Instruction Set and Data Formats

Timing Diagram

Instruction cycle

The CPU fetches one instruction from the memory at a time and executes it. One instruction cycle can consists of 1-6 machine cycle.

Note:

An instruction cycle consists of a fetch cycle and execute cycle.

Machine Cycle

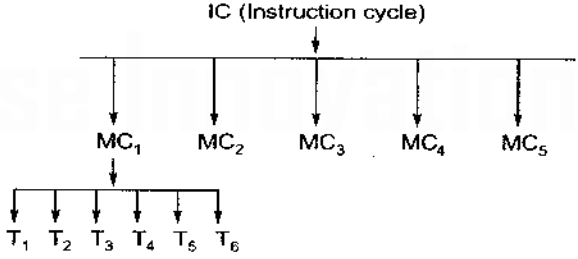
The time required by the microprocessor to complete the operation of accessing memory or I/O device is called as a machine cycle. One machine cycle can consists of 3-6 T-states.

Note:

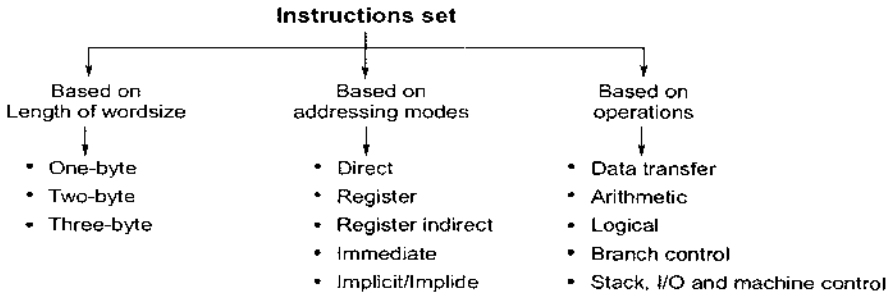
- An instruction cycle consists of several machine cycles.
- Different types of machine cycles are opcode fetch (4T-6T), memory read (3T), memory write (3T), I/O read(3T), I/O write (3T), INTR Acknowledge and BUS Idle(3T).

T-State

Microprocessor perform an operation in a specific time period i.e. specific clock cycles. Each clock cycle is called as T-state.



Classification of Instructions Set of 8085 Microprocessor



An instruction is a binary Pattern designed inside a microprocessor to perform a specific function.

Based on Length of Word Size

1. **One-Byte Instructions:** The one-byte instructions specify the operation to be performed and who is going to perform it. These instructions, required one or single memory location.
2. **Two-Byte Instructions:** The Two-byte instruction uses first byte to specify the operation i.e. opcode and second byte to specify the operand. These instructions required two successive memory locations in the memory.
3. **Three-Byte instructions:** First byte stores opcode, second byte stores lower order 8-bit of 16-bit operand or address and third byte stores higher order 8-bit of 16-bit operand or address.

Based on Addressing Modes

1. **Register addressing mode:** In register addressing mode the source and the destination are general purpose registers.
2. **Immediate addressing mode:** In immediate addressing mode the data (8/16 bit) is specified in the instruction itself.
3. **Direct addressing mode:** In direct addressing mode 16 bit address of the operand is given within the instruction itself.
4. **Indirect addressing mode:** In indirect addressing mode the instructions reference the memory through a register pair, i.e. the memory address where the operand is located is specified by the contents of a register pair.
5. **Implicit addressing mode:** The implicit mode of addressing does not require any operand. The data is specified within the opcode itself.

Based on Operations

Abbreviations used in the description of the instruction set:

R = Register	Rs = Register source
Rd = Register destination	M = Memory
Rp = Register pair	( ) = Contents of
XX = Random information	S = Sign flag
Z = Zero flag	AC = Auxiliary carry flag
P = Parity flag	CY = Carry flag

Data Transfer Instruction

MOV : Move -- Copy from Source to Destination

Opcode	Operand	Bytes	M-Cycles	T-States
MOV	Rd/M, Rs/M	1	1/2	4T/7T

Description:

This instruction copies the contents of the source register into the destination register; the contents of the source register are not altered.

MVI : Move Immediate 8-Bit

Opcode	Operand	Bytes	M-Cycles	T-States
MVI	R/M, 8-bit data	2	2/3	7T/10T

Description

The 8-bit data are stored in the destination register or memory.

LXI : Load Register Pair Immediate

Opcode	Operand	Bytes	M-Cycles	T-States
LXI	Rp, 16-bit data	3	3	10T

Description:

The instruction loads 16-bit data in the register pair designated in the operand.

Comments:

The reverse order in entering the code of 16-bit data. This is the only instruction that can directly load a 16-bit address in the stack pointer register.

LDA : Load Accumulator Direct

Opcode	Operand	Bytes	M-Cycles	T-States
LDA	16-bit address	3	4	13T

Description:

The contents of a memory location, specified by a 16-bit address in the operand, are copied to the accumulator. The contents of the source are not altered.

STA : Store Accumulator Direct

Opcode	Operand	Bytes	M-Cycles	T-States
STA	16-bit address	3	4	13T

Description:

The contents of the accumulator are copied to a memory location specified by the operand.

LHLD : Load H and L Registers Direct

Opcode	Operand	Bytes	M-Cycles	T-States
LHLD	16-bit address	3	5	16T

Description:

The instruction copies the contents of the memory location pointed our by the 16-bit address in register L and copies the contents of the next memory location in register H.

SHLD : Store H and L Registers Direct

Opcode	Operand	Bytes	M-Cycles	T-States
SHLD	16-bit address	3	5	16T

Description:

The contents of register L are stored in the memory location specified by the 16-bit address in the operand, and the contents of H register are stored in the next memory location by incrementing the operand. The contents of registers HL are not altered.

LDAX : Load Accumulator Indirect

Opcode	Operand	Bytes	M-Cycles	T-States
LDAX	B/D Rp	1	2	7T

Description:

The contents of the designated register pair point to a memory location. This instruction copies the contents of that memory location into the accumulator.

Flags:

No flags are affected.

STAX : Store Accumulator Indirect

Opcode	Operand	Bytes	M-Cycles	T-States
STAX	B/D Rp	1	2	7T

Description:

The contents of the accumulator are copied into the memory location specified by the contents of the operand (register pair).

Flags:

No flags are affected.

XCHG : Exchange H and L with D and E

Opcode	Operand	Bytes	M-Cycles	T-States
XCHG	None	1	1	4T

Description:

The contents of register H are exchanged with the contents of register D, and the contents of register L are exchanged with the contents of register E.

Arithmetic Instructions

ADD : Add Register to Accumulator

Opcode	Operand	Bytes	M-Cycles	T-States
ADD	R/M	1	1/2	4T/7T

Description:

The contents of the operand (register or memory) are added to the contents of the accumulator and the result is stored in the accumulator.

ADC : Add Register to Accumulator with Carry

Opcode	Operand	Bytes	M-Cycles	T-States
ADC	R/M	1	1/2	4T/7T

Description:

The contents of the operand (register or memory) and the Carry flag are added to the contents of the accumulator and the result is placed in the accumulator.

ACI : Add Immediate to Accumulator with Carry

Opcode	Operand	Bytes	M-Cycles	T-States
ACI	8-bit data	2	2	7

Description:

The 8-bit data (operand) and the Carry flag are added to the contents of the accumulator, and the result is stored in the accumulator.

ADI : Add Immediate to Accumulator

Opcode	Operand	Bytes	M-Cycles	T-States
ADI	8-bit data	2	2	7T

Description:

The 8-bit data (operand) are added to the contents of the accumulator, and the result is placed in the accumulator.

Flags:

All flags are modified to reflect the result of the addition.

SBB : Subtract Source and Borrow from Accumulator

Opcode	Operand	Bytes	M-Cycles	T-States
SBB	R/M	1	1/2	4T/7T

Description:

The contents of the operand (register or memory) and the Borrow flag are subtracted from the contents of the accumulator and the results are placed in the accumulator.

SUB : Subtract Register or Memory from Accumulator

Opcode	Operand	Bytes	M-Cycles	T-States
SUB	R/M	1	1/2	4T/7T

Description:

The contents of the register or the memory location specified by the operand are subtracted from the contents of the accumulator, and the results are placed in the accumulator.

SBI : Subtract Immediate with Borrow

Opcode	Operand	Bytes	M-Cycles	T-States
SBI	8-bit data	2	2	7T

Description:

The 8-bit data (operand) and the borrow are subtracted from the contents of the accumulator, and the results are placed in the accumulator.

SUI : Subtract Immediate from Accumulator

Opcode	Operand	Bytes	M-Cycles	T-States
SUI	8-bit data	2	2	7T

Description:

The 8-bit data (the operand) are subtracted from the contents of the accumulator, and the results are placed in the accumulator.

INR : Increment Contents of Register/Memory by 1

Opcode	Operand	Bytes	M-Cycles	T-States
INR	R/M	1	1/3	4T/10T

Description:

The contents of the designated register/memory are incremented by 1 and the results are stored in the same place.

DCR : Decrement Source by 1

Opcode	Operand	Bytes	M-Cycles	T-States
DCR	R/M	1	1/3	4T/10T

Description:

The contents of the designated register/memory is decremented by 1 and the results are stored in the same place.

Note:

In INR and DCR operation except carry all flags are affected.

INX : Increment Register Pair by 1

Opcode	Operand	Bytes	M-Cycles	T-States
INX	Rp	1	1	6T

Description:

The contents of the specified register pair are incremented by 1. The instruction views the contents of the two registers as a 16-bit number.

DCX : Decrement Register Pair by 1

Opcode	Operand	Bytes	M-Cycles	T-States
DCX	Rp	1	1	6T

Description:

The contents of the specified register pair are decremented by 1. This instruction views the contents of the two registers as a 16-bit number.

Flags:

No flags are affected.

DAA : Decimal Adjust Accumulator

Opcode	Operand	Bytes	M-Cycles	T-States
DAA	None	1	1	4T

Description:

1. If the value of the low-order four bits ($D_3 - D_0$) in the accumulator is greater than 9 or if AC flag is set, the instruction adds 6(06) to the low-order four bits.
2. If the value of the high-order four bits ($D_7 - D_4$) in the accumulator is greater than 9 or if the Carry flag is set, the instruction adds 6(60) to the high-order four bits.

Flags:

S, Z, AC, P, CY flags are altered to reflect the results of the operation.

DAD : Add Register Pair to H and L Registers

Opcode	Operand	Bytes	M-Cycles	T-States
DAD	Rp	1	3	10T

Description:

The 16-bit contents of the specified register pair are added to the contents of the HL register and the sum is saved in the HL register. The contents of the source register pair are not altered.

Flags:

If the result is larger than 16-bits the CY flag is set. No other flags are affected.

Note:

After the execution of the instruction, the contents of the stack pointer register are not altered.

Logical Instructions

ANA : Logical AND with Accumulator

Opcode	Operand	Bytes	M-Cycles	T-States
ANA	R/M	1	1/2	4T/7T

Description:

The contents of the accumulator are logically ANDed with the contents of the operand (register or memory), and the result is placed in the accumulator.

ANI : AND Immediate with Accumulator

Opcode	Operand	Bytes	M-Cycles	T-States
ANI	8-bit data	2	2	7T

Description:

The contents of the accumulator are logically ANDed with the 8-bit data (operand) and the results are placed in the accumulator.

ORA : Logically OR with Accumulator

Opcode	Operand	Bytes	M-Cycles	T-States
ORA	R/M	1	1/2	4T/7T

Description:

The contents of the accumulator are logically ORed with the contents of the operand (register of memory).

ORI : Logically OR Immediate

Opcode	Operand	Bytes	M-Cycles	T-States
ORI	8-bit data	2	2	7T

Description:

The contents of the accumulator are logically ORed with the 8-bit data in the operand and the results are placed in the accumulator.

XRA : Exclusive OR with Accumulator

Opcode	Operand	Bytes	M-Cycles	T-States
XRA	R/M	1	1/2	4T/7T

Description:

The contents of the operand (register or memory) are Exclusive ORed with the contents of the accumulator.

XRI : Exclusive OR Immediate with Accumulator

Opcode	Operand	Bytes	M-Cycles	T-States
XRI	8-bit data	2	2	7T

Description:

The 8-bit data (operand) are Exclusive ORed with the contents of the accumulator, and the results are placed in the accumulator.

Flags:

Z, S, P are altered to reflect the results of the operation. CY and AC are reset.

CMA : Complement Accumulator

Opcode	Operand	Bytes	M-Cycles	T-States
CMA	None	1	1	4T

Description:

The contents of the accumulator are complemented.

Flags:

No flags are affected.

CMP : Compare with Accumulator

Opcode	Operand	Bytes	M-Cycles	T-States
CMP	R/M	1	1/2	4T/7T

Description:

- The contents of the operand (register or memory) are compared with the contents of the accumulator. Both contents are preserved and the comparison is shown by setting the flags as follows:
  - ⇒ If (A) < (R/M) : Carry flag is set and Zero flag is reset.
  - ⇒ If (A) = (R/M) : Zero flag is set and Carry flag is reset.
  - ⇒ If (A) > (R/M) : Carry and Zero flags are reset.
- The comparison of two bytes is performed by subtracting the contents of the operand from the contents of the accumulator; however, neither contents are modified.

Flags:

- S, P, AC are also modified in addition to Z and CY to reflect the results of the operation.

CPI : Compare Immediate with Accumulator

Opcode	Operand	Bytes	M-Cycles	T-States
CPI	8-bit	2	2	7T

Description:

- The second byte (8-bit data) is compared with the contents of the accumulator.
  - ⇒ If (A) < Data : Carry flag is set and Zero flag is reset.
  - ⇒ If (A) = Data : Zero flag is set and Carry flag is reset.
  - ⇒ If (A) > Data : Carry and Zero flags are reset.
- The comparison of two bytes is performed by subtracting the data byte from the contents of the accumulator; however, neither contents are modified.

Flags:

S, P, AC are also modified in addition to Z and CY to reflect the result of the operation.

CMC : Complement Carry

Opcode	Operand	Bytes	M-Cycles	T-States
CMC	None	1	1	4T

Description:

The Carry flag is complemented.

Flags:

The Carry flag is modified, no other flags are affected.

STC : Set Carry

Opcode	Operand	Bytes	M-Cycles	T-States
STC	None	1	1	4T

Description:

The Carry flag is set.

Flags:

No other flags are affected.

RLC : Rotate Accumulator Left

Opcode	Operand	Bytes	M-Cycles	T-States
RLC	None	1	1	4T

Description:

Each binary bit of the accumulator is rotated left by one position. Bit D_7 is placed in the position of D_0 as well as in the Carry flag.

Flags:

CY is modified according to bit D_7. S, Z, P, AC are not affected.

RAL : Rotate Accumulator Left through Carry

Opcode	Operand	Bytes	M-Cycles	T-States
RAL	None	1	1	4T

Description:

Each binary bit of the accumulator is rotated left by one position through the Carry flag. Bit D_7 is placed in the bit in the Carry flag and the Carry flag is placed in the least significant position D_0.

Flags:

CY is modified according to bit D_7. S, Z, AC, P are not affected.

RRC : Rotate Accumulator Right

Opcode	Operand	Bytes	M-Cycles	T-States
RRC	None	1	1	4T

Description:

Each binary bit of the accumulator is rotated right by one position. Bit D_0 is placed in the position of D_7 as well as in the Carry flag.

Flags:

CY is modified according to bit D_0. S, Z, P, AC are not affected.

RAR : Rotate Accumulator Right through Carry

Opcode	Operand	Bytes	M-Cycles	T-States
RAR	None	1	1	4T

Description:

Each binary bit of the accumulator is rotated right by one position through the Carry flag. Bit D_0 is placed in the Carry flag and the bit in the Carry flag is placed in the most significant position D_7.

Flags:

CY is modified according to bit D_0. S, Z, P, AC are not affected.

Branch Control Instructions

JMP : Jump Unconditionally

Opcode	Operand	Bytes	M-Cycles	T-States
JMP	16-bit	3	3	10T

Description:

This instruction is equivalent to a 1-byte unconditional Jump instruction.The program sequence is transferred to the memory location specified by the 16-bit address.

PCHL : Load Program Counter with HL Contents

Opcode	Operand	Bytes	M-Cycles	T-States
PCHL	None	1	1	6T

Description:

The contents of registers H and L are copied into the program counter.

Flag:

No flags are affected.

CALL : Unconditional Subroutine Call

Opcode	Operand	Bytes	M-Cycles	T-States
CALL	16-bit address	3	5	18T

Description:

The program sequence is transferred to the address specified by the operand. Before the transfer, the address of the next instruction to CALL (the contents of the program counter) is pushed on the stack.

Flags:

No flags are affected.

RET : Return from Subroutine Unconditionally

Opcode	Operand	Bytes	M-Cycles	T-States
RET	None	1	3	10T

Description:

The program sequence is transferred from the subroutine to the calling program. The instruction is equivalent to POP program counter.

Flags:

No flags are affected.

XTHL : Exchange H and L with Top of Stack

Opcode	Operand	Bytes	M-Cycles	T-States
XTHL	None	1	5	16T

Description:

The contents of the L register are exchanged with the stack location pointed out by the contents of the stack pointer register. The contents of the H register are exchanged with the next stack location (SP + 1).

Flags:

No flags are affected.

Stack, I/O and Machine Control Instructions

NOP : No Operation

Opcode	Operand	Bytes	M-Cycles	T-States
NOP	None	1	1	4T

Description:

No operation is performed.

omments:

The instruction is used to fill in time delays or to delete and insert instructions while troubleshooting.

LT : Halt and Enter Wait State

Opcode	Operand	Bytes	M-Cycles	T-States
HLT	None	1	2 or more	4T or more

escription:

The microprocessor finishes executing the current instruction and halts any further execution.

ote:

NOP and HLT bot are same as no operation. However, when NOP is executed the PC is increased by 1 time where as for HLT PC remains the same.

USH : Push Register Pair onto Stack

Opcode	Operand	Bytes	M-Cycles	T-States
PUSH	Rp	1	3	12T

escription:

The contents of the register pair designated in the operand are copied into the stack in the following sequence. The stack pointer register is decremented and the contents of the high-order register (B, D, H, A) are copied into that location. The stack pointer register is decremented again and the contents of the low-order register (C, E, L, flags) are copied to that location.

ags:

No flags are modified.

OP : Pop off Stack to Register Pair

Opcode	Operand	Bytes	M-Cycles	T-States
POP	Rp	1	3	10T

escription:

The contents of the memory location pointed out by the stack pointer register are copies to the low-order register (such as C, E, L and flags) of the operand. The stack pointer is incremented by 1 and the contents of that memory location are copied to the high-order register (B, D, H, A) of the operand. The stack pointer register is again incremented by 1.

Flags:

No flags are modified.

Note:

The contents of the source, stack locations, are not altered after the POP instruction.

RSTn : Restart

Opcode	Operand	Bytes	M-Cycles	T-States
RSTn	None	1	3	12T

Description:

This instruction is equivalent to 1-byte CALL instruction.

Flags:

No flags are affected.

SPHL : Copy H and L Registers to the Stack Pointer

Opcode	Operand	Bytes	M-Cycles	T-States
SPHL	None	1	1	6T

Description:

The instruction loads the contents of the H and L registers into the stack pointer register.

Flags:

No flags are affected.

Note:

- This instruction performs the same function as MOV A, M except this instruction uses the contents of BC or DE as memory pointers.
- This instruction is used in conjunction with CALL or conditional call instructions.

IN : Input Data to Accumulator from a Port with 8-bit Address

Opcode	Operand	Bytes	M-Cycles	T-States
IN	8-bit port address	2	3	10T

Description:

The contents of the input port designated in the operand are read and loaded into the accumulator.

Flags:

No flags are affected.

)UT : Output Data from Accumulator to a Port with 8-Bit Address

Opcode	Operand	Bytes	M-Cycles	T-States
OUT	8-bit port address	2	3	10T

)escription:

The contents of the accumulator are copied into the output port specified by the operand.

lags:

No flags are affected.

I : Enable Interrupts

Opcode	Operand	Bytes	M-Cycles	T-States
EI	None	1	1	4T

escription:

The Interrupt Enable flip-flop is set and all interrupts are enabled.

ags:

No flags are affected.

omments:

After a system reset or the acknowledgment of an interrupt, the Interrupt Enable flip-flop is reset, thus disabling the interrupts. This instruction is necessary to enable the interrupts (except TRAP).

I : Disable Interrupts

Opcode	Operand	Bytes	M-Cycles	T-States
DI	None	1	1	4T

escription:

The Interrupt Enable flip-flop is reset and all the interrupts except the TRAP (8085) are disabled.

ags:

No flags are affected.

)mments:

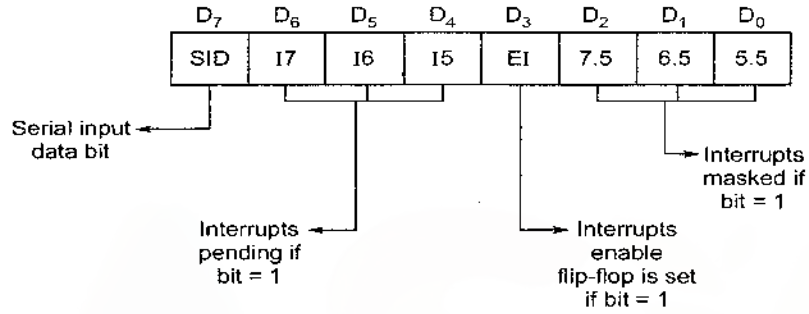
This instruction is commonly used when the execution of a code sequence cannot be interrupted.

M : Read Interrupt Mask

Opcode	Operand	Bytes	M-Cycles	T-States
RIM	None	1	1	4T

Description:

This is a multipurpose instruction used to read the status of interrupts 7.5, 6.5, 5.5 and to read serial data input bit.



Flags

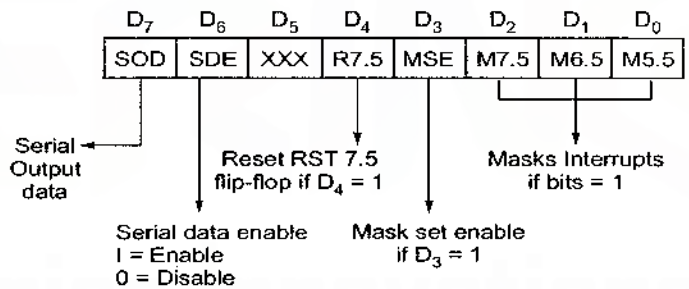
No flags are affected.

SIM : Set Interrupt Mask

Opcode	Operand	Bytes	M-Cycles	T-States
SIM	None	1	1	4T

Description:

This is a multipurpose instruction and used to implement the 8085 interrupts (RST 7.5, 6.5 and 5.5) and serial data output.



⇒ **SOD (Serial Output Data):** Bit D_7 of the accumulator is latched into the SOD output line and made available to a serial peripheral if bit $D_6 = 1$.

⇒ **SDE (Serial Data Enable):** If this bit = 1, enables the serial output. To implement serial output, this bit needs to be enabled.

⇒ **XXX:** Don't care

⇒ **R7.5 (Reset RST 7.5):** If this bit = 1, RST 7.5 flip-flop is reset. This is an additional control to reset RST 7.5.

⇒ **MSE (Mask Set Enable):** If this bit is high, it enables the functions of bits D_2, D_1, D_0. This is a master control over all the interrupt masking bits. If this bit is low, bits D_2, D_1 and D_0 do not have any effect on the masks.

⇒ **M7.5:** D_2 = 0, RST 7.5 is enabled.
= 1, RST 7.5 is masked or disabled

⇒ **M6.5:** D_1 = 0, RST 6.5 is correct.
= 1, RST 6.5 is masked or disabled.

⇒ **M6.5:** D_0 = 0, RST 5.5 is enabled.
= 1, RST 5.5 is masked or disabled.

Comments:

This instruction does not affect TRAP interrupt.

Counter and Time Delays

- Counters are used primarily to keep track of events.
- Time delays are important in setting up reasonably accurate timing between two events.
- Time delay can be introduced using a loop and total delay = time to execute instructions outside loop + time to execute loop instructions.

$$\boxed{T_D = T_O + T_{LA}}$$

where T_D = Total delay
T_O = Time to execute instructions outside loop.
T_{LA} = Time to execute loop instructions.

- Normally T_O is very small and neglected in most of the cases.
- The accuracy of time delay depends on the accuracy of the system's clock.
- Intel 8254 is a programmable timer chip that can be interfaced with microprocessor and programmed to provide with considerable accuracy.

Remember:

- Data copy instructions do not affect the flags.
- Operand PSW (Program Status Word) represents the contents of the accumulator and the flag register; the accumulator is high-order register and the flags are low-order register.
- Add and subtract are performed in relation to contents of the accumulator, however the increment or decrement operation can be performed in any register.
- Out of all instructions PUSH , CALL, RET, RSTn, INX, DCX, SPHL and PCHL uses 6 T-states for fetch machine cycle.
- DAD instruction uses bus idle machine cycle.
- In logical AND, AC is set and carry is reset.
- In other logical operation AC and carry both are reset and all other flags are changed according to result.
- NOT operation does not affect any flags.
- JMP: Instruction is a type of immediate addressing.
- Conditional jump instructions allow the microprocessor to make decision, based on certain conditions indicated by the flags.
- INX and DCX does not affect any of the flags.
- Compare instruction work like subtraction but content of accumulator and register does not change.
- Stack is a set of memory locations used to store binary information (byte) temporarily during execution of a program.
- Stack and stack pointer are two different things.
- Subroutine is also a program written outside main program.
- CALL and RET instructions are used for execution of subroutine and after execution, return to main program.

■■■■