Microprocessor

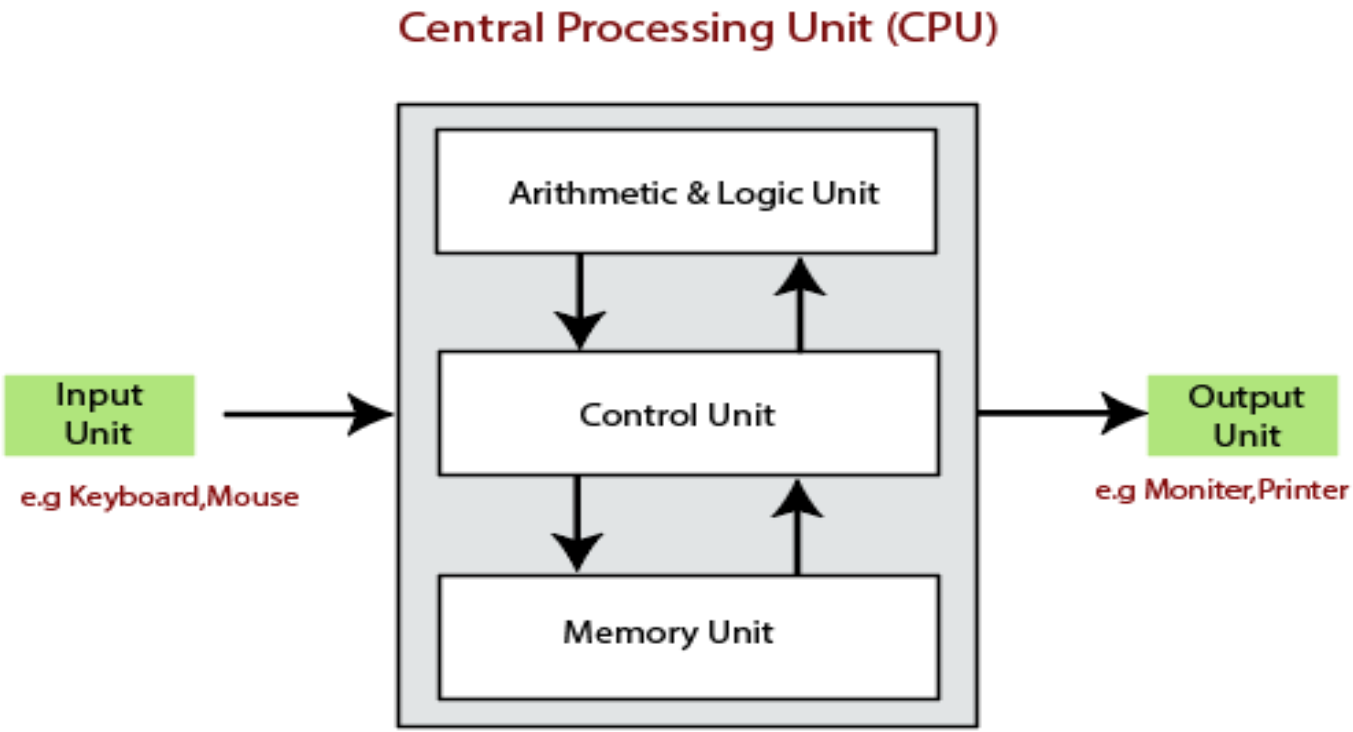
Basic Block Diagram of Computer

Lecture 01

Subscribe Now

By Vimal Sir

➢ Basic Block Diagram Of Computer



Input Unit

All the data received by the computer goes through the input unit. The input unit comprises different devices like a mouse, keyboard, scanner, etc. In other words, each of these devices acts as a mediator between the users and the computer.

CPU – Central Processing Unit

CPU, is the brain of the computer. It works the same way a human brain works. As the brain controls all human activities, similarly the CPU controls all the tasks.

Now the CPU comprises of two units, namely –

ALU (Arithmetic Logic Unit) and CU (Control Unit).

ALU – Arithmetic Logic Unit

The Arithmetic Logic Unit is made of two terms, arithmetic and logic. There are two primary functions that this unit performs.

1. Data is inserted through the input unit into the primary memory. Performs the basic arithmetical operation on it. Like addition, subtraction, multiplication, and division. It performs all sorts of calculations required on the data. Then sends back data to the storage.

2. The unit is also responsible for performing logical operations like AND, OR, Equal to, Less than, etc. In addition to this it conducts merging, sorting, and selection of the given data.

CU – Control Unit

The control unit as the name suggests is the controller of all the activities/tasks and operations. All this is performed inside the computer.

Memory Unit

All the data that has to be processed or has been processed is stored in the memory unit. The memory unit acts as a hub of all the data. It transmits it to the required part of the computer whenever necessary.

Output Unit

All the information sent to the computer once processed is received by the user through the output unit. Devices like printers, monitors, projectors, etc. all come under the output unit.

THANK YOU For Watching

SUBSCRIBE
LIKE
SHARE
COMMENT

SUBSCRIBE
LIKE
SHARE
COMMENT

Microprocessor

SUCCESS CLASSES BIJNOR
Join us
For Smart Study

What is Microprocessor

Evaluation of Microprocessor

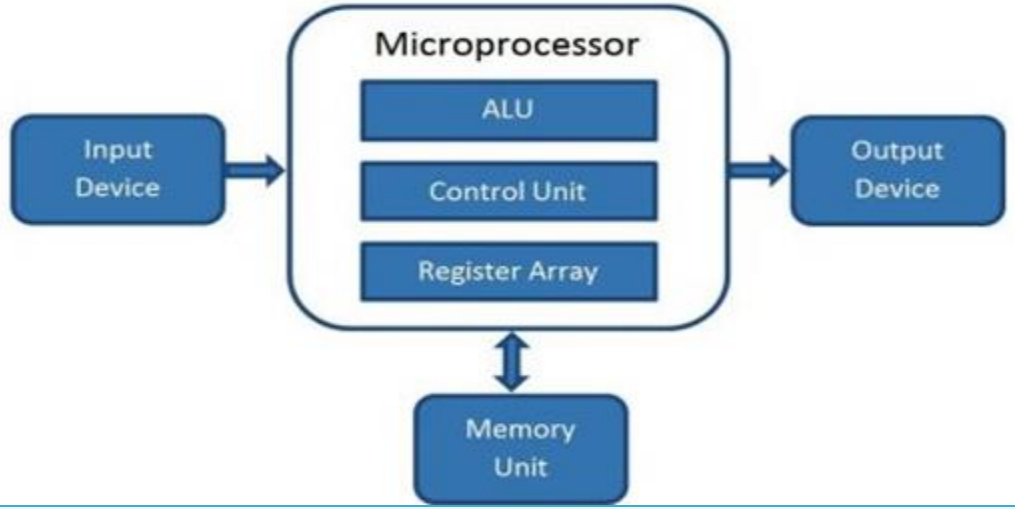
Lecture 02

Subscribe Now

By Vimal Sir

Microprocessor

A microprocessor is basically the brain of the computer. We can also call it simply a processor or CPU. Furthermore, a microprocessor is basically a computer processor that is mounted on a single IC (Integrated Circuit). It means that all the functions of the processor are included on a single chip.



Evolution of Microprocessor

Name	Year	Transistors	Clock speed	Data width
8080	1974	6,000	2 MHz	8 bits
8085	1976	6,500	5 MHz	8 bits
8086	1978	29,000	5 MHz	16 bits
8088	1979	29,000	5 MHz	8 bits
80286	1982	134,000	6 MHz	16 bits
80386	1985	275,000	16 MHz	32 bits
80486	1989	1,200,000	25 MHz	32 bits
Pentium	1993	3,100,000	60 MHz	32/64 bits
Pentium II	1997	7,500,000	233 MHz	64 bits
Pentium III	1999	9,500,000	450 MHz	64 bits
Pentium IV	2000	42,000,000	1.5 GHz	64 bits
Pentium IV "Prescott"	2004	125,000,000	3.6 GHz	64 bits
Intel Core 2	2006	291 million	3 GHz	64 bits
Pentium Dual Core	2007	167 million	2.93 GHz	64 bits
Intel 64 Nchalem	2009	781 million	3.33 GHz	64 bits

Microprocessor Operation

Microprocessor follows the following cycles or operations during program execution.

1. Fetch Cycle
2. Decoding Cycle
3. Execute Cycle

Features of 8085 Microprocessor

- 8 bit microprocessor(8085 microprocessor can read or write or perform arithmetic and logical operations on 8-bit data at time)
- It has 8 data lines and 16 address lines hence capacity is 216 = 64 kB of memory
- Cock frequency is 3 MHz
- It requires +5V power supply.
- It is a single chip NMOS device implemented with 6200 transistors.
- It provides 74 instructions with five addressing modes.
- It provides 5 hardware interrupt and 8 software interrupts.

THANK YOU For Watching

SUBSCRIBE
LIKE
SHARE
COMMENT

SUBSCRIBE
LIKE
SHARE
COMMENT

Microprocessor

SUCCESS CLASSES BIJNOR
Join us
For Smart Study

Architecture of 8085 Microprocessor

Lecture 03

By Vimal Sir

Subscribe Now

❖ 8085 Microprocessor Architecture

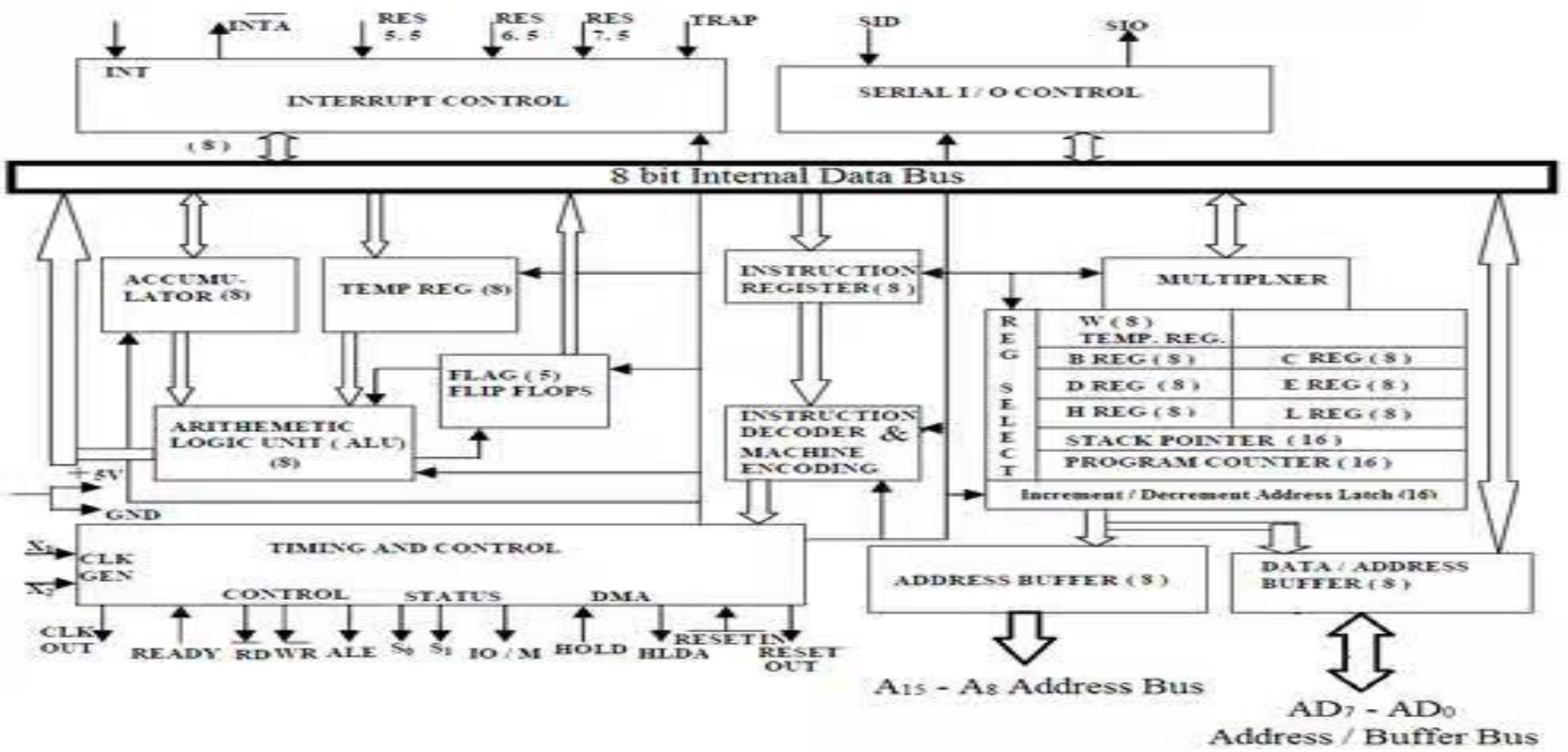
- The architecture of 8085 consists of three main sections, ALU (Arithmetic and Logical Unit), timing and control unit and Registers.

➢ Arithmetic and Logic Unit (ALU)

The ALU performs numerical and logical operations. ALU performs the following arithmetic and logical operations.

Addition, Subtraction
Logical AND, Logical OR, Logical Ex - OR
Complement (logical NOT)
Increment, Decrement
Left shift, Right shift etc.

ALU includes the accumulator, the temporary register, the arithmetic and logic circuits and flags. It always stores the results of operations in an Accumulator.



➢ Timing & Control Unit

- It generates timing and control signals, which are necessary for the execution of instructions.
- It controls data flow between the CPU and peripherals (including memory).
- It provides status, control and timing signals required to operate memory and I/O devices.

➢ **Program Counter (PC):** This 16-bit register deals with sequencing the execution of instructions. The microprocessor uses this register to sequence the execution of the instructions. The function of the program counter is to point to the memory address from which the next byte is to be fetched.

➢ **Stack Pointer (SP):** The stack pointer is a 16-bit register used as a memory pointer. It points to a memory location in read-write memory called the stack.

- **Instruction Register/Decoder:** Temporary store for the current instructions of a program. The latest instruction was sent here from memory before execution. The decoder then takes instruction and decodes or interprets the instruction. Decoded instructions are then passed to the next stage.

- **8085 System Bus:** The microprocessor communicates with memory and other devices (input and output) using three buses: Address Bus, Data Bus and Control Bus.

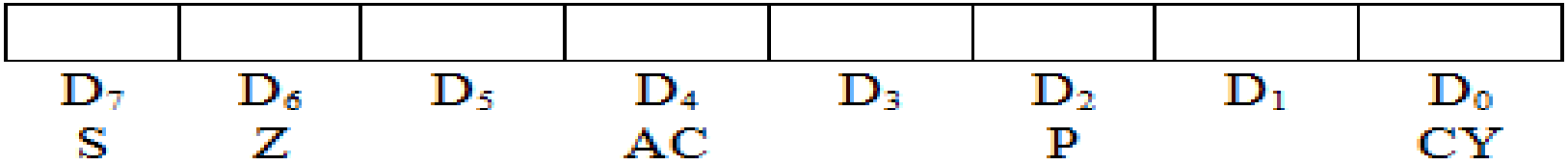
- **Address Bus:** The Address bus consists of 16 wires. The address bus's size determines the memory size, which can be used. To communicate with memory, the microprocessor sends an address on the address bus to the memory. The address bus is unidirectional, *i.e.,* numbers are only sent from the microprocessor to memory.
- **Data Bus:** Bus is bidirectional. The size of the data bus determines what arithmetic can be done. The data bus also carries instructions from memory to the microprocessor.

  Memory size = 2^A x D were, A denotes the address lines, and D denotes the data lines.
- **Control Bus:** Control buses are various lines that have specific functions for coordinating and controlling µP operations. The control bus carries control signals partly unidirectional and partly bidirectional. Control signals are things like reading or writing.

- **General Purpose Registers:** µP requires extra registers for versatility. It can be used to store additional data during a program.
- **Registers:** 8085 has six general-purpose registers to store 8-bit data, identified as B, C, D, E, H and L. They can be combined as register pairs BC, DE and HL to perform 16-bit operations.
- **Accumulator:** The accumulator is an 8-bit register included as a part of the Arithmetic Logic Unit (ALU). This register is used to store 8-bit data and to perform arithmetic and logical operations. The result of an operation is stored in the accumulator.
- **Flag Register:** The ALU includes five flip-flops. They are called Zero (Z), Carry (CY), Sign (S), Parity (P) and Auxiliary Carry (AC) flags. The microprocessor uses these flags to test data conditions. The software instructions test the flags' conditions (set or reset).

➢ Bit Position of Various Flags in Flag Register of 8085

The combination of the flag register and the accumulator is called Program Status Word (PSW), and PSW is the 16-bit unit for stack operation.



THANK YOU For Watching

SUBSCRIBE
LIKE
SHARE
COMMENT

SUBSCRIBE
LIKE
SHARE
COMMENT

Microprocessor

Pin Diagram

of

8085 Microprocessor

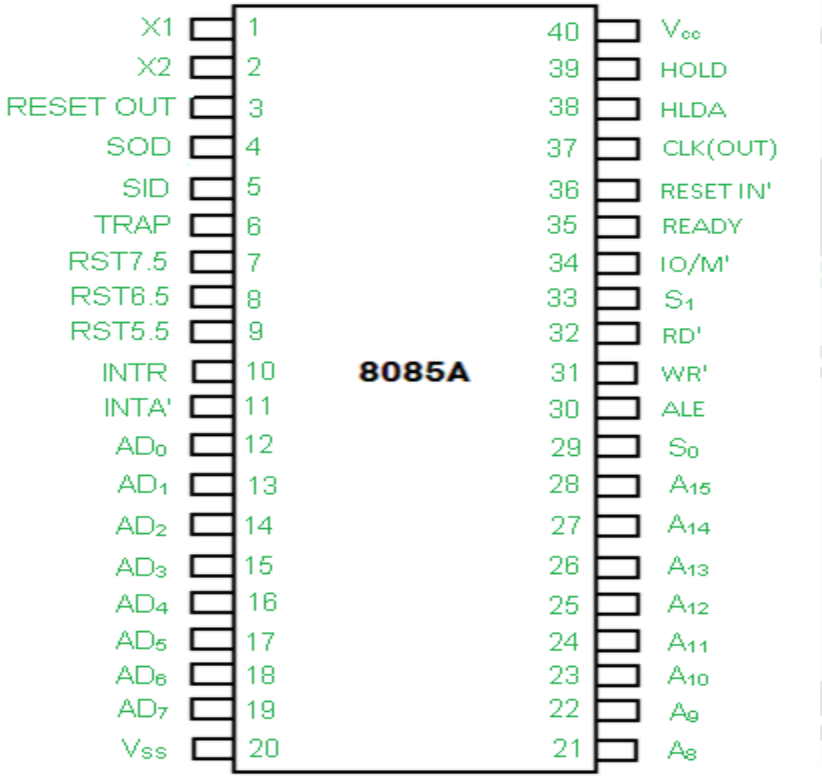
Subscribe Now

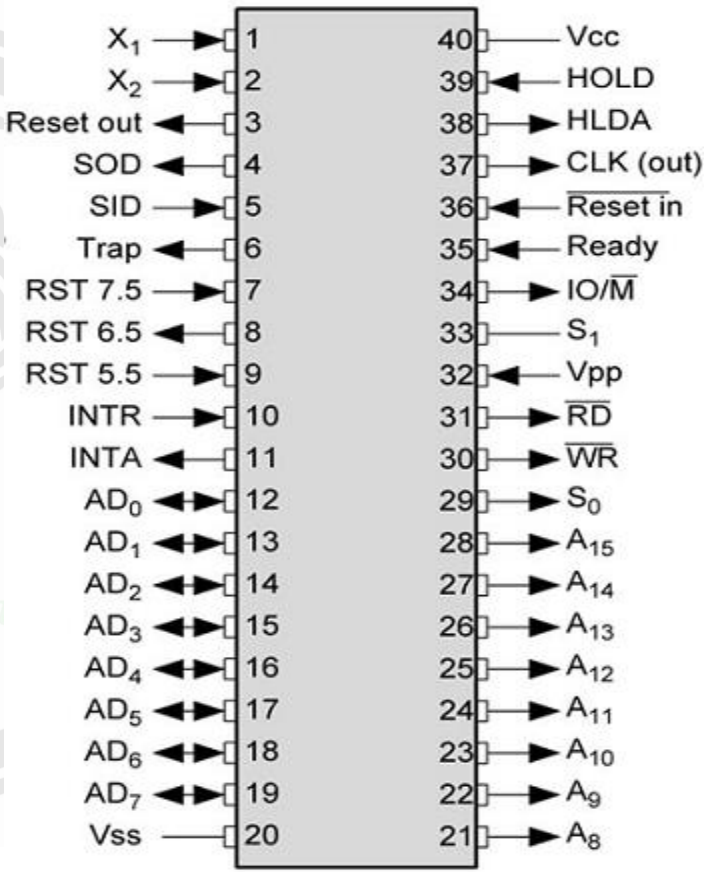
Lecture 04

By Vimal Sir

For PDF join our Telegram Channel from description

Pin Diagram Of 8085 Microprocessor





1. Address Bus and Data Bus:

The address bus is a group of sixteen lines i.e A0-A15. The address bus is unidirectional, i.e., bits flow in one direction from the microprocessor unit to the peripheral devices and uses the high order address bus.

2. Control and Status Signals:

ALE – It is an Address Latch Enable signal. It goes high during first T state of a machine cycle and enables the lower 8-bits of the address, if its value is 1 otherwise data bus is activated.

IO/M' – It is a status signal which determines whether the address is for input-output or memory. When it is high(1) the address on the address bus is for input-output devices. When it is low(0) the address on the address bus is for the memory.

SO, S1 – These are status signals. They distinguish the various types of operations such as halt, reading, instruction fetching or writing.

IO/M'	S1	S0	Data Bus Status
0	1	1	Opcode fetch
0	1	0	Memory read
0	0	1	Memory write
1	1	0	I/O read
1	0	1	I/O write
1	1	1	Interrupt acknowledge
0	0	0	Halt

RD' – It is a signal to control READ operation. When it is low the selected memory or input-output device is read.

WR' – It is a signal to control WRITE operation. When it goes low the data on the data bus is written into the selected memory or I/O location.

READY – It senses whether a peripheral is ready to transfer data or not. If READY is high(1) the peripheral is ready. If it is low(0) the microprocessor waits till it goes high. It is useful for interfacing low speed devices.

3. Power Supply and Clock Frequency:

Vcc – +5v power supply

Vss – Ground Reference

XI, X2 – A crystal is connected at these two pins. The frequency is internally divided by two, therefore, to operate a system at 3MHZ the crystal should have frequency of 6MHZ.

CLK (OUT) – This signal can be used as the system clock for other devices.

4. Interrupts and Peripheral Initiated Signals:

The 8085 has five interrupt signals that can be used to interrupt a program execution.

(i) INTR

(ii) RST 7.5

(iii) RST 6.5

(iv) RST 5.5

(v) TRAP

The microprocessor acknowledges Interrupt Request by INTA' signal. In addition to Interrupts, there are three externally initiated signals namely RESET, HOLD and READY. To respond to HOLD request, it has one signal called HLDA.

INTR – It is an interrupt request signal.

INTA' – It is an interrupt acknowledgement sent by the microprocessor after INTR is received.

5. Reset Signals:

RESET IN' – When the signal on this pin is low(0), the program-counter is set to zero, the buses are tristated and the microprocessor unit is reset.

RESET OUT – This signal indicates that the MPU is being reset. The signal can be used to reset other devices.

6. DMA Signals:

HOLD – It indicates that another device is requesting the use of the address and data bus. Having received HOLD request the microprocessor relinquishes the use of the buses as soon as the current machine cycle is completed. Internal processing may continue. After the removal of the HOLD signal the processor regains the bus.

HLDA – It is a signal which indicates that the hold request has been received after the removal of a HOLD request, the HLDA goes low.

7. Serial I/O Ports:

Serial transmission in 8085 is implemented by the two signals,

SID and SOD – SID is a data line for serial input where as SOD is a data line for serial output.

THANK YOU For Watching

SUBSCRIBE
LIKE
SHARE
COMMENT

SUBSCRIBE
LIKE
SHARE
COMMENT

Microprocessor

SUCCESS CLASSES BIJNOR
Join us
For Smart Study

Bus Structure of 8085

Microprocessor

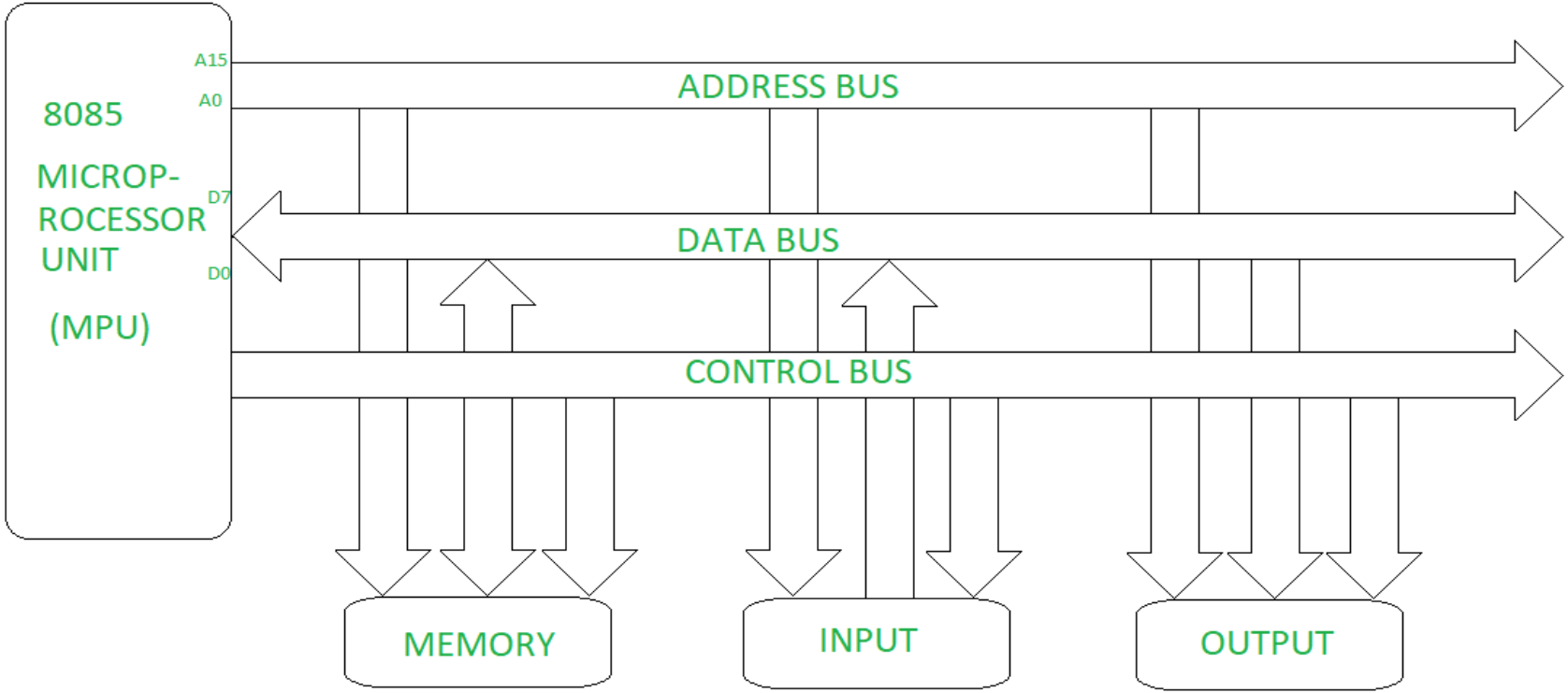
Lecture 05

By Vimal Sir

Subscribe Now

For PDF join our Telegram Channel from description

- **Bus Structure of 8085 Microprocessor**



Address bus:

- An address bus is a group of 16 lines generally called A0 – A15 to carry a 16-bit address of memory location.
- In a computer system, each peripheral or memory location is identified by a Hexadecimal number called an address. It represents 0000 H to FFFF H . H means hexadecimal no. This is similar to the postal address of a house.
- The address bus is unidirectional, which means bit flow in only one direction from MPU to the peripheral.
- MPU carries 16-bit address i.e. 216 = 65,536 or 64K memory locations.

Data Bus:

A data bus is a group of eight bidirectional lines used for data flow in both the directions between MPH and peripheral devices. The 8 data lines are manipulating 8-bit data ranging from 00 to FF i.e. (2^8 = 256) numbers from 0000 0000 -1111 1111. This 8-bit data is called word length and the register size of a microprocessor and MPH is called an 8-bit microprocessor.

Control bus:

The control bus is having various single lines used for sending control signals in the form of the pulse to the memory and I/O devices. The MPU generates specific control signals to perform a particular operation. Some of these control signals are memory read, memory write, I/O read and I/O write.

THANK YOU For Watching

SUBSCRIBE
LIKE
SHARE
COMMENT

SUBSCRIBE
LIKE
SHARE
COMMENT

Microprocessor

8085 Microprocessor Instruction Set

Data Transfer Group

Lecture 06

Subscribe Now

For PDF join our Telegram Channel from description

8085 Microprocessor Instructions:

An **instruction** of a computer is a command given to the computer to perform a specified operation on given data. In microprocessor, the **instruction** set is the collection of the instructions that the microprocessor is designed to execute.

These instructions have been classified into the following groups:

- Data Transfer Group
- Arithmetic Group
- Logical Group
- Branch Control Group

Data Transfer Instructions

- These instructions move data between registers, or between memory and registers.
- These instructions copy data from source to destination.
- While copying, the contents of source are not modified.

Data Transfer Instructions

Opcode	Operand	Description
MOV	Rd, Rs Rd, M M, Rs	Copy from source to destination.

- This instruction copies the contents of the source register into the destination register.
- The contents of the source register are not altered.
- If one of the operands is a memory location, its location is specified by the contents of the HL registers.
- **Example:** MOV B, C
- MOV B, M
- MOV M, C

Data Transfer Instructions

Opcode	Operand	Description
MVI	Rd, Data M, Data	Move immediate 8-bit

- The 8-bit data is stored in the destination register or memory.
- If the operand is a memory location, its location is specified by the contents of the H-L registers.
- **Example:** MVI A, 57H
- MVI M, 57H

Data Transfer Instructions

Opcode	Operand	Description
LXI	Reg. pair, 16-bit data	Load register pair immediate

- This instruction loads 16-bit data in the register pair.
- **Example:** LXI H, 2034 H

Data Transfer Instructions

Opcode	Operand	Description
LDA	16-bit address	Load Accumulator

- The contents of a memory location, specified by a 16-bit address in the operand, are copied to the accumulator.
- The contents of the source are not altered.
- **Example:** LDA 2034H

Data Transfer Instructions

Opcode	Operand	Description
STA	16-bit address	Store accumulator direct

- The contents of accumulator are copied into the memory location specified by the operand.
- **Example:** STA 2500 H

THANK YOU For Watching

SUBSCRIBE
LIKE
SHARE
COMMENT

SUBSCRIBE
LIKE
SHARE
COMMENT

Microprocessor

SUCCESS CLASSES BIJNOR
Join us
For Smart Study

8085 Microprocessor Instruction Set

Arithmetic Instruction Group

Lecture 07

Subscribe Now

For PDF join our Telegram Channel from description

Arithmetic Instructions

- These instructions perform the operations like:
  - Addition
  - Subtract
  - Increment
  - Decrement

Addition

- Any 8-bit number, or the contents of register, or the contents of memory location can be added to the contents of accumulator.
- The result (sum) is stored in the accumulator.
- No two other 8-bit registers can be added directly.
- **Example:** The contents of register B cannot be added directly to the contents of register C.

Subtraction

- Any 8-bit number, or the contents of register, or the contents of memory location can be subtracted from the contents of accumulator.
- The result is stored in the accumulator.
- Subtraction is performed in 2's complement form.
- If the result is negative, it is stored in 2's complement form.
- No two other 8-bit registers can be subtracted directly.

Increment / Decrement

- The 8-bit contents of a register or a memory location can be incremented or decremented by 1.
- The 16-bit contents of a register pair can be incremented or decremented by 1.
- Increment or decrement can be performed on any register or a memory location.

Arithmetic Instructions

Opcode	Operand	Description
ADD	R M	Add register or memory to accumulator

- The contents of register or memory are added to the contents of accumulator.
- The result is stored in accumulator.
- If the operand is memory location, its address is specified by H-L pair.
- All flags are modified to reflect the result of the addition.
- **Example:** ADD B or ADD M

Arithmetic Instructions

Opcode	Operand	Description
ADC	R M	Add register or memory to accumulator with carry

- The contents of register or memory and Carry Flag (CY) are added to the contents of accumulator.
- The result is stored in accumulator.
- If the operand is memory location, its address is specified by H-L pair.
- All flags are modified to reflect the result of the addition.
- **Example:** ADC B or ADC M

Arithmetic Instructions

Opcode	Operand	Description
ADI	8-bit data	Add immediate to accumulator

- The 8-bit data is added to the contents of accumulator.
- The result is stored in accumulator.
- All flags are modified to reflect the result of the addition.
- **Example:** ADI 45 H

Arithmetic Instructions

Opcode	Operand	Description
ACI	8-bit data	Add immediate to accumulator with carry

- The 8-bit data and the Carry Flag (CY) are added to the contents of accumulator.
- The result is stored in accumulator.
- All flags are modified to reflect the result of the addition.
- **Example:** ACI 45 H

Arithmetic Instructions

Opcode	Operand	Description
SUB	R M	Subtract register or memory from accumulator

- The contents of the register or memory location are subtracted from the contents of the accumulator.
- The result is stored in accumulator.
- If the operand is memory location, its address is specified by H-L pair.
- All flags are modified to reflect the result of subtraction.
- **Example:** SUB B or SUB M

Arithmetic Instructions

Opcode	Operand	Description
SBB	R M	Subtract register or memory from accumulator with borrow

- The contents of the register or memory location and Borrow Flag (i.e. CY) are subtracted from the contents of the accumulator.
- The result is stored in accumulator.
- If the operand is memory location, its address is specified by H-L pair.
- All flags are modified to reflect the result of subtraction.
- **Example:** SBB B or SBB M

Arithmetic Instructions

Opcode	Operand	Description
SUI	8-bit data	Subtract immediate from accumulator

- The 8-bit data is subtracted from the contents of the accumulator.
- The result is stored in accumulator.
- All flags are modified to reflect the result of subtraction.
- **Example:** SUI 45 H

Arithmetic Instructions

Opcode	Operand	Description
SBI	8-bit data	Subtract immediate from accumulator with borrow

- The 8-bit data and the Borrow Flag (i.e. CY) is subtracted from the contents of the accumulator.
- The result is stored in accumulator.
- All flags are modified to reflect the result of subtraction.
- **Example:** SBI 45 H

Arithmetic Instructions

Opcode	Operand	Description
INR	R M	Increment register or memory by 1

- The contents of register or memory location are incremented by 1.
- The result is stored in the same place.
- If the operand is a memory location, its address is specified by the contents of H-L pair.
- **Example:** INR B or INR M

Arithmetic Instructions

Opcode	Operand	Description
DCR	R M	Decrement register or memory by 1

- The contents of register or memory location are decremented by 1.
- The result is stored in the same place.
- If the operand is a memory location, its address is specified by the contents of H-L pair.
- **Example:** DCR B or DCR M

THANK YOU For Watching

SUBSCRIBE
LIKE
SHARE
COMMENT

SUBSCRIBE
LIKE
SHARE
COMMENT

Microprocessor

8085 Microprocessor Instruction Set

Logical Instruction Group

Lecture 08

Subscribe Now

For PDF join our Telegram Channel from description

Logical Instructions

- These instructions perform logical operations on data stored in registers, memory and status flags.

- The logical operations are:
  - AND
  - OR
  - XOR

AND, OR, XOR

- Any 8-bit data, or the contents of register, or memory location can logically have
  - AND operation
  - OR operation
  - XOR operation

  with the contents of accumulator.
- The result is stored in accumulator.

Compare

- Any 8-bit data, or the contents of register, or memory location can be compares for:
  - Equality
  - Greater Than
  - Less Than

  with the contents of accumulator.
- The result is reflected in status flags.

Logical Instructions

Opcode	Operand	Description
CMP	R M	Compare register or memory with accumulator

- The contents of the operand (register or memory) are compared with the contents of the accumulator.
- Both contents are preserved .
- The result of the comparison is shown by setting the flags of the PSW as follows:

Logical Instructions

Opcode	Operand	Description
CMP	R M	Compare register or memory with accumulator

- if (A) < (reg/mem): carry flag is set
- if (A) = (reg/mem): zero flag is set
- if (A) > (reg/mem): carry and zero flags are reset.
- **Example:** CMP B or CMP M

Logical Instructions

Opcode	Operand	Description
CPI	8-bit data	Compare immediate with accumulator

- The 8-bit data is compared with the contents of accumulator.
- The values being compared remain unchanged.
- The result of the comparison is shown by setting the flags of the PSW as follows:

Logical Instructions

Opcode	Operand	Description
CPI	8-bit data	Compare immediate with accumulator

- if (A) < data: carry flag is set
- if (A) = data: zero flag is set
- if (A) > data: carry and zero flags are reset
- **Example:** CPI 89H

Logical Instructions

Opcode	Operand	Description
ANA	R M	Logical AND register or memory with accumulator

- The contents of the accumulator are logically ANDed with the contents of register or memory.
- The result is placed in the accumulator.
- If the operand is a memory location, its address is specified by the contents of H-L pair.
- S, Z, P are modified to reflect the result of the operation.
- CY is reset and AC is set.
- **Example:** ANA B or ANA M.

Logical Instructions

Opcode	Operand	Description
ANI	8-bit data	Logical AND immediate with accumulator

- The contents of the accumulator are logically ANDed with the 8-bit data.
- The result is placed in the accumulator.
- S, Z, P are modified to reflect the result.
- CY is reset, AC is set.
- **Example:** ANI 86H.

Logical Instructions

Opcode	Operand	Description
XRA	R M	Exclusive OR register or memory with accumulator

- The contents of the accumulator are XORed with the contents of the register or memory.
- The result is placed in the accumulator.
- If the operand is a memory location, its address is specified by the contents of H-L pair.
- S, Z, P are modified to reflect the result of the operation.
- CY and AC are reset.
- **Example:** XRA B or XRA M.

Logical Instructions

Opcode	Operand	Description
ORA	R M	Logical OR register or memory with accumulator

- The contents of the accumulator are logically ORed with the contents of the register or memory.
- The result is placed in the accumulator.
- If the operand is a memory location, its address is specified by the contents of H-L pair.
- S, Z, P are modified to reflect the result.
- CY and AC are reset.
- **Example:** ORA B or ORA M.

Logical Instructions

Opcode	Operand	Description
ORI	8-bit data	Logical OR immediate with accumulator

- The contents of the accumulator are logically ORed with the 8-bit data.
- The result is placed in the accumulator.
- S, Z, P are modified to reflect the result.
- CY and AC are reset.
- **Example:** ORI 86H.

Logical Instructions

Opcode	Operand	Description
XRA	R M	Logical XOR register or memory with accumulator

- The contents of the accumulator are XORed with the contents of the register or memory.
- The result is placed in the accumulator.
- If the operand is a memory location, its address is specified by the contents of H-L pair.
- S, Z, P are modified to reflect the result of the operation.
- CY and AC are reset.
- **Example:** XRA B or XRA M.

Logical Instructions

Opcode	Operand	Description
XRI	8-bit data	XOR immediate with accumulator

- The contents of the accumulator are XORed with the 8-bit data.
- The result is placed in the accumulator.
- S, Z, P are modified to reflect the result.
- CY and AC are reset.
- **Example:** XRI 86H.

THANK YOU For Watching

SUBSCRIBE
LIKE
SHARE
COMMENT

SUBSCRIBE
LIKE
SHARE
COMMENT

Microprocessor

SUCCESS CLASSES BIJNOR
Join us
For Smart Study

8085 Microprocessor Instruction Set

Branching & Control Instruction Group

Lecture 09

Subscribe Now

For PDF join our Telegram Channel from description

Branching Instructions

- The branching instruction alter the normal sequential flow.
- These instructions alter either unconditionally or conditionally.

Branching Instructions

Opcode	Operand	Description
JMP	16-bit address	Jump unconditionally

- The program sequence is transferred to the memory location specified by the 16-bit address given in the operand.
- **Example:** JMP 2034 H.

Branching Instructions

Opcode	Operand	Description
Jx	16-bit address	Jump conditionally

- The program sequence is transferred to the memory location specified by the 16-bit address given in the operand based on the specified flag of the PSW.
- **Example:** JZ 2034 H.

Jump Conditionally

Opcode	Description	Status Flags
JC	Jump if Carry	CY = 1
JNC	Jump if No Carry	CY = 0
JP	Jump if Positive	S = 0
JM	Jump if Minus	S = 1
JZ	Jump if Zero	Z = 1
JNZ	Jump if No Zero	Z = 0
JPE	Jump if Parity Even	P = 1
JPO	Jump if Parity Odd	P = 0

Branching Instructions

Opcode	Operand	Description
CALL	16-bit address	Call unconditionally

- The program sequence is transferred to the memory location specified by the 16-bit address given in the operand.
- Before the transfer, the address of the next instruction after CALL (the contents of the program counter) is pushed onto the stack.
- **Example:** CALL 2034 H.

Branching Instructions

Opcode	Operand	Description
Cx	16-bit address	Call conditionally

- The program sequence is transferred to the memory location specified by the 16-bit address given in the operand based on the specified flag of the PSW.
- Before the transfer, the address of the next instruction after the call (the contents of the program counter) is pushed onto the stack.
- **Example:** CZ 2034 H.

Call Conditionally

Opcode	Description	Status Flags
CC	Call if Carry	CY = 1
CNC	Call if No Carry	CY = 0
CP	Call if Positive	S = 0
CM	Call if Minus	S = 1
CZ	Call if Zero	Z = 1
CNZ	Call if No Zero	Z = 0
CPE	Call if Parity Even	P = 1
CPO	Call if Parity Odd	P = 0

Branching Instructions

Opcode	Operand	Description
RET	None	Return unconditionally

- The program sequence is transferred from the subroutine to the calling program.
- The two bytes from the top of the stack are copied into the program counter, and program execution begins at the new address.
- **Example:** RET.

Branching Instructions

Opcode	Operand	Description
Rx	None	Call conditionally

- The program sequence is transferred from the subroutine to the calling program based on the specified flag of the PSW.
- The two bytes from the top of the stack are copied into the program counter, and program execution begins at the new address.
- **Example:** RZ.

Return Conditionally

Opcode	Description	Status Flags
RC	Return if Carry	CY = 1
RNC	Return if No Carry	CY = 0
RP	Return if Positive	S = 0
RM	Return if Minus	S = 1
RZ	Return if Zero	Z = 1
RNZ	Return if No Zero	Z = 0
RPE	Return if Parity Even	P = 1
RPO	Return if Parity Odd	P = 0

Control Instructions

- The control instructions control the operation of microprocessor.

Control Instructions

Opcode	Operand	Description
NOP	None	No operation

- No operation is performed.
- The instruction is fetched and decoded but no operation is executed.
- **Example:** NOP

Control Instructions

Opcode	Operand	Description
HLT	None	Halt

- The CPU finishes executing the current instruction and halts any further execution.
- An interrupt or reset is necessary to exit from the halt state.
- **Example:** HLT

Control Instructions

Opcode	Operand	Description
DI	None	Disable interrupt

- The interrupt enable flip-flop is reset and all the interrupts except the TRAP are disabled.
- No flags are affected.
- **Example:** DI

Control Instructions

Opcode	Operand	Description
EI	None	Enable interrupt

- The interrupt enable flip-flop is set and all interrupts are enabled.
- No flags are affected.
- This instruction is necessary to re-enable the interrupts (except TRAP).
- **Example:** EI

THANK YOU For Watching

SUBSCRIBE
LIKE
SHARE
COMMENT

SUBSCRIBE
LIKE
SHARE
COMMENT

Microprocessor

Addressing Modes of 8085 Microprocessor

Lecture 10

By Vimal Sir

Subscribe Now

For PDF join our Telegram Channel from description

Addressing Modes of 8085 Microprocessor:-

- The different methods to select (address) the operands are called the Addressing Modes.
- Types of addressing modes of 8085 microprocessor :
  - Immediate Addressing
  - Register Addressing
  - Direct Addressing
  - Indirect Addressing
  - Implied Addressing

- **Immediate Addressing Mode:**
- In immediate addressing mode, the data (8 / 16 bit) is specified in the instruction itself.
- The immediate addressing instructions are either 2 bytes or 3 bytes long.
- In 2 byte instruction, the first byte is OPCODE, and the second byte is the 8-bit data.
- In 3 byte instruction, the first byte is OPCODE, second and third bytes are 16-bit data.
- **The instruction containing the letter "I" indicate immediate addressing mode.**
- **Examples:**
- **MVI A, A0 H** : This instruction transfers immediate data (A0 H) to A register.
- **LXI H, C200 H** : This instruction transfer 16-bit immediate data C200 to HL register pair. Lower order data(00H) to L register and high order data (C2 H) to H register.

Register Addressing Mode:

- In register addressing mode the source and destination operands are general-purpose registers.
- The register addressing instructions are generally of 1 byte i.e OPCODE only.
- The OPCODE specifies the operation and registers to be used to perform the operation.

Examples:

MOV D, B : This instruction copies the contents of register B to the D register. The source and destination operands are both registers.

ADD B : This instruction adds the content of B register and A register, The data is present in both B and A registers. The result is stored in the accumulator.

Direct Addressing Mode:

- In direct addressing mode, the 16-bit address of the operand is given within the instruction itself.
- The instruction in the direct addressing mode is 3-byte instructions. The first byte is OPCODE, the second slower order address mode and the third is the higher-order address mode.
- For I/O instruction that uses direct addressing mode are 2-byte as the address if I/O is one byte.

Examples: LDA C200 H : Load accumulator directly from the memory location. In this instruction, the contents of C200 memory location are transferred to the accumulator.

STA C200 H : Store accumulator directly to memory location. In this instruction, the content of the accumulator are store at memory location C200 H.

Indirect Addressing Mode:

In indirect addressing mode the instruction reference the memory through a register pair.

i.e. the memory address where the operand is located is specified by the content of a register pair.

Examples:

MOV A, M : In this case, M is a memory pointer specifying the HL register pair where the address is stored. The contents of the HL pair are used as addresses and the content of that memory location is transferred to the accumulator.

LDAX B : In this case, the BC register pair is used as an address and the content of the memory location specified by the BC register pair is copied to the accumulator.

Implied Addressing or Implicit Addressing Mode:

The implied mode of addressing does not require any operand.

The data is specified within OPCODE itself.

Generally, the implied addressing mode instruction is a 1-byte instruction.

The data is supposed to be present generally in the accumulator.

Examples:

RAL : Rotate accumulator left, it operates on the data in accumulator only. So whenever RAL is used it is implied that the data to be operated on is available in the accumulator only.

CMC : Complement carry flag.

THANK YOU For Watching

SUBSCRIBE
LIKE
SHARE
COMMENT

SUBSCRIBE
LIKE
SHARE
COMMENT

Microprocessor

SUCCESS CLASSES BIJNOR
Join us
For Smart Study

Assembler Directives of 8085 Microprocessor

Subscribe Now

Lecture 11

By Vimal Sir

For PDF join our Telegram Channel from description

- **Assembler Directives**

- Assembler directives are the **instructions** used by the assembler at the time of assembling a source program. More specifically, we can say, assembler directives are the commands or instructions that control the operation of the assembler.
- Assembler directives are the instructions provided to the assembler, not the processor as the processor has nothing to do with these instructions. These instructions are also known as **pseudo-instructions** or **pseudo-opcode**.

- **Assembler Directives of 8085**

The assembler directives given below are used by 8085 assemblers:

DB: *Define Byte*

This directive is used for the purpose of allocating and initializing single or multiple data bytes.

```
AREA DB 30H, 52H, 35H
```

Memory name AREA has three consecutive locations where 30H, 52H and 35H are to be stored.

DW: *Define Word*

It is used for initialising single or multiple data words (16-bit).

```
MARK DW 1020H, 4216H
```

These two 16-bit data 1020H and 4216H are stored at 4 consecutive locations in the memory MARK.

END: *End of program*

This directive is used at the time of program termination.

EQU: *Equate*

It is used to assign any numerical value or constant to the variable.

```
DONE EQU 10H
```

Variable name ‘DONE’ has value 10H

ORG: *Origin*

This directive is used at the time of assigning starting address for a module or segment.

```
ORG 1050H
```

By this instruction, the assembler gets to know that the statements following this instruction, must be stored in the memory location beginning with address 1050H.

THANK YOU For Watching

SUBSCRIBE
LIKE
SHARE
COMMENT

SUBSCRIBE
LIKE
SHARE
COMMENT

Microprocessor

Machine Cycles of 8085 Microprocessor

Lecture 12

Subscribe Now

By Vimal Sir

For PDF join our Telegram Channel from description

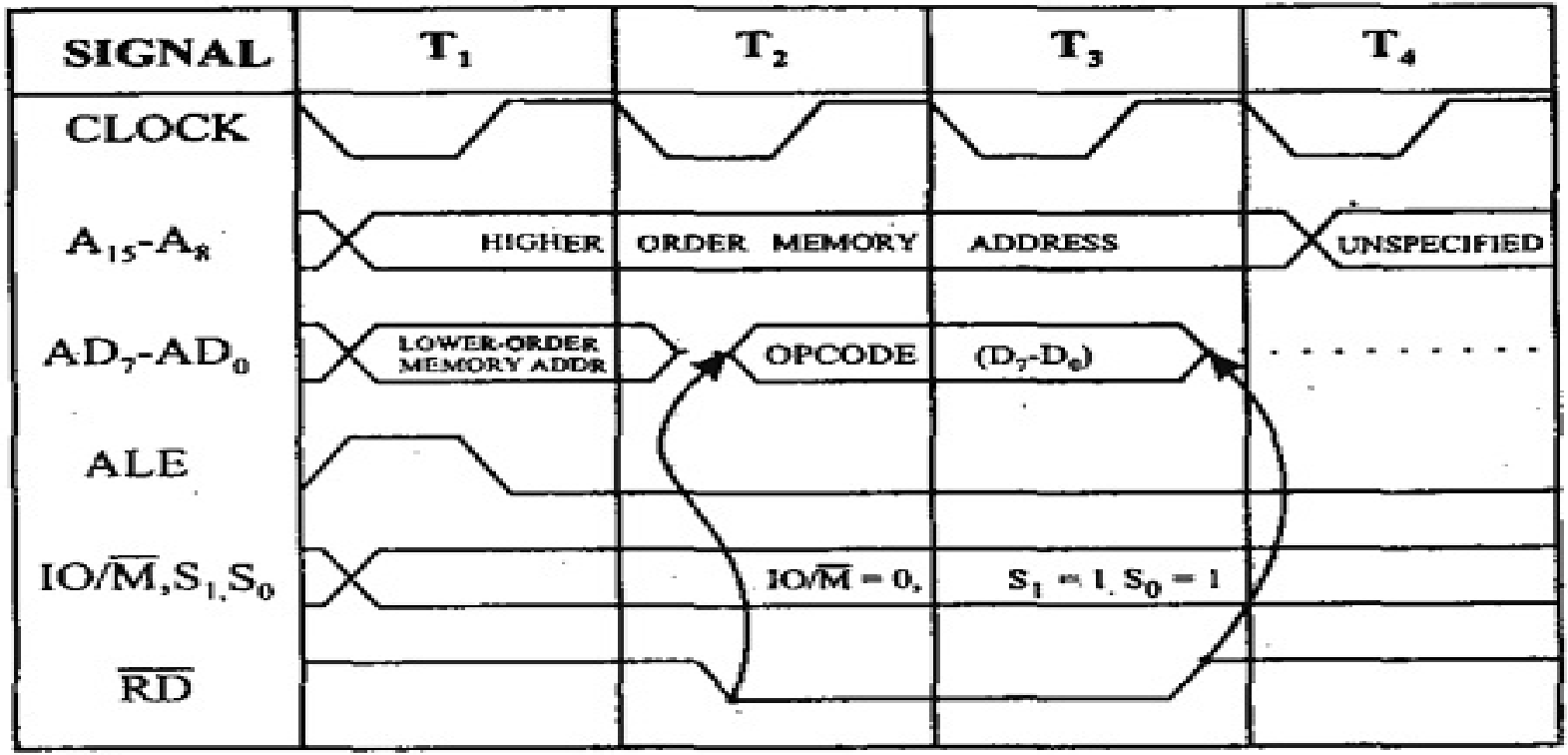
- **Machine Cycle in 8085 Microprocessor**

The time needed for completing one operation of accessing memory, I/O or acknowledging an external request is termed as Machine cycle. It is comprised of T-states. One subdivision of the operation completed in one clock period is termed as T-state. The following are the various machine cycles of 8085 microprocessor.

1. Opcode Fetch (OF)
2. Memory Read (MR)
3. Memory Write (MW)
4. I/O Read (IOR)
5. I/O Write (IOW)

Signal Representation

- **Opcode Fetch (OF) machine cycle of 8085:**



S. No	T state	Operation
1	T_1	The microprocessor places the higher order 8-bits of the memory address on A15 – A8 address bus and the lower order 8-bits of the memory address on AD7 – AD0 address / data bus.
2		The microprocessor makes the ALE signal HIGH and at the middle of T1 state, ALE signal goes LOW.
3		The status signals are changed as IO/M' = 0, S1 =1 and S0 = 1. These status signals do not change throughout the OF machine cycle.
4	T_2	The microprocessor makes the RD' line LOW to enable memory read and increments the Program Counter.
5		The contents on D7 – D0 (i.e. the Opcode) are placed on the address / data bus.
6	T_3	The microprocessor transfers the Opcode on the address / data bus to Instruction Register (IR).
7		The microprocessor makes the RD' line HIGH to disable memory read.
8	T_4	The microprocessor decodes the instruction.

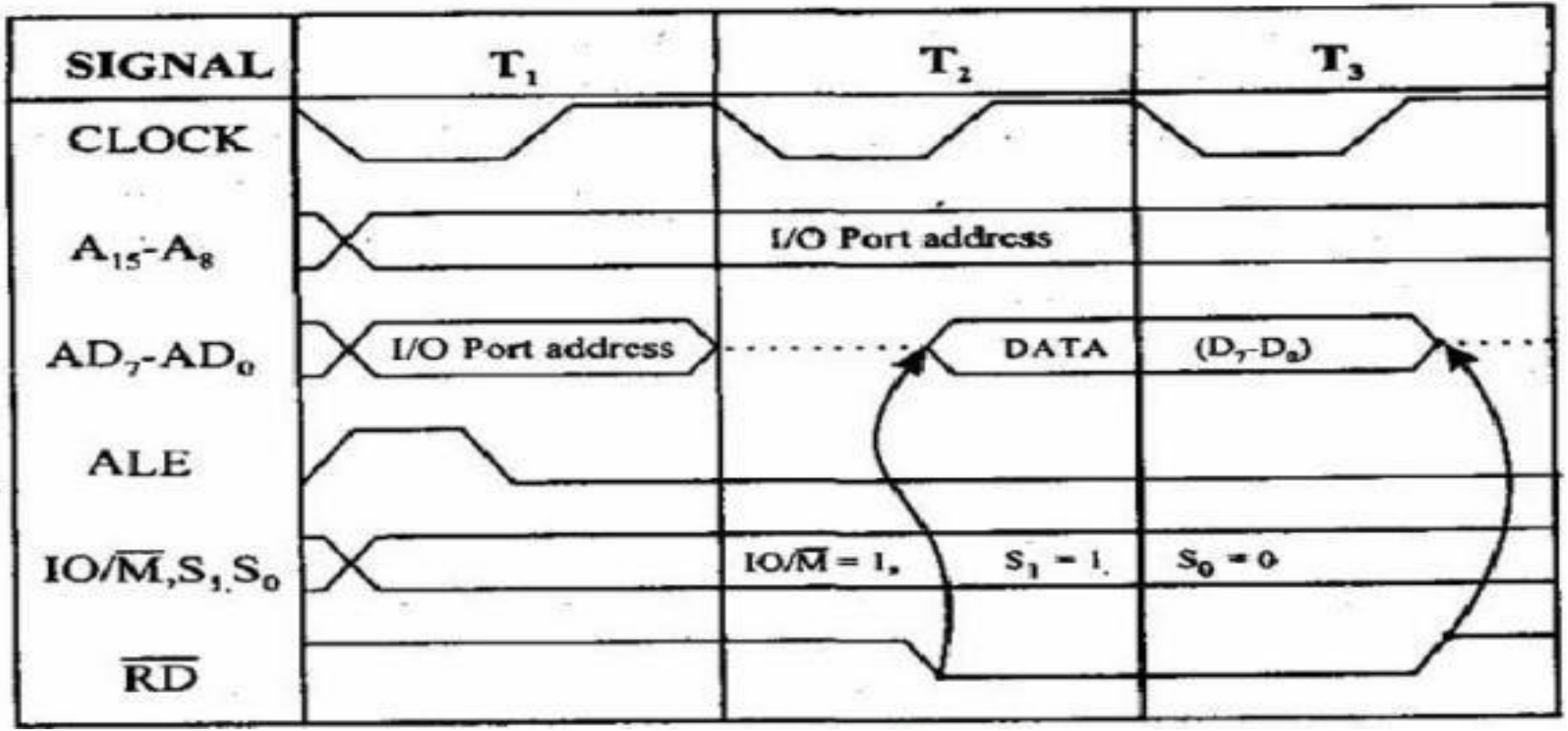
- **Memory Read Machine Cycle of 8085:**

S. No	T state	Operation
1	T_1	The microprocessor places the higher order 8-bits of the memory address on A15 – A8 address bus and the lower order 8-bits of the memory address on AD7 – AD0 address / data bus.
2		The microprocessor makes the ALE signal HIGH and at the middle of T1 state, ALE signal goes LOW.
3		The status signals are changed as IO/$\overline{M}$ = 0, S1 =1 and S0 = 0. These status signals do not change throughout the memory read machine cycle.
4	T_2	The microprocessor makes the $\overline{RD}$ line LOW to enable memory read and increments the Program Counter.
5		The contents on D7 – D0 (i.e. the data) are placed on the address / data bus.
6	T_3	The data loaded on the address / data bus is moved to the microprocessor.
7		The microprocessor makes the $\overline{RD}$ line HIGH to disable the memory read operation.

- **Memory Write Machine Cycle of 8085:**

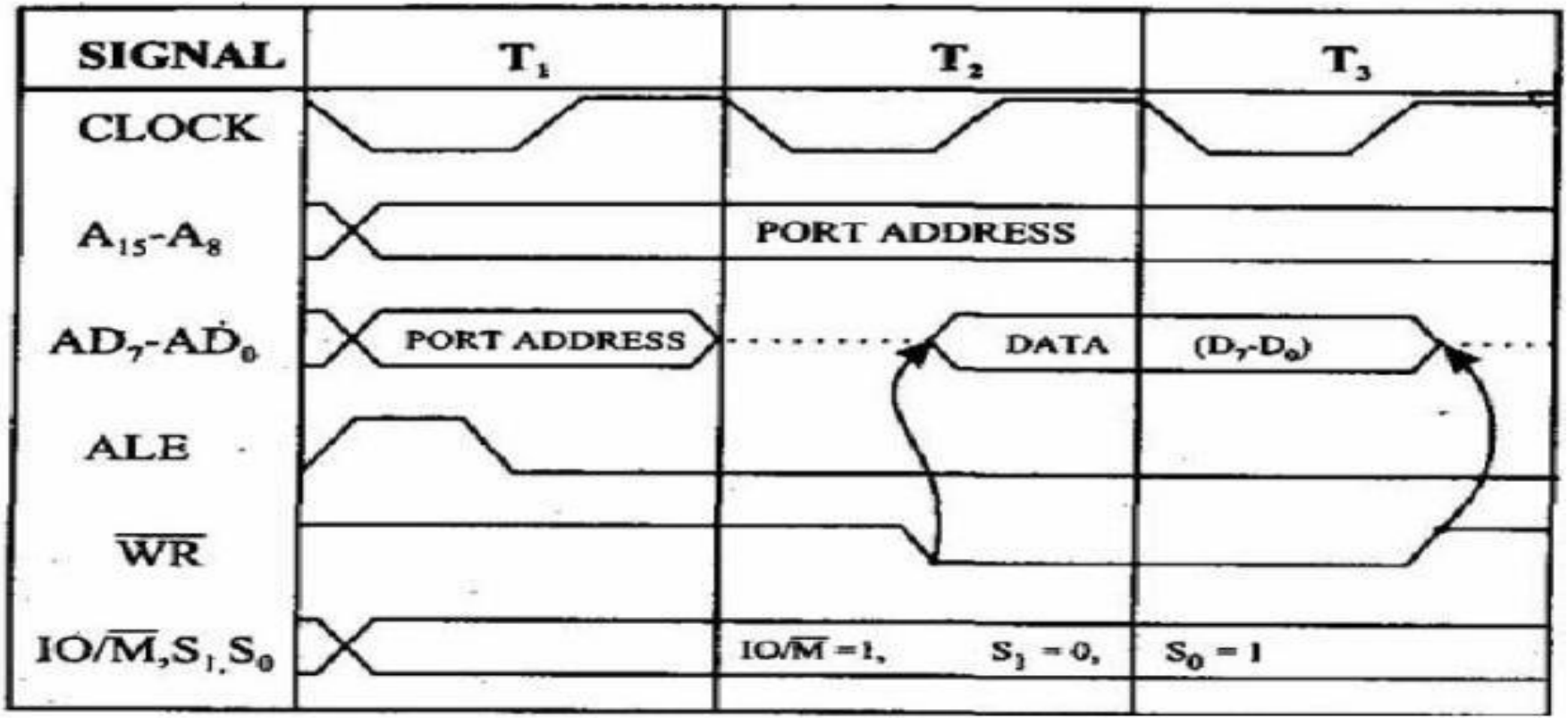
S. No	T state	Operation
1	T_1	The microprocessor places the higher order 8-bits of the memory address on A15 – A8 address bus and the lower order 8-bits of the memory address on AD7 – AD0 address / data bus.
2		The microprocessor makes the ALE signal HIGH and at the middle of T1 state, ALE signal goes LOW.
3		The status signals are changed as IO/M' = 0, S1 =0 and S0 = 1. These status signals do not change throughout the memory write machine cycle.
4	T_2	The microprocessor makes the WR' line LOW to enable memory write.
5		The contents of the specified register are placed on the address / data bus.
6	T_3	The data placed on the address / data bus is transferred to the specified memory location.
7		The microprocessor makes the WR' line HIGH to disable the memory write operation.

- **I/O Read Machine Cycle of 8085**



S. No	T state	Operation
1	T_1	The microprocessor places the address of the I/O port specified in the instruction on A15 – A8 address bus and also on AD7 – AD0 address / data bus.
2		The microprocessor makes the ALE signal HIGH and at the middle of T1 state, ALE signal goes LOW.
3		The status signals are changed as IO/M' = 0, S1 =1 and S0 = 0. These status signals do not change throughout the I/O read machine cycle.
4	T_2	The microprocessor makes the RD' line LOW to enable I/O read.
5		The contents on D7 – D0 (i.e. the data) are placed on the address / data bus.
6	T_3	The data loaded on the address / data bus is moved to the microprocessor ie., to the accumulator.
7		The microprocessor makes the RD' line HIGH to disable the I/O read operation.

- **I/O Write Machine Cycle of 8085**



S. No	T state	Operation
1	T_1	The microprocessor places the address of the I/O port specified in the instruction on A15 – A8 address bus and also on AD7 – AD0 address / data bus.
2		The microprocessor makes the ALE signal HIGH and at the middle of T1 state, ALE signal goes LOW.
3		The status signals are changed as IO/M' = 0, S1 =0 and S0 = 1. These status signals do not change throughout the I/O write machine cycle.
4	T_2	The microprocessor makes the WR' line LOW to enable I/O write.
5		The contents of the Accumulator are placed on the address / data bus.
6	T_3	The data placed on the address / data bus is transferred to the specified I/O port.
7		The microprocessor makes the WR' line HIGH to disable the I/O write operation

THANK YOU For Watching

SUBSCRIBE
LIKE
SHARE
COMMENT

SUBSCRIBE
LIKE
SHARE
COMMENT