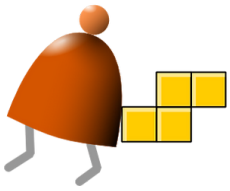
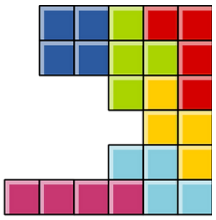
From Nand to Tetris

Building a Modern Computer From First Principles





- Home
- Projects
- Book
- Software
- Demos
- License
- Papers
- Cool Stuff
- Team
- Stay in Touch
- Q&A

Software

The Nand to Tetris Software Suite contains all the tools and files necessary for completing all the projects described in this site, and in the book *The Elements of Computing Systems*. Once you download the software suite to your PC, there is no need to download anything else throughout your Nand to Tetris learning experience.

The software runs as is on Windows, Unix, and Mac OS.

The software can be used freely under the terms of the GNU GPL (General Public License).
The software is open source. If you wish to inspect, modify, or extend the software, see the bottom of this page.

Download

(Note: after we update the software, which happens now and then, there is a limited time period in which some browsers may warn that "this file is not frequently downloaded and can be dangerous", or something like this. Please ignore this warning, and proceed to download and extract the nand2tetris.zip file to your computer.)

Download the Nand2tetris Software Suite Version 2.6 (about 730K).

Mac users: after downloading, read this Setup Guide for Apple MacOS, written by Yong Bakos.

Windows users: after downloading, put the downloaded zip file in an empty directory on your computer, and extract its contents as is, without changing the directories structure and names.

In order to use the nand2tetris software tools, your computer must be equipped with a Java Run-time Environment. The JRE can be downloaded freely from many sites including this one. For best performance, download the latest available version.

Problems? Describe your problem in our Q&A Forum and you will get help soon

About the Software

The Nand2tetris Software Suite consists of two directories: *projects*, and *tools*.

The projects directory is divided into 14 project directories named 00, 01, ..., 13 (of which project 00 is relevant only to learners who take the course in Coursera, and project 13 is open-ended). These directories contain files that you have to modify and complete as you work on various nand2tetris projects.

The tools directory contains the nand2tetris software tools. It's a collection of programs and files that will be explained as you follow the various projects.

The remainder of this section should be used as reference; there is no need to read what follows until you will be asked to use a particular software tool.

The .bat and .sh files are batch and script files, used to invoke the nand2tetris software tools. These files are explained in detail below.

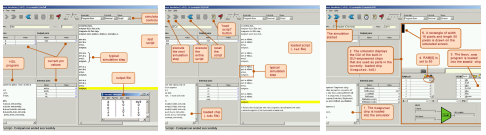
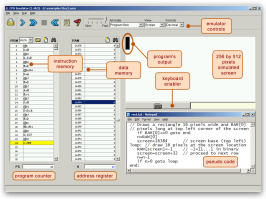
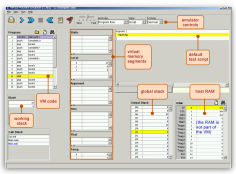
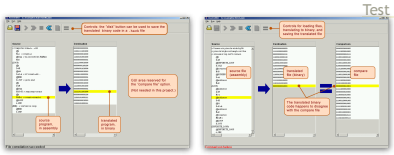
The bin directory contains the code of the nand2tetris software tools. It consists of several subdirectories containing Java class files and supporting files.

The builtInChips and the builtInVMCode directories contain files that are used by the supplied Hardware Simulator and VM Emulator, respectively.

The OS directory contains a compiled version of the Jack operating system.

Software Tools

The Nand to tetris Software Suite features the following software tools:

Tool	Description	Test Scripts
Hardware Simulator	Simulates and tests logic gates and chips implemented in the HDL (Hardware Description Language) described in the book. Used in hardware construction projects. Hardware Simulator Tutorial:	 Simulating a Xor gate — Running a test script — Simulating the topmost Computer chip
CPU Emulator	Emulates the operation of the Hack computer system. Used to test and run programs written in the Hack machine language, in both its binary and assembly versions. CPU Emulator Tutorial:	 Running a machine language program that draws a rectangle on the computer screen
VM Emulator	Emulates the operation of our virtual machine (similar to Java's JVM); used to run and test programs written in the VM language (similar to Java's Bytcode). VM Emulator Tutorial:	 Running a VM program
Assembler	Translates programs from the Hack assembly language to Hack binary code. The resulting code can be executed directly on the Computer chip (in the hardware simulator), or emulated on the supplied CPU Emulator (much faster and more convenient). Assembler Tutorial:	 Translating — Using a compare file
Compiler	Translates programs written in Jack (a simple, Java-like object-based language) into VM code. The resulting code can run on the supplied VM	(A GUI-less, command-level program)

	Emulator. Alternatively, the VM code can be translated further by the supplied VM translatorinto Hack assembly code that can then be executed on the supplied CPU Emulator.	

Tool	Description	Test Scripts
Operating system	Two OS implementations are supplied: (i) a collection of eight .vm class files, written originally in Jack (just like Unix is written in C), and (ii) a faster implementation of all the OS services, embedded in the supplied VM Emulator.	(GUI-less)

Tool	Description	Test Scripts
Text Comparer	This utility checks if two input text files are identical, up to white space differences. Used in various projects. In Unix use "diff" instead.	(A GUI-less, command-level program)

Running the Software Tools

The supplied software tools are designed to be run from your computer's command-line environment (also known as "terminal", or "shell"). Command-line environments vary from one operating system to another, and working in them requires some knowledge of various OS shell commands.

In order to eliminate this overhead, we supply batch files (for Windows) and scripts (for Unix and Mac OS), developed by Mark Armbrust. These batch and script files enable invoking the supplied nand2tetris tools from the command line on your computer, painlessly. They can be used from any working directory on your computer, without requiring full paths to the files on which they operate. Further, they accept spaces in directory and file names, so they will work if nand2tetris is installed under a directory named, say, "My Documents".

Mac and Linux users:

Before running the scripts, you must first change their file attributes to include "executable". You can then run the scripts by typing their name, as well as the .sh extension, in the terminal environment.

If you want to avoid typing the 'sh' extensions, you can create (once and for all) symbolic links in your ~/bin directory. Here is an example how to do it for, say, the HardwareSimulator tool:

```
ln -s ~/nand2tetris/tools/HardwareSimulator.sh HardwareSimulator
chmod +x HardwareSimulator
```

Windows users:

For the batch files to work from the command line, you must add (once and for all) the nand2tetris/toolsdirectory to your PATH variable.

To run a batch file from command-line, type its name, without the .bat extension.

If you use Windows 7 64-bit you need to install the 64-bit version of Java so that 64-bit cmdexe can run Java commands in batch files. If you get the output "'java' is not recognized..." you likely only have the 32-bit Java installed on your computer.

You can create desktop icons and use them to invoke the interactive versions of the following supplied tools: HardwareSimulator, Assembler, CPUEmulator and VMEmulator. This can be done by finding the disk locations of the respective batch files, right-clicking on them and picking "Send to > Desktop." Edit the shortcuts' properties and set "Run" to "minimized."

Usage

Hardware Simulator: To invoke the hardware simulator in interactive mode, type "HardwareSimulator" in the command line. For example:

C:\...\projects\02>HardwareSimulator
(a window will open up, running the interactive version of the Hardware Simulator)

To invoke the hardware simulator in batch (shell/cmd) mode, type "HardwareSimulator" in the command line. For example:

C:\...\projects\02>HardwareSimulator ALU.tst
(invokes the simulator, loads the given test script, executes it, and reports the result). Note that the simulator's interactive mode also enables loading and executing test scripts.

Successful test (example):

C:\...\projects\02>HardwareSimulator ALU.tst
End of script - Comparison ended successfully

Failed test (example):

C:\...\projects\02>HardwareSimulator ALU.tst
Comparison failure at line 24

Error in the associated HDL file:

C:\...\projects\02>HardwareSimulator ALU.tst
In HDL file C:\...\projects\02\ALU.hdl, Line 60, out[16]: the specified sub bus is not in the bus range: load ALU.hdl

CPU Emulator and VM Emulator: These operation of these tools follow the same convention described above. If you invokve either tool without a parameter, the tool will work in interactive mode; if you supply a parameter (test script), the tool will run batch-style.

Assembler: Typing "Assembler" will start the supplied assembler in interactive mode. Typing "Assembler xxx.asm" will assemble the specified xxx.asm file and generate a file named xxx.hack, containing the translated binary code. Note that the assembler's interactive mode also enables loading and translating .asm files.

Successful assembly (example):

C:\...\projects\04\fill>Assembler Fill.asm Assembling "c:\...\projects\04\fill\Fill.asm"

Failed assembly (example):

C:\...\projects\04\fill>Assembler Fill.asm Assembling "C:\...\projects\04\fill\Fill.asm" In line 15, Expression expected

To compare the resulting .hack code file to some expected .hack file, use the supplied TextComparer tool, described below.

Compiler: TTyping "JackCompiler fileName.jack" will compile the supplied Jack file. Typing "JackCompiler directoryName" will compile all the Jack file that are found in the specified directory. Wildcards are not supported. Here are some examples:

Compile the current directory:

C:\...\projects\09\Reflect>JackCompiler
Compiling "c:\...\projects\09\Reflect"

Compile a single file:

C:\...\projects\09\Reflect>JackCompiler Mirrors.jack Compiling "C:\...\projects\09\Reflect\Mirrors.jack"

Compile the "Reflect" directory (for example):

```
C:\...\projects\09>JackCompiler Reflect
Compiling "C:\...\projects\09\Reflect"
```

TextComparer: Compares two given files ignoring white space, and reports success or failure. For example, suppose you run the hardware simulator with some test script and get a comparison failure. If you want, you can then use the TextComparer to investigate the problem:

```
C:\...\projects\02>HardwareSimulator ALU.tst
Comparison failure at line 24

C:\...\projects\02>TextComparer ALU.cmp ALU.out
Comparison failure in line 23:
|0101101110100000|0001111011010010|1|1|0|0|0|0|0001111011010010|0|0|
|0101101110100000|0001111011010010|1|1|0|0|0|0|0001111011010010|0|1|
```

(Note the line number discrepancy between the reports of the two tools).

Help: In Windows, each batch file accepts a "/?" option that shows its intended usage. In Mac and Unix, use "-h". For example:

```
C:\...\projects\09>JackCompiler /?
Usage:
    JackCompiler Compiles all the .jack files in the current working directory.
    JackCompiler directoryName Compiles all the .jack files in the specified directory.
    JackCompiler fikeName.jack Compiles the specified Jack file
```

Source Code

All the nand2tetris software tools are written in Java. If you wish to inspect, modify, or extend some tool, you can download the source code. Before compiling the source code on your computer, read Readme.txt.
For details on what's new in the current version of the software (somewhat technical but useful for porting old modifications to the current version), read this ChangeLog.txt file.
If you wish to share your software extensions with others, please email us at nand2tetris@gmail.com.