Optimal 500MHz Arithmetic Logic Unit

Yiqiu Li
Jiawei Liang

January 20, 2019

Abstract

In this project, we complete the basic part with worst case delay of 1.03ns(post-simulation). For bonus part, we complete both schematic and layout for improved 4-bit ALU; schematic and layout except signed multiplier for 8-bit ALU. Due to time constraint, we only optimal the basic part and our optimal principle is that find out the simplest circuit configuration that we can realize; sizing the circuit; make the layout as dense as possible. After practising, we find that the most efficient way of optimization is to simplify the circuit configuration by reducing length of the critical path. In this report, we introduce our design decisions briefly, show our simulation result and raise some improvements that we can made in the future.

1 Introduction

The basic goal of this project is to design a simple ALU which takes a 10-bit instruction as input and perform addition, subtraction, unsigned multiplier or comparator then output the result in 8-bit 2s complement format with the minimum energy under the worst- case delay of 2 ns. The goal of bonus part is to design a simple ALU which takes 19-bit instruction as input and perform addition, subtraction, signed multiplier comparator or absolute-value comparator then output the result in 16-bit 2s complement format without considering the worst case delay.

2 Circuit Design

2.1 Adder

There are two main parts in adder as showing in Figure1(the same for 8-bit). One is for calculation and another is to put the output in 2's complement format. In the calculation part simply put four 1-bit full adder in series because ripple adder is faster when input is 4-8 bits. The logic in the part which is to deal with the 2's complement is that if the most significant bit of two inputs are the same abandon the carry in the last 1-bit full adder and let all undefined bit equal to the sum of the last 1-bit full adder otherwise let all undefined bits equal to the carry of the last 1-bit full adder. For example, for 4-bit adder.If $A3 \bigoplus B3 = 1$ then S7=S6=S5=S4=S3 because in this case S3 is the sign bit. If $A3 \bigoplus B3 = 0$ then

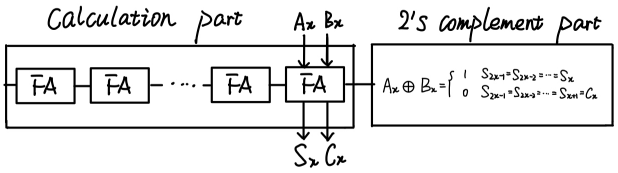

Figure 1: adder

S7=S6=S5=S4 because in this case S4 is the sign bit.

2.2 Subtraction

A-B=$A + \bar{B} + 1$. The structure is much similar to adder. Take A and the inverse of B as two inputs and let $C_{in} = 1$ in the first 1-bit full adder. Others are the same as adder.

2.3 4-bit unsigned multiplier

All the inputs are non-negative so the most significant bits are zeros. Simplify the 4×4 array multiplier which is provided in Lecture 11 and get a faster multiplier for the particular inputs in this project. The realization is shown in Figure 2.

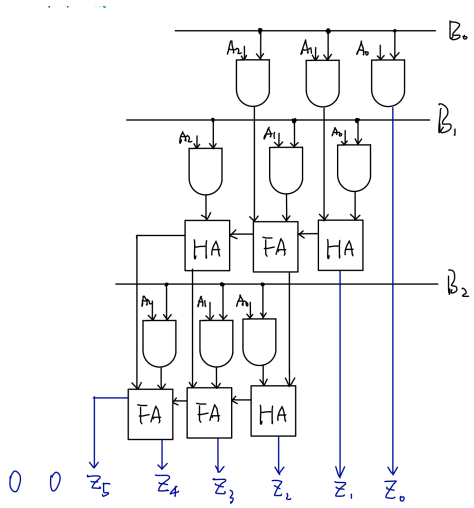

Figure 2: multiplier

2.4 comparator

The logic is shown in the below formulation.

$$A > B : \overline{\overline{A3}B3 + X3A2\overline{B2} + X3X2A1\overline{B1} + X3X2X1A0\overline{B0}}$$
$$= \overline{(\overline{A3}B3 + X3A2\overline{B2})}\,\overline{X3X2(A1\overline{B1} + X1A0\overline{B0})}$$
$$= \overline{\overline{A3}B3} \quad \overline{X3A2\overline{B2}} \quad (\overline{X3X2} + \overline{A1\overline{B1} + X1A0\overline{B1}})$$

$$A = B : X3X2X1X0 = \overline{\overline{X3X2X1X0}}$$
$$A < B : \overline{A > B + A = B}$$

with $Xn = An \bigoplus Bn$

2.5 signed multiplier
Apply Baugh-Wooley algorithm, the analysis processes are shown in Figure 3 & 4

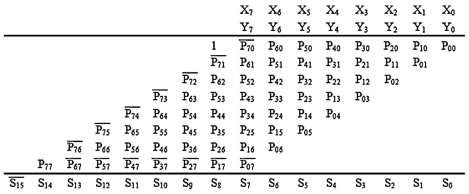

Figure 3: signed multiplier1

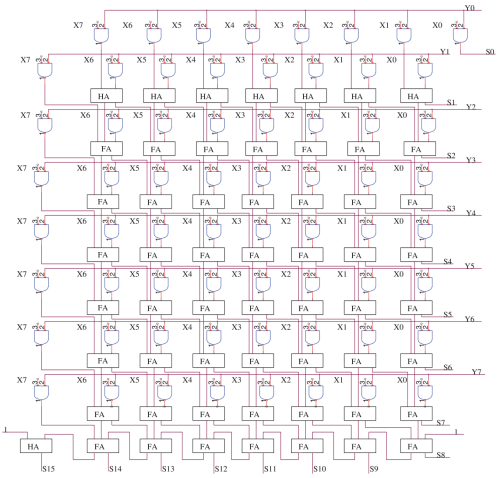

Figure 4: signed multiplier2

2.6 absolute-value comparator
$|A| < |B| : (A + B < 0)(A < B)$
$|A| = |B| : \overline{(A + B = 0)(A = B)}$
$|A| > |B| : \overline{(A = B) + (A < B)}$
Besides, get the compare result of A and B though comparator; get the result of A+B though adder.

2.7 the overall configuration
Do the four different kinds of calculation at the same time and then select the answer that

required though applying 8 (or 16) four-choice selector (or five-choice selector). See the overall configuration as below.

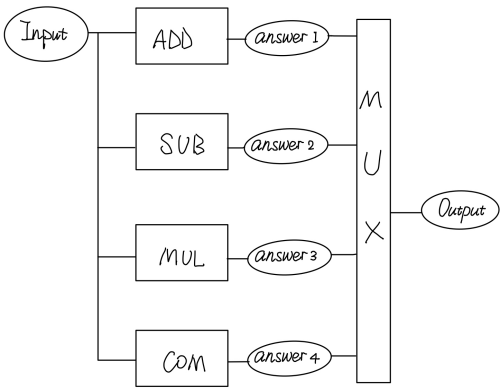

Figure 5: overall configuration

3 Optimization

3.1 circuit configuration optimization

3.1.1 adder

Apply the following structure instead of using inverter after every full adder so reduce $3t_{inv}$. Besides, applying transmission gate for xor also reduce the delay.

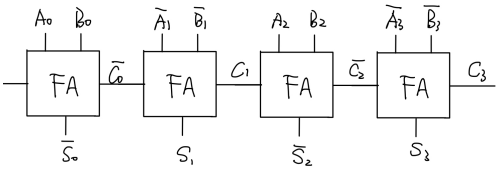

Figure 6: adder

3.1.2 multiplier

Since A3=B3=0 for non-negative input, it is enough to apply a 3×3 array multiplier so only use three 1-bit half adder and 1-bit full adder and the longest path passes three half adders and two half adder.(The configuration see Figure 2)

3.1.3 comprartor

Apply CMOS logic instead of introducing subtraction straightly so the delay is much smaller the other three calculation blocks. If introduce subtraction in this part this path will be worst case delay.

3.2 Sizing

Apply the sizing skill learned in class to make the all the transistors' worst case delay are equal to that of a unit-sized inverter has a sizing ratio of$W_P : W_N = 2 : 1$

3.3 Layout optimization
Use one kind of metal for all vertical lines and other metal for all horizontal line so in the end only use two metal layers for the whole project.

4 Simulation result

The worst case is 1111-1111 or 0111*0111. The critical path for 1111-1111 passes one inverter and four full adders(without inverter between) one XOR(based on transmission gate) and one transmission gate.The critical path for 0111*0111 passes one AND and three full adders(with inverter between) and two half adder. Because the result of 1111-1111 is 00000000 and we cannot get the delay time though the traditional method, we use 0111*0111 as worst case. Here is the simulation result for our basic part. The left one is pre-simulation and the right one is post-simulation. The delay time for pre-simulation is 0.757ns and 1.03226 for post-simulation when VDD=2V.

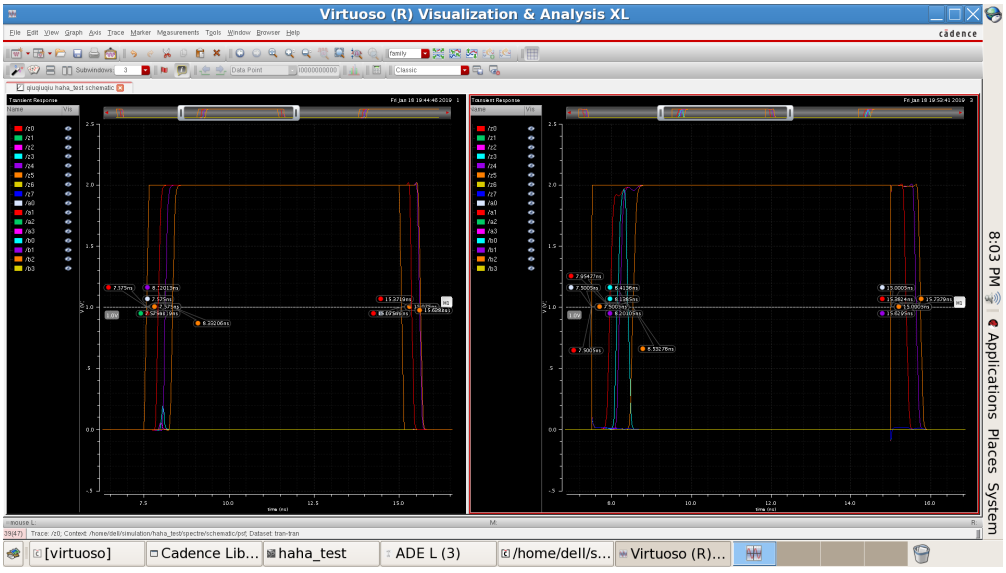

Figure 7: simulation result

5 Conclusion

The project reaches the expected goal and in the end we want to highlight some features and future improvements. In addition, more things can be found in appendices
5.1 Three most important features of our design

- We only use two metal layers which is the minimum amount of layers that we can use. It is a good way to use one metal for vertical lines and another for horizontal lines.
- When drawing layout, it is a fast way to draw all vertical lines and then only draw horizontal lines. This small trick can not only increase the speed of drawing layout but also reduce the possibility of short circuit.
- When considering optimization, consider the circuit configuration first, sizing next and then a better layout.

5.2 Three things that we can improve in the future

- Apply transmission gate adder instead of mirror adder
- better organize the layout to make it looks more tidy
- Apply Wallace Tree Architecture for 8 or more bit ALU

6 Appendices

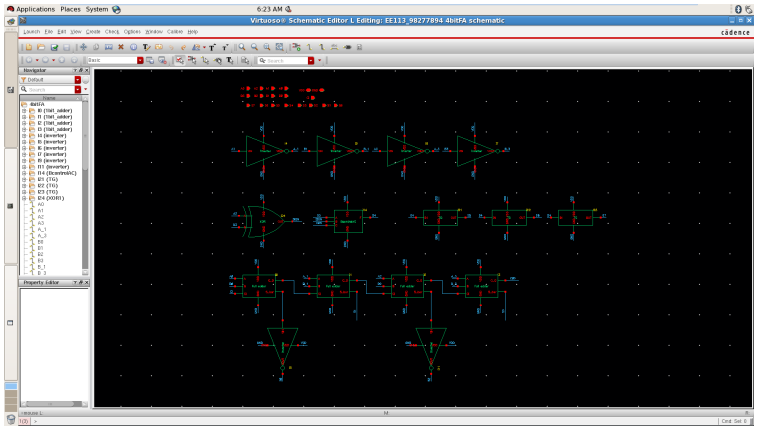

Figure 8: 4-bit adder schematic

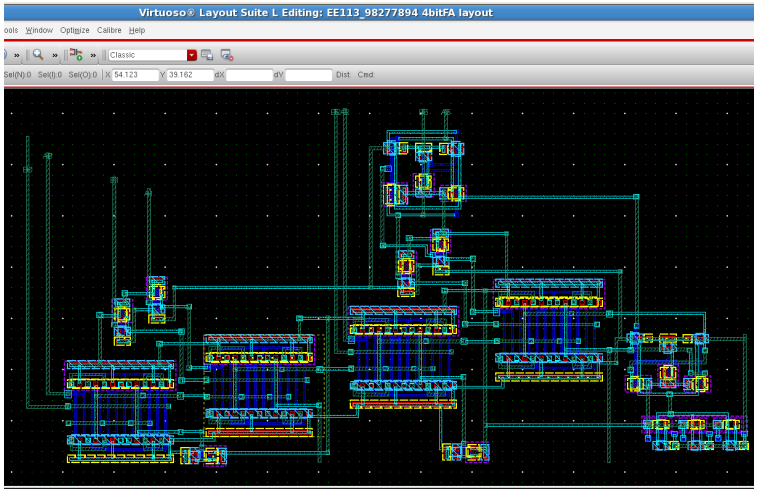

Figure 9: 4-bit adder layout

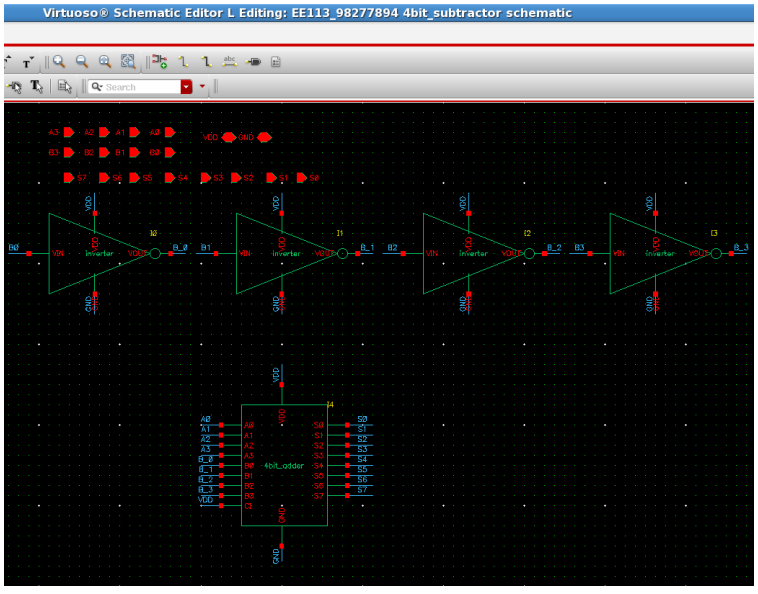

Figure 10: 4-bit subtraction schematic

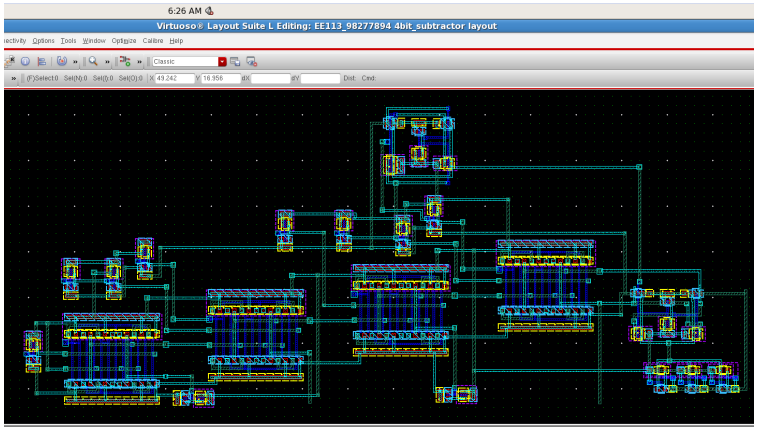

Figure 11: 4-bit subtraction layout

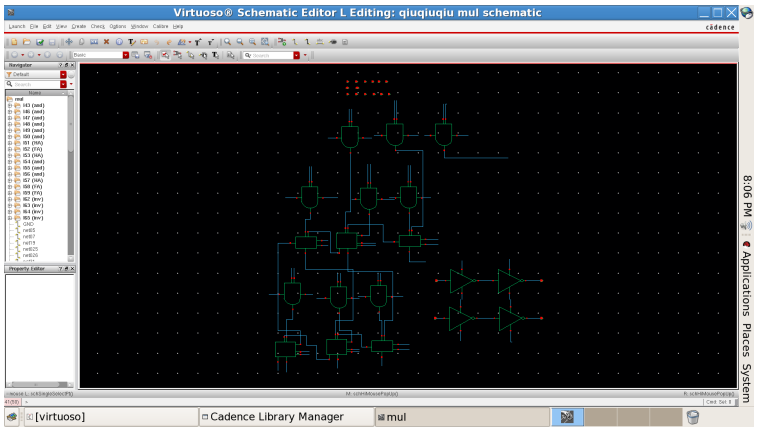

Figure 12: 4-bit unsigned multiplier schematic

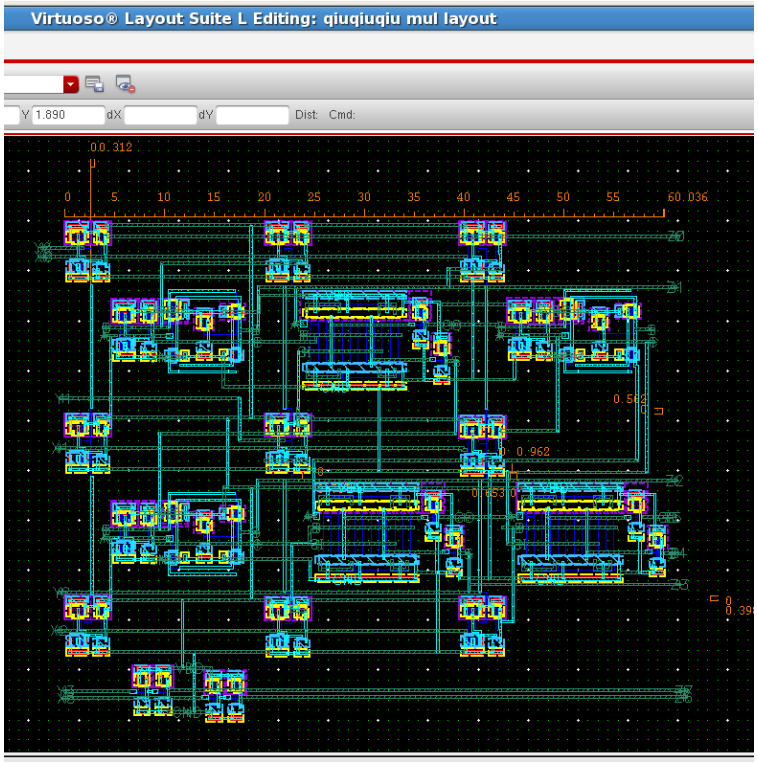

Figure 13: 4-bit unsigned multiplier layout

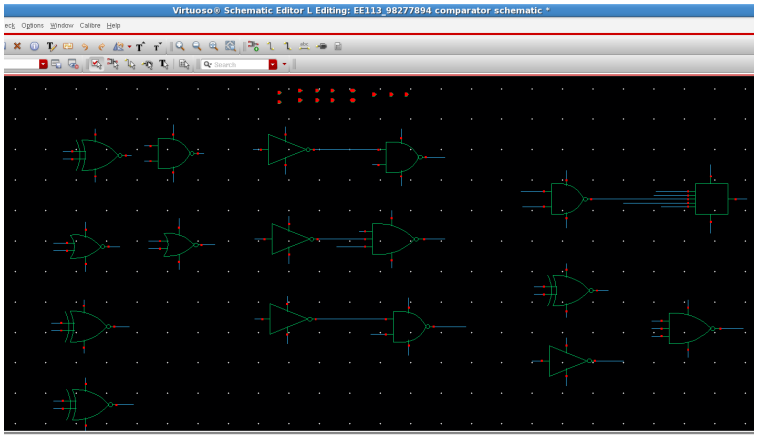

Figure 14: 4-bit comparator schematic

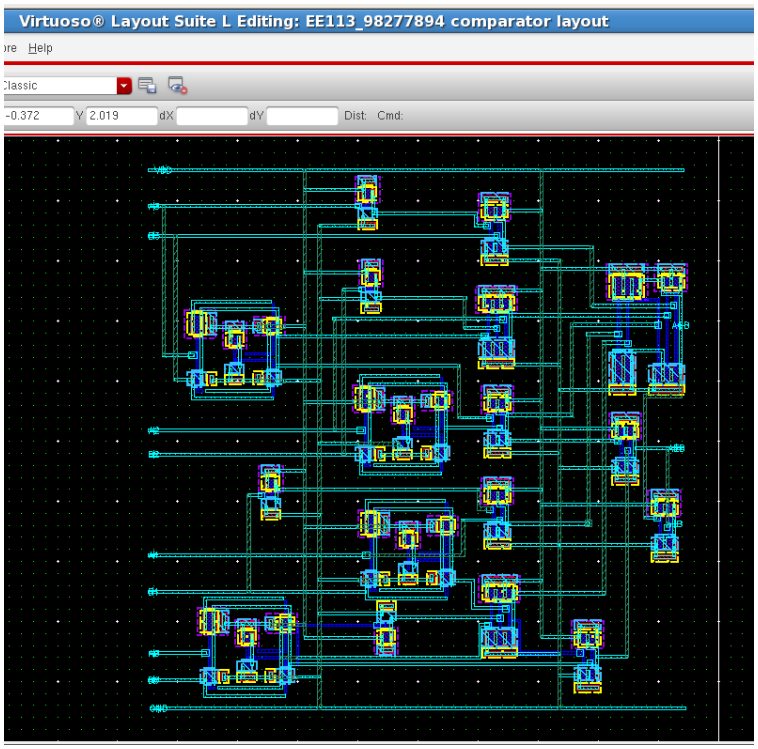

Figure 15: 4-bit comparator layout

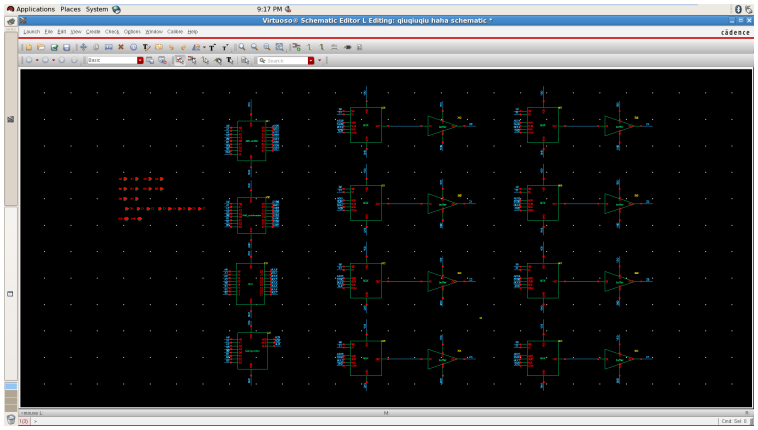

Figure 16: basic part overall schematic

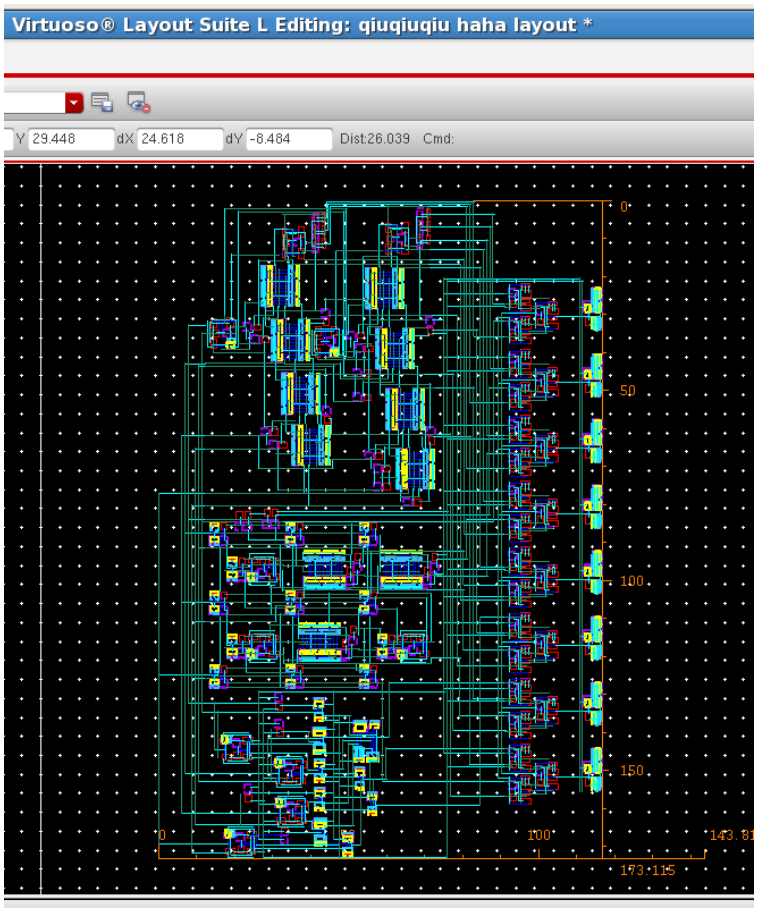

Figure 17: overall basic part schematic

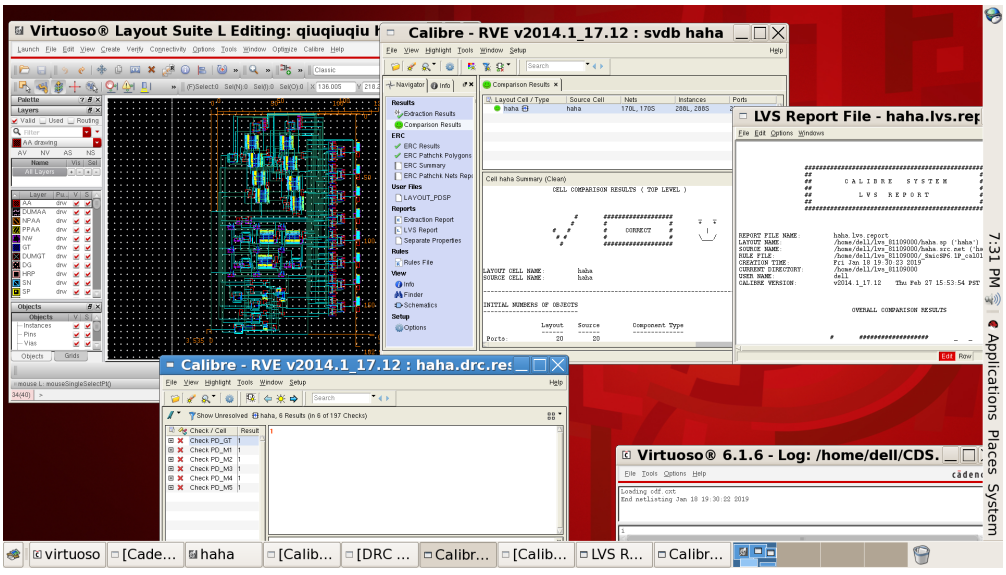

Figure 18: overall layout with drc and lvs report

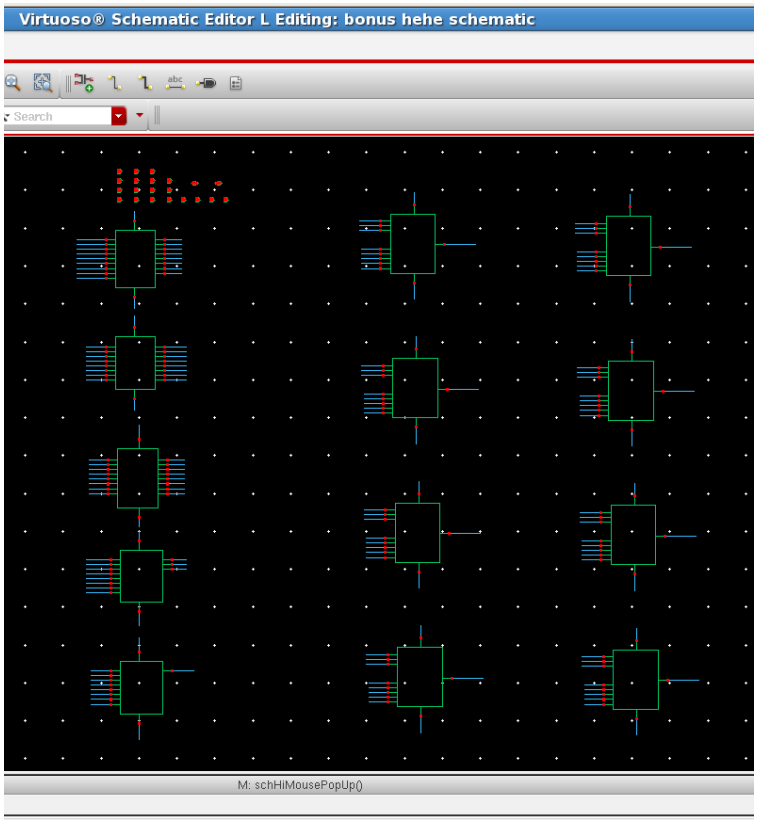

Figure 19: overall schematic for bonus 1&2

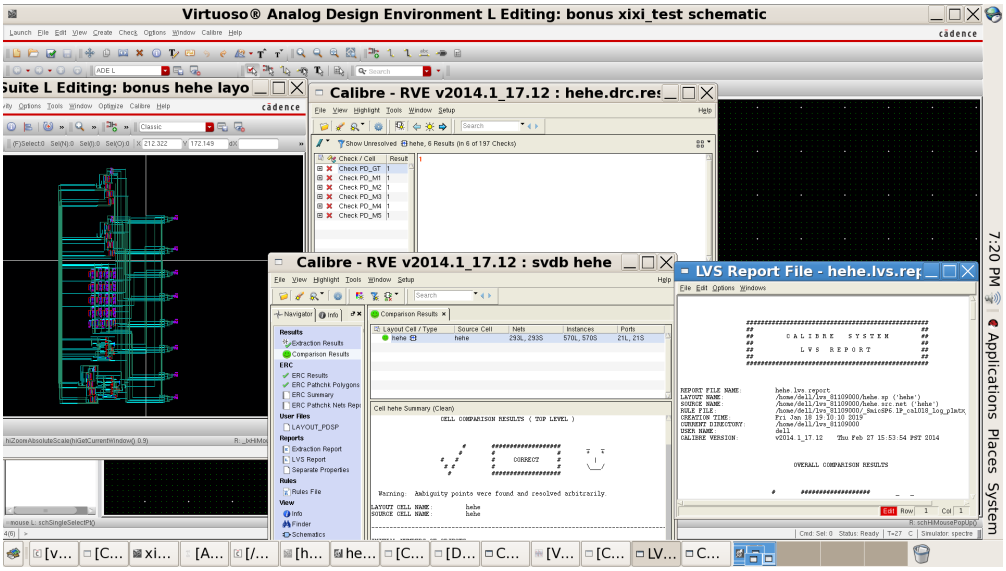

Figure 20: overall layout for bonus 1&2 with drc and lvs report

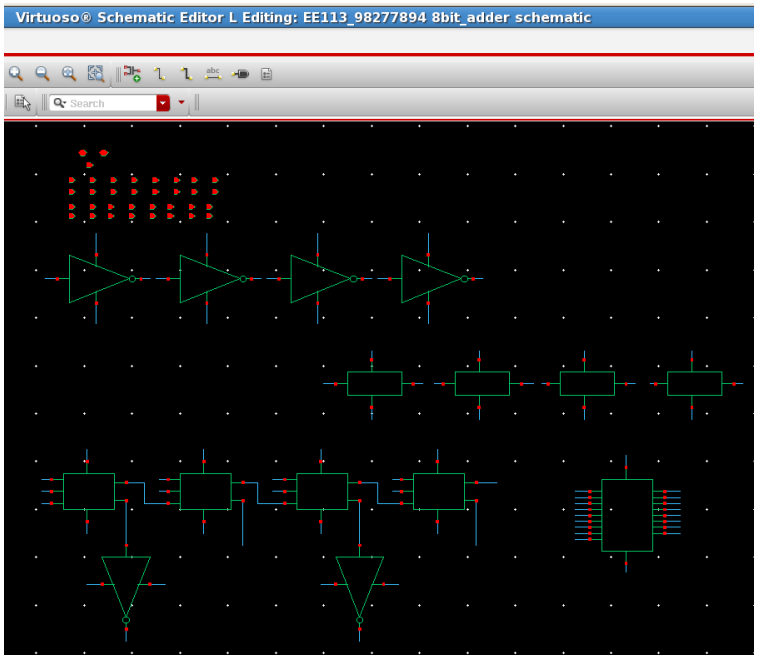

Figure 21: 8-bit adder schematic

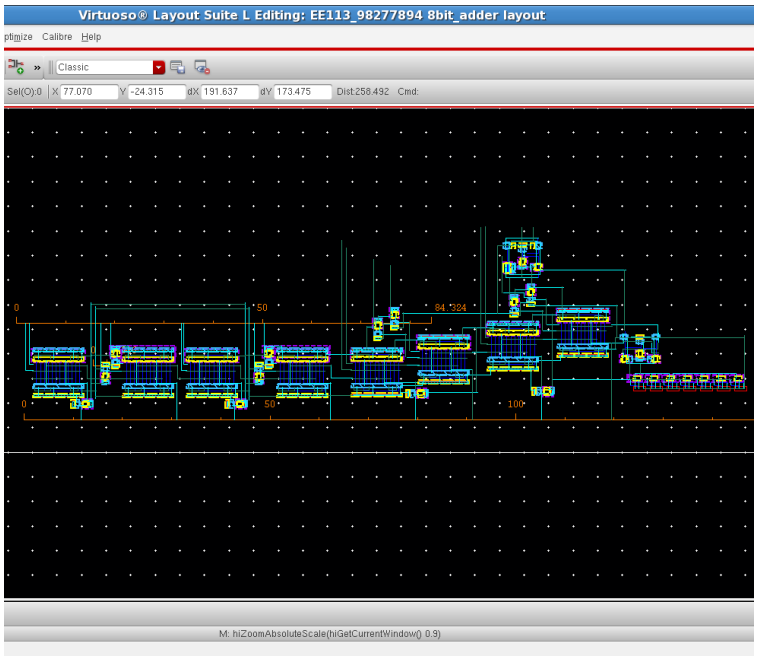

Figure 22: 8-bit adder layout

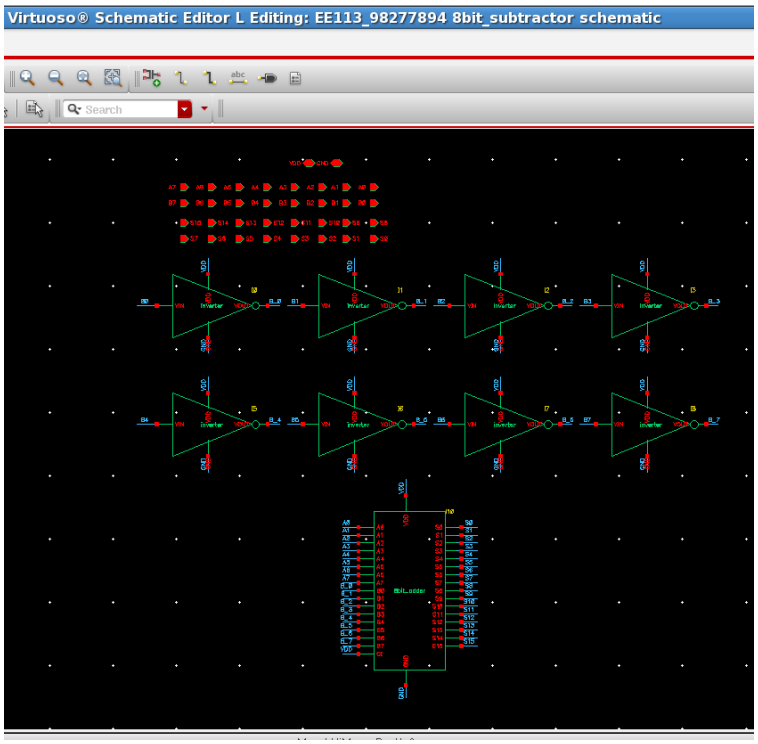

Figure 23: 8-bit subtraction schematics

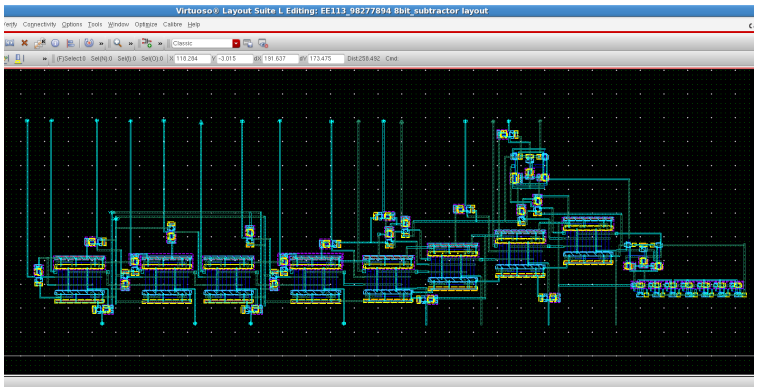

Figure 24: 8-bit subtraction layout

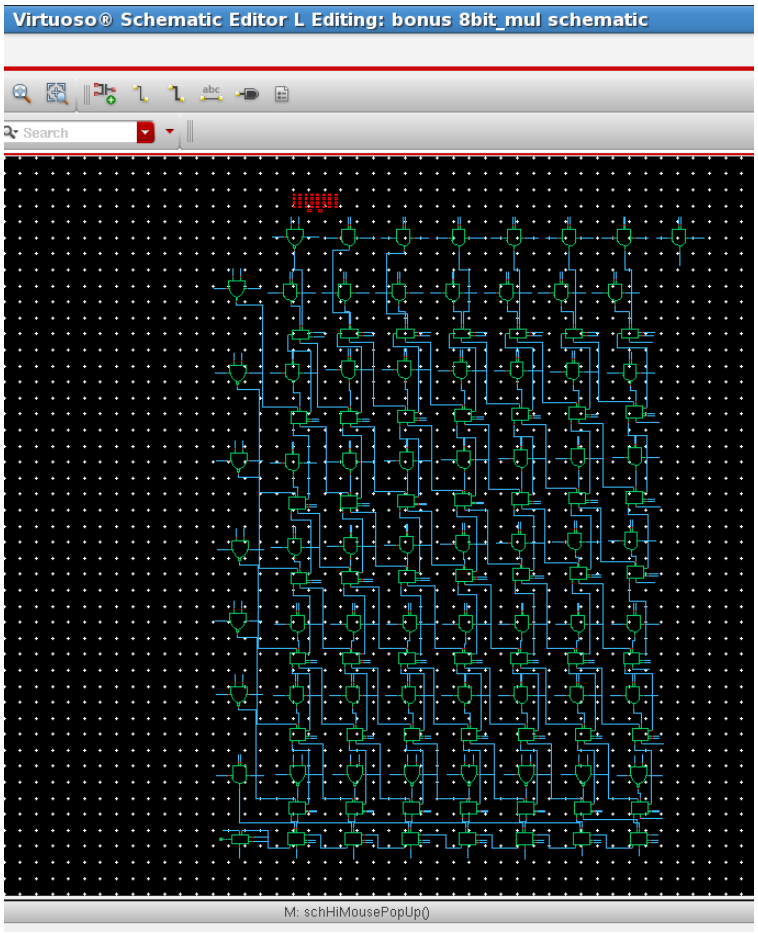

Figure 25: 8-bit multiplier schematic

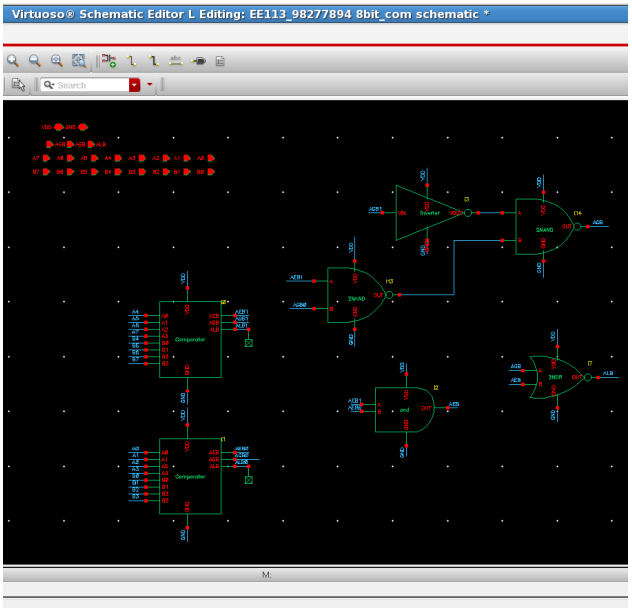

Figure 26: 8-bit comparator schematic

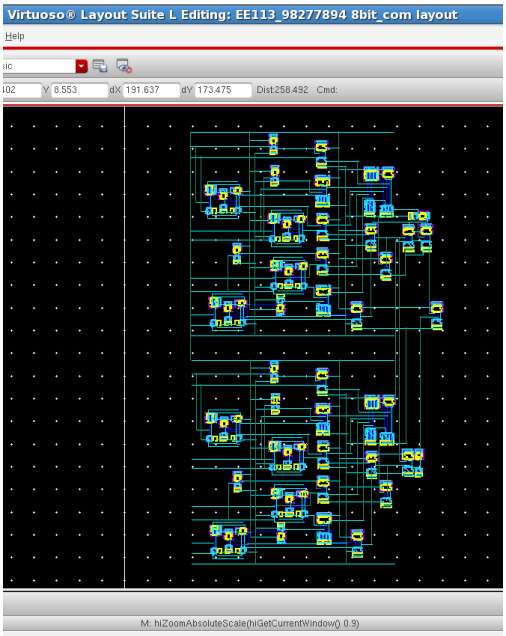

Figure 27: 8-bit comparator layout

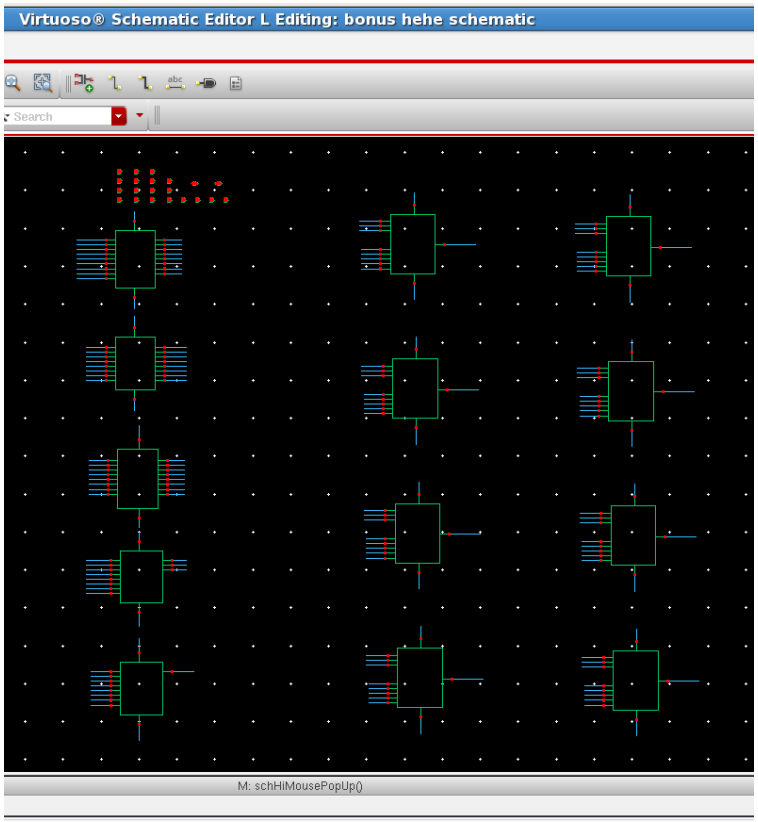

Figure 28: 8-bit ALU schematic