Problem 2

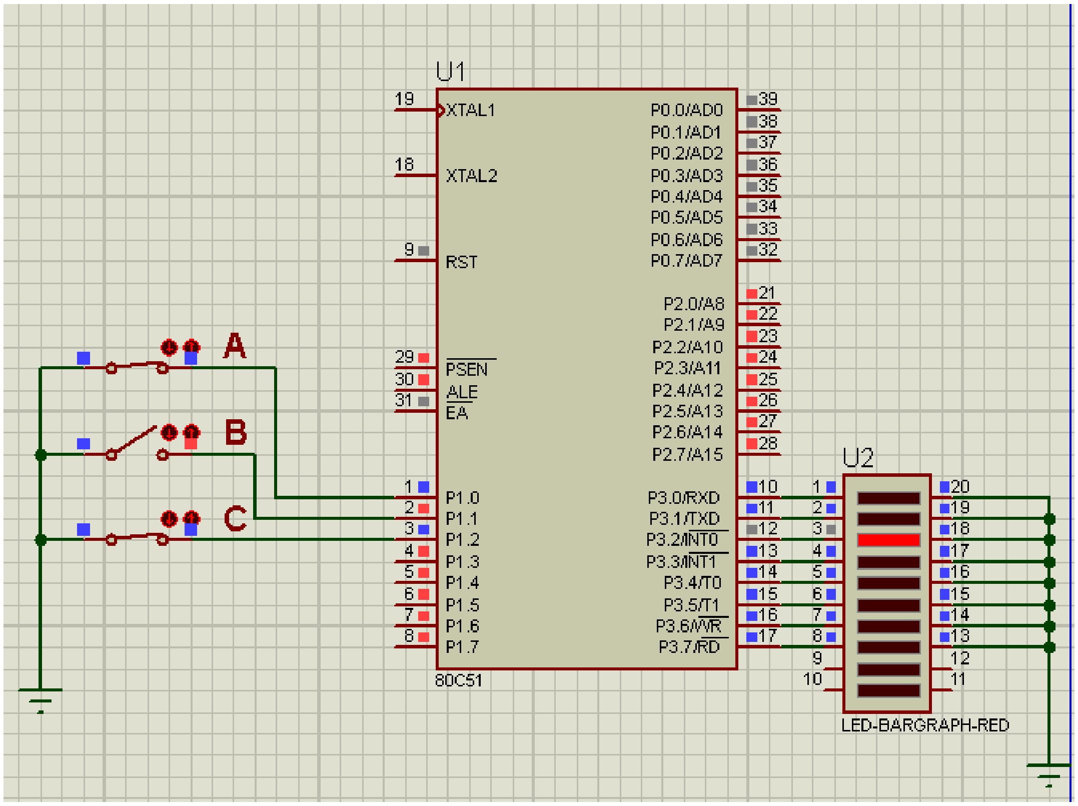
In this problem, we choose a 3 to 8 decoder circuit to be implemented

Inputs are A, B, and C, where output will be the full port pins of PORT3.

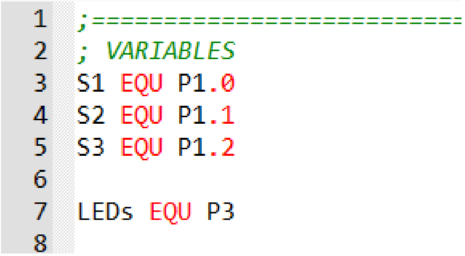
The 3 to 8 decoder has the following truth table

Inputs			Outputs							
A	B	C	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Here is the circuit



Section 1 variable definitions



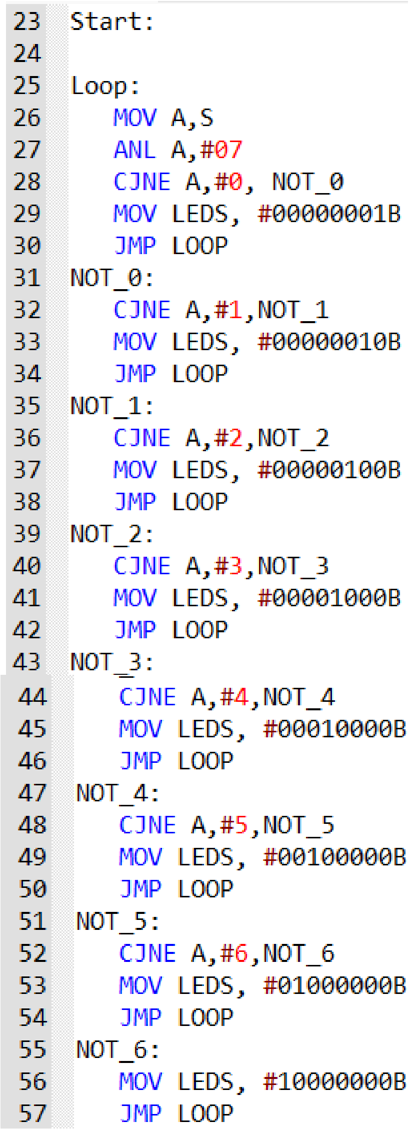
```
;=========================
; VARIABLES
S1 EQU P1.0
S2 EQU P1.1
S3 EQU P1.2

LEDs EQU P3

```

Now we use the whole port P3 as output with the name of LEDs

Section 2 code



```
23  Start:
24
25  Loop:
26     MOV A,S
27     ANL A,#07
28     CJNE A,#0, NOT_0
29     MOV LEDS, #00000001B
30     JMP LOOP
31  NOT_0:
32     CJNE A,#1,NOT_1
33     MOV LEDS, #00000010B
34     JMP LOOP
35  NOT_1:
36     CJNE A,#2,NOT_2
37     MOV LEDS, #00000100B
38     JMP LOOP
39  NOT_2:
40     CJNE A,#3,NOT_3
41     MOV LEDS, #00001000B
42     JMP LOOP
43  NOT_3:
44     CJNE A,#4,NOT_4
45     MOV LEDS, #00010000B
46     JMP LOOP
47  NOT_4:
48     CJNE A,#5,NOT_5
49     MOV LEDS, #00100000B
50     JMP LOOP
51  NOT_5:
52     CJNE A,#6,NOT_6
53     MOV LEDS, #01000000B
54     JMP LOOP
55  NOT_6:
56     MOV LEDS, #10000000B
57     JMP LOOP
```

The code starts by reading the state of switches A, B, and C. It reads first P1, then it zeros out all its bits except the first 3 bits (A, B, C) as in lines 26,27.

Note that `ANL P1,#7` is a bitwise logical 'AND', that will perform AND operation between each bit of 'A' and its corresponding bit on '7'.

By observing that '7' = 00000111 in binary, the `ANL` operation will result in leaving only the first 3 bits of 'A' which represents the input code.

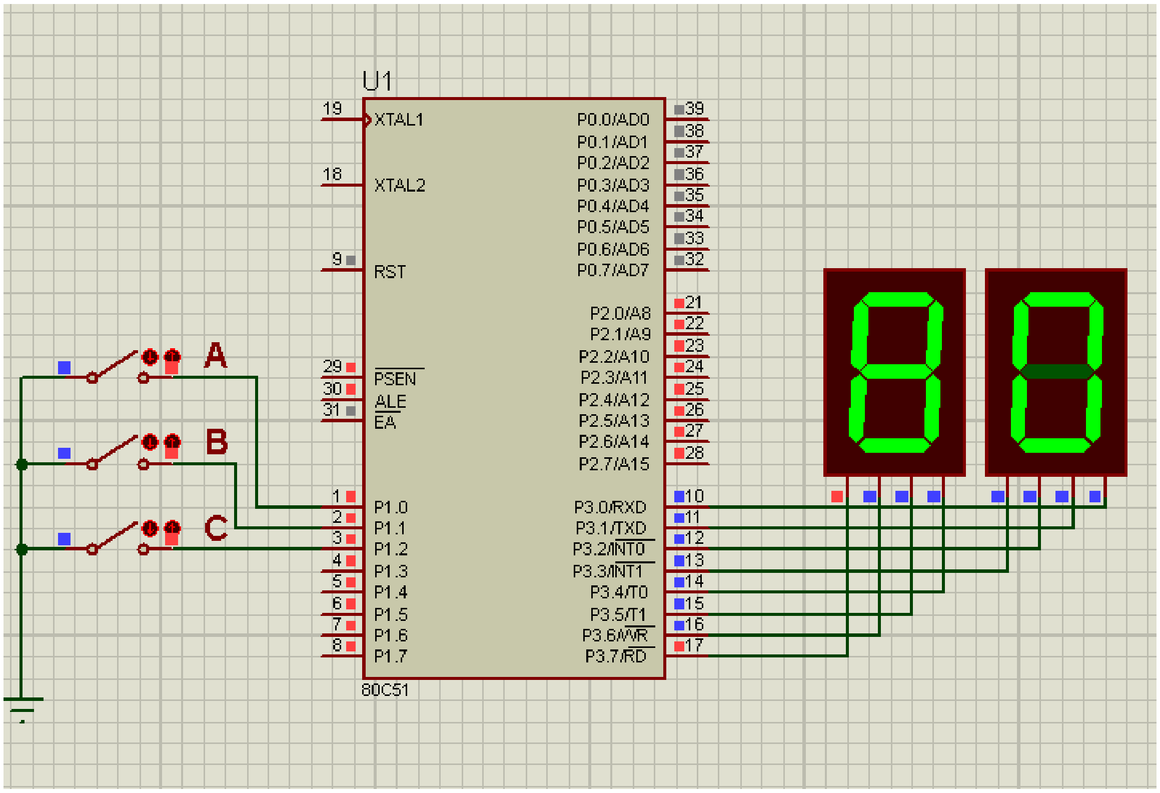
Now we will compare this code to all possible values using the "Compare and Jump if Not Equal" (CJNE) instruction to determine the required output.

For example `CJNE A,#0,NOT_0`, will jump and skip the next instruction if 'A' is not equal '0'. Hence if A = 0, the next instruction (`MOV LEDS,#00000001B`) will be executed.

The process repeats for all other possible value.

Version 2

In this version we replace the LEDs with HEX 7-Segment display as shown



Now the display will output show the hex value of the output as indicated in the following table

Inputs			Outputs								Hex output
A	B	C	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	
0	0	0	0	0	0	0	0	0	0	1	01
0	0	1	0	0	0	0	0	0	1	0	02
0	1	0	0	0	0	0	0	1	0	0	04
0	1	1	0	0	0	0	1	0	0	0	08
1	0	0	0	0	0	1	0	0	0	0	10
1	0	1	0	0	1	0	0	0	0	0	20
1	1	0	0	1	0	0	0	0	0	0	40
1	1	1	1	0	0	0	0	0	0	0	80

There is no change in the code as we just change the display type.