Counters & Time Delays
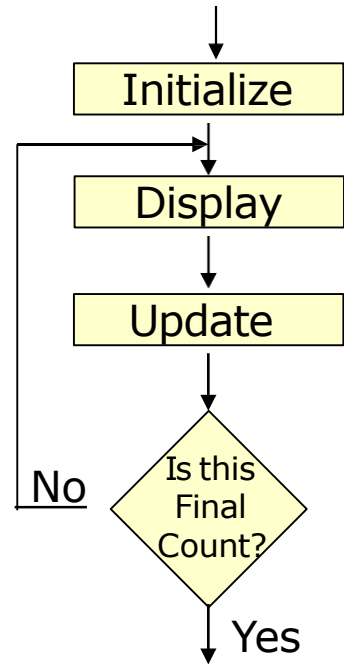
What are Counters & Time Delays

✓ COUNTERS ARE USED TO KEEP TRACK OF EVENTS.

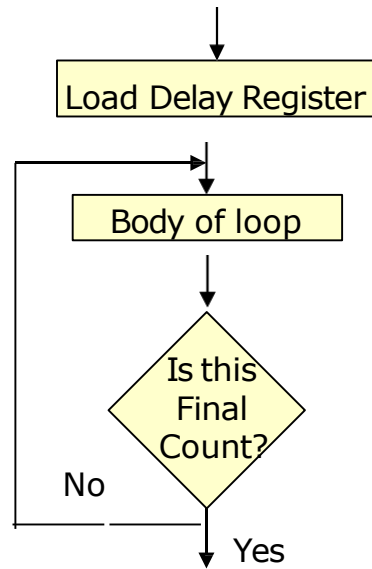✓ TIME DELAYS ARE IMPORTANT IN SETTING UP REASONABLY ACCURATE TIMING BETWEEN TWO EVENTS.

COUNTERS

Flowchart of a Counter

A Counter is designed simply by loading an appropriate number into one of the registers and using the INR (Increment by one) or the DCR (Decrement by one) 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.

TIME   DELAYS

```
        │
        ▼
┌──────────────────────┐
│  Load Delay Register │
└──────────────────────┘
        │
        ▼
┌──────────────────────┐
│     Body of loop     │
└──────────────────────┘
        │
        ▼
       ╱ ╲
      ╱   ╲
     ╱ Is this╲
    ╱  Final   ╲
    ╲  Count?  ╱
     ╲        ╱
  No  ╲      ╱
       ╲   ╱
        ▼
       Yes
```

Flowchart of a Time Delay

▶ The procedure used to design a specific delay is similar to that used to set up a counter. A register is loaded with a number, depending on the time delay required, and then the register is decremented until it   reaches zero by setting up a   loop with a conditional jump instruction. The loop causes the delay, depending upon the   clock period of the system.

CalculatingTime Delays

✓ Each instruction passes through different combinations of Opcode Fetch, Memory Read, and Memory Write cycles.

✓ Knowing the combinations of cycles, one can calculate how long such an instruction would require to complete.

  ✓ Number of Bytes
  ✓ Number of Machine Cycles
  ✓ Number of T-State.

Calculating Time Delays

✓ Knowing how many T-States an instruction requires, and keeping in mind that a T-State is one clock cycle long, we can calculate the time delay using the following formula:

Time Delay = No. of T-States * Clock Period

✓ For example,

"MVI" instruction uses 7 T-States.
Therefore, if the Microprocessor is running at 2 MHz, the instruction would require 3.5 µS to complete.

We can design Time Delay using following three Techniques:

1. Using One Register.
2. Using a Register Pair.
3. Using a Loop with in a Loop

USING ONE REGISTER

✓ A count is loaded in a register, and We can use a loop to produce a certain amount of time delay in a program.

✓ The following is an example of a delay using One Register:

	MVI C, FFH	7 T-States
LOOP	DCR C	4 T-States
	JNZ LOOP	10 T-States

✓ The first instruction initializes the loop counter and is executed only once requiring only 7 T-States.

✓ The following two instructions form a loop that requires 14 T-States to execute and is repeated 255 times until C becomes 0.

USING ONE REGISTER

✓ We need to keep in mind though that in the last iteration of the loop, the JNZ instruction will fail and require only 7 T-States rather than the 10.

✓ Therefore, we must deduct 3 T-States from the total delay to get an accurate delay calculation.

✓ To calculate the delay, we use the following formula:

$$T_{delay} = T_O + T_L$$

T_{delay} = total delay
T_O = delay outside the loop
T_L = delay of the loop

✓ T_O is the sum of all delays outside the loop.

✓ T_L is calculated using the formula

$$T_L = T * \text{Loop T-States} * N \text{ (no. of iterations)}$$

USING ONE REGISTER

✓ Using these formulas, we can calculate the time delay for the previous example:

✓ $T_O = 7$ T-States

 (Delay of the MVI instruction)

✓ $T_L = (14 \times 255) - 3 = 3567$ T-States

 (14 T-States for the 2 instructions repeated 255 times (FF$_{16}$ = 255$_{10}$) reduced by the 3 T-States for the final JNZ.)

✓

$$T_{delay} = [(T_O + T_L)/f]$$
$$= (7 + 3567)/2MHz$$
$$= (3574) \times 0.5 \ \mu Sec$$
$$= 1.787 \ mSec$$

(Assuming **f** = 2 MHz)

Using a Register Pair

✓ Using a single register, one can repeat a loop for a maximum count of 255 times.

✓ It is possible to increase this count by using a register pair for the loop counter instead of the single register.

    ✓ A minor problem arises in how to test for the final count since DCX and INX do not modify the flags.

    ✓ However, if the loop is looking for when the count becomes zero, we can use a small trick by ORing the two registers in the pair and then checking the zero flag.

Using a Register Pair

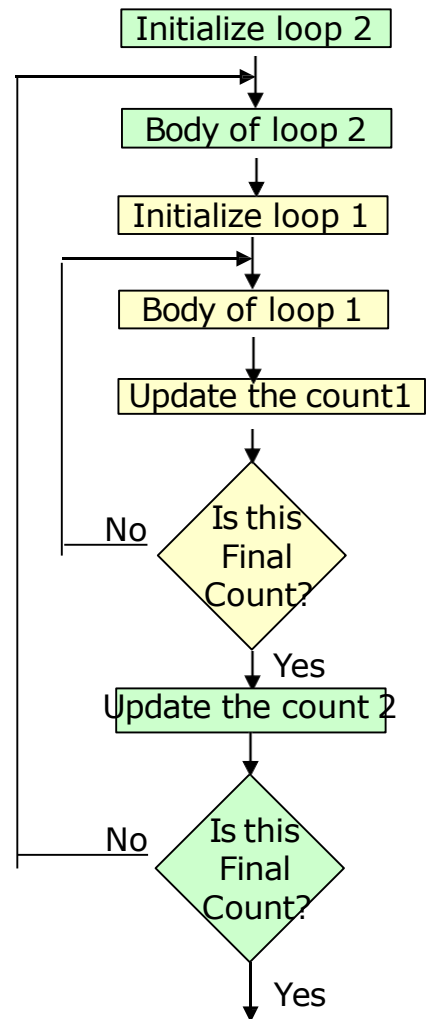✓ The following is an example of a delay loop set up with a register pair as the loop counter.

	LXI B, 1000H	10 T-States
LOOP	DCX B	6 T-States
	MOV A, C	4 T-States
	ORA B	4 T-States
	JNZ LOOP	10 T-States

B=00
C=00

Using a Register Pair

- ✓ Using the same formula from before, we can calculate:

- ✓ T_O = 10 T-States
    (The delay for the LXI instruction)

- ✓ T_L = (24 X 4096) - 3 = 98301 T- States
    (24 T-States for the 4 instructions in the loop repeated 4096 times ($1000_{16} = 4096_{10}$) reduced by the 3 T-States for the JNZ in the last iteration.)

- ✓ T_{Delay} = (10 + 98301) X 0.5 mSec = 49.155 mSec

Using a Loop with in a Loop

Initialize loop 2

Body of loop 2

Initialize loop 1

Body of loop 1

Update the count1

Is this Final Count?

No

Yes

Update the count 2

Is this Final Count?

No

Yes

Flowchart for time delay with two loops

✓ Nested loops can be easily setup in Assembly language by using two registers for the two loop counters and updating the right register in the right loop.

Using a Loop with in a Loop

✓ Instead (or in conjunction with) Register Pairs, a nested loop structure can be used to increase the total delay produced.

	MVI B, 10H	7 T-States
	MVI B, 10H	7 T-States
LOOP2	MVI C, FFH	7 T-States
LOOP1	DCR C	4 T-States
	JNZ LOOP1	10 T-States
	DCR B	4 T-States
	JNZ LOOP2	10 T-States

[{(255*14)-3}+21]*16-3

Using a Loop with in a Loop

✓ The calculation remains the same except that the formula must be applied recursively to each loop.

Start with the inner loop, then plug that delay in the calculation of the outer loop.

✓ Delay of inner loop,

T_{O1} = 7 T-States

(MVI C, FFH instruction)

T_{L1} = (255 X 14) - 3 = 3567 T-States

(14 T-States for the DCR C and JNZ instructions repeated 255 times (FF_{16} = 255_{10}) minus 3 for the final JNZ.)

T_{LOOP1} = 7 + 3567 = 3574 T-States

Using a Loop with in a Loop

✓ Delay of outer loop

$T_{O2} = 7$ T-States

(MVI B, 10H instruction)

$T_{L1} = (16 \times (14 + 3574)) - 3 = 57405$ T-States

(14 T-States for the DCR B and JNZ instructions and 3574 T-States for loop1 repeated 16 times ($10_{16} = 16_{10}$) minus 3 for the final JNZ.)
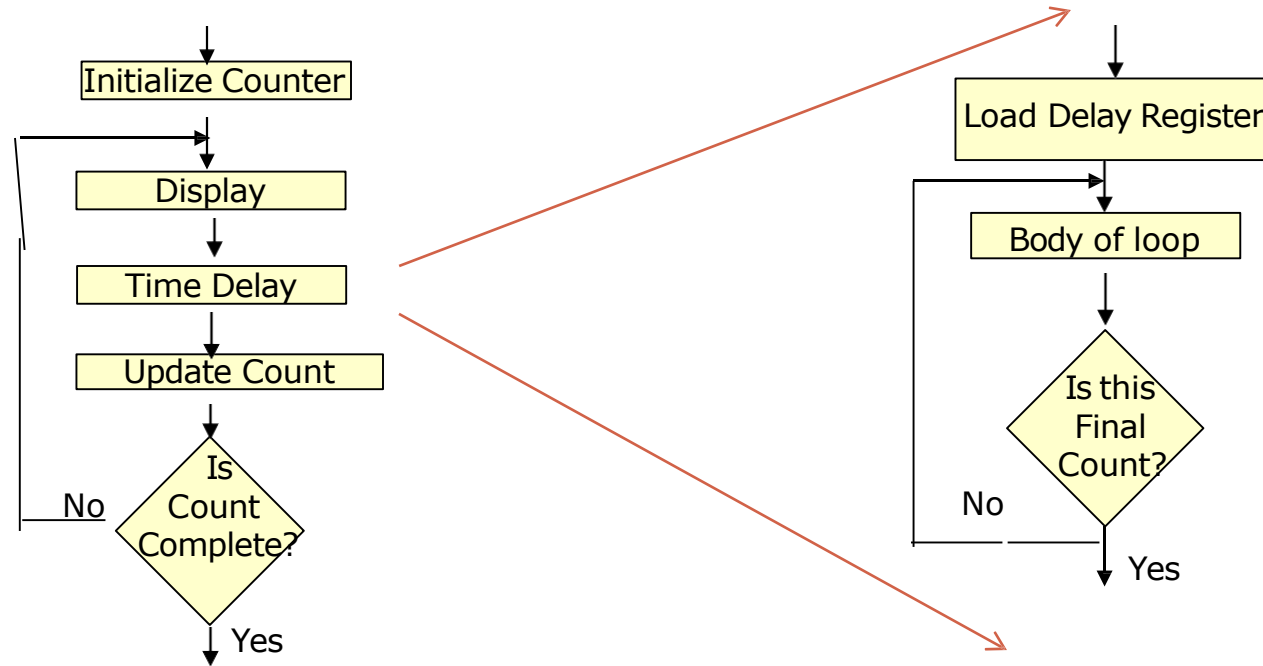
$T_{Delay} = 7 + 57405 = 57412$ T-States

✓ Total Delay

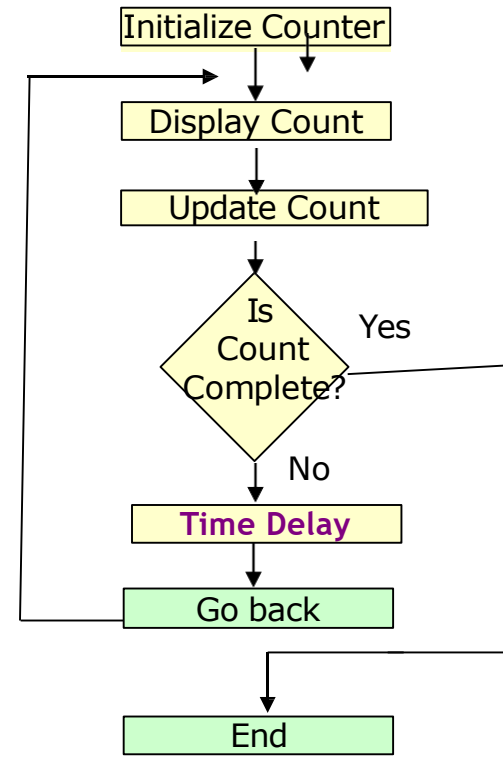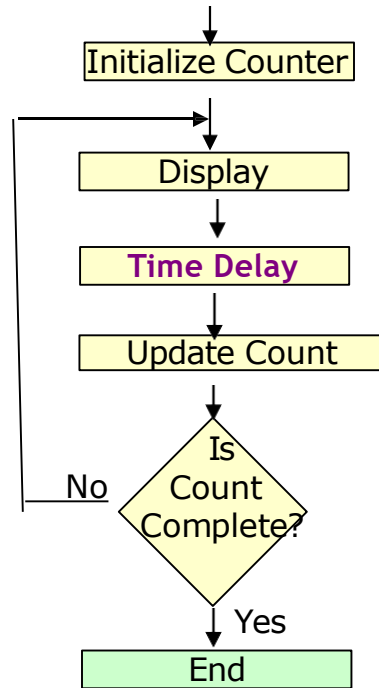$T_{Delay} = 57412 \times 0.5 \ \mu Sec = 28.706$ mSec

Increasing the Time Delay

✓ The Delay can be further increased by using register pairs for each of the loop counters in the nested loops setup.

✓ It can also be increased by adding dummy instructions (like NOP) in the body of the loop.

Counter Design with Time Delay

Variations Of Counter Flowchart

ILLUSTRATIVE PROGRAMS
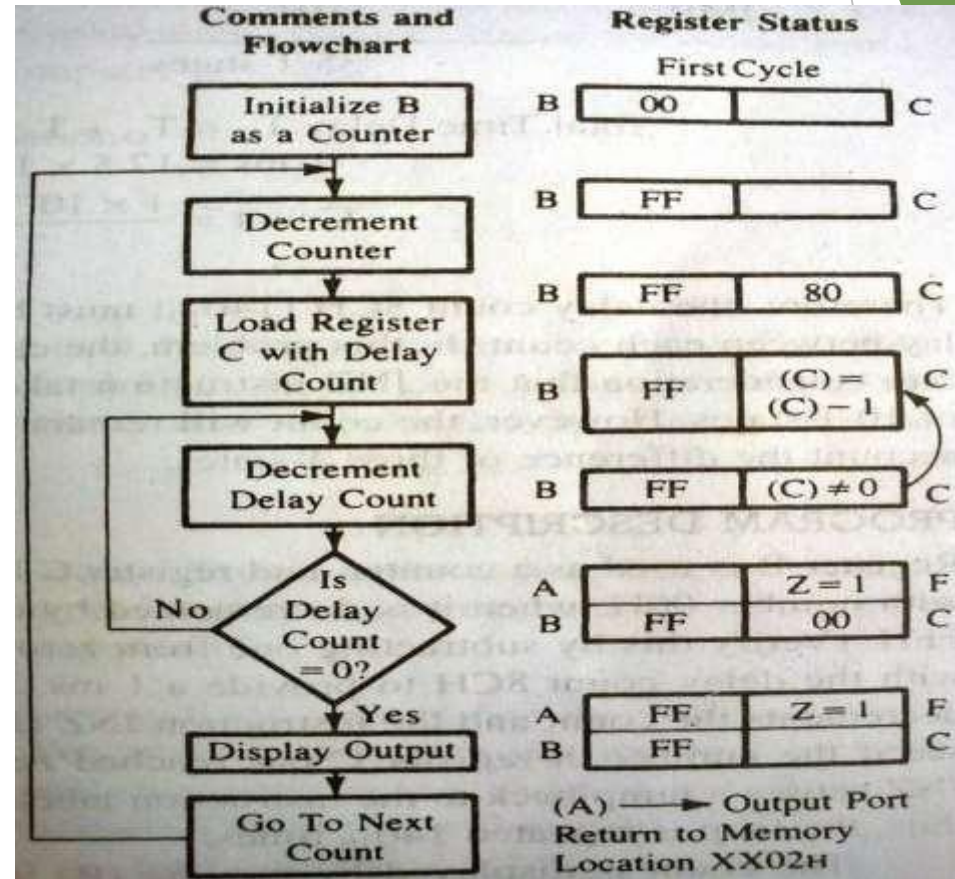
Hexadecimal counter

Write a program to count continuously In hexadecimal from FFh to 00 h in a System with a 0.5 microsec   clock period. Use register C to set up a **one Millisecond** delay between each count and display the numbers at one of the output ports.

Hexadecimal counter

✓ This problem has two parts; the first is to set up a continuous down-counter, and the second is to design a given delay between two counts.

1. The hexadecimal counter is set up by loading a register with an appropriate starting number and decrementing it until it becomes zero. After zero count, the register goes back to FF because decrementing zero results in a (-1), which is FF in 2's complement.

2. The 1 ms delay between each count is set up by using delay techniques.

Hexadecimal counter

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


Comments and Flowchart	Register Status First Cycle		
Initialize B as a Counter	B	00	C
Decrement Counter	B	FF	C
Load Register C with Delay Count	B	FF 80	C
	B	FF (C) = (C) − 1	C
Decrement Delay Count	B	FF (C) ≠ 0	C
Is Delay Count = 0? No	A Z = 1 F; B FF 00 C		
Yes	A FF Z = 1 F; B FF C		
Display Output			
Go To Next Count	(A) → Output Port Return to Memory Location XX02H		

Time Delay Calculation

✓ Delay loop includes two instructions: DCR C and JNZ with 14 T-states. Therefore the time delay T_L in the loop (without accounting for the fact that JNZ requires 7 T-States in the last cycle, because count will remain same even if the calculations take into account the difference of 3 T-States) is:

$$T_L = 14 \text{ T-states X T (Clock period) X Count}$$
$$= 14 \text{ X } (0.5 \text{ x } 10^{-6}) \text{ x Count}$$
$$= (7.0 \text{ X } 10^{-6}) \text{ X Count}$$

✓ The Delay outside the loop includes the following instructions:

DCR B	4T	Delay outside the loop:
MVI C,COUNT	7T	T_o = 35 T-States * T
MOV A,B	4T	= 35 * (0.5* 10^{-6})
OUT PORT	10T	= 17.5 µs
JMP	10T	
	35 T-States	

Total Time Delay $T_D = T_o + T_L$

1ms= $17.5 * 10^{-6} + (7.0 * 10^{-6})$ *Count

Count= $\dfrac{1\text{x } 10^{-3} - 17.5 \text{ x } 10^{-6}}{7.0 \text{ x } 10^{-6}} \approx 140_{10}$ ms

Hence, a delay count $8CH(140_{10})$ must be loaded in C.

Modulo TEN counter (0-9)

Write a program to count from 0 - 9 with a one-second delay Between each count. At count of 9, the counter should reset Itself to 0 and repeat the sequence continuously. Use Register pair HL to set up the Delay, and display each count at one of the output ports. Assume the clock frequency of the Microcompter is 1 M hz.

Modulo TEN counter (0-9)

START:	MVI B,ooH	7
	MOV A,B	4
DISPLAY:	OUT PORT#	10
	LXI H,16-Bit	10
LOOP:	DCX H	6
	MOV A,L	4
	ORA H	4
	JNZ LOOP	10/7
	INR B	4
	MOV A,B	4
	CPI 0AH	7
	JNZ DISPLAY	10/7
	JZ START	10/7
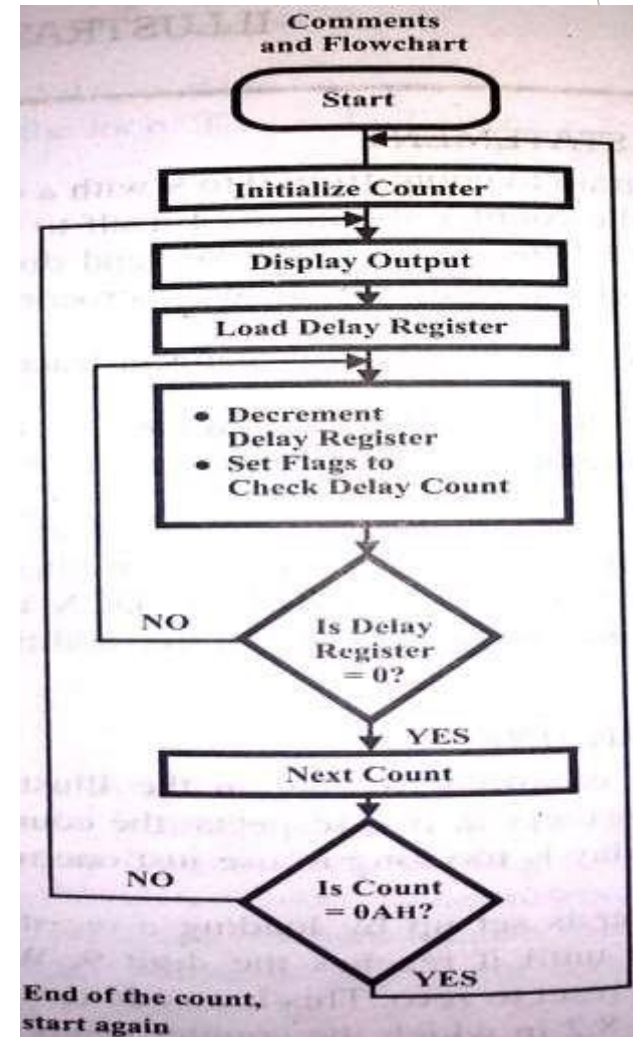
Time Delay Calculation

✓ The major delay between two counts is provided by the 16-bit number in the delay register HL(inner loop in flow chart). This delay is set up by using a register pair.

Loop Delay T_L= 24 T-states *T * Count

1 second= 24*1.0 *10^{-6} *Count

Count=___1___= 41666 = A2C2H

24 x 10^{-6}

A2C2H would provide approx 1 sec delay between two counts. To achieve higher accuracy in the delay, the instructions outside the loop must be accounted for delay calculation. (will be 41665).

Generating Pulse waveforms

Write a program to generate a continuous square wave with the period of 500 micro sec.
Assume the system clock period is 325 ns,   and use bit d0 to Output the square wave.

Generating   Pulse Waveforms

```
                MVI D,AA              7T
ROTATE:         MOV A,D               4T
                RLC                   4T
                MOV D,A               4T
                ANI 01H               7T          //mask MSB 7 Bits
                OUT PORT1             10T
                MVI B,COUNT
DELAY:          DCR B                 4T
                JNZ DELAY             10/7 T
                JMP ROTATE            10T
```

A   1 0 1 0 1 0 1 0
After RLC   0 1 0 1 0 1 0 1
A AND 01H   0 0 0 0 0 0 0 1

COUNT= 52.4_{10} =34H

Debugging counter & time- delay programs

1. Errors in counting T-States in a delay loop. Typically, the first instruction – to load a delay register – is mistakenly included in the loop.
2. Errors in recognizing how many times a loop is repeated.
3. Failure to convert a delay count from a decimal number into its hexadecimal equivalent.
4. Conversion error in converting a delay count from decimal to hexadecimal number or vice versa.
5. Specifying a wrong jump location.
6. Failure to set a flag, especially with 16-bit decrement/increment instructions.
7. Using a wrong jump instruction.
8. Failure to display either the first or the last count.
9. Failure to provide a delay between the last and the last-but- one count.

End of Session:

Thank You