Hands-On Circuits in CS1*

Conference Tutorial

Elena Miller, Katy Shaw, Zachary Dodds

Harvey Mudd College
Claremont, CA 91711

{zdodds, elmiller, kshaw}@g.hmc.edu

Many introductory computer science courses - including ours - focus on the science and art of programming. In such classes - including ours - the predominant activity is editing: editing one's electronic documents and editing one's conceptual models of computation. Both of these important skills rest on the foundation of a conceptual model of computation's physical processes.

Today's computing interactions rightfully abstract away the switching of transistors and the cascade of logical circuits. As CS1 finds itself a GenEd requirement for more and more students and institutions, as is true for us, experiencing its bigger-picture connections becomes more important. In order to make the physical underpinnings of computing accessible to *all* students, many of whom will never again touch a transistor or logic gate[1], we have developed, tested, refined, and deployed a short activity accessible to all students, regardless of background. This tutorial invites its participants to try out these activities and materials.

Participants in this tutorial will work through our "transistor and logic gates" lab, in which they will create (1) a NOT gate from two transistors and (2) an arithmetic circuit from logic gates. Both are accomplished with low-cost, off-the-shelf[2] parts, totaling about $10. No specialty equipment - no scopes or generators - are involved, and the materials are extremely portable. Through the tutorial, participants will

---

[1] We will be delighted if such an accessible experience increases this number!

[2] This does require a shelf with electronics components, which is perhaps less common in '23 than in prior decades. Happily, Amazon's shelves have it all!

(re)experience setting up their own breadboard and assembling resistors, transistors, wires, and LEDs.

As a GenEd, our course includes assignments not present in every Comp1/CS1, e.g., hands-on assembly language, (simulated) circuit design, and several "non-programming" computational models, e.g., FSMs and TMs. We will share how we integrate this hands-on lab so that prior experience is neither necessary nor expected -- and such that no future experience is anticipated. We will share, too, the online resources we find complement these paths, e.g., CircuitVerse and the NAND Game [2,3].

We are grateful to stand on the shoulders of prior work demonstrating the value of hands-on circuit design [1]. In addition, we acknowledge the ambition and scope of courses, such as *From NAND to Tetris*, that center the technology stack that is modern computation [4]. For us, practicing software-authoring is of paramount importance for all students, so this tutorial's more modest approach works well.

In such a spirit of adaptation, we will be delighted if participants adapt this tutorial's materials to suit their home institutions' computing pathways. Plus, whether adaptation is possible or not, we are excited for *all* participants to have built a logic gate from physical transistors and an arithmetic circuit from physical logic gates -- and to encounter how accessible such experiences can be.

References

[1] Wu, H.-T., Hsu, P.-C., Lee, C.-Y., Wang, H.-J., & Sun, C.-K. (2014). The impact of supplementary hands-on practice on learning in introductory computer science course for freshmen. Computers & Education, 70, 1-8.

[2] CircuitVerse. (n.d.). CircuitVerse - Online Digital Logic Circuit Simulator. Retrieved June 29, 2023, from https://circuitverse.org/

[3] The NAND Game https://nandgame.com/

[4] Schocken, Shimon. Nand to Tetris: Building a Modern Computer System from First Principles. SIGCSE '18: Proceedings of the 49th ACM Technical Symposium on Computer Science Education February 2018, 1052. doi.org/10.1145/3159450.3162353