Introduction to Digital Logic
EECS/CSE 31L

Final Project: Simple Processor

Prepared by: Jack Melcher
Student ID: 67574625

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

March 10, 2015

1 Block Description

The processor fetches a 32-bit set of instructions, decodes the instructions in a controller, performs the specified task, and stores the results into a register file.

The processor is made up of several components.

- Counter (assignment 4)
- Memory
- Controller
- Register File (assignment 4)
- Selector for A input
- Selector for B input
- ALU 32 (assignment 3)

2 Input/Output Port Description

Package(c31L_pack): Contains all the necessary constants used throughout all the blocks

Constant Name	Data Type	Default Value	Description
BW	INTEGER	32	Size of logic that is being used
OP_BINARY	STD_LOGIC	010010 (18)	Number of Operations in the Memory
Reg_field	INTEGER	6	Determines size of register file and used to as its addr size
NBIT	INTEGER	6	Max bits the counter uses
Alu_function_type	std_logic_vector	3 downto 0	Used to help tie an alu operation to a given number
Mux_instructions	Array of std_logic_vector	((OP-1) downto 0) (BW-1 downto 0)	Stores all the hard coded instructions for the processor

Processor (The outputs are used to help display values of processor instruction results)

Port name	Port size	Port Type	Description
clk	1	IN	Trigger the processor fetch and write events
Pc_enable	1	IN	Enable the counter to count
Rst_s	1	IN	Synchronous reset for the Counter and RegFile
memory_addrs	NBIT-1 downto 0	OUT	Determine which instruction is read
Instruction	BW-1 downto 0	OUT	Shows the 32 bit instruction
reg_source	BW-1 downto 0	OUT	The first value read from register file
reg_target	BW-1 downto 0	OUT	The second value read from register file
immediate	BW-1 downto 0	OUT	The immediate value from instruction
carry	1	OUT	The carry out value from ALU
output	BW-1 downto 0	OUT	Results of ALU operation

Counter

Port name	Port size	Port Type	Description
Clk	1	IN	Triggers the counting
Rst_s	1	IN	Will reset the counter to zero when active
dout	NBIT-1 downto 0	OUT	The counter's value after counting

Variable Name	Data Type	Port size	Description
count	INTEGER	NBIT-1 downto 0	Aids in the counting process

Memory

Port name	Port size	Port Type	Description
addr	NBIT-1 downto 0	IN	Determines which instructions are fetched
Read_data	BW-1 downto 0	OUT	The instructions that are sent to controller

Controller

Port name	Port size	Port Type	Description
instruction	31 downto 0	IN	Instructions fetched from memory
rt_and_imm	31 downto 0	OUT	Immediate value
write_enable	1	OUT	Determines whether the register file can write
rs_index	2 downto 0	OUT	Address of register source
rt_index	1	OUT	Address of register target
rd_index	1	OUT	Address of register destination
b_mux_sel	31 downto 0	OUT	Instruction type to determine B input of ALU
alu_func	alu_function_type	OUT	Operation the ALU is supposed to perform

Register File

Port name	Port size	Port Type	Description
Clk	1	IN	Triggers the register process
Rst_s	1	IN	Will reset the registers to zero
write_enable	1	IN	Enables writing to the registers
rs_index	Reg_field -1 downto 0	IN	The address of the register to be read from
rt_index	Reg_field -1 downto 0	IN	The address of the register to be read from
rd_index	Reg_field -1 downto 0	IN	The address of the register to be written to
reg_source_out	BW-1 downto 0	OUT	The data value read from register file
reg_target_out	BW-1 downto 0	OUT	The data value read from register file
reg_dest_new	BW-1 downto 0	IN	The data value to be written to the register

Variable Name	Data Type	Port size	Description
regfile	Array (0 to 2** Reg_field)	NBIT-1 downto 0	The registers

Selector for A input

Port name	Port size	Port Type	Description
reg_source	31 downto 0	IN	The data value read from register file
immediate	31 downto 0	IN	Immediate provided by the controller
alu_op	1	INOUT	The expected ALU operation to be performed
instruction_type			Selects A input based on RI value
opsel	2 downto 0	IN	The ALU operation select value
mode	1	OUT	The ALU mode select value
output	31 downto 0	OUT	What value is sent to A input of ALU

Selector for B input

Port name	Port size	Port Type	Description
in0	BW-1 downto 0	IN	Register target value
in1	BW-1 downto 0	IN	Immediate value
sel	1	INOUT	Selects B input based on RI value
output	BW-1 downto 0		What value is sent to B input of ALU

ALU 32-bit

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	SIGNAL	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

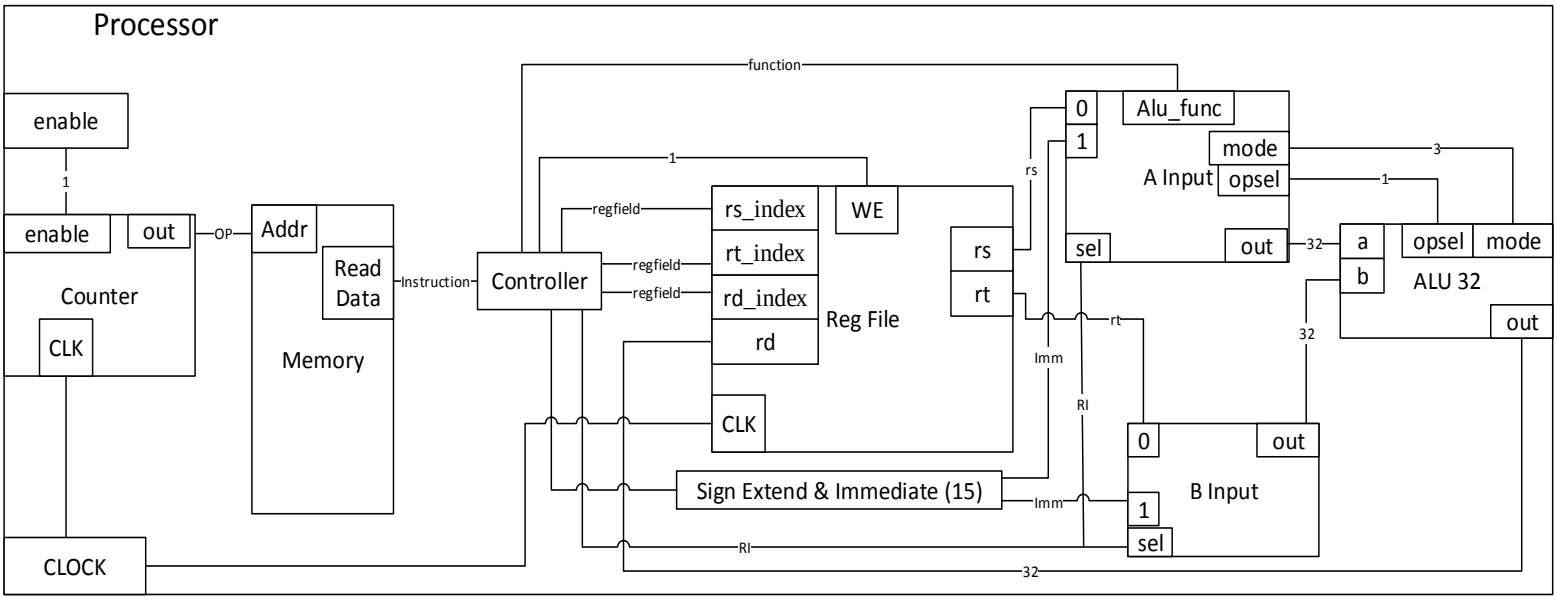
Instruction Format

1 bit	6 bit	6 bit	4 bit	6 bit	9bit
Instruction Type (RI)	**Register Source (rs)**	**Register Destination (rd)**	**Function**	**Register Target (rt)**	**Immediate (Imm)**
Whether ALU works with Immediate	Address of rs	The address for which the results of ALU will be written to	Code for ALU operation	Address of rt and the first 6 bits of Immediate	The last 9 bits of the Immediate

ALU Operations:

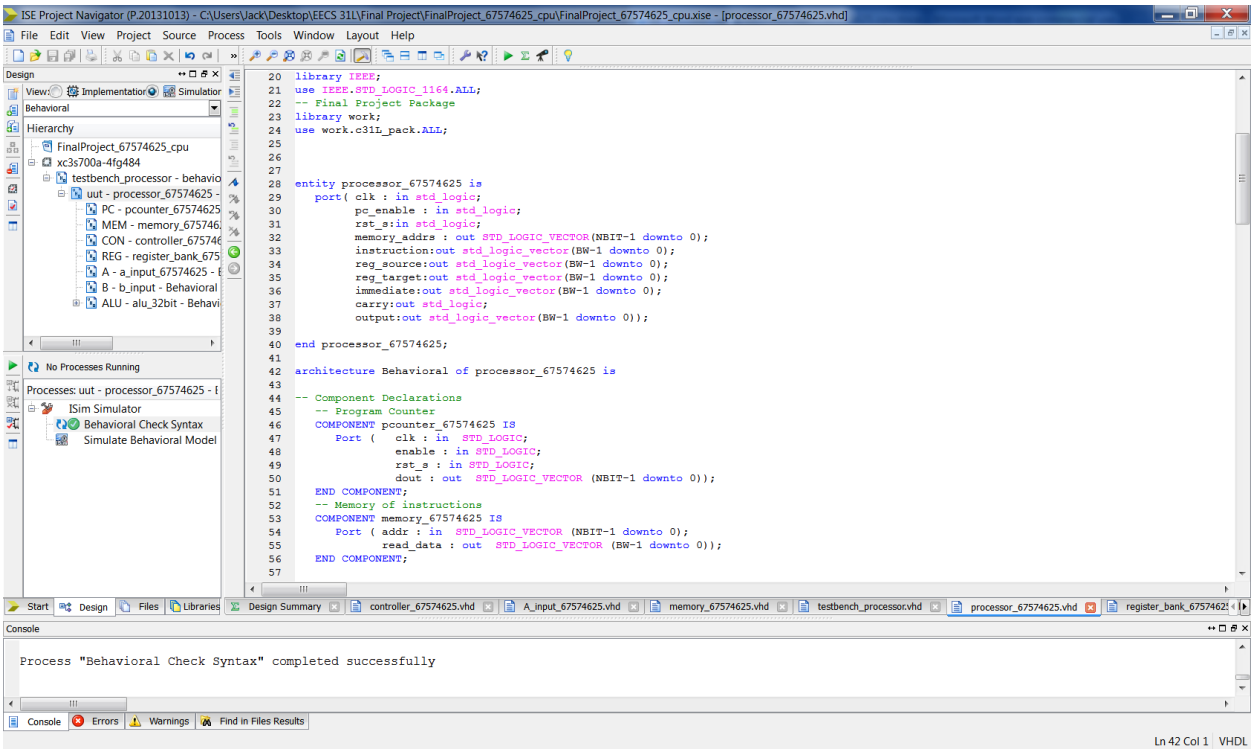
Function	Register Operation (RI = '0')	Immediate Operation (RI = '1')	Description
0000	NOP	NOP	No Operation
0001	rs + rt	a+ Imm	Add
0010	rs + rt '	a+ Imm '	Sub
1011	rs	Imm	Move
1100	rs +1	Imm +1	increment
1101	rs -1	Imm -1	decrement
1110	rs + rt +1	a+ Imm +1	Add & Increment
0101	rs AND rt	a AND Imm	Bit-wise AND
0110	rs OR rt	a OR Imm	Bit-wise OR
1000	rs XOR rt	a XOR Imm	Bit-wise Exclusive OR
0111	rs '	Imm'	Compliment
1001	shl rs	shl Imm	1 Bit Shift Left

Design:

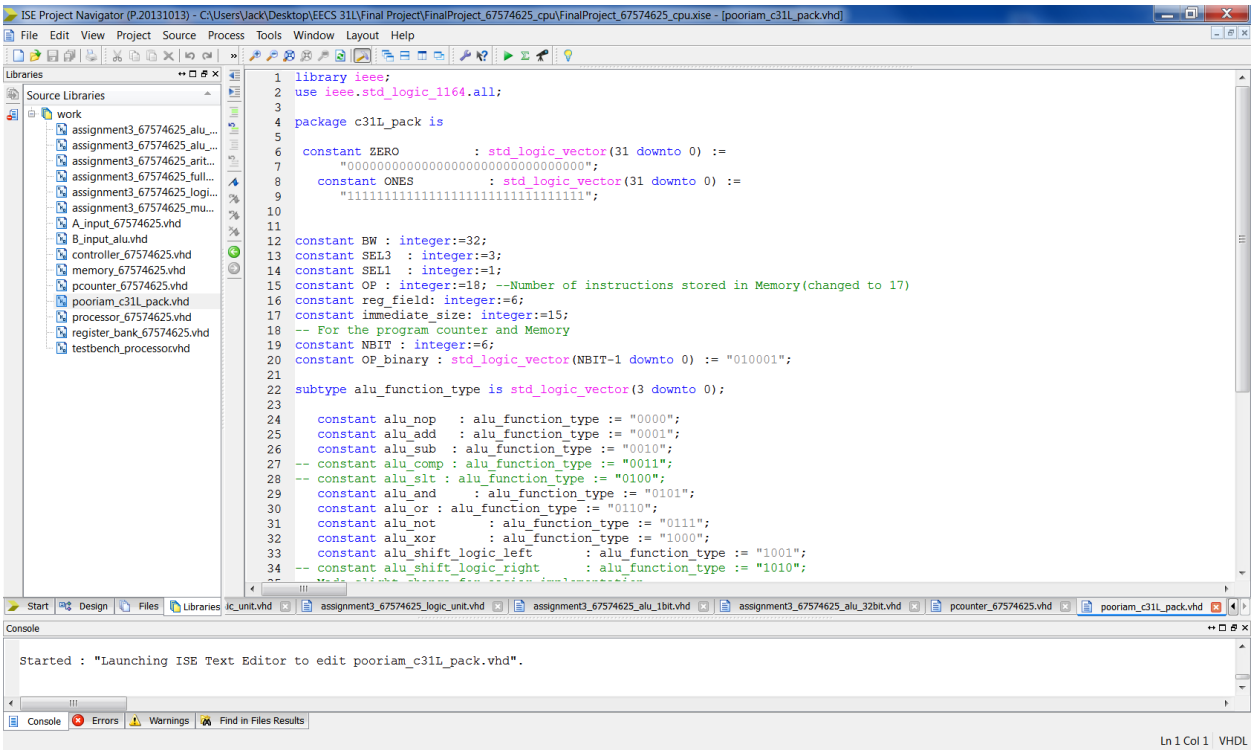


4 Compilation

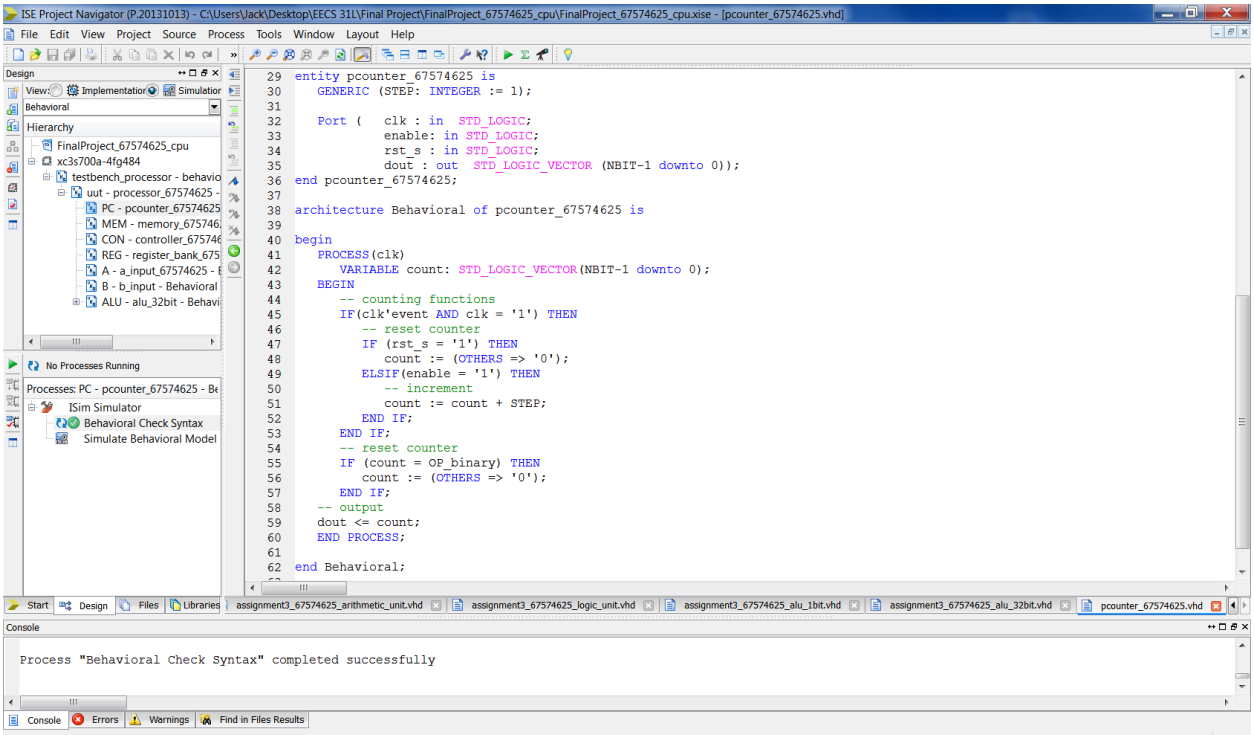
Processor



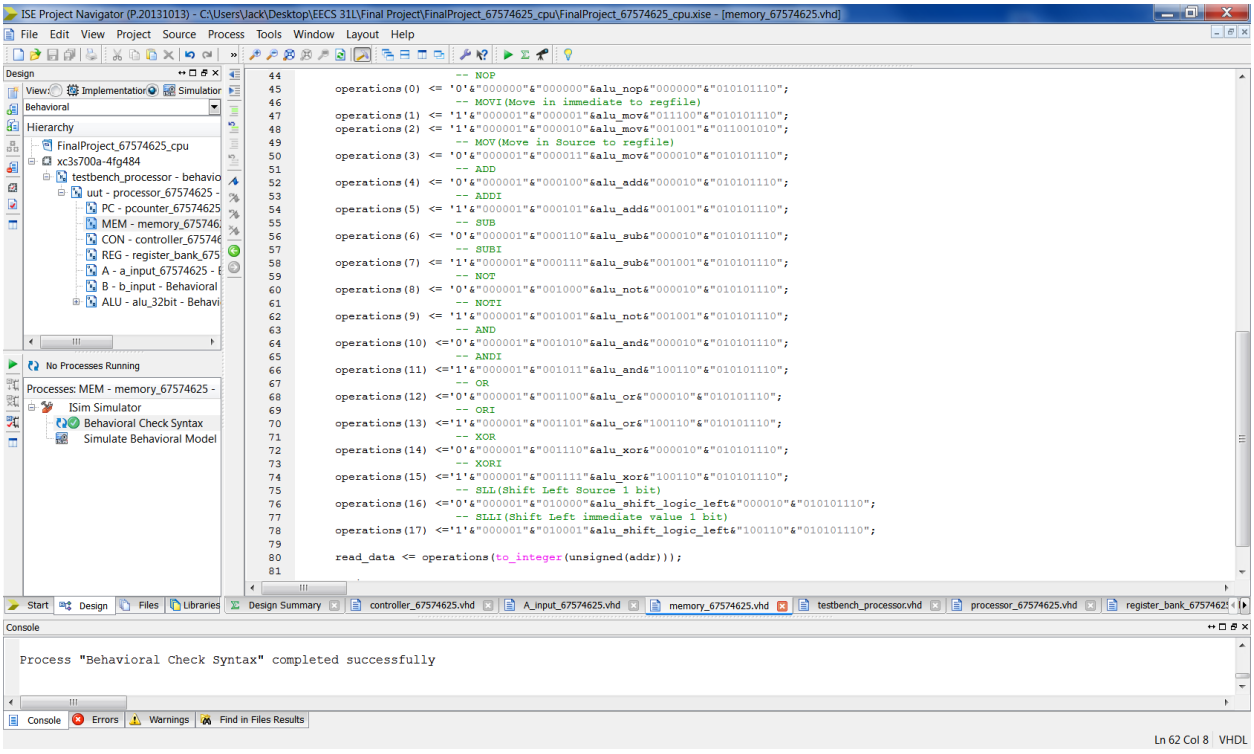
C31L_pack



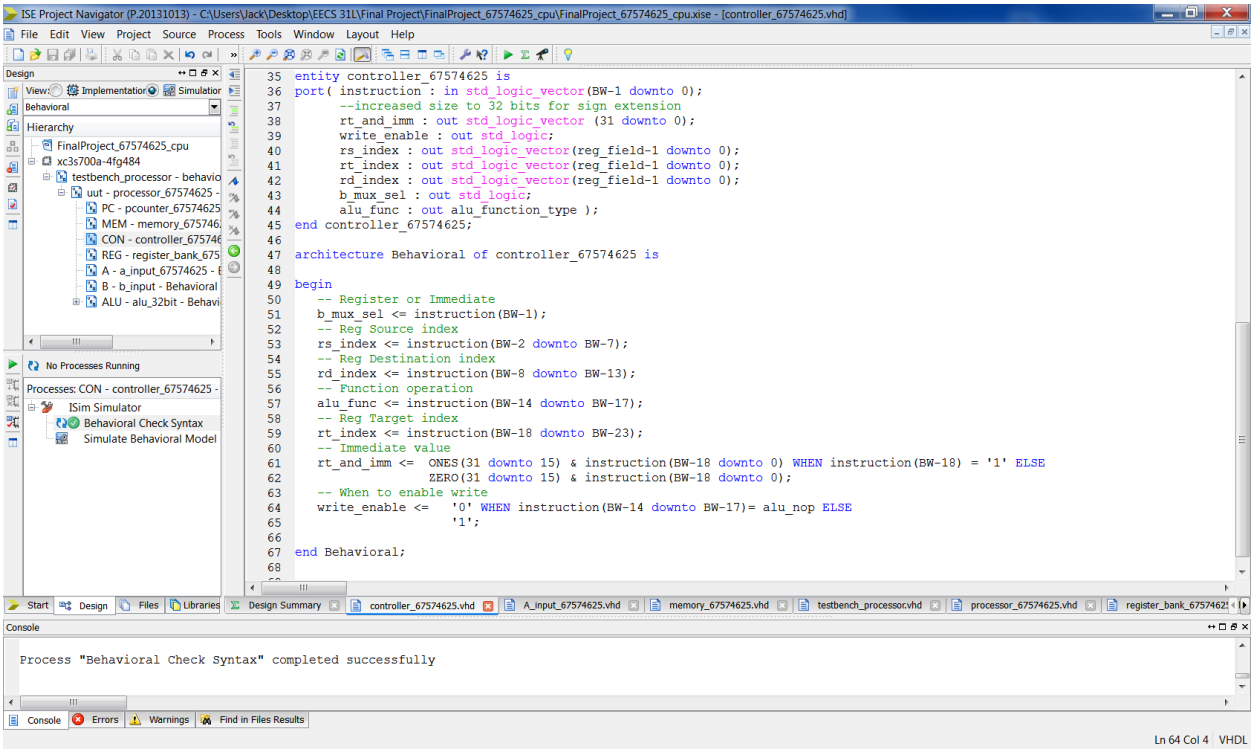
Counter



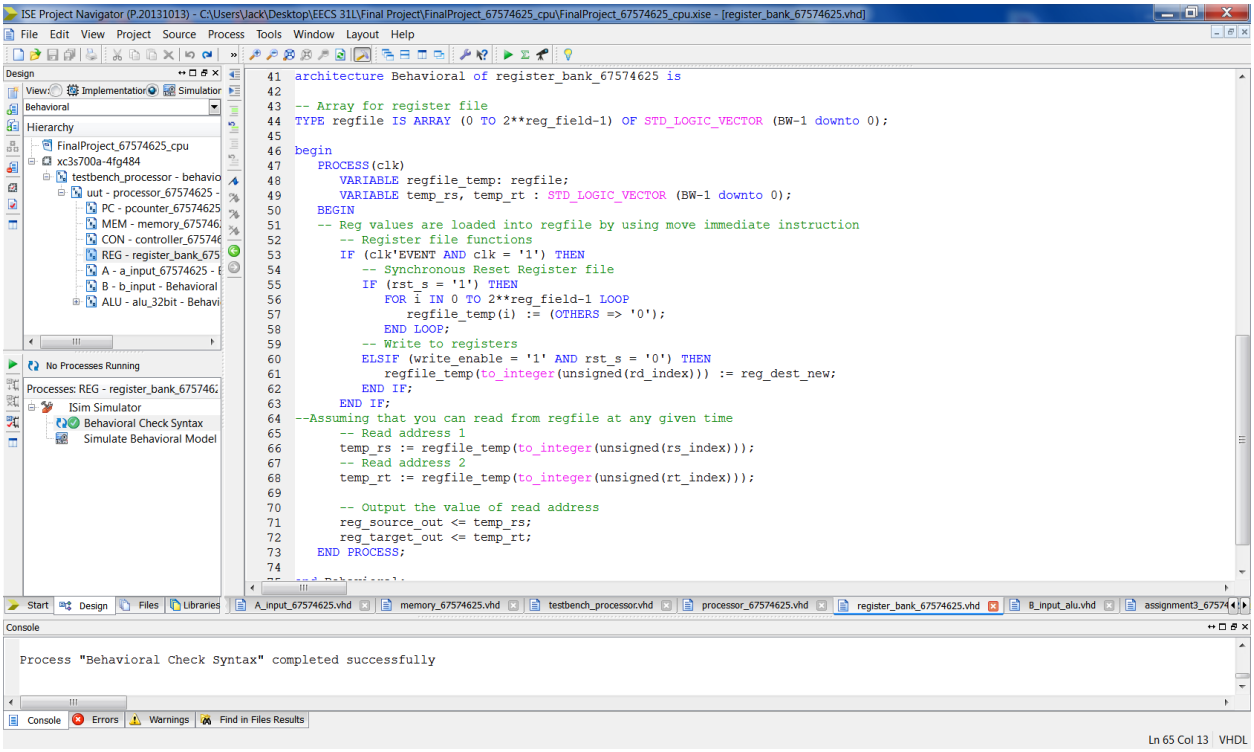
Memory



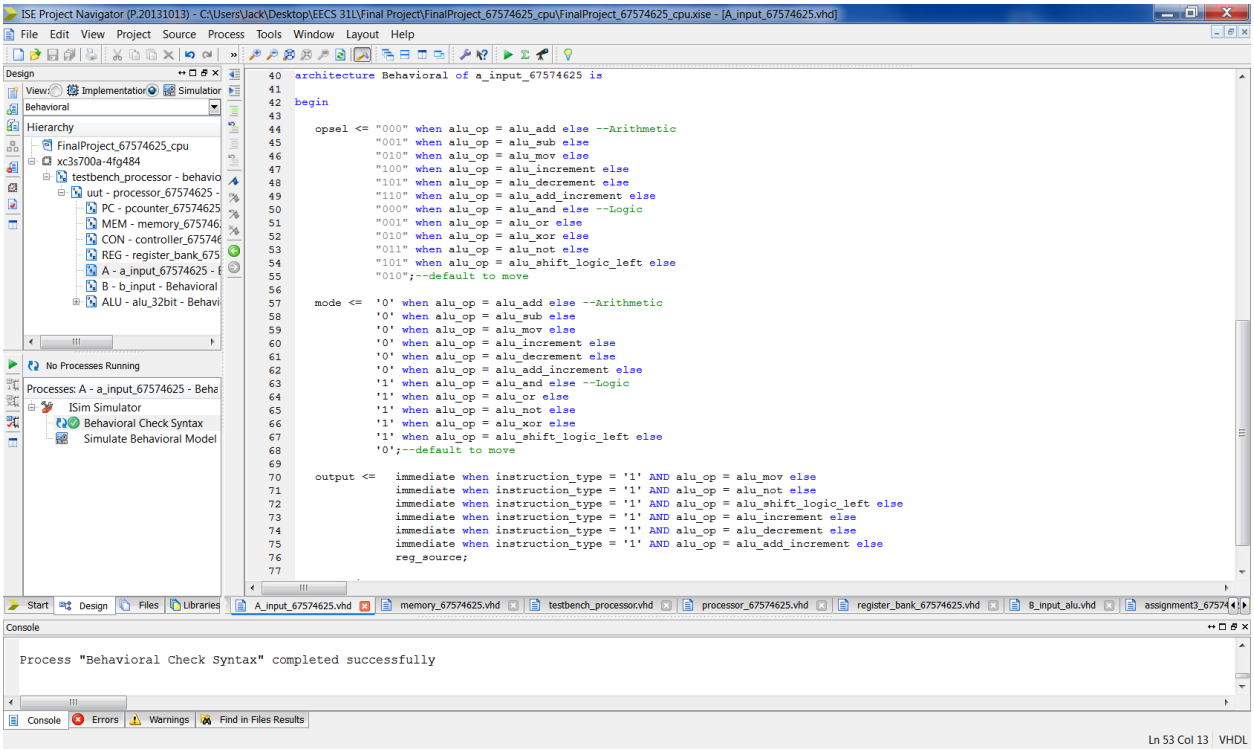
Controller



Register File

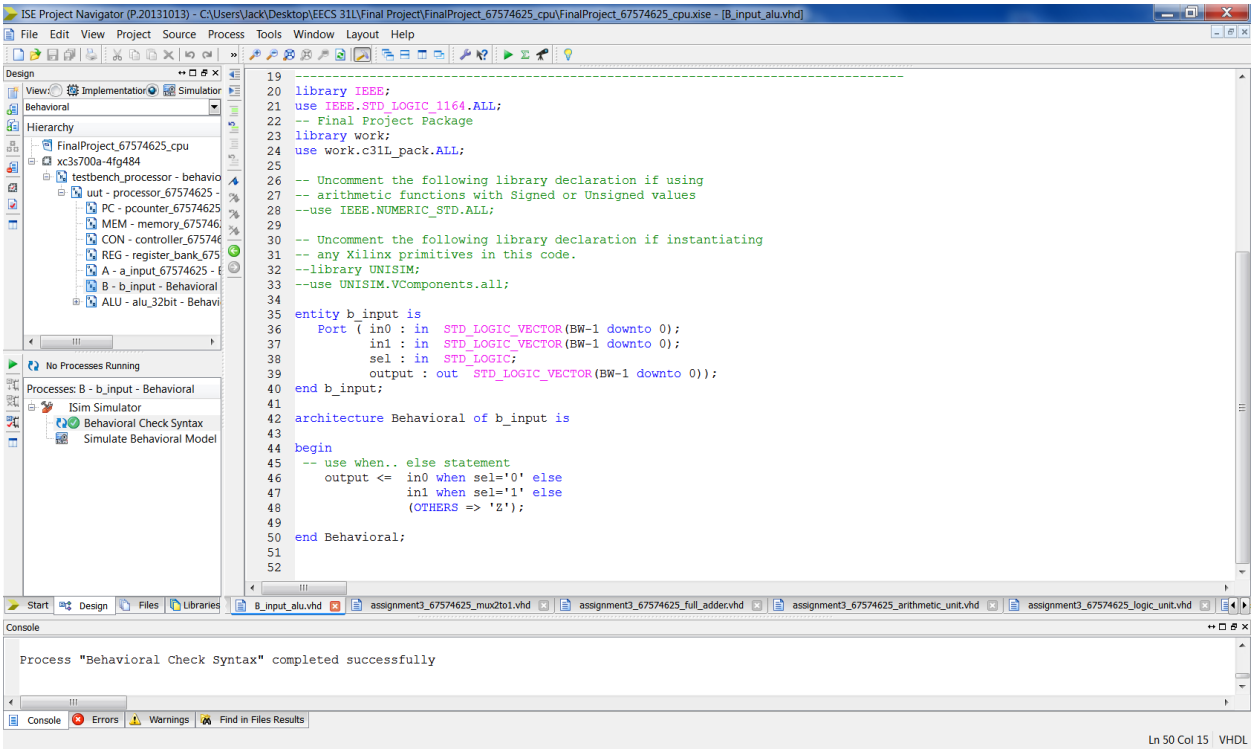


Selector for A input



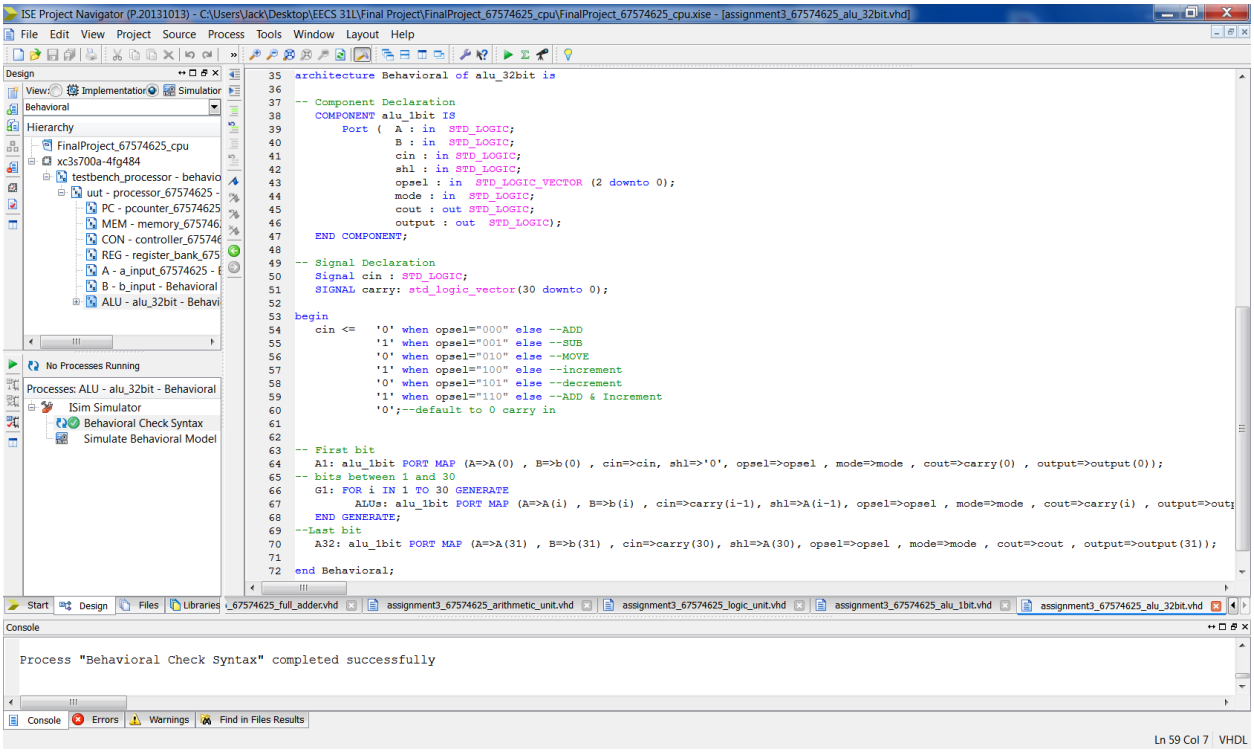

```
40  architecture Behavioral of a_input_67574625 is
41
42  begin
43
44     opsel <= "000" when alu_op = alu_add else --Arithmetic
45              "001" when alu_op = alu_sub else
46              "010" when alu_op = alu_mov else
47              "100" when alu_op = alu_increment else
48              "101" when alu_op = alu_decrement else
49              "110" when alu_op = alu_add_increment else
50              "000" when alu_op = alu_and else --Logic
51              "001" when alu_op = alu_or else
52              "010" when alu_op = alu_xor else
53              "011" when alu_op = alu_not else
54              "101" when alu_op = alu_shift_logic_left else
55              "010";--default to move
56
57     mode <=  '0' when alu_op = alu_add else --Arithmetic
58              '0' when alu_op = alu_sub else
59              '0' when alu_op = alu_mov else
60              '0' when alu_op = alu_increment else
61              '0' when alu_op = alu_decrement else
62              '0' when alu_op = alu_add_increment else
63              '1' when alu_op = alu_and else --Logic
64              '1' when alu_op = alu_or else
65              '1' when alu_op = alu_not else
66              '1' when alu_op = alu_xor else
67              '1' when alu_op = alu_shift_logic_left else
68              '0';--default to move
69
70     output <=   immediate when instruction_type = '1' AND alu_op = alu_mov else
71                 immediate when instruction_type = '1' AND alu_op = alu_not else
72                 immediate when instruction_type = '1' AND alu_op = alu_shift_logic_left else
73                 immediate when instruction_type = '1' AND alu_op = alu_increment else
74                 immediate when instruction_type = '1' AND alu_op = alu_decrement else
75                 immediate when instruction_type = '1' AND alu_op = alu_add_increment else
76                 reg_source;
77
```

Selector for B input

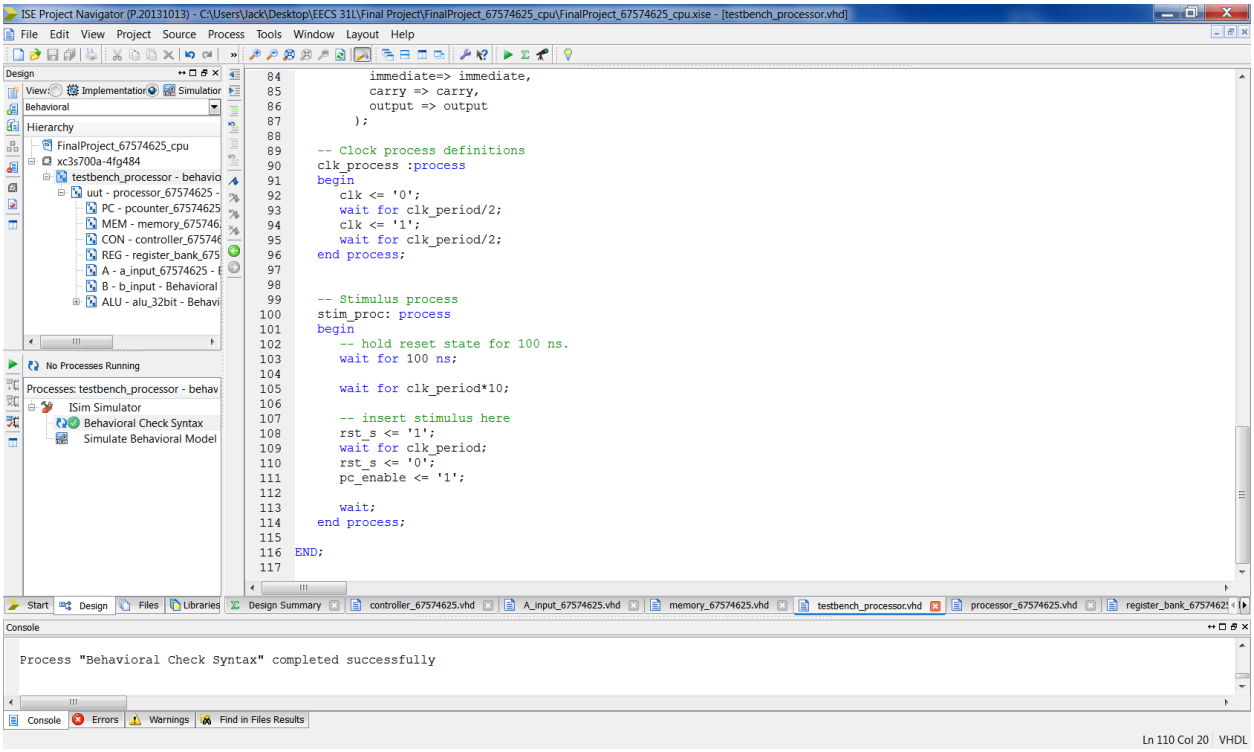



```
19  ----------------------------------------------------------------------------------
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22  -- Final Project Package
23  library work;
24  use work.c31L_pack.ALL;
25
26  -- Uncomment the following library declaration if using
27  -- arithmetic functions with Signed or Unsigned values
28  --use IEEE.NUMERIC_STD.ALL;
29
30  -- Uncomment the following library declaration if instantiating
31  -- any Xilinx primitives in this code.
32  --library UNISIM;
33  --use UNISIM.VComponents.all;
34
35  entity b_input is
36     Port ( in0 : in  STD_LOGIC_VECTOR(BW-1 downto 0);
37            in1 : in  STD_LOGIC_VECTOR(BW-1 downto 0);
38            sel : in  STD_LOGIC;
39            output : out  STD_LOGIC_VECTOR(BW-1 downto 0));
40  end b_input;
41
42  architecture Behavioral of b_input is
43
44  begin
45   -- use when.. else statement
46      output <=  in0 when sel='0' else
47                 in1 when sel='1' else
48                 (OTHERS => 'Z');
49
50  end Behavioral;
51
52
```

ALU 32



Testbench



5 Elaboration

Assumptions:

Functions

- Shifting left the register source by an immediate number of times was altered to just be shift left immediate value one bit
- Because the ALU could do increment, decrement, and add & increment, these functions were given their own code and can be used

Counter

- Need only to increment and then loop back to the first instruction

Memory

- The instructions had to be hard coded into this component
- Register 1 and register 2 in the Register File are assigned values using the MOVI funciton and then those values are used for the remainder of the

ALU

- This didn't need any adjustments aside from the change of the cin port to a signal

A input

- Because some functions asked for the Immediate to be moved or shifted, another component was designed to handle this.
- In order to translate the function code to the ALU opsel and mode, this component also had to handle it.

Errors and Challenges:

- Lot of errors encountered when trying to make sure all the components are wired together in the correct sequence.
- A lot of assumptions had to be throughout the course of the final project
- It took a lot of time to design the schematic for the processor because of these assumptions

Simulation Log:

Started : "Simulate Behavioral Model".

Determining files marked for global include in the design...
Running fuse...
Command Line: fuse -intstyle ise -incremental -o {C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/testbench_processor_isim_beh.exe} -prj {C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/testbench_processor_beh.prj} work.testbench_processor {}
Running: C:\Xilinx\14.7\ISE_DS\ISE\bin\nt64\unwrapped\fuse.exe -intstyle ise -incremental -o C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/testbench_processor_isim_beh.exe -prj

C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/testbench_processor_beh.prj work.testbench_processor
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/Final Project/FinalProject_67574625_cpu/assignment3_67574625_mux2to1.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/assignment3_67574625_logic_unit.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/assignment3_67574625_full_adder.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/assignment3_67574625_arithmetic_unit.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/pooriam_c31L_pack.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/assignment3_67574625_alu_1bit.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/register_bank_67574625.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/pcounter_67574625.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/memory_67574625.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/controller_67574625.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/B_input_alu.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/A_input_67574625.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/assignment3_67574625_alu_32bit.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/processor_67574625.vhd" into library work
Parsing VHDL file "C:/Users/Jack/Desktop/EECS 31L/Final Project/FinalProject_67574625_cpu/testbench_processor.vhd" into library work
Starting static elaboration
Completed static elaboration
Compiling package standard

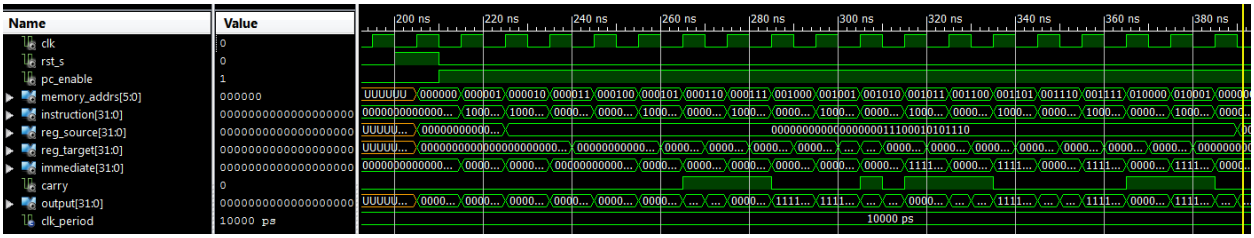
Compiling package std_logic_1164
Compiling package c31l_pack
Compiling package std_logic_arith
Compiling package std_logic_unsigned
Compiling package numeric_std
Compiling architecture behavioral of entity pcounter_67574625 [\pcounter_67574625(1)\]
Compiling architecture behavioral of entity memory_67574625 [memory_67574625_default]
Compiling architecture behavioral of entity controller_67574625 [controller_67574625_default]
Compiling architecture behavioral of entity register_bank_67574625
[register_bank_67574625_default]
Compiling architecture behavioral of entity a_input_67574625 [a_input_67574625_default]
Compiling architecture behavioral of entity b_input [b_input_default]
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 behavioral of entity processor_67574625 [processor_67574625_default]
Compiling architecture behavior of entity testbench_processor
Time Resolution for simulation is 1ps.
Waiting for 11 sub-compilation(s) to finish...
Compiled 33 VHDL Units
Built simulation executable C:/Users/Jack/Desktop/EECS 31L/Final
Project/FinalProject_67574625_cpu/testbench_processor_isim_beh.exe
Fuse Memory Usage: 37900 KB
Fuse CPU Usage: 1200 ms
Launching ISim simulation engine GUI...
"C:/Users/Jack/Desktop/EECS 31L/Final
Project/FinalProject_67574625_cpu/testbench_processor_isim_beh.exe" -intstyle ise -gui -
tclbatch isim.cmd -wdb "C:/Users/Jack/Desktop/EECS 31L/Final
Project/FinalProject_67574625_cpu/testbench_processor_isim_beh.wdb"
ISim simulation engine GUI launched successfully

Process "Simulate Behavioral Model" completed successfully

6 Waveforms

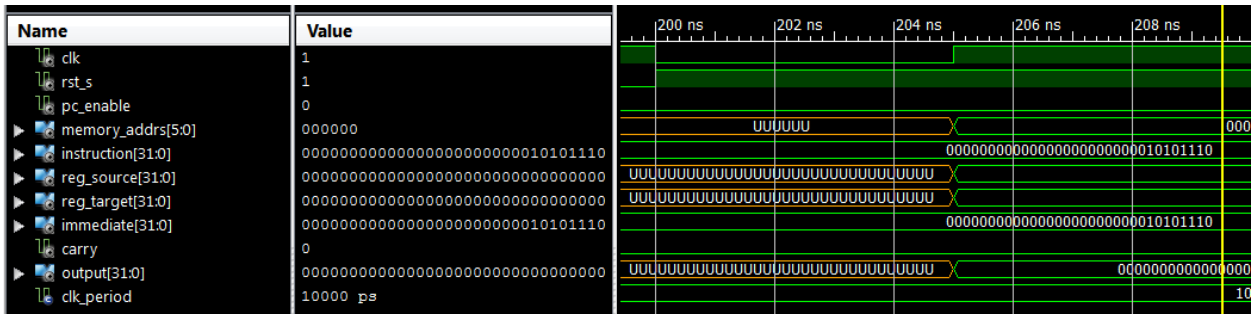
Note: At the rising edge of clock, the output shown is saved and the memory_addrs changes

Full Waveform Snapshot 1

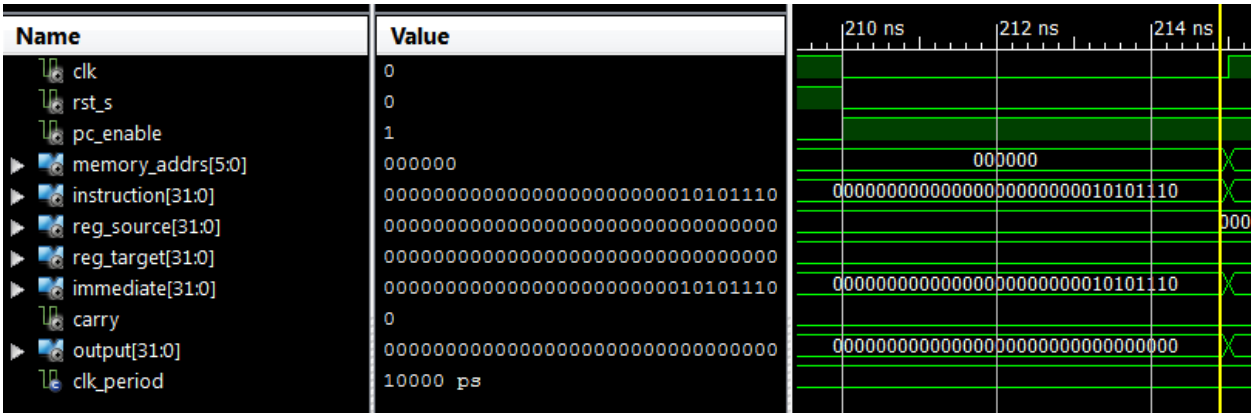


Snapshots of Waveform

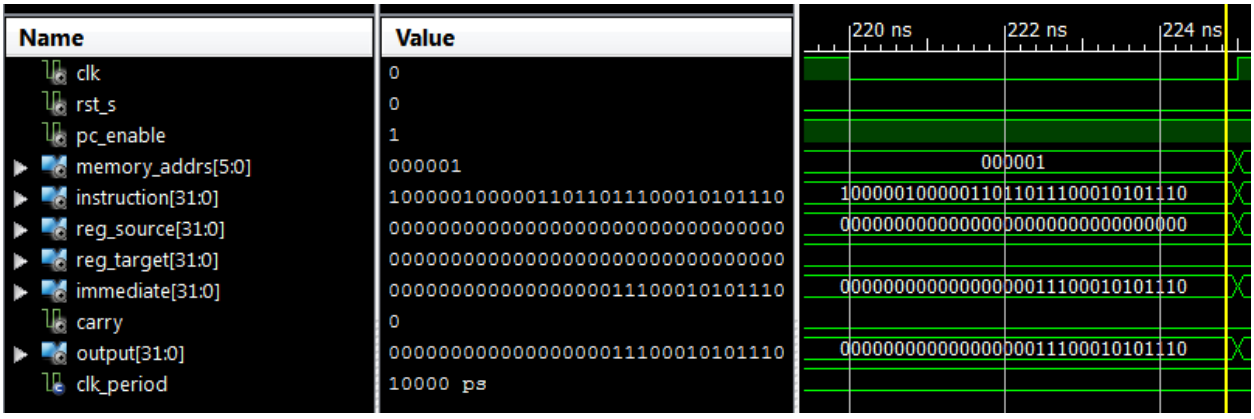
RESET (Not a function)



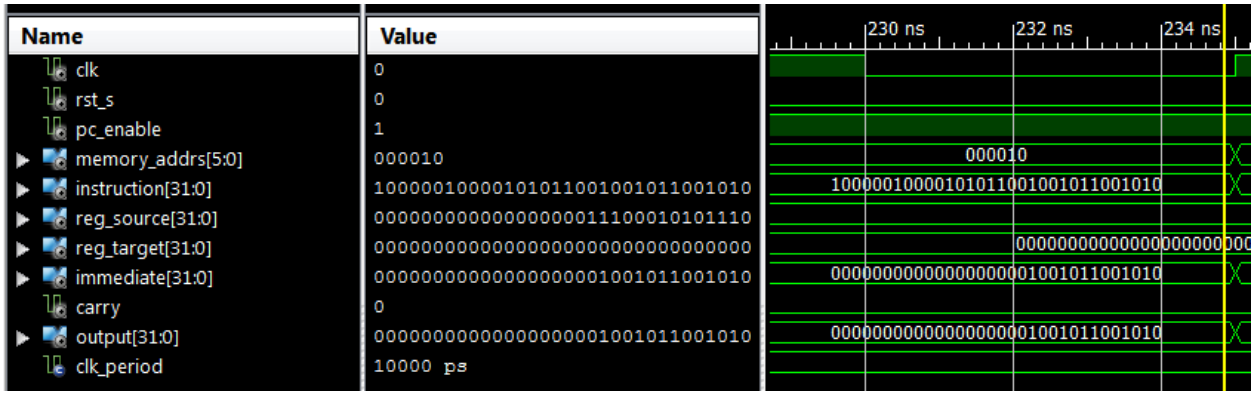
NOP



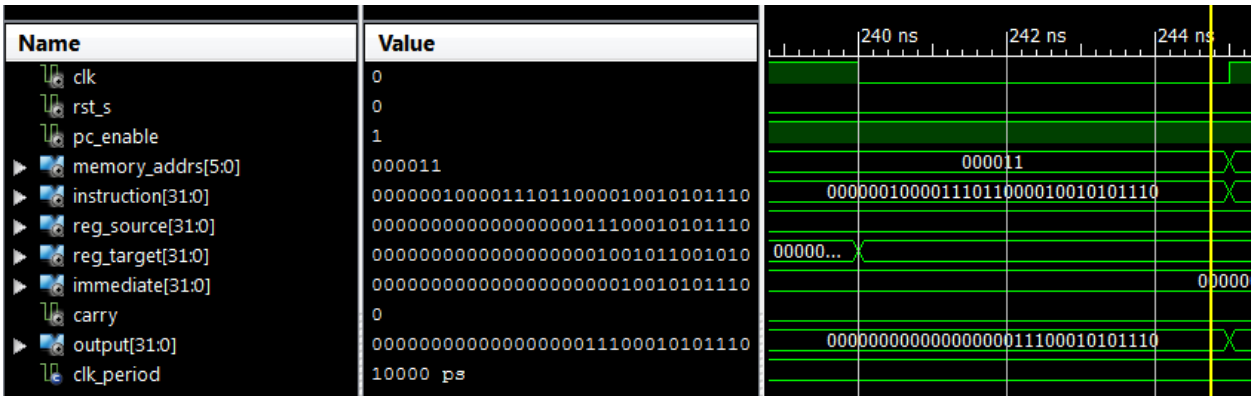
MOVI (to give value to a register)



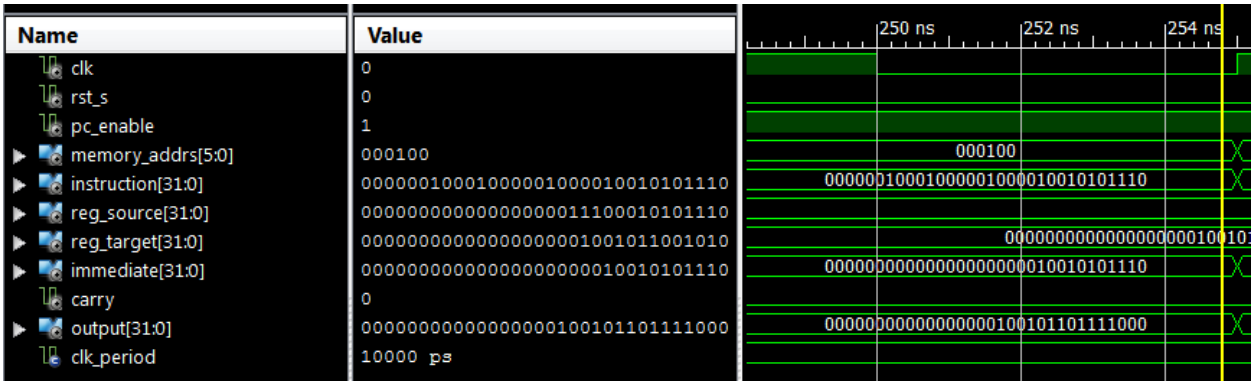
MOVI (to give value to a second register)



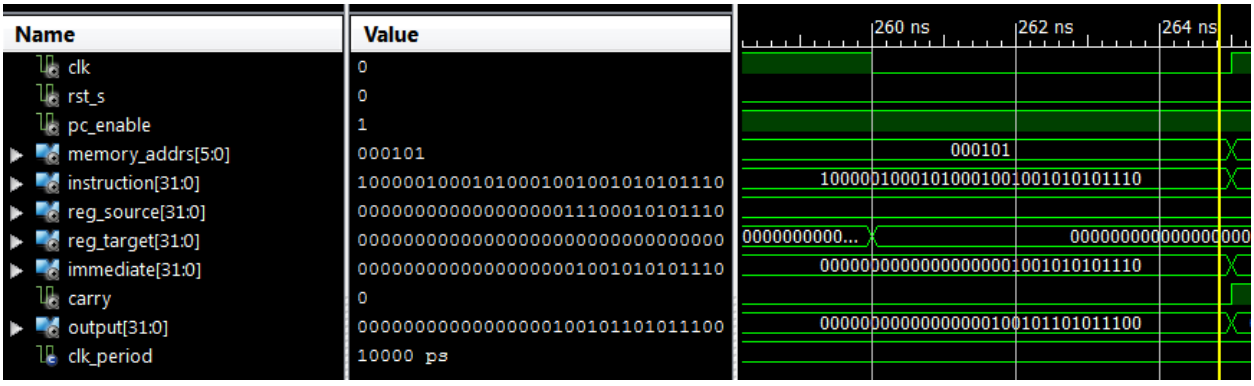
MOV



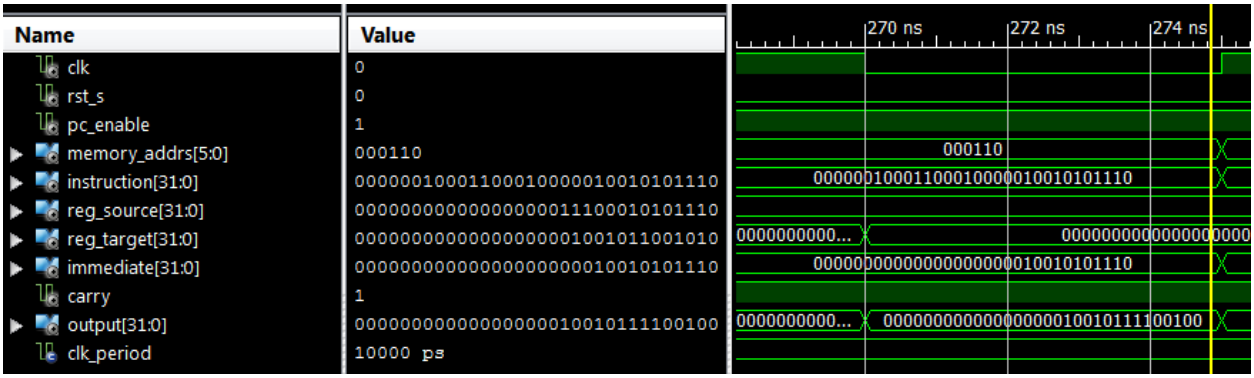
ADD



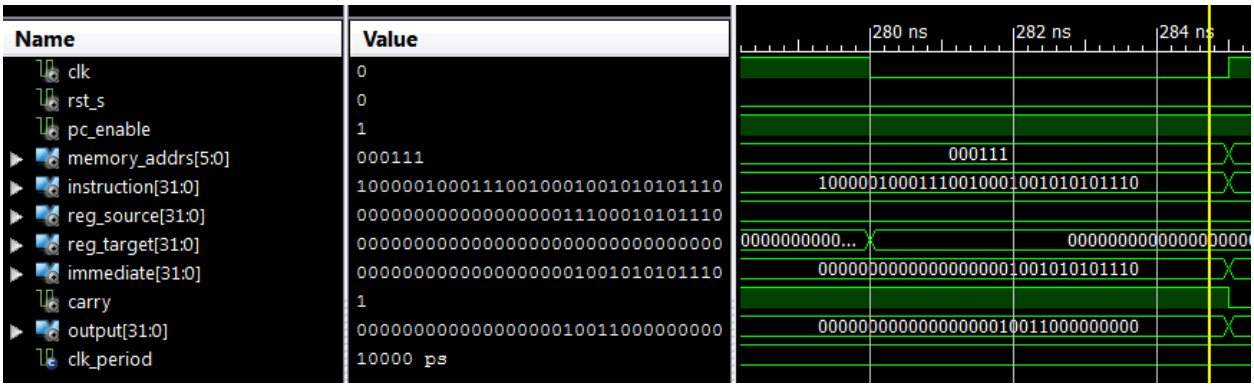
ADDI



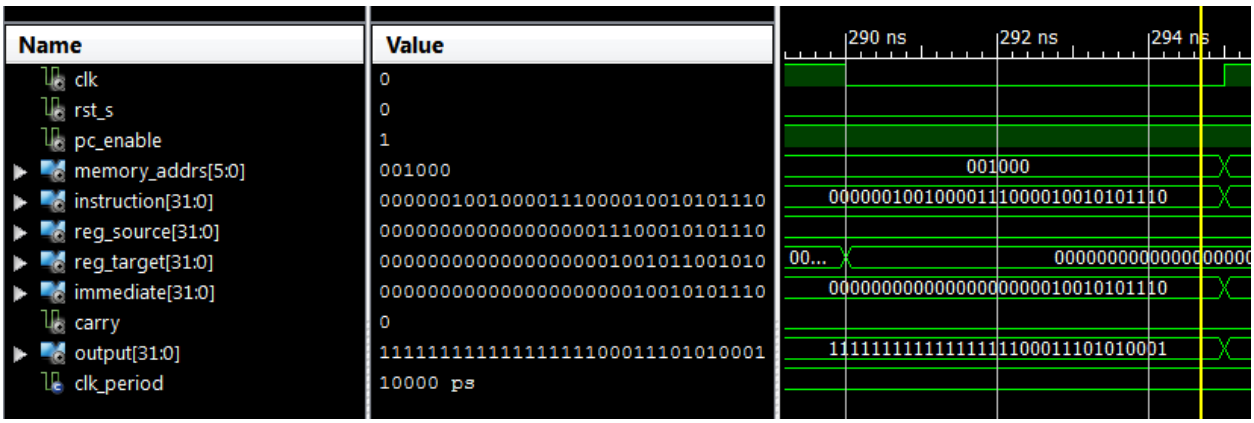
SUB



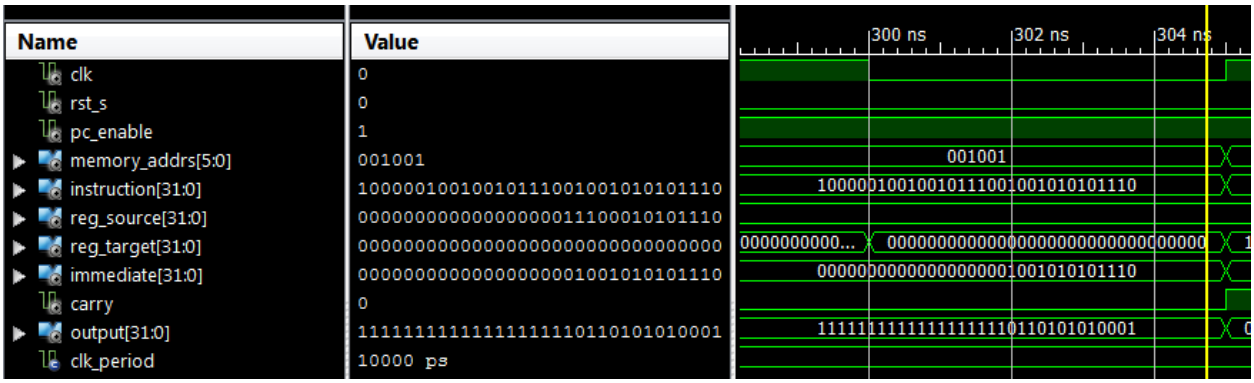
SUBI



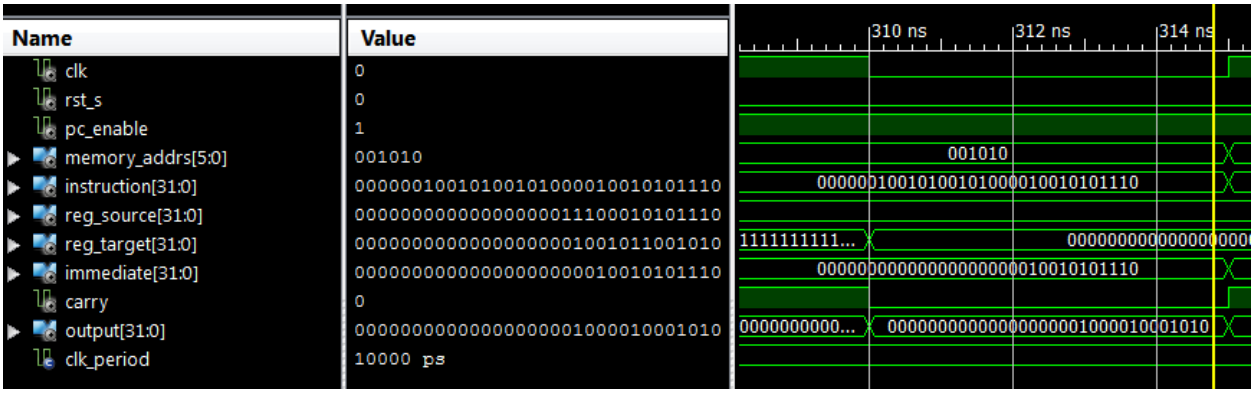
NOT



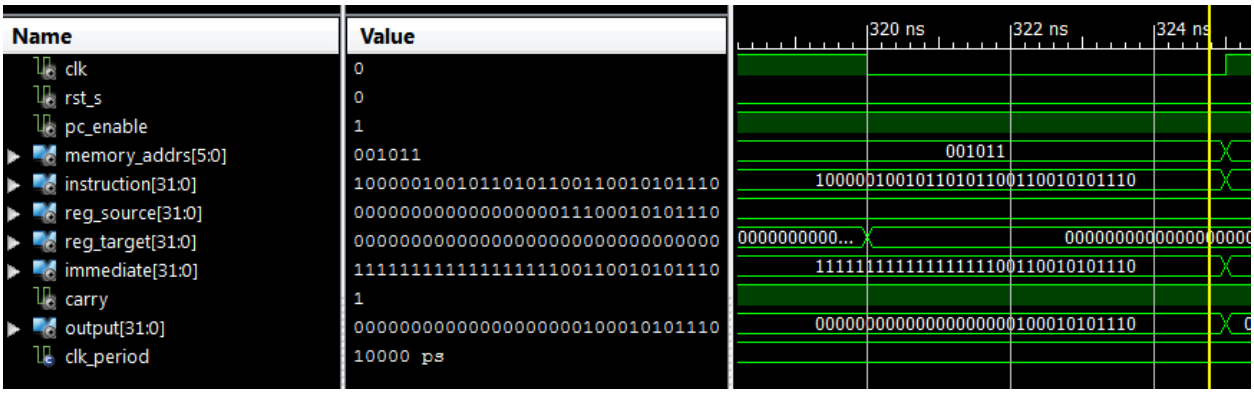
NOTI



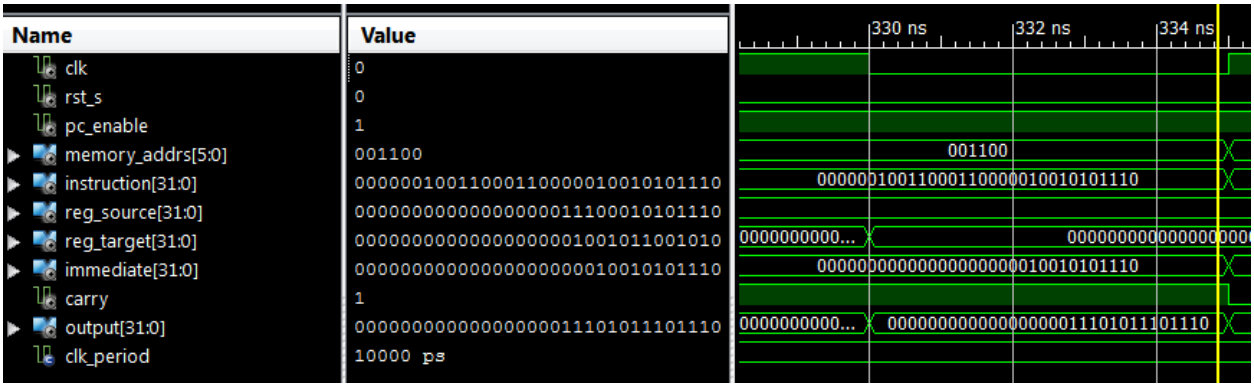
AND



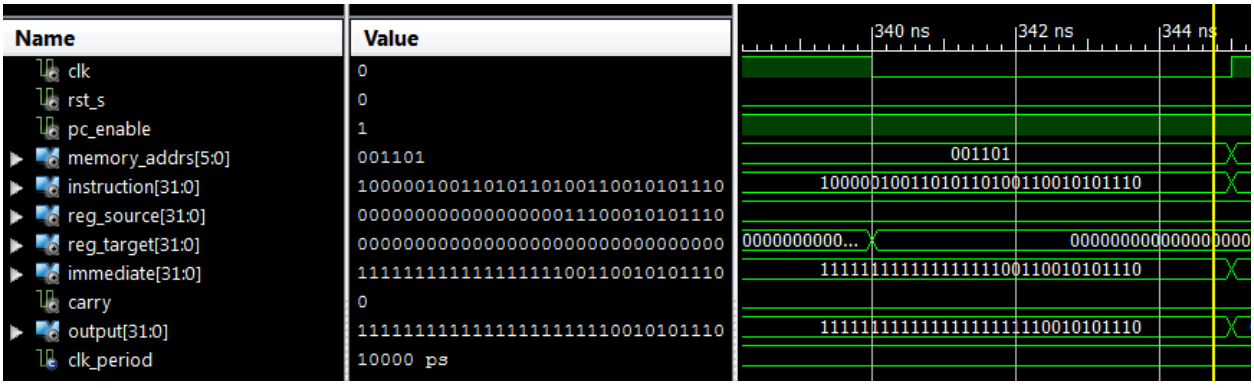
ANDI



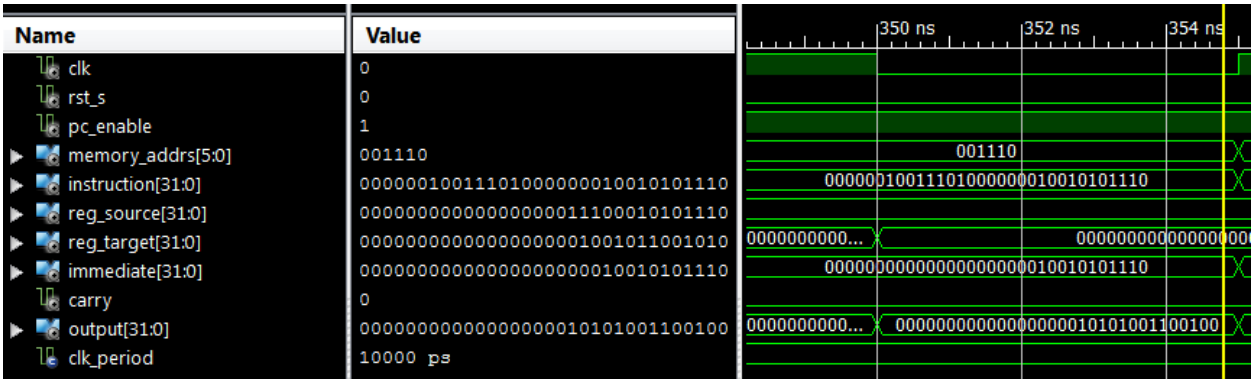
OR



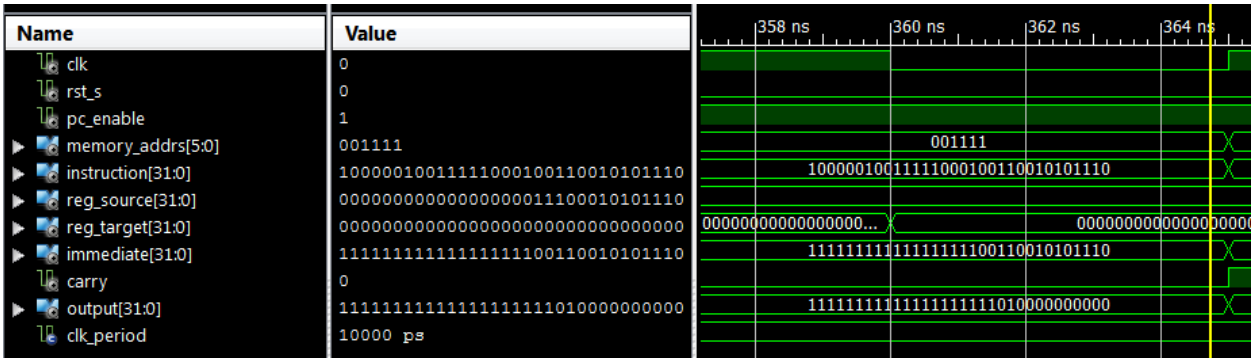
ORI



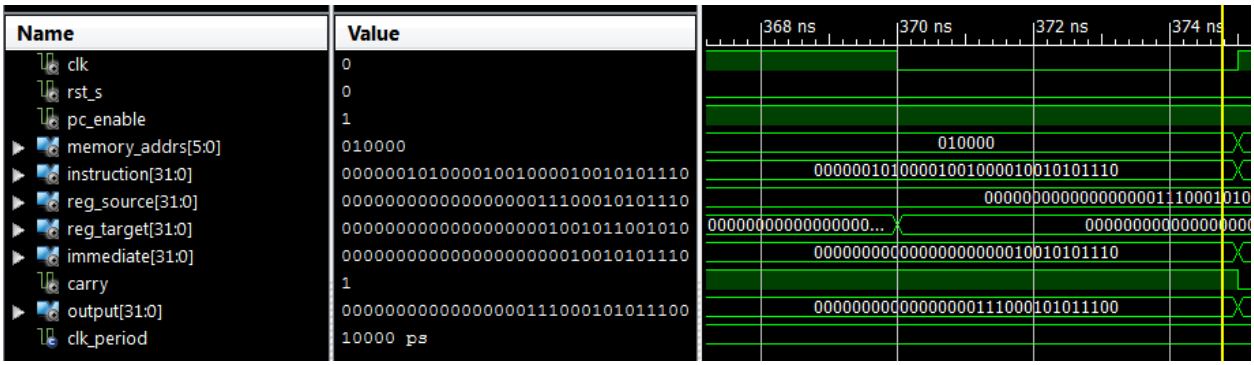
XOR



XORI



SLL



SLLI