AHB-Lite Interconnect Module - Lattice Propel Builder

User Guide

FPGA-IPUG-02051-1.2

May 2021

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
AHB	Advanced High-performance Bus
AMBA	Advanced Microcontroller Bus Architecture
FPGA	Field Programmable Gate Array
RTL	Register Transfer Level

1. Introduction

This document provides technical information about AHB-Lite Interconnect Module and aims to provide information essential for IP/system developers, verification, and software for integration, testing, and validation.

In general, this document covers design specification from RTL to IP packaging and details the procedures for IP generation and integration.

The design is implemented in Verilog HDL. The IP can be configured and based on Table 1.1.

Table 1.1. FPGA Software for IP Configuration, Generation, and Implementation

Supported FPGA Family	IP Configuration and Generation	IP Implementation (Synthesis, Map, Place and Route)
MachXO2™	Lattice Propel™ Builder software	Lattice Diamond® software
MachXO3™	Lattice Propel Builder software	Lattice Diamond software
MachXO3D™	Lattice Propel Builder software	Lattice Diamond software
Mach™-NX	Lattice Propel Builder software	Lattice Diamond software
CrossLink™-NX	Lattice Propel Builder software	Lattice Radiant™ software
Certus™-NX	Lattice Propel Builder software	Lattice Radiant software

1.1. Features

The key features of the AHB-Lite Interconnect Module include:

- Compliance with AMBA 3 AHB-Lite protocol
- Fully parameterized design
- Data bus width of up to 1024 bits [8, 16, 32, 64, 128, 256, 512, 1024]
- Address width of up to 32-bits [11,12,...,32]
- Support of up to 32 masters and 32 slaves
- Full or Sparse connection between masters and slaves
- Slave port address decoding
- Support of fragmented address space of up to eight fragments per slave
- Slave side arbitration
- Selectable arbitration scheme:
  - Round robin
  - Fixed priority

2. Functional Description

2.1. Overview

The Lattice Semiconductor AHB-Lite Interconnect Module is a fully parameterized soft IP, high performance, low latency interconnect fabric for AMBA 3 AHB-Lite based systems, enabling one or more masters to be connected to one or more slaves.

AHB-Lite, defined in the AMBA 3 protocol, is a subset of the full AHB specification for use in designs where only a single bus master is used. AHB-Lite also does not include signals from the original AHB: request and grant. In addition, there is no support for a split or retry response. The only responses are okay or error.

AHB-Lite implements the features required for high-performance, high clock frequency systems including:

- Burst transfers: one address phase followed by multiple data phases, where each data phase is called a beat. 4/8/16-beat wrapping and incrementing bursts are supported. Incrementing bursts of undefined length are also supported.
- Single-clock edge operation
- Non-tristate implementation
- Wide data bus configurations: 64, 128, 256, 512, or 1024 bits wide.
  Note that the following data bus widths are also supported: 8, 16, 32 bits wide

Refer to AMBA 3 AHB-Lite Protocol v1.0 Specifications for more information on the protocol.

Even though AHB-Lite is a single master bus protocol, AHB-Lite Interconnect Module enables the connection of multiple masters to one or more slaves by implementing multi-layer interconnect. Each master is considered to be on its own layer and isolated from each other, but can share access to the slaves. Each master can access different slaves in parallel. When more than one master tries to access the same slave, slave-side arbitration is performed by the multi-layer interconnect.

2.2. Signal Description

Figure 2.1 shows the interface diagram for the AHB-Lite Interconnect Module. The diagram shows all of the available ports for the IP core.

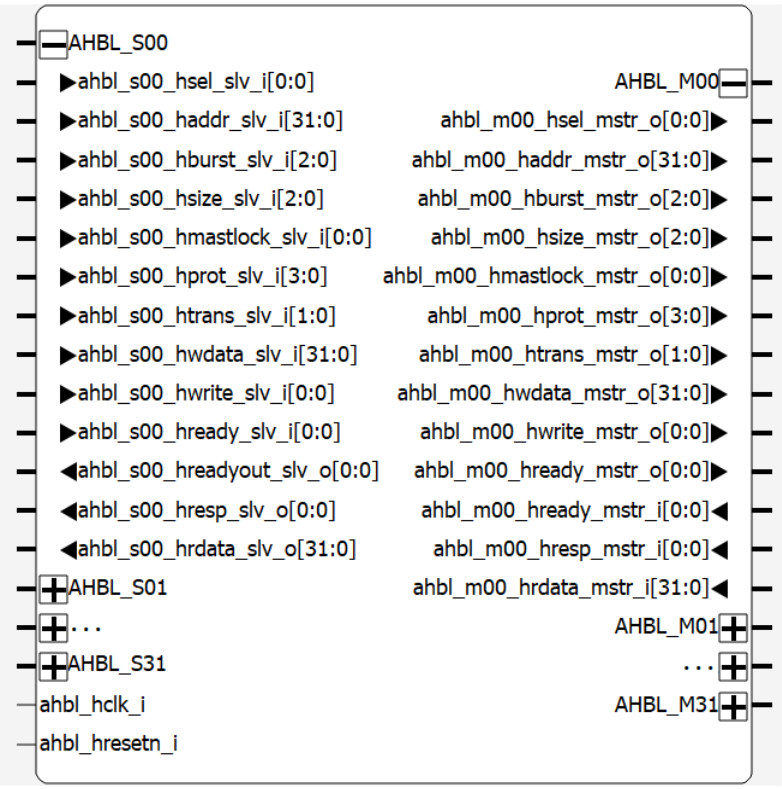

Figure 2.1. AHB-Lite Interconnect Module Interface Diagram

Table 2.1. AHB-Lite Interconnect Module Signal Description

Pin Name	Direction	Width (Bits)	Description
Clock and Reset			
ahbl_hclk_i	In	1	AHB-Lite clock
ahbl_hresetn_i	In	1	AHB-Lite active LOW reset
AHB-Lite Slave Interface 00 (AHBL_M00)			
ahbl_m00_hsel_mstr_o	Out	1	Slave select
ahbl_m00_haddr_mstr_o	Out	M_ADDR_WIDTH	Address
ahbl_m00_hburst_mstr_o	Out	3	Burst type (SINGLE, INCR, INCR4/8/16)
ahbl_m00_hsize_mstr_o	Out	3	Transfer size (8/16/32/64/128/256/512/1024)
ahbl_m00_hmastlock_mstr_o	Out	1	Current transfer is part of a locked transfer.
ahbl_m00_hprot_mstr_o	Out	4	Protection control
ahbl_m00_htrans_mstr_o	Out	2	Transfer type of the current transfer (IDLE/BUSY/NSEQ/SEQ)
ahbl_m00_hwdata_mstr_o	Out	DATA_WIDTH	Write data bus
ahbl_m00_hwrite_mstr_o	Out	1	Write = 1, Read = 0
ahbl_m00_hready_mstr_o	Out	1	Indicates transfer completion.
ahbl_m00_hready_mstr_i	In	1	Indicates transfer completion. This signal is driven low by the slave to extend the transfer.
ahbl_m00_hresp_mstr_i	In	1	Slave response (OKAY/ERROR)
ahbl_m00_hrdata_mstr_i	In	DATA_WIDTH	Read data bus

Pin Name	Direction	Width (Bits)	Description
AHB-Lite Slave Interface 01 (AHBL_M01)			
ahbl_m01_hsel_mstr_o	Out	1	Slave select
ahbl_m01_haddr_mstr_o	Out	M_ADDR_WIDTH	Address
ahbl_m01_hburst_mstr_o	Out	3	Burst type (SINGLE, INCR, INCR4/8/16)
ahbl_m01_hsize_mstr_o	Out	3	Transfer size (8/16/32/64/128/256/512/1024)
ahbl_m01_hmastlock_mstr_o	Out	1	Current transfer is part of a locked transfer
ahbl_m01_hprot_mstr_o	Out	4	Protection control
ahbl_m01_htrans_mstr_o	Out	2	Transfer type of the current transfer (IDLE/BUSY/NSEQ/SEQ)
ahbl_m01_hwdata_mstr_o	Out	DATA_WIDTH	Write data bus
ahbl_m01_hwrite_mstr_o	Out	1	Write = 1, Read = 0
ahbl_m01_hready_mstr_o	Out	1	Indicates transfer completion.
ahbl_m01_hready_mstr_i	In	1	Indicates transfer completion. This signal is driven low by the slave to extend the transfer.
ahbl_m01_hresp_mstr_i	In	1	Slave response (OKAY/ERROR)
ahbl_m01_hrdata_mstr_i	In	DATA_WIDTH	Read data bus
…	—	—	—
AHB-Lite Slave Interface xx (AHBL_Mxx)			
ahbl_mxx_hsel_mstr_o	Out	1	Slave select
ahbl_mxx_haddr_mstr_o	Out	M_ADDR_WIDTH	Address
ahbl_mxx_hburst_mstr_o	Out	3	Burst type (SINGLE, INCR, INCR4/8/16)
ahbl_mxx_hsize_mstr_o	Out	3	Transfer size (8/16/32/64/128/256/512/1024)
ahbl_mxx_hmastlock_mstr_o	Out	1	Current transfer is part of a locked transfer.
ahbl_mxx_hprot_mstr_o	Out	4	Protection control
ahbl_mxx_htrans_mstr_o	Out	2	Transfer type of the current transfer (IDLE/BUSY/NSEQ/SEQ)
ahbl_mxx_hwdata_mstr_o	Out	DATA_WIDTH	Write data bus
ahbl_mxx_hwrite_mstr_o	Out	1	Write = 1, Read = 0
ahbl_mxx_hready_mstr_o	Out	1	Indicates transfer completion.
ahbl_mxx_hready_mstr_i	In	1	Indicates transfer completion. This signal is driven low by the slave to extend the transfer.
ahbl_mxx_hresp_mstr_i	In	1	Slave response (OKAY/ERROR)
ahbl_mxx_hrdata_mstr_i	In	DATA_WIDTH	Read data bus
AHB-Lite Master Interface 00 (AHBL_S00)			
ahbl_s00_hsel_slv_i	In	1	Slave select
ahbl_s00_haddr_slv_i	In	M_ADDR_WIDTH	Address
ahbl_s00_hburst_slv_i	In	3	Burst type (SINGLE, INCR, INCR4/8/16).
ahbl_s00_hsize_slv_i	In	3	Transfer size (8/16/32/64/128/256/512/1024).
ahbl_s00_hmastlock_slv_i	In	1	Current transfer is part of a locked transfer.
ahbl_s00_hprot_slv_i	In	4	Protection control.
ahbl_s00_htrans_slv_i	In	2	Transfer type of the current transfer (IDLE/BUSY/NSEQ/SEQ).
ahbl_s00_hwdata_slv_i	In	DATA_WIDTH	Write data bus.
ahbl_s00_hwrite_slv_i	In	1	Write = 1, Read = 0
ahbl_s00_hready_slv_i	In	1	Indicates transfer completion.
ahbl_s00_hreadyout_slv_o	Out	1	Indicates transfer completion. This signal is driven low by the slave to extend the transfer.
ahbl_s00_hresp_slv_o	Out	1	Slave response (OKAY/ERROR)
ahbl_s00_hrdata_slv_o	Out	DATA_WIDTH	Read data bus

Pin Name	Direction	Width (Bits)	Description
AHB-Lite Master Interface 01 (AHBL_S01)			
ahbl_s01_hsel_slv_i	In	1	Slave select
ahbl_s01_haddr_slv_i	In	M_ADDR_WIDTH	Address
ahbl_s01_hburst_slv_i	In	3	Burst type (SINGLE, INCR, INCR4/8/16)
ahbl_s01_hsize_slv_i	In	3	Transfer size (8/16/32/64/128/256/512/1024)
ahbl_s01_hmastlock_slv_i	In	1	Current transfer is part of a locked transfer
ahbl_s01_hprot_slv_i	In	4	Protection control
ahbl_s01_htrans_slv_i	In	2	Transfer type of the current transfer (IDLE/BUSY/NSEQ/SEQ)
ahbl_s01_hwdata_slv_i	In	DATA_WIDTH	Write data bus
ahbl_s01_hwrite_slv_i	In	1	Write = 1, Read = 0
ahbl_s01_hready_slv_i	In	1	Indicates transfer completion
ahbl_s01_hreadyout_slv_o	Out	1	Indicates transfer completion. This signal is driven low by the slave to extend the transfer.
ahbl_s01_hresp_slv_o	Out	1	Slave response (OKAY/ERROR)
ahbl_s01_hrdata_slv_o	Out	DATA_WIDTH	Read data bus
…	—	—	—
AHB-Lite Master Interface yy (AHBL_Syy)			
ahbl_syy_hsel_slv_i	In	1	Slave select
ahbl_syy_haddr_slv_i	In	M_ADDR_WIDTH	Address
ahbl_syy_hburst_slv_i	In	3	Burst type (SINGLE, INCR, INCR4/8/16)
ahbl_syy_hsize_slv_i	In	3	Transfer size (8/16/32/64/128/256/512/1024)
ahbl_syy_hmastlock_slv_i	In	1	Current transfer is part of a locked transfer
ahbl_syy_hprot_slv_i	In	4	Protection control
ahbl_syy_htrans_slv_i	In	2	Transfer type of the current transfer (IDLE/BUSY/NSEQ/SEQ)
ahbl_syy_hwdata_slv_i	In	DATA_WIDTH	Write data bus
ahbl_syy_hwrite_slv_i	In	1	Write = 1, Read = 0
ahbl_syy_hready_slv_i	In	1	Indicates transfer completion.
ahbl_syy_hreadyout_slv_o	Out	1	Indicates transfer completion. This signal is driven low by the slave to extend the transfer.
ahbl_syy_hresp_slv_o	Out	1	Slave response (OKAY/ERROR)
ahbl_syy_hrdata_slv_o	Out	DATA_WIDTH	Read data bus

Notes:

- xx – Possible values are (02, 03,…, *Total AHB-Lite Slaves* -1).
- yy – Possible values are (02, 03,…, *Total AHB-Lite Masters* -1).
- All AHB-Lite Master/Slave Interface are compliant with AHB-Lite protocol. Refer to AMBA 3 AHB-Lite Protocol v1.0 Specifications for the timing diagrams and for more information on the protocol.

2.3. Attributes Summary

Table 2.2 provides the list of user-configurable attributes for the AHB-Lite Interconnect Module. The attribute values are specified using the IP core Configuration user interface in the Propel Builder software as shown in Table 2.2.

Table 2.2. Attributes Table

Attribute	Selectable Values	Default	Dependency on Other Attributes[4]
General Tab			
General Group			
Total AHB-Lite Masters	1–32	2	*Total AHB-Lite Masters* and *Total AHB-Lite Slaves* cannot be both 1
Total AHB-Lite Slaves	1–32	2	
Master Address Width(bits) (M_ADDR_WIDTH)	11–32	32	—
Full Address Decoding up to 1kB[1]	Checked, Unchecked	Unchecked	—
Data Bus Width(bits) (DATA_WIDTH)	8, 16, 32, 64, 128, 256, 512, 1024	32	—
Registered Output[2]	Checked, Unchecked	Unchecked	—
Master 0 Connection Setting Group			
Master 0 Slave 0 Connect Enable	Checked, Unchecked	Checked	en_m0 and en_s0
Master 0 Slave 1 Connect Enable	Checked, Unchecked	Checked	en_m0 and en_s1
…	—	—	—
Master 0 Slave 31 Connect Enable	Checked, Unchecked	Checked	en_m0 and en_s31
Master 1 Connection Setting Group			
Master 1 Slave 0 Connect Enable	Checked, Unchecked	Checked	en_m1 and en_s0
Master 1 Slave 1 Connect Enable	Checked, Unchecked	Checked	en_m1 and en_s1
…	—	—	—
Master 1 Slave 31 Connect Enable	Checked, Unchecked	Checked	en_m1 and en_s31
…			
Master 31 Connection Setting Group			
Master 31 Slave 0 Connect Enable	Checked, Unchecked	Checked	en_m31 and en_s0
Master 31 Slave 1 Connect Enable	Checked, Unchecked	Checked	en_m31 and en_s1
…	—	—	—
Master 31 Slave 31 Connect Enable	Checked, Unchecked	Checked	en_m31 and en_s31
Main Settings Tab			
Slave 0 Settings Group			
S0_FRAGMENT_CNT	1–8	1	en_s0
Base Address 0	0–($2^{M\_ADDR\_WIDTH}$ – 'h400)	32'h00000000	en_s0 and (*S0_FRAGMENT_CNT* > 0)
Base Address 1	0–($2^{M\_ADDR\_WIDTH}$ – 'h400)	32'h00000400	en_s0 and (*S0_FRAGMENT_CNT* > 1)
…	—	—	—
Base Address 7	0–($2^{M\_ADDR\_WIDTH}$ – 'h400)	32'h00001C00	en_s0 and (*S0_FRAGMENT_CNT* > 7)
Address Range 0	'h400–$2^{M\_ADDR\_WIDTH}$	32'h00000400	en_s0 and (*S0_FRAGMENT_CNT* > 0)
Address Range 1	'h400–$2^{M\_ADDR\_WIDTH}$	32'h00000400	en_s0 and (*S0_FRAGMENT_CNT* > 1)
…	—	—	—
Address Range 7	'h400–$2^{M\_ADDR\_WIDTH}$	32'h00000400	en_s0 and (*S0_FRAGMENT_CNT* > 7)
Slave 1 Settings Group	—	—	—
S1_FRAGMENT_CNT	1–8	1	en_s1
Base Address 0	0–($2^{M\_ADDR\_WIDTH}$ – 'h400)	32'h00002000	en_s1 and (*S1_FRAGMENT_CNT* > 0)
Base Address 1	0–($2^{M\_ADDR\_WIDTH}$ – 'h400)	32'h00002400	en_s1 and (*S1_FRAGMENT_CNT* > 1)
…	—	—	—
Base Address 7	0–($2^{M\_ADDR\_WIDTH}$ – 'h400)	32'h00003C00	en_s1 and (*S1_FRAGMENT_CNT* > 7)

Attribute	Selectable Values	Default	Dependency on Other Attributes[4]
Address Range 0	'h400–$2^{M\_ADDR\_WIDTH}$	32'h00000400	en_s1 and (*S1_FRAGMENT_CNT* > 0)
Address Range 1	'h400–$2^{M\_ADDR\_WIDTH}$	32'h00000400	en_s1 and (*S1_FRAGMENT_CNT* > 1)
…	—	—	—
Address Range 7	'h400–$2^{M\_ADDR\_WIDTH}$	32'h00000400	en_s1 and (*S1_FRAGMENT_CNT* > 7)
…			
Slave 31 Settings Group	—	—	—
S31_FRAGMENT_CNT	1–8	1	en_s31
Base Address 0	0–($2^{M\_ADDR\_WIDTH}$ – 'h400)	32'h0003E000	en_s31 and (*S31_FRAGMENT_CNT* > 0)
Base Address 1	0–($2^{M\_ADDR\_WIDTH}$ – 'h400)	32'h0003E400	en_s31 and (*S31_FRAGMENT_CNT* > 1)
…	—	—	—
Base Address 7	0–($2^{M\_ADDR\_WIDTH}$ – 'h400)	32'h0003EC00	en_s31 and (*S31_FRAGMENT_CNT* > 7)
Address Range 0	'h400–$2^{M\_ADDR\_WIDTH}$	32'h00000400	en_s31 and (*S31_FRAGMENT_CNT* > 0)
Address Range 1	'h400–$2^{M\_ADDR\_WIDTH}$	32'h00000400	en_s31 and (*S31_FRAGMENT_CNT* > 1)
…	—	—	—
Address Range 7	'h400–$2^{M\_ADDR\_WIDTH}$	32'h00000400	en_s31 and (*S31_FRAGMENT_CNT* > 7)
Master Priority Settings Tab			
Slave 0 Settings Group			
Arbiter Scheme[3]	Round Robin, Fixed Priority, Weighted Round Robin	Round Robin	en_s0
Master 0 Priority	0–32	1	en_s0 and en_m1 and (*Arbiter Scheme* == Fixed Priority)
Master 1 Priority	0–32	2	en_s0 and en_m1 and (*Arbiter Scheme* == Fixed Priority)
Master 2 Priority	0–32	3	en_s0 and en_m2 and (*Arbiter Scheme* == Fixed Priority)
…	—	—	—
Master 31 Priority	0–32	32	en_s0 and en_m31 and (*Arbiter Scheme* == Fixed Priority)
Slave 1 Settings Group			
Arbiter Scheme[2]	Round Robin, Fixed Priority, Weighted Round Robin	Round Robin	en_s1
Master 0 Priority	0–32	1	en_s1 and en_m1 and (*Arbiter Scheme* == Fixed Priority)
Master 1 Priority	0–32	2	en_s1 and en_m1 and (*Arbiter Scheme* == Fixed Priority)
Master 2 Priority	0–32	3	en_s1 and en_m2 and (*Arbiter Scheme* == Fixed Priority)
…	—	—	—
Master 31 Priority	0–32	32	en_s1 and en_m31 and (*Arbiter Scheme* == Fixed Priority)
…	—	—	—
Slave 31 Settings Group			
Arbiter Scheme[2]	Round Robin, Fixed Priority, Weighted Round Robin	Round Robin	en_s31
Master 0 Priority	0–32	1	en_s31 and en_m1 and (*Arbiter Scheme* == Fixed Priority)

Attribute	Selectable Values	Default	Dependency on Other Attributes[4]
Master 1 Priority	0–32	2	en_s31 and en_m1 and (*Arbiter Scheme* == Fixed Priority)
Master 2 Priority	0–32	3	en_s31 and en_m2 and (*Arbiter Scheme* == Fixed Priority)
…	—	—	—
Master 31 Priority	0–32	32	en_s31 and en_m31 and (*Arbiter Scheme* == Fixed Priority)
Max Burst Size Settings Tab			
Max Burst Size Slave Settings			
Slave 0 Max Burst Size	0, 32, 64, 128, 256	0	en_s0
Slave 1 Max Burst Size	0, 32, 64, 128, 256	0	en_s1
…	—	—	—
Slave 31 Max Burst Size	0, 32, 64, 128, 256	0	en_s31

Notes:

1. [*Full Address Decoding up to 1kB* = Unchecked] is currently not supported.
2. [*Registered Output* = Checked] is currently not supported.
3. [*Arbiter Scheme* = Weighted Round Robin] is currently not supported.
4. To simplify the condition, en_s0, en_s1 to en_s31 are used for the condition that the corresponding connection to external slave is enabled. Similarly, en_m0, en_m1, en_m2 and en_m31 are used for the condition that the corresponding connection to external master is enabled.

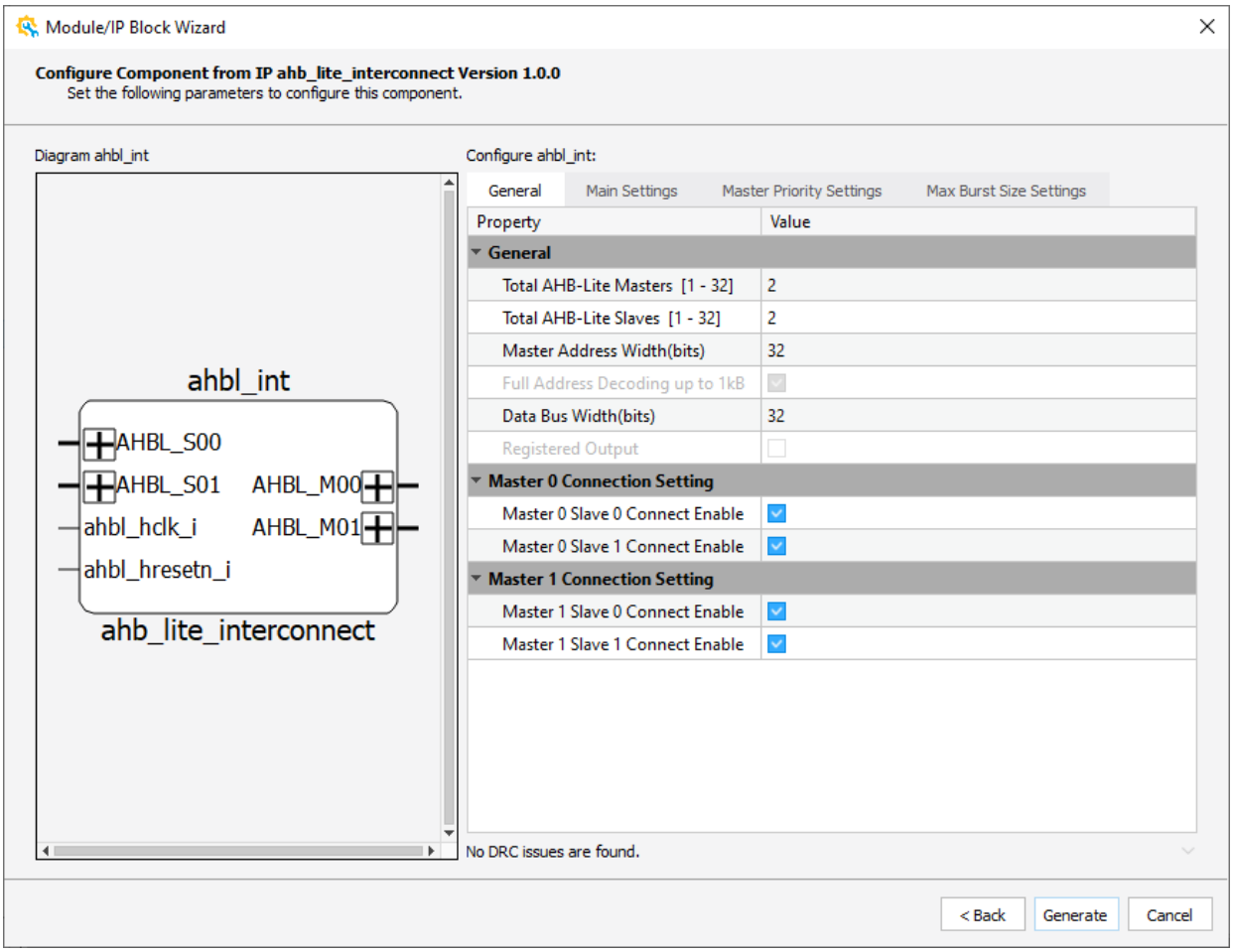

Figure 2.2. AHB-Lite Interconnect Module Configuration User Interface

Table 2.3. Attributes Description

Attribute	Description
General Tab	
General Group	
Total AHB-Lite Masters	Specifies the number of AHB-Lite masters that are connected to this module.
Total AHB-Lite Slaves	Specifies the number of AHB-Lite slaves that are connected to this module.
Master Address Width(bits) (M_ADDR_WIDTH)	Specifies the bit width of all the HADDR signals.
Data Bus Width(bits) (DATA_WIDTH)	Specifies the bit with of all the HWDATA and HRDATA signals.
Registered Output*1	Unchecked – Output register is disabled (bypassed). Checked – Output register is enabled. This option is currently not supported.
Master <M> Connection Setting Group	
Master <M> Slave <N> Connect Enable	Unchecked – AHB-Lite Master <M> can access the AHB-Lite Slave <N> and corresponding logic is not generated. Checked – AHB-Lite Master <M> can access the AHB-Lite Slave <N>.
Main Settings Tab	
Slave <N> Settings Group	
S<N>_FRAGMENT_CNT	Specifies the number of fragments in AHB-Lite slave <N>. Each fragment has its own base address and range boundary, wherein: • Start Address = Base Address • End Address = Base Address + Address Range – 1
Base Address <F>	Specifies base address for AHB-Lite slave <N> – fragment <F>. If [*Full Address Decoding up to 1kB* = Checked]: Must be aligned to 1 kB. If [*Full Address Decoding up to 1kB* = Unchecked]: Must be aligned to (or multiple of) the value of Address Range rounded up to power of 2.
Address Range <F>	Specifies number of bytes allocated for AHB-Lite slave <N> – fragment <F>. If [*Full Address Decoding up to 1kB* = Checked]: The value must be multiple of 1 kB. If [*Full Address Decoding up to 1kB* = Unchecked]: The value must be power of 2.
Master Priority Settings Tab	
Slave <N> Settings Group	
Arbiter Scheme*2	Specifies the arbitration scheme to be implemented for AHB-Lite slave <N>.
Master <M> Priority	Specifies priority index of AHB-Lite master <M> to access AHB-Lite slave <N>. [0]: disabled connection – master cannot access the slave. [1]: Highest priority … [31]: Lowest priority
Max Burst Size Settings Tab	
Max Burst Size Slave Settings Group	
Slave <N> Max Burst Size	Specifies the maximum burst size for the AHB-Lite slave <N> before the Interconnect sends an error response to the master. This prevents a master from hogging the bus by generating very long bursts. If this is set to 0, it means the feature is disabled and the corresponding logic is not generated.

Notes:

- <M> - Master index [0,1,…,(*Total AHB-Lite Masters*)-1]
- <N> - Slave index [0,1,…,(*Total AHB-Lite Slaves*)-1]
- For *Address Range <F>* DRC of the Module/IP Block Wizard flags an error when range is not power of 2. If your target range is not power of 2, you should select the next power of 2 value.

2.4. Use Models

The AHB-Lite Interconnect Module connects one or more AHB-Lite master devices to one or more AHB-Lite slave devices. Each connected AHB-Lite master device could either be:

- A device that originates AHB-lite transactions (endpoint master) or
- A master interface of an upstream AHB-Lite Interconnect core being cascaded

Similarly, each connected AHB-Lite slave device could either be:

- The final target of AHB-Lite transactions (endpoint slave) or
- A slave interface of a downstream AHB-Lite Interconnect core being cascaded

In general, AHB-Lite Interconnect Module can be configured for the following connectivity patterns:

- Single Master Interconnect – refer to the Single Master Interconnect section.
- Multi-Master Interconnect – refer to the Multi-Master Interconnect section.

An example of Single Master Interconnect application is shown in Figure 2.3. The arrows in the figure are AHB-Lite interface connections, where *M* stands for an AHB-Lite master port, and *S* stands for an AHB-Lite slave port. In this example, the Instruction port of the CPU is directly connected to one port of Dual Port Memory while the Data port of the CPU is connected to the slave port of Single Master Interconnect. This connection allows parallel instruction fetch and data access execution. The master ports of Single Master Interconnect are connected to one port of Dual Port Memory, Accelerator, DDR3 Memory Controller, and AHBL2APB Bridge. This allows the CPU Data port to access the said slaves. The APB slaves connected to AHBL2APB Bridge are not shown in this diagram.

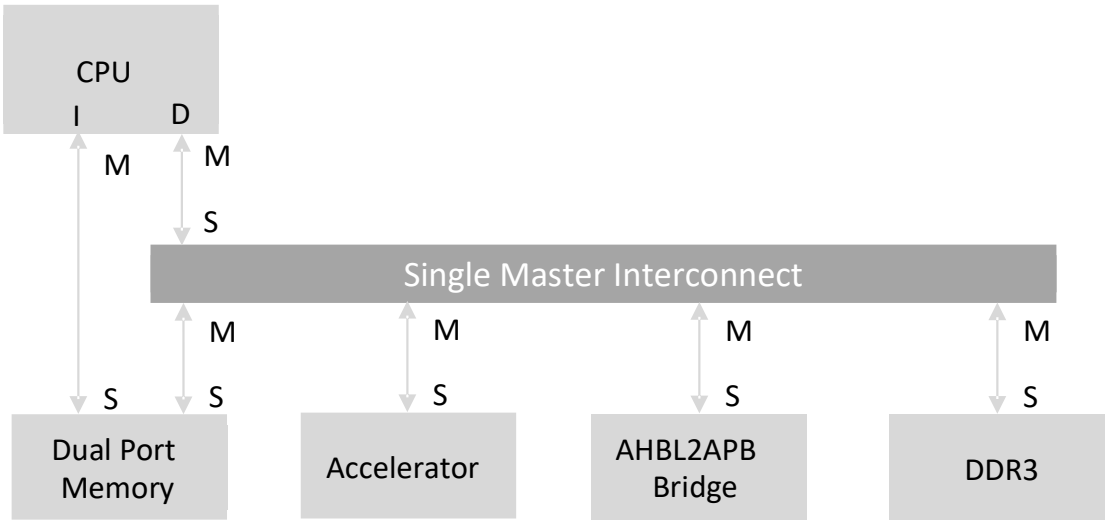

Figure 2.3. Example of Single Master Interconnect Application

An example of Multi-Master Interconnect application is shown in Figure 2.4. This is similar to Figure 2.3 with the addition of PCIe. In this example, PCI is configured to have one AHB-Lite master port for reading and writing data to DDR3 Memory and one AHB-Lite slave port for register access by the CPU. The Multi-Master Interconnect can be configured to remove the connection from PCIe M0 to other slaves, thus, saving resources and improving Fmax.

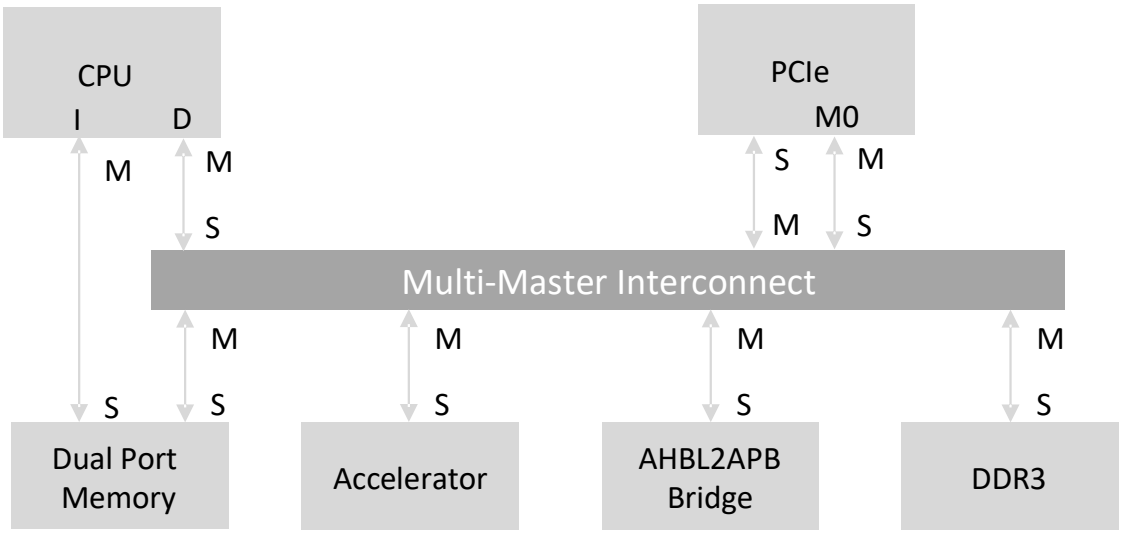

Figure 2.4. Example of Multi-Master Interconnect Application

2.4.1. Single Master Interconnect

For single master to multiple slave configuration, Lattice Semiconductor AHB-Lite Interconnect Module behavior is based on Section 4: Bus Interconnection of AMBA 3 AHB-Lite Protocol v1.0 Specifications. The bus interconnect logic is encapsulated inside the Soft IP, which consists of an address decoder, a slave-to-master multiplexer and a default slave. This is shown in Figure 2.5.

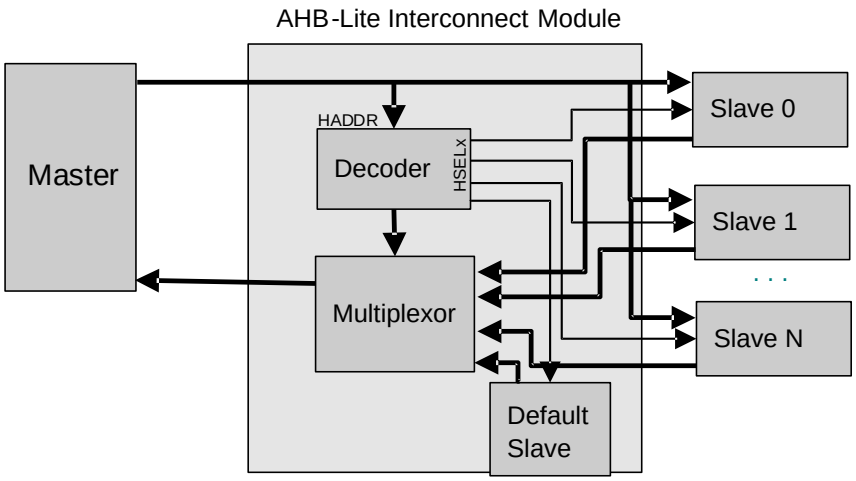

Figure 2.5. Functional Block Diagram of Single Master Interconnect

2.4.1.1. Decoder

The Decoder IP component performs address decoding for each bus transfer and provides a select signal for each slave on the bus. Decoding is done by comparing the appropriate address bits with the user-provided memory-map (*Base Address <F>* and *Address Range <F>*) settings and master connect enable settings during configuration. This is shown in Figure 2.6. The memory map is static setting; it cannot be changed during operation. During a bus transfer, the select signal goes high for a single slave involved in the transfer. The decoder also provides control signals to the multiplexor. The master connect enable settings are mainly used for Multi-Master Interconnect.

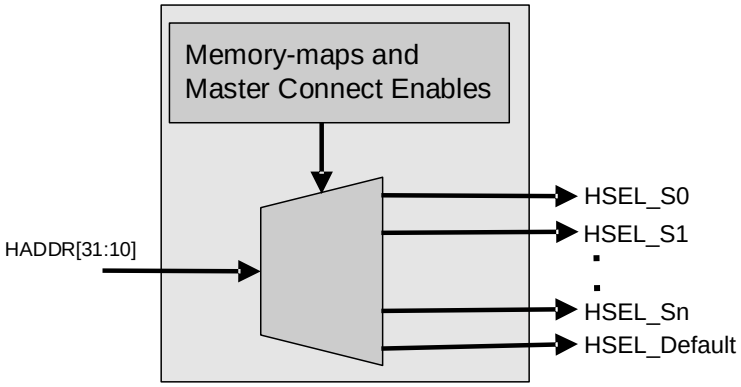

Figure 2.6. Block Diagram of AHB-Lite Decoder

According to the AMBA 3 AHB-Lite Specifications, the minimum address space that can be allocated to a single slave is 1 kB. When configuring the IP, slave Address Range attribute value must be greater than or equal to 1 kB. All masters are designed so that they do not perform incrementing transfers over a 1 kB address boundary. This ensures that a burst never crosses an address decode boundary and that only one slave is accessed by the master during a burst transfer.

The HSEL signal is a combinatorial decode of the high-order address signals. This is done by comparing the higher bits of the address signal with the *Base Address <F>* and *Address Range <F>* settings. The behavior of decoder is controlled by the *Full Address Decoding up to 1 kB* attribute as follows:

- [*Full Address Decoding up to 1kB* is Checked]: HSEL signal asserts if the following conditions are met:
  - ahbl_sxx_haddr_slv_i[*M_ADDR_WIDTH* – 1:10] >= *Base Address <F>*[*M_ADDR_WIDTH* – 1:10]
  - ahbl_sxx_haddr_slv_i[*M_ADDR_WIDTH* – 1:10] <= (Max Address <F>)[*M_ADDR_WIDTH* – 1:10]
    Wherein Max Address <F> = *Base Address <F>* + *Address Range <F>*
- [*Full Address Decoding up to 1kB* is Unchecked]: HSEL signal asserts if the following condition is met:
  - ahbl_sxx_haddr_slv_i[*M_ADDR_WIDTH* – 1:XBIT] == *Base Address <F>*[*M_ADDR_WIDTH* – 1: XBIT]
    Wherein XBIT = Round_up($\log_2$(*Address Range <F>*))

When *Full Address Decoding up to 1kB* is Unchecked, the *Address Range <F>* is not in used in comparison because it is in power of 2 value. Thus, the behavior of the interconnect is undefined if this attribute is not power of 2 value.

The user-defined memory maps of the slaves are parameters to the decoder IP and are used to select the appropriate slave. If a system design does not contain a completely filled memory map, then a default slave is selected when a transfer is attempted to a nonexistent address location.

2.4.1.2. Default Slave

When a transfer is attempted to an address that does not map to a slave, the default slave provides a response based on the transfer type (HTRANS) as follows:

- NONSEQUENTIAL or SEQUENTIAL (single or burst transfer) – ERROR response
- IDLE or BUSY (master is not providing data) – Zero wait state OKAY response

2.4.1.3. Multiplexor

A slave-to-master multiplexor is required to select the HRDATA, HREADY, and HRESP signals from the slaves to the master as shown in Figure 2.7. The decoder provides the multiplexor with the HSELx signals to enable the multiplexor to route the appropriate signals from the selected slave to the master. AHB-Lite has an address phase followed by one or more data phases. Therefore, the select signals should be flopped to align with the data phase(s). The registers shown below sample the HSEL signals. They should be updated when HREADY is asserted, indicating that the current transfer is complete.

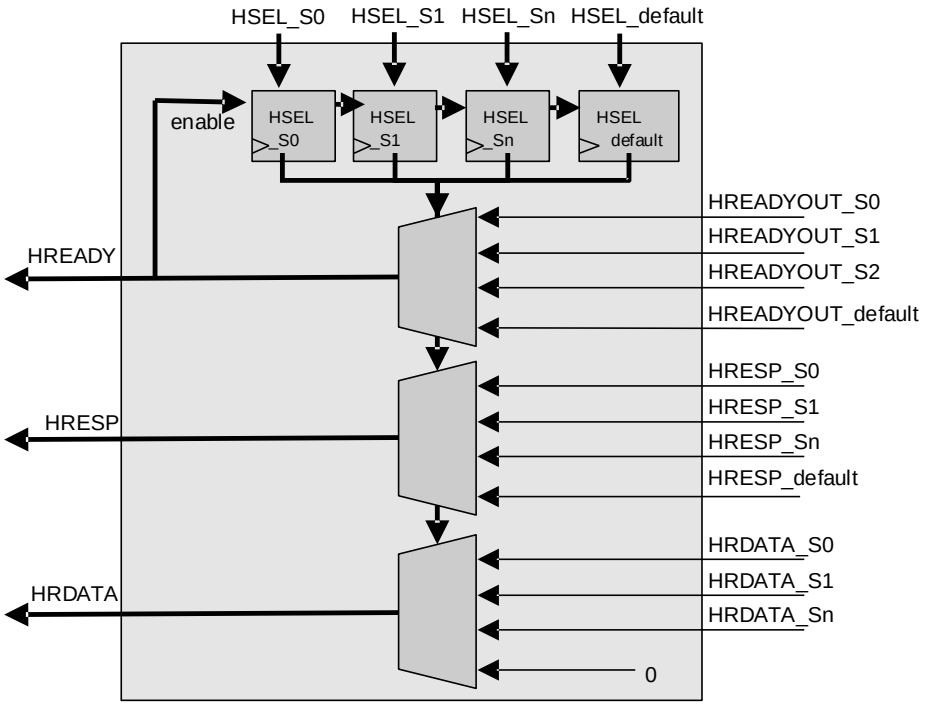

Figure 2.7. Block Diagram of AHB-Lite Multiplexor

2.4.2. Multi-Master Interconnect

The Lattice Semiconductor AHB-Lite Interconnect Module implements multi-layer AHB-Lite Interconnect Matrix. Each master is considered in its own layer and is isolated from other masters. The Multi-Master Interconnect allows non-contending transactions (between disjoint masters and slaves) to proceed concurrently. When multiple masters access the same slave, arbitration occurs and the interconnect matrix negates the HREADY to the masters that were not given access to the slave. Arbitration is the process of changing the grant from one master to another. The arbitration process has one wait cycle (HREADY is negated) penalty. If only the granted master access the slave for a given time, no arbitration occurs. However, if another not granted master access the slave, arbitration occurs to change the grant to that master even if no other master is accessing the same slave.

The Multi-Master Interconnect implements slave-side arbitration. Thus, multiple arbitration can occur in parallel if there are contentions in accessing multiple slaves at the same time. You can specify a different arbitration scheme for each slave, which is used to select a master during bus contention. Currently, there are two arbitration scheme options:

- Round-robin (default) – Each master is serviced in round-robin manner, giving each master equal access to the slave.
- Fixed priority – High priority masters are always given access to the slave in preference over lower priority masters.

By default, all masters are connected to all slaves. However, this is a waste of resource if at least one master never accesses at least one slave in the target application. An example of this is shown in Figure 2.8. Multi-Master Interconnect optimizes this by implementing configurable master to slave connection which is controlled by Master <M> Connection Setting attributes. When specific master to slave connection is disabled, corresponding logic is not generated; thus, improving resource utilization and Fmax (fan-out of select signals are reduced).

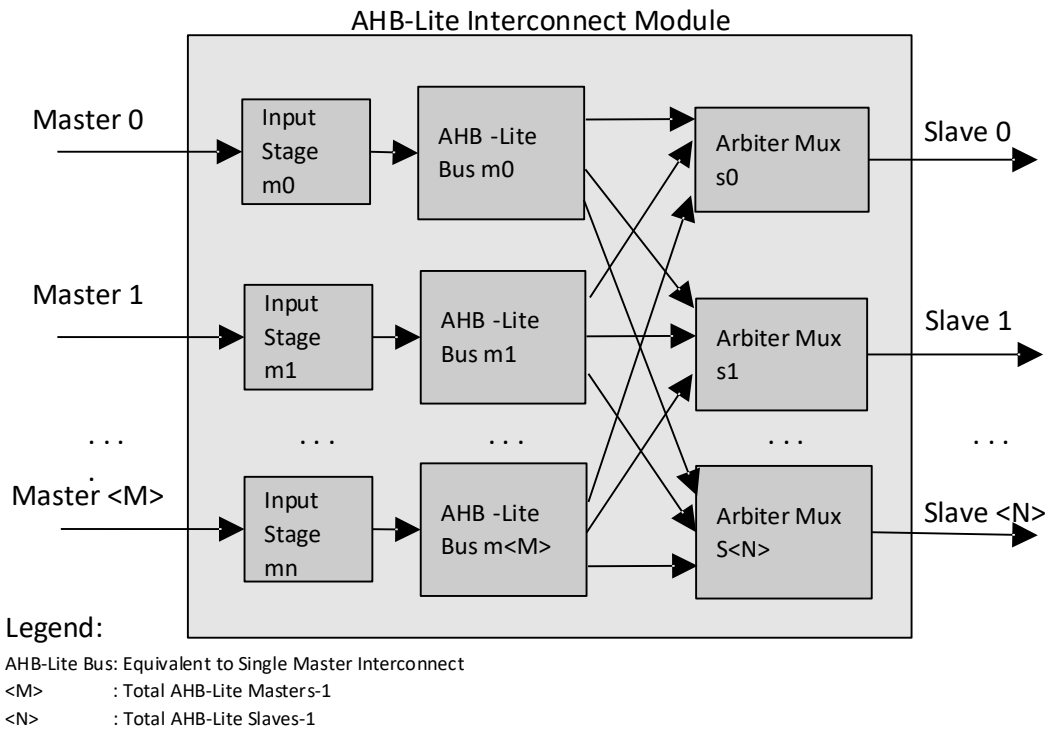

Figure 2.8. Block Diagram of Multi-Master Interconnect

2.4.2.1. Input Stage

The address phase cannot be stalled, hence, the address and control signals need to be registered so that when the master is not granted access, the registered address/control signals can be used to communicate with the slave. The input stage IP block is used to store the address and control until the master is given access. Control signal consists of all the signals that are sampled at the address phase: HWRITE, HSIZE, HBURST, HPROT, HTRANS, and HMASTLOCK.

Each Arbiter Mux sends stall signal to corresponding Input Stage if a not granted master initiates transaction.

2.4.2.2. Arbiter Mux

The Arbiter Mux uses the slave select (HSELx) and HTRANS input signals as requests to the internal arbiter block, which grants only one master. The grant signal is used in the multiplexer to select the master signals to the slave. In addition, the grant signal is used by the Arbiter Mux to lower the HREADY signal for all masters that are not selected and to generate the stall signals to the Input Stage.

Each Arbiter Mux has its own Arbiter.

References

- MachXO2 FPGA Web Page in latticesemi.com
- MachXO3 FPGA Web Page in latticesemi.com
- MachXO3D FPGA Web Page in latticesemi.com
- CrossLink-NX Web Page in latticesemi.com
- Certus-NX Web Page in latticesemi.com
- Mach-NX FPGA Web Page in latticesemi.com

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 1.2, May 2021

Section	Change Summary
Introduction	Updated document introduction, including Table 1.1 to add MachXO2 and MachXO3 as supported FPGA family.
References	Updated content to add reference for MachXO2 and MachXO3.

Revision 1.1, November 2020

Section	Change Summary
Introduction	• Updated information of Features section. • Added Table 1.1.
Functional Description	• Updated content of Overview, Use Models, Decoder, and Multi-Master Interconnect section. • Updated Figure2.2. • Updated Table 2.2. and Table 2.3.
References	Updated content to remove reference links for Lattice Propel and Lattice Diamond user guide; and to add reference links for Mach-NX, CrossLink-NX, and Certus-NX web page.

Revision 1.0, May 2020

Section	Change Summary
All	Initial release.

www.latticesemi.com