

EI

MVI A, CCH

SIM

} find which interrupts are masked
find interrupts available
find serial data

- ① D3 is control over D2-D0, if it is 1 they are significant else invalid.
- ② D0-D2 indicate whether interrupt masked or available.
- ③ D4 → It is an extra provision to reset RST 7.5 flip flop.
- ④ D6 is a control bit over D7.
- ⑤ D7 is a serial data to be transmitted.

Q for the execution of falling instruction, find

- ① Interrupts masked RST 7.5
- ② " available RST 6.5, 5.5
- ③ Serial data transmitted. (1) Ans

{ EI
MVI A, CCH
~~DI~~ SIM

Q if content of A is 4BH in above instruction.

0000 1011

D3 = 1

- ① RST 6.5, 5.5
- ② 7.5
- ③ '0' Ans

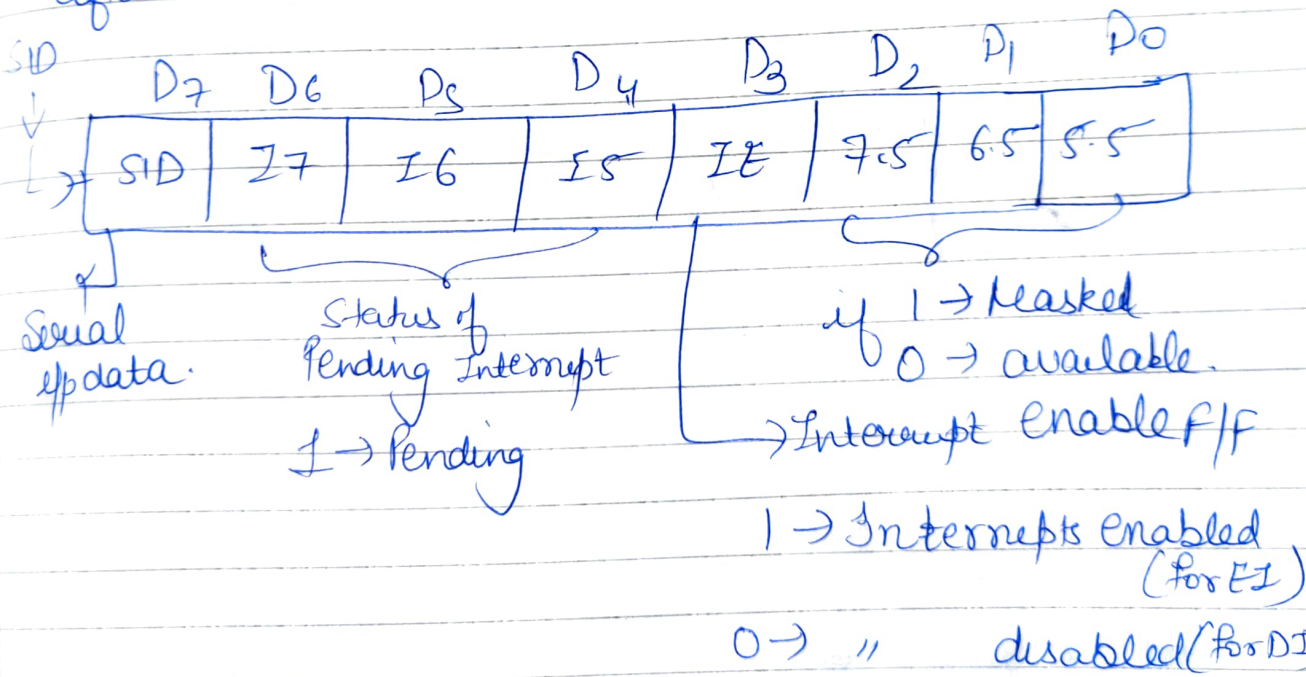
RIM Read interrupt mask

RIM - IB, 1, 4 → Read Interrupt Mask.

It is a multipurpose instruction used to know the status of pending interrupts & also to receive serial data through SID pin.

RIM
↓
Status
↓
A
↓
xx

⇒ Valid for RST 7.5, RST 6.5, RST 5.5. The status of the interrupts is loaded into accumulator after the execution of RIM instruction.



RIM A
↓
IC

- ⇒ D3 indicates whether interrupts enabled or disable
- ⇒ D0-D2 " whether interrupts masked or available
- ⇒ D4-D6 indicates status of pending interrupts
- ⇒ D7 is serial data received.

Q After the execution of RIM, the content of accumulator is 1C. Find

- 1) interrupts masked = RS7.5
- 2) " available = RS.5, RS1-6.5
- 3) " pending = RS.5
- 4) Serial data received. = 0

② B8 H

1011 1000

- 4) Serial - 0
- 1) Interrupt masked - 0
- 2) " available - 7.5, 6.5, 5.5
- 3) " Pending 5.5, 6.5

17/11/13

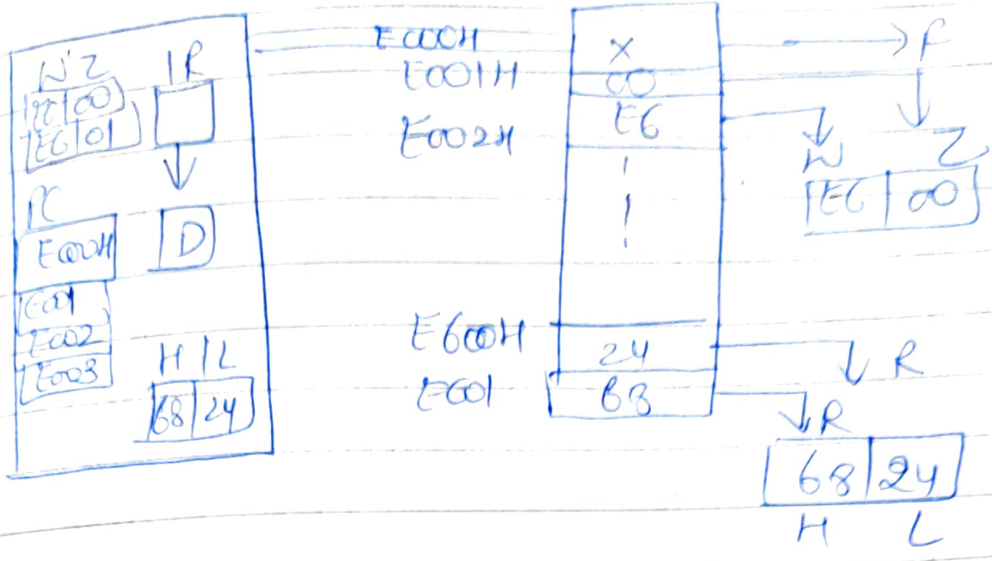
Special Instructions

LDA 16 bit address - 38

LHLD 16 bit Address - 3

Load HL pair ~~at~~ direct with data present at 16 bit address (2 Bytes data)

eg E000H : LHLD 6000H

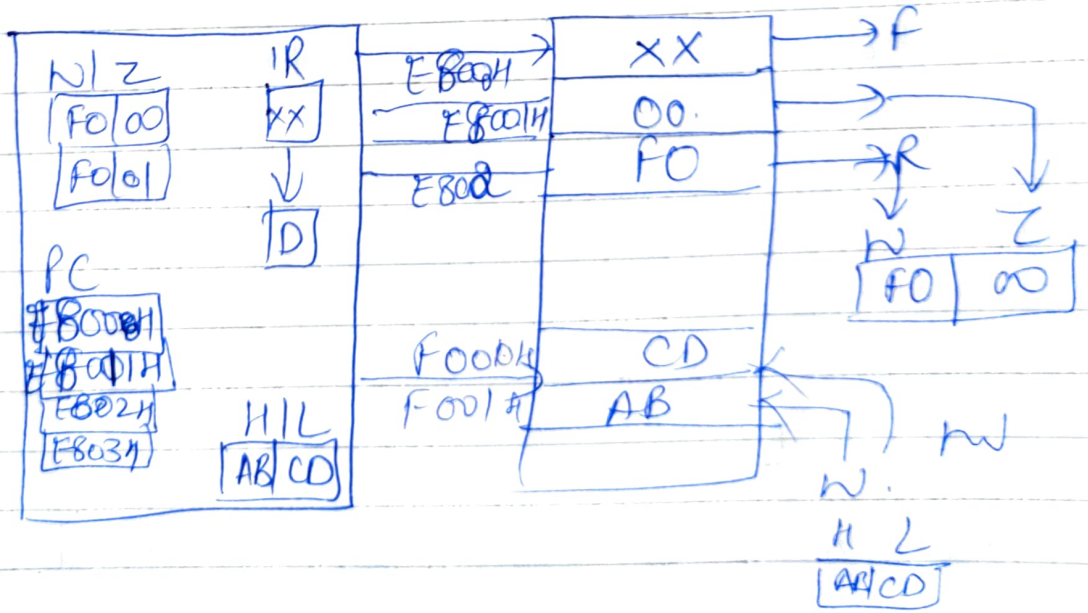


F R R R R → 5 m/c
 4 3 3 3 3 → 16 T states

SHLD 16 bit Address → 3B, 5, 16

Store the content of HL pair direct at 16 bit address

eg E800H : SHLD F000H ; if [HL] → ABCD H.



F R R N W → 5 m/c
 4 3 3 3 3 → 16 T.

Branching instruction

Learn 67

PCHL - 1 Byte, 1, 6
Load PC with HL content

PCHL is known as 1 byte unconditional jump

eg. 10F0: LXI H, 10FFH
10F3: MOV A, L
10F4: PCHL
10FF: HLT

H	L
10	FF

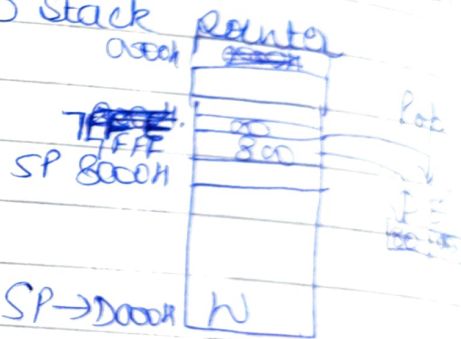
A [FF]

Register - Register Addressing mode → PCHL

SPHL - 1B, 1, 6 → Data Transfer, Register Addressing mode

Copy the content of HL pair to stack pointer

eg. LXI SP, D000H
LXI H, 8000H
SPHL
PUSH H
POP D
HLT



SP → D000H
H | L
80 | 00

What is value of SP & data at 7FFF after execution.
(8000H) (80H)

DAD Rp 1B, 3, 10

Add the content of Register pair to HL pair.

Accumulator is unchanged for the instruction
 Result is stored in HL pair.
 Affected. Carry out of 16 bits carry flag is
 remaining flags are unchanged.

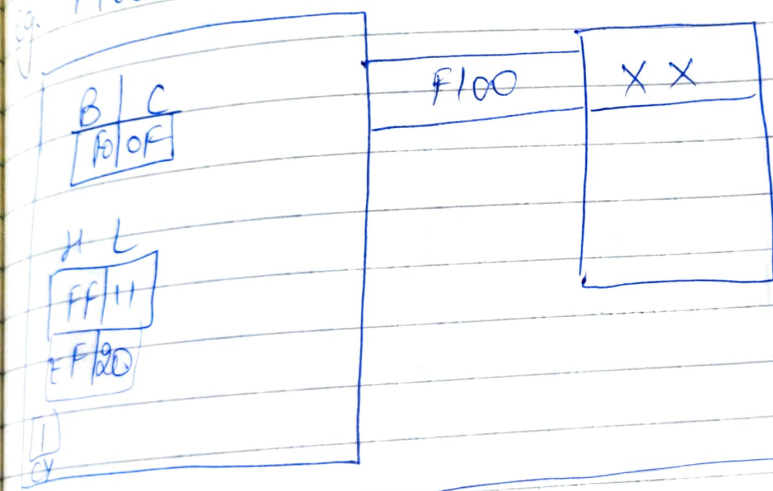
DAD B

DAD D

DAD H ← Multiply by 2. $\times 2$.

DAD SP

Flow: DAD B



foof
 +ff11
 EF20

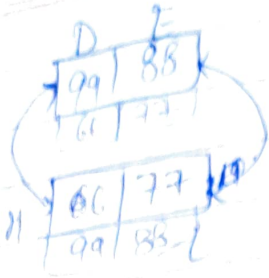
F B B
 4 3 3 → 10T.

BC & HL add not directly but using accumulator.

XCHG B, 1, 4.

Exchange the contents of DE & HL register pairs

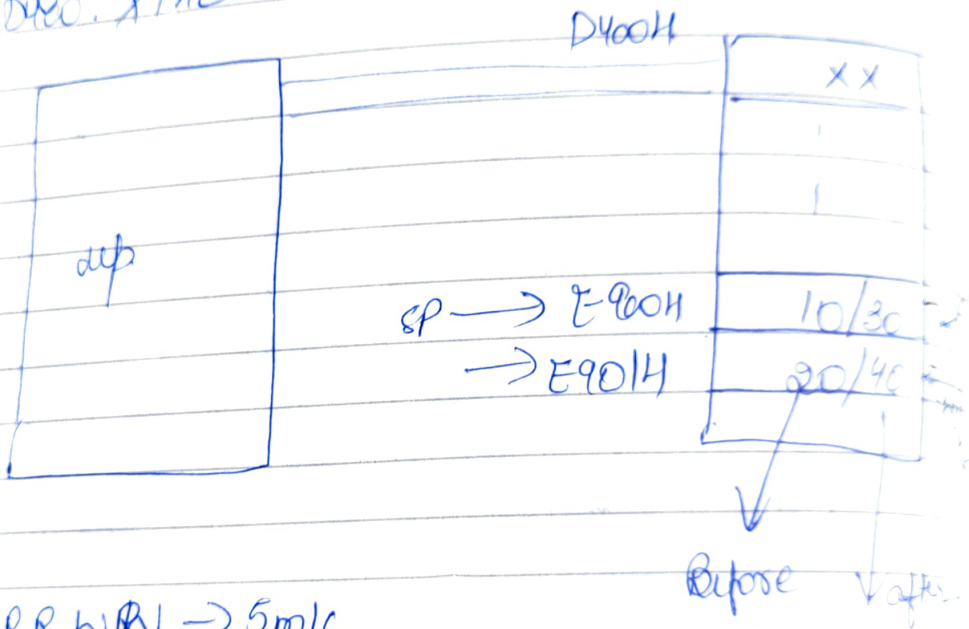
> Data Transfer instructions



* Data Transfer

XTHL - 1B, 5, 16

Exchange the data present at Top of stack (pointed by SP) with HL content
eg. D10: XTHL



FRRWRN \rightarrow 5m/c.
43383 \rightarrow 16T

DAA 1B, 1, 4

Decimal adjust accumulator after addition.
 \Rightarrow This is the only instruction which works with the status of auxiliary carry flag in BCD conversion. It converts 8bit data present in accumulator into two 4bit BCD no. used after addition.