**cache.vhd:**

Input/Output:

Cache takes in clk, ovrwr, addr, LRU, din and we signals. It outputs new\_LRU, dout, miss and dirty signals. Clk is used to synchronize the we of the cache memory to the low half of the clock cycle to give the address time to reach the memory. Ovrwr is used to tell the cache that the allocate state is currently overwriting a block and that it does not matter if the tag of the block matches. Addr is the 26 bit tag and index of the block we are interested in. LRU is 1 bit representing whether or not the top line or the bottom line was the least recently used. Din is the data that would be written to the block at addr if we is high. New\_LRU is 1 bit representing what line will be the least recently used after the current operation finishes. Dout is the data read of from the block at addr. Miss and dirty are 1 bit signals indicating whether addr misses or whether the hit block is dirty, respectively.

Function:

Cache maintains two csram instances with index width 5 and bit width 535. One each for the top and bottom lines of each index location. Cache takes addr and din and generates would be replacements for the top and bottom lines found at the index indicated by addr as if they were going to be written. The din to the top and bottom line csrams are determined by muxxes that choose either the top or the bottom as the would be writes and the other as a loopback. The csrams only write when clk is low, we is high, and there is a valid hit. On a read the output of the csrams is compared to the tag from addr using cmp\_n components. If neither match, miss is set high. If there is a match, miss is set low and the dirty bit is set based on the dirty bit of the matched block. If there is a valid hit, dout is set the matched block, otherwise dout is set the the block indicated by LRU.

**L1cache.vhd:**

Input/Output:

L1cache takes in clk, rst, en, cpuReq, cpuWr, cpuAddr, cpuDin, l2Din and l2Ready signals. It ouputs cpuDout, cpuReady, l2req, l2Wr, l2Addr, l2Dout, hit\_cnt, miss\_cnt and evict\_cnt signals. Clk, rst and enable are your typical control inputs. The input cpu signals get registered and are used in the state machine to inform the components as to what they are reading or writing and where. Cpu outputs are just the registered outputs from the cache. The l2 outputs are control for the lower level memory. The l2 inputs are the output of memory and its ready signal. (hit/miss/evict)\_cnt are signals that keep track of how many hits, misses and evictions there have been, respectively.

Function:

L1cache maintains NextState\_ctrl, RegWE\_ctrl and cache instances and has registers for the current state, the last state, cpuWr, cpuAddr, cpuDin, cpuReady, cpuDout, the block-to-be-replaced’s address, the block-to-be-replaced’s data, the LRU bit of each index location and (hit/miss/evict)\_cnt.

The write enables of each of these registers is determined in the RegWe\_ctrl component. The input of the current state register is determined by the NextState\_ctrl component. The input of the cpu registers is