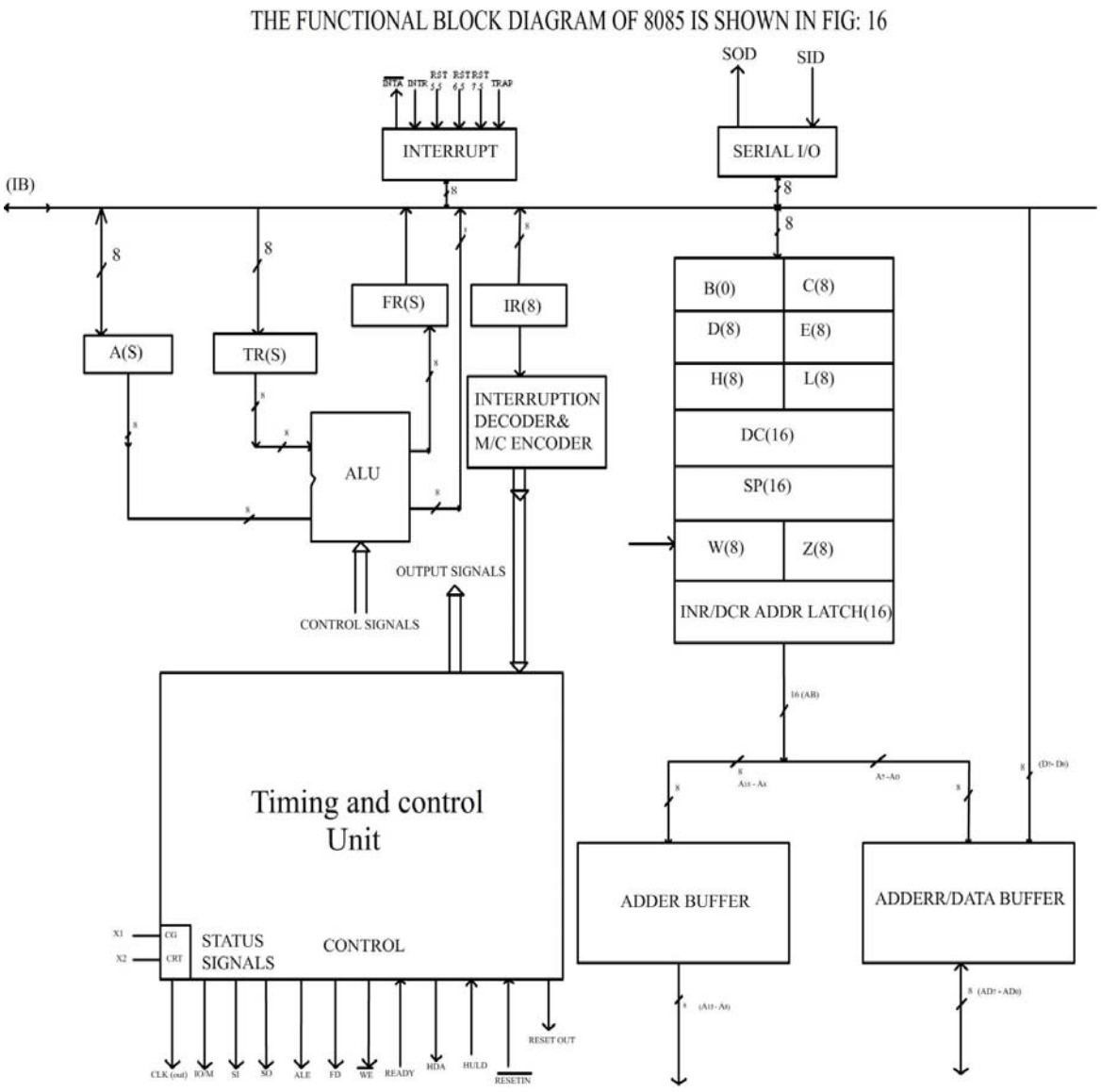
Lecture-12

INTERNAL ARCHITECTURE OF INTEL 8085: The Functional Block Diagram Of 8085 Is Shown In Fig 16.



It consists of five essential blocks.

(1) ARITHMEDIC LOGIC SECTION
(2) REGISTER SECTION
(3) THE INTERRUPT CONTROL SECTION
(4) SERIAL I/O SECTION
(5) THE TIMING AND CONTROL UNIT

The description of each unit follows.

There is an internal bi directional bus of 8 bits. All the internal registers which transfer data to the internal bus are tri state registers.

(1) ARITHMETIC LOGIC SECTION: This section consists of
   (a) Accumulation(A)
   (b) Temporarily Register (TR)
   (c) A flag register (FR)
   (d) An arithmetic logic unit (ALU)

   (1) ACCUMULATION (A): It is a 8 bit tri state register accessible to the user. Its tri state output is connected to the internal bus in addition, it has a two state 8 bit output. The content of the accumulator is always available at this two state output as the accumulator can be manipulated through instruction. Its content can be incremented its content can be decremented. Its content can be transferred to memory location. The content of a memory location can be transfers to the accumulator. All these can be done through instructions. The result of an arithmetic operation carried out by ALU shall also be stored back in the accumulator the result of the operation, hence the name accumulator.
   (2) TEMPORARILY REGISTER (TR): This is an 8 bit register not accessible to the user. It is used by the µp for internal operations. The second operand as and when necessary shall be loaded into this register by the µp before the desired operation takes place in the ALU. The register has 8 bits two state output. This shall be the second operand to the ALU.
   (3) ARITHMETIC LOGIC UNIT (ALU): ALU is a combination logic block which performs the desired operation on the two operands. One from the A and the other from the TR as the dictate of the control signals. Generated by the timing & control unit. In 8085 µp binary addition operation, binary subtraction operations are the only arithmetic section possible. The result of only arithmetic section possible. The results of the operation shall the stored back in a accumulator when the instruction to be executed is

subtraction operation, then the content of the TR shall subtracted from the content of the accumulator and result shall be stored back in the accumulator.

(4) FLAG REGISTER: Flag register is a bit register accessible to the user through instruction each bit in the flag register has a specific functions only of the bit are used as shown in fig 17.

D7

D0

S	Z		AC		P		CY

The three crossed bits are redundant bits and not used. They can be either '0' or '1'. It is immaterial but normally forced to be zero. These five bits are affected as a result of execution of an instruction. All instruction execution do not affected the flags e.g. data transferring operation do not affect these flags the arithmetic operation effect all these flags the meaning & the effect of the fleegs are as follow;

CY CARRY FLAG BIT: this particular bit is SET if there is a carry from the MSB position during an addition operation or if there is a borrow during the subtraction operation, otherwise this flag is RESET.

P-PARITY FLAG BIT: The E flag is SET if the result of an operation contains even nos of 1's otherwise it is RESET.

AC- AOXILIARY CARRY FLAG BIT: This bit is SET if there is a carry from A_3 bit to A_4 bit of the accumulator during the process of executing operation connected with an accumulator otherwise it is RESET. The AE flag is useful for arithmetic & is used in a particular instruction known as DAA (Decimal adjust accumulates).

Z-ZERO FLAG: Zero flag bit is SET if the result of an operation is zero, otherwise it is RESET.

S-SIGN FLAG: This flag is SET if the MSB of the result is a '1' otherwise it is RESET. As an example, let us consider the execution of the instruction ADDB. ADD is the mnemonic for addition B is the second operand. The

first operand is known to exist as the content of the accumulator. The meaning of the instruction is add the content of the B register to the content of A register and store the result back in the accumulator, symbolically, we write the macro RTL complemented.

(A) ⟵ (A) + (B).

Let us suppose the contents of the A& B register are,

(A) = 9BH & (B) = A5H.before the execution of the instruction. It mean content of (A) & (B) are,

$$(A) \rightarrow 1001 \quad 1011$$

$$(B) \rightarrow 1010 \quad 0101$$

$$(A)+(B) \rightarrow (1)0100 \quad 0000 \quad \leftarrow (A)$$

$$\text{cy} \qquad 1111 \quad 1111\,\text{AC}$$

As a result of addition there is a carry from A_3 will be A_4 position in the example and therefore, AC will be SET. Also there is a carry from the MSB cut & therefore CY flag will also be SET soon after the execution of ADD B instruction the accumulator certain (A) = $(0100\ 0000)_2$ 40_{H} and is not zero. Therefore the Z flag is RESET to zero. Also the result certain only one '1' an add number. Therefore the parity pit will also be RESET to '0', therefore, the sign flag shall be RESET to '0'. Thus the flag register contains soon after the execution instruction are 0001 0001 B = 11_{H}.

As a second example, consider another instruction DCR C. DCR is the mnemonic for decrement. C is the operand. This information means decrement the content of the C register by '1' and store it back in the C register, the MACRO RTL implemented is

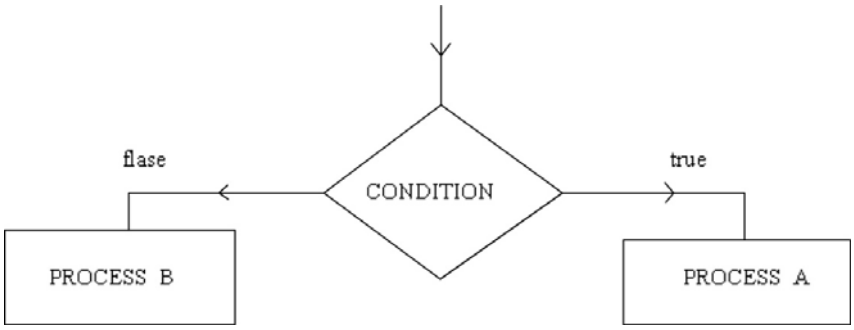
$$(C) \leftarrow (C) \ -1 \ (C) \ \rightarrow C$$

Let us suppose C contains (C) =D2H before the execution of the instruction after the instruction, C shall contains D1H and therefore in not zero. Therefore the flag register will be affected as follows.

	S	Z	X	AC		X	P	X	CY
FR =	1	0	0	0		0	1	0	0

On the other hand, if a contains 01_H just before the execution if the instruction, C shall contain 00_H. Since the result of the operation is '0' the zero flag shall now be SET to '1'. Other flag will be affected in the normal way.

These flag bits are utilized in many instructions for branding operations during the execution of a programme normally one of these bits are tested for TRUE or FALSE condition depending upon the condition the programme branches. This is shown in fig 18.



REGISTER SECTION:

There are 6-8 bit register designed B, C, D, E, H, & L. all are accessible to the user. In an instruction these six 8bit register along with the accumulator A shall be identified by a 8 bit code designated either SSS or DDD. Whenever SSS is used, it corresponds to service register. Whenever, DDD is used, it corresponds to destination register. The code used as follows,

SSS or DDD
000 ----- B
001 ----- C
010 ----- D
011 ----- E
100 ----- H
101 ----- L
111 ----- A

Note in the above code 110 is not used. Whenever 110 is used for SSS or DDD, it means a specific register pair (H,L), together to from 16 bit register known as memory address register (MAR) or M- pointer.

As an example consider the instruction MOV V_1, V_2

This is an ALP statement, MOV is the mnemonic for move, and V_1, v_2 are the operand register, in the statement, V_2 is the source register and V_1 is the destination register. The meaning of the instruction is MOVE the contents of v_2 register into V_1 register5. Symbolically this basic operation can be described by a basic RTL statement (V_1) ---- (V2). This is a single byte instruction. The single byte being the operation code. The arrangement of the op code single byte is shown in fig 19.

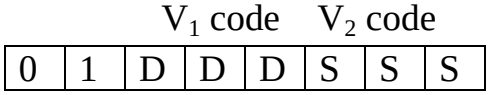
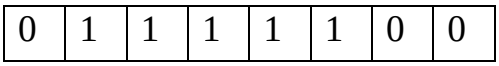


Fig 19 (a) gives the example of MOV A, H code for this statement is,



For move fig 19 (a).

The opcode is read is together 0111 1100 B = $7C_H$. when the instruction$7C_H$ is executed content of 'H' register shall be transferred to 'A' register. Note that content of H register is not destroyed. However, the original content of 'A' register is lost. Let us take another example for the use of code 110.

Consider the instruction MOV D, M

This is an ALP statement, M is the source of the operand and D is the destination register. MOV is the mnemonic for move. The meaning of the instruction is move the content of this memory location whose address is available in (H, L) pair into the D register. This is a single byte instruction. The operation code is 01010110 B = 56 B

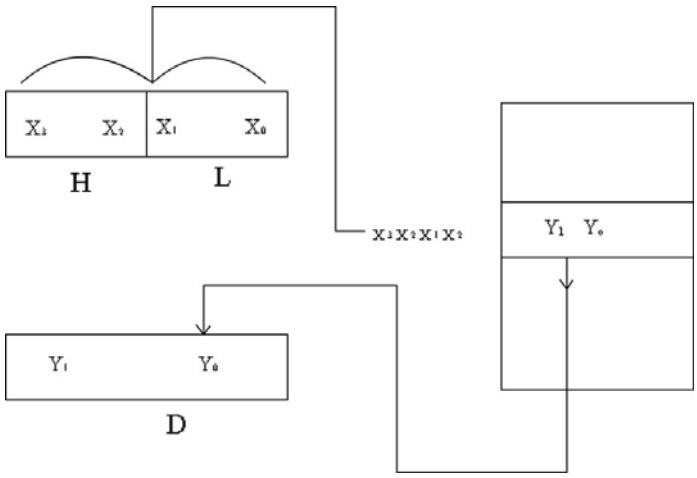


Figure 20

Whenever the instruction 56_H is executed content of the memory location whose address is available in (H,L) pair shall be loaded into the D register. The content of the memory location is not destroyed. However, the content of the memory location Y_1 Y_0 H whose address is X_3 X_2 X_1 X_0 H available in (H,L) pair goes into the D register. The original content of D is lost. This is illustrated in fig 20. The six general purpose register B, C, E, H, L can also be combined together as register pairs are possible. (B,C) pair with lower order 8 bits & B higher order 8 bits; (P,E) pair , E lower order 8 bits, D higher order 8 bits, (H,L) pair with L- lower order 8 bits & H – higher order 8 bits. There is another register name stock pointer, (S,F) which is 16 bits register itself. Whenever an instruction refers to the register pair (B,C), (D,E), (H,L), or (S,P) a 2 bit code RP is used to identify the register pairs (R stands for the register pairs. (R stands for one bit & P stands for other bit)

RP
00 ----- (B, C)
01 ----- (D, E)
10 ----- (H, L)
11 ----- SP --------- (SPH, SPL)