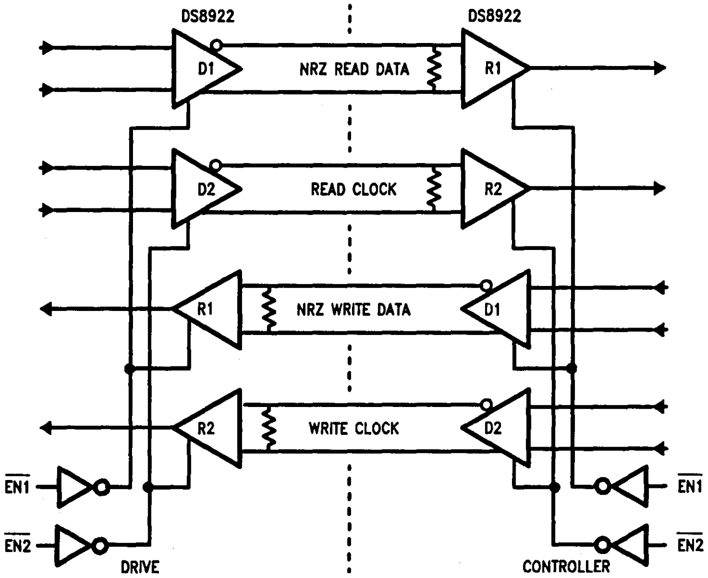
Typical Applications

ST506 and ST412HP Application



TL/F/8512-8

ESDI Application



TL/F/8512-9

National Semiconductor Corporation

PRELIMINARY

DS8922A/DS8923A TRI-STATE® RS-422 Dual Differential Line Driver and Receiver Pairs

General Description

The DS8922A and DS8923A are Dual Differential Line Driver and Receiver pairs. These devices are designed specifically for applications meeting the ST506, ST412HP and ESDI Disk Drive Standards. In addition, both devices meet the requirements of the EIA Standard RS-422.

The DS8922A and DS8923A receivers offer an input sensitivity of 200 mV over a ±7V common mode operating range. Hysteresis is incorporated (typically 70 mV) to improve noise margin for slowly changing input waveforms. An input fail-safe circuit is provided such that if the receiver inputs are open the output assumes the logical one state.

The DS8922A and DS8923A drivers are designed to provide unipolar differential drive to twisted pair or parallel wire transmission lines. Complementary outputs are logically ANDed and provide an output skew of 0.5 ns with propagation delays of 12 ns.

Both devices feature TRI-STATE outputs. The DS8922A has independent control functions common to a driver and receiver pair. When a logic one is applied to the disable input a driver/receiver pair is TRI-STATE while normal operation can be maintained by the other pair. The DS8923A has separate driver and receiver control functions. When a logic one is applied to the driver disable both drivers are TRI-STATE while normal operation is maintained by both receivers.

Power up/down circuitry is featured which TRI-STATE the outputs and prevents erroneous glitches on the transmission lines during system power up or power down operation.

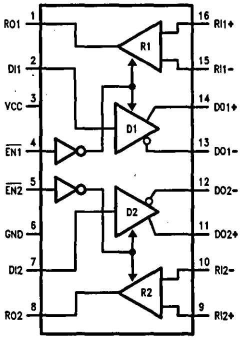
The DS8922A and DS8923A are designed to be compatible with TTL and CMOS.

Features

- 12 ns typical propagation delay
- Output skew—±0.5 ns typical
- Meets the requirements of EIA Standard RS-422
- Complementary Driver Outputs
- High differential or common-mode input voltage ranges of ±7V
- ±0.2V receiver sensitivity over the input voltage range
- Receiver input fail-safe circuitry
- Receiver input hysteresis—±70 mV typical
- Glitch free power up/down
- TRI-STATE outputs

Connection Diagrams

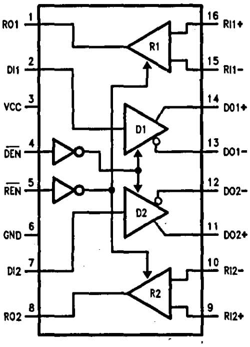
DS8922A Dual-In-Line



TL/F/8511-1

Order Number DS8922AN, J
See NS Package N16A, J16A

DS8923A Dual-In-Line



TL/F/8511-2

Order Number DS8923AN, J
See NS Package N16A, J16A

Truth Tables

DS8922A

$\overline{\text{EN1}}$	$\overline{\text{EN2}}$	RO1	RO2	DO1	DO2
0	0	ACTIVE	ACTIVE	ACTIVE	ACTIVE
1	0	HI-Z	ACTIVE	HI-Z	ACTIVE
0	1	ACTIVE	HI-Z	ACTIVE	HI-Z
1	1	HI-Z	HI-Z	HI-Z	HI-Z

DS8923A

$\overline{\text{DEN}}$	$\overline{\text{REN}}$	RO1	RO2	DO1	DO2
0	0	ACTIVE	ACTIVE	ACTIVE	ACTIVE
1	0	ACTIVE	ACTIVE	HI-Z	HI-Z
0	1	HI-Z	HI-Z	ACTIVE	ACTIVE
1	1	HI-Z	HI-Z	HI-Z	HI-Z

Absolute Maximum Ratings (Note 1)

Supply Voltage	7V
Drive Input Voltage	−0.5V to +7V
Output Voltage	5.5V
Receiver Output Sink Current	50 mA
Receiver Input Voltage	±10V
Differential Input Voltage	±12V
Storage Temperature Range	−65°C to +165°C
Lead Temp. (Soldering, 10 seconds)	300°C

Recommended Operating Conditions

	Min	Max	Units
Supply Voltage	4.5	5.5	V
Temperature (T_A)	0	70	°C

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

DS8922A and DS8923A Electrical Characteristics (Notes 2, 3, and 4)

Symbol	Conditions	Min	Typ	Max	Units
RECEIVER					
V_{TH}	$-7V \leq V_{CM} \leq +7V$	−200	±35	+200	mV
V_{HYST}	$-7V \leq V_{CM} \leq +7V$	25	70		mV
R_{IN}	$-7V \leq V_{CM} \leq +7V$	4.0	6.0		kΩ
I_{IN}	V_{IN} = 10V			3.25	mA
	V_{IN} = −10V			−3.25	mA
V_{OH}	V_{CC} = MIN I_{OH} = − 400 μA	2.5			V
V_{OL}	V_{CC} = MAX I_{OL} = 8 mA			0.5	V
I_{SC}	V_{CC} = MAX V_{OUT} = 0V	−15		−100	mA
DRIVER					
V_{OH}	V_{CC} = MIN I_{OH} = −20 mA	2.5			V
V_{OL}	V_{CC} = MIN I_{OL} = +48 mA			0.5	V
I_{OFF}	V_{CC} = 0V V_{OUT} = 5.5V			100	μA
$\|VT\|-\|\overline{VT}\|$				0.4	V
VT		2.0			V
$\|V_{OS}-\overline{V_{OS}}\|$				0.4	V
I_{SC}	V_{CC} = MAX V_{OUT} = 0V	−30		−150	mA
DRIVER and RECEIVER					
I_{OZ} TRI-STATE Leakage	V_{CC} = MAX V_{OUT} = 2.5V			20	μA
	V_{OUT} = 0.4V			−20	μA
I_{CC}	V_{CC} = MAX ACTIVE			76	mA
	TRI-STATE			78	mA
DRIVER and ENABLE INPUTS					
V_{IH}		2.0			V
V_{IL}				0.8	V
I_{IL}	V_{CC} = MAX V_{IN} = 0.4V		−40	−200	μA
I_{IH}	V_{CC} = MAX V_{IN} = 2.7V			20	μA
I_I	V_{CC} = MAX V_{IN} = 7.0V			100	μA
V_{CL}	V_{CC} = MIN I_{IN} = −18 mA			−1.5	V

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The Table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: All currents into device pins are shown as positive values; all currents out of the device are shown as negative; all voltages are referenced to ground unless otherwise specified. All values shown as max or min are classified on absolute value basis.

Note 3: All typical values are V_{CC} = 5V, T_A = 25°C.

Note 4: Only one output at a time should be shorted.

Receiver Switching Characteristics (*Figures 1, 2* and *3*)

Parameter	Conditions	Min	Typ	Max	Units
T_{pLH}	CL = 30 pF		12		ns
T_{pLH}	CL = 30 pF		12		ns
$\lvert T_{pLH}-T_{pHL}\rvert$	CL = 30 pF		0.5		ns
T_{pLZ}	CL = 15 pF S2 Open		15		ns
T_{pHZ}	CL = 15 pF S1 Open		15		ns
T_{pZL}	CL = 30 pF S2 Open		20		ns
T_{pZH}	CL = 30 pF S1 Open		20		ns

Driver Switching Characteristics (*Figures 4, 5* and *6*)

Parameter	Conditions	Min	Typ	Max	Units
T_{pLH}	CL = 30 pF		12		ns
T_{pHL}	CL = 30 pF		12		ns
T_{TLH}	CL = 30 pF		5		ns
T_{THL}	CL = 30 pF		5		ns
$\lvert T_{pLH}-T_{pHL}\rvert$	CL = 30 pF		0.5		ns
Skew	CL = 30 pF (Note 5)		0.5		ns
T_{pLZ}	CL = 30 pF		15		ns
T_{pHZ}	CL = 30 pF		15		ns
T_{pZL}	CL = 30 pF		20		ns
T_{pZH}	CL = 30 pF		20		ns

Differential Switching Characteristics (Note 6, *Figure 7*)

Parameter	Conditions	Min	Typ	Max	Units
T_{pLH}	CL = 30 pF		12		ns
T_{pHL}	CL = 30 pF		12		ns
$\lvert T_{pLH}-T_{pHL}\rvert$	CL = 30 pF		0.5		ns

Note 5: Difference between complementary outputs at the 50% point.

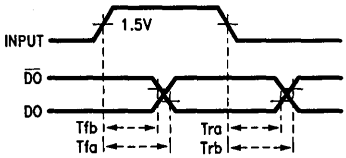
Note 6: Differential Delays are defined as calculated results from single ended rise and fall time measurements. This approach in establishing AC performance specifications has been taken due to limitations of available Automatic Test Equipment (ATE). Table I. shows the results of each parameter specified under Differential Characteristics while figures X, Y, Z show actual measurements recorded using an AC Sampling Scope.

The calculated ATE results assume a linear transition between measurement points and are a result of the following equations:

$$\text{Tcr} = \frac{(\text{Tfb} \times \text{Trb}) - (\text{Tra} \times \text{Tfa})}{\text{Trb} - \text{Tra} - \text{Tfa} + \text{Tfb}}$$

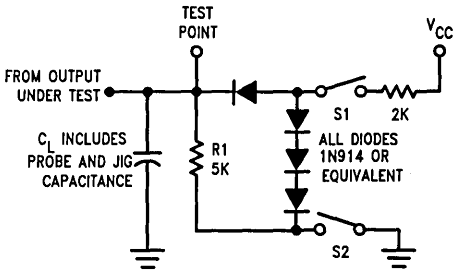
Where: Tcr = Crossing Point

Tra, Trb, Tfa and Tfb are time measurements with respect to the input.



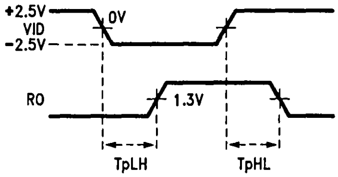
TL/F/8511-3

AC Test Circuits and Switching Waveforms


TEST POINT
V$_{CC}$
FROM OUTPUT UNDER TEST
S1
2K
C$_L$ INCLUDES PROBE AND JIG CAPACITANCE
R1
5K
ALL DIODES 1N914 OR EQUIVALENT
S2

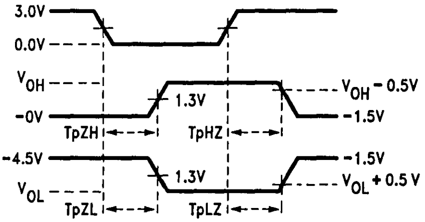
TL/F/8511-4

FIGURE 1



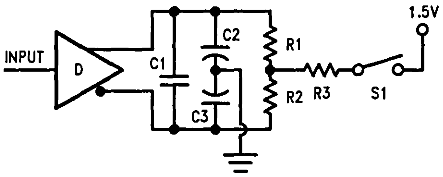
TL/F/8511-5

FIGURE 2



TL/F/8511-6

FIGURE 3

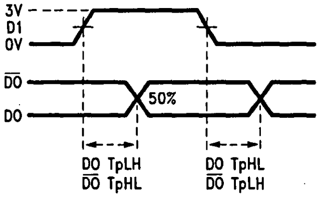


NOTE: C1=C2=C3=30 pF, R1=R2=50 Ω, R3=500 Ω

TL/F/8511-7

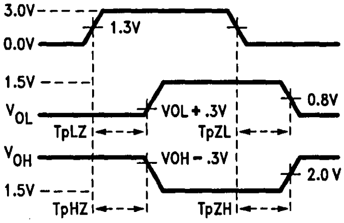
FIGURE 4

AC Test Circuit and Switching Waveforms (Continued)



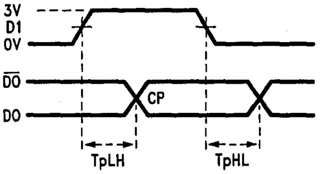
TL/F/8511-8

FIGURE 5



TL/F/8511-9

FIGURE 6

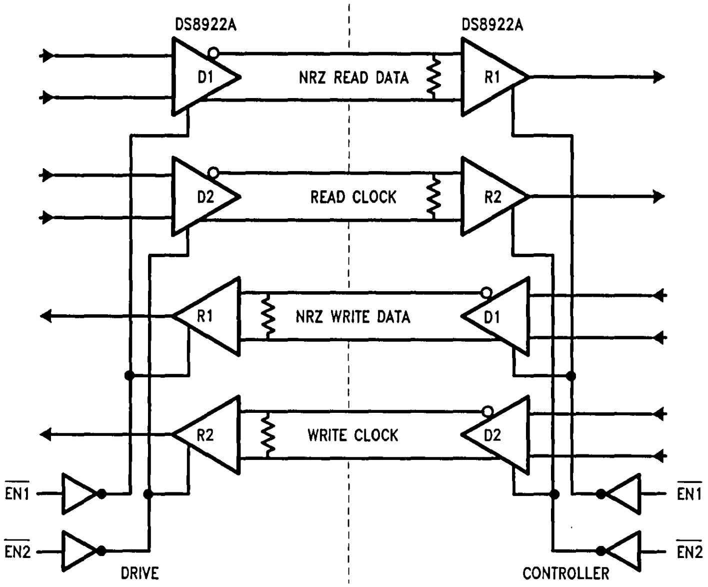


TL/F/8511-10

FIGURE 7

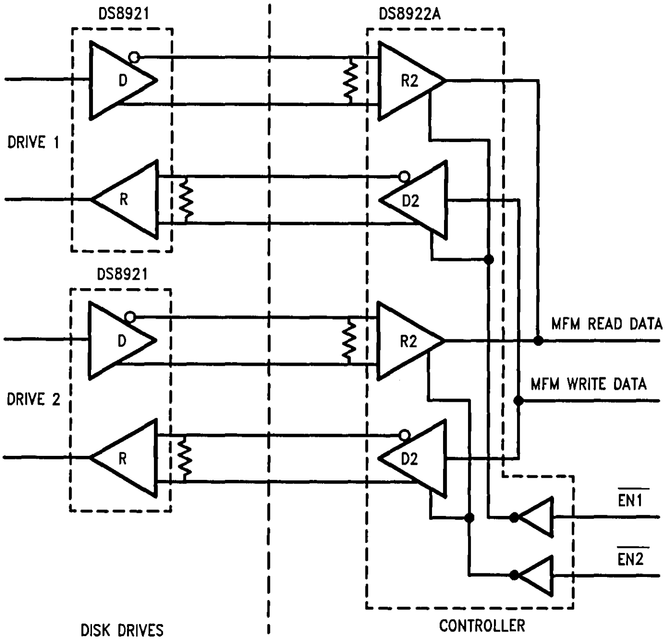
Typical Applications

ESDI Applications



TL/F/8511-11

ST506 and ST412HP Application



TL/F/8511-12

Section 6

Physical Dimensions/ Appendices

Section Contents

National
Semiconductor
Corporation

Bookshelf of Technical Support Information

National Semiconductor Corporation recognizes the need to keep you intormed about the availability of current technical literature.

This bookshelf is a compilation of books that are currently available. The listing that follows shows the publication year and section contents for each book.

Please contact your local National sales office for possible complimentary copies. A listing of sales offices follows this bookshelf.

We are interested in your comments on our technical literature and your suggestions for improvement Please send them to:

Technical Communications Dept. M/S 23-200
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090

For a recorded update of this listing plus ordering information for these books from National's Literature Distribution operation, please call (408)749-7378.

DATA CONVERSION/ACQUISITION DATABOOK—1984

Selection Guides • Active Filters • Amplifiers • Analog Switches • Analog-to-Digital Converters
Analog-to-Digital Display (DVM) • Digital-to-Analog Converters • Sample and Hold • Sensors/Transducers
Successive Approximation Registers/Comparators • Voltage References

HYBRID PRODUCTS DATABOOK—1982

Operational Amplifiers • Buffers • Instrumentation Amplifiers • Sample & Hold Amplifiers • Comparators
Non-Linear Functions • Precision Voltage Regulators and References • Analog Switches
MOS Clock Drivers • Digital Drivers • A-D Converters • D-A Converters • Fiber-Optic Products
Active Filters & Telecommunication Products • Precision Networks • 883/RETS

INTERFACE/BIPOLAR LSI/BIPOLAR MEMORY/PROGRAMMABLE LOGIC DATABOOK—1983

Transmission Line Drivers/Receivers • Bus Transceivers • Peripheral/Power Drivers
Level Translators/Buffers • Display Controllers/Drivers • Memory Support • Dynamic Memory Support
Microprocessor Support • Data Communications Support • Disk Support • Frequency Synthesis
Interface Appendices •Bipolar PROMs • Bipolar and ECL RAMs • 2900 Family/Bipolar Microprocessor
Programmable Logic

INTUITIVE IC CMOS EVOLUTION—1984

Thomas M. Frederiksen's new book targets some of the most significant transitions in semiconductor technology since the change from germanium to silicon. *Intuitive IC CMOS Evolution* highlights the transition in the reduction in defect densities and the development of new circuit topologies. The author's latest book is a vital aid to engineers, and industry observers who need to stay abreast of the semiconductor industry.

INTUITIVE IC OP AMPS—1984

Thomas M. Frederiksen's new book, *Intuitive IC Op Amps,* explores the many uses and applications of different IC op amps. Frederiksen's detailed book differs from others in the way he focuses on the intuitive groundwork in the basic functioning concepts of the op amp. Mr. Frederiksen's latest book is a vital aid to engineers, designers, and industry observers who need to stay abreast of the computer industry.

LINEAR APPLICATIONS HANDBOOK—1986

The purpose of this handbook is to provide a fully indexed and cross-referenced collection of linear integrated circuit applications using both monolithic and hybrid circuits from National Semiconductor.

Individual application notes are normally written to explain the operation and use of one particular device or to detail various methods of accomplishing a given function. The organization of this handbook takes advantage of this innate coherence by keeping each application note intact, arranging them in numerical order, and providing a detailed Subject Index.

LINEAR SUPPLEMENT DATABOOK—1984

Amplifiers • Comparators • Voltage Regulators • Voltage References • Converters • Analog Switches
Sample and Hold • Sensors • Filters • Building Blocks • Motor Controllers • Consumer Circuits
Telecommunications Circuits • Speech • Special Analog Functions

LOGIC DATABOOK VOLUME I—1984

CMOS AC Switching Test Circuits and Timing Waveforms • CMOS Application Notes • MM54HC/MM74HC
MM54HCT/MM74HCT • CD4XXX • MM54CXXX/MM74CXXX • LSI/VLSI

LOGIC DATABOOK VOLUME II—1984

Introduction to Bipolar Logic • Advanced Low Power Schottky • Advanced Schottky • Low Power Schottky
Schottky • TTL • Low Power

MEMORY SUPPORT HANDBOOK—1986

Dynamic Memory Control • Error Checking and Correction • Microprocessor Interface and Applications
Memory Drivers and Support

MOS MEMORY DATABOOK—1984

Standard Terminology • MOS Memory Cross Reference Guide • Dynamic RAMs • NMOS Static RAMs
CMOS Static RAMs • EPROMs • EEPROMs • Military/Aerospace • Reliability

THE NSC800 MICROPROCESSOR FAMILY DATABOOK—1985

CPU • Peripherals • Evaluation Board • Logic Devices • MA2000 Macrocomponent Family

SERIES 32000 DATABOOK—1985

CPUs • Slave Processors • Peripherals • OEM Products • Development Tools • Software

THE SWITCHED-CAPACITOR FILTER HANDBOOK—1985

Introduction to Filters • National's Switched-Capacitor Filters • Designing with Switched-Capacitor Filters
Application Circuits • Filter Design Program • Nomographs and Tables

TRANSISTOR DATABOOK—1982

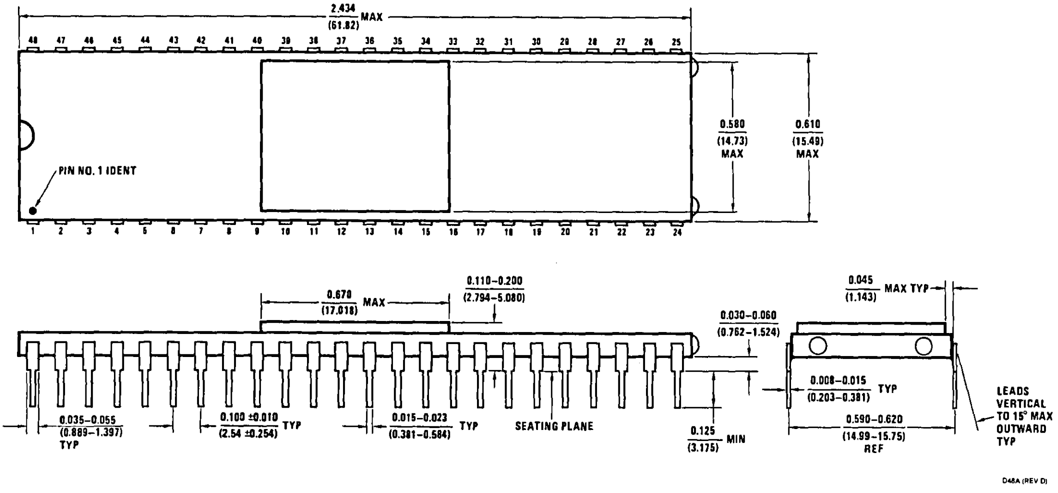
NPN Transistors • PNP Transistors • Junction Field Effect Transistors • Selection Guides • Pro Electron Series
Consumer Series • NA/NB/NR Series • Process Characteristics Double-Diffused Epitaxial Transistors
Process Characteristics Power Transistors • Process Characteristics JFETs • JFET Applications Notes

VOLTAGE REGULATOR HANDBOOK—1982

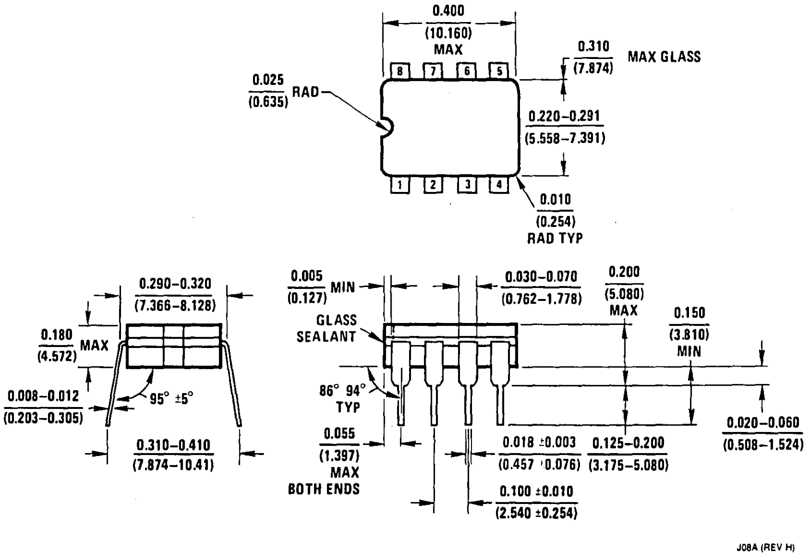
Product Selection Procedures • Heat Flow & Thermal Resistance • Selection of Commercial Heat Sink
Custom Heat Sink Design • Applications Circuits and Descriptive Information • Power Supply Design
Data Sheets

48-SERIES MICROPROCESSOR HANDBOOK—1980

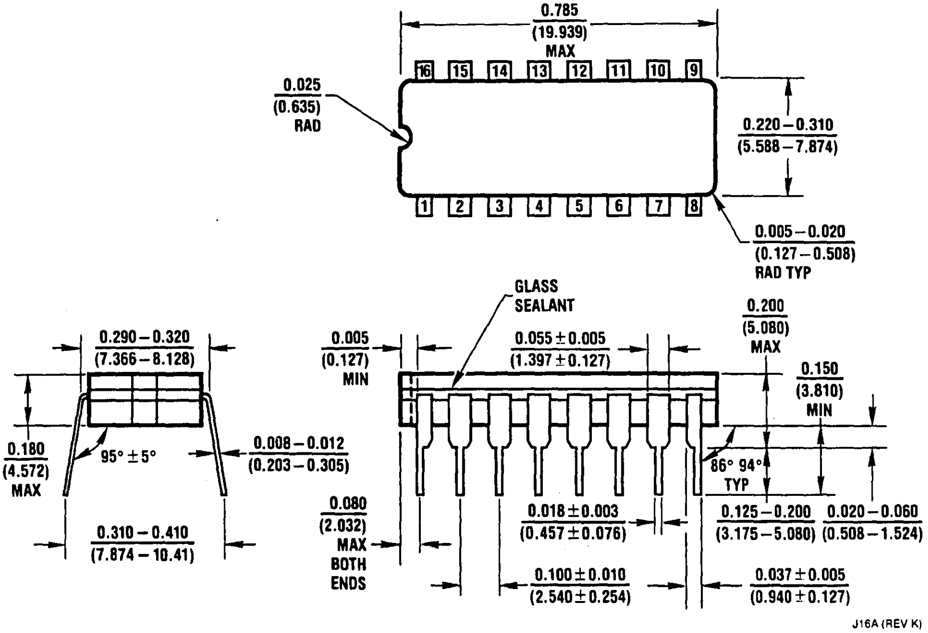
The 48-Series Microcomputers • The 48-Series Single-Chip System • The 48-Series Instruction Set
Expanding the 48-Series Microcomputers • Applications for the 48-Series • Development Support
Analog I/O Components • Communications Components • Digital I/O Components • Memory Components
Peripheral Control Components



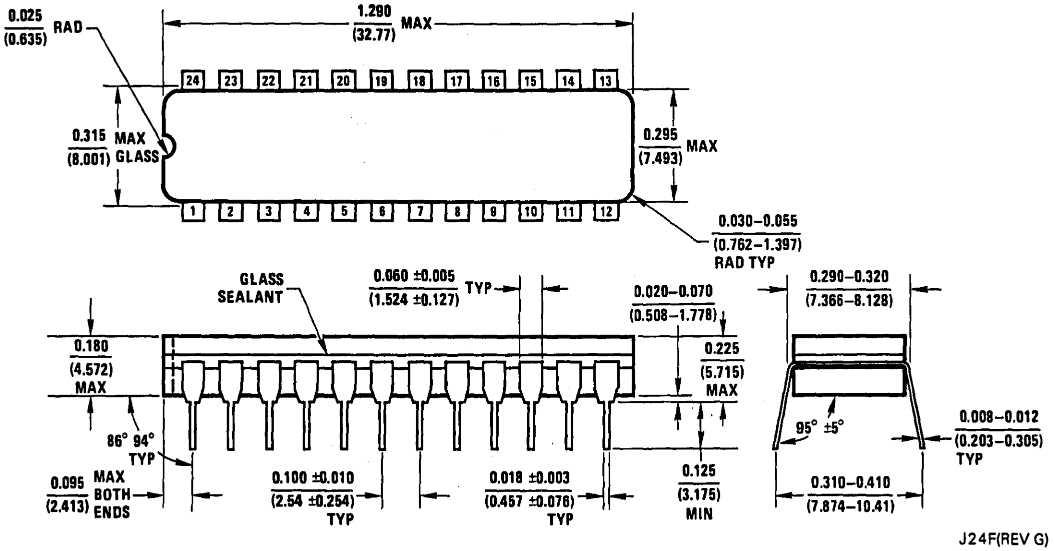
NS Package D48A



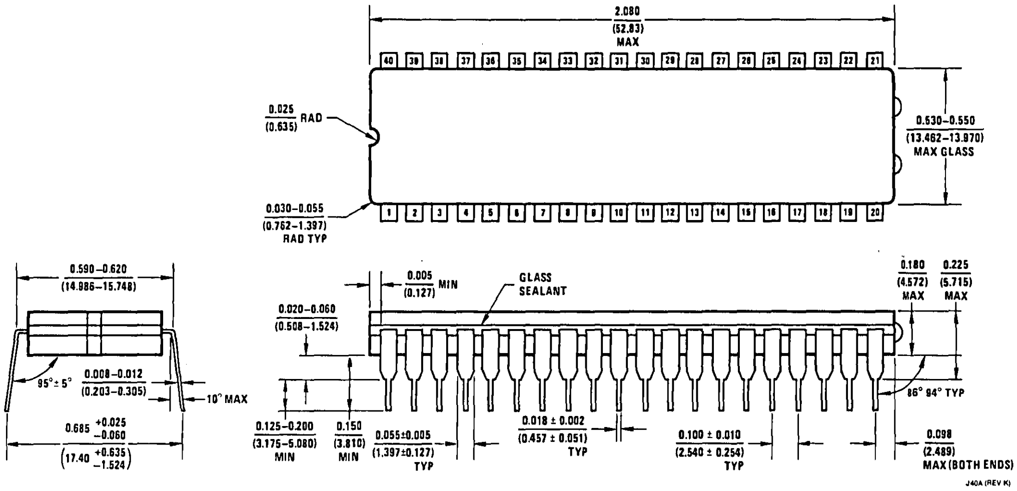
NS Package J08A



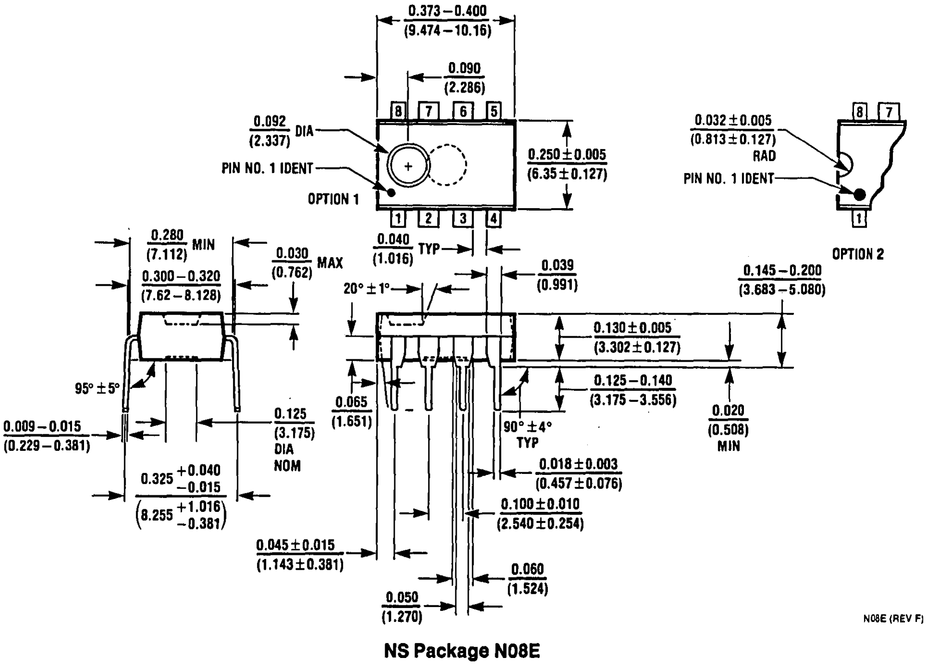
NS Package J16A



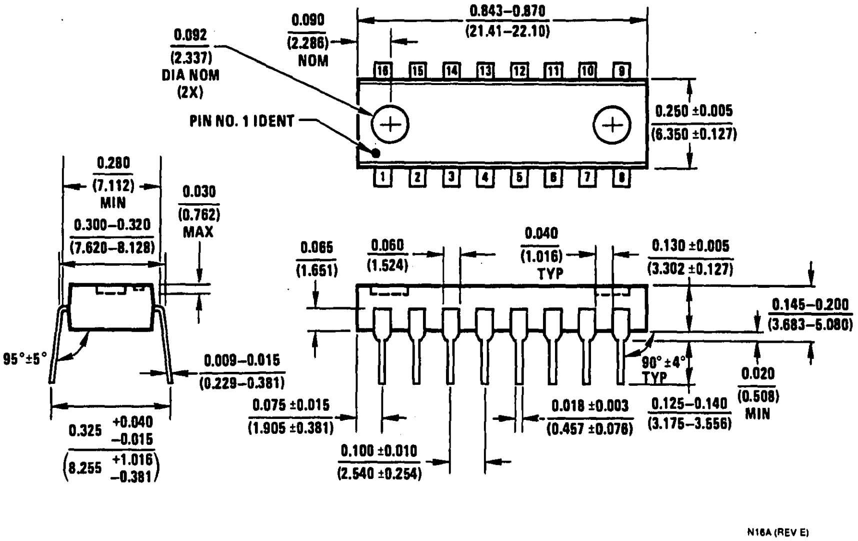
NS Package J24F



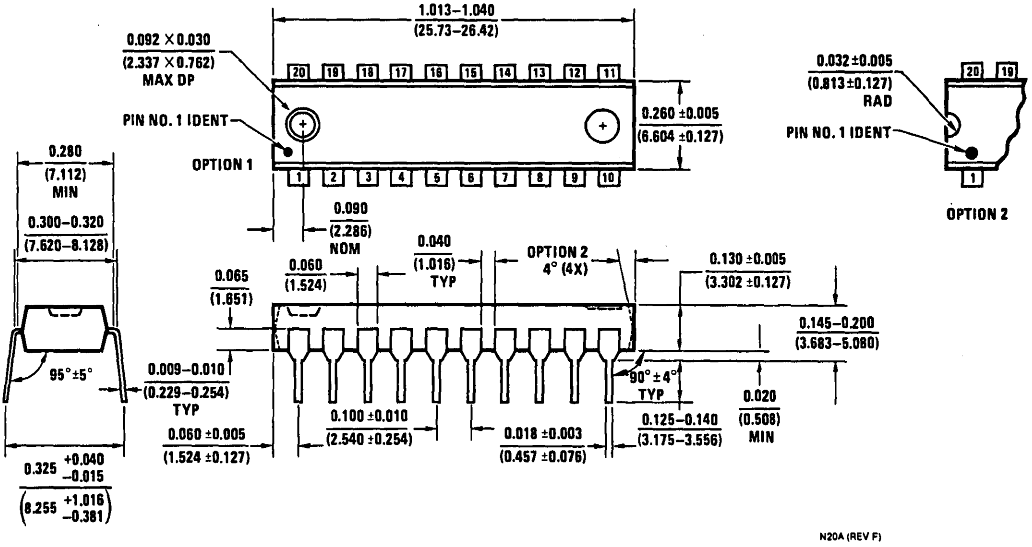
NS Package J40A



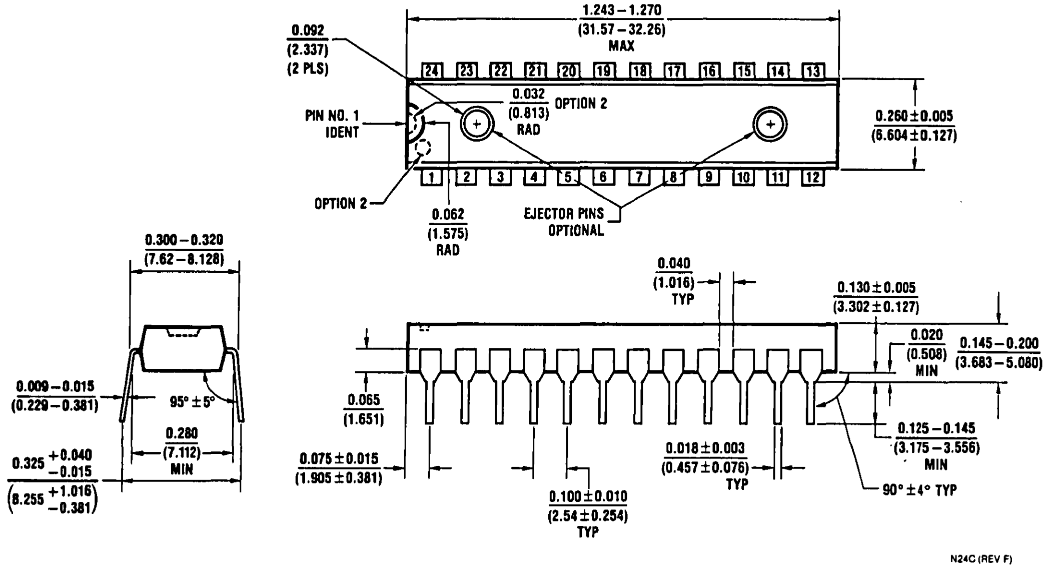
NS Package N08E



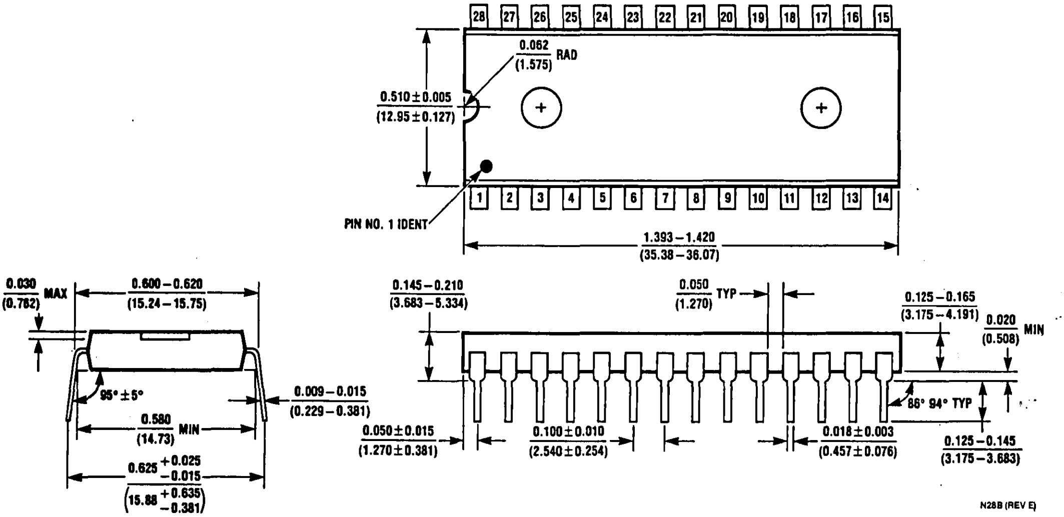
NS Package N16A



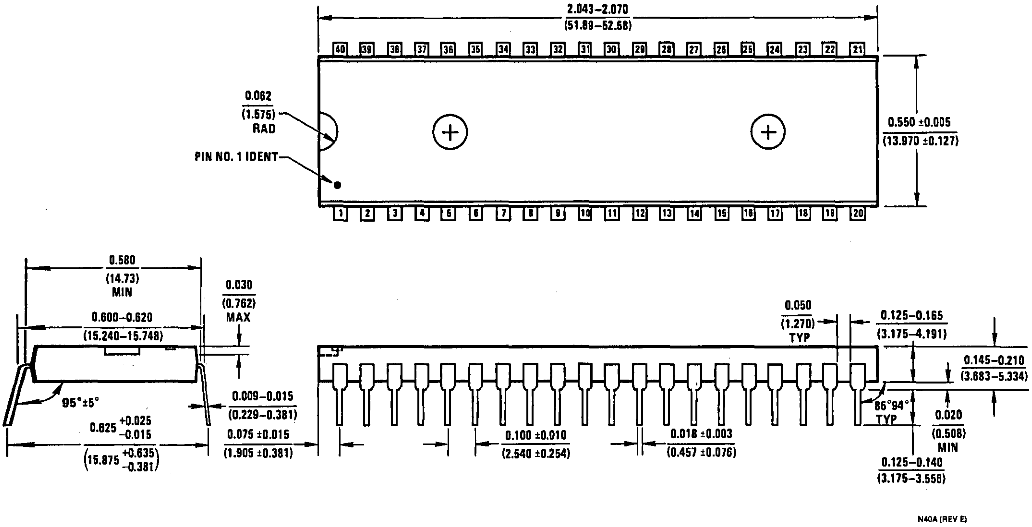
NS Package N20A



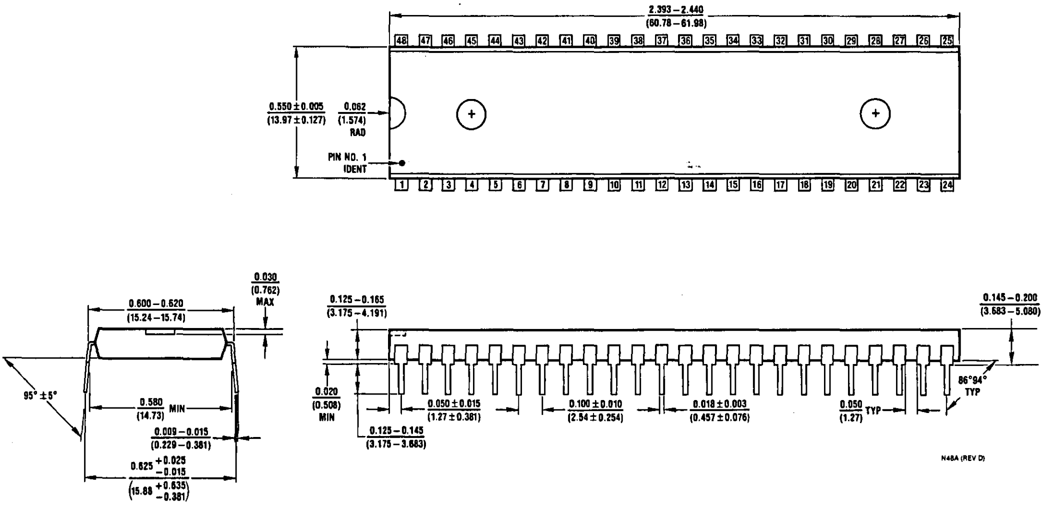
NS Package N24C



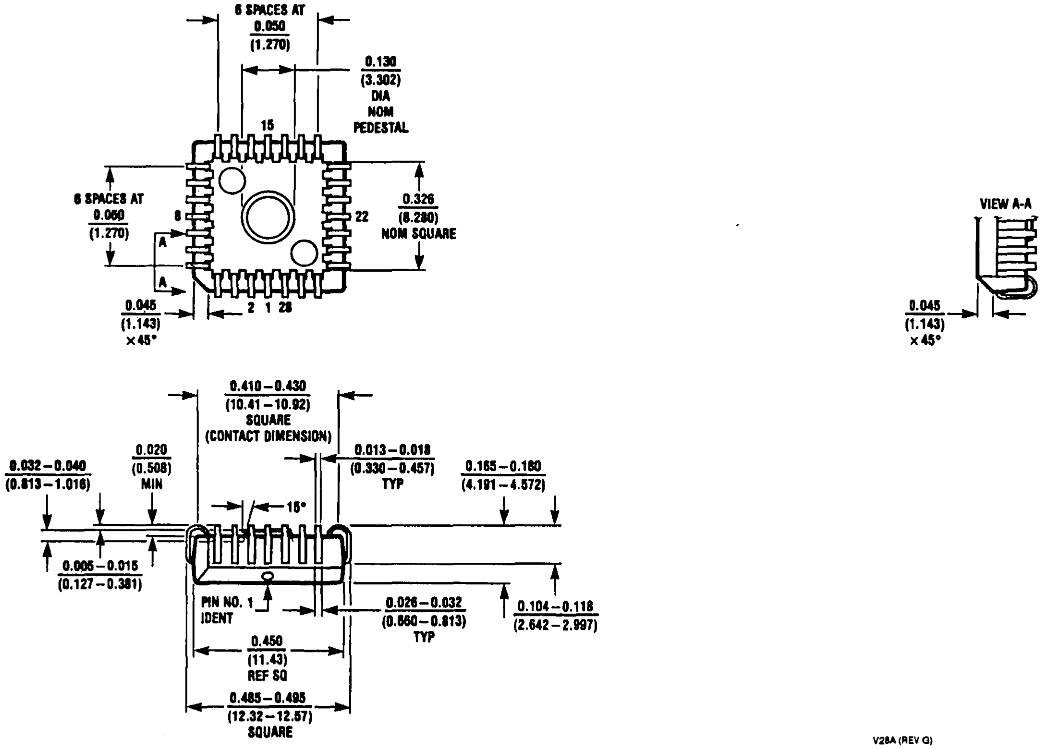
NS Package N28B



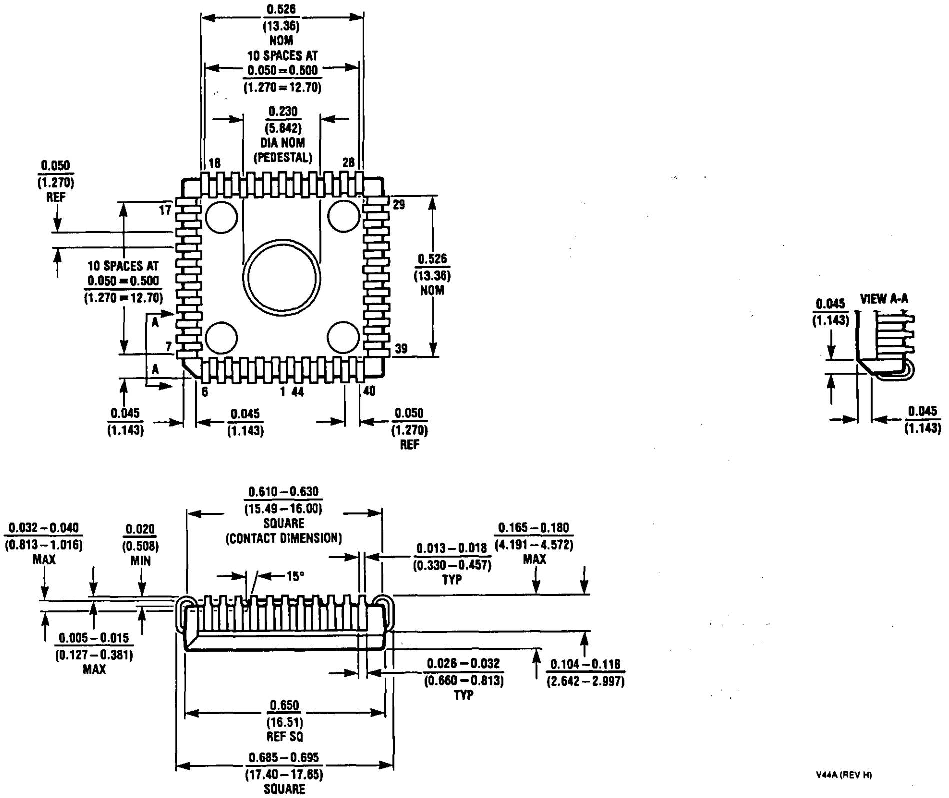
NS Package N40A



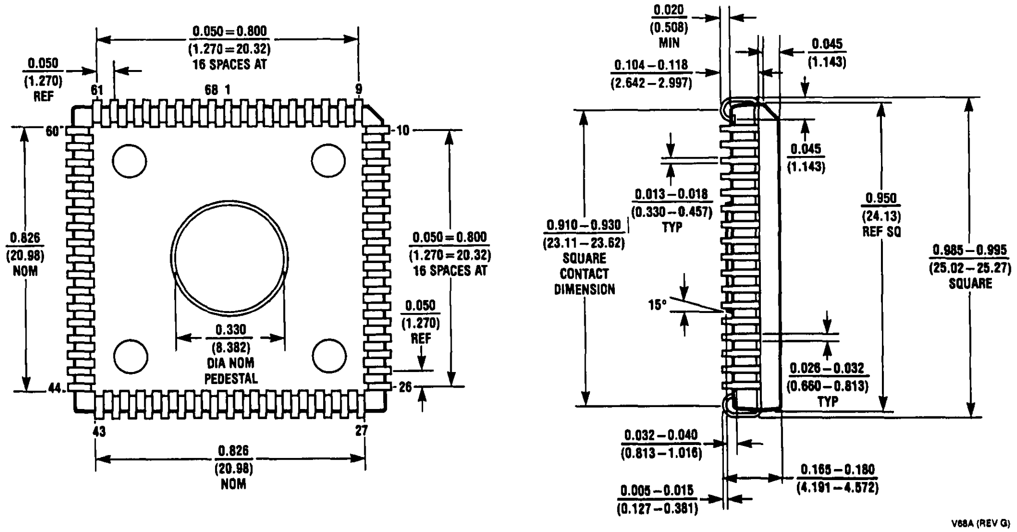
NS Package N48A



NS Package V28A



NS Package V44A



NS Package V68A

NOTES

NOTES

NOTES

NOTES

National
Semiconductor
Corporation

National Semiconductor Corporate Headquarters
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: (408) 721-5000
TWX: (910) 339-9240

SALES OFFICES AND REPRESENTATIVES (Continued)

INTERNATIONAL OFFICES

Electronica NSC de Mexico SA
Juventino Rosas No. 118-2
Col Guadalupe Inn
Mexico, 01020 D.F. Mexico
Tel: (905) 524-9402

National Semicondutores Do Brasil Ltda.
Av. Brig. Faria Lima, 830
8 Andar
01452 Sao Paulo, SP. Brasil
Tel: (55/11) 212-5066
Telex: 391-1131931 NSBR BR

National Semiconductor GmbH
Westendstrasse 193-195
D-8000 Munchen 21
West Germany
Tel: (089) 5 70 95 01
Telex: 522772

National Semiconductor (UK) Ltd.
301 Harpur Centre
Horne Lane
Bedford MK40 1TR
United Kingdom
Tel: 0234-47147
Telex: 826 209

National Semiconductor Benelux
Ave Charles Quint 545
B-1080 Bruxelles
Belgium
Tel: (02) 4661807
Telex: 61007

National Semiconductor (UK) Ltd.
1, Bianco Lunos Alle
DK-1868 Copenhagen V
Denmak
Tel: (01) 213211
Telex: 15179

National Semiconductor
Expansion 10000
28, Rue de la Redoute
F-92 260 Fontenay-aux-Roses
France
Tel: (01) 660-8140
Telex: 250956

National Semiconductor S.p.A.
Via Solferino 19
20121 Milano
Italy
Tel: (02) 345-2046/7/8/9
Telex: 332835

National Semiconductor AB
Box 2016
Stensatravagen 4/11 TR
S-12702 Skarholmen
Sweden
Tel: (08) 970190
Telex: 10731

National Semiconductor
Calle Nunez Morgado 9
(Esc. Dcha. 1-A)
E-Madrid 16
Spain
Tel: (01) 733-2954/733-2958
Telex: 46133

National Semiconductor Switzerland
Alte Winterthurerstrasse 53
Postfach 567
CH-8304 Wallisellen-Zurich
Tel: (01) 830-2727
Telex: 59000

National Semiconductor
Pasilanraitio 6C
SF-00240 Helsinki 24
Finland
Tel: (90) 14 03 44
Telex: 124854

NS Japan Ltd.
4-403 Ikebukuro, Toshima-ku
Tokyo 171, Japan
Tel: (03) 988-2131
Fax: 011-81-3-988-1700

National Semiconductor Hong Kong Ltd.
Southeast Asia Marketing
Austin Tower, 4th Floor
22-26A Austin Avenue
Tsimshatsui, Kowloon, H.K.
Tel: 3-7231290
Cable: NSSEAMKTG
Telex: 52996 NSSEA HX

National Semiconductor (Australia) PTY, Ltd.
21/3 High Street
Bayswater, Victoria 3153
Australia
Tel: (03) 729-6333
Telex: AA32096

National Semiconductor (PTE), Ltd.
51 Goldhill Plaza, No. 10-01
Newton Road
Singapore 1130
Tel: 2506884
Telex: RS 33877

National Semiconductor (Far East) Ltd.
Taiwan Branch
P.O. Box 68-332 Taipei
7th Floor, Nan Shan Life Bldg.
302 Min Chuan East Road,
Taipei, Taiwan R.O.C.
Tel: (02) 501-7227
Telex: 22837 NSTW
Cable: NSTW TAIPEI

National Semiconductor (Far East) Ltd.
Korea Office
Third Floor, Hankyung Bldg.
4-25 Hannam-Dong
Yongsan-Ku, Seoul 140, Korea
Tel: 797-8001/3
Telex: K24942 NSRK

GF/1825 RRD-RRD50M046/Printed in U.S.A.