Lab 2: Adding and Subtracting Binary Numbers

Introduction

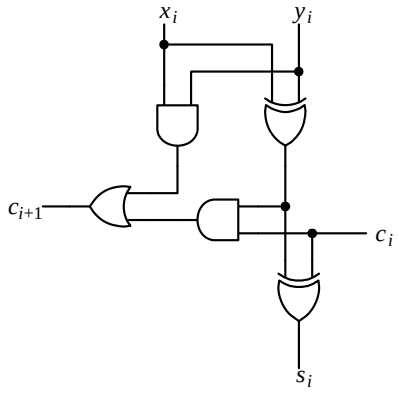
In this lab you will learn the following:

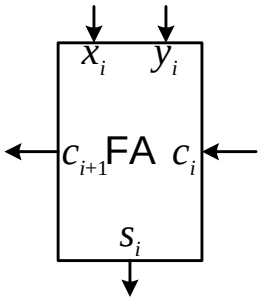
- Experiment with several digital logic circuits for adding and subtracting binary numbers. You will implement these circuits on the DE1 development board to test and verify their operations.

1. Full Adder

The full adder (FA) circuit is a circuit that adds two 1-bit numbers and produces a 2-bit result. The truth table, reduced circuit and symbol are shown next. The circuit uses two 2-input AND gates, two 2-input XOR gates, and a 2-input OR gate.

x_i	y_i	c_i	c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1





Create your personal folder for this class on the E drive if you have not already done so.

Create a new Quartus project named Lab2 and save it in your personal folder on the E drive.

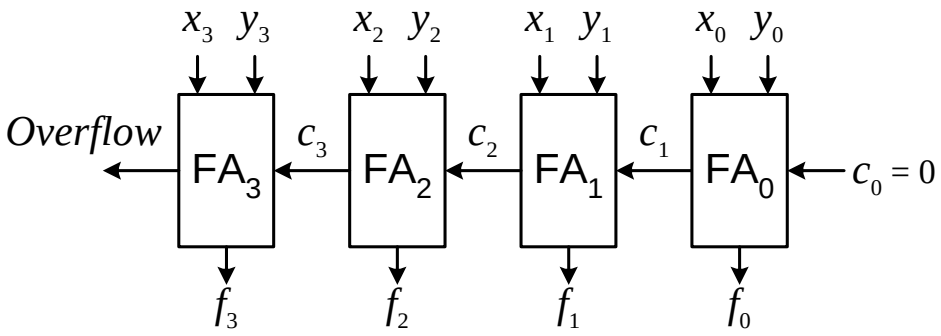
Draw the above full adder circuit and save it using the name **FA**. Create a symbol (also called **FA**) for it. Test it on the DE1 board and make sure that it works correctly. To test this file, you need to temporary set this file to be the top level file for your project. Right-click on this file, **FA.bdf**, and select **Set as Top-Level Entity**. Map the I/O signals as follows:

x_i	SW[2]
y_i	SW[1]
c_i	SW[0]
c_{i+1}	LEDR[1]
s_i	LEDR[0]

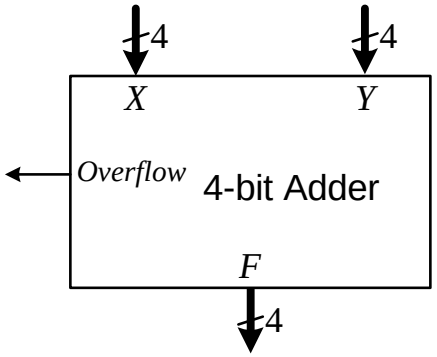
Verify that your circuit works according to the given truth table.

2. 4-Bit Ripple-Carry Adder

The 4-bit ripple-carry adder is a simple adder to add two 4-bit binary numbers, and produces a 4-bit result and an overflow bit, i.e., $x_{3\text{-}1} + y_{3\text{-}1} = f_{3\text{-}1}$. The circuit for the 4-bit ripple-carry adder shown next uses four instances of the FA circuit that you created in step 1.



Open a new schematic drawing file. Insert four instances of the FA circuit from your project library. Draw this ripple-carry adder circuit and save it using the name **Adder**. Create a symbol for it using the same name.



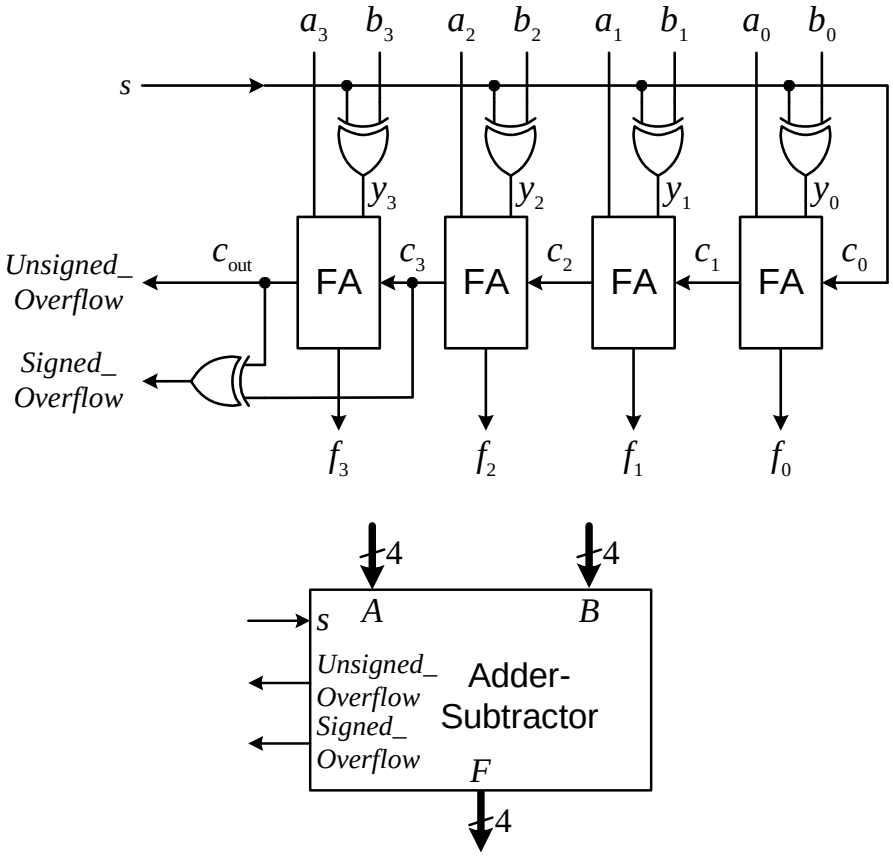
Test it on the DE1 board and make sure that it works correctly. To test this file, you need to temporary set this file to be the top level file for your project. Right-click on this file, **Adder.bdf**, and select **Set as Top-Level Entity**. Map the I/O signals as follows:

y_3-y_0	SW[3]-SW[0]
x_3-x_0	SW[7]-SW[4]
f_3-f_0	LEDR[3]-LEDR[0]
Overflow	LEDR[4]

Verify that your circuit does add two 4-bit binary numbers correctly.

3. 4-Bit Adder-Subtractor Circuit

s	Function	Operation
0	Add	$F = A + B$
1	Subtract	$F = A - B = A + B' + 1$



Draw this adder-subtractor circuit and save it using the name **AddSub**. Create a symbol for it. Map the I/O signals as follows:

b_3-b_0	SW[3]-SW[0]
a_3-a_0	SW[7]-SW[4]
s	SW[8]
f_3-f_0	LEDR[3]-LEDR[0]
Unsigned overflow	LEDR[4]
Signed overflow	LEDR[5]

Test it on the DE1 board and make sure that it works correctly.