# Lab 3

# Adders

Updated: January 20, 2020

## 3.1 Introduction

One of the primary goals of this course (Digital Logic Design) is for you be able to design logical circuits that implement a desired function. To do this, you often have to work in reverse—given a desired function, what circuit would produce it? This lab is the beginning of getting you to think in that mindset. In addition, this lab also introduces you to a family of logic "chips" (integrated circuits or ICs) that can be used to design simple logic circuits.

## 3.2 Objectives

After completing this lab, you should be able to:

- Develop circuits that implement a given logic function
- Describe the operation of half, full, and ripple adders
- Develop a moderately complex circuit on a breadboard using standard electrical parts
- Test and verify operation of the circuit

## 3.3 Background

Logic circuits began as discrete components (as in the previous lab, individual resistors and transistors), which were then packaged into "integrated circuits" (ICs or "chips") that might contain several individual gates or more complicated circuits. While these ICs have been superseded by more powerful and smaller devices, they are still in production and used on rare occasion. We will make use of them here because they are a good next step before getting into *programmable* logic devices.

## 3.4 Pre-Lab

Adders do what their name implies—they add binary numbers. A *half adder* (HA) adds two bits (we'll call them $A$ and $B$) and produces a two-bit result, a sum ($S$) and a "carry" output ($C$). Consider for a moment why two input bits requires two output bits. What are the decimal equivalents of the possible values of the inputs and outputs? Print the page titled "Pre-lab Submission Page" at the end of this handout, and fill in the table with the outputs of the half adder. Convert the combined output $[CS]$ to its decimal equivalent. (*Hint*: How many ones are there in each row?)

A *full adder* (FA) includes a carry input ($C_{in}$) with the two inputs of a HA, for a total of three bits, but also produces a two-bit result, carry out ($C_{out}$) and sum ($S$). Again, consider the possible values that the output can be. Fill out the truth table for the full adder. (Same *hint*.)

The additional carry input allows FAs to be chained together to form a multi-bit adder. This is a called a "ripple adder" because the carry signal "ripples" through each stage (or bit). We'll explore the inner workings of the ripple adder in the lab, but for the pre-lab, complete the truth table for a two-bit ripple adder. The behavior is the same as the HA and FA, except you can't simply count the number of ones. Inputs $A$ and $B$ need to be treated as binary numbers that are each two bits (e.g. $A = 11_2 = 3_{10}$). The sum output is also two bits, and the combined output of $[C_{out}S_2S_1]$ can be treated as a three-bit value. Convert both inputs and outputs to their decimal equivalents for this truth table.

## 3.5 Materials

You will need:

- two 7486 Xor ICs

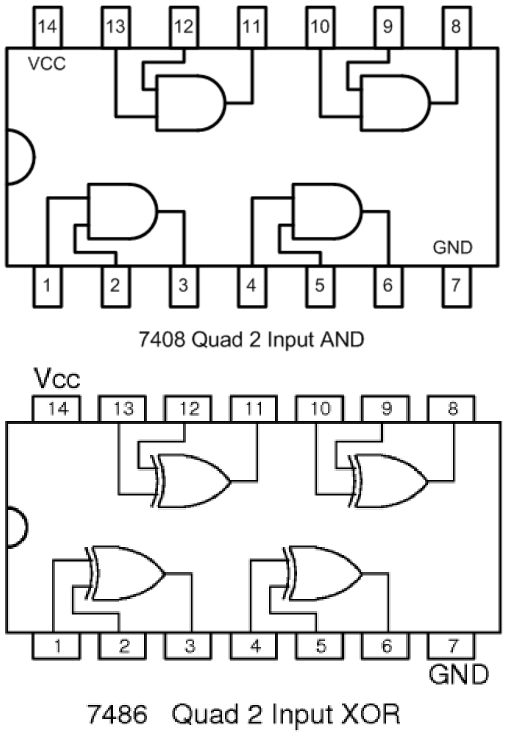

Figure 3.1: Pin diagrams for 7408 and 7486

- two 7408 And ICs
- breadboard
- wires

The pinout diagrams for these chips are shown in Figure 3.1.

## 3.6 IMPORTANT! Lab Procedures

To make your circuit easier to test and debug (check for errors), you should follow a few good practices:

- Typically, you should draw both a circuit "schematic" and a "wiring diagram." For logic circuits, a schematic is the "gate level" representation, showing the connections between gates. A wiring diagram is different from a schematic in that it uses the actual components and shows every connection that must be made. For example, your wiring diagram should show to which pins on each chip you are connecting and include the power (+5V and GND) connections.
- Color code your wires. Typically, DC power (e.g. +5V) is red and ground is black. Beyond that, develop your own scheme that has some kind of structure. For example, you could make the A inputs one color and B inputs another color, or if you don't have enough colors, all of your inputs might be the same.
- Arrange your components with plenty of space between them for connecting wires.
- Try to minimize wire lengths whenever possible.
- Order your inputs and outputs according to bit significance. In other words, when you look at your switches or LEDs, they should read like a binary number, with the highest or most-significant bit (MSB) on the left and lowest or least-significant bit (LSB) on the right. Do this even if it means longer wires (or an additional wire) because it will be much easier to read your input/outputs combinations and verify the operation of your circuit.
- When trying to remove chips from breadboard, slide the tip of a mechanical pencil or pen along the breadboard groove under each end of the chip. This will push the chip upward without bending or breaking pins. Alternatively, you can pinch the top and bottom with your thumb and index finger, pressing the tips of your fingers against the breadboard. Use the tips of your fingers as leverage to pull the chip *straight* up out of the holes.

## 3.7 Experiment

### 3.7.1 Half Adder

Now that you have the truth tables from the pre-lab, we need to determine the circuit that produces that output. The HA is easiest, so let's start with it. Look at the columns for the carry ($C$) and sum ($S$) outputs in your truth table. These are actually one of the basic gates! What are they? On the Circuit Drawing Page at the end of this handout, draw the HA schematic by drawing these two gates with wires showing the inputs $A$ and $B$ and the outputs $C$ and $S$.

Before building, create a wiring diagram from the schematic you just drew. Think about the arrangement of your chips as you select which gates and pins to use. Also think about where your inputs and outputs will connect. For the HA, you will have two inputs and two outputs. Be sure these read from left to right on the switches and LEDs, as discussed in the Lab Procedures section. Build your circuit according to your wiring diagram. Test it and demonstrate it to your instructor. Take a picture to include in your report.

### 3.7.2 Full Adder

The truth table for the FA is slightly more complicated, and you probably haven't yet covered how to implement it directly. However, we can break it up mathematically. We can add two of the bits together and get an intermediate answer, then add the final bit to this. The first part of this is just a HA.

The second part involves adding a two-bit number and a one-bit number. When adding a one-bit number to a two-bit number, we add the LSBs and then see if there is a carry to add. Let's start by adding the two LSBs - the sum from the first HA and the carry in bit. Here we have a second HA!

The second HA has a sum bit, which goes out as the sum of the FA. We also have a second carry bit, which we must combine with the carry of the first HA. Let's think about these two carries. In our case, the largest number we can produce is 3 (binary 11), so we cannot have a carry from both the first HA (the two-bit number) and from the second addition of the LSBs. Why? In your report, complete the expanded truth table in Table 3.1, and use this to prove the statement just made. Note: $c1$ and $s1$ are the carry and sum output of the first HA.

Table 3.1: FA expanded truth table

| $C_{in}$ | A | B | c1 | s1 | c2 | s2 | $C_{out}$ | S |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | ? | ? | ? | ? | ? | ? |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

The net result is we will have one or the other carry active, but not both, and we just need to know when either of them is on. Two gates can meet the requirements. Which ones? Which should we use? Why?

Draw your circuit schematic and wiring diagram on the Circuit Drawing Page. Build your circuit according to your wiring diagram. Test it and demonstrate it to your instructor. Take a picture to include in your report.

### 3.7.3 Two-Bit Adder

Now we want to build the two-bit ripple adder. Using another pair of ICs, build and test another FA. Once you have two working FAs, connect the carry output of one adder to the carry input of the other adder (instead of a switch). The first adder will be your LSB, and the second adder is the MSB. Make sure these are arranged on the LEDs and switches so that they read right-to-left, LSB-to-MSB. You should have 5 switches as inputs (3 to LSB adder and 2 to MSB adder) and 3 LEDs as outputs (1 from LSB sum, and 2 from MSB sum and carry out). Test it and demonstrate it to your instructor. Take a picture to include in your report.

## 3.8 Deliverables

Submit a report using the provided template that includes the following:

1. The Circuit Demonstration Page with the HA and FA schematics and wiring diagrams, and three instructor signatures
2. Pictures of each of your circuits
3. Proof that the carry outputs of the first and second stage HAs cannot both be high at the same time (truth table)
4. Answers to the questions: Which gates could we use for combining the carry bits? Which one should we use and why?

# Circuit Demonstration Page

Half adder schematic

Half adder wiring diagram

Half adder: ____________

Full adder schematic

Full adder wiring diagram

Full adder: ____________

2-bit adder: ____________

Forrest Knee

## Pre-Lab Submission Page

Half adder truth table

| A | B | C | S | Decimal |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 2 |

Full adder truth table

| $C_{in}$ | A | B | $C_{out}$ | S | Decimal |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 2 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 2 |
| 1 | 1 | 0 | 1 | 0 | 2 |
| 1 | 1 | 1 | 1 | 1 | 3 |

2-bit adder truth table

| $C_{in}$ | $A_2A_1$ | $B_2B_1$ | $C_{out}$ $S_2$ $S_1$ |
|---|---|---|---|
| 0 | 00 | 00 | 0 0 0 |
| 0 | 00 | 01 | 001 |
| 0 | 01 | 01 | 010 |
| 0 | 10 | 01 | 011 |
| 0 | 10 | 10 | 100 |
| 1 | 10 | 10 | 101 |
| 1 | 11 | 11 | 111 |

Decimal equivalent

| $C_{in}$ | A | B | Out |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 2 |
| 0 | 2 | 1 | 3 |
| 0 | 2 | 2 | 4 |
| 1 | 2 | 2 | 5 |
| 1 | 3 | 3 | 7 |