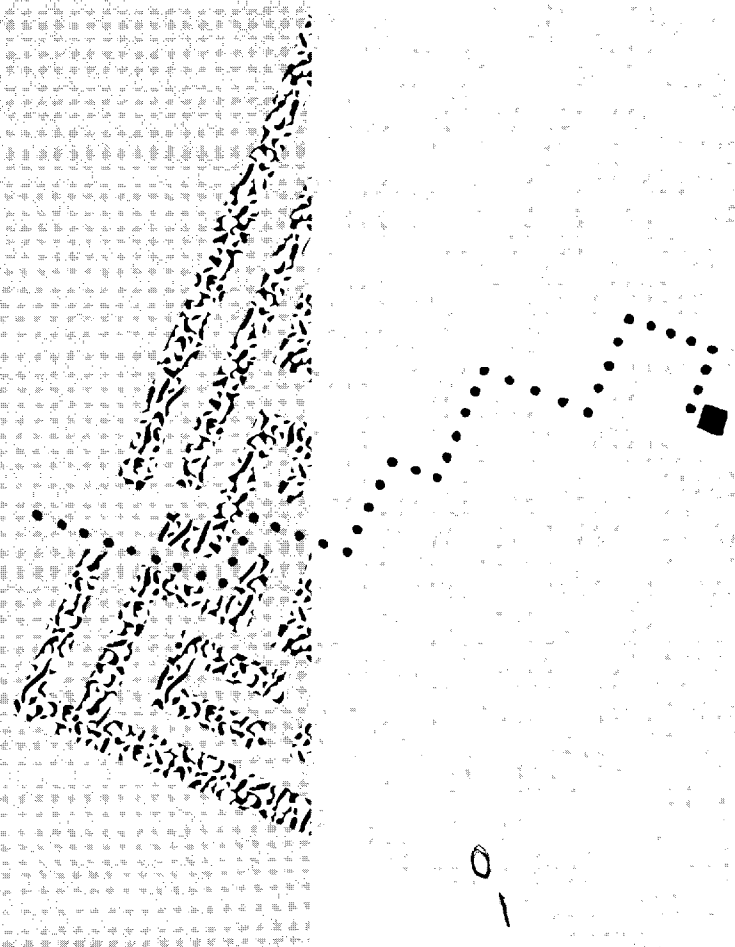
LSI LOGIC

5304804 0012274 4T2 LLC

L64853A
Enhanced SBus DMA Controller
Technical Manual



5304804 0012275 339 LLC

Second Edition
Document Number MS71-000104-99 B
This document applies to revision A of the L64853A Enhanced SBus DMA Controller and to all subsequent versions unless otherwise indicated in a subsequent edition or an update to this edition of the document.

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.

© 1991, 1993 LSI Logic Corporation. All rights reserved.

TRADEMARK ACKNOWLEDGMENT
LSI Logic logo design is a registered trademark of LSI Logic Corporation. SPARC is a registered trademark of SPARC International, Inc. Products bearing the SPARC trademarks are based on an architecture developed by Sun Microsystems, Inc. All other brand and product names may be trademarks of their respective companies.

Preface

This book is the primary reference and technical manual for the L64853A Enhanced SBus DMA Controller. It contains a complete functional description for the L64853A and includes complete physical and electrical specifications for the L64853A.

Audience

This book assumes that you have some familiarity with microprocessors and related support devices. It also assumes readers have access to additional information about the SPARC workstation—in particular, the SBus specification, which can be obtained from Sun Microsystems, Inc. This book provides information for both system-level programmers and hardware designers as follows:

- **System-Level Programmers** — For the system-level programmer, this manual briefly describes the SPARC workstation architecture, then fully describes the function of the chip with respect to the architecture.

  Chapter 1 describes the basic features and operation of the L64853A Enhanced DMA Controller and the chip's relationship to the other SPARC chips. Chapter 2 describes the internal, programmable registers of the L64853A, and it describes how to access both internal and external registers. Chapter 3 covers the two modes of operation and describes the internal cache memory. Chapters 4 and 5 further describe the operation of the D and E channels. To aid in implementation of the chip in a typical environment, Chapter 8 shows how to access the Emulex SCSI Processor (ESP-100) and the AMD Am7990 Local Area Network Controller for Ethernet (LANCE) internal registers.

- **Hardware Designers** — For the hardware designer, this manual provides the electrical, logical, and mechanical data necessary to integrate the L64853A DMA Controller into the SPARC workstation.

  Chapter 1 describes the architecture and operation of the L64853A. Chapters 4 and 5 provide detailed information on the D and E channels, respectively. Chapter 6 describes the data transfer cycles for the D and E channels and the signals used by the L64853A DMA Controller during read and write operations. Chapter 7 provides the AC, DC, environmental, and mechanical specifications for the L64853A. Chapter 8 discusses how to interface the L64853A to the ESP-100 and the LANCE.

Organization

This book has the following chapters:

- Chapter 1, **Introduction**, describes the basic features and operation of the L64853A Enhanced DMA Controller, showing its relationship to the other SPARC system elements.
- Chapter 2, **Registers**, describes the types of registers available on the L64853A and how to access internal and external registers.
- Chapter 3, **L64853A Operation**, describes the modes of operation on the SBus, which is used for data transfers between the peripheral subsystems and main memory. This chapter also discusses the internal cache memory, which provides data buffering between the peripheral subsystems and main memory.
- Chapter 4, **D-Channel Operation**, describes the DMA Controller's D channel, an eight-bit DMA control channel.
- Chapter 5, **E-Channel Operation**, describes the DMA Controller's E channel, a 16-bit DMA control channel.
- Chapter 6, **Interface Description**, describes the input and output signals used by the L64853A.
- Chapter 7, **Specifications**, provides the AC, DC, environmental, and mechanical specifications for the L64853A.
- Chapter 8, **Applications**, describes the types of SPARC workstation applications that the L64853A Enhanced DMA Controller fits into and provides interface examples to Emulex SCSI Processor (ESP-100) and to an AMD Local Area Network Controller (AM7990).
- Appendix A, **L64853 and L64853A Differences**, lists the major differences between the L64853 and L64853A SBus DMA Controllers.
- Appendix B, **Customer Feedback**, is a form for you to fill out with your comments on the content and quality of this document.

Related Publications

Building a Low-Cost SPARC-Based Multimedia Workstation White Paper, Order No. M22004.A

Designing with the SparKIT Chipset White Paper, Order No. M22001.A

SBus Solutions for Graphics, DMA, and All Other Applications White Paper, Order No. M22002.A

The SBus Specification available from Sun Microsystems, Inc., 2550 Garcia Avenue, Mountain View, CA 94042

Conventions Used in this Manual

The first time a word or phrase is defined in this manual, it is *italicized.*

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 (¯) over its name.
- An edge-significant signal that initiates actions on a HIGH-to-LOW transition always has an overbar (¯) over its name.

The word *assert* means to drive a signal true or active. The word *deassert* means to drive a signal false or inactive.

Hexadecimal numbers are indicated by the prefix "0x" before the number—for example, 0x32CF. Binary numbers are indicated by a subscripted "2" following the number—for example, $0011.0010.1100.1111_2$.

Contents

5304804 0012280 7T6 LLC

Chapter 1
Introduction

This chapter describes the basic capabilities and operation of the L64853A Enhanced SBus DMA Controller. This chapter is particularly important for system-level programmers and hardware designers.

1.1 General Description

The L64853A Enhanced SBus DMA Controller provides a complete SBus interface for SBus peripheral subsystems. The L64853A, implemented in a 1.5-micron CMOS process and manufactured by LSI Logic Corporation, is packaged in an inexpensive, 120-pin, plastic quad flat package (PQFP).

The L64853A contains two independent DMA channels, a 16-bit channel and an 8-bit channel. The two channels support DMA for use in applications that require operation as an SBus Master. Such applications include:

- Ethernet controllers
- Eight-bit controllers

The L64853A's two channels can also be used for applications that rely on programmed I/O and thus use only the L64853A's SBus slave capability. These applications include:

- Serial ports
- Analog-to-Digital converters

The 8-bit channel is called the *D channel*, and the 16-bit channel is called the *E channel*. The L64853A generates, upon request from a device attached to either channel, sequences of SBus data transfers (that is, reads or writes) between the peripheral controller and main memory. To perform this function, the L64853A can become an SBus Master through the use of the SBus Request and Grant signals. In SBus terminology, the L64853A is a *Direct Virtual Memory Access (DVMA) Master*, that is, it generates virtual addresses on the SBus data lines and employs the SBus controller's Memory Management Unit (MMU) to translate these virtual addresses

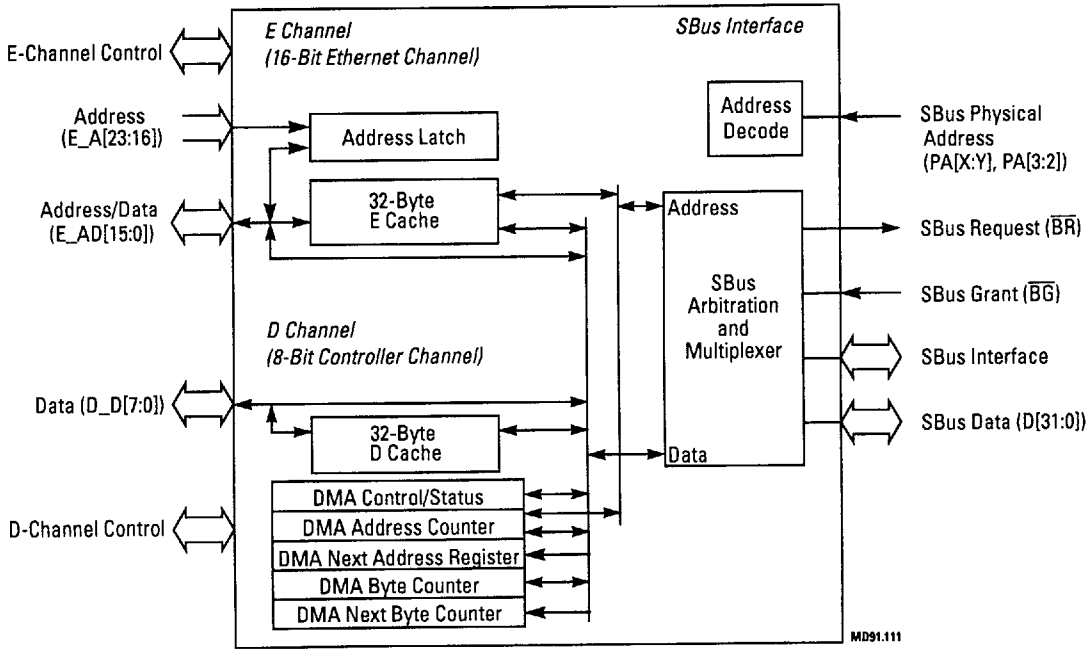
into physical addresses. (See *The SBus Specification* from Sun Microsystems for more details.)

The L64853A is programmable. Programs running on the system CPU may set parameters that both govern the transfers to and from the peripheral controllers performed by the L64853A and also read the current status of the chip. Software writes and reads the internal registers on the L64853A and also on the two peripheral controllers. To permit this writing and reading, the L64853A becomes an *SBus Slave* when the CPU asserts the Slave Select signal specific to the L64853A. While the L64853A acts as an SBus Slave, the CPU can use the L64853A as a conduit to the two peripheral controller chips, thus allowing the CPU to program them.

Architecture

The principal components of the L64853A are its two functionally distinct DMA channels and its SBus interface with associated bus arbitration logic. Figure 1.1 shows the chip's internal architecture. Refer to Chapter 6 for a detailed description of the L64853A DMA Controller's signals.

Figure 1.1
L64853A Internal Block Diagram



The *E Channel* (16-bit Ethernet Channel) has the following functional blocks:

- An *Address Latch* temporarily stores memory addresses from the E-channel controller
- A 32-byte *E Cache* buffers data between the 16-bit interface and the SBus

The *D Channel* (eight-bit I/O Controller Channel) has the following functional blocks:

- A 32-byte *D Cache* buffers data between the 8-bit interface and the SBus
- A *Control/Status Register* contains programmable parameters and status fields
- An *Address Counter Register* holds the memory address of the next byte the D channel accesses
- A *Next Address Register* holds the memory address for the next data block transfer on the SBus
- A *Byte Counter Register* counts down the number of bytes transferred during a transfer of a block of data
- A *Next Byte Counter Register* holds the number of bytes to be transferred during the next data block transfer

Channel Operations

When the L64853A is an SBus Master, it performs high-speed data transfers between peripheral controllers and main memory. When the L64853A is an SBus Slave, a Bus Master performs register reads and writes to the L64853A, and also performs register reads and writes to the D-channel and E-channel controllers through the L64853A. All channel operations are ultimately governed by the SPARC environment in which the L64853A is implemented.

Chapter 3 describes the features of the L64853A common to both the D and E channels. Chapters 4 and 5 provide detailed information on the D and E channels, respectively.

1.2 Key Features

This section summarizes the key features of the L64853A. These features are described in more detail in subsequent chapters.

- Operates at speeds up to 25 MHz
- Performs DMA transfers, assuming an SBus latency of up to 40 clock cycles, at a maximum 8.3 Mbytes/second at 25 MHz for both the D and E channels
- Supports 8- or 16-bit peripherals
- Packs/unpacks SBus words into bytes or halfwords for use by the peripheral controllers
- Supports for byte, halfword, word, or four-word burst transfers on the SBus
- Operates in virtual address space with the SPARC MMU providing virtual-to-physical address translations
- Supports rerun acknowledgments
- Includes 24-bit address and data counters
- Uses a single clock input
- Performs data block chaining on the D-channel interface
- Packaged in a low-cost, 120-pin plastic quad flat package

1.3 Enhancements

The L64853A Enhanced SBus DMA Controller is an enhanced version of the L64853 SBus DMA Controller. The features added to the L64853A are:

- Larger internal data buffer to support four-word data burst transfers.
- Two additional D-channel registers (Next Address Register and Next Byte Counter) that allow data transfer setups to be pipelined, thus reducing the overhead between data transfers.
- Programmable E_ALE/$\overline{\text{E\_AS}}$ pin on the E channel for compatibility with various 16-bit controllers.
- FASTER option that increases the speed of D-channel read and write operations.

Chapter 2
Registers

This chapter discusses the L64853A's registers and tells how to access both internal and external registers. This chapter is intended primarily for system programmers.

The L64853A has two types of internal registers:

- Programmable registers

  The L64853A's five *programmable registers* control the operation of the chip and contain status information.

- ID Register

  The *ID Register* is pre-programmed with a unique, read-only value for the L64853A chip.

The following sections discuss the programmable and ID registers in more detail. Chapter 3 provides additional details about the fields and bits in these registers while the D and E channels are operating.

2.1 Internal Programmable Registers

This section describes the five programmable L64853A registers:

- DMA Control/Status Register (CSR)
- DMA Address Counter and Next Address Registers
- DMA Byte and Next Byte Counters

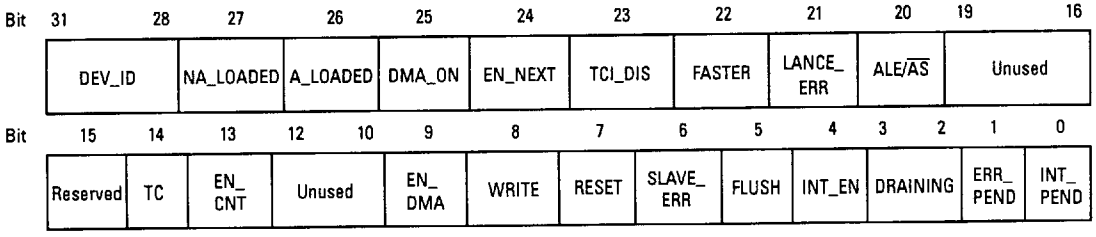
The DMA Control/Status Register (CSR) contains control and status information for both the D and E channels. The remaining programmable registers control only the D channel and the SBus. The E channel is controlled entirely by external registers on the E-channel peripheral controller itself.

The following subsections define the fields within the L64853A registers in detail. In these definitions, the set state = 1 and the clear state = 0. This convention applies to all bits in the L64853A internal registers.

DMA Control/ Status Register (CSR)

The 32-bit Control/Status Register (shown in Figure 2.1 below) controls the operation of the D channel and reports the status of the D and E channels. The following description provides the functions of the individual fields in the CSR.

Figure 2.1
DMA Control/Status Register (CSR)



INT_PEND **Interrupt Pending (Read Only)** **0**

The L64853A automatically sets INT_PEND to indicate a pending interrupt. Asserting $\overline{\text{D\_IRQ}}$ or setting TC (when not disabled by TCI_DIS) sets INT_PEND. If no interrupts are pending, this bit is clear.

ERR_PEND **Error Pending (Read Only)** **1**

ERR_PEND is set when an error condition occurs during a D-channel memory access (for example, a parity error, protection violation, or time-out). Setting ERR_PEND causes an interrupt ($\overline{\text{INTREQ}}$ asserted) if INT_EN is set. DMA transfers are stopped when ERR_PEND is asserted. ERR_PEND is reset when either the FLUSH bit is set, the RESET bit is set, or $\overline{\text{RESET}}$ is asserted.

DRAINING **Draining (Read Only)** **[3:2]**

Dirty data are data in the D cache that are pending a write to the SBus. *Draining* is the process of writing the dirty data to the SBus. If one of the D channel's cache lines is draining dirty data, the DRAINING bits (bits [3:2]) read as 11_2. While these bits are ones, do not set the RESET or FLUSH bits or write to the DMA Address Counter Register. The DRAINING bits are not valid while ERR_PEND is set or during D-channel read operations and should be ignored. Note that these two bits are backward-compatible with the PACK_CNT bits, which occupy the same bit positions in the L64853 CSR.

INT_EN **Interrupt Enable (Read/Write)** **4**
When set, INT_EN enables the SBus $\overline{\text{INTREQ}}$ signal when INT_PEND or ERR_PEND is set.

FLUSH **Flush Buffer (Write Only)** **5**
Setting FLUSH marks all bytes in the D-channel cache as invalid and resets ERR_PEND and TC. If EN_NEXT = 1, A_LOADED and NA_LOADED are also reset. Note that the FLUSH bit also resets itself, hence it always reads as zero. Note also that software should never set this bit while the EN_DMA or DRAINING bits are set.

SLAVE_ERR **Slave Error (Read/Write)** **6**
The L64853A sets the SLAVE_ERR bit when an SBus Master tries to access the L64853A with an unsupported SBus size. In this situation, the L64853A responds with an SBus Error Acknowledgement and sets this bit. Write a one to this bit to reset it.

RESET **Reset DMA (Read/Write)** **7**
When set, RESET acts as a hardware reset. The L64853A is initialized into the following state: ERR_PEND, INT_EN, FLUSH, EN_NEXT, DRAINING, SLAVE_ERR, WRITE, EN_DMA, EN_CNT, TC, DMA_ON, and FASTER are set to zero, and RESET is set to one. All valid/dirty bits in the cache are cleared. An SBus reset ($\overline{\text{RESET}}$ = 0) sets all CSR bits to zero except for the DEV_ID field. The D_RESET signal remains asserted for as long as the RESET bit of the CSR or the $\overline{\text{RESET}}$ signal is active. Note that software should never set this bit while the EN_DMA or DRAINING bits are set.

WRITE **Memory Read/Write (Read/Write)** **8**
WRITE determines the direction of the D-channel DMA transfer. When HIGH, the flow is to memory (memory write); when LOW, the flow is from memory (memory read).

EN_DMA **Enable DMA (Read/Write)** **9**
When set, EN_DMA allows the L64853A to respond to DMA requests by the D-channel controller, as long as DMA activity is not stopped due to an interrupt, etc. When this bit is one, do not set the RESET or FLUSH bits or write to the DMA Address Counter Register. See the description of the DMA_ON bit for more information.

Unused **Not Used (Read Only)** **[19:16], [12:10]**
These bits are unused; they always read as zeros.

EN_CNT **Enable Counter (Read/Write)** **13**
As long as EN_CNT is set, the internal Byte Counter is enabled. The Byte Counter decrements on byte transfers between the D channel and the L64853A. The set state also enables the operation of the TC bit. EN_CNT and EN_NEXT control the operating modes of the D-channel programmable registers. Refer to the table in the EN_NEXT description on page 2-5 for these modes.

TC **Terminal Count (Read Only)** **14**
When set, TC indicates that the Byte Counter has expired. That is, TC is set when a byte count makes a transition from 0x00.0001 to 0x00.0000. When the L64853A sets the TC bit, an interrupt is generated on the $\overline{\text{INTREQ}}$ pin, as long as interrupts are enabled by INT_EN and not disabled by TCI_DIS. Also, if the operation was a DMA write, the L64853A queues all dirty bytes in the cache for draining. When EN_NEXT = 0, TC is cleared only by the FLUSH bit, the RESET bit, or the $\overline{\text{RESET}}$ pin. When EN_NEXT = 1, TC can also be cleared by writing a one to it.

Reserved **Reserved (Read/Write)** **15**
Bit 15 is reserved for future enhancements. For correct operation, always write a zero to this bit.

ALE/$\overline{\text{AS}}$ **Address Latch Enable/Address Strobe (Read/Write)** **20**
ALE/$\overline{\text{AS}}$ defines pin 27 as either ALE (active HIGH signal) or $\overline{\text{AS}}$ (active LOW signal). If this bit = 1, pin 27 is ALE. If this bit = 0, pin 27 is $\overline{\text{AS}}$. The default value for this bit is zero.

LANCE_ERR **LANCE Error (Read Only)** **21**
LANCE_ERR is set when a memory error occurs on a transfer to or from the E channel. The L64853A does not generate an interrupt in response to LANCE_ERR being set. The L64853A does not respond to any subsequent E-channel transfers until LANCE_ERR is cleared. A slave write to the E channel clears this bit.

FASTER **Fast Speed (Read/Write)** **22**
The FASTER bit is set when a faster access time is required for the D channel. Refer to the timing diagrams in Section 4.5, "Read/Write Transactions," for specific details.

TCI_DIS **TC Interrupt Disable (Read/Write)** **23**

When TCI_DIS is set, interrupts and auto-draining are disabled from TC. The default value for this bit is zero.

EN_NEXT **Enable Next (Read/Write)** **24**

As long as EN_NEXT is set, the next address auto-load mechanism is enabled. The default value for this bit is zero. EN_NEXT and EN_CNT control the operating modes of the DMA Address Counter Register, Next Address Register, Byte Counter, and Next Byte Counter as shown in the following table.

EN_CNT	*EN_NEXT*	*Mode*
0	x[1]	Backward-compatible with the L64853
1	0	Backward-compatible with the L64853 using the Byte Counter. FLUSH clears the TC flag.
1	1	Next Address and Next Byte Count values loaded into the Address Counter and Byte Counter Registers on byte count expiration. DMA stops on byte count expiration if the Next Address Register is not loaded.

1. x = don't care.

DMA_ON **DMA On (Read Only)** **25**

When DMA_ON is set, the L64853A responds to DMA requests from the D channel. This bit reads as one when ((A_LOADED OR NA_LOADED) & EN_DMA & NOT ERR_PEND) is true. When this bit is clear, the L64853A ignores DMA requests from the D channel.

A_LOADED **Address Loaded (Read Only)** **26**

When either software writes to the Address Counter Register or the L64853A copies the contents of the Next Address Register into the Address Counter Register, the A_LOADED bit is set. This bit is cleared by RESET assertion or Byte Counter expiration. A_LOADED is also reset by FLUSH when EN_NEXT = 1.

NA_LOADED **Next Address Loaded (Read Only)** **27**

Writing to the Next Address Register while EN_NEXT = 1 sets NA_LOADED. This bit is cleared when RESET is set, EN_NEXT is cleared, EN_CNT is cleared, or when the Next Address Register is copied to the Address Counter Register. The latter case occurs when both NA_LOADED = 1 and A_LOADED = 0. NA_LOADED is also reset by FLUSH when EN_NEXT = 1.

DEV_ID **Device ID (Read Only)** **[31:28]**
For the current implementation, this field is set to 1001_2.

DMA Address Counter and Next Address Registers

The DMA Address Counter Register and the DMA Next Address Register, shown in Figure 2.2, contain the virtual addresses for the current and next memory transfers, respectively. The Next Address Register is a 32-bit register. The Address Counter Register is divided into a 24-bit counter and an 8-bit register.

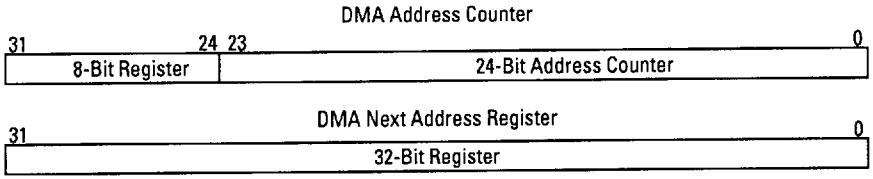


Figure 2.2 DMA Address Counter and Next Address Registers

The eight-bit register in the Address Counter Register contains the high-order byte of the virtual address; the upper eight bits are fixed since DMA transfers are limited to 16 Mbytes. The 24-bit counter contains the lower three bytes of the virtual address. The Address Counter Register always points to the next byte to be accessed by the D-channel controller, independent of which bytes in memory have been accessed by the L64853A.

The Next Address Register, in conjunction with the Next Byte Counter Register, allows data transfer setups to be pipelined, thus reducing the amount of overhead between block transfers. The virtual address for the next data transfer is preloaded into the Next Address Register. When NA_LOADED = 1 and A_LOADED = 0, the L64853A automatically copies the contents of the Next Address Register into the Address Counter Register.

If data block chaining is enabled (EN_NEXT = 1), both the Address Counter Register and the Next Address Register are used in the DMA operation. If data block chaining is disabled (EN_NEXT = 0), only the Address Counter Register is used.

After a RESET, the Address Counter Register and Next Address Register both contain indeterminate values.

Note that software should never write to the Address Counter Register while the EN_DMA or DRAINING bits are set. If the Address Counter

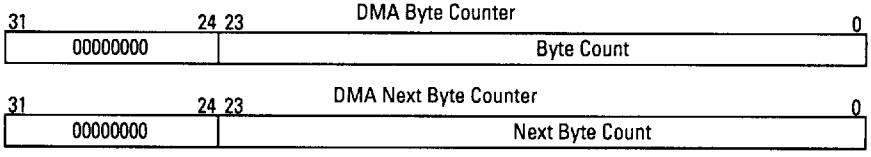
Register is written to, the L64853A marks all bytes in the D cache as invalid.

See Section 4.2, "Register Operation," for more information on the DMA Address Counter Register and the DMA Next Address Register.

DMA Byte and Next Byte Counters

The Byte Counter and the Next Byte Counter are 32-bit registers that hold the number of bytes in a D-channel DMA transfer. The Byte Counter contains the byte count for the current memory transfer. The Next Byte Counter contains the byte count for the next memory transfer. These registers are shown in Figure 2.3.

Figure 2.3 DMA Byte and Next Byte Counters



The upper byte of these registers always reads as zero. If data block chaining is enabled (EN_NEXT = 1), both the Byte Counter and the Next Byte Counter are used during DMA transfers (provided that EN_CNT = 1). If data block chaining is disabled (EN_NEXT = 0), only the Byte Counter is used.

Loading the Byte Counter with 0 allows the transfer of 2^{24} bytes (16 Mbytes) before it expires.

The Next Byte Counter Register, in conjunction with the Next Address Register, allows data transfer setups to be pipelined, thus reducing the amount of overhead between block transfers.

After a RESET, the Byte Counter and Next Byte Counter both contain indeterminate values.

See Section 4.2, "Register Operation," for more information on these registers.

2.2 Internal and External ID Registers

The SBus DMA contains a 32-bit internal identification (ID) register and a facility for accessing an external ID register or PROM. The external ID register or PROM supports the automatic configuration feature of *The SBus Specification*. Upon power-up, the CPU reads certain memory locations or slots to determine the system configuration. When the CPU

accesses the L64853A slot with physical address 0, the DMA controller either outputs the value in the internal ID register or begins an access similar to a slave access to the D channel to read the external ID register or PROM. The access uses $\overline{\text{ID\_CS}}$ as the chip select instead of $\overline{\text{D\_CS}}$.

The L64853A determines whether to use an internal or external ID by the state of the $\overline{\text{ID\_CS}}$ signal. If the bidirectional $\overline{\text{ID\_CS}}$ signal is tied LOW, then the DMA controller uses the internal ID register value. If $\overline{\text{ID\_CS}}$ is pulled HIGH through a 4.7-kΩ resistor, then the L64853A uses an external ID register or PROM. The $\overline{\text{ID\_CS}}$ pin then becomes a chip select output. The L64853A pulls the $\overline{\text{ID\_CS}}$ pin LOW to indicate an external ID access.

The L64853A internal ID register is hardwired to 0xFE81.0102. The internal ID register should be used mainly for system testing purposes. Actual system implementations should use an external ID register or PROM to avoid ID conflicts with multiple DMA devices on a single system. The information programmed in the ID PROM can range from a manufacturer's name and model number to a device driver. Note that if a slot does not respond with that information during the ID access, that slot is subsequently ignored.

2.3 Addressing Internal and External Registers

Any SBus peripheral device that is built using the L64853A contains two types of configuration and control registers: the internal L64853A registers, described in Section 2.1, and the external registers in the peripheral controllers that are attached to the L64853A's two channels. Examples of external registers are listed in Chapter 8.

The implementation of the SBus controller and the way that the hardware designer connects the physical address signals determine the system addresses that specify the internal or external registers.

With regard to the SBus controller implementation, the SBus uses a geographical addressing scheme that assigns ranges of addresses to each SBus connector. When the SBus controller detects access to a particular range, it asserts the $\overline{\text{SEL}}$ signal dedicated to the related connector.

When the SBus controller asserts the L64853A's $\overline{\text{SEL}}$ input, the L64853A uses four address inputs to further decode the address present on SBus signals PA[27:0]: PA[X:Y] and PA[3:2]. The PA[X:Y] signals are connected to two of the SBus's 28 physical address lines, PA[27:0]; for example on the L64853A in the SPARCstation 2, PA[X:Y] = PA[23:22]. The L64853A

uses PA[X:Y] to select the category of registers for access according to Table 2.1. The size of the data item that must be transferred over the SBus, specified by SIZ[2:0], is also shown for each category of register.

Table 2.1 Accesses by Register Type

PA[X:Y]	*Addressed Register Type*	*Size*
0 0	Internal ID Register (or external if $\overline{ID\_CS}$ = HIGH)	Byte, Halfword[1], Word[1]
0 1	Internal Programmable Registers	Word
1 0	D-channel Registers (External)	Byte, Halfword[1], Word[1]
1 1	E-channel Registers (External)	Halfword, Word[1]

1. Allowed due to SBus bus-sizing protocol.

Table 2.2 lists the addresses of the programmable registers. These registers are selected when PA[X:Y] = 01_2, $\overline{SEL}$ = $\overline{AS}$ = 0.

Table 2.2 Internal Programmable Registers

PA[3:2]	*EN_NEXT*[1]	*A_LOADED*[1]	*Type*	*Register*
0 0	x	x	R/W	Control/Status Register (CSR)
0 1	0	x	R/W	Address Counter Register
0 1	1	0	R/W	Address Counter Register
0 1	1	1	R	Address Counter Register
0 1	1	1	W	Next Address Register
1 0	0	x	R/W	Byte Count Register[2]
1 0	1	0	R/W	Byte Count Register[2]
1 0	1	1	R	Byte Count Register[2]
1 0	1	1	W	Next Byte Count Register[2]
1 1	x	x	R	Reserved for Testing

1. EN_NEXT and A_LOADED are bits within the CSR.
2. The Byte Counter Registers are used only when the EN_CNT bit in the Control/ Status Register equals 1.

Other physical address lines are wired directly to the two peripheral controllers and are used in an implementation-dependent manner to select specific registers on these two chips.

2.4 Internal Register Reads and Writes

The accessible registers in the L64853A include the Control/Status Register, Address Counter Register, and Byte Counter. The Next Address Register and the Next Byte Counter are write-only registers. The following steps describe how to read or write one of these internal L64853A registers:

1. The CPU executes a LOAD or STORE instruction and asserts the L64853A's Slave Select signal ($\overline{SEL}$). The physical address is also driven onto PA[27:0] at this time along with RD, SIZ[2:0], and $\overline{AS}$.
2. Because the L64853A recognizes that the read or write access is intended for itself ($\overline{SEL}$ is asserted), the L64853A interprets the two-bit value (00_2 or 01_2) on the physical address lines PA[X:Y]. This two-bit value indicates that an internal L64853A register is the target of the read or write. Refer to Table 2.2 for the assignment of physical addresses to L64853A internal registers.
3. If the target register is internal to the L64853A, the data are either read from the intended register to the data lines D[31:0] or written from the data lines to the intended register.
4. Finally, the L64853A asserts $\overline{ACK2}$ to inform the SBus controller that the slave-mode operation is complete.

Figure 2.4 illustrates both a register read and a register write operation for the L64853A. See *The SBus Specification* for further details on SBus signals not shown here.

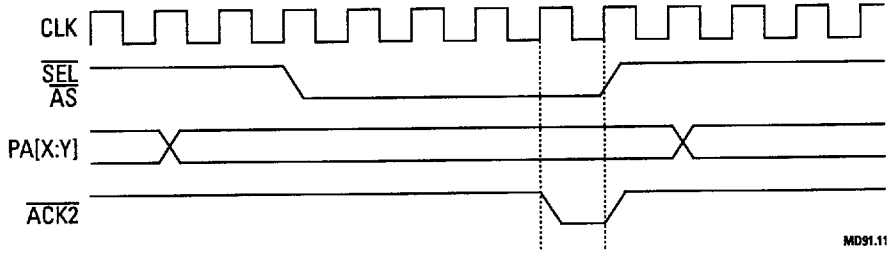


Figure 2.4 L64853A Register Read/Write Cycles

Chapter 3
L64853A Operation

This chapter describes the operational modes of the L64853A and the internal cache organization. These descriptions apply to both the D and E channels. Details specific to the individual channels are provided in Chapter 4, "D-Channel Operation," and Chapter 5, "E-Channel Operation."

3.1 Modes of Operation

The L64853A supports two basic modes of operation: Master and Slave. The basic operations of both the D and E channels are DMA Reads and Writes, executed when the L64853A is in Master mode, and Register Reads and Writes, executed when the L64853A is an SBus Slave.

DMA transfers are accomplished by the coordinated activity of all chips in the system: the peripheral controller, the L64853A, the CPU, the SBus controller, and the memory controller.

Master Mode

In Master mode, the L64853A is an SBus DVMA Master and performs DMA reads and writes. The L64853A generates virtual addresses on the SBus data lines. These addresses are translated into physical addresses by the SBus controller's MMU. Data is then transferred between the internal cache of the L64853A and either memory or another SBus Slave.

All DMA reads from memory are in four-word bursts (see Figure 3.1). DMA writes to memory can be any size: byte, halfword, word, or four-word burst. The L64853A SBus interface always uses the largest size possible when writing to memory. The largest possible size is determined by the total number of bytes written to the internal cache by the I/O device. Note that because the L64853A uses four-word bursts, the slave device that it is accessing must also support four-word bursts.

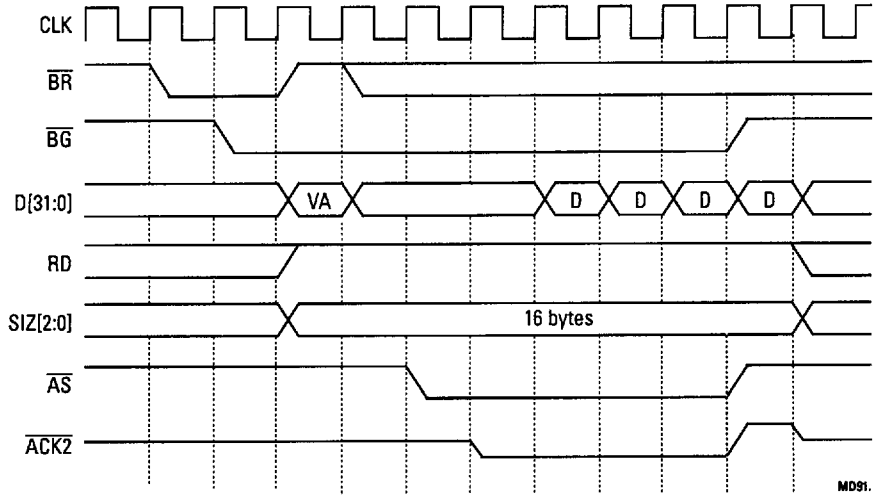


Figure 3.1 SBus DMA Burst Read

The L64853A SBus interface supports rerun acknowledgments from slaves. The requested DMA transfer repeats until either it completes or an error indicator, for example, $\overline{ACK0}$, is received, whereupon the transfer aborts. When the L64853A receives a rerun acknowledgment, it deasserts $\overline{BR}$ for one clock to give the CPU and other DMA Masters a chance to win SBus arbitration. The L64853A then reasserts $\overline{BR}$ to retry the cycle. After the L64853A asserts $\overline{BR}$, it does not deassert it until $\overline{BG}$ or $\overline{RESET}$ is asserted.

Slave Mode

In Slave mode, the CPU is the SBus Master, and it performs register reads and writes of the L64853A as an SBus Slave. The SBus physical address lines access the L64853A's internal registers and those of the two peripheral controllers, using a register addressing scheme that is implementation-dependent. The L64853A's internal cache is bypassed in the Slave mode.

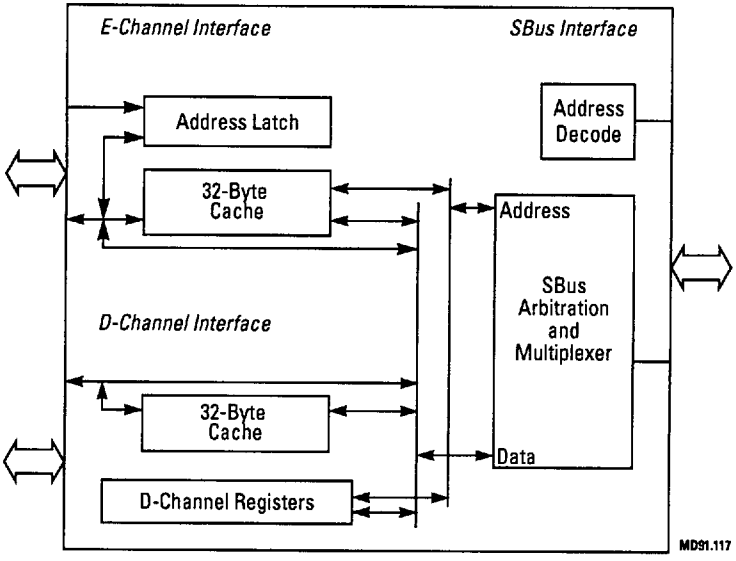
If the SBus Master attempts a slave access to the D or E channel while the channel is currently active with a DMA transfer to/from the L64853A, the L64853A forces the Master to rerun the slave access. This rerun acknowledgment ensures that a deadlock does not occur between the peripheral controllers and the SBus Master.

The L64853A allows the SBus Master to access internal L64853A registers while the L64853A is active with a DMA transfer to/from one of the peripheral controllers.

3.2 Internal Cache Memory

This section discusses the L64853A's internal cache configuration. The internal cache is used only during DMA transfers. The D and E channels each include a 32-byte cache memory, as shown in Figure 3.2. The caches are the data buffers between the SBus and the peripheral controllers. Data are packed or unpacked in the appropriate cache memory. Packing and unpacking the data in the cache allows for a four-word burst and reduces the impact on the SBus bandwidth.

Figure 3.2 L64853A Block Diagram

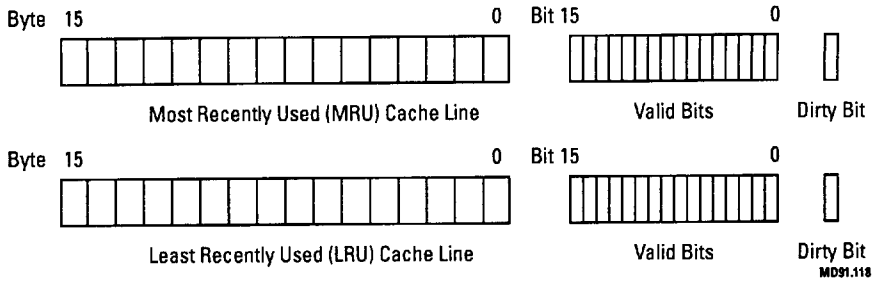


Each cache is divided into two lines of 16 bytes (four words) each. The cache lines are referred to as Most Recently Used (MRU) and Least Recently Used (LRU), depending on when each line was last accessed, where an access is either a read or a write. The line most recently accessed by the peripheral controller is marked as MRU, and the other as LRU. Once the last byte of the MRU cache line is accessed (the least significant address bits of the last byte = 0xF), that cache line is then marked LRU and the other cache line is marked as MRU.

Each 16-byte cache line has a dirty bit and 16 valid bits associated with it, where each valid bit corresponds to a byte in the cache line. When set, the valid bit indicates that its corresponding byte contains valid data for a memory read or write. The dirty bit refers to the entire cache line. When set, the dirty bit indicates that the cache line contains data for a memory

write. When the dirty bit is reset, the cache line does not contain data for a write. Figure 3.3 illustrates the cache lines with their associated bits.

Figure 3.3 Cache Line Configuration



Memory Reads

For memory reads, a cache miss occurs when the data requested by the peripheral device are not in the cache line. When a cache miss occurs, the L64853A loads the cache with a four-word burst from memory. The data from the four-word aligned, four-word burst are loaded into the LRU cache line. As each byte is read into the cache, the L64853A sets the byte's corresponding valid bit. When the data are transferred to the peripheral device, the valid bit for the corresponding byte or bytes is cleared. Once the valid bit is cleared, data at that particular cache address cannot be read again. This step ensures consistency between the internal cache and memory.

The L64853A performs *read-ahead operations* in order to maximize the number of cache hits. Read-ahead operations fill an entire cache line with a line from memory. These operations occur when the least significant address bits of the requested data are 0100_2 for a D-channel read and 0110_2 for an E-channel read. Refer to Section 4.3, "D-Channel Cache Operation," and Section 5.2, "E-Channel Cache Operation," for more details on read-ahead operations.

Memory Writes

For memory writes, the peripheral device first writes the data to the cache. Writing the first byte to the cache sets the dirty bit for that cache line. The valid bit that corresponds to each byte written to the cache is also set. The data are packed into the cache until either the line is full or another draining condition is met. The cache line is marked LRU, and the valid data in the line are scheduled for draining to memory. The other cache line is marked MRU, and transfers between the peripheral device and the L64853A can continue simultaneously with the SBus transfer.

Draining conditions are different for the D and E channels. Refer to Section 4.3, "D-Channel Cache Operation," and Section 5.2, "E-Channel Cache Operation," for more information on draining in the D and E caches.

Chapter 4
D-Channel Operation

Chapter 4 describes the organization and operation of the D channel, for both programmers and hardware designers. This chapter is divided into the following sections:

- General Description: This section presents the D channel in general terms.
- Register Operation: This section describes the register and counter operation during memory transfers between the D channel and the L64853A.
- D-Channel Cache Operation: This section explains the D-cache data transfer for both memory read and write operations.
- Memory Errors and Interrupts: This section discusses the detection and resolution of memory errors and interrupts.
- Read/Write Transactions: This section describes the signal sequences required for DMA and slave register read/write operations.
- Programming Notes: This section provides programming tips on data transfers and register operation.

4.1 General Description

The D channel is an eight-bit DMA control channel. The L64853A D-channel interface supports DMA memory reads and writes (Master mode) and register read and write operations (Slave mode).

A DMA transfer from the D channel to memory consists of two separate transfers: one or more eight-bit transfers from the D channel to the D cache, then a byte, halfword, or word transfer from the D cache to memory. Similarly, a transfer from memory to the D channel consists of two transfers: a four-word burst from memory to the D cache, and one or more eight-bit transfers from the D cache to the D channel. Note that all memory reads are in four-word bursts.

The D channel contains a 32-byte cache (the D cache) and five programmable registers. The D cache holds data that are being transferred between the D channel and memory. See Section 3.2 for more information on the internal cache setup and Section 4.3 for D-cache specific information.

The five registers are the DMA Address Counter Register, DMA Next Address Register, DMA Byte Counter, DMA Next Byte Counter, and Control/Status Register (CSR). The Address Counter and Next Address Register contain the virtual addresses for the current and next memory transfers, respectively. The Byte Counter and Next Byte Counter contain the byte counts for the current and next memory transfers, respectively. The two Next registers are used exclusively in the data block chaining mode described in Section 4.2.

The CSR contains several programmable fields that configure the D-channel interface. The CSR also provides status information on the D and E channels.

4.2 Register Operation

This section describes the operation of the address and byte count registers during a memory transfer between the D channel and the L64853A. It discusses two modes of operation: with and without data block chaining. Data block chaining provides a method of pipelining data block setups, thereby reducing latency time between block transfers. Data block chaining is enabled by the EN_NEXT bit in the CSR.

Note that the subsections below assume that the EN_CNT bit in the CSR is set so that Byte Counter is enabled.

Operation without Data Block Chaining

When data block chaining is disabled (EN_NEXT = 0), the Address Counter Register and Byte Counter (if EN_CNT = 1) are used in the DMA transfer. Before a DMA transfer begins, the byte count for the block transfer is loaded into the Byte Counter. The starting virtual address is loaded into the Address Counter.

Each time a byte is transferred between the L64853A and the D channel, the byte count is decremented by one and the address is incremented by one. The Byte Counter expires (changes from 0x00.0001 to 0x00.0000) upon completion of the block transfer. When the Byte Counter expires, the L64853A sets the Terminal Count (TC) bit in the CSR.

When the first block transfer completes, the L64853A generates an interrupt. At this time, the Address Counter Register and Byte Counter can be loaded with the values for the next memory transfer.

Operation with Data Block Chaining

When data block chaining is enabled (EN_NEXT = 1), the Address Counter Register, Next Address Register, Byte Counter, Next Byte Counter, and Control/Status Register are all used in the DMA transfer. By using the Next Address Register and the Next Byte Counter, the starting virtual address and data block size for an ensuing data transfer can be loaded while another transfer is occurring. This concurrence of operation reduces the amount of overhead between transfers.

The byte count for the block transfer is loaded into the Byte Counter. The L64853A automatically copies this byte count into the Next Byte Counter. The starting virtual address is written into the Address Counter Register, then each time a byte is transferred between the L64853A and the D channel, the byte count decrements by one and the address increments by one.

During the block transfer, the parameters for the next memory transfer can be set up. Software loads the Next Byte Counter with the byte count of the next block. (This step is optional if the byte count for the next block is the same as for the current block.) Software loads the Next Address Register with the starting address of the next block. When the Next Address Register is loaded, the L64853A sets the NA_LOADED bit in the CSR.

The Byte Counter expires upon completion of the block transfer. At this time, the L64853A sets the TC bit and clears the A_LOADED bit in the CSR. If TCI_DIS is equal to 0, then the L64853A generates an interrupt when the TC bit is set. When A_LOADED is cleared, the contents of the Next Address Register and Next Byte Counter are transferred to the Address Counter Register and Byte Counter, respectively. This transfer sets A_LOADED and clears NA_LOADED. The L64853A immediately services the next data block transfer.

If the Next Address Register is not loaded before the first block transfer completes, then DMA activity stops after the terminal count expires. The next data block transfer begins when an address is written into the Address Counter Register. As shown previously in Table 2.2, a write when EN_NEXT = 1 and A_LOADED = 0 loads the Address Counter Register and not the Next Address Register.

Refer to Section 4.6, "Programming Notes," for more information on handling the next block transfer.

4.3 D-Channel Cache Operation

This section discusses how the L64853A handles data in the D cache for DMA read and write operations. The D cache resides between the D channel and the SBus interfaces. For memory reads, the L64853A loads data from memory into the D cache before transferring the data to the D channel. For memory writes, the L64853A loads data into the D cache from the D channel before transferring the data to memory. Slave accesses to the D channel bypass the D cache. Refer to Section 3.2, "Internal Cache Memory," for more details on the D cache.

Memory Reads

The D-channel controller initiates memory reads. On a memory read, the L64853A first checks for the data in the D cache, using the address loaded in the Address Counter Register. If the data requested by the D channel are not in the D cache or are marked as invalid, then a cache miss occurs. The Least Recently Used cache line is filled with a four-word burst from memory containing the requested data. The four words are aligned on a four-word boundary, and the word containing the requested data is the first one to be read from memory. The L64853A marks each byte it reads into the D cache as valid. As soon as the requested data are written to the D cache, the L64853A transfers the data to the D channel, even if the entire D-cache line has not yet been filled. As soon as the L64853A transfers a byte to the D channel, it marks the byte as invalid.

If the D channel requests data that are in the D cache and are marked as valid, then a cache hit occurs. The L64853A transfers the data to the D channel and marks the byte or bytes as invalid.

If the least significant address bits of the requested data are 0100_2, the next four-word line in memory is on the same four-Kbyte page as the current cache line, and a cache hit occurs, then the L64853A performs a read-ahead operation. When a read-ahead occurs, the L64853A fills the cache line that does not contain the data requested by the D channel with the next line from memory. The read-ahead happens concurrently with the transfer of the requested data to the D channel.

If the least significant address bits of the requested data are 0100_2, the next four-word line in memory is on the same four-Kbyte page as the current cache line, and a cache miss occurs, the L64853A performs two four-word

bursts. The first burst operation fulfills the cache miss and the second burst operation performs the read-ahead operation. The L64853A does not assert the $\overline{\text{D\_ACK}}$ signal to acknowledge the D-channel request until the SBus burst transfer caused by the cache miss is complete.

Software can mark any data left in the cache after the read as invalid either by setting the FLUSH bit in the CSR or by writing a new address in the DMA Address Counter Register.

Memory Writes

DMA memory write operations write a byte or series of bytes from the D channel to the D cache. When the L64853A packs data into a cache line for a memory write, the L64853A sets the dirty bit for the cache line and marks each byte of data written into the cache line as valid. When the L64853A writes the last byte (byte 15) of that cache line, it queues the dirty bytes for draining to memory. The DMA Controller marks that cache line as the LRU and starts packing data into the other cache line (now the MRU). Once the L64853A transfers the data to memory, the L64853A clears the dirty and valid bits.

Any time the L64853A detects a D-channel interrupt, byte count expiration, or CPU slave access (read or write) to a D-channel-related register (Next Address, Next Byte Counter, Address Counter, Byte Counter, CSR, or any register in the D channel), the L64853A queues all valid bytes in the D-cache lines for draining to memory, unless a memory error has occurred. Hardware automatically drains queued bytes to memory. When the L64853A is draining dirty bytes, the DRAINING field in the CSR reads as 11_2.

4.4 Memory Errors and Interrupts

Interrupts to the L64853A occur as a result of memory access errors, interrupts from the D-channel controller, or completion of a DMA transfer. When one of these interrupts occurs, the L64853A generates an SBus interrupt request by asserting $\overline{\text{INTREQ}}$ provided the INT_EN bit in the CSR is set.

As mentioned in Section 4.1, a transfer from the D channel to memory consists of two separate transfers: one between the D channel and the L64853A, and one between the L64853A and memory. A memory, time-out, or protection error only occurs in a transfer between the L64853A and memory. When an error is detected, the L64853A sets the ERR_PEND bit in the CSR and generates an interrupt to the SBus (if INT_EN = 1). No

transfers occur between the L64853A and memory until the interrupt is cleared; however, up to seven additional words could be transferred from the D channel to the L64853A before the interrupt is generated. The interrupt is active until the ERR_PEND bit is cleared by either setting the FLUSH or RESET bits in the CSR.

Parity, memory time-out, or protection errors may occur on transfers from memory to the L64853A. Because of read-ahead operations, the L64853A reads data that the D channel has not requested. The L64853A may detect an error on this data. Whether the D channel requested that data or not, the L64853A does not write the data to the D channel. The L64853A sets the ERR_PEND bit and generates an interrupt to the SBus as long as INT_EN = 1. The interrupt is active until the ERR_PEND bit is cleared. To clear the ERR_PEND bit, set either the FLUSH or RESET bits in the CSR.

The L64853A sets the INT_PEND bit in the CSR when the D-channel controller asserts $\overline{\text{D\_IRQ}}$. The L64853A then generates an interrupt to the SBus as long as INT_EN = 1.

Similarly, the L64853A sets the INT_PEND bit when the Byte Counter expires provided that TCI_DIS = 0. The L64853A then generates an interrupt to the SBus as long as INT_EN = 1.

4.5 Read/Write Transactions

The L64853A D-channel interface supports DMA Master read/write and Slave register read/write operations. This section describes the signal sequences for each type of operation. Functional timing diagrams of the various operations are provided. This section is intended for hardware designers.

DMA Reads and Writes (Master Mode)

This subsection discusses DMA read and write operations for the D channel. Read operations are discussed first followed by a description of write operations.

D-Channel Read Operations

A DMA read consists of an SBus read from memory to the L64853A and a write from the L64853A to the D channel. A DMA read indicates a

transfer from memory to the D-channel device. The signal sequence for a D-channel read operation is as follows:

1. The D-channel controller asserts the Request signal (D_REQ) in order to request a DMA transfer. Since the WRITE bit in the CSR is LOW, the L64853A performs a read operation.
2. The L64853A compares the value in the Address Counter Register with the address tag on each cache line. If the values do not match or if the data at the requested location are marked as invalid, then a cache miss occurs.

   If a cache miss occurs, the L64853A asserts $\overline{BR}$ to request control of the SBus. When the SBus controller asserts $\overline{BG}$ to give control of the SBus to the L64853A, the L64853A outputs the virtual address of the data requested from the Address Counter Register onto the SBus data lines, D[31:0], along with the appropriate signals to indicate a four-word burst read.
3. The SBus controller's MMU translates this virtual address into a 28-bit physical address, and places the physical address on the SBus physical address lines, PA[27:0].
4. The L64853A reads the SBus data lines D[31:0] after each SBus clock where $\overline{ACK2}$ is asserted, until all four words are transferred to the D cache.
5. If a cache hit occurs or the L64853A has loaded the data into the D cache from the cache miss above, the L64853A asserts $\overline{D\_ACK}$ and $\overline{D\_WR}$, and begins transferring data to the D-channel peripheral.
6. The L64853A writes the requested data onto the D_D[7:0] lines from the D cache until all requested bytes are transferred.

The setup of the FASTER bit in the CSR and the $\overline{SLOW}$ pin determines the speed of the read operation. Figures 4.1 through 4.3 illustrate DMA read operations. Table 4.1 lists the figures with their respective settings for the FASTER bit and the $\overline{SLOW}$ pin.

Table 4.1 DMA Read Operation Settings

Figure Number	*FASTER bit*	*$\overline{SLOW}$ pin*
Figure 4.1	x[1]	0
Figure 4.2	0	1
Figure 4.3	1	1

1. x = don't care.

In all three figures, the data requested by the D channel are present in the D cache. If the data are not present, the L64853A delays asserting the $\overline{\text{D\_ACK}}$ and $\overline{\text{D\_WR}}$ strobes until an SBus memory read fills the cache (see the SBus Read Cycle).

Figure 4.1 DMA Read with $\overline{\text{FASTER}}$ = x, $\overline{\text{SLOW}}$ = 0

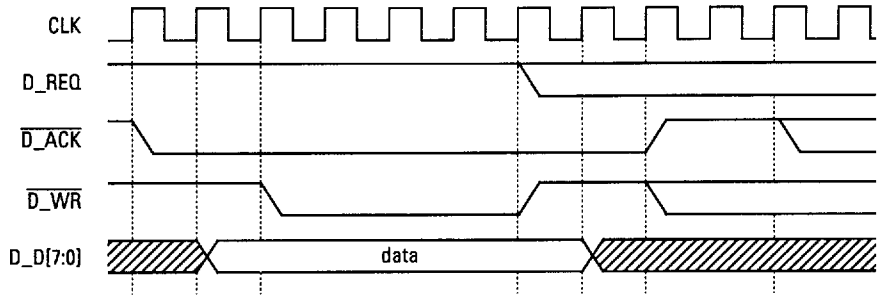


Figure 4.2 DMA Read with $\overline{\text{FASTER}}$ = 0, $\overline{\text{SLOW}}$ = 1

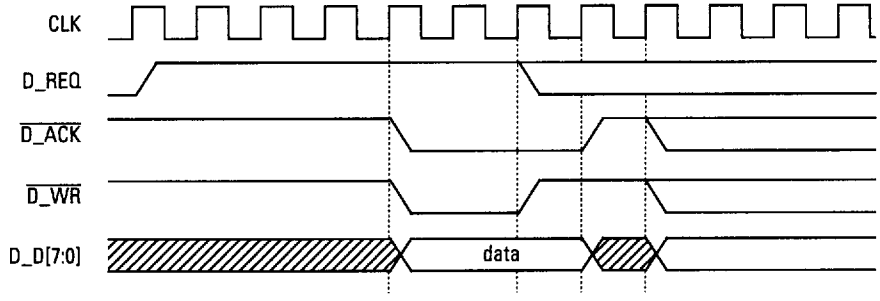
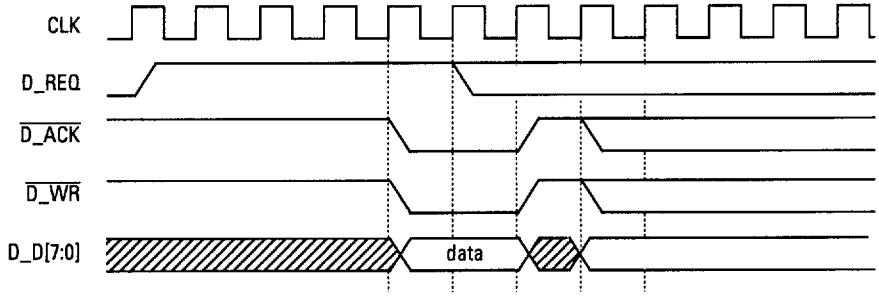


Figure 4.3 DMA Read with $\overline{\text{FASTER}}$ = 1, $\overline{\text{SLOW}}$ = 1



D-Channel Write Operations

A DMA write indicates a transfer from the D-channel device to memory. The write consists of a read from the D channel to the L64853A and an SBus write from the L64853A to memory. The signal sequence for a D-channel write operation is as follows:

1. The D-channel controller asserts the Request signal (D_REQ) in order to request a DMA transfer.
2. The L64853A asserts $\overline{\text{D\_ACK}}$ and $\overline{\text{D\_RD}}$ in response to the D channel's request.
3. The L64853A reads the data from the D_D[7:0] lines into the D cache until the last byte in the MRU cache line is loaded or an interrupt occurs.
4. The L64853A asserts $\overline{\text{BR}}$ in order to request control of the SBus. The SBus Controller asserts $\overline{\text{BG}}$ to grant control to the L64853A. During the SBus transfer, the L64853A can also be reading data into the LRU cache line.
5. The L64853A outputs a 32-bit virtual address from the Address Counter Register onto the SBus data lines, D[31:0].
6. The SBus Controller's MMU translates this virtual address into a 28-bit physical address, and places the physical address on the SBus physical address lines, PA[27:0].
7. The L64853A outputs the contents of the D cache onto the SBus data lines, D[31:0]. The memory controller transfers the data to the memory array.

The setup of the FASTER bit in the CSR and the $\overline{\text{SLOW}}$ pin determines the speed of the write operation. Figures 4.4 through 4.6 illustrate DMA write operations. Table 4.2 lists the figures with their respective settings for the FASTER bit and the $\overline{\text{SLOW}}$ pin.

Table 4.2 DMA Write Operation Settings

Figure Number	*FASTER bit*	*$\overline{\text{SLOW}}$ pin*
Figure 4.4	x[1]	0
Figure 4.5	0	1
Figure 4.6	1	1

1. x = don't care.

Figure 4.4
DMA Write with FASTER = x, SLOW = 0

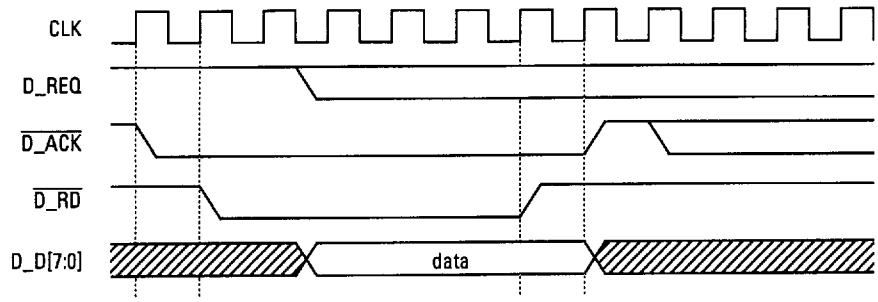


Figure 4.5
DMA Write with FASTER = 0, SLOW = 1

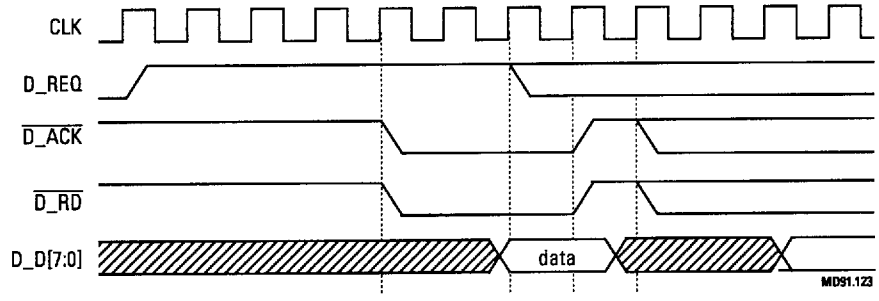
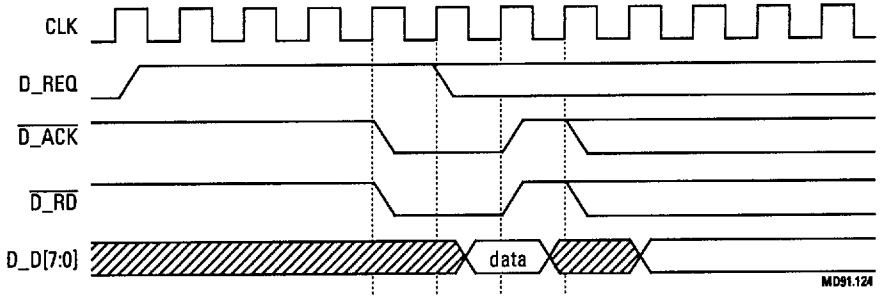


Figure 4.6
DMA Write with FASTER = 1, SLOW = 1



Register Reads and Writes (Slave Mode)

Register reads and writes allow the CPU to read status conditions and to set up data transfer parameters. The CPU is the Bus Master throughout these operations, and the L64853A is the Slave.

The following steps summarize a D-channel register read or write operation:

1. The CPU executes a LOAD or STORE instruction using the system-dependent address that asserts the L64853A's Slave Select signal ($\overline{SEL}$). The physical address is also driven onto PA[27:0] along with the appropriate states of RD, SIZ[2:0], and $\overline{AS}$.

2. The L64853A recognizes the read or write access ($\overline{SEL}$ is asserted), and interprets the value (10_2) on the physical address lines PA[X:Y]. This value indicates that an external D-channel controller register is the target of the read or write. Note that Chapter 8, "Applications," describes the register addresses for the external registers on the SPARCstation 1.
3. The D-channel data transfer size over the SBus is one byte. The L64853A passes the data between the SBus and the D-channel controller without any buffering in the D cache. The data are transferred over the D_D[7:0] lines to or from an eight-bit register in the D-channel controller. To accomplish this transfer, the L64853A also asserts various strobes ($\overline{D\_CS}$, $\overline{D\_RD}$, $\overline{D\_WR}$).
4. Finally, the L64853A asserts $\overline{ACK1}$, thus informing the SBus controller that the Slave-mode operation has completed.
5. If the D channel is busy transferring data to or from the D cache when the CPU selects that channel, the L64853A asserts Rerun Acknowledgment on the $\overline{ACK}$[2:0] pins to tell the CPU to rerun its operation.

Figure 4.7 illustrates a D-channel register read operation in fast mode ($\overline{SLOW}$ = 1). Figure 4.8 illustrates a D-channel register read operation in slow mode ($\overline{SLOW}$ = 0). The main difference between the two operations is that $\overline{D\_CS}$ and $\overline{D\_RD}$ are held LOW three clock cycles longer in slow mode than in fast mode.

Figure 4.7 Fast Mode D-Channel Register Read ($\overline{SLOW}$ = 1)

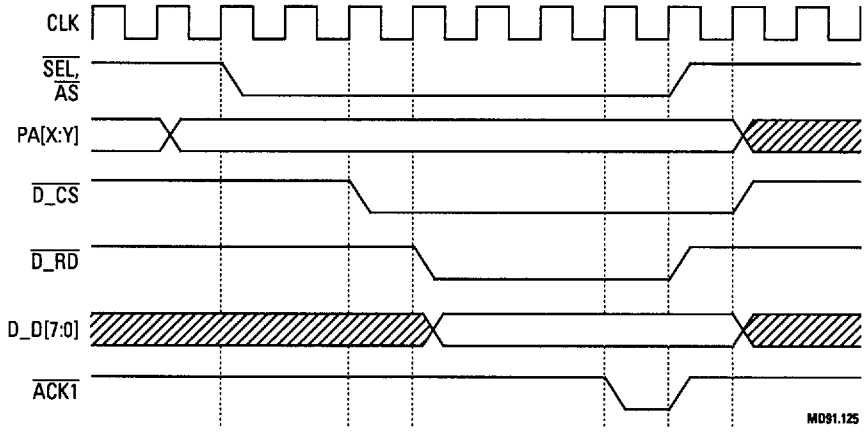


Figure 4.8 Slow Mode D-Channel Register Read ($\overline{SLOW}$ = 0)

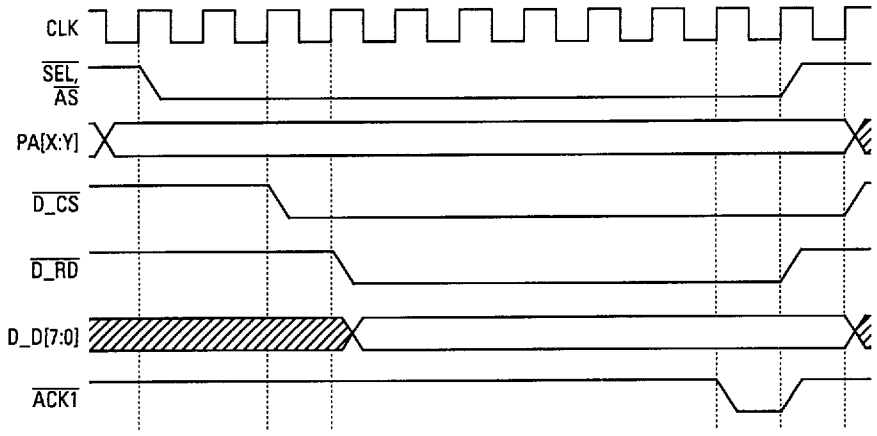
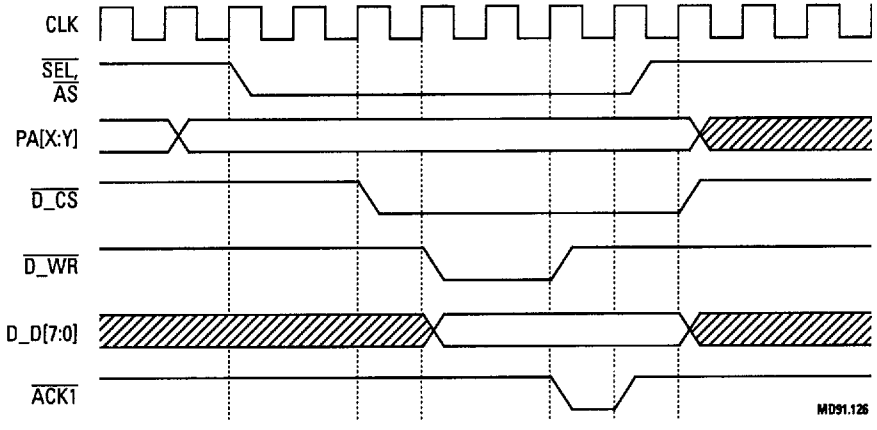


Figure 4.9 illustrates a D-channel register write operation in fast mode ($\overline{SLOW}$ = 1). Figure 4.10 illustrates a D-channel register write operation in slow mode ($\overline{SLOW}$ = 0). The main differences between the two operations are that $\overline{D\_CS}$ and $\overline{D\_WR}$ are extended by two clock cycles in slow mode.

Figure 4.9 Fast-Mode D-Channel Register Write ($\overline{SLOW}$ = 1)



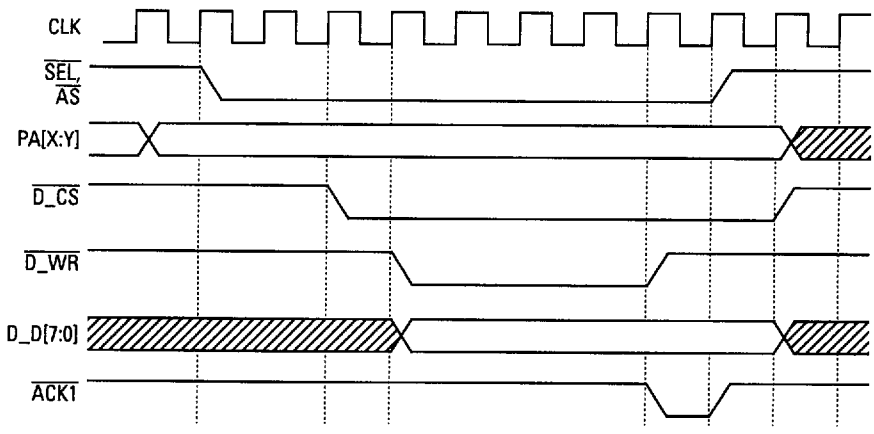


Figure 4.10 Slow-Mode D-Channel Register Write ($\overline{SLOW}$ = 0)

4.6 Programming Notes

This section provides some D-channel programming hints for the software programmer on the following topics:

- Setting Up, Suspending, and Stopping Transfers
- Using the Byte Counter with EN_NEXT = 0
- Using the Byte Counter with EN_NEXT = 1

Setting Up, Suspending, and Stopping Transfers

This subsection discusses procedures for setting up, suspending, and halting DMA transfers. The following procedure is recommended for correct operation of transfers between the L64853A and the D channel:

Step 1. Configure the CSR for the data transfer.

First clear the CSR to ensure no error bits are set. To clear the CSR, issue either a FLUSH or RESET command. Then program the WRITE, INT_EN, EN_CNT, FASTER, EN_DMA, EN_NEXT, and TCI_DIS bits for the operation.

Step 2. Load the transfer size into the Byte Counter.

Step 3. Load the starting address of the data transfer into the Address Counter Register. The L64853A is now ready to service any D-channel master transactions, provided that DMA_ON is set.

Step 4. Program the D-channel controller for the particular transfer and initiate the transfer.

To suspend transfers between the D channel and the L64853A, the device driver clears the EN_DMA bit in the CSR. The L64853A ignores new

DMA requests from the D channel when EN_DMA = 0. Note that the L64853A can still access SBus memory even with EN_DMA = 0. The device driver restarts the DMA transfer by setting EN_DMA.

One of three events halts the transfer:

- An error. The device driver must poll the D-channel controller for status.
- An interrupt. The device driver must provide service.
- Expiration of the Byte Counter.

Using the Byte Counter with EN_NEXT = 0

The following procedure is recommended for correct operation of the internal Byte Counter when EN_NEXT = 0:

Step 1. Load the CSR with:

INT_EN = EN_DMA = EN_CNT = 1
TCI_DIS = FLUSH = RESET = EN_NEXT = Reserved = 0
WRITE = ALE/$\overline{AS}$ = as read or required for the operation

Note that the SBus interrupt request signal ($\overline{INTREQ}$) is enabled (INT_EN = 1) and the Terminal Count flag is enabled (EN_CNT = 1 and TCI_DIS = 0). The expiration of the byte count sets the TC bit in the CSR and generates an interrupt on $\overline{INTREQ}$.

Step 2. Load the transfer size for the first block into the Byte Counter.

Step 3. Load the starting address of the first block into the Address Counter Register.

Once the address is loaded, the L64853A is ready to service any D-channel requests.

Step 4. Tell the D-channel controller to initiate the transfer.

Data are transferred until the Byte Counter expires. At this time, the L64853A halts DMA activity, sets the TC flag in the CSR, and generates an interrupt. The L64853A also clears A_LOADED, which in turn clears DMA_ON and prevents further DMA activity. DMA remains stopped, independent of the value of EN_DMA, until a new value is loaded into the Address Counter Register. The interrupt service routine should clear EN_DMA before writing a new address to the Address Counter Register.

To initiate another DMA operation:

- Issue a FLUSH command to clear the TC flag.
- Repeat the process from Step 1 above.

Using the Byte Counter with EN_NEXT = 1

This subsection discusses block transfer operations of the L64853A when EN_NEXT = 1. The stages discussed include the first multiple block transfer, interrupts, and subsequent block transfers.

The following example discusses how to implement two sequential multiple block transfers. The discussion covers setting up the L64853A for the first multiple block transfer, processing interrupts that occur between blocks, handling the last block, and setting up the L64853A for the next multiple block transfer.

First Multiple Block Transfer

The transfer of the first block is set up as follows:

Step 1. Load the CSR with:

INT_EN = EN_DMA = EN_CNT = EN_NEXT = 1
TCI_DIS = FLUSH = RESET = Reserved = 0
WRITE = ALE/$\overline{AS}$ = as read or required for the operation

Step 2. Load the transfer size for the first block into the Byte Counter.

The L64853A automatically copies this transfer size into the Next Byte Counter.

Step 3. Load the starting address of the first block into the Address Counter Register.

Step 4. Tell the D-channel controller to initiate the transfer.

Step 5. Set up the L64853A for the second block transfer. Load the transfer size of the second block into the Next Byte Counter. (This step is optional because the initial loading of the Byte Counter automatically loads the Next Byte Counter.) Load the starting address of the second block into the Next Address Register.

Always load the Next Address Register after the Next Byte Counter. Once the Next Address Register is loaded, the L64853A sets NA_LOADED = 1. At this time, an ensuing DMA block transfer begins immediately after the first transfer completes. If

the Next Byte Counter has not been loaded, the second transfer uses the same transfer size as in the first transfer.

Interrupt Service Routine

This section covers interrupt handling during multiple block transfers. Only interrupts generated by Byte Counter expiration are discussed here.

Step 1. Read the CSR for status information.

TC and DMA_ON should both be set. If DMA_ON = 0, then either the Next Address Register was not updated, or an error is pending (ERR_PEND = 1).

Step 2. If TC and DMA_ON are both set and the next block is not the last block to be transferred:

- Write a "1" to the TC bit in the CSR to clear the interrupt. Continue on to Step 3 below.

If TC and DMA_ON are both set and the peripheral instead of the Terminal Count generates the interrupt at the end of the next transfer (last block):

- Clear the TC bit and set the TC_DIS bit. Since the TC_DIS bit is set, the Byte Counter expiration does not generate an interrupt. The Next Address Register is not loaded, so the transfer stops upon completion of the next block transfer. In this case, Steps 3 and 4 are not implemented.

Step 3. Load the next transfer size into the Next Byte Counter. (This step is optional because the initial loading of the Byte Counter automatically loads the Next Byte Counter.)

Step 4. Load the starting address of the next block into the Next Address Register.

Subsequent Block Transfers

The following procedure is recommended for performing subsequent multiple block transfers:

Step 1. Write these settings to the CSR with the following:

INT_EN = EN_DMA = EN_CNT = EN_NEXT = 1
TCI_DIS = FLUSH = RESET = Reserved = 0
WRITE = ALE/$\overline{AS}$ = as read or required for the operation

These settings assume that the L64853A is in a known state (for example, A_LOADED = 0). If not, the write to the Byte Counter and Address Counter Register may go to the Next Registers.

Step 2. Write the byte count of the initial block into the Byte Counter.

Step 3. Write the starting address of the initial block into the Address Counter Register.

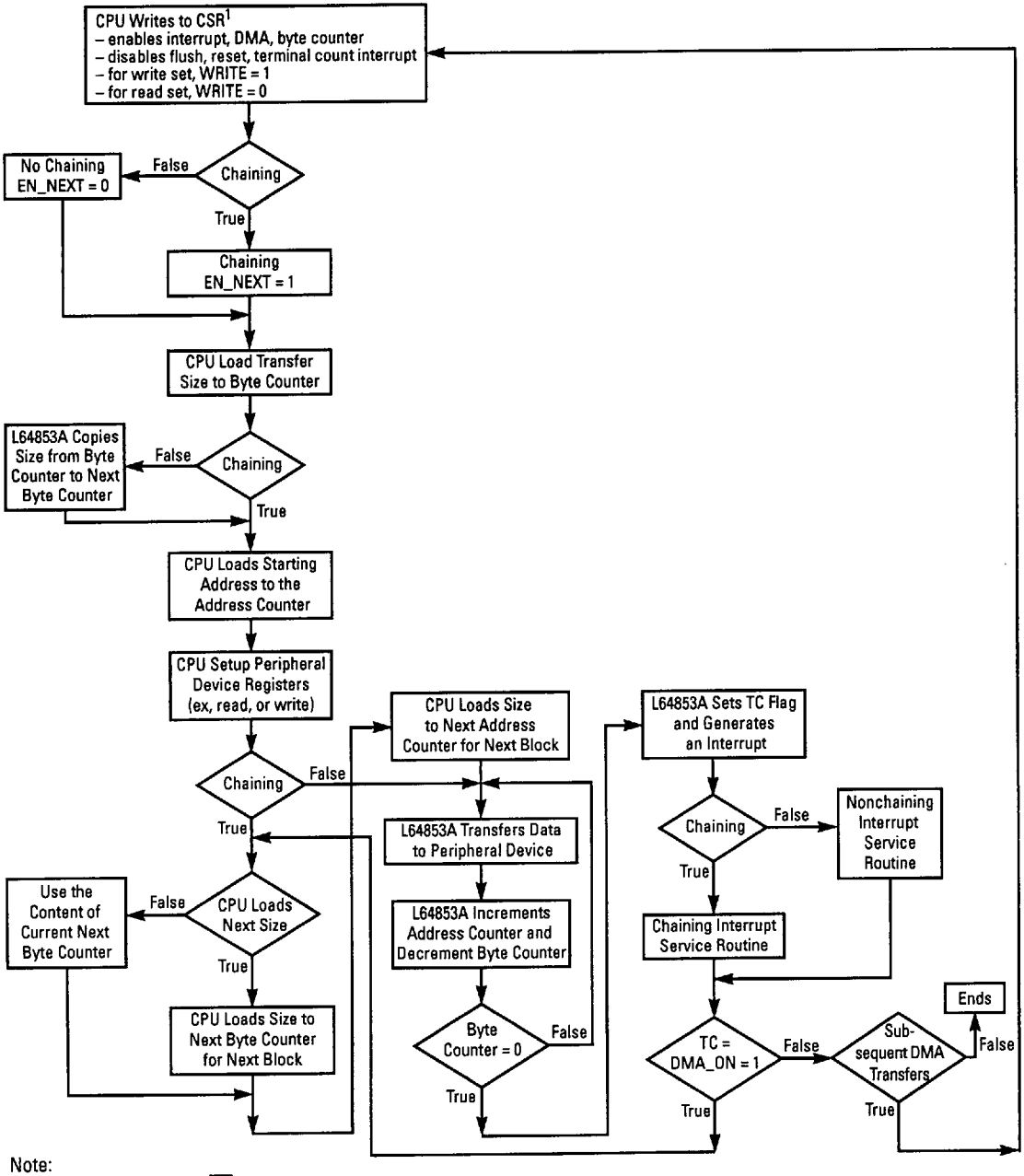
Step 4. Set up and start the D-channel controller.

Step 5. Write the byte count of the next block into the Next Byte Counter. This step is optional because the initial loading of the Byte Counter automatically loads the Next Byte Counter.

Step 6. Write the address of the next block into the Next Address Register.

Figure 4.11 illustrates the flow of the procedures described in this subsection.

Figure 4.11
Flow of DMA Transfers on the D Channel



Note:
1. Do not modify the ALE/$\overline{AS}$ bit.

Chapter 5
E-Channel Operation

Chapter 5 describes the organization and operation of the E channel for both programmers and hardware designers.

This chapter is divided into the following sections:

- General Description: This section presents the E channel in general terms.
- E-Channel Cache Operation: This section explains how data are transferred in the E cache for both memory read and write operations.
- Memory Errors: This section discusses how the L64853A detects and resolves memory errors.
- Read/Write Transactions: This section describes the signal sequences required for DMA read/write (Master mode) and DMA register read/write (Slave mode) operations.

5.1 General Description

The E channel is a 16- or 8-bit DMA control channel. The E channel provides an SBus interface for peripherals that contain their own DMA services. Therefore the E channel is suitable for Master-type peripherals—for example, Ethernet controllers or compression/expansion processors. The E channel is a higher priority channel than the D channel; when both channels request SBus access, the E channel gets access first.

Like the D channel, the E channel supports DMA memory read and write operations (Master mode) as well as external register read and write operations (Slave mode). Unlike the D channel, however, the E channel is programmed through the external registers.

A transfer from the E channel to the memory consists of two separate transfers: first one or more 16- or 8-bit transfers from the E channel to the E cache, and then a byte, halfword, word, or four-word transfer from the E cache to memory. Similarly, a transfer from memory to the E channel con-

sists of two transfers: a four-word burst from memory to the E cache followed by one or more 16- or 8-bit transfers from the E cache to the E channel. Note that all memory reads are in four-word bursts.

The E channel contains a 32-byte cache and a 24-bit Address Latch. For DMA transfers, the E channel uses a multiplexed address/data bus for address and data transfers. The Address Latch register stores the address of the data being transferred. The address is output as a 32-bit value where the upper 8 bits are driven to 0xFF. The data are packed into the E cache to allow efficient burst transfers over the SBus. See Section 3.2, "Internal Cache Memory," for more information on the internal cache setup and Section 5.2, "E-Channel Cache Operation," for E-cache-specific information.

During a DMA read or write operation, the E-channel controller provides the memory address for the read or write of the data. The E-channel controller transfers this 24-bit address over the E_A[23:16] and E_AD[15:0] lines to the Address Latch on the L64853A. The L64853A then appends a high-order, 8-bit value of 0xFF to this 24-bit address to create the 32-bit SBus virtual address, which it sends out on the SBus data lines D[31:0] during a DVMA Master cycle.

5.2 E-Channel Cache Operation

This section discusses how the L64853A controls the E cache for memory read and write operations. The E cache resides between the E-channel I/O and the SBus interface. For memory reads, the L64853A loads data from memory into the E cache before transferring the data to the E channel. For memory writes, data are loaded from the E channel into the E cache before being transferred to memory. Slave accesses to the E channel bypass the E cache. Refer to Section 3.2, "Internal Cache Memory," for more details on the E cache.

Memory Reads

The E-channel controller initiates memory reads. When the E channel requests a byte or halfword, the L64853A first checks for the data in the E cache. If the data requested by the E channel are in the cache and are marked as valid, a cache hit occurs. The L64853A transfers the data to the E channel and marks the data as invalid.

If the data requested by the E channel are not in the E cache or are marked as invalid, then a cache miss occurs. The L64853A fills the Least Recently Used (LRU) cache line with a four-word burst from memory that contains the requested data. The four words are aligned on a four-word boundary

and the word containing the requested data is the first one read from memory. The L64853A marks each byte it reads into the E cache as valid. As soon as the requested byte or halfword is written to the E cache, the L64853A transfers that data to the E channel, even if the entire E-cache line is not yet filled. The L64853A transfers the data as either bytes or halfwords, depending on the E_BYTE pin (0 = halfword; 1 = byte). As soon as the L64853A transfers a byte to the E channel, it marks the byte as invalid.

If the least significant address bits of the requested data are 0110_2, the next four-word line in memory is on the same four-Kbyte page as the current cache line, and a cache hit occurs, then the L64853A performs a read-ahead operation. When a read-ahead occurs, the L64853A fills the E-cache line that does not contain the requested data with the next line from memory; this prefetch maximizes the percentage of cache hits. The read-ahead happens concurrently with the transfer of the requested data to the E channel.

If the least significant address bits of the requested data are 0110_2, the next four-word line in memory is on the same four-Kbyte page as the current cache line, and a cache miss occurs, then the L64853A performs two four-word bursts. The first burst operation resolves the cache miss, and the second burst operation performs the read-ahead operation. The L64853A does not assert E_RDY to acknowledge the E-channel request until the SBus burst transfer caused by the cache miss is complete, and the data are available for transfer to the E channel.

A CPU Slave write to any of the internal registers of the AMD Am7990 Local Area Network Controller for Ethernet (LANCE) marks all bytes in the E-cache line as not valid. This write ensures that data fetched from memory cannot be used in a subsequent E-channel read, and provides an easy way of clearing the E-cache valid bits.

Memory Writes

Memory write operations write a series of halfwords or bytes from the E channel to the E cache. Data packing into a cache line for a memory write sets the dirty bit for the cache line. Additionally, as the L64853A writes each byte of data into the cache, it marks the byte as valid. The L64853A packs data into the E cache until either it loads the last byte (byte 15) of the cache line or it completes the data transfer ($\overline{\text{E\_HOLD}}$ is deasserted). When either of these two conditions are met, the L64853A queues all valid bytes

of that cache line for draining to memory. If the transfer is not complete, the L64853A packs the remaining data into the other cache line.

The L64853A always transfers the largest size of data possible. If the entire line contains valid bits, the L64853A performs a four-word burst operation. If only four consecutive bytes are valid, the L64853A performs a word transfer.

The L64853A detects the end of an E-channel transfer when the E-channel controller deasserts E_HOLD. At this time, the L64853A queues all valid bytes in the E cache for draining to memory. This operation ensures that data are not left in the E cache when the E channel interrupts the CPU.

If the E-channel controller writes to the L64853A at a variety of addresses without deasserting $\overline{\text{E\_HOLD}}$, both cache lines could contain valid bytes. If the E-channel controller attempts to write to an address whose tag does not match one of the E-cache tags, the L64853A automatically queues both cache lines for draining in order to make room for new data.

Note that if the E-channel controller makes more than one write to the same address without deasserting $\overline{\text{E\_HOLD}}$, the L64853A overwrites the data previously stored at that address in the E cache with the new data.

5.3 Memory Errors

As mentioned in Section 5.1, a data transfer consists of two separate occurrences: a transfer between the E channel and the L64853A followed by a transfer between the L64853A and memory. A memory error only occurs in a transfer between the L64853A and memory. Note that the L64853A reports memory errors on transfers to and from the E channel by forcing the E-channel controller to time-out. The L64853A forces the E channel to time-out by not asserting the $\overline{\text{E\_RDY}}$ signal after the L64853A has asserted $\overline{\text{E\_DAS}}$. When it times out, the E channel should generate an interrupt to the CPU.

To restart the E-channel DMA activity after an error, the device driver must write to the E channel's internal registers. This slave write to the E channel clears the LANCE_ERR bit in the CSR and resets the E cache, which marks all bytes as invalid.

Read Operations

Parity, memory time-out, or protection errors can occur on transfers from memory to the E channel. Because of read-ahead operations, the L64853A often reads data that the E channel has not requested. If the L64853A detects an error on the data it did not request, it does not assert E_RDY the next time the E-channel controller requests data, thus forcing the E-channel controller to time-out.

If an error occurs on a memory read that the E channel did request, the L64853A does not transfer the data to the E-channel controller, thus forcing the E-channel controller to time-out.

Write Operations

Memory time-out or protection errors can occur on transfers from the L64853A to memory. Each transfer from the E channel to the L64853A is completed before the data from that transfer are sent to memory. If an error occurs during the transfer to memory, the L64853A cannot force the E-channel controller to time-out until the E-channel controller tries to transfer more data. In the case of the LANCE Ethernet controller, the LANCE is guaranteed to request another transfer within 1.6 milliseconds of its last transfer as long as it is enabled; therefore 1.6 milliseconds is the maximum possible latency from the time the error occurs until the LANCE is forced to time-out. After the transfer that caused the error, no further transfers take place from the L64853A to memory; however, up to seven additional words can be transferred from the E channel to the L64853A before the E-channel controller is forced to time-out.

5.4 Read/Write Transactions

The L64853A E-channel interface supports DMA read/write and slave register read/write operations. This section describes the signal sequences for each type of operation and provides functional timing diagrams of the various operations. This section is intended for hardware designers.

DMA Reads and Writes (Master Mode)

This subsection discusses DMA read and write operations for the E channel. Read operations are discussed first, followed by a description of write operations.

E-Channel Read Operations

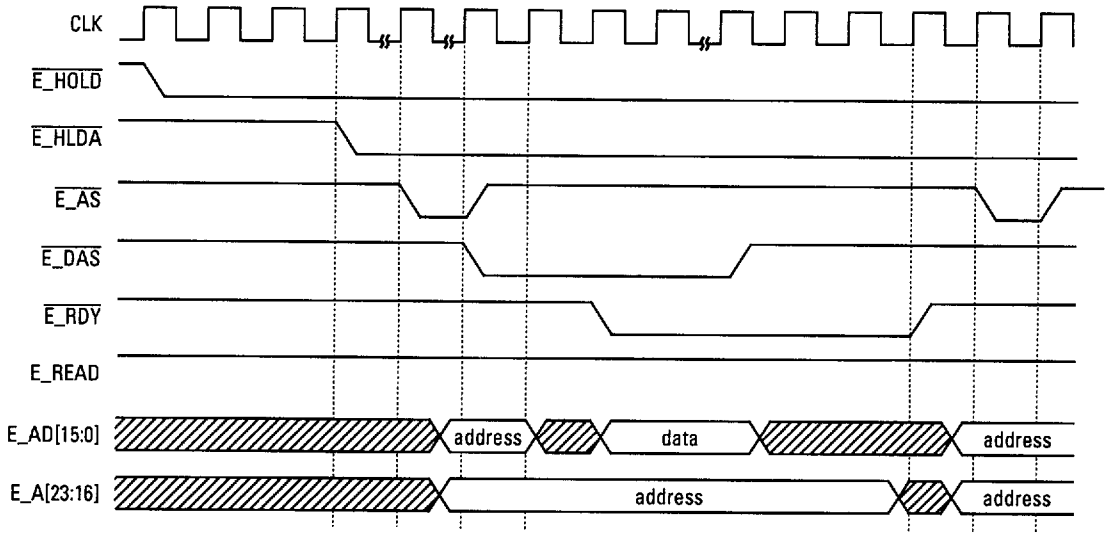
A DMA read indicates a transfer from memory to the E-channel device. The signal sequence for an E-channel read operation is as follows:

1. The E-channel controller asserts the Hold signal ($\overline{\text{E\_HOLD}}$) in order to request a DMA transfer.
2. The L64853A asserts the Hold Acknowledge signal ($\overline{\text{E\_HLDA}}$) in response to the E channel's request, then waits for the E channel to drive the lower 16 address bits onto E_AD[15:0], to drive the upper eight address bits onto E_A[23:16], and to strobe $\overline{\text{E\_AS}}$ to indicate when the address can be latched. E_READ is held HIGH to indicate a read operation.
3. The E channel then asserts $\overline{\text{E\_DAS}}$, and waits for the L64853A to assert $\overline{\text{E\_RDY}}$ indicating valid data on E_AD[15:0]. If the requested data are in the cache, proceed to Step 8.
4. If, however, the requested data are not in the cache, an SBus read is required. The L64853A requests the SBus by asserting $\overline{\text{BR}}$.
5. The SBus controller asserts $\overline{\text{BG}}$ to give control of the SBus to the L64853A. At this time, the L64853A outputs the address of the data requested onto the SBus data lines.
6. The SBus controller's MMU translates this virtual address into a 28-bit physical address on the SBus physical address lines PA[27:0].
7. The L64853A reads the SBus data lines D[31:0] when the data from the four-word burst comes back from memory and fills the E cache.
8. The L64853A asserts $\overline{\text{E\_RDY}}$ when valid data are on E_AD[15:0].

Figure 5.1 illustrates an E-channel DMA read operation with a cache hit. In Figure 5.1, $\overline{\text{E\_HLDA}}$ is asserted only when the interface is not busy. $\overline{\text{E\_HOLD}}$ remains asserted for burst mode accesses.

Steps 4 through 7 are not shown in Figure 5.1. These steps occur between the assertion of $\overline{\text{E\_DAS}}$ and the assertion of $\overline{\text{E\_RDY}}$. If a cache miss occurs, the L64853A asserts $\overline{\text{E\_RDY}}$ later than shown in Figure 5.1 because of the memory read that must occur to fill the cache.

Figure 5.1
E-Channel DMA Read
with Cache Hit



E-Channel Write Operations

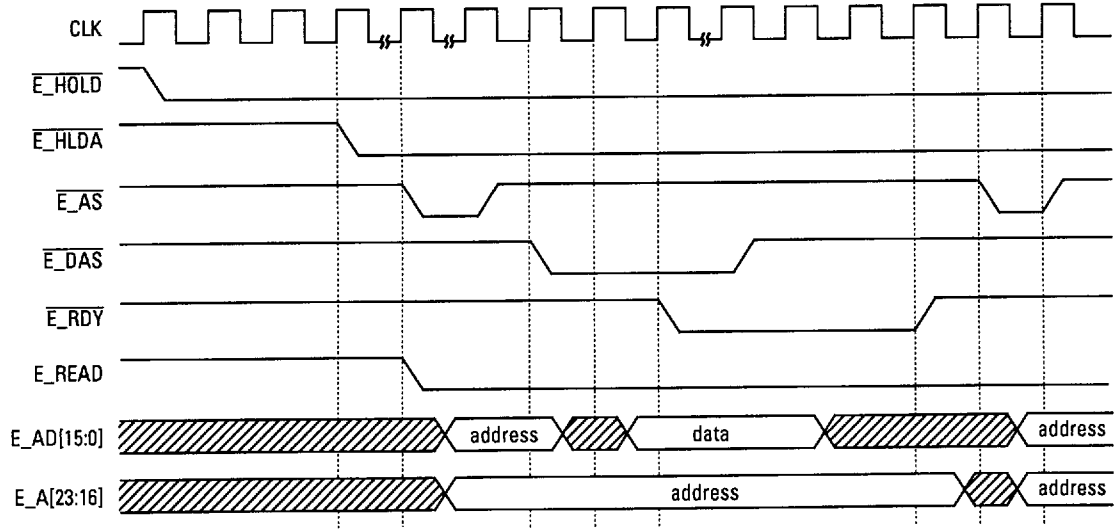
The basic steps for an E-channel write operation are listed below.

1. The E-channel controller asserts the Hold signal ($\overline{\text{E\_HOLD}}$) in order to request a DMA transfer.
2. The L64853A asserts the Hold Acknowledge signal ($\overline{\text{E\_HLDA}}$) in response to the E channel's request, thus prompting the E-channel controller to begin the data write operation.
3. E_READ goes LOW to indicate a write operation. The L64853A receives a 24-bit address from the E-channel controller on the E_AD[15:0] and E_A[23:16] lines. The address on E_AD[15:0] and E_A[23:16] is latched into the L64853A when $\overline{\text{E\_AS}}$ is strobed. This address serves as the lowest 24 bits of the memory address for the write.
4. The E-channel controller drives data on E_AD[15:0], and drives $\overline{\text{E\_DAS}}$ LOW until the L64853A asserts $\overline{\text{E\_RDY}}$ to indicate that the L64853A has loaded the data into the E cache. The E-channel controller then deasserts $\overline{\text{E\_DAS}}$. The L64853A responds by deasserting $\overline{\text{E\_RDY}}$.

5. Steps 3 and 4 are repeated until the L64853A writes to byte 15 in the cache line or the L64853A completes the transfer from the E channel ($\overline{\text{E\_HOLD}}$ is deasserted).
6. The L64853A asserts the Bus Request signal ($\overline{\text{BR}}$) in order to request control of the SBus. The SBus Controller asserts the Bus Grant signal ($\overline{\text{BG}}$) to grant control to the L64853A.
7. The L64853A outputs the 32-bit virtual address onto the SBus data lines D[31:0]. The 32-bit address is formed by appending 0xFF as the high-order byte to the 24-bit address.
8. The SBus Controller's MMU translates this virtual address into a 28-bit physical address on the SBus physical address lines PA[27:0].
9. The L64853A outputs the contents of the E cache onto the SBus data lines D[31:0]. The SBus controller transfers the data to the memory array.

Figure 5.2 illustrates an E-channel DMA burst-write operation. This figure assumes that a cache line is available for the write operation.

Figure 5.2
E-Channel DMA Write



Register Reads and Writes (Slave Mode)

Register reads and writes allow the CPU to read status conditions and to set up data transfer parameters. The CPU is the Bus Master throughout these operations, and the L64853A is a Slave.

The following steps summarize an E-channel register read or write operation:

1. The CPU executes a LOAD or STORE instruction using the system-dependent address that asserts the L64853A's Slave Select signal ($\overline{SEL}$).
2. As a result of the CPU's LOAD or STORE instruction, the SBus controller asserts the $\overline{SEL}$ signal into the L64853A, places a value on the physical address lines PA[27:0] and asserts various strobes that specify a halfword transfer and a read or write.
3. The L64853A recognizes the read or write access ($\overline{SEL}$ is asserted) and interprets the value (11_2) on the physical address lines PA[X:Y]. This value indicates an external E-channel controller register is the target of the read or write. Note that Chapter 8, "Applications," describes the register addresses for the external registers on the SPARCstation 1.
4. The E-channel data transfer size over the SBus is the halfword. The L64853A transfers the data over the E_AD[15:0] lines to or from a 16-bit register in the E-channel controller. To accomplish this transfer, the L64853A also asserts various strobes ($\overline{E\_CS}$, $\overline{E\_DAS}$, E_READ).
5. The 16-bit E-channel controller responds to a read cycle by driving the appropriate data onto E_AD[15:0] and asserting $\overline{E\_RDY}$ to indicate valid data. The controller responds to a write cycle by asserting $\overline{E\_RDY}$ when it is ready to store the data from E_AD[15:0].
6. Finally, the L64853A responds with a halfword acknowledgment on the $\overline{ACK}$[2:0] pins, thus informing the SBus controller that the Slave-mode operation is complete.
7. If the E channel is busy transferring data to or from the E cache when the CPU selects that channel, the DMA chip asserts Rerun Acknowledgment through the $\overline{ACK}$[2:0] outputs to inform the CPU to rerun its operation.

Figure 5.3 illustrates a register read operation for the E channel. Figure 5.4 illustrates a register write operation.

Figure 5.3
E-Channel Register Read Cycle

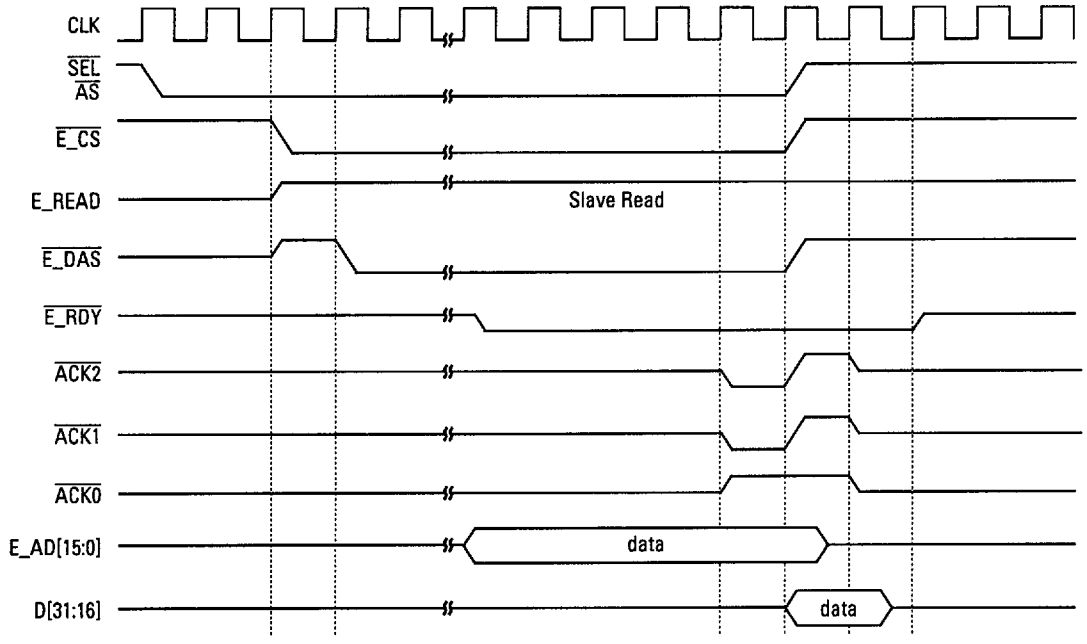
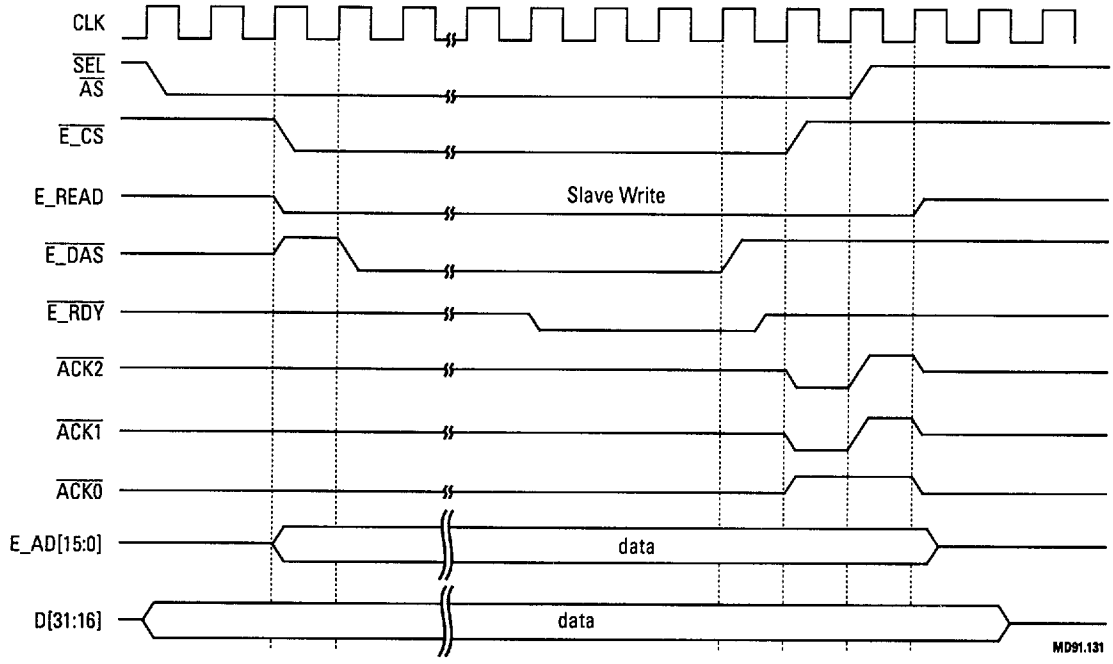


Figure 5.4
E-Channel Register Write Cycle

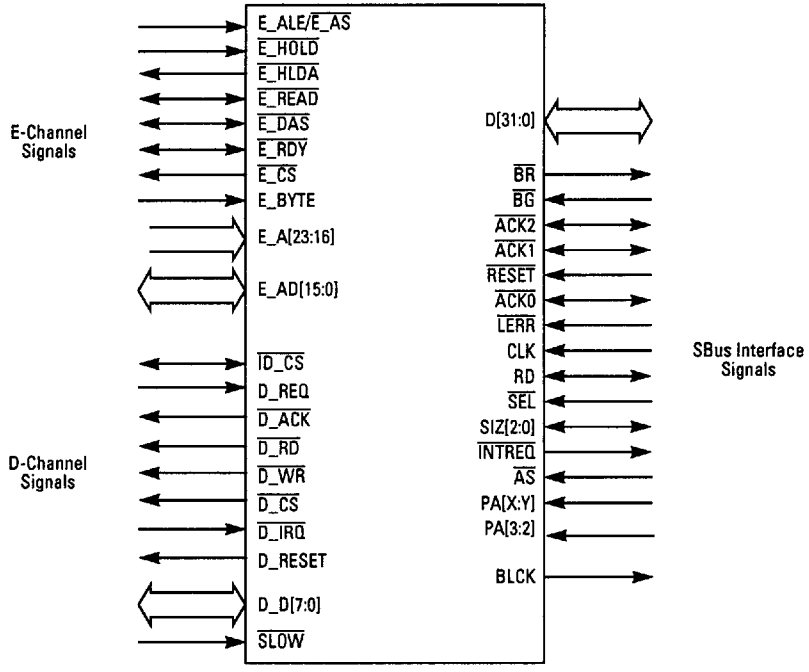


Chapter 6
Interface Description

This chapter describes the signals used by the L64853A. This information is useful primarily for hardware designers.

Figure 6.1 shows the logic diagram for the L64853A. For ease of reference, the signal descriptions are divided into four functional groups: D-channel, E-channel, SBus interface, and miscellaneous. Within each group, the signals are listed alphabetically by abbreviation. The SBus interface signal descriptions are minimal; refer to *The SBus Specification* from Sun Microsystems for additional information.

Figure 6.1 L64853A Logic Symbol



6.1 D-Channel Signals

The following D-channel signals define the interface to an external eight-bit controller chip.

$\overline{\text{D\_ACK}}$ **DMA Acknowledge** **Output**

DMA Acknowledge notifies the eight-bit controller that it is granted a DMA read or write cycle; that is, when the eight-bit controller can transfer data through the L64853A to or from the SBus. $\overline{\text{D\_ACK}}$ and $\overline{\text{D\_CS}}$ are never active at the same time.

$\overline{\text{D\_CS}}$ **DMA Chip Select** **Output**

DMA Chip Select selects the eight-bit controller as an I/O device; that is, the controller can read or write the eight-bit controller's internal registers. $\overline{\text{D\_CS}}$ and $\overline{\text{D\_ACK}}$ are never active at the same time.

D_D[7:0] **DMA Data Bus** **Bidirectional**

Data lines D_D[7:0] connect the eight-bit controller and the L64853A. During DMA transfers, D_D[7:0] send data to and from the D cache in the L64853A. During slave transfers, D_D[7:0] send data through internal L64853A logic from which it emerges as D[31:24] of the 32-bit L64853A data bus. These signals also serve as the input for external ID data when $\overline{\text{ID\_CS}}$ is asserted ($\overline{\text{D\_RD}}$ must also be asserted).

$\overline{\text{D\_IRQ}}$ **DMA Interrupt Request** **Input**

The eight-bit controller pulls DMA Interrupt Request LOW to signal an interrupt; for instance, when a data transfer completes, the eight-bit controller signals the completion to the L64853A.

$\overline{\text{D\_RD}}$ **DMA Read Strobe** **Output**

The L64853A asserts DMA Read Strobe to signal a read access to the eight-bit controller. The L64853A uses this signal both for reading eight-bit controller internal registers and for writing from the D-channel controller through the L64853A to the SBus (that is, during a DMA write). $\overline{\text{D\_CS}}$ or $\overline{\text{D\_ACK}}$ must also be active for the read to occur.

D_REQ **DMA Request** **Input**

The D-channel controller uses DMA Request to request permission to perform a DMA operation.

D_RESET **D-Channel Reset** **Output**

D_RESET resets the eight-bit controller.

$\overline{\text{D\_WR}}$ **DMA Write Strobe** **Output**

The L64853A uses DMA Write Strobe to signal a write access to the eight-bit controller. The L64853A uses this signal both for writing eight-bit controller internal registers and for reading from the SBus through the L64853A to the D-channel controller (that is, during a DMA read). $\overline{\text{D\_CS}}$ or $\overline{\text{D\_ACK}}$ must also be active for the write to occur.

$\overline{\text{ID\_CS}}$ **Secondary Device Select** **Bidirectional**

The $\overline{\text{ID\_CS}}$ line is pulled HIGH to specify the existence of an external PROM. The D_D[7:0] lines are the inputs for external ID data when $\overline{\text{ID\_CS}}$ is asserted ($\overline{\text{D\_RD}}$ must also be asserted). When $\overline{\text{ID\_CS}}$ is tied LOW, the L64853A ID is returned. For more details about such a PROM, see *The SBus Specification.*

$\overline{\text{SLOW}}$ **Fast/Slow DMA Acknowledge Cycles** **Input**

The $\overline{\text{SLOW}}$ signal specifies either fast or slow mode, to control the transfer speed on the D channel. Use the Fast mode with the ESP SCSI controller. Because of an internal pull-up resistor, this signal requires no external connection for Fast mode.

6.2 E-Channel Signals

The following E-channel signals define an interface to an external 16-bit E-channel controller chip:

E_A[23:16] **E-Channel Address** **Input**

E_A[23:16] contain the upper eight bits of a virtual address. These bits are combined with the E_AD[15:0] lines to form the 24 bits of a virtual address.

E_AD[15:0] **E-Channel Address/Data Bus** **Bidirectional**

E_AD[15:0] are multiplexed address/data bus lines connecting the E-channel controller and the L64853A. These lines send data to and receive data from the E cache or send data through internal L64853A logic, from which the data emerge as D[31:16] of the 32-bit L64853A data bus. These lines also send the lower 16 bits of virtual addresses to the address latch in the L64853A.

E_ALE/$\overline{\text{E\_AS}}$ **E-Channel Address Latch Enable/Address Strobe** **Input**

The setting of the ALE/$\overline{\text{AS}}$ bit in the Control/Status Register determines the polarity of the E_ALE/$\overline{\text{E\_AS}}$ pin. If the ALE/$\overline{\text{AS}}$ bit is HIGH, this signal is E_ALE (active HIGH). If the ALE/$\overline{\text{AS}}$ bit is LOW, this signal is $\overline{\text{E\_AS}}$ (active LOW). The E-channel controller drives $\overline{\text{E\_AS}}$ LOW to signal an address transfer. The

address on E_AD[15:0] is latched into the L64853A when $\overline{\text{E\_AS}}$ makes a LOW-to-HIGH transition.

E_BYTE **E-Channel Byte Marker** **Input**

When HIGH, E_BYTE indicates that a byte transfer is in progress on the data lines (E_AD[15:0]). When LOW, E_BYTE indicates that a halfword transfer is occurring. The lowest bit of the address determines the byte being transferred, if E_BYTE is set. The following table shows which byte is transferred.

E_BYTE	*E_AD0*	*Transferred Byte*
1	0	E[15:8]
1	1	E[7:0]

$\overline{\text{E\_CS}}$ **E-Channel Chip Select** **Output**

$\overline{\text{E\_CS}}$ selects the E-channel controller as an I/O device so that the E-channel controller can read or write its own internal registers.

$\overline{\text{E\_DAS}}$ **E-Channel Data Strobe** **Bidirectional**

When the E channel is Master, $\overline{\text{E\_DAS}}$ is an input driven by the E channel. $\overline{\text{E\_DAS}}$ is driven HIGH during the address portion of a memory access and driven LOW during the data portion. Data on E_AD[15:0] are latched into the L64853A while $\overline{\text{E\_DAS}}$ and $\overline{\text{E\_RDY}}$ are LOW. For slave accesses, $\overline{\text{E\_DAS}}$ is an output driven by the L64853A.

$\overline{\text{E\_HLDA}}$ **E-Channel Hold Acknowledge** **Output**

The L64853A uses $\overline{\text{E\_HLDA}}$ to acknowledge a prior $\overline{\text{E\_HOLD}}$ from the E-channel controller, and thus to signal the controller to proceed with a DMA transfer. While $\overline{\text{E\_HLDA}}$ is asserted, the L64853A does not allow slave transfers to the E channel and forces the Master to rerun the slave transfer. Once $\overline{\text{E\_HOLD}}$ goes inactive (HIGH), so does $\overline{\text{E\_HLDA}}$.

$\overline{\text{E\_HOLD}}$ **E-Channel Hold** **Input**

The E-channel controller asserts $\overline{\text{E\_HOLD}}$ when the controller requires access to memory. $\overline{\text{E\_HOLD}}$ is held LOW for the entire ensuing bus transaction. Once $\overline{\text{E\_HOLD}}$ goes inactive (HIGH), so does $\overline{\text{E\_HLDA}}$.

$\overline{\text{E\_RDY}}$ **E-Channel Ready** **Bidirectional**

When the E channel is a Bus Master, $\overline{\text{E\_RDY}}$ is an output. The L64853A asserts $\overline{\text{E\_RDY}}$ to indicate that either the L64853A is

ready to accept data for a memory write or that the L64853A has data available for a memory read.

When the E channel is a Bus Slave, $\overline{\text{E\_RDY}}$ is an input. The E-channel controller asserts $\overline{\text{E\_RDY}}$ to indicate that either the E channel has data ready for a read operation or the E channel is ready to read data for a write operation. $\overline{\text{E\_RDY}}$ in this case is asserted in response to the assertion of $\overline{\text{E\_DAS}}$. $\overline{\text{E\_RDY}}$ goes HIGH after $\overline{\text{E\_DAS}}$ is driven HIGH.

E_READ **E-Channel Read** **Bidirectional**

E_READ indicates the type of operation to be performed in the current bus cycle. The L64853A drives E_READ in Slave mode. When HIGH, E_READ indicates a read from the E channel to the L64853A. When LOW, E_READ indicates a write from the L64853A to the E channel.

The E channel drives E_READ in Master mode. A HIGH indicates a read from the L64853A. A LOW indicates a write to the L64853A.

6.3 SBus Interface Signals

These SBus signals provide the interface between the L64853A and other system components:

$\overline{\text{ACK}}$[2:0] **SBus Acknowledge** **Bidirectional**

During slave cycles, the SBus Acknowledge signals provide information on the current slave cycle as indicated in the following table.

ACK2	*ACK1*	*ACK0*	*Function*
0	0	0	Reserved
0	0	1	16-bit Slave cycle acknowledgment
0	1	0	Reserved
0	1	1	32-bit Slave cycle acknowledgment
1	0	0	Rerun acknowledgment
1	0	1	Eight-bit Slave cycle acknowledgment
1	1	0	Error
1	1	1	Insert wait states

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

A LOW on $\overline{\text{AS}}$ indicates the address on the physical address lines is valid and marks the beginning of an SBus Slave cycle.

$\overline{\text{BG}}$	**SBus Bus Grant**	**Input**
	A LOW on the SBus Bus Grant signal indicates that the SBus controller has granted control of the SBus to the L64853A.	
$\overline{\text{BR}}$	**SBus Bus Request**	**Output**
	To perform a Master cycle, the L64853A first drives $\overline{\text{BR}}$ LOW to request control of the SBus.	
CLK	**SBus Clock**	**Input**
	The SBus Clock signal controls internal operations and rates of data transfer. CLK is usually derived from the Master system clock or an associated CPU clock.	
D[31:0]	**SBus Data Bus**	**Bidirectional**
	D[31:0] contain the 32-bit SBus data.	
$\overline{\text{INTREQ}}$	**SBus Interrupt Request**	**Output**
	SBus Interrupt Request is the SBus interrupt line. $\overline{\text{INTREQ}}$ is activated when INT_EN is set and an external interrupt or any of various L64853A error conditions occurs.	
$\overline{\text{LERR}}$	**SBus Late Error**	**Input**
	The assertion of $\overline{\text{LERR}}$ indicates that an error occurred during a preceding data transfer, even though the Slave issued a byte, halfword, or word acknowledgment.	
PA[X:Y]	**SBus Physical Address**	**Input**
	These two physical address lines decode Slave cycle addresses.	
PA[3:2]	**SBus Physical Address**	**Input**
	The PA[3:2] lines contain two SBus physical address bits.	
RD	**SBus Read/Write**	**Bidirectional**
	RD indicates the direction of data transfers on the SBus from the perspective of the SBus Master (read = HIGH, write = LOW).	
$\overline{\text{RESET}}$	**SBus Reset**	**Input**
	SBus Reset is the SBus reset line.	
$\overline{\text{SEL}}$	**SBus Select**	**Input**
	SBus Select is the slave select line for the L64853A.	

SIZ[2:0] **SBus Transfer Size** **Bidirectional**

The SIZ[2:0] lines specify the size of the data transfer on the SBus as shown in the following table.

SIZ2	*SIZ1*	*SIZ0*	*Transfer Size*
0	0	0	Word
0	0	1	Byte
0	1	0	Halfword
0	1	1	Reserved
1	0	0	Four-word burst (16 bytes)
1	0	1	Eight-word burst (32 bytes)
1	1	0	16-word burst (64 bytes)
1	1	1	Two-word burst (8 bytes)

6.4 Miscellaneous Signals

BCLK **Buffered SBus Clock** **Output**

BCLK is a buffered version of the SBus Clock signal, CLK. Use it to prevent excessive loading of CLK. Note that BCLK is skewed from CLK (10.6 ns maximum).

V_{DD} **Power** **Input**

There are seven +5 volt power lines.

V_{SS} **Ground** **Input**

There are nine ground pins.

Chapter 7
Specifications

This chapter provides AC, DC, environmental, and mechanical specifications for the L64853A. Hardware designers will find the information in this chapter useful.

7.1 AC Characteristics

During AC testing, HIGH inputs are driven at 3.0 V, and LOW inputs are driven at 0 V. For transitions between HIGH, LOW, and invalid states, timing measurements are made at 1.5 V, as shown in Figure 7.1. The test load, C_L, for each output signal is given in Table 7.1.

For 3-state outputs, timing measurements are made from the point at which the output turns ON or OFF. An output is ON when its voltage is greater than 3.5 V or less than 1.5 V. An output is OFF when its voltage is less than V_{DD} – 1.5 V or greater than 1.5 V, as shown in Figure 7.2.

Figure 7.1 AC Test Load and Waveform for Standard Outputs

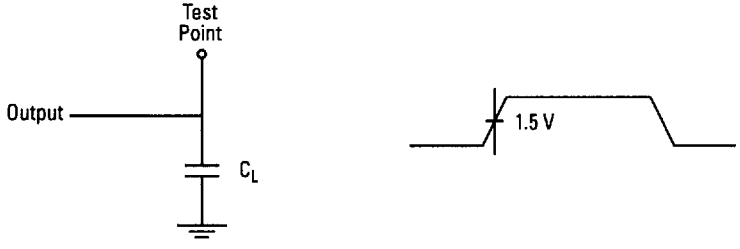


Figure 7.2 AC Test Load and Waveform for 3-State Outputs

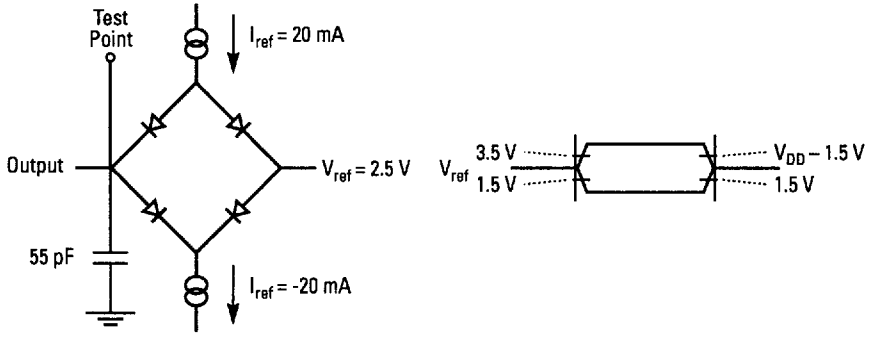


Table 7.1 contains the AC characteristics for the L64853A. The test conditions for the AC timing values are $V_{DD} = 5\ V \pm 5\%$ and $T_A = 0\ °C$ to $70\ °C$. Figure 7.3 through Figure 7.12 illustrate the AC timing waveforms. Note that in Table 7.1, CLK is the SBus clock.

Table 7.1
AC Characteristics

Parameter	*Description*	*25 MHz*			C_L *(pF)*
		Min	*Max*	*Units*	*Max*
1. t_{CY}	SBus Clock Period	40	60	ns	–
2. t_{SASIS}	$\overline{AS}$, $\overline{SEL}$ Setup to CLK	15	–	ns	–
3. t_{SASIH}	$\overline{AS}$, $\overline{SEL}$ Hold from CLK	0	–	ns	–
4. t_{SPAIS}	PA[X:Y], PA[3:2], SIZ[2:0] Setup to CLK	15	–	ns	–
5. t_{SPAIH}	PA[X:Y], PA[3:2], SIZ[2:0] Hold from CLK	0	–	ns	–
6. t_{SAKIS}	$\overline{ACK}$[2:0] Setup to CLK	15	–	ns	–
7. t_{SAKIH}	$\overline{ACK}$[2:0] Hold from CLK	0	–	ns	–
8. t_{SLEIS}	$\overline{LERR}$ Setup to CLK	15	–	ns	–
9. t_{SLEIH}	$\overline{LERR}$ Hold from CLK	0	–	ns	–
10. t_{SBDIS}	D[31:0] Setup to CLK	15	–	ns	–
11. t_{SBDIH}	D[31:0] Hold from CLK	0	–	ns	–
12. t_{SRDIS}	RD Setup to CLK	15	–	ns	–
13. t_{SRDIH}	RD Hold from CLK	0	–	ns	–
14. t_{SBGIS}	$\overline{BG}$ Setup to CLK	15	–	ns	–
15. t_{SBGIH}	$\overline{BG}$ Hold from CLK	0	–	ns	–
16. t_{SBROD}	CLK to $\overline{BR}$ Active	2.5	22.5	ns	20
17. t_{SBROH}	CLK to $\overline{BR}$ Inactive	2.5	22.5	ns	20
18. t_{SRDOD}	CLK to RD, SIZ[2:0] Valid	2.5	22.5	ns	100
19. t_{SRDOAZ}	CLK to RD, SIZ[2:0] 3-State	0	25.1	ns	100
20. t_{SAKOD}	CLK to $\overline{ACK}$[2:0] Valid	2.5	22.5	ns	100
21. t_{SAKOAZ}	CLK to $\overline{ACK}$[2:0] 3-State	0	25	ns	100
22. t_{SBDOD}	CLK to D[31:0] Valid	2.5	22.5	ns	100
23. t_{SBDOAZ}	CLK to D[31:0] 3-State	0	27.4	ns	100
24. t_{DCSOD}	CLK to $\overline{D\_CS}$, $\overline{ID\_CS}$ Active (Register Access)	0	15	ns	15
25. t_{DCSOH}	CLK to $\overline{D\_CS}$, $\overline{ID\_CS}$ Inactive (Register Access)	0	15	ns	15
26. t_{DRDOD}	CLK to $\overline{D\_RD}$ Active (Register Read)	0	20	ns	15

(Sheet 1 of 3)

Table 7.1 (Cont.)
AC Characteristics

Parameter	*Description*	*25 MHz* *Min*	*Max*	*Units*	C_L *(pF)* *Max*
27. t_{DRDOH}	CLK to $\overline{D\_RD}$ Inactive (Register Read)	0	20	ns	15
28. t_{DDIS}	D_D[7:0] Valid Setup to CLK (Register Read)	15	–	ns	–
29. t_{DRDDD}	D_RD Inactive to D_D[7:0] Invalid (Register Read)	0	–	ns	–
30. t_{DWROD}	CLK to $\overline{D\_WR}$ Active (Register Write)	0	20	ns	15
31. t_{DWROH}	CLK to $\overline{D\_WR}$ Inactive (Register Write)	0	20	ns	15
32. t_{DD7OD}	CLK to D_D[7:0] Valid (Register Write)	0	20	ns	15
33. t_{DD7OAZ}	CLK to D_D[7:0] 3-State (Register Write)	0	20	ns	15
34. t_{DREOD}	CLK to D_RESET Active	0	20	ns	75
35. t_{DREOH}	CLK to D_RESET Inactive	0	20	ns	75
36. t_{DQIS}	D_REQ Setup to CLK[1]	15	–	ns	–
37. t_{DAKDQ}	$\overline{D\_ACK}$ Active to D_REQ Inactive	0	–	ns	–
38. t_{DQIIS}	D_REQ Inactive Setup to CLK	20	–	ns	–
39. t_{DAKOD}	CLK to $\overline{D\_ACK}$ Active	0	20	ns	15
40. t_{DAKOH}	CLK to $\overline{D\_ACK}$ Inactive	0	20	ns	15
41. t_{DRDODD}	CLK to $\overline{D\_RD}$ Active (DMA Cycle)	0	20	ns	15
42. t_{DRDOHD}	CLK to $\overline{D\_RD}$ Inactive (DMA Cycle)	0	20	ns	15
43. t_{DD7IS}	D_D[7:0] Valid Setup to CLK (DMA Write)	15	–	ns	–
44. t_{DRDDD7}	$\overline{D\_RD}$ Inactive to D_D[7:0] Invalid (DMA Cycle)	0	–	ns	–
45. t_{DWRODD}	CLK to $\overline{D\_WR}$ Active (DMA Cycle)	0	20	ns	15
46. t_{DWROHD}	CLK to $\overline{D\_WR}$ Inactive (DMA Cycle)	0	20	ns	15
47. t_{DD7IS}	CLK to D_D[7:0] Valid (DMA Cycle)	0	40	ns	15
48. t_{DWRDD7}	$\overline{D\_WR}$ Inactive to D_D[7:0] Invalid				
	(DMA, SLOW = 5 V, FASTER = 1)	4	–	ns	15
	(DMA, SLOW = 5 V, FASTER = 0)	1	–	CLK	15
	(DMA, SLOW = 0 V)	1	–	CLK	15
	$\overline{D\_WR}$ Inactive to D_D[7:0] 3-State	–	2	CLK	15
49. t_{ECSOD}	CLK to $\overline{E\_CS}$ Active (Register Access)	0	20	ns	15
50. t_{ECSOH}	CLK to $\overline{E\_CS}$ Inactive (Register Access)	0	20	ns	15
51. t_{ERDOD}	CLK to E_READ, $\overline{E\_DAS}$ Valid	0	20	ns	15
52. t_{ERDOAZ}	CLK to E_READ, $\overline{E\_DAS}$ 3-State	0	20	ns	15
53. t_{EDIS}	CLK to $\overline{E\_DAS}$ Active	0	20	ns	15

(Sheet 2 of 3)

Table 7.1 (Cont.)
AC Characteristics

Parameter	*Description*	*25 MHz* *Min*	*Max*	*Units*	*C_L (pF)* *Max*
54. t_{EDIH}	CLK to $\overline{E\_DAS}$ Inactive	0	20	ns	15
55. t_{ERDIS}	$\overline{E\_RDY}$ Setup to CLK[1]	15	–	ns	–
56. t_{ERDIH}	$\overline{E\_RDY}$ Hold from CLK[1]	2.5	–	ns	–
57. t_{EARD}	E_AD[15:0] Setup to $\overline{E\_RDY}$ Active (Read)	0	–	ns	–
58. t_{EDEA}	$\overline{E\_DAS}$ Inactive to E_AD[15:0] Invalid (Read)	0	–	ns	–
59. t_{EAOD}	CLK to E_AD[15:0] Valid (Write)	0	25	ns	15
60. t_{EAOAZ}	CLK to E_AD[15:0] 3-State (Write)	0	25	ns	15
61. t_{EHIS}	$\overline{E\_HOLD}$ Setup to CLK[1]	15	–	ns	–
62. t_{EHIH}	$\overline{E\_HOLD}$ Hold from CLK[1]	2.5	–	ns	–
63. t_{EHAOD}	CLK to $\overline{E\_HLDA}$ Active	0	20	ns	15
64. t_{EHAOH}	CLK to $\overline{E\_HLDA}$ Inactive	0	20	ns	15
65. t_{EAS}	E_ALE/$\overline{E\_AS}$ Pulse Width	1	–	CLK	–
66. t_{EASIS}	E_ALE/$\overline{E\_AS}$ Setup to CLK[1]	15	–	ns	–
67. t_{EASIH}	E_ALE/$\overline{E\_AS}$ Hold from CLK[1]	2.5	–	ns	–
68. t_{EAEAS}	E_READ, E_BYTE, E_A[23:16] Setup to E_ALE/$\overline{E\_AS}$ Active	0	–	ns	–
69. t_{EDIS}	$\overline{E\_DAS}$ Setup to CLK[1]	15	–	ns	–
70. t_{EDIH}	$\overline{E\_DAS}$ Hold from CLK[1]	2.5	–	ns	–
71. t_{EDEA}	$\overline{E\_DAS}$ Active to E_AD[15:0] Valid (DMA Write)	–	1	CLK	–
72. t_{EDEAZ}	$\overline{E\_DAS}$ Inactive to E_READ, E_BYTE, E_A[23:16] Invalid	0	–	ns	–
73. t_{EROD}	CLK to $\overline{E\_RDY}$ Active	0	20	ns	15
74. t_{EROH}	CLK to $\overline{E\_RDY}$ Inactive	0	20	ns	15
75. t_{EAIS}	E_AD[15:0] Setup to E_ALE/$\overline{E\_AS}$ Inactive (DMA Read)	15	–	ns	–
76. t_{EAIH}	E_AD[15:0] Hold from E_ALE/$\overline{E\_AS}$ Inactive	10	–	ns	–
77. t_{EAOD}	CLK to E_AD[15:0] Valid	–	25	ns	15
78. t_{EDEAR}	$\overline{E\_DAS}$ Inactive to E_AD[15:0] Invalid (DMA Read)	0	–	ns	–
79. t_{EDEAW+}	$\overline{E\_DAS}$ Inactive to E_AD[15:0] Invalid (DMA Write)	1	–	CLK	–
80. t_{BCLK}	CLK Rising Edge to BCLK Rising Edge		10.6	ns	–

(Sheet 3 of 3)

1. If this timing parameter is not met, the L64853A may sample an incorrect value on the rising edge of CLK. The L64853A samples the correct value either on this clock edge or the next one.

Figure 7.3
SBus Input Signals

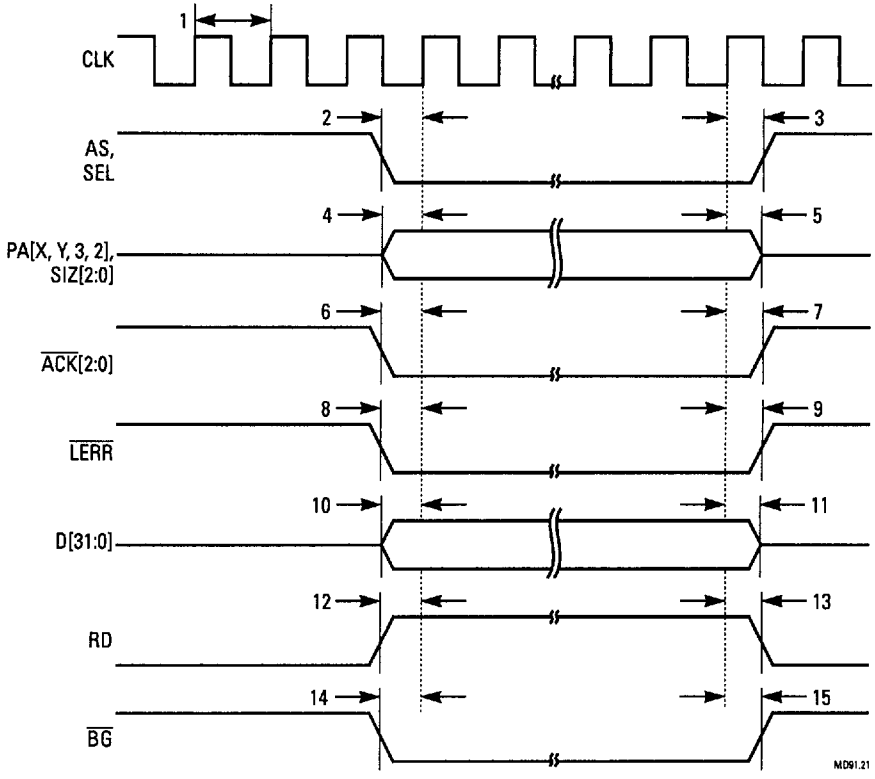


Figure 7.4
SBus Output Signals

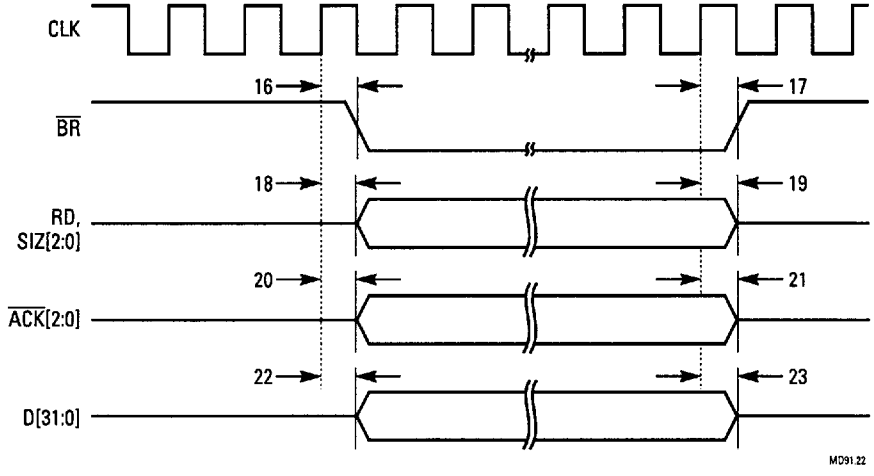
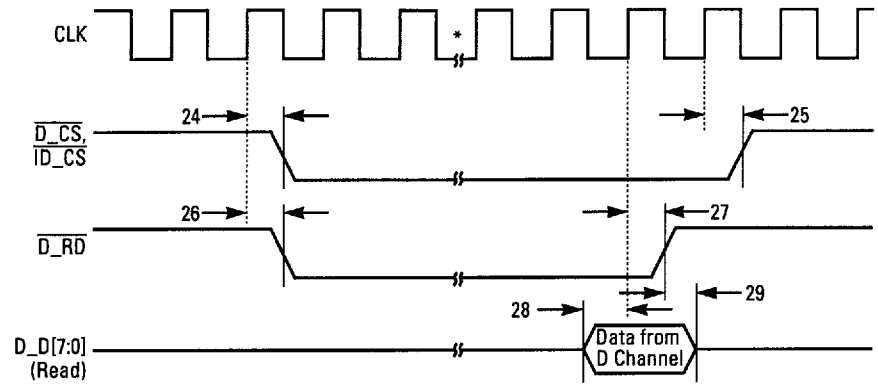
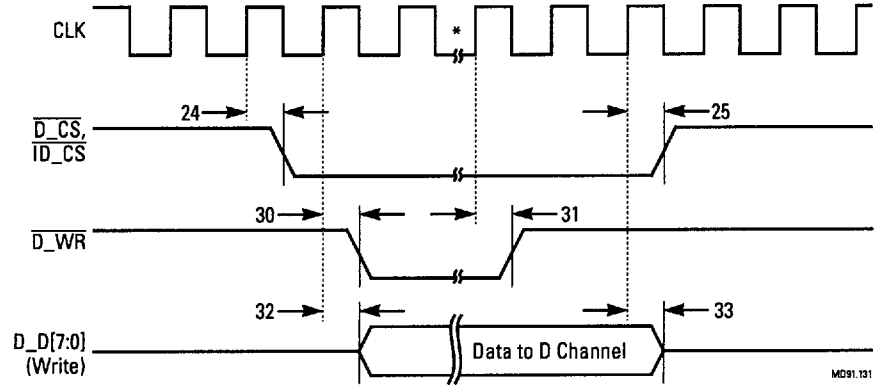


Figure 7.5
D-Channel Register Read Timing



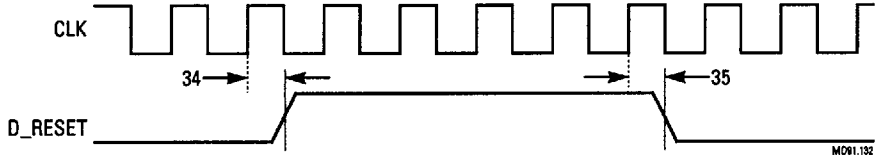
* These interruptions in the SBus CLK signal are inserted so that this figure applies to both fast and slow cycles. See Section 4.5, "Read/Write Transactions," for the actual number of clock cycles that $\overline{D\_RD}$ and $\overline{D\_CS}$ are asserted.

Figure 7.6
D-Channel Register Write Timing



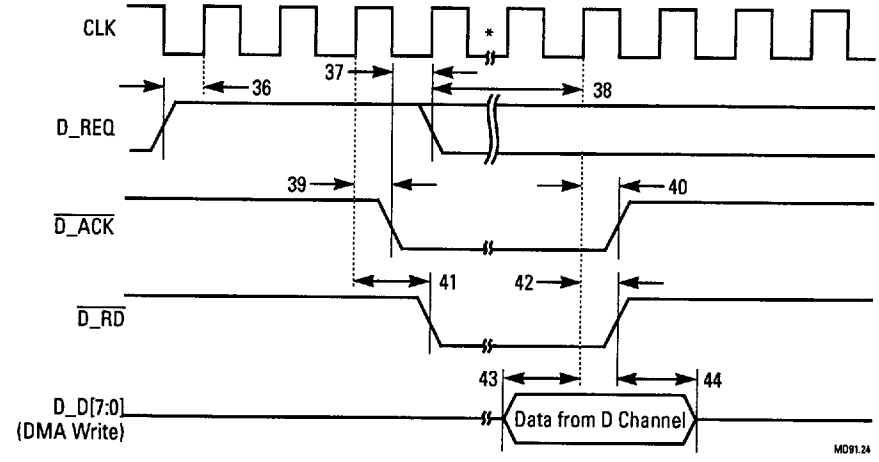
* These interruptions in the SBus CLK signal are inserted so that this figure applies to both fast and slow cycles. See Section 4.5, "Read/Write Transactions," for the actual number of clock cycles that $\overline{D\_WR}$ and $\overline{D\_CS}$ are asserted.

Figure 7.7
D-Channel Reset Timing



In Figure 7.8, a D-Channel DMA write indicates a data transfer from the D channel to memory.

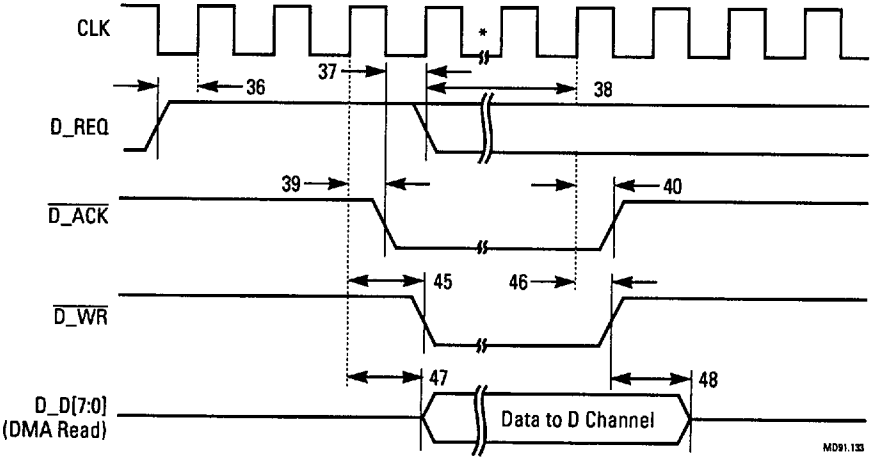
Figure 7.8 D-Channel DMA Write Timing



* These interruptions in the SBus CLK signal are inserted so that this figure applies to both fast and slow cycles. See Section 4.5, "Read/Write Transactions," for the actual number of clock cycles that $\overline{D\_RD}$ and $\overline{D\_ACK}$ are asserted.

In Figure 7.9, a D-Channel DMA read indicates a data transfer from memory to the D channel.

Figure 7.9 D-Channel DMA Read Timing



* These interruptions in the SBus CLK signal are inserted so that this figure applies to both fast and slow cycles. See Section 4.5, "Read/Write Transactions," for the actual number of clock cycles that $\overline{D\_WR}$ and $\overline{D\_ACK}$ are asserted.

Figure 7.10
E-Channel Register Access Timing

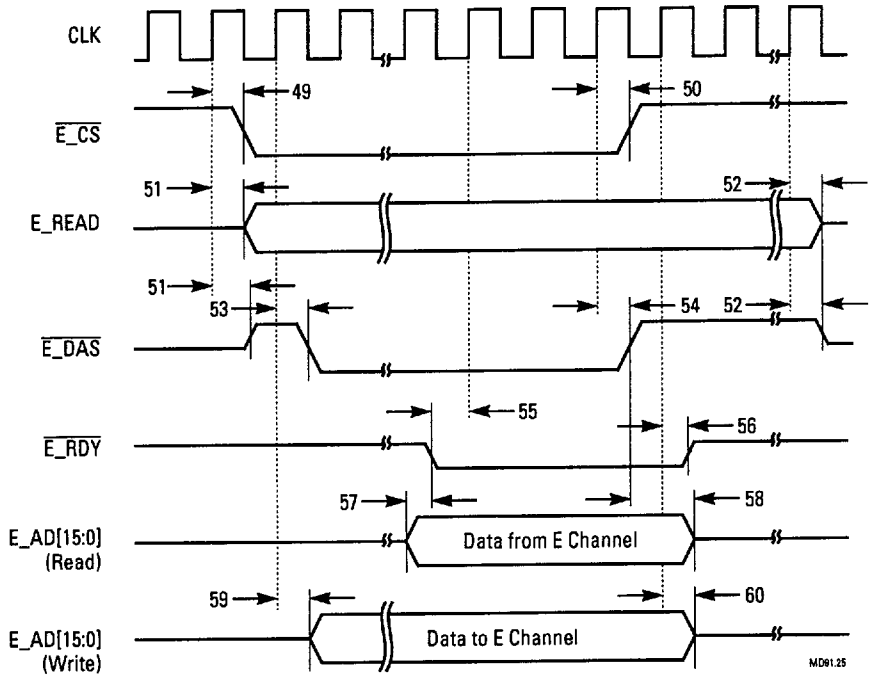
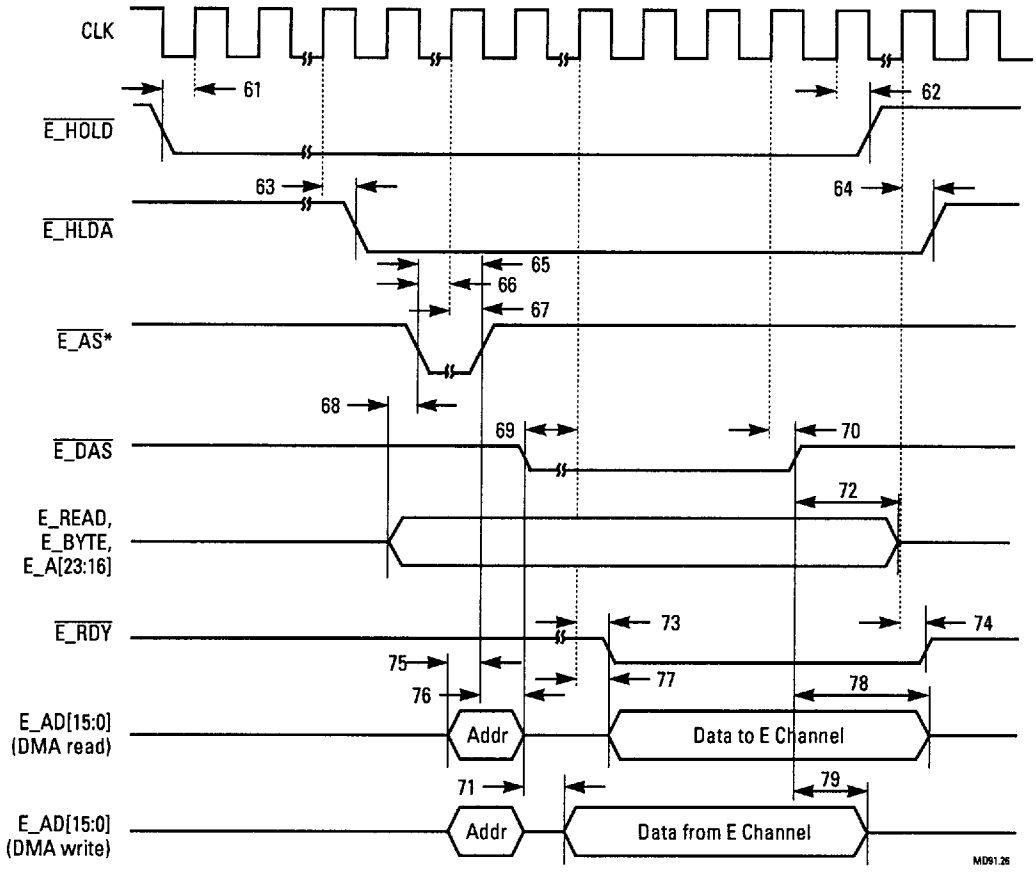
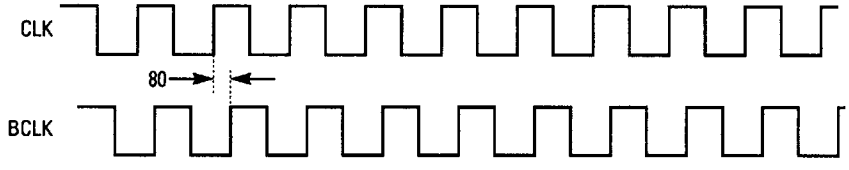


Figure 7.11
E-Channel DMA Cycle Timing



* This example uses E_AS as the output for the E_ALE/E_AS pin. Bit 20 of the Control/Status Register determines the polarity of the E_ALE/E_AS pin.

Figure 7.12
CLK to BCLK Skew



7.2 Electrical Specifications

This section specifies the electrical requirements for the L64853A DMA Controller. Five tables list electrical data in the following categories:

- Absolute Maximum Ratings (Table 7.2)
- Recommended Operating Conditions (Table 7.3)
- Capacitance (Table 7.4)
- DC Characteristics (Table 7.5)
- Alphabetical Pin Description Summary (Table 7.6)

Table 7.2 Absolute Maximum Ratings

Symbol	*Parameter*	*Limits*[1]	*Unit*
V_{DD}	DC Supply	0.3 to +7	V
V_{IN}	Input Voltage	-0.3 to V_{DD} + 0.3	V
I_{IN}	DC Input Current	±10	mA
T_{STG}	Storage Temperature Range	-40 to +125	°C

1. Referenced to V_{SS}.

Table 7.3 Recommended Operating Conditions

Symbol	*Parameter*	*Limits*	*Unit*
V_{DD}	DC Supply	+4.75 to +5.25	V
T_A	Ambient Temperature	0 to +70	°C

Table 7.4 Capacitance

Symbol	*Parameter*[1]	*Min*	*Typ*	*Max*	*Units*
C_{IN}	Input Capacitance	–	7.5	–	pF
C_{OUT}	Output Capacitance	–	8	–	pF
C_{IO}	I/O Bus Capacitance	–	8	–	pF

1. Measurement conditions are V_{IN} = 5.0 V, T_A = 25 °C, and clock frequency = 1 MHz.

Table 7.5
DC Characteristics

Symbol	*Parameter*	*Condition*[1]	*Min*	*Typ*	*Max*	*Units*
V_{IL}	Voltage Input Low		–	–	0.8	V
V_{IH}	Voltage Input High		2.0	–	–	V
V_{OH}	Voltage Output High	I_{OH} = -4.0 mA	2.4	4.5	–	V
V_{OL}	Voltage Output Low	I_{OL} = 4.0 mA	–	0.2	0.4	V
I_{IL}	Current Input Leakage	V_{DD} = Max, V_{IN} = V_{DD} or V_{SS}	-10	±1	10	µA
I_{OZ}	Current 3-State Output Leakage	V_{DD} = Max, V_{OUT} = V_{SS} or V_{DD}	-10	±1	10	µA
I_{IPU}	Current Input w/Pullup[2]	V_{IN} = V_{SS} or 3.5 V	-15	-45	-130	µA
I_{IPD}	Current Input w/Pulldown[2]	V_{IN} = V_{DD} or 0.8 V	10	35	110	µA
I_{OZ}	Current 3-State Output	V_{OH} = V_{SS} or V_{DD}	-10	±1	10	µA
I_{OSP4}	Current P-Channel Output Short Circuit (4 mA Output Buffers)[3]	V_{DD} = Max, V_{OUT} = V_{SS}	-35	-70	-90	mA
I_{OSN4}	Current N-Channel Output Short Circuit (4 mA Output Buffers)[3]	V_{DD} = Max, V_{OUT} = V_{DD}	35	62	83	mA
I_{DD}	Quiescent Supply Current	V_{IN} = V_{DD} or V_{SS}	–	–	2	mA
I_{CC}	Dynamic Supply Current	V_{DD} = Max, f = 25 MHz	–	–	95	mA

1. Specified at V_{DD} equals 5 V ± 5% at ambient temperature over the specified range.
2. To identify L64853A signals that have internal pullup or pulldown resistors, see the column "Type" in Table 7.6.
3. Not more than one output may be shorted at a time for a maximum duration of one second. See the column "Drive" in Table 7.6 to identify the drive levels of L64853A output or bidirectional signals.

Table 7.6 Alphabetical Pin Description Summary

Mnemonic	*Description*	*Type*[1]	*Drive (mA)*	*Active*
$\overline{\text{ACK}}$[2:0]	SBus Acknowledge	3-State Bidirectional, PU	4	Low
$\overline{\text{AS}}$	SBus Address Strobe	TTL Input, PU	–	Low
BCLK	Buffered SBus Clock	3-State Output	4	–
$\overline{\text{BG}}$	SBus Bus Grant	TTL Input, PU	–	Low
$\overline{\text{BR}}$	SBus Bus Request	3-State Output	4	Low
CLK	SBus Clock	Direct Input Clock Driver	–	–
D[31:0]	SBus Data Bus	3-State Bidirectional	4	–
$\overline{\text{D\_ACK}}$	DMA Acknowledge	3-State Output	4	Low
$\overline{\text{D\_CS}}$	DMA Chip Select	3-State Output	4	Low
D_D[7:0]	DMA Data Bus	3-State Bidirectional, PD	4	–
$\overline{\text{D\_IRQ}}$	DMA Interrupt Request	TTL Input, PU	–	Low
$\overline{\text{D\_RD}}$	DMA Read Strobe	3-State Output	4	Low
D_REQ	DMA Request	TTL Input, PD	–	High
D_RESET	D-Channel Reset	3-State Output	4	High
$\overline{\text{D\_WR}}$	DMA Write Strobe	3-State Output	4	Low
E_A[23:16]	E-Channel Address	TTL Input, PD	–	–
E_AD[15:0]	E-Channel Address/Data Bus	3-State Bidirectional, PU	4	–
E_ALE/$\overline{\text{E\_AS}}$	E-Channel Address Latch Enable/Address Strobe	TTL Input, PU	–	–
E_BYTE	E-Channel Byte Marker	TTL Input, PU	–	High
$\overline{\text{E\_CS}}$	E-Channel Chip Select	3-State Output	4	Low
$\overline{\text{E\_DAS}}$	E-Channel Data Strobe	3-State Bidirectional, PU	4	Low
$\overline{\text{E\_HLDA}}$	E-Channel Hold Acknowledge	3-State Output	4	Low
$\overline{\text{E\_HOLD}}$	E-Channel Hold	TTL Input, PU	–	Low
$\overline{\text{E\_RDY}}$	E-Channel Ready	3-State Bidirectional, PU	4	Low
E_READ	E-Channel Read	3-State Bidirectional, PU	4	High
$\overline{\text{ID\_CS}}$	Secondary Device Select	3-State Bidirectional, PU	4	Low
$\overline{\text{INTREQ}}$	SBus Interrupt Request	3-State Bidirectional, Open Drain	4	Low
$\overline{\text{LERR}}$	SBus Late Error	TTL Input, PU	4	Low
PA[3:2]	SBus Physical Address [3:2]	TTL Input, PU	–	–
PA[X:Y]	SBus Physical Address [X:Y]	TTL Input, PU	–	–
RD	SBus Read/Write	3-State Bidirectional, PU	4	High
$\overline{\text{RESET}}$	SBus Reset	TTL Input, PU	–	Low
$\overline{\text{SEL}}$	SBus Select	TTL Input, PU	–	Low
SIZ[2:0]	SBus Transfer Size	3-State Bidirectional, PU	4	–
$\overline{\text{SLOW}}$	Fast/Slow DMA Acknowledge Cycles	TTL Input, PU	–	Low
V_{DD}	Power	–	–	–
V_{SS}	Ground	–	–	–

1. PU = internal pullup, PD = internal pulldown.

7.3 Mechanical Specifications

The L64853A is available in a 120-pin Plastic Quad Flat Pack (PQFP) package. Table 7.7 provides ordering information.

Table 7.7 L64853A Ordering Information

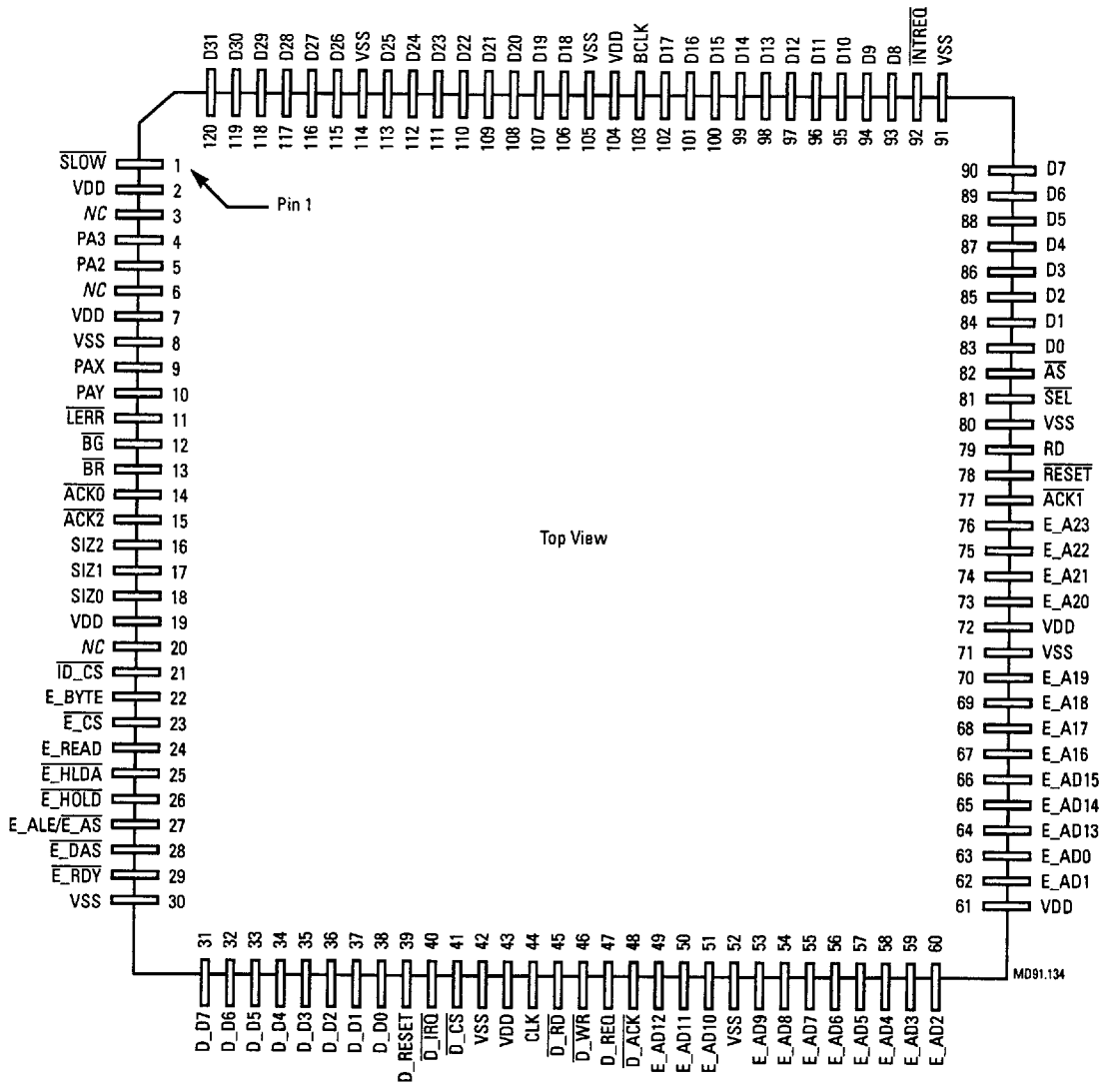
Order Number	*Clock Frequency (MHz)*	*Package Type*	*Operating Range*
L64853AQC-25	25	120-Pin PQFP	Commercial

Table 7.8 provides an alphabetical pin list for the L64853A. Figure 7.13 and Figure 7.14 provide the pinout diagram and the mechanical drawing, respectively.

Table 7.8
Alphabetical Pin List
for the 120-Pin PQFP

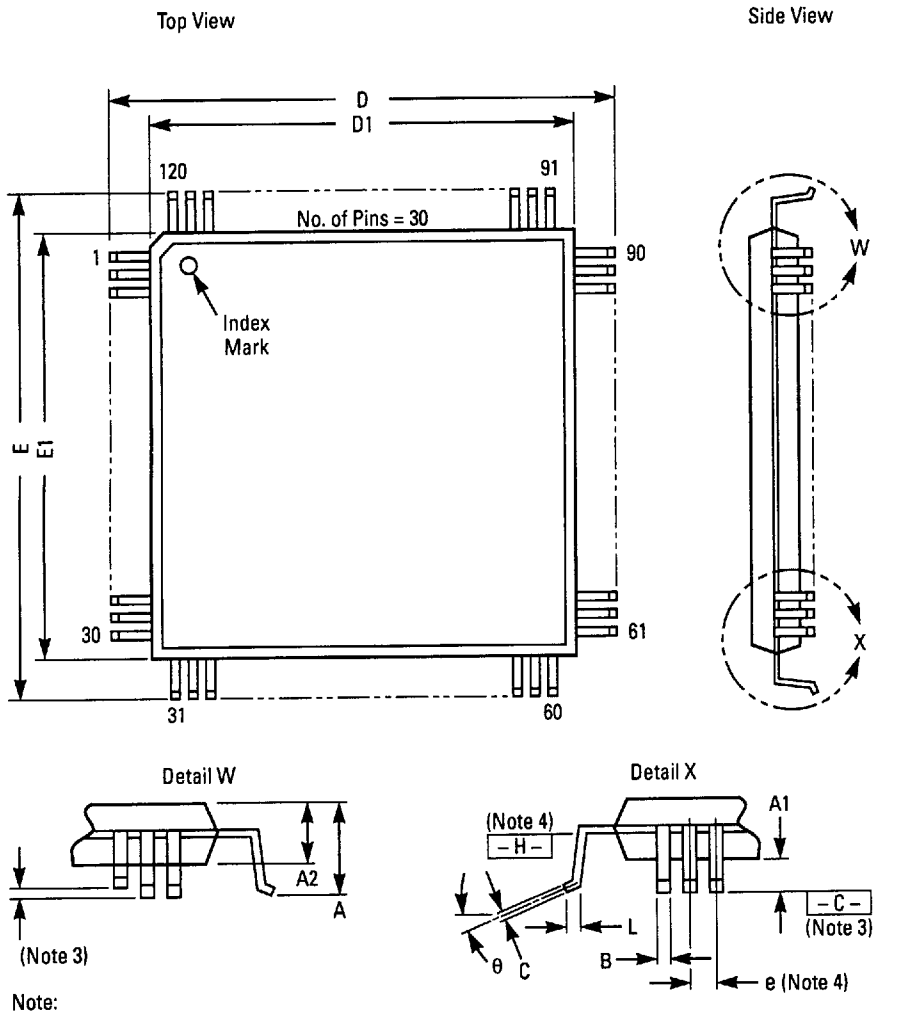
Signal	*Pin*	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*	*Pin*
ACK0	14	D23	111	E_A23	76	*NC*	20
ACK1	77	D24	112				
ACK2	15	D25	113	E_AD0	63	PA2	5
		D26	115	E_AD1	62	PA3	4
AS	82	D27	116	E_AD2	60	PAX	9
		D28	117	E_AD3	59	PAY	10
BCLK	103	D29	118	E_AD4	58		
BG	12	D30	119	E_AD5	57	RD	79
BR	13	D31	120	E_AD6	56	RESET	78
				E_AD7	55		
CLK	44	D_ACK	48	E_AD8	54	SEL	81
		D_CS	41	E_AD9	53	SIZ0	18
D0	83			E_AD10	51	SIZ1	17
D1	84	D_D0	38	E_AD11	50	SIZ2	16
D2	85	D_D1	37	E_AD12	49	SLOW	1
D3	86	D_D2	36	E_AD13	64		
D4	87	D_D3	35	E_AD14	65	VDD	2
D5	88	D_D4	34	E_AD15	66	VDD	7
D6	89	D_D5	33			VDD	19
D7	90	D_D6	32	E_ALE/E_AS	27	VDD	43
D8	93	D_D7	31	E_BYTE	22	VDD	61
D9	94			E_CS	23	VDD	72
D10	95	D_IRQ	40	E_DAS	28	VDD	104
D11	96	D_RD	45	E_HLDA	25		
D12	97	D_REQ	47	E_HOLD	26	VSS	8
D13	98	D_RESET	39	E_RDY	29	VSS	30
D14	99	D_WR	46	E_READ	24	VSS	42
D15	100					VSS	52
D16	101	E_A16	67	ID_CS	21	VSS	71
D17	102	E_A17	68	INTREQ	92	VSS	80
D18	106	E_A18	69			VSS	91
D19	107	E_A19	70	LERR	11	VSS	105
D20	108	E_A20	73			VSS	114
D21	109	E_A21	74	*NC*	3		
D22	110	E_A22	75	*NC*	6		

Figure 7.13
120-Pin PQFP
Pinout Diagram



Note:
1. *NC* (not connected) pins must *not* be connected to anything.

Figure 7.14
120-Pin PQFP
Mechanical Drawing



Dimension		mm
A	Max	4.01
A1	Min	0.25
A2	Min	3.20
	Nom	3.40
	Max	3.60
B	Min	0.30
	Max	0.45
C	Min	0.13
	Max	0.23
D	Min	31.60
	Nom	32.00
	Max	32.40
D1	Min	27.90
	Max	28.10
e	BSC	0.80
E	Min	31.60
	Nom	32.00
	Max	32.40
E1	Min	27.90
	Max	28.10
L	Min	0.73
	Nom	0.88
	Max	1.03
θ	Min	0°
	Max	7°

Note:
1. Total number of pins is 120.
2. Drawing is not to scale.
3. Coplanarity of all leads shall be within 0.10 mm (difference between the highest and lowest lead with seating plane – C – as reference).
4. Datum plane – H – is located at mold parting line and is coincident with the bottom of the lead, where the lead exits the plastic body. Lead pitch determined at – H –.
5. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. These dimensions to be determined at – H –.
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 PE.

MD92.PEc

Chapter 8
Applications

This chapter provides information on using the L64853A DMA Controller in two types of SPARC Workstation environments and also describes how the controller can be configured with two peripheral chips: the AMD Am7990 Local Area Network Controller for Ethernet (LANCE) and the Emulex SCSI Processor ESP-100 (ESP). The L64853A DMA Controller interfaces directly with both peripheral chips without additional circuitry to offer a low chip-count solution for networking and hard disk requirements.

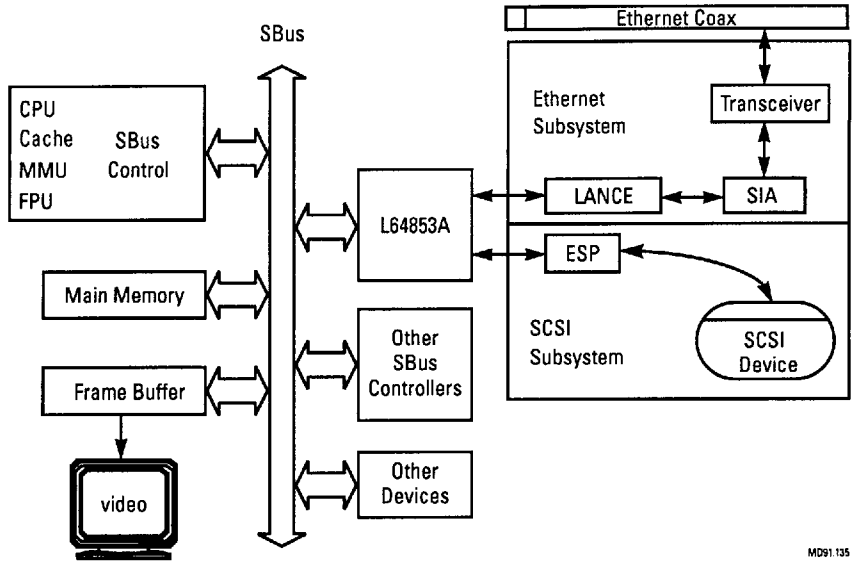
For more information on these peripheral chips, refer to the appropriate documentation from Advanced Micro Devices, Inc. and Emulex Corporation.

8.1 The SBus and the SPARC Workstation

LSI Logic developed the L64853A for use with its SPARC family of products. As such, it fits into one of two distinct kinds of SPARC workstation environments: a host-based system and a symmetric system.

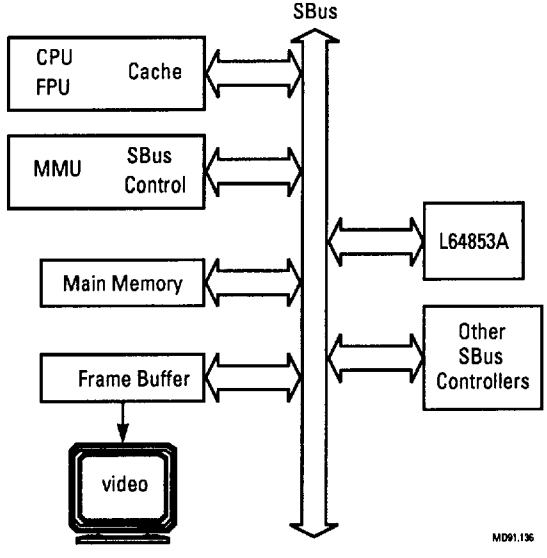
In a host-based system, the CPU has a private address translation facility; in SBus terminology, this kind of system has a *CPU Master.* Host-based architectures are applicable in very high-performance systems, such as the Sun SPARCstation 1. Figure 8.1 shows a typical host-based configuration.

Figure 8.1 Host-Based System: SPARCstation 1



In a symmetric system, the CPU uses the same MMU as all the other SBus Masters; in SBus terminology, all SBus Masters in this kind of system are *Direct Virtual Memory Address (DVMA) Masters*. This type of architecture is useful in low-cost systems that do not need the extra circuitry that supplies the CPU with special address translation facilities. Figure 8.2 shows a symmetric system.

Figure 8.2 Symmetric Configuration



8.2 Using the EMULEX SCSI Processor

This section illustrates how to interconnect the L64853A DMA Controller with the Emulex ESP-100 SCSI Processor. This section also shows how to access the ESP's internal registers.

L64853A to ESP-100 Interconnection

The interface between the L64853A and the Emulex ESP-100 is straightforward and requires no additional logic. Figure 8.3 illustrates the interface.

Figure 8.3 L64853A to ESP-100 Interface

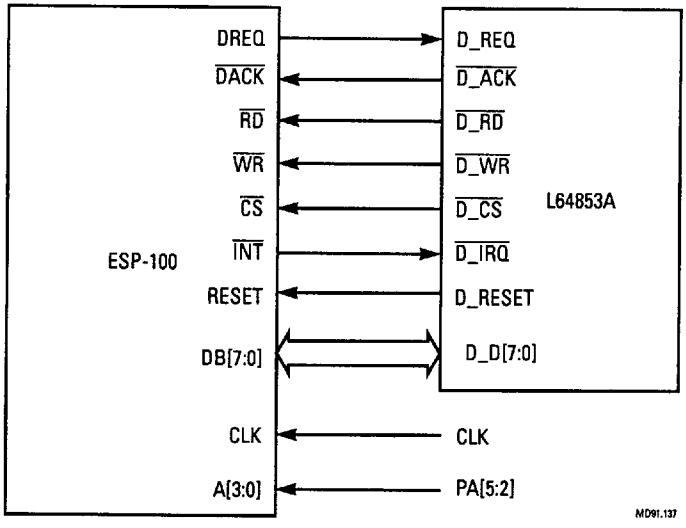
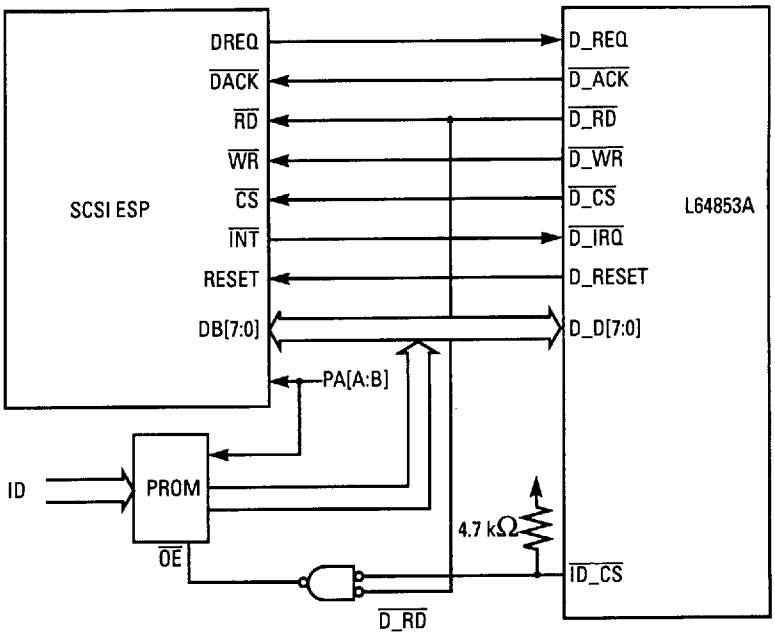


Figure 8.4 shows an ESP interface example using an external boot PROM.

Figure 8.4 ESP Interface Example Using External Boot PROM



Accessing ESP Internal Registers

Table 8.1 shows how the physical address lines of the SBus on the SPARCstation 1 select the external registers of the ESP SCSI chip.

Table 8.1 Selecting External Registers on ESP SCSI Chip

PA[X:Y] = PA[23:22]	*PA[5:0]*	*Register Name*
10	0x0	Transfer Count Low (Read/Write)
10	0x4	Transfer Count High (Read/Write)
10	0x8	FIFO Data (Read/Write)
10	0xC	Command (Read/Write)
10	0x10	Status (Read)/Bus ID (Write)
10	0x14	Interrupt Status (Read)/Time-out (Write)
10	0x18	Sequence Step (Read)/Sync Period (Write)
10	0x1C	FIFO Flags (Read)/Sync Offset (Write)
10	0x20	Configuration (Read/Write)
10	0x24	Clock Conversion Factor (Write)
10	0x28	ESP Test (Write)
10	0x2C	Configuration (Read/Write)
10	0x30 – 0x3C	Reserved

Because the ESP SCSI Controller has its own Byte Counter register, do not use the DMA Byte Counter Register (set the EN_CNT bit of the DMA Control/Status Register to 0).

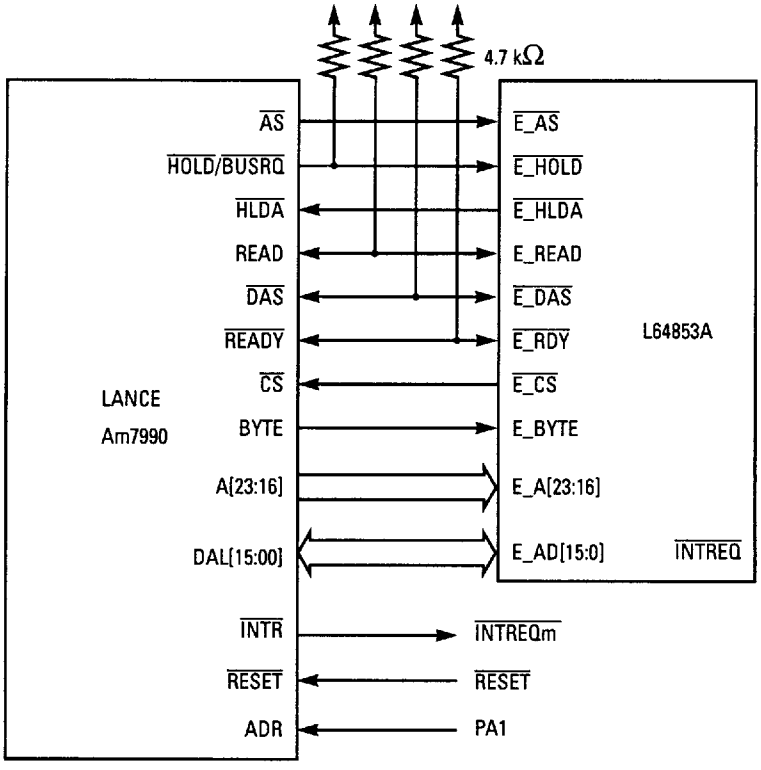
8.3 Using the AMD LANCE Ethernet Controller

This section shows the interconnection between the L64853A DMA Controller and the AMD Am7990 Local Area Network Controller for Ethernet (LANCE). This section also indicates how to access the LANCE's internal registers.

L64853A to LANCE Interconnection

The interface between the L64853A and the LANCE is straightforward and requires no additional logic. Figure 8.5 illustrates the interface.

Figure 8.5 L64853A to Am7990 LANCE Interface



Accessing LANCE Internal Registers

The LANCE contains four Control and Status Registers. Two bus addressable ports, an address port (RAP) and a data port (RDP), provide access to these registers. Table 8.2 shows how physical address lines of the SBus on the SPARCstation 1 access the two address ports.

Table 8.2 Addressing Registers on LANCE

SEL	*$\overline{AS}$*	*PA[X:Y] = PA[23:22]*	*PA1*	*Port Accessed*
0	0	11	0	Register Data Port (RDP)
0	0	11	1	Register Address Port (RAP)

Appendix A
L64853 and L64853A Differences

The L64853A Enhanced SBus DMA Controller is both pin- and software-compatible with the L64853 SBus DMA Controller. This appendix focuses on the differences between the two devices.

A.1 Line Buffers

Each peripheral channel of the L64853 has one four-byte line buffer, also known as the Pack/Unpack Register. Each peripheral channel of the L64853A contains two 16-byte line buffers, also known as Cache. Each line buffer in the L64853A allows for a 16-byte burst operation. Two line buffers per channel allow for simultaneous transfers over the SBus and the peripheral interface—one line drains while the other line fills.

The L64853A can operate with existing L64853 software drivers. With the existing drivers, the L64853A performs burst operations, but does not perform data block chaining operations.

A.2 Pin Changes

Three pins are different on the L64853A. These differences are listed in Table A.1.

Table A.1 Pin Differences

Pin Number	*L64853*	*L64853A*
6	PA1	No Connection
27	$\overline{\text{E\_AS}}$	E_ALE/$\overline{\text{E\_AS}}$
103	No Connection	Buffered SBus Clock (BCLK)

A.3 Internal Register Changes

The L64853A contains two new registers: the Next Address Register and the Next Byte Counter Register. These registers are used for D-channel data block chaining. In addition to the new registers, the L64853A also includes several changes to the DMA Control/Status Register (CSR). Table A.2 lists the CSR differences between the L64853 and the L64853A.

Table A.2
CSR Differences

CSR Bits	L64853	L64853A
31:28	DEV_ID = 1000_2	DEV_ID = 1001_2
27	Reserved	NA_LOADED
26	Reserved	A_LOADED
25	Reserved	DMA_ON
24	Reserved	EN_NEXT
23	Reserved	TCI_DIS
22	Reserved	FASTER
21	Reserved	LANCE_ERR
20	Reserved	ALE/$\overline{AS}$
15	ILACC	Reserved
12:10	REQ_PEND, BYTE_ADDR	Unused
6	DRAIN	SLAVE_ERR
3:2	PACK_CNT	DRAINING

The L64853 and L64853A handle the end of a D-channel DMA write differently. The PACK_CNT field in the L64853 contains the number of bytes in the line buffer. For systems using the L64853, at the end of a block transfer, the device driver must read the PACK_CNT field to check if any remaining bytes are in the line buffer. If the line buffer is not empty, the device driver writes a "1" to the DRAIN field. When the driver sets DRAIN, the L64853 then drains the data from the line buffer to memory.

To clear PACK_CNT, set FLUSH or DRAIN. FLUSH merely clears PACK_CNT, whereas DRAIN writes the contents of the line buffer to memory and then clears PACK_CNT.

The L64853A automatically drains dirty data when one of the draining conditions is met, so the PACK_CNT and DRAIN functions are not needed. The DRAINING field is provided to indicate when the L64853A is draining dirty data. The L64853A automatically clears the DRAINING field when draining is complete. Refer to Section 4.3, "D-Channel Cache Operation," and Section 5.2, "E-Channel Cache Operation," for more information on draining.

The L64853A SLAVE_ERR field replaces the L64853 DRAIN field. The L64853A sets SLAVE_ERR when the L64853A accesses a device with a certain size encoding, using SIZ[2:0], and that device acknowledges with a size different than requested using $\overline{ACK}$[2:0].

The L64853 sets the REQ_PEND bit when transfers are taking place to and from the D-channel. The device driver polls this bit until REQ_PEND is cleared, and then takes any necessary *clean-up action* by setting the FLUSH or DRAIN bits. The L64853A does not use this field; because the L64853A automatically performs clean-up action, this field is removed.

BYTE_ADDR on the L64853 contains the lowest two address bits of the next byte to be accessed by the D-channel controller. The L64853A does not use this field, which reads as zero.

Appendix B
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.