Exp 0: The Half Adder (HA)

Objectives:

- To be exposed to the Xilinx Design Methodology
- To design, implement, and test a digital circuit

Background: Designing circuits that perform arithmetic operations is one of the more common pursuits in digital design-land. The half adder, or HA, is one of the most basic digital circuits. The HA adds two bits and outputs the result of the addition (the sum) and a bit that indicates whether the operation generated a carry out (CO).

Assignment: Design a half adder and implement your design in iSim and on the development board. Assign the inputs and outputs to the devices on the board as listed in Table 0.1.

Input		**Output**	
OP_A	OP_B	SUM	CO
left-most switch	right-most switch	left-most LED	right-most LED

Table 0.1: Input/Output specification for the HA.

Questions:

1. In your own words, what is meant by the term "methodology"?

2. What is a "PLD"? Name at least two different types of PLDs.

3. Xilinx is not the only company in the world that provides software for implementing circuits on PLDs. In what ways, if any, could you imagine the PLD circuit implementation methodology differing with another company's Computer Aided Design (CAD) software?

4. Briefly describe the level of awe you feel due to your success in this lab activity: you designed a digital circuit that actually did something (performed a math operation).

5. How many signal assignment statements did you use in your verilog model for the HA?

Exp 1: Full Adder (FA) - SOP & POS Forms

Objectives:

- To receive more exposure to the Xilinx Design Methodology
- To design, implement, and test a standard digital circuit
- To prove that SOP and POS forms are functionally equivalent
- To show that multiple functions can be placed in the same verilog model

Background: As you know by now, the half adder lacks the ability to handle a carry-in input. The full adder, or FA, is essentially a half adder with a carry-in input. The outputs of FA are the same as the outputs of the HA.

Assignment: Design a full adder and implement it simultaneously (in the same verilog model) using both SOP and POS forms. Test your design in iSim then assign the inputs and outputs to the development board as listed in Table 0.1.

Input			**Output**			
OP_A	OP_B	Cin	SUM_SOP	SUM_POS	CO_SOP	CO_POS
left-most switch	right-most switch	Some middle switch	two left-most LEDs		two right-most LEDs	

Table 0.1: Input/Output specification for the FA.

Questions:

1. In your own words, define what is meant by the term "functionally equivalent".

2. If the sets of LEDs in this activity did not light-up in pairs, what would that indicate to you? Briefly explain.

3. Which form was easier for you to implement: the SOP or POS form? Briefly explain.

4. Could you make a general statement that either the SOP or POS forms of a function would be easier to implement? Briefly explain.