3.11
CY7C961 Description

3.11.1 Introduction

The CY7C961 is a CY7C960 Slave VMEbus Interface Controller with the addition of a master block transfer capability. Full-featured Slave boards can be built, using the CY7C961, that offer a flexible Master block transfer facility for bursting data across the VMEbus. The CY7C961 can receive its instructions from a VMEbus Master or by programming registers locally. The CY7C961 interprets the instructions and then moves data accordingly as a VMEbus Master. This optimizes performance and bus utilization.

The CY7C961 is a true superset of the CY7C960. Signal pins have been added to control CY7C964 DMA functions. Unidirectional VMEbus pins have been changed to bidirectional. A few additional signals have been provided to complete a master interface, such as a data port and VMEbus requester signals. As a VMEbus Slave, the CY7C961 behaves in every respect like the CY7C960. It has more pins, a master block transfer facility, and (because of the addition of BBSY*) full lock cycle support.

From a system perspective, this CY7C961 master block transfer capability can be viewed as a DMA channel which resides on the slave card, and is controlled by a dual ported on-chip register file. It is possible to program the DMA channel from the VMEbus or from the local side of the interface, or both. Once programmed, the CY7C961 acquires the VMEbus and transfers data in one of 20 user-selected protocols.

Circuitry on the local side of the CY7C961 sees the same control signals as were described for the CY7C960. For example, the REGION inputs to the CY7C961 are driven by an external address decoder. The address decoder sits on the local address bus, which is driven from the CY7C964s. The local memory or I/O involved in the data transfer is enabled through the CS outputs and the DRAM control signals from the CY7C961, so they must be configured correctly for the REGION inputs being driven.

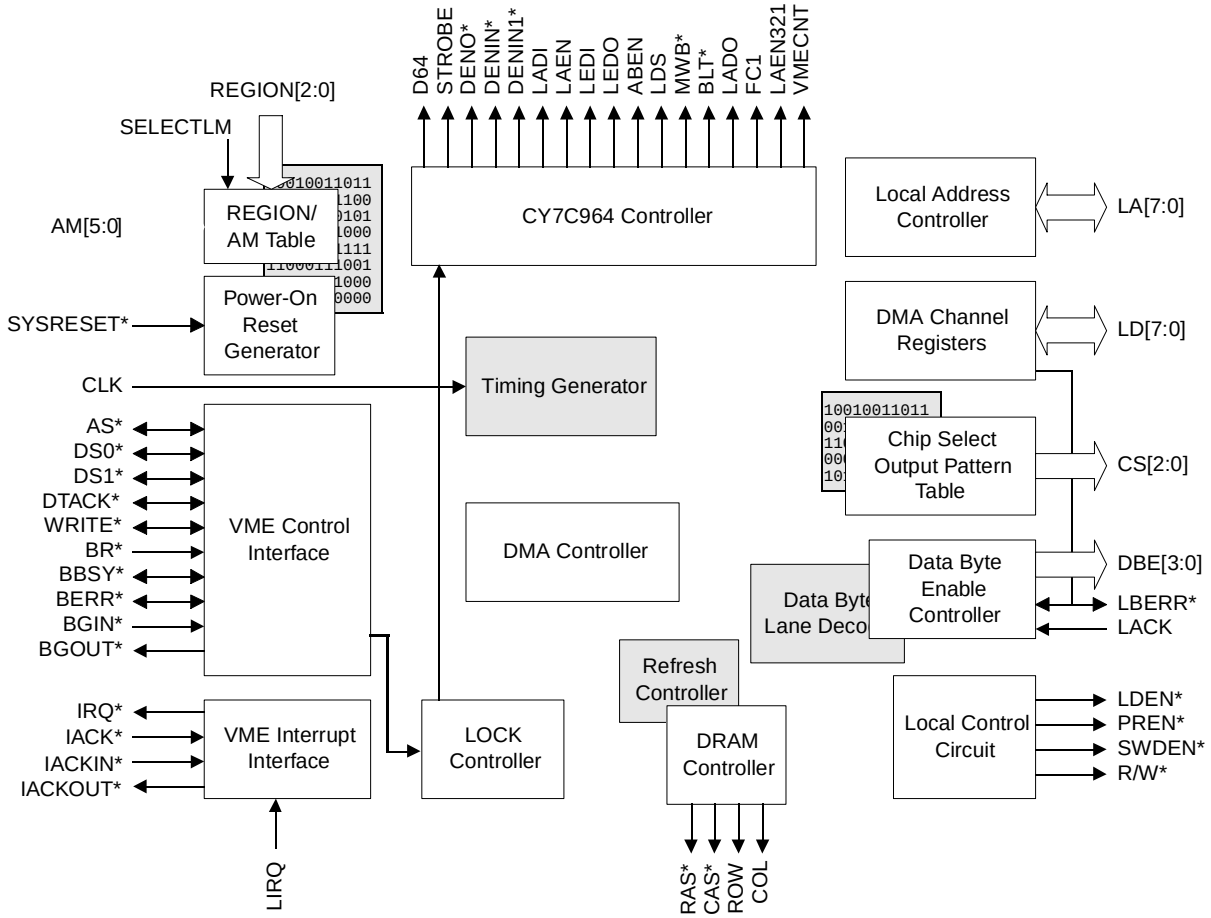


Figure 3-34. CY7C961 Block Diagram

This implies that the VCOMP* outputs from the CY7C964s cannot form part of the address decoder for master operations, because they are looking not at the local address, but at the VMEbus address, which by definition must be pointing somewhere other than the CY7C961's address space.

Figure 3-34 shows the block diagram of the CY7C961. It is very similar to the CY7C960 block diagram (see Introduction), with the addition of some CY7C964 control signals, signals needed to support VMEbus Master Transactions, and other minor changes.

3.11.2 CY7C961 Lock Cycle Support

3.11.2.1 Overview

Lock commands are Address-Only-Cycles-With-Handshake (ADOH) cycles that are used to lock the other port(s) of multiported resources, where one port is on VMEbus. Each Slave's resource that is addressed with a lock command is to lock out all other accesses to that resource. Use of a lock command signifies the start of a locked sequence of VMEbus cycles

which ends with the end of the current VMEbus Master's bus tenure. The CY7C961 allows locked cycles to be decoded, and drives a local lock indicator while the ensuing VMEbus locked sequence is in progress.

3.11.2.2 Description

Each of the lock commands (A16, A24, A32, A40, A64) consists of an address phase which is presented to the VMEbus and handshaken by the targeted Slave. The CY7C961 can be programmed to decode lock commands on any ONE of its 16 decode regions. Any combination of the five lock commands can be enabled for that region, but only one lock indicator signal is provided. When a lock command is decoded, the CY7C961 drives its MWB pin Low to indicate the beginning of the locked VMEbus sequence. The lock indication on MWB will be maintained as long as BBSY* or AS* remains asserted to the CY7C961. A decoded lock cycle will not cause chip selects or byte enables to become active. A decoded lock cycle will not interfere in any way with slave accesses to the CY7C961. *Figure 3-35* shows the timing of the various signals associated with LOCK cycles.

The MWB signal is active during block transfer register accesses and during CY7C961 master block transfers. MWB indicates a default lock condition during master block transfer. This should be considered when designing the local resource lock control circuitry in which MWB is used. If the lock default is not needed during master block transfers, the CY7C961 signal FC1 can be used to disable lock during master block transfers.

3.11.3 CY7C961 Master Block Facility

3.11.3.1 Overview

The CY7C961 Master Block Facility provides "block transfer on demand" capability for slave cards built around the Cypress CY7C961/CY7C964 chip set. This facility allows DMA control by writing a short series of commands to the CY7C961/CY7C964 chip set, telling it how much data to move, where to get it from, where to put it, and what transfer protocol to use while moving it. Blocks can be moved over the VMEbus as indivisible single cycles or BLTs. The protocol menu includes D8, D16, D32, MD32, or D64. A16, A24, A32, A40, and A64 address spaces can be specified. Burst lengths from 16 bytes to 8 megabytes can be requested. Eight registers accessible from the VMEbus or the local side of the interface make the facility simple to configure and simple to control. The facility has a busy semaphore, a VMEbus Interrupt on completion feature with a programmable statusID byte, and a built in requester and bus grant daisychain.

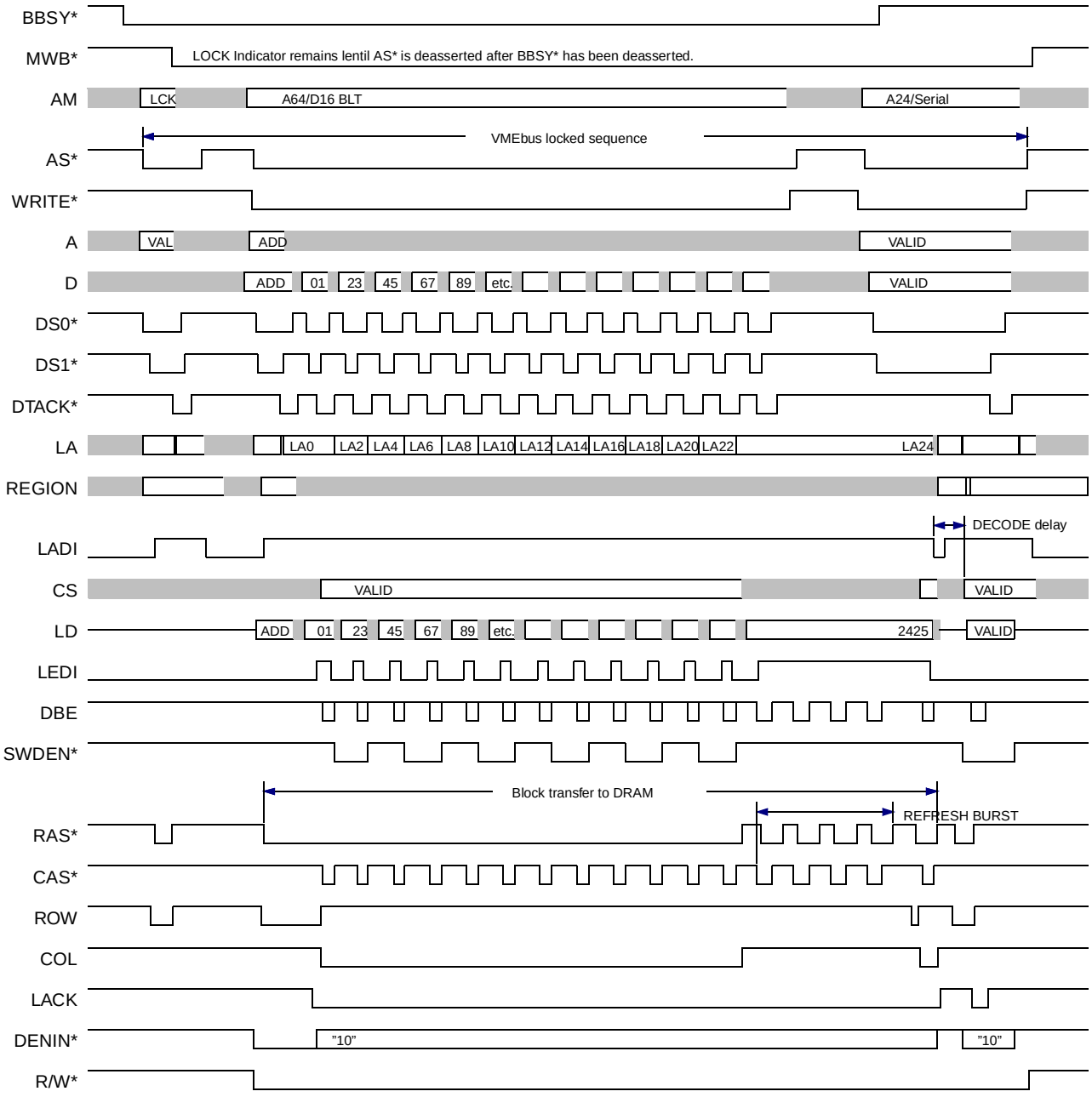


Figure 3-35. LOCK Cycle Timing

It is expected that the system designer will choose either local programming of the DMA channel or VMEbus programming and design decode and other support circuitry based on that choice. There is nothing in the design of the CY7C961 to prevent dual porting of the DMA control registers, but dual porting, like multi-master VMEbus control requires polling of the CY7C961 control register semaphore and/or added complexity in hardware and software design.

3.11.3.2 Master Block Transfer Control from VMEbus

All control of the master block facility is by register access. Eight registers are designed to be accessed over the VMEbus by single cycle VME masters. In order to reach the Master Block facility, the VMEbus address must cause the SELECTLM* signal and a valid REGION vector to be asserted to the CY7C961. Refer to *Figure 3-36* for the register access detail. The assertion of SELECTLM* blocks any local response by the CY7C961. Instead, internal gating and external buffer control signaling appropriate to the register access specified by LA[4:2] is substituted. The CY7C961 asserts DTACK* for valid register accesses or BERR* if an error access is attempted. Eight registers are defined, selected by LA[4:2].

3.11.3.3 Master Block Transfer Control from Local Side of Interface

All control of the master block facility is by register access. Eight registers are designed to be accessed directly through control of LA[4:2], LD[31:0], R/W*, and SELECTLM*. The “local bus holdoff” feature of the CY7C961 must be enabled and the CY7C961 must be in its “holdoff” state before register access is begun. (See section for a complete description of “Local Bus Holdoff.”) In order to reach the Master Block facility, LA[4:2] and R/W are set up to the assertion of the SELECTLM*. For register write cycles, LD[31:0] must also be set up to SELECTLM*.

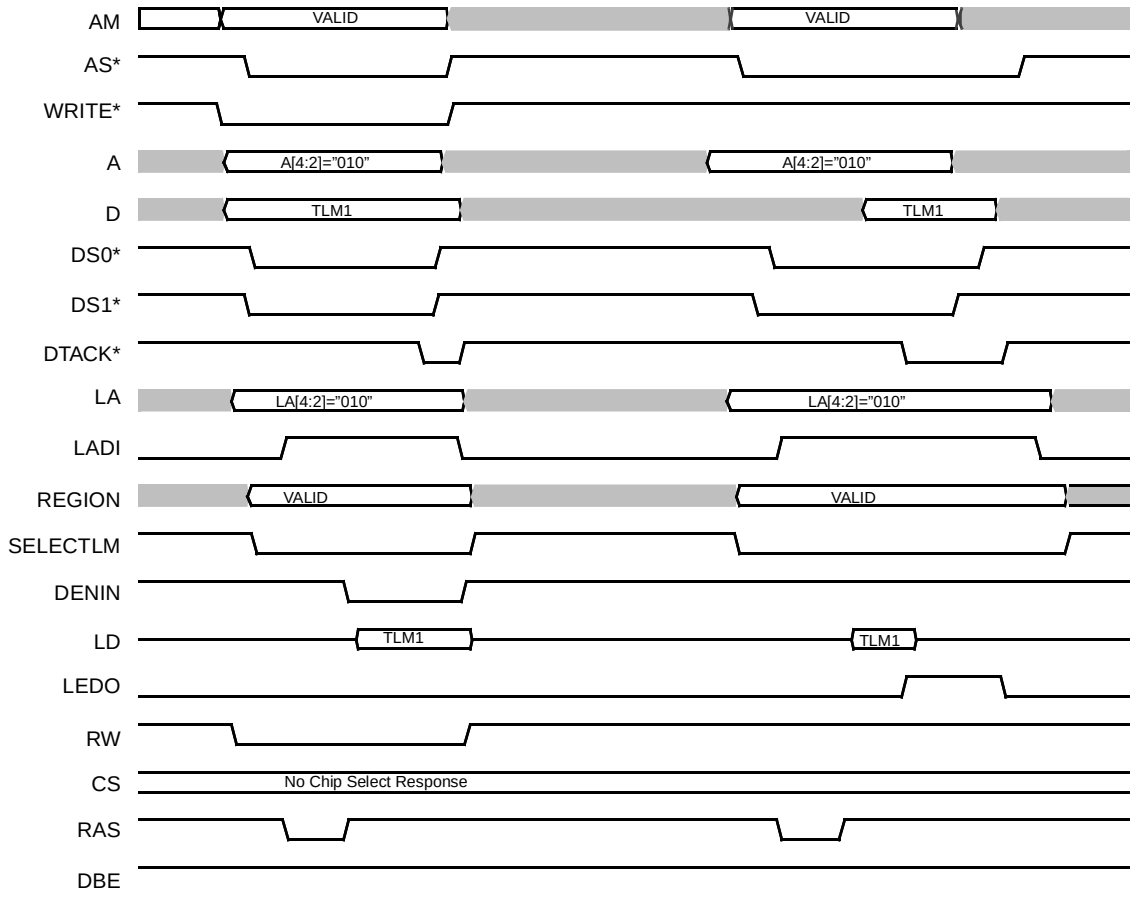


Figure 3-36. DMA Control Register Access from VMEbus

The response of the CY7C961 is self-timed and no acknowledge handshake is provided. SELECTLM* must be asserted for a minimum of 100 ns (eight CLK periods) and LA[4:2], R/W*, and LD[31:0] must be driven valid until SELECTLM* is deasserted. On register reads, CY7C961 drives LD[7:0] two clocks after SELECTLM* is sampled asserted and three-states LD[7:0] one clock after SELECTLM* is sampled deasserted. Refer to *Figure 3-37* for Local Register access signaling.

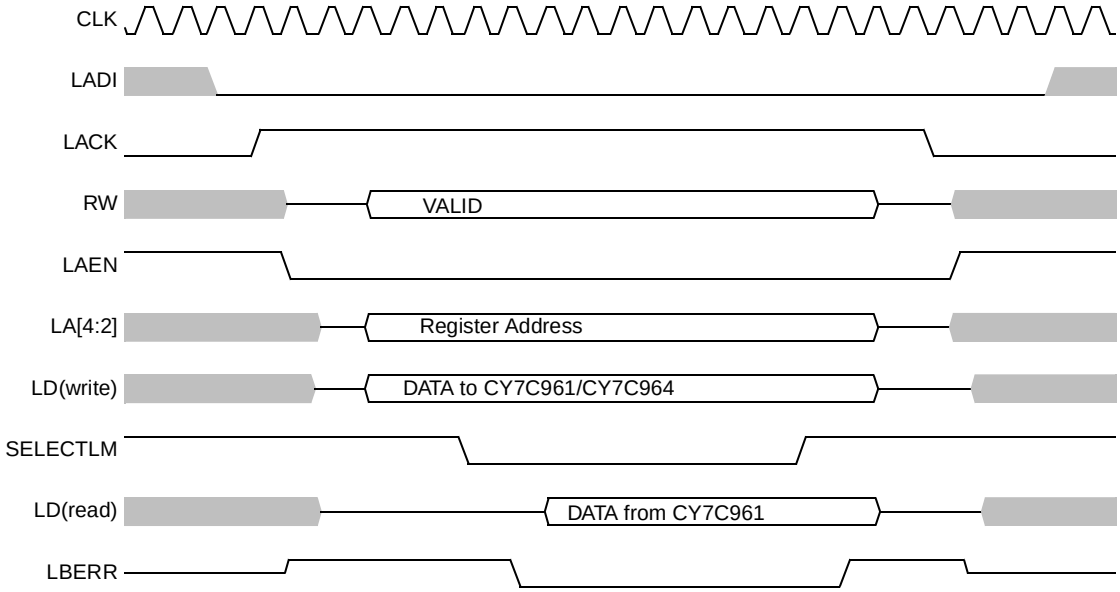


Figure 3-37. DMA Control Register Local Access

The CY7C961 provides control signals to the CY7C964 to control the loading of registers inside the device for operations such as establishing the local starting address and the VME-bus starting address, both of which could be 32-bit quantities stored inside the CY7C964s.

If an illegal register access is attempted, the CY7C961 will signal by driving LBERR* Low within two clocks after SELECTLM* is asserted. LBERR* deasserts one clock after SELECTLM* is sampled deasserted.

3.11.3.4 Programming the Master Block Facility

In general, the set-up programming is very simple. First the semaphore must be read. This register read unlocks the facility. Next, values are written to Transfer Length, Transfer Type, Upper Address (optional), and Master Block StatusID (optional). These parameters need not be refreshed each time the facility is used. After that, the VME Starting Address is written, then Local Starting Address and GO. This last register write operation starts the block transfer. The CY7C961 will assert BR* and, upon being granted the VMEbus, move data until the transfer length is exhausted.

Note that the CY7C961 can be programmed through the initialization bit stream to release and re-request the VMEbus at every 256-byte boundary as DMA transfer progresses. This

behavior can be useful in providing VMEbus access to high-priority traffic interleaved with the DMA transaction.

If more than one VMEbus master is to control the block transfer facility, or if the facility is dual-ported between a local and one or more VMEbus masters, the Semaphore Test & Set register can be polled by any master to determine if the facility is “idle” or “busy.” If the polling result is “idle”, that register read sets the semaphore to “busy”, eliminating the need for a read-modify operation. The semaphore can be written to “idle.” The semaphore is changed by the register write, but this operation is always acknowledged with an error response by the CY7C961.

The semaphore is reset by the CY7C961 on block transfer completion. To do another block transfer requires a minimum of three register accesses, namely, (1) read Semaphore Test & Set, (2) write VME Starting Address, and (3) write Local Starting Address and GO. Of course any of the set-up parameters can be read (with some limitations) or written before writing the GO register. Once the Local Starting Address and GO register is written, all register write attempts are greeted with error acknowledge, cancellation of the DMA operation, and reset of the semaphore. Register reads are always permitted.

The CY7C961 is equipped with BERR* and LBERR* inputs which can be used to interrupt a DMA transfer at any time after it is started (BBSY* is asserted). No consideration as to the timing of BERR* or LBERR* assertion with respect to the progress of the DMA needs to be given. These signals provide a mechanism for aborting DMA transfers in favor of higher priority VMEbus traffic.

Table 3-11. Master Block Transfer Control Registers

LA[4:0]	Register Function
000xx	Semaphore Test & Set (read only)
001xx	Transfer Length Multiplier 0 (TLM0)
010xx	Transfer Length Multiplier 1 (TLM1)
011xx	Transfer Type
100xx	Local Starting Address and GO
101xx	VME Starting Address
110xx	A40/A64 Upper Address
111xx	Master Block StatusID & Interrupt Enable

3.11.3.5 Register Definitions

The following descriptions apply equally to VMEbus register and local register access, but are written from the VMEbus perspective. Local access occurs in the context of “local bus holdoff” as previously described, and has as its error response LBERR* substituted for BERR*.

3.11.3.5.1 Semaphore Test & Set (read only)

Location 000xx is a semaphore test and set. It is a read cycle. CY7C961 responds with the current value of the semaphore. If the Master Block resource of the CY7C961 is idle, this read will return logic “1” on LD0 which is reflected on D0 of the VME data bus. This register read causes the semaphore to be set to busy. The CY7C961 will now respond to additional register accesses. Any subsequent read of the semaphore will return logic “0” on LD[0] and D[0] indicating that the block transfer resource is “busy.”

The semaphore is cleared by the CY7C961 upon completion of a block transfer or when an illegal combination of register values is held when a transfer is attempted. If the interrupt on completion feature is enabled, semaphore clear is delayed until the interrupt generated by block transfer completion is serviced. When BERR* is received during a block transfer, the transfer terminates and the semaphore is cleared. If the interrupt on completion feature is enabled, an interrupt will be signaled on termination and the semaphore clear is delayed until that interrupt is serviced.

Any write to location 000xx will cause the CY7C961 to drive BERR* and clear the semaphore. If access to any of the seven other defined registers is attempted before the semaphore is in the “busy” state, CY7C961 will BERR* those attempts and the semaphore will remain in the idle state.

The CY7C961 drives status conditions on LD[6:1] passed through to D[6:1] of the VMEbus during semaphore read. The normal value for these status bits is logic “1.” A logic “0” on any bit indicates a condition which would prevent a block transfer from starting (cause CY7C961 to BERR a write to Local Starting Address and GO). *Table 3-12* summarizes the semaphore Test and Set Status Bit functions.

Table 3-12. Semaphore Test and Set Status Bits

Status Bit	Error Indication
bit 0	Semaphore. (Logic “1” if idle)
bit 1	Master Block interrupt is pending.
bit 2	Transfer Length Multiplier registers are 0.
bit 3	Transfer Type is undefined.
bit 4	Data size is incompatible with VME or local starting address.
bit 5	Address alignment violated on multiplexed data BLT.
bit 6	VME starting address has not been updated.
bit 7	BERR* or LBERR* asserted during transaction.

A logic “0” on bit 1 means that interrupt on completion is enabled and a block transfer has terminated or completed, but the interrupt signalled on termination/completion has not yet been serviced by a VMEbus interrupt handler. Bit 0 is always logic “0” in this case since the

semaphore must be in state “busy.” Any attempt to access other registers of the block transfer facility while bit 1 is logic “0” will be BERRed by the CY7C961. This status bit will clear with the pending interrupt.

A logic “0” on bit 2 indicates that both registers Transfer Length Multiplier1 and Transfer Length Multiplier0 contain values of zero. The transfer length multiplier must be nonzero. To clear this status bit, a nonzero value must be written to either Transfer Length Multiplier register.

A logic “0” on bit 3 indicates the value in the transfer type register is undefined. To clear this condition, a valid code must be written to the Transfer Type register. (See Transfer Type below.)

A logic “0” on bit 4 means that either the local starting address or VME starting address is not compatible with the programmed data size. This error condition amounts to specifying unaligned starting addresses for a block transfer. For example, a local starting address with an LSB of 0x03 specified with a data size of D16 would result in an unaligned transfer request. Similarly, a VME starting address LSB of 0x02 specified for a D32 operation would start at an unaligned address. Either condition would cause this status bit to be logic “0.” This status bit is computed on the current values of VMEbus and local starting addresses. Since the Local Starting Address and GO register is written at the time a block transfer is started, an alignment error on the local starting address will always cause CY7C961 to assert BERR*. A subsequent read of the Semaphore Test and Set will indicate the error on this status bit.

A logic “0” on bit 5 indicates violation of a restriction placed on starting address when the transaction type is multiplexed data. The restriction is that VME starting address [7:0] be equal to local starting address [7:0]. This restriction is in addition to the address alignment requirement described with respect to status bit 4 above. The restriction applies expressly to transfer type codes: “0110011x”, “0110111x”, “0111010x”, “0101011x”, “0101111x”, and “0111111x”. This status bit is computed on the current values of VMEbus and local starting addresses. Since the Local Starting Address and GO register is written at the time a block transfer is started, a violation of this restriction may be created when a new local starting address is written causing CY7C961 to assert BERR*. A subsequent read of the Semaphore Test and Set will indicate the error on this status bit.

A logic “0” on bit 6 indicates that the VME starting address has not been written since the last master block transfer was started. This status bit is cleared by writing a new VME Starting Address.

3.11.3.5.2 Transfer Length Multiplier(1 & 0)

Two registers, Transfer Length Multiplier1 and Transfer Length Multiplier0 hold respectively the MSB and LSB of a 16-bit transfer length multiplier parameter. The transfer length of a block transfer in bytes can be computed by multiplying the transfer length multiplier by a block length factor for that transfer type. The block length factor is a function of the transfer data size as shown in *Table 3-13*.

Table 3-13. Transfer Length Calculation

Data Size	Block Length Factor	TLM1,TLM0	Transfer Length
D8	16 Bytes	X TLM[15:0]	= BYTES
D16	32 Bytes	X TLM[15:0]	= BYTES
D32	64 Bytes	X TLM[15:0]	= BYTES
D64	128 Bytes	X TLM[15:0]	= BYTES

For example, if TLM1= 4 and TLM0 = 6 and the data size is D32 (specified in the Transfer Type register), then the transfer length for the block transfer would be 64 X 1030 = 65,920 bytes. The number of data cycles is always 16 times the transfer length multiplier, so the data bytes transferred is obviously proportional to the width of the transaction. The transfer length multiplier registers are read/write with data sent/received on LD[7:0] and reflected on D[7:0] of the VMEbus. Bit 2 of the Semaphore Test and Set status word will be logic “0” if both TLM1and TLM0 are set to 0. This is an error condition which will block a transfer start. Both registers are cleared to 0 after power up or SYSRESET*.

3.11.3.5.3 Transfer Type

The Transfer type register specifies which of 22 possible block transfer operations is to be performed. The user can select from among A40BLT, D64MBLT, D32BLT, D16BLT, D8BLT, and single cycle block move options. The 8-bit hex codes shown in *Table 3-14* are valid contents of the transfer type register:

Table 3-14. Transfer Type Field—Block Transfer Operations

Transfer Type[7:0]	AmCode	Block Transfer Description
20,21,22,23,24,25	3F	A24 supervisory block transfer (D8BLT or D16BLT or D32BLT)
A8,A9,AA,AB,AC,AD	3E	A24 supervisory program access (D8 or D16 or D32)
B0,B1,B2,B3,B4,B5	3D	A24 supervisory data access (D8 or D16 or D32)
66,67	3C	A24 supervisory 64-bit block transfer (MBLT)
28,29,2A,2B,2C,2D	3B	A24 nonprivileged block transfer (D8BLT or D16BLT or D32BLT)
B8,B9,BA,BB,BC,BD	3A	A24 nonprivileged program access (D8 or D16 or D32)
A0,A1,A2,A3,A4,A5	39	A24 nonprivileged data access (D8 or D16 or D32)
6E,6F	38	A24 nonprivileged 64-bit block transfer (MBLT)
40,41,42,43	37	A40 block transfer (D8BLT or D16BLT)
74,75	37	A40 32-bit block transfer (MD32)
90,91,92,93,94,95	2D	A16 supervisory access (D8 or D16 or D32)
98,99,9A,9B,9C,9D	29	A16 nonprivileged access (D8 or D16 or D32)

Table 3-14. Transfer Type Field—Block Transfer Operations (continued)

Transfer Type[7:0]	AmCode	Block Transfer Description
10,11,12,13,14,15	0F	A32 supervisory block transfer (D8BLT or D16BLT or D32BLT)
C0,C1,C2,C3,C4,C5	0E	A32 supervisory program access (D8 or D16 or D32)
C8,C9,CA,CB,CC,CD	0D	A32 supervisory data access (D8 or D16 or D32)
56,57	0C	A32 supervisory 64-bit block transfer (MBLT)
18,19,1A,1B,1C,1D	0B	A32 nonprivileged block transfer (D8BLT or D16BLT or D32BLT)
D0,D1,D2,D3,D4,D5	0A	A32 nonprivileged program access (D8 or D16 or D32)
D8,D9,DA,DB,DC,DD	09	A32 nonprivileged data access (D8 or D16 or D32)
5E,5F	08	A32 nonprivileged 64-bit block transfer (MBLT)
48,49,4A,4B,4C,4D	03	A64 block transfer (D8BLT or D16BLT or D32BLT)
7E,7F	00	A64 64-bit block transfer (MBLT)

Bits [2:1] of the Transfer Type register specify the data size of the block transfer. *Table 3-15* defines this Transfer Type field. The CY7C961 checks for consistency between transfer type and data size. Illogical combinations will be reported on Semaphore Test and Set status bit 3 as an undefined transfer type and will prevent the start of a block transfer.

Table 3-15. Transfer Type Field—Data Size

Transfer Type [2:1]	Block Transfer Data Size
00	D8
01	D16
10	D32
11	D64

Bit 0 of the Transfer Type register specifies block transfer data direction. Logic “1” specifies master read with data moving to the CY7C961 interface, logic “0” specifies master write with data moving from the CY7C961interface to a VMEbus slave.

The transfer type register is read or written with data transmitted on LD[7:0] and reflected on D[7:0] of the VMEbus. Illegal codes cause block transfer to abort when the GO transaction is received. Both undefined transfer type fields and illogical transfer type data size combinations will BERR when Local Starting Address and GO is written.

3.11.3.5.4 Local Starting Address & GO

The Local starting address and GO register is used to write the local block address to be used in the DMA block transfer. This address can be up to 32 bits of address information provided as VME data when the register is written. Local starting address is loaded directly

into the CY7C964's local address counters C1 and the CY7C961 local address register. A read of the local starting address will yield only the LSB of the starting address on LD[7:0] reflected on D[7:0] of the VME data bus.

The CY7C961 signal BLT* is driven low for 2 CLK periods to load the local address into the CY7C964s from the LD bus. The value of LD on the rising edge of BLT* is stored in the CY7C964 counter C1. Refer to *Figure 3-38*.

The control function of this register is activated only for writes to the register. GO causes the start of the block transfer by signaling the CY7C961 requester to take the VMEbus and begin the transfer. Once CY7C961 acquires the VMEbus, it will not release the VMEbus until the block transfer finishes or is terminated by a BERR* during a transfer attempt. BBSY* is released when the transfer length count is exhausted, not at VMEbus boundary crossings. Any

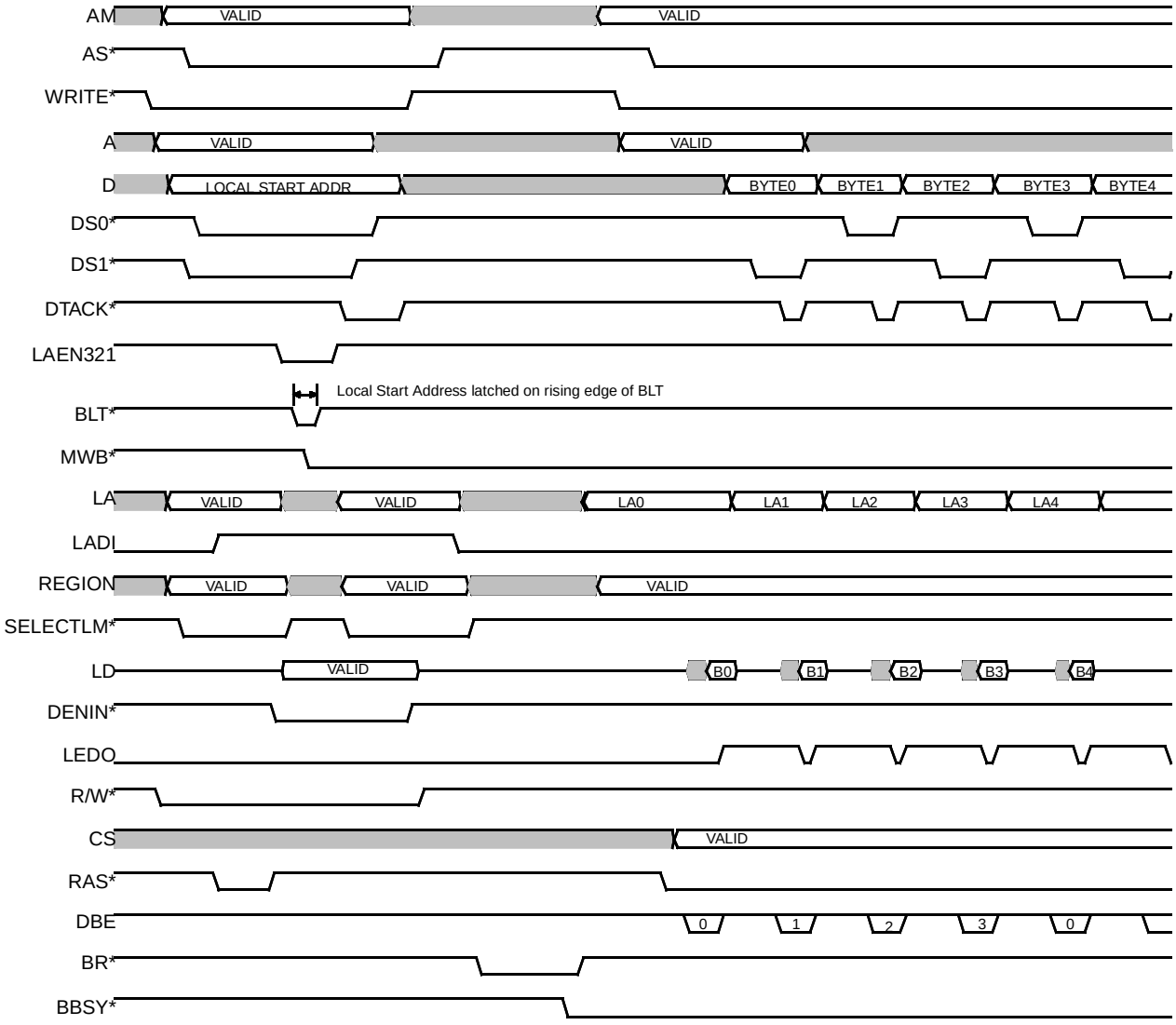


Figure 3-38. Local Starting Address and Go, Block Transfer Start Timing

write to a set-up register after the GO transaction will be BERRed by CY7C961, the block move aborted, and the semaphore reset.

3.11.3.5.5 VME Starting Address

The VME Starting Address register is used to write the VMEbus block address to be used in the DMA block transfer. This address can be up to 32 bits of address information written as VME data to the CY7C961. It is actually stored in latch L9 of the CY7C964s and the LSB of the address is also captured by the CY7C961 to facilitate counting to the VME address boundary. A read of the VMEbus Starting Address will yield only the LSB of the starting address on LD[7:0] reflected on D[7:0] of the VME data bus. Note that this specifier is a true byte address. The CY7C961 takes care of VMEbus LWORD* and DS1*/DS0* encoding. This register must be written each time a block transfer is executed. Bit 6 of the Semaphore Test and Set status register will indicate when the VME starting address has not been updated. This requirement guarantees that the CY7C964 and CY7C961 will be using the same VMEbus starting address.

3.11.3.5.6 A40/A64 Upper Address

A40/A64 Upper Address register is not a register residing in either the CY7C961 or CY7C964. Instead, it is a facility for reading/writing extended address information to external hardware on the slave card that is not part of the CY7C961/CY7C964 interface. Latch and enable signaling allows for reading and writing during register access as well as data enable during address broadcast as the DMA block transfer runs. A40 and A64 extensions are supported.

The latch signal for capturing extended address from the local data bus is (SELECTLM* & LEDI) where LEDI is a CY7C961 output. The active Low enable for driving the latched data onto the local data bus is (LDEN* | MWB*). If the extended address is to be incremented, VCOUT* of the appropriate CY7C964 and LADO are used as count enable and clock respectively for extended address counting. Refer to *Figure 3-39* for signaling.

The latch signal is designed to latch up to 32 bits of upper address from LD[31:0] passed through the interface from VME D[31:0] when Upper Address is written. The enable signal is designed to allow the stored address to be read from the VMEbus when Upper Data is read, as well as to enable the stored address onto LD when the master block transfer address broadcast requires it.

3.11.3.5.7 Master Block StatusID & Interrupt Enable

Master Block status ID & Interrupt Enable is a CY7C961 register which holds the Master Block statusID byte. This register can be read or written. Bits 7–1 of the register will be reflected in the statusID byte sourced by CY7C961 when the Master Block interrupt is serviced. Bit 0 controls enabling of interrupt on completion: logic “1” enables interrupt on completion.

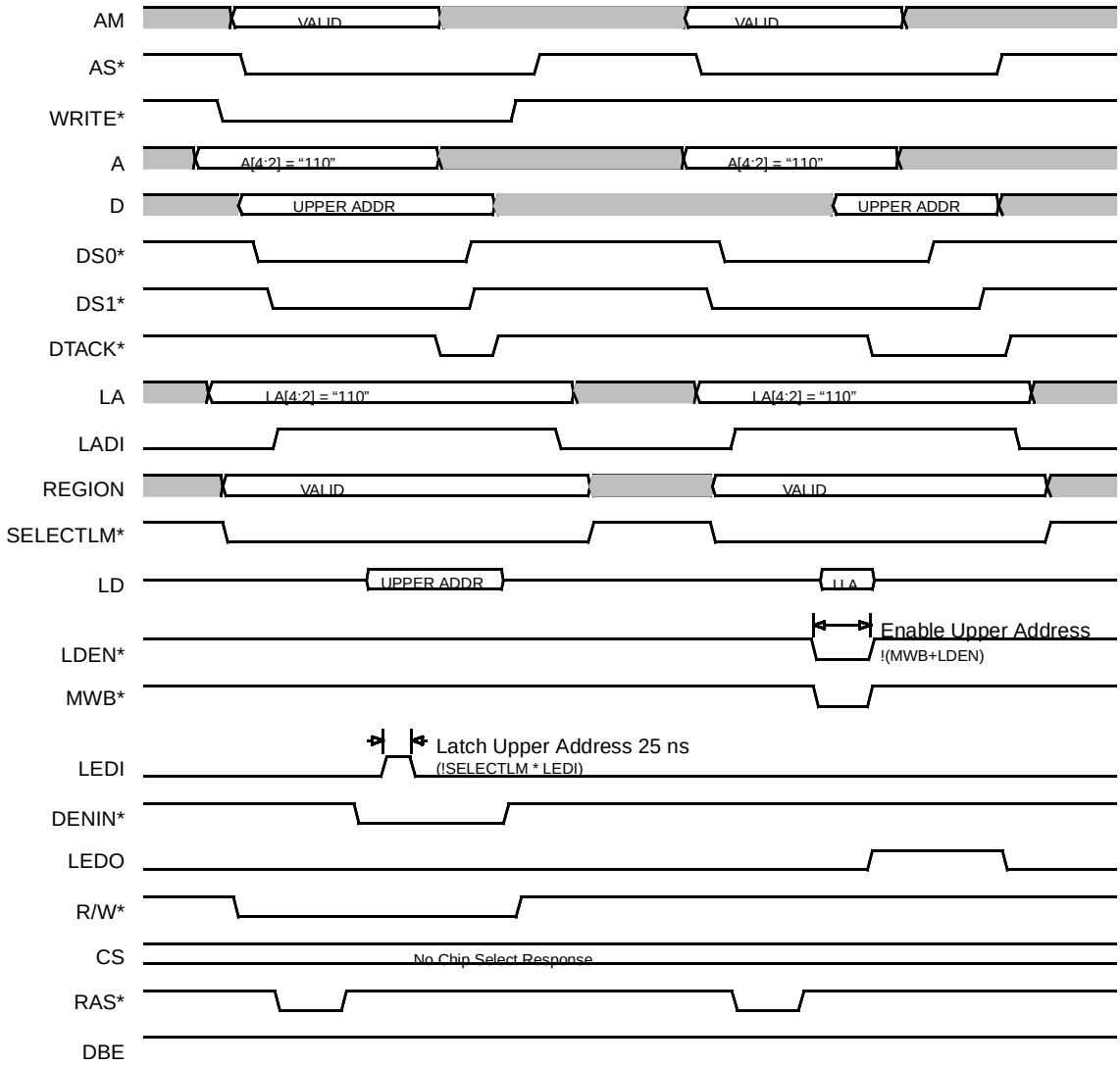


Figure 3-39. Upper Address Register Access Timing

When the interrupter status is read by the handler, bit 0 of the statusID indicates completion status. Logic “1” means normal completion. Logic “0” means abnormal termination (BERR* or LBERR* was received during block transfer). The interrupter is a Release on Acknowledge (ROAK) interrupter. If interrupt on completion is enabled, CY7C961 will reset its semaphore on interrupt acknowledge. If interrupt on completion is not enabled, CY7C961 will reset its semaphore with BBSY deassertion at the end of the DMA block transfer. Refer to *Figure 3-40* for Interrupter timing.

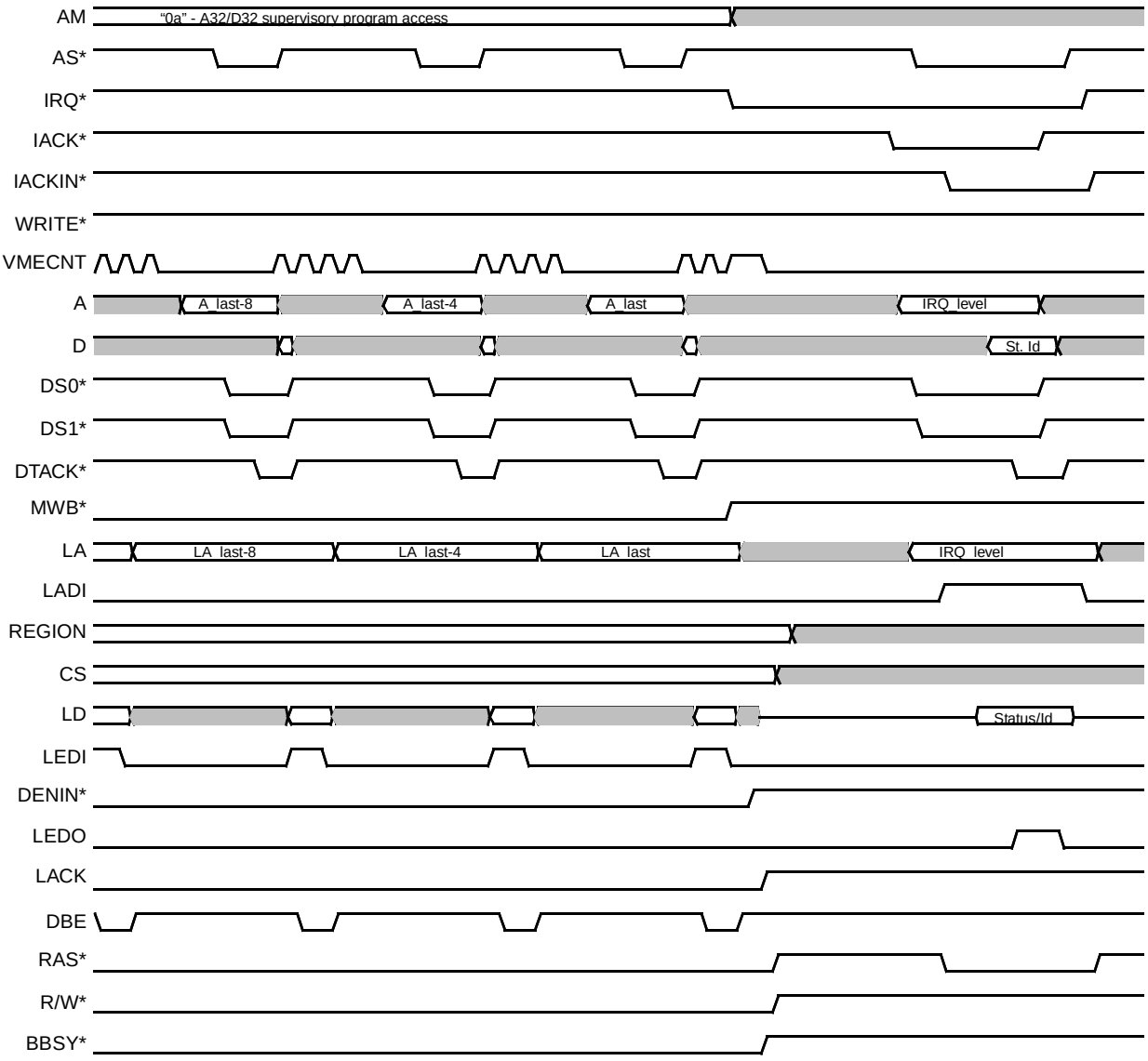
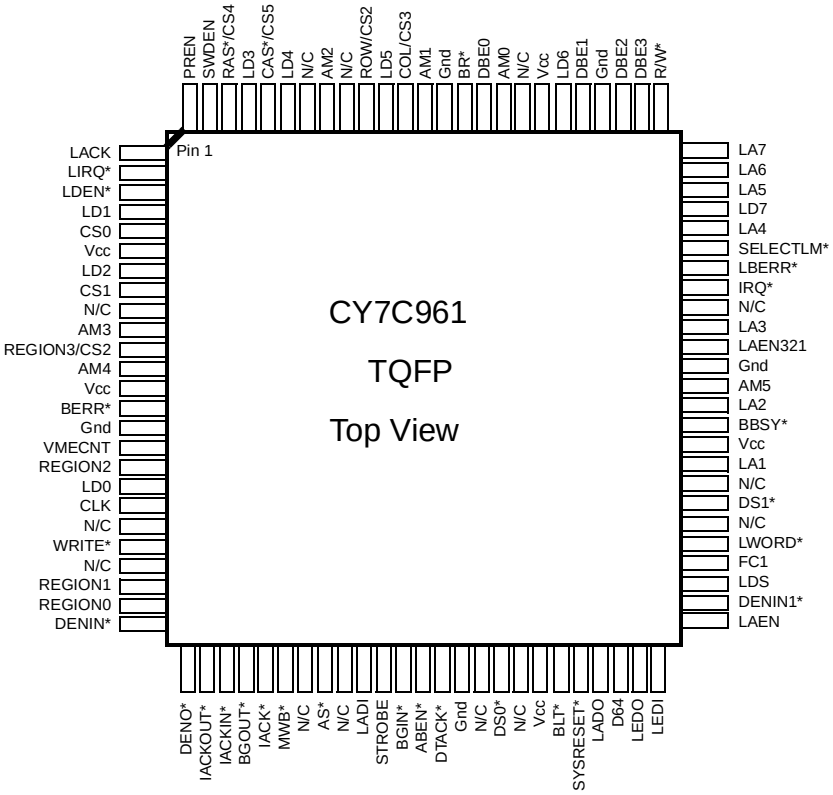


Figure 3-40. Interrupter Timing

3.11.4 Pin Description Addendum



The following signals are either additional to, or redefined from, the CY7C960 pin descriptions in Chapter 3.3.

3.11.4.1 VMEbus Signals

AM[5:0] - VMEbus Address Modifier

Input: Yes
Output: Yes
Drive: 48 mA

Signals AM[5:0] are the VMEbus Address Modifier I/Os. When inputs, these signals are used to decode the VMEbus data transaction type. The CY7C961 provides support for both pre-defined and user-defined VMEbus Address Modifiers. During master block transfers VMEbus AM codes are driven out on these signal pins.

AS* - VMEbus Address Strobe

Input: Yes
Output: Yes
Drive: 64 mA
Active: Low

Address Strobe is the VMEbus signal that informs VMEbus slaves that a valid address is on the VMEbus. This signal is used by the CY7C961 to qualify the VMEbus Address Modifiers AM[5:0] and REGION[3:0] inputs to determine if a valid slave cycle should be performed. During master block transfers VMEbus AS* is driven out on this signal pin.

DS0*,DS1* - VMEbus Data Strobes

Input: Yes
Output: Yes
Drive: 64 mA
Active: Low

DS0* and DS1* are the VMEbus Data Strobes. As inputs, these signals inform the CY7C961 that the data phase of the VMEbus cycle has begun. These signals in conjunction with the VMEbus LWORD* (connected to LA[0]) signal encode the data transfer width or number of bytes, 1 through 4. This information is necessary to enable the appropriate CY7C964 data bytes. During master block transfers VMEbus DS0* and DS1* are driven out on these signal pins.

WRITE* - VMEbus Write Line

Input: Yes
Output: Yes
Drive: 48 mA
Active: Low

The VMEbus WRITE* line specifies the data direction of the VMEbus data cycle in progress. If this signal is asserted then a VMEbus WRITE operation is in progress. During such a transaction, if the VMEbus address decodes properly, the CY7C961 responds by asserting the local R/W* signal and performing the appropriate local cycle. During master block transfers, VMEbus WRITE* is driven out on this signal pin indicating the direction of data for the DMA transfer.

DTACK* - VMEbus Data Acknowledge

Input: Yes
Output: Yes
Drive: 64 mA
Active: Low

The DTACK* signal is asserted by the CY7C961 when a valid VMEbus transaction is in progress and has remained valid for the proper length of time. The assertion of this signal informs the VMEbus Master that the slave has either accepted the data during write operations or has sourced the data during read operations. This signal is a rescinding output. During master block transfers, the CY7C961 receives VMEbus data acknowledge from the target slave.

BR* - VMEbus Request

Input: No
Output: Yes
Drive: 48 mA
Active: Low

Signal BR* is the VMEbus Request output. This output is open-collector with a low state sink current of 48 mA. It is asserted by the CY7C961 when the VMEbus is required for a block transfer.

BGIN* - VMEbus Bus Grant In

Input: Yes
Output: No
Active: Low

BGIN* is a VMEbus Bus Grant In signal. It is generated by the VMEbus arbiter and signals that CY7C961 may use the VMEbus. BGIN* and BGOUT* signals form a bus grant daisy chain.

BGOUT* - VMEbus Bus Grant Out

Input: No
Output: Yes
Drive: 8 mA
Active: Low

BGOUT* is a VMEbus Bus Grant Out signal. It is driven Low by the CY7C961 in response to an assertion of the BGIN* signal when CY7C961 does not want to use the VMEbus.

BBSY* - VMEbus Bus Busy

Input: Yes
Output: Yes
Drive: 64 mA
Active: Low

BBSY* is the VMEbus bus busy signal. It is driven Low by the CY7C961 to indicate the VMEbus is being used for a block transfer. When the block transfer operation is done, CY7C961 drives BBSY* High and then three-states the signal. In response to decoded VMEbus lock cycles, the CY7C961 monitors BBSY* to determine when a locked VMEbus sequence is ending.

BERR* - VMEbus Bus Error

Input: Yes
Output: Yes
Drive: 48 mA
Active: Low

BERR* is the VMEbus bus error signal. It is driven Low by CY7C961 in two cases. First, decoded slave accesses which attempt to illegally configure the CY7C961 block transfer facility will result in a BERR* acknowledge. Second, BERR* will be signaled to acknowledge a slave block data cycle if LBERR* is sampled asserted at the time VMEbus DSA* was received

by the CY7C961. BERR* is three-stated on the deassertion of VMEbus signals DS0* and DS1*. CY7C961 monitors BERR* during master block transfer operation, truncating the block transfer if a BERR* acknowledge is detected. BERR* assertion will also signal the end of any locked VMEbus sequence.

3.11.4.2 Local Buffer Control Signals

BLT*

Input: No
Output: Yes
Drive: 8 mA
Active: Low

BLT* is driven Low for 2 clock periods during the block transfer register access “Local starting address and GO.” The local starting address for block transfer is latched into CY7C964 internal counters on the High-going edge of this signal. The CY7C961 BLT* pin should be connected to the BLT* pin of each CY7C964 in the interface.

MWB*

Input: No
Output: Yes
Drive: 8 mA
Active: Low

MWB* is driven Low by CY7C961 during the block transfer register access “Local starting address and GO.” It remains in the Low state throughout master block transfer except at 256-byte local boundaries. When a 256-byte local boundary is crossed, MWB* will be pulsed High for one clock period. The High to Low transition on this pulse increments the interface CY7C964 local block counters. MWB* is deasserted on completion of a block transfer. MWB* also signals VMEbus lock status. MWB* is driven Low on decode of a VMEbus lock cycle. It is deasserted when the VMEbus lock sequence is complete. The CY7C961 MWB* pin should be connected to the MWB* pin of each CY7C964 in the interface.

LADO

Input: No
Output: Yes
Drive: 8 mA

LADO is pulsed High by CY7C961 at VMEbus 256-byte boundaries during master block transfers. LADO is driven High at the same time VMEbus data strobes are asserted for the last cycle of the address page. LADO is driven Low one clock after DTACK* is detected Low in that cycle. The High to Low transition on LADO increments the VMEbus block counters of the interface CY7C964s. The CY7C961 LADO pin should be connected to the LADO pins of all interface CY7C964s EXCEPT the one connected to VMEbus addresses A[7:1].

FC1

Input: No
Output: Yes
Drive: 8 mA
Active: High

FC1 is driven High by CY7C961 at the beginning of the block transfer's VMEbus tenure. It is deasserted at the end of the block transfer's VMEbus tenure. The CY7C961 FC1 pin should be connected to the FC1 pins of each CY7C964 in the interface. This signal can serve as a DMA complete indicator for local control purposes.

VMECNT

Input: No
Output: Yes
Drive: 8 mA

VMECNT is a clock signal driven by CY7C961 to the LADO pin of the CY7C964 connected to VMEbus addresses A[7:1]. This signal adjusts the VMEbus block counter of this CY7C964 to source LWORD correctly for D8 and D16 block transfers. VMECNT also burst clocks this block counter to $h00 in cases where the VMEbus block starting address is not aligned to a 256-byte boundary. The clock pulse width is one CLK period.

LAEN321

Input: No
Output: Yes
Drive: 8 mA
Active: High

LAEN321 is driven Low for four CLK periods during the block transfer register access "Local starting address and GO." The High to Low transition happens one clock before the High to Low transition on BLT*. LAEN321 is used in conjunction with BLT* and MWB* to place the interface CY7C964s in BLT_STATE. The CY7C961 LAEN321 pin should be connected to the LAEN pins of all interface CY7C964s EXCEPT the one connected to VMEbus addresses A[7:1].

3.11.4.3 Local Signals

LD[7:0] - Local Data Signals

Input: Yes
Output: Yes
Drive: 8 mA

LD[7:0] make up the local bidirectional data bus. These pins should be connected to LD[7:0] of the interface CY7C964 connected to VMEbus data D[7:0]. During block transfer register accesses and block transfer complete interrupt acknowledge accesses, data is read from or written to the CY7C961 via this data port.

SELECTLM* - Select Load Master Signal

Input: Yes
Output: No
Active: Low

SELECTLM* is a chip select for accessing the eight CY7C961 registers that control the block transfer facility. A decoded VMEbus slave cycle for which SELECTLM* is also asserted will be interpreted by CY7C961 as a block transfer register access. SELECTLM* timing should imitate that of the REGION input.

LBERR* - Local Bus Error Signal

Input: Yes
Output: Yes
Drive: 8 mA
Active: Low

LBERR* has three distinct functions. First, it provides a mechanism for aborting DMA transfers in progress. At any time after the GO register access completes, LBERR* may be asserted to cause DMA completion (with error status). If interrupt on completion is enabled, IRQ* will be asserted within two CLK periods after LBERR* assertion. A second function of LBERR* is as an error signaling mechanism for slave block transfer. Each time the VMEbus DSA* signal is asserted to the CY7C961, the state of the LBERR* signal is sampled. If it is asserted, that VMEbus cycle will receive a BERR* acknowledge instead of a DTACK* acknowledge. The third function of LBERR* is as an error acknowledge output for DMA register accesses during “local bus holdoff.” LBERR* will be asserted (driven low) by CY7C961 to indicate an error condition during a register access in progress. Note that LBERR* is always driving out of the CY7C961 during “local bus holdoff.”

R/W* - Read/Write

Input: Yes
Output: Yes
Drive: 8 mA

R/W* is the local signal that determines if the cycle in progress is a read operation or a write operation. The CY7C961 asserts this signal Low during write operations.

When accessing the Master Block Facility from the local side of the interface - R/W* is an input to the CY7C961. The local bus holdoff feature of the CY7C961 must be enabled before register access is initiated.

3.11.4.4 Master Block Transfer Performance

From the perspective of local bus timing, there is no difference between local bus signaling for master DMA and slave accesses. The master block facility uses the same circuits and the same self-timed acknowledge constants as the slave. Master block transfers sample REGION

at each 256-byte boundary. This allows a block transfer to, for example, start in an SRAM region and transparently cross a hardware boundary into a DRAM region.

The master block facility has performance limits set by CY7C961 internal synchronous state machines. 6 CLK periods is the minimum required per VMEbus data cycle for master write operations, 7 CLK periods for master read VMEbus data cycles. Local bus signaling limits D64 transfer rates to 80 megabytes per second, and MD32 to 40 megabytes/second. The performance of D32 block transfer is limited by VMEbus slave response, with an ideal slave response making block read performance of 40 megabytes/second and block write performance of 50 megabytes per second possible. These are, of course, burst maximums. Sustained D64 block transfer rate will be 73 megabytes/second to an ideal slave with local cycles set up for 50 ns.

The block transfers using single-cycle protocol will be considerably slower than the true block protocols because the CY7C961 must increment the CY7C964 after each data cycle. For D32 this represents an 8 CLK period overhead. Transfer rate will not exceed 20 megabytes/second. D16 single-cycle protocol will be down around 12 megabytes/second. D8 single-cycle will top out at 5.5 megabytes/second.

For BLT transfers, CY7C961 will cross VME address boundaries without releasing the VMEbus unless the Interleave function is programmed ON in the serial bit stream. Address strobe will be cycled and address rebroadcast. BBSY does not have an early release mode, and is asserted throughout the block transfer. D64MBLT will rebroadcast at 256-byte boundaries. If single-cycle accesses are specified, the block move will consist of an indivisible packet of single-cycle transfers sufficient to satisfy the transfer length parameter, transfer type dictating the AM code used.

3.11.5 Examples of Block Transfers

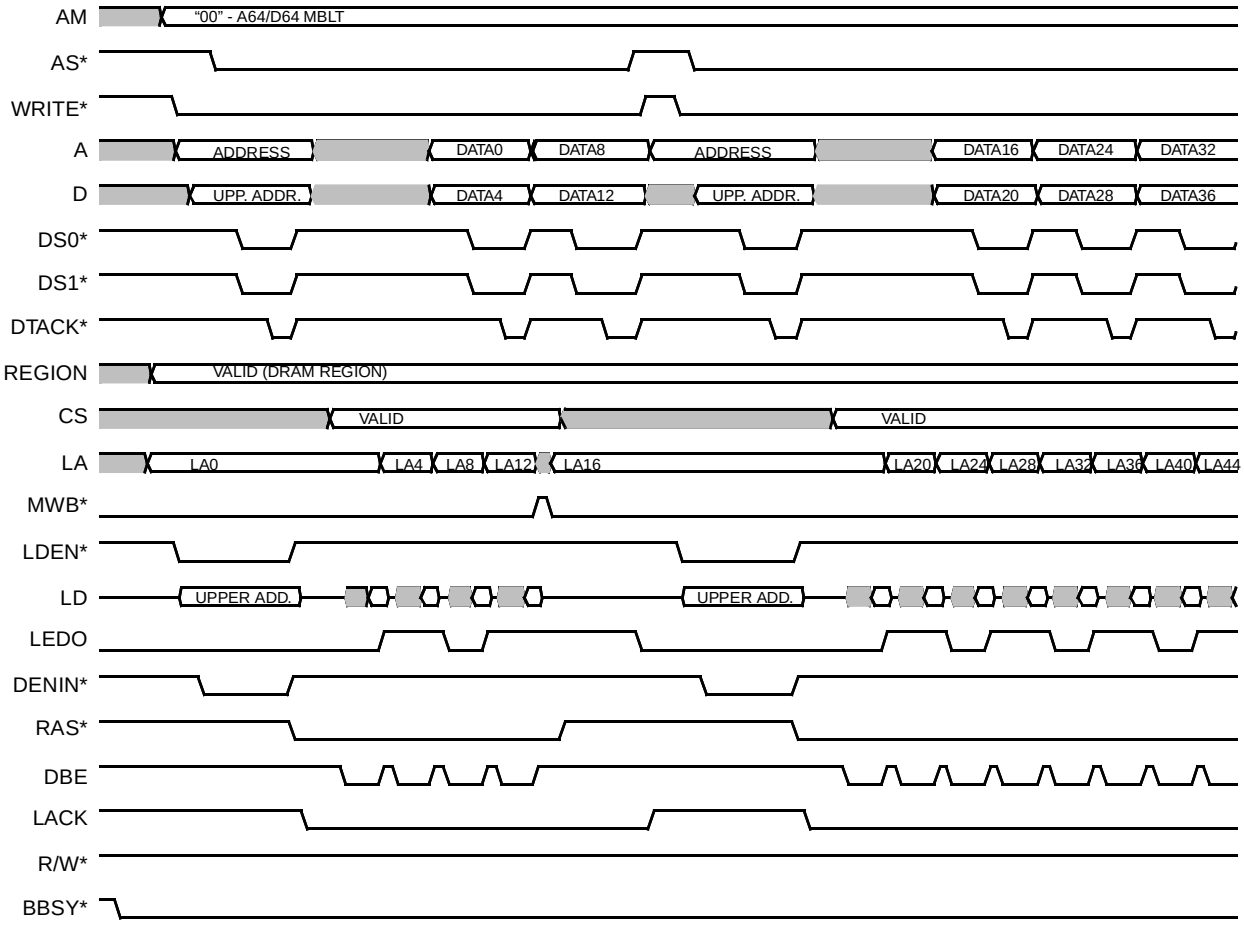


Figure 3-41. A64/D64 MBLT Master Write

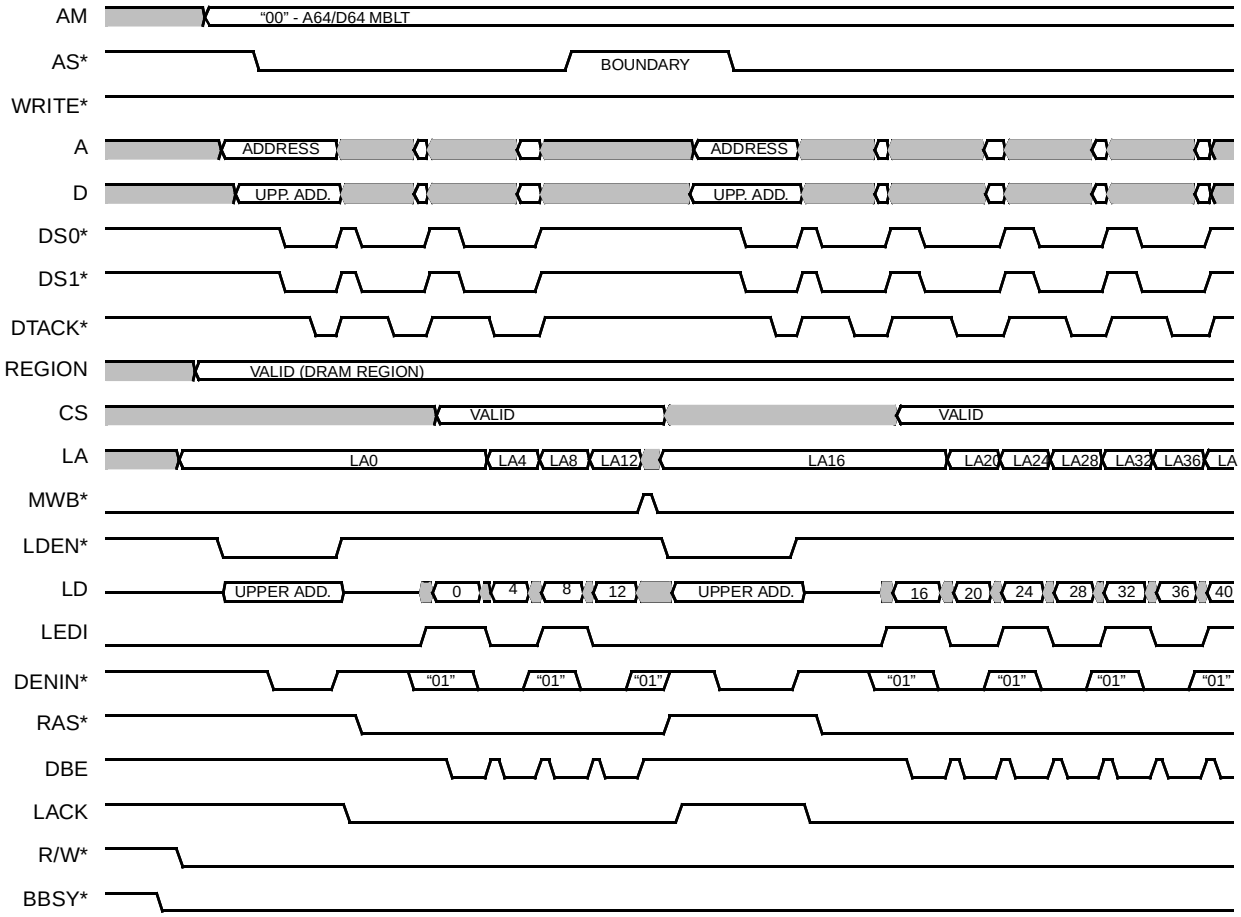


Figure 3-42. CY7C961 A64/D64 MBLT Master Read Example

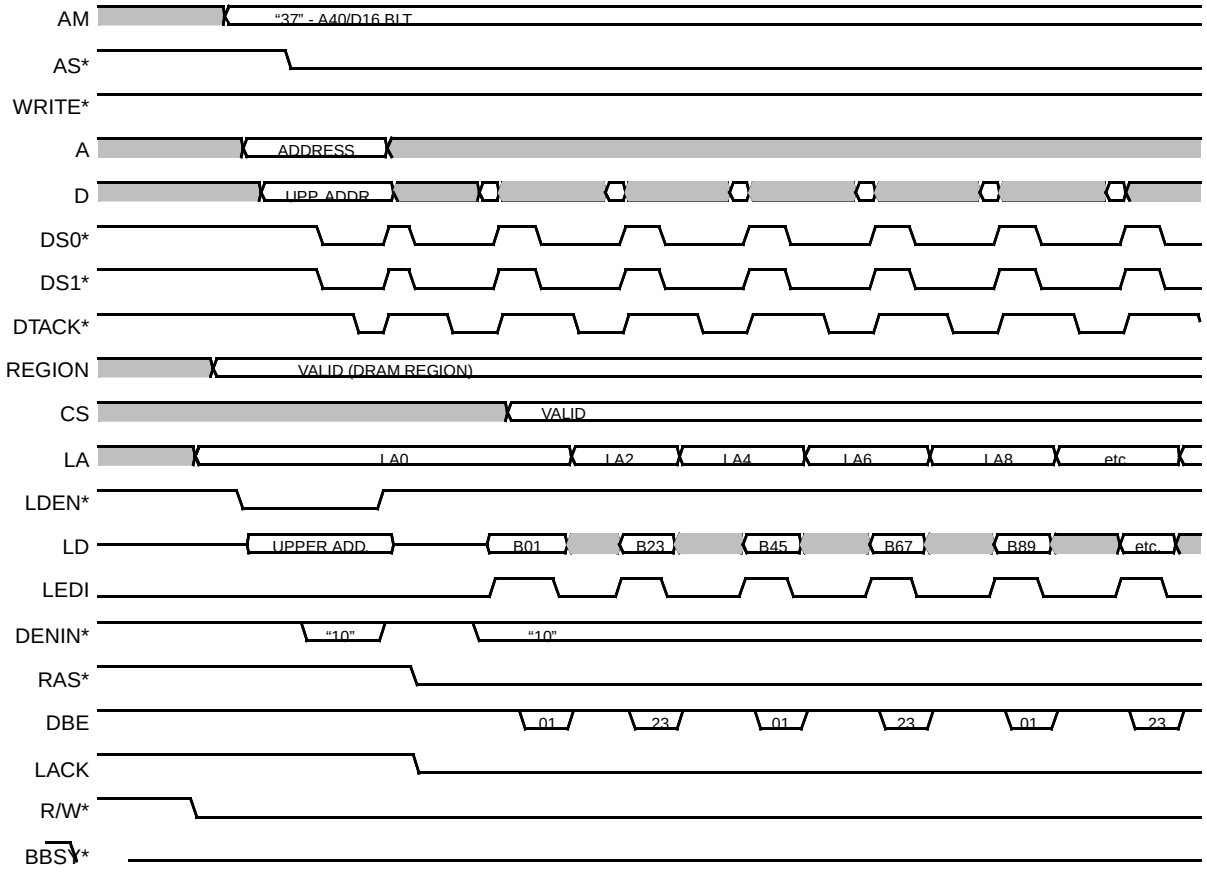


Figure 3-43. CY7C961 A40/D16 BLT Master Read Example

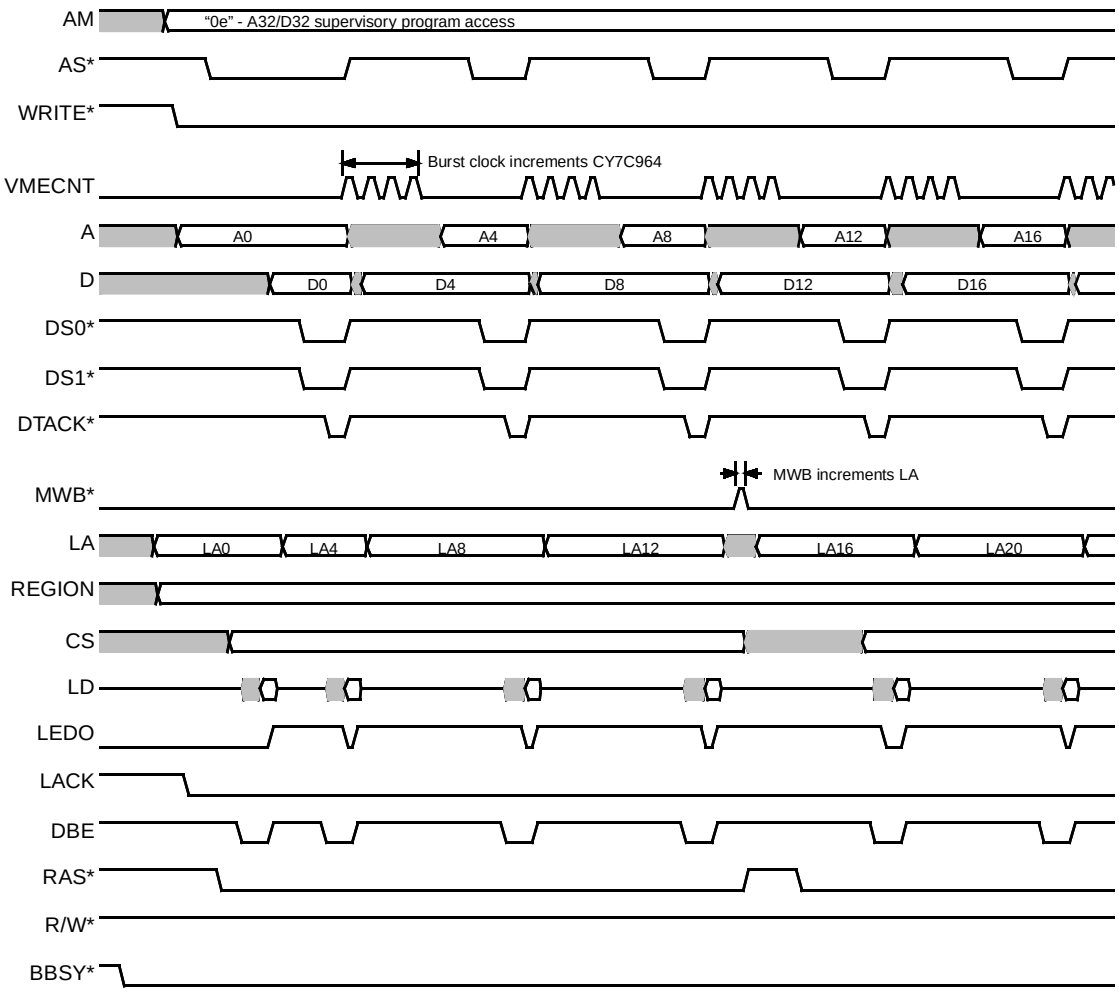


Figure 3-44. A32/D32 Single Cycle Write Example