Project 3: Memory

The computer's main memory, also called *Random Access Memory*, or RAM, is an addressable sequence of registers, each designed to hold an *n*-bit value. In this project you will gradually build a RAM unit. This involves two main issues: (i) using gate logic to store bits persistently, over time, and (ii) using gate logic to locate ("address") the memory register on which we wish to operate.

Objective

Build the following chips:

- DFF (given)
- Bit
- Register
- RAM8
- RAM64
- RAM512
- RAM4K
- RAM16K
- PC

In principle, the DFF logic can be realized using Nand gates. However, we consider DFF primitive, and thus there is no need to implement it.

Files: For each chip Xxx in the list, we provide a skeletal Xxx.hdl program, also called *stub file*, with a missing PARTS section. In addition, for each chip we provide an Xxx.tst script that tells the hardware simulator how to test the chip, along with an Xxx.cmp compare file containing the correct outputs that the supplied test is expected to generate. Your task is writing, and testing, the chip implementations (specifically: Completing the supplied Xxx.hdl files).

Contract: For each chip in the list, your chip implementation (modified Xxx.hdl file), tested by the supplied Xxx.tst file, must generate the outputs listed in the supplied Xxx.cmp file. If the actual outputs generated by your chip disagree with the desired outputs, the simulator will report error messages.

Building the chips

A new online IDE (Integrated Development Environment) was recently launched for learners of Nand to Tetris courses. Therefore, there are now two options for completing project 3:

If you are using the Nand2Tetris IDE Online (which is recommended), all the Xxx.hdl, Xxx.tst and Xxx.cmp files are available in your browser memory. To develop and test a particular chip, select the project / chip from the simulator's drop-down menus. Your edited HDL code will be saved automatically. To download the HDL files to your local PC, click the *download* button. The current version of all the project's Xxx.hdl files will be downloaded as one zip file.

Using the desktop Nand2Tetris hardware simulator is also possible. If you've downloaded the Nand to Tetris software suite from www.nand2tetris.org, and extracted it into a folder named nand2tetris on your computer, the nand2tetris/tools folder contains the desktop version of the hardware simulator, and the folders nand2tetris/projects/3/a and nand2tetris/projects/3/b contain all the files needed for completing this project. The two subfolders a and b are needed for technical reasons. Leave this folders structure as is, and don't move files from one subfolder to another.

References

HDL Guide

Chips Set API

Tutorials

The tutorials below focus on using the desktop version of the hardware simulator. Tutorials for the online simulator, the preferred tool for this project, will be available soon. However, you can apply the principles from these tutorials to perform similar actions in the online simulator (a major difference is that there is no need to load any files in the online simulator).

Clock Demo

Register Demo

RAM Demo

Program Counter Demo

Consult each reference / tutorial as needed: There is no need to go through the entire resource.

Implementation Tips

All the *Implementation Tips* of project 1 apply to this project also, so read them now.

There is only one modification: When implementing the chips of project 3, you can use, as chip-parts, any of the chips listed in projects 1, 2, and 3.

If you are using the desktop simulator, you will notice that nand2tetris/projects/3 consists of two subfolders named a and b. These subfolders are needed for technical reasons. Leave the structure of nand2tetris/projects/3 as is, and don't move files from one subfolder to another.

Submission guideline

If your course instructions guide you to submit a zip file that includes your project 3 chips, this zip file must have the following structure (next page):

```
project3.zip
    └── a
        ├── Bit.hdl
        ├── PC.hdl
        ├── RAM64.hdl
        ├── RAM8.hdl
        └── Register.hdl
    └── b
        ├── RAM16K.hdl
        ├── RAM4K.hdl
        └── RAM512.hdl
```

The a and b sub-folders are needed for technical reasons, related to auto-grading issues. To avoid grading penalties, make sure that your zip file has the structure shown above. This special guideline is relevant to project 3 only.