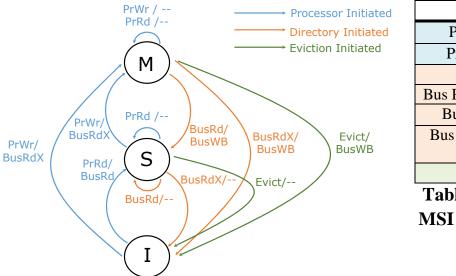# Quiz 3 Handout

Figures A-1 and A-2 show the state transition diagrams for the MSI and MOSI snoopy coherence protocols. Table A-1 lists the actions in these state transition diagrams.
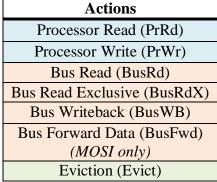
Actions
Processor Read (PrRd)
Processor Write (PrWr)
Bus Read (BusRd)
Bus Read Exclusive (BusRdX)
Bus Writeback (BusWB)
Bus Forward Data (BusFwd) *(MOSI only)*
Eviction (Evict)

Table A-1: Actions in the MSI and MOSI protocols.

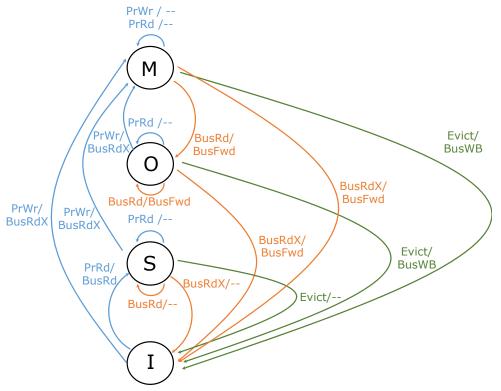**Figure A-1. MSI protocol state transition diagram.**

Figure A-2: MOSI protocol state transition diagram.

These protocols use writeback caches and rely on a shared, atomic bus.

In the MSI protocol, when a cache observes a `BusRd` or `BusRdX` request for a cache line in the `M` (modified) state, it responds with a `BusWB` (bus writeback), which includes the contents of the dirty cache line. Each `BusWB` serves two purposes: it updates main memory and forwards the line contents to the cache that initiated the `BusRd` or `BusRdX` request.

The MOSI protocol improves on MSI to reduce the number of writes to main memory. MOSI adds a new state, `O` (Owner), and a new bus action, `BusFwd` (bus data forward). When a cache observes a `BusRd` request for a cache line in the `M` state, it transitions to `O` instead of `S`. A cache line in `O` state has read-only permissions, like `S`. But `O` also has the responsibility to write back the line to main memory on an eviction, and to forward the line to other caches upon observing a `BusRd` or `BusRdX` transaction. This forwarding is done using `BusFwd`, which puts the data on the bus but *does not cause a write to main memory*. Using `O` and `BusFwd`, caches can exchange dirty data among them without updating memory (`BusWB` happens only on evictions).