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 ⊕ B) ⊕ Cin

Cout = (A)(B) + (Cin)(A ⊕ B)

Two’s Complement Full Adder

Sum = (A ⊕ (B ⊕ Cin)) ⊕ Cin

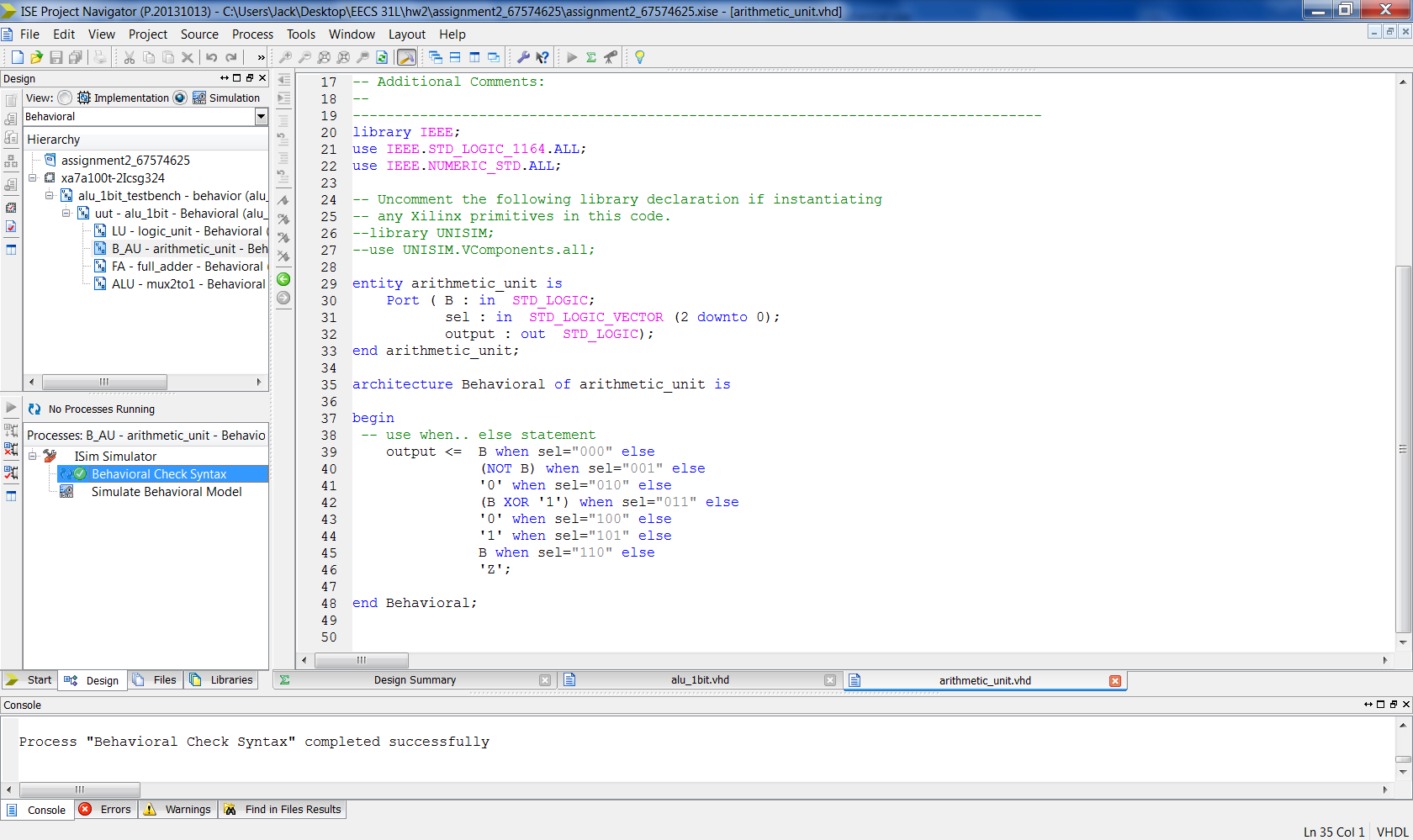
Cout = (A)(B ⊕ Cin) + ((Cin)(A ⊕ (B ⊕ Cin)))

**Design:**

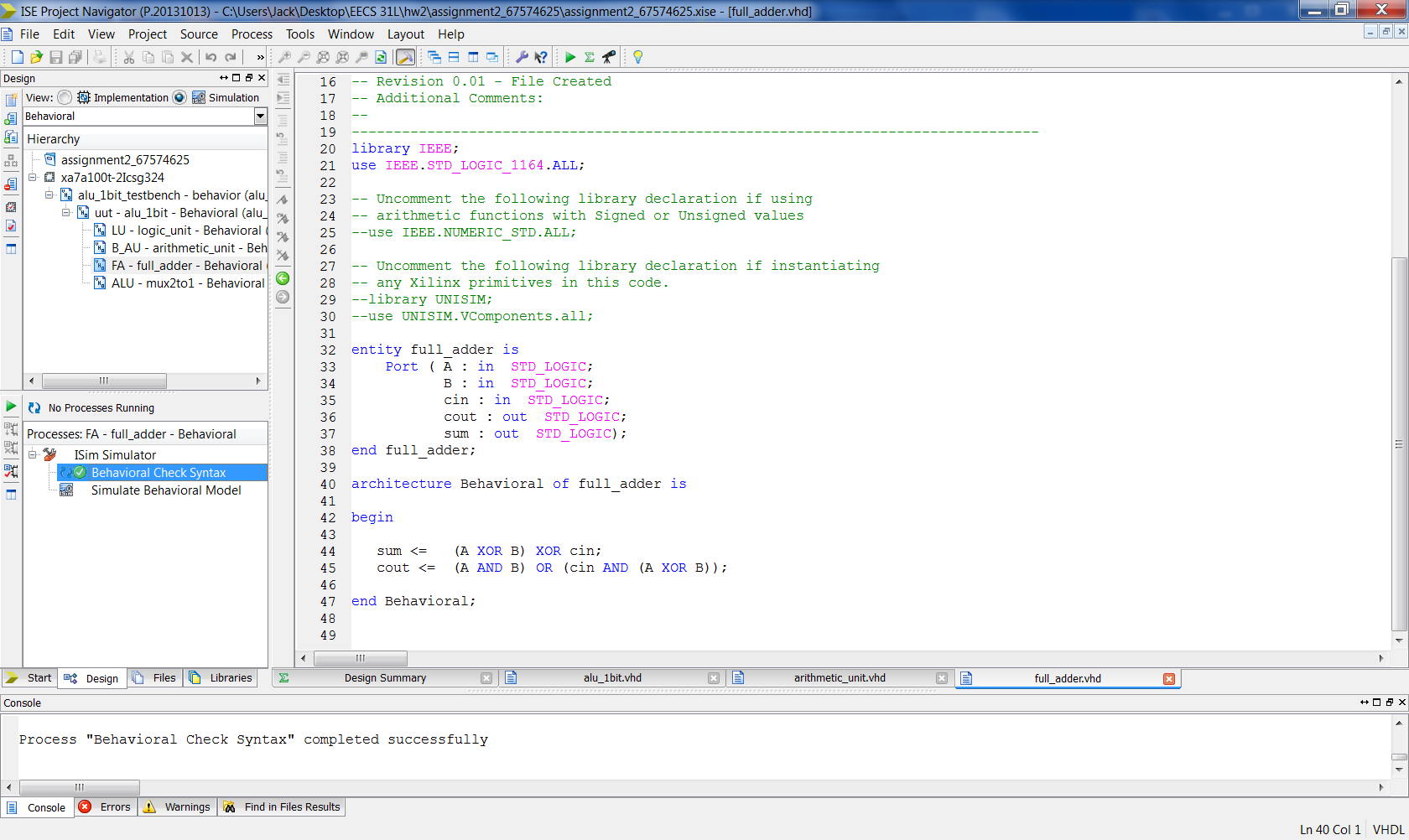


**4 Compilation**

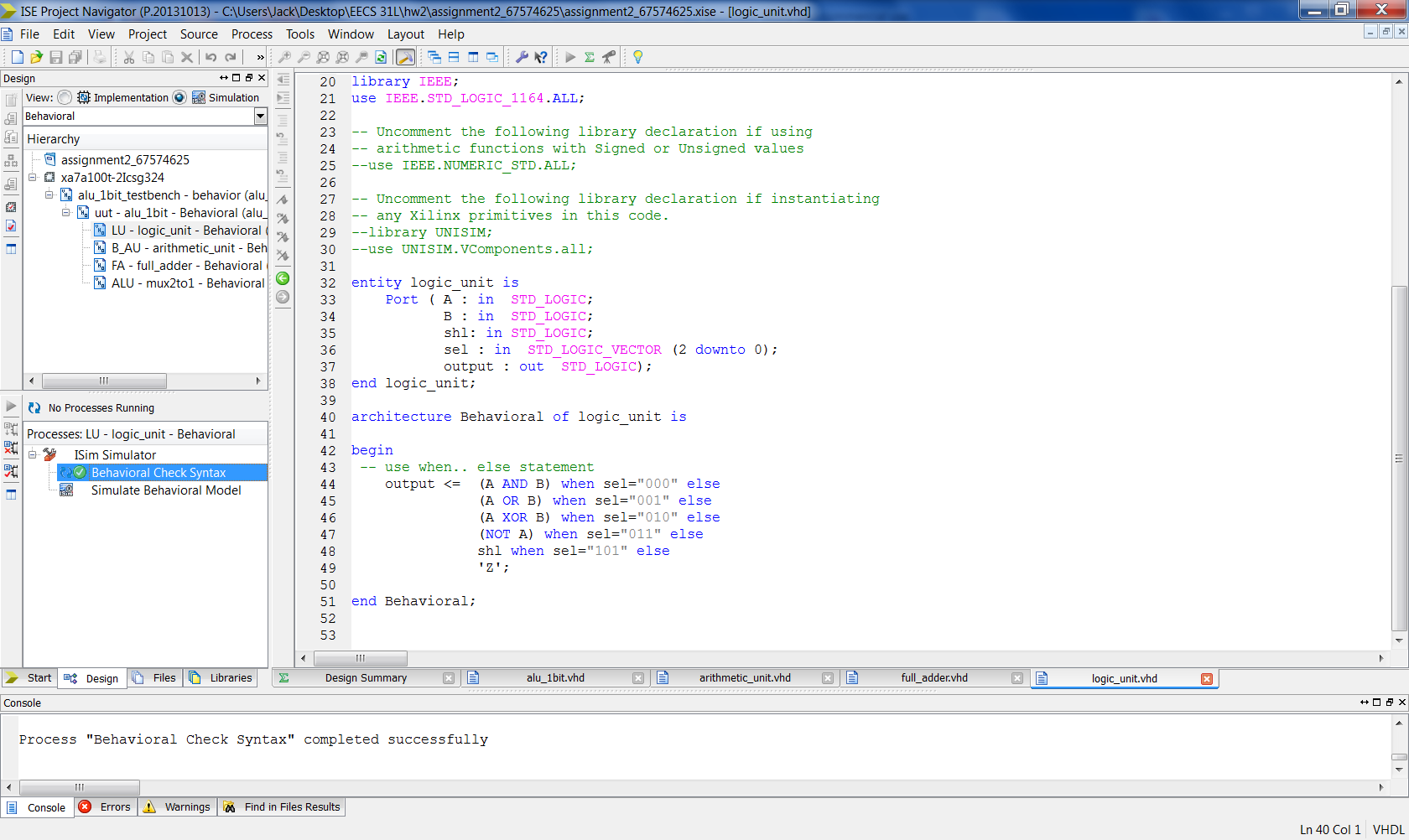
Arithmetic unit



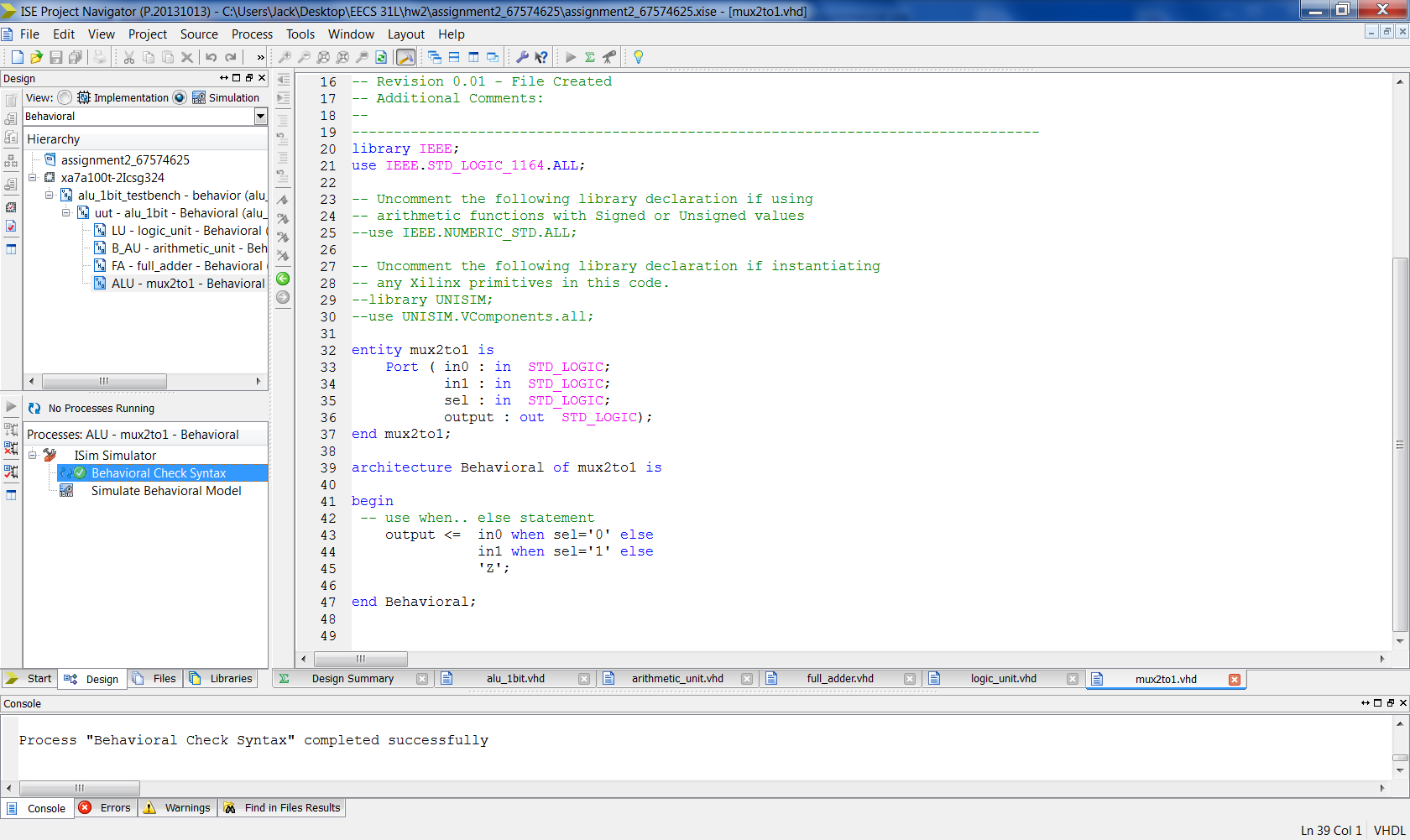
Full Adder



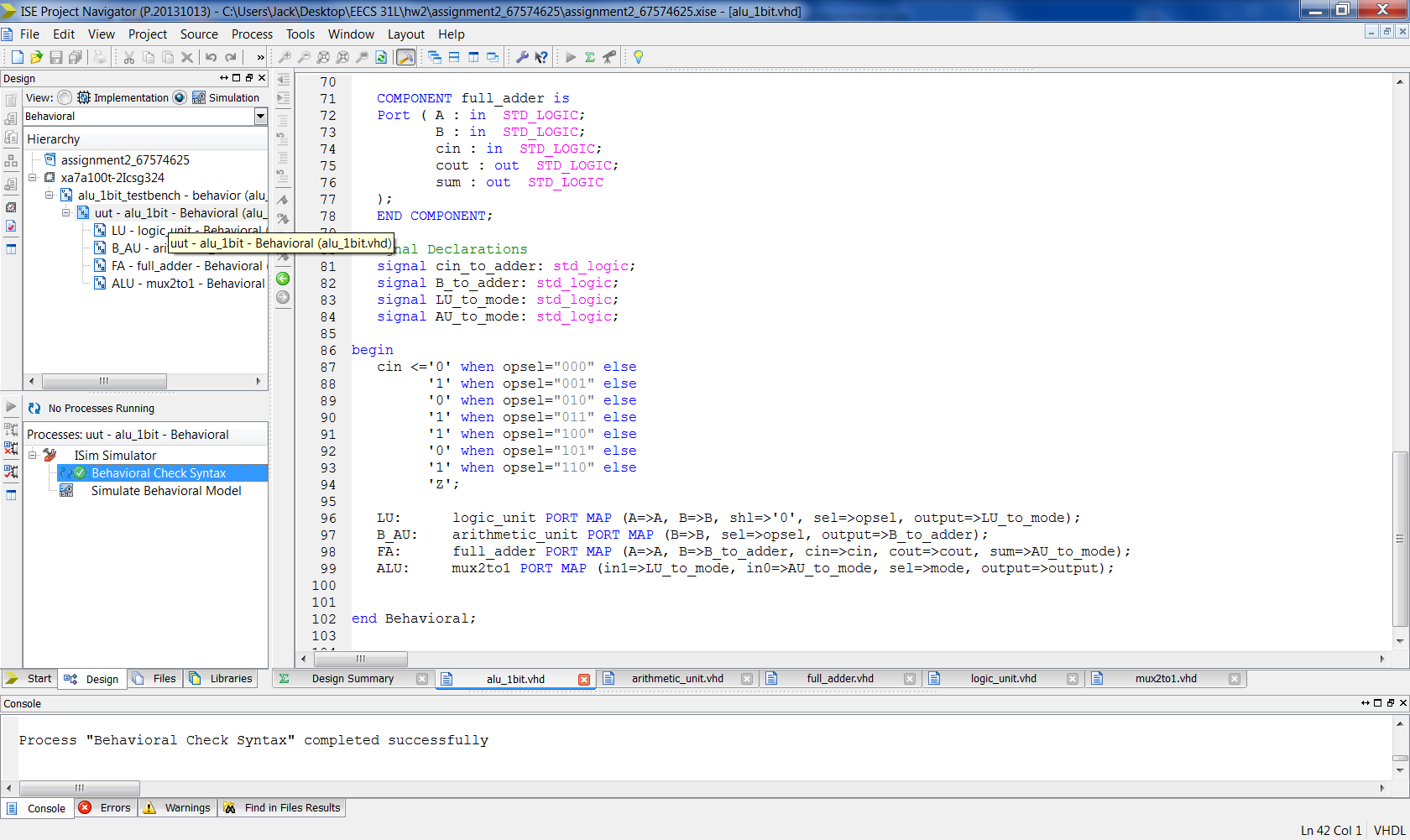
Logic Unit



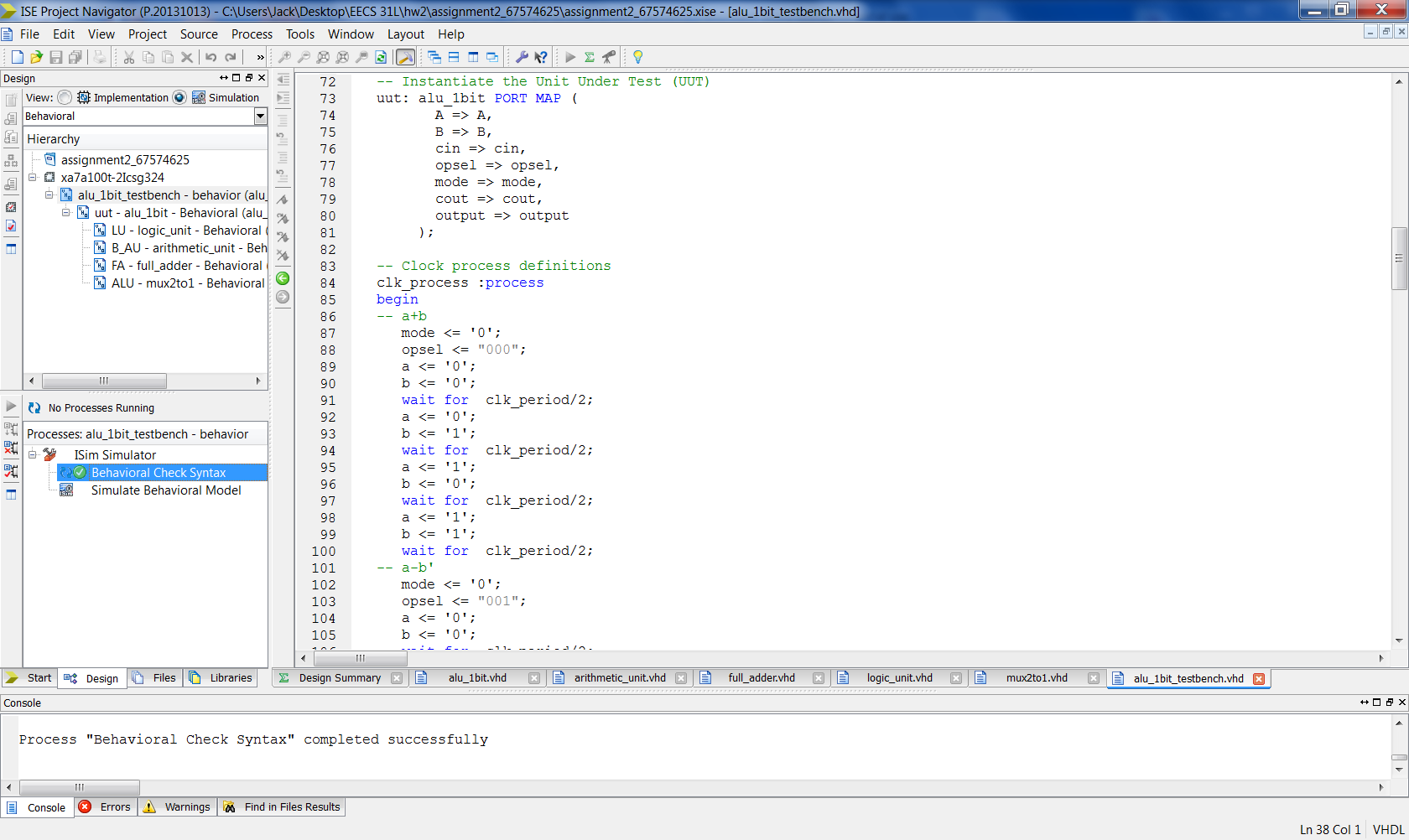
Mux 2to1



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...

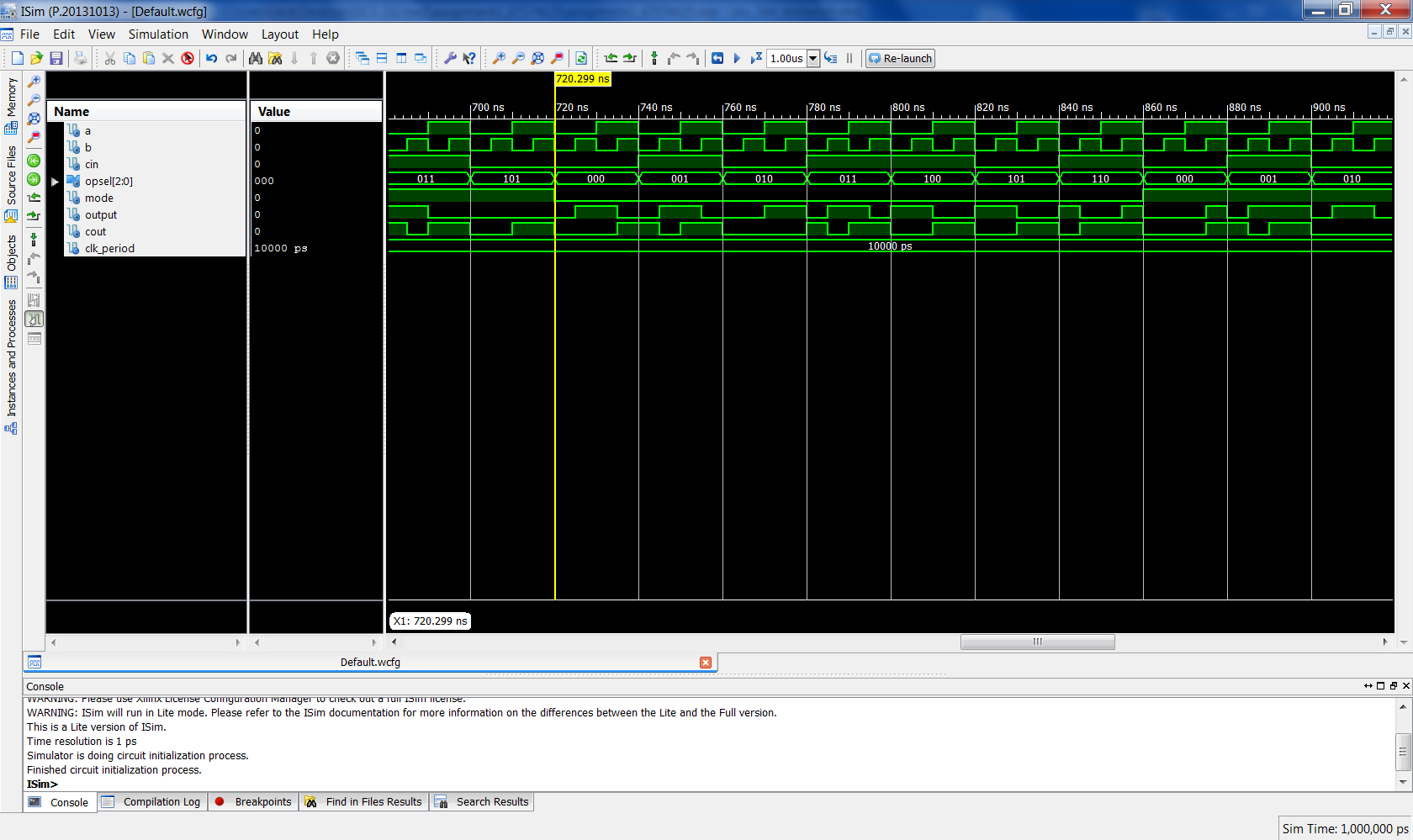
"C:/Users/Jack/Desktop/EECS 31L/hw2/assignment2\_67574625/alu\_1bit\_testbench\_isim\_beh.exe" -intstyle ise -gui -tclbatch isim.cmd -wdb "C:/Users/Jack/Desktop/EECS 31L/hw2/assignment2\_67574625/alu\_1bit\_testbench\_isim\_beh.wdb"

ISim simulation engine GUI launched successfully

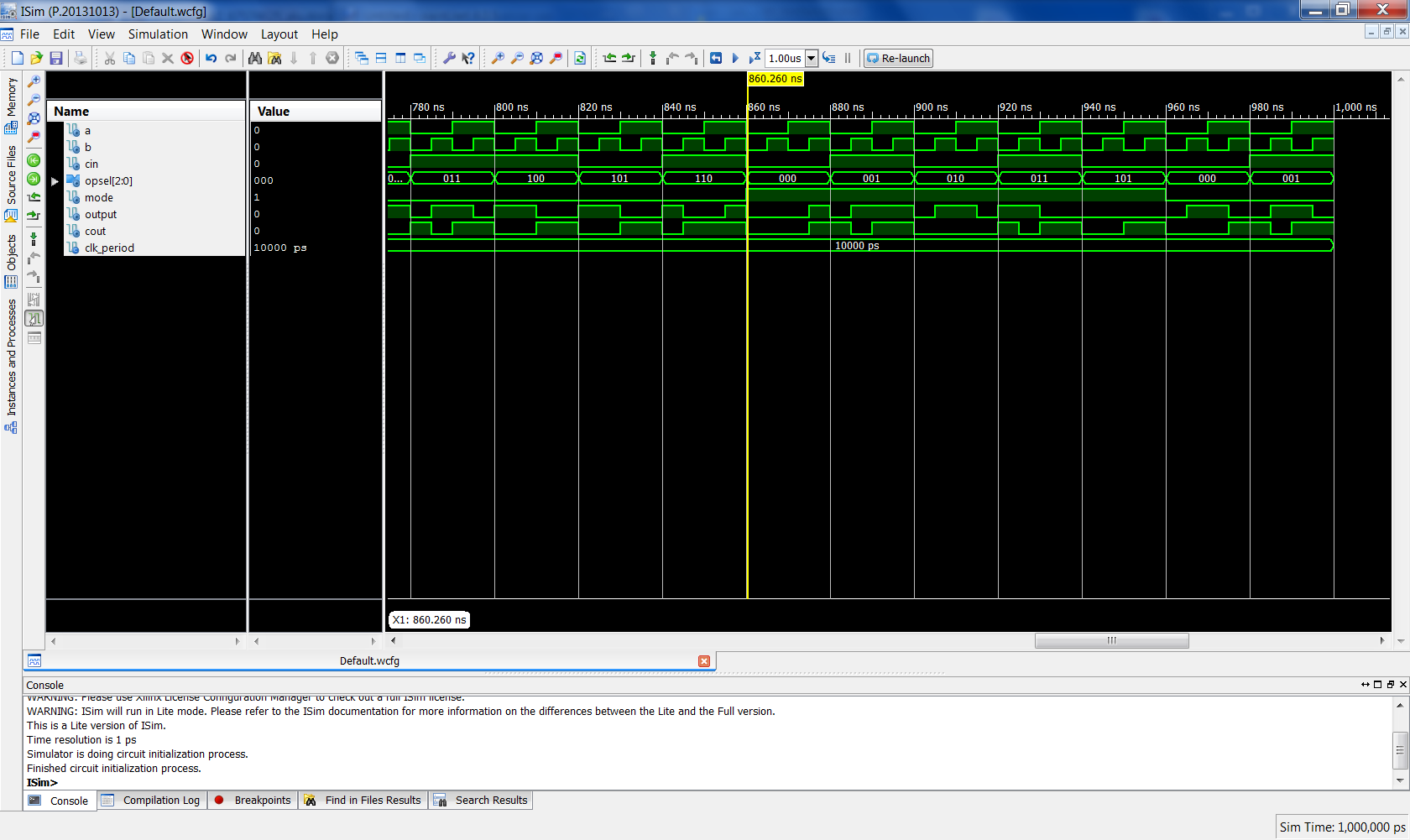
Process "Simulate Behavioral Model" completed successfully

**6 Waveforms**

Full Waveform Snapshot 1

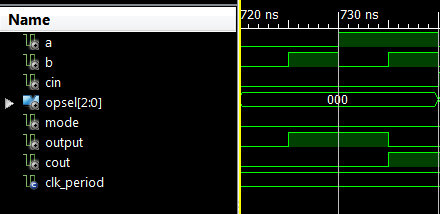


Full Waveform Snapshot 2

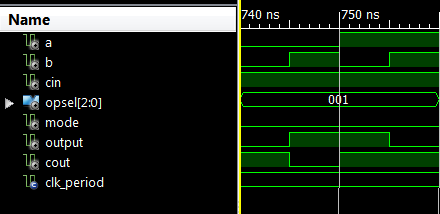


Snapshots of Waveforms

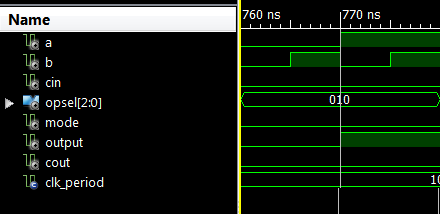
a+b



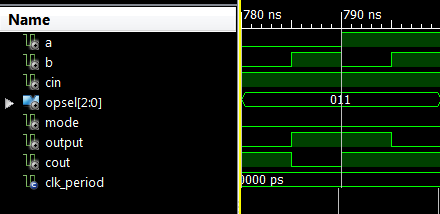
a-b'



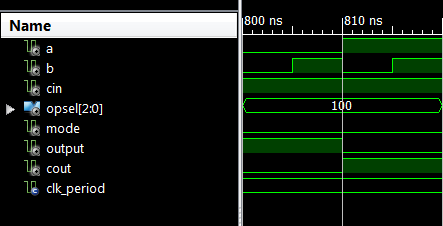
a



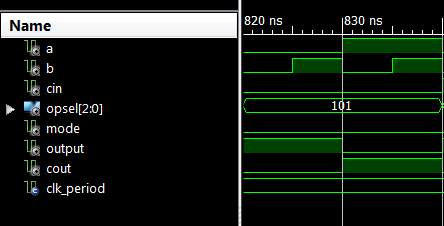
a+b'+1



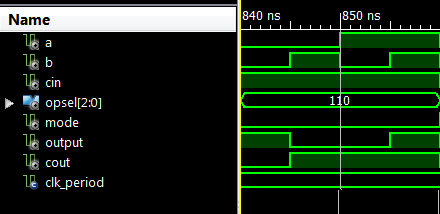
a+1



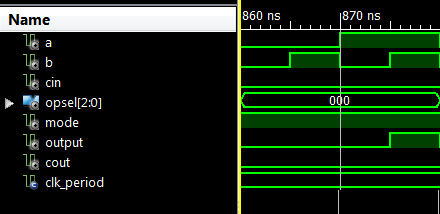
a-1



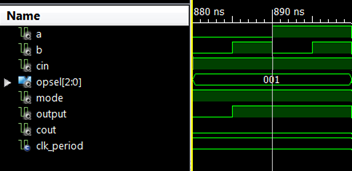
a+b+1



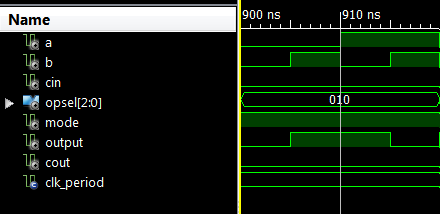
a AND b



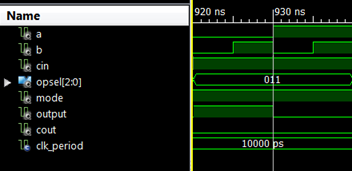
a OR b



a XOR b



a'



shift left a

