Course Overview

The Nand to Tetris course takes you on a self-paced fascinating voyage of discovery in which you will go all the way from Boolean algebra and elementary logic gates to building a Central Processing Unit, a memory system, and a hardware platform, leading up to a general-purpose computer that can run any program that you fancy. In the process of building this computer you will become familiar with many important hardware abstractions, and you will implement them, hands on. But most of all, you will enjoy the tremendous thrill of building a complex and useful system from the ground up.

The course homework consists of a series 6 projects:

- Project 1: Building elementary logic gates like And, Or, Not, Multiplexor, and more
- Project 2: Building a family of adder chips, culminating in the construction of an Arithmetic Logic Unit (ALU)
- Project 3: Building registers and memory units, culminating in the construction of a Random Access Memory (RAM)
- Project 4: Learning a machine language and using it to write some illustrative low-level programs
- Project 5: Using the chipset built in projects 1-3 to build a Central Processing Unit (CPU) and a hardware platform capable of executing programs written in the machine language introduced in project 4
- Project 6: Developing an assembler, i.e. a capability to translate programs written in symbolic machine language into binary, executable code.

You will build the hardware projects on your home computer, using a simple Hardware Description Language (HDL), learned in the course, and a hardware simulator, supplied by us. A hardware simulator is a software system that enables building and simulating logic gates and chips before actually committing them to silicon. This is exactly what hardware engineers do in practice: they build and test computers in simulation, using HDL and hardware simulators.

Prerequisite knowledge: in order to take this course you have to know... well... absolutely nothing about computer science or engineering! All the knowledge necessary to build the computer and complete this course successfully is given in the course lectures and projects. The only prerequisites are curiosity and passion to build a cool working example of the most important invention of the 20th century.

The course has a rich history: since we launched the course several years ago, versions of Nand2Tetris have been taught at Harvard, Stanford, Chicago, and many other universities and high schools around the world. Students who completed Nand2Tetris routinely say it's the best computer science course they ever took.

The Game of the Name: "Nand" is the name of the most elementary logic gate from which computers are normally built, and "Tetris" is the name of a mythological computer game. In "Nand to Tetris / Part I" (this course) we start with Nand gates, and gradually build the chipset and hardware platform of a simple computer system called Hack, as well as an assembler (the linchpin of the two courses). In "Nand to Tetris / Part II" (a follow-up course) we start from the Hack computer and build a modern software hierarchy that includes a virtual machine, a compiler for a simple Java-like language called Jack, and a basic operating system. The result is a general-purpose computer that can run cool games like Tetris, Snake, Pong, Space Invaders, and practically any program that comes to your mind. Part I and Part II are two separate and stand-alone courses, each granting its own credits and certificate. You can take both courses, or any one of them, in any desirable order, although it is recommended to take Part I and then part II.

Work Load: The Nand to Tetris / Part I course consists of 7 modules. In each module you will spend about 5-10 hours watching video lectures and building the module's project.

Grading: Projects 1 and 2 contribute 16% each to the total course grade; projects 3,4,5 and 6 contribute 17% each to the final course projects, for a total possible course grade of 100. You must pass every assignment with a score of at least 75 to pass the course.

The Course Instructors are Noam Nisan and Shimon Schocken.