EECS 4612: Digital Very Large Scale Integration

Section Z

Project 1: 12-Bit Adder with the Lowest EDP

By: Ariel Laboriante

Student Number: 212951984

Date of Submission: February 29, 2016

Instructor: Sebastian Magierowski

EECS 4612 Project 1: 12-Bit Adder with the Lowest EDP

Part 1. Adder design schematics

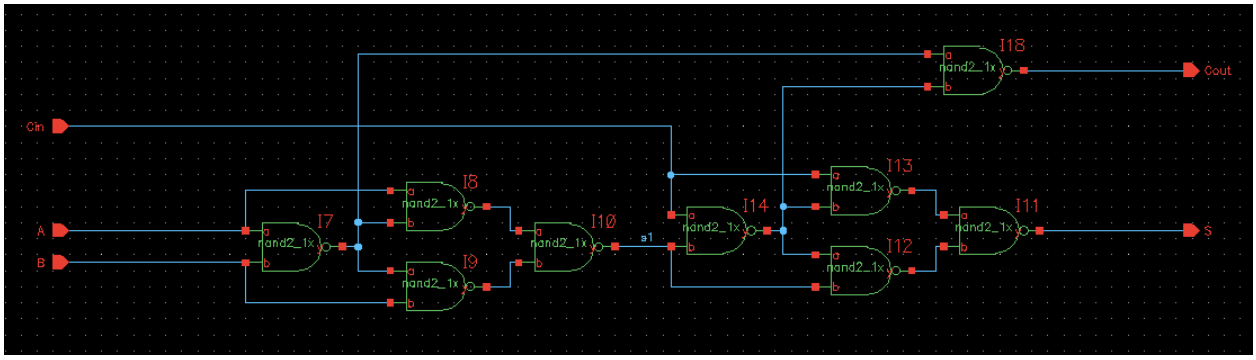

Figure 1: `al_fulladder` schematic

The 1-bit full adder schematic was changed so that it only included NAND gates.

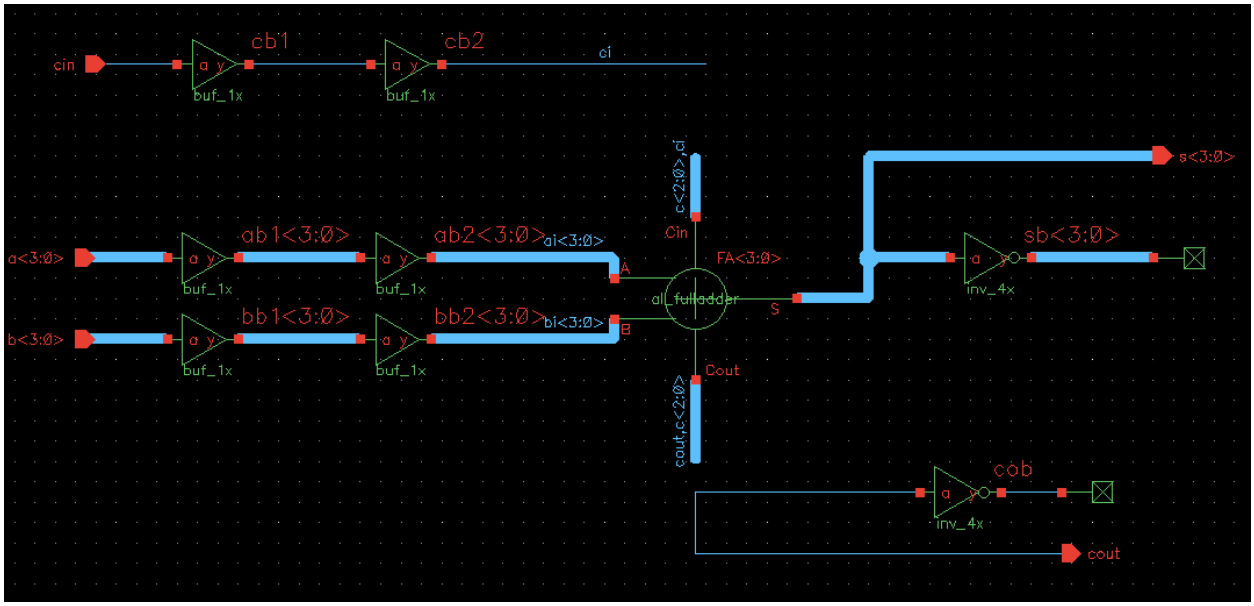

Figure 2: `ripple_adder4` schematic

The ripple-carry adder schematic was changed slightly so that it added 4-bits, produced a 4-bit sum, and a 1-bit carry out.

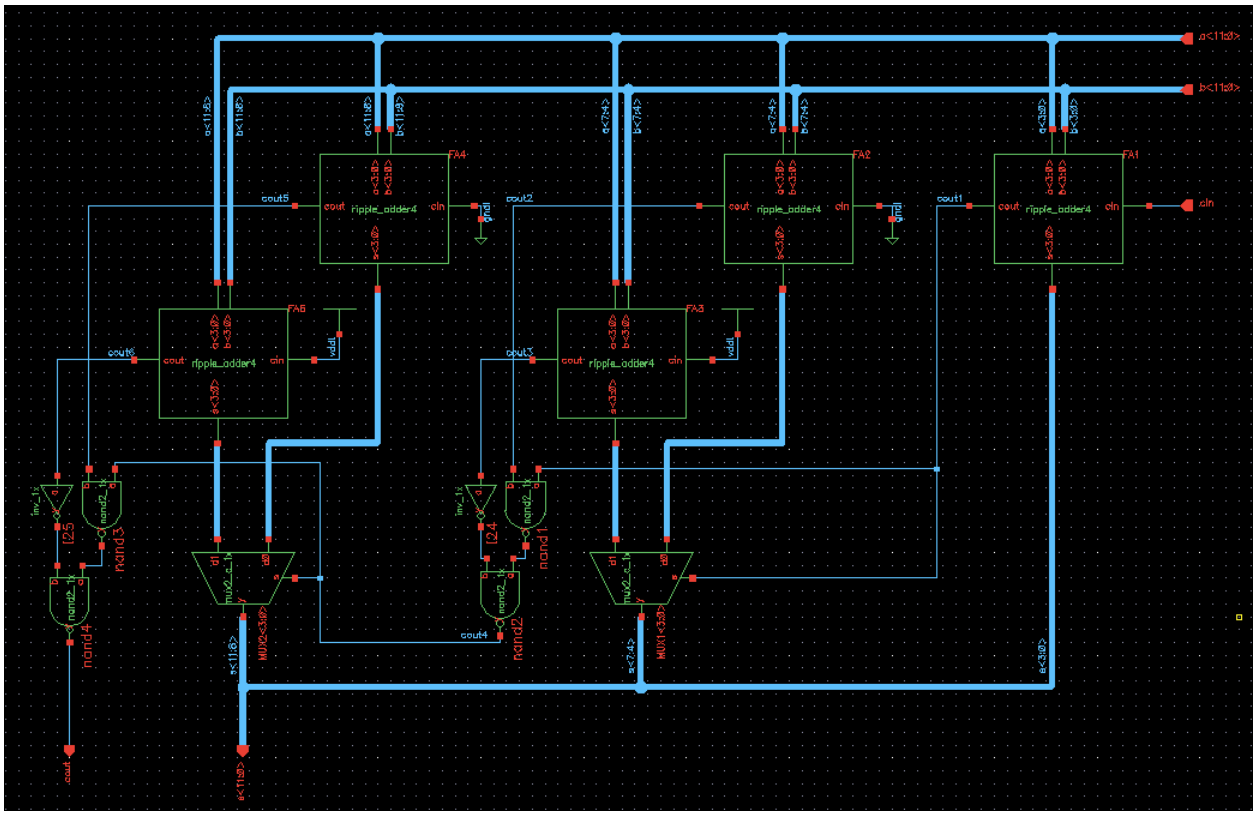

Figure 3: `carryselect_adder` schematic

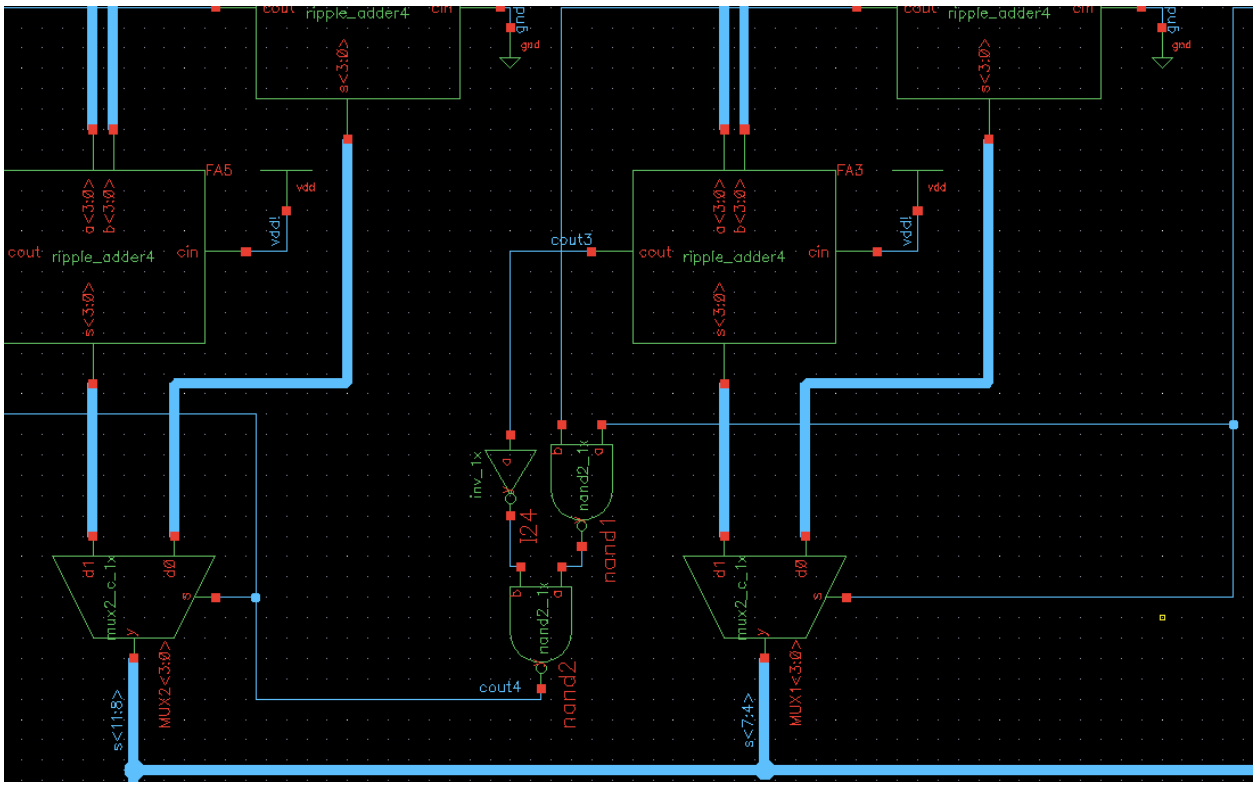

Figure 4: close up view of `carryselect_adder` schematic

The implementation of the linear carry select adder used five 4-bit ripple carry adders, two MUXes, two inverters and four NAND gates.

Part 2. Test plan and adder.tv

Table 1: Test plan for the carry select adder

First Operand	Second Operand	Expected result
0000 0000 0001	1111 1111 1111	1 0000 0000 0000
0000 0000 0001	0111 1111 1111	0 1000 0000 0000
0000 0000 0001	1011 1111 1111	0 1100 0000 0000
0000 0000 0001	1101 1111 1111	0 1110 0000 0000
0000 0000 0001	1110 1111 1111	0 1111 0000 0000
0000 0000 0001	1111 0111 1111	0 1111 1000 0000
0000 0000 0001	1111 1011 1111	0 1111 1100 0000
0000 0000 0001	1111 1101 1111	0 1111 1110 0000
0000 0000 0001	1111 1110 1111	0 1111 1111 0000
0000 0000 0001	1111 1111 0111	0 1111 1111 1000
0000 0000 0001	1111 1111 1011	0 1111 1111 1100
0000 0000 0001	1111 1111 1101	0 1111 1111 1110
0000 0000 0001	1111 1111 1110	0 1111 1111 1111
1111 1111 1111	0000 0000 0001	1 0000 0000 0000
0111 1111 1111	0000 0000 0001	0 1000 0000 0000
1011 1111 1111	0000 0000 0001	0 1100 0000 0000
1101 1111 1111	0000 0000 0001	0 1110 0000 0000
1110 1111 1111	0000 0000 0001	0 1111 0000 0000
1111 0111 1111	0000 0000 0001	0 1111 1000 0000
1111 1011 1111	0000 0000 0001	0 1111 1100 0000
1111 1101 1111	0000 0000 0001	0 1111 1110 0000
1111 1110 1111	0000 0000 0001	0 1111 1111 0000
1111 1111 0111	0000 0000 0001	0 1111 1111 1000
1111 1111 1011	0000 0000 0001	0 1111 1111 1100
1111 1111 1101	0000 0000 0001	0 1111 1111 1110
1111 1111 1110	0000 0000 0001	0 1111 1111 1111
0000 0000 0001	0000 0000 0001	0 0000 0000 0010
0000 0000 0010	0000 0000 0010	0 0000 0000 0100
0000 0000 0100	0000 0000 0100	0 0000 0000 1000
0000 0000 1000	0000 0000 1000	0 0000 0001 0000
0000 0001 0000	0000 0001 0000	0 0000 0010 0000
0000 0010 0000	0000 0010 0000	0 0000 0100 0000
0000 0100 0000	0000 0100 0000	0 0000 1000 0000
0000 1000 0000	0000 1000 0000	0 0001 0000 0000
0001 0000 0000	0001 0000 0000	0 0010 0000 0000
0010 0000 0000	0010 0000 0000	0 0100 0000 0000
0100 0000 0000	0100 0000 0000	0 1000 0000 0000
1000 0000 0000	1000 0000 0000	1 0000 0000 0000

With 12 bits for each operand, we have a total of 2^{24} = 16, 777, 216 combinations of bits. It would be tedious and time consuming to test for all of the possible operands. This test plan focuses on the carry propagation since the adders are assumed to work correctly.

The first test adds 1 to 1111 1111 1111 which causes the result to be zero with a carry out of 1. The next 12 tests add 1 to the bit patterns with a single "0" like 1011 1111 1111. All the bits to the right should be set to zero and all the bits to the left should be set to 1 (excluding the carry out).

The next 13 tests check the commutative property of the carry select adder.

The last 12 tests add the 12 "powers of two" to see if the bits shift correctly as expected.

Part 3. Transistor-level netlist

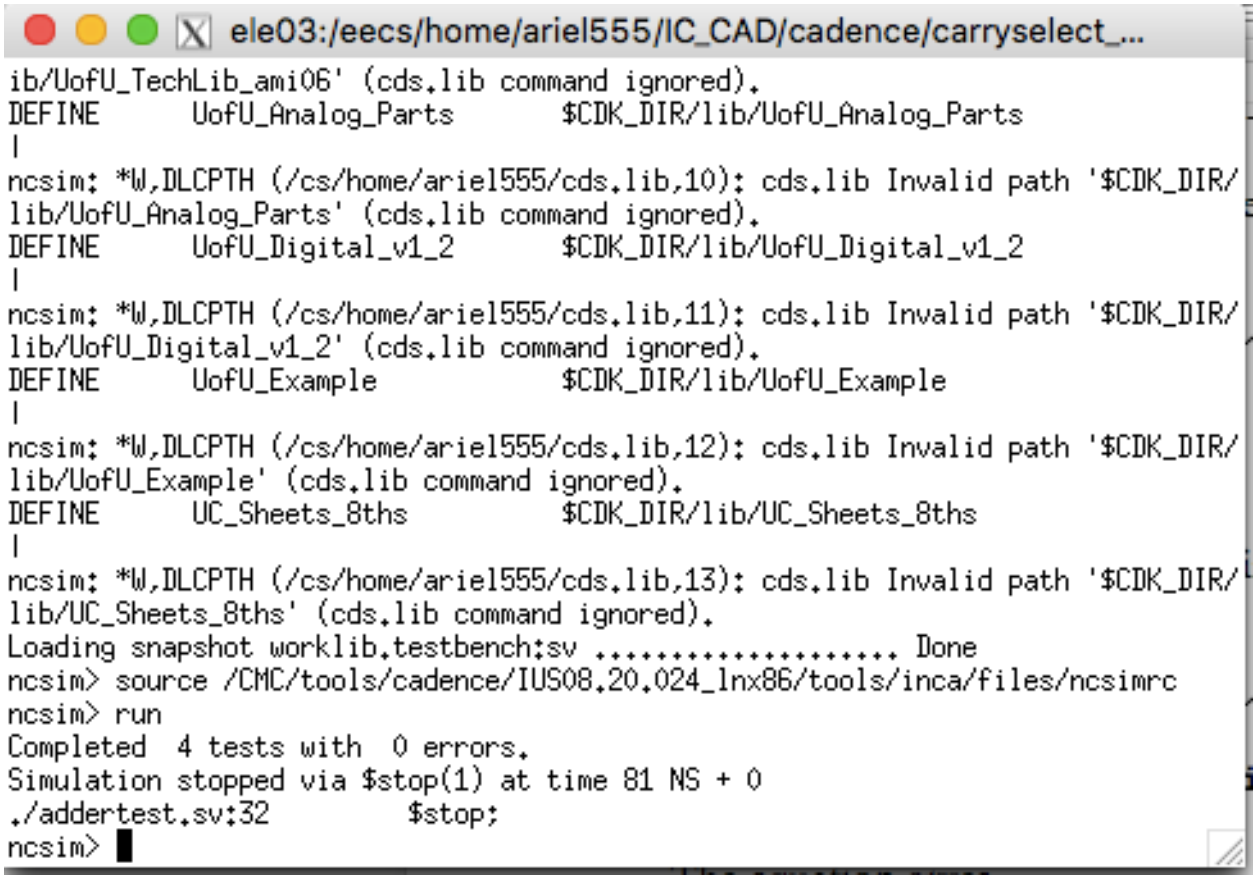

Figure 4: Terminal window showing that the simulation of the test vectors in `adder.tv` for the schematic was successful

I was unable to put my adder.tv file in because addertest.sv kept giving errors. I think it may have something to do with the size difference (since I have more test vectors).

Part 4. DRC and LVS of the `chip` layout

The equation gives

$$EDP = \frac{Average\ power \times Cycle\ time^2}{\alpha} = \frac{5.975\ \times (6)^2}{0.1579} = 1362\ pJ \cdot ns$$

From the simulation, the EDP of my carry select adder was around 1300 pJ-ns.
The cycle time was 6 ns, and the frequency was 167 MHz.

```
The cycle time is now 6.000000 ns

>info:         ***** hspice job concluded
real 2.00
user 1.06
sys 0.00

ans =

     0

-5.9751E-03
SUCCESS, PASSED THE TEST!

ans =

     0

rm: remove regular file `testbench.lis'? y
```

Figure 5: MATLAB window showing that the simulation was successful at 6 ns

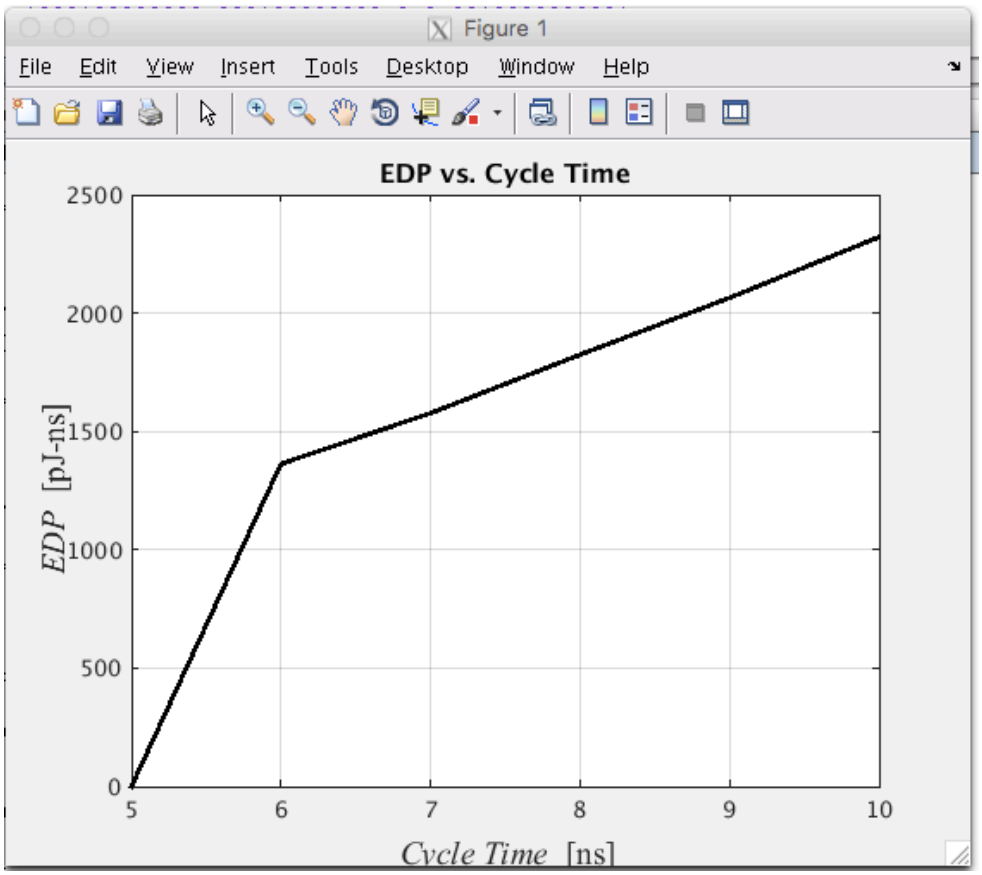

Figure 6: MATLAB window showing EDP vs cycle time

In order to improve on the EDP, I reduced the length of the ripple carry adder (RCA) and made the adders run in parallel (as in the carry select adder design). Although this increases the area, it greatly reduces the delay of the critical path, which is the carry. The carry select adder predicts the sum and the MUX chooses the correct solution. The schematic for the carry select adder was created in reference to the lecture slides.

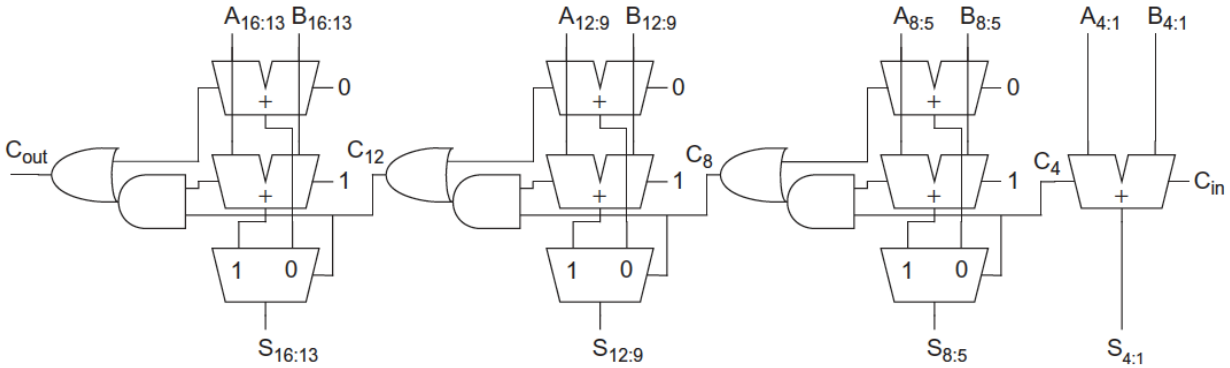

Figure 7: 16-bit carry select adder schematic from the lecture slides