MICROCONTROLLER AND ITS APPLICATIONS (ECE-3003)

**TASK-2**

Varun Agarwal

16BEC0450

L37+38

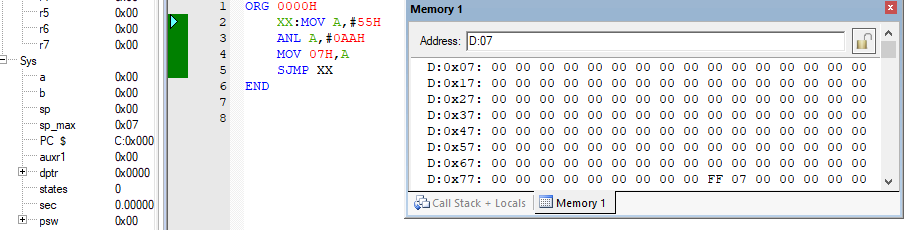
**Program-1**

Write an 8051 ALP for the following

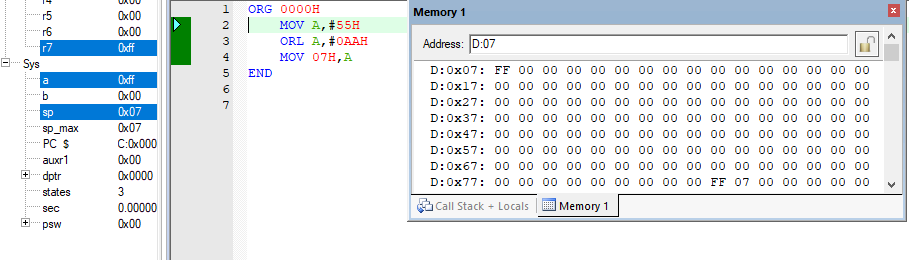
* 55H AND AAH
* 55H OR AAH
* 55H XOR AAH

**Screenshot:**

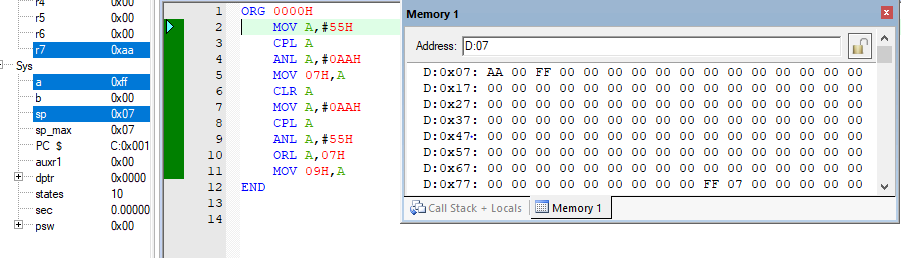
AND:



OR:



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

CLR A

MOV A, #0AAH

CPL A

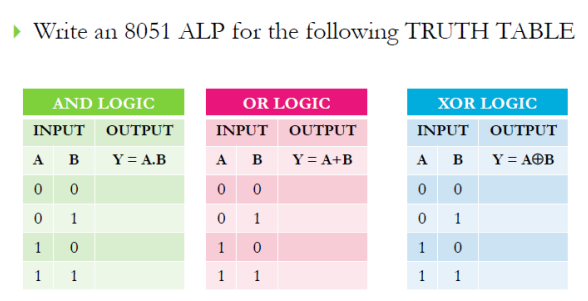
ANL A, #55H

ORL A, 07H

MOV 09H, A

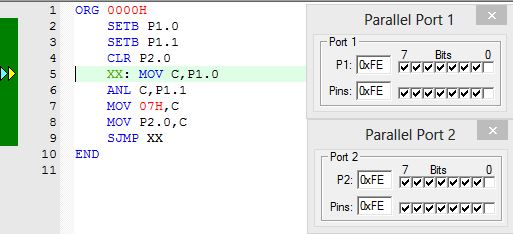
END

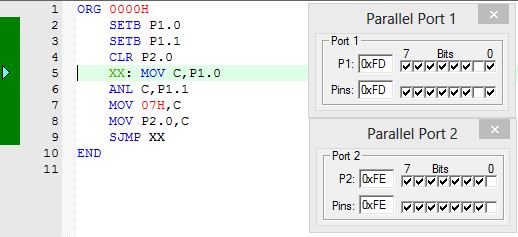
**Program-2**

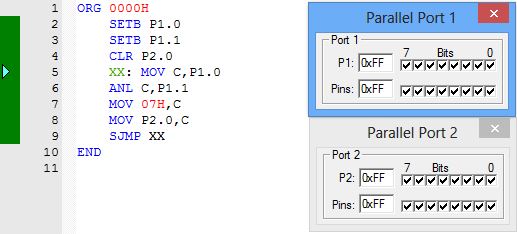


**Screenshot:**

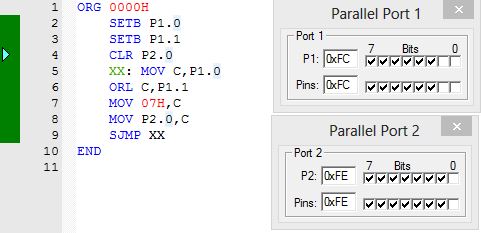
AND LOGIC:

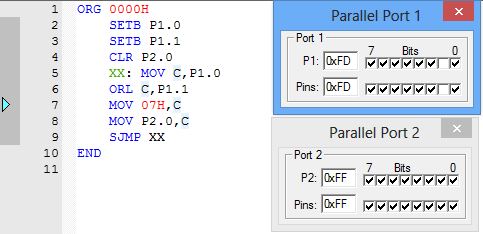




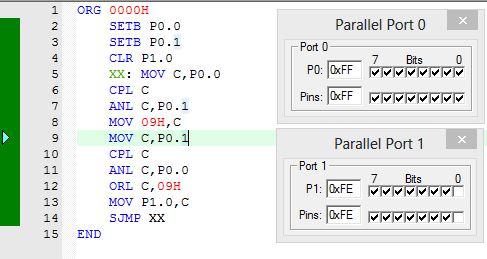


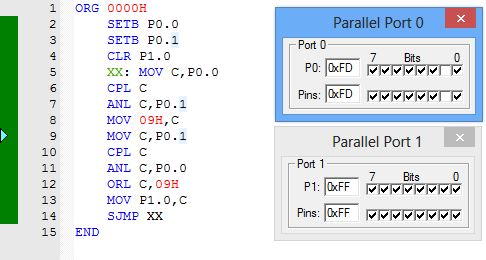
OR LOGIC:

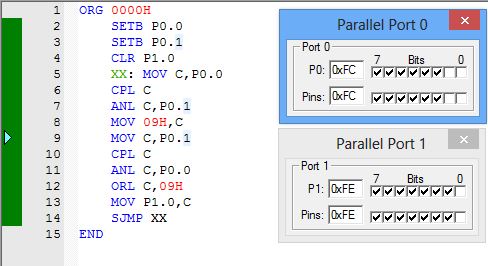




XOR LOGIC:







**Code:**

AND LOGIC:

ORG 0000H

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

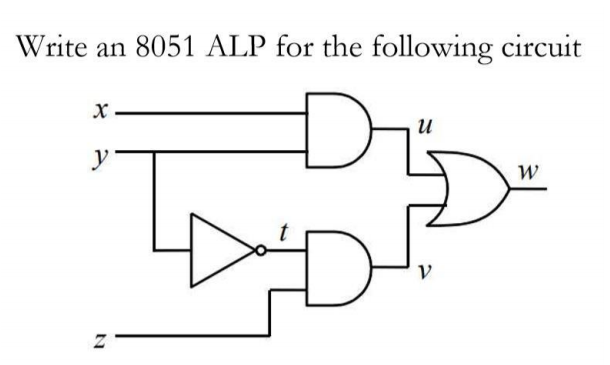
MOV P1.0, C

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**



The reduced expression of this logic circuit diagram is:

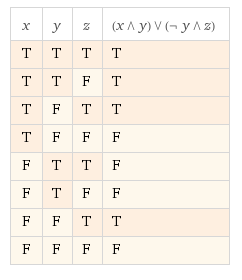


and also



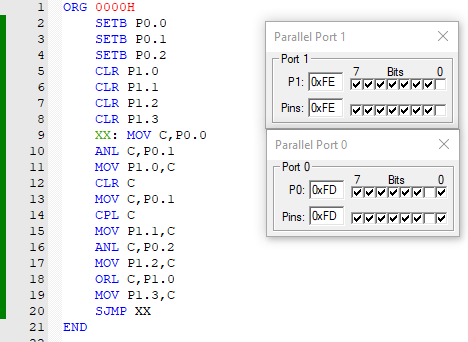


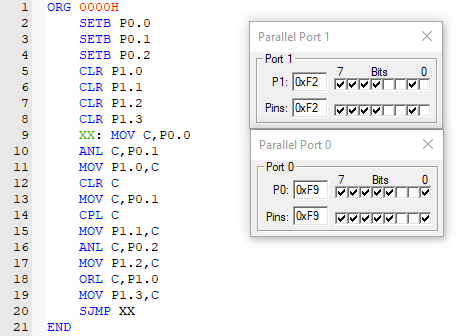


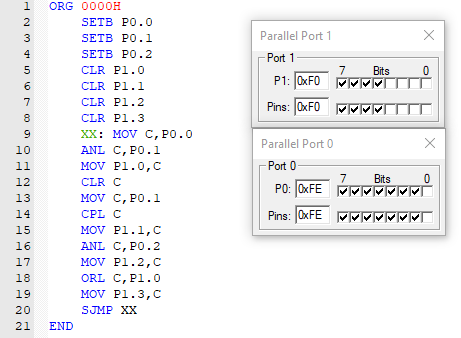


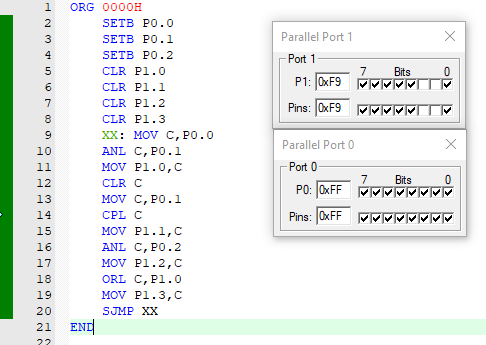
Let us assume ‘x’, ‘y’ and ‘z’ to be stored in port 0 bits P0.0, P0.1 and P0.2 respectively. And the outputs ‘u’,‘t’, ‘v’ and ‘w’ in port 1 bits P1.0, P1.1, P1.2 and P1.3 respectively. So the output screenshot and code for the expression will be:

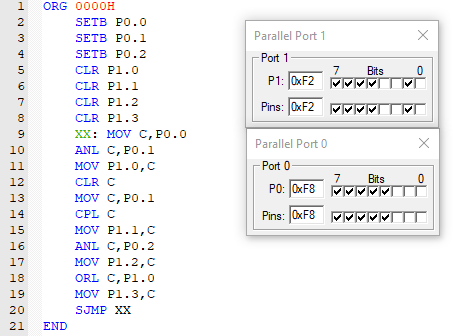
**Screenshot:**











**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

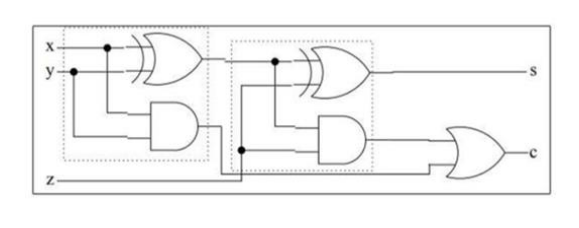
ORL C, P1.0

MOV P1.3, C

SJMP XX

END

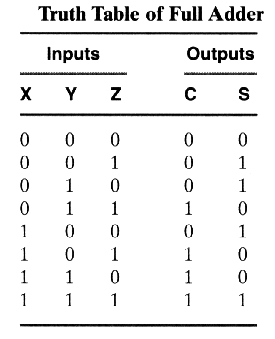
**Program-2B**



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

****

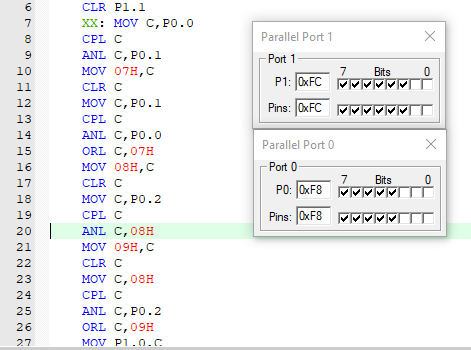




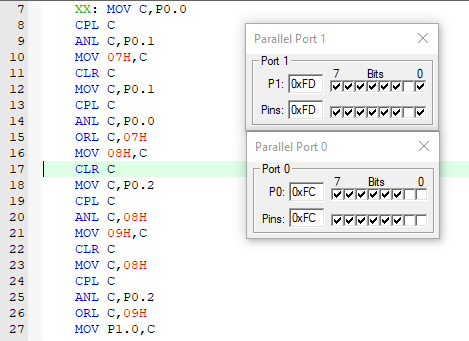
Let us assume ‘x’, ‘y’ and ‘z’ to be stored in port 0 bits P0.0, P0.1 and P0.2 respectively. And the outputs‘s’ and ‘c’ in port 1 bits P1.0, P1.1 respectively. So the output screenshot and code for the expression will be:

**Screenshot:**

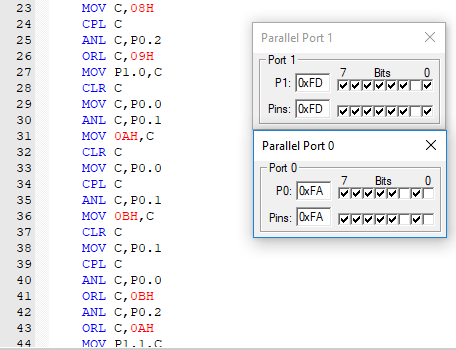
1. X=0; y=0; z=0 so s=0 and c=0



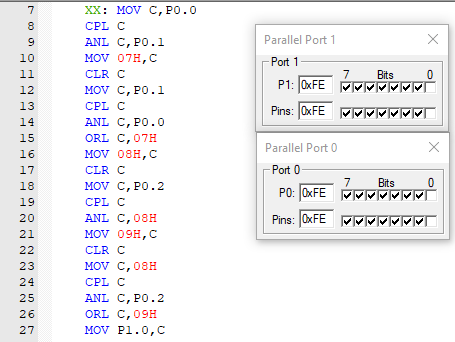
2. x=0;y=0;z=1 so s=1 and c=0



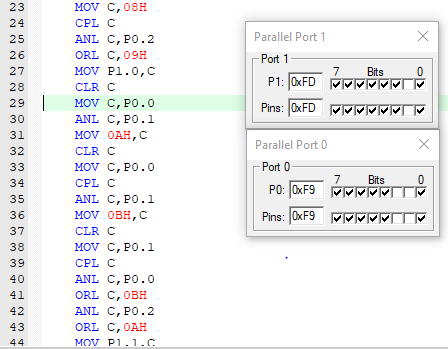
3. x=0;y=1;z=0 so s=1 and c=0



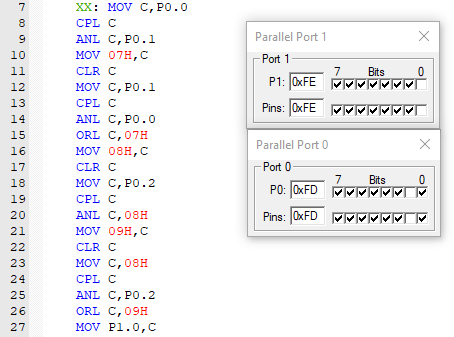
4. x=0;y=1;z=1 so s=0 and c=1



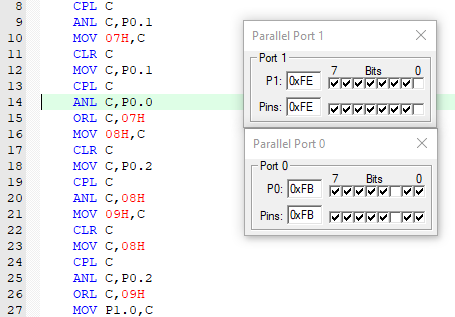
5. x=1;y=0;z=0 so s=1 and c=0



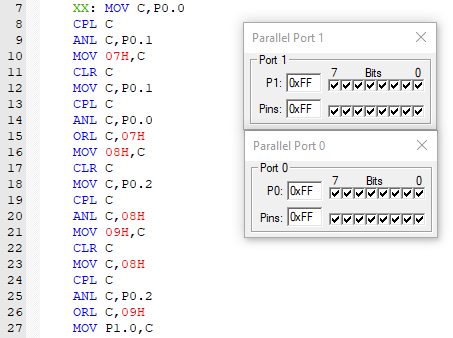
6. x=1;y=0;z=1 so s=0 and c=1



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



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



**Code:**

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

CLR C

MOV C, P0.1

CPL C

ANL C, P0.0

ORL C, 0BH

ANL C, P0.2

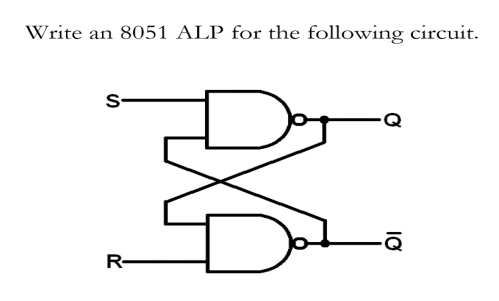
ORL C, 0AH

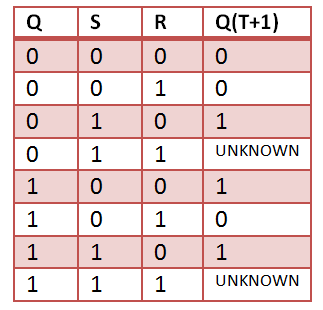
MOV P1.1, C

SJMP XX

END

**Program-2C**:

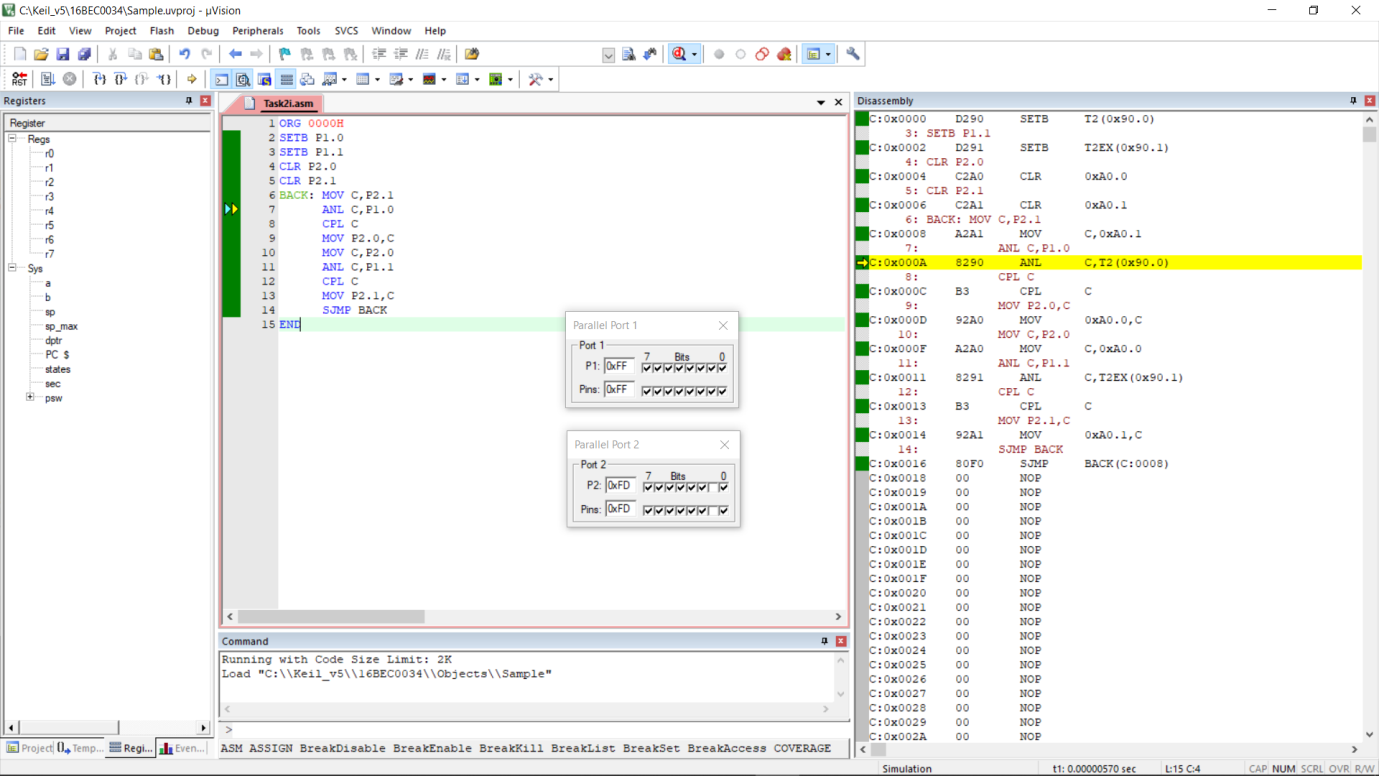


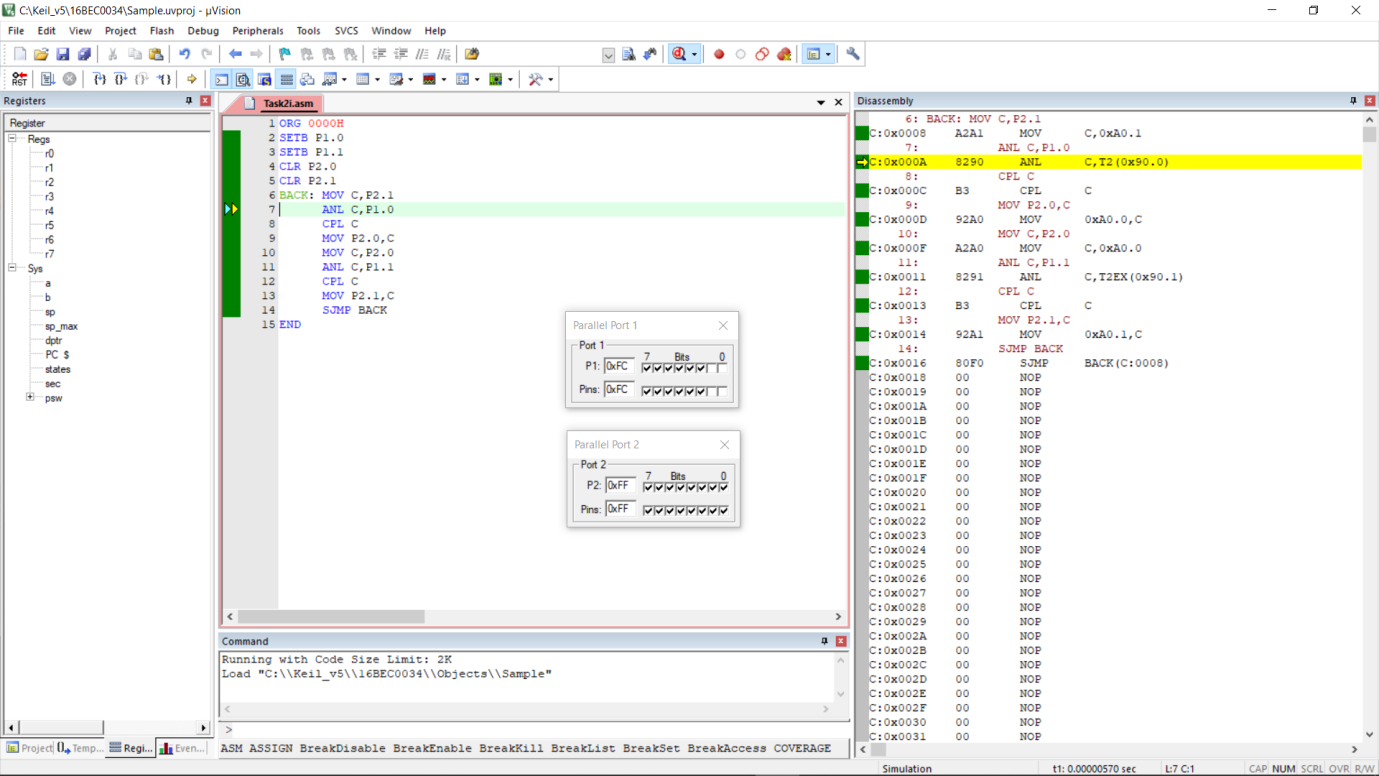


The reduced expression of this logic circuit diagram is: (RS flip flop)

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

**Screenshot:**





**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