University of Victoria

CENG 241

Digital Design I

Lab 4 - 4-bit Binary Adder / Subtractor

Instructor:
Dr. Amirali Baniasadi

Teaching Assistant:
Grace Hui

Yves Senechal V00213837
Tyler Stephen V00812021
A01 - B03

June 22, 2015

1 Introduction

Combinational logic circuits can be implemented to add and subtract binary numbers. In this lab, a 4-bit binary adder/subtractor was designed, implemented and tested through the Xilinx ISE and Verilog HDL.

2 Discussion

2.1 Binary addition and subtraction using complement representations

Addition of two 1-bit binary numbers (i.e. A + B) is achieved simply by adding A + B where a sum (S) and carry (C) are the outputs. Two half adders (see Lab 2) can be combined into a full adder to execute binary addition. Instead of creating new circuits for subtraction, the dual problem will be solved. Instead of subtracting B from A we will add the additive inverse of B to A.

$$A - B = A + (-B)$$

For a signed binary number system, negative numbers can be interpreted as ones' and twos' complement. The ones' complement of a number is its bitwise inversion. The MSB is the sign bit where 0 indicates positive and 1 indicates negative. The twos' complement is identical to the ones' complement except 1 is added to the LSB of negative numbers. The twos' complement system is used because its domain is larger by one element.

Number	Ones complement	Twos complement
5	0101	0101
-5	1010	1011

Implementing the complement representation for negative numbers with logic gates is fairly simple. XOR gates have characteristic functions $X \oplus 0 = X$, and $X \oplus 1 = X'$. The ones' complement is generated by putting XOR gates in front of the full adder input for B. The twos' complement is created by adding a 1 to the carry input of the first adder. All of these operations are controlled by a single input called CONTROL.

For addition, CONTROL is set to 0, so the XOR gate provides no change to B; B is added to A. For subtraction, CONTROL is set to 1 which inverts B through the XOR gate, CONTROL is then inputed as CIN which effectively adds 1 to the inverted B to become its twos' complement.

2.2 Building the hierarchical adder / subtractor

To construct the complex 4-bit adder/subtractor, a hierarchical design was implemented to build the circuit in stages which greatly simplified it. The adder/subtractor shown in Figure 5, required four full-adders (FA) and four XOR gates. The FAs were constructed from two half-adders (HA) and an OR gate shown in Figure 3. Each HAs were constructed using only logic gates illustrated in Figure 1.

2.2.1 Half adder

First, the HA was designed on the Xilinx ISE (Figure 1). To ensure a properly functioning circuit, a simulation was performed to ensure proper outputs from any inputs; the results

appear in Figure 2.

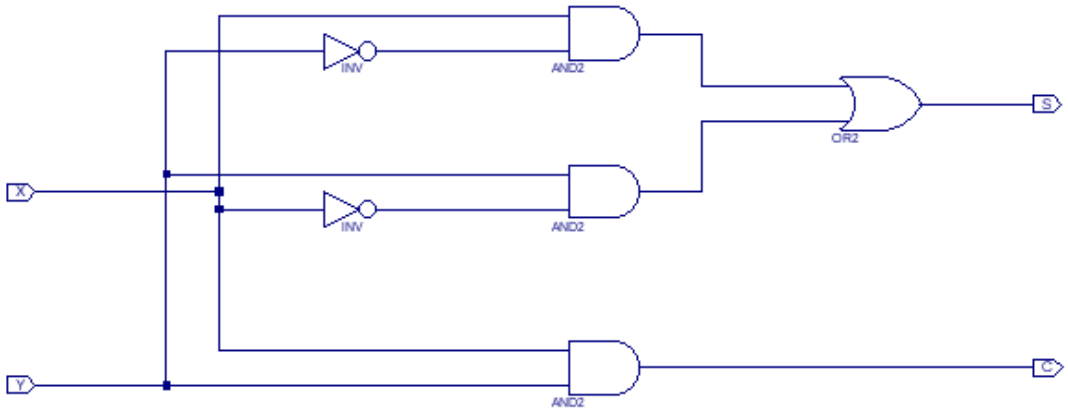


Figure 1: Schematic of the half adder

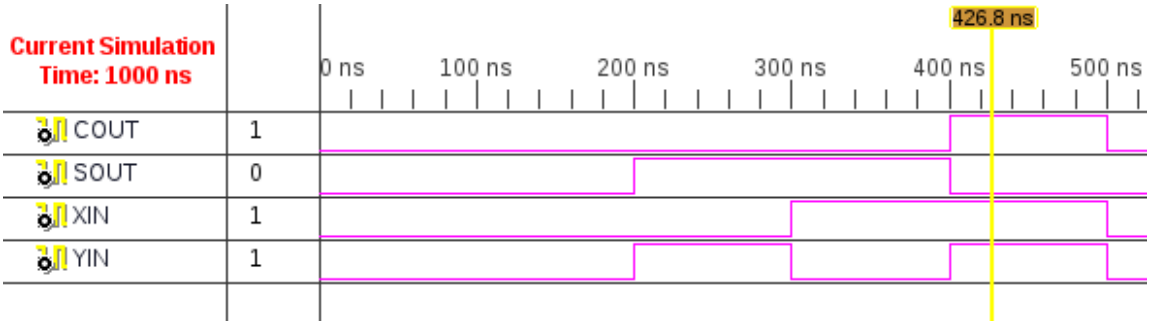


Figure 2: Test results of the half adder

2.2.2 Full adder

Second, the FA, shown in Figure 3, was designed on the schematic editor. The circuit was similarly tested as the HA to ensure proper functionality; its results appear in Figure 4.

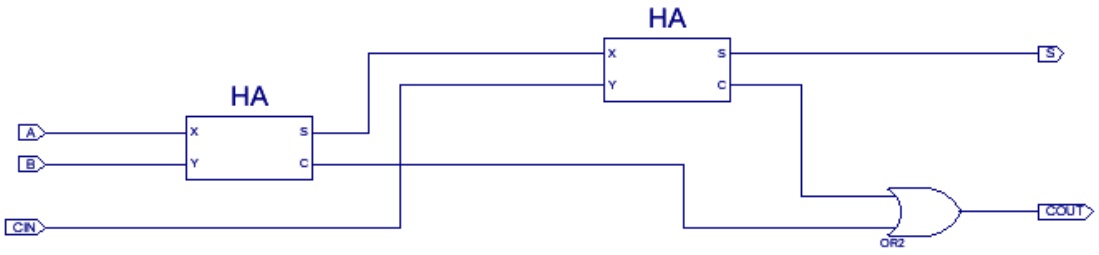


Figure 3: Schematic of the full adder

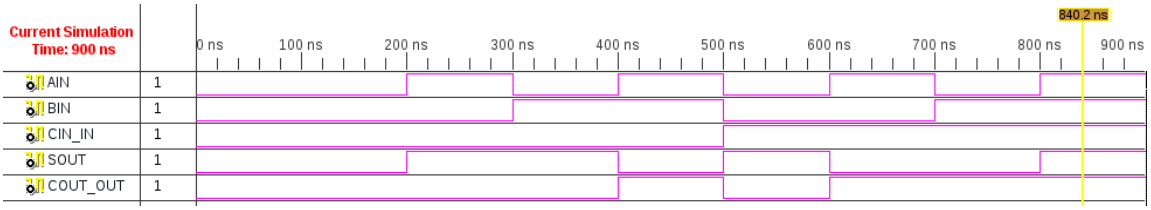


Figure 4: Test results of the full adder

2.2.3 Adder / subtractor

Next, the adder/subtractor was laid out as indicated in Figure 5. Notice, there are two 4-bit binary numbers plus a single bit CONTROL that comprise the inputs, while the outputs are a 4-bit sum with a single bit carry. As indicated earlier, the CONTROL bit is responsible for the equation's operation: either 0 for addition, or 1 for subtraction.

The adder/subtractor was tested with various inputs that can be observed in Figure 6. The results are summarized in Table 1.

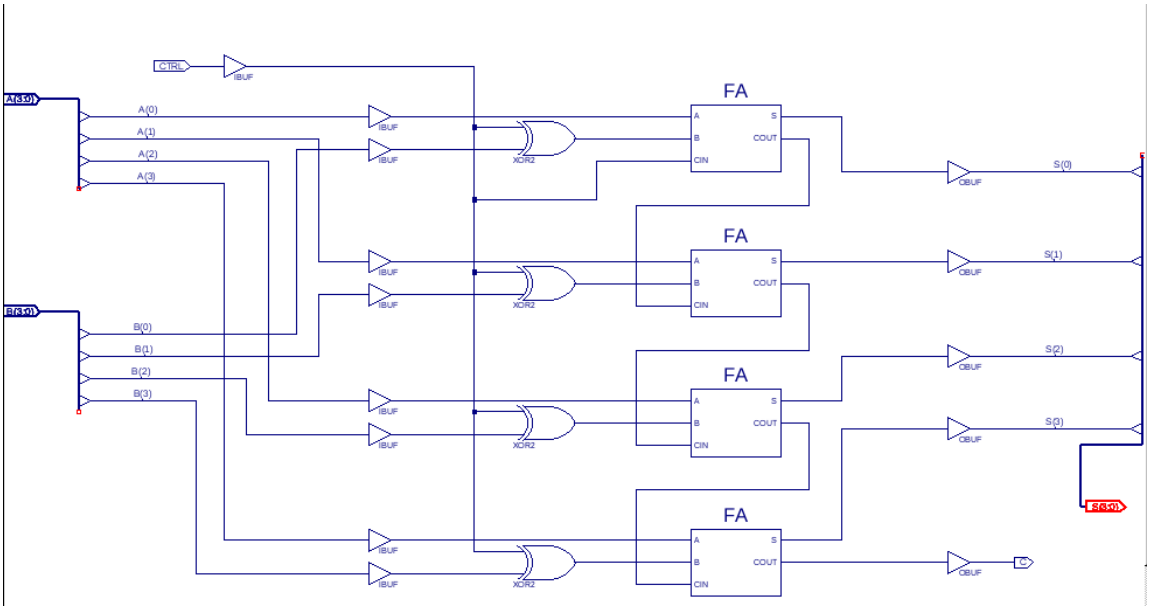
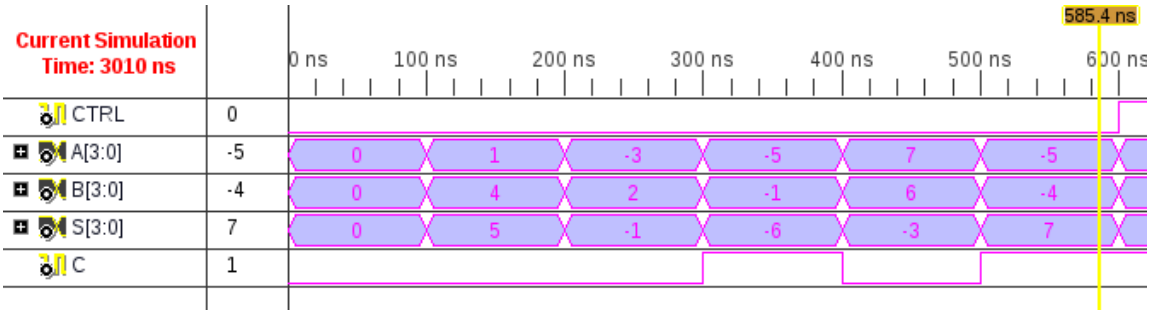
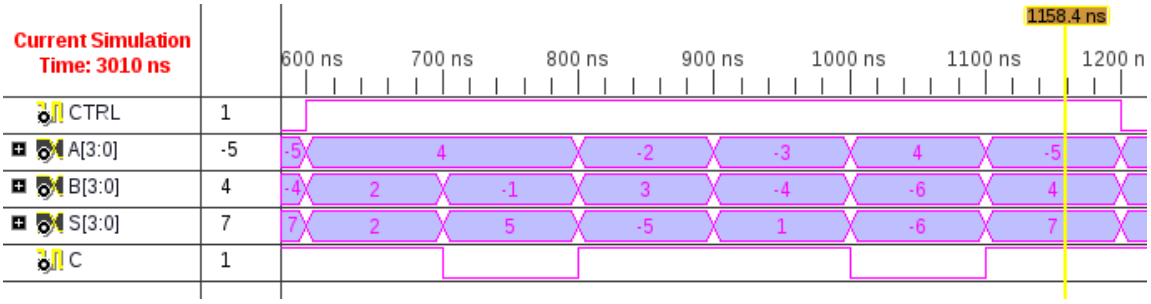


Figure 5: Schematic of the adder / subtractor



(a) Addition



(b) Subtraction

Figure 6: Test results of the adder / subtractor

CTRL	A	B	S	C
0	1	4	5	0
0	−3	2	−1	0
0	−5	−1	−6	1
0	7	6	−3	0
0	−5	−5	7	1
1	4	2	2	1
1	4	−1	5	0
1	−2	3	−5	1
1	−3	−4	1	1
1	4	−6	−6	0
1	−5	4	7	1

Table 1: Output of the adder / subtractor for several test cases, in decimal

2.3 Verilog hierarchical adder / subtractor

Similar to the schematic method, a hierarchical design for the adder/subtractor was implemented with the Verilog HDL. The FA, displayed in FA.v, included the HA design from Lab 2. The code forming the adder/subtractor is laid out in AddSub.v.

2.3.1 Listings

```
module FA(A, B, CIN, S, COUT)
  input A, B, CIN;
  output S, COUT;

  HA U0(A, B, S0, C0);
  HA U1(S0, CIN, S, C1);

  assign COUT = C0 | C1;
```

FA.v

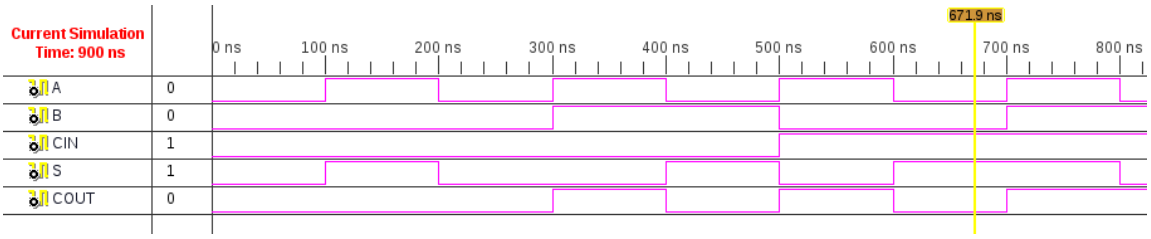
```
module AddSub(A0, A1, A2, A3, B0, B1, B2, B3, CTRL, S0, S1, S2, S3, C)
  input A0, A1, A2, A3, B0, B1, B2, B3, CTRL;
  output S0, S1, S2, S3, C;

  FA U0(A0, B0 ^ CTRL, CTRL, S0, COUT0);
  FA U1(A1, B1 ^ CTRL, COUT0, S1, COUT1);
  FA U2(A2, B2 ^ CTRL, COUT1, S2, COUT2);
  FA U3(A3, B3 ^ CTRL, COUT2, S3, C);
```

AddSub.v

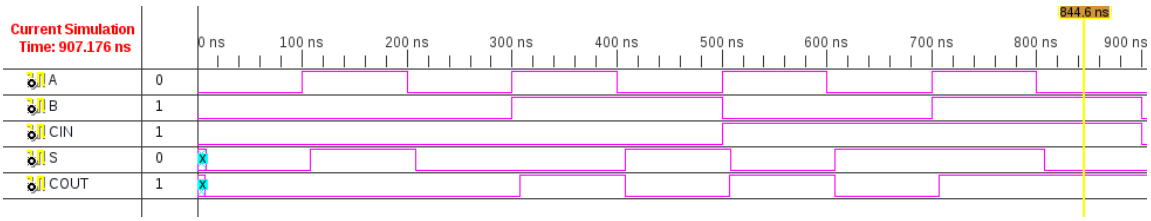
2.3.2 Simulation results and delay times

To ensure the functionality of the FA, a functional simulation was performed which can be viewed in Figure 7(a). Post place simulation indicated a total delay of 8.0 ns which can be viewed in Figures 7(b)&(c). The delay for COUT is 0.7 ns less than S because it resolves faster (see Figures 1, 3, Lab 2).

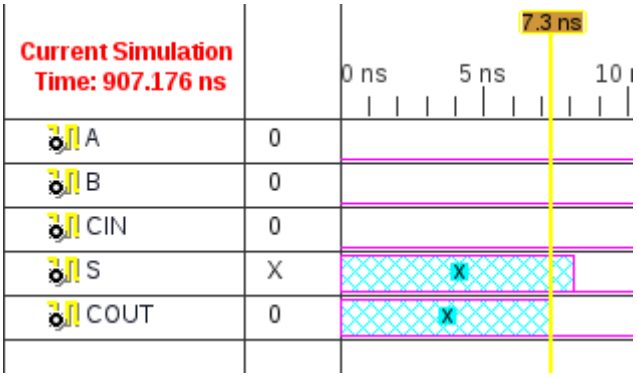
The adder/subtractor was then tested for functionality, and the summations and deductions can be observed in Figure 8. The test cases were a subset of the results from Table 1 and the simulation returns the expected result. Many attempts of observing a propagation delay of the adder/subtractor verilog through post place simulation were ultimately unsuccessful. It is likely that the total delay would be approximately 32 ns since each full adder must resolve before the next bitwise addition can occur.



(a) Functional simulation



(b) Post place timing



(c) Delay

Figure 7: Simulation results for the full adder written in verilog

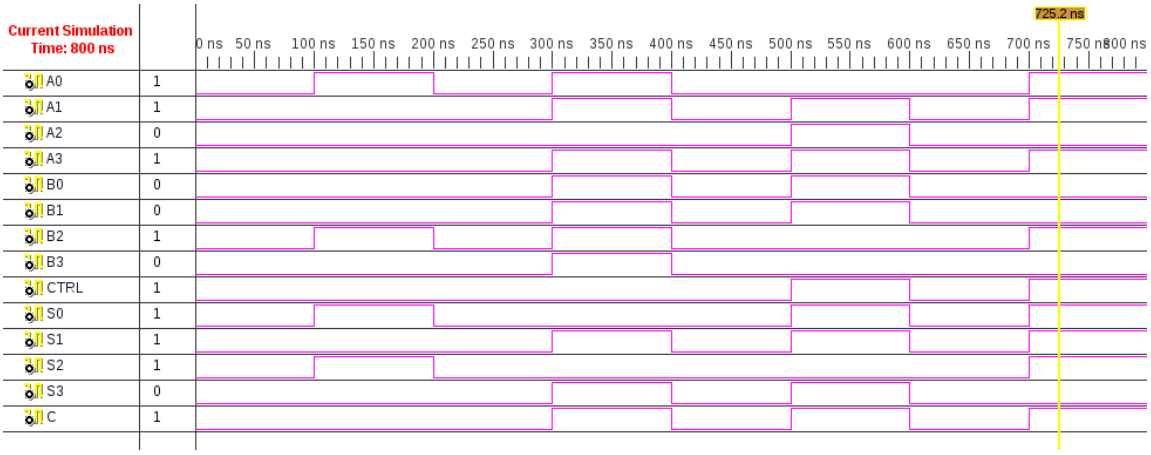


Figure 8: Functional simulation results for adder / subtractor written in verilog

3 Conclusion

This lab proved the possibility of creating a hierarchic 4-bit adder/subtractor from a single control bit, XOR gates, and FAs on two separate platforms. The schematic method worked flawlessly, while the verilog method, possibly due to a glitch, would not propagate the delay.