LEARNINGNERD

HOME › CATEGORIES: DAILY LEARNING NOTES

Daily Learning Notes for January 29, 2016

Woohoo! Yesterday I finished the fourth week of NAND2Tetris! Today I'm going to start on week five, learning about computer architecture and finally completing a design for a working (simulated) computer!

Chapter 5 reading notes

Learning materials for this week:

- Chapter 5 in the book (PDF)
- Video lectures on Coursera
- Project for week 5

They start this chapter with the concept of a **stored program**, which apparently was formulated by a few different mathematicians independently back in the 1930s. They say this concept is the very foundation of modern computer science!

So what is it? They don't spell out the definition liek they do for some other tern here's my understanding of it: the stored program concept is the idea that you can store a set of instructions the same way you can store date, so that the instructions are decoupled from the hardware that runs them. This is why a general-purpose computer is possible! What a simple idea! And yet it took *so* long for humanity to realize it.

Then they talk about the **von Neumann architecture**, a model for the basic design of a computer that stores data and instructions in memory, which interfaces with a central processing unit, which takes input and gives output using a control unit, registers, and an arithmetic logic unit.

Oh, how about that! The rest of the background section is all just review of stuff they've covered in the previous weeks! I guess this week will be much more focused on the project than on the reading. That's cool.

The end of the chapter covers some new material, but I don't really understand it. Something about "two-cycle logic" – most computers fetch instructions and execute them at separate times, but the Hack computer can do both in the same cycle because the address space is partitioned into two parts. They say that with this simpler hardware design, "programs cannot be changed dynamically." I don't know what that means, though!

They also point out some other important difference between the Hack computer and modern computers, like the lack of a graphics card!

Most importantly, they include this note:

> "Finally, it should be stressed that most of the effort and creativity in designing computer hardware is invested in acheiving better performance."

They completely skipped over anything related to efficiency and optimization in this course! But that makes sense. They cover a lot for an introductory course, and in any case, you need to understand how something works before you can start worrying about how to optimize it.

Specifications for the Hack computer

The rest of the chapter is dedicated to the specifications for the Hack computer, which I now need to build. Eeek!

OK, I read over all of the specifications. It makes sense in a big-picture sort of way. I have to connect a bunch of parts (most of which I built for the previous projects) and some built-in parts like the screen and keyboard. Learning how to build a screen and keyboard from first principles is definitely out of the scope of this course, but obviously I want to learn about that one day too! Ah well, one step at a time.

The main challenge will be to build the CPU. I need to design something with logic gates that can decode instructions written in machine code and do the appropriate operations with the ALU and the registers. Oh, and it also needs to fetch the next instruction, which depends on the previous instruction.

I thought that controlling the program counter would be a challenge too, but they say that it can be "easily effected" with their proposed CPU design. They include a diagram for a CPU design to help me out, but they just don't include all the control logic.

I guess I'll start building it after the weekend. I'm excited!

New concepts and jargon

- A **stored program** is a set of instructions stored in memory, decoupled from the computer's underlying hardware.

- The **von Neumann architecture** is a design for a general-purpose computer that stores data and instructions in memory, which interfaces with a central processing unit, which takes input and gives output. (No super concise definition here, but that's my best attempt for now!)

Questions

- What are the alternative models to the von Neumann architecture?
- Why did it take humanity so long to come up with the idea of a stored program?
- What is two-cycle logic and what does it have to do with partitions of the address space?
- Why does the Hack computer's partitioned address space mean that "programs cannot be changed dynamically"?

Learning summary (#TIL)

- I learned the gist of what the von Neumann architecture is and how the parts of a computer work together!
- I reviewed all the specifications for the compuer I'm going to build for this week's project.

Next steps

- Complete the project for week 5

Time breakdown

- Study time: 2 hours 20 min
  - Active reading: 1 hour 33 min
  - Watching video lectures: 47 min

Written by Liz Krane

Liz is a full-time nerd, sharing everything she learns as she tries to learn everything!

Share on Twitter | Share on Facebook | Share on Google+

Updated January 29, 2016

0 Comments **LearningNerd** 1 **Login**

♡ **Recommend** **Share** **Sort by Best**

Start the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Be the first to comment.

ALSO ON **LEARNINGNERD**

English Grammar: Types of Clauses

1 comment • 7 months ago

sousou — HI, what about the concessive clause?

English Punctuation Overview

1 comment • 3 months ago

essay writing services — This is very much in demand to fix the punctuation in English learning approaches. In addition this is making

English Parts of Speech: Adjectives, Determiners, and Adverbs

1 comment • 7 months ago

stevej101 — Nice tips!

English Grammar: Basic Sentence Elements

1 comment • 7 months ago

case study help online — Thanks posting these important terms of the basic sentence elements for English grammar. These are all important

Subscribe **Add Disqus to your siteAdd DisqusAdd** **Privacy**