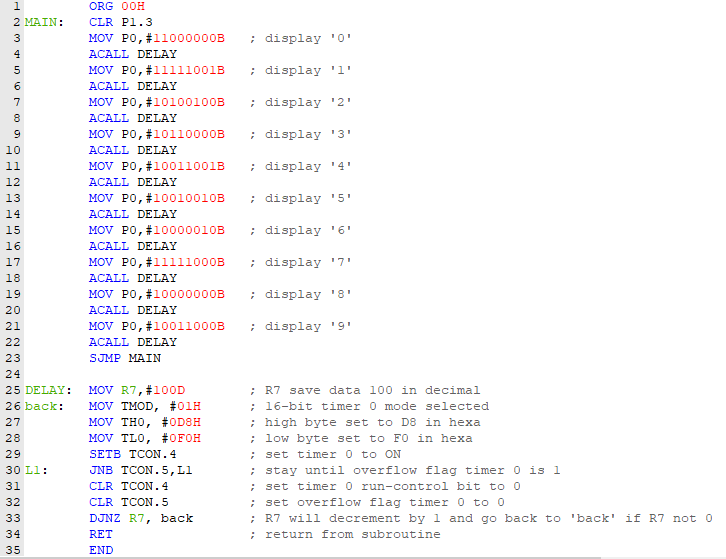
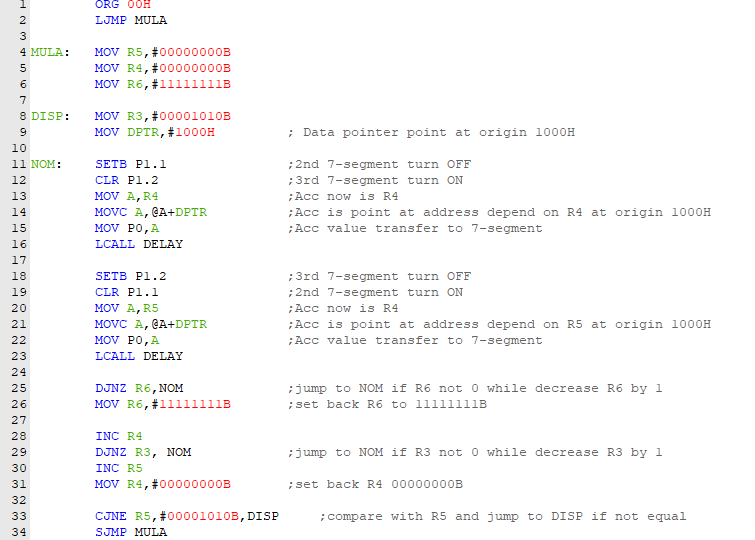
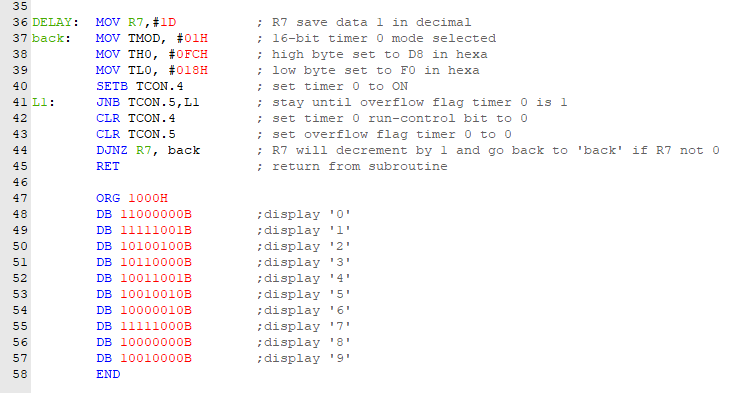
**ONE DIGIT 7-SEGMENT COUNT FROM ‘0’ TO ‘9’**



1. 4 7-segment digit is control by port 1.0, 1.1, 1.2 and 1.3 while 7segment is control by port 0.
2. Firstly, port 1.3 must be clear since we want 4th digit to count. Then, we set at port 0 11000000B to display number ‘0’ because this 7segment active-LOW type.
3. DELAY is called so it will jump to DELAY to be executed for next instruction
4. R7 data is set to 100 in decimal
5. Timer mode will set to 01 in hexa to select 16-bit timer 0
6. For crystal oscillator, we want timer to count for 10ms so, there need to make timer count up to 10 000 since 1 machine cycles is 1µs. So, 65 536 (full timer count for crystal oscillator) must minus 10 000, 65 536 – 10 000 = 55 536 = DBF0H
7. TH0 is for high byte set so DB will be set while TH1 is for low byte set so F0 will be set
8. TCON.4 is for Timer 0 run-control bit so it will be set 1 to timer start counting
9. The instruction for timer count will keep remain until overflow flag is 1. In other words, Timer 0 finish counting.
10. TCON.4 and TCON.5 will set back to 0 to be ready to execute next instruction
11. R7 will decrement by 1 so, R7 data now is 99 and instruction will execute ‘back’ until R7 decrease until 0.
12. After R7 become 0 the instruction will return from subroutine where DELAY is called. This will give 7-segment to display ‘0’ for 100x10ms=1s
13. Then, we set port 0 111111001B to make 7segment display ‘1’ then the DELAY is called to make it display for another 1s
14. The cycle continue until ‘9’ and it will jump back to first instruction to display back ‘0’. The 7-segment will continue count from ‘0’ to ‘9’ non-stop

**TWO DIGIT 7-SEGMENT COUNT FROM ‘0’ TO ‘99’**

1. Firstly set R5 and R4 to 00000000B while R6 to 11111111B and R3 to 00001010B. Then also set data pointer to origin 1000H
2. For origin 1000H list down 7-segment code to display number ‘0’ to ‘9’
3. Set P1.1 to 1 while P1.2 to 0. This will turn OFF 2nd 7-segment while turn ON 3rd 7-segment.
4. Move current R4 which is 00H to accumulator. Then move code byte relative to DPTR to accumulator. Data pointer is now take value in address 00H at origin 1000H which is 11000000B.
5. Data in accumulator now is move to P0 which give the 7-segment display number ‘0’.
6. DELAY is called so it will jump to DELAY to be executed for next instruction
7. R7 data is set to 1 in decimal
8. Timer mode will set to 01 in hexa to select 16-bit timer 0
9. For crystal oscillator, we want timer to count for 1ms so, there need to make timer count up to 1 000 since 1 machine cycles is 1µs. So, 65 536 (full timer count for crystal oscillator) must minus 1 000, 65 536 – 1 000 = 64 536 = FC18H
10. TH0 is for high byte set so FC will be set while TH1 is for low byte set so 18 will be set
11. TCON.4 is for Timer 0 run-control bit so it will be set 1 to timer start counting
12. The instruction for timer count will keep remain until overflow flag is 1. In other words, Timer 0 finish counting.
13. TCON.4 and TCON.5 will set back to 0 to be ready to execute next instruction
14. R7 will decrement by 1 so, R7 data now is 0 and instruction will return back from subroutine was called. So, 3rd 7-segment will show number ‘0’ for 1ms
15. Set P1.2 to 1 while P1.1 to 0. This will turn ON 2nd 7-segment while turn OFF 3rd 7-segment.
16. The next instruction will be same as before until DELAY called. Then these instruction will kept repeating for 255 times since the instruction is jump to NOM until R6 become 0.
17. These cycle will give 2nd and 3rd 7-segment turn ON OFF repeatedly but in fast cycle which is 255x1ms = 255ms for display number ‘0’ only. Because of the fast cycle, the 7-segment will show as seem not blinking and we can see the 2nd and 3rd digit displaying number ‘00’ for about 2x255ms = 0.5s.
18. Then, R4 will increase by 1 and it will jump back to NOM if R3 is not 0 while decrease R3 by 1.
19. The cycle will same as before, but R4 is pointing at address 01H at origin 1000H and this will give 7-segment show ‘01’ for another 0.5s.
20. Then, cycle will keep repeating by increasing R4 until 7-segment show ‘09’ and the cycle will stop after R3 become 0
21. After count until ‘09’, it will increment for R5 and R4 is set back 00H to give back another cycle from NOM. This time 7-segment will show ‘10’ and keep increasing until ‘99’
22. The cycle stop after R5 is same as 9 which mean finish count up to ‘99’ and it will jump back to MULA and the count will start from beginning back which is ‘00’