Team: Loren Lugosch, Harsh Aurora

Title: Multi-level Cache Coherency Circuit

One way to make memory accesses faster on average without using a larger cache or faster main memory is to use a multi-level cache [1]. Multiprocessors with shared caches need to perform cache coherence checks to ensure that shared memory accesses are correct. We propose to design and compare cache coherence checking circuits in VHDL for a multiprocessor with a multi-level cache. The memory system will consist of two split L1 caches, one common L2 cache, and a main memory. We will emulate processors in simulation using test signal assignments in our testbench. If we have time, we will replace the emulated processors with two of Altera's Nios II IP cores and write test software for those cores with shared memory accesses.

Our project will investigate the MESI, MSI and Dragon cache coherence protocols [2]. We will build all three circuits and evaluate their performance by comparing 1) best-case, worst-case and average-case memory access times and 2) number of coherence misses for a given set of accesses.

References:

[1] Hennessy, John L., and David A. Patterson. Computer architecture: a quantitative approach. Elsevier, 2012.

[2] Tuan Bui, Brian Greskamp, I-Ju Liao, Mike Tucknott , “Measuring and Characterizing Cache Coherence Traffic” CS433, Spring 2004