EE275 **Spring 2017**

San Jose State University
Department of Electrical Engineering

Mini-Project III (Due Apr. 26)

Mini-project III is to be carried out individually. In this mini-project, you are to design a single-level cache simulator (assume von Neuman architecture) as discussed in class. The "instruction cycle-accurate" simulator may be written in HDL (i.e., Verilog or VHDL) or SDL (i.e., C/C++ or Python).

The simulator should accept the following cache design parameters and simulate the working of the benchmark programs in the mini-project 2:

- Main memory size (in bytes)
- Cache size (in bytes)
- Line size (in bytes)
- Cache placement algorithm (cache organization, i.e., direct mapped, set-associative, etc.)
- Set size (i.e., set associativity)
- Cache replacement algorithm (i.e., LRU, FIFO, random)
- Write policy (i.e., write-back, write-through, or write-once)

It is suggested the size of the main memory, and thus the corresponding cache memory, be set very small, just enough to hold the benchmark program.

Your cache simulator should show the cache memory (tag and data among others) contents at each instruction execution cycle.

NOTE: To show the authenticity of your work, attach the last two digits of your student ID to every signal and variable name used in your Verilog/ VHDL/C/C++ code. For example, if your SID is 123456789, signal names should be x89, a89, etc. Each of the screen captures of your code and simulation results should bear these signal names.

C.Choo (4/10/2017)