VIC64 Design Notes

Introduction

Cypress Semiconductor has supplemented its Bus Interface product family with the VIC64, a member of the industry-standard VIC family of VMEbus interface products. The VIC64 implements 64-bit wide block transfers, in addition to 32- and 16-bit block transfers and 32-, 16-, and 8-bit single-cycle transfers, all using the same backplane hardware. VIC64 is software and hardware compatible with the VIC068A VMEbus Interface Circuit.

This document provides the designer with the information needed to evaluate and develop VIC64-based boards. You should have already read the User's Guide for the VIC068A. This document provides information on the enhancements found within the VIC64. Another recent addition to Cypress's Bus Interface Products portfolio, the CY7C964 Bus Interface Logic Chip, is described later in this book.

Like the Cypress VIC068A, the VIC64 contains all the circuitry necessary to manage VMEbus transfers, either as a slave or a master. It can also be programmed as the VMEbus system controller. The VIC64 contains circuitry intended to minimize the problems associated with the development of a VMEbus interface such as an interrupt controller, a DMA controller, a DRAM refresh controller, and many other features normally found on VMEbus boards. The VIC64 is a logical extension to the capabilities of the VIC068A, the industry-standard VMEbus Interface Controller chip, and it is fully compatible with the VIC068A.

The primary benefit of using the VIC64 is that you can perform 64-bit VMEbus transfers. The VIC64 also contains some enhancements to the VIC068A, including some performance improvements and additional features.

A board that has been designed to use the VIC068A is not likely to implement D64 VMEbus transfers, but there are several reasons why a user may choose to replace the VIC068A with a VIC64. For example, to take advantage of the enhancements of the VIC64, or to evaluate the device and use an existing board to speed the evaluation.

Compatibility

All pin assignments and register assignments are the same as those of the VIC068A, therefore the VIC64 will work flawlessly when used to replace a VIC068A. In fact, several of the VIC068's functions have been enhanced in the VIC64, allowing VIC068A applications to run faster in some cases. Naturally, some attention must be paid to the additional controls for the improved functionality to ensure that the original hardware and software supports the improvements. Therefore, several bits have been added to the VIC068A registers to control the enhancements. They map into the unused bits within the VIC068A register space; assuming that the VIC068A developer has not inadvertently set the bits, VIC068A code will run on the VIC64 without modification.

To add 64-bit functionality to the VIC family and still retain plug compatibility, the SCON* pin has been modified in the VIC64. Whereas previously it performed only the input function of selecting VIC068A to be the system controller, in VIC64 the pin is sensed and latched during reset to determine whether the system controller function is enabled. After reset the pin becomes an output to control external circuitry during 64-bit transfers. The new name for the pin is SCON*/D64. If you simply replace the VIC068A with the VIC64, the VIC64 will function in an identical manner to the VIC068A, whether it is the system controller or not. It is recommended that the VIC64 SCON*/D64 pin be connected via a resistive pull-down/up of greater than 4.7kΩ to enable/disable the system controller function.

64-Bit Operation

VMEbus Specification

The primary reason for the development of the VIC64 was to provide users with the capability to perform 64-bit-wide data transfers in a manner consistent with the goals of the proposed 64-bit VMEbus specification, more commonly known as the VME64 specification. Although at the date of this publication the specification was not finalized, the protocol for 64-bit MBLT (Multiplexed Block Transfer) is well known and stable. The VIC64 implements this protocol.

Address Modifier Codes

VIC64 responds as a slave to the address modifier (AM) codes associated with MBLT transfers as follows:

3C, 38, 0C, 08; Performs D64 operation as implied by the actual AM code, and the contents of the Slave Configuration Registers $C3 and $CB, and Block Transfer Definition Register $AB.

00 - 07; No response: these codes are associated with 64 bit address operations, and the VIC64 does not support 64-bit address operations.

As a master, VIC64 will use the MBLT protocol to transfer data if the appropriate conditions occur:

AM codes 3C, 38, 0C, 08 are selected, and the appropriate bits of the configuration registers are set (see later for exact details).

In summary, VIC64 performs A32/D64 operations in addition to the D8/16/32 single cycles and A16/24/32..D16/32 block transfers performed by VIC068A.

Boundary Crossing

There are several implications of the 64-bit VMEbus protocol and the requirement for compatibility that you should consider. The VIC64, being a 144-pin device, can connect to only the lower byte of the VMEbus address. For block transfers other than D64 transfers, the VMEbus specification requires that the VMEbus address be rebroadcast at 256-byte boundary crossings; this quantity maps neatly into the byte of address that the VIC64 can monitor. MBLT transfers, however, are required to rebroadcast the address only at 2-Kbyte boundaries. VIC64 has no means of determining how the starting address of a master block transfer relates to the 2K boundary (it has access to only the lower 8 address bits), and therefore VIC64 rebroadcasts the address at every 256-byte boundary. This is still compatible with the specification, but has a small impact on the sustained transfer rate. If you wish to take advantage of the increased performance of 2-Kbyte boundaries, then VIC64 can be programmed to rebroadcast the address every 2048 bytes, and the starting address must then be aligned on a 2-Kbyte boundary.

External Circuit Complexity

The VIC64 is a flexible building block that can be used in many different configurations. The VMEbus specification is written to allow many levels of circuit complexity to conform to the specification. Such circuitry may include slave address decode circuitry, local DMA transfer, slave read modify cycles, and more. The VIC64 and the VIC068A provide dual-path operation, a mechanism whereby the local bus master can perform single-cycle VMEbus operations during the time that the VMEbus is between block transfer bursts (interleave period). They also provide a mechanism allowing master write-posting, and slave read modify cycles to occur concurrently without harming the posted data. All this circuitry must be duplicated externally for the higher-order data bytes if you want these features.

You may choose to implement only those features that your system requires, thereby simplifying the necessary external circuitry. Alternatively, the user may decide to use the companion device, the CY7C964, and gain access to all the features using only three small devices. The CY7C964 is described in the second half of this publication.

VIC068A User's Guide: Additional Information

The following sections are related to the *VIC068A User's Guide*. The chapter numbers are those sections of the *VIC068A User's Guide* that require clarification or modification for the VIC64. All other information in the *VIC068A User's Guide* is applicable to the VIC64.

VIC64 Signal Description (Chapter 2)

All pins are identical to those of the VIC068A with the following exception:

SCON*/D64

Input: Yes

Output: Yes

Drive: 16mA

This is the dual-function signal by which the VIC64 determines whether system controller functions are required, and by which the VIC64 controls external logic during 64-bit VMEbus transfers. During the time that IRESET* is asserted, this pin is the SCON* input whose state is latched internally when IRESET* goes inactive. A Low state causes VIC64 to become the VMEbus system controller. During the time that IRESET* is inactive, this pin becomes the D64 output whose state is normally Low, becoming High only during the Data Phase of D64 transactions.

System Controller Operations (Chapter 4)

The VIC64 functions identically to VIC068A as a system controller, except that the SCON* pin of the VIC068A has been renamed to be SCON*/D64 on the VIC64. During the period that IRESET* is asserted (Low), VIC64 assumes that the pin is an input whose state is latched on the rising edge of IRESET*. The latched state is then used to determine whether the VIC64 is the system controller: if the state is Low, then the VIC64 is the system controller.

SCON*/D64 becomes an output after the rising edge of IRESET*; the state of the output is used to enable 64-bit data transfers (Chapter 10, Block Transfer Functions contains information on this operation).

VMEbus Master Operations (Chapter 5)

There is no provision for single-cycle D64 transfers in the VMEbus specification. The VIC64 does not perform single-cycle 64-bit transfers.

The VIC64 uses the same pins and register bits as the VIC068A to configure and select 64-bit block transfers. The release modes are identical, and the address broadcast phase is identical to the VIC068A, except that the AM code reflects the D64 transaction (see Table 1).

Table 1. Master Transfer AM Code Control Map for D64 Operations

VIC64 Master Access Inputs				VIC64 AM Code Output	
ASIZ1/0	**Address Size**	**Blk**	**FC2**	**Operation Type**	**AM[5:0]**
01	A32 addressing	Yes	0X	User block	$08
			1X	Supervisory block	$0C
11	A24 addressing	Yes	0X	User block	$38
			1X	Supervisory block	$3C

As the VIC64 has an identical local bus interface to that of the VIC068A, some mention must be made of the protocol used to transfer the 64-bit VMEbus data to the 32-bit local bus. First, it should be noted that data byte D0 is transferred on VMEbus address [A31:A24], and byte D7 is transferred on VMEbus data [D7:D0]. Two local transactions are required for each VMEbus transaction. For maximization of performance a pipelined architecture is used. The VIC64 provides the appropriate timing for latch controls to implement the pipe externally for those bytes that the VIC64 itself does not connect to.

D64 Master Write Cycles

In the case of master write cycles, the first local cycle fetches the first [D0−D3] longword and the VIC64 places it into a two-stage pipe: the next local cycle fetches the next longword and presents it to the VMEbus data bus, while the piped data is presented to the VMEbus address bus. Then the next local cycle can commence without waiting for the completion of the VMEbus cycle, as the first stage of the pipe is now free. See the timing diagrams for full details of this operation.

D64 Master Read Cycles

In the case of Master Read Cycles, 64 bits of VMEbus data are latched under the control of VIC64. The [D4−D7] long-word is placed into a two-stage pipe, while the [D0−D3] long-word is presented to the local bus. After the local bus write cycle, the piped data is then presented to the local bus, and the next VMEbus cycle can commence as the first stage of the pipe is now free. See the timing diagrams for full details of this operation.

VMEbus Slave Operations (Chapter 6)

Upon detecting SLSEL0* or SLSEL1* asserted, the VIC64 behaves in an identical manner to the VIC068A except that if the AM code for the slave transaction is $08, $0C, $38, or $3C, the VIC64 configures itself for a D64 slave block transfer (see Table 2).

Table 2. Slave Transfer AM Code Control Map for D64 Operations

VIC64 AM Code Inputs		VIC64 Slave Access Outputs		
Operation Type	**AM[5:0]**	**Address Size**	**Block Transfer**	**FC2/1**
User block	$08	A32 addressing	Yes	00
Supervisory block	$0C			
User block	$38	A24 addressing	Yes	00
Supervisory block	$3C			

D64 Slave Read Cycles

As in the case of master write cycles, the first local cycle fetches the first [D0–D3] longword and the VIC64 places it into a two-stage pipe. The next local cycle fetches the next longword and presents it to the VMEbus data bus, while the piped data is presented to the VMEbus address bus. Then the next local cycle can commence without waiting for the completion of the VMEbus cycle, as the first stage of the pipe is now free.

D64 Slave Write Cycles

As in the case of Master Read Cycles, 64 bits of VMEbus data are latched under the control of VIC64. The [D4–D7] long-word is placed into a two-stage pipe, while the [D0–D3] longword is presented to the local bus. After the local bus write cycle, the piped data is then presented to the local bus, and the next VMEbus cycle can commence because the first stage of the pipe is now free.

Interrupts (Chapter 9)

The VIC64 can be programmed to perform either D8, D16, or D32 interrupt acknowledge cycles. The method by which this is performed is simply to drive the values on SIZ1/0, in a similar fashion to a master read or write operation. The SIZ1/0 lines are sensed by the VIC64 following the assertion of FCIACK* by the local processor. Note that no provision is made for non-aligned status/ID vector: The VIC64 enables the appropriate local bus drivers for either 8, 16, or 32 bit Status/ID.

Table 3. VIC64 Interrupt Acknowledge Cycle Selection

SIZ1/0	VMEbus Data Width
00	32
01	8
10	16
11	32

VIC64 Block Transfer Functions (Chapter 10)

As the VIC64 is a superset of the VIC068A, all the VIC068A block transfer functionality is reproduced in the VIC64. The additional features provided by the VIC64 are D64 transfers and performance enhancements.

D64 Transfers, VMEbus Boundary Crossing

The VME64 specification allows D64 block transfers to exceed the 256-byte boundary-crossing limitation that the original VMEbus specification contains. The new specification allows for 2-Kbyte boundaries. As the VIC64 can only discern 8 bits of address, it has no means of determining which 256-byte boundary is the 2048-byte boundary, and therefore the VIC64 rebroadcasts the address every 256 bytes unless BTDR[7] is set: this bit causes the address to be rebroadcast on 2-Kbyte boundaries, but the VIC64 then assumes that the transfer starts on the 2-Kbyte boundary.

Miscellaneous Features (Chapter 12)

Selection of System Controller Functionality

The VIC068A is configured to be system controller by strapping SCON* Low. In VIC64, the SCON*/D64 pin performs this function: the state of the pin is latched during any of the possible Reset operations, and this state determines whether VIC64 is system controller. Following the Reset operation, the SCON*/D64 pin becomes an output whose state controls the external circuitry (such as the CY7C964) used during the data phase of D64 transfers. The detailed timing of this operation depends upon internal states such as DRAM refresh timing, in addition to the external stimuli such as SYSRESET*, IRESET*, and IPL0*. Use of an external pull-up/pull-down resistor to determine the state of the SCON* pin during the Reset operation is all that is required to ensure correct operation.

Enhanced Turbo Mode

In addition to the use of ICR[1] another performance enhancement is possible in the VIC64. Setting BTDR[5] reduces the DSACK*-to-DTACK* time defined in the slave select control registers by 0.5 clock period for both master and slave block transfers. The reduced times are 0, 1.5, 2, 2.5,...,8.5 clock periods. See the AC Timing Parameters section for details on which times are affected by this bit.

Register Map and Descriptions (Chapter 13)

There are some differences between the VIC068A and the VIC64 register assignments and contents.

Interprocessor Communications Register 5

Name: ICR5

Address: $77

Description: This register provides the VIC64 revision number.

Block Transfer Definition Register

Name: BTDR

Address: $AB

Description: Configures master block transfers for boundary crossing, dual-path and user defined address modifiers. There are four additional bits defined for VIC64:

Bit 4 (0/0/0)	Enables D64 Master Operations when BTCR[6] is set
Bit 5 (0/0/0)	Enables Accelerated Block Transfer Operations as discussed above.
Bit 6 (0/0/0)	Enables D64 Slave Operations
Bit 7 (0/0/0)	Enables 2-k-byte boundary crossing for D64 Master Operations. If this bit is set, VIC64 assumes that the transfer is aligned to a 2-k-byte boundary.

Release Control Register

Name:	RCR
Address:	$D3
Description:	This register configures the VMEbus release mode, and the burst length for block transfers with local DMA.
Bits 5-0 (0/0/0)	For MBLT operations (D64 transfers), the burst length is 4 times the actual field contents. A value of 0 is interpreted to mean 4 x 64.

For non-MBLT operations, the burst length is simply the field contents. A value of 0 in this field is interpreted to mean 64.

Block Transfer Length Register 2

Name:	BTLR2
Address:	$E7
Description:	This register provides the most significant byte of the 24-bit value used to determine the byte count for block transfers with local DMA.
Bits 7-0 (0/0/0)	Bits 23:16 of the block transfer length.

AC Performance Specifications (Chapter 15)

AC Timings for D64 Operations (Commercial)[1]

Operation		Min.	Max.
Master D64 Block Transfer with Local DMA (Initiation Cycle)[2]			
A1	MWB*[0] & PAS*[0] & DS*(0) to BRi*[L]	T+7	2T+32
A2	MWB*[0] & PAS*[0] & DS*(0) to LADO[H]	T+9	2T+31
A3	MWB*[0] & PAS*[0] & DS*(0) to BLT*[L]	T+9	2T+26
A4	MWB*[0] & PAS*[0] & DS*(0) to DSACK1*[L]	T+11	2T+46
A5	MWB*[0] & PAS*[0] & DS*(0) to DSACK0*[L]	T+11	2T+46
Master D64 Block Transfer Address Broadcast Cycle[2]			
B1	DTACK*[0] to LBR*[L]	24	65
B2	DTACK*[0] to DSi*[H]	8	24
B3	DTACK*[0] to SCON*/D64[H]	16	59
Master D64 Block Transfer with Local DMA (Write)			
First Longword Fetch			
C1	DSACKi*[0] and DS*[L] to DS*[H][2, 3, 4]	MBAT0+8	MBAT0+T+36
C2	DSACKi*[0] and DS*[L] to LEDO[H] [2, 3, 4]	MBAT0+7	MBAT0+T+33
C3	DSACKi*[0] and DS*[L] to LA(7:0)[2, 3, 4]	MBAT0+.5T+9	MBAT0+2T+30
C4	DS*(H) to DS*[L][2, 3, 4, 5]	T+8	3T+31
Second Longword Fetch			
C5	DSACKi*[0] and DS*[L] to DS*[H] [2, 4]	MBAT1+14	MBAT1+T+46
C6	DSACKi*[0] and DS*[L] to DENO*[L] [4]	MBAT1+11	MBAT1+T+37
C7	DSACKi*[0] and DS*[L] to LA(7:0) [2, 4]	MBAT1+.5T+9	MBAT1+2T+31
C8	DSACKi*[0] and DS*[L] to DSi*[L] [6]	MBAT1+3T+12	MBAT1+4T+32
C9	DSACKi*[0] and DS*[L] to LEDO[L] [4]	MBAT1+16	MBAT1+T+56
C10	DS*(H) to DS*[L] [2, 4, 5]	T+8	3T+31
C11	DTACK*[0] to DSi*[H]	7	22
C12	DTACK*[0] to DENO*[H] [2]	8	24
Master Block Transfer with Local DMA (Read)			
First Longword Write			
D1	LBG*[0] to LWDENIN*[L] [2]	2T+11	3T+41
D2	DTACK*[0] to LEDI[H] [7]	2T+6	3T+23
D3	DTACK*[0] to DSi*[H] [2, 7]	2T+9	3T+28
D4	DTACK*[0] to DS*[L] [7]	2T+13	3T+36
D5	DSACKi*[0] and DS*[L] to DS*[H] [2, 3, 4]	MBAT0+8	MBAT0+T+37

AC Timings for D64 Operations (Commercial)

	Operation	Min.	Max.
D6	DSACKi*[0] and DS*[L] to LEDI[L] [3, 4]	MBAT0+13	MBAT0+T+52
D7	DSACKi*[0] and DS*[L] to UWDENIN*[L] [3, 4]	MBAT0+8	MBAT0+T+35
D8	DSACKi*[0] and DS*[L] to LA(7:0) [2, 3, 4]	MBAT0+.5T+9	MBAT0+2T+29
D9	DSACKi*[0] and DS*[L] to DSi*[L] [3, 4]	MBAT0+22	MBAT0+T+56
D10	DS*[H] to DS*[L] [2, 4, 5]	T+8	3T+29
Second Longword Write			
D12	DSACKi*[0] and DS*[L] to DS*[H] [2, 4]	MBAT1+8	MBAT1+T+36
D13	DSACKi*[0] and DS*[L] to UWDENIN*[H] [2, 4]	MBAT1+16	MBAT1+T+56
D14	DSACKi*[0] and DS*[L] to LA(7:0) [2, 4]	MBAT1+.5T+9	MBAT1+2T+29
D15	DSACKi*[0] and DS*[L] to LD(7:0) [2, 4]	MBAT1+.5T+12	MBAT1+2T+39
Master D64 Block Transfer with Local DMA (Boundary Crossing) [2]			
E1	DS*[0] to BLT*[H]	3	28
E2	DS*[1] to BLT*[L]	3	19
E3	DSi*[0] to LADO first transition	3	19
E4	DSi*[1] to LADO second transition	3	19
Slave D64 Block Transfer Address Broadcast Cycle [2]			
F1	DSi*[1] to LBR*[L]	11	36
F2	DSi*[0] to DTACK*[L]	2T+9	3T+28
F3	DSi*[1] to DTACK*[H]	9	28
F4	DSi*[1] to SCON*/D64[H]	10	33
F5	DSi*[0] and AS*[0] and SLSELi[0] to LADI[H]	1.5T+5	2T+25
Slave D64 Block Transfer (Write)			
First Longword Cycle			
G1	DSi*[0] to DS*[L] [2, 4]	3T+11	4T+38
G2	DSi*[0] to DTACK*[L] [7]	2T+9	3T+23
G3	DSi*[0] to LEDI[H] [2, 4]	T+11	2T+37
G4	DSACKi*[0] and DS*[L] to DS*[H] [2, 4, 8]	SBAT0+8	SBAT0+T+36
G5	DSACKi*[0] and DS*[L] to LEDI[L] [2, 4, 8]	SBAT0+13	SBAT0+T+52
G6	DSACKi*[0] and DS*[L] to UWDENIN*[L] [2, 4, 8]	SBAT0+8	SBAT0+T+31
G7	DSACKi*[0] and DS*[L] to LA(7:0) [2, 4, 8]	SBAT0+.5T+9	SBAT0+2T+29
G8	DS*[H] to DS*[L] [2, 4, 5]	T+8	3T+31

AC Timings for D64 Operations (Commercial)

Operation		Min.	Max.
Second Longword Cycle [2, 4]			
G9	DSACKi*[0] and DS*[L] to UWDENIN*[H]	SBAT1+20	SBAT1+T+64
G10	DSACKi*[0] and DS*[L] to DS*[H]	SBAT1+11	SBAT1+T+34
G11	DSACKi*[0] and DS*[L] to LA(7:0)	SBAT1+.5T+9	SBAT1+2T+29
G12	DSACKi*[0] and DS*[L] to LD(7:0)	SBAT1+.5T+11	SBAT1+2T+40
Slave D64 Block Transfer (Read)			
First Longword Cycle			
H1	DSACKi*[0] and DS*[L] to LEDO[H] [4, 8]	SBAT0+7	SBAT0+T+36
H2	DSACKi*[0] and DS*[L] to DS*[H] [4, 8]	SBAT0+8	SBAT0+T+39
H3	DSACKi*[0] and DS*[L] to LA(7:0) [2, 4, 8]	SBAT0+.5T+9	SBAT0+T+29
H4	DS*[H] to DS*[L] [2, 4, 5]	T+8	3T+30
Second Longword Cycle			
H5	DSACKi*[0] and DS*[L] to LEDO[L] [4]	SBAT1+19	SBAT1+T+64
H6	DSACKi*[0] and DS*[L] to DENO*[L] [4]	SBAT1+11	SBAT1+T+37
H7	DSACKi*[0] and DS*[L] to DS*[H] [2, 4]	SBAT1+24	SBAT1+T+72
H8	DSACKi*[0] & DS*[L] & DSi*[0] to DTACK*[L] [4]	SBAT1+13	SBAT1+T+33
H9	DSACKi*[0] and DS*[L] to LA(7:0) [2, 4]	SBAT1+.5T+9	SBAT1+2T+29
H10	DS1/0*[1] to DENO*[H] [2]	6	21
Slave D64 Block Transfer (Boundary Crossing)			
H11	DS*[0] to LADI[L] [2]	11	26
H12	DS*[1] to LADI[H] [2]	6	13

AC Timings for D64 Operations (Industrial)

Operation		Min.	Max.
Master D64 Block Transfer with Local DMA (Initiation Cycle) [2]			
A1	MWB*[0] & PAS*[0] & DS*(0) to BRi*[L]	T+7	2T+33
A2	MWB*[0] & PAS*[0] & DS*(0) to LADO[H]	T+8	2T+32
A3	MWB*[0] & PAS*[0] & DS*(0) to BLT*[L]	T+8	2T+27
A4	MWB*[0] & PAS*[0] & DS*(0) to DSACK1*[L]	T+10	2T+48
A5	MWB*[0] & PAS*[0] & DS*(0) to DSACK0*[L]	T+10	2T+48
Master D64 Block Transfer Address Broadcast Cycle [2]			
B1	DTACK*[0] to LBR*[L]	20	69
B2	DTACK*[0] to DSi*[H]	7	26
B3	DTACK*[0] to SCON*/D64[H]	15	62

AC Timings for D64 Operations (Industrial)

	Operation	Min.	Max.
Master D64 Block Transfer with Local DMA (Write)			
First Longword Fetch			
C1	DSACKi*[0] and DS*[L] to DS*[H] [2, 3, 4]	MBAT0+7	MBAT0+T+38
C2	DSACKi*[0] and DS*[L] to LEDO[H] [2, 3, 4]	MBAT0+6	MBAT0+T+35
C3	DSACKi*[0] and DS*[L] to LA(7:0) [2, 3, 4]	MBAT0+.5T+8	MBAT0+2T+32
C4	DS*(H) to DS*[L] [2, 3, 4, 5]	T+7	3T+32
Second Longword Fetch			
C5	DSACKi*[0] and DS*[L] to DS*[H] [2, 4]	MBAT1+13	MBAT1+T+48
C6	DSACKi*[0] and DS*[L] to DENO*[L] [4]	MBAT1+10	MBAT1+T+39
C7	DSACKi*[0] and DS*[L] to LA(7:0) [2, 4]	MBAT1+.5T+8	MBAT1+2T+32
C8	DSACKi*[0] and DS*[L] to DSi*[L] [6]	MBAT1+3T+10	MBAT1+4T+33
C9	DSACKi*[0] and DS*[L] to LEDO[L] [4]	MBAT1+15	MBAT1+T+57
C10	DS*(H) to DS*[L] [2, 4, 5]	T+7	3T+32
C11	DTACK*[0] to DSi*[H]	6	23
C12	DTACK*[0] to DENO*[H] [2]	7	26
Master Block Transfer with Local DMA (Read)			
First Longword Write			
D1	LBG*[0] to LWDENIN*[L] [2]	2T+10	3T+43
D2	DTACK*[0] to LEDI[H] [7]	2T+5	3T+24
D3	DTACK*[0] to DSi*[H] [2, 7]	2T+8	3T+29
D4	DTACK*[0] to DS*[L] [7]	2T+12	3T+37
D5	DSACKi*[0] and DS*[L] to DS*[H] [2, 3, 4]	MBAT0+7	MBAT0+T+38
D6	DSACKi*[0] and DS*[L] to LEDI[L] [3, 4]	MBAT0+12	MBAT0+T+53
D7	DSACKi*[0] and DS*[L] to UWDENIN*[L] [3, 4]	MBAT0+7	MBAT0+T+36
D8	DSACKi*[0] and DS*[L] to LA(7:0) [2, 3, 4]	MBAT0+.5T+8	MBAT0+2T+31
D9	DSACKi*[0] and DS*[L] to DSi*[L] [3, 4]	MBAT0+20	MBAT0+T+58
D10	DS*[H] to DS*[L] [2, 4, 5]	T+7	3T+31
Second Longword Write [2, 4]			
D12	DSACKi*[0] and DS*[L] to DS*[H]	MBAT1+7	MBAT1+T+38
D13	DSACKi*[0] and DS*[L] to UWDENIN*[H]	MBAT1+15	MBAT1+T+59
D14	DSACKi*[0] and DS*[L] to LA(7:0)	MBAT1+.5T+8	MBAT1+2T+31
D15	DSACKi*[0] and DS*[L] to LD(7:0)	MBAT1+.5T+10	MBAT1+2T+42
Master D64 Block Transfer with Local DMA (Boundary Crossing) [2]			
E1	DS*[0] to BLT*[H]	2	30
E2	DS*[1] to BLT*[L]	2	20

AC Timings for D64 Operations (Industrial)

	Operation	Min.	Max.
E3	DSi*[0] to LADO first transition	3	20
E4	DSi*[1] to LADO second transition	3	20
Slave D64 Block Transfer Address Broadcast Cycle [2]			
F1	DSi*[1] to LBR*[L]	10	39
F2	DSi*[0] to DTACK*[L]	2T+8	3T+29
F3	DSi*[1] to DTACK*[H]	8	35
F4	DSi*[1] to SCON*/D64[H]	9	35
F5	DSi*[0] and AS*[0] and SLSELi[0] to LADI[H]	1.5T+4	2T+26
Slave D64 Block Transfer (Write)			
First Longword Cycle			
G1	DSi*[0] to DS*[L] [2, 4]	3T+10	4T+40
G2	DSi*[0] to DTACK*[L] [7]	2T+8	3T+24
G3	DSi*[0] to LEDI[H] [2, 4]	T+10	2T+39
G4	DSACKi*[0] and DS*[L] to DS*[H] [2, 4, 8]	SBAT0+7	SBAT0+T+38
G5	DSACKi*[0] and DS*[L] to LEDI[L] [2, 4, 8]	SBAT0+12	SBAT0+T+54
G6	DSACKi*[0] and DS*[L] to UWDENIN*[L] [2, 4, 8]	SBAT0+7	SBAT0+T+33
G7	DSACKi*[0] and DS*[L] to LA(7:0) [2, 4, 8]	SBAT0+.5T+8	SBAT0+2T+31
G8	DS*[H] to DS*[L] [2, 4, 5]	T+7	3T+32
Second Longword Cycle [2, 4]			
G9	DSACKi*[0] and DS*[L] to UWDENIN*[H]	SBAT1+19	SBAT1+T+67
G10	DSACKi*[0] and DS*[L] to DS*[H]	SBAT1+10	SBAT1+T+36
G11	DSACKi*[0] and DS*[L] to LA(7:0)	SBAT1+.5T+8	SBAT1+2T+31
G12	DSACKi*[0] and DS*[L] to LD(7:0)	SBAT1+.5T+10	SBAT1+2T+42
Slave D64 Block Transfer (Read)			
First Longword Cycle [4, 8]			
H1	DSACKi*[0] and DS*[L] to LEDO[H]	SBAT0+6	SBAT0+T+37
H2	DSACKi*[0] and DS*[L] to DS*[H] [4, 8]	SBAT0+7	SBAT0+T+41
H3	DSACKi*[0] and DS*[L] to LA(7:0) [2, 4, 8]	SBAT0+.5T+8	SBAT0+T+31
H4	DS*[H] to DS*[L] [2, 4, 5]	T+7	3T+32
Second Longword Cycle			
H5	DSACKi*[0] and DS*[L] to LEDO[L] [4]	SBAT1+18	SBAT1+T+67
H6	DSACKi*[0] and DS*[L] to DENO*[L] [4]	SBAT1+10	SBAT1+T+39
H7	DSACKi*[0] and DS*[L] to DS*[H] [2, 4]	SBAT1+23	SBAT1+T+75
H8	DSACKi*[0] & DS*[L] & DSi*[0] to DTACK*[L] [4]	SBAT1+12	SBAT1+T+34

AC Timings for D64 Operations (Industrial)

	Operation	Min.	Max.
H9	DSACKi*[0] and DS*[L] to LA(7:0) [2, 4]	SBAT1+.5T+8	SBAT1+2T+31
H10	DS1/0*[1] to DENO*[H] [2]	5	22
Slave D64 Block Transfer (Boundary Crossing) [2]			
H11	DS*[0] to LADI[L]	10	28
H12	DS*[1] to LADI[H]	5	14

AC Timings for D64 Operations (Military)

	Operation	Min.	Max.
Master D64 Block Transfer with Local DMA (Initiation Cycle) [2]			
A1	MWB*[0] & PAS*[0] & DS*(0) to BRi*[L]	4T+5	5T+38
A2	MWB*[0] & PAS*[0] & DS*(0) to LADO[H]	T+8	2T+38
A3	MWB*[0] & PAS*[0] & DS*(0) to BLT*[L]	T+8	2T+30
A4	MWB*[0] & PAS*[0] & DS*(0) to DSACK1*[L]	T+10	2T+54
A5	MWB*[0] & PAS*[0] & DS*(0) to DSACK0*[L]	T+10	2T+54
Master D64 Block Transfer Address Broadcast Cycle [2]			
B1	DTACK*[0] to LBR*[L]	20	75
B2	DTACK*[0] to DSi*[H]	7	30
B3	DTACK*[0] to SCON*/D64[H]	15	70
Master D64 Block Transfer with Local DMA (Write)			
First Longword Fetch			
C1	DSACKi*[0] and DS*[L] to DS*[H] [2, 3, 4]	MBAT0+7	MBAT0+T+42
C2	DSACKi*[0] and DS*[L] to LEDO[H] [2, 3, 4]	MBAT0+6	MBAT0+T+36
C3	DSACKi*[0] and DS*[L] to LA(7:0) [2, 3, 4]	MBAT0+.5T+8	MBAT0+2T+35
C4	DS*(H) to DS*[L] [2, 3, 4, 5]	T+7	3T+35
Second Longword Fetch			
C5	DSACKi*[0] and DS*[L] to DS*[H] [2, 4]	MBAT1+13	MBAT1+T+52
C6	DSACKi*[0] and DS*[L] to DENO*[L] [4]	MBAT1+10	MBAT1+T+42
C7	DSACKi*[0] and DS*[L] to LA(7:0) [2, 4]	MBAT1+.5T+8	MBAT1+2T+35
C8	DSACKi*[0] and DS*[L] to DSi*[L] [6]	MBAT1+3T+10	MBAT1+4T+36
C9	DSACKi*[0] and DS*[L] to LEDO[L] [4]	MBAT1+15	MBAT1+T+64
C10	DS*(H) to DS*[L] [2, 4, 5]	T+7	3T+35
C11	DTACK*[0] to DSi*[H]	6	25
C12	DTACK*[0] to DENO*[H] [2]	7	30

AC Timings for D64 Operations (Military)

	Operation	Min.	Max.
Master Block Transfer with Local DMA (Read)			
First Longword Write			
D1	LBG*[0] to LWDENIN*[L] [2]	2T+10	3T+48
D2	DTACK*[0] to LEDI[H] [7]	2T+5	3T+27
D3	DTACK*[0] to DSi*[H] [2, 7]	2T+8	3T+32
D4	DTACK*[0] to DS*[L] [7]	2T+12	3T+41
D5	DSACKi*[0] and DS*[L] to DS*[H] [2, 3, 4]	MBAT0+7	MBAT0+T+42
D6	DSACKi*[0] and DS*[L] to LEDI[L] [3, 4]	MBAT0+12	MBAT0+T+60
D7	DSACKi*[0] and DS*[L] to UWDENIN*[L] [3, 4]	MBAT0+7	MBAT0+T+41
D8	DSACKi*[0] and DS*[L] to LA(7:0) [2, 3, 4]	MBAT0+.5T+8	MBAT0+2T+35
D9	DSACKi*[0] and DS*[L] to DSi*[L] [3, 4]	MBAT0+20	MBAT0+T+64
D10	DS*[H] to DS*[L] [2, 4, 5]	T+7	3T+35
Second Longword Write [2, 4]			
D12	DSACKi*[0] and DS*[L] to DS*[H]	MBAT1+7	MBAT1+T+42
D13	DSACKi*[0] and DS*[L] to UWDENIN*[H]	MBAT1+15	MBAT1+T+66
D14	DSACKi*[0] and DS*[L] to LA(7:0)	MBAT1+.5T+8	MBAT1+2T+35
D15	DSACKi*[0] and DS*[L] to LD(7:0)	MBAT1+.5T+10	MBAT1+2T+48
Master D64 Block Transfer with Local DMA (Boundary Crossing) [2]			
E1	DS*[0] to BLT*[H]	2	33
E2	DS*[1] to BLT*[L]	2	21
E3	DSi*[0] to LADO first transition	2	21
E4	DSi*[1] to LADO second transition	2	21
Slave D64 Block Transfer Address Broadcast Cycle [2]			
F1	DSi*[1] to LBR*[L]	10	42
F2	DSi*[0] to DTACK*[L]	2T+8	3T+32
F3	DSi*[1] to DTACK*[H]	8	39
F4	DSi*[1] to SCON*/D64[H]	9	39
F5	DSi*[0] and AS*[0] and SLSELi[0] to LADI[H]	1.5T+4	2T+29
Slave D64 Block Transfer (Write)			
First Longword Cycle			
G1	DSi*[0] to DS*[L] [2, 4]	3T+10	4T+44
G2	DSi*[0] to DTACK*[L] [7]	2T+8	3T+26
G3	DSi*[0] to LEDI[H] [2, 4]	T+10	2T+42
G4	DSACKi*[0] and DS*[L] to DS*[H] [2, 4, 8]	SBAT0+7	SBAT0+T+42
G5	DSACKi*[0] and DS*[L] to LEDI[L] [2, 4, 8]	SBAT0+12	SBAT0+T+60

AC Timings for D64 Operations (Military)

	Operation	Min.	Max.
G6	DSACKi*[0] and DS*[L] to UWDENIN*[L] [2, 4, 8]	SBAT0+7	SBAT0+T+37
G7	DSACKi*[0] and DS*[L] to LA(7:0) [2, 4, 8]	SBAT0+.5T+8	SBAT0+2T+35
G8	DS*[H] to DS*[L]	T+7	3T+35
Second Longword Cycle [2, 4]			
G9	DSACKi*[0] and DS*[L] to UWDENIN*[H]	SBAT1+19	SBAT1+T+74
G10	DSACKi*[0] and DS*[L] to DS*[H]	SBAT1+10	SBAT1+T+42
G11	DSACKi*[0] and DS*[L] to LA(7:0)	SBAT1+.5T+8	SBAT1+2T+35
G12	DSACKi*[0] and DS*[L] to LD(7:0)	SBAT1+.5T+10	SBAT1+2T+48
Slave D64 Block Transfer (Read)			
First Longword Cycle			
H1	DSACKi*[0] and DS*[L] to LEDO[H] [4, 8]	SBAT0+6	SBAT0+T+41
H2	DSACKi*[0] and DS*[L] to DS*[H] [4, 8]	SBAT0+7	SBAT0+T+45
H3	DSACKi*[0] and DS*[L] to LA(7:0) [2, 4, 8]	SBAT0+.5T+8	SBAT0+T+35
H4	DS*[H] to DS*[L] [2, 4, 5]	T+7	3T+35
Second Longword Cycle			
H5	DSACKi*[0] and DS*[L] to LEDO[L] [4]	SBAT1+18	SBAT1+T+74
H6	DSACKi*[0] and DS*[L] to DENO*[L] [4]	SBAT1+10	SBAT1+T+42
H7	DSACKi*[0] and DS*[L] to DS*[H] [2, 4]	SBAT1+23	SBAT1+T+85
H8	DSACKi*[0] & DS*[L] & DSi*[0] to DTACK*[L] [4]	SBAT1+12	SBAT1+T+38
H9	DSACKi*[0] and DS*[L] to LA(7:0) [2, 4]	SBAT1+.5T+8	SBAT1+2T+35
H10	DS1/0*[1] to DENO*[H] [2]	5	25
Slave D64 Block Transfer (Boundary Crossing) [2]			
H11	DS*[0] to LADI[L]	10	32
H12	DS*[1] to LADI[H]	5	15

Notes:

1. All minimum times are guaranteed, not tested.
2. These timings are specified for information, but not tested.
3. For second and all subsequent longword fetches, MBAT1 is used in the timing equations.
4. When the Enhanced Turbo Bit is set, all these times are reduced by 0.5T.
5. Min and Max Times are programmable: see Register Descriptions.
6. When the Enhanced Turbo Bit is set, these times become MBAT1+.5T+D min., MBAT1+1.5T +D max.
7. When the Enhanced Turbo Bit is set, all these items are reduced to 0.5T min., 1.0T max, plus ap propriate asynchronous delay from the table. Minimum times reflect unloaded device pins. Actual in-system delays will be in accordance with the VMEbus specification.
8. For second and all subsequent longword fetches, SBAT1 is used in the timing equations.

DC Performance Specifications

VMEbus Signals (AS*, DS1*, DS0*, BCLR*, SYSCLK)

Parameter	Description	Test Conditions		Comm.	Industrial	Military	Units
V_{IH}	Minimum high-level input voltage			2.0	2.0	2.0	V
V_{IL}	Maximum low-level input voltage			0.8	0.8	0.8	V
V_{OH}	Minimum high-level output voltage	V_{CC} = Min., I_{OH} = −3 mA		2.4	2.4	2.4	V
V_{OL}	Minimum low-level output voltage	V_{CC} = Min., I_{OL} = 64 mA (com'l), 56 mA (Ind.), 48 mA (Mil.)		0.6	0.6	0.6	V
I_L	Maximum input leakage current	V_{CC} = Max., V_{IN} = 0.6−2.4		±5	±5	±5	µA
V_{IK}	Input clamp voltage	V_{CC} = Min.	I_{IN} = −18 mA	−1.2	−1.2	−1.2	V
			I_{IN} = 18 mA	V_{CC}+1.2	V_{CC}+1.2	V_{CC}+1.2	V
I_{OZ}	Maximum output leakage current	V_{CC} = Max., GND ≤ V_{OUT} ≤ V_{CC}, all outputs disabled		±10	±10	±10	µA
V_{IK}	Input clamp voltage	V_{CC} = Min.	I_{IN} = −18 mA	−1.2	−1.2	−1.2	V
			I_{IN} = 18 mA	V_{CC}+1.2	V_{CC}+1.2	V_{CC}+1.2	V
I_{OZ}	Maximum output leakage current	V_{CC} = Max., V_{OUT} = 0.6/2.4V, all outputs disabled		±5	±5	±10	µA

VMEbus Signals (Low Drive. All VMEbus, Daisy-Chain Signals.)

Parameter	Description	Test Conditions		Comm.	Industrial	Military	Units
V_{IH}	Maximum high-level input voltage			2.0	2.0	2.0	V
V_{IL}	Maximum low-level input voltage			0.8	0.8	0.8	V
V_{OH}	Minimum high-level output voltage	V_{CC} = Min., I_{OH} = −8 mA		2.4	2.4	2.4	V
V_{OL}	Minimum low-level output voltage	V_{CC} = Min., I_{OL} = 8 mA		0.6	0.6	0.6	V
I_L	Maximum input leakage current	V_{CC} = Max., V_{IN} = 0.6−2.4		±5	±5	±5	µA
V_{IK}	Input clamp voltage	V_{CC} = Min.	I_{IN} = −18 mA	−1.2	−1.2	−1.2	V
			I_{IN} = 18 mA	V_{CC}+1.2	V_{CC}+1.2	V_{CC}+1.2	V
I_{OZ}	Maximum output leakage current	V_{CC} = Max., V_{OUT} = 0.6/2.4V, all outputs disabled		±5	±5	±10	µA

VMEbus Signals (Medium Drive. All Non-High, Non-Low Drive Signals, All VAC068A VMEbus Signals.)

Parameter	Description	Test Conditions	Comm.	Industrial	Military	Units
V_{IH}	Maximum high-level input voltage		2.0	2.0	2.0	V
V_{IL}	Maximum low-level input voltage		0.8	0.8	0.8	V

VMEbus Signals (Medium Drive. All Non-High, Non-Low Drive Signals, All VAC068A VMEbus Signals.) (continued)

Parameter	Description	Test Conditions		Comm.	Industrial	Military	Units
V_{OH}	Minimum high-level output voltage	V_{CC} = Min., I_{OH} = −3 mA		2.4	2.4	2.4	V
V_{OL}	Minimum low-level output voltage	V_{CC} = Min., I_{OL} = 48 mA		0.6	0.6	0.6	V
I_L	Maximum input leakage current	V_{CC} = Max., V_{IN} = 0.6−2.4		±5	±5	±5	µA
V_{IK}	Input clamp voltage	V_{CC} = Min.	I_{IN} = −18 mA	−1.2	−1.2	−1.2	V
			I_{IN} = 18 mA	V_{CC}+1.2	V_{CC}+1.2	V_{CC}+1.2	V
I_{OZ}	Maximum output leakage current	V_{CC} = Max., V_{OUT} = 0.6/2.4V, all outputs disabled		±5	±5	±10	µA

Non-VMEbus Signals

Parameter	Description	Test Conditions	Comm.	Industrial	Military	Units
V_{IH}	Maximum High-Level Input Voltage		2.0	2.0	2.0	V
V_{IL}	Maximum Low-Level Input Voltage		0.8	0.8	0.8	V
V_{OH}	Minimum High-Level Output Voltage	V_{CC} = Min., I_{OH} = −8 mA	2.4	2.4	2.4	V
V_{OL}	Minimum Low-Level Output Voltage	V_{CC} = Min., I_{OL} = 8 mA	0.6	0.6	0.6	V

Non-VMEbus Signals (Continued)

Parameter	Description	Test Conditions		Comm.	Industrial	Military	Units
I_L	Maximum Input Leakage Current	V_{CC} = Max., V_{IN} = 0.00/V_{CC}		±5	±5	±5	µA
V_{IK}	Input Clamp Voltage	V_{CC} = Min.	I_{IN} = −18 mA	−1.2	−1.2	−1.2	V
			I_{IN} = 18 mA	V_{CC}+1.2	V_{CC}+1.2	V_{CC}+1.2	V
I_{OZ}	Maximum Output Leakage Current	V_{CC} = Max. GND ≤ V_{OUT} ≤ V_{CC} All Outputs Disabled		±5		±10	µA

Capacitance

C_{IN}	Input Capacitance	T_A = 25°C, f = 64 MHz, V_{CC} = 5.0V	5	pF
C_{OUT}	Output Capacitance		7	pF

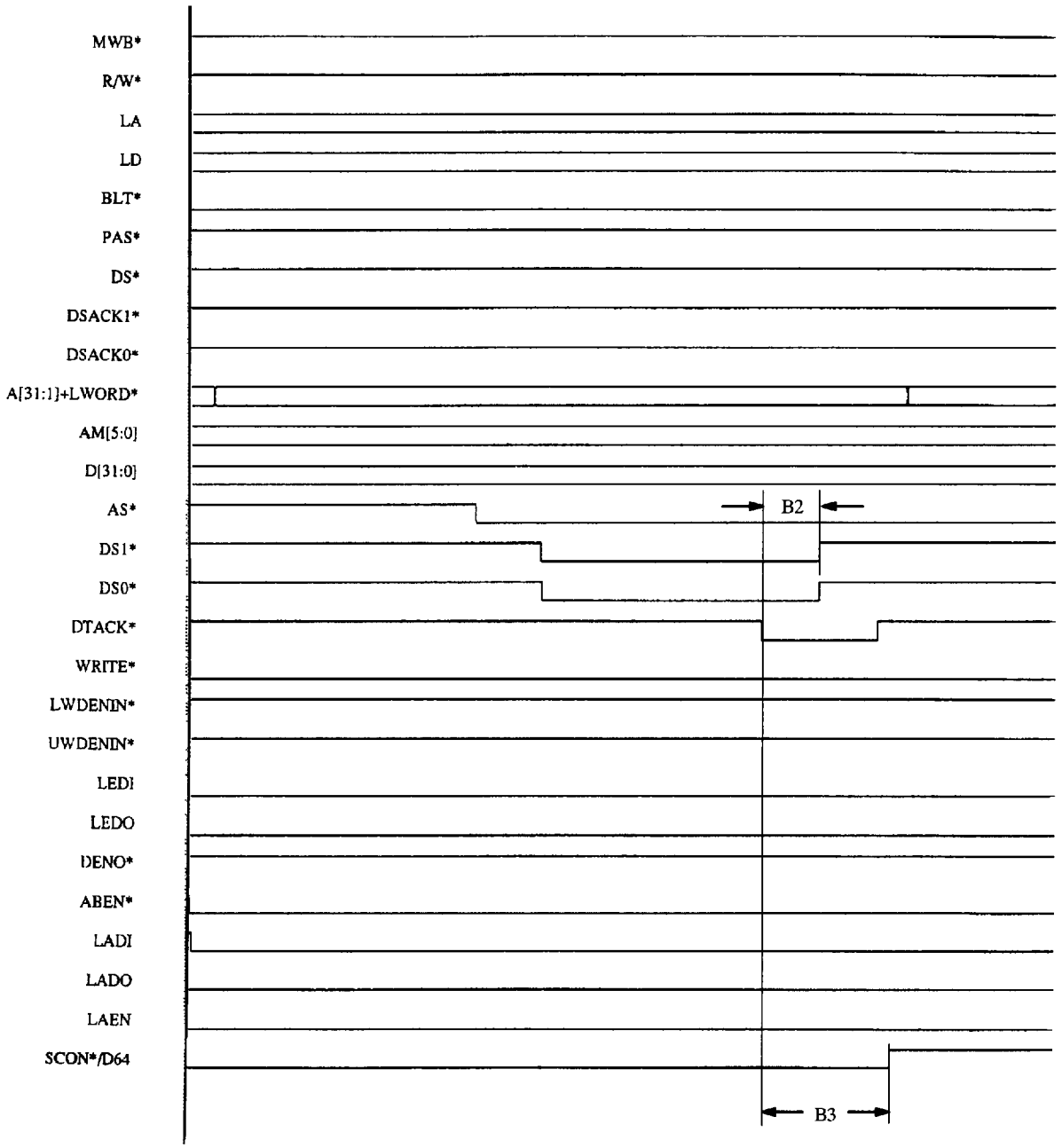


Figure 1. Master Address Broadcast Cycle

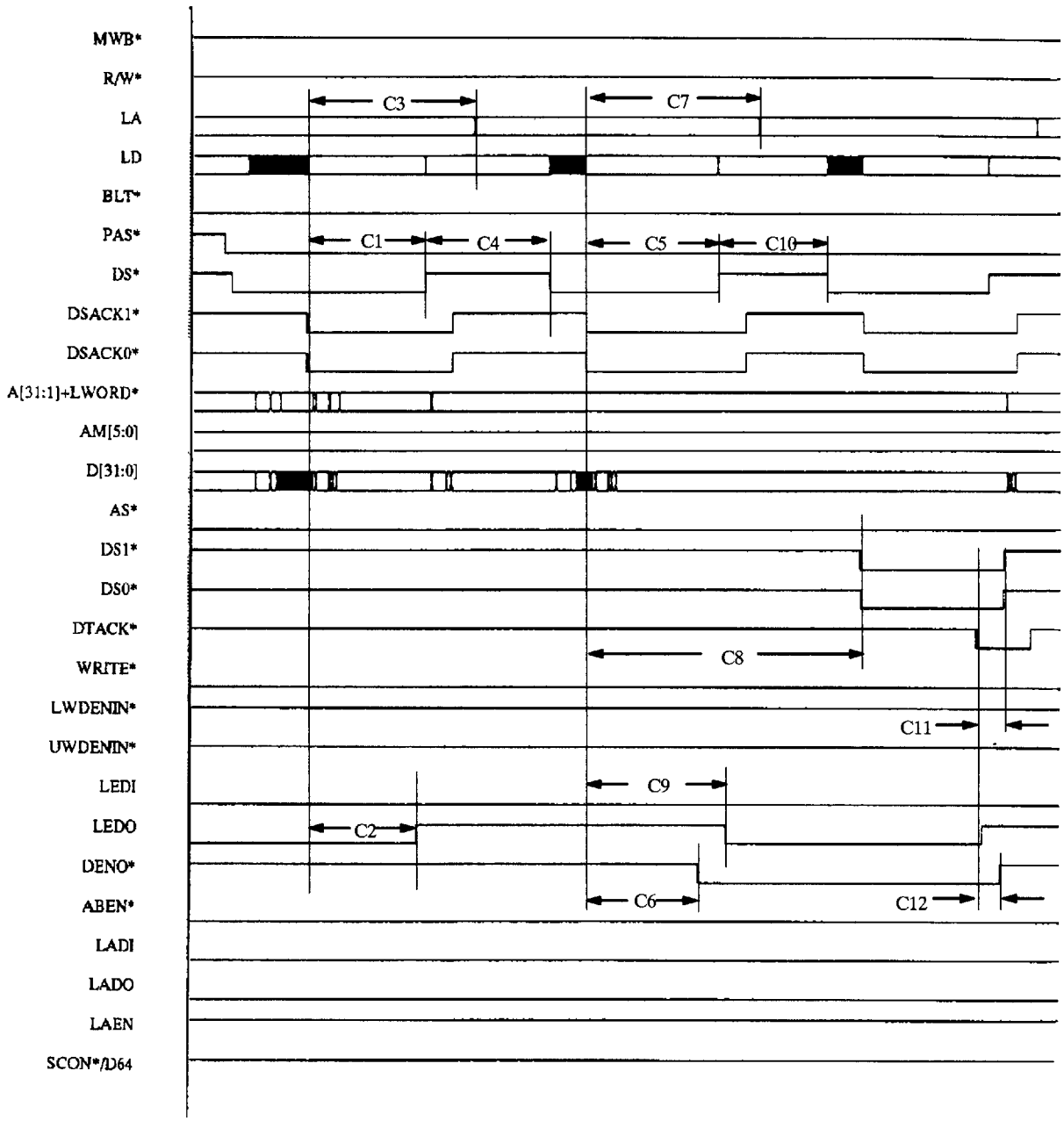


Figure 2. Master D64 Write Operation: Detail

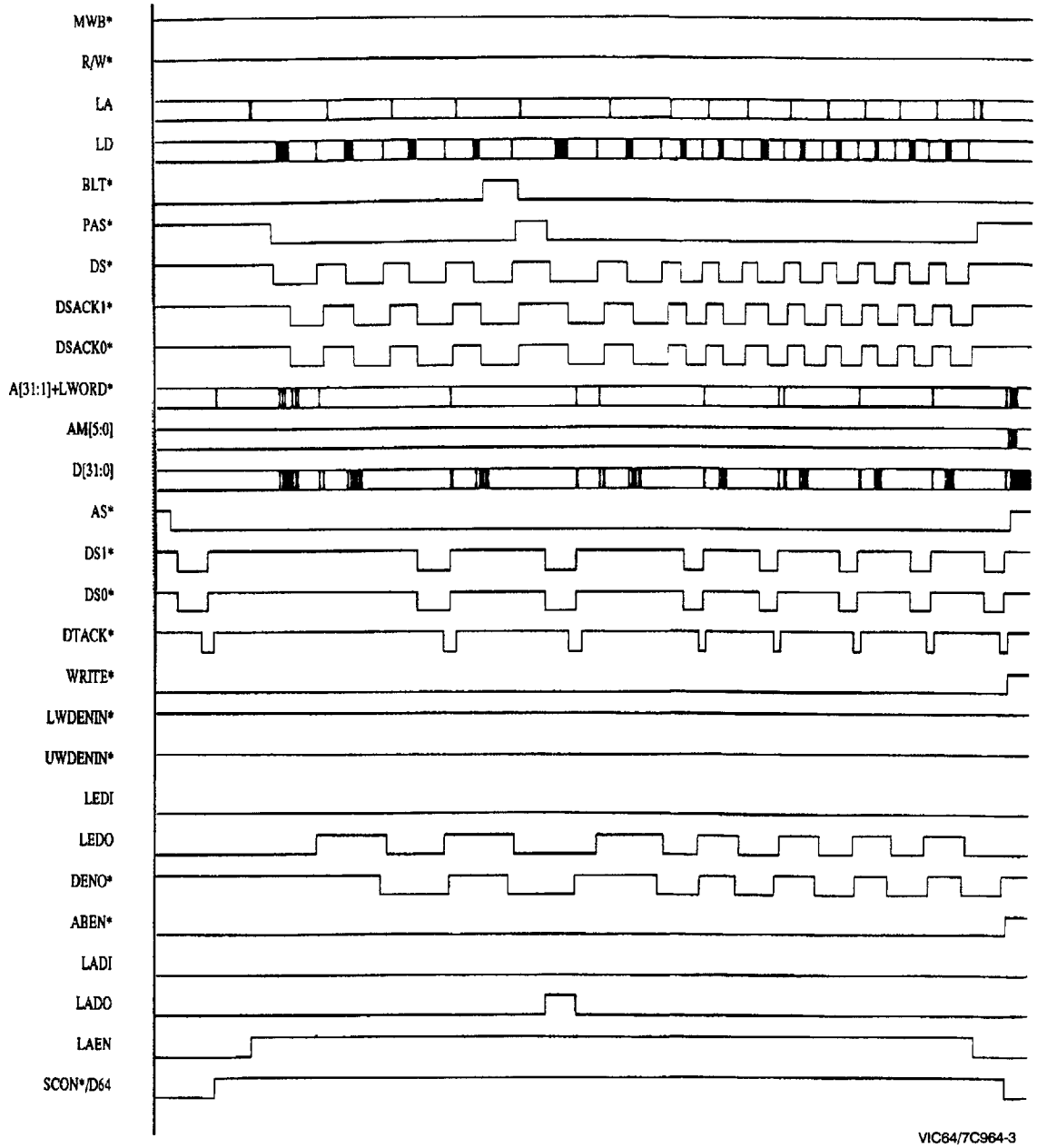


Figure 3. Master D64 Write Operation: Block Transfer

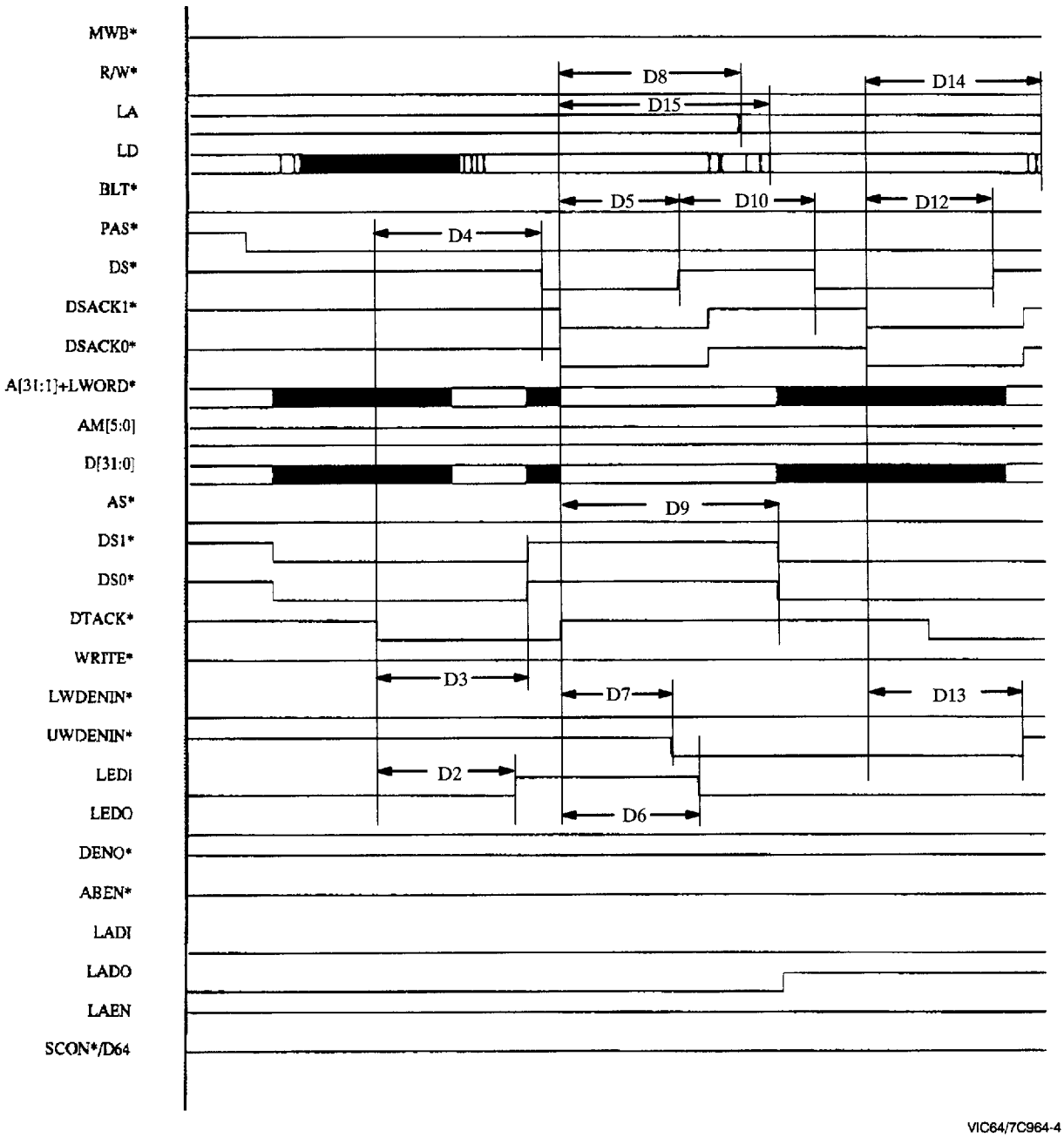


Figure 4. Master D64 Read Operation: Detail

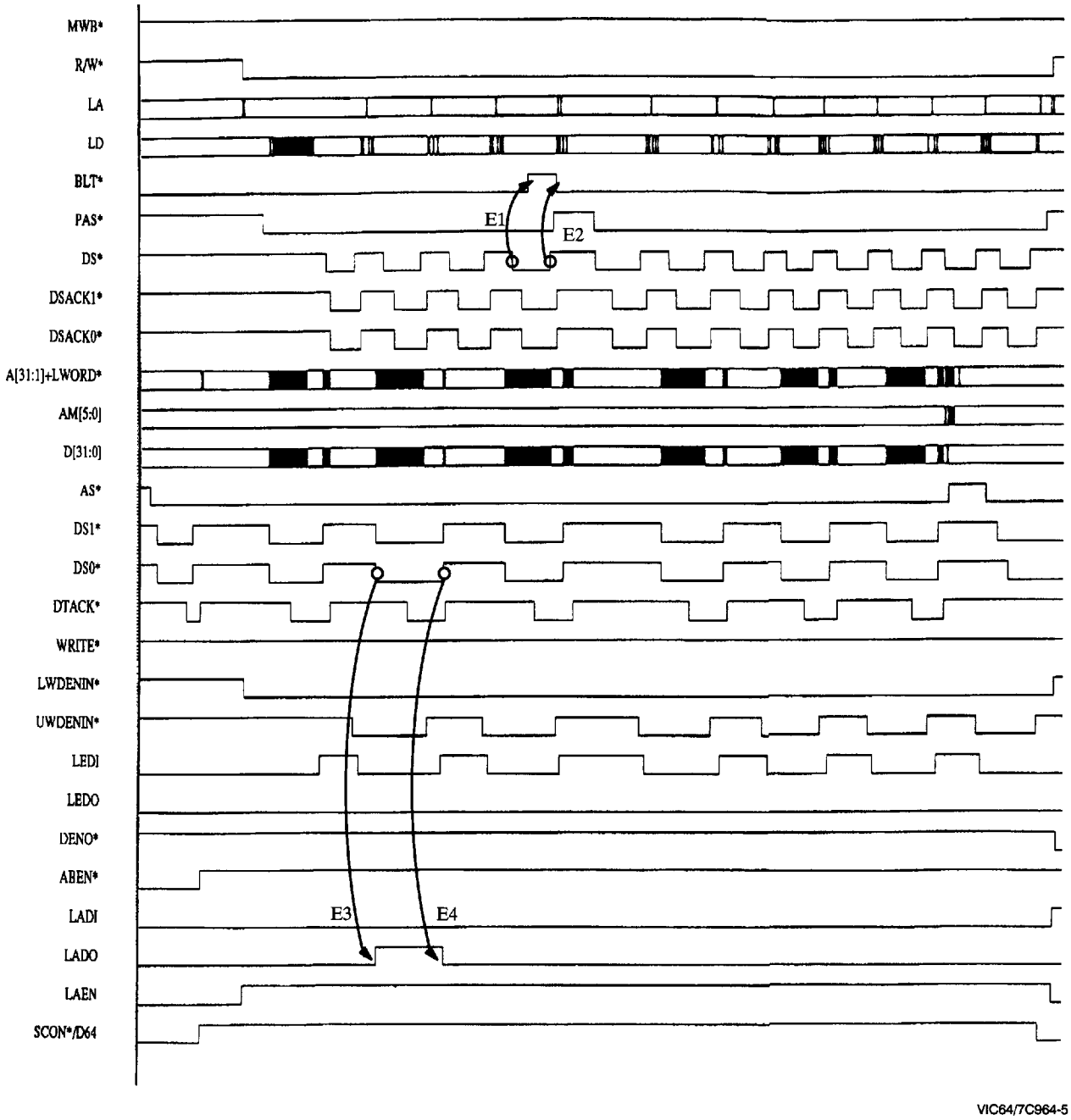


Figure 5. Master D64 Read Operation: Block Transfer

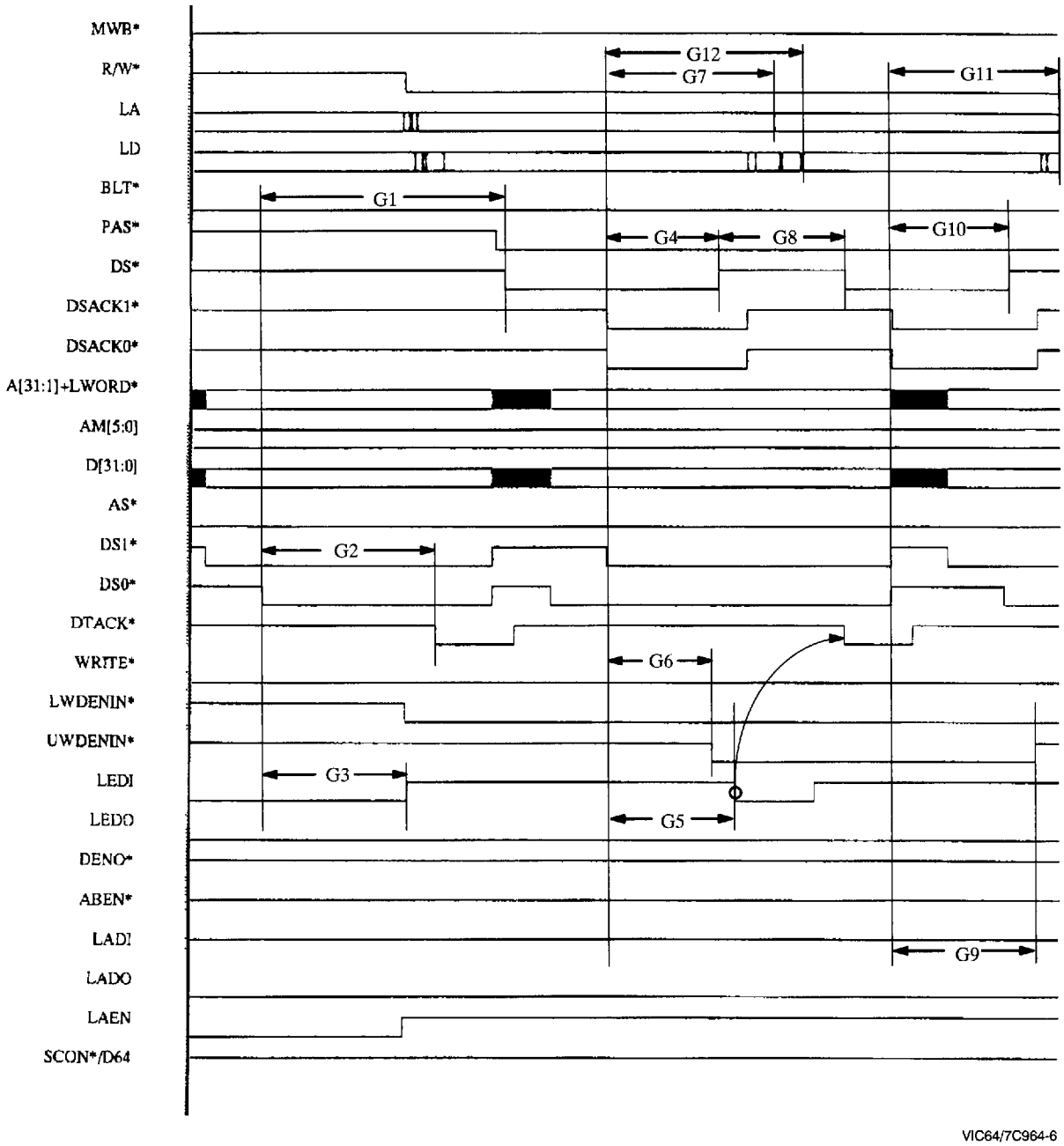


Figure 6. Slave D64 Write Operation: Detail

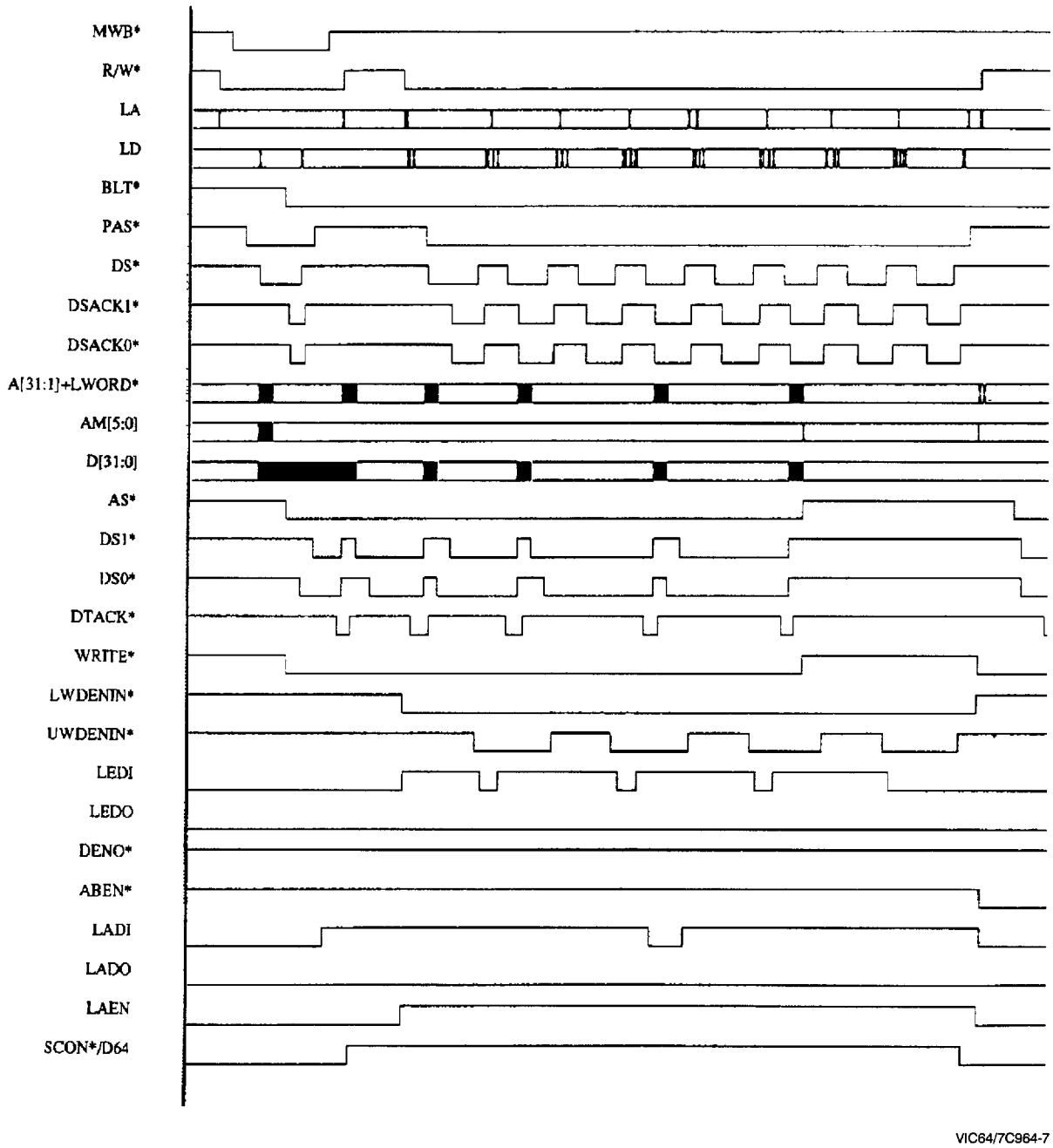


Figure 7. Slave D64 Write Operation: Block Transfer

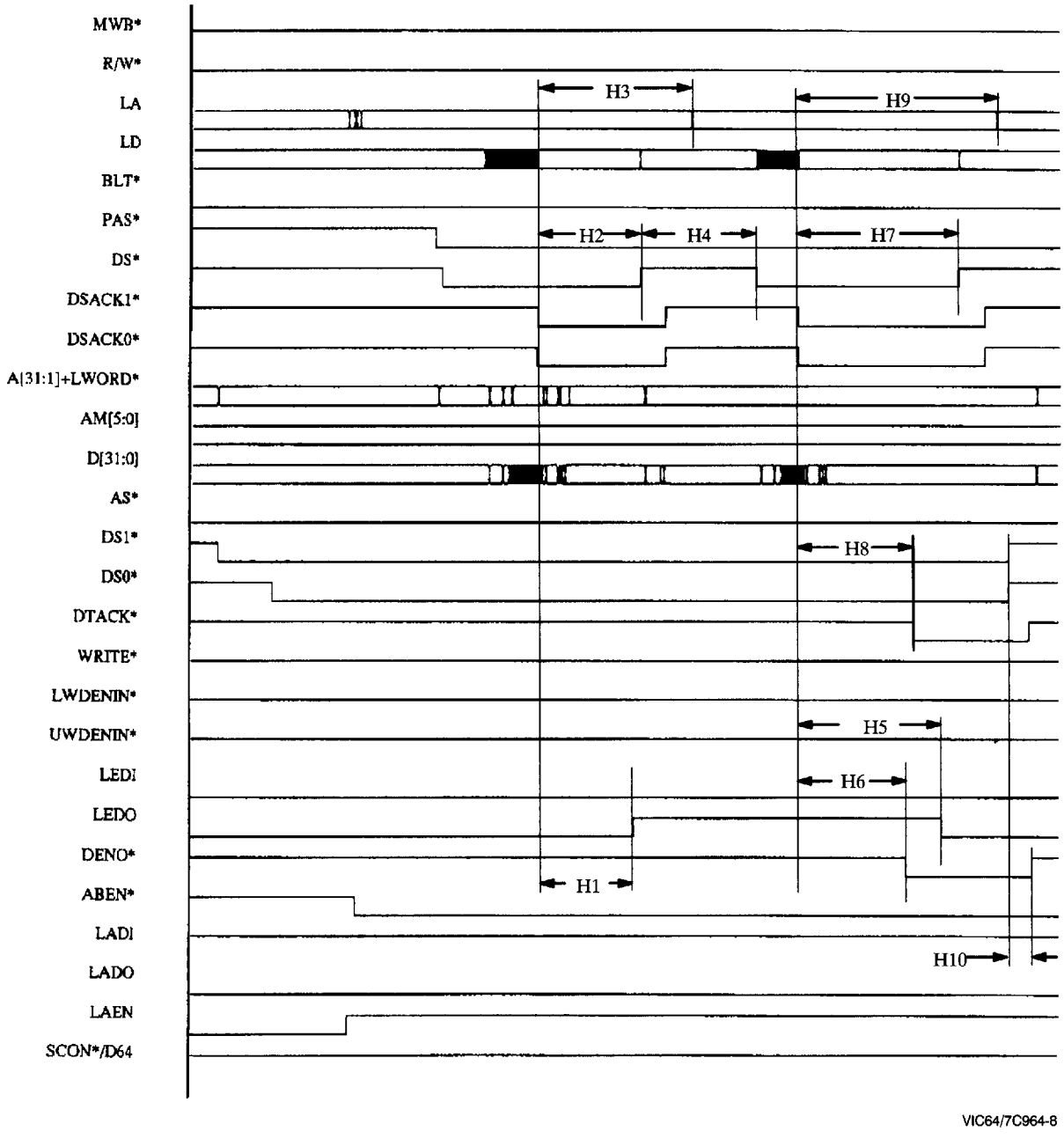


Figure 8. Slave D64 Read Operation: Detail

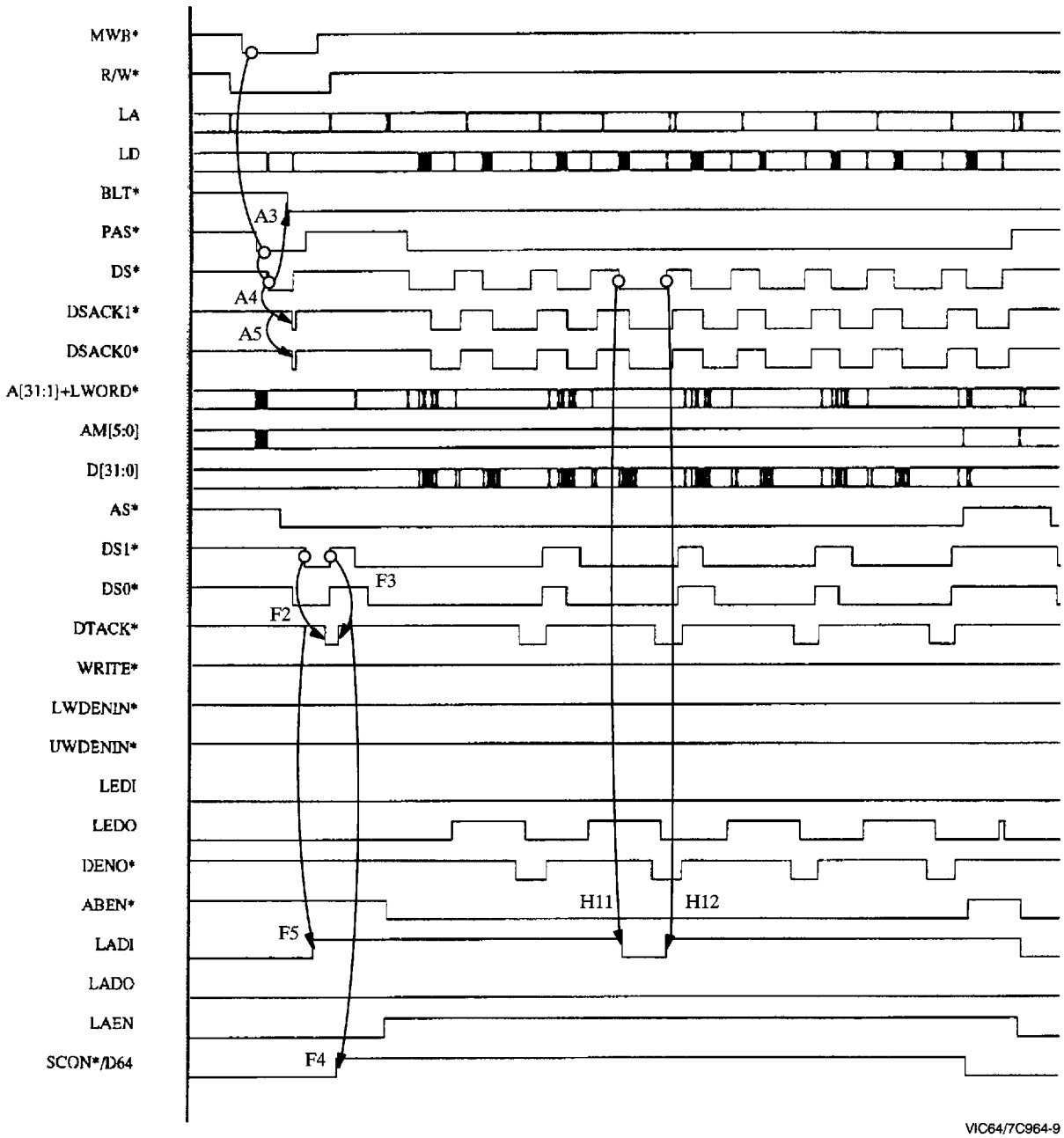


Figure 9. Slave D64 Read Operation: Block Transfer

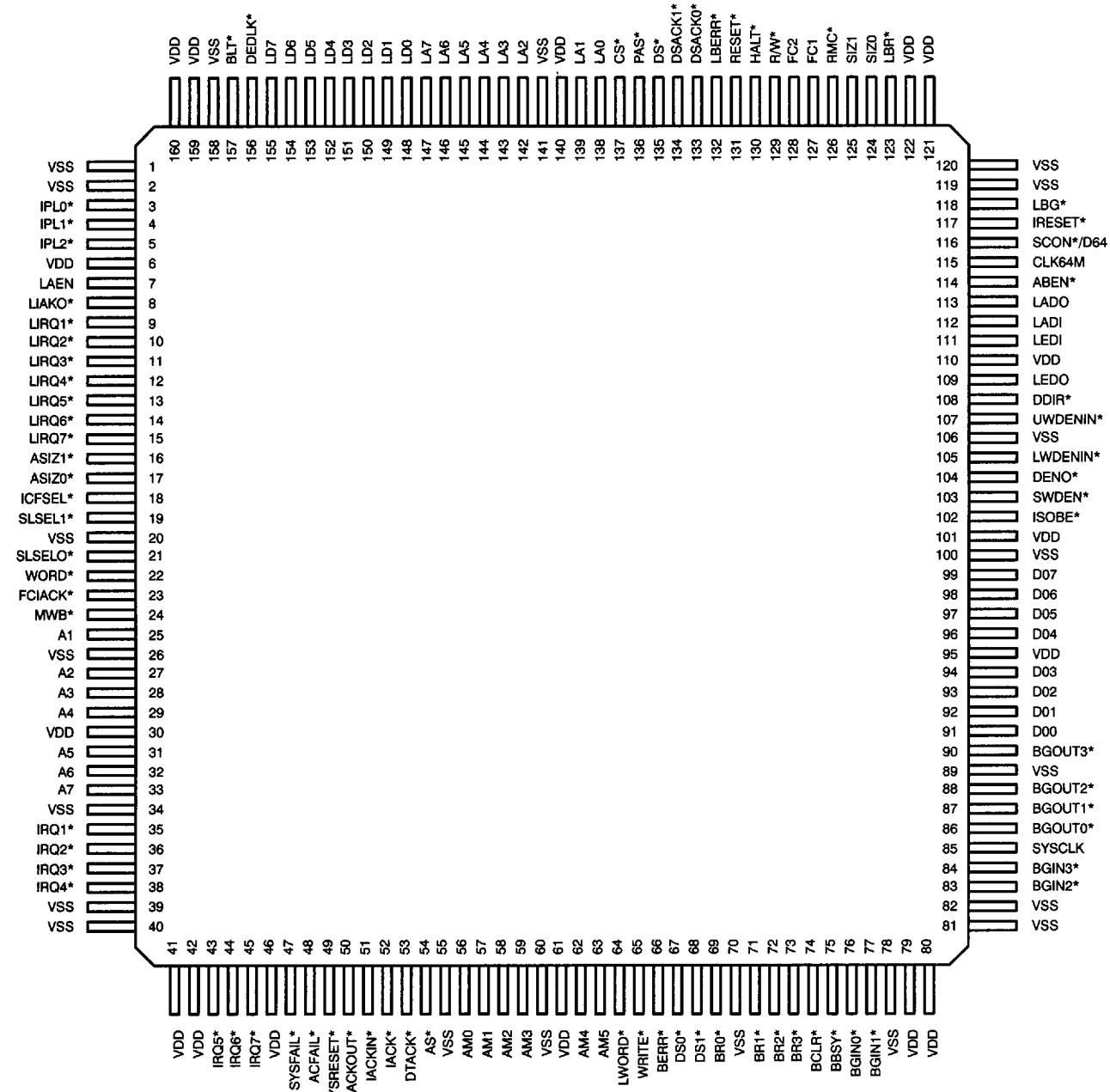
Pin Configurations

Bottom View
PGA

A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	
VSS	IPL2*	LIACKO*	LIRQ2*	LIRQ5*	ASIZ1*	ASIZ0*	SLSEL1*	WORD	FIACK*	A02	A04	VDD	VSS	IRQ4*	1
LD6	BLT*	IPL1*	VDD	LIRQ1*	LIRQ4*	LIRQ6*	ICFSEL*	MWB*	A01	A03	A05	A07	IRQ3*	IRQ7*	2
LD2	LD5	DEDLK*	IPL0*	LAEN	LIRQ3*	LIRQ7*	VSS	SLSEL0*	VSS	A06	IRQ1*	IRQ2*	IRQ6*	ACFAIL*	3
LD1	LD3	LD7	LOCATOR PIN									IRQ5*	VDD	IACKOUT*	4
LA7	LD0	LD4										SYSFAIL*	SYSRESET*	DTACK*	5
LA3	LA5	LA6										IACKIN*	IACK*	AM0	6
LA2	LA4	VSS										VSS	AS*	AM1	7
LA1	LA0	VCC7										VSS	AM2	AM3	8
CS*	DSACK1*	$\overline{DS}$										VDD	LWORD*	AM4	9
PAS*	LBERR*	RESET*										BERR*	WRITE*	AM5	10
DSACK0*	R/W*	FC1										BR2*	DS1*	DS0*	11
HALT*	RMC*	LBR*										BBSY*	BR1*	BR0*	12
FC2	SIZ0	SCON*/D64	CLK64M	LADI	VSS	VDD	VSS8	VCC5	D00	BG1OUT*	BG2IN*	BG0IN*	$\overline{BR3}$	VSS	13
SIZ1	IRESET*	LADO	LEDI*	DDIR*	LWDENIN*	DENO*	D06	D03	D01	VSS7	BG0OUT*	BG3IN*	BG1IN*	BCLR*	14
LBG*	ABEN*	VDD	LEDO	UWDENIN*	SWDEN*	ISOBE*	D07	D05	D04	D02	BG3OUT*	BG2OUT*	SYSCLK	VSS	15

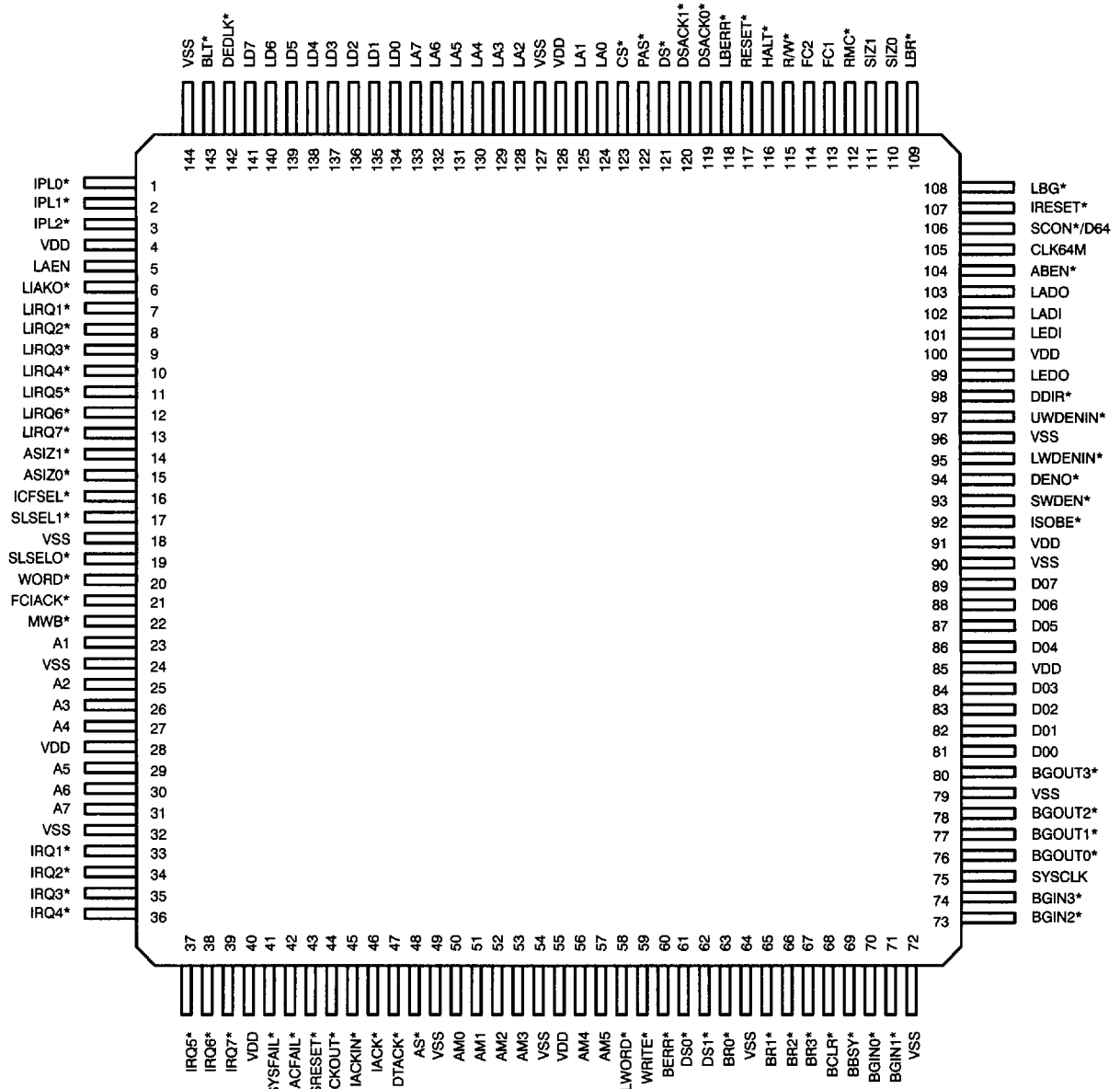
VIC64/7C964-10

Top View
PQFP/CQFP



VIC64/7C964-11

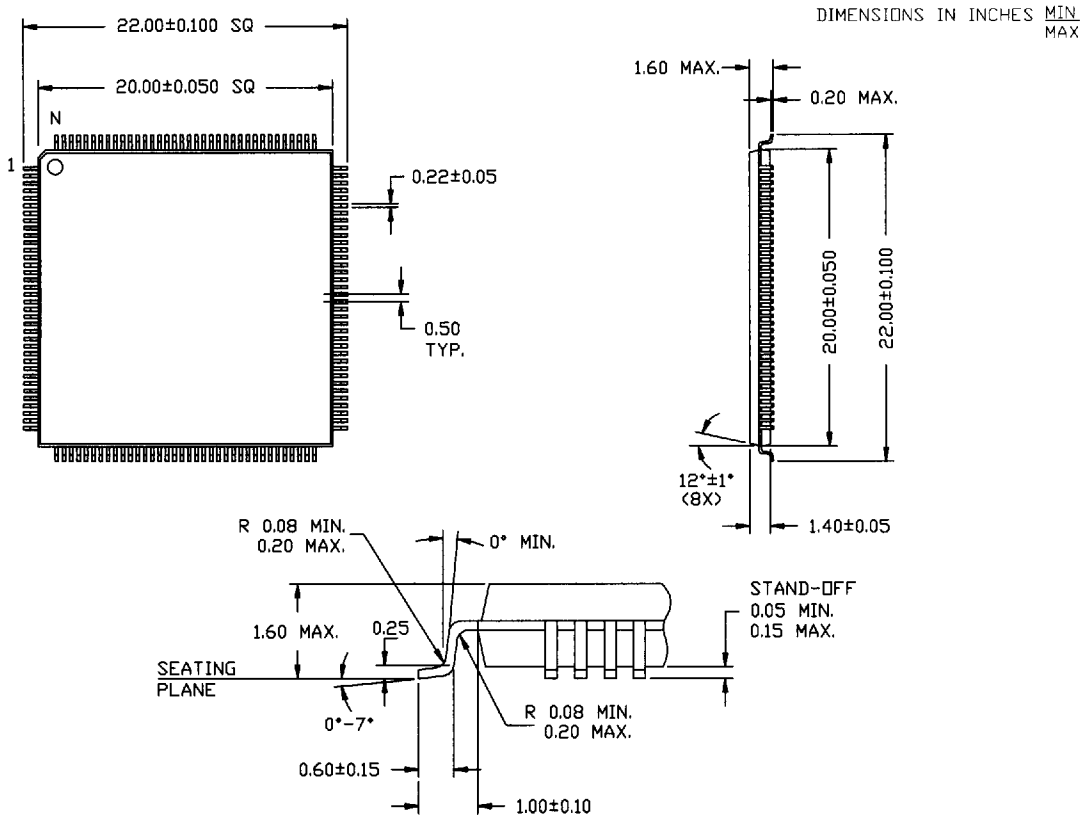
Top View
TQFP



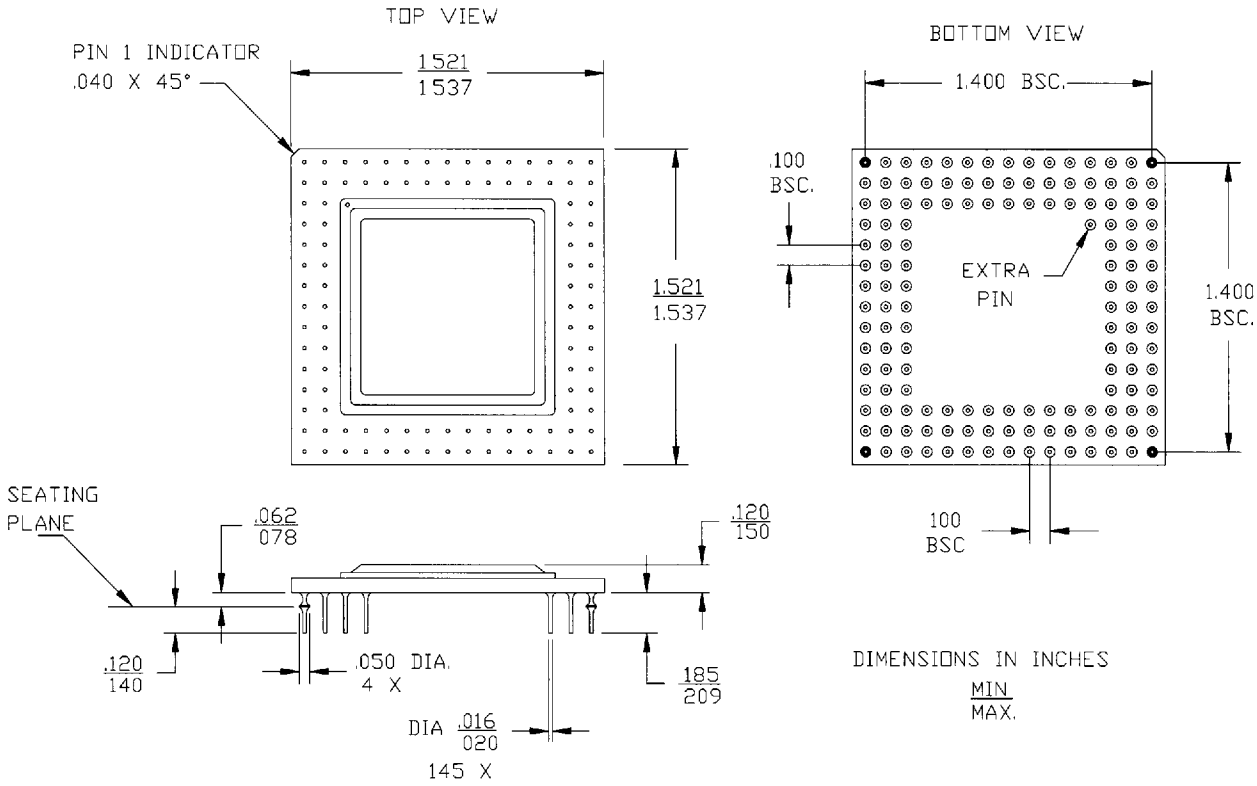
VIC64/7C964-12

Package Diagrams

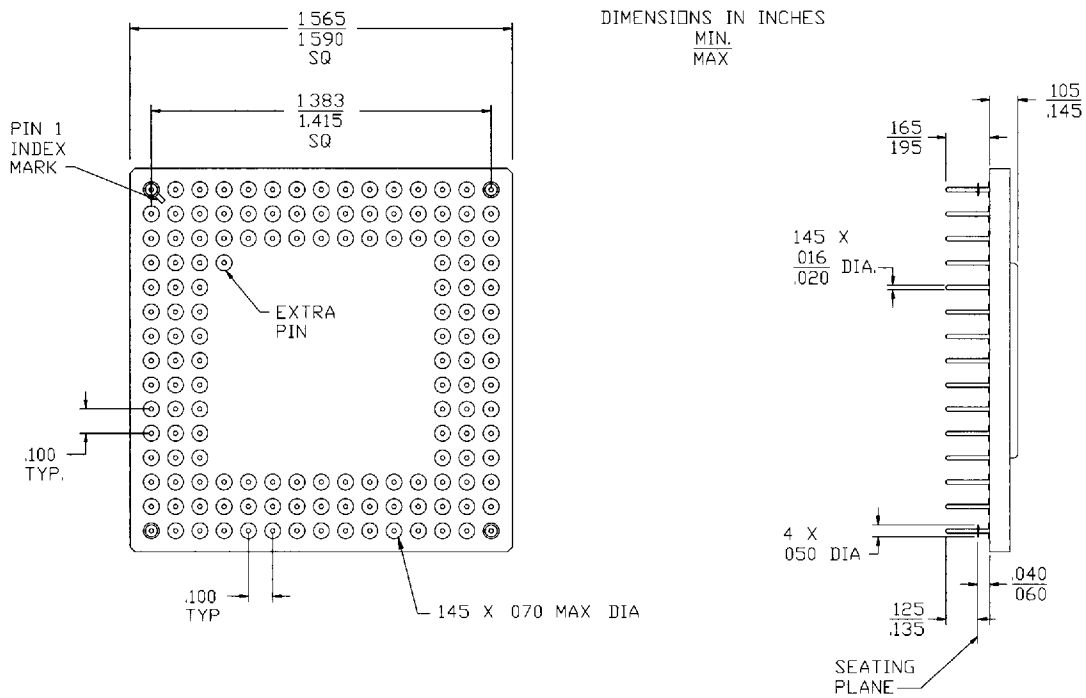
144-Pin Thin Quad Flat Pack A144



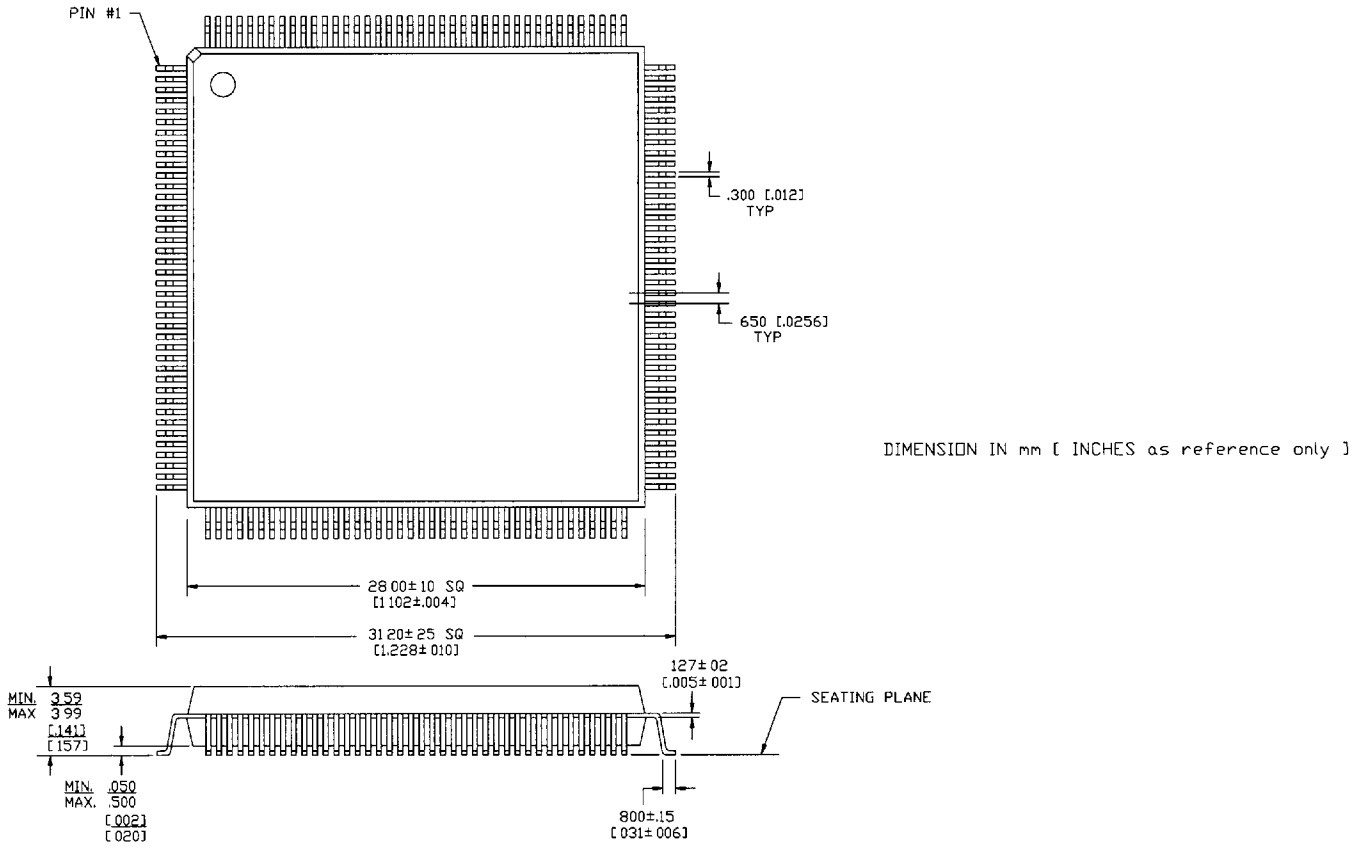
145-Plastic Pin Grid Array (Cavity Up) B144



145-Ceramic Pin Grid Array (Cavity Up) G145



160-Lead Plastic Quad Flatpack N160



160-Lead Ceramic Quad Flatpack U162