

	Mem Data	Mem Ack	All Acks	L2 Replacement	L2 Replacement Clean
NP	-	-	-	-	-
SS	-	-	send Dir. to Req. / send Dir. to Req. /	send Dir. to Req. / send Dir. to Req. /	send Dir. to Req. / send Dir. to Req. /
M	-	-	-	send Dir. to Req. / send Dir. to Req. /	send Dir. to Req. / send Dir. to Req. /
MT	-	-	-	send Dir. to Req. / send Dir. to Req. /	send Dir. to Req. / send Dir. to Req. /
MCT	-	-	-	send Dir. to Req. / send Dir. to Req. /	send Dir. to Req. / send Dir. to Req. /
MCTI	-	-	-	send Dir. to Req. / send Dir. to Req. /	send Dir. to Req. / send Dir. to Req. /
LI	-	-	-	send Dir. to Req. / send Dir. to Req. /	send Dir. to Req. / send Dir. to Req. /
SJ	-	-	-	send Dir. to Req. / send Dir. to Req. /	send Dir. to Req. / send Dir. to Req. /
SS	send PutM to req. / send AckM to req. /	-	-	recycle	recycle
IS	send AckM to req. / SS	-	-	recycle	recycle
IM	send AckM to req. / MT, MI	-	-	recycle	recycle
SM	-	-	-	recycle	recycle
MT, MI	-	-	-	recycle	recycle
MT, MI	-	-	-	recycle	recycle
MT, MI	-	-	-	recycle	recycle

Table 2: Transition Table for MESI Inclusive Protocol L2, Part 2: events generated by memory messages / at the L2.

	IFetch	Load	Store	Inv	Fwd GetS	Fwd GetM	Data	DataM	Data from L1	Data All Acks	Fwd Ack	WB Ack	All Acks	Repl.
I	send <i>GetS</i> only / IS	send <i>GetS</i> / IS	send <i>GetM</i> / IM	send <i>Ack</i>	-	-	-	-	-	-	-	-	-	-
S	hit	hit	send <i>GetM</i> / SM	send <i>Ack</i> / I	-	-	-	-	-	-	-	-	-	/ I
E	hit	hit	hit / M	send <i>Ack</i> / I	send <i>Data</i> to req / send <i>Data</i> to dir / S	send <i>Data</i> to req / I	-	-	-	-	-	-	-	send <i>Put</i> / M_I
M	hit	hit	hit	send <i>Data</i> to dir / I	send <i>Data</i> to req / send <i>Data</i> to dir / S	send <i>Data</i> to req / I	-	-	-	-	-	-	-	send <i>Put</i> / M_I
IS	stall	stall	stall	send <i>Ack</i> / IS_I	-	-	-	send <i>UnblockM</i> / E	send <i>UnblockS</i> / S	/ S	-	-	-	stall
IM	stall	stall	stall	send <i>Ack</i>	-	-	incAcks / SM	-	-	send <i>UnblockM</i> / M	incAcks	-	-	stall
SM	stall	stall	stall	send <i>Ack</i> / IM	-	-	-	-	-	-	incAcks	send <i>UnblockM</i> / M	-	stall
IS_I	stall	stall	stall	send <i>Ack</i>	-	-	-	send <i>UnblockM</i> / E	send <i>UnblockS</i> hit / I	hit / I	-	-	-	stall
M_I	stall	stall	stall	send <i>Data</i> to dir / SINK	send <i>Data</i> to req / send <i>Data</i> to dir / SINK	send <i>Data</i> to req / SINK	-	-	-	-	-	-	/ I	stall
SINK	stall	stall	stall	send <i>Ack</i>	-	-	-	-	-	-	-	-	/ I	stall

Table 3: Transition Table for MESI Inclusive Protocol L1.

States	Accelerator Events			XG Requests Invalidate	XG Responses			
	Load	Store	Replacement		DataM	DataE	DataS	WBAck
M	hit	hit	send <i>PutM</i> / B	send <i>Dirty WB</i> / I	-	-	-	-
E	hit	hit / M	send <i>PutE</i> / B	send <i>Clean WB</i> / I	-	-	-	-
S	hit	send <i>GetM</i> / B	send <i>PutS</i> / B	send <i>InvAck</i> / I	-	-	-	-
I	send <i>GetS</i> / B	send <i>GetM</i> / B	-	send <i>InvAck</i>	-	-	-	-
B	stall	stall	stall	send <i>InvAck</i>	/ M	/ E	/ S	/ I

Table 4: Sample accelerator L1 cache implementing Crossing Guard’s interface.

	Accelerator Events			L2 Requests Invalidate	L2 Responses		
	Load	Store	Replacement		DataM	DataS	WBAck
M	hit	hit	send <i>PutM</i> / MI	send <i>WB</i> / I	-	-	-
S	hit	send <i>GetM</i> / IM	/ I	send <i>InvAck</i> / I	-	-	-
I	send <i>GetS</i> / IS	send <i>GetM</i> / IM	-	send <i>InvAck</i>	-	-	-
IS	<i>stall</i>	<i>stall</i>	<i>stall</i>	send <i>InvAck</i> / ISI	-	/ S	-
IM	<i>stall</i>	<i>stall</i>	<i>stall</i>	send <i>InvAck</i>	/ M	-	-
MI	<i>stall</i>	<i>stall</i>	<i>stall</i>	send <i>InvAck</i>	-	-	/ I
ISI	<i>stall</i>	<i>stall</i>	<i>stall</i>	send <i>InvAck</i>	-	hit / I	-

Table 5: Sample accelerator L1 for use with two-level accelerator cache.

	L1 Requests			L1 Responses		XG Request	XG Responses			L2 Events		
	GetM	GetS	PutM	InvAck	Writeback	Invalidate	DataM	DataS	WBAck	All Acks	L2 Repl.	L2 Repl. Clean
I	<i>send GetM / IM</i>	<i>send GetS / IS</i>	-	-	-	<i>send InvAck</i>	-	-	-	-	-	-
S	<i>send Inv to sharers / SM</i>	<i>send DataS</i>	-	-	-	<i>send Inv to sharers / SI</i>	-	-	-	-	-	<i>send Inv to sharers / SR</i>
MO	<i>send DataM / M</i>	<i>send DataS / MOS</i>	-	-	-	<i>send WB / I</i>	-	-	-	-	<i>send PutM / MRI</i>	-
M	<i>send Inv to owner / MM</i>	<i>send Inv to owner / MMOS</i>	<i>send WBAck / MO</i>	-	-	<i>send Inv to owner / MI</i>	-	-	-	-	<i>send Inv to owner / MR</i>	-
MOS	<i>send Inv to sharers / MOSM</i>	<i>send DataS</i>	-	-	-	<i>send Inv to sharers / MOSI</i>	-	-	-	-	<i>send Inv to sharers / MOSR</i>	<i>send Inv to sharers / ER</i>
IM	<i>recycle</i>	<i>recycle</i>	-	-	-	<i>send InvAck</i>	<i>send DataM / M</i>	-	-	-	/stall	/stall
IS	<i>recycle</i>	record	-	—	-	<i>send InvAck</i>	<i>send DataS / MOS</i>	<i>send DataS / S</i>	-	-	/stall	/stall
SM	<i>recycle</i>	<i>recycle</i>	-	incAcks	-	<i>recycle</i>	-	-	-	<i>send GetM / IM</i>	/stall	/stall
SI	<i>stall</i>	<i>stall</i>	-	incAcks	-	-	-	-	-	<i>send InvAck / I</i>	-	-
SR	<i>stall</i>	<i>stall</i>	-	incAcks	-	/ SI	-	-	-	<i>send PutS / SRI</i>	-	-
SRI	<i>stall</i>	<i>stall</i>	-	-	-	<i>send InvAck</i>	-	-	/ I	-	-	-
MR	<i>stall</i>	<i>stall</i>	<i>send WBAck / MRi</i>	<i>send Inv to owner</i>	<i>send PutM / MRI</i>	/ MI	-	-	-	-	-	-
MRI	<i>stall</i>	<i>stall</i>	-	-	-	<i>send InvAck</i>	-	-	/ I	-	-	-
MI	<i>stall</i>	<i>stall</i>	<i>send WBAck / Mli</i>	<i>send Inv to owner</i>	<i>send WB / I</i>	-	-	-	-	-	-	-
MOSI	<i>stall</i>	<i>stall</i>	-	incAcks	-	-	-	-	-	<i>send WB / I</i>	-	-
MOSR	<i>stall</i>	<i>stall</i>	-	incAcks	-	/ MOSI	-	-	-	<i>send PutM / MRI</i>	-	-
ER	<i>stall</i>	<i>stall</i>	-	incAcks	-	/ MOSI	-	-	-	<i>send PutE / MRI</i>	-	-
MM	<i>recycle</i>	<i>recycle</i>	<i>send WBAck / MMi</i>	<i>send Inv to owner</i>	<i>send WBAck send DataM / M</i>	<i>recycle</i>	-	-	-	-	<i>stall</i>	<i>stall</i>
MMOS	<i>recycle</i>	<i>recycle</i>	<i>send WBAck / MMOSi</i>	<i>send Inv to owner</i>	<i>send WBAck send DataS / MOS</i>	<i>recycle</i>	-	-	-	-	<i>stall</i>	<i>stall</i>
MOSM	<i>recycle</i>	<i>recycle</i>	-	incAcks	-	<i>recycle</i>	-	-	-	<i>send DataM / M</i>	<i>stall</i>	<i>stall</i>
Mli	<i>stall</i>	<i>stall</i>	-	<i>send WB / I</i>	-	-	-	-	-	-	-	-
MRi	<i>stall</i>	<i>stall</i>	-	<i>send PutM / MRI</i>	-	/ Mli	-	-	-	-	-	-
MER	<i>stall</i>	<i>stall</i>	-	<i>send DataM / M</i>	-	<i>recycle</i>	-	-	-	-	<i>stall</i>	<i>stall</i>