Features

- 232 Byte Register File.
- Register to Register Architecture.
- 10 bit A/D Converter with S/H.
- Five 8 bit I/O ports.
- 20 Interrupt Sources.
- Pulse Width Modulation Output.
- High speed I/O subsystem.
- Dedicated Baud Rate Generator.
- Full Duplex Serial Port.
- 16 bit Watchdog Timer.

Introduction (cont..)

- The MSC-96 family members are all high performance microcontroller with a 16 bit CPU and atleast 230 bytes of on-chip RAM.

- Intel MSC-96 family easily handles high speed calculations and fast input/out operations.

- All of the MCS-96 components share a common instruction set and architecture.

Introduction (cont..)

- However the CHMOS components have enhancements to provide higher performance with lower power consumption.

- These microcontroller contains dedicated I/O subsystem and perform 16-bit arithmetic instructions including multiply and divide operations.

- **CPU** :The major components of the MCS-96 CPU are the Register File and the Register / Arithmetic Logic Unit (RALU).

Introduction (cont..)

- Location 00H through 17H are the I/O control registers or Special function registers (SFR).

- Locations 18H and 19H contains the stack pointer, which can serve as general purpose RAM when not performing stack operations.

- The remaining bytes of the register file serve as general purpose RAM, accessible as bytes, words or double-words.

Introduction (cont..)

- Calculations performed by the CPU take place in the RALU. The RALU contains a 17bit ALU, the program status word (PSW), the program Counter (PC), a loop counter and three temporary registers.

- The RALU operates directly on the Register Files, thus eliminating accumulator bottleneck and providing for direct control of I/O operations through the SFR.

Architecture (cont..)

- The MCS-96 supports a complete instruction set which includes bit operations, byte operations, word operations, double-word operations (unsigned 32 bit), long operations (signed 32 bit), flag manipulations as well as jump and call instructions.

- All the standard logical and arithmetic instructions function as both byte and word operations.

- The jump bit set and jump bit clear instructions can operate on any of the SFR or bytes in the lower register files. These fast bit manipulations allow for rapid I/O functions.

Architecture (cont..)

- Byte and word operations make up the instruction set. The assembly language ASM-96 uses a "B" suffix on a mnemonic for a byte operation or for word operation.

- Addressing modes: This supports the following modes.

- Register-direct, indirect, indirect with auto-increment, immediate, short-indexed and long-indexed.

- These modes increase the flexibility and overall execution speed of the MCS-96 devices.
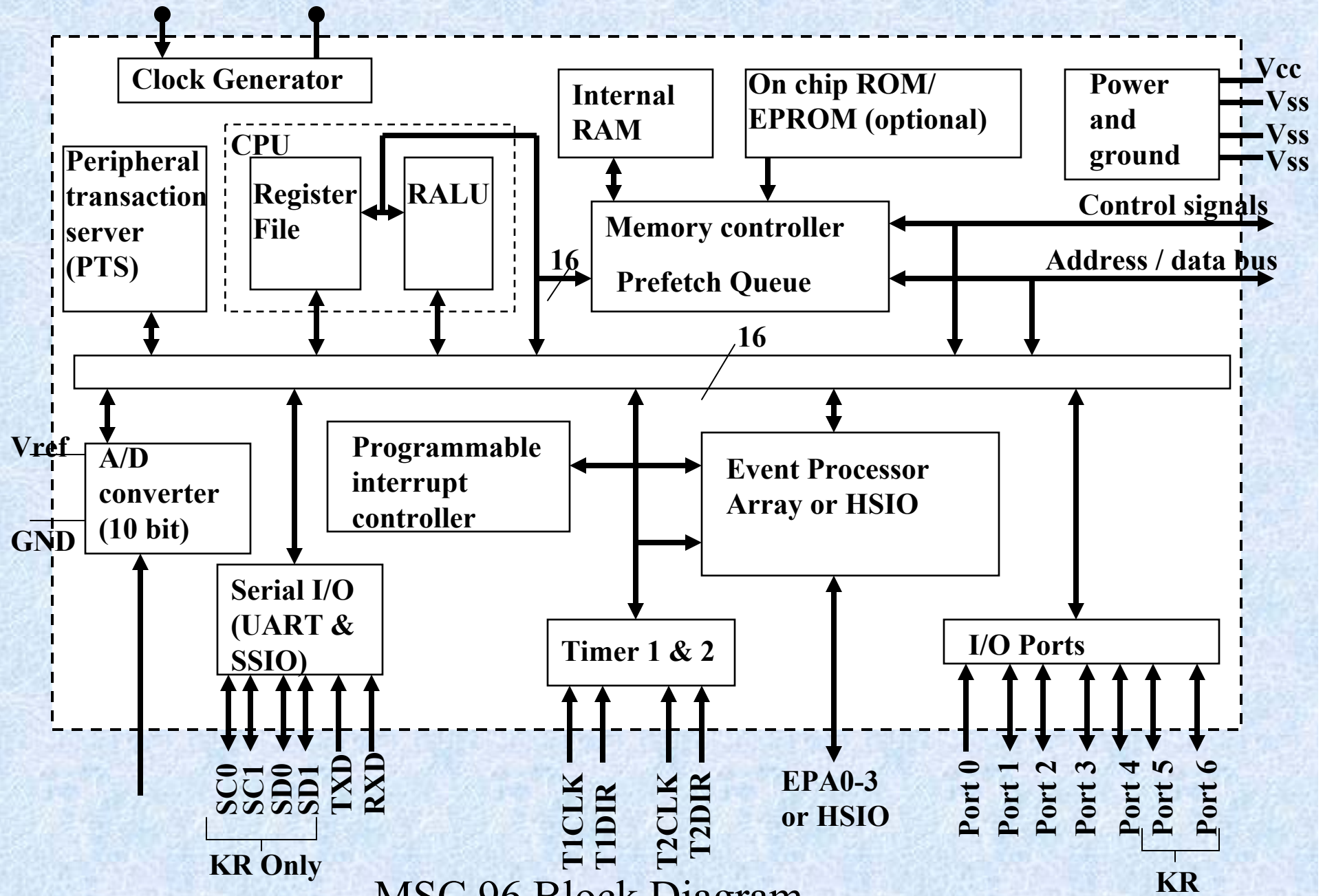
MSC 96 Block Diagram
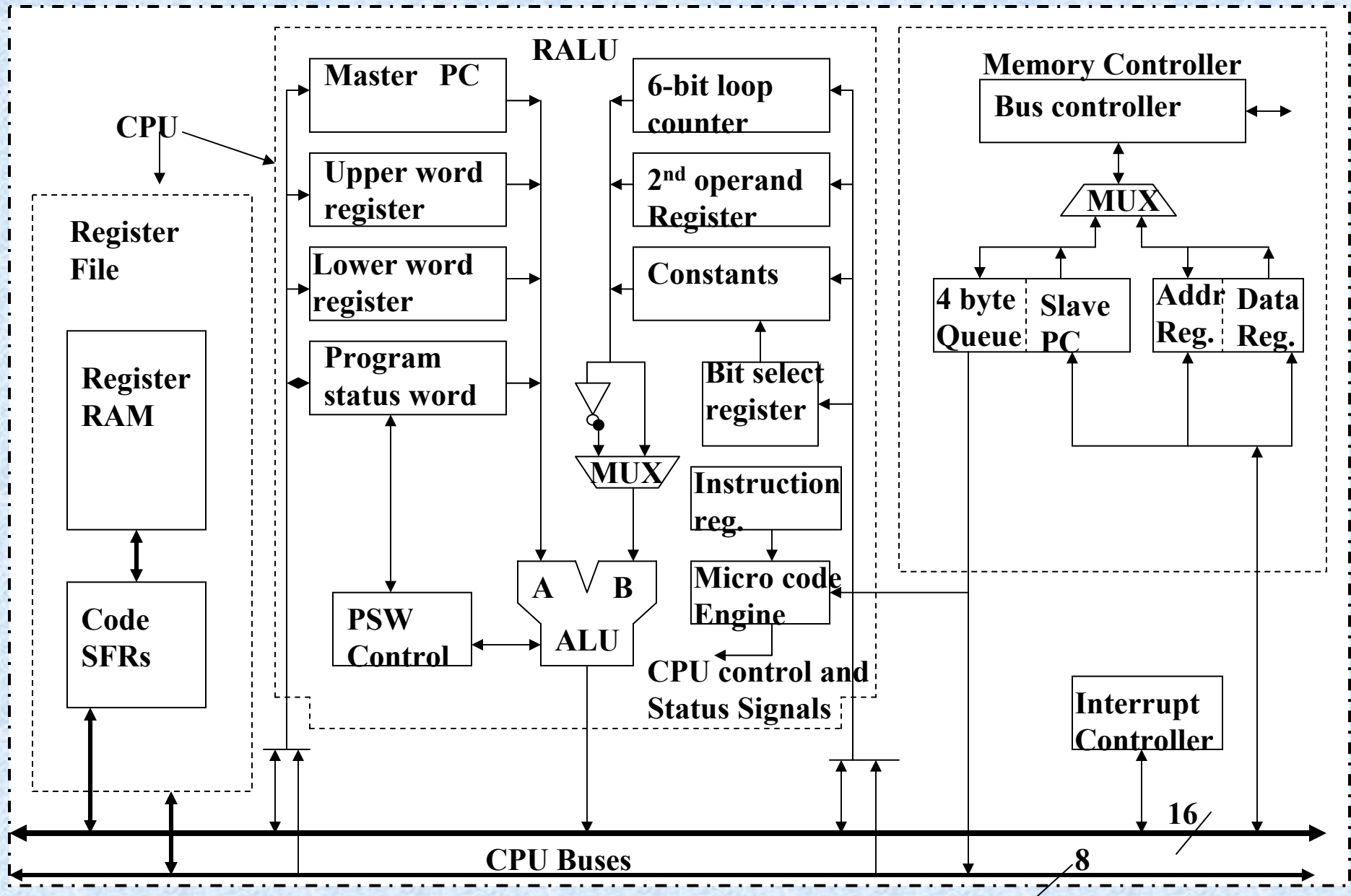
Block Diagram of Register File, RALU, Memory Controller and Interrupt Controller

Mnem	Dest or Src1	; One operand direct
Mnem	Dest, Src1	; Two operand direct
Mnem	Dest, Src1, Src2	; Three operand direct
Mnem	#Src1	; One operand immediate
Mnem	Dest, #Src1	; Two operand immediate
Mnem	Dest, Src1, Src2	; Three operand immediate
Mnem	[addr]	; one operand indirect
Mnem	[addr] +	; one operand indirect auto-increment
Mnem	Dest, [addr]	; Two operand indirect
Mnem	Dest, [addr] +	; Two operand indirect auto-increment
Mnem	Dest, Src1, [addr]	; Three operand indirect
Mnem	Dest, Src1, [addr] +	; Three operand indirect auto-increment
Mnem	Dest, offs [addr]	; Two operand indexed (short or long)
Mnem	Dest, Src1, offs [addr]	; Three operand indexed (short or long)

Instruction Format

8096 Peripherals (cont..)

- **Standard I/O Ports** – The 8096 has five 8 bit I/O ports.

- Port 0 is an input port that is also the analog input for the A/D converter.

- Port 1 is a quasi-bidirectional port.

- Port 2 contains three types of port lines.

- Quasi-Bidirectional, input and output. Other functions on the 8096 share the input and output lines with Port 2.

- Port 3 and 4 are open-drain bidirectional ports that share their pins with the address/data bus.

8096 Peripherals (cont..)

- **Timers** – The 8096 has two 16 bit timers. Timer 1 and Timer 2.

- An internal clock increments the Timer 1 value every 8 state times. (A state time is 3 oscillator periods)

- An external clock increments Timer 2 on every positive and negative transition.

- Either an internal or external source can reset Timer 2.

- This two timers can generate an interrupt when crossing the 0FFFFH/0000H boundary.

- The 8096 includes separate, dedicated timers for serial port baud rate generator and watchdog timer.

8096 Peripherals (cont..)

- The watchdog Timer is an internal timer that resets the system if the software fails to operate properly.

- **High Speed Input Unit (HSI)** – The 8096 HIS unit can record times of external events with a 9 state time resolution. It can monitor four independently configurable HSI lines and captures the value of timer 1 when events takes place.

- The four types of events that can trigger captures include: rising edge only, falling edge only, rising or falling edges, or every eight rising edge.

- The HSI unit can store upto 8 entries (Timer 1 values).

8096 Peripherals (cont..)

- Reading the HSI holding register unloads the earliest entry placed in the FIFO.

- The HSI unit can generate an interrupt when loading an entry into the HSI holding register or loading the sixth entry into the FIFO.

8096 Peripherals (cont..)

- **High Speed Output Unit (HSO)** – The 8096 HSO unit can trigger events at specified times based on Timer1 or Timer2.

- These programmable events include: starting an A/D conversion, resetting Timer2, generating upto four software time delays, and setting or clearing one or more of the six HSO output lines.

- The HSO unit stores pending event and specified times in a Content Addressable Memory (CAM) file. This file stores upto 8 commands.

8096 Peripherals (cont..)

- Each command specifies the action time, the nature of the action, whether an interrupt is to occur, and whether Timer1 or Timer2 is the reference timer.

- Every 8 state times the HSO compares the CAM locations for time matches. The HSO unit triggers the specified event when it finds a time match.

- A command is cleared from the CAM as soon as it executes.

8096 Peripherals (cont..)

- **Serial Port** – The serial port on the 8096 has one synchronous (Mode 0) and three asynchronous modes (Modes 1, 2 and 3).

- The asynchronous modes are full duplex.

- Mode 0, the synchronous mode, is to expand the I/O capability of the 8096 using shift register.

- Mode 1 is the standard asynchronous mode used for normal serial communication.

- Modes 2, 3 are 9-bit modes commonly used for multiprocessor communications.

8096 Peripherals (cont..)

- **Pulse Width Modulator (PWM)** – The PWM output waveform is a variable duty cycle pulse that repeats every 256 state times.

- The PWM output can perform digital to analog conversions and drive several types of motors that require a PWM waveform for more efficient operation.

- **A/D Converter** – The 8096 A/D converts an analog input to a 10 bit digital equivalent.

- The main components of the A/D Converter are: 8 analog inputs, an 8 to 1 multiplexer, a sample and hold capacitor and resistor ladder.

8096 Peripherals (cont..)

- The A/D Converter can start a conversion immediately or the High Speed Output unit can trigger a conversion at a preprogrammed time.

- The A/D converter performs a conversion in 88 state times. Upon completion of each conversion the converter can generate a conversion complete interrupt.

- The 8X9X provides separate V_{REF} and ANGND supply pins to isolate noise on the Vcc or Vss lines.

8096 Peripherals

- **Interrupts** – There are 21 interrupts sources and 8 interrupt vector on the 8096.

- When the interrupt controller detects one of the 8 interrupts it sets the corresponding bit in the interrupt pending register. Individual interrupts are enabled or disabled by setting or clearing bits in the interrupt mask register.

- When the interrupt controller decides to process an interrupt, it executes a "call" to an interrupt service routine ISR. The corresponding interrupt vector contains the address of the ISR. The interrupt controller then clears the associated pending bit.

7	6	5	4	3	2	1	0
				0	0	1	1
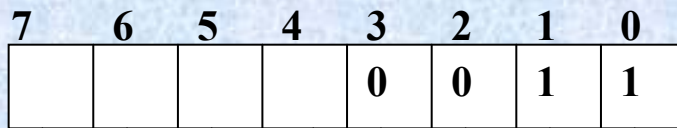
Chip configuration byte
CCB (2018H)   (ROM or EPROM)
Set to 1 for compatibility with future parts.

Bus width select
If set, then BUSWIDTH pin determines the bus
width. If cleared, then external 8-bit data bus is
selected.

Write strobe mode select
If set, then WRH# / BHE# becomes BHE#   and
WRL# / WR# becomes WR#.
If cleared, then WRH# / BHE# becomes WRH# and
WRL# / WR# becomes WRL#.

Address valid strobe select
If set, then ALE / ADV# becomes ALE
If cleared , then ALE / ADV# becomes ADV#

IRC1	IRC0	Internal ready control mode
0	0	Limit to 1 wait state
0	1	Limit to 2 wait state
1	0	Limit to 3 wait state
1	1	Disable internal ready control

LOC1	LOC0	Internal ROM / EPROM lock modes
0	0	Read protected ; EPROM is also write protected.
0	1	Read protected
1	0	EPROM part is write protected
1	1	No operation

Chip configuration byte (CCB)

Configuring the 8096 (cont..)

- The 8096 can be operated in either the single-chip mode, or two of its ports can be redefined to bring out the internal address bus and data bus.

- For the single chip mode, the internal ROM and EPROM must be accessed. This choice is made by tying the EA# pin high.

- When EA pin is tied high, the internal ROM or EPROM is accessed during instruction and data fetches from addresses 2080 to 3FFFH and for interrupt vectors located at addresses 2000 to 2011H.

Configuring the 8096 (cont..)

8096BH

+5V —•— EA

Single chip mode

8096BH

+5V —— BUSWIDTH

+5V or —•— EA
GND

READY ←— Memory read
INST —→ Instruction fetch
ADV —→ Address valid
RD —→ Read control
WRH —→ Write high control
WRL —→ Write low control
AD15 – AD0 ←—/→ Multiplexed bus
                16

CCB (2018H)

| | | | | 0 | 0 | 1 | |

EA: Tie to +5V to use internal ROM or EPROM

EA: Tie to GND to disable internal ROM or EPROM
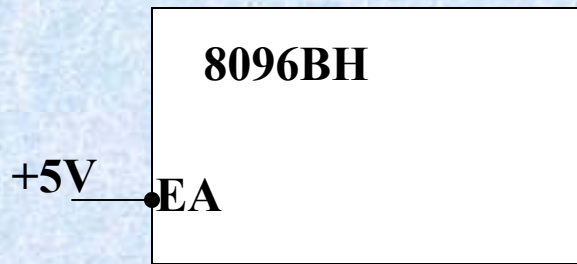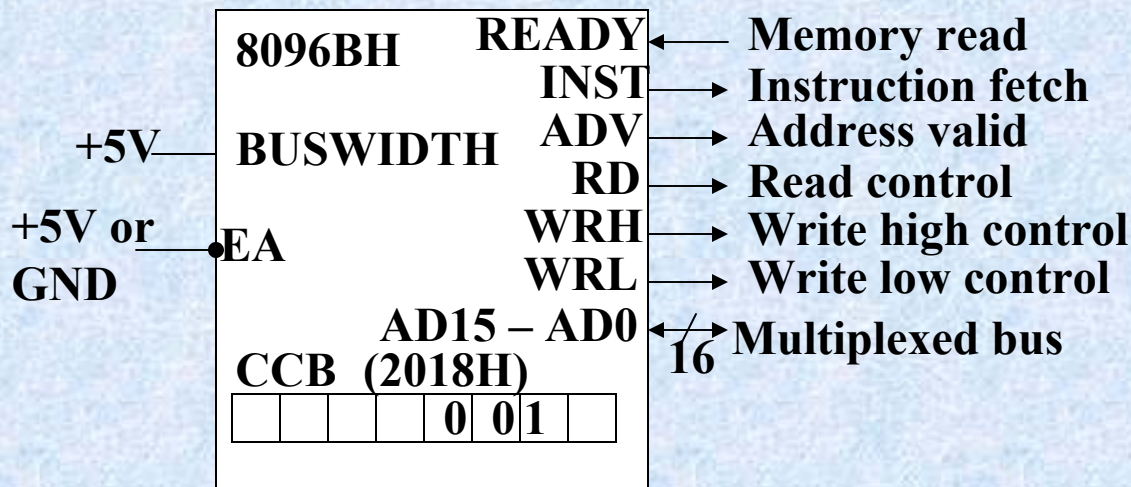
Expanded mode with 16 bit multiplexed bus

Modes of Intel 8096 operation.

Configuring the 8096 (cont..)

8096BH		
	READY	Memory read
	INST	Instruction fetch
BUSWIDTH	ADV	Address valid
	RD	Read control
EA	WR	Write control
	A15 – A8 $\frac{8}{}$	Write low control
	AD7 – AD0 $\frac{8}{}$	Multiplexed bus

+5V or GND

CCB (2018H)

				0	0	1	

EA: Tie to +5V to use internal ROM or EPROM

EA: Tie to GND to disable internal ROM or EPROM
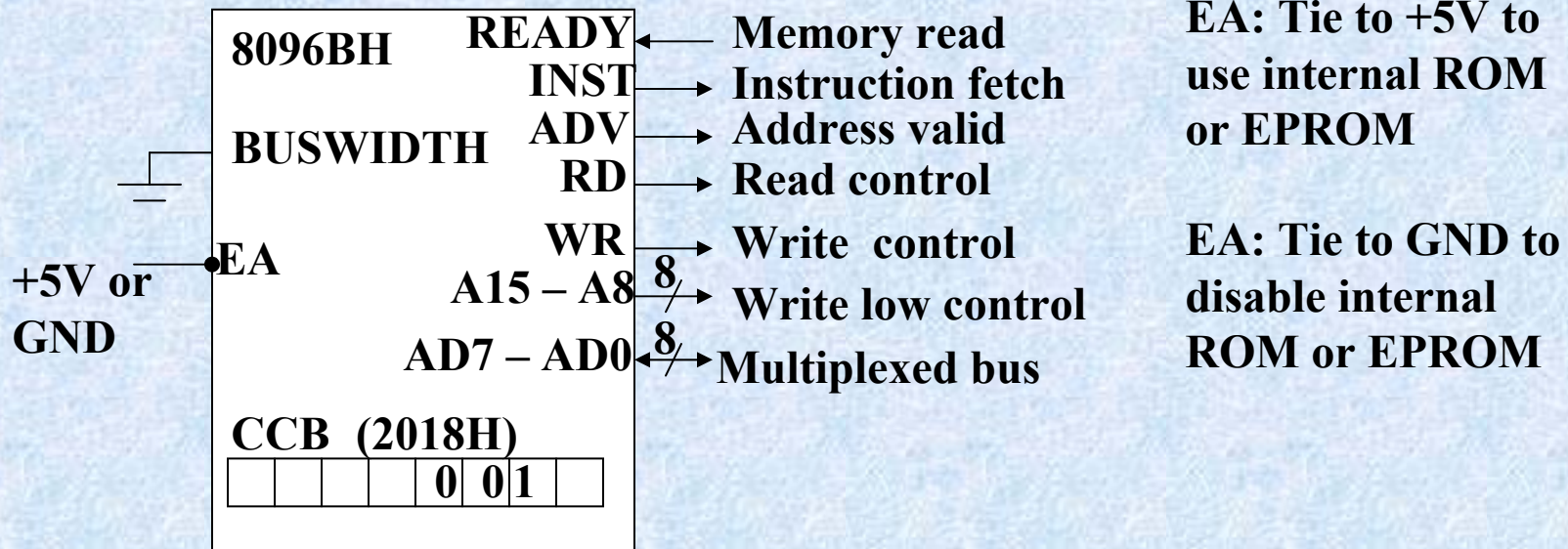
Expanded mode with 8 bit multiplexed bus

Configuring the 8096 (cont..)

8096BH   **READY** ← **+ 5V**

      CCB   **(No wait state)**

		X	X				
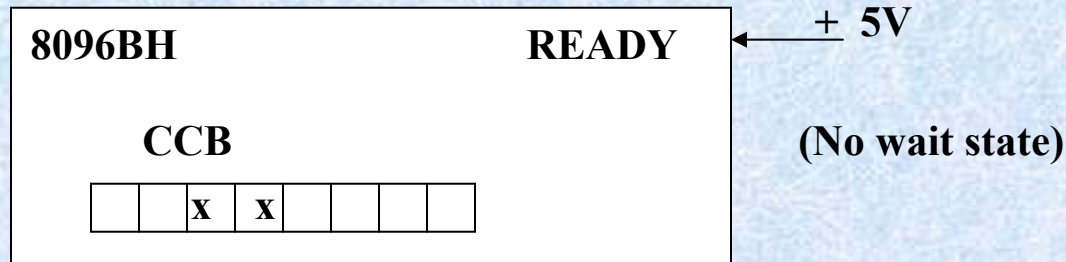
(a) Avoiding wait states entirely, for use with fast external parts.

8096BH   **READY** ← **GND**

      CCB   **(One, two or three wait states)**
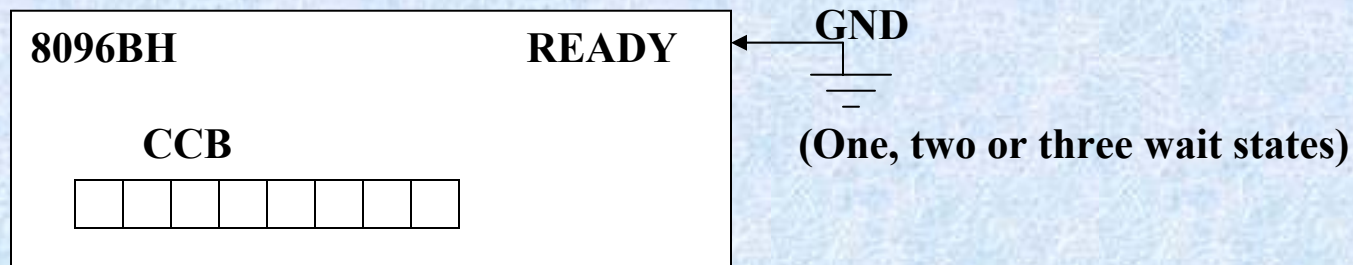
     0   **0**   **One wait state = 250 extra nanoseconds**

     0   **1**   **Two wait state = 500 extra nanoseconds**

     1   **0**   **Three wait state = 750 extra nanoseconds**

(b) Use with slower external parts.

Alternative use of the READY input.

Configuring the 8096 (cont..)

```
8096BH                              READY
  + 5 V                             AD15              ┌──┐
(16 bit      BUSWIDTH                                 │  │ ●────► A15
multiplexed     CCB
bus)                                        (Address-dependent wait state)
           ┌──┬──┬──┬──┬──┬──┬──┐
           │  │  │ 0│ 0│  │  │  │
           └──┴──┴──┴──┴──┴──┴──┘
```

```
8096BH                              READY
             BUSWIDTH                A15              ●────► A15
                 CCB
(8 bit                                      (Address-dependent wait state)
multiplexed
bus)       ┌──┬──┬──┬──┬──┬──┬──┐
           │  │  │ 0│ 0│  │  │  │
           └──┴──┴──┴──┴──┴──┴──┘
```
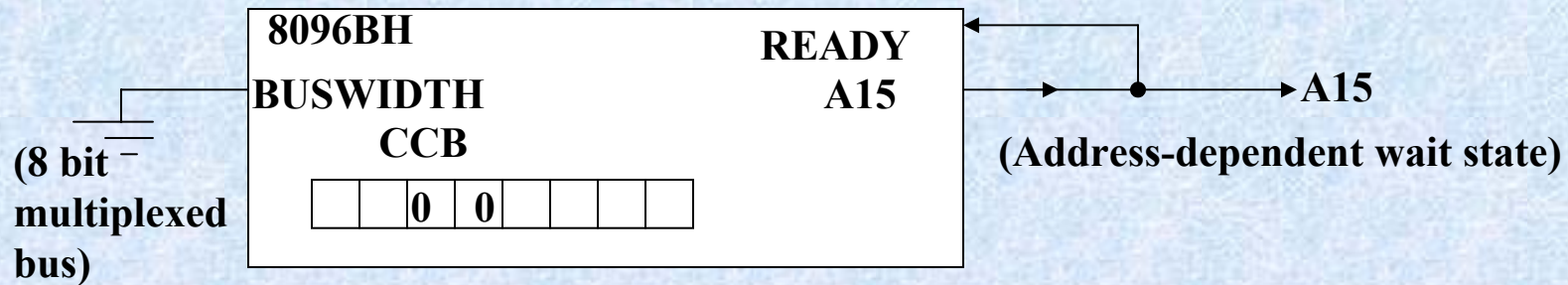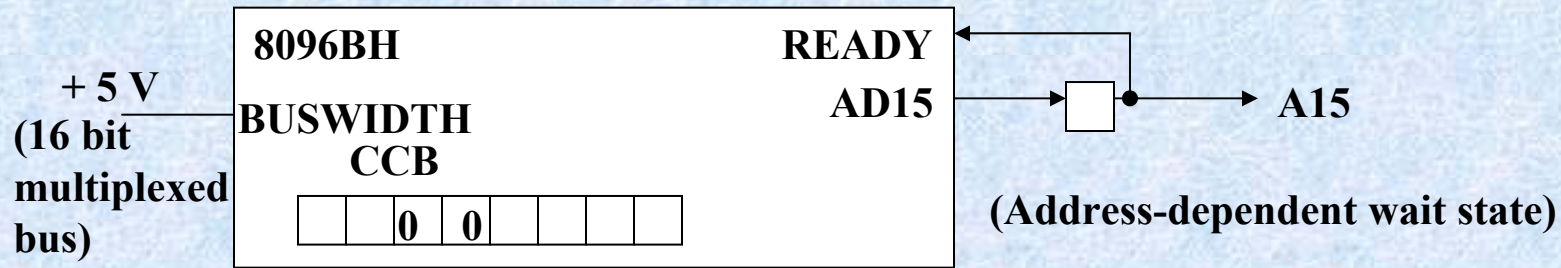
(c) external accesses to address below 8000H get an extra wait state, where as accesses above 8000H get no extra wait state.

Configuring the 8096 (cont..)

- When operated in the expanded mode the internal ROM or EPROM can still be used by tying EA# high.

- Accesses to the addresses 2000 to 2011H and 2080 to 3FFFH can be made to access off-chip memory by tying the EA# pin low.

- If the EA pin is high, then we have the option of using the internal ROM or EPROM together with external memory and devices.

- One of the options made available by the BH series over the original 8096 family is the option to deal with either a 16 bit external data bus or else an 8 bit external data bus.

Configuring the 8096 (cont..)

- The latter options permits expanding the 8096 with a single byte wide static RAM chip or with a single byte wide EPROM chip for program memory.

- The latter is particularly convenient for users who can either put their application program into a single EPROM or who do not have the EPROM programming capability to separate their object code into even addresses and odd addresses as required for the two byte wide EPROM used with a 16 bit data bus.

Configuring the 8096 (cont..)

- The choice of bus width is made in two places. When the 8096 comes out of reset, it reads the content of address 2018H of our ROM or EPROM . This is called the chip configuration CCB.

- The 8096 stores this byte in a chip configuration register which is unaccessable by our software.

- Bit 1 works together with the external BUSWIDTH pin to determine the data bus width (when the EA pin is tied low).

Configuring the 8096 (cont..)

- While the BUSWIDTH pin is tied either high or low, it can actually be changed during each bus cycle of normal operation.

- If it is tied to the A_{15} address lines, then accesses to external addresses 8000 to FFFFH would use a 16-bit data bus while accesses to external addresses below this would use an 8-bit data bus. In either case, the full 16 bit address bus is brought out.

- When an 8-bit data bus is brought out, the lines which bring out the upper half of the address bus do not have to be multiplexed.

Configuring the 8096 (cont..)

- In this case, the designers of the chip have saved users the need for an external latch for the upper half of the address bus by latching the address internally.

- The original 8096 parts gave the user of the expanded chip an ALE output. This was used to latch the address. The new option is selected with a 0 in bit 3 of CCB. The ADV# line remains high during any machine cycles which are not accessing external memory, but goes low during external accesses. Because of this ADV# can be used to simplify the decoding to enable external devices.

Configuring the 8096 (cont..)

- In addition to the external access, ADV# drop low at precisely the current time to latch the multiplexed address. Consequently, it can serve double duty, both helping with decoding and also latching the multiplexed address.

- Another feature of the original 8096 parts operating in the expanded mode was the need to decode a BHE# signal.

- This was used during writes to a byte at an odd address so that the lower byte on the 16-bit data bus could be left unchanged.

- Users of the original 8096 parts had to gate BHE# together with a WR# signal to generate two write signals.

Configuring the 8096 (cont..)

- One for chips connected to the upper half of the data bus and one for chip connected to the lower half of the data bus.

- This option is selected with 0 bit 2 of CCB.

- INST output is a signal which takes on when the RD# line is active, signaling that a read from an external device is taking place. If the read is an instruction fetch, then INST will be high. Otherwise it will be low during the read cycle.

- Users of logia analyzers and designers of 8096 emulators can use this signal to help sort out the activity on external bus.

Configuring the 8096 (cont..)

- The READY control line permits the 8096 to run at full speed for its internal accesses and yet to slow down for some of its external accesses.

- It is used in conjunction with bits 4 and 5 of CCB to introduce extra 250ns (assuming a 12 MHz crystal) wait states into external read and write cycles.

- If the READY line is tied high then the CCB bits do not matter and no external wait states are introduced into external read or writes.

- If the READY line is tied low(signifying that external devices are not ready), then this READY signal can be overridden by the CCB bits.

Configuring the 8096 (cont..)

- Thus 00 in bits 4 and 5 of CCB will now limit the delay to a signal wait state.

- READY line can be changed by dynamically from cycle to cycle. If it is tied to the upper address lines, then we can position external devices which can run at full speed in the 8000 to FFFF address range and slower external device needing an extra wait state at lower addresses.

- The lock mode is selected by the coding of bit 6 and 7. whether the software is in on-chip manage to get the chip to execute code from external memory and then have that external program dump the internal memory to the serial port.

Configuring the 8096 (cont..)

- In a read-protected mode, only code executing from internal memory can read from memory addresses between 2020 to 3FFFH.

- In a write-protected mode, no code can write to memory address between 2000 and 3FFFH.

- One problem arises with a memory protection scheme such as, if we purchase ROM-protected parts from Intel, then before we use them, we would like to test them.

- We can drive the EA# line low and use our own program to test all the resource on the chip. This does not test the ROM contents.

Configuring the 8096 (cont..)

- Intel supports the verification of ROM by including a 16byte security key, located at address 2020 to 202FH. Before protected memory can read, the chip must read external memory locations 4020 to 402FH and compare the contents with the internal security key.

- Access to protected memory will only be allowed if a match found for all 16 bytes.

- The first 26 addresses from the register file, used to set up and access almost all of the on-chip resources. The rest of the page 0 is dedicated to internal RAM, for a total of 230 bytes of RAM.

- While the ROM or EPROM extends from 2000 to 3FFFH, Intel reserves addresses 2012 to 2017H.
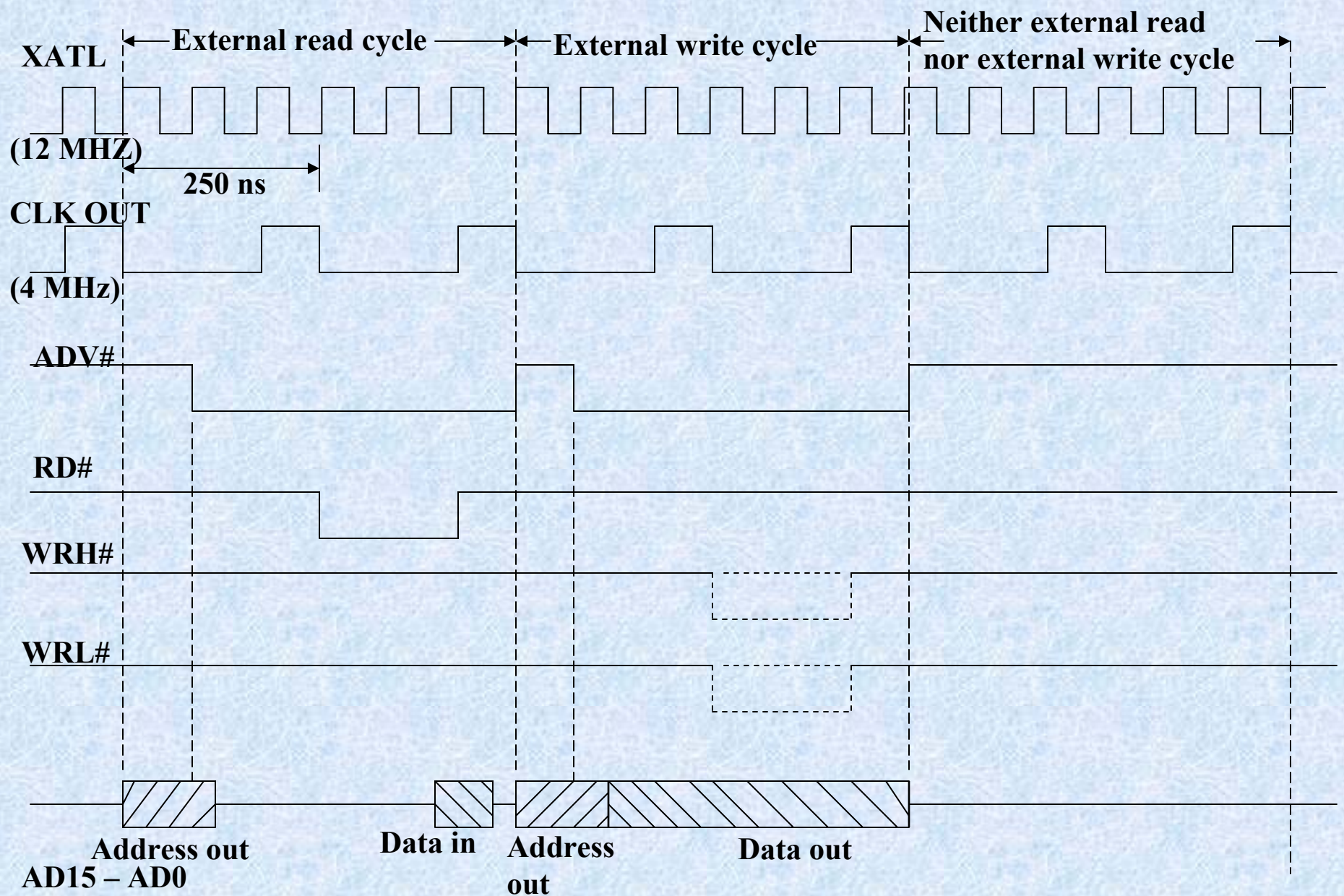
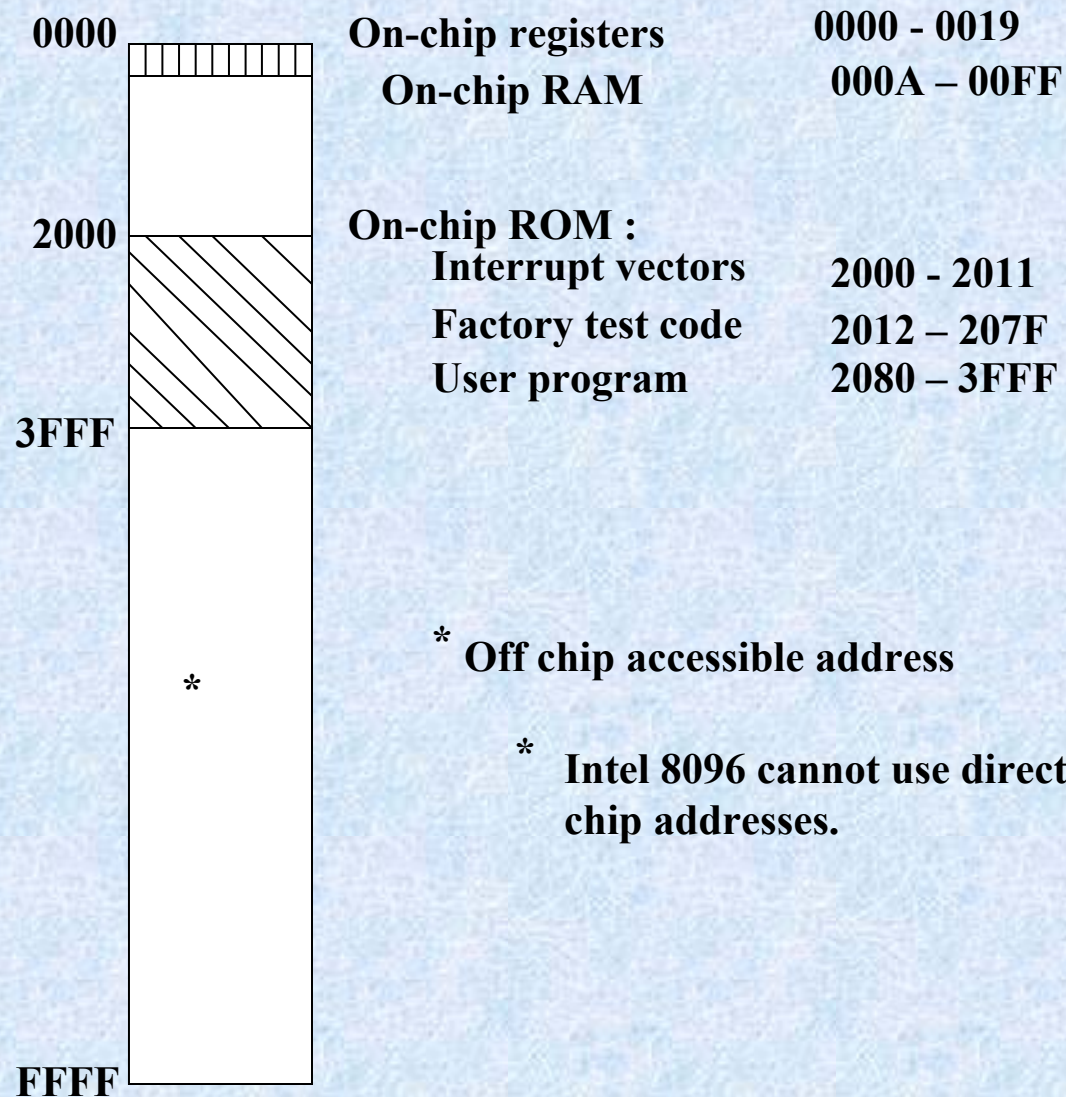Timing for expand mode with 16 bit multiplexed bus
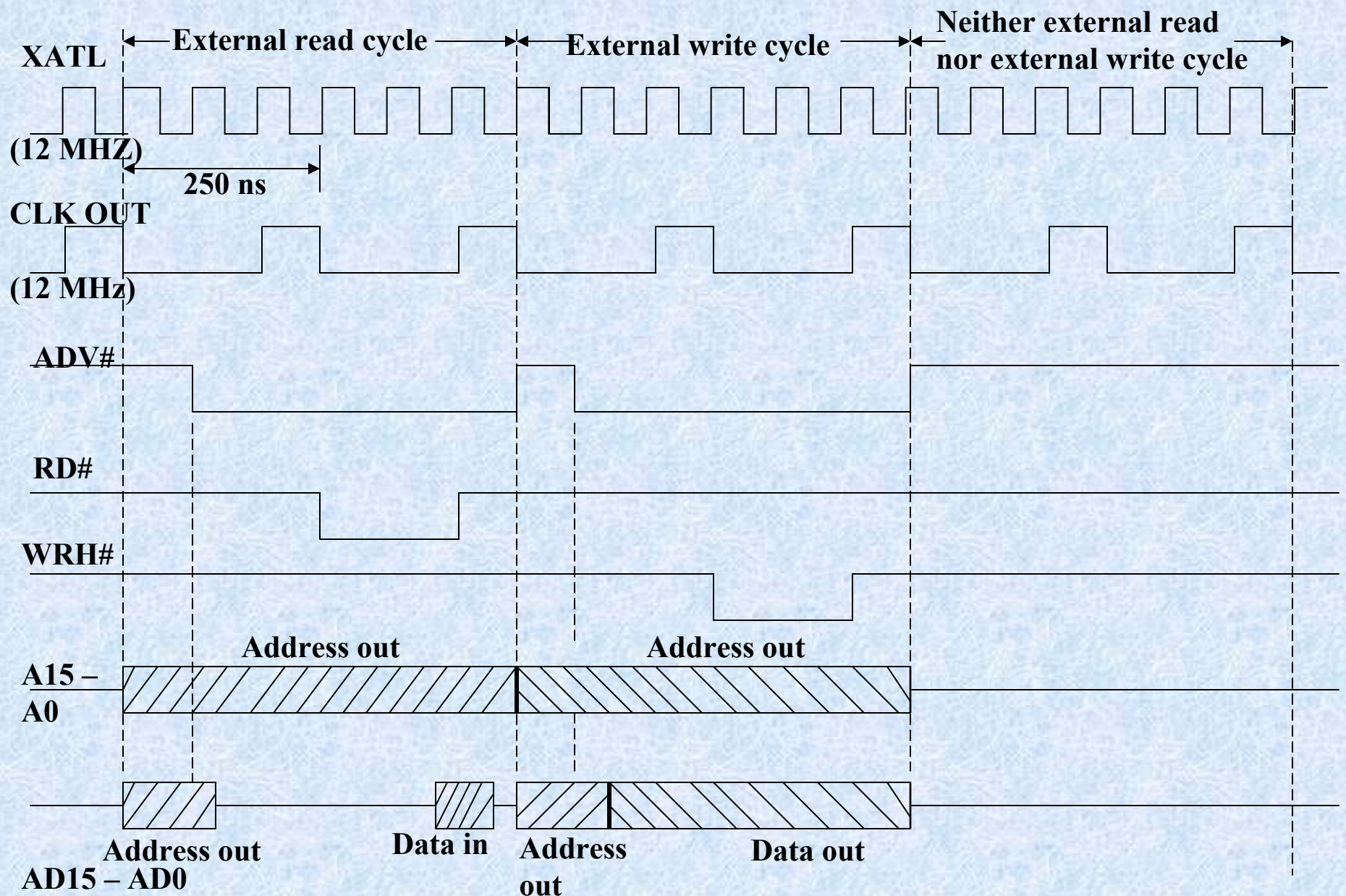
Timing for expand mode with 8 bit multiplexed bus

REGISTERS	EQU	0000H	; Registers extend upto 0019H
RAM	EQU	001AH	; RAM extend upto 00FFH
EXTERNAL 1	EQU	0100H	; External memory space (upto 1FFDH)
PORT3	EQU	1FFEH	
PORT4	EQU	1FFFH	
ROM	EQU	2000H	;ROM or EPROM, extends up to 3FFFH
INTERRUPT-VECTORS	EQU	2000H	;Vector extends up to 2011H
RESERVED	EQU	2012H	;ROM up to 207FH is reserved for factory ;test code (except for CCB)
CCB	EQU	2018H	;Chip configuration byte
START	EQU	2080H	;User program or data up to 3FFFH
EXTERNAL2	EQU	4000H	;External memory space (upto FFFFH)

Intel 8096 memory space allocation.

SYMBOLIC NAMES FOR THE I/O REGISTERS OF THE 8096

R0	EQU	00H:WORD	;R	Zero register (reads as 0000H)
AD_COMMAND	EQU	02H:BYTE	; W	A/D command register
AD_RESULT_LO	EQU	02H:BYTE	;R	A/D result, lo byte (byte read only)
AD_RESULT_HI	EQU	03H:BYTE	;R	A/D result, hi byte (byte read only)
HSI_MODE	EQU	03H:BYTE	; W	HSI mode register
HS0_TIME	EQU	04H:WORD	; W	HS0 time hi/lo (word write only)
HSI_TIME	EQU	04H:WORD	;R	HSI time hi/lo (word read only)
HS0_COMMAND	EQU	06H:BYTE	; W	HS0 command register
HSI_STATUS	EQU	06H:BYTE	;R	HSI status register
SBUF	EQU	07H:BYTE	;R/W	Receive buffer (read), Transmit buffer (write)
INST_MASK	EQU	08H:BYTE	;R/W	Interrupt mask register
INT_PENDING	EQU	09H:BYTE	;R/W	Interrupt pending register
WATCHDOG	EQU	0AH:BYTE	; W	Watchdog timer register
TIMER 1	EQU	0AH:WORD	;R	Timer 1 hi/lo (word read only)
TIMER 2	EQU	0CH:WORD	;R	Timer 2 hi/lo (word read only)
BAUD_RATE	EQU	0EH:BYTE	; W	Baud rate control register
PORT 0	EQU	0EH:BYTE	;R	Port 0
PORT 1	EQU	0FH:BYTE	;R/W	Port 1
PORT 2	EQU	10H:BYTE	;R /W	Port 2
SP_CON	EQU	11H:BYTE	; W	Serial port control register
SP_STAT	EQU	11H:BYTE	;R	Serial port status register
SP	EQU	18H:BYTE	;R /W	Stack pointer

0000

2000

3FFF

*

FFFF

On-chip registers   **0000 - 0019**
  On-chip RAM   **000A – 00FF**

On-chip ROM :
  Interrupt vectors   **2000 - 2011**
  Factory test code   **2012 – 207F**
  User program   **2080 – 3FFF**

* **Off chip accessible address**

  * **Intel 8096 cannot use direct (page 0) addressing for the off-chip addresses.**
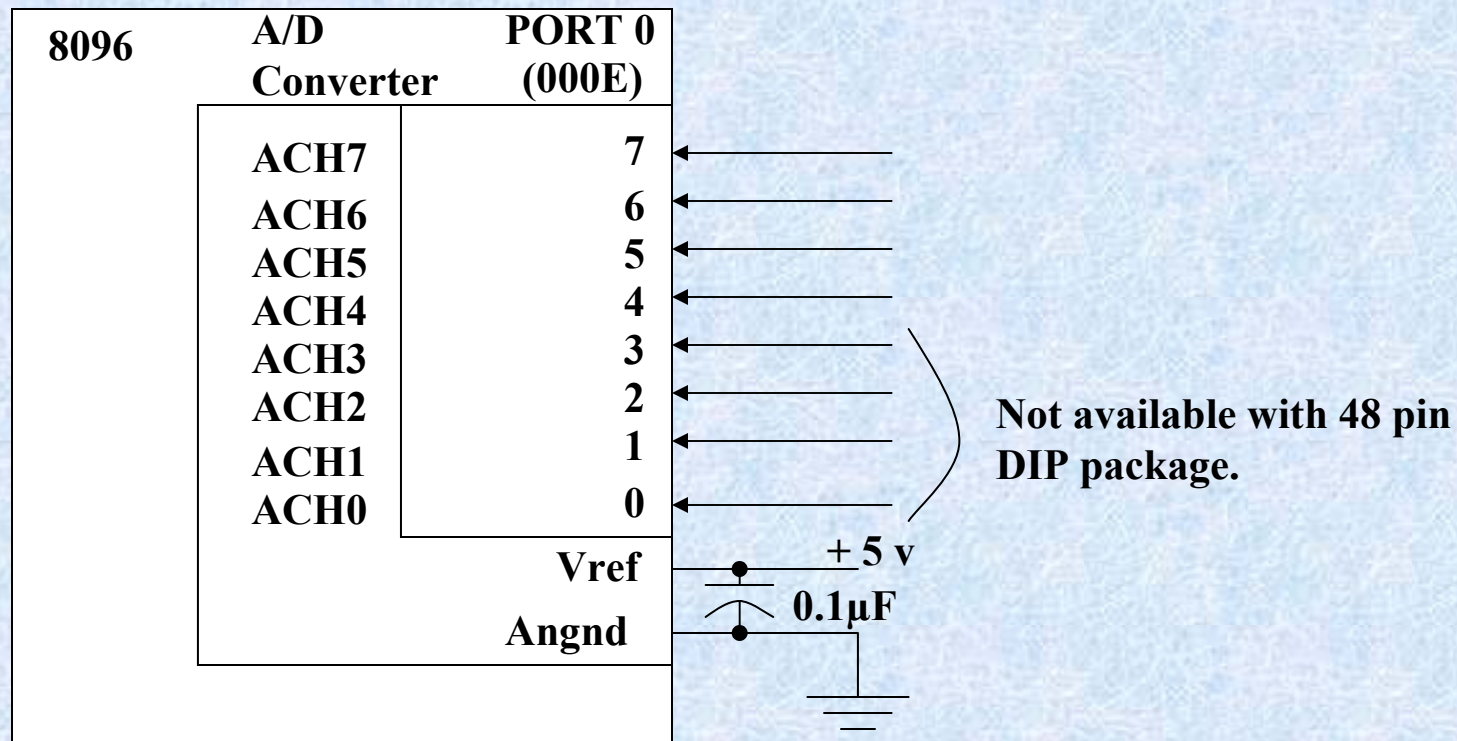
Intel 8096 expanded memory map

General Purpose I/O Ports   (cont..)

- Port 0 whose lines can serve as either general purpose inputs or alternatively as input to the analog-to-digital converter family.

- Port 1 is a quasi-bidirectional I/O port.

- Port 2 includes four input lines, two output lines, and two quasi-bidirectional I/O lines.

- Port 3 and 4 when used as ports, they have open drain outputs.

- By writing anything but a 1 to a line, it can serve as an input even as other lines serve as an outputs.
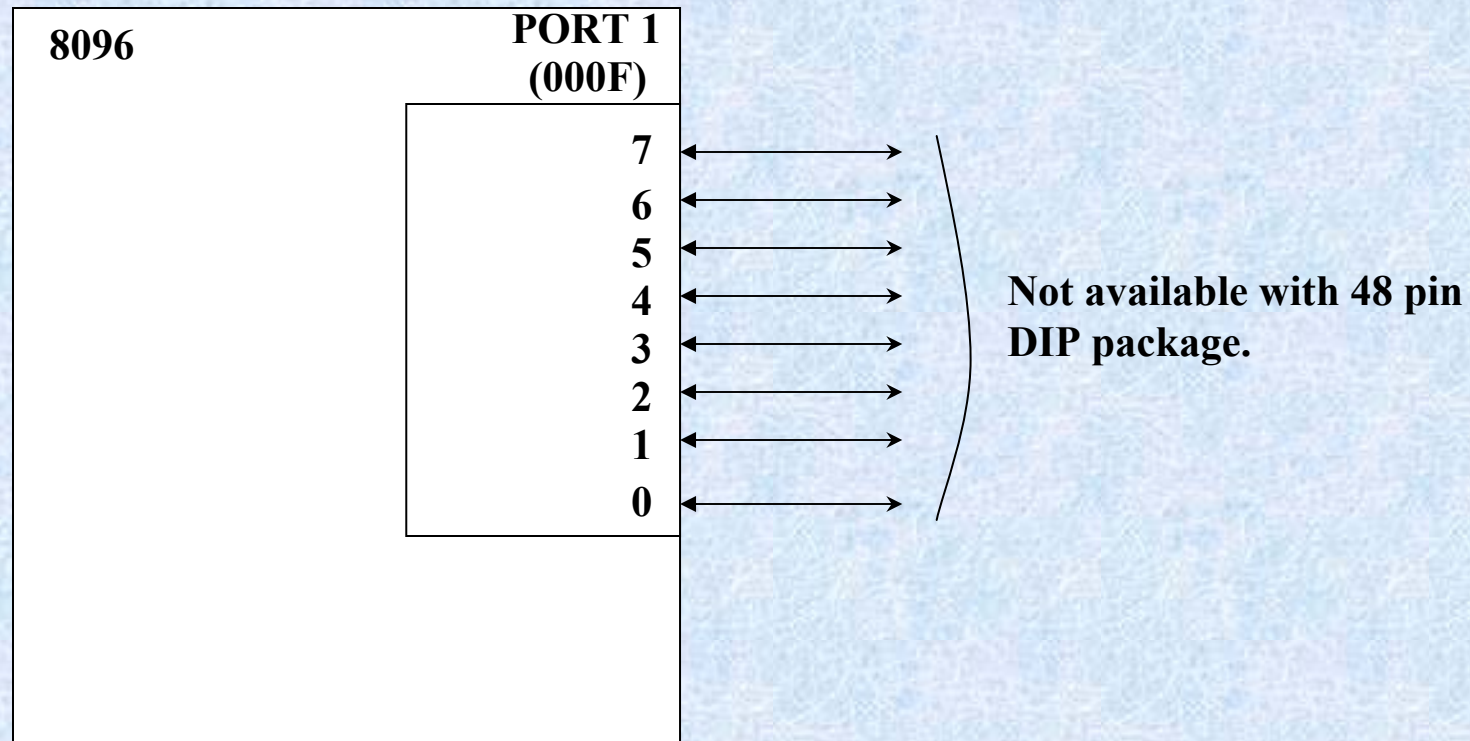
General Purpose I/O Ports.

- Each output line needs the addition of a pullup resistor having a value of 15kΩ.

- In the expanded mode the bus lines gain the ability to drive both high and low, forming the expansion bus without the need of pullup resistors.

General Purpose I/O Ports.

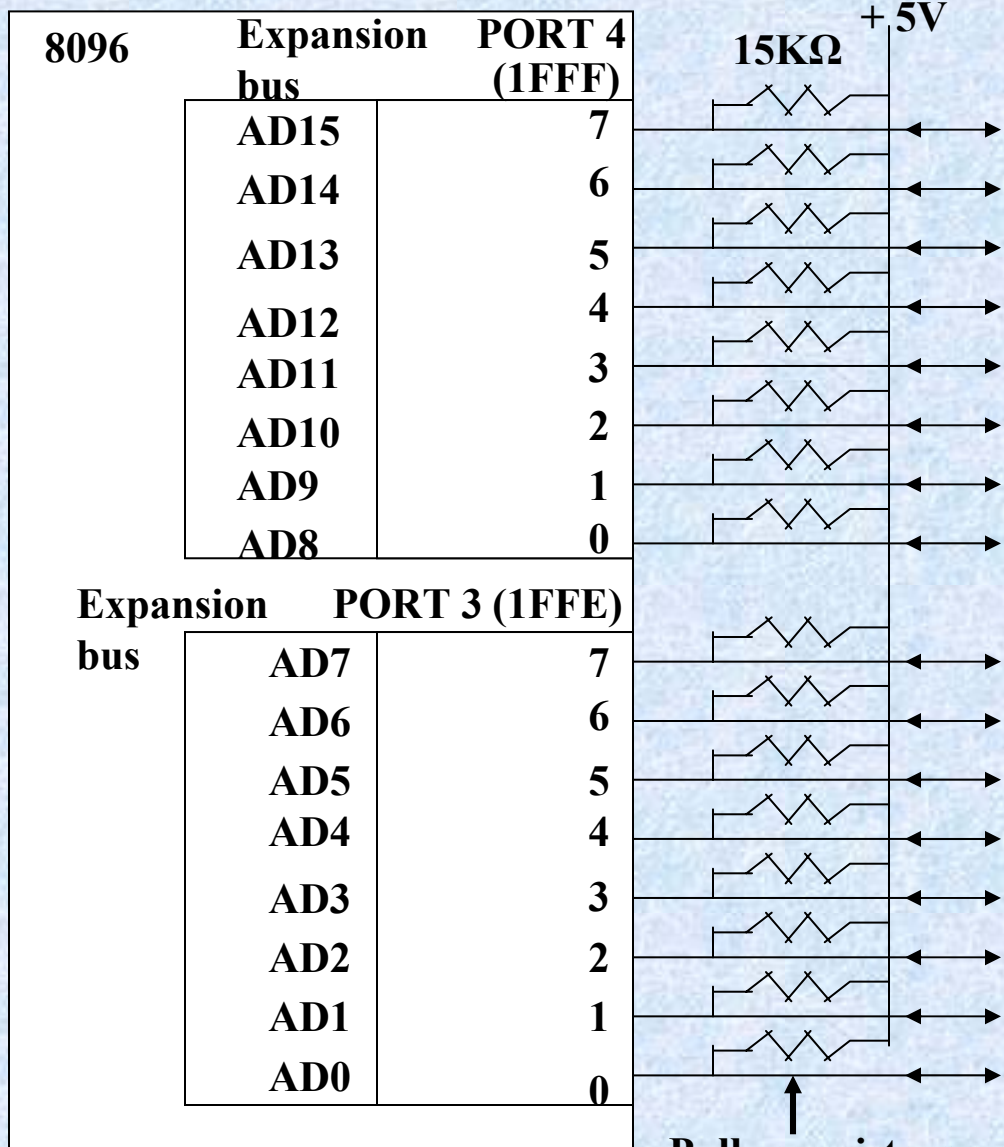| 8096	A/D Converter	PORT 0 (000E)

ACH7	7
ACH6	6
ACH5	5
ACH4	4
ACH3	3
ACH2	2
ACH1	1
ACH0	0

Not available with 48 pin DIP package.

Vref + 5 v

0.1μF

Angnd

Port 0

General Purpose I/O Ports.

8096

PORT 1
(000F)

7
6
5
4
3
2
1
0

Not available with 48 pin
DIP package.

Port 1 use as a quasi-bidirectional I/O port.

General Purpose I/O Ports.

Bit 6 and 7 are quasi bidirectional I/O lines with same behavior and drive characteristic as port 1 lines.

Port 2

Port 3

Pullup resistors are only needed on lines
to be used as output port lines