Lab 2: 4-bit binary adder subtractor

Goals

- The objective of this lab is to implement a 4-bit binary adder subtractor using continuous assignment and module instantiation.
- The goal is that the student is familiarized with module instances and propagation of signals through hierarchy and practice defining logic with continuous assignment.

Introduction

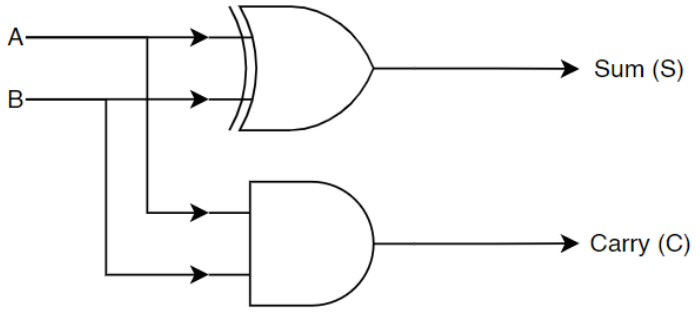
A binary adder is a circuit that adds two binary numbers, it is one of the most used circuits for various more complex applications and circuits. The simplest circuit is the half-adder circuit that adds two bits and keeps a carry bit for overflow when both inputs are high. A full adder is a circuit that adds three binary numbers where one can be the carry of a previous stage, this is done via two half-adder circuits. A binary adder-subtractor can do both the addition and subtraction of binary numbers in one circuit itself. The operation is performed depending on the binary value the control signal holds.

Lab description

Design and implement a 4-bit binary adder subtractor via instances of other modules like half-adders and full-adders

Step 1.

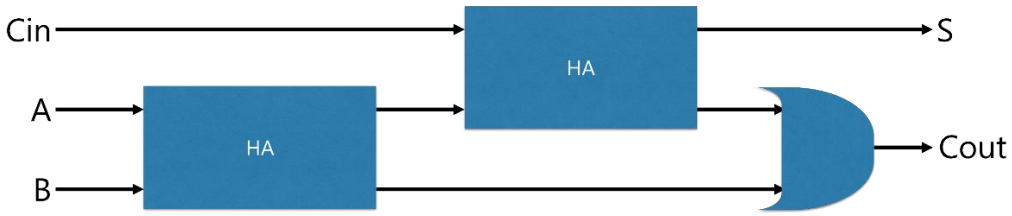
Create a module to define the half-adder circuit that satisfies this circuit:



Define the module inputs and outputs and then use primitive gates or if you're familiar with operands use continuous assignment with gate operators. You can also use procedural assignment but do not use the "+" operator.

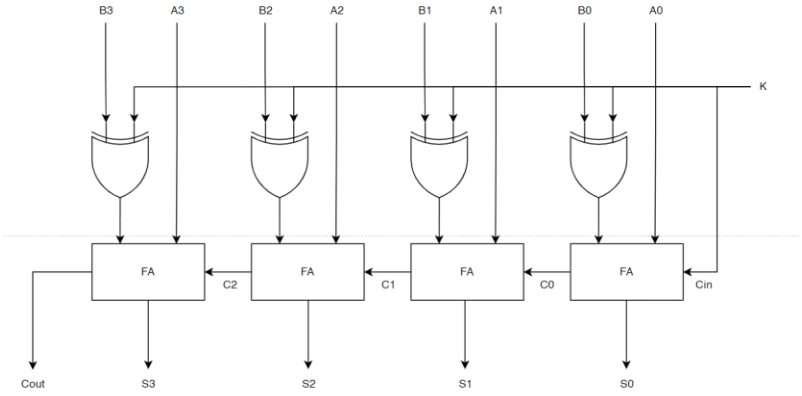
Step 2.

Create a module for the full adder that instantiates the half-adder to satisfy this circuit:



Step 3.

Then use the full adder instance to create the following design at a higher level, the module can have any legal name you wish it to have.



Step 4.

- Create a testbench to verify that the addition and subtraction work, you will verify via the waveform that the sum and subtraction are correct. An example of the testbench could be a module called tb that instantiates your circuit and then sends random stimuli to it.

Step 5.

Implement the circuit on the FPGA.

- Use the following **switches** for input B: **SW[3] to SW[0]**
- Use the following **switches** for input A: **SW[17] to SW[14]**
- Use the following **button** for addition/substraction selection K: **BTN[0]**
- Use the following red **leds** for input B visualization: **LEDR[3] to LEDR[0]**
- Use the following red **leds** for input A visualization: **LEDR[17] to LEDR[14]**
- Use the following green **leds** for outputs result S signal: **LEDG[3] to LEDG[0]**

Finally, instantiate lab 1 as a binary input that can display both A and B operands and the result S into pairs of 7 segment displays for each one respectively for human interface better visualization of binary numbers.

Submission

Implementation description

Generate a document providing a detailed description of the implementation. This should include an explanation of design decisions, challenges encountered, and how they were overcome. Include diagrams, schematics, tables, or equations used in the design development.

Video:

Upload a short video to the Teams group, recording the operation of the implemented circuit on the FPGA and providing a brief description of its functionality.