Typical System 13

CHAPTER 13
TYPICAL SYSTEM

INTRODUCTION

The 80960SA/SB processor and bus have been discussed in previous chapters. All processor systems, in order to have a practical value, must be connected to a memory subsystem and to one or more I/O ports. The memory may take the form of RAM, ROM, magnetic storage or some combination. I/O devices are very diverse and range from serial (RS-232 or similar) to laser printer, LAN or some special purpose port. This chapter will focus on optimized DRAM and EPROM memory subsystems because they are the memory types most often encountered. I/O systems will be discussed in more general terms, with the interface to the 82C54 triple timer counter serving as an example.

BLOCK DIAGRAM

Figure 13-1 shows a typical implementation of a processor system built around the 80960SA/SB. The system contains a processor coupled to the rest of the system by a bus. These are discussed in earlier sections of this manual. There are some general control functions, such as clock, address latch reset, data transceivers and address buffers that provide services to the entire system. In addition there are three major subsystems; DRAM, EPROM and I/O.

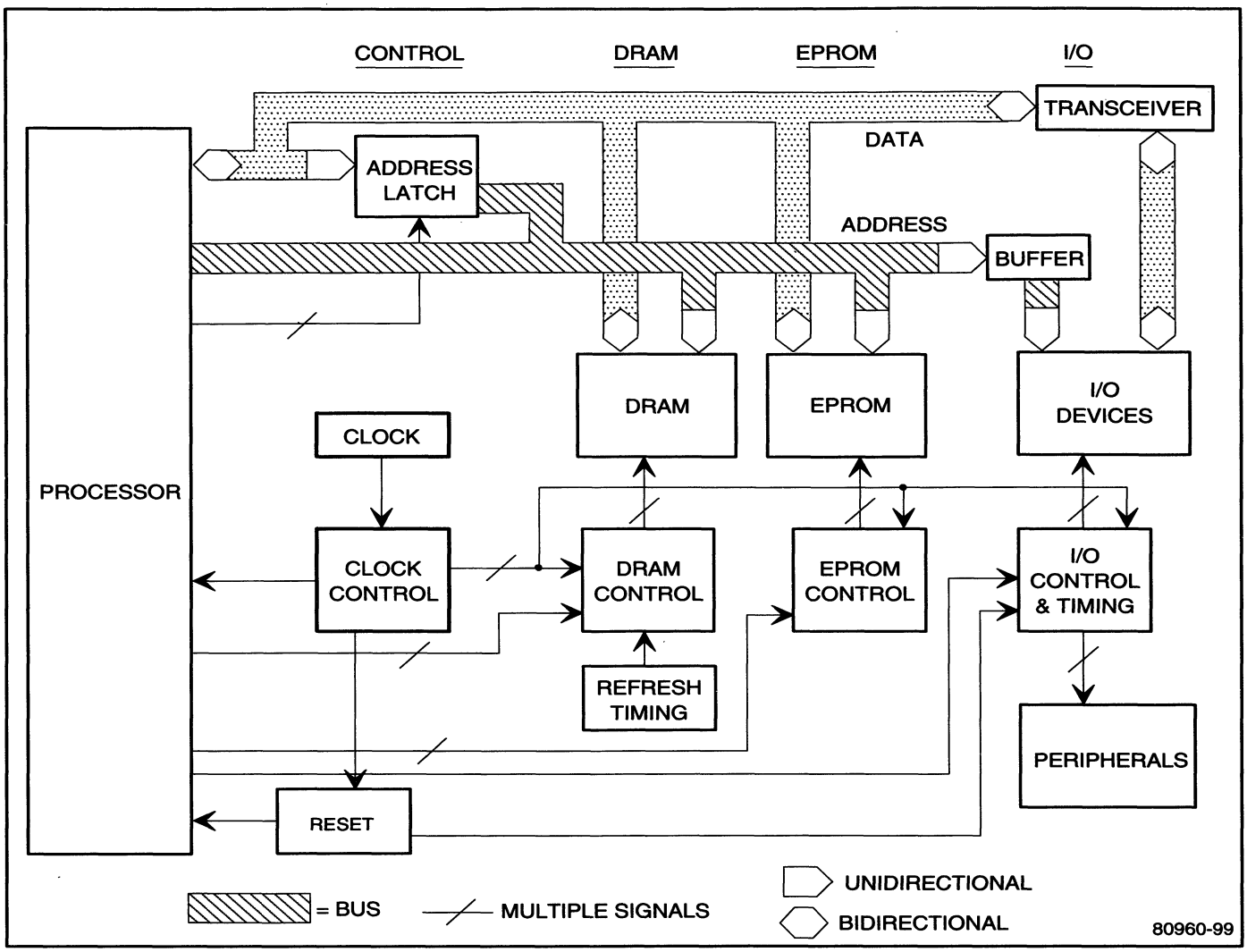

Figure 13-1: System Block Diagram

Dynamic RAM provides a random access storage area for executable code and data. It is characterized by fast access times (less than 150 ns.) and a size sufficient to accommodate all of the data and code required for the application. The subsystem consists of the DRAM devices and the required address, refresh, data routing and timing controls.

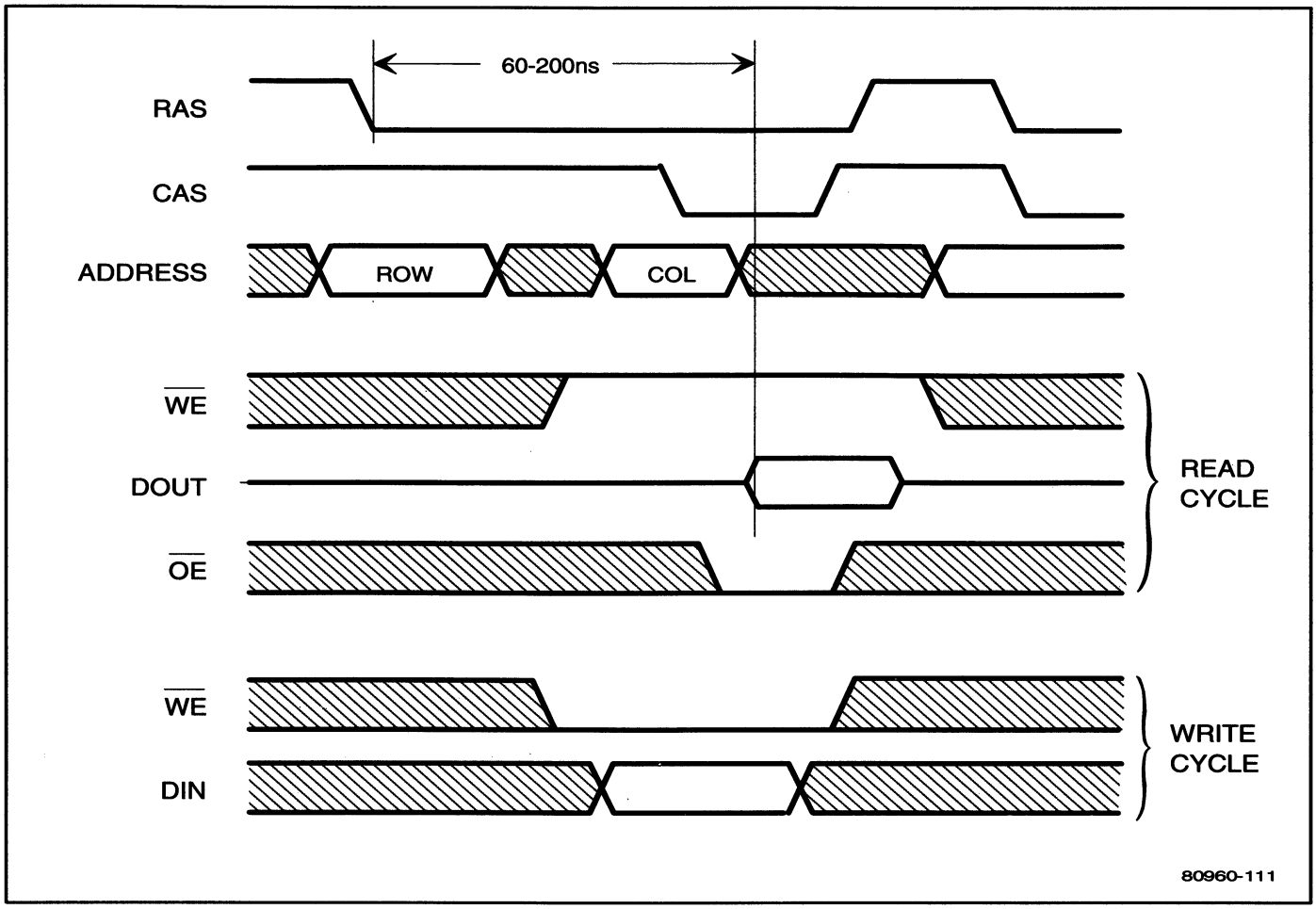

Figure 13-2: 1 Meg X 4 Single Read/Write Cycle

The Erasable Programmable Read Only Memory (EPROM) subsystem provides an area for startup (BOOT) code as well as some specialized I/O routines that are required for the system. The primary characteristic of this subsystem is that the memory content survives system shutdown.

The I/O subsystem may contain a variety of different devices, all with varying characteristics. The common trait of these different elements is timing and control of external processes. I/O ports may be input only, output only or bidirectional. Input only and output only are special cases of bidirectional ports.

MEMORY OPTIMIZATION

In any high performance processor system, the best results are obtained when the memory is matched to the processor. If the processor is much faster than the memory, a large number of "wait states" must be programmed into the operating system so that data is not lost. On the other hand it is usually not economically feasible to install memory with a cycle time much faster than the processor. There are now techniques that allow memory with slower access times to be used with a high performance processor without introducing many (sometimes any) wait states. One of these techniques is called fast page mode.

DRAM is usually organized so that any given address is expressed as the sum of a row address and a column address. In order to access a sequence of data, stored in adjacent memory cells, the fast page mode allows the designer to design a system that addresses the row that contains the data, latch that row address into the memory and then address multiple columns in that row. Figure 13-2 illustrates a single read/write cycle, and Figure 13-3 illustrates the fast page mode read cycle (the fast page mode write cycle exhibits similar characteristics). The fast page mode shows the column addresses appearing on the address bus just before each Column Address Strobe ($\overline{CAS}$). The Row Address Strobe remains active low during the column address sequence. Both $\overline{RAS}$ and $\overline{CAS}$ rise at the end of the memory cycle. Memory devices that would normally require one or more wait states for each address, only require one wait state when the row address is accessed. Column addresses are faster, requiring no wait states. For example, a typical 100 ns fast page mode DRAM has an access time of 100 ns, requiring one wait state for each memory access on a 16 MHz bus (62.5 ns/cycle), but in the fast page mode, each access after $\overline{RAS}$ is set up only requires 55 ns -- eliminating the need for the wait state.

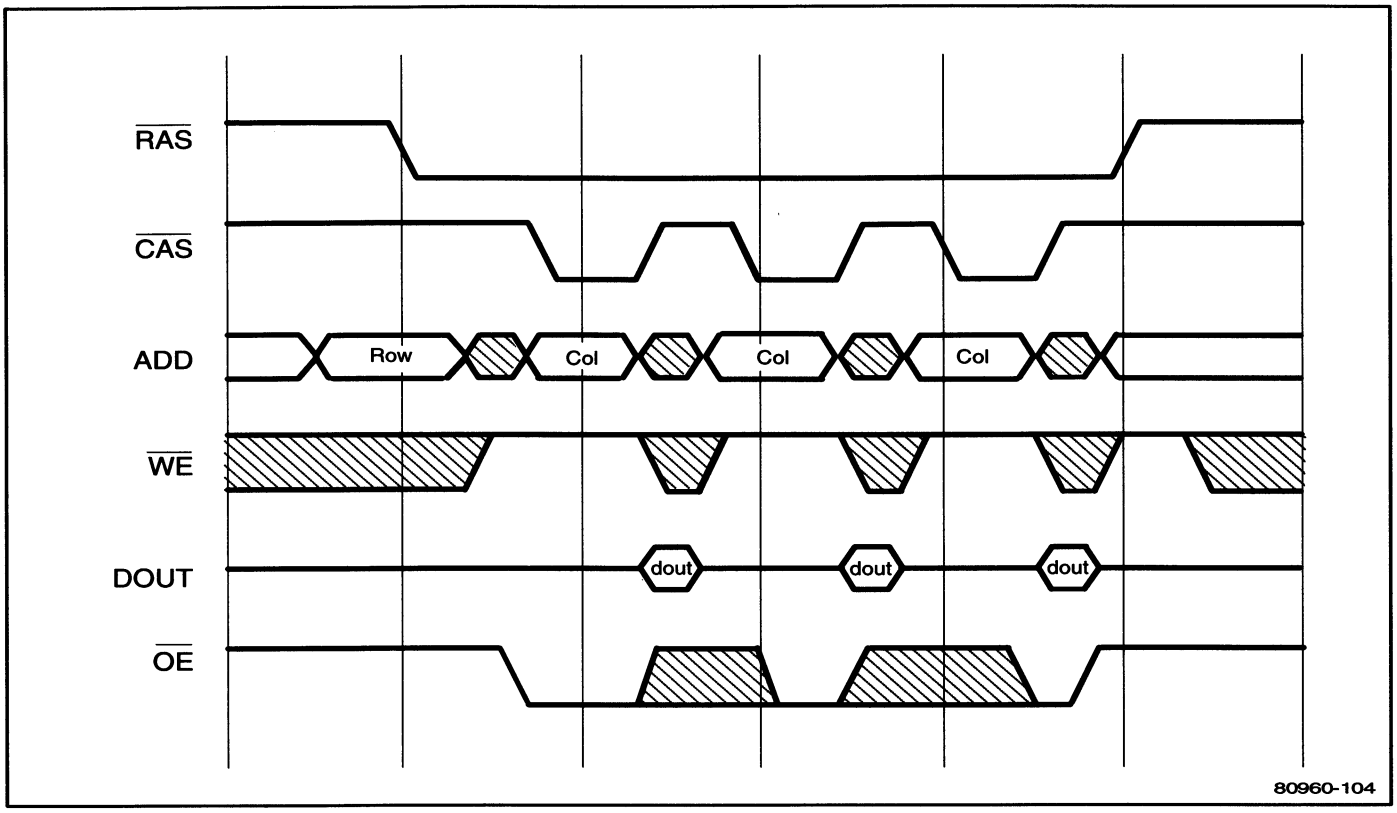

Figure 13-3: Fast Page Mode Read Cycle

The fast page technique is similar to what was introduced in Chapter 12 as burst mode. The 80960SA/SB processor burst mode can take advantage of the fast page mode to improve memory access performance. Figure 13-4 shows the timing relationships of the 80960SA/SB in the burst mode. Notice the similarities in the way the row and column addresses appear. The primary difference between fast page mode and burst mode is that burst mode limits the number of column addresses to eight, while the fast page mode would allow up to 1024 column accesses per row address (for 1 meg by n DRAMs). In the 80960SA/SB, writing to eight memory cells requires 32 processor cycles (or 2 microseconds in a 16 MHz system), while an eight cell write in the burst mode would require only 11 processor cycles (or 688 ns). In this section, the concept of burst mode applied to DRAM will be examined more closely.

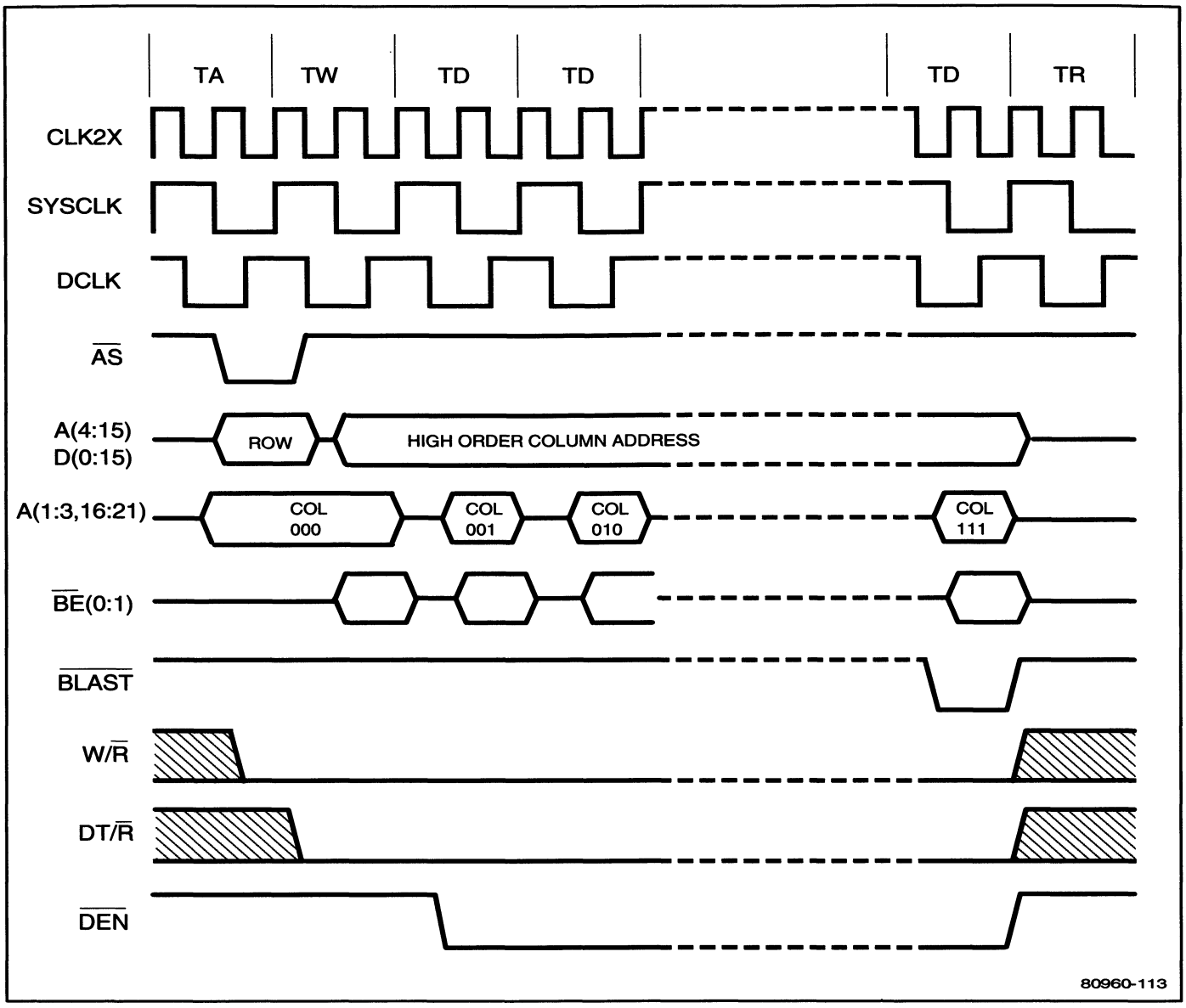

Figure 13-4: 80960SA/SB Timing in Burst Mode (Read Cycle Shown)

In Figure 13-4, the signal DCLK is derived from the system clock, SYSCLK, but leads SYSCLK by one quarter cycle. This gives the designer four extra clock edges per cycle to accommodate the very strict timing requirements of high performance memory systems. The processor initiates a burst mode read or write cycle in much the same way as a normal read or write; that is the processor initiates a T_a state asserting $\overline{AS}$ and placing the address of the desired cells on the address bus. During this state, several other signals must be asserted: W/R (depending on whether the operation is a write or a read), and Byte Enable ($\overline{BE}(0:1)$). Chip Select ($\overline{CS}$) from the address decoder is also asserted at this time. Circuits must be added to generate some signals for the memory devices. The signals required for the memory devices include Row Address Strobe ($\overline{RAS}$), Column Address Strobe ($\overline{CAS}$), Write Enable ($\overline{WE}$), and Output Enable ($\overline{OE}$) (on read cycles). Figure 13-5 shows the system clock signals at the top of the figure with the processor generated signals at the center, and DRAM controller generated signals to the DRAM devices at the bottom. The sequence of events is as follows:

The T_a state sets up the memory subsystem with a row address. The processor then initiates a T_w (wait) state during which, the $\overline{RAS}$ is asserted and stays asserted until the T_r cycle. If the cycle is to be a write cycle, the data is placed on the data bus during this time and the first column address (Col 000) is output to the address bus. The processor now initiates a data state T_d that the DRAM subsystem sees as a CAS state. During a CAS state, the Column Address Strobe, $\overline{CAS}$ is applied to the selected DRAMs followed by the $\overline{WE}$ low (if a write cycle) or a $\overline{WE}$ high and $\overline{OE}$ low (if a read cycle). The processor initiates up to seven more T_d (CAS) states. During the last CAS state, the processor asserts a $\overline{BLAST}$ signal initiating the signals required for a T_r (PREcharge) state. A PRE state puts the address and data bus lines in the tristate condition along with $\overline{BE}(0:1)$, $\overline{CE}$, $\overline{BLAST}$, $\overline{RAS}$, and $\overline{CAS}$ are deasserted. The system goes into a T_a (idle) state waiting for further instructions.

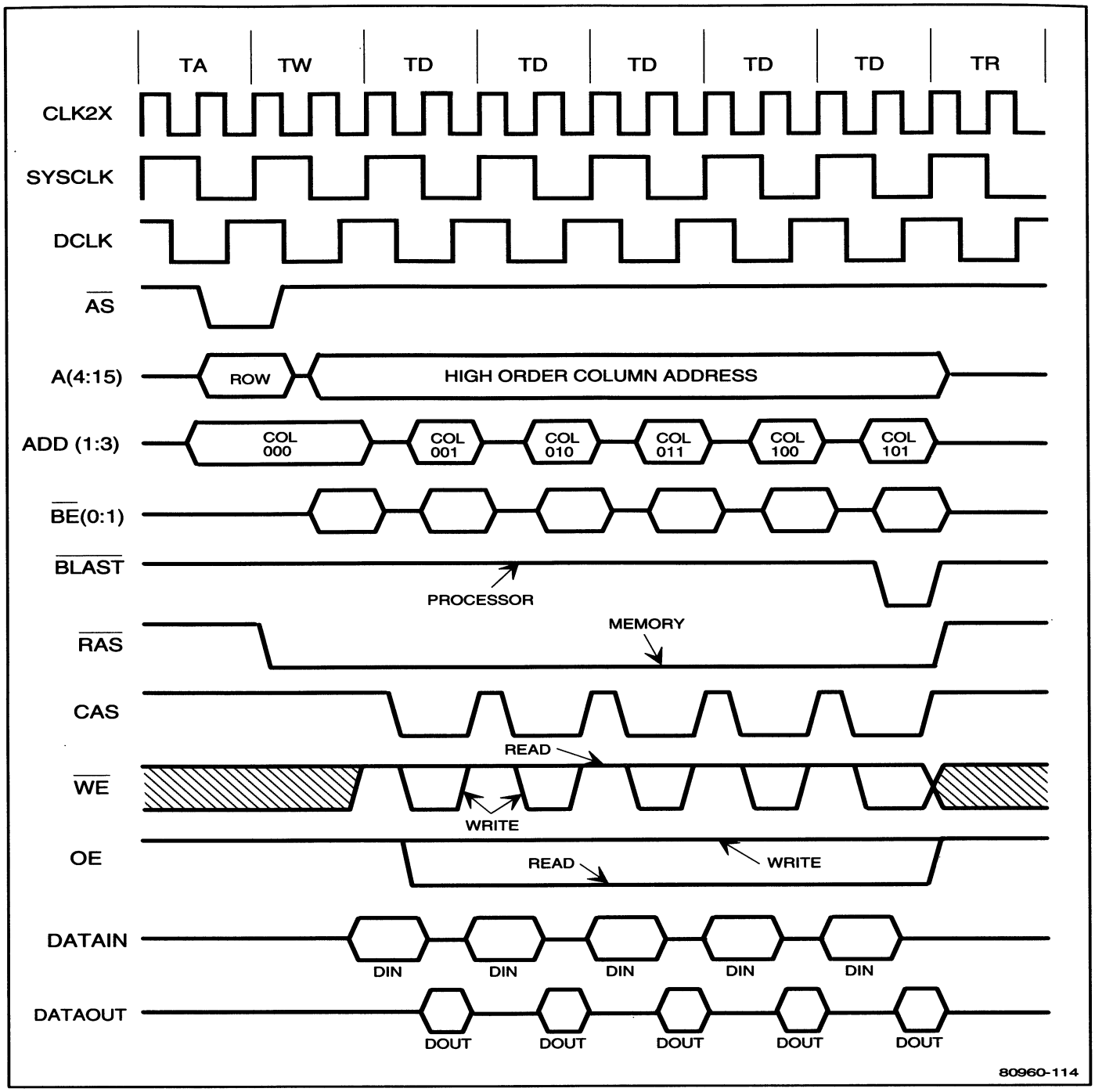

Figure 13-5: Processor vs. Memory Signals 5 Byte Burst

Interleaved Memory

The 80960SA/SB has an external 16-bit data bus and an internal 32bit data bus. When designing with EPROMs, the most efficient design solution is to interleave the memory. This is accomplished by dividing the data into odd and even bytes and then storing all of the odd bytes in one EPROM (odd) and all of the even bytes in another EPROM (even). The EPROMs are then alternately selected by the lowest order address bit and the data multiplexed on the data bus. This technique is discussed at more length in the "EPROM Subsystem" section.

COMMON SUBSYSTEMS

There are several blocks on Figure 13-1 that provide services to one or more subsystems. These common blocks are:

- System Clock
- Reset
- Address Latch
- Bus Buffers and Transceivers

Each of these blocks is examined in detail in this section.

System Clock

In order to provide the accuracy required for a high performance system, particular attention should be paid to the generation of system clocks. In the example system, the clock oscillator runs at 4 times the system clock frequency. The reason for this is twofold:

1. The processor clock of the 80960SA/SB requires that the applied clock be at twice the frequency or the system clock, because the processor has an internal divide by two circuit.
2. The example system uses Programmable Logic Devices (PLD's) to generate the critical timing signals. Using a 4X clock oscillator allows the designer to generate signals synchronized to the rising or falling edge of the processor clock.

A 64 MHz clock module is used in the example system. These modules are readily available and are small in size. The clock signals generated in the example system are CLK2X for the processor, SYSCLK1 and SYSCLK2 for the rest of the system, DCLK for critical DRAM timing. and XBCLK for the external bus clocking. Figure 13-6 shows a block diagram of the system clock generator. Note the presence of the 22 ohm termination resistors. These resistors provide isolation for the clock signals and help ensure the integrity of the signals going to other parts of the system. In the example system, a registered PLD is used to generate clock signals and other time sensitive signals. This design keeps clock skew at a mimimum because all signals are generated synchronously from the same device. There is ample room on the PLD to generate other time critical signals. These signals are discussed when the blocks that use them are discussed. Figure 13-7 shows the timing relationship between the various clock signals.

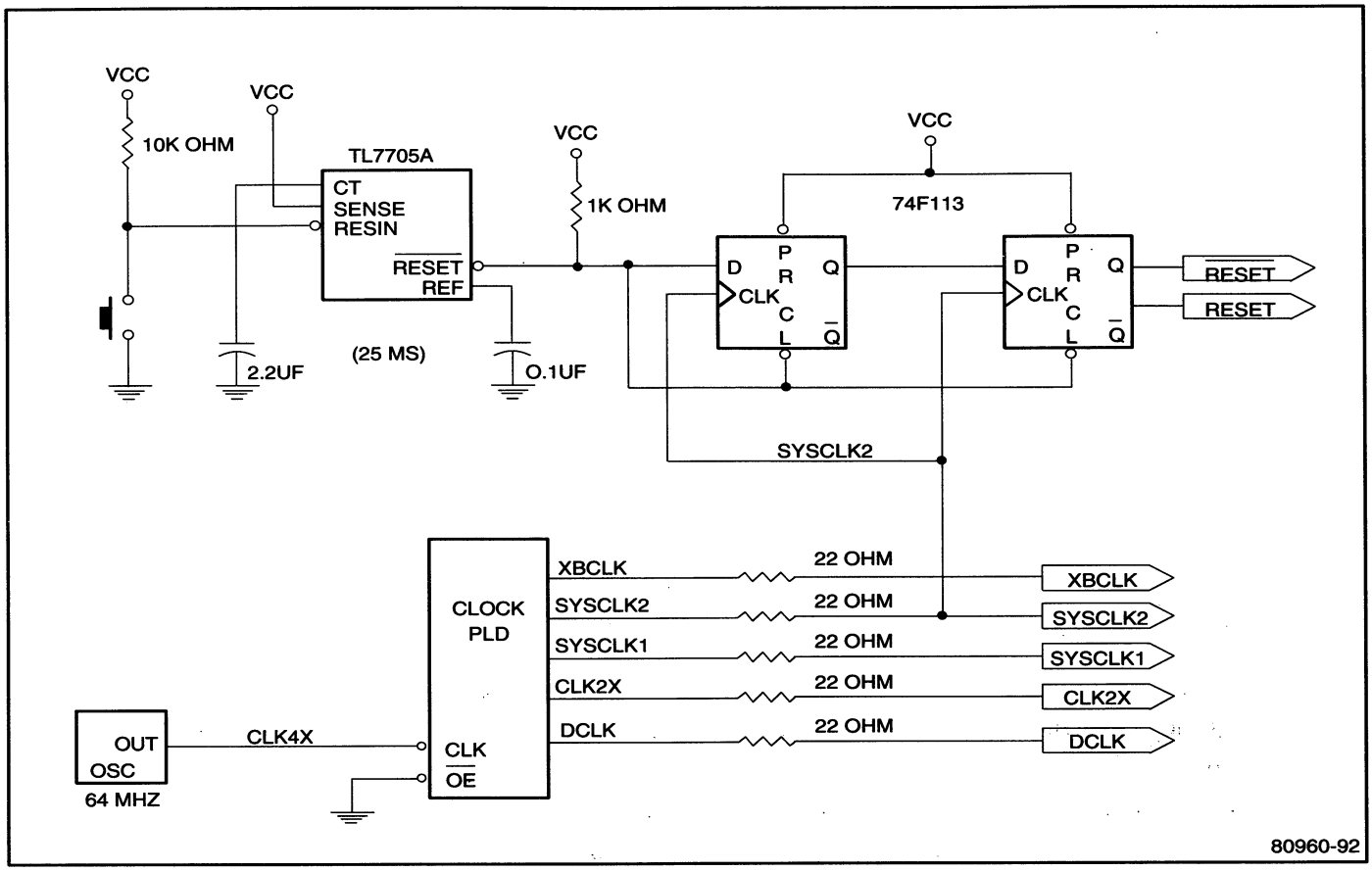

Figure 13-6: Clock and Reset Block Diagram

In addition to the system clock signals generated in the PLD there are several other clocks generated the are used for asynchronous actions. One good example of an asynchronous action is DRAM refresh. The example system uses a master asynchronous oscillator running at 16 MHz. A 4 stage programmable counter provides asynchronous clocks (with respect to the processor) of 1, 2, 4 , and 8 MHz. Of these signals, the 8 MHz and 2 MHz signals are the only ones discussed in this section. The 8 MHz clock is used for DRAM refresh and for the I/O timer and is discussed below. The 2 MHz signal is also used for the I/O timer.

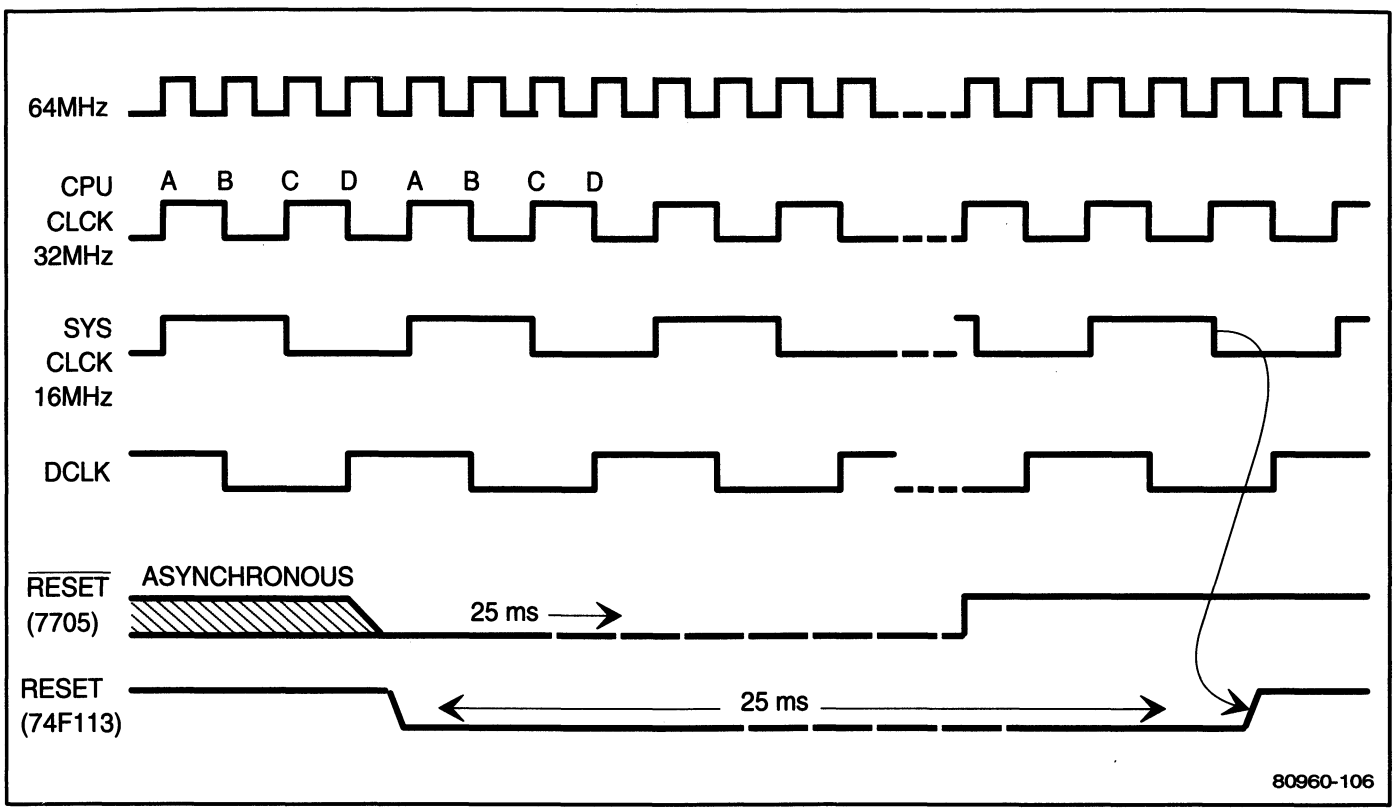

Figure 13-7: Clock and Reset Timing

Reset

The reset circuit is provided to ensure that the processor can synchronize the system clock with the bus. In the circuit discussed below, RESET's rising edge causes the next CLK2X edge to be the processor "A" clock edge. A block diagram for an example circuit is shown in Figure 13-6. In this circuit, the pushbutton switch enables a TL7705A reset generator. This device is essentially a smart monostable multivibrator. It has some built-in features that "debounce" switch contacts and ensure a clean output pulse to the rest of the reset circuit. The timing capacitor, connected to the C_t terminal sets the width of the reset pulse. In this example the reset generator provides a single negative edge and remains at a low level for about 25ms. The reset level output is connected to a dual negative edge triggered D flop at the D terminal of the first device and the CLR terminals of both devices. The clock for the D flops is SYSCLK2. SYSCLK1 could have been used. The choice was arbitrary. This combination of signals provides a reset signal that meets the system requirement for an asynchronous reset that the reset line be held low for at least 41 CPU clocks (1.3 microsecs.).

Address Latch

The address latch is shown in the block diagram of Figure 13-8. This circuit is required because the processor shares the first sixteen lines with data (0:15). The latch is necessary to preserve the latch address after it is deasserted. Since the processor provides an active high ALE (Address Latch Enable), no inverter is necessary for data latch integrated circuits that require an active high signal. Latch/demultiplexer circuits usually also have an output tristate control. This may be used in the event that the latch/demultiplexer outputs are in their high impedance state when another bus master controls the system. The 80960SA/SB has a HLDA (Hold Acknowledge) signal to place the latch/demultiplexer in the high impedance state.

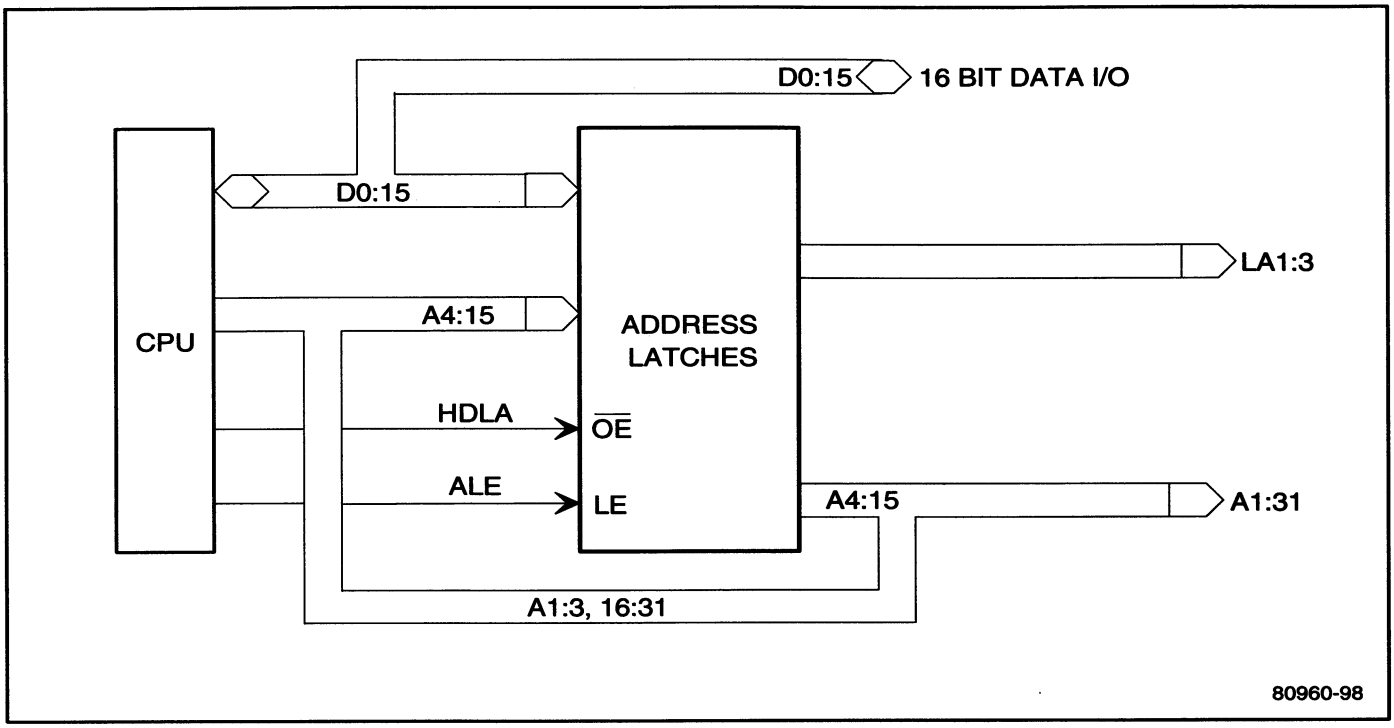

Figure 13-8: Address Latch

Bus Buffers and Transceivers

In large systems, bus buffers and data transceivers are required to prevent signal degradation due to line loading effects. If the system designed around the 80960SA/SB has many I/O ports, external devices or long lines attached to the address or data lines, it is a good practice to buffer these lines. In the case of address lines, a unidirectional buffer is all that is required. Such a circuit is shown at the top of Figure 13-9. No timing or enable lines are required or used in this implementation. Data lines are usually bidirectional and therefore require bidirectional bus transceiver. Such a circuit is shown at the bottom of Figure 13-9. Here the data bus between the system and the I/O subsystem is buffered with a transceiver with directional control provided by the DT/R signal. Refer back to Figure 13-4 for timing information.

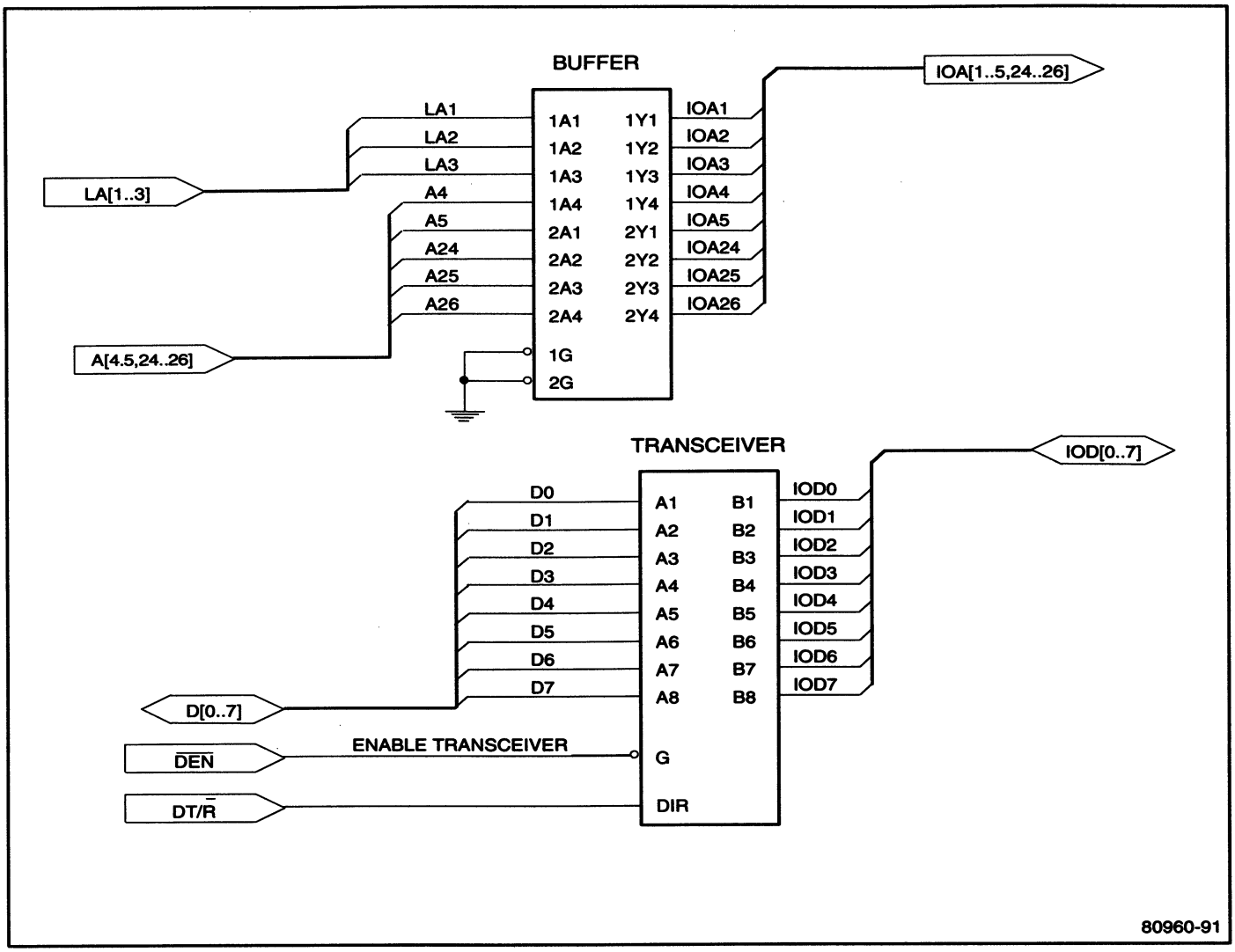

Figure 13-9: Bus Buffers and Transceivers

Because the memory system is optimized, no buffering is used in either the DRAM or EPROM subsystems. While buffering might offer some advantages, the delays introduced would degrade system performance. In the case of the EPROM subsystem, buffering is accomplished by the data multiplexers at the output of the interleaved devices. These subsystems are more fully described below.

HIGH PERFORMANCE DRAM EXAMPLE

To illustrate an optimized DRAM subsystem, the following example is used. The DRAM devices are 1 megabit by 4 fast page mode DRAMs. For the 16 Mhz processor system, the 100 ns grade devices will be sufficient since the fast page mode cycle time is 55 ns, while one burst bus cycle time is 62.5 ns. Some of the optimization techniques used include:

- the use of Programmable Logic Devices to generate subsystem timing signals.
- a separate clock that leads the system clock by one quarter cycle (DCLK) is used to set timing on critical signals.
- address and data buffering are not used.
- DRAM read and write cycles are fast page mode.

The block diagram of the DRAM subsystem is shown in Figure 13-10. The design of this subsystem is implemented with a minimum component count. The DRAMs are HM514400-100 one megabit by 4 fast page mode dynamic RAMs. For a 16 bit wide data bus, only four of these devices are required for each megabyte of memory. RAM timing and control signals are generated in two Programmable Logic Devices dedicated to the DRAM subsystem and a portion of the PLD used to generate the system clock signals. The rest of the subsystem is composed of a few counter, and multiplexer devices.

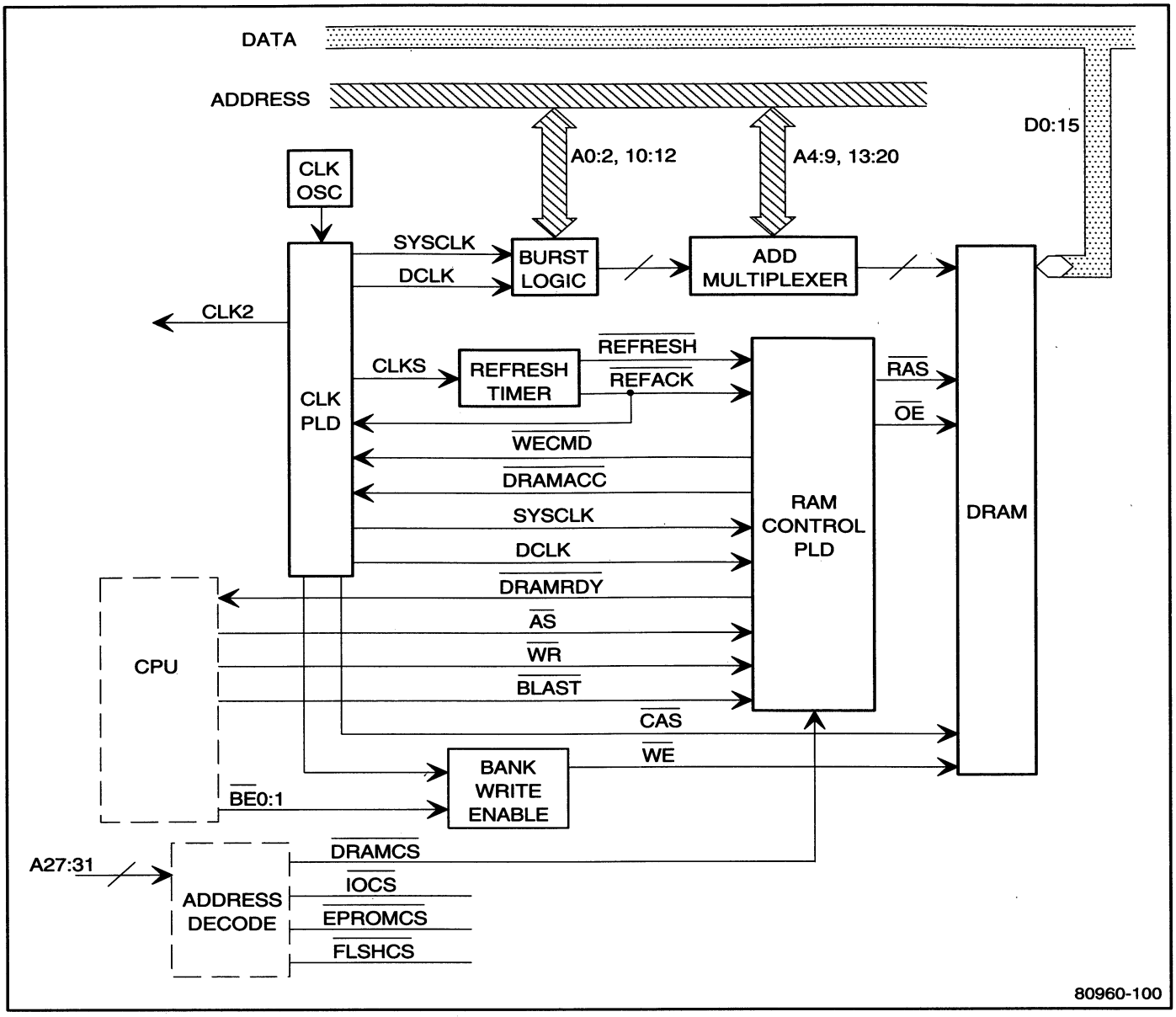

Figure 13-10: DRAM Subsystem Block Diagram

Some details about the DRAMs that form the heart of this subsystem are in order. These DRAMs, like most current high density RAM devices are organized in a square matrix of cells. The matrix is 1024 by 1024 cells and is four bits wide. Accessing any address is accomplished for a 1 megabit x 4 DRAM by asserting and latching a row address and then asserting and latching a column address. The address word size is 10 bits for the row segment and 10 bits for the column segment. In addition to addresses the DRAM also has inputs for a Row Address Strobe ($\overline{RAS}$). Column Address Strobe ($\overline{CAS}$), Output Enable ($\overline{OE}$) and Write Enable ($\overline{WE}$). The four output pins are bidirectional for both data input and data output. The output portion of the data section is tristated and is controlled by the Output Enable pin. The basic timing for these devices in the fast page mode is shown in Figure 13-3 earlier in this chapter. All of the control signals for these memory devices are asserted low. There are many timing constraints on these high performance devices and the reader is referred to the manufacturer's data sheets for details. In this example, parameters such as access times and hold times are discussed. A complete discussion of timing is beyond the scope of this simple example.

In fast page mode operation, the row address is asserted on the DRAM device pins. $\overline{RAS}$ is then asserted and remains asserted for the rest of the access cycle. After one processor clock cycle time, the address lines are switched to the starting column address followed by the $\overline{CAS}$ signal. During a write cycle the input data is presented on the data lines, $\overline{OE}$ is asserted high to assure that data out is in the high impedance state. $\overline{WE}$ low follows, writing the data to the addressed cells.

In the system, the timing of the signal events at the DRAM terminals must be synchronized with the processor cycles, causing some compromise in the performance of the memory devices. But with the clock selections available in the example design, very accurate clocks are available and any performance compromise is minor.

The 80960SA/SB address bus is 32 bits wide, but only nine lines are required to address the DRAM. The address multiplexer switches the row and column addresses to the address inputs on the memory devices. The circuit is shown in block diagram form in Figure 13-11. The row/column multiplexer accepts address bits 4 through 9 (high order row address) and 12 through 21 (high order column address) and multiplexes them onto Memory Address (MA) lines 3 through 9. The low order bits A 1 through 3 (row), and 10 through 12 (column) are multiplexed through the address counter. The address counter cycles the low order address range indicated by data presented to the counter, up to eight addresses, maximum. The row address is loaded into the preset counter and shifted out on the next DCLK. Then a three bit number representing the number of columns to be accessed (1 through 8) is input to the counter. The counter then counts from zero to the desired number to complete the burst cycle.

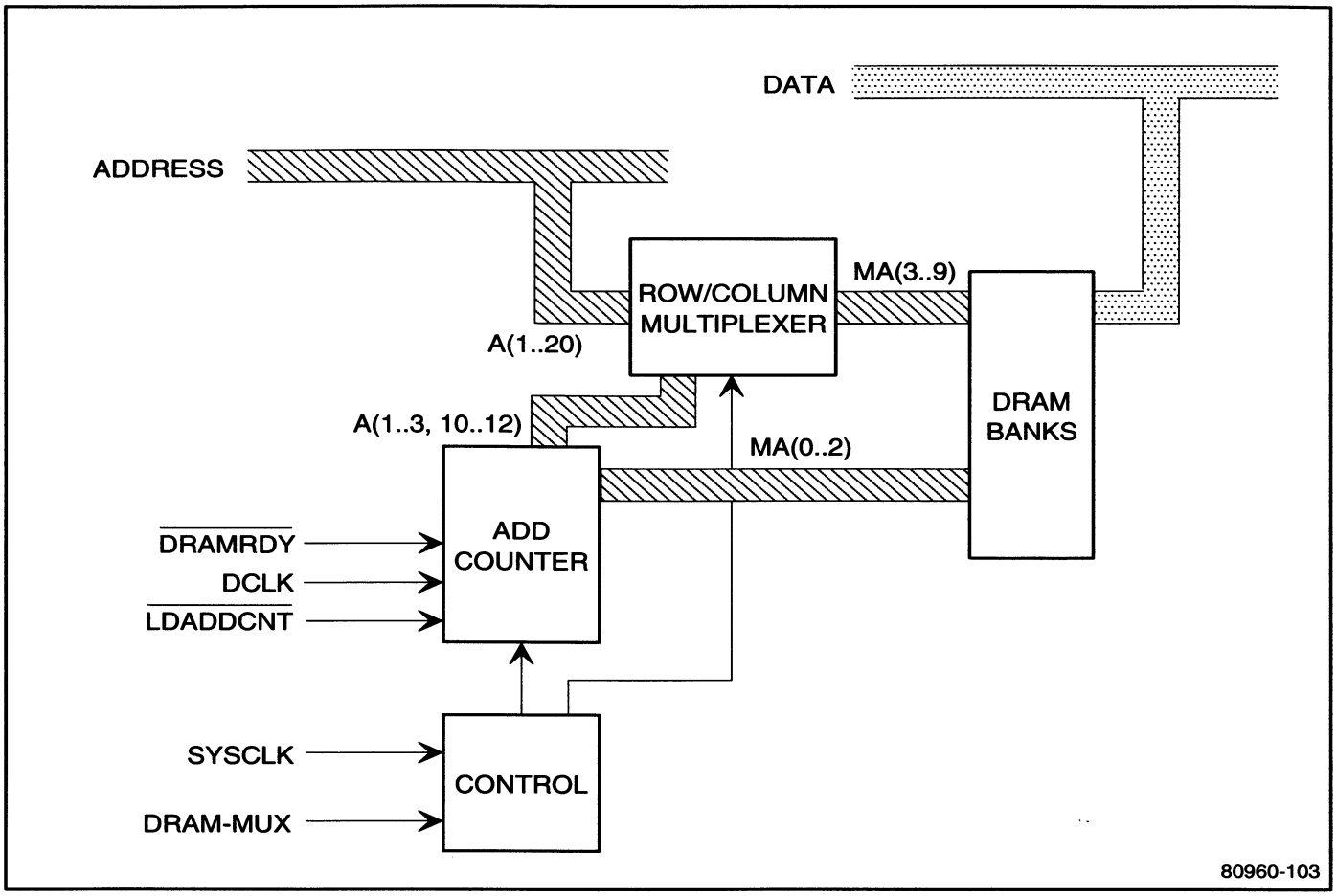

Figure 13-11: DRAM Address Section

The DRAM control signals are generated in the PLDs, as shown in Figure 13-10. $\overline{CAS}$, $\overline{WE_0}$ and $\overline{WE_1}$ are generated in the clock PLD along with SYSCLK and DCLK. The reason for this is that the clock PLD is strobed with the crystal oscillator running at 64 MHz -- four times the system frequency, thus making very fine control of these critical edges possible. $\overline{RAS}$, $\overline{OE}$, the burst logic and address multiplexer signals are generated by the RAM control PLD. The RAM control PLD also generates signals accessible to the rest of the system:

- $\overline{WECMD}$ and $\overline{DRAMACC}$ are used to enable generation of the DRAM $\overline{WE}$(0:1) and $\overline{DRAMCAS}$ signals in the clock PLDs.
- $\overline{REFACK}$ -- acknowledging a refresh request.
- MUX and load address count LDADDCNT signals to the address multiplexer.

The DRAM control state diagram is shown in Figure 13-12, and the DRAM access & refresh state diagrams are shown in Figure 13-13. While no accesses are requested, the subsystem stays in the IDLE mode, equivalent to the processor T_i (idle) state (except when refresh is requested -- see below). When AS and DRAM CS are asserted, the subsystem enters the RAS state equal to a processor wait state (T_w). In this state the row address is set up and RAS is asserted. Next, the memory enters a CAS state (a processor T_d state). Here the first column address is set up and a CAS is asserted. The next memory state is either a PRE -- indicating the end of a cycle or another CAS state. The branch chosen depends on the BLAST signal. BLAST is asserted low only when the last CAS is in process.

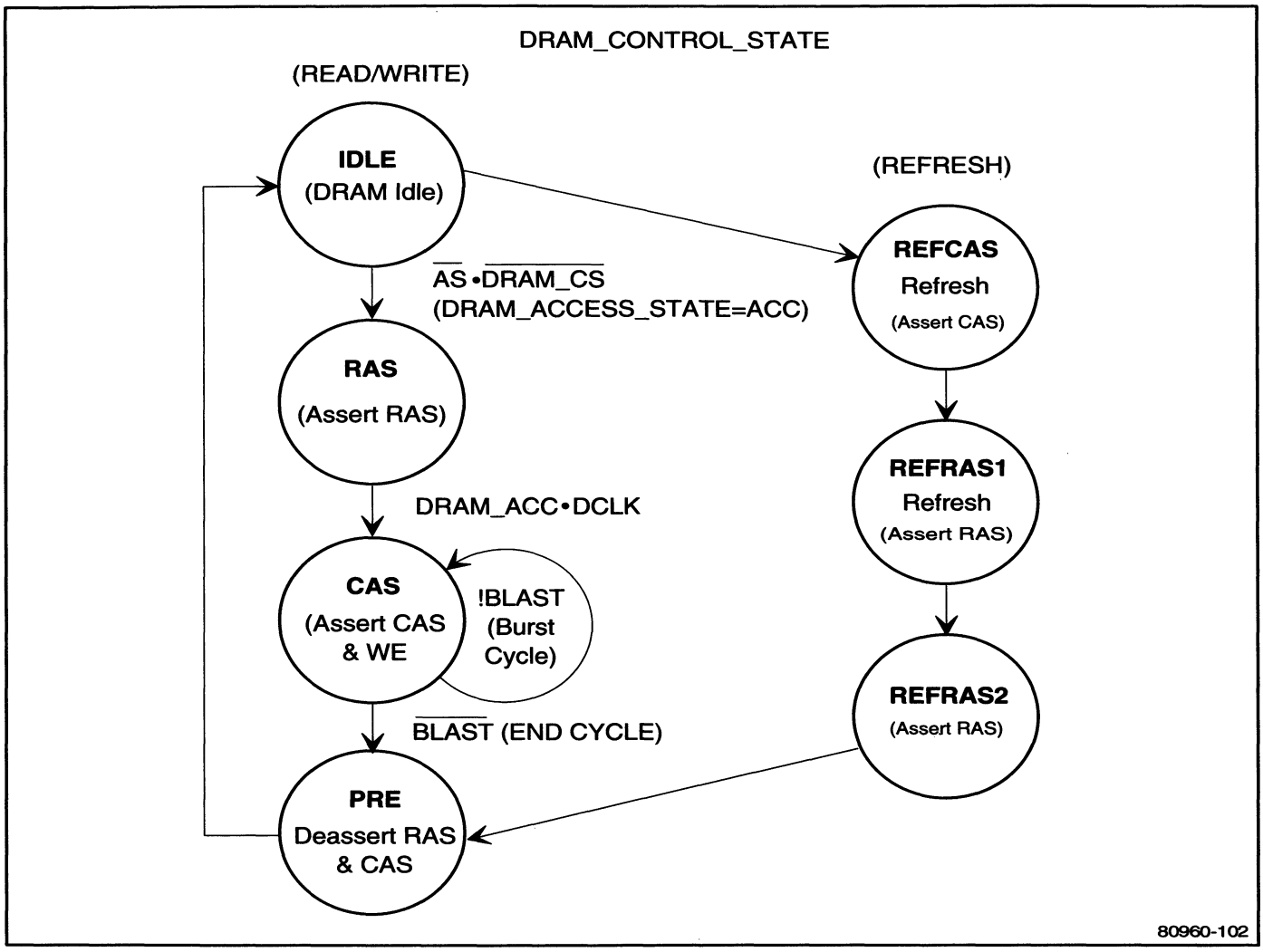

Figure 13-12: DRAM Control State Diagram

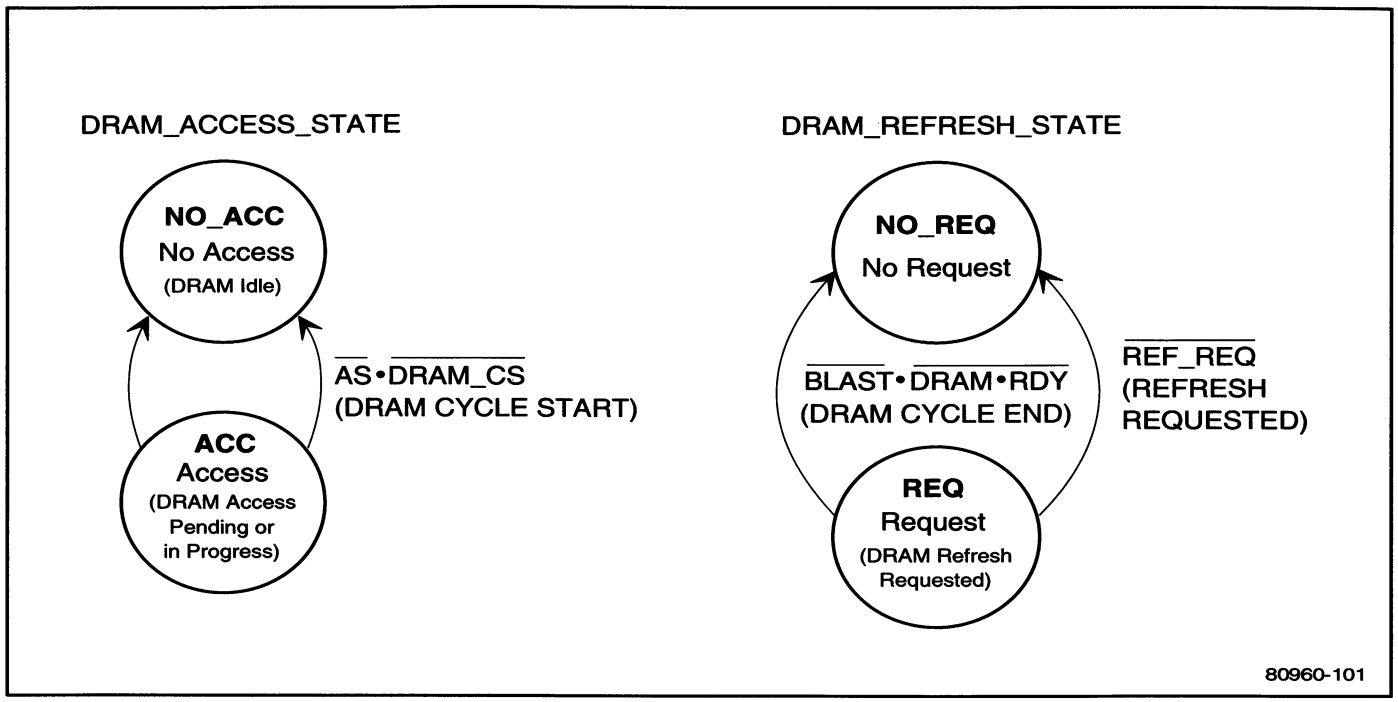

Figure 13-13: DRAM Access and Refresh State Diagram

The refresh request is generated by the refresh timer which is clocked by an asynchronous 8 MHz clock. Any row access refreshes all cells in that row, so a complete refresh cycle can be accomplished by accessing all 1024 rows. Total time for this operation is about 125 microseconds. The memory devices have an internal counter that increments the row refresh address after every CAS before RAS cycle, avoiding the necessity of "shutting down" the memory while a refresh is in progress. Access to the refresh counter is accomplished by asserting CAS before RAS. The memory refreshes the next row and then returns control to the system. As long as all cells are refreshed at an interval no longer than 16 ms, there is no need to do a complete refresh. The memory control PLD arbitrates contention and collision between refresh requests and memory access requests. Basic refresh timing is illustrated in Figure 13-14.

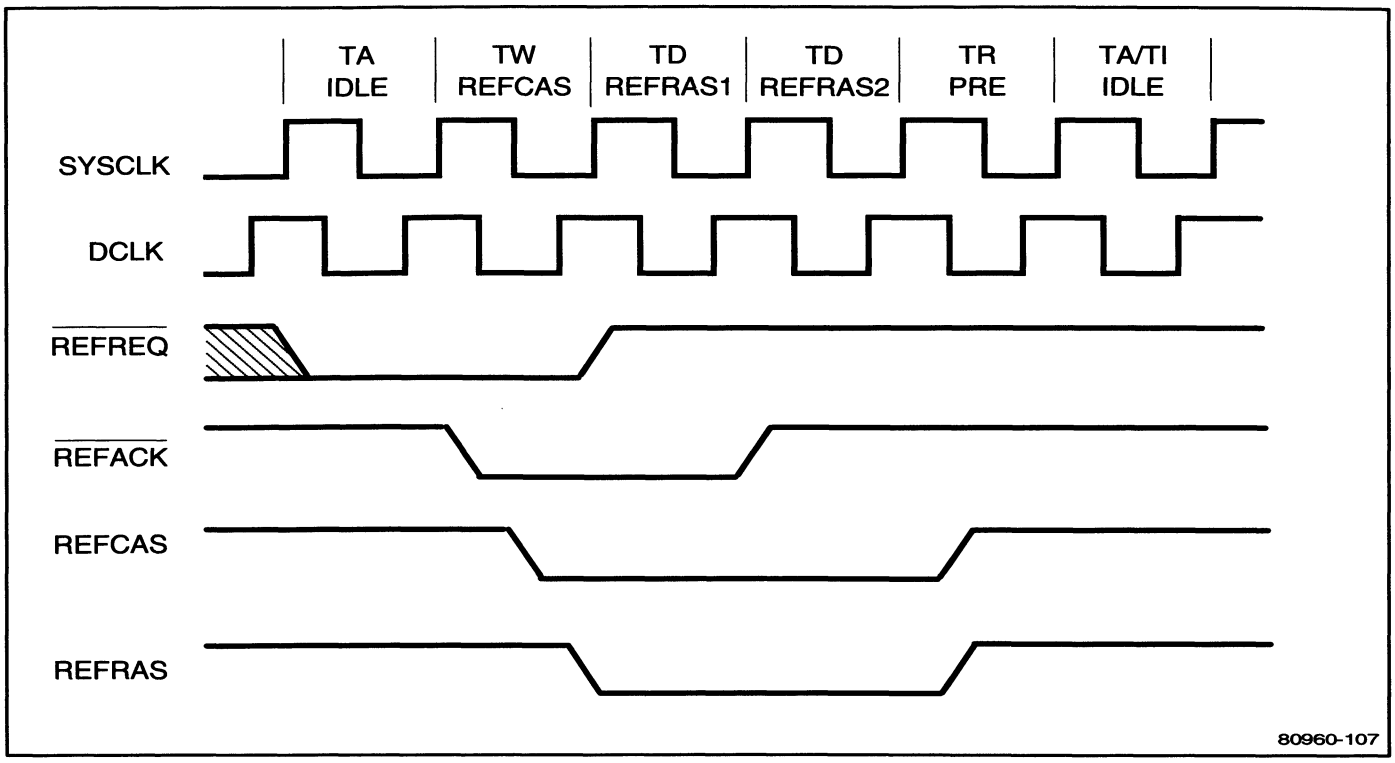

Figure 13-14: DRAM Refresh Timing

HIGH PERFORMANCE EPROM EXAMPLE

EPROMs are usually included in systems requiring non-volatile memory, that is memory whose content survives a power down condition, a system reset or other interruption of normal operation. The content of these memories is typically startup information, some hardware driver or configuration information and executable code that needs to be "resident" in the system.

EPROMS can be either electrically erasable or ultraviolet (UV) light erasable. Electrically erasable EPROMs may be erased and reprogrammed while still plugged into the system. UV erasable devices generally require removal from the socket for erasure and reprogramming.

A typical EPROM subsystem is shown in the block diagram of Figure 13-15. In this example there is room for two different programmable array types: a flash memory and an interleaved EPROM. In this example the interleaved EPROM will be stressed because all of the same factors apply, plus the interleaved array has an output multiplexer that requires some explanation. In addition to the blocks mentioned above, there is a control block that accepts input signals from the rest of the system and creates control signals appropriate to the memory devices installed. In addition to the address bus entering the subsystem and the data bus leaving it, there are seven signals coming from the system. Of these, five come directly from the processor: AS, BLAST, W/R, and BE(0:1). The SYSCLK signal comes from the same clock generator as the DRAM section. The remaining signal, EPROM CS, comes from an address decoder that selects the device in accordance with an address code in the five high order bits (27:31) of the address bus.

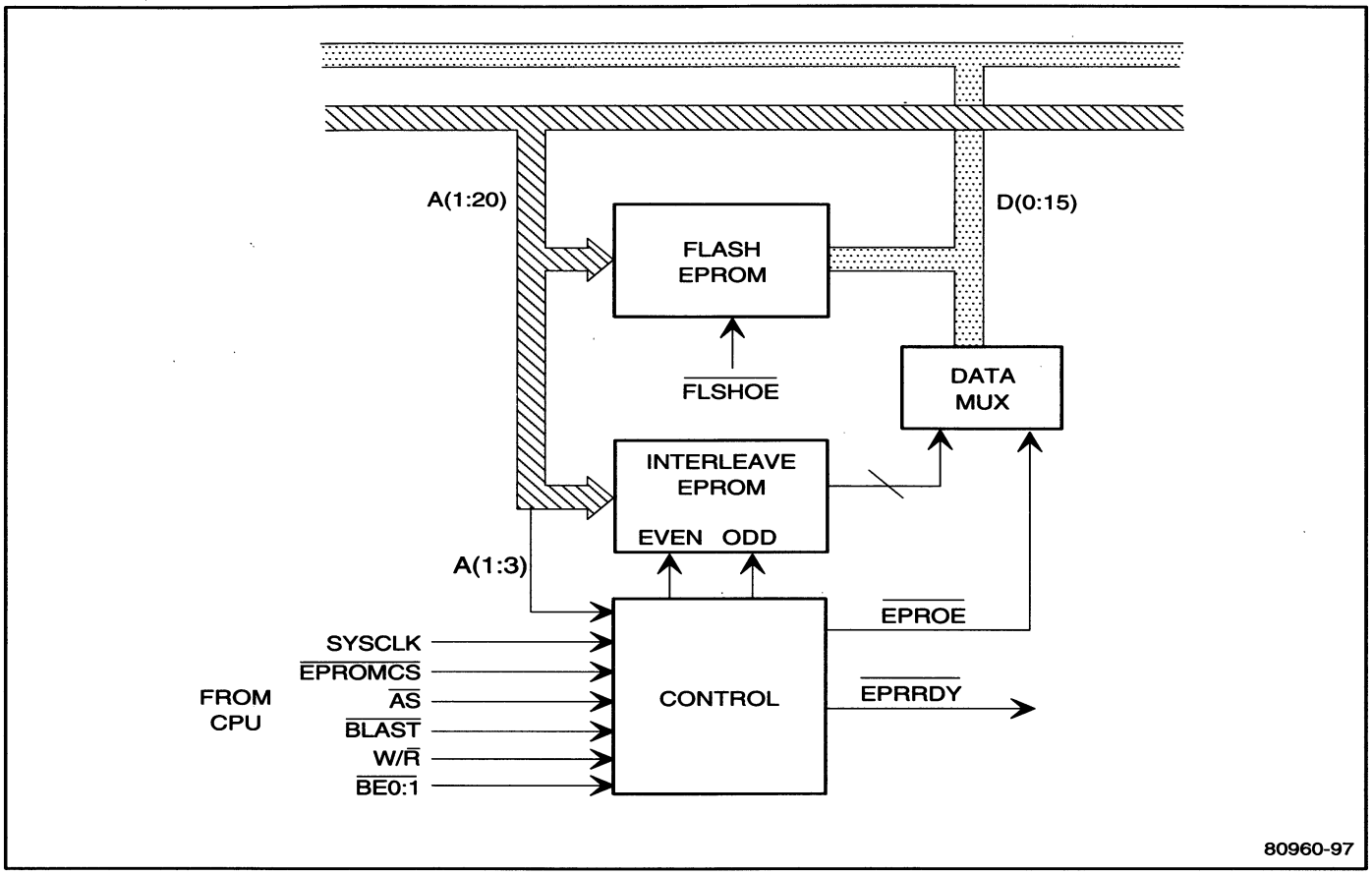

Figure 13-15: EPROM Subsystem Block Diagram

Interleaving is the process of dividing up the bytes of the memory such that the even data words are placed into one EPROM and the odd data words are placed in the other. When the data is read out of the memory the same address is applied to both devices and the low order address bit, A1, is used to strobe first the EPROM containing the odd words and then the EPROM containing the even words. This process continues until all of the required data has been read. Why should this be done? EPROMs generally have much slower performance that static or dynamic RAMs. Interleaving improves system performance by reducing access time for burst reads. The output multiplexers speed up data turn, off times, reducing the chance of output contention between leaves.

In order to complete the system, an output multiplexer is required to keep the output of one EPROM from interfering with its companion. This is accomplished with standard 8 line to 4 line data multiplexers. The enable and select inputs to the multiplexers are driven by the EPR OE and address line 1. Timing for a typical interleaved EPROM is shown in Figure 13-16.

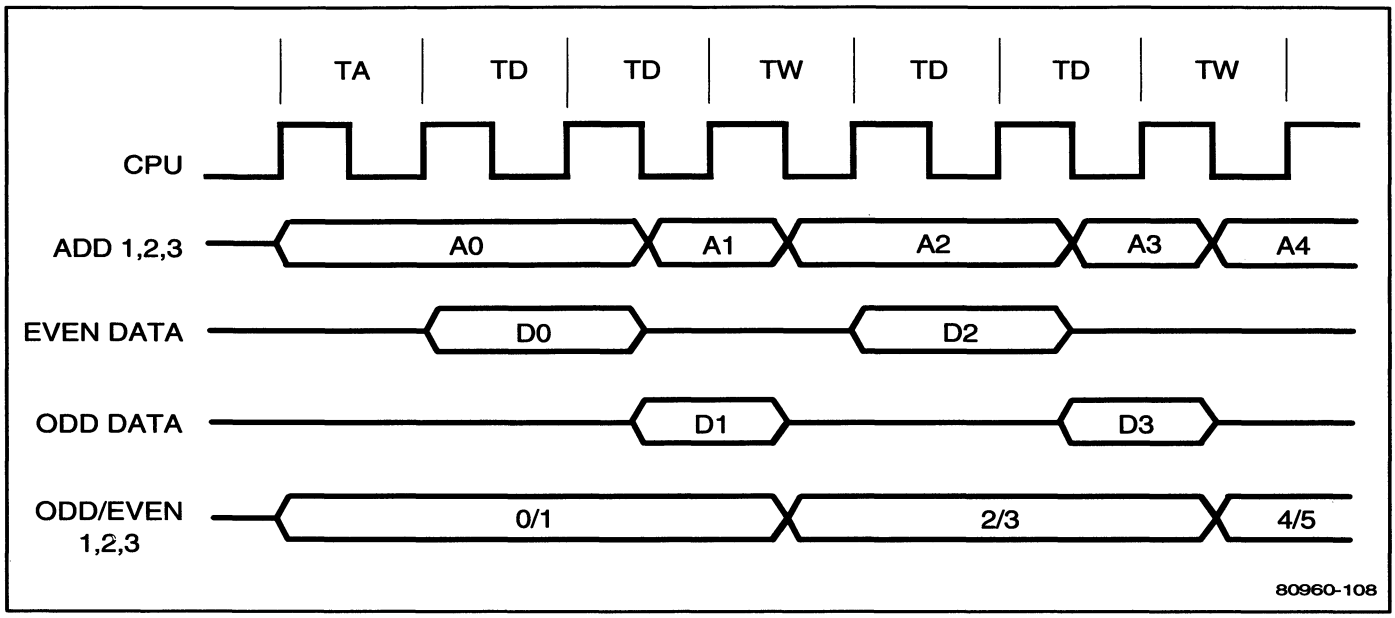

Figure 13-16: EPROM Subsystem Timing

I/O SUBSYSTEM EXAMPLE

Microprocessor systems can drive a large variety of input/output (I/O) devices. Some of these devices are:

- Timer/Counters
- LAN Controllers
- Laser Printer Interface
- Serial Port (RS-232)
- Parallel Port
- DIP Switches
- Indicators (LED)

Figure 13-17 shows a block diagram of such an I/O subsystem. The 80960SA/SB processor uses a memory mapped address space to access I/O devices. Consequently memory-like instructions can perform I/O operations. For example, the 80960SA/SB processor's LOAD and STORE instructions directly support 8 bit and 16 bit data moves to and from I/O devices.

I/O devices can be divided into three types from the perspective of a processor: read only, write only and read/write. In this chapter, a read/write device is examined because read only and write only devices are simply special cases of the read/write device type.

In Figure 13-17 the decode and wait generator block contains three kinds of circuit: address decoders, PLDs and Wait State Counters. The address decoders generate specific I/O Read (NNNN RD), I/O Write (NNNN WR) and I/O Chip Select (NNNN CS) that access devices connected to the system. The PLD, like the PLDs discussed in the DRAM section provide accurate timing and control signals to the rest of the subsystem. In particular, the signals generated in the example system are: IORD and IOWR as inputs to the address decoders, I/O Ready (IORDY) and I/O Enable ($\overline{IOEN_0}$ & $\overline{IOEN_1}$). The PLD also generates a signal that resets the wait state counter (IOWSRES). When asserted this signal resets the counter to 0 and starts the count.

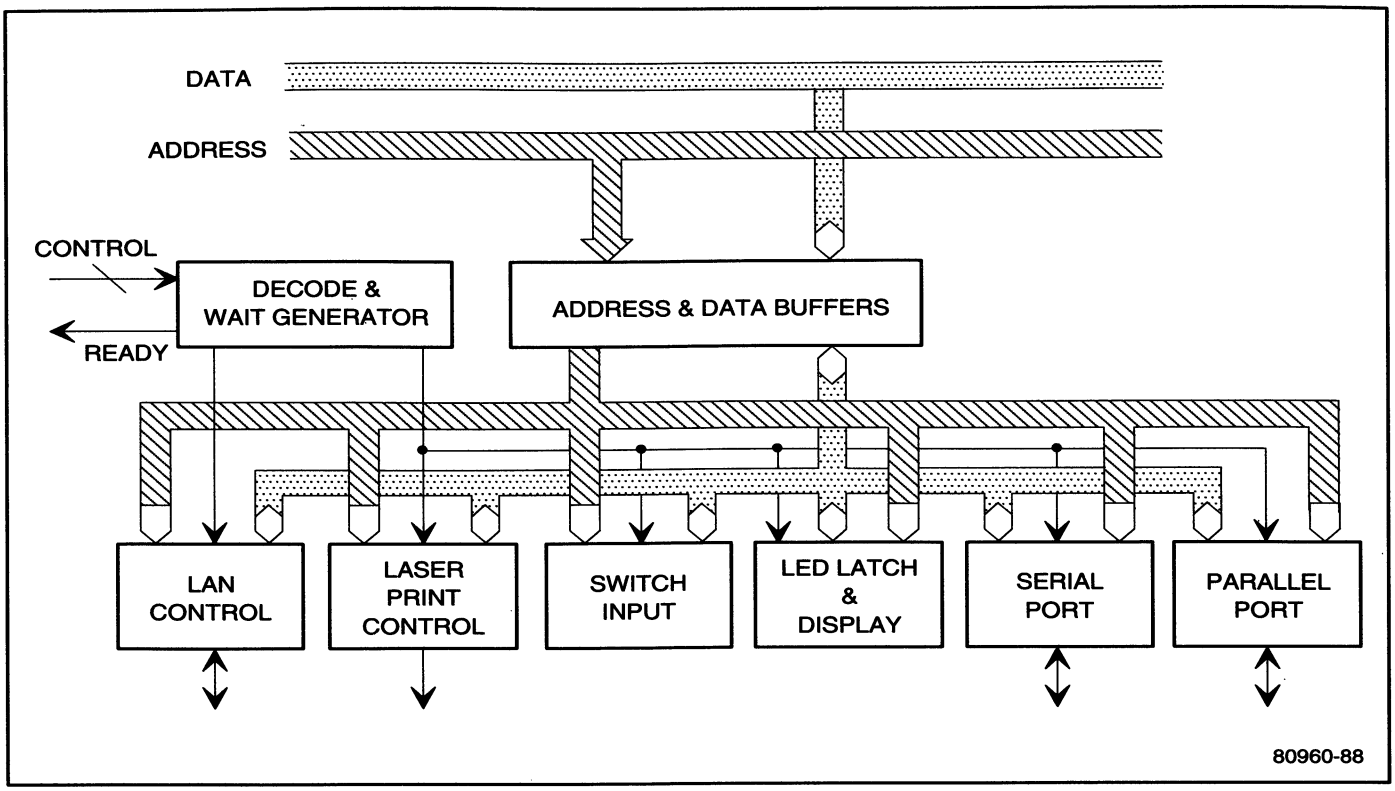

Figure 13-17: I/O Subsystem

The address and data buffers are described earlier in this section under the heading "Common Subsystems."

General timing for I/O operations is controlled by the I/O state machine in the I/O PLD. The state diagram for this state machine is shown in Figure 13-18. The associated timing diagram is shown in Figure 13-19.

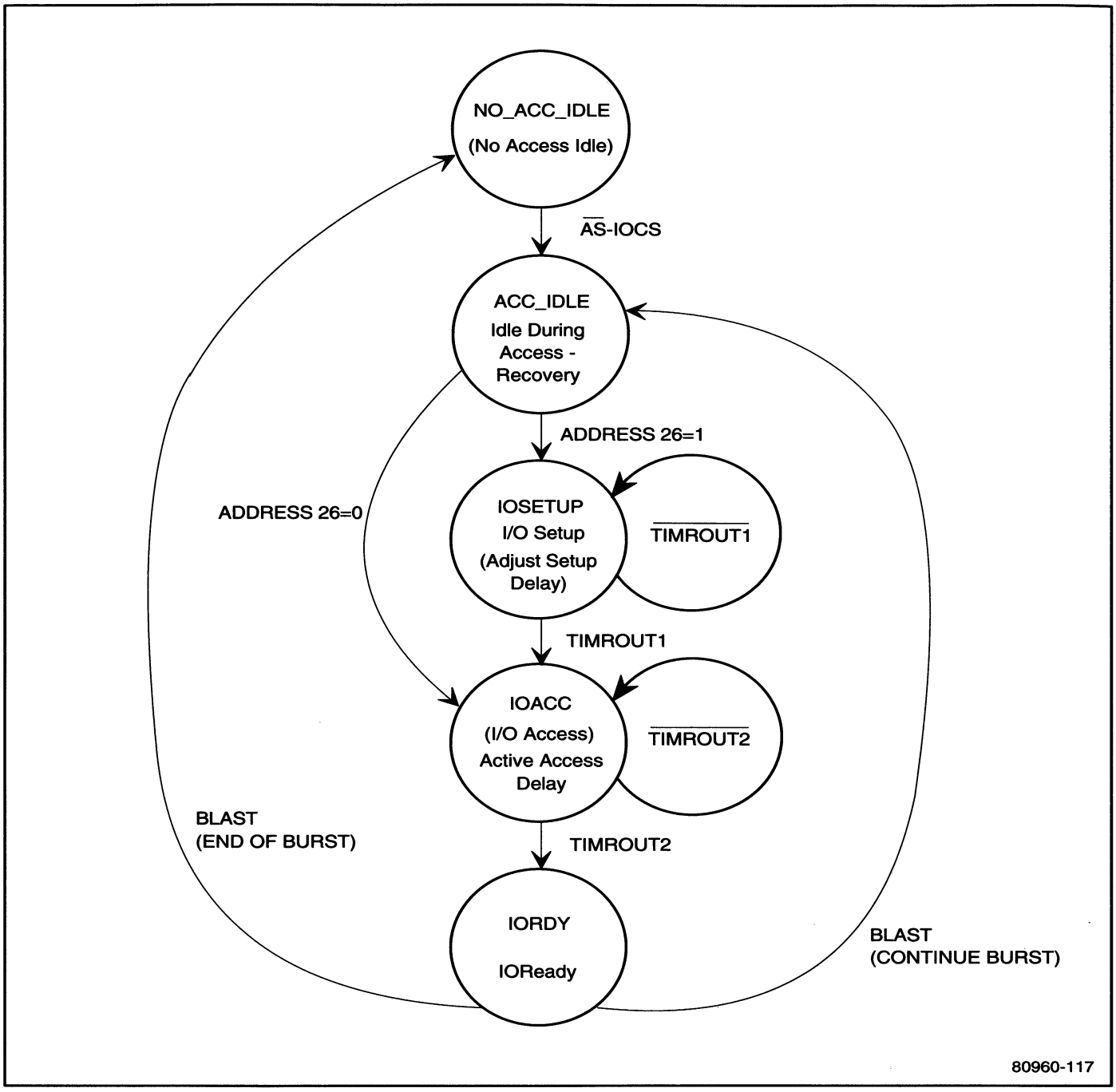

Figure 13-18: I/O State Diagram

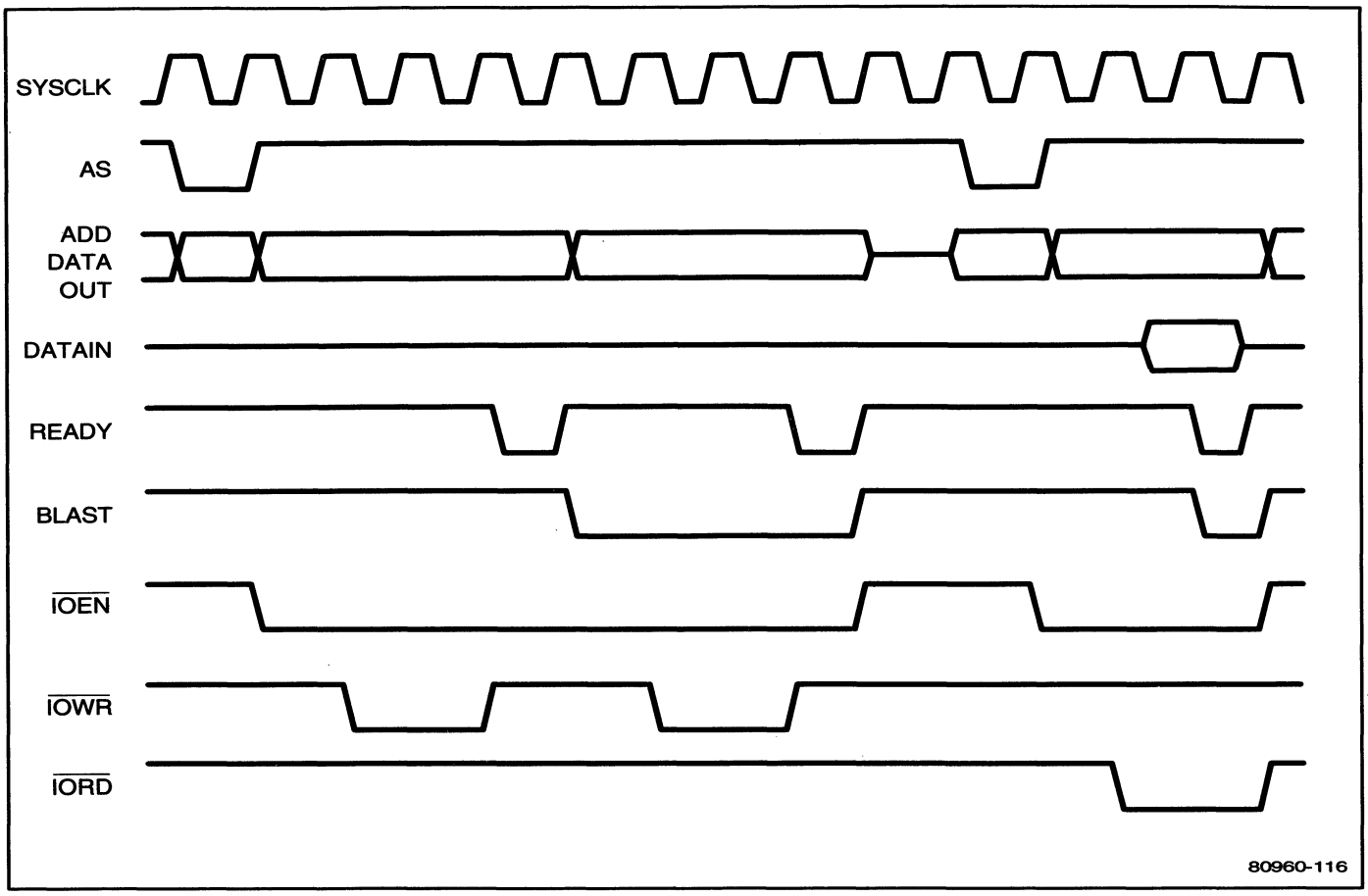

Figure 13-19: I/O Timing Diagram

On the state diagram the top state "No Access Idle" indicates that the I/O is unselected and the CPU is not accessing the I/O drivers controlled by the I/O PLD. The delay counter is reset. The second idle state "Access Idle" occurs when I/O is selected but the read and write strobes are deactivated and the device selects are idle. In this state, the delay counter is also reset. This second idle state guarantees the setup time required by the controlled device between accesses, allowing the device to recover.

The I/O Setup state occurs when an I/O is selected, but the read and write strobes are waiting for a predefined time (depending on the I/O address) before activating the controlled device.This is the signal Timrout1. Whenever Timrout1 deasserts, the state machine proceeds to I/O Access. In this state, I/O is selected and the read and write strobes are active. Timrout2 defines the elapsed time before the I/O is deactivated. This timer also defines the wait states for the read and write pulse widths.

The I/O Ready state deactivates the read and write strobes and returns the READY to the CPU. The state machine returns to the No Access Idle or the Access Idle, depending on the condition of the BLAST signal. If a burst is still in process, the state returns to Access Idle, otherwise the No Access Idle state is selected.

Table 13-1 shows the address, write pulse width and hold time requirements for three typical I/O port devices. These time vary from 0 to 260 ns. These widely varying time requirements argue eloquently for the implementation of a generic I/O controller containing software selectable timing and wait state selection. The alternative is a space consuming collection of individual timing generators.

Table 13-1: Typical I/O Port Device Requirements

Device	T_{add}	T_{wid}	T_{low}
82C54 Timer	30	185	0
Z8536 Parallel Port/Timer	80	240	0
NSC16C552V Serial Port	15	40	0

As an example, the 82C54 Programmable Interval Counter Timer is examined in more detail to illustrate the concept of the way that I/O operations are handled by the 80960SA/SB processor system. The 82C54 contains three independent 16 bit software programmable counter timers. Under software control each of these counters can be programmed for a specific count and a specific mode of operation. There are six modes that the counters can assume:

Mode 0 Interrupt on Terminal Count (Event Counter)

Mode 1 Hardware-retriggerable one-shot

Mode 2 Rate Generator (Divide by N Counter)

Mode 3 Square Wave Counter (Baud Rate Generator)

Mode 4 Software-triggered Strobe

Mode 5 Hardware-triggered Strobe

Each of these modes and the theory of operation for the 82C54 is given in the data sheet for the device. In this section timing and control information relating to the interface with the processor will be given. The illustration in Figure 13-20 shows a block diagram of the 82C54. The device consists of six sub-blocks. The first block is the data bus buffer, a 3 state bidirectional 8-bit buffer that connects the device to the system bus. The read/write logic accepts inputs from the decode and wait generator block and generates internal control signals for the other blocks in the device. The two address inputs select one of the three counters or the control word register. The control word register accepts a control word from the processor that is used to define the operation of one of the counters. The last three blocks are the three counters. Each of the counters is identical with the others, but is independent and each can operate in a different mode.

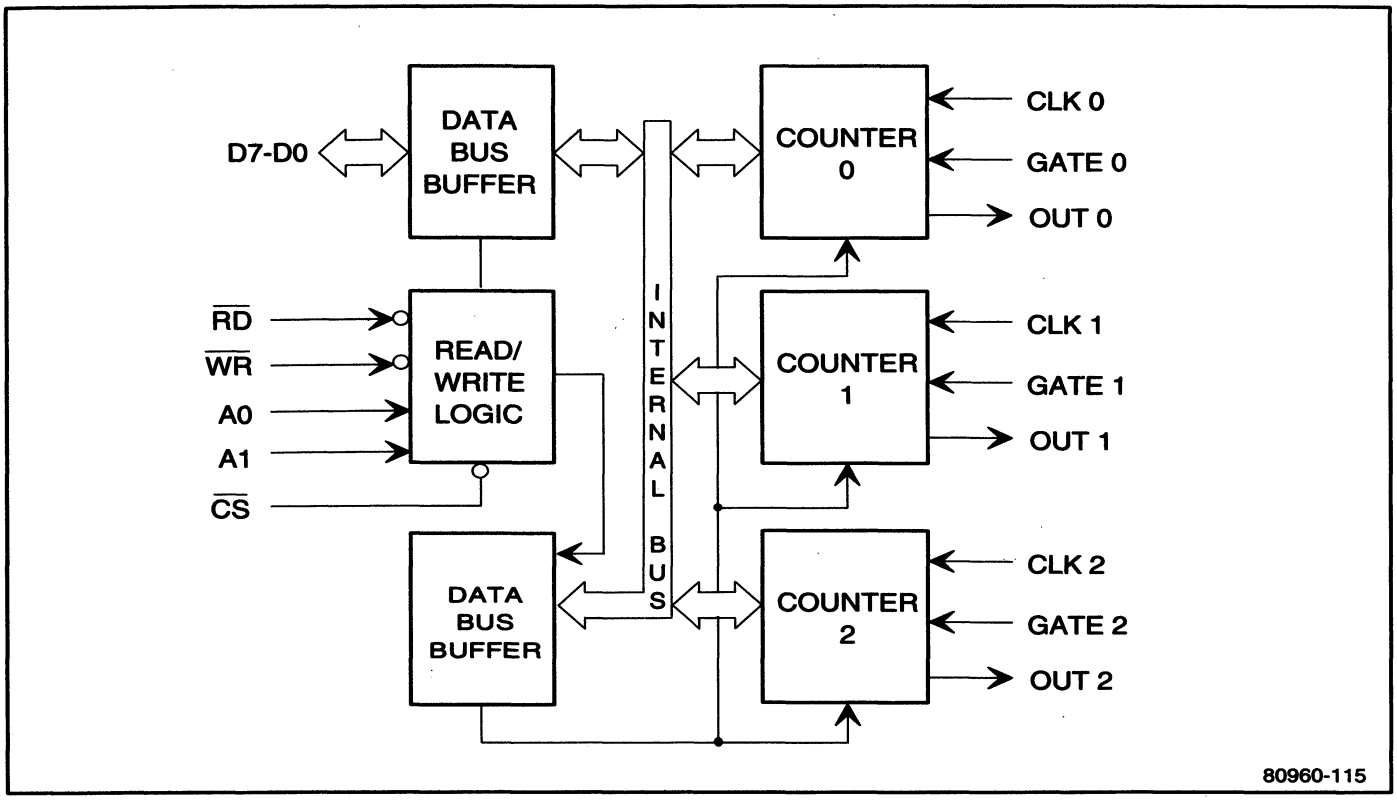

Figure 13-20: Counter/Timer Block Diagram

To use a counter a control word must be written to the control word register. The control word contains a pointer to a specific counter (2 bits), a mode select (3 bits) read/write control (2 bits) and a BCD/binary switch (1 bit). Once the control word has been written, an initial count is written to the selected counter and operation may begin. Figure 13-21 shows the timing diagram for a typical operation.

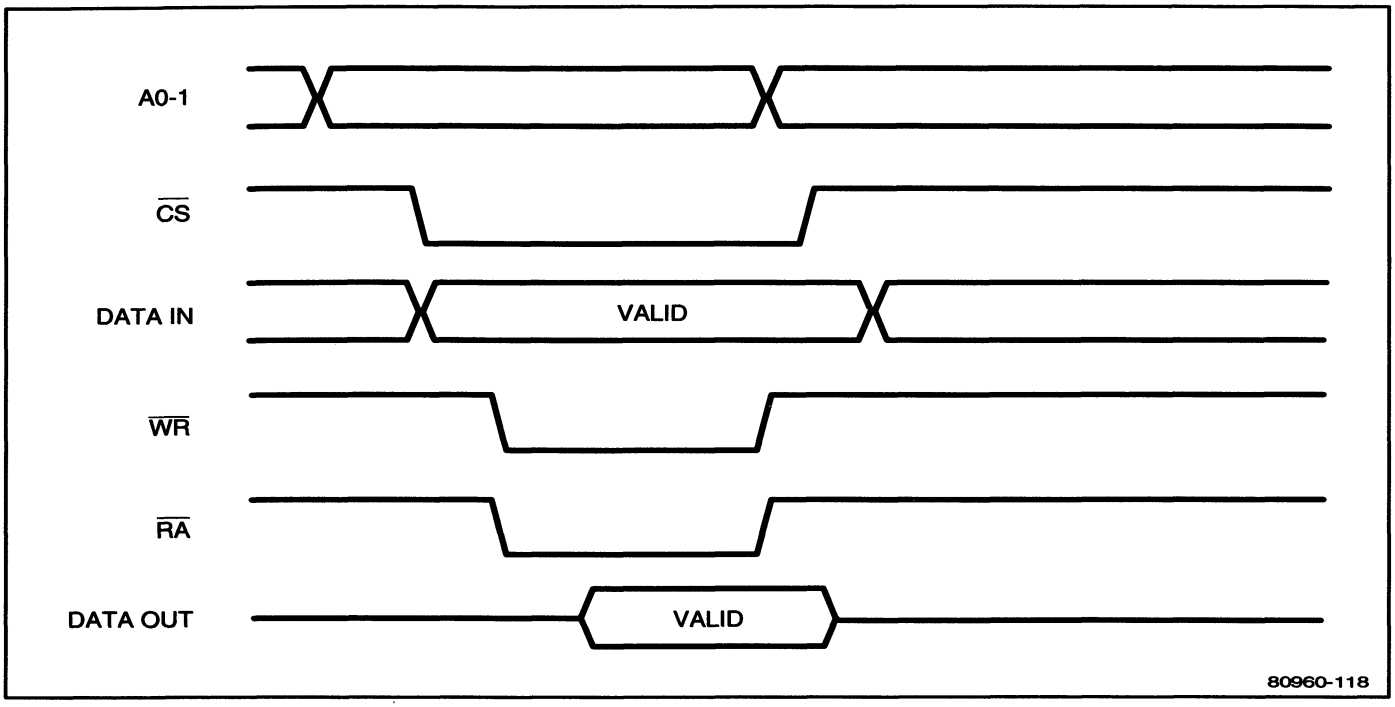

Figure 13-21: Counter/Timer Timing