13 Interfacing with External Memory

CHAPTER 13
INTERFACING WITH EXTERNAL MEMORY

The 8XC196KC/KD can interface with a variety of external memory devices. It supports either a fixed 8-bit bus width or a dynamic 8-bit/16-bit bus width, internal Ready control for slow external memory devices, a bus-hold protocol that enables external devices to take over the bus, and several bus-control modes. These features provide a great deal of flexibility when interfacing with external memory devices. This chapter lists the external memory signals and describes the registers that control the external memory interface and the various external bus modes and features.

13.1. EXTERNAL MEMORY INTERFACE SIGNALS

Table 13-1 describes the external memory interface signals. Many of these signals have alternate functions. When applicable, the "Alternate Functions" column lists the alternate functions and the "Selected by" column lists the register bit and value that selects the function listed in column one.

Table 13-1. External Memory Interface Signals

Function Name	Alternate Functions	Selected by	Type	Description
AD0–AD7 AD8–AD15	P3.0–P3.7 P4.0–P4.7	Bus access to external address	I/O	System Address/Data Bus. These pins provide a multiplexed address and data bus. During the address phase of the bus cycle, address bits 0–15 are output onto the bus and can be latched using ALE or ADV#. During the data phase, 8- or 16-bit data is transferred. When a bus access is not occurring, these pins revert to their I/O port function.
ADV#	ALE	CCR.3 = 0	O	Address Valid. This active-low output signal is asserted only during external memory accesses. ADV# indicates that valid address information is available on the system address/data bus. The signal remains low while a valid bus cycle is in progress and is returned high as soon as the bus cycle completes. An external latch can use this signal to demultiplex the address from the address/data bus. A decoder can also use this signal to generate chip selects for external memory.
ALE	ADV#	CCR.3 = 1	O	Address Latch Enable. This active-high output signal is asserted only during external memory accesses. ALE indicates that valid address information is available on the system address/data bus and signals the start of a valid bus cycle. ALE differs from ADV# in that it does not remain active during the entire bus cycle. An external latch can use this signal to demultiplex the address from the address/data bus.

Table 13-1. External Memory Signals (Continued)

Function Name	Alternate Functions	Selected by	Type	Description
BHE#	WRH#	CCR.2 = 1	O	Byte High Enable. This active-low output signal is asserted only during word writes and high byte writes to external memory. Used in conjunction with A0 to select the memory byte to be written.
BREQ#	P1.5	WSR.7 = 1	O	Bus Request. This active-low output signal is asserted during a HOLD cycle when the bus controller has a pending external memory cycle. The bus controller can assert BREQ# at the same time it asserts HLDA#. BREQ# remains asserted until HOLD# is removed. When this function is active, the pin acts as a standard output (not quasi-bidirectional).
BUSWIDTH	—	CCR.1 = 1	I	Bus Width. When CCR.1 = 1, this signal selects the bus width during external accesses. When high BUSWIDTH selects a 16-bit bus width; when low it selects an 8-bit bus width. When CCR.1 = 0, the bus width is always 8 bits and the BUSWIDTH signal is ignored.
EA#	Programming mode select (EA# = V_{EA})	—	I	External memory access. This active-low signal directs memory accesses to off-chip memory. When high, it selects on-chip OTPROM.
HLDA#	P1.6	WSR.7 = 1	O	Bus Hold Acknowledge. This active-low output indicates that the bus controller has relinquished control of the bus. This occurs in response to an external device asserting the HOLD# signal.
HOLD#	P1.7	WSR.7 = 1	I	Bus Hold. This active-low signal indicates that an external device is requesting control of the bus. When this function is active, the pin acts as a standard input (not quasi-bidirectional).
INST	—	—	O	Instruction Fetch. The signal is valid only during external memory read cycles. When high, INST indicates that an instruction is being fetched; when low, it indicates that data is being read. INST can be used in applications that require separate memory banks for instructions and data. (Note that CCB bytes and interrupt vectors are considered data.)
RD#	—	—	O	External Read. This active-low output signal is the external memory read signal.
READY	—	—	I	Ready. This active-high input signal indicates that external memory is ready to send or receive data.
WR#	WRL#	CCR.2 = 1	O	External Write. This active-low output signal is the external memory write signal.

Table 13-1. External Memory Signals (Continued)

Function Name	Alternate Functions	Selected by	Type	Description
WRH#	BHE#	CCR.2 = 0	O	Write High Byte. In 16-bit bus mode, WRH# is asserted during high byte writes and word writes. In 8-bit bus mode (CCR.1 = 0), WRH# is asserted for high byte, low byte, and word writes.
WRL#	WR#	CCR.2 = 0	O	Write Low Byte. In 16-bit bus mode, WRL# is asserted during low byte writes and word writes. In 8-bit bus mode (CCR.1 = 1), WRL# is asserted for high byte, low byte, and word writes.

13.2. CHIP CONFIGURATION REGISTER

The Chip Configuration Register (CCR) controls the bus width, write strobe signal generation, address valid signal generation, and the number of wait states that can be inserted while the READY pin is held low. It is the first byte fetched from memory after a device reset. Figure 13-1 shows the format of the CCR.

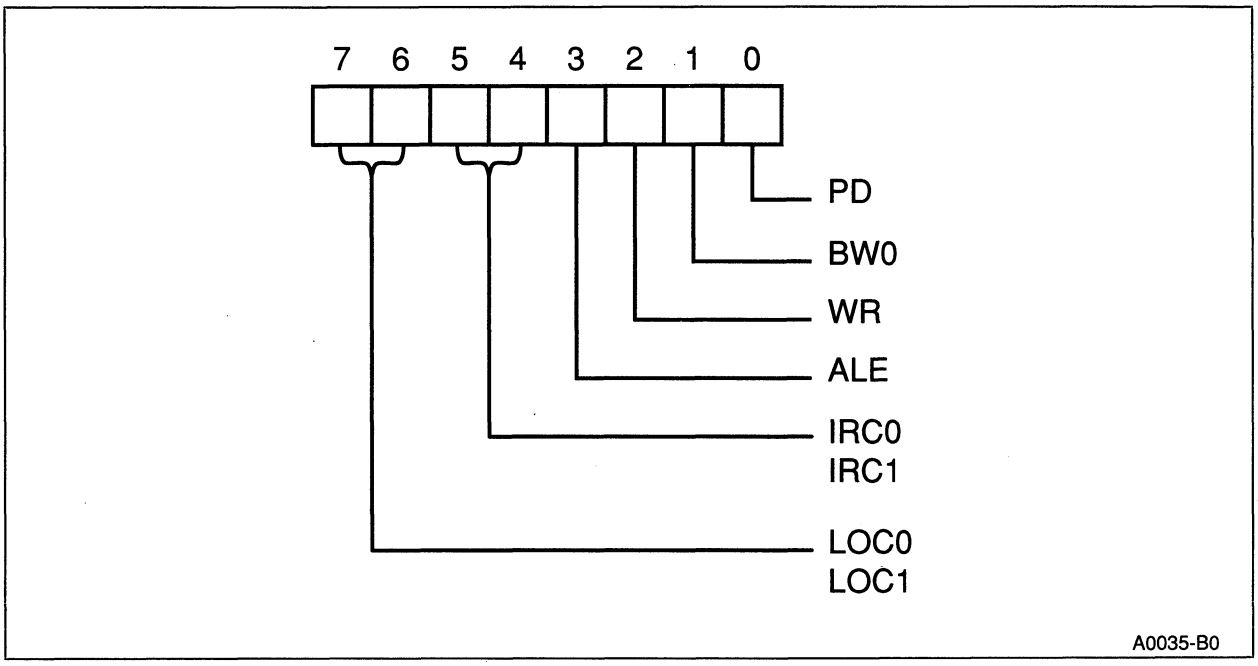

Figure 13-1. Chip Configuration Register

When the device is reset, the bus controller fetches the Chip Configuration Byte (CCB) from location 2018H and loads it into the CCR. The address is always strongly driven during a CCB fetch, just as in normal operation. The CCB is fetched from internal OTPROM if the EA# signal is high and from external memory if the EA# signal is low. The CCR is loaded only during the reset sequence. Once it is loaded, the CCR cannot be changed until the next device reset. Typically, the CCB is programmed once when the user program is compiled and is not redefined during normal operation.

The default setting of the CCR allows the READY and BUSWIDTH signals to control the timing and bus width during the CCB fetch. If the CCB is stored in slow external memory devices, pull the READY signal low. This causes the bus controller to automatically insert up to three wait states into the CCB fetch. (CCR.4 and CCR.5 control the number of wait states; the default configuration specifies a maximum of three wait states.) If the CCB is located in 16-bit external memory, drive the BUSWIDTH signal high; if the CCB is located in 8-bit external memory, drive BUSWIDTH low.

Some early designs may hold the BUSWIDTH pin low (8-bit bus) during the CCB fetch even when 16-bit external memory devices are used. To prevent bus contention, we recommend that location 2019H be loaded with 20H. Under these conditions, the 8XC196KC/KD will strongly drive 20H out onto Port 4 during the entire CCB fetch. If the external memory outputs 20H on its high byte, there will be no bus contention.

After the CCB is loaded into the CCR, the bus width is configured as either fixed 8-bit or dynamically controlled by the BUSWIDTH signal, as specified by CCR.1.

13.3. BUS WIDTH AND MEMORY CONFIGURATIONS

The 8XC196KC/KD external bus can operate as either an 8-bit or 16-bit multiplexed address/data bus (see Figure 13-2). The value of CCR.1 defines the bus width as either a fixed 8-bit bus width or a dynamic 16-bit/8-bit bus width controlled by the BUSWIDTH signal. If CCR.1 is clear, the bus controller is locked into an 8-bit bus mode. Expect some performance degradation when executing code from an 8-bit bus. Word reads and writes to external memory take an extra bus cycle, and the prefetch queue is not fully utilized.

If CCR.1 is set, the BUSWIDTH signal controls the bus width. The bus is 16 bits wide when BUSWIDTH is high and 8 bits wide when BUSWIDTH is low. The BUSWIDTH signal is sampled after the address is on the bus, as shown in Figure 13-3.

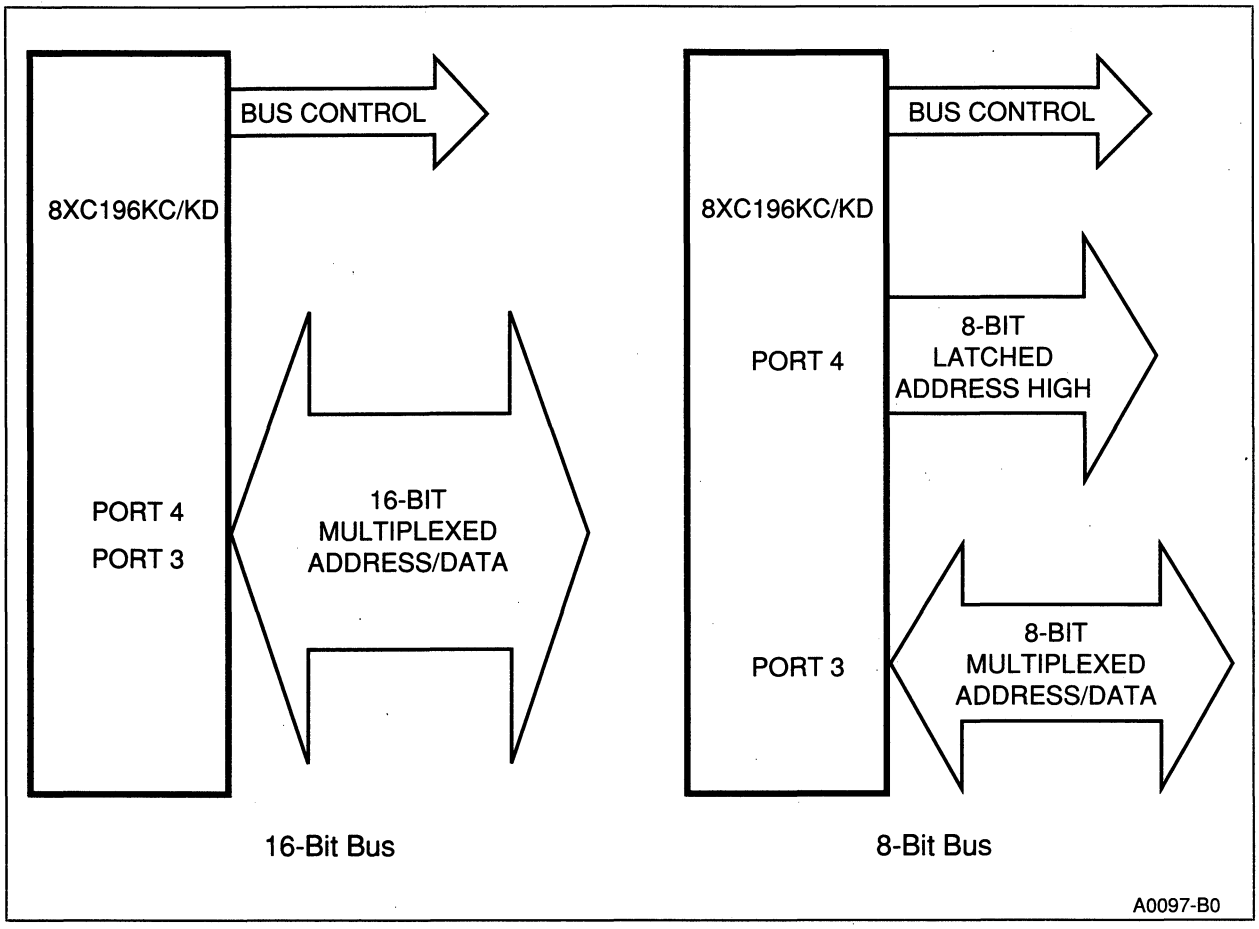

Figure 13-2. Bus Width Options

The BUSWIDTH signal can be used in numerous applications. For example, a system could store code in a 16-bit memory device and data in 8-bit memory device. The BUSWIDTH signal could be tied to the chip-select input of the 8-bit memory device as shown in Figure 13-11 on page 13-16. When BUSWIDTH is low, it enables 8-bit bus mode and selects the 8-bit memory device. When BUSWIDTH is high, it enables 16-bit bus mode and deselects the 8-bit memory device.

13.3.1. Timing Requirements for BUSWIDTH

When using BUSWIDTH to dynamically change between 8-bit and 16-bit bus widths, setup and hold timings must be met for proper operation (see Figure 13-3). Because a decoded, valid address is used to generate the BUSWIDTH signal, the setup time is specified relative to the address being valid. This specification, T_{AVGV}, indicates how much time one has to decode the valid address and generate a valid BUSWIDTH signal.

BUSWIDTH must be held valid until the minimum hold specification, T_{CLGX}, has been met. Typically this hold time is 0 ns minimum after CLKOUT goes low. In all cases, refer to the data sheet for current specifications for T_{AVGV} and T_{CLGX}.

NOTE

Earlier HMOS devices used a BUSWIDTH setup timing that was referenced to the falling edge of ALE (T_{LLGV}). This specification is not meaningful for CMOS devices, which use an internal two-phase clock; it is included for comparison only.

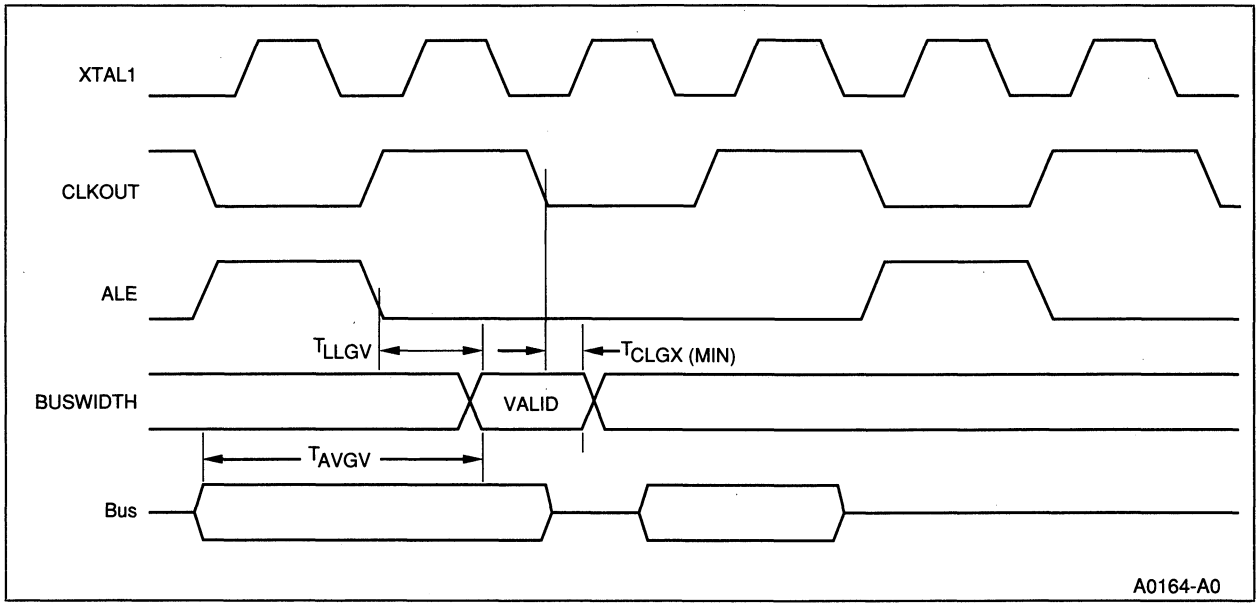

Figure 13-3. BUSWIDTH Timing Diagram

13.3.2. 16-Bit Bus Width

When the 8XC196KC/KD is configured to operate in the 16-bit bus-width mode (that is, when CCR.1 is set and the BUSWIDTH signal is high), lines AD0–AD15 form a 16-bit multiplexed address/data bus. The address/data bus shares pins with Ports 3 and 4 (see Table 13-1 on page 13-1). Figure 13-4 shows an idealized timing diagram for the external read and write cycles. See the data sheet for a list of specifications that must be met by external memory devices.

Address Latch Enable (ALE) demultiplexes the address bus by strobing transparent latches (such as a 74AC373). The address ("Address Out" on Bus) will be valid on the bus before ALE drops.

13.3.2.1. 16-BIT READ CYCLES

The bus controller floats the bus and then drives RD# low so that it can receive data. The external memory must put data ("Data In") onto the bus before the rising edge of RD#. The data sheet specifies the maximum time the memory device has to output valid data after RD# is asserted. When INST is asserted, it indicates that the read operation is an instruction fetch.

13.3.2.2. 16-BIT WRITE CYCLES

The bus controller drives WR# low, then puts data onto the bus. The rising edge of WR# signifies that data is valid. At this time, the external system must latch the data.

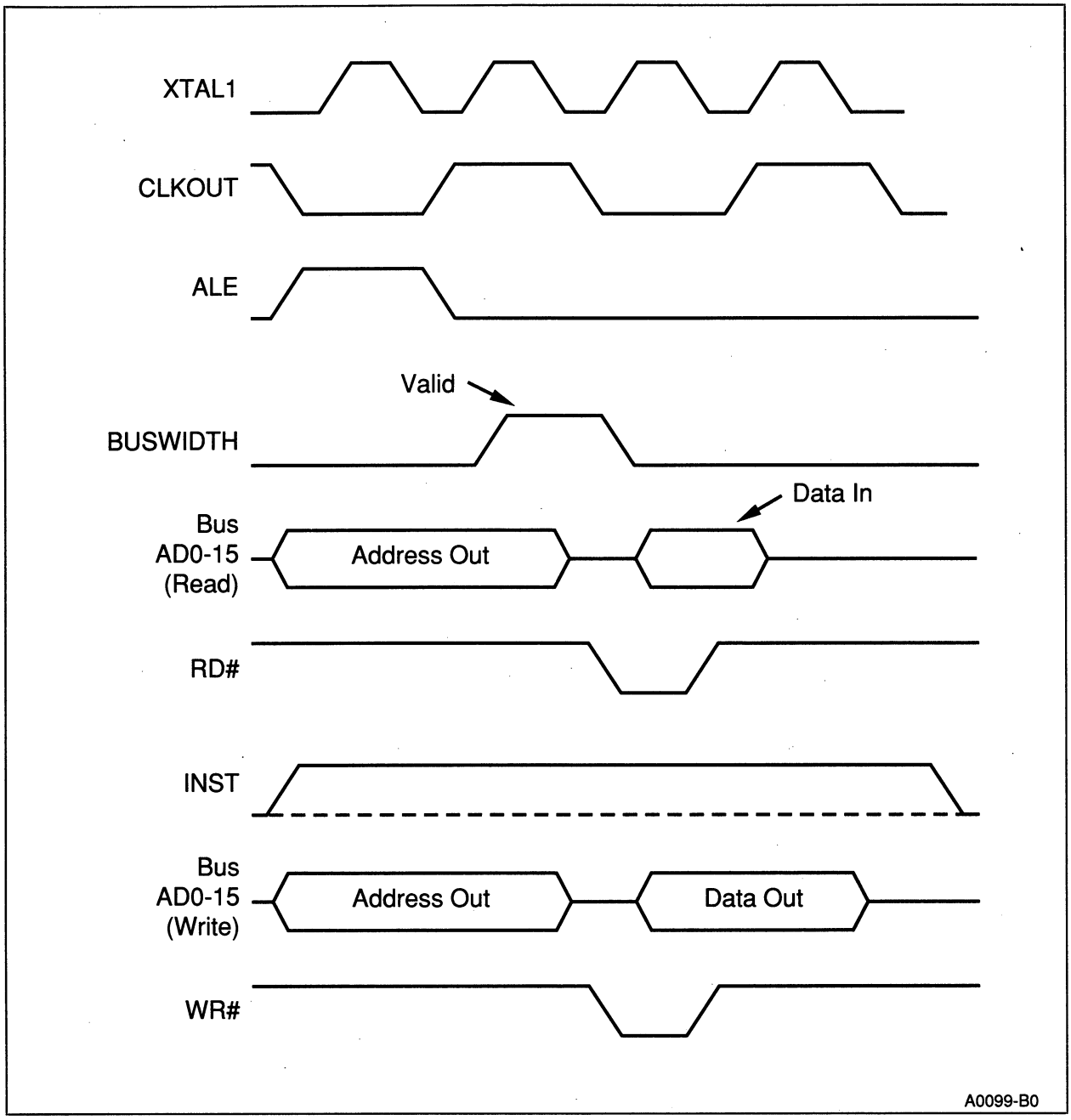

Figure 13-4. Idealized Bus Timings for 16-Bit External Bus

13.3.3. 8-Bit Bus Width

When the 8XC196KC/KD is configured to operate in the 8-bit bus mode, lines AD0–AD7 form a multiplexed lower address and data bus. Lines AD8–AD15 are not multiplexed; the upper address is latched and remains valid throughout the bus cycle. Figure 13-5 shows an idealized timing diagram for the external read and write cycles.

The ALE signal is used to demultiplex the lower address by strobing a transparent latch (such as a 74AC373).

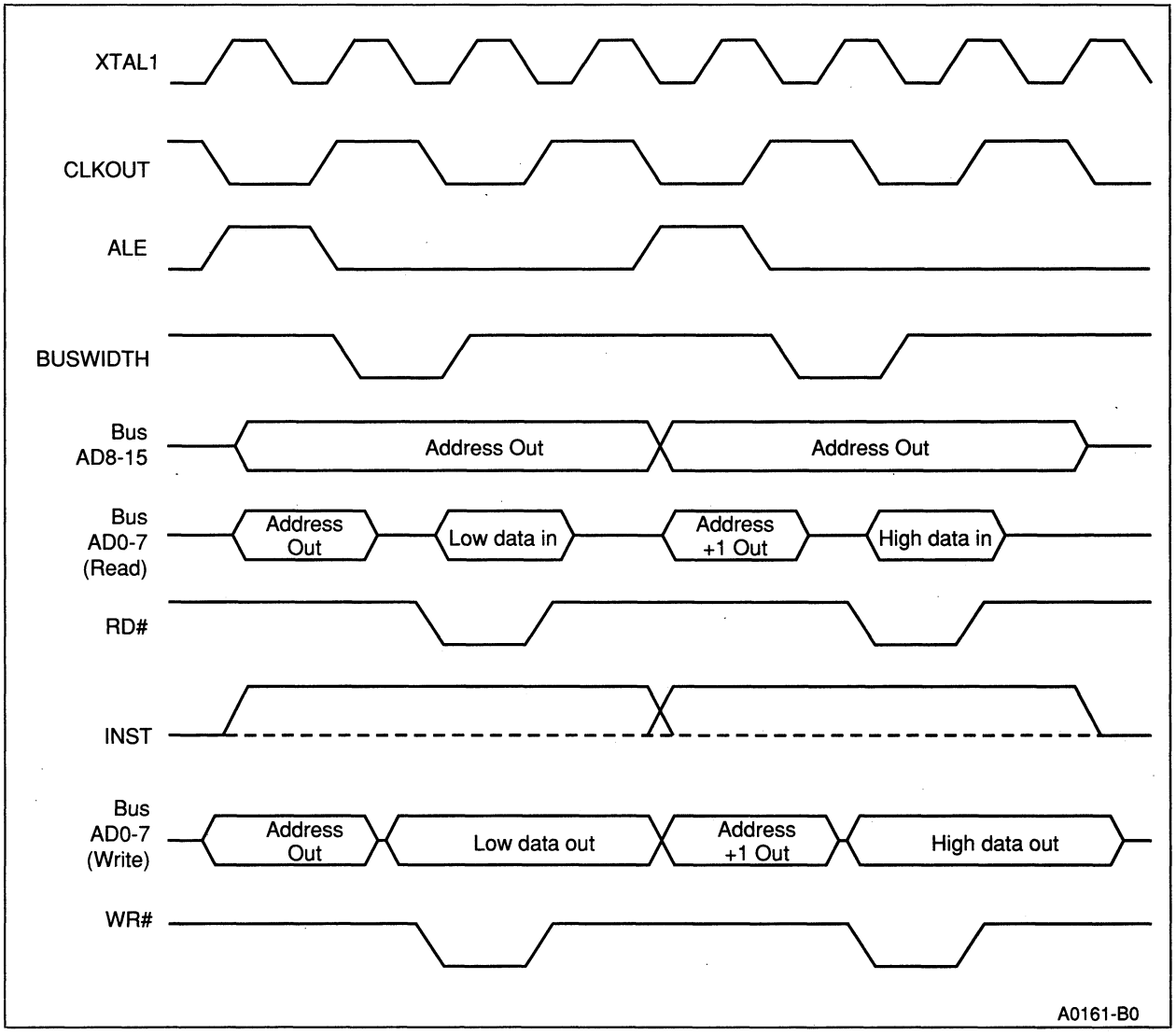

Figure 13-5. Idealized Bus Timings for 8-Bit External Bus

13.3.3.1. 8-BIT BUS READ CYCLES

After ALE falls, the bus controller floats the bus and drives the RD# signal low. The external memory then must put its data on the bus. That data must be valid at the rising edge of the RD# signal. To read a data word, the bus controller performs two consecutive reads, reading the low byte first, followed by the high byte.

13.3.3.2. 8-BIT BUS WRITE CYCLES

After ALE falls, the bus controller outputs data on AD0–7 and then drives WR# low. The external memory must latch the data by the time WR# goes high. That data will be valid on the bus until slightly after WR# goes high. To write a data word, the bus controller performs two consecutive writes, writing the low byte first, followed by the high byte.

13.4. READY CONTROL

The internal Ready control circuitry allows slow external memory devices to increase the length of the 8XC196KC/KD's read and write bus cycles. If the external memory device is not ready for access, it pulls the READY signal low and holds it low until it is ready to complete the operation. While READY is low, the bus controller inserts wait states into the bus cycle.

CCR bits 4 and 5 (IRC0 and IRC1) define the maximum number of wait states that will be inserted (see Table 13-2). When both bits are set, the bus controller inserts wait states until the external memory device asserts the READY signal. Otherwise, the bus controller inserts wait states until either the external memory device asserts the READY signal or the number of wait states equals the number (1, 2, or 3) defined by CCR.4 and CCR.5.

Table 13-2. Wait State Control

CCR.5	CCR.4	Maximum Wait States
0	0	Limit to 1
0	1	Limit to 2
1	0	Limit to 3
1	1	Infinite (controlled by READY signal)

When using the READY signal to control the number of wait states, be sure to add external hardware to count wait states and release READY within a specified period of time. (See the data sheet for READY specifications.) Otherwise, a defective memory device could tie up the address/data bus indefinitely.

NOTE

Ready control is valid only for external memory; you cannot add wait states when accessing internal RAM and OTPROM space.

13.4.1. Timing Requirements for READY

Setup and hold timings must be met when using the READY signal to insert wait states into a bus cycle (see Figure 13-6). Because a decoded, valid address is used to generate the READY signal, the setup time is specified relative to the address being valid. This specification, T_{AVYV} , indicates how much time one has to decode and assert READY from when the address is valid. The READY signal must be held valid until the T_{CLYX} timing specification is met. Typically, this is a minimum of 0 ns from the time CLKOUT goes low. Do not exceed the maximum T_{CLYX} specification or additional (unwanted) wait states might occur. In all cases, refer to the data sheets for the current specifications for T_{AVYV} and T_{CLYX}.

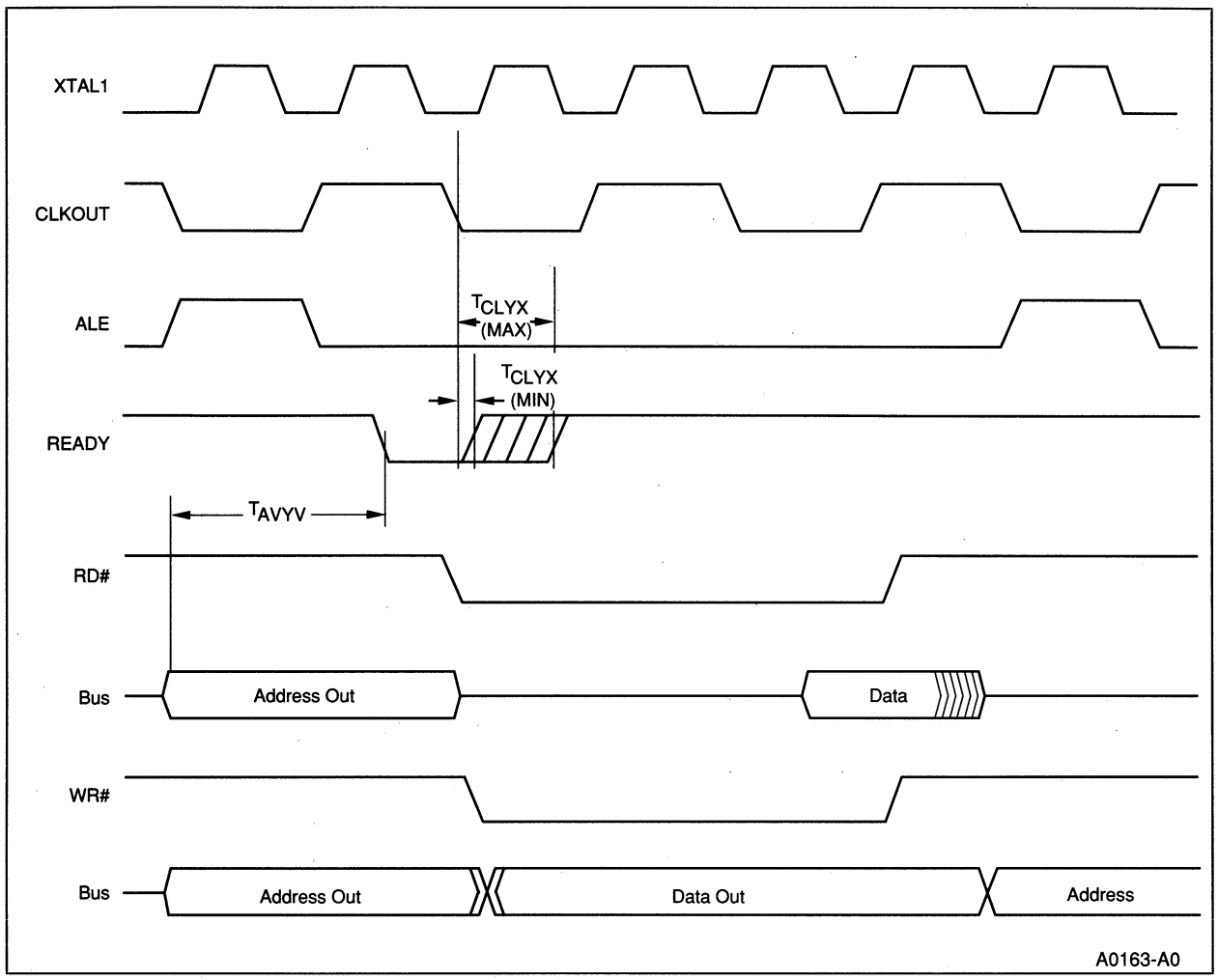

Figure 13-6. READY Timing Diagram

When the CCR is programmed for limited wait states (i.e., 1, 2, or 3 maximum), you may want to tie READY directly to the chip enable of slow external memory device. This is a simple way to insert 1–3 wait states when the device is selected.

NOTE

Earlier HMOS devices specified a READY setup time that was referenced to the falling edge of ALE (T_{LLYV}). This specification is not meaningful for CMOS devices, which use an internal 2-phase clock; it is included for comparison only.

13.5. BUS-HOLD PROTOCOL

The 8XC196KC/KD supports a bus-hold protocol that allows external devices to gain control of the address/data bus. The protocol uses three signals, HOLD# (Hold Request), HLDA# (Hold Acknowledge), and BREQ# (Bus Request), which are Port 1 alternate functions. When a device wants to use the 8XC196KC/KD bus, it asserts the HOLD# signal. HOLD# is sampled while CLKOUT is low. The 8XC196KC/KD responds by releasing the bus and asserting HLDA#. The address/data bus floats and ALE, RD#, WR#, and BHE# are weakly held in their inactive states during this hold time. Figure 13-7 shows the timing for bus-hold protocol. Refer to the data sheet for specific timing requirements.

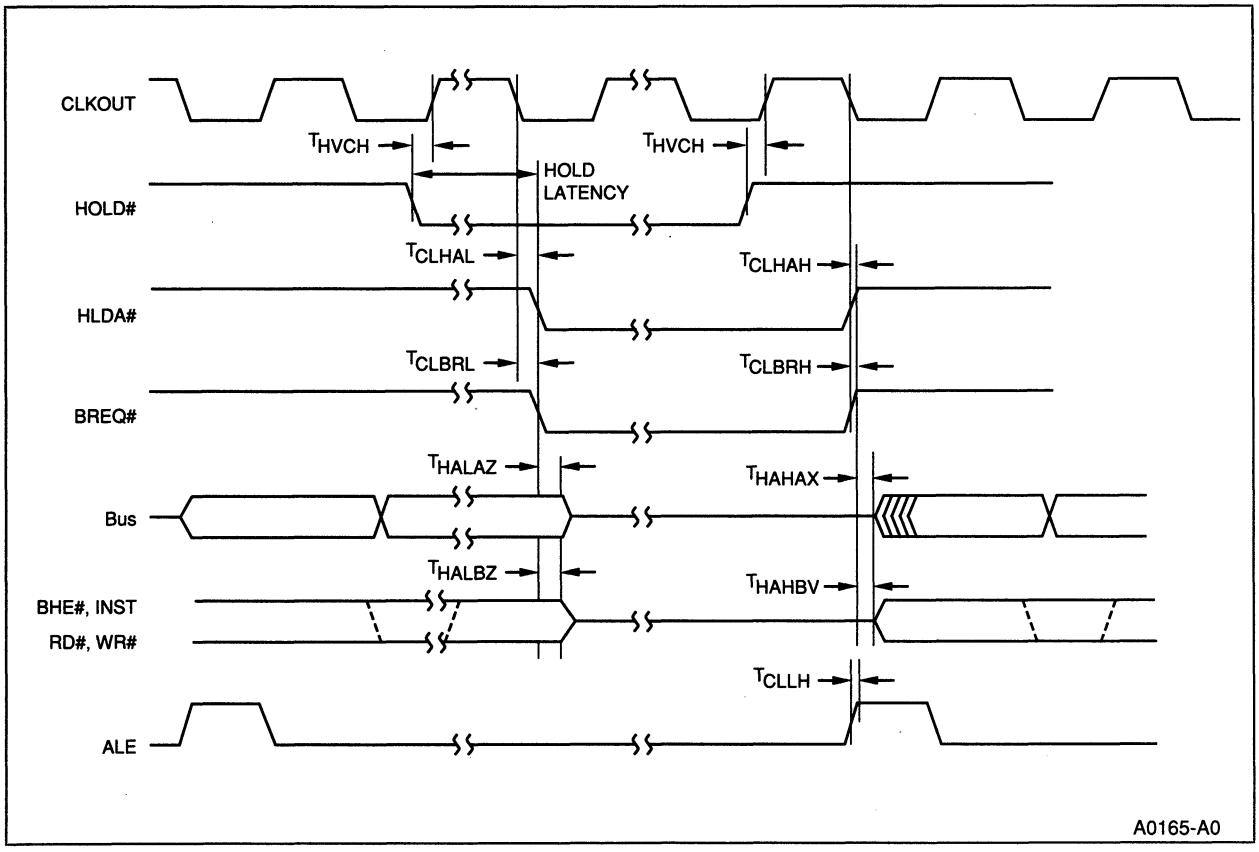

Figure 13-7. HOLD#, HLDA# Timings

When the external device is finished with the bus, it relinquishes control by driving HOLD# high. In response, the 8XC196KC/KD drives HLDA# high and assumes control of the bus. While the 8XC196KC/KD is in hold, it can request control of the bus by asserting BREQ#. After the external device responds by driving HOLD# high, the 8XC196KC/KD exits hold and then deasserts BREQ# and HLDA#.

13.5.1. Enabling the Bus-Hold Protocol

The HLDEN bit in the Window Select register (WSR.7) enables the bus-hold protocol. Setting this bit enables the protocol and selects the alternate functions of port pins P1.7 (HOLD#), P1.6 (HLDA#), and P1.5 (BREQ#). Once the alternate functions are enabled, the pins act as standard (not quasi-bidirectional) inputs or outputs. Once the bus-hold protocol has been selected, the port function cannot be reselected without resetting the device. However, the hold function can be dynamically enabled and disabled as described below.

13.5.2. Disabling the Bus-Hold Protocol

To disable hold requests, clear WSR.7. The 8XC196KC/KD does not take over the bus immediately after HLDEN is cleared. Instead, it waits for the current HOLD# request to finish and then disables the bus-hold feature and ignores any new requests until the bit is set again.

Sometimes it is important to prevent another device from taking control of the bus while a block of code is executing. One way to protect a block of code is to clear WSR.7, and then execute a JBC instruction to check the status of the HLDA# signal. The JBC instruction prevents the RALU from executing the protected block until current HOLD# requests are serviced and the hold feature is disabled.

```
        DI                          ;Disable interrupts to prevent
                                    ;code interruption
        ANDB WSR, #7FH              ;Disable hold requests
WAIT:   JBC IOPORT1,6, WAIT         ;Check the HLDA# signal
                                    :If set, execute
                                    ;protected instruction

        ORB WSR,#80H                ;Enable hold requests
        EI                          :Enable interrupts
```

13.5.3. Hold Latency

When an external device asserts HOLD#, the 8XC196KC/KD finishes the current bus cycle and then asserts HLDA#. The time it takes the 8XC196KC/KD to assert HLDA# after the external device asserts HOLD# is called *hold latency* (see Figure 13-7). Table 13-3 lists the maximum hold latency for each type of bus cycle.

Table 13-3. Maximum Hold Latency

Bus Cycle Type	Maximum Hold Latency
Internal Execution or Idle Mode	1.5 State Times
16-Bit External Execution	2.5 State Times
8-Bit External Execution	4.5 State Times

13.5.4. Regaining Bus Control

While HOLD# is asserted, the 8XC196KC/KD can execute code out of internal memory. When it needs to access external memory, it asserts BREQ# and waits for the external device to deassert HOLD#. After asserting BREQ#, the 8XC196KC/KD cannot respond to any interrupt requests, including NMI, until the external device deasserts HOLD#. One state time after HOLD# goes high, the 8XC196KC/KD deasserts HLDA# and resumes control of the bus.

If the 8XC196KC/KD is reset while in hold, bus contention can occur. For example, a CPU-only device would try to fetch the Chip Configuration Byte from external memory after RESET# is brought high. Bus contention would occur because both the external device and the 8XC196KC/KD would attempt to access memory. One solution is to use the RESET# signal as the system reset; then all bus masters (including the 8XC196KC/KD) are reset at once. Chapter 11, "Minimum Hardware Considerations," shows system reset circuit examples.

13.6. BUS-CONTROL MODES

CCR.2 and CCR.3 define which bus-control signals will be generated during external read and write cycles. Table 13-4 lists the four bus-control modes and shows the CCR.3 and CCR.2 settings for each.

Table 13-4. Bus-Control Mode

Bus-Control Mode	Bus-Control Signals	CCR.3	CCR.2
Address Valid with Write Strobe mode	ADV#, RD#, WRL#, WRH#	0	0
Address Valid Strobe mode	ADV#, RD#, WR#, BHE#	0	1
Write Strobe mode	ALE#, RD# WRL#, WRH#	1	0
Standard Bus-Control mode	ALE, RD#, WR#, BHE#	1	1

13.6.1. Standard Bus-Control Mode

In the standard bus-control mode, the 8XC196KC/KD generates the standard bus-control signals: ALE, WR#, RD# and BHE# (see Figure 13-8). ALE is asserted while the address is driven, and it can be used to latch the address externally. When low, BHE# selects the bank of memory that is addressed by the high byte of the data bus. RD# is asserted for every external memory read, and WR# is asserted for every external memory write.

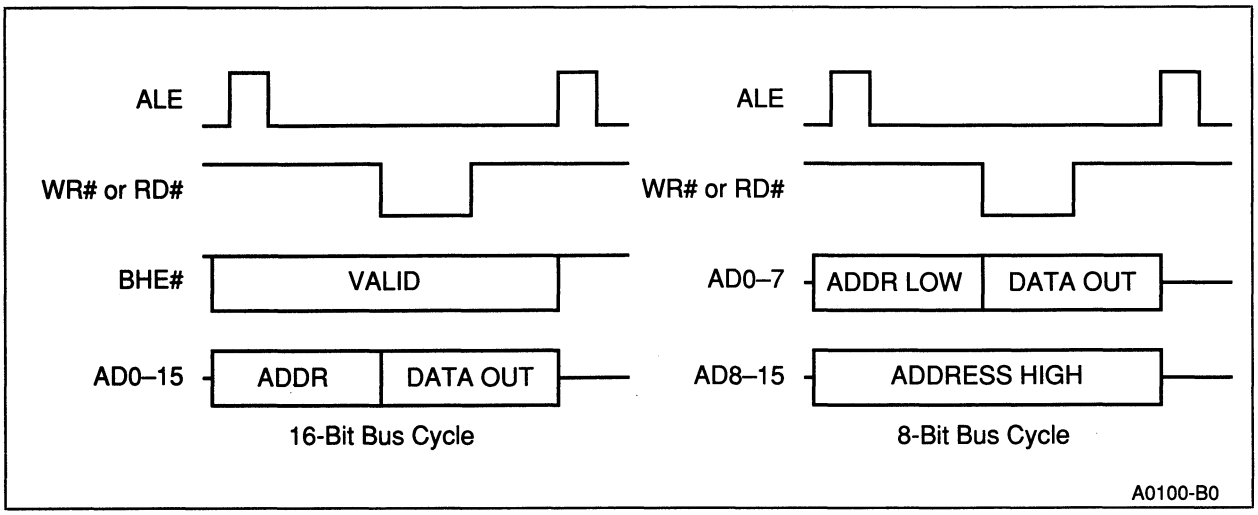

Figure 13-8. Standard Bus Control

When the device is configured to use a 16-bit bus, separate low- and high-byte write signals must be generated for single-byte writes. Figure 13-9 shows a sample circuit that combines BHE# and A0 to produce these signals (WRL# and WRH#). A similar pair of signals for read is unnecessary. For a single-byte read with the 16-bit bus, both bytes are put onto the data bus and the processor discards the unwanted byte.

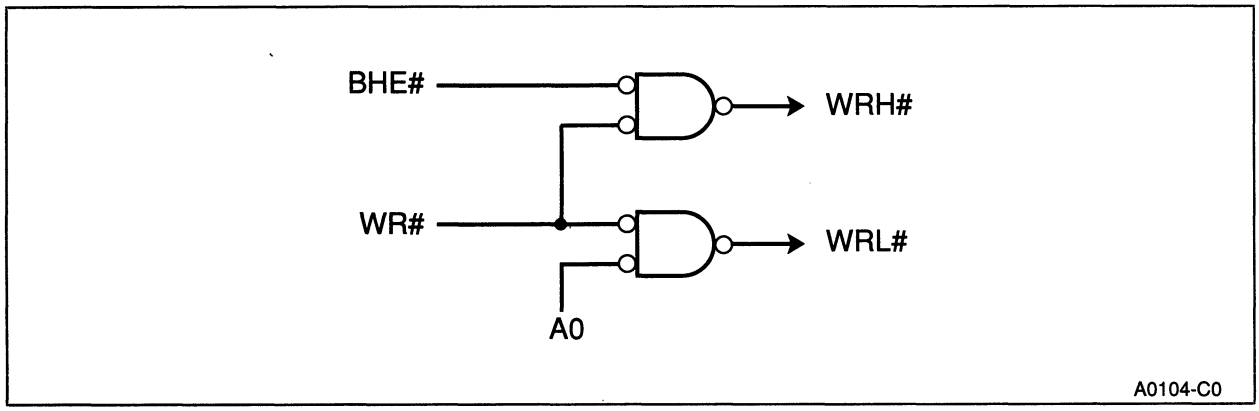

Figure 13-9. Decoding WRL# and WRH#

Figure 13-10 shows an 8-bit system with both EPROM and RAM. The EPROM is the lower half of memory, and the RAM is the upper half. This system configuration uses the most-significant address bit (A15) as the chip-select signal and ALE as the address-latch signal.

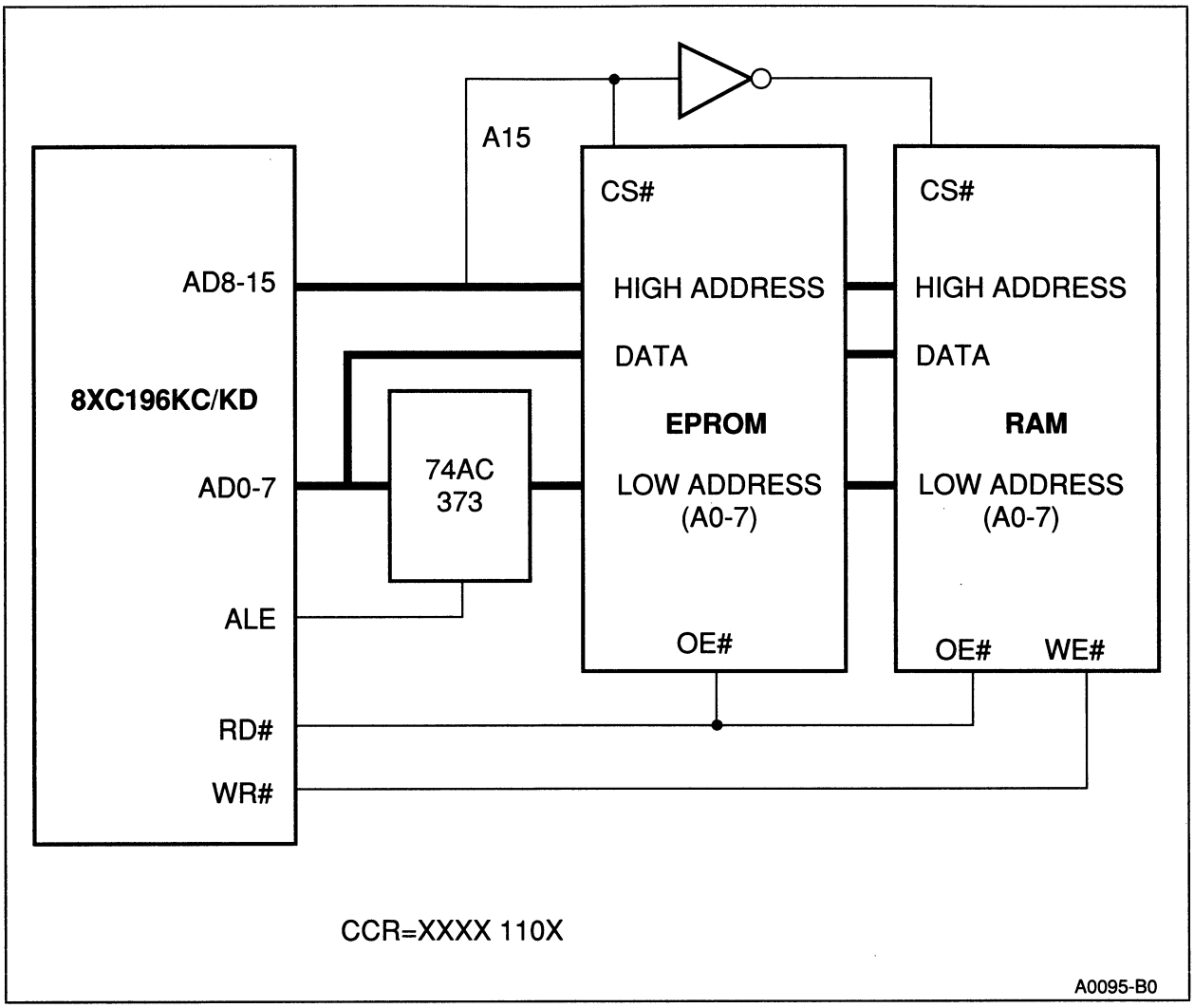

Figure 13-10. 8-Bit System with EPROM and RAM

Figure 13-11 shows a system that uses the dynamic bus-width feature (CCR.1 set). Code is executed from the two EPROMs and data is stored in the byte-wide RAM. The RAM is the high memory and is selected by driving A15 high, which also selects the 8-bit bus width mode by driving the BUSWIDTH signal low.

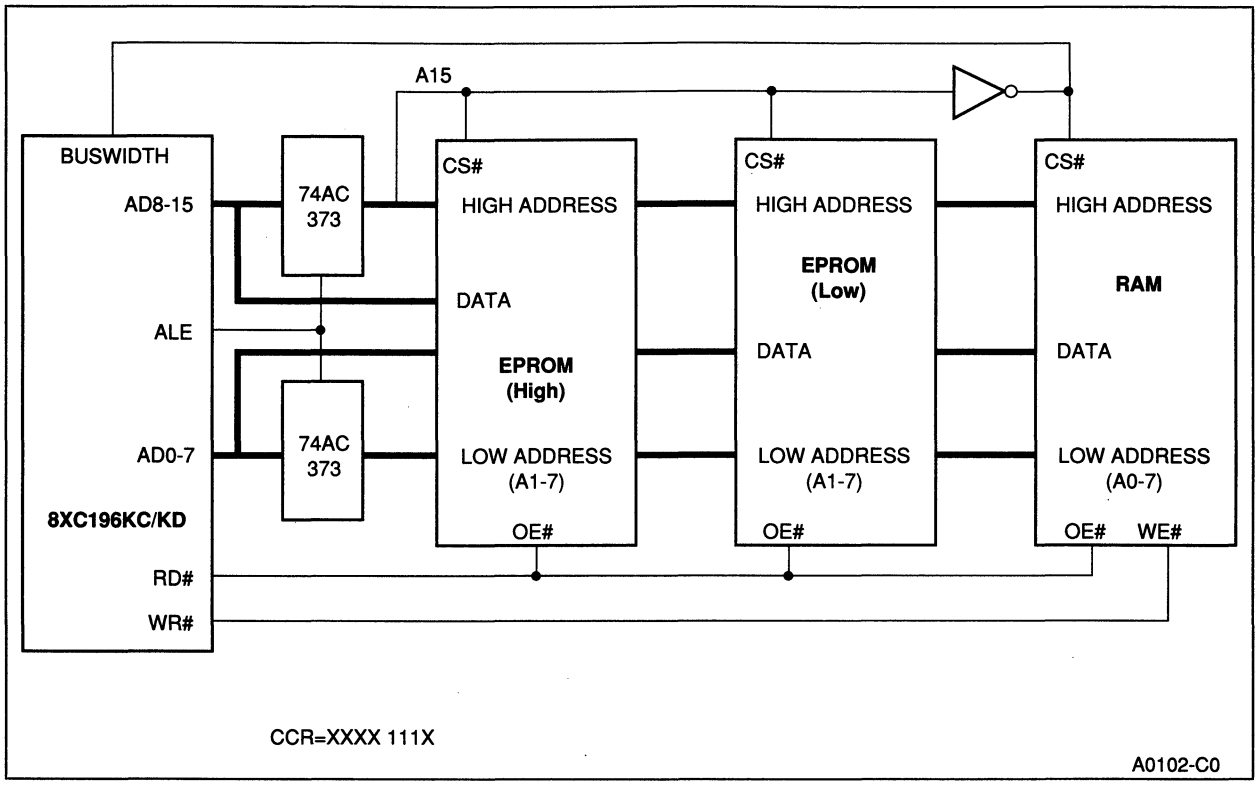

Figure 13-11. 16-Bit System with Dynamic Bus Width

13.6.2. Write Strobe Mode

The Write Strobe mode eliminates the need to externally decode high- and low-byte writes to external 16-bit RAM in 16-bit bus mode. When the Write Strobe mode is selected, the 8XC196KC/KD generates WRL# and WRH# instead of WR# and BHE#. WRL# is asserted for all low byte writes (even addresses) and all word writes. WRH# is asserted for all high byte writes (odd addresses) and all word writes. In the 8-bit bus mode, WRH# and WRL# are asserted for both even and odd addresses. The Write Strobe mode timing is shown in Figure 13-12.

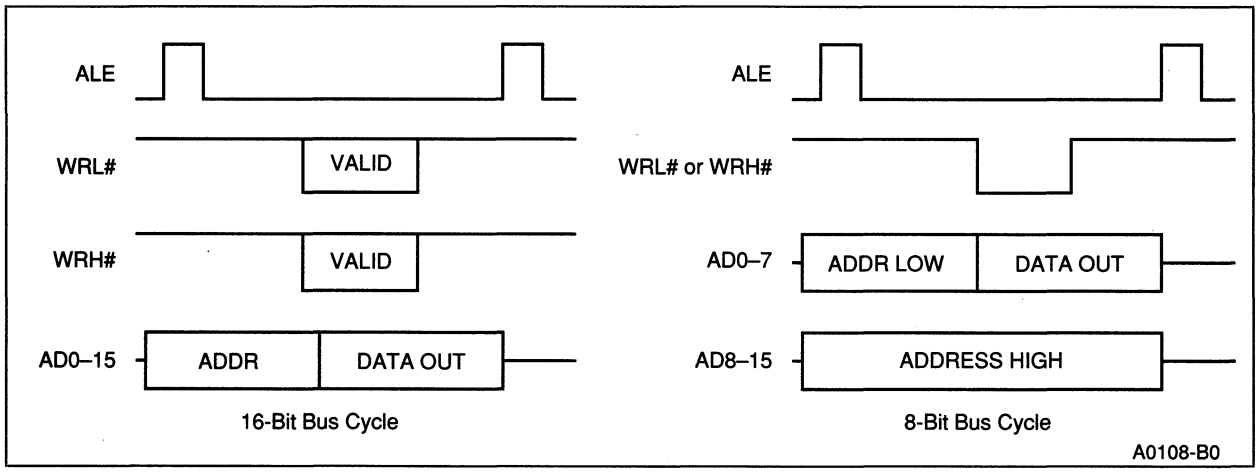

Figure 13-12. Write Strobe Mode

Figure 13-13 shows a 16-bit system that uses two EPROMs and two RAMs. This example is configured to use the Write-Strobe mode. ALE latches the address, and A15 is the chip select for the EPROMs and RAMs. WRL# is asserted during low byte writes and word writes. WRH# is asserted during high byte writes and word writes. Note that RAM devices do not use A0. WRL# and WRH# determine whether the low byte (A0 = 0) or high byte (A0 = 1) is selected.

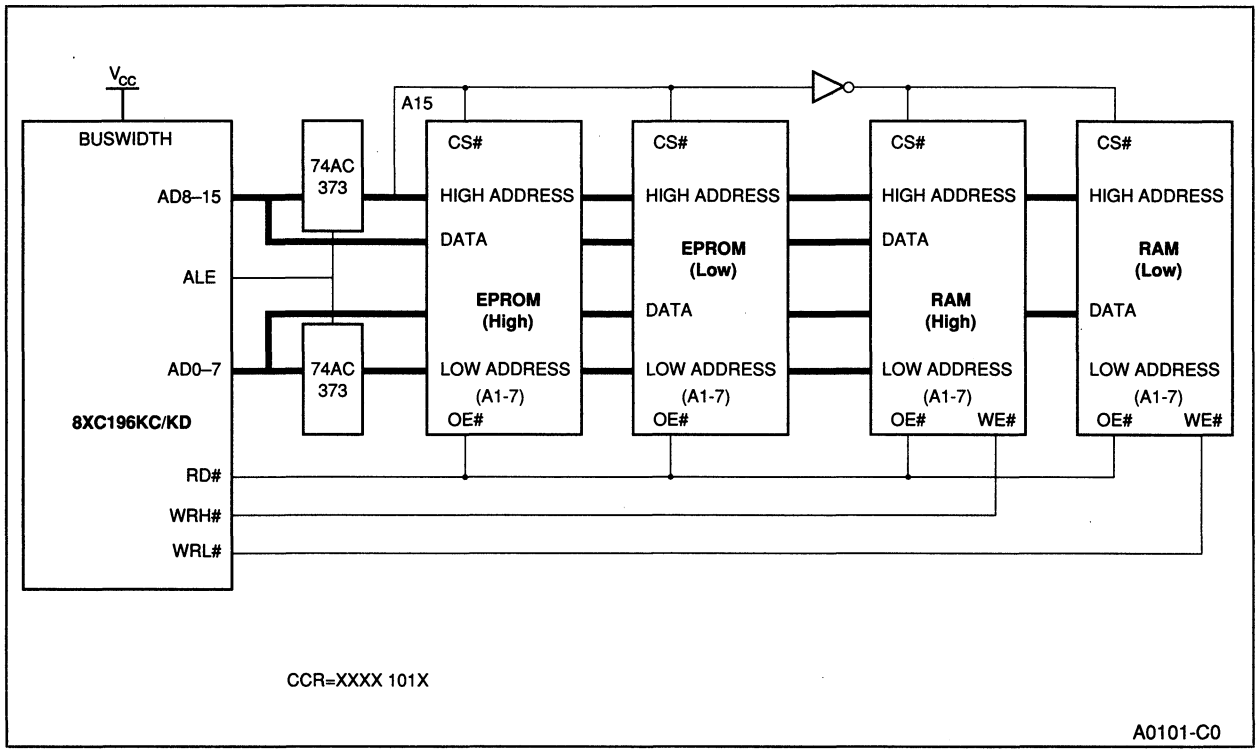

Figure 13-13. 16-Bit System with Single-Byte Writes to RAM

13.6.3. Address Valid Strobe Mode

When the Address Valid Strobe mode is selected, the 8XC196KC/KD generates the Address Valid signal (ADV#) instead of the Address Latch Enable signal (ALE). ADV# is asserted after an external address is valid (see Figure 13-14). This signal can be used to latch the valid address and simultaneously enable an external memory device.

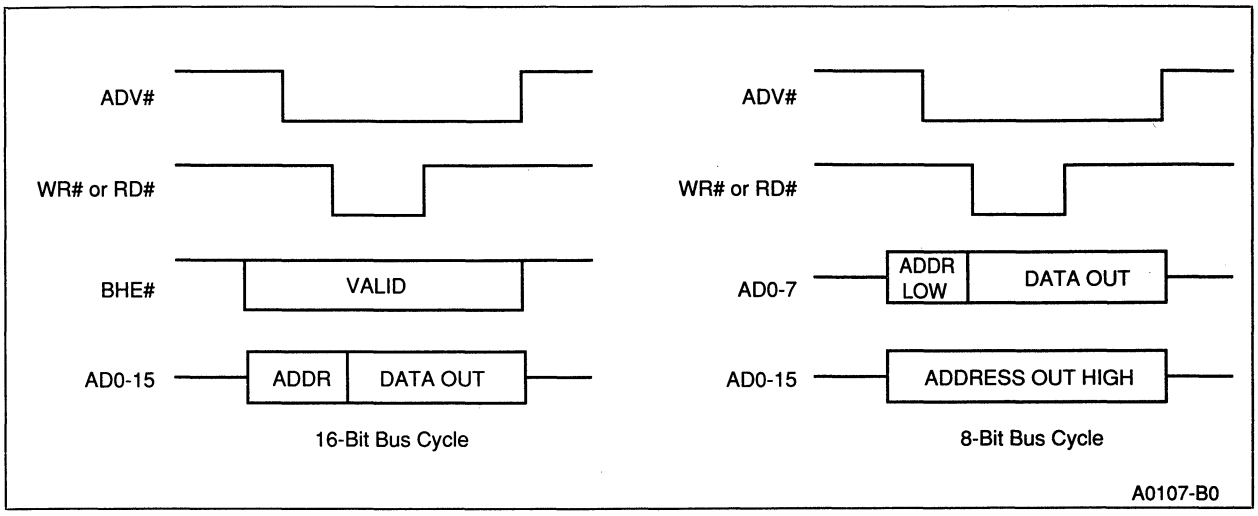

Figure 13-14. Address Valid Strobe Mode

The difference between ALE and ADV# is that ADV# is asserted for the entire bus cycle, not just to latch the address. Figure 13-15 shows the difference between ALE and ADV# for a single read or write cycle. Note that for back-to-back bus access, the ADV# function will look identical to the ALE function. The difference becomes apparent only when the bus is idle. Because ADV# is high during these periods, external memory will be disabled, thus saving power. Figures 13-16 and 13-17 show sample circuits that use Address Valid Strobe Mode.

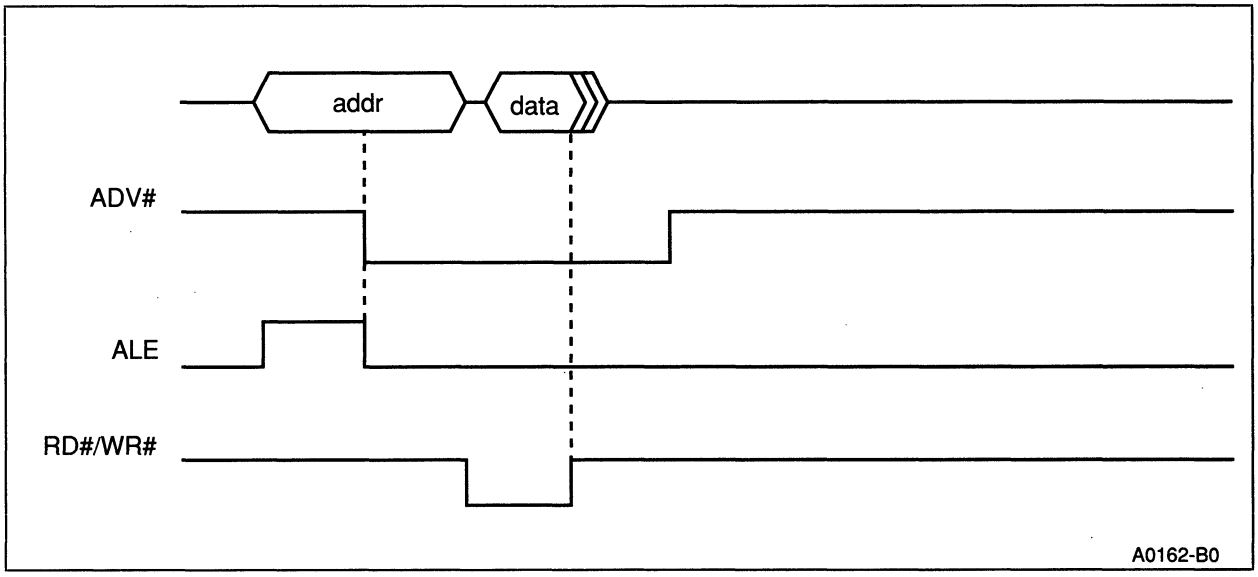

Figure 13-15. Comparison of ALE and ADV# Bus Cycles

Figure 13-13 shows a 16-bit system that uses two EPROMs and two RAMs. This example is configured to use the Write-Strobe mode. ALE latches the address, and A15 is the chip select for the EPROMs and RAMs. WRL# is asserted during low byte writes and word writes. WRH# is asserted during high byte writes and word writes. Note that RAM devices do not use A0. WRL# and WRH# determine whether the low byte (A0 = 0) or high byte (A0 = 1) is selected.

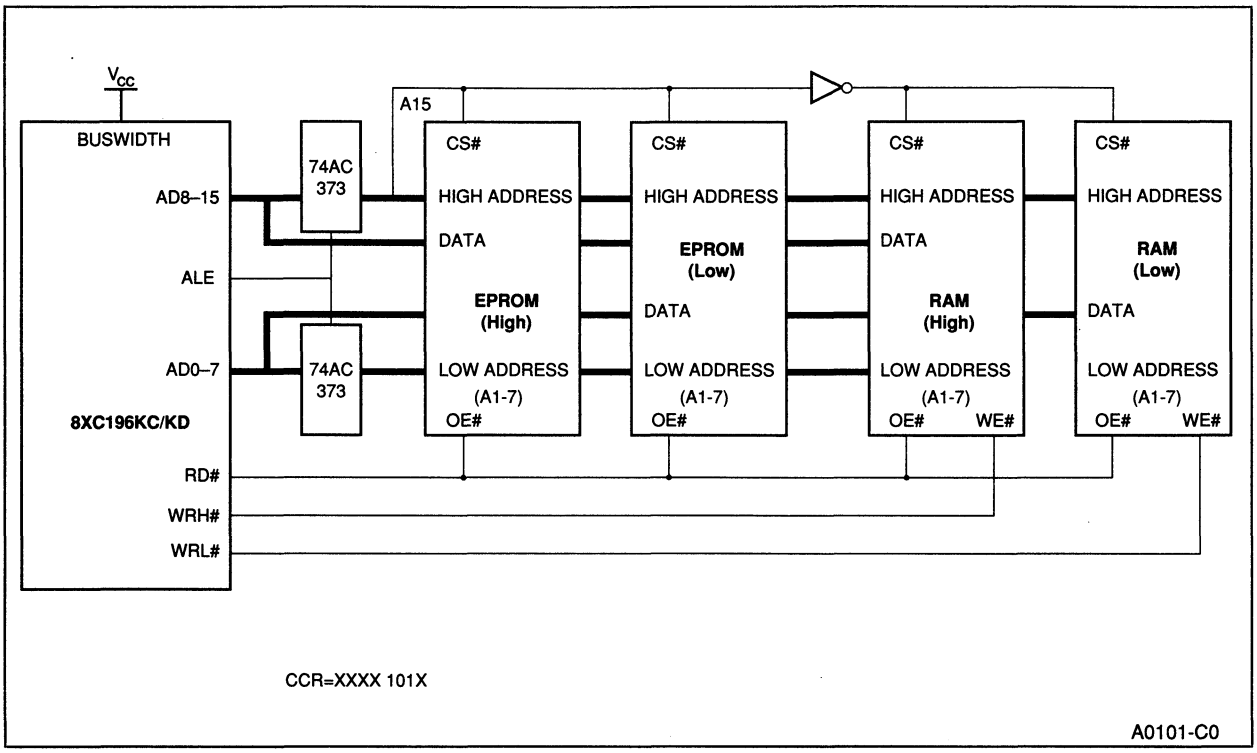

Figure 13-13. 16-Bit System with Single-Byte Writes to RAM

13.6.3. Address Valid Strobe Mode

When the Address Valid Strobe mode is selected, the 8XC196KC/KD generates the Address Valid signal (ADV#) instead of the Address Latch Enable signal (ALE). ADV# is asserted after an external address is valid (see Figure 13-14). This signal can be used to latch the valid address and simultaneously enable an external memory device.

Figure 13-16 shows a simple 8-bit system with a single EPROM. It is configured for the Address Valid Strobe mode. This system configuration uses the ADV# signal as both the EPROM chip-select signal and the address-latch signal.

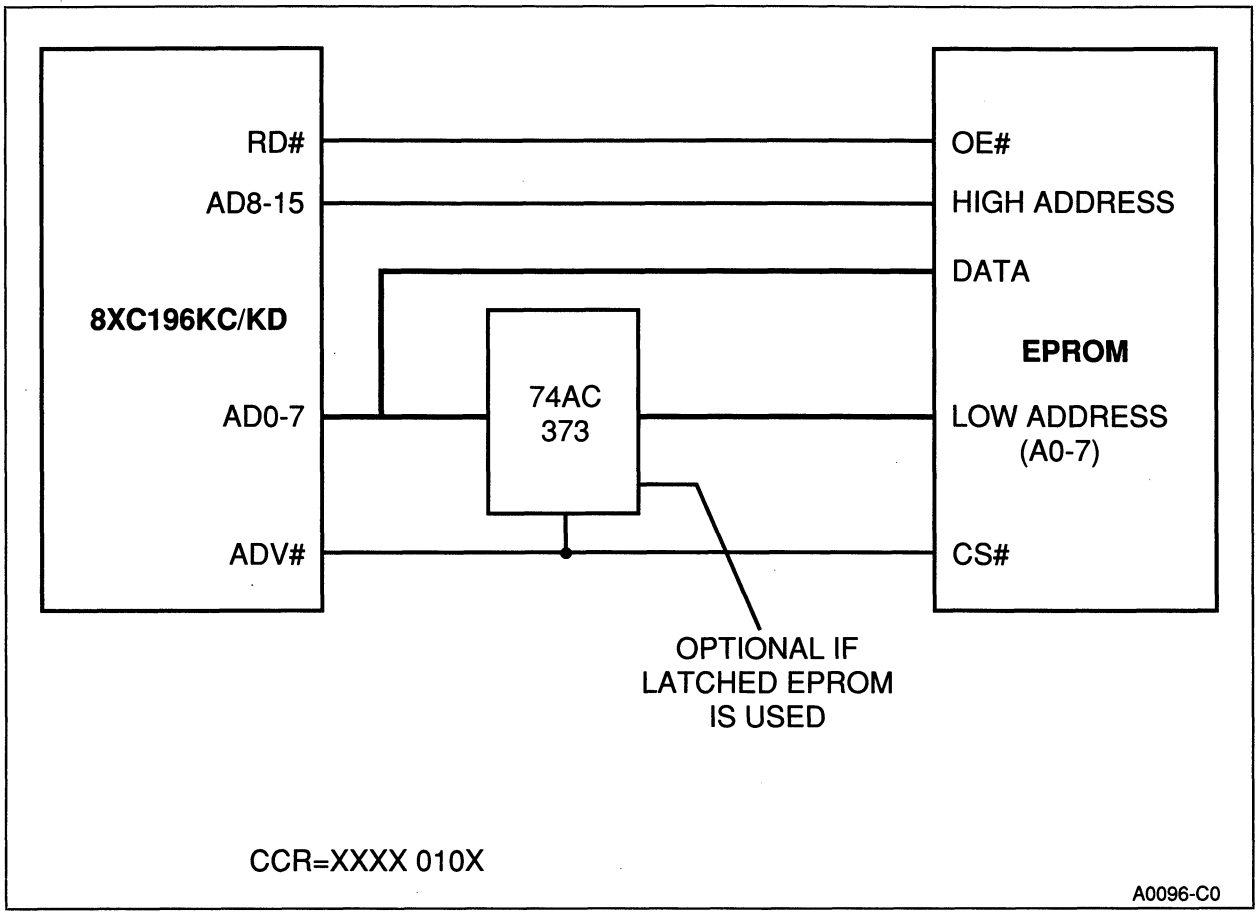

Figure 13-16. 8-Bit System with EPROM

Figure 13-17 shows a 16-bit system with two EPROMs. This system configuration uses the ADV# signal as both the EPROM chip-select signal and the address-latch signal.

13.6.4. Address Valid with Write Strobe Mode

When the Address Valid with Write Strobe mode is selected, the 8XC196KC/KD generates the ADV#, WRL#, and WRH# bus-control signals. This mode is used for a simple system using external 16-bit RAM. Figure 13-18 shows the timing. The RD# signal (not shown) is similar to WRL#, WRH# and WR#. The example system of Figure 13-19 uses Address Valid with Write Strobe.

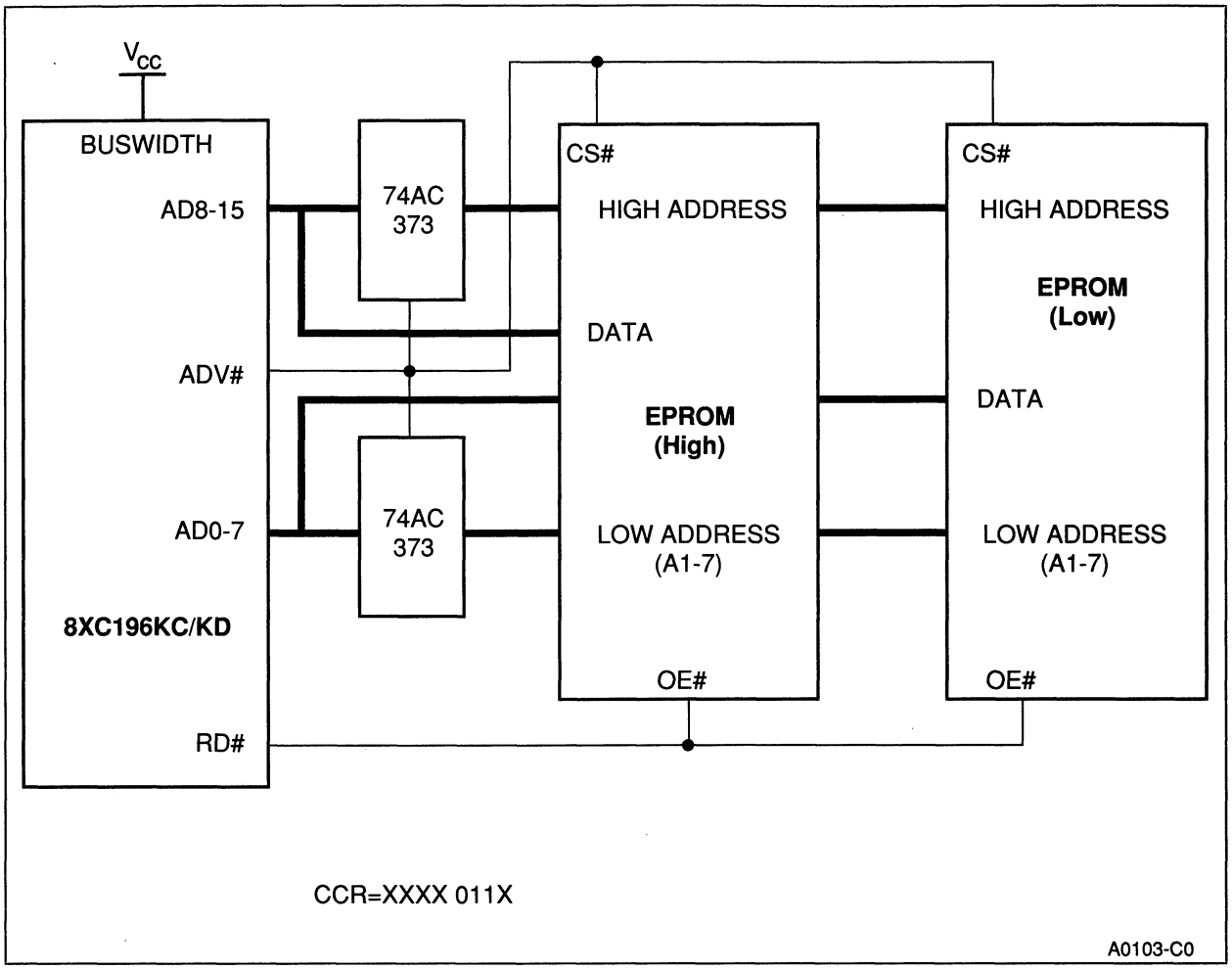

Figure 13-17. 16-Bit System with EPROM

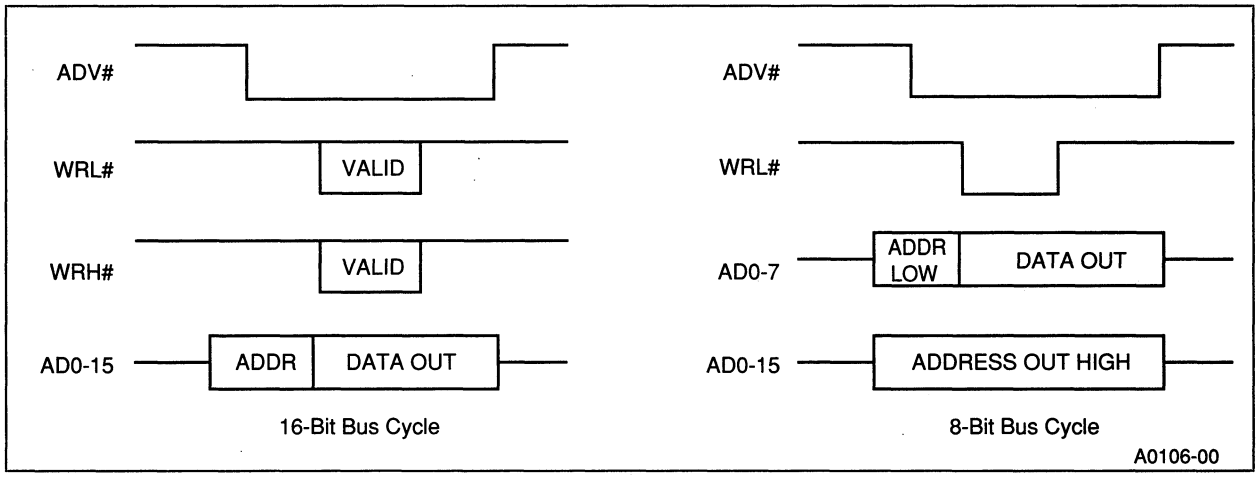

Figure 13-18. Timings of Address Valid with Write Strobe Mode

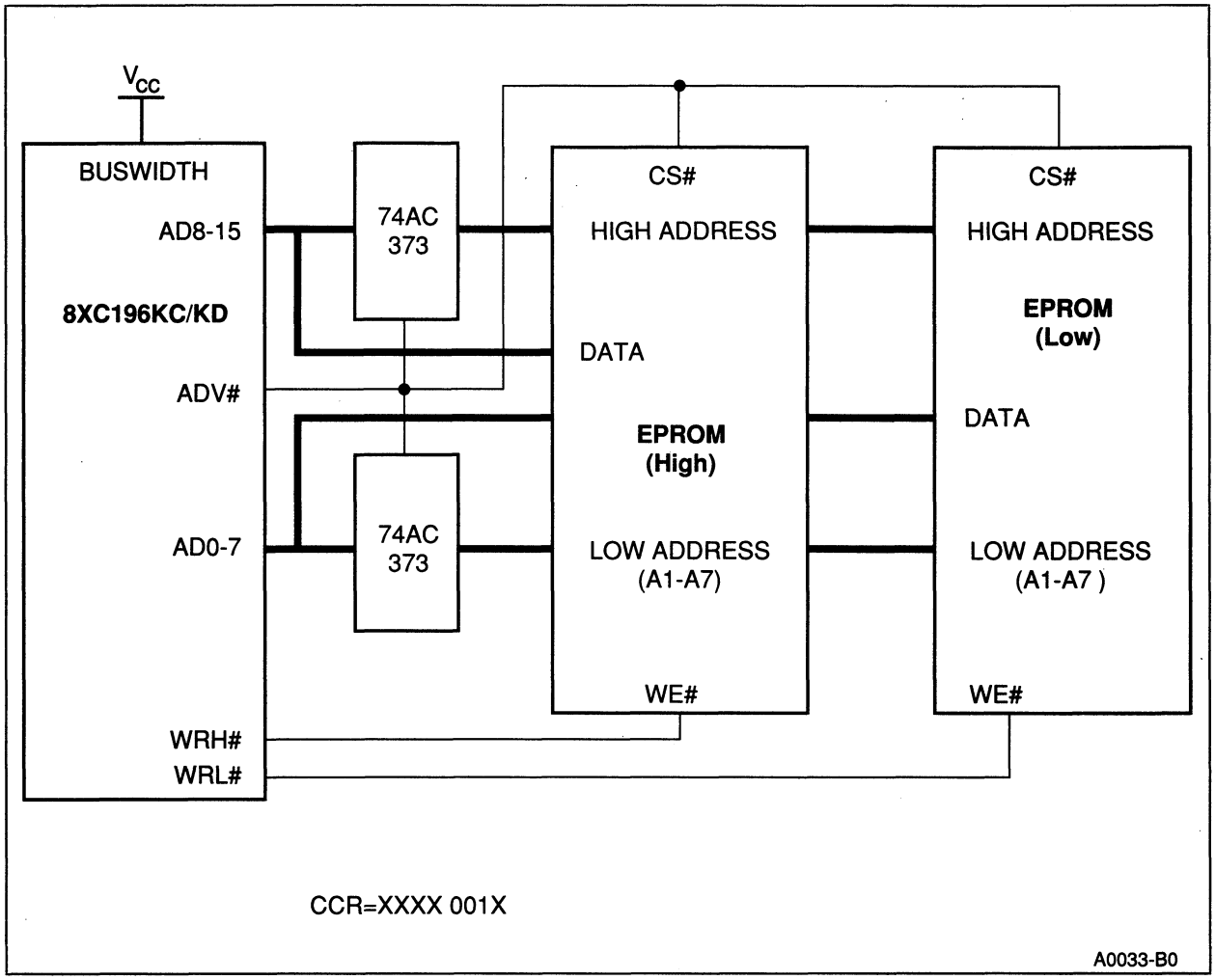

Figure 13-19. 16-Bit System with RAM

13.7. SYSTEM BUS AC TIMING SPECIFICATIONS

Refer to the latest data sheet for the AC timings to make sure your system meets specifications. The major system bus timing specifications are shown in Figure 13-20 and described in Table 13-5.

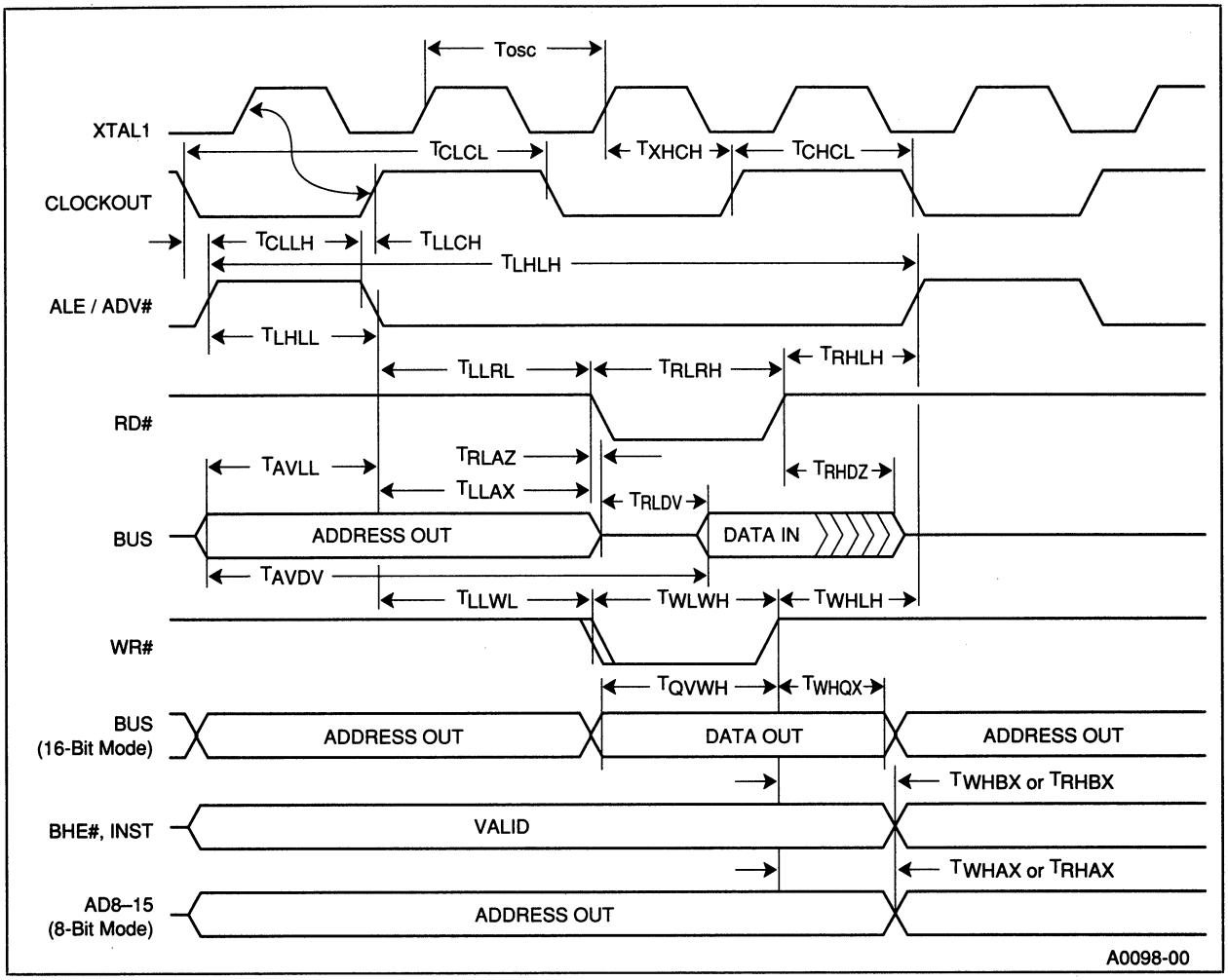

Figure 13-20. System Bus Timing

13.7.1. Explanation of AC Symbols

Each symbol is two pairs of letters prefixed by "T" (for time). The characters in a pair indicate a signal and its condition, respectively. Symbols represent the time between the two signal/condition points.

Conditions:

H— High
L— Low
V— Valid
X— No Longer Valid
Z— Floating

Signals:

A— Address
B— BHE#
BR— BREQ#
C— CLKOUT
D— DATA
G— BUSWIDTH
H— HOLD#
HA— HLDA#
L— ALE/ADV#
W— WR#/WRH#/WRL#
Q— Data Out
R— RD#
X— XTAL1
Y— READY

Table 13-5. AC Timing Definitions

TIMING THE MEMORY SYSTEM MUST PROVIDE:	
T_{AVYV}	Address Valid to READY Setup: Maximum time the memory system has to assert READY after the address is output by the 8XC196KC/KD to guarantee that at least one wait state will occur.
T_{LLYH}*	ALE Low to READY Setup: Maximum time the memory system has to assert READY after ALE falls to guarantee that at least one wait state will occur.
T_{CLYX}	READY Hold after CLKOUT Low: Minimum hold time is always 0 ns. If maximum specification is exceeded, additional wait states will occur.
T_{LLYX}	READY Hold after ALE Low: Minimum time the level of the READY signal must be valid after ALE falls. If the maximum value is exceeded, additional wait states will occur.
T_{AVGV}	Address Valid to BUSWIDTH Valid: Maximum time after address is valid until BUSWIDTH must be valid. If this specification is exceeded, the 8XC196KC/KD may not respond with the specified bus cycle.
T_{LLGV}	ALE Low to BUSWIDTH Valid: Maximum time after ALE/ADV falls until BUSWIDTH must be valid. If this specification is exceeded, the 8XC196KC/KD may not respond with the specified bus cycle.
T_{CLGX}	BUSWIDTH Hold after CLKOUT Low: Minimum time BUSWIDTH must be held valid after CLKOUT falls. Always 0 ns on the 8XC196KC/KD.
T_{AVDV}	Address Valid to Input Data Valid: Maximum time the memory device has to output valid data after the 8XC196KC/KD outputs valid address.
T_{RLDV}	RD# Low to Input Data Valid: Maximum time the memory system has to output valid data after the 8XC196KC/KD asserts RD#.
T_{CLDV}	CLKOUT Low to Input Data Valid: Maximum time the memory system has to output valid data after CLKOUT falls.
T_{RHDZ}	RD# High to Input Data Float: Time after RD# is inactive until the memory system must float the bus. If this timing is not met, bus contention will occur.
T_{RHDX}	Data Hold after RD# High: Time after RD# is inactive that the memory system must hold data on the bus. Always 0 ns on the 8XC196KC/KD.
F_{XTAL}	Frequency on XTAL: Frequency of the signal input on the XTAL1 input. The internal bus speed of the 8XC196KC/KD device is 1/2 F_{XTAL}.
T_{OSC}	$1/F_{XTAL}$: All AC Timings are referenced to T_{OSC}.
T_{XHCH}	XTAL1 High to CLKOUT High or Low.
T_{CLCL}	CLKOUT Cycle Time: Normally 2 T_{OSC}.
T_{CHCL}	CLKOUT High Period: Needed in systems which use CLKOUT as clock for external devices.
T_{CLLH}	CLKOUT Falling to ALE/ADV# Rising: Use to derive other timings.
T_{LLCH}	ALE/ADV# Falling to CLKOUT Rising: Use to derive other timings.

* Included only for comparison with HMOS device timings.

Table 13-5 AC Timing Definitions (Continued)

TIMING THE 8XC196KC/KD WILL PROVIDE:	
T_{LHLH}	ALE Cycle Time: Minimum time between ALE pulses.
T_{LHLL}	ALE/ADV# High Period. Use this specification when designing the external latch.
T_{AVLL}	Address Setup to ALE/ADV# Low: Length of time ADDRESS is valid before ALE/ADV# falls. Use this specification when designing the external latch.
T_{LLAX}	Address Hold after ALE/ADV# Low: Length of time ADDRESS is valid after ALE/ADV# falls. Use this specification when designing the external latch.
T_{LLRL}	ALE/ADV# Low to RD# Low: Length of time after ALE/ADV# falls before RD# is asserted. Could be needed to ensure proper memory decoding takes place before a device is enabled.
T_{RLCL}	RD# Low to CLKOUT Low: Length of time from RD# asserted to CLKOUT falling edge.
T_{RLRH}	RD# Low to RD# High: RD# pulse width.
T_{RHLH}	RD# High to ALE/ADV# Asserted: Time between RD# going inactive and the next ALE/ADV#. Useful in calculating time between inactive and next Address valid.
T_{RLAZ}	RD# Low to Address Float: Used to calculate when the 8XC196KC/KD stops driving Address on the bus.
T_{LLWL}	ALE/ADV# Low to WR# Low: Length of time after ALE/ADV# falls before WR# is asserted. Could be needed to ensure proper memory decoding takes place before a device is enabled.
T_{CLWL}	CLKOUT Low to WR# Low: Time between CLKOUT going low and WR# being asserted.
T_{QVWH}	Data Valid to WR# High: Time between data being valid on the bus and WR# going inactive. Memory devices must meet this specification.
T_{CHWH}	CLKOUT High to WR# High: Time between CLKOUT going high and WR# going inactive.
T_{WLWH}	WR# Low to WR# High: WR# pulse width.
T_{WHQX}	Data Hold after WR# High: Length of time after WR# rises that the data stays valid on the bus. Memory devices must meet this specification.
T_{WHLH}	WR# High to ALE/ADV# High: Time between WR# going inactive and next ALE/ADV#. Also used to calculate WR# inactive and next Address valid.
T_{WHBX}	BHE#, INST Hold after WR# High: Minimum time these signals will be valid after WR# inactive.
T_{RHBX}	BHE#, INST Hold after RD# High: Minimum time these signals will be valid after RD# inactive.
T_{WHAX}	AD8–15 Hold after WR# High: Minimum time the high byte of the address in 8-bit mode will be valid after WR# inactive.
T_{RHAX}	AD8–15 Hold after RD# High: Minimum time the high byte of the address in 8-bit mode will be valid after RD# inactive.