CPE 408330
Assembly Language and Microprocessors

Chapter 8: THE 8088 AND 8086 MICROPROCESSORS AND THEIR MEMORY AND INPUT/OUTPUT INTERFACES

[Computer Engineering Department,
Hashemite University, © 2008]

Lecture Outline

- 8.1 The 8088 and 8086 Microprocessors
- 8.2 Minimum-Mode and Maximum-Mode System
- 8.3 Minimum-Mode Interface
- 8.4 Maximum-Mode Interface
- 8.5 Electrical Characteristics
- 8.6 System Clock
- 8.7 Bus Cycle and Time States
- 8.8 Hardware Organization of the Memory Address Space

Lecture Outline

- 8.9 Memory Bus Status Codes
- 8.10 Memory Control Signals
- 8.11 Read and Write Bus Cycles
- 8.12 Memory Interface Circuits
- 8.13 Programmable Logic Arrays
- 8.14 Types of Input/Output
- 8.15 An Isolated Input/Output Interface
- 8.16 Input/Output Data Transfer
- 8.17 Input/Output Instructions
- 8.18 Input/Output Bus Cycles

8.1 The 8088 and 8086 Microprocessors

- The 8086, announced in 1978, was the first 16-bit microprocessor introduced by Intel Corporation.
- 8086 and 8088 are internally 16-bit MPU. However, externally the 8086 has a 16-bit data bus and the 8088 has an 8-bit data bus.

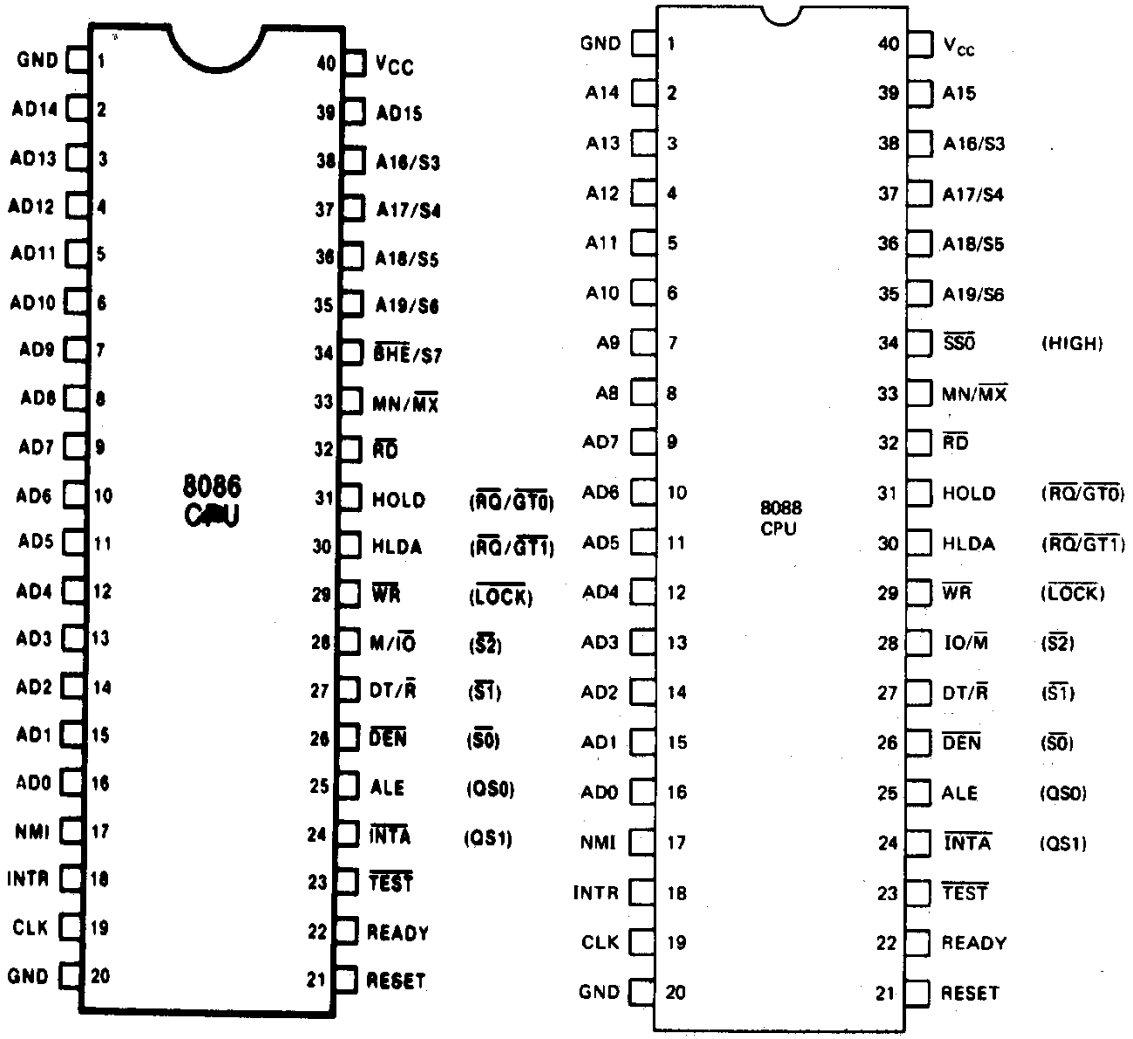
8.1 The 8088 and 8086 Microprocessors

- 8086 and 8088 both have the ability to address up to 1 Mbyte of memory and 64K of input/output port.
- The 8088 and 8086 are both manufactured using ***high-performance metal-oxide semiconductor*** **(HMOS) *technology.***
- The 8088 and 8086 are housed in a 40-pin dual inline package and many pins have multiple functions.

8.1 The 8088 and 8086 Microprocessors

- **CMOS, Complementary Metal-Oxide-Semiconductor, is a major class of integrated** circuits used in chips such as microprocessors, microcontrollers, static RAM, digital logic circuits, and analog circuits such as image sensors.
- Two important characteristics of CMOS devices are high noise immunity and low static power supply drain. Significant power is only drawn when its transistors are switching between on and off states; consequently, CMOS devices do not produce as much heat as other forms of logic such as TTL. CMOS also allows a high density of logic functions on a chip.

8.1 The 8088 and 8086 Microprocessors



Pin layout of the 8086 and 8088 microprocessor

- Pin functions
- Most pins are independent and serve a single function
- Examples:

CLK—clock

INTR—interrupt request

READY—bus ready

- Some multi-functions pins-different times/different mode
- Examples:

AD_0-AD_{15}- multiplexed address/data lines at different times

A_{16}/S_3—multiplexed address and status line at different times

IO/M* or S_2* Control line in one mode or bus

8.2 Minimum-Mode and Maximum-Mode Systems

- The 8086 and 8088 microprocessors can be configured to work in either of two modes:
  - The minimum mode – MN/MX'=1
  - The maximum mode – MN/MX'=0
- The mode selection feature lets the 8088 or 8086 better meet the needs of a wide variety of system requirement.
- Minimum mode 8088/8086 systems are typically smaller and contain a single processor.
- Depending on the mode of operation selected, the assignment for a number of the pins on the microprocessor package are changed.

8.2 Minimum-Mode and Maximum-Mode Systems

Common signals		
Name	Function	Type
AD7-AD0	Address/data bus	Bidirectional, 3-state
A15-A8	Address bus	Output, 3-state
A19/S6-A16/S3	Address/status	Output, 3-state
MN/$\overline{MX}$	Minimum/maximum Mode control	Input
$\overline{RD}$	Read control	Output, 3-state
$\overline{TEST}$	Wait on test control	Input
READY	Wait state control	Input
RESET	System reset	Input
NMI	Nonmaskable Interrupt request	Input
INTR	Interrupt request	Input
CLK	System clock	Input
V_{CC}	+5 V	Input
GND	Ground	

(a)

Signals common to both minimum and maximum mode

- 8088 signals/pins categorized as
  - Common—same function both modes
    Examples: Pin 9 (AD_7)- pin 16 (AD_0)
  - Minimum Mode—special minimum mode operations
    Examples: pins 26-28 are DEN*, DT/R*, and IO/M*
  - Maximum Mode—special maximum mode operations
    Example: pins 26-28 are S_0*, S_1*, and S_2*

8.2 Minimum-Mode and Maximum-Mode Systems

Minimum mode signals (MN/$\overline{MX}$ = V_{CC})		
Name	Function	Type
HOLD	Hold request	Input
HLDA	Hold acknowledge	Output
$\overline{WR}$	Write control	Output, 3-state
IO/$\overline{M}$	IO/memory control	Output, 3-state
DT/$\overline{R}$	Data transmit/receive	Output, 3-state
$\overline{DEN}$	Data enable	Output, 3-state
$\overline{SSO}$	Status line	Output, 3-state
ALE	Address latch enable	Output
$\overline{INTA}$	Interrupt acknowledge	Output

(b)

Maximum mode signals (MN/$\overline{MX}$ = GND)		
Name	Function	Type
$\overline{RQ}$/$\overline{GT1}$, 0	Request/grant bus access control	Bidirectional
$\overline{LOCK}$	Bus priority lock control	Output, 3-state
$\overline{S2}$-$\overline{S0}$	Bus cycle status	Output, 3-state
QS1, QS0	Instruction queue status	Output

(c)

(b)Unique minimum-mode signals (c) Unique maximum-mode signals

8.2 Minimum-Mode and Maximum-Mode Systems

- EXAMPLE

  Which pins provide different signal functions in the minimum-mode 8088 and minimum-mode 8086?

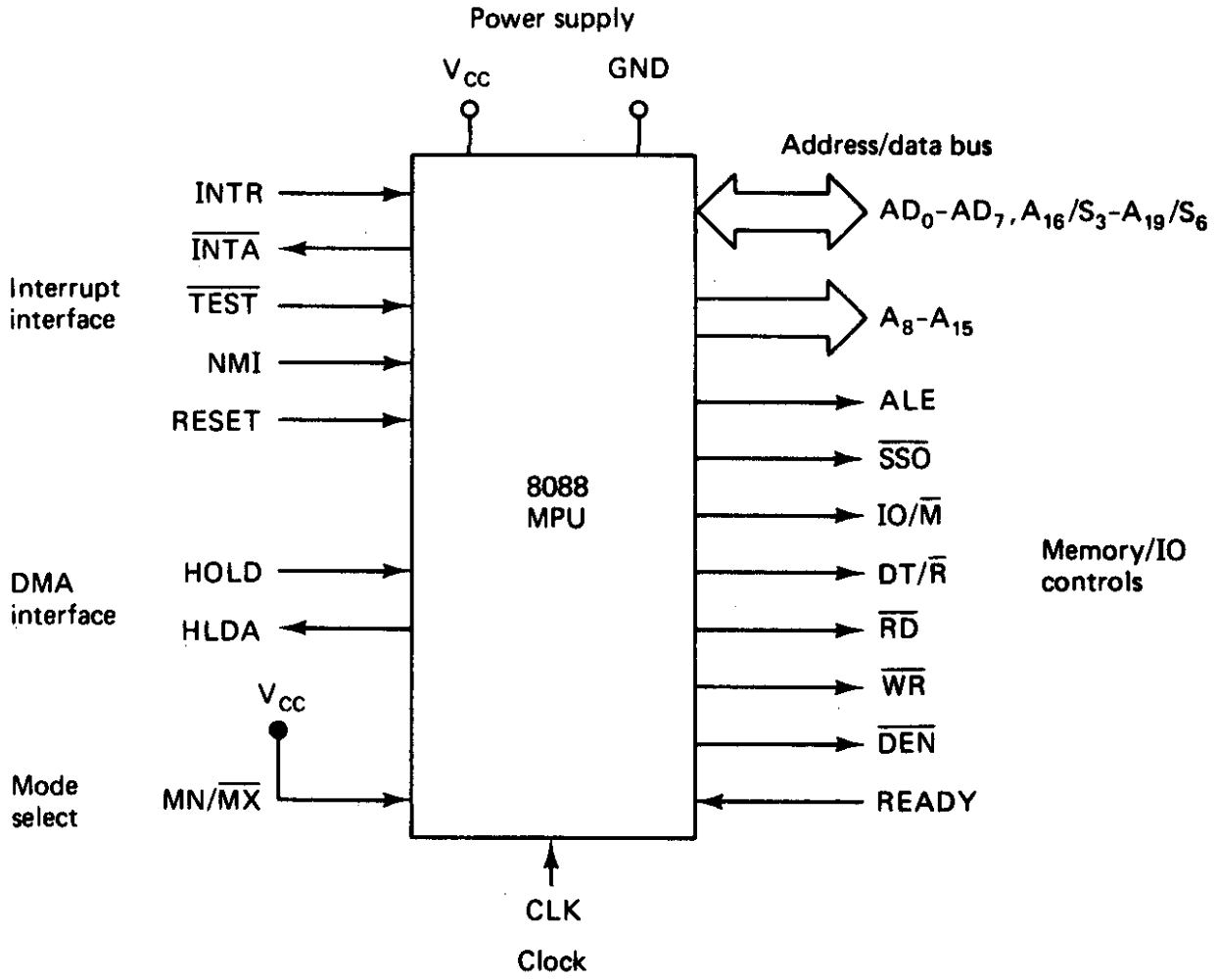
- Solution:

(a) Pins 2 through 8 on the 8088 are address lines A_{14} through A_8, but on the 8086 they are address/data lines AD_{14} through AD_8.

(b) Pin 28 on the 8088 is IO/M' output and on the 8086 it is the M/IO' output.

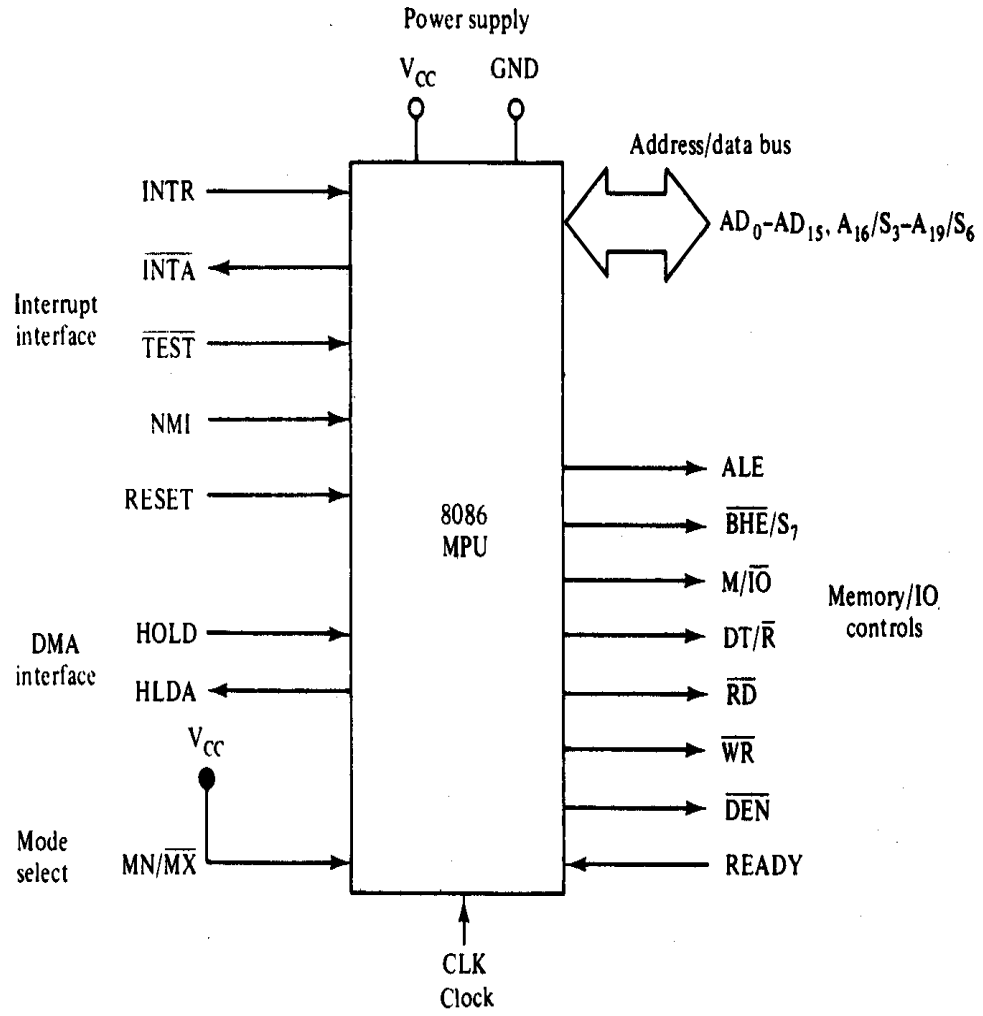
(c) Pin 34 of the 8088 is the SSO' output, and on the 8086 this pin supplies the BHE'/S_7.

8.3 Minimum-Mode Interface



Block diagram of the minimum-mode 8088 MPU

8.3 Minimum-Mode Interface-Differences



Block diagram of the minimum-mode 8086 MPU

- Data bus
  - 16-bit wide
  - D_{15}-D_0
  - Multiplexed with A_{15} through A_0
  - Allows 3 types of data transfers
    - Word—over D_{15}-D_0
    - Low byte—over D_7-D_0
    - High byte—over D_{15}-D_8
- Memory/IO Controls
  - SSO* → BHE* (bank high enable)
    - Used to signal external circuitry whether or not a byte transfer is taking place over the upper 8 data bus lines
  - A0 now does the same for a byte transfer over the lower 8 data bus line

8.3 Minimum-Mode Interface

- The minimum-mode signals can be divided into the following basic groups:
  - Address/Data bus
  - Status signals
  - Control signals
  - Interrupt signals
  - DMA interface signals

8.3 Minimum-Mode Interface

- Address/Data bus
  - The address bus is used to carry address information to the memory and I/O ports.
  - The address bus is 20-bit long and consists of signal lines A_0 through A_{19}.
  - A 20-bit address gives the 8088 a 1 MByte memory address space.
  - Only address line A_0 through A_{15} are used when addressing I/O. This give an I/O address space of 64 Kbytes.
  - The 8088 has 8 multiplexed address/data bus lines (A_0~A_7) while 8086 has 16 multiplexed address/data bus lines (A_0~A_{15}).

8.3 Minimum-Mode Interface

S6 S5 S4 S3

- Status signals
  - The four most significant address, A_{19} through A_{16} are multiplexed with ***status signal S_6 through S_3.***
  - Bits S_4 and S_3 together form a 2-bit binary code that identifies which of the internal segment registers was used to generate the physical address. S_5 is the logic level of the internal interrupt flag. S_6 is always at the 0 logic level.

S_4	S_3	Address Status
0	0	Alternate (relative to the ES segment)
0	1	Stack (relative to the SS segment)
1	0	Code/None (relative to the CS segment or a default of zero)
1	1	Data (relative to the DS segment)

8.3 Minimum-Mode Interface

- Control signals
  - The ***control signals*** are provided to support the memory and I/O interfaces of the 8088 and 8086.
    - ALE – Address Latch Enable
    - IO/M' – IO/Memory (8088)
    - M/IO' – Memory/IO (8086)
    - DT/R' – Data Transmit/Receive (8088/8086)
    - SSO' – System Status Output (8088)
    - BHE' – Bank High Enable (8086)
    - RD' – Read (8088/8086)
    - WR' – Write (8088/8086)
    - DEN' – Data Enable (8088/8086)
    - READY – Ready (8088/8086)

8.3 Minimum-Mode Interface

- Interrupt signals
  - The ***interrupt signals*** can be used by an external device to signal that it needs to be serviced.
    - INTR' – Interrupt Request
    - INTA' – Interrupt Acknowledge
    - TEST' – Test (can be use to synchronize MPU)
    - NMI – Nonmaskable Interrupt
    - RESET – Reset (hardware reset of the MPU)

8.3 Minimum-Mode Interface

- DMA interface signals
  - When an external device wants to take control of the system bus, it signals this fact to the MPU by switching HOLD to the 1 logic level.
  - When in the hold state, signal lines AD_0 through AD_7, A_8 through A_{15}, A_{16}/S_3 through A_{19}/S_6, SSO', IO/M', DT/R', RD', WR', DEN', and INTR are all put into high-Z state.
  - The 8088 signals external devices that the signal lines are in the high-Z state by switching its HLDA output to the 1 logic level.

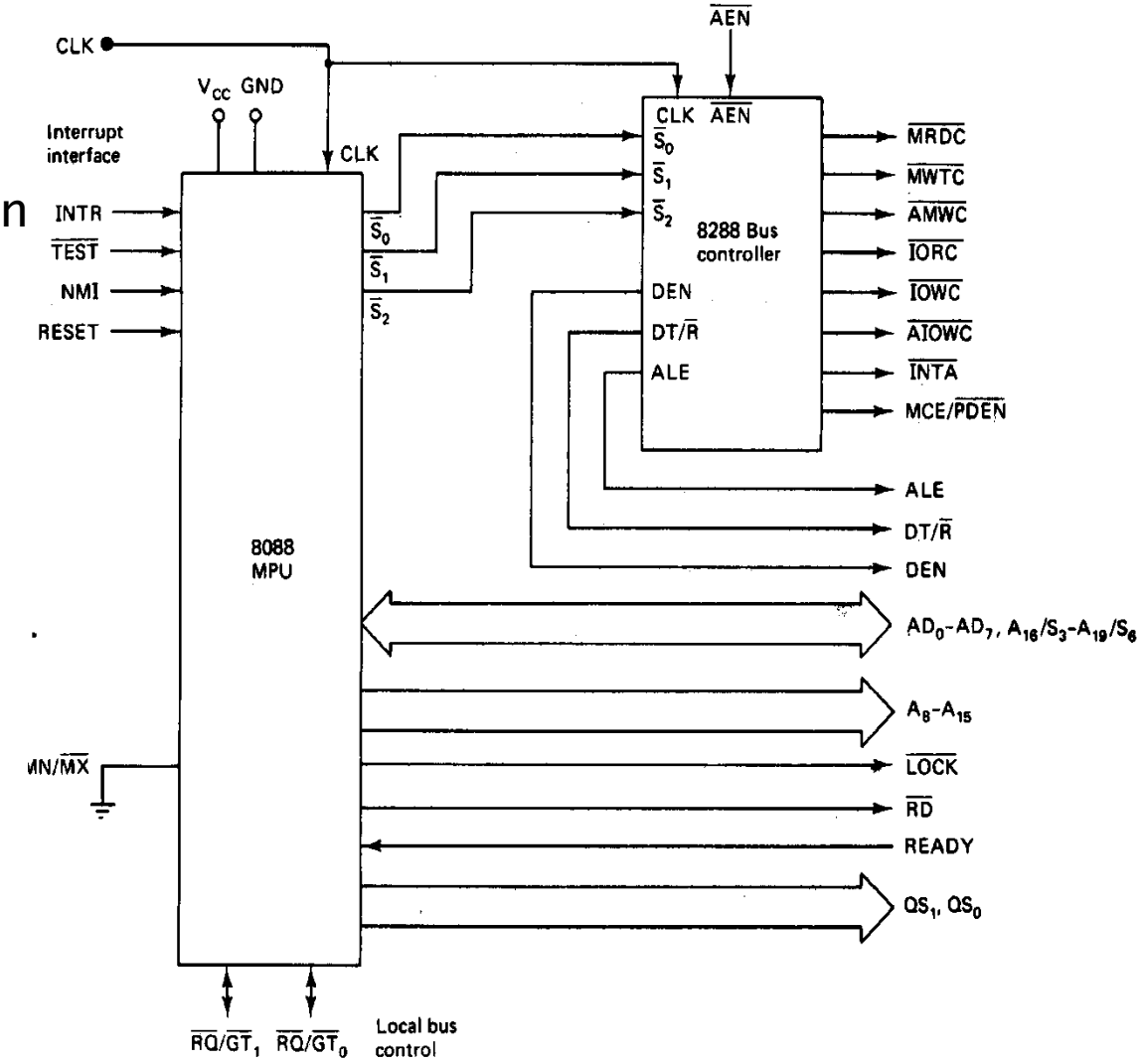
8.4 Maximum-Mode Interface

- The maximum-mode configuration is mainly used for implementing a ***multiprocessor/coprocessor system environment.***
- Global resources and local resources
- In the maximum-mode, facilities are provided for implementing allocation of global resources and passing bus control to other microprocessors sharing the system bus.

8.4 Maximum-Mode Interface

- *8288 bus controller:*

- Maximum-mode configuration
  - MN/MX* pin = 0 → GND
  - Most memory, IO, and interrupt interface outputs produced by an external 8288 bus controller

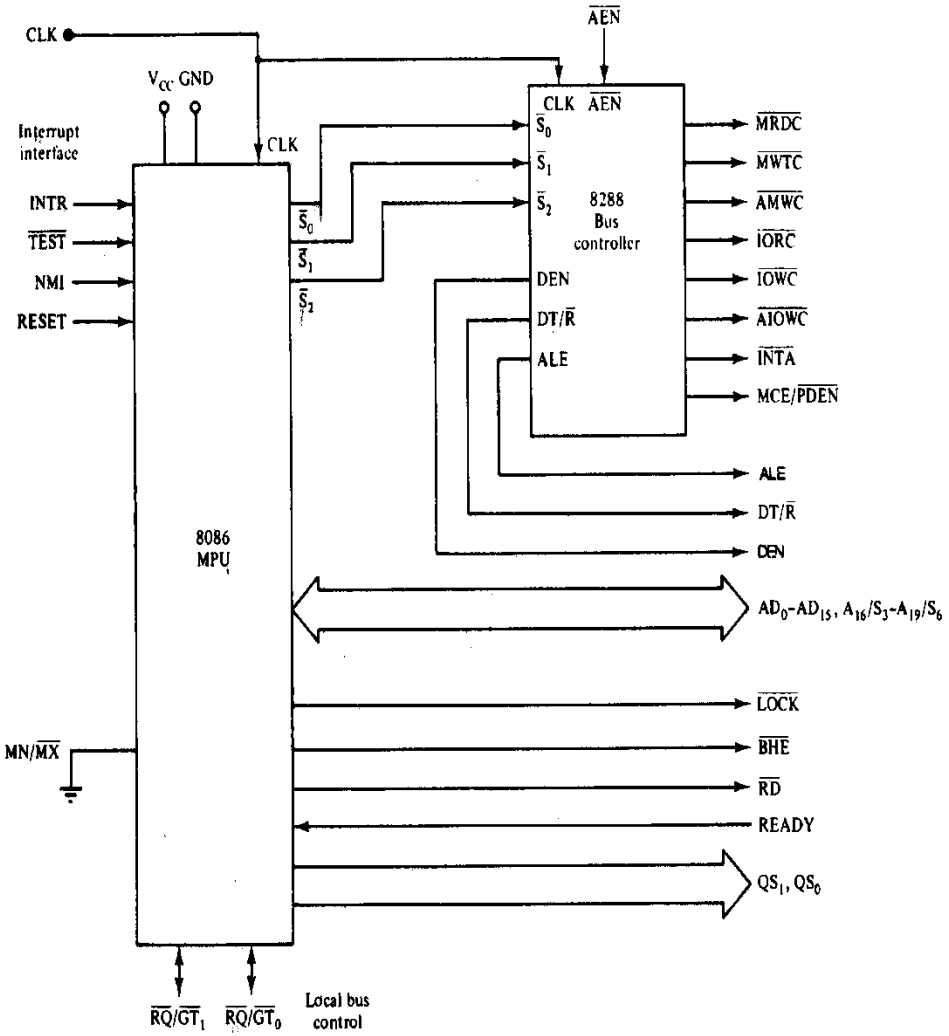


8088 maximum-mode block diagram

8.4 Maximum-Mode Interface

- *8288 bus controller:*

- Differences from 8088 Maximum mode interface
  - 16-bit multiplexed data bus
  - BHE* output



8086 maximum-mode block diagram

8.4 Maximum-Mode Interface

- 8288 bus controller
  - In the maximum-mode, 8088/8086 outputs a status code on three signal line, S_0, S_1, S_2, prior to the initialization of each bus cycle.
  - The 3-bit bus status code identifies which type of bus cycle is to follow and are input to the external bus controller device, 8288.
  - The 8288 produces one or two command signals for each bus cycle.

8.4 Maximum-Mode Interface

- *8288 bus controller:*

Status Inputs			CPU Cycle	8288 Command
$\overline{S2}$	$\overline{S1}$	$\overline{S0}$		
0	0	0	Interrupt Acknowledge	$\overline{INTA}$
0	0	1	Read I/O Port	$\overline{IORC}$
0	1	0	Write I/O Port	$\overline{IOWC}$, $\overline{AIOWC}$
0	1	1	Halt	None
1	0	0	Instruction Fetch	$\overline{MRDC}$
1	0	1	Read Memory	$\overline{MRDC}$
1	1	0	Write Memory	$\overline{MWTC}$, $\overline{AMWC}$
1	1	1	Passive	None

Bus status code

8.4 Maximum-Mode Interface

- 8288 bus controller connection
  - Inputs are codes from the 3-bit bus status lines S_2*S_1*S_0* = bus status code
  - Outputs produced by 8288 instead of 8088
    - Based on bus status code → active 0

      MRDC*= Memory read command

      MWTC*= Memory write command

      AMWC*= Advanced memory write command

      IORC*= I/O read command

      IOWC*= I/O write command

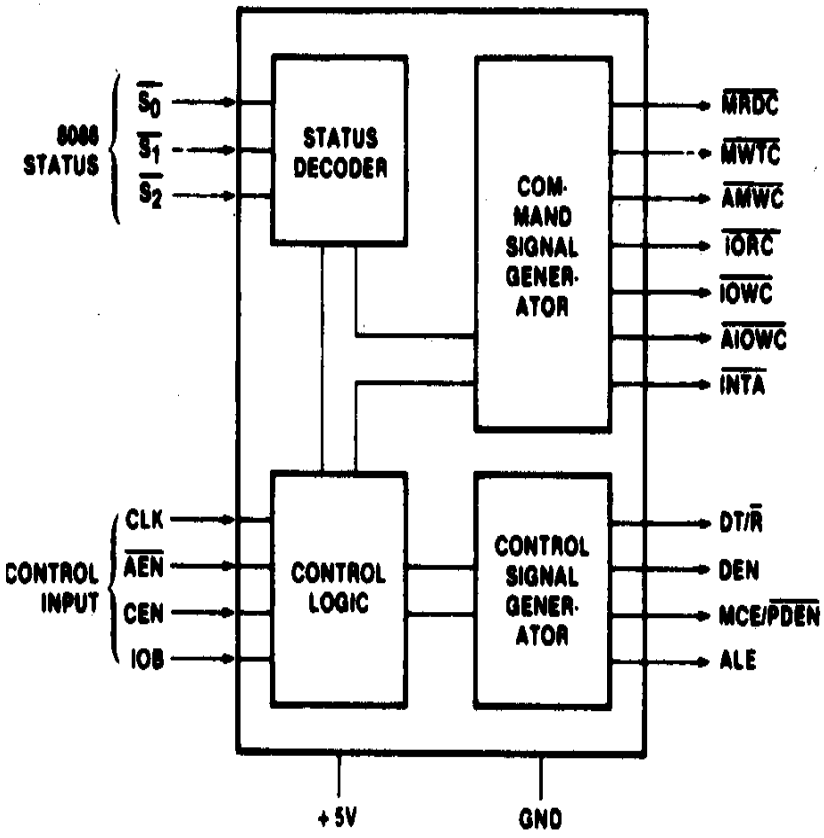
      AIOWC*= advanced I/O write command
  - Produced for all bus cycles

    ALE= Address latch enable

    DT/R*= Data transmit/receive

    DEN= Data enable (complement)

    INTA*= Interrupt acknowledge



Block diagram of 8288

8.4 Maximum-Mode Interface

- Lock signal
  - The lock signal (LOCK') is meant to be output (logic 0) whenever the processor wants to lock out the other processor from using the bus.
- Local bus control signals
  - The request/grant signals (RQ'/GT'$_0$, RQ'/GT'$_1$) provide a prioritized bus access mechanism for accessing the local bus.

8.4 Maximum-Mode Interface

- Queue status signals
  - The 2-bit queue status code QS_0 and QS_1 tells the external circuitry what type of information was removed form the queue during the previous clock cycle.

QS1	QS0	Queue Status
0 (low)	0	No Operation. During the last clock cycle, nothing was taken from the queue.
0	1	First Byte. The byte taken from the queue was the first byte of the instruction.
1 (high)	0	Queue Empty. The queue has been reinitialized as a result of the execution of a transfer instruction.
1	1	Subsequent Byte. The byte taken from the queue was a subsequent byte of the instruction.

Queue status code

8.4 Maximum-Mode Interface

- EXAMPLE

  If the bus status code $S'_2S'_1S'_0$ equals 101, what type of bus activity is taking place? Which command output is produced by the 8288?

- Solution:

  Looking at the bus status table, we see that bus status code 101 identifies a read memory bus cycle and causes the MRDC' output of the bus controller to switch to logic 0.

8.5 Electrical Characteristics

- Power is applied between pin 40 (Vcc) and pins 1 (GND) and 20 (GND) (pin1 and 20 are connected together).
- The nominal value of Vcc is specified as +5V dc with a tolerance of ±10% (4.5V~5.5V will work correctly)
- Both 8088 and 8086 draw a maximum of 340mA from the supply.

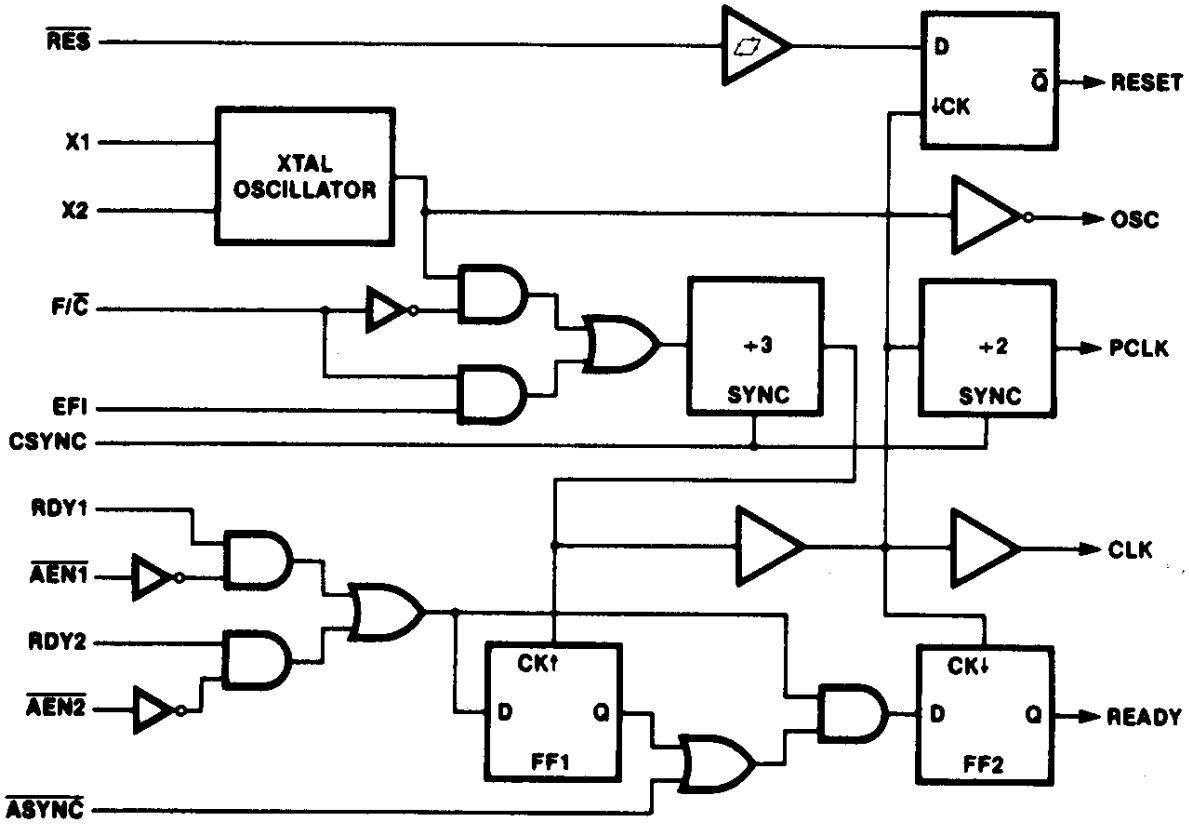
Symbol	Meaning	Minimum	Maximum	Test condition
V_{IL}	Input low voltage	−0.5 V	+0.8 V	
V_{IH}	Input high voltage	+2.0 V	V_{cc} + 0.5 V	
V_{OL}	Output low voltage		+0.45 V	I_{OL} = 2.0mA
V_{OH}	Output high voltage	+2.4 V		I_{OH} = −400 uA

8.6 System Clock

- The time base for synchronization of the internal and external operations of the microprocessor in a microcomputer system is provided by the *clock (CLK)* input signal.
- 8088 is available in two speeds. The standard 8088 operates at 5 MHz and the 8088-2 operates at 8 MHz.
- The 8086 is manufactured in three speeds: 5-MHz 8086, 8-MHz 8086-2, and the 10-MHz 8086-1.
- The CLK is externally generated by the 8284 clock generator and driver IC.

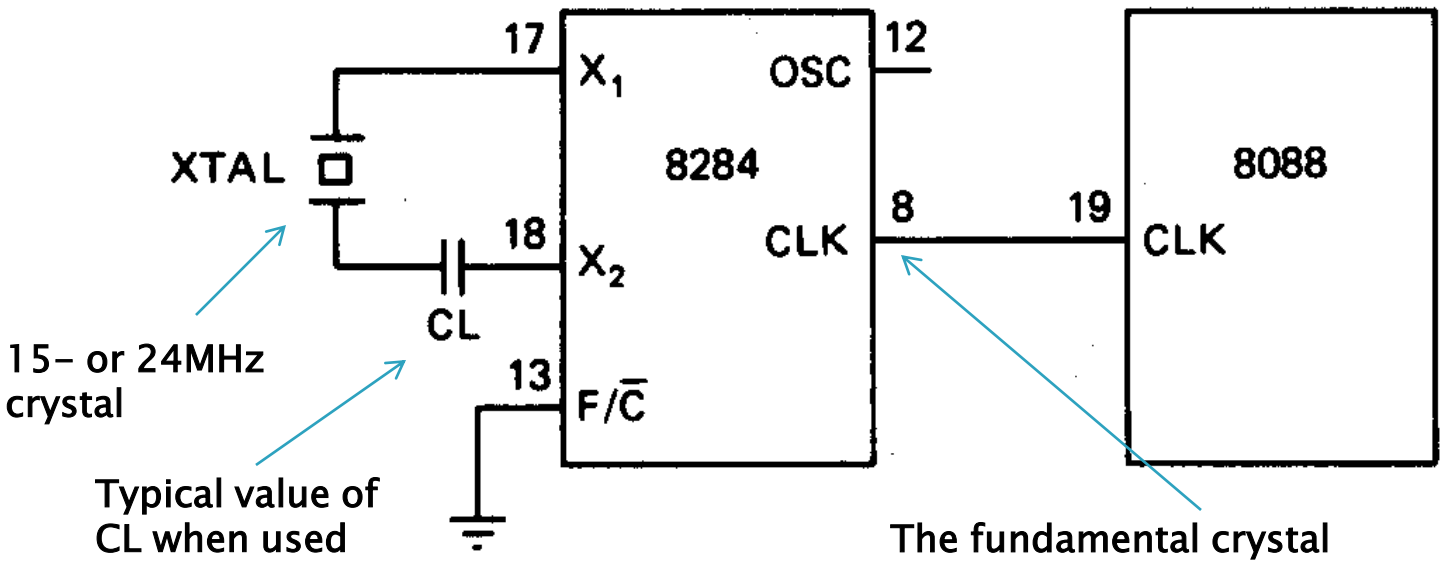
8.6 System Clock

- Block diagram of the 8284 clock generator



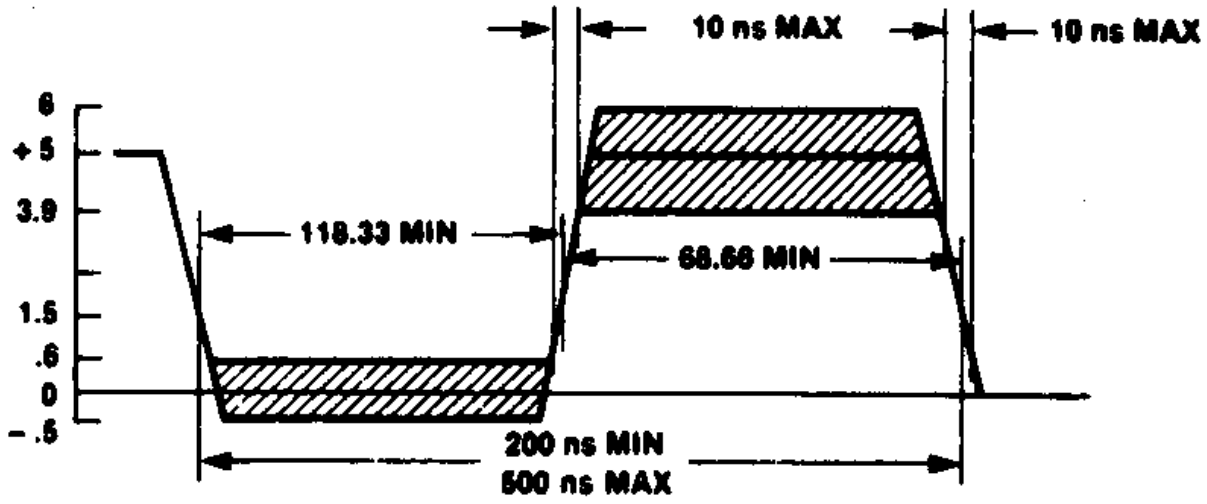
8.6 System Clock

- Connecting the 8284 to the 8088



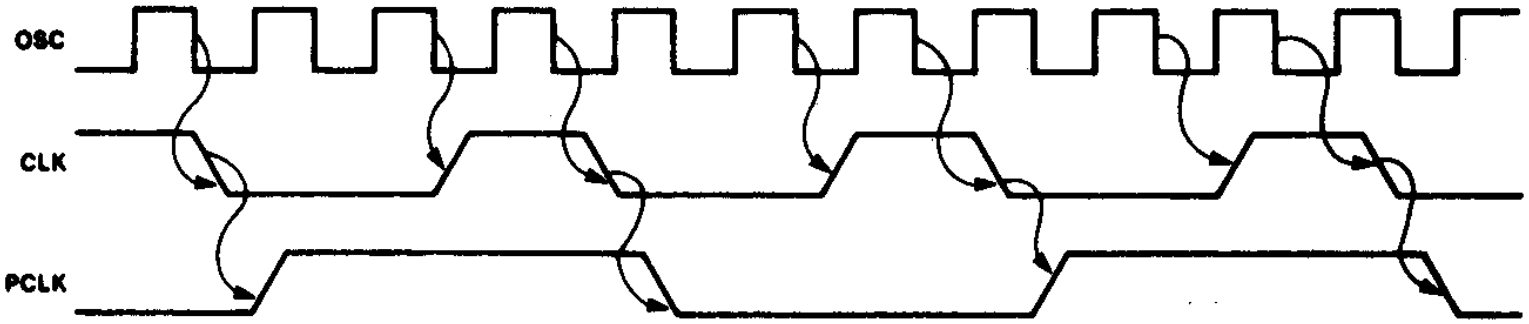
8.6 System Clock

- CLK waveform
  - The signal is specified at Metal Oxide Semiconductor (MOS)-compatible voltage level (rather than TTL)
  - The period of the 5-MHz 8088 can range from 200 ns to 500 ns, and the maximum rise and fall times of its edges equal 10 ns.



8.6 System Clock

- PCLK and OSC signals
  - The peripheral clock (PCLK) and oscillator clock (OSC) signals are provided to drive peripheral ICs.
  - The clock output at PCLK is half the frequency of CLK. The OSC output is at the crystal frequency which is three times of CLK.



8.6 System Clock

- EXAMPLE

  If the CLK input of an 8086 MPU is to be driven by a 9-MHz signal, what speed version of the 8086 must be used and what frequency crystal must be attached to the 8284

- Solution:

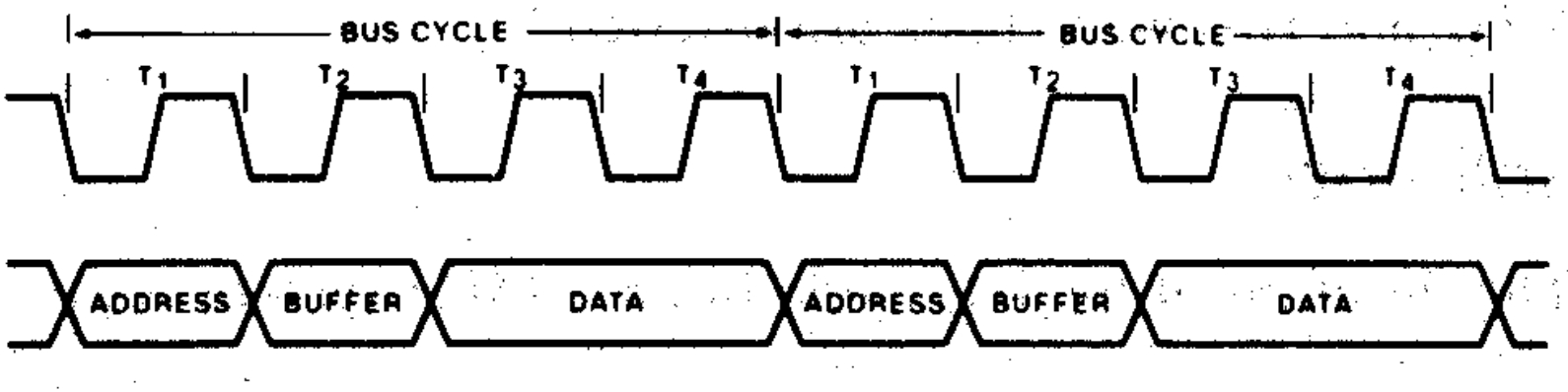
  The 8086-1 is the version of the 8086 that can be run at 9-MHz. To create the 9-MHz clock, a 27-MHz crystal must be used on the 8284.

8.7 Bus Cycle and Time States

- A bus cycle defines the basic operation that a microprocessor performs to communicate with external devices.
- Examples of bus cycles are the memory read, memory write, input/output read, and input/output write.
- The bus cycle of the 8088 and 8086 microprocessors consists of at least **four clock periods.**
- If no bus cycles are required, the microprocessor performs what are known as ***idle states.***
- When READY is held at the 0 level, ***wait states*** are inserted between states T_3 and T_4 of the bus cycle.
- For write cycle, at T1 the address is prepared, data is ready at T2 and maintained during T3 and T4.
- For read cycle, at T1 the address is prepared, at T2 the bus is at Z state, and the data is ready at T3 and maintained at T4.

8.7 Bus Cycle and Time States

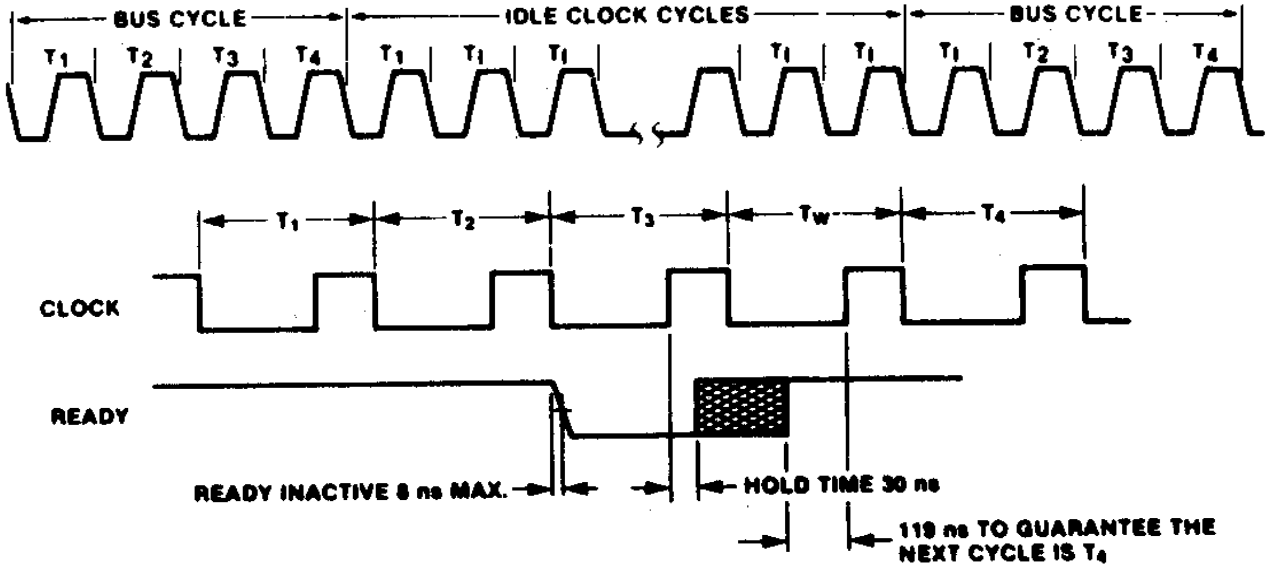
- Multiplexed address/data transfer operation
  - Address output during T1
  - Bus lines in high-Z state in T2
  - Data transfer takes place during states T3 and T4



Bus cycle clock periods

8.7 Bus Cycle and Time States

- Bus cycle with idle
  - If no bus activity is necessary, microprocessor inserts idle states between bus cycles
  - Identified as T1
  - May be due to the fact that the instruction queue is already full so no instructions need to be fetched
- Bus cycle with wait states
  - If the memory or I/O device is not able to respond in the duration of a bus cycle (500ns @8MHz) (slow I/O and memory) , it must make READY 0 during T3 to extend the bus cycle
  - Wait states (Tw) are inserted to extend the bus cycle until READY returns to 1



Bus cycle with idle state, and wait state

8.7 Bus Cycle and Time States

- EXAMPLE

  What is the duration of the bus cycle in the 8088-based microcomputer if the clock is 8 MHz and the two wait states are inserted.

- Solution:

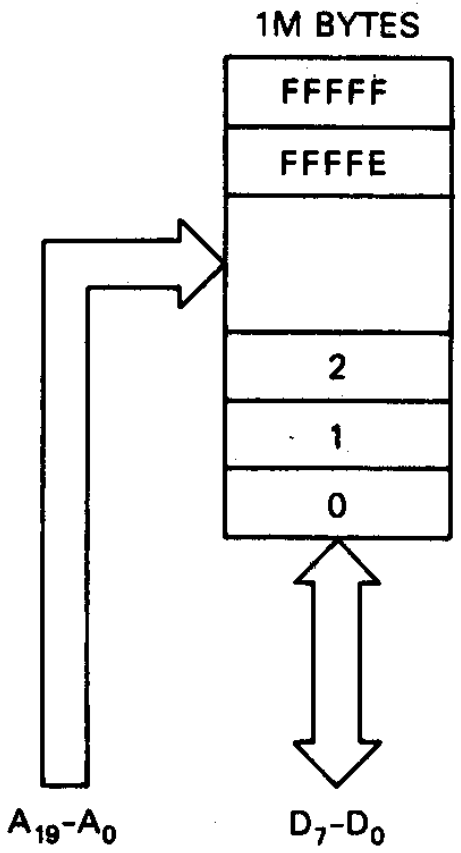
  The duration of the bus cycle in an 8 MHz system is given by

t_{cyc} = 500 ns + N x 125 ns

  In this expression the N stands for the number of waits states. For a bus cycle with two wait states, we get

t_{cyc} = 500 ns + 2 x 125 ns = 500 ns + 250 ns
= 750 ns

8.8 Hardware Organization of the Memory Address Space



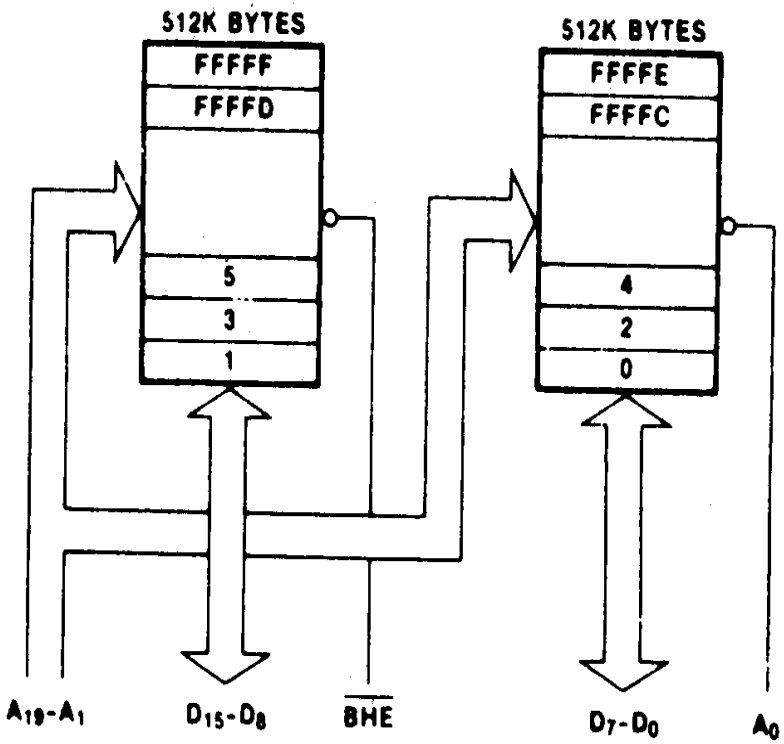
1Mx8 memory bank of the 8088

- 8088 memory hardware is organized as a single byte-wide memory bank
  - Size—1M X 8 bits
  - Physical address range— 0H-FFFFFH
  - Address/data bus de-multiplexed in external hardware
  - Input:
    20-bit address bus— A_{19} through A_0
  - Input/output:
    8-bit data bus—D_7 Through D_0

Memory size = $2^{\text{number of address lines}}$

→ $1M = 2^{20}$ (needs 20 address line)

8.8 Hardware Organization of the Memory Address Space



High and low memory banks of the 8086

Memory size = 2 number of address lines

→ 512K = 2 19 (needs 19 address line)

- 8086 memory hardware is organized as a two bytewide memory bank
- Bank size—512K X 8 bits
- Low-bank holds even addressed bytes—0H through FFFFEH
- High-bank holds odd addressed bytes—1H through FFFFFH
- Address/data bus demultiplexed in external hardware
- Input:

20-bit+ address bus— A_{19} through A_0, and BHE*

A_1-A_{19} = selects storage location

A_0 = 0 enables low bank

BHE* = 0 enables high bank

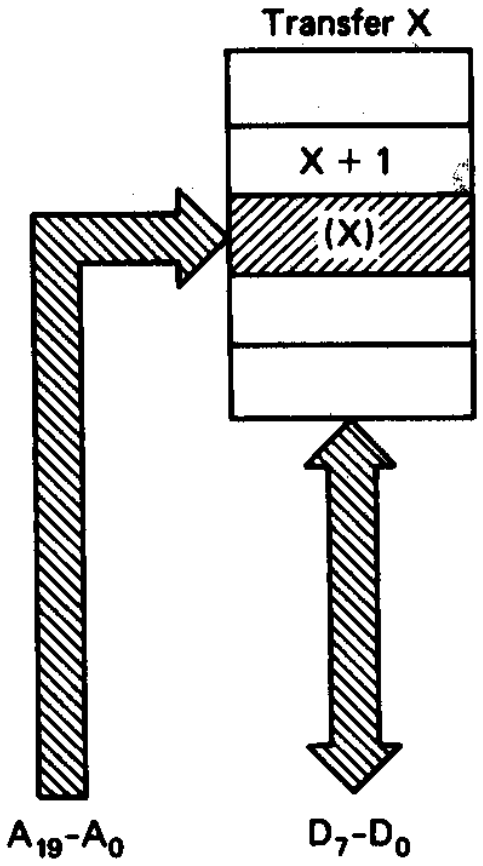
- Input/Output:

16-bit data bus—D_{15} Through D_0

D_7-D_0 → even addressed byte accesses

D_{15}-D_8 → odd addressed byte

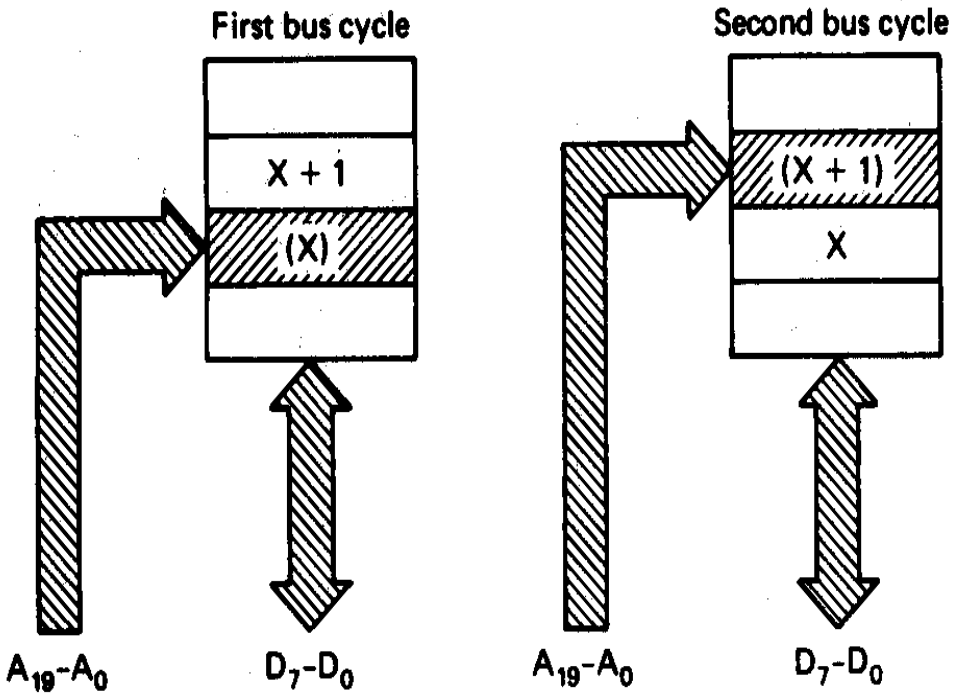
8.8 Hardware Organization of the Memory Address Space



Byte transfer by the 8088

- Byte access bus cycle
  - MPU applies address of storage location to be accessed over address lines A_{19}–A_0
    A_{19}—most significant bit
    A_0—least significant bit
  - Byte of data written into or read from address X transferred over data lines D_0 through D_7
    D_7—most significant bit
    D_0—least significant bit
  - Byte access takes a minimum of one bus cycle of duration
    @5MHz—800ns
    @8MHz—500ns

8.8 Hardware Organization of the Memory Address Space



Word transfer by the 8088

- Word access bus cycles
  - MPU must access two consecutive storage locations in memory—X and X+1
  - Requires two bus cycles
  - Address X accessed during cycle 1
  - Address X+1 accessed during cycle 2
  - Word access duration is a minimum of two bus cycle
    @5MHz—2 X 800ns = 1600ns
    @8MHz—2 X 500ns = 1000ns

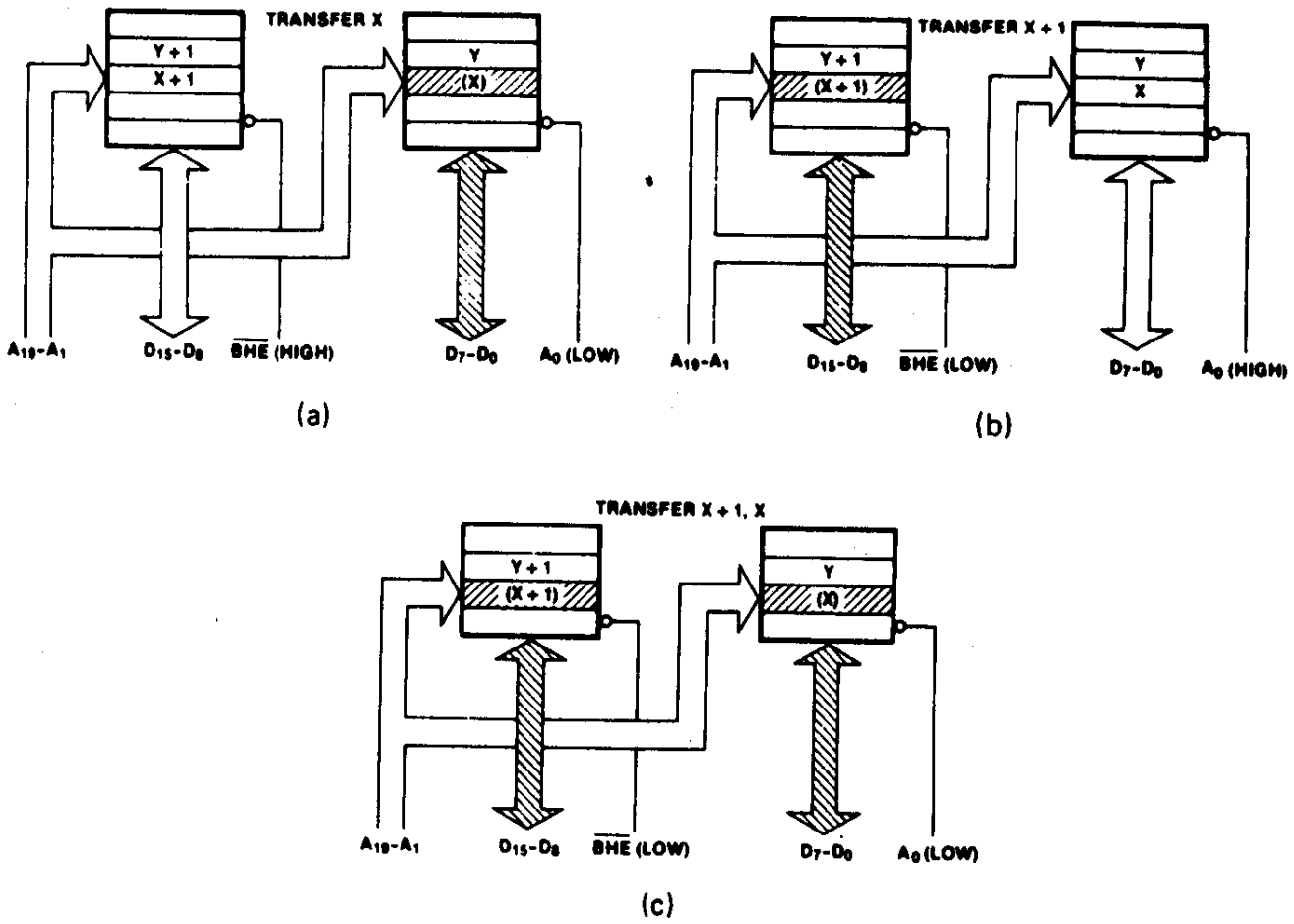
8.8 Hardware Organization of the Memory Address Space

- Low bank byte access bus cycle
  - MPU applies even address X to both banks over address lines A_{19}–A_0
  - MPU enables just the low bank BHE*A_0 = 10 → enables low bank
  - Byte of data written into or read from address X transferred over data lines D_0 through D_7
- High bank access bus cycle differences
  - Odd address X+1 applied to both banks
  - High bank enabled BHE*A_0 =01 = → enable high bank
  - Byte-wide data transfer takes place over data line D_8 through D_{15}
- Word access bus cycle differences
  - Even word address X applied to both banks
  - MPU enables both banks BHE*A_0 =00 = → enable low and high bank
  - Word-wide data transfer takes place over D_0 through D_{15}
  - All accesses takes a minimum of one bus cycle of duration

  @5MHz—800ns

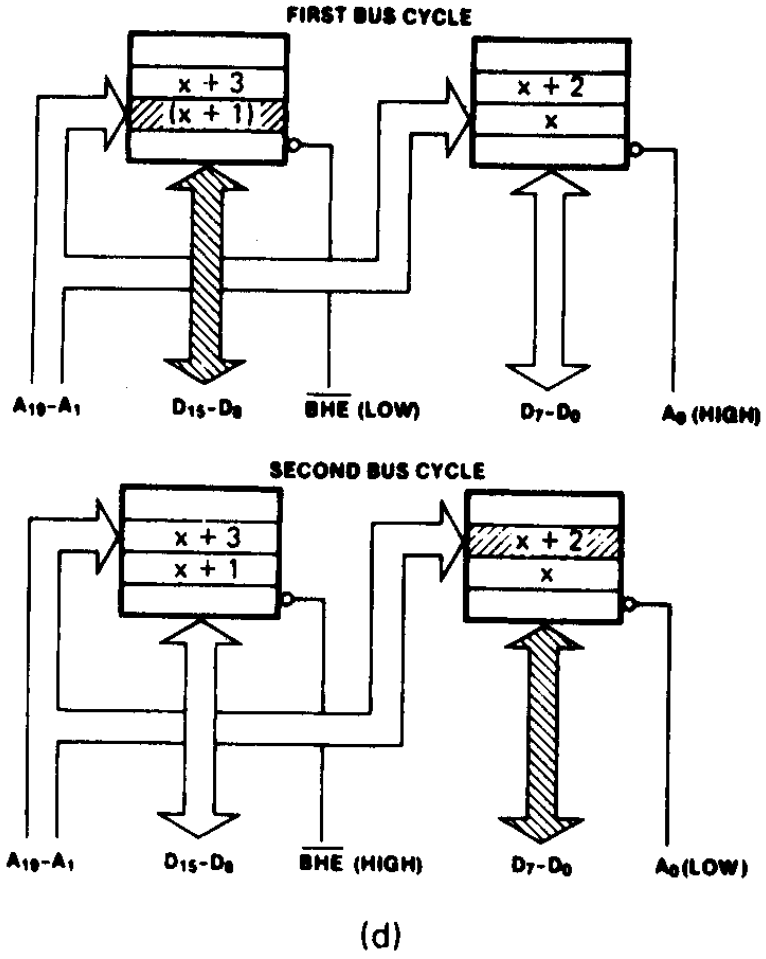
  @8MHz—500ns

8.8 Hardware Organization of the Memory Address Space



(a) Even address byte (b) Odd address byte transfer by the 8086
(c) Even address word transfer

8.8 Hardware Organization of the Memory Address Space



(d)

Odd address word transfer by the 8086

- Misaligned-word access bus cycles
  - Word starting at address X+1 is misaligned
  - Requires two bus cycles
    - Access byte at address X+1 during cycle 1
      A_{19}-A_0 = X+1
      BHE*A_0 =01→ enables high bank
      D_{15}-D_8 → carries data
    - Access byte at address X+2 during cycle 2
      A_{19}-A_0 = X+2
      BHE*A_0 =10 → enables low bank
      D_7-D_0 → carries data
  - Word access duration is a minimum of two bus cycle
    @5MHz—2 X 800ns = 1600ns
    @8MHz—2 X 500ns = 1000ns
  - Impact on performance—software should minimize accessing

8.8 Hardware Organization of the Memory Address Space

- EXAMPLE

  Is the word at memory address 01231_{16} of an 8086-based microcomputer aligned or misaligned? How many cycle are required to read it from memory?

- Solution:

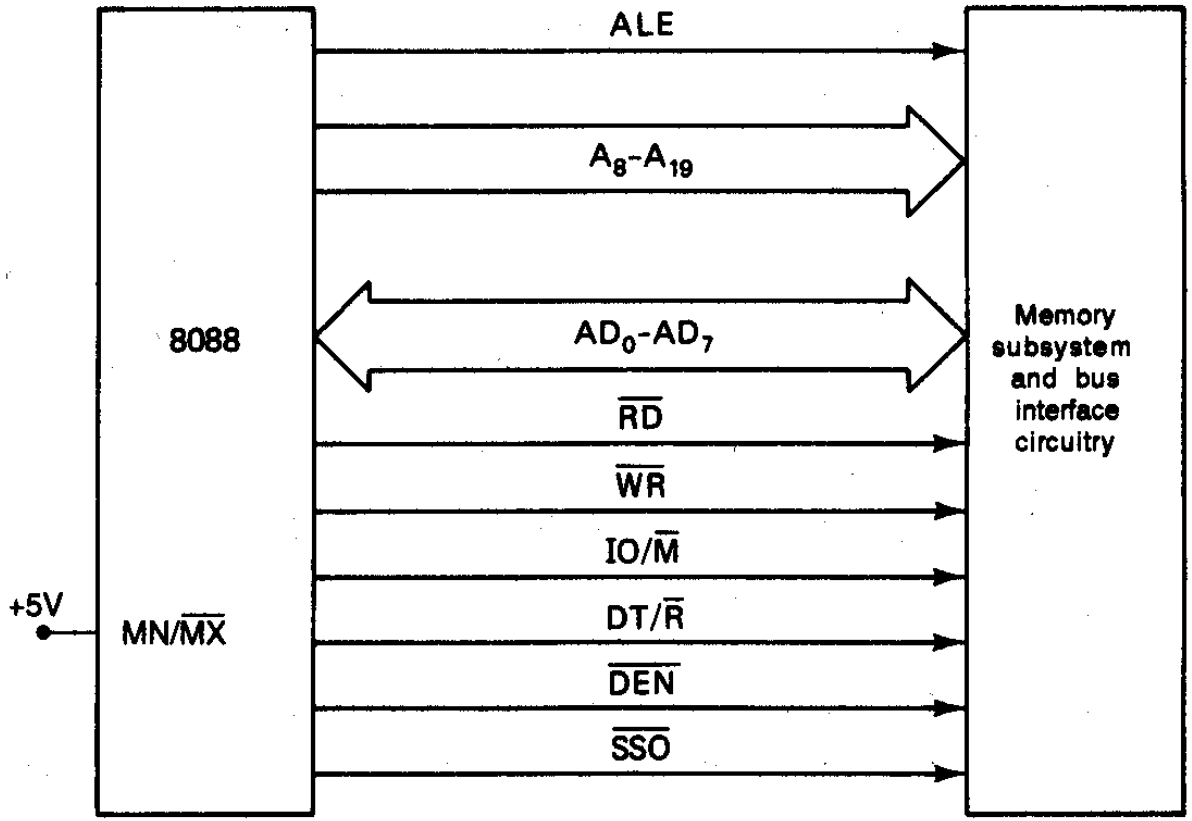
  The first byte of the word is the second byte at the aligned-word address 01230_{16}. Therefore, the word is misaligned and required two bus cycles to be read from memory.

8.9 Address Bus Status Codes

- Whenever a memory bus cycle is in progress, an address bus status code S_4S_3 is output by the processor.
- S_4S_3 identifies which one of the four segment register is used to generate the physical address in the current bus cycle:
- S_4S_3 =00 identifies the extra segment register (ES)
- S_4S_3 =01 identifies the stack segment register (SS)
- S_4S_3 =10 identifies the code segment register (CS)
- S_4S_3 =11 identifies the data segment register (DS)
- Since each combination of S_4S_3 leads to select different memory segment, the memory address reach of the microprocessor can thus be expanded to 4 Mbytes.

8.10 Memory Control Signals

- Minimum-mode memory control signals



8.10 Memory Control Signals

- Minimum-mode memory control signals
  - **ALE – Address Latch Enable** – used to latch the address in external memory (valid address on the bus).
  - **IO/M' – Input-Output/Memory** – signal external circuitry whether a memory of I/O bus cycle is in progress.
  - **DT/R' – Data Transmit/Receive** – signal external circuitry whether the 8088 is transmitting or receiving data over the bus.
  - **RD' – Read** – identifies that a read bus cycle is in progress.
  - **WR' – Write** – identifies that a write bus cycle is in progress.
  - **DEN' – Data Enable** – used to enable the data bus.
  - **SSO' – Status Line** – identifies whether a code or data access is in progress.

8.10 Memory Control Signals

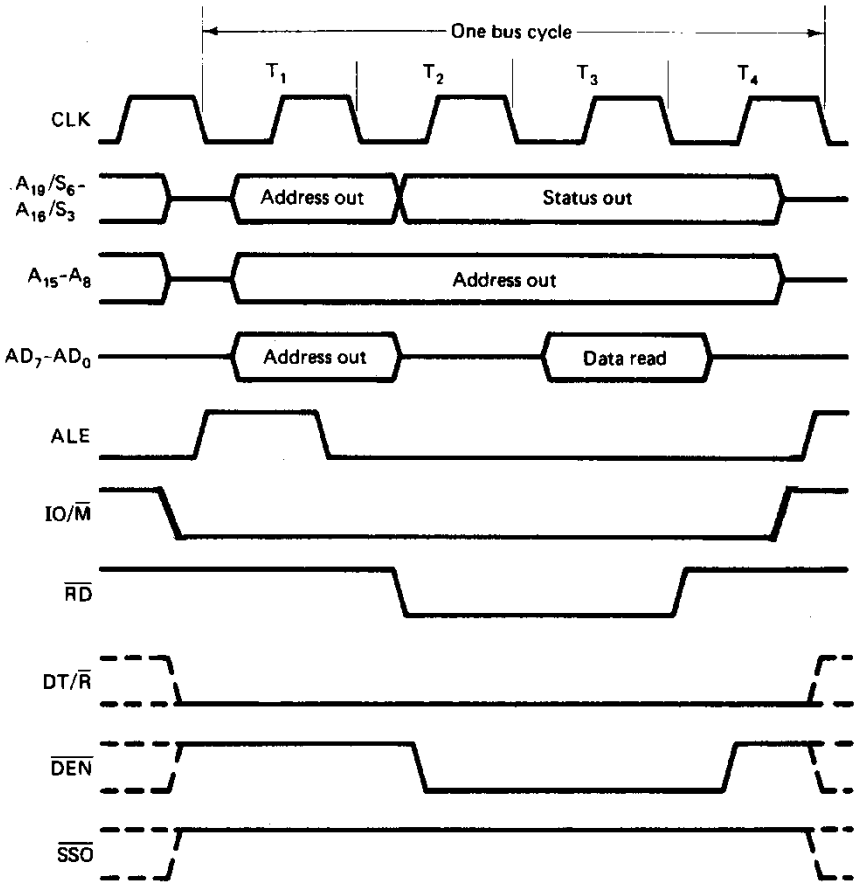
- The control signals for the 8086's minimum mode memory interface differs in three ways:
  - IO/M signal is replaced by M/IO signal.
  - The signal SSO is removed from the interface.
  - BHE (bank high enable) is added to the interface and is used to select input for the high bank of memory in the 8086's memory subsystem.

8.10 Memory Control Signals

- Maximum-mode memory control signals
  - **MRDC – Memory Read Command**
  - **MWTC – Memory Write Command**
  - **AMWC – Advanced Memory Write Command**

Status Inputs			CPU Cycle	8288 Command
$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$		
0	0	0	Interrupt acknowledge	$\overline{INTA}$
0	0	1	Read I/O port	$\overline{IORC}$
0	1	0	Write I/O port	$\overline{IOWC}$, $\overline{AIOWC}$
0	1	1	Halt	None
1	0	0	Instruction fetch	$\overline{MRDC}$
1	0	1	Read memory	$\overline{MRDC}$
1	1	0	Write memory	$\overline{MWTC}$, $\overline{AMWC}$
1	1	1	Passive	None

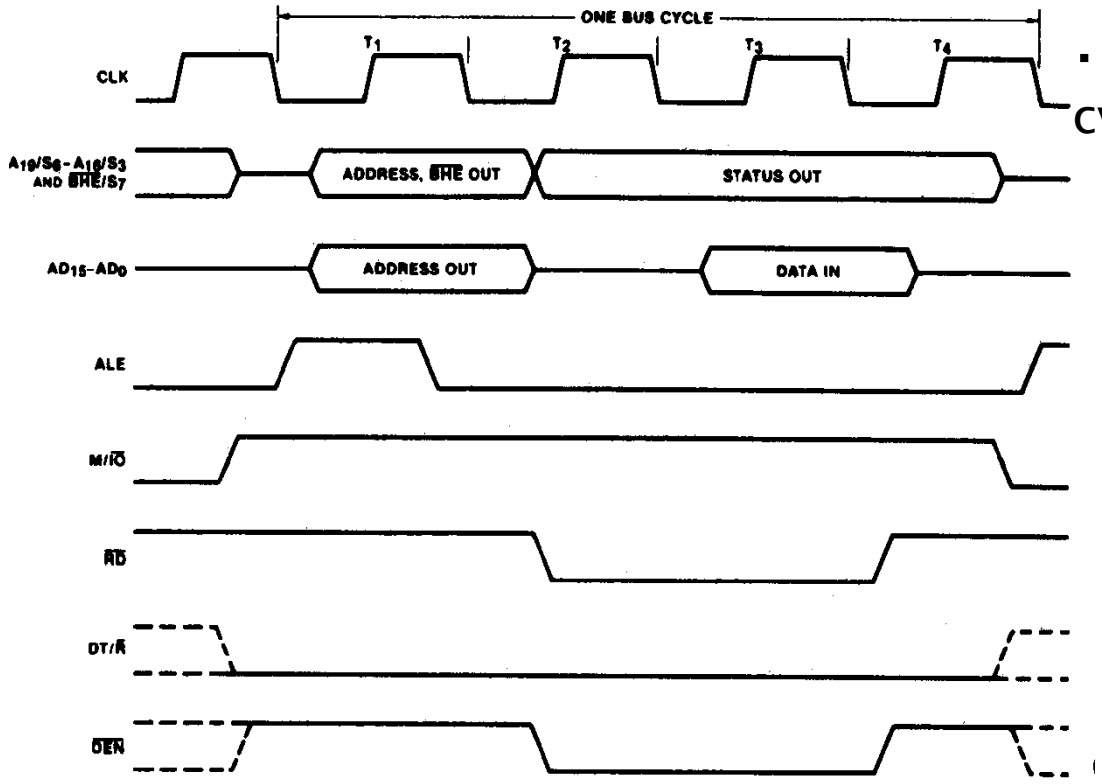
8.11 Read and Write Bus Cycle



Minimum-mode memory read bus cycle of the 8088

- Read bus cycle timing diagram—shows relationship between signals relative to times states
- T1 state—read cycle begins
  - Address output on A0–A19
  - Pulse produced at ALE--address should be latched in external circuitry on trailing edge of ALE
  - IO/M* set to 0 → memory bus cycle
  - DT/R* set to 0 → set external data bus control circuitry for receive mode (read)
- T2 state
  - Status code output on S3–S6
  - AD0 through AD7 tri-stated in preparation for data bus operation
  - RD* set to 0 → read cycle
  - DEN* set to 0 → enable external data bus control circuitry
- T3 state
  - Data on D0–D7 read by the MPU
- T4 state—read cycle finishes
  - RD* returns to 1→ inactive level
  - Complete address/data bus tristate
  - IO/M* returned to 1 → IO bus cycle
  - DEN* returned to 1→ inactive level
  - DT/R* returns to 1→ transmit level

8.11 Read and Write Bus Cycle

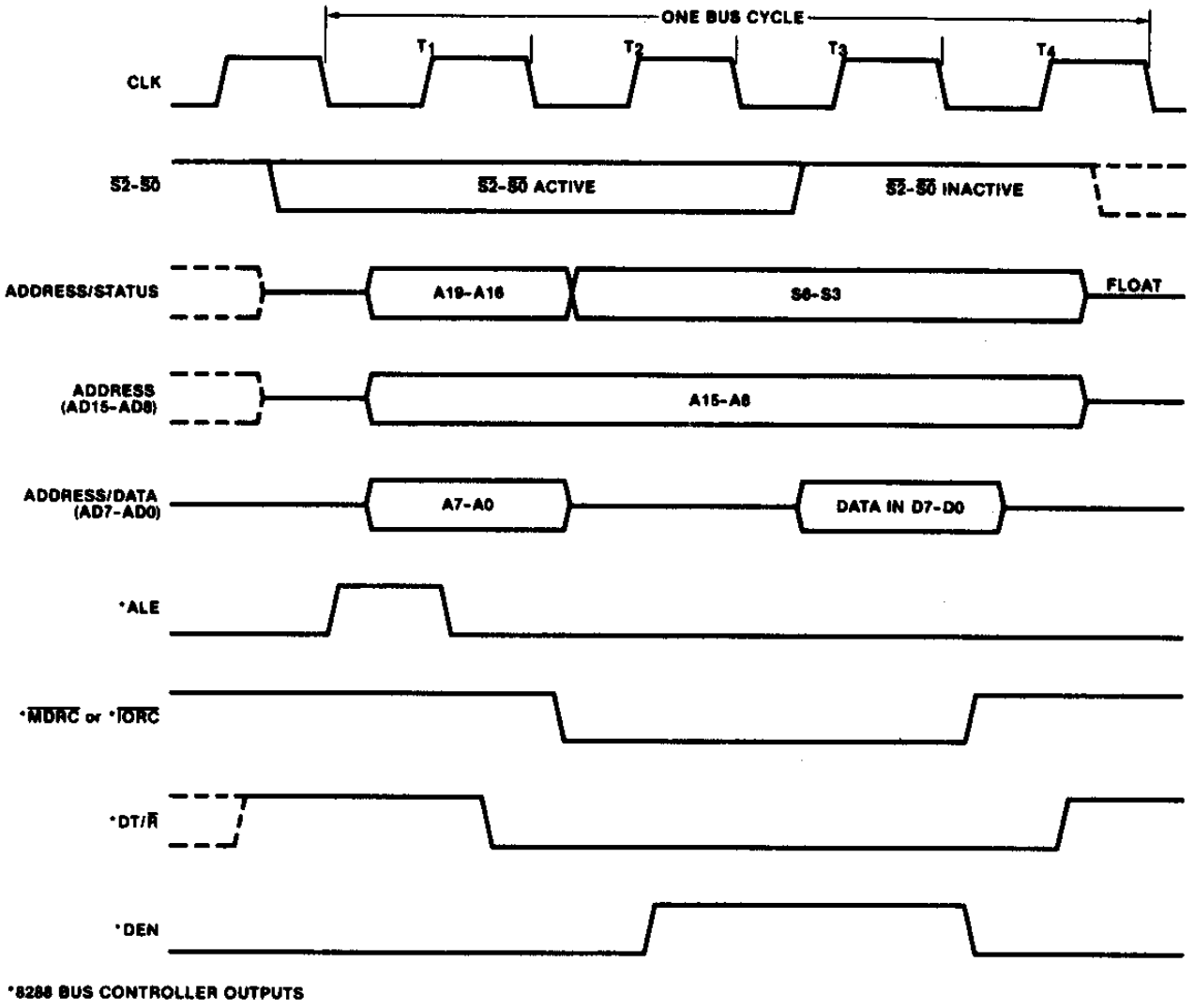


Minimum-mode memory read bus cycle of the 8086

- Differences of 8086 read bus cycle
  - BHE* is output along with the address in T1
  - Data read by the MPU can be carried over all 16 data bus lines
  - M/IO*—which replaces IO/M*—switches to 1 instead of 0 at the beginning of T1
  - SSO* signals is not produce

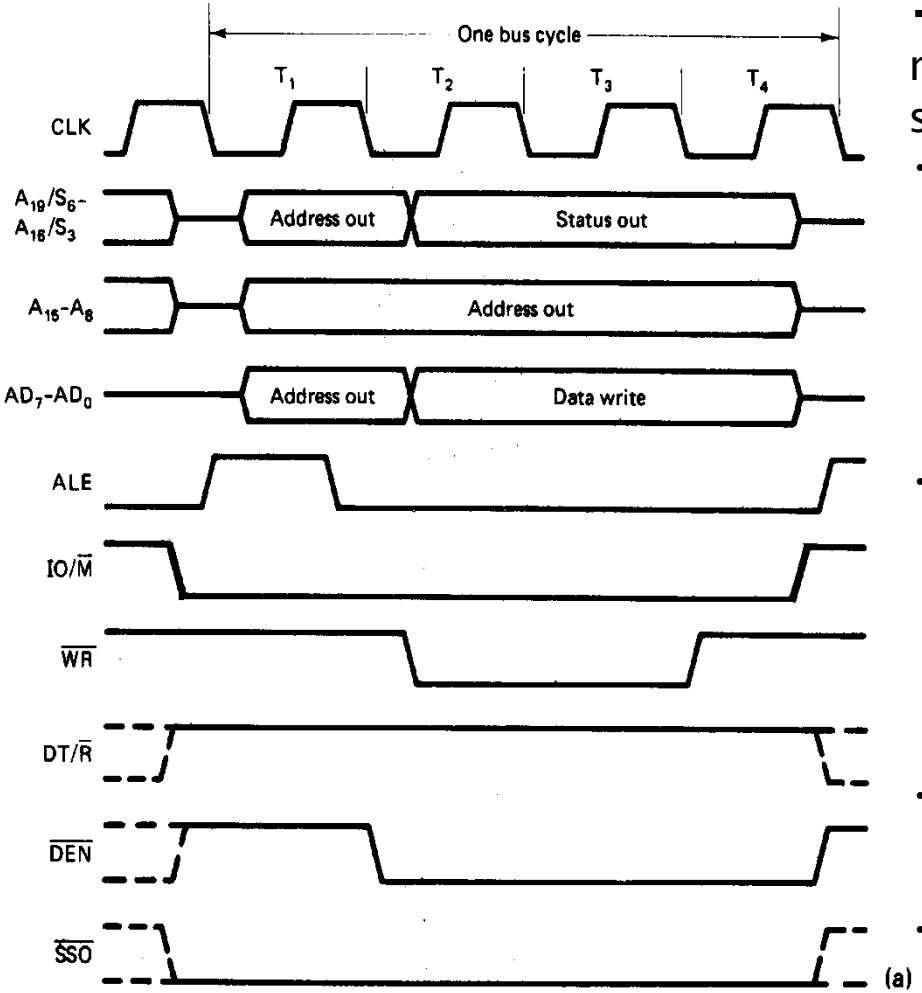
8.11 Read and Write Bus Cycle

- Read cycle



Maximum-mode memory read bus cycle of the 8086

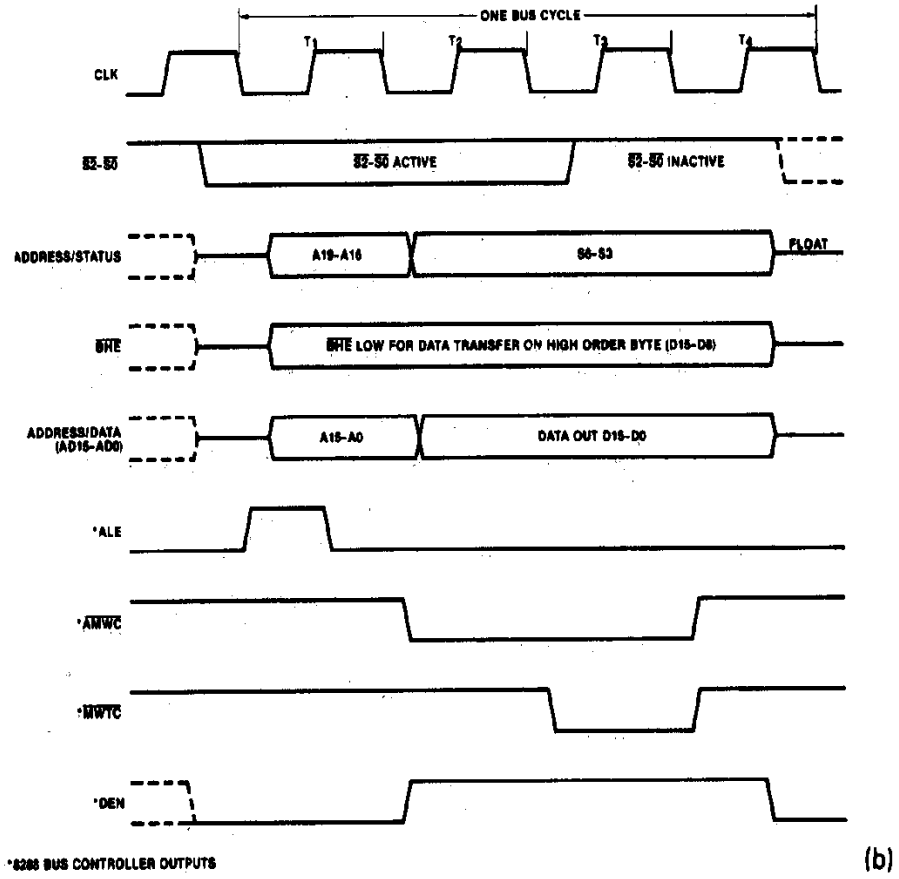
8.11 Read and Write Bus Cycle



Minimum-mode memory write bus cycle of the 8088

- Write bus cycle timing diagram—shows relationship between signals relative to times states
- T1 state—write cycle begins
  - Address output on A0–A19
  - Pulse produced at ALE and address latched in external circuitry on trailing edge of ALE
  - IO/M* set to 0 → memory bus cycle
  - DT/R* remains at 1→ external data bus control circuitry for transmit mode (write)
- T2 state
  - Status code output on S3–S6
  - AD0 through AD7 transitioned to data bus and write data placed on bus
  - DEN* set to 0 → enable external data bus control circuitry
  - WR* set to 0 → write cycle
- T3 or T4 state
  - Data on D0–D7 written into memory (memory decides when!)
- T4 state—write cycle finishes
  - WR* returns to 1→ inactive level
  - Complete address/data bus tri-stated
  - IO/M* returned to 1 → IO bus cycle
  - DEN* returned to 1→ inactive level

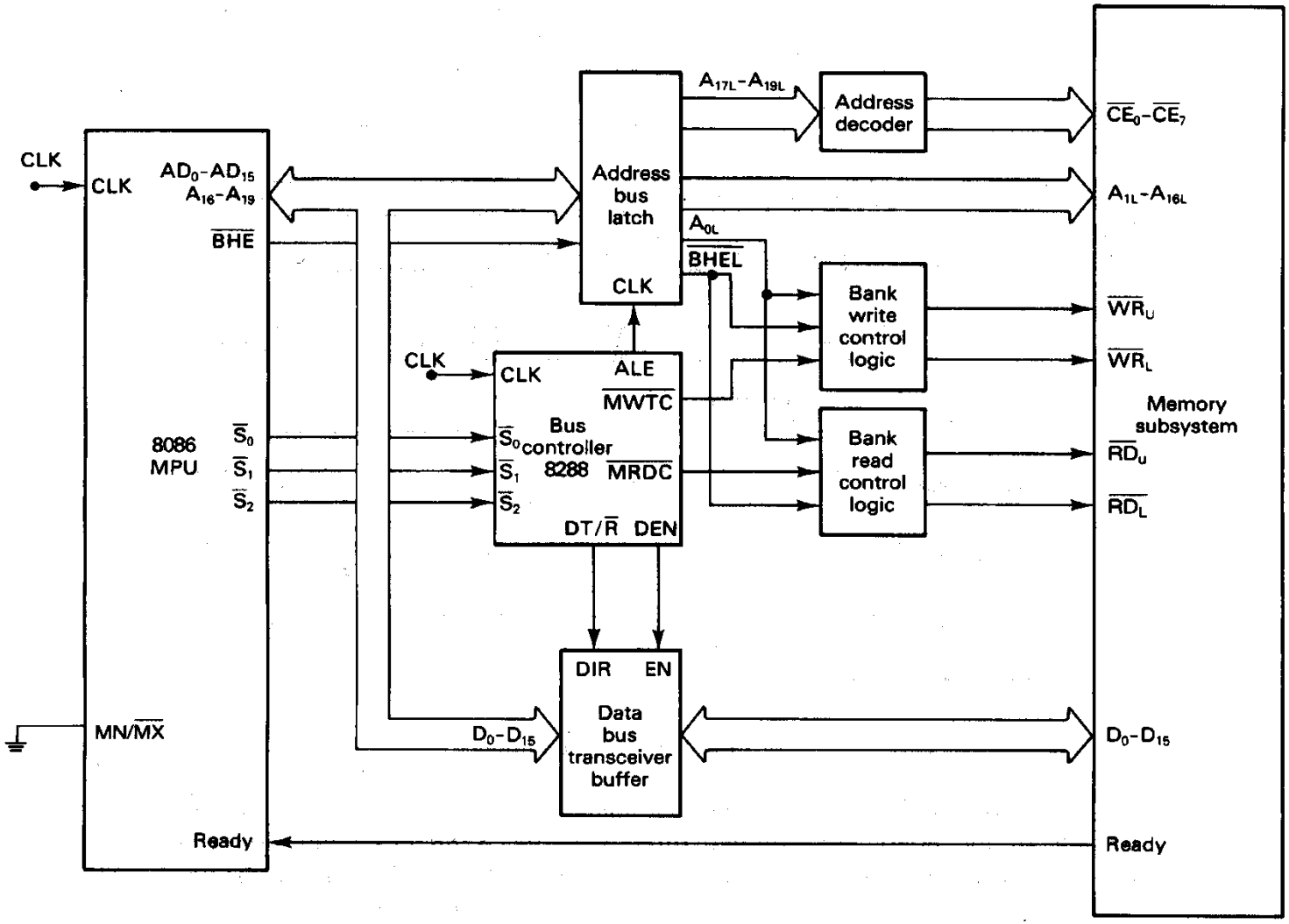
8.11 Read and Write Bus Cycle



Maximum-mode memory write bus cycle of the 8088

- Similar to 8088/8086 minimum-mode write bus cycle
  - Address and data transfer operation identical
  - Transfer may be a high-byte, low-byte, word
  - Differences is the 8288 produces the bus control signals—ALE, DEN, AMWC*, and MWTC*
  - Bus status code S2*-S0* output prior to T1 and held through T2
  - AMWC* and MWTC* replace WR* (Note timing difference)
  - DEN =1 produced instead of DEN* =0 (change in external circuitry!)

8.12 Memory Interface Circuit

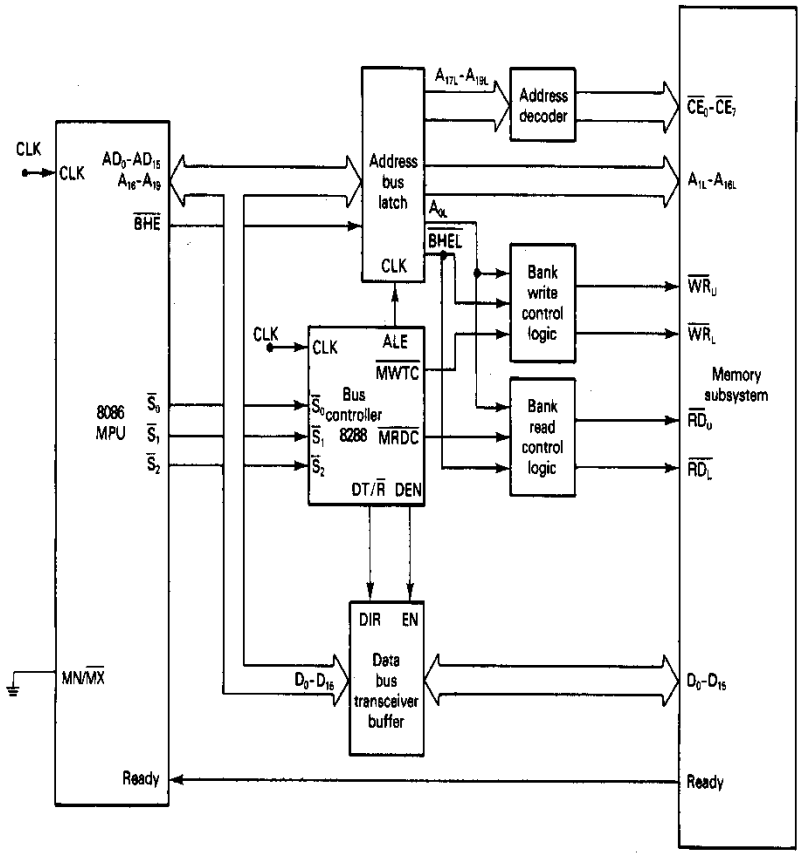


Memory interface block diagram

8.12 Memory Interface Circuit

- Address bus latches and buffers
- Bank write and bank read control logic
- Data bus transceivers/buffers
- Address decoders

8.12 Memory Interface Circuit

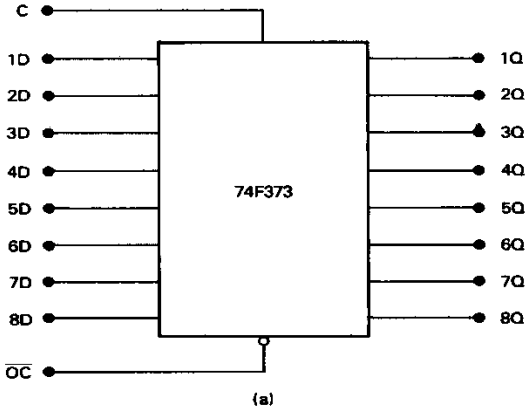


Memory interface block diagram

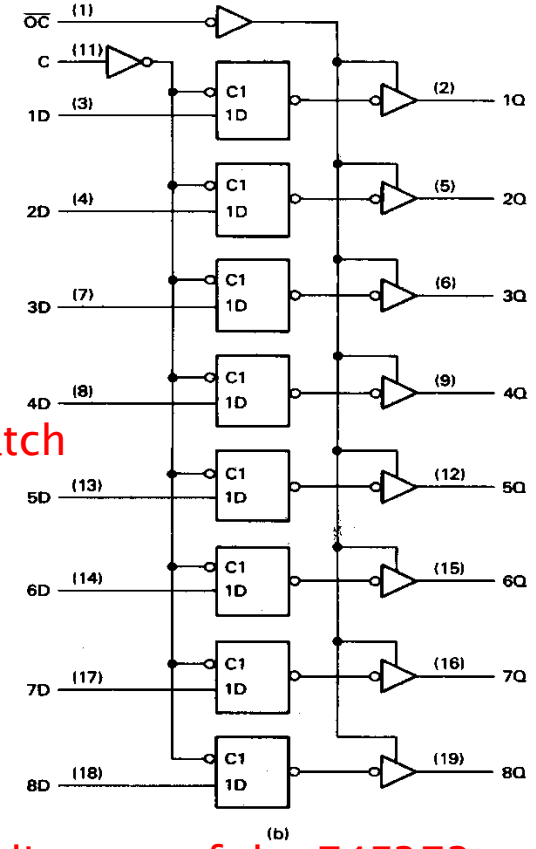
- Overview of the memory access
  - Bus status code for type of memory access output to 8288 on S_2*–S_0*
  - 8288 decodes to produce the command and control signals need to coordinate the data transfer
  - Address is latched, buffered, and decoded to:
    - Produce chip enable signals for the memory array
    - Select a specific memory location
    - Select upper, lower, or both banks of the memory array
  - MWTC* and MRDC* combined with A_{0L} and BHEL* to set the appropriate bank(s) of the memory array for write or read, respectively
  - DT/R* and DEN are used to enable the data bus transceiver/buffer and set it for the transmit (write) or receive (read) direction

8.12 Memory Interface Circuit

- **Roles of the address latch**
  - Latch address signals A_0–A_{19} and BHE*
  - Buffer signals so that they may be used to drive a large memory system, IO peripherals, and other interfaces
- Requirements
  - 21 bit wide latch/buffer
  - Low propagation delay– allows use of slower memories
- 74F373—Octal D-type latch
  - 8 independent buffered D-type flipflops (latches)
  - Enable to output propagation delay = 13 ns
  - Outputs sink 24 mA (buffering)
  - Output Control (OC*) =0 enables latch circuits



Block diagram of a D-type latch



Circuit diagram of the 74F373

Operation of the 74F373

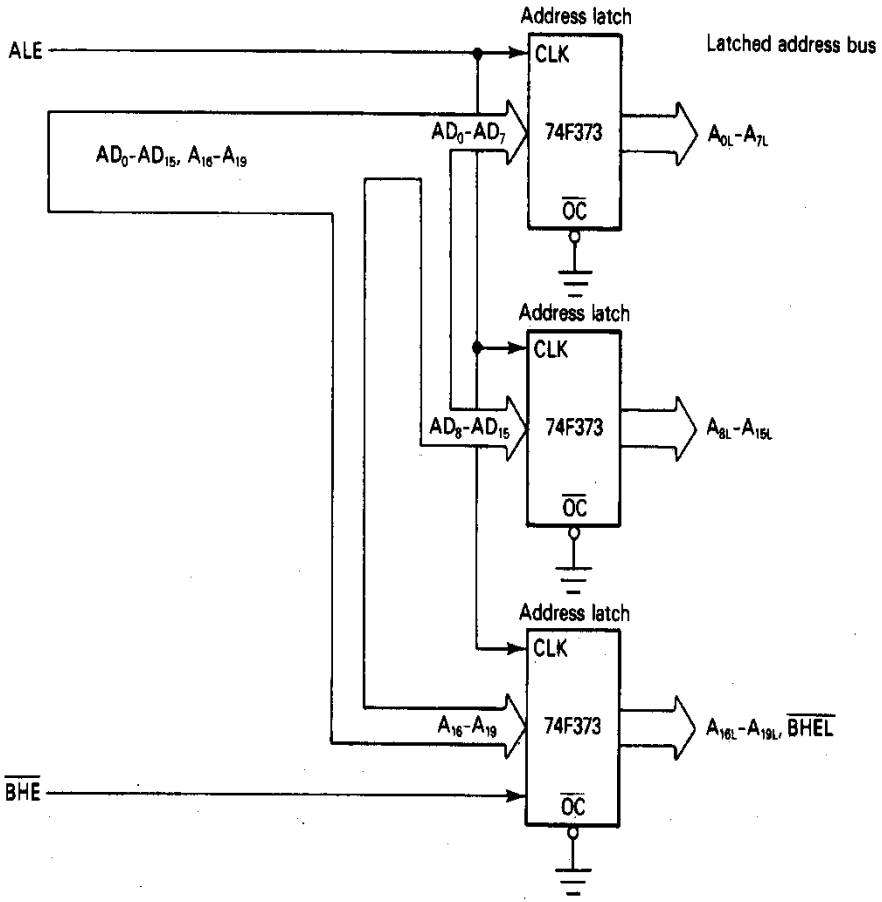
Inputs			Output
$\overline{OC}$	Enable C	D	Q
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

(c)

Address bus latches and buffers

8.12 Memory Interface Circuit

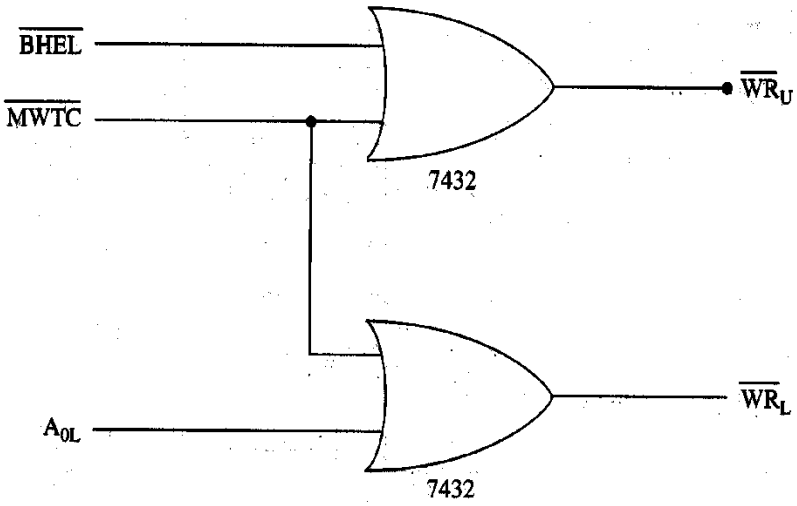
- Address bus latches and buffers



Address latch circuit

- Implemented with 3 74F373 Octal-D-type latches
  - Inputs AD_0–AD_{15}, A_{16}–A_{19}, and BHE* from MPU
  - All devices permanently enabled by fixing the OC* inputs at logic 0
  - All latches clocked in parallel with pulse at ALE from 8288
  - Latched and buffered outputs are: A_{0L}–A_{19L}, and BHEL*
  - Parts of address applied to the address inputs of memory subsystem, address decoder, and read/write control logic

8.12 Memory Interface Circuit

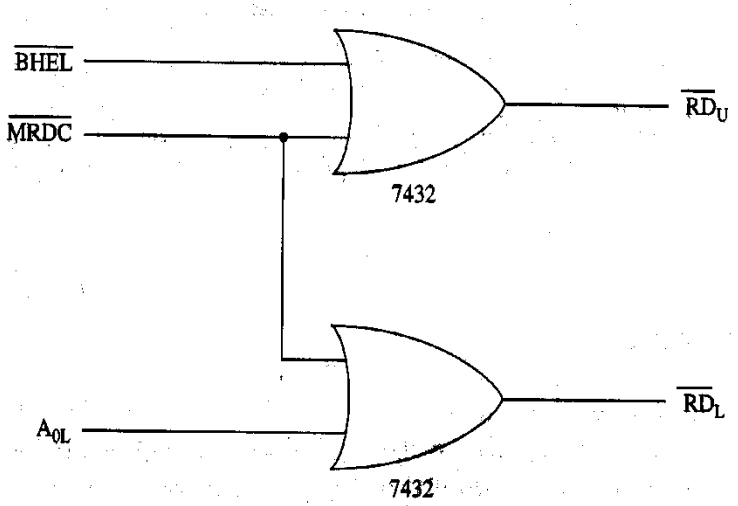


Bank write control logic

- **Role of write control logic**
- Memory array is organized in upper and lower banks
- Types of data writes that may take place:
- Byte to a storage location in the upper (odd) bank
- Byte to a storage location in the lower (even) bank
- Word to storage locations in both banks
- Write control logic must decode A_{0L}, BHEL*, and MWTC* to produce independent write signals—WRU* and WRL*
- 7432—2-input OR gate solution
- MWTC*(MWRC*) =0 enables both gates

BHEL*	A_{0L}	WRU*	WRL*	Result
0	0	0	0	Both banks enabled
1	0	1	0	Lower (even) bank enabled
0	1	0	1	Upper (odd) bank enabled

8.12 Memory Interface Circuit



Bank read control logic

- **Role of read control logic**
  - Types of data reads that may take place:
    - Byte from the lower (even) bank
    - Byte from upper (odd) bank
    - Word of data from both banks or an instruction fetch
  - Read control logic must decode A_{0L}, BHEL*, and MRDC* to produce independent read signals—RDU* and RDL*
  - 7432-based solution is similar to bank write control logic
  - MRDC* =0 enables both gates

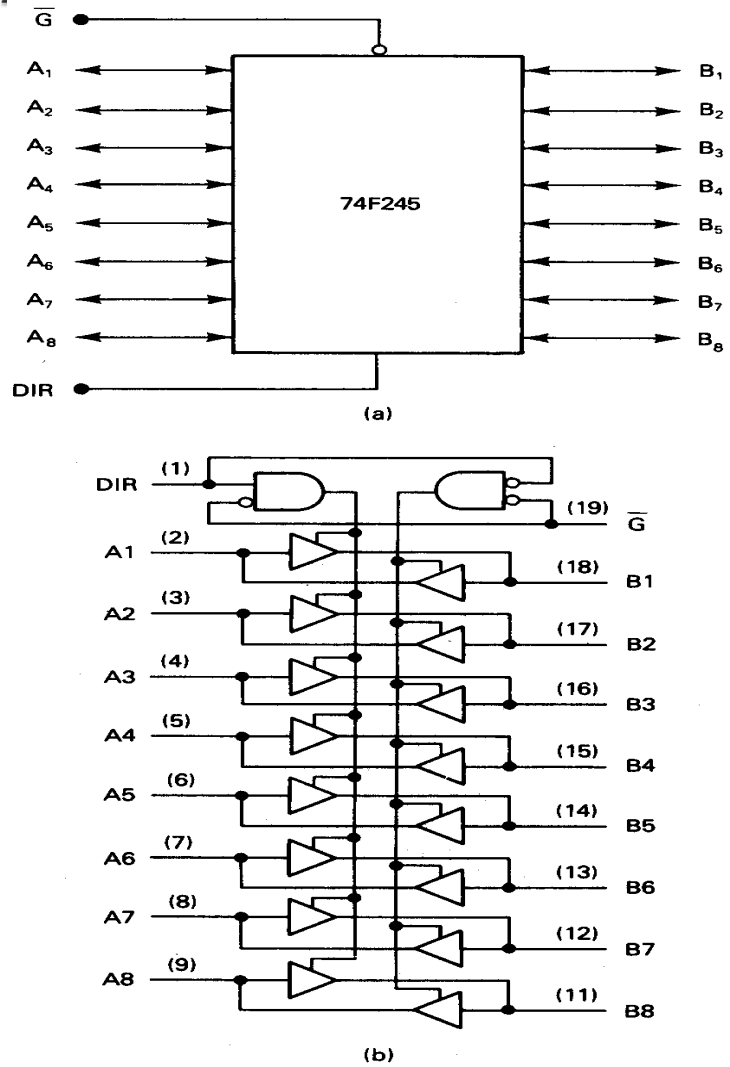
BHEL*	A_{0L}	RDU*	RDL*	Result
0	0	0	0	Both banks enabled
1	0	1	0	Lower (even) bank enabled
0	1	0	1	Upper (odd) bank enabled

8.12 Memory Interface Circuit

- **Roles of the data bus transceiver/buffer**
  - Set the direction of the data path been memory and the MPU bus
  - Appropriately time the enabling of the transceivers to coincide with the read/write data transfer
  - Buffer the data bus lines so that they may be used to drive a large memory system, IO peripherals, and other interfaces
- Requirements
  - 8/16 bit wide bi-directional transceiver
  - Low propagation delay- allows use of slower memories
- 74F245—Octal bi-directional bus transceiver
  - 8 independent bus transceivers
  - DIR input selects direction of data transfer

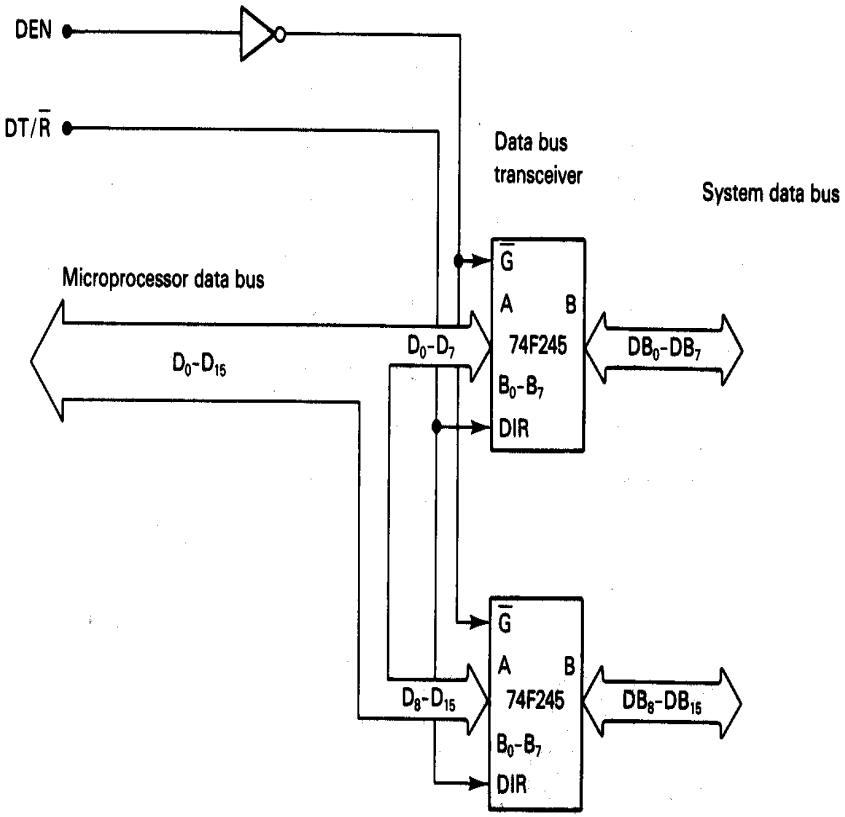
  0 = B to A → read

  1 = A to B → write
  - G* =0 enables all transceivers
  - Outputs sink 64 mA (buffering)



Block diagram and circuit diagram of the 74F245 octal bus transceiver

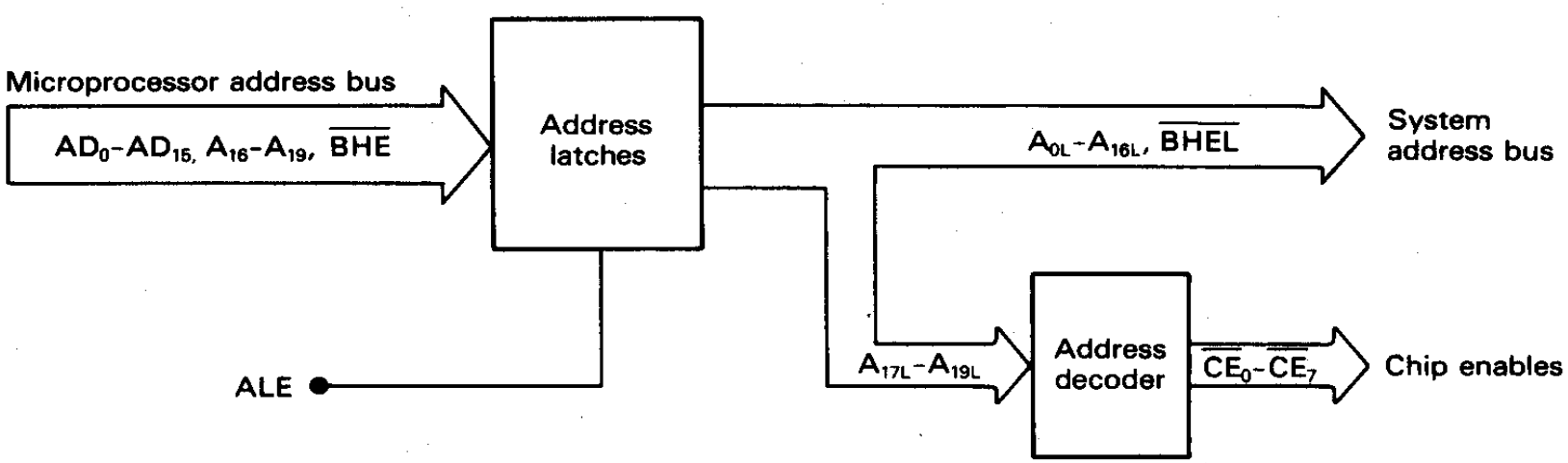
8.12 Memory Interface Circuit



Data bus transceiver circuit

- Implemented with 2 74F245 Octal bi-directional bus transceivers
  - A inputs/outputs are D_0–D_{15} directly from MPU
  - Direction of both devices set by logic level of DT/R*
  - Both devices enabled at appropriate time for data transfer by DEN=1
  - B inputs/outputs are the buffered data bus lines DB_0 through DB_{15}
  - Buffered data bus lines applied directly to the memory subsystem

8.12 Memory Interface Circuit

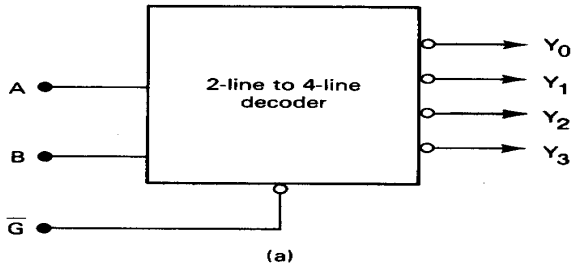


Address bus configuration with address decoding

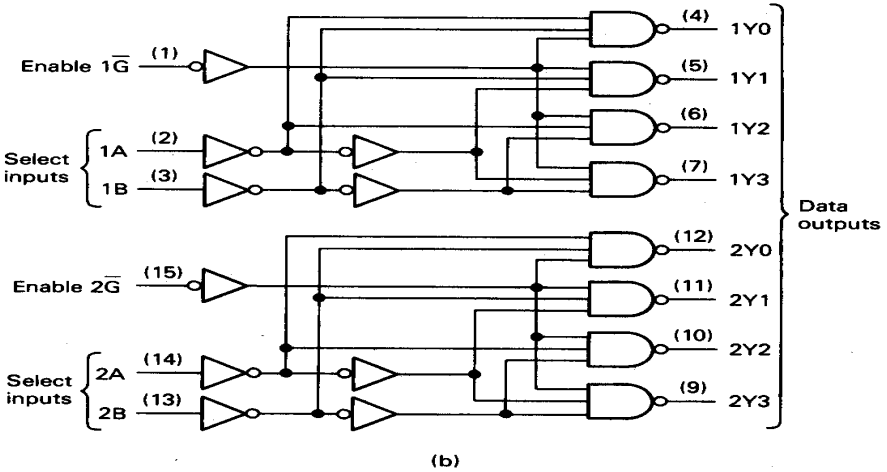
- **Role of address decoder**
  - Part of the buffered/latched address is decoded to create chip enables
  - Address inputs A_{19L} A_{18L} A_{17L} is decoded to produce 8 independent chip enable outputs CE_0* through CE_7*

8.12 Memory Interface Circuit

- Requirements of Address Decoder
  - Requires standard decoder/multiplexer functions
  - 2-input 4-output—74F139
  - 3-input 8-output—74F138
  - 74F139 decoder/multiplexer
  - Dual 2-input 4-output decoder/multiplexer
    - Two-bit input BA
    - Four independent outputs Y0 through Y3
    - G* input enables the associated decoder
  - Operation
  - G* = 1 forces all outputs to 1
  - G* = 0 enables circuit and the Y output associated with the input code switches to the active 0 level



(a)



(b)

INPUTS			OUTPUTS			
ENABLE $\overline{G}$	SELECT B	SELECT A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

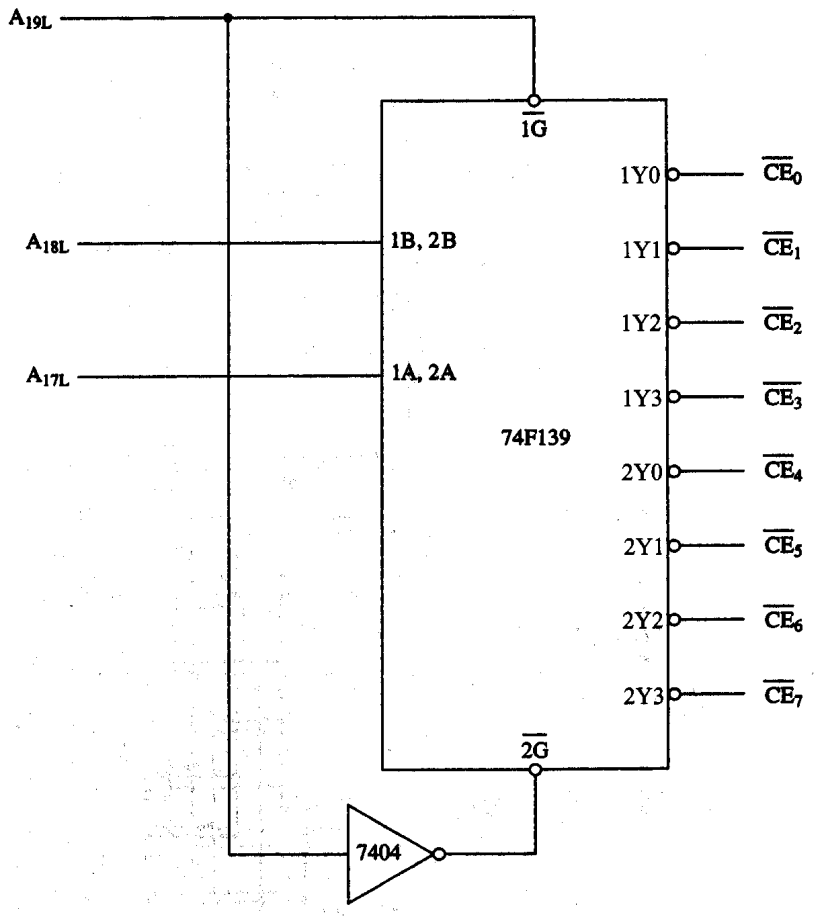
(c)

Block diagram, operation and circuit diagram of the 74F139 decoder

8.12 Memory Interface Circuit

- Address decoder

- 3-input 8-output address decoder made with two 2-input 4-output decoder/multiplexer circuits
  - A_{19L} =0 makes 1G* =0 and 2G* =1 and enables outputs CE_0*- CE_3*
  - A_{19L} =1 makes 1G* =1 and 2G* =0 and enables CE_4*-CE_7*
  - Code at A_{18L} A_{17L} applied to the BA inputs of both multiplexers in parallel
  - Output associated with the input code on the enabled decoder becomes active

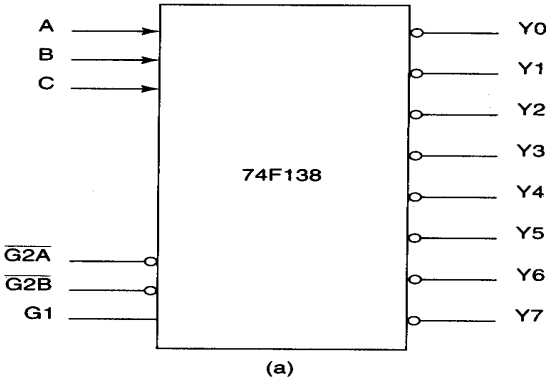


Address decoder circuit using 74F139

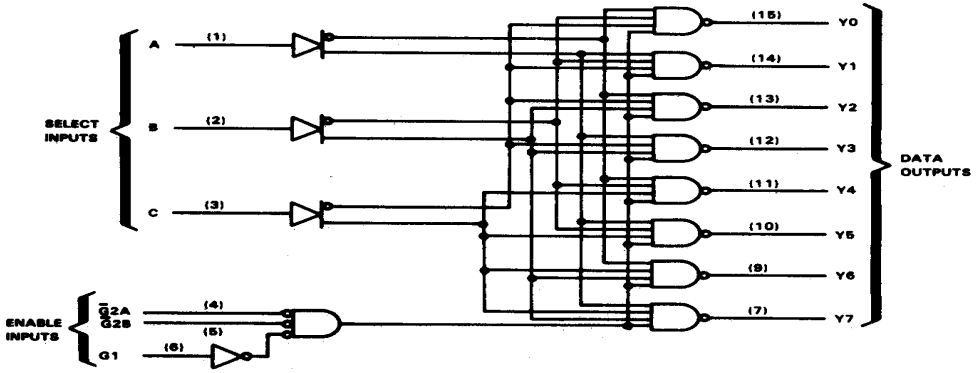
8.12 Memory Interface Circuit

- Address decoder

Block diagram, operation and circuit diagram of the 74F138 decoder



(a)



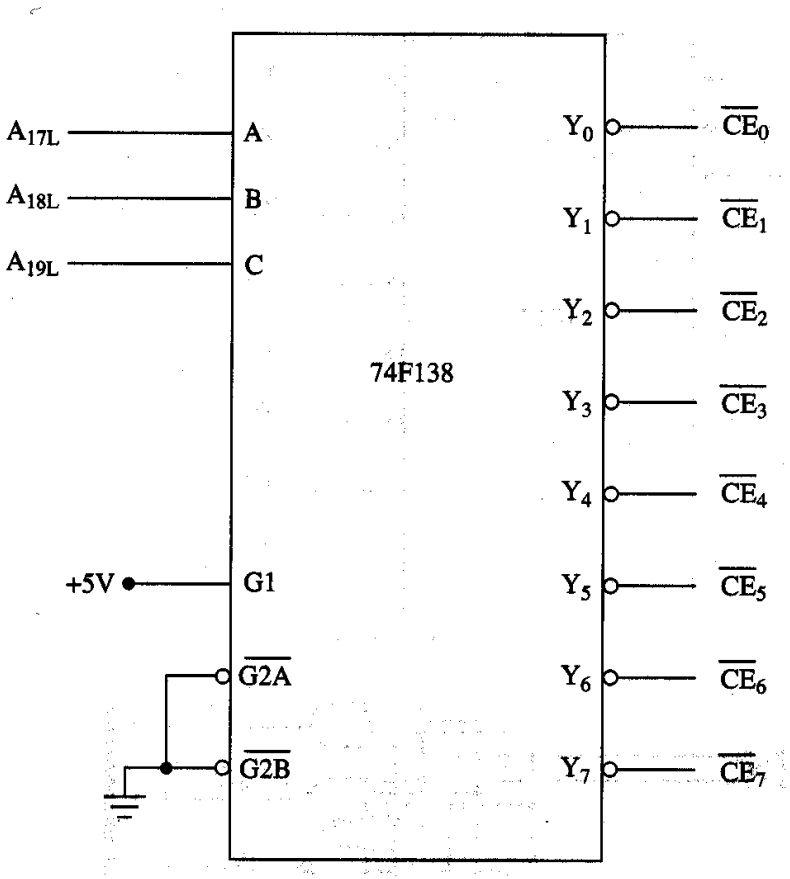
(b)

ENABLE INPUTS			SELECT INPUTS			OUTPUTS							
G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

(c)

8.12 Memory Interface Circuit

- Address decoder



Address decoder circuit using 74F138

8.13 Programmable Logic Arrays

- **Programmable logic array, PLA,** are general purpose logic devices that have the ability to perform a wide variety of specialized logic functions.
- A PLA contains a general-purpose AND-OR-NOT array of logic gate circuits.
- The process used to connect or disconnect inputs of the AND gate array is known as programming, which leads to the name programmable logic array.

8.13 Programmable Logic Arrays

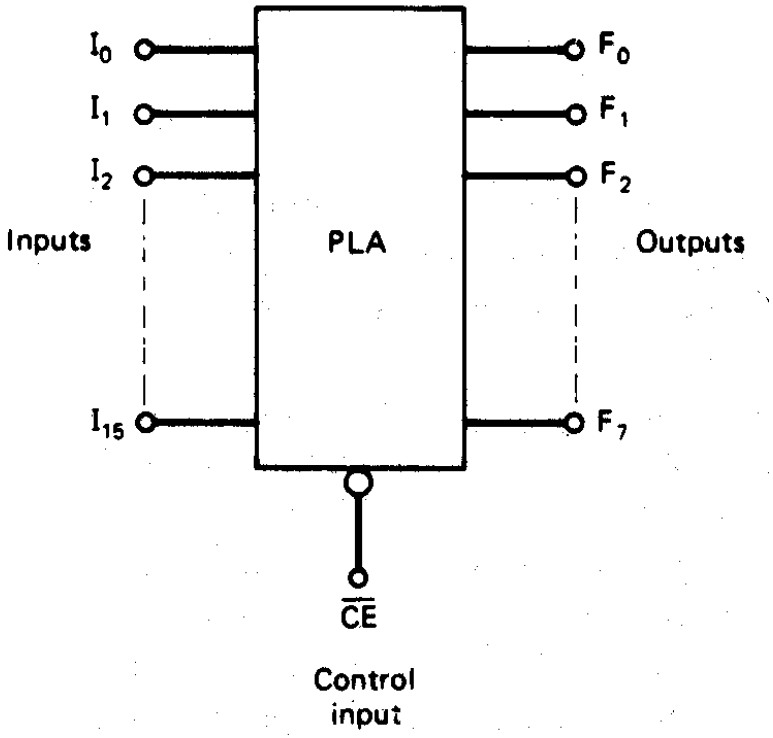
- Major types of programmable logic architecture
  - Simple Programmable Logic Devices (**SPLDs**)
    - PAL, GAL, PLA, EPLD
  - Complex Programmable Logic Devices (**CPLDs**)
    - EPLD, PEEL, EEPLD, MAX
  - Field Programmable Gate Arrays (**FPGAs**)
    - LCA, pASIC, FLEX, APEX, ACT, ORCA, Virtex,pASIC
  - Field Programmable InterConnect (**FPICs**)

8.13 Programmable Logic Arrays

- PLAs, GALs, and EPLDs
  - Early PLA devices were all manufactured with the bipolar semiconductor process.
  - Bipolar devices are programmed with an interconnect pattern by burning out fuse links within the device.
  - PLAs made with bipolar technology are characterized by slower operating speeds and higher power consumption.
  - Two kinds of newer PLA, manufactured with the CMOS process, are in wide use today: the **GAL** and **EPLD.**

8.13 Programmable Logic Arrays

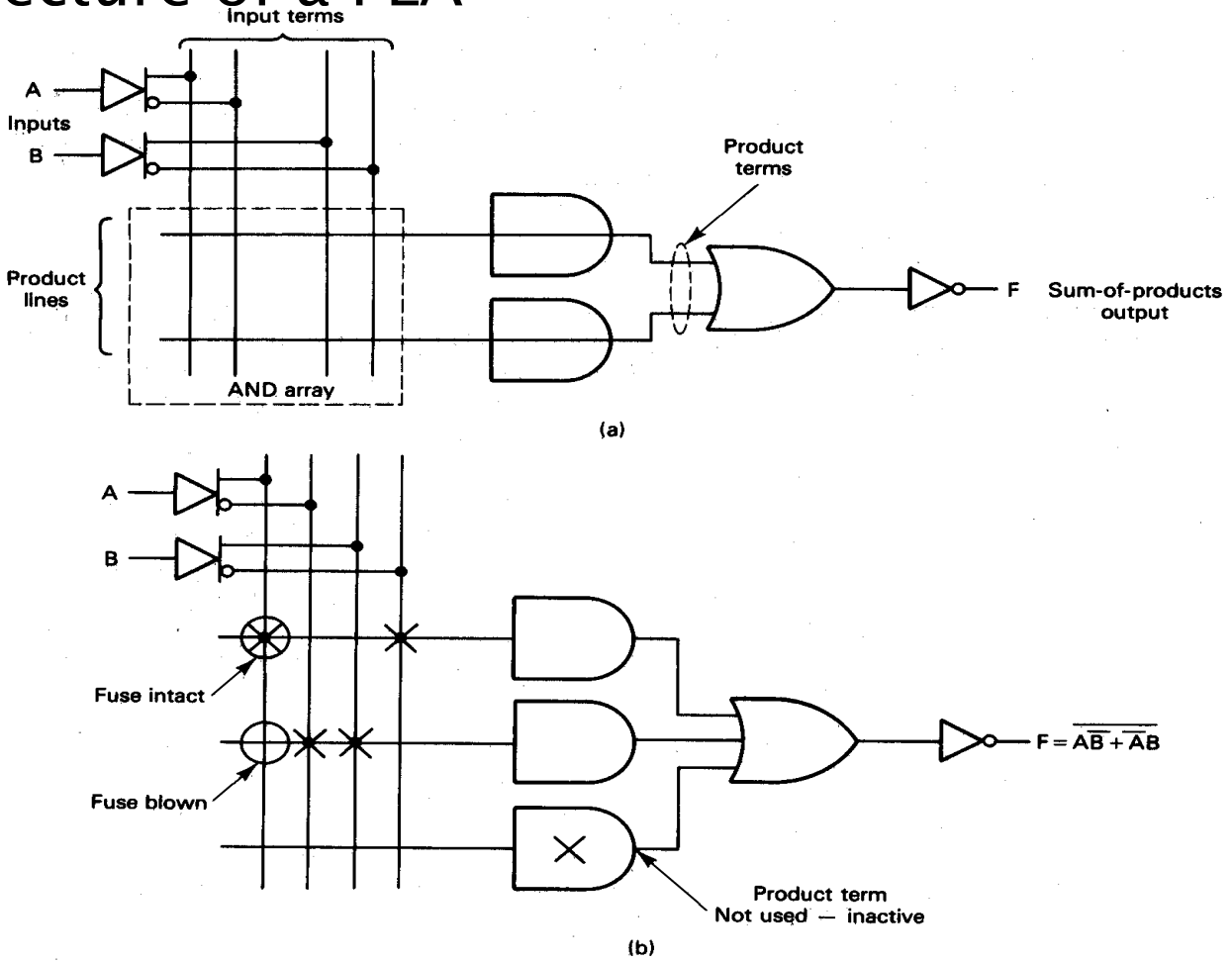
- Block diagram of a PLA
  - The logic levels applied at inputs I_0 through I_{15} and the programming of the AND array determine what logic levels are produced at outputs F_0 through F_{15}.
  - The capacity of a PLA is measured by three properties: the number of inputs, the number of outputs, and the number of product terms (P-terms)



Block diagram of a PLA

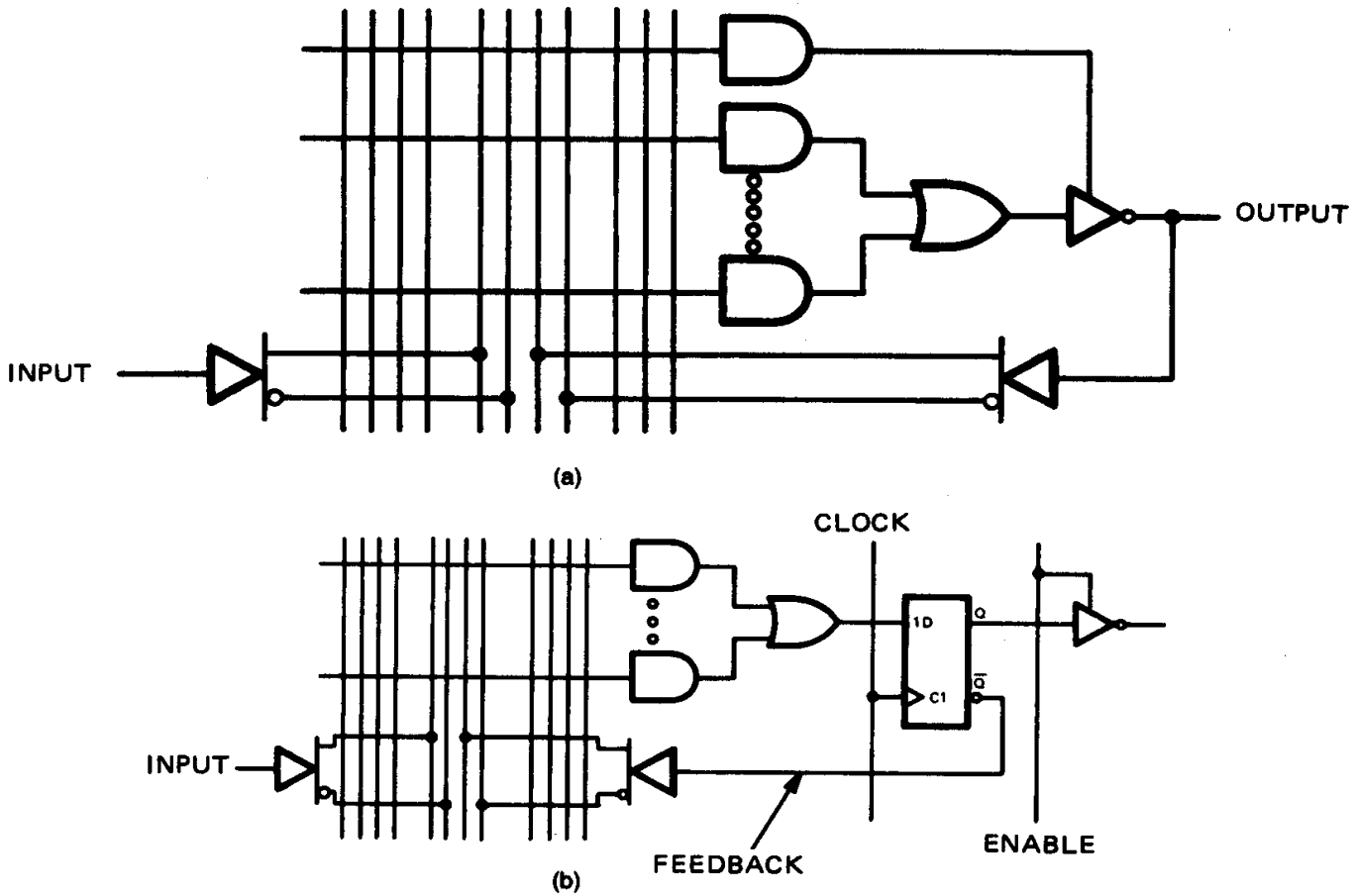
8.13 Programmable Logic Arrays

- Architecture of a PLA



8.13 Programmable Logic Arrays

- Architecture of a PLA



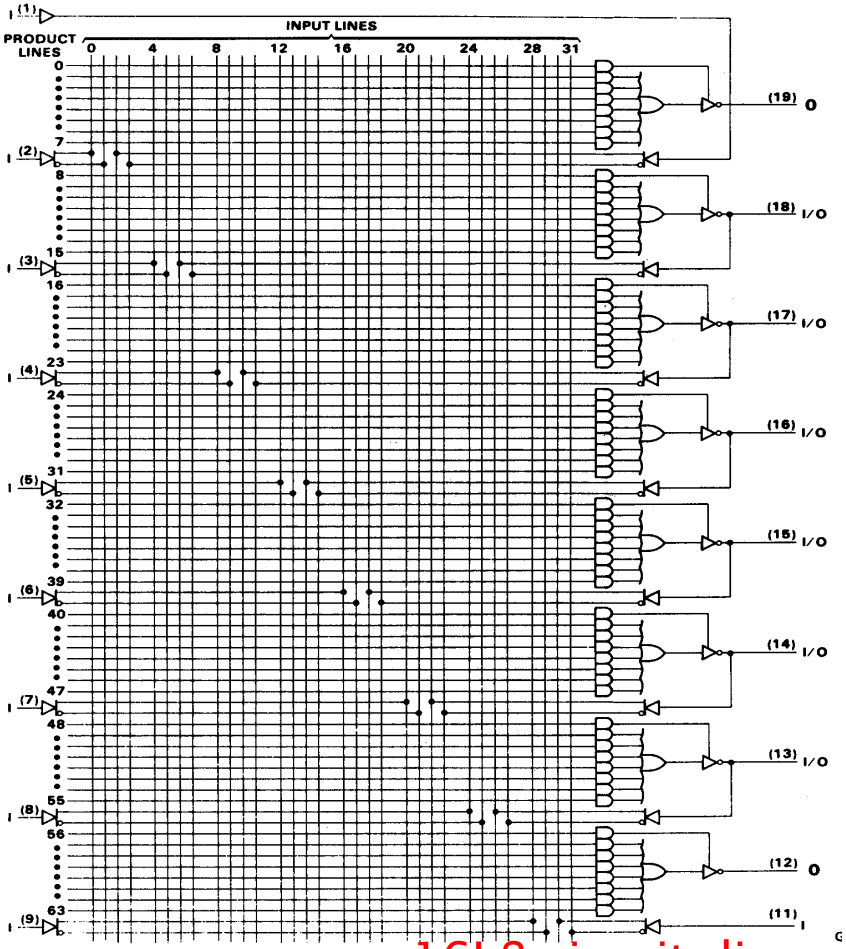
(a) Typical PLA architecture. (b) PLA with output latch

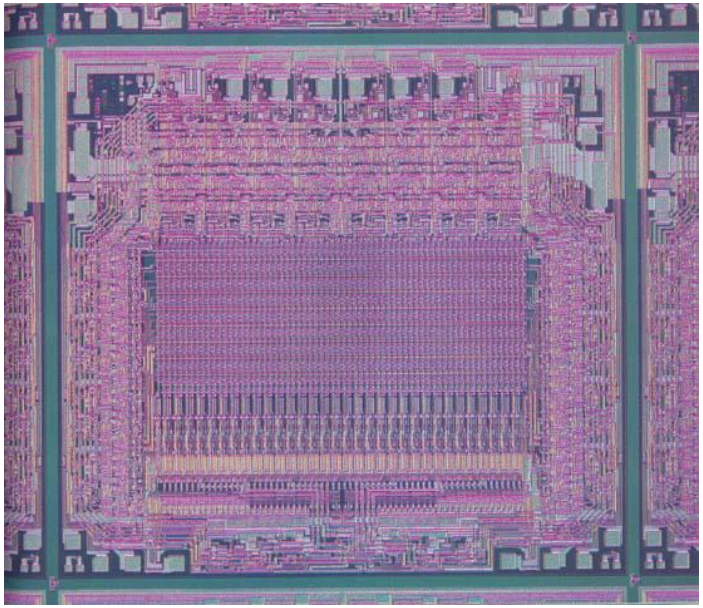
8.13 Programmable Logic Arrays

- Standard PAL™ device
  - A PAL, programmable array logic, is a PLA in which the OR array is fixed; only the AND array is programmable.
  - The 16L8 is a widely used PAL IC. It is housed in a 20-pin package. It has 10 dedicated input, 2 dedicated outputs, and 6 programmable I/O lines.
  - The 16L8 is manufactured with bipolar technology. It operates from a +5V±10% dc power supply and draw a maximum of 180mA.
  - The 20L8 has 20 inputs, 8 outputs and 64 P-terms.
  - The 20R8 is the register output version of 20L8.

8.13 Programmable Logic Arrays

- Standard PAL™ device



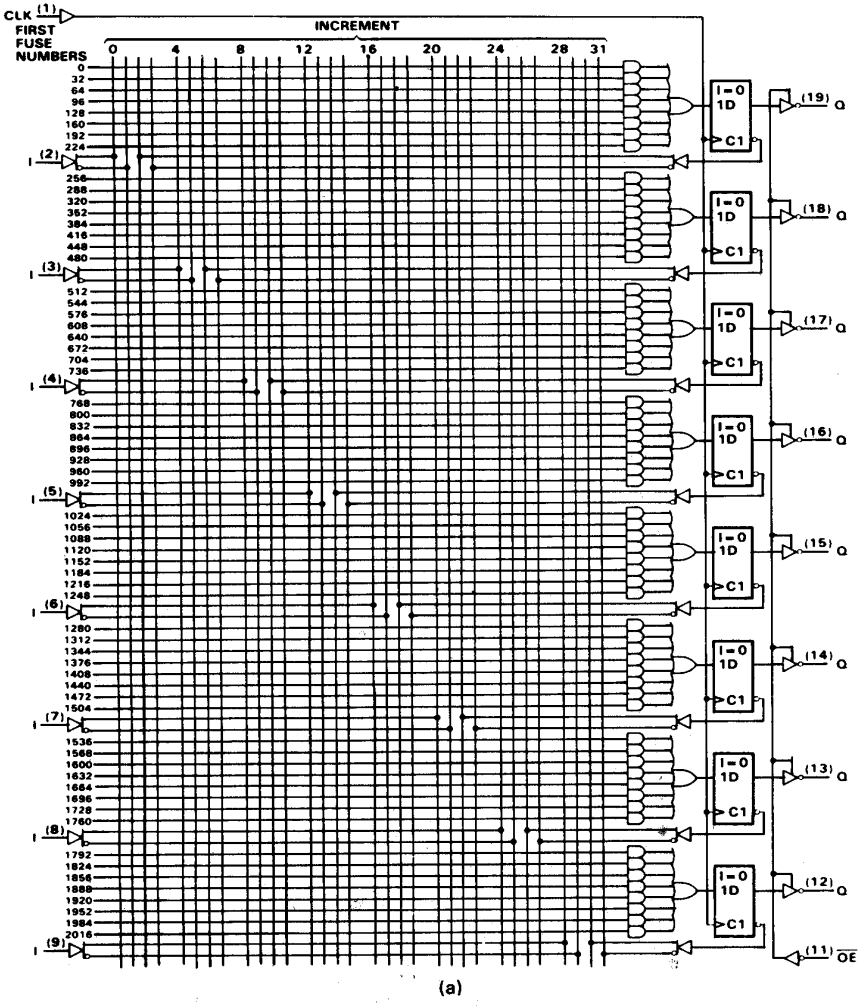


16L8 circuit diagram and pin layout

8.13 Programmable Logic Arrays

- Standard PAL™ device

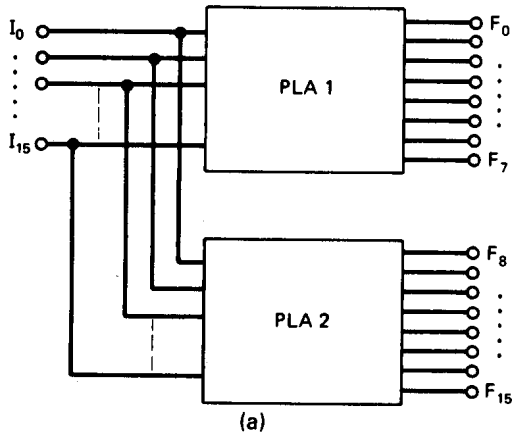
20L8 circuit diagram and pin layout



8.13 Programmable Logic Arrays

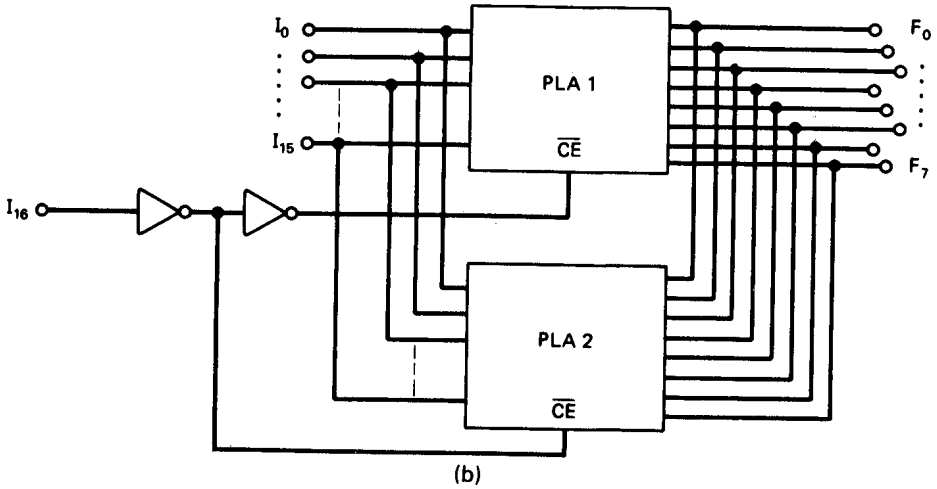
- Expanding PLA capacity

Expanding output word length



(a)

Expanding input word length



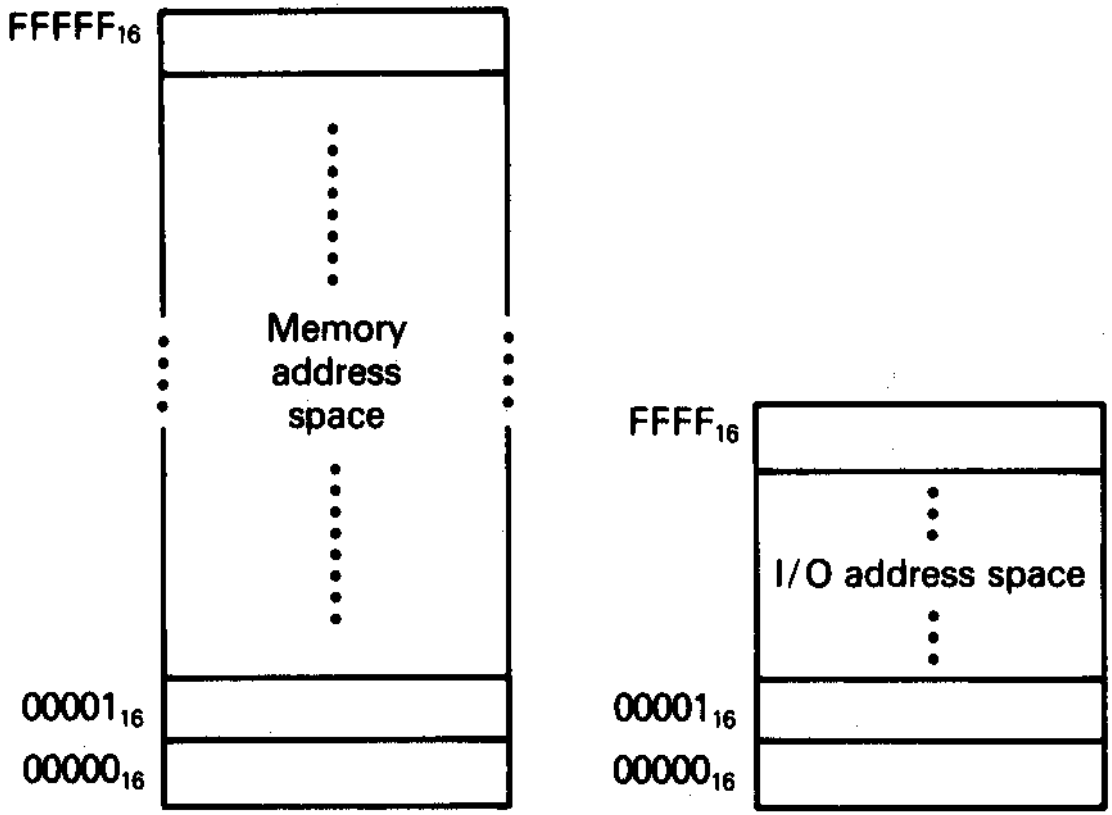
(b)

8.14 Types of Input/Output

- Isolated input/output
  - When using isolated I/O in a microcomputer system, the I/O device are treated separate from memory.
  - The memory address space contains 1 M consecutive byte address in the range 00000H through FFFFFH; and that the I/O address space contains 64K consecutive byte addresses in the range 0000H through FFFFH.
  - All input and output data transfers must take place between the AL or AX register and I/O port.

8.14 Types of Input/Output

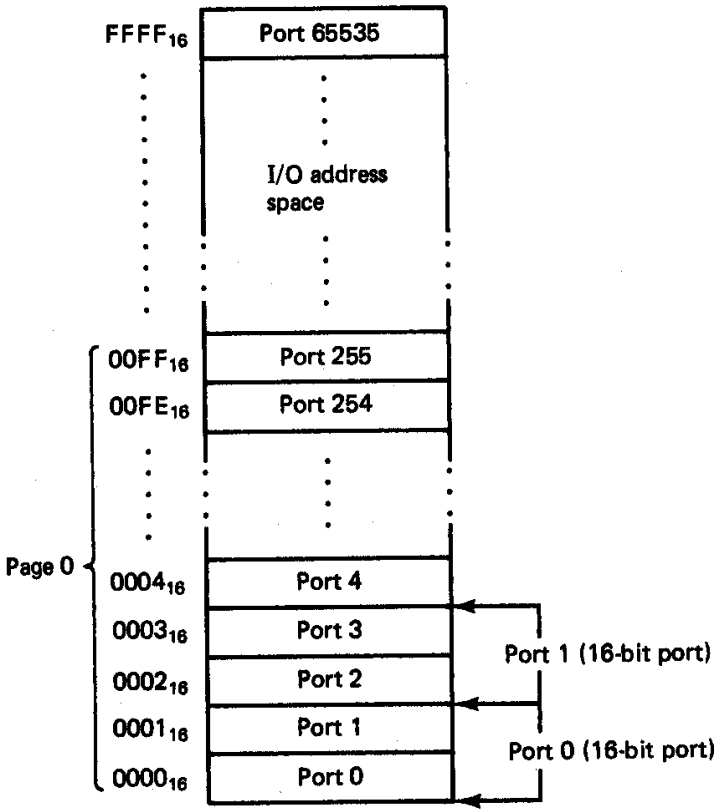
- Isolated input/output



8088/8086 memory and I/O address spaces

8.14 Types of Input/Output

Isolated input/output



Isolated I/O ports

- Input/output data organization
  - Supports byte and word I/O ports
    - 64K independent byte-wide I/O ports
    - 32K independent aligned word-wide I/O ports
    - Word ports may also be misaligned
  - Examples:

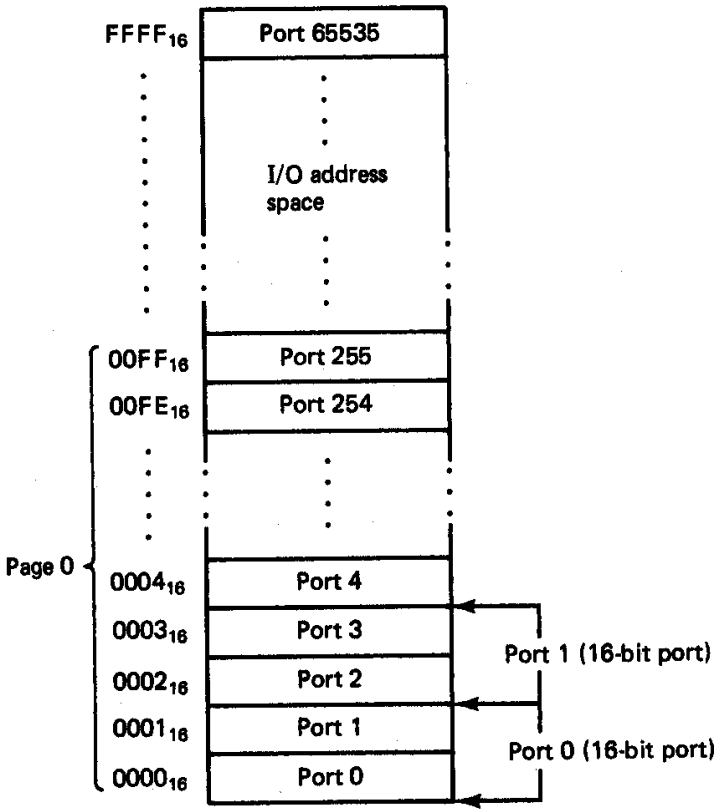
Byte ports 0,1,2 → addresses 0000H, 0001H, and 0002H

Word ports 0,1,2 → addresses 0000H, 0002H, 0004H

  - Advantages of isolated I/O
    - Complete memory address space available for use by memory
    - I/O instructions tailored to maximize performance
  - Disadvantage of Isolated I/O
    - All inputs/outputs must take place between an I/O port and accumulator register (A)

8.14 Types of Input/Output

Isolated input/output



Isolated I/O ports

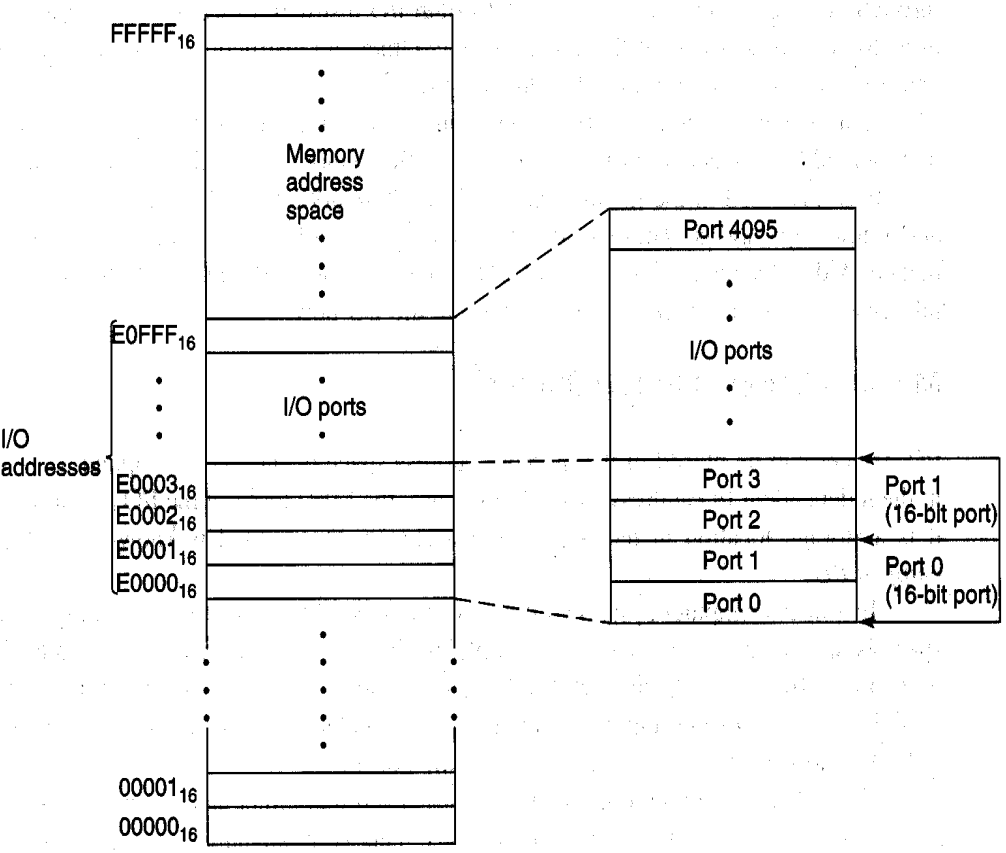
- Input/output data organization
  - All I/O accesses take either one or two bus cycles
    - Byte input/output= 1 bus cycle
    - Aligned word input/output= 1 bus cycle—on 8086
    - Misaligned word input/output = 2 bus cycles
- Page 0
  - First 256 byte addresses → 0000H – 00FFH
  - Can be accessed with direct or variable I/O instructions
  - Ports F8H through FFH reserved

8.14 Types of Input/Output

- Memory-mapped input/output
  - In the case of memory-mapped I/O, MPU looks at the I/O port as though it is a storage location in memory.
  - Some of the memory address space is dedicated to I/O ports.
  - Instructions that affect data in memory are used instead of the special I/O instructions.
  - The memory instructions tend to execute slower than those specifically designed for isolated I/O.

8.14 Types of Input/Output

- Memory-mapped input/output



Memory mapped I/O ports

- Example:
- E0000H-E0FFFH → 4096 memory addresses assigned to I/O ports
- E0000H, E0001H, and E0002H correspond to bytewide ports 0,1, and 2
- E0000H and E0001H correspond to word-wide port 0 at address E0000H
- Advantages of memory mapped I/O
  - Instructions that affect data in memory (MOV, ADD, AND, etc.) can be used to perform I/O operations
  - I/O transfers can take place between and I/O port and any of the registers
- Disadvantage of memory mapped I/O
  - Memory instructions perform slower
  - Part of the memory address space cannot be used to implement memory

8.15 Isolated Input/Output Interface

- I/O devices:
  - Keyboard (input)
  - Printer (output)
  - Mouse (input)
  - 82C55A, etc. (PPI section .13)
- Functions of interface circuit:
  - Select the I/O port
  - Latch output data
  - Sample input data
  - Synchronize data transfer
  - Translate between TTL voltage levels and those required to operate the I/O devices.

8.15 Isolated Input/Output Interface

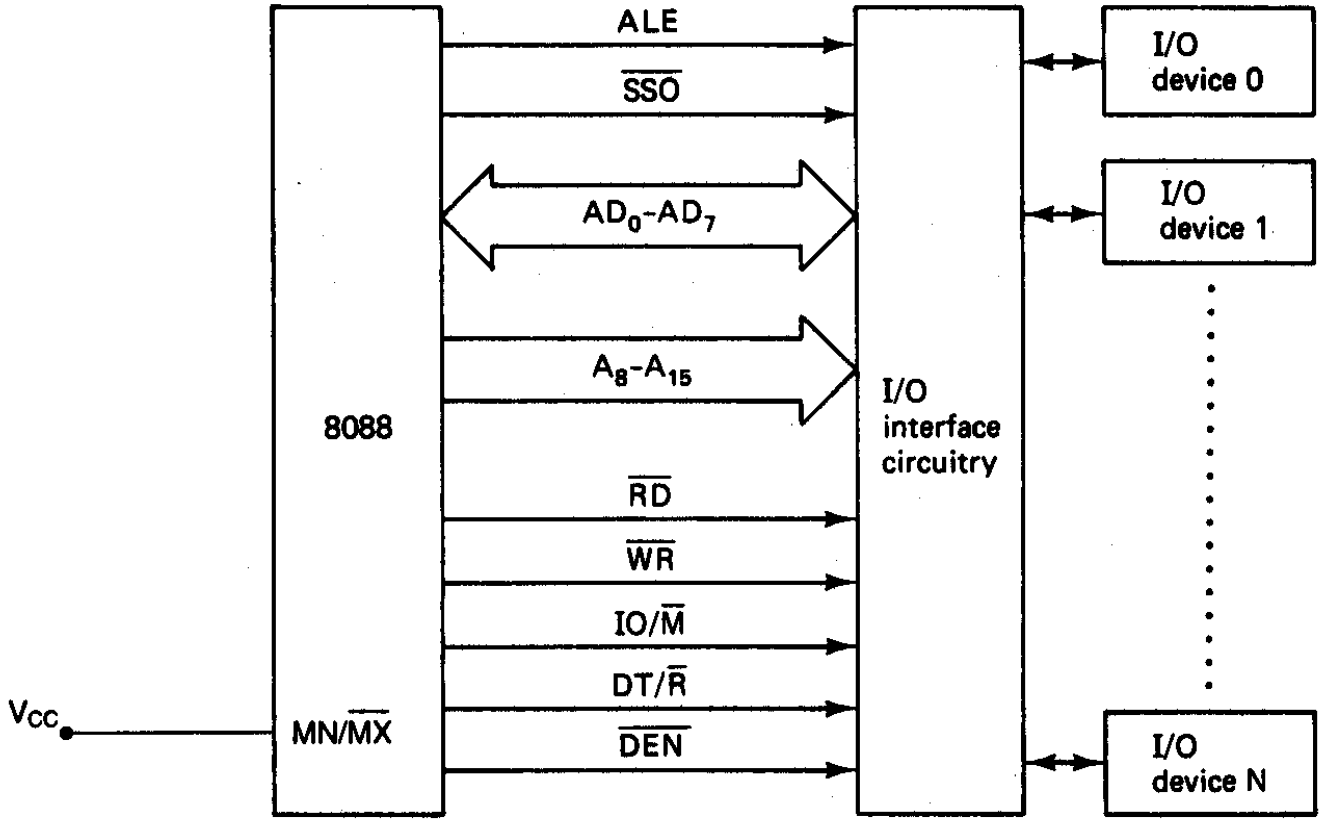
Minimum-mode interface:

- Similar in structure and operation to memory interface
  - I/O devices—can represent LEDs, switches, keyboard, serial communication port, printer port, etc.
  - I/O data transfers take place between I/O devices and MPU over the multiplexed-address data bus

    AD0-AD7
    A8-A15
- Control signal review
  - ALE = pulse to logic 1 tells bus interface circuitry to latch I/O address
  - RD* = logic 0 tells the I/O interface circuitry that an input (read) is in progress
  - WR*= logic 0 tells the I/O interface circuitry that an output (write) is in progress
  - IO/M*= logic 1 tells I/O interface circuits that the data transfer operation is for the IO subsystem
  - DT/R* = sets the direction of the data bus for input (read) or output (write) operation
  - DEN*= enables the interface between the I/O subsystem and MPU data bus

8.15 Isolated Input/Output Interface

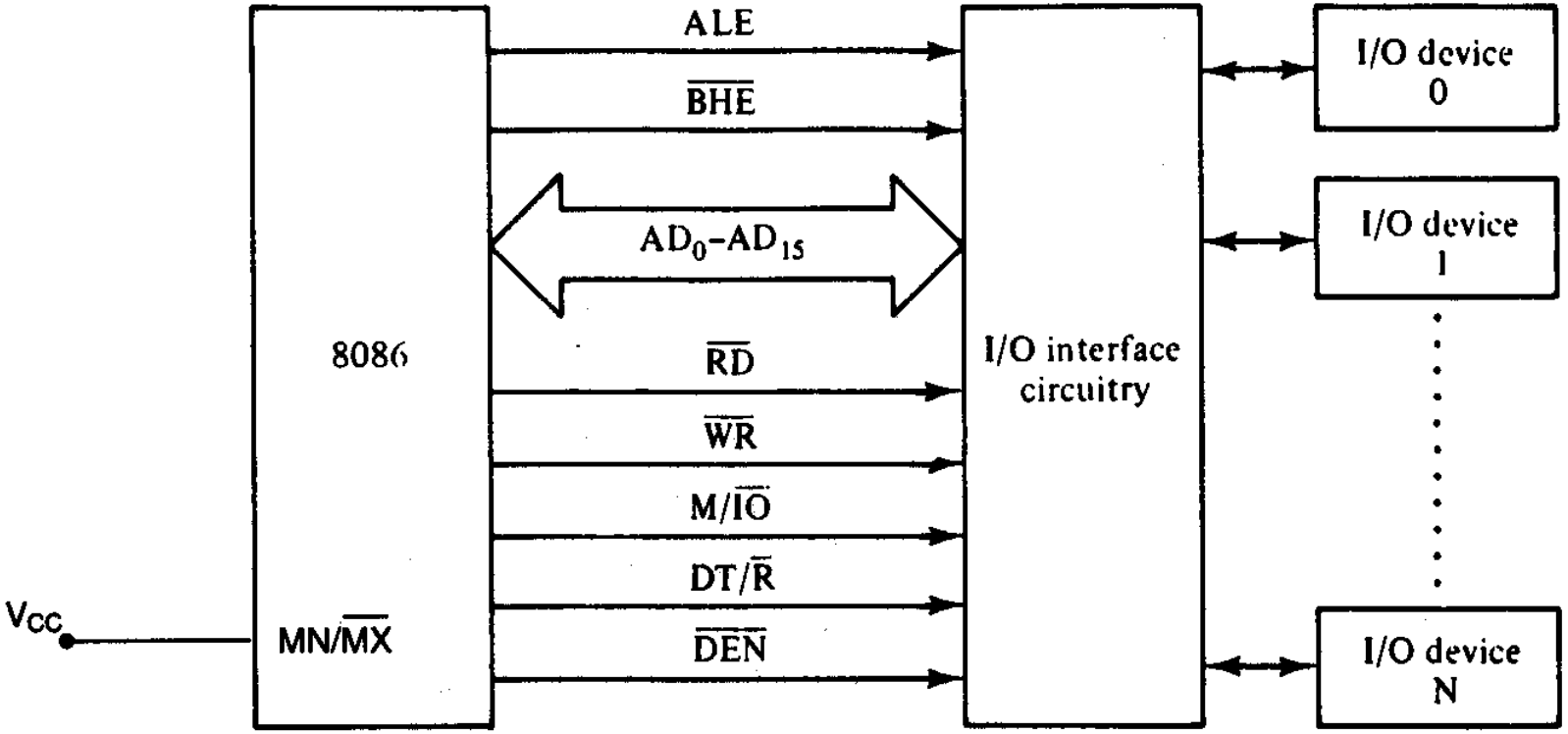
- Minimum-mode interface:



Minimum-mode 8088 system I/O interface

8.15 Isolated Input/Output Interface

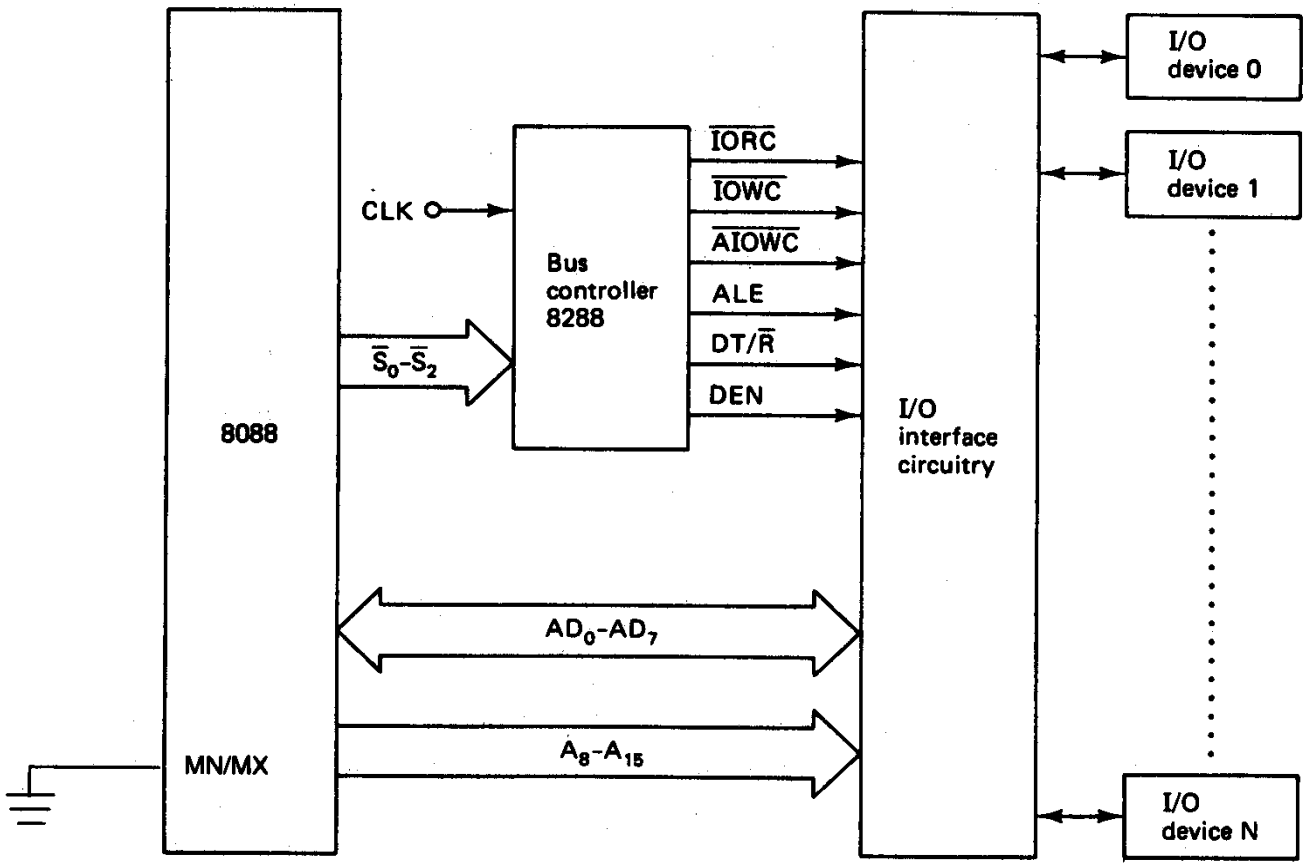
- Minimum-mode interface:



Minimum-mode 8086 system I/O interface

8.15 Isolated Input/Output Interface

- Maximum -mode interface:



Maximum-mode 8088 system I/O interface

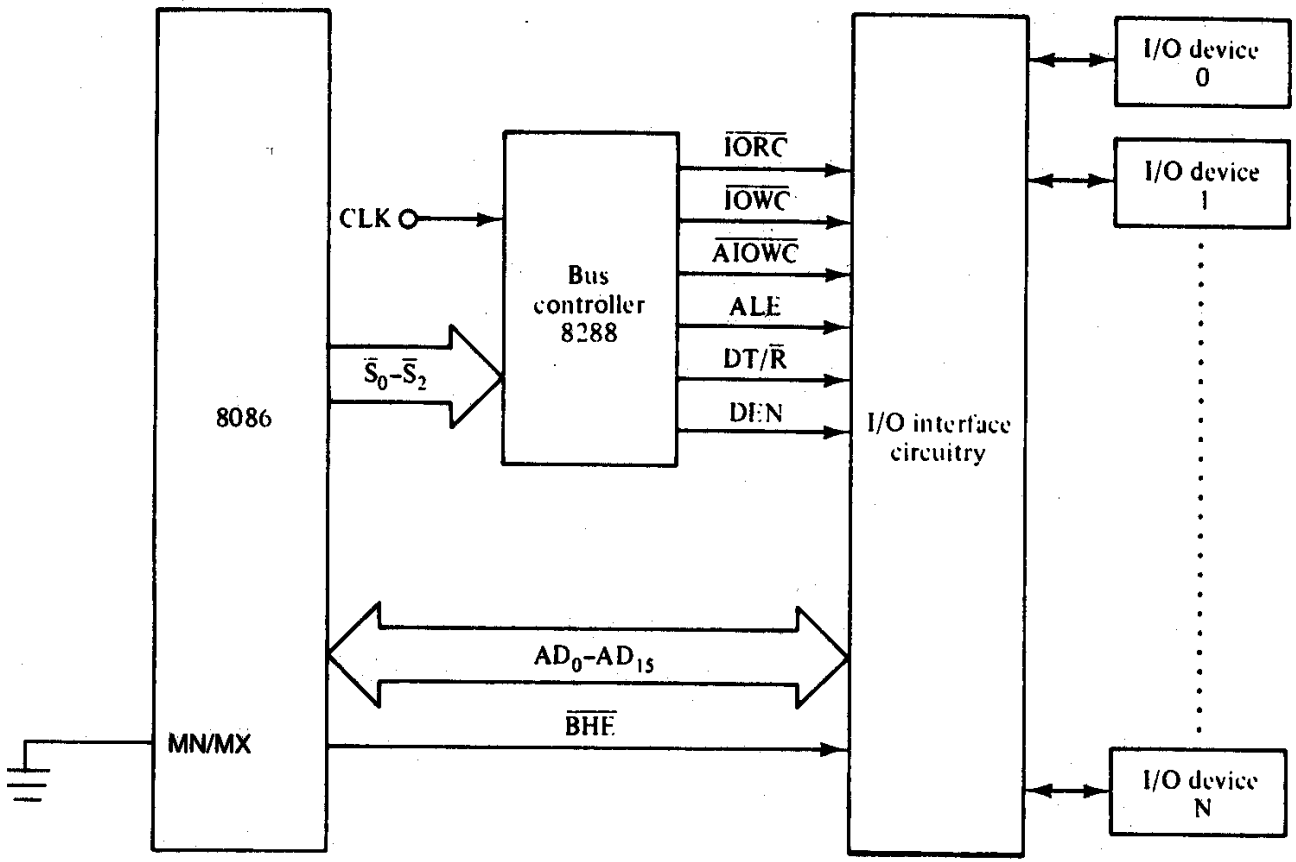
8.15 Isolated Input/Output Interface

- Maximum -mode interface:

- 8288 bus controller produces the control signals
- Signal changes
  - IORC* replaces RD*
  - IOWC* and AIOWC* replace WR*
  - DEN is complement of DEN*
  - IO/M* no longer needed (bus controller creates separate IO read/write controls)
  - SSO* no longer part of interface

8.15 Isolated Input/Output Interface

- Maximum –mode interface:



Maximum-mode 8086 system I/O interface

8.15 Isolated Input/Output Interface

- Maximum –mode interface:

Status inputs			CPU cycle	8288 command
$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$		
0	0	0	Interrupt acknowledge	$\overline{INTA}$
0	0	1	Read I/O port	$\overline{IORC}$
0	1	0	Write I/O port	$\overline{IOWC}$, $\overline{AIOWC}$
0	1	1	Halt	None
1	0	0	Instruction fetch	$\overline{MRDC}$
1	0	1	Read memory	$\overline{MRDC}$
1	1	0	Write memory	$\overline{MWTC}$, $\overline{AMWC}$
1	1	1	Passive	None

I/O bus cycle status codes

- Bus status code review
  - During all I/O accesses one of two bus cycle status code are output by the MPU
    - Read I/O port
    - Write I/O port
  - 8288 decodes to produce appropriate control command signals
    - IORC* → input (read I/O)
    - IOWC* → output (write I/O)
    - AIOWC* → output (write I/O)

8.16 Input/Output Data Transfers

- Input/output data transfers in the 8088 and 8086 microcomputers can be either byte-wide or word-wide.
- The port that is to be addressed is specified by the IO address.
- I/O addresses are 16 bits in length and are output by the 8088 to the I/O interface over bus lines AD_0 through AD_7 and A_8 through A_{15}.why?
- In 8088, the word transfers is performed as two consecutive byte-wide data transfer and takes two bus cycle.
- In 8086, the word transfers can takes either one or two bus cycle.
- Word-wide I/O ports should be aligned at even-address boundaries to ensure that one cycle is enough to complete the word operation.

8.17 Input/Output Instructions

- Types of instructions
  - Direct I/O instructions—only allow access to ports at page 0 addresses
  - Variable I/O instructions—allows access of ports anywhere in the I/O address space
- Direct I/O instructions

  IN Acc,Port

  OUT Port,Acc
  - Port = 8-bit direct address—limited to 0H through FFH (page 0)
  - Acc = accumulator register AX (word transfer); AH or AL (byte transfer)
  - Example:

    IN AL, 0FEH

    (FE) → AL (byte input operation)
  - Also known as accumulator I/O—because source or destination must always be in accumulator (Acc) register

Mnemonic	Meaning	Format	Operation	
IN	Input direct	IN Acc,Port	(Acc) ← (Port)	Acc = AL or AX
	Input indirect (variable)	IN Acc,DX	(Acc) ← ((DX))	
OUT	Output direct	OUT Port,Acc	(Port) ← (Acc)	
	Output indirect (variable)	OUT DX,Acc	((DX)) ← (Acc)	

8.17 Input/Output Instructions

- Types of instructions
  - Direct I/O instructions—only allow access to ports at page 0 addresses
  - Variable I/O instructions—allows access of ports anywhere in the I/O address space
- Variable I/O instructions

```
IN Acc,DX
OUT DX,Acc
```

  - DX = 16-bit indirect address—allows access to full I/O address space
  - Acc = accumulator register AX (word transfer); AH or AL (byte transfer)
  - Example:

```
MOV DX,A000H ;load I/O address
IN AL,DX ;input value to AL
MOV BL,AL ;copy value to BL
```

(A000H) → BL (byte input operation)

Mnemonic	Meaning	Format	Operation	
IN	Input direct	IN Acc,Port	(Acc) ← (Port)	Acc = AL or AX
	Input indirect (variable)	IN Acc,DX	(Acc) ← ((DX))	
OUT	Output direct	OUT Port,Acc	(Port) ← (Acc)	
	Output indirect (variable)	OUT DX,Acc	((DX)) ← (Acc)	

8.17 Input/Output Instructions

- EXAMPLE

  Write a sequence of instructions that will output the data FFH to a byte-wide output port at address ABH of the I/O address space.

- Solution:

  First, the AL register is loaded with FFH as an immediate operand in the instruction

  MOV AL, FFH

  Now the data in AL can be output to the byte-wide output port with the instruction

  OUT ABH, AL

8.17 Input/Output Instructions

- EXAMPLE

  Write a series of instructions that will output FFH to an output port located at address B000H of the I/O address space.

- Solution:

  The DX register must first be loaded with the address of the output port. This is done with the instruction

  MOV DX, B000H

  Next, the data that are to be output must be loaded into AL with the instruction

  MOV AL, FFH

  Finally, the data are output with the instruction

  OUT DX, AL

8.17 Input/Output Instructions

- EXAMPLE

  Data are to be read in from two byte-wide input ports at addresses AAH and A9H and then output as a word-wide output port at address B000H. Write a sequence of instructions to perform this input/output operation.

- Solution:

  First read in the byte at address AAH into AL and move it into AH.

```
IN AL, AAH
MOV AH, AL
```

Now the other byte can be read into AL by the instruction

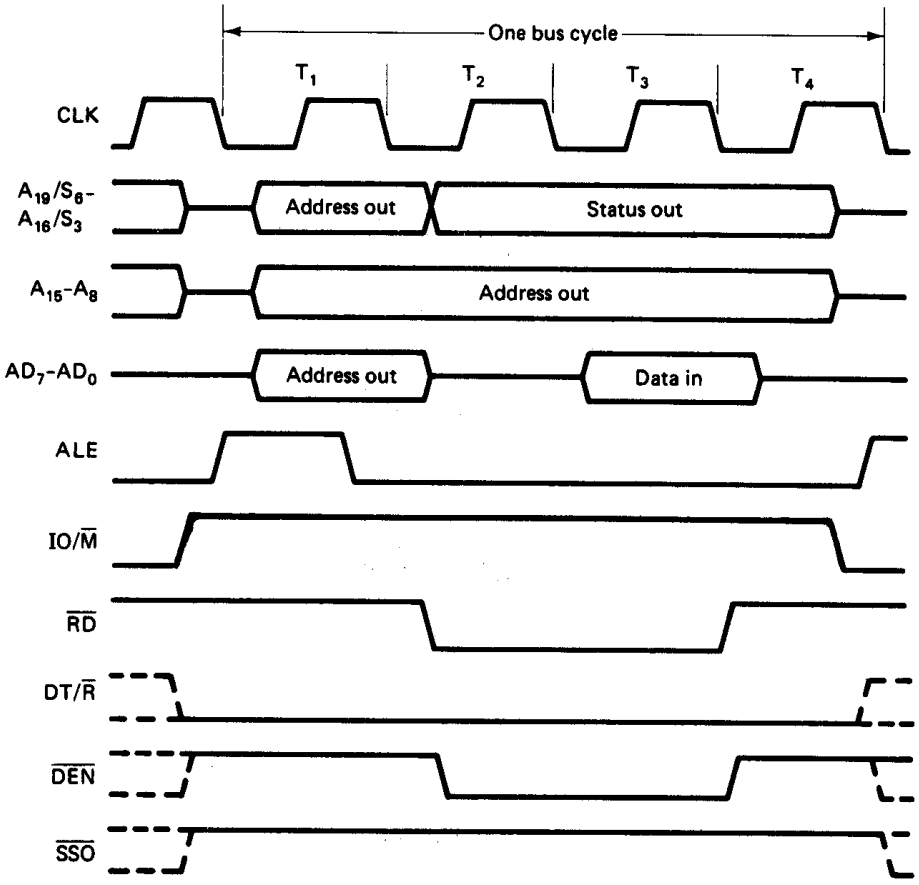
```
IN AL, A9H
```

And to write out the word of data

```
MOV DX, B000H
OUT DX, AX
```

8.18 Input/Output Bus Cycle

- Input bus cycle of the 8088



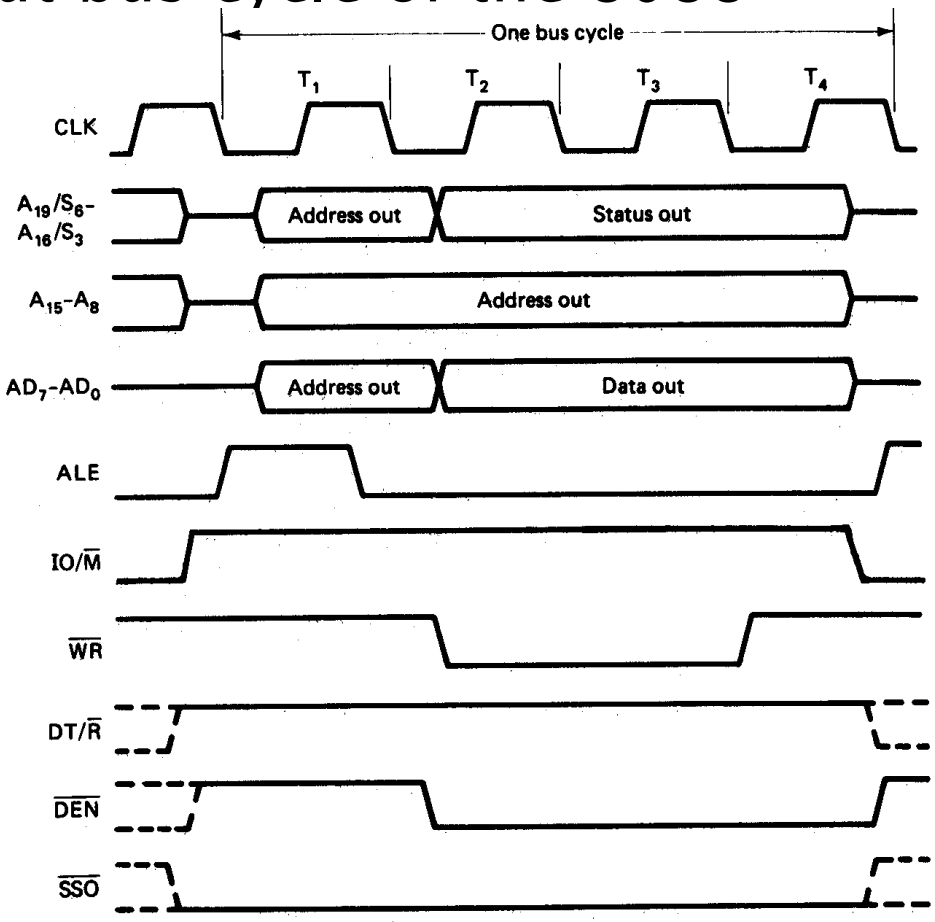
8.18 Input/Output Bus Cycle

- Input bus cycle of the 8088

- Input (I/O read) bus cycle timing diagram—shows relationship between signals relative to time states
  - T1 state—input cycle begins
    - Address output on A0–A15
    - Pulse produced at ALE--address should be latched in external circuitry on trailing edge of ALE
    - IO/M* set to 1→ I/O bus cycle
    - DT/R* set to 0 → set external data bus control circuitry for receive mode (input)
  - T2 state
    - Status code output on S3–S6
    - AD0 through AD7 tri-stated in preparation for data bus operation
    - RD* set to 0 → input cycle
    - DEN* set to 0 → enable external data bus control circuitry
  - T3 state
    - Data on D0–D7 input (read) by the MPU
  - T4 state—input cycle finishes
    - RD* returns to 1→ inactive level
    - Complete address/data bus tri-stated
    - IO/M* returned to 0 → memory bus cycle
    - DEN* returned to 1→ inactive level
    - DT/R* returns to 1→ transmit level

8.18 Input/Output Bus Cycle

- Output bus cycle of the 8088



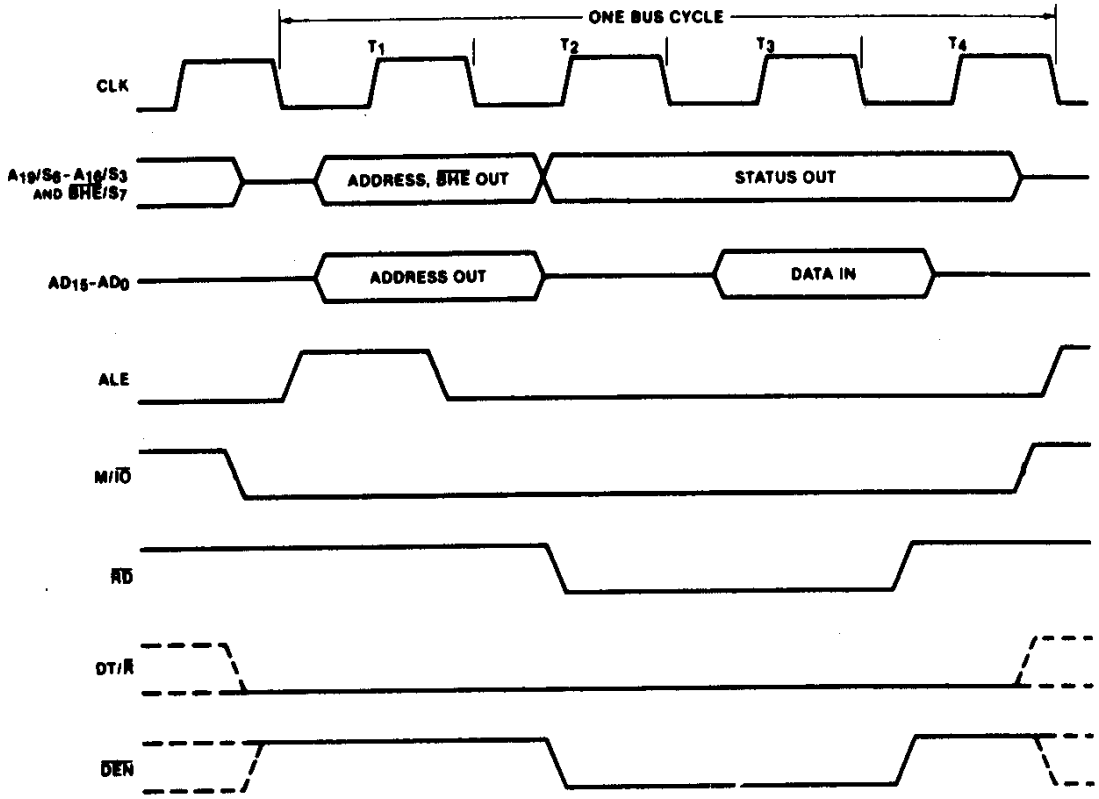
8.18 Input/Output Bus Cycle

Output bus cycle timing diagram of the 8088

- T1 state—output cycle begins
  - Address output on A0-A15
  - Pulse produced at ALE and address latched in external circuitry on trailing edge of ALE
  - IO/M* set to 1→ I/O bus cycle
  - DT/R* set to 1→ external data bus control circuitry for transmit mode (output)
- T2 state
  - Status code output on S3-S6
  - AD0 through AD7 transitioned to data bus and output data placed on bus
  - DEN* set to 0 → enable external data bus control circuitry
  - WR* set to 0· output cycle
- T3 or T4 state
  - Data on D0-D7 output (write) into I/O port (I/O device decides when!)
- T4 state—output cycle finishes
  - WR* returns to 1→ inactive level
  - Complete address/data bus tri-stated
  - IO/M* returned to 0 → memory bus cycle
  - DEN* returned to 1→ inactive level

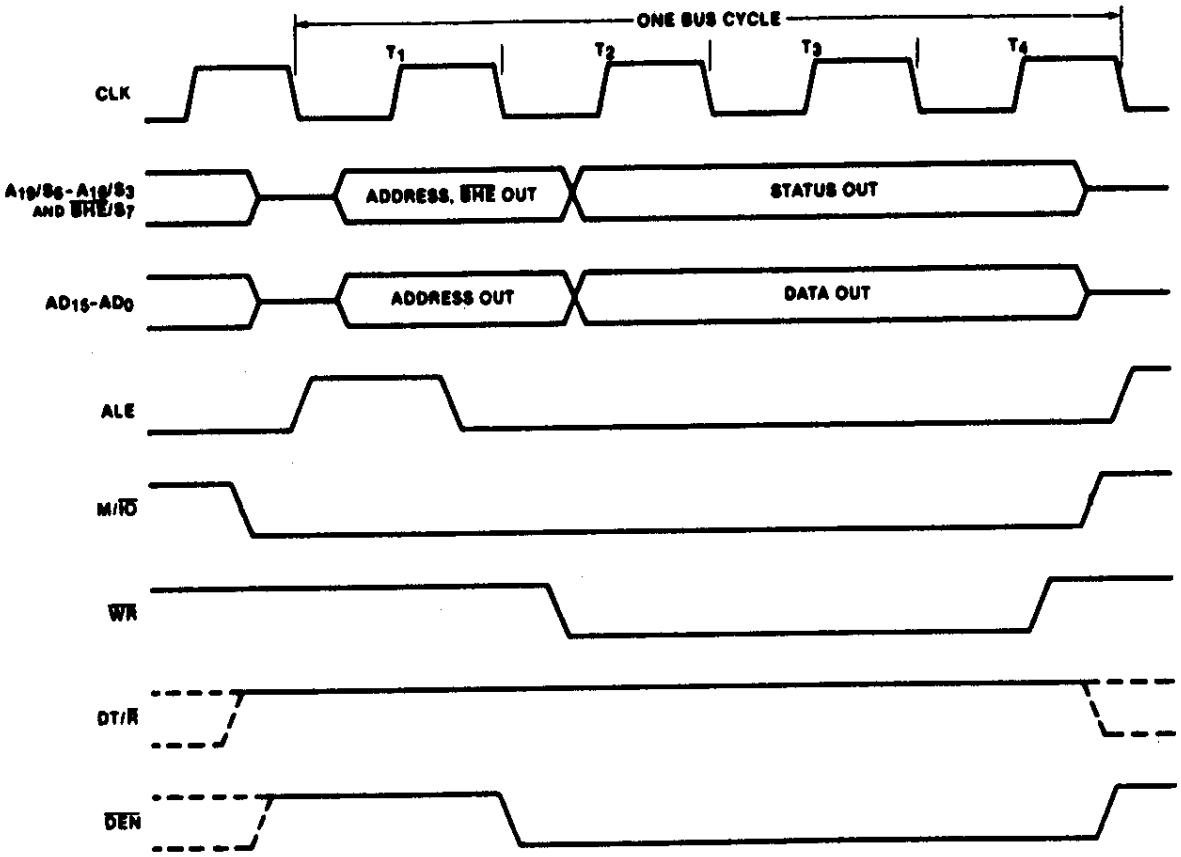
8.18 Input/Output Bus Cycle

- Input bus cycle of the 8086



8.18 Input/Output Bus Cycle

- Output bus cycle of the 8086



H.W. #8

- Solve the following problems from Chapter 8 from the course textbook:

  8, 22, 26, 35, 39, 49, 55, 59, 66, 86, 89, 99, 101, 107