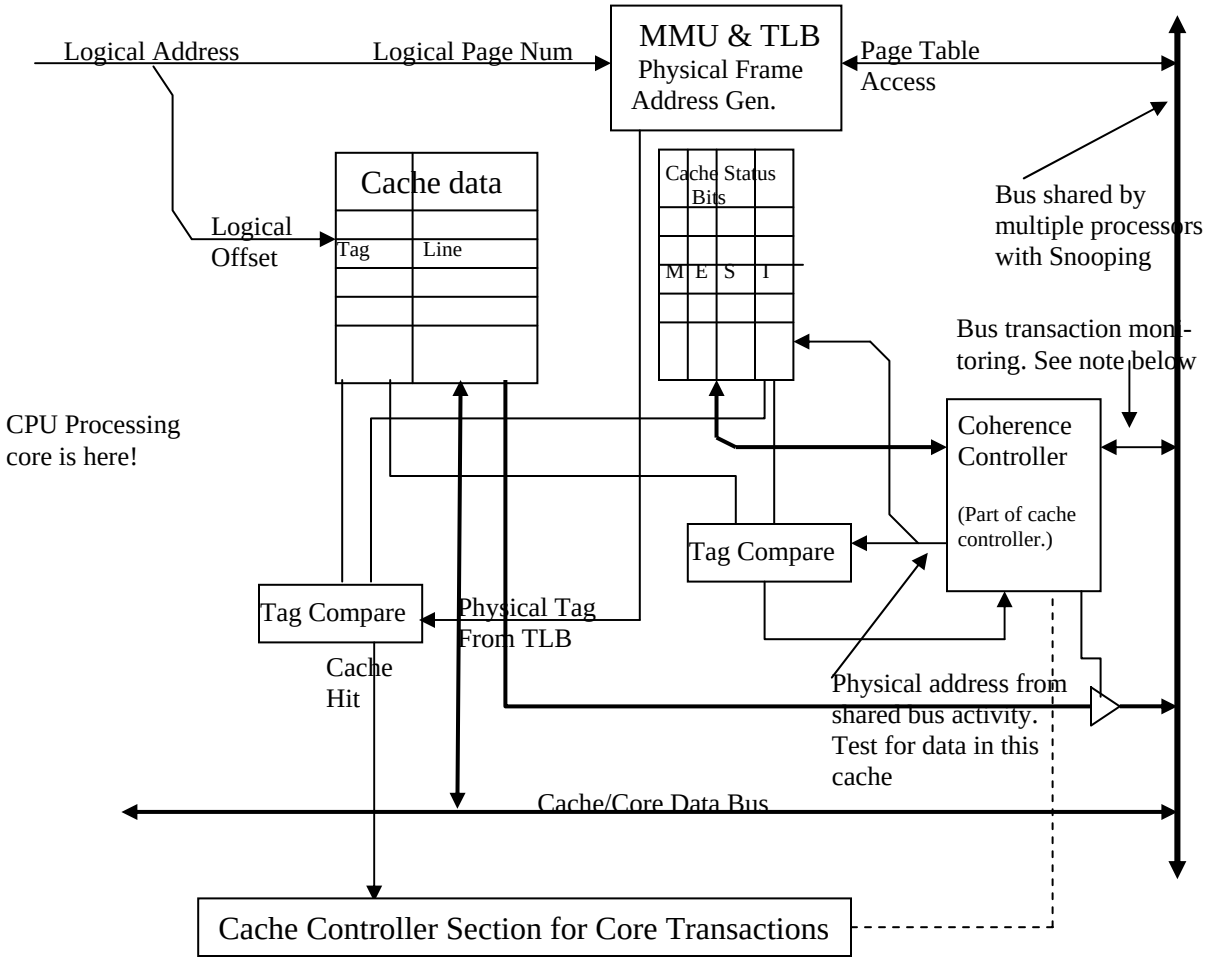

Additions to a Single-level Cache System to Support Snooping with the MESI Protocol: The cache status bits are augmented with bits for Exclusive-cleean and Shared status states. The cache tag and status fields are dual-ported for snooping. The cache line is dual-ported on read so that modified data may be returned to the bus when another processor needs the current value of that data. Using this second port, the "coherence controller" for this cache snoops on the shared bus activity and tests for a match between data addressed on the shared bus with data in the cache. The cache status is updated and transactions started on the bus to maintain coherence between data in the several caches and main memory.

NOTE: The link between the section of the cache controller that handles maintaining coherence and the shared bus of the SMP system only looks at the types of transactions on the bus and the addresses of the data on the bus. It never fetches data. The signal it returns is only whether this processor has data at that cache line address in its cache.

There are three types of data transfer on the shared bus of an SMP multiprocessor. These are initiated by either a processor or I/O unit and are monitored by all other processors. The response to them by other processors is shown in the state diagram for the MESI protocol shown below. The transaction types are:

- Bus Read (*BusRd*) – A processor read to cache has missed and the cache controller is asking for a copy simply for a read and not a write operation.
- Bus Read Exclusive (*BusRdX*) – A processor wants to write data that it does not have exclusive control of. The bus must return the current value of the data, perhaps from another cache that has modified the data since this processor last received it. All other copies of the data are marked invalid.
- Bus Write Back (*BusWB*) – A cache line is being evicted and the modified data is stored back to main memory.

For the MESI snooping protocol there is an additional control signal for the bus that indicates whether the data in a read transaction is present in another cache. In concept this is a wired NOR line driven by all processors with an open-drain driver and pulled up in the bus controller. That signal is called "S" in the state diagram of the MESI protocol below. The purpose of the signal is to eliminate unnecessary shared-bus transactions in fetching new data from memory or on writing cache data by eliminating the need to signal changes to data that is known to exist on only one processor.

If there is modified data in one processor cache when another processor does a BusRd cycle, then the processor with the modified data reads that data from its cache onto the bus and marks its own status as Shared. A processor with no data or shared data that it wishes to overwrite, initiates a BusRdX operation first and this invalidates any cached copies of the data on other processors.

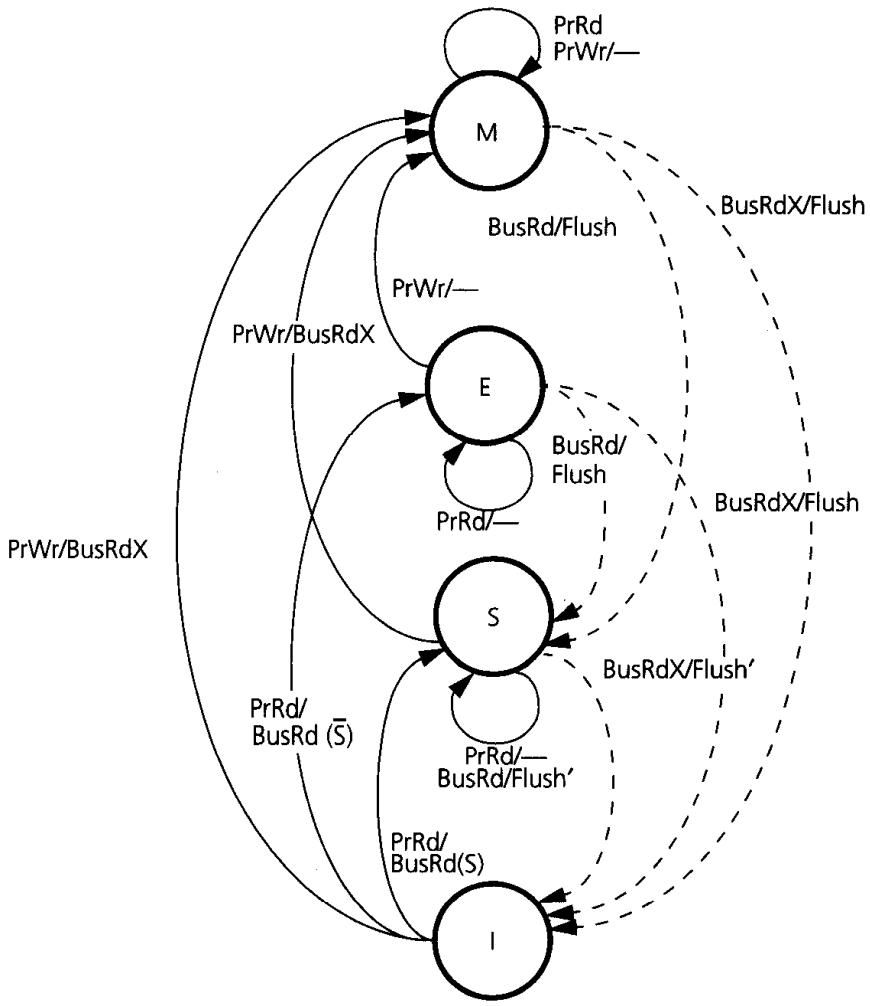

FIGURE 5.15 State transition diagram for the Illinois MESI protocol. MESI stands for the modified (dirty), exclusive, shared, and invalid states, respectively. The notation is the same as that in Figure 5.13. The E state helps reduce bus traffic for sequential programs where data is not shared. Whenever feasible, the Illinois version of the MESI protocol makes caches, rather than main memory, supply data for BusRd and BusRdX transactions. Since multiple processors may have a copy of the memory block in their cache, we need to select only one to supply the data on the bus. Flush′ is true only for that processor; the remaining processors take their usual action (invalidation or no action). In general, Flush′ in a state diagram indicates that the block is flushed only if cache-to-cache sharing is in use and then only by the cache that is responsible for supplying the data.