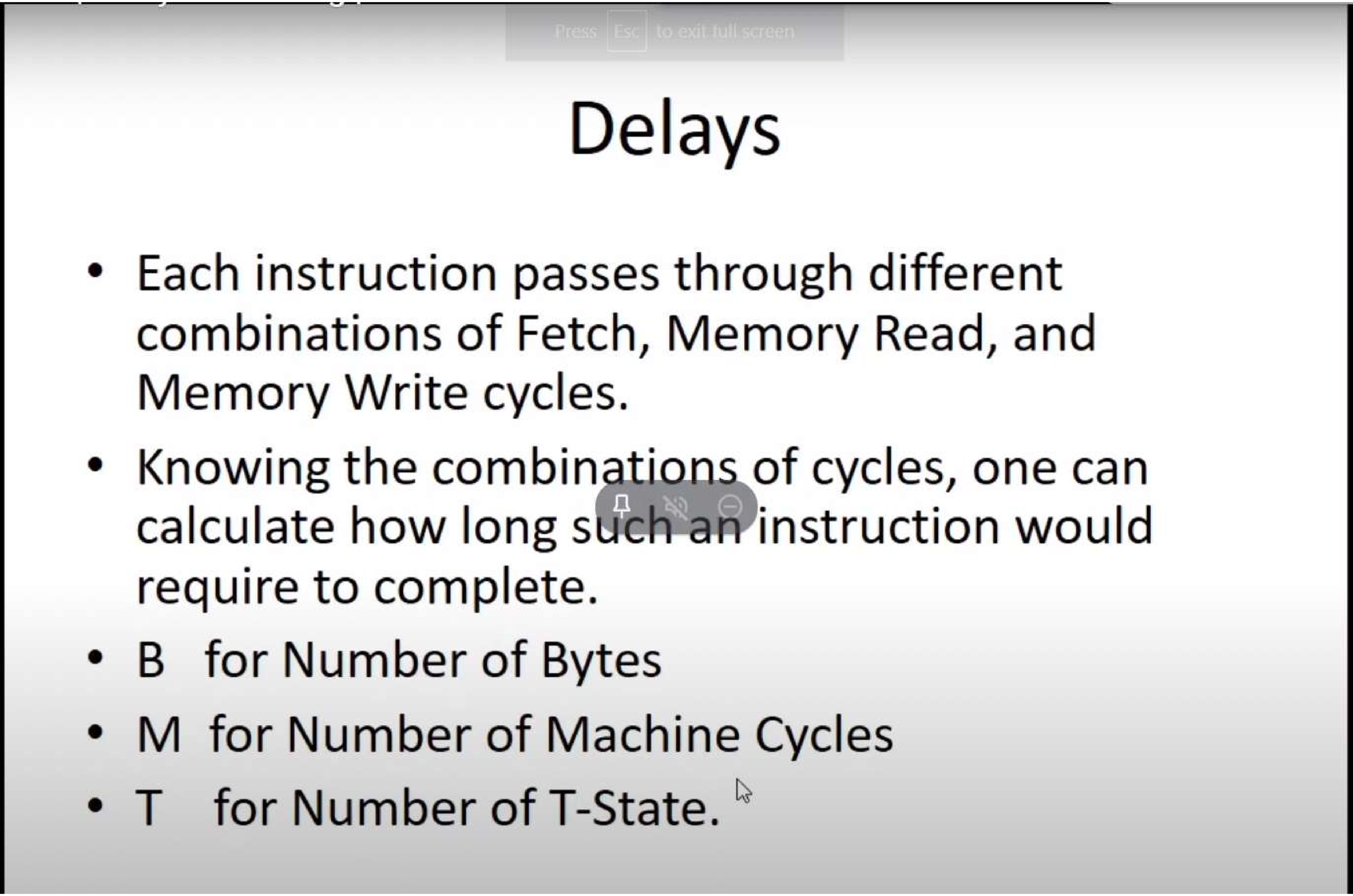


Delays

- Each instruction passes through different combinations of Fetch, Memory Read, and Memory Write cycles.
- Knowing the combinations of cycles, one can calculate how long such an instruction would require to complete.
- B   for Number of Bytes
- M for Number of Machine Cycles
- T   for Number of T-State.

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 using the following formula:
- Delay = No. of T-States / Frequency
- For example a “MVI” instruction uses 7 T-States. Therefore, if the Microprocessor is running at 2 MHz, the instruction would require 3.5 µSeconds to complete.

Delay loops

- We can use a loop to produce a certain amount of time delay in a program.
- The following is an example of a delay loop:

```
        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

Delay Loops

- 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:

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

Tdelay= total delay, TO= delay outside the loop, TL= delay of the loop

- TO is the sum of all delays outside the loop.

Clock frequency of the system $f = 2$ MHz

Clock period $T = 1/f = 1/2 \times 10^{-6} = 0.5\ \mu s$

Time to execute MVI $= 7$ T-states $\times 0.5$

$= 3.5\ \mu s$

The time delay in the loop T_L with 2 MHz clock frequency is calculated as

$$TL = (T \times \text{Loop T-states} \times N10)$$

where T_L = Time delay in the loop

T = System clock period

N_{10} = Equivalent decimal number of the hexadecimal count loaded in the delay register

$T_L = (0.5 \times 10^{-6} \times 14 \times 255)$

$= 1785\ \mu s$

≈ 1.8 ms

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

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

$$\text{Total Delay} = \begin{array}{l}\text{Time to execute instructions}\\ \text{outside loop}\end{array} + \begin{array}{l}\text{Time to execute}\\ \text{loop instructions}\end{array}$$

$$\begin{aligned} T_D &= T_O + T_{LA} \\ &= (7 \times 0.5\ \mu s) + 1783.5\ \mu s = 1787\ \mu s \\ &\approx 1.8\ \text{ms} \end{aligned}$$

Using a Register Pair as a Loop Counter

- 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 as a Loop Counter

- 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
```

Label	Opcode	Operand	Comments	T-states
	LXI	B,2384H	;Load BC with 16-bit count	10
LOOP:	DCX	B	;Decrement (BC) by one	6
	MOV	A,C	;Place contents of C in A	4
	ORA	B	;OR (B) with (C) to set Zero flag	4
	JNZ	LOOP	;If result ≠ 0, jump back to LOOP	10/7

The time delay in the loop is calculated as in the previous example. The loop includes four instructions: DCX, MOV, ORA, and JNZ, and takes 24 clock periods for execution. The loop is repeated 2384H times, which is converted to decimals as

$$\begin{aligned} 2384\text{H} &= 2 \times (16)^3 + 3 \times (16)^2 + 8 \times (16)^1 + 4(16^0) \\ &= 9092_{10} \end{aligned}$$

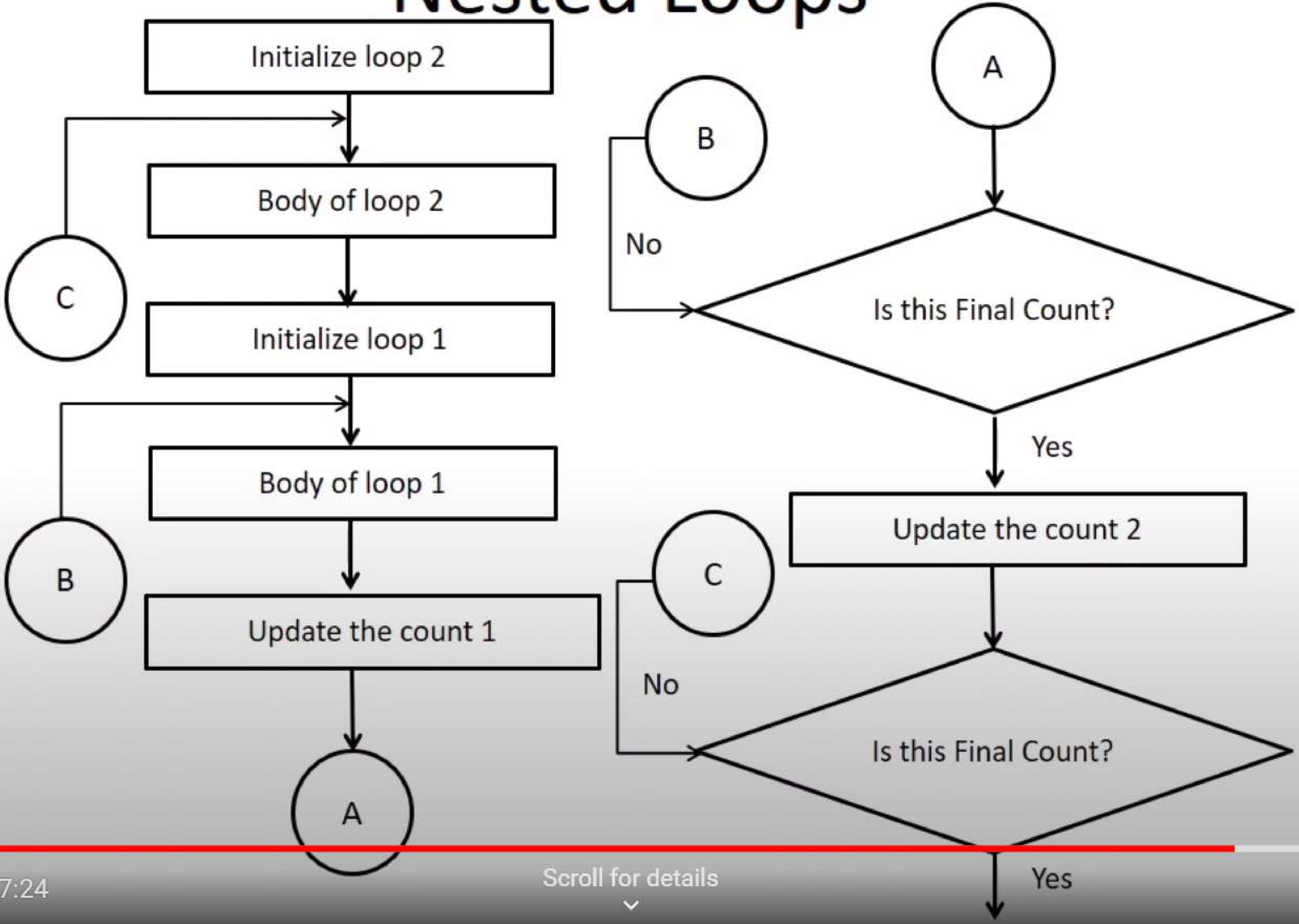
If the clock period of the system = 0.5 µs, the delay in the loop T_L is

$$\begin{aligned} T_L &= (0.5 \times 24 \times 9092_{10}) \\ &\approx 109 \text{ ms (without adjusting for the last cycle)} \\ \text{Total Delay } T_D &= 109 \text{ ms} + T_O \\ &\approx 109 \text{ ms (The instruction LXI adds only 5 µs.)} \end{aligned}$$

Nested Loops

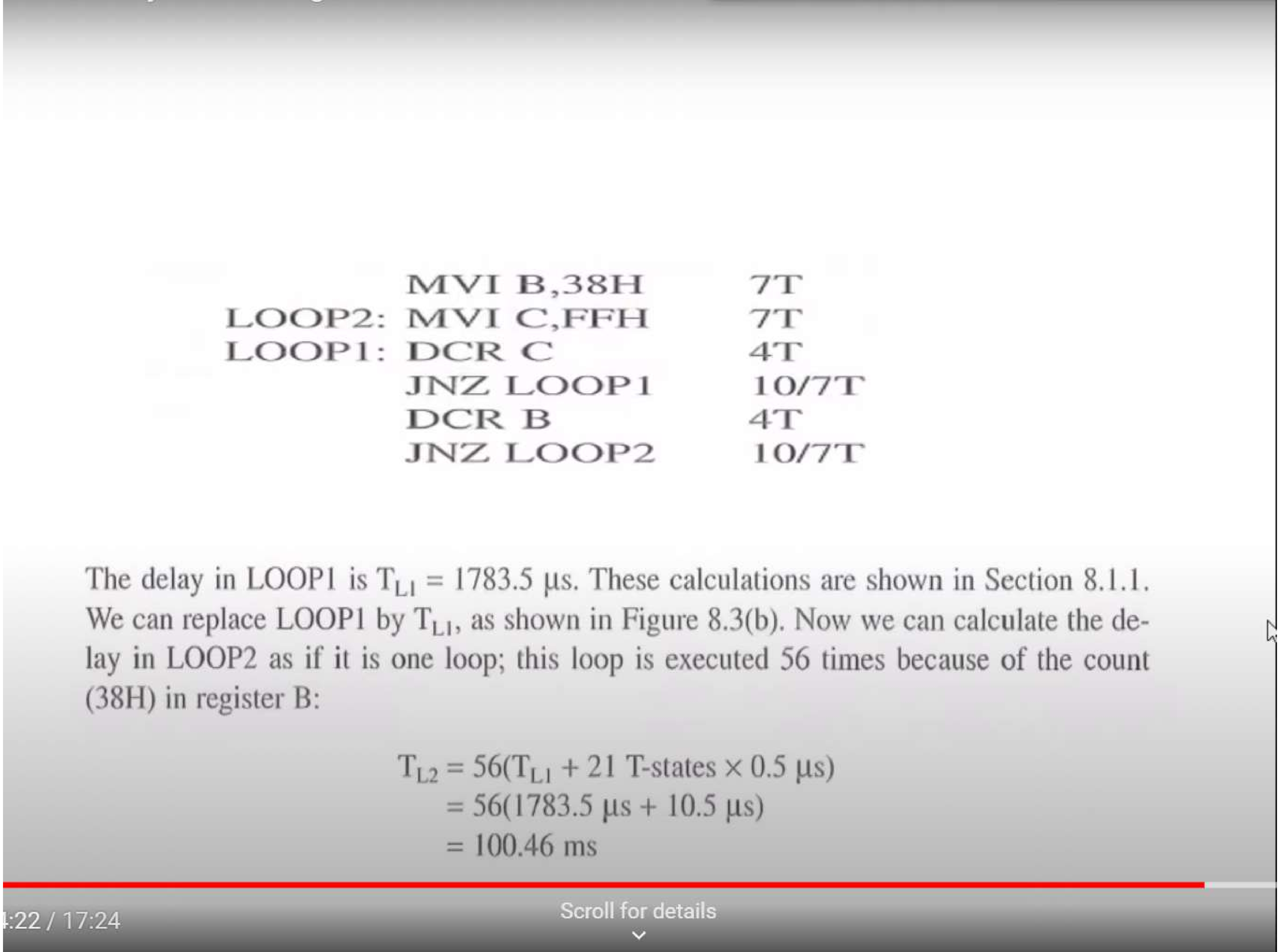
- Nested loops can be easily setup in Assembly language by using two registers for the two loop counters and updating the right register in the right loop.
- In the figure, the body of loop2 can be before or after loop1.

Nested Loops



4:04 / 17:24

Scroll for details



```
          MVI B,38H       7T
LOOP2:    MVI C,FFH       7T
LOOP1:    DCR C           4T
          JNZ LOOP1       10/7T
          DCR B           4T
          JNZ LOOP2       10/7T
```

The delay in LOOP1 is $T_{L1} = 1783.5\ \mu s$. These calculations are shown in Section 8.1.1. We can replace LOOP1 by T_{L1}, as shown in Figure 8.3(b). Now we can calculate the delay in LOOP2 as if it is one loop; this loop is executed 56 times because of the count (38H) in register B:

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

Nested Loops for Delay

- 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
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
```

Delay Calculation of Nested Loops

- The calculation remains the same except that it 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
  - TO1= 7 T-States
- MVI C, FFH instruction
  - TL1= (255 X 14) -3 = 3567 T-States
- 14 T-States for the DCR C and JNZ instructions repeated 255 times (FF16= 25510) minus 3 for the final JNZ

Increasing the 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.