CSCE 692 HW Lab 1 - Cacti Cache Simulator

2nd Lt David Crow – 6 January 2019

(100 Points)

Due: NLT 0800 Thursday, 10 January 2019

**Lab Overview:**

Compile CACTI 6.5 on a Linux machine. Use the resulting program, together with the configuration file, to make comparisons of cache performance under various circumstances.

**Guidance:**

* Write your name and a page number on each page
* Show your work
* Clearly indicate your answers
* Explain assumptions and provide references (sources) for any additional information you had to research in order to complete any problem.
* CACTI 6.5 is available in an online form at http://www.hpl.hp.com/research/cacti/ with the tar file on CANVAS and placed at L:\Courses\CSCE\CSCE692\Project\Lab 1
* Remember to use 1024 for KB in CACTI
* For direct mapped, (1-way), input 1 in the associativity box
* You are looking for only two CACTI output items:
  + Access time (ns)
  + Cycle time (ns) – used as the assumed CPU cycle time

**Assignment tasks:**

1. Compile CACTI 6.5 (or run provided binary and determine it works on your machine).
2. Compare access times in Lab1\_initial\_configuration\_input.cfg as you change “associativity.”
   1. Access/cycle times for associativity 1 0.301390 ns 0.183927 ns
   2. Access/cycle times for associativity 4 0.344226 ns 0.240763 ns
   3. Access/cycle times for associativity 8 0.561563 ns 0.149119 ns
   4. Access/cycle times for associativity 16 0.799303 ns 0.159446 ns
3. Compare access times across cache sizes (return to associativity = 1).
   1. Access/cycle times for 32 KB cache 0.387672 ns 0.183927 ns
   2. Access/cycle times for 128 KB cache 0.731590 ns 0.183927 ns
   3. Access/cycle times for 1 MB cache 1.705250 ns 0.127245 ns
4. Choose two different and interesting parameters to vary in the cache configuration input file.
   1. Observe and record the output changes.

If I change the from , which parallelizes data and tag access, to , which serializes data and tag access, the total access time increases from to . The cycle time changes from to .

If I change from to , the total access time decreases from to . The cycle time increases from to .

* 1. Offer an explanation as to why the varying input would cause the varying output.

It seems evident that serializing a parallel process slows a program’s execution. Here, we serialize our data access, so our access time necessarily increases. Our cycle time marginally improves, though; this is probably because the overhead for serial accesses is slightly less than the overhead for parallel accesses.

One should expect that a larger data bus, which allows more data to traverse said bus at a given time, allows for faster data access. As shown by the simulation, doubling the bus’s width certainly allows for faster access times. However, it also slows the cycle time. Decreased cache performance (more cache misses, for example) could be a possible factor for this.