

Introduction to Digital Logic

EECS/CSE 31L

Assignment 3: 32-Bit ALU

Prepared by: Jack Melcher
Student ID: 67574625

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

February 7, 2015

1 Block Description

Take in two 32-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	31 downto 0	IN	Value of first input
B	31 downto 0	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
cout	1	OUT	The carry out value
output	31 downto 0	OUT	The output value of operation

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

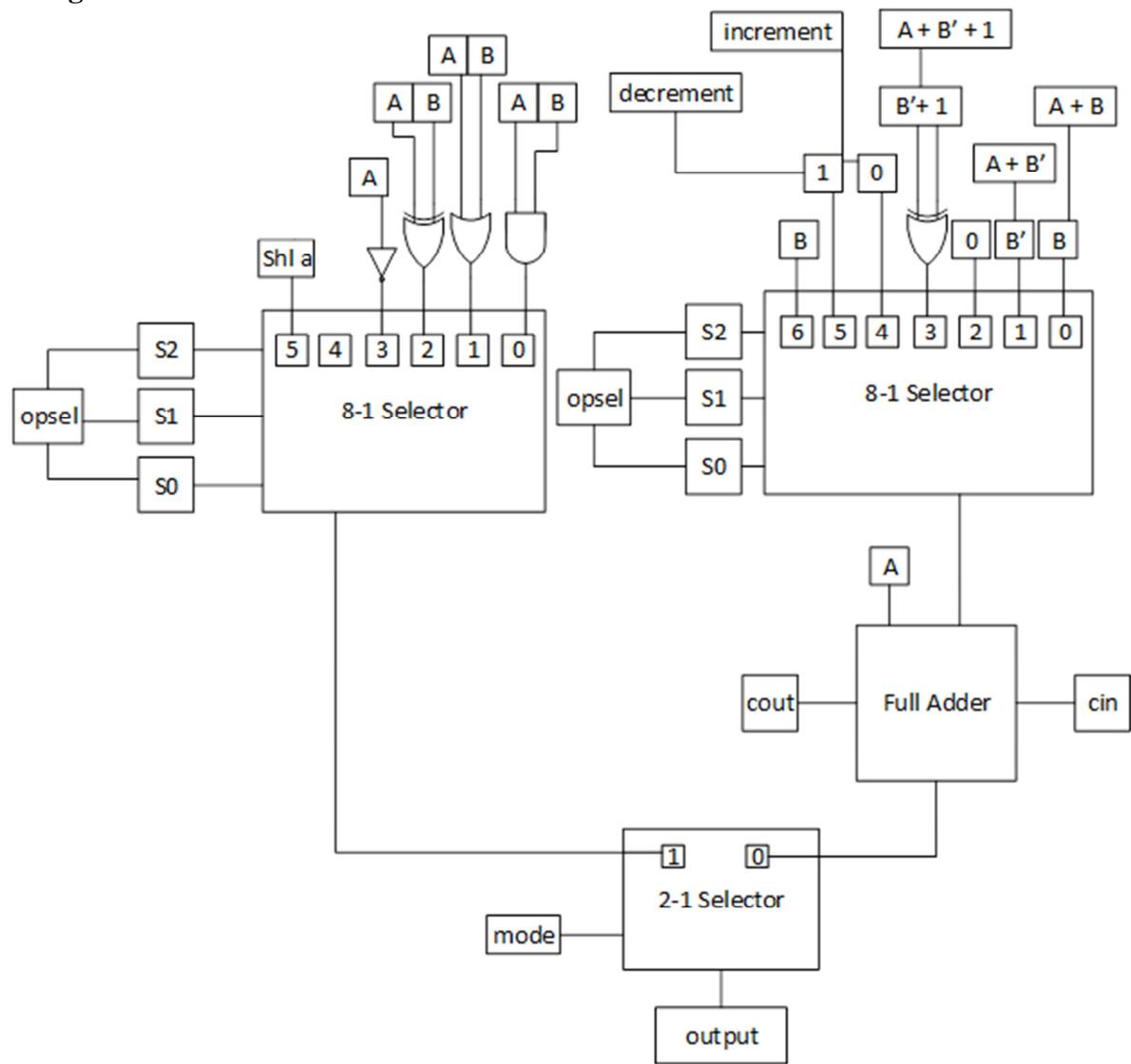
In_0	In_1	Cin	Cout	Sum
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	1	0
0	1	1	0	1
1	0	1	1	1
1	1	1	1	0

Boolean Expressions:

Full Adder

$$\begin{aligned} \text{Sum} &= (A \oplus B) \oplus \text{Cin} \\ \text{Cout} &= (A)(B) + (\text{Cin})(A \oplus B) \end{aligned}$$

Design:



4 Compilation

Arithmetic unit

The screenshot shows the ISE Project Navigator interface. The left pane displays the project hierarchy under 'Design' view, specifically the 'Behavioral' tab. The main pane shows the VHDL code for the 'arithmetic_unit' entity. The code defines a port with a 2-bit input 'B' and a single-bit output 'output'. It uses a case statement based on the value of 'B' to calculate the output. The code editor has syntax highlighting for VHDL. Below the code editor is a 'Design Summary' window showing 'mux2to1.vhd' and 'full_adder.vhd'. At the bottom, a 'Console' window shows the message 'Process "Behavioral Check Syntax" completed successfully'. The status bar at the bottom right indicates 'Ln 45 Col 1 | VHDL'.

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

Full Adder

The screenshot shows the ISE Project Navigator interface. The left pane displays the project hierarchy under 'Design' view, specifically the 'Behavioral' tab. The main pane shows the VHDL code for the 'full_adder' entity. The code defines a port with three inputs: 'A' (1-bit), 'B' (1-bit), and 'cin' (1-bit), and three outputs: 'cout' (1-bit), 'sum' (1-bit), and 'cout' (1-bit). It uses a series of logic operations (XOR, AND, OR) to calculate the sum and carry. The code editor has syntax highlighting for VHDL. Below the code editor is a 'Design Summary' window showing 'mux2to1.vhd' and 'full_adder.vhd'. At the bottom, a 'Console' window shows the message 'Process "Behavioral Check Syntax" completed successfully'. The status bar at the bottom right indicates 'Ln 40 Col 1 | VHDL'.

```
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

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity logic_unit is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           sh1: in STD_LOGIC;
           sel : in STD_LOGIC_VECTOR (2 downto 0);
           output : out STD_LOGIC);
end logic_unit;
architecture Behavioral of logic_unit is
begin
    use when.. else statement
    output <= (A AND B) when sel="000" else
                (A OR B) when sel="001" else
                (A XOR B) when sel="010" else
                (NOT A) when sel="011" else
                'Z' when sel="101" else
                'Z';
end Behavioral;

```

Process "Behavioral Check Syntax" completed successfully

Mux 2to1

```

-- Revision 0.01 - File Created
-- Additional Comments:
--
-----
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity mux2to1 is
    Port ( in0 : in STD_LOGIC;
           in1 : in STD_LOGIC;
           sel : in STD_LOGIC;
           output : out STD_LOGIC);
end mux2to1;
architecture Behavioral of mux2to1 is
begin
    -- use when.. else statement
    output <= in0 when sel='0' else
                in1 when sel='1' else
                'Z';
end Behavioral;

```

Process "Behavioral Check Syntax" completed successfully

ALU 1bit

ISE Project Navigator (P.20131013) - C:\Users\Jack\Desktop\EECS 311\hw3\assignment3_67574625\assignment3_67574625.xise - [alu_1bit.vhd]

```

64  COMPONENT arithmetic_unit is
65    Port ( B : in STD_LOGIC;
66      sel : in STD_LOGIC_VECTOR (2 downto 0);
67      output : out STD_LOGIC
68  );
69  END COMPONENT;
70
71  COMPONENT full_adder is
72    Port ( A : in STD_LOGIC;
73      B : in STD_LOGIC;
74      cin : in STD_LOGIC;
75      cout : out STD_LOGIC;
76      sum : out STD_LOGIC
77  );
78  END COMPONENT;
79
80 --Signal Declarations
81 signal cin_to_adder: std_logic;
82 signal B_to_adder: std_logic;
83 signal LU_to_mode: std_logic;
84 signal AU_to_mode: std_logic;
85
86 begin
87
88   LU: logic_unit PORT MAP (A=>A, B=>B, shl=>shl, sel=>opsel, output=>LU_to_mode);
89   BAU: arithmetic_unit PORT MAP (B=>B, sel=>opsel, output=>BAU_to_mode);
90   FA: full_adder PORT MAP (A=>A, B=>B_to_adder, cin=>cin, cout=>cout, sum=>AU_to_mode);
91   ALU: mux2to1 PORT MAP (in1=>LU_to_mode, in0=>AU_to_mode, sel=>mode, output=>output);
92
93
94 end Behavioral;
95
96
97

```

Design Summary mux2to1.vhd full_adder.vhd arithmetic_unit.vhd logic_unit.vhd alu_1bit.vhd alu_32bit.vhd

Console

```
Started : "Launching ISE Text Editor to edit alu_32bit.vhd".
```

Ln 43 Col 2 VHDL

ALU 32bit

ISE Project Navigator (P.20131013) - C:\Users\Jack\Desktop\EECS 311\hw3\assignment3_67574625\assignment3_67574625.xise - [alu_32bit.vhd]

```

44 -- Component Declaration
45 COMPONENT alu_1bit IS
46   Port ( A : in STD_LOGIC;
47         B : in STD_LOGIC;
48         cin : in STD_LOGIC;
49         shl : in STD_LOGIC;
50         opsel : in STD_LOGIC_VECTOR (2 downto 0);
51         mode : in STD_LOGIC;
52         cout : out STD_LOGIC;
53         output : out STD_LOGIC
54   );
55 END COMPONENT;
56
57 -- Signal Declaration
58 SIGNAL carry: std_logic_vector(30 downto 0);
59
60 begin
61   cin <='0' when opsel="000" else
62     '1' when opsel="001" else
63     '0' when opsel="010" else
64     '1' when opsel="011" else
65     '1' when opsel="100" else
66     '0' when opsel="101" else
67     '1' when opsel="110" else
68     '2';
69
70 -- First bit
71   A1: alu_1bit PORT MAP (A=>A(0), B=>b(0), cin=>cin, shl=>'0', opsel=>opsel, mode=>mode, cout=>carry(0), output=>output(0));
72 -- bits between 1 and 30
73   G1: FOR i IN 1 TO 30 GENERATE
74     ALU: alu_1bit PORT MAP (A=>A(i), B=>b(i), cin=>carry(i-1), shl=>A(i-1), opsel=>opsel, mode=>mode, cout=>carry(i), output=>output(i));
75   END GENERATE;
76 --Last bit
77   A32: alu_1bit PORT MAP (A=>A(31), B=>b(31), cin=>carry(30), shl=>A(30), opsel=>opsel, mode=>mode, cout=>cout, output=>output(31));
78
79 end Behavioral;
80
81

```

Design Summary mux2to1.vhd full_adder.vhd arithmetic_unit.vhd logic_unit.vhd alu_1bit.vhd alu_32bit.vhd

Console

```
Process "Behavioral Check Syntax" completed successfully
```

Ln 67 Col 35 VHDL

Testbench

```
ISE Project Navigator (P.20131013) - C:\Users\Jack\Desktop\EECS 311\hw3\assignment3_67574625\assignment3_67574625.xise - [alu_32bit_testbench.vhd]
File Edit View Project Source Process Tools Window Layout Help
Design Implementation Simulation
View: Behavioral Hierarchy
Behavioral
assignment3_67574625
  alu32bit
    alu32bit_behavior - behavior
      uut - alu_32bit - Behavioral
        A1 - alu_1bit - Behaviora
        LU - logic_unit - Behav
        B_AU - arithmetic_unit
        FA - full_adder - Behav
        ALU - mux2to1 - Behav
        ALUS - alu_1bit - Behav
        LU - logic_unit - Behav
        B_AU - arithmetic_unit
        FA - full_adder - Behav
No Processes Running
Processes: alu_32bit_testbench - behavior
  ISim Simulator
  Behavioral Check Syntax
  Simulate Behavioral Model
Start Design Files Libraries Design Summary mux2to1.vhd full_addervhdl arithmetic_unit.vhd logic_unit.vhd alu_1bit.vhd alu_32bit.vhd alu_32bit_testbench.vhd
Console
Process "Behavioral Check Syntax" completed successfully
Console Errors Warnings Find in Files Results
Ln 108 Col 4 VHDL
```

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.

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 = 0, and cin = 1
- For a+b+1 to work properly, cin = 1

Logic

- Because A is now multiple bits, only the first bit becomes zero and the rest get shifted over to the left

ALU 1bit

- cin port type in the 1bit ALU was changed from INOUT to just IN
- Added in a shl IN port in order to be able to shift multiple bits

Regarding how I wrote my vhdl code

- I wrote 4 components and then the top level ALU 1bit
- And then ontop of that, ALU 32bit top level vhdl file with a for generate loop
- The cin value set was moved to the 32bit ALU
- 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
- Figuring out that I needed an ALU module just for the First, and Last, and a for generate for the rest of the bits inbetween
- Properly setting up a for generate loop allowing the carryout to go to the carryin

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/hw3/assignment3_67574625/alu_32bit_testbench_isim_beh.exe} -prj  
{C:/Users/Jack/Desktop/EECS 31L/hw3/assignment3_67574625/alu_32bit_testbench_beh.prj}  
work.alu_32bit_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/hw3/assignment3_67574625/alu_32bit_testbench_isim_beh.exe -prj  
C:/Users/Jack/Desktop/EECS 31L/hw3/assignment3_67574625/alu_32bit_testbench_beh.prj  
work.alu_32bit_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

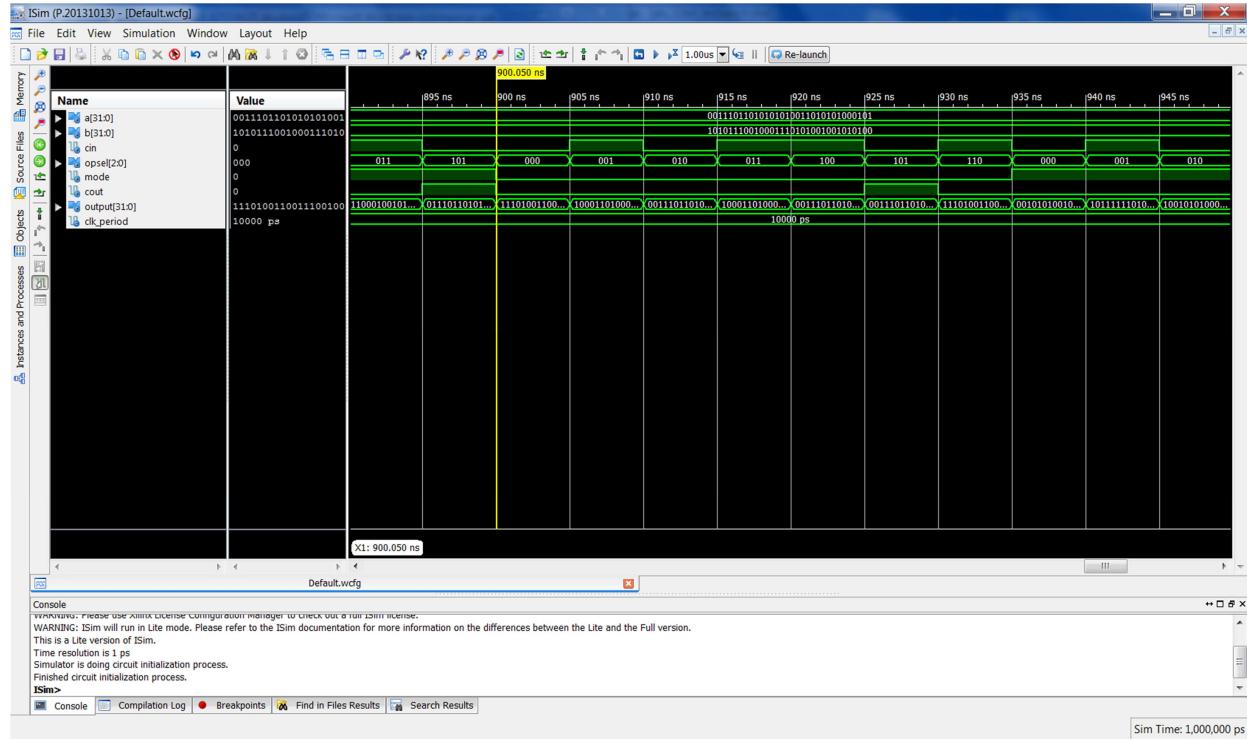
```
31L/hw3/assignment3_67574625/mux2to1.vhd" into library work
```

Parsing VHDL file "C:/Users/Jack/Desktop/EECS
31L/hw3/assignment3_67574625/logic_unit.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS
31L/hw3/assignment3_67574625/full_adder.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS
31L/hw3/assignment3_67574625/arithmetic_unit.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS
31L/hw3/assignment3_67574625/alu_1bit.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS
31L/hw3/assignment3_67574625/alu_32bit.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS
31L/hw3/assignment3_67574625/alu_32bit_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 behavioral of entity alu_32bit [alu_32bit_default]
Compiling architecture behavior of entity alu_32bit_testbench
Time Resolution for simulation is 1ps.
Waiting for 1 sub-compilation(s) to finish...
Compiled 16 VHDL Units
Built simulation executable C:/Users/Jack/Desktop/EECS
31L/hw3/assignment3_67574625/alu_32bit_testbench_isim_beh.exe
Fuse Memory Usage: 35984 KB
Fuse CPU Usage: 592 ms
Launching ISim simulation engine GUI...
"C:/Users/Jack/Desktop/EECS
31L/hw3/assignment3_67574625/alu_32bit_testbench_isim_beh.exe" -intstyle ise -gui -tclbatch
isim.cmd -wdb "C:/Users/Jack/Desktop/EECS
31L/hw3/assignment3_67574625/alu_32bit_testbench_isim_beh.wdb"
ISim simulation engine GUI launched successfully

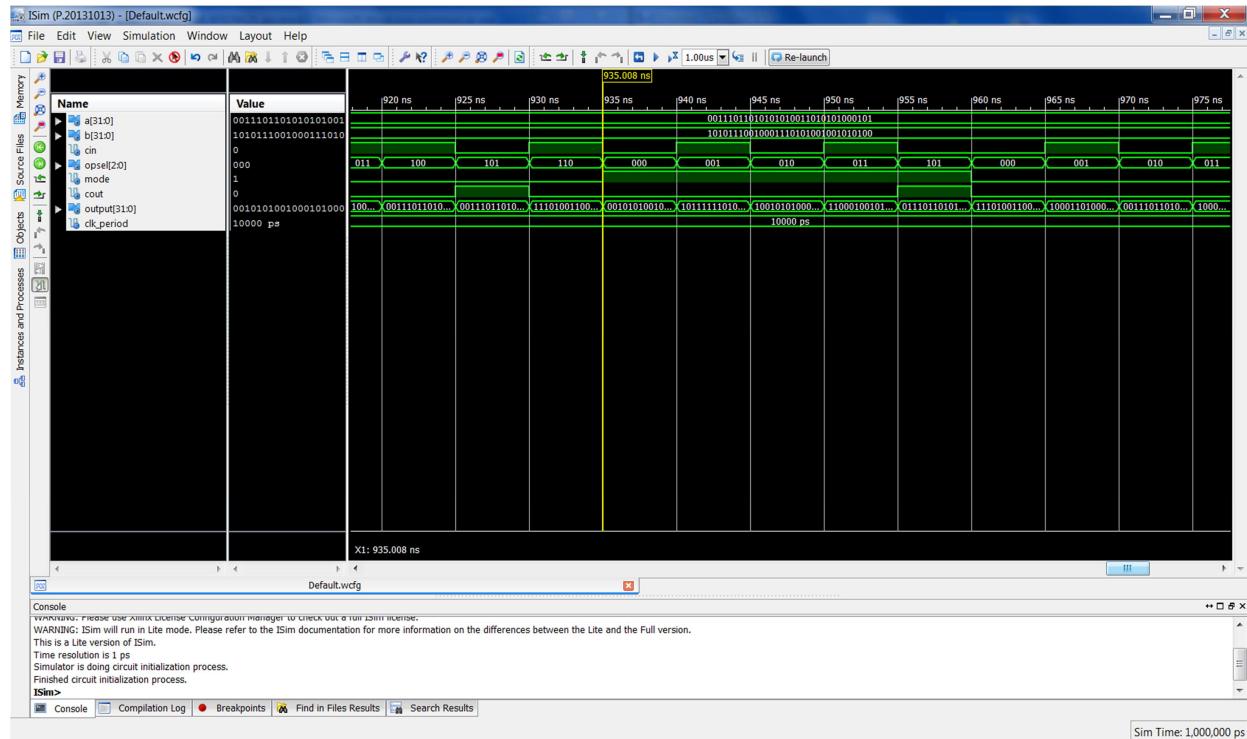
Process "Simulate Behavioral Model" completed successfully

6 Waveforms

Full Waveform Snapshot 1



Full Waveform Snapshot 2



Snapshots of Waveform

a+b

Name	Value	900 ns	902 ns	904 ns
► a[31:0]	0011101101010101010011010101000101			
► b[31:0]	10101110010001110101001001010100			
└ cin	0		000	
► opsel[2:0]	000			
└ mode	0			
└ cout	0			
► output[31:0]	11101001100111001000011110011001	11101001100110010000011110011001		
└ clk_period	10000 ps			

a-b'

Name	Value	906 ns	908 ns
► a[31:0]	00111011010101010011010101000101	00111011010101010011010101000101	
► b[31:0]	10101110010001110101001001010100	10101110010001110101001001010100	
└ cin	1		001
► opsel[2:0]	001		
└ mode	0		
└ cout	0		
► output[31:0]	10001101000011011110001011110001	10001101000011011110001011110001	
└ clk_period	10000 ps		10000 ps

a

Name	Value	910 ns	912 ns	914 ns
► a[31:0]	00111011010101010011010101000101			
► b[31:0]	10101110010001110101001001010100			
└ cin	0		010	
► opsel[2:0]	010			
└ mode	0			
└ cout	0			
► output[31:0]	00111011010101010011010101000101	00111011010101010011010101000101		
└ clk_period	10000 ps			

a+b'+1

Name	Value	916 ns	918 ns
► a[31:0]	00111011010101010011010101000101	00111011010101010011010101000101	
► b[31:0]	10101110010001110101001001010100	10101110010001110101001001010100	
└ cin	1		011
► opsel[2:0]	011		
└ mode	0		
└ cout	0		
► output[31:0]	10001101000011011110001011110001	10001101000011011110001011110001	
└ clk_period	10000 ps		10000 ps

a+1

Name	Value	920 ns	922 ns	924 ns
► a[31:0]	0011101101010101010011010101000101	00101		
► b[31:0]	10101110010001110101001001010100	10100		
► cin	1			
► opsel[2:0]	100		100	
► mode	0			
► cout	0			
► output[31:0]	00111011010101010011010101000110	00111011010101010011010101000110		
clk_period	10000 ps			

a-1

Name	Value	926 ns	928 ns
► a[31:0]	0011101101010101010011010101000101	00111011010101010011010101000101	
► b[31:0]	10101110010001110101001001010100	10101110010001110101010101000100	
► cin	0		
► opsel[2:0]	101		101
► mode	0		
► cout	1		
► output[31:0]	00111011010101010011010101000100	00111011010101010011010101000100	
clk_period	10000 ps		10000 ps

a+b+1

Name	Value	930 ns	932 ns	934 ns
► a[31:0]	0011101101010101010011010101000101	0101000101		
► b[31:0]	10101110010001110101001001010100	1001010100		
► cin	1			
► opsel[2:0]	110		110	
► mode	0			
► cout	0			
► output[31:0]	11101001100111001000011110011010	11101001100111001000011110011010		
clk_period	10000 ps			

a AND b

Name	Value	936 ns	938 ns
► a[31:0]	0011101101010101010011010101000101		
► b[31:0]	10101110010001110101001001010100		
► cin	0		
► opsel[2:0]	000		000
► mode	1		
► cout	0		
► output[31:0]	00101010010001010001000001000100	00101010010001010001000001000100	
clk_period	10000 ps		

a OR b

Name	Value	940 ns	942 ns	944 ns
► a[31:0]	0011101101010101010011010101000101	11010101010011010101000101	10101010001001110101001001010100	
► b[31:0]		1001000111010100100101010100		
└ cin	1			
► opsel[2:0]	001		001	
└ mode	1			
└ cout	0			
► output[31:0]	10111111010101110111011101010101	10111111010101110111011101010101		
└ clk_period	10000 ps	10000 ps		

a XOR b

Name	Value	946 ns	948 ns
► a[31:0]	0011101101010101010011010101000101		
► b[31:0]	10101110010001110101001001001010100		
└ cin	0		
► opsel[2:0]	010		010
└ mode	1		
└ cout	0		
► output[31:0]	10010101000100100110011100010001	10010101000100100110011100010001	
└ clk_period	10000 ps		

a'

Name	Value	950 ns	952 ns	954 ns
► a[31:0]	0011101101010101010011010101000101	100101		
► b[31:0]	10101110010001110101001001001010100	10100		
└ cin	1			
► opsel[2:0]	011		011	
└ mode	1			
└ cout	0			
► output[31:0]	11000100101010101100101010111010	11000100101010101100101010111010		
└ clk_period	10000 ps			

shift left a

Name	Value	956 ns	958 ns
► a[31:0]	0011101101010101010011010101000101		
► b[31:0]	10101110010001110101001001001010100		
└ cin	0		
► opsel[2:0]	101		101
└ mode	1		
└ cout	1		
► output[31:0]	01110110101010100110101010001010	01110110101010100110101010001010	
└ clk_period	10000 ps		