AHB-Lite Multilayer Interconnect

Datasheet (v1.2)

https://roalogic.github.io/ahb3lite_interconnect

04-Nov-2020

Contents

1 Introduction **1**

1.1 Features . . . 1

2 Specifications **2**

2.1 Functional Description . . . 2

2.2 Master Port . . . 2

2.2.1 Master Priority . . . 3

2.2.2 Bus Locking Support . . . 3

2.2.3 Specifying the number of Master Ports . . . 3

2.3 Slave Port . . . 3

2.3.1 Address Space Configuration . . . 4

2.3.2 Slave Port HREADYOUT and HREADY Routing . . . 5

2.3.3 Specifying the number of Slave Ports . . . 6

3 Configuration **7**

3.1 Introduction . . . 7

3.2 Core Parameters . . . 7

3.2.1 HADDR_SIZE . . . 7

3.2.2 HDATA_SIZE . . . 7

3.2.3 MASTERS . . . 7

3.2.4 SLAVES . . . 7

3.2.5 SLAVE_MASK[ ] . . . 7

3.2.6 ERROR_ON_SLAVE_MASK[ ] . . . 8

3.3 Core Macros . . . 8

3.3.1 RECURSIVE_FUNCTIONS_SUPPORTED . . . 8

4 Interfaces **9**

4.1 Global Signals . . . 9

4.1.1 HRESETn . . . 9

4.1.2 HCLK . . . 9

4.2 Master Interface . . . 9

4.2.1 mst_HSEL . . . 10

4.2.2 mst_HTRANS . . . 10

4.2.3 mst_HADDR . . . 10

4.2.4 mst_HWDATA . . . 11
4.2.5 mst_HRDATA . . . 11
4.2.6 mst_HWRITE . . . 11
4.2.7 mst_HSIZE . . . 11
4.2.8 mst_HBURST . . . 11
4.2.9 mst_HPROT . . . 12
4.2.10 mst_HREADYOUT . . . 12
4.2.11 mst_HMASTLOCK . . . 12
4.2.12 mst_HREADY . . . 12
4.2.13 mst_HRESP . . . 12
4.2.14 mst_priority . . . 12
4.3 Slave Interface . . . 13
4.3.1 slv_addr_base . . . 13
4.3.2 slv_addr_mask . . . 13
4.3.3 slv_HSEL . . . 14
4.3.4 slv_HADDR . . . 14
4.3.5 slv_HRDATA . . . 14
4.3.6 slv_HWDATA . . . 14
4.3.7 slv_HWRITE . . . 14
4.3.8 slv_HSIZE . . . 14
4.3.9 slv_HBURST . . . 14
4.3.10 slv_HPROT . . . 15
4.3.11 slv_HTRANS . . . 15
4.3.12 slv_HMASTLOCK . . . 15
4.3.13 slv_HREADYOUT . . . 16
4.3.14 slv_HREADY . . . 16
4.3.15 slv_HRESP . . . 16

5 Resources **17**

6 Revision History **18**

1. Introduction

The Roa Logic AHB-Lite Multi-layer Interconnect is a fully parameterized soft IP High Performance, Low Latency Interconnect Fabric for AHB-Lite. It allows a virtually unlimited number of AHB-Lite Bus Masters and Slaves to be connected without the need of bus arbitration to be implemented by the Bus Masters. Instead, Slave Side Arbitration is implemented for each Slave Port within the core.

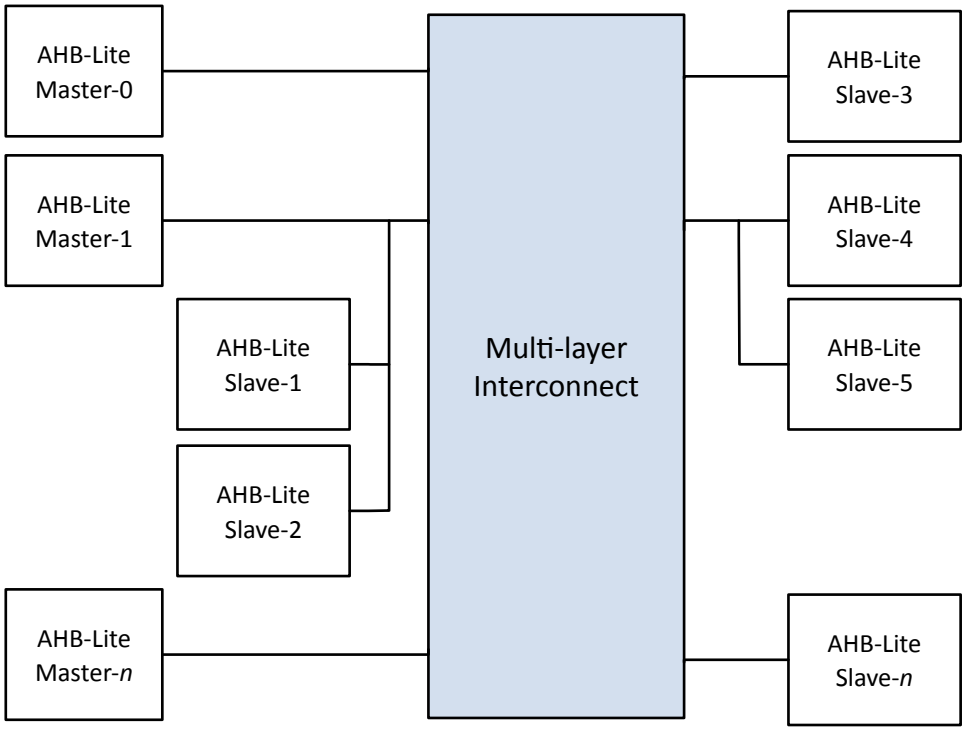

Figure 1.1: Multi–layer Interconnect Usage Example

The Multi-layer Interconnect supports priority based and Round-Robin based arbitration when multiple Bus Masters request access to the same Slave Port. Typically arbitration completes within 1 clock cycle.

1.1 Features

- AMBA AHB-Lite Compatible
- Fully parameterized
- Unlimited number of Bus Masters and Slaves[1]
- Slave side arbitration
- Priority and Round-Robin based arbitration
- Slave Port address decoding

[1] The number of Bus Masters and Slaves is physically limited by the timing requirements.

2. Specifications

2.1 Functional Description

The Roa Logic AHB-Lite Multi-layer Interconnect is a highly configurable Interconnect Fabric for AMBA AHB-Lite based systems, enabling multiple Masters to be connected to multiple Slaves.

Connections are dynamically created based on which Slave a Master is addressing, and once created enable direct communication between Master and Slave without other Masters being aware or interfering.

A new connection is typically created within one clock cycle, providing high bandwidth and low latency communication between Master and Slave.

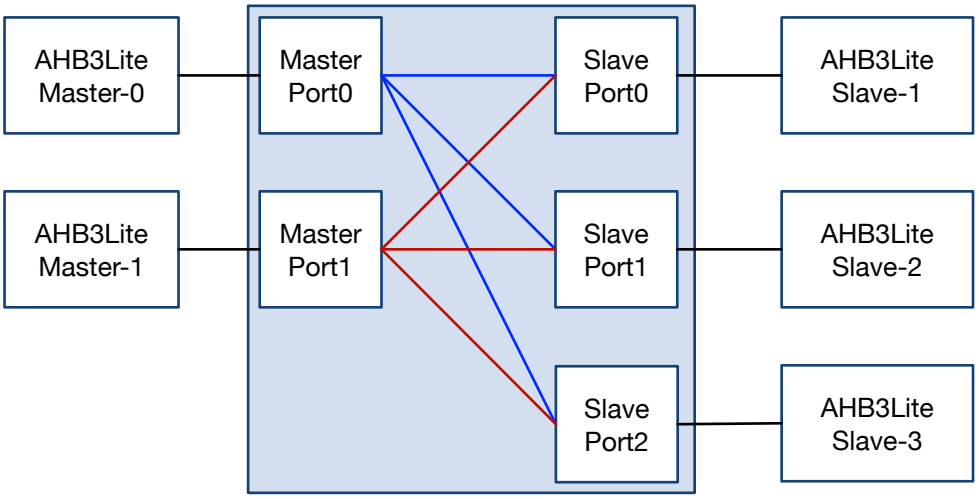

Figure 2.1: Example Master / Slave Communication Setup

2.2 Master Port

An AHB-Lite Bus Master connects to a Master port of the Multi-layer Interconnect. The Master port is implemented as a regular AHB-Lite Slave Interface thereby allowing support for complex bus structures.

The following figure shows an example bus structure where a Bus Master – Master-1 – has two directly connected Slaves; the Interconnect-Master-Port1 and Slave-4

To access a Slave, the Interconnect first checks if the designated Slave Port is available. If it is available the Slave Port immediately switches to the requesting Master. If the Slave Port is occupied due to another Master accessing the Slave, the Master Port generates wait states until the requested Slave becomes available. Note the pipelined nature of the AHB-Lite bus may cause a single wait state to be inserted when the Slave switches to a new Master.

The Slave Port always retains the connection to the Master until another Master requests access to that Slave Port; this enables the original Master to request further access to the Slave without incurring any delay due to arbitration.

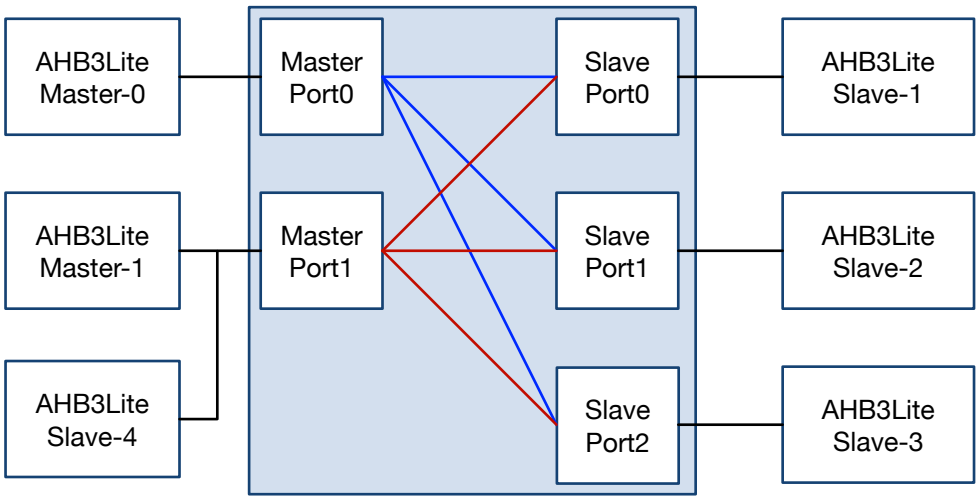

Figure 2.2: Connectivity Example for 2 Bus Masters, 4 Slaves

2.2.1 Master Priority

Each Master Port has a priority level port (`mst_priority[]`).

When multiple Masters with different priority levels request access to the same Slave Port, access is always granted to the Master with the highest priority level. If a new Master requests access while a transaction is already in progress, access will be granted according to its priority, ahead of any waiting lower priority Masters. If Masters have the same priority level, then access is granted based on a Round-Robin scheme.

Master priority may be set dynamically, however assigning a static priority results in a smaller Interconnect and reduces timing paths. The priority value may only be changed while the Master Port is idle; i.e. `mst_HSEL` is negated ('0') and/or when `mst_HTRANS` is IDLE.

2.2.2 Bus Locking Support

The priority levels determine the order in which Masters are granted access to the Slave Port. The Slave Port switches between masters when the current accessing master is idle (`mst_HSEL` is negated and/or `mst_HTRANS` = IDLE) or when the current burst completes.

However the current Master may lock the bus by asserting `HMASTLOCK`; this prevents the Slave port switching.

2.2.3 Specifying the number of Master Ports

The number of Master Ports is specified by the `MASTERS` parameter.

2.3 Slave Port

An AHB-Lite Bus Slave connects to a Slave Port of the Multi-layer Interconnect. The Slave Port is implemented as a regular AHB3-Lite Master Interface thereby allowing support for complex bus structures such as shown below:

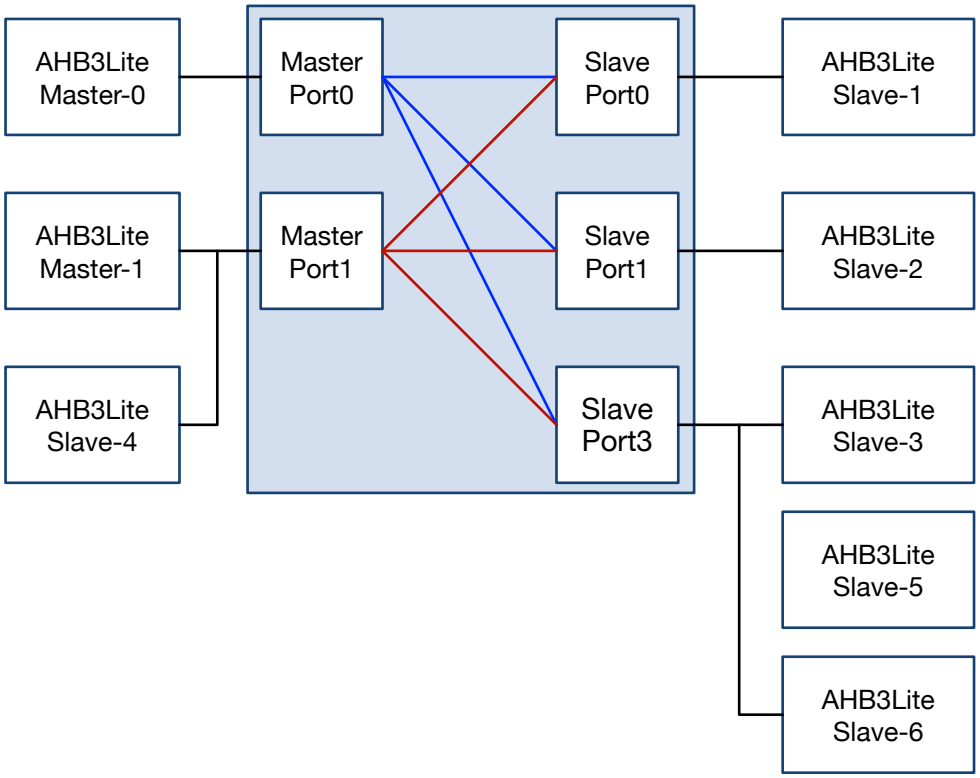

Figure 2.3: Connectivity Example for 2 Bus Masters, 6 Slaves

2.3.1 Address Space Configuration

Each Slave Port has an Address Base (`slv_addr_base`) and Address Mask (`slv_addr_mask`) port. Together these set the address range covered by the Slave Port.

The Address Base port specifies the base address for the address range covered by the Slave Port and the Address Mask port defines the address range covered by the Slave Port. The internal port select signal is specified as `slv_addr_base` AND `slv_addr_mask`.

The Address Base and Address Mask values may be changed dynamically, however assigning static values results in a smaller Interconnect and reduces timing paths. Address Base and Address Mask may only be changed when the slave port(s) are idle. Since multiple masters may be active at the same time trying to access the Interconnect, special care must be taken to ensure NO master accesses the Interconnect while updating the Address Base and Address Mask values.

The Slave Port asserts `HSEL` when accesses are within the port's address range. When the port is not being accessed `HSEL` is negated ('0'), but `HTRANS` and other AMBA signals will still provide data. These signals must be ignored while `HSEL` is negated ('0').

The slave port will output the full address, i.e. all `HADDR_SIZE` bits, on its address bus (`slv_HADDR`). Connected AMBA slaves should use the relevant least significant bits (LSBs) only.

Example 1

```
slave_addr_base = 32'h1000_0000
slave_addr_mask = 32'hF000_0000
Address-range = 32'h1000_0000 to 32'h1FFF_FFFF
```

Example 2

```
slave_addr_base = 32'h4000_0000
slave_addr_mask = 32'hE000_0000
Address-range = 32'h4000_0000 to 32'h5FFF_FFFF
```

2.3.2 Slave Port HREADYOUT and HREADY Routing

The Slave Port has an `HREADYOUT` port, which is not part of the AHB-Lite specification. It is required to support slaves on the master's local bus. The `HREADY` signal, generated by the multiplexor, is driven to the addressed slave's `HREADYOUT` port.

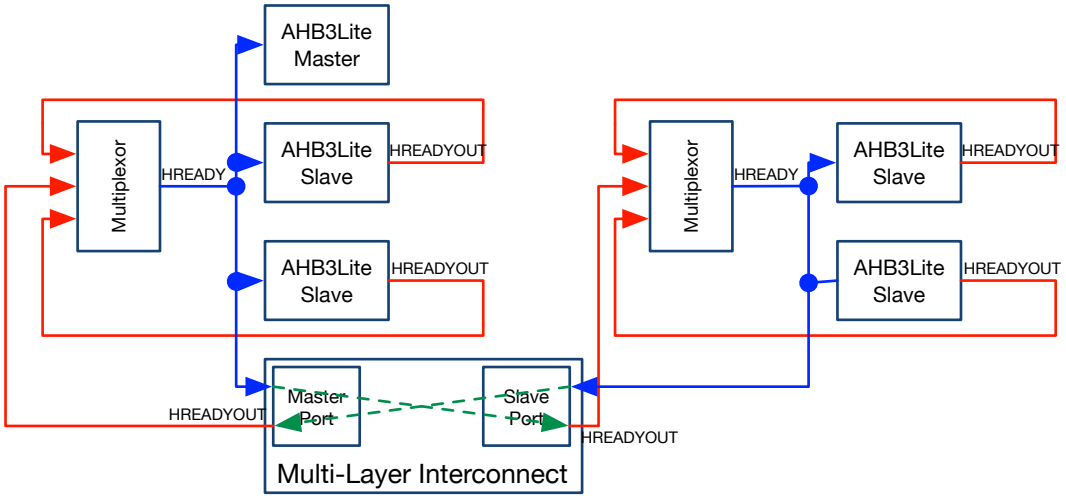

Figure 2.4: HREADYOUT and HREADY Routing

The simple case of where only one master is connected to a Master Port or where only a single slave is connected to a Slave Port is illustrated below.

There are no multiplexors on either the Master Bus or the Slave Bus. Since there is no other slave on the Master Bus, its `HREADY` signal is only driven by the Master Port's `HREADYOUT` signal. Thus the Master Port's `HREADYOUT` drives both the Master's `HREADY` input and the Master Port's `HREADY` input.

Similarly since there is no other slave on the Slave Bus, the Slave Port's `HREADYOUT` signals drives the slave's `HREADY` input and the slave's `HREADYOUT` signal drives the Slave Port's `HREADY` input.

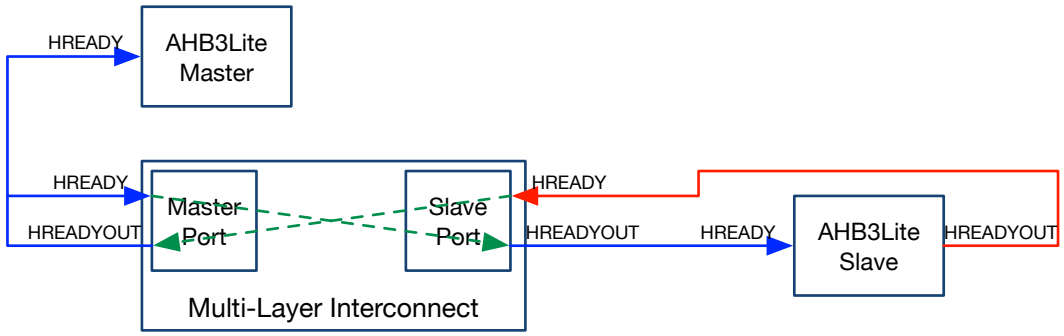

Figure 2.5: Single Master/Slave Routing

2.3.3 Specifying the number of Slave Ports

The number of Slave Ports is specified by the `SLAVES` parameter.

3. Configuration

3.1 Introduction

The Roa Logic AHB-Lite Multi-layer Interconnect is a highly configurable Interconnect Fabric for AMBA AHB-Lite based systems. The core parameters and configuration options are described in this section.

3.2 Core Parameters

Parameter	Type	Default	Description
HADDR_SIZE	Integer	32	Address Bus Size
HDATA_SIZE	Integer	32	Data Bus Size
MASTERS	Integer	3	Number of Master Ports
SLAVES	Integer	8	Number of Slave Ports
SLAVE_MASK[MASTERS]	Array	All '1's	Mask Slaves accessible by each Master
ERROR_ON_SLAVE_MASK[MASTERS]	Array	inv(SLAVE_MASK)	Enable Error Reporting for masked Slaves

Table 3.1: Core Parameters

3.2.1 HADDR_SIZE

The HADDR_SIZE parameter specifies the width of the address bus for all Master and Slave ports.

3.2.2 HDATA_SIZE

The HDATA_SIZE parameter specifies the width of the data bus for all Master and Slave ports.

3.2.3 MASTERS

The MASTERS parameter specifies the number of Master Ports on the Interconnect fabric.

3.2.4 SLAVES

The SLAVES parameter specifies the number of Slave Ports on the Interconnect Fabric.

3.2.5 SLAVE_MASK[ ]

The SLAVE_MASK[] parameter determines if a master may access a slave. Defining which master may access individual slaves (rather than allowing all masters to access all slaves) may significantly reduce the logic area of the interconnect and improve overall performance.

There is one SLAVE_MASK parameter per master, each SLAVES bits wide. i.e. SLAVE_MASK[] is an array of dimensions MASTERS x SLAVES.

Setting a SLAVE_MASK[] bit to '0' indicates that master cannot access the slave. Conversely, setting a SLAVE_MASK[] bit to '1' indicates that master may access the slave.

3.2.6 ERROR_ON_SLAVE_MASK[ ]

The ERROR_ON_SLAVE_MASK[] parameter enables generating an AHB error response when the master attempts to access a masked Slave Port.

There is one ERROR_ON_SLAVE_MASK parameter per master, each SLAVES bits wide. i.e. ERROR_ON_SLAVE_MASK[] is an array of dimensions MASTERS x SLAVES.

Setting an ERROR_ON_SLAVE_MASK[] bit to '0' indicates that an AHB error response will not be generated if the master is masked from accessing the corresponding slave. Conversely, setting a ERROR_ON_SLAVE_MASK[] bit to '1' indicates that an AHB error response will be generated if the master is masked from accessing the corresponding slave.

The default value of ERROR_ON_SLAVE_MASK[] is the bitwise inverse of SLAVE_MASK[] - i.e. inv(SLAVE_MASK[]). If SLAVE_MASK[] is assigned a value, then ERROR_ON_SLAVE_MASK[] is by default inv(SLAVE_MASK[]).

3.3 Core Macros

Macro	Description
RECURSIVE_FUNCTIONS_SUPPORTED	Enable use of recursive functions

Table 3.2: Core Macros

3.3.1 RECURSIVE_FUNCTIONS_SUPPORTED

Recursive functions and modules are used within the verilog source code. However EDA tools vary in their support for recursion, with some supporting recursive functions, whereas others support recursive modules or both. For example, Intel Quartus v19.1 and earlier supports recursive modules, but not recursive functions.

To accomodate these toolchain differences, recursive modules are the default method used for implementation. However recursive functions may instead be enabled by setting the synthesis macro RECURSIVE_FUNCTIONS_SUPPORTED

4. Interfaces

4.1 Global Signals

The common signals are shared between all devices on the AHB bus. The AHB-Lite Interconnect has Master and Slave AHB-Lite buses and they all use the global signals.

Port	Size	Direction	Description
HRESETn	1	Input	Asynchronous active low reset
HCLK	1	Input	System clock input

Table 4.1: AMBA3 Global Signals

4.1.1 HRESETn

When the active low asynchronous HRESETn input is asserted ('0'), the core is put into its initial reset state.

4.1.2 HCLK

HCLK is the system clock. All internal logic operates at the rising edge of the system clock. All AHB bus timings are related to the rising edge of HCLK. All Master and Slave ports must operate at the same HCLK clock.

4.2 Master Interface

The Master Ports are regular AHB3-Lite slave interfaces. All signals are supported. See the AHB-Lite specifications for a complete description of the signals. In addition, a custom master priority port is included per interface to enable prioritisation when multiple masters attempt to simultaneously access the same slave. This prioritisation may be defined statically or dynamically changed during operation.

The AHB-Lite Multi-layer Interconnect implements 1 or more interfaces to AHB-Lite masters as defined by the MASTERS parameter. Therefore the following signals are all arrays reflecting the number of masters supported.

Port	Size	Direction	Description
mst_HSEL	1	Input	Bus Select
mst_HTRANS	2	Input	Transfer Type
mst_HADDR	HADDR_SIZE	Input	Address Bus
mst_HWDATA	HDATA_SIZE	Input	Write Data Bus
mst_HRDATA	HDATA_SIZE	Output	Read Data Bus
mst_HWRITE	1	Input	Write Select
mst_HSIZE	3	Input	Transfer Size
mst_HBURST	3	Input	Transfer Burst Size
mst_HPROT	4	Input	Transfer Protection Level
mst_HMASTLOCK	1	Input	Transfer Master Lock

Port	Size	Direction	Description
mst_HREADYOUT	1	Output	Transfer Ready Output
mst_HREADY	1	Input	Transfer Ready Input
mst_HRESP	1	Input	Transfer Response

Table 4.2: Master Interface AHB-Lite Port

Port	Size	Direction	Description
mst_priority	$clog_2$(MASTERS)	Input	Master Priority Levels

Table 4.3: Master Interface Custom Port

Note: $clog_2$() refers to the System Verilog function by the same name, defined below, and is used to determine the required bitwidth of a bus required to represent the defined range of values:

> *The system function $clog2 shall return the ceiling of the log base 2 of the argument (the log rounded up to an integer value). The argument can be an integer or an arbitrary sized vector value. The argument shall be treated as an unsigned value, and an argument value of 0 shall produce a result of 0.*

4.2.1 mst_HSEL

The Master Port only responds to other signals on its bus when mst_HSEL is asserted ('1'). When mst_HSEL is negated ('0') the Master Port considers the bus IDLE and asserts HREADYOUT ('1').

4.2.2 mst_HTRANS

mst_HTRANS indicates the type of the current transfer. It is driven to the connected slave.

HTRANS	Type	Description
00	IDLE	No transfer required
01	BUSY	Connected master is not ready to accept data, but intents to continue the current burst.
10	NONSEQ	First transfer of a burst or a single transfer
11	SEQ	Remaining transfers of a burst

Table 4.4: Transfer Type (HTRANS)

4.2.3 mst_HADDR

mst_HADDR is the address bus. Its size is determined by the HADDR_SIZE parameter. It is driven to the connected slave.

4.2.4 mst_HWDATA

mst_HWDATA is the write data bus. Its size is determined by the HDATA_SIZE parameter. It is driven to the connected slave.

4.2.5 mst_HRDATA

mst_HRDATA is the read data bus. Its size is determined by HDATA_SIZE parameter. The connected slave drives it.

4.2.6 mst_HWRITE

mst_HWRITE is the read/write signal. mst_HWRITE asserted ('1') indicates a write transfer. It is driven to the connected slave.

4.2.7 mst_HSIZE

mst_HSIZE indicates the size of the current transfer. It is driven to the connected slave.

HSIZE	Size	Description
000	8bit	Byte
001	16bit	Half Word
010	32bit	Word
011	64bits	Double Word
100	128bit	
101	256bit	
110	512bit	
111	1024bit	

Table 4.5: Transfer Size Values (HSIZE)

4.2.8 mst_HBURST

The burst type indicates if the transfer is a single transfer or part of a burst. It is driven to the connected slave.

HBURST	Type	Description
000	SINGLE	Single access
001	INCR	Continuous incremental burst
010	WRAP4	4-beat wrapping burst
011	INCR4	4-beat incrementing burst
100	WRAP8	8-beat wrapping burst
101	INCR8	8-beat incrementing burst
110	WRAP16	16-beat wrapping burst
111	INCR16	16-beat incrementing burst

Table 4.6: Burst Types (HBURST)

4.2.9 mst_HPROT

The protection signals provide information about the bus transfer. They are intended to implement some level of protection. It is driven to the connected slave.

Bit #	Value	Description
3	1	Cacheable region addressed
	0	Non-cacheable region addressed
2	1	Bufferable
	0	Non-bufferable
1	1	Privileged Access
	0	User Access
0	1	Data Access
	0	Opcode fetch

Table 4.7: Protection Signals (HPROT)

4.2.10 mst_HREADYOUT

When a slave is addressed, the `mst_HREADYOUT` indicates that the addressed slave finished the current transfer. The Interconnect IP routes the addressed slave's `HREADY` signal to the master.

When no slave is addressed, the `mst_HREADYOUT` signal is generated locally, inside the Interconnect.

4.2.11 mst_HMASTLOCK

The master lock signal indicates if the current transfer is part of a locked sequence, commonly used for Read-Modify-Write cycles. While the `mst_HMASTLOCK` is asserted, the Interconnect IP cannot switch the addressed slave to another master, even if that master has a higher priority. Instead the current master retains access to slave until it releases `mst_HMASTLOCK`.

4.2.12 mst_HREADY

`mst_HREADY` indicates the status of the local `HREADY` on the master's local bus. It is routed to the `HREADYOUT` port of the connected slave.

4.2.13 mst_HRESP

`mst_HRESP` is the transfer response from the connected slave, it can either be OKAY ('0') or ERROR ('1'). The Interconnect IP routes the connected slave's `HRESP` port to `mst_HRESP`.

4.2.14 mst_priority

`mst_priority` is a custom port per master interface and defines the priority of the attached master. The width of the bus is calculated as clog_2(`MASTERS`), with the lowest priority value defined as 0 and the highest priority being `MASTERS-1`.

For example, for a system with 4 masters the width of `mst_priority` will be 2 bits to enable each master to have a unique prioritisation level, with lowest priority being 0 and highest priority being 3.

4.3 Slave Interface

The Slave Ports are regular AHB-Lite master interfaces.. All signals are supported. In addition each Slave Port has a non-standard `slv_HREADYOUT`. See the AHB-Lite specifications for a complete description of the signals.

The AHB-Lite Multi-layer Interconnect implements 1 or more interfaces to AHB-Lite slaves as defined by the `SLAVES` parameter. Therefore the following signals are all arrays reflecting the number of slaves supported.

Port	Size	Direction	Description
`slv_addr_base`	`HADDR_SIZE`	Input	Slave Base Address
`slv_addr_mask`	`HADDR_SIZE`	Input	Slave Address Space Mask

Table 4.8: Slave Interface Customisation Port

Port	Size	Direction	Description
`slv_HSEL`	1	Output	Bus Select
`slv_HADDR`	`HADDR_SIZE`	Output	Address
`slv_HWDATA`	`HDATA_SIZE`	Output	Write Data Bus
`slv_HRDATA`	`HDATA_SIZE`	Input	Read Data Bus
`slv_HWRITE`	1	Output	Write Select
`slv_HSIZE`	3	Output	Transfer size
`slv_HBURST`	3	Output	Transfer Burst Size
`slv_HPROT`	4	Output	Transfer Protection Level
`slv_HTRANS`	2	Input	Transfer Type
`slv_HMASTLOCK`	1	Output	Transfer Master Lock
`slv_HREADY`	1	Input	Transfer Ready Input
`slv_HRESP`	1	Input	Transfer Response

Table 4.9: Slave Interface AHB-Lite Port

4.3.1 slv_addr_base

`slv_addr_base` is a `SLAVES` sized array of addresses, each `HADDR_SIZE` bits wide, defining the base address of each attached slave device.

4.3.2 slv_addr_mask

`slv_addr_mask` is a `SLAVES` sized array of `HADDR_SIZE` bit wide signals. Each `slv_addr_base` address is masked with the corresponding `slv_addr_mask` value to define the addressable memory space of the attached slave. Setting a bit of `slv_addr_mask` to '0' enables the corresponding address bit.

See section 2.3.1 for specific examples.

4.3.3 slv_HSEL

Slaves connected to the Slave Port must only respond to other signals on the bus when `slv_HSEL` is asserted ('1'). When `slv_HSEL` is negated ('0') the interface is idle and the connected Slaves must assert their `HREADYOUT` ('1').

4.3.4 slv_HADDR

`slv_HADDR` is the data address bus. Its size is determined by the `HADDR_SIZE` parameter. The connected master drives `slv_HADDR`.

4.3.5 slv_HRDATA

`slv_HRDATA` is the read data bus. Its size is determined by the `HDATA_SIZE` parameter. It is driven to the connected master.

4.3.6 slv_HWDATA

`slv_HWDATA` is the write data bus. Its size is determined by the `HDATA_SIZE` parameter. The connected master drives `slv_HADDR`.

4.3.7 slv_HWRITE

`slv_HWRITE` is the read/write signal. `slv_HWRITE` asserted ('1') indicates a write transfer. The connected master drives `slv_HWRITE`.

4.3.8 slv_HSIZE

`slv_HSIZE` indicates the size of the current transfer. The connected master drives `slv_HSIZE`.

HSIZE	Size	Description
000	8bit	Byte
001	16bit	Half Word
010	32bit	Word
011	64bits	Double Word
100	128bit	
101	256bit	
110	512bit	
111	1024bit	

Table 4.10: Data Transfer Sizes

4.3.9 slv_HBURST

The burst type indicates if the transfer is a single transfer or part of a burst. The connected master drives it.

HBURST	Type	Description
000	Single	Single access
001	INCR	Continuous incremental burst
010	WRAP4	4-beat wrapping burst
011	INCR4	4-beat incrementing burst
100	WRAP8	8-beat wrapping burst
101	INCR8	8-beat incrementing burst
110	WRAP16	16-beat wrapping burst
111	INCR16	16-beat incrementing burst

Table 4.11: Burst Types (HBURST)

4.3.10 slv_HPROT

The data protection signals provide information about the bus transfer. They are intended to implement some level of protection. The connected master drives `slv_HPROT`.

Bit#	Value	Description
3	1	Cacheable region addressed
	0	Non-cacheable region addressed
2	1	Bufferable
	0	Non-bufferable
1	1	Privileged access. CPU is not in User Mode
	0	User access. CPU is in User Mode
0	1	Data transfer, always '1'

Table 4.12: Data Protection Signals

4.3.11 slv_HTRANS

`slv_HTRANS` indicates the type of the current data transfer.

slv_HTRANS	Type	Description
00	IDLE	No transfer required
01	BUSY	*Not used*
10	NONSEQ	First transfer of an data burst
11	SEQ	Remaining transfers of an data burst

Table 4.13: Data Transfer Type

4.3.12 slv_HMASTLOCK

The master lock signal indicates if the current transfer is part of a locked sequence, commonly used for Read-Modify-Write cycles. The connected master drives `slv_MASTLOCK`.

4.3.13 slv_HREADYOUT

The `slv_HREADYOUT` signal reflects the state of the connected Master Port's `HREADY` port. It is provided to support local slaves connected directly to the Master's AHB-Lite bus. It is driven by the connected master's `HREADY` port.

Note: `slv_HREADYOUT` is not an AHB-Lite Signal.

4.3.14 slv_HREADY

`slv_HREADY` indicates whether the addressed slave is ready to transfer data or not. When `slv_HREADY` is negated ('0') the slave is not ready, forcing wait states. When `slv_HREADY` is asserted ('0') the slave is ready and the transfer completed. It is driven to the connected master's `HREADYOUT` port.

4.3.15 slv_HRESP

`slv_HRESP` is the data transfer response, it can either be OKAY ('0') or ERROR ('1'). It is driven to the connected master.

5. Resources

Below are some example implementations when targeting the Altera Cyclone-V family of FPGAs. All implementations are push button, no effort has been undertaken to reduce area or improve performance.

	Configuration: Masters x Slaves							
↓ Res/Config →	10x5	8x5	8x3	5x3	3x5	3x8	5x8	5x10
ALM	6438	4272	2927	1934	1753	2644	4522	5703
Registers	1220	926	842	533	338	377	668	725
Fmax (MHz)	47	61	68	71	109	104	66	63

Table 5.1: Resource Utilisation Examples

6. Revision History

Date	Rev.	Comments
13-Oct-2017	1.0	Initial Release
16-Sep-2019	1.1	Updated to add `SLAVE_MASK[]` Parameter
04-Nov-2020	1.2	Add Recursive Function support

Table 6.1: Revision History