HourGlass: Predictable Time-based Cache Coherence Protocol for Mixed-Time Critical Multi-Cores *

Nivedita Sritharan, Anirudh Kaushik, Mohamed Hassan, Hiren Patel

November 30, 2017

This technical report provides additional details of the HourGlass cache coherence protocol for mixed-time critical systems (MTCS). Table 1 and Table 3 show the complete state machines for the HourGlass protocol for the private L1-D cache controllers, and the shared memory, respectively. We use $*$ to identify transitions that differ based on the criticality level of the requestor in Table 1, and we describe these in detail in Table 2.

HourGlass Protocol's States and Transitions

The HourGlass cache coherence protocol consists of *stable states* and *transient states*. There are three stable states: modified (M), shared (S), and invalid (I). Transient states are intermediate states between a pair of stable states. They are primarily used to denote that a core's cache controller is waiting for data response and/or for its memory request to be ordered on the snooping bus. A memory request is said to be ordered on the snooping bus when all the cores and the shared memory observe the request on the snooping bus. All the cores and shared memory observe the same ordering of memory requests on the snooping bus. For example, consider the transient state IM^{AD} in Table 1. A core that issues a store request to a cache line that it has in the I state moves to the IM^{AD} state. This is shown in the first row of Table 1. The IM^{AD} state is a transient state between the stable states I and M. The superscript A denotes that the core is waiting for its memory request to be ordered on the bus, and the superscript D denotes that the core is waiting for its data response. Therefore, the request currently in the IM^{AD} is waiting for its request to be placed on the bus, and the response for the data as well. Similarly, there are transient states between other pairs of stable states.

A finite state machine in the core's cache controller observes coherence messages that are broadcasted on the snooping bus, and triggers transitions between the states for the cache line being accessed. These transitions change the coherence state of the cache lines. There are three standard coherence messages that conventional MSI cache coherence protocols include, which are as follows.

- $GetM$ coherence message is broadcasted on stores to signal that data is going to be modified.
- $GetS$ coherence message on loads to signal that data is going to only be read.
- $PutM$ coherence message on dirty cache line replacements to signal that data needs to be written back.

In addition to these standard coherence messages, HourGlass adds the $SelfInv$, $AllInv$, and $SendData$ coherence messages.

- $SelfInv$ message is broadcasted when the core needs to downgrade or self-invalidate from the shared state (S, S^TI, SI^A) to I state.
- $AllInv$ message is broadcasted by the shared memory when all sharers have self-invalidated. This coherence message is necessary to signal a pending store to a shared data that all sharers have self-invalidated, which eliminates violation of the fundamental SWMR invariant.
- $SendData$ message is broadcasted by a core that is ready to send a cache line to a requestor.

*Supplementary material

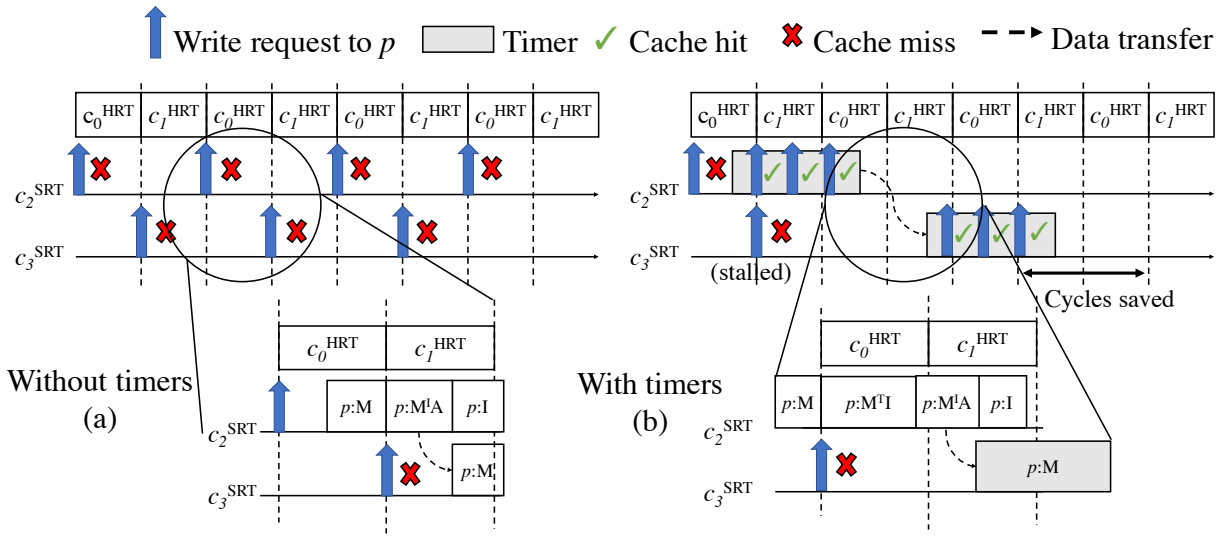

Figure 1: Effect of timers on execution time of SRT cores.

Note that a core views coherence messages on the bus as either Own or $Other$ coherence message (prefixed to the above coherence messages). An Own coherence message such as $OwnGetM$/$OwnGetS$ denotes that the requesting core has observed its own store/load request ordered on the bus. An $Other$ coherence message such as $OtherGetS$/$OtherGetM$ denotes that the core observes a remote (another core) load/store request on the bus. Transitions based on Own and $Other$ coherence messages are tabulated in Table 1. Using the examples presented in the manuscript that describe and evaluate HourGlass, we describe the state transitions of HourGlass.

Detailed explanation of Example in Figure 2b

Figure 2 in the submitted manuscript is presented as Figure 1 above. Consider Figure 1b in the submitted manuscript. Initially c_2^{SRT} has cache line p in the M state. The shared memory also has p in the M state. c_2^{SRT} holds p for a time duration, which enables other stores from c_2^{SRT} to p be cache hits. This is because stores to a cache line in M state are cache hits (line 14, Table 1). When c_2^{SRT} observes a $OtherGetM$ – due to c_3^{SRT}, c_2^{SRT} moves from M to M^TI, and records the $SendData$ coherence message in the PRSP buffer of c_2^{SRT} with the requesting core id set to c_3^{SRT}. The M^TI/S^TI are transient states that indicate that the timers for the cache line are currently counting down to zero, and once the timers expire, the cache line's final state is I. The $SendData$ coherence message will be broadcasted in c_3^{SRT}'s slot (requestor slot) once c_2^{SRT}'s timer expires. c_3^{SRT} moves from I to IM^{AD}, and on seeing its $OwnGetM$, moves from IM^{AD} to IM^D. Once c_2^{SRT}'s timer expires, p in c_2^{SRT} moves from M^TI to MI^A (line 15, Table 1). Note that loads and stores to p in MI^A are cache hits (line 16, Table 1). It sets the valid bit for the $SendData$ message in the PRSP buffer entry. In c_3^{SRT}'s slot, the $SendData$ message is broadcasted, and on seeing $OwnSendData$, c_2^{SRT} sends the requested data to c_3^{SRT} (line 16, Table 1). c_3^{SRT} on receiving the data, moves p from IM^D to M. The HourGlass state transitions in Figure 3b closely resemble those described for Figure 2b.

Detailed explanation of Example in Figure 4b

Figure 4 in the submitted manuscript is presented as Figure 2. Consider Figure 2b. Initially c_1^{HRT} has cache line p in the M state. c_2^{SRT} issues a store request to p, and moves from I to IM^{AD}. When c_1^{HRT} sees a $OtherGetM$ – srt message on the bus, it moves p from M to M^TI state. It records a $SendData$ in its PRSP buffer with the requesting core id set to c_2^{SRT}. This $SendData$ message will be broadcasted in c_2^{SRT}'s slot after the timer for p on c_1^{HRT} expires. c_2^{SRT} moves from IM^{AD} to IM^D on seeing its $OwnGetM$. While c_1^{HRT} is in the M^TI state, a remote store request from c_0^{HRT} to p is observed on the bus. Since the current requesting core id in the PRSP buffer entry for p in c_1^{HRT}

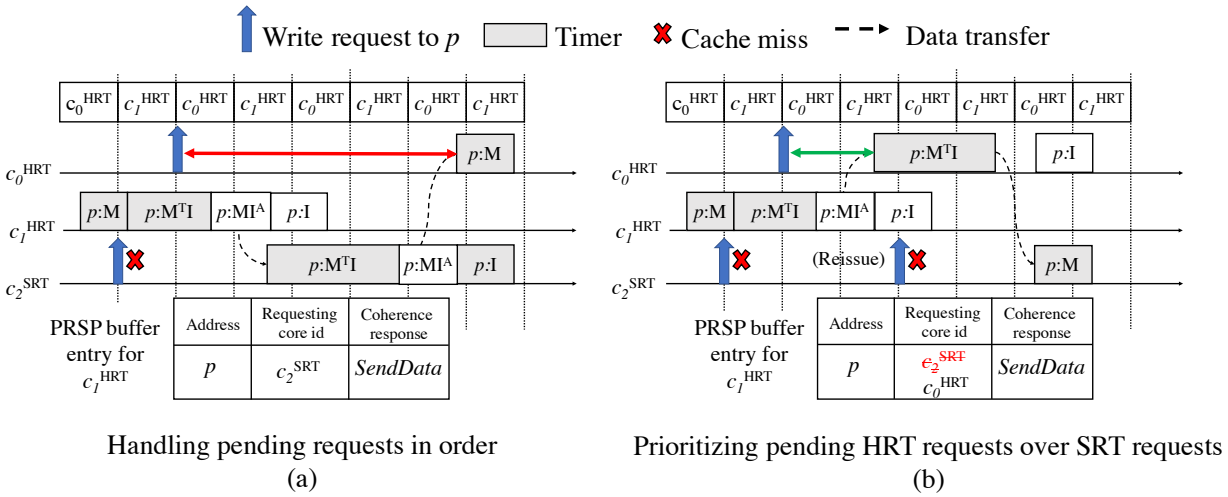

Figure 2: Handling pending SRT and HRT memory requests.

(c_2^{SRT}) has a lower criticality level than c_0^{HRT}, the requesting core id for this entry is updated to c_0^{HRT}. Hence, data is sent from c_1^{HRT} to c_0^{HRT} even though c_0^{HRT}'s request to p is broadcasted *after* c_2^{SRT}'s request to p. From Table 2, c_2^{SRT} is in the IM^D state, and observes a $OtherGetM-$ (due to c_0^{HRT}) coherence message (line 7, Table 2). Since HourGlass prioritizes HRT requests over pending SRT requests to the same cache line to tighten the WCL bounds of HRT tasks, c_2^{SRT} reissues the store, clears any pending responses to p in its PRSP buffer, and moves to the IM^{AD} state. Once the timer for p in c_1^{HRT} expires, c_1^{HRT} moves p from M^TI to MI^A, and sets the valid bit in the $SendData$ PRSP buffer entry for p. In c_0^{HRT}'s slot, the $SendData$ message is broadcasted, and on seeing $OwnSendData$, c_1^{HRT} sends the requested data to c_0^{HRT} (line 16, Table 1). c_0^{HRT} on receiving the data, moves p from IM^D to M. When p in c_0^{HRT} is in M state, the reissued store request from c_2^{SRT} is observed on the bus. Hence, c_0^{HRT} will send the data to c_2^{SRT} once its timer for p expires.

#	State	Core events				Bus events - common to HRT and SRT cores								
		Load	Store	Replacement	Timeout	OwnGetS	OwnGetM	OwnPutM	OwnSelfInv	AllInv	OwnSendData	Data	OtherGetS-hrt or srt	OtherGetM-hrt or srt
1	I	issue GetS/IS^{AD}	issue GetM/IM^{AD}	X	X	X	X	X	-	-	X	X	-	-
2	IS^{AD}	X	X	X	X	/IS^{D}	X	X	-	-	X	X	-	-
3	IS^{D}	X	X	X	X	X	X	X	-	-	X	load/S and ST	*	*
4	IS^{D}I	X	X	X	X	X	X	X	-	-	X	load/S^{T}I and ST	*	*
5	S	hit	/S^{T}M and WT	issue SelfInv/SI^{A}	/S and RT	X	X	X	X	X	X	X	-	issue SelfInv/ S^{T}I and WT
6	S^{T}I	hit	S^{T}M and WT	issue SelfInv/SI^{A}	set valid bit in PRSP/SI^{A}	X	X	X	X	X	X	X	-	issue SelfInv
7	SI^{A}	hit	issue SelfInv and GetM/SM^{A}	issue SelfInv	X	X	X	X	/SI	X	X	X	-	issue SelfInv
8	SI	hit	issue GetM/IM^{AD}	/I	X	X	X	X	X	/I	X	X	-	-
9	S^{T}M	hit	X	X	issue SelfInv and GetM and set valid bit in PRSP/SM^{A}	X	X	X	X	X	X	X	-	issue SelfInv
10	SM^{A}	hit	X	X	X	X	/IM^{D}	X	/IM^{AD}	X	X	X	-	issue SelfInv
11	IM^{AD}	X	X	X	X	X	/IM^{D}	X	-	-	X	X	-	-
12	IM^{D}	X	X	X	X	X	X	X	-	-	X	store/M and ST	*	*
13	IM^{D}I	X	X	X	X	X	X	X	-	-	X	store/M^{T}I and ST	*	*
14	M	hit	hit	issue PutM/MI^{A}	/M and RT	X	X	X	X	X	X	X	issue SendData / M^{T}I and WT	issue SendData / M^{T}I and WT
15	M^{T}I	hit	hit	issue PutM and SendData/MI^{A}	set valid bit in PRSP/MI^{A}	X	X	X	X	X	X	X	issue SendData	issue SendData
16	MI^{A}	hit	hit	issue PutM	X	X	X	WB/I	X	X	send data to requestor/I	X	issue SendData	issue SendData

Table 1: HourGlass private cache coherence states. Shaded rows are new states introduced by HourGlass. WT: Wait for timer timeout, RT: Restart timer, ST: Start timer. $msg/state$ denotes that a core issues the message msg, and moves to coherence state $state$. Cells marked as "$\times$" indicate that a particular transition cannot happen, and cells marked as "$-$" denote that a cache line in that state does not change state with a core event or bus event.

#	Criticality	State	Bus events			
			OtherGetS-hrt	OtherGetS-srt	OtherGetM-hrt	OtherGetM-srt
1	HRT	IS^D	-	-	issue SelfInv /IS^DI	issue SelfInv /IS^DI
2	HRT	IS^DI	-	-	issue SelfInv	issue SelfInv
3	HRT	IM^D	issue SendData /IM^DI	issue SendData /IM^DI	issue SendData /IM^DI	issue SendData /IM^DI
4	HRT	IM^DI	issue SendData	issue SendData	issue SendData	issue SendData
5	SRT	IS^D	-	-	reissue GetS and clear PRSP(addr) /IS^{AD}	issue SelfInv /IS^DI
6	SRT	IS^DI	-	-	reissue GetS and clear PRSP(addr) /IS^{AD}	issue SelfInv /IS^DI
7	SRT	IM^D	reissue GetM and clear PRSP(addr) /IM^{AD}	issue SendData /IM^DI	reissue GetM and clear PRSP(addr) /IM^{AD}	issue SendData /IM^DI
8	SRT	IM^DI	reissue GetM and clear PRSP(addr) /IM^{AD}	issue SendData /IM^DI	reissue GetM and clear PRSP(addr) /IM^{AD}	issue SendData /IM^DI

Table 2: Different HourGlass transitions based on core criticality. “clear PRSP(addr)” denotes clear PRSP entries for that address.

State	Bus events					Shared memory event
	GetS-/srt	GetM-/srt	SendData	PutM	Data	AllInv
I	send Data /S	send Data /M	X	X	X	X
S	send Data /S	add to PR LUT /SM	X	X	X	/I
SM	add to PR LUT	add to PR LUT	X	X	X	/I and issue AllInv and send data based on pending requests
M	add to PR LUT	add to PR LUT	update owner/M if GetM is the first request in PR LUT wait for data/S_D if GetS is the first request in PR LUT	wait for data/M_D	X	X
M_D	X	X	X	X	write to memory /I	X
S_D	X	X	X	X	write to memory /S	X

Table 3: State transitions at the shared memory

Table 4 describes the HourGlass private L1-D cache coherence states, along with their read/write permissions. The fields in Table 4 are:

- *State*: The HourGlass coherence state
- *Previous states*: Coherence states that precede a particular state. For example, the states I, IS^D and IS^DI are previous states to IS^{AD}. Based on some memory activity, a cache line in I, IS^D or IS^DI state changes to IS^{AD}.
- *Cause*: Reasons for a change in coherence state from the previous states. For example, a cache line moves from I to IS^{AD} on a load request.
- *Action*: Coherence and data responses issued by a core on moving to the next state based on the cause. For example, when I moves to IS^{AD} on a load request, it issues a $GetS$ coherence message.
- *Next states*: Coherence states that are immediately reachable from a state based on some memory activity. For example, a cache line in IS^{AD} can move to the IS^D state based on some memory activity.
- *Permissions*: The read and write permissions of a cache line based on the coherence state. None denotes no permissions, Read-only denotes read permissions, and Read-write denotes both read and write permissions. For example, cache lines in S state can only be read, and cache lines in M state can be read and written to.

We use a simple example to guide the readers in reading Table 4. Consider a store request to a cache line p that a core has in the I state. The core issues a store request, moves to the IM^{AD} state, and issues a $GetM$ message on the bus (lines 1 and 11, Table 4). Once the core observes its $OwnGetM$ coherence message on the bus, it changes the coherence state of p from IM^{AD} to IM^D (line 12, Table 4). When the shared memory or a remote core responds with the data, p changes coherence state from IM^D to M, and begins the timer countdown (line 14, Table 4).

#	State	Previous states	Cause	Action	Next states	Permissions
1	I	SI, MI^A	Cache line does not exist in a valid state in the private cache	-	IS^{AD}, IM^{AD}	None
2	IS^{AD}	I, IS^D, IS^DI	Load request or reissue due to pending HRT store requests to same cache line	Issue $GetS$	IS^D	None
3	IS^D	IS^{AD}	Observed $OwnGetS$	Waiting for data response	S, IS^DI	None
4	IS^DI	IS^D, S^TI	Observed $OtherGetM$	Waiting for data response	I	None
5	S	IS^D	Data response received for load request	Begin timers	S^TM, SI^A, S^TI	Read-only
6	S^TI	IS^DI, S	Data response received for load request (from IS^DI) or Observed $OtherGetM$ (from S)	Begin timers (from IS^DI) or wait for timer to expire before self-invalidating, and record $SelfInv$ message in PRSP buffer (from S)	S^TM, SI^A	Read-only
7	S^TM	S^TI, S	During timer countdown, observed local store.	Wait for timer to expire before completing store.	SM^A	Read-only
8	SM^A	S^TM, SI^A	Timer expired, pending store can be serviced once all sharers have self-invalidated	Issue $SelfInv$ message followed by $GetM$ message	IM^{AD}, IM^D	Read-only
9	SI^A	S, S^TI	Core performs cache line replacement.	Issue $SelfInv$ message.	SM^A, SI	Read-only
10	SI	SI^A	Observed $OwnSelfInv$ message	Wait for $AllInv$ coherence message before moving to I state.	I	Read-only
11	IM^{AD}	I, IM^D, IM^DI	Store request or reissue due to pending HRT store requests to same cache line	Issue $GetM$	IM^D	None
12	IM^D	IM^{AD}	Observed $OwnGetM$	Waiting for data response	M, IM^DI	None
13	IM^DI	IM^D	Observed $OtherGetM$	Waiting for data response	I	None
14	M	IM^D	Data response received for store request	Begin timers (ST)	MI^A, M^TI	Read-write
15	M^TI	M	Observed $OtherGetS$ or $OtherGetM$	Wait for timer to expire before self-invalidating, and record $SelfData$ message in PRSP buffer	MI^A	Read-write
16	MI^A	M, M^TI	Core performs cache line replacement or timer expires (from M^TI).	Issue $PutM$ messages. On seeing $OwnPutM$ move to I.	I	Read-write

Table 4: Description of HourGlass private cache coherence states.