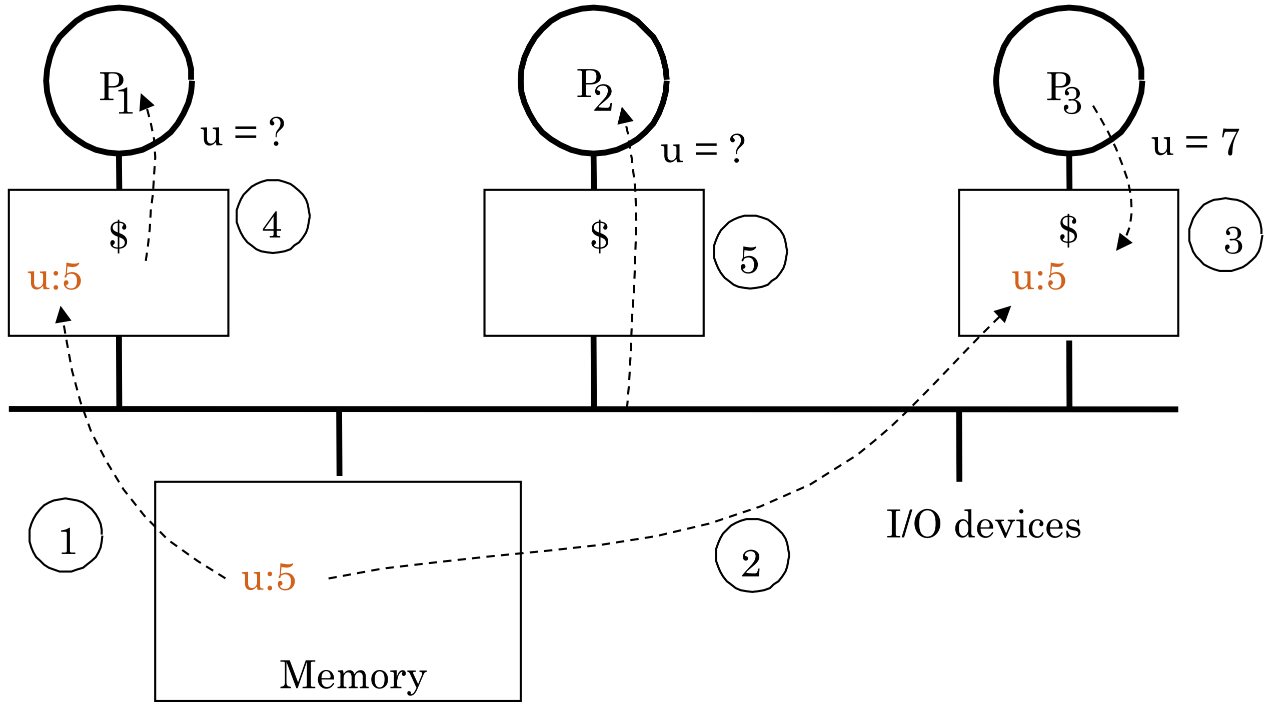
CACHE COHERENCE

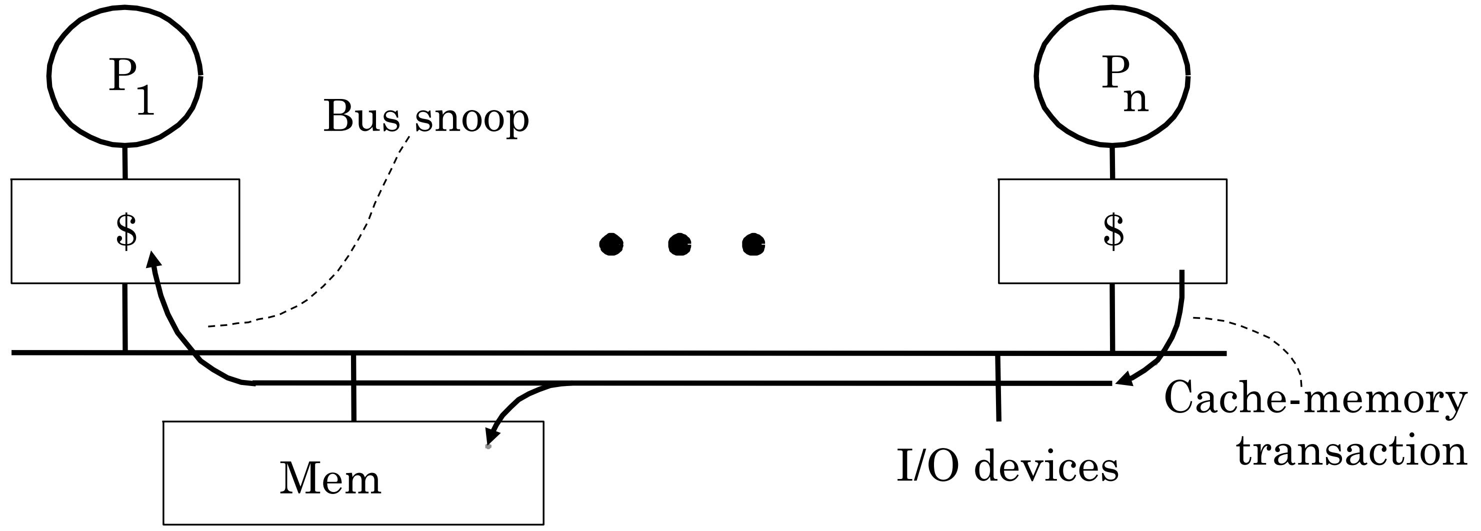
Dr. Madhu Mutyam
Computer Architecture and Systems Laboratory
Department of Computer Science and Engineering
Indian Institute of Technology, Madras
madhu@cse.iitm.ac.in
http://www.cse.iitm.ac.in/~madhu

The Cache Coherence Problem

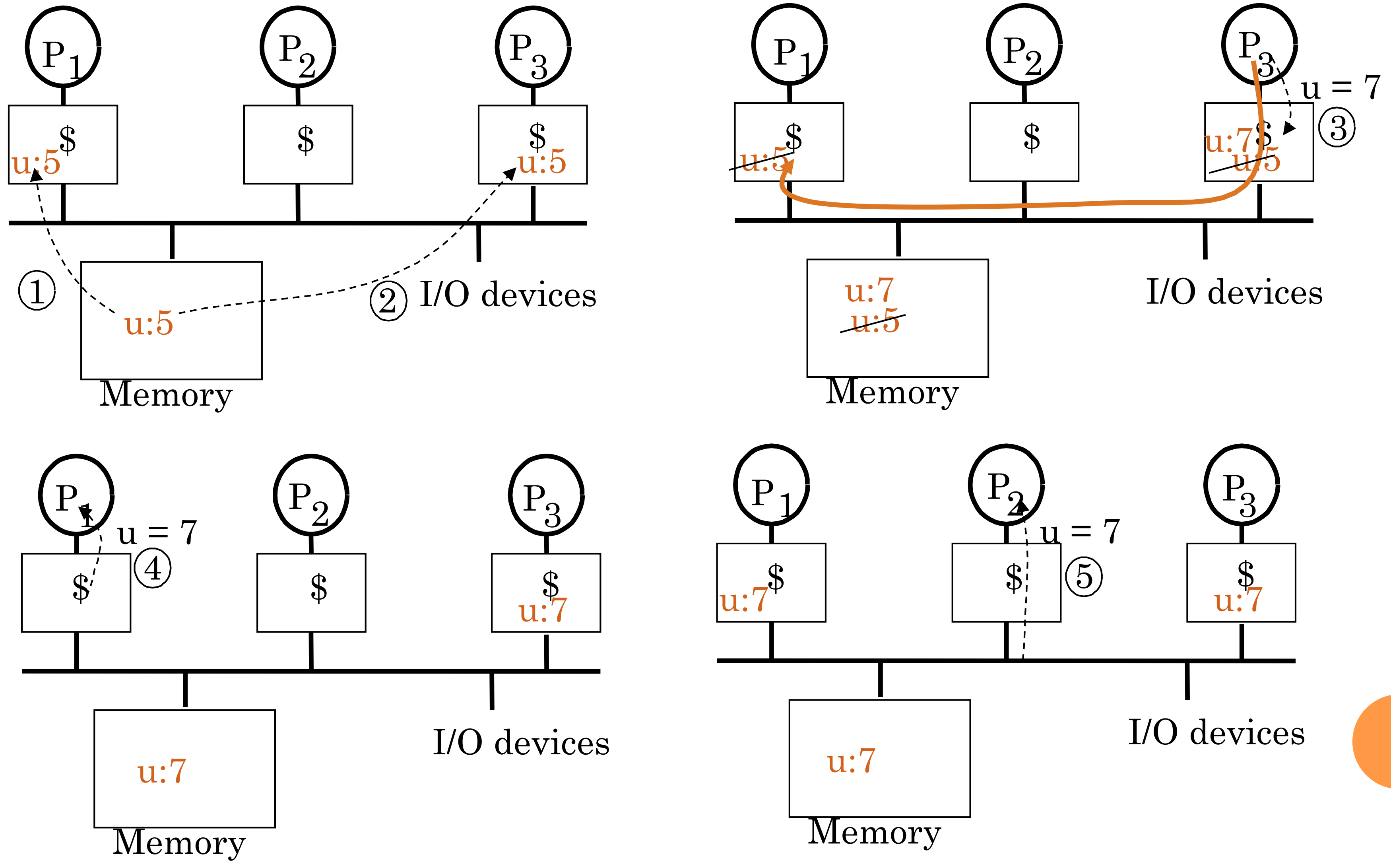


- Processors will see different value for u after event 3
- Write-through caches: P_1 reads a stale copy
- Write-back caches: P_1 and P_2 read a stale copy
  - The value written back to main memory depends on which cache writes back and when

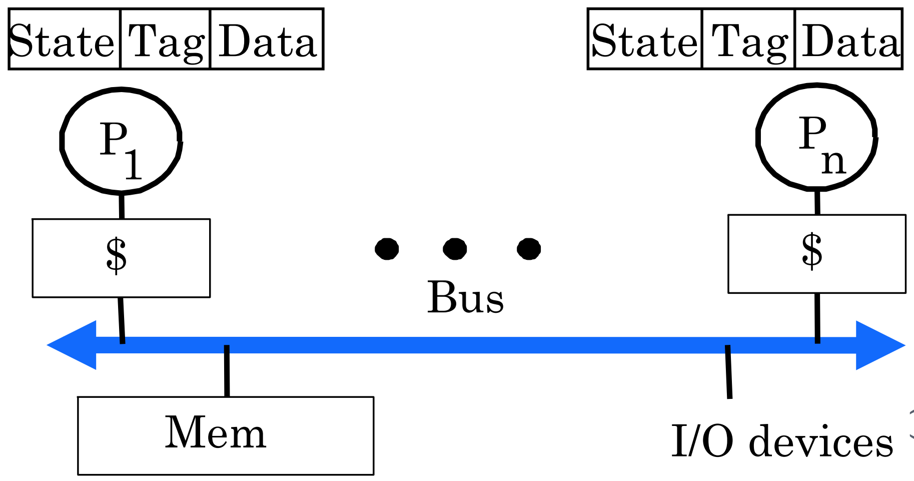
Cache-Coherence Through Bus Snooping



Example: Write-Through Caches With Invalidation-Based Protocol



SNOOPY PROTOCOL



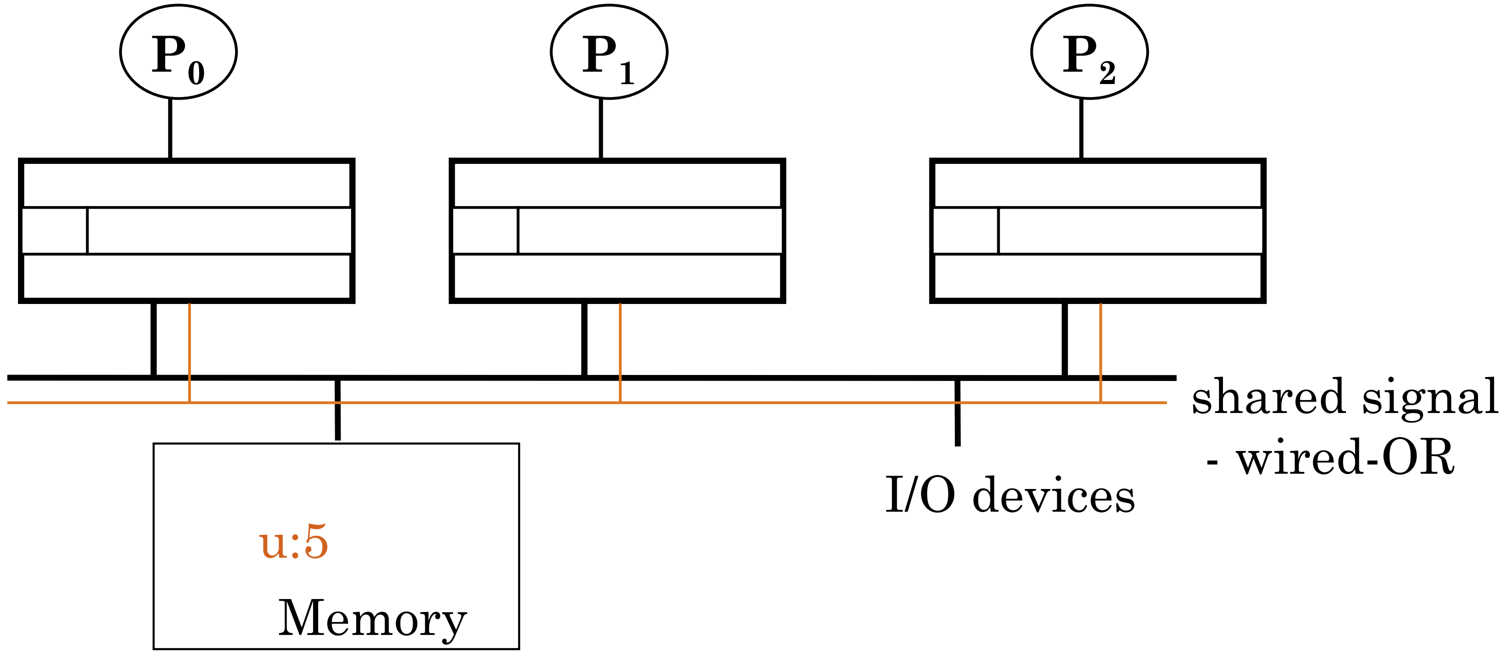
- Cache block state transition diagram:
  - Each cache block has a state associated with it
  - FSM specifies how the state of a block changes
- Controller updates state of cache blocks in response to processor and snoop events and generates bus transactions
- A snooping protocol is a distributed algorithm represented by a collection of cooperating FSMs

MESI Write-Back Invalidation

- Add *exclusive* state
  - distinguish exclusive (writable) and owned (written)
  - main memory is up to date, so cache not necessarily owner
  - can be written locally
- States
  - I -- Invalid
  - E – exclusive (only this cache has copy, but not modified)
  - S -- shared (two or more caches may have copies)
  - M -- modified (dirty)
- I → E on PrRd if no cache has copy

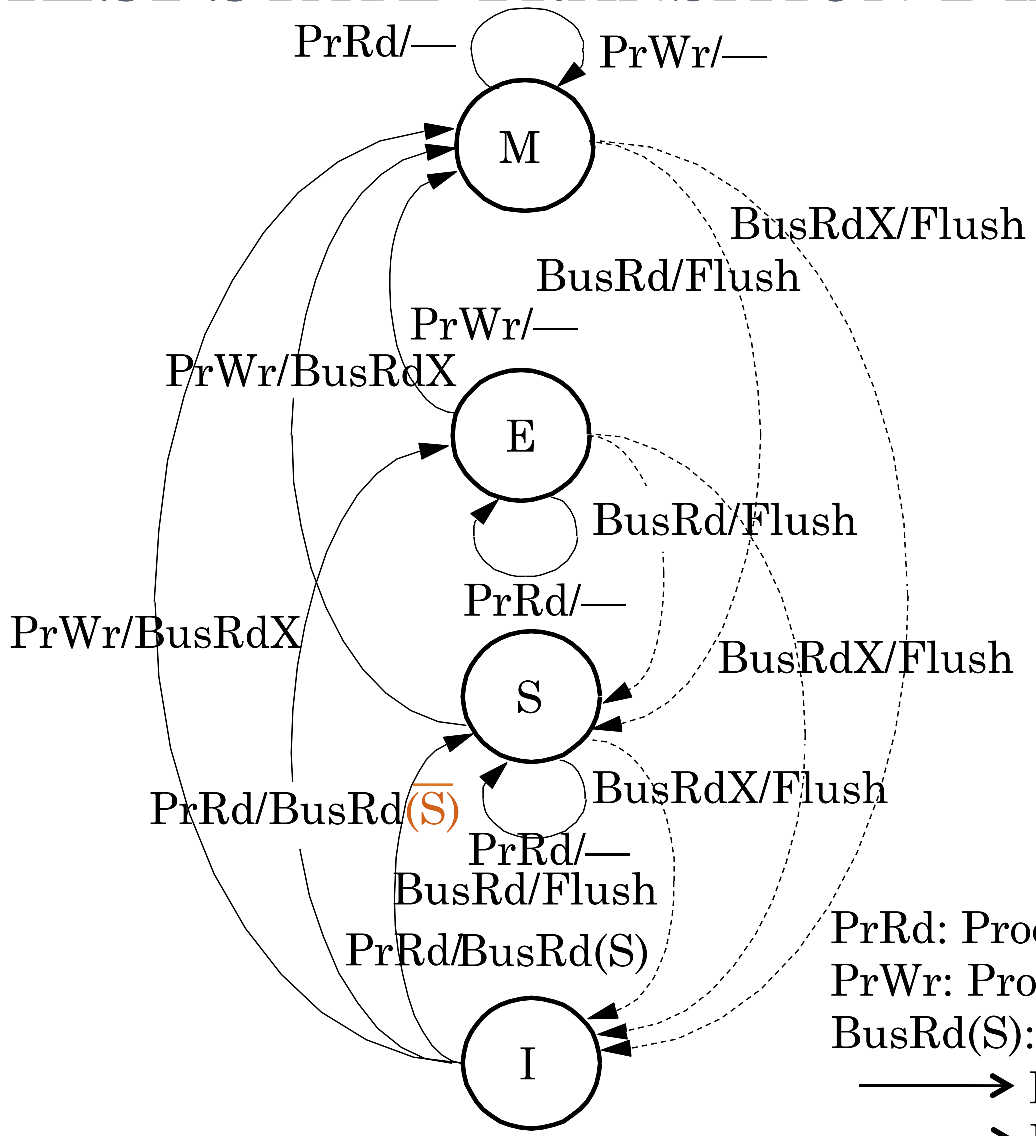
  => How can you tell?

Hardware Support For MESI Protocol



- All cache controllers snoop on BusRd
- During the address phase of bus transaction, caches which have a copy of the block assert the shared signal
- The controller making the request chooses between S and E

MESI State Transition Diagram



PrRd: Processor Read BusRd: Bus Read
PrWr: Processor Write BusRdX: Bus Write
BusRd(S): Shared line asserted on BusRd
$\longrightarrow$ Processor initiated transaction
$\dashrightarrow$ Bus-snooper-initiated transaction

THINGS TO BE DONE

- Write a simulator to implement MESI protocol for 4-core shared memory system
- Input: Trace of a multi-threaded program in a file
- Assumption: 64B block, 4-way set-associative, 16KB L1 data cache
- Deliverables:
  - State transitions (NP, I, E, S, M) per 1000 data memory references
  - Number of coherence misses