# CMI Slave bridge

|  |  |  |
| --- | --- | --- |
| Ver | Date | History |
| 0.0 | 12 Apr 2019 | 1. Initial Version |
| 0.1 | 15 Apr 2019 | 1. Reduced Response channel’s sideband width by 1-bit. |
| 1.0 | 16 Apr 2019 | 1. Updated block diagram to support more than 2 VCs. |
| 1.1 | 26 Apr 2019 | 1. Moved route lookup to before req FIFOs. 2. Added input FIFO for shared rsp and rdcpl credits. |
| 1.2 | 19 Jun 2019 | 1. Updated feature list. |
| 1.3 | 20 Jun 2019 | 1. Added note on using MS-bit of loopback info fields by bridges. |
| 1.4 | 26 Jul 2019 | 1. Add bubble avoidance FIFO for Rdcpl in block diagram. 2. Remove fab credit FIFO and parity check/create from block diagram as they are planned for post 1907. |
| 1.5 | 05 Nov 2019 | 1. Updated block diagram to reflect RSSB supported configuration. |

## Features:

Legend: Not supported yet. Not supported.

* CMI Spec 1.1.
* Supported Opcodes: MRd, MWr, MWrPtl, MPCmt, NDTC.
* Configurable VCs (1, 2-4).
* Cross mapping of CMI VCs between Master and Slave bridges.
* Parameterized FIFO depths for credit FIFOs. Configurable register value to exercise a depth lesser than hardware configured config depth.
* ~~Abort response (even if set\_srsp=0) for route lookup error, or address parity error~~. Host is expected to not send a transaction that can cause route lookup or address parity error.
* Tunnelling – Supported only if *req\_no\_addr*=0.
* 32B/64B request lengths.
* Route lookup based on either request address or destination Id (Configurable by NoC) followed by address hashing.
* Interface width = 8/16/32B.
* WDATA\_DELAY – default, fixed. Variables delay modes are for clock crossing between host and bridge, so verification not targeted for the same.
* RDATA\_DELAY – default, fixed Variables delay modes are for clock crossing between host and bridge, so verification not targeted for the same.
* Fabric credits –
  + No shared Fab credit FIFOs, so zero fab credits published to host.
  + Non-zero fab credits published to host agent only for async mode (host and bridge on async clocks).
  + Can receive and honor fab credits published by host agent.
* Unordered fabric. So all write requests have responses. (set\_srsp=0 not supported for writes).
* No interleaving of read completion data.
  + Interleaving of read completion data.
* Parity
  + Parity generation before sending transaction to the receiving host (irrespective of whether it is being sent by the transmitting host).
  + Error response if *req\_address\_parity* error.
  + Parity check and reporting for all fields.
* Data forward – Not supported.
* Dual data mode – Not supported.
  + On the secondary channel, there is no request associated with data and responses. So for wr\_data, there is no information on where to route this data since such information will be available in req channel. In that cases, the secondary request channel’s master bridge might need to get such information from primary request channel’s master bridge.
* Power management –
  + Q-channel or any power related interface.
* Upstream stall acknowledgement.
* ~~Error response if~~ *~~req\_address\_parity~~* ~~error~~.
* Out-of-band QoS support.
* Meta data forwarding.
* Clock crossing.
  + On NoC Side – Bridge and router are on different clocks.
  + CMI Interface side – Bridge and host are on different clocks.
* Early valid signals – Assert few cycles before valid signals, de-assert few cycles after valid signals de-assert.
* Loopback info.
* For below optional CMI features, no processing/generation of the related fields are done in the bridges, but they are forwarded through the bridges.
  + Tunnelling.
  + Speculative read.
  + Near Memory/Far memory
  + clos
  + Directory state.
  + ~~ECC~~.
  + Poison
  + Mirroring.

## Mapping CMI channels to NoC channels

Please refer to ***CMI Master bridge uArch*** document.

## Block diagram



**Problem**:

If route lookup is done after VC arbitration, choke up of req channel can happen due to lack of the selected VC’s credit at TX switch’s input (from router).

Solution#1 (Chosen post 1904): Storage FIFO VC (CMI VC) arbitration to be done after address decoding, and the arbitration to take into account VC (NoC VC) credit availability (from router to switch). Drawback. Increased FIFO width to store decoded address output. But since the FIFO depths will be for small transactions (<10), its fine. But current switches don’t provide VC (NoC VC) credit availability for each VC (NoC VC) of its interface. Switches to be updated. Till then, we can let the choke up of channel~~, or when valid is provided to a switch for a VC, if the grant is not available from the VC, arbitration can move to the next VC~~.

Solution#2 (Not Chosen): FIFO VC arbitration done before address decoding. But VC credit availability (from router to switch) is used for VC arbitration. But current switches don’t provide VC credit availability for each VC of its interface. Switches to be updated.

Solution#3 (Chosen for 1904): Connect each CMI VC to separate switch interface. Limits max VCs support to 2. Chosen till VCs can provide VC credit availability to host side.

## NoC Channel packet widths

Please refer to ***CMI Master bridge uArch*** document.

## Bridge props in NoC Studio

### CMI parameters

Please refer to ***CMI Master bridge uArch*** document.

### Port-en parameters

Please refer to ***CMI Master bridge uArch*** document

## Rsp and Rdcpl channel FIFOs

* Separate FIFOs per VC for each of the Rsp and Rdcpl channels.
* NoC Studio configurable FIFO depths for each instantiated FIFOs. Additional programmable register limits (should be ≤ NoC studio configured depth) the FIFO depth to be used.
* TBD. If clock crossing between slave host and slave bridge, a shared FIFO (fab credit FIFO) for each of rsp and rd\_cpl channels is present at the CMI interface boundary. The FIFO depth programmable by NoCStudio. Credits relevant to this FIFO are exchanged as *rsp\_fab\_credit*s and *rd\_cpl\_fab\_credits*.
* If no clock crossing between slave host and slave bridge, no shared FIFO within rsp and rdcpl channels, hence no fab credits are published for these channels.
* *WR\_DELAY* support.
  + rd\_cpl\_data arrives RD\_DELAY cycles (default or fixed delay mode) after rd\_cpl. Slave bridge aligns these two channels before sending to Noc. The Master bridge is expected to re-align these two CMI channels before sending to CMI interface.
  + Valid outs for rd\_cpl\_fifo and for first flit of rd\_cpl\_data\_fifo is available only if both the FIFOs are not-empty.
  + TBD. To support an offset between rd\_cpl and rd\_cpl\_Data along the NoC link, switches and routers may be modified. Then there will be separate <NOC\_RD\_DELAY> parameter for the NOC. FIFO read in this case must ensure that rd\_cpl and rd\_cpl\_data are offset by this <NOC\_RD\_DELAY> before sending out on the switches.



* + The Tx switch can provide a grant on its host side interface’s output only if the host side has a valid data to be fed to the switch. So the FIFOs are configured to prefetch an entry. An entry is popped out of FIFO only if grant is available from the downstream.

## Route lookup

* Slave bridge uses Id lookup to identify route to the destination (Master bridge).
* Responses or read completions for the same source ID may take a different route on the NoC. So route lookup may also include CMI VC. The CMI VC input to route look up also helps map CMI VC to a NoC VC.

## Packetization

Please refer to ***CMI Master bridge uArch*** document.

## Req channel

* *WR\_DELAY* support.
  + Slave bridge expects alignment between incoming NoC packets for Wr req and the first packet of Wr data.
  + Slave bridge delays Wr data for WR\_DELAY number of cycles before sending out to Slave agent.

### Bubble avoidance FIFO for req data

CMI spec disallows any bubble within 64B of req\_data sent on CMI interface if no async crossing between connected CMI agents (host and bridge). So a FIFO is instantiated for the purpose which accumulates 64B (32B if req\_length=1) of req\_data before sending over CMI interface.

## RX path Arbitration

Read and Write requests are separate NoC Channels, which are arbitered and sent to CMI request channel. Read request takes a higher priority over Write requests.

## ISM and flow control

Slave bridge implementation follows CMI spec 1.1 (ISM requester within Section 4) recommended timing diagrams for Credit initialization state machine.

Slave bridge publishes fabric credits if req fab credit FIFO is of non-zero depth. Else Slave bridge doesn’t publish fabric credits to the Slave CMI agent, during ism initialization and during active transactions. For credits received from Slave CMI agent, the Slave CMI agent may or may not send fabric credits to the Slave bridge. If the Slave CMI agent doesn’t support fab credits, then Slave agent is expected to tie 0 to the fabric credit put signals driven from Slave CMI agent to Slave CMI bridge.

### CFG BYPASS for ISM INIT

Please refer to ***CMI Master bridge uArch*** document.

### Credit re-initialization

Please refer to ***CMI Master bridge uArch*** document.

### CHANCT=1 support

CMI Spec allows a values of 2-4 for CHANCT parameter. In addition, the current Slave bridge also supports CHANCT=1. If CHANCT=1, FIFOs and most of the logic related to VC1 will not be instantiated. For response and read completion channels, credits are expected for VC0 within first 8 cycles. For request channels, credits are published for VC0 in first 8 cycles. Queue depth credits are published in the next 8 cycles. Since only 1 VC is present if CHANCT=1, queue depths credit can also be ignored by the slave agent as the values are identical to the credits for VC0.