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

1. checkL2
   1. Checks to see if hit/miss.
   2. This should cause the MESI bits to load in the MUX to be grabbed by the readL2 cache.
2. writeL1:
   1. if operation is “DR”: write “L1DR” to L1Bus
   2. else if operation is “DW”: write “L1DW” to L1Bus
   3. else write “L1IR” to L1Bus
3. readL2:
   1. should just grab MESI bits from the MUX because checkL2 should have caused the data to propagate through
4. updateLRU:
   1. updates LRU appropriately
5. updateMESI:
   1. updates MESI bits appropriately
6. readSharedBus
   1. should have options to just read or to read for ownership
   2. a read would put an “R” on the sharedOperationBus w/ address on sharedBus
   3. a RFO would put an “M” on the sharedOperationBus w/ address on sharedBus
7. writeSharedBus
   1. writes address on sharedBus and “W” on sharedOperationBus
8. sendInvalidate
   1. sends an invalidate “I” on the sharedOperationBus w/ address on sharedBus
9. invalidateL2
   1. invalidates line at address
10. PutSnoopResult
    1. Puts desired snoop result on the snoopBus