Counters & Time Delays:

→ Counters and time delays are important techniques used in applications such as traffic signals, digital clocks, process control and serial data transfer

→ Counters are used primarily for keeping track of events.

→ Time delays are important in setting up reasonably accurate timing between two events.

→ A counter is designed by simply loading an appropriate number into one of the registers and using INR or DCR instructions.

A loop is established to update the count and each count is checked to determine whether it has reached the final number; if not the loop is repeated.

Scanned with CamScanner

Time delay :

(1) Time delay using One Register

→ A count is loaded in a register

→ A loop is executed until the count reaches zero.

Ex:

			T-states
[redacted]	MVI	C, FFH	7
Loop :	DCR	C	4
	JNZ	Loop	10/7

Calculations:

(1) Time to execute MVI instruction.

clock freq $= f = 2\,MHz$

clock period $T = \frac{1}{f} = 0.5\,\mu s$

$\therefore$ Time to execute MVI $= 7T \times 0.5\,\mu s$

$= 3.5\,\mu s$

(2) The next two instructions DCR, JNZ form a loop of 14 (4 + 10) T-states. The loop is repeated 255 times.

$\therefore T_L = (T \times \text{loop T-states} \times N_{10})$

$= (0.5\,\mu)(14)(255) = 1785\,\mu s \cong 1.8\,ms$

Scanned with CamScanner

The T-states for JNZ are shown as 10/7

8085 requires 10-T-states to execute a conditional jump when it jumps or changes the sequence of program and seven T-states when the program falls through the loop (goes to the instruction following JNZ).

→ In the above case, the loop is executed 255 times; in the last cycle, JNZ will be executed in 7 T-states.

∴ Adjusted loop delay

$$T_{LA} = T_L - (3 \text{ T-states} \times \text{Clock period})$$

$$= 1785.0\mu s - (3 \times 0.5\mu s)$$

$$= 1785.0\mu s - 1.5\mu s = 1783.5\mu s$$

∴ Total Time delay = Time to execute instructions outside loop + Time to execute loop instructions

$$\therefore T_D = T_0 + T_{LA}$$

$$= (7 \times 0.5\mu s) + (1783.5\mu s)$$

$$= 1787\mu s$$

$$\simeq 1.8 ms$$

Note :- The difference between loop delay T_L and Total delay T_D is only 2 μs and can be ignored in most cases.

Scanned with CamScanner

→ The time delay can be varied by changing the count FFH ; however, to increase the time delay beyond 1.8ms in a 2MHz system, a register pair or a loop within a loop technique is to be used.

(2) Time Delay using a Register pair :

Ex:

			T-states
	LXI	B, 2384H	10
loop:	DCX	B	6
	MOV	A, C	4
	ORA	B	4
	JNZ	loop	10/7

Discussion : DCX instruction does not set Z flag. Hence we are using MOV A, C and ORA B to check whether BC register pair is zero or not.

∴ the loop is repeated 2384H no. of times.

Scanned with CamScanner

$2384H = 2 \times 16^3 + 3 \times 16^2 + 8 \times 16^1 + 4 \times 4^0$

$= (9092)_{10}$.

$\therefore$ For $f = 2MHz$ $\quad T = \frac{1}{f} = 0.5\mu s$

The delay in the loop is calculated as

$T_L = 0.5\mu \times 24$ T-states $\times$ Loop count

$= 0.5\mu s \times 24 \times 9092$

$\simeq 109$ ms (without adjusting last cycle).

Total delay $= 109ms + T_0$

$= 109ms + (10 \times 0.5\mu s)$

$= 109ms + 5\mu s$

$\simeq 109ms$. (The instruction LXI adds only $5\mu s$)

Scanned with CamScanner

(3) Time delay using a loop within a loop Technique

```
                        MVI  B, 38H      7T
Loop2:                  MVI  C, FFH      7T
Loop1:                  DCR  C           4T
                        JNZ  Loop1       10/7T
                        DCR  B           4T
                        JNZ  Loop2       10/7T
```

Discussion:

Loop 1 executes 255 times, every loop involves 14 T-states.

∴ time for executing inner loop (Loop1): 1783.5 μs.

The execution time of outer loop will be

$$T_{L2} = 56\,(T_{L1} + 21 \text{ T-states} \times 0.5\,\mu s)$$
$$= 56\,(1783.5\,\mu s + 10.5\,\mu s)$$
$$\simeq 100.46 \text{ ms}$$

Scanned with CamScanner

Note: The time delay within a loop can be further increased by using instructions that will not affect the program except to increase the delay. Example: NOP can add four T-states in the delay loop.

Counter Design with Time Delay:

Illustrative program: Hexadecimal Counter

Write a program to count continuously in hexadecimal from FFH to 00H in a system with 0.5 μs clock period. Use Reg. C to set up a 1ms delay between each count and display the no. at one of the output ports.

Scanned with CamScanner

```
        MVI   B, 00H
NEXT:   DCR   B
        MVI   C, COUNT
DELAY:  DCR   C
        JNZ   DELAY
        MOV   A, B
        OUT   PORT #
        JMP   NEXT
```

Delay calculations:

→ Delay is accomplished with the inner loop. It involves DCR C, JNZ instructions taking 14 T states.

$\therefore T_L = (14\ T\text{-states} \times 0.5\ \mu s) \times \text{count}$

$= (7\ \mu s) \times \text{COUNT}$

Scanned with CamScanner

The delay outside the loop includes

DCR B	4T
MVI C,count	7T
MOV A,B	4T
OUT PORT	10T
JMP	10T
	35T.

$\therefore$ delay outside loop

$T_0 = 35T \times T$

$= 35 \times 0.5\mu s$

$= 17.5\mu s.$

$\therefore$ Total Time delay $T_D = T_0 + T_L$

$= 17.5\mu s + 7\mu s \times count$

$\therefore 1ms = 17.5\mu + 7\mu \times COUNT$

$\therefore COUNT = \dfrac{1ms - 17.5\mu s}{7\mu s} \simeq 140_{10}.$

$\therefore$ delay count $140_{10} = (8C)_H$

Scanned with CamScanner

Illustrative Program: Zero-To-Nine (Modulo-10) counter

Problem Statement:

Write a program to count from 0 to 9 with a one-second delay between each count. At the count of 9, the counter should reset itself to 0 and repeat the sequence continuously. Use the register pair H L to set up the delay, and display the count at one of the output ports. Assume clock freq is 1MHz.

Sol

		T-states	
START:	MVI B, 00H		
DISPLAY:	OUT PORT#	10	T_D
	LXI H, 16bit	10	
LOOP:	DCX H	6	T_L: 24-Tstates
	MOV A, L	4	
	ORA H	4	
	JNZ LOOP	10/7	
	INR B	4	T_D
	MOV A, B	4	
	CPI 0A H	7	
	JNZ DISPLAY	10/7	
	JZ START		

Scanned with CamScanner

Delay calculation:

Loop delay T_L = 24 T. states × T × count

1 second = 24 × 1 μ × count

∴ count = $\frac{1}{24\ \mu s}$ = 41666

∴ count = $(41666)_{10}$ = $(A2C2)_H$.

OR

∴ Total delay = $T_0 + T_L$

= (45 T. × 1 μs) + (24 T. × T × count)

1 second = 45 × 1 μs + 24 × 1 μs × count

∴ count = $\frac{1 - 45\ \mu s}{24\ \mu s}$

count ≅ 41665

∴ The difference between two delay counts calculated above is of very little significance in many applications.

Scanned with CamScanner