UNIT-III

(ARCHITECTURE OF 8086 & INTERFACING)

Syllabus: Pin diagram of 8086-minimum mode and maximum mode of operation, Timing diagram, memory interfacing to 8086 (static RAM and EPROM). Need for DMA, DMA data transfer method, interfacing with 8237/8257.

INTRODUCTION

This unit explains how to design and implement an 8086 based microcomputer system. To design an 8086 based system, it is necessary to know how to interface the 8086 microprocessor with memory and input and output devices. Due to the mismatch in the speed between the microprocessor and other devices, a set of latches and buffers are required to interface the microprocessor with other devices. In this unit, you will learn about the way in which address/data buses, latches and buffers are used in the process of interfacing. To understand the interfacing principles and concepts it is necessary to learn the various types of bus cycles and bus timings. Overall, this unit makes you to understand how 8086 microprocessor is interfaced with memory and peripherals and how an 8086 based microcomputer system works.

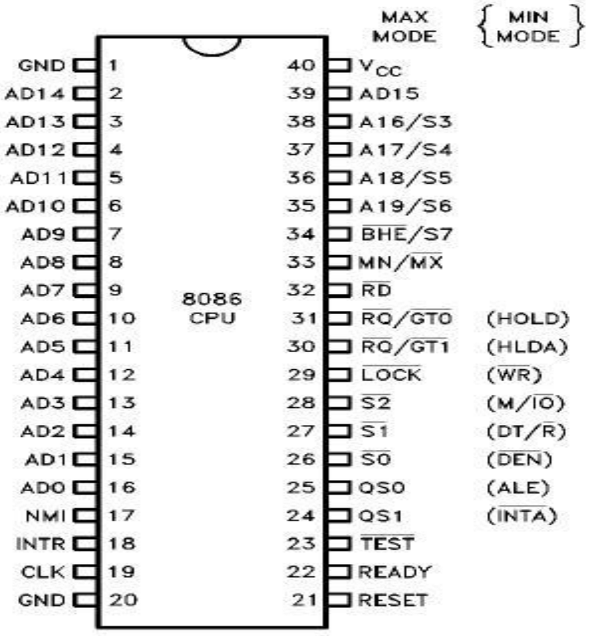
PIN DIAGRAM OF 8086 MICROPROCESSOR

The Microprocessor 8086 is a 16-bit MICROPROCESSOR available in different clock rates (5, 8, 10 MHz) and packaged in a 40 pin DIP or plastic package. The 8086 operates in single processor or multiprocessor configuration to achieve high performance. The pins serve a particular function in minimum mode (single processor mode) and other function in maximum mode configuration (multiprocessor mode).

The minimum mode is selected by applying logic 1 to the MN / $\overline{MX}$ input pin. This is a single microprocessor configuration.

The maximum mode is selected by applying logic 0 to the MN / $\overline{MX}$ input pin. This is a multi micro processors configuration.

The figure below shows the pins/signals of 8086 processor. Here the pins within the brackets (minimum mode pins) are minimum mode pins.



Signal description:

The 8086 signals can be categorized in three groups.

- The first are the signal having common functions in minimum as well as maximum mode.
- The second are the signals which have special functions for minimum mode.
- Third are the signals which have special functions for maximum mode.

✓ **The following signal descriptions are common for both modes:**

Vcc: It requires +5V single power supply for the operation of the internal circuit.

GND: ground for the internal circuit.

AD15-AD0: These are the time multiplexed memory I/O address and data lines. These lines serve two functions. The 16 data bus lines D0 through D15 are actually multiplexed with address lines A0 through A15 respectively. By multiplexed we mean that the bus work as an address bus during first machine cycle and as a data bus during next machine cycles. D15 is the MSB and D0 LSB. When acting as a data bus, they carry read/write data for memory, input/output data for I/O devices, and interrupt type codes from an interrupt controller.

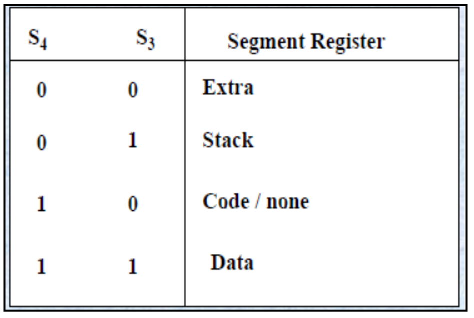
A19/S6, A18/S5, A17/S4, and A16/S3: These are the time multiplexed address and status lines. During T1 these are the most significant address lines for memory operations. During I/O operations, these lines are low. During memory or I/O operations, status information is available on those lines for T2, T3, Tw and T4.

The status of the interrupt enable flag bit is updated at the beginning of each clock cycle. The status is displayed on **S_5** pin.

The **S4** and **S3** combinedly indicate which segment register is presently being used for memory accesses as in below fig.

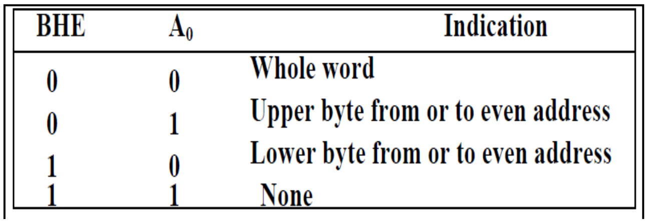
The last status bit **S6** is always at the logic 0 level.

The address bits are separated from the status bit using latches controlled by the ALE signal.



S_4	S_3	Segment Register
0	0	Extra
0	1	Stack
1	0	Code / none
1	1	Data

$\overline{BHE}$/S7: The bus high enable is used to indicate the transfer of data over the higher order (D15-D8) data bus as shown in table. It goes low for the data transfer over D15-D8 and is used to derive chip selects of odd address memory bank or peripherals. $\overline{BHE}$ is low during T1 for read, write and interrupt acknowledge cycles, whenever a byte is to be transferred on higher byte of data bus.



BHE	A_0	Indication
0	0	Whole word
0	1	Upper byte from or to even address
1	0	Lower byte from or to even address
1	1	None

$\overline{RD}$ **(Read):** This signal on low indicates the peripheral that the processor is performing s memory or I/O read operation. $\overline{RD}$ is active low and shows the state for T2, T3, and Tw of any read cycle. The signal remains tristated during the hold acknowledge.
READY: This is the acknowledgement from the slow device or memory that they have completed the data transfer. This signal is provided by an external clock generator device and can be supplied by the memory or I/O subsystem to signal the 8086 when they are ready to permit the data transfer to be completed.
NMI-Non maskable interrupt: This is an edge triggered input which causes a type2 interrupt. The NMI is not maskable internally by software. A transition from low to high initiates the interrupt response at the end of the current instruction.

INTR: 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 an interrupt request has been recognized by the 8086, it indicates this fact to external circuit with pulse to logic 0 at the $\overline{INTA}$ output.

$\overline{TEST}$**:** This input is examined by a 'WAIT' instruction. If the TEST pin goes low, execution will continue, else the processor remains in an idle state.

CLK: Clock Input: The clock input provides the basic timing for processor operation and bus control activity. Its an asymmetric square wave with 33% duty cycle.

RESET: This input causes the processor to terminate the current activity and start execution from FFFF0H.

MN/$\overline{MX}$: The logic level at this pin decides whether the processor is to operate in either minimum or maximum mode.

- ✓ **The following pin functions are for the minimum mode operation of 8086:**

M/$\overline{IO}$**:** This is a status line logically equivalent to S2 in maximum mode. The logic level of M/$\overline{IO}$ tells external circuitry whether a memory or I/O transfer is taking place over the bus. When it is low, it indicates the processor is having an I/O operation, and when it is high, it indicates that the processor is having a memory operation.

$\overline{WR}$**:** The signal write $\overline{WR}$ indicates that a write bus cycle is in progress. The 8086 switches $\overline{WR}$ to logic 0 to signal external device that valid write or output data are on the bus.

$\overline{INTA}$**: (Interrupt Acknowledge**)-This signal is used as a read strobe for interrupt acknowledge cycles. i.e. when it goes low, the processor has accepted the interrupt.

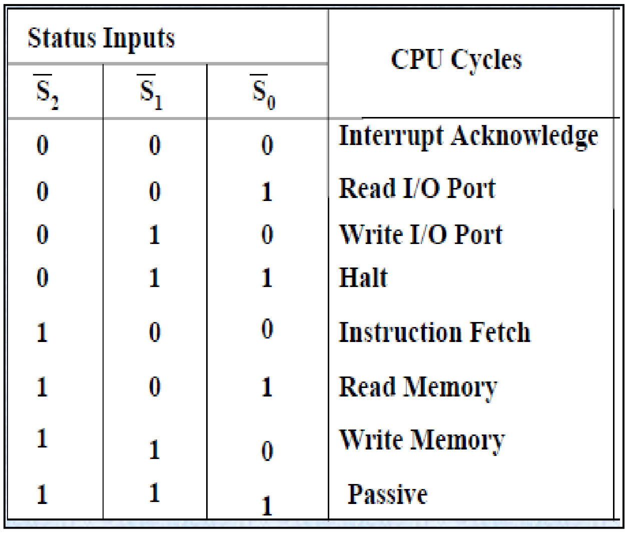
ALE – Address Latch Enable: This output signal indicates the availability of the valid address on the address/data lines, and is connected to latch enable input of latches. This signal is active high and is never tristated.

DT/$\overline{R}$ **– Data Transmit/Receive**: This output is used to decide the direction of data flow through the transreceivers (bidirectional buffers). When the processor sends out data, this signal is high and when the processor is receiving data, this signal is low.

$\overline{DEN}$ **– Data Enable**: This signal indicates the availability of valid data over the address/data lines. It is used to enable the transreceivers (bidirectional buffers) to separate the data from the multiplexed address/data signal.

HOLD and HLDA: The direct memory access DMA interface of the 8086 minimum mode consist of the HOLD and HLDA signals. When the HOLD line goes high, it indicates to the processor that another master is requesting the bus access. The processor, after receiving the HOLD request, issues the hold acknowledge signal on HLDA pin, in the middle of the next clock cycle after completing the current bus cycle.

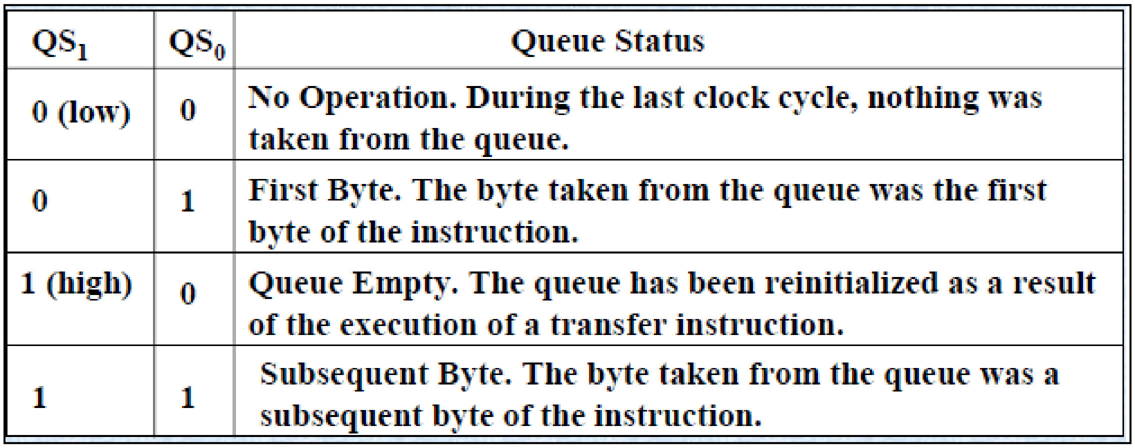
The following pin functions are applicable for maximum mode operation of 8086:
$\overline{S2}$,$\overline{S1}$**, and** $\overline{S0}$ **– Status Lines**: These are the status lines which reflect the type of operation, being carried out by the processor.



Status Inputs			CPU Cycles
$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	
0	0	0	Interrupt Acknowledge
0	0	1	Read I/O Port
0	1	0	Write I/O Port
0	1	1	Halt
1	0	0	Instruction Fetch
1	0	1	Read Memory
1	1	0	Write Memory
1	1	1	Passive

LOCK: This output pin indicates that other system bus master will be prevented from gaining the system bus, while the LOCK signal is low.

QS1, QS0 – Queue Status: These lines give information about the status of the code-prefetch queue. Two new signals that are produced by the 8086 in the maximum-mode system are queue status outputs QS0 and QS1. Together they form a 2-bit queue status code, QS1QS0. Following table shows the four different queue status.



QS_1	QS_0	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.

RQ0/0GT, RQ1/GT1 – Request/Grant: These pins are used by the other local bus master in maximum mode, to force the processor to release the local bus at the end of the processor current bus cycle. Each of the pin is bidirectional with RQ/GT0 having higher priority than RQ/GT1.

OPERATING MODES OF 8086

There are two modes of operation for Intel 8086 namely the minimum mode and the maximum mode. When only one 8086 microprocessor is to be used in a micro computer system the 8086 is used in the minimum mode of operation. In this mode the microprocessor issues the control signals required by memory and I/O devices. In a multi processor system it operates in the maximum mode. In case of maximum mode of operation control signals are issued by Intel 8288 bus controller which is used with 8086 for this purpose. The level of the pin $MN/\overline{MX}$ decides the operating mode of 8086. When $MN/\overline{MX}$ is high the microprocessor operates in a minimum mode. When it is low the microprocessor operates in the maximum mode. From pin 24 to 31 issue two different sets of signals. One set of signals is issued when the microprocessor is operating in the minimum mode. The other sort of signal is issued when the microprocessor is operating in the maximum mode. Thus the pins from 24-31 have alternate functions.

TIMINGS:

Timing plays a crucial role, not only in sports like cricket_but also in digital electronic equipments like microprocessors. Timing and Timing diagram plays a vital role in microprocessors. The timing diagram is the diagram which provides information about the various conditions of signals such as high/low, when a machine cycle is being executed. Without the knowledge of timing diagram it is not possible to match the peripheral devices to the microprocessors. These peripheral devices includes memories, ports etc. Such devices can only be matched with microprocessors with the help of timing diagram.

Before dealing with timing diagram, we have to make ourselves familiar with certain terms.

Machine cycle: A basic microprocessor operation such as reading a byte from memory or writing a byte to a port is called a machine cycle (Bus cycle). A machine cycle consists of at least 4 clock cycles/ clock states (T-states) for accessing the data.

Instruction cycle: The time a microprocessor requires to fetch and execute an entire instruction is referred to as an instruction cycle. An instruction cycle consists of one or more machine cycles.

T-state: T-state is nothing but one subdivision of the operation performed in one clock period. These subdivisions are internal state of the microprocessor synchronized with system clock.

So, an instruction cycle is made up of machine cycles, and a machine cycle is made up of states. The time for the state is determined by the frequency of the clock signal.

GENERAL BUS OPERATION CYCLES:

The 8086 has a combined address and data bus commonly referred as a time multiplexed address and data bus. The main reason behind multiplexing address and data over the same pins is the maximum utilization of processor pins and it facilitates the use of 40 pin standard DIP package. The bus can be demultiplexed using a few latches and transreceivers, when ever required. Basically, all the processor bus cycles consist of at least four clock cycles. These are referred to as T1, T2, T3, and T4. The address is transmitted by the processor during T1. It is present on the bus only for one cycle.

The above figs shows the signal activities on the 8086 microcomputer buses during simple read and write operations. The first line look at is the clock waveform, CLK, at the top. This represent s the crystal controlled clock signal sent to the 8086 from an external clock generator device such as the 8284. One clock cycle of this clock is called a state. The time interval labeled T1 in the figure is an example of a state. Different versions of the 8086 have maximum clock frequencies of between 5 MHz and 10 MHz, so the minimum time for one sate will be between 100 and 200ns, depending on the part use and the crystal used.

The negative edge of this ALE pulse is used to separate the address and the data or status information. In maximum mode, the status lines $\overline{S0}$, $\overline{S1}$ and $\overline{S2}$ are used to indicate the type of operation. Status bits S3 to S7 are multiplexed with higher order address bits and the $\overline{BHE}$ signal. Address is valid during T1 while status bits S3 to S7 are valid during T2 through T4.

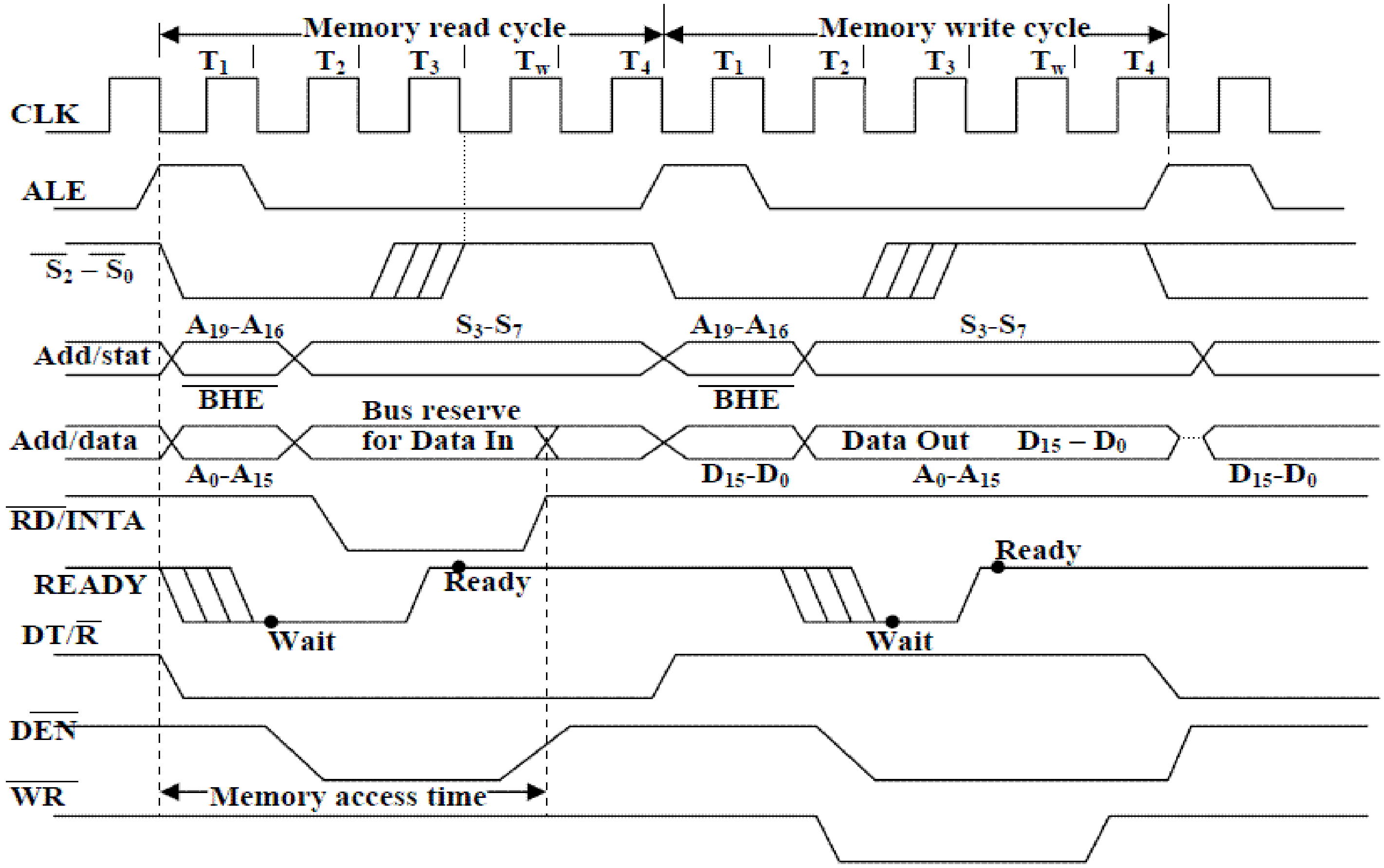


Fig. General bus operation timing diagram

8086 Bus activities during a Read machine cycle:

Fig above (left half portion) shows the timing diagram of 8086 read machine cycle with WAIT state. The clock (CLK) signal is obtained from the clock-generator 8284. Each cycle of the clock is referred to as a state. Minimum number of states to access a data is four. They are T1, T2, T3, and T4 states.

During T1 state of a read machine cycle an 8086 first asserts the M/$\overline{\mathrm{IO}}$ signal. It will assert this signal high if it is going to read from memory during memory read cycle and it will assert M/$\overline{\mathrm{IO}}$ low if it is going to do a read from an Input port during its read cycle. The timing diagram in fig shows two lines for the M/$\overline{\mathrm{IO}}$signal, because the signal may be going LOW or going HIGH for a read cycle. The point where the two lines cross indicate the time at which the signal becomes valid for this machine cycle.

After asserting M/$\overline{\mathrm{IO}}$, the 8086 sends out a high on the address latch enable signal, ALE. The microprocessor sends out on AD0-AD15, A16 through A19 and $\overline{\mathrm{BHE}}$lines, and the address of the memory location that it wants to read. Since the latches are enabled by ALE being high, this address information passes through the latches to their outputs. The 8086 then makes the ALE output low. This disables the latches (8282) and holds the address information latched on the latch outputs. The address information latched on the latch outputs can now be used to select the desired memory or port location.

In the timing diagram, the first point at which the two ($AD_0 – AD_{15}$) cross represents the time at which the 8086 has put a valid address on these lines. Two lines DO NOT indicate that all 16 lines are going high or going low at this point. The crossed lines indicate the time at which a valid address is on the bus.

Since the address information is now held on the latch, the 8086 does not need to send it out any more. As shown in fig. 12 the 8086 floats the AD0 - AD15 lines so that they can be used to input data from memory or from a port. At about the same time the 8086 also remove the $\overline{\mathrm{BHE}}$and A16-A19 information from the upper lines and sends out some status information on these lines.

The 8086 is now ready to read data from the addressed memory locations or port. During T2-state the 8086 asserts its $\overline{\mathrm{RD}}$ signal low. This signal is used to enable the addressed memory device or port device.

At the end of T3 state the microprocessor makes the $\overline{\mathrm{RD}}$signal high and reads the data available on the data bus, provided the READY input signal is high. It is the duty of the external circuit to see that valid data is made available on the data bus.

If the READY input pin is not high at the sampled time in a machine cycle, the 8086 will insert one or more WAIT states between T3 and T4 states in that machine cycle. An external hardware device is set up to pulse READY low before the rising edge of the clock in T2 state. After the 8086 finishes T3 of the machine cycle, it enters a WAIT state.

If the READY input is still low at the end of a WAIT state, then the 8086 will insert another WAIT state. The 8086 will continue inserting WAIT states until the READY input is sampled high again. If the READY input is sampled high again during T3 or during the WAIT state, the microprocessor comes out of the WAIT state and will initiate T4 of the machine cycle.

The $\overline{\mathrm{DEN}}$ signal is used to enable bi-directional buffers on the data bus. The data enable signal, $\overline{\mathrm{DEN}}$, from the 8086 will enable the data buffer when it is asserted LOW. The data transmit / receive signal DT/$\overline{\mathrm{R}}$ from the 8086 is used to specify the direction in which the buffers are enabled. When DT/$\overline{\mathrm{R}}$ is asserted high, the buffers will, if enabled by $\overline{\mathrm{DEN}}$, transmit data from the 8086 to Memory or I/O ports. When DT/$\overline{\mathrm{R}}$ is asserted low, the buffers, if enabled by $\overline{\mathrm{DEN}}$, will allow data to be received from Memory or I/O ports of the 8086. DT/$\overline{R}$ is asserted during T1 of the machine cycle. The $\overline{\mathrm{DEN}}$ is asserted after the 8086 finishes using the data bus to send the lower 16 address bits

8086 Bus activities during write machine cycle:

The 8086 write operation is very similar to the read cycle. During T1 of a write machine cycle the 8086 asserts M/$\overline{\mathrm{IO}}$ low if the write is going to a port and it asserts M/$\overline{\mathrm{IO}}$ high if the write is going to memory. At about the same time the 8086 raises ALE high to enable the address latches. The 8086 then assert $\overline{\mathrm{BHE}}$ and on the lines AD0 - AD19, it output the address that it will be writing to. When writing to a port, line A16 - A19 will always be low, because the 8086 only sends out 16-bits port addresses. The 8086 brings ALE low again to latch the address on the outputs of the latches. In addition to holding the address, the latches also function as buffers for the address lines. After the address information is latched, the 8086 remove the address information from AD0 - AD15 and outputs the desired data on these lines.

If the READY input is sampled LOW by the 8086 before or during T2 of the machine cycle, the 8086 will insert a WAIT state after T3. If the READY input is sampled high before the end of the WAIT state, the 8086 will go on with state T4 as soon as it completes the WAIT state. The 8086 will continue to insect wait states for as long as the READY is sampled low just before the end of each WAIT state.

MINIMUM MODE 8086 SYSTEM AND TIMINGS

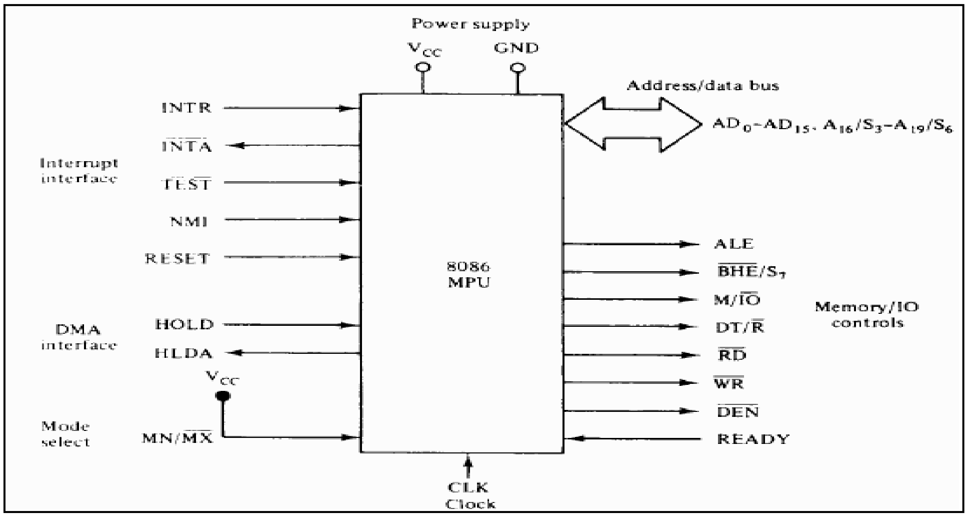


Fig.1: Pin diagram of minimum mode 8086 system

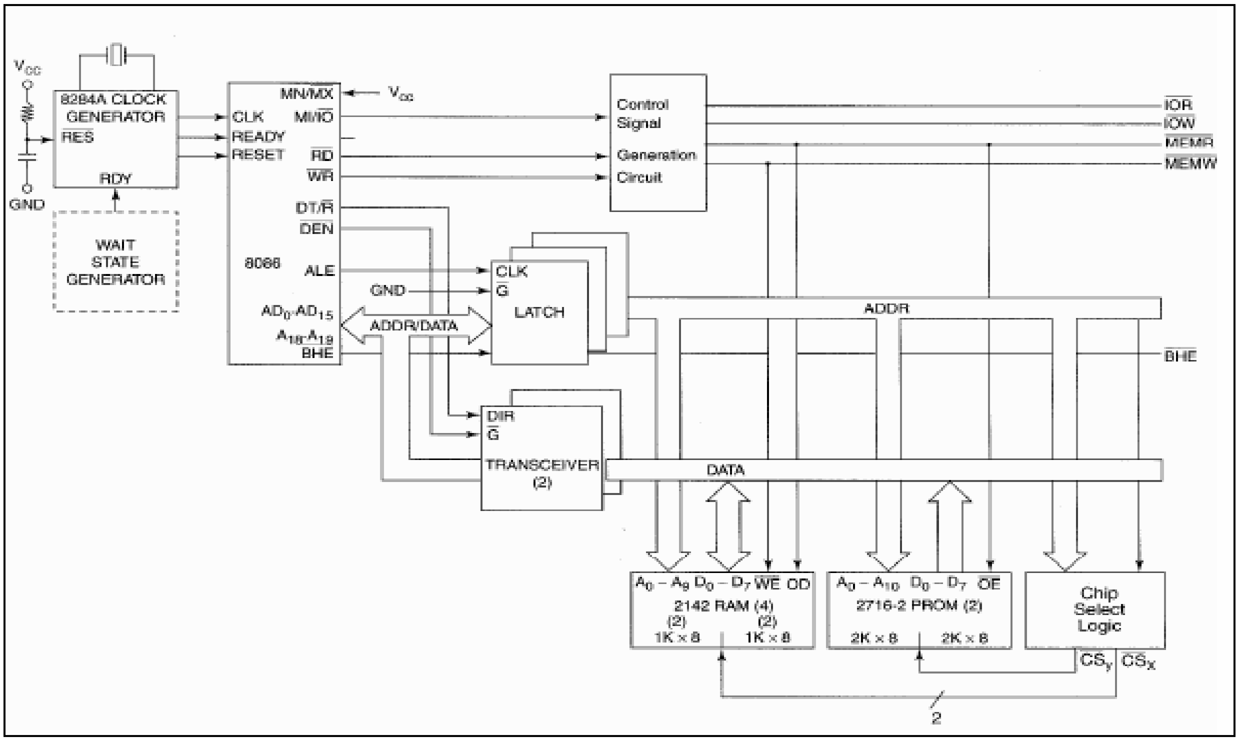


Fig.2: Minimum mode 8086 system configuration

•In a minimum mode 8086 system, the microprocessor 8086 is operated in minimum mode by strapping its MN/$\overline{MX}$ pin to logic 1. In this mode, all the control signals are given out by the microprocessor chip itself. There is a single microprocessor in the minimum mode system. The remaining components in the system are latches, transreceivers, clock generator, memory and I/O devices. Some type of chip selection logic may be required for selecting memory or I/O devices, depending upon the address map of the system.

•Latches are generally buffered output D-type flip-flops like 74LS373 or 8282. They are used for separating the valid address from the multiplexed address/data signals and are controlled by the ALE signal generated by 8086.

•Transreceivers are the bidirectional buffers and sometimes they are called as data amplifiers. They are required to separate the valid data from the time multiplexed address/data signals. They are controlled by two signals namely, $\overline{DEN}$ and DT/$\overline{R}$. The DT/$\overline{R}$ signal indicates the direction of data, i.e. from or to the processor. The system contains memory for the monitor and users program storage.

•Usually, EPROM is used for monitor storage, while RAM for user's program storage. A system may contain I/O devices.

•The working of the minimum mode configuration system can be better described in terms of the timing diagrams rather than qualitatively describing the operations.

•The opcode fetch and read cycles are similar. Hence the timing diagram can be categorized in two parts, the first is the timing diagram for **read cycle** and the second is the timing diagram for **write cycle.**

Read Cycle:

The read cycle begins in T1 with the assertion of address latch enable (ALE) signal and also M / $\overline{IO}$ signal. During the negative going edge of this signal, the valid address is latched on the local bus.

•The $\overline{BHE}$ and A0 signals address low, high or both bytes. From T1 to T4 , the M/$\overline{IO}$ signal indicates a memory or I/O operation.

•At T2, the address is removed from the local bus and is sent to the output. The bus is then tristated. The read ($\overline{RD}$) control signal is also activated in T2.
•The read ($\overline{RD}$) signal causes the address device to enable its data bus drivers. After $\overline{RD}$ goes low, the valid data is available on the data bus.
•The addressed device will drive the READY line high. When the processor returns the read signal to high level, the addressed device will again tristate its bus drivers.

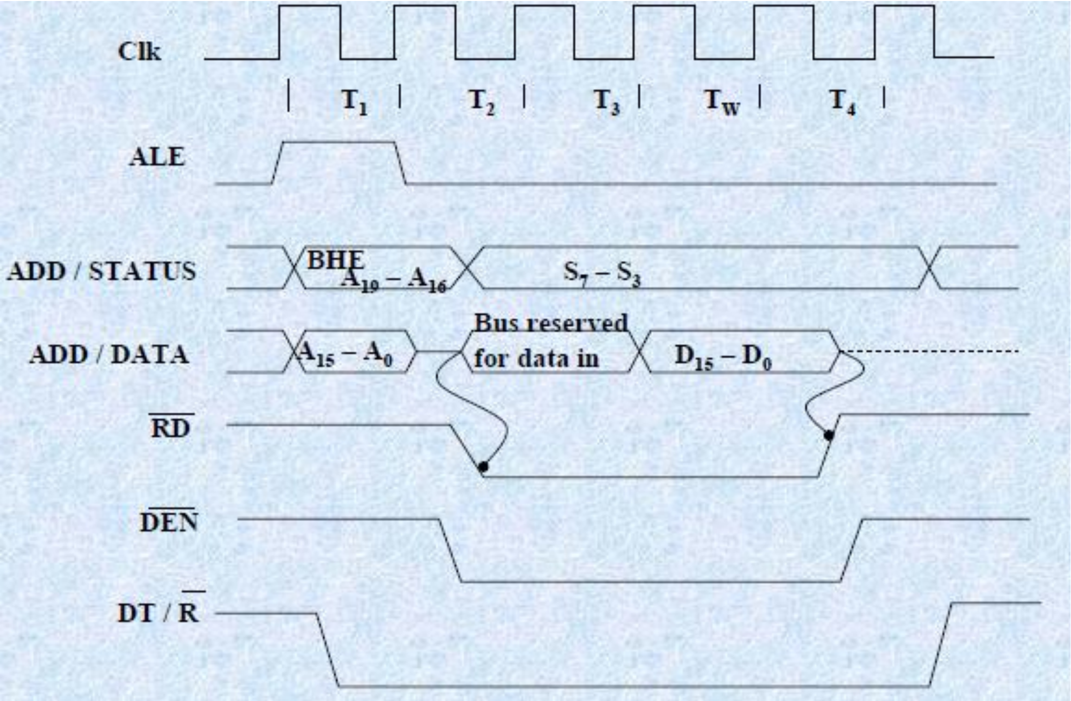


Fig. Read cycle timing diagram for minimum mode operation

Write Cycle:

•A write cycle also begins with the assertion of ALE and the emission of the address. The M/$\overline{IO}$ signal is again asserted to indicate a memory or I/O operation. In T2, after sending the address in T1, the processor sends the data to be written to the addressed location.
•The data remains on the bus until middle of T4 state. The $\overline{WR}$ becomes active at the beginning of T2 (unlike $\overline{RD}$ is somewhat delayed in T2 to provide time for floating).
•The $\overline{BHE}$ and A0 signals are used to select the proper byte or bytes of memory or I/O word to be read or write.

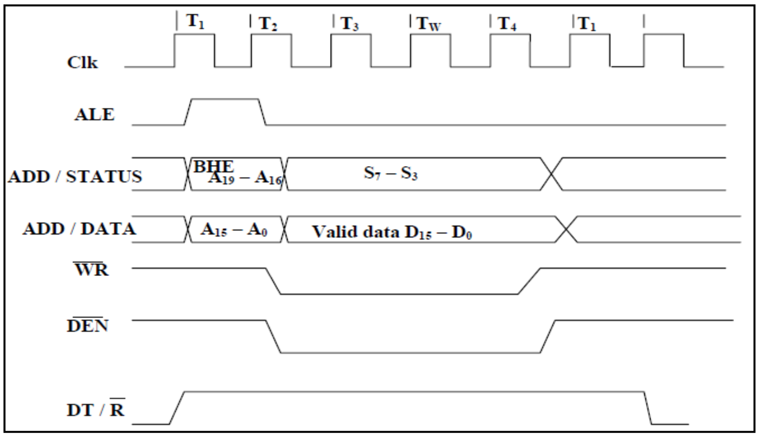


Fig. Write cycle timing diagram for minimum mode operation

The M/$\overline{IO}$, $\overline{RD}$ and $\overline{WR}$ signals indicate the type of data transfer as specified in table below.

M / $\overline{IO}$	$\overline{RD}$	$\overline{WR}$	Transfer Type
0	0	1	I / O read
0	1	0	I/O write
1	0	1	Memory read
1	1	0	Memory write

MAXIMUM MODE 8086 SYSTEM AND TIMINGS

When the 8086 is set for the maximum-mode configuration, it provides signals for implementing a multiprocessor / coprocessor system environment. By multiprocessor environment we mean that one microprocessor exists in the system and that each processor is executing its own program. Usually in this type of system environment, there are some system resources that are common to all processors. They are called as global resources. There are also other resources that are assigned to specific processors. These are known as local or private resources. Coprocessor also means that there is a second processor in the system. In this two processor does not access the bus at the same time. One passes the control of the system bus to the other and then may suspend its operation. In the maximum-mode 8086 system, facilities are provided for implementing allocation of global resources and passing bus control to other microprocessor or coprocessor.

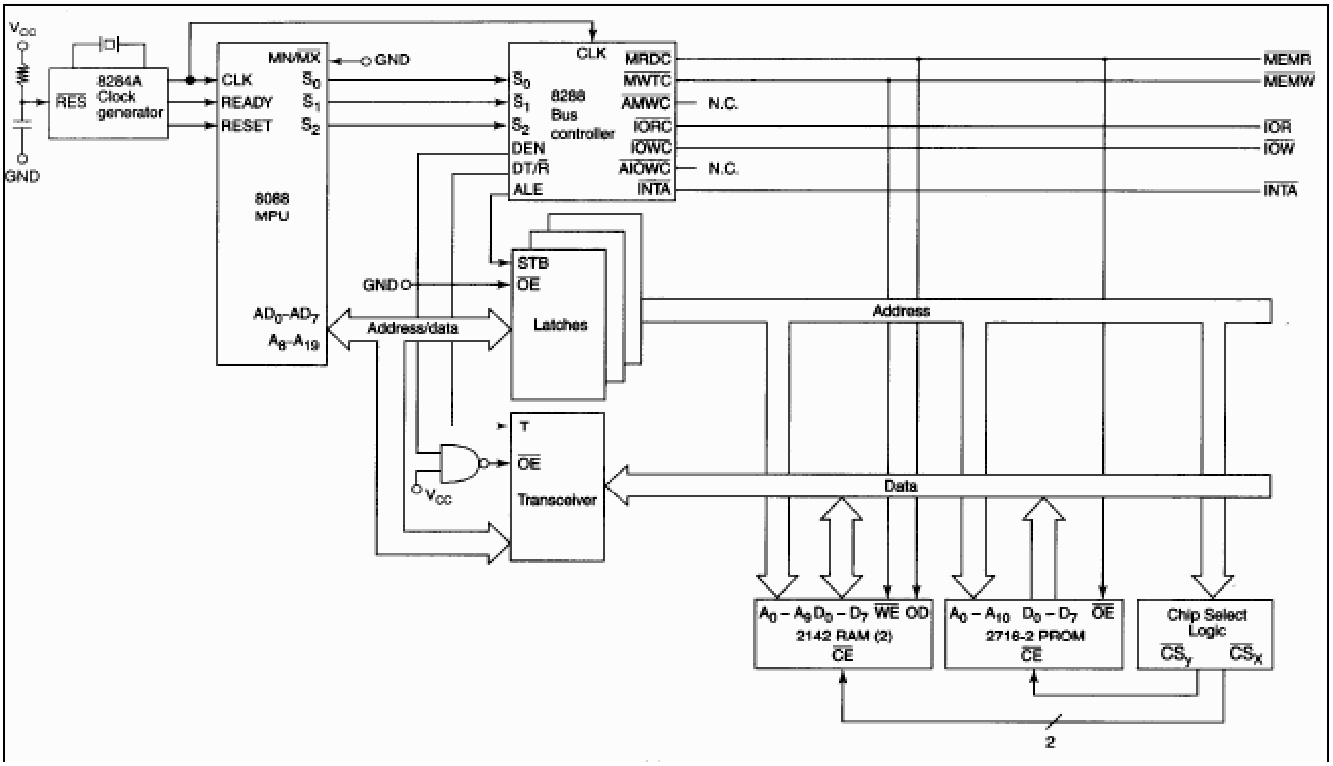


Fig: Maximum mode 8086 system configuration

•In the maximum mode, the 8086 is operated by strapping the MN/$\overline{MX}$ pin to ground. In this mode, the processor derives the status signal $\overline{S2}$, $\overline{S1}$, $\overline{S0}$. Another chip called bus controller derives the control signal using this status information. In the maximum mode, there may be more than one microprocessor in the system configuration. The components in the system are same as in the minimum mode system.

•The basic function of the bus controller chip IC8288 is to derive control signals like $\overline{RD}$ and $\overline{WR}$ (for memory and I/O devices), $\overline{DEN}$, DT/$\overline{R}$, ALE etc. using the information by the processor on the status lines.

•The bus controller chip has input lines $\overline{S2}$, $\overline{S1}$, $\overline{S0}$ and CLK. These inputs to 8288 are driven by MICROPROCESSOR.

•It derives the outputs ALE, $\overline{DEN}$, DT/$\overline{R}$, $\overline{MRDC}$, $\overline{MWTC}$, $\overline{AMWC}$, $\overline{IORC}$, $\overline{IOWC}$ and $\overline{AIOWC}$. The AEN, IOB and CEN pins are specially useful for multiprocessor systems.

•AEN and IOB are generally grounded. CEN pin is usually tied to +5V. The significance of the MCE/PDEN output depends upon the status of the IOB pin.

•If IOB is grounded, it acts as master cascade enable to control cascade 8259A, else it acts as peripheral data enable used in the multiple bus configurations.

•$\overline{INTA}$ pin used to issue two interrupt acknowledge pulses to the interrupt controller or to an interrupting device.

•$\overline{IORC}$, $\overline{IOWC}$ are I/O read command and I/O write command signals respectively. These signals enable an IO interface to read or write the data from or to the address port.

•The $\overline{MRDC}$, $\overline{MWTC}$ are memory read command and memory write command signals respectively and may be used as memory read or write signals.

•All these command signals instructs the memory to accept or send data from or to the bus.

•For both of these write command signals, the advanced signals namely $\overline{AIOWC}$ and $\overline{AMWTC}$ are available.

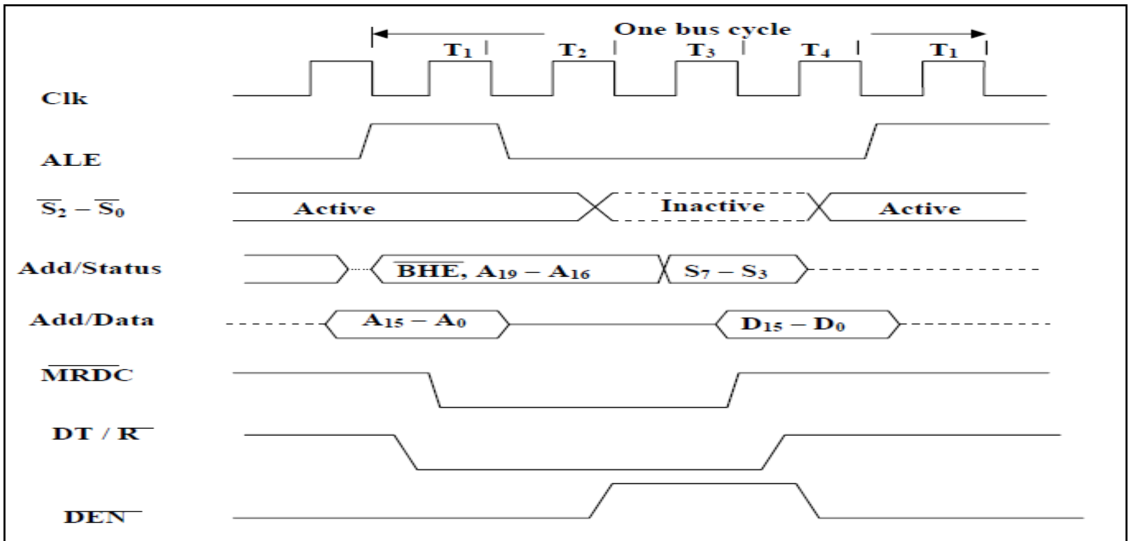
•Here the only difference between in timing diagram between minimum mode and maximum mode is the status signals used and the available control and advanced command signals.

•R0, S1, S2 are set at the beginning of bus cycle.8288 bus controller will output a pulse as on the ALE and apply a required signal to its DT / R pin during T1.

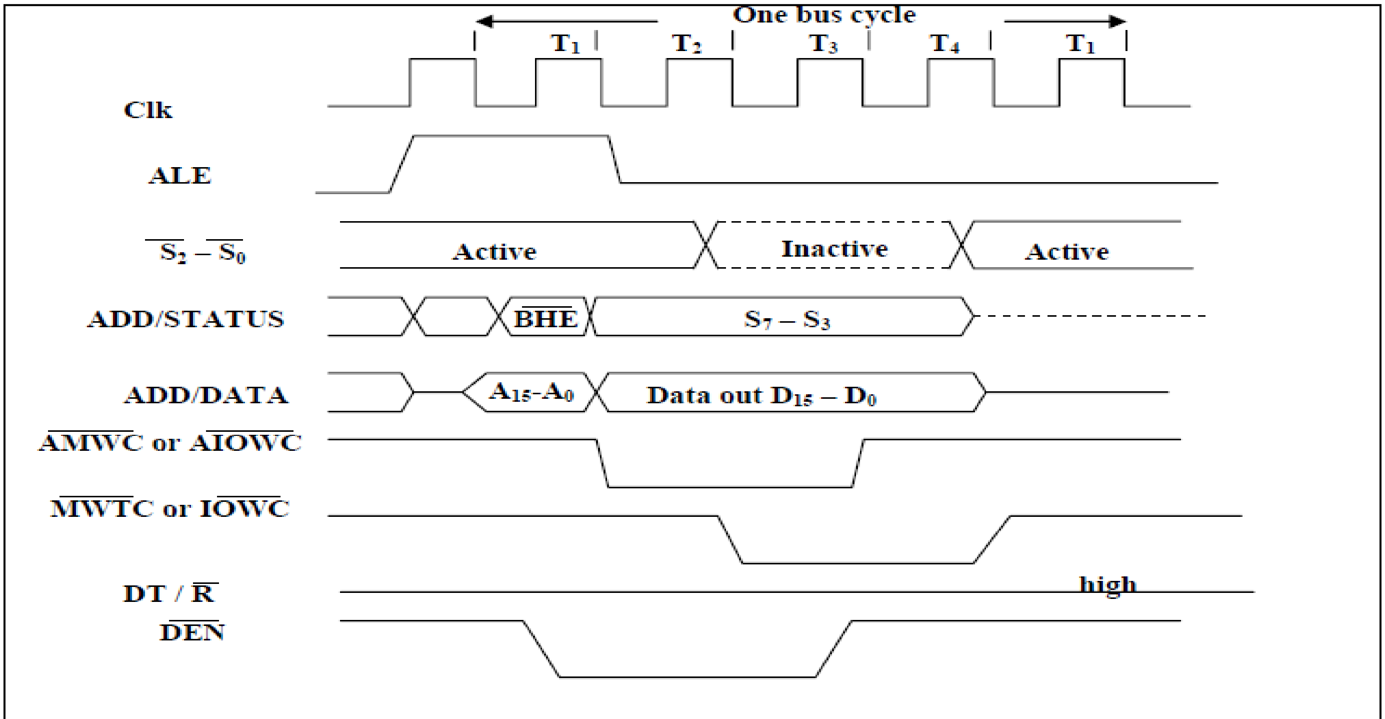
•In T2, 8288 will set DEN=1 thus enabling transceivers, and for an input it will activate MRDC or IORC. These signals are activated until T4. For an output, the AMWC or AIOWC is activated from T2 to T4 and MWTC or IOWC is activated from T3 to T4.

•The status bit S0 to S2 remains active until T3 and become passive during T3 and T4.

•If reader input is not activated before T3, wait state will be inserted between T3 and T4.



Memory Read Timing in Maximum Mode



Memory Write Timing in Maximum mode.

ADDRESS AND DATA BUS

The BIU has a combined address and data bus, commonly referred to as a time-multiplexed bus. Time multiplexing address and data information makes the most efficient use of device package pins. A system with address latching provided within the memory and I/O devices can directly connect to the address/data bus. The local bus can be demultiplexed with a single set of address latches to provide non-multiplexed address and data information to the system.

The programmer views the memory or I/O address space as a sequence of bytes. Memory space consists of 1 Mbyte, while I/O space consists of 64 Kbytes. Any byte can contain an 8-bit data element, and any two consecutive bytes can contain a 16-bit data element (identified as a word). The discussions in this section apply to both memory and I/O bus cycles. For brevity, memory bus cycles are used for examples and illustration.

The memory address space on a 16-bit data bus is physically implemented by dividing the address space into two banks of up to 512 Kbytes each (see Figure 1). One bank connects to the lower half of the data bus and contains even-addressed bytes (A0=0). The other bank connects to the upper half of the data bus and contains odd-addressed bytes (A0=1). Address lines A19:1 selects a specific byte within each bank. A0 and Byte High Enable ($\overline{BHE}$) determine whether one bank or both banks participate in the data transfer.

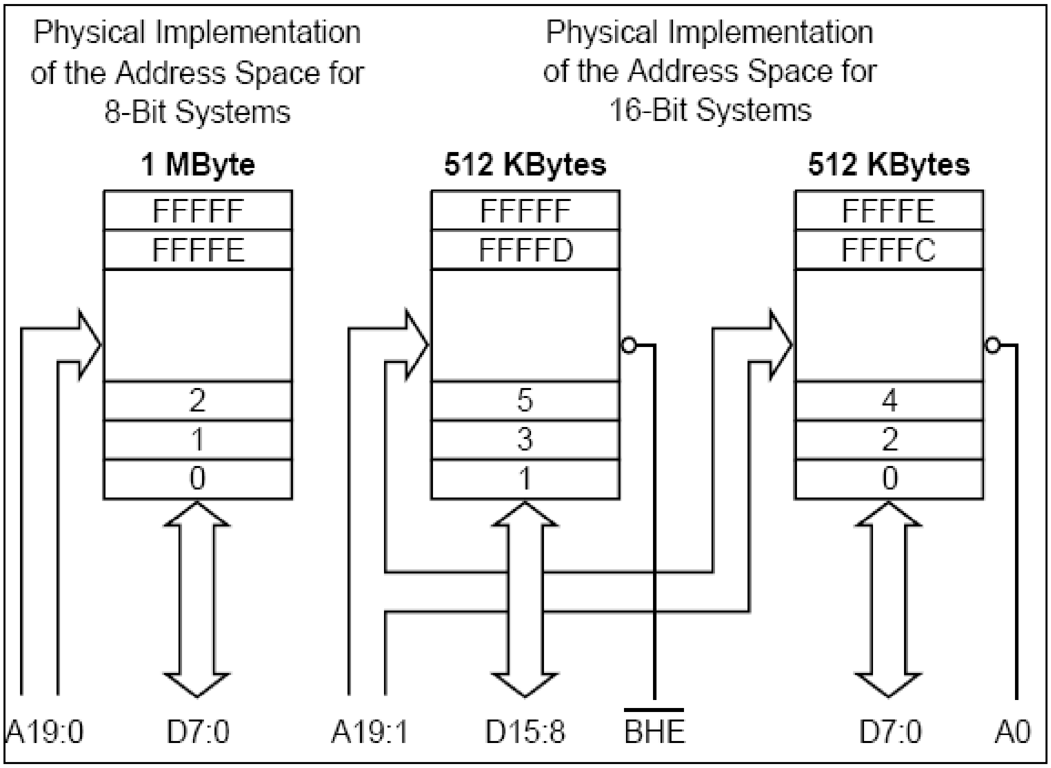

Physical Implementation of the Address Space for 8-Bit Systems
Physical Implementation of the Address Space for 16-Bit Systems
1 MByte
512 KBytes
512 KBytes
FFFFF
FFFFE
2
1
0
FFFFF
FFFFD
5
3
1
FFFFE
FFFFC
4
2
0
A19:0
D7:0
A19:1
D15:8
BHE
D7:0
A0

Figure 1. Physical Databus Models

Byte transfers to even addresses transfer information over the lower half of the data bus (see Figure-2a). A0 low enables the lower bank, while BHE high disables the upper bank. The data value from the upper bank is ignored during a bus read cycle. BHE high prevents a write operation from destroying data in the upper bank. Byte transfers to odd addresses transfer information over the upper half of the data bus (see Figure -2b). BHE low enables the upper bank, while A0 high disables the lower bank. The data value from the lower bank is ignored during a bus read cycle. A0 high prevents a write operation from destroying data in the lower bank. To access even-addressed 16-bit words (two consecutive bytes with the least-significant byte at an even address), information is transferred over both halves of the data bus (see Figure-3).

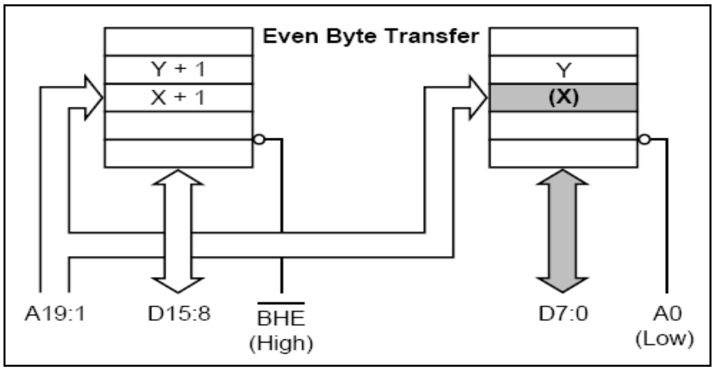


Figure.2a: 16-bit Data Bus Byte Transfers

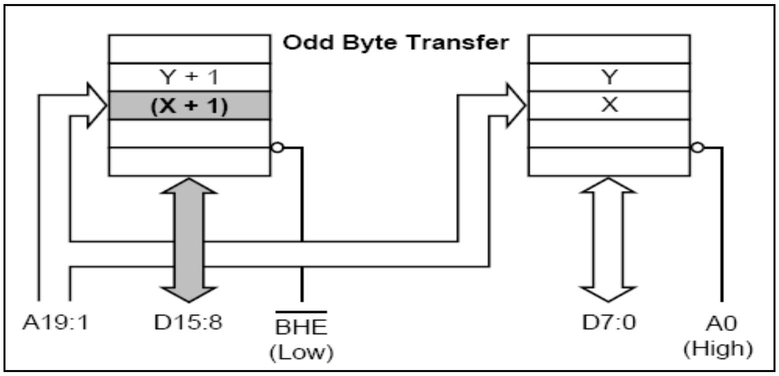


Figure.2b: 16-bit Data Bus Byte transfers

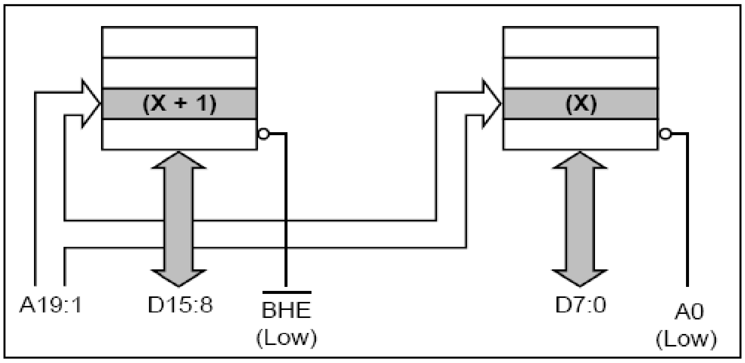


Figure-3: 16-Bit Data Bus Even Word Transfers

A19:1 select the appropriate byte within each bank. A0 and $\overline{BHE}$ drive low to enable both banks simultaneously. Odd-addressed word accesses require the BIU to split the transfer into two byte operations (see Figure-4). The first operation transfers data over the upper half of the bus, while the second operation transfers data over the lower half of the bus. The BIU automatically executes the two-byte sequence whenever an odd-addressed word access is performed.

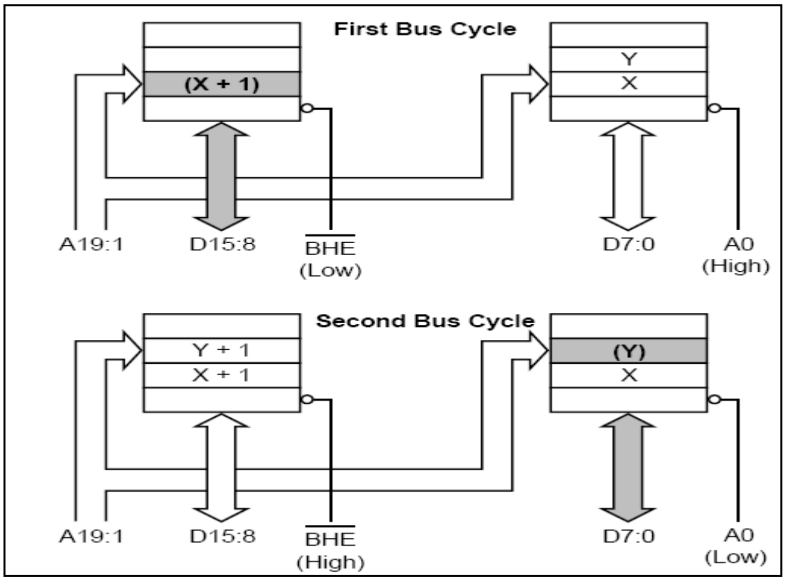


Figure -4 16-Bit data Bus Odd Word Transfers

MEMORY INTERFACING:

8086 Memory Banks

8086 has a 20 bit address bus and hence it can address 2^{20} or 1,048,576 addresses. In each location a byte is stored. So when a word is stored in the memory, it is stored in two consecutive memory locations. Strictly speaking, both memory read and memory write operations require more than one memory cycle. If we want 8086 to complete memory read and memory write operations to be completed with one machine cycle, the memory is to be organized in the form of two banks. Each bank will have 524,288 bytes each.

One memory bank contains all the even addressed locations like 00000, 00002 and 00004. The data lines of this bank are connected to the lower eight data lines, D0 through D7 of the 8086. The other memory bank has all the odd addressed locations like 00001, 00003 and 00005. The data lines of this bank are connected to the upper eight data lines, D8 through D15 of the 8086. Address line A0 is used as part of the enabling for memory in the lower bank. Address lines A1 through A19 are used to select the desired memory device in the bank to address the desired byte in the service. These address lines from A1 through A19 are also used to access a particular location in the upper bank. An additional signal called Bus High Enable (BHE – Active Low) is used to enable the upper memory bank. An external latch, strobed by ALE, grabs the BHE (Active Low) signal that holds it stable for the rest of the machine cycle. The following table shows the required logic levels on the BHE (Active Low) and A0 signals for various types of memory accesses.

Address	Data type	Bhe (active low)	A0	Bus cycles	Data lines used
0000	BYTE	1	0	ONE	D0-D7
0000	WORD	0	0	ONE	D0-D15
0001	BYTE	0	1	ONE	D7-D15
0001	WORD	0 1	1 0	FIRST SECOND	D0-D7 D7-D15

Case 1

Read/Write a byte form/to an even address – A0 will be low and BHE (Active Low) will be high – Byte is transferred to/from low bank through D0-D7

Example – MOV AH, DS: BYTE PTR [0000]

Case 2

Similar to case 1 except the word access instead of the byte access – Both A0 and BHE (Active Low) will be asserted low – Low byte of the word through D0-D7 and high byte of the word through D8-D15

Example – MOV AX,DS:WORD PTR[0000]

Case 3

Read/Wrire a byte from/to an odd address – A0 will be high and BHE (Active Low) will be asserted low – Low bank is disabled and high bank is enabled – Byte is transferred through D0-D7

Example – MOV AL,DS:BYTE PTR[0001]

Case 4

Read/Write a word from/to an odd address – 8086 requires two bus cycles – During the first machine cycle assert BHE (Active Low) as low and A0 as high – First byte is transferred through D0-D7 and the second byte is transferred through D8-D15

Example – MOV AX,DS:WORD PTR[0001H]

The memory is made up of semiconductor material used to store the programs and data. Three types of memory is,

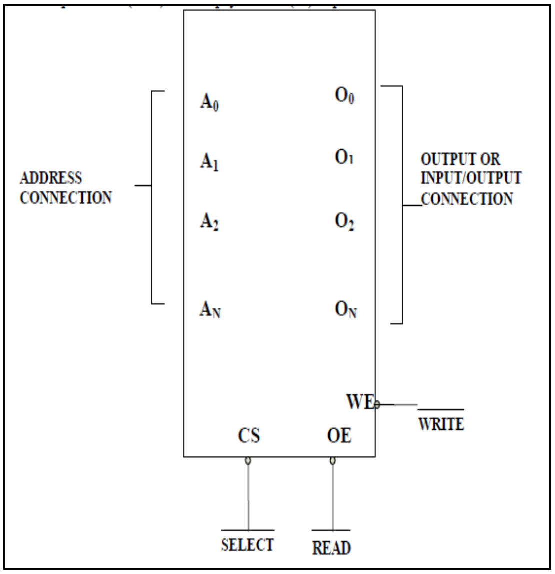
- Process memory
- Primary or main memory
- Secondary memory

Typical EPROM and static RAM:

- A typical semiconductor memory IC will have n address pins, m data pins (or output pins).
- Having two power supply pins (one for connecting required supply voltage (V and the other for connecting ground).
- The control signals needed for static RAM are chip select (chip enable), read control (output enable) and write control (write enable).
- The control signals needed for read operation in EPROM are chip select (chip enable) and read control (output enable).

Pin connections common to all memory devices are: The address input, data output or input/outputs, selection input and control input used to select a read or write operation.

- ***Address connections***: All memory devices have address inputs that select a memory location within the memory device. Address inputs are labeled from A0 to An.
- ***Data connections***: All memory devices have a set of data outputs or input/outputs. Today many of them have bi-directional common I/O pins.
- ***Selection connections***: Each memory device has an input, that selects or enables the memory device. This kind of input is most often called a chip select ($\overline{CS}$), chip enable ($\overline{CE}$) or simply select ($\bar{S}$) input.



Memory component illustrating the address, data and, Control connections

RAM memory generally has at least one $\overline{CS}$ or $\bar{S}$ input and ROM at least one $\overline{CE}$. A RAM memory device has either one or two control inputs. If there is one control input it is often called R/W. This pin selects a read operation or a write operation only if the device is selected by the selection input ($\overline{CS}$). If the RAM has two control inputs, they are usually labeled $\overline{WE}$ or $\overline{W}$ and $\overline{OE}$ or G.

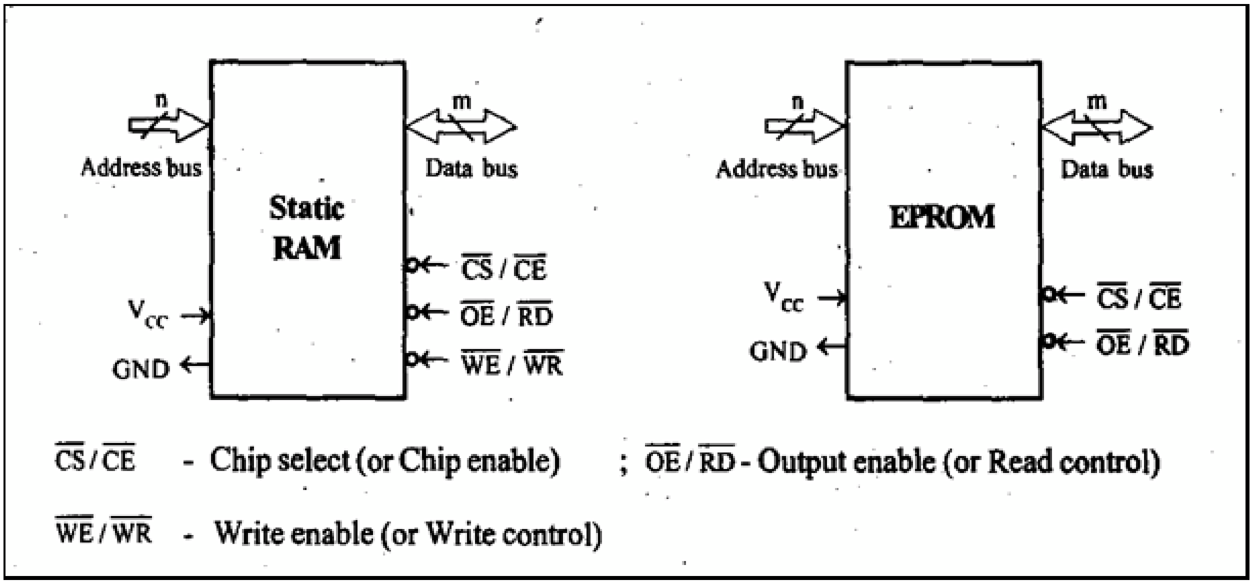
The ROM read only memory permanently stores programs and data and data was always present, even when power is disconnected. It is also called as nonvolatile memory.

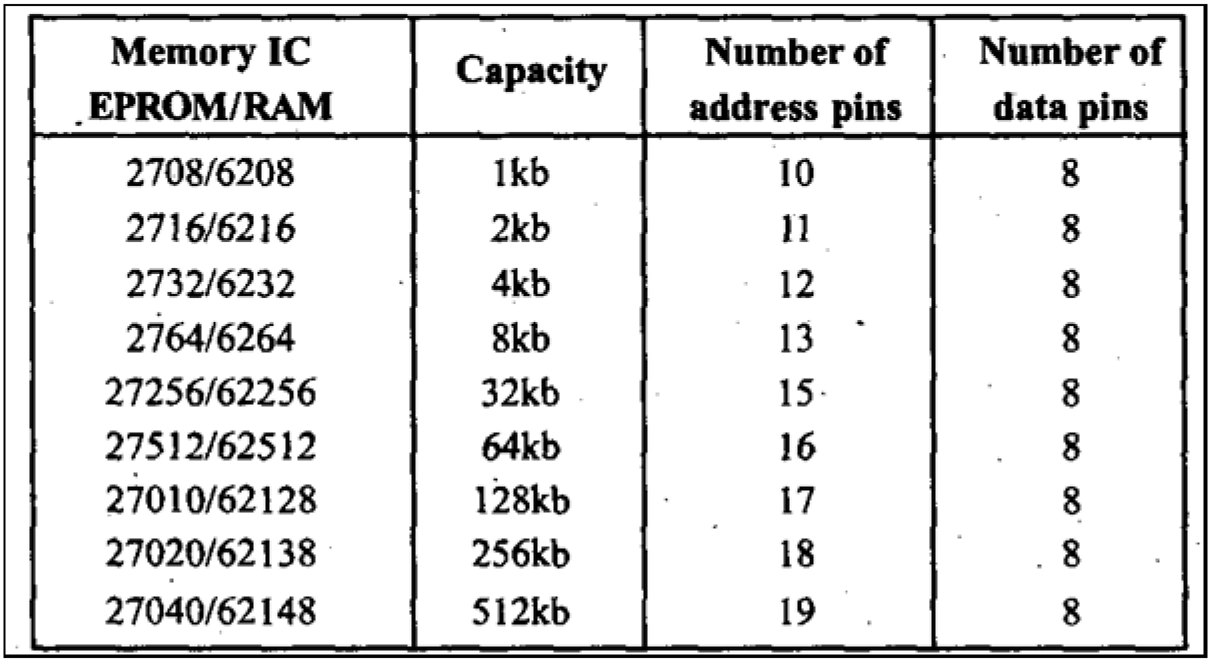
- EPROM (erasable programmable read only memory) is also erasable if exposed to high intensity ultraviolet light for about 20 minutes or less, depending upon the type of EPROM.
- We have PROM (programmable read only memory)
- RMM (read mostly memory) is also called the flash memory.
- The flash memory is also called as an EEPROM (electrically erasable programmable ROM), EAROM (electrically alterable ROM), or a NOVROM (nonvolatile ROM).
- These memory devices are electrically erasable in the system, but require more time to erase than a normal RAM.

- EPROM contains the series of 27XXX contains the following part numbers: 2704(512 * 8), 2708(1K * 8), 2716(2K * 8), 2732(4K * 8), 2764(8K * 8), 27128(16K * 8) etc.
- Each of these parts contains address pins, eight data connections, one or more chip selection inputs ($\overline{CE}$) and an output enable pin ($\overline{OE}$). This device contains **11** address inputs and **8** data outputs. If both the pin connection $\overline{CE}$ and $\overline{OE}$ are at logic 0, data will appear on the output connection. If both the pins are not at logic 0, the data output connections remain at their high impedance or off state.
- To read data from the EPROM Vpp pin must be placed at logic 1.

  - ✓ Static RAM memory device retain data for as long as DC power is applied. Because no special action is required to retain stored data, these devices are called as static memory. They are also called volatile memory because they will not retain data without power.
  - ✓ The main difference between a ROM and RAM is that a RAM is written under normal operation, while ROM is programmed outside the computer and is only normally read.

- ✓ The SRAM stores temporary data and is used when the size of read/write memory is relatively small.


n
Address bus
Static RAM
m
Data bus
$\overline{CS}$ / $\overline{CE}$
$\overline{OE}$ / $\overline{RD}$
$\overline{WE}$ / $\overline{WR}$
V_{cc}
GND
n
Address bus
EPROM
m
Data bus
V_{cc}
GND
$\overline{CS}$ / $\overline{CE}$
$\overline{OE}$ / $\overline{RD}$
$\overline{CS}$ / $\overline{CE}$ - Chip select (or Chip enable) ; $\overline{OE}$ / $\overline{RD}$ - Output enable (or Read control)
$\overline{WE}$ / $\overline{WR}$ - Write enable (or Write control)



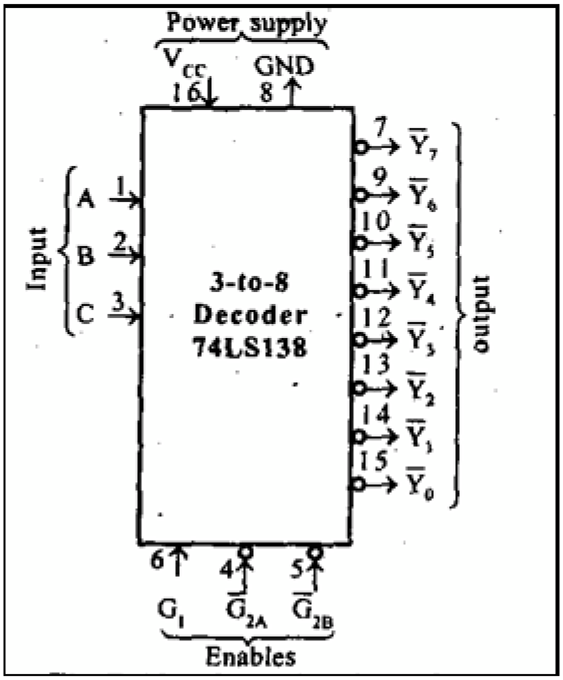
Memory IC EPROM/RAM	Capacity	Number of address pins	Number of data pins
2708/6208	1kb	10	8
2716/6216	2kb	11	8
2732/6232	4kb	12	8
2764/6264	8kb	13	8
27256/62256	32kb	15	8
27512/62512	64kb	16	8
27010/62128	128kb	17	8
27020/62138	256kb	18	8
27040/62148	512kb	19	8

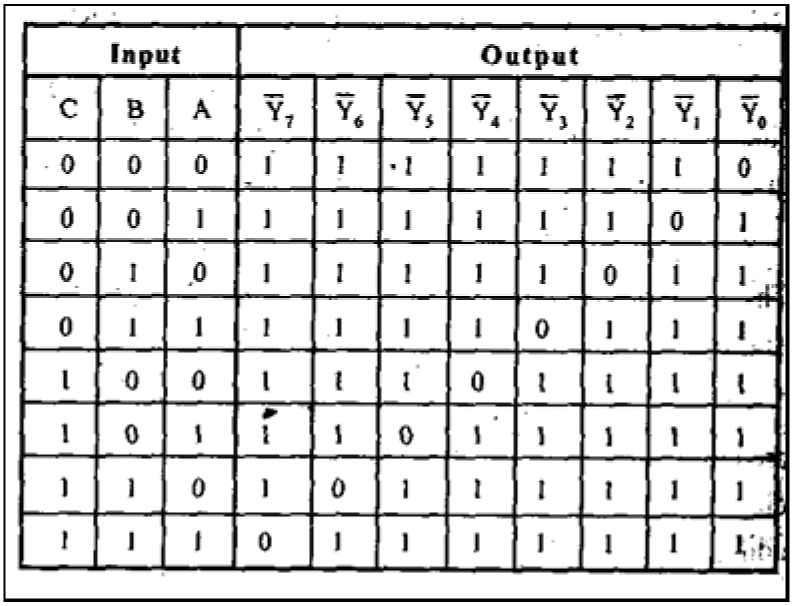
Table - Number of Address Pins and Data Pins in Memory ICs

Decoder:

It is used to select the memory chip of processor during the execution of a program. No of IC's used for decoder is,

- 2-4 decoder (74LS139)
- 3-8 decoder (74LS138)





Input			Output							
C	B	A	$\overline{Y}_7$	$\overline{Y}_6$	$\overline{Y}_5$	$\overline{Y}_4$	$\overline{Y}_3$	$\overline{Y}_2$	$\overline{Y}_1$	$\overline{Y}_0$
0	0	0	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	0	1
0	1	0	1	1	1	1	1	0	1	1
0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	1	1	0	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1
1	1	0	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1

Fig - Block diagram and Truth table of 3-8 decoder

Static RAM Interfacing

- The semiconductor RAM is broadly two types – Static RAM and Dynamic RAM.
- The semiconductor memories are organized as two dimensional arrays of memory locations.
- For example 4K * 8 or 4K byte memory contains 4096 locations, where each locations contains 8-bit data and only one of the 4096 locations can be selected at a time. Once a location is selected all the bits in it are accessible using a group of conductors called Data bus.
- For addressing the 4K bytes of memory, 12 address lines are required.
- In general to address a memory location out of N memory locations, we will require at least n bits of address, i.e. n address lines where n = Log2 N.
- Thus if the microprocessor has n address lines, then it is able to address at the most N locations of memory, where 2n=N. If out of N locations only P memory locations are to be interfaced, then the least significant p address lines out of the available n lines can be directly connected from the microprocessor to the memory chip while the remaining (n-p) higher order address lines may be used for address decoding as inputs to the chip selection logic.
- The memory address depends upon the hardware circuit used for decoding the chip select ($\overline{CS}$). The output of the decoding circuit is connected with the $\overline{CS}$ pin of the memory chip.

- The general procedure of static memory interfacing with 8086 is briefly described as follows:

1. Arrange the available memory chip so as to obtain 16- bit data bus width. The upper 8-bit bank is called as odd address memory bank and the lower 8-bit bank is called as even address memory bank.
2. Connect available memory address lines of memory chip with those of the microprocessor and also connect the memory $\overline{RD}$ and $\overline{WR}$ inputs to the corresponding processor control signals. Connect the 16-bit data bus of the memory bank with that of the microprocessor 8086.
3. The remaining address lines of the microprocessor, $\overline{BHE}$ and A0 are used for decoding the required chip select signals for the odd and even memory banks. The $\overline{CS}$ of memory is derived from the o/p of the decoding circuit.

- As a good and efficient interfacing practice, the address map of the system should be continuous as far as possible, i.e. there should not be no windows in the map and no fold back space should be allowed.
- A memory location should have a single address corresponding to it, i.e. absolute decoding should be preferred and minimum hardware should be used for decoding.

Example: Interface 8K*8 EPROM and 4K*8 SRAM chips with 8086 microprocessor. Select suitable map.

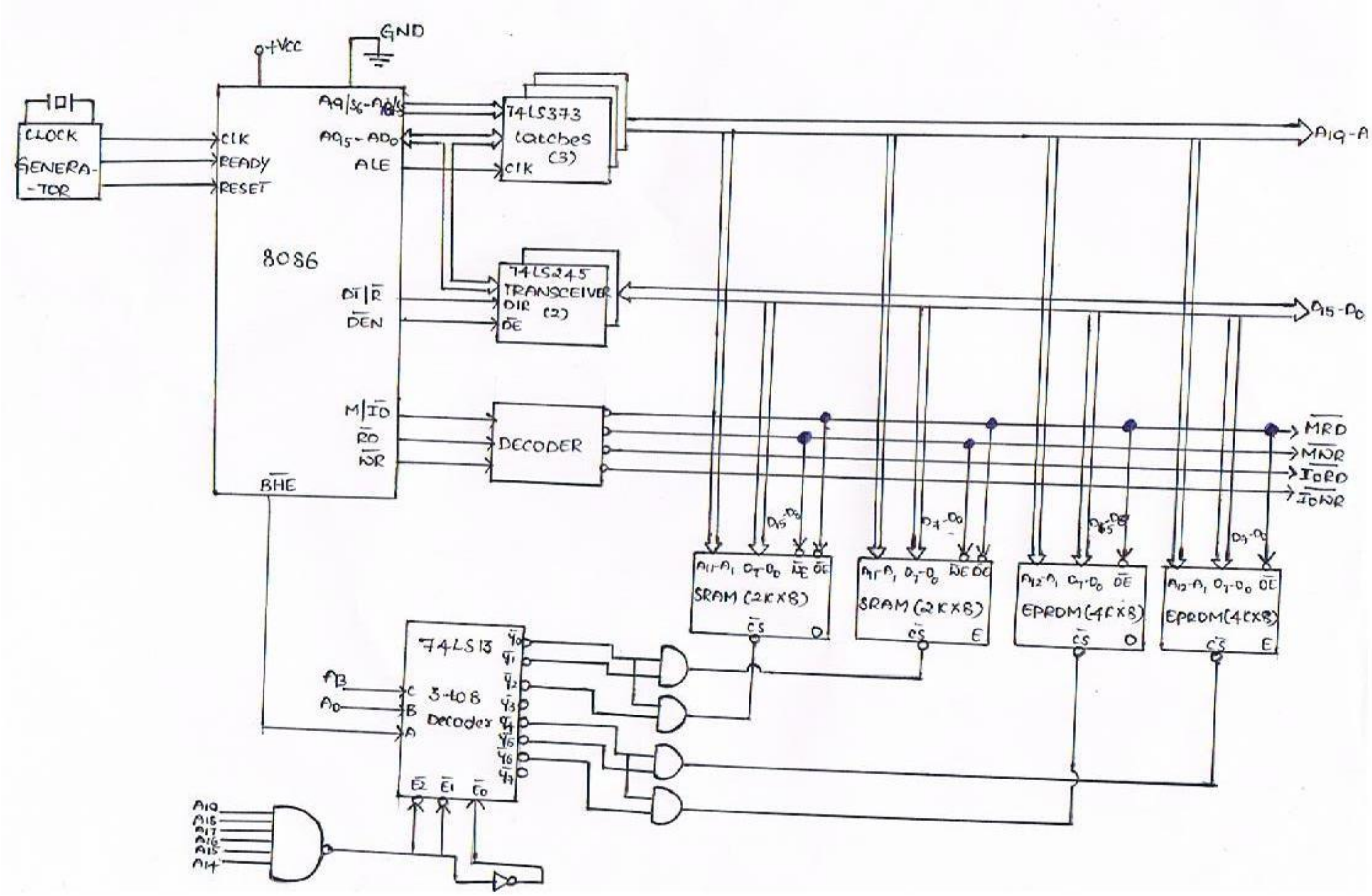
Memory map:

	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
FFFFF	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
EPROM																				
FE000	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
FDFFF	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
SRAM																				
FC000	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Memory chip selection:

Input C	**B**	**A**	**Data transfer**	**Memory chips selection**
$A13$	$A0$	$\overline{BHE}$		
0	0	0	Word transfer through D15-D0	Both even & odd banks in SRAM
0	0	1	Lower byte transfer through D7-D0	Even bank in SRAM
0	1	0	Higher byte through D15-D8	Odd bank in SRAM
1	0	0	Word transfer through D15-D0	Both even & odd banks in EPROM
1	0	1	Lower byte transfer through D7-D0	Even bank in EPROM
1	1	0	Higher byte through D15-D8	Odd bank in EPROM

Interfacing:



DIRECT MEMORY ACCESS

Need for DMA & DMA Transfer method

An important aspect governing the Computer System performance is the transfer of data between memory and I/O devices. The operation involves loading programs or data files from disk into memory, saving file on disk, and accessing virtual memory pages on any secondary storage medium. Consider a typical system consisting of a CPU, memory and one or more input/output devices as shown in fig. Assume one of the I/O devices is a disk drive and that the computer must load a program from this drive into memory. The CPU would read the first byte of the program and then write that byte to memory. Then it would do the same for the second byte, until it had loaded the entire program into memory. This process proves to be inefficient. Loading data into, and then writing data out of the CPU significantly slows down the transfer. The CPU does not modify the data at all, so it only serves as an additional stop for data on the way to it's final destination. The process would be much quicker if we could bypass the CPU & transfer data directly from the I/O device to memory. Direct Memory Access does exactly that.

A DMA controller implements direct memory access in a computer system.

It connects directly to the I/O device at one end and to the system buses at the other end. It also interacts with the CPU, both via the system buses and two new direct connections. It is sometimes referred to as a channel. In an alternate configuration, the DMA controller may be incorporated directly into the I/O device.

Direct Memory Access--the ability of an I/O subsystem to transfer data to and from a memory subsystem without processor intervention

DMA Controller--a device that can control data transfers between an I/O subsystem and a memory subsystem in the same manner that a processor can control such transfers.

The DMA controller can issue commands to the memory that behave exactly like the commands issued by the microprocessor. The DMA controller in a sense is a second processor in the system but is dedicated to an I/O function. The DMA controller as shown below connects one or more I/O ports directly to memory, where the I/O data stream passes through the DMA controller faster and more efficiently than through the processor as the DMA channel is specialised to the data transfer task.

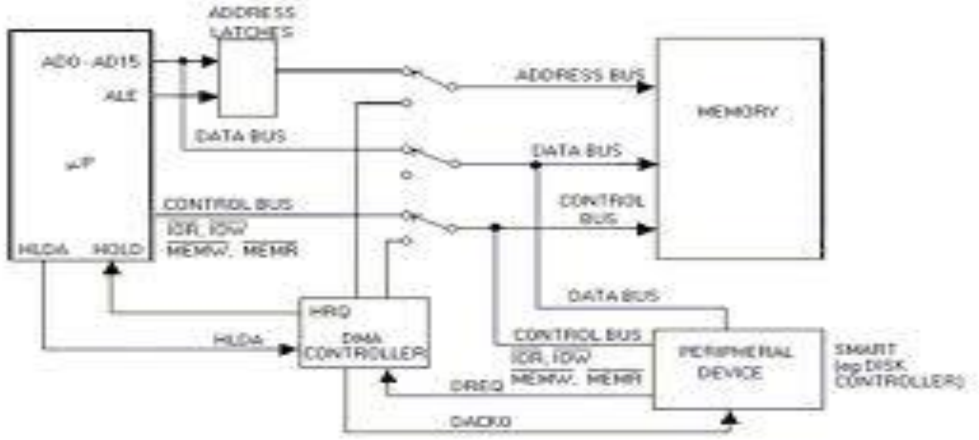

ADDRESS LATCHES
AD0 - AD15
ALE
μP
HLDA
HOLD
DATA BUS
CONTROL BUS
$\overline{IOR}$, $\overline{IOW}$
$\overline{MEMW}$, $\overline{MEMR}$
ADDRESS BUS
MEMORY
DATA BUS
CONTROL BUS
HRQ
HLDA
DMA CONTROLLER
DATA BUS
CONTROL BUS
$\overline{IOR}$, $\overline{IOW}$
$\overline{MEMW}$, $\overline{MEMR}$
DREQ
DACK0
PERIPHERAL DEVICE
SMART (eg DISK CONTROLLER)

Figure 2: Block diagram showing how a DMA controller operates in a microcomputer system

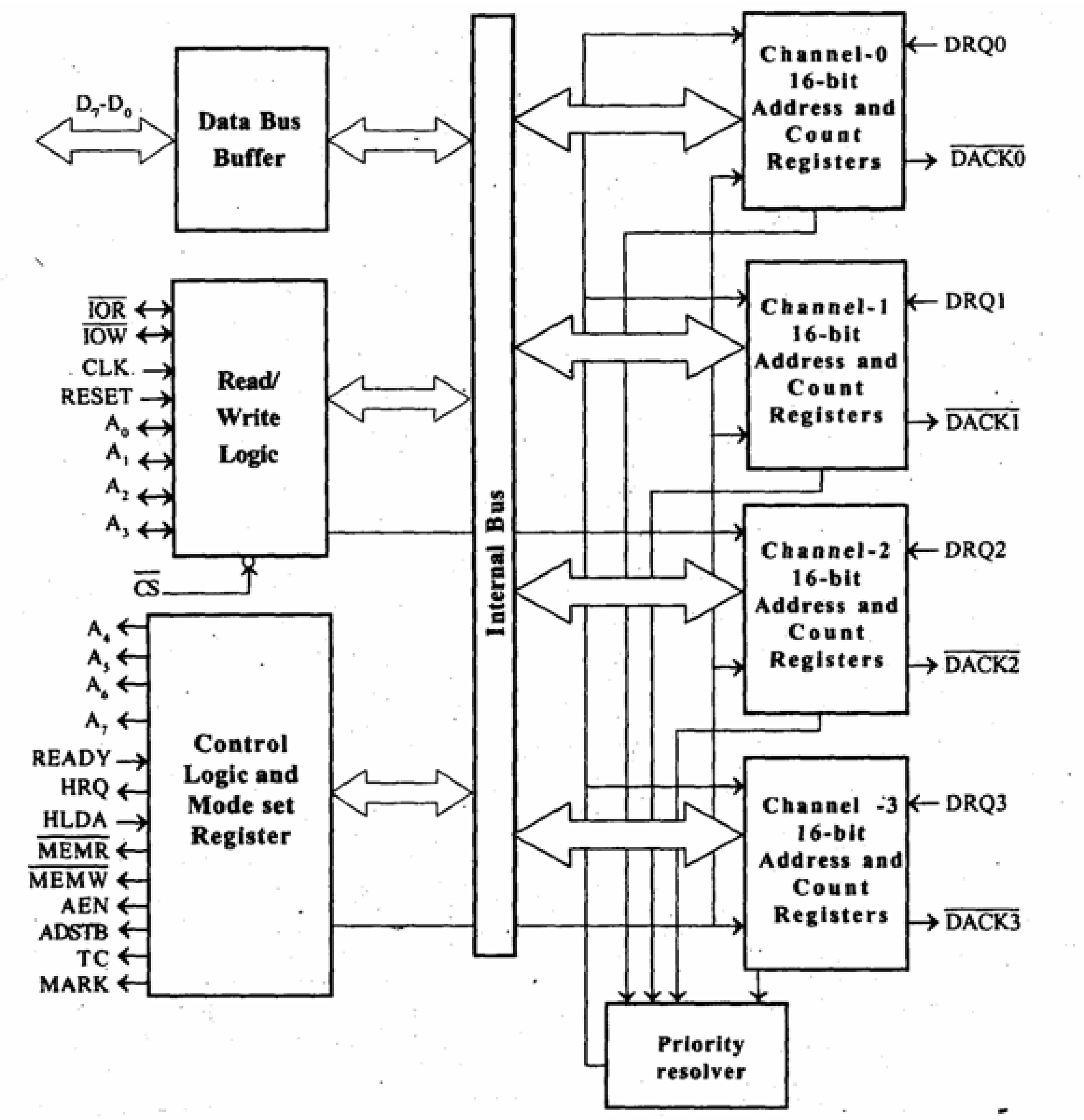
8257 DMA CONTROLLER

The Intel 8257 is a 4-channel direct memory access (DMA) controller. It is specifically designed to simplify the transfer of data at high speeds for the Intel® microcomputer systems. Its primary function is to generate, upon a peripheral request, a sequential memory address which will allow the peripheral to read or write data directly to or from memory. The 8257 has priority logic that resolves the peripherals requests and issues a composite hold request to the microprocessor. It maintains the DMA cycle count for each channel and outputs a control signal to notify the peripheral that the programmed number of DMA cycles is complete.

Features:

- Compatible with 8085, 8086/88
- It is a 4-Channel DMA Controller. So 4- I/O devices can be interfaced to DMA.
- Each channel has 16-bit address and 14 bit counter.
- It provides chip priority resolver that resolves priority of channels in fixed or rotating mode.
- Provides Terminal Count and Modulo 128 Outputs
- It requires Single TTL Clock
- It requires Single + 5V power Supply
- Available in Standard Temperature Range

Architecture:


D7-D0
Data Bus Buffer
IOR
IOW
CLK
RESET
A0
A1
A2
A3
Read/ Write Logic
CS
A4
A5
A6
A7
READY
HRQ
HLDA
MEMR
MEMW
AEN
ADSTB
TC
MARK
Control Logic and Mode set Register
Internal Bus
Channel-0 16-bit Address and Count Registers
DRQ0
DACK0
Channel-1 16-bit Address and Count Registers
DRQ1
DACK1
Channel-2 16-bit Address and Count Registers
DRQ2
DACK2
Channel -3 16-bit Address and Count Registers
DRQ3
DACK3
Priority resolver

Block Diagram Description

DMA Channels

The 8257 provides four separate DMA channels (labelled CH-0 to CH-3). Each channel includes two sixteen-bit registers: (1) a DMA address register, and (2) a terminal count register. Both registers must be initialized before a channel is enabled. The DMA address register is loaded with the address of the first memory location to be accessed. The value loaded into the low-order 14-bits of the terminal count register specifies the number of DMA cycles minus one before the Terminal Count (TC) output is activated. For instance, a terminal count of 0 would cause the TC output to be active in the first DMA cycle for that channel. In general, if N = the number of desired DMA cycles, load the value N-1 into the low-order 14-bits of the terminal count register. The most significant two bits of the terminal count register specify the type of DMA operation for that channel.

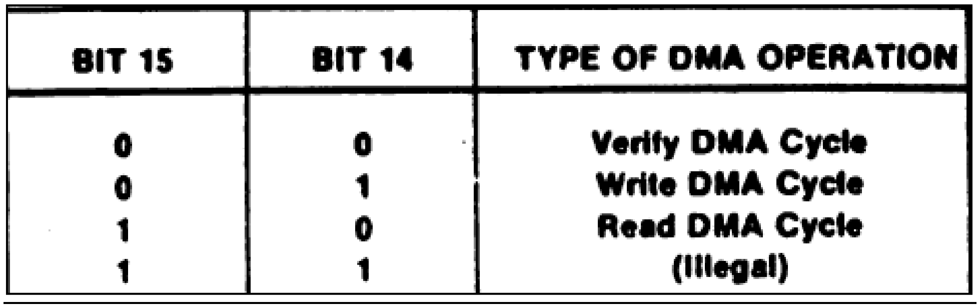
DMA Request (DRQ 0-DRQ 3): These are individual asynchronous channel request inputs used by the peripherals to obtain a DMA cycle. If not in the rotating priority mode then DRQ 0 has the highest priority and DRQ 3 has the lowest. A request can be generated by raising the request line and holding it high until DMA acknowledge. For multiple DMA cycles (Burst Mode) the request line is held high until the DMA acknowledge of the last cycle arrives.

DMA Acknowledge (DACK 0 - DACK 3) : An active low level on the acknowledge output informs the peripheral connected to that channel that it has been selected for a DMA cycle. The DACK output acts as a "chip select' for the peripheral device requesting service. This line goes active (low) and inactive (high) once for each byte transferred even if a burst of data is being transferred.

Data bus buffer

This bi-directional 8-bit interfaces the 8257 to the microprocessor system data bus.

Data Bus Lines: These are bi-directional three-state lines. When the 8257 is being programmed by the CPU. Eightbits of data for a DMA address register, a terminal count register or the Mode Set register are received on the data bus. When the CPU reads a DMA address register, a terminal count register or the Status register, the data is sent to the CPU over the data bus. During DMA cycles (when the 8257 is the bus master), the 8257 will output the most significant eight-bits of the memory address (from one of the DMA address registers) to the 8212 latch via the data bus.



BIT 15	BIT 14	TYPE OF DMA OPERATION
0	0	Verify DMA Cycle
0	1	Write DMA Cycle
1	0	Read DMA Cycle
1	1	(Illegal)

Read/Write Logic

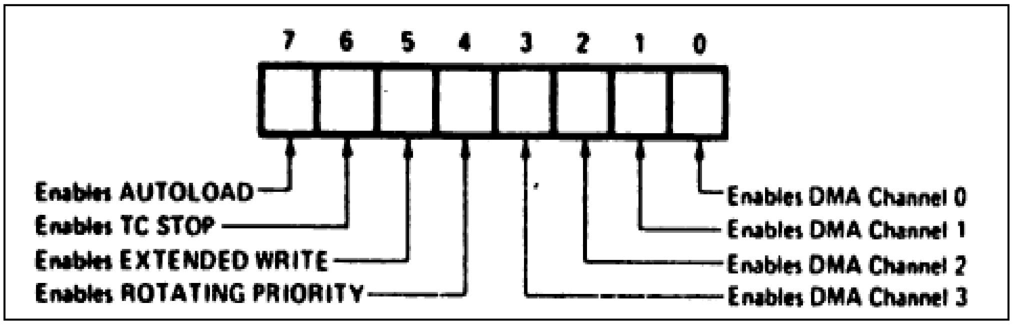
When the CPU is programming or reading one of the 8257's registers (i.e., when the 8257 is a "slave" device on the system bus), the Read/Write Logic accepts the I/O Read (USE) or I/O Write (175OT) signal, decodes the least significant four address bits, (A0-A3), and either writes the contents of the data bus into the addressed register (if I/OW is true) or places the contents of the addressed register onto the data bus (if I/OR is true). During DMA cycles (i.e., when the 8257 is the bus "master"), the Read/Write Logic generates the I/O read and memory write (DMA write cycle) or I/O Write and memory read (DMA read cycle) signals which control the data link with the peripheral that has been granted the DMA cycle.

Control Logic

This block controls the sequence of operations during all DMA cycles by generating the appropriate control signals and the 16-bit address that specifies the memory location to be accessed.

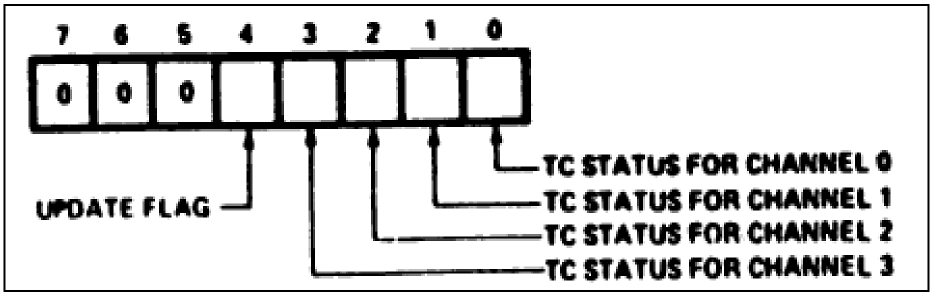
Mode Sat Register

When set, the various bits in the Mode Set register enable each of the four DMA channels, and allow four different options for the 8257:

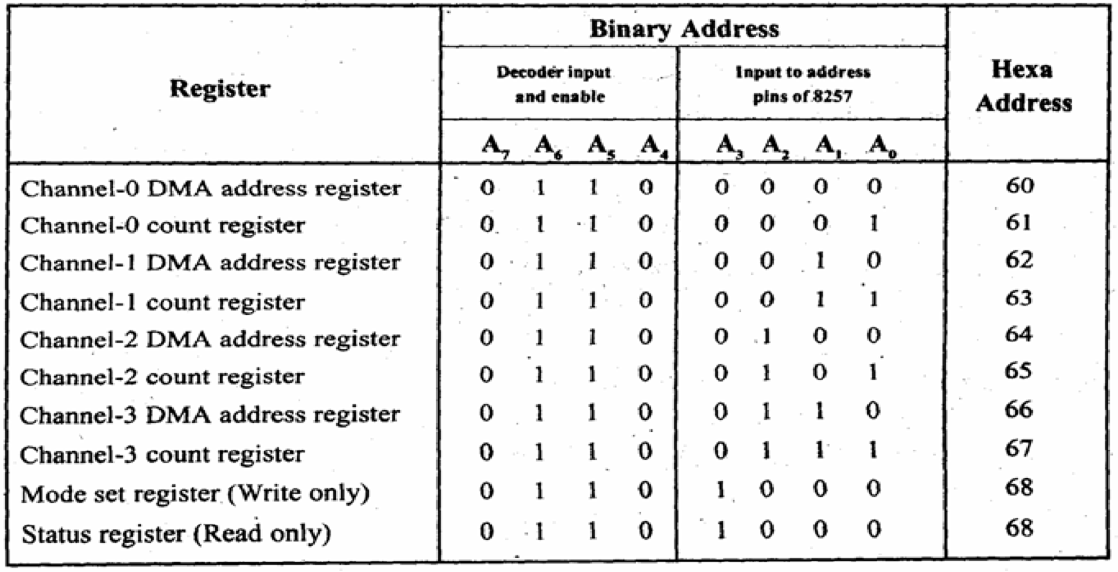


Status Register

The eight-bit status register indicates which channels have reached a terminal count condition and includes the update flag described previously. The TC status bits are set when the Terminal Count (TC) output is activated for that channel. These bits remain set until the status register is read or the 8257 is reset. The UPDATE FLAG, however, is not affected by a status register read operation. The UPDATE FLAG can be cleared by resetting the 8257. by changing to the non-auto load mode (i.e.. by resetting the AUTO LOAD bit in the Mode Set register) or it can be left to clear itself at the completion of the update cycle.

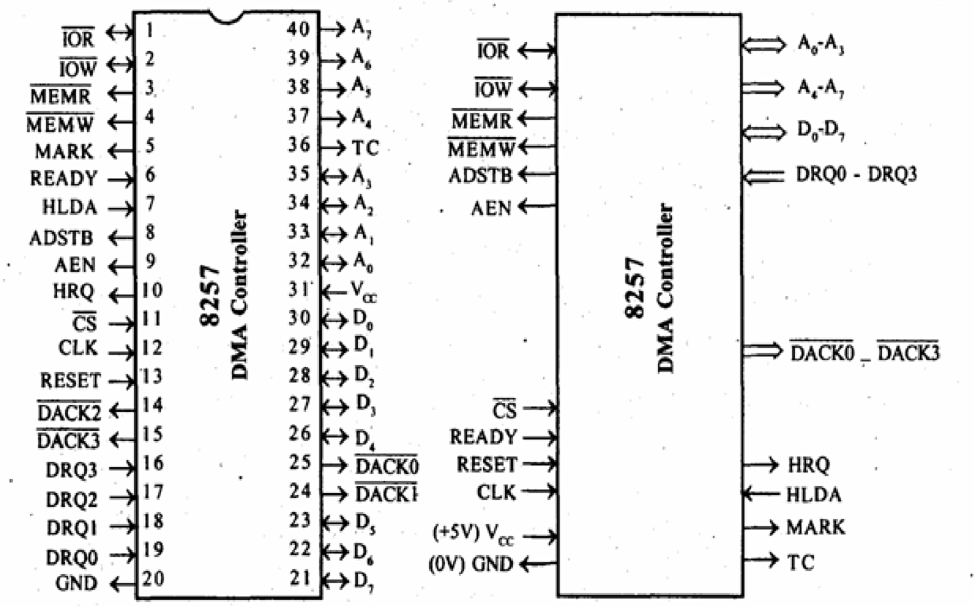


Register Selection in 8257



Register	Binary Address								Hexa Address
	Decoder input and enable				Input to address pins of 8257				
	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
Channel-0 DMA address register	0	1	1	0	0	0	0	0	60
Channel-0 count register	0	1	1	0	0	0	0	1	61
Channel-1 DMA address register	0	1	1	0	0	0	1	0	62
Channel-1 count register	0	1	1	0	0	0	1	1	63
Channel-2 DMA address register	0	1	1	0	0	1	0	0	64
Channel-2 count register	0	1	1	0	0	1	0	1	65
Channel-3 DMA address register	0	1	1	0	0	1	1	0	66
Channel-3 count register	0	1	1	0	0	1	1	1	67
Mode set register (Write only)	0	1	1	0	1	0	0	0	68
Status register (Read only)	0	1	1	0	1	0	0	0	68

PIN CONFIGURATION OF 8257



Pin descriptions

D0-D7:

- These are bidirectional, tri state, Buffered, Multiplexed data (D0-D7) and (A8-A15).
- In the slave mode it is a bidirectional (Data is moving).
- In the Master mode it is a unidirectional (Address is moving)

IOR:

- It is active low ,tristate ,buffered ,Bidirectional lines.
- In the slave mode it function as a input line. IOR signal is generated by microprocessor to read the contents 8257 registers.
- In the master mode it function as a output line. IOR signal is generated by 8257 during write cycle.

IOW:

- It is active low ,tristate ,buffered ,Bidirectional control lines.
- In the slave mode it function as a input line. IOR signal is generated by microprocessor to write the contents 8257 registers.
- In the master mode it function as a output line. IOR signal is generated by 8257 during read cycle.

CLK:

- It is the input line ,connected with TTL clock generator.
- This signal is ignored in slave mode.

RESET: Used to clear mode set registers and status registers

A0-A3: These are the tristate, buffer, bidirectional address lines. In slave mode, these lines are used as address inputs lines and internally decoded to access the internal registers. In master mode, these lines are used as address outputs lines,A0-A3 bits of memory address on the lines.

CS: It is active low, Chip select input line. In the slave mode, it is used to select the chip. In the master mode, it is ignored.

A4-A7: These are the tristate, buffer, output address lines. In slave mode, these lines are used as address outputs lines. In master mode, these lines are used as address outputs lines,A0-A3 bits of memory address on the lines.

READY: It is a asynchronous input line. In master mode, When ready is high it is received the signal. When ready is low, it adds wait state between S1 and S3 In slave mode, this signal is ignored.

HRQ: It is used to receiving the hold request signal from the output device.

HLDA: It is acknowledgment signal from microprocessor.

$\overline{MEMR}$: It is active low ,tristate ,Buffered control output line.In slave mode, it is tristated. In master mode, it activated during DMA read cycle.

$\overline{MEMW}$: It is active low ,tristate ,Buffered control input line. In slave mode, it is tristated. In master mode ,it activated during DMA write cycle.

AEN (Address enable): It is a control output line. In master mode ,it is high. In slave mode ,it is low.Used it isolate the system address ,data ,and control lines.

ADSTB (Address Strobe): It is a control output line. Used to split data and address line. It is working in master mode only. In slave mode it is ignore.

TC (Terminal Count): It is a status of output line. It is activated in master mode only. It is high , it selected the peripheral. It is low ,it free and looking for a new peripheral.

MARK:

- It is a modulo 128 MARK output line.
- It is activated in master mode only.
- It goes high, after transferring every 128 bytes of data block.

DRQ0-DRQ3(DMA Request):

- These are the asynchronous peripheral request input signal.
- The request signals are generated by external peripheral device.

$\overline{DACK0}$-$\overline{DACK3}$:

- These are the active low DMA acknowledge output lines.
- Low level indicates that, peripheral is selected for giving the information (DMA cycle).
- In master mode it is used for chip select.

OPERATING MODES OF 8257

The operating modes of 8257 DMA controller are

- Fixed priority mode
- Rotating priority mode
- Extended write mode
- TC stop mode
- Auto load mode

Fixed priority mode: If the ROTATING PRIORITY bit is not set (set to a zero), each DMA channel has a fixed priority. In this mode Channel 0 has the highest priority and Channel 3 has the lowest priority.

Rotating priority mode: In the Rotating Priority Mode, the priority of the channels has a circular sequence. After each DMA cycle, the priority of each channel changes. The channel which had just been serviced will have the lowest priority.

Extended write mode: If the EXTENDED WRITE bit is set, the duration of the MEMW and I/OW signals is extended by activating them earlier in the DMA cycle. Data transfers within micro computer systems proceed asynchronously to allow use of various types of memory and I/O devices with different access times.

TC stop mode: If the TC STOP bit is set a channel is disabled (i.e.. its enable bit is reset) after the Terminal Count (TC) output goes true, thus automatically preventing further DMA operation on that channel. The enable bit for that channel must be re-programmed to continue or begin another DMA operation. If the TC STOP bit is not set. The occurrence of the TC output has no effect on the channel enable bits.

Auto load mode: The Auto Load mode permits Channel 2 to be used for repeat block or block chaining operations, without immediate software intervention between blocks. Channel 2 registers are initialized as usual for the first data block; Channel 3 registers, however, are used to store the block re-initialization parameters (DMA starting address, terminal count and DMA transfer mode).

INTERFACING OF 8257 WITH 8086

--Refer your NOTE BOOK--