Time Delay and Counter

Lecture objectives: at the end of this lecture the student will able to:

1- **Define the time delay.**

2- **Study types of time delay.**

3- **Design all types of counters.**

6.1 Time Delay:

6.1.1 Definition of time delay: it is number of instructions that written to keep a track for certain interval. Time delay or (software delay) can be designed through executing group of instruction number of times. Flow chart as an example of time delay is shown in Fig. (6-1) below.

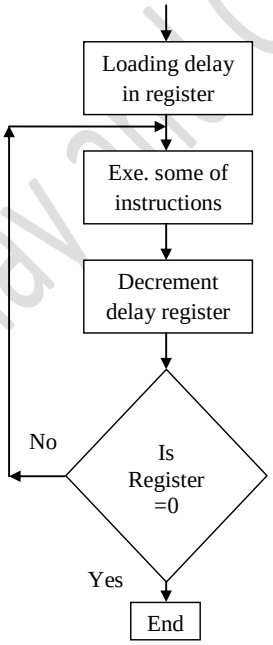

Figure (6-1): Example of time delay.

6.1.2 Types of time delay: there are three types of time delay as shown below:

A. Time Delay using NOP instruction: NOP instruction does nothing but take 4 T-states of processor time to execute. So by executing NOP instruction between two instructions we can get delay of 4 T-state where:

$$1\,T_{state} = \frac{1}{operating\ frequency\ of\ 8085\ processor} \qquad 6.1$$

B. Time Delay Using Counter: Time delay can be created using counting process which means executing number of instructions many times where the initial value of counter required to get specific time delay can be determined. there are two types of delay using counter as below:

B.1 time delay using one register (8-bit counter): in this type of time delay the register delay is one register loaded with 8-bit number in one loop as shown in program 6.1below :

Program 6.1

```
        MVI C,37 (delay reg.)
LOOP1   MVI A, 33
        RAR (some ins.)
        DCR C (decrement delay reg.)
        JNZ  LOOP1 (condition)
        HLT
```

B. 2 time delay using register pair: in this type of time delay the register delay is register pair loaded with 16-bit number in one loop as shown in program 6.2 below:

Program 6.2

```
        LXI B,234B          (delay reg.)
LOOP1   MVI A, 33
        RAR                 (some ins.)
        DCX B       (decrement delay reg.)
        MOV A,C
        ORA B
        JNZ  LOOP1          (condition)
        HLT
```

C. time delay using loop with in loop: this time delay is used two loop one internal and the other loop is external, these two loops can be designed by using one register or register pair as shown in program 6.3 below:

Program 6.3

```
        MVI B,37      (delay reg. external loop)
LOOP 2  MVI D,FF      (delay reg. internal loop)
LOOP1   MVI A, 33
        RAR           (some ins.)
        DCR D         (decrement delay reg. int. loop)
        JNZ           LOOP1 (condition)
        DCR B         (decrement delay reg. ext. loop)
        JNZ           LOOP2 (condition)
        HLT
```

6.1.3 Calculation of time delay: the interval of any program can be calculated by using the equation below:

$$T_t = T_o + T_i \quad (6.2)$$

Where T_t is total time interval. T_o is the out loop instructions time. T_i is in loop instructions time.

$$T_o = T_{state}\,(total) * t \quad (6.3)$$

$$T_i = T_{state}\,(total) * t * N_{10} \quad (6.4)$$

Where N_{10} is the number that loaded in delay register in decimal. t is the processor time clock.

Example 6.1: Calculate the time delay to programs (6.4, 6.5 and 6.6) below, (let the microprocessor frequency is 1MHz)

Solution:

First t=1/F=1/1*10^6 =1µS

Program 6.4

```
        MVI C,37      (delay reg.)        7Tstate
LOOP1   MVI A, 33                         7Tstate
        RAR           (some ins.)         4Tstate
        DCR C         (decr. delay reg.)  4Tstate
        JNZ LOOP1     (condition)         10/7Tstate
        HLT                               6Tstate
```

$T_t = T_o + T_i$

$T_o = [7T_{state}$ **(MVI C,37)** + $7T_{state}$ **(JNZ loop1)** + $6T_{state}$ **(HLT)]***t

$=20\ T_{state}*1\ \mu S = 20\ \mu S$

$T_i = [7T_{state}$ **(MVI A,33)** + $4T_{state}$ **(RAR)** + $4T_{state}$ **(DCR C)** + $10T_{state}$ **(JNZ loop1)]*t***55

$=[25T_{state}]*t*55$

$=[25\ \mu S]*55=1375\ \mu S$

T_t=20 µS+1375 µS=1395 µS =1.395 mS

Program 6.5

	LXI B,234B	**(delay reg.)**	**$10T_{state}$**
LOOP1	**MVI A, 33**		**$7T_{state}$**
	RAR	**(some ins.)**	**$6T_{state}$**
	DCX B	**(decr. delay reg.)**	**$4T_{state}$**
	MOV A,C		**$4T_{state}$**
	ORA B		**$4T_{state}$**
	JNZ LOOP1	**(condition)**	**$10/7T_{state}$**
	HLT		**$6T_{state}$**

Program 6.6

	MVI B,37	**(delay reg. external loop)**	**$7T_{state}$**
LOOP 2	**MVI D,FF**	**(delay reg. internal loop)**	**$7T_{state}$**
LOOP1	**MVI A, 33**		**$7T_{state}$**
	RAR	***(some ins.)***	**$6T_{state}$**
	DCR D	**(decr. delay reg. int. loop)**	**$4T_{state}$**
	JNZ *LOOP1*		**$10/7T_{state}$**
	DCR B	**(decr. delay reg. ext. loop)**	**$4T_{state}$**
	JNZ *LOOP2*	**(condition)**	**$10/7T_{state}$**
	HLT		**$6T_{state}$**

For program 6.5

T_o=[$10T_{state}$ (**LXI B,234B)** + $7T_{state}$ (**JNZ loop)** + $6T_{state}$ (**HLT)**] = 23 µS

T_i=[$7T_{state}$ (**MVI A,33)** +$4T_{state}$ (**RAR)** + $4T_{state}$ (**DCX B)** + $4T_{state}$ (**MOV A,C)** +$4T_{state}$ (**ORA B)** + $10T_{state}$ (**JNZ loop1)**]*9035 = (33 T_{state})*9035=(33 µS)*9035=298155 µS

T_t=23 µS+298155 µS=298178 µS=298.178ms

For program 6.6

T_o=[$7T_{state}$ (**MVI B,37)** + $7T_{state}$ (**JNZ loop2)** + $6T_{state}$ (**HLT)]**

=20 T_{state}=20 µS

T_i=[$7T_{state}$ (**MVI D,FF)** + ***{****$7T_{state}$ **(MVI A,33)** + $4T_{state}$ **(RAR)** + $4T_{state}$ **(DCR D)} *** *255* + *{*$10T_{state}$ **(JNZ loop1) *** *254}*+ $7T_{state}$ (**JNZ loop1)** +$4T_{state}$ (**DCR B)**] * 55 +$10T_{state}$ (**JNZ loop2)***54

=[$7T_{state}$ (**MVI D,FF)** + {*3825 T_{state}* }+*{2450 T_{state}}*+$7T_{state}$ (**JNZ loop1)** +$4T_{state}$ (**DCR B)**]*55 + 540

= [18 + 3825 + 2450] * 55 + 540

= 346,115 + 540

=346500 T_{state}=346,655 µS

T_t=T_o + T_i =20 µS+346,655 µS=346,675 µS=346.68ms

Comment [DS1]: Internal loop

Comment [DS2]: External loop

6.2 Counters:

6.2.1 Definition of counter: A counter program is an program written to design a counter (all types of counter such as up, down, Johnson, serial, parallel, decimal, etc.) with flexibility in controlling of time interval between counting states.

Example 6.2: write ALP (Assembly Language Program) to design down counter mod (255) and display the counting states on out port with time delay (1ms) for each counting state. Let the processor operate with frequency 2MHz.

Solution: Fig. (6.2) show the flow chat of the counter program

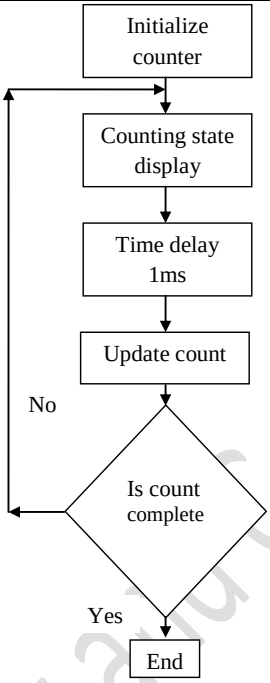

Figure (6.2): flow chart of Ex.6.2

	MVI A,FF	**7T$_{state}$**
A1:	**OUT 02**	**10T$_{state}$**
	MVI D, XX	**7T$_{state}$**
A2:	**DCR D**	**4T$_{state}$**
	JNZ A2(delay)	**10/7T$_{state}$**
	DCR A	**4T$_{state}$**
	JNZ A1(Count display)	**10/7T$_{state}$**
	HLT	**6T$_{state}$**

Now, must loading register **D** with certain number
to get the time delay 1ms, this number calculated as below:

First $T_{state} = 1/F = 1/2*10^6 = 0.5\mu S$

$T_t=T_o + T_i$

Where T_t=1ms

T_o=[10T_{state} (**OUT 02)** + 7T_{state} (**MVI DXX)** + 7T_{state} (**JNZ A2)**+ 4T_{state} (**DCR A)** +10T_{state} (**JNZ A1)]***
0.5µS

=[38 T_{state}] ***** 0.5µS=19 µS

T_i=[(4T_{state} (**DCR D)** * N_{10})+ **(**10T_{state} (**JNZ A2)** * (N_{10}-1))]*0.5 µS

=(2* N_{10})+5* (N_{10}-1)

=(7 N_{10}-5) µS

By return to initial equation

$T_t=T_o + T_i$

1000 µS=19 µS+ (7 N_{10}-5) µS

1000 µS=19 µS+7 N_{10} µS-5 µS

1000 µS=14 µS+7 N_{10} µS

N_{10}=(1000-14)/7=140.857=141

Therefore must loading equivalent to 141 in hexadecimal which equal (8D) in register **D.**

other solution:

Tt = [10T (OUT 02) + 7T (MVI D, XX) + 4T*N_{10} (DCR D) + 10T*N_{10}-1 (JNZ A2) + 7T (JNZ A2) + 4T (DCR A) + 10T (JNZ A1)] * 0.5µS

1000µS =[38T + 4T*N_{10} + 10T* (N_{10}-1)] * 0.5µS

1000µS = 19µS + 2*N_{10} µS+ 5*N_{10} µS- 5µS

1000µS = 14µS + 7*N_{10} µS

N_{10} = (1000 - 14)/7 = 140.857=141

Therefore must loading equivalent to 141 in hexadecimal which equal (8D) in register **D.**

Home work:

1- Calculate the required time to execute the JPE instruction if the condition is satisfying. Let the processor operate with 4 MHz frequency.

2- Write ALP to make the microprocessor working as up/down counter mod (32) with time delay 2ms between each two counting states. (let the frequency is 2 MHz)