The CPU Emulator is used for emulating the execution of Hack programs on the Hack computer.

A Hack program is a sequence of instructions written in the Hack machine language.

The Hack computer is emulated by visually rendering the following elements of the Hack hardware platform: ROM (instruction memory), RAM (data memory), D (data) register, A (address) register, and PC (program counter). The I/O devices (screen and keyboard) are also emulated.

The screen has 256 rows of 512 black or white pixels each. The screen is continuously refreshed from the screen memory map (RAM addresses 16,384 to 24,575): Writing values in any of these addresses results in rendering black or white pixels on the screen.

The keyboard: When the *enable keyboard* button is clicked, pressing a key on the physical keyboard results in two side effects: (i) the character code is written to the keyboard memory map (RAM address 24,576), and (ii) the character code is displayed in the keyboard visualization.

The CPU emulator has a builtin assembler: When loading an .asm file into the emulator, the emulator's software translates each symbolic instruction into its binary version. In the process, all the symbolic variables and labels are translated into physical addresses. The loaded code can be inspected in either its symbolic or binary version, using the display format menu of the ROM panel.

Loading machine language programs

There are two ways to load a program into the CPU emulator:

- From the CPU emulator: Use the 'Load file' icon to load an .asm or a .hack file.
- From the Assembler: Use the 'Load to the CPU Emulator' icon in the 'Binary code panel'.

Executing machine language programs

0. Assuming that a program was already loaded into the ROM:

1. If the program expects to process inputs, enter test values into the relevant RAM addresses.

2. To run the program's code, use the step/run/reset controls above the Screen panel.

3. If the program folder includes a test script, the test script will be loaded in the Test panel. To run the script's commands, use the step/run/reset controls in the Test panel.

The program's or script's execution speeds can be controlled by using the speed sliders.

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).

www.nand2tetris.org