2.4
VIC068A: Additional Information

The following sections are related to Section 1, The VIC068A VMEbus Interface Controller. The chapter numbers are those chapters of Section 1 that require clarification or modification for the VIC64. All other information in Section 1 is applicable to the VIC64.

2.4.1 VIC64 Signal Description (Chapter 1.2)

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

SCON*/D64

Input: Yes
Output: Yes
Drive: 16 mA

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 RESET* is asserted, this pin is the SCON* input whose state is latched internally when RESET* goes inactive. A Low state causes VIC64 to become the VMEbus system controller. During the time that RESET* is inactive, this pin becomes the D64 output whose state is normally Low, becoming High only during the Data Phase of D64 transactions.

2.4.2 System Controller Operations (Chapter 1.4)

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 RESET* is asserted (Low), VIC64 assumes that the pin is an input whose state is latched on the rising edge of RESET*. 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 RESET*; the state of the output is used to enable 64-bit data transfers (Chapter 1.10, Block Transfer Functions contains information on this operation).

2.4.3 VMEbus Master Operations (Chapter 1.5)

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 2−1*).

Table 2−1. 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 Byte(0) is transferred on VMEbus address [A31:A24], and Byte(7) 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.

2.4.3.1 D64 Master Write Cycles

In the case of master write cycles, the first local cycle fetches the first [Byte(0)−Byte(3)] 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.

2.4.3.2 D64 Master Read Cycles

In the case of master read cycles, 64 bits of VMEbus data are latched under the control of VIC64. The [Byte(4)−Byte(7)] longword is placed into a two-stage pipe, while the [Byte(0)−Byte(3)] 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 as the first stage of the pipe is now free. See the timing diagrams for full details of this operation.

2.4.4 VMEbus Slave Operations (Chapter 1.6)

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 2−2*).

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

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

2.4.4.1 D64 Slave Read Cycles

As in the case of master write cycles, the first local cycle fetches the first [Byte(0)−Byte(3)] 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.

2.4.4.2 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 [Byte(4)−Byte(7)] longword is placed into a two-stage pipe, while the [Byte(0)−Byte(3)] 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 as the first stage of the pipe is now free.

2.4.5 Interrupts (Chapter 1.9)

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 2−3. VIC64 Interrupt Acknowledge Cycle Selection

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

2.4.6 VIC64 Block Transfer Functions (Chapter 1.10)

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.

2.4.6.1 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.

2.4.7 Miscellaneous Features (Chapter 1.11)

2.4.7.1 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.

2.4.7.2 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.

2.4.8 Register Map and Descriptions (Chapter 1.12)

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

2.4.8.1 Interprocessor Communications Register 5

Name: ICR5

Address: $77

Description: This register provides the VIC64 revision number.

2.4.8.2 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-Kbyte boundary crossing for D64 Master Operations. If this bit is set, VIC64 assumes that the transfer is aligned to a 2-Kbyte boundary.

2.4.8.3 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.

2.4.8.4 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.

2.4.9 AC Performance Specifications (Chapter 1.13)

AC Timing for D64 Operations[2]

Operation		Notes	Commercial		Industrial		Military	
			Min.	Max.	Min.	Max.	Min.	Max.
Master D64 Block Transfer with Local DMA (Initiation Cycle)[3]								
A1	MWB*[0] & PAS*[0] & DS*(0) to BRi*[L]		T+7	2T+32	T+7	2T+33	4T+5	5T+38
A2	MWB*[0] & PAS*[0] & DS*(0) to LADO[H]		T+9	2T+31	T+8	2T+32	T+8	2T+38
A3	MWB*[0] & PAS*[0] & DS*(0) to BLT*[L]		T+9	2T+26	T+8	2T+27	T+8	2T+30
A4	MWB*[0] & PAS*[0] & DS*(0) to DSACK1*[L]		T+11	2T+46	T+10	2T+48	T+10	2T+54
A5	MWB*[0] & PAS*[0] & DS*(0) to DSACK0*[L]		T+11	2T+46	T+10	2T+48	T+10	2T+54
Master D64 Block Transfer Address Broadcast Cycle[3]								
B1	DTACK*[0] to LBR*[L]		24	65	20	69	20	75
B2	DTACK*[0] to DSi*[H]		8	24	7	26	7	30
B3	DTACK*[0] to SCON*/D64[H]		16	59	15	62	15	70
Master D64 Block Transfer with Local DMA (Write)								
** First Longword Fetch **								
C1	DSACKi*[0] and DS*[L] to DS*[H]	3, 4, 5	MBAT0+8	MBAT0+T +36	MBAT0+7	MBAT0+T +38	MBAT0+7	MBAT0+T +42
C2	DSACKi*[0] and DS*[L] to LEDO[H]	3, 4, 5	MBAT0+7	MBAT0+T +33	MBAT0+6	MBAT0+T +35	MBAT0+6	MBAT0+T +36
C3	DSACKi*[0] and DS*[L] to LA(7:0)	3, 4, 5	MBAT0+ .5T+9	MBAT0+2T +30	MBAT0+ .5T+8	MBAT0+2T +32	MBAT0+ .5T+8	MBAT0+2T +35
C4	DS*(H) to DS*[L]	3, 4, 5, 6	T+8	3T+31	T+7	3T+32	T+7	3T+35
** Second Longword Fetch **								
C5	DSACKi*[0] and DS*[L] to DS*[H]	3, 5	MBAT1+14	MBAT1+T +46	MBAT1+13	MBAT1+T +48	MBAT1+13	MBAT1+T +52
C6	DSACKi*[0] and DS*[L] to DENO*[L]	5	MBAT1+11	MBAT1+T +37	MBAT1+10	MBAT1+T +39	MBAT1+10	MBAT1+T +42
C7	DSACKi*[0] and DS*[L] to LA(7:0)	3, 5	MBAT1+ .5T+9	MBAT1+2T +31	MBAT1+ .5T+8	MBAT1+2T +32	MBAT1+ .5T+8	MBAT1+2T +35
C8	DSACKi*[0] and DS*[L] to DSi*[L]	7	MBAT1+ 3T+12	MBAT1+4T +32	MBAT1+ 3T+10	MBAT1+4T +33	MBAT1+ 3T+10	MBAT1+4T +36
C9	DSACKi*[0] and DS*[L] to LEDO[L]	5	MBAT1+16	MBAT1+T +56	MBAT1+15	MBAT1+T +57	MBAT1+15	MBAT1+T +64
C10	DS*(H) to DS*[L]	3, 5, 6	T+8	3T+31	T+7	3T+32	T+7	3T+35
C11	DTACK*[0] to DSi*[H]		7	22	6	23	6	25
C12	DTACK*[0] to DENO*[H]	3	8	24	7	26	7	30

Operation		Notes	Commercial		Industrial		Military	
			Min.	Max.	Min.	Max.	Min.	Max.
Master Block Transfer with Local DMA (Read)								
** First Longword Write **								
D1	LBG*[0] to DENIN*[L]	3	2T+11	3T+41	2T+10	3T+43	2T+10	3T+48
D2	DTACK*[0] to LEDI[H]	8	2T+6	3T+23	2T+5	3T+24	2T+5	3T+27
D3	DTACK*[0] to DSi*[H]	3, 7	2T+9	3T+28	2T+8	3T+29	2T+8	3T+32
D4	DTACK*[0] to DS*[L]	7	2T+13	3T+36	2T+12	3T+37	2T+12	3T+41
D5	DSACKi*[0] and DS*[L] to DS*[H]	3, 4, 5	MBAT0+8	MBAT0+T +37	MBAT0+7	MBAT0+T +38	MBAT0+7	MBAT0+T +42
D6	DSACKi*[0] and DS*[L] to LEDI[L]	4, 5	MBAT0+13	MBAT0+T +52	MBAT0+12	MBAT0+T +53	MBAT0+12	MBAT0+T +60
D7	DSACKi*[0] and DS*[L] to DENIN1*[L]	4, 5	MBAT0+8	MBAT0+T +35	MBAT0+7	MBAT0+T +36	MBAT0+7	MBAT0+T +41
D8	DSACKi*[0] and DS*[L] to LA(7:0)	3, 4, 5	MBAT0+ .5T+9	MBAT0+2T +29	MBAT0+ .5T+8	MBAT0+2T +31	MBAT0+ .5T+8	MBAT0+2T +35
D9	DSACKi*[0] and DS*[L] to DSi*[L]	4, 5	MBAT0+22	MBAT0+T +56	MBAT0+20	MBAT0+T +58	MBAT0+20	MBAT0+T +64
D10	DS*[H] to DS*[L]	3, 5, 6	T+8	3T+29	T+7	3T+31	T+7	3T+35
** Second Longword Write **								
D12	DSACKi*[0] and DS*[L] to DS*[H]	3, 5	MBAT1+8	MBAT1+T +36	MBAT1+7	MBAT1+T +38	MBAT1+7	MBAT1+T +42
D13	DSACKi*[0] and DS*[L] to DENIN1*[H]	3, 5	MBAT1+16	MBAT1+T +56	MBAT1+15	MBAT1+T +59	MBAT1+15	MBAT1+T +66
D14	DSACKi*[0] and DS*[L] to LA(7:0)	3, 5	MBAT1+ .5T+9	MBAT1+2T +29	MBAT1+ .5T+8	MBAT1+2T +31	MBAT1+ .5T+8	MBAT1+2T +35
D15	DSACKi*[0] and DS*[L] to LD(7:0)	3, 5	MBAT1+ .5T+12	MBAT1+2T +39	MBAT1+ .5T+10	MBAT1+2T +42	MBAT1+ .5T+10	MBAT1+2T +48
Master D64 Block Transfer with Local DMA (Boundary Crossing)[3]								
E1	DS*[0] to BLT*[H]		3	28	2	30	2	33
E2	DS*[1] to BLT*[L]		3	19	2	20	2	21
E3	DSi*[0] to LADO first transition		3	19	3	20	2	21
E4	DSi*[0] to LADO second transition		3	19	3	20	2	21
Slave D64 Block Transfer Address Broadcast Cycle[3]								
F1	DSi*[1] to LBR*[L]		11	36	10	39	10	42
F2	DSi*[0] to DTACK*[L]		2T+9	3T+28	2T+8	3T+29	2T+8	3T+32
F3	DSi*[1] to DTACK*[H]		9	28	8	35	8	39
F4	DSi*[1] to SCON*/D64[H]		10	33	9	35	9	39
F5	DSi*[0] and AS*[0] and SLSELi*[0] to LADI[H]		1.5T+5	2T+25	1.5T+4	2T+26	1.5T+4	2T+29

Operation		Notes	Commercial		Industrial		Military	
			Min.	Max.	Min.	Max.	Min.	Max.
Slave D64 Block Transfer (Write)								
** First Longword Cycle **								
G1	DSi*[0] to DS*[L]	3, 5	3T+11	4T+38	3T+10	4T+40	3T+10	4T+44
G2	DSi*[0] to DTACK*[L]	7	2T+9	3T+23	2T+8	3T+24	2T+8	3T+26
G3	DSi*[0] to LEDI[H]	3, 5	T+11	2T+37	T+10	2T+39	T+10	2T+42
G4	DSACKi*[0] and DS*[L] to DS*[H]	3, 5, 9	SBAT0+8	SBAT0+T+36	SBAT0+7	SBAT0+T+38	SBAT0+7	SBAT0+T+42
G5	DSACKi*[0] and DS*[L] to LEDI[L]	3, 5, 9	SBAT0+13	SBAT0+T+52	SBAT0+12	SBAT0+T+54	SBAT0+12	SBAT0+T+60
G6	DSACKi*[0] and DS*[L] to DENIN1*[L]	3, 5, 9	SBAT0+8	SBAT0+T+31	SBAT0+7	SBAT0+T+33	SBAT0+7	SBAT0+T+37
G7	DSACKi*[0] and DS*[L] to LA(7:0)	3, 5, 9	SBAT0+.5T+9	SBAT0+2T+29	SBAT0+.5T+8	SBAT0+2T+31	SBAT0+.5T+8	SBAT0+2T+35
G8	DS*[H] to DS*[L]	3, 5, 6	T+8	3T+31	T+7	3T+32	T+7	3T+35
** Second Longword Cycle[3, 5] **								
G9	DSACKi*[0] and DS*[L] to DENIN1*[H]		SBAT1+20	SBAT1+T+64	SBAT1+19	SBAT1+T+67	SBAT1+19	SBAT1+T+74
G10	DSACKi*[0] and DS*[L] to DS*[H]		SBAT1+11	SBAT1+T+34	SBAT1+10	SBAT1+T+36	SBAT1+10	SBAT1+T+42
G11	DSACKi*[0] and DS*[L] to LA(7:0)		SBAT1+.5T+9	SBAT1+2T+29	SBAT1+.5T+8	SBAT1+2T+31	SBAT1+.5T+8	SBAT1+2T+35
G12	DSACKi*[0] and DS*[L] to LD(7:0)		SBAT1+.5T+11	SBAT1+2T+40	SBAT1+.5T+10	SBAT1+2T+42	SBAT1+.5T+10	SBAT1+2T+48
Slave D64 Block Transfer (Read)								
** First Longword Cycle **								
H1	DSACKi*[0] and DS*[L] to LEDO[H]	5, 9	SBAT0+7	SBAT0+T+36	SBAT0+6	SBAT0+T+37	SBAT0+6	SBAT0+T+41
H2	DSACKi*[0] and DS*[L] to DS*[H]	5, 9	SBAT0+8	SBAT0+T+39	SBAT0+7	SBAT0+T+41	SBAT0+7	SBAT0+T+45
H3	DSACKi*[0] and DS*[L] to LA(7:0)	3, 5, 9	SBAT0+.5T+9	SBAT0+T+29	SBAT0+.5T+8	SBAT0+T+31	SBAT0+.5T+8	SBAT0+T+35
H4	DS*[H] to DS*[L]	3, 5, 6	T+8	3T+30	T+7	3T+32	T+7	3T+35
** Second Longword Cycle **								
H5	DSACKi*[0] and DS*[L] to LEDO[L]	5	SBAT1+19	SBAT1+T+64	SBAT1+18	SBAT1+T+67	SBAT1+18	SBAT1+T+74
H6	DSACKi*[0] and DS*[L] to DENO*[L]	5	SBAT1+11	SBAT1+T+37	SBAT1+10	SBAT1+T+39	SBAT1+10	SBAT1+T+42
H7	DSACKi*[0] and DS*[L] to DS*[H]	3, 5	SBAT1+24	SBAT1+T+72	SBAT1+23	SBAT1+T+75	SBAT1+23	SBAT1+T+85
H8	DSACKi*[0] and DS*[L] and DSi*[0] to DTACK*[L]	5	SBAT1+13	SBAT1+T+33	SBAT1+12	SBAT1+T+34	SBAT1+12	SBAT1+T+38
H9	DSACKi*[0] and DS*[L] to LA(7:0)	3, 5	SBAT1+.5T+9	SBAT1+2T+29	SBAT1+.5T+8	SBAT1+2T+31	SBAT1+.5T+8	SBAT1+2T+35
H10	DS1/0*[1] to DENO*[H]	3	6	21	5	22	5	25

Operation		Notes	Commercial		Industrial		Military	
			Min.	Max.	Min.	Max.	Min.	Max.
Slave D64 Block Transfer (Boundary Crossing)								
H11	DS*[0] to LADI[L]	3	11	26	10	28	10	32
H12	DS*[1] to LADI[H]	3	6	13	5	14	5	15

Notes:

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

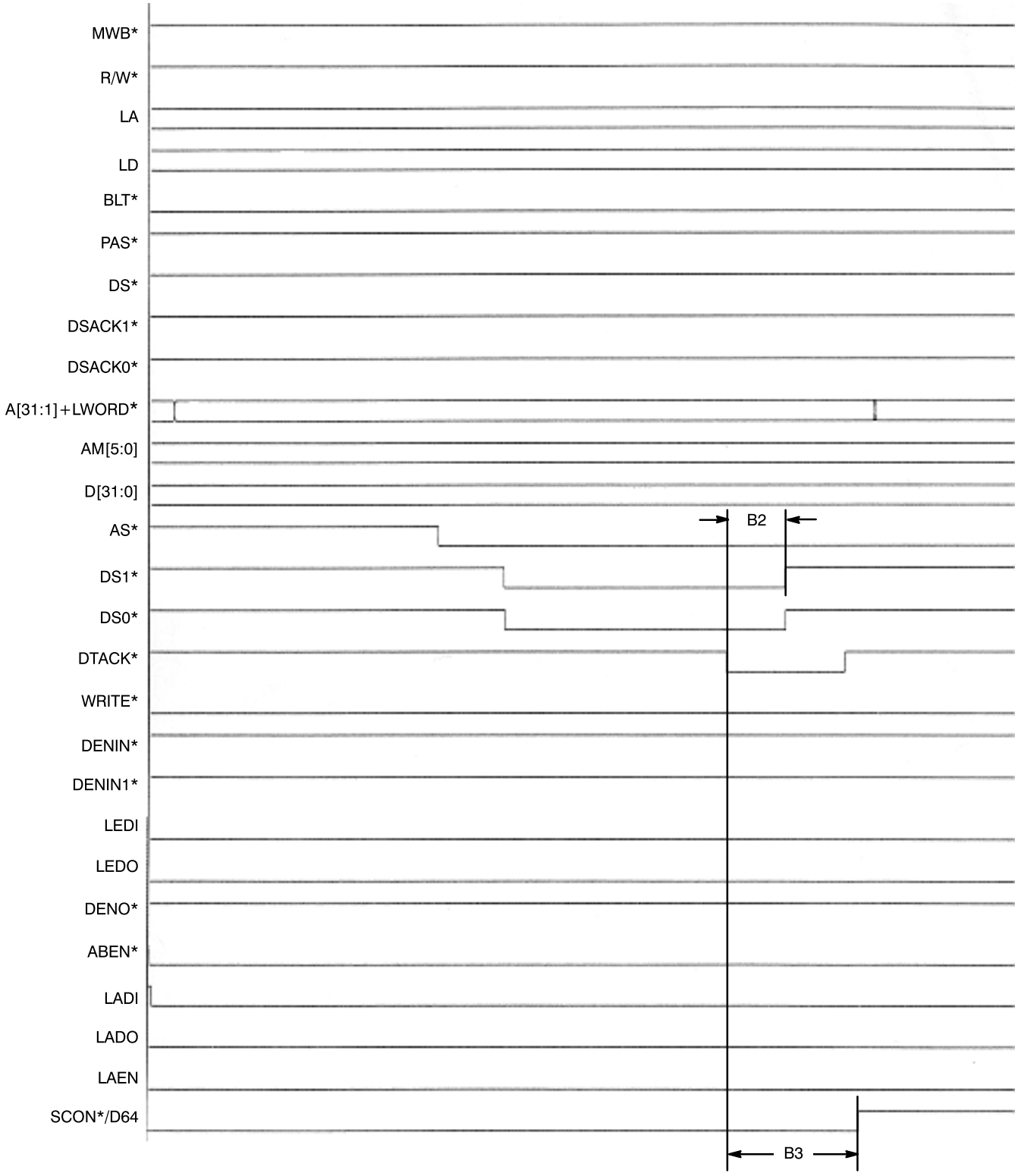

Figure 2−1. Master Address Broadcast Cycle

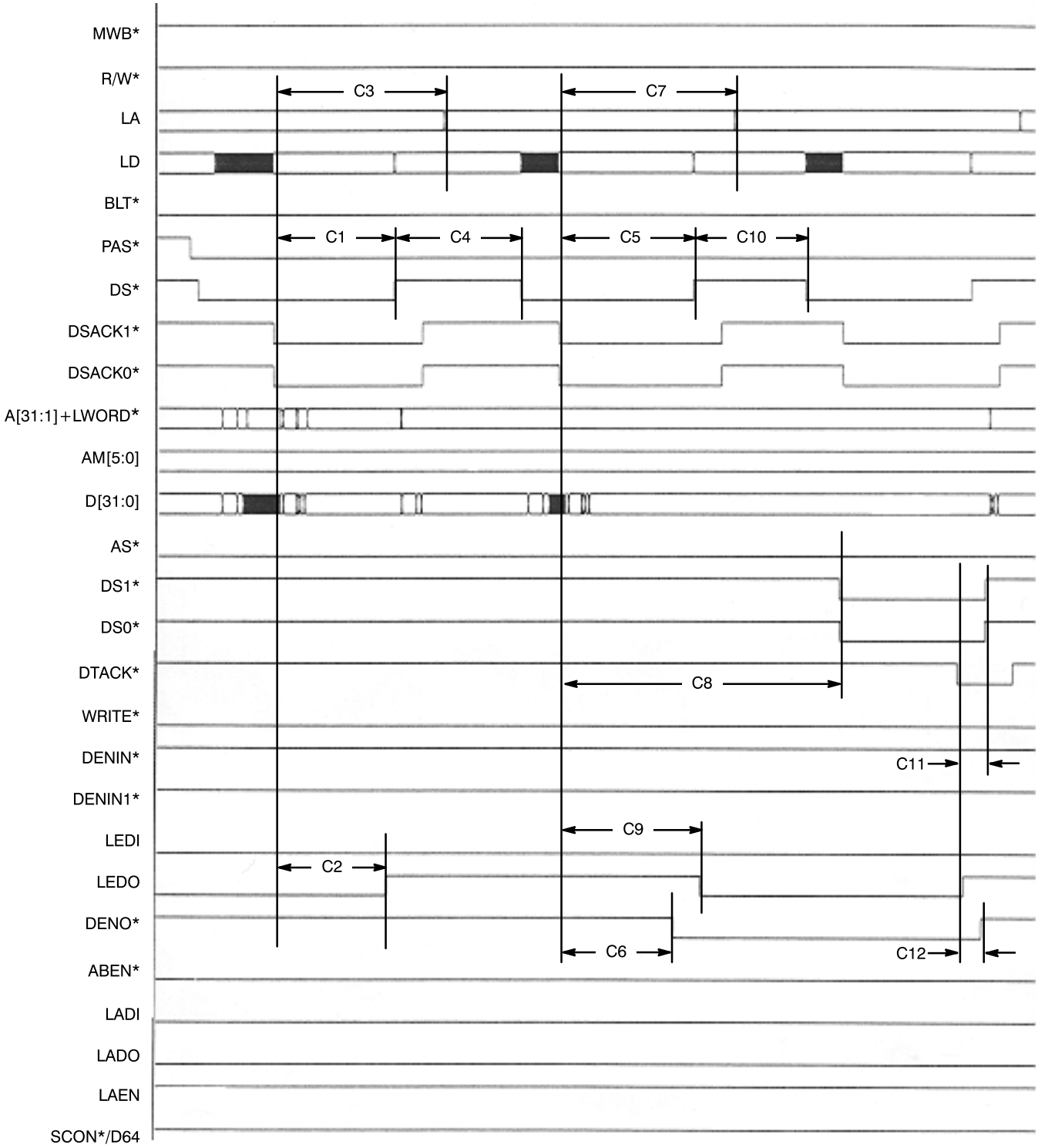

Figure 2−2. Master D64 Write Operation: Detail

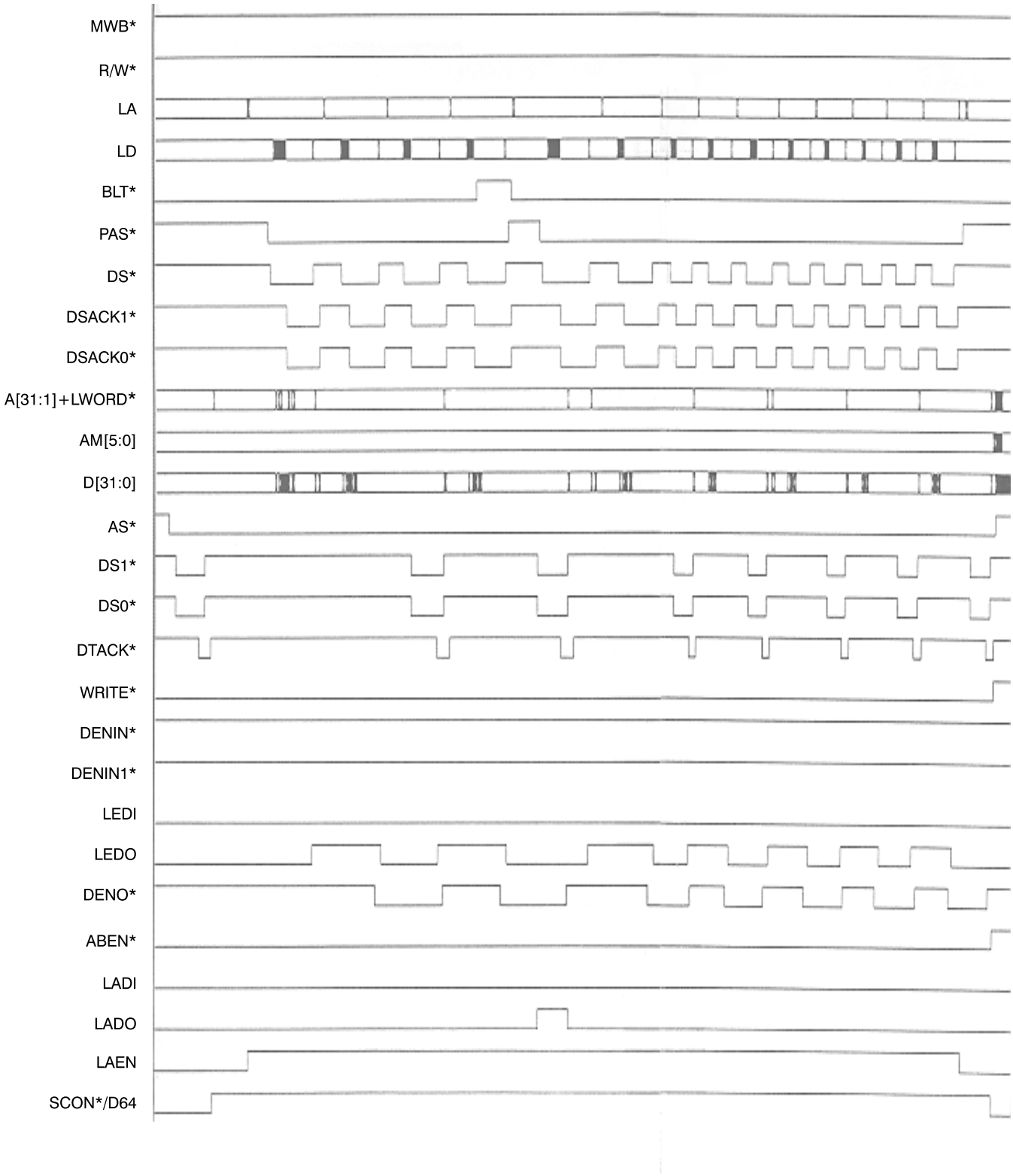

Figure 2−3. Master D64 Write Operation: Block Transfer

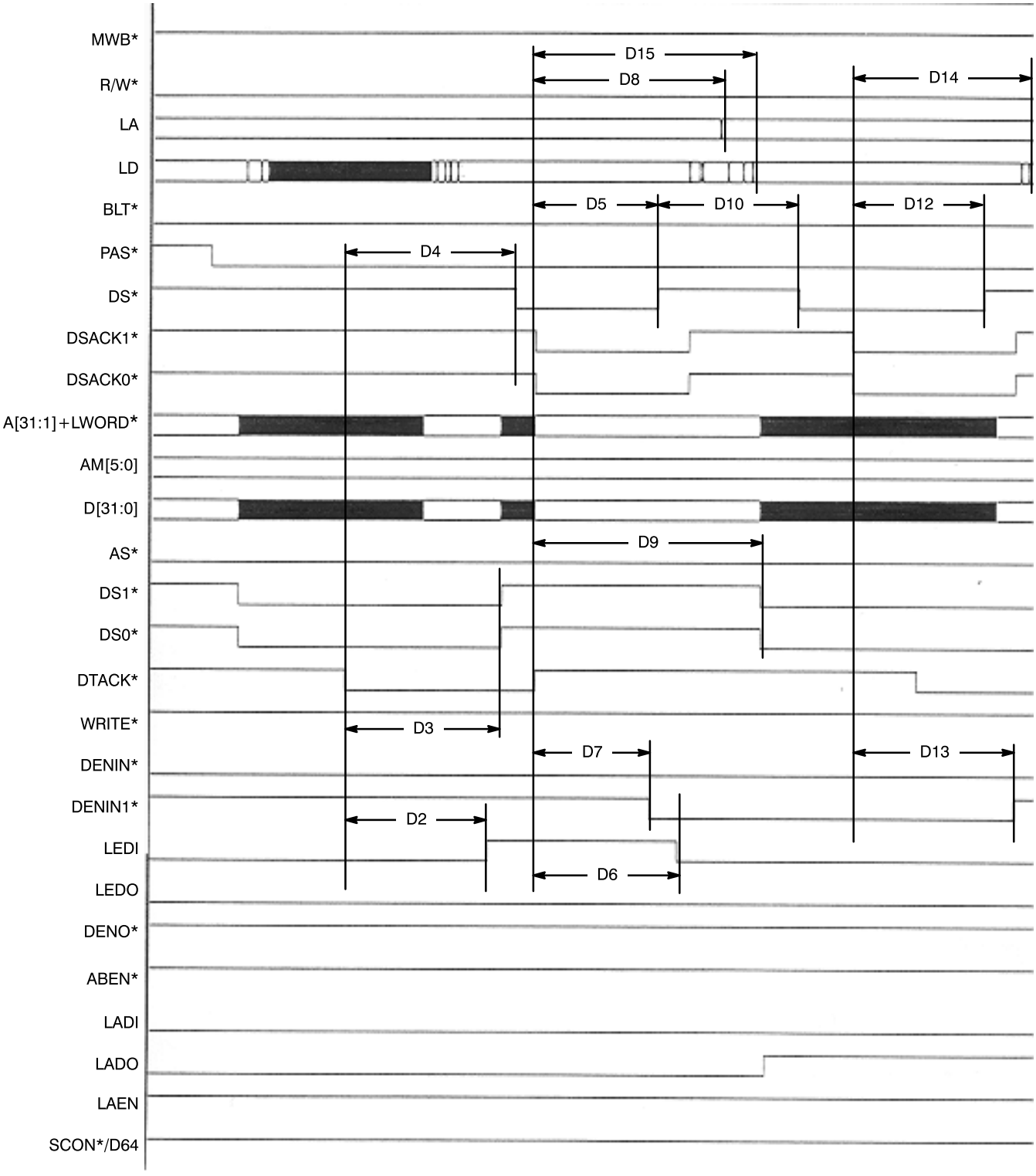

Figure 2−4. Master D64 Read Operation: Detail

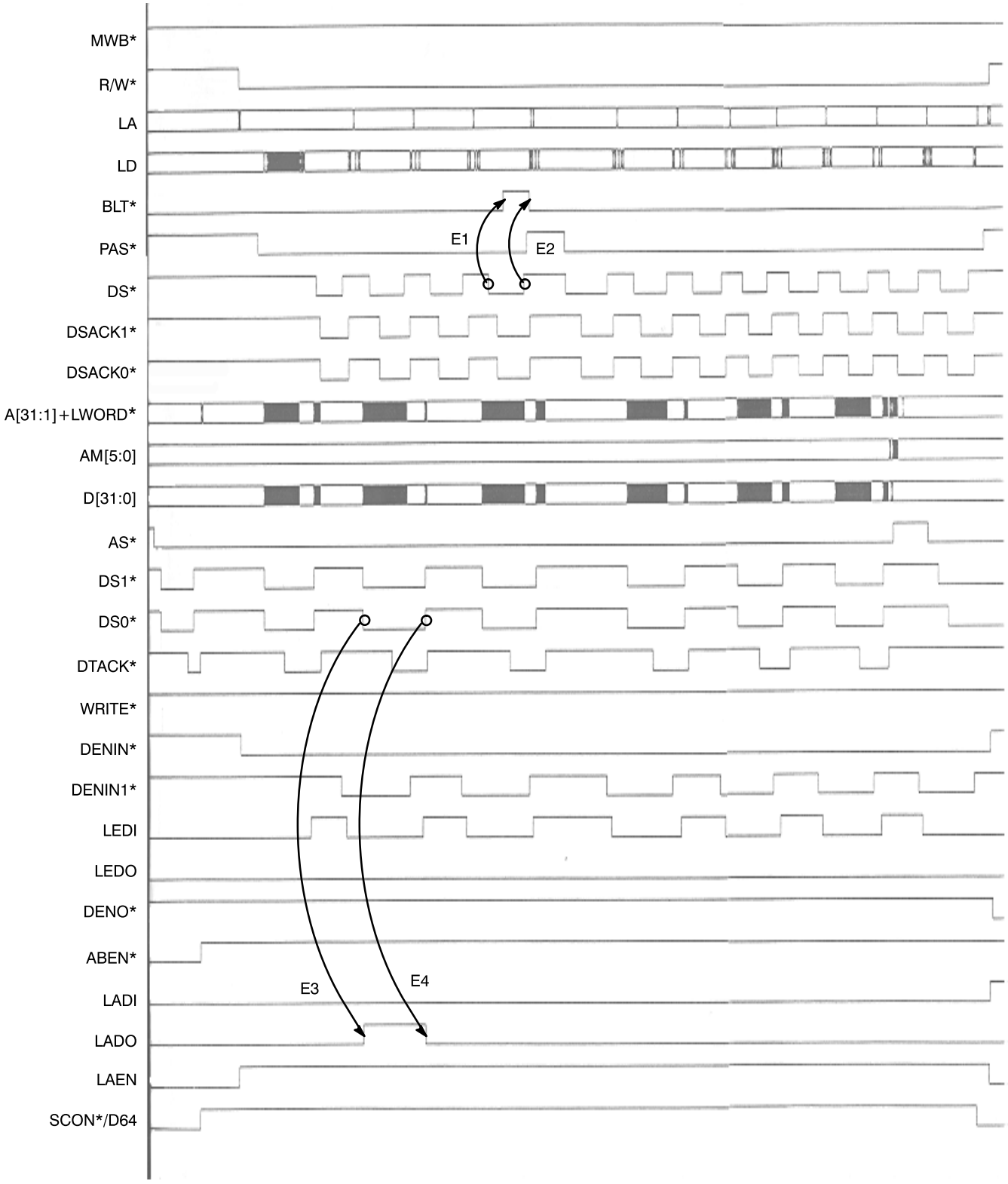

Figure 2−5. Master D64 Read Operation: Block Transfer

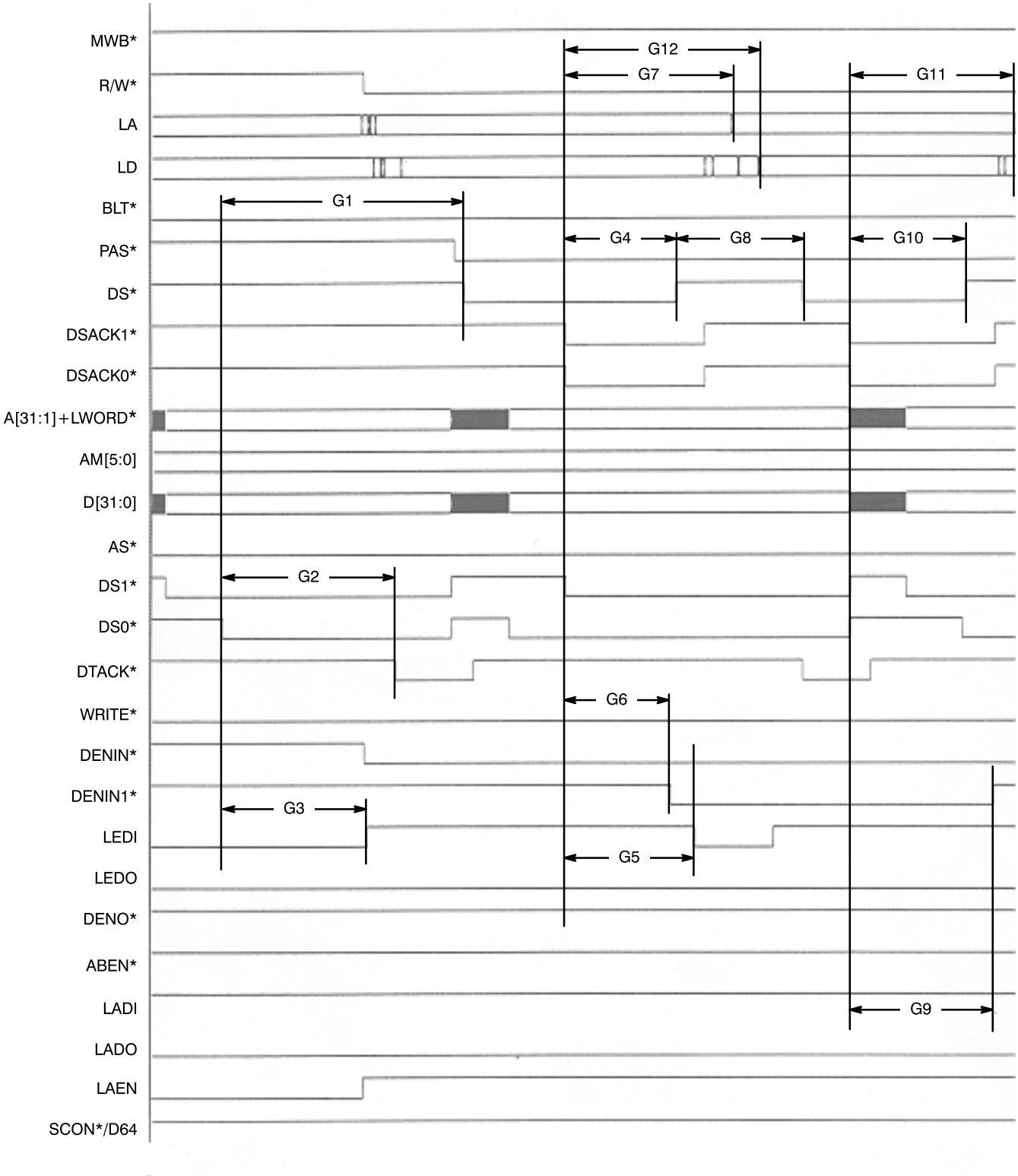

Figure 2−6. Slave D64 Write Operation: Detail

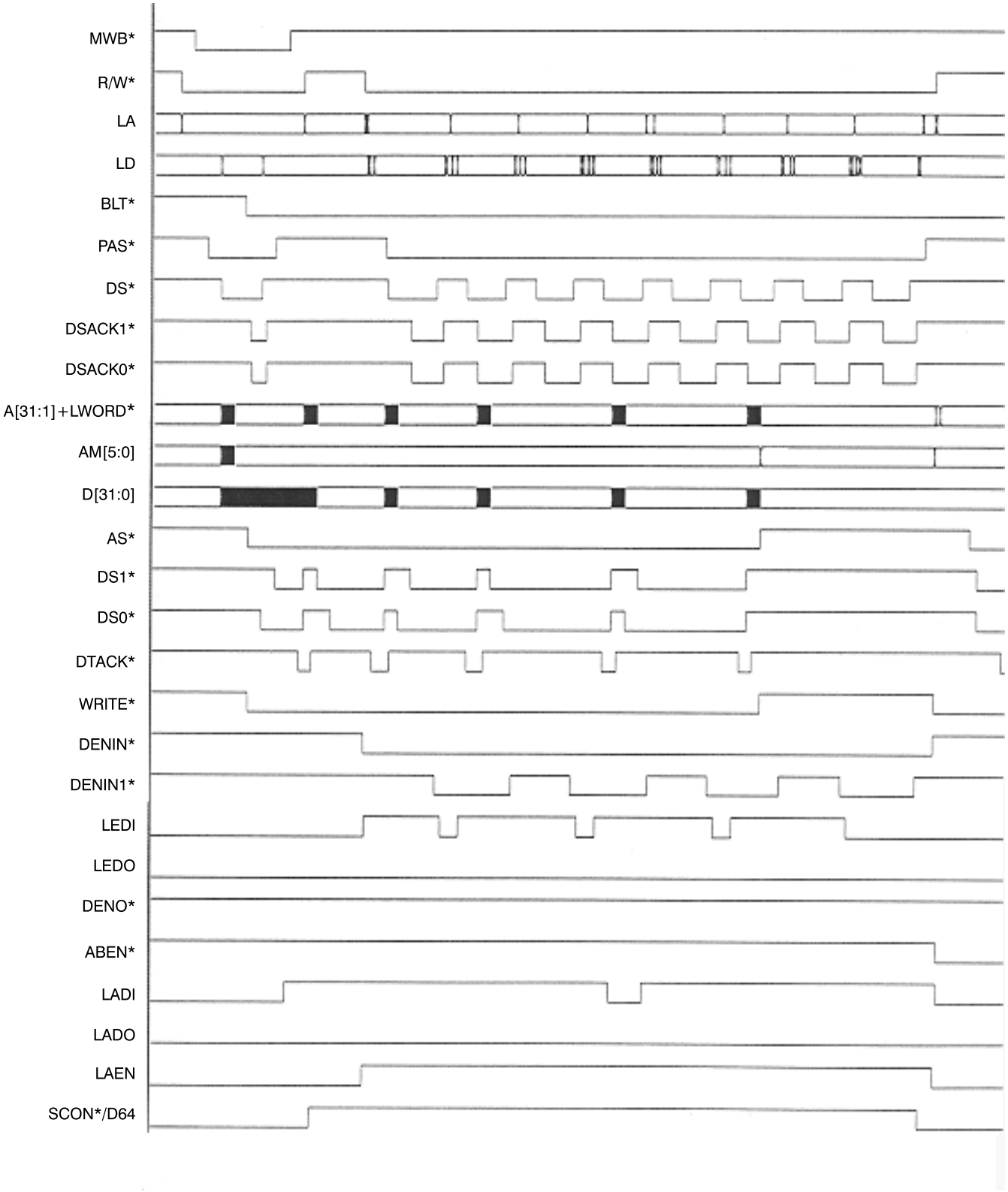

Figure 2−7. Slave D64 Write Operation: Block Transfer

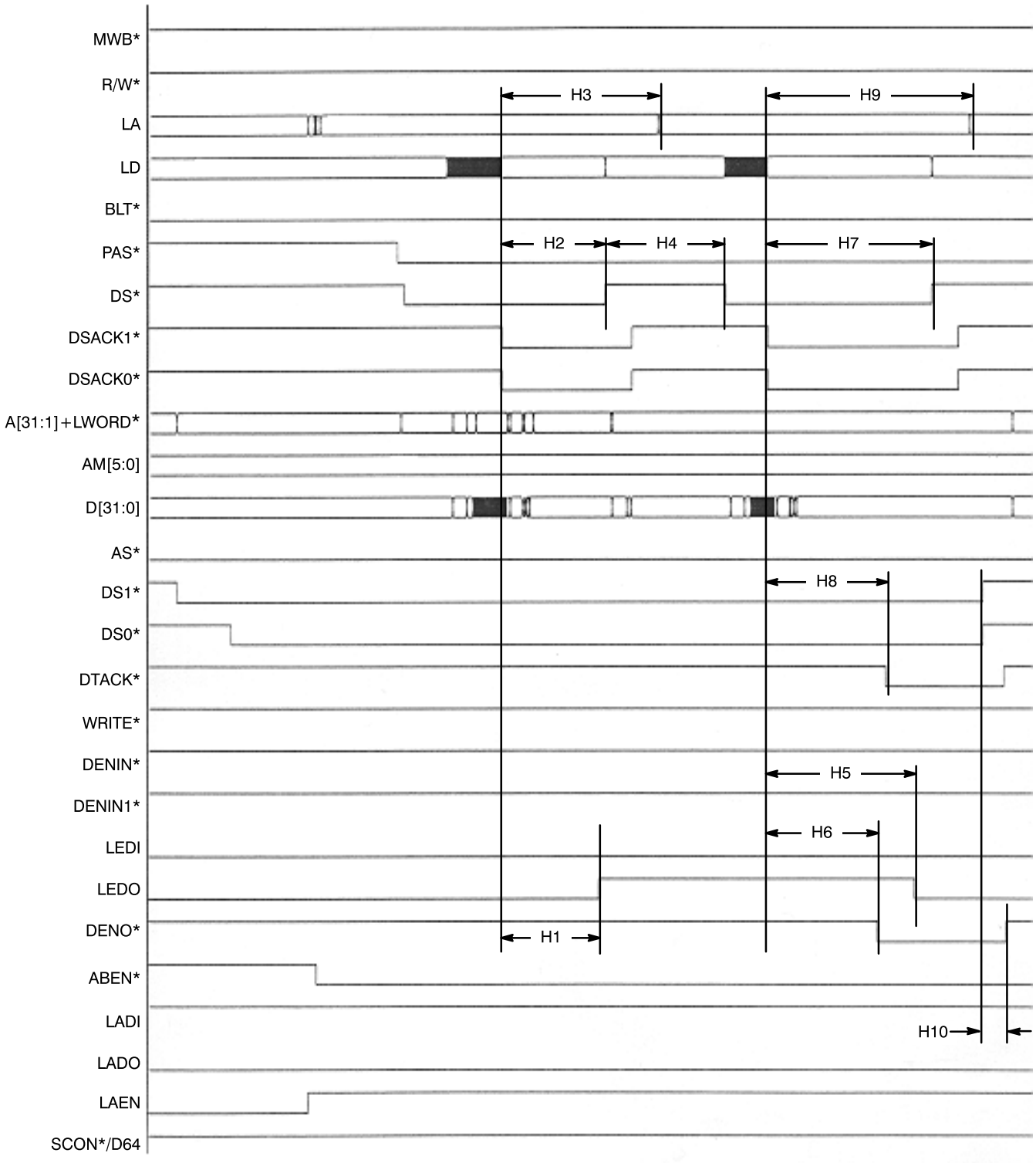

Figure 2−8. Slave D64 Read Operation: Detail

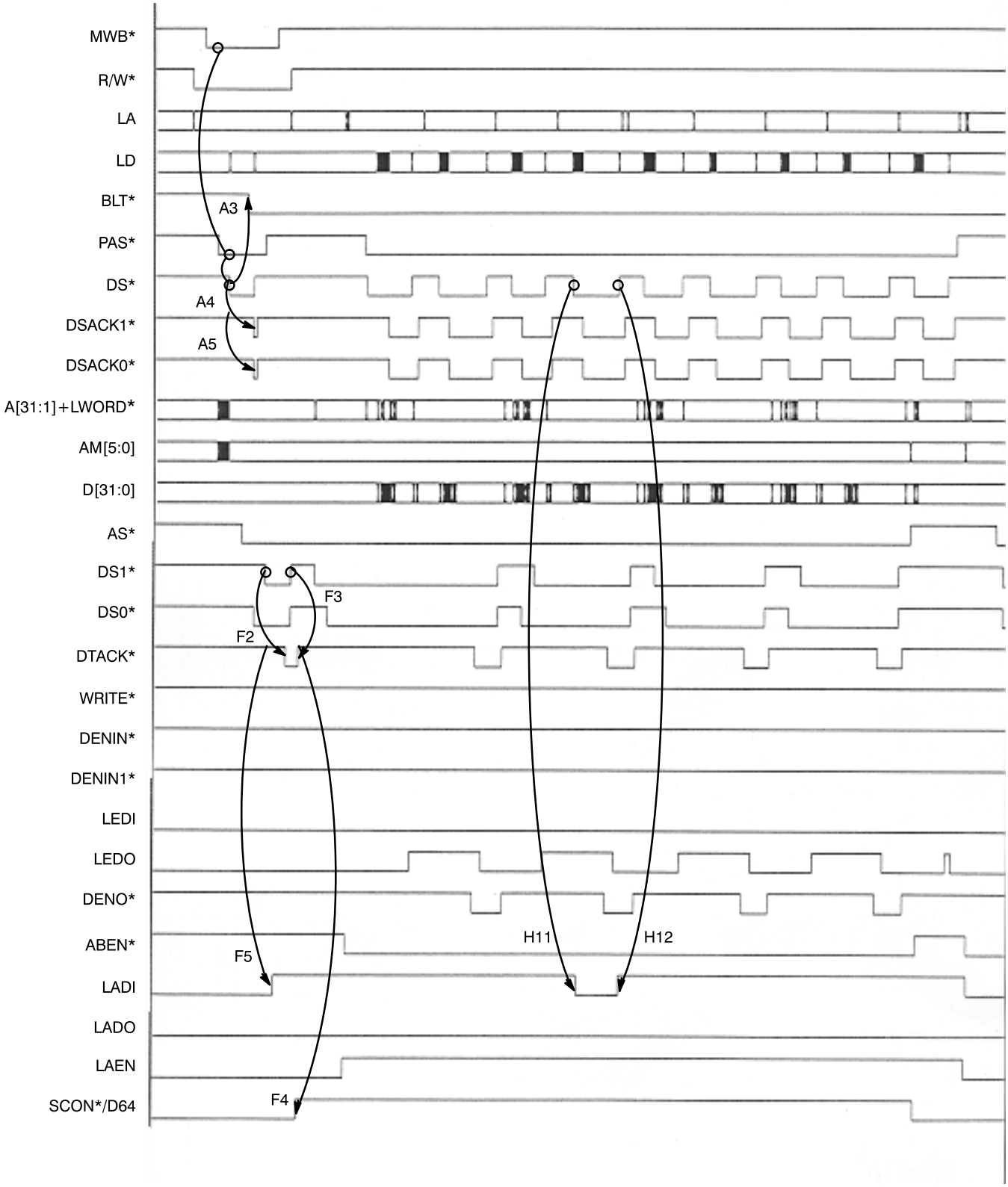

Figure 2−9. Slave D64 Read Operation: Block Transfer