Nand2Tetris Software

In order to complete the Nand to Tetris projects, you will need various software tools. These tools come in two versions:

Nand2Tetris IDE online: This web-based version features a Hardware Simulator, a CPU emulator, and a Hack Assembler – everything you need for completing projects 1-6.

Nand2Tetris Software Suite: This desktop version features all the tools mentioned above, plus the tools needed for completing projects 7-12.

What to do: Download the software suite, even if you plan on using the online version. The software suite includes files that are mentioned in various Nand to Tetris projects, and it's important to have these files handy on your PC.

The Nand to Tetris Software Suite

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. It can be used freely under the terms of the GNU GPL license.

1. 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 2.7 (about 1MB).

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

Windows users: Extract (unzip) the nand2tetris.zip file to your computer. Don't change the folders structure and folder/file 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.

2. About the Software

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

The projects folder is divided into 14 project folders named 00, 01, ..., 13. These folders contain files that you have to modify and complete as you work on various nand2tetris projects.

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

The .bat and .sh files are batch and script files, used to invoke the nand2tetris software tools. These files are used on Mac OS and on Windows, respectively.

The bin folder contains the code of the nand2tetris software tools. It consists of several sub-folders containing Java class files and supporting files.

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

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

3. Using the Software Suite

The supplied software tools are designed to be run from your computer's command-line environment (also known as "terminal", or "shell"). To make things simpler, 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 folder on your computer, without requiring full paths to the files on which they operate. Further, they accept spaces in folder and file names.

Mac and Linux users: Before running the scripts, change (once and for all) 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 folder. 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/tools folder 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 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."

4. Specific Usage Guidelines

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 *fileName*" in the command line. For example:

C:\...\projects\02>HardwareSimulator ALU.tst

This command invokes the simulator, loads the given test script, executes it, and reports results. 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: The operations of these tools follow the same conventions described above. If you invoke either tool without a filename (test script), the tool will work in interactive mode; if you supply a filename, the tool will run batch-style.

Assembler: Typing "Assembler" will start the Hack 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 TextComparer tool, described below.

Compiler: Typing "JackCompiler fileName.jack" will compile the supplied Jack file. Typing "JackCompiler folderName" will compile all the Jack files in the specified folder. Wildcards are not supported. Examples:

Compile a single file:

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

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

Compile a folder named "reflect":

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. On 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 tools in the desktop Nand to Tetris Software 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.