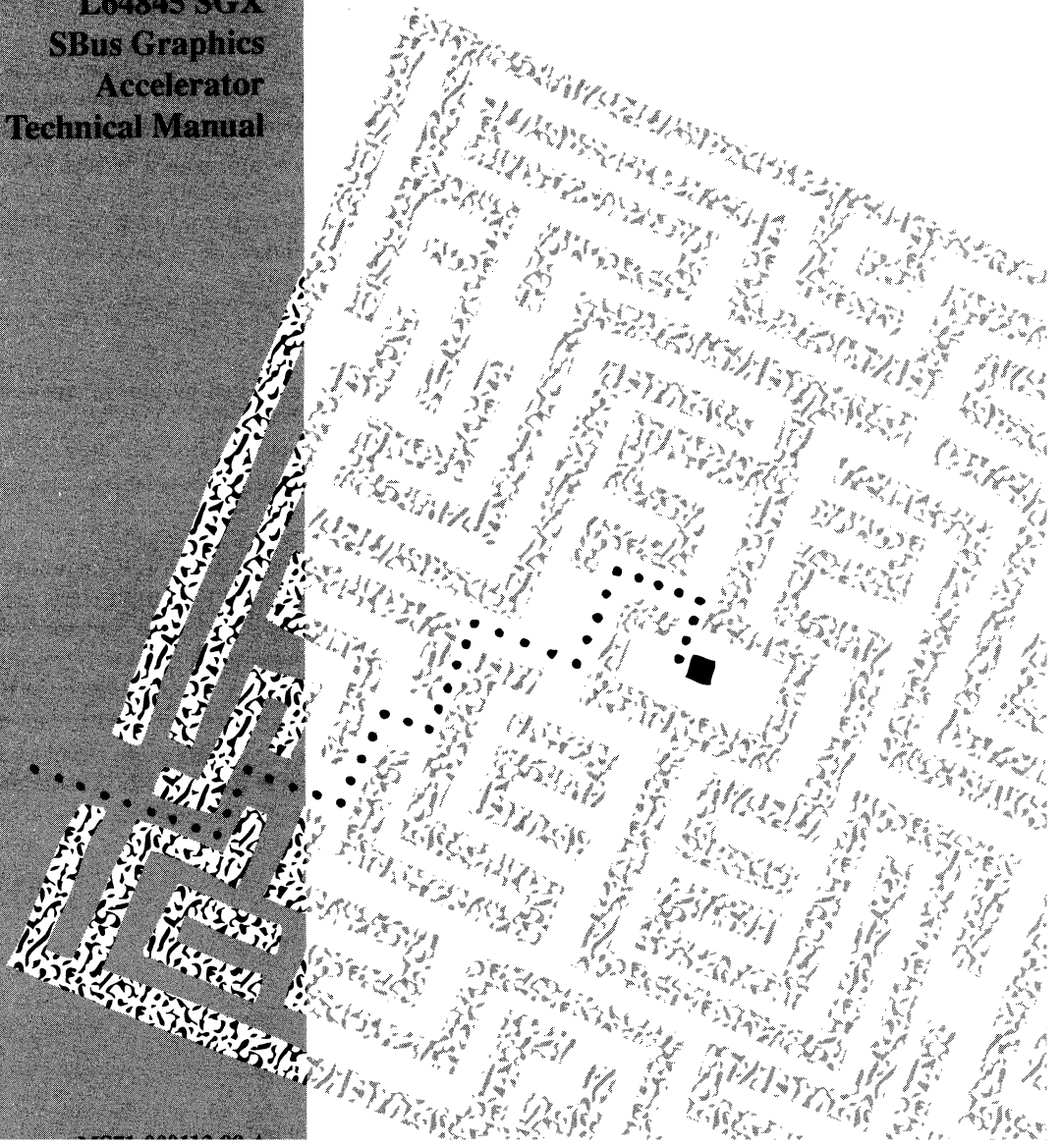
LSI LOGIC

L64845 SGX SBus Graphics Accelerator Technical Manual

LSI LOGIC

L64845 SGX SBus Graphics Accelerator Technical Manual



This document is preliminary. As such, it contains data derived from functional simulations and performance estimates. LSI Logic has not verified either the functional descriptions, or the electrical and mechanical specifications using production parts.

Publications are stocked at the address given below. Requests should be addressed to:

LSI Logic Corporation
Literature Distribution, M/S D-102
1551 McCarthy Boulevard
Milpitas, CA 95035
Fax (408) 433-8989

LSI Logic Corporation reserves the right to make changes to any products herein at any time without notice. LSI Logic does not assume any responsibility or liability arising out of the application or use of any product described herein, except as expressly agreed to in writing by LSI Logic; nor does the purchase or use of a product from LSI Logic convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual property rights of LSI Logic or third parties.

This document is derived from documents created by Sun Microsystems, Inc., and thus constitutes a derivative work.

TRADEMARK ACKNOWLEDGMENT

The LSI Logic logo design is a registered trademark of LSI Logic Corporation. Sun Microsystems is a registered trademark and Sun and Solaris are trademarks of Sun Microsystems, Inc. SPARC is a registered trademark and SPARCstation and SPARCsystems are trademarks of SPARC International, Inc. Products bearing the SPARC trademark are based on an architecture developed by Sun Microsystems, Inc. SparKIT is a trademark of SPARC International, Inc. and is exclusively licensed to LSI Logic Corporation. Brooktree and Bt458 RAMDAC are trademarks of Brooktree Corporation. X Window System is a product of the Massachusetts Institute of Technology.

Preface

This manual describes the L64845 SGX SBus Graphics Accelerator from LSI Logic. It describes the SGX chip and the SGX card that can be made from LSI Logic's SGX Manufacturing Kit. It also contains information that helps in the design of a custom SGX card.

This manual does not include complete register descriptions or software information about the SGX.

Audience

This book assumes that you are familiar with microprocessors and related support devices. The book is written for:

- Engineers and managers who are evaluating the SGX.
- Hardware engineers who are integrating the SGX into a system.

Organization

This book has the following chapters and appendices:

- Chapter 1, **Introduction**, provides an overview of the SGX.
- Chapter 2, **Functional Description**, describes the internal function of the SGX.
- Chapter 3, **Signal Descriptions**, describes the SGX I/O signals.
- Chapter 4, **SBus Interface**, describes the SGX SBus Interface.
- Chapter 5, **Memory Interfaces**, describes the SGX interfaces with memory.
- Chapter 6, **Video Interface**, describes the SGX interfaces with the video monitor.
- Chapter 7, **Alternate Interface**, describes the SGX alternate interface.
- Chapter 8, **Specifications**, provides AC specifications, DC specifications, and packaging information.
- Appendix A, **Schematics**, provides the schematics for the SGX card.
- Appendix B, **The SGX Card**, describes the SGX card configuration.
- Appendix C, **Bt458 RAMDAC**, describes the Bt458 RAMDAC.

Related Publications

Brooktree Product Databook, available from Brooktree Corporation, 9950 Barnes Canyon Road, San Diego, CA 92121-2790.

GX Installation Guide, available from Sun Microsystems, Inc.

SBus Specification, Revision B, available from Sun Microsystems, Inc., 2550 Garcia Avenue, Mountain View, CA 94043.

Single Chip GX Reference Card, available from Sun Microsystems, Inc.

XGL Graphics Library Technical White Paper, available from Sun Microsystems, Inc.

XGL Graphics Library Manuals, available from Sun Microsystems, Inc.

Conventions Used in this Manual

The following signal naming conventions are used throughout this manual:

- A level significant signal that is true or valid when the signal is low always has an overbar ($\overline{\quad}$) over its name.
- An edge significant signal that initiates actions on a high to low transition always has an overbar ($\overline{\quad}$) over its name.

Hexadecimal numbers are indicated by the prefix "0x" before the number—for example, 0x32CF. Binary numbers are followed by a subscripted 2.

Contents

Tables

Chapter 1
Introduction

This chapter provides an overview of the LSI Logic's L64845 SGX SBus Graphics Accelerator. It is divided into four sections:

- Section 1.1, **General Description**, briefly describes the SGX and the SGX card.
- Section 1.2, **SBus Family Description**, describes some of LSI Logic's other SBus products.
- Section 1.3, **Internal Organization**, briefly describes the internal organization of the SGX.
- Section 1.4, **Features Summary**, lists the important features of the SGX.

1.1 General Description

The SGX is an SBus-compatible, 8-bit color/grey-scale Graphics Accelerator which incorporates all the logic necessary for a complete and highly integrated graphics subsystem.

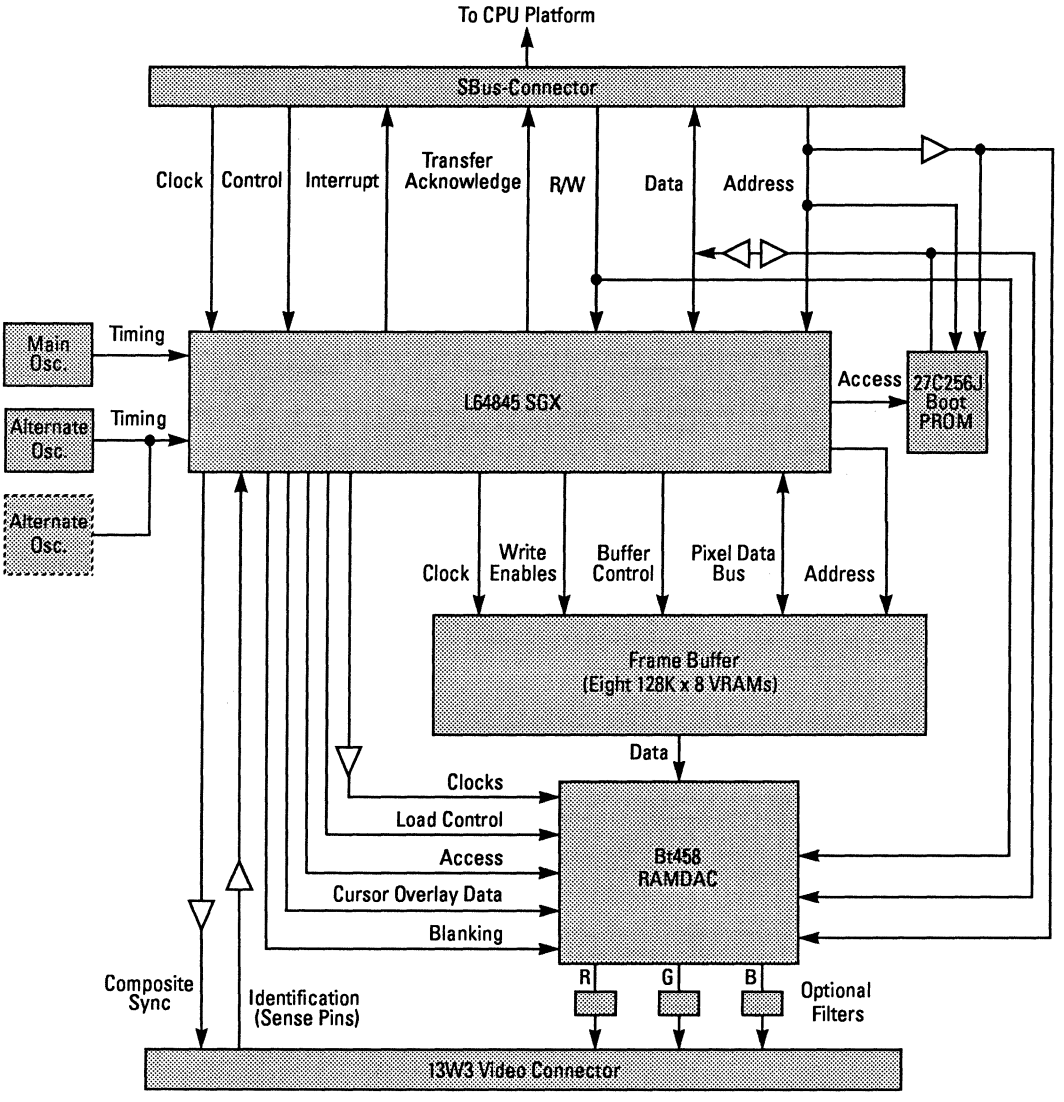
LSI Logic produces the SGX using its 1.0-micron drawn gate length (0.7-micron effective channel length) process. It is produced and marketed in a 223-pin ceramic pin grid array (CPGA) package.

A high performance graphics accelerator card can be created by adding eight 128K x 8 VRAMs, a RAMDAC, an external boot PROM and some connectors (with connecting logic).

LSI Logic provides a manufacturer's kit for an SGX single-wide SBus Graphics Accelerator card. The card connects directly to both the older 66-Hz and the newer 76-Hz Sun color monitor. The card is 100% software compatible with SPARCstation GX software and works with the SPARCstation 1, SPARCstation 1+, SPARCstation 2, SPARCstation IPC, and SPARCstation IPX workstations.

Figure 1.1 shows a block diagram of the SGX card. An identical block diagram with signal names can be found in Appendix A, "Schematics."

Figure 1.1
SGX Card
Block Diagram



1.2 SBus Family Description

In addition to the SGX, LSI Logic supplies several other components for the SBus architecture. These components provide video interfaces, DMA control, and an interface to the Mbus.

Video Interfaces

The L64825 SBus Video Frame Buffer and the L64855 SBus Graphics Controller each provide a complete interface between an SBus and a monochrome or color graphics subsystem. Both controllers support the Sun workstation monitor resolution of 1152 x 900, while the L64855 also supports higher resolution monitors and the less expensive Super VGA monitors.

DMA Controllers

The L64853 and L64853A SBus DMA Controllers each provide a complete SBus interface, with independent 16- and 8-bit DMA channels.

Both channels support DMA for use in applications that require operation as an SBus master. Such applications include Ethernet and SCSI bus controllers.

The two channels can also be used for applications that rely on programmed I/O and thus only use the L64853A's SBus slave capability. Such applications include serial ports and analog-to-digital converters.

Mbus-to-SBus Interface

The L64852 Mbus/SBus Interface and Controller (M2S) is a high-performance CMOS integrated circuit that provides all of the control, arbitration, and memory-management functions needed to interface the Mbus to SBus subsystems, such as the SGX card.

The M2S converts bus protocol while moving data between Mbus and SBus devices. When an Mbus device initiates I/O with an SBus device, the M2S chip responds as an Mbus slave and then becomes the SBus master for the data transfer. When an SBus device initiates a data access to the Mbus, the M2S chip responds as an SBus slave and then becomes the Mbus master.

1.3 Internal Organization

The SGX chip can be divided into three main functional blocks:

- Transformation Engine – Transforms 2D and 3D coordinates.
- Hardware Cursor – Generates the cursor.
- Frame Buffer Controller (FBC)
  - Executes the DRAW, BLIT, and FONT commands.
  - Performs object picking and clipping.
  - Controls foreground and background color, raster operations, plane masking, pixel masking, and the pattern used for operations.
  - Controls VRAM address multiplexing.

Chapter 2, "Functional Description," describes these blocks in detail.

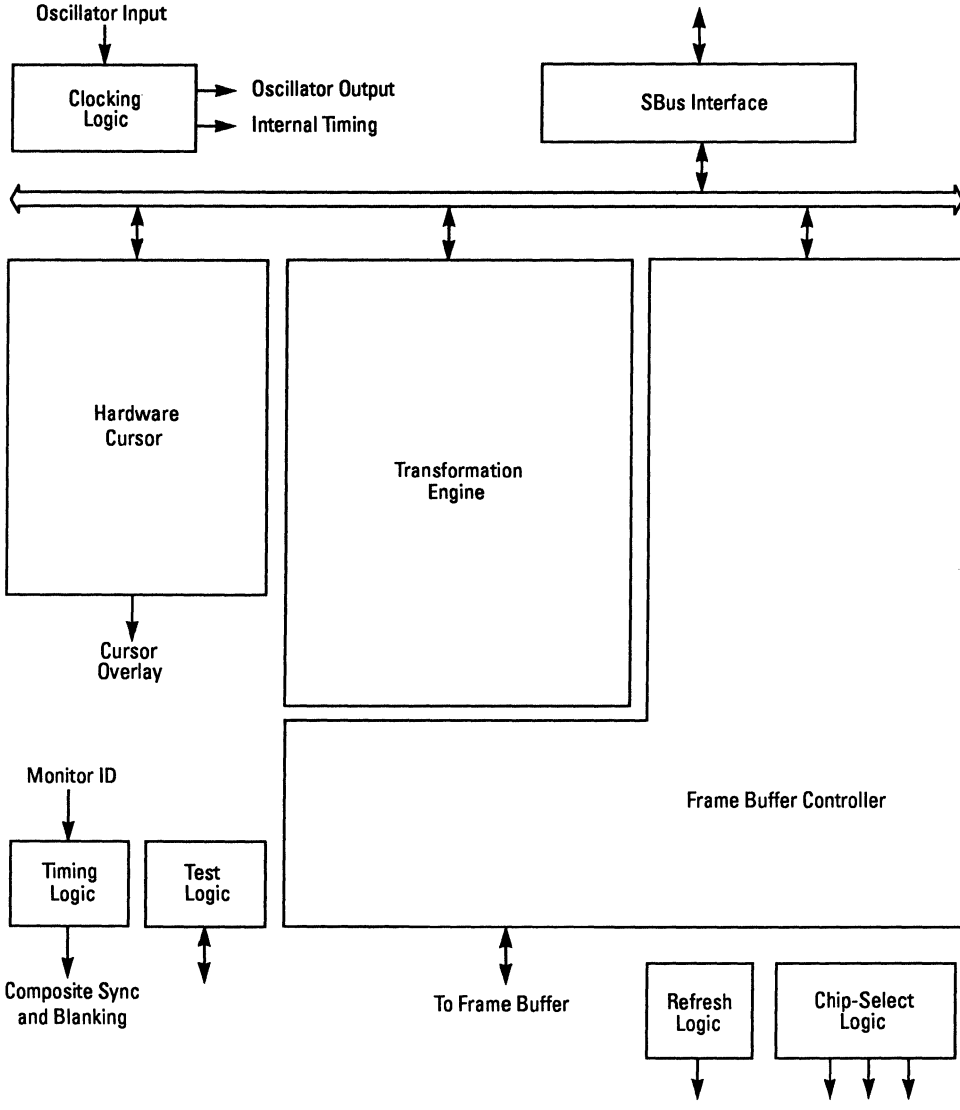
The SGX chip contains the following interface logic:

- SBus interface (to SBus)
- Memory interfaces
  - ROM interface (to PROM)
  - Frame buffer interface (to VRAM)
- Video interfaces
  - DAC interface (to RAMDAC)
  - Video timing interface (to video connector)
- ALT interface (unused, reserved for future expansion)

Chapters 4, 5, and 6 describe these interfaces.

Figure 1.2 shows the internal organization of the SGX.

Figure 1.2
SGX Chip Block Diagram



1.4 Features Summary

The SGX has the following features:

- Hardware cursor (32 x 32 pixels in three colors)
- Frame buffer (VRAM) and RAMDAC interfaces
- Flat shading in hardware
- Three graphics data modes:
  - Five-bit grey-scale (anti-aliased, high-resolution monochrome)
  - One-bit two-color
  - Eight-bit full-color graphics (256-color display, 16.8 million color palette)
- 2D/3D clipping (pick or render mode)
- Hardware 2D/3D transformations (translate, scale and rotate)
- Hardware DRAW, BLIT and FONT commands with optional raster operations, plane masking, and stenciling capabilities
- 46-MFLOPS 4 x 4 matrix transformation engine with a 64-register file and a 32-bit, signed multiply/accumulator (for matrix concatenations, dot products or convolutions)
- Hardware support for a generalized graphics pipeline with the following options:
  - Modeling and viewing transform
  - 2D/3D clip check
  - Perspective divide
  - Virtual device coordinates transform
- Manufactured using a 1.0-micron drawn gate length (0.7-micron effective channel length) process and supplied in a 223-pin Ceramic Pin Grid Array (CPGA)
- Includes all necessary drivers to support Solaris 1.X
- Supports Sun's Direct Graphics Access (DGA) software acceleration
- Manufacturing kit available for the one-slot SGX SBus card

Chapter 2
Functional Description

This chapter describes the internal function of the L64845 SGX SBus Graphics Accelerator. It is divided into four sections:

- Section 2.1, **Graphics Acceleration**, explains hardware graphics acceleration in the SGX.
- Section 2.2, **Software Support**, discusses software support for the SGX.
- Section 2.3, **Capabilities**, describes the functional capabilities of the SGX.
- Section 2.4, **Internal Logic**, describes the main functional blocks of the SGX.

2.1 Graphics Acceleration

The SGX improves on simple frame buffer architectures by using hardware graphics acceleration.

Frame buffers are areas of video memory (usually VRAM) into which the CPU directly writes pixel data. A display area is represented as a series of horizontal lines each made of adjacent pixels. When a program draws a horizontal line into memory, it simply fills a series of adjacent memory locations with data that represents the pixel color. Drawing the line requires two or more instructions for each word of pixel data: one instruction is required to write the data and at least one instruction is required to step to the next pixel. An arbitrary sloped line can take many hundreds of calculations to draw.

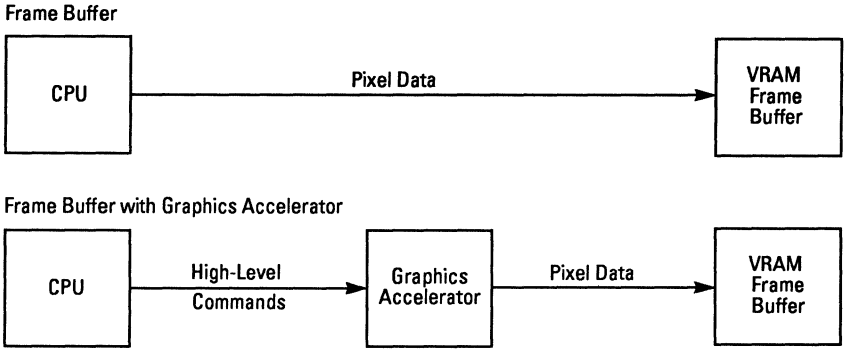
The SGX architecture places an intelligent processing capability between the CPU and the frame buffer. This processing capability enables the CPU to send high-level commands to the frame buffer instead of sending individual pixel values. So when the CPU tells the SGX to draw a line, it only needs to send the two end points and the line color (or intensity). The

SGX then calculates the individual pixel values for the operation and writes these pixels to video memory.

Graphics Acceleration improves system efficiency by off-loading time consuming tasks from the CPU. It also reduces the bus load, since the CPU sends less data to the SGX than it would to a simple frame buffer.

Figure 2.1 illustrates the difference between a simple frame buffer and one with a graphics accelerator.

Figure 2.1 Graphics Acceleration



Please note that a simple frame buffer configuration is sometimes called a dumb frame buffer (DFB). The L64825 SBus Video Frame Buffer is considered to be a DFB.

Graphics acceleration makes good use of RISC principles. According to these principles, it is best to perform a complex task by using low-level primitives. Graphics operations such as drawing, clipping, filling, block move operations, and geometric transformations can be structured so that they are performed by a small number of primitive operations in hardware. The SGX often performs graphics operations much faster than the CPU, but it is difficult to measure the performance speedup since the CPU can perform other tasks while the graphics coprocessor is rendering pixels to the screen. Also some graphics operations lend themselves to hardware acceleration more readily than others.

The SGX provides the greatest performance improvement when rendering clipped 3D images with perspective. This performance improvement is due to an integrated 4 x 4 matrix multiplier in the SGX. The algorithms used for 3D transformations, including those for rotation, translation, scaling, shearing, and perspective divide, all require 4x4 matrix operations. In these operations, three of the four matrix terms are used to represent the x,

y, and z axes; the fourth matrix term is used as a normalizing function that reduces the total number of calculations required for a series of complex transformations.

The following equation illustrates one possible matrix transformation. This particular example performs a 3D rotation about the Z axis by θ°.

$$R_Z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & -\cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ N \end{bmatrix}$$

Similar (but more complex) calculations are used to map and clip images into a viewport and to add perspective. The more transformations that are required to render an image, the greater the performance improvement with the SGX.

All of the functions that can be performed in 3D can also be performed in 2D. For rotation, scaling, shearing, and clipping of 2D images, only three of the four matrix terms are used: the first two for the x and *y* dimensions, and the third for the normalizing function. Perspective divide is not commonly used with 2D images.

2.2 Software Support

LSI Logic does not currently support software for the SGX, but the driver is a part of Solaris 1.0.

Software accesses the SGX functionality in two ways:

- System software accesses the SGX window display and movement functions.
- Software compiled with Sun Microsystem's XGL Graphics Library, SunPhigs or SunGKS uses the SGX acceleration features for rendering of 2D and 3D images with optional clip checking and perspective.

The SGX accelerates windowing performance by providing high-level hardware functions for window display, movement, and clip checking. To use these SGX system-level functions, the appropriate device driver must be linked into the kernel. This procedure is usually performed automatically during system start-up. See the installation instructions from Sun Microsystems for more information on device driver installation.

The XGL Graphics Library is a library of C functions that supports both immediate and display-list picking. It supports rendering of 2D and 3D clipped and depth-cued images using either an indexed or an RGB color model. The XGL functions support ambient, positional, spot, and directional light sources for both flat and Gouraud shading. XGL functions work with any Open Windows supported tool kit or window manager and are fully compliant with the X Window System protocol. The library includes functions for context and attribute setup, line rendering, curve rendering, curve/surface tessellation, surface rendering, marker rendering, stroke fonts, and object picking. See Sun Microsystem's *XGL Graphics Library Technical White Paper* for additional general information and the *XGL Graphics Library Manuals* for additional detailed information.

2.3 Capabilities

This section describes the capabilities of the SGX as implemented with the LSI Logic L64845 SGX Manufacturing Kit. It is divided into three subsections:

- Graphics Modes
- Commands
- Advanced Operations

These capabilities are all under software control.

Graphics Modes

The SGX supports three graphics data modes:

- High Resolution Monochrome Mode (HRMONO)
- Two Color Mode (COLOR1)
- Full Color Mode (COLOR8)

These modes are described below.

High Resolution Monochrome Mode (HRMONO)

This mode provides a single-plane, antialiased, high-resolution monochrome display (using five bits of data per physical pixel).

A programmer sees the resolution as four times the actual screen resolution in both the x and y directions. For a 1152 x 900 pixel screen, the software

sees a frame buffer size of 4608 x 3600 sub-pixels (16 sub-pixels for each physical pixel).

The SGX logically maps this large address space into the frame buffer address space. The resultant grey-scaled values are used to partly turn on selected pixels on the edge of an object. This technique, antialiasing, smooths the jagged or sawtooth forms that can appear on diagonal lines and high contrast edges.

Two Color Mode (COLOR1)

This mode provides one-plane color (one bit, two colors). All pixels are displayed on the screen in either the background color, the foreground color, or a blend of the two colors.

The image is expanded to eight planes of data with the correct raster operation and color. The addresses are pixel coordinates.

Full Color Mode (COLOR8)

This mode provides eight-plane color (eight bits of data mapped to one display pixel). A 256-entry 24-bit-wide color look-up table (CLUT) allows the display of 256 (2^8) colors out of a palette of 16, 777,216 (2^{24}) colors. The addresses are pixel coordinates. This mode is the only mode used for unaccelerated frame buffer accesses.

Commands

The SGX accelerates three basic graphics commands that are normally performed by the CPU:

- DRAW
- BLIT
- FONT

These functions are described below.

DRAW

The DRAW command draws points, lines, triangles, rectangles, and quadrilaterals using integer coordinates and a dual Bresenham algorithm to perform scan conversion into the frame buffer. A 16-by-16 monochrome repeating pattern can be used when rendering an object with the DRAW

command. The pattern can be aligned in both the x and y dimensions. This pattern can be overridden by specifying an option for a pattern of all ones or zeros.

A raster operation may be performed during the transfer. The raster operation determines a new destination pixel value by applying a logical operation on the corresponding source and destination pixels. A linear raster operation can be used for grey-scale antialiasing. The raster operation also expands the monochrome pattern data into the specified foreground and background colors.

BLIT

The BLIT (Block Image Transfer) command moves a rectangular block of pixels from one location to another equivalent-sized location within the frame buffer. The software specifies the source and destination areas for BLITs by loading the diagonal corners of source and destination regions into vertex registers. If the source and destination regions overlap, then the block is moved in such a way that the written destination data does not eradicate source data that has not yet been moved. The BLIT command always assumes COLOR8 mode and pixel coordinates.

A raster operation may be done during the transfer.

FONT

The FONT command transfers a raster image into the frame buffer using precomputed bit maps. Images described in a single-plane data structure can be written to the frame buffer in HRMONO and COLOR1 mode. Images described in an eight-plane data structure can be written to the frame buffer in COLOR8 mode. For COLOR1 mode, the image is expanded to eight planes of data with the correct raster operation and color. For HRMONO mode, the image is expanded to the proper grey scale values. The data is transferred 32 bits at a time providing up to 32 monochrome pixels, 32 monochrome sub-pixels, or four 8-bit color pixels.

The SGX determines the target location for the font by loading the horizontal left and right edges of the font as well as its scan line y-value. This mechanism allows the rendering of fonts with different pixel widths on the same font scan line. The same clipping and address processing logic used for drawing is then used to draw the font characters, with the font characters themselves treated as the source data for writing to the VRAM.

For the FONT command, the SGX supports x-y addressing, address auto-incrementing, clipping, font width specifications and frame buffer alignment.An optional raster operation may be done during the transfer.

Advanced Operations

The SGX includes a large number of operations which are not normally available in a single-chip graphics accelerator. These operations include:

- Clipping
- Raster operations
- Masking
- Transformations
- Hardware cursor support
- Video timing generation
- Graphics Transformation Pipeline

These operations are described below.

Clipping

Graphics operations need only be performed on objects and portions of objects that are within a viewing window (a rectangle for 2D, a viewing pyramid for 3D, or a double pyramid for homogeneous graphics). The process of selecting what objects are within a window is called clipping. The SGX allows two modes of clipping:

- Render mode
- Pick mode

In render mode the card clips an object to the interior of the clip window and then renders it into the frame buffer.

In pick mode the object is not rendered to the frame buffer. Instead, the pick status bit is set if any part of the object falls inside the clip window, indicating that a graphics operation will write data to the frame buffer.

Raster Operations

Raster operations allow boolean operations to be done in COLOR1 or COLOR8 mode and linear operations to be done in HRMONO mode.

There are 16 different boolean operations (linear operations are a subset of these 16) that may be done on the source and destination pixels. Foreground and background colors may be specified to provide mapping from monochrome data to color data.

Masking

Two types of masking are available:

- Plane masking
- Stenciling

Plane masking is used to enable or disable writes to each of eight individual planes.

Stenciling aligns a 32-bit pixel mask to the screen and enables or disables writes to individual pixels within all the planes simultaneously.

Transformations

The SGX supports up to 4 x 4 matrix arithmetic that can be used to translate an object from one location to another, to scale an object, and to rotate an object about an arbitrary point. The hardware also supports generalized vector operations. It can perform matrix concatenations, dot products, and convolutions. The SGX transforms virtual device coordinates into screen coordinates, so transformation results may be displayed directly. These transformations accept and produce signed integer, signed fixed point (16 bits), or single precision IEEE floating-point coordinate values.

Hardware Cursor Support

The SGX provides a programmable 32 x 32-pixel hardware cursor in three colors (using two bits).

Video Timing Generation

The SGX provides programmable vertical and horizontal timing signals for the monitor.

Graphics Transformation Pipeline

The SGX hardware also supports a generalized graphics transformation pipeline consisting of four optional functions:

- Modeling and viewing (MV) transform – transforms an image from world coordinates to viewing coordinates and performs transformations (including scaling, rotation, and translation)
- 2D/3D clip check – clips the image within a rectangle for 2D, a viewing pyramid for 3D, or a double pyramid for homogeneous graphics
- Perspective divide – projects 3D points onto a projection plane with perspective
- Virtual device coordinates (VDC) transform – transforms the image into final screen coordinates

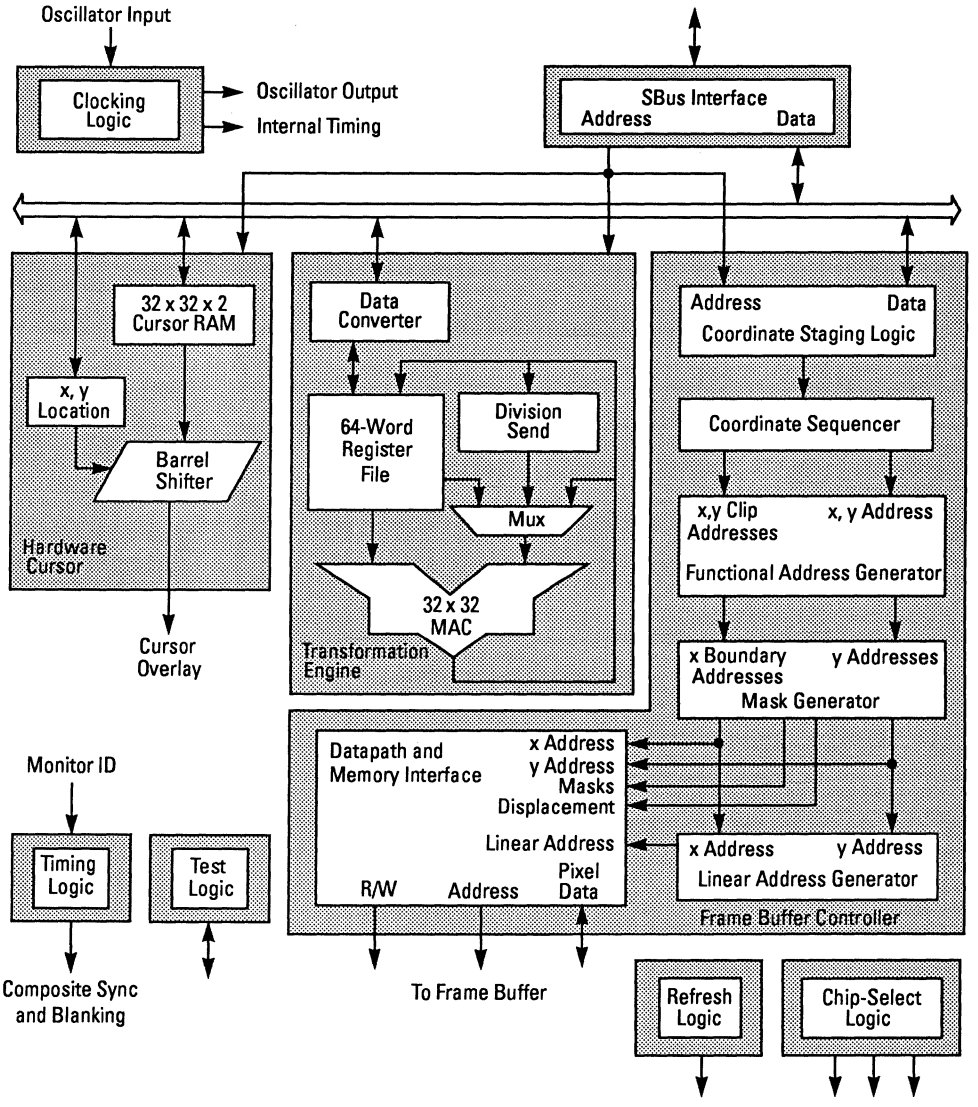
2.4 Internal Logic

The internal logic of the SGX chip is implemented in three main blocks:

- Transformation Engine
- Hardware Cursor
- Frame Buffer Controller (FBC)

Descriptions of these blocks follow Figure 2.2, which shows a block diagram of the internal logic of the SGX.

Figure 2.2
SGX Chip Detailed
Block Diagram



Transformation Engine

The Transformation Engine portion of the SGX transforms 2D and 3D coordinates. It contains three functional blocks of interest:

- Data Converters
- Register File
- Multiplier Accumulator (MAC)

Descriptions of these blocks follow.

Data Converter

The data converter converts data between the Transformation Engine's I/O data formats and its internal format. Data written directly to the Transformation Engine is always 32-bits wide.

The following data types are supported and are externally accessible:

- Single-precision IEEE floating-point: one sign bit, eight exponent bits, and 23 fraction bits
- Two's complement fixed point: signed 16 integer bits, and 16 fraction bits
- Two's complement integer: signed 32 integer bits

Register File

The Register File is a 64-entry, 32-bit-word multiport file used for storage of source, intermediate, and result data. The built-in routines in the Transformation Engine make use of this storage area for the calculation of 2D and 3D clipped transformations, including rotation, translation, scaling, shearing, and perspective divide transformations.

Multiplier Accumulator (MAC)

The Multiplier Accumulator (MAC) is a hard-wired, 32 x 32 signed matrix multiplier/accumulator. The MAC has a pipelined architecture and is capable of supporting 50 million floating-point operations per second.

The datapath logic supporting the hard-wired MAC (also called a Hard MAC) supports cyclic and divide operations. The multiplicand is taken directly from the register file. The multiplier is either reused, divided down

by the division send unit, or taken from the register file. The Transformation Engine's MAC uses an internal format that is hidden from the user.

Hardware Cursor

The logic for the hardware cursor contains the following main blocks:

- Cursor RAM
- Barrel Shifter

Descriptions of these blocks follow.

Cursor RAM

To store the cursor shape, two blocks of 32 x32-bit RAM are used. This configuration supports a cursor with four colors, of which one is transparent. The transparent color is used in a read-modify-write cycle to logically add back already resident data from where the cursor is to be drawn. This operation is repeated every time the cursor is moved.

One color is used for the cursor body, and the opposite color for the cursor outline. Use of such color inverting ensures that the cursor does not disappear when rendered on a solid object of the same color. The third color can be used for cursor detail.

Barrel Shifter

The Barrel Shifter reads the current x and y location of the cursor and rotates the data in the cursor RAM so that the cursor data aligns with pixel word boundaries.

Frame Buffer Controller (FBC)

The FBC portion of the SGX executes the DRAW, BLIT, and FONT commands and also performs object picking and clipping. It controls foreground and background color, raster operations, plane masking, pixel masking, and the pattern used for operations. It is also responsible for all VRAM address multiplexing.

The FBC supports the same data types as the Transformation Engine. Note, however, that the FBC processes arbitrary filled quadilaterals. The quadilaterals can be self-intersecting and degenerate. The FBC can therefore render points, lines, and triangles since they are all instances of degenerate quadilaterals.

The FBC contains seven functional blocks of interest:

- Coordinate Staging Logic
- Coordinate Sequencer
- Functional Address Generator
- Mask Generator
- Linear Address Generator
- Datapath and Memory Interface
- Timing and Refresh Logic

Coordinate Staging Logic

To define each quadilateral, up to four x, y coordinate pairs are loaded into the Coordinate Staging Logic. The FBC supports special addressing modes to allow the definition of degenerate quadilaterals without loading redundant vertices. The FBC also supports chained addressing so that common points in polylines do not need to be written to the FBC more than once.

The Coordinate Staging Logic extracts the x, y pixel coordinates from the pixel addresses sent to the FBC from the CPU, or passes x, y coordinates through from the Transformation Engine.

The Coordinate Staging Logic then creates setup status information based on the vertices of the polygon. (An example of setup status information is data that indicates whether the object is to be scanned from bottom to top, or top to bottom.)

Coordinate Sequencer

The Coordinate Sequencer determines which pairs of edges of the polygon are to be drawn. Combinational decoding determines whether the polygon is self-intersecting, hidden, or degenerate. Information on the x boundary and the y addresses for the two selected edges are then passed on for actual address generation

Functional Address Generator

The Functional Address Generator uses a modification of the standard Bresenham line-drawing function to calculate the edges of the polygon.

See any college-level text on computer graphics for a description of the Bresenham function.

The Functional Address Generator contains two Bresenham engines. Each engine calculates one x coordinate for each y coordinate that is passed from coordinate sequencing. The Functional Address Generator also contains clip boundary information. The Bresenham engines walk the polygon boundary as fast as the mask generation logic can accept data.

Degenerate quadilaterals, such as points and lines, simply cause the two Bresenham engines to process the same edge. The two engines run in parallel and so there is no loss in performance.

Mask Generator

The Mask Generator calculates the intersection of the clip window and the edges of the polygon to determine which pixels in the frame buffer need to be changed.

Clipping is a computation-intensive process. The FBC uses a standard two-pass clipping algorithm. The first pass eliminates all polygons that fall outside a clip test window (a window that is larger than the clip window). Software can set the size of the clip test window. The second pass performs clipping on the polygons that intersect the clip window.

Linear Address Generator

The Linear Address Generator calculates a series of horizontal pixel addresses using the edge coordinates that were passed from the mask generation logic. Prior to this stage, the FBC has only processed the edges of the polygon, since the edges have provided all the necessary boundary information. To generate intermediate pixel values, the Linear Address Generator first creates a scan line and then fills the scan line with data from pattern RAM.

Datapath and Memory Interface

The Datapath and Memory Interface generates the final pixel to be written to the VRAM frame buffer (or the CPU) by merging the bit mask information with the linear address. If rendering is not enabled (as in pick mode) the data is not written back to the frame buffer, but it may be read by the CPU via the SBus interface.

Chapter 3
Signal Definitions

This chapter defines the interface signals of the L64845 SGX SBus Graphics Accelerator. It is divided into nine sections:

- Section 3.1, **SGX Logic Symbol**, shows the SGX's logic symbol.
- Section 3.2, **SBus Interface Signals**, describes the SGX signals that interface with the SBus.
- Section 3.3, **ROM Interface Signal**, describes the SGX signal that interfaces with the ROM (PROM).
- Section 3.4, **Frame Buffer Interface Signals**, describes the SGX signals that interface with the frame buffer (VRAM).
- Section 3.5, **Video Connector Interface Signals**, describes the SGX signals that interface directly with the video connector.
- Section 3.6, **DAC Interface Signals**, describes the SGX signals that interface with the DAC (RAMDAC).
- Section 3.7, **ALT Interface Signal**, describes the SGX signal that has been set aside to interface with an optional external device.
- Section 3.8, **Clock Input Signals**, describes the SGX clock input signals.
- Section 3.9, **External Transfer Signal**, describes the external transfer signal.

Appendix C, "Bt458 RAMDAC," describes all of the Bt458 RAMDAC signals.

Chapters 4, 5 and 6 show the functional waveforms for the interface signals.

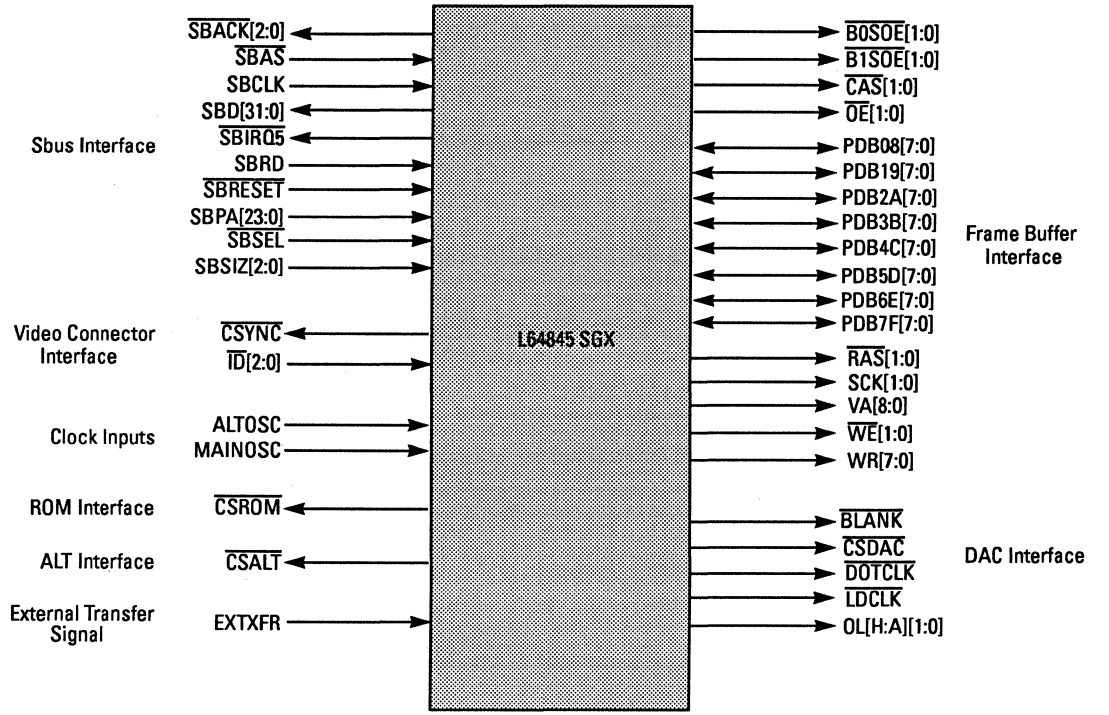
For information on the electrical characteristics of these signals, see Chapter 8, "Specifications."

The signal input and output direction is relative to the SGX chip.

3.1 SGX Logic Symbol

Figure 3.1 shows the logic symbol for the SGX. In the figure, a bar over the signal name indicates the signal is active LOW. Arrows indicate whether the signal is an input, an output, or bidirectional.

Figure 3.1
SGX Logic Symbol



3.2 SBus Interface Signals

This section describes each signal that interfaces the SBus with the SGX chip. SBus signal names are prefixed with 'SB' and are listed in alphabetical order.

For more information on the SBus see Chapter 4, "SBus Interface," or refer to *The SBus Specification.*

$\overline{\text{SBACK}}$[2:0] SBus Transfer Acknowledge **Output**

These output signals indicate the conclusion of an SGX SBus transaction to the SBus. These signals are usually 3-state and are only actively driven during the two last cycles of an SGX SBus access. The following table shows the encoding of $\overline{\text{SBACK}}$[2:0].

Bit 2	*Bit 1*	*Bit 0*	*Encoding*
1	1	1	Idle/Wait
1	1	0	Error acknowledgment
1	0	1	Byte data acknowledgment
1	0	0	Rerun acknowledgment
0	1	1	Word data acknowledgment
0	1	0	Reserved
0	0	1	Half-word data acknowledgment
0	0	0	Reserved

$\overline{\text{SBAS}}$ SBus Address Strobe **Input**

This input signal indicates that the SBus address, data, and control signals are now valid on the SBus.

SBCLK SBus Clock **Input**

This input signal provides the basic timing for the SGX SBus interface logic. SBCLK should run in the 16–25 MHz range and should be symmetrical.

SBD[31:0] SBus Data Bus **Bidirectional**

These signals transfer data between the SGX and the SBus. They may also contain virtual addresses from a SBus master before the addresses go through the MMU translation.

$\overline{\text{SBIRQ5}}$ SBus Interrupt for Vertical Blanking **Output**

This signal is used by the SGX to request an SBus interrupt during vertical blanking.

SBPA[23:0] SBus Physical Address Bus **Input**

These signals contain the physical address of the data. The SGX uses 24 of the 32 SBus address lines. The physical address is driven to a valid state no later than the next clock cycle after $\overline{\text{SBAS}}$ is asserted, and remains stable until the next clock cycle after $\overline{\text{SBAS}}$ is deasserted.

SBRD SBus Read/Write **Input**

This signal tells the SGX whether the current SBus request is a read (HIGH) or a write (LOW).

$\overline{\text{SBRESET}}$ **SBus Reset** **Input**
This signal usually causes the SGX to reset to a known state. It is used in test mode.

$\overline{\text{SBSEL}}$ **SBus Slave Select** **Input**
This signal indicates that the SBus master is selecting the SGX for access.

SBSIZ[2:0] **SBus Data Size** **Input**
These signals tell the SGX the data size of the current access. These lines are only used when $\overline{\text{SBSEL}}$ and $\overline{\text{SBAS}}$ are active. The SGX returns an error if the SBSIZ lines indicate that a burst cycle has been requested.

3.3 ROM Interface Signal

This section describes the signal that interfaces the onboard PROM with the SGX chip.

$\overline{\text{CSROM}}$ **ROM Chip Select** **Output**
This output signal drives the $\overline{\text{CE}}$ and $\overline{\text{OE}}$ PROM inputs. It is asserted LOW when the SGX detects an SBus access intended for the PROM.

3.4 Frame Buffer Interface Signals

This section describes each signal that interfaces the frame buffer (VRAMs) with the SGX chip. The signal names are listed in alphabetical order.

$\overline{\text{B0SOE}}$[1:0] **VRAM Serial Output Enable for Buffer 0** **Output**
These output signals drive the VRAM $\overline{\text{SOE}}$ inputs. B0SOE0 drives a bank processing the first four pixels, B0SOE1 drives a bank processing the second four pixels. In the SGX card, $\overline{\text{B0SOE}}$[1:0] alternate for each shift clock so the VRAM can multiplex eight pixels into the four-pixel DAC input. These signals are synchronous within the oscillator input selected for video timing generation (MAINOSC or ALTOSC).

$\overline{\text{B1SOE}}$[1:0] **VRAM Serial Output Enable for Buffer 1** **Output**
$\overline{\text{B1SOE}}$[1:0] is intended for double-buffered boards and is not supported in the current SGX card implementation.

$\overline{\text{CAS}}$[1:0] **Column Address Strobe for Buffer 1 / Buffer 0** **Output**
These output signals drive the VRAM $\overline{\text{CAS}}$ inputs. These signals indicate that the current address is a DRAM column address.

CAS1 is intended for double-buffered boards and is not supported in the current SGX card implementation.

$\overline{OE}$[1:0] **Output Enable for Buffer 1 / Buffer 0** **Output**
These output signals drive the VRAM $\overline{OE}$ inputs. These signals enable the VRAM output during reads.

OE1 is intended for double-buffered boards and is not supported in the current SGX card implementation.

PDBxx[7:0] **Pixel Data Bus** **Bidirectional**
These signals are the eight-pixel bidirectional data lines between the SGX and the VRAM.The following table shows how the data lines should be connected. In the table, Pixel 0 is the left-most pixel and pixel 7 is the right-most pixel.

VRAM Bank	*Data*	*Buffer 0*	*Buffer 1*[1]
0	PDB08	Pixel 0	Pixel 0
0	PDB19	Pixel 1	Pixel 1
0	PDB2A	Pixel 2	Pixel 2
0	PDB3B	Pixel 3	Pixel 3
1	PDB4C	Pixel 4	Pixel 4
1	PDB5D	Pixel 5	Pixel 5
1	PDB6E	Pixel 6	Pixel 6
1	PDB7F	Pixel 7	Pixel 7

1. Signals reserved for use in double-buffered configuration.

$\overline{RAS}$[1:0] **Row Address Strobe for Buffer 1 / Buffer 0** **Output**
These output signals drive the VRAM $\overline{RAS}$ inputs. These signals indicate that the current address is a DRAM row address.

RAS1 is intended for double-buffered boards and is reserved for future use.

SCK[1:0] **Serial Shift Clock for Buffer 1 / Buffer 0** **Output**
These signals drive the VRAM SC inputs. This output is synchronous with the oscillator input selected for video timing generation (MAINOSC or ALTOSC).

SCK1 is intended for double-buffered boards and is reserved for future use.

VA[8:0] **VRAM Address** **Output**
These signals drive the VRAM A[8:0] inputs. The linear address bits that appear on the VA[8:0] bus are a function of the memory

control operation and the type of VRAM. The following table describes the usage of the VA[8:0] lines.

Memory Control Operation	*VA[8:0] for 128K x 8 VRAM*[1, 2]
Normal $\overline{RAS}$	LA[19:11]
Normal $\overline{CAS}$	LA[11:4], B
Transfer $\overline{RAS}$	TA[8:0]
Transfer $\overline{CAS}$	0s

1. LA[xx] refers to linear address bits.
2. TA[xx] refers to the transfer address counter.

Note that the SGX always resets the transfer address counter to zero during vertical blanking.

$\overline{WE}$[1:0] **Write Enable for Buffer 1 / Buffer 0** **Output**
WE[1:0] is intended for double-buffered boards and is not supported in the current SGX card implementation.

In a double-buffered configuration these signals are logically ANDed with the WR[7:0] outputs and the result used to drive the corresponding Buffer 0 / Buffer 1 VRAM $\overline{WE}$ inputs.

WR[7:0] **Write Enable** **Output**
These output signals drive the VRAM $\overline{WE}$ inputs. These signals are used to enable individual pixels during VRAM writes. WR[7:0] are driven LOW when the row address for loading the plane mask into the VRAMs is on the address bus.

3.5 Video Connector Interface Signals

This section describes each signal that directly interfaces the video connector with the SGX chip. The signal names are listed in alphabetical order.

$\overline{CSYNC}$ **Composite Sync** **Output**
This output signal drives the combined vertical and horizontal sync input signal for the monitor. This output is synchronous with the oscillator input selected for video timing generation (MAINOSC or ALTOSC).

$\overline{ID}$[2:0] **Identification** **Input**
These input signals contain the identification code for the monitor. The SGX uses $\overline{ID}$[2:0] to select the oscillator input.

See Chapter 6, "Video Interface," for more information.

3.6 DAC Interface Signals

This section describes each signal that interfaces the RAMDAC with the SGX chip. The signal names are listed in alphabetical order.

Appendix C, "Bt458 RAMDAC," describes all of the Bt458 RAMDAC signals.

$\overline{\text{BLANK}}$ **Blanking** **Output**
This output signal drives the RAMDAC input blanking signal ($\overline{\text{BLANK}}$). This output is synchronous with the oscillator input selected for video timing generation (MAINOSC or ALTOSC).

$\overline{\text{CSDAC}}$ **DAC Chip Select** **Output**
This output signal drives the RAMDAC chip select input ($\overline{\text{CS}}$). It is asserted LOW when the SGX detects an SBus access intended for the DAC.

$\overline{\text{DOTCLK}}$ **Video Dot Clock** **Output**
This output signal is used to generate the RAMDAC differential clock inputs (CLKP and $\overline{\text{CLKN}}$). This output reflects either the MAINOSC or ALTOSC oscillator input.

$\overline{\text{LDCLK}}$ **Load Clock** **Output**
This output signal connects to the RAMDAC load control input ($\overline{\text{LD}}$). This output is one-fourth the $\overline{\text{DOTCLK}}$ output.

OL[H:A][1:0] **Overlay Plane 1 / Plane 0** **Output**
These output signals drive the RAMDAC overlay select inputs (OL[D:A][1:0]).

The current SGX card implementation uses only OL[D:A][1:0] for bit plane 1 and 0 of the hardware cursor. Bit A is the left-most pixel. OL[H:E][1:0] are unused. These output signals are synchronous with the oscillator input selected for video timing generation (MAINOSC or ALTOSC).

3.7 ALT Interface Signal

This section describes the signal that interfaces an alternate external device with the SGX chip. This signal is unused in the present SGX configuration.

$\overline{\text{CSALT}}$ **Alternate Chip Select** **Output**
This output drives the alternate device enable input. It is asserted LOW when the SGX detects an SBus access intended for the device (accessed through the alternate device address space).

3.8 Clock Input Signals

This section describes the clock input signals to the SGX chip. The signal names are listed in alphabetical order.

ALTOSC **Alternative Oscillator Input** **Input**

If selected, this input signal provides basic video timing control (by being used to generate $\overline{\text{DOTCLK}}$, $\overline{\text{LDCLK}}$, and related synchronous signals). The maximum frequency of ALTOSC is 117 MHz.

MAINOSC **Main Oscillator Input** **Input**

This oscillator input (or binary divisions of it) clocks the core logic of the SGX and its $\overline{\text{RAS}}/\overline{\text{CAS}}/\overline{\text{OE}}$ memory timing. MAINOSC can also be selected to provide basic video timing control (by being used to generate $\overline{\text{DOTCLK}}$, $\overline{\text{LDCLK}}$, and related synchronous signals). The intended frequency of MAINOSC is 92.9405 MHz.

3.9 External Transfer Signal

This section describes the external transfer signal.

EXTXFR **External Transfer** **Input**

When enabled, this signal indicates to the SGX that an SBus master is requesting a transfer. Instead of using the transfer signal generated by the SGX timing generation logic, the leading (rising) edge of EXTXFR is synchronized and fed to the memory control arbitration logic. External transfers are given the highest priority and are thus guaranteed service as soon as memory accesses already in progress are concluded. EXTXFR needs to be asserted for a minimum time greater than or equal to 13 clock periods (551 ns if the FBC is running at 23.24 MHz). The trailing (falling) edge of EXTXFR will cause the deassertion of $\overline{\text{OE}}$[1:0] thus completing a transfer cycle.

Chapter 4
SBus Interface

This chapter describes how the L64845 SGX SBus Graphics Accelerator interfaces with the SBus. It is divided into five sections:

- Section 4.1, **Features**, describes the main features of the SGX SBus interface.
- Section 4.2, **SBus Connector**, describes the SBus connector and shows its pin list.
- Section 4.3, **SBus Cycle**, describes the SBus cycle.
- Section 4.4, **Register Reads and Writes**, describes reads and writes to the SGX internal registers.
- Section 4.5, **Internal Register Access**, describes the Sbus interface software.

The SBus is a high-performance system bus that allows the addition of add-in cards to highly integrated workstations. The SBus is an onboard bus, not a backplane bus.

The SGX SBus interface follows the specifications described in *The SBus Specification*. Please refer to this publication for a more complete description of the SBus.

Chapter 3, "Signal Definitions," defines the SBus interface signals.

4.1 Features

The SGX is an SBus slave device.

The SGX SBus interface:

- Provides SBus handshaking for SGX internal and external SBus transfers
- Synchronizes operation over the full range of SBus clock rates (16.67 to 25 MHz)

- Provides a 32-bit datapath
- Provides a 24-bit physical address space
- Provides one interrupt level
- Supports 1, 2, and 4 byte data transfers (burst-mode transfers are not supported)
- Configures the card automatically using machine independent code
- Provides geographical addressing, via a single slave select signal, $\overline{\text{SBSEL}}$, which eliminates the need for slave address jumpers
- Decodes SBus Physical Address signals, SBPA[23:0], into appropriate chip select signals
- Supports Direct Virtual Memory Access (DVMA) by all bus masters, greatly simplify operating system and software memory management
- Samples all signals on the rising edge of the clock
- Synchronizes the SBus clock and the local board clock
- Buffers SBus address, data, and control signals for fast write capability
- Buffers address, data, and control signals for Transformation Engine and Hardware Cursor (TEC) to Frame Buffer Controller (FBC) transfers (autoload cycles)

4.2 SBus Connector

The SGX card connects to the SBus via a 96-pin male mini-DIN connector. This connector, marked 'J1' on the SGX card, carries both signals and power.

Chapter 3, "Signal Definitions," defines the SBus interface signals.

Table 4.1 shows the pin list for the 96-pin SBus connector.

Table 4.1 SBus Connector Pin List

Pin #	*Description*	*Pin #*	*Description*	*Pin #*	*Description*
01	VSS	33	$\overline{\text{SBAS}}$	65	SBD18
02	N/C	34	SBPA8	66	SBD20
03	$\overline{\text{SBSEL}}$	35	SBPA10	67	SBD22
04	N/C	36	$\overline{\text{SBACK0}}$	68	VSS
05	SBD0	37	SBPA12	69	SBD24
06	SBD2	38	SBPA14	70	SBD26
07	SBD4	39	SBPA16	71	SBD28
08	N/C	40	$\overline{\text{SBACK1}}$	72	VSS
09	SBD6	41	SBPA18	73	SBD30
10	SBD8	42	SBPA20	74	SBSIZ1
11	SBD10	43	SBPA22	75	SBRD
12	N/C	44	$\overline{\text{SBACK2}}$	76	VSS
13	SBD12	45	N/C	77	SBPA1
14	SBD14	46	N/C	78	SBPA3
15	SBD16	47	N/C	79	SBPA5
16	N/C	48	N/C	80	+5V
17	SBD19	49	SBCLK	81	SBPA7
18	SBD21	50	N/C	82	SBPA9
19	SBD23	51	$\overline{\text{SBAS}}$	83	SBPA11
20	$\overline{\text{SBIRQ5}}$	52	VSS	84	VSS
21	SBD25	53	SBD1	85	SBPA13
22	SBD27	54	SBD3	86	SBPA15
23	SBD29	55	SBD5	87	SBPA17
24	N/C	56	VDD	88	VDD
25	SBD31	57	SBD7	89	SBPA19
26	SBSIZ0	58	SBD9	90	SBPA21
27	SBSIZ2	59	SBD11	91	SBPA23
28	N/C	60	VSS	92	VSS
29	SBPA0	61	SBD13	93	N/C
30	SBPA2	62	SBD15	94	N/C
31	SBPA4	63	SBD17	95	$\overline{\text{SBRESET}}$
32	N/C	64	VDD	96	N/C

4.3 SBus Cycle

The SBus clock signal, SBCLK, provides the basic timing for SBus data transfers.

The SBus protocol employs three principles:

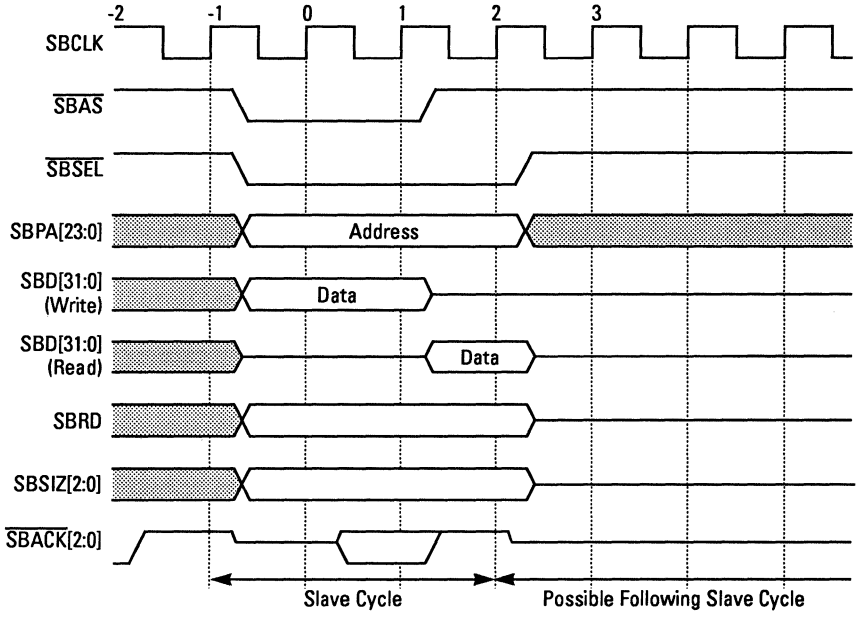
- **Synchronous Operation** – The SBus controller generates a fixed-frequency (set between 16.67 and 25 MHz, based on the CPU clock) clock with a clock skew of 2.5 ns or less. All signals are sampled on the rising edge of this clock (SBCLK). These signals all have a setup time of 15 ns and a hold time of 2.5 ns. In the SGX, there is one

asynchronous signal, $\overline{\text{SBIRQ5}}$, used for interrupts. The SBus controller synchronizes $\overline{\text{SBIRQ5}}$ to the appropriate clock.

- **Active Drive** – Before the source stops driving an asserted 3-state control signal, it drives the signal to the unasserted state. This procedure enables the bus to operate at speeds up to 25 MHz without needing low-resistance pullup resistors and output drives to sink the resulting static current.
- **No Driver Overlap** – Output drivers overlapping can result in unreliable operation and excessive power dissipation. Except for the open-drain interrupt signal, $\overline{\text{SBIRQ5}}$, no SBus signal is driven by two outputs during the same clock cycle.

Figure 4.1 shows a basic single-word SBus cycle (the relationship of signals used during a read/write) on the SGX card.

Figure 4.1
Basic SBus Cycle



The SBus controller places an address on the physical address lines (SBPA[23:0]) and asserts both the address strobe signal ($\overline{\text{SBAS}}$) and the slave select signal ($\overline{\text{SBSEL}}$).

For a write, the controller drives read/write (SBRD) LOW, asserts the transfer size (SBSIZ[2:0]), and drives the data lines (SBD[31:0]). The SGX then has up to 255 clock cycles to accept the transfer and issue the

proper acknowledgment by driving $\overline{\text{SBACK}}$[2:0] LOW, then back to the idle state (unasserted) for one clock cycle.

For a read, the controller drives read/write (SBRD) HIGH and asserts the transfer size (SBSIZ[2:0]). The SGX generates the acknowledgment and then drives the data lines on the following clock cycle for exactly one cloak cycle. SBus reads from the internal sections of the SGX take between five and seven SBus clock cycles when the chip is idle.

When a read or write is complete, the SBus controller deasserts SBPA[23:0], SBRD, $\overline{\text{SBSEL}}$, and SBSIZ[2:0]. The SGX deasserts $\overline{\text{SBACK}}$[2:0].

See *The SBus Specification* for information on time-out, rerun, and other bus error cycles.

The SBus protocol includes specifications for burst-mode transactions. The SGX does not support burst-mode transactions and issues an error acknowledgment if they are attempted.

$\overline{\text{SBRESET}}$, used only in test mode, resets the SGX to a known state.

The SGX returns $\overline{\text{SBIRQ5}}$ to the SBus in response to a vertical sync interrupt.

4.4 Register Reads and Writes

The SBus sees the SGX internal registers as 32-bit ports. Data transfers must synchronize with the local clock. Writes take three SBus clock cycles to acknowledge. Reads take five to seven SBus clock cycles to acknowledge. Both reads and writes generally conform to the SBus slave cycle.

Figure 4.2 shows the best- and worst-case register read cycle. The SGX attempts to return data as soon as possible. Figure 4.3 shows the register write cycle

Figure 4.2 Register Read Cycle

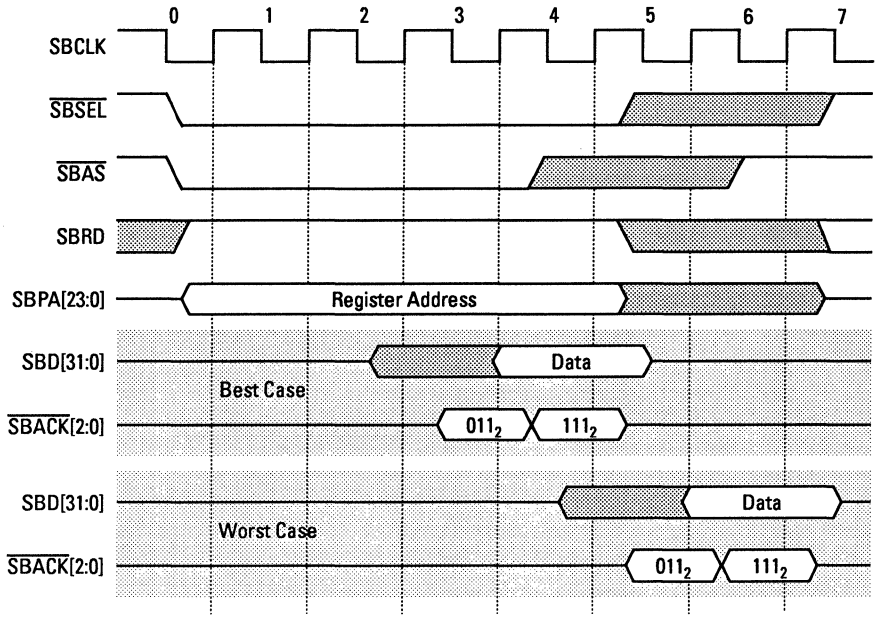
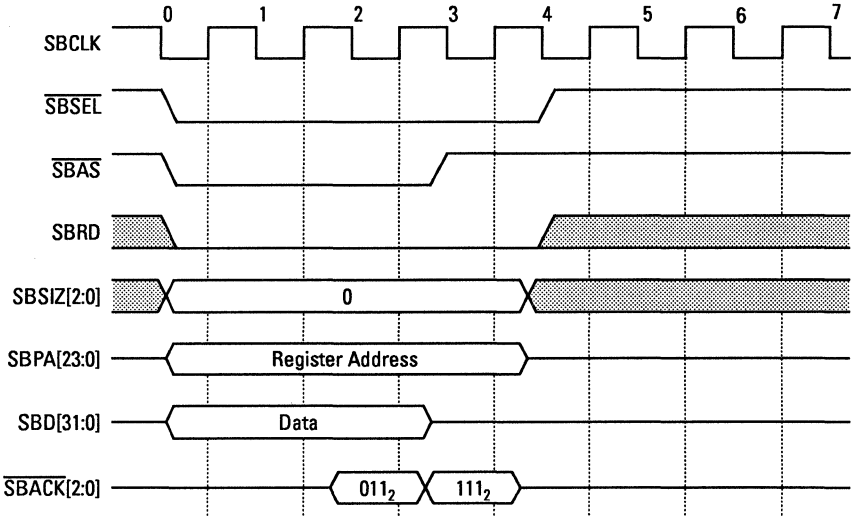


Figure 4.3 Register Write Cycle



4.5 Internal Register Access

The software driver controls the SGX card by reading and writing to registers. These registers are accessed through seven data structures that control the corresponding parts of the SGX card. The software driver for the SGX card must contain the following elements:

- Code for reading the ID code and EPROM at boot time.
- Code for reading the FORTH code stored in the EPROM.
- Code for loading the Palette DAC (RAMDAC).
- Code for initializing the TEC and FBC.
- Code for accessing the Dumb Frame Buffer (DFB) - minimum function for a console.

Table 4.2, the Global Address Map, shows the locations of the data structures for the various parts of the SGX card.

Table 4.2
Global Address Map

Part	*Function*	*Starting Address*[1]	*Ending Address*
ROM	Read-Only Memory	S4base+0x000000	S4base+0x1FFFFF
DAC	Digital-to-Analog Converter	S4base+0x200000	S4base+0x27FFFF
ALT	Alternate Address Space	S4base+0x280000	S4base+0x280FFF
FHC[2]	FBC Hardware Configuration	S4base+0x300000	S4base+0x300FFF
THC[2]	TEC Hardware Configuration	S4base+0x301000	S4base+0x301FFF
FBC[2]	Frame Buffer Controller	S4base+0x700000	S4base+0x700FFF
TEC[2]	Transformation Engine and Cursor	S4base+0x701000	S4base+0x701FFF
DFB[2]	Dumb Frame Buffer	S4base+0x800000	S4base+0xFFFFFF

1. *S4base is the base address of the card's slot.*
2. *These are internal select signals.*

The internal select signals (FHC, THC, FBC, TEC, and DFB) are synchronized with the internal clock. The internal clock is based on MAINOSC. Accesses to the ROM, DAC and ALT spaces are sent off-chip to external devices on the SGX card. These external transfers happen synchronously with the SBus clock.

The FHC register is not changed during normal operation. The *Single Chip GX Reference Card* shows the information contained in the FHC register. The FHC map is:

Starting Word Address	*Mnemonic*	*Function*
fhc+0x0000	CONFIG	Configuration Register

The THC registers are not changed during normal operation; they are loaded by the boot PROMs during diagnostics. The *Single Chip GX Reference Card* show the information contained in each of the registers.

Chapter 5
Memory Interfaces

This chapter describes how the L64845 SGX SBus Graphics Accelerator interfaces with memory. It is divided into two sections:

- Section 5.1, **ROM Interface**, describes the ROM interface to the PROM.
- Section 5.2, **Frame Buffer Interface**, describes the frame buffer interface to the VRAM. It discusses video memory architecture, data side timing, and video side timing.

Chapter 3, "Signal Definitions," defines the memory interface signals.

5.1 ROM Interface

The SGX accesses the boot PROM through the ROM address space. Transfers to this address space are always acknowledged with an 8-bit acknowledge ($\overline{SBACK}[2:0] = 5$). The SBus acknowledges write attempts but the external signal $\overline{CSROM}$ remains inactive. Reads to this space cause an eight SBus clock cycle output on the $\overline{CSROM}$ pin. The timing is designed for 250 ns or faster PROMs.

The ROM address space read cycle conforms to the general SBus read cycle described in Chapter 4, "SBus Interface."

Chapter 3, "Signal Definitions," defines the ROM interface signals.

The SGX card contains an FCode (FORTH) program in PROM that:

- Provides basic functionality
- Uniquely identifies the card
- Contains auto-configuration information
- Performs boot time initialization for the card
- Passes additional configuration information to the operating system

Figure 5.1 shows the ROM address space read cycle.

Table 5.1 describes the PROM fields. Table 5.2 shows the ROM map for the SGX card.

Figure 5.1
ROM Address
Space Read Cycle

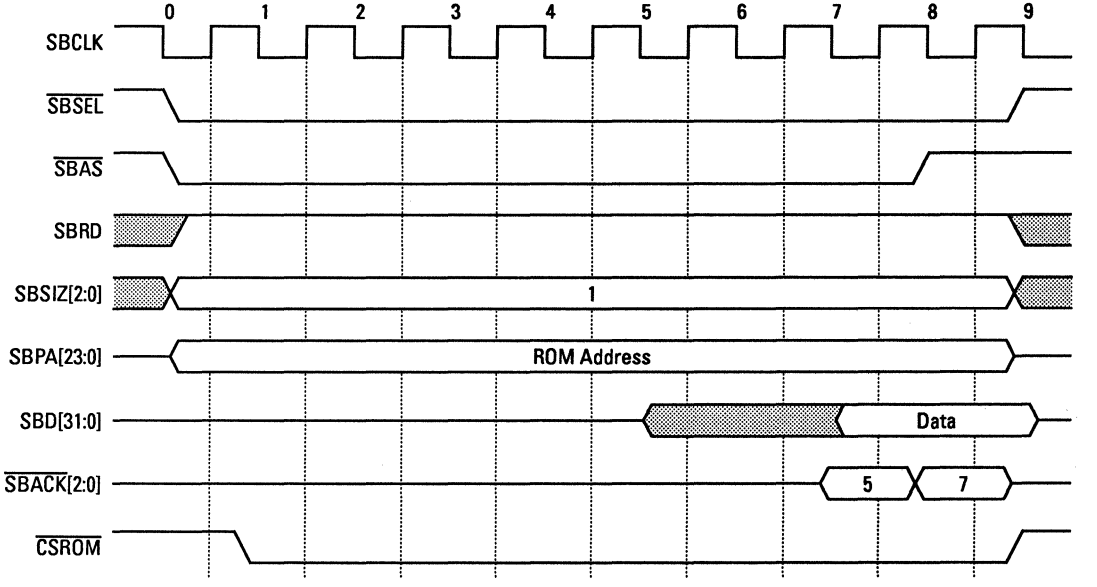


Table 5.1
PROM Fields

Size	*Field Name*	*Description*
One byte	Magic Token	Contains the constant 0xFD.
One byte	Version Number	Contains a unique number for each SGX card version.
Two bytes	Check Sum	Contains the checksum computed for the whole program.
Four bytes	Length	Contains the length of the header and the end token. (This is equal to the offset to the first byte that is not part of the program.)
Zero or more bytes	Program	Contains FCode (a FORTH program).
One byte	End Token	Contains the constant 0x00.

Table 5.2
ROM MAP

Starting Word Address	*Ending Word Address*	*Function*
ROM+0x0000	ROM+0x0003	SBus ID for SGX card
ROM+0x0004	ROM+0x3fff	Boot Code

5.2 Frame Buffer Interface

The SGX accesses a one megabyte frame buffer (eight 128K x 8, 100 ns VRAMs). This memory configuration provides a single frame buffer of 1152 x 900 resolution. In the future, additional memory configurations will be possible.

During a frame buffer access, the SGX provides the column address strobe ($\overline{CAS}$x), the row address strobe ($\overline{RAS}$x), the pixel write enables (WR[7:0]), and the appropriate address (VA[8:0]) to the VRAM.

The SGX passes eight pixels of data to the VRAM via the 64-bit pixel data bus, PDBxx[7:0]. During the serial output enable signal, $\overline{B0SOE0}$, the VRAM transfers pixels zero through three to the 32-bit serial input of the Palette DAC (RAMDAC) on VRAM signal lines P[D:A][7:0]. During the serial output enable signal, $\overline{B0SOE1}$, the VRAM transfers pixels four through seven on the same lines.

Chapter 3, "Signal Definitions," defines the frame buffer interface signals.

Refer to Appendix B, "SGX Card," for a block diagram of the memory architecture used on the SGX card.

VRAM Accesses

The VRAM address, VA[8:0], is common to all memory chips regardless of configuration. Configurations with more than 16 memory chips need external buffering on the VA[8:0] bus.

The address is multiplexed on VA[8:0] as row and column address. The VRAM rows do not correspond to scan lines on the screen. There is a new row address every 2048 pixels for the 128K x 8 VRAM configuration.

There are two types of VRAM accesses:

- Random Access – requires both a row and a column address
- Fast-Page-Mode Access – only requires a new column address This can be used when the row address for consecutive memory accesses does not change. The VRAMs "remember" the row address so that

subsequent addresses only require a column address and a column address strobe.

In either case, changing the row address ads a three cycle (129 ns) penalty to an access.

Reads

All reads are double accesses. The first access is to an even address and the second access is to an odd address. The first read is likely to be random access. Subsequent accesses to the same VRAM row are then fast-page-mode accesses. A random read takes 6 cycles (258 ns) and a fast-page-mode read takes three cycles (129 ns).

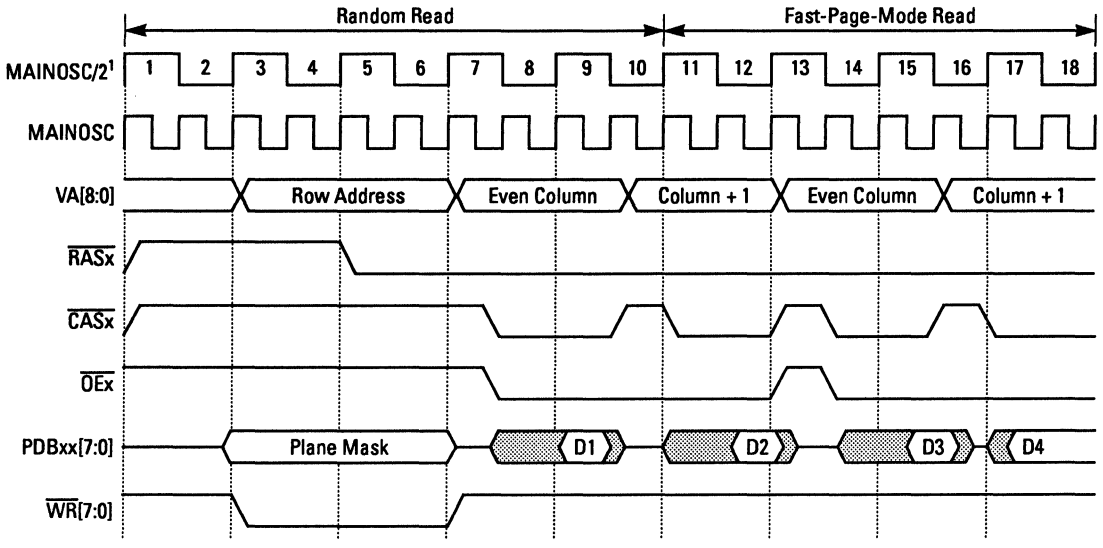
Figure 5.2 shows two read operations: a random read immediately followed by a fast-page-mode read. The random read starts with a $\overline{\text{RASx}}$ precharge high pulse lasting two clock cycles. The falling edge of $\overline{\text{RASx}}$ latches in the row address from the 9-bit VRAM address bus, VA[8:0].

The plane mask determines which bits of the pixel will be written. Although the plane mask is actually only used for writing to the VRAM, it is logically simpler to update the plane mask for every random access. Therefore, the write enable signals, $\overline{\text{WR}}$[7:0], are also asserted LOW at the falling edge of $\overline{\text{RASx}}$. $\overline{\text{WR}}$[7:0] update the pixel plane mask inside the VRAM.

The falling edge of $\overline{\text{CASx}}$ latches the column address into the VRAM. When $\overline{\text{CASx}}$ and $\overline{\text{OEx}}$ are both asserted, the VRAM data is driven onto the pixel data bus (PDBxx[7:0]).

The second falling edge of $\overline{\text{CASx}}$ without an intervening $\overline{\text{RASx}}$ identifies the following as a fast-page-mode access.

Figure 5.2
Read from VRAM



Note:
1. MAINOSC/2 is not available external to the SGX chip.

Writes

All SGX writes are buffered. First the SBus master writes to the SGX SBus slave interface (following the standard SBus slave protocol described in Chapter 4, "SBus Interface"). Then the SGX writes to the frame buffer as described in this section.

SGX writes can be divided into two types:

- Frame-Buffer Controller (FBC) writes are used when the SGX writes to the VRAM.
- Dumb Frame Buffer (DFB) writes are used when the CPU writes directly to the VRAM via the SBus interface.

FBC Writes

FBC writes are double accesses. The first write is usually random access. The second write is always fast-page-mode access. A random write takes five cycles (215 ns) and fast-page-mode write takes two cycles (86 ns). All drawing operations, font operations, and BLIT operations use FBC writes.

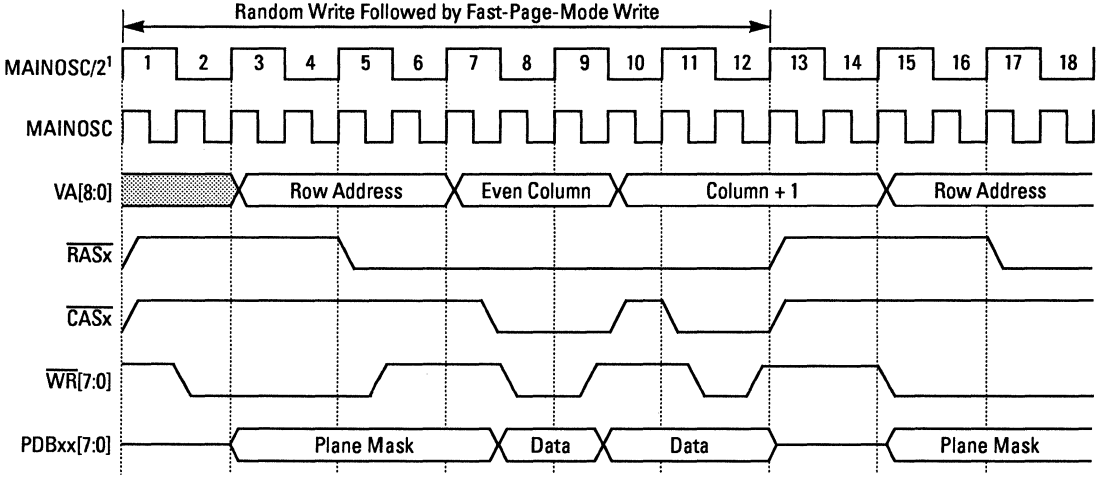
Figure 5.3 shows an FBC write. The first phase of a random write cycle is similar to a random read cycle, starting with a $\overline{RASx}$ precharge high pulse. The falling edge of $\overline{RASx}$ latches in the row address and plane mask.

The falling edge of $\overline{CASx}$ latches in the column address. The SGX uses a late write or $\overline{OEx}$-controlled write as opposed to an early write. This means that the write data is not latched into the VRAM until the falling edge of the write enable signal, $\overline{WR}$[7:0].

The write lines WR[7:0] determine which of the eight pixels are actually written to the VRAM. Any combination of pixels can be enabled by using a pixel mask register inside the SGX.

The SGX writes a total of 16 pixels during an FBC write. First it writes eight pixels to an even column address (with the first $\overline{CASx}$ pulse), then it writes the last eight pixels to the odd column address. Both writes are fast-page-mode accesses if the previous access had the same row address.

Figure 5.3
FBC Write to VRAM



Note:
1. MAINOSC/2 is not available external to the SGX chip.

Dead State

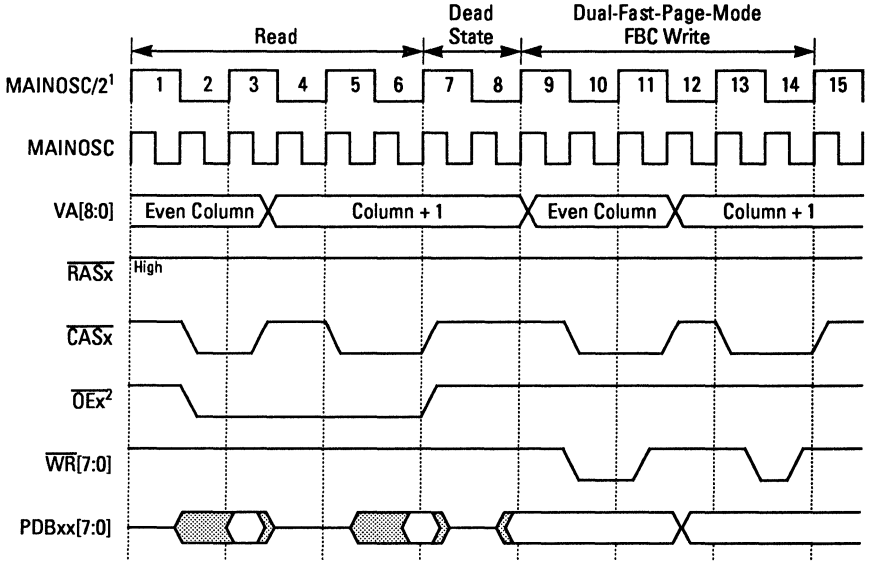
A dead state is required to change the direction of the pixel data bus from read to write.

A dead state is not required when going from a write to a read because all read cycles have half a clock cycle of dead time built into the beginning of the cycle.

Figure 5.4 shows a read followed by an FBC write.

The VRAM output enable signal, $\overline{OEx}$, is inactive during the dead state.

Figure 5.4
Read Followed by an FBC Write to VRAM



Note:
1. MAINOSC/2 is not available external to the SGX chip.
2. $\overline{OEx}$ is asserted high during the write.

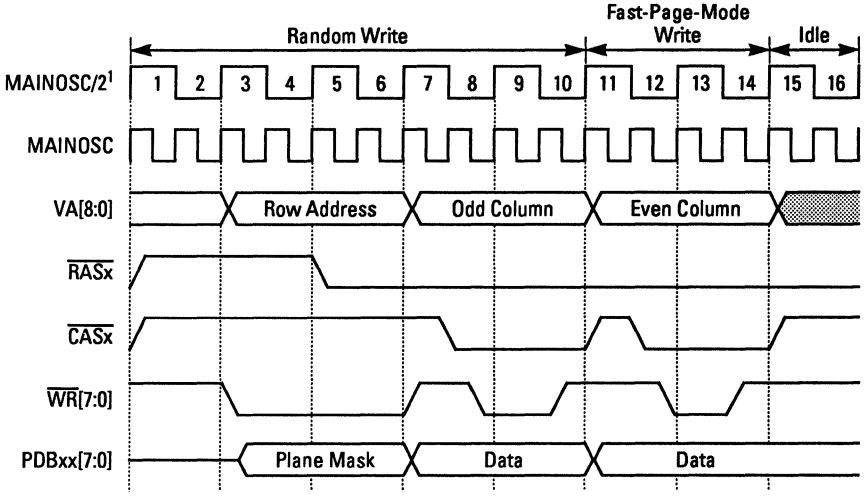
DFB Writes

DFB writes provide the CPU direct access to the frame buffer memory (VRAM) through the SBus. Since the SBus interface is only 32 bits wide, only single-word writes (four pixels) to the VRAM are allowed. DFB writes take three cycles (129 ns).

In DFB writes, a single $\overline{\text{CASx}}$ write cycle is used. $\overline{\text{CASx}}$ stays low for 1.5 clock cycles instead of the 1.0 clock period used for FBC accesses. Otherwise a DFB write is similar to an FBC write.

Figure 5.5 shows a DFB write.

Figure 5.5 DFB Write to VRAM



Note:
1. MAINOSC/2 is not available external to the SGX chip.

Refresh

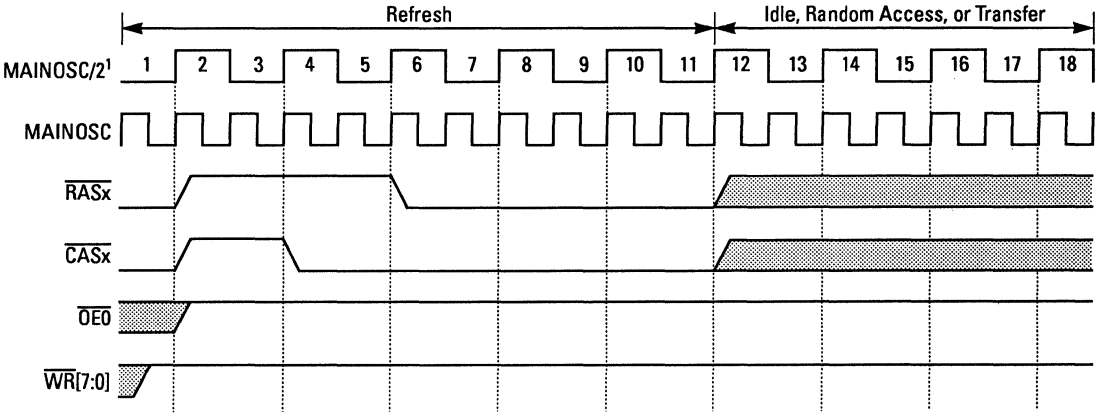
VRAM is a dynamic memory technology, using tiny capacitors to store information. These tiny capacitors must be recharged periodically. The 1-Mbit VRAMs supported by SGX must be refreshed every 8.0 ms. The SGX performs this task by refreshing one of the 512 rows every 15.6 ns.

A refresh takes five clocks cycles. $\overline{\text{CASx}}$-before-$\overline{\text{RASx}}$ refresh is used, which uses a row counter inside each VRAM to determine the row to be refreshed.

Refresh overwrites the row address stored in the VRAM so the first access after a refresh is forced to be a random access.

Figure 5.6 shows a refresh.

Figure 5.6
Refresh



Note:
1. MAINOSC/2 is not available external to the SGX chip.

Chapter 6
Video Interface

This chapter describes how the L64845 SGX SBus Graphics Accelerator interfaces with the video monitor. It is divided into three sections:

- Section 6.1, **Monitor Interface**, describes the monitor interface (video connector).
- Section 6.2, **DAC Interface**, describes the DAC interface to the RAMDAC.
- Section 6.3, **Video Transfer**, describes a serial transfer from the frame buffer (VRAM) to the RAMDAC.

For more information on the RAMDAC see Appendix C, "Bt458 RAMDAC," or the *Brooktree Product Databook.*

Chapter 3, "Signal Definitions," defines the video interface signals.

6.1 Monitor Interface

Information between the monitor and the SGX card is passed through a video cable connected to the video connector. The current Sun video cable contains three mini-coaxial cables as well as five twisted pairs. The video connector, mounted on the SGX card, is a 13-pin D-connector with three coaxial (75 ohm) pins for the red, green, and blue gun signals. The R, G and B video signals come from the RAMDAC (see Appendix C).

The composite horizontal and vertical sync signal ($\overline{\text{CSYNC}}$) feeds directly into the video connector from the SGX chip. Monitor identification information ($\overline{\text{ID}}$[2:0]) feeds directly into the SGX chip from the video connector sense pins.

The SGX provides $\overline{\text{CSYNC}}$, and $\overline{\text{BLANK}}$ (blanking signal) for the following types of monitors:

- 1152 x 900 @ 66 Hz
- 1152 x 900 @ 76 Hz

Chapter 3, "Signal Definitions," defines the video monitor interface signals.

Figure 6.1 shows the video connector. Table 6.1 shows the pin list for the video connector. Table 6.2 shows the encoding for the sense pins.

Figure 6.1
Video Connector

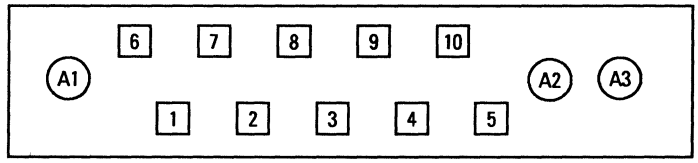


Table 6.1
Video Connector
Pin List

Pin	*Connection*[1]	*Usage*
1	N/C	Reserved
2	N/C	Reserved
3	$\overline{ID2}$	Sense Pin 2 (See Table 6.2)
4	GND	Ground (Referenced to Frame Buffer Ground)
5	$\overline{CSYNC}$	Combined Horizontal And Vertical Sync
6	N/C	Reserved
7	N/C	Reserved
8	$\overline{ID1}$	Sense Pin 1 (See Table 6.2)
9	$\overline{ID0}$	Sense Pin 0 (See Table 6.2)
10	GND	Ground (Return for Pin 5)
A1	R	Red
A2	G	Green (Gray Scale Video)
A3	B	Blue

1. N/C = Not Connected.

Table 6.2
Sense Pin
Allocation

Code	*Resolution and Scan Rate*[1]	*Sense Pin 2*[2]	*Sense Pin 1*[3]	*Sense Pin 0*
7	1152x900 66Hz (default)	N/C	N/C	N/C
6	1152x900 76Hz	N/C	N/C	GND
5	Reserved	N/C	GND	N/C
4	1152x900 76Hz	N/C	GND	GND
3	1152x900 66Hz	GND	N/C	N/C
2	Reserved	GND	N/C	GND
1	Reserved	GND	GND	N/C
0	Reserved	GND	GND	GND

1. Conformance may be waived.
2. GND = Logic ground, Pin 4
3. N/C = Not Connected

6.2 DAC Interface

The SGX is designed to interface directly to the Brooktree Bt458 RAMDAC. For specific information on the Bt458 RAMDAC see Appendix C, "Bt458 RAMDAC," or the *Brooktree Product Databook.*

The SGX accesses the RAMDAC registers through the DAC address space. This address space is treated as a 32-bit port on the SBus and always responds to an access with a 32-bit acknowledgment, $\overline{\text{SBACK}}$[2:0] = 3.

Although the SGX responds with a 32-bit acknowledgment, only the upper byte, SBD[31:24], maps into the RAMDAC's eight-bit port. Transfers occur synchronously with the SBus clock, with writes and reads both taking four SBus clock cycles. (The timing is designed for a Brooktree Bt458 RAMDAC or compatible part.)

Chapter 3, "Signal Definitions," defines the DAC interface signals.

Table 6.3 shows the address map for the SGX card's RAMDAC.

Table 6.3 RAMDAC Address Map

Starting Word Address	*Read Function*	*Write Function*
dac+0x0000	Read Address Register	Write Address Register
dac+0x0004	Read Color Palette	Write Color Palette
dac+0x0008	Read Control Register	Write Control Register
dac+0x000c	Read Overlay Color	Write Overlay Color

See Chapter 4, "Sbus Interface," for more information on the SBus address map for the SGX.

CPU accesses to the DAC address space conform to the standard SBus slave protocol described in Chapter 4, "Sbus Interface."

Figure 6.2 shows the DAC address space read cycle.

Figure 6.3 shows the DAC address space write cycle.

Figure 6.2
DAC Address Space
Read Cycle

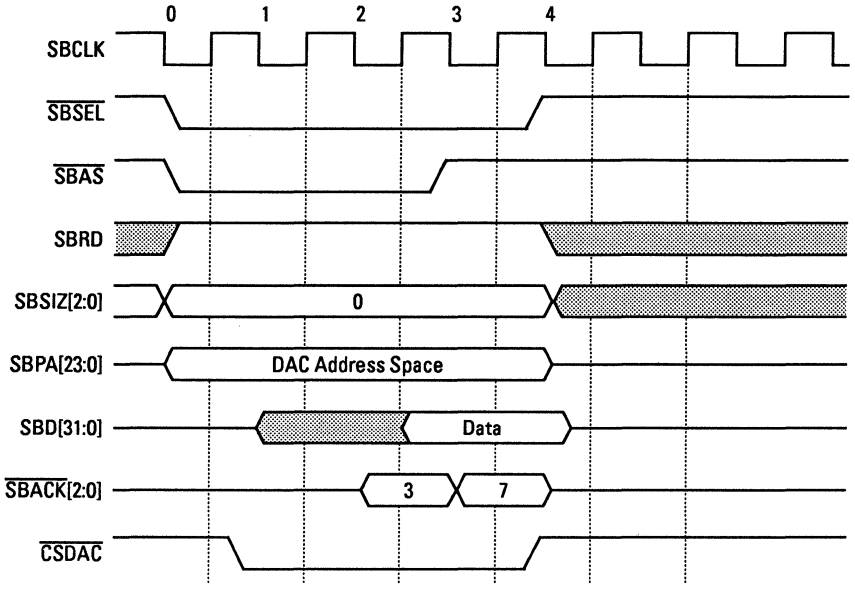
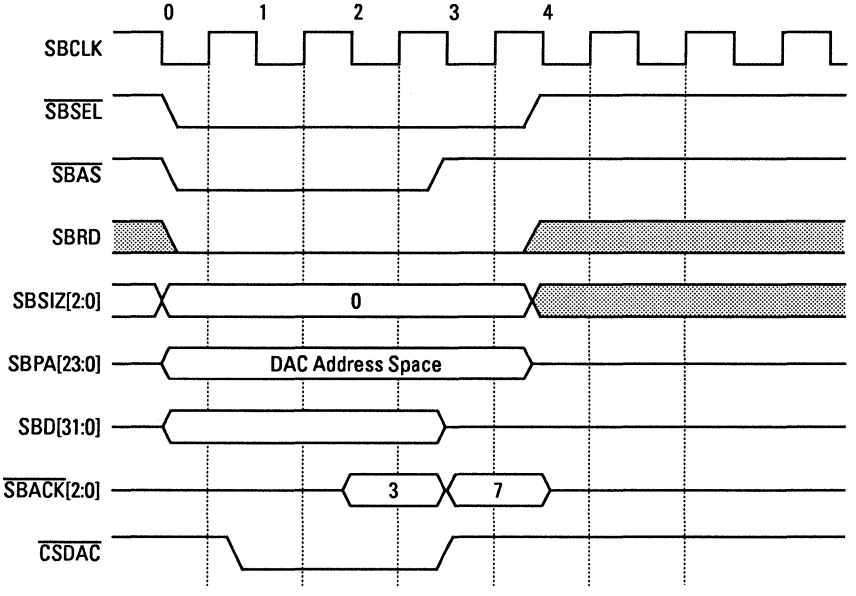


Figure 6.3
DAC Address Space
Write Cycle



6.3 Video Transfers

VRAMs are dual ported, having a serial access port and a random access port. In the realtime read transfer cycle, the VRAM loads one row of data into its serial access memory in parallel and then shifts it out while the SGX is using the random access port.

Figure 6.4 shows a realtime read transfer and serial read.

The transfer cycle starts when $\overline{OEx}$ falls while $\overline{RASx}$ is high. The transfer row address is latched into the VRAM by the falling edge of $\overline{RASx}$. When $\overline{CASx}$ falls, the VRAM latches in the starting address for the serial shifter (which is always zero on the SGX card). The end of the $\overline{OEx}$ (LOW) pulse, which is synchronized with the video shift clock (SCKx), triggers the actual data transfer from the main memory to the serial access memory.

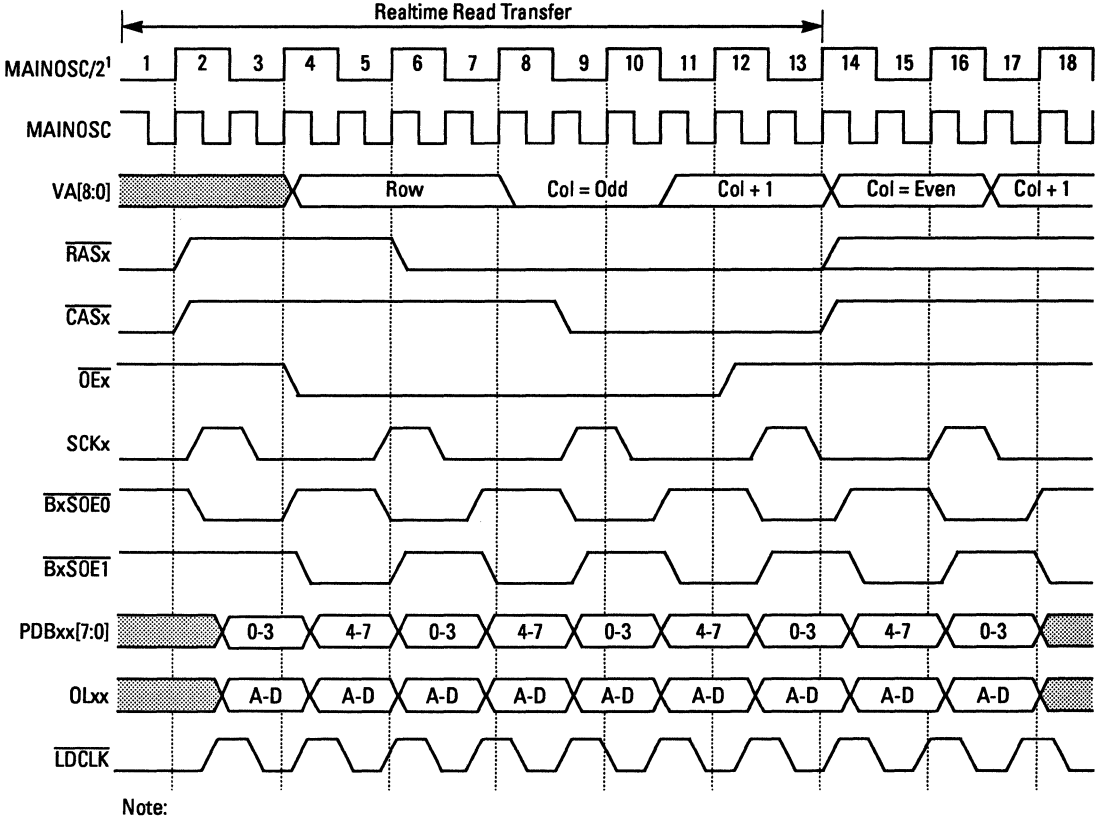
In a serial port read, the video shift clock advances the serial shifter in all the VRAMs simultaneously. The serial output enable signals, $\overline{BxSOE0}$ and $\overline{BxSOE1}$, alternate to enable four pixels at a time onto the pixel bus, PDBxx[7:0].

The DAC loads pixel and overlay data on the rising edge of $\overline{LDCLK}$. The overlay data, used for the hardware cursor, comes directly from the SGX.

In SGX configurations with 128K x 8 VRAMs, a video transfer occurs every 2048 pixels.

For more information on the internal workings of the RAMDAC see Appendix C or the *Brooktree Product Databook.*

Figure 6.4
Realtime Read Transfer and Serial Read



Note:
1. MAINOSC/2 is not available external to the SGX chip.

Chapter 7
Alternate Interface

This chapter describes how the L64845 SGX SBus Graphics Accelerator interfaces with an alternate external device.

The alternate (ALT) address space has been set aside for future use as an interface for an SGX external device.

The SBus views the ALT address space as a synchronous 32-bit port. Writes and reads both take five SBus clock cycles. ALT address space accesses conform to the general SBus slave protocol described in Chapter 4, "SBus Interface."

Chapter 3, "Signal Definitions," defines the alternate interface signals.

Figure 7.1 shows the ALT address space read cycle.

Figure 7.2 shows the ALT address space write cycle.

Figure 7.1
ALT Address Space Read Cycle

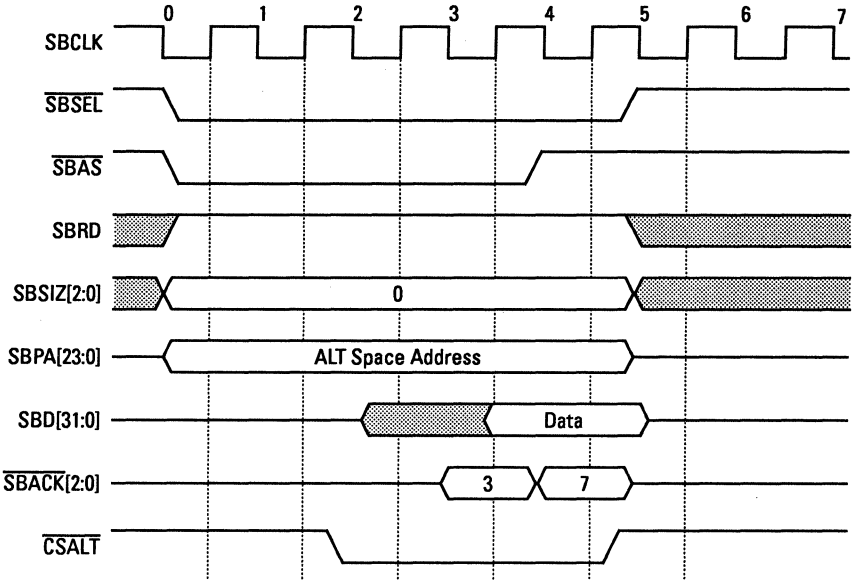
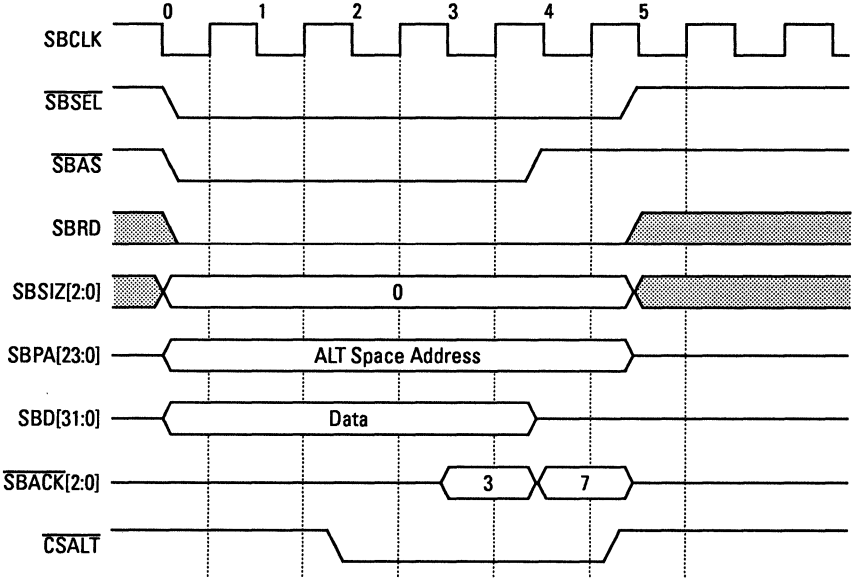


Figure 7.2
ALT Address Space Write Cycle



Chapter 8
Specifications

This chapter specifies the L64845 SGX SBus Graphics Accelerator's electrical and mechanical characteristics. It is divided into three sections:

- Section 8.1, **AC Timing**, describes the AC timing specifications and waveforms.
- Section 8.2, **Electrical Requirements**, describes the electrical characteristics and requirements.
- Section 8.3, **Packaging**, provides the pinout and mechanical dimensions for the L64845 package.

Appendix B, "SGX Card," contains more information on the SGX card.

8.1 AC Timing

This section provides the AC timing specifications for the SGX and the SBus.

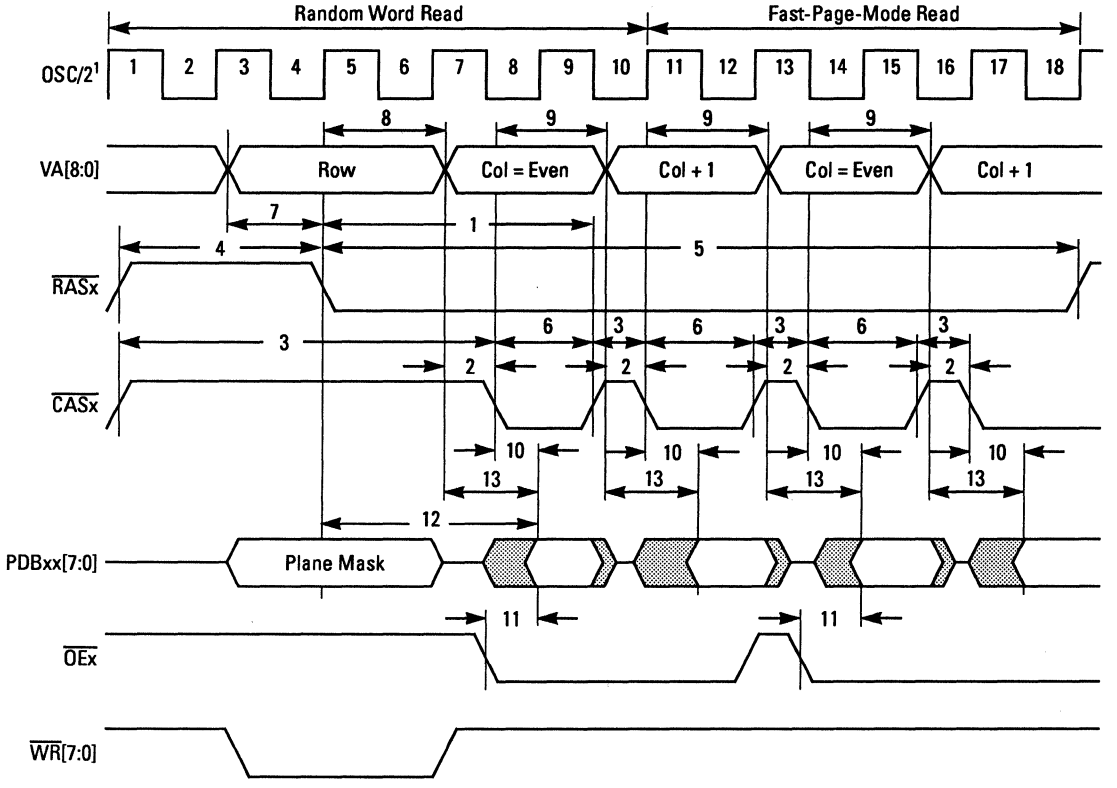
Figure 8.1 shows the SGX read cycle. Figure 8.2 shows the SGX write cycle. Figure 8.3 shows the CAS before RAS refresh. Figure 8.4 shows the real time read transfer. Figure 8.5 shows the horizontal sync and blanking cycle (for the 66 Hz configuration on the video side of the SGX). Note that there are 94 pixel groups per horizontal line and each pixel group contains 16 pixels. Figure 8.6 shows the vertical sync and blanking cycle (for the 66 Hz configuration on the video side of the SGX). Note that there are 937 horizontal lines.

Table 8.1 shows the SGX memory control AC characteristics. Table 8.2 shows the pin loading assumptions.

Figure 8.7 shows some SBus signals with rise and fall times. Figure 8.8 shows the SBus reset. Figure 8.9 shows the SBus read cycle. Figure 8.10 shows the SBus write cycle.

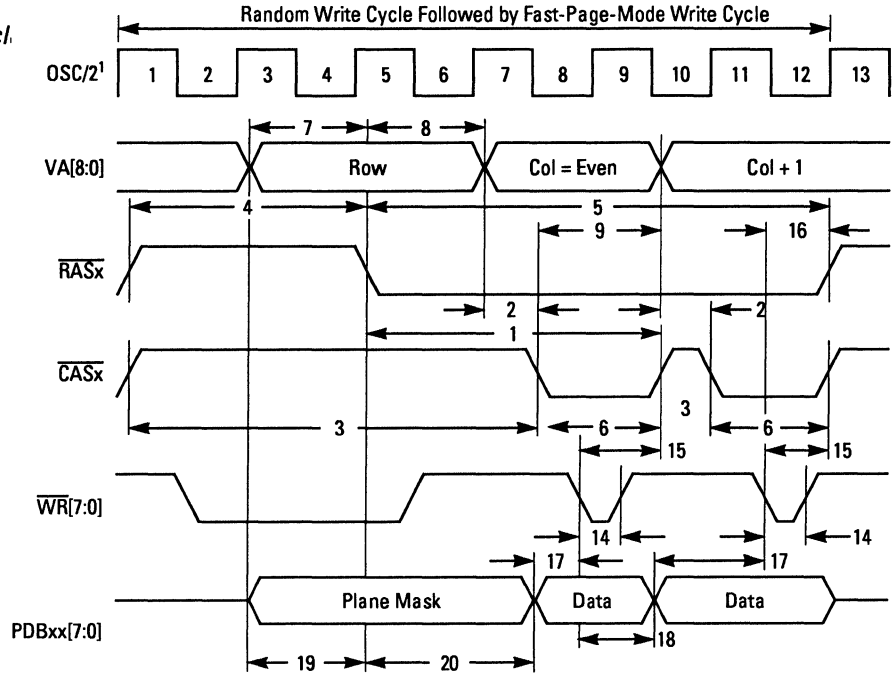
Table 8.3 shows the SGX SBus AC characteristics.

Figure 8.1
SGX Read Cycle



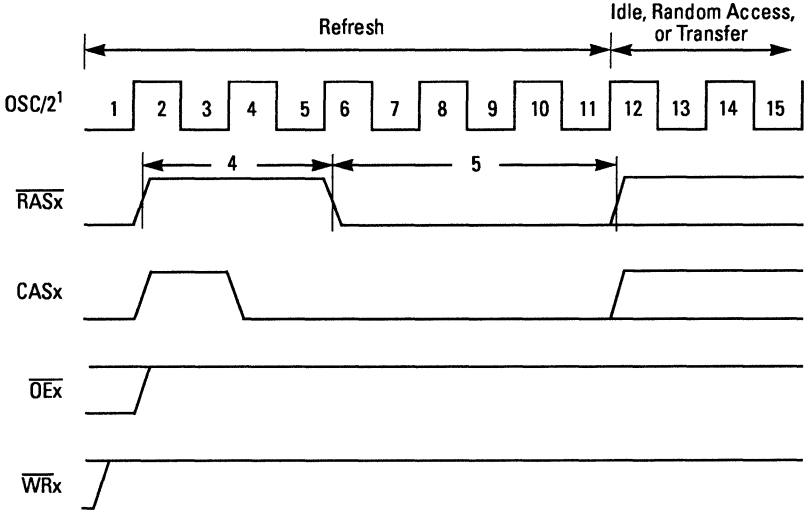
Note:
1. OSC/2 is not available external to the SGX chip.

Figure 8.2
SGX FBC Write Cycl



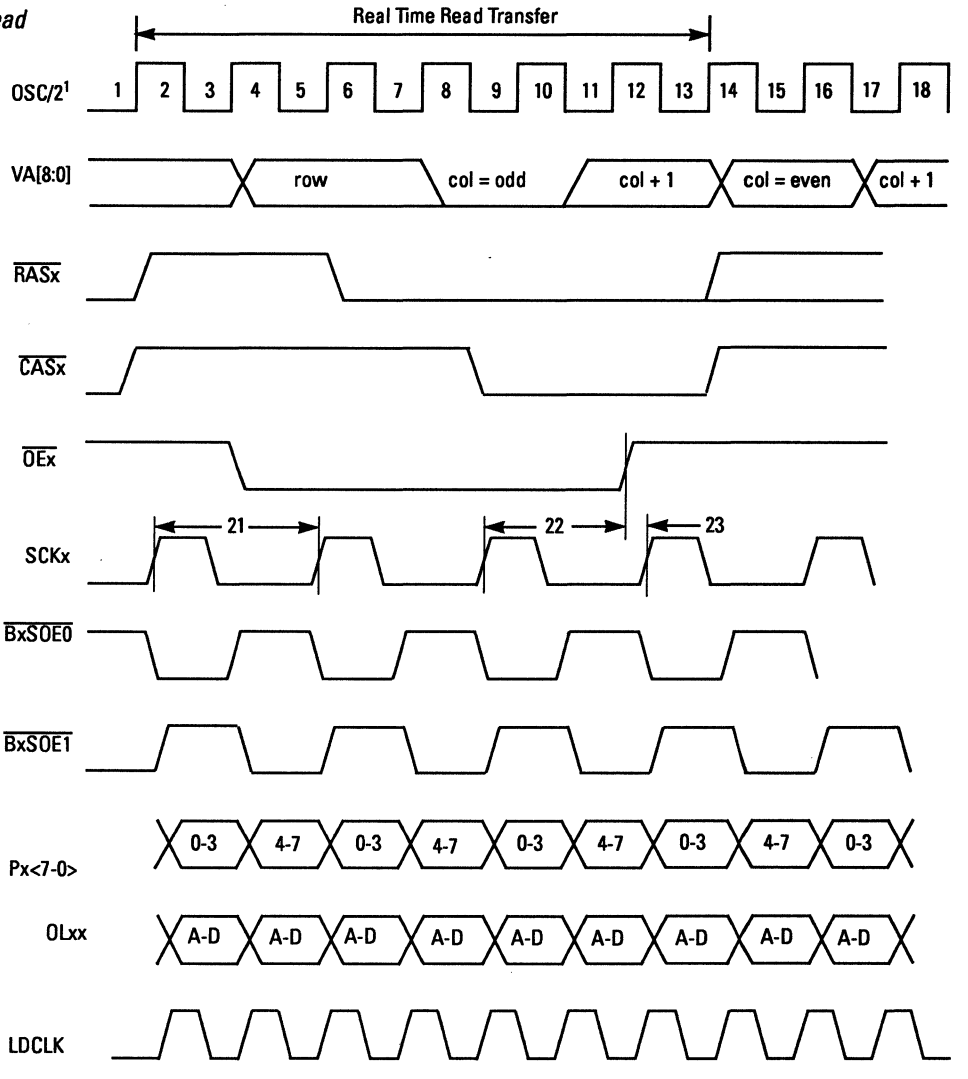
Note:
1. OSC/2 is not available external to the SGX chip.

Figure 8.3
$\overline{CAS}$ Before $\overline{RAS}$ Refresh



Note:
1. OSC/2 is not available external to the SGX chip.

Figure 8.4
Real Time Read Transfer



Note:
1. OSC/2 is not available external to the SGX chip.

Figure 8.5
Horizontal Sync and Blanking Cycle

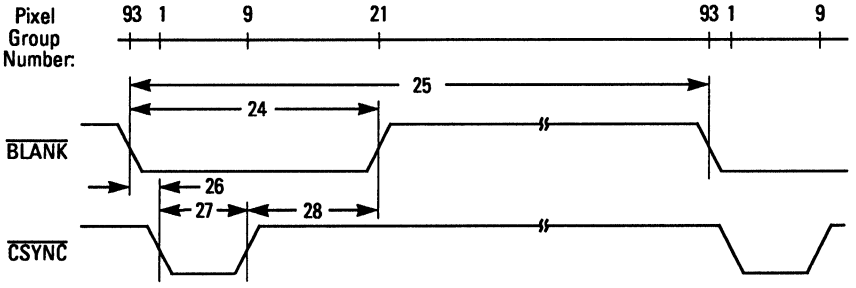


Figure 8.6
Vertical Sync and Blanking Cycle

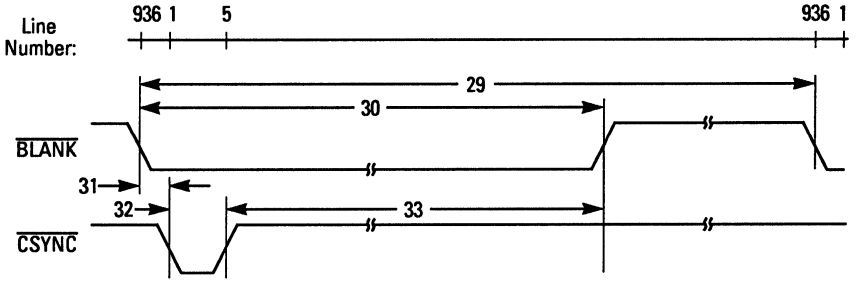


Table 8.1
Memory Control AC Characteristics

	Symbol	*From (Output)*	*To (Output)*	*Timing*	*Min*	*Max*	*Units*
Common Timing							
1.	t_{CSH}	$\overline{RAS}$x Low	$\overline{CAS}$x High	$\overline{CAS}$ Hold	100	–	ns
2.	t_{ASC}	VAx Valid	$\overline{CAS}$x Low	$\overline{CAS}$ Setup	1	–	ns
3.	t_{CP}	$\overline{CAS}$x High	$\overline{CAS}$x Low	High Pulse Width	20	–	ns
4.	t_{RP}	$\overline{RAS}$x High	$\overline{RAS}$x Low	High Pulse Width	80	–	ns
5.	t_{RAS}	$\overline{RAS}$x Low	$\overline{RAS}$x High	Low Pulse Width	100	–	ns
6.	t_{CAS}	$\overline{CAS}$x Low	$\overline{CAS}$x High	Low Pulse Width	30	–	ns
7.	t_{ASR}	VAx Valid	$\overline{RAS}$x Falling	$\overline{RAS}$ Setup	1	–	ns
8.	t_{RAH}	$\overline{RAS}$x Low	VAx Invalid	Hold	15	–	ns
9.	t_{CAH}	$\overline{CAS}$x Low	VAx Invalid	Hold	20	–	ns
Read Timing							

(Sheet 1 of 3)

Table 8.1 (Continued)
Memory Control AC Characteristics

	Symbol	*From (Output)*	*To (Output)*	*Timing*	*Min*	*Max*	*Units*
10.	t_{CAC}	$\overline{CASx}$ Low	PDBxxx Valid	$\overline{CAS}$ Access	–	30	ns
11.	t_{OEA}	$\overline{OEx}$ Low	PDBxxx Valid	$\overline{OE}$ Access	–	30	ns
12.	t_{RAC}	$\overline{RASx}$ Low	PDBxxx Valid	$\overline{RAS}$ Access	–	100	ns
13.	t_{AA}	VAx Valid	PDBxxx Valid	Address Access	–	50	ns
Write Timing							
14.	t_{WP}	WRx Low	WRx High	Low Pulse Width	20	–	ns
15.	t_{CWL}	WRx Low	$\overline{CASx}$ High		30	–	ns
16.	t_{RWL}	WRx Low	$\overline{RASx}$ High		30	–	ns
17.	t_{DS}	PDBxxx Low	WRx Low	Data Setup	1	–	ns
18.	t_{DH}	WRx Low	PDBxxx Valid	Data Hold	25	–	ns
19.	t_{MS}	PDBxxx Valid	$\overline{RASx}$ Low	Setup	1	–	ns
20.	t_{MH}	$\overline{RASx}$ Low	PDBxxx Invalid	Hold	15	–	ns
Video Output							
21.	t_{SC}	SCKx High	SCKx High		35	–	ns
22.	t_{SL}	SCKx High	$\overline{OEx}$ High		10	–	ns
23.	t_{SD}	$\overline{OEx}$ High	SCKx High		15	–	ns
24.	t_{HBL}	$\overline{BLANK}$ Low	$\overline{BLANK}$ High	Horizontal Blank Length[1]	3.787	3.787	µs
25.	t_{HBP}	$\overline{BLANK}$ Low	$\overline{BLANK}$ Low	Horizontal Blank Period	16.182	16.182	µs
26.	t_{HSS}	$\overline{BLANK}$ Low	$\overline{HSYNC}$ Low		0.344	0.344	µs
27.	t_{HLP}	$\overline{HSYNC}$ Low	$\overline{HSYNC}$ High	Horizontal Sync Hold	1.377	1.377	µs
28.	t_{HBH}	$\overline{HSYNC}$ High	$\overline{BLANK}$ High		2.064	2.064	µs
29.	t_{VBP}	$\overline{BLANK}$ Low	$\overline{BLANK}$ Low	Vertical Blank Period	15.163	15.163	ms
30.	t_{VBL}	$\overline{BLANK}$ Low	$\overline{BLANK}$ High	Vertical Blank Length[2]	599	599	µs
31.	t_{BVS}	$\overline{BLANK}$ Low	$\overline{VSYNC}$ Low	Vertical Sync Setup	32	32	µs

(Sheet 2 of 3)

Table 8.1 (Continued)
Memory Control AC Characteristics

	Symbol	*From (Output)*	*To (Output)*	*Timing*	*Min*	*Max*	*Units*
32.	t_{VSH}	$\overline{\text{VSYNC}}$ Low	$\overline{\text{VSYNC}}$ High	Vertical Sync Hold	65	65	µs
33.	t_{VSB}	$\overline{\text{VSYNC}}$ High	$\overline{\text{BLANK}}$ High		502	502	µs

(Sheet 3 of 3)

1. Horizontal Blank Width is 352 pixels.
2. Vertical Blank Width is 37 lines.

Table 8.2
Pin Loading Assumptions

Output	*Signal Type*	*Loading*	*Units*
BLANK	Output	15	pF
B0SOE[1:0]	Output	60	pF
B1SOE[1:0]	Output	60	pF
CAS[1:0]	Output	120	pF
CSYNC	Output	15	pF
CSDAC	Output	25	pF
CSALT	Output	25	pF
CSROM	Output	25	pF
DOTCLK	Output	20	pF
LDCLK	Output	15	pF
OE[1:0]	Output	120	pF
OL[H:A][1:0]	Output	15	pF
PDBxx[7:0]	Bidirectional, internal pullup	30	pF
PERFOSC	Output	10	pF
RAS[1:0]	Output	120	pF
$\overline{\text{SBACK}}$[2:0]	Output	120	pF
SBD[31:0]	Bidirectional, internal pullup	160	pF
SBIRQ5	Open-drain output	120	pF
SCK[1:0]	Output	120	pF
VA[8:0]	Output	100	pF
WE[1:0]	Output	50	pF
WR[7:0]	Output	40	pF

Figure 8.7
SBus Signals with Rise/Fall Times

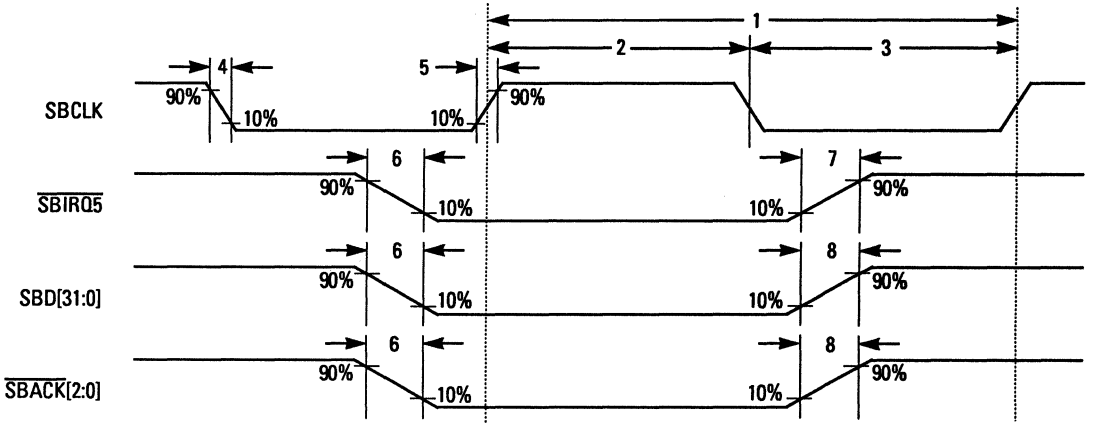


Figure 8.8
SBus Reset

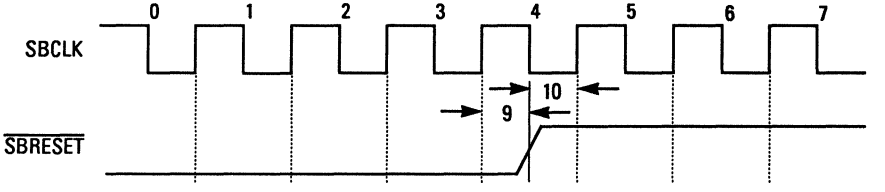


Figure 8.9
SBus Read
Cycle

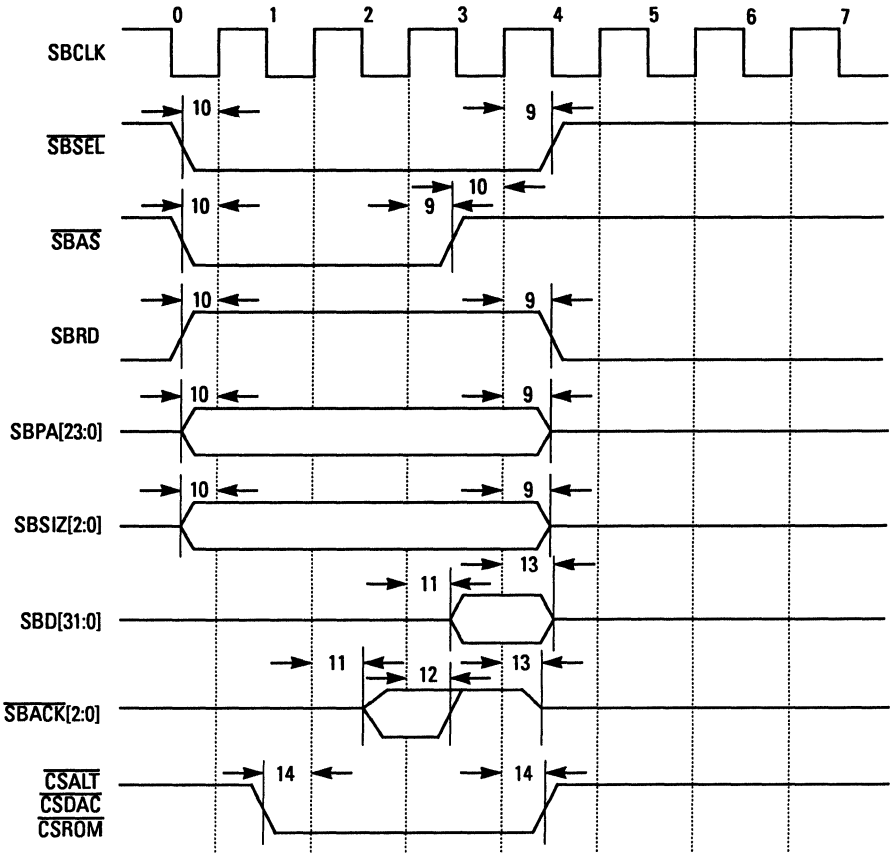


Figure 8.10
SBus Write
Cycle

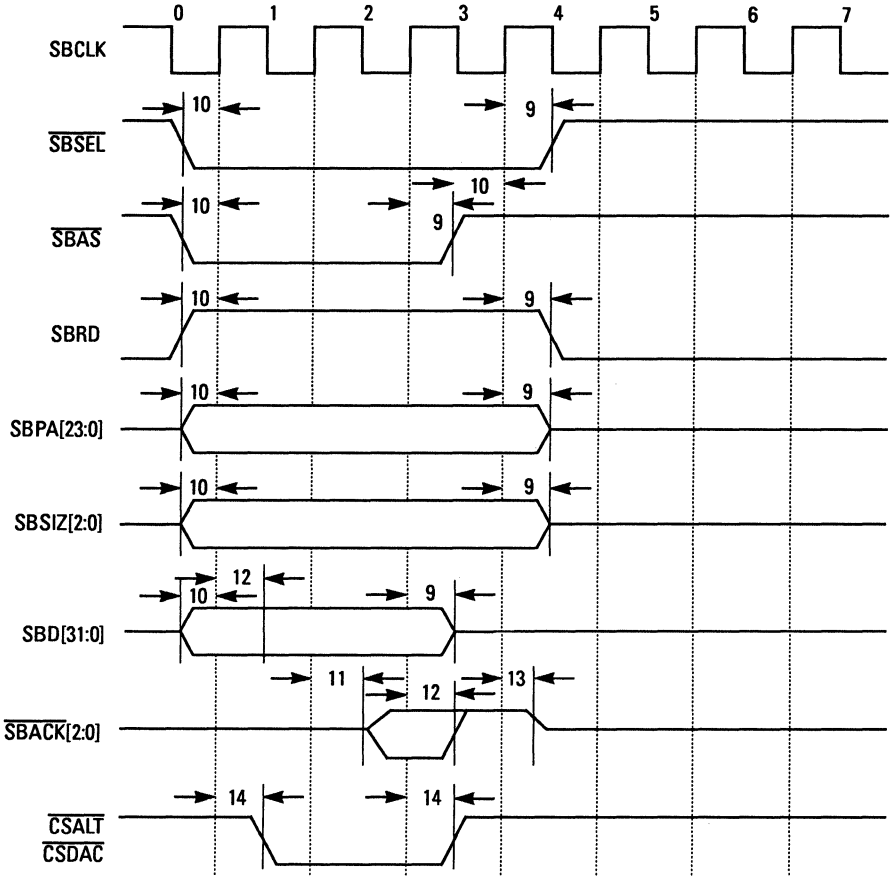


Table 8.3
SGX SBus AC Characteristics

Parameter		*Description*	*Min*	*Max*	*Units*	*Load (pF)*
1	t_{CP}	Clock Period	40	60	ns	
2	t_{CH}	Clock High Time	17		ns	
3	t_{CL}	Clock Low Time	17		ns	
4	t_{CF}	Clock Fall Time	1	3	ns	160
5	t_{CR}	Clock Rise Time	1	3	ns	160
6	t_F	$\overline{\text{SBIRQ5}}$[1], $\overline{\text{SBACK}}$, SBD Fall Time	5	20	ns	160
7	t_{IR}	$\overline{\text{SBIRQ5}}$ Rise Time	5	1200	ns	160
8	t_R	SBACK, SBD Rise Time	5	20	ns	160
9	t_{IH}	SBSIZ, SBRD, $\overline{\text{SBAS}}$, $\overline{\text{SBSEL}}$, SBD, SBPA, SBRESET[2] Hold from SBCLK High			ns	160
10	t_{IS}	SBSIZ, SBRD, $\overline{\text{SBAS}}$, $\overline{\text{SBSEL}}$, SBD, SBPA, SBRESET Setup to SBCLK High	15		ns	160
11	t_{OD25}	SBD, $\overline{\text{SBACK}}$ Valid from SBCLK High (25 MHz)	2.5	20	ns	100
12	t_{OH}	SBD, $\overline{\text{SBACK}}$ Hold from SBCLK High	2.5		ns	
13	t_Z	SBD, $\overline{\text{SBACK}}$ 3-State from SBCLK High		35	ns	
14	t_{CSD}	$\overline{\text{CSROM}}$, $\overline{\text{CSDAC}}$, $\overline{\text{CSALT}}$ Valid/Invalid from SBCLK High	0	20	ns	

1. $\overline{\text{SBIRQ5}}$ is not synchronous with SBCLK. The CPU clears $\overline{\text{SBIRQ5}}$.
2. Asserting $\overline{\text{SBRESET}}$ doesn't require timing. $\overline{\text{SBRESET}}$ must remain asserted for at least 512 SBCLK cycles.

8.2 Electrical Requirements

This subsection specifies the electrical requirements for the L64845. Five tables list electrical data in the following categories:

- Absolute Maximum Ratings (Table 8.4)
- Recommended Operating Conditions (Table 8.5)
- Capacitance (Table 8.6)
- DC Characteristics (Table 8.7)
- Pin Description Summary (Table 8.8)

Table 8.4 Absolute Maximum Ratings

Symbol	*Parameter*	*Limits*[1]	*Units*
VDD	DC Supply	-0.3 to +7	V
VIN	Input Voltage	-0.3 to VDD +0.3	V
IIN	DC Input Current	±10	mA
TSTG	Storage Temperature Range (Ceramic)	-65 to +150	°C

1. Referenced to VSS.

Table 8.5 Recommended Operating Conditions

Symbol	*Parameter*	*Limits*	*Units*
VDD	DC Supply	+4.75 to +5.25	V
TA	Ambient Temperature	0 to +70	°C

Table 8.6 Capacitance

Symbol	*Parameter*[1]	*Min*	*Max*	*Units*
CIN	Input Capacitance		15.5	pF
COUT	Output Capacitance		15	pF
CBID	Bidirectional Capacitance		15.5	pF

1. Measurement conditions are VIN = 5.0 V, TA = 25 °C, and clock frequency = 1 MHz

Table 8.7 DC Characteristics

Symbol	*Parameter*	*Condition*[1]	*Min*	*Typ*	*Max*	*Units*
VIH	Voltage Input High		2.0	–	–	V
VIL	Voltage Input Low		–	–	0.8	V
VOH	Voltage Output High	IOH = -1.0, -2.0, -4.0, -6.0, -8.0 mA	2.4	4.5	–	V

Table 8.7 (Continued)
DC Characteristics

Symbol	*Parameter*	*Condition*[1]	*Min*	*Typ*	*Max*	*Units*
VOL	Voltage Output Low	IOL = 1.0, 2.0, 4.0, 6.0, 8.0 mA	–	0.2	0.4	V
VT+	Schmitt Trigger, Positive-going Threshold	TTL	–	1.8	2.0	V
VT-	Schmitt Trigger, Negative-going Threshold	TTL	0.7	0.9	–	V
IIL	Current Input Leakage	VDD = Max, VIN = VDD or VSS	-10	±1	10	µA
IIPU	Input Pull-Up Current	VDD = Max, VIN = VSS or 3.5 V	-35	-115	-350	µA
IOL	Current Output Leakage (non-3-statable)	VDD = Max, VOUT = VSS or VDD	-1.0	±1	1.0	µA
IOZ	Current 3-State Output Leakage	VDD = Max, VOUT = VSS or VDD	-10	±1	10	µA
IOZU	Current 3-State Output w/ Pull-up	VOUT = VSS or 3.5 V	-10	–	10	µA
IOSP1	Current P-Channel Output Short Circuit (1 mA Output Buffers)[2]	VDD = Max, VOUT = VSS	-7.5	–	-37	mA
IOSP2	Current P-Channel Output Short Circuit (2 mA Output Buffers)[2]	VDD = Max, VOUT = VSS	-15	–	-74	mA
IOSP4	Current P-Channel Output Short Circuit (4 mA Output Buffers)[2]	VDD = Max, VOUT = VSS	-30	–	-148	mA
IOSP6	Current P-Channel Output Short Circuit (6 mA Output Buffers)[2]	VDD = Max, VOUT = VSS	-45	–	-222	mA
IOSP8	Current P-Channel Output Short Circuit (8 mA Output Buffers)[2]	VDD = Max, VOUT = VSS	-60	–	-296	mA
IOSN1	Current N-Channel Output Short Circuit (1 mA Output Buffers)[2]	VDD = Max, VOUT = VDD	9.5	–	32.5	mA
IOSN2	Current N-Channel Output Short Circuit (2 mA Output Buffers)[2]	VDD = Max, VOUT = VDD	19	–	65	mA
IOSN4	Current N-Channel Output Short Circuit (4 mA Output Buffers)[2]	VDD = Max, VOUT = VDD	38	–	130	mA
IOSN6	Current N-Channel Output Short Circuit (6 mA Output Buffers)[2]	VDD = Max, VOUT = VDD	57	–	195	mA
IOSN8	Current N-Channel Output Short Circuit (8 mA Output Buffers)[2]	VDD = Max, VOUT = VDD	76	–	260	mA
IDD	Quiescent Supply Current	VIN = VDD or VSS	–	–	2	mA
ICC	Dynamic Supply Current	VDD = Max, f = 25MHz	–		300	mA

1. Specified at VDD equals 5V ± 5% at ambient temperature over the specified range.
2. Not more than one output may be shorted at a time for a maximum duration of one second.

Table 8.8
Pin Description Summary

Mnemonic	*Description*	*Input/Output*	*Drive (mA)*	*Active*
$\overline{\text{ALTOSC}}$	Alternative Oscillator Input	Schmidt trigger TTL input, internal pullup		Low
$\overline{\text{BLANK}}$	Blanking	Output	2	Low
$\overline{\text{B0SOE}}$[1:0]	VRAM Serial Output Enable for Buffer 0	Output	4	Low
$\overline{\text{B1SOE}}$[1:0]	VRAM Serial Output Enable for Buffer 1	Output	4	Low
$\overline{\text{CAS}}$[1:0]	Column Address Strobe	Output	6	Low
$\overline{\text{CSYNC}}$	Composite Synch	Output	2	Low
$\overline{\text{CSDAC}}$	DAC Chip Select	Output	2	Low
$\overline{\text{CSALT}}$	Alternate Chip Select	Output	2	Low
$\overline{\text{CSROM}}$	ROM Chip Select	Output	2	Low
$\overline{\text{DOTCLK}}$	Video Dot Clock	Output	4	Low
EXTXFR	External Transfer	Input, internal pullup		High
$\overline{\text{ID}}$[2:0]	Identification	Input, internal pullup		Low
$\overline{\text{LDCLK}}$	Load Clock	Output	2	Low
MAINOSC	Main Oscillator Input	Schmidt trigger TTL input, internal pullup		High
$\overline{\text{OE}}$[1:0]	Output Enable	Output	6	Low
OL[H:A][1:0]	Overlay	Output	2	High
PDBxx[7:0]	Pixel Data Bus	Bidirectional, internal pullup	4	–
PERFOSC[1]	Performance Oscillator	Output	1	High
$\overline{\text{RAS}}$[1:0]	Row Address Strobe	Output	6	Low
$\overline{\text{SBACK}}$[2:0]	SBus Transfer Acknowledge	Output	8	Low
$\overline{\text{SBAS}}$	SBus Address Strode	Input, internal pullup		Low
SBCLK	SBus Clock	Schmidt trigger TTL input, internal pullup		High
SBD[31:0]	SBus Data Bus	Bidirectional, internal pullup	4	–
$\overline{\text{SBIRQ5}}$	SBus Interrupt for Vertical Blanking	Open-drain output	4	Low
SBRD	SBus Read/Write	Input, internal pullup		High
$\overline{\text{SBRESET}}$	SBus Reset	Schmidt trigger TTL input, internal pullup		Low
$\overline{\text{SBSEL}}$	SBus Slave Select	Input, internal pullup		Low
SBSIZ[2:0]	SBus Data Size	Input, internal pullup		High
SCK[1:0]	Serial Shift Clock	Output	6	High
VA[8:0]	VRAM Address	Output	4	High
TESTMODE[1]	Test Mode	Schmidt trigger input, internal pullup		High
$\overline{\text{WE}}$[1:0]	Write Enable for Double Buffering	Output	2	Low
WR[7:0]	Write Enable	Output	4	High

1. For testing and debugging purposes only.

8.3 Packaging

LSI Logic supplies the SGX in a 223-pin Ceramic Pin Grid Array (CPGA).

Note there is no pin 1 on the SGX; It is intentionally omitted to eliminate any possibility of misalignment. Pin 2 is the leftmost pin in the corner on the top row (near the SBus connector) on the component side.

Figure 8.11 shows the SGX pinout (top view). Table 8.9 shows the SGX pin list. Figure 8.12 shows the L64845 SGX mechanical drawing.

Figure 8.11
L64845 SGX Pinout

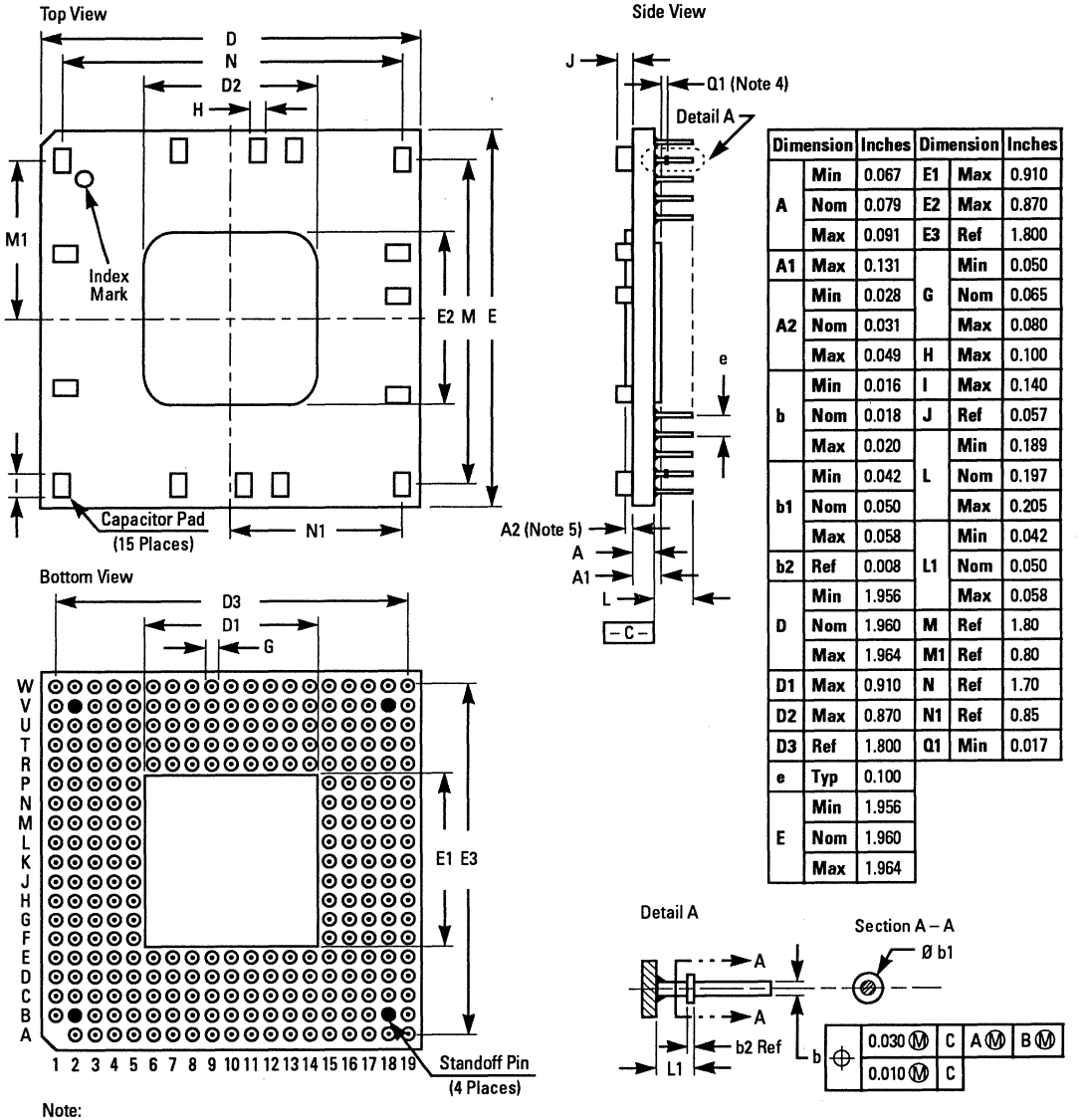
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A		VDD	SBPA4	SBPA0	SBSIZ1	SBD29	SBD25	VSS	VDD	VSS	SBD19	SBD15	SBD11	SBD7	SBD2	SBCLK	VDD	VSS
B	VDD	VSS	SBPA5	SBPA1	$\overline{\text{SBIRQ5}}$	SBD31	SBD27	SBD23	SBD21	SBD20	SBD17	SBD13	SBD9	SBD5	SBD1	$\overline{\text{SBSEL}}$	VSS	VDD
C	SBPA9	SBPA8	SBPA6	SBPA2	*N/C*	SBSIZ2	SBD30	SBD26	SBD22	SBD18	SBD14	SBD10	SBD6	SBD4	$\overline{\text{SBAS}}$	VA7	VA6	VA5
D	$\overline{\text{CAS1}}$	$\overline{\text{OE1}}$	SBPA11	SBPA7	SBPA3	SBRD	SBSIZ0	SBD28	SBD24	SBD16	SBD12	SBD8	SBD3	SBD0	VA8	VA3	VA2	VA1
E	SBPA15	SBPA13	SBPA12	SBPA10											VA4	PDB7F6	PDB7F4	PDB7F2
F	SBPA19	SBPA17	SBPA14	$\overline{\text{RAS1}}$											VA0	PDB7F7	PDB7F0	PDB6E7
G	SBPA22	SBPA21	SBPA18	SBPA16											PDB7F5	PDB7F3	PDB6E5	PDB6E3
H	VSS	SBPA23	MAIN OSC	SBPA20											PDB7F1	WR7	PDB6E2	PDB6E1
J	VDD	$\overline{\text{SB RESET}}$	$\overline{\text{DOTCLK}}$	ALTOSC				Top View							PDB6E6	PDB6E4	WR6	VDD
K	VSS	EXTXFR	$\overline{\text{SBACK0}}$	$\overline{\text{SBACK2}}$											PDB5D7	PDB6E0	PDB5D6	VSS
L	*N/C*	SBACK1	$\overline{\text{CSROM}}$	$\overline{\text{CSALT}}$											PDB5D3	PDB5D5	PDB5D4	VDD
M	$\overline{\text{B1SOE0}}$	$\overline{\text{B1SOE1}}$	$\overline{\text{LDCLK}}$	$\overline{\text{CSDAC}}$											WR5	PDB5D1	PDB5D0	PDB5D2
N	$\overline{\text{B0SOE0}}$	$\overline{\text{B0SOE1}}$	$\overline{\text{BLANK}}$	SCK1											PDB4C4	PDB4C6	PDB4C5	PDB4C7
P	$\overline{\text{CSYNC}}$	SCK0	$\overline{\text{WE1}}$	$\overline{\text{CAS0}}$											PDB3B7	PDB4C1	PDB4C2	PDB4C3
R	PERF OSC	$\overline{\text{WE0}}$	$\overline{\text{RAS0}}$	$\overline{\text{ID2}}$	OLB0	OLA1	OLE1	WR0	PDB083	PDB192	PDB196	PDB2A1	PDB2A4	WR3	PDB3B3	PDB3B6	WR4	PDB4C0
T	$\overline{\text{OE0}}$	$\overline{\text{ID0}}$	$\overline{\text{ID1}}$	OLC0	OLF0	OLC1	OLG1	PDB081	PDB085	PDB190	PDB194	WR2	PDB2A3	PDB2A2	PDB2A7	PDB3B2	PDB3B4	PDB3B5
U	VDD	VSS	OLA0	OLD0	OLG0	OLB1	OLF1	PDB080	PDB082	PDB084	PDB087	PDB191	PDB195	PDB2A0	PDB2A6	PDB3B1	VDD	VSS
V	VSS	VDD	$\overline{\text{TEST MODE}}$	OLE0	OLH0	OLD1	OLH1	VSS	VDD	VSS	PDB086	WR1	PDB193	PDB197	PDB2A5	PDB3B0	VSS	VDD

Note:
1. *N/C* pins are not connected.

Table 8.9
L64845 SGX
Pin List

Signal	*Pin*	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*	*Pin*
ALTOSC	J4	PDB190	T10	PDB6E5	G17	SBD26	C8	VA5	C18
$\overline{\text{BLANK}}$	N3	PDB191	U12	PDB6E6	J15	SBD27	B7	VA6	C17
$\overline{\text{B0SOE1}}$	N2	PDB192	R10	PDB6E7	F18	SBD28	D8	VA7	C16
$\overline{\text{B0SOE0}}$	N1	PDB193	V13	PDB7F0	F17	SBD29	A6	VA8	D15
$\overline{\text{B1SOE1}}$	M2	PDB194	T11	PDB7F1	H15	SBD30	C7	VDD	A2
$\overline{\text{B1SOE0}}$	M1	PDB195	U13	PDB7F2	E18	SBD31	B6	VDD	A9
$\overline{\text{CAS0}}$	P4	PDB196	R11	PDB7F3	G16	$\overline{\text{SBIRQ5}}$	B5	VDD	A17
$\overline{\text{CAS1}}$	D1	PDB197	V14	PDB7F4	E17	SBPA0	A4	VDD	B1
$\overline{\text{CSYNC}}$	P1	PDB2A0	U14	PDB7F5	G15	SBPA1	B4	VDD	B18
$\overline{\text{CSDAC}}$	M4	PDB2A1	R12	PDB7F6	E16	SBPA2	C4	VDD	J1
$\overline{\text{CSALT}}$	L4	PDB2A2	T14	PDB7F7	F16	SBPA3	D5	VDD	J18
$\overline{\text{CSROM}}$	L3	PDB2A3	T13	PERFOSC	R1	SBPA4	A3	VDD	L18
$\overline{\text{DOTCLK}}$	J3	PDB2A4	R13	$\overline{\text{RAS0}}$	R3	SBPA5	B3	VDD	U1
EXTXFR	K2	PDB2A5	V15	$\overline{\text{RAS1}}$	F4	SBPA6	C3	VDD	U17
$\overline{\text{ID0}}$	T2	PDB2A6	U15	$\overline{\text{SBACK0}}$	K3	SBPA7	D4	VDD	V2
$\overline{\text{ID1}}$	T3	PDB2A7	T15	$\overline{\text{SBACK1}}$	L2	SBPA8	C2	VDD	V9
$\overline{\text{ID2}}$	R4	PDB3B0	V16	$\overline{\text{SBACK2}}$	K4	SBPA9	C1	VDD	V18
$\overline{\text{LDCLK}}$	M3	PDB3B1	U16	$\overline{\text{SBAS}}$	C15	SBPA10	E4	VSS	A8
MAINOSC	H3	PDB3B2	T16	SBCLK	A16	SBPA11	D3	VSS	A10
$\overline{\text{OE0}}$	T1	PDB3B3	R15	SBD0	D14	SBPA12	E3	VSS	A18
$\overline{\text{OE1}}$	D2	PDB3B4	T17	SBD1	B15	SBPA13	E2	VSS	B2
OLA0	U3	PDB3B5	T18	SBD2	A15	SBPA14	F3	VSS	B17
OLA1	R6	PDB3B6	R16	SBD3	D13	SBPA15	E1	VSS	H1
OLB0	R5	PDB3B7	P15	SBD4	C14	SBPA16	G4	VSS	K1
OLB1	U6	PDB4C0	R18	SBD5	B14	SBPA17	F2	VSS	K18
OLC0	T4	PDB4C1	P16	SBD6	C13	SBPA18	G3	VSS	U2
OLC1	T6	PDB4C2	P17	SBD7	A14	SBPA19	F1	VSS	U18
OLD0	U4	PDB4C3	P18	SBD8	D12	SBPA20	H4	VSS	V1
OLD1	V6	PDB4C4	N15	SBD9	B13	SBPA21	G2	VSS	V8
OLE0	V4	PDB4C5	N17	SBD10	C12	SBPA22	G1	VSS	V10
OLE1	R7	PDB4C6	N16	SBD11	A13	SBPA23	H2	VSS	V17
OLF0	T5	PDB4C7	N18	SBD12	D11	SBRD	D6	$\overline{\text{WE0}}$	R2
OLF1	U7	PDB5D0	M17	SBD13	B12	$\overline{\text{SBRESET}}$	J2	$\overline{\text{WE1}}$	P3
OLG0	U5	PDB5D1	M16	SBD14	C11	$\overline{\text{SBSEL}}$	B16	WR0	R8
OLG1	T7	PDB5D2	M18	SBD15	A12	SBSIZ0	D7	WR1	V12
OLH0	V5	PDB5D3	L15	SBD16	D10	SBSIZ1	A5	WR2	T12
OLH1	V7	PDB5D4	L17	SBD17	B11	SBSIZ2	C6	WR3	R14
PDB080	U8	PDB5D5	L16	SBD18	C10	SCK1	N4	WR4	R17
PDB081	T8	PDB5D6	K17	SBD19	A11	SCK0	P2	WR5	M15
PDB082	U9	PDB5D7	K15	SBD20	B10	$\overline{\text{TESTMODE}}$	V3	WR6	J17
PDB083	R9	PDB6E0	K16	SBD21	B9	VA0	F15	WR7	H16
PDB084	U10	PDB6E1	H18	SBD22	C9	VA1	D18	N/C	C5
PDB085	T9	PDB6E2	H17	SBD23	B8	VA2	D17	N/C	L1
PDB086	V11	PDB6E3	G18	SBD24	D9	VA3	D16		
PDB087	U11	PDB6E4	J16	SBD25	A7	VA4	E15		

Figure 8.12
L64845 223-pin CPGA
Mechanical Drawing



Dimension		Inches	Dimension		Inches
A	Min	0.067	E1	Max	0.910
	Nom	0.079	E2	Max	0.870
	Max	0.091	E3	Ref	1.800
A1	Max	0.131	G	Min	0.050
A2	Min	0.028		Nom	0.065
	Nom	0.031		Max	0.080
	Max	0.049	H	Max	0.100
b	Min	0.016	I	Max	0.140
	Nom	0.018	J	Ref	0.057
	Max	0.020	L	Min	0.189
b1	Min	0.042		Nom	0.197
	Nom	0.050		Max	0.205
	Max	0.058	L1	Min	0.042
b2	Ref	0.008		Nom	0.050
D	Min	1.956		Max	0.058
	Nom	1.960	M	Ref	1.80
	Max	1.964	M1	Ref	0.80
D1	Max	0.910	N	Ref	1.70
D2	Max	0.870	N1	Ref	0.85
D3	Ref	1.800	Q1	Min	0.017
e	Typ	0.100			
E	Min	1.956			
	Nom	1.960			
	Max	1.964			

Note:
1. Drawing is not to scale.
2. Size of array is 19 x 19. Corner detail and index mark are LSI Logic Corporation options.
3. Total number of pins is 279.
4. Envelope for maximum lid.
5. Envelope for integral heat slug.
6. For board layout and manufacturing, you may obtain engineering drawings from your LSI Logic Products marketing representative by requesting the outline drawing for package code NZ.

MD92.NZ

Appendix A
Schematics

This appendix shows the wiring for the SGX card.

Figure A.1 shows a block diagram with signal names.

The following figures show the schematics:

- Figure A.2 – Bus Interface/Frame Buffer Control
- Figure A.3 – Video Memory
- Figure A.4 – Video/Monitor Control
- Figure A.5 – Sbus Interface

Figure A.1
SGX Card Signal
Block Diagram

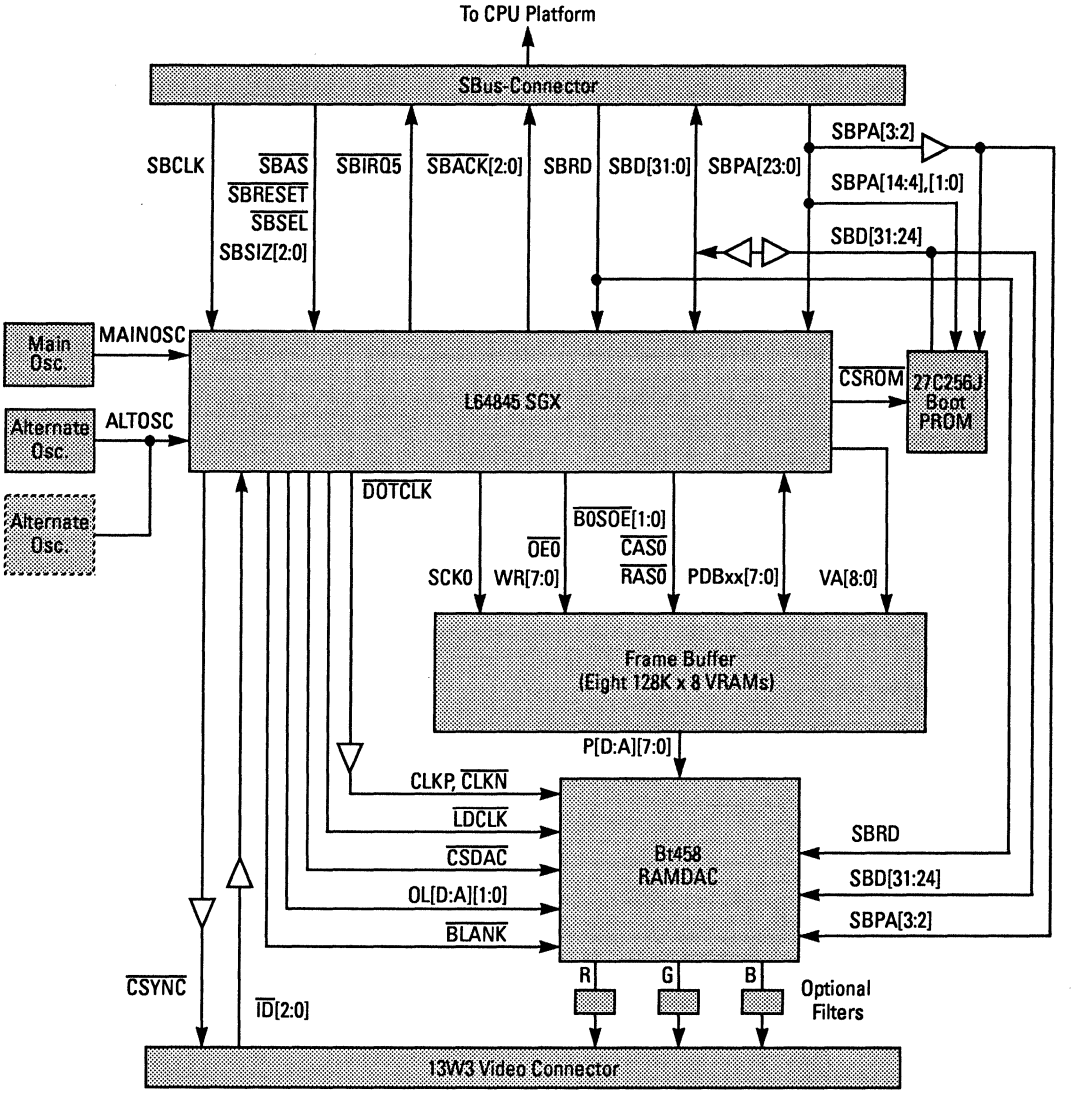
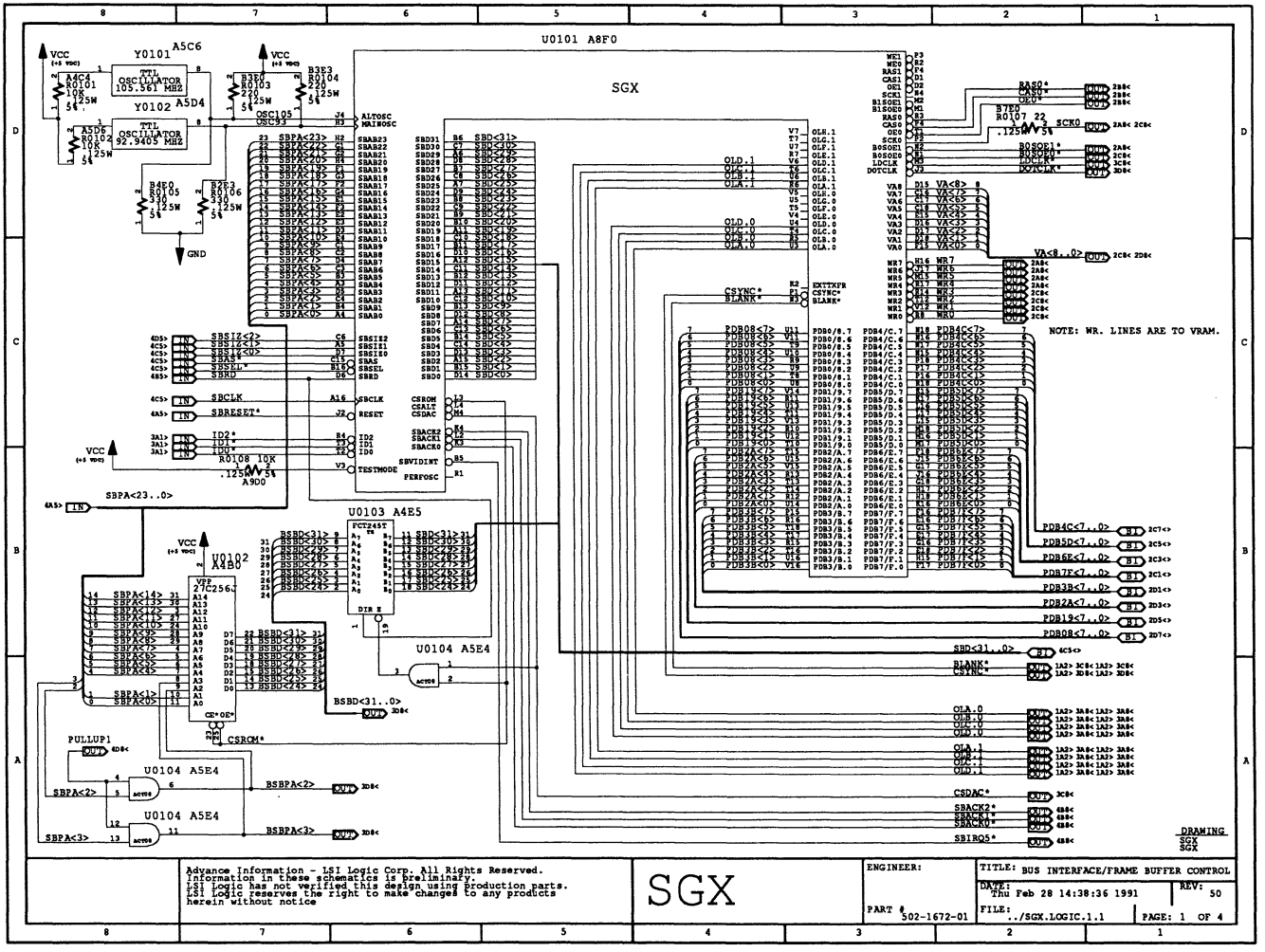


Figure A.2
Bus Interface/Frame
Buffer Control
Schematic



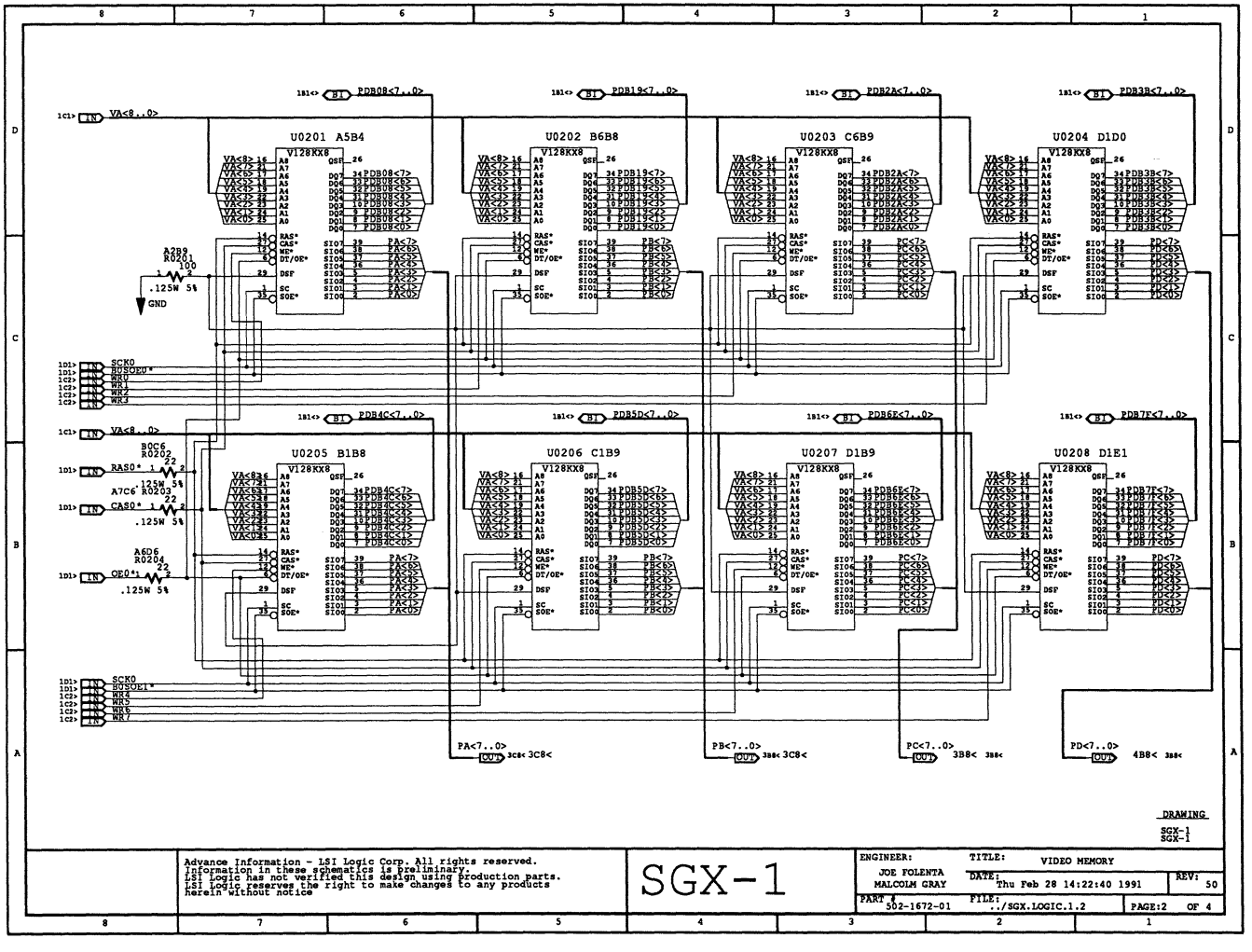
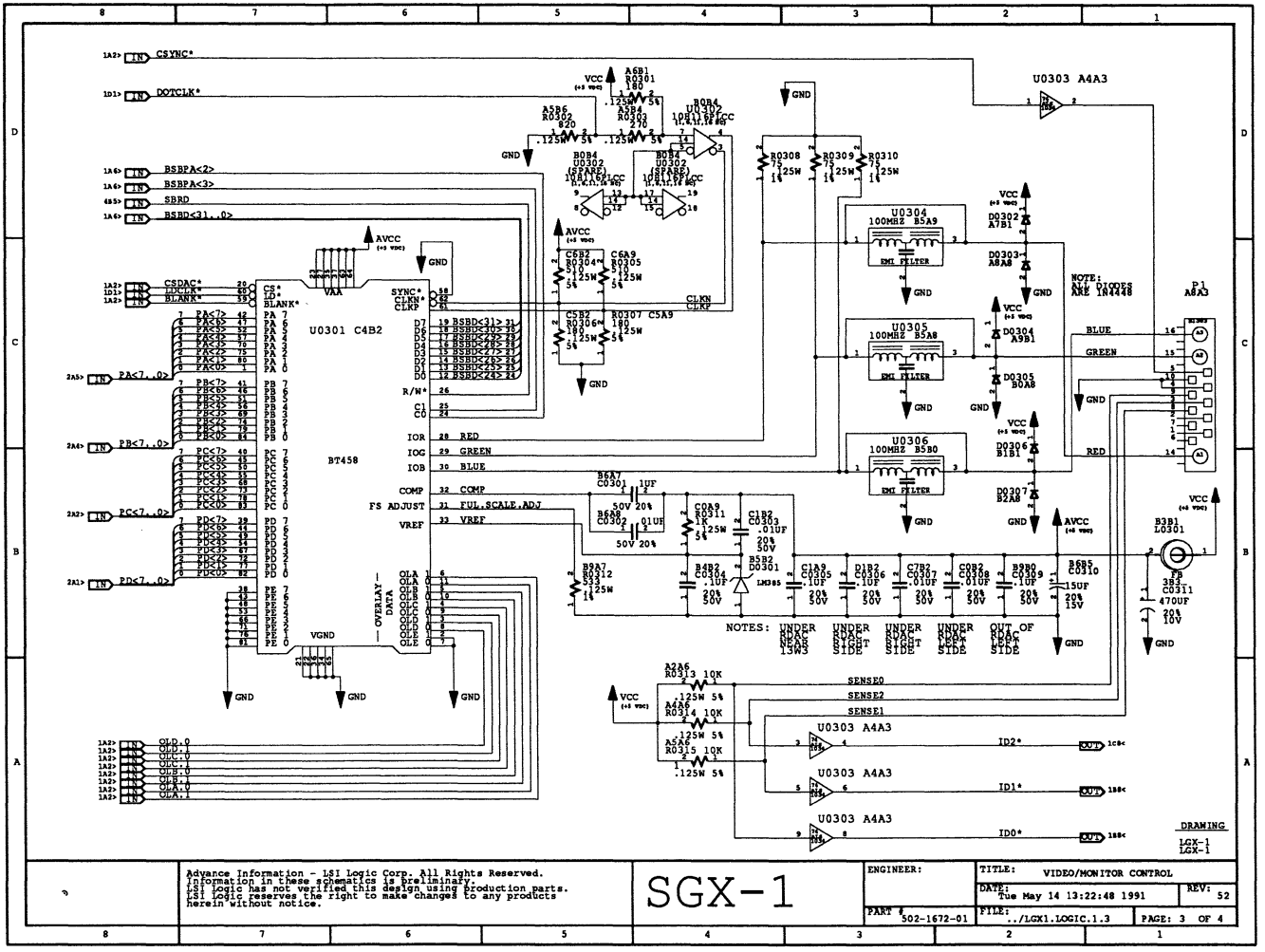
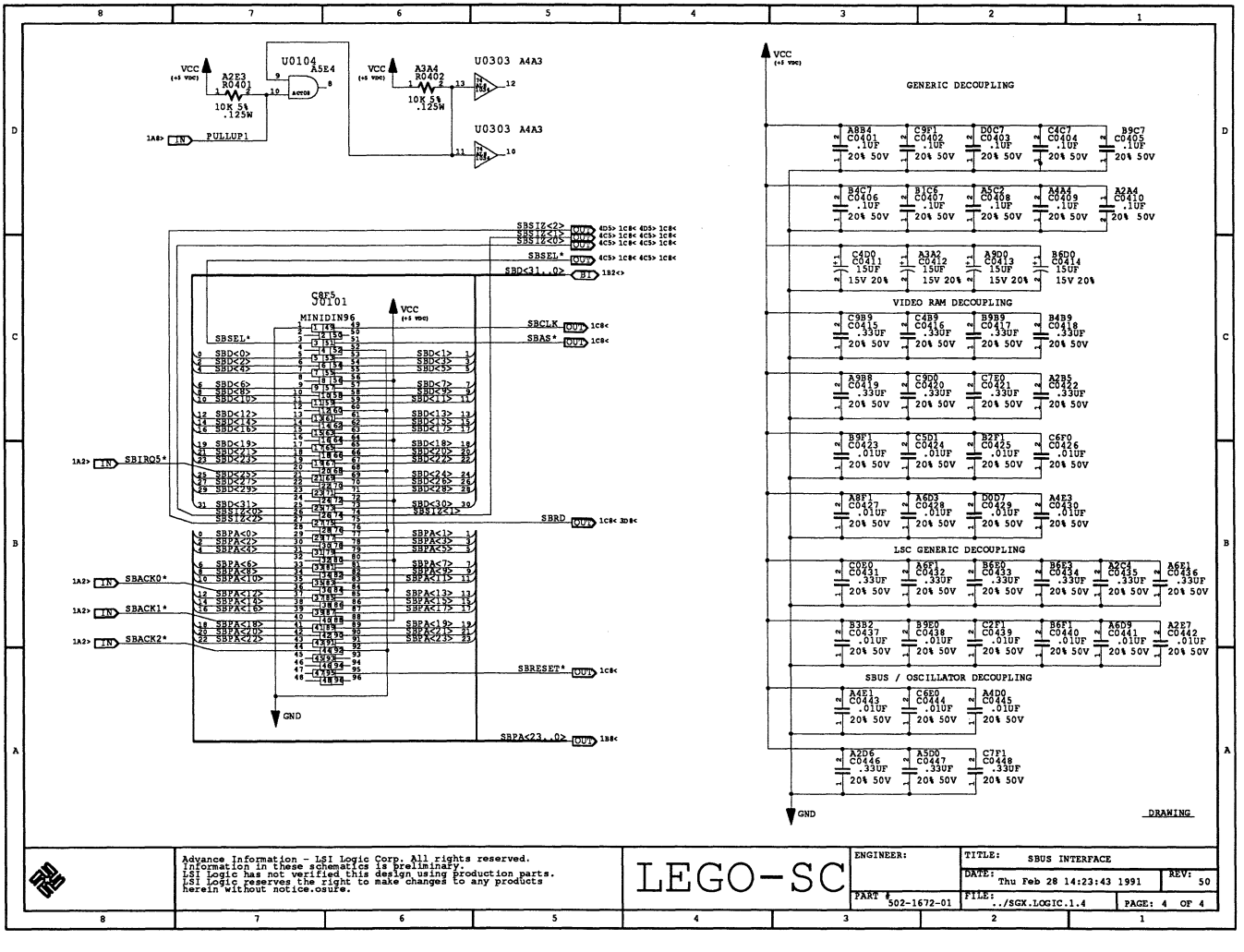


Figure A.3 Video Memory Schematics

Figure A.4
Video/Monitor
Control Schematic





LEGO-SC

ENGINEER:	
PART #:	502-1672-01
TITLE:	SBUS INTERFACE
DATE:	Thu Feb 28 14:23:43 1991
REV:	50
FILE:	../SGX.LOGIC.1.4
PAGE:	4 OF 4

Figure A.5
SBus Interface
Schematics

Appendix B
SGX Card

This appendix describes the SGX card. It is divided into two sections:

- Section B.1, **Overview**, describes the SGX card, shows its layout and lists supporting hardware platforms.
- Section B.2, **SGX Card Specification Summary**, describes the physical specifications for the SGX card. This section also contains a block diagram of the video memory architecture.

B.1 Overview

LSI Logic provides a Manufacturer's kit for an SGX SBus Graphics Accelerator card. The card is a single-wide (146.7 mm long by 83.82 mm wide) SBus expansion card that conforms to Sun's single-wide Sbus card specification and can be used either with the 66 MHz or the 76 MHz Sun color monitor. It is 100% software backward compatible with the SPARCstation GX graphics accelerator cards from Sun Microsystems.

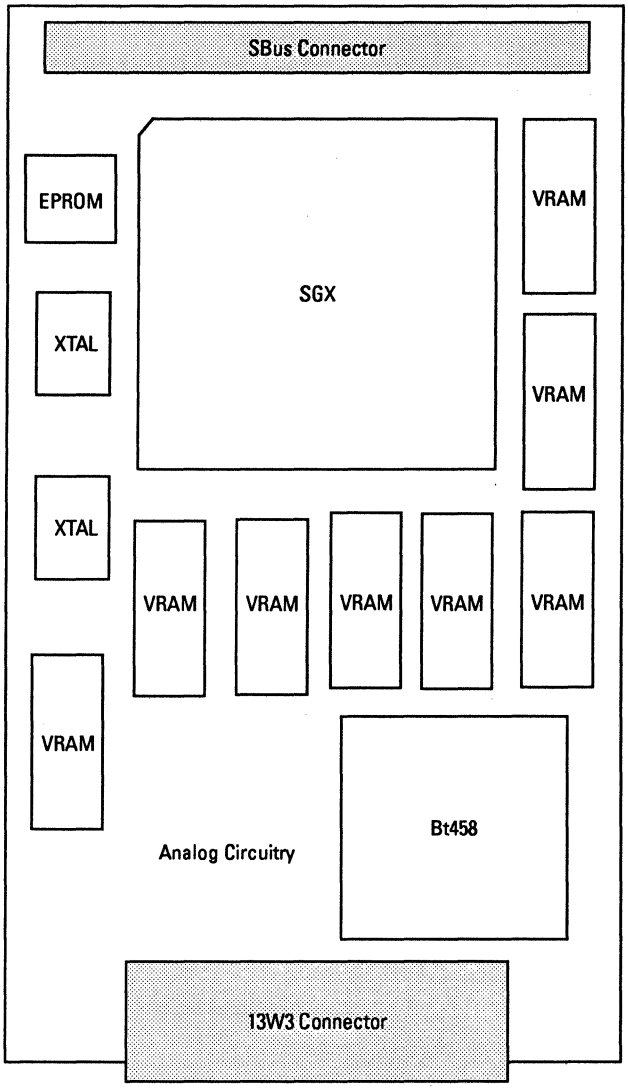
Figure B.1 shows the SGX card layout.

The L64845 SGX Sbus Graphics Accelerator Manufacturing Kit contains:

- Schematics
- Gerber files
- Drill and aperture information
- Assembly and fabrication drawings
- Bill of materials
- EPROM Forth code

Order Code: KT-999-MFGGX1

Figure B.1
SGX Card Layout



The SGX is supported on the following SBus based SPARCstations and SPARCsystems:

- SPARCstation 1
- SPARCstation 1+
- SPARCstation 2
- SPARCstation IPC
- SPARCstation IPX
- LSI SparKIT

Refer to the *GX Installation Guide* for SGX configuration options.

B.2 SGX Card Specification Summary

This section describes the physical specifications for the SGX card and shows a block diagram of the video memory architecture.

Physical Specifications

Single slot board.

Length: 146.70 mm.

Width: 83.82 mm.

Weight: 5.7 oz.

Electrical Specifications

Current: 1 amp (typical), 1.5 amps (worst case)

Power Requirements: +5 volts ±5%

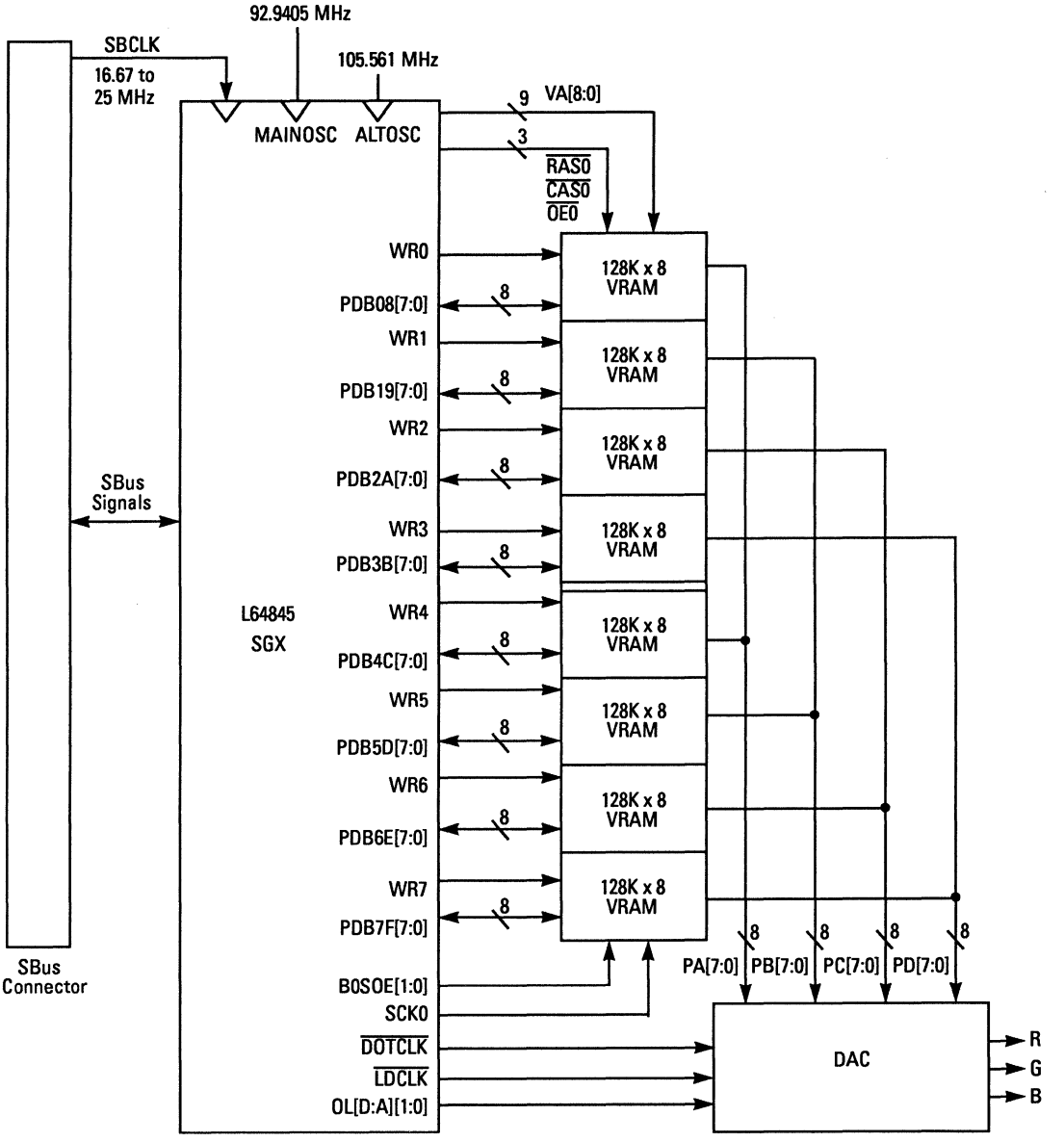
Heat dissipation: 5 watts (approximate)

Load Condition: Running special code designed for exercising the frame buffer to the maximum power (not typical of normal applications).

Environmental Specifications	Temperature range: Between 32 and 104 degrees Fahrenheit (0 and 40 degrees Celsius). Humidity: Between 5% and 80% (relative noncondensing) Altitude: Between 0 and 10,000 feet (0 and 3048 meters)
Reliability	MTBF: 163,000 hours MTTR: 30 minutes
Performance	Assuming 23 MHz (92.9405 MHz/4) clock and 1152 x 900 resolution: Horizontal fill: 120 Mpixels per second Vertical fill: 4.8 Mpixels per second Vector and Polygon Performance: Depends on the CPU platform and software efficiency.
DAC	Bt458 compatible.
Resolution	1152 x 900 @66 Hz 1152 x 900 @76 Hz
Interface Requirements	Connector type: SBus
Agency Certification	FCC Class B DOC Class B VCCI Class 2 UL CSA TUV

Compatibility	SBus compatible.
Memory Devices	Uses eight 128k x 8 100 ns VRAMs. Refer to Chapter 5, "Memory Interfaces," for additional information.
Component Packaging	The SGX IC is packaged in a ceramic 223-pin grid array package. Two alphanumeric pin identification schemes have been used on the card. ■ Pins are numbered from 2 through 224 on the 18 x 18 grid. There is no pin 1 on the chip or on the socket. Pin 1 is intentionally omitted to eliminate any possibility of misalignment. Pin 2 is the leftmost pin in the corner on the top row (near the SBus connector) on the component side. ■ Pins can be identified with their relationship to the alphanumeric rows 1 to 18 and A to V.
Video Timing Diagrams	See Chapter 6, "Video Interface."
Video Memory Architecture	1MB consisting of eight 128K x 8 VRAMs. Figure B.2 shows the SGX video memory architecture.

Figure B.2
SGX Video Memory Architecture



Appendix C
Bt458 RAMDAC

This appendix describes the Bt548 RAMDAC. It is divided into five sections:

- Section C.1, **General Description**, briefly describes the Bt458 RAMDAC.
- Section C.2, **Functional Description**, describes the functionality of the Bt458 RAMDAC with specific reference to its use on the SGX card.
- Section C.3, **Logic Symbol**, shows the Bt458 RAMDAC's logic symbol.
- Section C.4, **Signal Descriptions**, describes the Bt458 RAMDAC interface signals.
- Section C.5, **Programming the RAMDAC**, describes how to write to the RAMDAC registers.

See the *Brooktree Product Databook* for more information on the Bt458.

Chapter 6, "Video Interface," describes the DAC interface to the RAMDAC.

C.1 General Description

The Bt458 RAMDAC, made by Brooktree Corporation, is a palette digital-to-analog converter specifically designed for high performance and high resolution color graphics. It contains an input buffer (pixel data registers) and a 256 x 24 color lookup table (CLUT) with three 8-bit video D/A converters. The video outputs generate RS–343A compatible RGB video signals that drive doubly-terminated 75-ohm coaxial cable. EMI low-pass filters may be used on the outputs. The RAMDAC contains an input multiplexer, blink and read mask logic, a 4 x 24 overlay table, and CPU interface logic.

C.2 Functional Description

This section describes the functionality of the Bt458 RAMDAC with particular reference to video and frame buffer interfaces, internal multiplexing, color selection, and I/O signals as used on the SGX card.

The RAMDAC loads four pixels from the VRAM into the input buffer over pixel data lines PA[7:0] through PD[7:0] on the rising edge of the 23.24 MHz (43 nS) load clock ($\overline{\text{LDCLK}}$). PE[7:0] is connected to ground. Then it serially reads these pixels out of the input buffer through a four-to-one multiplexer using the 92.9405 MHz pixel clock (CLKP). The CLUT maps each of the pixels in the serial stream to a physical color. The three DACs process the digital output of the CLUT to create an RS-343A compatible RGB analog output.

The Bt458 can simultaneously display 256 (2^8) colors out of an available set of 16.8 million (2^{24}) colors. Additional features include programmable blink rates, bit plane masking, blinking, color overlay capability, and a dual-port color palette RAM.

CPU Interface

The CPU accesses the CLUT and the internal registers of the RAMDAC via the CPU interface. Table C.1 shows how SBPA[3:2] determines the access type.

Table C.1
CPU Access Type

SBPA3	*SBPA2*	*Type of Access*
0	0	Address Register AR[7:0]
0	1	Look-Up Table
1	0	Control Registers
1	1	Overlay Registers

An 8-bit Address Register, AR[7:0] generates addresses for the Color Memory locations and the Control Registers. This register increments at the end of each third (blue) access, when SBPA2 = 1, so that the CPU does not have to rewrite it. The Address Register operates as shown in Table C.2.

Table C.2
Address Register

AR[7:0]	*SBPA3*	*SBPA2*	*Location or Register Addresses*
0x00-0xFF	0	1	CLUT location 00 through FF
0x00	1	1	Overlay Register 0
0x01	1	1	Overlay Register 1
0x02	1	1	Overlay Register 2
0x03	1	1	Overlay Register 3
0x04	1	0	Read Mask Register
0x05	1	0	Blink Mask Register
0x06	1	0	Command Register
0x07	1	0	Test Register

- A modulo 3 counter (ARb, ARa) controls which color byte of the 24-bit Color Memory word the RAMDAC accesses. (The CPU cannot access this counter.) It increments at the end of each CPU access with SBPA2 = 1 (signifying a color operation) and is reset to zero at the end of each CPU access with SBPA2 = 0 (signifying a control register operation). This counter controls color as shown in Table C.3.

Table C.3
Color Byte Access

ARb	*ARa*	*Color Byte Being Accessed*
0	0	Red
0	1	Green
1	0	Blue

Frame Buffer (VRAM) Interface

The frame buffer (VRAM) sends pixel data to the RAMDAC input buffer. The rising edge of the 23.24 MHz clock ($\overline{\text{LDCLK}}$) loads the following signals into the input buffer:

- Pixel data inputs P[D:A][7:0] from the VRAM
- The eight overlay data inputs, OL[D:A][1:0], from the SGX
- $\overline{\text{BLANK}}$ from the SGX

Pixel data inputs, PE[7:0], overlay data inputs, OLE[1:0], and combined sync signal, $\overline{\text{SYNC}}$, are all connected to ground on the SGX card.

The 92.9405 MHz pixel clock (CLKP) gates the four pixel bytes out of the Input Multiplexer in series, thus providing a four-to-one multiplexing of the pixel bytes. (The pixel clock frequency is four times that of the load clock.) The multiplexer outputs pixel PA[7:0] first, then PB[7:0], PC[7:0], and PD[7:0]. It then repeats this cycle with the next four bytes loaded from the VRAM.

Note that the eight overlay data signals are also multiplexed four-to-one. Thus the multiplexer outputs OLAx first, then OLBx, OLCx, and OLDx.

When $\overline{\text{LDCLK}}$ and $\overline{\text{BLANK}}$ rise, the color and overlay data for four consecutive pixels are latched into the RAMDAC via the pixel select inputs, P[D:A][7:0]. $\overline{\text{LDCLK}}$ is used to clock external circuitry and to generate the basic video timing. $\overline{\text{SYNC}}$ is connected to ground.

Internal Multiplexing

$\overline{\text{LDCLK}}$ may be phase shifted from CLKP by any amount to simplify the frame buffer interface timing. On the SGX card, $\overline{\text{LDCLK}}$ is derived externally by dividing CLKP by four. This should be independent of the propagation delays of the $\overline{\text{LDCLK}}$ generation logic. As a result, the pixel and overlay data are latched on the rising time of $\overline{\text{LDCLK}}$, independent of the phase of CLKP.

Internal logic maintains an internal load signal, synchronous to CLKP, that is guaranteed to follow $\overline{\text{LDCLK}}$ by at least one, but not more than four clock cycles. This load signal transfers the latched pixel and overlay data into a second set of latches which are then internally multiplexed at the clock rate (up to 125 MHz).

Since 4:1 multiplexing is specified in the SGX card implementation, only one rising edge of $\overline{\text{LDCLK}}$ should occur every four clock cycles. Otherwise, the internal load generation circuitry assumes that it is not locked onto $\overline{\text{LDCLK}}$ and continuously attempts to resynchronize itself.

Color Selection

On the rising edge of the clock cycle, the Read Mask Register, the Blink Mask Register, and the Command Register process eight bits of color information, Px[7:0], and two bits of overlay information, OLx[1:0], for each pixel x. The RAMDAC uses this information to select an entry in the CLUT or an Overlay Register. Display of individual bit planes may be enabled or disabled via the control registers. Individual bit planes may also be blinked at one of four blink rates and duty cycles.

The eight-bit Read Mask Register selectively enables or disables bit planes from being presented to the CLUT. The eight-bit Blink Mask Register selectively enables or disables blinking of a bit plane. Bits four and five of the Command Register determine the blink-rate duty cycle. The RAMDAC increments the counter that generates the internal blink clock

during the vertical retrace intervals. These intervals are detected when $\overline{\text{BLANK}}$ is low for a period of at least 256 load clock cycles.

Table C.5 shows how to select the color palette or overlay.

Table C.4 Color Palette and Overlay Selection

Bit 6[1]	*OLx1*	*OLx0*	*Px[7:0]*	*Palette Entry Addressed*
1	0	0	0x00	Color Palette Entry $00
1	0	0	0x01	Color Palette Entry $01
:	:	:	:	:
1	0	0	0xFF	Color Palette Entry $FF
0	0	0	0xxx	Overlay Color 0
x	0	1	0xxx	Overlay Color 1
x	1	0	0xxx	Overlay Color 2
x	1	1	$0xxx	Overlay Color 3

1. Bit 6 of Command Register.

Overlay Color Selection

The two overlay bits in the four 24-bit Overlay Registers allow selection of four different overlay colors. Bits zero and one of the Command Register act as a read mask, and bits two and three act as a blink mask on the overlay inputs. Thus an overlay output (OVLADDR0 or OVLADDR1) is present only when the corresponding read mask bit and blink mask bit are true. When the blink mask is enabled, an internal blink clock turns the overlay output off and on.

Table C.6 shows how to use the overlay outputs to select the Overlay Register. Note that when OVLADDR1 = 0 and OVLADDR0 = 0, bit six of the Command Register selects between Overlay Register 0 and the CLUT output addressed by the eight pixel bits.

Table C.5 Overlay Register Selection

OVLADDR1	*OVLADDR0*	*Bit 6[1] = 0*	*Bit 6[1] = 1*
0	0	Overlay Register 0	CLUT
0	1	Overlay Register 1	Overlay Register 1
1	0	Overlay Register 2	Overlay Register 2
1	1	Overlay Register 3	Overlay Register 3

1. Bit 6 of Command Register.

The OL[D:A][1:0] overlay select inputs are latched on the rising edge of $\overline{\text{LDCLK}}$. On the SGX card, the OLE[1:0] inputs are connected to ground. The overlay select inputs, in conjunction with bit six of the Command Register, specify which palette is to be used for color information. See Table C.7 for details.

Table C.6
Palette Selection

OLx1	*OLx0*	*Bit 6*[1]	*Palette*
0	0	1	CLUT
0	1	1	Overlay Color 1
1	0	1	Overlay Color 2
1	1	1	Overlay Color 3

1. Bit 6 of Command Register.

Video Generation

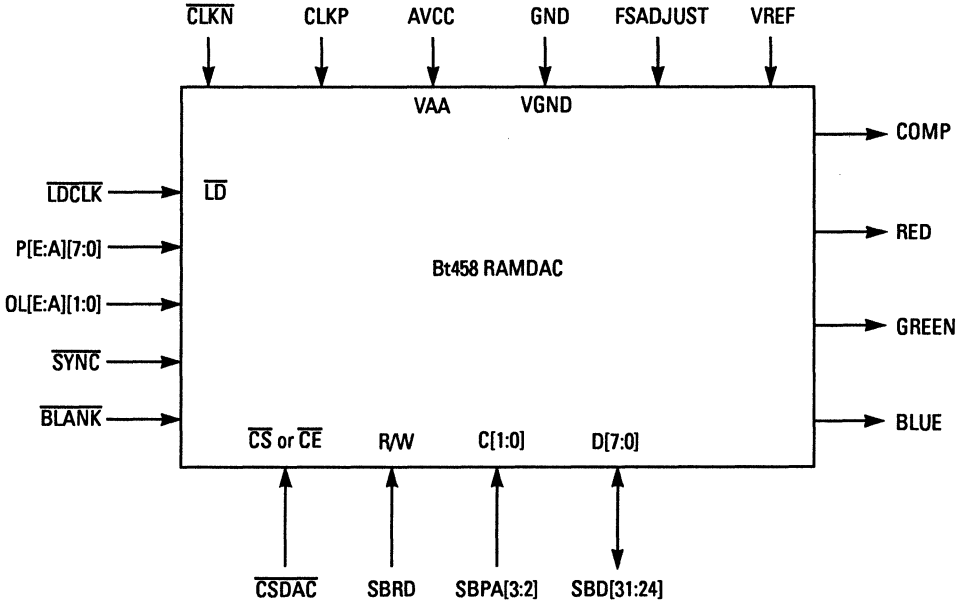
During each pixel clock cycle, the RAMDAC sends 24 bits of information from either the CLUT or an Overlay Register to the three 8-bit DACs. These DACs convert the color memory digital output into RGB RS-343A analog signals for the red, green, and blue guns of the monitor.

On the SGX card, $\overline{\text{BLANK}}$ is routed to all three DACs through a pipeline delay circuit that delays this signal the same amount that the video stream is delayed. $\overline{\text{SYNC}}$ (connected to ground) goes only to the Green DAC.

C.3 Logic Symbol

Figure C.1 shows the logic symbol for the Bt458 RAMDAC. In the figure, a bar over the signal name indicates the signal is active LOW. Arrows indicate whether the signal is an input or an output.

Figure C.1
Bt458 Logic Symbol



C.4 Signal Descriptions

This section describes the interface signals of the Bt458 RAMDAC chip. The signal input or output direction is relative to the Bt458 chip, not to the SGX chip.

AVCC — **Analog Power** — **Input**
The VAA inputs with this signal are connected to +5 VDC.

$\overline{\text{BLANK}}$ — **Video Blank** — **Input**
The video blank input connects to the composite video blank output from the SGX chip. A logic zero drives the analog outputs to the blanking level. It is latched on the rising edge of $\overline{\text{LDCLK}}$. When $\overline{\text{BLANK}}$ is a logical zero, the pixel and overlay inputs are ignored. It overrides the color pixel and overlay data to force the red, green, and blue video output signals to their blanking levels. Blanking is required during the monitor's vertical and horizontal retrace period.

BLUE — **Blue Video** — **Output**
The blue video current output is a high-impedance current source capable of directly driving a doubly-terminated 75-ohm coaxial cable.

CLKP, $\overline{\text{CLKN}}$ — **Differential Clock Inputs** — **Input**
The differential clock inputs are generated from the SGX output signal, $\overline{\text{DOTCLK}}$. These differential clock inputs are designed to be driven by ECL logic configured for single supply (+5V) operation. This clock rate is the pixel clock rate of the system.

COMP — **Compensation Pin** — **Output**
The compensation pin provides compensation for the internal reference amplifier. A 0.01 µF ceramic capacitor must be connected between this pin and VAA.

$\overline{\text{CSDAC}}$ — **Chip Select Palette** — **Input**
This signal, from the SGX chip, connects to the RAMDAC chip enable control input ($\overline{\text{CS}}$ in the schematics, $\overline{\text{CE}}$ in the *Brooktree Product Databook*). This chip-select signal enables the CPU Interface logic. Data on the Local Data bus SBD[31:24] is internally latched on the rising edge of $\overline{\text{CSDAC}}$ during writes. This input must be a logical zero to enable data to be written to or read from the RAMDAC.

FSADJUST **Full-Scale Adjust Control** **Input**

This signal provides full-scale adjust control. A resistor (RSET) connected between this pin and GND controls the magnitude of the full scale video signal. Using the oscilloscope, the voltage at this pin should be approximately equal to the voltage at the VREF pin. The relationship between RSET and the full scale output current on GREEN is:

```
RSET (ohms) = 11,294 * VREF (V) / GREEN (mA)
```

The full scale output current on RED and BLUE for a given RSET is:

```
RED,BLUE (mA) = 8067 * VREF (V) / RSET (ohms)
```

GND **Analog Ground** **Input**

All inputs with this signal are connect to ground.

GREEN **Green Video** **Output**

The green video current output is a high-impedance current source capable of directly driving a doubly-terminated 75-ohm coaxial cable.

$\overline{\text{LDCLK}}$ **Load Control Input** **Input**

The load clock output from the SGX chip, connects to the RAMDAC load control input, $\overline{\text{LD}}$. This signal is one-fourth the rate of the pixel clock frequency, CLKP. On the SGX card P[D:A][7:0], OL[D:A][1:0], and $\overline{\text{BLANK}}$ are latched on the rising edge of $\overline{\text{LDCLK}}$ while $\overline{\text{SYNC}}$, PE[7:0] and OLE[1:0] are connected to ground.

OL[E:A][1:0] **Overlay Select** **Input**

The overlay select inputs connect to the SGX overlay plane outputs. They are latched on the rising edge of $\overline{\text{LDCLK}}$, and in conjunction with bit 6 of the Command Register, specify which palette is to be used for color information. On the SGX card, OL[D:A][1:0] are used for the cursor, and OLE[1:0] is connected to ground.

P[E:A][7:0] **Pixel Select** **Input**

The pixel select inputs connect to the pixel outputs from the VRAM. These inputs are used to specify, on a pixel basis, which one of the 256 entries in the color palette RAM provides color information. On the SGX card, four consecutive pixels (eight bits per pixel) are input through this port. Pixel data are latched on the rising edge of $\overline{\text{LDCLK}}$. Note that PA[7:0] is output first, followed by PB[7:0], and so on, until all four pixels have been output. Then

the cycle repeats. On the SGX card, P[D:A][7:0] are used for pixel data, and PE[7:0] is connected to ground.

RED — **Red Video** — **Output**

The red video current output is a high-impedance current source capable of directly driving a doubly-terminated 75-ohm coaxial cable.

SBD[31:24] — **Data Bus** — **Bidirectional**

The data bus signals connect to the RAMDAC data bus, D[7:0]. Data is transferred between the SBus and the RAMDAC over this eight bit bidirectional data bus. D0 is the least significant bit.

SBPA[3:2] — **Command Control** — **Input**

These signals, from the Sbus physical address lines, drive the RAMDAC command control inputs C[1:0]. They specify the type of read or write being performed and are latched on the falling edge of $\overline{\text{CSDAC}}$.

SBRD — **Read/Write** — **Input**

This signal, the Sbus Read/Write, drives the RAMDAC R/W input. This signal is the read/write control input. To write to the RAMDAC, both $\overline{\text{CSDAC}}$ and SBRD must be a logical zero. To read from the RAMDAC, $\overline{\text{CSDAC}}$ must be a logical zero and SBRD must be a logical one. SBRD is latched on the falling edge of $\overline{\text{CSDAC}}$.

$\overline{\text{SYNC}}$ — **Synchronization Control** — **Input**

This signal is the composite sync control input.

On the SGX card, this input is connected to ground. This connection (logical zero) switches off a current source on the G output. (Note that the SGX signal, $\overline{\text{CSYNC}}$, is routed to the video connector.)

VREF — **Voltage Reference** — **Input**

An external reference circuit must supply the voltage reference input with a 1.235V (typical) reference. A 0.01 µF ceramic capacitor must be used to decouple this input to VAA. The use of a resistor network to generate the reference is not recommended as any low frequency power supply noise on VREF will be directly coupled onto the analog outputs.

C.5 Programming the RAMDAC

The user can access and program the following RAMDAC registers: Address, Command, Read Mask, Blink Mask, and Test.

Address Register

When SBPA3 and SBPA2 are both zero the Bt458 accesses the Address Register. If SBRD is low, local data bus lines SBD[31:24] are loaded into the Address Register. If SBRD is high, the contents of the Address Register are loaded onto local data bus lines SBD[31:24].

Command Register

The Command Register is an eight-bit register located at address 0x06. Table C.8 describes this register.

Table C.7
Command Register

Bit	*Name*	*Definition*	*Description*
7	Multiplex Select	0 = 4:1 Multiplexing 1 = 5:1 Multiplexing	On the SGX card, this bit is set to zero for 4:1 multiplexing pixel and overlay data.
6	RAM Enable	0 = Use Overlay Color Zero 1 = Use Color Palette RAM	When the overlay select bits are zero, this bit specifies whether to use the color palette RAM or overlay color zero to provide color information. The default for this bit is one.
5,4	Blink Rate Selection	00 = 1.024 Seconds (75/25) 01= 0.512 Seconds (50/50) 10= 1.024 Seconds (50/50) 11 = 2.048 Seconds (50/50)	These two bits specify the blink rate cycle time and duty cycle of the internal blink clock (assuming a 60 Hz non-interlaced or 30 Hz interlaced refresh rate). For applications having other refresh rates (i.e., 57 Hz non-interlaced) the blink rate is determined by scaling the specified values (i.e., multiplying by 57/60). The numbers in parentheses specify the duty cycle (% on/off). The default for these bits is zero.
3	OLx1 Blink Enable	0 = Disable Blinking 1 = Enable Blinking	When set to one, this bit forces the OLx1 inputs to toggle between logical zero and the input at the blink rate selected prior to palette selection. A value of logical zero does not affect the value of OLx1 inputs. To make the overlay bit plane one blink, bit one of the Command Register must be set to a logical one. The default for this bit is zero.
2	OLx0 Blink Enable	0 = Disable Blinking 1 = Enable Blinking	When set to one, this bit forces the OLx0 inputs to toggle between logical zero and the input at the blink rate selected prior to palette selection. A value of logical zero does not affect value of the OLx0 inputs. To make the overlay bit plane zero blink, bit zero of the Command Register must be set to a logical one. The default for this bit is zero.

Table C.7 (Continued)
Command Register

Bit	*Name*	*Definition*	*Description*
1	OLx1 Display Enable	0 = Disable 1 = Enable	When set to zero, this bit forces the OLx1 inputs to a logical zero before selecting the palettes. When set to one, it does not affect the value of the OLx1 inputs. The default for this bit is one.
0	OLx0 Display Enable	0 = Disable 1 = Enable	When set to zero, this bit forces the OLx0 inputs to a logical zero before selecting the palettes. When set to one, it does not affect the value of the OLx0 inputs. The default for this bit is one.

Read Mask Register

The Read Mask Register is an eight-bit register located at address 0x04. It enables (logical one) or disables (logical zero) pixel bit planes from addressing the CLUT. Bit zero masks inputs PA0 through PD0, bit one masks inputs PA1 through PD1, and so on. Overlay plane masking is controlled by bits one and zero of the Command Register as shown in Table C.8.

Blink Mask Register

The Blink Mask Register is an eight-bit register located at address 0x05. It enables (logical one) or disables (logical zero) pixel bit planes from blinking. Bit zero masks inputs PA0 through PD0, bit one masks inputs PA1 through PD1, and so on. When enabled, the bit plane toggles between its original value and logical zero at the selected rate and duty cycle. Overlay plane masking is controlled by bits two and three of the Command Register as shown in Table C.8.

Test Register

The Test Register is an eight-bit register located at address 0x07. It can read (back) pixel or overlay data presented to the DACs from the Color Memory. Bits seven through four contain color information and bits three through zero contain control information as described in Table C.9.

Table C.8
RAMDAC Test Register

Test Register Bit	*Function*	
7–4	Color nibble (4 bits of red, green, or blue)	
3	0 = Select high nibble	1 = Select low nibble
2	0 = Disable blue data	1 = Enable blue data
1	0 = Disable green data	1 = Enable green data
0	0 = Disable red data	1 = Enable red data

Bits seven through four are defined only when exactly one of bits two through zero is enabled. When unused, this register should be initialized to 0x00.

Appendix D
Customer Feedback

We would appreciate your feedback on this document. Please copy the following page, add your comments, and fax it to us at:

LSI Logic Corporation
Microprocessor Publications
M/S G-812
Fax 408.433.8989

If appropriate, please also fax copies of any marked-up pages from this document.

IMPORTANT: Please include your name, phone number, FAX number, and company address so that we may contact you directly for clarification or additional information. Thank you for your help in improving the quality of our documents.

Reader's Comments

FAX your comments to:

LSI Logic Corporation
Microprocessor Publications
M/S G-812
Fax 408.433.8989

Does the publication meet your needs? Yes__ No__

Is the material:

- Complete? Yes__ No__
- Easy to use? Yes__ No__
- Written for the appropriate technical level? Yes__ No__

What could we do to improve this document?

_____

_____

_____

_____

If you found errors in this document, please specify the error and page number. If appropriate, please fax a marked-up copy of the page(s).

_____

_____

_____

Please complete the information below.

Name _____ Date _____

Telephone _____ FAX _____

Title _____

Department _____ Mail Stop _____

Company Name _____

Street _____

City _____

State _____ Zip _____

Your feedback is important to us. Thank you for your comments.

Index

Sales Offices and Design Resource Centers

LSI Logic Corporation
U.S. Headquarters
■ **1551 McCarthy Blvd**
Milpitas CA 95035
Tel: 408.433.8000
Telex: 171092
Fax: 408.434.6457

Alabama
■ 600 Boulevard South
Suite 104J
Huntsville AL 35802
Tel: 205.883.3527
Fax: 205.883.3529

Arizona
8283 N Hayden Road
Suite 270
Scottsdale AZ 85258
Tel: 602.951.4560
Fax: 602.951.4580

California
■ 2540 N First Street
Suite 201
San Jose CA 95131
Tel: 408.954.1561
Fax: 408.954.1565

■ Two Park Plaza
Suite 1000
Irvine CA 92714
Tel: 714.553.5600
Fax: 714.474.8101

10731 Treena Street
Suite 209
San Diego CA 92131
Tel: 619.689.7140
Fax: 619.689.7145

■ 15821 Ventura Blvd
Suite 275
Encino CA 91436
Tel: 818.379.2400
Fax: 818.783.5548

Colorado
3333 Quebec Street
Suite 6014
Denver CO 80207
Tel: 303.399.1112
Fax: 303.399.6558

Florida
1333 Gateway Drive
Suite 1019C
Melbourne FL 32901
Tel: 407.728.9481
Fax: 407.728.9587

■ 1900 Glades Road
Suite 201
Boca Raton FL 33431
Tel: 407.395.6200
Fax: 407.394.2865

Illinois
■ 1450 American Lane
Suite 1610
Schaumburg IL 60173
Tel: 708.995.1600
Fax: 708.995.1622

Maryland
■ 6903 Rockledge Dr
Suite 230
Bethesda MD 20817
Tel: 410.897.5800
Fax: 410.897.8389

10480 Little Patuxent Pkwy
Suite 500
Columbia MD 21044-3502
Tel: 301.740.5664
Fax: 301.740.2048

Massachusetts
■ 9 Hillside Avenue
Prospect Place
Waltham MA 02154
Tel: 617.890.0180
Fax: 617.890.6158

Minnesota
■ 8300 Norman Center Drive
Suite 730
Minneapolis MN 55437
Tel: 612.921.8300
Fax: 612.921.8399

New Jersey
■ 379 Thornall Street
Edison NJ 08837
Tel: 908.549.4500
Fax: 908.549.4802

New York
1065 Route 82
Hopewell Junction
New York NY 12533
Tel: 914.226.1620
Fax: 914.226.1315

Eastview Park
7979 Victor-Pittsford Road
Victor NY 14564
Tel: 716.223.8820
Fax: 716.223.8822

North Carolina
■ 4601 Six Forks Road
Phase 2, Suite 528
Raleigh NC 27609
Tel: 919.783.8833
Fax: 919.783.8909

Ohio
6784 Loop Road
Centerville OH 45459
Tel: 513.438.2821
Fax: 513.438.2317

Oregon
■ 15455 NW Greenbrier Pkwy
Suite 210A
Beaverton OR 97006
Tel: 503.645.9882
Fax: 503.645.6612

Pennsylvania
2300 Computer Ave
Building G
Willow Grove PA 19090
Tel: 215.830.1404
Fax: 215.657.4920

Texas
1120 Capital of Texas
Highway South
Building 2
Suite 300
Austin TX 78746
Tel: 512.329.1044
Fax: 512.329.6331

■ 5080 Spectrum Drive
Suite 1010 West
Dallas TX 75248
Tel: 214.788.2966
Fax: 214.233.9234

Washington
■ 3015 112th Avenue NE
Suite 205
Bellevue WA 98004
Tel: 206.822.4384
Fax: 206.827.2884

LSI Logic Corporation of Canada Inc Headquarters
■ Petro-Canada Centre
Suite 3410
150 6th Avenue SW
Calgary AB T2P 3Y7
Tel: 403.262.9292
Fax: 403.262.9494

■ 6400 Roberts Street
Suite 492
Burnaby BC V5G 4C9
Tel: 604.294.8444
Fax: 604.294.8443

■ P.O. Box 8249
Station F
Edmonton AB T6H 4P1
Tel: 403.450.4400
Fax: 403.450.4411

■ 260 Hearst Way
Suite 400
Kanata ON K2L 3H1
Tel: 613.592.1263
Telex: 053.3849
Fax: 613.592.3253

■ 755 St Jean Boulevard
Suite 600
Pointe Claire PQ H9R 5M9
Tel: 514.694.2417
Fax: 514.694.2699

■ 401 The West Mall
Suite 1110
Etobicoke ON M9C 5J5
Tel: 416.620.7400
Fax: 416.620.5005

Denmark
EV Johanssen Electronic AS
■ Titangade 15
DK-2200 Copenhagen
Tel: 45.31.839022
Fax: 45.31.839222

Finland
Oy Fintronic AB
Heikkilantie 2 A
00210 Helsinki
Tel: 358.0.6926022
Fax: 358.0.6821251

France
LSI Logic SA
■ Tour Chenonceaux
204 Rond-Point
du Pont de Sevres
F-92516 Boulogne-Billancourt
Paris
Tel: 33.146.206600
Telex: 631475
Fax: 33.146.203138

Microel SA
Immeuble "Micro"
Avenue de la Baltique
ZA de Courtaboeuf BP 3
F-91941 Les Ulis Cedex
Tel: 33.1.69070824
Fax: 33.1.69071723

Germany
LSI Logic GmbH
European Headquarters
Kronstadter Strasse 9
W-8000 Munich 80
Tel: 49.89.99313100
Fax: 49.89.936806

■ Arabellastrasse 33
W-8000 Munich 81
Tel: 49.89.9269030
Fax: 49.89.917096

■ Am Seestern
Niederkasseler Lohweg 8
W-4000 Dusseldorf 11
Tel: 49.211.5961066
Fax: 49.211.592130

■ Mittlerer Pfad 4
W-7000 Stuttgart 31
Tel: 49.711.139690
Fax: 49.711.8661428

AE Advanced Electronics GmbH
■ Otto-Hahn-Strasse
W-6277 Bad Camberg
Tel: 49.6434.5041
Fax: 49.6434.4277

AL Advanced Logic GmbH
Strassfurther Strasse 1
0-3014 Magdeburg
Tel: 37.91.48112
Fax: 37.91.48112

AE Advanced Electronics GmbH
■ Stefan-George-Ring 19
W-8000 Munich 81
Tel: 49.89.9300980
Fax: 49.89.93009866

AE Advanced Electronics GmbH
Mittlerer Pfad 4
W-7000 Stuttgart 31
Tel: 49.711.881118
Fax: 49.711.8661359

Israel
LSI Logic Limited
■ 40 Sokolov St
Ramat Hasharon 47235
P.O.B. 1331
Tel: 972.3.5403741
Telex: 371662
Fax: 972.3.5403747

Italy
LSI Logic SPA
■ Centro Direnzionale Colleoni-
Palazzo Orione Ingresso 1
20041 Agrate Brianza
Milano
Tel: 39.39.6056881
Fax: 39.39.6057867

Sales Offices and Design Resource Centers (Continued)

Japan
LSI Logic KK
Headquarters
■ Kokusai-Shin Akasaka Bldg.
West Wing, 6-1-20 Akasaka
Minato-ku Tokyo 107
Tel: 81.33.589.2711
Fax: 81.33.589.2740

■ 2-10-1 Kashuga
Tsukuba-shi 305
Tel: 81.298.52.8371
Fax: 81.298.52.8376

■ Chrystal Tower 14F
1-2-27 Shiromi
Chuo-ku Osaka 540
Tel: 81.6.947.5281
Fax: 81.6.947.5287

■ Tama-Plaza Daisen Bldg.
2-19-2 Utsukushigaoka
Midori-ku Yokohama 227
Tel: 81.45.902.4111
Fax: 81.45.902.4533

Tsukuba Mitsui Building
1-6-1 Takezono
Tsukuba-shi 305
Tel: 81.298.56.5011
Fax: 81.298.56.5030

LSI Logic Corporation of Korea Limited
■ 7th Floor Namseoul Bldg.
1304-3 Seocho-Dong
Seocho-ku Seoul
Tel: 82.2.561.2921
Fax: 82.2.554.9327

Netherlands
Arcobel BV
■ Griekenweg 25
Postbox 344
NL-5340 AH Oss
Tel: 31.4120.30335
TWX: 37489
Fax: 31.4120.30635

Norway
Art Chip AS
Ole Deviks vei 44
N-0668 Oslo 6
Tel: 47.2.720740
Fax: 47.2.656960

Spain
LSI Logic SA
Orense 68
28020 Madrid
Tel: 34.1.5705600
Fax: 34.1.5702807

Sweden
LSI Logic Limited
■ Torhamnsgatan 39
S-16440 KISTA
Tel: 46.8.703.4680
Fax: 46.8.7506647

Switzerland
LSI Logic Sulzer AG
■ Erlenstrasse 36
CH-2555 Bruegg/Biel
Tel: 41.32.536363
Fax: 41.32.536367

Taiwan
LSI Logic Corporation
■ 3F 92 Tun-Hua S. Rd
Sec 2 Taipei ROC
Tel: 886.2.755.3433
Fax: 886.2.755.5176

United Kingdom
LSI Logic Limited
■ Grenville Place, The Ring
Bracknell Berkshire
England RG12 1BP
Tel: 44.344.426544
Telex: 848679
Fax: 44.344.481039

■ Lomond House
Beveridge Square
Livingstone Scotland
EH 54 6QF
Tel: 44.506.416767
Fax: 44.506.414836

Manhattan Skyline Limited
Manhattan House, Unit 1
The Switchback, Gardner Rd
Maidenhead Berkshire
England SL6 7RJ
Tel: 44.628.75851
Fax: 44.628.782812

■ Sales Offices with Design Resource Centers

Printed on Recycled Paper

Printed in USA
101.7111.553600.056.2K.TP.RRD

LSI LOGIC

LSI Logic Corporation
U.S. Headquarters
Milpitas CA
408.433.8000
Fax: 408.434.6457
Technical
Assistance:
800.548.RISC

LSI Logic GmbH
European Headquarters
Munich Germany
49.89.99313100
Fax: 49.89.936806

LSI Logic K.K.
Headquarters
Tokyo Japan
81.33.589.2711
Fax: 81.33.589.2740

LSI Logic Corporation
of Canada Inc.
Headquarters
Calgary
403.262.9292
Fax: 403.262.9494

Order No. M14015

Please see inside back pages for
information on the Sales Office or
Distributor nearest you.