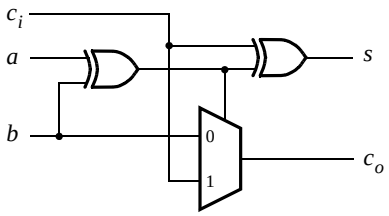
Laboratory Exercise 2

Numbers and Displays

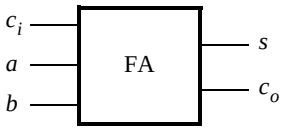
This is an exercise in designing combinational circuits that can perform binary-to-decimal number conversion and binary-coded-decimal (BCD) addition.

Part I

Figure 1a shows a circuit for a *full adder*, which has the inputs a, b, and c_i, and produces the outputs s and c_o. Parts b and c of the figure show a circuit symbol and truth table for the full adder, which produces the two-bit binary sum $c_o s = a + b + c_i$. Figure 1d shows how four instances of this full adder module can be used to design a circuit that adds two four-bit numbers. This type of circuit is usually called a *ripple-carry* adder, because of the way that the carry signals are passed from one full adder to the next. Write Verilog code that implements this circuit, as described below.



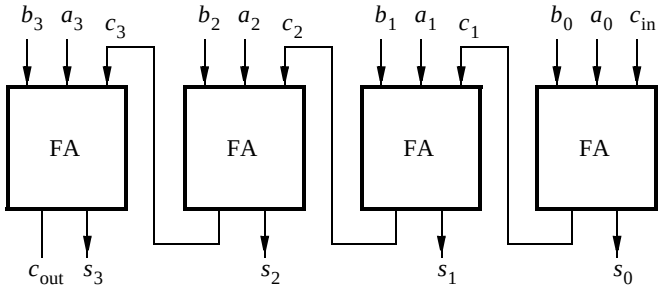
a) Full adder circuit



b) Full adder symbol

b	a	c_i	c_o	s
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

c) Full adder truth table



d) Four-bit ripple-carry adder circuit

Figure 1: A ripple-carry adder circuit.

1. Create a new project for the adder circuit. Write a Verilog module for the full adder subcircuit and write a top-level Verilog module that instantiates four instances of this full adder.
2. Use switches SW_{7-4} and SW_{3-0} to represent the inputs A and B, respectively. Use SW_8 for the carry-in c_{in} of the adder. Connect the outputs of the adder, c_{out} and S, to the red lights LEDR.
3. Include the necessary pin assignments for your FPGA board, compile the circuit, and download it into the FPGA chip.
4. Test your circuit by trying different values for numbers A, B, and c_{in}.

Part II

For this part you are to design a circuit that has two decimal digits, X and Y, as inputs. Each decimal digit is represented as a 4-bit number. In technical literature this is referred to as the *binary coded decimal* (BCD) representation.
You are to design a circuit that adds the two BCD digits. The inputs to your circuit are the numbers X and Y, plus a carry-in, c_{in}. When these inputs are added, the result will be a 5-bit binary number. But this result is to be displayed on 7-segment displays as a two-digit BCD sum S_1S_0. SW_9 decides on the digit shown. If it is logic-0, the 7-segment shows the first digit. It shows the second digit if it is logic-1. For a sum equal to zero you would display $S_1S_0 = 00$, for a sum of one $S_1S_0 = 01$, for nine $S_1S_0 = 09$, for ten $S_1S_0 = 10$, and so on. Note that the inputs X and Y are assumed to be decimal digits, which means that the largest sum that needs to be handled by this circuit is $S_1S_0 = 9 + 9 + 1 = 19$.
Perform the steps given below.

1. Create a new project for your BCD adder. You should use the four-bit adder circuit from part I to produce a four-bit sum and carry-out for the operation X + Y.

2. Use switches SW_{7-4} and SW_{3-0} for the inputs X and Y, respectively, and use SW_8 for the carry-in. Connect the four-bit sum and carry-out produced by the operation X + Y to the red lights LEDR. Display the BCD values of X and Y on the 7-segment display depending on the SW_9

3. Include the necessary pin assignments for your FPGA board, compile the circuit, and download it into the FPGA chip.

4. Test your circuit by trying different values for numbers X, Y, and c_{in}.