ECE 429 Laboratory 8: Carry-Ripple Addition III

Alexander Lukens

Illinois Institute of Technology

November 14, 2020

Acknowledgement: I acknowledge all of the work (including figures and codes) belongs to me and/or persons who are referenced.
Signature: *Alexander Lukens*

1 Introduction

In this laboratory, students will continue to elaborate on their ripple adder design by implementing a 4-bit adder based on their single-bit adder design. This will require students to create multiple instances of the Full-adder cellview, design a schematic that correctly connects the ripple adder, and create a corresponding layout cellview that corresponds to the schematic created earlier. This layout cellview will utilize the full adder layout implemented in Laboratory 7

2 Theory\Pre-lab

CMOS designs use numerous design “masks” to define the various separate regions that make up the physical design. These designs can be very complex, so Electronic Design Automation (EDA) software is utilized to create a physical layout for the circuit. One such EDA software suite is Cadence Virtuoso. In Cadence Virtuoso, each of the individual layers can be modeled and tested against the physical design rules for a certain process, reducing the time required to design an integrated circuit dramatically.

Inside Cadence Virtuoso, there are a variety of tools available to further decrease the probability of design errors and overall design time. The first tool is design rule checking (DRC). This allows for the physical layout of a circuit to be checked for compliance with the design rules of a certain process (on which the circuit will eventually be produced). This ensures that when the circuit is created, it will not fail due to physical spacing errors. Another tool is Layout vs Schematic (LVS) testing. LVS testing allows the physical layout produced in Cadence Virtuoso to be compared with a schematic cellview to ensure that they will function identically. This means that the overall design can first be created as a schematic (to first ensure the digital logic design functions correctly) before spending time designing the physical layout.

3 Implementation

The first step in implementing a 4-bit adder in Cadence Virtuoso, using the carry-ripple full adder design, is to create a schematic cellview using 4 instances of the full adder. This is done by making a new cell for the 4-bit adder, and then creating a schematic cellview.

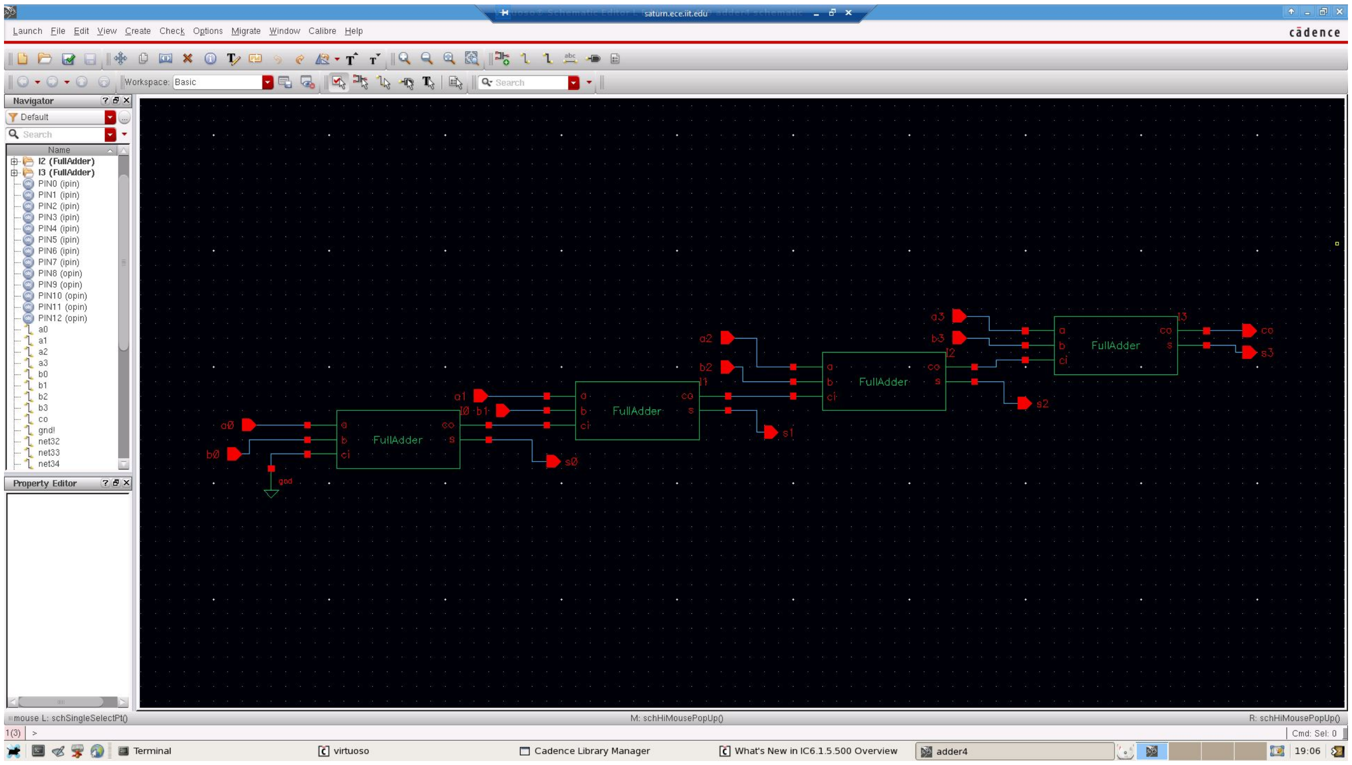


Figure 1: 4-bit Adder Schematic Cellview

After defining the various ports in the schematic cellview, a symbol cellview can be easily created by using the "Create Cellview from Cellview" command.

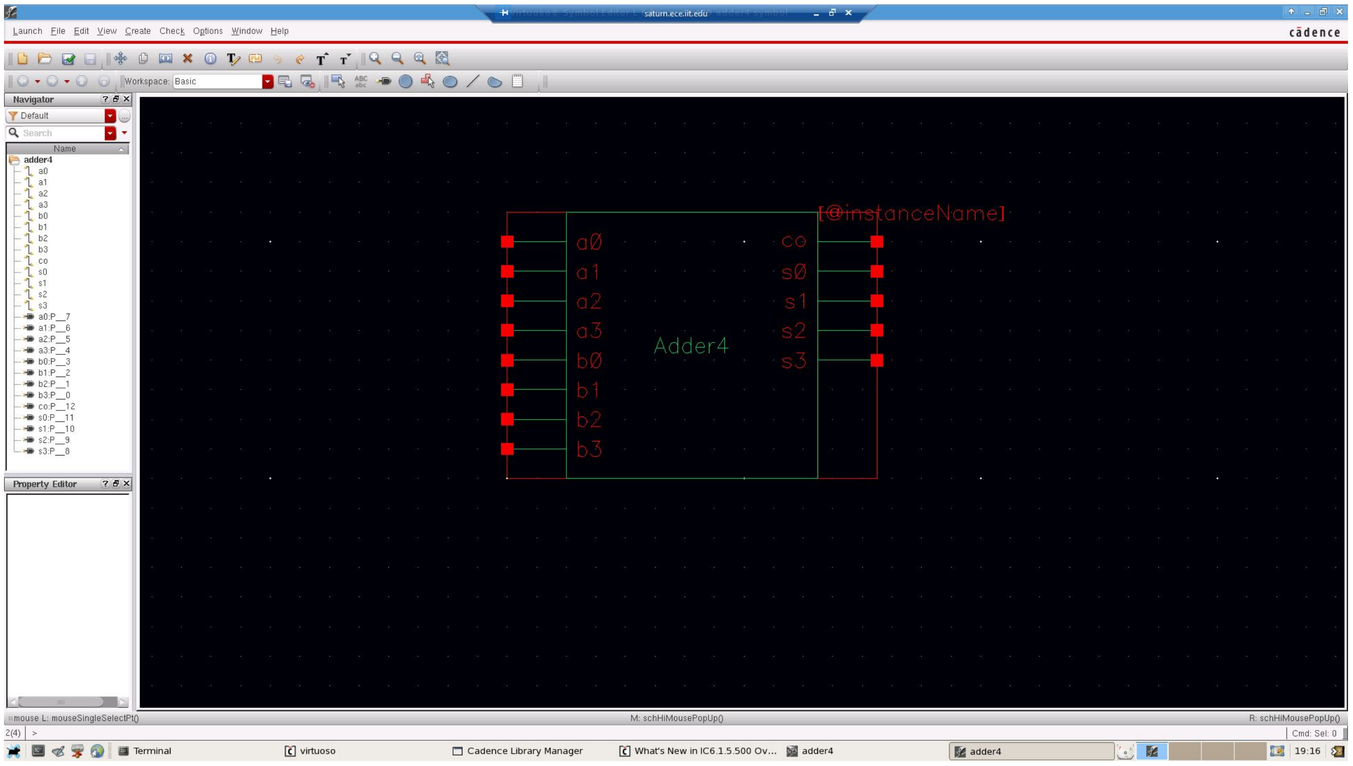


Figure 2: 4-bit Adder Symbol Cellview

After creating a schematic and symbol cellview, the next step is to implement the 4-bit adder in a layout cellview. This requires utilizing 4 instances of the full-adder layout created in the previous laboratory session, and connecting the inputs and outputs as specified in the schematic cellview.

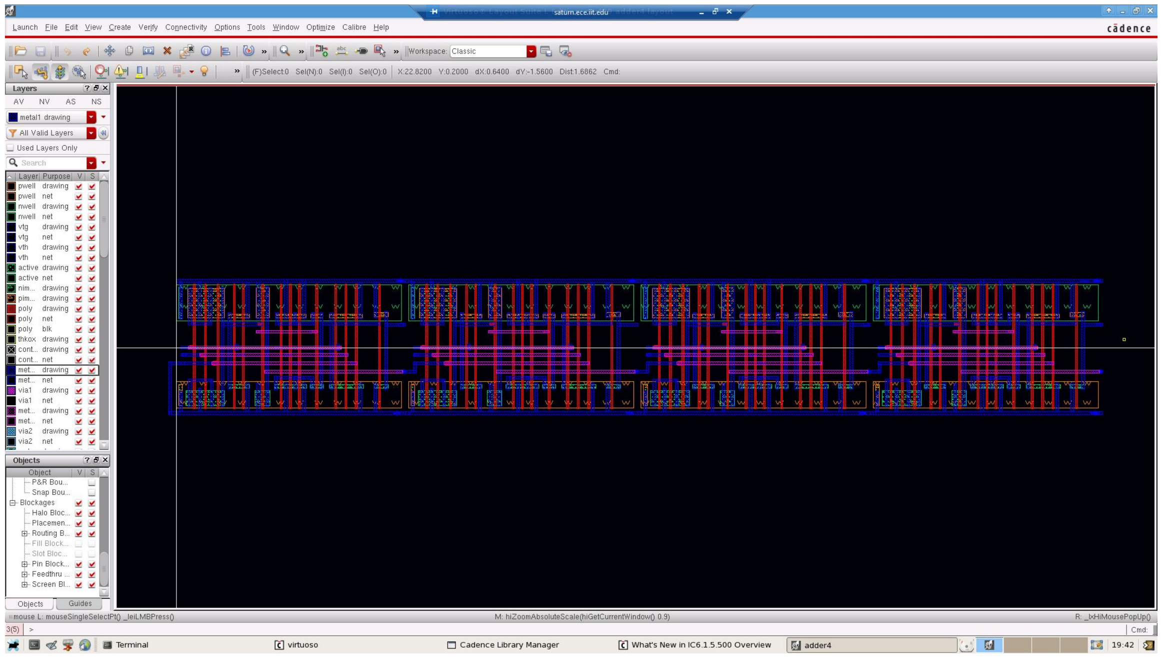
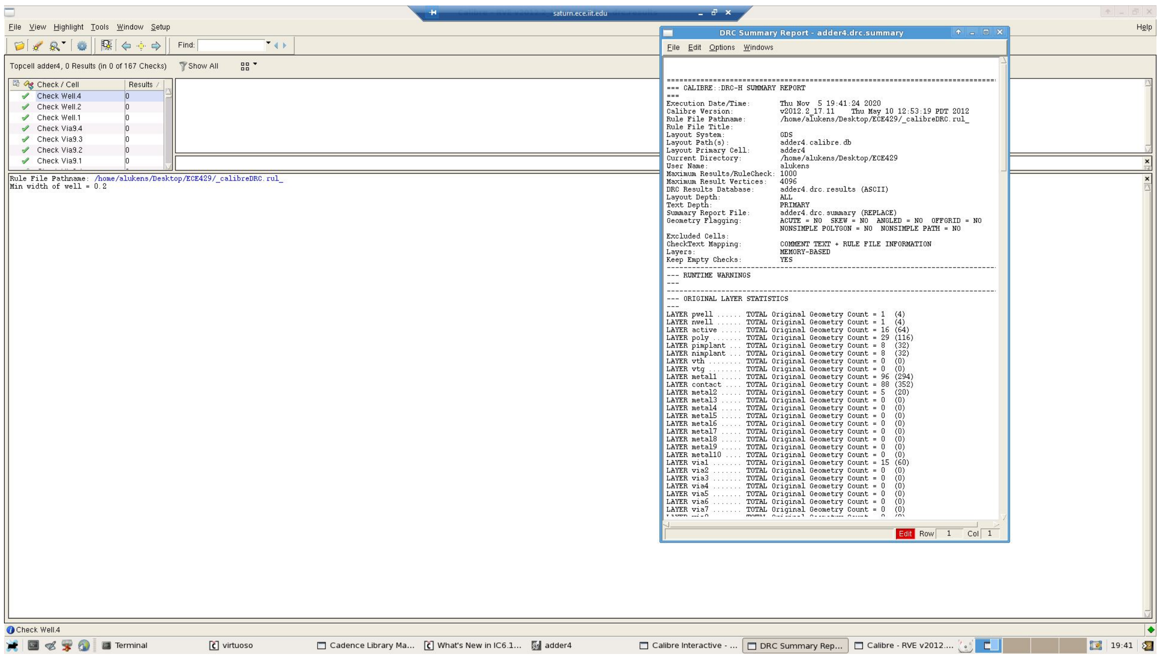


Figure 3: 4-bit Adder Layout Cellview

Once all ports and connections are created according to the schematic cellview, design rule checking (DRC) and Layout vs Schematic (LVS) testing can be performed on the layout.



Topcell adder4, 0 Results (in 0 of 167 Checks)

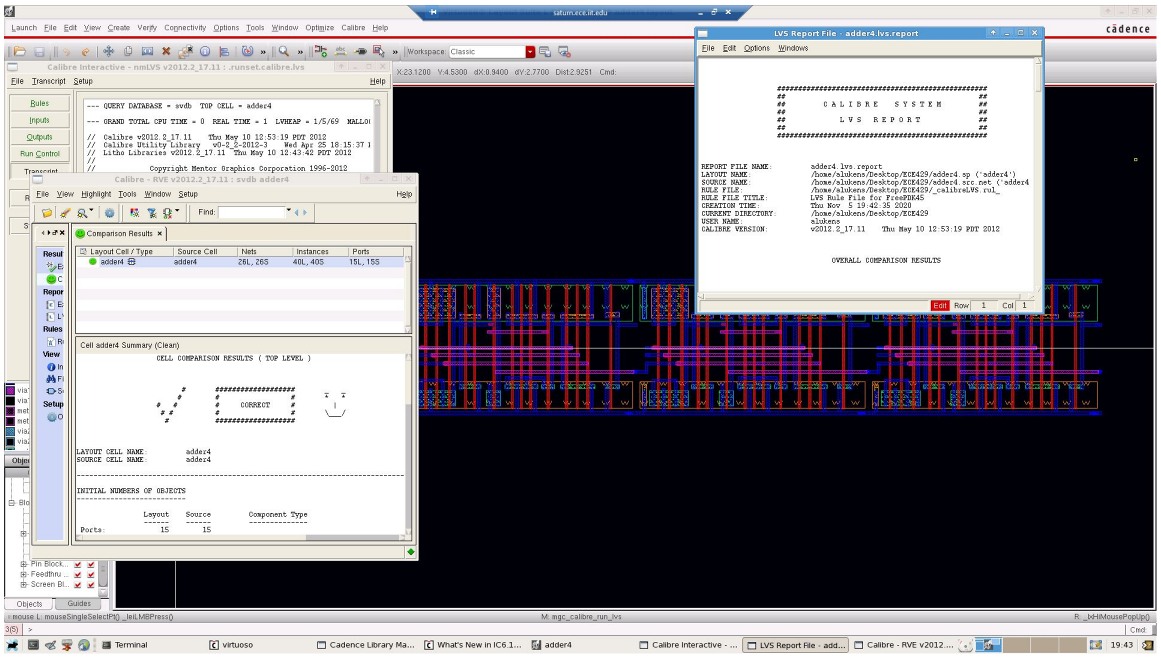
Check / Cell	Results
✓ Check Well.4	0
✓ Check Well.2	0
✓ Check Well.1	0
✓ Check Via9.4	0
✓ Check Via9.3	0
✓ Check Via9.2	0
✓ Check Via9.1	0

```
Rule File Pathname: /home/alukens/Desktop/ECE429/_calibreDRC.rul_
Min width of well = 0.2
```

DRC Summary Report - adder4.drc.summary

```
================================================================================
=== CALIBRE::DRC-H SUMMARY REPORT
===
Execution Date/Time:       Thu Nov  5 19:41:24 2020
Calibre Version:           v2012.2_17.11    Thu May 10 12:53:19 PDT 2012
Rule File Pathname:        /home/alukens/Desktop/ECE429/_calibreDRC.rul_
Rule File Title:
Layout System:             GDS
Layout Path(s):            adder4.calibre.db
Layout Primary Cell:       adder4
Current Directory:         /home/alukens/Desktop/ECE429
User Name:                 alukens
Maximum Results/RuleCheck: 1000
Maximum Result Vertices:   4096
DRC Results Database:      adder4.drc.results (ASCII)
Layout Depth:              ALL
Text Depth:                PRIMARY
Summary Report File:       adder4.drc.summary (REPLACE)
Geometry Flagging:         ACUTE = NO  SKEW = NO  ANGLED = NO  OFFGRID = NO
                           NONSIMPLE POLYGON = NO  NONSIMPLE PATH = NO
Excluded Cells:
CheckText Mapping:         COMMENT TEXT + RULE FILE INFORMATION
Layers:                    MEMORY-BASED
Keep Empty Checks:         YES
--------------------------------------------------------------------------------
--- RUNTIME WARNINGS
---
--------------------------------------------------------------------------------
--- ORIGINAL LAYER STATISTICS
---
LAYER pwell ...... TOTAL Original Geometry Count = 1  (4)
LAYER nwell ...... TOTAL Original Geometry Count = 1  (4)
LAYER active ..... TOTAL Original Geometry Count = 16 (64)
LAYER poly ....... TOTAL Original Geometry Count = 29 (116)
LAYER pimplant ... TOTAL Original Geometry Count = 8  (32)
LAYER nimplant ... TOTAL Original Geometry Count = 8  (32)
LAYER vth ........ TOTAL Original Geometry Count = 0  (0)
LAYER vtg ........ TOTAL Original Geometry Count = 0  (0)
LAYER metal1 ..... TOTAL Original Geometry Count = 96 (294)
LAYER contact .... TOTAL Original Geometry Count = 88 (352)
LAYER metal2 ..... TOTAL Original Geometry Count = 5  (20)
LAYER metal3 ..... TOTAL Original Geometry Count = 0  (0)
LAYER metal4 ..... TOTAL Original Geometry Count = 0  (0)
LAYER metal5 ..... TOTAL Original Geometry Count = 0  (0)
LAYER metal6 ..... TOTAL Original Geometry Count = 0  (0)
LAYER metal7 ..... TOTAL Original Geometry Count = 0  (0)
LAYER metal8 ..... TOTAL Original Geometry Count = 0  (0)
LAYER metal9 ..... TOTAL Original Geometry Count = 0  (0)
LAYER metal10 .... TOTAL Original Geometry Count = 0  (0)
LAYER via1 ....... TOTAL Original Geometry Count = 15 (60)
LAYER via2 ....... TOTAL Original Geometry Count = 0  (0)
LAYER via3 ....... TOTAL Original Geometry Count = 0  (0)
LAYER via4 ....... TOTAL Original Geometry Count = 0  (0)
LAYER via5 ....... TOTAL Original Geometry Count = 0  (0)
LAYER via6 ....... TOTAL Original Geometry Count = 0  (0)
LAYER via7 ....... TOTAL Original Geometry Count = 0  (0)
```

Figure 4: Layout DRC



Calibre Interactive - nmLVS v2012.2_17.11 : .runset.calibre.lvs

```
--- QUERY DATABASE = svdb  TOP CELL = adder4

--- GRAND TOTAL CPU TIME = 0  REAL TIME = 1  LVHEAP = 1/5/69  MALLO

//  Calibre v2012.2_17.11    Thu May 10 12:53:19 PDT 2012
//  Calibre Utility Library   v0-2_2-2012-3    Wed Apr 25 18:15:37
//  Litho Libraries v2012.2_17.11  Thu May 10 12:43:42 PDT 2012
//
//              Copyright Mentor Graphics Corporation 1996-2012
```

Calibre - RVE v2012.2_17.11 : svdb adder4

Comparison Results

Layout Cell / Type	Source Cell	Nets	Instances	Ports
adder4	adder4	26L, 26S	40L, 40S	15L, 15S

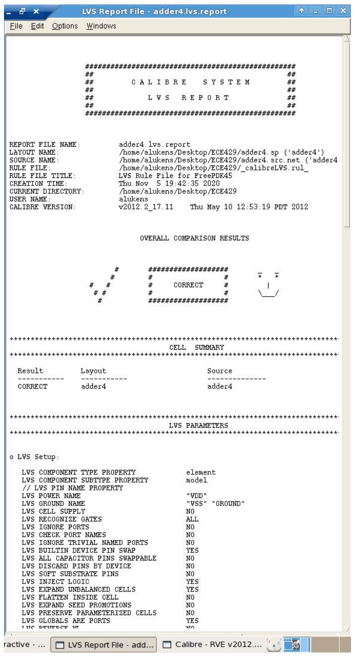
Cell adder4 Summary (Clean)

```
               CELL COMPARISON RESULTS ( TOP LEVEL )



                #       ##################
               #        #                #
            #  #        #     CORRECT    #       _  _
             # #        #                #        |
              #         ##################      \___/


LAYOUT CELL NAME:         adder4
SOURCE CELL NAME:         adder4

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

INITIAL NUMBERS OF OBJECTS
--------------------------

               Layout    Source         Component Type
               ------    ------         --------------
 Ports:          15        15
```

LVS Report File - adder4.lvs.report

```
          ##################################################
          ##                                              ##
          ##         C A L I B R E    S Y S T E M         ##
          ##                                              ##
          ##             L V S   R E P O R T              ##
          ##                                              ##
          ##################################################


REPORT FILE NAME:         adder4.lvs.report
LAYOUT NAME:              /home/alukens/Desktop/ECE429/adder4.sp ('adder4')
SOURCE NAME:              /home/alukens/Desktop/ECE429/adder4.src.net ('adder4
RULE FILE:                /home/alukens/Desktop/ECE429/_calibreLVS.rul_
RULE FILE TITLE:          LVS Rule File for FreePDK45
CREATION TIME:            Thu Nov  5 19:42:35 2020
CURRENT DIRECTORY:        /home/alukens/Desktop/ECE429
USER NAME:                alukens
CALIBRE VERSION:          v2012.2_17.11    Thu May 10 12:53:19 PDT 2012



                     OVERALL COMPARISON RESULTS
```

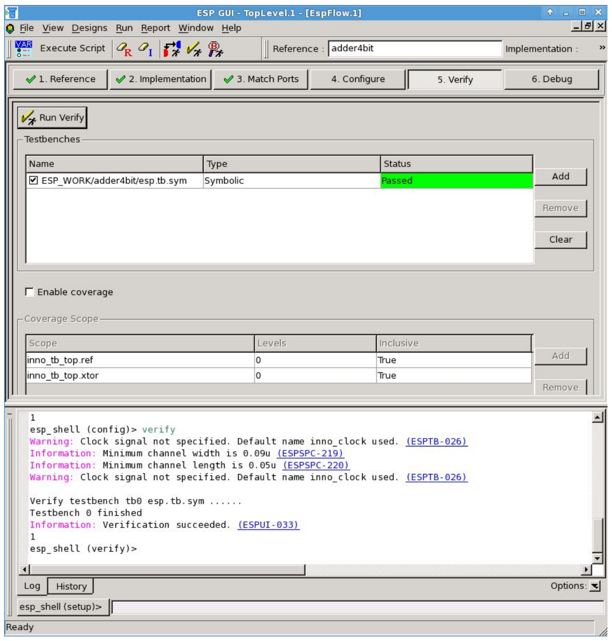
Figure 5: Layout LVS Result



```
               ##################################################
               ##                                              ##
               ##         C A L I B R E    S Y S T E M         ##
               ##                                              ##
               ##             L V S   R E P O R T              ##
               ##                                              ##
               ##################################################


REPORT FILE NAME:         adder4.lvs.report
LAYOUT NAME:              /home/alukens/Desktop/ECE429/adder4.sp ('adder4')
SOURCE NAME:              /home/alukens/Desktop/ECE429/adder4.src.net ('adder4
RULE FILE:                /home/alukens/Desktop/ECE429/_calibreLVS.rul_
RULE FILE TITLE:          LVS Rule File for FreePDK45
CREATION TIME:            Thu Nov  5 19:42:35 2020
CURRENT DIRECTORY:        /home/alukens/Desktop/ECE429
USER NAME:                alukens
CALIBRE VERSION:          v2012.2_17.11    Thu May 10 12:53:19 PDT 2012


                          OVERALL COMPARISON RESULTS


                    #        ###################
                   #         #                 #
               #  #          #     CORRECT     #
                # #          #                 #
                 #           ###################


**************************************************************************
                                CELL  SUMMARY
**************************************************************************

  Result        Layout                        Source
  -----------   -----------                   --------------
  CORRECT       adder4                        adder4



**************************************************************************
                                LVS PARAMETERS
**************************************************************************


o LVS Setup:

   LVS COMPONENT TYPE PROPERTY            element
   LVS COMPONENT SUBTYPE PROPERTY         model
   // LVS PIN NAME PROPERTY
   LVS POWER NAME                         "VDD"
   LVS GROUND NAME                        "VSS" "GROUND"
   LVS CELL SUPPLY                        NO
   LVS RECOGNIZE GATES                    ALL
   LVS IGNORE PORTS                       NO
   LVS CHECK PORT NAMES                   NO
   LVS IGNORE TRIVIAL NAMED PORTS         NO
   LVS BUILTIN DEVICE PIN SWAP            YES
   LVS ALL CAPACITOR PINS SWAPPABLE       NO
   LVS DISCARD PINS BY DEVICE             NO
   LVS SOFT SUBSTRATE PINS                NO
   LVS INJECT LOGIC                       YES
   LVS EXPAND UNBALANCED CELLS            YES
   LVS FLATTEN INSIDE CELL                NO
   LVS EXPAND SEED PROMOTIONS             NO
   LVS PRESERVE PARAMETERIZED CELLS       NO
   LVS GLOBALS ARE PORTS                  YES
```

Figure 6: Layout LVS Report

If all layout tests pass, the design should then be compared to a verilog file performing the 4-bit adder operation to ensure that the layout functions as intended. This is done using Cadence Formality ESP, which can compare a netlist to a verilog file.



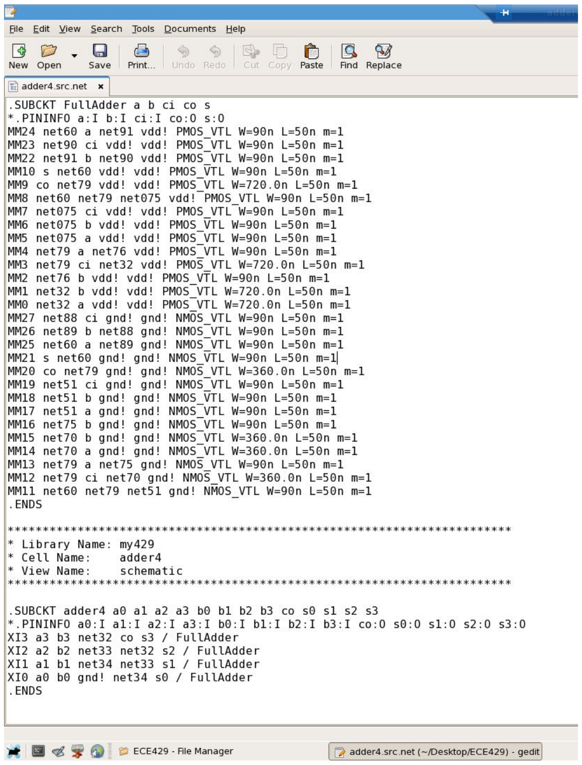
Name	Type	Status
☑ ESP_WORK/adder4bit/esp.tb.sym	Symbolic	Passed

Scope	Levels	Inclusive
inno_tb_top.ref	0	True
inno_tb_top.xtor	0	True

```
1
esp_shell (config)> verify
Warning: Clock signal not specified. Default name inno_clock used. (ESPTB-026)
Information: Minimum channel width is 0.09u (ESPSPC-219)
Information: Minimum channel length is 0.05u (ESPSPC-220)
Warning: Clock signal not specified. Default name inno_clock used. (ESPTB-026)

Verify testbench tb0 esp.tb.sym ......
Testbench 0 finished
Information: Verification succeeded. (ESPUI-033)
1
esp_shell (verify)>
```

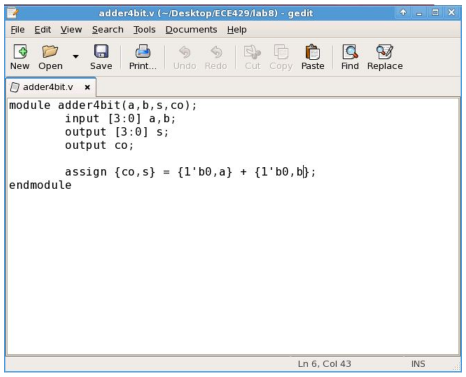
```
esp_shell (setup)>
```

Figure 7: ESP GUI Results



```
.SUBCKT FullAdder a b ci co s
*.PININFO a:I b:I ci:I co:O s:O
MM24 net60 a net91 vdd! PMOS_VTL W=90n L=50n m=1
MM23 net90 ci vdd! vdd! PMOS_VTL W=90n L=50n m=1
MM22 net91 b net90 vdd! PMOS_VTL W=90n L=50n m=1
MM10 s net60 vdd! vdd! PMOS_VTL W=90n L=50n m=1
MM9 co net79 vdd! vdd! PMOS_VTL W=720.0n L=50n m=1
MM8 net60 net79 net075 vdd! PMOS_VTL W=90n L=50n m=1
MM7 net075 ci vdd! vdd! PMOS_VTL W=90n L=50n m=1
MM6 net075 b vdd! vdd! PMOS_VTL W=90n L=50n m=1
MM5 net075 a vdd! vdd! PMOS_VTL W=90n L=50n m=1
MM4 net79 a net76 vdd! PMOS_VTL W=90n L=50n m=1
MM3 net79 ci net32 vdd! PMOS_VTL W=720.0n L=50n m=1
MM2 net76 b vdd! vdd! PMOS_VTL W=90n L=50n m=1
MM1 net32 b vdd! vdd! PMOS_VTL W=720.0n L=50n m=1
MM0 net32 a vdd! vdd! PMOS_VTL W=720.0n L=50n m=1
MM27 net88 ci gnd! gnd! NMOS_VTL W=90n L=50n m=1
MM26 net89 b net88 gnd! NMOS_VTL W=90n L=50n m=1
MM25 net60 a net89 gnd! NMOS_VTL W=90n L=50n m=1
MM21 s net60 gnd! gnd! NMOS_VTL W=90n L=50n m=1
MM20 co net79 gnd! gnd! NMOS_VTL W=360.0n L=50n m=1
MM19 net51 ci gnd! gnd! NMOS_VTL W=90n L=50n m=1
MM18 net51 b gnd! gnd! NMOS_VTL W=90n L=50n m=1
MM17 net51 a gnd! gnd! NMOS_VTL W=90n L=50n m=1
MM16 net75 b gnd! gnd! NMOS_VTL W=90n L=50n m=1
MM15 net70 b gnd! gnd! NMOS_VTL W=360.0n L=50n m=1
MM14 net70 a gnd! gnd! NMOS_VTL W=360.0n L=50n m=1
MM13 net79 a net75 gnd! NMOS_VTL W=90n L=50n m=1
MM12 net79 ci net70 gnd! NMOS_VTL W=360.0n L=50n m=1
MM11 net60 net79 net51 gnd! NMOS_VTL W=90n L=50n m=1
.ENDS

************************************************************************
* Library Name: my429
* Cell Name:    adder4
* View Name:    schematic
************************************************************************

.SUBCKT adder4 a0 a1 a2 a3 b0 b1 b2 b3 co s0 s1 s2 s3
*.PININFO a0:I a1:I a2:I a3:I b0:I b1:I b2:I b3:I co:O s0:O s1:O s2:O s3:O
XI3 a3 b3 net32 co s3 / FullAdder
XI2 a2 b2 net33 net32 s2 / FullAdder
XI1 a1 b1 net34 net33 s1 / FullAdder
XI0 a0 b0 gnd! net34 s0 / FullAdder
.ENDS
```

Figure 8: 4-bit Adder Generated Netlist



```
module adder4bit(a,b,s,co);
        input [3:0] a,b;
        output [3:0] s;
        output co;

        assign {co,s} = {1'b0,a} + {1'b0,b};
endmodule
```

Figure 9: 4-bit Adder Verilog File

After ensurign that the adder functions as expected, a testing circuit was created in Cadence Virtuoso. This circuit will assist in performing timing analysis on the 4-bit adder using various input stimuli.

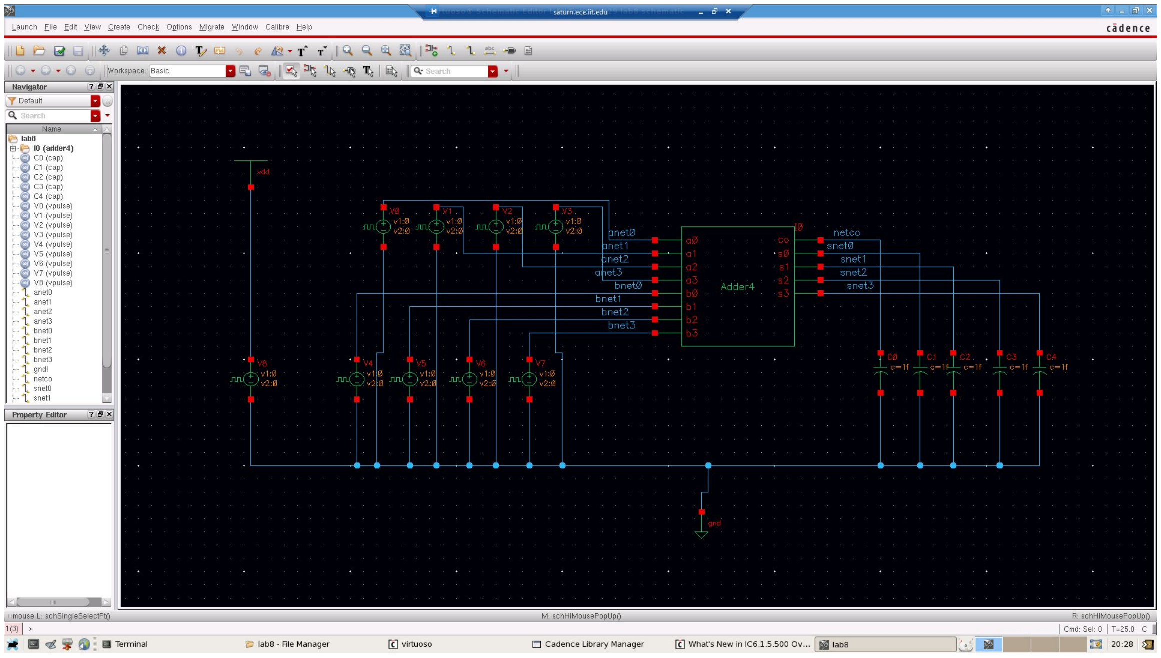
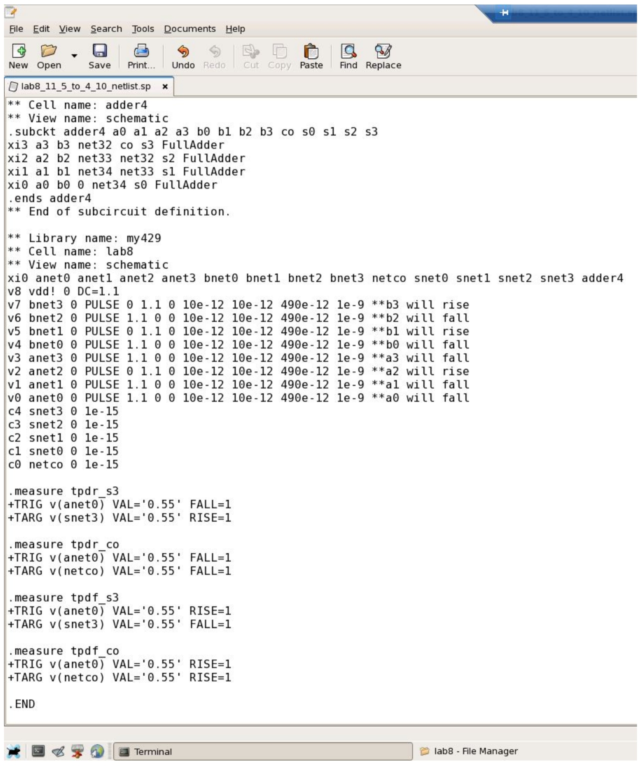


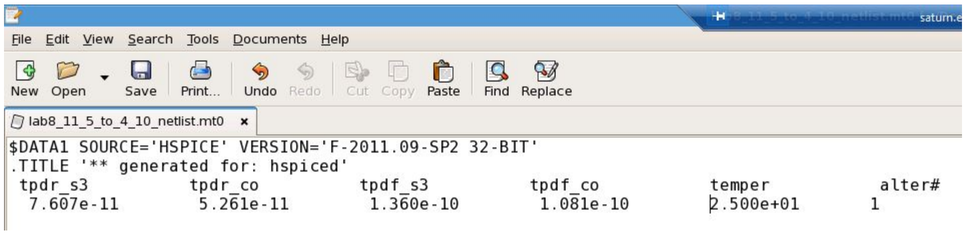
Figure 10: 4-bit Adder Testing Circuit

A SPICE netlist was created to perform the delay analysis. The SPICE file and the resultant data is shown below



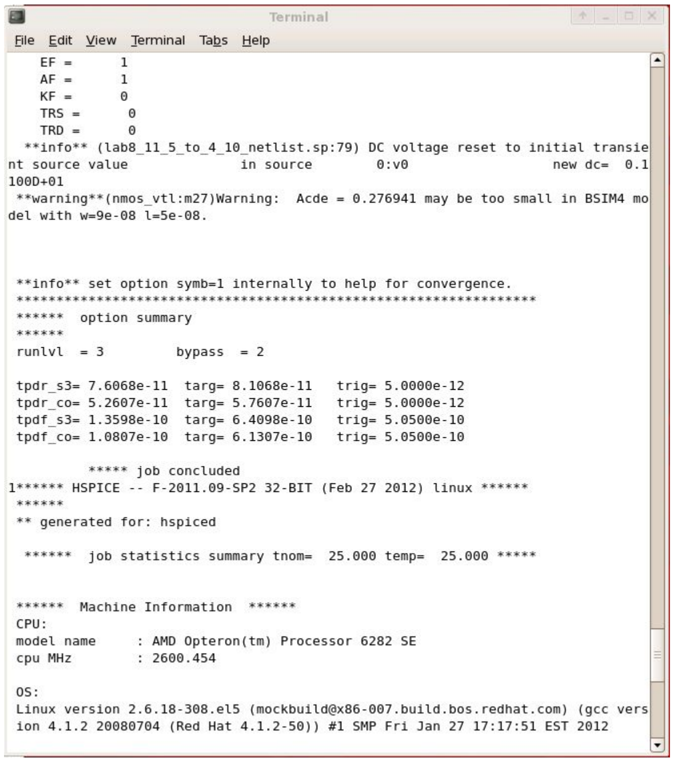
```
** Cell name: adder4
** View name: schematic
.subckt adder4 a0 a1 a2 a3 b0 b1 b2 b3 co s0 s1 s2 s3
xi3 a3 b3 net32 co s3 FullAdder
xi2 a2 b2 net33 net32 s2 FullAdder
xi1 a1 b1 net34 net33 s1 FullAdder
xi0 a0 b0 0 net34 s0 FullAdder
.ends adder4
** End of subcircuit definition.

** Library name: my429
** Cell name: lab8
** View name: schematic
xi0 anet0 anet1 anet2 anet3 bnet0 bnet1 bnet2 bnet3 netco snet0 snet1 snet2 snet3 adder4
v8 vdd! 0 DC=1.1
v7 bnet3 0 PULSE 0 1.1 0 10e-12 10e-12 490e-12 1e-9 **b3 will rise
v6 bnet2 0 PULSE 1.1 0 0 10e-12 10e-12 490e-12 1e-9 **b2 will fall
v5 bnet1 0 PULSE 0 1.1 0 10e-12 10e-12 490e-12 1e-9 **b1 will rise
v4 bnet0 0 PULSE 1.1 0 0 10e-12 10e-12 490e-12 1e-9 **b0 will fall
v3 anet3 0 PULSE 1.1 0 0 10e-12 10e-12 490e-12 1e-9 **a3 will fall
v2 anet2 0 PULSE 0 1.1 0 10e-12 10e-12 490e-12 1e-9 **a2 will rise
v1 anet1 0 PULSE 1.1 0 0 10e-12 10e-12 490e-12 1e-9 **a1 will fall
v0 anet0 0 PULSE 1.1 0 0 10e-12 10e-12 490e-12 1e-9 **a0 will fall
c4 snet3 0 1e-15
c3 snet2 0 1e-15
c2 snet1 0 1e-15
c1 snet0 0 1e-15
c0 netco 0 1e-15

.measure tpdr_s3
+TRIG v(anet0) VAL='0.55' FALL=1
+TARG v(snet3) VAL='0.55' RISE=1

.measure tpdr_co
+TRIG v(anet0) VAL='0.55' FALL=1
+TARG v(netco) VAL='0.55' FALL=1

.measure tpdf_s3
+TRIG v(anet0) VAL='0.55' RISE=1
+TARG v(snet3) VAL='0.55' FALL=1

.measure tpdf_co
+TRIG v(anet0) VAL='0.55' RISE=1
+TARG v(netco) VAL='0.55' RISE=1

.END
```

Figure 11: 4-bit Adder SPICE Delay Netlist


File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
lab8_11_5_to_4_10_netlist.mt0

```
$DATA1 SOURCE='HSPICE' VERSION='F-2011.09-SP2 32-BIT'
.TITLE '** generated for: hspiced'
 tpdr_s3          tpdr_co          tpdf_s3          tpdf_co          temper          alter#
  7.607e-11        5.261e-11        1.360e-10        1.081e-10        2.500e+01        1
```

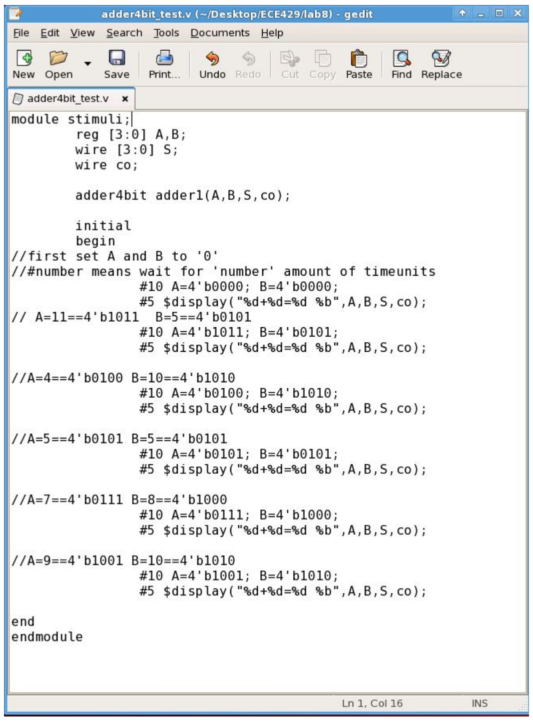
Figure 12: 4-bit Adder Testing Data



```
    EF =      1
    AF =      1
    KF =      0
    TRS =      0
    TRD =      0
  **info** (lab8_11_5_to_4_10_netlist.sp:79) DC voltage reset to initial transie
nt source value              in source        0:v0                  new dc=  0.1
100D+01
 **warning**(nmos_vtl:m27)Warning:  Acde = 0.276941 may be too small in BSIM4 mo
del with w=9e-08 l=5e-08.



 **info** set option symb=1 internally to help for convergence.
 ******************************************************************
 ******  option summary
 ******
 runlvl  = 3         bypass  = 2

 tpdr_s3= 7.6068e-11  targ= 8.1068e-11   trig= 5.0000e-12
 tpdr_co= 5.2607e-11  targ= 5.7607e-11   trig= 5.0000e-12
 tpdf_s3= 1.3598e-10  targ= 6.4098e-10   trig= 5.0500e-10
 tpdf_co= 1.0807e-10  targ= 6.1307e-10   trig= 5.0500e-10

          ***** job concluded
1****** HSPICE -- F-2011.09-SP2 32-BIT (Feb 27 2012) linux ******
 ******
 ** generated for: hspiced

  ******  job statistics summary tnom=  25.000 temp=  25.000 *****


 ******  Machine Information  ******
 CPU:
 model name     : AMD Opteron(tm) Processor 6282 SE
 cpu MHz        : 2600.454

 OS:
 Linux version 2.6.18-308.el5 (mockbuild@x86-007.build.bos.redhat.com) (gcc vers
 ion 4.1.2 20080704 (Red Hat 4.1.2-50)) #1 SMP Fri Jan 27 17:17:51 EST 2012
```

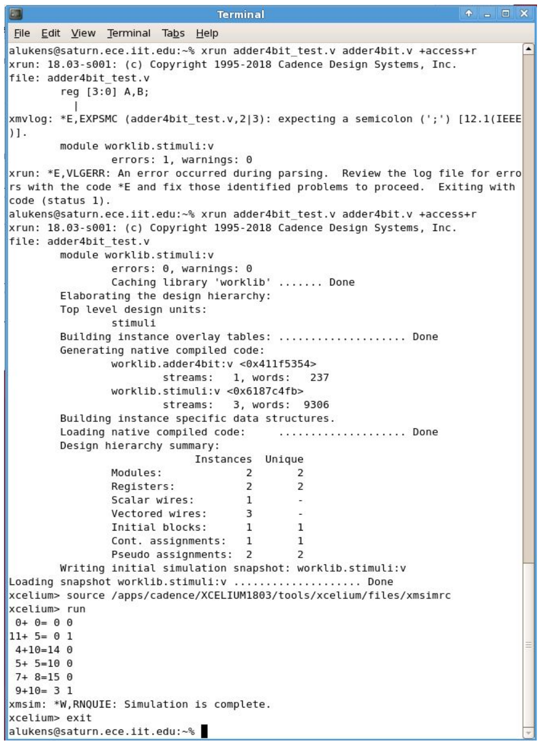
Figure 13: SPICE Terminal Output Screen

Finally, XRUN was utilized to validate the correctness of the verilog file created earlier. This was done to ensure that the adder functions correctly over a larger variety of test cases.



```
module stimuli;
        reg [3:0] A,B;
        wire [3:0] S;
        wire co;

        adder4bit adder1(A,B,S,co);

        initial
        begin
//first set A and B to '0'
//#number means wait for 'number' amount of timeunits
                #10 A=4'b0000; B=4'b0000;
                #5 $display("%d+%d=%d %b",A,B,S,co);
// A=11==4'b1011  B=5==4'b0101
                #10 A=4'b1011; B=4'b0101;
                #5 $display("%d+%d=%d %b",A,B,S,co);

//A=4==4'b0100 B=10==4'b1010
                #10 A=4'b0100; B=4'b1010;
                #5 $display("%d+%d=%d %b",A,B,S,co);

//A=5==4'b0101 B=5==4'b0101
                #10 A=4'b0101; B=4'b0101;
                #5 $display("%d+%d=%d %b",A,B,S,co);

//A=7==4'b0111 B=8==4'b1000
                #10 A=4'b0111; B=4'b1000;
                #5 $display("%d+%d=%d %b",A,B,S,co);

//A=9==4'b1001 B=10==4'b1010
                #10 A=4'b1001; B=4'b1010;
                #5 $display("%d+%d=%d %b",A,B,S,co);

end
endmodule
```

Figure 14: Adder Testbench Verilog File



```
alukens@saturn.ece.iit.edu:~% xrun adder4bit_test.v adder4bit.v +access+r
xrun: 18.03-s001: (c) Copyright 1995-2018 Cadence Design Systems, Inc.
file: adder4bit_test.v
        reg [3:0] A,B;
          |
xmvlog: *E,EXPSMC (adder4bit_test.v,2|3): expecting a semicolon (';') [12.1(IEEE
)].
        module worklib.stimuli:v
                errors: 1, warnings: 0
xrun: *E,VLGERR: An error occurred during parsing.  Review the log file for erro
rs with the code *E and fix those identified problems to proceed.  Exiting with
code (status 1).
alukens@saturn.ece.iit.edu:~% xrun adder4bit_test.v adder4bit.v +access+r
xrun: 18.03-s001: (c) Copyright 1995-2018 Cadence Design Systems, Inc.
file: adder4bit_test.v
        module worklib.stimuli:v
                errors: 0, warnings: 0
                Caching library 'worklib' ....... Done
        Elaborating the design hierarchy:
        Top level design units:
                stimuli
        Building instance overlay tables: .................... Done
        Generating native compiled code:
                worklib.adder4bit:v <0x411f5354>
                        streams:   1, words:   237
                worklib.stimuli:v <0x6187c4fb>
                        streams:   3, words:  9306
        Building instance specific data structures.
        Loading native compiled code:     .................... Done
        Design hierarchy summary:
                             Instances  Unique
                Modules:             2       2
                Registers:           2       2
                Scalar wires:        1       -
                Vectored wires:      3       -
                Initial blocks:      1       1
                Cont. assignments:   1       1
                Pseudo assignments:  2       2
        Writing initial simulation snapshot: worklib.stimuli:v
Loading snapshot worklib.stimuli:v .................... Done
xcelium> source /apps/cadence/XCELIUM1803/tools/xcelium/files/xmsimrc
xcelium> run
 0+ 0= 0 0
11+ 5= 0 1
 4+10=14 0
 5+ 5=10 0
 7+ 8=15 0
 9+10= 3 1
xmsim: *W,RNQUIE: Simulation is complete.
xcelium> exit
alukens@saturn.ece.iit.edu:~%
```

Figure 15: XRUN Simulation Results

4 Deliverables

- **How is the functionality of your adder design validated/verified at various abstraction levels (i.e. Verilog, transistor schematic, layout)?**

  Functionality of the 4-bit adder design was tested at several different points during this laboratory. At the lowest level, the layout cellview of the adder was checked for compliance with the design rules of the process being utilized (FreePDK45). This ensures that the design could actually be created using the layout in a physical process. Above this, Layout vs Schematic testing can be implemented to validate the correctness of the layout implementation. LVS testing compares the transistors and ports defined in the layout to those defined in the transistor schematic.

  After this correctness is confirmed, Formality ESP can be utilized to compare the functionality of a netlist (generated from the layout or transistor schematic) to a verilog file performing identical functionality. This allows designs created in Cadence Virtuoso to be compared to their verilog equivilent functionality to ensure that the design will perform the operation intended. Additionally, XRUN can be used to simulate a Verilog design over a variety of test cases. In this lab it was used to test numerous stimuli to the 4-bit adder Verilog file to ensure that the add functionality works as intended.

5 Conclusions

This laboratory should be considered a success. Students were successfully able to implement a 4-bit adder in Cadence Virtuoso. This required creating a schematic, symbol, and layout cellview. Students ensured the correctness of their designs by utilizing Design Rule Checking and Layout vs Schematic testing inside Cadence Virtuoso. Additionally, students used Formality ESP and XRUN simulations to compare their 4-bit adder to designs implemented using Verilog.

In future laboratories, students will utilize the skills gained in this laboratory session to design increasingly complex circuits. Students knowledge in implementing Verilog will be used to simulate complex circuits, and automatically generate layouts in Cadence Virtuoso.

6 Resources

- Choi, Ken. “ECE 429 Laboratory 8 Manual.” Illinois Institute of Technology, November 14, 2020.
- Kim, Victoria. “ECE 429 Guideline for Writing Report & Grading Criteria.” Illinois Institute of Technology, November 14, 2020.