LAB # 13

Fall 2020

CSE304L Computer Organization & Architecture Lab

Submitted by: **ASHLEY ALEX JACOB**

Registration No: **18PWCSE1649**

Class Section: **A**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: _____

Submitted to:

Engr. Amaad Khalil

March 19, 2021 (Friday)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Lab Tasks:

Task:

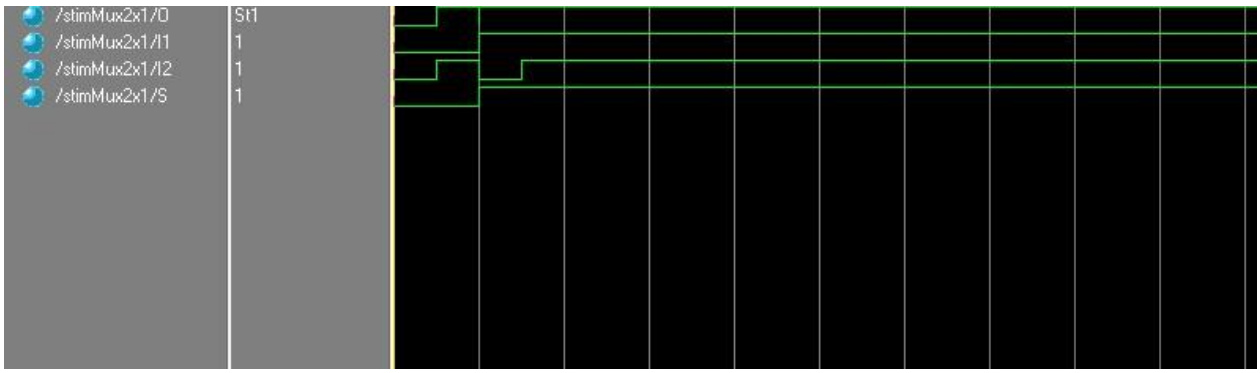
Write a Verilog code for 2x1 MUX using Dataflow Level modeling.

Code:

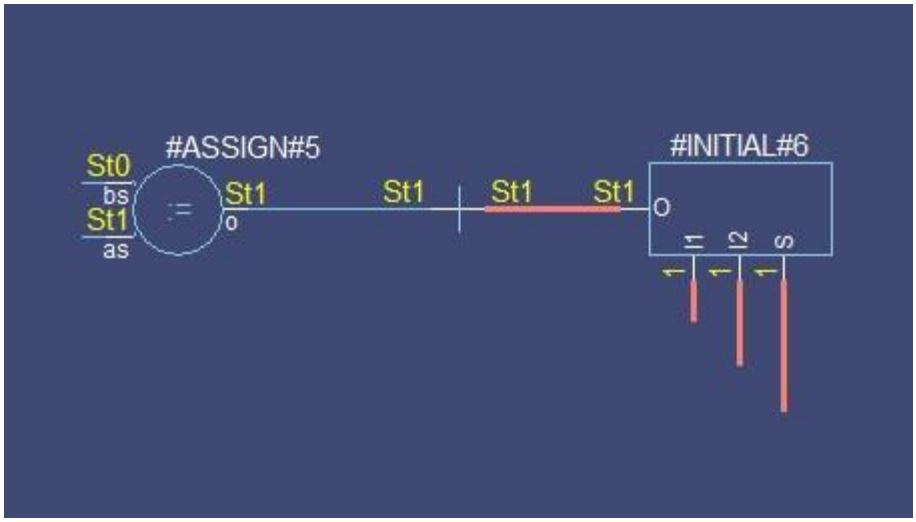
```
module mux2x1(output o,input a,b,s);

wire as = a&s;

wire s_= ~s;

wire bs = b&s_;

assign o = as | bs;



endmodule



module stimMux2x1();

wire O;

reg I1,I2,S;

mux2x1 m(O,I1,I2,S);



initial

begin
```

```
$display("S     I1      I2      Y");

I1 = 0;

I2 = 0;

S = 0;

#1 $display("%b     %b      %b      %b",S,I1,I2,O);

I1 = 0;

I2 = 1;

S = 0;

#1 $display("%b     %b      %b      %b",S,I1,I2,O);

I1 = 1;

I2 = 0;

S = 1;

#1 $display("%b     %b      %b      %b",S,I1,I2,O);

I1 = 1;

I2 = 1;

S = 1;

#1 $display("%b     %b      %b      %b",S,I1,I2,O);

end
```

```
endmodule
```




```
# Loading ash.stimMux2x1
# Loading ash.mux2x1
run
# S   I1   I2   Y
# 0   0    0    0
# 0   0    1    1
# 1   1    0    1
# 1   1    1    1

VSIM 38>
```



Task:

Write a Verilog code for 2x4 Decoder using Dataflow Level modeling.

Code:

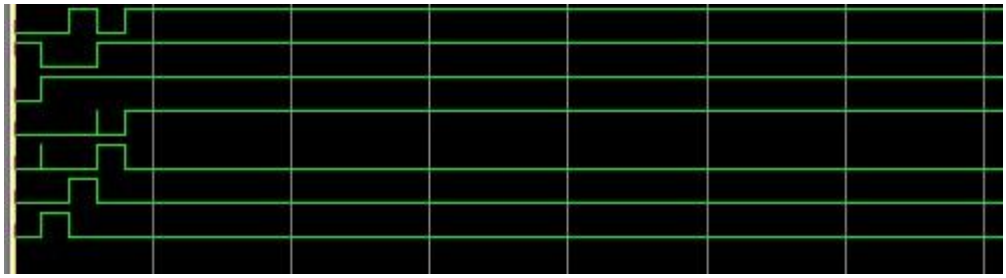
```
module dec2x4(input a1,a0,e,output y3,y2,y1,y0);
```

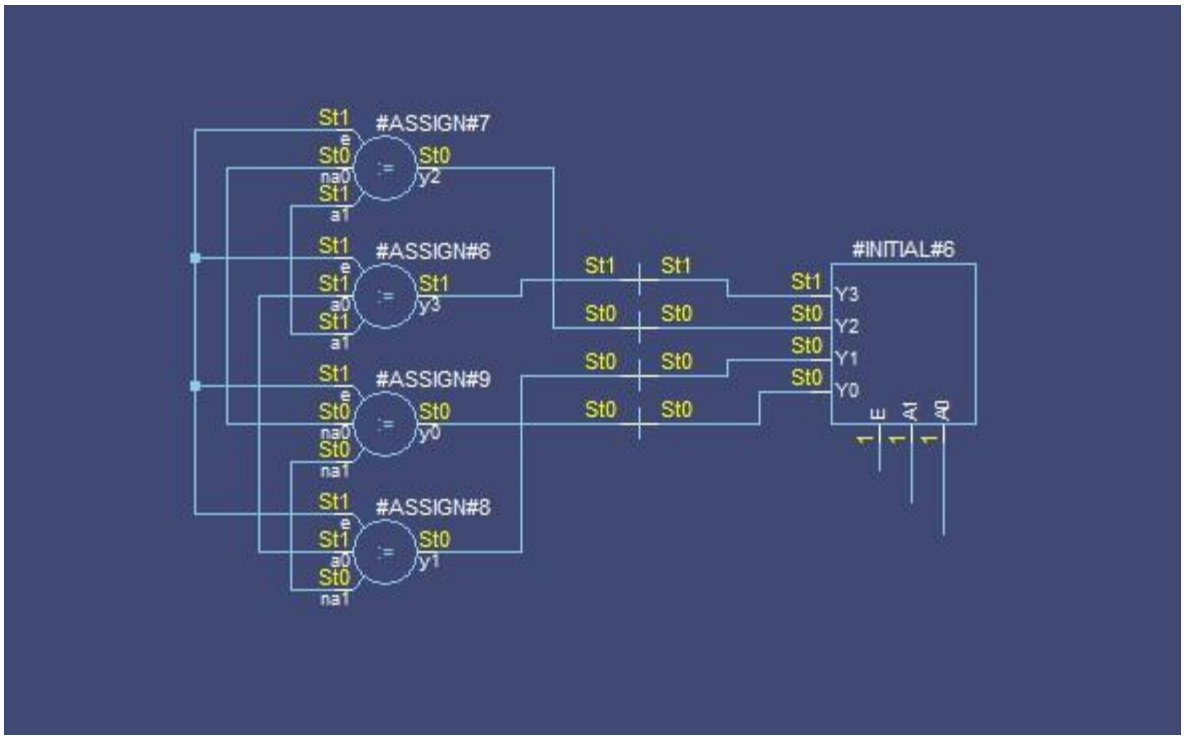
```
wire na0 = ~a0;

wire na1 = ~a1;

assign y3 = a1&a0&e;

assign y2 = a1&na0&e;

assign y1 = na1&a0&e;

assign y0 = na1&na0&e;



endmodule




module stimDec2x4();

reg A0,A1,E;

wire Y3,Y2,Y1,Y0;

dec2x4 d(A1,A0,E,Y3,Y2,Y1,Y0);

initial

begin

$display("E         A1 A0      Y3 Y2 Y1 Y0");

E=0;
```

```
A1=1;

A0=0;

#1 $display("%b           %b         %b      %b          %b     %b
%b",E,A1,A0,Y3,Y2,Y1,Y0);

E=1;

A1=0;

A0=0;

#1 $display("%b           %b         %b      %b          %b     %b
%b",E,A1,A0,Y3,Y2,Y1,Y0);

E=1;

A1=0;

A0=1;

#1 $display("%b           %b         %b      %b          %b     %b
%b",E,A1,A0,Y3,Y2,Y1,Y0);

E=1;

A1=1;

A0=0;

#1 $display("%b           %b         %b      %b          %b     %b
%b",E,A1,A0,Y3,Y2,Y1,Y0);

E=1;

A1=1;
```

```
A0=1;

#1 $display("%b           %b         %b       %b          %b     %b
%b",E,A1,A0,Y3,Y2,Y1,Y0);

end

endmodule
```

```
# vsim ash.stimDec2x4
# Loading ash.stimDec2x4
# Loading ash.dec2x4
run
# E  A1 A0   Y3 Y2 Y1 Y0
# 0  1   0    0   0   0   0
# 1  0   0    0   0   0   1
# 1  0   1    0   0   1   0
# 1  1   0    0   1   0   0
# 1  1   1    1   0   0   0

VSIM 10>
```





Task:

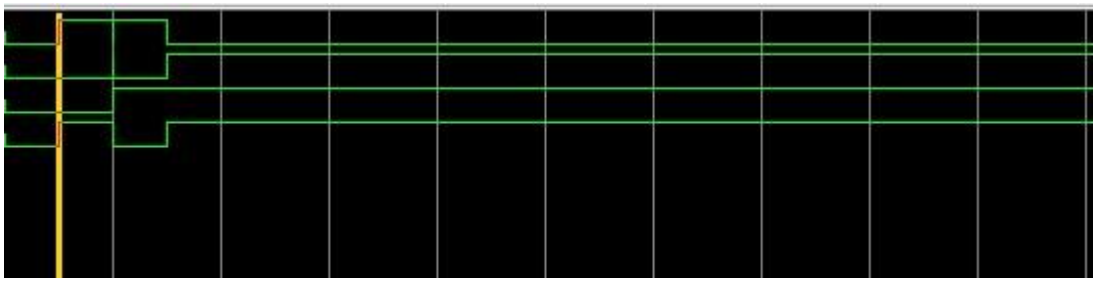
Write a Verilog code for Half Adder using Dataflow Level modeling.

Code:

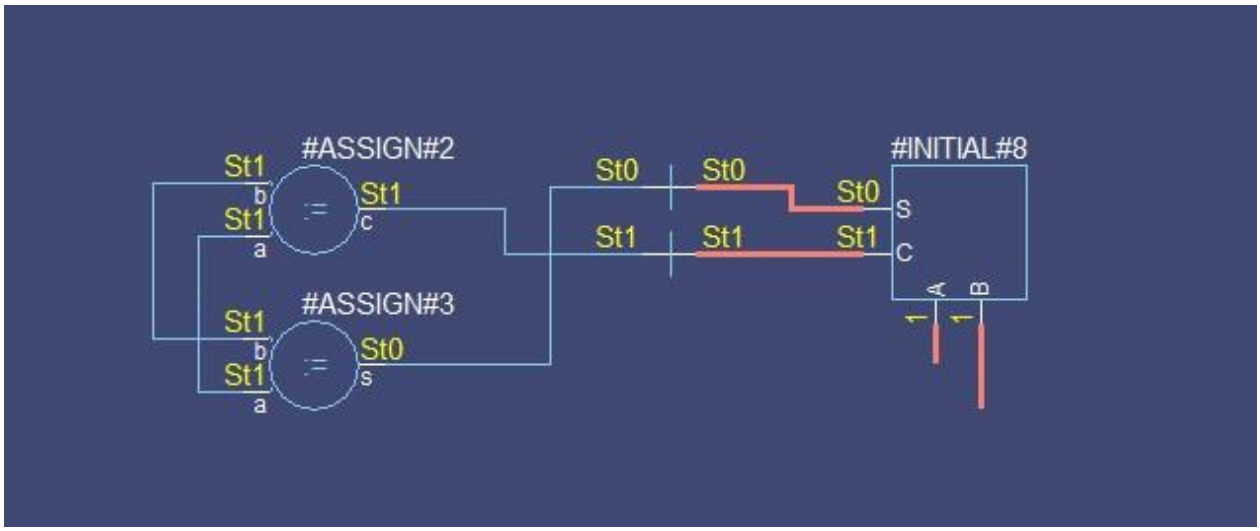
```
module halfAdd(output s,c,input a,b);

assign c = a&b;

assign s = a ^ b;

endmodule



module simHalfAdd();

wire S,C;

reg A,B;
```

```
halfAdd ha(S,C,A,B);



initial

begin



$display("A     B       S       C");

A=0;

B=0;

#1 $display("%b     %b      %b      %b",A,B,S,C);



A=0;

B=1;

#1 $display("%b     %b      %b      %b",A,B,S,C);



A=1;

B=0;

#1 $display("%b     %b      %b      %b",A,B,S,C);
```

```
A=1;

B=1;

#1 $display("%b     %b      %b      %b",A,B,S,C);



end

endmodule
```



```
vsim ash.simHalfAdd
# vsim ash.simHalfAdd
# Loading ash.simHalfAdd
# Loading ash.halfAdd
run
# A   B    S    C
# 0   0    0    0
# 0   1    1    0
# 1   0    1    0
# 1   1    0    1
destroy .wave

VSIM 14>
```



Task:

Write a Verilog code for Full Adder using Dataflow Level modeling.

Code:

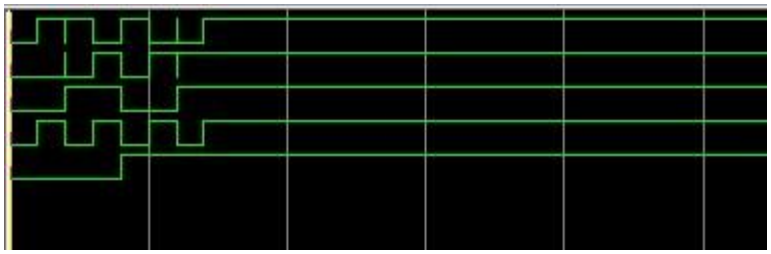
```
module fullAdd( output s,cOut,input a,b,cIn);

wire n1 = a ^ b;

assign s = cIn ^ n1;

wire n2  = n1 & cIn;

wire n3 = a & b;

assign cOut = n2 | n3;

endmodule




module simFullAdd();
```

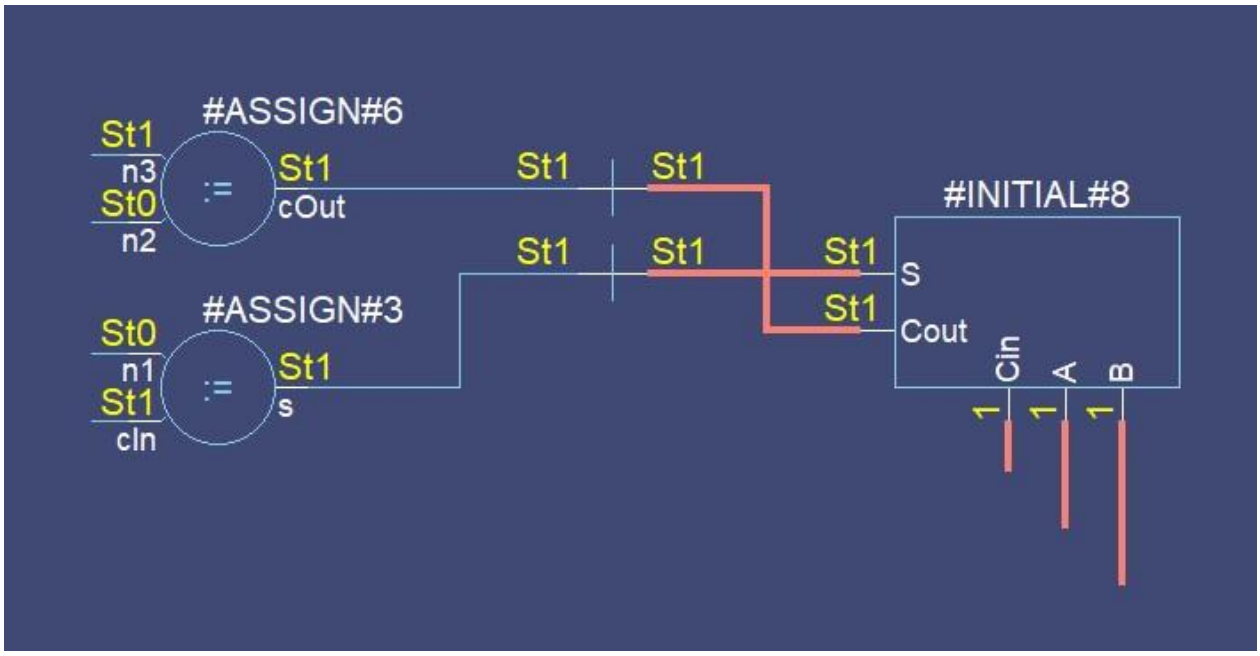
```
wire S,Cout;

reg A,B,Cin;



fullAdd fa(S,Cout,A,B,Cin);



initial

begin



$display("Cin    A      B    S     Cout");

Cin=0;

A=0;

B=0;

#1 $display("%b     %b      %b      %b      %b",Cin,A,B,S,Cout);

Cin=0;

A=0;

B=1;

#1 $display("%b     %b      %b      %b      %b",Cin,A,B,S,Cout);

Cin=0;
```

```
A=1;

B=0;

#1 $display("%b     %b      %b      %b      %b",Cin,A,B,S,Cout);

Cin=0;

A=1;

B=1;

#1 $display("%b     %b      %b      %b      %b",Cin,A,B,S,Cout);

Cin=1;

A=0;

B=0;

#1 $display("%b     %b      %b      %b      %b",Cin,A,B,S,Cout);

Cin=1;

A=0;

B=1;

#1 $display("%b     %b      %b      %b      %b",Cin,A,B,S,Cout);

Cin=1;

A=1;

B=0;
```

```
#1 $display("%b      %b       %b       %b       %b",Cin,A,B,S,Cout);

Cin=1;

A=1;

B=1;

#1 $display("%b      %b       %b       %b       %b",Cin,A,B,S,Cout);




end

endmodule
```

```
# vsim ash.simFullAdd
# Loading ash.simFullAdd
# Loading ash.fullAdd
run
# Cin   A    B   S    Cout
# 0    0    0    0    0
# 0    0    1    1    0
# 0    1    0    1    0
# 0    1    1    0    1
# 1    0    0    1    0
# 1    0    1    0    1
# 1    1    0    0    1
# 1    1    1    1    1

VSIM 19>
```





Task:

Write a Verilog code for Half Subtractor using Dataflow Level modeling.

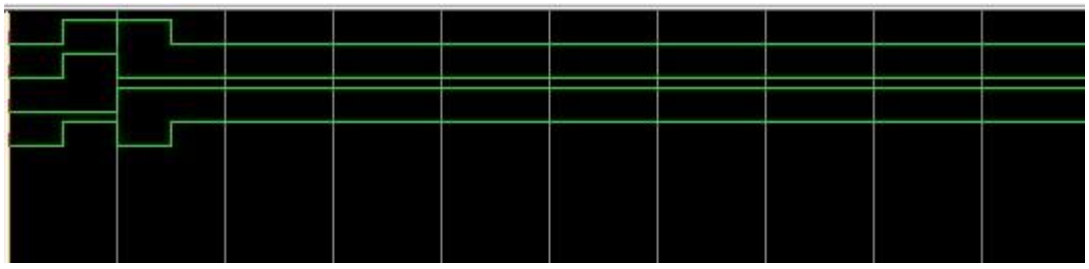
Code:

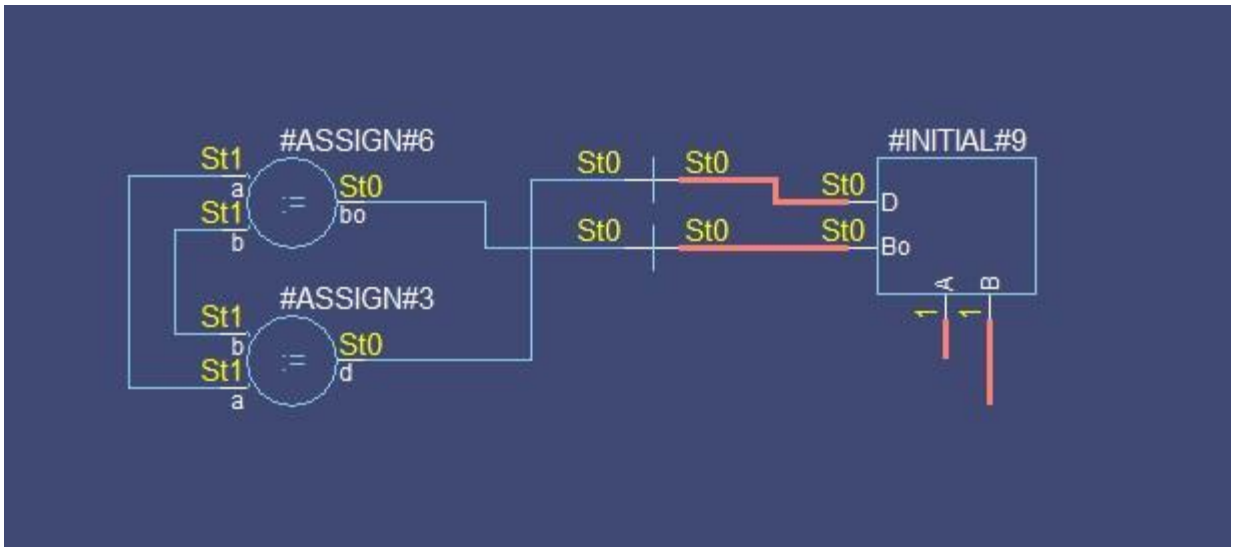
```
module halfSub(output d,bo,input a,b);

assign d = a ^ b;

assign bo  = b & (~a);

endmodule



module simHalfSub();

wire D,Bo;

reg A,B;
```

```
halfSub hs(D,Bo,A,B);



initial

begin



$display("A     B       D       B");

A=0;

B=0;

#1 $display("%b     %b      %b      %b",A,B,D,Bo);



A=0;

B=1;

#1 $display("%b     %b      %b      %b",A,B,D,Bo);



A=1;

B=0;

#1 $display("%b     %b      %b      %b",A,B,D,Bo);
```

```
A=1;

B=1;

#1 $display("%b     %b      %b      %b",A,B,D,Bo);



end

endmodule
```

```
# vsim ash.simHalfSub
# Loading ash.simHalfSub
# Loading ash.halfSub
run
#A   B    D    B
#0   0    0    0
#0   1    1    1
#1   0    1    0
#1   1    0    0

VSIM 16>
```




#ASSIGN#6
St1
a
St1
b
:=
St0
bo
#ASSIGN#3
St1
b
St1
a
:=
St0
d
St0
St0
St0
St0
St0
St0
#INITIAL#9
D
Bo
A
B
1
1