Cache Coherence in gem5

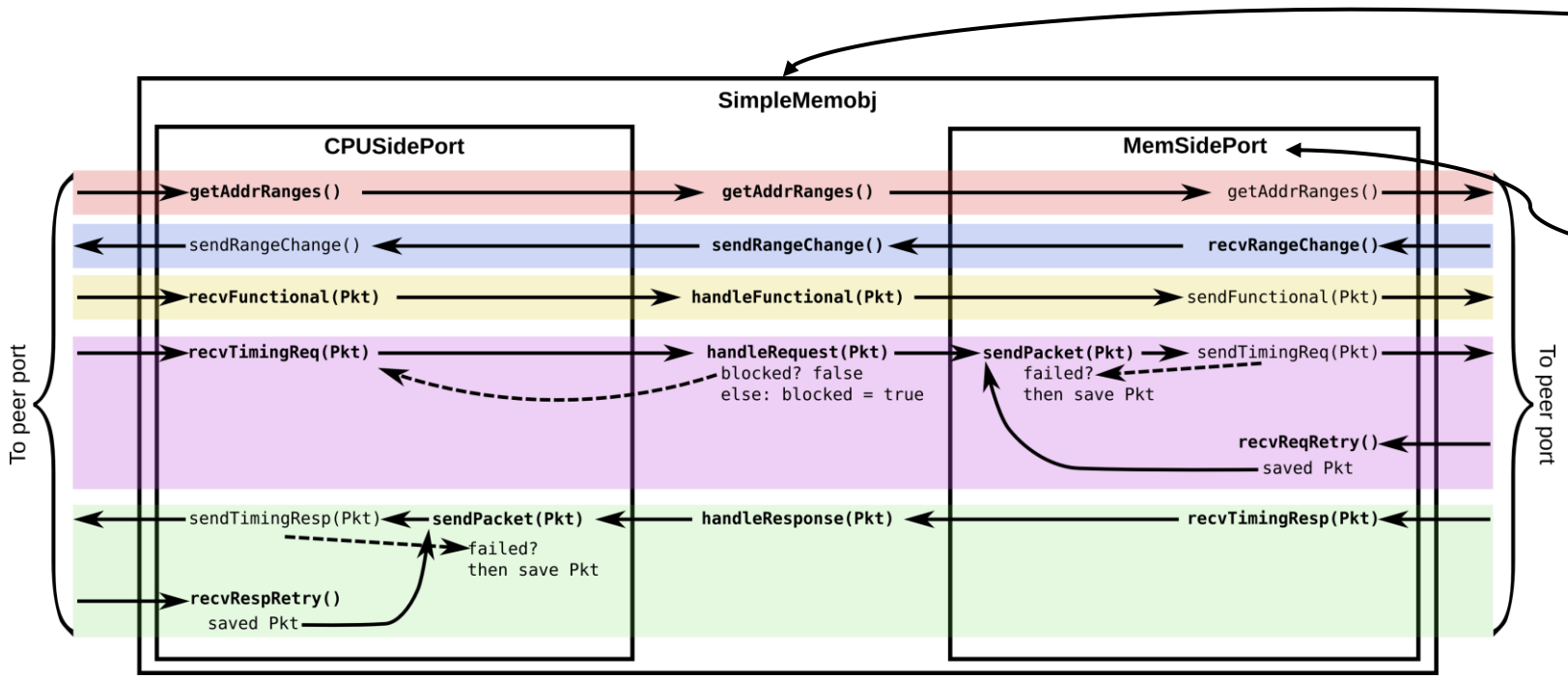
Yang Fan

2021/12/24

Cache Coherence/gem5 history

M5: “classic cache” CPU model master/slave port

GEMS: ruby network Memory system: access type



object

port

connections

request

packet

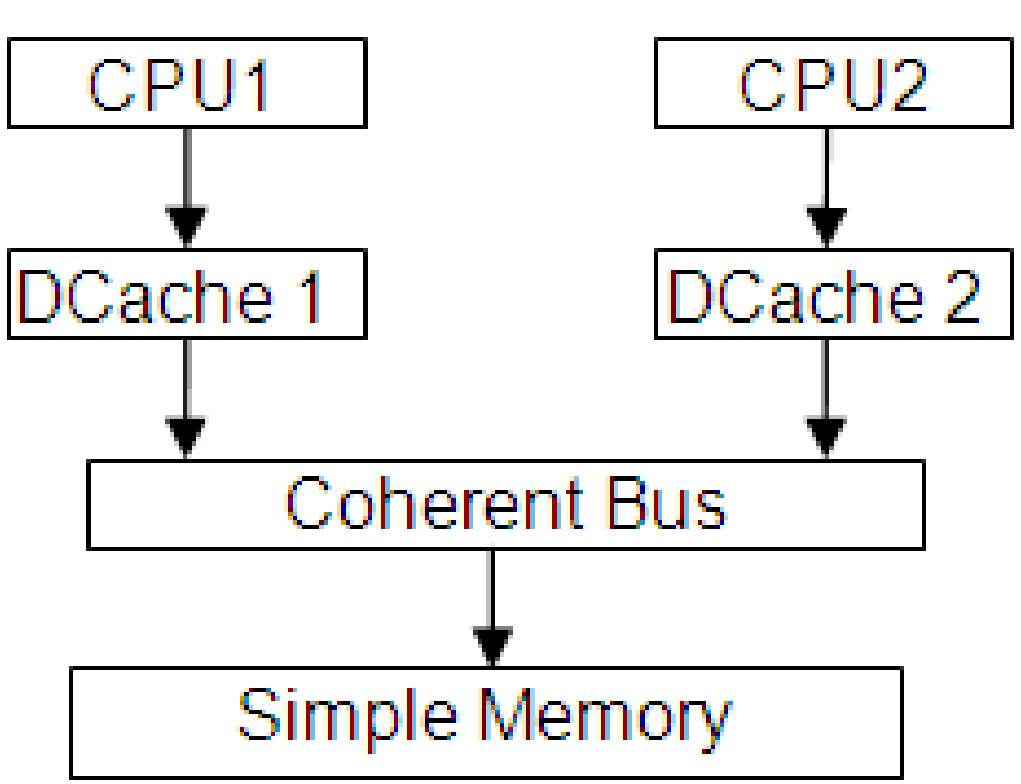
https://www.gem5.org/documentation/learning_gem5/part2/memoryobject/

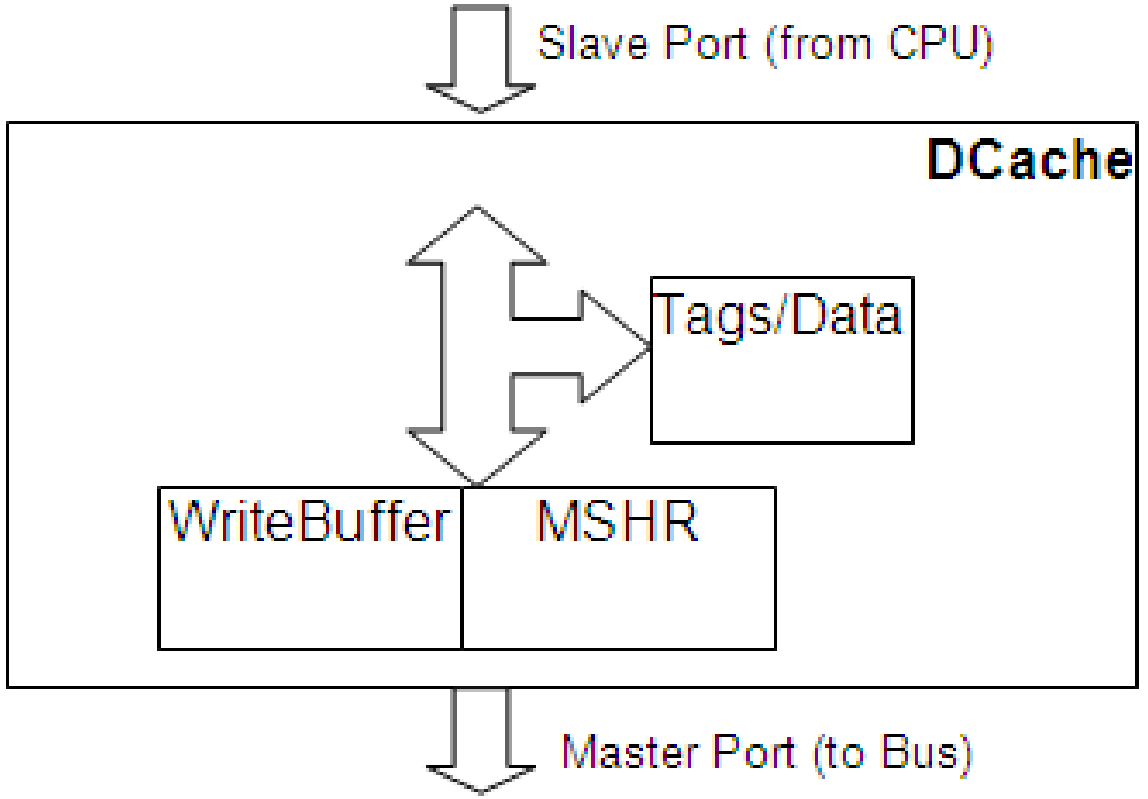
Outline

1. Simple memory cache Coherence
2. “Classic cc” problem
3. Ruby
4. SLICC

Simple memory cache Coherence

```
configs/example/fs.py --caches --cpu-type=arm_detailed --num-cpus=2
```





https://www.gem5.org/documentation/general_docs/memory_system/gem5_memory_system/

Simple memory cache Coherence——buffer/MSHR

Cached memory reads

Cached memory writes

Uncached memory reads

Uncached memory write

Evicted (& dirty) cache lines

Dcache is blocked:

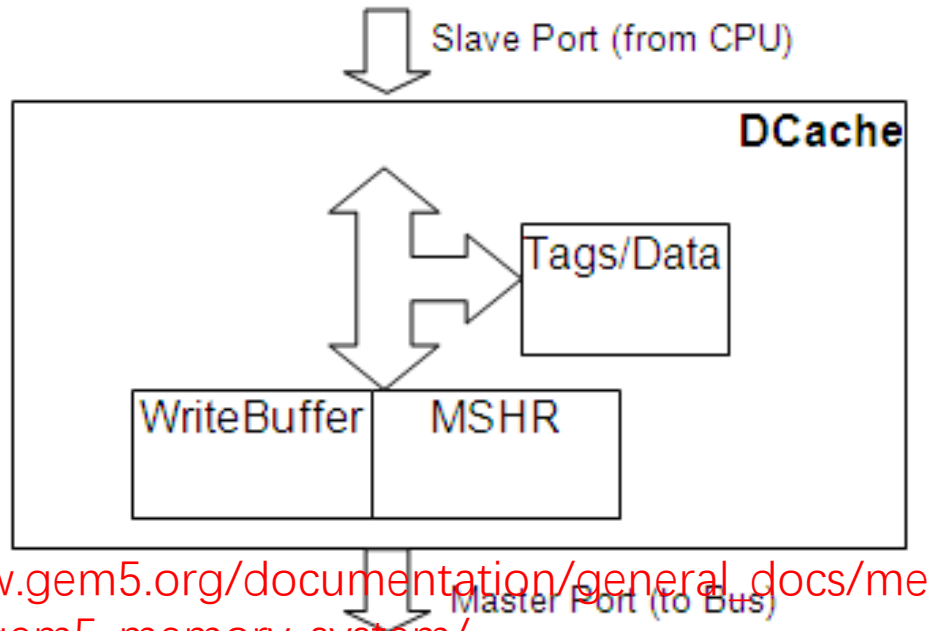
1. MSHR block is full
2. Writeback block is full
3. To reach threshold

Simple memory cache Coherence

Tags & Data Block

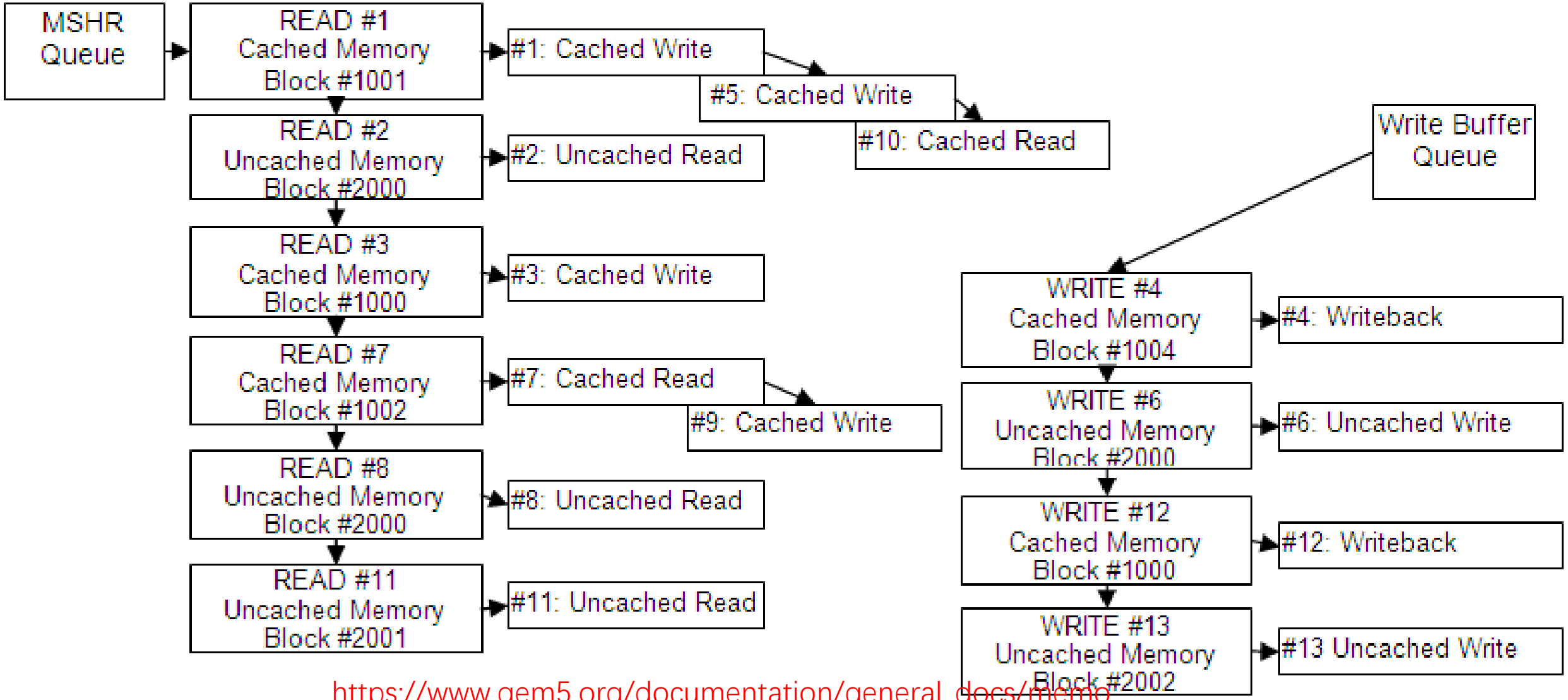
Valid Read Write Dirty

MSHR and Write Buffer Queues



https://www.gem5.org/documentation/general_docs/memory_system/gem5_memory_system/

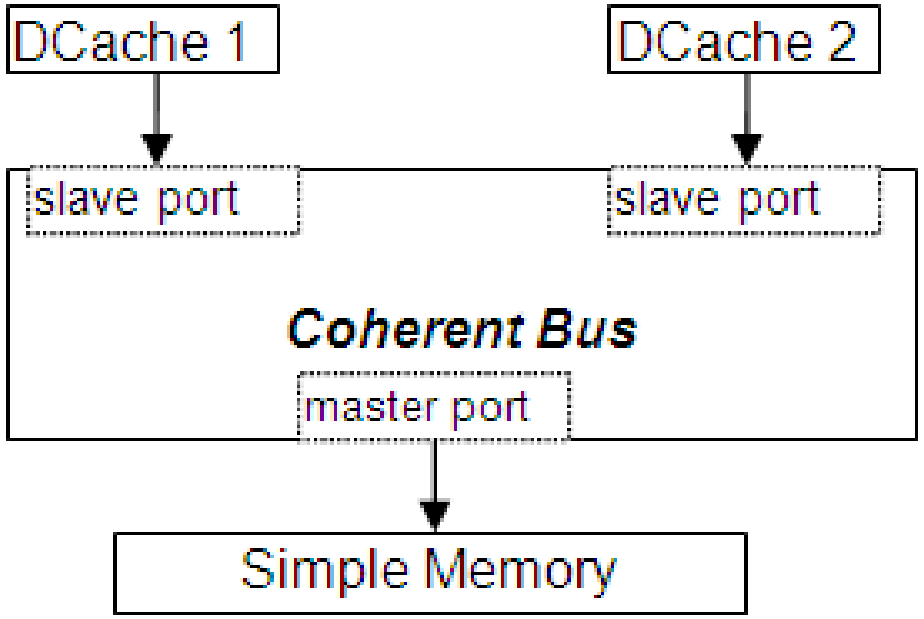
Memory Access Ordering

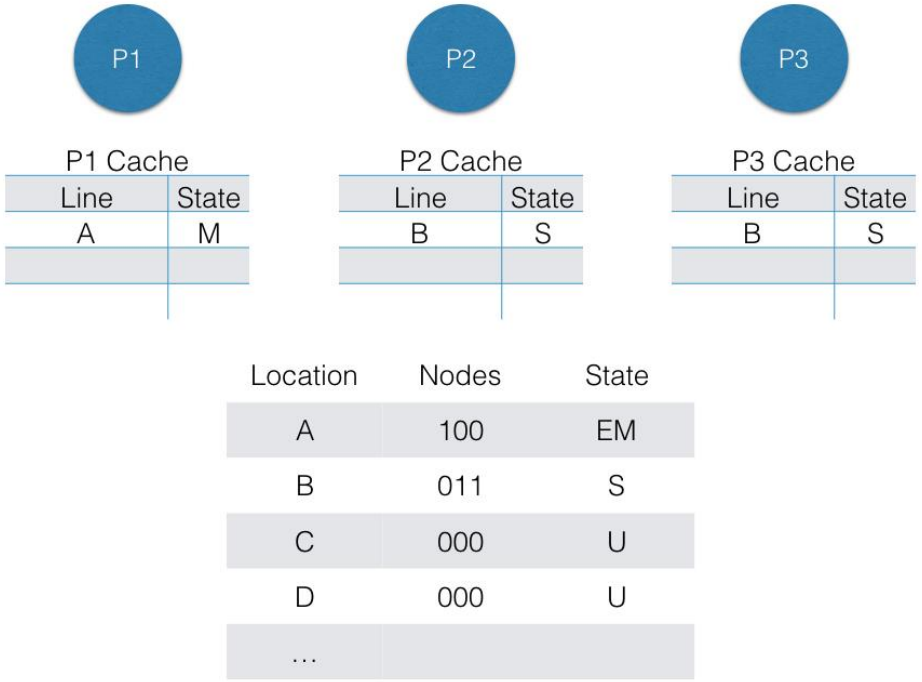


https://www.gem5.org/documentation/general_docs/memory_system/gem5_memory_system/

Coherent Bus Object

https://en.wikipedia.org/wiki/File:Full_bit_vector_format_diagram.jpg





Snoop Protocol

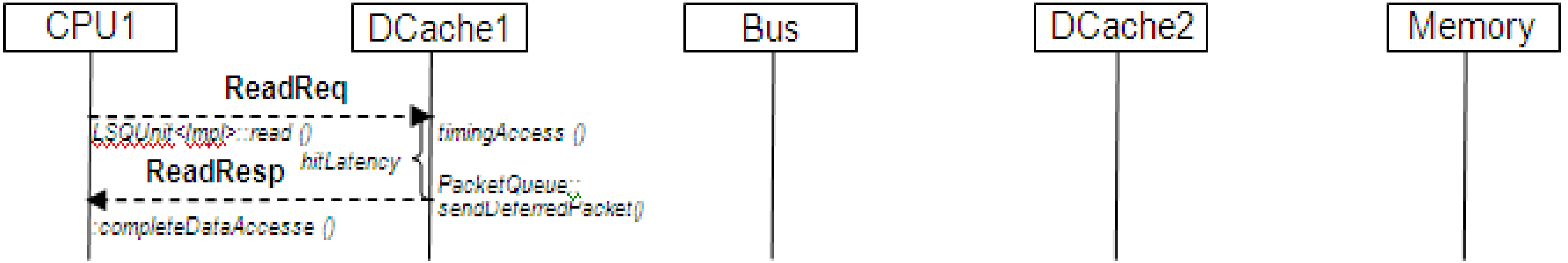
MESI、MOSI、MOESI

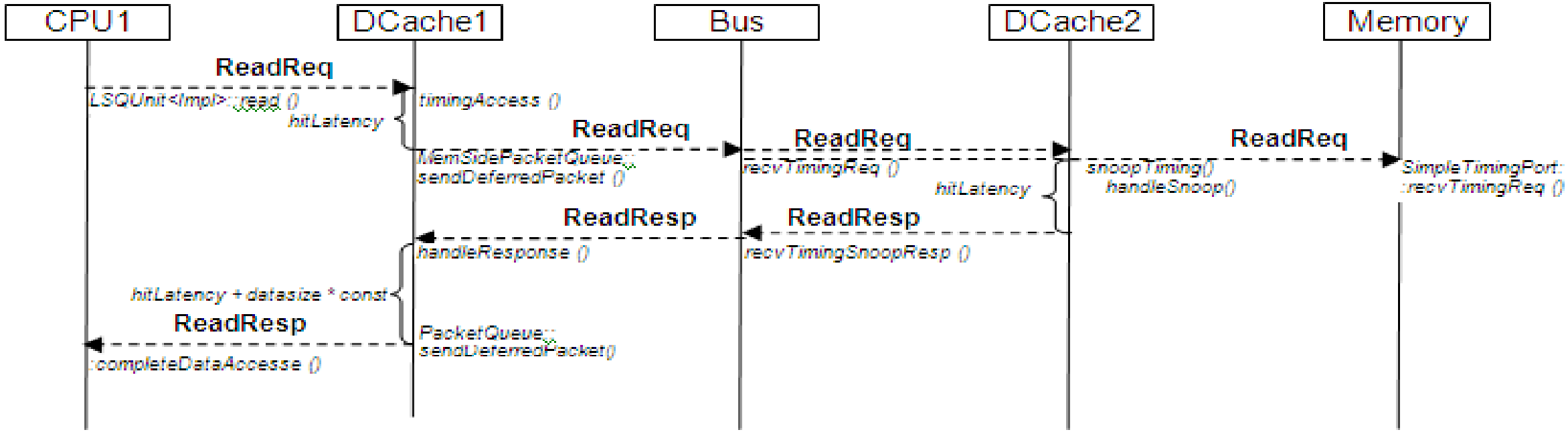
Directory-Based

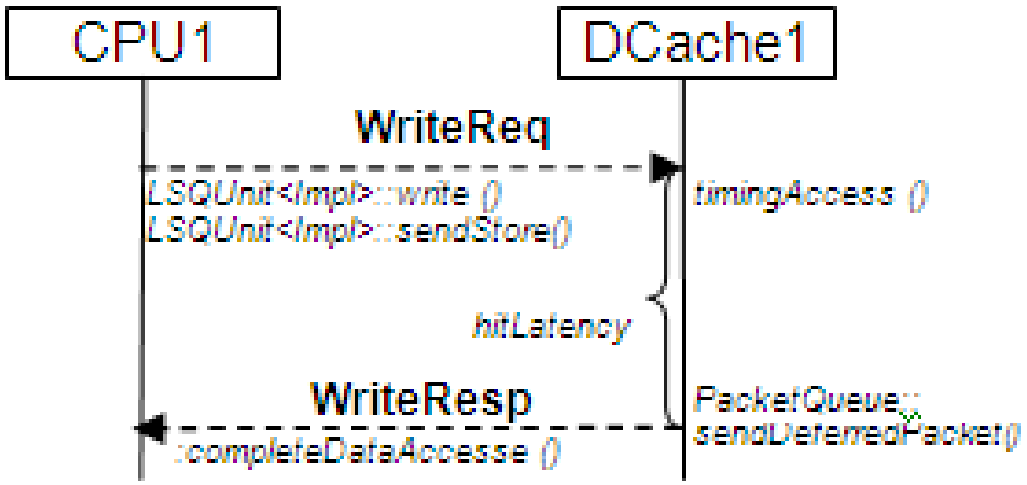
https://www.gem5.org/documentation/general_docs/memory_system/gem5_memory_system/

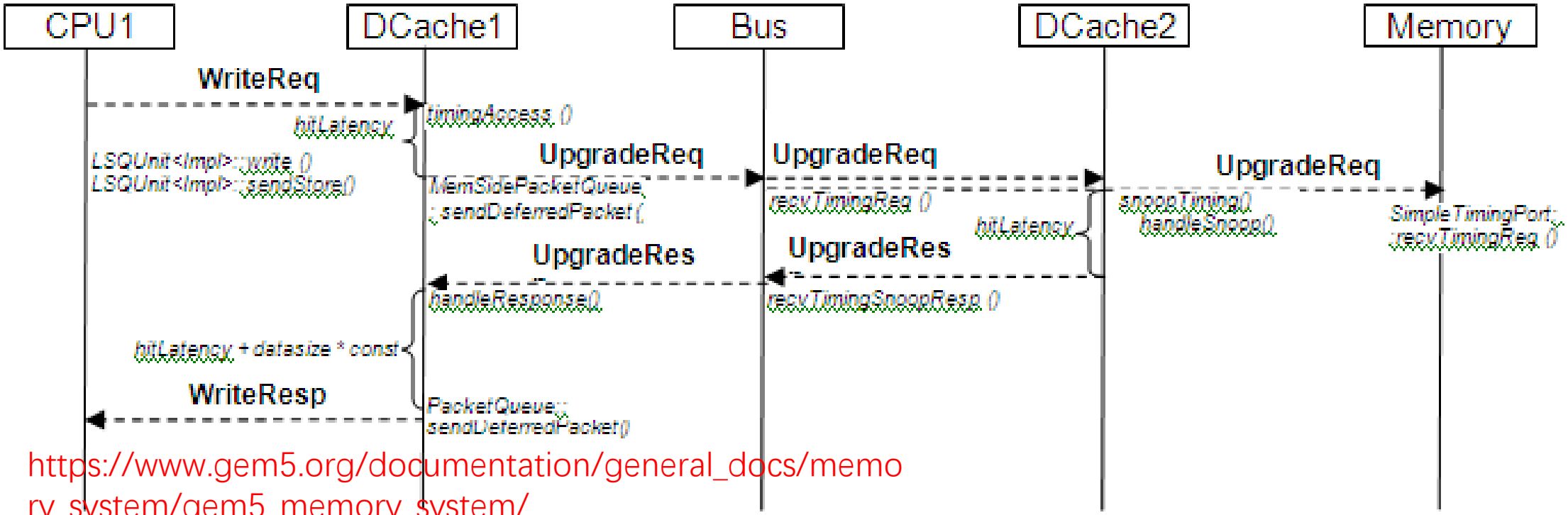
Message Flow

https://www.gem5.org/documentation/general_docs/memory_system/gem5_memory_system/

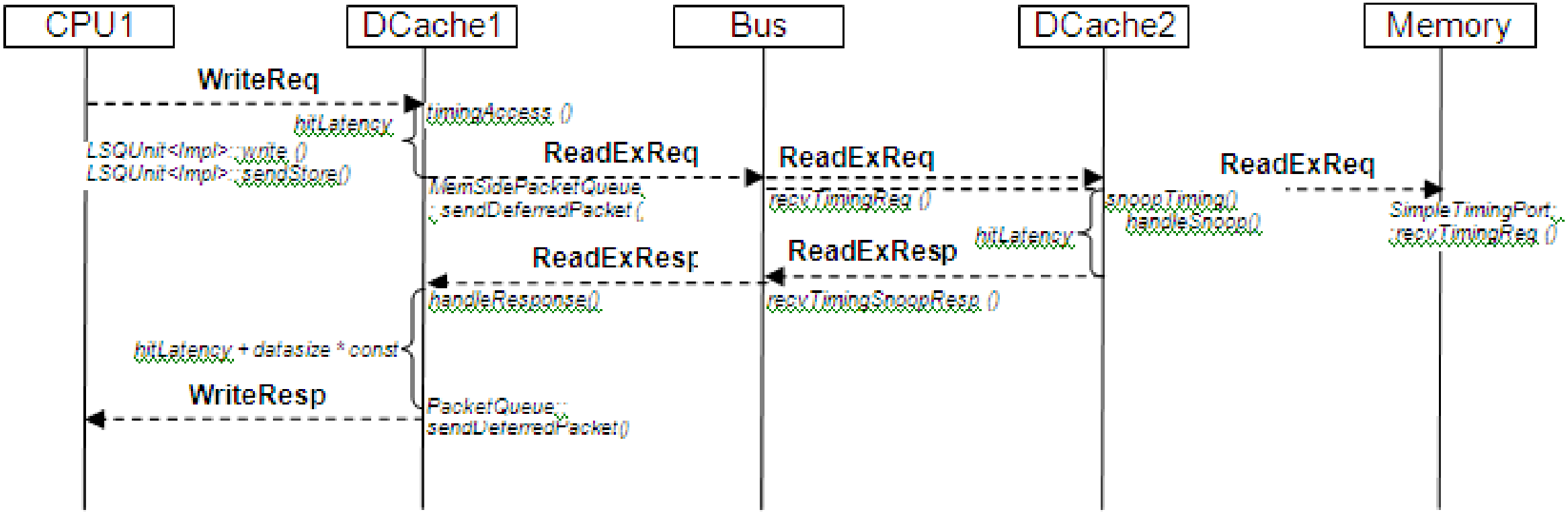






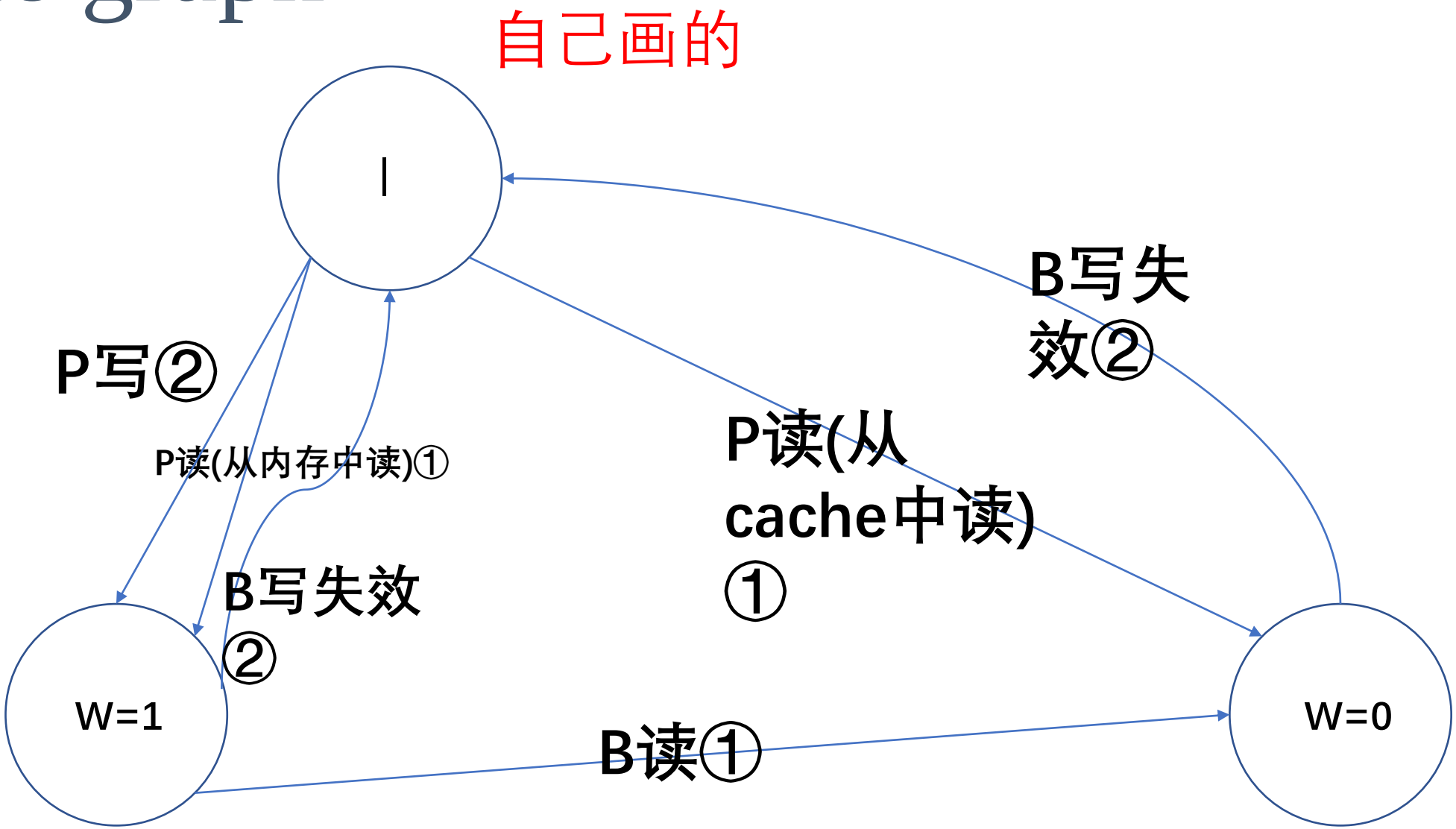


https://www.gem5.org/documentation/general_docs/memory_system/gem5_memory_system/



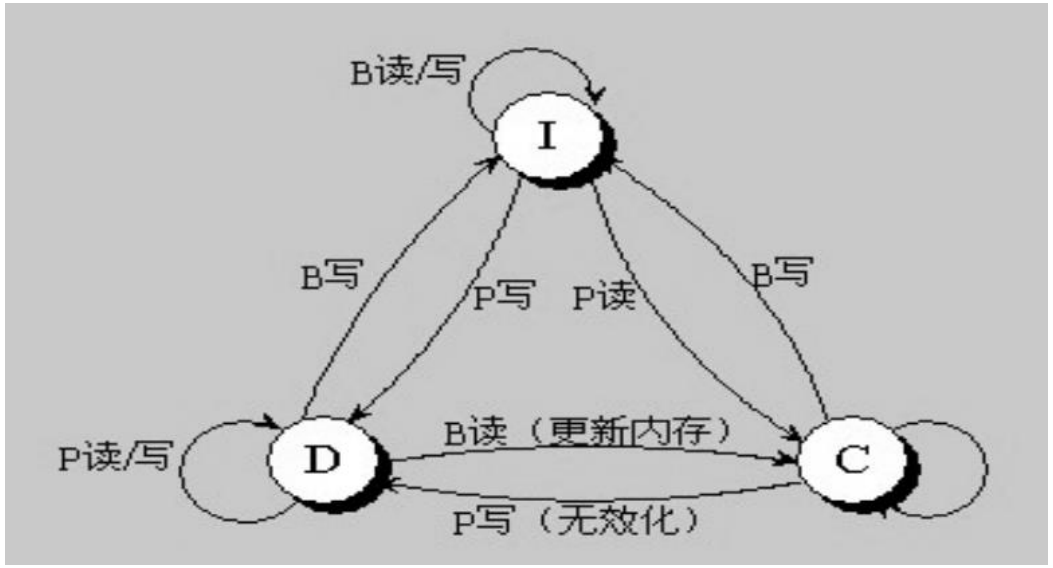
https://www.gem5.org/documentation/general_docs/memory_system/gem5_memory_system/

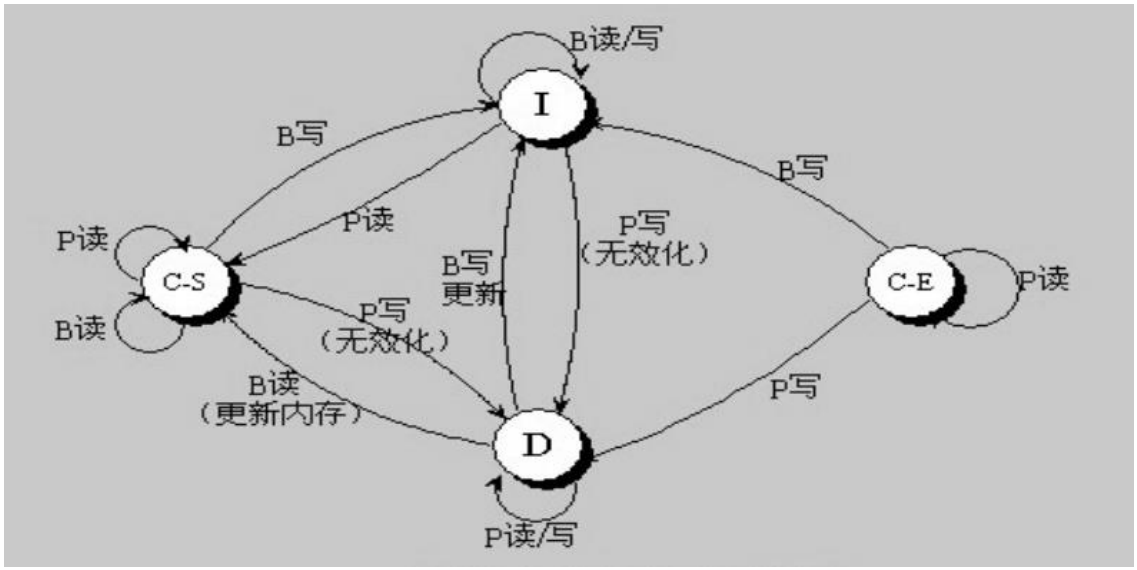
State graph

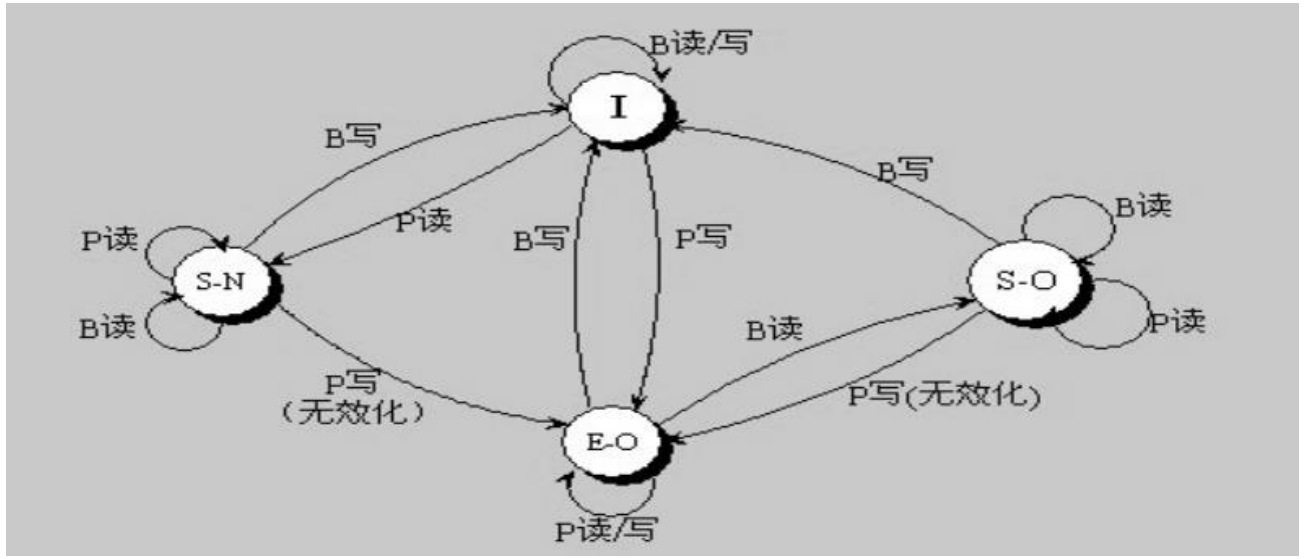


Cache Coherence in gem5

课件截图(第七页)

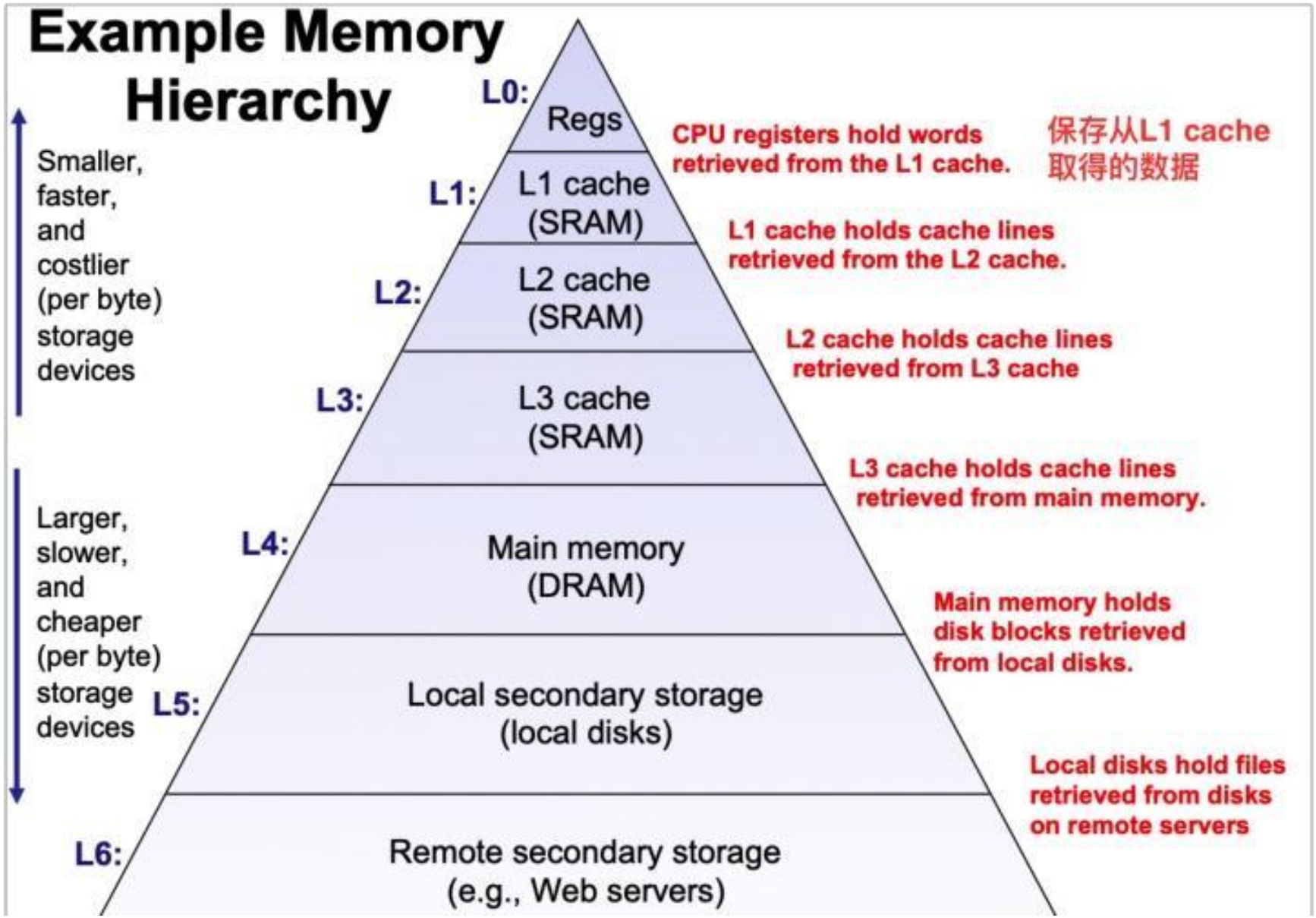






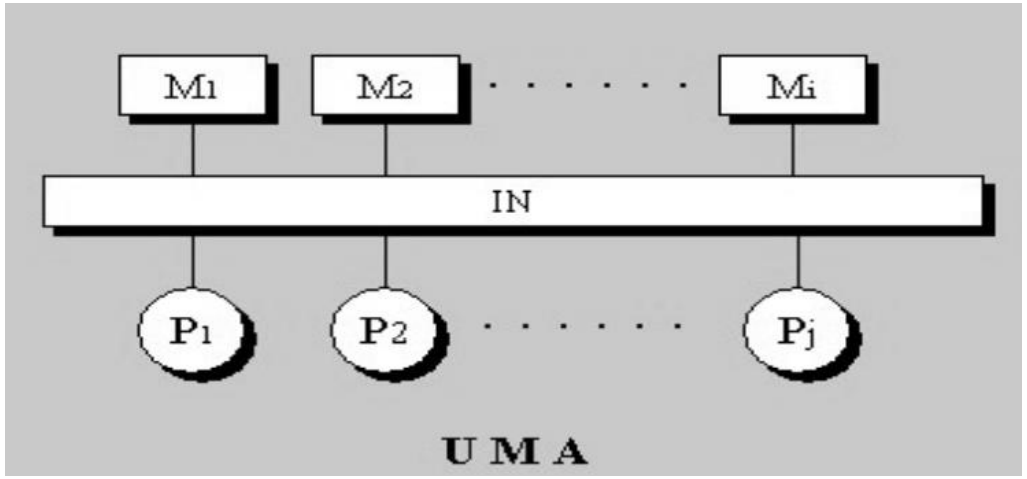
L1/L2 cache?

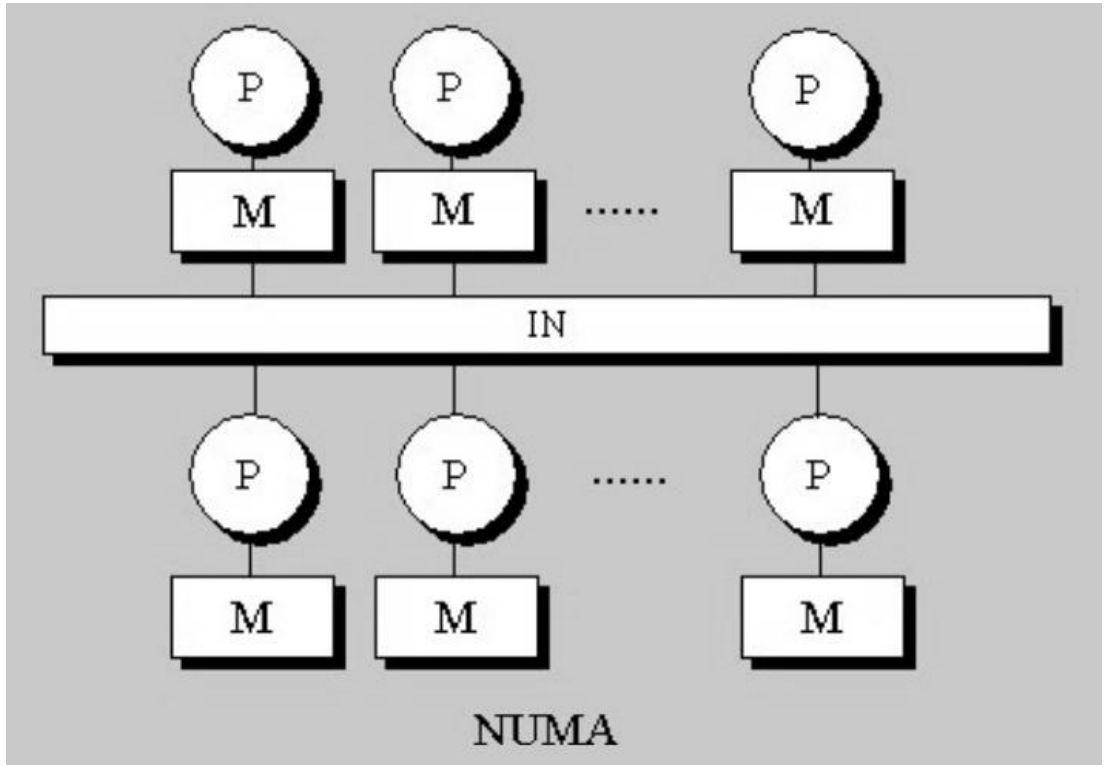
百度上找的cache层级图… 暂时找不到了…

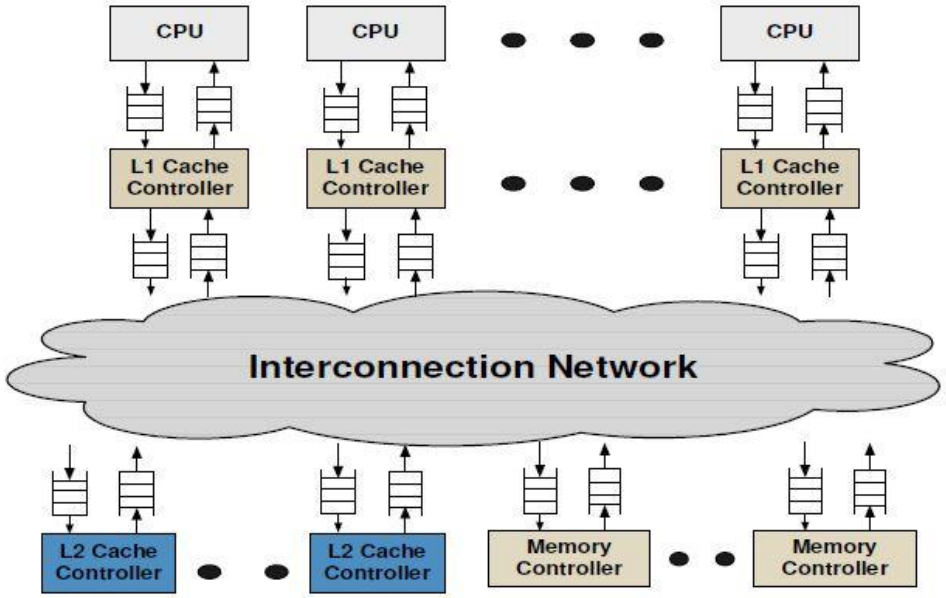


Topology

课件截图(第三页)







Ruby

https://www.youtube.com/watch?v=XTIrVBb86aM

Classic

+ Flexible model
+ Arbitrary cache hierarchy
+ Multiple replacement policies and prefetchers

– Rigid coherence protocol – magic snoops
– Only cross-bar interconnect

Therefore → Not realistic for Large SoC

Ruby

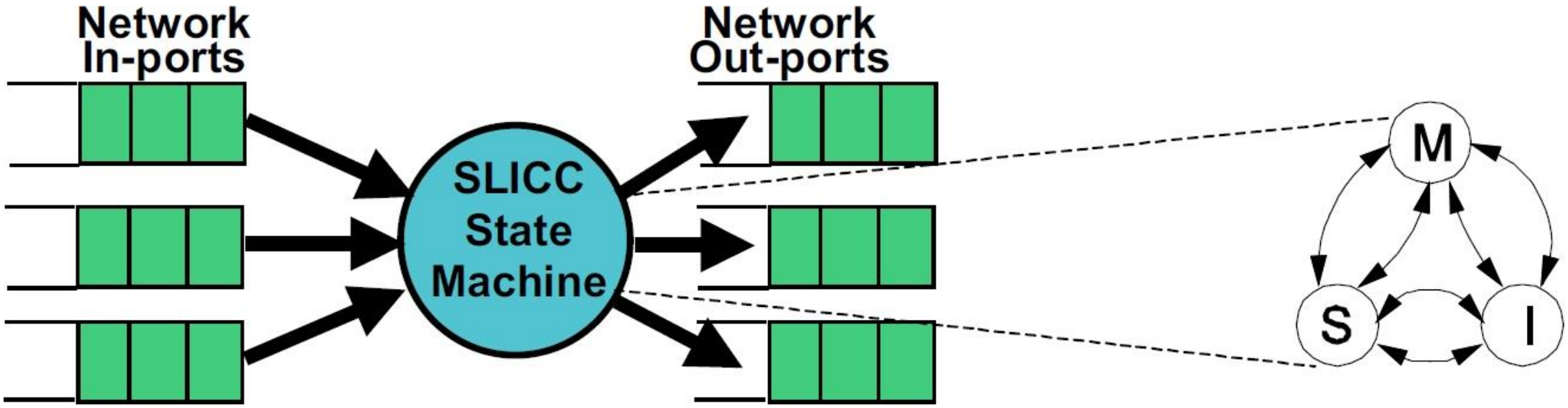
+ Detailed and extendible coherence protocols
+ Flexible interconnect modeling – mesh, cross-bar

– Rigid cache hierarchies
– Need specific protocol per cache configuration
– No (or very limited) prefetching support
– Poor integration with latest classic features

SLICC

https://www.gem5.org/documentation/general_docs/ruby/

Specification Language for Implementing Cache Coherence



MI_example-cache.sm: cache controller specification
MI_example-dir.sm: directory controller specification
MI_example-dma.sm: dma controller specification
MI_example-msg.sm: message type specification

SLICC?

```
machine(MachineType:L1Cache, "MI Example L1 Cache")
  : Sequencer * sequencer,
    CacheMemory * cacheMemory,
    int cache_response_latency = 12,
    int issue_latency = 2 {
      // Add rest of the stuff
    }
```

https://www.gem5.org/documentation/general_docs/ruby/slicc/

```
MessageBuffer requestFromCache, network="To", virtual_network="2", ordered="true";
MessageBuffer responseFromCache, network="To", virtual_network="4", ordered="true";
```

```
state_declaration(State, desc="Cache states") {
   I, AccessPermission:Invalid, desc="Not Present/Invalid";
   II, AccessPermission:Busy, desc="Not Present/Invalid, issued PUT";
   M, AccessPermission:Read_Write, desc="Modified";
   MI, AccessPermission:Busy, desc="Modified, issued PUT";
   MII, AccessPermission:Busy, desc="Modified, issued PUTX, received nack";
   IS, AccessPermission:Busy, desc="Issued request for LOAD/IFETCH";
   IM, AccessPermission:Busy, desc="Issued request for STORE/ATOMIC";
}
```

Thanks

Yang Fan

2021/12/24