VLSI Design, Verification and Fabrication of an Arithmetic Logic Unit (ALU) Using the State-of-the-Art Cadence Virtuoso

Tewodros Mamo, Nian Zhang, and Wagdy Mahmoud
Department of Electrical and Computer Engineering
University of the District of Columbia
4200 Connecticut Avenue, NW
Washington, D.C. 20008 USA
tewodros.mamo@udc.edu, nzhang@udc.edu, wmahmoud@udc.edu

Abstract

Although the Cadence Virtuoso has been widely used in many universities, there is lack of a shareable publication or tutorial on the very large-scale integration (VLSI) design, verification and fabrication. Few online tutorials or tutoring video are quite outdated. They either cannot be used as a step-by-step tutorial, or they only cover part of the entire design flow and are missing the fabrication part, which doesn't meet a designer's need. The lack of up to date publication or document has caused a huge barrier for universities to teach a VLSI lab. In addition, some tutorial only provides schematic design and pre-layout simulation, but doesn't include any layout as well as post-layout simulation. They also do not describe any pad-frame with pin mappings. Moreover, there is very few sources of educational tutorials of adding pad-frame, which is a complex but extremely important component before fabrication. Therefore, there is an urgent need to create and disseminate a shareable publication or document on VLSI design, verification and fabrication with the latest version of Cadence Virtuoso.

This paper conducts a comprehensive study on the newest version of Cadence Virtuoso, a state-of-the-art CAD tool for VLSI design. An 8-bit arithmetic logic unit (ALU) is used as a proof-of-concept example to perform the major VLSI design flow, including schematic capture, pre-layout simulation, physical layout, extract, design rule check (DRC), and layout vs. schematic (LVS). In addition, since this specific ALU is designed to perform three logical operations (i.e. AND, OR, and XOR) and two arithmetic operations (i.e. addition and subtraction), their respective schematic, symbol, testbench, pre-layout simulation, physical layout, post-layout simulation results are demonstrated with detailed snapshots. Moreover, the design layout is wired to a 40-pin MOSIS Tiny Chip pad-frame. The ALU circuit is then simulated along with the pad-frame and the simulation results are analyzed. Universities and designers around the world will find it an invaluable document, which ensures an efficient and fast VLSI design, verification and fabrication using the latest Cadence Virtuoso software.

1. Introduction

1.1 Motivation

Although the Cadence Virtuoso has been widely used in many universities, there is lack of a shareable publication or tutorial on the very large-scale integration (VLSI) design, verification and fabrication. Few online tutorials or tutoring video are quite outdated [1]-[9]. They either cannot be used as a step-by-step tutorial, or they only cover part of the entire design flow and are missing the fabrication part, which doesn't meet a designer's need. The lack of up to date publication or document has caused a huge barrier for universities to teach a VLSI lab. In addition, some tutorial only provides schematic design and pre-layout simulation, but doesn't include any layout as well as post-layout simulation [10]. They also do not describe any pad-frame with pin mappings. Moreover, there is very few sources of educational tutorials of adding pad-frame, which is a complex but extremely important component before fabrication. Therefore, there is an urgent need to create and disseminate a shareable publication or document on VLSI design, verification and fabrication with the latest version of Cadence Virtuoso.

This paper conducts a comprehensive study on the newest version of Cadence Virtuoso, a state-of-the-art CAD tool for VLSI design. An 8-bit arithmetic logic unit (ALU) is used as a proof-of-concept example to perform the major VLSI design flow, including schematic capture, pre-layout simulation, physical layout, extract, design rule check (DRC), and layout vs. schematic (LVS). An ALU is at the heart of a central processing unit that carries out arithmetic and logic operations on the operands in computer instruction words [11]. Building such an ALU would not only pave a path to better understanding of the VLSI flow, but also deepens students' understanding of VLSI workflow in Cadence. In addition, since this specific ALU is designed to perform three logical operations (i.e. AND, OR, and XOR) and two arithmetic operations (i.e. addition and subtraction), their respective schematic, symbol, testbench, pre-layout simulation, physical layout, post-layout simulation results are demonstrated with detailed snapshots. Moreover, the design layout is wired to a 40-pin MOSIS Tiny Chip pad-frame. The ALU circuit is then simulated along with the pad-frame and the simulation results are analyzed. Universities and designers around the world will find it an invaluable document, which ensures an efficient and fast VLSI design, verification and fabrication using the latest Cadence Virtuoso software.

1.2 Background

The design process starts with building each individual component logic gates that forms the ALU. The individual gates are made up of NMOS and PMOS devices using MOSIS 0.6u CMOS technology in Cadence; all with minimum width except for the inverter. The inverter has a width to length ratio of $\frac{W}{L} = \frac{2.6u}{1.5u} = 1.733$. This is done to make it have a symmetric input output characteristic around the threshold voltage Vth = $2.5\ volts$. For each gate and functional units, the schematic is first designed using transistors, then the layout is drawn, then each layout must pass DRC rule check after which LVS is performed to check whether the Layout and the schematics are a match and only then the design is completed. To aid in ease of assembly of the ALU and other functional units, such as FULL ADDER, and the gates are designed as a standard unit with a fixed height of 31.8 microns and a fixed location for NMOS and PMOS within the given height with an intent to build a custom library called project.

After individual gates and functional units have been designed as described above, a more complex logic function such as ALU can now be implemented using the predesigned project library components. One important reason for designing ALU this way is modularity. Since this ALU is a self-contained modular unit, it can be incorporated into any larger designs with little modification to its interconnects to repurpose it to fit given design needs or requirements. In addition, upon a successful design of such an ALU the designer will have built most of the basic logical units to realize more complex and robust designs manually or automatically using scrip or HDL.

2. Design of arithmetic logic unit (ALU)

In order to build and implement the ALU schematic and layout, we have first developed and tested numerous fundamental gates, including the transmission gates, 2 to 1 multiplexer, 4 to 1 multiplexer, AND gate, OR gate, XOR gate, full adder, and inverter. Due to consideration of page limit, this section will only describe the entire design of ALU as an example.

2.1 Schematic Design, Testbench, and Layout of 1-bit ALU

The schematic of the 1-bit ALU is shown in Figure 1a. The corresponding symbol is showed in Figure 1b. Figure 1c illustrates a testbench for the 1-bit ALU. We performed pre-layout simulation for the 1-bit ALU and the simulation results are shown in Figure 1d. Figure 1e shows the layout of the 1-bit ALU. The extracted circuit of the 1-bit ALU is shown in Figure 1f.

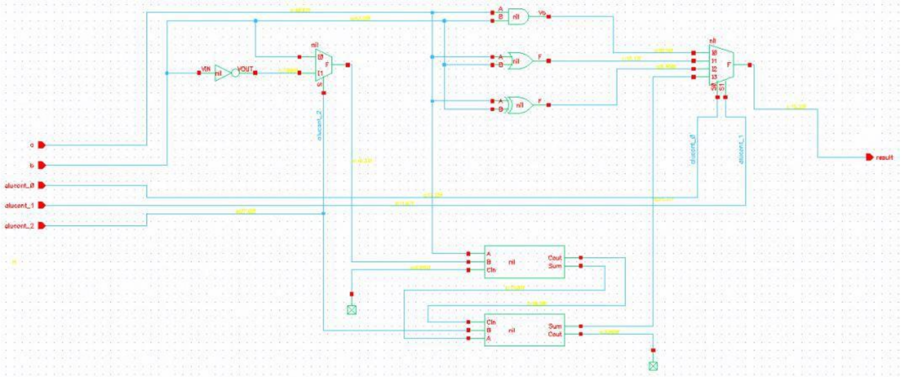
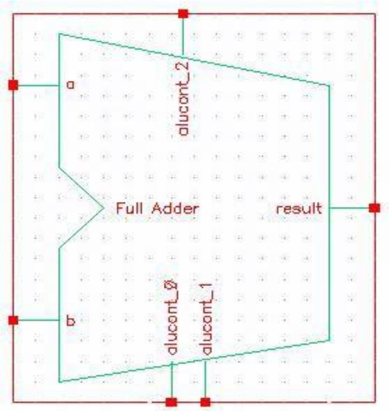


Figure 1a. 1-bit ALU schematics



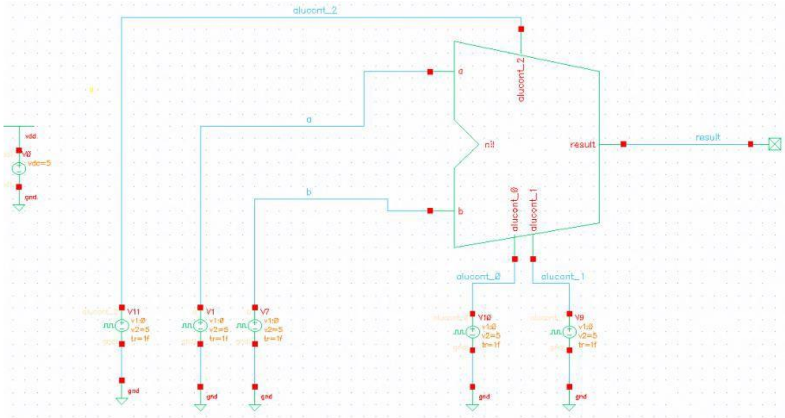


Figure 1b. Symbol of 1-bit ALU

Figure 1c. Testbench for 1-bit ALU

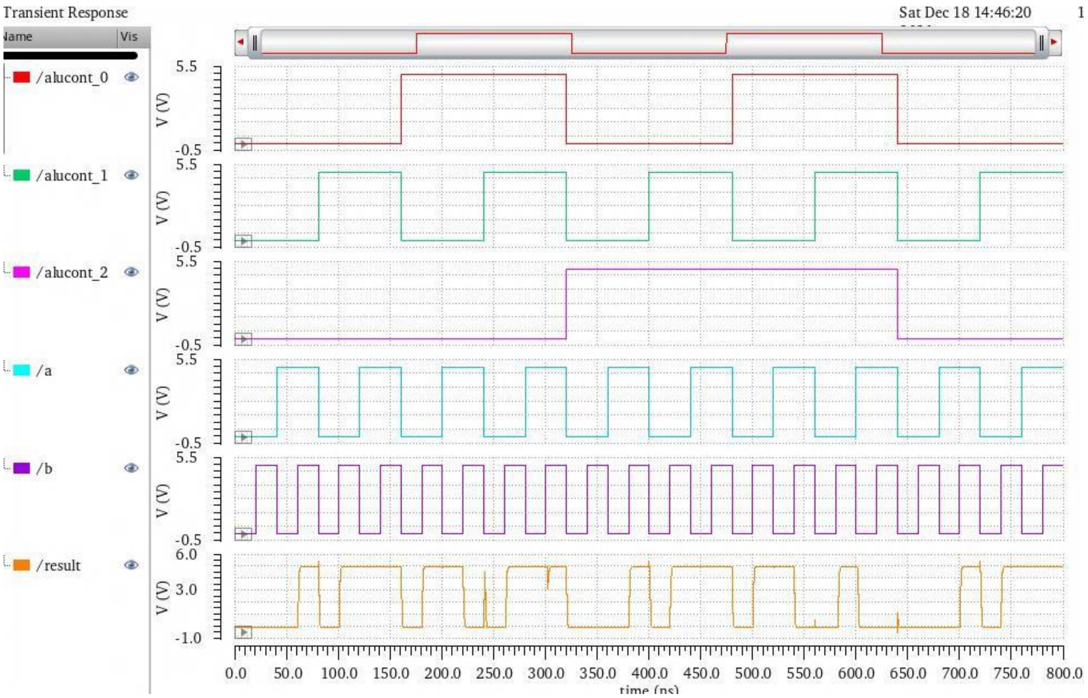


Figure 1d. Output of 1-bit ALU

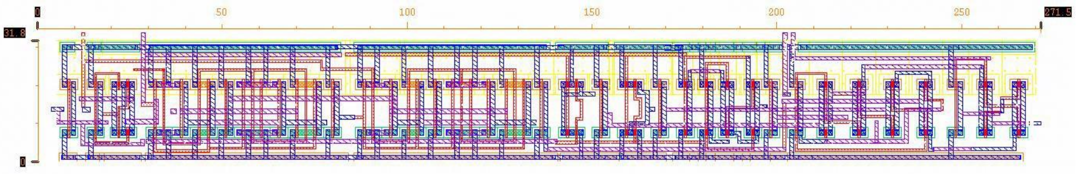


Figure 1e. Layout of 1-Bit ALU

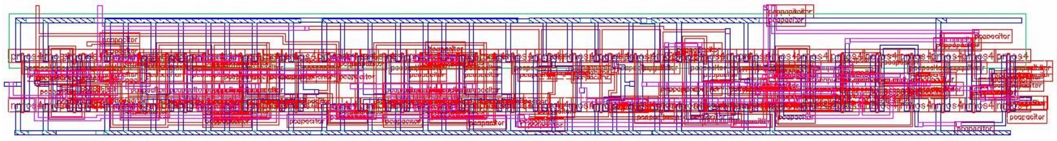


Figure 1f. Extracted view of 1-bit ALU

2.2 The Design of the 8-Bit ALU

The 8-bit ALU is designed to perform a total of five operation including three logical operations and two arithmetic operations. It has 3 control inputs *alucont_0*, *alucont_1*, and *alucont_2*. Each input vector has 8 bits in width, on which ALU operation will be performed. The ALU has 8-bit output data lines. The operation of the ALU is summarized in Table 1 as below.

Table 1. ALU operation and control signals

Alucont_0	Alucont_1	Alucont_2	A<1:8>	B<1:8>	Function	Result<1:8>
0	0	0	10010010	01010001	AND	00010000
0	1	0	10110011	11001111	OR	11111111
1	0	0	10110011	11111111	XOR	01001100
1	1	0	010010000	00000101	ADD	01001101
1	1	1	10010010	01010001	SUB	01000001

The testbench of the 8-bit ALU is shown in Figure 2a.

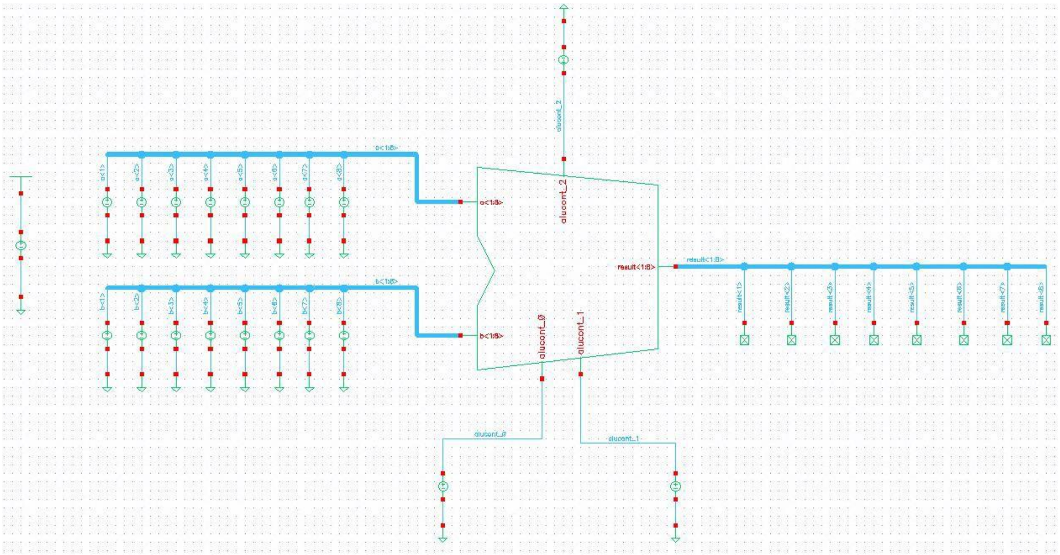


Figure 2a. Testbench for 8-Bit ALU

We performed the pre-layout simulation for the 8-bit ALU. Figure 2b shows the layout of the 8-bit ALU. The extracted circuit of the 8-bit ALU is shown in Figure 2c. The ALU test runs and the simulation results will be demonstrated in Section 4.

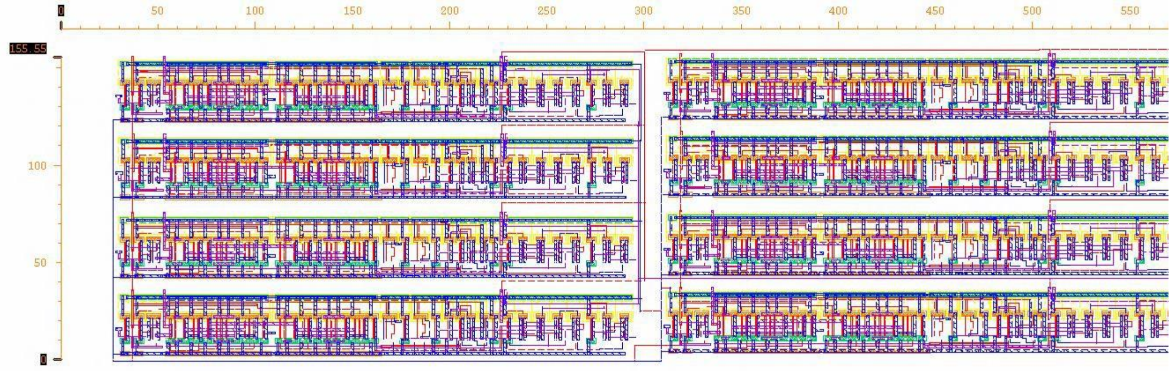


Figure 2b. Layout of 8-bit ALU

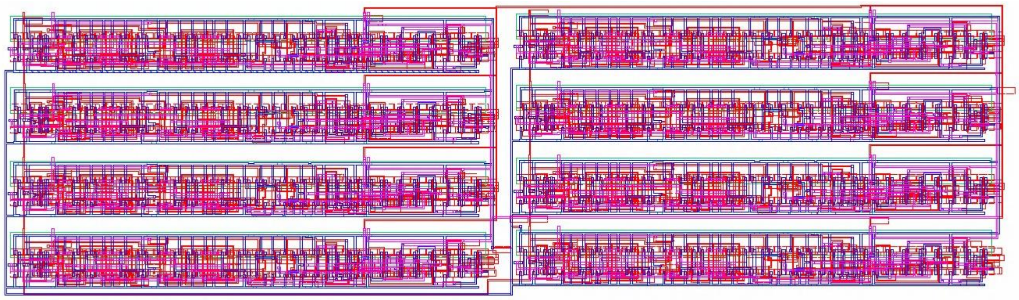


Figure 2c. Extracted view of 8-bit ALU layout

3. Pad-frame with Pin Mappings

The pad-frame is an indispensable part of a design. It makes the connection of the inner circuits and the pins. The pad-frame with pins is shown in Figure 3a. The pad-frame with I/O pins is shown in Figure 3b. There are 40 Pins around the Pad-frame, all the logic circuits of your design should be put inside the square blank in the central and the input/ output ports should be connected to the pins. There are different pins available, some of them are for input/ output, and some of them are for power supply. You can select each pad in the pad-frame, then replace it with any other pads that needed.

Before replacing the pads, you might need to estimate how many input/ output pins and VDD/ GND pins would be used in your chip, try to arrange them properly because this might directly affect the difficulty of wiring.All the wires should use Metal 1, Metal 2 or Metal 3 to connect to the pads, remember to use Poly Contacts or Vias to connect different layers.

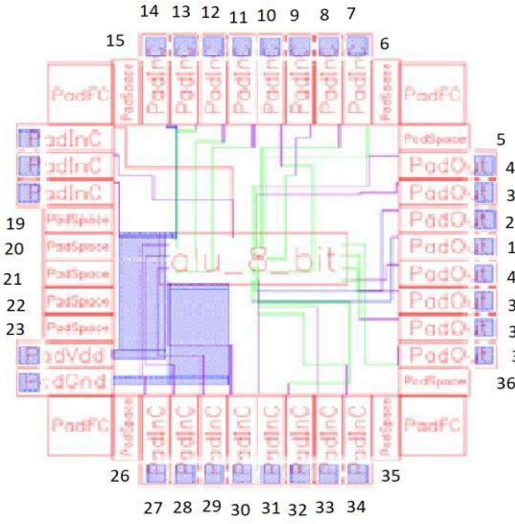


Figure 3a. Pad-Frame with Pins Figure

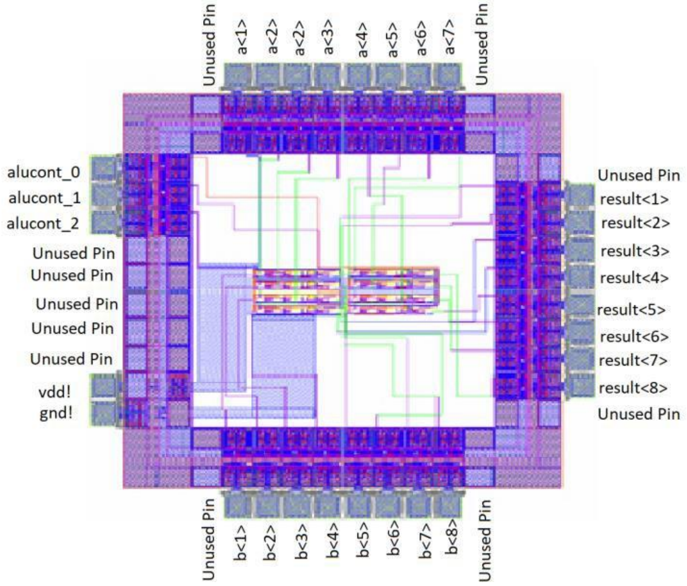


Figure 3b. Pad-Frame with IO pins

The chip representation and the chip representation with logos are shown in Figure 3d and Figure 3e, respectively.

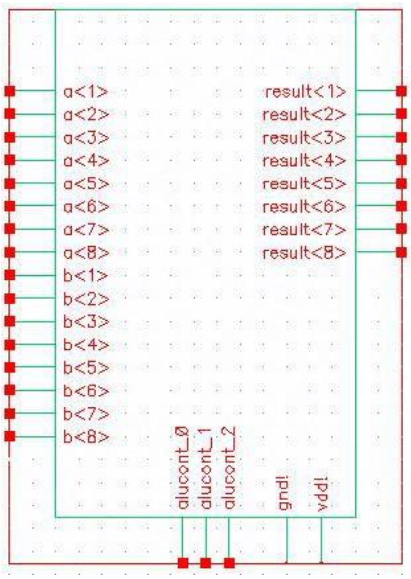


Figure 3d. Chip representation

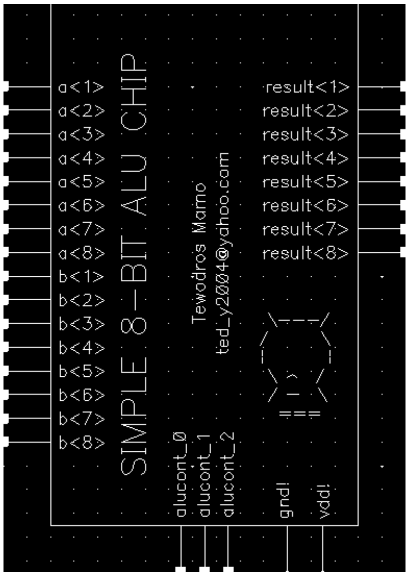


Figure 3e. Chip representation and logo

Table 2 shows the PIN assignment of ALU pad-frame, as below.

Table 2 PIN assignment of ALU pad-frame.

Pin Number	**Pin Name**	**Function**
1	result<4>	output
2	result<3>	output
3	result<2>	output
4	result<1>	output
7	a<8>	Input
8	a<7>	Input
9	a<6>	Input
10	a<5>	Input
11	a<4>	Input
12	a<3>	Input
13	a<2>	Input
14	a<1>	Input
16	alucont_0	Input
17	alucont_1	Input
18	alucont_2	Input
24	vdd!	Input
25	gnd!	Input
27	b<1>	Input
28	b<2>	Input
29	b<3>	Input
30	b<4>	Input
31	b<5>	Input
32	b<6>	Input
33	b<7>	Input
34	b<8>	Input
37	result<8>	output
38	result<7>	output
39	result<6>	output
40	result<5>	output
5,6,15,19,20,21,22,23,26,35,36	Unused Pins	No Connection

4. Test and Measured Results

The MOSIS fabricated chip under microscope (place holder) is showed in Figure 4. The chip is tested on a breadboard using 5V from power supply unit as an input to all data and control inputs. Slide switches are used to control the input voltage between VDD (5V) and ground (0V), and LEDs will be used as an output. However, logic analyzer can also be used if available to test the circuit by providing input waveforms and capturing the output waveforms. The oscilloscope is used to capture and display the outputs, delay and frequency.

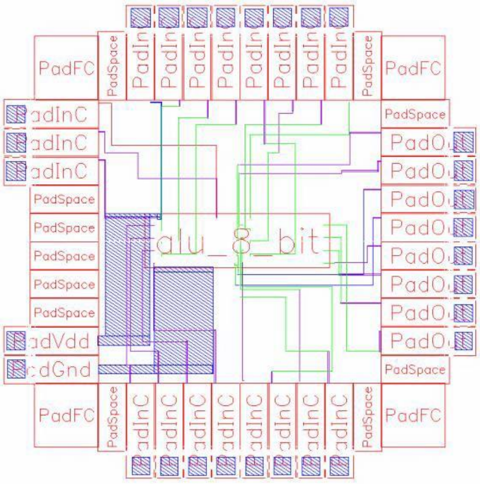


Figure 4. MOSSIS fabricated chip under microscope

After finishing building the layout and applying the pad-frame, we run a final DRC and extraction followed by the simulation. The simulated setup in Cadence is shown in Figure 5.

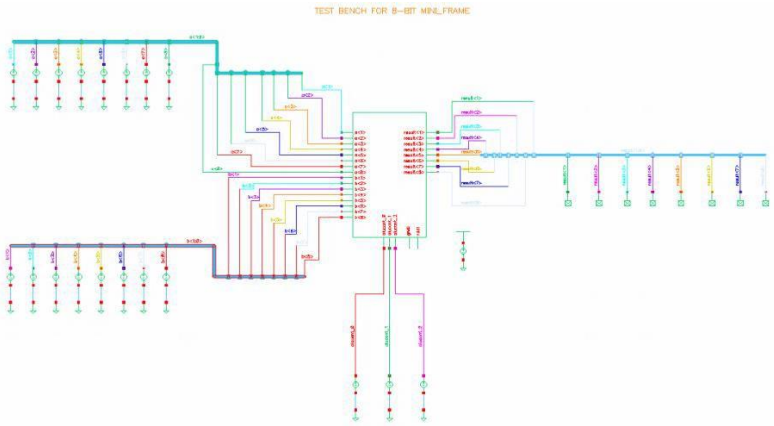


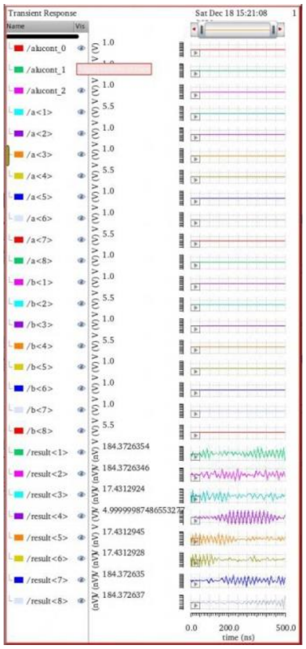
Figure 5. Simulated Setup in Cadence

After performing the simulation on the extracted view of the layout design, we obtained the following results in Table 3, which is exactly what we expect the 8-bit ALU would produce.

Table 3 Simulation Results of the 8-bit ALU

Alucont_0	Alucont_1	Alucont_2	A<1:8>	B<1:8>	Function	Simulated Result<1:8>
0	0	0	10010010	01010001	AND	00010000
0	1	0	10110011	11001111	OR	11111111
1	0	0	10110011	11111111	XOR	01001100
1	1	0	010010000	00000101	ADD	01001101
1	1	1	10010010	01010001	SUB	01000001

Further experiments were conducted to compare the simulated output of AND gate and the measured output of the AND gate. The results are shown in Fig. 6a and Figure 6b, respectively.



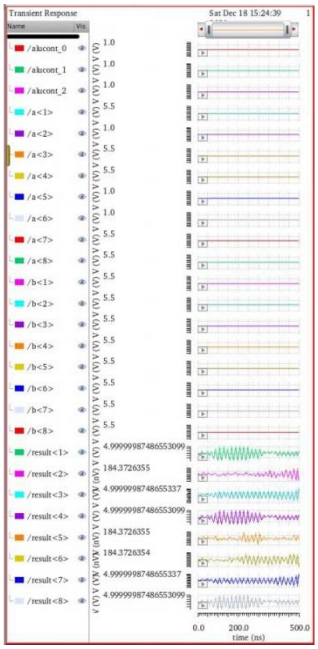
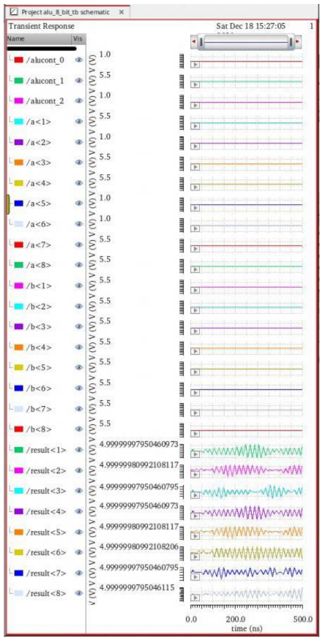


Figure 6a. AND simulated output

Figure 6b. AND measured output

We also compared the simulated output of OR gate and the measured output of the OR gate. The results are shown in Fig. 7a and Figure 7b, respectively.



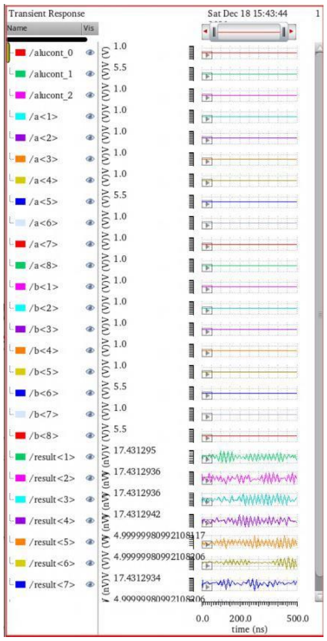
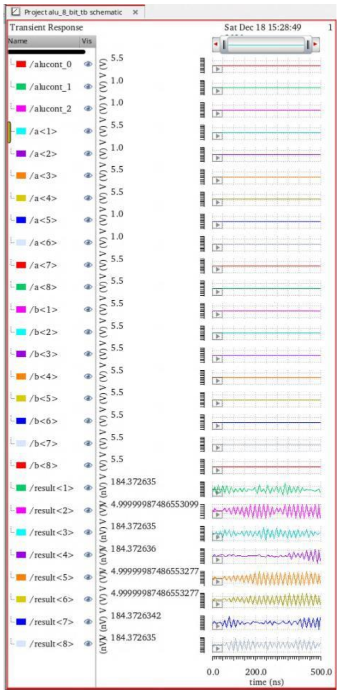


Figure 7a. OR simulated output Figure 7b. OR measured output

Additional experiment was performed to compare the simulated output of XOR gate and the measured output of the XOR gate. The results are shown in Fig. 8a and Figure 8b, respectively.



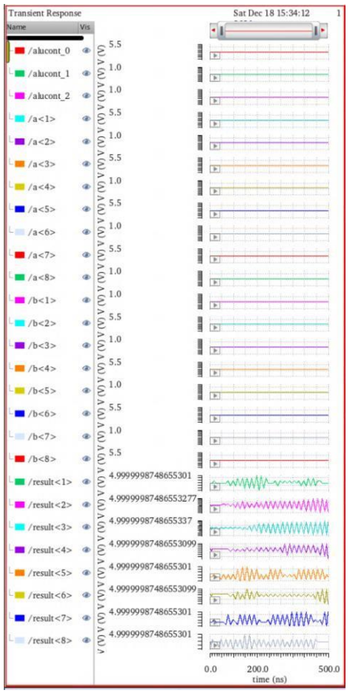
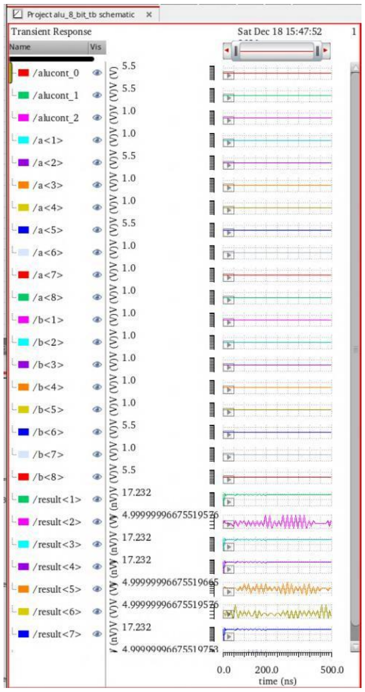


Figure 8a. XOR simulated output Figure 8b. XOR measured output

We also compared the simulated output of addition operation and the measured output of the addition operation. The results are shown in Fig. 9a and Figure 9b, respectively.



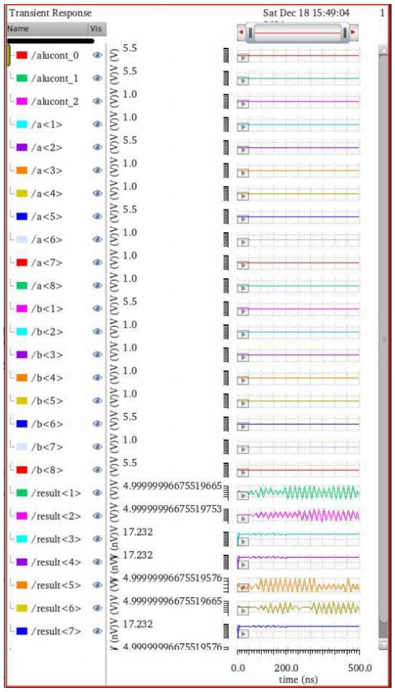
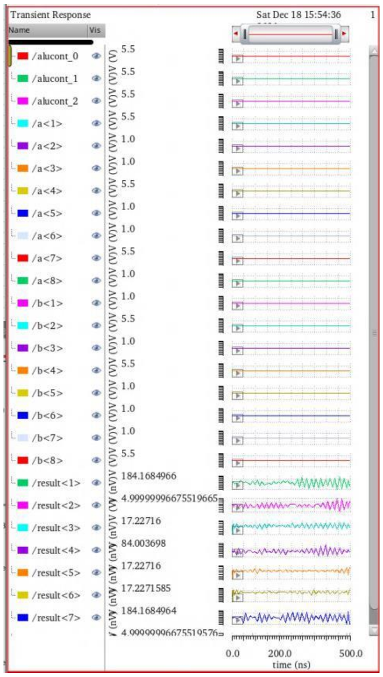


Figure 9a. ADD simulated output

Figure 9b. ADD measured output

Further experiment was performed to compare the simulated output of subtraction operation and the measured output of the subtraction operation. The results are shown in Fig. 10a and Figure 10b, respectively.



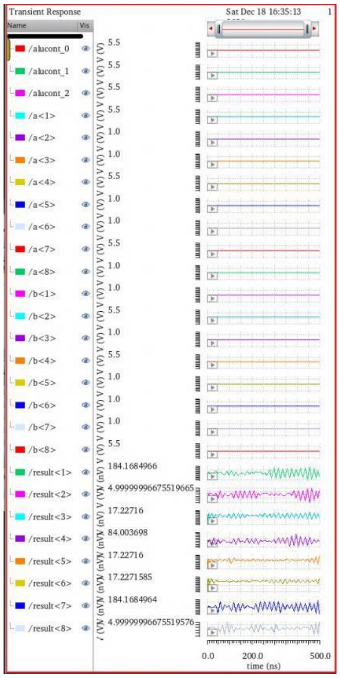


Figure 10a. SUB simulated output

Figure 10b. SUB measured output

5. Conclusions

This paper provides a comprehensive study on the latest version of Cadence Virtuoso on the VLSI design of an 8-bit arithmetic logic unit (ALU). The 8-bit arithmetic logic unit is used as a proof-of-concept example to go through the major VLSI design flow, including schematic capture, pre-layout simulation, physical layout, extract, design rule check (DRC), and layout vs. schematic (LVS). In addition, since this specific ALU is designed to perform three logical operations (i.e. AND, OR, and XOR) and two arithmetic operations (i.e. addition and subtraction), their respective schematic, symbol, testbench, pre-layout simulation, physical layout, post-layout simulation results are demonstrated with detailed snapshots. Moreover, the design layout is wired to a 40-pin MOSIS Tiny Chip pad-frame. The ALU circuit is then simulated along with the pad-frame and the simulation results are analyzed.

Universities and designers around the world will find this paper an invaluable shareable document, which ensures an efficient and fast VLSI design, verification and fabrication using the latest Cadence Virtuoso software.

Acknowledgement

This work was supported in part by the National Science Foundation (NSF) grant #2011927.

References

1. Cadence Virtuoso Tutorial - University of Southern California, 2015. https://ee.usc.edu/~redekopp/ee209/virtuoso/setup/USCVLSI-VirtuosoTutorial.pdf.
2. Cadence Virtuoso Schematic Design and Circuit Simulation Tutorial, 2016. https://www.brown.edu/Departments/Engineering/Courses/engn1600/Assignments/Cadence_Tutorial_EN1600.pdf.
3. Cadence Virtuoso – Schematic & Simulations – Inverter (45nm), 2015. https://sudip.ece.ubc.ca/cadence-virtuoso-schematic-simulations.
4. Virtuoso Schematic Composer Tutorial, 2003. http://www2.ece.ohio-state.edu/~bibyk/ee721/comptut.pdf.
5. Cadence Virtuoso Schematic Composer Introduction, 2009. https://engineering.purdue.edu/~vlsi/ECE559_Fall09/Lab/559TUT_1_ver7_fall09.pdf.
6. N. Zhang, J. F. R. Rochac, E. T. Ososanya, W. H. Mahmoud, and S. Lakeou, "VLSI Design and Verification of a CMOS Inverter Using the Tanner EDA: A Case Study," *The 7th International Multi-Conference on Engineering and Technological Innovation (IMETI 2014),* Orlando, Florida, July 15-18, 2014.
7. N. Zhang and D. C. Wunsch II, "Speeding up VLSI Layout Analysis and Verification Using Fuzzy Attributed Graph Approach," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 6, pp. 728-737, December 2006.
8. N. Zhang and D. C. Wunsch II, "VLSI Layout Verification with Neural Networks," *The Eighth International Conference on Cognitive and Neural Systems (ICCNS)*, Boston University, Boston, Massachusetts, May 19-22, 2004.
9. N. Zhang and D. C. Wunsch II, "A Novel Subcircuit Extraction Algorithm Using Heuristic Dynamic Programming," *The International Conference on VLSI*, Las Vegas, pp. 38-44, June 24-27, 2002.
10. G. S. Yuvashree, S. Sumanth,"A 8 Bit ALU Design using Cadence", *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 8, Issue VIII, pp. 162-182, 2020.
11. M. Morris Mano and Michael Ciletti, Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog, 6th Edition, Pearson, 2017.