Computer System Design Lab
Design Experiment 3b:
Memory Design

Indian Institute of Technology Tirupati

Sept 24, 2020

1 The Problem Statement:

Memory system is an integral part of stored-program based CPU system design. Efficient memory system design is the goal of every computer architects and designers for high-performance computer system. In this experiment we will be designing a set of memory modules which will be used to design computer system. All the memory are word-addressable ie each word is assigned a unique memory address where a word length is 16 bit. All the designs needs to be coded in BHDL and needs to be verified using test bench and simulator.

Chip name	Functionality
RAM8	A RAM system with 8×16 size.
RAM64	A RAM system with 64×16 size.
RAM512	A RAM system with 512×16 size.
RAM4K	A RAM system with $4K \times 16$ size.
RAM16K	A RAM system with $16K \times 16$ size.
RAM32K	A RAM system with $32K \times 16$ size.
RAM64K	A RAM system with $64K \times 16$ size.

Design Specification:

- In all the above memory modules RAMn means a random access memory with $n \times 16$ size where n indicates the number of memory locations and 16 indicate that each location is of size 16 bit. In such memory system the total number of address line (address bus) would be $\lceil log_2 n \rceil$ and the size of data bus would be 16. The K here can be interpreted as 2^{10}.
- Each memory module must be designed by reusing the previous modules accordingly which must ensure the hierarchical addressing of each location. For example: RAM64 could be designed using eight RAM8 modules, where, to enable a particular locations the addressing need to designed in such that three lines to be used to enable one among the eight RAM8 module and then the remaining lines be used to enable the desired location within that module.
- The computer system that will be designed will have two separate memories: Instruction memory and Data memory.

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. Along with the TSL script, a comparison file in .cmp format for each design needs to be prepared as per the format of .cmp.

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 cs16b01RAM8.
- 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: This is the same online document where all the previous experiments have been reported.

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.

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

As done in the previous experiments, kindly report whatever has been done in this experiment in the shared document file.