



# Module 2: Boolean Arithmetic and the ALU

## Roadmap

**In a nutshell:** Using the chipset that we've built in the previous module, we will now proceed to build a family of *adders*-- chips designed to add numbers. We will then take a big step forward and build an *Arithmetic Logic Unit*. The ALU, which is designed to perform a whole set of arithmetic and logical operations, is the computer's calculating brain. Later in the course we will use this ALU as the centerpiece chip from which we will build the computer's *Central Processing Unit*, or CPU. Since all these chips operate on binary numbers (0's and 1's), we will start this module with a general overview of binary arithmetic, and only then delve into building the ALU.

**Key concepts:** Binary numbers, binary addition, the two's complement method, half-adders, full-adders, n-bit adders, counters, Arithmetic Logic Unit (ALU), combinational logic.

### WATCH:

- Unit 2.1: [Binary Numbers](#)
- Unit 2.2: [Binary Addition](#)
- Unit 2.3: [Negative Numbers](#)
- Unit 2.4: [Arithmetic Logic Unit \(ALU\)](#)
- Unit 2.5: [Project 2 Overview](#)
- Unit 2.6: [Perspectives](#)

### DO:

- [Project 2: Boolean Arithmetic](#)
- You should pack all the \*.hdl files that you wrote as one zip file named project2.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](#) .

### GET HELP:

- [Module 2 Discussion Forum](#)

