

ECE 429 Case Study:
32-bit Pipelined CPU design with New ALU Architecture

Alexander Lukens

Illinois Institute of Technology

November 25, 2020

Acknowledgment: I acknowledge all of the work (including figures and codes) belongs to me and/or persons
who are referenced.

Signature: Alexander Lukens

Contents

1 Case Study - ALU Adder Design	3
1.1 Theory	3
1.2 Carry-Lookahead Adder Implementation	5
1.2.1 RTL Simulation	5
1.2.2 Logic Synthesis Timing & Cell Reports	6
1.2.3 Post Logic Synthesis Simulation	7
1.2.4 Physical Synthesis Timing Report	8
1.2.5 Post Physical Synthesis Simulation	10
1.2.6 RTL Simulation with New Testbench	10
1.3 Carry-Ripple Adder Implementation	12
1.3.1 RTL Simulation	12
1.3.2 Logic Synthesis Timing & Cell Reports	14
1.3.3 Post Logic Synthesis Simulation	16
1.3.4 Physical Synthesis Timing Report	17
1.3.5 Post Physical Synthesis Simulation	19
1.3.6 RTL Simulation with New Testbench	20
1.4 Carry-Skip Adder Implementation	22
1.4.1 RTL Simulation	22
1.4.2 Logic Synthesis Timing & Cell Reports	24
1.4.3 Post Logic Synthesis Simulation	25
1.4.4 Physical Synthesis Timing Report	26
1.4.5 Post Physical Synthesis Simulation	27
1.4.6 RTL Simulation with New Testbench	28
1.5 Carry-Select Adder Implementation	30
1.5.1 RTL Simulation	30
1.5.2 Logic Synthesis Timing & Cell Reports	31
1.5.3 Post Logic Synthesis Simulation	32
1.5.4 Physical Synthesis Timing Report	33
1.5.5 Post Physical Synthesis Simulation	34
1.5.6 RTL Simulation with New Testbench	35
1.6 Deliverables	36
1.7 Conclusions	36
2 Case Study - ALU Comparator Design	37
2.1 Theory	37
2.2 Single Bit Comparator Design	37
2.3 4-to-2 Multiplexer Design	38
2.4 32-bit Comparator Design	38
2.5 New Testbench for Verifying Functionality of Comparator	39
2.6 RTL Simulation of CPU with Comparator	40
2.7 Logic Synthesis Timing & Cell Reports	42
2.8 Post Logic Synthesis Simulation	44
2.9 Physical Synthesis Timing Report	45
2.10 Post Physical Synthesis Simulation	46
2.11 Conclusions	46
2.12 Resources	47

1 Case Study - ALU Adder Design

In this case study, students will implement a new ALU design for a 32-bit pipelined CPU using Verilog. Specifically, a variety of adders (Carry-Lookahead, Carry-Ripple, Carry-Skip, and Carry-Select) will be implemented, and their impact on the performance of the CPU will be analyzed. A new testbench file will be created to ensure that the adders function as intended, and the results of the testbench simulation will be tabulated to ensure correctness. Students will also utilize the ASIC Design Flow to perform logic synthesis and physical synthesis on the CPU design, and analyze the impact the Adder designs have on the maximum frequency of the CPU (and therefore performance of the CPU).

1.1 Theory

ASIC Design Flow An important process for automating the creation of digital logic circuits is RTL-to-GDSII flow, or 'ASIC' design flow. Using this flow, a design is first created using the Verilog Hardware Design Language (HDL). In this Verilog file, all of the functionality of the intended design will be created. The functionality of this circuit should first be affirmed using an additional Verilog file as a testbench (used to provide stimuli to the main Verilog circuit). This Verilog file can then be run through a logic synthesis program that converts the RTL level Verilog file originally created into a Verilog file that is completely in terms of a standard cell library. This logic synthesis step prepares the file to be placed and routed, and provides insight into the power, area, and delay of the finalized design. After the completion of logic synthesis, the synthesized design should again be tested using the Verilog testbench in Cadence Formality or Cadence Verilog-XL. This ensures that the synthesized design performs identically to the original Verilog design. If the synthesized design passes all design criteria, the next step is to place and route the design. This adds additional specificity to the design (adds physical interconnect between the standard cells) and provides in-depth timing, area, and power calculations. If the routed design again satisfies all constraints, the design can be imported into Cadence Virtuoso using the produced GDSII file, and the complete layout checked.

Adder Design Choosing an adder implementation is a key metric for determining the performance of a digital logic design. There are several common types of adders:

- **Carry-Ripple Adder** This adder design is created by linking several Full Adders (FA) together in parallel, with the Carry-Out bit of the previous FA connected to the Carry-in bit of the next FA. This connection between full adders results in a large critical path delay from the Carry-in bit to the Carry-out bit, causing the design to be very low performance. This design becomes proportionately slower as the data size increases, causing the adder to be a bottleneck in high performance design.

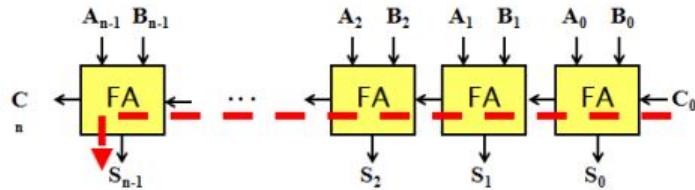


Figure 1: Carry-Ripple Adder Diagram

- **Carry-Lookahead Adder** This adder is more complex than the Carry-Ripple adder design. This design utilizes more complex logic to determine the carry-in for each FA based on the signals of a group of FA's. The larger the group size, the more logic is required to determine the Carry signals, but the faster the design will perform. In this design, the Carry-Lookahead groups are set to be 4 bits each as a compromise between complexity and space.

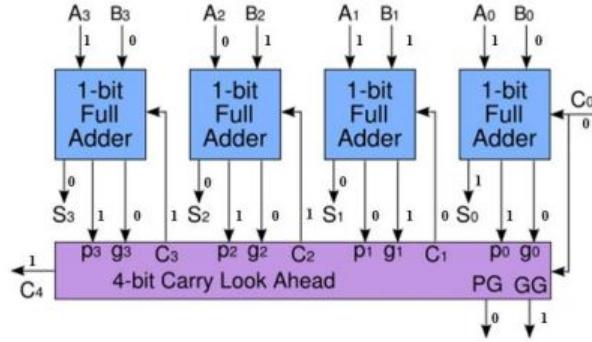


Figure 2: Carry-Lookahead Adder Diagram

- **Carry-Sel ect Adder** This adder design utilizes two Ripple-Carry adders that perform the adder operation in parallel. One adder calculates the sum for Carry-in=1, and the other calculates the sum for Carry-in=0. The output for each bit of the sum is multiplexed together, and the correct sum is chosen by the actual Carry-in bit. By splitting a 32-bit long sum into 4-bit Carry-select adders, the total time required to calculate the entire sum can be reduced.

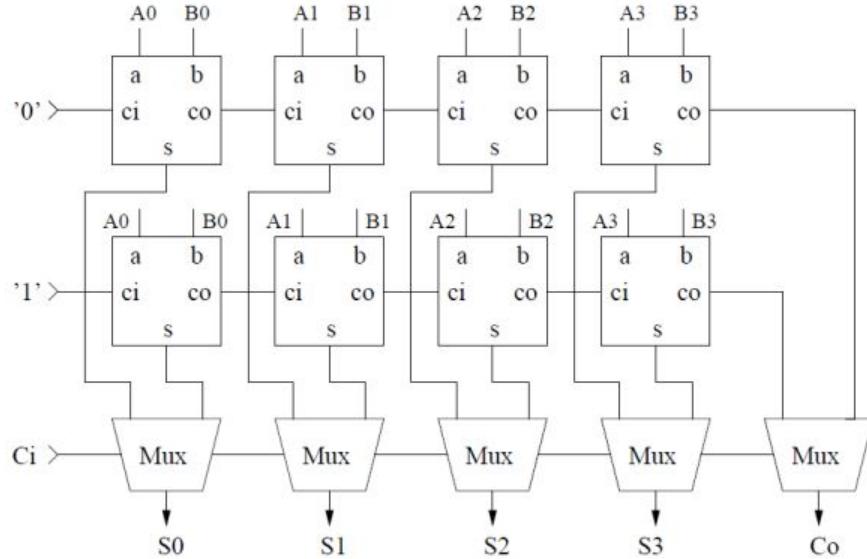


Figure 3: Carry-Select Adder Diagram

- **Carry-Skip Adder** This adder design splits the 32-bit operand into smaller blocks. In each block, a carry ripple adder is utilized to produce sum and Carry-out bits, and an observation about the Carry-out from the block is used to improve efficiency. This observation is that if all "Carry-propagate" signals produced for each bit in the group are =1, the Carry-in bit can be directly propagated to the output. This has the potential to improve performance on some specific input combinations, but is not guaranteed to improve performance on all input combinations. Therefore, the critical delay through this adder is similar to that of the carry-ripple adder.

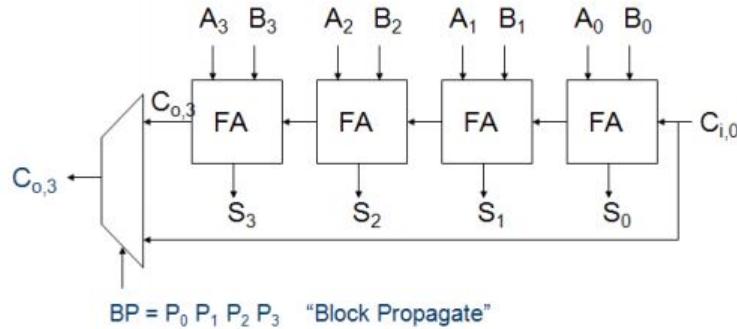


Figure 4: Carry-Skip Adder Diagram

1.2 Carry-Lookahead Adder Implementation

1.2.1 RTL Simulation

```

Terminal
File Edit View Terminal Tabs Help
xmelab: *W,CUVWSP (../cpu_CLA.v,615|24): 1 output port was not connected:
xmelab: (../cpu_CLA.v,575): over

carrylookahead_4 CLA1(.s(s[7:4]), .a(a[7:4]), .b(b[7:4]), .c0(c4), .c4(c8),
.g_4(g[1]), .p_4(p[1]));
|
xmelab: *W,CUVWSP (../cpu_CLA.v,616|24): 1 output port was not connected:
xmelab: (../cpu_CLA.v,575): over

carrylookahead_4 CLA2(.s(s[11:8]), .a(a[11:8]), .b(b[11:8]), .c0(c8), .c4(c12),
.g_4(g[2]), .p_4(p[2]));
|
xmelab: *W,CUVWSP (../cpu_CLA.v,617|24): 1 output port was not connected:
xmelab: (../cpu_CLA.v,575): over

Top level design units:
    stimulus
Building instance overlay tables: ..... Done
Generating native compiled code:
    worklib.stimulus:v <0x19f0146c>
        streams: 13, words: 17664
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
    Instances Unique
    Modules:       6607   33
    Primitives:    7187    6
    Registers:    2166   14
    Scalar wires: 3692   -
    Expanded wires: 1198   41
    Vectored wires: 6   -
    Always blocks: 2156   4
    Initial blocks: 3   3
    Cont. assignments: 11   15
    Pseudo assignments: 43   43
Writing initial simulation snapshot: worklib.stimulus:v
Loading snapshot worklib.stimulus:v ..... Done
xcelium> source /apps/cadence/XCELIUM1803/tools/xcelium/files/xmsimrc
xcelium> run
xmsim: *W,SHMPOPT: Some objects excluded from $shm_probe due to optimizations.
    File: ./tb_cpu.v, line = 28, pos = 11
    Scope: stimulus
    Time: 0 FS + 0

Simulation complete via $finish(1) at time 501 NS + 0
./tb cpu.v:30 #1 $finish;
xcelium> exit
alukens@saturn.ece.iit.edu:~%
```

Figure 5: Carry-Lookahead RTL Simulation

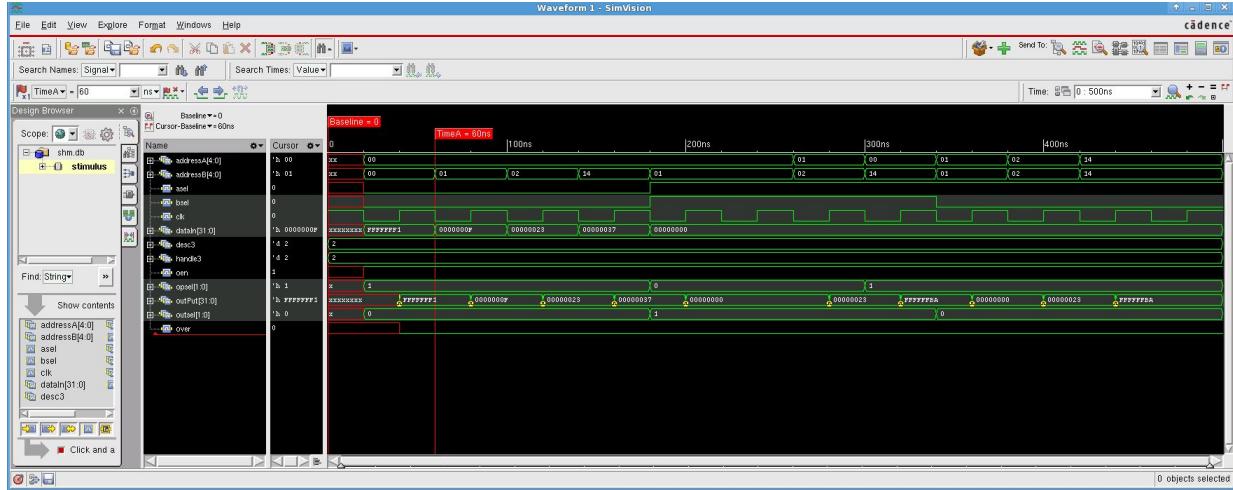


Figure 6: Carry-Lookahead Simvision Data

1.2.2 Logic Synthesis Timing & Cell Reports

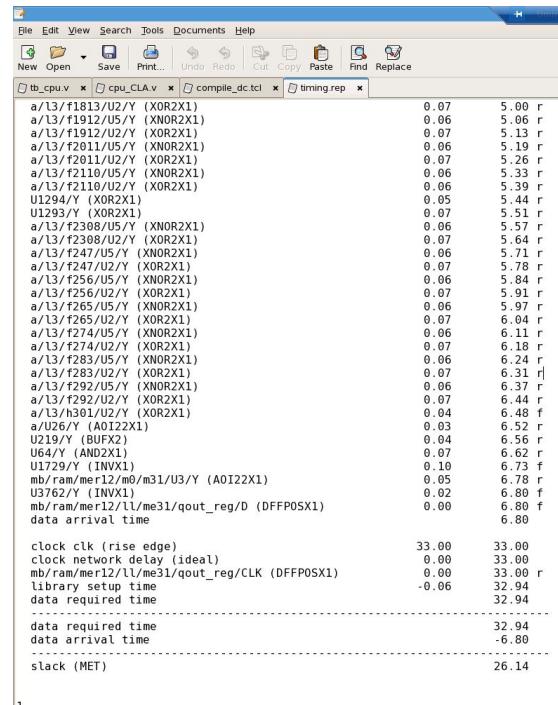


Figure 7: Carry-Lookahead Logic Synthesis Timing Report

15442	mb/ram/me3l/t_b/c19/bl	TBUFX2	gsc145nm	3.754400	n
15443	mb/ram/me3l/t_b/c20/bl	TBUFX2	gsc145nm	3.754400	n
15444	mb/ram/me3l/t_b/c21/bl	TBUFX2	gsc145nm	3.754400	n
15445	mb/ram/me3l/t_b/c22/bl	TBUFX2	gsc145nm	3.754400	n
15446	mb/ram/me3l/t_b/c23/bl	TBUFX2	gsc145nm	3.754400	n
15447	mb/ram/me3l/t_b/c24/bl	TBUFX2	gsc145nm	3.754400	n
15448	mb/ram/me3l/t_b/c25/bl	TBUFX2	gsc145nm	3.754400	n
15449	mb/ram/me3l/t_b/c26/bl	TBUFX2	gsc145nm	3.754400	n
15450	mb/ram/me3l/t_b/c27/bl	TBUFX2	gsc145nm	3.754400	n
15451	mb/ram/me3l/t_b/c28/bl	TBUFX2	gsc145nm	3.754400	n
15452	mb/ram/me3l/t_b/c29/bl	TBUFX2	gsc145nm	3.754400	n
15453	mb/ram/me3l/t_b/c30/bl	TBUFX2	gsc145nm	3.754400	n
15454	mb/ram/me3l/t_b/c31/bl	TBUFX2	gsc145nm	3.754400	n
15455	mm/gpa/ce/qout_req	DFPPOSXL1	gsc145nm	7.979100	n
15456	mm/gpa/ops/ml/qout_req	DFPPOSXL1	gsc145nm	7.979100	n
15457	mm/gpa/ops/ml/qout_req	DFPPOSXL1	gsc145nm	7.979100	n
15458	mm/gpa/outs/ml/qout_req	DFPPOSXL1	gsc145nm	7.979100	n
15459	mm/gpa/outs/ml/qout_req	DFPPOSXL1	gsc145nm	7.979100	n
15460	o/tr/t0/b1	TBUFX2	gsc145nm	3.754400	n
15461	o/tr/t1/b1	TBUFX2	gsc145nm	3.754400	n
15462	o/tr/t2/b1	TBUFX2	gsc145nm	3.754400	n
15463	o/tr/t3/b1	TBUFX2	gsc145nm	3.754400	n
15464	o/tr/t4/b1	TBUFX2	gsc145nm	3.754400	n
15465	o/tr/t5/b1	TBUFX2	gsc145nm	3.754400	n
15466	o/tr/t6/b1	TBUFX2	gsc145nm	3.754400	n
15467	o/tr/t7/b1	TBUFX2	gsc145nm	3.754400	n
15468	o/tr/t8/b1	TBUFX2	gsc145nm	3.754400	n
15469	o/tr/t9/b1	TBUFX2	gsc145nm	3.754400	n
15470	o/tr/t10/b1	TBUFX2	gsc145nm	3.754400	n
15471	o/tr/t11/b1	TBUFX2	gsc145nm	3.754400	n
15472	o/tr/t12/b1	TBUFX2	gsc145nm	3.754400	n
15473	o/tr/t13/b1	TBUFX2	gsc145nm	3.754400	n
15474	o/tr/t14/b1	TBUFX2	gsc145nm	3.754400	n
15475	o/tr/t15/b1	TBUFX2	gsc145nm	3.754400	n
15476	o/tr/t16/b1	TBUFX2	gsc145nm	3.754400	n
15477	o/tr/t17/b1	TBUFX2	gsc145nm	3.754400	n
15478	o/tr/t18/b1	TBUFX2	gsc145nm	3.754400	n
15479	o/tr/t19/b1	TBUFX2	gsc145nm	3.754400	n
15480	o/tr/t20/b1	TBUFX2	gsc145nm	3.754400	n
15481	o/tr/t21/b1	TBUFX2	gsc145nm	3.754400	n
15482	o/tr/t22/b1	TBUFX2	gsc145nm	3.754400	n
15483	o/tr/t23/b1	TBUFX2	gsc145nm	3.754400	n
15484	o/tr/t24/b1	TBUFX2	gsc145nm	3.754400	n
15485	o/tr/t25/b1	TBUFX2	gsc145nm	3.754400	n
15486	o/tr/t26/b1	TBUFX2	gsc145nm	3.754400	n
15487	o/tr/t27/b1	TBUFX2	gsc145nm	3.754400	n
15488	o/tr/t28/b1	TBUFX2	gsc145nm	3.754400	n
15489	o/tr/t29/b1	TBUFX2	gsc145nm	3.754400	n
15490	o/tr/t30/b1	TBUFX2	gsc145nm	3.754400	n
15491	o/tr/t31/b1	TBUFX2	gsc145nm	3.754400	n
15492	vb/bd/me0/qout_req	DFPPOSXL1	gsc145nm	7.979100	n
15493	vb/bd/me1/qout_req	DFPPOSXL1	gsc145nm	7.979100	n
15494	vb/bd/me2/qout_req	DFPPOSXL1	gsc145nm	7.979100	n
15495	vb/bd/me3/qout_req	DFPPOSXL1	gsc145nm	7.979100	n
15496	vb/bd/me4/qout_req	DFPPOSXL1	gsc145nm	7.979100	n
15497	-----				
15498	Total 14859 cells			40566.448185	
15499	1				
	xxxxx				

Figure 8: Carry-Lookahead Logic Synthesis Cell Report

1.2.3 Post Logic Synthesis Simulation

```

Terminal
File Edit View Terminal Tabs Help
Elaborating the design hierarchy:
Top level design units:
  BUFX4
  BUFX4I
  CLKBUF3
  CLKBUF3
  DFFNFGX1
  DFFSR
  FAX1
  INVX1
  INVX2
  INVX4
  INVX8
  LATCH
  MUX2X1
  MUX2X1
  NOR2X1
  NOR3X1
  OA122X1
  OR2X2
  TBUFX1
  stimulus
Building instance overlay tables: ..... Done
Generating native code: ..... Done
  worklib.stimulus:v_0x416cb96f>
    streams: 13, words: 17664
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
  Instances Unique
  Modules: 14379 34
  UDPs: 1628 4
  Primitives: 25226 6
  Timing outputs: 14379 19
  Registers: 1638 18
  Scalar wires: 16237 -
  Composed wires: 46 5
  Always blocks: 1 1
  Initial blocks: 3 3
  Pseudo assignments: 9 9
  Timing checks: 9769 1625
  Simulation timescale: 10ps
Writing initial simulation snapshot: worklib.BUFX4.v
Loading initial simulation snapshot: worklib.BUFX4.v ..... Done
xcelium-- source /apps/cadence/XCELUM1803/tools/xcelium/files/xmsimrc
xcelium-- run
Simulation complete via $finish() at time 501 NS + 0
./tb_cpu.v:30 # $finish;
xcelium-- exit
alukens@saturn.ece.iit.edu:~%
```

Figure 9: Carry-Lookahead Logic Synthesis Simulation

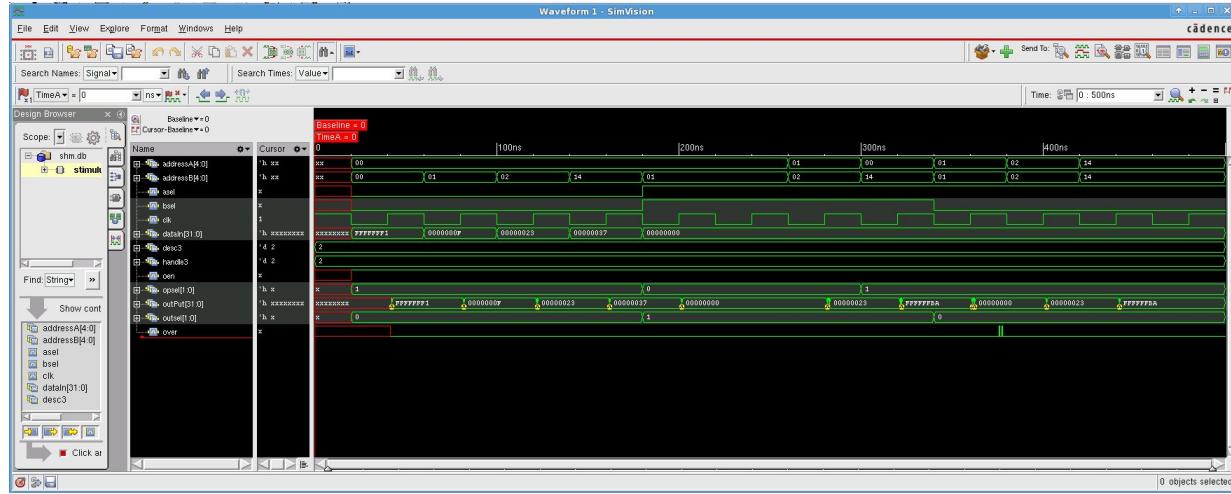


Figure 10: Carry-Lookahead Logic Synthesis Simvision Data

1.2.4 Physical Synthesis Timing Report

Report Screenshot After physical synthesis, the slack time of the Carry-Lookahead CPU design was found to be 17.921. This means that the period can be decreased by 17.921ns and the design will not violate any timing constraints.

```
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
File1.vhd timing.rep.5final [ ]
```

```
1 #####*****#
2 # Generated by: Cadence Encounter 10.13-a252_1
3 # Host ID: Linux x86_64 (Host ID: saturn.ece.iit.edu)
4 # Generated on: Tue Nov 24 10:27:23 2020
5 # Design: cpu
6 # Command: report_timing -nworst 10 -net > timing.rep.5.final
7 # File: File1.vhd
# Post-Synthesis Setup Check with Pin sb/mg/mes30/l1/me1/qut/reg0/CLK
# Endpoint: mb/rammer30/l1/me1/qut/reg0 (/) checked with leading edge of
10 "clk"
11 BeginningPoint: m0pd/bb/me1/qut/reg0/Q (v) triggered by leading edge of
12 "clk"
13 End Arrival Time 0.344
14 - Setup 4.499
15 + Phase Shift 33.000
16 = Required Time 28.846
17 - Arrival Time 10.925
18 + Slack 17.921
19 Clock Rise Edge 0.000
20 + Clock Network Latency (Prop) 0.333
21 = BeginningPoint Arrival Time 0.333
22 +-----+
23 | Pin | Edge | Net | Cell | Delay | Arrival | Required |
24 | | | | | | Time | | Time |
25 +-----+
26 | m0pd/bb/me1/qut/reg/CLK | ^ | clk_L4_N9 | DFFFSQ1 | 0.241 | 0.438 | 16.504 |
27 | m0pd/bb/me1/qut/reg/Q | v | B1[1] | AND2X1 | 0.094 | 0.438 | 16.504 |
28 | U1882/R | v | n11 | AND2X1 | 0.112 | 0.437 | 16.566 |
29 | U1882/Y | v | l13/p1[1] | AND2X1 | 0.112 | 0.437 | 16.668 |
30 | U1222/B | v | l13/p1[1] | AND2X1 | 0.000 | 0.747 | 16.669 |
31 | U1222/Y | v | l13/f02/m3 | AND2X1 | 0.049 | 0.796 | 16.717 |
32 | U1223/A | v | l13/f02/n3 | AND2X1 | 0.000 | 0.796 | 16.717 |
33 | U1223/Y | v | l13/f02/n3 | AND2X1 | 0.000 | 0.796 | 16.722 |
34 | a13/f02/03/C | ^ | n64 | CA12X1 | 0.000 | 0.801 | 16.723 |
35 | a13/f02/03/Y | v | l13/c01[1] | CA12X1 | 0.128 | 0.828 | 16.849 |
36 | U1385/A | v | l13/c01[1] | INR2X1 | 0.000 | 0.828 | 16.849 |
37 | U1385/R | x | n2111 | INR2X1 | 0.000 | 1.000 | 16.921 |
38 | a13/f12/03/A | v | l13/c02[1] | NOR2X1 | 0.000 | 1.000 | 16.922 |
39 | a13/f03/02/Y | v | l13/p0e[2] | NOR2X1 | 0.043 | 1.044 | 16.965 |
40 | U924/A | v | l13/p0e[2] | AND2X1 | 0.000 | 1.044 | 16.965 |
41 | U924/Y | v | n2110 | AND2X1 | 0.036 | 1.080 | 19.001 |
42 | U925/R | v | n11 | INR2X1 | 0.000 | 1.080 | 19.001 |
43 | U925/Y | v | n515 | INR2X1 | 0.004 | 1.084 | 19.005 |
44 | a13/f12/03/C | ^ | n515 | CA12X1 | 0.000 | 1.084 | 19.005 |
45 | a13/f12/03/Y | v | l13/cl[1] | CA12X1 | 0.127 | 1.211 | 19.132 |
46 | U1387/A | v | l13/cl[1] | INR2X1 | 0.000 | 1.211 | 19.132 |
47 | U1387/R | x | n2112 | INR2X1 | 0.000 | 1.227 | 19.137 |
48 | a13/f13/02/A | v | n2111 | NOR2X1 | 0.000 | 1.286 | 19.207 |
49 | a13/f13/02/Y | v | l13/ps1[2] | NOR2X1 | 0.048 | 1.334 | 19.255 |
50 | U1216/A | v | l13/ps1[2] | AND2X1 | 0.001 | 1.335 | 19.256 |
51 | U1216/Y | v | n2197 | AND2X1 | 0.037 | 1.372 | 19.293 |
52 | U1217/R | v | n77 | INR2X1 | 0.000 | 1.372 | 19.293 |
53 | U1217/Y | v | n661 | INR2X1 | 0.004 | 1.376 | 19.298 |
54 | a13/f22/03/C | ^ | n661 | CA12X1 | 0.000 | 1.376 | 19.298 |
55 | a13/f22/03/Y | v | l13/c2[1] | CA12X1 | 0.131 | 1.508 | 19.429 |
56 | U1389/A | v | l13/c2[1] | INR2X1 | 0.000 | 1.508 | 19.429 |
57 | U1389/R | x | n2198 | INR2X1 | 0.000 | 1.508 | 19.429 |
58 | a13/f23/02/A | v | n2198 | NOR2X1 | 0.000 | 1.582 | 19.503 |
59 | a13/f23/02/Y | v | l13/pa2[2] | NOR2X1 | 0.048 | 1.630 | 19.551 |
60 | U916/A | v | l13/pa2[2] | AND2X1 | 0.000 | 1.630 | 19.551 |
```

Figure 11: Carry-Lookahead Physical Synthesis Timing Report

Maximum Clock Frequency Estimation To calculate the maximum clock frequency, we must know the simulated frequency and the total slack time available at that frequency after physical synthesis. For the Carry-Lookahead adder design, the CPU was simulated at 30MHz, and the total slack time was found to be

17.921ns after physical synthesis. The period time for a CPU running at 30MHz is

$$T_c = \frac{1}{f} = 33.33\text{ns} \quad (1)$$

Knowing that the period time can be diminished by approximately 17.9ns, the maximum frequency is found to be

$$T_f = T_c - 17.9\text{ns} = 15.43\text{ns} \quad (2)$$

$$f = \frac{1}{T_f} = 64.795\text{MHz} \quad (3)$$

Simulating the CPU again, and setting the frequency to be approximately 64.79MHz results in the slack time to be around 0.308ns after physical synthesis. This is very close to the minimum acceptable slack time (0ns).

```

#####
# Generated by: Cadence Encounter 10.13 s292_1
# OS: Linux x86_64(Host ID saturn.ece.iit.edu)
# Generated on: Tue Nov 24 16:07:29 2020
# Design: cpu
# Command: report_timing -nworst 10 -net > timing.rep.5.final
#####
Path 1: MET Setup Check with Pin mb/ram/mer11/l1/me31/qout_reg/CLK
Endpoint: mb/ram/mer11/l1/me31/qout_reg/D (^) checked with leading edge of
'clk'
Beginpoint: m0pd/aa/mel/qout_reg/Q          (^) triggered by leading edge of
'clk'
Other End Arrival Time      0.344
- Setup                   4.251
+ Phase Shift              15.434
= Required Time            11.527
- Arrival Time             11.219
= Slack Time               0.308
  Clock Rise Edge          0.000
  + Clock Network Latency (Prop) 0.337
  = Beginpoint Arrival Time  0.337
+
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|   Pin    |   Edge   |       Net       |   Cell   |   Delay  | Arrival | Required |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| m0pd/aa/mel/qout_reg/Q | ^ | clk | 14 NS |          |          |  A 337 |  A 645 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 12: Carry-Lookahead Maximum Frequency Report

1.2.5 Post Physical Synthesis Simulation

```

Terminal
File Edit View Terminal Tabs Help
Errors: 0, warnings: 0
Caching library 'worklib' ..... Done
Elaborating the design hierarchy: ..... Done
Top level design units:
BUF4X
CLKBUF1
CLKBUF2
CLKBUF3
DFFBUF3
DFPHRGX1
DFFSR
FAXI
HAXI
INVX2
LATCH
NAND2X1
NAND3X1
NOR3X1
OA12X1
OR2X2
TBUFX1
stimulus
Building instant overlay tables: ..... Done
Generating native compiled code:
worklib.cpu.v <0x1072c054>
streams: 0, words: 0
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
      Instances Unique
Modules:        14464    34
UDPs:          1628     4
Primitives:    25311     6
Timing outputs: 14464    19
Scalar wires:  1638    18
Scalar wires:  16322    -
Expanded wires: 46      5
Always blocks: 1       1
Initial blocks: 3       3
Pseudo assignments: 9       9
Timing checks:  9769   1625
Timing checks:  10ps
Writing initial simulation snapshot: worklib.BUFX4:v
Loading snapshot worklib.BUFX4:v ..... Done
xcelium> source /apps/cadence/XCELIUM1803/tools/xcelium/files/xmsimrc
xcelium> run
Simulation complete via $finish() at time 501 NS + 0
./tb_cpu.v:>0 $finish;
xcelium> exit
alukens@saturn.ece.iit.edu:~%
```

Figure 13: Carry-Lookahead Physical Synthesis Simulation

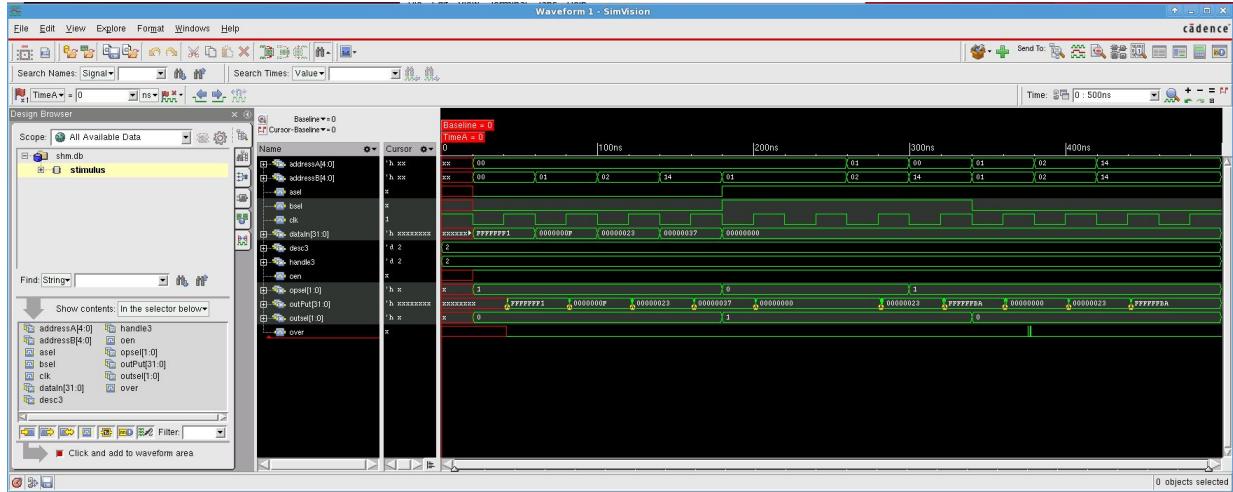


Figure 14: Carry-Lookahead Physical Synthesis Simvision Data

1.2.6 RTL Simulation with New Testbench

To perform the new testbench simulation, the runtime of the simulation was increased to 1500ns in the testbench file. This was done by closing the simulation database after 1500ns (altered portion near top of testbench).

```

Terminal
File Edit View Terminal Tags Help
xmelab: *W,CUWWSPI ./cpu_CLA.v,617[24]: 1 output port was not connected:
xmelab: (./cpu_CLA.v,575): over

Top level design units:
    stimulus
Building instance overlay tables: ..... Done
Generating native compiled code:
    worklib.muxv <0x203101>
        streams: 32, words: 6720
    worklib.as32bv <0x5203101>
        streams: 2, words: 420
    worklib.logic_fnv <0x5af3a3eb>
        streams: 1, words: 1113
    worklib.alterv <0x212a0d0>
        streams: 1, words: 1246
    worklib.tristate_32v <0x71d5a1b7>
        streams: 0, words: 0
    worklib.mux2to1v <0x3f664099>
        streams: 0, words: 0
    worklib.mux1to1_32v <0xd00009bec>
        streams: 1, words: 0
    worklib.memoryBlockv <0x090a1896>
        streams: 1, words: 208
    worklib.dregiv <0x60a2e9c9>
        streams: 2, words: 264
    worklib.stimulusv <0x19f014cc>
        streams: 13, words: 35077
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
Instances Unique
Modules: 6607 33
Variables: 7117 6
Registers: 2466 14
Scalar wires: 3692 -
Expanded wires: 1198 41
Vectorized wires: 6 -
Always blocks: 2156 4
Initial blocks: 3 3
Constant assignments: 111 15
Pseudo assignments: 43 43
Writing initial simulation snapshot: worklib.stimulus:v
Loading snapshot worklib.stimulus:v ..... Done
xcelium> source /apps/cadence/XCELIUM1083/tools/xcelium/files/xsimrc
xcelium> run
xsim: *W,SIMOPT: Some objects excluded from $shm_probe due to optimizations.
    File: /tb_test.v, line = 28, pos = 11
    Scope: stimulus
    Time: 0 FS + 0
Simulation complete via $finish(1) at time 1501 NS + 0
./tb_test.v:30 # $finish;
xcelium> exit
atukens@saturn.ece.iit.edu:-

```

Figure 15: Carry-Lookahead New Testbench RTL Simulation

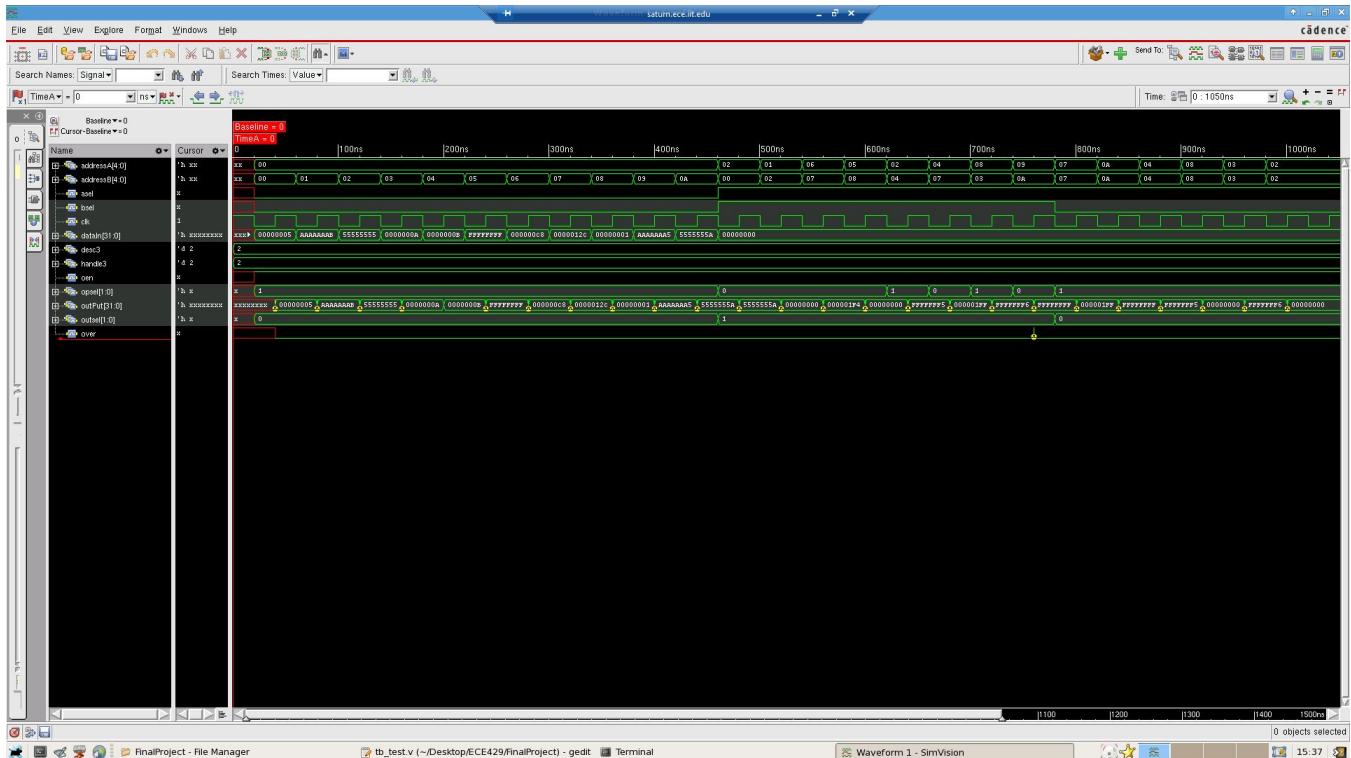
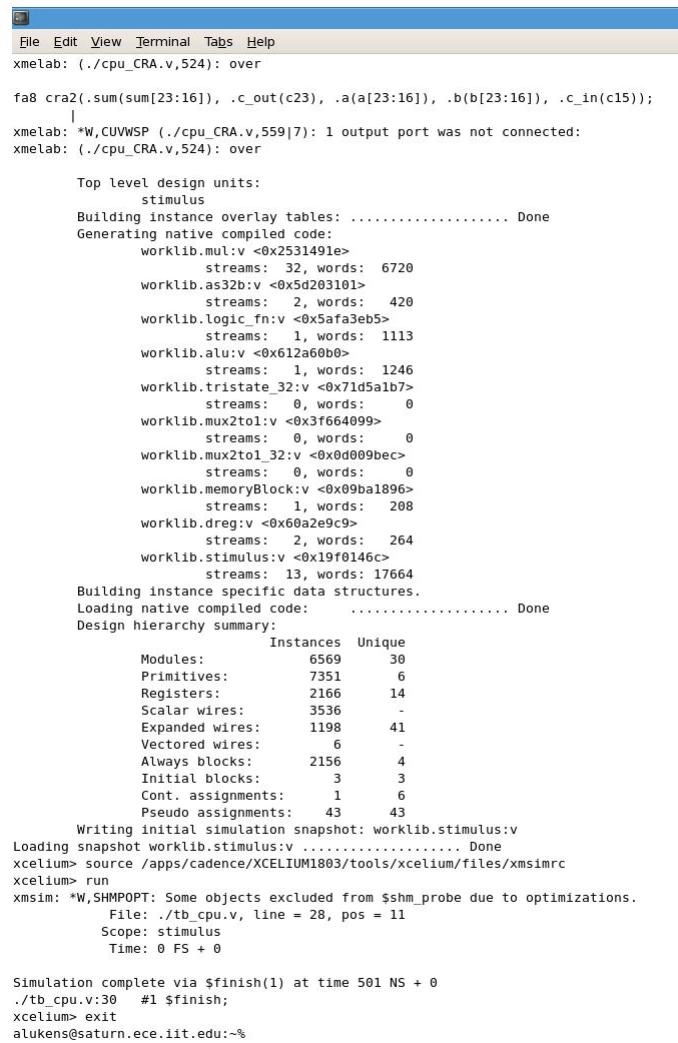


Figure 16: Carry-Lookahead New Testbench RTL Data

1.3 Carry-Ripple Adder Implementation

1.3.1 RTL Simulation



```
File Edit View Terminal Tabs Help
xmelab: (./cpu_CRA.v,524): over
fa8 cra2(.sum(sum[23:16]), .c_out(c23), .a(a[23:16]), .b(b[23:16]), .c_in(c15));
|
xmelab: *W,CUVWSP (./cpu_CRA.v,559|7): 1 output port was not connected:
xmelab: (./cpu_CRA.v,524): over

Top level design units:
    stimulus
Building instance overlay tables: ..... Done
Generating native compiled code:
    worklib.mul:v <0x2531491e>
        streams: 32, words: 6720
    worklib.as32b:v <0x5d203101>
        streams: 2, words: 420
    worklib.logic_fn:v <0x5afa3eb5>
        streams: 1, words: 1113
    worklib.alu:v <0x612a60b0>
        streams: 1, words: 1246
    worklib.tristate_32:v <0x71d5a1b7>
        streams: 0, words: 0
    worklib.mux2to1:v <0x3f664099>
        streams: 0, words: 0
    worklib.mux2to1_32:v <0x0d0009bec>
        streams: 0, words: 0
    worklib.memoryBlock:v <0x09ba1896>
        streams: 1, words: 208
    worklib.dreg:v <0x60a2e9c9>
        streams: 2, words: 264
    worklib.stimulus:v <0x19f0146c>
        streams: 13, words: 17664
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
      Instances Unique
Modules:       6569   30
Primitives:    7351    6
Registers:    2166   14
Scalar wires:  3536    -
Expanded wires: 1198   41
Vectorized wires: 6    -
Always blocks: 2156   4
Initial blocks: 3    3
Cont. assignments: 1    6
Pseudo assignments: 43   43
Writing initial simulation snapshot: worklib.stimulus:v
Loading snapshot worklib.stimulus:v ..... Done
xcelium> source /apps/cadence/XCELIUM1803/tools/xcelium/files/xmsimrc
xcelium> run
xmsim: *W,SHMPOPT: Some objects excluded from $shm_probe due to optimizations.
      File: ./tb_cpu.v, line = 28, pos = 11
      Scope: stimulus
      Time: 0 FS + 0

Simulation complete via $finish() at time 501 NS + 0
./tb_cpu.v:30 #1 $finish;
xcelium> exit
alukens@saturn.ece.iit.edu:~%
```

Figure 17: Carry-Ripple RTL Simulation

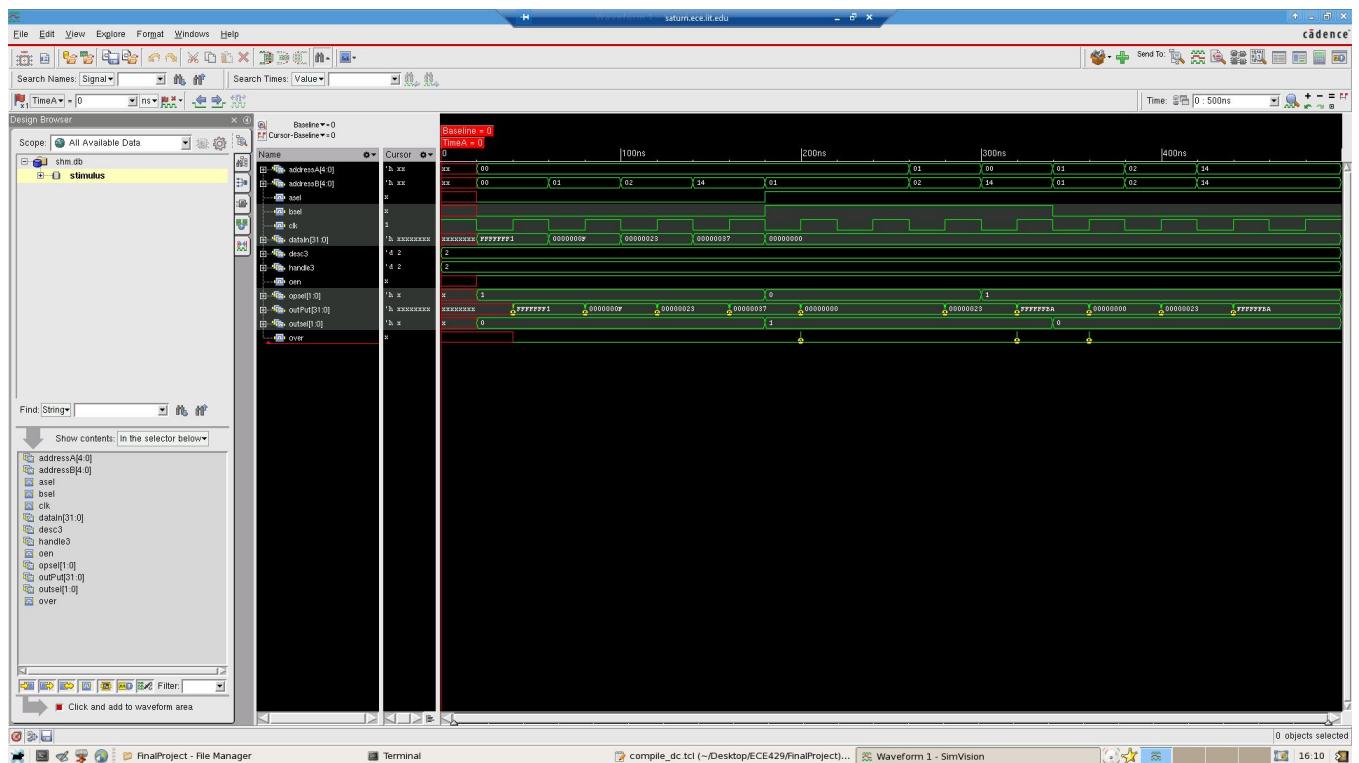
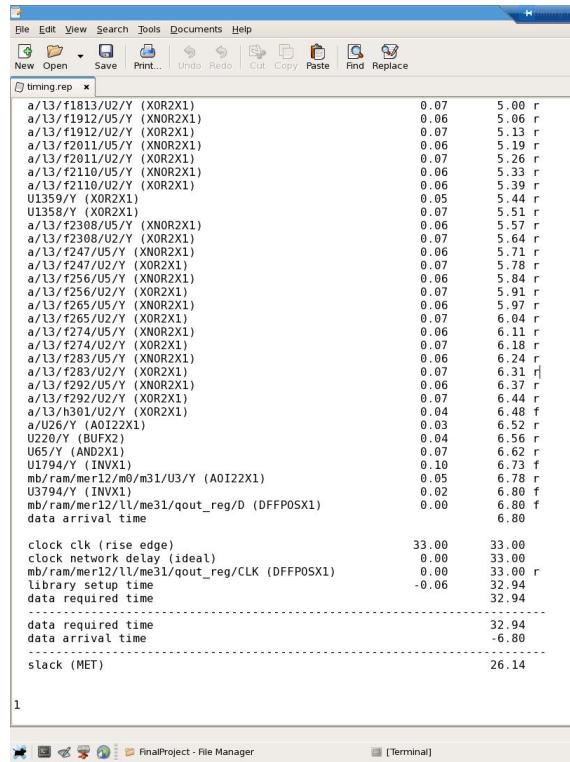


Figure 18: Carry-Ripple Simvision Data

1.3.2 Logic Synthesis Timing & Cell Reports



```

File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
timing.rep x
a/l3/f1813/u2/y (XOR2X1) 0.07 5.00 r
a/l3/f1912/u5/y (XNOR2X1) 0.06 5.06 r
a/l3/f1912/u2/y (XOR2X1) 0.07 5.13 r
a/l3/f2011/u5/y (XNOR2X1) 0.06 5.19 r
a/l3/f2011/u2/y (XOR2X1) 0.07 5.26 r
a/l3/f2110/u5/y (XNOR2X1) 0.06 5.33 r
a/l3/f2110/u2/y (XOR2X1) 0.06 5.39 r
U1359/r (XOR2X1) 0.05 5.44 r
U1358/r (XNOR2X1) 0.07 5.51 r
a/l3/f2310/u5/y (XNOR2X1) 0.06 5.57 r
a/l3/f2409/u2/y (XOR2X1) 0.07 5.64 r
a/l3/f247/u5/y (XNOR2X1) 0.06 5.71 r
a/l3/f247/u2/y (XOR2X1) 0.07 5.78 r
a/l3/f256/u5/y (XNOR2X1) 0.06 5.84 r
a/l3/f256/u2/y (XOR2X1) 0.07 5.91 r
a/l3/f265/u5/y (XNOR2X1) 0.06 5.97 r
a/l3/f265/u2/y (XOR2X1) 0.07 6.04 r
a/l3/f274/u5/y (XNOR2X1) 0.06 6.11 r
a/l3/f274/u2/y (XOR2X1) 0.07 6.18 r
a/l3/f283/u5/y (XNOR2X1) 0.06 6.24 r
a/l3/f283/u2/y (XOR2X1) 0.07 6.31 r
a/l3/f292/u5/y (XNOR2X1) 0.06 6.37 r
a/l3/f292/u2/y (XOR2X1) 0.07 6.44 r
a/l3/h301/u2/y (XOR2X1) 0.04 6.48 f
a/U26/r (A0I22X1) 0.03 6.52 r
U220/y (BUFX2) 0.04 6.56 r
U65/y (AND2X1) 0.07 6.62 r
U1794/r (INV1) 0.10 6.73 f
mb/ram/mer12/l1/mem31/qout_reg/CLK (DFFP0SX1) 0.05 6.78 r
U3794/y (INVX1) 0.02 6.80 f
mb/ram/mer12/l1/mem31/qout_reg/D (DFFP0SX1) 0.00 6.80
data arrival time 26.14

clock clk (rise edge) 33.00 33.00
clock network delay (ideal) 0.00 33.00
mb/ram/mer12/l1/mem31/qout_reg/CLK (DFFP0SX1) 0.00 33.00 r
library setup time -0.06 32.94
data required time 32.94
----- data required time 32.94
data arrival time -6.80
----- slack (MET) 26.14

1
FinalProject - File Manager [Terminal]

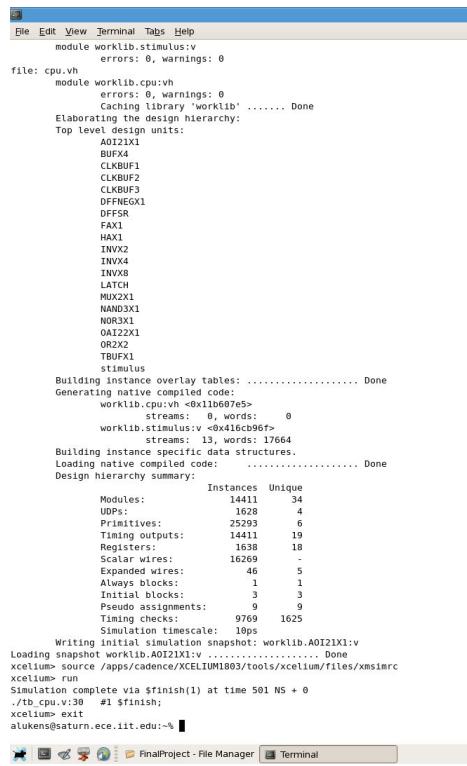
```

Figure 19: Carry-Ripple Logic Synthesis Timing Report

File Edit View Search Tools Documents Help					
New Open	Save Print...	Undo Redo	Cut Copy Paste	Find	Replace
cell.rep					
mb/ram/mer3l/t_b/t3l/b1	TBUFX2	gscl45nm	3.754400	n	
mmOpn/oe/qout_req	DFFPOSX1	gscl45nm	7.978100	n	
mmOpn/ops/m0/qout_req	DFFPOSX1	gscl45nm	7.978100	n	
mmOpn/ops/m1/qout_req	DFFPOSX1	gscl45nm	7.978100	n	
mmOpn/outs/m0/qout_req	DFFPOSX1	gscl45nm	7.978100	n	
mmOpn/outs/m1/qout_req	DFFPOSX1	gscl45nm	7.978100	n	
o/tr/t0/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t1/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t2/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t3/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t4/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t5/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t6/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t7/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t8/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t9/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t10/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t11/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t12/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t13/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t14/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t15/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t16/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t17/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t18/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t19/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t20/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t21/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t22/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t23/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t24/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t25/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t26/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t27/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t28/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t29/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t30/b1	TBUFX2	gscl45nm	3.754400	n	
o/tr/t31/b1	TBUFX2	gscl45nm	3.754400	n	
wb/bd/me0/qout_req	DFFPOSX1	gscl45nm	7.978100	n	
wb/bd/me1/qout_req	DFFPOSX1	gscl45nm	7.978100	n	
wb/bd/me2/qout_req	DFFPOSX1	gscl45nm	7.978100	n	
wb/bd/me3/qout_req	DFFPOSX1	gscl45nm	7.978100	n	
wb/bd/me4/qout_req	DFFPOSX1	gscl45nm	7.978100	n	
Total 14390 cells			48610.093084		
1					

Figure 20: Carry-Ripple Logic Synthesis Cell Report

1.3.3 Post Logic Synthesis Simulation



```

File Edit View Terminal Tabs Help
      module worklib.stimulus:v
      errors: 0, warnings: 0
file: cpu.vh
      module worklib:cpu:vh
      errors: 0, warnings: 0
      Caching library 'worklib' ..... Done
Elaborating the design hierarchy:
Top level design units:
      AOI21X1
      BUFX4
      CLKBUF1
      CLKBUF2
      CLKBUF3
      DFFNFGX1
      DFFPSR
      FAX1
      HAX1
      INVX2
      INVX4
      INVXB
      LATCH
      MUX2X1
      NAMD3X1
      NOR3X1
      OA122X1
      OR2X2
      TBUFX1
      TBUFZ1
Building instance overlay tables: ..... Done
Generating native compiled code:
  worklib.cpu:vh <0x11b607e5>
    streams: 0, words: 0
  worklib.stimulus:v <0x416cb96f>
    streams: 13, words: 17664
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
  Instances Unique
  Modules: 14411 34
  UDPs: 1628 4
  Streams: 2548 6
  Timing Outputs: 14411 19
  Registers: 1638 18
  Scalar wires: 16269 -
  Expanded wires: 46 5
  Always blocks: 1 1
  Initial blocks: 3 3
  Node assignments: 9 9
  Timing checks: 9769 1625
  Simulation timescale: 10ps
Writing initial simulation snapshot: worklib.AOI21X1:v
Loading snapshot worklib.AOI21X1:v ..... Done
xcelium source /apps/cadence/XCELUM1B03/tools/xcelium/files/xmsimrc
xcelium run
Simulation complete via $finish() at time 501 NS + 0
./tb_cpu v30 #1 $finish;
xcelium -exit
alukens@saturn.ece.iit.edu:~% 

```

Figure 21: Carry-Ripple Logic Synthesis Simulation

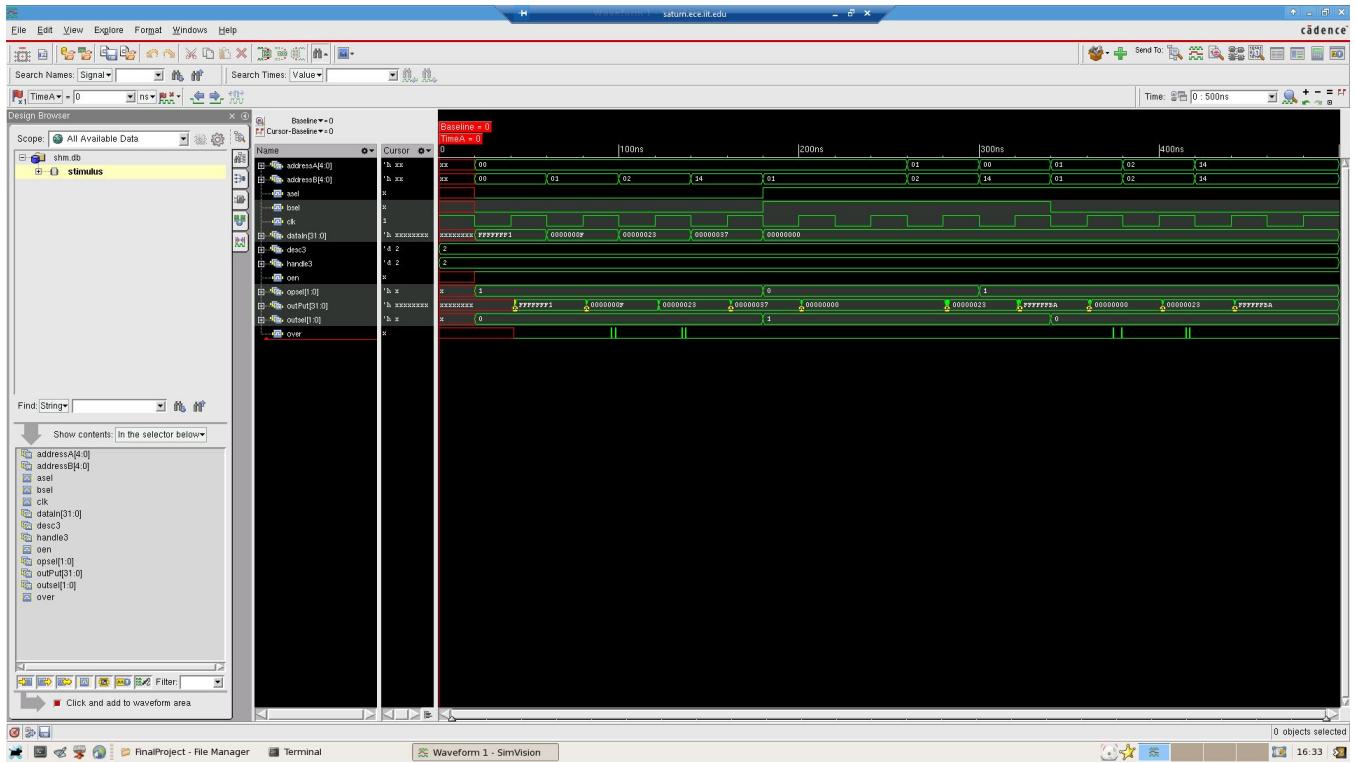


Figure 22: Carry-Ripple Logic Synthesis SimVision Data

1.3.4 Physical Synthesis Timing Report

Report Interpretation After physical synthesis, the slack time of the Carry-Ripple CPU design was found to be 17.458ns. This means that the period can be decreased by 17.458ns and the design will not violate any timing constraints.

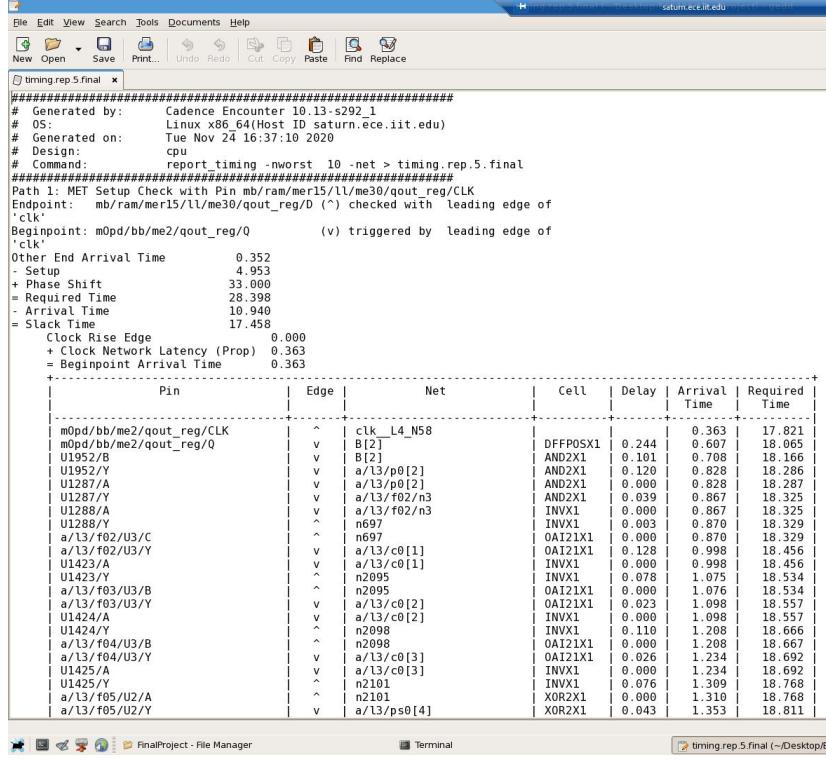


Figure 23: Carry-Ripple Physical Synthesis Timing Report

Maximum Clock Frequency Estimation To calculate the maximum clock frequency, we must know the simulated frequency and the total slack time available at that frequency after physical synthesis. For the Carry-Ripple adder design, the CPU was simulated at 30MHz, and the total slack time was found to be 17.458ns after physical synthesis. The period time for a CPU running at 30MHz is

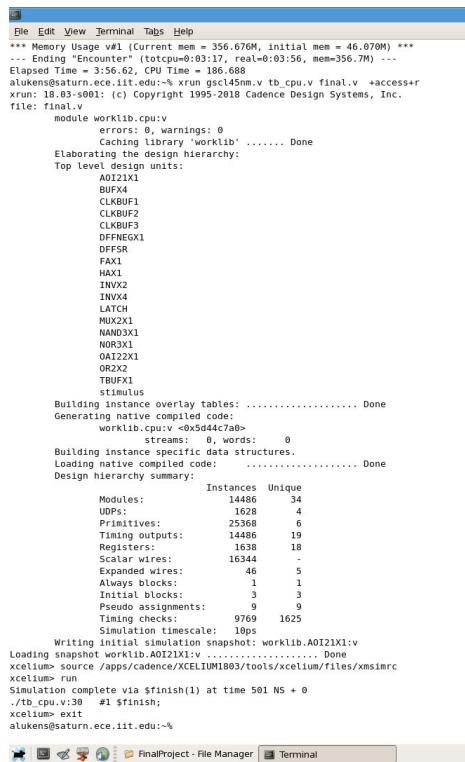
$$T_c = \frac{1}{f} = 33.33\text{ns} \quad (4)$$

Knowing that the period time can be diminished by approximately 17.458ns, the maximum frequency is found to be

$$T_f = T_c - 17.458\text{ns} = 15.875\text{ns} \quad (5)$$

$$f = \frac{1}{T_f} = 62.991\text{MHz} \quad (6)$$

1.3.5 Post Physical Synthesis Simulation



```

File Edit View Terminal Tabs Help
*** Memory Usage v1 (Current mem = 356.070M, initial mem = 46.070M) ***
.. Ending "Encounter" (Time=0:03:17, test=0:03:56, mem=356.7M) ...
Elapsed Time: 0:03:56.62, CPU Time: 0:03:56.608
alukens@saturn.ece.iit.edu:~% xrun gsch45nm.v tb.cpu.v final.v +access+r
xrun: 18.03+0001: (c) Copyright 1995-2018 Cadence Design Systems, Inc.
file: final.v
    module worklib.cpu.v
        error: 0 warnings: 0
        Caching library 'worklib' ..... Done
    Elaborating the design hierarchy:
    Top level design units:
        A0I2IX1
        BUF4X
        CLKBUF1
        CLKBUF2
        CLKBUF3
        DFNFNEGX1
        DFFSR
        FAX1
        HAX1
        INVX2
        INVX4
        LATCH
        MUX2X1
        NAND3X1
        NOR3X1
        OA122X1
        O2X2
        TBDFX1
        stimulus
Building instance overlay tables: ..... Done
Generating native compiled code:
worklib.cpu.v<0xd4d4c7a8>
    streams: 0, words: 0
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
    Instances Unique
    Modules: 14486 34
    UPDs: 1628 4
    Streams: 25488 6
    Timing outputs: 14486 19
    Registers: 1638 18
    Scalar wires: 16344 -
    Expanded wires: 46 5
    Always blocks: 1 1
    General blocks: 3 3
    Pseudo assignments: 0 9
    Timing checks: 9769 1625
Simulation timescale: 10ps
Writing initial simulation snapshot: worklib.A0I2IX1:v
Loading snapshot worklib.A0I2IX1:v ..... Done
xcelium source /apps/cadence/XCELIUM1803/tools/xcelium/files/xmsimrc
xcelium run
Simulation complete via $finish() at time 501 NS + 0
.tb cpu v:30 #1 $finish;
xcelium> exit
alukens@saturn.ece.iit.edu:~%

```

Figure 24: Carry-Ripple Physical Synthesis Simulation

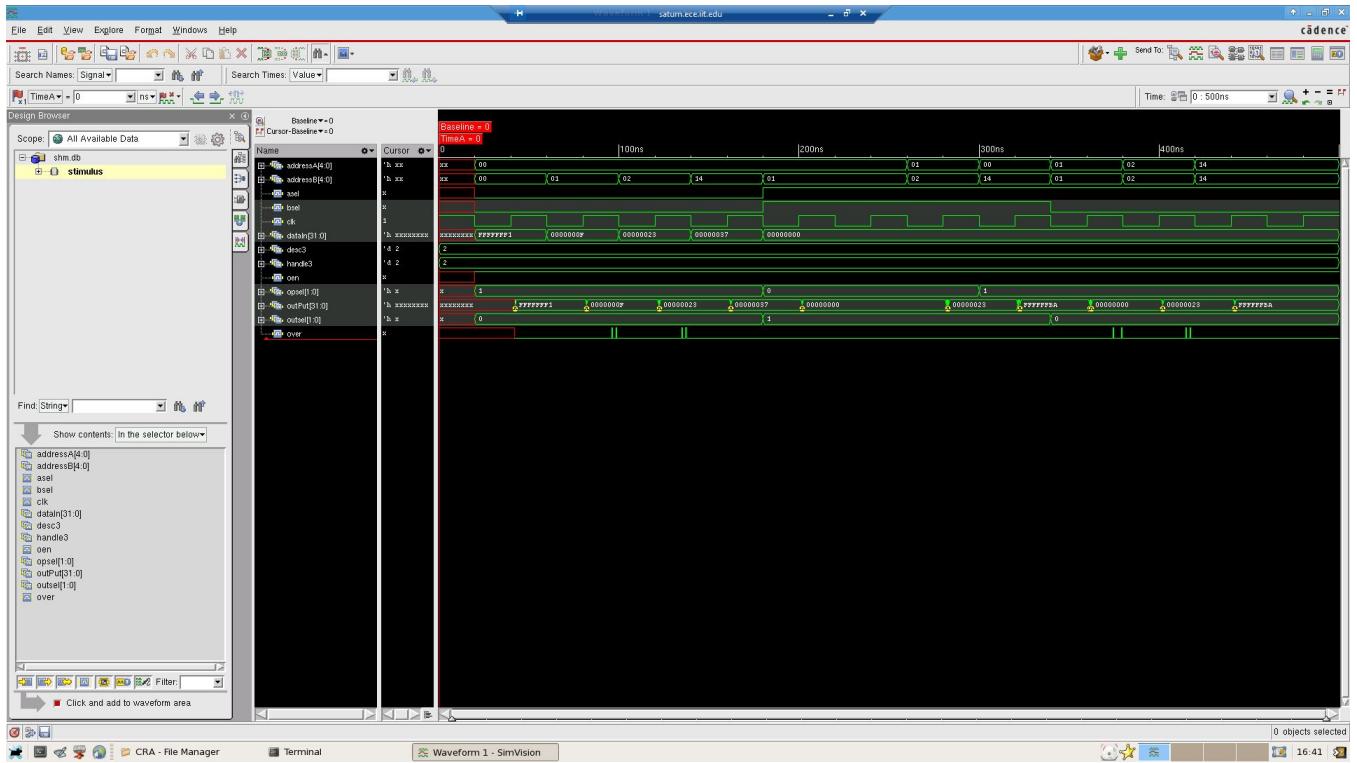


Figure 25: Carry-Ripple Physical Synthesis SimVision Data

1.3.6 RTL Simulation with New Testbench

To perform the new testbench simulation, the runtime was increased to 1500ns in the testbench file. This was done by closing the simulation database after 1500ns (altered portion near top of testbench file).

```

File Edit View Terminal Tabs Help
fa8 cra2(.sum(sum[23:16]), .c_out(c23), .a(a[23:16]), .b(b[23:16]), .c_in(c15));
|
xmelab: *W,CUWNSP (.cpu_CRA.v,559|): 1 output port was not connected:
xmelab: (.cpu_CRA.v,524): over

Top level design units:
    stimulus
Building instance overlay tables: ..... Done
Generating native compiled code
worklib.tb1.v:<0x2023101>
    streams: 32, words: 6720
worklib.as32b.v:<0x5d283101>
    streams: 2, words: 420
worklib.logic_fnv:<0x5af3a3eb5>
    streams: 1, words: 1113
worklib.tadag:<0x2023101>
    streams: 1, words: 1246
worklib.tristate_32.v:<0xb7105a1b7>
    streams: 0, words: 0
worklib.mux2to1v:<0x3f664099>
    streams: 0, words: 0
worklib.mux2to1_32.v:<0xb00999ec>
    streams: 0, words: 0
worklib.mux4to2:<0xb009a1896>
    streams: 1, words: 288
worklib.dregiv:<0x60a2e9c9>
    streams: 2, words: 264
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
Instances Unique
Modules: 6569 30
Primitives: 7351 6
Registers: 2166 14
Scalar wires: 3536 -
Vectorized wires: 1198 41
Vectorized wires: 6
Always blocks: 2156 4
Initial blocks: 3 3
Cont. assignments: 1 6
Pseudo assignments: 43 43
Writing initial simulation snapshot: worklib.stimulus:v
Loading snapshot: worklib.stimulus:v ..... Done
xcelium> source /apps/cadence/XCELUM1083/tools/xcelium/files/xmsimrc
xcelium> run
xmsim: *W,SHMPOPT: Some objects excluded from $shm probe due to optimizations.
    File: ./tb_test.v, lin = 28, pos = 11
    Scope: stimulus
    Time: 0 FS + 0

Simulation complete via $finish(1) at time 1501 NS + 0
./tb_test.v:30 #1 $finish;
xcelium> exit
alukens@saturn.ece.iit.edu:~%
```

Figure 26: Carry-Ripple New Testbench RTL Simulation

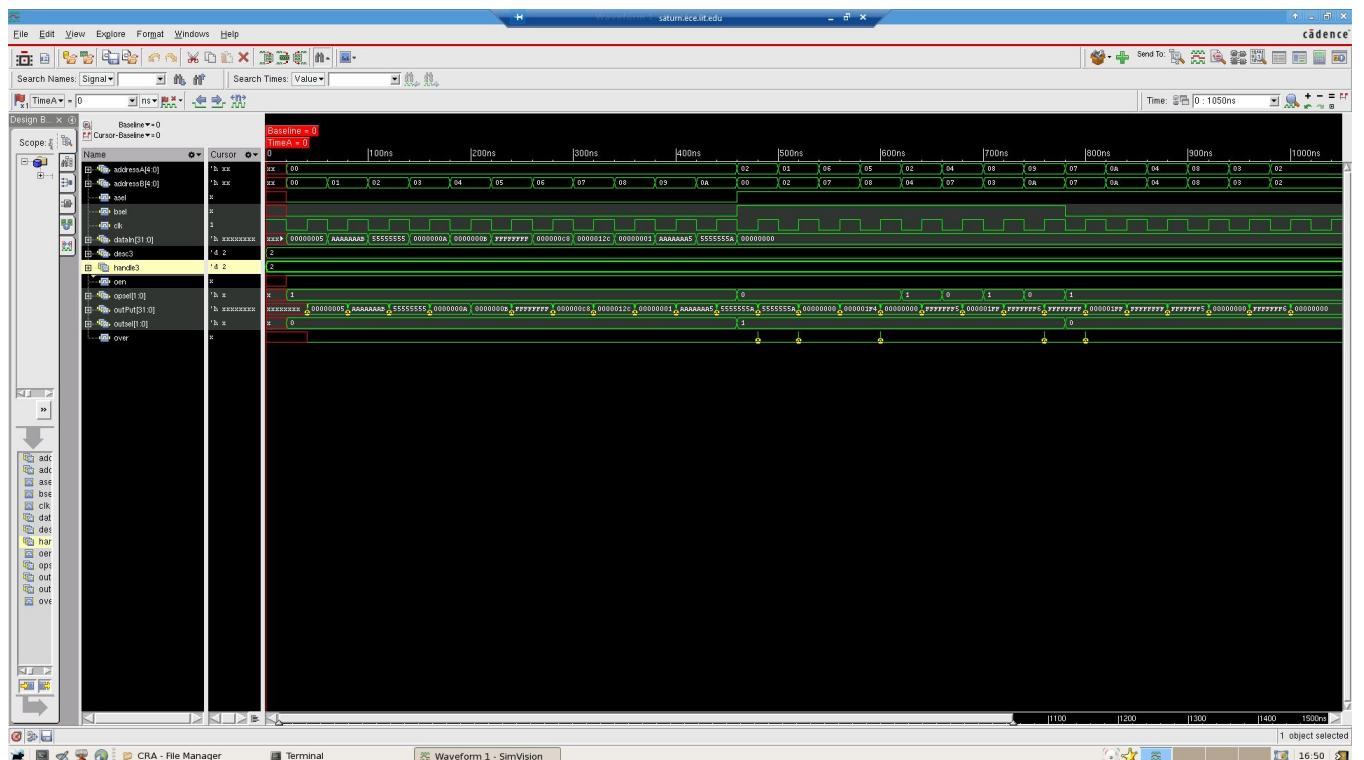
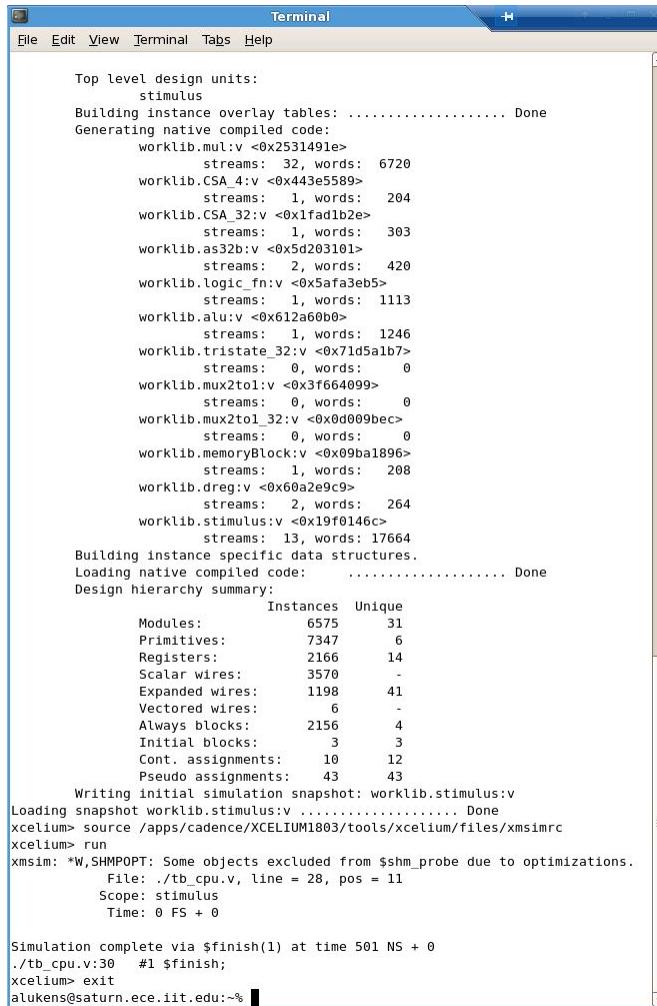


Figure 27: Carry-Ripple New Testbench RTL SimVision Data

1.4 Carry-Skip Adder Implementation

1.4.1 RTL Simulation



The screenshot shows a terminal window with the following text output:

```
Terminal
File Edit View Terminal Tabs Help

Top level design units:
    stimulus
Building instance overlay tables: ..... Done
Generating native compiled code:
    worklib.mul:v <0x253149le>
        streams: 32, words: 6720
    worklib.CSA_4:v <0x443e5589>
        streams: 1, words: 204
    worklib.CSA_32:v <0x1fad1b2e>
        streams: 1, words: 303
    worklib.as32b:v <0x5d203101>
        streams: 2, words: 420
    worklib.logic_fn:v <0x5afa3eb5>
        streams: 1, words: 1113
    worklib.alu:v <0x612a60b0>
        streams: 1, words: 1246
    worklib.tristate_32:v <0x71d5alb7>
        streams: 0, words: 0
    worklib.mux2tol:v <0x3f664099>
        streams: 0, words: 0
    worklib.mux2tol_32:v <0x0d009bec>
        streams: 0, words: 0
    worklib.memoryBlock:v <0x09ba1896>
        streams: 1, words: 208
    worklib.dreg:v <0x60a2e9c9>
        streams: 2, words: 264
    worklib.stimulus:v <0x19f0146c>
        streams: 13, words: 17664
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
      Instances   Unique
Modules:          6575     31
Primitives:       7347      6
Registers:        2166     14
Scalar wires:     3570      -
Expanded wires:   1198     41
Vectorized wires: 6         -
Always blocks:   2156      4
Initial blocks:   3         3
Cont. assignments: 10      12
Pseudo assignments: 43      43
Writing initial simulation snapshot: worklib.stimulus:v
Loading snapshot worklib.stimulus:v ..... Done
xcelium> source /apps/cadence/XCELIUM1803/tools/xcelium/files/xmsimrc
xcelium> run
xmsim: *W,SHMPOPT: Some objects excluded from $shm_probe due to optimizations.
      File: ./tb_cpu.v, line = 28, pos = 11
      Scope: stimulus
      Time: 0 FS + 0
Simulation complete via $finish(1) at time 501 NS + 0
./tb_cpu.v:30  #1 $finish;
xcelium> exit
alukens@saturn.ece.iit.edu:~%
```

Figure 28: Carry-Skip RTL Simulation

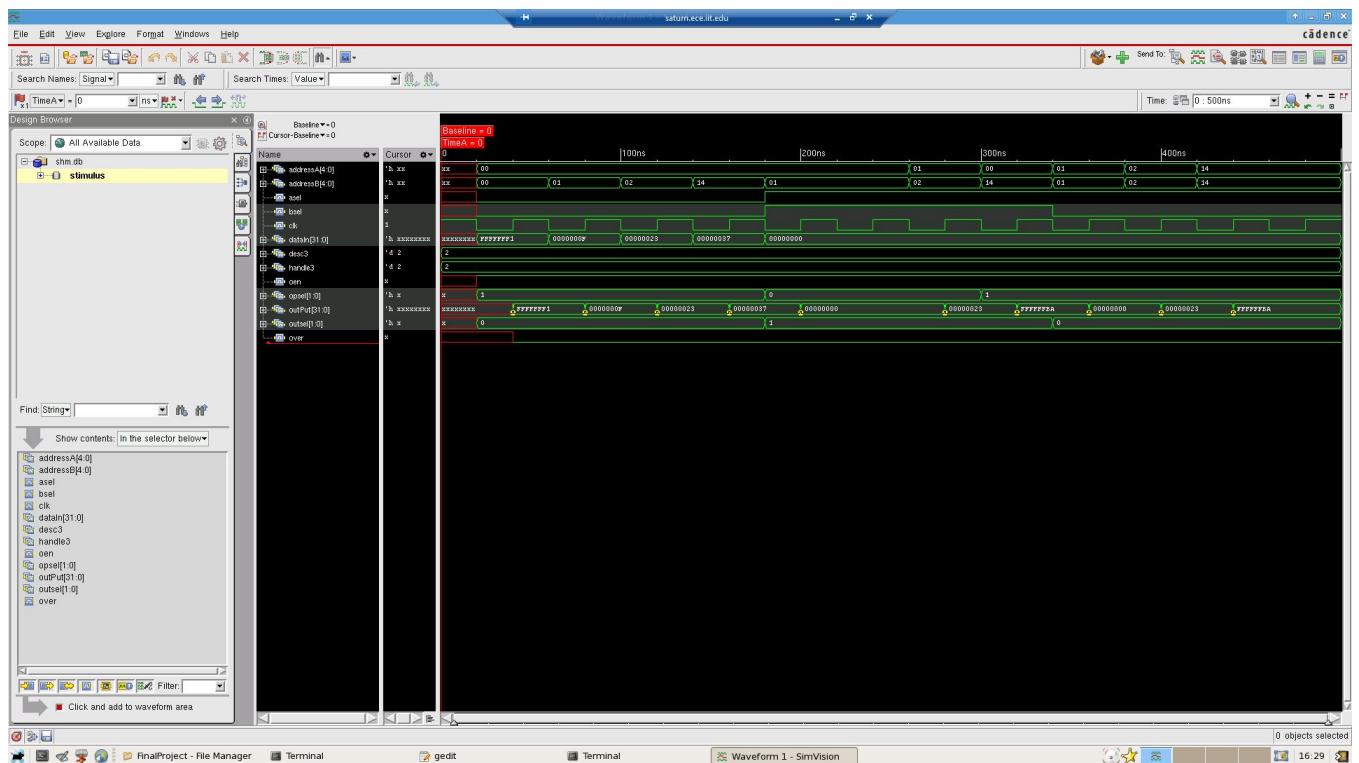
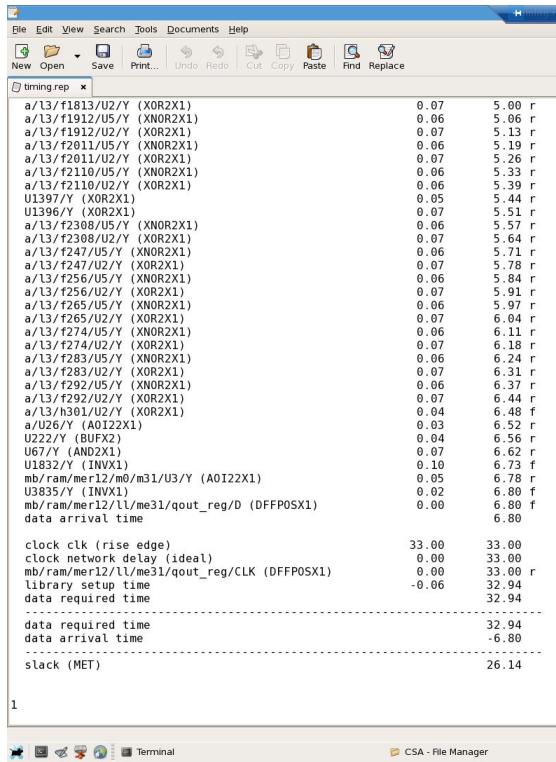


Figure 29: Carry-Skip RTL SimVision Data

1.4.2 Logic Synthesis Timing & Cell Reports



```

File Edit View Search Tools Documents Help
New Open Save Print Undo Redo Cut Copy Paste Find Replace
timing.rep x
a/13/f1813/U2/Y (XOR2X1) 0.07 5.00 r
a/13/f1912/U5/Y (XOR2X1) 0.06 5.06 r
a/13/f1912/U2/Y (XOR2X1) 0.07 5.13 r
a/13/f2011/U5/Y (XOR2X1) 0.06 5.19 r
a/13/f2011/U2/Y (XOR2X1) 0.07 5.26 r
a/13/f2110/U5/Y (XOR2X1) 0.06 5.33 r
a/13/f2110/U2/Y (XOR2X1) 0.06 5.39 r
U1397/Y (XOR2X1) 0.05 5.44 r
U1396/Y (XOR2X1) 0.07 5.51 r
a/13/f2308/U5/Y (XOR2X1) 0.06 5.57 r
a/13/f2308/U2/Y (XOR2X1) 0.07 5.64 r
a/13/f247/U5/Y (XOR2X1) 0.06 5.71 r
a/13/f247/U2/Y (XOR2X1) 0.07 5.78 r
a/13/f256/U5/Y (XOR2X1) 0.06 5.84 r
a/13/f256/U2/Y (XOR2X1) 0.07 5.91 r
a/13/f265/U5/Y (XOR2X1) 0.06 5.97 r
a/13/f265/U2/Y (XOR2X1) 0.07 6.04 r
a/13/f274/U5/Y (XOR2X1) 0.06 6.11 r
a/13/f274/U2/Y (XOR2X1) 0.07 6.18 r
a/13/f283/U5/Y (XOR2X1) 0.06 6.24 r
a/13/f283/U2/Y (XOR2X1) 0.07 6.31 r
a/13/f292/U5/Y (XOR2X1) 0.06 6.37 r
a/13/f292/U2/Y (XOR2X1) 0.07 6.44 r
a/13/h301/U2/Y (XOR2X1) 0.04 6.48 f
a/U26/Y (A0I2X21) 0.03 6.52 r
U222/Y (BUF2X2) 0.04 6.56 r
U67/Y (AND2X1) 0.07 6.62 r
U1832/Y (M12X1) 0.10 6.73 f
mb/ram/mer12/l1/me31/qout_reg/CLK (DFFP05X1) 0.05 6.78 r
U3835/Y (INVX1) 0.02 6.80 f
mb/ram/mer12/l1/me31/qout_reg/D (DFFP05X1) 0.00 6.80 f
data arrival time 6.80

clock clk (rise edge) 33.00 33.00
clock network delay (ideal) 0.00 33.00
mb/ram/mer12/l1/me31/qout_reg/CLK (DFFP05X1) 0.00 33.00 r
library setup time -0.06 32.94
data required time 32.94
-----
data required time 32.94
data arrival time -6.80
-----
slack (MET) 26.14

1

```

The screenshot shows a software interface for logic synthesis timing analysis. The main window displays a table of timing data for various logic cells and components, such as XOR2X1, AND2X1, and BUF2X2. The table includes columns for setup time, hold time, and slack. Below the table, there are sections for clock timing, library setup time, and slack calculations. At the bottom, there is a toolbar with icons for file operations and a status bar indicating 'CSA - File Manager'.

Figure 30: Carry-Skip Logic Synthesis Timing Report

Component	Type	Value	Size
mb/ram/mer31/t_b/t31/bl	TBUFX2	gscl45nm	3.754400 n
mmOpn/oe/qout_req	DFFPOSX1	gscl45nm	7.978100 n
mmOpn/ops/m0/qout_req	DFFPOSX1	gscl45nm	7.978100 n
mmOpn/ops/m1/qout_req	DFFPOSX1	gscl45nm	7.978100 n
mmOpn/outs/m0/qout_req	DFFPOSX1	gscl45nm	7.978100 n
mmOpn/outs/m1/qout_req	DFFPOSX1	gscl45nm	7.978100 n
o/tr/t0/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t1/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t2/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t3/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t4/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t5/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t6/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t7/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t8/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t9/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t10/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t11/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t12/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t13/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t14/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t15/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t16/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t17/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t18/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t19/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t20/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t21/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t22/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t23/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t24/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t25/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t26/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t27/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t28/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t29/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t30/bl	TBUFX2	gscl45nm	3.754400 n
o/tr/t31/bl	TBUFX2	gscl45nm	3.754400 n
wb/bd/me0/qout_req	DFFPOSX1	gscl45nm	7.978100 n
wb/bd/me1/qout_req	DFFPOSX1	gscl45nm	7.978100 n
wb/bd/me2/qout_req	DFFPOSX1	gscl45nm	7.978100 n
wb/bd/me3/qout_req	DFFPOSX1	gscl45nm	7.978100 n
wb/bd/me4/qout_req	DFFPOSX1	gscl45nm	7.978100 n

Total 14479 cells
1

Figure 31: Carry-Skip Logic Synthesis Cell Report

1.4.3 Post Logic Synthesis Simulation

```

Terminal
File Edit View Terminal Tabs Help
Recompiling... reason: file './cpu.vh' is newer than expected.
expected: Tue Nov 24 16:29:13 2020
actual:  Tue Nov 24 17:31:48 2020
Caching library 'worklib' ..... Done
Elaborating the design hierarchy:
Top level design units:
  BUFX4
  CLKBUF1
  CLKBUF2
  CLKBUF3
  DFNFNEGX1
  DFNFCSR
  FAI1
  HAX1
  INVX2
  INVX4
  INVX8
  LATCH
  N032X1
  N032X1
  OAI22X1
  OR2X2
  TBUFX1
  stimulus
Building instance overlay tables: ..... Done
Generating native code: .....
worklib.cpuvh <0x4dbb8656>
  streams:  0, words:  0
  worklib.stimulus:v <0x41cb96f>
  streams: 13, words: 17664
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
  Instances Unique
  Modules: 14499 34
  UDPs: 1628 4
  Primitives: 25424 6
  Timing outputs: 14499 19
  Timers: 1638 18
  Scalar wires: 16357 -
  Expanded wires: 46 5
  Always blocks: 1 1
  Initial blocks: 3 3
  Pseudo assignments: 9 9
  Total checks: 8769 1625
  Simulation timescale: 10ps
Writing initial simulation snapshot: worklib.AO121X1:v
Loading snapshot: worklib.AO121X1:v ..... Done
xcelium source /apps/cadence/XCELLUM1803/tools/xcelium/files/xmsimrc
xcelium run
Simulation complete via $finish() at time 501 NS + 0
./tb_cpu.v:30 # $finish;
xcelium -t 10ns
alukens@saturn.ece.iit.edu:~%
```

Figure 32: Carry-Skip Logic Synthesis Simulation

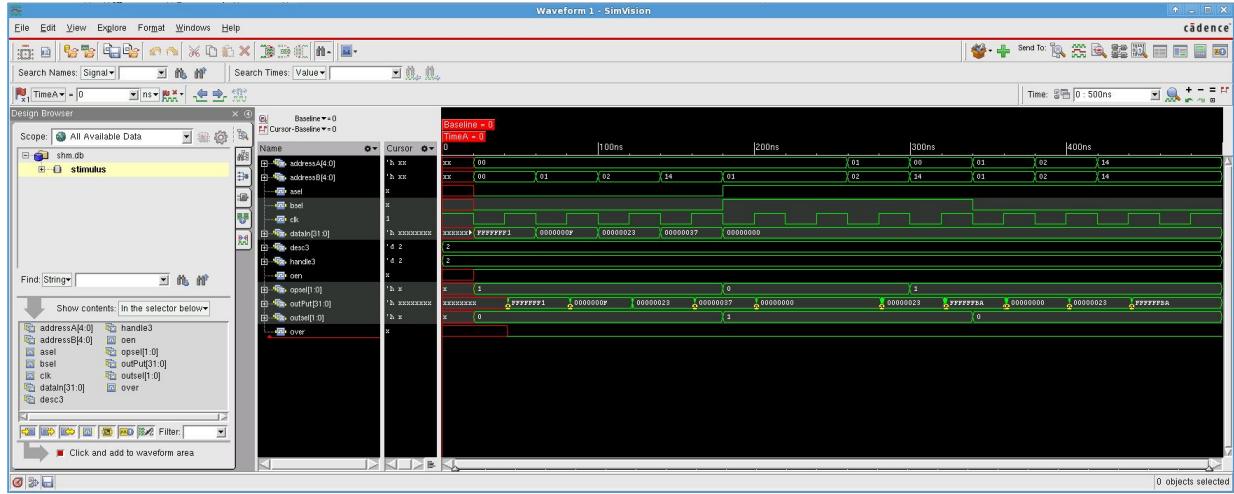


Figure 33: Carry-Skip Logic Synthesis SimVision Data

1.4.4 Physical Synthesis Timing Report

Report Interpretation After physical synthesis, the slack time of the Carry-Skip CPU design was found to be 18.318ns. This means that the period can be decreased by 18.318ns and the design will not violate any timing constraints.

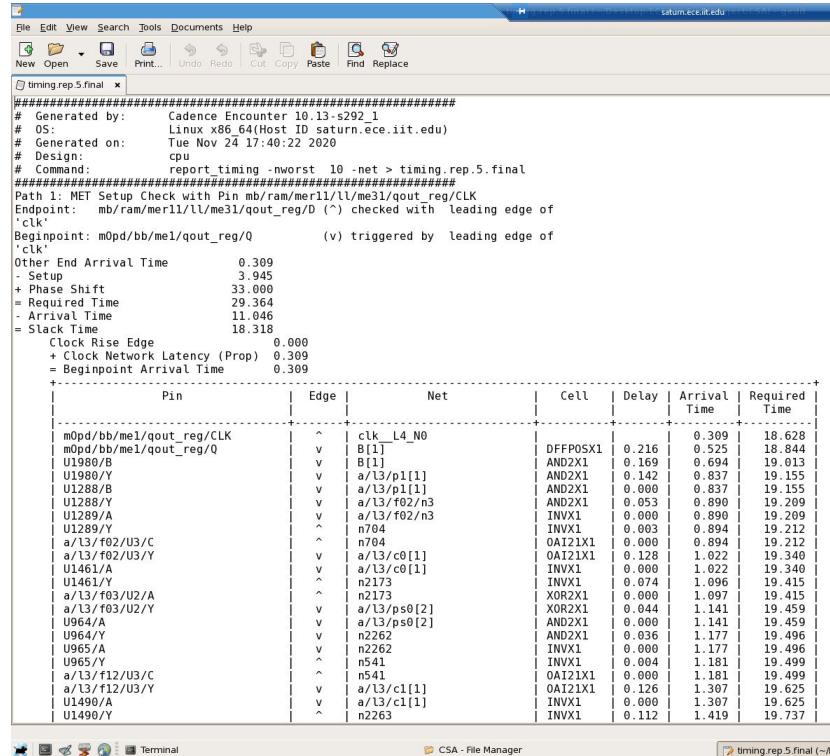


Figure 34: Carry-Skip Physical Synthesis Timing Report

Maximum Clock Frequency Estimation To calculate the maximum clock frequency, we must know the simulated frequency and the total slack time available at that frequency after physical synthesis. For the Carry-Skip adder design, the CPU was simulated at 30MHz, and the total slack time was found to be

18.318ns after physical synthesis. The period time for a CPU running at 30MHz is

$$T_c = \frac{1}{f} = 33.33\text{ns} \quad (7)$$

Knowing that the period time can be diminished by approximately 18.318ns, the maximum frequency is found to be

$$T_f = T_c - 18.318\text{ns} = 15.015\text{ns} \quad (8)$$

$$f = \frac{1}{T_f} = 66.598\text{MHz} \quad (9)$$

1.4.5 Post Physical Synthesis Simulation

```

Terminal
File Edit View Terminal Tabs Help
Building instance overlay tables: ..... Done
Generating native compiled code:
  worklib.cpu.v <0x10d12a8>
    streams: 0, words: 0
Sending signal (2) to pid: 5103
  Building instance specific data structures.
xrun: *E,ELBERR: Error during elaboration (status 245), exiting.
alukens@saturn.ece.iit.edu:~$ xrun gsc145nm.v tb.cpu.v final.v +access+r
xrun: 18.03-5001: (c) Copyright 1995-2018 Cadence Design Systems, Inc.
Caching library 'worklib' ..... Done
Elaborating the design hierarchy:
Top level design units:
  A0I2IX1
    BUF4X1
    CLKBUF1
    CLKBUF2
    CLKBUF3
    DFFBUF6X1
    DFFSR
    FAX1
    HAX1
    INVX2
    LATCH
    OR2X2
    NOR3X1
    OAI2ZX1
    OR2X2
    TBUFX1
    stimulus
Building instance overlay tables: ..... Done
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
  Instances Unique
  Modules: 14583 34
  UDPs: 1628 4
  Pin wires: 2538 6
  Timing outputs: 14583 19
  Registers: 1638 18
  Scalar wires: 16441 -
  Expanded wires: 46 5
  Always blocks: 1 1
  Initial blocks: 3 3
  Push/pop segments: 0 9
  Timing checks: 9769 1625
Simulation timescale: 10ps
Writing initial simulation snapshot: worklib.A0I2IX1:v
Loading snapshot worklib.A0I2IX1:v ..... Done
xcelium -source /apps/cadence/XCELUM1603/tools/xcelium/files/xmsimrc
xcelium -run
Simulation complete via $finish(1) at time 501 NS + 0
$tb.cpu.v:30 #1 $finish;
xcelium -exit
alukens@saturn.ece.iit.edu:~%

```

Figure 35: Carry-Skip Physical Synthesis Simulation

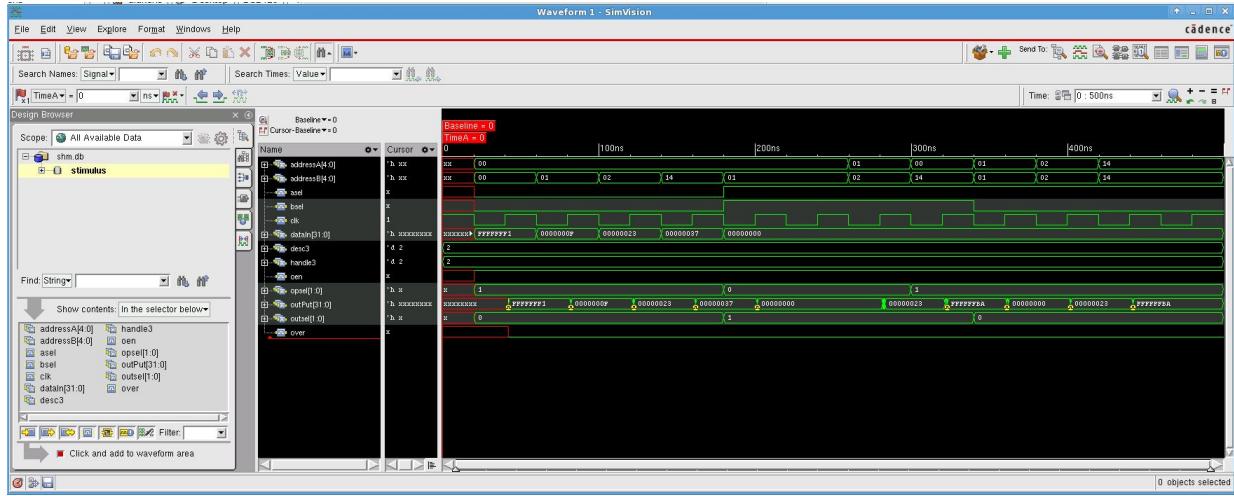


Figure 36: Carry-Skip Physical Synthesis SimVision Data

1.4.6 RTL Simulation with New Testbench

To perform the new testbench simulation, the runtime was increased to 1500ns in the testbench file. This was done by closing the simulation database after 1500ns (altered portion near top of testbench file).

```

Terminal
File Edit View Terminal Tabs Help
xmelab: ./CUWNSP (.cpu.CSA.v.584|11): 2 output ports were not connected:
xmelab: (.cpu.CSA.v.553): c.cover_1
xmelab: (.cpu.CSA.v.553): c.cover_2

Top level design units:
    stimulus
Building instance overlay tables: ..... Done
Generating native compiled code:
    worklib.mul:v <0x2531491e>
        streams: 32, words: 6720
    worklib.as32bv:<0x5d293101>
        streams: 2, words: 420
    worklib.as32bv:<0x5d293105>
        streams: 1, words: 1113
    worklib.aluv:<0x612a6b0b>
        streams: 1, words: 1246
    worklib.tristate32:v <0x71d5a1b7>
        streams: 0, words: 0
    worklib.xor32:v <0x612a6b09>
        streams: 0, words: 0
    worklib.mux2tol_32:v <0xd6d099bec>
        streams: 0, words: 0
    worklib.memoryblock:v <0x09ba1896>
        streams: 1, words: 288
    worklib.reg32:v <0x612a6b02>
        streams: 1, words: 264
    worklib.stimulus:v <0x19f0146c>
        streams: 13, words: 35977

Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
    Instances Unique
    Modules: 6575 31
    Primitives: 7347 6
    Registers: 2166 14
    Scalar wires: 3570 -
    Expanded Wires: 1198 41
    Aligned wires: 6 -
    Always blocks: 2156 4
    Initial blocks: 3 3
    Cont. assignments: 18 12
    Pseudos assignments: 43 43

Writing initial simulation snapshot: worklib.stimulus:v
Loading snapshot worklib.stimulus:v ..... Done
xcelium source /apps/cadence/XCELLUM1B03/tools/xcelium/files/xmsimrc
xcelium run
xmsim: *W,SHMPOPT: Some objects excluded from $shm_probe due to optimizations.
    File: ./tb.test.v, line = 28, pos = 11
    Scope: stimulus
    Time: 0 FS + 0

Simulation complete via $finish(1) at time 1501 NS + 0
./tb.test.v:30 #1 $finish;
xcelium exit
alukens@saturn.ece.iit.edu:~%
```

Figure 37: Carry-Skip New Testbench RTL Simulation

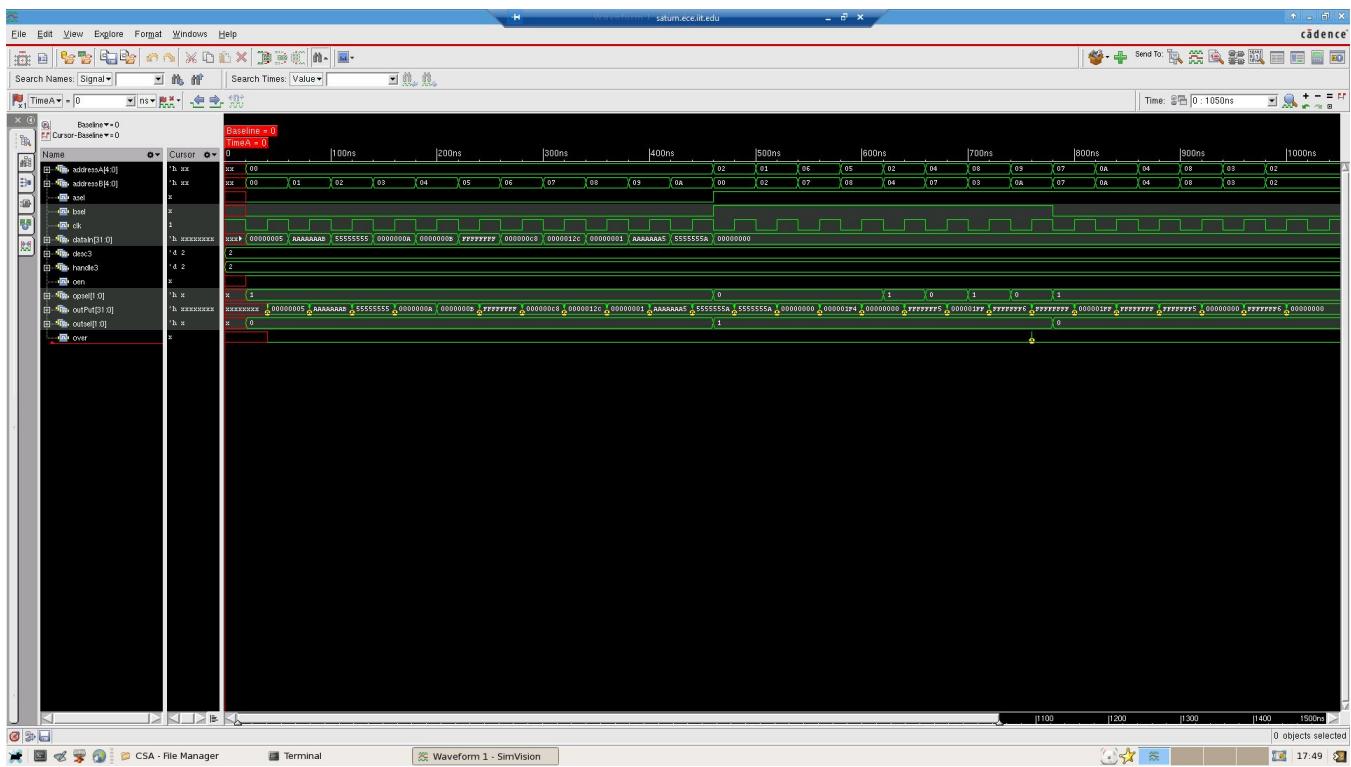


Figure 38: Carry-Skip New Testbench RTL SimVision Data

1.5 Carry-Select Adder Implementation

1.5.1 RTL Simulation

```
File Edit View Terminal Tabs Help
stimulus
Building instance overlay tables: ..... Done
Generating native compiled code:
    worklib.mul:v <0x2531491e>
        streams: 32, words: 6720
    worklib.CSeA_4:v <0x168f0294>
        streams: 1, words: 379
    worklib.as32b:v <0x5d203101>
        streams: 2, words: 420
    worklib.logic_fn:v <0x5afa3eb5>
        streams: 1, words: 1113
    worklib.alu:v <0x612a60b0>
        streams: 1, words: 1246
    worklib.tristate_32:v <0x71d5a1b7>
        streams: 0, words: 0
    worklib.mux2tol:v <0x3f664099>
        streams: 0, words: 0
    worklib.mux2tol_32:v <0x0d009bec>
        streams: 0, words: 0
    worklib.memoryBlock:v <0x09ba1896>
        streams: 1, words: 208
    worklib.dreg:v <0x60a2e9c9>
        streams: 2, words: 264
    worklib.stimulus:v <0x19f0146c>
        streams: 13, words: 17664
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
      Instances Unique
Modules:       6607   31
Primitives:    7507   6
Registers:    2166   14
Scalar wires:  3571   -
Expanded wires: 1230   49
Vectorized wires: 6   -
Always blocks: 2156   4
Initial blocks: 3   3
Cont. assignments: 9   11
Pseudo assignments: 43   43
Writing initial simulation snapshot: worklib.stimulus:v
Loading snapshot worklib.stimulus:v ..... Done
xcelium> source /apps/cadence/XCELIUM1803/tools/xcelium/files/xmsimrc
xcelium> run
xmsim: *W,SHMPOPT: Some objects excluded from $shm_probe due to optimizations.
  File: ./tb_cpu.v, line = 28, pos = 11
  Scope: stimulus
  Time: 0 FS + 0

Simulation complete via $finish(1) at time 501 NS + 0
./tb_cpu.v:30  #1 $finish;
xcelium> exit
alukens@saturn.ece.iit.edu:~%
```

Figure 39: Carry-Select RTL Simulation

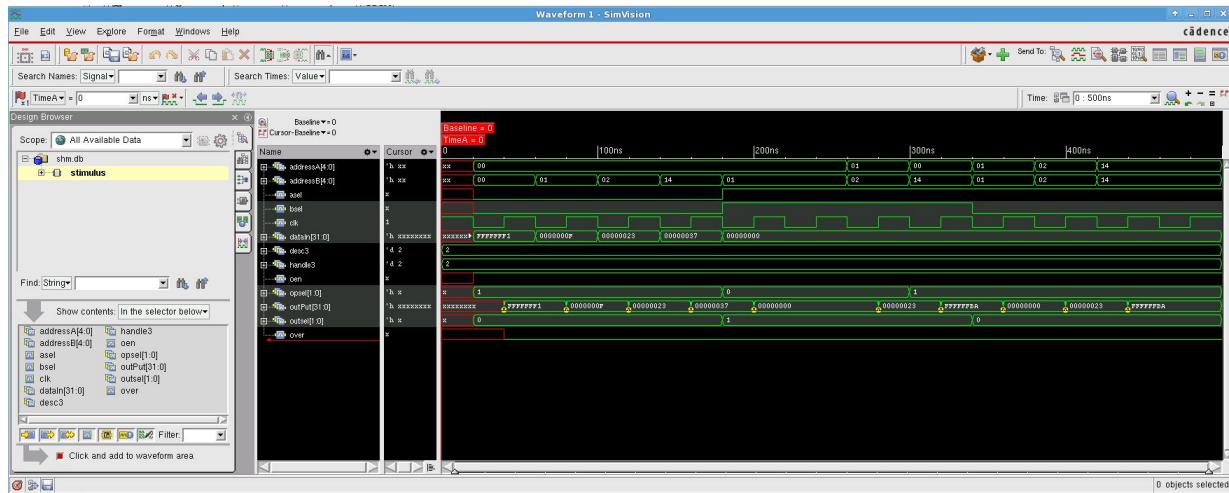


Figure 40: Carry-Select RTL SimVision Data

1.5.2 Logic Synthesis Timing & Cell Reports

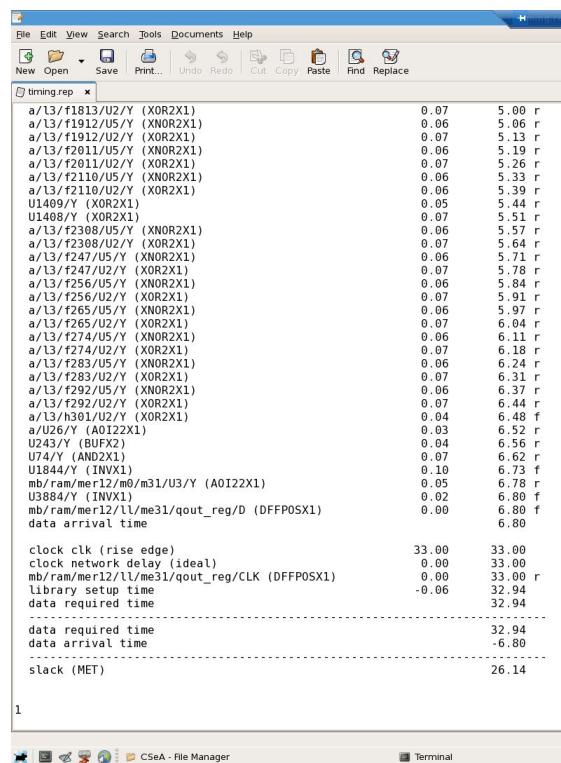


Figure 41: Carry-Select Logic Synthesis Timing Report

Figure 42: Carry-Select Logic Synthesis Cell Report

1.5.3 Post Logic Synthesis Simulation

```

File Edit View Terminal Tabs Help
alukens@saturn.ece.iit.edu:~% xrun gsc145nm.v tb.cpu.v cpu.vh +access+r
xrun: 18.03-5001: (c) Copyright 1995-2018 Cadence Design Systems, Inc.
Recompiling... Reason: unable to find module worklib.cpu:v (COD) <0x10d1b2a8>.
Caching library 'worklib': .... Done
Elaborating the design hierarchy:
Top level design units:
  A0T2I1X1
  BUFX4
  CLKBUF1
  CLKBUF2
  CLKBUF3
  CLKNFBGK1
  DFFP0
  FAX1
  HAX1
  INVX2
  INVX4
  INVX8
  INVX16
  INVX32
  NAND3X1
  NOR3X1
  OA12ZX1
  OR2X2
  TBUFX1
  TBUFX2
Building instance overlay tables: ..... Done
Generating native compiled code:
  worklib.cpu:v <0x722b6d1>
    streams: 0, words: 0
  worklib.stimulus:v <0x416cb96>
    streams: 13, words: 17664
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
  Instances Unique
  Modules: 14598 34
  Wires: 1625 4
  Primitives: 2528 6
  Timing outputs: 14598 19
  Registers: 1638 18
  Scalar wires: 16456 -
  Expanded wires: 46 5
  Always blocks: 1 1
  Pseudo assignments: 3 3
  Pseudo assignments: 9 9
  Timing checks: 9769 1625
  Simulation timescale: 10ps
Writing initial simulation snapshot: worklib.A0T2I1X1:v
Loading snapshot worklib.A0T2I1X1:v ..... Done
xcelium source /apps/cadence/XCELLUM1B03/tools/xcelium/files/xmsimrc
xcelium -run
Simulation complete via $finish() at time 501 NS + 0
./tb.cpu.v-30 #1 $finish;
xcelium -exit
alukens@saturn.ece.iit.edu:~%

```

Figure 43: Carry-Select Logic Synthesis Simulation

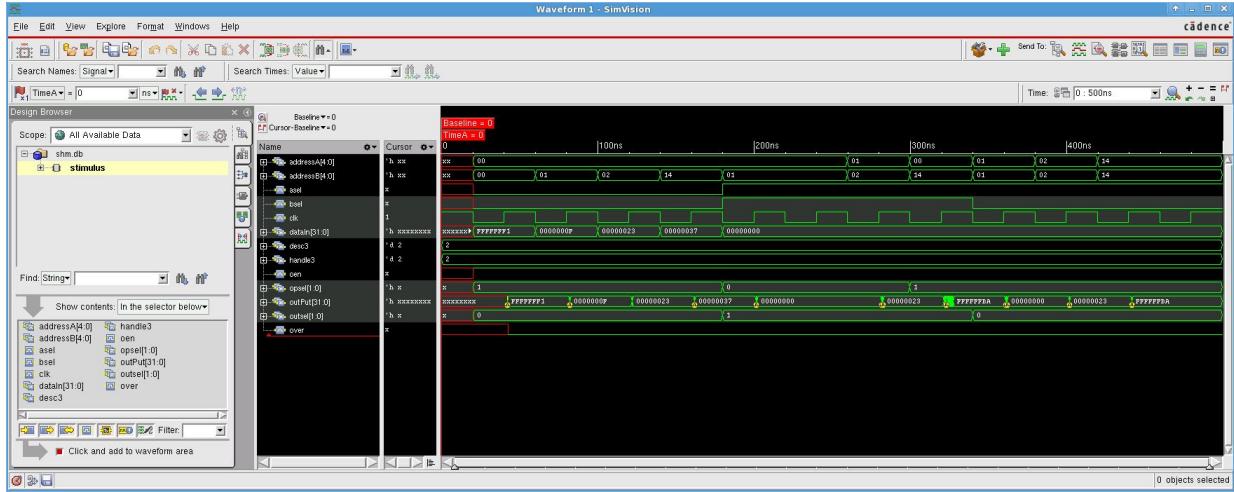


Figure 44: Carry-Select Logic Synthesis SimVision Data

1.5.4 Physical Synthesis Timing Report

Report Interpretation After physical synthesis, the slack time of the Carry-Select CPU design was found to be 18.144ns. This means that the period can be decreased by 18.144ns and the design will not violate any timing constraints.

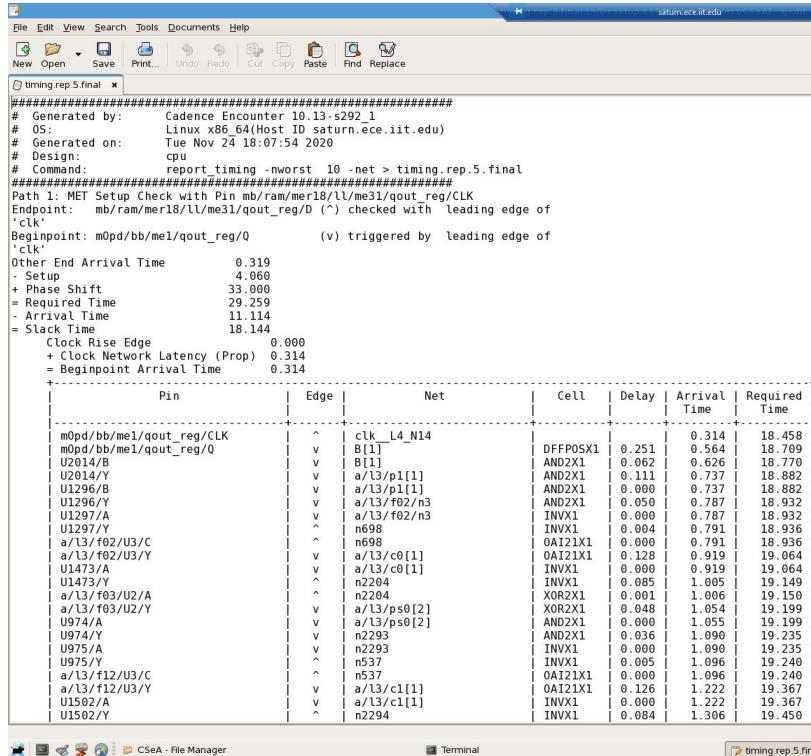


Figure 45: Carry-Select Physical Synthesis Timing Report

Maximum Clock Frequency Estimation To calculate the maximum clock frequency, we must know the simulated frequency and the total slack time available at that frequency after physical synthesis. For the Carry-Select adder design, the CPU was simulated at 30MHz, and the total slack time was found to be

18.144ns after physical synthesis. The period time for a CPU running at 30MHz is

$$T_c = \frac{1}{f} = 33.33ns \quad (10)$$

Knowing that the period time can be diminished by approximately 18.144ns, the maximum frequency is found to be

$$T_f = T_c - 18.144ns = 15.5189ns \quad (11)$$

$$f = \frac{1}{T_f} = 65.835MHz \quad (12)$$

1.5.5 Post Physical Synthesis Simulation

```

Terminal
File Edit View Terminal Tabs Help
Elapsed Time = 3:31.79, CPU Time = 192.117
alukens@saturn.ece.illinois.edu:~$ xrun gsc145nm.v tb.cpu.v final.v +access+r
xrun: 18.03-500: (c) Copyright 1995-2018 Cadence Design Systems, Inc.
file: final.v
module worklib.cpuv
  errors: 0, warnings: 0
    Caching library 'worklib' ..... Done
  Elaborating the design hierarchy:
  Top level design units:
    A0I21X1
    MUX2X1
    CLKBUF1
    CLKBUF2
    CLKBUF3
    DFNFNGX1
    DFFSR
    FAX1
    INV1
    INVX2
    INVX4
    LATCH
    MUX2X1
    NAND3X1
    NOT1X1
    OR42X1
    OR2X2
    TBIFX1
    stimulus
  Building instance overlay tables: ..... Done
  Generating native compiled code:
    worklib.A0I21X1.vf2>
      streams: 0, words: 0
  Building instance specific data structures.
  Loading native compiled code: ..... Done
  Design hierarchy summary:
    Instances Unique
    Modules: 14689 34
    UDPs: 1638 4
    Primitives: 25730 6
    Timing outputs: 14689 19
    Registers: 1638 18
    Scalar wires: 16538 -
    Expanded blocks: 46 5
    Alias blocks: 1 1
    Initial blocks: 3 3
    Pseudo assignments: 9 9
    Timing checks: 9769 1625
  Simulation timescale: 10ps
  Writing initial simulation snapshot: worklib.A0I21X1:v
  Loading snapshot worklib.A0I21X1:v ..... Done
  xcelium> source /apps/cadence/XCELUM1803/tools/xcelium/files/xmsimrc
  xcelium> run
  Simulation complete via $finish(1) at time 501 NS + 0
  ./tb.cpu.v:30 # $finish;
  xcelium> exit
alukens@saturn.ece.illinois.edu:~%

```

Figure 46: Carry-Select Physical Synthesis Simulation

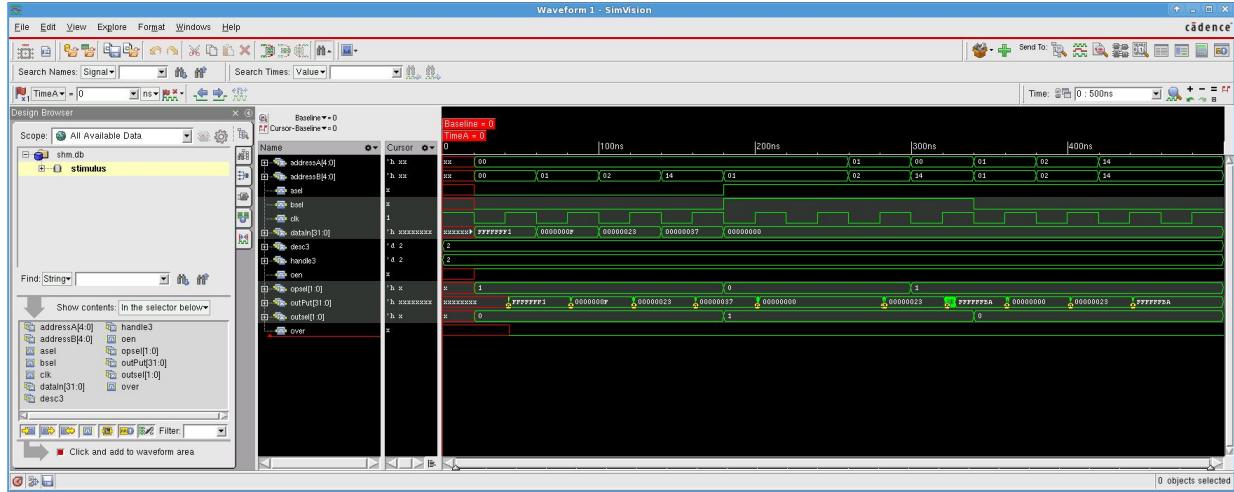


Figure 47: Carry-Select Physical Synthesis SimVision Data

1.5.6 RTL Simulation with New Testbench

To perform the new testbench simulation, the runtime was increased to 1500ns in the testbench file. This was done by closing the simulation database after 1500ns (altered portion near top of testbench file).

```

Terminal
File Edit View Terminal Tabs Help
xmeLab: (./cpu_CSeA.v,553): over
CSeA_4 c3(.s{11:8}), .cout(c[3]), .a(a[11:8]), .b(b[11:8]), .cin(c[2]));
| 
xmeLab: "W,CUWSP (./cpu_CSeA.v,593): 1 output port was not connected;
xmeLab: (./cpu_CSeA.v,553): over

Top level design units:
    stimulus
Building instance overlay tables: ..... Done
Generating native compiled code:
    worklib:buf1:v<0x5a0>
        streams: 32, words: 6720
    worklib:as32v:<0x5a2b3101>
        streams: 2, words: 420
    worklib:logic_fnv:<0x5afab3e05>
        streams: 1, words: 1113
    worklib:alut:<0xd6a6d06>
        streams: 1, words: 1246
    worklib:tristate_32v:<0x71d5a1b7>
        streams: 0, words: 0
    worklib:mux2to1v:<0x3f664099>
        streams: 0, words: 0
    worklib:mux2to1_32v:<0xb0d09b9c>
        streams: 0, words: 0
    worklib:multyblock:<0x5a1b906>
        streams: 1, words: 288
    worklib:dregiv:<0x60a2e9c>
        streams: 2, words: 264
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary
          Instances Unique
Modules:           6607   31
Primitives:        7507   6
Registers:         2166   14
Scalar wires:      3571   -
Explored wires:   1230   49
Encountered wires: 6   -
Always blocks:    2156   4
Initial blocks:    3   3
Cont. assignments: 9   11
Pseudo assignments: 43   43
Writing initial simulation snapshot: worklib.stimulus:v..... Done
Loading snapshot worklib.stimulus:v..... Done
xcelium> source /apps/cadence/XCELIUM03/tools/xcelium/files/xmsimrc
xcelium> run
xmsim: *W_SHMPOPT: Some objects excluded from $shm_probe due to optimizations.
      File: ./tb_test.v, line = 28, pos = 11
      Scope: stimulus
      Time: 0 FS + 0
Simulation complete via $finish() at time 1501 NS + 0
./tb_test.v:30 # $finish;
xcelium> exit
alukens@saturn.ece.iit.edu:~%
```

Figure 48: Carry-Select New Testbench RTL Simulation

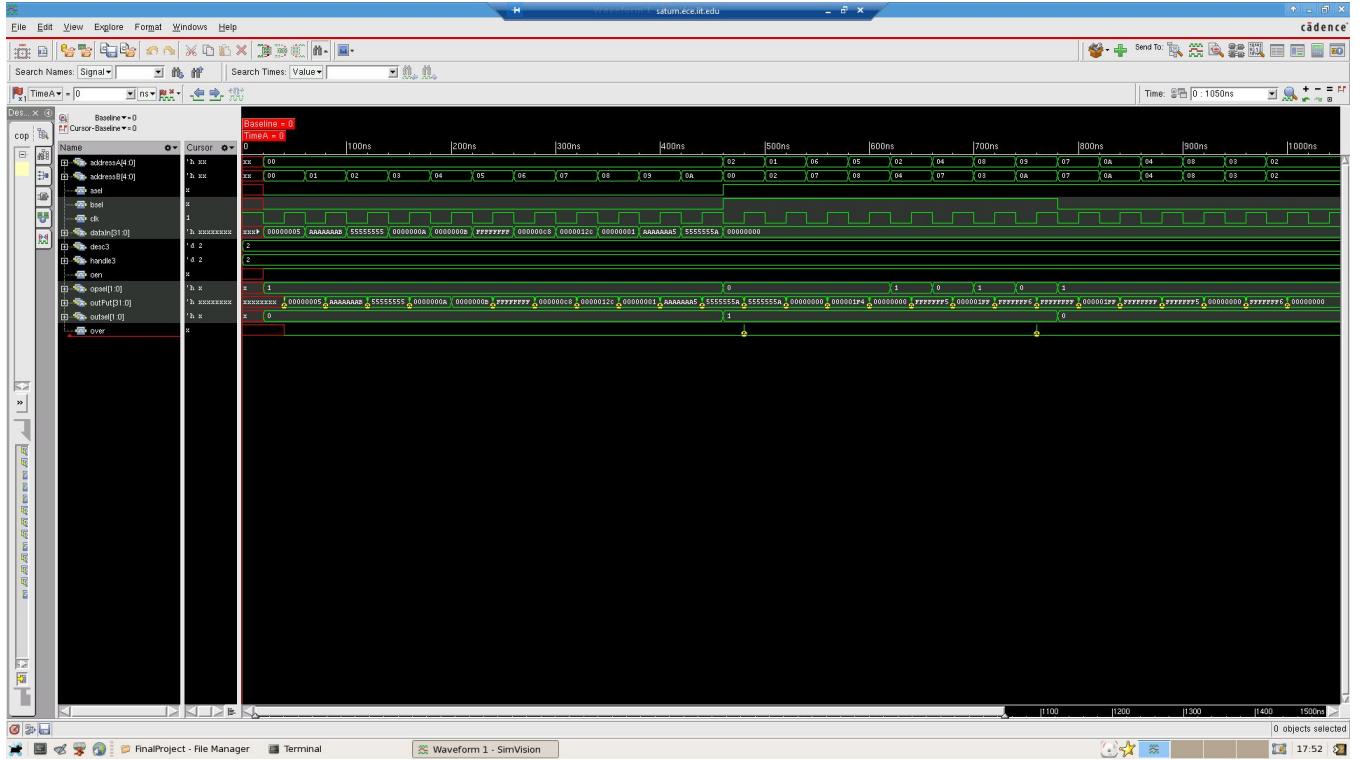


Figure 49: Carry-Select New Testbench RTL SimVision Data

1.6 Deliverables

The results of the new testbench simulations are tabulated below. The stored data shows the values stored in the 2nd operand memory location at the completion of the instruction. The arrival times shown in the table are in nanoseconds. Based on the calculations performed, the fastest adder design was found to be the Carry-Lookahead adder. Despite having some scenarios where the delay was longer, the average arrival time was found to be smaller than the other designs.

	Instruction	Stored Data	CRA	CLA	CSA	CSeA
Calculate the Path Delay for Each Operation (Post-Synthesis Gate- Level Delay)	ADD[2][0]	5555_555A	3.19	2.96	3.74	3.42
	ADD[1][2]	0000_0000	19.97	16.96	9.4	10.25
	ADD[6][7]	0000_01F4	4.6	4.75	5.15	4.14
	ADD[5][8]	0000_0000	20.9	17.5	10.33	10.61
	SUB[2][4]	FFFF_FFF5	15.673	18.27	8.04	7.04
	ADD[4][7]	0000_01FF	1.95	2.08	1.95	2.52
	SUB[8][3]	FFFF_FFF6	1.94	2.24	8.35	3.55
	ADD[9][10]	FFFF_FFFF	2.22	2.06	3.45	2.44

Figure 50: New Testbench Arrival Data

1.7 Conclusions

The first case study should be considered a success. Students were successfully able to implement various adder implementations into the CPU design, and test the functionality at multiple stages in the ASIC design flow. Students were also successful in elaborating on a verilog testbench design to test additional functionality of the CPU design. This allows the verilog design to be tested rigorously before being sent to

a silicon fabrication facility to be produced. It is important that the design is tested comprehensively before being sent to the fabrication facility, because once the design is implemented in silicon, it is near impossible to correct errors.

In the future, students will utilize the ASIC design flow to develop more complex circuits using verilog, perform logic synthesis on these designs, and place and route these designs. The skills obtained in this laboratory are not limited to any single design process, which allows students to adapt these abilities as the situation requires.

2 Case Study - ALU Comparator Design

In the second case study, students will elaborate on the CPU architecture introduced in the first case study. Specifically, students will be required to write new Verilog modules to implement a single bit comparator, and a 4-to-2 bit multiplexer. These modules will then be instantiated to create a 32-bit comparator for use in the ALU of the CPU design explored previously. Students will then elaborate on a Verilog testbench in order to test the new ALU capabilities, and adjust parameters of the testbench file as needed.

2.1 Theory

Hardware Description Language When using the ASIC design flow, it is important to add all required functionality to a design at the hardware description language (HDL) level. A common language to implement such functionality is Verilog. In Verilog, a design can be separated into individual design "modules". This allows each individual portion of a design to be implemented and verified separately. This ensures that each subsection of a design will function correctly on its own, and ensures that the design will function as intended when combined into a finalized design.

Modular Design Additionally, individual modules should be very low level. For example, when creating a 32-bit design, it is important to create a smaller module that incorporates the functionality on a smaller dataset. This is recommended, because a larger module can be created by instantiating several smaller modules and having a smaller module makes it easy to instantiate a larger design by using a varying amount of the smaller modules.

Floorplanning An important stage in improving the performance of a VLSI design is optimizing the physical layout of the design. This process, floorplanning, involves managing the physical position of various cells/modules on the layout of the VLSI design. For lower performance designs and proof-of-concepts, automatic floorplanning can be performed by the logic and physical synthesis CAD tools. For more advanced/high-performance designs, manual floorplanning is often necessary. In manual floorplanning, priority placement is given to modules that require complex interconnections to other modules. In other words, the designer manually specifies the location of critical modules such that the critical path delay is minimized, increasing performance. Other, non-critical modules may still be placed automatically.

2.2 Single Bit Comparator Design

In order to implement a 32-bit comparator in the ALU of the specified CPU, it is first important to create a single-bit comparator module. This will allow designers in the future to create comparators of various size by instantiating varying amounts of single-bit comparators.

The single-bit comparator takes in two input bits (A and B), and returns two output values (f1 and f0). If A is greater than B, the comparator should output f1=0 and f0=1. If A is less than B, the comparator should output f1=0 and f0=0. If A is equal to B, the comparator should output f1=1 and f0=0.

```

//one bit comparator
module one_bit_comp(a, b, f1, f0);
input a, b;
output reg f1, f0;

//if (a > b) then {f1, f0} = 2'b01
//if (a < b) then {f1, f0} = 2'b00
//if (a == b) then {f1, f0} = 2'b10

always@(*) begin
    if(a>b) begin          //if a>b, output 01;
        f1=1'b0;
        f0=1'b1;
    end
    if(a<b) begin          //if a<b output 00;
        f1=1'b0;
        f0=1'b0;
    end
    if(a==b) begin          //if a==b output 10;
        f1=1'b1;
        f0=1'b0;
    end
end

```

Figure 51: Verilog Model of Single-Bit Comparator

2.3 4-to-2 Multiplexer Design

Another module important to the 32-bit comparator design is the 4-to-2 multiplexer. The output of this multiplexer is dependent on the value of hi_f1. If hi_f1=0, then the outputs should be consistent with the hi_f1 and hi_f0 inputs. Otherwise, the results should be dependent on the lo_f1 and lo_f0 inputs.

```

//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 reg f1, f0;

//use hi_f1 to select the correct outputs

always@(*) begin
    if(hi_f1==1'b0) begin
        f1=hi_f1;
        f0=hi_f0;
    end
    else begin
        f1=lo_f1;
        f0=lo_f0;
    end
end
//write your code here

```

Figure 52: Verilog Model of 4-to-2 Multiplexer

2.4 32-bit Comparator Design

Next, several instances of the single-bit comparator and the 4-to-2 multiplexer can be instantiated in order to implement the full 32-bit comparator design. Verilog generate statements can be used to simplify the instantiation of the comparators and multiplexers.

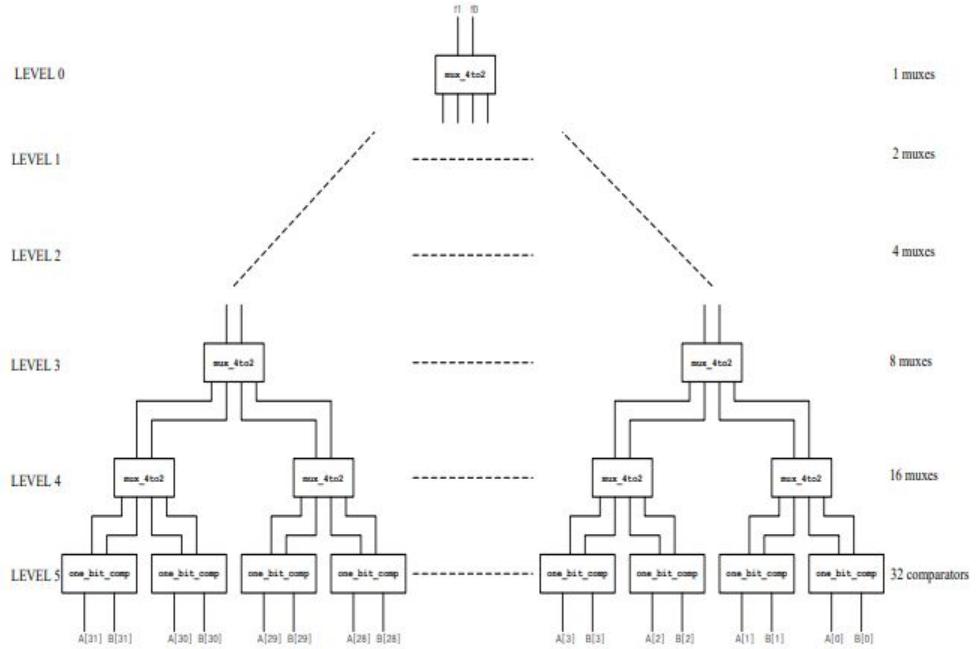


Figure 53: Schematic Diagram of 32-bit Comparator

```

//write your code here
genvar i;
//Level 5: 32 one_bit_comp go here
generate
    for(i=0; i<32; i=i+1) begin
        one_bit_comp i0(A[i],B[i],f1_L5[i],f0_L5[i]); //generate 32 one bit comparators
    end
endgenerate

//Level 4: 16 mux_4to2 go here
generate
    for(i=0; i<16; i=i+1) begin
        mux_4to2 i0(f1_L5[2*i+1], f0_L5[2*i+1], f1_L5[2*i], f0_L5[2*i], f1_L4[i], f0_L4[i]); //generate 16 4to2 muxes, connect L5 inputs
    end
endgenerate

//Level 3: 8 mux_4to2 go here
generate
    for(i=0; i<8; i=i+1) begin
        mux_4to2 i0(f1_L4[2*i+1], f0_L4[2*i+1], f1_L4[2*i], f0_L4[2*i], f1_L3[i], f0_L3[i]); //generate 8 4to2 muxes, connect L4 inputs
    end
endgenerate

//Level 2: 4 mux_4to2 go here
generate
    for(i=0; i<4; i=i+1) begin
        mux_4to2 i0(f1_L3[2*i+1], f0_L3[2*i+1], f1_L3[2*i], f0_L3[2*i], f1_L2[i], f0_L2[i]); //generate 4 4to2 muxes, connect L3 inputs
    end
endgenerate

//Level 1: 2 mux_4to2 go here
generate
    for(i=0; i<2; i=i+1) begin
        mux_4to2 i0(f1_L2[2*i+1], f0_L2[2*i+1], f1_L2[2*i], f0_L2[2*i], f1_L1[i], f0_L1[i]); //generate 2 4to2 muxes, connect L2 inputs
    end
endgenerate

//Level 0: 1 mux_4to2 goes here
    mux_4to2 i0(f1_L1[1], f0_L1[1], f1_L1[0], f0_L1[0], f1, f0); //generate 1 4to2 muxes, connect L1 inputs. Set outputs to f1 and f0

endmodule

```

Figure 54: Verilog Model of 32-bit Comparator

2.5 New Testbench for Verifying Functionality of Comparator

A new testbench must be developed to ensure the complete functionality of the comparator design. First, to become compliant with the new CPU specification, the size of the OUTSEL register must be altered in the Verilog testbench. Because the ALU mux has expanded to incorporate the 32-bit comparator, the OUTSEL

register must be changed from [1:0] size to [2:0] size. This means that this change must be reflected in each instruction utilized in the testbench.

2.6 RTL Simulation of CPU with Comparator

```

saturn.ece.iit.edu - Remote Desktop Connection
File Edit View Terminal Help
Building instance overlay tables: ..... Done
Generating native compiled code:
    worklib.stimulus:v <0x49aa807>
        streams: 13, words: 42094
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
    Instances Unique
    Modules: 6672 37
    Primitives: 7187 6
    Registers: 2293 18
    Scalar Wires: 3819 -
    Expanded Wires: 1199 41
    Vectorized Wires: 7 -
    Always blocks: 2220 6
    Initial blocks: 3 3
    Constant assignments: 12 16
    Pseudo assignments: 43 43
Writing initial simulation snapshot: worklib.stimulus:v
Loading snapshot worklib.stimulus:v ..... Done
xcelium> source /apps/cadence/XCELUM1B03/tools/xcelium/files/xmsimrc
xcelium> run
xmsim: *W,$HMPOPT: Some objects excluded from $shm_probe due to optimizations.
    File: ./tb_test_comp.v, line = 29, pos = 11
    Scope: stimulus
    Time: 0 FS + 0

Please check Select Lines!
Simulation complete via $finish(1) at time 1501 NS + 0
./tb_test_comp.v:31 #1 $finish;
xcelium> exit
alukens@saturn.ece.iit.edu:~% simvision
simvision: 18.03-s001: (c) Copyright 1995-2018 Cadence Design Systems, Inc.
txe: 18.03-s001: (c) Copyright 1995-2020 Cadence Design Systems, Inc.

-----
SimVision received an "interrupt" (SIGINT) signal such as a Ctrl-C and
had to exit.

-----
alukens@saturn.ece.iit.edu:~% xrun tb_test_comp.v cpu_comp.v +access+r
xrun: 18.03-s001: (c) Copyright 1995-2018 Cadence Design Systems, Inc.
Loading snapshot worklib.stimulus:v ..... Done
xcelium> source /apps/cadence/XCELUM1B03/tools/xcelium/files/xmsimrc
xcelium> run
xmsim: *W,$HMPOPT: Some objects excluded from $shm_probe due to optimizations.
    File: ./tb_test_comp.v, line = 29, pos = 11
    Scope: stimulus
    Time: 0 FS + 0

Please check Select Lines!
Simulation complete via $finish(1) at time 1501 NS + 0
./tb_test_comp.v:31 #1 $finish;
xcelium> exit
alukens@saturn.ece.iit.edu:~%

```

Figure 55: RTL Simulation of CPU with Comparator Result

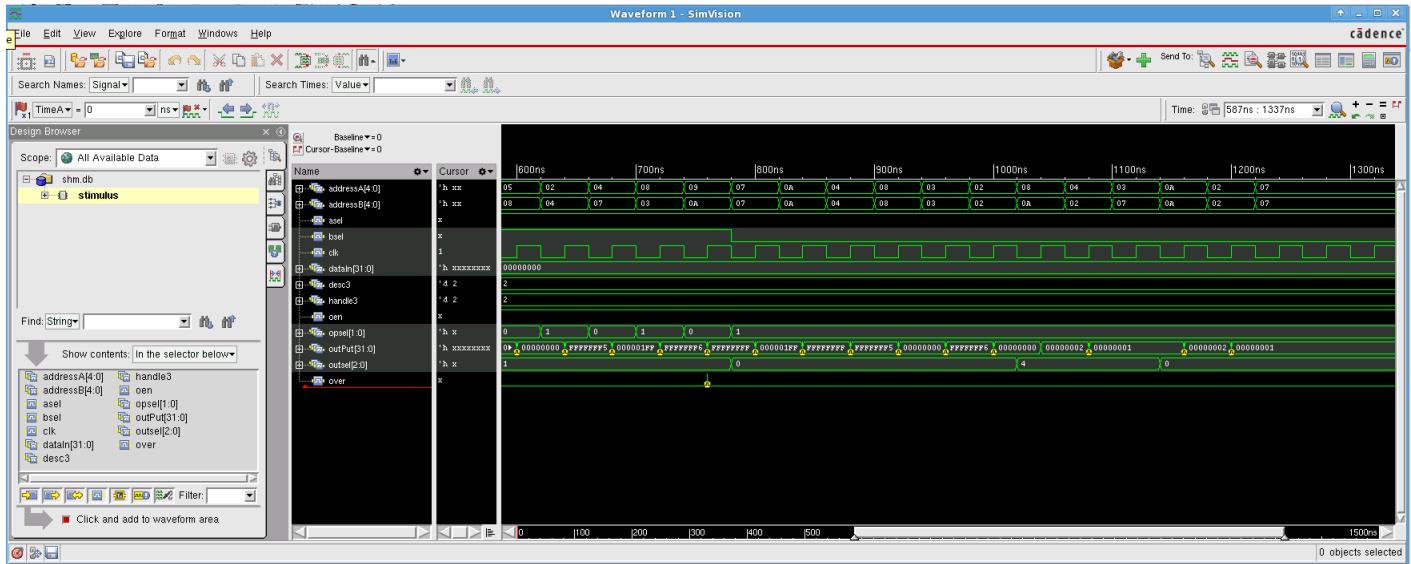


Figure 56: RTL Simulation of CPU with Comparator Simvision Data

The results of the "CMP" instructions begin at t=1020ns in the screenshot of SimVision.

The first compare instruction is CMP[8][10]. At this point, the contents of [8]=0000_0000 and [10]=FFFF_FFFF. Because the comparator does not know signed comparison, the result is found that the 1st operand is less than the 2nd operand, which is signified by the output 32'b10 = 2.

The next compare instruction is CMP[4][2], where [4]=FFFF_FFF5 and [2]=0000_0000. The result of this comparison is that the 1st operand is greater than the 2nd operand, which is signified by the output 32'b01 = 1.

The final compare instruction is CMP[3][7], where [3]=FFFF_FFF6 and [7]=0000_01FF. The result of this comparison is that the 1st operand is greater than the 2nd operand, which is signified by the output 32'b01 = 1.

The subsequent read instructions uphold these results, as the result of each compare operation overwrites the 2nd operands memory location. The read instructions show that [10]=0000_0002, [2]=0000_0001, and [7]=0000_0001 as expected.

2.7 Logic Synthesis Timing & Cell Reports

```

saturn.ece.iit.edu - Remote Desktop Connection
timing.rep (~/Desktop/ECE429/Fi...
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
compile_dc.tcl timing.rep

a/l3/f1516/U2/Y (XOR2X1) 0.07 4.61 r
a/l3/f1615/U5/Y (XNOR2X1) 0.06 4.67 r
a/l3/f1615/U2/Y (XOR2X1) 0.07 4.74 r
a/l3/f1714/U5/Y (XNOR2X1) 0.06 4.80 r
a/l3/f1714/U2/Y (XOR2X1) 0.07 4.87 r
a/l3/f1813/U5/Y (XNOR2X1) 0.06 4.93 r
a/l3/f1813/U2/Y (XOR2X1) 0.07 5.01 r
a/l3/f1912/U5/Y (XNOR2X1) 0.06 5.07 r
a/l3/f1912/U2/Y (XOR2X1) 0.07 5.14 r
a/l3/f2011/U5/Y (XNOR2X1) 0.06 5.20 r
a/l3/f2011/U2/Y (XOR2X1) 0.07 5.27 r
a/l3/f2110/U5/Y (XNOR2X1) 0.06 5.33 r
a/l3/f2110/U2/Y (XOR2X1) 0.06 5.40 r
U1348/(XOR2X1) 0.05 5.45 r
U1347/(XOR2X1) 0.07 5.52 r
a/l3/f2308/U5/Y (XNOR2X1) 0.06 5.58 r
a/l3/f2308/U2/Y (XOR2X1) 0.07 5.65 r
a/l3/f247/U5/Y (XNOR2X1) 0.06 5.71 r
a/l3/f247/U2/Y (XOR2X1) 0.07 5.78 r
a/l3/f256/U5/Y (XNOR2X1) 0.06 5.85 r
a/l3/f256/U2/Y (XOR2X1) 0.07 5.92 r
a/l3/f265/U5/Y (XNOR2X1) 0.06 5.98 r
a/l3/f265/U2/Y (XOR2X1) 0.07 6.05 r
a/l3/f274/U5/Y (XNOR2X1) 0.06 6.11 r
a/l3/f274/U2/Y (XOR2X1) 0.07 6.18 r
a/l3/f283/U5/Y (XNOR2X1) 0.06 6.25 r
a/l3/f283/U2/Y (XOR2X1) 0.07 6.32 r
a/l3/f292/U5/Y (XNOR2X1) 0.06 6.38 r
a/l3/f292/U2/Y (XOR2X1) 0.05 6.43 f
U1783/(AND2X1) 0.03 6.46 f
a/l3/rc31/qout_reg/D (DFFP0SX1) 0.00 6.46 f
data arrival time 6.46

clock clk (rise edge) 33.00 33.00
clock network delay (ideal) 0.00 33.00
a/l3/rc31/qout_reg/CLK (DFFP0SX1) 0.00 33.00 r
library setup time -0.06 32.94
data required time 32.94

data required time 32.94
data arrival time -6.46

slack (MET) 26.48

```

Figure 57: Logic Synthesis of CPU with Comparator Timing Report

saturn.ece.iit.edu - Remote Desktop Connection

File Edit View Search Tools Documents Help

New Open Save Print... Undo Redo Cut Copy Paste Find Replace

cell rep x

cell rep

mn0pn/oe/qout_reg	DFFP0SX1	gsc145nm	7.978100	n
mn0pn/ops/m0/qout_reg	DFFP0SX1	gsc145nm	7.978100	n
mn0pn/ops/m1/qout_reg	DFFP0SX1	gsc145nm	7.978100	n
mn0pn/outs/m0/qout_reg	DFFP0SX1	gsc145nm	7.978100	n
mn0pn/outs/m1/qout_reg	DFFP0SX1	gsc145nm	7.978100	n
mn0pn/outs/m2/qout_reg	DFFP0SX1	gsc145nm	7.978100	n
o/tr/t0/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t1/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t2/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t3/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t4/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t5/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t6/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t7/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t8/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t9/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t10/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t11/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t12/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t13/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t14/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t15/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t16/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t17/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t18/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t19/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t20/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t21/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t22/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t23/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t24/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t25/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t26/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t27/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t28/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t29/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t30/bl	TBUFX2	gsc145nm	3.754400	n
o/tr/t31/bl	TBUFX2	gsc145nm	3.754400	n
wb/bd/m0/qout_reg	DFFP0SX1	gsc145nm	7.978100	n
wb/bd/mel/qout_reg	DFFP0SX1	gsc145nm	7.978100	n
wb/bd/m2/qout_reg	DFFP0SX1	gsc145nm	7.978100	n
wb/bd/m3/qout_reg	DFFP0SX1	gsc145nm	7.978100	n
wb/bd/m4/qout_reg	DFFP0SX1	gsc145nm	7.978100	n

Total 14679 cells 49388.192473

1

Figure 58: Logic Synthesis of CPU with Comparator Cell Report

2.8 Post Logic Synthesis Simulation

```

Terminal
File Edit View Terminal Tabs Help
module worklib.cpu:vh
    errors: 0, warnings: 0
    Caching library 'worklib' ..... Done
Elaborating the design hierarchy:
Top level design units:
    BUF4X
    CLKBUF1
    CLKBUF2
    CLKBUF3
    DFFNEGX1
    DFFSR
    FAX1
    HAX1
    INVX2
    INVX4
    INVX8
    MUX2X1
    OA122X1
    OR2X2
    TBUFX1
    stimulus
Building instance overlay tables: ..... Done
Generating native compiled code:
    worklib.LATCH:v <0x60600d13>
        streams: 0, words: 0
    worklib.cpu:vh <0x435a093d>
        streams: 0, words: 0
    worklib.stimulus:v <0x1362d78>
        streams: 13, words: 42094
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
          Instances Unique
Modules:           14696   34
UDPs:             1660    4
Primitives:       25673   6
Timing outputs:   14696   19
Registers:        1670   18
Scalar wires:     16586   -
Expanded wires:   47      5
Always blocks:    1       1
Initial blocks:   3       3
Pseudo assignments: 9      9
Timing checks:   9930   1659
Simulation timescale: 10ps
Writing initial simulation snapshot: worklib.BUF4X:v
Loading snapshot worklib.BUF4X:v ..... Done
xcelium> source /apps/cadence/XCELUM1803/tools/xcelium/files/xmsimrc
xcelium> run
Simulation complete via $finish(1) at time 1501 NS + 0
./rtb_test_comp.v:31 #1 $finish;
xcelium> exit
alukens@saturn.ece.iit.edu:~%

```

Figure 59: CPU with Comparator Post Logic Synthesis Simulation

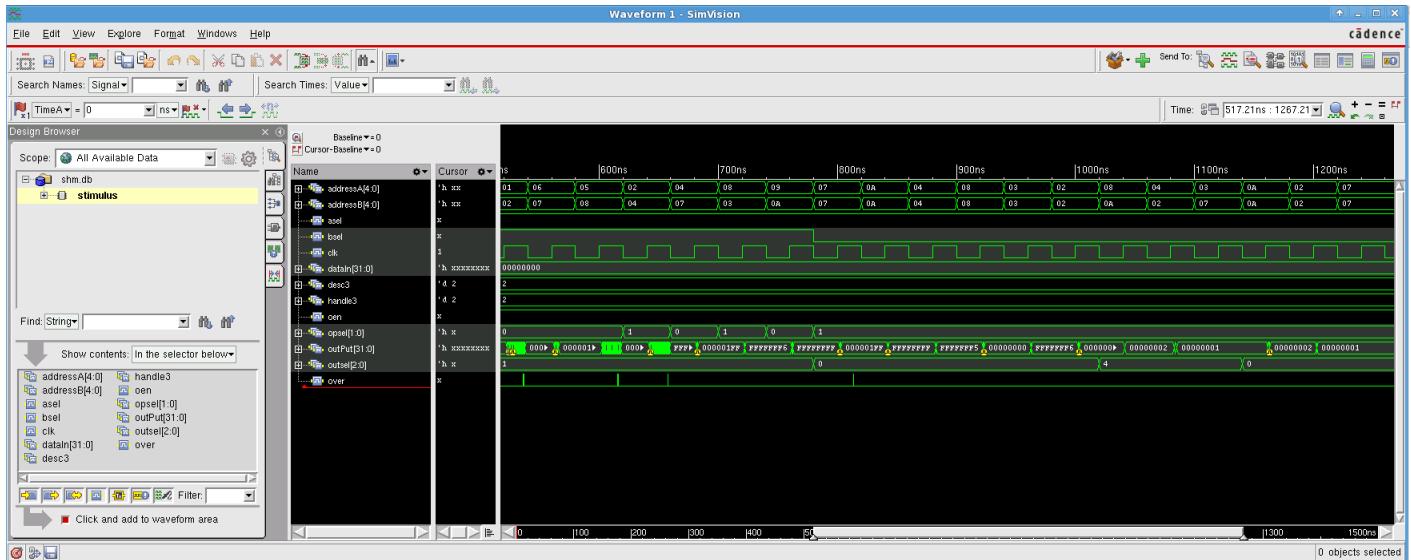


Figure 60: CPU with Comparator Post Logic Synthesis Simvision Data

2.9 Physical Synthesis Timing Report

Report Interpretation After physical synthesis, the slack time of the CPU with Comparator design was found to be 20.078ns. This means that the period can be decreased by 20.078ns and the design will not violate any timing constraints

```

saturn.ece.iit.edu - Remote Desktop Connection
timing.rep.5.final (~/Desktop/ECE429/FinalProject/Case2Stuff) - gedit

#####
# Generated by: Cadence Encounter 10.13-s292_1
# OS: Linux x86_64(Host ID saturn.ece.iit.edu)
# Generated on: Tue Nov 24 22:16:48 2020
# Design: cpu
# Command: report_timing -nworst 10 -net > timing.rep.5.final
#####
Path 1: MET Setup Check with Pin a/13/rC31/qout_reg/CLK
Endpoint: a/13/rC31/qout_reg/D ('') checked with leading edge of 'clk'
Beginpoint: mOpd/bb/mel/qout_reg/Q (v) triggered by leading edge of 'clk'
Other End Arrival Time 0.308
- Setup 3.298
+ Phase Shift 33.000
= Required Time 30.010
- Arrival Time 9.932
= Slack Time 20.078
    Clock Rise Edge 0.000
    + Clock Network Latency (Prop) 0.300
    = Beginpoint Arrival Time 0.300
+-----+
| Pin | Edge | Net | Cell | Delay | Arrival Time | Required Time |
+-----+
| mOpd/bb/mel/qout_reg/CLK | ^ clk_L4_N33 | DFFPOSX1 | 0.249 | 0.549 | 20.379 |
| mOpd/bb/mel/qout_reg/Q | v B[1] | AND2X1 | 0.101 | 0.649 | 20.627 |
| U1958/B | v B[1] | AND2X1 | 0.119 | 0.769 | 20.728 |
| U1958/Y | v a/13/p1[1] | AND2X1 | 0.000 | 0.769 | 20.847 |
| U1280/B | v a/13/p1[1] | AND2X1 | 0.049 | 0.818 | 20.896 |
| U1280/Y | v a/13/f02/n3 | AND2X1 | 0.000 | 0.818 | 20.896 |
| U1281/A | v a/13/f02/n3 | INVX1 | 0.000 | 0.825 | 20.904 |
| U1281/Y | ^ n706 | INVX1 | 0.000 | 0.825 | 20.904 |
| a/13/f02/U3/C | v n706 | OA121X1 | 0.000 | 0.955 | 21.033 |
| a/13/f02/U3/Y | v a/13/c0[1] | OA121X1 | 0.129 | 0.955 | 21.033 |
| U1402/A | v a/13/c0[1] | INVX1 | 0.000 | 0.955 | 21.033 |
| U1402/Y | ^ n2168 | INVX1 | 0.072 | 1.027 | 21.105 |
| a/13/f03/U2/A | v n2168 | XOR2X1 | 0.000 | 1.027 | 21.105 |
| a/13/f03/U2/Y | v a/13/ps0[2] | XOR2X1 | 0.044 | 1.070 | 21.149 |
| U932/A | v a/13/ps0[2] | AND2X1 | 0.000 | 1.070 | 21.149 |
| U932/Y | v n2257 | AND2X1 | 0.036 | 1.106 | 21.184 |
| U933/A | v n2257 | INVX1 | 0.000 | 1.106 | 21.184 |
| U933/Y | ^ n522 | INVX1 | 0.003 | 1.109 | 21.187 |
| a/13/f12/U3/C | ^ n522 | OA121X1 | 0.000 | 1.109 | 21.187 |
| a/13/f12/U3/Y | v a/13/c1[1] | OA121X1 | 0.127 | 1.236 | 21.315 |
| U1431/A | v a/13/c1[1] | INVX1 | 0.000 | 1.236 | 21.315 |
| U1431/Y | v n2258 | INVX1 | 0.074 | 1.310 | 21.389 |
| a/13/f13/U2/A | ^ n2258 | XOR2X1 | 0.000 | 1.311 | 21.389 |
| a/13/f13/U2/Y | v a/13/n51[2] | XOR2X1 | 0.045 | 1.356 | 21.435 |

```

Figure 61: CPU with Comparator Physical Synthesis Timing Report

Maximum Clock Frequency Estimation To calculate the maximum clock frequency, we must know the simulated frequency and the total slack time available at that frequency after physical synthesis. For the CPU with Comparator design, the CPU was simulated at 30MHz, and the total slack time was found to be 20.078ns after physical synthesis. The period time for a CPU running at 30MHz is

$$T_c = \frac{1}{f} = 33.33\text{ns} \quad (13)$$

Knowing that the period time can be diminished by approximately 20.078ns, the maximum frequency is found to be

$$T_f = T_c - 20.078\text{ns} = 13.255\text{ns} \quad (14)$$

$$f = \frac{1}{T_f} = 75.44\text{MHz} \quad (15)$$

2.10 Post Physical Synthesis Simulation

```

Terminal
File Edit View Terminal Tabs Help
*** Memory Usage #1 (Current mem = 359.105M, initial mem = 46.070M) ***
-- Ending "Encounter" (totcpu=0:03:24, real=0:05:04, mem=359.1M) --
Elapsed Time = 5:05.09, CPU Time = 193.575
alukens@saturn.ece.iit.edu:~% xrun gsc145nm.v tb_test_comp.v final.v +access+R
xrun: 18.03-#001: (c) Copyright 1995-2018 Cadence Design Systems, Inc.
file: final.v
    module worklib.cpu:v
        errors: 0 warnings: 0
        Caching library 'worklib' ..... Done
    Elaborating the design hierarchy:
    Top level design units:
        BUFX4
        CLKBUF1
        CLKBUF2
        CLKBUF3
        DFFNEGX1
        DFFSR
        FAX1
        HAX1
        INVX2
        MUX2X1
        OAI22X1
        OR2X2
        TBUFX1
        stimulus
    Building instance overlay tables: ..... Done
    Generating native compiled code:
        worklib.cpu:v <0x7b012534>
            streams: 0, words: 0
    Building instance specific data structures.
    Loading native compiled code: ..... Done
    Design hierarchy summary:
    Instances Unique
    Modules: 14793 34
    UDPs: 1660 4
    Primitives: 25770 6
    Timing outputs: 14793 19
    Registers: 1670 18
    Scalar wires: 16683 -
    Expanded wires: 47 5
    Always blocks: 1 1
    Initial blocks: 3 3
    Pseudo assignments: 9 9
    Timing checks: 9930 1659
    Simulation timescale: 10ps
    Writing initial simulation snapshot: worklib.BUFX4:v
    Loading snapshot worklib.BUFX4:v ..... Done
    xcclium> source /apps/cadence/XCELLIUM1803/tools/xcclium/files/xmsimrc
    xcclium> run
    Simulation complete via $finish() at time 1501 NS + 0
    ./tb_test_comp.v:31 #1 $finish;
    xcclium> exit
alukens@saturn.ece.iit.edu:~%

```

Figure 62: CPU with Comparator Physical Synthesis Simulation

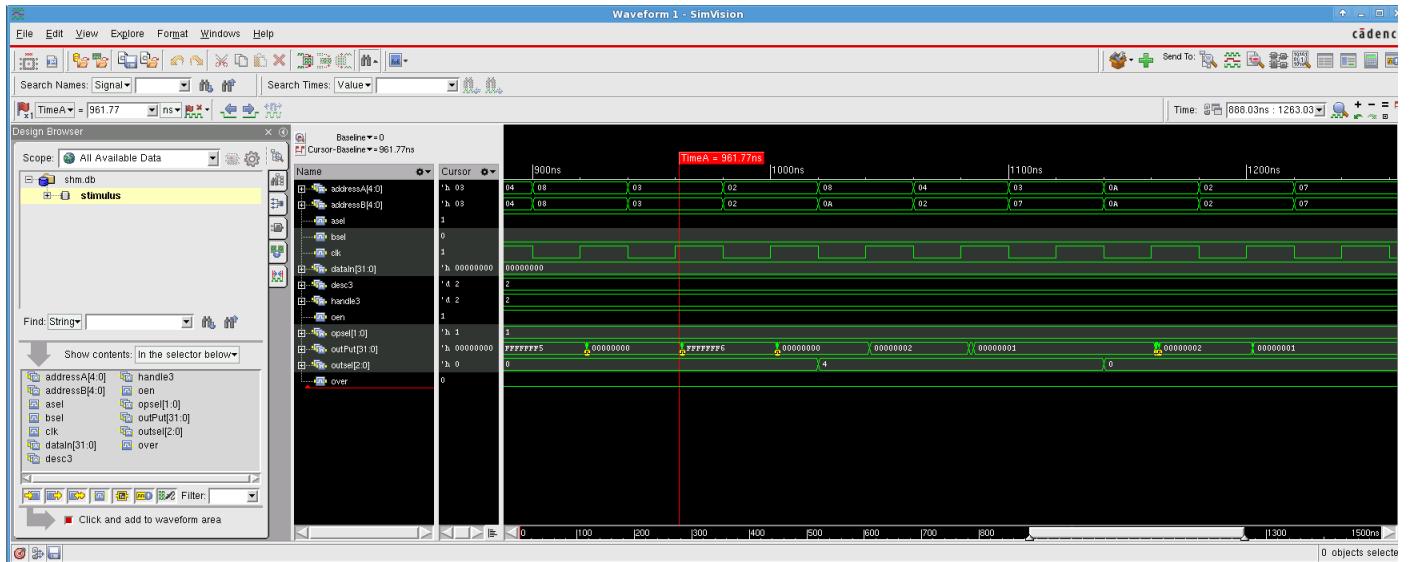


Figure 63: CPU with Comparator Physical Synthesis SimVision Data

2.11 Conclusions

This case study should be considered a success. Students were successfully able to implement a 32-bit comparator module design using Verilog. This required the implementation of a single-bit comparator and

a 4-to-2 multiplexer. These designs were successfully debugged and compiled such that they can be used in the CPU design. Additionally, a Verilog testbench was implemented to allow for the functionality of the comparator to be asserted.

The skills obtained in this case study will be utilized to develop increasingly complex digital logic designs in Verilog. Additionally, students will be able to apply the ASIC design flow to a variety of VLSI fabrication processes due to the process-independent nature of this case study. To further improve the skills utilized in this process, students should attempt to manually floorplan a design using the ASIC design flow. This will allow for the datapath to be further optimized, decreasing the critical path delay and allowing the CPU design to operate at a higher frequency.

2.12 Resources

- Choi, Ken. “ECE 429 Final Project Fall 2020 Manual.” Illinois Institute of Technology, November 25, 2020.
- Kim, Victoria. “ECE 429 Guideline for Writing Report & Grading Criteria.” Illinois Institute of Technology, November 25, 2020.