

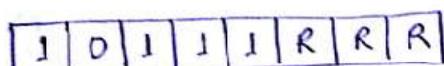
Microprocessors, UNIT- 4

1. Description of Some Important Instructions of 8085

③⑥ CMP : Compare with Accumulator

Opcode	Operand	Bytes	M-cycles	T-states	Hex code
CMP	R	1	1 (F)	4	Reg. Hex
CMP	M	1	2 (FR)	7	B B8
					C B9
					D BA
					E BB
					H BC
					L BD
					M BE
					A BF

Opcode
format



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.

③⑦ CPI : Compare Immediate with Accumulator

Opcode	Operand	Bytes	M-cycles	T-states	Hex code
CPI	8-bit data	2	2 (FR)	7	FE

Description : The 8-bit data is compared with the contents of the accumulator. The values being compared remain unchanged and the results of the comparison are indicated by setting the flags as follows:

- If $(A) < \text{Data}$: CY = 1, ZF = 0
- If $(A) = \text{Data}$: CY = 0, ZF = 1
- If $(A) > \text{Data}$: CY = 0, ZF = 0

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

eg : a) $B = 62 \text{ H}$ $A = 57 \text{ H}$

A	57	XX	F
B	62	XX	C

CMP B

A	57	XX
B	62	XX

$$F \rightarrow S=1, Z=0, AC=1 \\ P=1, CY=1.$$

b) $A = 57 \text{ H}$

CPT C2 H After execution

$$S=0, Z=1, AC=1 \\ P=1, CY=0.$$

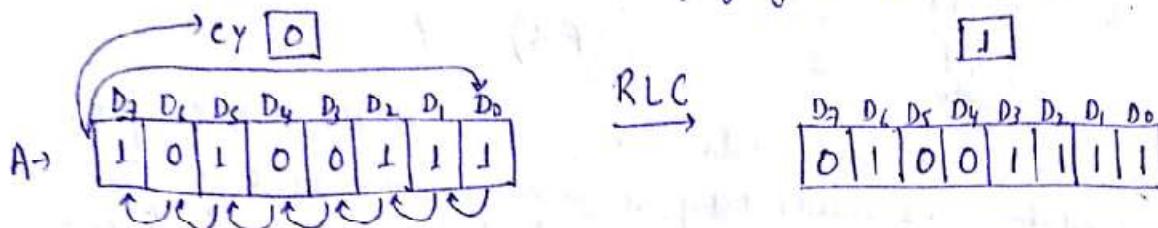
3) RLC : Rotate Accumulator Left

Opcode	Operand	Bytes	M-cycles	T-states	Hex Code
RLC	None	1	1 (F)	4	07

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.

eg - Rotate the contents of the accumulator left, assuming it contains A7 H and the carry flag is reset to 0.



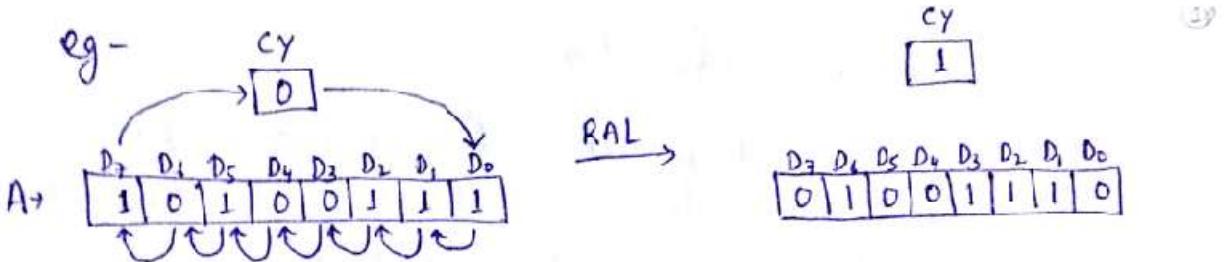
40

2016-17

RAL : Rotate Accumulator Left through Carry.

Opcode	Operand	Bytes	M-cycles	T-states	Hex code
RAL	None	1	1 (F)	4	17

Description : Each binary bit of the accumulator is rotated left by one position through the Carry bit. Bit D_7 is placed in the Carry flag and the Carry flag is placed in the D_0 .
 Flags : CY is modified according to bit D_7 . S, Z, AC, P are not affected.



41

2016-17

RAR : Rotate Accumulator Right through Carry

Opcode	Operand	Bytes	M-cycles	T-states	Hex code
RAR	None	1	1 (F)	4	1F

Description : Each binary bit of the accumulator is rotated right by one position through the Carry flag. Bit D_0 is replaced 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.



20	<u>DAA</u>	Decimal-Adjust Accumulator	5-014-15			
Opcode	Operand	Bytes	M-cycles	T-states	Hex Code	2013-14
DAA	None	1	1 (F)	4	27	91

Description: The contents of the accumulator are changed from a binary value to two 4-bit binary coded decimal (BCD) digits. This is the only instruction that uses the auxiliary flag (internally) to perform the binary-to-BCD conversion.

Flags: All flags are affected.

Instruction DAA converts the binary contents of the accumulator as follows:

- If the value of the lower order 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 lower order four bits.
- 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.

eg → Add 68_{BCD} to the accumulator which contains 85_{BCD} .

$$\begin{array}{r}
 A = 85_{BCD} = 1000 \ 0101 \\
 68_{BCD} \quad \quad \quad 0110 \ 1000 \\
 \hline
 153_{BCD} \quad \quad \quad 1110 \ 1101
 \end{array}$$

The binary sum is $(ED)H$. The values of both, low-order and

19 ~~DAD~~ ²⁰¹³⁻¹⁴ Add Register Pair to H and L Registers

Opcode	Operand Bytes	M-cycles	T-states	Hex-codes	Reg. Pair	Code.
DAD	R _p	1	3 (FBB)	10		
			^{Bus Idle}			
Opcode format	0 0 R R I 0 0 1				B	09
DAD D	0 0 0 1 1 0 0 1	= 19 H			D	19
					H	29
					SP	39

Description: The 16-bit contents of the register pair are added to the contents of H-L pair. The result is stored in H-L pair.

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

eg:- DAD D
HL = HL + R

CY | 0

B	C	
D	10	E
H	40	L

After \rightarrow
execution

CY | 0

B	C	
D	10	E
H	50	L

⑪ XTHL ; Exchange H and L with Top of Stack

Opcode	Operand	Bytes	M-cycles	T-states	Hex code
XTHL	None	1	5(FRRW)	16	E3

② 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); however the contents of the stack pointer are not altered.

Flags: No flags are affected.

H	A2	L	57
	2700		38
	2701		67
SP	2700		

XTHL

H	67	L	38
	SP	2700	
	2700		57

2700		57
2701		A2
2702		

⑨ ~~XCHG~~²⁰¹⁴⁻¹⁵ : Exchange H and L with D and E

Opcode	Operand	Bytes	M-cycles	T-states	Hex code
XCHG	None	1	1(F)	4	EB

Description: The contents of H are exchanged with the contents of register D. The contents of L are exchanged with the contents of register E.

Flags: No flags are affected.

A		F	
B		C	
D	20	E	40
H	70	L	80

XCHG

A		F	
B		C	
D	70	E	80
H	20	L	40

⑥ STAX : Store Accumulator Indirect

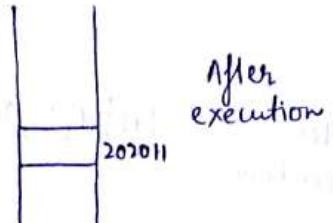
Opcode	Operand	Bytes	M-cycles	T-states	Hex Code
STAX	R _p	1	2 (FW)	7	Reg. code BC 02 DE 12

Description : The contents of the accumulator are copied into the memory location specified by the contents of the register pair.

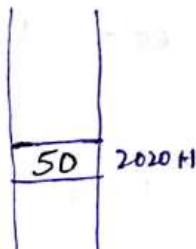
Flags : No flags are affected.

eg:- STAX B

A	50	F	
B	20	C	20
D		E	
H		L	



A	50	F	
B	20	C	20
D		E	
H		L	



⑦ SHLD : Store H and L Registers Direct

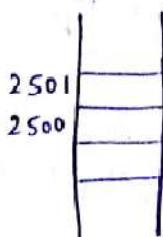
Opcode	Operand	Bytes	M-cycles	T-states	Hex Code
SHLD	16-bit Address	3	5 (FRRWW)	16	22

Description : The contents of register L are stored into memory location specified by the 16-bit address. The contents of register H are stored into the next memory location.

Flags : No flags are affected.

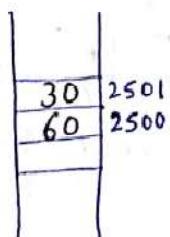
eg → SHLD 2500H

H	30	L	60
---	----	---	----



After execution

H	30	L	60
---	----	---	----



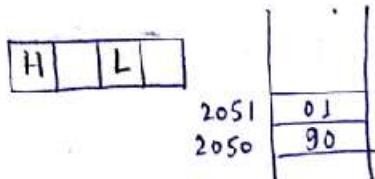
⑧ LHLD : Load H and L Registers Direct

Opcode	Operand	Bytes	M-cycles	T-states	Hex Code
LHLD	16-bit Address	3	5(FRRRR)	16	2A

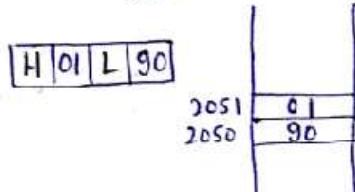
Description : The instruction copies the contents of the memory location pointed by the 16-bit address in register L and copies the contents of the next memory location in register H.

Flags : No flags are affected.

eg - LHLD 2050 H Hex Code : 2A
 50
 20



After
execution



20/11/13
58

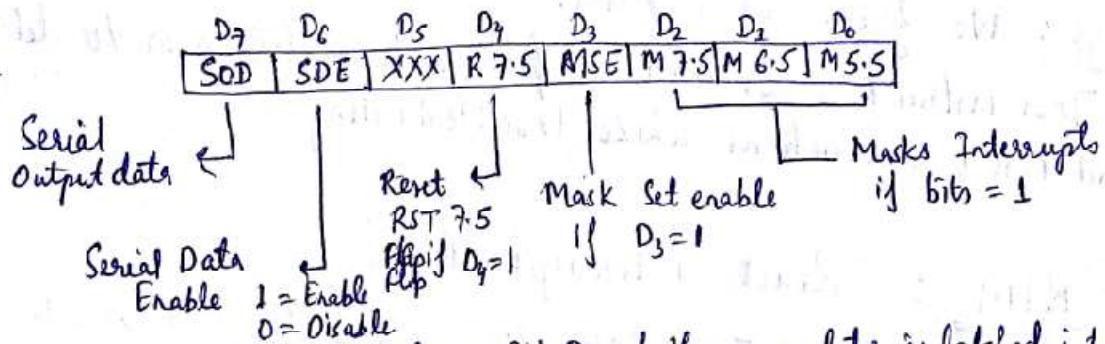
Typ. group 1

SIM : Set Interrupt Mask.

Opcode	Operand	Bytes	M-cycles	T-states	Hex Code
SIM	None	1	1 (F)	4	30

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

The instruction interprets the accumulator contents as follows:-



- SOD - Serial Output Data : Bit D₇ of the accumulator is latched into the SOD output line & made available to a serial peripheral if bit D₆ = 1.
- SDE - Serial Data Enable : If this bit = 1, it enables the serial output.
- XXX - Don't care
- MSE - Mask Set Enable : If this bit is high, it enables the functions of bits D₂, D₁ & D₀. This is a master control over all the interrupt masking bits. If this bit is low, bits D₂, D₁ & D₀ do not have any effect on the masks.
- M 7.5 - D₂ = 0, RST 7.5 is enabled ; D₂ = 1, RST 7.5 is masked or disabled
- M 6.5 - D₁ = 0, RST 6.5 is enabled ; D₁ = 1, RST 6.5 is disabled or masked
- M 5.5 - D₀ = 0, RST 5.5 is enabled ; D₀ = 1, RST 5.5 is masked or disabled.

eg → Write instructions to enable interrupt RST 5.5 and mask other interrupts.

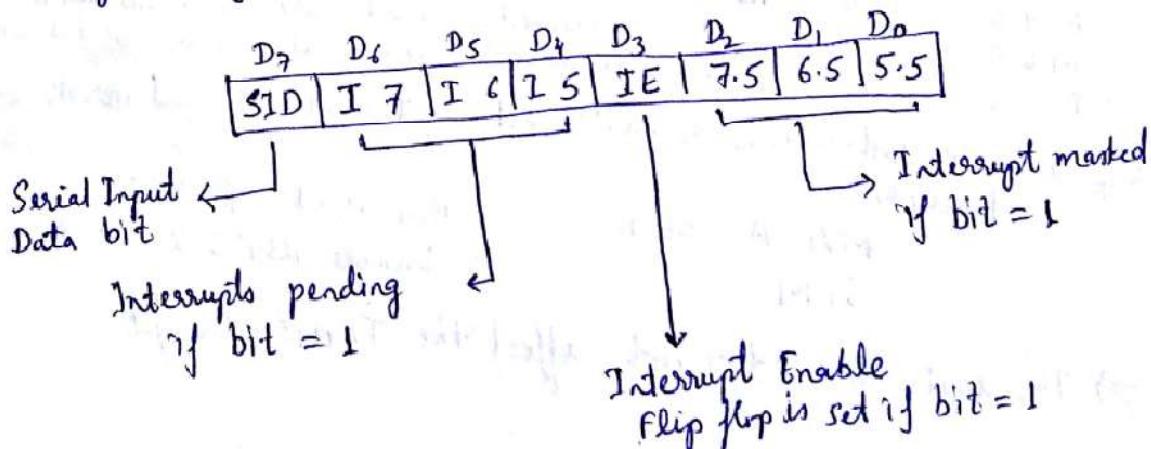
MVI A, 0E H ; Bits D₃=1 & D₀=0
SIM ; Enable RST 5.5.

⇒ This instruction does not affect the TRAP interrupt.

(57) RIM : Read Interrupt Mask

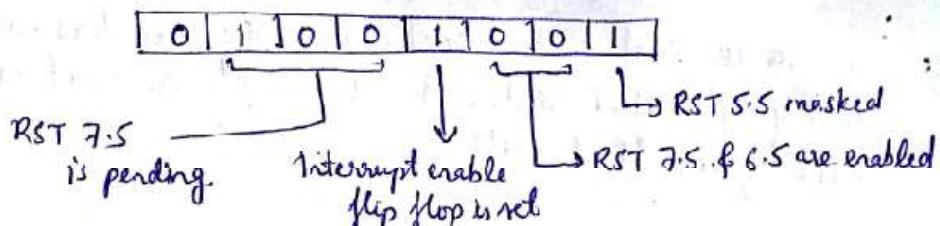
Opcode	Operand	Bytes	M-cycles	T-states	Hex-Code
RIM	None	1	1 (F)	4	20

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. The instruction loads eight bits in the accumulator with the following interpretations.



Flags : No flags are affected

Eg - (A) : 49H



2016/1
51)

PUSH : Push Register Pairs onto Stack.

Opcode	Operand	Bytes	M-cycles	T-states	Hex Codes	Hex.
PUSH	R _p	1	3 (SWW)	12	R _p B D H PSW	C5 D5 E5 F5

Description : 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 is decremented again and the contents of the low-order register (C, E, L, flags) are copied to that location.

Flags : No flags are modified.

eg — Assume the stack pointer register contains 2099 H, register B contains 32 H and register C contains 57 H. Save the contents of the BC register pair on the stack.

PUSH B

Register contents
before execution

B

32	57
C	

SP

2099

Stack contents
after execution

2099

57
32
XX

2098
2099

Register contents
after execution

B

32	57
C	

SP

2097

⇒ The contents of the source register are not altered after the PUSH instruction.

52) POP : Pop off Stack to Register Pair 0-03-14

Opcode	Operand	Bytes	M-cycles	T-states	Hex code
POP	R _p	1	3(FRR)	10	Reg. code
					C1
					D1
					E1
					F1
					PSW

Description : The contents of the memory location pointed out by the stack pointer register are copied to the low-order register (C, E, L & 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.

2g - POP H

Register contents
before execution

H	XX	XX	2
SP	2090		

Stack
contents

2090	F5
2091	01
2092	

Register contents
after execution

H	01	F5	2
SP	2092		

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

(46) ~~2014c13~~ CALL : Unconditional Subroutine Call

Opcode	Operand	Bytes	M-cycles	T-states	Hex Code
CALL	16-bit address	3	5 (SRRWW)	18	CD

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.

Eg - Write CALL instruction at memory location 2010 H to call a subroutine located at 2050 H.

Memory Address	Hex Code	Mnemonics
2010	CD	CALL 2050 H
2011	50	
2012	20	

Execution of CALL : The address of program counter (2013 H) is placed on the stack:

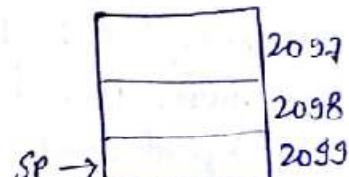
Stack pointer is decremented to 2098

MSB is stored

Stack pointer is again decremented

LSB is stored

Call address (2050 H) is temporarily stored in internal WZ registers and placed on the bus for the fetch cycle.



⇒ : The CALL instruction should be accompanied by one of the return (RET or conditional return) instructions in the subroutine.

(47) Conditional Call to Subroutine

Opcode	Operand - 16-bit address	Description	Flag Status	Hex Code	(S R RWW) M-cycles/T-states
CC	Call on Carry	$CY = 1$	DC	2	/9 (if condition is not true)
CNC	Call on No Carry	$CY = 0$	D4		
CP	Call on positive	$S = 0$	F4		
CM	Call on minus	$S = 1$	FC	5	/18 (if condition is true)
CPE	Call on Parity Even	$P = 1$	EC		
CPO	Call on Parity Odd	$P = 0$	E4		
CZ	Call on Zero	$Z = 1$	CC		
CNZ	Call on No Zero	$Z = 0$	C4		

Flags : No flags are affected.

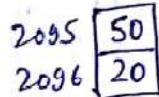
(48) ~~2013-14~~ RET : Return from Subroutine Unconditionally

Opcode	Operand	Bytes	M-cycles	T-states	Hex code
RET	None	1	3 (FRR)	10	C9

Description : The program sequence is transferred from the subroutine to the calling program. The two bytes from the top of stack are copied into the program counter and the program execution begins at the new address.

Flags : No flags are affected.

eg → Assume the SP is pointing at location 2095H.



After instruction RET, the program execution is transferred to location 2050 and the SP is shifted to 2097.

⇒ This instruction is used in conjunction with CALL or conditional call instructions.

(49)

Opcode	Description	Flag Status	Operand: None	(S R R)	
				Hex Code	M-Cycles/T-States
RC	Return on Carry	$CY=1$	D8	1/6	(if condition is
RNC	Return on No Carry	$CY=0$	D0		not true)
RP	Return on positive	$S=0$	F0		
RM	Return on Minus	$S=1$	FB		
RPE	Return on Parity even	$P=1$	EB	3/12	(if condition is
RPO	Return on Parity odd	$P=0$	ED		true).
RZ	Return on Zero	$Z=1$	C8		
RNZ	Return on No Zero	$Z=0$	CO		

Flags: No flags are affected.

8085 Assembly Language Programs & Explanations

Problem ① $A = 93H$
 $C = B7H$

Instruction ADD C

$$\begin{array}{r} \text{Op} \quad D_7 \ D_6 \ D_5 \ D_4 \ D_3 \ D_2 \ D_1 \ D_0 \\ \hline 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \\ \hline \boxed{1} \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \\ \text{Acc} \end{array}$$

Flag status: $S=0, Z=0, CY=1$

② Add. $35H$ to the sum of problem ①.

$$\begin{array}{r} \text{ADD } 35H \\ \hline \boxed{1} \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \\ \boxed{0} \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \\ \hline \boxed{0} \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \\ \text{Acc} \end{array}$$

$$\begin{array}{r} \text{Op} \quad D_7 \ D_6 \ D_5 \ D_4 \ D_3 \ D_2 \ D_1 \ D_0 \\ \hline \boxed{1} \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \\ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \\ \hline \boxed{0} \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \\ \text{Acc} \end{array}$$

Problem ③ A reg. holds FFH . Illustrate the differences between in the flag. set by adding $01H$ & by incrementing the acc. content.

Sol. ① ADD 01H

$$\begin{array}{r} \text{Op} \\ \hline \text{(A)} \quad FFH = 1111 \ 1111 \\ + 01H = 0000 \ 0001 \\ \hline \boxed{1} \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \end{array}$$

$$\boxed{S=0, Z=1, CY=1}$$

② INR A

$$\boxed{A = 00H}$$

$$\boxed{S=0, Z=1, CY=NA}$$

Problem : WAP to perform the following functions & verify the output.

- (a) Load the no. 88H in D.
- (b) Load the no. 6FH in C.
- (c) Increment the contents of C by one.
- (d) Add the contents of C and D, and display the sum at the I/O port-1.

- Draw the flow chart also.

MVI D, 88H

MVI C, 6FH

INR C

MOV A, C

ADD D

OUT 01H

HLT

Problem - (1) Reg B has 65H and the acc. has 07H. Subtract the content of register B from the content of the acc.

SUB B

② WAP to do the following -

- ① load the no. 30H in reg. B and 39H in C.
2. subtract 39H from 30H
3. display the answer at port 1.

MVI B, 30H

MVI C 39H

MOV A, B

SUB C

OUT 01H

HLT

problem - Assume B holds 92H, A holds 15H. Illustrate results of the instruction ORA B, XRA B & CMA.

ORA B

10010010

00010101

10010111

XRA B

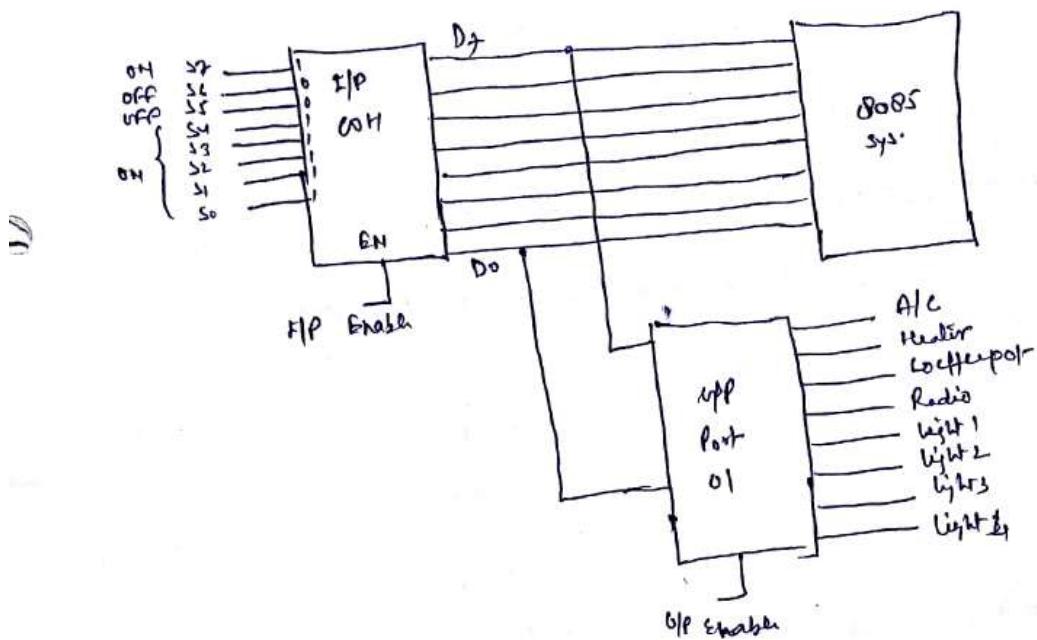
10010010

00010101

10000101

CMA 11101010

Problem - To conserve energy and to avoid an electrical overload on a hot afternoon, implement the following procedures to control the appliances throughout the house.
(fig) Assume that the control switches are located in the kitchen, and they are available to anyone in the house.



26 write a set of instructions to -

- set of instructions to -

 - ④ turn on A/C if switch S7 of the input port coil is ON.
 - ⑤ ignore all other switches of the input port even if several attempts to turn on other appliances.

$$\text{Suf. Data} \quad 10000000 = 804$$

mu) A, Data

AM 80H

WT of H

HLT.

- ② Keep the radio on continuously without affecting the other function of other appliances, even if someone turns off the switch S_4 .

Sol.

$$\text{IN ROM: } A = D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0$$

$$\text{ORF ROM} = 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0$$

$$\overline{D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0}$$

- ③ Assume it is winter, and turn off the air conditioner without affecting the other appliances.

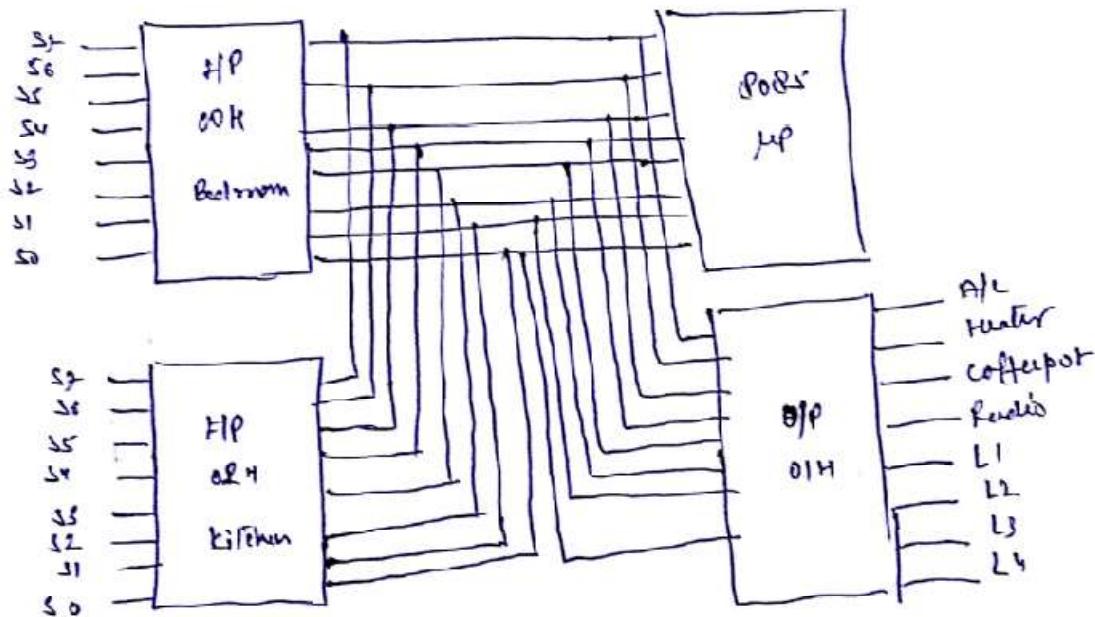
$$\text{IN ROM: } A = D_7 D_6 D_5 D_4 D_3 D_2 D_0$$

$$\text{ANT ROM} = \overline{0 \ 1 \ 1 \ 1 \ 1 \ 1}$$

$$\overline{D \ D_6 D_5 D_4 D_2 D_1}$$

Flag status: $CF=0$, others depends on the data bits

Problem - An additional input port with 8 switches and the address 01H (fig) is connected to the system shown in last fig. to control the same appliances and lights from the bedroom as well as from the kitchen. Write instructions to turn on the device from any of the input ports.



Soln

```

IN 00H
MOV B, A

IN 02H
ORA B

OUT 01H
HLT.

```

Problem 2016-17 Six bytes of data are stored in memory locations starting at $10000H$. Add odd byte data bytes. Use register B to save any carry generated, while adding the data bytes. Display the entire sum at two output ports, or store the sum at two consecutive memory locations $10070H$ & $10071H$. 2012-13

Op.

```

XRA A
MOV B, A
MVI C, 06H
LDI H, 10050H

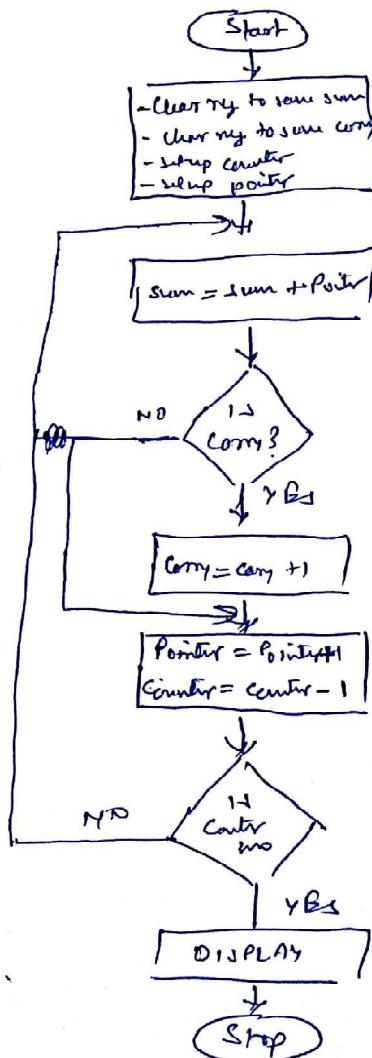
NEXTBYTE: ADD M
JNC NEXTMEM
INR B
NEXTMEM: INX H
DCR C
JNC NEXTBYTE
WT PORT1
MOV A,B
MOV PORT2
HLT

```

```

LDI H 10050H
MOV M,A
INR H
MOV M,B
HLT

```



DATA - 20 AL
21 FA
32 DF
53 BF
54 98
55 08

Problem ²⁰¹⁶⁻¹⁷ - 16 bytes of data are stored in memory locations at $2040H$ to $205FH$. Transfer the entire block of data to new mem. locations starting at $2070H$.

Ans - Write the inst. to add the contents of the mem. $2040H$ to (A) , and subtract the contents of the $2041H$ from the first sum. Assume $A \rightarrow 30H$, $2040 \rightarrow 68H$ & $2041 \rightarrow 7FH$.

LXI H, 2040H

ADD M

INX H

SUB M

HLT.

Time delay using Register Pair

2015-16
200

{ delay of
0.5μs with
crystal for
6MHz

LXI B, 2384H — T statis
10

Loop:	DCX B	6
	Mov A, C	4
	ORA B	4
	JNZ Loop	10/7

2012-13
20 ms

$$2384H = 9092_{10}$$

$$T_L = 0.5 \times 10^{-6} \times 24 \times 9092_{10} \approx 109 \text{ ms.}$$

(without adjusting for last cycle)

$$\begin{aligned} \text{Total} &\Rightarrow 109 \text{ ms} + T_0 \\ &= 109 \text{ ms} + 5 \text{ ms} \approx \underline{\underline{109 \text{ ms}}} \end{aligned}$$

Using Nested loop (loop inside loop)

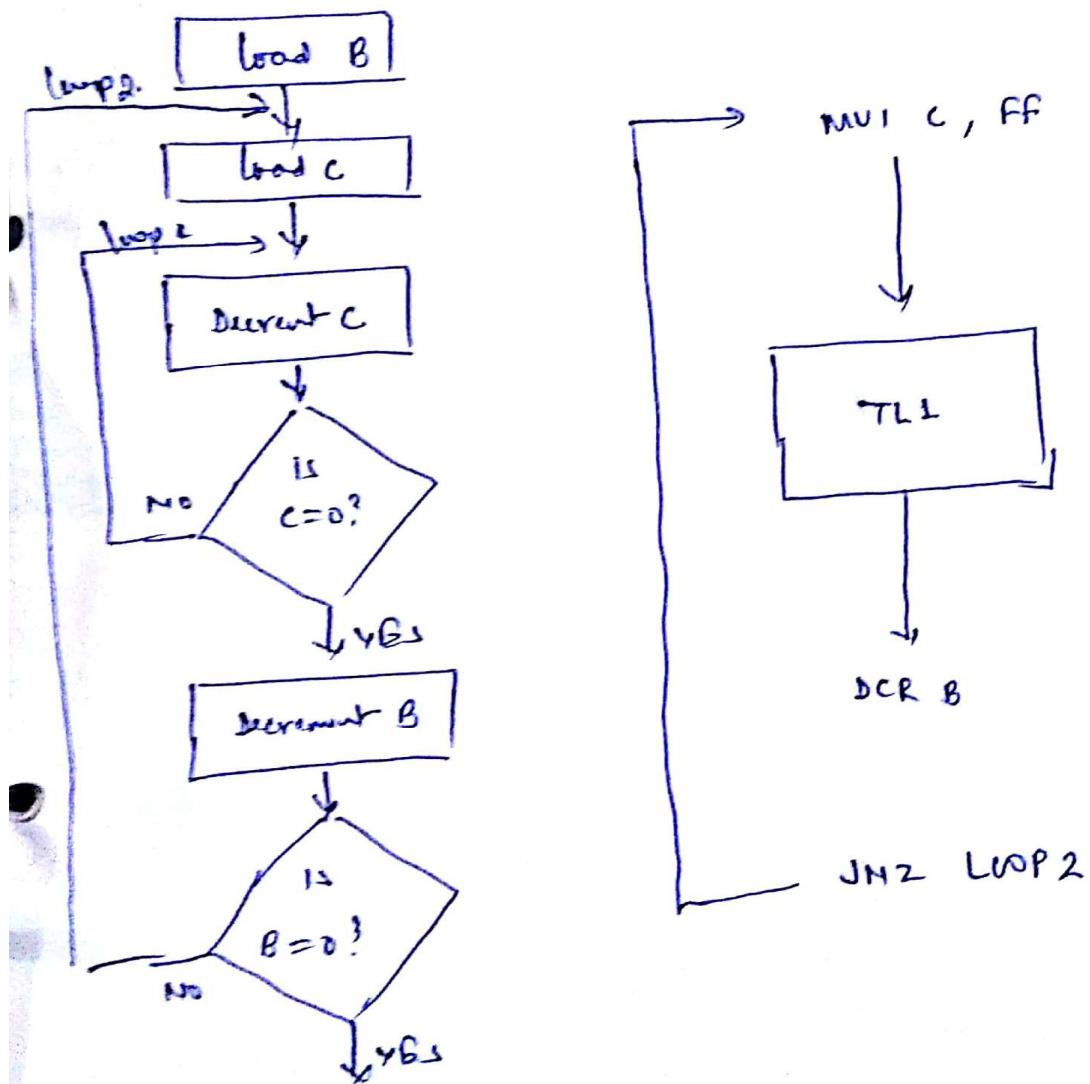
		<u>T statis</u>
	MVI B, 38H	7
Loop1:	MVI C, FFH	7
Loop2:	DCR C	4
	JNZ Loop1	10/7
	DCR B	4
	JNZ Loop2	10/7

$$T_{L1} \text{ (delay in loop 1)} = 17.83.5 \text{ ms}$$

$$T_{L2} = 56 \left(T_{L1} + (T_{\text{stat}} \times 0.5 \text{ ms}) \right)$$

$$= 56 (17.83.5 \text{ ms} + 10.5 \text{ ms})$$

$$= 100.46 \text{ ms}$$



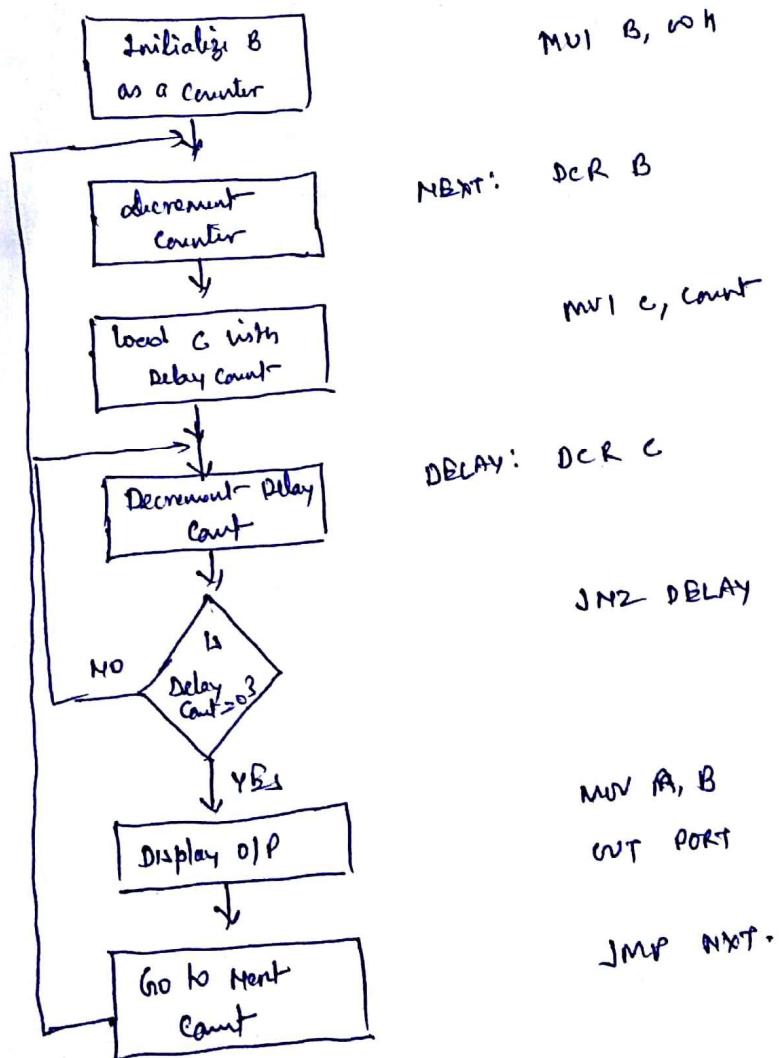
Hexadecimal Counter

2012-13

1 sec delay

Ques. WAP to count continuously in hexadecimal from ~~00~~ FFF to 00 in a system with a 0.5us clock period. Use register C to setup a one millisecond (ms) delay between each count and display the no.s at the port pins.

Sol.



Delay Calculation

$$\begin{aligned} T_L &= 14 T_{\text{status}} \times T_{\text{(Clock period)}} \times \text{Count} \\ &= 14 \times 0.5 \times 10^{-6} \times x \\ &= 7.0 \times 10^{-6} \times x \end{aligned}$$

DCR B — 4

MVI C, COUNT — 7

MVR A, B — 4

OUT PORT — 10

JMP — 10

35 T

$$T_0 = 35 \times 0.5 \times 10^{-6}$$

$$T_{\text{Total}} = T_D + T_0$$

$$1 \text{ ms} = 12.5 \times 10^{-6} + (7.0 \times 10^{-6}) \times x$$

$$x = 140_{10} \Rightarrow \underline{\underline{80}} \text{ H Ans.}$$

Problem (Zero to Nine : MOD 10 counter). 2016-17

WAP to count 0 to 9 with a 1 sec. delay between each count. At the count of 9, the count should reset itself to 0 and repeat the sequence continuously. Use reg pair HL to set up the delay, and display each count at one of the o/p ports.

Assume CLK freq = 1 MHz.

Help - use - LXI, DCX, INX.

Sol.

START: MVI B, 00H

DISPLAY: OUT PORT

LXI H, 16bit

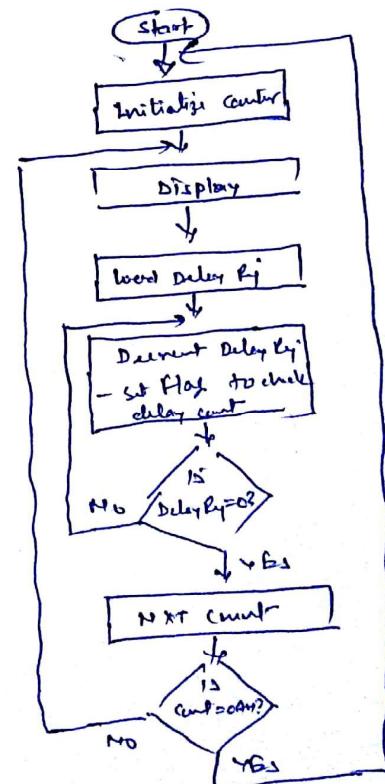
LOOP: DCX H
MOV A, L
ORA H

JNZ LOOP

INR B
MOV A, B
CPF 0A H

JNZ DISPLAY

JZ START



Delay Calculation

$$T_L = 24T \times T \times \text{Count}$$

$$1 \text{ sec.} = 24 \times 1.0 \times 10^{-6} \times x$$

$$x = 41666 = 42 \text{ CLK}$$

T_{outside}

OUT	- 10
Lx1	- 10
INR	- 4
MVR	- 4
CPF	- 7
JM2	- 10
	<u><u>45T</u></u>

Accurate -

$$T_{\text{total}} = T_0 + T_L$$

$$1 \text{ sec.} = (45 \times 1.0 \times 10^{-6}) + (24 \times 1.0 \times 10^{-6} \times \text{Count})$$

$$\text{Count} = 41665$$

Generating Pulse Waveforms

Problem: WAP to generate a continuous ~~square~~²⁰¹²⁻¹³ wave with the period of 500 ms. Assume the sys. clock period is 325 ms. and use bit D0 to output the square wave.

Hint: RAL, RAR, RLC, RRC

Sol. MVI D, AA -7

ROTATE: MW A,D -4

RLC -4

MW D,A -4

ANI 01H -7

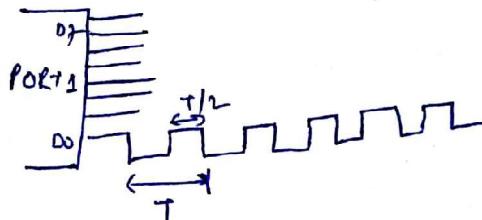
OUT 01H -10

MVI B, Count -7

DELAY: DCR B -4 }

JNZ ~~DELAY~~ -10/-7 }

JMP ROTATE -10.



$$T = 500 \text{ ms}$$

$$T/2 = 250 \text{ ms}$$

$$T_0 = 46 \times 325 \text{ ms} = 14.95 \text{ ms}$$

$$T_L = 14 \times 325 \text{ ms} \times (\text{Count} - 1) + 11 \times 325 \text{ ms}$$

$$= 4.5 \text{ ms} (\text{Count} - 1) + 3.575 \text{ ms}$$

$$T_{\text{total}} = T_0 + T_L$$

$$250 \text{ ms} = 14.95 \text{ ms} + 4.5 \text{ ms} (\text{Count} - 1) + 3.575 \text{ ms}$$

$$\text{Count} = 52.4_{10} = \underline{\underline{34H}}$$

Stack & subroutines

2016-17

Stack :

* A set of memory locations in the R/W memory, specified by a programmer in a main program.

- These memory locations are used to store binary information (bytes) temporarily during the execution of a program.
- The beginning of the stack is defined by the pointer called stack pointer (SP), with the help of instruction -

LXI SP, 16 bit data.

Data bytes in the register pairs of the 8085 CPU can be stored on the stack (two at a time) in reverse order.

by using the instruction PUSH.

- Data bytes can be transferred from the stack to the respective registers by using the instruction POP.
- SP tracks the storage and retrieval of the information.

Ex.

```
LXI SP, 2400H  
LXI H, 2100H  
LXI B, 2200H  
MOV A, M  
PUSH H  
PUSH B  
PUSH PSW  
  
↓  
POP PSW  
POP H
```

Problem

Write 20 ms time delay
subroutine using Reg Pair.
Clear Z flag without
affecting any other flags
in the flag reg. & return
to the main programme

Prob: Write a program to perform the following functions: 2-012-13

1. Clear all flags
2. Load zero in the acc., & demonstrate that the zero flag is not affected by the data transfer instruction.
3. logically OR the acc. with itself to set the zero flag, and display the flag at the port 1 or store all the flags on the stack.

Sol.

```
LXI SP, XX99H  
MVI L, 00H  
PUSH H  
POP PSW  
MVI A, 00H  
PUSH PSW  
POP H  
MOV A, L  
WT 00H
```

MVI A, 00H

ORA A

PUSH PSW

POP H

MOV A, L

ANI 40H

OUT 01H

HLT

MVI A, 00H

ORA A

PUSH PSW

HLT.

2012-13

- Problem - 1. WAP to count continuously in bin with a one sec. delay between each count.
2. W A service routine at XX70H to flash FFH five times when the program is interrupted, with the same appropriate delay between each count.

LXI SP XX99H

BI

MVI A, 00H

NEXTCNT: OUT 01H

MVI C, 01H

CALL DELAY

INR A

JMP NEXTCNT

SERV :

PUSH B
 PUSH PSW
 MVI B, 0AH
 MVI A, 00H

FLASH :

OUT PORT
 MVI C, 01H
CALL DELAY
CMA
DEC B
 JNZ FLASH
POP PSW
POP A
BI
RET.

The 8085 interrupt & vector locations. 2016-17, 2015-16
2014-15

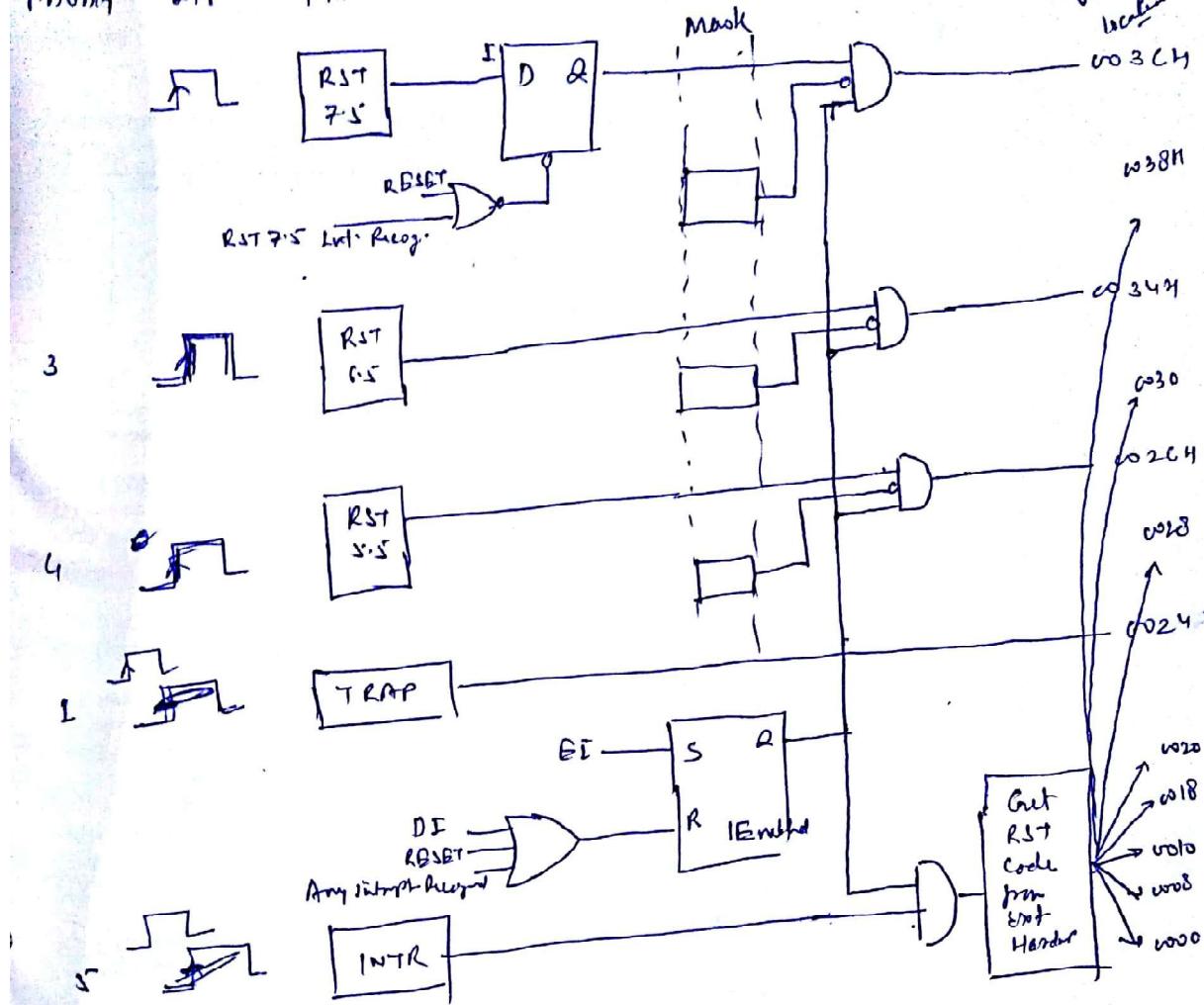
16-17, 2015-16
214

~~015-10~~

2014

2019-15

Prisoners slip pins



TRAP - highest priority, Nonmaskable, need not be enabled & cannot disabled.

- transfers 0024H.

- generally used for critical events

(RST \Rightarrow 5, 6, 5, 5) - these are enabled under program control with two

instructions: EF & SIM.