EE533 Lab 2 Report

Introduction

In this Lab, we use Xilinx ISE 10.1 running on the provided Windows XP VM to practice a complete FPGA design flow from both schematic and Verilog to implementation. We design and simulate a synchronous 8-bit adder from 8 1-bit full adder and D flip-flops without using IP cores (i.e., using gate level logic), then extended the design hierarchically into a 32-bit ALU that supports addition, subtraction, shifting and bitwise and & or logic. Finally, we implement and equivalent 32-bit ALU in Verilog, mapping both the schematic and RTL designs, and compare their mapper reports and waveform results to understand the resource usage tradeoffs in LUT and flip-flops from both designs.

Part 1: Xilinx ISE 10.1 In-Depth Tutorial and Reproduction of Demo Design

In this part, we review *ISE In-Depth Tutorial* and get familiar with the design environment using Windows XP based VM in VMWare Workstation Pro.

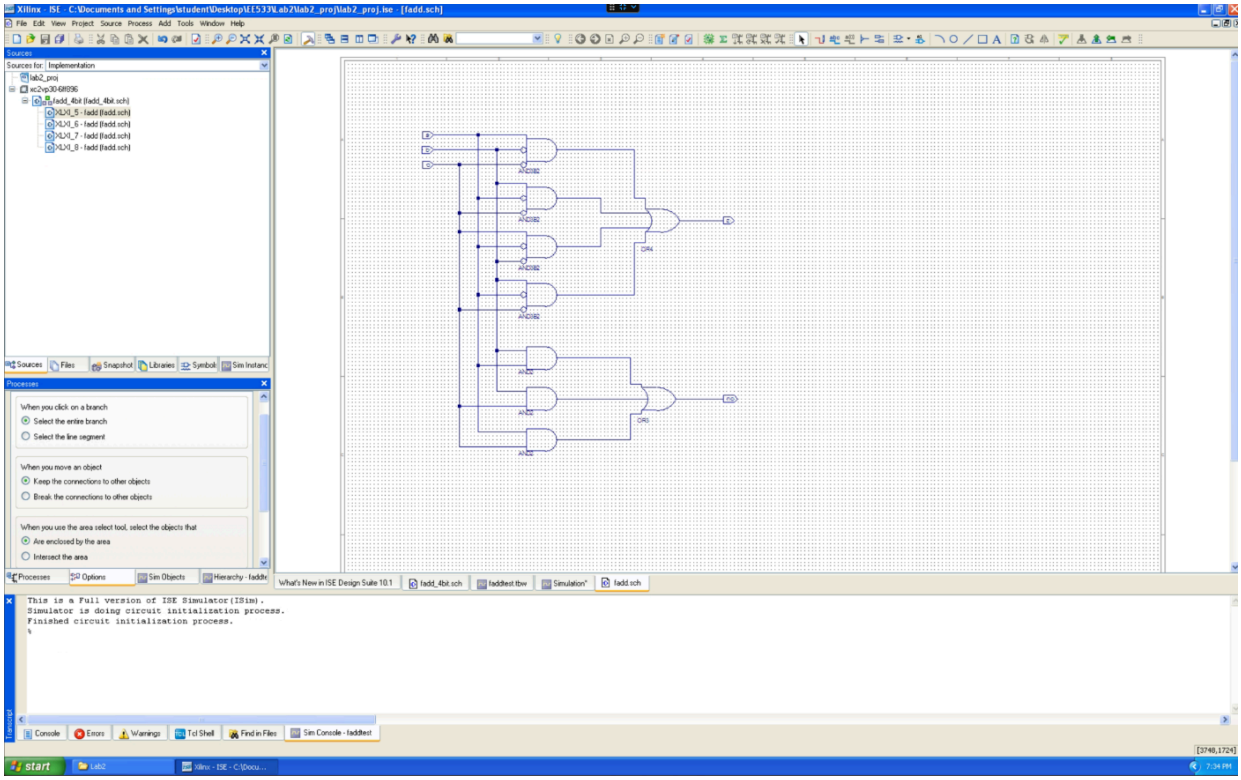

Figure 1. Reproduce Demo Video Full Adder Design

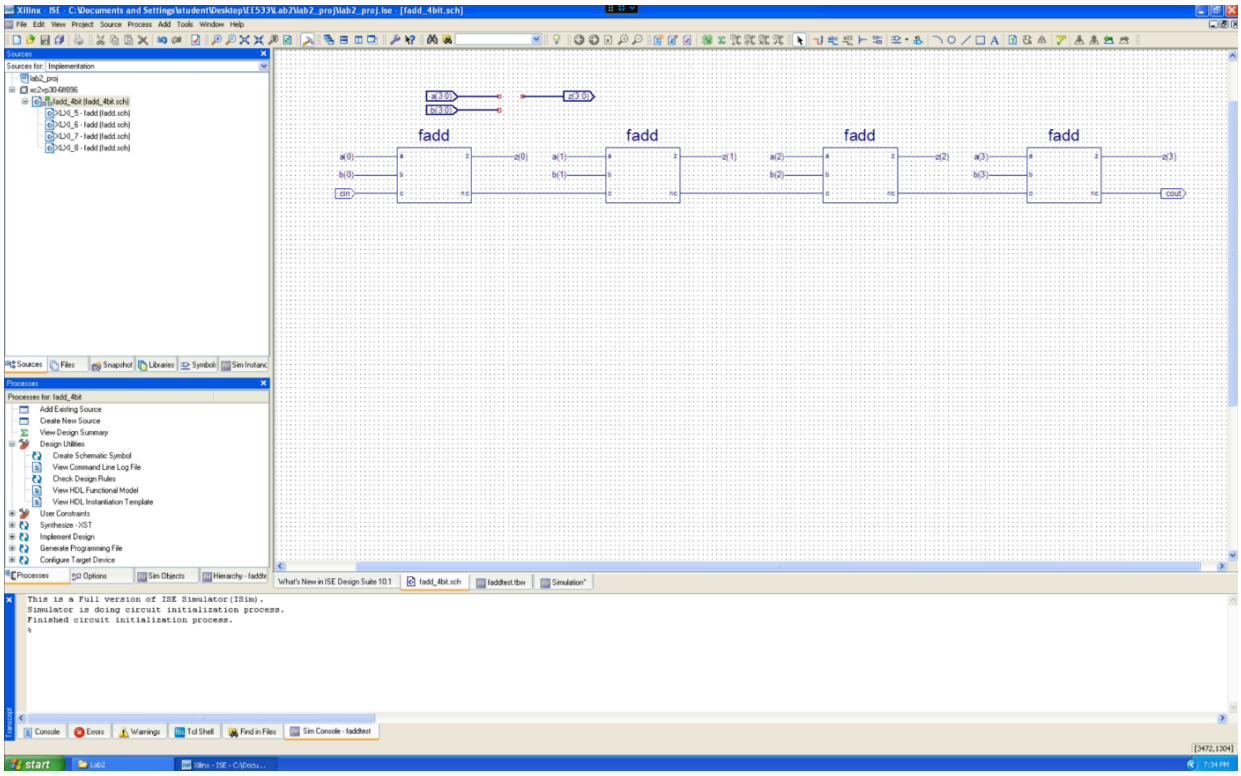

Figure 2. Reproduce Demo Video 4-Bit Full Adder Design

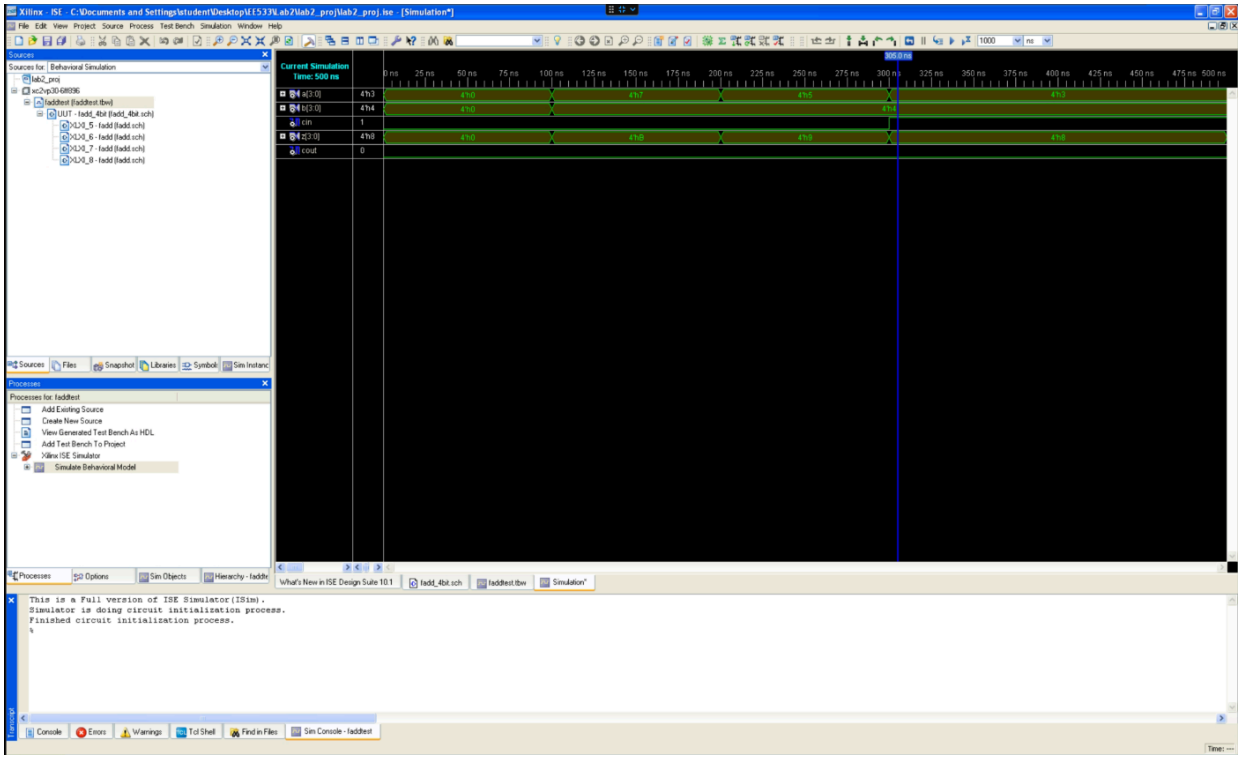

Figure 3. Function Simulation for 4-bit Full Adder Design

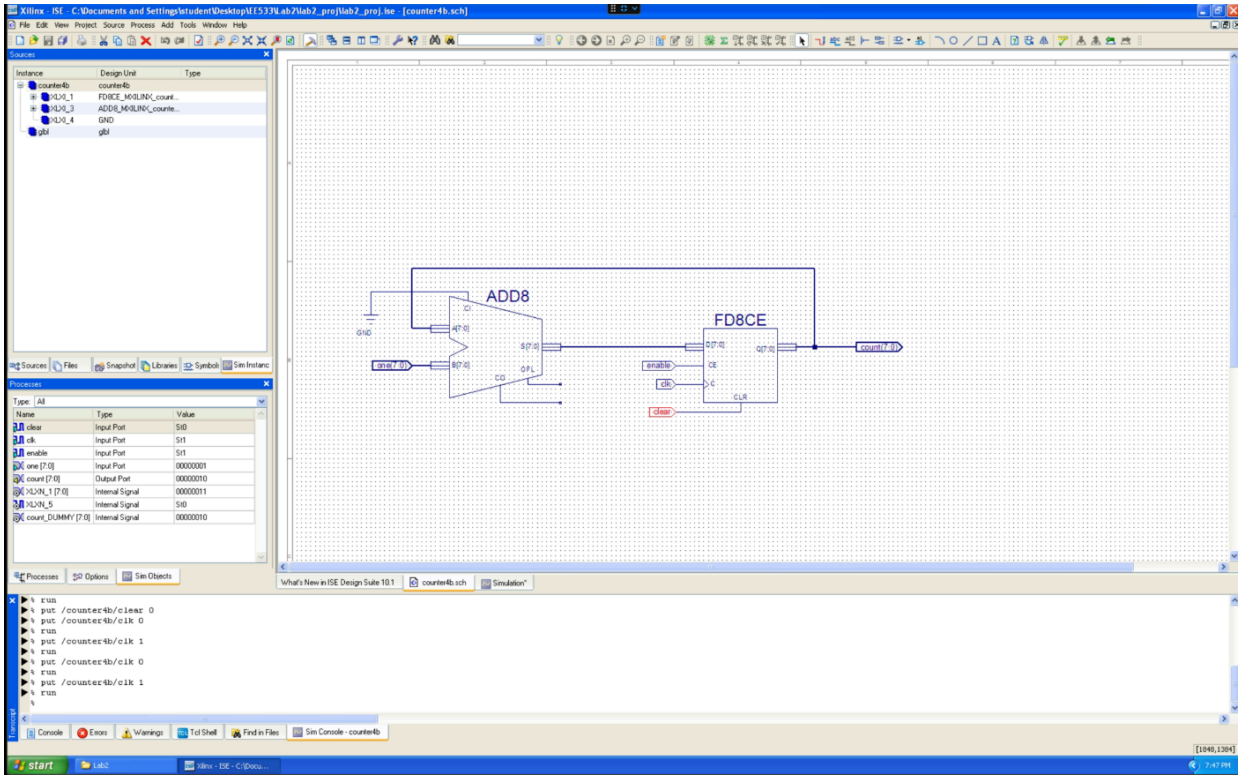

Figure 4. Reproduce Demo Video Counter Design

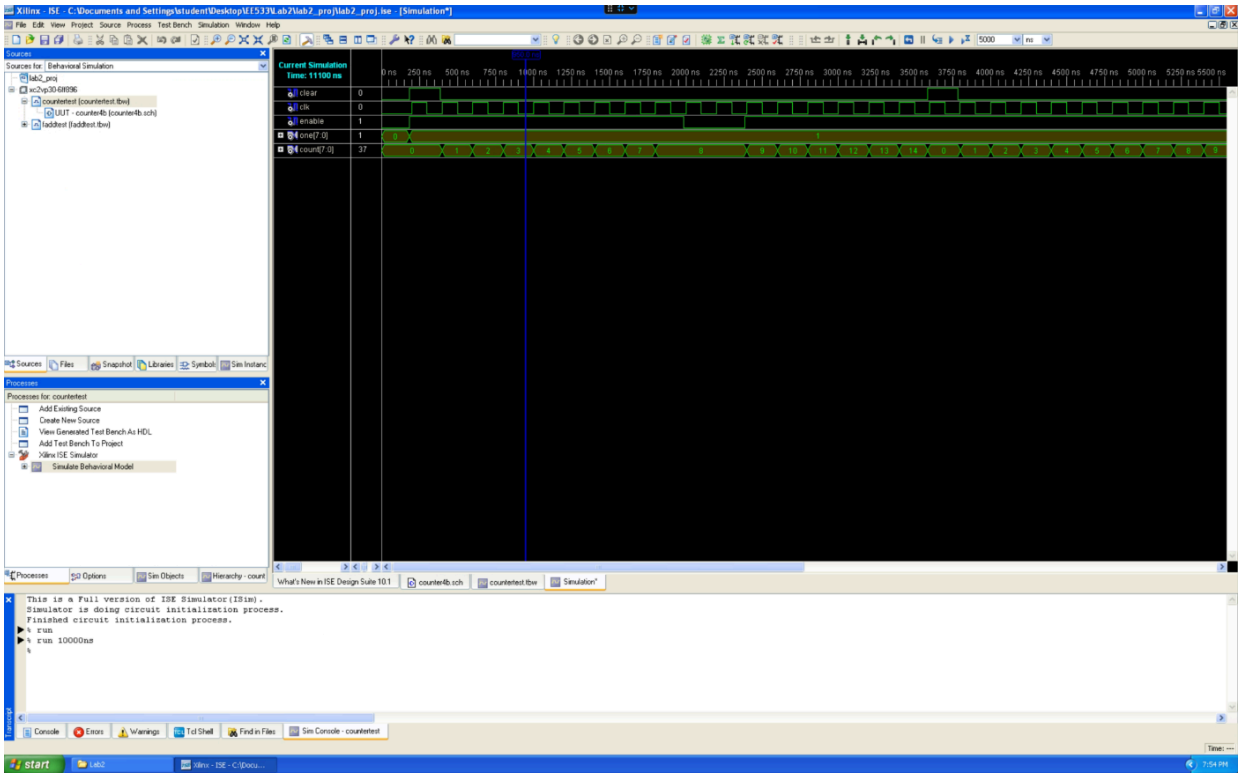

Figure 5. Function Simulation of Counter Design

Part II: Designing and Simulating Synchronous 8-Bit Adder

In this part, we redesign the full adder with a new schematic provided in the lab manual and add D flip-flops before the input pins and after the output pins of the adder circuit to create a synchronous 8-bit adder

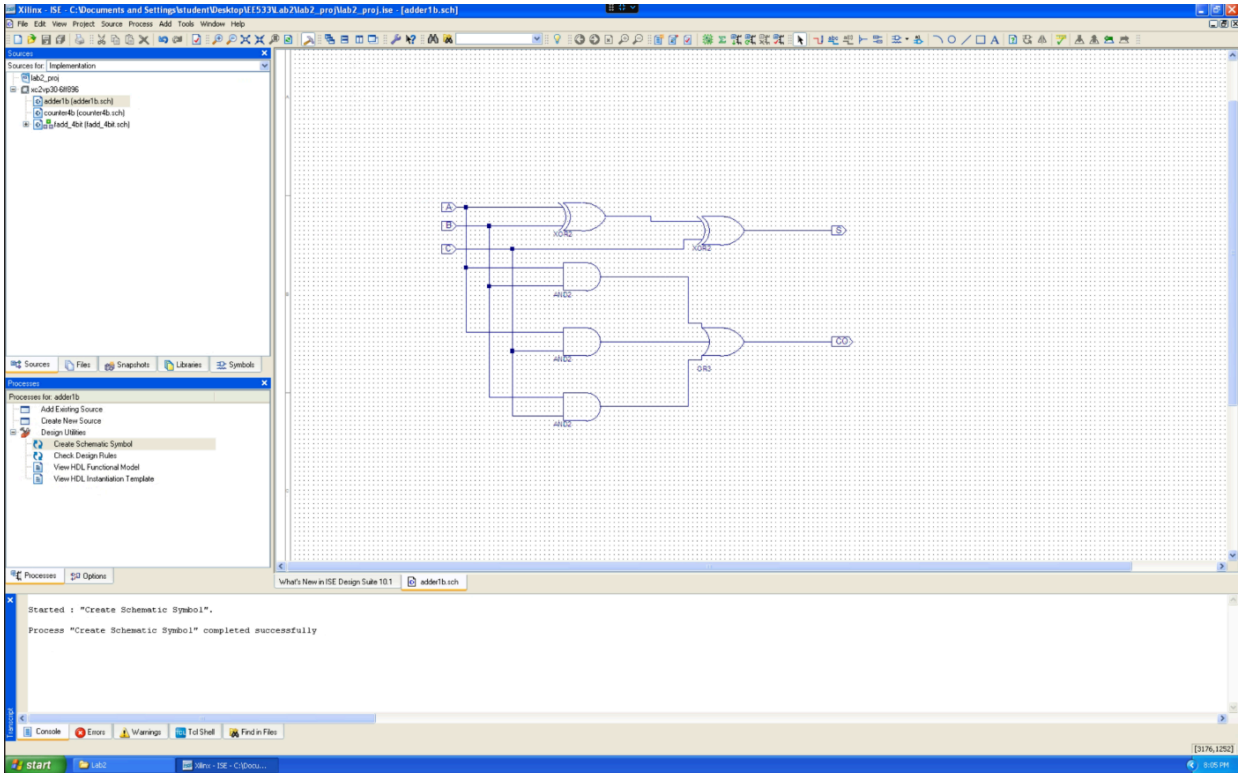

Figure 6. Newly Designed 1-Bit Full Adder

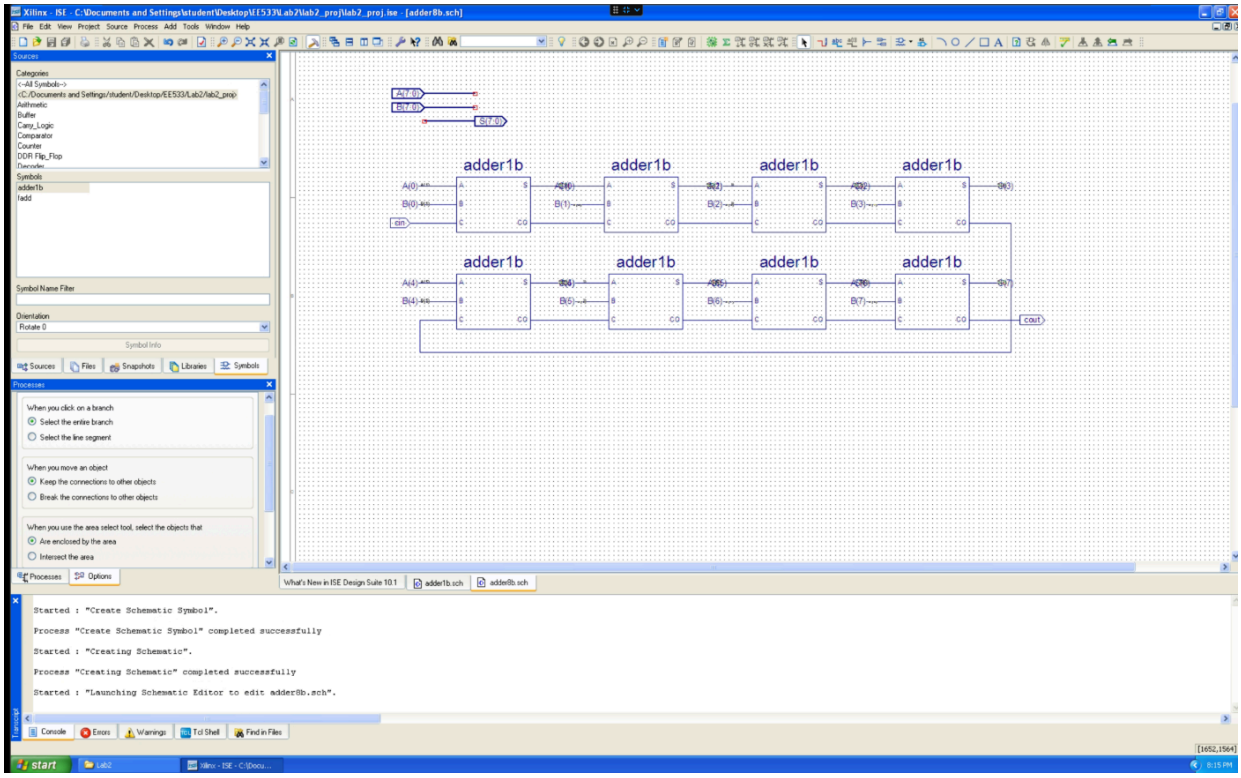

Figure 7. 8-Bit Full Adder Based on New Schematic

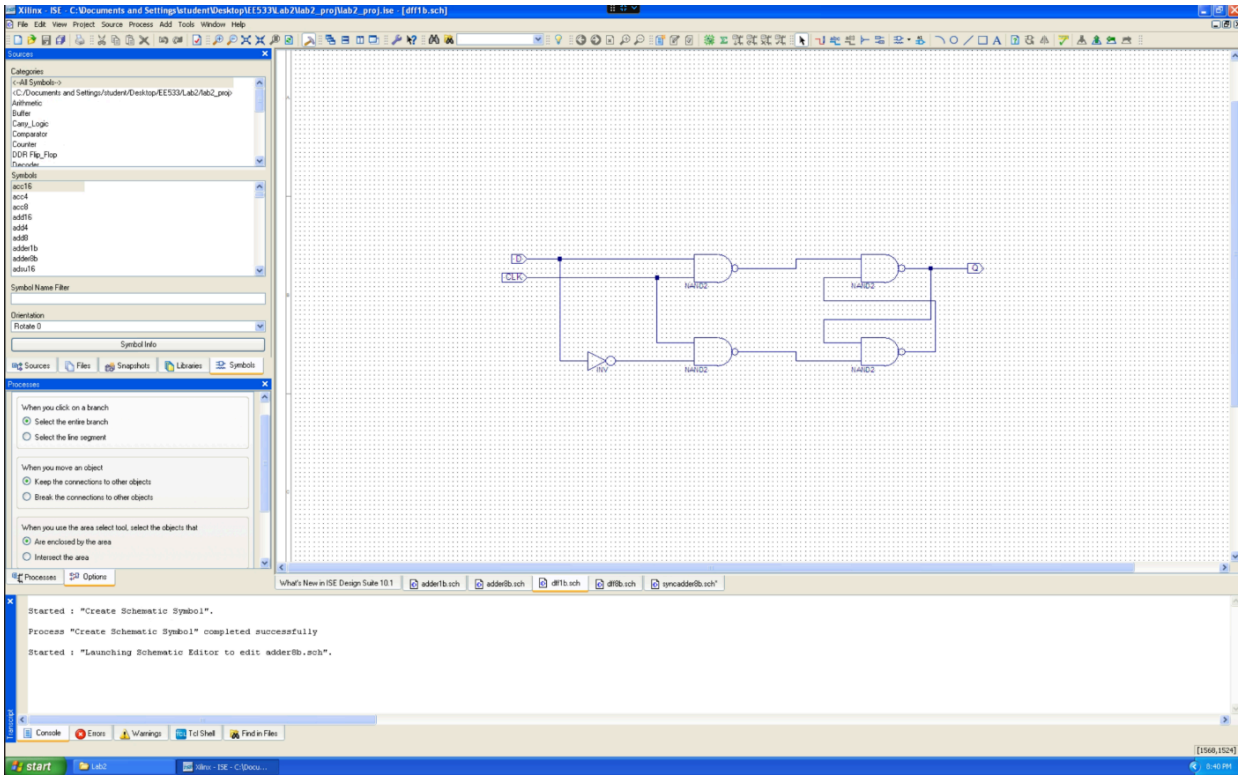

Figure 8. D Flip-Flop Schematic Design

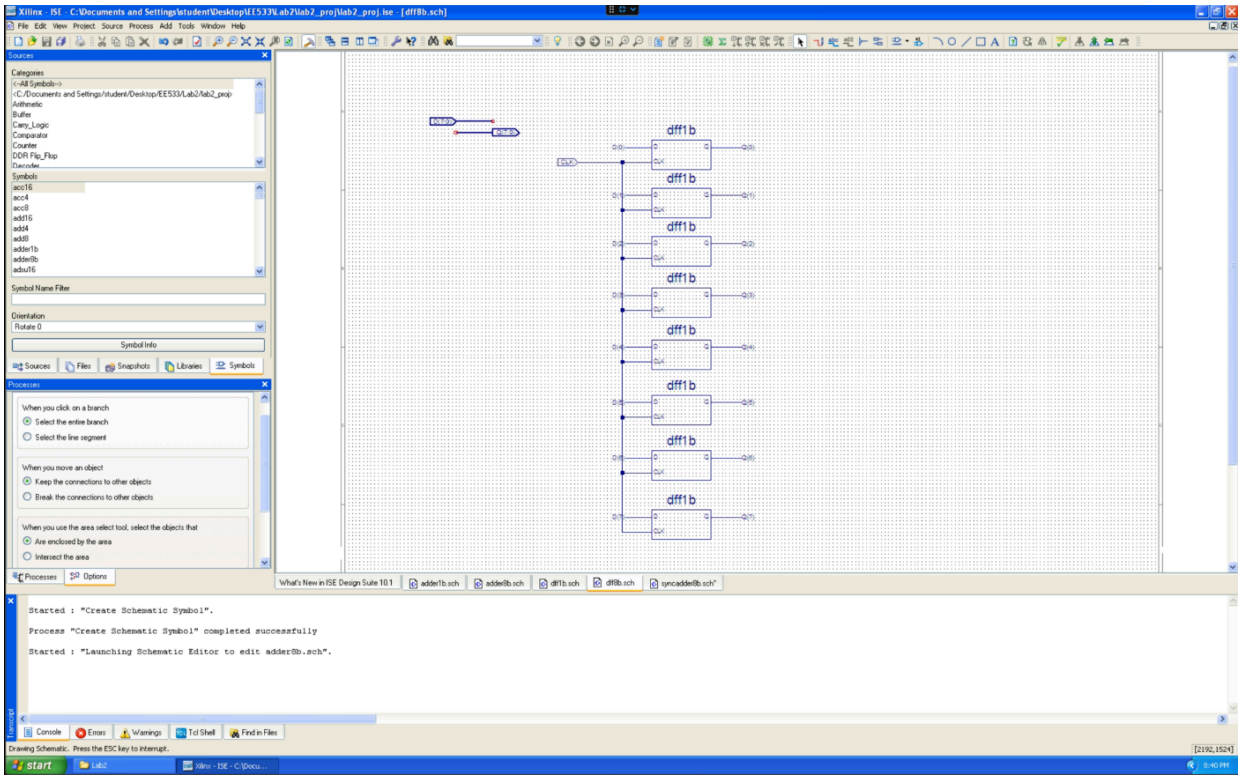

Figure 9. 8-Bit D Flip-Flop Design

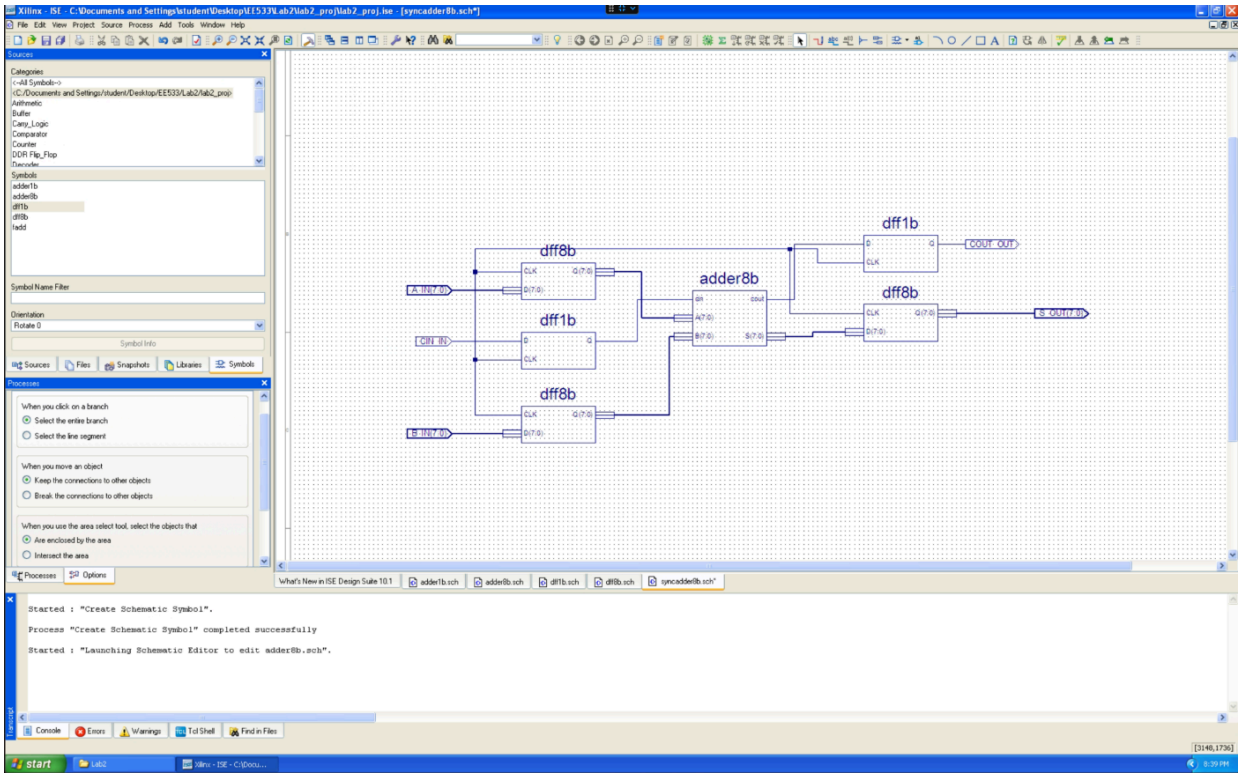

Figure 10. 8-Bit Full Adder with D Flip-Flop Design

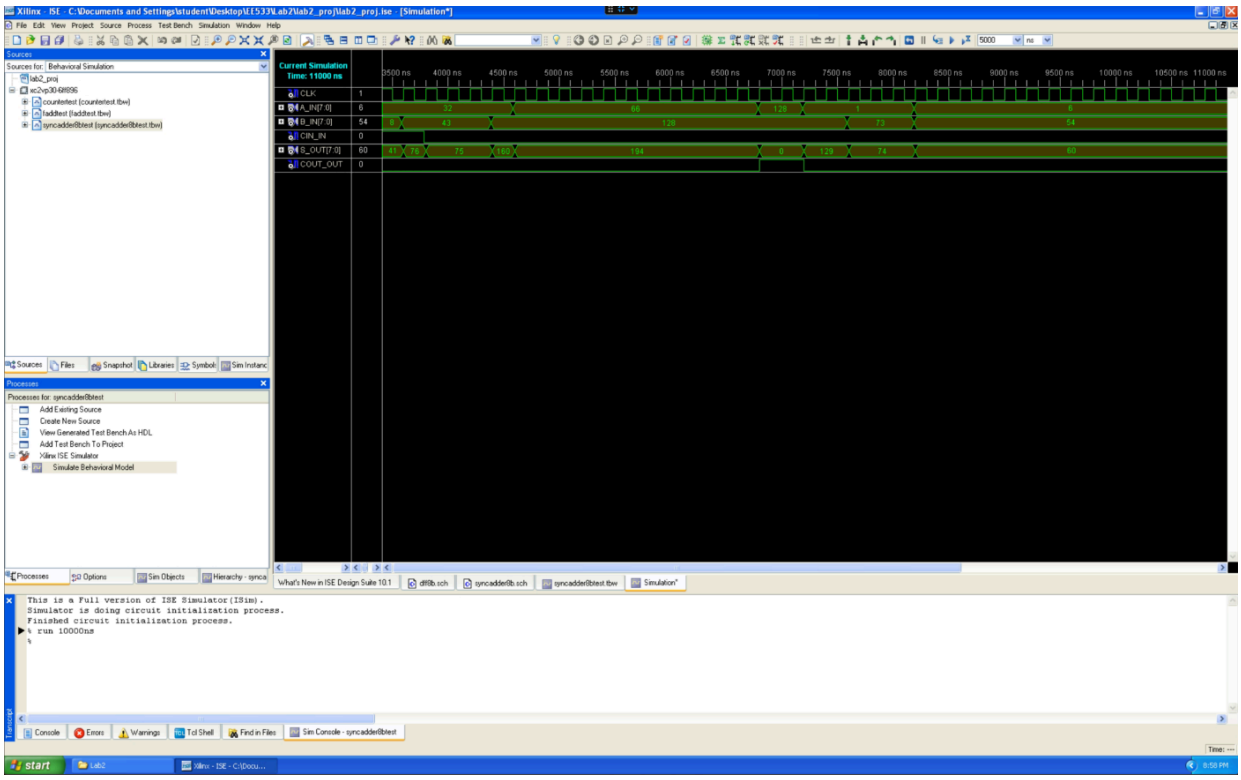

Figure 11. 8-Bit Full Adder with D Flip-Flop Design Simulation Result

Part III: Extending Adder into 32-Bit ALU

We divide this whole part into 3 sub-parts for A) Extend 8-bit adder into 32-bit adder B) Extend 32-bit adder to have other functions including subtractor, shifter, bitwise and and bitwise or c) Write Verilog equivalent of 32-bit ALU

PART III (A): EXTEND 8-BIT ADDER INTO 32-BIT ADDER

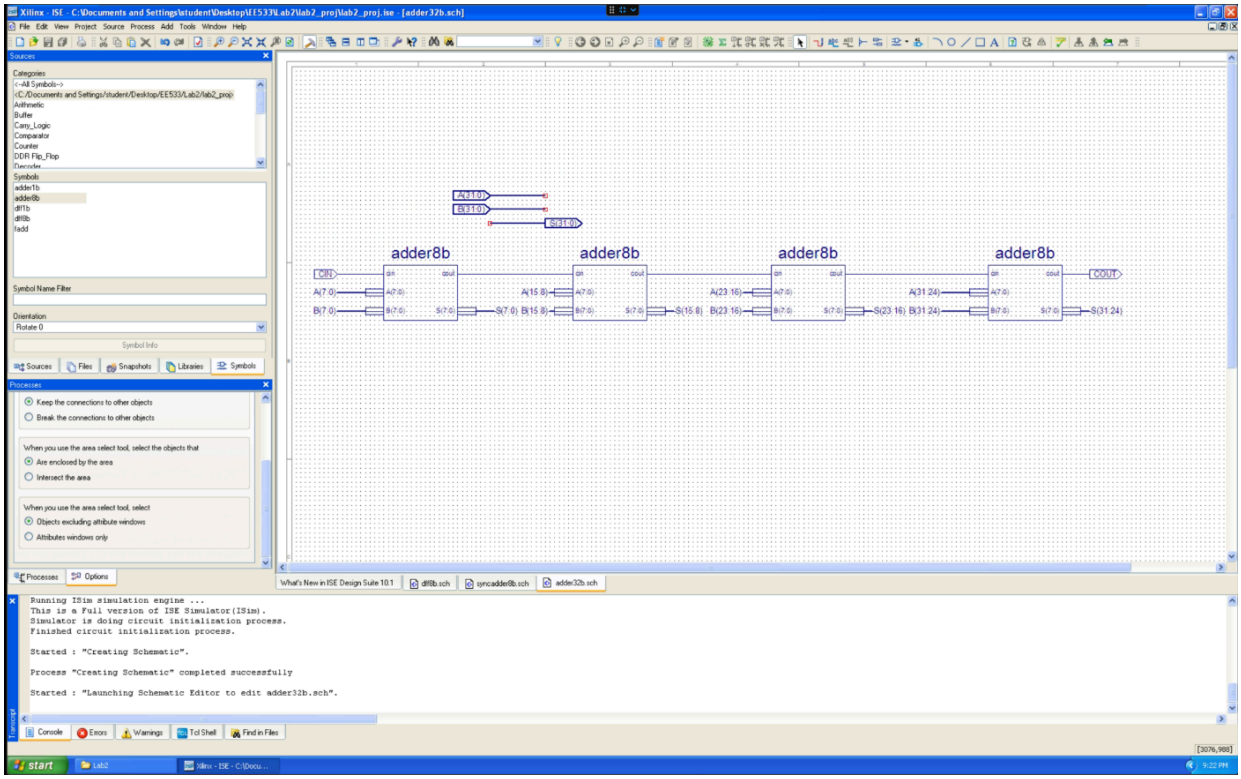

Figure 12. 32-Bit Adder Design Using 4 8-Bit Adders

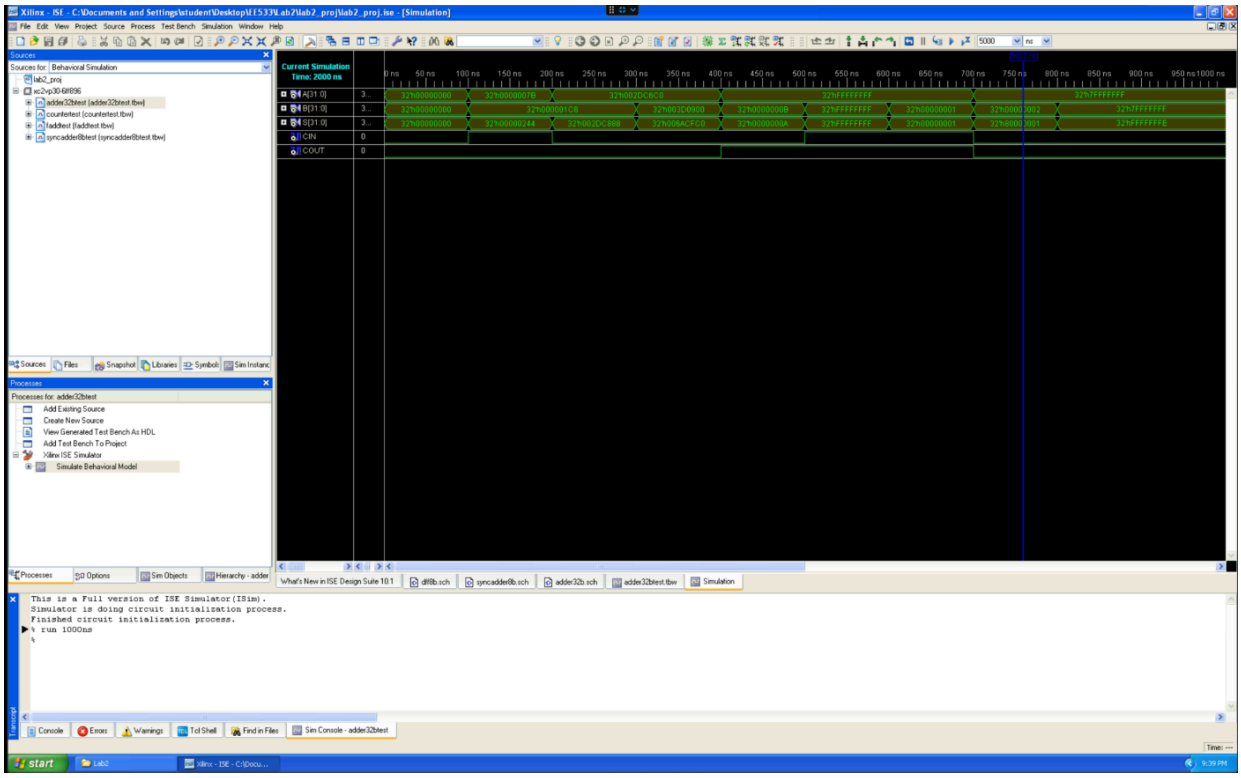

Figure 13. 32-Bit Adder Function Verification

PART III (B): 32-BIT ALU DESIGN

Implemented design for 32 bit adder/subtractor (reuse), left/right shifter (logic) one bit

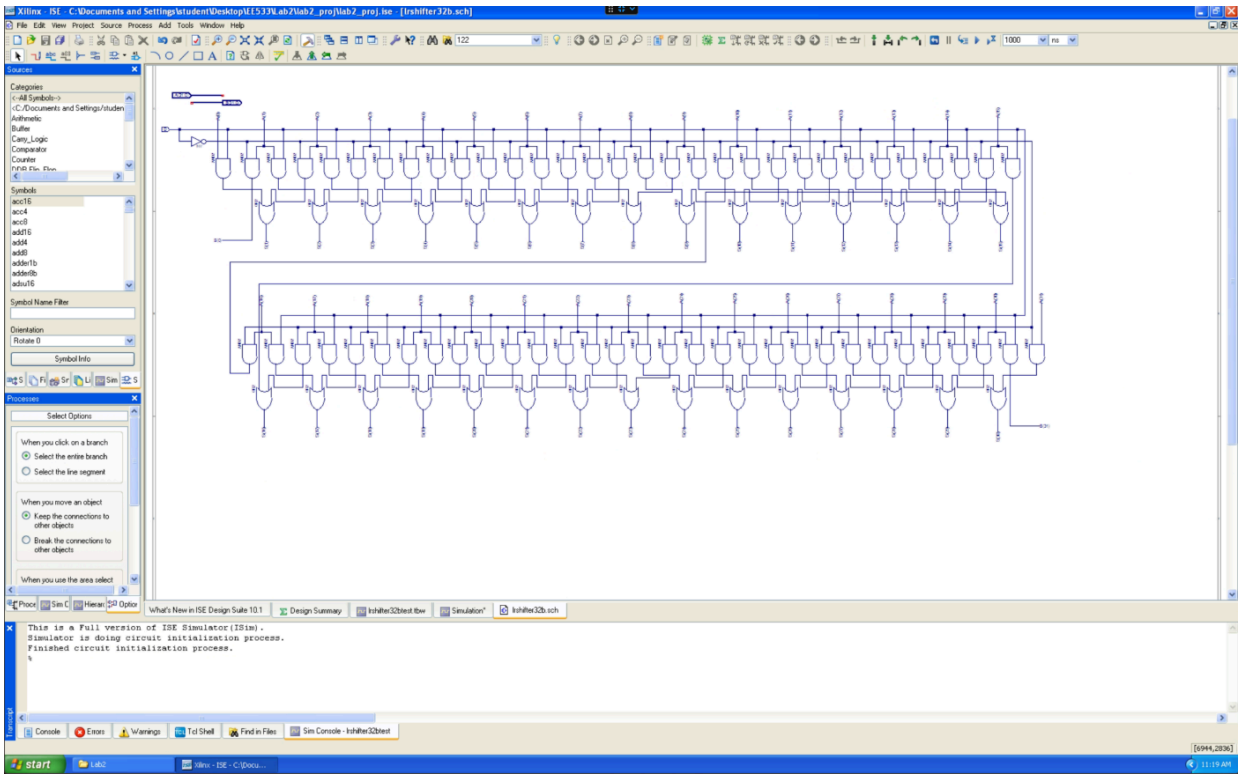

Figure 14. 32-Bit Left and Right Shifter Design

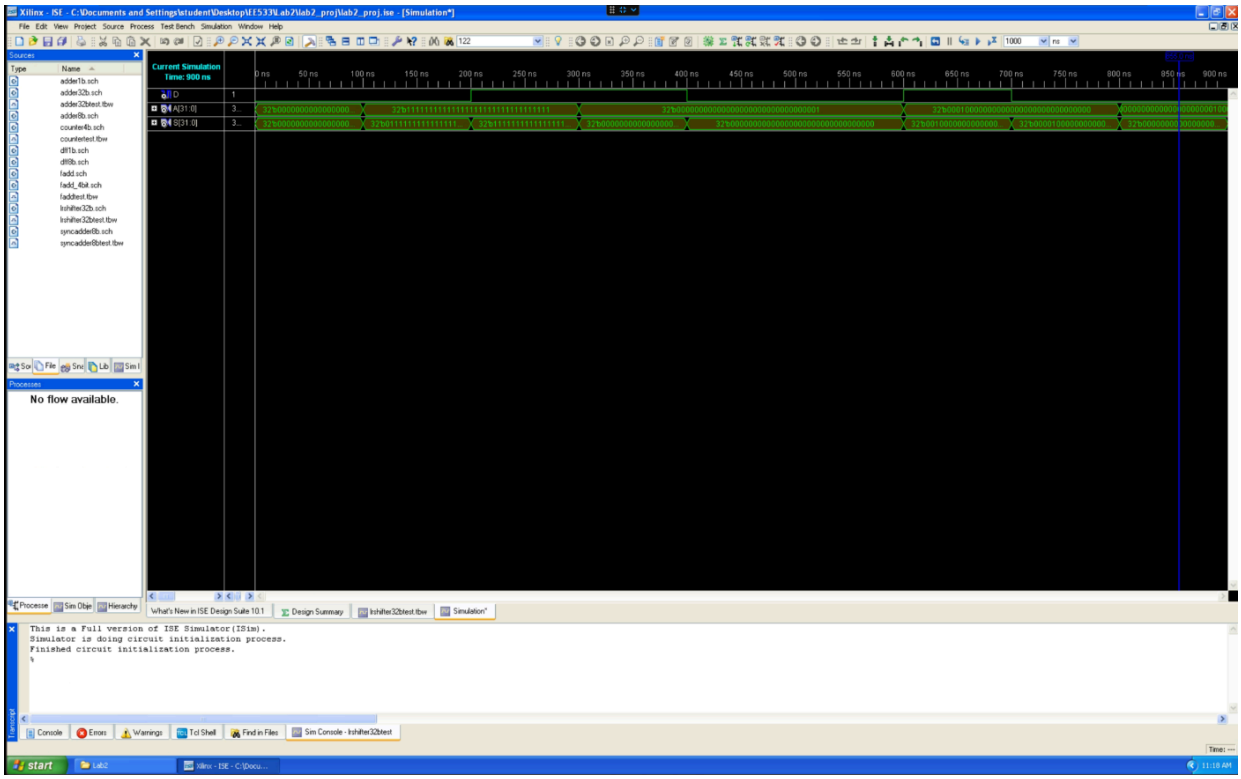

Figure 15. 32-Bit Left and Right Shifter Function Simulation

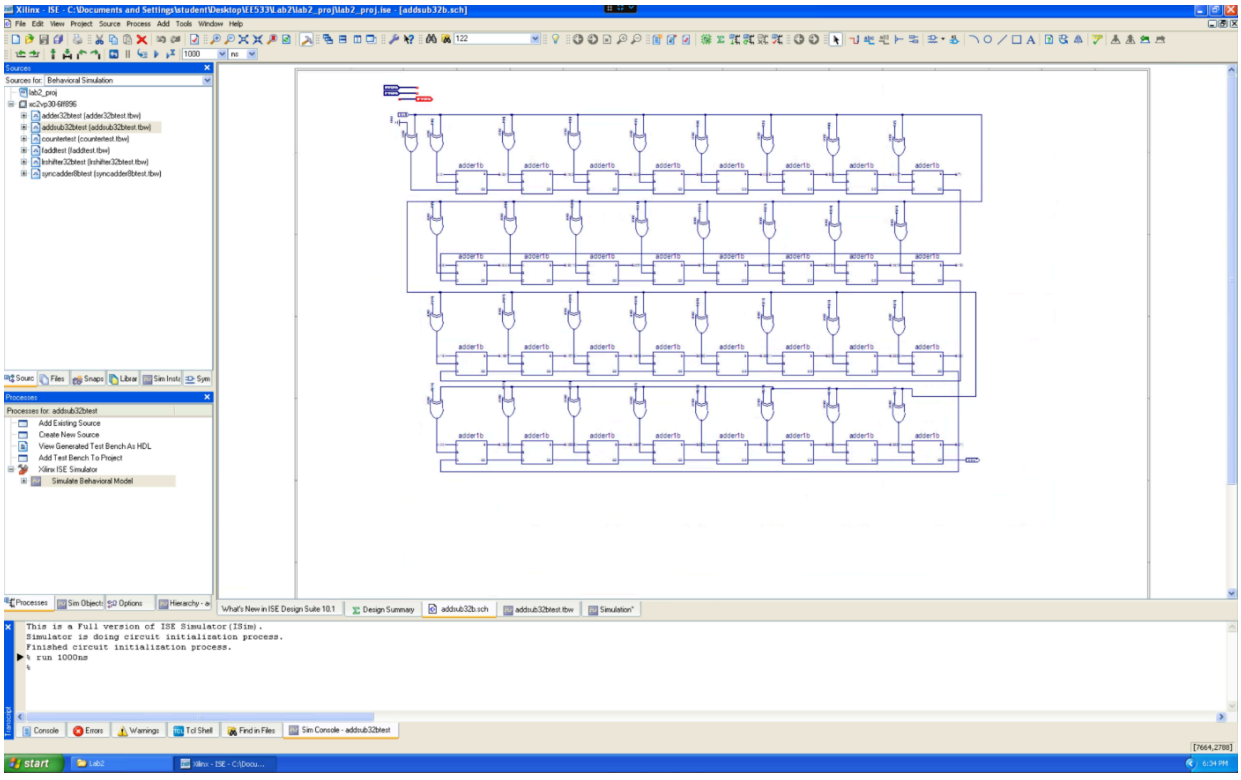

Figure 16. 32-Bit Add/Sub Unit Design

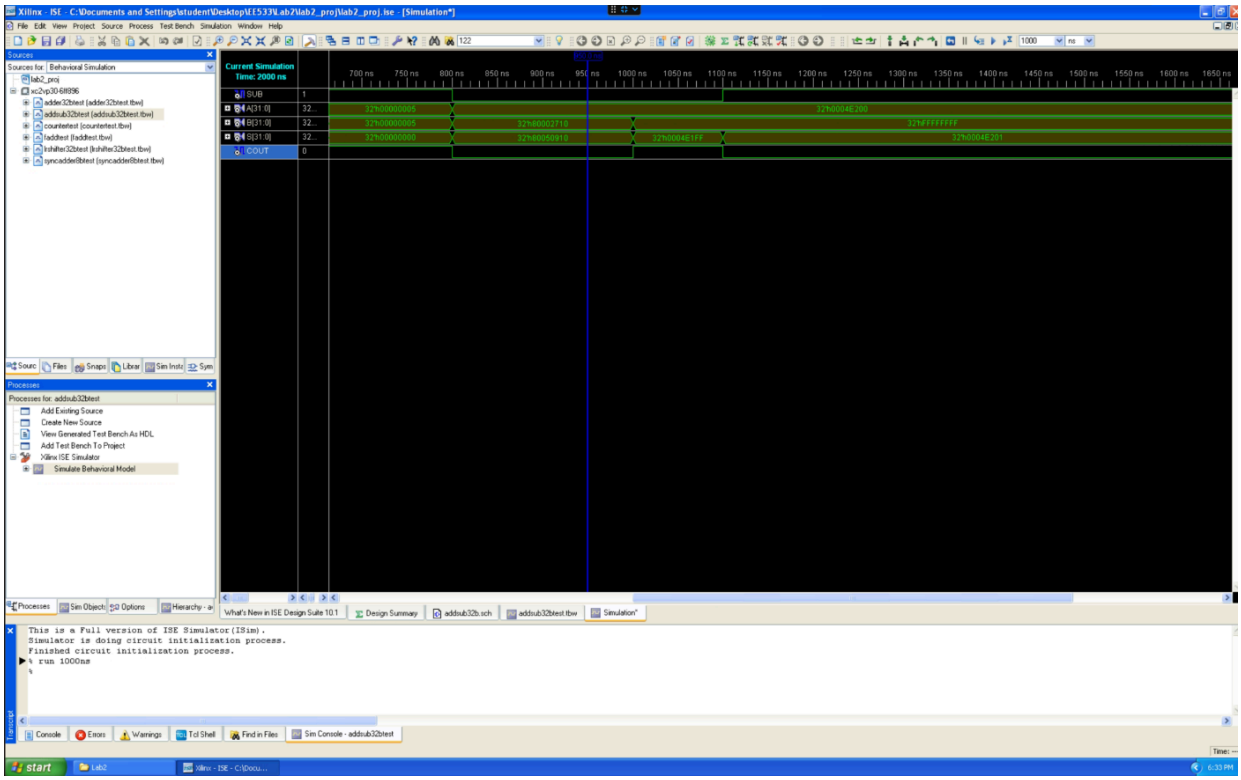

Figure 17. 32-Bit Add/Sub Unit Function Verification

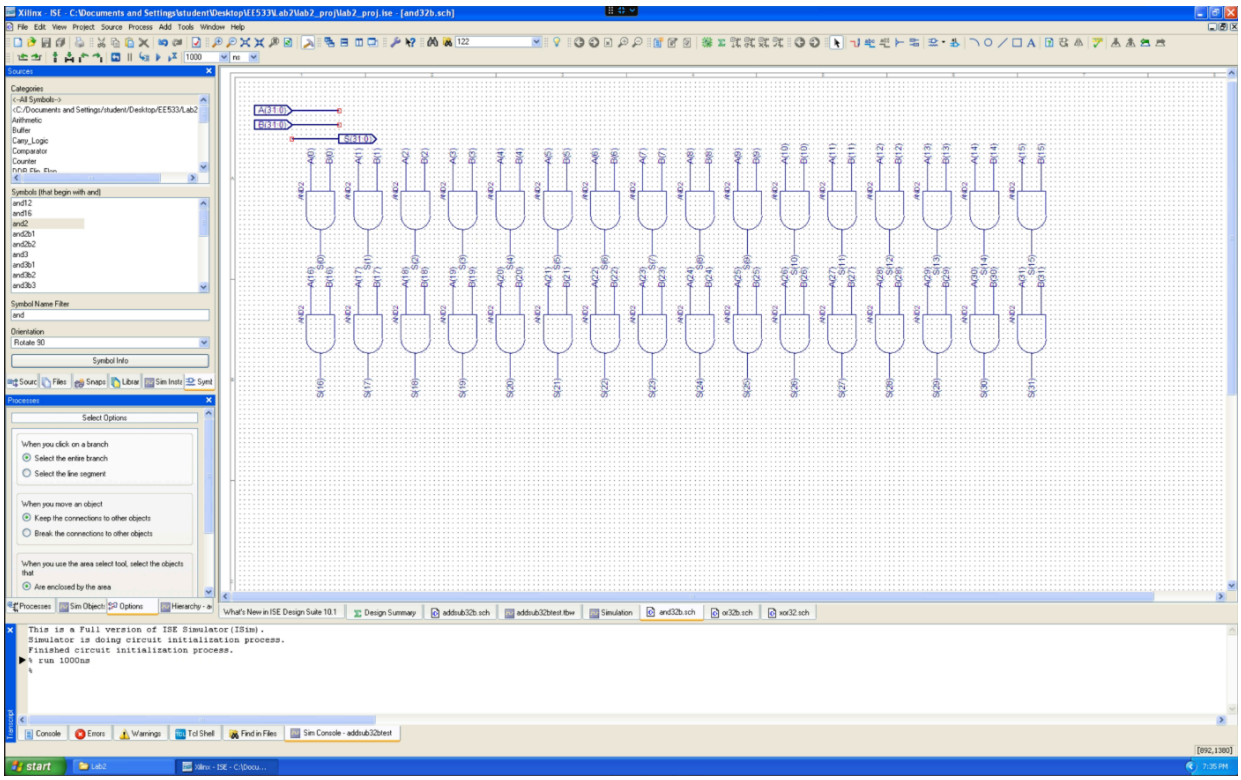

Figure 18. 32-Bit Bitwise ADD Function Unit Design

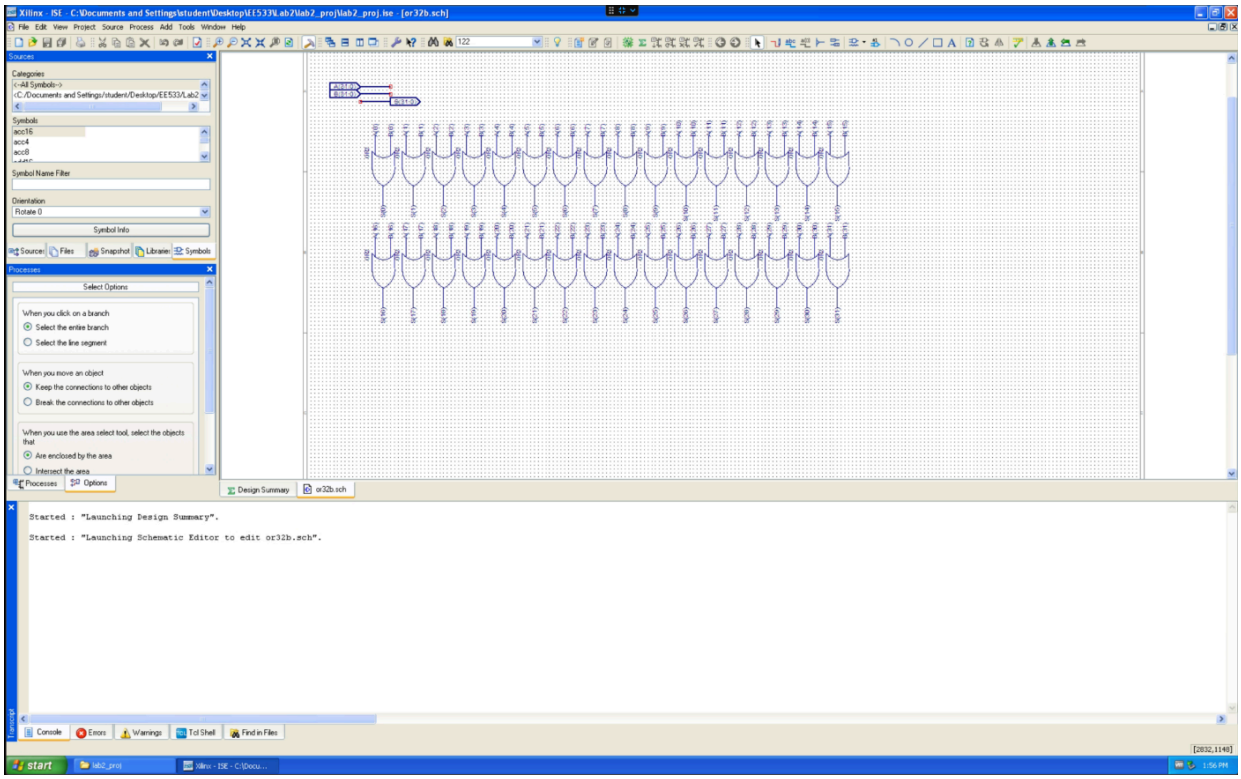

Figure 19. 32-Bit Bitwise OR Function Unit Design

Here since the bitwise add and or unit designs are trivial, we don't include the function simulation result in this report.

There are 6 total functions for this ALU: ADD, SUB, shift left, shift right, bitwise AND & bitwise OR. In this case, we need 3-bit opcode for this design to identify the operation types.

	Function	Opcode
1	ADD	3'b000
2	SUB	3'b001
3	Shift Left	3'b010
4	Shift Right	3'b011
5	AND	3'b100
6	OR	3'b101
7	Reserved	3'b110
8	Reserved	3'b111

We define the ALU interface and A(31:0), B(31:0), OPCODE(2:0) as inputs, and S(31:0) as output. Also, since we decide to make different function units work in parallel, we therefore only need to generate two signals from the opcode: SUB and D (for shifting. 1: shift left 0: shift right)

Since there are only 4 function unites, the output selection multiplexer needs 2-bit to select from 3 unites:

	Function Unit	Selection Code
1	ADD/SUB	2'b00
2	Shifter	2'b01
3	AND	2'b10
4	OR	2'b11

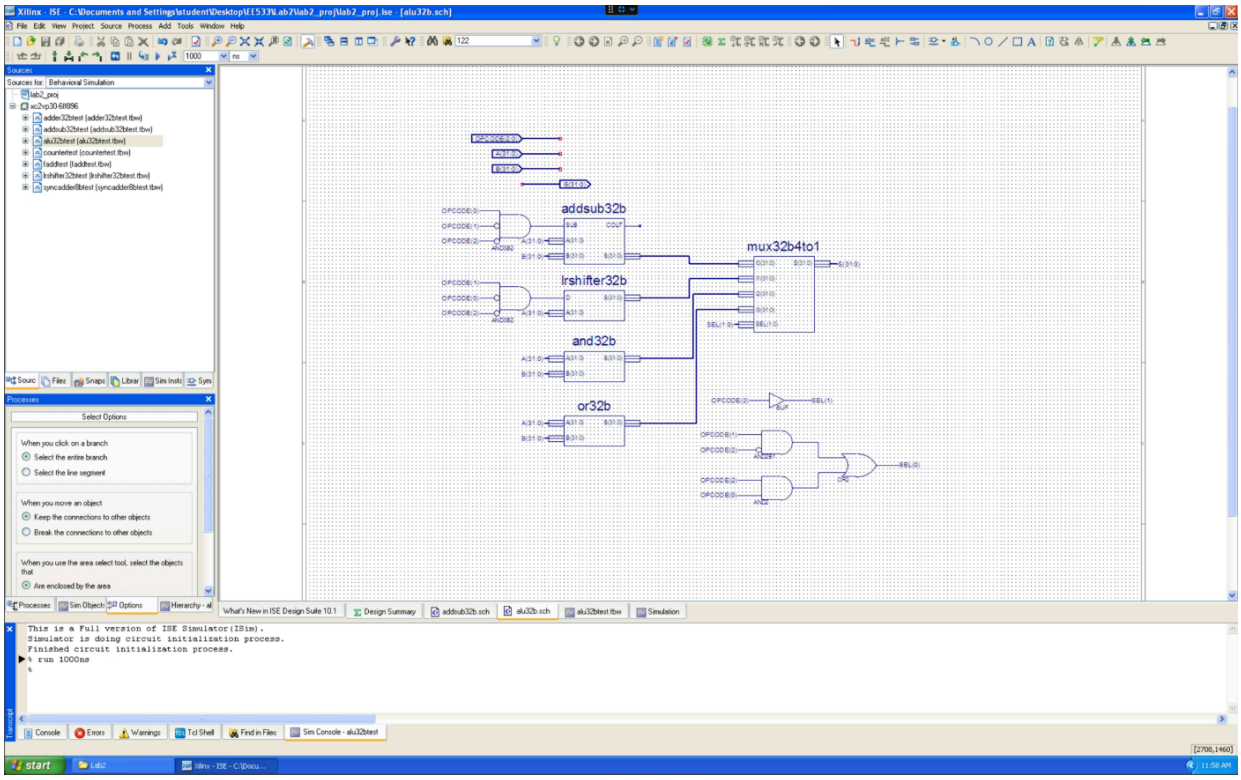

Figure 20. 32-Bit ALU Design Schematic

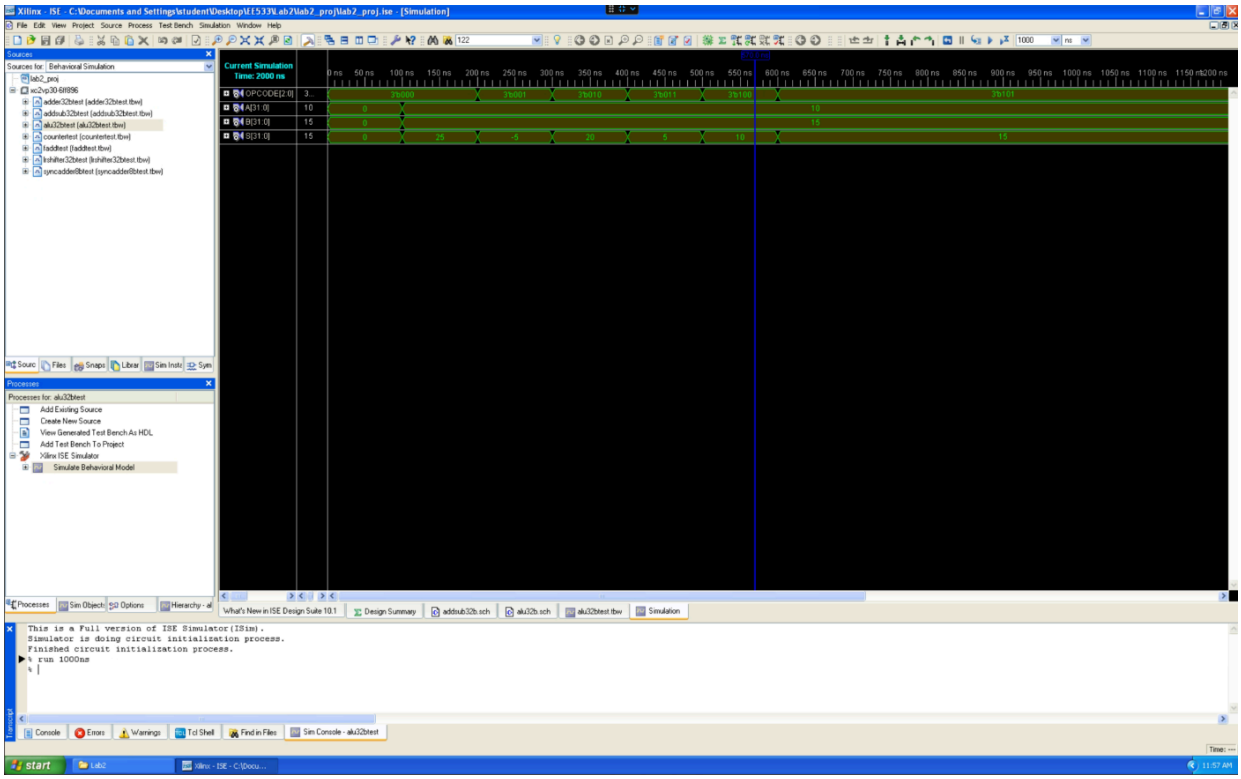

Figure 21. 32-Bit ALU Design Function Simulation

Mapper Report for ALU design based on hierarchical schematic design:

```
Release 10.1 Map K.31 (nt)
Xilinx Mapping Report File for Design 'alu32b'

Design Information
------------------
Command Line   : map -ise "C:/Documents and
Settings/student/Desktop/EE533/Lab2/lab2_proj/lab2_proj.ise" -intstyle ise -p
xc2vp30-ff896-6 -cm area -pr off -k 4 -c 100 -tx off -o alu32b_map.ncd
alu32b.ngd alu32b.pcf
Target Device  : xc2vp30
Target Package : ff896
Target Speed   : -6
Mapper Version : virtex2p -- $Revision: 1.46 $
Mapped Date    : Fri Jan 23 12:04:31 2026

Design Summary
--------------
Number of errors:      0
Number of warnings:    0
Logic Utilization:
  Number of 4 input LUTs:                131 out of  27,392    1%
Logic Distribution:
  Number of occupied Slices:             130 out of  13,696    1%
    Number of Slices containing only related logic:     130 out of     130 100%
    Number of Slices containing unrelated logic:          0 out of     130   0%
```

```
      *See NOTES below for an explanation of the effects of unrelated logic.
  Total Number of 4 input LUTs:         226 out of  27,392    1%
    Number used as logic:               131
    Number used as a route-thru:         95
  Number of bonded IOBs:                 99 out of     556   17%

Peak Memory Usage:  199 MB
Total REAL time to MAP completion:  1 secs
Total CPU time to MAP completion:   1 secs

NOTES:

   Related logic is defined as being logic that shares connectivity - e.g. two
   LUTs are "related" if they share common inputs.  When assembling slices,
   Map gives priority to combine logic that is related.  Doing so results in
   the best timing performance.

   Unrelated logic shares no connectivity.  Map will only begin packing
   unrelated logic into a slice once 99% of the slices are occupied through
   related logic packing.

   Note that once logic distribution reaches the 99% level through related
   logic packing, this does not mean the device is completely utilized.
   Unrelated logic packing will then begin, continuing until all usable LUTs
   and FFs are occupied.  Depending on your timing budget, increased levels of
   unrelated logic packing may adversely affect the overall timing performance
   of your design.

Table of Contents
-----------------
Section 1 - Errors
Section 2 - Warnings
Section 3 - Informational
Section 4 - Removed Logic Summary
Section 5 - Removed Logic
Section 6 - IOB Properties
Section 7 - RPMs
Section 8 - Guide Report
Section 9 - Area Group and Partition Summary
Section 10 - Modular Design Summary
Section 11 - Timing Report
Section 12 - Configuration String Information
Section 13 - Control Set Information
Section 14 - Utilization by Hierarchy

Section 1 - Errors
------------------

Section 2 - Warnings
--------------------

Section 3 - Informational
-------------------------
INFO:MapLib:562 - No environment variables are currently set.
INFO:LIT:244 - All of the single ended outputs in this design are using slew
```

```
   rate limited output drivers. The delay on speed critical single ended outputs
   can be dramatically reduced by designating them as fast outputs.

Section 4 - Removed Logic Summary
---------------------------------
   4 block(s) removed
   1 block(s) optimized away
   3 signal(s) removed

Section 5 - Removed Logic
-------------------------

The trimmed logic report below shows the logic removed from your design due to
sourceless or loadless signals, and VCC or ground connections.  If the removal
of a signal or symbol results in the subsequent removal of an additional signal
or symbol, the message explaining that second removal will be indented.  This
indentation will be repeated as a chain of related logic is removed.

To quickly locate the original cause for the removal of a chain of logic, look
above the place where that logic is listed in the trimming report, then locate
the lines that are least indented (begin at the leftmost edge).

Loadless block "XLXI_2/XLXI_56/XLXI_5" (OR) removed.
 The signal "XLXI_2/XLXI_56/XLXN_8" is loadless and has been removed.
  Loadless block "XLXI_2/XLXI_56/XLXI_6" (AND) removed.
 The signal "XLXI_2/XLXI_56/XLXN_7" is loadless and has been removed.
  Loadless block "XLXI_2/XLXI_56/XLXI_4" (AND) removed.
 The signal "XLXI_2/XLXI_56/XLXN_6" is loadless and has been removed.
  Loadless block "XLXI_2/XLXI_56/XLXI_3" (AND) removed.

Optimized Block(s):
TYPE         BLOCK
GND         XST_GND

To enable printing of redundant blocks removed and signals merged, set the
detailed map report option and rerun map.

Section 6 - IOB Properties
--------------------------

+-----------------------------------------------------------------------------------------
| IOB Name                           | Type             | Direction | IO Standard | Dr
|                                    |                  |           |             | St
+-----------------------------------------------------------------------------------------
| A<0>                               | IOB              | INPUT     | LVCMOS25    |
| A<1>                               | IOB              | INPUT     | LVCMOS25    |
| A<2>                               | IOB              | INPUT     | LVCMOS25    |
| A<3>                               | IOB              | INPUT     | LVCMOS25    |
| A<4>                               | IOB              | INPUT     | LVCMOS25    |
| A<5>                               | IOB              | INPUT     | LVCMOS25    |
| A<6>                               | IOB              | INPUT     | LVCMOS25    |
| A<7>                               | IOB              | INPUT     | LVCMOS25    |
| A<8>                               | IOB              | INPUT     | LVCMOS25    |
| A<9>                               | IOB              | INPUT     | LVCMOS25    |
| A<10>                              | IOB              | INPUT     | LVCMOS25    |
```

A<11>	IOB	INPUT	LVCMOS25
A<12>	IOB	INPUT	LVCMOS25
A<13>	IOB	INPUT	LVCMOS25
A<14>	IOB	INPUT	LVCMOS25
A<15>	IOB	INPUT	LVCMOS25
A<16>	IOB	INPUT	LVCMOS25
A<17>	IOB	INPUT	LVCMOS25
A<18>	IOB	INPUT	LVCMOS25
A<19>	IOB	INPUT	LVCMOS25
A<20>	IOB	INPUT	LVCMOS25
A<21>	IOB	INPUT	LVCMOS25
A<22>	IOB	INPUT	LVCMOS25
A<23>	IOB	INPUT	LVCMOS25
A<24>	IOB	INPUT	LVCMOS25
A<25>	IOB	INPUT	LVCMOS25
A<26>	IOB	INPUT	LVCMOS25
A<27>	IOB	INPUT	LVCMOS25
A<28>	IOB	INPUT	LVCMOS25
A<29>	IOB	INPUT	LVCMOS25
A<30>	IOB	INPUT	LVCMOS25
A<31>	IOB	INPUT	LVCMOS25
B<0>	IOB	INPUT	LVCMOS25
B<1>	IOB	INPUT	LVCMOS25
B<2>	IOB	INPUT	LVCMOS25
B<3>	IOB	INPUT	LVCMOS25
B<4>	IOB	INPUT	LVCMOS25
B<5>	IOB	INPUT	LVCMOS25
B<6>	IOB	INPUT	LVCMOS25
B<7>	IOB	INPUT	LVCMOS25
B<8>	IOB	INPUT	LVCMOS25
B<9>	IOB	INPUT	LVCMOS25
B<10>	IOB	INPUT	LVCMOS25
B<11>	IOB	INPUT	LVCMOS25
B<12>	IOB	INPUT	LVCMOS25
B<13>	IOB	INPUT	LVCMOS25
B<14>	IOB	INPUT	LVCMOS25
B<15>	IOB	INPUT	LVCMOS25
B<16>	IOB	INPUT	LVCMOS25
B<17>	IOB	INPUT	LVCMOS25
B<18>	IOB	INPUT	LVCMOS25
B<19>	IOB	INPUT	LVCMOS25
B<20>	IOB	INPUT	LVCMOS25
B<21>	IOB	INPUT	LVCMOS25
B<22>	IOB	INPUT	LVCMOS25
B<23>	IOB	INPUT	LVCMOS25
B<24>	IOB	INPUT	LVCMOS25
B<25>	IOB	INPUT	LVCMOS25
B<26>	IOB	INPUT	LVCMOS25
B<27>	IOB	INPUT	LVCMOS25
B<28>	IOB	INPUT	LVCMOS25
B<29>	IOB	INPUT	LVCMOS25
B<30>	IOB	INPUT	LVCMOS25
B<31>	IOB	INPUT	LVCMOS25
OPCODE<0>	IOB	INPUT	LVCMOS25
OPCODE<1>	IOB	INPUT	LVCMOS25

```
| OPCODE<2>                          | IOB              | INPUT     | LVCMOS25    |
| S<0>                               | IOB              | OUTPUT    | LVCMOS25    | 12
| S<1>                               | IOB              | OUTPUT    | LVCMOS25    | 12
| S<2>                               | IOB              | OUTPUT    | LVCMOS25    | 12
| S<3>                               | IOB              | OUTPUT    | LVCMOS25    | 12
| S<4>                               | IOB              | OUTPUT    | LVCMOS25    | 12
| S<5>                               | IOB              | OUTPUT    | LVCMOS25    | 12
| S<6>                               | IOB              | OUTPUT    | LVCMOS25    | 12
| S<7>                               | IOB              | OUTPUT    | LVCMOS25    | 12
| S<8>                               | IOB              | OUTPUT    | LVCMOS25    | 12
| S<9>                               | IOB              | OUTPUT    | LVCMOS25    | 12
| S<10>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<11>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<12>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<13>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<14>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<15>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<16>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<17>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<18>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<19>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<20>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<21>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<22>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<23>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<24>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<25>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<26>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<27>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<28>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<29>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<30>                              | IOB              | OUTPUT    | LVCMOS25    | 12
| S<31>                              | IOB              | OUTPUT    | LVCMOS25    | 12
+------------------------------------+------------------+-----------+-------------+----

Section 7 - RPMs
----------------

Section 8 - Guide Report
------------------------
Guide not run on this design.

Section 9 - Area Group and Partition Summary
--------------------------------------------

Partition Implementation Status
-------------------------------

  No Partitions were found in this design.

-------------------------------

Area Group Information
----------------------
```

```
  No area groups were found in this design.

----------------------

Section 10 - Modular Design Summary
-----------------------------------
Modular Design not used for this design.

Section 11 - Timing Report
--------------------------
This design was not run using timing mode.

Section 12 - Configuration String Details
-----------------------------------------
Use the "-detail" map option to print out Configuration Strings

Section 13 - Control Set Information
------------------------------------
No control set information for this architecture.

Section 14 - Utilization by Hierarchy
-------------------------------------

This feature is not supported for this architecture.
```

PART III (C): VERILOG EQUIVALENT OF 32-BIT ALU

We write the Verilog equivalent of 32-bit ALU in previous part as follows:

```
module alu32b(
input wire [2:0] OPCODE,
input wire [31:0] A,
input wire [31:0] B,
output reg [31:0] S
    );

always @(*) begin
    case (OPCODE)
        3'b000 : S = A + B;
        3'b001 : S = A - B;
        3'b010 : S = A << 1;
        3'b011 : S = A >> 1;
        3'b100 : S = A & B;
        3'b101 : S = A | B;
        default: S = 32'b0;
    endcase
end

endmodule
```

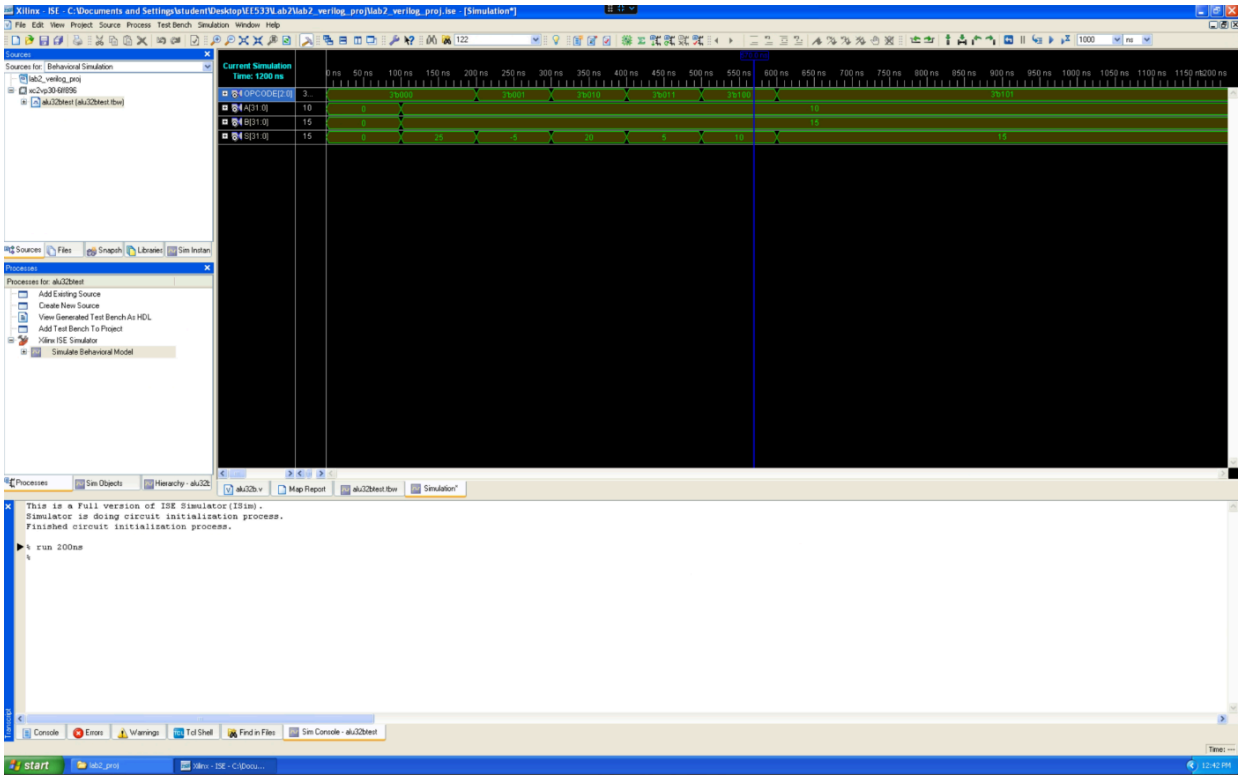

Figure 22. Function Simulation for Verilog Based ALU design

Here we also attach the mapper report:

```
Release 10.1 Map K.31 (nt)
Xilinx Mapping Report File for Design 'alu32b'

Design Information
------------------
Command Line   : map -ise "C:/Documents and
Settings/student/Desktop/EE533/Lab2/lab2_verilog_proj/lab2_verilog_proj.ise"
-intstyle ise -p xc2vp30-ff896-6 -cm area -pr off -k 4 -c 100 -tx off -o
alu32b_map.ncd alu32b.ngd alu32b.pcf
Target Device  : xc2vp30
Target Package : ff896
Target Speed   : -6
Mapper Version : virtex2p -- $Revision: 1.46 $
Mapped Date    : Fri Jan 23 12:36:59 2026

Design Summary
--------------
Number of errors:      0
Number of warnings:    0
Logic Utilization:
  Number of 4 input LUTs:                144 out of  27,392    1%
Logic Distribution:
  Number of occupied Slices:              72 out of  13,696    1%
    Number of Slices containing only related logic:      72 out of      72 100%
```

```
    Number of Slices containing unrelated logic:          0 out of      72   0%
      *See NOTES below for an explanation of the effects of unrelated logic.
  Total Number of 4 input LUTs:         144 out of  27,392    1%
  Number of bonded IOBs:                 99 out of     556   17%

Peak Memory Usage:  198 MB
Total REAL time to MAP completion:  1 secs
Total CPU time to MAP completion:   1 secs

NOTES:

   Related logic is defined as being logic that shares connectivity - e.g. two
   LUTs are "related" if they share common inputs.  When assembling slices,
   Map gives priority to combine logic that is related.  Doing so results in
   the best timing performance.

   Unrelated logic shares no connectivity.  Map will only begin packing
   unrelated logic into a slice once 99% of the slices are occupied through
   related logic packing.

   Note that once logic distribution reaches the 99% level through related
   logic packing, this does not mean the device is completely utilized.
   Unrelated logic packing will then begin, continuing until all usable LUTs
   and FFs are occupied.  Depending on your timing budget, increased levels of
   unrelated logic packing may adversely affect the overall timing performance
   of your design.

Table of Contents
-----------------
Section 1 - Errors
Section 2 - Warnings
Section 3 - Informational
Section 4 - Removed Logic Summary
Section 5 - Removed Logic
Section 6 - IOB Properties
Section 7 - RPMs
Section 8 - Guide Report
Section 9 - Area Group and Partition Summary
Section 10 - Modular Design Summary
Section 11 - Timing Report
Section 12 - Configuration String Information
Section 13 - Control Set Information
Section 14 - Utilization by Hierarchy

Section 1 - Errors
------------------

Section 2 - Warnings
--------------------

Section 3 - Informational
-------------------------
INFO:MapLib:562 - No environment variables are currently set.
INFO:LIT:244 - All of the single ended outputs in this design are using slew
   rate limited output drivers. The delay on speed critical single ended outputs
```

```
   can be dramatically reduced by designating them as fast outputs.

Section 4 - Removed Logic Summary
---------------------------------

Section 5 - Removed Logic
-------------------------

Section 6 - IOB Properties
--------------------------

+------------------------------------------------------------------------------------------
| IOB Name                                | Type             | Direction | IO Standard | Dr
|                                         |                  |           |             | St
+------------------------------------------------------------------------------------------
| A<0>                                    | IOB              | INPUT     | LVCMOS25    |
| A<1>                                    | IOB              | INPUT     | LVCMOS25    |
| A<2>                                    | IOB              | INPUT     | LVCMOS25    |
| A<3>                                    | IOB              | INPUT     | LVCMOS25    |
| A<4>                                    | IOB              | INPUT     | LVCMOS25    |
| A<5>                                    | IOB              | INPUT     | LVCMOS25    |
| A<6>                                    | IOB              | INPUT     | LVCMOS25    |
| A<7>                                    | IOB              | INPUT     | LVCMOS25    |
| A<8>                                    | IOB              | INPUT     | LVCMOS25    |
| A<9>                                    | IOB              | INPUT     | LVCMOS25    |
| A<10>                                   | IOB              | INPUT     | LVCMOS25    |
| A<11>                                   | IOB              | INPUT     | LVCMOS25    |
| A<12>                                   | IOB              | INPUT     | LVCMOS25    |
| A<13>                                   | IOB              | INPUT     | LVCMOS25    |
| A<14>                                   | IOB              | INPUT     | LVCMOS25    |
| A<15>                                   | IOB              | INPUT     | LVCMOS25    |
| A<16>                                   | IOB              | INPUT     | LVCMOS25    |
| A<17>                                   | IOB              | INPUT     | LVCMOS25    |
| A<18>                                   | IOB              | INPUT     | LVCMOS25    |
| A<19>                                   | IOB              | INPUT     | LVCMOS25    |
| A<20>                                   | IOB              | INPUT     | LVCMOS25    |
| A<21>                                   | IOB              | INPUT     | LVCMOS25    |
| A<22>                                   | IOB              | INPUT     | LVCMOS25    |
| A<23>                                   | IOB              | INPUT     | LVCMOS25    |
| A<24>                                   | IOB              | INPUT     | LVCMOS25    |
| A<25>                                   | IOB              | INPUT     | LVCMOS25    |
| A<26>                                   | IOB              | INPUT     | LVCMOS25    |
| A<27>                                   | IOB              | INPUT     | LVCMOS25    |
| A<28>                                   | IOB              | INPUT     | LVCMOS25    |
| A<29>                                   | IOB              | INPUT     | LVCMOS25    |
| A<30>                                   | IOB              | INPUT     | LVCMOS25    |
| A<31>                                   | IOB              | INPUT     | LVCMOS25    |
| B<0>                                    | IOB              | INPUT     | LVCMOS25    |
| B<1>                                    | IOB              | INPUT     | LVCMOS25    |
| B<2>                                    | IOB              | INPUT     | LVCMOS25    |
| B<3>                                    | IOB              | INPUT     | LVCMOS25    |
| B<4>                                    | IOB              | INPUT     | LVCMOS25    |
| B<5>                                    | IOB              | INPUT     | LVCMOS25    |
| B<6>                                    | IOB              | INPUT     | LVCMOS25    |
| B<7>                                    | IOB              | INPUT     | LVCMOS25    |
```

B<8>	IOB	INPUT	LVCMOS25	
B<9>	IOB	INPUT	LVCMOS25	
B<10>	IOB	INPUT	LVCMOS25	
B<11>	IOB	INPUT	LVCMOS25	
B<12>	IOB	INPUT	LVCMOS25	
B<13>	IOB	INPUT	LVCMOS25	
B<14>	IOB	INPUT	LVCMOS25	
B<15>	IOB	INPUT	LVCMOS25	
B<16>	IOB	INPUT	LVCMOS25	
B<17>	IOB	INPUT	LVCMOS25	
B<18>	IOB	INPUT	LVCMOS25	
B<19>	IOB	INPUT	LVCMOS25	
B<20>	IOB	INPUT	LVCMOS25	
B<21>	IOB	INPUT	LVCMOS25	
B<22>	IOB	INPUT	LVCMOS25	
B<23>	IOB	INPUT	LVCMOS25	
B<24>	IOB	INPUT	LVCMOS25	
B<25>	IOB	INPUT	LVCMOS25	
B<26>	IOB	INPUT	LVCMOS25	
B<27>	IOB	INPUT	LVCMOS25	
B<28>	IOB	INPUT	LVCMOS25	
B<29>	IOB	INPUT	LVCMOS25	
B<30>	IOB	INPUT	LVCMOS25	
B<31>	IOB	INPUT	LVCMOS25	
OPCODE<0>	IOB	INPUT	LVCMOS25	
OPCODE<1>	IOB	INPUT	LVCMOS25	
OPCODE<2>	IOB	INPUT	LVCMOS25	
S<0>	IOB	OUTPUT	LVCMOS25	12
S<1>	IOB	OUTPUT	LVCMOS25	12
S<2>	IOB	OUTPUT	LVCMOS25	12
S<3>	IOB	OUTPUT	LVCMOS25	12
S<4>	IOB	OUTPUT	LVCMOS25	12
S<5>	IOB	OUTPUT	LVCMOS25	12
S<6>	IOB	OUTPUT	LVCMOS25	12
S<7>	IOB	OUTPUT	LVCMOS25	12
S<8>	IOB	OUTPUT	LVCMOS25	12
S<9>	IOB	OUTPUT	LVCMOS25	12
S<10>	IOB	OUTPUT	LVCMOS25	12
S<11>	IOB	OUTPUT	LVCMOS25	12
S<12>	IOB	OUTPUT	LVCMOS25	12
S<13>	IOB	OUTPUT	LVCMOS25	12
S<14>	IOB	OUTPUT	LVCMOS25	12
S<15>	IOB	OUTPUT	LVCMOS25	12
S<16>	IOB	OUTPUT	LVCMOS25	12
S<17>	IOB	OUTPUT	LVCMOS25	12
S<18>	IOB	OUTPUT	LVCMOS25	12
S<19>	IOB	OUTPUT	LVCMOS25	12
S<20>	IOB	OUTPUT	LVCMOS25	12
S<21>	IOB	OUTPUT	LVCMOS25	12
S<22>	IOB	OUTPUT	LVCMOS25	12
S<23>	IOB	OUTPUT	LVCMOS25	12
S<24>	IOB	OUTPUT	LVCMOS25	12
S<25>	IOB	OUTPUT	LVCMOS25	12
S<26>	IOB	OUTPUT	LVCMOS25	12
S<27>	IOB	OUTPUT	LVCMOS25	12

```
| S<28>                                | IOB              | OUTPUT    | LVCMOS25    | 12
| S<29>                                | IOB              | OUTPUT    | LVCMOS25    | 12
| S<30>                                | IOB              | OUTPUT    | LVCMOS25    | 12
| S<31>                                | IOB              | OUTPUT    | LVCMOS25    | 12
+-------------------------------------------------------------------------------------

Section 7 - RPMs
----------------

Section 8 - Guide Report
------------------------
Guide not run on this design.

Section 9 - Area Group and Partition Summary
--------------------------------------------

Partition Implementation Status
-------------------------------

  No Partitions were found in this design.

-------------------------------

Area Group Information
----------------------

  No area groups were found in this design.

----------------------

Section 10 - Modular Design Summary
-----------------------------------
Modular Design not used for this design.

Section 11 - Timing Report
--------------------------
This design was not run using timing mode.

Section 12 - Configuration String Details
-----------------------------------------
Use the "-detail" map option to print out Configuration Strings

Section 13 - Control Set Information
------------------------------------
No control set information for this architecture.

Section 14 - Utilization by Hierarchy
-------------------------------------

This feature is not supported for this architecture.
```

We can compare the resource difference in following table:

	Metrics	Schematic-based design	Verilog-based design
1	**4-input LUTs (logic utilization)**	**131 / 27,392 (1%)**	**144 / 27,392 (1%)**
2	**Occupied slices**	**130 / 13,696 (1%)**	**72 / 13,696 (1%)**
3	**Total 4-input LUTs**	**226 / 27,392 (1%)**	144 / 27,392 (1%)
4	Bonded IOBs	99 / 556 (17%)	99 / 556 (17%)

Here we conclude that the schematic based design has less efficiency compared to verilog based design. This is mainly because that the schematic we draw is using logic gates, and we don't apply any optimization in regards to the FPGA resources. However, since the verilog is a simpler design, the tool is able to apply necessary optimization for given RTL code.

Appendix

GitHub Link to Design Files: https://github.com/jeremycai99/EE533-Labs/tree/728a28fb1e1e40987a7a307500239185c58bf52e/Lab2