NetSpeed IMG4 Master Bridge Design

May 19, 2015

NetSpeed IMG4 Master Bridge Design

About This Document

This document describes the NetSpeed IMG4 master bridge design implementation. Using NocStudio, users can define NoC architectures, describe specifications and requirements, optimize the NoC design and generate one or multiple IMG4 master bridges within the NoC design that interface with IMG4 masters.

Audience

This document is intended for users of NocStudio:

* NoC Architects
* NoC Designers
* SoC Architects

Related Documents

The following documents can be used as a reference to this document.

* NetSpeed NocStudio User Manual

Related Documents

The following documents can be used as a reference to this document.

* NetSpeed NocStudio Orion User Manual
* NetSpeed Orion Physical Design Guidelines

Customer Support

For technical support about this product, please contact [support@netspeedsystems.com](mailto:support@netspeedsystems.com)

For general information about NetSpeed products refer to: [www.netspeedsystems.com](http://www.netspeedsystems.com)

Contents

[About This Document 2](#_Toc419792297)

[Audience 2](#_Toc419792298)

[Related Documents 2](#_Toc419792299)

[Related Documents 2](#_Toc419792300)

[Customer Support 2](#_Toc419792301)

[1 Introduction 6](#_Toc419792302)

[2 Design Implementation 7](#_Toc419792303)

[3 Design Assumptions 15](#_Toc419792304)

List of Figures

[Figure 1: AR channel barrier FSM 11](#_Toc419751832)

[Figure 2: AW channel barrier FSM 13](#_Toc419751833)

[Figure 3: Top level IMG4 master bridge diagram 14](#_Toc419751834)

List of Tables

[Table 1: cmd\_cache, cmd\_cache\_snoop mapping to AxCACHE, AxSNOOP, AxDOMAIN 8](#_Toc419751608)

# Introduction

The IMG4 protocol is close to the AXI4 protocol. The AXI4 master bridge design will be used as the basis of the IMG4 master bridge implementation, along with a few extra knobs and parameters.

The primary difference between the two protocols is that the IMG4 protocol has a single bus for both reads and writes. A “read-not-write” bit indicates whether a transaction is a read or a write.

# Design Implementation

The IMG4 master bridge implementation strives to add no additional latency than already present in the AXI4 master bridge. A top file “ns\_img4mstrbrdg.v” will be added to the AXI4 master bridge directory: hw/ns\_acemstrbrdg/rtl/

1. Burst type encoding is different in AXI. Map IMG4 to AXI.
   1. AXI: 0b00 FIXED, 0b01 INCR, 0b10 WRAP, 0b11 Reserved
   2. IMG4: b’00 : Incrementing, b’01 : Fixed, b’10 : Wrapping, b’11 : Reserved

The register in, register out policy of the IMG4 spec requires that these muxes are placed after the input fifos. These are placed in the mprt\_AR\*\_int pipeline.

1. Logic within “ns\_acemstrbrdg\_core.v” requires WLAST signal. ns\_img4mstrbrdg.v will contain logic that creates this signal based on cmd\_burst\_length[3:0]. Since cmd could be far ahead of data, the burst length needs to be stored in a fifo. The depth of this fifo will be 2 + depth of the input fifo. (The depth of the “input fifo” in case of ratio clock crossers is 2). Additionally, there will be a counter that increments on every wdata\_valid & wdata\_enable, and resets when this value is equal to that in the fifo (assert WLAST at this time).
2. The <port>\_cmd\_cache, <port>\_cmd\_cache\_snoop IMG4 signals need to be mapped to the equivalent AxCACHE, AxSNOOP, AxDOMAIN signals. The following table is a proposal to the mapping.

Table 1: cmd\_cache, cmd\_cache\_snoop mapping to AxCACHE, AxSNOOP, AxDOMAIN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IMG4 | IMG4 Description | AXI4 equivalent | AXI4 equivalent when <port>\_cmd\_cache\_snoop = 1'b0 | AXI4 equivalent when <port>\_cmd\_cache\_snoop = 1'b1 |
| <port>\_cmd\_cache = 2'b00 | Do not cache - 00 : The data should not be cached On writes, this would typically be used for data that is not going to be read by that master or not likely to be read by another master while still in the cache. On reads, this would typically indicate that the data will not be reread by the master. | AxCACHE = 4'b000x | AxDOMAIN = 2 'b11 | AxDOMAIN = 2'b10 |
| <port>\_cmd\_cache = 2'b01 | Write through - 01 : Cache (write-through) On a write, it is recommended that the data is cached and that the data should be written to system memory as soon as possible. On a read, it indicates that the data may be re-read and should be cached. | AxCACHE = 4'b1110 | AxDOMAIN = 2 'b11 | AxDOMAIN = 2'b10 |
| <port>\_cmd\_cache = 2'b01 | Write back - 10 : Cache (write-back) On a write, it is recommended that the data is not stored to system memory until it is evicted or invalidated. This is useful for masters that use the cache as a temporary buffer. On a read, it indicates that the data may be re-read and should be cached. The eviction or invalidation is system specific but a fence will require that all data written by the master that issued the flush be evicted to system memory. | AxCACHE = 4'b1111 | AxDOMAIN = 2 'b11 | AxDOMAIN = 2'b10 |
| <port>\_cmd\_cache = 2'b11 | Read-don't allocate - 11 : Cache (read-don’t allocate) This value is not defined for writes and should not be used. For reads it indicates that the cache may contain the required data from a previous read, but if it does not, then do not allocate space in the cache as the data will not be re-read. | AxCACHE = 4'b1010 | AxDOMAIN = 2 'b11 | AxDOMAIN = 2'b10 |

1. The <port>\_cmd\_priority signal is a single-bit signal indicating two possible levels of priority. This will be connected to AxQOS[0]. AxQOS[3:1] will be tied off to 3’b000.
2. The <port>\_cmd\_security signal will be connected to AxPROT[1]. AxPROT[0] and AxPROT[2] will be tied off to 0.
3. <port>\_cmd\_int\_tag – description is “Identifies which of the internal master of the IP are making this request. This can be used for treating the requests differently for each internal master. The values used are specific to each IP.”
   1. This can potentially be used to map cmd\_priority -> AxQOS differently for different values of cmd\_int\_tag.
   2. It could also be used to concatenate to cmd\_tag. AxID = {cmd\_int\_tag, cmd\_tag}.
   3. Currently, this will not be used.
4. <port>\_wresp\_error. This indicates an exclusive access failure. In the P\_IMG4\_MODE=1 mode, a bit is added to the widtbl to store the <port>\_cmd\_exclusive bit. On return of the write response, this bit is checked against a non-EXOKAY status and asserted accordingly.
5. <port>\_rdata\_error. This indicates an exclusive access failure. In the P\_IMG4\_MODE=1 mode, a bit is added to the ridtbl to store the <port>\_cmd\_exclusive bit. On return of the write response, this bit is checked against a non-EXOKAY status and asserted accordingly.
6. <port>\_cmd\_fence. This is an in-band signal indicating a read or a write barrier. According to Section 4.6 of the IMG4 spec:

“On a write transaction, the write response should only be returned once all outstanding write transactions have been written to the destination. This is typically used as an indicator that all data is written to memory before signalling an event indicating that state.

On a read transaction, the fence bit indicates all previously issued read transactions should be completed before returning the read data for this transaction.

It is not a requirement to prevent any future accesses completing before the fence is completed.

It is a requirement of the bus fabric to implement the requirements of the fence including ensuring that any data in caches from this master are written to memory. The flushing of caches is not required if alltransfers from all masters go through that cache such that the cache will always accurately reflect the intended contents of the system memory.”

The AMBA bridge has logic that deals with AMBA barriers. These have the below states.

ACE

IMG4

1

7

8

2

4

5

10

3

6

9

11

Figure 1: AR channel barrier FSM

Arcs for ACE as numbered:

1. Detection of an AR memory barrier, inner or outer shareable, and a barrier grant from the AW channel.
2. If 1 does not occur, the detection of an AR synchronization barrier, and a barrier grant from the AW channel.
3. If both 1 and 2 do not occur, wait here.
4. No outstanding regular responses (~ridtbl\_outstd\_found & ~widtbl\_outstd\_found) and no outstanding read barrier responses.
5. If 4 does not occur, and there are no outstanding regular read and write responses, but there is an outstanding read barrier response.
6. If both 4 and 5 do not occur, wait here.
7. If rdwr\_bar\_resp\_rcvd==1 and there are no outstanding barrier responses.
8. If rdwr\_bar\_resp\_rcvd==1 and there is an outstanding barrier response.
9. If both 7 and 8 do not occur, wait here.
10. If there is no outstanding barrier response.
11. If 10 does not occur, wait here.

Arcs for IMG4 as numbered:

1. Illegal path. Read and write fences are not coupled, so we cannot go to WAIT\_FOR\_RW\_BAR\_RESP.
2. If either bit in ARBAR[1:0] is set.
3. If 2 does not occur.
4. No outstanding read responses and no outstanding barrier responses.
5. If 4 does not occur, and there are no outstanding regular read responses, but there is an outstanding barrier response.
6. If both 4 and 5 do not occur, wait here.
7. Illegal state.
8. Illegal state.
9. Illegal state.
10. If there is no outstanding barrier response.
11. If 10 does not occur, wait here.

2

4

5

10

3

6

9

11

1

8

IMG4

ACE

7

Figure : AW channel barrier FSM

Arcs for ACE as numbered:

1. Detection of an AR memory barrier, inner or outer shareable, and a non-zero count of AW barriers.
2. If 1 does not occur, and an AR synchronization barrier is detected along with a non-zero count of AW barriers.
3. If 1 and 2 do not occur, wait here.
4. If there are no outstanding regular read and write responses, and there is no outstanding write barrier response.
5. If 4 does not occur, and there are no outstanding regular read and write responses, but there is an outstanding write barrier response.
6. If 4 and 5 do not occur, wait here.
7. If rdwr\_bar\_resp\_rcvd==1, and there is no outstanding write barrier response.
8. If rdwr\_bar\_resp\_rcvd==1, and there is an outstanding write barrier response.
9. If 7 and 8 do not occur, wait here.
10. If there is no outstanding barrier write response.
11. If 10 does not occur, wait here.

Arcs for IMG4 as numbered:

1. Illegal path. Read and write fences are not coupled, so we cannot go to WAIT\_FOR\_RW\_BAR\_RESP.
2. If either bit in AWBAR[1:0] is set.
3. If 2 does not occur, wait here.
4. If there are no outstanding regular write responses, and there is no outstanding write barrier response.
5. If 4 does not occur, and there are no outstanding regular write responses, but there is an outstanding write barrier response.
6. If 4 and 5 do not occur, wait here.
7. Illegal state.
8. Illegal state.
9. Illegal state.
10. If there is no outstanding barrier write response.
11. If 10 does not occur, wait here.

ns\_img4mstrbrdg.v

Set P\_IMG4\_MODE=1’b1 for barrier FSMs, cmd\_cache/cmd\_cache\_snoop logic.

Burst\_length fifo and logic for WLAST generation

AXI4 host Interface

IMG4 host Interface

ns\_acemstrbrdg\_core.v

Noc Interface

Figure : Top level IMG4 master bridge diagram

# Design Assumptions

1. Functionality for out-of-band signals <port>\_request\_priority and <port>\_limited\_throughput will not be implemented.
2. Narrows will not be supported. This was confirmed with the IMG team.
3. Error responses such as decode errors, slave errors will be converted to OKAY responses since the interface does not have a bit to indicate these (error signal is single bit and is used for exclusive status). This was confirmed with the IMG team.
4. Optional interface signals are present on the command interface. These will all be exposed at the ns\_img4mstrbrdg module level.
   * cmd\_burst\_type
   * cmd\_cache
   * cmd\_cache\_snoop
   * cmd\_fence
   * cmd\_exclusive
   * cmd\_priority
   * cmd\_security
   * cmd\_int\_tag
   * wresp\_enable
   * wresp\_error
   * rdata\_error
   * request\_priority
   * limited\_throughput

2670 Seely Avenue

Building 11

San Jose CA 95134

[www.netspeedsytems.com](file:///C:\Users\AE1\Downloads\www.netspeedsytems.com)