



**KTU ASSIST**

a ktu students community

[www.ktuassist.in](http://www.ktuassist.in)



# KTU STUDY MATERIALS



**APJ ABDUL KALAM  
TECHNOLOGICAL UNIVERSITY**

DOWNLOAD KTU ASSIST APP FROM PLAYSTORE

Lecture No: 1    Title: Evolution of Microprocessor

<u>Generation</u>	<u>Year</u>	<u>Technology</u>	<u>Processor</u>
First Generation	b/w 1971 & 1973	PMOS P-type metallic oxide Semiconductor	4-bit
Second	1973 - 1978	NMOS N-type metallic oxide Semiconductor	8-bit
Third	1978 - 1980	HMOS hybrid / high density metallic oxide Semiconductor	16-bit
Fourth	1980 - 1993	HCMOS High Speed Complementary metal oxide Semiconductor	32-bit
Fifth	1993 onwards		64-bit

Additional Points

→ First Generation

- ✓ PMOS technology provided
  - \* low cost
  - \* slow speed
  - \* low output current

✓ Require as high as 30 ICs to form a system.

✓ 1 generation processors  $\leftrightarrow$  4-bit ; 16 pins.

Mar Basilio College of Engineering & Technology

eg: Intel Corporation introduced 4004,  
the first microprocessor in 1971.

i.e. INTEL 4004      } 4-bit processors  
INTEL 4040 etc...      }

INTEL 8008      } 8-bit processors  
MOSTEK 5065 etc...      }

NATIONAL IMP 16      } 16-bit processors  
NATIONAL PACE      }

### SECOND GENERATION

- ✓ Marked the beginning of very efficient 8-bit microprocessors.
- ✓ NMOS technology offers faster speed and higher density than PMOS.
- ✓ 40 pins
- ✓ More no. of on-chip decoded timing sigs.
- ✓ Ability to address large memory spaces.
- ✓ Ability to address more I/O ports.
- ✓ Faster operations
- ✓ More powerful instruction set.
- ✓ Better interrupt handling capabilities.

eg: INTEL 8080      ZILOG Z80      } 8-bit  
INTEL 8085

Mar Baselios College of Engineering & Technology  
0.5 MIPS (execute 0.5 million instructions per second)

INTERSIL 6100      } 12-bit processor  
TOSHIBA TLCS-12      }

TI TMS 9900      }  
General Instrument CP 1600      } 16-bit processor

### \* Third Generation (1978 - 1980)

- ✓ This age is dominated by 16-bit microprocessors
- ✓ Some of them were:

INTEL's 8086 / 80186 / 80286

MOTOROLA 68000 / 68010

- ✓ Provided with 40/48/64 pins.
- ✓ High Speed and very strong processing capability.
- ✓ Easier to program.
- ✓ Size of internal registers are 8/16/32 bits.
- ✓ Physical memory space is from 1 to 16 Megabytes
- ✓ flexible I/O port addressing.
- ✓ Different modes of operations.

### \* Fourth Generation (1980 - 1995)

- ✓ This era marked the beginning of 32-bit microprocessors.
- ✓ Using low-power version of the HMOS technology called HCMOS.

- \* (2<sup>24</sup>) 16 Mb physical memory space.
  - \* (2<sup>40</sup>) 1 Tb virtual memory space.
  - \* Floating point hardware is incorporated.
  - \* Supported increase no. of addressing modes.
- eg: INTEL 80386 / 80486 → 54 MIPS  
(execute 54 Million instructions per second)
- MOTOROLA MC68100 / M68020 / M68030

### FIFTH Generation

→ This age the emphasis is on introducing chips that carry on-chip functionalities an improvement in the speed of memory and I/O devices, along with introduction of "64-bit microprocessors"

→ Intel leads the show here with Pentium, Celeron and very recently dual & quad core processors working with upto 3.5 GHz speed. A pentium processor can execute 2000 MIPS.

Question: Explain the evolution of microprocessors

### Reference

Microprocessors and microcontrollers, 2e.  
 A. Nagoor kani Page No: 1.3 to 1.5

### Title: Comparison of Microprocessor Vs Microcontroller

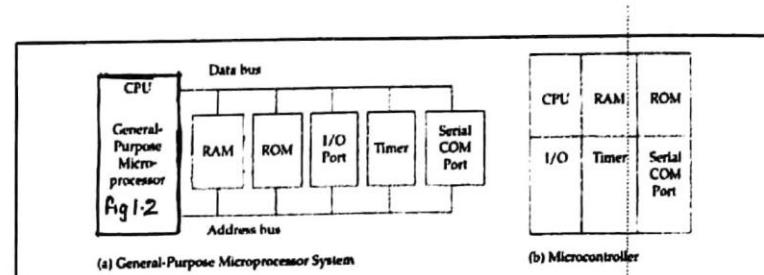


Figure 1-1. Microprocessor System Contrasted With Microcontroller System

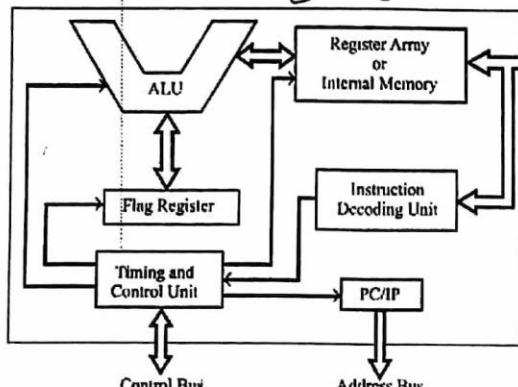
### Microprocessor

- \* Functional blocks
  - ALU, Registers & timing & control unit.
- \* Large no. of instructions for moving data b/w the external memory and the microprocessor.
- \* Few no. of bit manipulating instructions.
- \* Require interfacing of a large no. of additional ICs to form a microcomputer based system.
- \* Used for designing general purpose digital computing system.

### Microcontroller

- \* Functional blocks
  - ↳ microprocessor, timer, parallel & serial I/O port, Internal RAM, EEPROM/EEPROM memory & ADC or DAC converter.
- \* Few instructions for data transfer.
- \* large no. of bit manipulating instructions
- \* Microcontroller can be used to form a single chip microcomputer based system without any additional ICs.
- \* Used for designing application specific dedicated systems

For any microprocessor, there will be a set of instructions given by the manufacturer of the processor. For doing any useful work with the microprocessor, we have to write a program using these instructions and store them in a memory device external to the microprocessor.



3.12: Block diagram showing basic functional blocks of a microprocessor.

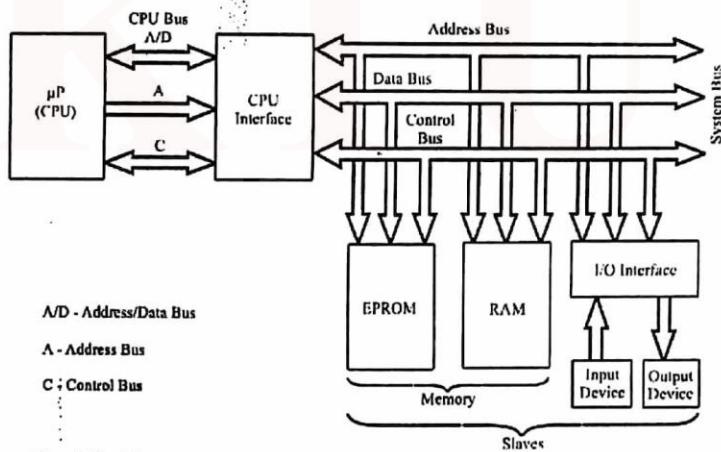


Fig. 13: Microprocessor-based system (organization of microcomputer).

**EPROM:** Store permanent programs and data.

**RAM :** Store temporary programs and data.

**Input device:** Used to enter the programs, data and to operate the system.

**Output device:** Used for examining the results.

**CPU Interface:** The speed of I/O devices doesn't match with the speed of the microprocessor since this is provided b/w the system bus of the I/O devices.

## Explanation of fig 1.2 Basic functional blocks of a Microprocessor

Programmable IC which is capable of performing arithmetic and logical operations.

### Basic functional Units

#### 1) ALU

Computational unit of the microprocessor, which performs arithmetic and logical operations on binary data.

#### 2) Flag Register

from ALU  
Various conditions of the result are stored as status bits called flags in the flag register.  
eg: If the result is '-ve', then '1' is stored in the sign flag and if the result is '+ve', then 0 is stored in the sign flag.

#### 3) Register Array (internal storage device) or internal memory

\* Input data for ALU  
\* Output data of ALU  
\* Any other binary info needed for processing

} are stored in the register array.  
4) PC / IP (Program Counter / Instructions pointer)  
Generates the "address of the instructions to be fetched from the memory" and send that address through the address bus to the memory.

### 5) Instruction Decoding Unit

The mly will send the instruction codes and data through the data bus.

⇒ Instruction codes are decoded by the decoding unit and indicate the ALU "what operation is to be perform".

### 6) Timing and Control Unit

Generate the necessary control signals for the internal and external operations of the microprocessor.

e.g: mly read signal, mly write signal  
I/O read signal, I/O write signal

### Explanation fig 1.3 (Microprocessor-based s/m)

In this s/m, the "microprocessor is the master" and all "other peripherals are slaves".

Mastee controls all the peripherals (memories, input device, o/p device and interfacing devices) and initiates all operations.

⇒ Buses are groups of lines that carry "data", "addresses" or "control signals":

i.e. Data Bus, Address Bus, Control Bus.

⇒ At any one time, communication takes place between the master and one of the slaves.

### Questions

- ① Compare microprocessor and microcontroller.
- ② Explain the basic functional blocks of a microprocessor with neat diagrams.
- ③ Explain the microprocessor based system or organization of microcomputer with neat diagrams

### Reference

Microprocessors and microcontrollers, 2e.

A. Nagoodkani Page No: 2.1, 1.5 to 1.7

### Introduction to 8086 processor

- Developed by INTEL in 1978.
- 3rd generation of processor.
- 20-bit Address bus
- $2^{20}$  i.e. 1 Mb mly space.
- 16-bit data bus
- Registers are 16 bit, but it can also access as 8 bit.
- 40 pin IC

## Title: Architecture of 8086 Microprocessor & Memory Organisation

8086 has a pipelined architecture. In pipelined architecture, the processor will have a no. of functional units. The execution time of the functional units are overlapped.

The architecture can be internally divided into 2 separate functional units:

- (1) Bus Interface Unit (2) Execution Unit

### (1) BUS INTERFACE UNIT

⇒ main blocks/units are Segment registers, IP, Instruction queue, Address Generation Unit, Bus control unit.

⇒ Basic functions.

- ✓ fetches instructions
- ✓ read data from M/I/O ports
- ✓ write data to M/I/O ports.

### Memory Organisation

8086 processor has a memory space of  $(2^{20})$  [i.e. 1MB, M/I/O space]. It can be divided into segments of 64KB each. The segments are Code Segment (CS), Data Segment (DS), Extra Segment (ES) & Stack Segment (SS). This allows the system designer to allocate separate areas for storing ~~variables~~ and data.

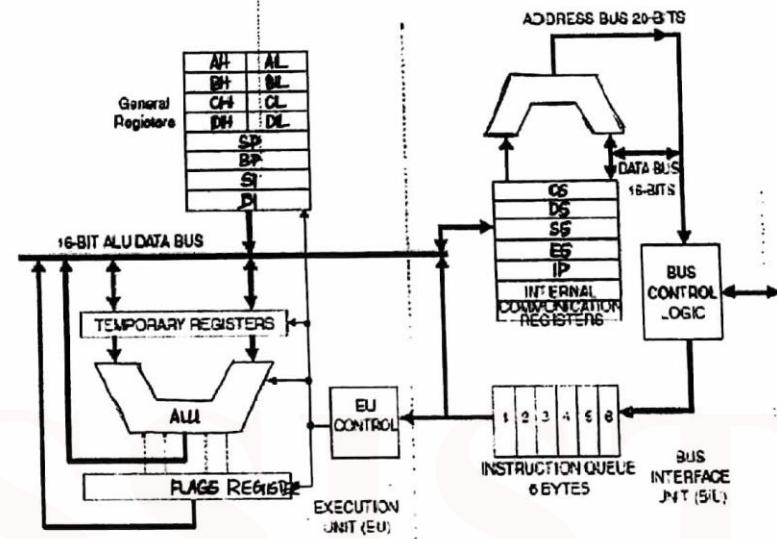
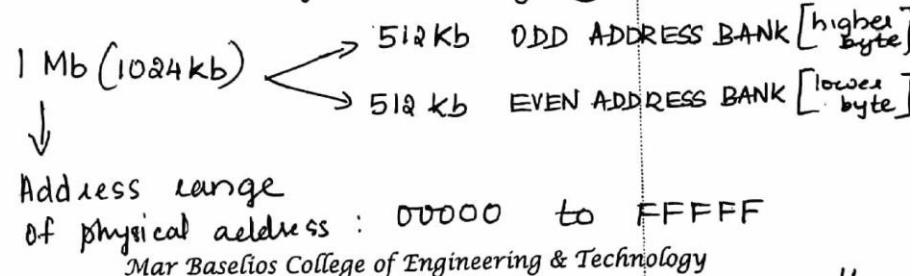
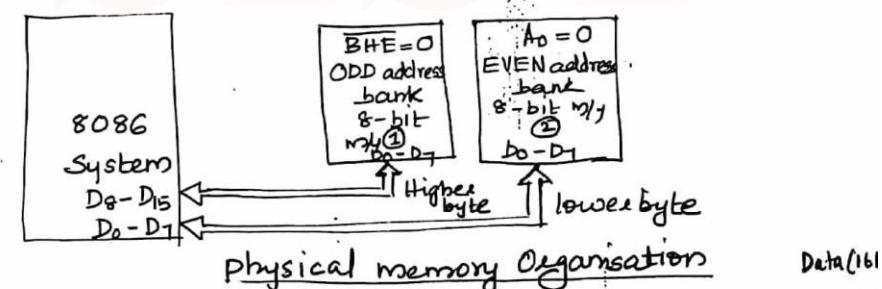


Fig 1.4 ARCHITECTURE OF 8086 MICROPROCESSOR



Status of A<sub>0</sub> & BHE during memory access

Memory Bank	Operand type	Status of A <sub>0</sub> , BHE	Datelines used for my access	No. of Bus cycles
Even	Byte	0 : 1	D <sub>0</sub> - D <sub>7</sub>	1
Odd	Byte	1 : 0	D <sub>8</sub> - D <sub>15</sub>	1
Every/odd	Word	0 : 0	D <sub>0</sub> - D <sub>15</sub>	1
Odd/Even	Word	1 : 0	D <sub>8</sub> - D <sub>15</sub>	First cycle
		0 : 1	D <sub>0</sub> - D <sub>7</sub>	Second cycle

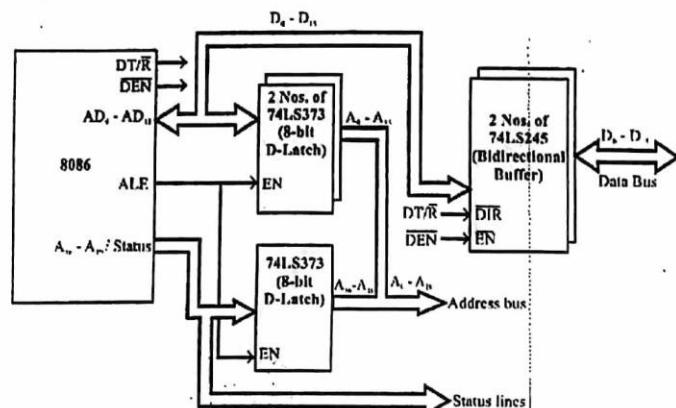


Fig. 1.5: Demultiplexing of address and data lines in an 8086 processor.

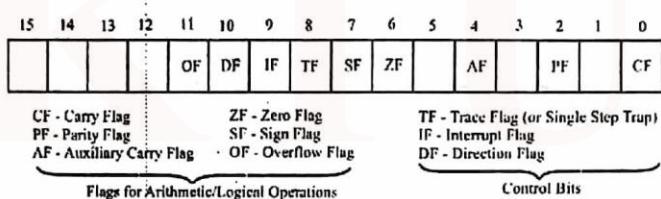
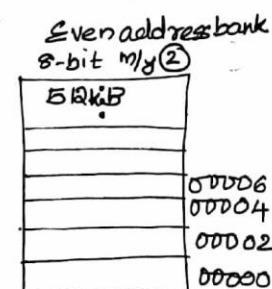


Fig. 1.6: Bit positions of various flags in the flag register of 8086.

The lower byte of a 16-bit data is stored at the first address i.e. 00000H in 8-bit m<sub>y</sub>(1).

The higher byte of a 16-bit data is stored in the next 00001 address i.e. 00001H in 8-bit m<sub>y</sub>(1).



- ✓ When Addressline 0 (A<sub>0</sub>) become low, then the even m<sub>y</sub> bank is enabled.
- ✓ When Bus High Enable (BHE) become low, then the odd m<sub>y</sub> bank is enabled.

### 1.a) Segment Registers (16-bit)

4 segment registers (CS, DS, ES, SS) holds four segment base address respectively.

Contents of segment registers are programmable. Hence the programmer can access the code and data in any part of the m<sub>y</sub> by changing the contents of the segment registers.

### 1.b) Instruction Pointer (IP)

IP holds the address of the next instruction to be fetched & gets incremented by two after every bus cycle.

### 1.c) Address Generation Unit (AGU)

Generates a 20-bit physical address from the segment base address and an offset or effective address.

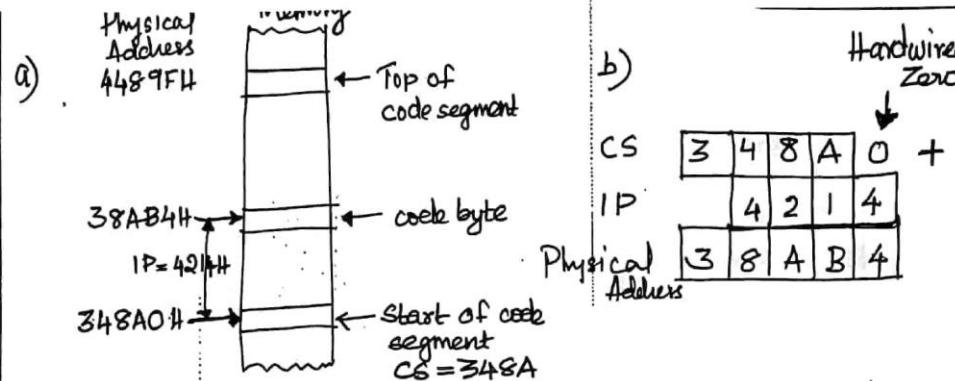


figure 1.7 Addition of IP to CS to produce the physical address of the code byte .

a) Diagrams   b) Computation

$$\text{Physical Memory Address} = \text{Base address} \times 10H + \text{offset}$$

Case-I

$$(CS) \times 10H + IP$$

Case-II

$$(DS) \times 10H + \text{Offset specified by the instruction}$$

Case-III

$$(ES) \times 10H + \text{offset specified by the instr.}$$

Case-IV

$$(SS) \times 10H + SP$$

#### 1. d) Bus Control logic Unit

Generate all the bus control signals such as ready, write, I/O read, I/O write etc....

#### 1.e) Instruction Queue

FIFO group of registers where size = 6 bytes. BIU fetches the instruction code from memory as the addresses generated from AGU and stores it in the queue. Then the Execution Unit (EU) fetches the instruction code from the queue.

#### 2. Execution Unit (EU)

Basic functions

- ✓ Decodes the fetched instruction from the queue
- ✓ then execute it.

EU contain \* Internal Control Systems

- \* ALU
- \* General purpose register
- \* Flag register.

##### 2.a) Internal Control Systems

Decodes the instruction & translates the instruction.

##### 2.b) ALU

16 bit ALU perform arithmetic & logical operations.

##### 2.c) General Purpose Registers (16-bit)

AX, BX, CX, DX, SI, DI, BP, SP

- ✓ AX, BX, CX, DX registers can also access as 8-bit registers such that AH, AL, BH, BL, CH, CL, DH, DL.
- ✓ G.P.R can be used for data storage. When they are not involved in any, special functions assigned to them.

### Registers

AX

#### Special functions

##### 16-bit accumulator.

(stores the 16-bit results of certain arithmetic and logical operations.)

AL

##### 8-bit accumulator

Base Register: hold the base value in based addressing mode to access my data. [Data Segment]

CX

Counter Register: hold the count value in SHIFT, ROTATE and LOOP instructions

DX

Data Register: hold data for multiplication and division operations.

SP

Stack Pointer: hold the offset address of Top of the stack my (TOS).

BP

Base Pointer: hold the base value in base addressing mode using the stack segment. (offset)

} SI

Source Index: hold the index value of the source operand.

} DI

Destination Index: hold the index value of the destination operand.

### 2.d) Flag Registers (fig: 1.6)

- ✓ 3 flags are known as control bits [DF, IF, TF]
  - used to control the processor operations.
- ✓ 6 flags for ALU operations. [CF, AF, PF, ZF, SF, OF]
  - used to indicate the status of the result of the ALU operations.

① CF (Carry Flag)  $\begin{cases} 1 \text{ (high)}; & \text{If there is a carry from the addition or borrow from the subtraction.} \\ 0; & \text{no carry} \end{cases}$

② AF (Auxiliary Carry Flag)  $\begin{cases} 1; & \text{If there is a carry from low nibble to high nibble of a 8 bit no.} \\ 0; & \text{no carry} \end{cases}$

③ OF (Overflow Flag)  $\begin{cases} 1; & \text{If there is an arithmetic overflow ie. If the size of the result exceeds the capacity of the destination location.} \\ 0; & \text{no overflow} \end{cases}$

④ PF (Parity Flag)  $\begin{cases} 1; & \text{even parity.} \\ 0; & \text{odd parity.} \end{cases}$

⑤ ZF (Zero Flag)  $\begin{cases} 1; & \text{result is zero.} \\ 0; & \text{result is not zero.} \end{cases}$

⑥ SF (Sign Flag)  $\begin{cases} 1; & \text{result is negative.} \\ 0; & \text{result is positive.} \end{cases}$

⑦ DF (Direction Flag)  $\begin{cases} 1; & \text{auto-decrement SI \& DI my pointer} \\ 0; & \text{auto-increment SI \& DI my pointer} \end{cases}$

## Title: 8051 Pinout and Signals

Vss (GND)	1	40	Vcc (+5V)
AD14	2	39	AD15
AD13	3	38	A16/S3
AD12	4	37	A17/S4
AD11	5	36	A18/S5
AD10	6	35	A19/S6
AD9	7	34	B8/E/S7
AD8	8	33	MN/MX
AD7	9	32	RD MODE
AD6	10	31	RQ/GTO HOLD
AD5	11	30	RQ/GTI HLDA
AD4	12	29	LOCK WR
AD3	13	28	S2 M/I/O
AD2	14	27	ST DT/R
AD1	15	26	SD DEN
ADO	16	25	QSO ALE
NMI	17	24	QS1 INTA
INTR	18	23	TEST
CLK	19	22	READY
Vss (GND)	20	21	RESET

8086

MIN MODE

MAX MODE

⑧ IF (Interrupt Flag)  $\leftrightarrow$  1; { to enable interrupt (during external interrupt request)  
                         0; to disable interrupt.

⑨ TF (Trap Flag)  $\leftrightarrow$  1; Step by step execution (8086 generate an internal interrupt after execution of each instruction)  
                         0; otherwise.

Questions

1) Compute the physical address

a) CS: F272 & IP: 0F2F      b) 2F72 : 2971

2) Draw and Explain the flag register of 8086.

3) With a neat diagram, explain the architecture of 8086.

4) Explain the physical memory organisation of 8086.

5) Explain the register organisation of 8086.

References

1. A. Nagarkar, Microprocessors and Microcontrollers Second Edition Page No: 1.29 - 1.32

2. Douglas V. Hall, SSSP Rao, Microprocessors and Interfacing, 2e.

✓ First 16-bit processor released by INTEL in 1978

✓ Designed using the HMOS Technology.

✓ Contains \* 29,000 transistors

\* 40 pin DIP

\* Requires single 5 Volt Supply

\* No internal clock circuit

\* External clock  $\rightarrow$  8284 clock generator generates max internal clock of 5MHz

✓ In addition to 20 address lines, 8086 uses a separate 16 bit address; so it generates 64kb I/O addresses.

To differentiate  
     I/O addresses      } using M/I/O  $\rightarrow$  high  
     I/O addresses      } M/I/O  $\rightarrow$  low  
                             Mar Baselios College of Engineering & Technology

<u>Pin</u>	<u>Function</u>
① & ⑧ D0	Ground
⑩ VCC	VCC
⑯ to ②	AD0 to AD <sub>14</sub>
⑩ 39	AD <sub>15</sub>
⑩ 38 to ⑤	A <sub>16</sub> /S <sub>3</sub> , A <sub>17</sub> /S <sub>4</sub> , A <sub>18</sub> /S <sub>5</sub> , A <sub>19</sub> /S <sub>6</sub>
⑨	Clock } 8284 clock generator chip used to generate the required clock.

fig. 1.5 explain the demultiplexing of Address/Data and Address/Status lines in 8086 processors.

For Demultiplexing address/data using 2 Nos of 8-bit 74LS373 D-latches. When ALE become high, two latches contain addresses A<sub>0</sub> to A<sub>7</sub> and A<sub>8</sub> to A<sub>15</sub> and third latch contains address A<sub>16</sub> to A<sub>19</sub>. Then the address given out through the 20 bit address bus.

When ALE become low, two bidirectional buffer (8 bit) contain data D<sub>0</sub> to D<sub>7</sub> & D<sub>8</sub> to D<sub>15</sub> and data given out through the 16 bit data bus.

Third latch contain status signals S<sub>3</sub> to S<sub>6</sub> given out through the status lines.

Mar Baselios College of Engineering & Technology

S <sub>4</sub>	S <sub>3</sub>	
0	0	ES
0	1	SS
1	0	CS
1	1	DS

Specifies the segment Register

S<sub>5</sub> → Indicates the status of an 8086 interrupt enable flag.

S<sub>6</sub> → low ; 8086 is the bus master.  
→ high ; local bus master gain the bus control from the processor.

⑩ 34 BHE / S<sub>7</sub> → 0 ; select odd address bank  
Bus High Enable

⑩ 16 A<sub>0</sub> (address line 0) → 0 ; select even address bank

⑩ 17 & ⑩ 18 → Interrupt pins  
17 → NMI (Non maskable Interrupt)  
18 → Interrupt Request

⑩ 33 MN/MX → high ; 8086 in minimum mode  
→ low ; 8086 in maximum mode

⑩ 32 RD → low ; m/y or I/O read  
Read stg → high ; no read

(28) TEST  
 ↓  
 low;  
 } stop execution  
 } of WAIT instruction.

Used to synchronize an external activity to the processor internal operation.

8086 will enter a WAIT state after execution of the WAIT instruction. It will resume execution only when TEST is made low.

(29) READY → normally 1 (high)

→ low; Introduce WAIT states  
 OR

Used by the memory or I/O device to get extra time for data transfer.

(30) RESET → high; system reset for atleast 4 clock cycles.

CS points to FFFF and DS, SS, ES, IP & flag registers are cleared & queue is emptied

8086 operates in 2 Modes

Min mode

- \* MN/M<sub>16</sub> → high
- \* S/m → uniprocessor s/m
- \* 8086 itself generates bus control signals.

Max mode

- \* MN/M<sub>16</sub> → low
- \* multiprocessor s/m
- \* Use External bus controller 8288 to generate bus control signals.

→ Signals assigned to pin 24 to 31 will be different for minimum and maximum mode. All other pins are common for min and max mode of operations.

### Min Mode

PIN

Function

(31) HOLD → high ; Request to processor from other bus master for getting bus control.

(30) HLDA → high ; Acknowledgement sig by the processor to the bus master after giving bus control.

(29) WR → low ; to write data to memory/ I/O port.

(28) M/I<sub>O</sub> → high ; memory access.  
 → low ; I/O address access.

(27) DT/R → high ; to transmit data.  
 → low ; to receive data.

(26) DEN → low ; to enable the external bi-directional data buffers.

(25) ALE → high ; address lines  
 → low ; data lines/status lines  
 { used to demultiplex address & datalines }

(24)  $\overline{\text{INTA}}$   $\Rightarrow$  an acknowledgement signal produced by the processor, when the interrupt request is accepted.

### Maximum Mode

- 31  $\overline{\text{RQ}}/\overline{\text{GTO}}$  } (Bus Request and Bus Grant)  
 30  $\overline{\text{RQ}}/\overline{\text{GTI}}$  } Request used by the local bus masters to force the processor in order to release the local bus at the end of the processor's current bus cycle.

$\overline{\text{GTO}} > \overline{\text{GTI}}$   
Priority

(25)  $\overline{\text{LOCK}}$   $\Rightarrow$  While executing an instruction prefixed by  $\overline{\text{LOCK}}$ , this pin is tied low in order to prevent other bus masters from gaining control of the system bus!

(26)  $\overline{\text{S2}}, \overline{\text{S1}}, \overline{\text{SO}}$   $\Rightarrow$  status signals (Used by 8288 bus controller in order to generate timing & control signals)

$\overline{\text{S2}}$	$\overline{\text{S1}}$	$\overline{\text{SO}}$	Machine cycle
0	0	0	Interrupt ack
0	0	1	Read I/O port
0	1	0	Write I/O port
0	1	1	Halt
1	0	0	Code access
1	0	1	Read my Write my
1	1	0	Write my

(25) & (24)  $\Rightarrow$  Instruction Queue Status  
 $\text{QSO}, \text{QSI}$  Used by external device, to track the internal status of the queue in 8086

$\text{QSI}$	$\text{QSO}$	Queue Operation
0	0	No operation
0	1	First byte of an opcode from queue.
1	0	Empty the queue
1	1	Subsequent byte from queue.

### Questions

- Explain the pins and signals of 8086.
- What are the functions of the TEST and READY input pins of 8086?
- Discuss the functions of the signals which are significant with respect to maximum mode in 8086.
- Discuss the functions of the signals which are significant with respect to minimum mode in 8086.

## MINIMUM MODE 8086 SYSTEM AND TIMINGS

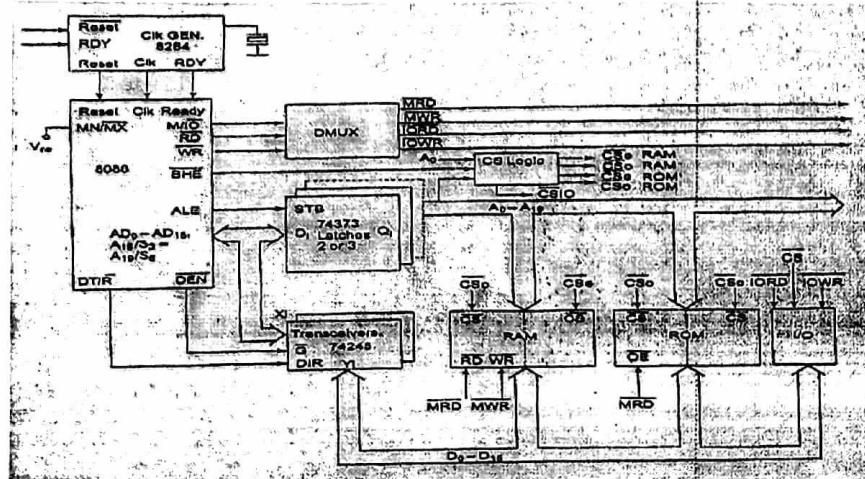


Fig 1.1. Minimum Mode 8086 System

In a minimum mode 8086 system, the microprocessor 8086 is operated in minimum mode by strapping its MN/MX\* pin to logic1.

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.

The 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 signal. They are controlled by two signals, namely, DEN\* and DT/R\*. The DEN\* signal indicates that the valid data is available on the data bus, while DT/R indicates the direction of data, i.e. from or to the processor.

The system contains memory for the monitor and users program storage. Usually, EPROMS are used for monitor storage, while RAMs for users program storage.

A system may contain I/O devices for communication with the processor as well as some special purpose I/O devices.

The clock generator generates the clock from the crystal oscillator and then shapes it and divides to make it more precise so that it can be used as an accurate timing reference for the system. The clock generator also synchronizes some external signals with the system clock.

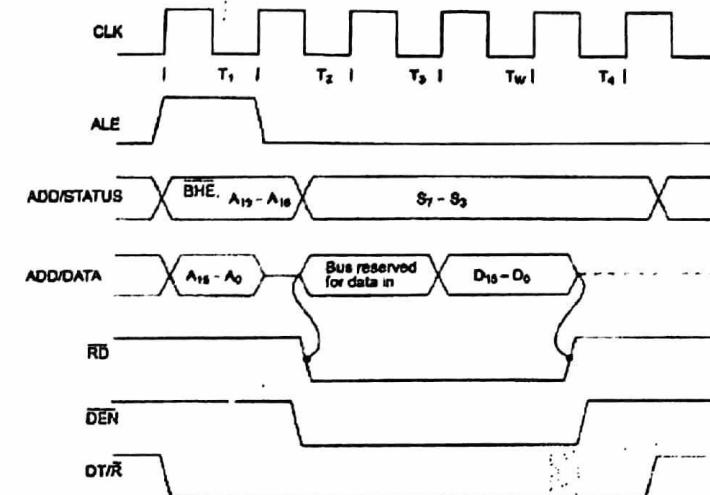


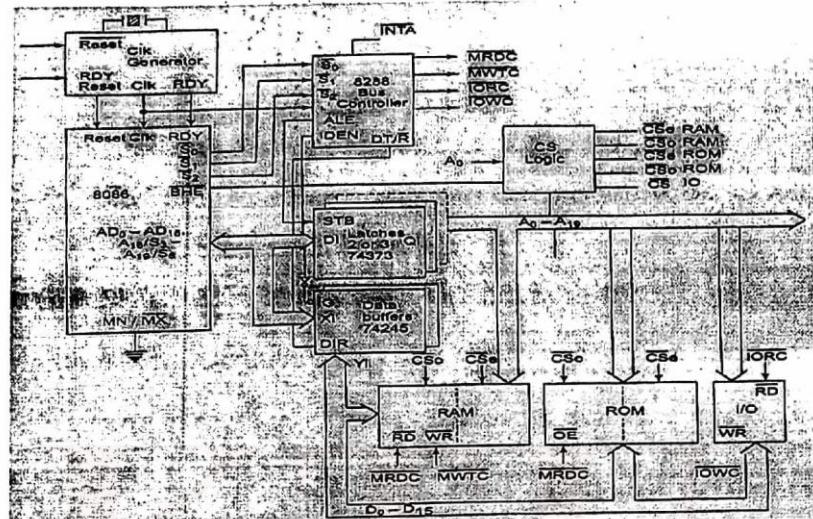
Fig. 2.4 Read cycle timing diagram for minimum mode

The read cycle begins in T<sub>1</sub> with the assertion of the Address latch Enable(ALE) high and M/I/O signal ( $M/I/O \rightarrow$  high; my address)  $\leftarrow$  low; I/O addresses). During the negative edge of this sig, the valid address is latched on the ~~data~~ bus.

At T<sub>2</sub> the address is removed from the ~~data~~ bus and is sent to the output.

Then the bus is triated. The read( $\overline{RD}$ ) signal is also activated in T<sub>2</sub>. i.e. become low that causes the addressed device to enable its data bus driven. After  $\overline{RD}$  goes low, the valid data is available on the data bus.

## MAXIMUM MODE 8086 SYSTEM



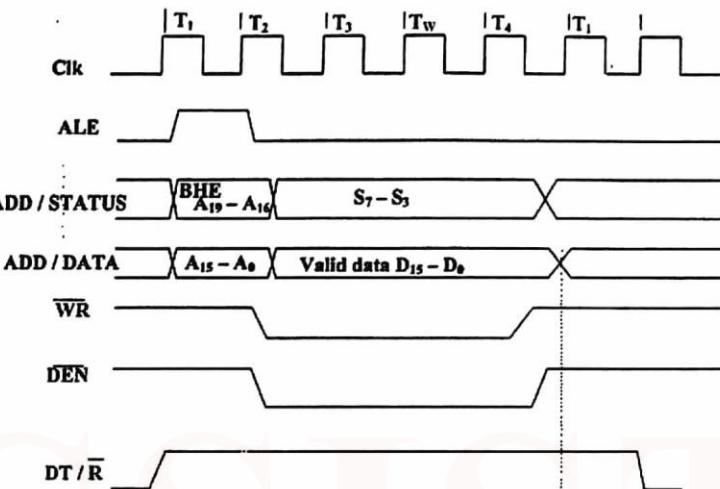
In the maximum mode, the 8086 is operated by strapping the MN/MX\* pin to ground.

In this mode, the processor derives the status signals S2\*, S1\* and S0\*. Another chip called bus controller derives the control signals using this status information.

In the maximum mode, there may be more than one microprocessor in the system configuration. The other components in the system are the same as in the minimum mode system.

The basic functions of the bus controller chip IC8288, is to derive control signals like RD\* and WR\* (for memory and I/O devices), DEN\*, DT/R\*, ALE, etc. using the information made available by the processor on the status lines. The bus controller chip has input lines S2\*, S1\* and S0\* and CLK. These inputs to 8288 are driven by the CPU. It derives the outputs ALE, DEN\*, DT/R\*, MWTC\*, AMWC\*, IORC\*, IOWC\* and 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.

IORC\*, 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 addressed port. The MRDC\*, MWTC\* are memory read command and memory write command signals respectively and may be used as memory read and write signals. All these command signals instruct the memory to accept or send data from or to the bus. For both of these write command signals, the advanced signals namely AIOWC\* and AMWTC\* are available. They also serve the same purpose, but are activated one clock cycle earlier than the IOWC\* and MWTC\* signals, respectively.



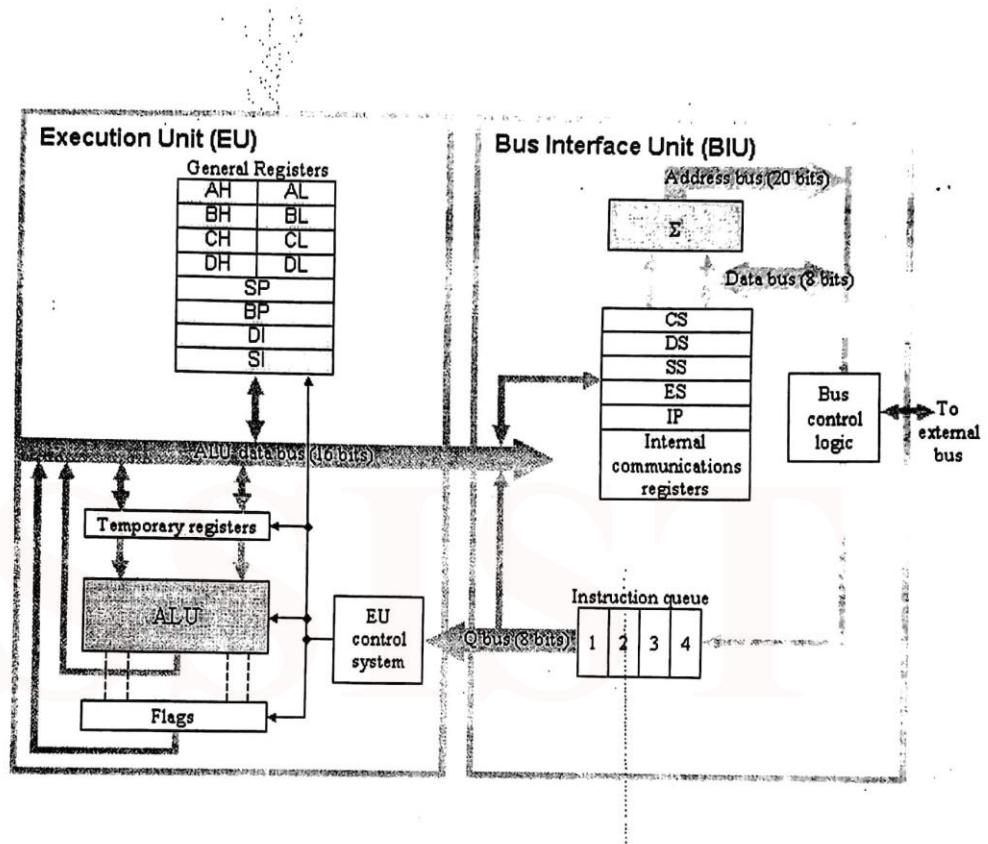
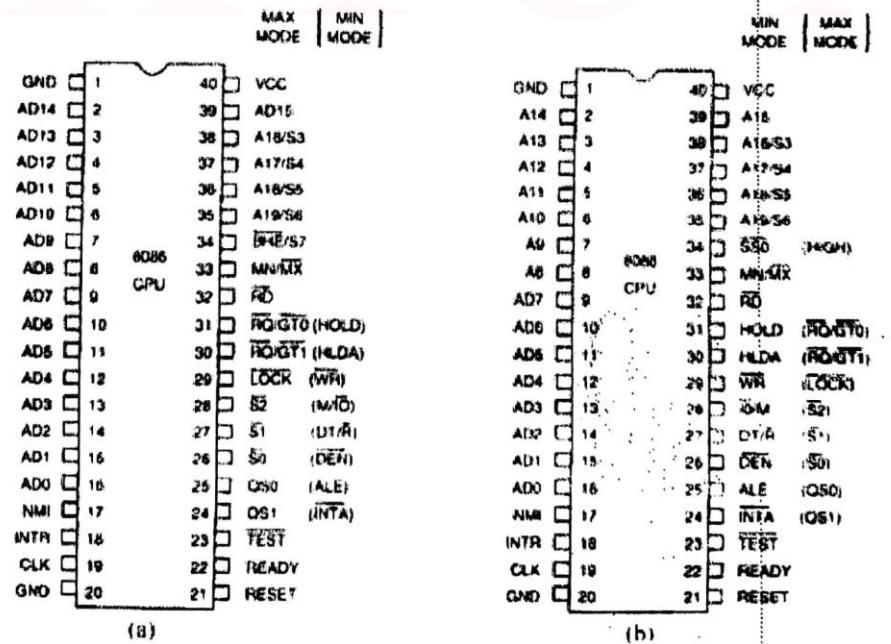
Write Cycle Timing Diagram for Minimum Mode



## Comparison between 8086 and 8088

<b>8086</b>	<b>8088</b>
*Access memory in bytes and words	*Access memory in bytes only
*16 bit data bus and 20 bit address bus	*8 bit data bus and 20 bit address bus
* <b>BHE</b> signal to access higher byte	* Data bus is 8bit wide, it does not have <b>BHE</b> signal.
* 6 byte instruction queue.	*4 byte instruction queue.
* <b>(M0)</b> for memory or I/O access.	* <b>OM</b> for memory or I/O access.
<p>*Pin no:34 is <b>BHE/S</b>. During T2, T3, T4 it carries S7.            In maximum mode 8087 monitors this pin to identify the CPU as 8086 or 8088 and accordingly sets the queue length to 4 or 6 bytes.</p>	<p>*Pin no. 34 is <b>SS0</b>. It acts as <b>S0</b> in the minimum mode of 8088. This signal is combined with <b>OM</b> and <b>(DT/R)</b> to decode the function of the current bus cycle.</p>

## 8086/8088 Pin Configuration



**END**