---

**Lab 2: Adding and Subtracting**
**Worth 65 points (60 lab + 5 report)**

---

**Lab Objectives:**

You now have had a chance to play around with some basic combinational logic circuits. Let's move on to something a little more interesting, sequential logic. From class you learned that combinational logic was just a function of current inputs and that sequential logic was a function not only of current input, but some past sequence of inputs. We are going to use sequential and combinational logic that allows us to sum or subtract a sequence of numbers together.

**Lab Overview:**

To build our summer we are going to need several parts.  Refer to the full diagram below to see how it all fits together.

- Please name your file Lab2.lgi.
- Make an overview page containing the input and output for your circuit along with the control buttons.
- Separate out the other parts onto separate pages.  Test these parts separately to make sure they work before integrating them with the rest of the circuit.
- Be sure to look up "signal sender/receiver" to route signal between pages.
- There is a file called "Starter_Parts.lgi" which includes various useful pieces for your use.
- You are encouraged to add LEDs or other devices to your schematic to make it easier to debug.
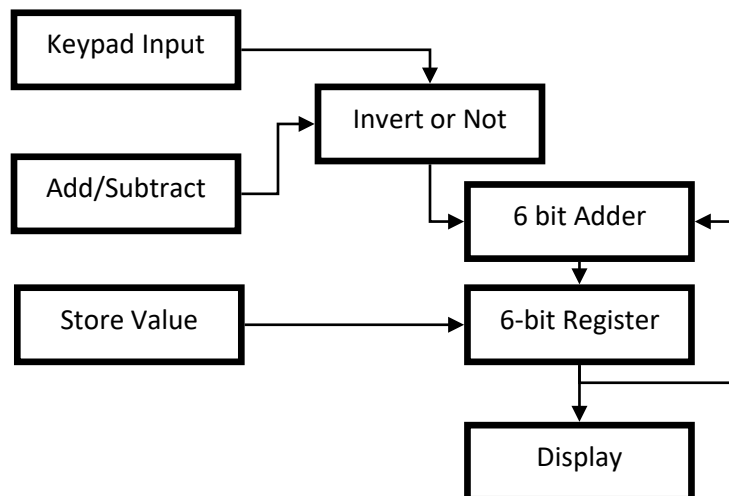
**Part 1: Register**

In this part we need to build our register file. This will store our running total and will display it to the user as a hex number.

- The register will be 6 bits wide. This means that you will need 6 D Flip-Flops for storage. The part in mml is a little bit complicated to setup and is included in the included .lgi file.

- Connect all the CLR lines to one switch. When this switch is set to 0 it will set all of our Flip-Flops to zero. When starting the summer you will need to flip this switch to 1 to start adding numbers.

- Connect all the clocks to one momentary switch. This will be our clock. When the switch is pressed the data will be stored in our flip flops. Unless the button is pressed the value in the flip-flips should not change.

- Hook up the outputs of the Flip-Flops to 7 segment LEDs.

**Part 2: Input**

While our memory is 6 bits we are only going to be inputting 4-bit numbers to simplify things. In the starter file there is a keypad already hooked up with an output. Familiarize yourself with the keypad and make sure you can use it. Once you have done so copy it into your main lab file.

- The flip-flops are an important part of the keypad. If they are not there the keypad only outputs a value while the key is pressed.

**Part 3: Addition**

Build a 6-bit full adder. I would highly suggest building a one bit full adder and copying it six times. Give yourself some debug tools and connect the inputs and outputs up to LEDs or 7 segments so that you can see what is going on.

**Part 4: Subtraction (or Inverting and Adding One)**

Instead of building a dedicated hardware to for subtraction we will use the additive inverse. To do so we need to invert our input and use the carry in bit on our adder. Build the circuit that allows you to toggle between the normal and inverted 4-bit number with a switch. Hook up this same switch to the carry in for your full adder's least significant bit.

- This is a good place to use a multiplexer but considering it is a 2-to-1 mux it is not much harder to build it using discrete gates.

**Lab Write-up Requirements**

Discuss issues you had building the circuit.  Describe what you added to the minimum specifications to make debugging easier.

What happens when you subtract a larger number from a smaller number?  Does the result make sense?  Your 7 segment display might not.  What happens when you add two numbers that won't fit in 6 bits?  You will need to add several times to do this as you only have 4-bit input.

To alleviate file format issues we want lab reports in plain text.  Feel free to use a word processor if you like to type it up but please submit a plain text file.

**Collaboration: *You are allowed to discuss this lab with other students on this lab but all the work must be your own.***

**Files to Submit on Canvas**
- Lab2.LGI
- Lab2_report.txt

**Check-off**
You should demonstrate your lab to the TA/Tutor when finished to get it checked off.

**Point Breakdown**

- 5 pts: Input
- 10 pts: Inverter
- 20 pts: Adder
- 20 pts: Register
- 5 pts: Output
- 5 pts: Writeup