IIT Armour College of Engineering
ILLINOIS INSTITUTE OF TECHNOLOGY

ECE 429 – INTRODUCTION TO VLSI DESIGN (FALL 2014)

FINAL PROJECT

CASE STUDY FOR 32-BIT PIPELINED CPU DESIGN WITH NEW ALU ARCHITECTURE

NAME: KRISHNA PRAMOD KANAKAPURA UMESH

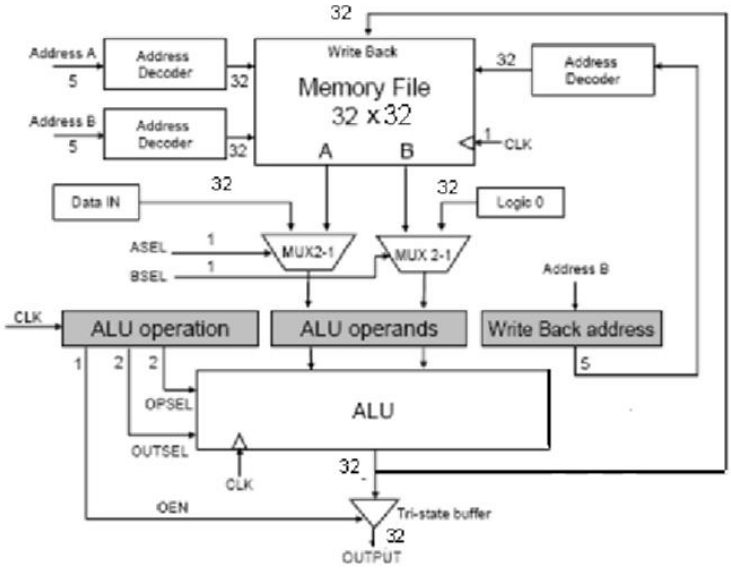
CWID: A20337195

INTRODUCTION:

The main objective of this project is to understand the working of a 32-bit Pipelined Central Processing Unit (CPU). As the name of the project indicates, the word length of the data used in the circuits is 32 bits. Also, since the CPU is pipelined, we are able to execute more than one instruction simultaneously. An externally set clock signal is used which synchronizes with the operation of the circuit. The other objective of this project is to determine minimum operating period using the critical data path delay of the circuit.

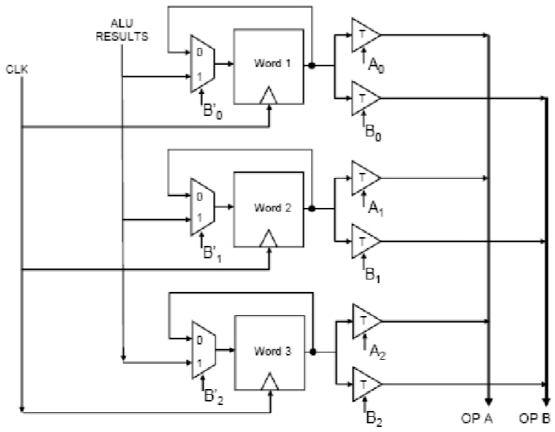
CIRCUIT DESCRIPTION:

Figure below shows the primary building blocks and signals of the CPU. As shown in below fig., ALU and the memory file are the primary building blocks. The capture and release of data within the memory file is synchronized by external clock signal. The CPU is pipelined and each instruction is explained in two clock cycles. In the first clock cycle, the external address selection signals used for specifying the contents of the memory file is decoded using the two decoders. The operands used to perform the operation in the ALU are selected by the multiplexer blocks. In the second clock cycle, the specified operation is executed by the ALU. The tri-state buffer is used to read the results of the ALU from the outside of the CPU depending upon the externally specified OEN (output enable) signal value. Lastly, the ALU results is written back in the memory file through the word specified by the Address B.



MEMORY FILE:

The figure below shows the internal configuration of the memory file. The memory file of this design is capable of storing 32 32-bit words. There is one write port and two read ports for the memory file. The external 5-bit words Address A and Address B are used to specify the words to be read in each clock cycle. The D-register is the primary storing element in the memory file.



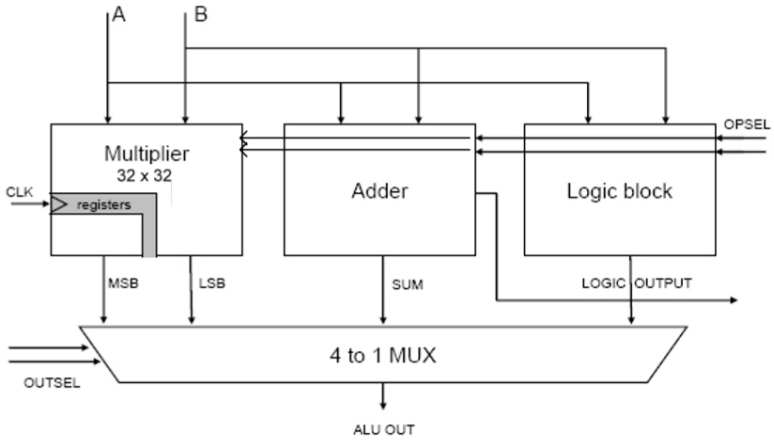
ARITHMETIC LOGIC UNIT (ALU):

Block diagram of the ALU circuit is as shown in the below figure. The ALU shown in the below figure has two operands A and B and can implement the following eight functions:

- A * B : multiplication
- A + B : addition
- A - B : subtraction
- B - A : subtraction
- A or B : logic OR function
- A and B : logic AND function
- A xor B : logic XOR function
- A xnor B : logic XNOR function

Operands A and B selection are done through the external signals ASEL and BSEL respectively. The multiplier, the adder and the logic function block are the three function blocks of the ALU.

The multiplier executes the multiplication function which is executed in 3 clock cycles. The addition and subtraction operations of the ALU are performed by the adder circuit which is a 32-bit circuit. The operation selection is done by the externally defined operation select signals OPSEL. This signal is also used to specify the operation executed in the logic block. The output signal OUTSEL is used to specify the final output of the ALU.



SYNCHRONIZATION:

The external control signals determine the instruction to be executed by the CPU. Figure below shows an example of an instruction word. The instruction is executed in two steps in the CPU. In the first step, the signals that control the operation of the CPU will specify the contents of the memory file to be read from the read ports A and B. It will also specify the operands of the ALU. In the second step, the control signals will determine the operation to be executed by the ALU and the result of the ALU.

CASE STUDY- 1

32-BIT CPU DESIGN WITH - CARRY RIPPLE ADDER, CARRY LOOKAHEAD ADDER, CARRY SKIP ADDER, AND CARRY SELECT ADDER

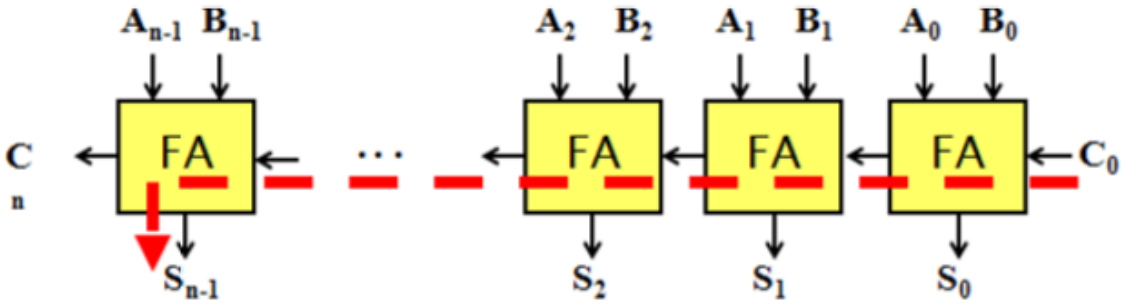
INTRODUCTION:

The main objective of this case study is to perform the logical synthesis and physical synthesis of the CPU with Carry Ripple Adder, Carry Lookahead Adder, Carry Skip Adder and Carry Select Adder using ASIC flow.

EXPLANATION OF THE DIFFERENT ADDERS:

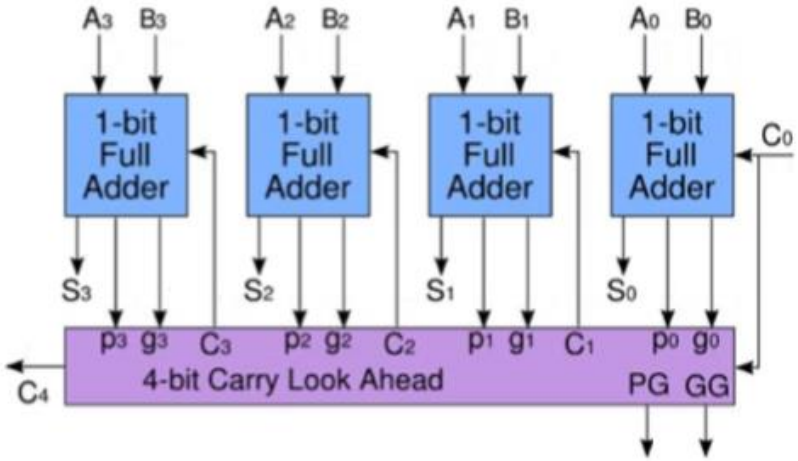
1. CARRY RIPPLE ADDER:

The block diagram of the Carry Ripple Adder is shown in below figure. An n-bit CRA is formed by concatenating ‘n’ Full Adders in series. The carry output of one full adder is connected as the carry input to the next full adder.



2. CARRY LOOKAHEAD ADDER:

These types of adders are special circuits which are used to reduce the time to perform addition operation but at the expense of a more complex hardware. The block diagram of the Carry Lookahead adder is as shown in below figure.



where

$$C_1 = g_0 + p_0 C_0$$

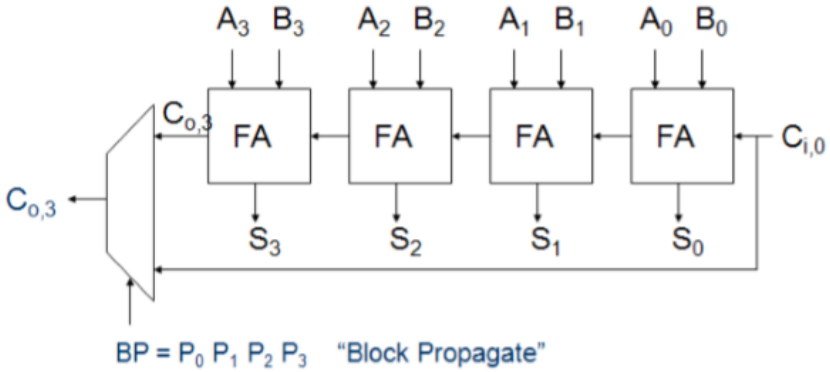
$$C_2 = g_1 + p_1(g_0 + p_0 C_0) = g_1 + p_1 g_0 + p_1 p_0 C_0$$

$$C_3 = g_2 + p_2(g_1 + p_1(g_0 + p_0 C_0)) = g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 C_0$$

*Note: $p_i = A_i + B_i$ (or $A_i \oplus B_i$) and $g_i = A_i \bullet B_i$

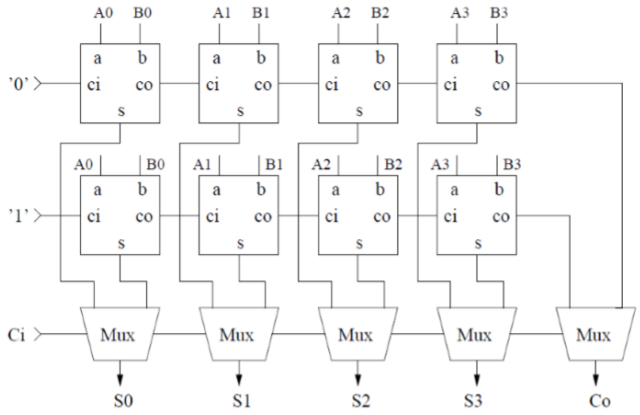
3. CARRY SKIP ADDER:

In this type of adders the operands are divided into r-bit blocks. Each block can produce the sum bits and a carryout bit by utilizing the ripple carry adder. The block diagram of a Carry Skip Adder is as below.

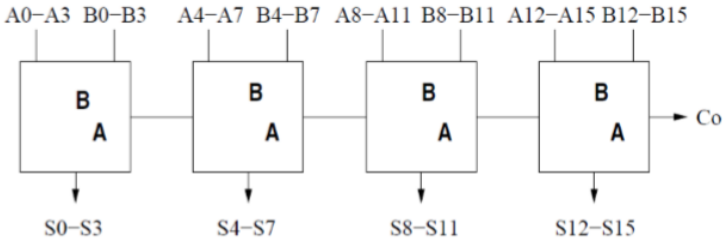


4. CARRY SELECT ADDER:

The Carry Select Adder also divides the operands to be added into r bit blocks similar to carry skip adder. But here in each block, 2 r-bit ripple carry adders operates simultaneously to produce 2 sets of sum bits and carryout signals. Each ripple carry adder has 2 carry in signals. One ripple carry adder has a carry in of 1 and the other has a carry in of 0. Figure below show the structure of a 4-bit Carry Select Adder.



The structure of a 16-bit Carry Select Adder is as shown below.

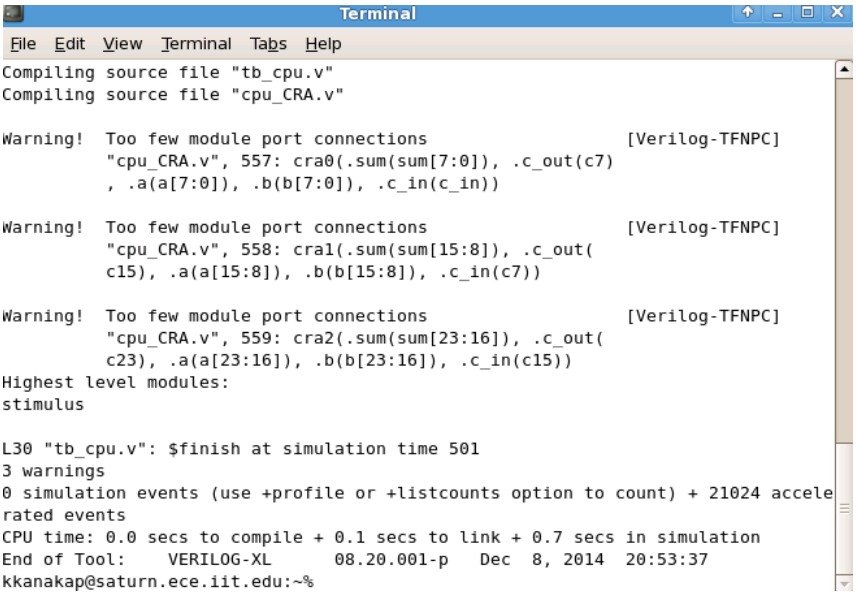


REPORT FOR CASE STUDY-1:

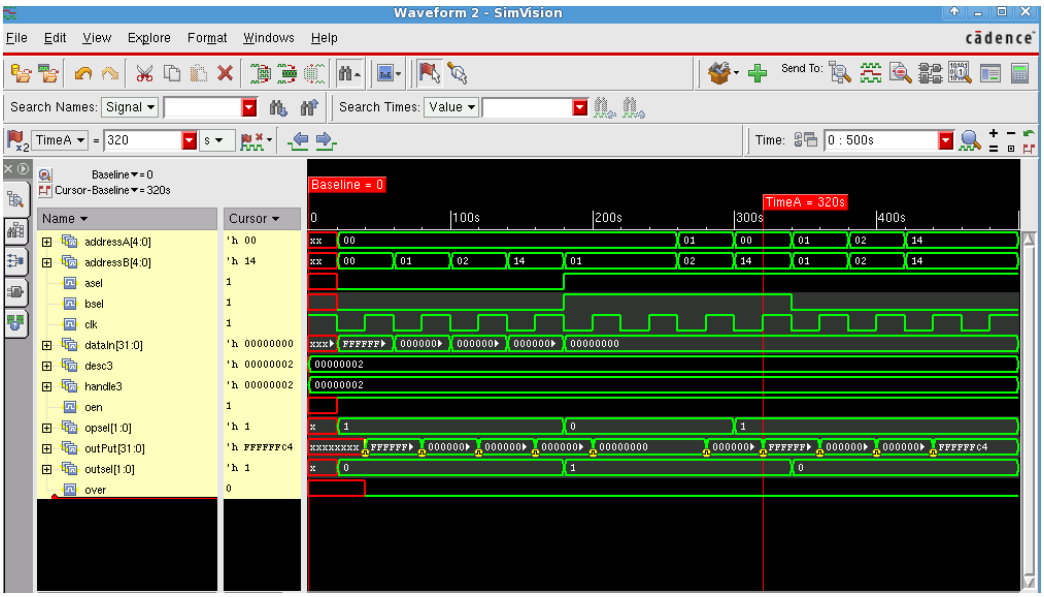
FOR TEST BENCH tb_cpu.v:

1. CARRY RIPPLE ADDER

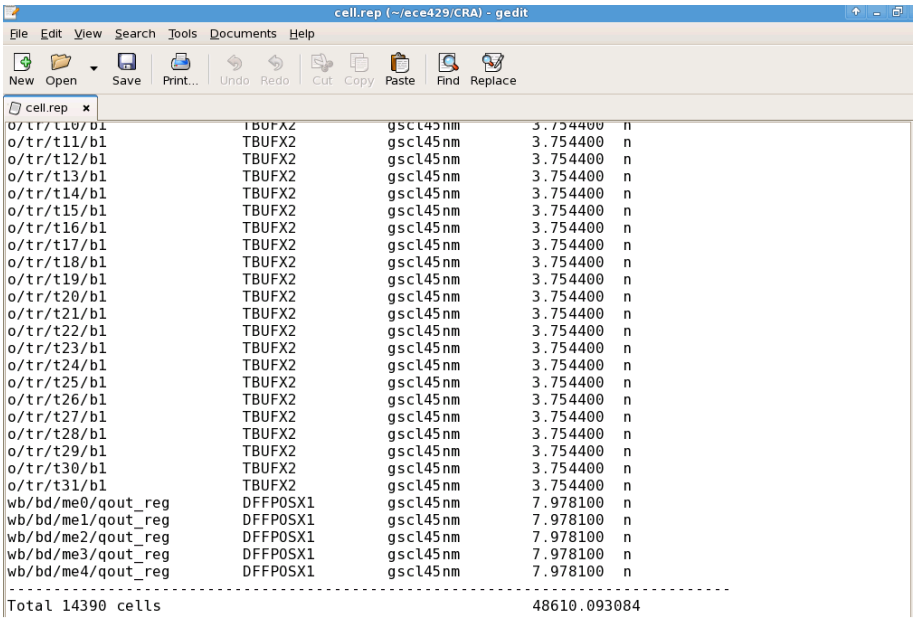
RTL Simulation output:



Simvision output:

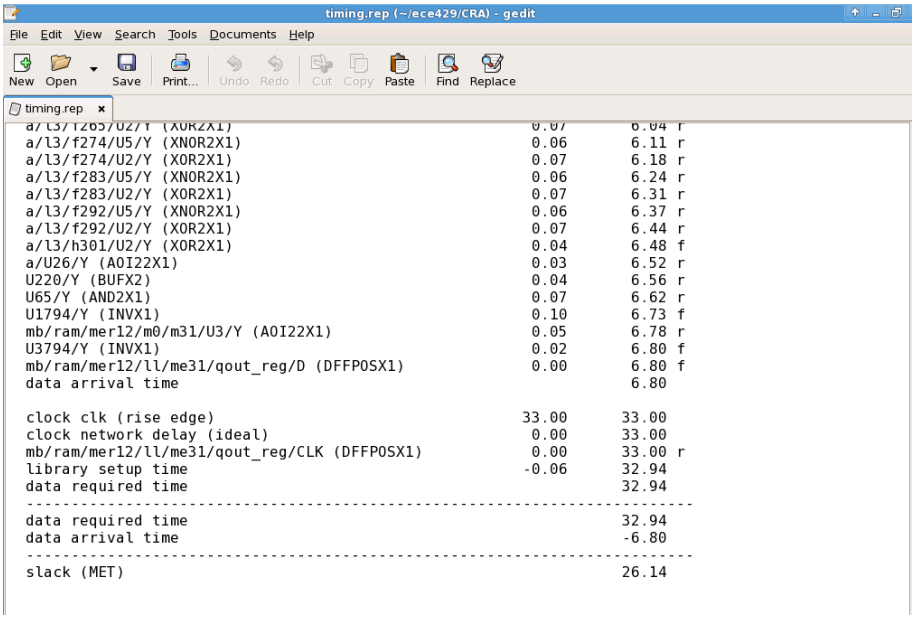


cell.rep:



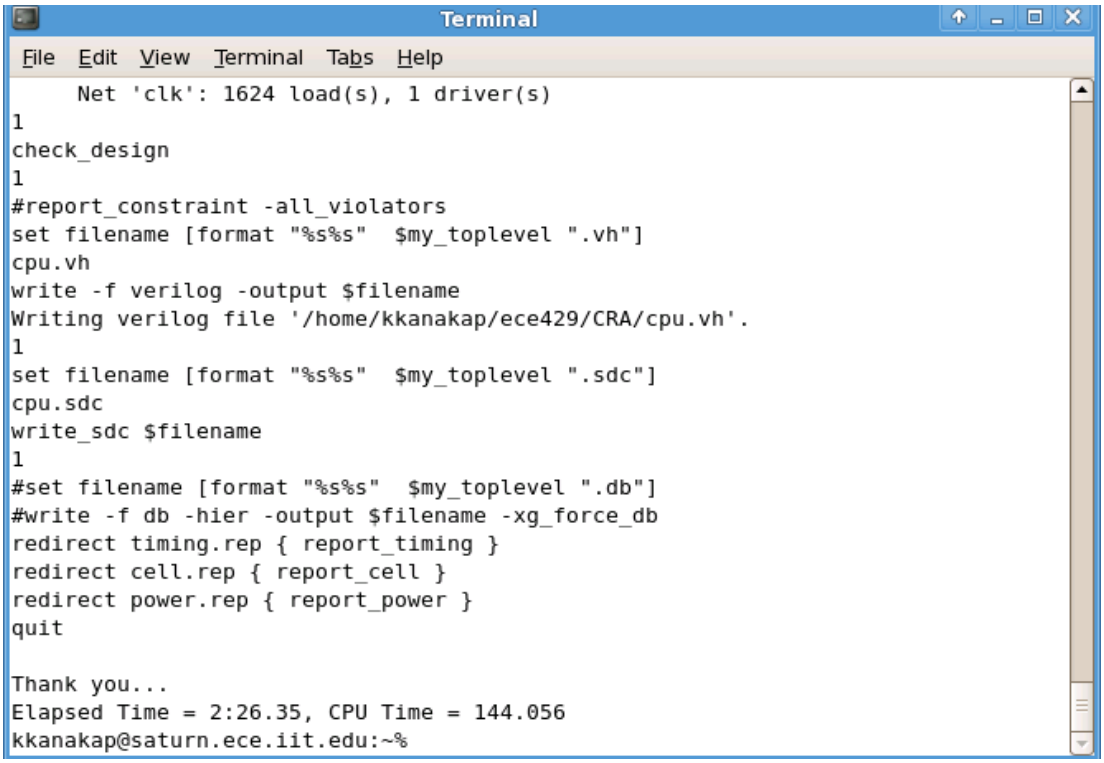
The file cell.rep contains the area of each cell in the module and gives the total area of the module.

timing.rep:

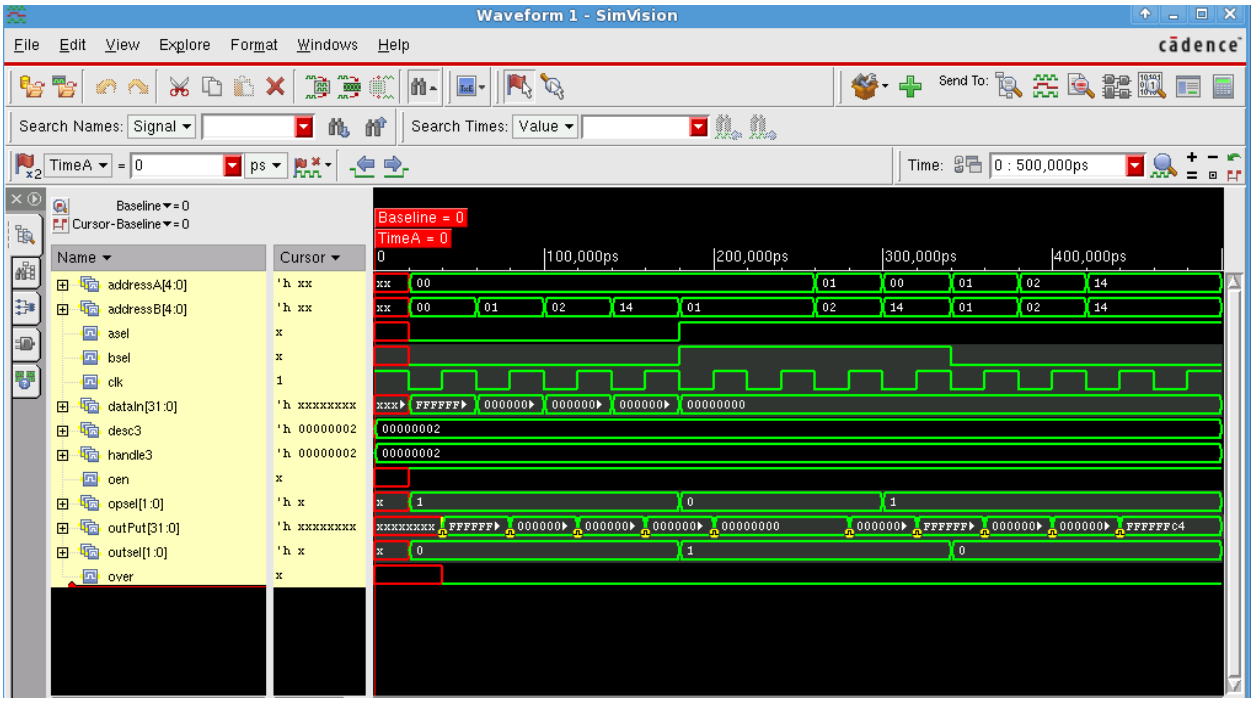


The file timing.rep contains the path delay, critical delay and total path delay of the design.

Post-synthesis Simulation:



Simvision output:



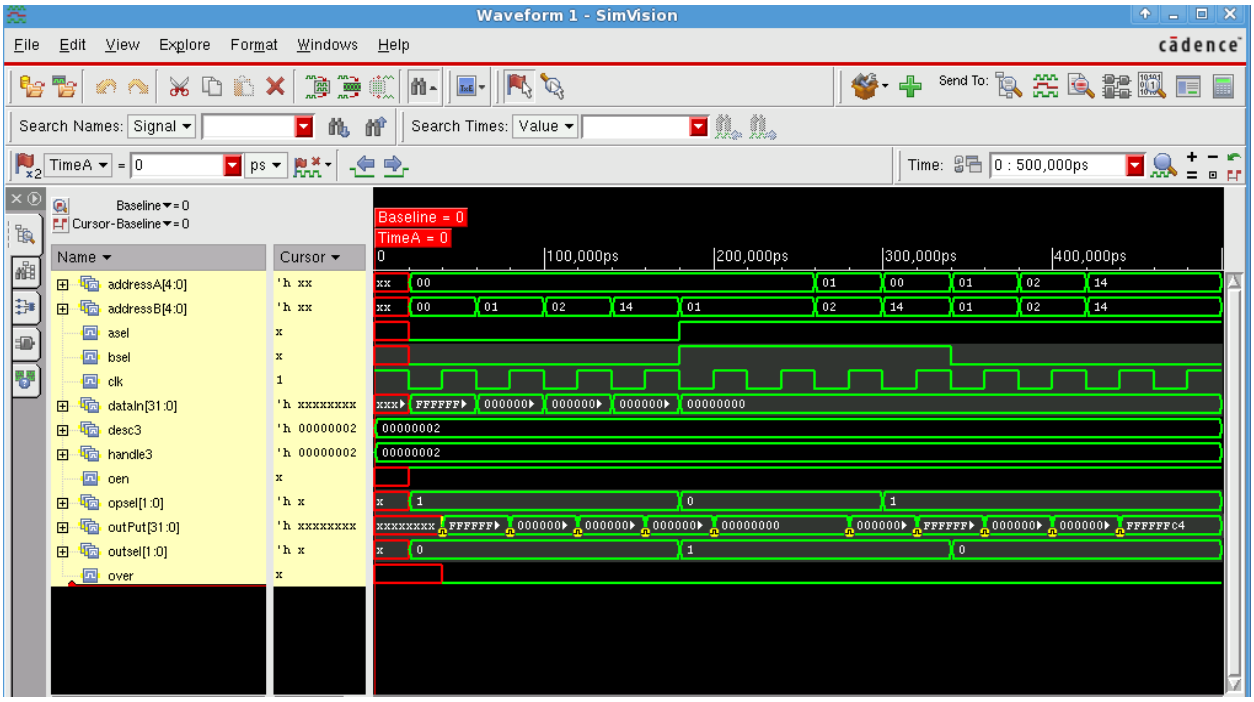
Maximum clock frequency = 96.77 KHz

Post-P&R Simulation:

```
CPU time: 0.2 secs to compile + 0.3 secs to link + 0.2 secs in simulation
End of Tool:    VERILOG-XL      08.20.001-p   Dec  4, 2014  18:23:50
encounter 2> reportGateCount -limit 0
Gate area 2.8158 um^2
[0] cpu Gates=15749 Cells=14466 Area=44346.5 um^2
encounter 3>
```

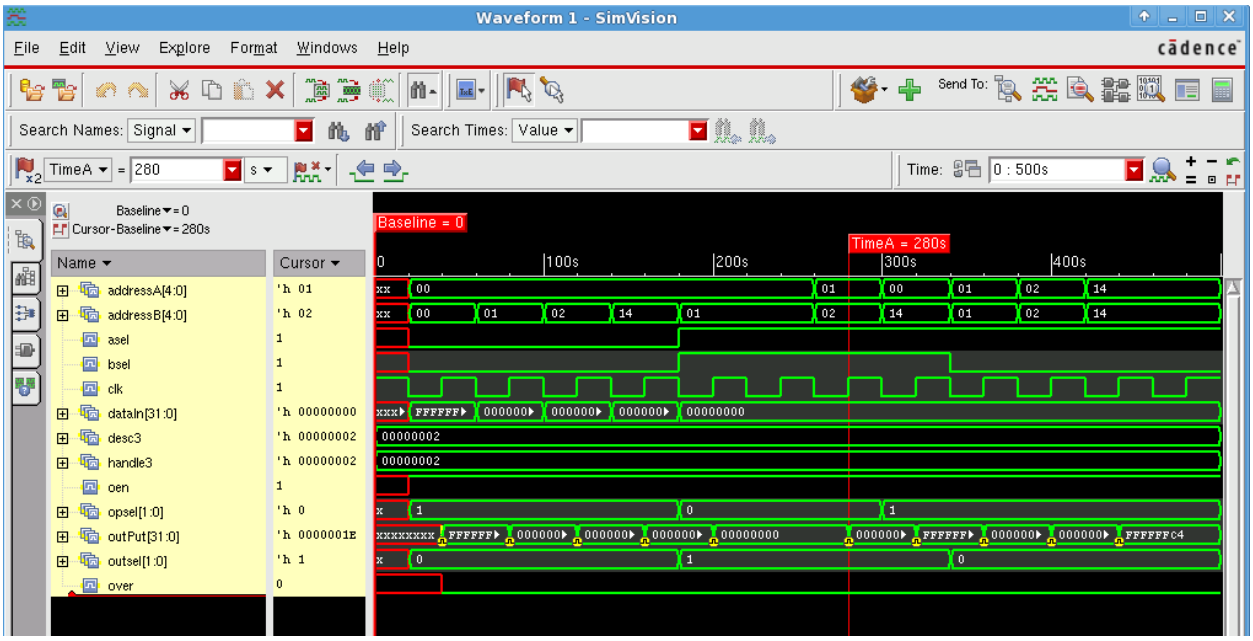
```
--------------------------------------------------------------------------------
---------
*       Power Distribution Summary:
*               Highest Average Power:              clk__L4_I32 (INVX8):
 0.009632
*               Highest Leakage Power: mb/ram/mer2/ll/me31/qout_reg (DFFPOSX1):
 5.498e-05
*               Total Cap:      1.80592e-10 F
*               Total instances in design: 14466
*               Total instances in design with no power:     0
*                Total instances in design with no activty:     0

*               Total Fillers and Decap:     0
--------------------------------------------------------------------------------
```

Simvision output:

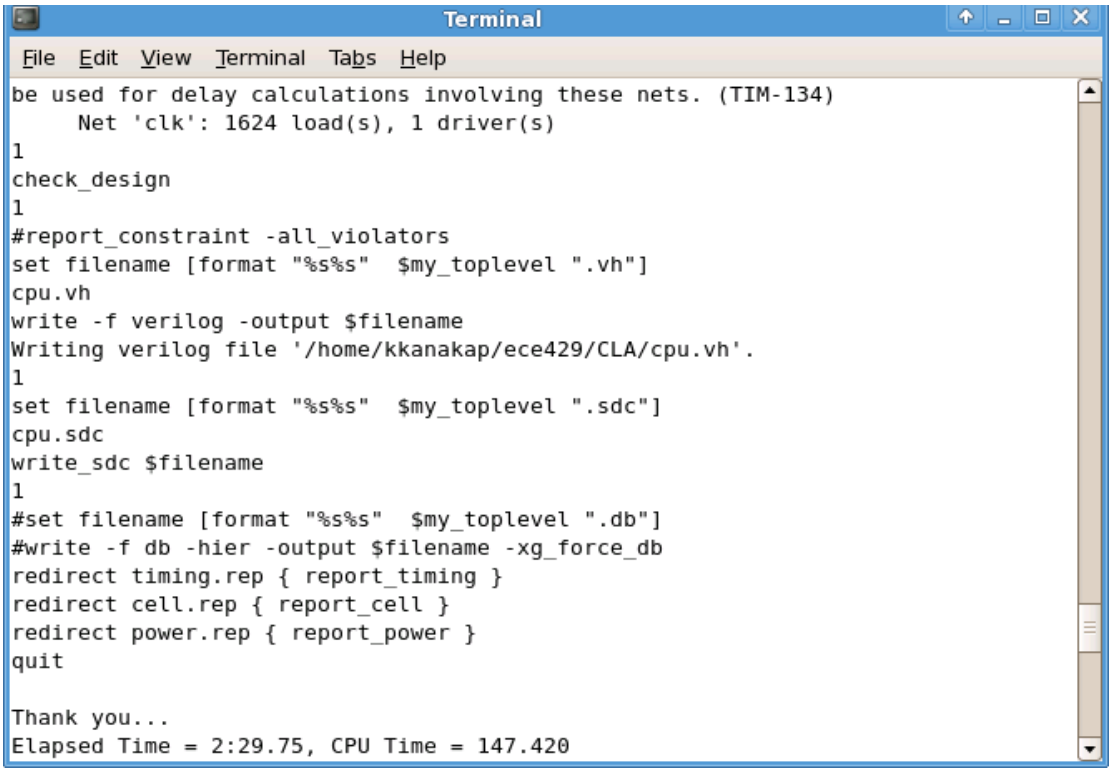


2. CARRY LOOKAHEAD ADDER:

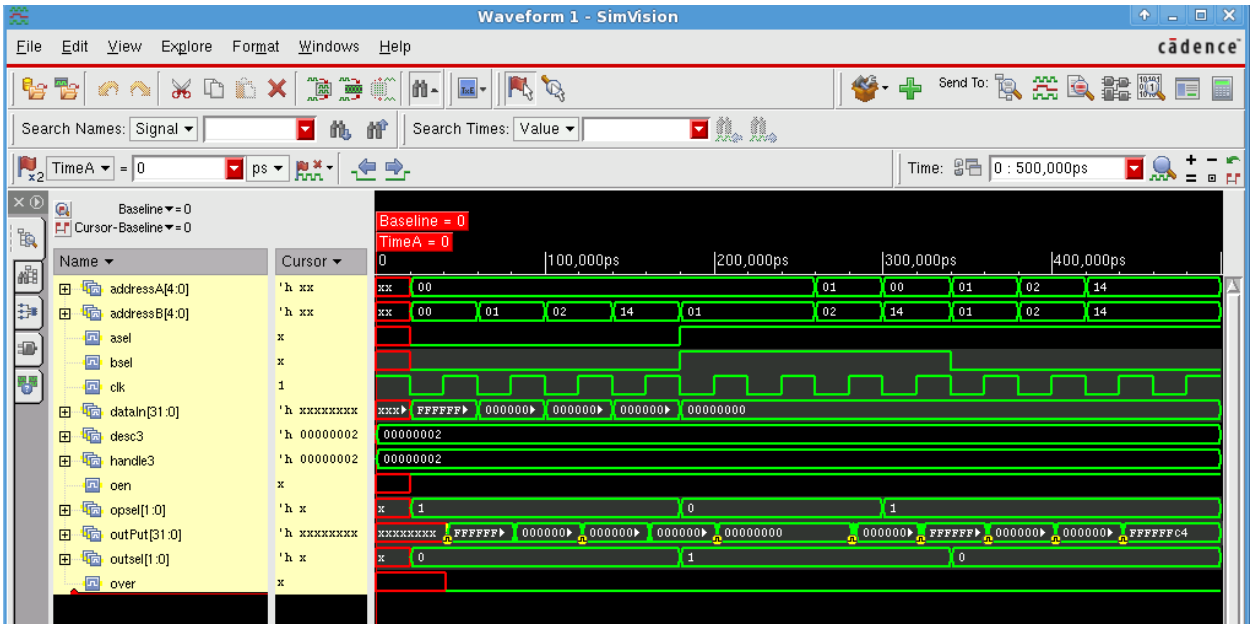
RTL Simulation:



Post Synthesis Simulation:



Simvision Output:



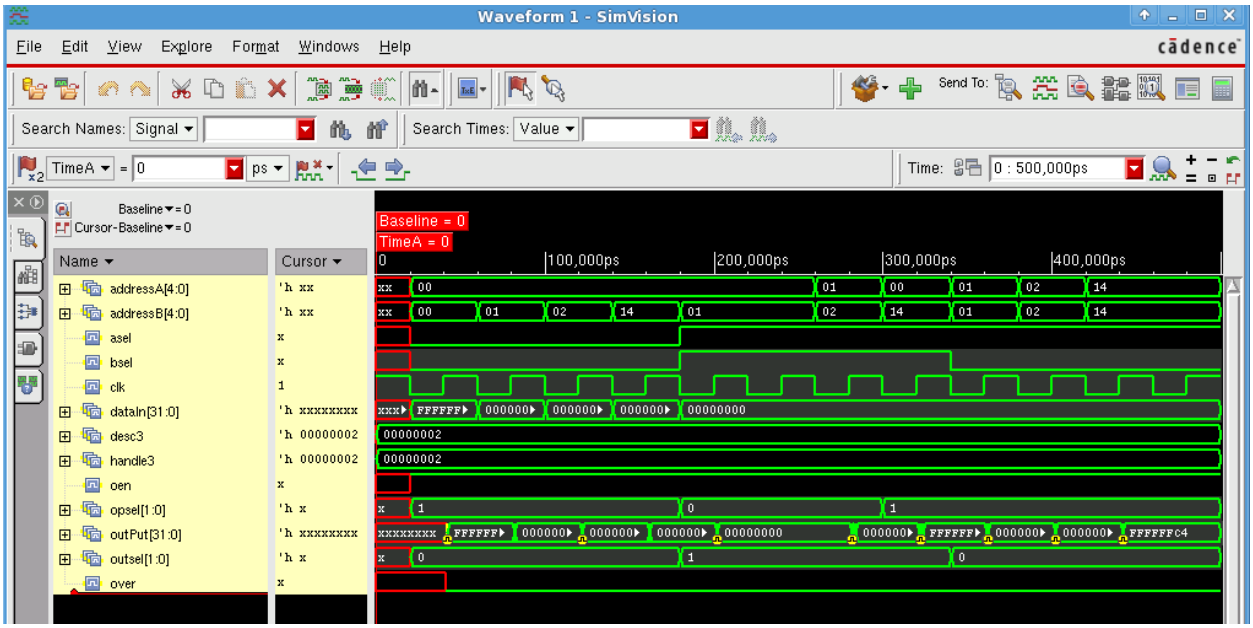
Maximum clock frequency = 96.53 KHz

Post-P&R Simulation:

```
CPU time: 0.2 secs to compile + 0.3 secs to link + 0.2 secs in simulation
End of Tool:    VERILOG-XL      08.20.001-p   Dec  4, 2014  19:01:36
encounter 2> reportGateCount -limit 0
Gate area 2.8158 um^2
[0] cpu Gates=15744 Cells=14446 Area=44333.4 um^2
encounter 3>
```

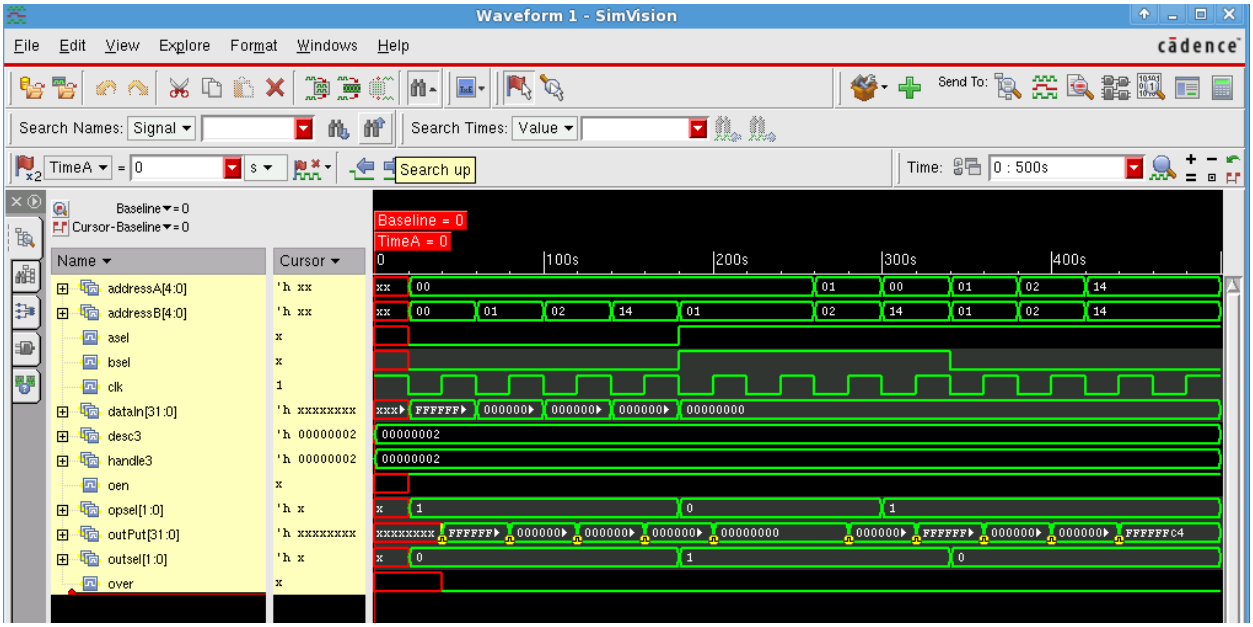
```
--------------------------------------------------------------------------------
---------
*       Power Distribution Summary:
*               Highest Average Power:              clk__L4_I28 (INVX8):
 0.009625
*               Highest Leakage Power: mb/ram/mer2/ll/me31/qout_reg (DFFPOSX1):
 5.498e-05
*               Total Cap:      1.80425e-10 F
*               Total instances in design: 14446
*               Total instances in design with no power:     0
*                Total instances in design with no activty:     0

*               Total Fillers and Decap:     0
--------------------------------------------------------------------------------
```

Simvision output:

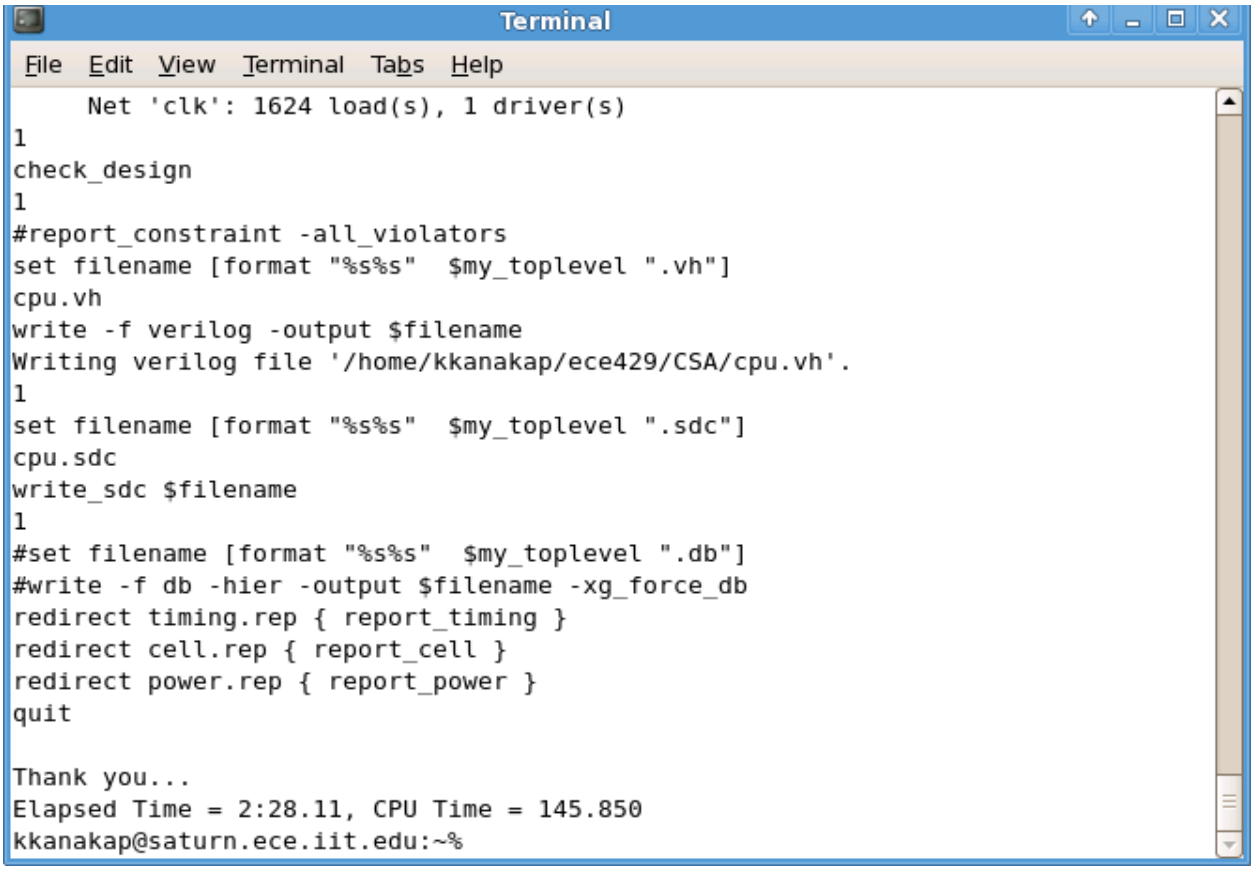


3. CARRY SKIP ADDER:

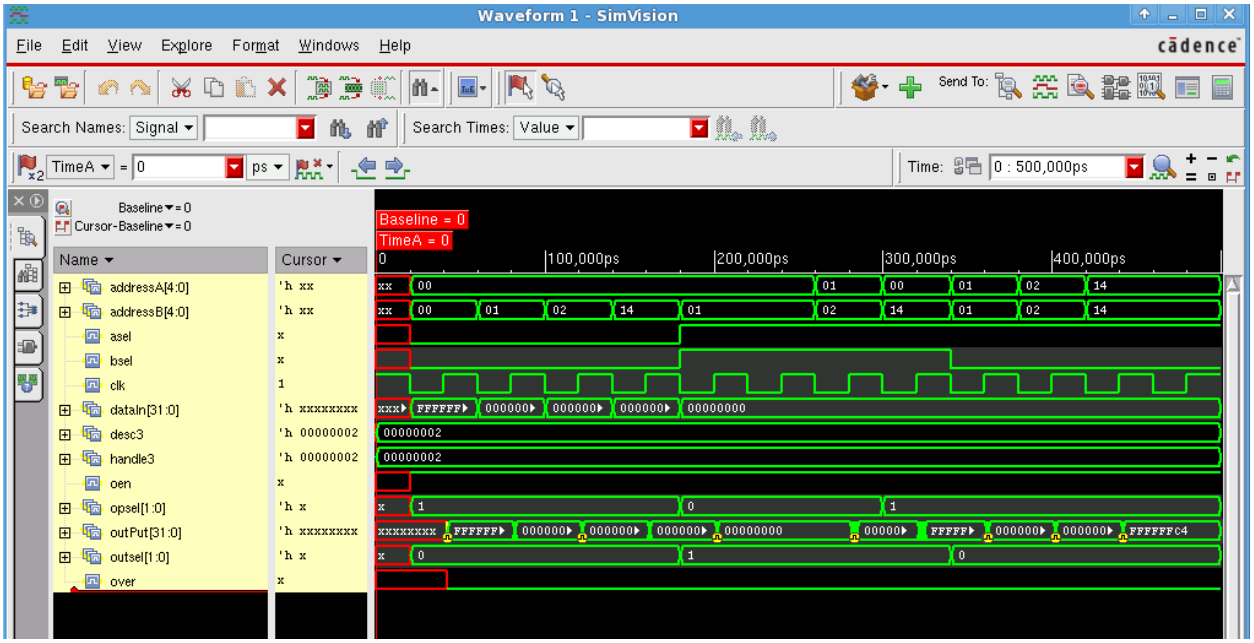
RTL Simulation:



Post Synthesis Simulation:



Simvision output:

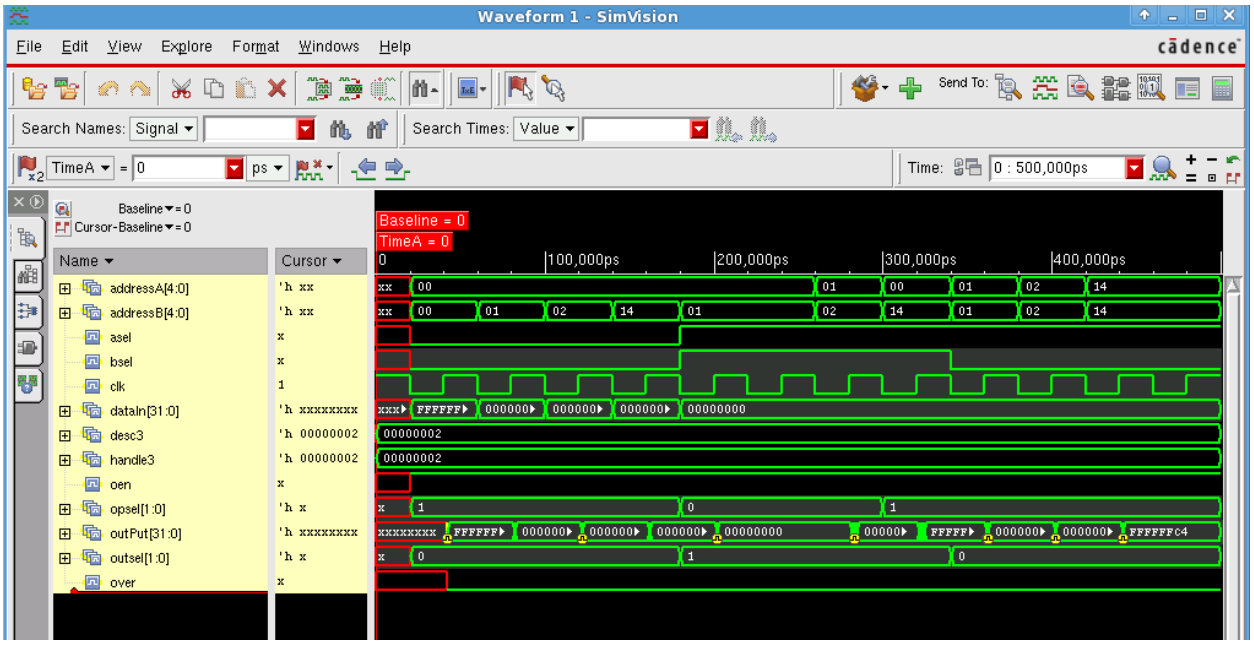


Maximum clock frequency = 95.59 KHz

Post-P&R Simulation:

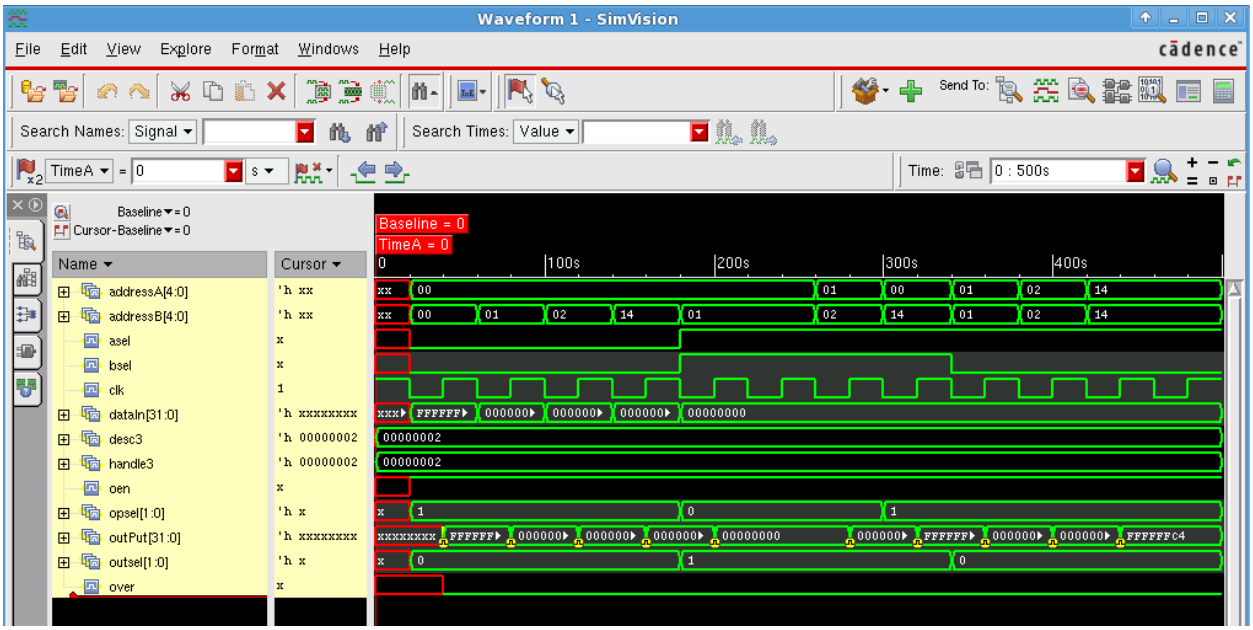
```
L30 "tb_cpu.v": $finish at simulation time 50100
0 simulation events (use +profile or +listcounts option to count) + 207877 accel
erated events + 345558 timing check events
CPU time: 0.2 secs to compile + 0.3 secs to link + 0.2 secs in simulation
End of Tool:    VERILOG-XL      08.20.001-p   Dec  4, 2014  19:23:16
encounter 2> reportGateCount -limit 0
Gate area 2.8158 um^2
[0] cpu Gates=15848 Cells=14565 Area=44626.2 um^2
encounter 3>
```

```
---------
*       Power Distribution Summary:
*               Highest Average Power:                clk__L4_I7 (INVX8):
 0.009491
*               Highest Leakage Power: mb/ram/mer2/ll/me31/qout_reg (DFFPOSX1):
 5.498e-05
*               Total Cap:      1.81192e-10 F
*               Total instances in design: 14565
*               Total instances in design with no power:     0
*                Total instances in design with no activty:     0

*               Total Fillers and Decap:     0
--------------------------------------------------------------------------------
```

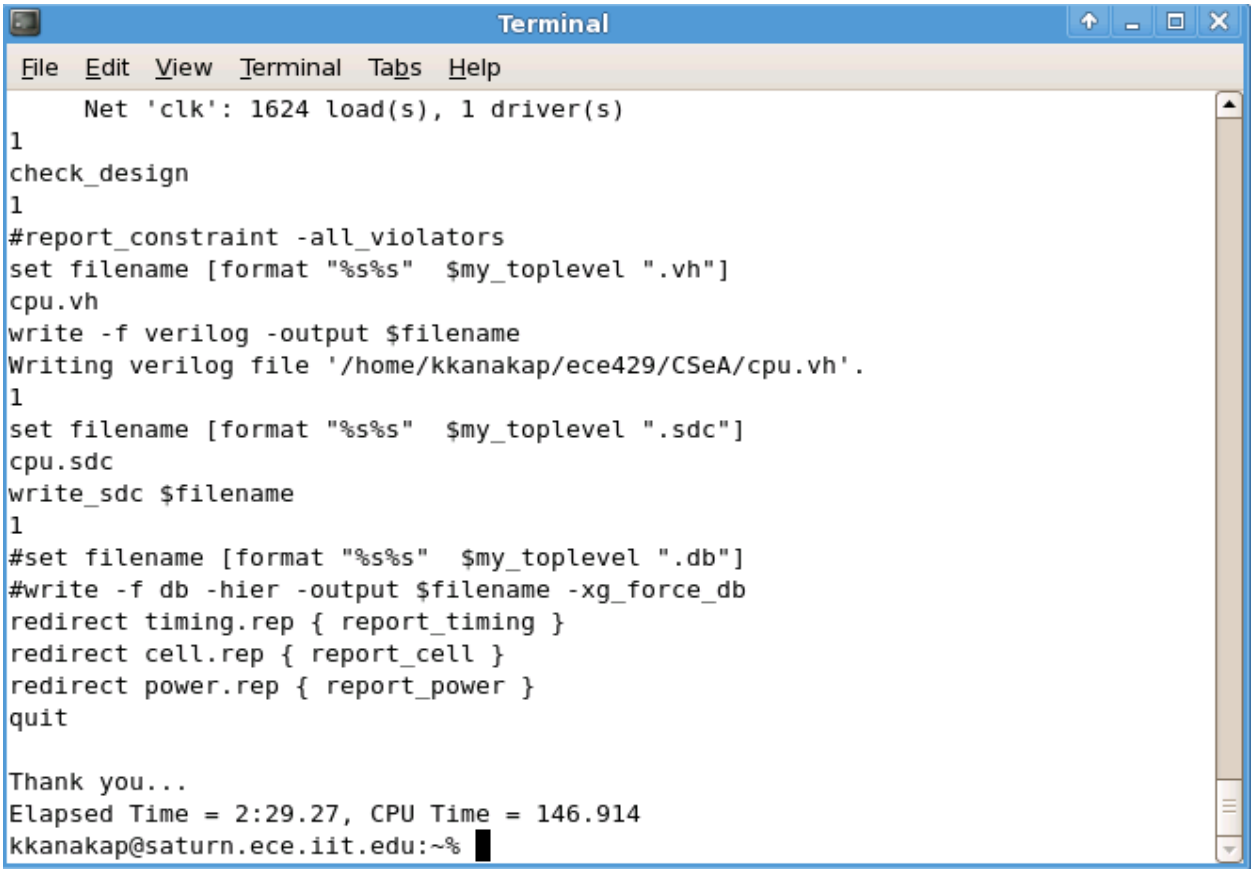


4. CARRY SELECT ADDER:

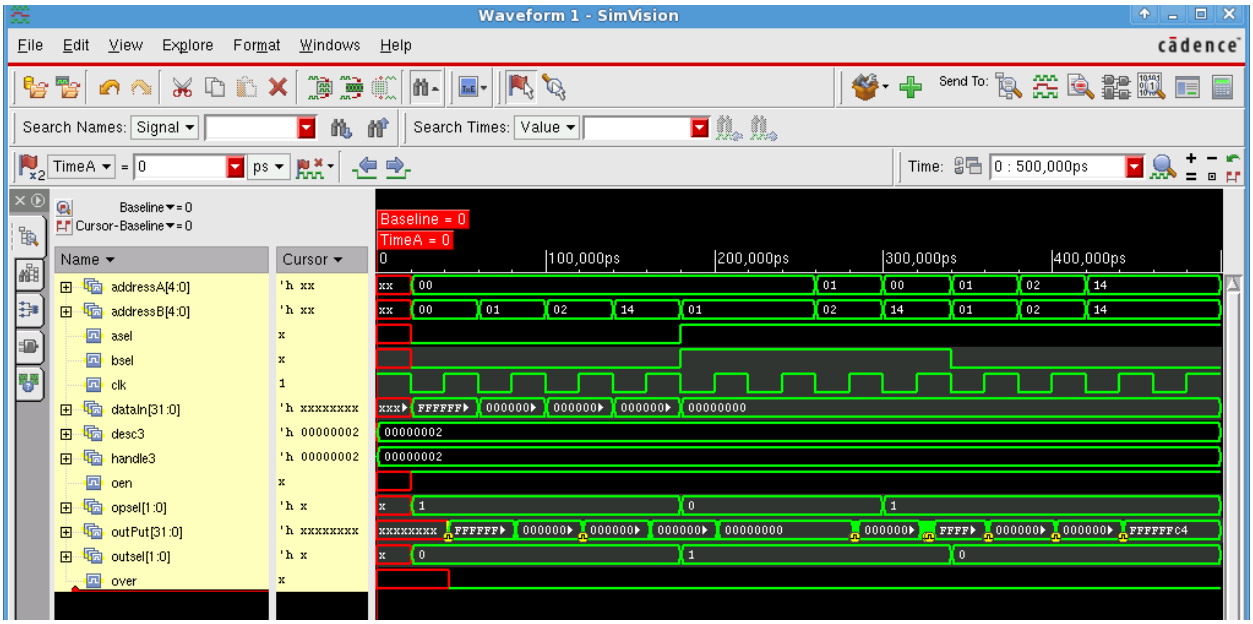
RTL Simulation:



Post Synthesis Simulation:



Simvision output:



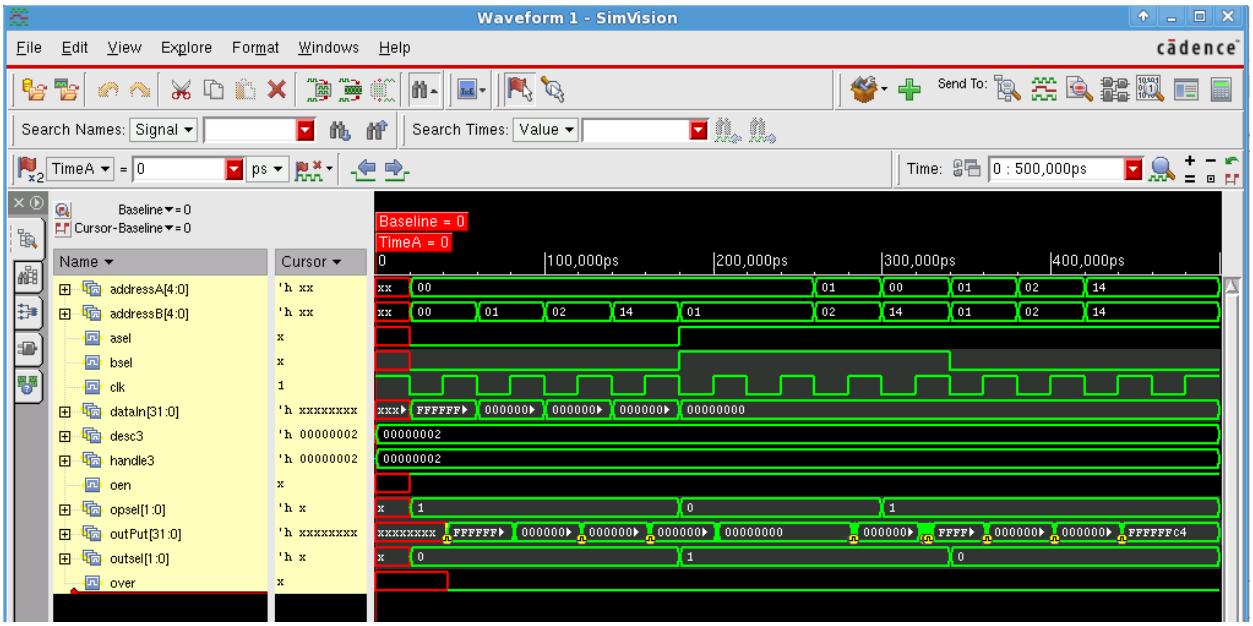
Maximum clock frequency = 94.32 KHz

Post-P&R Simulation:

```
L30 "tb_cpu.v": $finish at simulation time 50100
0 simulation events (use +profile or +listcounts option to count) + 211984 accel
erated events + 345638 timing check events
CPU time: 0.2 secs to compile + 0.3 secs to link + 0.3 secs in simulation
End of Tool:    VERILOG-XL      08.20.001-p   Dec  4, 2014  19:42:58
encounter 2> reportGateCount -limit 0
Gate area 2.8158 um^2
[0] cpu Gates=15933 Cells=14660 Area=44864.6 um^2
encounter 3>
```

```
---------
*       Power Distribution Summary:
*               Highest Average Power:              clk__L4_I26 (INVX8):
 0.009877
*               Highest Leakage Power: mb/ram/mer2/ll/me31/qout_reg (DFFPOSX1):
 5.498e-05
*               Total Cap:      1.84055e-10 F
*               Total instances in design: 14660
*               Total instances in design with no power:     0
*                Total instances in design with no activty:     0

*               Total Fillers and Decap:     0
--------------------------------------------------------------------------------
```

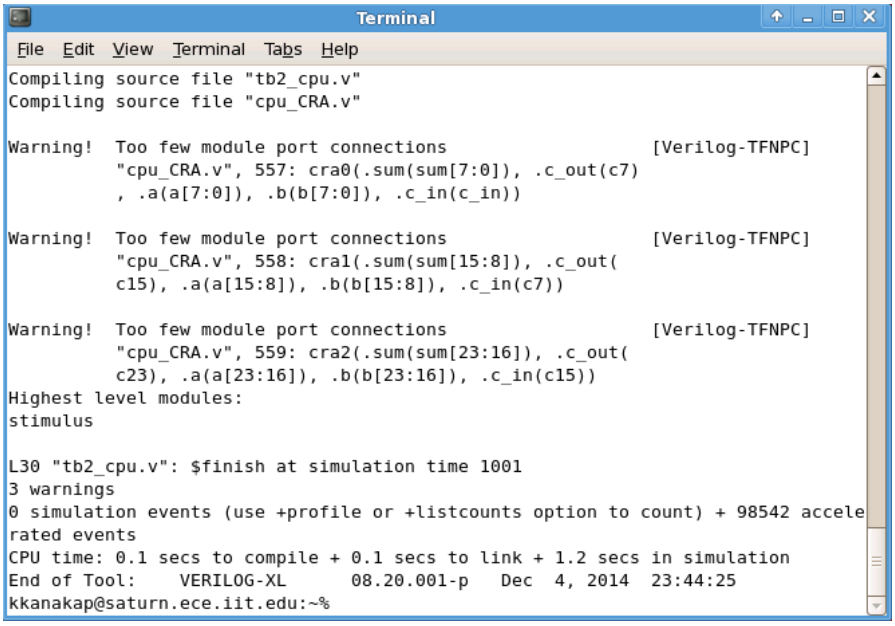
Simvision output:

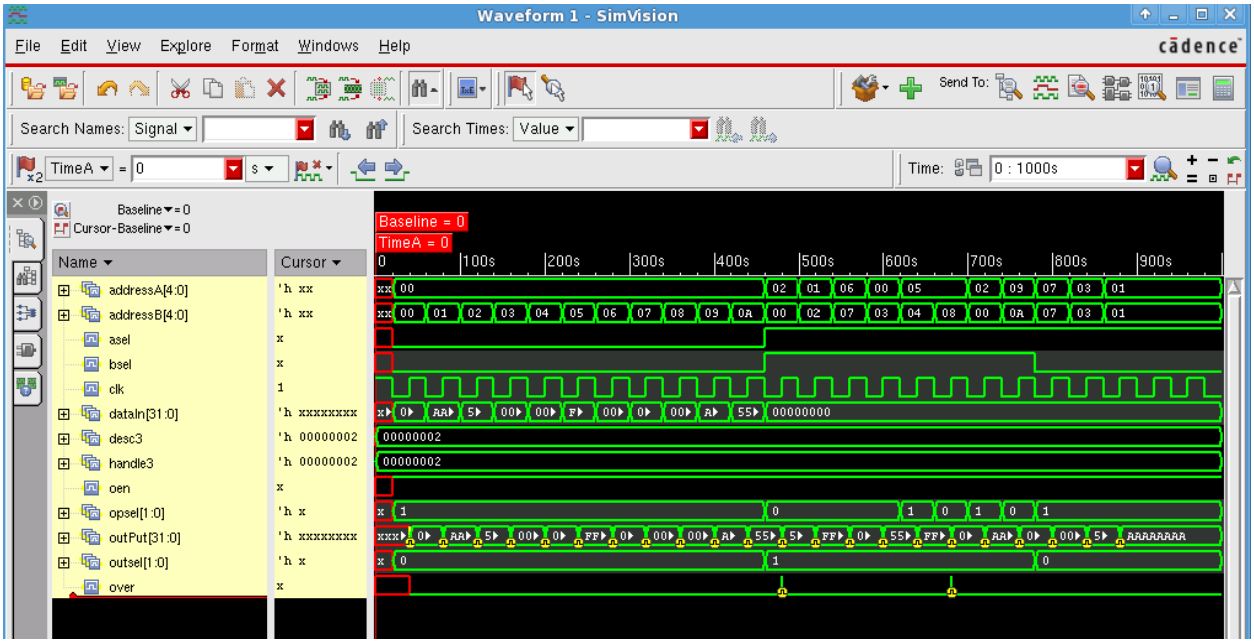


FOR TESTBENCH tb_test.v:

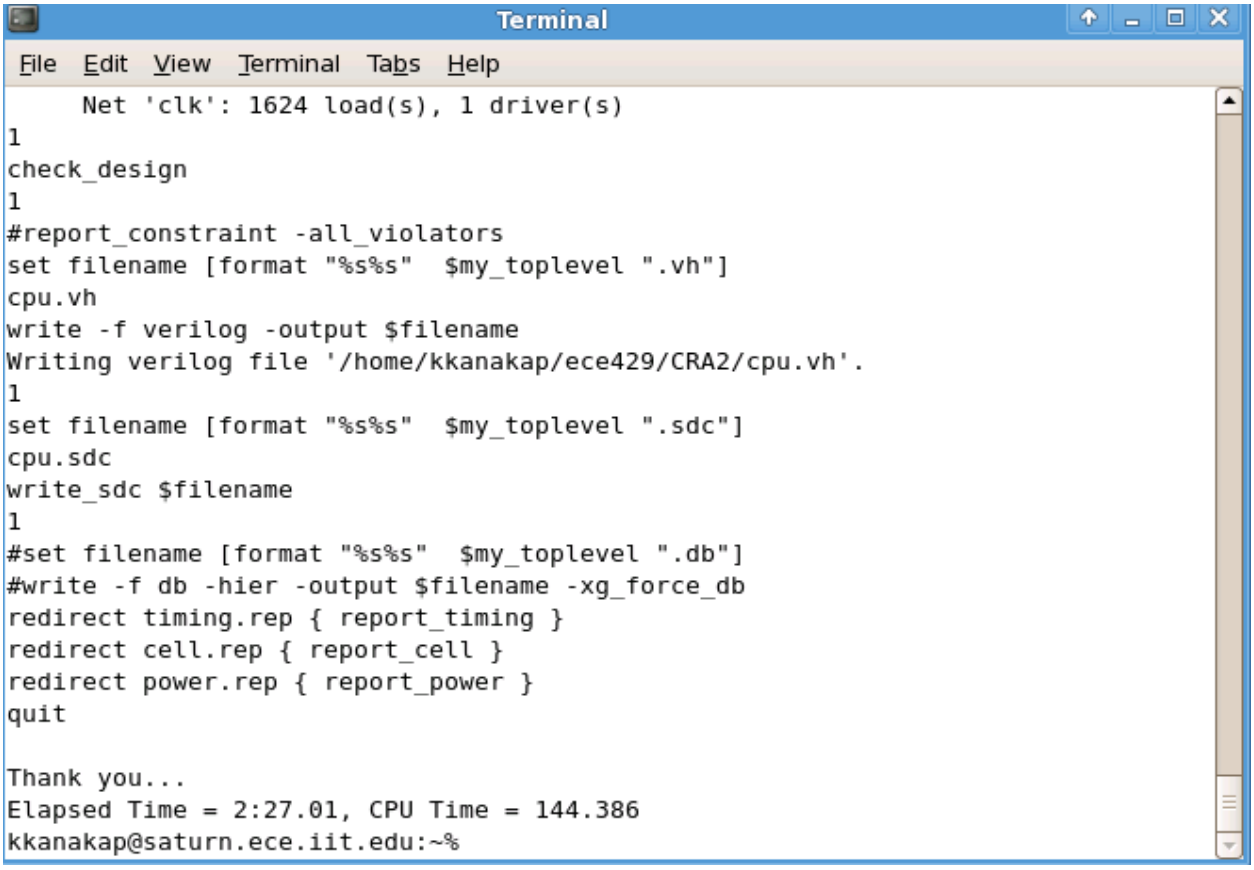
1. CARRY RIPPLE ADDER:

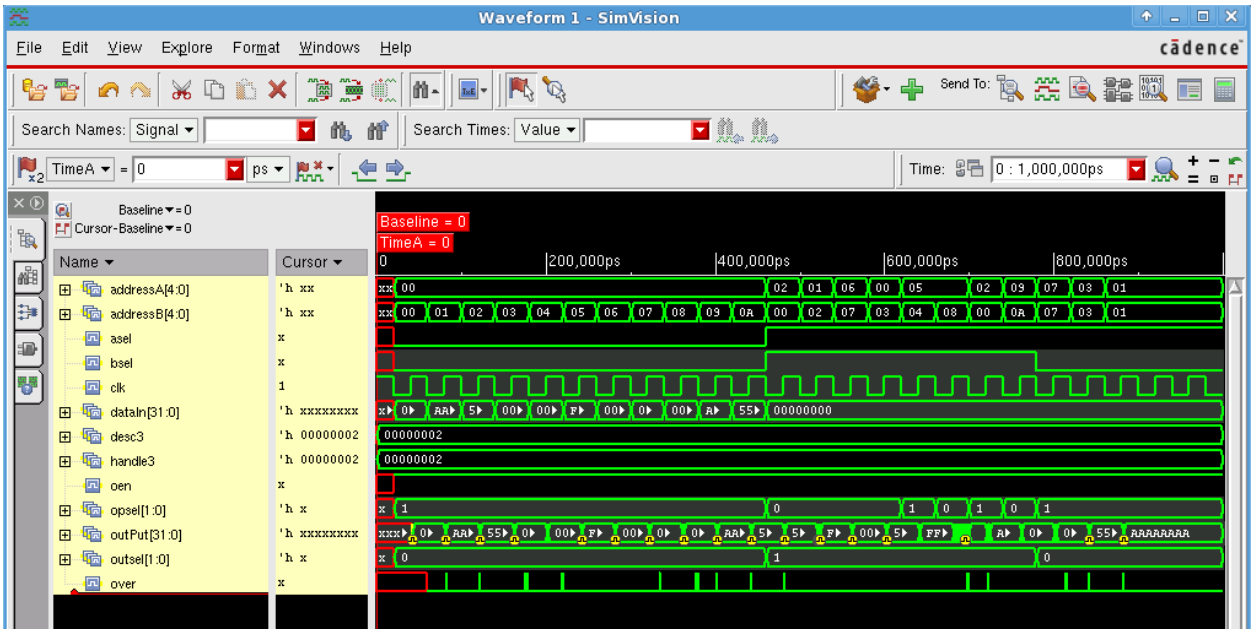
RTL Simulation:





Post Synthesis Simulation:

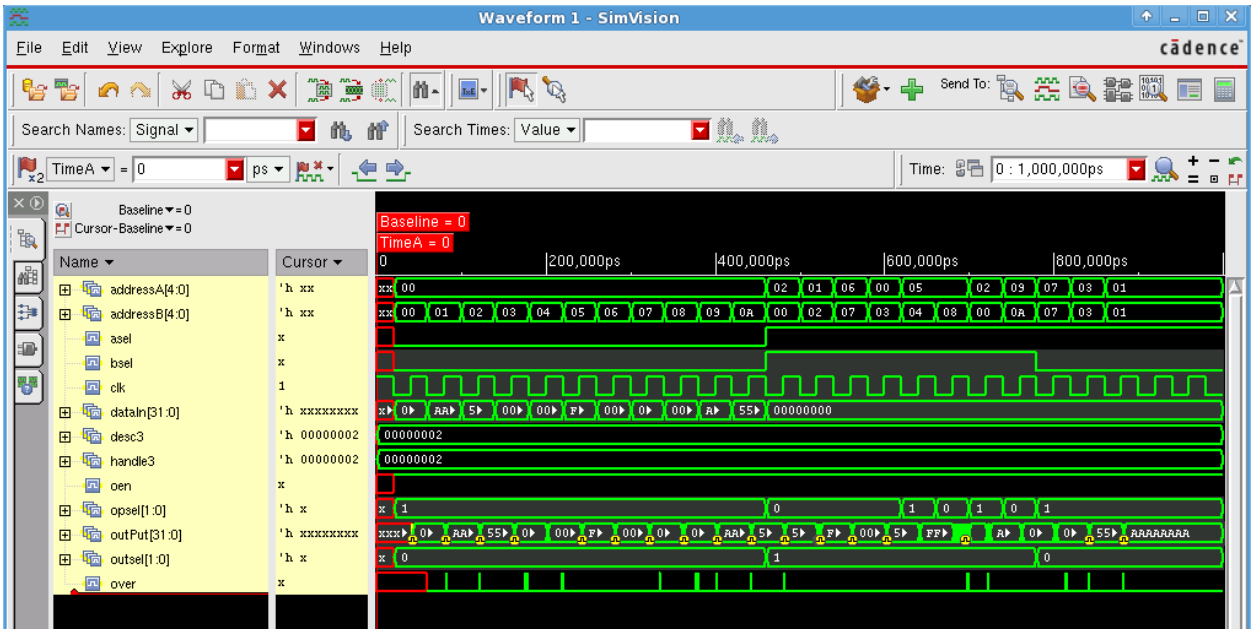




Post-P&R Simulation:

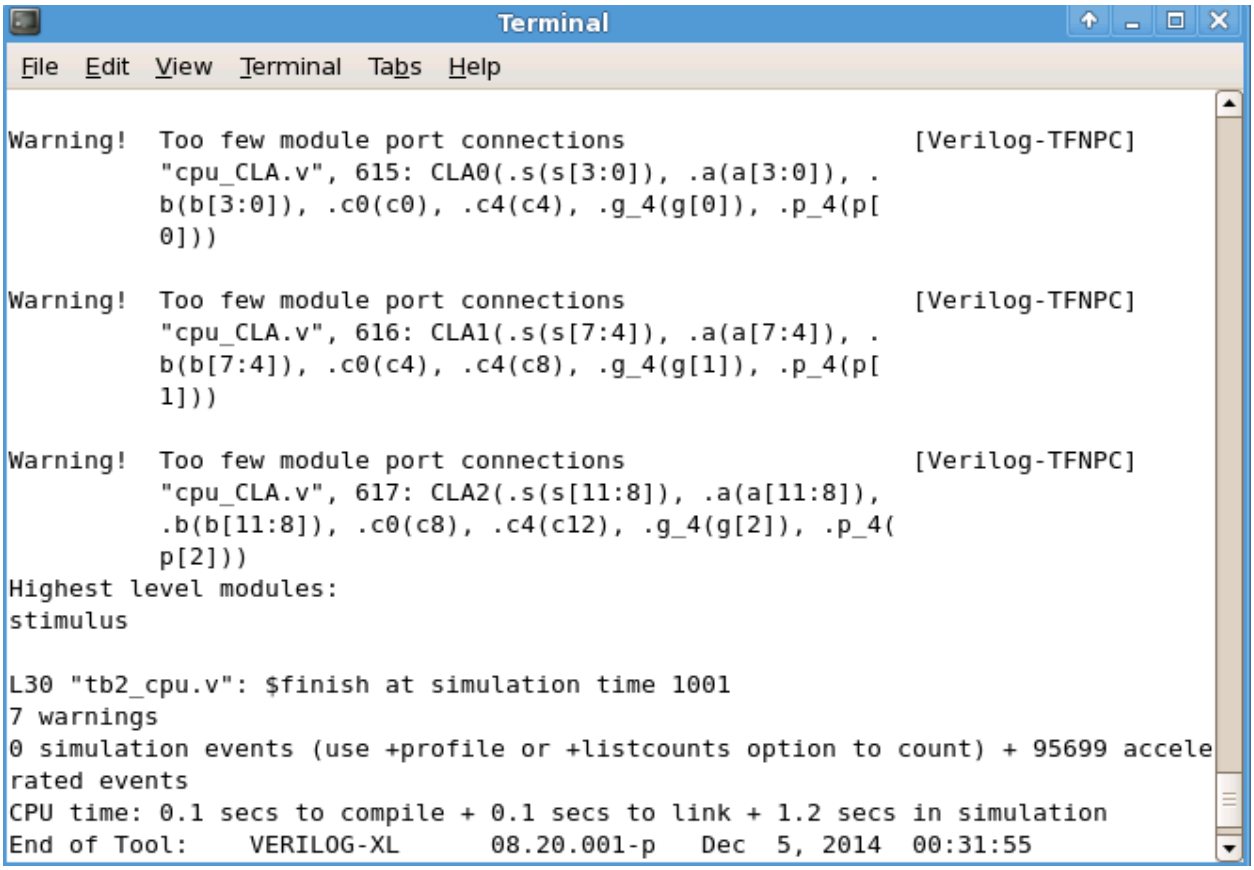
```
CPU time: 0.2 secs to compile + 0.3 secs to link + 0.5 secs in simulation
End of Tool:     VERILOG-XL      08.20.001-p    Dec  5, 2014  00:13:38
encounter 2> reportGateCount -limit 0
Gate area 2.8158 um^2
[0] cpu Gates=15749 Cells=14466 Area=44346.5 um^2
encounter 3>
```

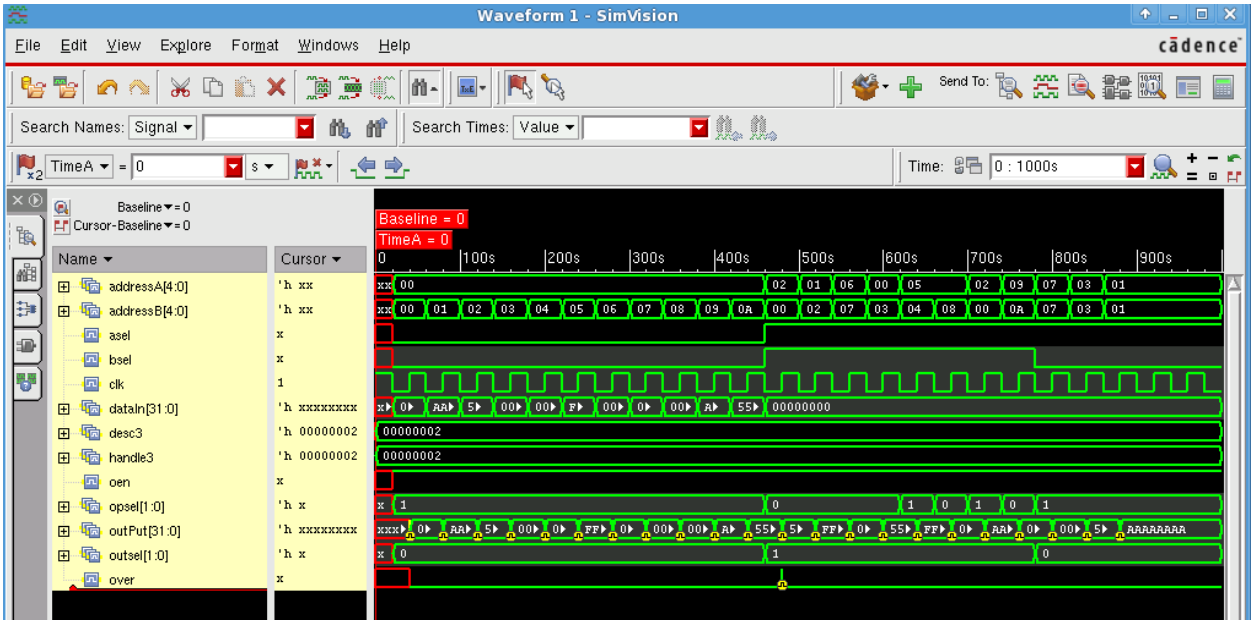
```
---------------------------------------------------------------------------
---------
*       Power Distribution Summary:
*               Highest Average Power:              clk__L4_I32 (INVX8):
 0.009632
*               Highest Leakage Power: mb/ram/mer2/ll/me31/qout_reg (DFFPOSX1):
 5.498e-05
*               Total Cap:      1.80592e-10 F
*               Total instances in design: 14466
*               Total instances in design with no power:     0
*                Total instances in design with no activty:     0

*               Total Fillers and Decap:     0
---------------------------------------------------------------------------
```



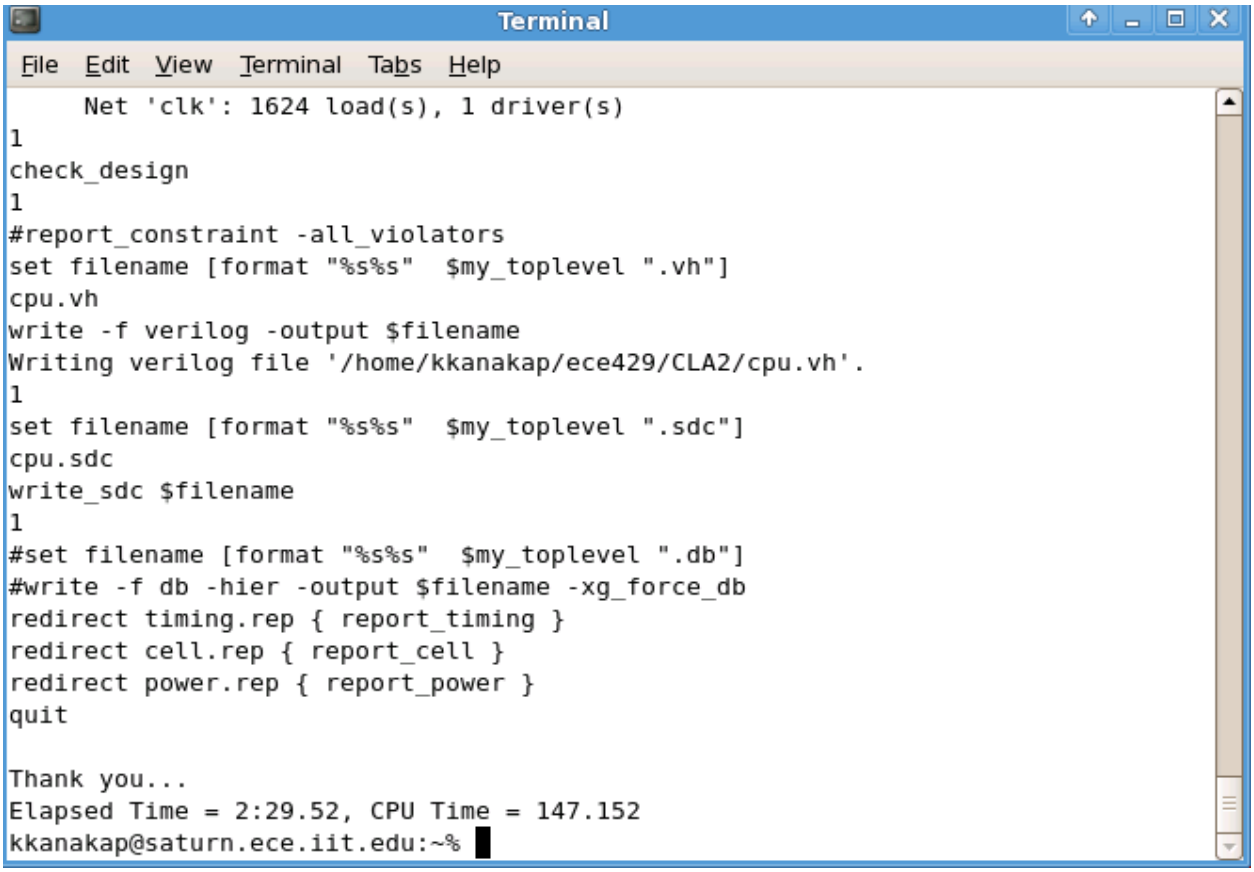
2. CARRY LOOKAHEAD ADDER:

RTL Simulation:

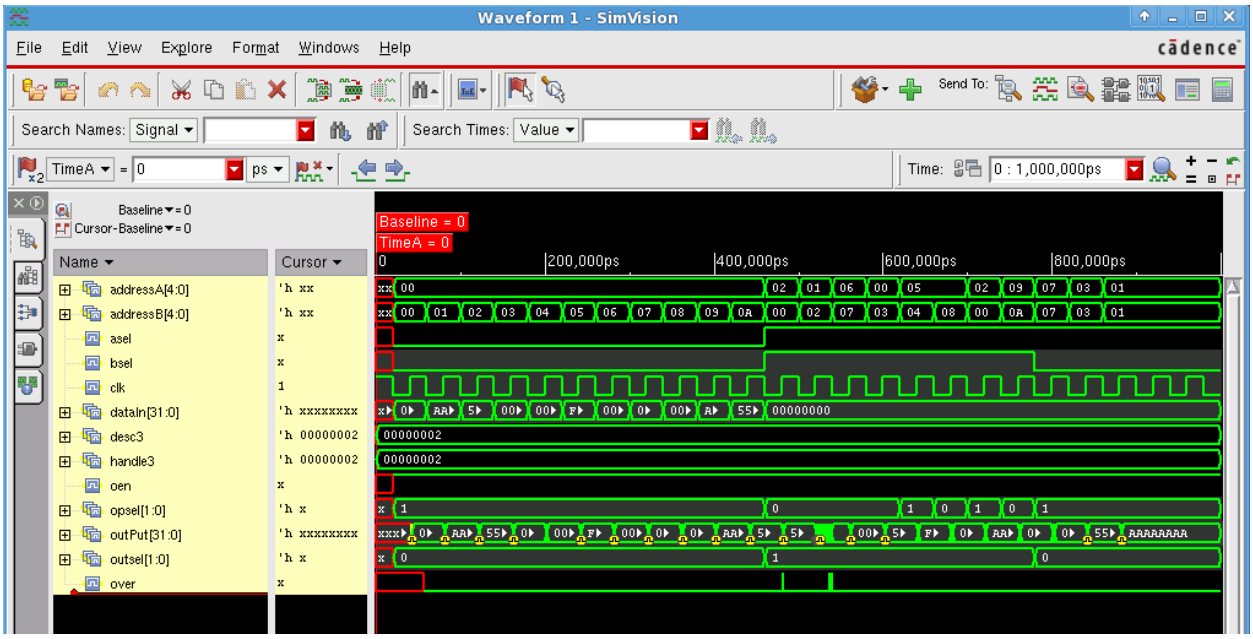




Post Synthesis Simulation:



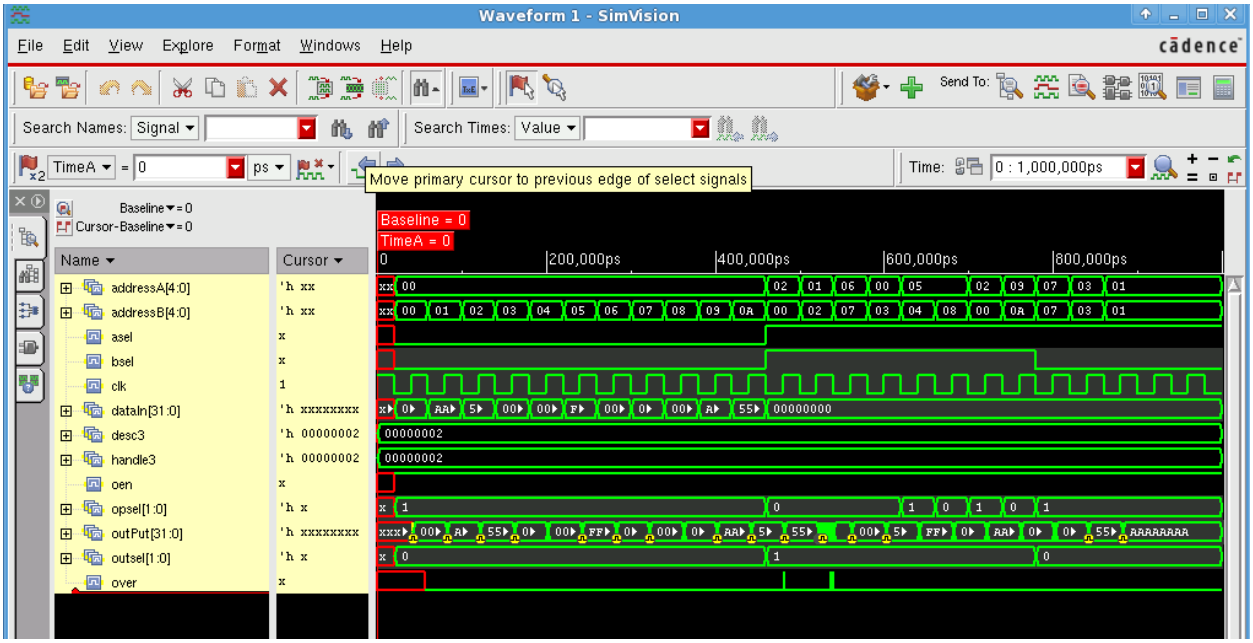
Simvision output:



Post-P&R Simulation:

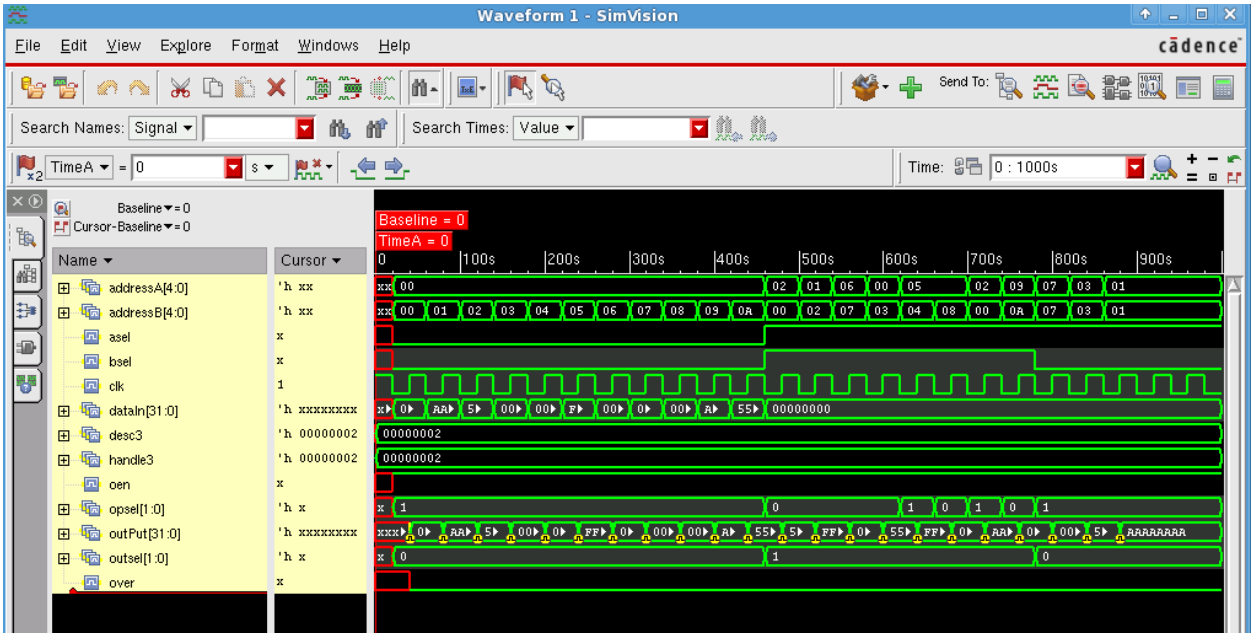
```
CPU time: 0.2 secs to compile + 0.3 secs to link + 0.5 secs in simulation
End of Tool:    VERILOG-XL      08.20.001-p   Dec  5, 2014  01:17:23
encounter 2> reportGateCount -limit 0
Gate area 2.8158 um^2
[0] cpu Gates=15744 Cells=14446 Area=44333.4 um^2
encounter 3>
```

```
---------
*       Power Distribution Summary:
*               Highest Average Power:              clk__L4_I28 (INVX8):
 0.009625
*               Highest Leakage Power: mb/ram/mer2/ll/me31/qout_reg (DFFPOSX1):
 5.498e-05
*               Total Cap:      1.80425e-10 F
*               Total instances in design: 14446
*               Total instances in design with no power:     0
*                Total instances in design with no activty:     0

*               Total Fillers and Decap:     0
--------------------------------------------------------------------------------
```

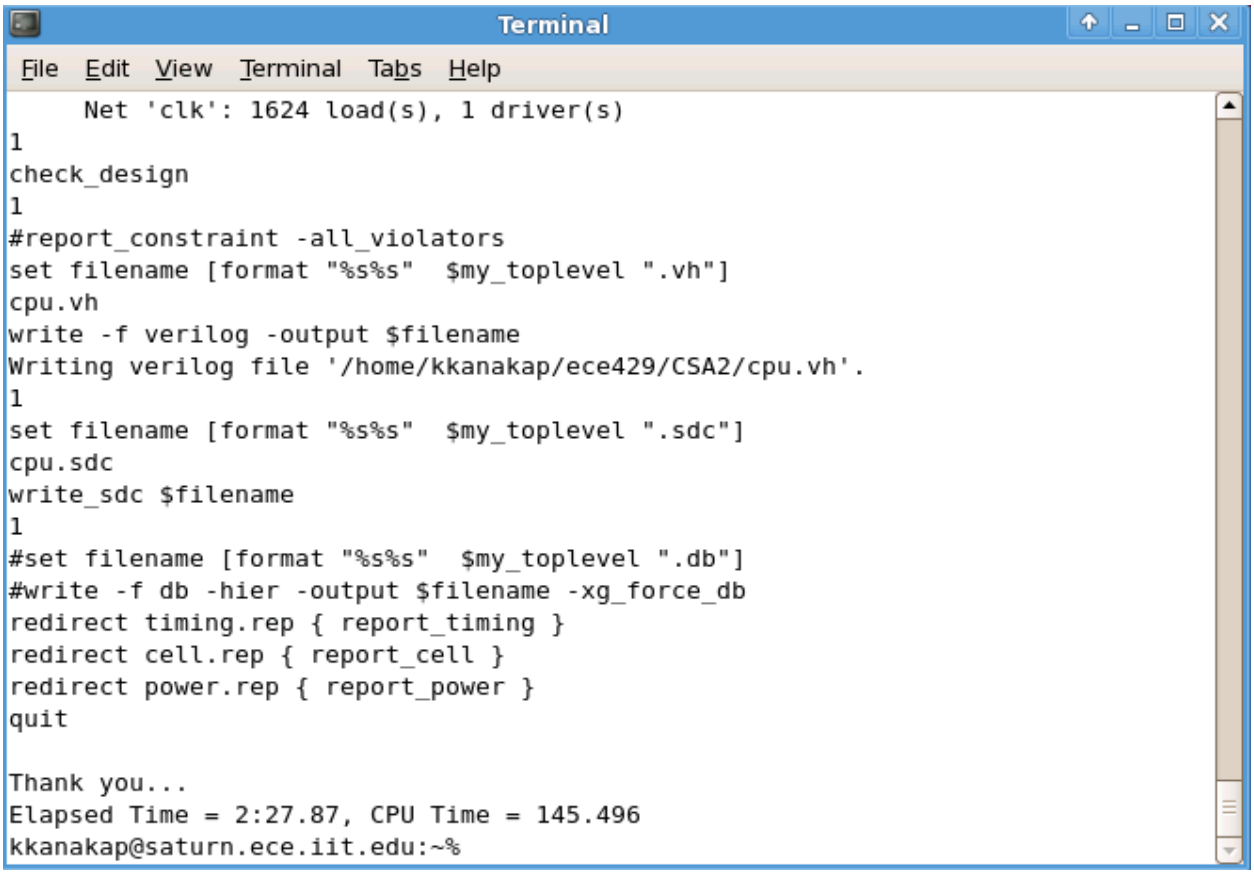


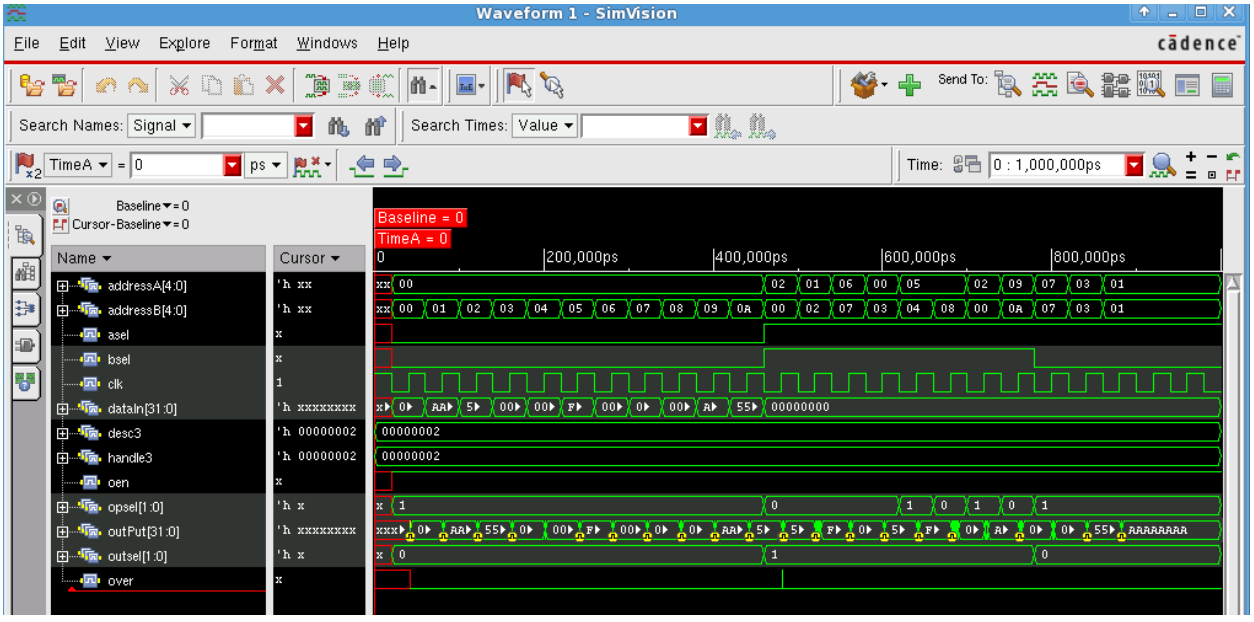
3. CARRY SKIP ADDER:

RTL Simulation:



Post Synthesis Simulation:

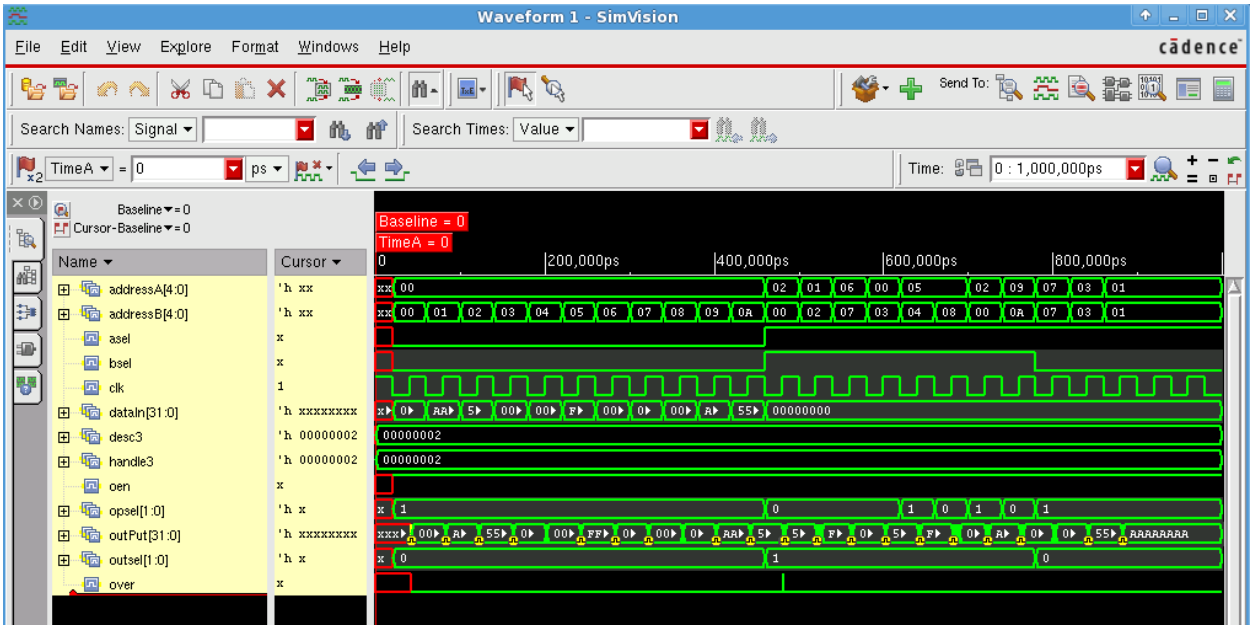




Post-P&R Simulation:

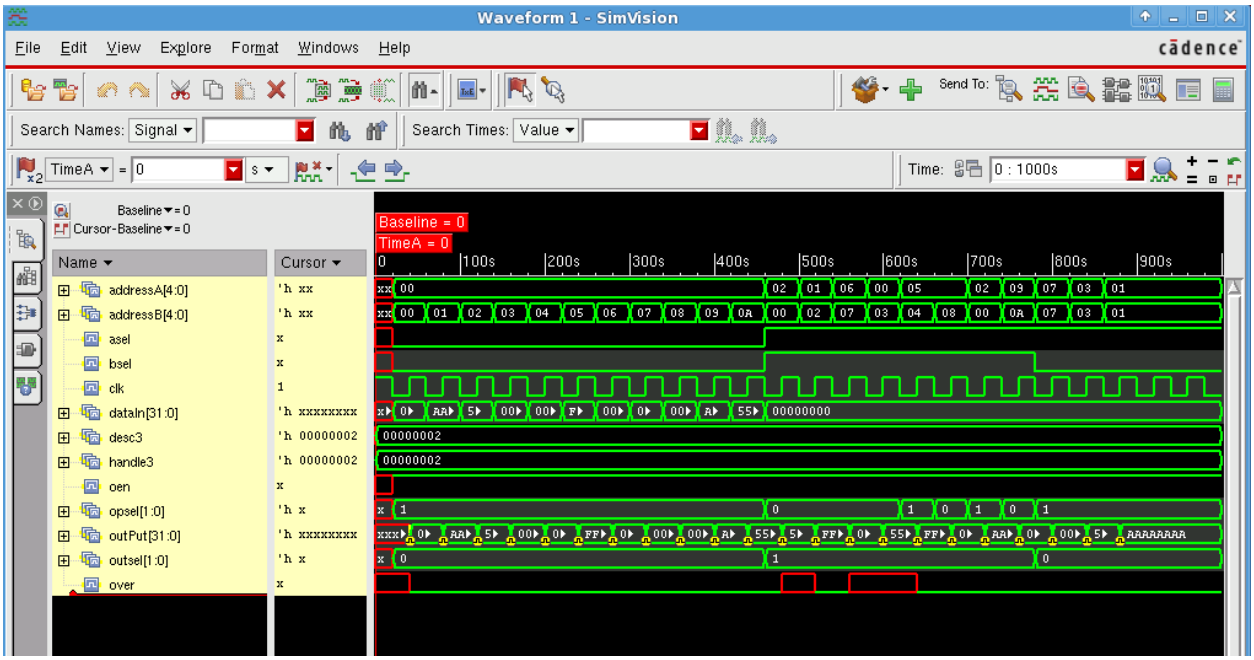
```
CPU time: 0.2 secs to compile + 0.3 secs to link + 0.5 secs in simulation
End of Tool:     VERILOG-XL      08.20.001-p   Dec  5, 2014  01:46:10
encounter 2> reportGateCount -limit 0
Gate area 2.8158 um^2
[0] cpu Gates=15848 Cells=14565 Area=44626.2 um^2
encounter 3>
```

```
--------------------------------------------------------------------------------
---------
*       Power Distribution Summary:
*               Highest Average Power:                clk__L4_I7 (INVX8):
 0.009491
*               Highest Leakage Power: mb/ram/mer2/ll/me31/qout_reg (DFFPOSX1):
 5.498e-05
*               Total Cap:      1.81192e-10 F
*               Total instances in design: 14565
*               Total instances in design with no power:     0
*                Total instances in design with no activty:     0

*               Total Fillers and Decap:     0
--------------------------------------------------------------------------------
```

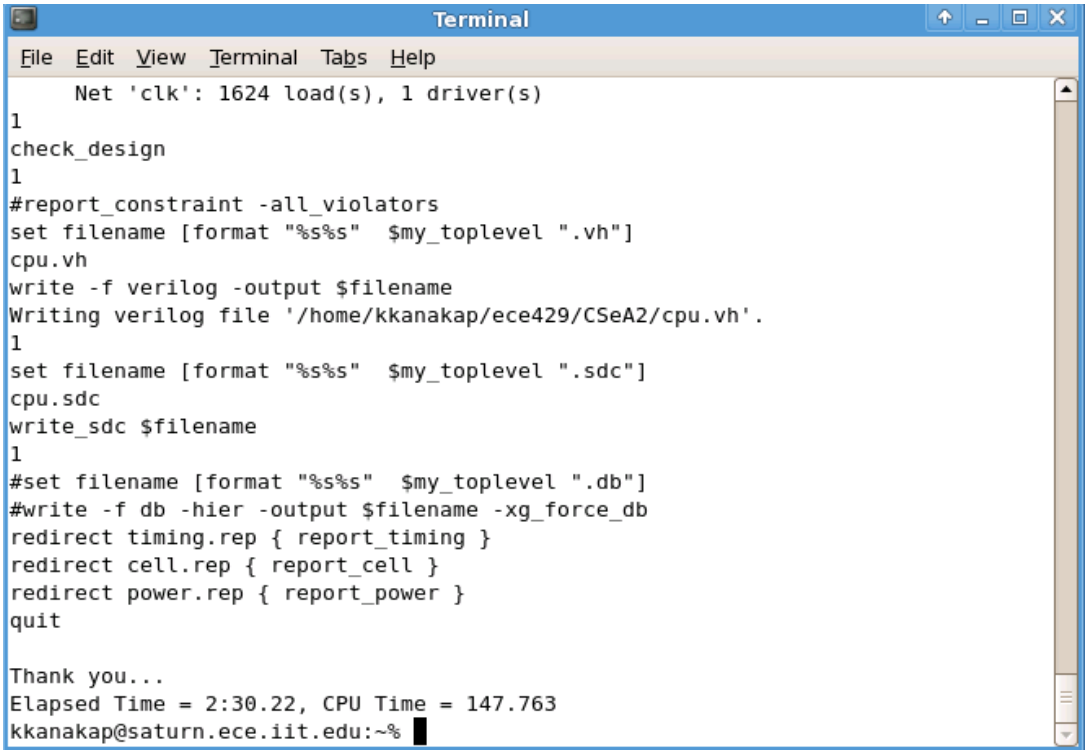


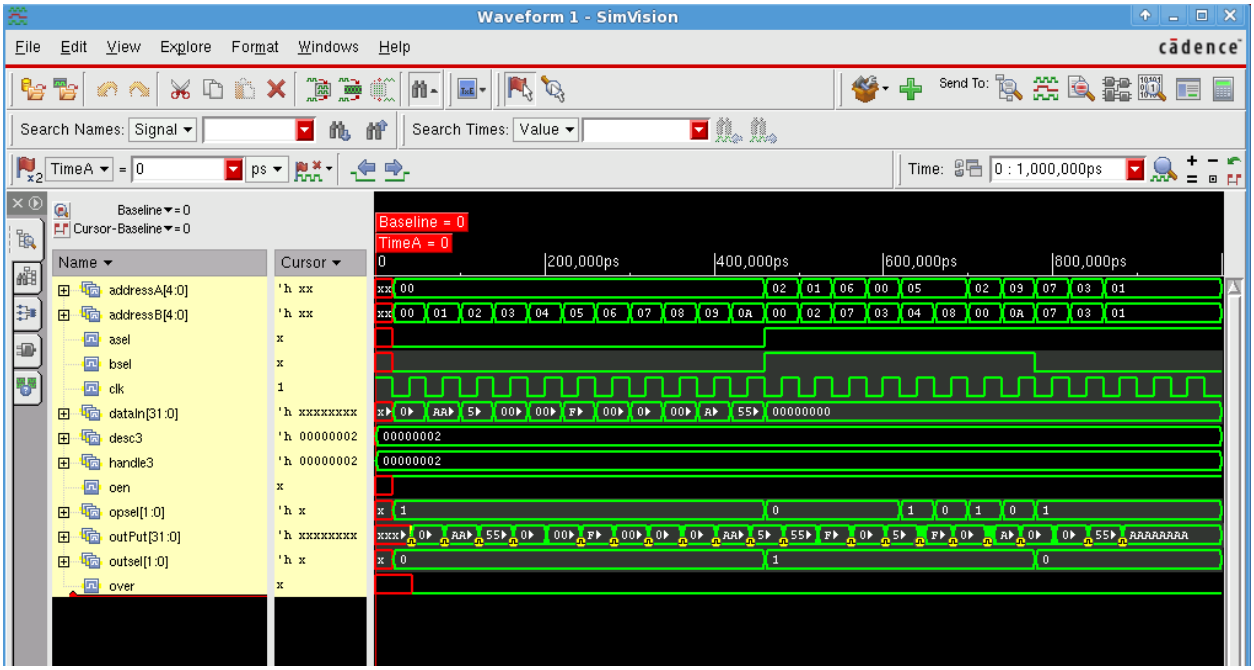
4. CARRY SELECT ADDER:

RTL Simulation:



Post Synthesis Simulation:

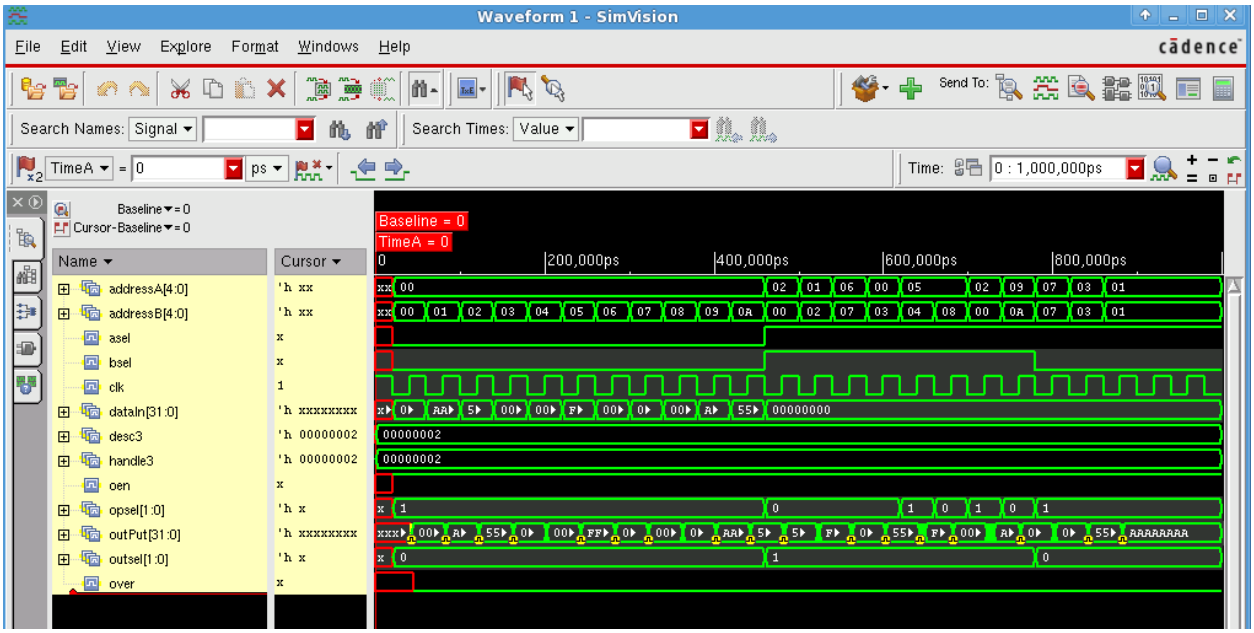




Post-P&R Simulation:

```
CPU time: 0.2 secs to compile + 0.3 secs to link + 0.5 secs in simulation
End of Tool:    VERILOG-XL      08.20.001-p   Dec  5, 2014  02:11:36
encounter 2> reportGateCount -limit 0
Gate area 2.8158 um^2
[0] cpu Gates=15987 Cells=14717 Area=45017.1 um^2
encounter 3>
```

```
---------------------------------------------------------------------------
---------
*       Power Distribution Summary:
*               Highest Average Power:              clk__L4_I19 (INVX8):
 0.009642
*               Highest Leakage Power: mb/ram/mer2/ll/me31/qout_reg (DFFPOSX1):
 5.498e-05
*               Total Cap:      1.80167e-10 F
*               Total instances in design: 14717
*               Total instances in design with no power:     0
*                Total instances in design with no activty:     0

*               Total Fillers and Decap:     0
---------------------------------------------------------------------------
```



		CRA	CLA	CSA	CSeA
Path Delay for Each Operation (Post-Synthesis Gate-Level Delay)	5555_5555 + 5	2.20ns	2.18ns	2.15ns	1.20ns
	AAAA_AAAA + 5555_5555	2.88ns	2.87ns	2.33ns	1.65ns
	0000_000C8 + 0000_012C	2.43ns	2.75ns	2.22ns	1.74ns
	5 + 0000_000A	2.70ns	2.56ns	2.15ns	1.70ns
	FFFF_FFFF – 0000_0001	2.68ns	2.60ns	2.24ns	2.10ns
	FFFF_FFFF + 0000_0001	2.54ns	2.30ns	2.26ns	1.95ns
	5555_5555 – 5	2.15ns	2.00ns	1.82ns	1.60ns
	AAAA_AAAB + 5555_5555	2.86ns	2.27ns	1.79ns	1.62ns

CONCLUSION:

The Verilog codes and the testbench for CPU was written and all the synthesis as mentioned in the manual was successfully implemented and verified.

CASE STUDY-2

32-BIT CPU DESIGN WITH NEW ALU ARCHITECTURE

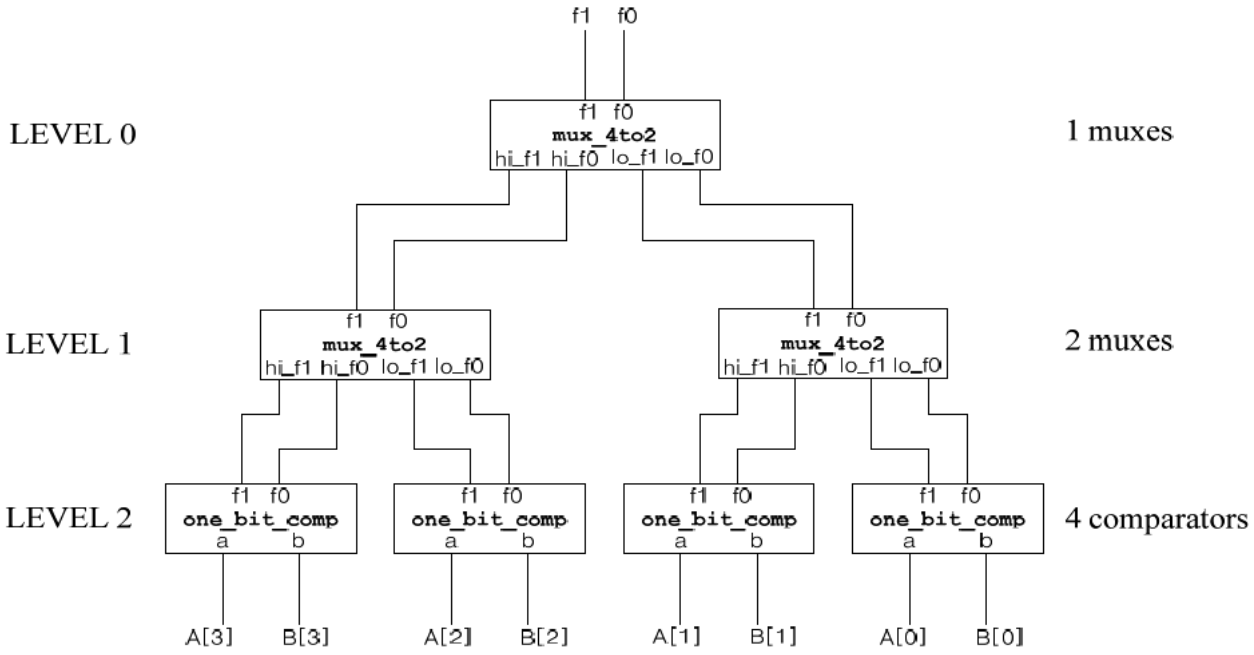
INTRODUCTION:

The main objective of this case study is to add a 32-bit comparator block into the ALU designed in case study-1. Then we perform logical synthesis and physical synthesis using the testbench.

The function of a 32-bit comparator in Verilog is as shown in below table.

	f1	f0
A > B	0	1
A < B	0	0
A = B	1	0

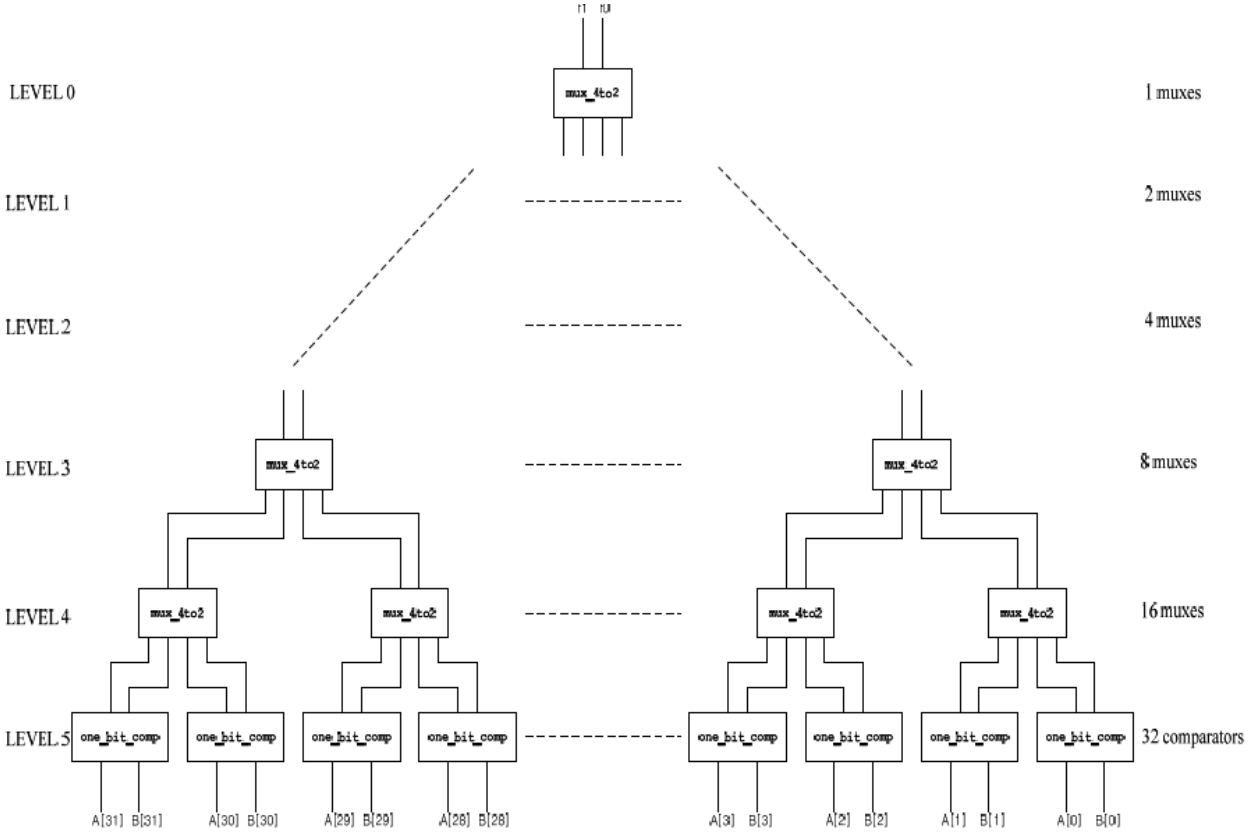
STRUCTURE OF 4-BIT COMPARATOR:



The structure is designed in a tree structure. As shown in the above figure, at the bottom level there are four 1-bit comparators which are used to compare the corresponding bit in A and B. The outputs f1 and f0 are same as the table shown for 32-bit comparator.

From the above figure we can notice that the number of mux_4to2 is 3 which is equal to 4 – 1 and the level of the tree is 3 which is equal to log2(4) + 1. More generally, if two N-bit (N is the power of 2) unsigned integers are compared, then the tree comparator will be (log2(N) + 1) levels, and it will consists of N -1 mux_4to2 and N one_bit_comp.

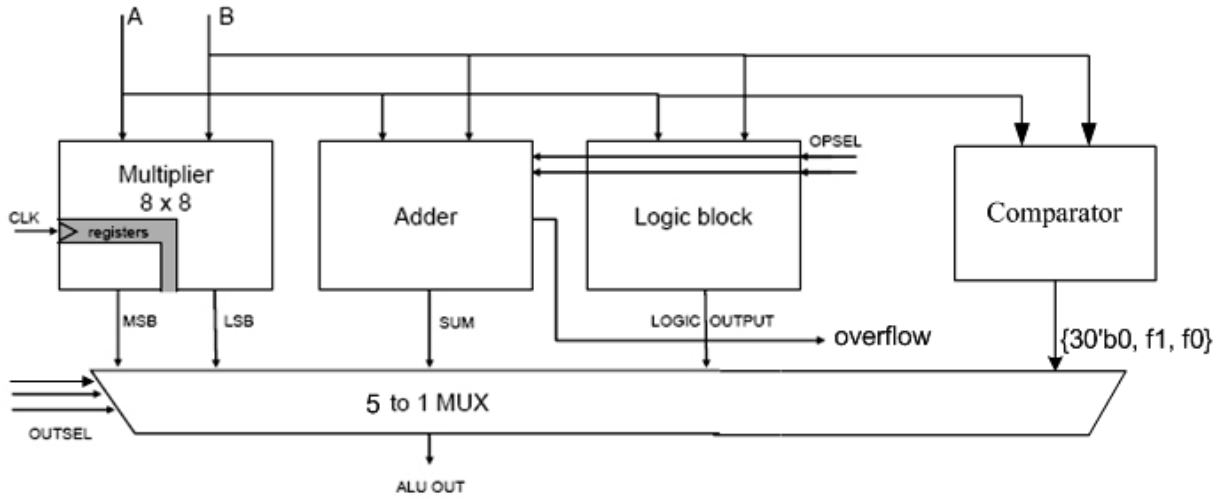
STRUCTURE VIEW OF A 32-BIT COMPARATOR:



The structure of a 32-bit comparator is as shown in above figure.

THE NEW ALU DESIGN:

The ALU design after adding the 32-bit comparator is as shown below.



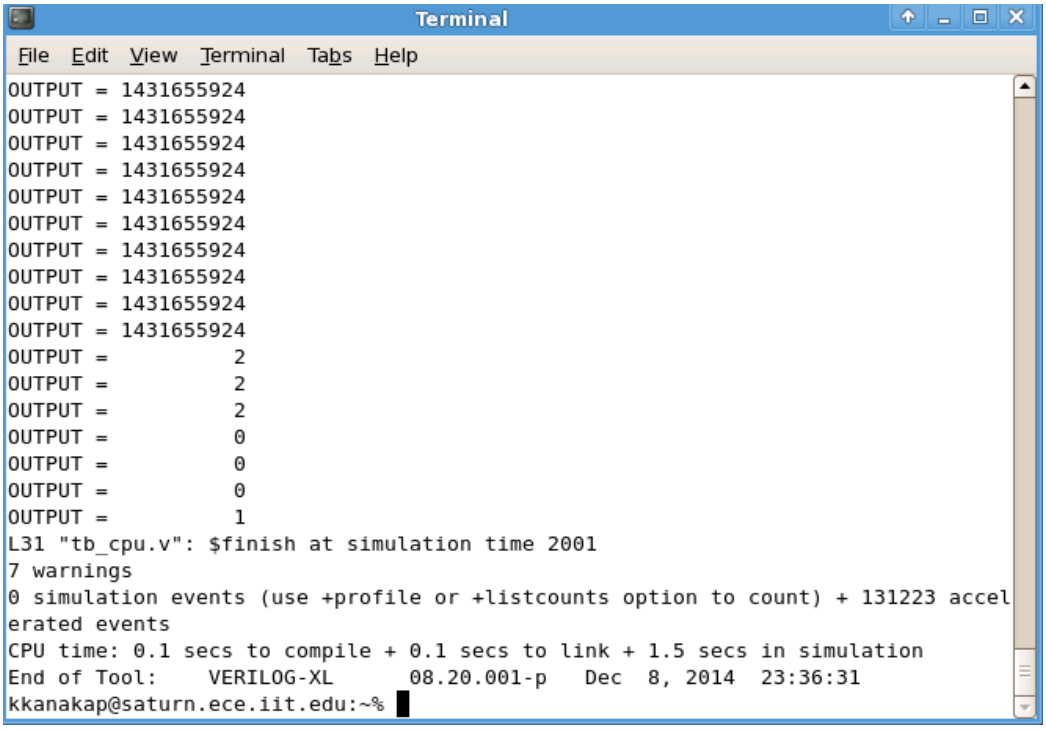
CODE FOR COMPARATOR (cpu_comp.v):

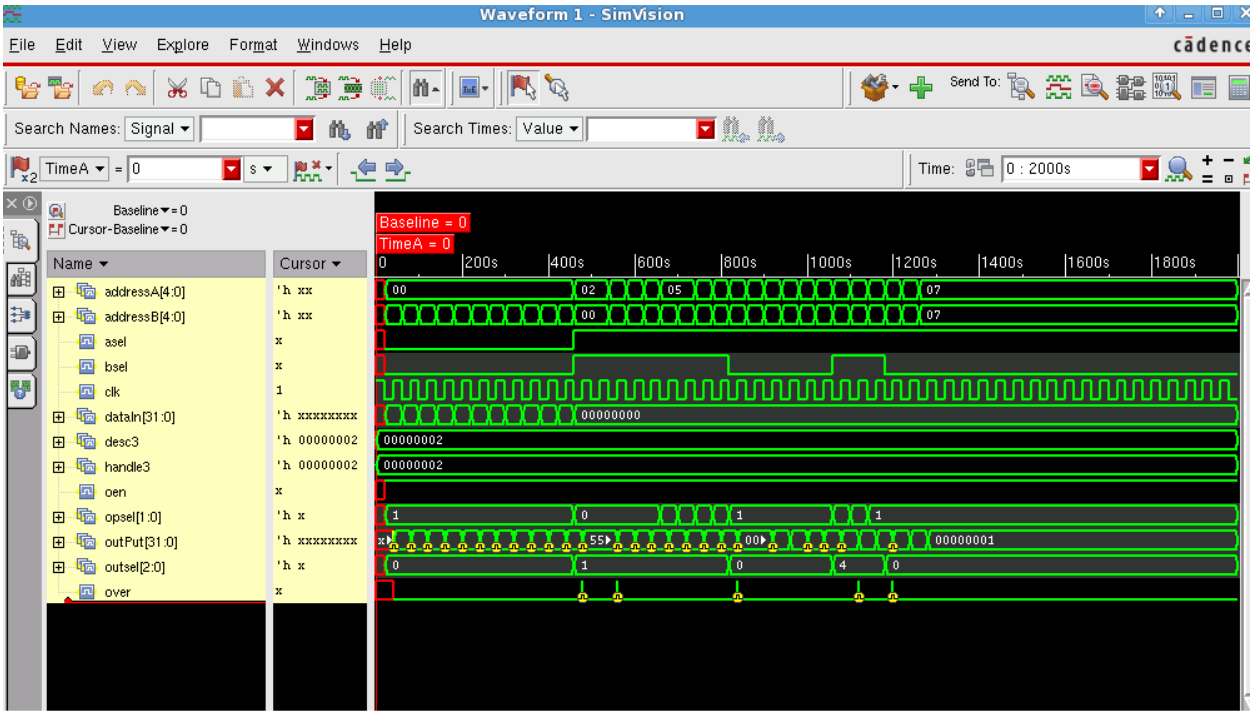
```
//one bit comparator
module one_bit_comp(x, y, f1, f0);
input x, y;
output f1, f0;

assign f0 = {(x | y) & (x | ~y) & (~x | ~y)};    // Assigning f0 and
assign f1 = {(~x | y) & (x | ~y)};               // Assigning f1

endmodule

//mux to select the f1 f0 outputs
module mux_4to2(hi_f1, hi_f0, lo_f1, lo_f0, f1, f0);

input hi_f1, hi_f0, lo_f1, lo_f0;
output f1, f0;

assign f1 =  {(hi_f1 & lo_f1)};
assign f0 = {((~hi_f1) & hi_f0) | (hi_f1 & lo_f0)};

endmodule

//32-bit tree comparator
module tree_comp(A, B, f1, f0);
input [31 : 0] A, B;
output f1, f0;
```

```
wire [31 : 0] f1_L5, f0_L5;
wire [15 : 0] f1_L4, f0_L4;
wire [7 : 0] f1_L3, f0_L3;
wire [3 : 0] f1_L2, f0_L2;
wire [1 : 0] f1_L1, f0_L1;


//Level 5: 32 one_bit_comp go here
one_bit_comp bc0(A[0], B[0], f1_L5[0], f0_L5[0]);
one_bit_comp bc1(A[1], B[1], f1_L5[1], f0_L5[1]);
one_bit_comp bc2(A[2], B[2], f1_L5[2], f0_L5[2]);
one_bit_comp bc3(A[3], B[3], f1_L5[3], f0_L5[3]);
one_bit_comp bc4(A[4], B[4], f1_L5[4], f0_L5[4]);
one_bit_comp bc5(A[5], B[5], f1_L5[5], f0_L5[5]);
one_bit_comp bc6(A[6], B[6], f1_L5[6], f0_L5[6]);
one_bit_comp bc7(A[7], B[7], f1_L5[7], f0_L5[7]);
one_bit_comp bc8(A[8], B[8], f1_L5[8], f0_L5[8]);
one_bit_comp bc9(A[9], B[9], f1_L5[9], f0_L5[9]);
one_bit_comp bc10(A[10], B[10], f1_L5[10], f0_L5[10]);
one_bit_comp bc11(A[11], B[11], f1_L5[11], f0_L5[11]);
one_bit_comp bc12(A[12], B[12], f1_L5[12], f0_L5[12]);
one_bit_comp bc13(A[13], B[13], f1_L5[13], f0_L5[13]);
one_bit_comp bc14(A[14], B[14], f1_L5[14], f0_L5[14]);
one_bit_comp bc15(A[15], B[15], f1_L5[15], f0_L5[15]);
one_bit_comp bc16(A[16], B[16], f1_L5[16], f0_L5[16]);
one_bit_comp bc17(A[17], B[17], f1_L5[17], f0_L5[17]);
one_bit_comp bc18(A[18], B[18], f1_L5[18], f0_L5[18]);
one_bit_comp bc19(A[19], B[19], f1_L5[19], f0_L5[19]);
one_bit_comp bc20(A[20], B[20], f1_L5[20], f0_L5[20]);
one_bit_comp bc21(A[21], B[21], f1_L5[21], f0_L5[21]);
one_bit_comp bc22(A[22], B[22], f1_L5[22], f0_L5[22]);
one_bit_comp bc23(A[23], B[23], f1_L5[23], f0_L5[23]);
one_bit_comp bc24(A[24], B[24], f1_L5[24], f0_L5[24]);
one_bit_comp bc25(A[25], B[25], f1_L5[25], f0_L5[25]);
one_bit_comp bc26(A[26], B[26], f1_L5[26], f0_L5[26]);
one_bit_comp bc27(A[27], B[27], f1_L5[27], f0_L5[27]);
one_bit_comp bc28(A[28], B[28], f1_L5[28], f0_L5[28]);
one_bit_comp bc29(A[29], B[29], f1_L5[29], f0_L5[29]);
one_bit_comp bc30(A[30], B[30], f1_L5[30], f0_L5[30]);
one_bit_comp bc31(A[31], B[31], f1_L5[31], f0_L5[31]);


//Level 4: 16 mux_4to2 go here
mux_4to2 M0(f1_L5[1], f0_L5[1], f1_L5[0], f0_L5[0], f1_L4[0], f0_L4[0]);
mux_4to2 M1(f1_L5[3], f0_L5[3], f1_L5[2], f0_L5[2], f1_L4[1], f0_L4[1]);
```

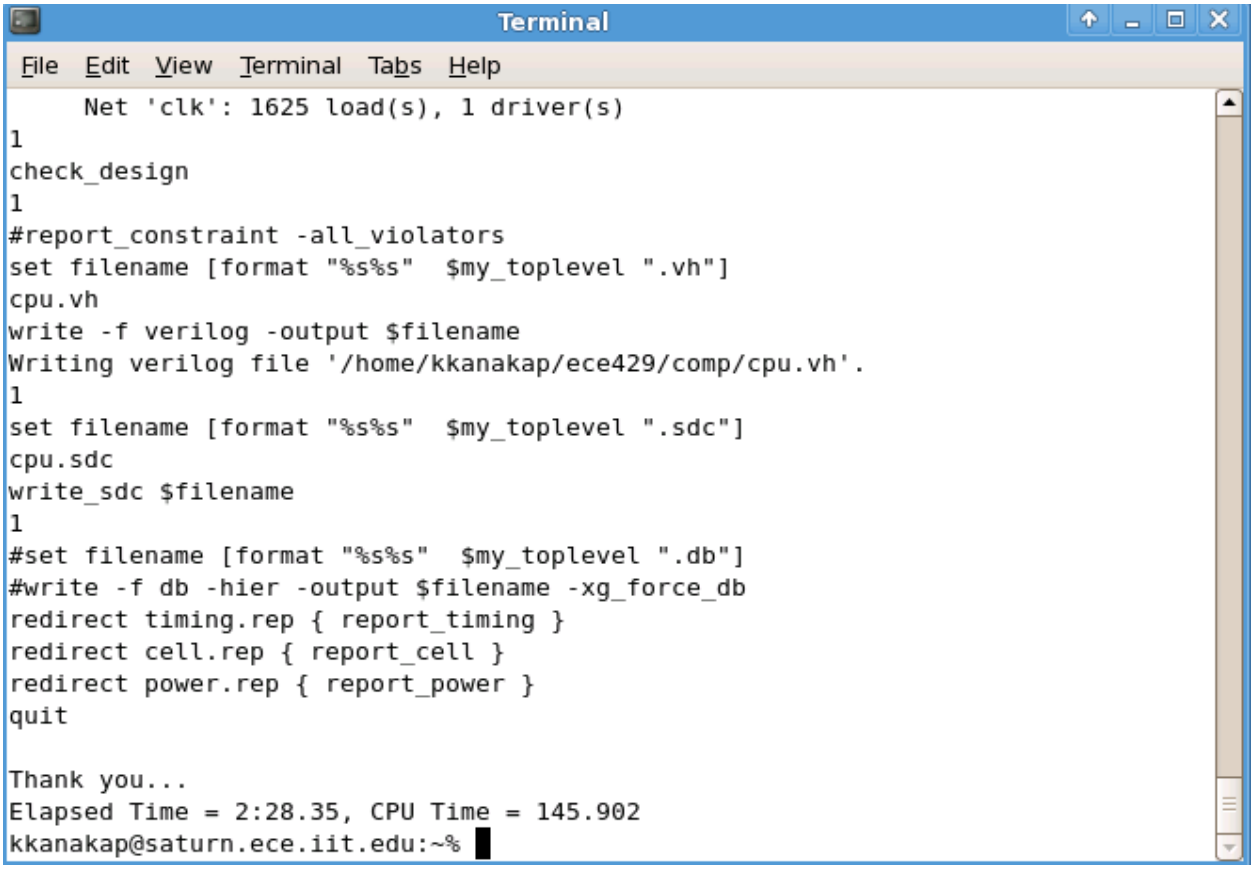
```
mux_4to2 M2(f1_L5[5], f0_L5[5], f1_L5[4], f0_L5[4], f1_L4[2], f0_L4[2]);
mux_4to2 M3(f1_L5[7], f0_L5[7], f1_L5[6], f0_L5[6], f1_L4[3], f0_L4[3]);
mux_4to2 M4(f1_L5[9], f0_L5[9], f1_L5[8], f0_L5[8], f1_L4[4], f0_L4[4]);
mux_4to2 M5(f1_L5[11], f0_L5[11], f1_L5[10], f0_L5[10], f1_L4[5], f0_L4[5]);
mux_4to2 M6(f1_L5[13], f0_L5[13], f1_L5[12], f0_L5[12], f1_L4[6], f0_L4[6]);
mux_4to2 M7(f1_L5[15], f0_L5[15], f1_L5[14], f0_L5[14], f1_L4[7], f0_L4[7]);
mux_4to2 M8(f1_L5[17], f0_L5[17], f1_L5[16], f0_L5[16], f1_L4[8], f0_L4[8]);
mux_4to2 M9(f1_L5[19], f0_L5[19], f1_L5[18], f0_L5[18], f1_L4[9], f0_L4[9]);
mux_4to2 M10(f1_L5[21], f0_L5[21], f1_L5[20], f0_L5[20], f1_L4[10], f0_L4[10]);
mux_4to2 M11(f1_L5[23], f0_L5[23], f1_L5[22], f0_L5[22], f1_L4[11], f0_L4[11]);
mux_4to2 M12(f1_L5[25], f0_L5[25], f1_L5[24], f0_L5[24], f1_L4[12], f0_L4[12]);
mux_4to2 M13(f1_L5[27], f0_L5[27], f1_L5[26], f0_L5[26], f1_L4[13], f0_L4[13]);
mux_4to2 M14(f1_L5[29], f0_L5[29], f1_L5[28], f0_L5[28], f1_L4[14], f0_L4[14]);
mux_4to2 M15(f1_L5[31], f0_L5[31], f1_L5[30], f0_L5[30], f1_L4[15], f0_L4[15]);


//Level 3: 8 mux_4to2 go here
mux_4to2 M16(f1_L4[1], f0_L4[1], f1_L4[0], f0_L4[0], f1_L3[0], f0_L3[0]);
mux_4to2 M17(f1_L4[3], f0_L4[3], f1_L4[2], f0_L4[2], f1_L3[1], f0_L3[1]);
mux_4to2 M18(f1_L4[5], f0_L4[5], f1_L4[4], f0_L4[4], f1_L3[2], f0_L3[2]);
mux_4to2 M19(f1_L4[7], f0_L4[7], f1_L4[6], f0_L4[6], f1_L3[3], f0_L3[3]);
mux_4to2 M20(f1_L4[9], f0_L4[9], f1_L4[8], f0_L4[8], f1_L3[4], f0_L3[4]);
mux_4to2 M21(f1_L4[11], f0_L4[11], f1_L4[10], f0_L4[10], f1_L3[5], f0_L3[5]);
mux_4to2 M22(f1_L4[13], f0_L4[13], f1_L4[12], f0_L4[12], f1_L3[6], f0_L3[6]);
mux_4to2 M23(f1_L4[15], f0_L4[15], f1_L4[14], f0_L4[14], f1_L3[7], f0_L3[7]);


//Level 2: 4 mux_4to2 go here
mux_4to2 M25(f1_L3[1], f0_L3[1], f1_L3[0], f0_L3[0], f1_L2[0], f0_L2[0]);
mux_4to2 M26(f1_L3[3], f0_L3[3], f1_L3[2], f0_L3[3], f1_L2[1], f0_L2[1]);
mux_4to2 M27(f1_L3[5], f0_L3[5], f1_L3[4], f0_L3[4], f1_L2[2], f0_L2[2]);
mux_4to2 M28(f1_L3[7], f0_L3[7], f1_L3[6], f0_L3[6], f1_L2[3], f0_L2[3]);

//Level 1: 2 mux_4to2 go here
mux_4to2 M29(f1_L2[1], f0_L2[1], f1_L2[0], f0_L2[0], f1_L1[0], f0_L1[0]);
mux_4to2 M30(f1_L2[3], f0_L2[3], f1_L2[2], f0_L2[2], f1_L1[1], f0_L1[1]);


//Level 0: 1 mux_4to2 goes here
mux_4to2 M31(f1_L1[1], f0_L1[1], f1_L1[0], f0_L1[0], f1, f0);

endmodule
```

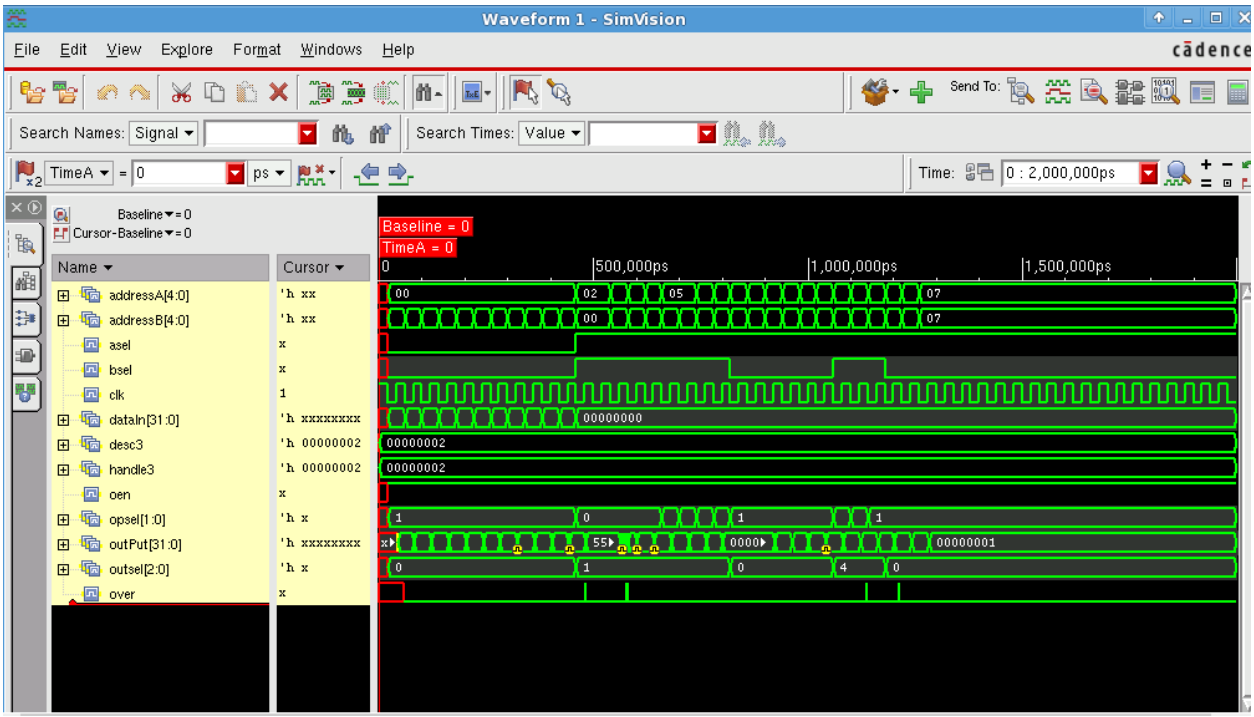
RTL Simulation:





Post Synthesis Simulation:

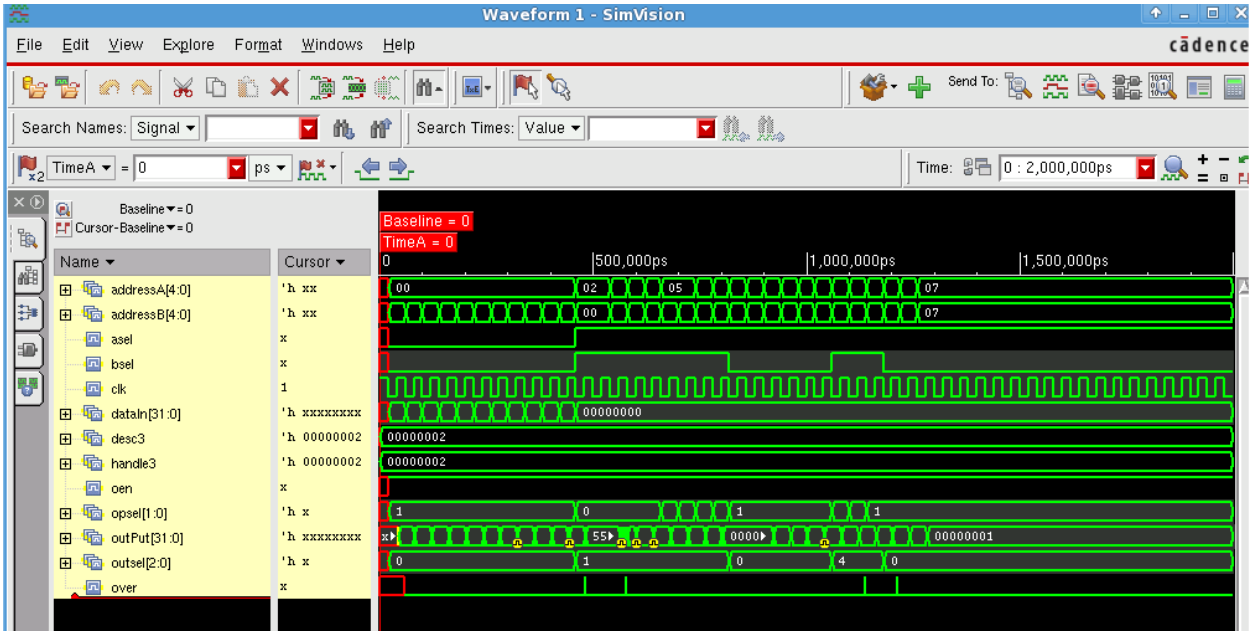




Post-P&R Simulation:

```
CPU time: 0.2 secs to compile + 0.3 secs to link + 0.8 secs in simulation
End of Tool:     VERILOG-XL      08.20.001-p    Dec  9, 2014  00:01:25
encounter 2> reportGateCount -limit 0
Gate area 2.8158 um^2
[0] cpu Gates=15989 Cells=14707 Area=45022.8 um^2
encounter 3>
```

```
--------------------------------------------------------------------------------
---------
*       Power Distribution Summary:
*               Highest Average Power:                clk__L4_I9 (INVX8):
 0.009407
*               Highest Leakage Power: mb/ram/mer2/ll/me31/qout_reg (DFFPOSX1):
 5.498e-05
*               Total Cap:      1.81018e-10 F
*               Total instances in design: 14707
*               Total instances in design with no power:     0
*                Total instances in design with no activty:     0

*               Total Fillers and Decap:     0
--------------------------------------------------------------------------------
```



Maximum clock frequency = 115.2 KHz

CONCLUSION:

The Verilog codes and the testbench for CPU was written and all the synthesis as mentioned in the manual was successfully implemented and verified.