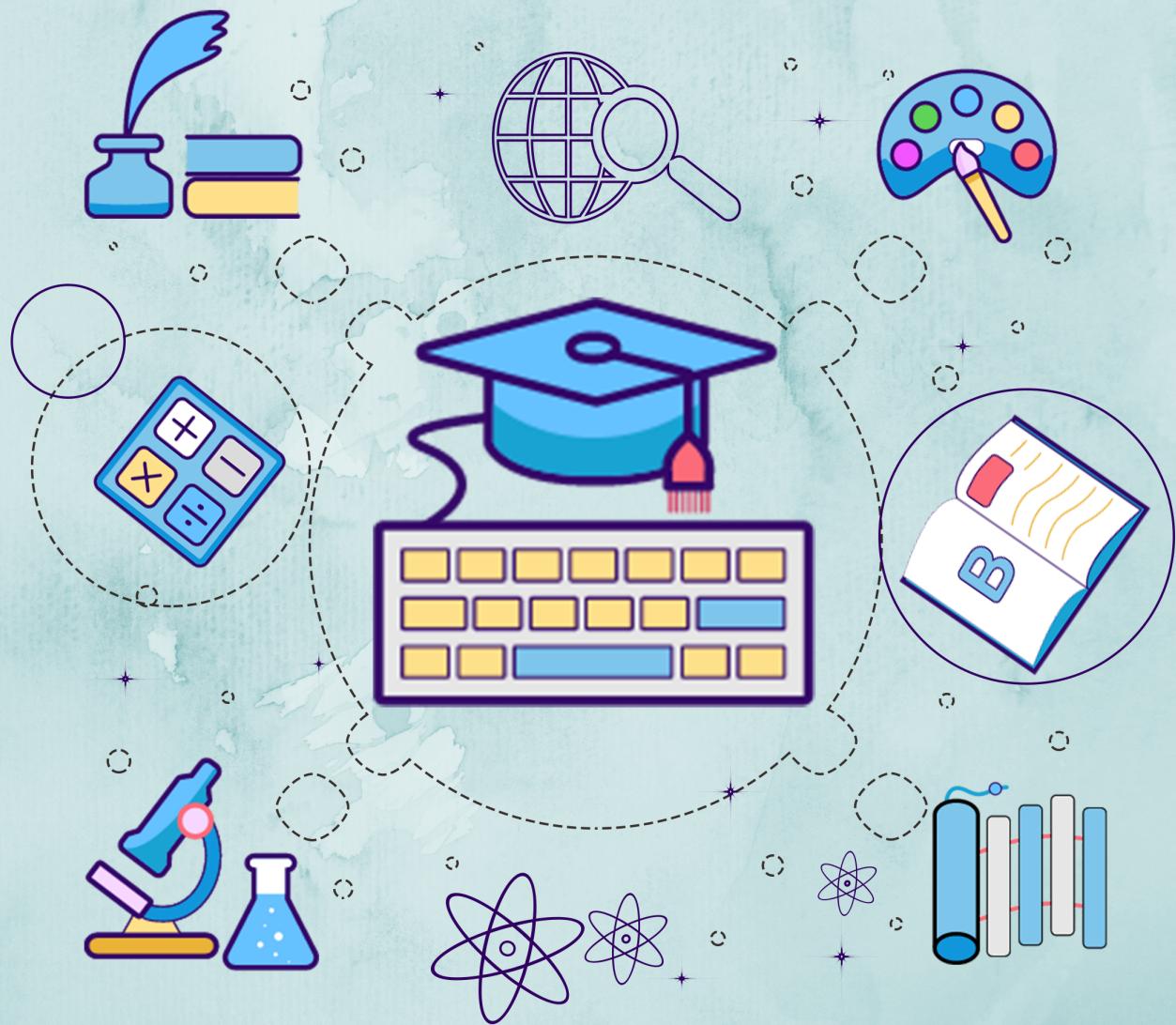


# Kerala Notes



**SYLLABUS | STUDY MATERIALS | TEXTBOOK**

**PDF | SOLVED QUESTION PAPERS**



## KTU STUDY MATERIALS

# MICROPROCESSORS AND MICROCONTROLLERS

### CST 307

## Module 1

#### Related Link :

- KTU S5 STUDY MATERIALS
- KTU S5 NOTES
- KTU S5 SYLLABUS
- KTU S5 TEXTBOOK PDF
- KTU S5 PREVIOUS YEAR  
SOLVED QUESTION PAPER

# **Microprocessors and Microcontrollers (CST 307) – 2019 Scheme**

## **Module 1:**

8085 microprocessor (-Basic Architecture only). 8086 microprocessor – Architecture and signals, Physical Memory organization, Minimum and maximum mode of 8086 system and timings. Comparison of 8086 and 8088. Machine language Instruction format.

### **Introduction to Microprocessors**

A microprocessor is a computer processor which incorporates the functions of a computer's central processing unit (CPU) on a single integrated circuit (IC), or at most a few integrated circuits. The microprocessor is a multipurpose, clock driven, register based, digital-integrated circuit which accepts binary data as input, processes it according to instructions stored in its memory, and provides results as output. Microprocessors contain both combinational logic and sequential digital logic. Microprocessors operate on numbers and symbols represented in the binary numeral system.

Generation of Microprocessors:

#### **INTEL 4004 (1971)**

- 4-bit microprocessor
- 4 KB main memory
- 45 instructions
- PMOS technology
- was first programmable device which was used in calculators

#### **INTEL 8008 (1972)**

- 8-bit version of 4004
- 16 KB main memory
- 48 instructions
- PMOS technology
- Slow

#### **INTEL 8086/8088**

Year of introduction 1978 for 8086 and 1979 for 8088

- 16-bit microprocessors
- Data bus width of 8086 is 16 bit and 8 bit for 8088
- 1 MB main memory
- 400 nanoseconds clock cycle time
- 6 byte instruction cache for 8086 and 4 byte for 8088

#### **INTEL 80286 (1983)**

- 16-bit high performance microprocessor with memory management & protection
- 16 MB main memory
- Few additional instructions to handle extra 15 MB

#### **Intel 8080 (1973)**

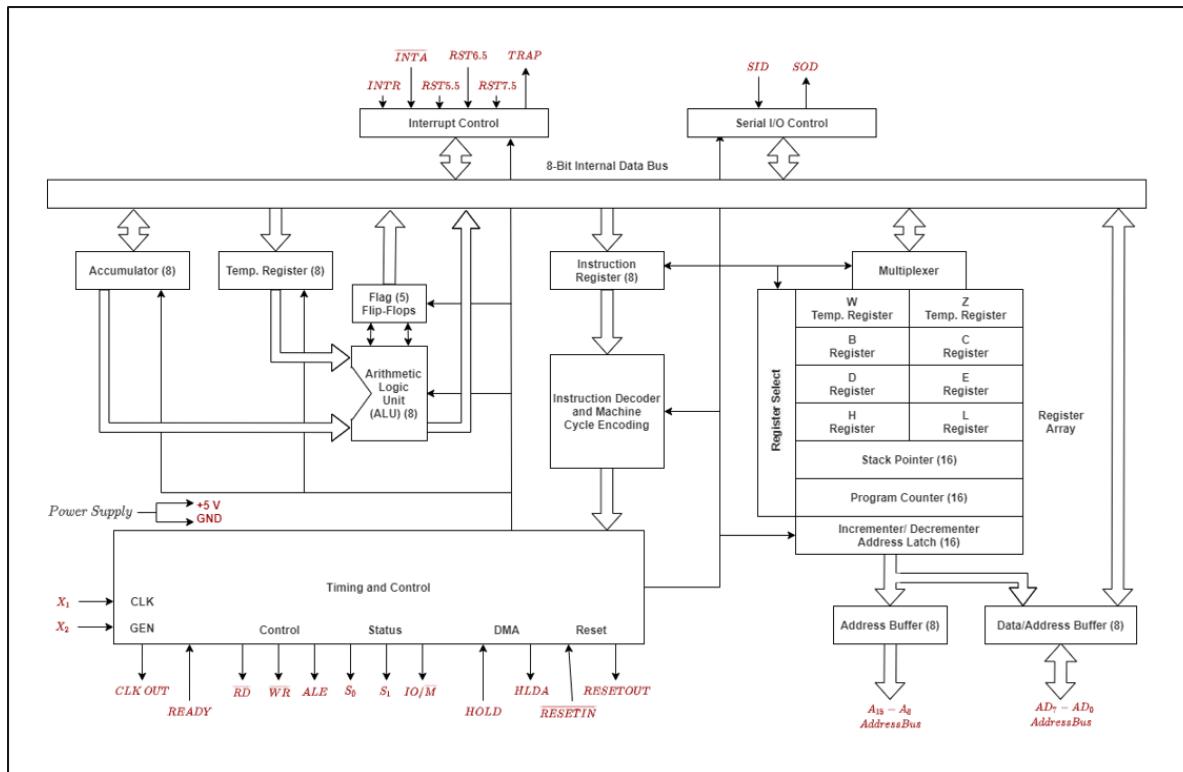
- 8-bit microprocessor
- 64 KB main memory
- 2 microseconds clock cycle time
- 500,000 instructions/sec
- 10X faster than 8008
- NMOS technology
- Drawback was that it needed three power supplies.
- Small computers (Microcomputers) were designed in mid in 1970's

#### **INTEL 80186 (1982)**

- 16-bit microprocessor-upgraded version of 8086
- 1 MB main memory
- Contained special hardware like programmable counters, interrupt controller etc.

- **Intel 80386** (1986)
- **Intel 80486** (1989)
- **Pentium** (1993)
- **Pentium pro**(1995)
- Latest is Intel i9 processor

## 8085 Architecture



- 8-bit data bus
- 16-bit address bus, which can address up to 64KB
- A 16-bit program counter
- A 16-bit stack pointer

- Six 8-bit registers arranged in pairs: BC, DE, HL
- Requires +5V supply to operate at 3.2 MHZ single phase clock

8085 consists of the following functional units:

- **Accumulator:** It is an 8-bit register used to perform arithmetic, logical, I/O & LOAD/STORE operations. It is connected to internal data bus & ALU.
- **Arithmetic and logic unit:** As the name suggests, it performs arithmetic and logical operations like Addition, Subtraction, AND, OR, etc. on 8-bit data.
- **General purpose register:** There are 6 general purpose registers in 8085 processor, i.e. B, C, D, E, H & L. Each register can hold 8-bit data. These registers can work in pair to hold 16-bit data and their pairing combination is like B-C, D-E & H-L
- **Program counter:** It is a 16-bit register used to store the memory address location of the next instruction to be executed. Microprocessor increments the program whenever an

instruction is being executed, so that the program counter points to the memory address of the next instruction that is going to be executed.

- **Stack pointer:** It is also a 16-bit register works like stack, which is always incremented/decremented by 2 during push & pop operations.
- **Temporary register:** It is an 8-bit register, which holds the temporary data of arithmetic and logical operations.

- **Flag register:** It is an 8-bit register having five 1-bit flip-flops, which holds either 0 or 1 depending upon the result stored in the accumulator.

These are the set of 5 flip-flops –

Sign (S), Zero (Z), Auxiliary Carry (AC), Parity (P), Carry (C)

Its bit position is shown in the following table –

D7	D6	D5	D4	D3	D2	D1	D0
S	Z		AC		P		CY

- **Instruction registers and decoder:** It is an 8-bit register. When an instruction is fetched from memory then it is stored in the Instruction register. Instruction decoder decodes the information present in the Instruction register.
- **Timing and control unit:** It provides timing and control signal to the microprocessor to perform operations. Following are the timing and control signals, which control external and internal circuits –
  - Control Signals: READY, RD', WR', ALE
  - Status Signals: S0, S1, IO/M'
  - DMA Signals: HOLD, HLDA
  - RESET Signals: RESET IN, RESET OUT
- **Interrupt control:** As the name suggests it controls the interrupts during a process. When a microprocessor is executing a main program and whenever an interrupt occurs, the microprocessor shifts the control from the main program to process the incoming request. After the request is completed, the control goes back to the main program.

There are 5 interrupt signals in 8085 microprocessors: INTR, RST 7.5, RST 6.5, RST 5.5, TRAP.

- **Serial Input/output control:** It controls the serial data communication by using these two instructions: SID (Serial input data) and SOD (Serial output data).
- **Address buffer and address-data buffer:** The content stored in the stack pointer and program counter is loaded into the address buffer and address-data buffer to communicate

with the CPU. The memory and I/O chips are connected to these buses; the CPU can exchange the desired data with the memory and I/O chips.

- **Address bus and data bus:** Data bus carries the data to be stored. It is bidirectional, whereas address bus carries the location to where it should be stored and it is unidirectional. It is used to transfer the data & Address I/O devices.

## 8086 Architecture

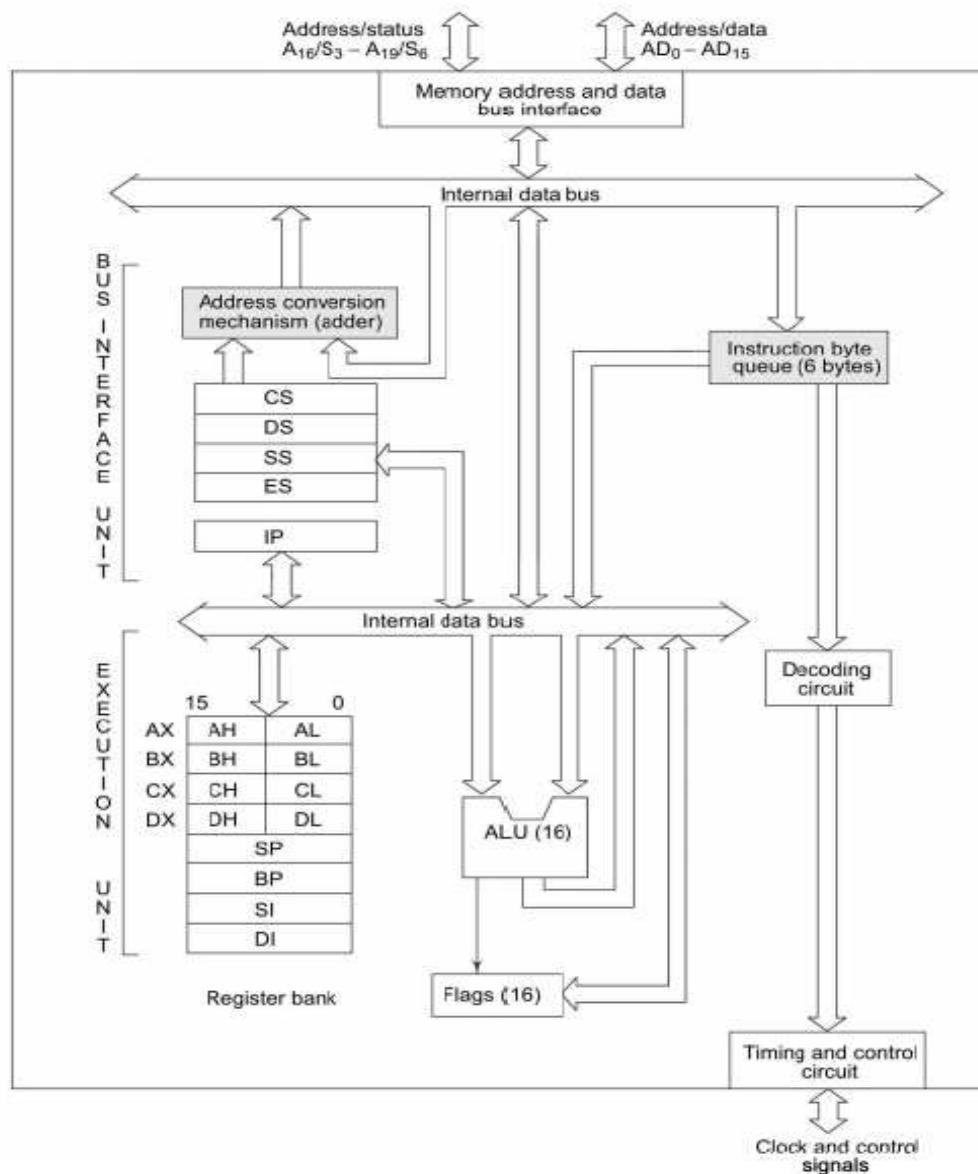


Fig. 1.2 8086 Architecture

The architecture of 8086 can be divided into 2 parts as follows:

- Bus Interface Unit (BIU)
- Execution Unit (EU)

### 1) Bus Interface Unit (BIU)

The Instruction Queue contains the set of instruction which is to be executed. To make the processing faster, the 8086 pre-fetches up to 6 instructions in advance and stores them in the

Instruction queue. So, whenever one instruction completes its execution, the control unit need not wait for the next instruction to be fetched and then brought for execution because this job is already done and the next instruction that is to be executed is ready in the Instruction queue.

## 2) Segment Registers

Each Segment register can work with 16 bits of binary data. There are 4 types of segment registers:

- CS: Code Segment Register
- DS: Data Segment Register
- SS: Stack Segment Register
- ES: Extra Segment Register

## 3) Instruction Pointer

The Instruction pointer contains the address of the next instruction that is to be executed.

## Execution Unit (EU)

### 1) Control Unit (CU)

All the Instructions are executed inside the Control Unit. It is the main component which is responsible for the processing of any processor.

### 2) Arithmetic Logic Unit (ALU)

All the Mathematical and Logical Operations are performed inside the ALU. So, if any instruction needs to perform such operation, the Control Unit handovers it to the ALU.

### 3) Flag Register

The flag Register is of 16-bit length which consists of 9 flags, and the rest 7 bits are of don't care cases.

- **Conditional Flags:** This flag represents the result of the last arithmetic or logical instruction executed. Conditional flags are:
  - Carry Flag
  - Auxiliary Flag
  - Parity Flag
  - Zero Flag
  - Sign Flag
  - Overflow Flag
- **Control Flags:** It controls the operations of the execution unit. Control flags are:
  - Trap Flag
  - Interrupt Flag
  - Direction Flag

## 4) General Purpose Registers

The General-Purpose Registers are used as containers for storing the values which may be required for executing the instructions. Each General Purpose Register consists of 16 bits. There are 4 types of General Purpose Registers:

- AX = [AH:AL]
- BX = [BH:BL]
- CX = [CH:CL]
- DX = [DH:DL]

## 5) Pointers and Index Registers

The 8086 contains following Pointers and Index Registers. (Here, IP is not mentioned because it is a part of the BIU)

- BP: Base Pointer
- SP: Stack Pointer
- SI: Source Index
- DI: Destination Index

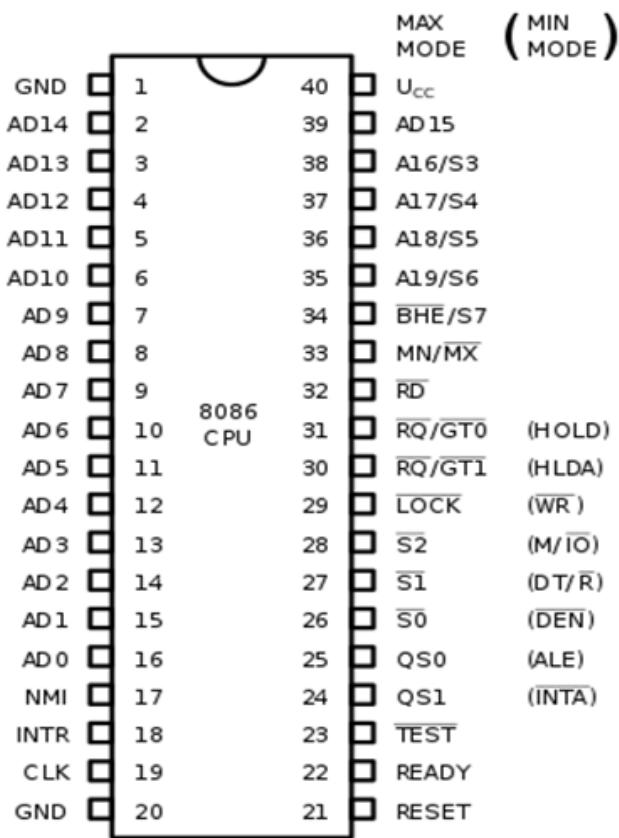
## 6) Operands

These may be used within the instructions.

## Difference Between 8085 and 8086 Microprocessors

<b>8085 Microprocessor</b>	<b>8086 Microprocessor</b>
It is an 8-bit microprocessor.	It is a 16-bit microprocessor.
It has a 16-bit address line.	It has a 20-bit address line.
It has a 8-bit data bus.	It has a 16-bit data bus.
The memory capacity is 64 KB.	The memory capacity is 1 MB.
The Clock speed of this microprocessor is 3 MHz	The Clock speed of this microprocessor varies between 5, 8 and 10 MHz for different versions.
It has five flags.	It has nine flags.
8085 microprocessor does not support memory segmentation.	8086 microprocessor supports memory segmentation.
It does not support pipelining.	It supports pipelining.
It is accumulator based processor.	It is general purpose register based processor.
It has no minimum or maximum mode.	It has minimum and maximum modes.
In 8085, only one processor is used.	In 8086, more than one processor is used. An additional external processor can also be employed.
It contains less number of transistors compare to 8086 microprocessor. It contains about 6500 transistor.	It contains more number of transistors compare to 8085 microprocessor. It contains about 29000 in size.
The cost of 8085 is low.	The cost of 8086 is high.

## 8086 Pin Diagram



**AD0-AD15 (Address Data Bus):** Bidirectional address/data lines. These are low order address bus. They are multiplexed with data.

When these lines are used to transmit memory address, the symbol A is used instead of AD, for example, A0- A15.

**A16 - A19 (Output):** High order address lines. These are multiplexed with status signals.

**A16/S3, A17/S4:** A16 and A17 are multiplexed with segment identifier signals S3 and S4.

**A18/S5:** A18 is multiplexed with interrupt status S5.

**A19/S6:** A19 is multiplexed with status signal S6.

**BHE/S7 (Output):** Bus High Enable/Status. During T1, it is low. It enables the data onto the most significant half of data bus, D8-D15. 8-bit

device connected to upper half of the data bus use BHE signal. It is multiplexed with status signal S7. S7 signal is available during T3 and T4.

**RD (Read):** For read operation. It is an output signal. It is active when LOW.

**Ready (Input):** The addressed memory or I/O sends acknowledgment through this pin. When HIGH, it denotes that the peripheral is ready to transfer data.

**RESET (Input):** System reset. The signal is active HIGH.

**CLK (input):** Clock 5, 8 or 10 MHz.

**INTR:** Interrupt Request.

**NMI (Input):** Non-maskable interrupt

**TEST (Input):** Wait for test control. When LOW the microprocessor continues execution otherwise waits.

**VCC:** Power supply +5V dc.

**GND:** Ground.

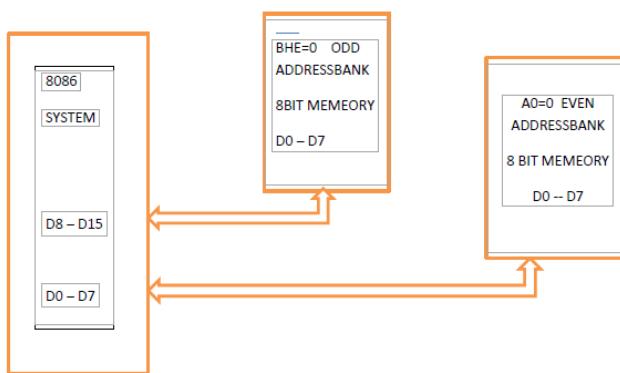
## Minimum and Maximum Mode in 8086

	<b>MIN MODE</b>	<b>MAX MODE</b>
1	It is a <b>uniprocessor mode</b> . 8086 is the only processor in the circuit.	It is a <b>multiprocessor mode</b> . Along with 8086, there can be other processors like 8087 and 8089 in the circuit.
2	Here <b>MN/ MX</b> is connected to <b>Vcc</b> .	Here <b>MN/ MX</b> is connected to <b>Ground</b> .
3	<b>ALE</b> for the latch is <b>given by 8086 itself</b> .	As there are multiple processors, <b>ALE</b> for the latch is <b>given by 8288 bus controller</b> .
4	<b>DEN</b> and <b>DT/ R</b> for the transreceivers are <b>given by 8086 itself</b> .	As there are multiple processors, <b>DEN</b> and <b>DT/ R</b> for the transreceivers is <b>given by 8288 bus controller</b> .
5	Direct control signals like <b>M/ IO</b> , <b>RD</b> and <b>WR</b> are produced by <b>8086 itself</b> .	Instead of control signals, all processors produce status signals <b>S<sub>2</sub></b> , <b>S<sub>1</sub></b> and <b>S<sub>0</sub></b> .
6	Control signals <b>M/ IO</b> , <b>RD</b> and <b>WR</b> are decoded by a 3:8 decoder IC 74138.	Status signals <b>S<sub>2</sub></b> , <b>S<sub>1</sub></b> and <b>S<sub>0</sub></b> require special decoding are decoded by 8288 bus controller.
7	<b>INTA</b> for interrupt acknowledgement is produced by <b>8086</b> .	<b>INTA</b> for interrupt acknowledgement is produced by <b>8288 Bus Controller</b> .
8	<b>Bus request are grant</b> is handled using <b>HOLD</b> and <b>HLDA</b> signals.	<b>Bus request are grant</b> is handled using <b>RQ / GT</b> signals.
9	Since <b>74138</b> does not independently generate any signals, it <b>does not need a CLK</b> .	Since <b>8288</b> independently generates control signals, it <b>needs a CLK from 8284 clock generator</b> .
10	The circuit is <b>simpler but does not support multiprocessing</b> .	The circuit is <b>more complex but supports multiprocessing</b> .

## Physical Memory Organization

- In 8086 based system, 1MB memory is physically organized as an odd bank and an even bank, each of 512kbytes, addressed in parallel by processor.
- Byte data with an even bank address is transferred on D7-D0, while the byte data with odd address is transferred on D15-D8 bus lines.
- The processor provides two enable signals, BHE and A0 for selection of either even or odd or both the banks.
- Commercially available memory chips are only 1 Byte size that is they can store only one byte in one memory location.
- To store 16bit data, two successive memory locations are used and the lower byte of 16 bit data can be stored in first memory location while the second byte is stored in next location.

- In a 16 read or write operation both of these bytes will be read or written in a single



- Bits D0-D7 of a 16 bit data will be transferred over D0 - D7 (lower byte)of 16 bit data bus to/from 8bit memory.
- Bits D8-D15 of the 16 bit data will be transferred over D8 - D15 (higher byte)of the 16 bit data bus to/from 8 bit memory.
- The lower byte of a 16 bit data is stored at the first address of the map 00000H and is to be transferred over D0-D7 of the microprocessor bus.so 00000H must be in 8 bit memory.
- Higher byte of the 16 bit data is stored in the next address 00001H is to be transferred over D8-D15 of the microprocessor bus.so the address 00001H must be in 8 bit memory.
- The complete memory map of 8086 is filled with 16 bit data, all the lower bytes(D0-D7)will be stored in the 8 bit odd memory bank and all the higher bytes(D8-D15) will be stored in 8 bit even memory bank .
- Thus the complete memory map of 8086 system is divided into even and odd address memory banks.
- If 8086 transfer a 16 bit data to/from memory, both of these banks must be selected for 16 bit operation.
- To maintain the upward compatibility with 8085, 8088 must be able to implement 8 bit operations.
- The first 8 bit operation with an even memory bank and second one is 8 bit operation with odd memory bank.
- The two signals A0 and BHE solve the problem of selection of appropriate memory banks.
- Certain memory locations are preserved for specific cpu operations. The locations from FFFF0H to FFFFFH are reserved for operations including jump to initialization program and I/O processor initialization.
- The location 00000H to 003FFH are reserved for interrupt vector table.
- The interrupt structure provides space for a total of 256 interrupt vectors(1KB locations).

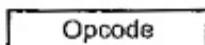
## Difference Between 8085, 8086 and 8088 Microprocessors

<b>8085</b>	<b>8086</b>	<b>8088</b>
8085 is an 8 bit microprocessor.	8086 is a 16 bit microprocessor.	8088 is a 16 bit microprocessor.
It has 8 bit data bus.	It has 16 bit data bus.	It has 8 bit data bus.
It has 8 bit ALU.	It has 16 bit ALU.	It has 16 bit ALU.
8085 does not require memory banking as it has an 8 bit data bus.	8086 requires memory banking to transfer 16 bit data at a time.	8088 does not require memory banking as it has an 8 bit data bus.
8085 performs slower memory operations as it can transfer only 8 bits in one cycle.	8086 performs faster memory operations as it can transfer 16 bits in one cycle.	8088 performs slower memory operations as it can transfer only 8 bits in one cycle.
8085 does not support pipeline architecture.	8086 supports pipeline architecture.	8088 supports pipeline architecture.
8085 has no pre-fetch queue as it does not support pipelining.	8086 has a 6 byte pre-fetch queue for pipelining.	8088 has a 4 byte pre-fetch queue for pipelining.
8085 has an IO/ pin to differentiate between memory and I/O operations.	8086 has an M/ pin to differentiate between memory and I/O operations.	8088 has an IO/ pin to differentiate between memory and I/O operations.

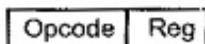
## Instruction Format

The 8086 Instruction Format vary from 1 to 6 bytes in length. Fig. 6.8 shows the instruction formats for 1 to 6 bytes instructions. As shown in the Fig. 6.8, displacements and operands may be either 8-bits or 16-bits long depending on the instruction. The opcode and the addressing mode is specified using first two bytes of an instruction.

One byte instruction implied operands



One byte instruction register mode



Register to register



Register to/ from memory with no displacement



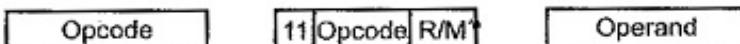
Register to/ from memory with displacement ( 8-bit )



Register to/ from memory with displacement ( 16-bit )



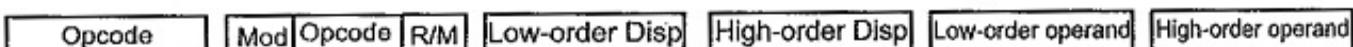
Immediate operand to register ( 8-bit )



Immediate operand to register ( 16-bit )



Immediate operand to memory with 16-bit displacement



**Fig. 6.8 Sample 8086 instruction formats**

The opcode/addressing mode byte(s) may be followed by :

- **No additional byte**
- **Two byte EA (For direct addressing only).**
- **One or two byte displacement**
- **One or two byte immediate operand**
- **One or two byte displacement followed by a one or two byte immediate operand**
- **Two byte displacement and a two byte segment address (for direct intersegment addressing only).**

Most of the opcodes in 8086 has a special 1-bit indicates. They are :

**W-bit :** Some instructions of 8086 can operate on byte or a word. The W-bit in the opcode of such instruction specify whether instruction is a byte instruction ( $W = 0$ ) or a word instruction ( $W = 1$ ).

**D-bit :** The D-bit in the opcode of the instruction indicates that the register specified within the instruction is a source register ( $D = 0$ ) or destination register ( $D = 1$ ).

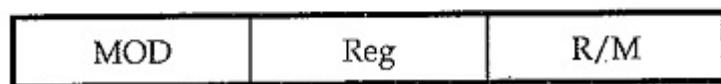
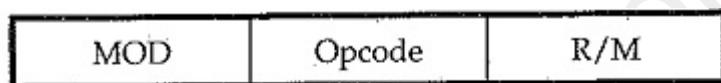
**S-bit :** An 8-bit 2's complement number can be extended to a 16-bit 2's complement number by making all of the bits in the higher-order byte equal the most significant bit in the low order byte. This is known as sign extension. The S-bit along with the W-bit indicate :

S	W	Operation
0	0	8-bit operation
0	1	16-bit operation with 16-bit immediate operand
1	0	
1	1	16-bit operation with a sign extended 8-bit immediate operand

**V-bit :** V-bit decides the number of shifts for rotate and shift instructions. If V = 0, then count = 1; if V = 1, the count is in CL register. For example, if V = 1 and CL = 2 then shift or rotate instruction shifts or rotates 2-bits

**Z-bit :** It is used for string primitives such as REP for comparison with ZF Flag. (Refer Appendix A for instruction formats)

As seen from the Fig. 6.8 if an instruction has two opcode/addressing mode bytes, then the second byte is of one of the following two forms .



where Mod, Reg and R/M fields specify operand as described in the following tables.

Mode	Displacement
0 0	Disp = 0 Low order and High order displacement are absent
0 1	Only Low order displacement is present with sign extended to 16-bits.
1 0	Both Low-order and High-order displacements are present.
1 1	r/m field is treated as a 'Reg' field.

**Table 6.2 'Mod' field assignments**

Word Operand (W = 1)		Byte Operand (W = 0)		Segment	
0 0 0	AX	0 0 0	AL	0 0	ES
0 0 1	CX	0 0 1	CL	0 1	CS
0 1 0	DX	0 1 0	DL	1 0	SS
0 1 1	BX	0 1 1	BL	1 1	DS
1 0 0	SP	1 0 0	AH		
1 0 1	BP	1 0 1	CH		
1 1 0	SI	1 1 0	DH		
1 1 1	DI	1 1 1	BH		

Table 6.3 'Reg' field assignment

R/M	Operand Address
0 0 0	$EA = (BX) + (SI) + \text{Displacement}$
0 0 1	$EA = (BX) + (DI) + \text{Displacement}$
0 1 0	$EA = (BP) + (SI) + \text{Displacement}$
0 1 1	$EA = (BP) + (DI) + \text{Displacement}$
1 0 0	$EA = (SI) + \text{Displacement}$
1 0 1	$EA = (DI) + \text{Displacement}$
1 1 0	$EA = (BP) + \text{Displacement}$
1 1 1	$EA = (BX) + \text{Displacement}$

Table 6.4 'R/M' field assignment