The Hardware Simulator is used for building and testing all the chips discussed in Nand to Tetris projects 1, 2, 3, and 5. Each chip is defined in a chipName.hdl file, and the most recent versions of these files are autosaved and persisted in your browser memory. This means that the next time you will use the hardware simulator, you will see the most recent versions of these files.

Submitting HDL files: If the course that you take requires submitting HDL files (e.g. for grading), you can write and test the files using this hardware simulator. When a chip passes its tests, you can copy its HDL code from the simulator's editor panel and paste it into any standard text editor.

The simulator features three panels: HDL editor, Pins panel, and Test panel.

HDL editor

To build or edit a chip (HDL program), select the chip name from the *Project* menu and the menu right next to it. This results in three actions: The chipName.hdl file is loaded into the editor, the chip's pins are displayed, and the chip's test script is displayed. Any change that you make to the HDL code is saved automatically.

Builtin chips: Each chip in projects 1, 2, 3 and 5 has a builtin version. The builtin version features the chip's interface, and a builtin implementation which is part of the simulator's software. The builtin version allows experimenting with the chip's operations before implementing it in HDL. To do so, select the *BuiltIn* toggle in the HDL panel and test the chip using either interactive or script-based simulation (described next).

Pins panel

Displays the names of the chip's input pins, output pins, and internal pins, and their current values. The pin values are computed by the chip logic when the user clicks the *Eval* button, or when an "eval" command is executed in the chip's test script. If a pin's width is more than one bit, a *dec*/*bin* toggle allows displaying its value in either a decimal or a binary format. To evaluate a chip interactively, change one or more of its input pin values, and click the *Eval* button.

A chip is said to be *sequential* if it contains a sequential chip-part, or if one of its chip-parts contains a sequential chip-part. In particular, the DFF chip is sequential, and all the chips that use it directly or indirectly are sequential.

The clock and reset buttons are enabled only for sequential chips. Clicking the *clock* button advances the clock forward in either one tick or one tock. The resulting time step is displayed. Clicking the *reset* button resets the clock.

Chip visualizations: Some builtin chips have visualizations, which are displayed automatically by the simulator (and can be turned off). The chip visualization helps inspect the chip's operations. For example, the ALU visualization displays the name of the current ALU operation, and the memory chips visualizations display their internal states.

Test panel

The test panel displays the test script supplied for the loaded chip. The "current command", which is highlighted in yellow, is the test script command that will be executed next. To test a chip, use the *step* button (executes the current command), *run* button (executes the entire test script, from the current command onward), or the *reset* button (makes the first command in the script the current command).

The compare file (if one is supplied) and the output file generated by the test script can be displayed by clicking the respective tabs.

The test script can be edited, but it is recommended to start testing the chip using the supplied script. Changes made to the test script (if any) are not saved.

Bug / issue reports

To report a bug or propose an improvement, click the *bug* icon. You will be asked to login to your GitHub account (if you have one).