Workshop 01 - Introduction

The aim of this workshop is to familiarise you with some of the basic tools that will assist your learning in the course and introduce you to the HDL language. This is based on project 01 from the Nand2Tetris course.

Preparing for this Workshop

Reading

Make sure you've read the first chapter of the textbook. Then read sections I, II, and III of the tutorial on the hardware simulator and the Hardware Description Language (HDL). These can currently be found at the end of the Software page of the nand2Tetris website:

https://www.nand2tetris.org/software (https://www.nand2tetris.org/software)

Note that page 26 contains some important notes about how the simulator searches for hardware definitions.

Important: you must use our version of the nand2tetris tools which can be found on our Nand2Tetris Resources (https://myuni.adelaide.edu.au/courses/101158/pages/nand2tetris-resources-2) page.

Pencil and Paper

You will find it much easier to understand what is happening if you bring some paper and something to write with. Drawing pictures of how gates fit together and truth tables of what inputs produce what outputs can be very helpful.

Setting up your Environment

If you're using a Linux machine in one of the labs, everything is installed and ready-to-go, otherwise you'll need to setup the tools for this course. Follow the steps on the Nand2Tetris Resources (https://myuni.adelaide.edu.au/courses/101158/pages/nand2tetris-resources-2) page ensure you have everything installed.
Note: you must use our version of the nand2tetris tools, found on the page above.

Let's make sure everything works.

1. Download the files for Practical Assignment 1 (https://myuni.adelaide.edu.au/courses/101158/files/18016620?wrap=1) (https://myuni.adelaide.edu.au/courses/101158/files/18016620/download?download_frd=1) if you haven't already and unzip them.

2. Open a terminal and navigate to the directory containing the files from [1]

3. Use the CLI Hardware Simulator and the test script `And.tst` to test the *And* chip:

   ```
   HardwareSimulator.sh And.tst
   ```

   - This runs the test script on the simulator, produces an output file, `And.out` and compares it to a file of expected output, `And.cmp`.
   - Does the And chip behave as expected?
   - Review the output file, `And.out` and check how this output compares to what you expect.
   - **Note:** If you're not using the lab PCs, then you'll need to supply the full path to `HarwareSimulator.sh`

4. Now try with the GUI Hardware Simulator:
   Again, make sure your terminal is in the directory containing the files from [1]

   ```
   HardwareSimulator.sh
   ```

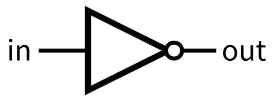
   - This should open a user interface that you can use to open and test your HDL files.
   - Loading the `And.tst` script, and running it using the Run » button.
   - Use the view option to see the output and compare values.
   - **Note:** Again, if you're not using the lab PCs, then you'll need to supply the full path to `HarwareSimulator.sh`

Ask your workshop supervisor if you get stuck.

✓ Building your first Chips

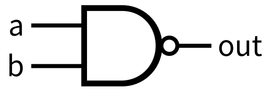
We'll start by implementing the Not Chip:

INPUT	OUTPUT
in	out
0	1
1	0



To do so we'll use a single Nand Chip (our most basic building block in this course):

INPUT		OUTPUT
a	b	out
0	0	1
0	1	1
1	0	1
1	1	0



How can we connect the pins of the Nand chip to act in the same way as a Not chip?

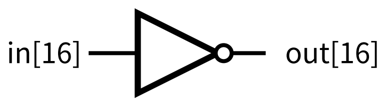
1. Edit the `Not.hdl` file in your preferred editor and attempt to implement the *Not* chip using a *Nand* chip. Test your *Not* chip.
   A .hdl file is just a text file and be edited in exactly the same way that you edited your .cpp and .h files in earlier programming courses.

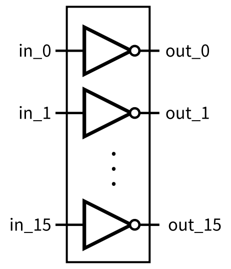
2. Edit the `And.hdl` file in your preferred editor and attempt to implement the *And* chip using a *Nand* chip and your *Not* chip. Test your *And* chip.

3. Open Gradescope using the link in Practical Assignment 1 and submit your hdl files.

✓ Building your first bus Chips

Bus chips behave like regular chips, except the input and output pins are several bits wide; we represent them as a single pin to simplify our logic:





1. Edit the `Not16.hdl` file in your preferred editor and attempt to implement the 16-bit *Not* chip using multiple *Not* chips. Test your chip.

2. Try the same for your `And16.hdl` file.

3. Open Gradescope using the link in Practical Assignment 1 and submit the hdl files.

ℹ Q&A Time

If you have any further questions on content from Weeks 1, 2 or assignments, clarify them in this workshop.

Prac 1

If you have time left, you can use the remainder of this session to work on your assignments for this week.