Microprocessor

Chapter 2: Basic Computer Architecture

Er. Sachita Nand Mishra
M.E in Computer and Electronics Engineering

Contents

- 8085 Microprocessor Architecture and Operations
  - ✓ Address, Data And Control Buses
  - ✓ Internal Data Operation and Registers
  - ✓ Externally Initiated Operations
  - ✓ Addressing Modes
  - ✓ Memory and Memory Operations
  - ✓ Flag and Flag Register
  - ✓ 8085 Pin Diagram and Functions
  - ✓ Multiplexing and De-multiplexing of address/data bus
  - ✓ Generation Of Control Signals
- 8086 Microprocessor
  - ✓ Logical Block Diagram
  - ✓ Segment Registers,
  - ✓ Memory Segmentation
  - ✓ Bus Interface Unit and Execution Unit
  - ✓ Pipelining

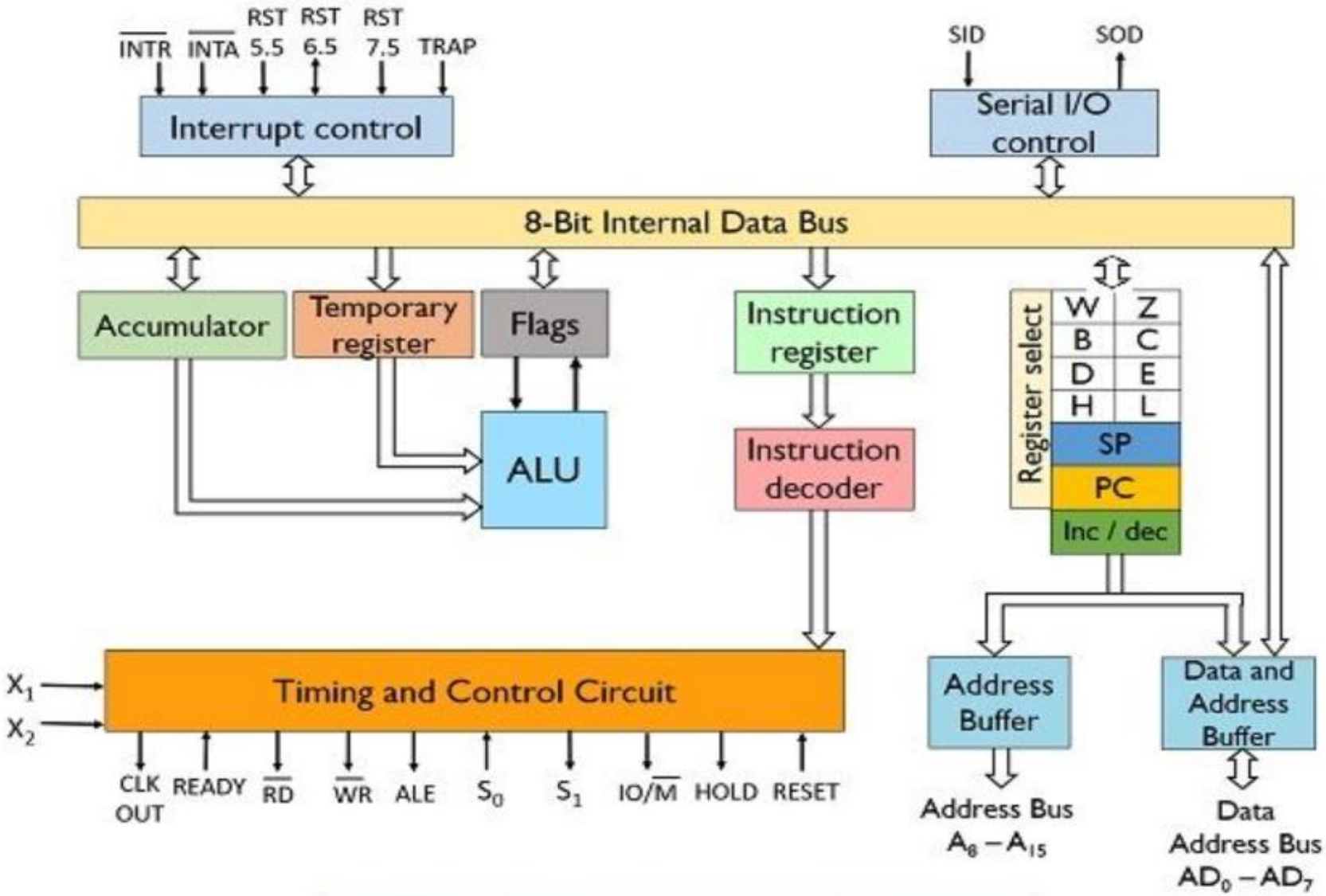
Introduction to 8085 Microprocessor

- 8085 is pronounced as "eighty-eighty-five" microprocessor.
- It is an 8-bit microprocessor designed by Intel in 1977.
- It has the following configuration:
  - ✓ 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 upward compatible.
- It is used in washing machines, microwave ovens, mobile phones, etc.

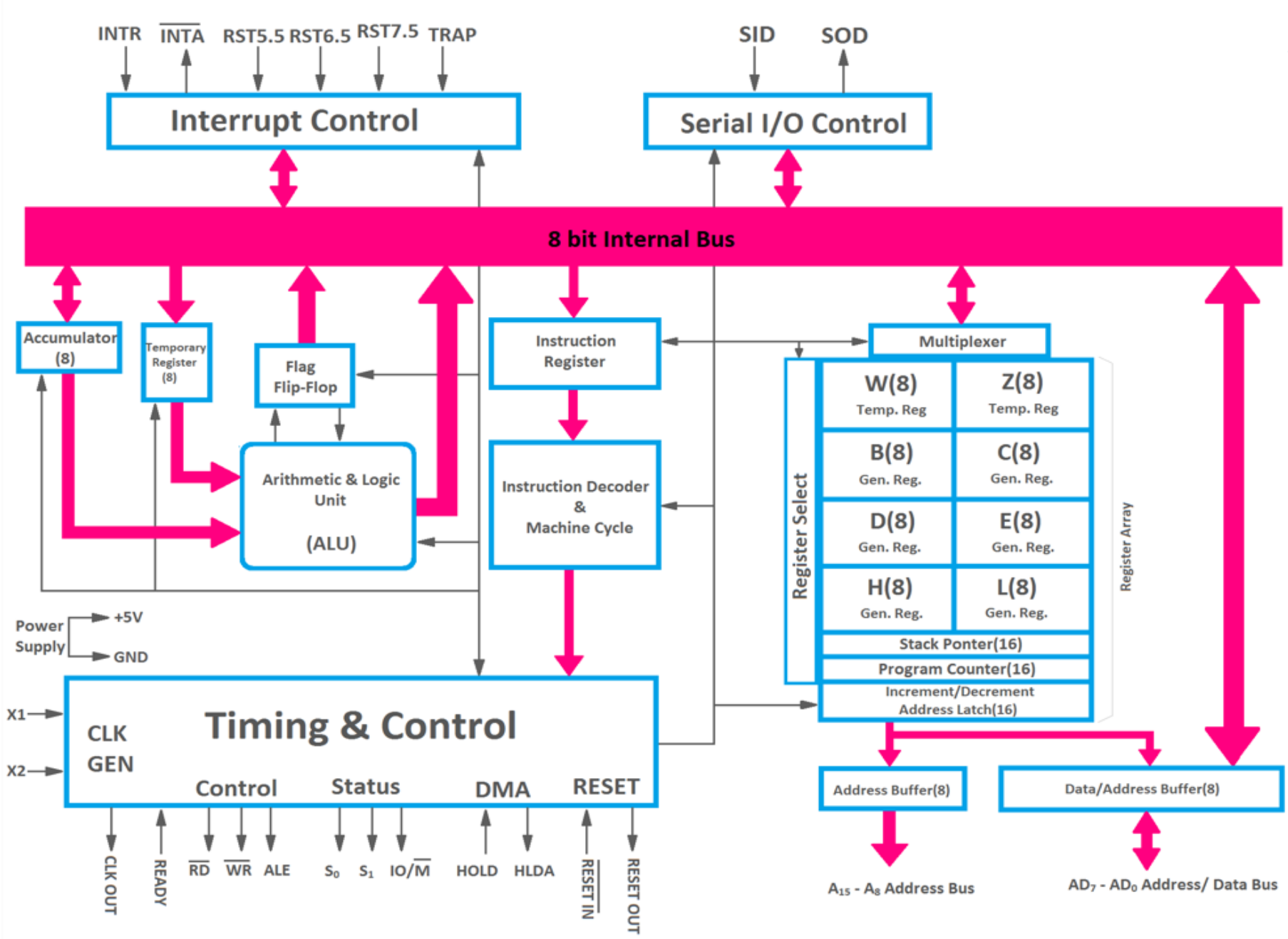
Architecture of 8085 Microprocessor

- The architecture of 8085 microprocessor provides the idea about what are the operations to be executed and how these are performed.
- **It can perform operations that are given below**:
  - ✓ Operates on and stores 8-bit data.
  - ✓ It executes arithmetic and logic operations.
  - ✓ 8085 also sequences the instructions to be executed.
  - ✓ Stores data temporarily.
- The Intel 8085 A is a complete 8 bit parallel central processing unit.
- The main components of 8085A are array of registers, the arithmetic logic unit, the encoder/decoder, interrupt and timing and control circuits linked by an internal data bus.
- The block diagram is shown below:

Architecture of 8085 Microprocessor



Architecture of 8085 Microprocessor


INTR
INTA
RST5.5
RST6.5
RST7.5
TRAP
Interrupt Control
SID
SOD
Serial I/O Control
8 bit Internal Bus
Accumulator (8)
Temporary Register (8)
Flag Flip-Flop
Arithmetic & Logic Unit (ALU)
Instruction Register
Instruction Decoder & Machine Cycle
Multiplexer
Register Select
W(8) Temp. Reg
Z(8) Temp. Reg
B(8) Gen. Reg.
C(8) Gen. Reg.
D(8) Gen. Reg.
E(8) Gen. Reg.
H(8) Gen. Reg.
L(8) Gen. Reg.
Register Array
Stack Ponter(16)
Program Counter(16)
Increment/Decrement Address Latch(16)
Power Supply
+5V
GND
X1
X2
CLK GEN
Timing & Control
Control
Status
DMA
RESET
CLK OUT
READY
RD
WR
ALE
S0
S1
IO/M
HOLD
HLDA
RESET IN
RESET OUT
Address Buffer(8)
Data/Address Buffer(8)
A15 - A8 Address Bus
AD7 - AD0 Address/ Data Bus

Architecture of 8085 Microprocessor

- 8085 consists of the following functional units :

❑ Accumulator:

- ✓ The accumulator is an 8-bit register with supplies the data directly to the ALU during program executions to perform the arithmetic and logical operations.
- ✓ In fact, it helps to store load and store new data
- ✓ It is connected between internal data bus & ALU.

❑ Arithmetic and logic unit:

- ✓ It is the main part of the microprocessor.
- ✓ As its name suggests, it is a digital circuit that performs arithmetic or mathematic operations such as addition, subtraction, multiplication, division, and logical operations such as OR, AND, NOT, increment, decrement.
- ✓ It works with 8-bit data only.
- ✓ During the program executions, it takes data from the accumulator and temporary register.

❑ General purpose register:

- ✓ There are 6 general purpose registers in 8085 processors, 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.
- ✓ H & L can be used as a data pointer (holds memory address)

Architecture of 8085 Microprocessor

- Stack pointer:
  - ✓ It is also a 16-bit register works like stack, which is always incremented/decremented push & pop operations
  - ✓ The stack pointer is also a 16-bit register that is used to point into memory.
  - ✓ The memory this register points to is a special area called the stack.
  - ✓ The stack is an area of memory used to hold data that will be retrieved soon.
  - ✓ The stack is usually accessed in a Last In First Out (LIFO) fashion.

- 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.

Architecture of 8085 Microprocessor

- Temporary register:
  - ✓ The temporary register is an 8 bit register that is used to store data temporarily during the program executions.
  - ✓ When the ALU starts executing a program the data comes from the general-purpose register and is stored in the temporary register.
  - ✓ It helps to provide the supporting data to the ALU.
- 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)
  - ❖ Sign Flag (S) – After any operation if result is negative sign flag becomes set, i.e. If result is positive sign flag becomes reset i.e. 0.
    - Example:
      - MVI A 30 (load 30H in register A) ,MVI B 40 (load 40H in register B) ; SUB B (A = A – B) These set of instructions will set the sign flag to 1 as 30 – 40 is a negative number.
      - MVI A 40 (load 40H in register A), MVI B 30 (load 30H in register B) ;SUB B (A = A – B) These set of instructions will reset the sign flag to 0 as 40 – 30 is a positive number.

Architecture of 8085 Microprocessor

- **Zero Flag (Z):** After any arithmetical or logical operation if the result is 0 (00)H, the zero flag becomes set i.e. 1, otherwise it becomes reset i.e. 0.
  - Example:
    - MVI A 10 (load 10H in register A) ; SUB A (A = A – A)
    - These set of instructions will set the zero flag to 1 as 10H – 10H is 00H
- **Auxiliary Carry Flag (AC):** If intermediate carry is generated this flag is set to 1, otherwise it is reset to 0.
  - Example:
  - MOV A 2B (load 2BH in register A)
  - MOV B 39 (load 39H in register B)
  - ADD B (A = A + B)
  - These set of instructions will set the auxiliary carry flag to 1, as on adding 2B and 39, addition of lower order nibbles B and 9 will generate a carry.
- **Parity Flag (P):** If after any arithmetic or logical operation the result has even parity, an even number of 1 bits, the parity register becomes set i.e. 1, otherwise it becomes reset.
  - 1: accumulator has even number of 1 bits
  - 0: accumulator has odd parity

Architecture of 8085 Microprocessor

- **Carry Flag (CY):** Carry is generated when performing n bit operations and the result is more than n bits, then this flag becomes set i.e. 1, otherwise it becomes reset i.e. 0.
  - During subtraction (A-B), if A>B it becomes reset and if (A<B) it becomes set.
  - Carry flag is also called borrow flag.

- **Instruction register 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.

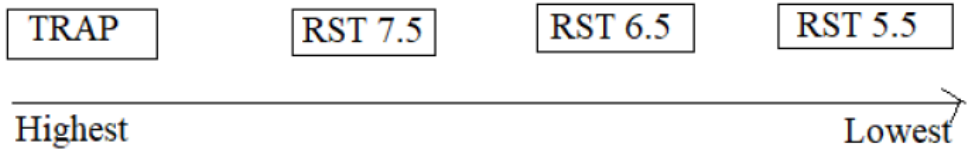
- **Timing and control unit**
  - ✓ It provides timing and control signal to all the components of the microprocessor to perform operations.
  - ✓ It not only provides signals to the internal components but also provides the timing and control signal to the external component and circuit connected to the microprocessor
  - ✓ Following are the timing and control signals-
  - Control Signals: READY, RD', WR', ALE
  - Status Signals: S0, S1, IO/M'
  - DMA Signals: HOLD, HLDA
  - RESET Signals: RESET IN, RESET OUT

Architecture of 8085 Microprocessor

❑ 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.
- ✓ When microprocessor receives interrupt signals, it sends an acknowledgement (INTA) to the peripheral which is requesting for its service.

- **Maskable and Non-Maskable Interrupts**
  - ***Maskable*** *Interrupts* are those which can be disabled or ignored by the microprocessor. *INTR, RST 7.5, RST 6.5, RST 5.5* are maskable interrupts in 8085 microprocessor.
  - **Non-Maskable** Interrupts are those which cannot be disabled or ignored by microprocessor. *TRAP* is a non-maskable interrupt.
- **Priority of Interrupts**
  - When microprocessor receives multiple interrupt requests simultaneously, it will execute the interrupt service request (ISR) according to the priority of the interrupts.



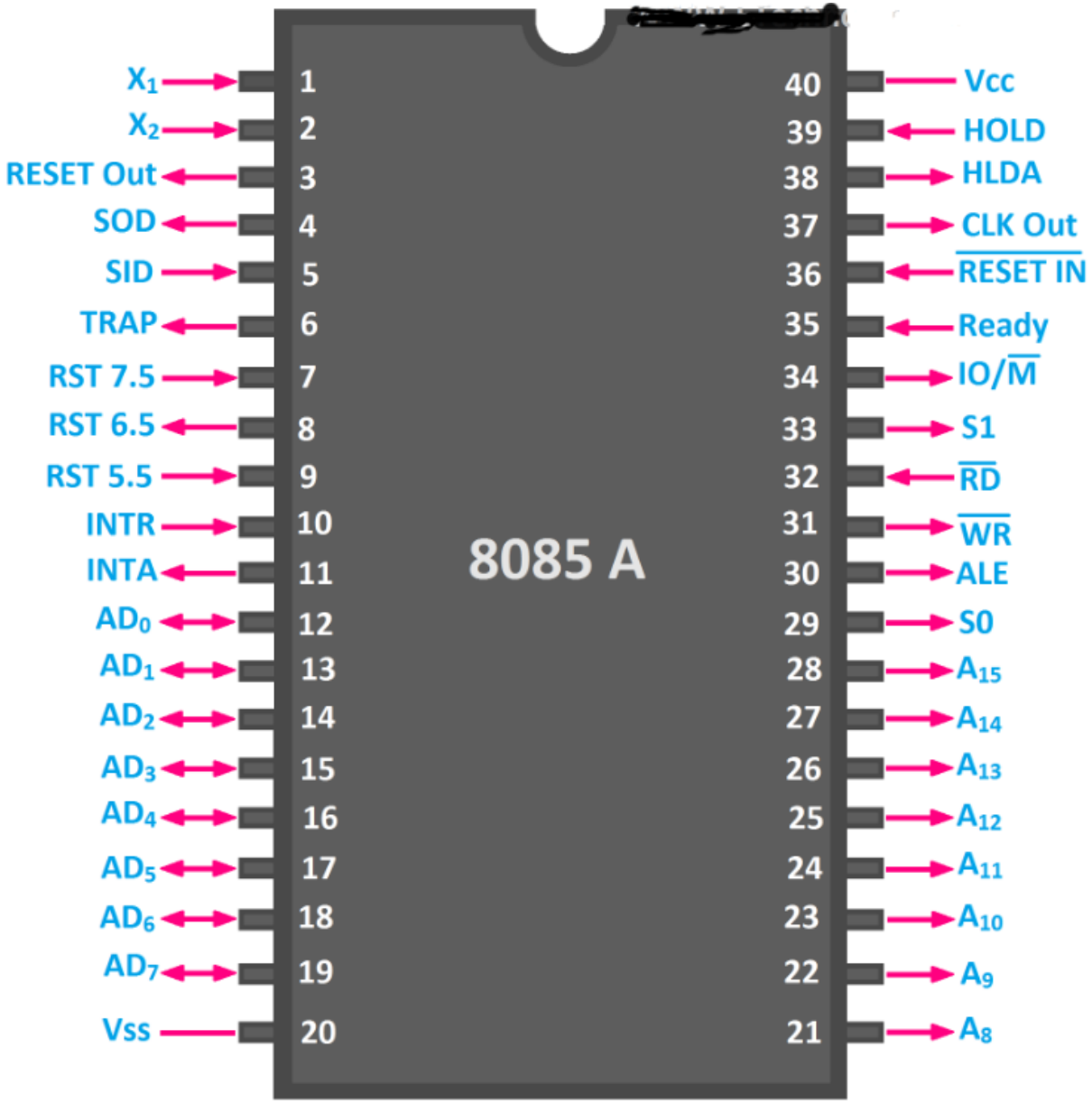
Architecture of 8085 Microprocessor

- **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.

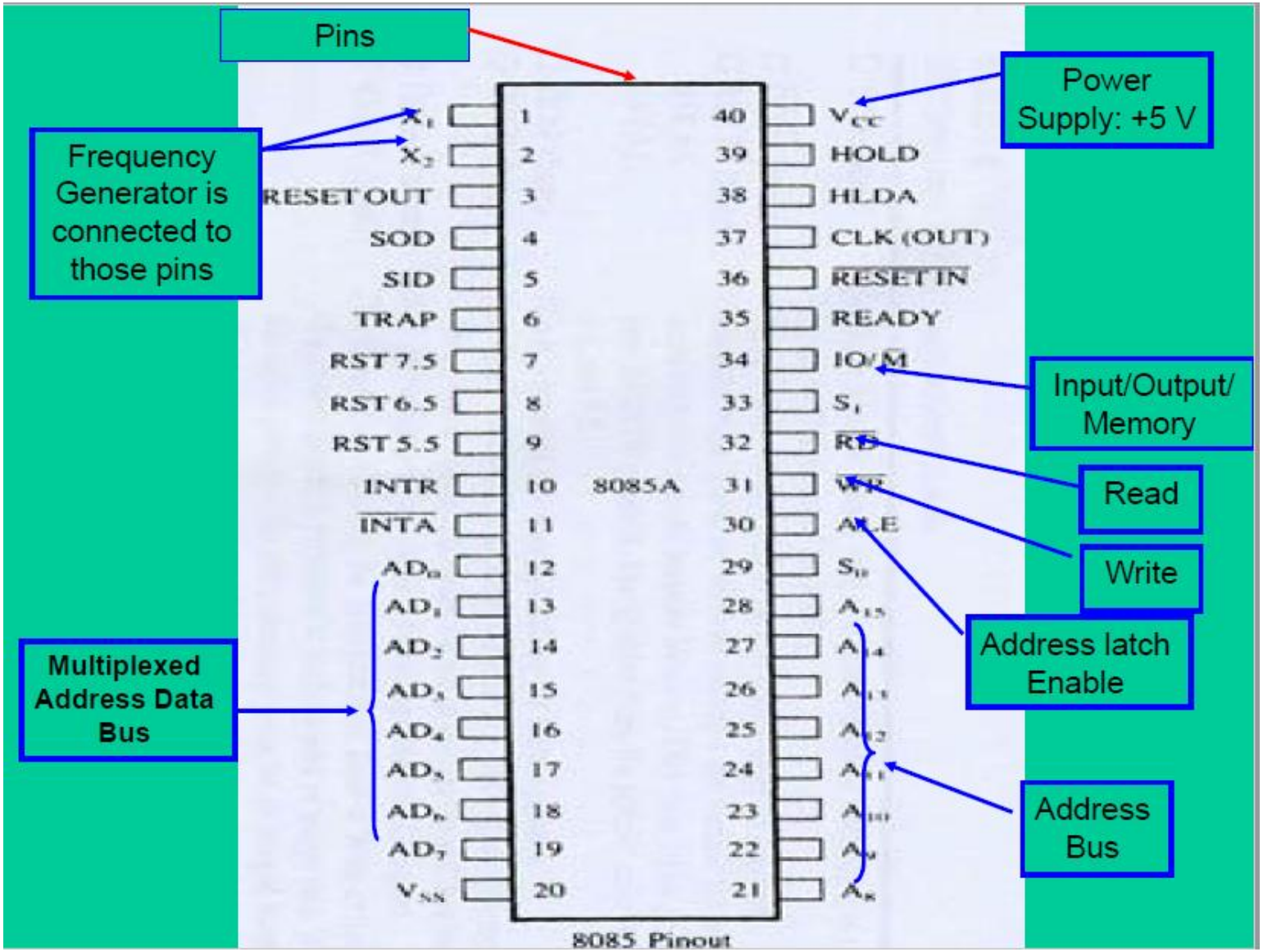
8085 Microprocessor Architecture

- 8-bit general purpose µp
- Capable of addressing 64 k of memory
- Has 40 pins
- Requires +5 v power supply
- Can operate with 3 MHz clock
- 8085 upward compatible

8085 Pin Configuration



8085 Pin Configuration



8085 Pin Configuration

- It is a Dual Inline Package(DIP) integrated circuit built with NMOS technology.
- The signals of this 40 pin IC is grouped into 7 categories, which are given below:
- Power supply and clock signals
- Data bus
- Address bus
- Serial I/O ports
- Control and status signals
- Interrupts and externally generated signals
- Direct memory access

8085 Pin Configuration Description

- **Power supply**
  - ✓ There are 2 power supply signals: VCC & VSS.
  - ✓ VCC indicates +5v power supply and VSS indicates ground signal.
- **V_{CC}** – Pin number 40 denotes V_{CC}, and an external power supply of + 5 V is provided at this pin.
- **V_{SS}** – Its pin number is 20. This pin shows the grounded connection of the microprocessor.
- **Clock signals**
  - ✓ There are 3 clock signals, i.e. X1, X2, CLK OUT.
  - – **X_1 and X_2** – These are represented by pin number 1 and 2 respectively in the pin configuration. These 2 pins are connected with a crystal or LC network to maintain the internal frequency of the clock generator.
  - – **CLK (OUT)** – It is the 37^{th} pin of the 8085 IC and acts as the system clock that keeps the record of time duration required by each operation to get completed.

8085 Pin Configuration Description

- **Address bus :** This category contains 8 pins.
  - ✓ A15-A8, it carries the most significant 8-bits of memory/IO address.
  - The address bus has 16 lines i.e.; it can carry 16 bits at a time. However, out of 16, 8 are multiplexed with the data bus and the leftover 8 are separately shown by pin number 21 to 28 in the pin configuration.
- **Data Bus with multiplexed address bus:** This category also contains 8 pins

✓ The size of the data bus of the 8085 microprocessor is 8 bits. However, to reduce the number of bus lines these 8-bit data bus lines are multiplexed with the 8-bit address bus.

✓ These are shown by pin number 12 to 19.

✓ The address bus is denoted by A whereas the data bus is denoted by D.

✓ The pin configuration denotes the lower order multiplexed address and data bus bits from AD_0 *to* AD_7.

8085 Pin Configuration Description

❑ Control and status signals

- ✓ These signals are used to identify the nature of operation.
- ✓ There are 3 control signal and 3 status signals.
- ✓ Three control signals are RD, WR & ALE.

a) **RD :** This pin is numbered 32 in the configuration and a low signal in this pin shows the read operation either from I/O devices or from the memory unit. Thereby indicating that the data bus is now in a state or position to accept the data from the memory or I/O devices.

b) **WR :** It is the 31st pin in the pin diagram and a low signal in this pin represents the write operation at the memory or I/O devices. This indicates that the data present in the data bus is to be written into the desired memory address or I/O device by the processor.

c) **ALE :** ALE is an acronym for address latch enable and is pin number 30 in the configuration. It is a positive going pulse generated when a new operation is started by the microprocessor. When the pulse goes high, it indicates address. When the pulse goes down it indicates data.

8085 Pin Configuration Description

- Three status signals are IO/M', S0 & S1.
- They help to show the current programming status of the microprocessor such as Halt, memory read-write, I/O read-write, opcode fetch, interrupt acknowledge, etc.

❖ **IO/M'**

- ✓ This signal is used to differentiate between IO and Memory operations, i.e. when it is high indicates IO operation and when it is low then it indicates memory operation.

❖ **S1 & S0**

- ✓ These signals are used to identify the type of current operation.
- ✓ The pins S_0 and S_1 represent the status signal at pin number 29 and 33 respectively.
- ✓ These signals show the type of recent operation of the microprocessor.

8085 Pin Configuration Description

- **Interrupts & externally initiated signals**
  - ✓ Interrupts are the signals generated by external devices to request the microprocessor to perform a task.
  - ✓ In the pin configuration, 5 types of interrupts are shown by 5 different pins from pin number 6 to 10. These pins are used to manage the interrupt.
  - ✓ There are 5 interrupt signals, i.e. TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR.
- **RESET IN**: It is pin number 36 in the pin diagram. An active low signal at this pin resets the PC of the microprocessor to 0. Or we can say, after resetting the PC holds its initial memory address.
- **RESET OUT:** It is the 3rd pin in the pin diagram. This pin generates a signal to provide information about the resetting of the microprocessor. Also, we can say that once a processor is reset then all the connected devices must also be reset.
- So, enabling this signal shows the resetting of the interconnected devices.
- **INTA**: It is the 11th pin of the 8085 pin configuration. A signal at this pin acknowledges the generated interrupt.

8085 Pin Configuration Description

❑ Serial I/O signals

- ✓ There are 2 serial signals, i.e. SID (Serial input data line) and SOD (Serial output data line) and these signals are used for serial communication.

- **SID**: SID denotes serial input data pin and its pin is numbered as 5. With this pin, data is serially fed to the processor directly through the input devices.
- **SOD**: SOD denotes serial output data pin and its pin number is 4, in the pin configuration of 8085. Once the data is processed in the microprocessor then this pin represents bit by bit results at the output devices.

8085 Pin Configuration Description

❑ Direct Memory Access (DMA) :

- We are aware of the fact that memory and I/O devices are connected with each other by the microprocessor. So, the intermediator i.e., CPU manages the data transfer between the input-output device and memory.
- However, when data in a large amount is to be transferred between I/O devices and memory the CPU gets disabled by tri-stating its buses. And this transfer is manageable by external control circuits.
- The DMA has 2 pins.
- **HOLD**: This signal is generated at pin number 39.

✓ This signal indicates that a peripheral such as DMA controller is requesting the use of the address and data buses.

- **HLDA**: This signal is generated at pin number 38.

✓ This signal is enabled at the time when the processor gets HOLD signal and it releases HLDA i.e., hold acknowledge signal.

✓ Indicates that the CPU has received the Hold request and acknowledge that request.

- **READY**: This is the 35^{th} numbered pin in the pin diagram that maintains synchronization between the processor and peripherals, memory.
  - ✓ It is the pin through which the external hardware tells or sends the message that they are ready to communicate with the microprocessor.

Flag and Flag Register

- Special Purpose Register.
- The flag register reflected by result stored in accumulator
- Shows the status of the microprocessor before/after an operation
- In 8085 microprocessor, flag register consists of 8 bits and only 5 of them are useful.
- S (sign flag), Z (zero flag), AC (Auxillary carry flag), P (parity flag) & CY (carry flag)
- These flags have critical importance in the decision-making process of the microprocessor.

Flag Register: Sign Flag

- Used for indicating the sign of the data in the accumulator

✓ The sign flag is set if negative (1 –negative)

✓ The sign flag is reset if positive (0 –positive)

- **Example:**
- MVI A 30 (load 30H in register A)
  MVI B 40 (load 40H in register B)
  SUB B (A = A – B)
  These set of instructions will set the sign flag to 1 as 30 – 40 is a negative number.
- MVI A 40 (load 40H in register A)
  MVI B 30 (load 30H in register B)
  SUB B (A = A – B)
  These set of instructions will reset the sign flag to 0 as 40 – 30 is a positive number

Flag Register: Zero Flag

- Is set if result obtained after an operation is 0
- Is set following an increment or decrement operation of that register
  - ✓ 1- zero result
    0- non-zero result
- **Example:**
- MVI A 10 (load 10H in register A)
  SUB A (A = A – A)
  These set of instructions will set the zero flag to 1 as 10H – 10H is 00H

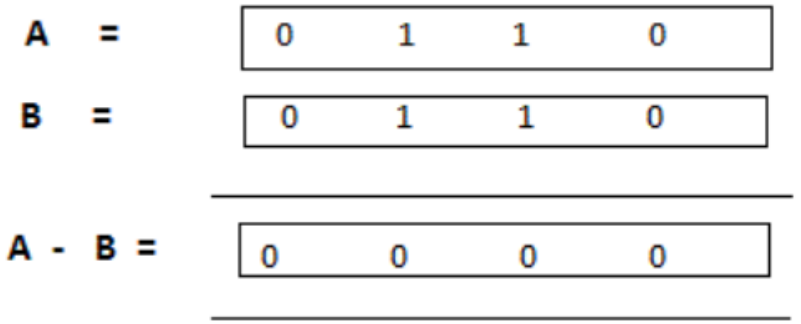
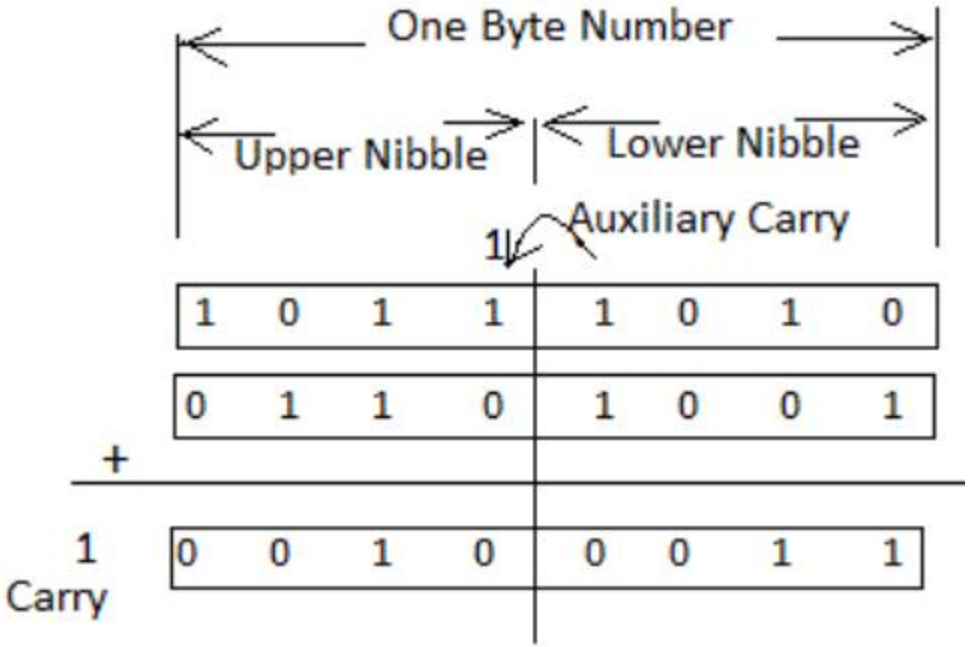


Figure : Example of Zero Flag

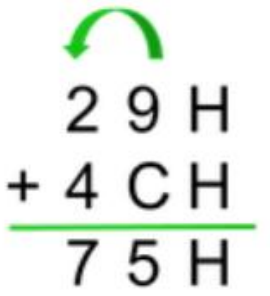
Flag Register: **Auxiliary Carry Flag (AC)**

- This flag is used in BCD number system(0-9).
- If after any arithmetic or logical operation D(3) generates any carry and passes on to D(4) this flag becomes set i.e. 1, otherwise it becomes reset i.e. 0.
- This is the only flag register which is not accessible by the programmer
  - ✓ 1: carry out from bit 3 on addition or borrow into bit 3 on subtraction
  - ✓ 0: otherwise

Flag Register: **Auxiliary Carry Flag (AC)**



: Example of Auxiliary Carry



9 + C(12) = 16(carry) + 5

Auxiliary Carry Flag in Hexadecimal Representation

29H = 0010 1001
+4CH = 0100 1100
75H = 0111 0101

There there is carry generated and forwarded to next nibble, so the auxiliary carry flag is set to one.

Flag Register: **Parity Flag (P)**

- If after any arithmetic or logical operation the result has even parity, an even number of 1 bits, the parity register becomes set i.e. 1, otherwise it becomes reset i.e. 0.
  - ✓1: accumulator has even number of 1 bits
  - ✓0: accumulator has odd parity
- **Example:**
- MVI A 05 (load 05H in register A)
  - This instruction will set the parity flag to 1 as the BCD code of 05H is 00000101, which contains even number of ones i.e. 2.

Flag Register: **Carry Flag (CY)**

- Carry is generated when performing n bit operations and the result is more than n bits, then this flag becomes set i.e. 1, otherwise it becomes reset i.e. 0.
- During subtraction (A-B), if A>B it becomes reset and if (A<B) it becomes set.
- Carry flag is also called borrow flag.
  - ✓ 1: Carry out from MSB bit on addition or borrow into MSB bit on subtraction
  - ✓ 0: No carry out or borrow into MSB bit

	D7	D6	D5	D4	D3	D2	D1	D0
	1	1	1	1	0	0	0	0
	0	1	1	1	1	0	0	0
Carry	1	0	0	0	0	1	0	0	0

How to find Flag

- A =54 H
- B =23 H

54H		0	1	0	1		0	1	0	0
23H		0	0	1	0		0	0	1	1
77H		0	1	1	1		0	1	1	1

- S=
- Z=
- Ac=
- P=
- Cy=

Flag Register: Question

The flag register of 8085 microprocessor contains the data 45H . Interpret its meaning

F= 45 H

F = 0100 0101

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	-	Ac	-	p	-	Cy
0	1	0	0	0	1	0	1

S=0 , means Ans is positive

Z=1, means Ans is equal to Zero

Ac=0, means there is no carry from D3 to D4

P=1, means in ans. there even no. of 1

Cy =1, means there is carry or borrow from MSB during execution

Multiplexing and De-multiplexing of address/data bus

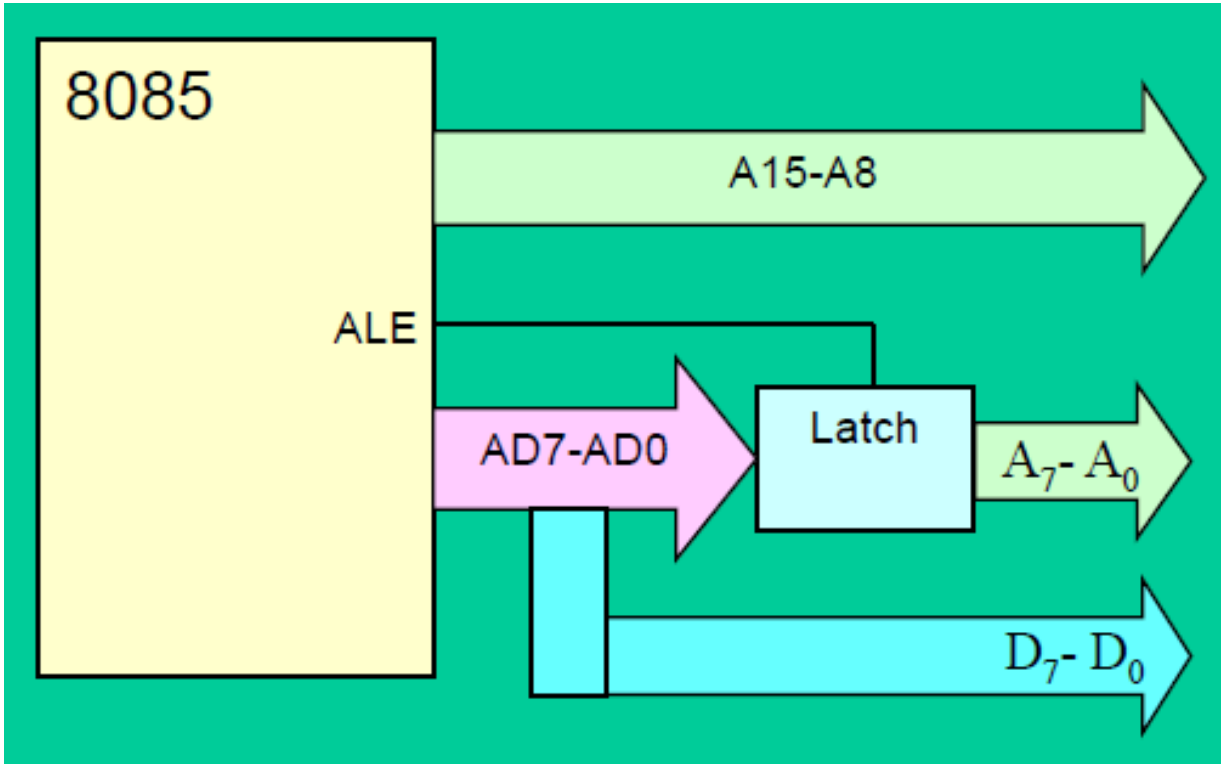
The Address and Data Busses

- The address bus has 8 signal lines A8–A15 which are unidirectional.
- The other 8 address bits are multiplexed (time shared) with the 8 data bits.
  - So, the bits AD0–AD7 are bi-directional and serve as A0–A7 and D0 –D7 at the same time.
- During the execution of the instruction, these lines carry the address bits during the early part, then during the late parts of the execution, they carry the 8 data bits.
  - In order to separate the address from the data, we use a latch to save the value before the function of the bits changes.

Demultiplexing of AD7-AD0

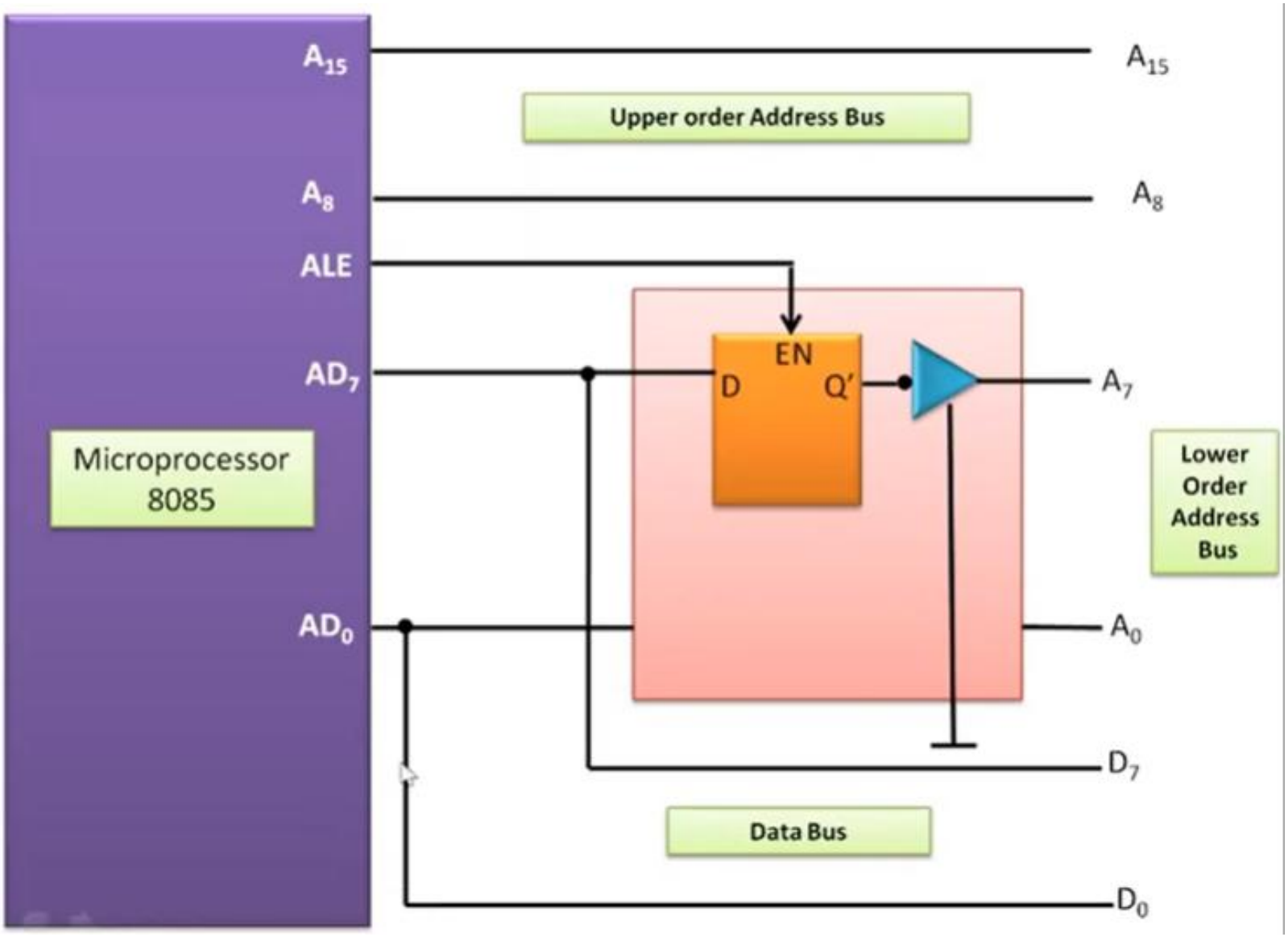
- From the above description, it becomes obvious that the AD7–AD0 lines are serving a dual purpose and that they need to be de-multiplexed to get all the information.
- In order to use the address bus, De-multiplexing of the lower order Address/Data bus is essential.
- The high order bits of the address remain on the bus for three clock periods. However, the low order bits remain for only one clock period and they would be lost if they are not saved externally. Also, notice that the low order bits of the address disappear when they are needed most.
- To make sure we have the entire address for the full three clock cycles, we use an external latch to save the value of AD7–AD0 when it is carrying the address bits.
  - ✓ We use the ALE signal to enable this latch.

Demultiplexing AD7-AD0

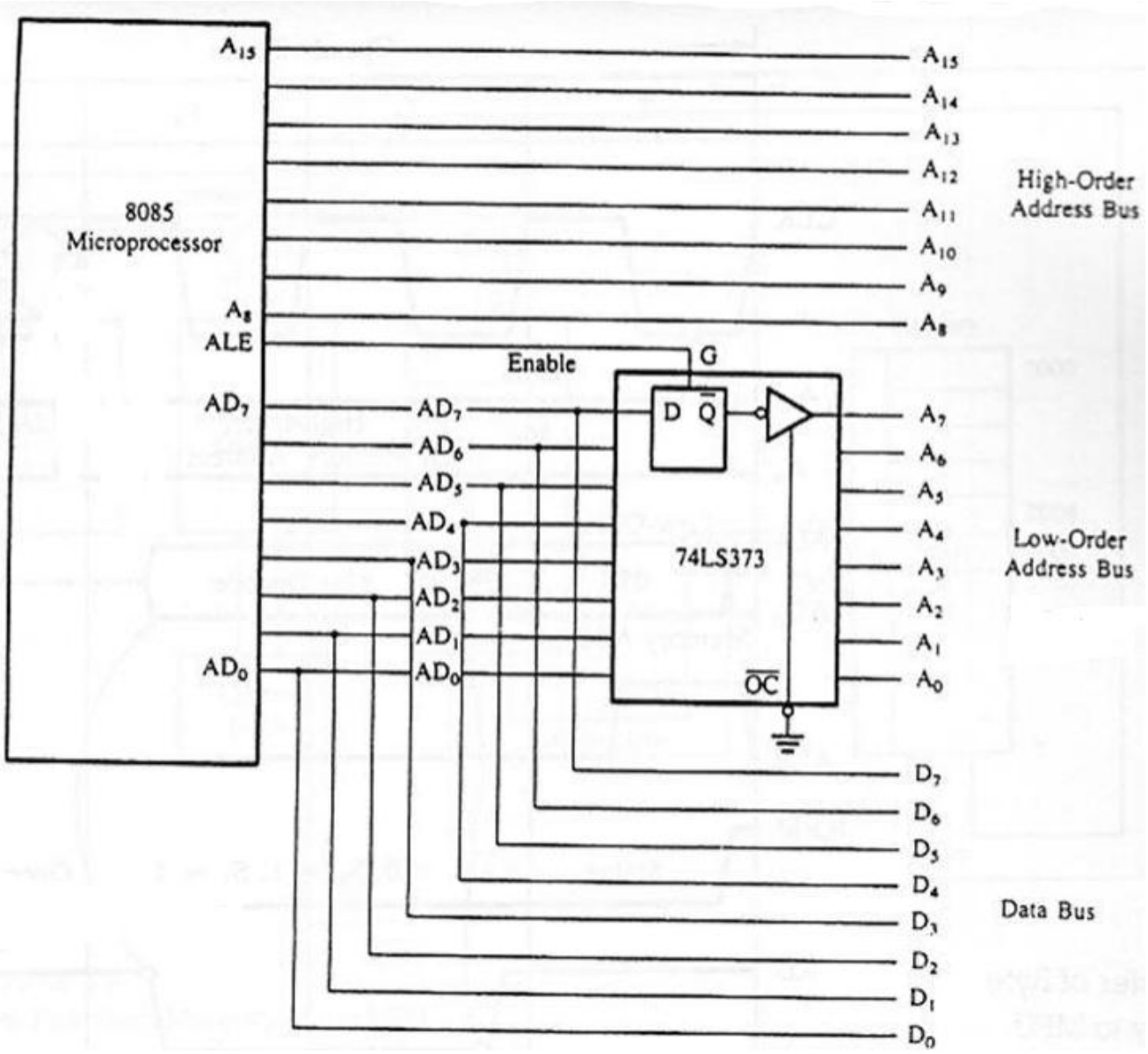


- Given that ALE operates as a pulse during T1, we will be able to latch the address. Then when ALE goes low, the address is saved and the AD7–AD0 lines can be used for their purpose as the bi-directional data lines.

Demultiplexing AD7-AD0



Demultiplexing AD7-AD0



Demultiplexing the Bus AD7–AD0

- The high order address is placed on the address bus and hold for 3 clock periods.
- The low order address is lost after the first clock period, this address needs to be hold however we need to use latch.
- The address AD7–AD0 is connected as inputs to the latch 74LS373.
- The ALE signal is connected to the enable (G) pin of the latch and the OC (Output control) of the latch is grounded.

Status Signal

- It is used to indicate the current status of the processor.
- basic status signals that indicate the operation that the microprocessor is performing at any given stage.

$IO/\bar{M}$	S_1	S_0	States
0	0	1	Memory write
0	1	0	Memory read
1	0	1	I/O write
1	1	0	I/O read
0	1	1	Opcode fetch

S_1	S_0	Operations
0	0	HALT
0	1	WRITE
1	0	READ
1	1	FETCH

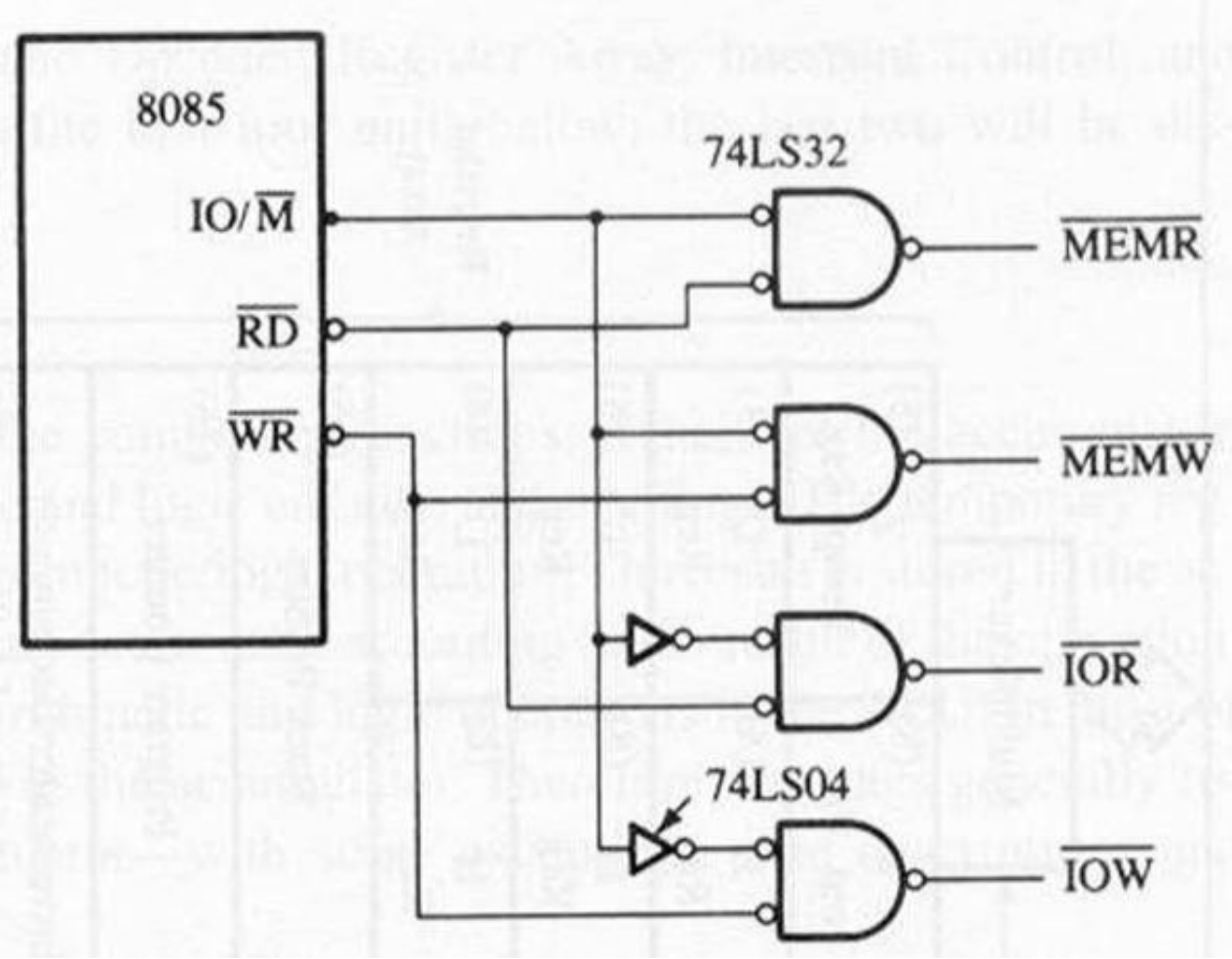
Status code for Intel 8085

Generation of Control Signals in 8085

- The 8085 Microprocessor provides RD and WR signals to initiate read or write cycle.
- Because these Control Signals of 8085 are used both for reading/writing memory and for reading/writing an input device, it is necessary to generate separate read and write signals for memory and I/O devices.
- The 8085 provides IO/M signal to indicate whether the initiated cycle is for I/O device or for memory device.
- Using IO/M signal along with RD and WR, it is possible to generate separate four Control Signals of 8085 :

$\overline{\text{MEMR}}$	(Memory Read)	: To read data from memory.
$\overline{\text{MEMW}}$	(Memory Write)	: To write data in memory.
$\overline{\text{IOR}}$	(I/O Read)	: To read data from I/O device.
$\overline{\text{IOW}}$	(I/O Write)	: To write data in I/O device.

Generation of Control Signals in 8085



Generation of Control Signals in 8085

- Table shows the truth table used to generate MEMR, MEMW, IOR and IOW signals.
- The signal IO/M goes low for memory operation. This signal is logically Negative NANDed (ORed) with RD and WR to get MEMR and MEMW signals.
- When both RD and IO/M signals go low, MEMR signal goes low.
- When both WR and IO/M Signals go low, MEMW signal goes low.
- To generate IOR and IOW signals for I/O operation, IO/M signal is first inverted and then logically Negative NANDed (ORed) with RD and WR signals.

- Note: OR Gate is functionally connected as negative NAND Gate.

$IO/\overline{M}$	$\overline{RD}$	$\overline{WR}$	Operation
0	0	1	$\overline{MEMR}$
0	1	0	$\overline{MEMW}$
0	1	1	NOP
1	0	1	$\overline{IOR}$
1	1	0	$\overline{IOW}$
1	1	1	NOP

8085 Interrupts

- Interrupt is an external signal initiated by peripheral device to get attention of CPU.
- Interrupt is a process where an external device can get attention of the microprocessor.
- An interrupt is considered to be an emergency signal that may be serviced.
- When the microprocessor receives an interrupt signal, it suspends the currently executing program and jumps to an interrupt service routine(ISR) to respond to the incoming interrupt.
- *TRAP, RST 7.5, RST 6.5, RST 5.5, INTR, INTA*

Interrupts: Hardware and Software Interrupts

- When microprocessors receive interrupt signals through pins (hardware) of microprocessor, they are known as *Hardware Interrupts*.
- There are 5 Hardware Interrupts in 8085 microprocessor.
- They are – *INTR, RST 7.5, RST 6.5, RST 5.5, TRAP*
- *Software Interrupts* are those which are inserted in between the program which means these are mnemonics of microprocessor.
- There are 8 software interrupts in 8085 microprocessor. They are – *RST 0, RST 1, RST 2, RST 3, RST 4, RST 5, RST 6, RST 7*.

Interrupts: Vectored and Non-Vectored Interrupts

- *Vectored Interrupts* are those which have fixed vector address (starting address of sub-routine) and after executing these, program control is transferred to that address.
  - ✓ Vector Addresses are calculated by the formula 8 * TYPE
- *Non-Vectored Interrupts* are those in which vector address is not predefined. The interrupting device gives the address of sub-routine for these interrupts.
  - ✓ The address of the memory location is sent along wit the interrupt
  - ✓ *INTR* is the only non-vectored interrupt in 8085 microprocessor.

INTERRUPT	VECTOR ADDRESS
TRAP (RST 4.5)	24 H
RST 5.5	2C H
RST 6.5	34 H
RST 7.5	3C H

Interrupts: Maskable and Non-Maskable Interrupts

- *Maskable Interrupts* are those which can be disabled or ignored by the microprocessor.
  - ✓ These interrupts are either edge-triggered or level-triggered, so they can be disabled.
  - ✓ *INTR, RST 7.5, RST 6.5, RST 5.5* are maskable interrupts in 8085 microprocessor.
- Non-Maskable Interrupts are those which cannot be disabled or ignored by microprocessor.
  - ✓ *TRAP* is a non-maskable interrupt.
  - ✓ It consists of both level as well as edge triggering and is used in critical power failure conditions.

Priority of Interrupts

- When microprocessor receives multiple interrupt requests simultaneously, it will execute the interrupt service request (ISR) according to the priority of the interrupts.

Externally Initiated Signals

- In addition to interrupt, three pins RESET,HOLD and READY accept the externally initiated signal as input.
- To respond to the HOLD request,8085 has one signal called HLDA(Hold Acknowledge)

Ready

- This signal indicates that the device is ready to send or receive data. If READY is low, then the CPU has to wait for READY to go high.

HOLD

- This signal indicates that another master is requesting the use of the address and data buses.

HLDA

- It indicates that the CPU has received the HOLD request and it will relinquish the bus in the next clock

Externally Initiated Signals

INTA

- It is an interrupt acknowledgment signal.

RESET IN

- This signal is used to reset the microprocessor by setting the program counter to zero.

RESET OUT

- This signal is used to reset all the connected devices when the microprocessor is reset.

The 8085 Addressing Modes

- To perform any operation, we have to give the corresponding instructions to the microprocessor.
- In each instruction, programmer has to specify 3 things:
  - Operation to be performed.
  - Address of source of data.
  - Address of destination of result.
- The method by which the address of source of data or the address of destination of result is given in the instruction is called **Addressing Modes.**
- The term addressing mode refers to the way in which the operand of the instruction is specified.

Types of Addressing Modes

- Intel 8085 uses the following addressing modes:
  1. Direct Addressing Mode
  2. Register Addressing Mode
  3. Register Indirect Addressing Mode
  4. Immediate Addressing Mode
  5. Implicit Addressing Mode

Direct Addressing Mode

- In direct addressing mode, the data to be operated is available inside a memory location and that memory location is directly specified as an operand

- In this mode, the address of the operand is given in the instruction itself.

LDA 2500 H	**Load the contents of memory location 2500 H in accumulator.**

  - LDA is the operation.
  - 2500 H is the address of source.
  - Accumulator is the destination.

Register Addressing Mode

- In register addressing mode, the data to be operated is available inside the register(s) and register(s) is(are) operands. Therefore, the operation is performed within various registers of the microprocessor.
- In this mode, the operand is in general purpose register.
- Examples:

MOV A, B	Move the contents of register B to A.

  - MOV is the operation.
  - B is the source of data.
  - A is the destination.

- ADD B (add contents of registers A and B and store the result in register A)
- INR A (increment the contents of register A by one)

Register Indirect Addressing Mode

- In register indirect addressing mode, the data to be operated is available inside a memory location and that memory location is indirectly specified by a register pair.
- In this mode, the address of operand is specified by a register pair.

MOV A, M	Move data from memory location specified by H-L pair to accumulator.

  - MOV is the operation.
  - M is the memory location specified by H-L register pair.
  - A is the destination.

Immediate Addressing Mode

- In this mode, the operand is specified within the instruction itself.

MVI A, 05 H	Move 05 H in accumulator.

- MVI is the operation.
- 05 H is the immediate data (source).
- A is the destination.

Implicit Addressing Mode

- In implied/implicit addressing mode the operand is hidden and the data to be operated is available in the instruction itself.
- If address of source of data as well as address of destination of result is fixed, then there is no need to give any operand along with the instruction.

CMA	Complement accumulator.

  - CMA is the operation.
  - A is the source.
  - A is the destination.

- RRC (rotate accumulator A right by one bit)
- RLC (rotate accumulator A left by one bit)

8086 Microprocessor

Introduction to 8086 Microprocessor

- 8086 Microprocessor is an enhanced version of 8085 Microprocessor that was designed by Intel in 1976.
- It is a 16-bit Microprocessor having 20 address lines and 16 data lines that provides up to 1MB storage.
- It consists of powerful instruction set, which provides operations like multiplication and division easily.
- It supports two modes of operation, i.e. **Maximum mode and Minimum mode**. **Maximum mode** is suitable for system having multiple processors and **Minimum mode** is suitable for system having a single processor.

Features 8086 Microprocessor

- It needs 5-MHz clock cycle for 8086, 8-MHz for 8086-2 and 10-MHz for 8086-1
- It has an **instruction queue**, which is capable of **storing six instruction bytes** from the memory resulting in faster processing.
- It was the first 16-bit processor having 16-bit ALU, 16-bit registers, internal data bus, and 16-bit external data bus resulting in faster processing.
- It is possible to perform bit, byte word and block operation in 8086. It performs the arithmetic and logic operations on bit, byte and decimal numbers including multiply and divide.
- The INTEL 8086 microprocessor architecture supports Multi-programming.

- It uses **two stages of pipelining**, i.e. **Fetch Stage and Execute Stage**, which improves performance. **Fetch stage** can prefetched up to 6 bytes of instructions and stores them in the queue. **Execute stage** executes these instructions.
- It consists of **29,000 transistors**.
- It can address upto **1 MB** of memory.

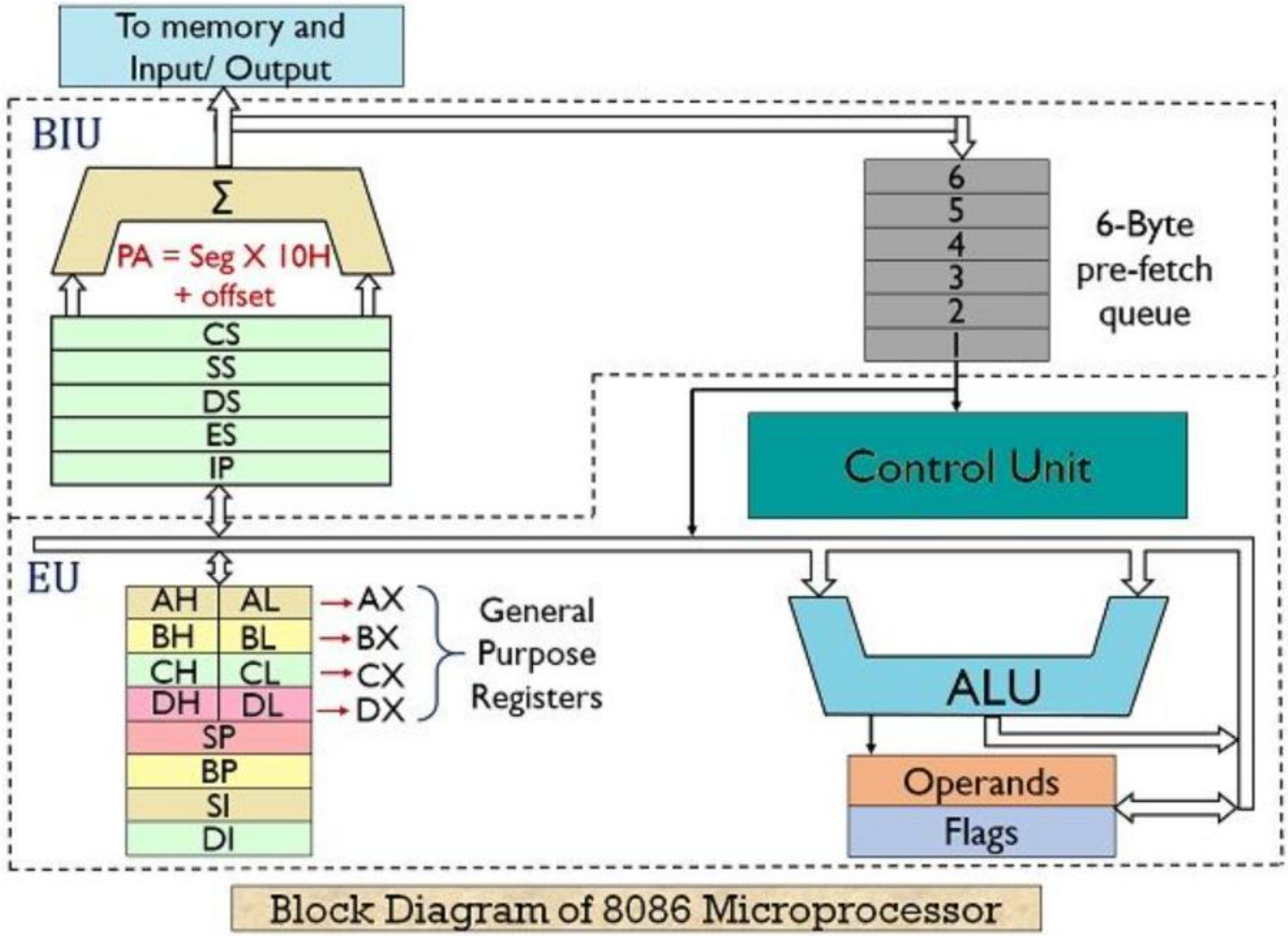
Comparison between 8085 & 8086

SN	8085 Microprocessor	8086 Microprocessor
1	It is an 8 bit microprocessor.	It is a 16 bit microprocessor.
2	It has 16 bit address line.	It has 20 bit address line.
3	It has 8- bit data bus.	It has 16- bit data bus.
4	The memory capacity is 64 KB.	The memory capacity is 1 MB.
5	Clock speed of this microprocessor is 3 MHz.	Clock speed of this microprocessor varies between 5, 8 and 10 MHz for different versions.
6	It has 5 flags.	It has 9 flags.
7	8085 microprocessor does not support memory segmentation.	8086 microprocessor supports memory segmentation.
8	It does not support pipelining.	It supports pipelining.
9	It is accumulator based processor.	It is general purpose register based processor

Comparison between 8085 & 8086

SN	8085 Microprocessor	8086 Microprocessor
10	It has no minimum or maximum mode.	It has minimum and maximum modes.
11	In 8085, only one processor is used.	In 8086, more than one processor is used. Additional external processor can also be employed.
12	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.
13	The cost of 8085 is low.	The cost of 8086 is high.

Block Diagram of 8086 Microprocessor



Block Diagram of 8086 Microprocessor

Block Diagram of 8086 Microprocessor

- The architecture of 8086 microprocessor is composed of 2 major units, the BIU i.e., Bus Interface Unit and EU i.e., Execution Unit.
- The figure above shows the block diagram of the architectural representation of the 8086 microprocessor:

Bus Interface Unit (BIU)

- The segment registers, instruction pointer and 6-byte instruction queue are associated with **the bus interface unit (BIU).**
- The BIU:

✓ Handles transfer of data and addresses,

✓ Fetches instruction codes, stores fetched instruction codes in first-in-first-out register set called a queue,

✓ Finds the physical address of that location in the memory where the instruction is stored

✓ Manages the 6-byte pre-fetch queue where the pipelined instructions are stored.

✓ Reads data from memory and I/O devices,

✓ Writes data to memory and I/O devices

BIU: Address/Data Bus

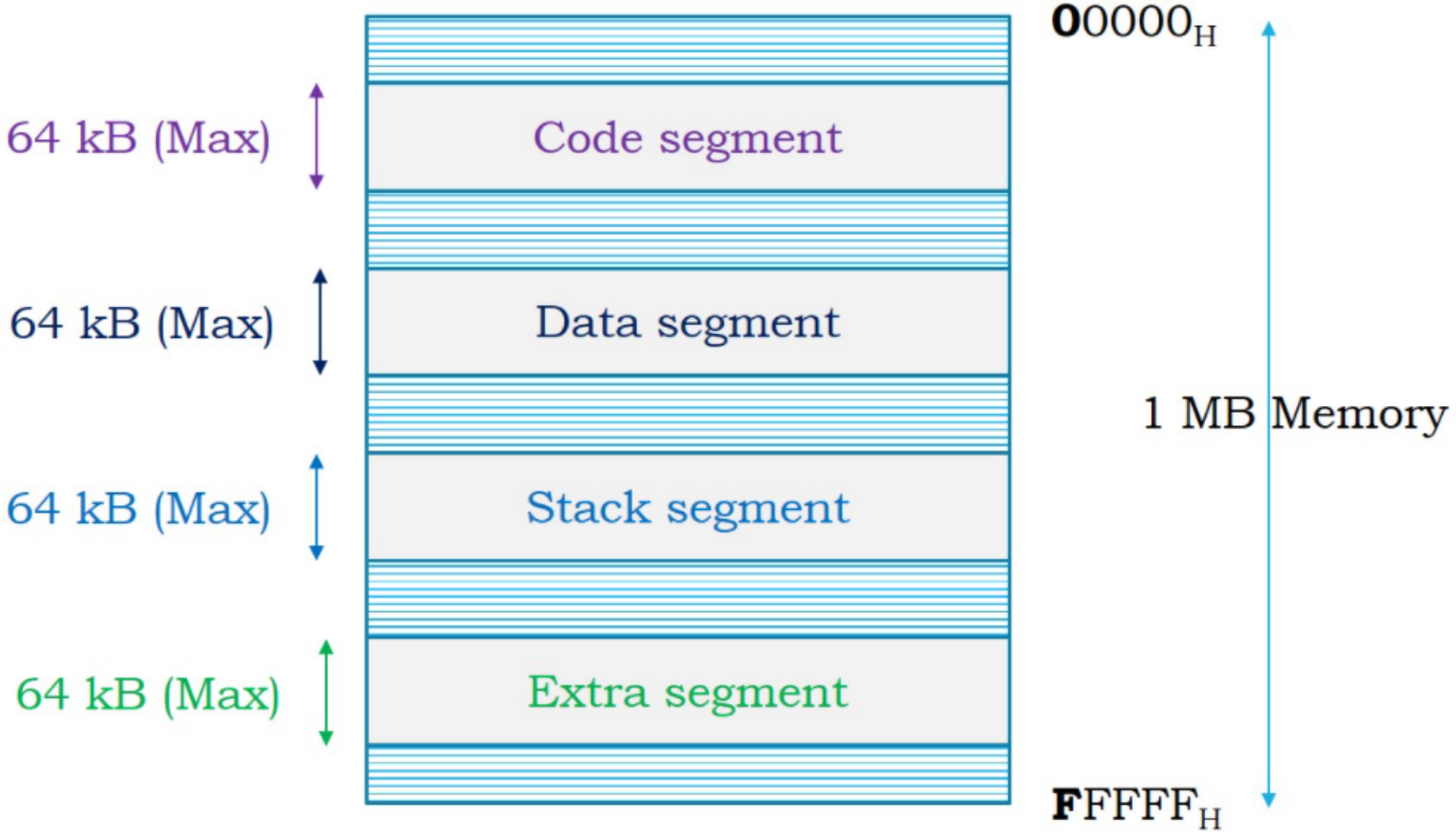
- These lines serve two functions.
  - ✓ As an address bus is 20 bits long and consists of signal lines A0 through A19.
  - ✓ A19 represents the MSB and A0 LSB.
- A 20 bit address gives the 8086 a 1Mbyte memory address space. More over it has an independent I/O address space which is 64K bytes in length.
- The 16 data bus lines D0 through D15 (D0 through D7 in 8088) are actually multiplexed with address lines A0 through A15 respectively.
- By multiplexed we mean that the bus work as an address bus during first machine cycle and as a data bus during next machine cycles. D15 is the MSB and D0 LSB.
- When acting as a data bus, they carry read/write data for memory, input/output data for I/O devices, and interrupt type codes from an interrupt controller.

Bus Interface Unit (BIU):Segmentation

- **Memory Segmentation**
  - ✓ To increase execution speed and fetching speed, 8086 segments the memory.
  - ✓ This 1 MB memory is divided into 16 Segment memories. The capacity of each memory segment is 64 KB. But 8086 can access at a time only memory segment
  - ✓ 8086 works only with four 64KB segments within the whole 1MB memory.
- There are 4 segment registers in 8086 as given below:
- **Code Segment Register (CS):** It is a 16-bit register and holds the address of the instruction or program stored in the code segment of the memory.
- **Data Segment Register (DS):** It holds the address of the data segment. The data, variables and constants given in the program are held in the data segment of the memory.
- The data segment stores the data in the memory whose address is present in this 16-bit register.
- **Stack Segment Register (SS):** Stack segment holds addresses and data of subroutines. It also holds the contents of registers and memory locations given in PUSH instruction.
- **Extra Segment Register (ES):** Here the starting address of the extra segment is present. Extra segment holds the destination addresses of some data of certain string instructions.

Memory Segmentation in 8086

The function four segment registers – CS, DS, SS and ES is to indicate the starting or base address of the code segment, data segment, stack segment and extra segment respectively in the memory.



Advantages and Disadvantages of Memory Segmentation

- Advantages:
- In 8086 microprocessor architecture it permits its programmer to access 1MB memory even though address associated with the Instruction is 16 bit.
- This makes easier interface with 8 and 16 bit wide memory boards and with 16 bit registers in the 8086.
- Instruction, data, stack of a program can be more than 64KB memory in 8086 microprocessor architecture
- 8086 microprocessor architecture permits separate memory area for instruction, data and stack. So one program can work on different sets of data.
- This method is very useful during the multitasking in 8086 microprocessor architecture
- Disadvantages:
- Even though memory capacity is 16*64KB, microprocessor can access at a time only 4*64 KB of memory.
- The segment: offset scheme introduces complexity in hardware and software design.

Bus Interface Unit (BIU) :IP

- Also, the IP in the block diagram is the instruction pointer which is a default register that is used by the processor in order to get the desired instruction.
- The **IP contains the offset address** of the next byte that is to be taken from the code segment.
- • IP is associated with CS register as (CS:IP)
- It is to be noteworthy that the physical address of the instruction is achieved by combining the segment address with that of the offset address.
- For E.g. - content of CS is 4000H and content of IP is 1250H. Now the physical address is given as.

$$PA = (4000)_{16} \times (16)_{10} + (1250)_{16}$$
$$= (40000)_{16} + (1250)_{16}$$
$$= (41250)_{16}$$

or

$$\begin{array}{r} 4000\blacksquare \\ 1250 \\ \hline \end{array}$$
$$PA = (41250)H$$

Bus Interface Unit (BIU): Pre Fetch Queue

- This queue is used in 8086 in order to perform pipelining.
- As at the time of decoding and execution of the instruction in EU, the BIU fetches the sequential upcoming instructions and stores it in this queue.
- The size of this queue is 6-byte. This means at maximum a 6-byte instruction can be stored in this queue. The queue exhibits **FIFO** behavior., first in first out.
- The BIU fetches this when EU is executing the current instruction.
- When EU is ready for next instruction, it simply reads from queue register instead of from memory. Therefore speed is increased.
- This is done in order to speedup the execution by overlapping instruction fetch with execution. This mechanism is known as pipelining.
- Fetching the next instruction while the current instruction executes is called pipelining.

Execution Unit(EU)

- The EU contains control circuitry which directs internal operations.
- The EU receives opcode of an instruction from the queue, decodes it and then executes it.
- While Execution, unit decodes or executes an instruction, then the BIU fetches instruction codes from the memory and stores them in the queue.
- It consists of following:
- **General Purpose Registers:**
  - ✓ There are four 16-bit general purpose registers: AX (Accumulator Register), BX (Base Register), CX (Counter) and DX.
- **AX (Accumulator)**
  - ✓ This is accumulator register. It gets used in arithmetic, logic and data transfer instructions. In manipulation and division, one of the numbers involved must be in AX or AL
- **BX (Base Register)**
  - ✓ This is base register. BX register is an address register. It usually contain a data pointer used for based, based indexed or register indirect addressing.

Execution Unit(EU)

- **CX (Count register)**
  - ✓ This is Count register. This serves as a loop counter. Program loop constructions are facilitated by it. Count register can also be used as a counter in string manipulation and shift/rotate instruction.
- **DX (Data Register)**
  - ✓ This is data register. Data register can be used as a port number in I/O operations. It is also used in multiplication and division.

Execution Unit(EU)

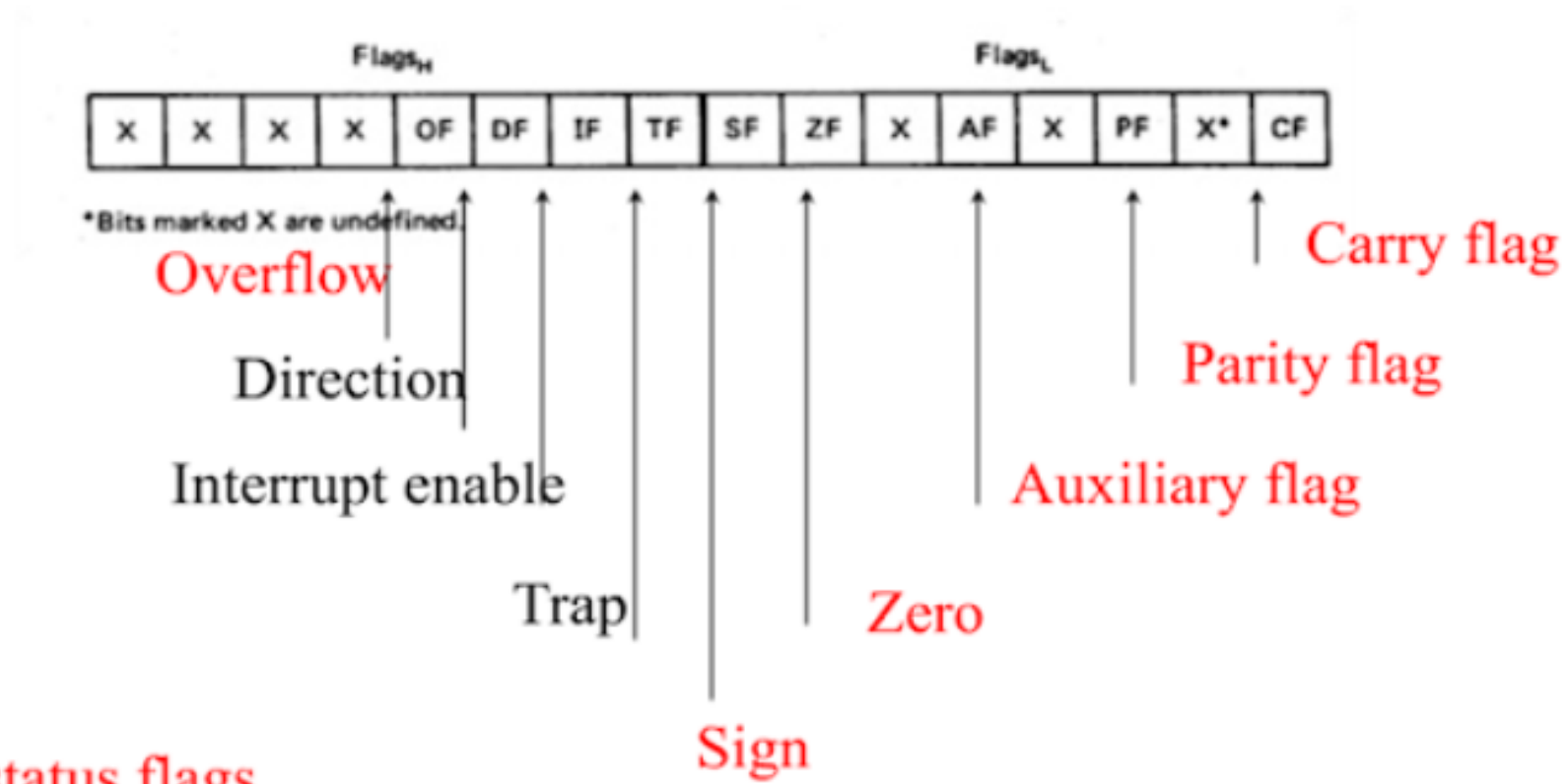
- **Index Register:**
  - ✓ The following four registers are in the group of pointer and index registers: Stack Pointer (SP), Base Pointer (BP), Source Index (SI), Destination Index (DI).
  - ✓ In this microprocessor architecture Pointer and index register are used to store 16-bit offset address.
- **ALU:**
  - ✓ It handles all arithmetic and logical operations such as addition, subtraction, multiplication, division, AND, OR, NOT operations.

SEGMENT	SEGMENT REGISTER	OFFSET REGISTER
Code Segment	CSR	**Instruction Pointer (IP)**
Data Segment	DSR	**Source Index (SI)**
Extra Segment	ESR	**Destination Index (DI)**
Stack Segment	SSR	**Stack Pointer (SP) / Base Pointer (BP)**

Execution Unit(EU)

- **Flag Register:**
  - ➢ It is a 16 bit register which exactly behaves like a flip-flop, means it changes states according to the result stored in the accumulator.
  - ➢ It has **9 flags** and they are divided into **2 groups i.e. conditional and control flags**.
  - ✓ **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

Flags



6 are status flags

3 are control flag

8086 Flag: Conditional Flag

- **Carry flag**:
  - ✓ This flag is set whenever there is an overflow from MSB after performing arithmetic operation, otherwise it is reset. For 8-bit operation overflow is from B7 bit for 16-bit operation overflow is from B15 bit.
- **Parity Flag**:
  - ✓ This flag is set if lower order 8-bit of the result consists of even number of 1's, otherwise it is reset.
- **Auxiliary Carry flag**:
  - ✓ This flag is set if there is overflow from lower 4-bits after performing arithmetic operation, otherwise it is reset.
- **Zero Flag**:
  - ✓ This flag is set if the result is zero after performing ALU operation, otherwise it is reset.
- **Sign flag**:
  - ✓ It is set if MSB of the result is equal to 1 after performing ALU operation, otherwise it is reset. (for 8-bit data it is set when B7=1, for 16-bit data it is set when B15=1).

8086 Flag: Conditional Flag

- **Overflow Flag**:
  - ✓ This flag is set, if the result cannot be stored in destination location, otherwise it is reset. This flag is checked during signed arithmetic.
  - ✓ It will set when
  - ➢ Overflow from B6 to B7, AND no overflow from B7(for 8-bit operation)
  - ➢ Overflow from B14 to B15 AND no overflow from B15(for 16-bit operation)

8086 Flag: Control Flag

- **Trap Flag:**
  - ✓ When this flag is set, 8086 enters into single stepping mode. In this mode system will execute one instruction and wait for further direction from the programmer. It is used to debug the program. If it is reset, control continues sequentially
- **Interrupt Enable Flag**:
  - ✓ When it is set, 8086 recognizes interrupt INTR. If it is “reset” it will not recognize interrupt INTR i.e. INTR is maskable interrupt.
  - ✓ If interrupt flag is set (1), the microprocessor will recognize interrupt requests from the peripherals.
  - ✓ If interrupt flag is reset (0), the microprocessor will not recognize any interrupt requests and will ignore them.
- **Directional Flag**:
- This flag is specifically used in string instructions.
- If directional flag is set (1), then access the string data from higher memory location towards lower memory location. If directional flag is reset (0), then access the string data from lower memory location towards higher memory location.
  - ✓ When it is set, content of SI or DI or both automatically decremented (by 1 or 2) after executing the string instruction.
  - ✓ If it is reset, content of SI or DI or both are automatically incremented (by 1 or 2) after executing the string instruction.

Execution Unit(EU)

- **Interrupts:** The Intel 8086 has two hardware interrupt pins:
  - NMI (Non-Maskbale Interrupt)
  - INTR (Interrupt Request) Maskable Interrupt.

Addressing Modes of 8086

- Instruction consists of opcode followed by operand.
- The opcode specifies operation to be performed.
- Operand specifies where data is, it may be in the instruction or may be in the register or in the memory or in the I/O port etc.
- Addressing modes help us to understand the types of operands and the way they are accessed while executing an instruction.
- The different ways by which microprocessor generates operand address are called addressing mode.

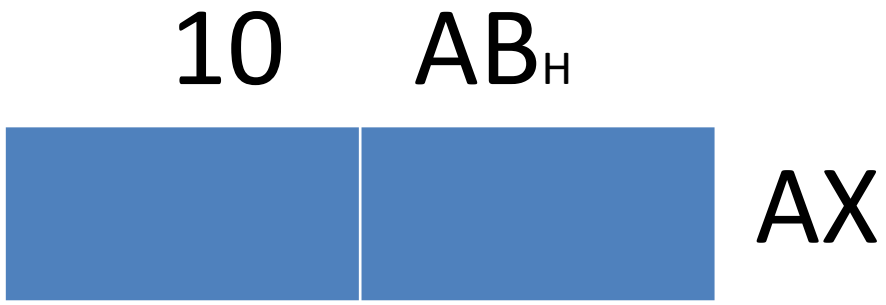
Types of addressing mode in 8086

1. Immediate addressing mode
2. Direct addressing mode
3. Register addressing mode
4. Register Indirect addressing mode
5. Register relative addressing mode
6. Base plus index addressing mode
7. Base relative plus index addressing mode

1: Immediate addressing mode

- In this type of mode, immediate data is part of instruction and appears in the form of successive byte or bytes
- For Example: **MOV AL, 25H** means move 8-bit data 25H into AL register, **MOV CX, 23F6H** means move 16-bit data 23F6 into CX register.
- Note that immediate data are constant data.

MOV AX,10AB$_H$



2: Direct addressing mode

- In this type of addressing mode a 16-bit memory address is directly specified in the instruction as a part of it.
- When a memory location is to be referenced, its offset address must be Specified
- Let the content of DS is 4000H. Now PA= 40000+1250 = 41250, i.e. the content.

Now P.A = 40000
1250
41250

i.e. the content of memory location 41250H is moved into CL register.

CL ← [41250]

MOV CX,[1250]H

CL ← [41250H]

CH ← [41251H]

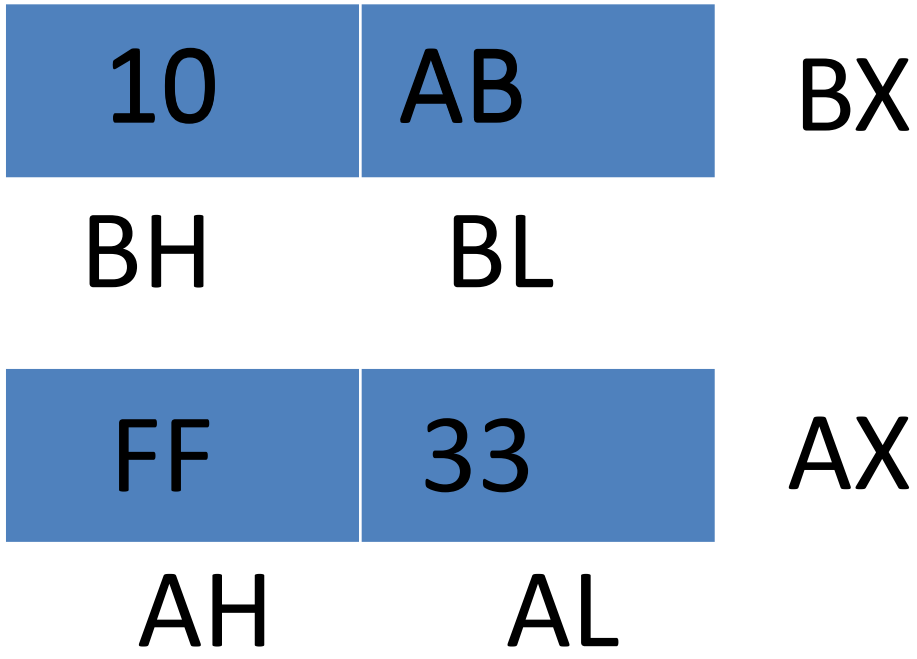
i.e. 16 bit data moved into CX register

3: Register addressing mode

- In this mode, the name of the register which has data and which will have data is specified in the instruction.
- In this type of addressing mode, the data is stored in the register and it can be a 8-bit or 16-bit register.
- All the registers, except IP, may be used in this mode.
- It is important to use registers of same size.

MOV AL,BLH

MOV AX,BXH



4: Register Indirect addressing mode

- The address of the memory location which contains data or operand is determined in a indirect way, using the offset register.
- In this mode, the name of the base register (BX) or the name of the index registers (SI or DI) is specified in the instruction.
- The physical address is generated by adding 0 to the right of DS and adding the value in base register or index register.
- For example: Here, [SI] =1250 and [BX] = 8214

MOV CL,[SI]

PA = 40000
+ 1250
41250

CL ← [41250H]

MOV CL,[BX]

PA = 40000
- 8214
48214

CL ← [48214H]

AX MOV AX,[BX]

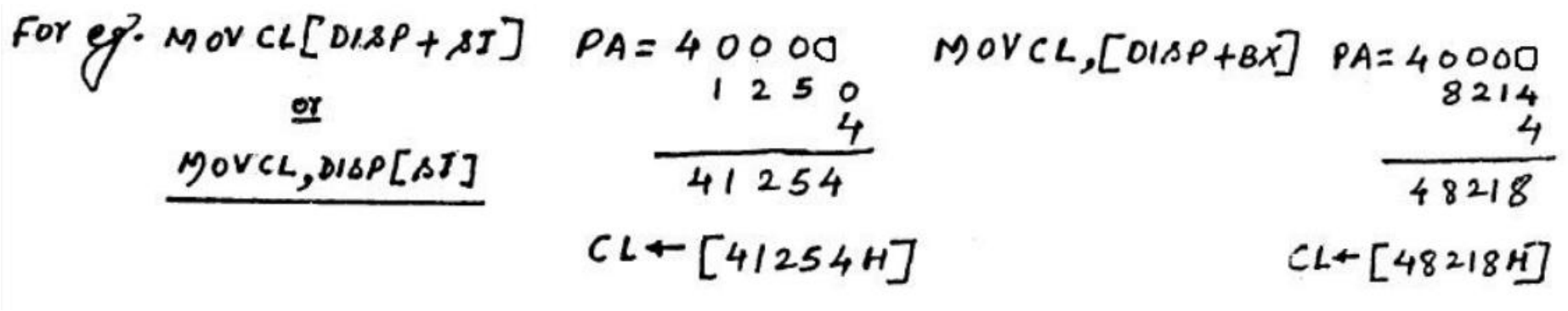
MOV CX,[BX] : Copy's the word contents of the data segment memory location addressed by BX into CX.

5: Register Relative addressing mode

- In this mode, the data is available at an effective address formed by adding an 8-bit or 16-bit displacement with the content of any one of the registers BX, BP, SI and DI in the default (either DS or ES) segment.
- Transfers a byte/word between a register and the memory location addressed by an index or base register plus a displacement.

MOV AX, 50H[BX]

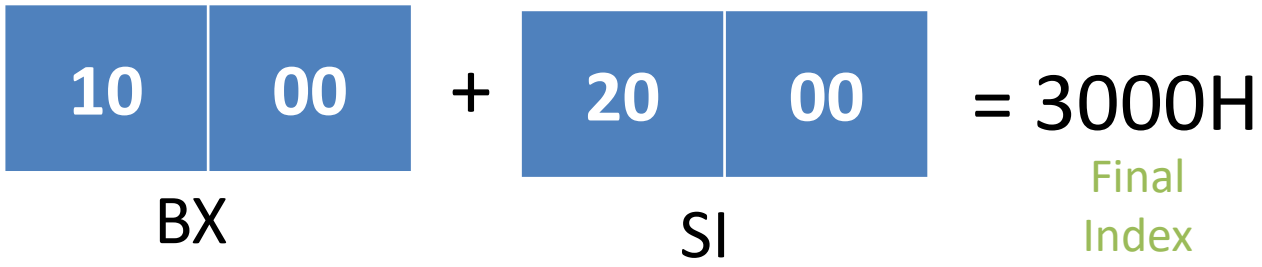
MOV CL, [BX+4]



6: Base and Indexed addressing mode

- In this mode the effective address is formed by adding content of a base register (any one of BX or BP) to the content of an index register (SI or DI). Default segment register DS.

MOV AX, [BX] [SI]



MOV CL, [BX + SI]

PA = 40000
12500
8214
49464

CL ← [49464H]

7: Base Relative Plus Index addressing mode

- In the effective address is formed by adding an 8 or 16-bit displacement with sum of contents of any one of the base registers (BX or BP) and any one of the index registers, in a default segment.
- The base relative plus index addressing mode is similar to the base plus index addressing mode but it adds a displacement to form a memory address.

MOV AH, [BP + SI + 29]

Or

MOV AH, [BP + 29 + SI]

Or

MOV AH, [SI][BP]+29

MOV AX,50H[BX][SI]