NAND 2/3: Arithmetic

Kerstin Eder

September 23, 2021

In this lab, you will build on your Logisim knowledge from the previous lab, and also use the NAND boards to experiment with new, slightly more complex logic designs. The labs allow you to develop practical skills that deepen your understanding while building the fundamental components used in computer architecture.

Goals of this lab

- Continue to familiarise yourself with Logisim for logic circuit design, simulation and testing.
- Transfer designs onto NAND boards.
- Combine your own work on the NAND boards with the work of others, **where possible**, demonstrating how simple structures can be used to build more complex designs.

Lab overview and guidelines

The lab is broken down into tasks, with each task of increasing difficulty or requiring some additional understanding. It is suggested that you proceed in the order the tasks are given in this lab sheet.

To achieve a good level of understanding and to complete this lab in the allocated time, you will need to come well prepared to this lab session.

Therefore, please make sure you familiarise yourself with all the tasks before starting the lab. In general, it is assumed that you have invested a significant amount of study time prior to attending the lab, e.g. watching the recording, engaging in the exercises, reading up on the topics covered in the lectures, practising, designing and testing your circuits in Logisim in your own time in preparation.

Remember that you can get support from the students sitting in close proximity to you, i.e. your informal lab group, and from your TA during the weekly lab sessions.

1 Preparation

1.1 Logisim

Logisim is a Java application, it can be obtained from http://sourceforge.net/projects/circuit/. Simply download and run it, for example:

```
java -jar ~/Downloads/logisim-generic-2.7.1.jar
```

1.2 NAND boards

1.2.1 Introduction

The NAND boards are a set of four chips, each containing four `NAND` gates. Each gate has a set of input pins and output pins. In addition to the pins for the gate input and output, there are pins providing `logic 1`. You may notice that there are no `logic 0` pins. This is because an unconnected input will default to 0 (there is a *pull-down* circuit that keeps the input at 0 if it is unconnected). LEDs on the boards signify the output level of each NAND gate. The boards are powered by a USB cable.

Current versions of the boards also feature four push-buttons, which provide `logic 0` when not pressed, `logic 1` when pressed. They are each connected to a block of four pins at the left-hand side of the board.

These boards are *open source*, and were created by Drs Simon Hollis and Dan Page. There is a GitHub repository containing the designs, photos, and more information on the boards: https://github.com/danpage/nandboard. You may want to use this for further reference.

1.2.2 Access to NAND boards

NAND board kits have either been sent out to you or you have been asked to pick a NAND board kit up. Please be patient as some may still be in the post or awaiting collection as we attempt this lab. If you do not yet have your NAND board kit you can still complete the Logisim part of this lab. For the practical work in this section, however, you need a real NAND board. **Please inform your TA as attendance is taken whether or not you have received your NAND board kit.**

If a piece of hardware does not work, or you accidentally break something, you must report this to the TAs in the lab as soon as possible. The boards are quite robust and rarely break. However, they are not expected to last forever and we will not blame you if one reaches the end of its natural life while in your hands. We will do our best to provide a replacement where possible.

1.2.3 Familiarisation with the NAND boards

To familiarise yourself with the NAND boards, please watch the introduction video at:

https://www.youtube.com/watch?v=DJDXp7yXp-w&feature=youtu.be

Figure 1 illustrates the basic layout of the NAND boards. You may want to use this as a template for designing your circuits before building them.

1.2.4 Transferring Logisim designs onto the NAND boards

Although the NAND boards are very simple, they can quickly become a confusing mess of wires. It is important to follow a good process for implementing designs on them.

It is most strongly recommended that you design and test using Logisim before building. This means that you first print or draw a Logisim tested design. You then choose where each NAND gate will go on the NAND board, before you connect wires in a sensible order, *marking down* when each part of your design is completed. That way, you are less likely to miss a wire, or get confused. The NAND board layout in Figure 1 can serve as a template for your designs.

Please note that, while the Logisim simulator offers a wide range of logic gates, the NAND boards only have NAND gates with *two* inputs. The fundamental idea is that the designs you build and test with the Logisim simulator are a one-to-one match with what you implement on the NAND boards. So, it is really important that your Logisim designs are directly implementable on the NAND boards, i.e. without modification of the logic. In particular, do not use 3-input NAND gates in your Logisim designs because the NAND board does not have these.

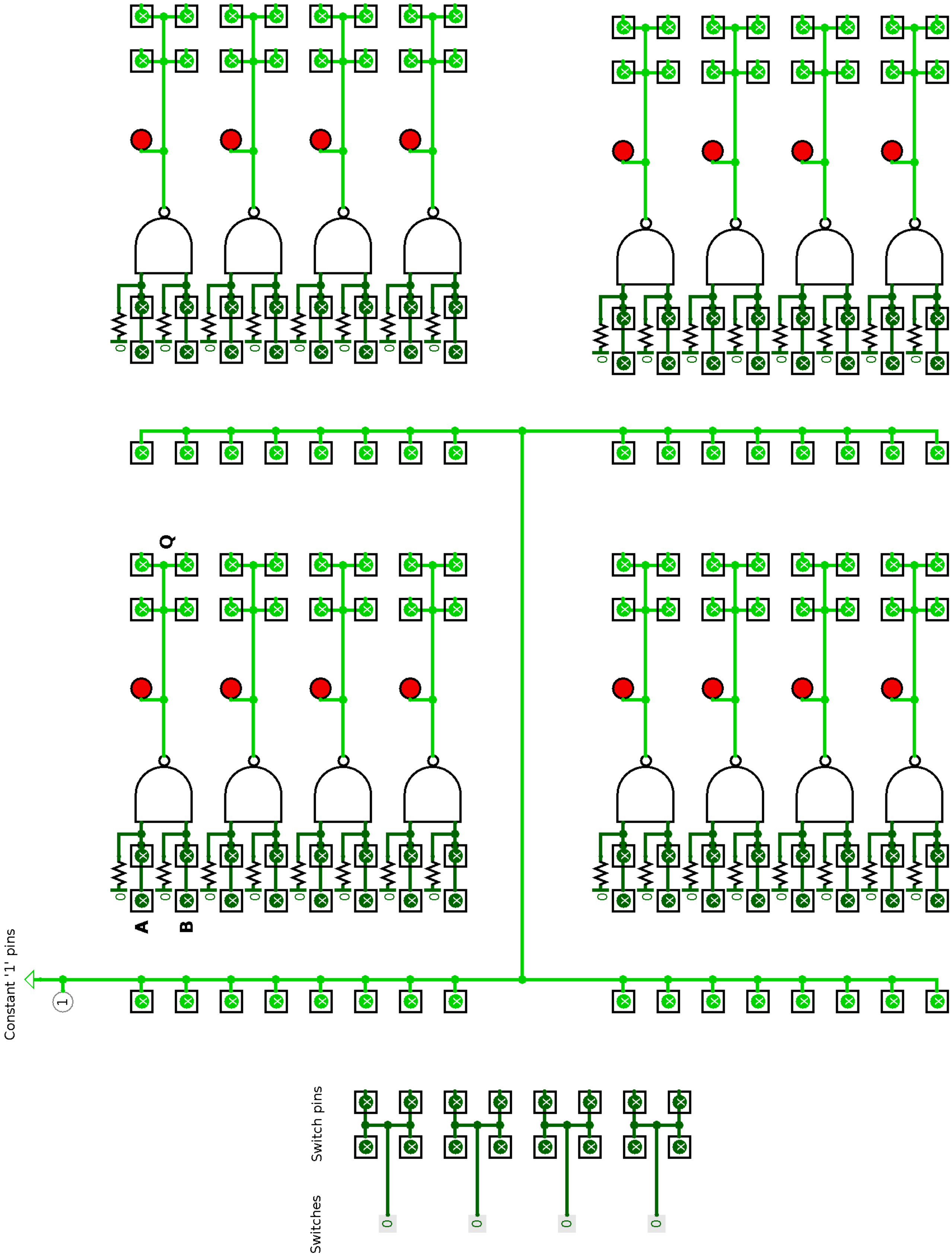

Figure 1: NAND board layout

1.2.5 Sensible layout of designs

To make life easy for yourselves, please try to observe the following conventions, wherever possible:

- Use the push-buttons as inputs, wherever possible.
- Try to use the bottom-right gates as outputs, so that the LEDs are easy to see.
- When demonstrating your physical implementations to others, e.g. students in your informal lab group or your TA, make it clear to them which button/wire is input A, input B, etc. Similarly, be sure to identify which outputs are Q, S, etc. This is also essential when you come to connect your design to other boards!

2 Stepping stone: Half adder

This section and the following sections contain the tasks that we recommend you should perform in the lab. Please **read the whole lab sheet first**, so that you can manage your time on each task, and see which tasks are connected.

Create a new Logisim file and implement a half adder. Do the same with the NAND board. The lecture slides on **"Simple devices"** contain material that may help you with this.

Test your designs to convince yourself that they work. Find a systematic way to select your inputs, also called *test patterns*. How many different test patterns are there to try?

This task is a stepping stone to building a full adder. Therefore, it is strongly recommended that you perform this task before moving to the full adder in the next section.

3 Full adder

The half adder has no carry capabilities. To make a more useful adder, you should build a full adder, which provides carry-in and carry-out. Again in Logisim and on the NAND-boards, implement a full adder.

Test your full adder systematically, both in Logisim and on a NAND board.

Think carefully about how best to demonstrate that your full adder works correctly, e.g. which test patterns make interesting tests and why? How many test patterns are there in total?

Approach and design optimisation

To start with, you can implement your full adder based on half adders. How many half adders are required to build a full adder? How many NAND gates do you need to implement a full adder in this way?

Now, optimise your adder design. What is the minimum number of NAND gates required to implement a full adder? Implement and test this optimised full adder design, both in Logisim and on a NAND board.

First, use a truth table to demonstrate that your optimised design implements a full adder. Next, use Boolean algebra to demonstrate that your optimised design implements a full adder.

4 Chained full adder

Please note, this task requires physical interaction. Therefore, students who work remotely and those self-isolating will not be able to perform this task. However, students in the lab should be able to perform this task and demonstrate their results.

This task involves combining several full adders. Although this is the most complex part of the lab, it should also be the most fun!

In principle, we could now connect all the n full adders built in this lab to obtain one large n-bit ripple carry adder. In the lab, where possible, please connect your full adder to working full adders of other students, not exceeding eight adders per chain. Please note that you will need to connect all adders to be chained together to the same power supply. Suitable power supplies will be provided in the lab.

It is possible to observe the signal propagation delay when several adders are chained together. Try this out and discuss with other students what you can observe and why.

Test your chained adder systematically. Think carefully about how best to demonstrate that your chained adder works correctly. Which inputs make interesting test cases and why? Which inputs exercise the carry chain?

It is possible to combine several Logisim adder designs into a chained full adder. So, if you cannot perform the task above using several NAND boards, you may want to build a 4-bit or 8-bit ripple-carry adder using Logisim. You can then test your chained adder design systematically and answer the questions on testing given above.

5 Extension

The following is an extension piece of work that you can complete in your own time *if you wish*. It may improve your understanding.

Extend your existing design to become an adder-subtractor. You will need to add a multiplexor with the capability to invert one input. A control signal will switch the multiplexor as well as provide the carry-in signal for 2's complement negation.

Again, test your adder-subtractor systematically. Identify interesting test patterns to demonstrate that your design works correctly.

If you wish to put this onto a NAND board, be careful to use 16 NAND gates or less. You may need to perform some research to find an optimised design, or you may be able to identify redundant parts that can be removed.

Well done **for completing the second NAND lab.**