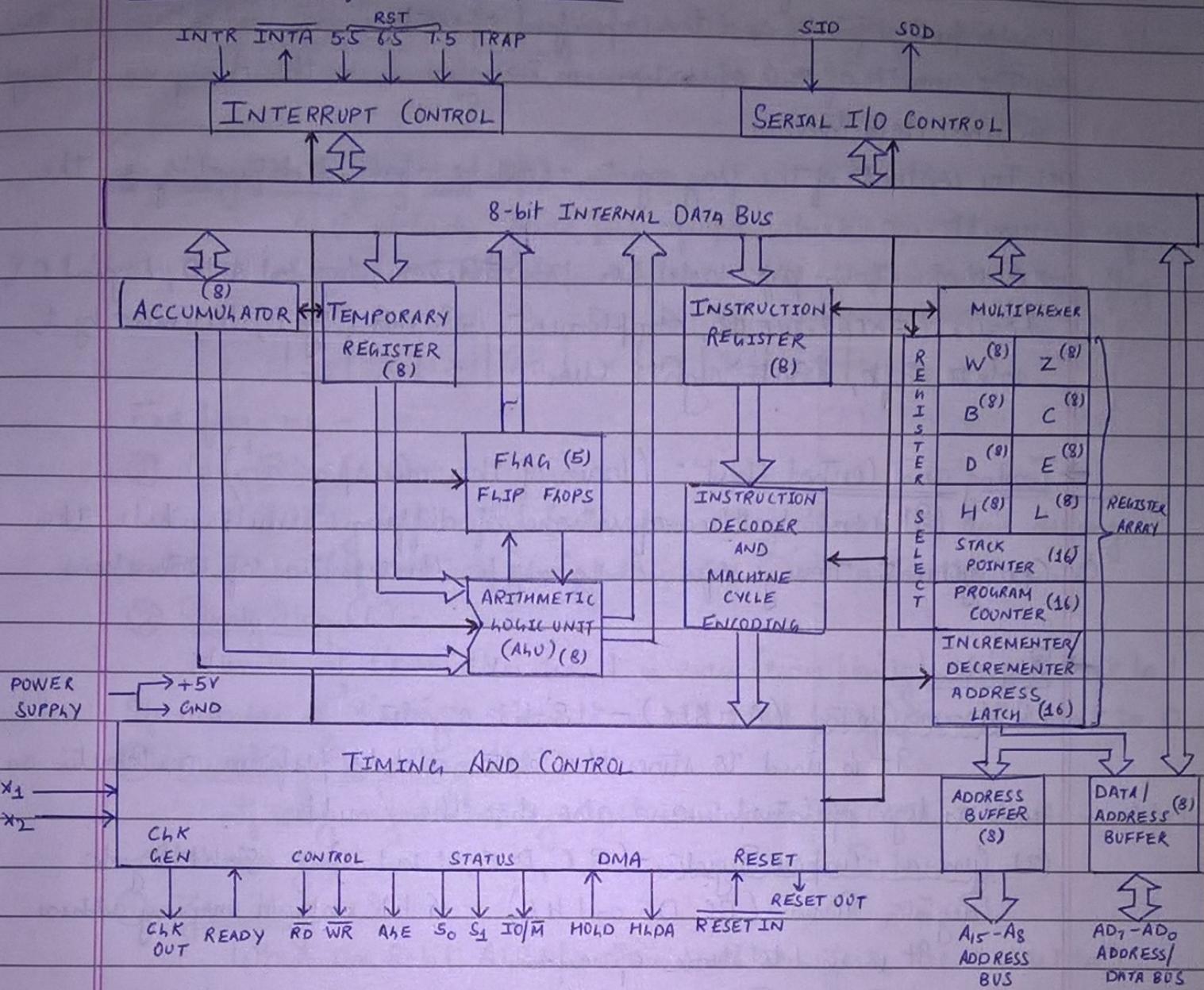


## UNIT - 2

### 8085 Microprocessor

(16 bit address bus, 8 bit data bus)

#### ① Architecture of 8085 microprocessor -



8085 consists of three main sections -

- (1) Arithmetic logic Unit (ALU)
- (2) Timing and Control Unit
- (3) Set of registers

#### → Arithmetic logic Unit (ALU) -

Sequence of operations in ALU are given below -

- (1) One operand is in the A register.
- (2) The other operand may be in the general purpose register or memory locations, which will be transferred to the temporary register.

- (3) Then contents of the accumulator and temporary registers are considered as inputs of ALU and the specified operation is carried out in the ALU.
- (4) The result of ALU operation is transferred in the A register through internal data bus.
- (5) The content of the flag register will be changed depending on the result.

→ ALU operations are addition, subtraction, logical AND, logical OR, logical EXCLUSIVE OR, complement, increment by 1, decrement by 1, rotate left, rotate right, clear.

#### → Timing and Control Unit - (brain of the microprocessor).

It controls the operations of different units while the CPU generates timing sequence signals for the execution of instructions.

#### → Registers -

##### (1) Accumulator (A or ACC) - 8-bit register

It is used to store the data and to perform arithmetic as well as logic operations. and also store the result.

##### (2) General-Purpose Registers - (B, C, D, E, H and L) - 8-bit registers

Register Pairs (BC, DE and HL) - 16 bit data or memory address

It is used to store operands

##### (3) Special-Purpose Registers

###### (i) Program Counter (PC) - 16-bit register

It is used to hold the memory address of the next instruction which will be executed by incrementing the content of the PC.

###### (ii) Stack Pointer (SP) - 16-bit register

It is used to point the memory location called the stack. The stack is a sequence of memory locations in the R/W memory.

###### (iii) Instruction Register (IR) - 8-bit register

It holds the operation code (opcode) of the current instruction of a program during an arithmetic / logical operations

(iv)

#### Temporary Register - 8-bit Register

It holds data during arithmetic and logical operations. This register can be used by the microprogram but is not accessible to the programmer.

(v)

#### Flag / Status Registers (SR) -

AhU includes five flip-flops which are set or reset after an AhU operation. The status of each flip-flop is known as a flag.

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
S	Z	X	AC	X	P	X	CY

Five flags one -

##### ① Carry Flag (CY) -

A carry is produced (when num is larger than 8 bits) and during subtraction (if borrow is generated), CY is set to 1. otherwise 0.

##### ② Parity Flag (P) -

Number of 1's in the result is even (even parity), then P is set to 1. If number of 1's in the result is odd (odd parity), then P is set to 0.

##### ③ Auxiliary Carry Flag (AC) -

If carry is generated by D<sub>3</sub> bit and passed on to D<sub>4</sub>, then AC is set to 1. otherwise zero.

##### ④ Zero Flag (Z) -

When an 8-bit AhU operations result in zero, the Z is set to 1. otherwise zero.

##### ⑤ Sign Flag (S) -

When a number is negative, the S is set to 1. otherwise zero.

#### Pin Diagram of Intel 8085 -

(1) A<sub>15</sub> - A<sub>8</sub> - (Output, 3-state  $\rightarrow$  HOLD, HALT & RESET)  $\Rightarrow$  Address Bus

(2) AD<sub>-</sub> AD<sub>0</sub> - (I/O, 3-state)  $\Rightarrow$  Address / Data Bus

(3) A<sub>5</sub>E (Address latch Enable) =

During the 1<sup>st</sup> clock rate of a machine cycle, it becomes high and enable the address to get latched either into the memory or external latch.

	$X_1$	1	40	Vcc	Control & Status Signals
	$X_2$	2	39	HOLD	$S_0, S_1, IO/M, RD, WR$
	RESET OUT	3	38	HLDA	
Serial I/O communication	{ 800	4	37	CLK (OUT)	
	SID	5	36	RESET IN	
Priority	{ 1 → TRAP	6	35	READY	
	2 → RST 7.5	7	34	IO/M	
	3 → RST 6.5	8	33	$S_1$	
	4 → RST 5.5	9	32	RD	
	5 → INTR	10	31	WR	
	INTA	11	30	ALE	Externally initiated signals
Pinouts	AD <sub>0</sub>	12	29	$S_0$	TRAP, RST 7.5, RST 6.5,
	AD <sub>1</sub>	13	28	A <sub>15</sub>	RST 5.5, INTR, READY
	AD <sub>2</sub>	14	27	A <sub>14</sub>	HOLD
	AD <sub>3</sub>	15	26	A <sub>13</sub>	
	AD <sub>4</sub>	16	25	A <sub>12</sub>	External Signal Acknowledgment
	AD <sub>5</sub>	17	24	A <sub>11</sub>	HLDA, INTA
	AD <sub>6</sub>	18	23	A <sub>10</sub>	
	AD <sub>7</sub>	19	22	A <sub>9</sub>	
	GND AD <sub>8</sub>	20	21	A <sub>8</sub>	

#### (iv) States codes and states of 8085 -

Machine Cycle Status			Status
IO/M	$S_1$	$S_0$	
0	0	1	Memory PWrite
0	1	0	Memory Read
0	1	1	Opcode Fetch
1	0	1	I/O PWrite
1	1	0	I/O Read
1	1	1	Interrupt Acknowledgement

- (5) RD (Read memory or I/O Devices) - 3-state, output  
 (6) WR (Write memory or I/O Devices) - 3-state, output  
 (7) READY (Input) - Indicates that the memory or I/O device is ready.  
 (8) HOLD (Input) - Requesting the use of address and data bus.  
 (9) HOLD ACKNOWLEDGEMENT (Output) - Hold acknowledgement  
 (10) INTR (Input) - Interrupt Request  
 (11) INTA (Output) - Interrupt Acknowledgement  
 (12) RST 7.5, RST 6.5, RST 5.5 - Restart Interrupts  
 (13) TRAP (Input) - Nonmaskable restart interrupt  
 (14) RESET IN (Input) - Reset PC = 0 and also Interrupt enable & HOLD flip-flop  
 (15) RESET OUT (Output) - Indicates CPU is in RESET condition  
 (16) X<sub>1</sub>, X<sub>2</sub> (Input) - Connected to crystal, 6MHz clock frequency.  
 (17) CLK (Output) - Used as system clock & twice X<sub>1</sub>, X<sub>2</sub> input time period  
 (18) SIO (Input) - Serial Input Data Line  
 (19) SOO (Output) - Serial Output Data Line  
 (20) Vcc (+5V supply)  
 (21) GND (Ground reference)

### ③ Addressing modes of 8085 -

Different techniques of specifying data, are called addressing modes. Four types are -

#### (i) Immediate Addressing -

Load the immediate data to the destination, which is given in the instruction. Eg: MVI D, FFH

#### (ii) Register Addressing -

Data is provided through the register. Eg - MOV A, B.

#### (iii) Direct Addressing -

It is used to read data from output devices, store it in accumulator or write data from accumulator to output devices. Eg - LDA 8000H.

#### (iv) Indirect Addressing -

The content of specified registers are used to specify the address of the operand. Eg - ADD M

## ④ Instruction set of 8085 -

### (1) Data Transfer Operations -

Types of Data Transfers	Examples
(1) Between Registers	MOV A, B MOV B, FFH
Move a specific data to a register or a memory location	MVI B, FFH
(2) Between a memory location & a register	MOV B, [3000H]
(3) Between an I/O device & the accumulator	IN 00H
(4) Between Register Pairs	LXI H, D

### (2) Arithmetic Operations -

Performs arithmetic operations such as addition, subtraction and increment/decrement data in registers or memory.

Eg:- ADD, SUB, INX, DCX

### (3) Logical Operations -

Performs logical operations such as AND, OR, Exclusive-OR,

(left/right) Rotate, compare and complement with the contents of accumulators

Eg:- ANA (AND), ORA (OR), XRA (Exclusive-OR), CMP, RLC, RRC, CMA (complement accumulator)

### (4) Branch Control Operations -

Instructions that change the sequence of program execution using conditional jumps & unconditional jumps, subroutine call and return, and restart. Eg:- JMP, CALL, RET

### (5) Stack, I/O and machine control operations -

Stack operations - PUSH, POP, XTHL (Exchange), SPHL (Stack pointer)

I/O operations - IN, OUT

Machine control operations - EI (Enable interrupt system), DI (Disable interrupt system), HLT, NOP (No operation).

(5) Assembly language Programs of 8085 microprocessor

(b) stuck -

It is a group of memory locations in Read / write (R/W) memory of any microcomputer and is used to store the contents of the registers, operand and memory ~~locations~~ addresses. The starting address location of the stack is defined by loading a 16 bit address into the stack pointer, (LXI SP).

9

## Subroutines -

Group of instructions are known as subroutines. When a main program calls a subroutine, the program execution is transferred to the subroutine and after the completion of the subroutine, the program execution returns to the main program. The microprocessor uses the stack to store the return address of the subroutine.

Two annotations instructions → CALC and RETURN

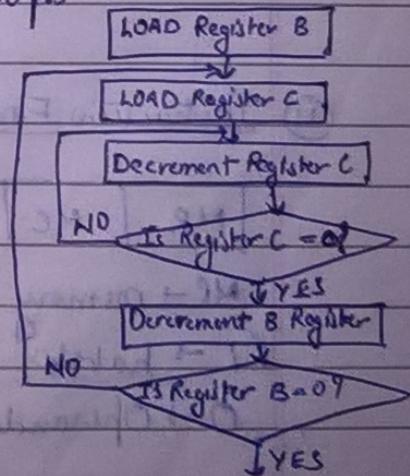
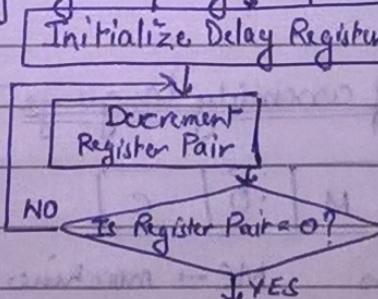
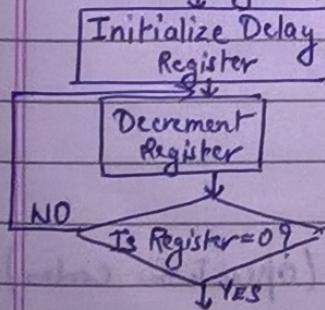
Types of subroutines → multiple CALL subroutines, Nested subroutines, multiple ending subroutines.

8

## Time-Delay loops -

When some time delay is required between two operations, a time-delay loop is used to provide it.

→ Time-Delay loops using registers / register pair / two loops



## ⑨ Modular Programming

## ⑨ Modular Programming -

Complete is divided into sub-problems or small modules. Each independent modules are separately named and are individually manageable program elements. The size of modules are reduced to a humanly comprehensible and manageable level.

### → Characteristics of modules -

- (1) Each module is independent of other modules.
- (2) Each module has one input and one output.
- (3) A module is small in size.
- (4) Programming a single function per module is a goal.

## ⑩ Macro -

Some new instructions can be developed using a sequence of known instructions. These new instructions are always assigned a name and known as MACRO, used in assembly-language programming.

Eg:- LARGE, DELAY, SMALL, MUL, DIV etc.

### General MACRO form -

Name.      MACRO arg → arguments  
Statement-1

Statement-2

Statement-n

ENDM

## ⑪ Instruction Format of assembly language -

MR	MC	L	M	O	C
----	----	---	---	---	---

MR → memory address

MC → machine code (operation codes)

L → labels

M → Mnemonics (states the operation which will be executed)

O → Operands

C → Comments (optional), first character must be semicolon