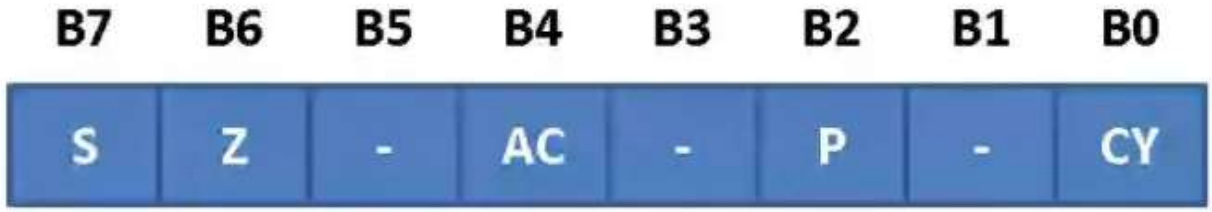
Registers in 8085

- **Specific Purpose Registers**
  - **Accumulator :** The accumulator is an 8-bit register (can store 8-bit data) that is the part of the arithmetic and logical unit (ALU). After performing arithmetical or logical operations, the result is stored in accumulator. Accumulator is also defined as register A.

Registers in 8085

- **Specific Purpose Registers**



  - **Flag registers:**

a. **Sign Flag:** It helps the programmer to know whether the number stored in the accumulator is positive or negative. If the sign flag 1, the number stored in the accumulator is negative else the number is positive.

b. **Zero Flag:**: It occupies the sixth bit of the flag register. It is set, when the operation performed in the ALU results in zero

c. **Auxiliary Carry Flag:** In an arithmetic operation, when a carry flag is generated by the third bit and passed on to the fourth bit, then Auxiliary Carry flag is set. **This is the only flag register in 8085 which is not accessible by user.**

d. **Parity Flag:** This flag tests for number of 1's in the accumulator. If the accumulator holds even number of 1's, then this flag is set otherwise it is reset.

e. **Carry Flag:** If the arithmetic operation results in a carry(if result is more than 8 bit), then Carry Flag is set; otherwise it is reset.

Registers in 8085

- **Specific Purpose Registers**
  - **Memory Registers** There are two 16-bit registers used to hold memory addresses. The size of these registers is 16 bits because the memory addresses are 16 bits.
    - **Program Counter:** This register is used 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. When a byte (machine code) is being fetched, the program counter is incremented by one to point to the next memory location.
    - **Stack Pointer:** It is used as a memory pointer. It points to a memory location in read/write memory, called the stack. It is always incremented/decremented by 2 during push and pop operation.

How does the microprocessor works?

- The microprocessor follows a sequence: Fetch, Decode, and then Execute.
- Initially, the instructions are stored in the memory in a sequential order.
- The microprocessor fetches those instructions from the memory, then decodes it and executes those instructions till STOP instruction is reached.
- Later, it sends the result in binary to the output port.
- Between these processes, the register stores the temporarily data and ALU performs the computing functions.

Logical Instructions

OP-CODE	OPERAND	DESTINATION	EXAMPLE
ANA	R	A = A AND R	ANA B
ANA	M	A = A AND Mc	ANA 2050
ANI	8-bit data	A = A AND 8-bit data	ANI 50
ORA	R	A = A OR R	ORA B
ORA	M	A = A OR Mc	ORA 2050
ORI	8-bit data	A = A OR 8-bit data	ORI 50

In the table, R stands for register, M stands for memory, Mc stands for memory contents, r.p. stands for register pair

Logical Instructions in 8085

- ANA R – Register; ANA M – Indirect
- Size of instruction
  - 1 byte
- Flags affected
  - All flags
- Example
  - ANA B
  - ANA 2000H

Logical Instructions in 8085

- ANI 8-bit immediate data
- The ANI instruction works exactly like the ANA instruction but performs logical AND of 8-bit immediate value with the contents of the accumulator register.
- Size of instruction
  - 2 bytes
- Flags affected
  - All flags
- Example
  - ANI 35H

Logical Instructions in 8085

- ORA R – Register, ORA M – Indirect
- Size of instruction
  - 1 byte
- Flags affected
  - All flags
- Example
  - ORA B
  - ORA 7580H

Logical Instructions

OP-CODE	OPERAND	DESTINATION	EXAMPLE
XRA	R	A = A XOR R	XRA B
XRA	M	A = A XOR Mc	XRA 2050
XRI	8-bit data	A = A XOR 8-bit data	XRI 50
CMA	none	A = 1's compliment of A	CMA
CMP	R	Compares R with A and triggers the flag register	CMP B
CMP	M	Compares Mc with A and triggers the flag register	CMP 2050
CPI	8-bit data	Compares 8-bit data with A and triggers the flag register	CPI 50

In the table, R stands for register, M stands for memory, Mc stands for memory contents, r.p. stands for register pair