3.5

VMEbus Interface Description

3.5.1 Definition of Terms

DSi*	Either or both of DS0* and DS1*, the VMEbus data strobe signals
DSA*	The first of DS0* or DS1* to be asserted
DSB*	The second of DS0* or DS1* to be asserted
6U, 3U	The two most common sizes for VMEbus boards

3.5.2 Overview

The CY7C960 and its companion part , the CY7C964, provide a VMEbus interface that is fully compliant with IEEE 1014 rev C (original VMEbus specification) and the VME64 specification. *Table 3−3* lists the slave accesses that are supported on the VMEbus.

Table 3−3. VMEbus Transactions Recognized by the CY7C960

VMEbus Cycle Type	Description
D8, D16, D32	8-, 16-, and 32-Bit Single Cycle Accesses. The VMEbus Master asserts DSi*, and the CY7C960 responds with DTACK*. Data is transferred in one cycle.
ADO	Address Only. A VMEbus Master broadcasts an address on the bus without any data strobe assertion. There is no data phase and no local access associated with this cycle.
ADOH	Address Only With Handshake. A VMEbus Master broadcasts an address on the bus and asserts data strobes. The CY7C960 will handshake the address broadcast by asserting DTACK* Low in response to the assertion of data strobes. There is no data phase and no local access associated with this cycle. (The CY7C960 asserts Chip Selects, but not DBE.)
MD32	Multiplexed 32-Bit Single Cycle Access. A VMEbus Master broadcasts a 40-bit address using the address and data lines. The CY7C960 will handshake the address broadcast by asserting DTACK* Low in response to the assertion of data strobes. The Master then initiates another cycle that uses 16 bits of the data bus and 16 bits of the address bus to transfer 32 bits of data in single bus cycle. This transaction is typically used in 3U applications.

Table 3−3. VMEbus Transactions Recognized by the CY7C960 (continued)

VMEbus Cycle Type	Description
D8:BLT, D16:BLT, D32:BLT	8-, 16-, and 32-Bit Block Transfers. The VMEbus Master asserts DSi*, the CY7C960 handshakes with DTACK*. The Master continues with DSi* assertions. When AS* is deasserted, the block transfer is ended.
MD32:BLT	Multiplexed 32-Bit Block Transfer. A VMEbus Master broadcasts a 40-bit address using the address and data lines. The CY7C960 will handshake the address broadcast by asserting DTACK* Low in response to the assertion of data strobes. The Master then initiates a series of cycles that use 16 bits of the data bus and 16 bits of the address bus to transfer 32 bits of data in each cycle of the burst. This transaction is typically used in 3U applications.
D64:MBLT	Multiplexed 64-Bit Block Transfer. A VMEbus Master broadcasts a 32- or 24-bit address on the address lines or a 64-bit address on the address and data lines. The CY7C960 will handshake the address broadcast by asserting DTACK* Low in response to the assertion of data strobes. The Master then initiates a series of cycles that use 32 bits of the data bus and 32 bits of the address bus to transfer 64 bits of data in each cycle of the burst. When AS* is deasserted, the block transfer is ended.
D8:RMW, D16:RMW, D32:RMW	8-, 16-, and 32-Bit Read-Modify-Write Cycles. A VMEbus Master reads data from the slave board, modifies that data, and writes it back to the board in a single transaction. The CY7C960 provides DTACK* in response to each DSi* assertion.
MD32:RMW	Multiplexed 32-Bit Read-Modify-Write. A VMEbus Master broadcasts a 40-bit address using the address and data lines. The CY7C960 will handshake the address broadcast by asserting DTACK* Low in response to the assertion of data strobes. The Master then initiates a cycle that reads 16 bits of the data bus and 16 bits of the address bus, modifies that data, and writes the data back in a single VMEbus access. This transaction is typically used in 3U applications. The CY7C960 provides DTACK* in response to each DSi* assertion.
D32:UAT	32 Bit Unaligned Transfer. A VMEbus Master transfers 16 or 24 bits of data using a single cycle access. For example, a Master may transfer 16 bits of data using VMEbus data Byte 1 and Byte 2. The CY7C960 provides DTACK* in response to the DSi* assertion..
A16, A24, A32, A40, A64	The CY7960 can be programmed to respond to any of the address-based AM codes.
CR/CSR	Configuration ROM/Control and Status Register accesses. The CY7C960 can be programmed to respond to this AM Code.
SERIAL	Serial cycles. The extension to the VMEbus specification defines the use of certain reserved AM Codes for this serial data interface. The CY7C960 can be programmed to respond to these AM Codes.

Table 3−3. VMEbus Transactions Recognized by the CY7C960 (continued)

VMEbus Cycle Type	Description
LOCK	LOCK cycles. The CY7C960 provides limited support by acknowledging the address broadcast from the VMEbus Master. No local signals are driven. The CY7C961 device provides a complete LOCK facility.
USER	16 VMEbus AM codes grouped as USER1(18 − 1F) and USER2 (10 − 17). The CY7C960 can be programmed to respond to these transactions.
IACK	The VMEbus Interrupt Handler provides an IACK cycle in response to a VMEbus interrupt. The CY7C960 responds to IACK cycles as appropriate.

There are many features of the CY7C960 interface solution which can be combined in various ways and augmented with external decoder logic to configure a slave board to respond in any of 16 VMEbus address regions. The CY7C960 can be programmed to respond to a subset of the transaction types listed in *Table 3−3* when they occur within user-specified regions of VMEbus address space. (Refer to Programming the CY7C960 for complete details on how to configure the device.) This section will describe how the CY7C960 uses its AM code programming and REGION inputs to respond to certain transactions within certain address regions and how the CY7C960 is selected by a Master. As the CY7C960 is designed to handle the complexity of the VMEbus transaction processing without external assistance, it is not necessary to understand the internal circuitry to any great extent. The following two sections provide the basic knowledge needed to use the device: the rest of the chapter can be viewed as reference material.

3.5.3 Region Mapping

The CY7C960 receives VMEbus address decoder information on its REGION inputs. Slave selection begins by decoding these inputs. There are four REGION inputs available when the CY7C960 is configured as an IO controller and three REGION inputs when it is configured as a combination DRAM/IO controller. Refer to the Local Interface Description for details on these configurations. The IO controller configuration will be assumed for the purpose of this discussion.

The purpose for providing this large number of Regions is to allow a board to respond differently to the various VMEbus transactions that can be enabled. Each Region can have its own “personality.” For example some respond to block transfers, some do not. One or more of the 16 decoded Regions can be designated as the “unselected” regions. The way this is done is described in the next section.

The CY7C964s are configured to drive the VME address onto the local address bus. VMEbus addresses flow through and allow external decode logic to take advantage of the VMEbus buffering provided by the CY7C964 transceivers. In the case of A64 and A40 transactions, the CY7C960 provides two signals, DENIN* and DENIN1*, which can be used to instruct the external decoder circuitry when to sample the VMEbus data signals (available on the local data bus, buffered by the CY7C964 devices) for this multiplexed address information. Thus, the user may choose to implement a comparator externally of any complexity and drive the REGION inputs from this external circuit.

The amount of time the CY7C960 waits before sampling the REGION inputs is specified during initialization. The parameter is called Decode Delay. The CY7C960 starts counting the number of clocks that occur from the assertion of AS* by the VMEbus Master. When the number of clocks equal the Decode Delay, the CY7C960 samples the four REGION inputs to determine which one of 16 possible address regions the cycle is directed towards.

Decode delay starts with the sampling of DSB* instead of AS* in the case of multiplexed address cycles such as A64. This is to allow address information to arrive on the VME data bus.

If the minimum decode delay is specified, then the REGION inputs must be stable when AS* is sampled Low. If the maximum decode delay is specified, then the REGION inputs must be stable five clocks after AS* assertion. See *Figure 3−16* for an example of a Decode Delay of 3 clock periods.

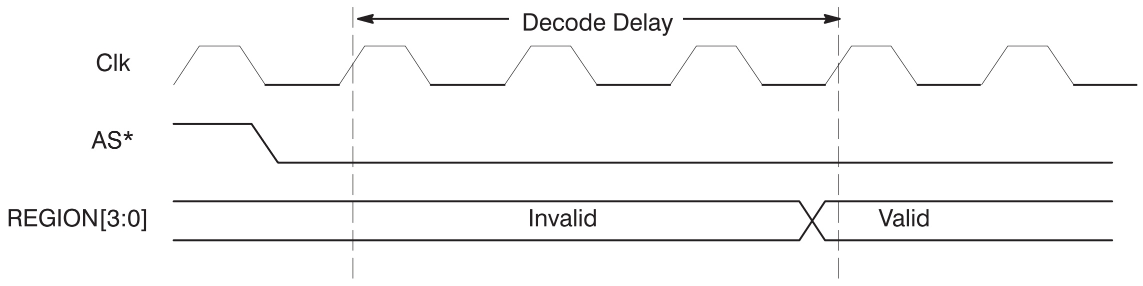

Figure 3−16. Decode Delay Example

3.5.3.1 AM/LA Multiplexing

The AM/LA Multiplexing feature of the CY7C960 is an extension to the Region Mapping capability described above and is enabled by setting a bit in the serial programming stream. This feature adds even more flexibility for mapping slave accesses to specific regions by providing VMEbus AM code information on LA[7:2] during the slave decode period of a VMEbus transaction. During this period the CY7C964s are driving upper address information on LA[31:8] and the CY7C960 is driving AM codes on LA[7:2]. With this feature enabled, external circuitry can be implemented to decode both address and AM code information in parallel during the decode delay period.

To demonstrate the power of this feature, consider a VME slave with one A24 CSR region and one A32 memory region with LOCK support. It is not sufficient to monitor just the VMEbus address lines because they may satisfy both the A24 and A32 decode conditions. For this application an external decoder must monitor the VMEbus AM lines in order to discriminate between A24 CSR and A32 LOCK cycles. This would require connecting AM[5:0] to both the CY7C960 and the external decoder, a violation of the VMEbus specification on signal loading. With the addition of local AM signalling, the CY7C960 and CY7C964s are the only load on the VMEbus and together they pass all available VMEbus addressing information to the local side of the interface for decode processing.

AM/LA multiplexing begins immediately after initialization. The address transceivers on the CY7C964s are enabled onto the upper bytes of the local address bus while the address transceivers on the CY7C960 are enabled onto the least significant byte. VMEbus addresses flow through the CY7C964s and VMEbus AM codes flow through the CY7C960 as shown in *Figure 3−17*.

The assertion of LADI is the event used to latch the local AM codes presented on LA[7:2] within an external decoder. The CY7C960 local address transceivers are disabled and the CY7C964 local address transceivers are enabled one clock after LADI assertion. The VMEbus signals A[7:1] and LWORD* will flow through to LA[7:0] of the least significant CY7C964 at this time. The CY7C960 then stores the values presented to it on LA[1] and LWORD* to complete the slave decode process. These values are used to compute the correct data byte enable pattern during the data phase of the access.

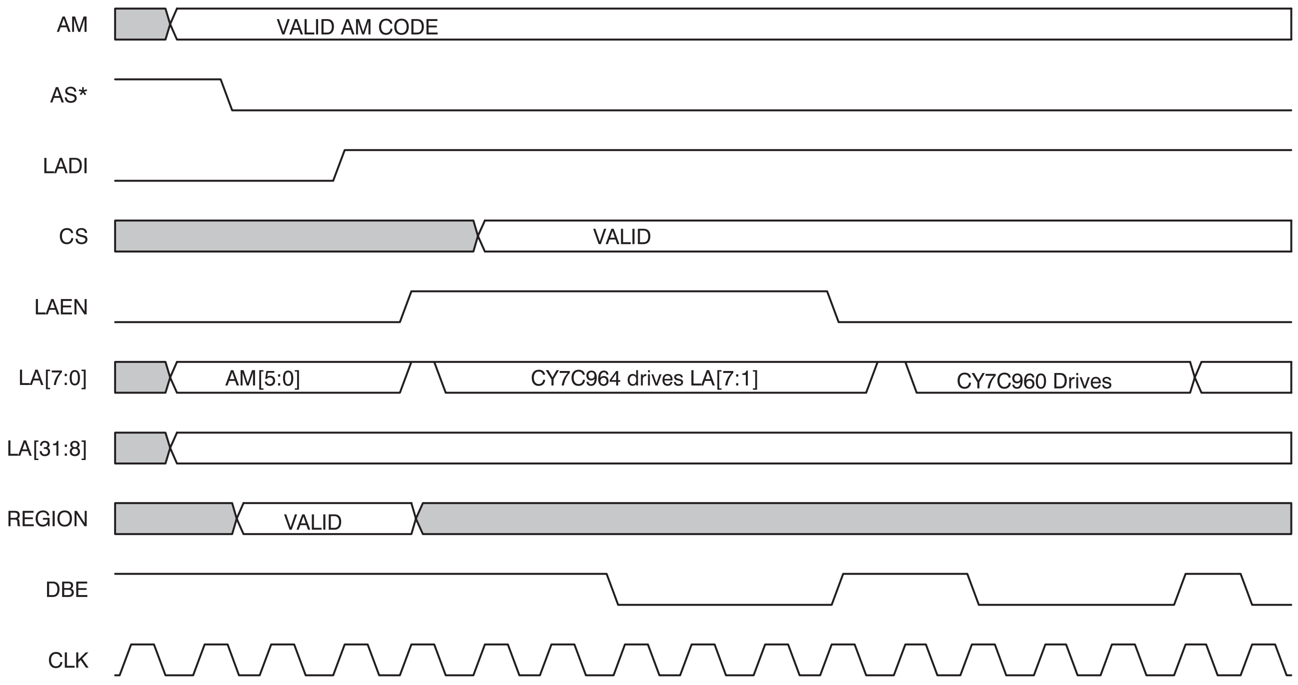

Figure 3−17. AM/LA Multiplexing with Decode Delay = 2T

Figure 3−18 shows the effect of introducing Decode Delay to the address phase of a slave access when AM/LA Multiplexing is enabled. The timing of AS* sampling, LADI assertion, and local address bus transition from AM[5:0] to LA[7:1] and LWORD* remain the same as in *Figure 3−17* while the duration of LA[7:1], LWORD*, and the time to valid chip selects will increase according the number of Decode Delay clocks programmed at initialization.

Figure 3−19 represents the same slave access as in *Figure 3−17* except that AM/LA multiplexing is turned off. In this case, all four CY7C964s are enabled onto the local bus while the CY7C960 is waiting to be accessed by a Master. The CY7C960 will not drive the local address bus until the second and subsequent beats of a Block Transfer.

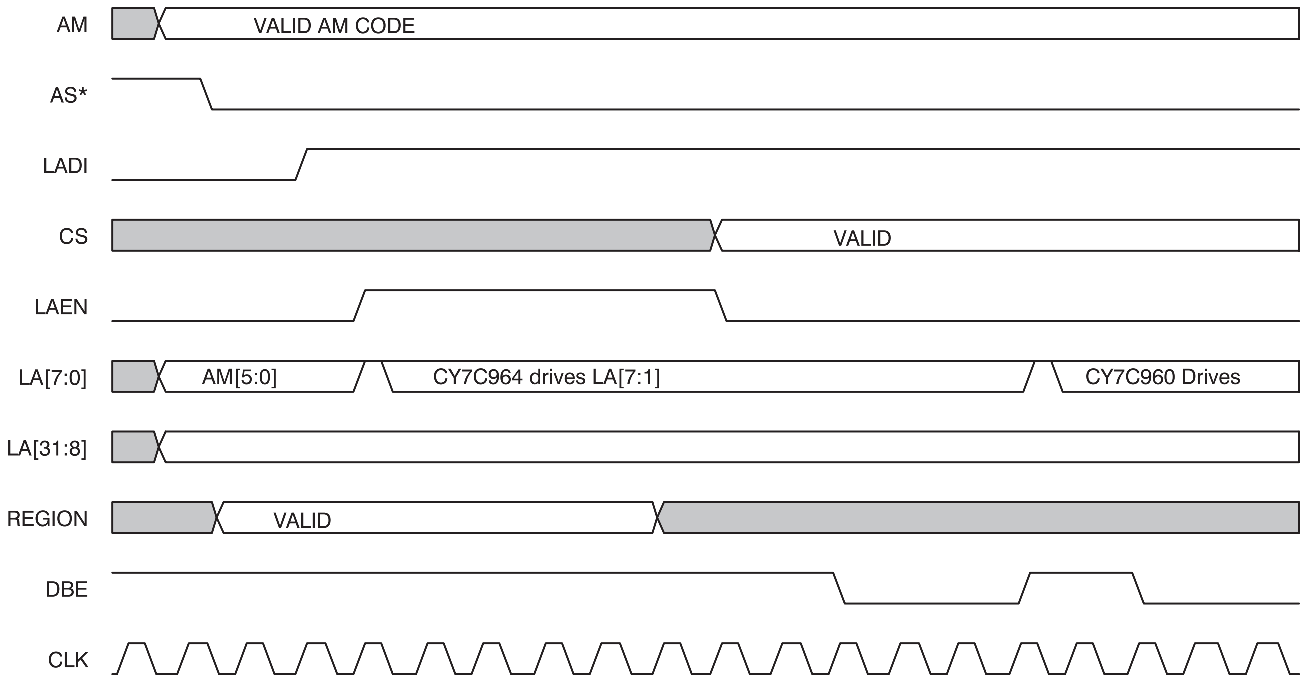

Figure 3−18. AM/LA Multiplexing with Decode Delay = 5T

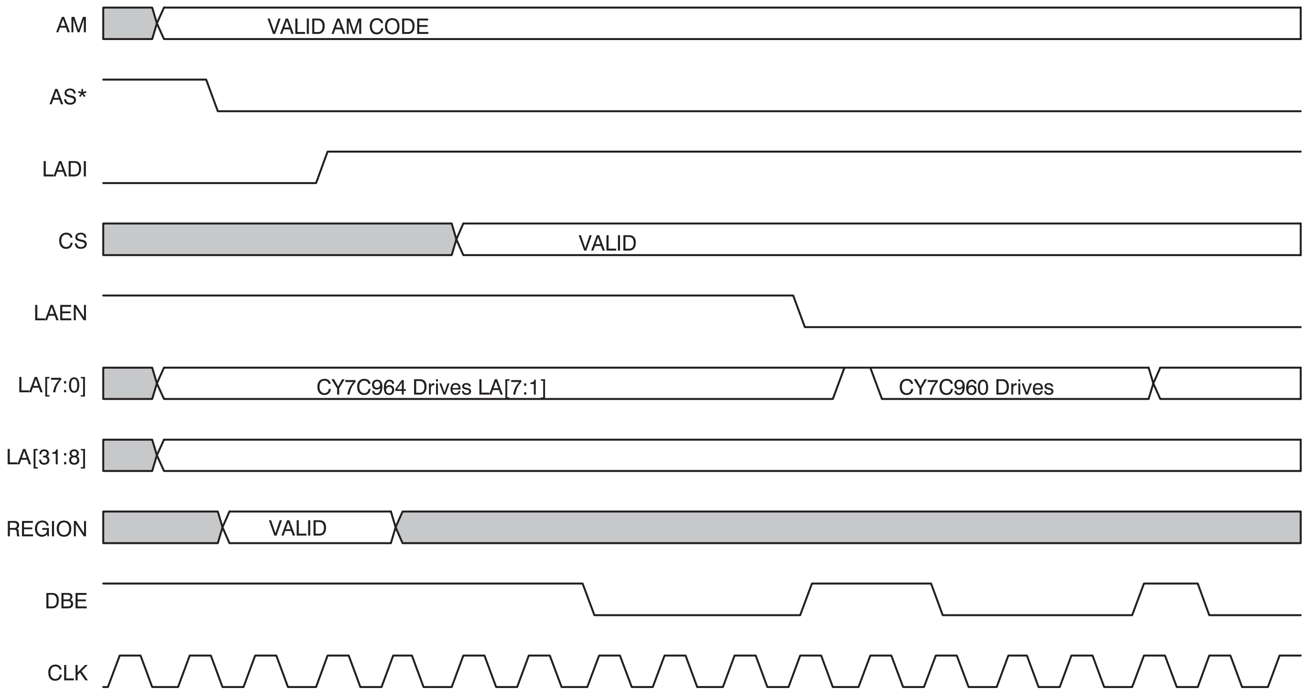

Figure 3−19. AM/LA Multiplexing Disabled with Decode Delay = 2T

3.5.4 Bus Holdoff

The Bus Holdoff feature of the CY7C960 provides the ability to three-state the slave interface chip set (CY7C960 and CY7C964s) and suspend slave decode activity in order to easily dual-port devices mapped as VMEbus slave resources. This feature also extends the functionality of the CY7C961 by allowing local configuration of its master DMA channel. The feature is enabled by setting a bit in the serial programming stream.

The mechanism for Bus Holdoff is overlaid on the CY7C960 LACK pin. When Bus Holdoff is enabled, LACK acquires a local bus grant function in addition to its previously described local transaction acknowledge function. By controlling LACK deassertion, it is possible to suspend all local response of the slave interface chip set, effectively delaying the next VMEbus slave access until LACK is reasserted.

While the CY7C960 is in its holdoff state, output signals DBE, RW, CS, and LA are three-state, and control signals DENIN[1:0]* and LAEN three-state the LD and LA pins of the least significant CY7C964 device. LAEN on the most significant CY7C964s must also be externally controlled during Bus Holdoff. This holdoff configuration is intended to allow simple design of address, data, and control multiplexing to achieve dual-ported local memory architectures.

The CY7C960 Bus Holdoff feature conceptually allows cycle stealing by other local bus masters. This is accomplished by using an output of the CY7C960, the LADI signal, as an interface busy signal. A local bus arbiter can monitor LADI and deassert LACK within two clocks after a deassertion edge on LADI. LADI deassertion represents the end of a VMEbus transaction regardless of whether it was decoded by the CY7C960 interface. If LACK is deasserted in the bus holdoff window, the local bus is three-state until LACK is again asserted. If LACK is deasserted outside the holdoff window, LACK performs its local acknowledge function.

3.5.4.1 Transaction Type Detection

There is another internal decoder: the CY7C960 decodes AM[5:0] and forms an internal Address Modifier Decode Word after the same Decode Delay interval that is used for sampling the REGION inputs. As the VMEbus specification dictates that AM Codes be stable prior to the assertion of AS* then the Decode Delay is unnecessary, but is used for convenience.

Two fields result from decoding the AM Codes: Address Size and Modifiers. *Table 3−4* provides a definition of the CY7C960's classification of AM Codes.

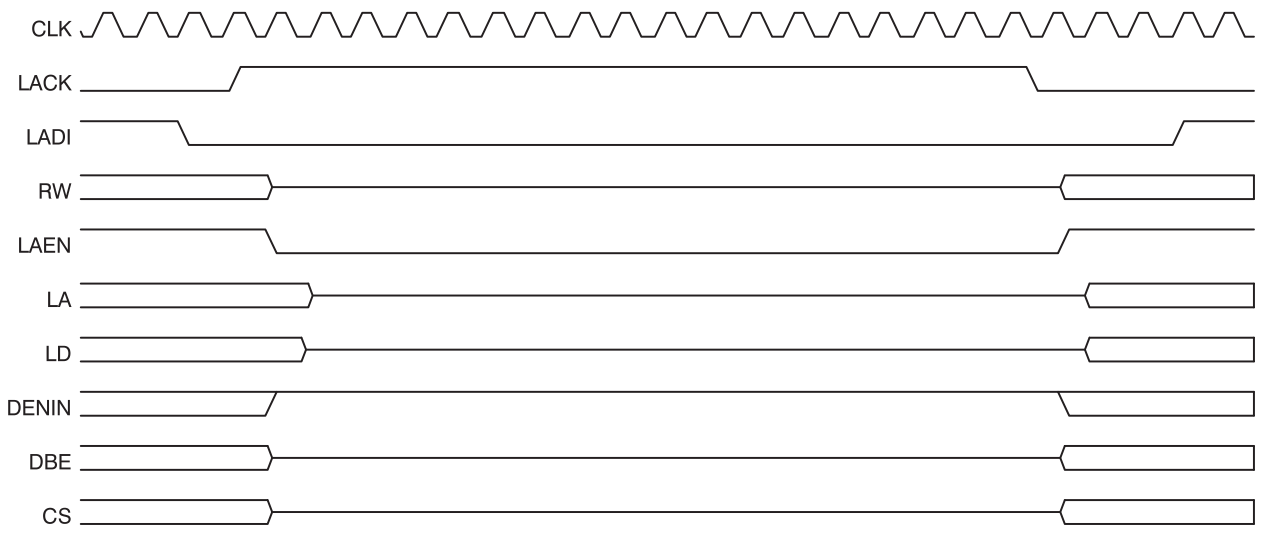

Figure 3−20. Local Bus Holdoff

Table 3−4. AM Code Decoder

Address Size	Modifier	VMEbus Transactions
A64		A64 block transfer (BLT); A64 single cycle access; A64 64-bit block transfer (MBLT); A64 lock command (LCK); A64 serial access
A40		A40 block transfer (BLT); A40 single cycle access; A40 lock command (LCK); A40 serial access
A32		A32 supervisory block transfer (BLT); A32 supervisory program access; A32 supervisory data access; A32 supervisory 64-bit block transfer (MBLT); A32 non-privileged block transfer (BLT); A32 non-privileged program access; A32 non-privileged data access; A32 non-privileged 64-bit block transfer (MBLT); A32 lock command (LCK); A32 serial access
A24		A24 supervisory block transfer (BLT); A24 supervisory program access; A24 supervisory data access; A24 supervisory 64-bit block transfer (MBLT); A24 non-privileged block transfer (BLT); A24 non-privileged program access; A24 non-privileged data access; A24 non-privileged 64-bit block transfer (MBLT); A24 lock command (LCK); A24 serial access
A16		A16 supervisory access; A16 non privileged access; A16 lock command (LCK); A16 serial access
	SUPER	Supervisory access
	NPRIV	Non-Privileged access
	PROG	Program access

Table 3−4. AM Code Decoder (continued)

Address Size	Modifier	VMEbus Transactions
	DATA	Data access
	BLT	Block transfer (inc. MBLT)
	LONG	32 or 64 bit access (not 8 or 16)
	LOCK	A64 lock command (LCK), A40 lock command (LCK), A32 lock command (LCK), A24 lock command (LCK), A16 lock command (LCK)
	SERIAL	A64 serial, A40 serial, A32 serial, A24 serial, A16 serial
	CR/CSR	Configuration ROM/Control and Status Register
	U1, U2	User Defined (U1 includes AM Codes 18 − 1F, U2 includes 10 − 17)

It should be noted that the parameter LONG is derived from a combination of DSi*, LWORD*, and AM Code. For more information on AM Code definitions, it is recommended that the VMEbus Specification be consulted.

The reason that the CY7C960 decodes the AM Code this way is to allow the user to enable VMEbus transactions selectively by AM Code classification. For example, a transaction with AM Code “0C” would result in the following settings: A32, SUPER, BLT, LONG. The CY7C960 would react to the transaction only if all four categories were enabled for the addressed Region. The CY7C960 consults a look-up table, configured during initialization, to determine whether the transaction is enabled or not. *Table 3−5* is an example of how this table could be programmed.

Table 3−5. Major AM-to-Region Mapping Example

REGION	A64	A40	A32	A24	A16	SUPER	NPRIV	PROG	DATA	BLT	LONG
0	✓					✓				✓	✓
1		✓				✓				✓	✓
2			✓			✓		✓			✓
3				✓			✓		✓		✓
4				✓	✓	✓		✓			
5			✓			✓	✓	✓	✓	✓	✓
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											

If the CY7C960 is configured according the table above, then it will respond as slave under the following conditions.

If the REGION inputs are “0000”, then the CY7C960 will respond to A64 block transfers of type BLT or MBLT.

If the REGION inputs are “0001”, then the CY7C960 will respond to A40 block transfers of type D8:BLT, D16:BLT, or MD32:BLT.

If the REGION inputs are “0010”, then the CY7C960 will respond to A32 supervisory program accesses of type D8, D16, or D32 (single cycle).

If the REGION inputs are “0011”, then the CY7C960 will respond to A24 non-privileged data accesses of type D8, D16, or D32 (single cycle).

If the REGION inputs are “0100”, then the CY7C960 will respond to A24 supervisory program, and A16 supervisory accesses of type D8 or D16 (single cycle). Note that D32s will not be responded to in this region because LONG is not set.

If the REGION inputs are “0101”, then the CY7C960 will respond to any A32 AM Code other than A32 LOCK and A32 SERIAL which must be enabled separately (see below).

If the REGION inputs are any other value, then the CY7C960 will assume that the cycle is intended for another board and will not respond.

LOCK, SERIAL, USER, and CR/CSR AM codes are treated somewhat differently from the “mainstream” codes. Each of these codes can be assigned to only a single specific REGION. Since these transactions are typically mapped to a single address space on a board, it was not necessary to include them in all 16 rows of the main REGION/AM decode lookup table. The user simply selects which, if any, of these transactions should be decoded and assigns a valid REGION pattern to them.

Table 3−6 shows an example: A32 LOCK transactions are enabled in Region 0; A40 and A16 SERIAL transactions are enabled in Region 8; User AM codes in group 1 are enabled in Region 10; the remaining two classifications are disabled (but had they been enabled they would have been enabled for Region 2).

Table 3−6. Minor AM-to-Region Mapping Example

AM Classification	A64	A40	A32	A24	A16	Enable	Region
LOCK			✓			✓	0
SERIAL		✓			✓	✓	8
USER1 (AM Codes 18 − 1F)	These do not have an associated address size					✓	10
USER2 (AM Codes 10 − 17)							2
CR/CSR							2

If the VMEbus transaction is determined by the above process to be enabled for the Region being addressed, then the CY7C960 internal flag DECODE is set, allowing local activity to commence. *Figure 3−21* shows a flowchart representation of the process of decoding a VMEbus Transaction.

A slave board that uses the CY7C960 can be programmed to respond in a wide variety of ways with a fine control on decode granularity by using the CY7C960's Region Mapping features. The following sections and the AC timing diagrams at the end of this document are intended for users who need to know further details of the features described above.

3.5.5 Decode Delay Timing

Slave decode delay timing is a process that conceptually begins whenever AS* is sampled low on the rising edge of CLK. However, at this time the CY7C960 slave decoder must wait for all local activity to clear before processing a VMEbus cycle. This is because it is possible for a local cycle to be in progress without a Master on the VMEbus.

For example, in the case of a posted slave write, the VMEbus Master is allowed to leave the bus with data still to be processed in the interface. If this is the case, then the local address bus will be holding an address from this posted write. Likewise, there are situations where the CY7C960 will perform a read-ahead cycle. This happens when block transfer reads do not terminate on a 256-byte boundary. The on-board slave decoder must wait for this address to clear and allow enough time for the new address to arrive and become stable on the local bus.

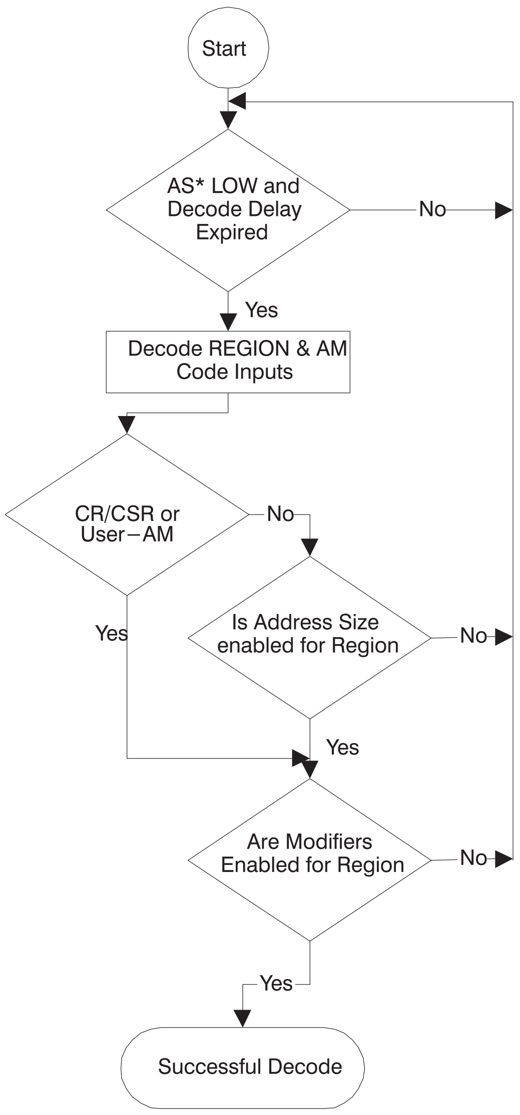
Start
AS* LOW and Decode Delay Expired
No
Yes
Decode REGION & AM Code Inputs
CR/CSR or User−AM
No
Yes
Is Address Size enabled for Region
No
Yes
Are Modifiers Enabled for Region
No
Yes
Successful Decode

Figure 3−21. Flowchart for Decoding VMEbus Transactions

The internal AM decoder then signals whether the first cycle will be an address broadcast cycle. Since MBLT and MD32 transactions use the address bus for data transfers, a Master initiating these transactions will first issue an address broadcast so the Slave can capture the starting address before the address bus changes into a data transfer bus. This address broadcast cycle also occurs for transactions that use the VME data bus as an address bus. These cycles include all A64 and A40 transactions since the width of the VME address bus is only 32 bits (24 bits for 3U boards).

If there will be no address broadcast cycle, then the decode delay timeout begins with AS*. An internal timer counts up to the programmed decode delay. If an address broadcast cycle is signalled, then the CY7C960 will wait for the DSB* event and then count up to the programmed decode delay.

In either case, after the decode delay has expired, the CY7C960 will determine if the addressing information for this cycle falls with its programmed address space. If it does, then an internal flag signals other internal circuitry that the cycle is a valid slave decode for the board. If it does not, then the internal slave decoder will wait until AS* is sampled High and then return to an idling state waiting for the next VMEbus cycle.

3.5.6 Slave Addressing Before Initialization

The CY7C960 can be programmed in one of three ways. The interface can be programmed from the VMEbus, the local bus, or a combination of the VME and local buses. If the CY7C960 is programmed using the VMEbus or the combination method, slave address information and device programming is loaded using a methodology compliant with the VMEbus Auto Slot ID specification using either a CR/CSR or A16 AM code. The configuration ROM/control and status register (CR/CSR) capability described by the latest revision of the VMEbus specification provide a mechanism for board identification and automatic initialization. Auto Slot ID is an optional method of assigning the CR/CSR base address to each VMEbus board. This Auto ID slave uses a level 2, D8 interrupter along with additional board specific hardware to obtain the CR/CSR base address. An Auto ID Master, called the Monarch, uses a level 2, D8 Interrupt Handler to acknowledge the Auto ID interrupt and assigns a base address to the slave board. Specifically, this Auto ID slave:

1. generates a level 2 interrupt if PDATA is sampled Low after power-on or a negative edge on SYSRESET*. (IRQ* must be connected to IRQ[2]* on the VMEbus.)
2. waits for and responds to the level 2 IACK cycle initiated by the Monarch.
3. asserts LDEN* thereby enabling a Status/ID byte that is sourced by external local hardware.
4. presents the Status/ID byte on the VMEbus. (all 32 bits of VMEbus data are driven)
5. drives DTACK* Low and releases IRQ*.

The Monarch must then:

1. write the base address of this board using a single cycle write with a CR/CSR or A16 AM code. If a CR/CSR AM code is used, then all subsequent programming cycles must use

a CR/CSR AM code. Likewise, if an A16 AM code is used, then all subsequent programming cycles must use an A16 AM code. It is mandatory that this access be indivisible with the previous IACK cycle on the backplane to avoid collisions with

2. program the CY7C960 internal registers using a series of single-cycle writes to the base address of the slave if the VMEbus Initialization Method is used. These accesses require a bit pattern of 'xx00' on REGION[2:0] if a CR/CSR AM code is used and 'xxx0' if an A16 AM code is used.

3. initialize the CY7C964 compare and mask registers using two single writes to the base address of the slave. These two writes will reassign the CR/CSR base address of the slave to its new base address in the system.

3.5.7 Address and Data Strobe Event Processing

Another process begins in parallel with slave decoding. The following sequence of events occurs whenever AS* is sampled Low by the CY7C960:

- The internal AM decoder signals whether the first cycle will be an address broadcast cycle. Since MBLT and MD32 transactions use the address bus for data transfers, a Master initiating these transactions will first issue an address broadcast cycle so the Slave can capture the starting address before the address bus changes into a data transfer bus. This address broadcast cycle also occurs for transactions that use the VME data bus as an address bus. These cycles include all A64 and A40 transactions since the width of the VME address bus is only 32 bits.
- The value of LWORD* is latched and held for the duration of the transaction. LWORD* on the VMEbus must be connected to A[0] on the least significant CY7C964 because LWORD* is captured by the CY7C960 on LA[0]. This does not present a problem because A[0] on the VMEbus is encoded on the VMEbus data strobes.
- The local address that has been flowing through the CY7C964s will now be latched. An important consideration to note here is that the transaction may be intended for another slave on the backplane and that slave may respond very quickly on its DTACK* or BERR* lines. Since the Master is permitted to remove the address after the responding Slave drives DTACK* or BERR* Low, the CY7C960 will signal the CY7C964 to latch its local address lines at this time in order to capture the addressing information before it can be removed. This is accomplished by the assertion of LADI from the CY7C960 to the CY7C964.

The CY7C960 then waits for the DSB* and DECODE events. The DSB* event is defined by the VMEbus specification, which states that maximum skew between the assertions DS1* and DS0* Low by a Master shall be no more than 20 ns at the Slave. When the CY7C960 samples either DS1* or DS0* Low by CLK, it waits two clocks and then sets an internal flag called DSB*. This two clock waiting period is designed to accommodate any data strobe skew within specification by ensuring that both strobes have had time to arrive before the device decodes addressing information from the strobes.

The DECODE event is another internal flag that is set 'true' or 'false' depending on whether the CY7C960 has been selected. Refer to the previous sections on slave address decoding for a description of this process.

Several things happen when these two events occur. Data size information is decoded from the values on DS1*, DS0*, LA[1], and LWORD*. This information is used to determine local byte enable patterns (DBE[3:0]), data byte swapping control (SWDEN*), and local address counting. The CY7C960 also disables the drivers on the least significant CY7C964 and begins to drive the starting address for the transaction on LA[7:0].

If either of these two events fail to occur, then the CY7C960 will wait until it samples AS* High, deassert LADI thereby letting VME address flow through to local address on the CY7C964's, and return to an idling state waiting for the next AS* Low event.

3.5.8 Slave Data Transfer Acknowledgement

Slave data transfers are acknowledged by a Low assertion of the DTACK* output. All assertions of DTACK* require the assertion of the internal DECODE and DSB* flags. If these two conditions are present, then DTACK* will be asserted Low on the backplane when

- the internal AM decoder signals that an address broadcast cycle is in progress. Address broadcast cycles do not generate local chip selects.
- the cycle is a slave read of any type. The CY7C960 will assert DTACK* Low when its local controller sets an internal flag that indicates data has been read from the local board and the data is being held in the interface (CY7C964s).
- the cycle is a slave block transfer write of any type. This is the case where data is posted in the interface and the VMEbus Master is allowed to start the next cycle or terminate the transaction. While the Master is starting the next cycle or terminating, the CY7C960

local controller writes the posted data to the local board. Slave write posting of block transfer cycles allow the CY7C960 to transfer data at a maximum rate of 80 Mbytes/s for MBLT bursts and 40 Mbytes/s for BLT bursts.

- the cycle is a single-cycle slave write of any type and the CY7C960 local controller has finished writing the data to the local board. This is the case where data is not posted in the interface and the VMEbus Master must wait for the write transfer to complete before starting the next cycle or terminating.

3.5.9 Slave Write Posting

Slave write posting is a means of obtaining maximum data transfer performance by decoupling the VMEbus from the local board. Every slave write data transaction is latched in the CY7C964s for further processing by the CY7C960 local controller and the VMEbus Master is allowed to begin its next cycle or terminate the transaction.

In other words, the VMEbus Master drives DSi*, and the CY7C960 drives DTACK* before the data has reached the local bus.

Data is latched in the interface when DSB* occurs during a slave write to a valid address region. At this time the CY7C960's local controller initiates a write cycle to the local board. This pipelined architecture allows fast VMEbus cycles that are not delayed by local affects such as DRAM refresh bursts, or resource contention. When the local cycle is complete, the data latches are opened and new data is allowed to flow into the interface. Also, LA[7:0] is incremented according to the data width of the transaction after the local board acknowledges that the posted data has been written.

3.5.10 Slave Read-Ahead Cycles

Slave read-ahead cycles are also used to achieve maximum data transfer rates for block transfer reads. When the CY7C960 completes a read cycle to a VMEbus Master during a block transfer operation, it will read ahead on the local bus in order to have data ready for the next cycle of the burst.

Data is read from the local board, latched in the interface, and driven on the VMEbus data lines when DSB* arrives to start a slave block transfer read from a valid address region.

Note that all 32 bits of data are driven on the VMEbus data lines, regardless of data size. The CY7C960 asserts DTACK* Low to signal that the data is ready to be taken. When the VMEbus Master releases the data strobes, the CY7C960 stops driving the VMEbus data lines and LA[7:0] are incremented according to the data width of the transaction. The CY7C960 then performs another local read in preparation for the next cycle of the read burst.

Since there is no indication to the slave as to how long a block transfer will be or when it will terminate, it is possible to have a read-ahead cycle occur that does not get transferred to the Master. In other words, the Master could terminate before, during, or after the read-ahead and the local data that was read would not be transferred to the VMEbus. However, the CY7C960 will not read ahead at a 256 byte boundary even if 2k byte boundaries are in effect as is the case for some MBLT transactions. The read circuitry was implemented this way in order to provide a means of averting read-ahead cycles and is possible because the CY7C960 drives LA[7:0], which enables it to detect when the address is about to overflow.

Also, as in the case of MBLT and MD32 transactions, there can be two local accesses to be processed for one VMEbus access. The CY7C960 handles all the pipelining and multiplexing control that must take place in the interface to accommodate reads or writes when the VME data bus is wider than the local data bus.

3.5.11 Interrupt Cycle Support

There are six pins associated with interrupt requesting on the VMEbus: IRQ*, IACK*, IACKIN*, IACKOUT*, LDEN*, and LIRQ*. There are two ways an interrupt can be signalled on the VMEbus backplane: a local interrupt request or a reset.

A power-on, system reset, or local reset condition causes the device to implement its initialization protocol. Refer to sections 3.4.3 and 3.4.5 for details on initialization.

LIRQ* is the input to the device from local circuitry demanding an interrupt cycle. If the LIRQ* signal is asserted Low, the CY7C960's IRQ* pin is asserted Low. The local circuitry should remove LIRQ* after the interrupt has been acknowledged: The CY7C960 simply drives the state of LIRQ* onto IRQ*. *Figure 3−22* illustrates the relationship between LIRQ* and IRQ*. s60 represents the assert and deassert delays.

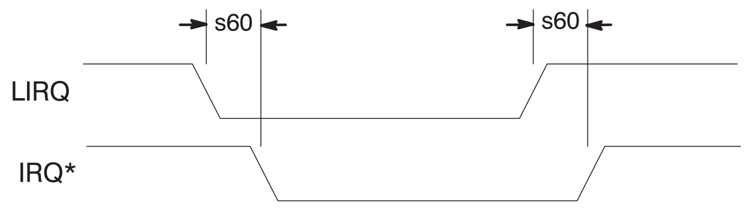

Figure 3−22. Interrupt Signal Timing

The CY7C960 actively drives the value of IACKIN* to IACKOUT* in conformance with the rules for VMEbus D8 Interrupt Acknowledge Cycles. If it is driving IRQ* and IACK* is sampled Low and IACKIN* is sampled Low and AS* is sampled Low, then LA[3:1] is compared with the programmed interrupt request level. If they are not equal then the Low IACKIN* is driven to IACKOUT*. If they are equal, then IACKOUT* remains High and LDEN* is driven Low to enable an external Status/ID vector onto the local data bus. This vector is in turn enabled onto the backplane and the VMEbus Interrupt Handler is acknowledged by a Low transition on the DTACK* line.

It is important to note that, while the CY7C960 is a D8 interrupter, it drives all 32 bits of VMEbus data on the backplane with the assertion of DENO* to the CY7C964s. If the user wishes to provide only 8 bits of status ID, then the value of the upper bytes should be set to FF hex to ensure full compliance with the VMEbus specification. If the user wishes to provide 16 or 32 bits of status ID, then the interrupt handler should be programmed to handle the provided width. The CY7C960 does not attempt to differentiate between 8-, 16-, or 32-bit interrupt acknowledge cycles.

3.5.12 Interrupt Handshake Support

The CY7C960 can be programmed at initialization to wait for a local acknowledge signal before terminating an IACK cycle. LACK is the local signal used to suspend completion of an IACK cycle until the local interrupter is ready to present a STATUS/ID vector to the VMEbus Interrupt Handler.

If the Handshake bit is set the CY7C960 will sample the state of the LACK pin three clocks after it samples a Low level on its IACKIN* pin during a VMEbus IACK cycle. If LACK is High at this time, then the CY7C960 will wait until LACK is Low before terminating the cycle with a falling edge on DTACK*.

If the Handshake bit is not set, the CY7C960 will ignore the state of the LACK pin and DTACK* the VMEbus Interrupt Handler four clocks after sampling a valid IACKIN*.

Handshake support is available immediately after local completion of the Combination Initialization Method or after full completion of the VMEbus or Local Initialization Methods. Refer to Chapter 3.4 for details on the three programming methods.