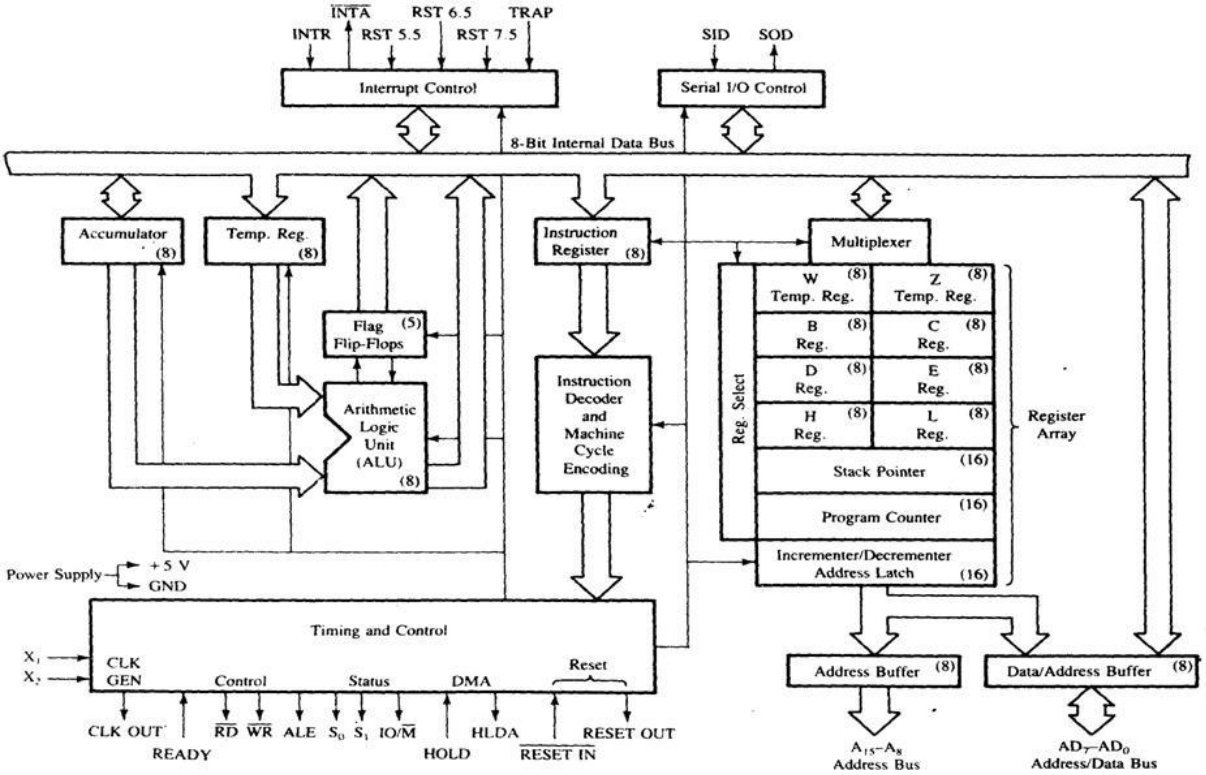
Experiment No 1

Aim: - To study the Architecture of 8085 microprocessor



The figure shows the architecture of 8085. We divide the architecture in different groups as follows:

1. Arithmetic and Logical group
2. Register group
3. Interrupt control group
4. Serial I/O control group
5. Instruction register, decoder and timing and control group.

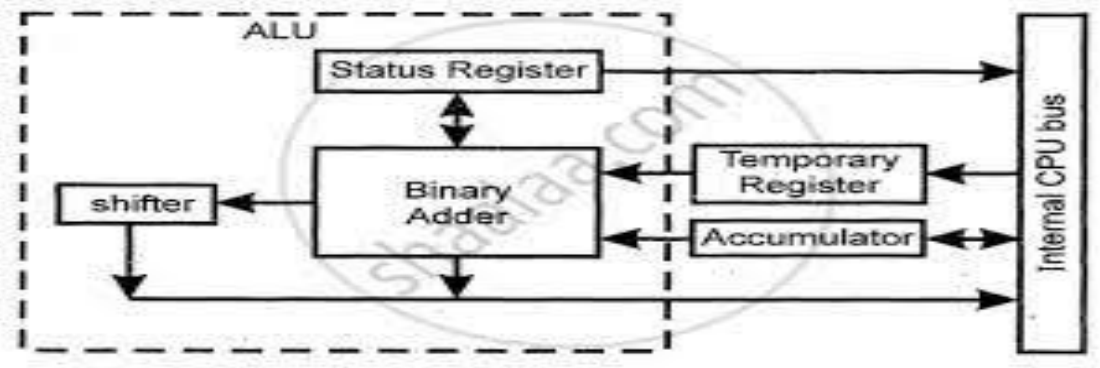
Arithmetic and Logical group

This group consists of ALU, Accumulator or A register, Temporary register, Flag and flip-flops.

i. ALU:

The ALU performs arithmetic and logical operations such as addition, subtraction, ANDing, Oring, EXORing, etc. The inputs to the Alu are provided by accumulator and a temporary register both are 8 bits. The ALU will perform the operation and output result on internal data bus. The ALU is

also of 8 bits so at a time, operation on 8 bit data can only be performed.



ii. Accumulator:

The accumulator is a 8 bit general purpose register connected to internal data bus and to ALU. It is also called as A register. As it is connected as one of the inputs to ALU, it is used in most of the arithmetic and logic instructions. After performing an operation the ALU places its result on internal data bus, from there it is generally stored in accumulator. So accumulator is an integral part in performing different operations along with ALU. Also accumulator is used during I/O operations.

iii. Temporary register:

The other input to ALU is given by temporary register. This register is not available for user, it is only used internally by the microprocessor, so the name given temporary register. To perform arithmetic and logical operations microprocessor assumes one data available in accumulator and takes another data from other register (depends on instruction) into temporary register and then performs operation on the two data operands.

Example: ADD B instruction adds A reg + B reg. The result is stored in register A. in this case one data is available in A reg, and other data from B register is transferred to temporary register and then add operation is performed on them.

This temporary register is also used for other operation also, such as register to register data transfer, etc.

iv. Flag:

The flag is nothing but a group of flip flops used to give status of different operations result. As flag register is connected to ALU, when an operation is performed by ALU the status of result will be stored in flip-flops. It is clear that for all other operations the flag doesn't get affected, it will only give status if a operation is performed in ALU. The position of various flags available in 8085as shown below:

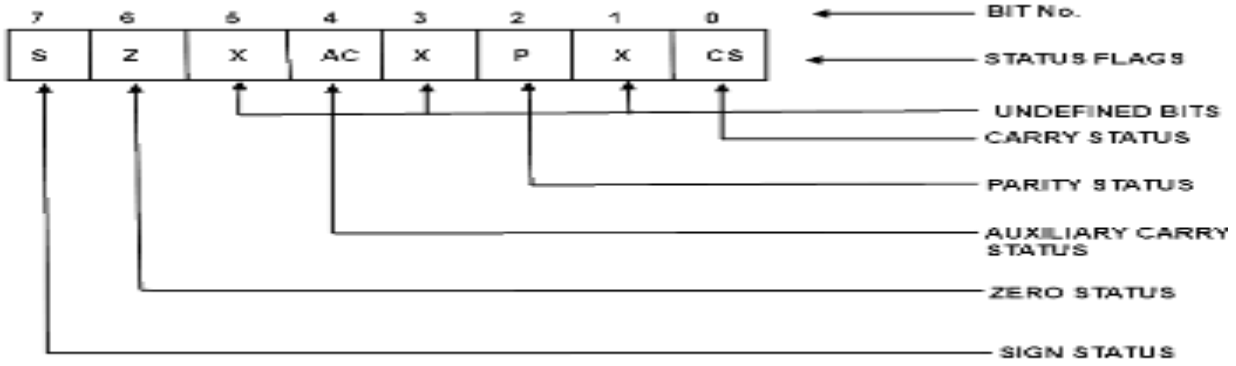


Figure 2: Status Flags of Intel 8085

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S	**Z**	**X**	**AC**	**X**	**P**	**X**	**CY**

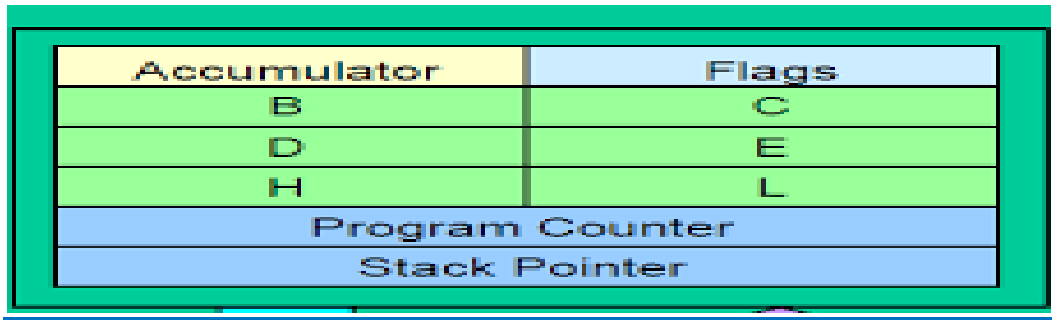
1. **CY- Carry Flag**: - If an operation performed in ALU generates a carry from D_7 to next stage, the CY flag is set. It works as 9th bit for addition and as borrow flag for subtraction. If there is no carry out of MSB bit i.e . D7 of result then the CY flag reset.
2. **AC – Auxiliary Carry Flag :-** If an operation performed by ALU generates a Carry from lower nibble i.e D0-D3 to upper/higher nibble i.e D4- D7 internally by microprocessor, the AC flag is SET. i.e. carry given by D3 to D4. This is not general flag. It is only used internally by microprocessor to perform binary to BCD conversion . It is not available for programmer for changing the sequence as it is for other 4 flags.
3. **Z – Zero Flag:** - If an operation results in zero as a result the zero flag is set. If the result is not zero the zero flag is reset.
4. **S – Sign Flag:** - In sign magnitude format always sign of a number is indicated by D_7 bit. This bit is exact replica of D_7 bit of result. If D_7 =1, the flag is set and if D_7 =0, the flag is reset.
5. **P – Parity Flag:** - This bit is used to indicate the parity of the result. if the result contain even number of 1's this flag is set. If the result contains odd

number of 1's this flag is reset.

Register Group:

This group consists of 3 types of registers.

i. Temporary registers
ii. General purpose registers
iii. Special purpose registers



i. Temporary registers (W and Z)

These are not available for user and used only for internal operations such as to store operand immediate operand or address of memory. These are used internally by the microprocessor for execution of certain instructions.

ii. General purpose registers:

The 8085 contains 6 general purpose registers of 8 bits each, named as B, C, D, E, H and L. these can be used to store 8 bits of data or can be used to form a register pair to store 16 bit data. The register pairs available are BC, DE and HL. These registers are programmable by user. User can store any data in these registers and use it to perform different operations.

iii. Special purpose registers:

The 8085 contains 3 special purpose registers such as program counter, increment/decrement latch and stack pointer.

a. Program counter:

This is a 16 bit register used for execution of program. This register always points to address of memory from where the next instruction is to be fetched and executed. When microprocessor performs one operation of taking instruction i.e. fetching, the PC contents are automatically incremented by one to point to next location. In this way PC keeps the trackfor execution of program. Upon reset PC contents are set to 0000H, so after reset operation microprocessor will start execution of program from 0000H onwards.

b. **Stack Pointer**:

This is a 16 bit register used to define the stack starting address. Stack is a reserved portion of memory where temporary information may be storedor taken back under software control. The stack pointer is used to keep track of data stored.

c. **Increment/Decrement latch:**

This 16 bit register is used in cordination with PC and SP, to increment ordecrement the contents of PC and SP registers.

In coordination with these registers two buffers are used.

- Address buffer:

This is an 8 bit uni-directional buffer are A_8 to A_{15} address lines. These are used to output higher order address on A_8 to A_{15}. When they are not in use or under certain conditions such as reset, hold, halt these are used to tristate A_8 to A_{15} address lines.

- Address/Data buffer:

This is an 8 bit bidirectional buffer used for address and data. These 2 signals are multiplexed on AD_0 to AD_7 lines. In earlier part it is used to output lower order address A_0 to A_7 and in later part it is used to input or output data D_0 to D_7. These addresses are taken from address lines and data is taken or transferred on internal data bus. Under certain conditions such as reset, hold, halt these are used to tristate A_0 to A_7 address/data

lines. The various sources of addresses for the address register includes program counter, stack pointer, temporary registers, BC pair, DE pair andHL pair.

Interrupt Control

This block accepts different interrupt request input such as TRAP, RST 7.5, RST 6.5,RST 5.5 and INTR, and informs control logic to take action in response to each signal. The response for TRAP, RST 7.5, RST 6.5, RST 5.5 is CALL at restart address. But for INTR, it generates a $\overline{INTA}$ signal and expects external device should insert a RST code or CALL instructions.

Hardware interrupts of 8085

Interrupt	Interrupt vector address	Maskable or non-maskable	Edge or level triggered	priority
TRAP	0024H	Non-makable	Level	1
RST 7.5	003CH	Maskable	Rising edge	2
RST 6.5	0034H	Maskable	Level	3
RST 5.5	002CH	Maskable	Level	4
INTR	Decided by hardware	Maskable	Level	5

Serial I/O Control Group

The data transferred on D_0 to D_7 lines is a parallel data, but under certain condition is advantageous to use serial data transfer. 8085 implements this using SID and SOD signal and the data on these lines is accepted or transferred under software control by serial I/O control block.

Instruction Register, Decoder and Control Group

i. Instruction register:

When an instruction is fetched from memory it is loaded in instruction register from there it is provided to decoder for decoding. This register is only activated when instruction code or opcode is available on internal data bus. Itis non programmable

register i.e. not available for programmer use. It accepts only opcode of instruction, operands are not accepted by this instead they are stored in registers.

ii. ***Instruction decoder***:

This accepts a bit pattern from Instruction register, decodes it and gives the different information to control logic. The information includes what operation is to be performed, who is going to perform it, how many operand bytes the instruction contains, etc.

iii. Timing and Control Unit:

This is a control section of 8085. This accepts information from instruction decoder and generates micro steps to perform it so 8085 is called as micro-programmed. In addition to this the block accepts clock inputs and performs sequencing and synchronizing operations required for communication between microprocessor and peripheral devices. To implement this it used different status and control signals.

For internal operations depending on decoder signals the steps or sequence is generally prepared. As an example we take the transfer of data from B register to A register. To implement this, steps taken by 8085 are:

i. Take data from b register on to internal data bus.
ii. Enable temporary register of ALU group to accept data from internal data bus.
iii. Again enable the temporary register to send the data on internal databus.
iv. Enable A register to accept data from internal data bus.

In this way, the steps are implemented for execution of instructions in 8085.

Conclusion: - Thus we have studied Architecture of 8085.