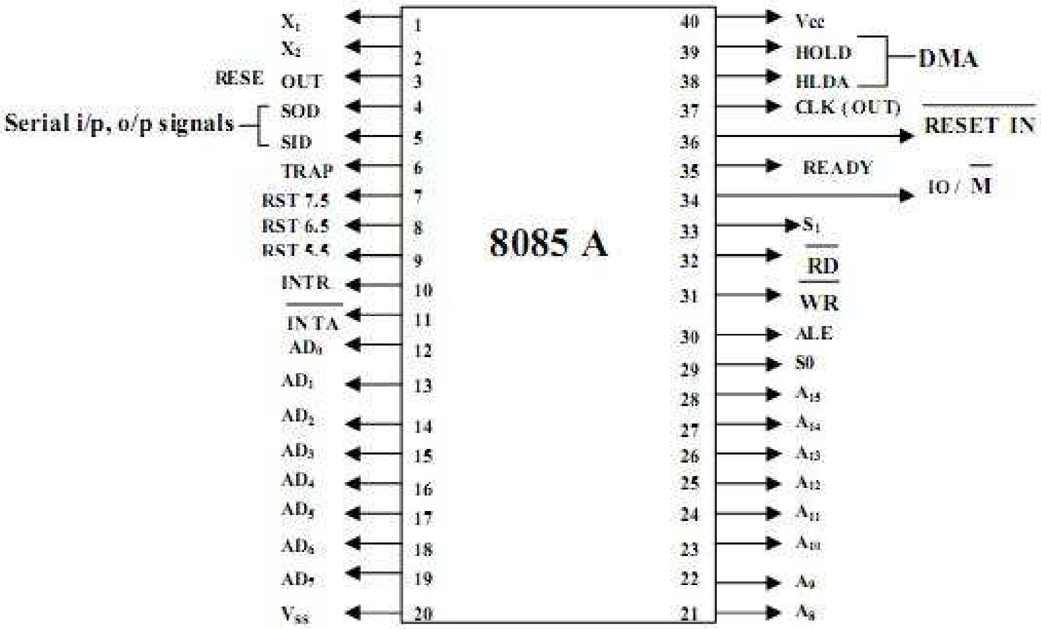
INTEL 8085 MICROPROCESSOR

CPU architecture

1. PIN Diagram



Pin Diagram of 8085

Address Bus:

The pins A8-A15 denote the address bus. They are used for the most significant bit of memory address.

Address/Data Bus:

AD0-AD7 constitutes the Address/Data bus. They are time multiplexed. These pins are used for least significant bits of address bus in the first machine clock cycle and used as data bus for second and third clockcycle.

ALE: Address Latch Enable

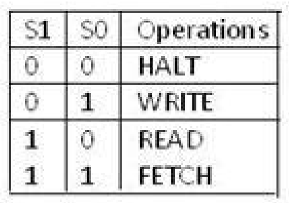
ALE helps in demultiplexing the lower order address and data bus. This signal goes high during the first clock cycle and enables the lower order address bits. The lower order address bus is added to memory or any external latch.

IO/M':

This distinguishes whether the address is for memory or IO. When this pin goes high, the address is for an I/O device. While the pin goes low, the address is assigned for the memory.

S0-S1:

S0 and S1 are status signals which provides different status and functions depending on their status.



S1	S0	Operations
0	0	HALT
0	1	WRITE
1	0	READ
1	1	FETCH

RD':

This is an active low signal. That is, an operation is performed when the signal goes low. This signal is used to control READ operation of the microprocessor. When this pin goes low the microprocessor reads the data from memory or I/O device.

WR':

WR' is also an active low signal which controls the write operations of the microprocessor. When this pin goes low, the data is written to the memory or I/O device.

READY:

READY is used by the microprocessor to check whether a peripheral is ready to accept or transfer data. These peripherals are connected to microprocessor using the READY pin. If READY is high then the periphery is ready for data transfer. If not the microprocessor waits until READY goes high.

HOLD:

It is used by different peripheral devices such as DMA controller for requesting the address bus, data bus needed for DMA operation. If this pin is high then it indicates request for DMA Operation.

HLDA:

HLDA is the acknowledgment signal for HOLD. It indicates whether the HOLD signal is received or not. After the execution of HOLD request, HLDA goes low.

INTR:

INTR is an interrupt request signal. It has the lowest priority among the interrupts. INTR can be enabled or disabled by using software. Whenever INTR goes high the microprocessor completes the current instruction which is being executed and then acknowledges the INTR signal and processes it.

INTA':

Whenever the microprocessor receives interrupt signal. It has to be acknowledged. This acknowledgement is done by INTA'. So whenever the interrupt is received INTA' goes high.

RST 5.5, 6.5, 7.5:

These are nothing but the restart interrupts. They insert an internal restart function automatically.
RST 7.5 ~~ Highest Priority
RST 6.5
RST 5.5 o Lowest Priority
These interrupts have a higher priority than the INTR. All the above mentioned interrupts are maskable interrupts. That is, they can be enabled or disabled using programs.

TRAP:

Among the interrupts of 8085 microprocessor, TRAP is the only non-maskable interrupt. It cannot be enabled or disabled using a program. It has the highest priority among the interrupts.

RESET IN':

This pin resets the program counter to 0 and resets interrupt enable and HLDA flip-flops. The CPU is held in reset condition until this pin is high. However the flags and registers won't get affected except for instruction register.

RESET OUT:

This pin indicates that the CPU has been reset by RESET IN'.

X1 X2:

These are the terminals which are connected to external oscillator to produce the necessary and suitable clock operation.

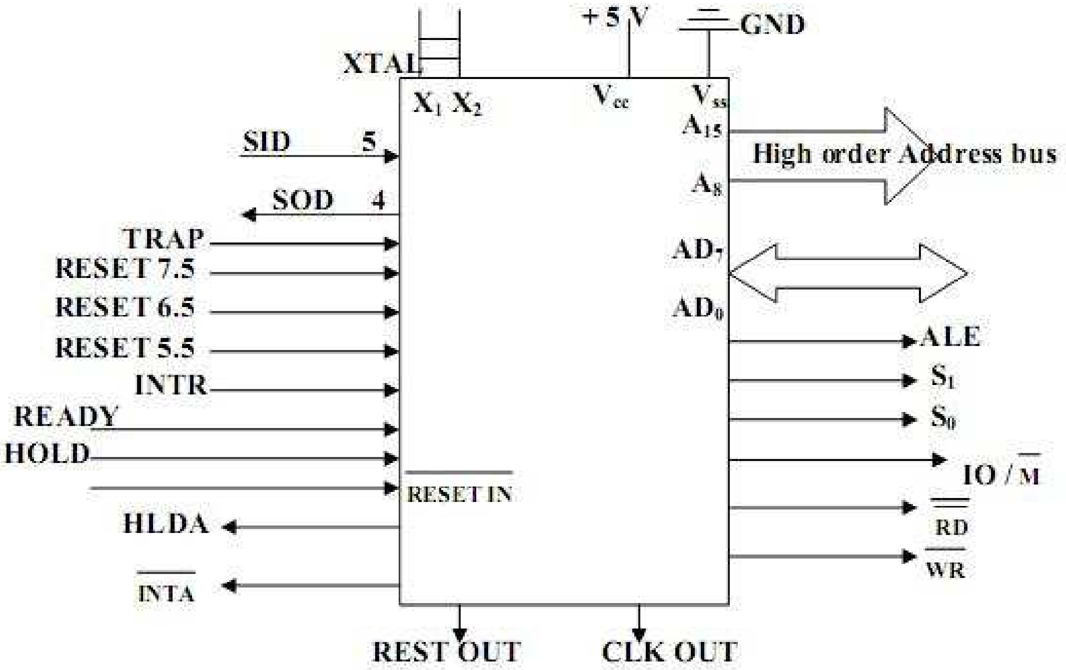
CLK:
Sometimes it is necessary for generating clock outputs from microprocessors so that they can be used for other peripherals or other digital IC's. This is provided by CLK pin. Its frequency is always same as the frequency at which the microprocessor operates.

SID:
This pin provides serial input data. The serial data on this pin is loaded into the seventh bit of the accumulator when RIM instruction is executed.

SOD:
This pin provides the serial output data. The serial data on this pin delivers its output to the seventh bit of the accumulator when SIM instruction is executed.

Vcc and Vss:
Vcc is +5v pin and Vss is ground pin.

2. Signal Diagram



The various signals in a microprocessor can be classified as

Power supply and Frequency signals: Signals which aids in supplying power and generating frequency are associated with this type. Pins like Vcc and ground are classified under this type.

Address signals: Signals associated with the lower order address bus and time multiplexed higher order address bus comes under this type of signals.

Data Signals: Signals associated with data bus comes under this type.

Control and Status Signals: Signals which are associated with timing and control unit such HOLD, RW', WR’ etc. comes under this type of signals.

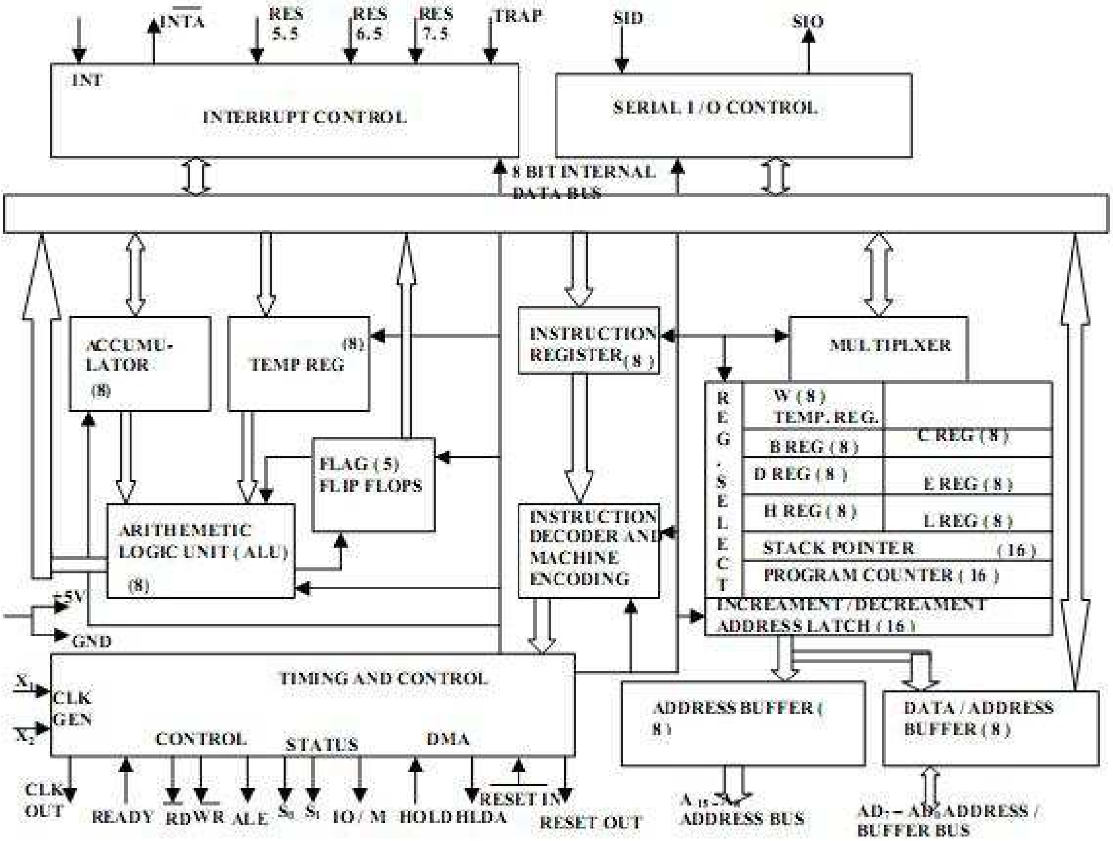
Interrupt Signals: We know that signals like TRAP, RST 5.5 etc. are interrupt signals. Such signals come under this category.

Serial I/O signals: These signals are used for giving serial input and output data. Signals like SID, SOD come under this category.

Acknowledgement Signals: Signals like INTA’, HLDA acts as acknowledgement signal for 8085 microprocessor.

3. FUNCTIONAL Block Diagram

The functional block diagram or architecture of 8085 Microprocessor gives the complete details about a Microprocessor. The following figure shows the Block diagram of a Microprocessor

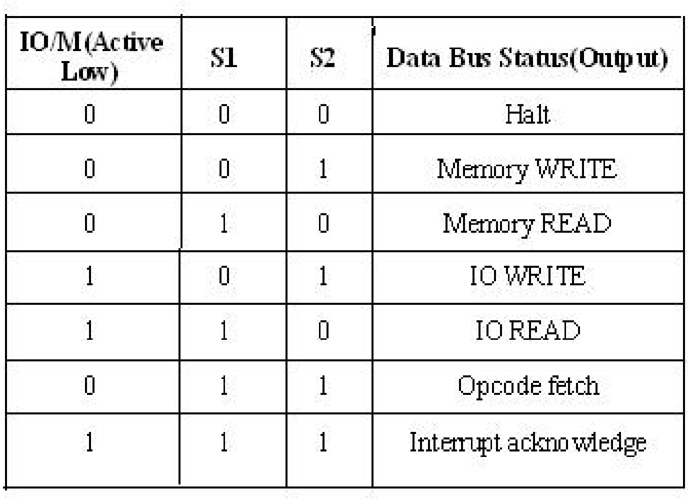


Arithmetic and Logic Unit (ALU):

- It is used to perform the arithmetic operations like addition, subtraction, multiplication, division, increment and decrement and logical operations like AND, OR and EX-OR.
- It receives the data from accumulator and registers.
- According to the result it set or reset the flags.

Timing and Control unit:

- It has three control signals ALE, RD (Active low) and WR (Active low) and three status signals IO/M(Active low), S0 and S1.
- ALE is used for provide control signal to synchronize the components of microprocessor and timing for instruction to perform the operation.
- RD (Active low) and WR (Active low) are used to indicate whether the operation is reading the data from memory or writing the data into memory respectively.
- IO/M(Active low) is used to indicate whether the operation is belongs to the memory or peripherals.



IO/M(Active Low)	S1	S2	Data Bus Status(Output)
0	0	0	Halt
0	0	1	Memory WRITE
0	1	0	Memory READ
1	0	1	IO WRITE
1	1	0	IO READ
0	1	1	Opcode fetch
1	1	1	Interrupt acknowledge

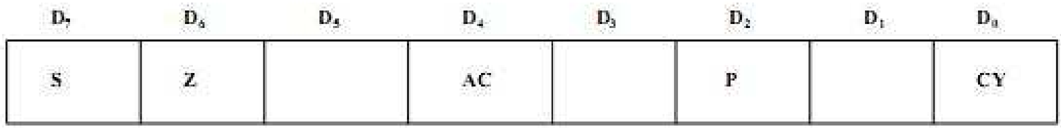
Interrupt Control Unit:

- It receives hardware interrupt signals and sends an acknowledgement for receiving the interrupt signal.

Flags

- The ALU includes five flip-flops that are set or reset according to the result of an operation.
- The microprocessor uses the flags for testing the data conditions.
- They are Zero (Z), Carry (CY), Sign (S), Parity (P), and Auxiliary Carry (AC) flags. The most commonly used flags are Sign, Zero, and Carry.

The bit position for the flags in flag register is,



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

1. Sign Flag (S):
After execution of any arithmetic and logical operation, if D7 of the result is 1, the sign flag is set. Otherwise it is reset.
D7 is reserved for indicating the sign; the remaining is the magnitude of number.
If D7 is 1, the number will be viewed as negative number. If D7 is 0, the number will be viewed as positive number.

2. Zero Flag (z):
If the result of arithmetic and logical operation is zero, then zero flag is set otherwise it is reset.

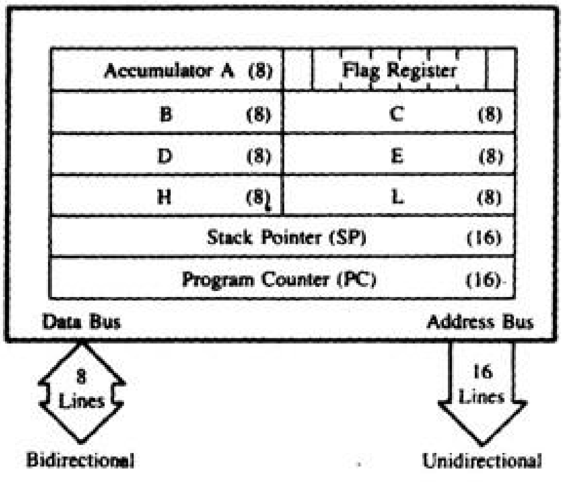
3. Auxiliary Carry Flag (AC):
If D3 generates any carry when doing any arithmetic and logical operation, this flag is set. Otherwise it is reset.

4. Parity Flag (P):
If the result of arithmetic and logical operation contains even number of 1's then this flag will be set and if it is odd number of 1's it will be reset.

5. Carry Flag (CY):
If any arithmetic and logical operation result any carry then carry flag is set otherwise it is reset.

6. Registers

- The 8085 have six general-purpose registers to store 8-bit data during program execution.
- These registers are identified as B, C, D, E, H, and L.
- They can be combined as register pairs-BC, DE, and HL to perform some 16-bit operations.
-



Accumulator (A):

- The accumulator is an 8-bit register that is part of the arithmetic/logic unit (ALU).
- This register is used to store 8-bit data and to perform arithmetic and logical operations.
- The result of an operation is stored in the accumulator.

Flags:

- The ALU includes five flip-flops that are set or reset according to the result of an operation.
- The microprocessor uses the flags for testing the data conditions.

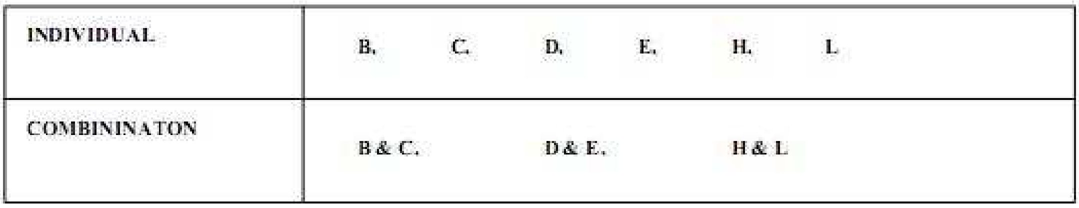
- They are Zero (Z), Carry (CY), Sign (S), Parity (P), and Auxiliary Carry (AC) flags. The most commonly used flags are Sign, Zero, and Carry.

Program Counter (PC):

- This 16-bit register sequencing the execution of instructions.
- It is a memory pointer. Memory locations have 16-bit addresses, and that is why this is a 16-bit register.
- The function of the program counter is to point to the memory address of the next instruction to be executed.
- When an opcode is being fetched, the program counter is incremented by one to point to the next memory location.

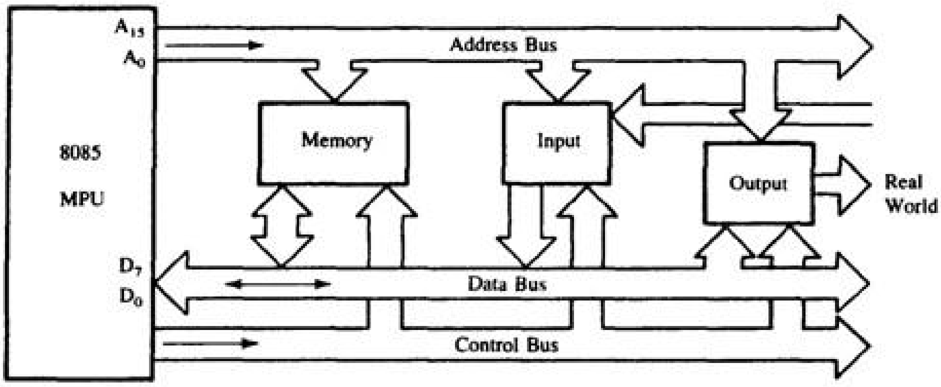
Stack Pointer (SP):

- The stack pointer is also a 16-bit register used as a memory pointer.
- It points to a memory location in R/W memory, called the stack.
- The beginning of the stack is defined by loading a 16-bit address in the stack pointer (register).



INDIVIDUAL	B, C, D, E, H, L
COMBININATON	B & C, D & E, H & L

Bus Structure



Address Bus:

- The address bus is a group of 16 lines generally identified as A0 to A15.
- The address bus is unidirectional: bits flow in one direction-from the CPU to peripheral devices.
- The CPU uses the address bus to perform the first function: identifying a peripheral or a memory location.

Data Bus:

- The data bus is a group of eight lines used for data flow.
- These lines are bi-directional - data flow in both directions between the CPU and memory and peripheral devices.
- The CPU uses the data bus to perform the second function: transferring binary information.
- The eight data lines enable the CPU to manipulate 8-bit data ranging from 00 to FF (28 = 256 numbers).
- The largest number that can appear on the data bus is 11111111.

Control Bus:

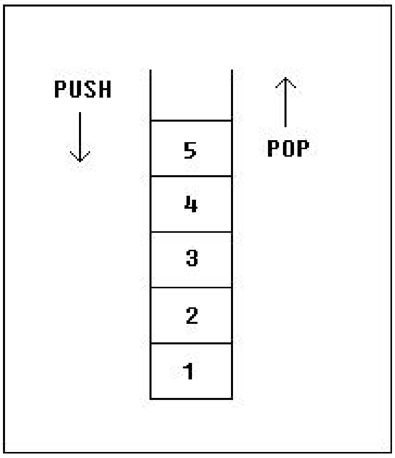
- The control bus carries synchronization signals and providing timing signals.
- The CPU generates specific control signals for every operation it performs. These signals are used to identify a device type with which the CPU wants to communicate.

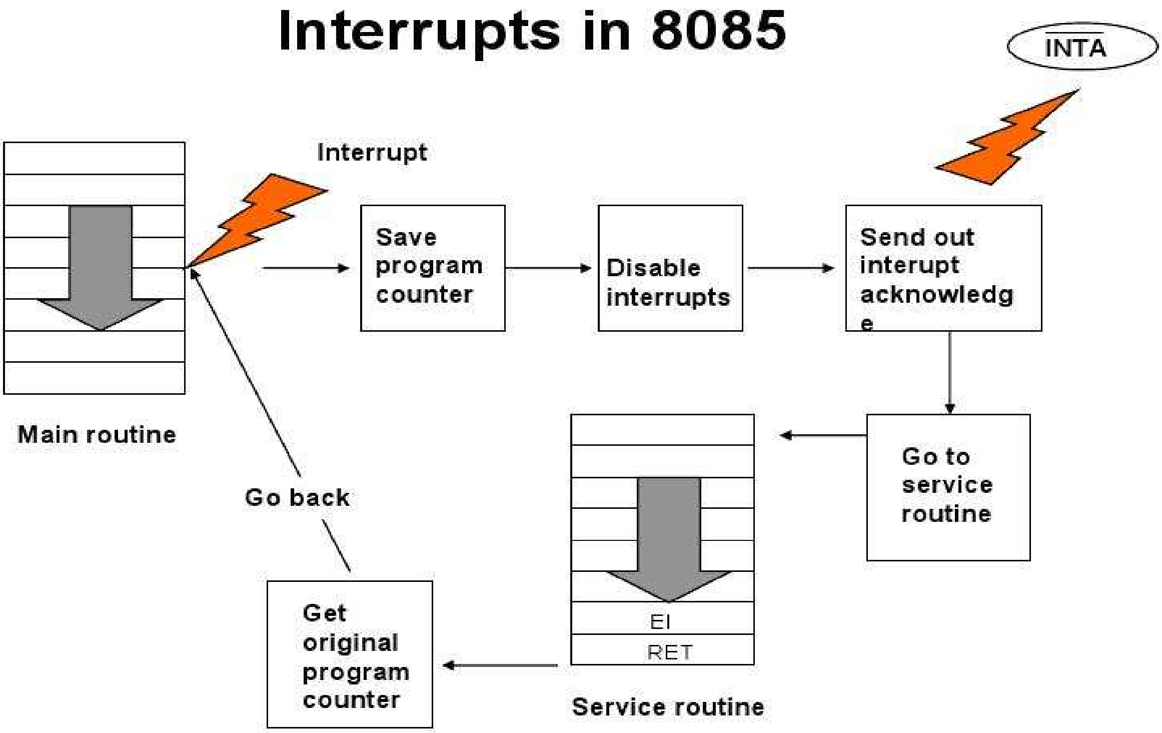
Stack Segment

The Stack is an area of memory for keeping temporary data. Stack is used by the CALL instruction to keep the return address for procedures The return RET instruction gets this value from the stack and returns to that offset. The same thing happens when an INT instruction calls an interrupt. It stores in the Stack the flag register, code segment and offset. The IRET instruction is used to return from interrupt call.

The Stack is a Last in First out (LIFO) memory. Data is placed onto the Stack with a PUSH instruction and removed with a POP instruction. The Stack memory is maintained by two registers: the Stack Pointer (SP) and the Stack Segment (SS) register. When a word of data is PUSHED onto the stack the High order 8-bit Byte is placed in location SP-1 and the Low 8-bit Byte is placed in location SP-2. The SP is then decremented by 2. The SP adds to the (SS x 10H) register, to form the physical stack memory address. The reverse sequence occurs when data is POPPED from the Stack. When a word of data is POPPED from the stack the High order 8-bit Byte is obtained in location SP-1 and the Low 8-bit Byte is obtained in location SP-2. The SP is then incremented by 2.

The stack uses LIFO (Last In First Out) algorithm, this means that if we push these values one by one into the stack: 1, 2, 3, 4, 5 the first value that we will get on pop will be 5, then 4, 3, 2, and only then 1.





When 8085 microprocessor receives an interrupt then it performs the following functions:

1. Finish the execution of current instruction.
2. Save the last content of the program counter (PC).
3. Disable interrupt request line and send interrupt acknowledgement to the interrupting device.
4. Branches to serve the interrupt service routine.
5. Retrieve the contents of program counter (PC).
6. Go back to the saved PC location when finished.

The 8085 microprocessor has 5 interrupts. They are presented below in the order of their priority (from lowest to highest):

- **INTR** is maskable 8080A compatible interrupt. When the interrupt occurs the processor fetches from the bus one instruction, usually one of these instructions:
    - One of the 8 RST instructions (RST0 - RST7).
    - CALL instruction (3 byte instruction).

- **RST5.5** is a maskable interrupt. When this interrupt is received the processor saves the contents of the PC register into stack and branches to 2Ch (hexadecimal) address.

- **RST6.5** is a maskable interrupt. When this interrupt is received the processor saves the contents of the PC register into stack and branches to 34h (hexadecimal) address.

- **RST7.5** is a maskable interrupt. When this interrupt is received the processor saves the contents of the PC register into stack and branches to 3Ch (hexadecimal) address.

- **TRAP** is a non-maskable interrupt. When this interrupt is received the processor saves the contents of the PC register into stack and branches to 24h (hexadecimal) address.

THE 8085 ADDRESSING MODES

The instructions MOV B, A or MVI A, 82H are to copy data from a source into a destination. In these instructions the source can be a register, an input port, or an 8-bit number (00H to FFH). Similarly, a destination can be a register or an output port. The sources and destination are operands. The various formats for specifying operands are called the ADDRESSING MODES. For 8085, they are:

- Immediate addressing.
- Register addressing.
- Direct addressing.
- Indirect addressing.

Immediate addressing

Data is present in the instruction. Load the immediate data to the destination provided. Example: MVI R,data

Register addressing

Data is provided through the registers.
Example: MOV Rd, Rs

Direct addressing

Used to accept data from outside devices to store in the accumulator or send the data stored in the accumulator to the outside device. Accept the data from the port 00H and store them into the accumulator or Send the data from the accumulator to the port 01H.
Example: IN 00H or OUT 01H

Indirect Addressing

This means that the Effective Address is calculated by the processor. And the contents of the address (and the one following) are used to form a second address. The second address is where the data is stored. Note that this requires several memory accesses; two accesses to retrieve the 16 -bit address and a further access (or accesses) to retrieve the data which is to be loaded into the register.

Instruction Format

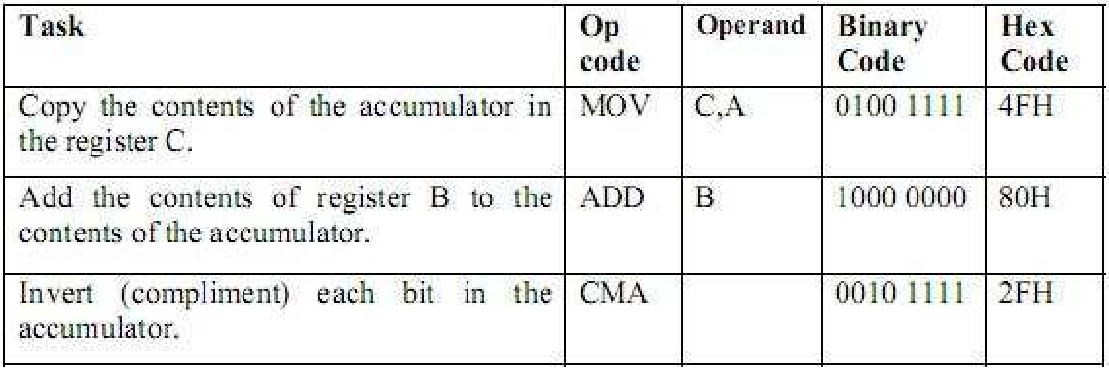
The 8085 instruction set is classified into the following three groups according to word size:

1. One-word or 1-byte instructions
2. Two-word or 2-byte instructions
3. Three-word or 3-byte instructions

One-Byte Instructions

A 1-byte instruction includes the opcodes and operand in the same byte. Operand(s) are internal register and are coded into the instruction.

For example:

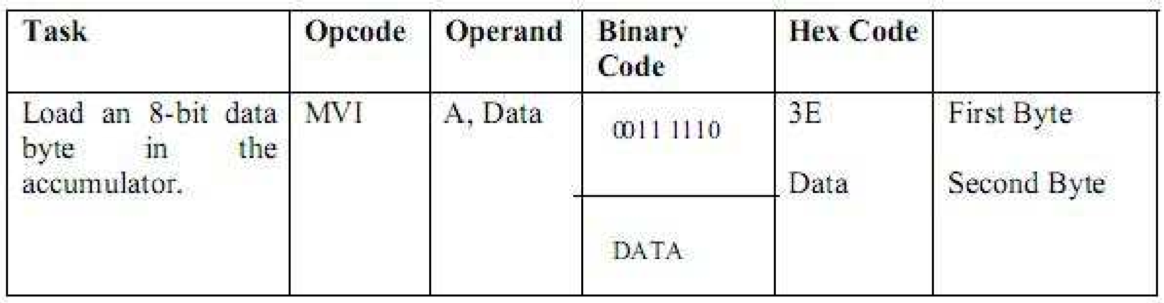


Task	Op code	Operand	Binary Code	Hex Code
Copy the contents of the accumulator in the register C.	MOV	C,A	0100 1111	4FH
Add the contents of register B to the contents of the accumulator.	ADD	B	1000 0000	80H
Invert (compliment) each bit in the accumulator.	CMA		0010 1111	2FH

Two-Byte Instructions

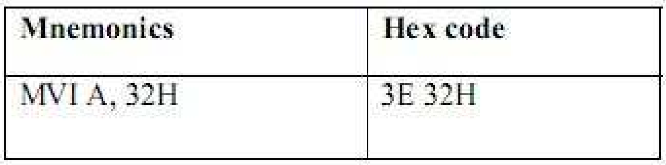
In a two-byte instruction, the first byte specifies the operation code and the second byte specifies the operand.

For example:



Task	Opcode	Operand	Binary Code	Hex Code	
Load an 8-bit data byte in the accumulator.	MVI	A, Data	0011 1110 DATA	3E Data	First Byte Second Byte

Assume that the data byte is 32H. The assembly language instruction is written as

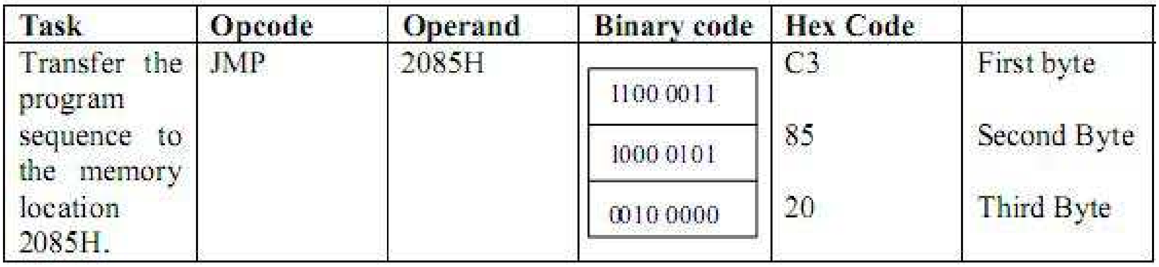


Mnemonics	Hex code
MVI A, 32H	3E 32H

Three-Byte Instructions

In a three-byte instruction, the first byte specifies the opcode, and the following two bytes specify the 16-bit address. Note that the second byte is the low-order address and the third byte is the high-order address. opcode + data byte + data byte

For example:



Task	Opcode	Operand	Binary code	Hex Code	
Transfer the program sequence to the memory location 2085H.	JMP	2085H	1100 0011 1000 0101 0010 0000	C3 85 20	First byte Second Byte Third Byte