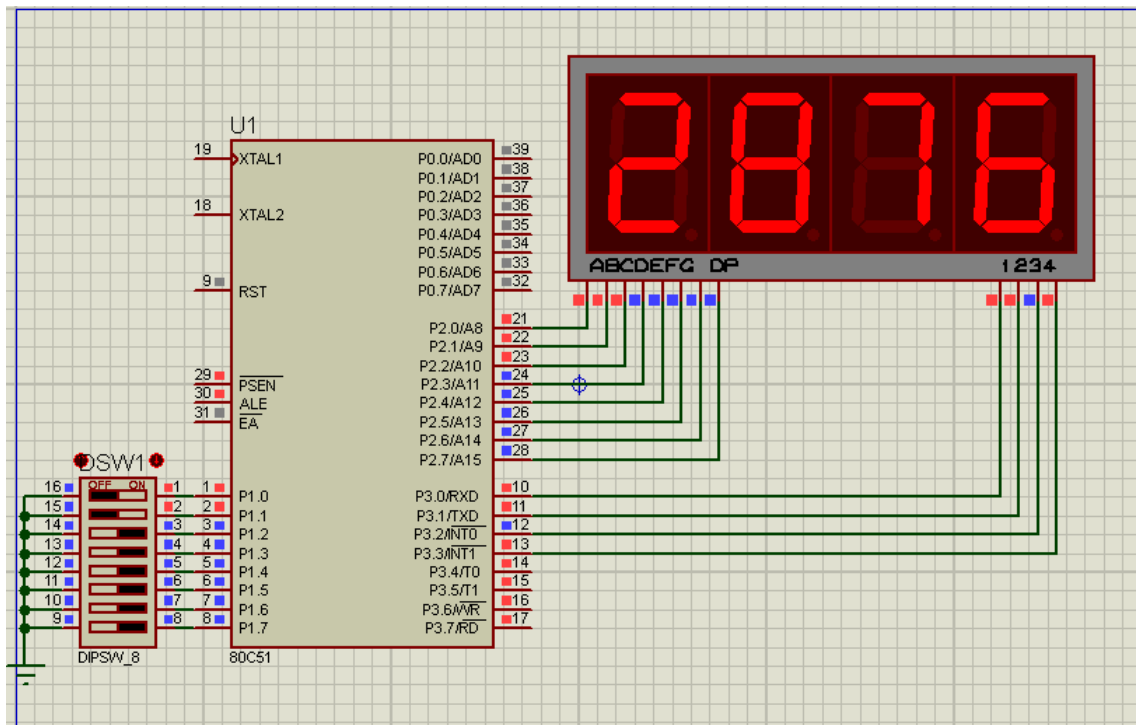


Problem 14



In this project, we display a 4 digit number on a multiplexed 7-segment display according the value read from the dip switches

If no dip switch is active → display blank

Each switch corresponds to a stored number, and if a switch is activated, its corresponding number is displayed

Multiple switch activated will display its corresponding numbers one after the other.

Section 1 initialization

```

19 DISP_PORT EQU P2
20 DIG0 EQU P3.3
21 DIG1 EQU P3.2
22 DIG2 EQU P3.1
23 DIG3 EQU P3.0
24
25 N1 EQU 30H
26 N2 EQU 31H
27 N3 EQU 32H
28 N4 EQU 33H
29
30 SW1 EQU P1.0
31 SW2 EQU P1.1
32 SW3 EQU P1.2
33 SW4 EQU P1.3
34 SW5 EQU P1.4
35 SW6 EQU P1.5
36 SW7 EQU P1.6
37 SW8 EQU P1.7

```

DISP_PORT the port that we send on it the 7-segment code
DIG0-DIG3 are the common cathode for each 7-segment.
N1-N4, are the current number to be displayed
SW1-SW8, the 8-bits of DIP switches

Section 2 Main code

```

54 Start:
55     SETB DIG0
56     SETB DIG1
57     SETB DIG2
58     SETB DIG3
59
60 LOOP:
61     JNB SW1,NOT1
62     MOV DPTR,#NUM1
63     CALL DISP_NUM
64 NOT1:
65     JNB SW2,NOT2
66     MOV DPTR,#NUM2
67     CALL DISP_NUM
68 NOT2:
69     JNB SW3,NOT3
70     MOV DPTR,#NUM3
71     CALL DISP_NUM
72 NOT3:
73     JNB SW4,NOT4
74     MOV DPTR,#NUM4

```

```

73     JNB SW4,NOT4
74     MOV DPTR,#NUM4
75     CALL DISP_NUM
76 NOT4:
77     JNB SW5,NOT5
78     MOV DPTR,#NUM5
79     CALL DISP_NUM
80 NOT5:
81     JNB SW6,NOT6
82     MOV DPTR,#NUM6
83     CALL DISP_NUM
84 NOT6:
85     JNB SW7,NOT7
86     MOV DPTR,#NUM7
87     CALL DISP_NUM
88 NOT7:
89     JNB SW8,NOT8
90     MOV DPTR,#NUM8
91     CALL DISP_NUM
92 NOT8:
93     JMP LOOP

```

55-58, we make all cathode = 5V → this means all digits are in-active (for common cathode, the cathode must be 0V to activate the digit)

In the main loop, we test each switch starting from sw1 to sw8.

If a switch is activated = "1", we load 'DPTR' with the address of the stored number corresponding to the activated switch, and call the function that will display this number.

For example if sw1 = 1, we load DPTR with the address of NUM1 (62)

The function DISP_NUM:

```

96     CLR A
97     MOVC A,@A + DPTR
98     INC DPTR
99     MOV N1,A
100    CLR A
101    MOVC A,@A + DPTR
102    INC DPTR
103    MOV N2,A
104    CLR A
105    MOVC A,@A + DPTR
106    INC DPTR
107    MOV N3,A
108    CLR A
109    MOVC A,@A + DPTR
110    INC DPTR
111    MOV N4,A

```

First we use the data pointer to read the corresponding digits into N1-N4

Next as shown in the following code, we pass through the four digits one at a time. Each digit will be displayed for 5ms → to pass through all 4 digits we will take 20ms approximately. If we repeat this process 50 times, this means that the number will be displayed for a time of 1 second.

```
113     MOV R4,#50
114 REPEAT_1000MS:
115     MOV A,N4
116     CALL SEND_CODE
117     CLR DIG0
118     CALL DELAY
119     SETB DIG0
120
121     MOV A,N3
122     CALL SEND_CODE
123     CLR DIG1
124     CALL DELAY
125     SETB DIG1
126
127     MOV A,N2
128     CALL SEND_CODE
129     CLR DIG2
130     CALL DELAY
131     SETB DIG2
132
133     MOV A,N1
134     CALL SEND_CODE
135     CLR DIG3
136     CALL DELAY
137     SETB DIG3
138
139     DJNZ R4,REPEAT_1000MS
140 RET
...
```

So, we will repeat the process 50 times (113).

- Each time we read one of the four digits to be displayed(115),
- Convert it to its corresponding 7-segment code (116).
- Activate its cathode (117)
- Allow it to be displayed for a time of 5ms (118)

This will be done for all 4 digits

Functions DELAY, SEND_CODE

```

142 ; delay for 5ms
143 DELAY:
144     MOV R6,#10
145 L1:
146     MOV R7,#250
147 L0:
148     DJNZ R7,L0
149     DJNZ R6,L1
150 RET
151
152 SEND_CODE:
153     MOV DPTR,#DIGIT_CODE
154     MOVC A,@A + DPTR
155     MOV DISP_PORT,A
156 RET

```

The function delay will make a pause of $250 \times 10 \times 2\mu\text{s} = 5\text{ ms}$

The function SEND_CODE, will convert the number to its corresponding 7-segment code from the table "DIGIT_CODE"

```

158 DIGIT_CODE:
159 DB 3FH ; digit drive pattern for 0
160 DB 06H ; digit drive pattern for 1
161 DB 5BH ; digit drive pattern for 2
162 DB 4FH ; digit drive pattern for 3
163 DB 66H ; digit drive pattern for 4
164 DB 6DH ; digit drive pattern for 5
165 DB 7DH ; digit drive pattern for 6
166 DB 07H ; digit drive pattern for 7
167 DB 7FH ; digit drive pattern for 8
168 DB 6FH ; digit drive pattern for 9
...

```

Finally the numbers corresponding to each switch

```

170 NUMBERS:
171 NUM1:
172     DB 1,2,3,4
173 NUM2:
174     DB 2,8,7,6
175 NUM3:
176     DB 3,7,1,0
177 NUM4:
178     DB 4,0,9,8
179 NUM5:
180     DB 5,5,3,0
181 NUM6:
182     DB 6,6,3,1
183 NUM7:
184     DB 7,1,2,8
185 NUM8:
186     DB 8,0,5,4

```

