Since we aren't actually working with data here these functions won't have normal functionality.

1. CheckForHit
   1. Dissects the current address
   2. Checks to see if hit/miss.
   3. If hit, it gets the selectedWay of the hit from the encoder
2. WriteL2Cache:
   1. Increment write counter
   2. hit
      1. write cacheData = “W”
   3. miss
      1. QueryLRU – gets LRU-way for replacement
      2. check mesi bits of selected way
         1. M:
            1. WriteSharedBus
            2. Then write cacheData = “W”
         2. E: write over, write cacheData = “W”
         3. S: write over, write cacheData = “W”
         4. I: write over, write cacheData = “W”
3. ReadL2Cache:
   1. Increment read counter
   2. WriteL1
4. QueryLRU: (only used for misses)
   1. Finds the LRU and outputs as selectedWay
5. UpdateLRU:
   1. updates LRU appropriately
6. readSharedBus
   1. should have options to just read or to read for ownership
   2. a read would put an “R” on the sharedOperationBusOut w/ address starting from beginning of cache line on sharedBusOut
   3. a RFO would put an “M” on the sharedOperationBusOut w/ address starting from beginning of cache line on sharedBusOut
7. WriteSharedBus
   1. writes address starting from beginning of cache line on sharedBusOut and “W” on sharedOperationBusOut
8. WriteL1
   1. If sharedOperationBusIn = “I”
      1. L1BusOut = sharedBusIn with first five bits as 0
   2. Else if L1OperationBusIn = “DR”
      1. L1BusOut = “L1DR”
   3. Else if L1OperationBusIn = “DW”
      1. L1BusOut = “L1DW”
   4. Else
      1. L1BusOut = “L1IR”
9. SendInvalidate
   1. sends an invalidate “I” on the sharedOperationBus w/ address on sharedBus
10. InvalidateL2
    1. Invalidates line at address in L2
    2. L1BusOut = address at beginning of line.
11. PutSnoopResult
    1. If hit
       1. M
          1. snoopBusOut = HITM
       2. E
          1. Command 3 – snoopBusOut = HIT
          2. Command 4 – snoopBusOut = HIT
          3. Command 5 – snoopBusOut = HIT
          4. Command 6 – snoopBusOut = HIT
       3. S
          1. Command 3 – snoopBusOut = HIT
          2. Command 4 – snoopBusOut = HIT
          3. Command 5 – snoopBusOut = HIT
          4. Command 6 – snoopBusOut = HIT
       4. I
          1. snoopBusOut = MISS;
    2. If miss
       1. snoopBusOut = MISS