Module 1: Boolean Functions and Gate Logic Roadmap

In a nutshell: We will start with a brief introduction of Boolean algebra, and learn how Boolean functions can be physically implemented using logic gates. We will then learn how to specify gates and chips using a *Hardware Description Language* (HDL), and how to simulate the behaviour of the resulting chip specifications using a *hardware simulator*. This background will set the stage for Project 1, in which you will build, simulate, and test 15 elementary logic gates. The chipset that you will build this module will be later used to construct the computer's *Arithmetic Logic Unit* (ALU) and memory system. This will be done in modules 2 and 3, respectively.

Key concepts: Boolean algebra, Boolean functions, gate logic, elementary logic gates, Hardware Description Language (HDL), hardware simulation.

WATCH:

- Unit 1.1: Boolean Logic
- Unit 1.2: Boolean Functions Synthesis
- Unit 1.3: Logic Gates
- Unit 1.4: Hardware Description Language
- Unit 1.5: Hardware Simulation
- Unit 1.6: Multi-Bit Buses
- Unit 1.7: Project 1 Overview
- Unit 1.8: Perspectives

DO:

Project 1: Elementary Logic Gates.

Before starting to work on Project 1, make sure you read Appendix A of the book, and the HDL Survival Guide (links to these resources are given below). There is no need to read these documents back to back. Rather, use them as technical references, on a need-to-know basis. That is, if you have an HDL problem, consult these references first before posting a question in the forum.

The outcome of your work will be a set of text files, containing the HDL code of the 15 chips that you are expected to build. If you succeed building all 15 chips, great. If you manage to build only some of the chips, you can still submit them for partial credit.

To ensure proper testing and feedback, the names of the HDL files that you submit must be the exact file names that appear in the nand2tetris/projects/01 folder on your computer. *Note that case matters*: a file named and.hdl or AND.hdl may fail our testing and grading procedures (the correct name must be And.hdl). You can save yourself all this hustle by neither creating nor naming any files: simply edit the *.hdl files in your nand2tetris/projects/01 folder using any text editor, write your HDL code in them, and save them using their given names.

You should pack all the *.hdl files that you wrote as one zip file named project1.zip (pack the files themselves, don't put them inside any folders), and submit it. If you build helper chips that were not specified by the course, you should include them in the zip file. You have an unlimited number of submissions, and the grade will be the maximum of all your submissions, so you can't lose points by submitting again.

If you are taking the course as an auditor, you can check your work yourself, using the tests described here. If you are taking the certificate option, submit your project zip file here.

READ:

- Chapter 1 of *The Elements of Computing Systems*.
- HDL Guide (except for A.2.4)
- Hack Chip Set (when writing your HDL programs, you can copy-paste chip-part signatures from here)
- FAQ, containing frequently asked questions by students. If you get an error message you can't solve, you should try looking here.

GET HELP:

- Module 1 Discussion Forum