3.1 Introduction

3.1.1 Feature List

Optimal Performance:	80 Mbyte per second Block Transfer Rates
Next-Generation Product:	VME64 transactions, including A64/D64, A40/MD32 transfers, Auto Slot ID, CR/CSR space, LOCK cycles etc.
Backwards Compatible:	All standard VMEbus transactions implemented VMEbus Interrupter
Simple to Use:	No Local CPU required Programmable from VMEbus or serial PROM
Highly Integrated:	DRAM controller, including refresh timers, Local I/O controller,
Innovative Architecture:	Flexible VMEbus address scheme User-configured VMEbus Personality
Ultra-Small footprint:	TQFP Packaging, and other options.

3.1.2 Family Overview

The CY7C960 and the CY7C961 are members of Cypress's industry-standard range of VMEbus controllers. Although they are each low cost, they are highly flexible, and designed to meet the needs of high performance VMEbus board developers. The CY7C960 is intended for applications whose primary requirement is board space savings—it offers Slave-only features. The CY7C961 provides Master support in addition to the Slave features of the '960, and hence occupies a little more board space. Each device has been optimized for VMEbus performance, supporting the full 80-Mbyte/sec transfer rate of the VME64 specification.

The other members of the family are VIC068A (the industry's most popular 32-bit VMEbus interface controller), VIC64 (the 64-bit version of the VIC068A), VAC068A (VMEbus address controller), and CY7C964 (a useful companion for either VIC068A, VIC64, CY7C960, or CY7C961).

This Section of the book first describes the CY7C960, then extends that description to the CY7C961. It is necessary to understand the operation of the CY7C960 before reading the CY7C961 sections.

3.1.3 CY7C960 Architectural Overview

The CY7C960 Slave VMEbus Interface Controller provides the board designer with an integrated, full-featured VME64 interface. This 64 pin device can be programmed to handle every transaction defined in the VME64 specification. The CY7C960 contains all the circuitry needed to control large DRAM arrays and local I/O circuitry without the intervention of a local CPU. There are no registers to read or write, no complex command blocks to be constructed in memory. The CY7C960 simply fetches its own configuration parameters during the power-on reset period. After reset the CY7C960 responds appropriately to VMEbus activity and controls local circuitry transparently.

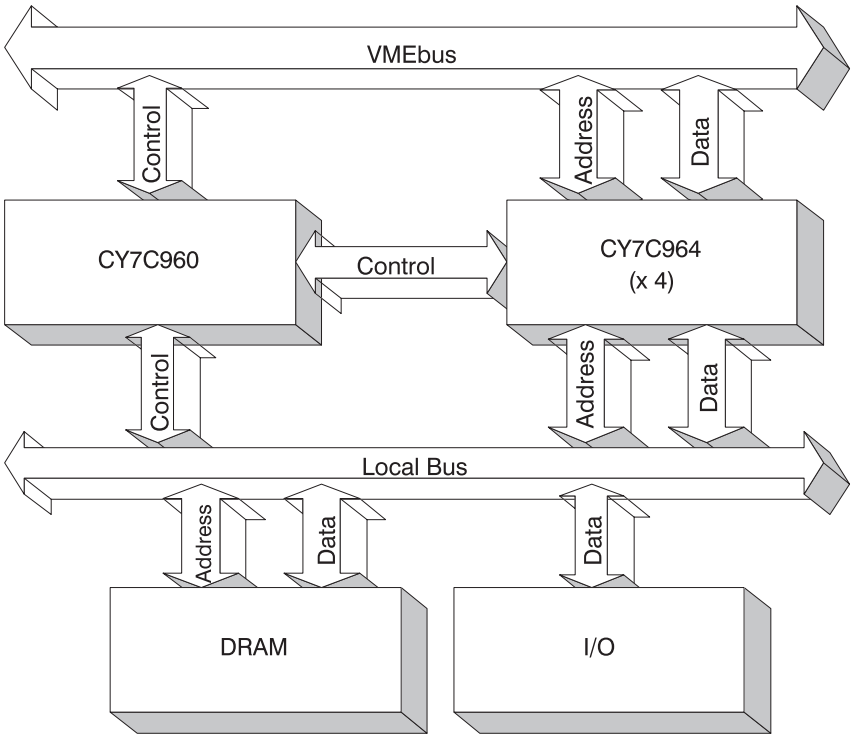

Figure 3–1. Typical System Block Diagram

The CY7C960 controls a bridge between the VMEbus and the local DRAM and I/O. Once programmed, the CY7C960 provides activities such as DRAM refresh and local I/O handshaking in a manner which requires no additional local circuitry. The VMEbus control signals are connected directly to CY7C960. The VMEbus address and data signals are connected to companion address/data transceivers which are controlled by CY7C960. The CY7C964 VMEbus Interface Logic Circuit is an ideal companion device: the CY7C964 provides a slice of data and address logic that has been optimized for VME64 transactions. In addition to providing the specified drive strength and timing for VME64 transactions, the CY7C964 contains all the circuitry needed to multiplex the address/data bus for multiplexed VMEbus transactions. It contains counters and latches needed during block transfer (BLT) operations. And it also contains address comparators which can be used in the board's Slave Address Decoder. For a 6U or 9U application, four CY7C964 devices are controlled by a single CY7C960. For 3U applications, the CY7C960 controls two CY7C964 devices and an address latch.

If the application does not need VME64 transactions, then the user may choose to implement the companion logic using Cypress's inexpensive FCT family of interface devices. The CY7C960 also provides all of the timing and control signals needed for this application also. (See Chapter 3.7.)

The design of the CY7C960 makes it unnecessary to know the details of the VMEbus transaction timing and protocol. The complex VMEbus activities are translated by the CY7C960 to be simple local cycles involving a few familiar control signals. Similarly, it is not necessary to understand the operation of the companion device, CY7C964: all control sequences for the part are generated automatically by the CY7C960 in response to VMEbus or local activity. If more information is desired, consult Section 4, The CY7C964 Bus Interface Logic Circuit.

VMEbus Transactions supported by CY7C960 include all transactions defined by the VME64 specification. CY7C960 functions as a VMEbus Interrupter, and supports the new Auto Slot ID standard and CR/CSR space. CY7C960 also handles LOCK cycles, although full LOCK support is not possible within the constraints of the CY7C960 pinout. (For full LOCK support see the CY7C961 description.)

On the local side, no CPU is needed to program the CY7C960, nor to manage transactions. All programmable parameters are initialized through the use of either the VMEbus, or a serial PROM. As the CY7C960 incorporates a reliable power-on reset circuit, parameters

are self-loaded by the device at power-up or after a system reset. If the VMEbus is used to provide parameters, a VMEbus Master provides the programming information using a protocol, described in a later section, which is compliant with the Auto Slot ID protocol from the VME64 specification.

Figure 3−2 shows the internal blocks that comprise the CY7C960. The architecture includes several functions that remove most of the VMEbus problems from the board designer's shoulders. All VMEbus control and response is automatic: the user loads the Region/AM table during configuration, and the CY7C960 then handles all appropriate VMEbus transactions. The CY7C964 controller works in lock step with the VMEbus Control Interface, providing the correct timing and control for the transaction in process. Local circuitry such as DRAM or I/O is simplified by the Refresh Controller, the DRAM Controller, and the Output Pattern Table. Block transfers are supported by the Local Address Controller together with the CY7C964 circuitry. Local timing is determined during configuration, and handshaking is available from the Data Byte Enable Controller. Local Interrupts are sup-

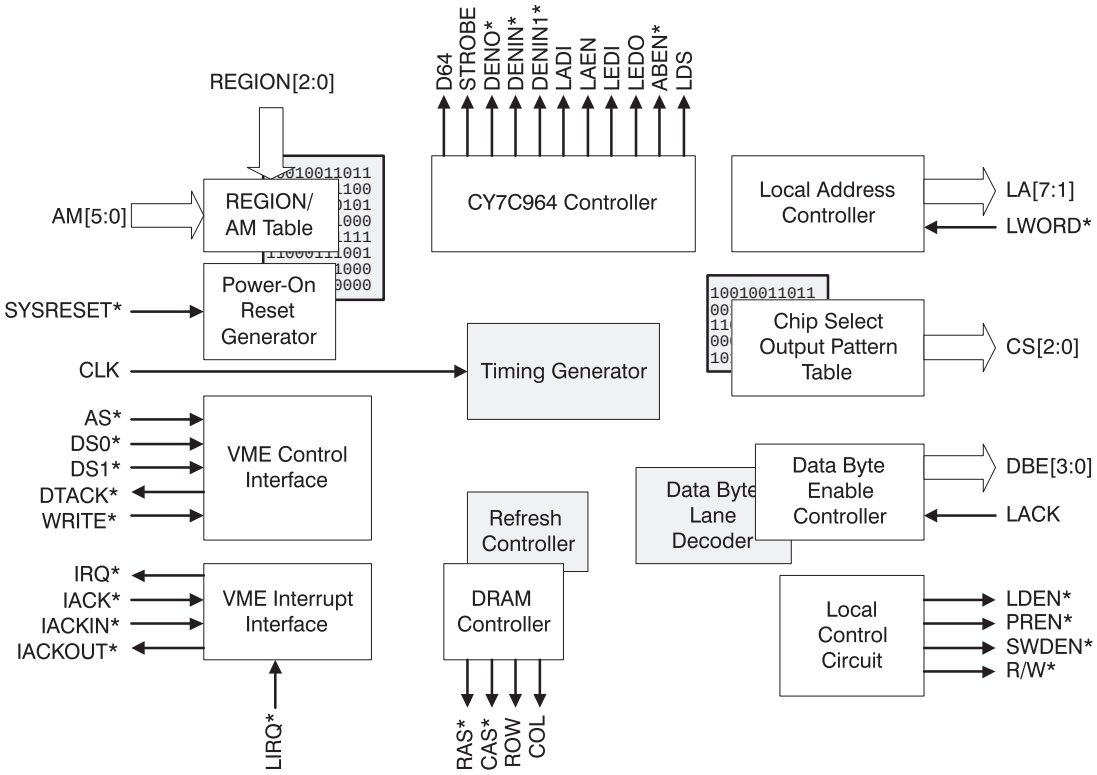

Figure 3−2. CY7C960 Block Diagram

ported through the VME Interrupt Interface. The CY7C960 contains an internal Power-on Reset circuit, and responds also to a VMEbus SYSRESET*.

To keep the size of the package as small as possible, several signal pins carry multiplexed signals, and other pins have functions that are programmable.

The Pin Description section, and other parts of this document, provide full information on the definition and use of these multiplexed signals.

3.1.4 Key Concepts

3.1.4.1 Local Bus Concepts

The CY7C960 has two modes of operation. The mode is selected during initialization. The user has a choice of DRAM/IO or I/O mode. If DRAM/IO mode is selected, then three Chip Select Outputs are provided along with the standard DRAM control signals. In I/O mode, six Chip Select Outputs are provided.

The CY7C960 introduces the concept of Region Mapping. This is a flexible method for selectively enabling VMEbus transactions to map to DRAM or I/O space. In DRAM/IO mode up to 8 Regions can be selectively mapped: in I/O mode up to 16 Regions are mappable. For example in DRAM/IO mode, Region 0 transfers could provide DRAM transactions while a transfer to Region 2 could disable DRAM, but provide Chip Select outputs.

The CY7C960 Chip Select Outputs are highly programmable: for each Region a different pattern can be driven from the available Chip Select Outputs. This allows external expansion if the three (six in I/O mode) outputs are insufficient. Additionally, the polarity of each chip select is programmable.

The VMEbus transfer mechanism allows for D8, D16, D32, D32UAT (Unaligned Transfer), MD32 (multiplexed), or D64 data transfers. The CY7C960 timing and control assume the use of the companion part CY7C964 that provides multiplexing control for the D64 (and for MD32) transfers. The CY7C964s are intended to connect to 32- or 16-bit local data buses: the CY7C960 provides appropriate signals to control byte-lane switching, pipelining and multiplexing as needed for the particular transaction in progress.

Four signals are provided for data byte enabling: DBE[3:0]. The CY7C960 controls these outputs as appropriate for the VMEbus transaction in progress. For example, in a three byte

unaligned transfer on the VMEbus the three correct DBE pins go active. In a VMEbus transaction that has been mapped to access local DRAM; RAS*, CAS*, ROW and COL go active in the normal manner for a DRAM cycle, plus the appropriate DBE pins for the data size being transferred. In a transaction that has mapped to I/O space; the Chip Select patterns plus the appropriate DBE pins are driven.

VMEbus 8- and 16-bit transfers are carried on D[15:0]. To allow the use of 32-bit memory, these bytes need to be swapped onto LD[31:16] for those addresses with LA[1] = 0. CY7C960 drives SWDEN* at the appropriate time to facilitate this byte swapping.

The on-chip DRAM Refresh Controller automatically provides bursts of 4 CAS*-Before-RAS* refresh cycles at the appropriate times, in a manner that does not interfere with any ongoing VMEbus or local transaction. No external intervention is needed on behalf of posted data that was pre-empted by a refresh burst—the cycle completes automatically in all cases.

The CY7C960 can be configured to remain off the local bus upon assertion of a local bus holdoff signal (LACK). This facilitates connection of the CY7C960 to a local processor, or dual porting of local memory resources. This requires care in design, however, as long local bus holdoff time translates to VMEbus timeout. See section 3.5.4 for a full description of Local Bus Holdoff.

If the CY7C960 cannot provide a refresh burst to DRAM because of local bus holdoff, it will “remember” the number of missed bursts (up to 64) and “catch up” at the first opportunity. This also may translate to a VMEbus timeout, as a long DRAM refresh burst will delay the acknowledgement of a VMEbus transaction directed to DRAM space.

3.1.4.2 VMEbus Concepts

The CY7C964 devices contain address comparator circuitry that can form all or part of the user’s Slave Address Decoder. The CY7C960 provides timing and control to the CY7C964s that assumes the CY7C964s are indeed part of the Slave Address Decoder. The CY7C964s therefore play a key role during the initialization process, providing the base address of the CR/CSR (Configuration ROM/Control Status Registers) space that the CY7C960 occupies during configuration. The final step in configuration “moves” the CY7C964 base address to the intended position in the VMEbus address space. If the application does not use CY7C964s, the CY7C960 still provides control signals as if the CY7C964s were present. These signals can be used by external circuitry to accomplish the same tasks.

The CY7C960 recognizes Slave Address Regions which are selected by the use of the 4 REGION inputs (3 in DRAM/IO mode). It is the responsibility of the Slave Address Decoder circuitry external to the CY7C960 to drive these pins in response to VMEbus address changes. Thus 16 unique regions (8 in DRAM/IO mode) are recognized by the CY7C960, although normally at least one region would be used as the "Board Not Selected" region.

The CY7C960 is highly flexible in the manner by which it responds to VMEbus transfers. An Address Modifier (AM) Code table is configured during initialization which determines whether specific AM Codes will be recognized. If a particular code is disabled, then no VMEbus DTACK* is provided and the cycle is ignored. Furthermore, an independent AM Code table is available for each of the possible regions in the Slave Address Map. This provides great flexibility to the user: an I/O region could be configured to respond only to single cycle 8/16 bit transfers, while a DRAM region could be configured to respond only to A64 Block transfers.

3.1.5 Address Mapping

The CY7C960 supports many different schemes for address mapping. The function of the external decoder circuitry is to provide signals (REGION[3:0]) to the CY7C960 that define whether the board is addressed in VMEbus Address Space. One simple use of the CY7C960 is shown in *Figure 3−3*. In this simple case, the CY7C960 has been configured to respond only to A16 accesses in the VMEbus space 0h − ffffh; to A24 and CR/CSR accesses from 10000h − ff ffffh. From 100 0000h − ffff ffffh the CY7C960 responds to A32 accesses, and drives local circuitry to route the data to VSB (or Raceway) in the lower part, and to DRAM in the upper part. Above ffff ffffh, the CY7C960 is programmed to respond to A64 or A40 accesses, routing the data to DRAM.

Of course, this fictitious example is impractical due the size of the DRAM segments described. The CY7C960 can support simple contiguous schemes like this one, or arbitrarily complex decoders at the will of the designer.

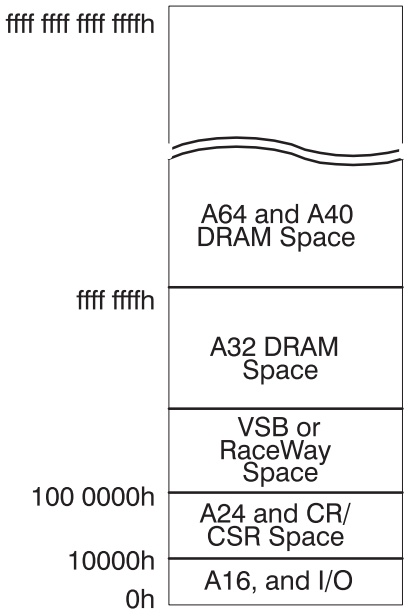

Figure 3−3. Possible VMEbus Slave Address Map