Flow Specification in GEM5

***Abstract*—This paper will go through a brief overview for the implemented instances of these flows are given. These flow specifications capture how messages are exchanged for different use cases. In this model, a message is defined with the following format, (Src,Dest,Cmd), where Src and Dest refer to the source and destination components of messages, Cmd refers to the operations that the destination component should perform.**

I. Write Flow Specification

The LPN as shown in Fig. **??** specifies a system flow where CPU1 initiates a memory write operation. In this flow, CPU1 initiates a memory write request to L1 Cache. Next, depends on whether the required data is included in cache1, the cache1 will generate three possible responses. First, the cache1 will send read exclusive request message msg2 to interconnect if data is not present in cache; Or if the data is shared by CPU0, it will send upgrade message msg16 to interconnect to disable CPU0's ownership of this block of data, else if CPU1 has exclusive right of required data, cache1 will perform the write operation and sent an response message msg15 to CPU1. Afterwards, interconnect will sent request to all connected component (data cache and instruction cache of each CPU and memory). Once the interconnect obtains the response from either CPU0 or memory, it will generate a response message to cache1, then cache1 will generate a write response message msg14 (or msg 21) to CPU1. The flow is symmetric for CPU2.

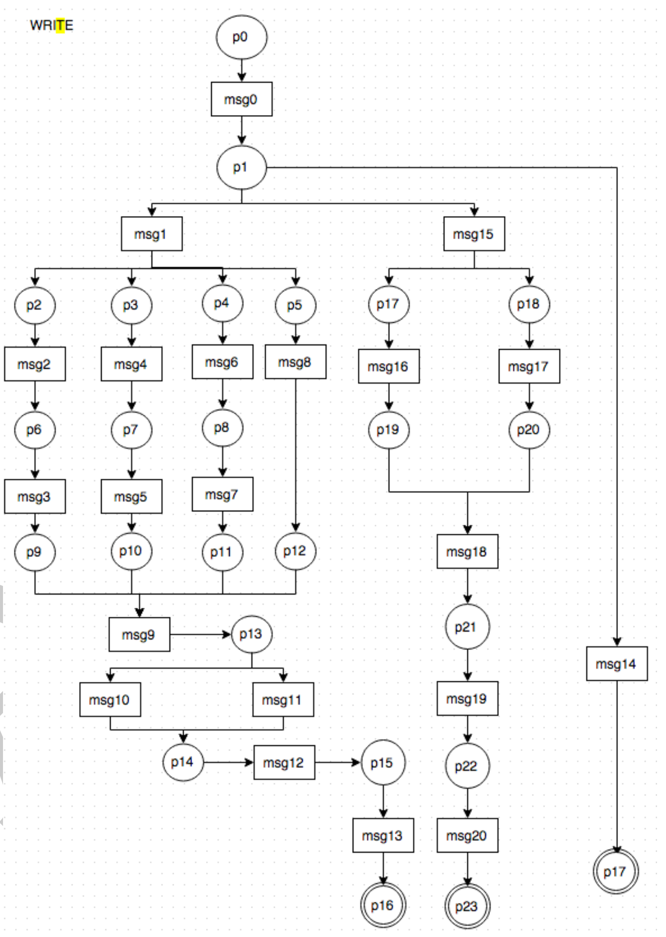
II. Read Flow Specification

Read operation has two flow specification as different coherence protocols are implemented depend on which cache the read operation is initialized to.

The LPN specification as shown in Fig. **??** captures the system flow where CPU1 initiate a memory read operation to its instruction cache. When CPU1 will first initiate a memory read request message msg1 to icache1, ICache1 can generate two possible responses. First, if the requested data is not in DCache1, ICache1 will generate an storeCondReq message msg2 to interconnect asking for data. Second, if ICache1 has the exclusive right of the requested data, it will generate a read response message msg14 to CPU1.

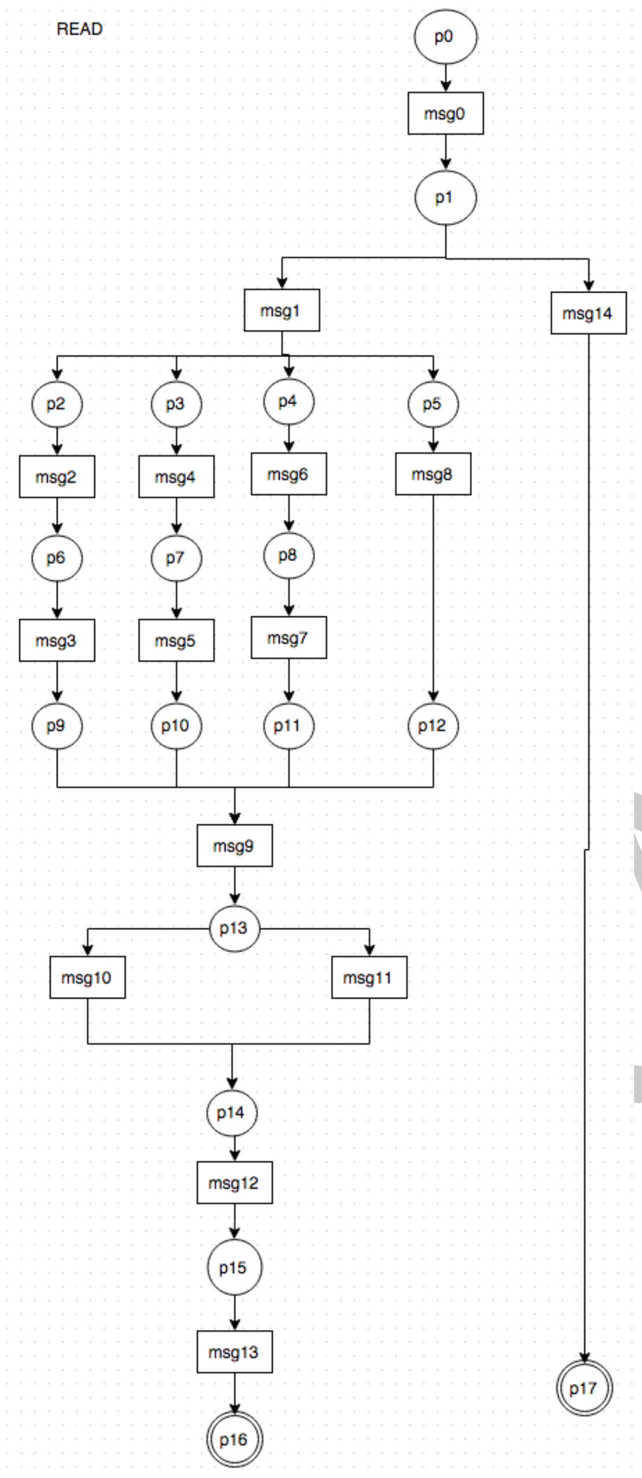
The LPN specification shown in Fig. **??** describes the system flow when CPU1 initiate a memory read operation to its data cache. DCache1 can also generate two possible responses to the read request. First, if the requested data is included in DCache1 but it's shared, cache1 will generate a load locked data request message msg21 to make sure it has the newest data. Second possibility is that when DCache1 has the requested, it will directly issue a read response message msg 33 to CPU1.

This two specifications are also symmetric for CPU2.



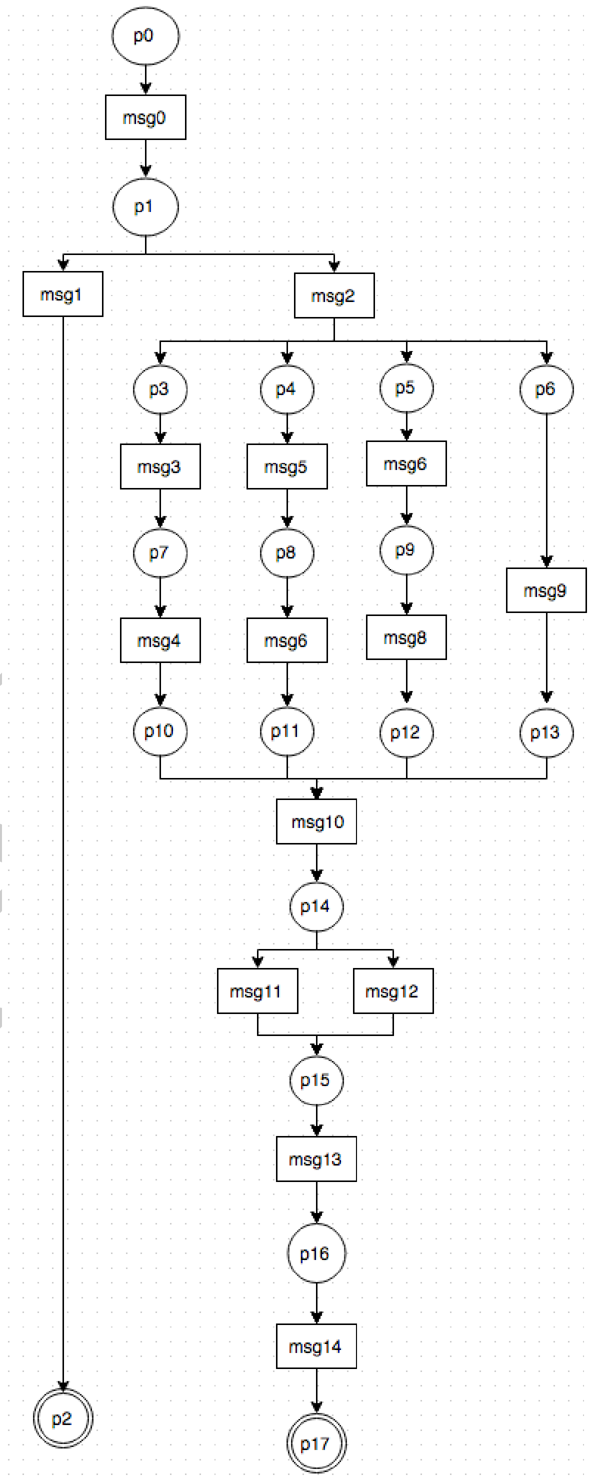
msg_0 : (	CPU1,	writeReq,	icache1)
msg_1 : (	dcache1,	readExreq ,	Bus)
msg_2 : (	Bus,	readExreq,	dcahce2)
msg_3 : (	dcache2,	readExreq,	cpu2)
msg_4 : (	Bus,	readExreq,	icahce2)
msg_5 : (	icache2,	readExreq,	cpu2)
msg_6 : (	Bus,	readExreq,	icahce1)
msg_7 : (	dcache1,	readExreq,	cpu1)
msg_8 : (	Bus,	readExreq,	Memory)
msg_9 : (	true)		
msg_{10} : (	Memory,	readExres,	Bus)
msg_{11} : (	icache2,	readExres,	Bus)
msg_{12} : (	Bus,	readExres,	dcache1)
msg_{13} : (	icache1,	writeRes,	CPU1)
msg_{14} : (	icache1,	writeRes,	CPU1)
msg_{15} : (	dcache1,	UpgradeReq)	
msg_{16} : (	Bus,	UpgradeReq,	icahce2)
msg_{17} : (	Bus,	UpgradeReq,	Memory)
msg_{18} : (	icache2,	UpgradeRes,	Bus)
msg_{19} : (	Bus,	UpgradeRes,	icache1)
msg_{20} : (	icache1,	WriteRes,	CPU1)

Fig. 1. Flow specification (F_1) of a cache coherent write operation initiated from CPU1



$msg0$: (	CPU1,	ReadReq,	icache1)
$msg1$: (	dcache1,	StoreCondreq ,	Bus)
$msg2$: (	Bus,	StoreCondreq,	icahce2)
$msg3$: (	icache2,	StoreCondreq,	cpu2)
$msg4$: (	Bus,	StoreCondreq,	dcahce2)
$msg5$: (	dcache2,	StoreCondreq,	cpu2)
$msg6$: (	Bus,	StoreCondreq,	dcahce1)
$msg7$: (	icache1,	StoreCondreq,	cpu1)
$msg8$: (	Bus,	StoreCondreq,	Memory)
$msg9$: (	true)		
$msg10$: (	Memory,	ReadRes,	Bus)
$msg11$: (	icache2,	ReadRes,	Bus)
$msg12$: (	Bus,	ReadRes,	dcache1)
$msg13$: (	icache1,	ReadRes,	CPU1)
$msg14$: (	icache1,	ReadRes,	CPU1)

Fig. 2. Flow specification (F_2) of a cache coherent read operation initiated from CPU1 to instruction cache



$msg0$: (	CPU1,	ReadReq,	dcache1)
$msg1$: (	dcache1,	ReadRes,	CPU1)
$msg2$: (	icache1,	LoadLockedreq ,	Bus)
$msg3$: (	Bus,	LoadLockedreq,	dcahce2)
$msg4$: (	dcache2,	LoadLockedreq,	cpu2)
$msg5$: (	Bus,	LoadLockedreq,	icahce2)
$msg6$: (	icache2,	LoadLockedreq,	cpu2)
$msg7$: (	Bus,	LoadLockedreq,	dcahce1)
$msg8$: (	icache1,	LoadLockedreq,	cpu1)
$msg9$: (	Bus,	LoadLockedreq,	Memory)
$msg10$: (	true)		
$msg11$: (	Memory,	ReadRes,	Bus)
$msg12$: (	icache2,	ReadRes,	Bus)
$msg13$: (	Bus,	ReadRes,	icache1)
$msg14$: (	dcache1,	ReadRes,	CPU1)

Fig. 3. Flow specification (F_2) of a cache coherent read operation initiated from CPU1 to data cache