Prepared by: Dr. Mohammed A. Al-Ebadi

The 8088/8086 Microprocessors and their Memory and Input/Output Interface

The **8086** and **8088** was announced in **1978 and 1979**, respectively, introduced by Intel Corporation.

The **8088** is internally a **16-bit MPU**. However, it has a 8-bit data bus externally. It has the ability to address up to **1 Mbyte** of memory via its **20-bit address bus**. In addition, it can address up to **64K of byte-wide input/output ports**.

The **8086** is internally a **16-bit MPU** and also it has a 16-bit data bus externally. This is the mean deference between 8088 and 8086 Microprocessors.

The 8088/8086 manufactured using **high-performance metal-oxide semiconductor (HMOS) technology** and the circuitry on their chips are equivalent to approximately **29,000 transistors**.

The 8088/8086 is housed in a **40-pin dual in-line package**. The signals pinned out to each lead are shown in Fig. (1).

The **address bus lines** A_0 through A_7 and **data bus lines** D_0 through D_7 are multiplexed in 8088 MPU. For this reason, these leads are labeled AD_0 through AD_7. By *multiplexed* we mean that the same physical pin carries an address bit at one time and the data bit at another time.

For 8086 MPU, the **address bus lines** A_0 through A_{15} and **data bus lines** D_0 through D_{15} are **multiplexed** and labeled AD_0 through AD_{15}.

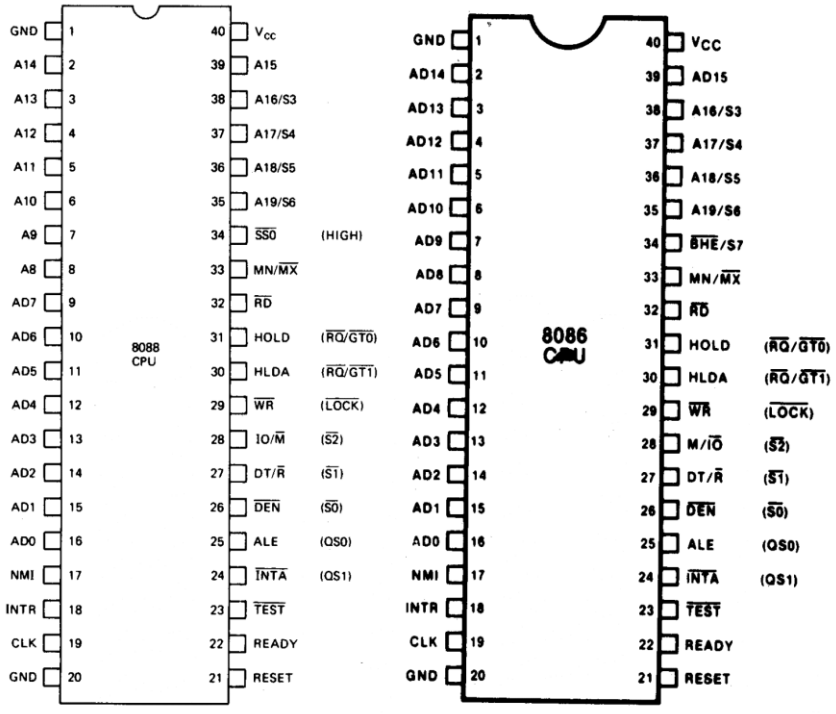


Fig. (1) 8088 and 8086 MPUs.

Minimum-Mode and Maximum-Mode

The 8088/8086 can be configured to work in either of two modes:

The **minimum** mode is selected by applying logic 1 to the MN/$\overline{MX}$ input lead. It is typically used for smaller single microprocessor systems. The **maximum** mode is selected by applying logic 0 to the MN/$\overline{MX}$ input lead. It is typically used for larger multiple microprocessor systems.

Depending on the mode of operation selected, the assignments for a number of the pins on the microprocessor package are changed. The pin functions specified in parentheses pertain to the maximum-mode. Table 1 a, b and c describe the common signal, minimum mode signals, and maximum mode signals of 8088 MPU, respectively.

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)

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)

Table 1

Minimum-Mode Interface Signals – 8088 MPU

In minimum mode, the 8088 itself provides all the control signals needed to implement the memory and I/O interfaces (see Fig. (2)).

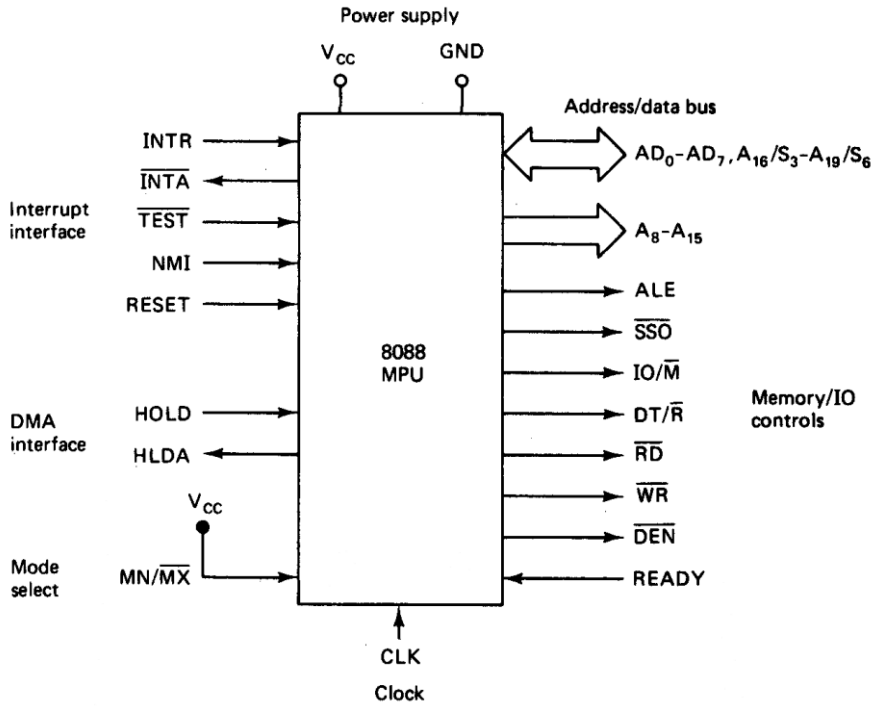


Fig. (2) Block diagram of minimum mode of 8088 and 8086 MPUs

The minimum mode signals can be divided into five basic groups: Address/Data bus, status, control, interrupt, and direct memory access (DMA).

Address/Data Bus: The address bus is 20 bits long and consists of signal lines A0 (LSB) through A19 (MSB). However, only address lines A0 through A15 are used when accessing I/O. The data bus lines are multiplexed with address lines. For this reason, they are denoted as AD0 through AD15. Data line D0 is the LSB.

Status Signals: The four most significant address lines A16 through A19 of the 8086 are multiplexed with status signals S_3 through S_6. These status bits are output on the bus at the same time that data are transferred over the bus lines AD0-AD7. S_3 and S_4 bits identify which the internal segment register was used to generate the physical address that was output on the address bus during the current bus cycle. Table 2 shows the S_4S_3 code and the corresponding segment register.

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)

Table 2

Control Signals: The control signals are provided to support the memory and I/O interface of 8088 MPU. They control functions such as when the bus carries a valid address, which direction data are transferred over the bus, when valid write data are on the bus, and when to put read data on the system bus.

• When Address latch enable (ALE) is logic 1 it signals that a valid address is on the bus. This address can be latched in external circuitry on the 1-to-0 edge of the pulse at ALE.

• IO/$\overline{M}$ (Input-output/Memory) tells external circuitry whether a memory or I/O transfer is taking place over the bus. Logic 0 signals a memory operation and logic 1 signals an I/O operation.

• DT/$\overline{R}$ (data transmit/receive) signals the direction of data transfer over the bus. Logic 1 indicates that the bus is in the transmit mode (i.e., data are either written into memory or to an I/O device). Logic 0 signals that the bus is in the receive mode (i.e., reading data from memory or from an input port).

• $\overline{WR}$ (write) is switched to logic 0 to signal external devices that valid output data are on the bus.

• $\overline{RD}$ (read) indicates that the MPU is performing a read of data off the bus. During read operations, one other control signal, $\overline{DEN}$ (data enable), is also supplied. It enables external devices to supply data to the microprocessor when it is at logic 0.

• The READY signal can be used to insert wait states into the bus cycle so that it is extended by a number of clock periods. This signal is supplied by a slow memory or I/O subsystem to signal the MPU when it is ready to permit the data transfer to be completed.

• $\overline{SSO}$ is a status signal indicates that either instruction code read ($\overline{SSO}$=0), or data access ($\overline{SSO}$=1).

Interrupt Signals:

• Interrupt request (INTR) is an input to the 8086 that can be used by an external device to signal that it needs to be serviced. Logic 1 at INTR represents an active interrupt request.

• When the MPU recognizes an interrupt request, it indicates this fact to external circuits with logic 0 at the interrupt acknowledge ($\overline{INTA}$) output.

• On the 0-to-1 transition of nonmaskable interrupt (NMI), control is passed to a nonmaskable interrupt service routine at completion of execution of the current instruction. NMI is the interrupt request with highest priority and cannot be masked by software.

• The RESET input is used to provide a hardware reset for the MPU. Switching RESET to logic 0 initializes the internal registers of the MPU and initiates a reset service routine.

• The $\overline{\text{TEST}}$ input is an external interrupt signal. This input is examined by a 'WAIT' instruction. When $\overline{\text{TEST}}$ switches to 1, 8088 suspends operation and goes into what is known as the idle state. If $\overline{\text{TEST}}$ becomes 0, 8088 resumes the execution with the next instruction in the program.

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 logic level 1. 8088 will enter hold state at the completion of the current bus cycle.

• When in the hold state, lines AD0 through AD15, A16/S3 through A19/S6, IO/$\overline{\text{M}}$, DT/$\overline{\text{R}}$, $\overline{\text{WR}}$, $\overline{\text{RD}}$, $\overline{\text{DEN}}$ and INTR are all put in the high-Z state. The MPU signals external devices that it is in this state by switching HLDA to 1.

Minimum-Mode Interfaces– 8086 MPU Interface Differences

- Data bus

  • 16-bit wide D15-D0 multiplexed with A15 through A0 and allows three types of data transfers:

  ✓ Word—over D15-D0

  ✓ Low byte—over D7-D0

  ✓ High byte—over D15-D8

- Memory/IO Controls

  • $\overline{SSO}$ replaced with $\overline{BHE}$ (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.

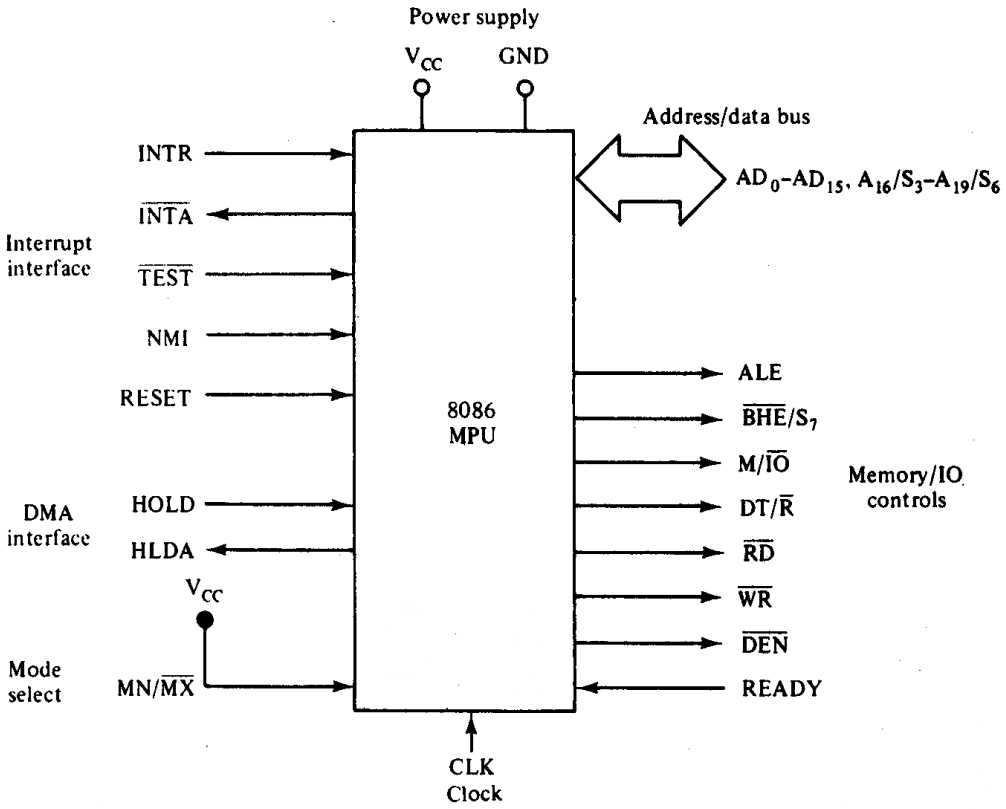


Fig. (3) 8086 minimum mode interface signals

Maximum-Mode Interface Signals – 8088 MPU

When the 8088 or 8086 microprocessor is set for the maximum-mode configuration, it produces signals for implementing multiprocessor/coprocessor system environment- By multiprocessor environment we mean that multiple microprocessors exist in the system and that each processor executes its own program. Usually in this type of system environment, some system resources are common to all processors. They are called global resources. There are also other resources that are assigned to specific processors. These dedicated resources are known as local or private resources.

In the maximum-mode system, facilities are provided for implementing allocation of global resources and passing bus control to other microprocessors sharing the system bus.

8288 Bus Controller: Bus Commands and Control Signals

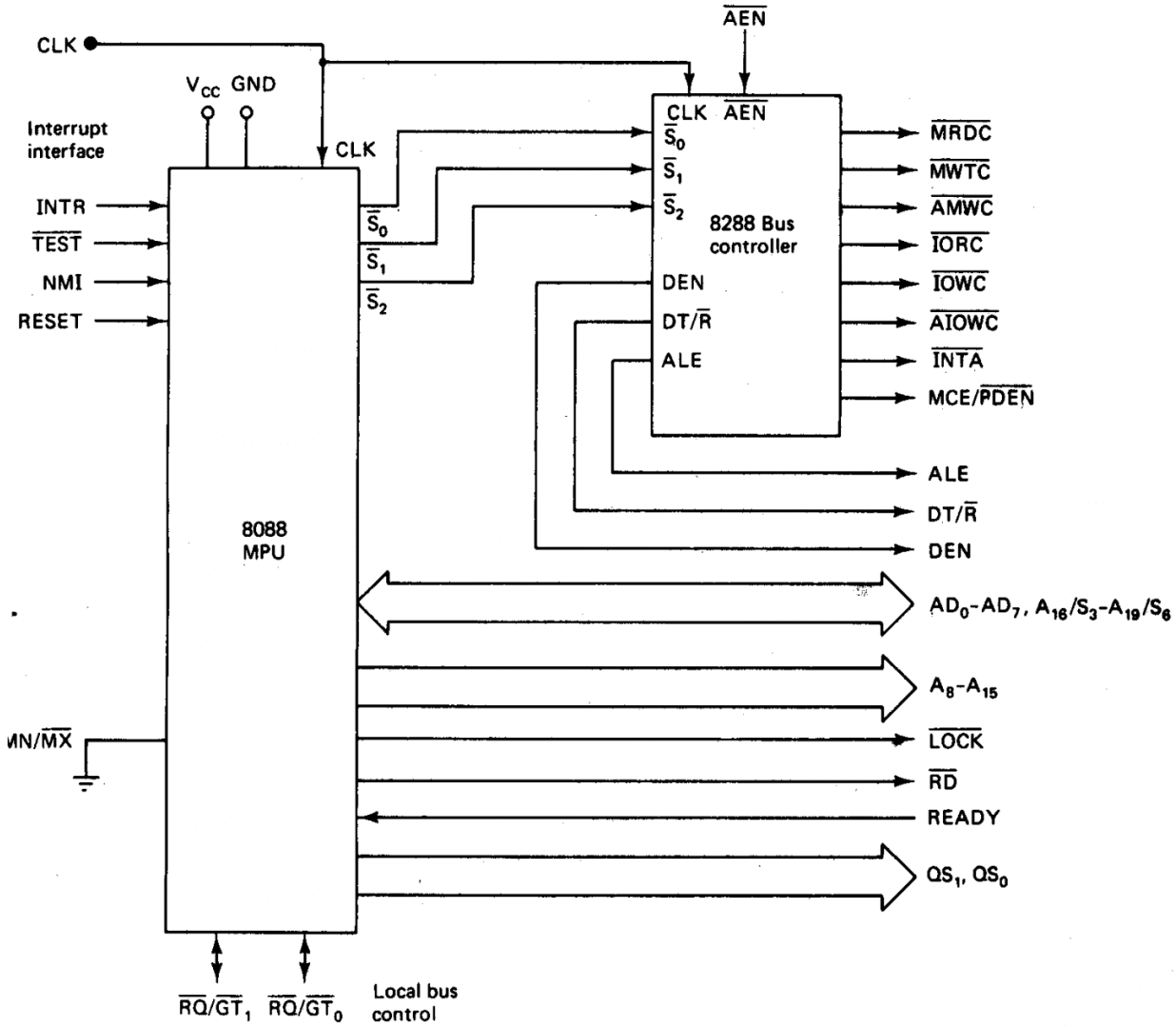


Fig. (4) 8088 maximum mode interface signals

Looking at the maximum-mode block diagram in Fig. (4), we see that the 8088 does not directly provide all the signals that are required to control the memory, I/O, and interrupt interfaces. Specifically, the $\overline{WR}$, IO/$\overline{M}$, DT/$\overline{R}$, $\overline{DEN}$, ALE, and $\overline{INTA}$ signals are no longer produced by the 8088. Instead, it outputs a status code on three signals lines $\overline{S_0}$, $\overline{S_1}$, and $\overline{S_2}$, prior to the initiation of each bus cycle. This 3 bit bits bus status identifies which type of bus cycle is to follow.

$\overline{S_2}\overline{S_1}\overline{S_0}$ are input to the external bus controller device, the 8288, which decodes them to identify the type of MPU bus cycle. The block diagram of the 8288 is shown in Fig. (5).

In response, the bus controller generates the appropriately timed command and control signals.

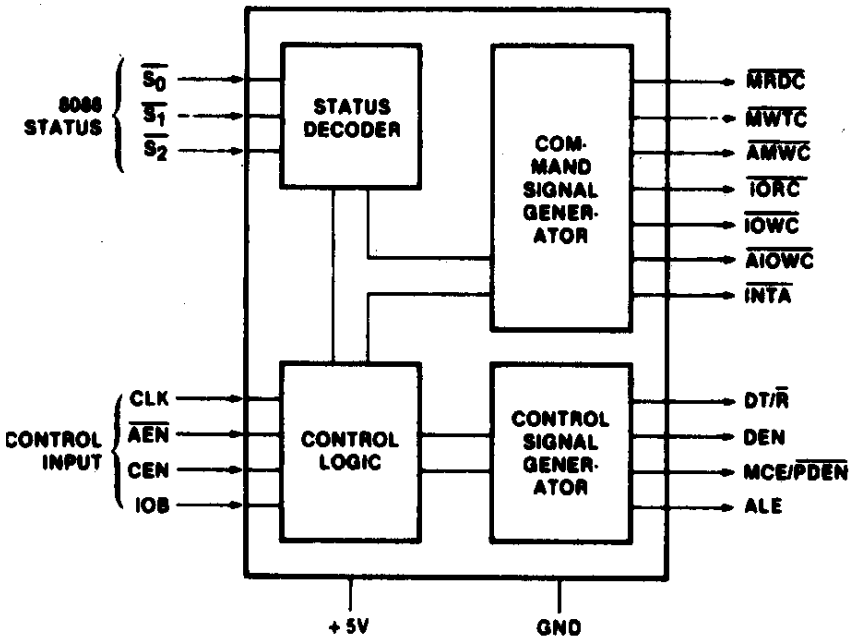


Fig. (5) Block diagram of the 8288 bus controller

Table 3 shows the relationship between the status codes and the type of bus cycle.

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

Table 3

The output signals generated to tell external circuitry which type of bus cycle is taking place. The output signals are:

$\overline{\text{MRDC}}$ Memory Read Command

$\overline{\text{MWRC}}$ Memory Write Command

$\overline{\text{AMWC}}$ Advanced Memory Write Command

$\overline{\text{IORC}}$ Input Output Read Command

$\overline{\text{IOWC}}$ Input Output Write Command

$\overline{\text{AIOWC}}$ Advanced Input Output Write Command

$\overline{\text{INTA}}$ Interrupt Acknowledge

The 8288 produces one or two of these seven command signals for each bus cycle. For instance, when the 8088 outputs the code $\overline{S_2}\,\overline{S_1}\,\overline{S_0}$ = 001, it indicates that an I/O read cycle is to be performed. In turn, the 8288 makes its $\overline{\text{IORC}}$ output switch to logic 0. On the other hand, if the code 111 is output by the 8088, it is signalling that no bus activity is to take place; the 8288 produces no command signals.

The other control outputs produced by the 8288 consist of $\overline{\text{DEN}}$, DT/$\overline{\text{R}}$, and ALE. These three signals provide the same functions as those described for the minimum mode.

LOCK Signal ($\overline{\text{LOCK}}$)

To implement a multiprocessor system, a signal called lock ($\overline{\text{LOCK}}$) is provided on the 8088 and 8086. This signal is meant to be output (logic 0) whenever the processor wants to lock out the other processors from using the bus. This would be the case when a shared resource is accessed. The LOCK signal is compatible with the *Multibus*, an industry standard for interfacing microprocessor systems in a multiprocessor environment.

Queue Status Signals

There are two queue status signals (QS_1, QS_0) produced by 8088 in the maximum mode system. They formed 2-bit code $QS_1QS_0$to tell the external circuitry what type of information was removed from the instruction queue during the previous clock cycle. Table 4 shows that.

QS1	QS0	Queue Status
0 (low)	0	No Operation. During the last clock cycle, nothing was taken form 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 of instruction.
1	1	Subsequent Byte. The byte taken from the queue was a subsequent byte of the instruction.

Table 4

Local Bus Control Signals

In a maximum mode configuration, the minimum-mode HOLD and HLDA interface of the 8088/8086 is also changed. These two signals are replaced by request/grant lines $\overline{RQ}/\overline{GT}_0$ and $\overline{RQ}/\overline{GT}_1$. They provide a prioritized bus access mechanism for accessing the local bus.

Maximum-Mode Interfaces– 8086 MPU Interface Differences

Differences from 8088 maximum mode interface in that it has 16-bit multiplexed data bus and $\overline{BHE}$ output.

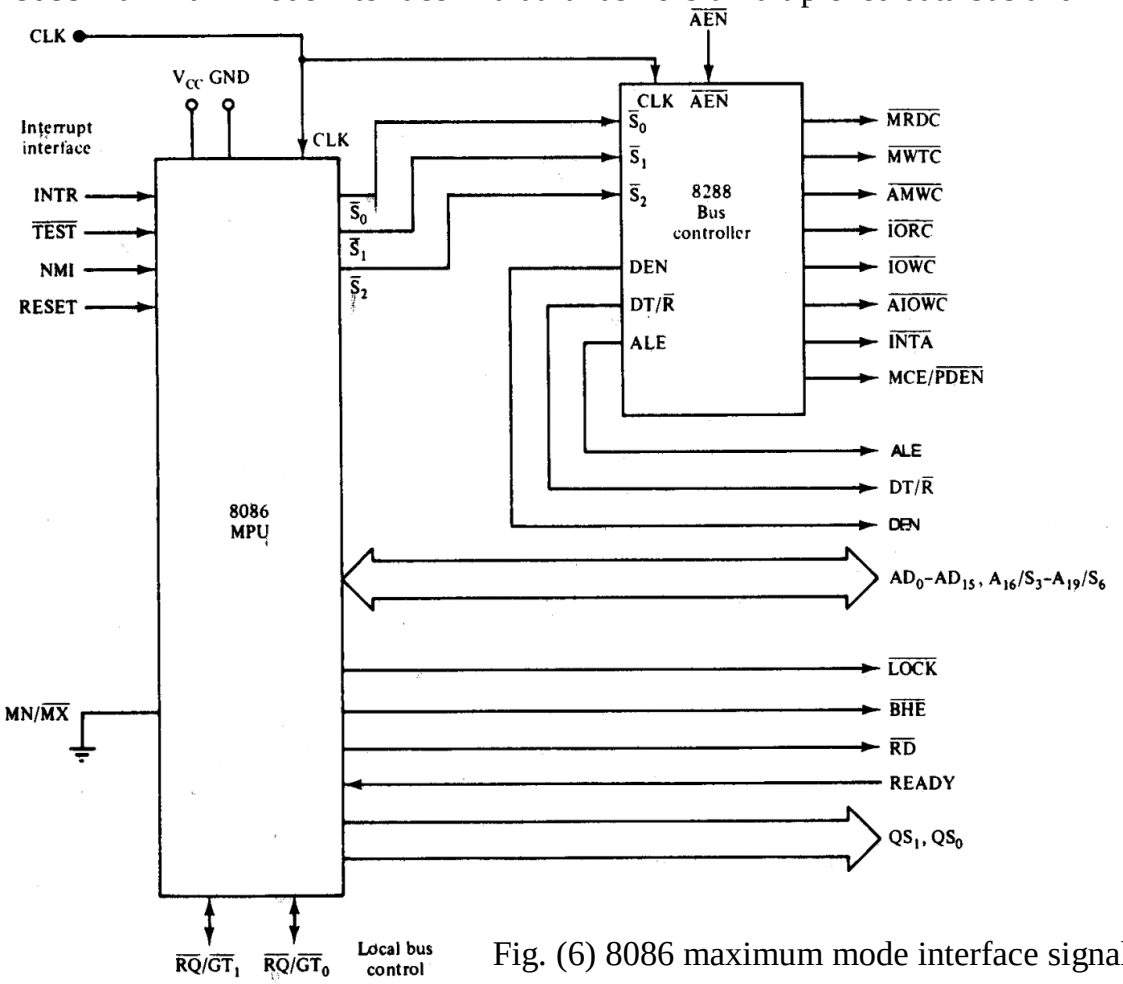


Fig. (6) 8086 maximum mode interface signals

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. The 8284 clock generator and driver IC generates CLK signal. Figure (7) is a block diagram of this device.

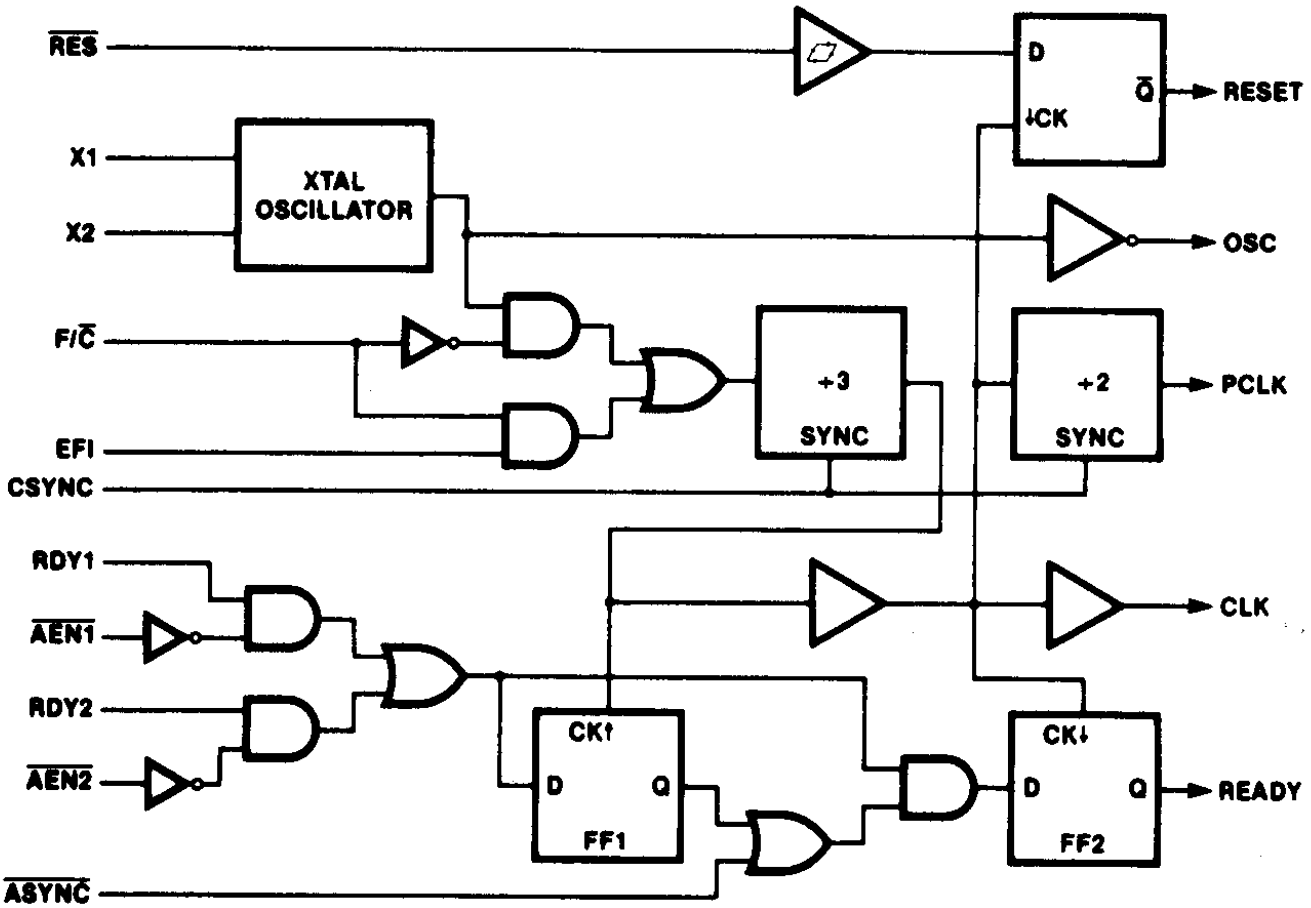


Fig. (7) Block diagram of 8284 clock generator

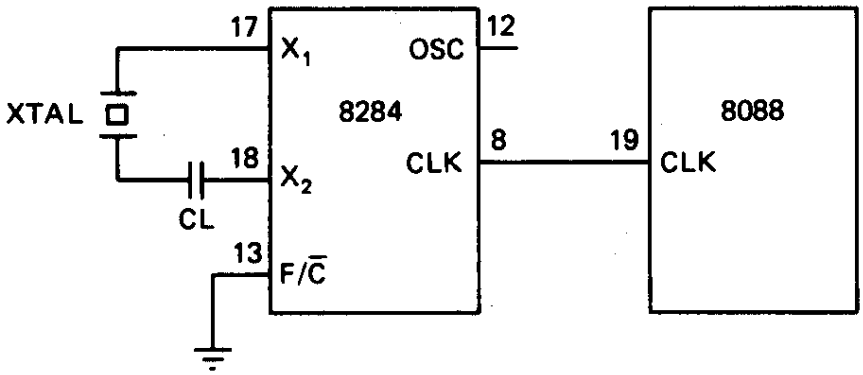


Fig. (8) connection of 8284 with 8088 MPU.

Because of 8088 are provided with 5 MHz and 8 MHz speeds, the crystal is provided between X1 and X2 inputs of 8284 either with 15 MHz or 24 MHz clock frequency in order to generate 5 MHz or 8 MHz clock frequency, respectively, by divided by 3 within the 8284 IC. Clock signal will output at CLK pin of 8284 that connected to CLK input pin of 8088 MPU, Fig. (8).

The peripheral clock PCLK output of 8284 is a signal with a half frequency of CLK output.

The oscillator clock OSC output is a signal at the crystal frequency which is three times that of CLK signal.

Fig. (9) shows the waveforms of OSC, CLK, and PCLK signals generated by 8284 IC.

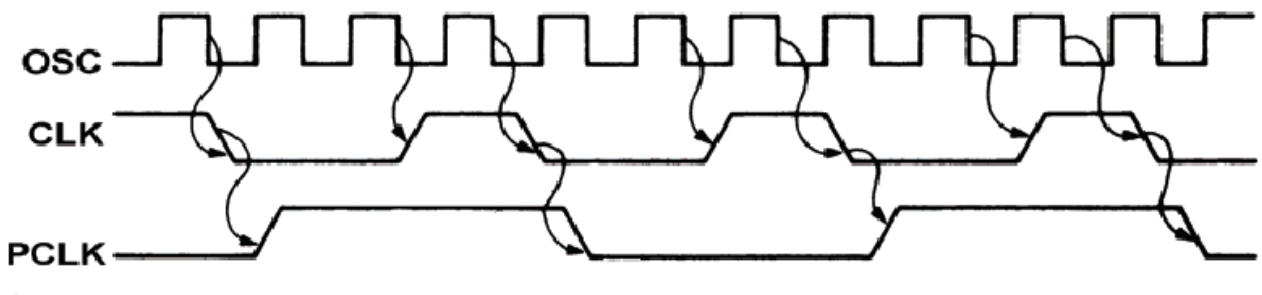


Fig. (9) OSC, CLK, and PCLK signals produced by 8284 IC.

The 8284 can be also driven from an external clock source applied to the external frequency input EFI. Input F/C is provided for clock source selection (0 for Crystal and 1 for External frequency). Clock synchronization CSYNC input can be used for external synchronization in systems that employ multiple clocks.

The 8086 is manufactured in three speeds: 5 MHz, 8 MHz and 10 MHz. For 8086, we connect either a 15, 24 or 30 MHz crystal between inputs X1 and X2 inputs of the 8284.

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 8086 microprocessor consists of at least four clock periods. These four time states are called T_1, T_2, T_3 and T_4.

During T_1, the MPU put an address on the bus. For a ***write memory cycle,*** data are put on the bus during stateT_2, and maintained through T_3 and T_4. When a ***read memory cycle*** is to be performed, the bus is first put in the high-Z state during T_2, and then the data to be read must be available on the bus during T_3 and T_4. These four clock states give a bus cycle duration of 125 ns * 4 = 500 ns in an 8-MHz 8088 system. Figure (10) shows the bus cycle and time states.

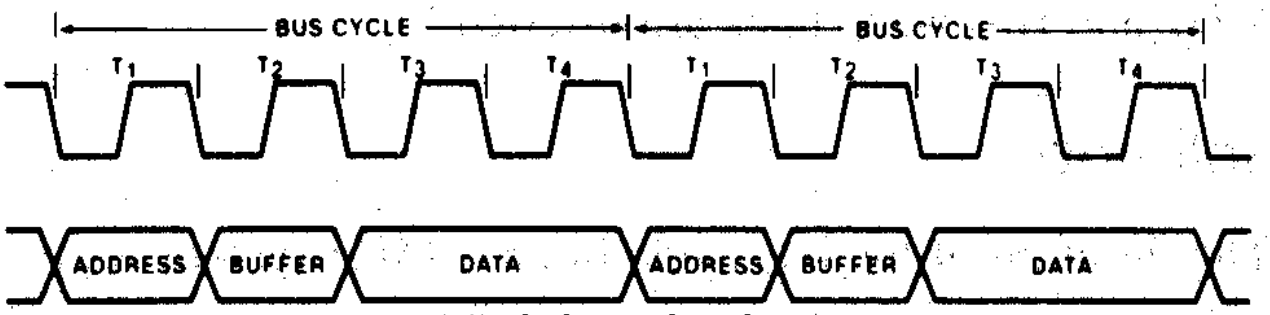


Fig. (10) The bus cycle and time states

- *Idle state:* If MPU does not performed any bus cycle, it will be on idle state which take one clock period long, and any number of idle states can be inserted between bus cycles. Figure (11) shows two bus cycles separated by idle states.

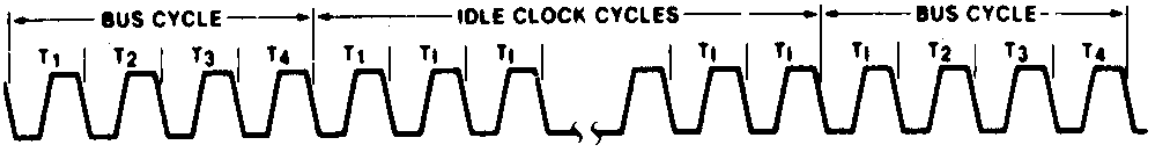


Fig. (11) The Idle state.

- *Wait state:* wait state can be inserted into a bus cycle. This is done by requesting from an external hardware. The READY input of the 8088/8086 MPU is provided for this purpose. Figure (12) shows the wait state generated in response to READY signal.

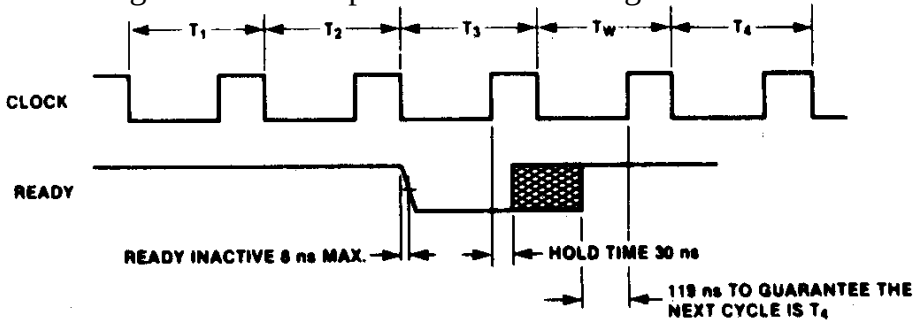


Fig. (12) The wait state.

In the Figure (12) above, T_W will be inserted between T_3 and T_4 because READY input becomes at logic 0 (when the memory or I/O device is not able to respond in the duration of a bus cycle), and still at this level until it returns to logic 1 to complete the current bus cycle with T_4. That is mean, the data will be maintained on bus during T_3 and T_W until T_4 was complete.

If there are two wait states during 8MHz bus cycle, the total time of complete bus cycle will be equal 750ns (500 ns + 2*(125 ns)).

Hardware Organization of the Memory Address Space

1– 8088 Microprocessor

• 8088 memory hardware is organized as a single byte-wide memory bank.

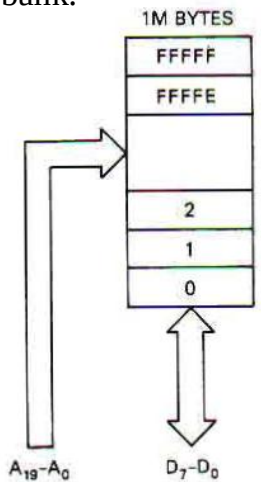
• Size: 1M x 8 bits.

• Physical address range: 00000H - FFFFFH.

• Address/data bus demultiplexed in external hardware.

• Input: 20-bit address bus, A_{19} through A_0.

• Input/Output: 8-bit data bus, D_7 through D_0.



- Byte access bus cycle

• MPU applies address of storage location to be accessed over address lines A_{19}-A_0.

• Byte of data written into or read from address X transferred over data lines D_0 through D_7.

• Byte access takes a minimum of one bus cycle of duration.

@5MHz—800ns

@8MHz—500ns

- 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

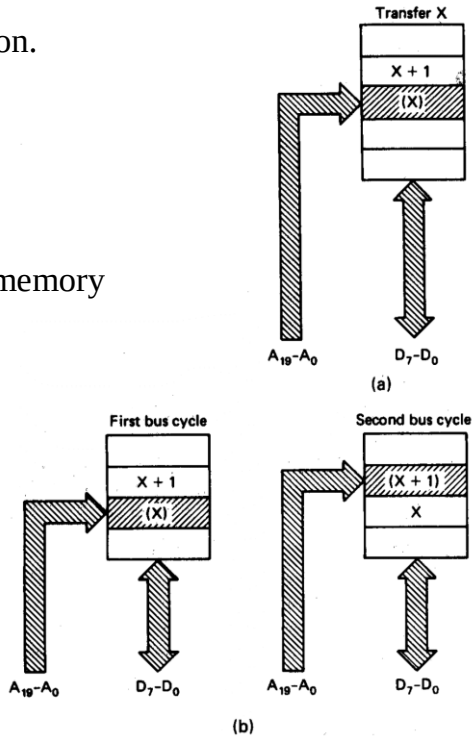


Fig.(13)- a) Byte transfer, b) Word transfer.

2– 8086 Microprocessor

• 8086 memory hardware is organized as a two byte-wide memory banks.

• Bank size is 512K x 8 bits

• Low-bank holds even addressed bytes

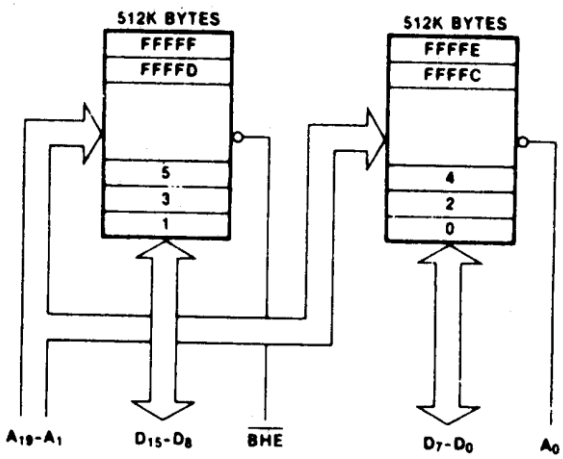
00000H through FFFFEH.

• High-bank holds odd addressed bytes

00001H through FFFFFH.

• Address/data bus demultiplexed in external hardware.

• Input: 20-bit address bus A_{19} through A_0, and $\overline{BHE}$.



A_1-A_{19} = selects storage location.

A_0 = 0 enables low bank.

$\overline{BHE}$ = 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 accesses.

D_{15}-D_0 - word accesses.

- 8086 Aligned Memory Accesses

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, $\overline{BHE}$=1 & A_0 = 0.
- Byte of data written into or read from address X transferred over data lines D_7-D_0.

High bank byte access bus cycle:

- MPU applies odd address X+1 to both banks.
- MPU enables high bank only, $\overline{BHE}$=0 &A_0 =1.
- Byte-wide data transfer takes place over data line D_{15}- D_8.

Word access bus cycle:

- MPU applies even word address X to both banks.
- MPU enables both banks $\overline{BHE}$=0 &A_0 =0.
- Word-wide data transfer takes place over D_{15}- D_0.
- All accesses take a minimum of one bus cycle of duration.

@5MHz—800ns

@8MHz—500ns

Figure (14) shows the three types of aligned memory access of 8086 MPU.

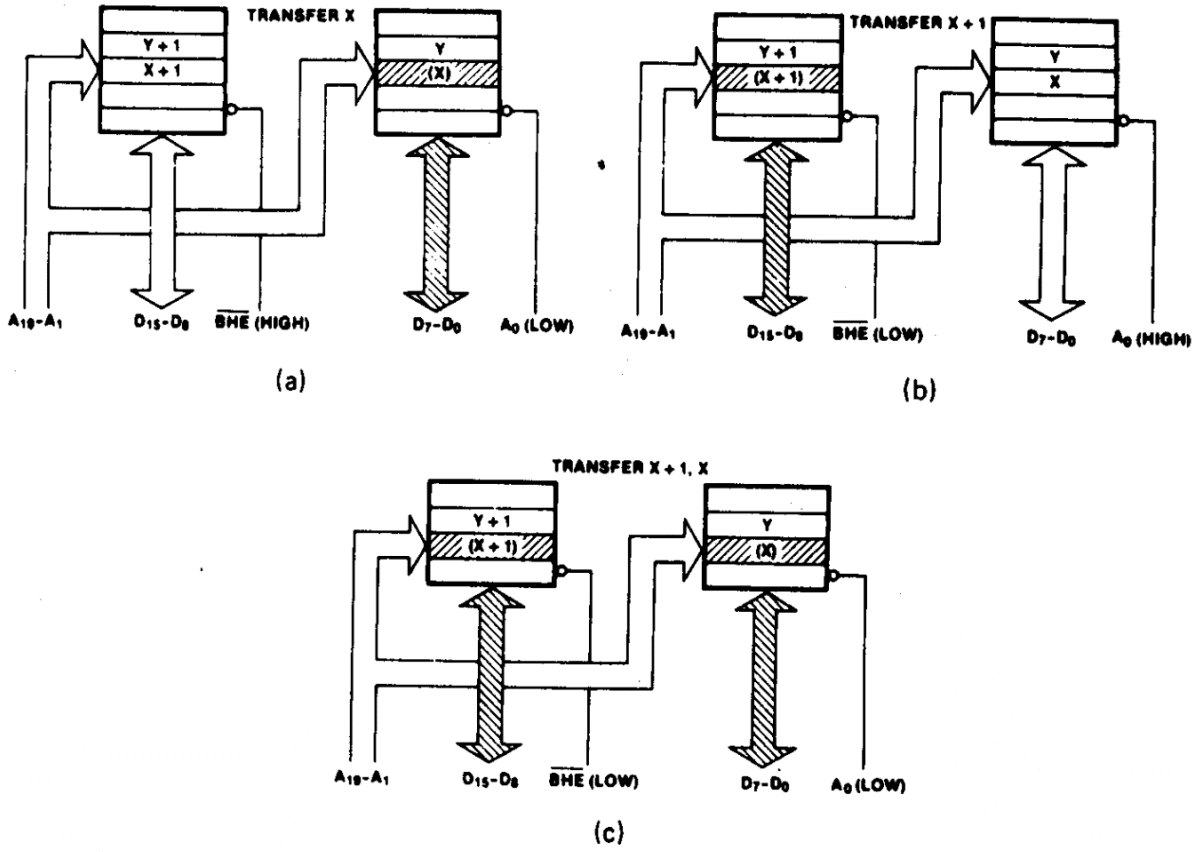


Fig.(14) – a) Low bank byte access, b) High bank byte access c) Word access.

- 8086 Misaligned Word Memory Access

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+1and $\overline{BHE}A_0$ =01 that is enabling high bank only.
- D_{15}-D_8 will carry the high byte of the word data.
- Access byte at address X+2 during cycle 2, A_{19}-A_0 = X+2 and $\overline{BHE}A_0$ =10 that is enabling low bank only.
- D_7-D_0 will carry the low byte of the word 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 misaligned data.

Figure (15) shows the misaligned word memory access of 8086 MPU.

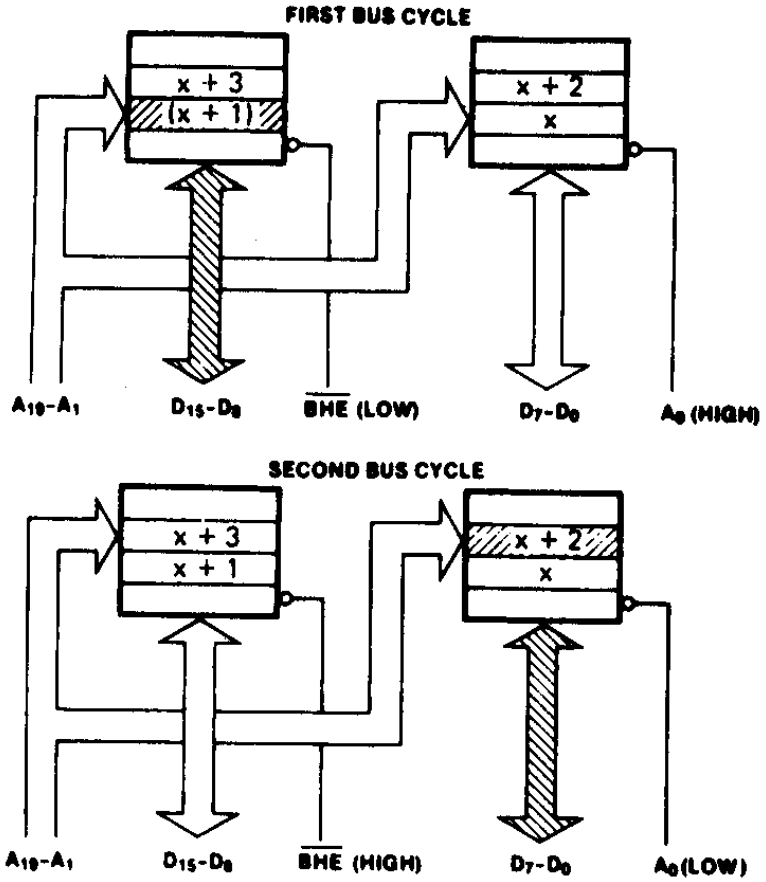


Fig.(15) A misaligned word memory access of 8086 MPU

Memory Control Signals– 8088 Minimum-Mode Interface

- Multiplexed-address data bus AD_0-AD_7 and A_8-A_{19}.

- Control signal review

  • ALE = pulse to logic 1 tells bus interface circuitry to latch address.

  • $\overline{RD}$ = logic 0 tells memory subsystem that a code or data read is in progress.

  • $\overline{WR}$= logic 0 tells memory subsystem that a data write is in progress.

  • IO/$\overline{M}$= Logic 0 tells interface circuits that the data transfer operation is for the memory subsystem.

  • DT/$\overline{R}$ = sets the direction of the external data bus for read (input) or write (output) operation.

  • $\overline{DEN}$= enables the interface between the memory subsystem and MPU data bus.

  • $\overline{SSO}$= tells memory interface whether the memory access is a code read or data access.

Figure (16) shows the memory control signals of 8088 minimum mode interface.

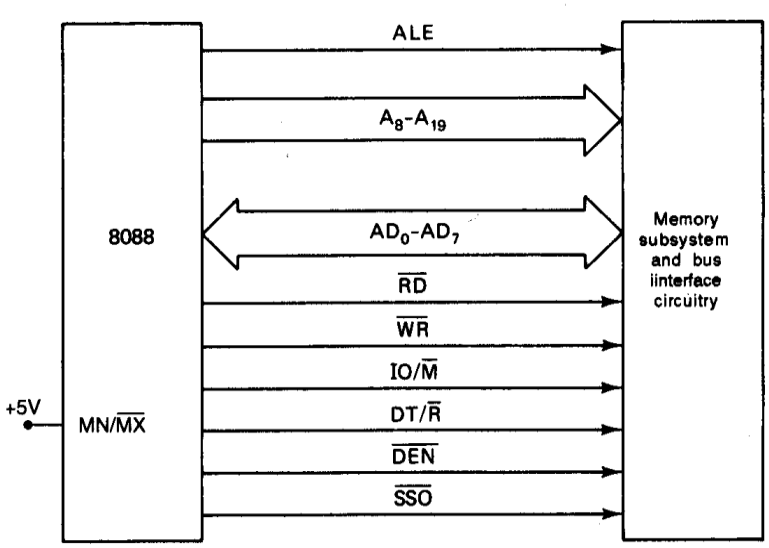


Fig.(16) The memory control signals of 8088 minimum mode interface.

Memory Control Signals– 8088 Maximum-Mode Interface

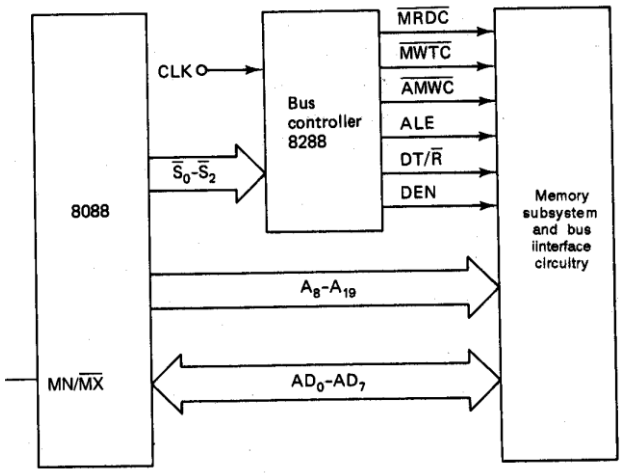
8288 bus controller produces the control signals:

- $\overline{MRDC}$ replaces $\overline{RD}$
- $\overline{MWTC}$ and $\overline{AMWC}$ replace $\overline{WR}$
- DEN is complement of $\overline{DEN}$
- IO/$\overline{M}$ no longer needed (bus controller creates separate memory and IO read/write controls)
- $\overline{SSO}$ no longer part of interface

Memory bus status code review:

- During all memory accesses one of three bus cycle status code are output by the MPU
- Instruction fetch
- Read memory
- Write memory
- 8288 decodes to produce appropriate control command signals
- $\overline{MRDC}$ for instruction fetch/memory read.
- $\overline{MWTC}$ for memory write.
- $\overline{AMWC}$ for advanced memory write.

Figure (17) shows the memory control signals and memory bus status codes of maximum mode 8088 microprocessor.



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

Fig.(17) The memory control signals and memory bus status codes of maximum mode 8088 MPU.

Read and Write Bus Cycles

1- 8088 Minimum Mode Read Bus Cycle

Figure (18) explains read bus cycle timing diagram and shows relationship between signals relative to times states.

- **T_1 state**—read cycle begins
  - Address output on A_0-A_{19}.
  - Pulse produced at ALE, and address should be latched in external circuitry on trailing edge of ALE.
  - IO/$\overline{M}$ set to 0, memory bus cycle.
  - DT/$\overline{R}$ set to 0 to set external data bus control circuitry for receive mode (read).
- **T_2 state**
  - Status code output on S_3-S_6.
  - AD_0 through AD_7 tri-stated in preparation for data bus operation.
  - $\overline{RD}$ set to 0, read cycle.
  - $\overline{DEN}$ set to 0 to enable external data bus control circuitry.
- **T_3 state**
  - Data on D_0-D_7 read by the MPU.
- **T4 state**—read cycle finishes
  - $\overline{RD}$ returns to 1, inactive level.
  - Complete address/data bus tri-stated.
  - IO/$\overline{M}$ returned to 1, IO bus cycle.
  - $\overline{DEN}$ returned to 1, inactive level.
  - DT/$\overline{R}$ returns to 1, transmit level.

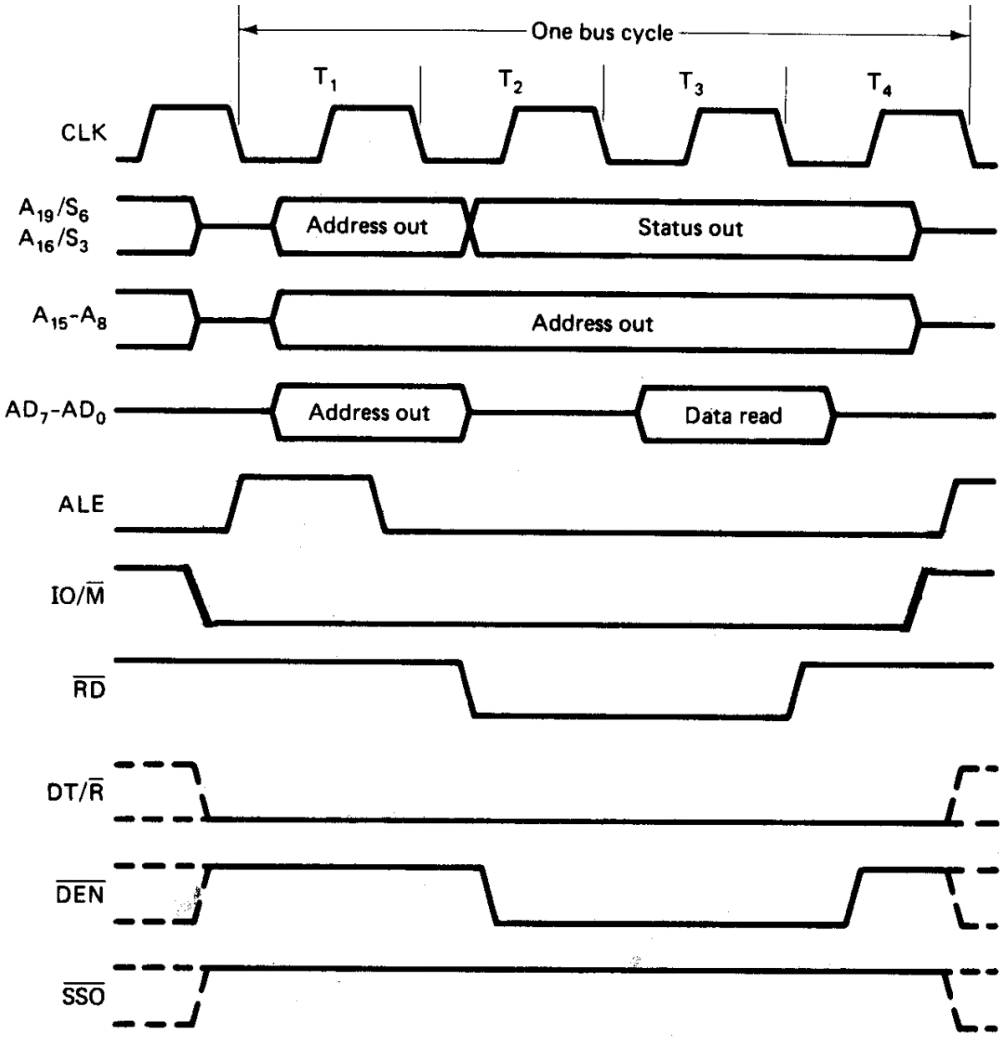


Fig. (18) 8088 minimum mode memory read bus cycle timing diagram.

2- 8086 Minimum Mode Read Bus Cycle

The differences of minimum mode 8086 read bus cycle from 8088 are:

• $\overline{BHE}$ is output along with the address in T_1.

• Data read by the MPU can be carried over all 16 data bus lines AD_{15}-AD_0.

• M/$\overline{IO}$ is replaced by IO/$\overline{M}$ which is switched to 1 instead of 0 at the beginning of T_1.

• $\overline{SSO}$ signal is not produce.

Fig (19) shows the waveform of 8086 minimum mode memory bus cycle timing diagram.

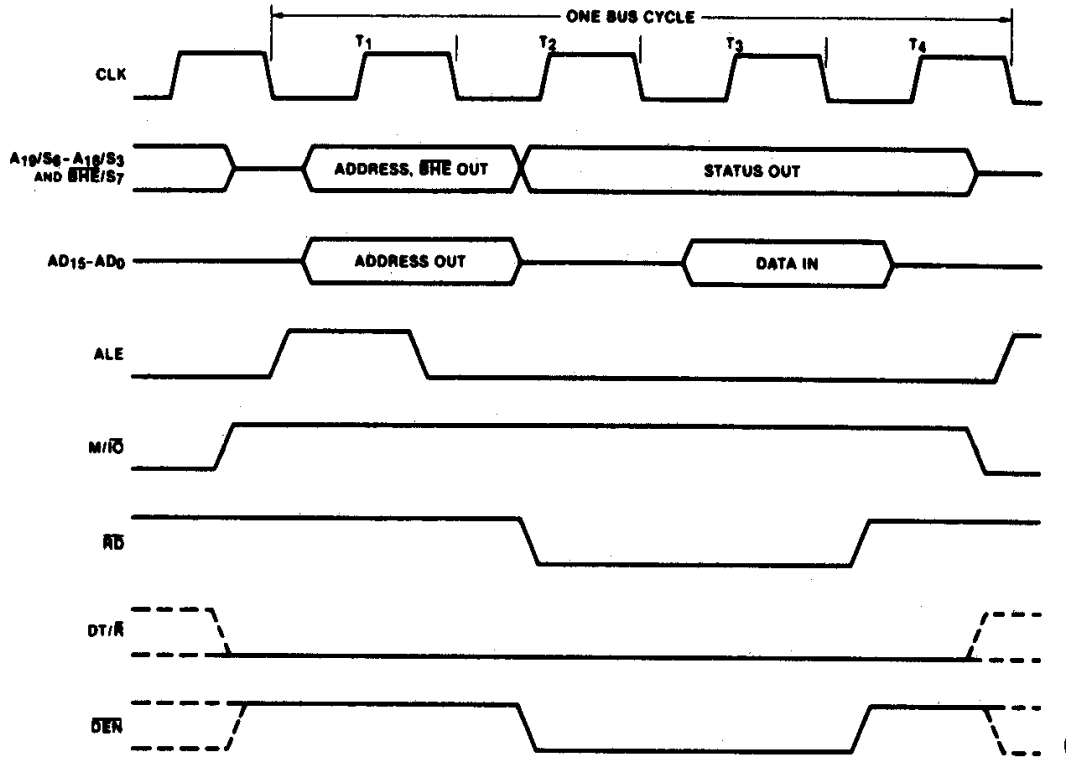


Fig. (19) 8086 minimum mode memory read cycle timing diagram.

3- 8088 Minimum Mode Write Bus Cycle

Fig. (20) explains write bus cycle timing diagram and shows relationship between signals relative to times states.

- T_1 state—write cycle begins.

- Address output on A_0-A_{19}.
- Pulse produced at ALE and address latched in external circuitry on trailing edge of ALE.
- IO/$\overline{M}$ set to 0, memory bus cycle.
- DT/$\overline{R}$ remains at 1 to set external data bus control circuitry for transmit mode (write).

- T_2 state

- Status code output on S_3-S_6.
- AD_0 through AD_7 transitioned to data bus and write data placed on bus.
- $\overline{DEN}$ set to 0 to enable external data bus control circuitry.
- $\overline{WR}$ set to 0, write cycle.

- T_3 or T_4 state

- Data on D_0-D_7 written into memory.

- T_4 state—write cycle finishes

- $\overline{WR}$ returns to 1, inactive level.
- Complete address/data bus tri-stated.
- IO/$\overline{M}$ returned to 1, IO bus cycle.

- $\overline{DEN}$ returned to 1, inactive level.

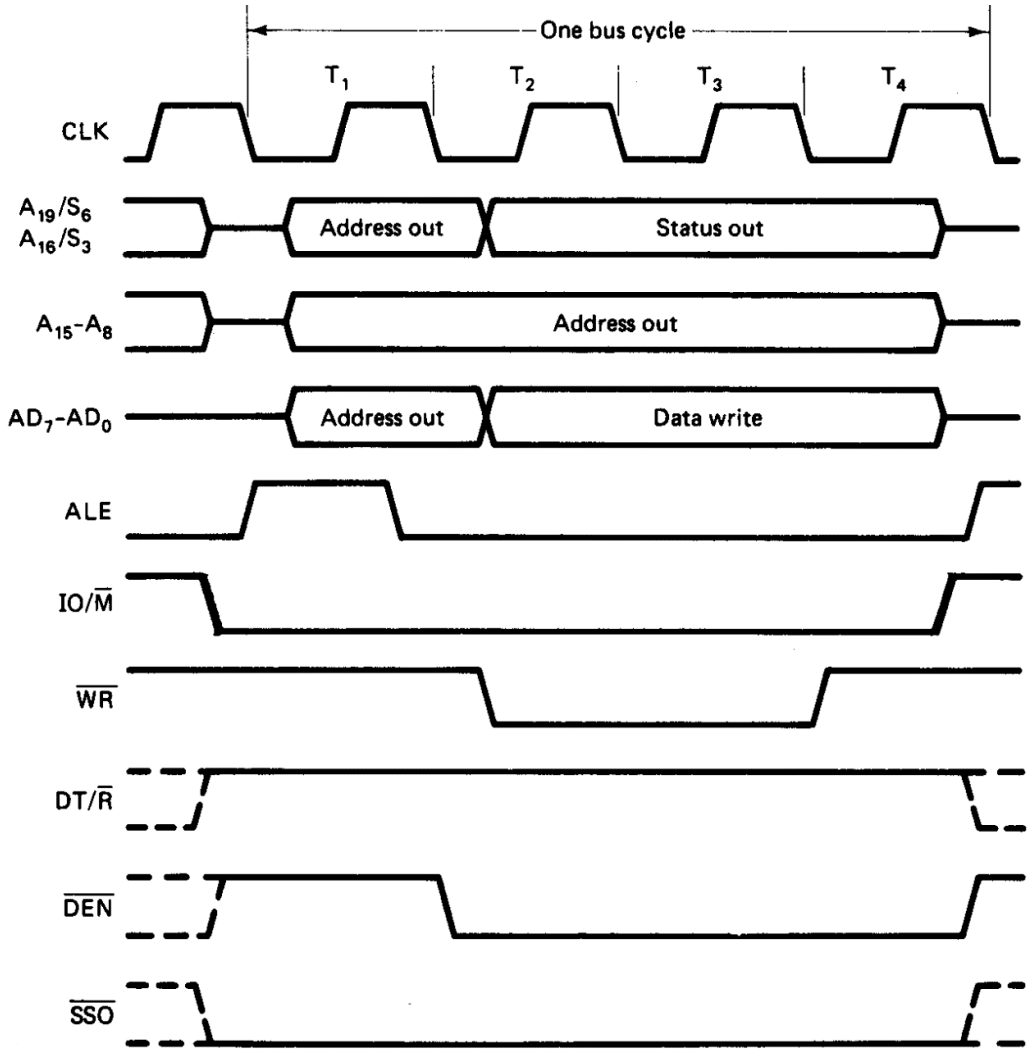


Fig. (20) 8088 minimum mode memory write cycle timing diagram.

4- 8086 Maximum Mode Write Bus Cycle

Similar to 8088/8086 minimum-mode write bus cycle.

- Address and data transfer operation identical.
- Transfer may be a high-byte, low-byte, or word.

The difference is that, 8288 produces the bus control signals ALE, DEN, $\overline{AMWC}$, and $\overline{MWTC}$.

- Bus status code $\overline{S}_2$-$\overline{S}_0$ output prior to T_1 and held through T_2.
- $\overline{AMWC}$ and $\overline{MWTC}$ replace $\overline{WR}$.
- DEN =1 produced instead of $\overline{DEN}$ =0.

Fig. (21) shows the waveform of 8086 maximum mode write cycle timing diagram.

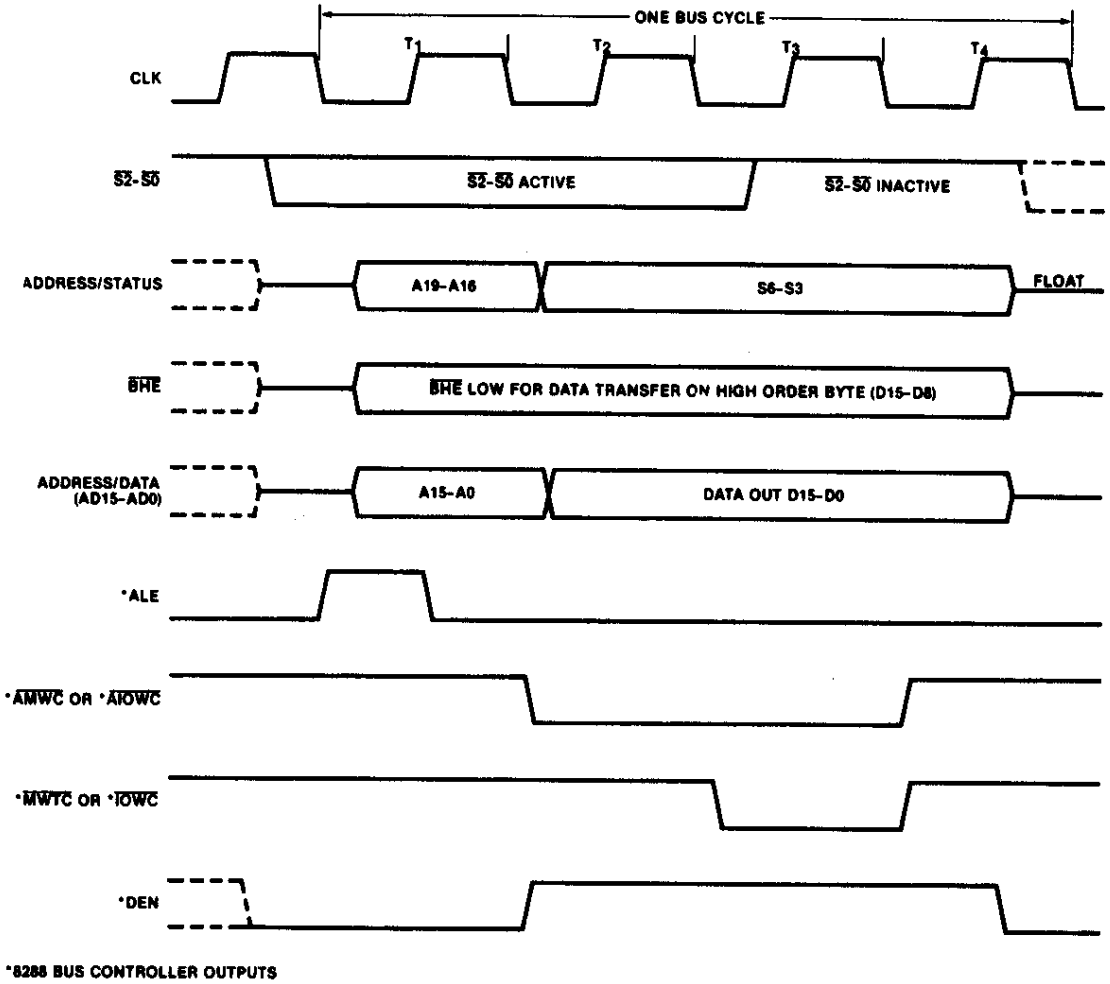


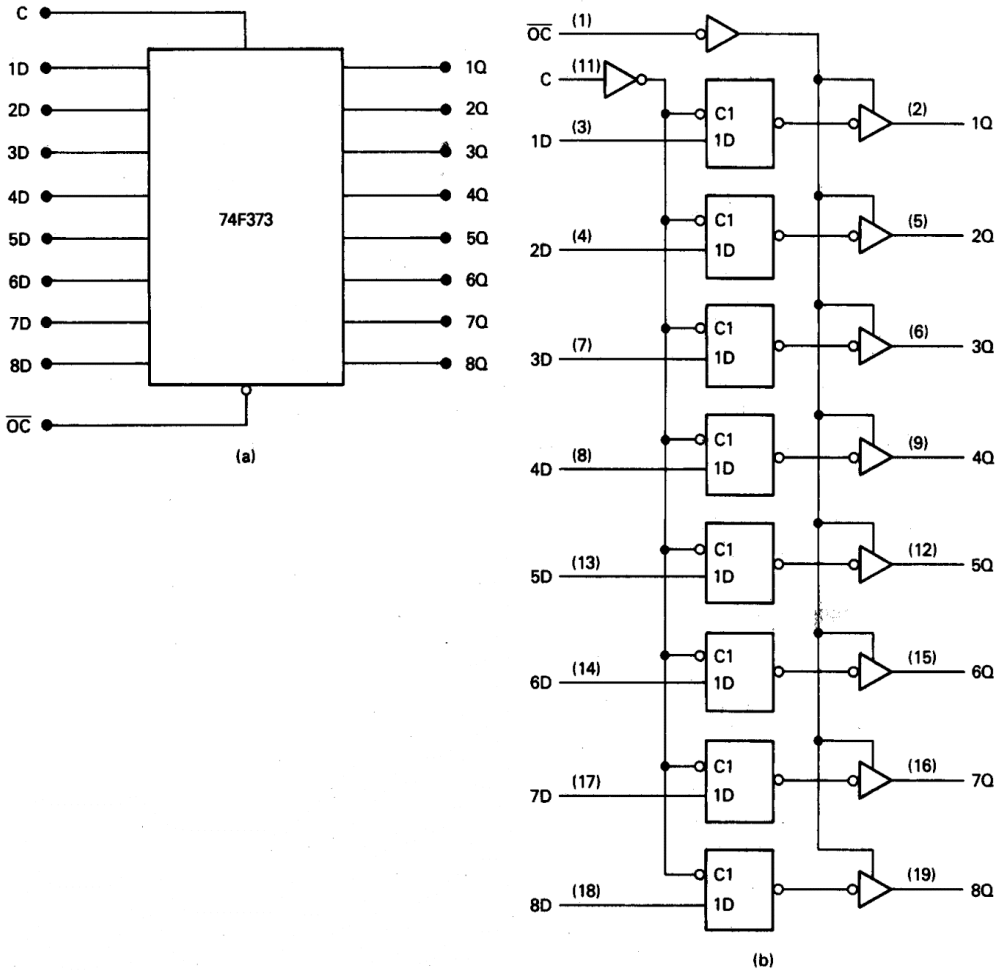
Fig. (21) 8086 maximum mode write cycle timing diagram.

Demultiplexing the Address/Data buses:

In 8088 microprocessor systems, a stable address is required and it must be available at the same time that data are to be transferred over bus. A Demultiplexing circuits used to provide a separated address and bus.

1. Address bus latches:

During bus cycles, a 20-bit address is output by MPU on AD0-A19 during T1 period of the bus cycle. Also, ALE is pulsed during T1 period. ALE is used to tell address bus lathe circuitry to latch the address (20-bit) on its output lines. Figure (22) shows the block diagram, the internal architecture, and the truth table of 74F37 - Octal D-type latch.



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

(c)

Fig.(22): (a)The block diagram, (b) the internal architecture, and (c)the truth table of 74F37 - Octal D-type latch.

Figure (23) illustrates the implementation of address bus latch with 3 74F373 Octal-D-type latches.

- Inputs AD0-AD7, A8-A19, from 8088 MPU.
- All devices permanently enabled by fixing the $\overline{OC}$ inputs at logic 0.
- All latches clocked in parallel with pulse at ALE from 8088 MPU.
- Latched and buffered outputs are: A0L-A19L.

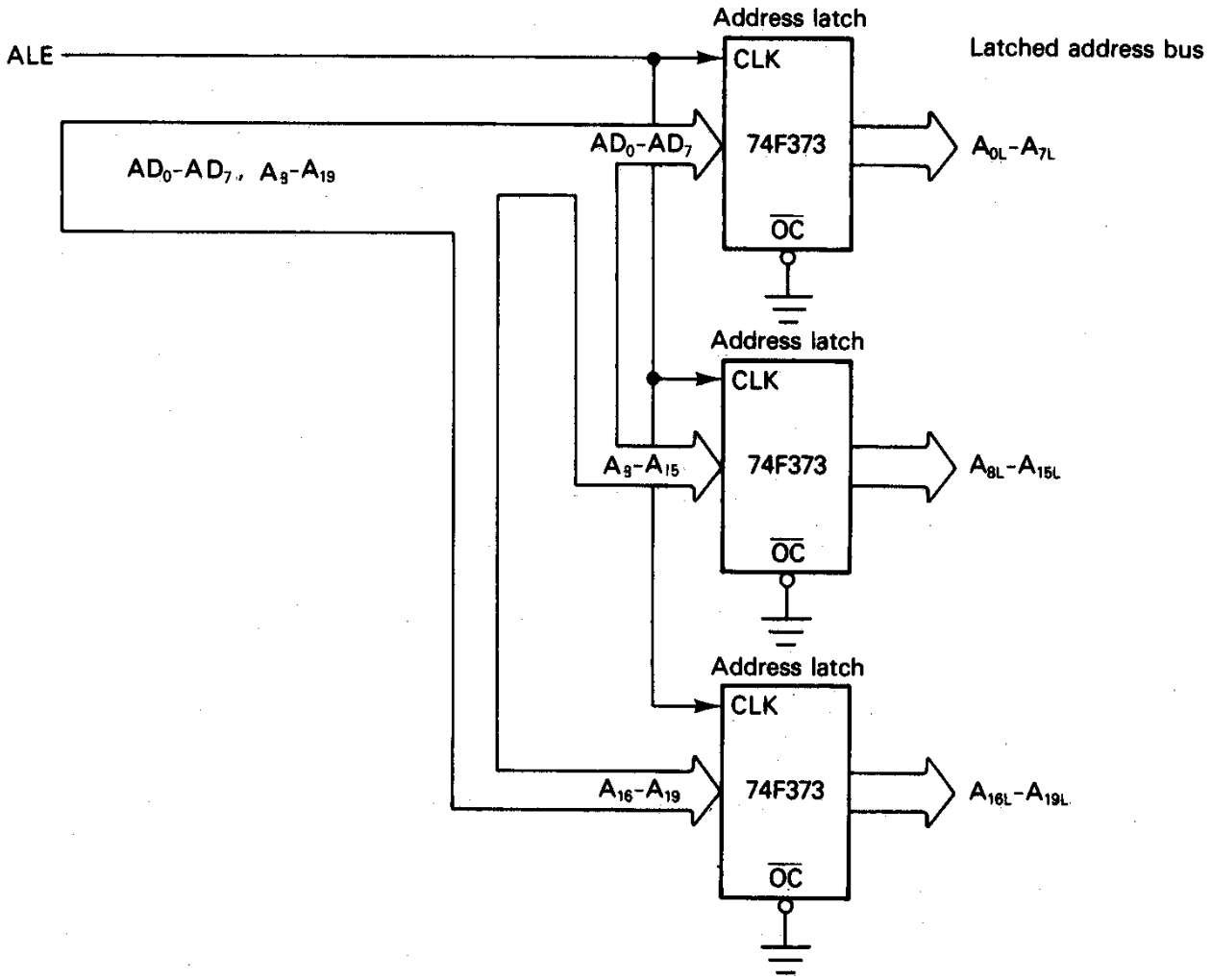
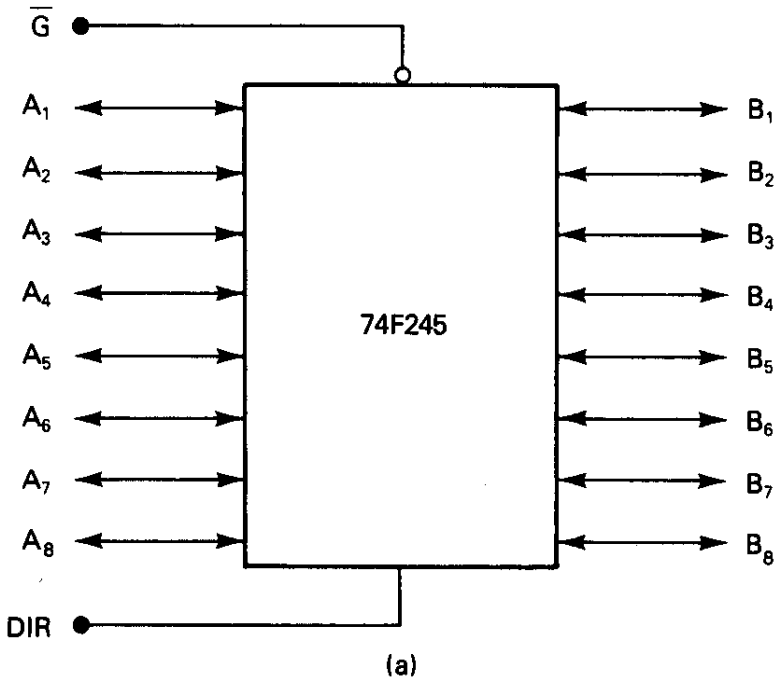


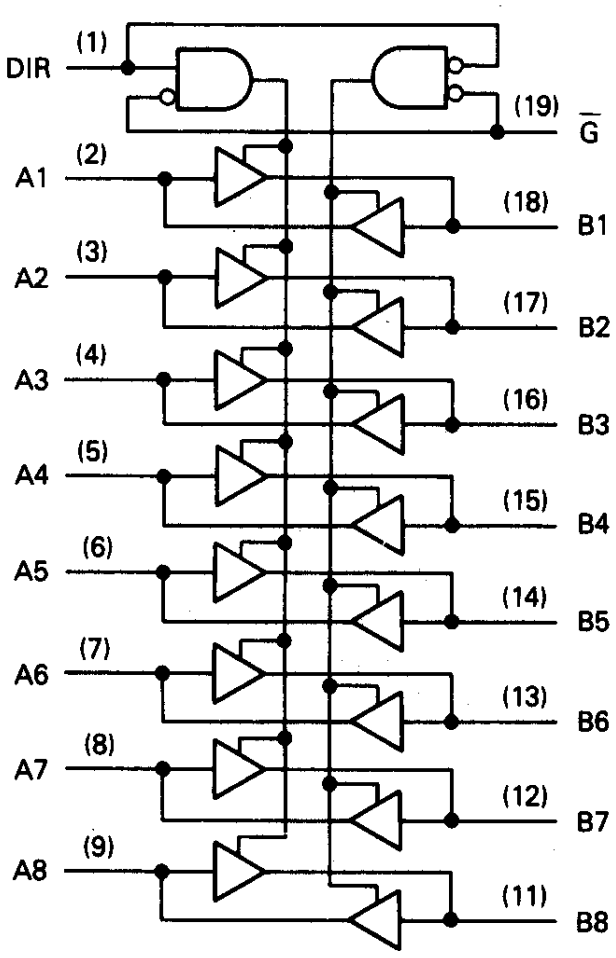
Fig.(23) Implementation of address bus latch with 3 74F373 Octal-D-type latches.

2- Data Bus Transceiver:

The data bus D0-D7 can be formed using bidirectional bus transceiver circuit 74F245. Fig.(24) sows the block diagram and internal architecture of 74F245.



(a)



(b)

Fig.(24): The block diagram and internal architecture of 74F245.

The 74F245 is an octal bi-directional bus transceiver, consists of 8 independent bus transceivers. The DIR input selects direction of data transfer as follows:

If DIR input is 0, the direction of data transfer is from B to A (read).

If DIR input is 1, the direction of data transfer is from A to B (write).

The G input should be 0 to enable all transceivers.

Fig.(25) shows the implementation of data bus transceiver of 8088 MPU. The **A** inputs/outputs are AD0-AD7 directly from MPU. The direction of the device set by logic level of DT/$\overline{R}$. The device enabled at appropriate time for data transfer by $\overline{DEN}$=0. The **B** inputs/outputs are the buffered data bus lines DB0-DB7. Buffered data bus lines applied directly to the memory or input/output subsystem.

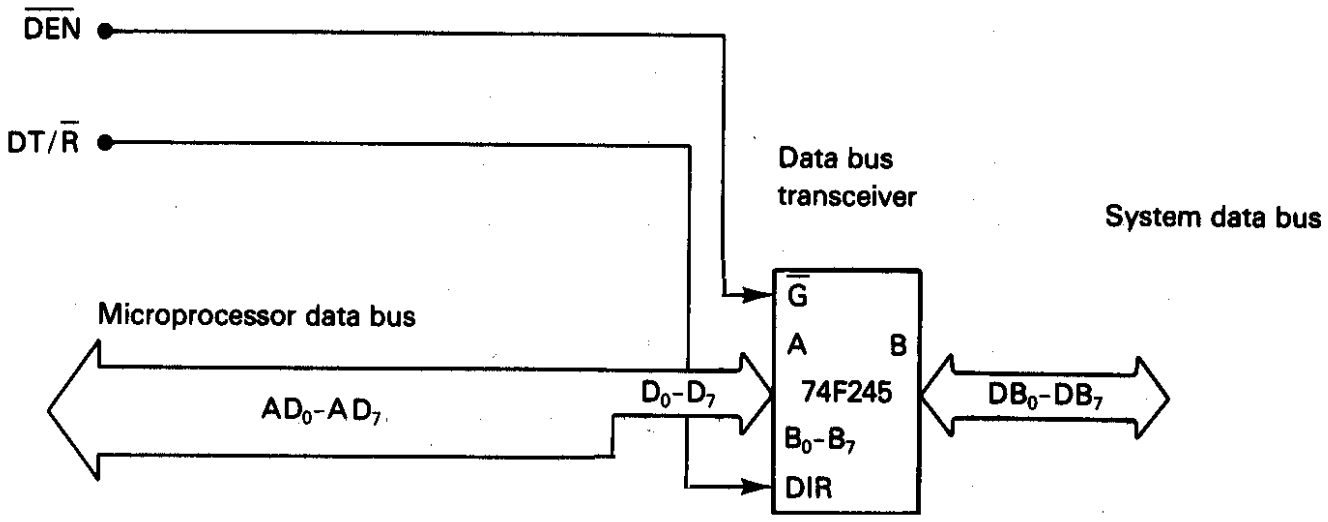
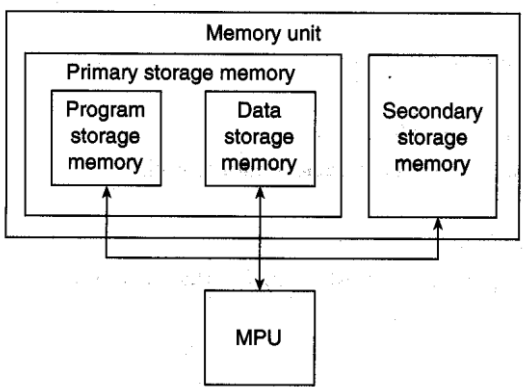


Fig.(25) The implementation of data bus transceiver of 8088 MPU using 74F245.

Memory Devices Circuits and Subsystem Design

1. Program and Data Storage Memory

- Memory provides the ability to store and retrieve digital information
    - Instructions of a program
    - Data to be processed
    - Results produced by processing
- Organization of the Microcomputer memory unit
    - **Secondary** storage—stores information that is not currently in use
        - Slow-speed
        - Very large storage capacity
        - Implemented with magnetic/optical storage devices—in PC
            - Hard disk drive
            - Floppy disk drive
            - Zip drive
    - **Primary** storage—stores programs and data that are currently active
        - High-speed
        - Smaller storage capacity
        - Implemented with semiconductor memory
- Partitioning of Primary Storage
    - **Program** storage memory—holds instructions of the program and constant information such as look-up tables
        - EPROM (BIOS in PC)
        - FLASH memory
        - DRAM (volatile code storage in a PC)
    - **Data** storage memory—holds data that frequently changes such as the information to be processed by a program
        - SRAM
        - DRAM (PC)

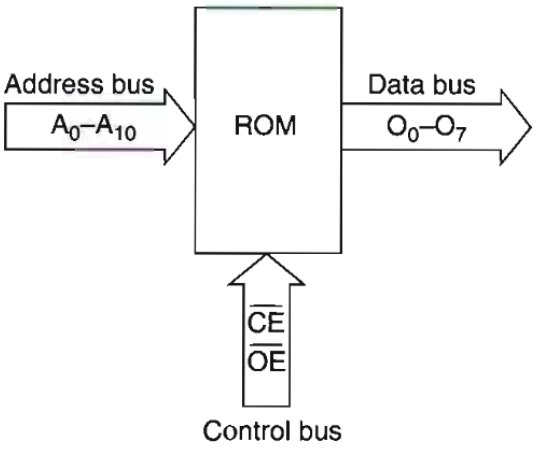


Read-Only Memory Types

Read-only memory (ROM) is:

- Used for storage of machine code of program
- Stored information can only be read by the MPU
- Information is non-volatile—not lost when power turned off
- Types:
    - ROM—mask-programmable read only memory
        - Programmed as part of manufacturing process
        - Lowest cost
        - High volume applications
    - PROM—one-time programmable read-only memory
        - Permanently programmed with a programming instrument
    - EPROM—erasable programmable read-only memory
        - Programmed like a PROM
        - Erasable by Ultraviolet light
- Electrically alterable ROM-like devices
    - FLASH memory
    - EEROM (E2ROM)

Read-Only Memory– Block Diagram:



Block diagram of the ROM, PROM, and EPROM are essentially the same.

- Signal interfaces
    - Address bus (A10-A0)—MPU inputs address information that selects the storage location to be accessed.
    - Data Bus (D7-D0)—information from the accessed storage location output to be read by MPU.
    - Control bus—enables device and/or enables output from device.
        - $\overline{CE}$ = chip enable—active 0; 1 low-power standby mode.
        - $\overline{OE}$ = output enable—active 0; 1 high-Z state.
- Byte capacity– number of bytes a device can store.
    - Calculated from number of address bits.
        EX: Address = 11-bit address, storage capacity $=2^{11}=$ 2048 bytes.
- Organization—how the size of a ROM is described.
    - Formed from capacity and data bus width.
        EX: 2048 X 8 or just 2K X 8
- Storage density—number of bits of storage in a ROM
    - Calculated from byte capacity and data width
        EX: Storage density = 2048 X 8 = 16384 bits (16K bits)

Example:

A ROM device has 15 address lines and 8 data lines. What are the address range, byte capacity, organization, and storage density?

Solution:

- Address range
    A14-A0 = $000\ 0000\ 0000\ 0000_2$ - $111\ 1111\ 1111\ 1111_2$ = 0000H - 7FFFH
- Byte capacity
    2^{15}= 32,768 bytes = 32K bytes
- Organization
    32768 X 8 bit
- Storage density
    32768 x 8 = 262144 bits = 256K bits

Standard EPROM ICs

EPROM	Density (bits)	Capacity (bytes)
2716	16K	2K × 8
2732	32K	4K × 8
27C64	64K	8K × 8
27C128	128K	16K × 8
27C256	256K	32K × 8
27C512	512K	64K × 8
27C010	1M	128K × 8
27C020	2M	256K × 8
27C040	4M	512K × 8

Expanding Byte Capacity

Many applications require more ROM capacity than is available in a single device.

- Need more bytes of storage
- Connects to a wider data bus

- Expanding byte capacity with 2 EPROMS
  - Connect address bus lines in parallel
  - Connect output lines in parallel
  - Connect $\overline{OE}$ in parallel
  - Enable chips with separate chip selects
    - Address bit A15 decoded to produce $\overline{CS0}$ and $\overline{CS1}$
    - A15=0 = $\overline{CS0}$
    - A15=1 = $\overline{CS1}$
    - Implemented with inverting buffer

- Byte capacity

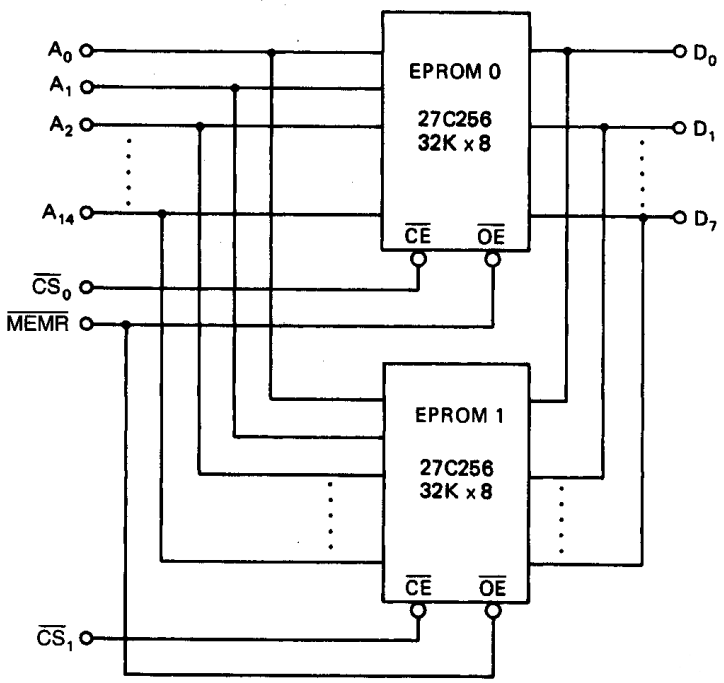
  2^{16} = 64K bytes

- Organization

  64K X 8 bit

- Storage density

  2 X 32K x 8 = 512K bit



Expanding Word Length

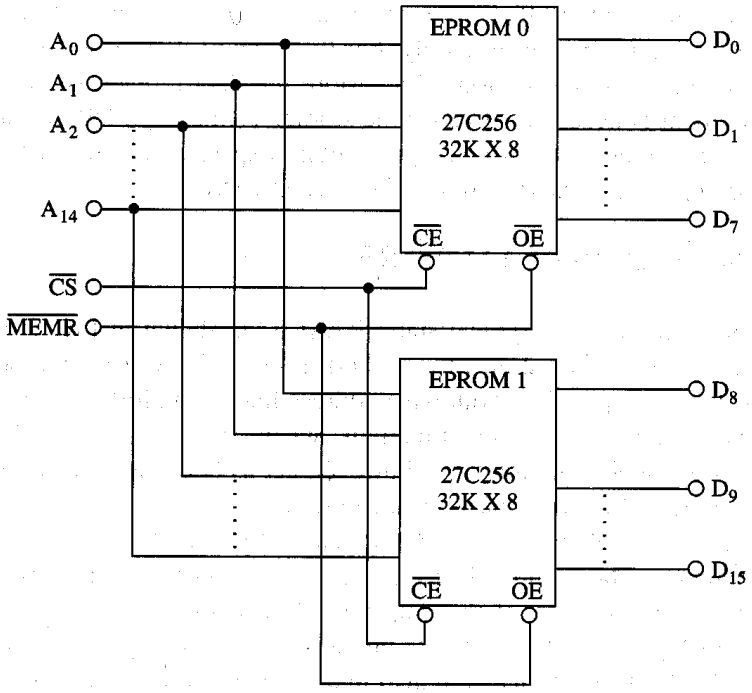
Expanding word length with 2 EPROM

- Connecting to 8086 16-bit data bus
  - Connect address bus lines in parallel
  - Connect $\overline{CE}$ in parallel
  - Connect $\overline{OE}$ in parallel
  - 8 data outputs of EPROM 0 used to supply the lower data bus lines D0-D7
  - 8 data outputs of EPROM 1 used to supply the upper 8 data bus lines D8-D15
- Byte capacity

  $2 \text{ X } 2^{15} = 64\text{K byte}$
- Organization

  32K X 16 bit
- Storage density

  32K x 16 = 512K bits



Random Access Read/Write Memories (RAM)

• Used for temporary storage of data and program information

• Stored information can be altered by MPU—read or written

- Information read from RAM
- Modified by processing
- Written back to RAM for reuse at a later time

• Information normally more frequently randomly accessed than ROM

• Information is volatile— lost when power turns off

• Types:

- Static RAM (SRAM) — data once entered remains valid as long as power supply is not turned off.
  - Lower densities
  - Higher cost
  - Higher speeds
- Dynamic RAM (DRAM)—data once entered requires both the power to be maintained and a periodic refresh.
  - Higher densities
  - Lower cost
  - Lower speeds
  - Refresh requires additional circuitry

SRAM Block Diagram

- Signal interfaces

• Address bus (A_{12}-A_0)—MPU inputs address information that selects the storage location to be accessed.

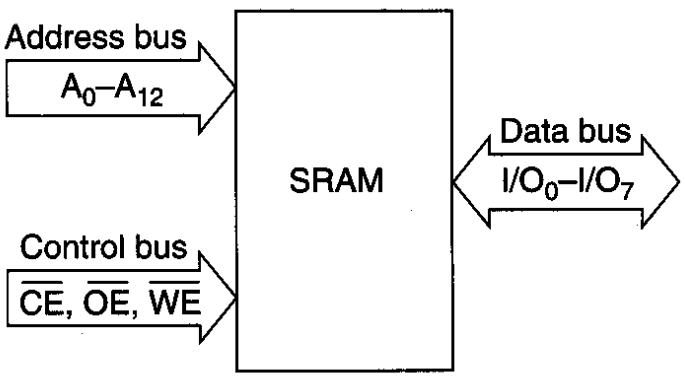
• Data Bus (I/O_7-I/O_0)—input/output of information for the accessed storage location from/to MPU.

• Control bus—enables device, enables output from device, and selects read/write operation.

- $\overline{CE}$ = chip enable—active 0
- $\overline{OE}$ = output enable—active 0
- $\overline{WE}$ = write enable

  0 = write to RAM

  1 = read from RAM



Standard SRAM ICs

Part numbers vary widely by manufacturer—Hitachi/NEC use "43xxx

SRAMs are available in a variety of densities and organization

- Typical SRAM densities
  - 64K bit
  - 256K bit
  - 1M bit
- Typical organizations of the 64K bit SRAM
  - 64K X 1 bit
  - 16K X 4 bit
  - 8K X 8 bit

SRAM	Density (bits)	Organization
4361	64K	64K × 1
4363	64K	16K × 4
4364	64K	8K × 8
43254	256K	64K × 4
43256A	256K	32K × 8
431000A	1M	128K × 8

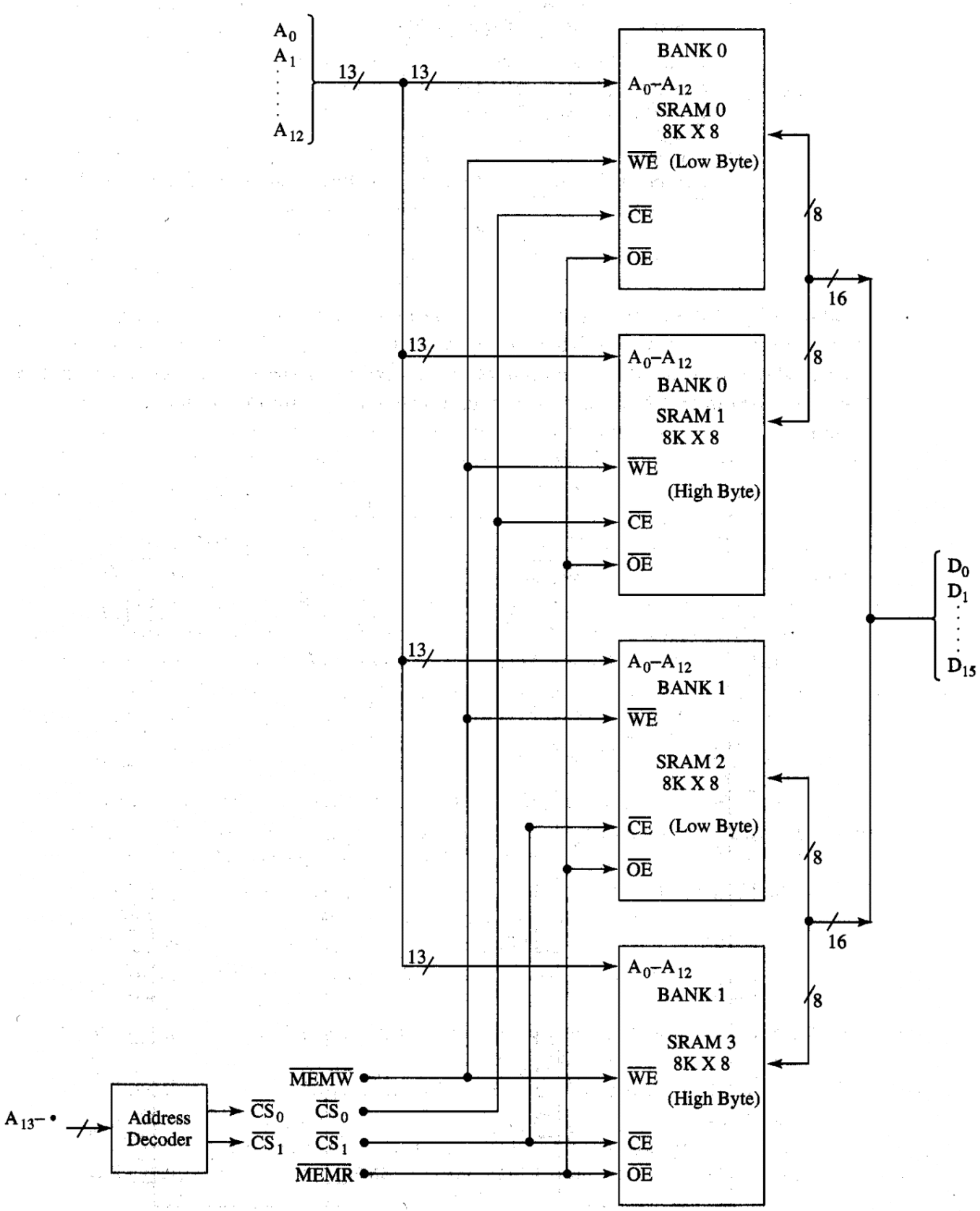
Expanding Word-Width and Capacity

Most SRAM subsystems:

- Require both word-width and bit capacity expansion.
- Require the ability to write on byte-wide or word wide basis– design only supports words
- Expansions performed in a similar way as for EPROMs
- 16K X 16-bit SRAM circuit
  - A0-A12 in parallel
  - A13 decoded to form $\overline{CS0}$ and $\overline{CS1}$
    - $\overline{CS0}$ to enable Bank 0
    - $\overline{CS1}$ to enable Bank 1
  - SRAMs 0 & 2—input/outputs connected in parallel and supply low byte of data bus.
  - SRAMs 1 & 3—input/ outputs connected in parallel and supply high byte of data bus.
  - $\overline{MEMW}$ and $\overline{MEMR}$ produces independent write and read enables.

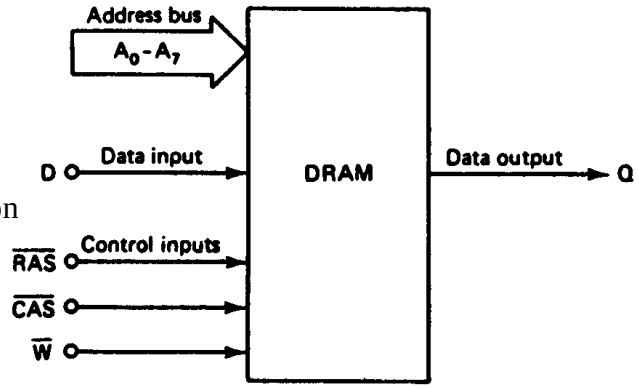
$\overline{MEMW}$	$\overline{MEMR}$	Data Transfer
0	0	Invalid
0	1	Word write
1	0	Word read
1	1	Inactive

Figure below show the memory expansion design.


A0
A1
A12
13
13
BANK 0
A0–A12
SRAM 0
8K X 8
WE (Low Byte)
CE
OE
8
16
13
A0–A12
BANK 0
SRAM 1
8K X 8
WE
(High Byte)
CE
OE
8
D0
D1
D15
13
A0–A12
BANK 1
WE
SRAM 2
8K X 8
CE (Low Byte)
OE
8
16
13
A0–A12
BANK 1
SRAM 3
8K X 8
WE
(High Byte)
CE
OE
8
MEMW
MEMR
A13–•
Address Decoder
CS0
CS1
CS0
CS1

DRAM Block Diagram

- DRAM signal interfaces
- Address multiplexed in external circuitry into a separate row and column address
  - Row address = A7-A0
  - Column address = A15-A8
- Special $\overline{RAS}$ and $\overline{CAS}$ inputs used to strobe address into DRAM
- Row and column addresses applied at different times to address inputs A0 through A7
  - Row address first
  - Column address second
  - Known as "RAS before CAS"
  - Address reassembled into 16-bit address inside DRAM
- Frequently data organizations are x1, x2, and x4
  - Separate data inputs and outputs
  - Data input labeled D
  - Data output labeled Q
- Read/write (W) input signals read or write operation



Standard DRAM ICs

DRAMs are available in a variety of densities and organization

• Typical DRAM densities
  • 64K bit
  • 256K bit
  • 1M bit, Etc.
  • Modern DRAMS as large as 1G bit

• Typical organizations of the 4M bit DRAM
  • 4M X 1 bit
  • 1M X 4 bit
  • Modern higher density devices also available in X8, X16, and X32organizations.

DRAM	Density (bits)	Organization
2164B	64K	64K × 1
21256	256K	256K × 1
21464	256K	64K × 4
421000	1M	1M × 1
424256	1M	256K × 4
44100	4M	4M × 1
44400	4M	1M × 4
44160	4M	256K × 16
416800	16M	8M × 2
416400	16M	4M × 4
416160	16M	1M × 16

Memory Circuit Design using DRAMS

Sixteen 64Kx1-bit DRAMs interconnected to form a 64K word memory subsystem—1M-bits of memory.

- Circuit connections:

• 8 multiplexed address inputs of all devices connected in parallel.

• RAS and CAS lines of all devices connected in parallel.

• Independent data lines arranged to form a 16-bit wide output bus.

• Independent input lines arranged to form a 16-bit wide input bus.

• W inputs of upper 8 DRAMs connected together and driven by $\overline{WR}0$.

• W inputs of lower 8 DRAMs connected together and driven by $\overline{WR}1$.

• Permits byte-wide or word-wide reads and writes.