ES 4 Lab 3: Binary Addition Using Multiplexers

Prelab due 24 hours before your lab session, week of September 27

Lab report due at your lab time, week of October 4

1 Introduction

In this lab, you'll build another useful logic circuit: an "adder" which adds two 2-bit binary numbers and produces a 3-bit result[1]. Unlike the previous lab where we used an assortment of logic gates (AND, OR, XOR, etc.), you'll build this circuit entirely with multiplexers. This can simplify the process of building the circuit, but it also provides a conceptual bridge to the way we'll be using our FPGAs in the future. To oversimplify, an FPGA is a chip which contains thousands of configurable multiplexers linked together, which can be programmed to compute nearly any logic function.

After successfully completing this lab, you should be able to:

- Use multiplexers to implement combinational logic.
- Use the Digital Discovery's pattern generator and logic analyzer to automate testing of a digital circuit.
- Measure timing characteristics of combinational logic.

2 Prelab

In this prelab, you will design a circuit to compute the sum of two binary numbers, $R = M + N$. You can use M_1 and M_0 to represent the values of the individual bits. M_1 is the most significant bit (i.e., $M = 2M_1 + M_0$).

$$\begin{array}{ccc} & M_1 & M_0 \\ + & N_1 & N_0 \\ \hline R_2 & R_1 & R_0 \end{array}$$

P1: Write the truth table for the least significant bit of the addition (i.e., write R_0 as a function of M_0 and N_0).

P2: Write the truth table for the carry out of the least significant bit (i.e., find the carry from the one's place to the two's place as a function of M_0 and N_0).

[1] A 16-bit adder would be more useful, but building it by hand would be awfully tedious.

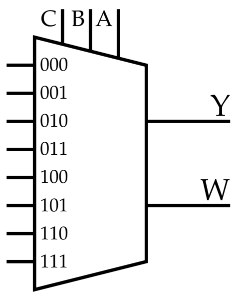
P3: Write the truth table for the second bit of the addition (i.e., write R_1 as a function of M_1, N_1, and C_{out0}, the carry from the least significant bit).

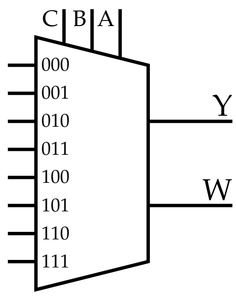
P4: Write the truth table for the carry out of the second bit of the addition (i.e., the carry from the two's place to the four's place as a function of M_1, N_1, and the carry in).

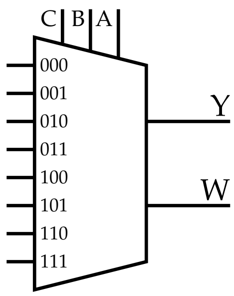
Refer to the datasheet for the 74HC151 multiplexer IC (posted on the course website) for the following questions. Note that the datasheet describes several different parts, so you'll need to read it carefully to get the right information.

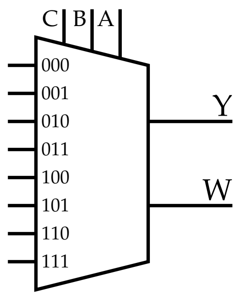
P5: What is the difference between the Y and W outputs?

P6: Draw a logic diagram showing how you will wire up multiplexers to implement each of the truth tables in P1-P3. *You'll need to figure out what to do with the unused multiplexer inputs.*







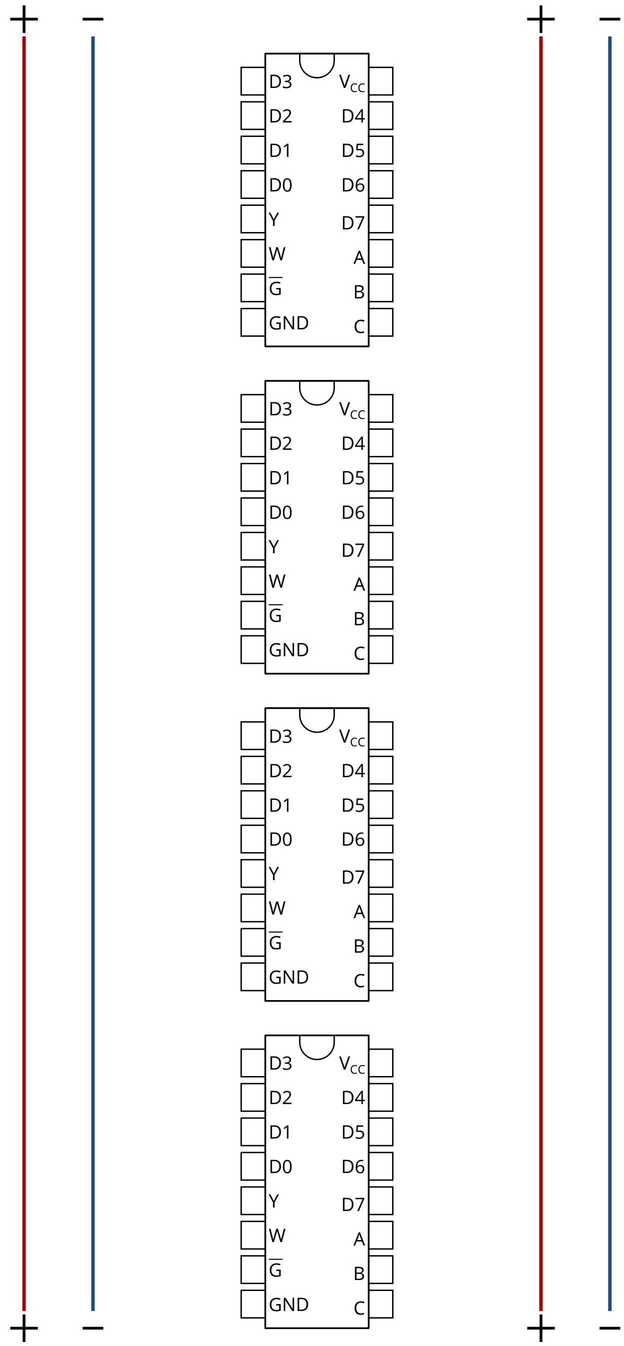


P7: What logic value should be applied to the $\overline{G}$ pin for the multiplexer to operate? We don't traditionally show this on the logic diagram, but it matters a lot! *Hint: Refer to the truth tables on page 2 of the datasheet for the 74HC151 multiplexer IC .*

P8: Finally, draw out the full 2-bit adder circuit as you would lay it out on the breadboard, showing all connections.

You'll have lots of connections to V_{cc} (aka V_{dd}) and ground, so use the power and ground "rails" to simplify your drawing.

Make sure to clearly label the inputs, outputs, and carry wires.



3 In lab

L1: Start by building the least significant bit of your adder.

You have two options for testing your adder: you can use the switches from the previous lab with some LEDs to show the output, or you can use the Digital Discovery. Information on how to use the Digital Discovery is included at the end of this handout.

L2: Finish building your adder, testing each piece as you go. (Don't forget to take notes!)

L3: Test your complete adder and record enough results to prove that it works.

L4: Optional: How long does it take your adder to reach a stable (and correct) result after the input changes? Document your process for figuring this out, and be sure to capture some screenshots or other supporting data for your lab report.

Some hints:

- Figure out what the worst case is (e.g., which bit flip will cause a change on the critical path, and which result bit will be the last to settle). From here you should be able to estimate the propagation delay on the critical path using information from the datasheet.
- You'll need to monitor both the input signal that changes and the output signal. You can use two channels on the oscilloscope or two bits on the Digital Discovery. The Digital Discovery is convenient because you can use it to drive the input directly while monitoring the output.
- The propagation delay is pretty fast (less than a microsecond), so you'll need to set up the instrumentation carefully to catch the event. We're happy to help you figure this out.

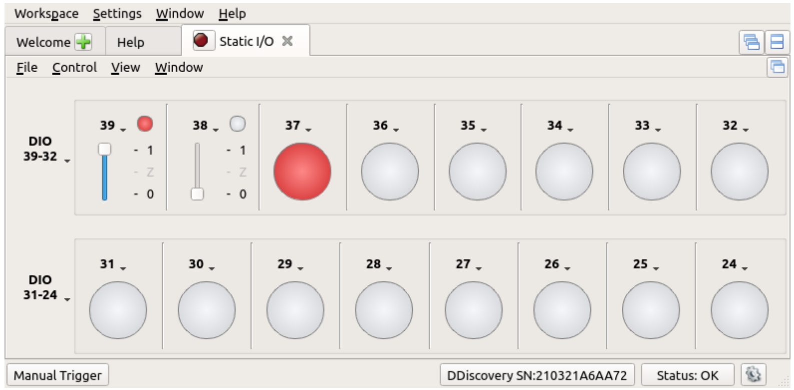
Using the Digital Discovery

You can use switches and LEDs as the inputs and outputs of your adder, but a more powerful approach is to use a *pattern generator* and a *logic analyzer*. A pattern generator is the digital equivalent of a waveform generator: it generates patterns of bits as a function of time. A logic analyzer is the digital equivalent of an oscilloscope: it displays the logic level (high or low) as a function of time.

The Digital Discovery is sort of a digital-logic swiss army knife: it combines a pattern generator and logic analyzer, as well as some other tools. We'll start by using its static I/O, i.e., outputs that you can control manually and inputs that appear visually on or off.

A1: Use two wires from the Digital Discovery to drive the inputs, and a third to monitor the output. Don't forget to connect one of the black wires to ground! *Note that only pins 24-39 can be outputs.*

1. Use the USB cable to connect the Digital Discovery to the computer and launch the Waveforms software. Search for "Waveforms" or click the icon.
2. Open the "Static I/O" pane from the left side of the start window.
3. Using the drop-down dialogs on each port number, set your two inputs to be push-pull switches, and the output to be an "LED".

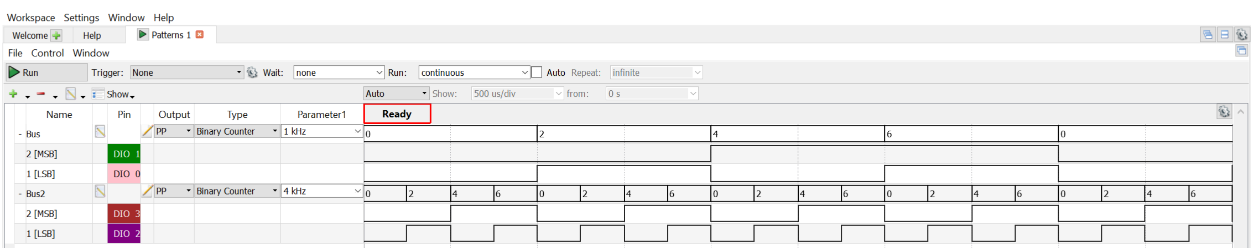


4. Toggle the switches to control the inputs, and visually check the output.

Ideally we would test the adder with every possible combination of all 4 input bits, which would be $2^4 = 16$ tests. Rather than click all the buttons by hand, we can automate it.

You can add a "bus" (a collection of signals), and then drive that bus with a counter. As the counter counts up in binary, it steps through every possible combination of the two bits.

If we set up both M and N as 2-bit buses driven by counters, and make one of the counters run $4\times$ faster than the other, then we will end up creating every possible input combination, as shown below:



A2: To do this with Waveforms:

1. Close the static I/O tab (since it will conflict with the pattern generator)
2. Open the pattern generator pane and create two buses, one for each input (A, B).
3. Set the type to "binary counter". Set one frequency to be $4\times$ the other (the dropdown doesn't have multiples of 4, but you can type a numbers in).
4. Click "run" to start the waveform generator, and open a new "Logic Analyzer" tab.
5. Add both your inputs and your outputs as buses here, so you can see both the decimal and binary values of the inputs and outputs.
6. Click "single" or "record" to capture the results, and check that the addition is working properly.

A3: This approach still relies on manually inspecting the output to confirm that it is correct. However, the Digital Discovery has a scripting interface which you can use to drive the inputs and read the resulting outputs, so you can automate all of the checks. If you'd like to do this, there is a starter script in the Lab 2 documents folder on the course website. *Make sure to include your completed script in your lab report!*

4 What to turn in

Your lab report should contain:

- Standard "front matter" (see the lab reports handout).
- Logic diagrams and circuit layout, updated as appropriate. Make sure to include the LEDs/switches/etc. as well.
- A photograph of your completed circuit.
- Test results demonstrating that your circuit works. This could be a written table or a screenshot from Waveforms, depending on how you tested it.
- If you measured the propagation delay: your plan for determining how fast the adder's output changes when the input changes, your results, and screenshots from Waveforms or pictures of the oscilloscope to support your conclusion.
- A description of problems you encountered, your debugging process for identifying them, and your solutions.
- Answers to the following questions:
  - What was the most valuable thing you learned, and why?
  - What skills or concepts are you still struggling with? What will you do to learn or practice these?
  - How long did it take you to complete the lab? This will help calibrate the workload for future iterations of the course.