MICROCONTROLLER AND ITS APPLICATIONS (ECE-3003)

TASK-2

Varun Agarwal

16BEC0450

L37+38

VARUN AGARWAL
16BEC0450

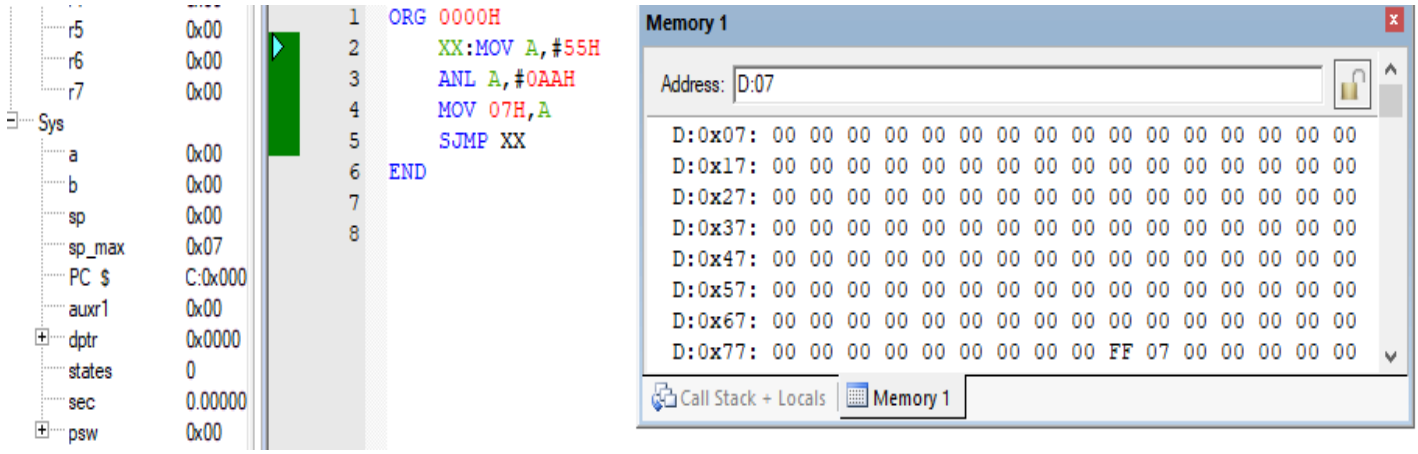
Program-1

Write an 8051 ALP for the following

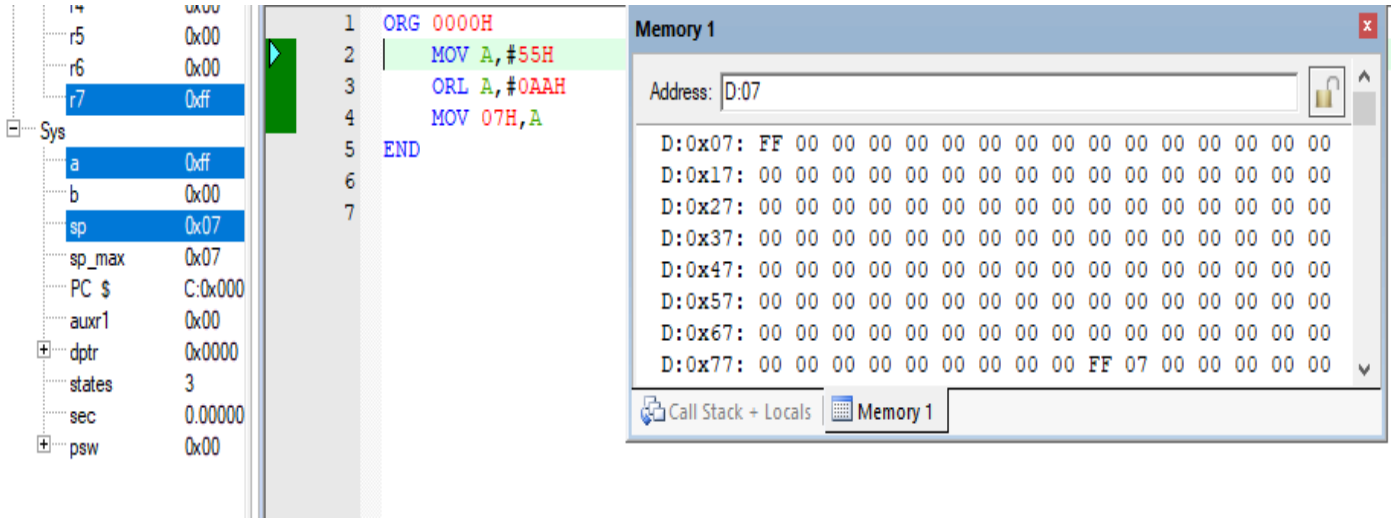
- 55H AND AAH
- 55H OR AAH
- 55H XOR AAH

Screenshot:

AND:

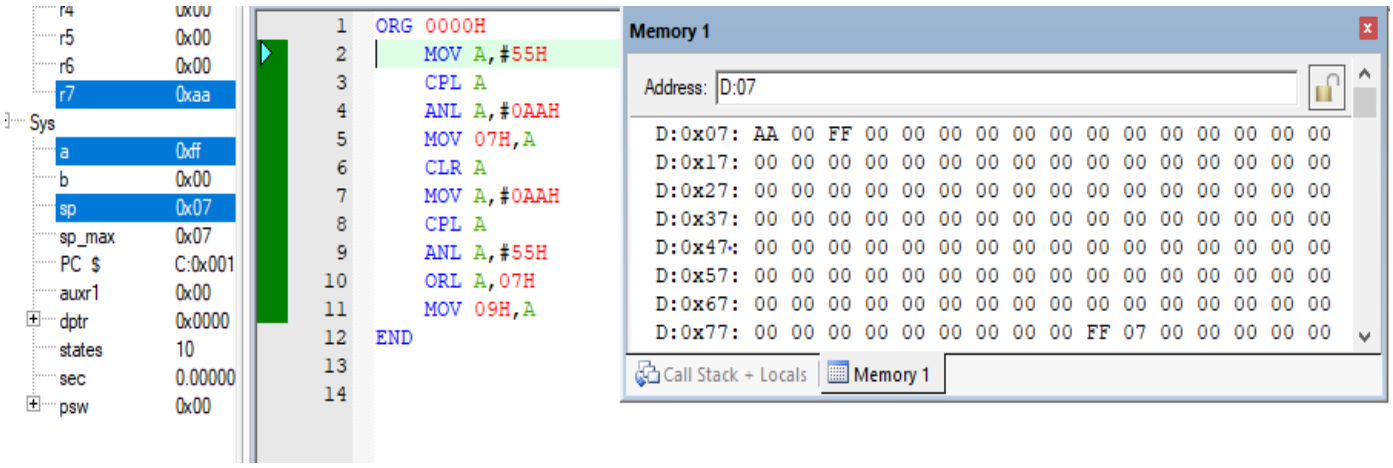


OR:



VARUN AGARWAL
16BEC0450

XOR:



Code:

AND:

```
ORG 0000H
        XX: MOV A, #55H
        ANL A, #0AAH
        MOV 07H, A
        SJMP XX
END
```

OR:

```
ORG 0000H
        MOV A, #55H
        ORL A, #0AAH
        MOV 07H, A
END
```

XOR:

```
ORG 0000H
        MOV A, #55H
        CPL A
        ANL A, #0AAH
        MOV 07H, A
```

VARUN AGARWAL
16BEC0450

```
        CLR A
        MOV A, #0AAH
        CPL A
        ANL A, #55H
        ORL A, 07H
        MOV 09H, A
    END
```

Program-2

- Write an 8051 ALP for the following TRUTH TABLE

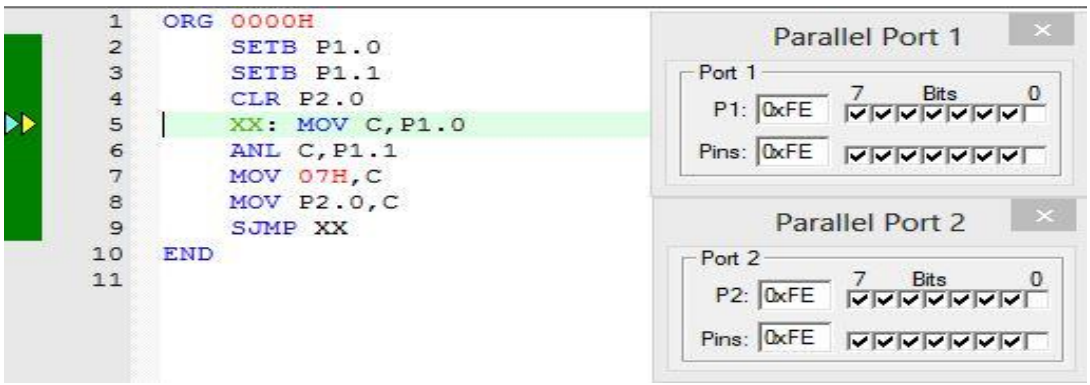
AND LOGIC		
INPUT		OUTPUT
A	B	Y = A.B
0	0	
0	1	
1	0	
1	1	

OR LOGIC		
INPUT		OUTPUT
A	B	Y = A+B
0	0	
0	1	
1	0	
1	1	

XOR LOGIC		
INPUT		OUTPUT
A	B	Y = A⊕B
0	0	
0	1	
1	0	
1	1	

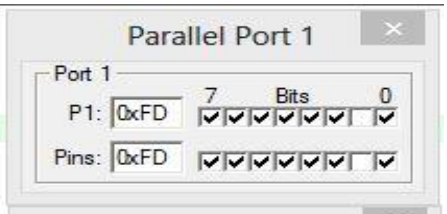
Screenshot:

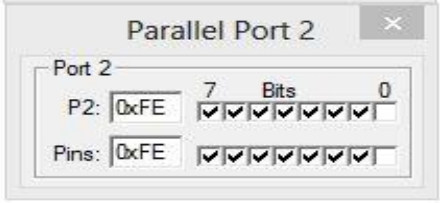
AND LOGIC:



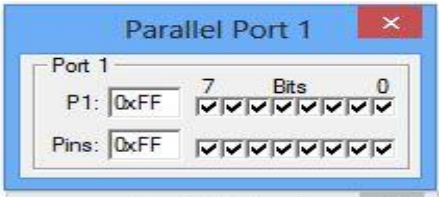
VARUN AGARWAL
16BEC0450

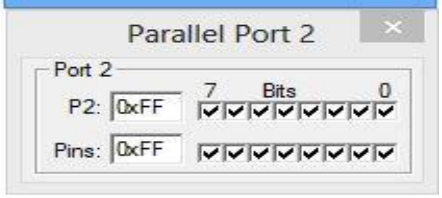
```
1   ORG 0000H
2       SETB P1.0
3       SETB P1.1
4       CLR P2.0
5       XX: MOV C,P1.0
6       ANL C,P1.1
7       MOV 07H,C
8       MOV P2.0,C
9       SJMP XX
10  END
11
```





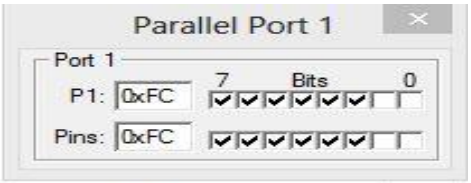

```
1   ORG 0000H
2       SETB P1.0
3       SETB P1.1
4       CLR P2.0
5       XX: MOV C,P1.0
6       ANL C,P1.1
7       MOV 07H,C
8       MOV P2.0,C
9       SJMP XX
10  END
11
```

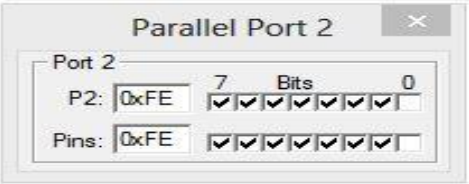




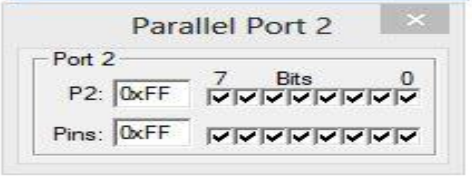
OR LOGIC:

```
1   ORG 0000H
2       SETB P1.0
3       SETB P1.1
4       CLR P2.0
5       XX: MOV C,P1.0
6       ORL C,P1.1
7       MOV 07H,C
8       MOV P2.0,C
9       SJMP XX
10  END
11
```



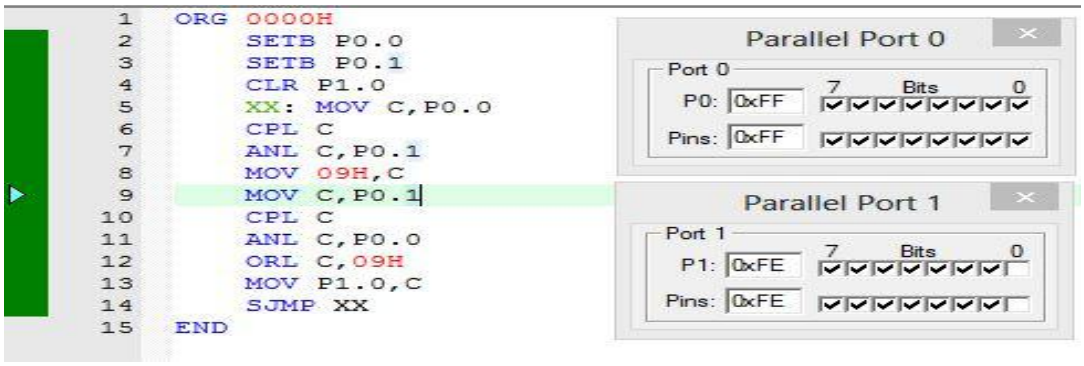


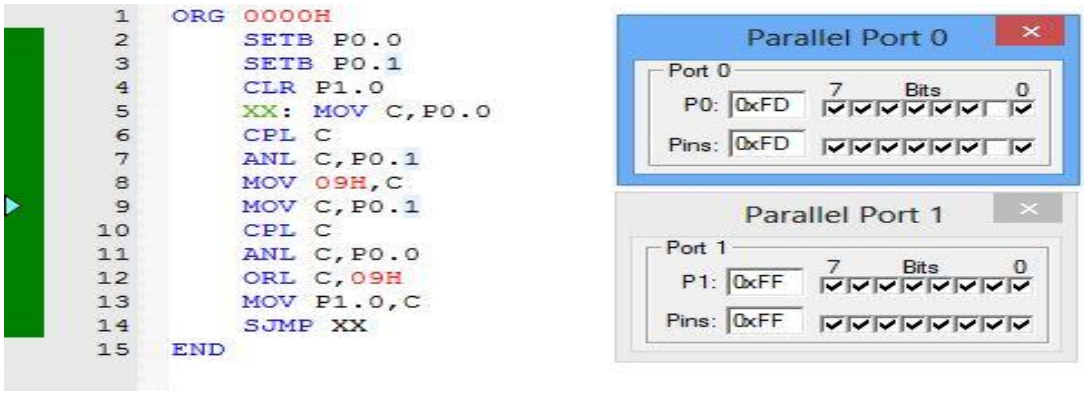

```
1   ORG 0000H
2       SETB P1.0
3       SETB P1.1
4       CLR P2.0
5       XX: MOV C,P1.0
6       ORL C,P1.1
7       MOV 07H,C
8       MOV P2.0,C
9       SJMP XX
10  END
11
```

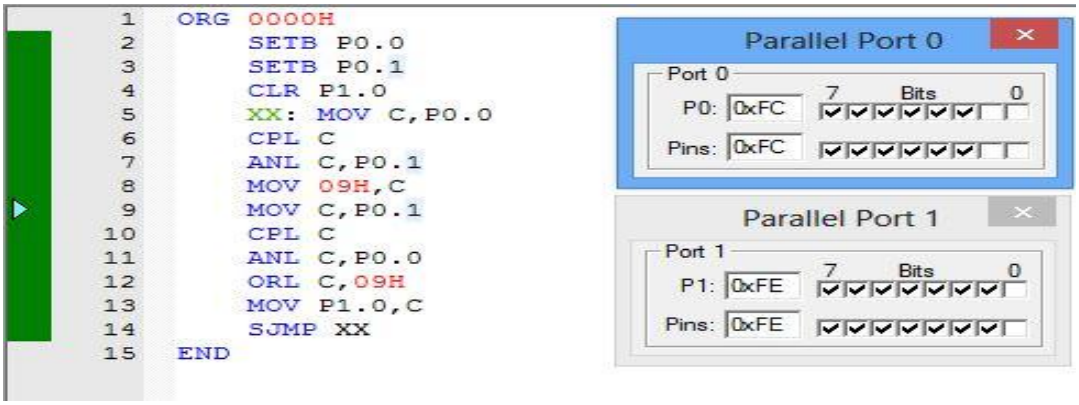


VARUN AGARWAL
16BEC0450

XOR LOGIC:







Code:

AND LOGIC:

```
ORG 0000H
```

VARUN AGARWAL
16BEC0450

```
        SETB P1.0
        SETB P1.1
        CLR P2.0
        XX: MOV C, P1.0
        ANL C, P1.1
        MOV 07H, C
        MOV P2.0, C
        SJMP XX
    END
```

OR LOGIC:

```
    ORG 0000H
        SETB P1.0
        SETB P1.1
        CLR P2.0
        XX: MOV C, P1.0
        ORL C, P1.1
        MOV 07H, C
        MOV P2.0, C
        SJMP XX
    END
```

XOR LOGIC:

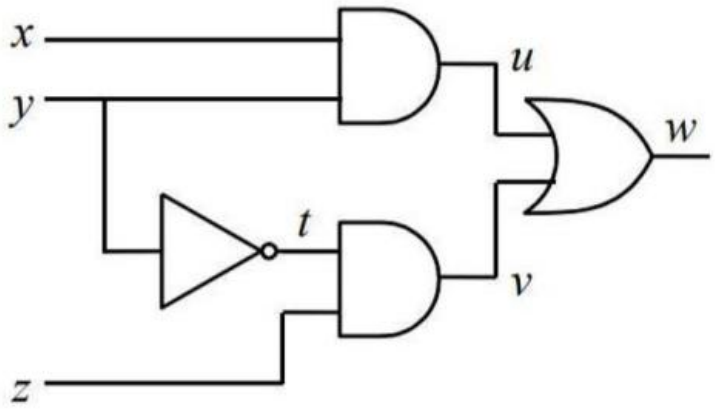
```
    ORG 0000H
        SETB P0.0
        SETB P0.1
        CLR P1.0
        XX: MOV C, P0.0
        CPL C
        ANL C, P0.1
        MOV 09H, C
        MOV C, P0.1
        CPL C
```

```
        ANL C, P0.0
        ORL C, 09H
        MOV P1.0, C
        SJMP XX
    END
```

Draw the truth table for the following circuits and verify the same by writing the 8051 ALP using Keil IDE.

Program-2A

Write an 8051 ALP for the following circuit



The reduced expression of this logic circuit diagram is:

$$x.y + \overline{y}.z$$

and also

$$u = x.y$$

$$t = \overline{y}$$

$$v = \overline{y}.z$$

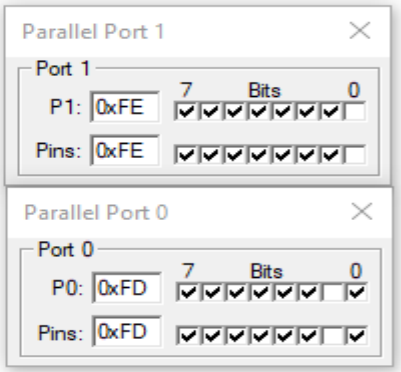
VARUN AGARWAL
16BEC0450

x	y	z	$(x \wedge y) \vee (\neg y \wedge z)$
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	F
F	T	F	F
F	F	T	T
F	F	F	F

Let us assume ‘x’, ‘y’ and ‘z’ to be stored in port 0 bits P0.0, P0.1 and P0.2 respectively. And the outputs ‘u’,‘t’, ‘v’ and ‘w’ in port 1 bits P1.0, P1.1, P1.2 and P1.3 respectively. So the output screenshot and code for the expression will be:

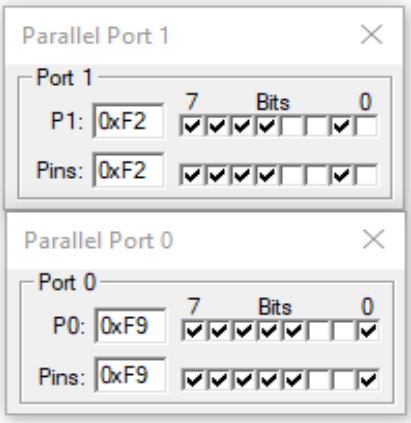
Screenshot:

```
ORG 0000H
     SETB P0.0
     SETB P0.1
     SETB P0.2
     CLR P1.0
     CLR P1.1
     CLR P1.2
     CLR P1.3
     XX: MOV C,P0.0
     ANL C,P0.1
     MOV P1.0,C
     CLR C
     MOV C,P0.1
     CPL C
     MOV P1.1,C
     ANL C,P0.2
     MOV P1.2,C
     ORL C,P1.0
     MOV P1.3,C
     SJMP XX
END
```

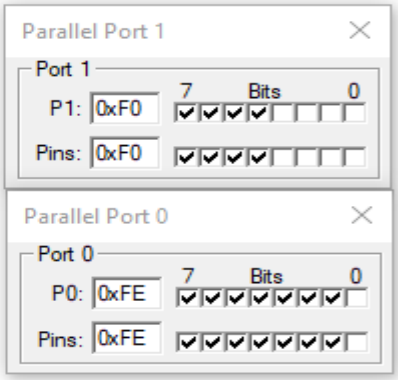


VARUN AGARWAL
16BEC0450

```
1  ORG 0000H
2       SETB P0.0
3       SETB P0.1
4       SETB P0.2
5       CLR P1.0
6       CLR P1.1
7       CLR P1.2
8       CLR P1.3
9       XX: MOV C,P0.0
10      ANL C,P0.1
11      MOV P1.0,C
12      CLR C
13      MOV C,P0.1
14      CPL C
15      MOV P1.1,C
16      ANL C,P0.2
17      MOV P1.2,C
18      ORL C,P1.0
19      MOV P1.3,C
20      SJMP XX
21 END
```

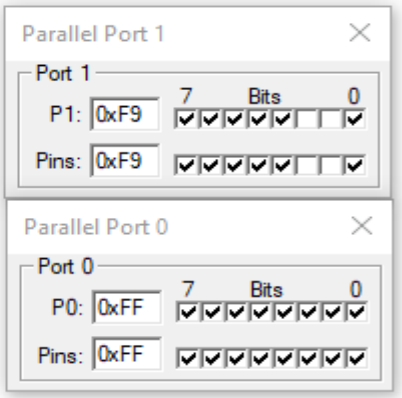



```
1  ORG 0000H
2       SETB P0.0
3       SETB P0.1
4       SETB P0.2
5       CLR P1.0
6       CLR P1.1
7       CLR P1.2
8       CLR P1.3
9       XX: MOV C,P0.0
10      ANL C,P0.1
11      MOV P1.0,C
12      CLR C
13      MOV C,P0.1
14      CPL C
15      MOV P1.1,C
16      ANL C,P0.2
17      MOV P1.2,C
18      ORL C,P1.0
19      MOV P1.3,C
20      SJMP XX
21 END
```

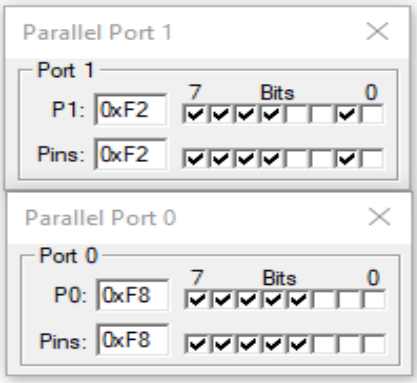


VARUN AGARWAL
16BEC0450

```
ORG 0000H
    SETB P0.0
    SETB P0.1
    SETB P0.2
    CLR P1.0
    CLR P1.1
    CLR P1.2
    CLR P1.3
    XX: MOV C,P0.0
    ANL C,P0.1
    MOV P1.0,C
    CLR C
    MOV C,P0.1
    CPL C
    MOV P1.1,C
    ANL C,P0.2
    MOV P1.2,C
    ORL C,P1.0
    MOV P1.3,C
    SJMP XX
END
```




```
ORG 0000H
    SETB P0.0
    SETB P0.1
    SETB P0.2
    CLR P1.0
    CLR P1.1
    CLR P1.2
    CLR P1.3
    XX: MOV C,P0.0
    ANL C,P0.1
    MOV P1.0,C
    CLR C
    MOV C,P0.1
    CPL C
    MOV P1.1,C
    ANL C,P0.2
    MOV P1.2,C
    ORL C,P1.0
    MOV P1.3,C
    SJMP XX
END
```

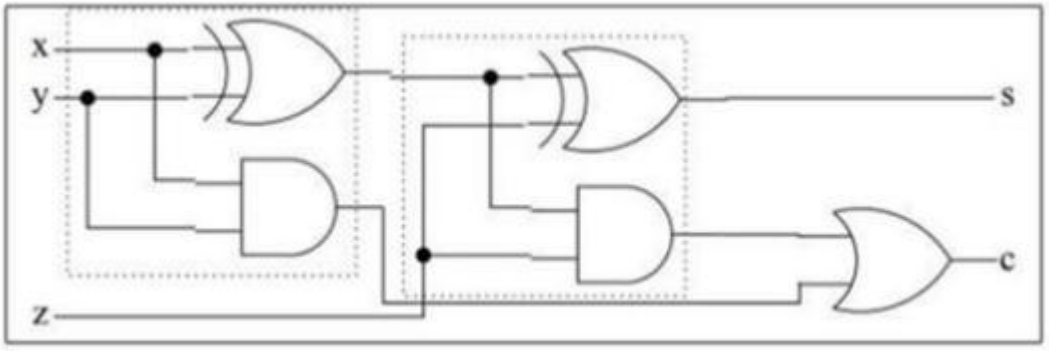


Code:

```
ORG 0000H
    SETB P0.0
    SETB P0.1
    SETB P0.2
    CLR P1.0
    CLR P1.1
    CLR P1.2
    CLR P1.3
    XX: MOV C, P0.0
    ANL C, P0.1
```

```
        MOV P1.0, C
        CLR C
        MOV C, P0.1
        CPL C
        MOV P1.1, C
        ANL C, P0.2
        MOV P1.2, C
        ORL C, P1.0
        MOV P1.3, C
        SJMP XX
END
```

Program-2B



The reduced expression of this logic circuit diagram is: (Full adder)

$$s = x \oplus y \oplus z$$
$$c = x.y + (x \oplus y).z$$

VARUN AGARWAL
16BEC0450

Truth Table of Full Adder

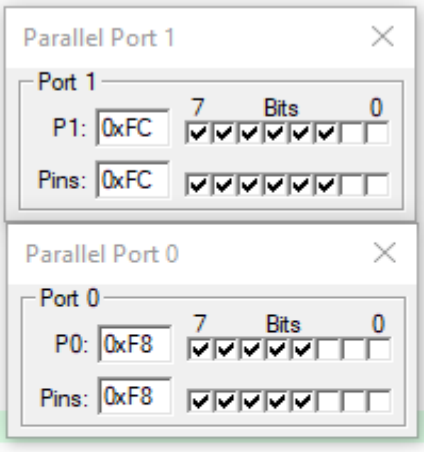
Inputs			Outputs	
X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Let us assume ‘x’, ‘y’ and ‘z’ to be stored in port 0 bits P0.0, P0.1 and P0.2 respectively. And the outputs‘s’ and ‘c’ in port 1 bits P1.0, P1.1 respectively. So the output screenshot and code for the expression will be:

Screenshot:

1. X=0; y=0; z=0 so s=0 and c=0

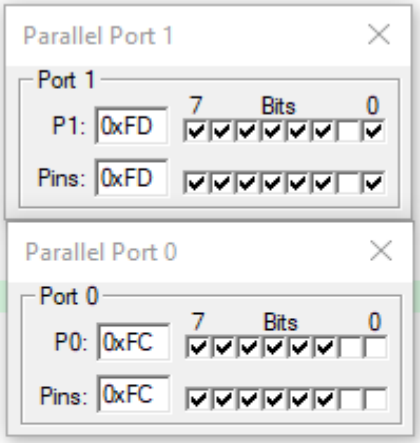
```
 6       CLR P1.1
 7       XX: MOV C,P0.0
 8       CPL C
 9       ANL C,P0.1
10       MOV 07H,C
11       CLR C
12       MOV C,P0.1
13       CPL C
14       ANL C,P0.0
15       ORL C,07H
16       MOV 08H,C
17       CLR C
18       MOV C,P0.2
19       CPL C
20       ANL C,08H
21       MOV 09H,C
22       CLR C
23       MOV C,08H
24       CPL C
25       ANL C,P0.2
26       ORL C,09H
27       MOV P1.0,C
```



VARUN AGARWAL
16BEC0450

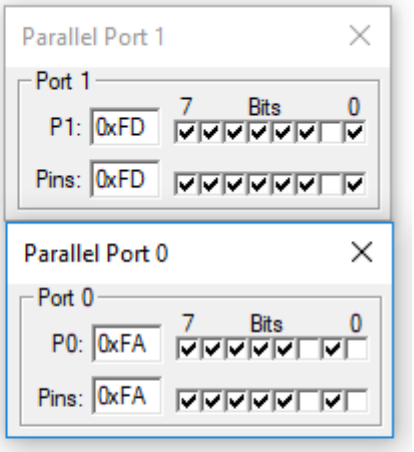
2. x=0;y=0;z=1 so s=1 and c=0

```
 7      XX: MOV C,P0.0
 8      CPL C
 9      ANL C,P0.1
10      MOV 07H,C
11      CLR C
12      MOV C,P0.1
13      CPL C
14      ANL C,P0.0
15      ORL C,07H
16      MOV 08H,C
17      CLR C
18      MOV C,P0.2
19      CPL C
20      ANL C,08H
21      MOV 09H,C
22      CLR C
23      MOV C,08H
24      CPL C
25      ANL C,P0.2
26      ORL C,09H
27      MOV P1.0,C
```



3. x=0;y=1;z=0 so s=1 and c=0

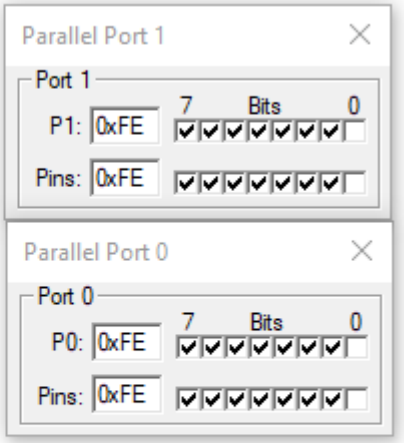
```
23      MOV C,08H
24      CPL C
25      ANL C,P0.2
26      ORL C,09H
27      MOV P1.0,C
28      CLR C
29      MOV C,P0.0
30      ANL C,P0.1
31      MOV 0AH,C
32      CLR C
33      MOV C,P0.0
34      CPL C
35      ANL C,P0.1
36      MOV 0BH,C
37      CLR C
38      MOV C,P0.1
39      CPL C
40      ANL C,P0.0
41      ORL C,0BH
42      ANL C,P0.2
43      ORL C,0AH
44      MOV P1.1,C
```



VARUN AGARWAL
16BEC0450

4. x=0;y=1;z=1 so s=0 and c=1

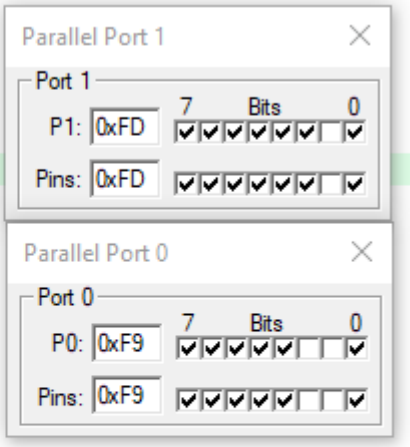
```
 7      XX: MOV C,P0.0
 8      CPL C
 9      ANL C,P0.1
10      MOV 07H,C
11      CLR C
12      MOV C,P0.1
13      CPL C
14      ANL C,P0.0
15      ORL C,07H
16      MOV 08H,C
17      CLR C
18      MOV C,P0.2
19      CPL C
20      ANL C,08H
21      MOV 09H,C
22      CLR C
23      MOV C,08H
24      CPL C
25      ANL C,P0.2
26      ORL C,09H
27      MOV P1.0,C
```



5. x=1;y=0;z=0 so s=1 and c=0

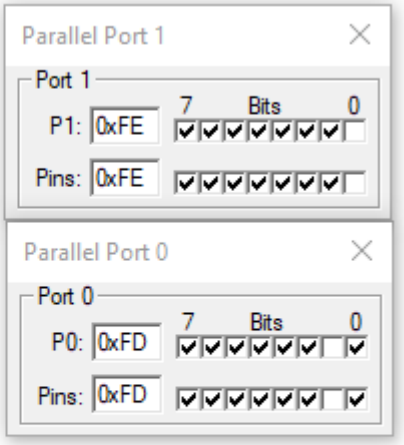
VARUN AGARWAL
16BEC0450

```
23      MOV C,08H
24      CPL C
25      ANL C,P0.2
26      ORL C,09H
27      MOV P1.0,C
28      CLR C
29      MOV C,P0.0
30      ANL C,P0.1
31      MOV 0AH,C
32      CLR C
33      MOV C,P0.0
34      CPL C
35      ANL C,P0.1
36      MOV 0BH,C
37      CLR C
38      MOV C,P0.1
39      CPL C
40      ANL C,P0.0
41      ORL C,0BH
42      ANL C,P0.2
43      ORL C,0AH
44      MOV P1.1,C
```



6. x=1;y=0;z=1 so s=0 and c=1

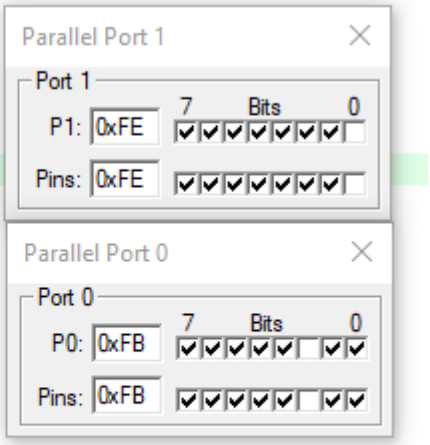
```
 7      XX: MOV C,P0.0
 8      CPL C
 9      ANL C,P0.1
10      MOV 07H,C
11      CLR C
12      MOV C,P0.1
13      CPL C
14      ANL C,P0.0
15      ORL C,07H
16      MOV 08H,C
17      CLR C
18      MOV C,P0.2
19      CPL C
20      ANL C,08H
21      MOV 09H,C
22      CLR C
23      MOV C,08H
24      CPL C
25      ANL C,P0.2
26      ORL C,09H
27      MOV P1.0,C
```



VARUN AGARWAL
16BEC0450

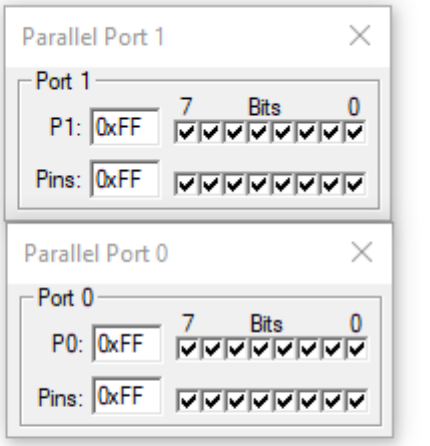
7. x=1;y=1;z=0 so s=0 and c=1

```
 8      CPL C
 9      ANL C,P0.1
10      MOV 07H,C
11      CLR C
12      MOV C,P0.1
13      CPL C
14      ANL C,P0.0
15      ORL C,07H
16      MOV 08H,C
17      CLR C
18      MOV C,P0.2
19      CPL C
20      ANL C,08H
21      MOV 09H,C
22      CLR C
23      MOV C,08H
24      CPL C
25      ANL C,P0.2
26      ORL C,09H
27      MOV P1.0,C
```



8. x=1;y=1;z=1 so s=1 and c=1

```
 7      XX: MOV C,P0.0
 8      CPL C
 9      ANL C,P0.1
10      MOV 07H,C
11      CLR C
12      MOV C,P0.1
13      CPL C
14      ANL C,P0.0
15      ORL C,07H
16      MOV 08H,C
17      CLR C
18      MOV C,P0.2
19      CPL C
20      ANL C,08H
21      MOV 09H,C
22      CLR C
23      MOV C,08H
24      CPL C
25      ANL C,P0.2
26      ORL C,09H
27      MOV P1.0,C
```



Code:

VARUN AGARWAL
16BEC0450

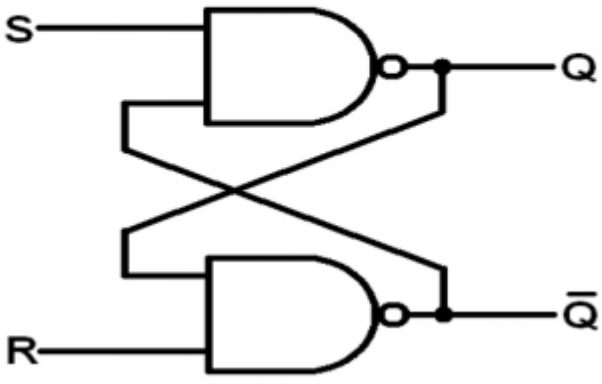
```
ORG 0000H
        SETB P0.0
        SETB P0.1
        SETB P0.2
        CLR P1.0
        CLR P1.1
        XX: MOV C, P0.0
        CPL C
        ANL C, P0.1
        MOV 07H, C
        CLR C
        MOV C, P0.1
        CPL C
        ANL C, P0.0
        ORL C, 07H
        MOV 08H, C
        CLR C
        MOV C, P0.2
        CPL C
        ANL C, 08H
        MOV 09H, C
        CLR C
        MOV C, 08H
        CPL C
        ANL C, P0.2
        ORL C, 09H
        MOV P1.0, C
        CLR C
        MOV C, P0.0
        ANL C, P0.1
        MOV 0AH, C
        CLR C
        MOV C, P0.0
        CPL C
        ANL C, P0.1
        MOV 0BH, C
```

VARUN AGARWAL
16BEC0450

```
        CLR C
        MOV C, P0.1
        CPL C
        ANL C, P0.0
        ORL C, 0BH
        ANL C, P0.2
        ORL C, 0AH
        MOV P1.1, C
        SJMP XX
    END
```

Program-2C:

Write an 8051 ALP for the following circuit.



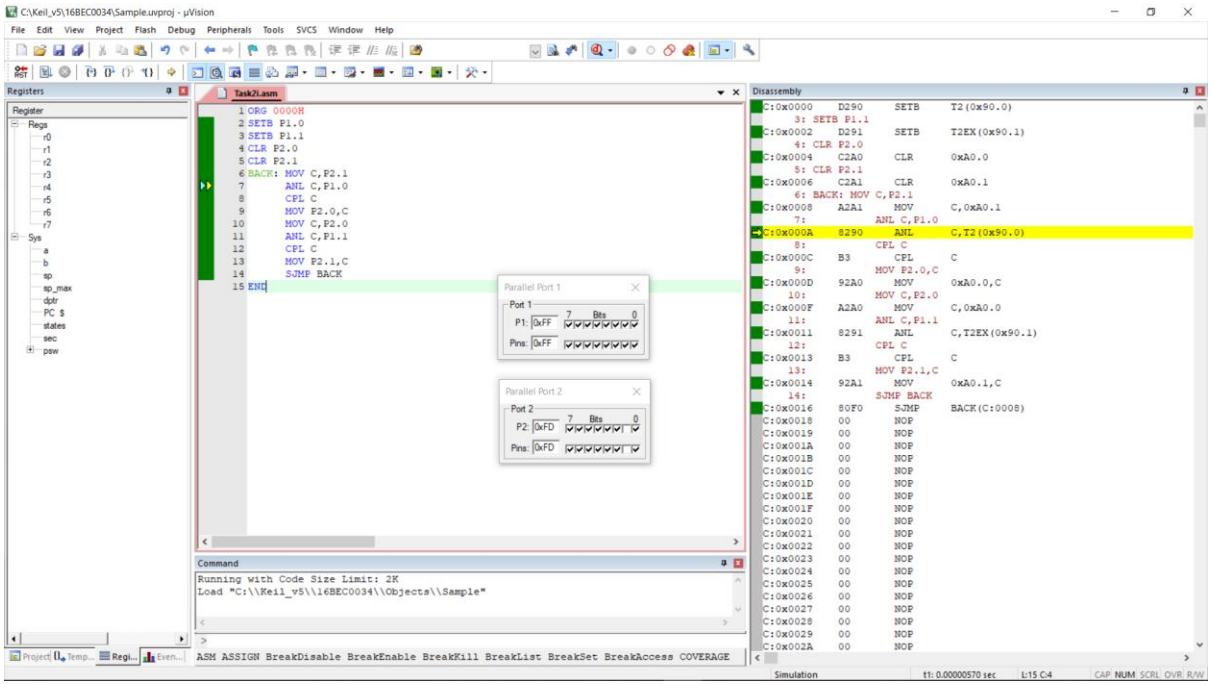
VARUN AGARWAL
16BEC0450

Q	S	R	Q(T+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	UNKNOWN
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	UNKNOWN

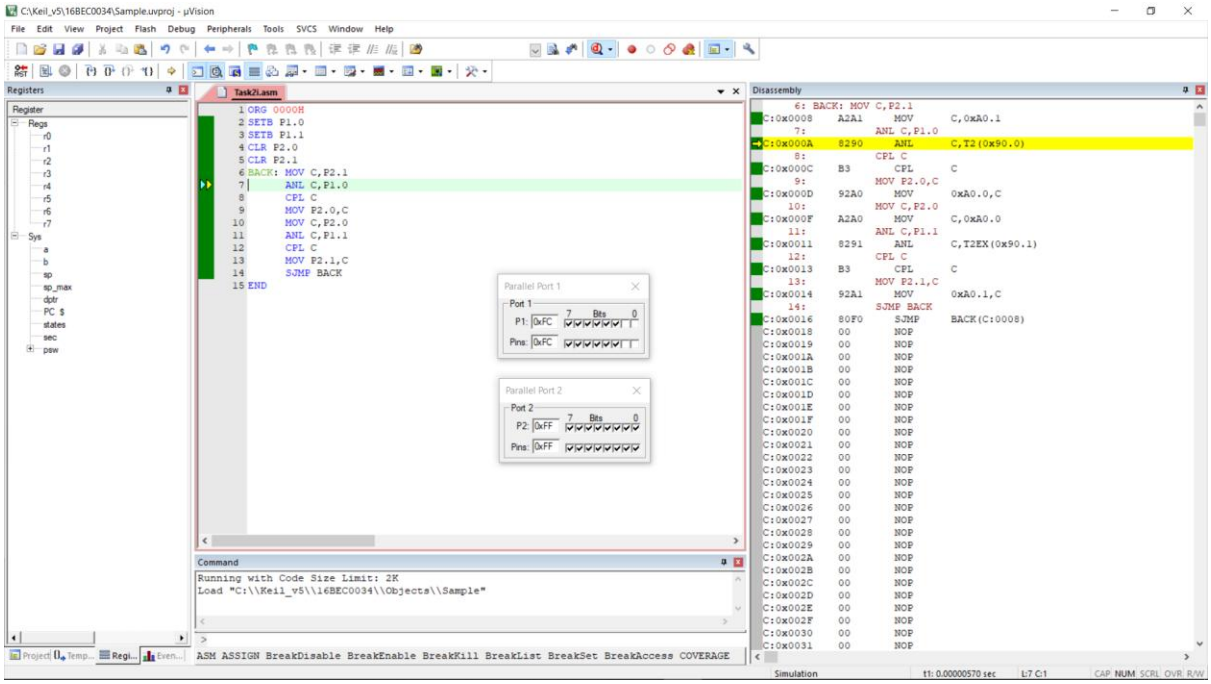
The reduced expression of this logic circuit diagram is: (RS flip flop)

Let us assume ‘S’ and ‘R’ to be stored in port 0 bits P0.0 and P0.1 respectively. And the outputs ‘Q’ and ‘$\bar{Q}$’ in port 1 bits P1.0, P1.1 respectively. So the output screenshot and code for the expression will be:

Screenshot:



VARUN AGARWAL
16BEC0450



Code:

```
ORG 0000H
SETB P1.0
SETB P1.1
CLR P2.0
CLR P2.1
BACK: MOV C, P2.1
        ANL C, P1.0
        CPL C
        MOV P2.0, C
        MOV C, P2.0
        ANL C, P1.1
        CPL C
        MOV P2.1, C
        SJMP BACK
END
```