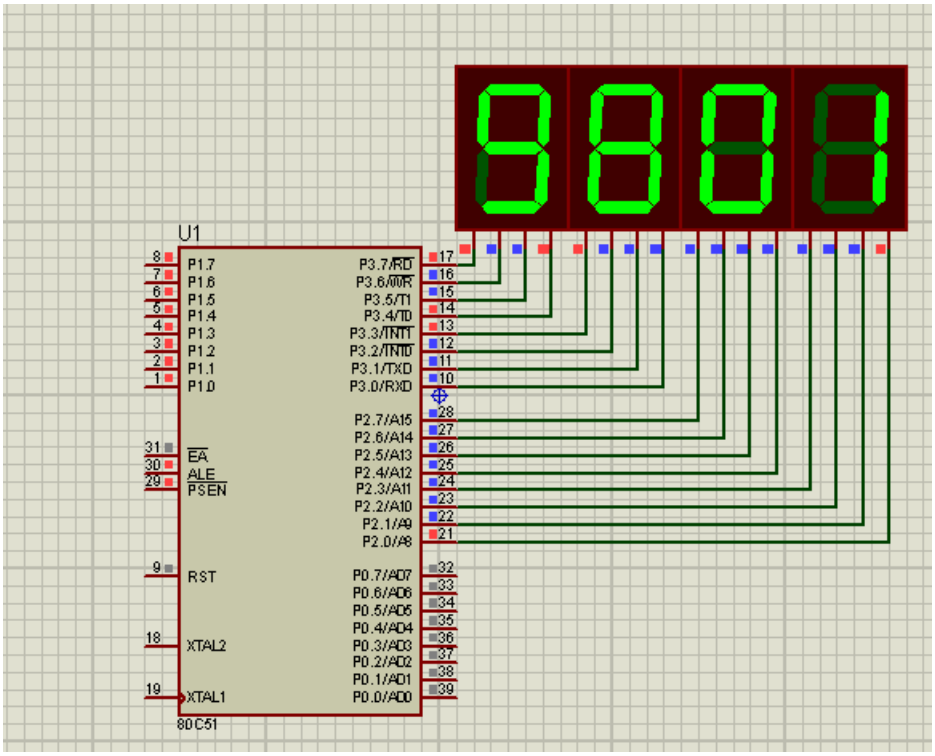
Problem 29_one_BCD



In this problem we will multiply two numbers, each number is represented by one BCD digits/byte with a length of 2 bytes

We will use the same previous algorithm

Functions

1-multiply_one_BCD

```
25  MULTIPLY_ONE_BCD:
26     MOV A,TEMP1
27     MOV B,TEMP2
28     MUL AB
29     MOV B,#10
30     DIV AB
31     MOV X1,B
32     MOV X2,A
33
34     MOV A,TEMP1
35     MOV B,TEMP2+1
36     MUL AB
37     MOV B,#10
38     DIV AB
39     MOV X3,B
40     MOV X4,A
41
42     MOV A,TEMP1+1
43     MOV B,TEMP2
44     MUL AB
45     MOV B,#10
46     DIV AB
47     MOV X5,B
48     MOV X6,A
49
49
50     MOV A,TEMP1+1
51     MOV B,TEMP2+1
52     MUL AB
53     MOV B,#10
54     DIV AB
55     MOV X7,B
56     MOV X8,A
```

25-32 multiply Y1*Z1 and store the result in X1,X2

34-40 multiply Y1*Z2 and store the result in X3,X4

42-48 multiply Y2*Z1 and store the result in X5,X6

50-56 multiply Y2*Z2 and store the result in X7,X8

Before saving partial product, we must isolate two BCD digits by using the division by 10 as in (29-30) → A will have the higher nibble and B the lower nibble

```
58      MOV RESULT,X1
59      MOV A,X2
60      ADD A,X3
61      ADD A,X5
62      MOV B,#10
63      DIV AB
64      MOV RESULT+1,B
65
66      ADD A,X4
67      ADD A,X6
68      ADD A,X7
69      MOV B,#10
70      DIV AB
71      MOV RESULT+2,B
72      ADD A,X8
73      MOV RESULT+3,A
74
75      CALL DISP_RESULT_HEX1
76
77  RET
```

Now we make the addition to get R1 to R4

R1 is just X1 (58)

R2 = X2+X3+X5 (59-64)

R3 = X4+X6+X7 (66-71)

R4 = X8 (73)

Before saving the result we must convert the value in ACC into BCD, so we divide it by 10 (62-63) and store the remainder in the result (64), and accumulate the higher digit for the next addition(66)