ES 4 Lab Reports: A Brief Guide

What's the point?

Lab reports are a pain to write and take work to grade, so why have them? In short, reports give you practice with three important skills:

- **Documenting your work**. In a few weeks, you will have forgotten almost all of the details of what you did — and when you run into the same problems in the future, you'll be glad you wrote it down!
- **Communicating effectively**. Have you ever tried to fix someone else's code or circuit with no documentation or explanation of how it was supposed to work or what was wrong? Don't do that to the poor souls who will have to build on your work in the future.
- **Analyzing your process**. Think of a time when you wasted hours trying to do something when a small change would have made it work the first time. Lab reports give you that opportunity to stop *doing* and *think* about your approach, which ultimately makes you more efficient.

What goes in my lab report?

Standard "front matter"

You should include the following items on the first page of every lab report:

- A title with the lab number
- Your name
- The date
- Your TA and lab time
- A one or two sentence description of what you're trying to accomplish in the lab.

Why? This helps us figure out what's going on if you accidentally submit in the wrong place or make some similar mistake. It also provides essential context for the report five years from now when you can't remember anything about where it came from.

Design

The design section should capture all of the information needed to precisely replicate what you did. This includes any of the following that apply:

- Boolean equations
- Truth tables
- Logic diagrams
- Schematics
- State diagrams
- Block diagrams

You're welcome to copy these from your prelab, but you should correct them as necessary and include any additional design notes that you create.

The end of this section should include a photograph of your completed circuit. Take a minute to compose this nicely and capture the best view you can. You should be able to rebuild your circuit from the photo if you had to.

Why? If you (or a colleage) ever needs to replicate what you did, you want that design to be clearly documented at a conceptual level (e.g., logic diagrams and schematics) and at a physical level (photographs). The first captures the work you did to create the design; the second captures what was actually built. This is particularly important if you rebuild the design and find that it does something different than the original — the difference could be due to an implementation quirk in the original that wasn't part of the conceptual documentation, and you need to know that!

Lab journal

Take notes about what you're doing as you build, test, and debug your designs, and include these notes in this section.

This should *not* be a beautifully curated narrative of your woes and triumphs in the lab. Instead it should be a real-time log of what you're doing, what you're thinking, and what you're going to try next.

In particular, each time you encounter and work through a problem, take the time to record your responses to the following questions:

- How did you know there was a problem?
- What are the possible causes?
- How did you resolve it?
- What did you learn?

You'll go through this problem/analysis/solution/reflection cycle several times during the lab, whether consciouly or unconsciously. The goal of the lab journal is to make it a conscious process, helping you to become more methodical in your debugging. Debugging is a huge learning opportunity, but the biggest gains only come when you take the time to reflect on the process.

You can do this however you'd like, but my recommendation is to do it on a tablet or in a physical notebook. For the first couple labs it will be very helpful to draw schematics and make free-form notes as you work, which is hard to do well on a laptop.

Why? I made a whole video to explain the "why" for lab journals; go view it in the course website.

Testing

How you test each lab will differ, but you should always include a description of your plan for testing your circuit, and a table or diagram of your results.

Why? This is the proof that your design works. It's good to demonstrate your working circuit to a TA, but a brief look is rarely sufficient to prove that the circuit works 100% correctly.

Once you graduate, you're going to be building things for which no one has the "right answer". It will be often up to *you* to prove that your design works, and to document those results in a clear and convincing way.

Reflection

For the reflection section, answer 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? We've reorganized many of the labs this year in attempt to balance the workload, and we want to know if this is working!

Why? This section serves two purposes: First, it forces you do some "metacognition" (i.e., thinking about your thinking) and conciously consider ways to improve. Second, it helps the teaching staff gauge how the labs are going and gives us data to help improve the course.

How should it be formatted?

However you want, as long as it's organized and easy to read. There is a simple template on the course website if you'd like a starting point.

Every company and academic journal has a different style for reports and papers, so there's little to be gained by imposing a particular format on your reports. We're not concerned about your ability to tweak the margins or bullet styles in Word; we're concerned about your ability to document and communicate your work effectively.

How long should it be?

Long enough to explain everything clearly, and no longer!

Your report should be readable by someone who has taken a digital logic course and has a copy of the lab handout. There shouldn't be any major jumps in thought, or any parts of the design which are not documented.

On the other hand, you don't need to repeat anything that's in the lab handout or the textbook (i.e., you don't need to tell us what an AND gate is, or explain how binary numbers work).

The goal is to write a document that lets knowledgeable people quickly figure out what you did. If adding detail helps, then add detail. If some details make it redundant or harder to follow, then cut them out. Getting the level of detail right is a skill which you will develop and refine during your time at Tufts and during your career as an engineer.