Lab 8: ad-hoc Design

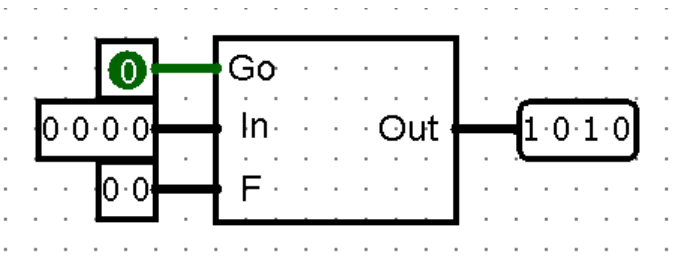
NOTE: This lab may be too simple!

PURPOSE

The purpose of this laboratory project is to introduce design of complex systems using more complex (datapath-level) components. In this lab, we will design a 4-bit calculator using existing modules. This lab also develops familiarity with twos-complement representation.

A 4-bit Calculator

Consider a device with the following inputs: a 4-bit input IN, a 2-bit input F, a 1-bit input GO, and a 4-bit output OUT. These are the inputs to a calculator that has a 4-bit memory (called the accumulator A). The device can perform the following functions, based upon the value of the input F.



F0F1	Function	
00	Set Data	A = IN
01	Add	A = A + IN
10	Subtract	A = A – IN
11	Multiply	A = A * IN

Figure 1: The 4-bit Calculator and function encoding table

The calculator receives is a clocked synchronous device that receives a continuous clock signal. The calculator constantly displays the contents of the accumulator on the output OUT. It ignores the inputs IN and F except when told to process a single operation by an assertion of GO. The indicated function should be performed one time, even if GO remains high for several clock ticks. No other function should be performed until GO is clocked at zero (indicating the end of one operation) and then asserted again on a different clock tick (indicating the beginning of the next operation).

F	GO	IN	OUT
00	0	0000	1010
00	1	0000	1010
00	1	0000	0000
00	0	0000	0000
00	0	0010	0000
00	1	0010	0000
00	1	0010	0010
00	0	0010	0010
01	0	0010	0010
01	1	0010	0010
01	1	0010	0100
01	0	0010	0100

Figure 2: Sample logging of input/outputs of the 4-bit calculator

4-bit Calculator Design and Implementation

Design and implement the 4-bit calculator. You may use ANY technique learned in the class (that is supported by the simulator). You may use high-level modules available in the simulator library, including multiplexors, adders, subtracters, multipliers, and registers. You will also need to use low-level devices possibly including gates, flip-flops, splitters, power, ground, and the like.

In your labbook, indicate what your design strategy is and WHY you've chosen that strategy. Use appropriate terminology. If you change design approaches, clearly indicate the problems encountered with the current approach and how you expect your next approach might differ.

Grading rubric

- [1 points] If your calculator can load any given 4-bit value into the accumulator and display this value.
- [2 points] If your calculator can **add** the INPUT to the current value stored in the accumulator and display this new value. Your test results in your labbook must demonstrate behavior on inputs under both unsigned magnitude and two-complement encodings.
- [1 points] if your calculator can **subtract** the INPUT to the current value stored in the accumulator and display this new value. Your test results in your labbook must demonstrate behavior on inputs under both unsigned magnitude and two-complement encodings.
- [1 points] If your calculator can **multiply** the INPUT to the current value stored in the accumulator and display this new value. Your test results in your labbook must demonstrate behavior on inputs under both unsigned magnitude and two-complement encodings.
- [1 points] if you calculator performs exactly ONE operation per "Cycle" of Go. That is, your device performs an operation exactly once when GO is asserted and performs NO other operation until after Go is returned to zero (for at least once clock event).

LAB 8: DEMONSTRATION

[2 points] Demonstrate your 4-bit calculator to your lab instructor on a set of commands provided by the lab instructor. Be prepared to answer questions about your design and implementation. Be prepared to test your devices using BOTH unsigned magnitude and/or twos-complement binary representations. Your lab instructor may select to use either representation.

Integrated Writing [2 points]: Refer to "Digital System Design: Engineering Journals & Lab Policies" for details. 0.5 points each for Completeness, Clarity, Organization, and Testing.