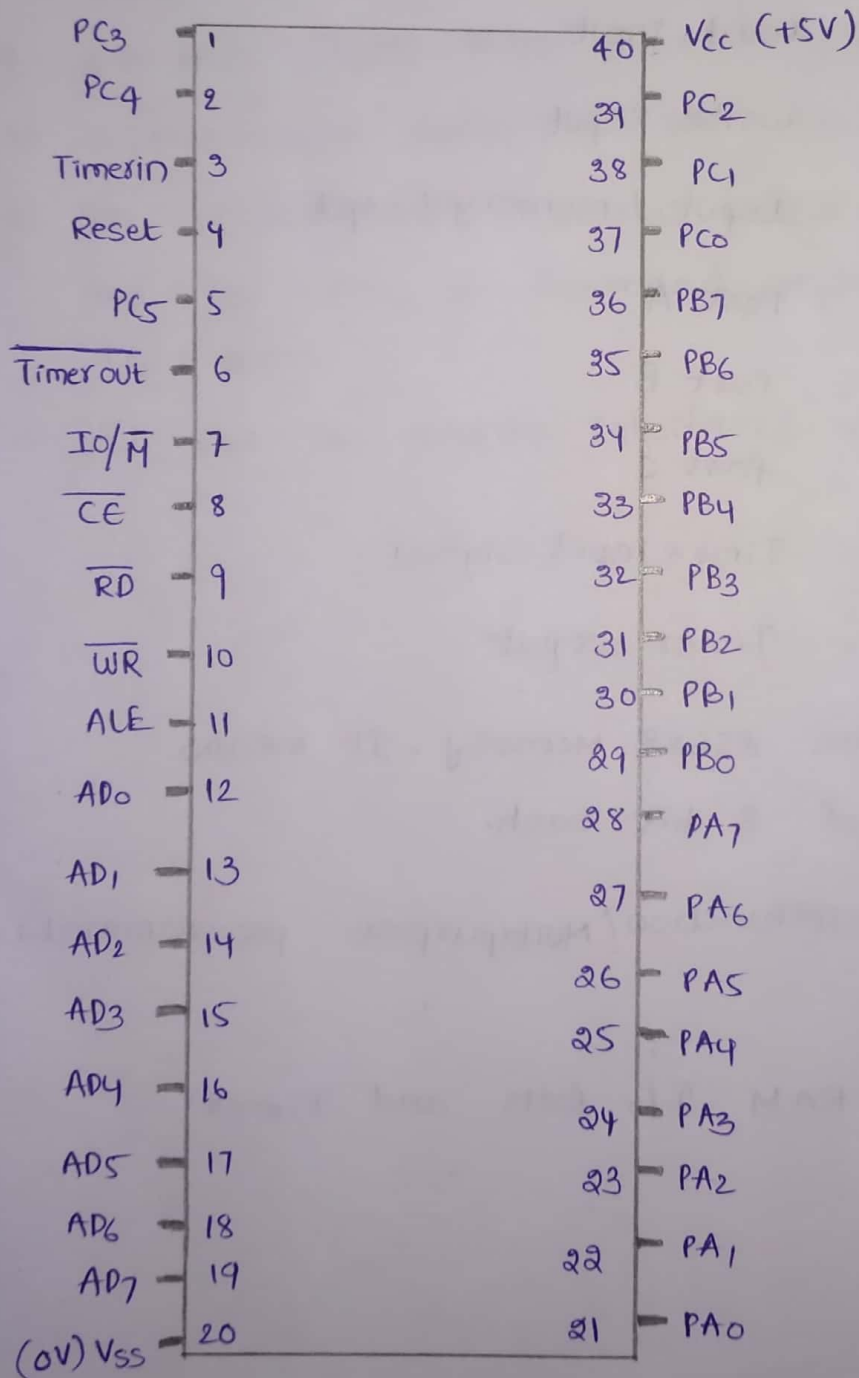


### **Group 10**

1. Draw neatly the pin diagram and architectural representation of a multipurpose programmable device 8155. Describe about its control logics, I/O ports. How can you determine the addresses for I/O ports, timer and control register?
2. Illustrate the interfacing of seven segment LED output ports using 8155.
3. Write a program to count from 0 to 20H with a delay of 100ms between each count. After the count 20H, the counter should reset itself and repeat the sequence. Use register pair DE as a delay register. Draw a flowchart and show your calculation to set up the 100ms delay.

1. Draw neatly the pin diagram and architectural representation of multipurpose programmable device 8155. Describe about its control logics, I/O parts. How can you determine the addresses for I/O parts, timer and control register?



Pin diagram of 8155 processor (Intel corporation)

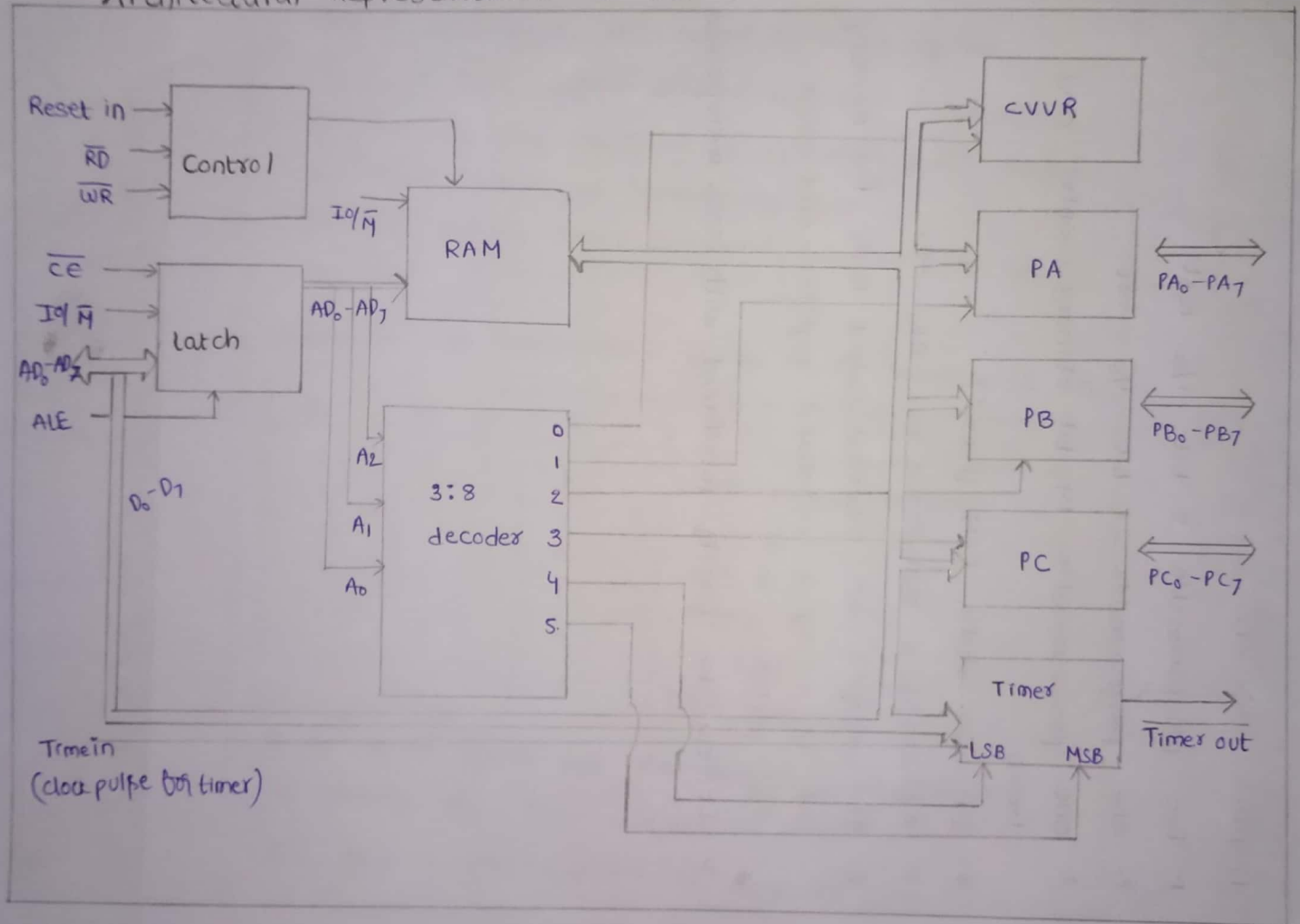
1.  $AD_0 - AD_7$  : Address/data bus multiplexed
2. RESET : RESET Input
3.  $\overline{CE}$  : chip Enable
4. ALE : Arithmetic latch Enable
5.  $\overline{RD}$  : Read input
6.  $\overline{WR}$  : write input
7.  $I/O/\overline{M}$  : Input / memory / output
8.  $PA_0 - PA_7$  : Port A
9.  $PB_0 - PB_7$  : Port B
10.  $PC_0 - PC_7$  : Port C
11. TIMER-IN : Timer input signal
12.  $\overline{TIMER-OUT}$  : Timer output

- This section has  $256 \times 8$  Memory. It means 256 locations of 8 bit each.
- 8155 is a multifunction/Multipurpose programmable device
- It contains RAM, I/O ports and Timer.

### Features of 8155 :

1. Two programmable 8 bit i/o port
2. one programmable 6 bit i/o port
3. one programmable 14 bit binary counter and timer.
4. 2K bits static RAM (256x8)
5. Address/Data Multiplexed bus  $AD_0 - AD_7$
6. It contains an internal select logic for memory and i/o using a command register and two i/o ports.
7. It can be easily interfaced with 8085 microprocessor.

## Architectural Representation of 8155 :



D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
TimerCommand		IEB	IEA	PC(ALT <sub>1</sub> and ALT <sub>0</sub> )		P <sub>B</sub>	P <sub>A</sub>

Control register is also called Command register

- It is a 8-bit which defines different functions.
- It defines the function of different ports such as input, output or handshaking mode.
- It also defines timer command for timer to start and stop.

Based on the combinations of A<sub>0</sub>, A<sub>1</sub>, A<sub>2</sub> selection is done.

A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	
0	0	0	Control/status register
0	0	1	Port A
0	1	0	Port B
0	1	1	Port C
1	0	0	LSB Timer
1	0	1	MSB Timer

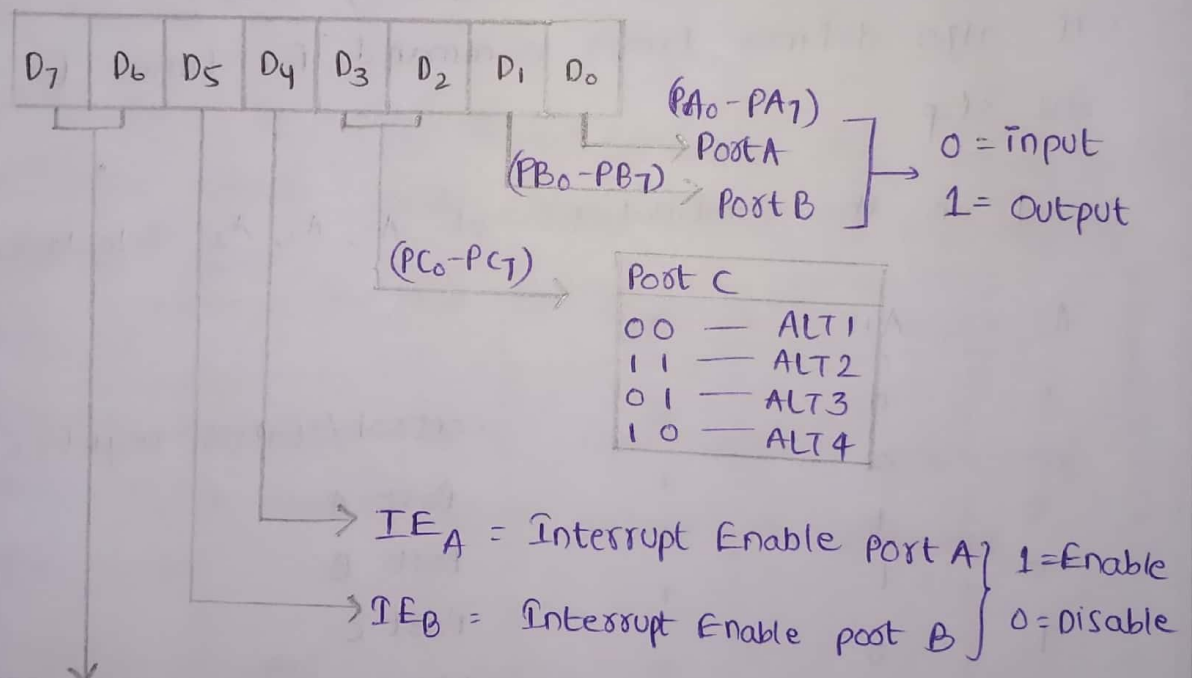
It is to be noted that the control/status register is having the same address (ie. 20H)

But the control register is accessed with  $\overline{WR}=0, \overline{RD}=1$ .  
For status register access  $\overline{WR}=1$  and  $\overline{RD}=0$ .

The control register can never be read.



A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Address	Register/Port/Timer
0	0	1	0	0	0	0	0	20H	Control/status register
0	0	1	0	0	0	0	1	21H	Port A
0	0	1	0	0	0	1	0	22H	Port B
0	0	1	0	0	0	1	1	23H	Port C
0	0	1	0	0	1	0	0	24H	LSB Timer
0	0	1	0	0	1	0	1	25H	MSB Timer



### Timer Commands

- 00 = NOP → No effect on timer. i.e. no effect on counter
- 01 = Stop → Stop counting if timer is running otherwise no effect on timer.
- 10 = Stop after terminal count, stop after end of count if timer is running else no affect
- 11 = Start - start timer if it is not running if timer is running - stop at end of count Reload new mode and count and start again

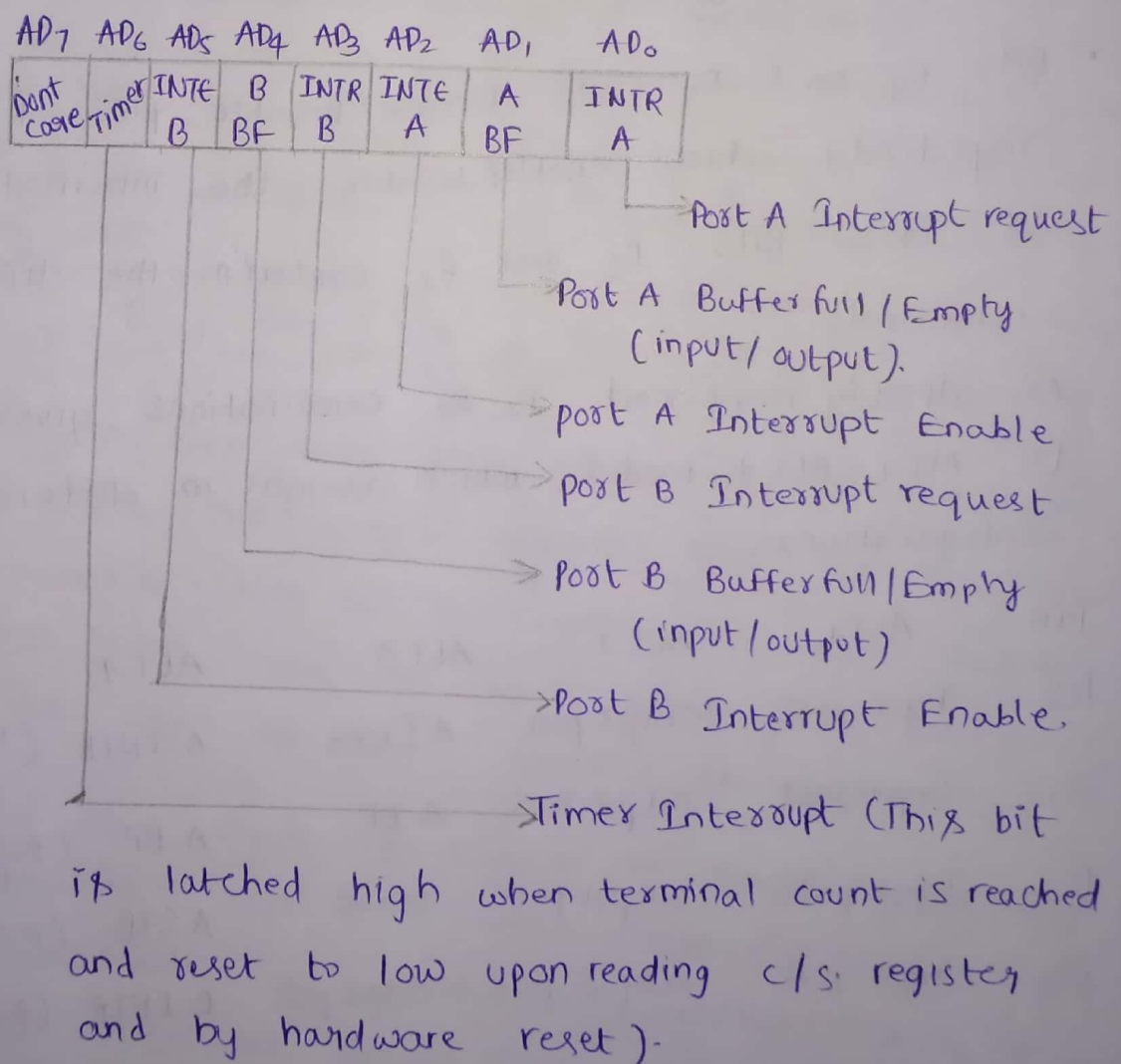
- The control register contains eight bits
- The content of the lower 2 bits,  $D_1 - D_0$  configure ports A and B as input/output.
- Bits  $D_2$  and  $D_3$  configure bits  $PC_0 - PC_5$  of port C (Port C is a 6-bit while port A and port B are of (eight) 8-bit) and can have four combinations ALT 1, ALT 2, ALT 3, ALT 4 depending on combinations of  $D_2$  and  $D_3$ .
- Bits  $D_4$  and  $D_5$  are Enable/Disable pins for A and B respectively which enable/disable the internal flip flop of 8155. Bits  $D_6$  and  $D_7$  contain the timer commands.

As already mentioned  $D_2, D_3$  combinations gives rise to ALT 1 - ALT 4 modes, which assigns in different configurations.

Pin	ALT 1	ALT 2	ALT 3	ALT 4
$PC_0$	Input port	Output port	A INTR	A INTR (Port A Interrupt)
$PC_1$	Input port	output port	A BF	A BF (Port A Buffer Full)
$PC_2$	Input port	output port	A STB	A STB (Port A Strobe)
$PC_3$	Input port	output port	output port	B INTR (Port B Interrupt)
$PC_4$	Input port	output port	output port	B BF (Port B Buffer Full)
$PC_5$	Input port	output port	output port	B STB (Port B Strobe)



- ALT1 and ALT2 correspond to simple input/output of port c respectively.
- In ALT3 mode PC<sub>0</sub>-PC<sub>2</sub> bits are used as control signals for port A, while pins PC<sub>3</sub>-PC<sub>5</sub> act as output pins.
- In ALT4 mode PC<sub>0</sub>-PC<sub>2</sub> bits are used as control signals for port A, while PC<sub>3</sub>-PC<sub>5</sub> bits are used as control signals for port B.



- It has seven bits. Bit D<sub>7</sub> is don't care bit. Bit D<sub>6</sub> contain the status of timer. Bits D<sub>5</sub>-D<sub>3</sub> concern to status of port B while bits D<sub>2</sub>-D<sub>0</sub> of port A.

### The 8155 timer :

Consists of two 8-bit registers

1. 8-bit LSB and 8-bit MSB
2. In these 16 bits, 14 bits are used for counter and two bits ( $M_2$  and  $M_1$ ) for mode selection.
3. The counter is 14-bit down counter. It can operate in four different modes of operation.

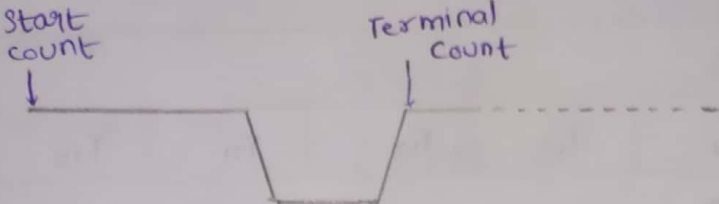



Timer MSB							
$M_2$	$M_1$	$T_{13}$	$T_{12}$	$T_{11}$	$T_{10}$	$T_9$	$T_8$
Timer LSB							
$T_7$	$T_6$	$T_5$	$T_4$	$T_3$	$T_2$	$T_1$	$T_0$

Mode 0: In this mode, timer gives only one cycle of Square wave, the output remains high for  $\frac{1}{2}$  count and remain low for  $\frac{1}{2}$  count. wave width depends on two factors: one is input clock pulse frequency, and the other is count loaded in counter.

Mode 1: This mode is similar to single square wave in operation but the when counter becomes zero, the count value is automatically reloaded. Thus it provide continuous square wave.

Mode 2: This mode gives a single clock pulse as a output of the end of count. The output is high normally, but it becomes low for 1 clock pulse and again it will become high and remain high.

Mode 4 : This mode is similar to mode 2 but when the counter becomes zero the count value is automatically reloaded. Thus it provide continuous pulse.

Mode bits		Timer out waveforms	
M <sub>2</sub>	M <sub>1</sub>		
0	0		Single square wave
0	1		Continuous square wave
1	0		Single pulse on terminal count
1	1		continuous pulses.

Timer modes and output.

2. Illustrate the interfacing of seven segment LED output ports using 8155.

Port Address

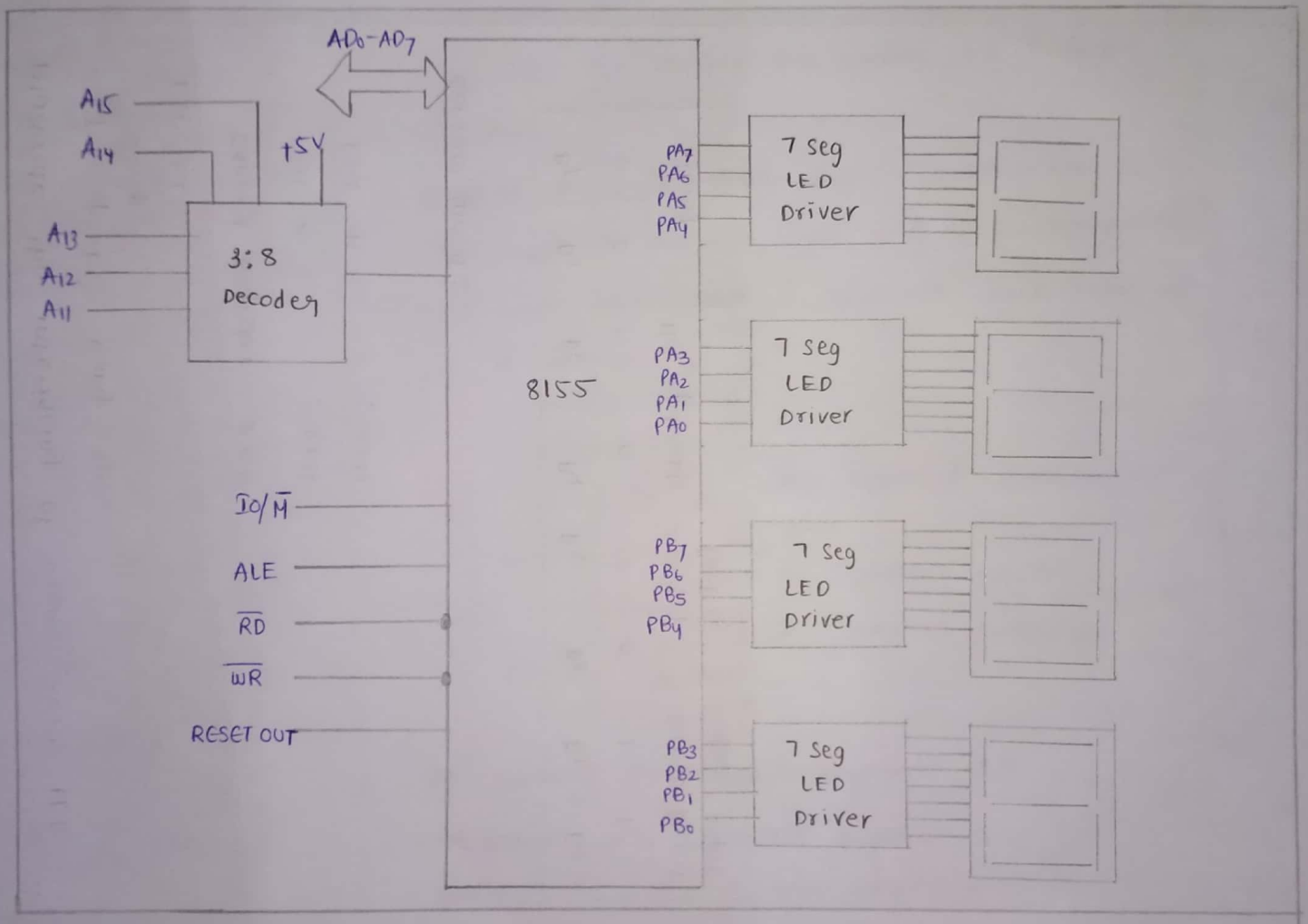
Control Register : 20H

Port A : 21H

Port B : 22H

Control word :

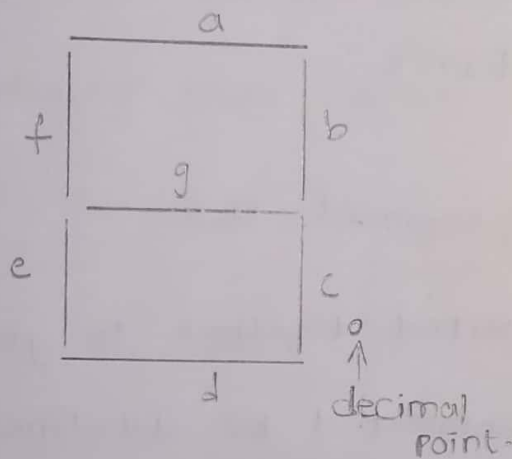
D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	0	0	0	0	0	1	1
Timer.		Not applicable		Use for Port C		Port B Output	Port A output





## Seven-segment LEDs

- often used to display BCD numbers and a few alphabets.
- A group of eight LEDs physically mounted in the shape of number eight and a decimal point.
- Each LED is called a Segment and labelled as 'a' through 'g'.

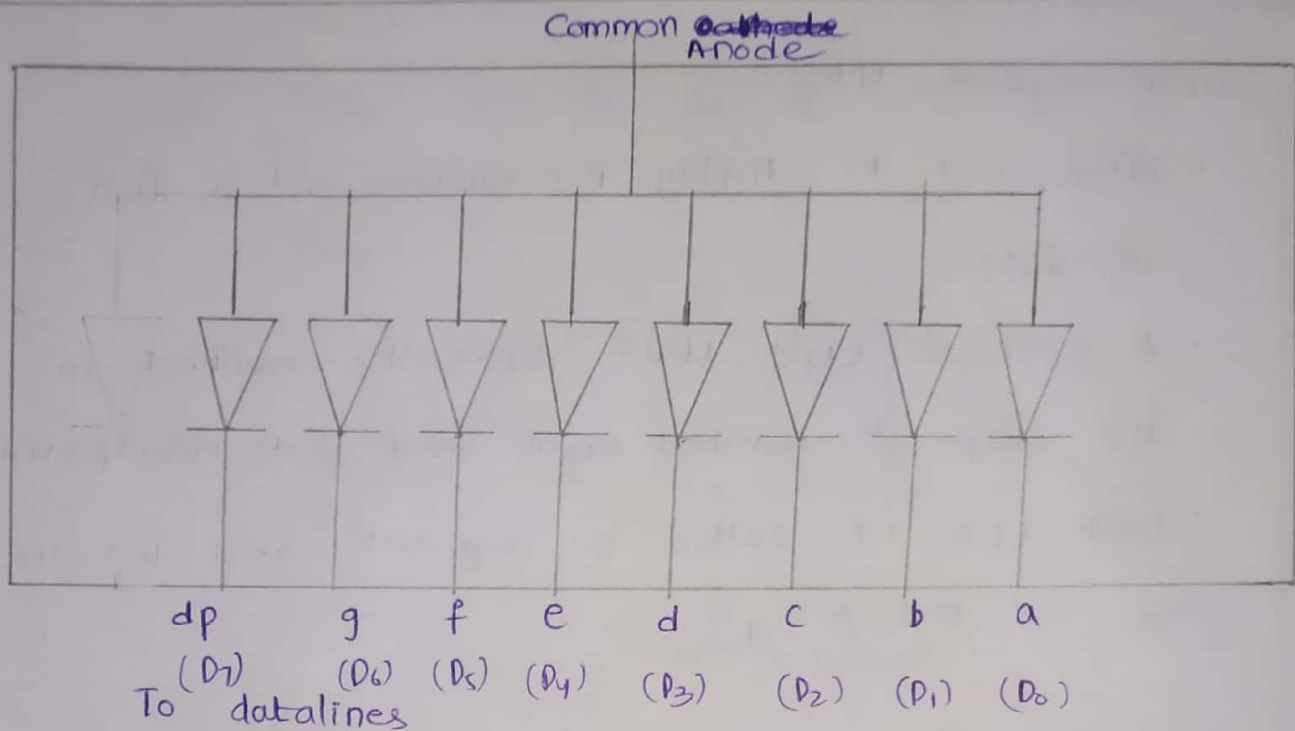


Two types of seven-segment LEDs

1. Common anode
2. Common cathode.

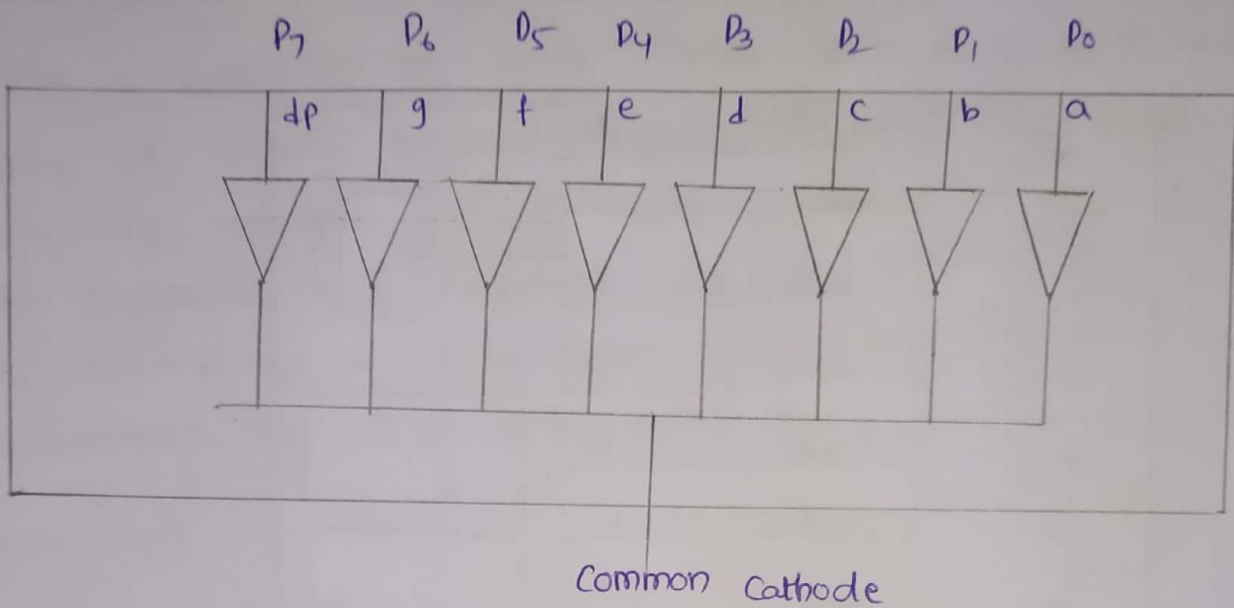
In common anode seven segment LED

- All anodes are connected together to a power supply and cathodes are connected to datalines.
- logic 0 turns on a segment.
- Example: To display digit 1, all segments except b and c should be off.
- Byte 1111 001 = F9H will display digit 1.



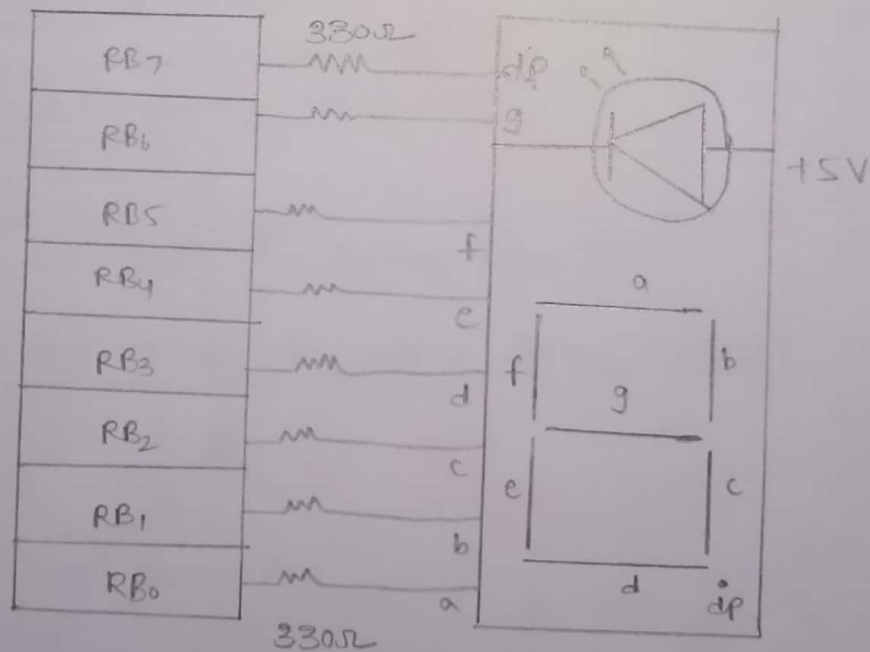
In common ~~cathode~~ seven segment LEDs.

- All cathodes are connected together to ground and the anodes are connected to datalines.
- Logic 1 turns on the segment.
- Example: To display digit 1, all segments except b and c should be off.
- Byte 00000110 = 06H will display digit 1.



From data lines through an interfacing device.

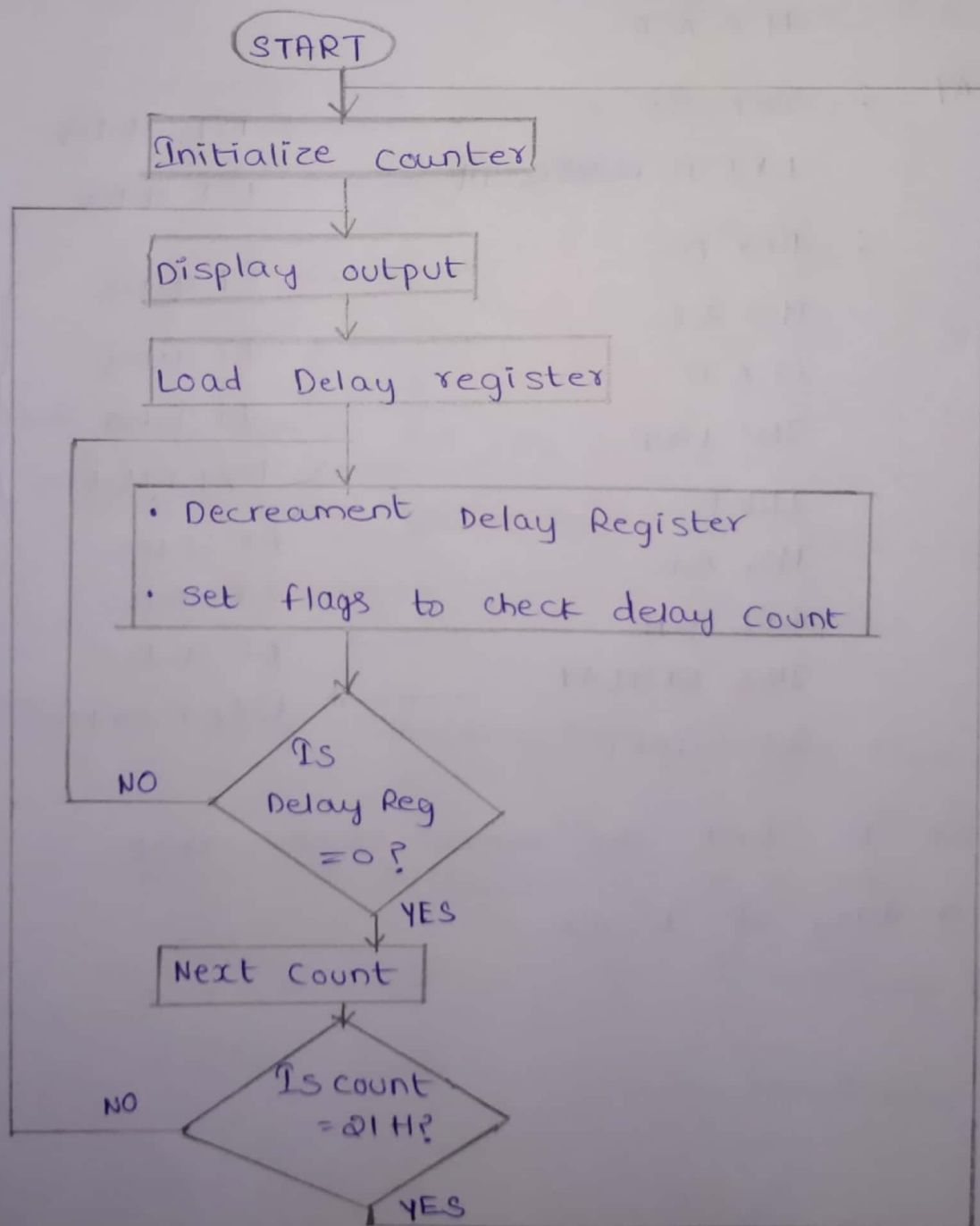
Interfacing Seven - Segment LEDs to PORT B and PORT C





3. write a program to count from 0 to 20 H with the delay of 100ms between each count. After the count 20 H, the counter should reset itself and repeat the sequence. Use register pair DE as a delay register. Draw a flow chart and show your calculation to setup the 100ms delay.

Flow chart :





Time delay = 100 ms

A 16-bit register pair (DE) is used as delay

Initial number = Count

Sample program:

#ORG 2000H

START : MVI B, 00H

MOV A, B

DISPLAY : OUT 00 → 10T states

LXI D, ~~0000~~ Count → 10T states

LOOP : DCX D → 6T states

MOV A, E → 4T states

ORA D → 4T states

JNZ LOOP → 10/7 T states. }

INR B → 4T states

MOV A, B → 4T states

CPI 15H → 7T states

JNZ DISPLAY → 10/7 T states.

JZ START

we have to find the number count that  
cause a delay of 100ms.

## Time delay Calculations

Time delay inside the loop :

$$\begin{aligned}\text{No. of T states inside loop} &= (6+4+4+10) \text{ T states} \\ &= 24 \text{ T}\end{aligned}$$

This loop runs 'count' no. of times

$$\Rightarrow (24 \times \text{Count}) \text{ T states}$$

Assuming clock frequency = 2 MHz .

$$f_{in} = 1 \text{ MHz}$$

$$T = \frac{1}{f_{in}} \times 10^{-6} \text{ sec}$$

$$T \Rightarrow \underline{10^{-6} \text{ sec}}$$

Time delay outside the loop :

$$\begin{aligned}\text{No. of T states outside loop} &= (10+10+4+4+7+10) \text{ T states} \\ &= 45 \text{ T states.}\end{aligned}$$

(100 ms)

total time delay = Time delay inside loop + Time delay outside loop

$$100 \times 10^{-3} \approx 24 \times \text{Count} \times 10^{-6} + 45 \times 10^{-6}$$

$$(\text{Count})_{10} = 4164 \quad \therefore \text{This is indecimal form.}$$

$$(\text{Count})_{16} = 1044 \quad \therefore \text{This is hexadecimal form.}$$

Program :

Address	Label	Mnemonics	HexCode	Comments
2000	START	MVI B, 00	06	// Initialize counter
2001				
2002		MOV A, B	78	// Moving to accumulator for output
2003	DISPLAY	OUT 00	D3	// Display at port 00
2004			00	
2005		LXI D, 1044H	11	// Load DE pair with (1044) <sub>16</sub> .
2006			45	// for delay
2007			10	
2008	LOOP	DCX D	18	// Decrement the content by 1
2009		MOV A, E	7B	// Move register content of E to A
200A		ORA D	B2	// Logical OR of contents A and E
200B		JNZ LOOP	C2	// to check for zeroes
200C			08	// Jump if not zero.
200D			20	
200E		INR B	04	// Increment [B] for next number
200F		MOV A, B	78	// [B] → [A]
2010		CPI 15H	FE	// Compare with (01) <sub>10</sub>
2011			15	

2012	JNZ DISPLAY	C2	Jump to display if $[A] \neq (01)_{10}$ .
2013		03	
2014		20	
2015	JZ START	CA	If $[A] = (01)_{10}$ Again start the
2016		00	Counting process.
2017		20	

NOTE: This is (infinite) continuous process counts from 0 to 20 again resets and continues. So the output is shown at an instant. The process continues as long as we stop execution manually.

## 8085 Assembly Language Editor

Assembler Disassembler

#ORG 2000H

```

START:  MVI B,00H
        MOV A,B
DISPLAY: OVI 00
        EXI D,1044H
LOOP:   DCX D
        MOV A,L
        ORA B
        JNZ LOOP
        INR B
        MOV A,B
        CPI 15H
        JNZ DISPLAY
        SJMP START

```

Autocorrect

Assemble

Registers Memory Devices

## Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	54	0	1	0	1	1	1	1	1
Register B	12	0	0	0	1	0	0	1	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	1	0	0	0
Register E	57	0	1	0	1	0	1	1	1
Register H	00	0	0	0	0	0	0	0	0
Register I	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	04	0	0	0	0	0	1	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer(HL)	0000
Program Status Word(PSW)	5F04
Program Counter(PC)	2000
Clock Cycle Counter	1040321
Instruction Counter	300035

SOD	SID	INTR	TRAP	R7.5	R6.5	R5.5
0	0	0	0	0	0	0

For SIM instruction

SOD	SDE	*	R7.5	MSE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

For RIM instruction

SID	R7.5	R6.5	R5.5	IE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

No. Converter Tool :

Hexadecimal	Decimal	Binary
0	0	0



## Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 2000	START	MVI B,00	06	2	2	7
2001			00			
✓ 2002		MOV A,B	78	1	1	4
✓ 2003	DISPL...	OUT 00	03	2	3	10
2004			00			
✓ 2005		LXI D,1011	11	3	3	10
2006			44			
2007			10			
✓ 2008	LOOP	DCX D	18	1	1	6
✓ 2009		MOV A,B	78	1	1	4
✓ 200A		ORA D	82	1	1	4
✓ 200B		JNZ LOOP	C2	3	3	10
200C			00			
200D			30			
✓ 200E		INR B	04	1	1	4
✓ 200F		MOV A,B	78	1	1	4
✓ 2010		CPI 15	1E	2	2	7
2011			15			
✓ 2012		JNZ DISPLAY	C2	3	3	10

## Simulate

Start From → 2000

Run all At a Time

Step By Step

## Registers

Register	Value	7	6	5	4	3	2	1	0
Accumulator	58	0	1	0	1	1	1	1	1
Register B	12	0	0	0	1	0	0	1	0
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	1	0	0	0
Register E	57	0	1	0	1	0	1	1	1
Register H	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	04	0	0	0	0	0	1	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	5F04
Program Counter(PC)	2000
Clock Cycle Counter	1040721
Instruction Counter	300035

SOD	SID	INTR	TRAP	R7.5	R6.5	R5.5
0	0	0	0	0	0	0

For SIM instruction	SOD	SDE	*	R7.5	MSE	M7.5	M6.5	M5.5
	0	0	0	0	0	0	0	0

For RIM instruction	SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
	0	0	0	0	0	0	0	0

## No. Converter Tool

Hexadecimal	Decimal	Binary
0	0	0

Editor Assembler

Assembler

Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 2000	START	MVI B,00	06	2	2	7
2001			00			
✓ 2002		MOV A,B	7B	1	1	4
✓ 2003	DISPL...	OUT 00	03	2	3	10
2004			00			
✓ 2005		LXI D,10FF	11	3	3	10
2006			44			
2007			10			
✓ 2008	LOOP	DCX B	1B	1	1	6
✓ 2009		MOV A,B	7B	1	1	4
✓ 200A		ORA B	53	1	1	4
✓ 200B		JNZ LOOP	C2	3	3	10
200C			00			
200D			20			
✓ 200E		INR B	04	1	1	4
✓ 200F		MOV A,B	7B	1	1	4
✓ 2010		CPI 15	FE	2	2	7
2011			15			
✓ 2012		INZ DISPLAY	C2	3	3	10

Simulate

Start From →

2000

Run all At a Time

Step By Step

Registers Memory Devices

Memory Editor

Memory Range: 0000 — FFFF

Memory Address	Value
2000	06
2002	7B
2003	03
2005	11
2006	44
2007	10
2008	1B
2009	7B
200A	53
200B	C2
200C	00
200D	20
200E	04
200F	7B
2010	FE
2011	15
2012	C2
2013	03
2014	20
2015	CA
2017	20

- ☐ Show entire memory content  
☒ Show only loaded memory location  
☐ Store directly to specified memory location

Editor Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 2000	START	MVI B,00	06	2	2	7
2001			00			
✓ 2002		MOV A,B	78	1	1	4
✓ 2003	DISPL...	OUT 00	03	2	3	10
2004			00			
✓ 2005		LXI D,1044	11	3	3	10
2006			44			
2007			10			
✓ 2008	LOOP	DCR D	1B	1	1	6
✓ 2009		MOV A,E	7B	1	1	4
✓ 200A		ORA D	82	1	1	4
✓ 200B		JNZ LOOP	C2	3	3	10
200C			0B			
200D			20			
✓ 200E		INR B	04	1	1	4
✓ 200F		MOV A,B	70	1	1	4
✓ 2010		CPI 15	FE	2	2	7
2011			15			
✓ 2012		JNZ DISPLAY	C2	3	3	10

Simulate

Start From --

2000

Run all At a Time

Step By Step

Registers Memory Devices

Interfacing device

I/O Port Editor

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	12	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00