Computer System Design Lab
Design Experiment 1

Indian Institute of Technology Tirupati

August 27, 2020

1 The Problem Statement:

To design the basic building blocks of digital system using hardware description language (HDL). The hierarchical design approach need to be adopted while designing the large block. NAND gate is considered to be the primitive gate from which rest of the gates would be described using the HDL. Each design has to be verified by simulation using the Simulator.

In this exercise the following logic gates are to be designed.

Chip name	Functionality
Not	An inverter gate with function $y = \bar{x}$.
And	A two input and single output AND gate.
Or	Two input and single output logical OR gate.
Xor	Two input and single output logical XOR gate.
Mux	Three input (two data input and one select input), one output multiplexor logic.
DMux	Two input, two output de-multiplexor logic.
Not8	8input and 8 output NOT array. (Reuse the Not).
Not16	16 input and 16 output NOT array. (Reuse the Not8).
And8	16 input and 8 output AND array. (Reuse the And).
And16	32 input and 16 output AND array. (Reuse the And8).
Or8	16 input and 8 output OR array. (Reuse the Or).
Or16	32 input and 16 output OR array. (Reuse the Or8).
Mux16	32 + 1 inputs and 16 outputs. (Reuse Mux).
Or8Way	8 input and one output. (Reuse the 2 input Or in hierarchical fashion).
Mux4Way16	(16 * 4) + 2 inputs and 16 outputs. (Reuse Mux16 in parallel).
Mux8Way16	(16 * 8) + 3 inputs and 16 outputs. (Reuse Mux16 as needed).
DMux4Way	1 + 2 inputs and 4 outputs. (Reuse DMux to design this).
DMux8Way	1 + 3 inputs and 8 outputs. (Reuse DMux4Way to design).

2 The Experimental Flow:

1. Design specification: to decide on what are the valid inputs and correct outputs for the given design.

   At first, you need to decide on the design specification. Which can be done, for the small design, in a truth table. For every design asked as part of this exercise, you need to specify the truth table.

2. Design description: to describe (code) the design using HDL.

   As part of this design exercise, we will be using the HDL language to describe the given designs. The HDL program could be edited in any of the editor of choice (gvim, emacs, atom etc) and then be read in to the *nand2tetris* synthesis tool for compilation. Refer to the tool user manual for "How to use".

3. Synthesis: to compile the design which would check for syntax error.

   In this stage you will be performing compilation of the design which you have described in HDL and make it error free to go to the next stage for simulation.

4. Design Verification: to simulate the design by writing a test bench in TSL (test script language).

   Once the design compilation is completed, the next task is to perform simulation to verify the correctness of the design. For any design to verify by simulation a test bench need to be written. The test bench can be written in TSL script, a manual for TSL script is provided in Appendix B of the text book.

5. Library Preparation: to keep the verified design in an appropriate location where the design could be reused in other larger design.

   Once the design verification is done the design could be placed in a specific directory such that it can be reused in further design.

3 Tools:

- Language: The Nand2Tetris HDL and TSL (test scripting language)
  Refer: Appendix A and B of text book.
- Tools: Hardware Simulator of Nand2Tetris.
  `https://www.nand2tetris.org/software`
- Machine and OS: x86_64 machines with any distribution of Linux (Ubuntu or CentOS).

4 Reporting and Evaluation

All designs will be evaluated as per the criteria of evaluation which is primarily based on correctness and elegance of the design. Elegance here means how well the concept of design reuse, naming of the chip and interconnects, and commenting on the function of each line has been done. And, most importantly how well your design is optimized in terms of length of critical path [1] and total number of basic gates.

A report need to be prepared for documentation of the work. A final report, at the end of all experiments which describe the design of computer system from basic to system level, will be evaluated during your final test.

[1] Critical path is defined as a longest combinational path between any two input and output either from primary side or from flip-flop.