Computer System Design Lab
Design Experiment 2: ALU and Its Component Design

Indian Institute of Technology Tirupati

August 26, 2019

1 The Problem Statement:

The final objective of this design experiment is to come up with a working ALU. This experiment has to make use of the gates and components that have been designed in Design Exp 1. Each design has to be tested through simulation tool by writing test script. It is important to select the appropriate test inputs such that the important features of the design are throughly tested to be correct.

Chip name	Functionality
Half Adder	A two bit adder.
Full Adder	A three bit adder.
16 bit Adder	A 16 bit two number adder (ripple carry) with 16 bit output.
32 bit Adder	A 32 bit two number adder (ripple carry) with 32 bit output.
16 bit Adder/Sub	A 16 bit two number adder/subtracter with 16 bit output.
16 bit Multiplier	a 16 bit two number multiplication with 16 bit output.
16 bit ALU	a 16 bit Arithmetic and Logic Unit with specification as described followed by.
16 bit Incrementer	A 16 bit incrementer with 16 bit input and 16 bit output.

Design Specification:

16 bit ALU: The ALU need to execute the following functionalities.
0, 1, -1, x, y, $\bar{x}$, $\bar{y}$, $-x$, $-y$, $x+1$, $y+1$, $x-1$, $y-1$, $x+y$, $x-y$, $y-x$, $x\&y$, $x \mid y$, $\overline{x\&y}$, $\overline{x \mid y}$, $x \oplus y$, $\overline{x \oplus y}$, $x < y$, $x = y$, $x > y$.

Note that in the above list of functionalities the multiplication has not been included. The ALU need to have two data input and few number of control inputs. If you follow the design that has been discussed in the class, it will have 6 control inputs. The ALU have one data output and additional status output for comparator. The comparator has not been discussed in the class, this you has to be additionally designed by you.

The ALU that has been discussed in the class is one of the possible designs. There could be much better and optimized design possible which you are encouraged to come up with.

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.

Design rules:

- Naming: The Chip name, Input and Output pins, proper commenting for statement are essential part of good coding. Each chip name (and file name) should be prefixed with your registration number such as CS16B01ALU32.
- Error Logging: It is good to maintain a log file each of the error that you encounter while compilation or simulation. It is a good practice to maintain proper log file with description of error and its fix.
- Report: Please prepare a document on various design decision that you take related to design optimization, simulation method etc.

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. The design which follows the Design Rule will be evaluated accordingly.

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.