Introduction to Digital Logic
EECS/CSE 31L

Assignment 2: 1-Bit ALU

Prepared by: Jack Melcher
Student ID: 67574625

EECS Department
Henry Samueli School of Engineering
University of California, Irvine

January 28, 2015

1 Block Description

Take in two 1-bit values (a and b) and 4 select values in order to perform a variety of operations

2 Input/Output Port Description

Port name	Port size	Port Type	Description
A	1	IN	Value of first input
B	1	IN	Value of second input
Cin	1	INOUT	The carry in value
opsel	2 downto 0	IN	Determine which operation to perform in units
mode	1	OUT	Determine whether the output is from arithmetic operation or logic operation
output	1	OUT	The output value of operation
cout	1	OUT	The carry out value

3 Design Schematics

Truth Tables:

ALU Operations

Mode	**OPSEL**	**Micro-operation**	**Description**
0	000	a+b	Add
0	001	a+b'	Sub with borrowed carry
0	010	a	Move
0	011	a+b'+1	Sub
0	100	a+1	increment
0	101	a-1	decrement
0	110	a+b+1	Add & Increment

1	000	a AND b	Bit-wise AND
1	001	a OR b	Bit-wise OR
1	010	a XOR b	Bit-wise Exclusive OR
1	011	a'	Compliment
1	101	shl a	1 Bit Shift Left

Cin table

Opsel	cin	**Description**
000	0	Add
001	1	Sub with borrowed carry
010	0	Move
011	1	Sub
100	1	increment
101	0	decrement
110	1	Add & Increment

Full Adder

In_0	In_1	Cin	Cout	Sum
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

Two's Complement Full Adder/Subtractor

In_0	In_1	Cin	Cout	Sum
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	1	0
0	1	1	0	1
1	0	1	1	1
1	1	1	1	0

Boolean Expressions:

Full Adder

Sum = $(A \oplus B) \oplus Cin$

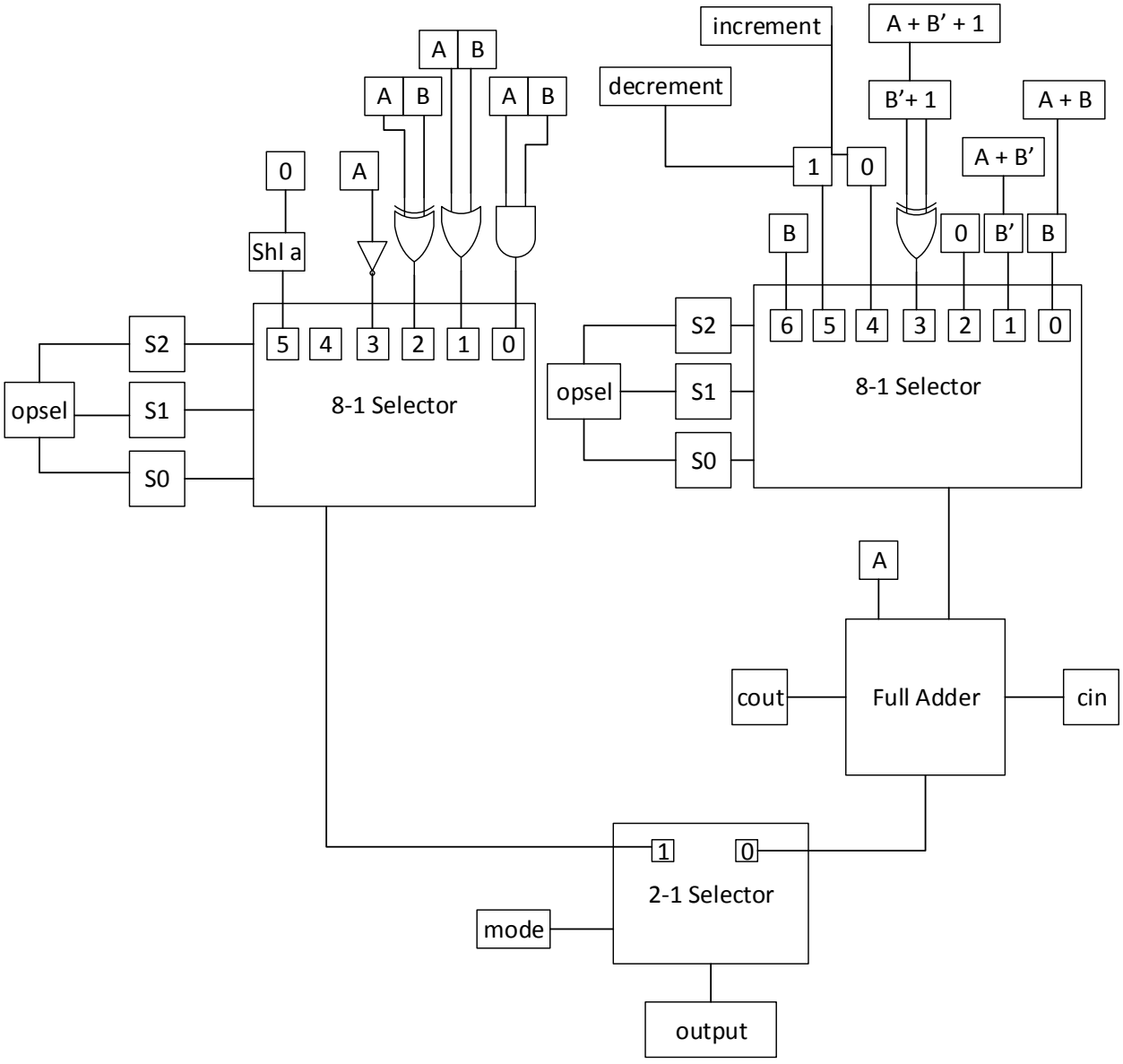
Cout = $(A)(B) + (Cin)(A \oplus B)$

Two's Complement Full Adder

Sum = $(A \oplus (B \oplus Cin)) \oplus Cin$

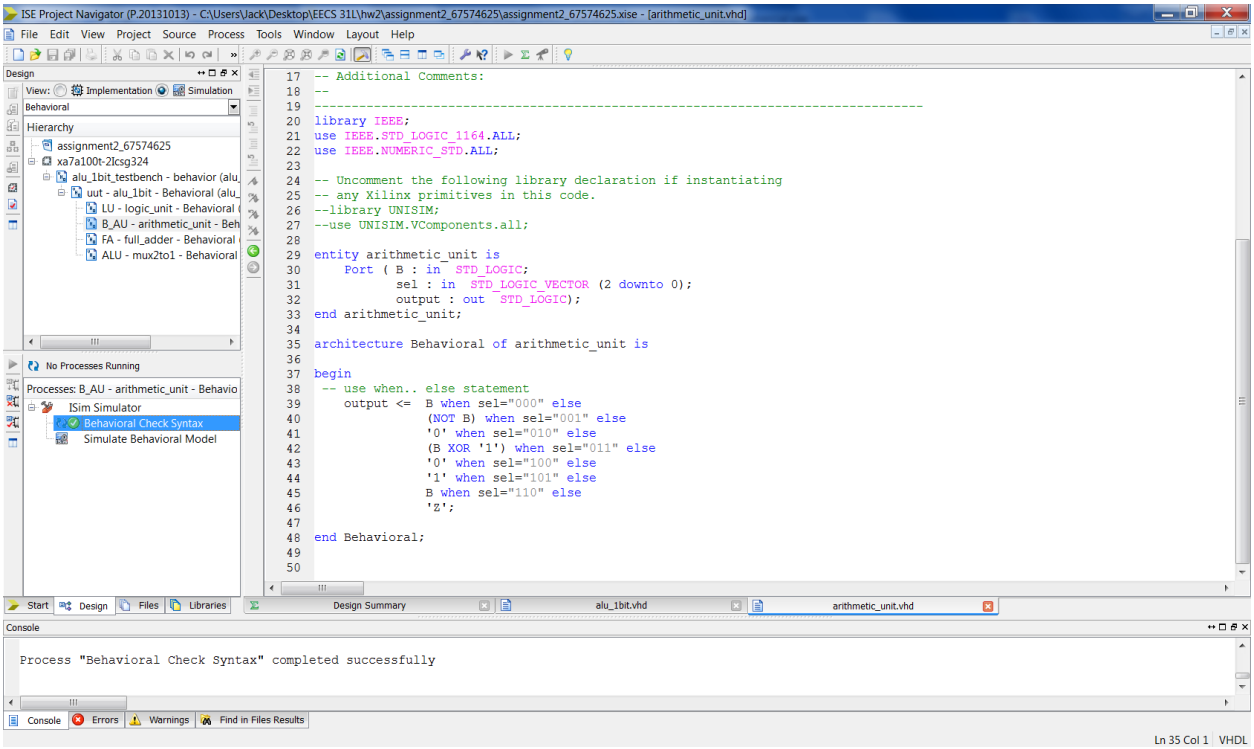
Cout = $(A)(B \oplus Cin) + ((Cin)(A \oplus (B \oplus Cin)))$

Design:


A
B
A
B
A
B
0
A
Shl a
5
4
3
2
1
0
S2
S1
S0
opsel
8-1 Selector
increment
A + B' + 1
decrement
B'+ 1
A + B
1
0
A + B'
B
0
B'
B
6
5
4
3
2
1
0
S2
S1
S0
opsel
8-1 Selector
A
cout
Full Adder
cin
1
0
2-1 Selector
mode
output

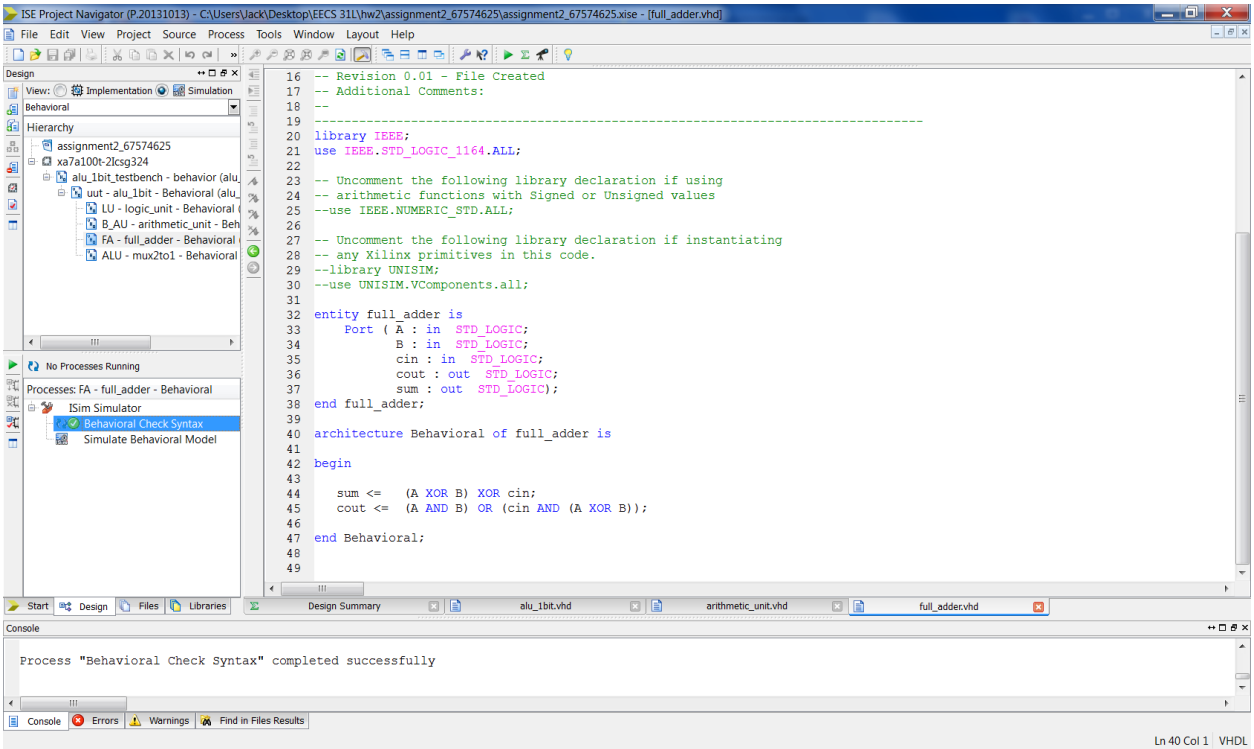
4 Compilation

Arithmetic unit



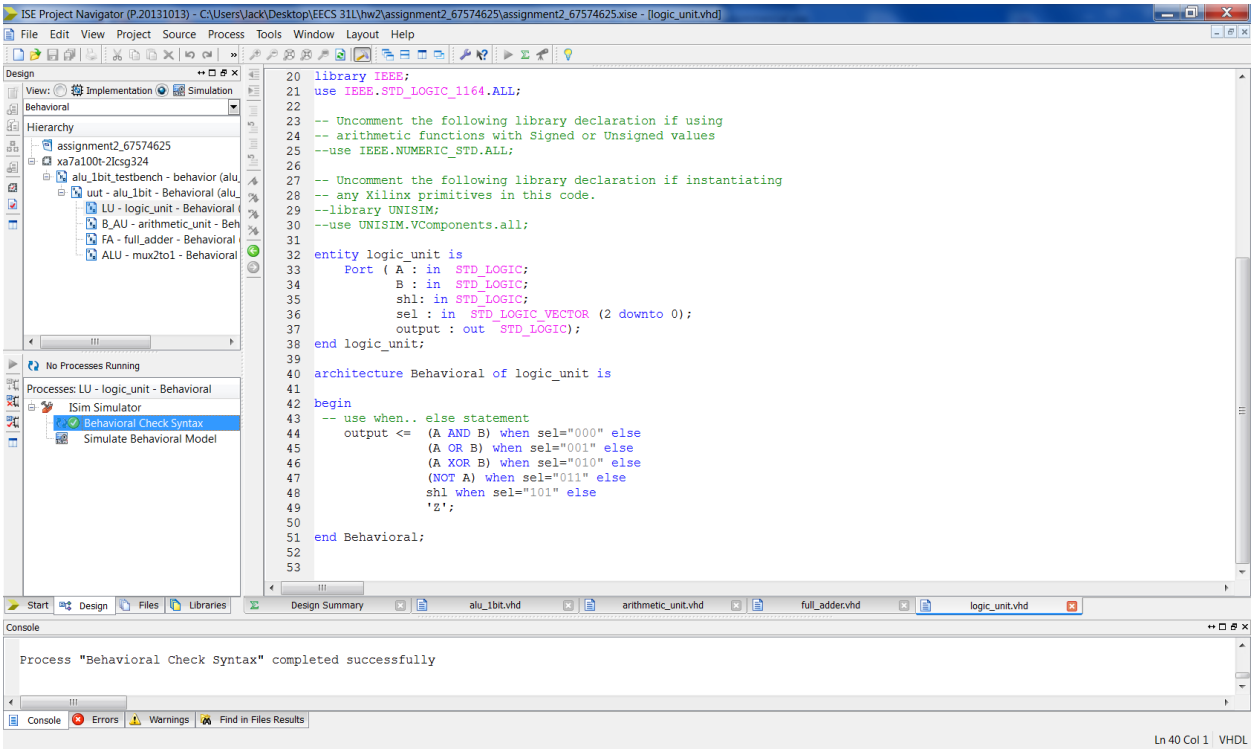

```
17  -- Additional Comments:
18  --
19  ----------------------------------------------------------------------------------
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22  use IEEE.NUMERIC_STD.ALL;
23
24  -- Uncomment the following library declaration if instantiating
25  -- any Xilinx primitives in this code.
26  --library UNISIM;
27  --use UNISIM.VComponents.all;
28
29  entity arithmetic_unit is
30      Port ( B : in  STD_LOGIC;
31             sel : in  STD_LOGIC_VECTOR (2 downto 0);
32             output : out  STD_LOGIC);
33  end arithmetic_unit;
34
35  architecture Behavioral of arithmetic_unit is
36
37  begin
38   -- use when.. else statement
39      output <=  B when sel="000" else
40                 (NOT B) when sel="001" else
41                 '0' when sel="010" else
42                 (B XOR '1') when sel="011" else
43                 '0' when sel="100" else
44                 '1' when sel="101" else
45                 B when sel="110" else
46                 'Z';
47
48  end Behavioral;
49
50
```

Full Adder

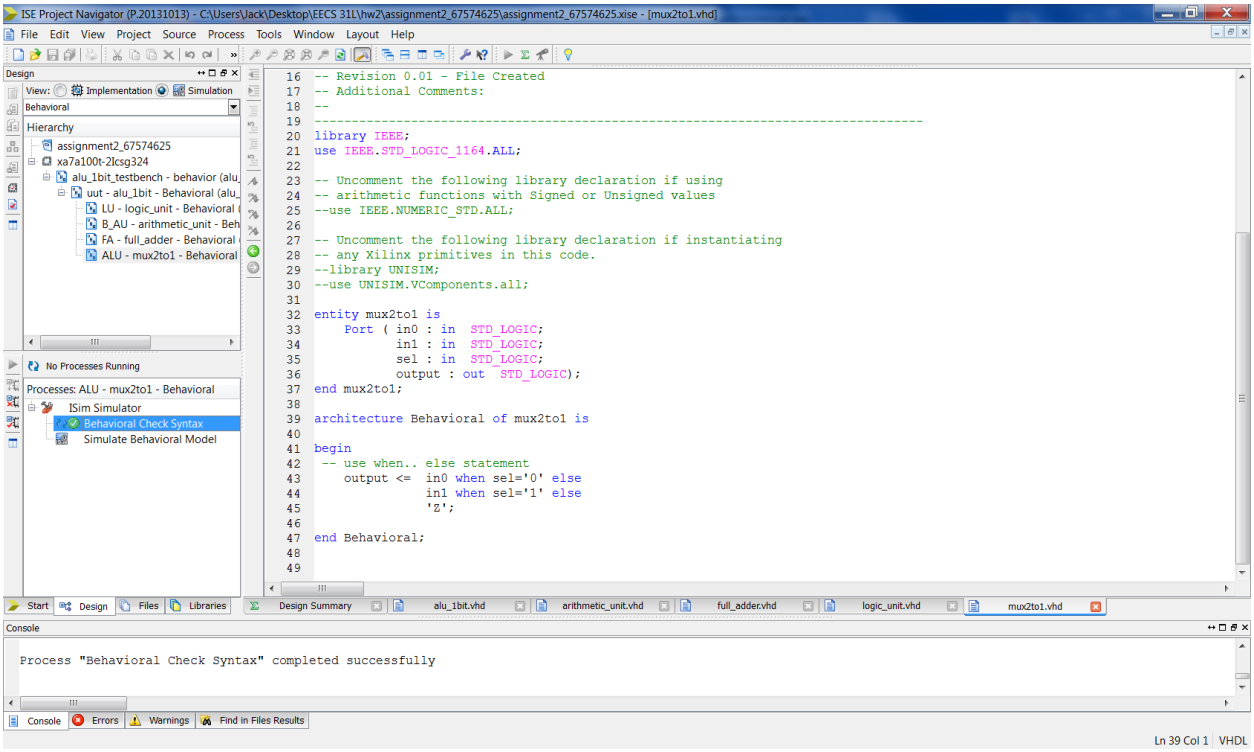



```
16  -- Revision 0.01 - File Created
17  -- Additional Comments:
18  --
19  ----------------------------------------------------------------------------------
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22
23  -- Uncomment the following library declaration if using
24  -- arithmetic functions with Signed or Unsigned values
25  --use IEEE.NUMERIC_STD.ALL;
26
27  -- Uncomment the following library declaration if instantiating
28  -- any Xilinx primitives in this code.
29  --library UNISIM;
30  --use UNISIM.VComponents.all;
31
32  entity full_adder is
33      Port ( A : in  STD_LOGIC;
34             B : in  STD_LOGIC;
35             cin : in  STD_LOGIC;
36             cout : out  STD_LOGIC;
37             sum : out  STD_LOGIC);
38  end full_adder;
39
40  architecture Behavioral of full_adder is
41
42  begin
43
44     sum <=   (A XOR B) XOR cin;
45     cout <=  (A AND B) OR (cin AND (A XOR B));
46
47  end Behavioral;
48
49
```

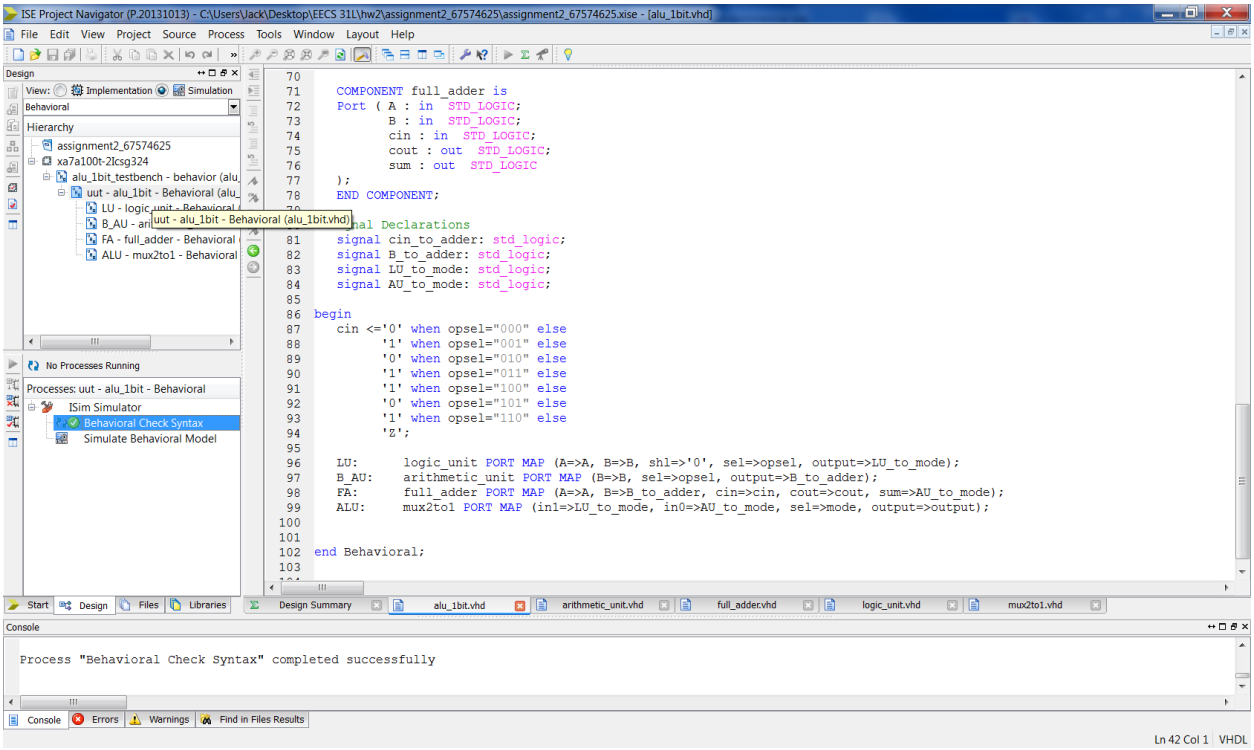
Logic Unit



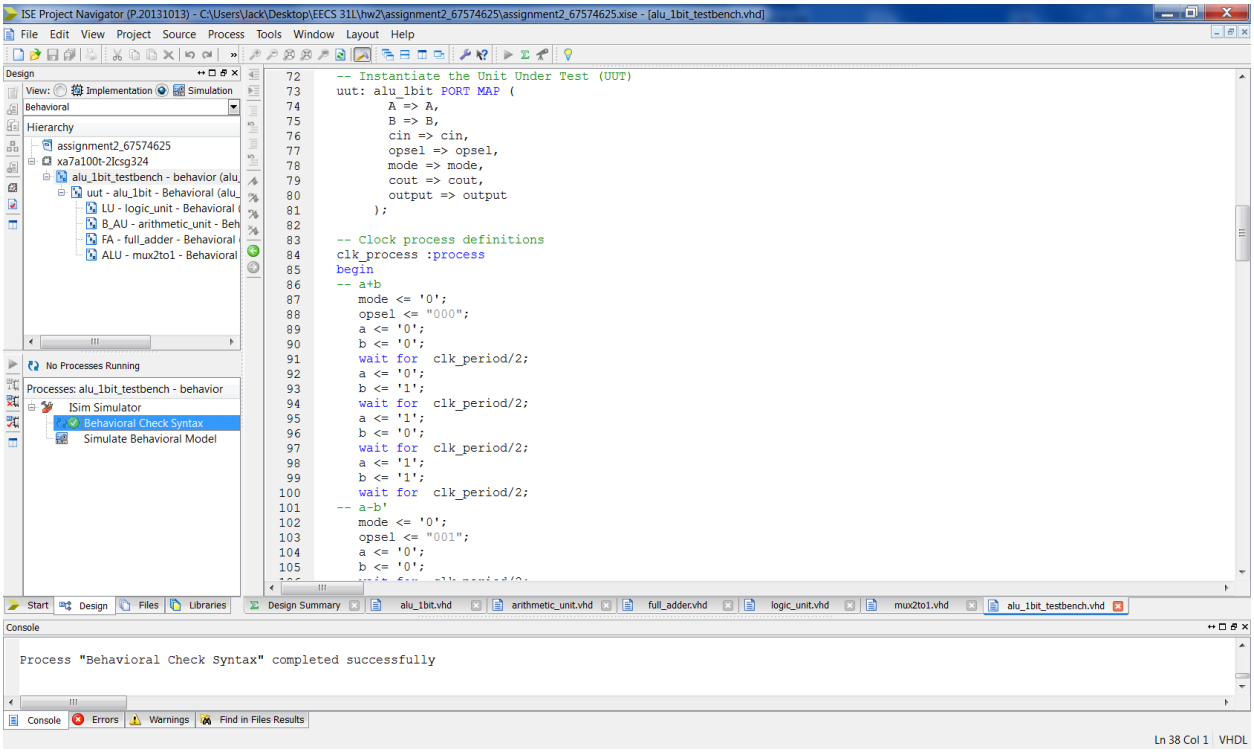
Mux 2to1



ALU



Testbench



5 Elaboration

Assumptions:

Value of cin

- I change cin from an IN port type to an INOUT port type. This way I could assign it a specific value that correlated to the function.
- This was done in the ALU architecture

Arithmetic

- a+b is just the full adder
- a +b' is just the full adder and will only subtract properly with cin = 1
- a+b'+1 subtraction is like two's complement full adder/subtractor; cin = 1 and b XOR 1
- Move A will only work properly if cin = 0
- To properly increment a number a 1-bit, A = A, B = 0 and cin = 1
- To properly decrement a number a 1-bit, A = A, B = 1, and cin = 0
- To properly add and increment, A = A, B = B, and cin = 1
- For a+b+1 to work properly, cin = 1

Logic

- Because A is only 1-bit, when A is shifted left, a 0 is always shifted in its place.

Regarding how I wrote my vhdl code

- I wrote 4 components and then the top level ALU 1bit
- Arithmetic unit that selects vale of b
- Full adder to handle the addition/subtraction
- Logic unit that selects logic operation
- And a 2to1 mux that uses mode to pick the final output.

Errors and Challenges:

- I had to learn how to use multiple vhdl modules, signals, component declarations, and port mapping in order for my alu to function properly
- Ran into errors about with setting up proper signaling and connecting components properly
- I was confused on how to properly set cin value within ALU and not the testbench. I went through a lot of trial and error in order to figure this out. I ended up changing cin

Simulation Log:

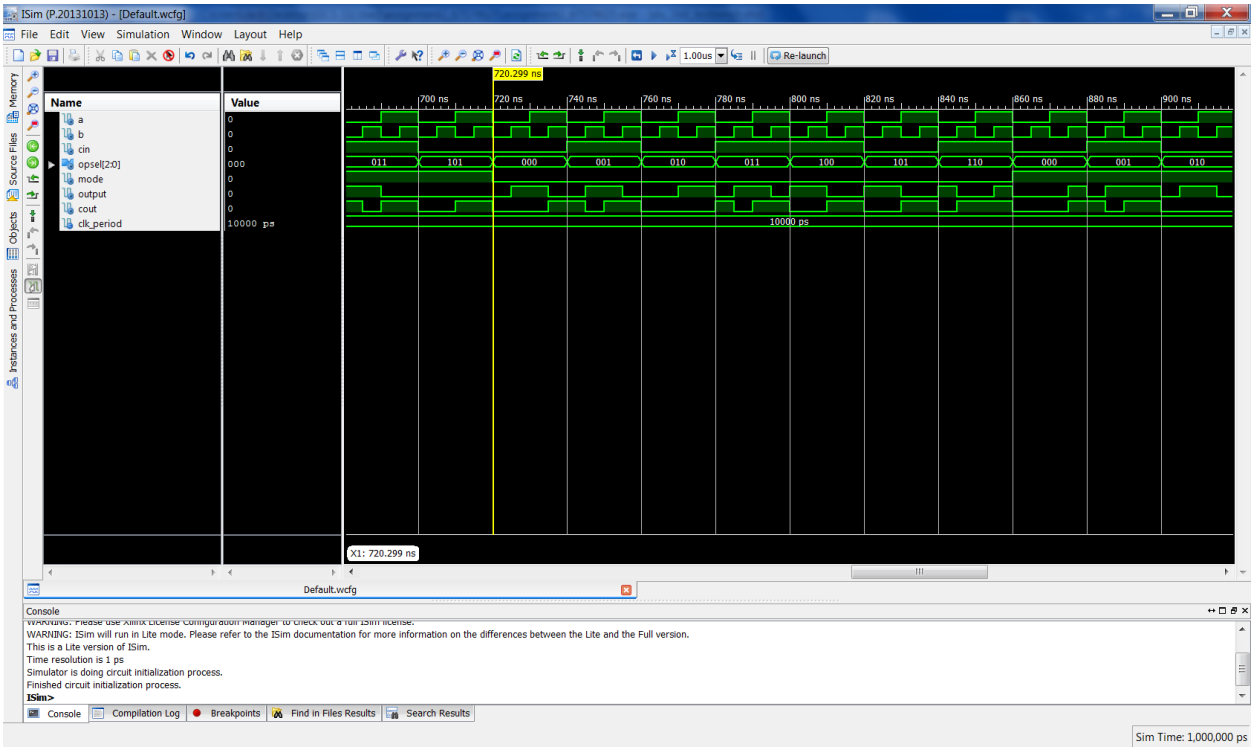
```
Started : "Simulate Behavioral Model".

Determining files marked for global include in the design...
Running fuse...
Command Line: fuse -intstyle ise -incremental -lib secureip -o {C:/Users/Jack/Desktop/EECS
31L/hw2/assignment2_67574625/alu_1bit_testbench_isim_beh.exe} -prj
{C:/Users/Jack/Desktop/EECS 31L/hw2/assignment2_67574625/alu_1bit_testbench_beh.prj}
work.alu_1bit_testbench {}
Running: C:\Xilinx\14.7\ISE_DS\ISE\bin\nt64\unwrapped\fuse.exe -intstyle ise -incremental -lib
secureip -o C:/Users/Jack/Desktop/EECS
31L/hw2/assignment2_67574625/alu_1bit_testbench_isim_beh.exe -prj
C:/Users/Jack/Desktop/EECS 31L/hw2/assignment2_67574625/alu_1bit_testbench_beh.prj
work.alu_1bit_testbench
ISim P.20131013 (signature 0x7708f090)
Number of CPUs detected in this system: 4
Turning on mult-threading, number of parallel sub-compilation jobs: 8
Determining compilation order of HDL files
Parsing VHDL file "C:/Users/Jack/Desktop/EECS
31L/hw2/assignment2_67574625/mux2to1.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS
31L/hw2/assignment2_67574625/logic_unit.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS
31L/hw2/assignment2_67574625/full_adder.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS
31L/hw2/assignment2_67574625/arithmetic_unit.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS
31L/hw2/assignment2_67574625/alu_1bit.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS
31L/hw2/assignment2_67574625/alu_1bit_testbench.vhd" into library work
Starting static elaboration
Completed static elaboration
Compiling package standard
Compiling package std_logic_1164
Compiling package numeric_std
Compiling architecture behavioral of entity logic_unit [logic_unit_default]
Compiling architecture behavioral of entity arithmetic_unit [arithmetic_unit_default]
Compiling architecture behavioral of entity full_adder [full_adder_default]
Compiling architecture behavioral of entity mux2to1 [mux2to1_default]
Compiling architecture behavioral of entity alu_1bit [alu_1bit_default]
```

Compiling architecture behavior of entity alu_1bit_testbench
Time Resolution for simulation is 1ps.
Waiting for 1 sub-compilation(s) to finish...
Compiled 14 VHDL Units
Built simulation executable C:/Users/Jack/Desktop/EECS
31L/hw2/assignment2_67574625/alu_1bit_testbench_isim_beh.exe
Fuse Memory Usage: 35992 KB
Fuse CPU Usage: 576 ms
Launching ISim simulation engine GUI...
"C:/Users/Jack/Desktop/EECS
31L/hw2/assignment2_67574625/alu_1bit_testbench_isim_beh.exe" -intstyle ise -gui -tclbatch
isim.cmd -wdb "C:/Users/Jack/Desktop/EECS
31L/hw2/assignment2_67574625/alu_1bit_testbench_isim_beh.wdb"
ISim simulation engine GUI launched successfully

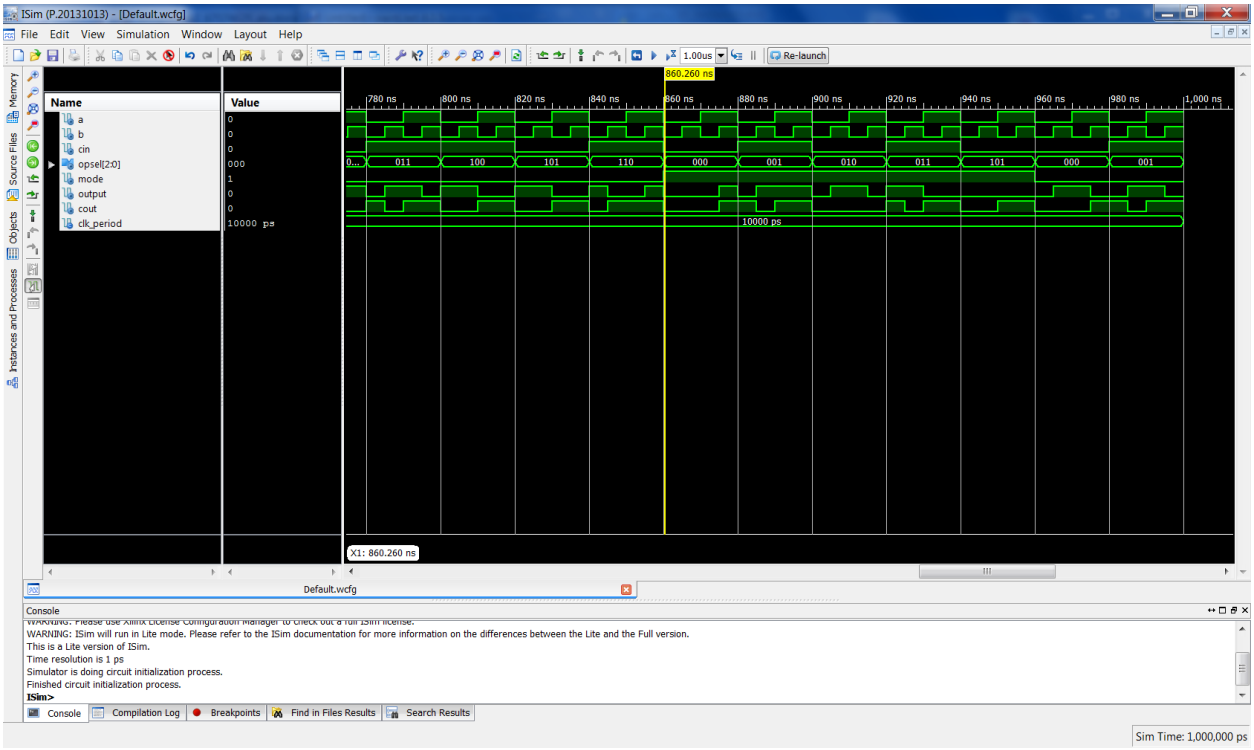
Process "Simulate Behavioral Model" completed successfully

6 Waveforms

Full Waveform Snapshot 1

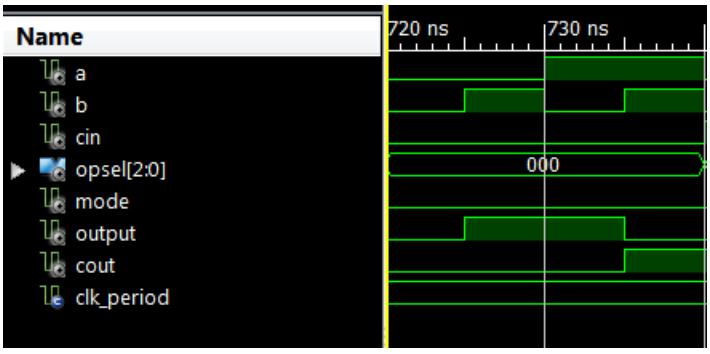


Full Waveform Snapshot 2

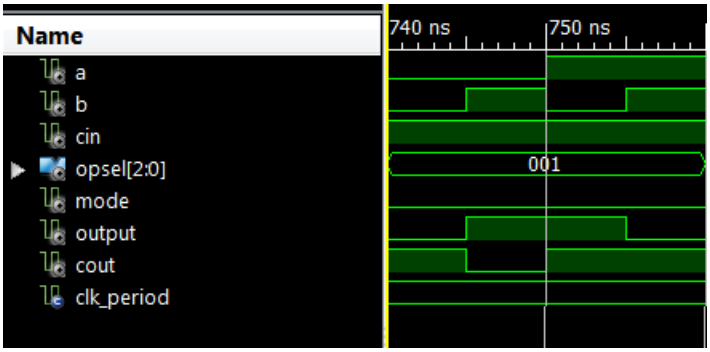


Snapshots of Waveforms

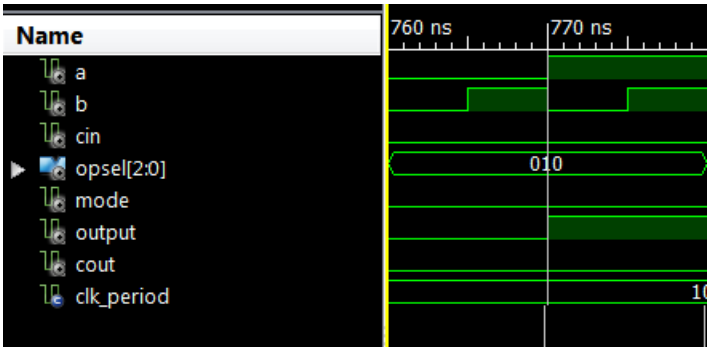
a+b



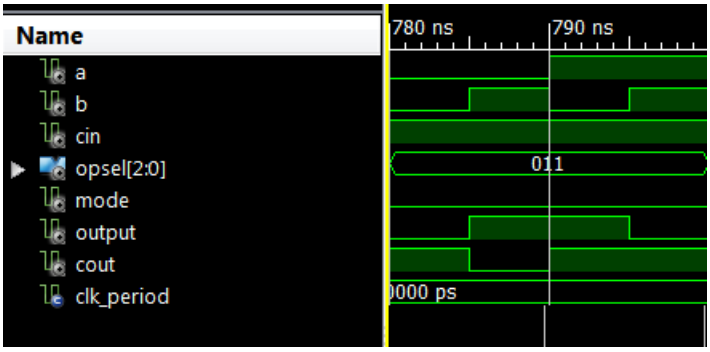
a-b'



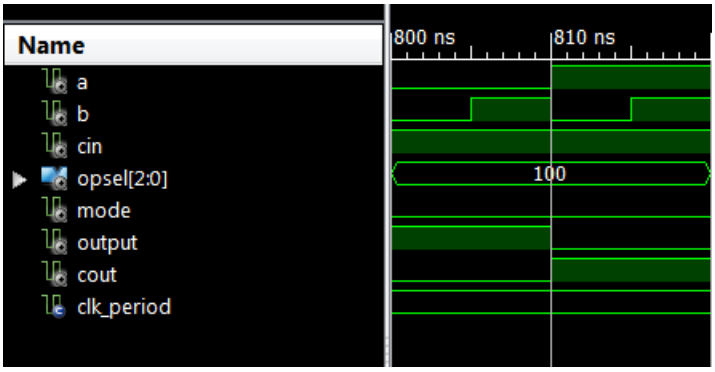
a


Name
a
b
cin
opsel[2:0]
mode
output
cout
clk_period
760 ns
770 ns
010

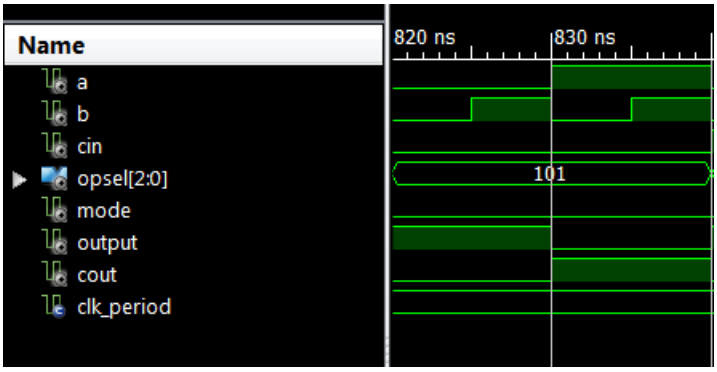
a+b'+1


Name
a
b
cin
opsel[2:0]
mode
output
cout
clk_period
780 ns
790 ns
011

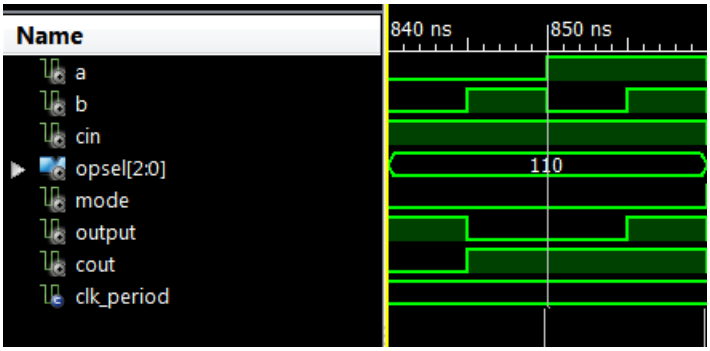
a+1


Name
a
b
cin
opsel[2:0]
mode
output
cout
clk_period
800 ns
810 ns
100

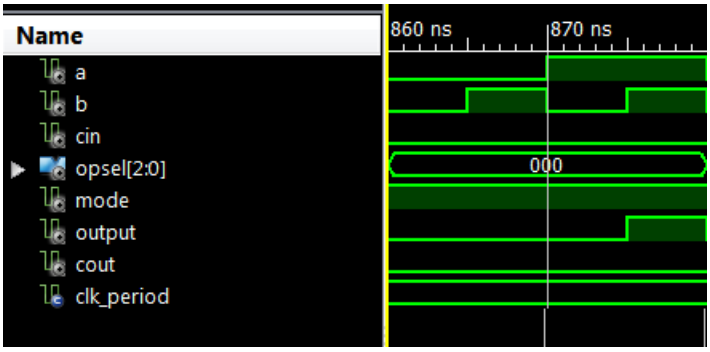
a-1


Name
a
b
cin
opsel[2:0]
mode
output
cout
clk_period
820 ns
830 ns
101

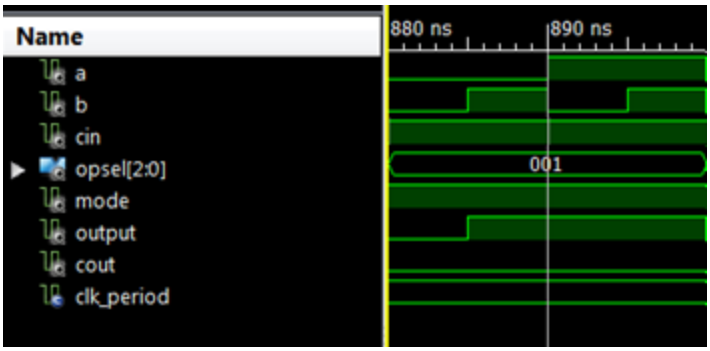
a+b+1



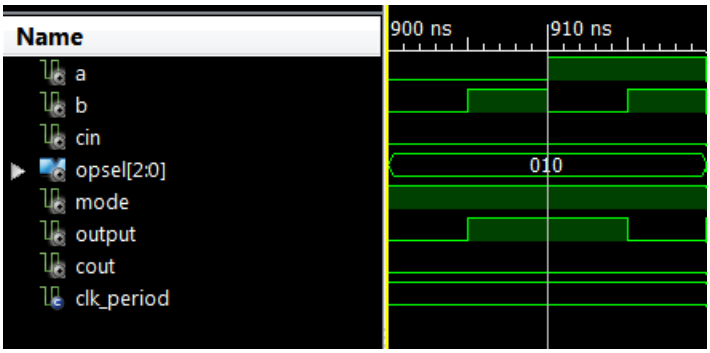
a AND b



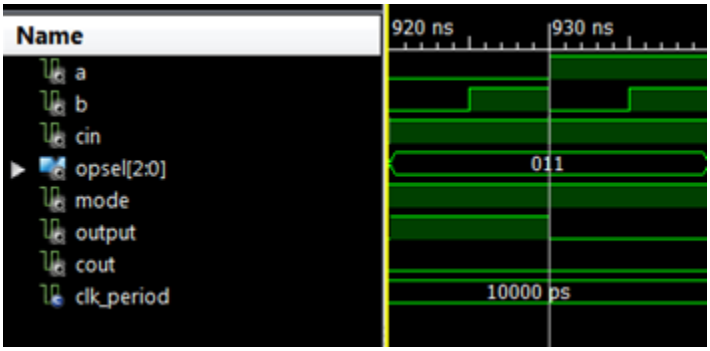
a OR b



a XOR b



a'



shift left a