AHBLITE multi-layer interconnect

Catalog

**[1. Introduction 3](#_Toc277443529)**

**[2. Features 4](#_Toc806394266)**

**[3. Functional description 5](#_Toc299132445)**

[3.1. Masterport 5](#_Toc259785488)

[3.2. Slaveport 5](#_Toc380442465)

[3.3. HREADY and HREADYOUT routing 6](#_Toc1037692136)

**[4. Interface 7](#_Toc777032465)**

[4.1. Parameter 7](#_Toc736581848)

**[5. Design 9](#_Toc1635378028)**

[5.1. Masterport 9](#_Toc155318643)

# Introduction

AHBLite multi-layer interconnect (The interconnect) is a fully parameterized high performance, low latency interconnect for AHBLite. It allows a virtually unlimited number of AHBLite bus masters and slaves to be connected without the need for bus arbitration to be implemeted by the bus masters. Instead, slave side arbitration is implemented for each slave port within the core.

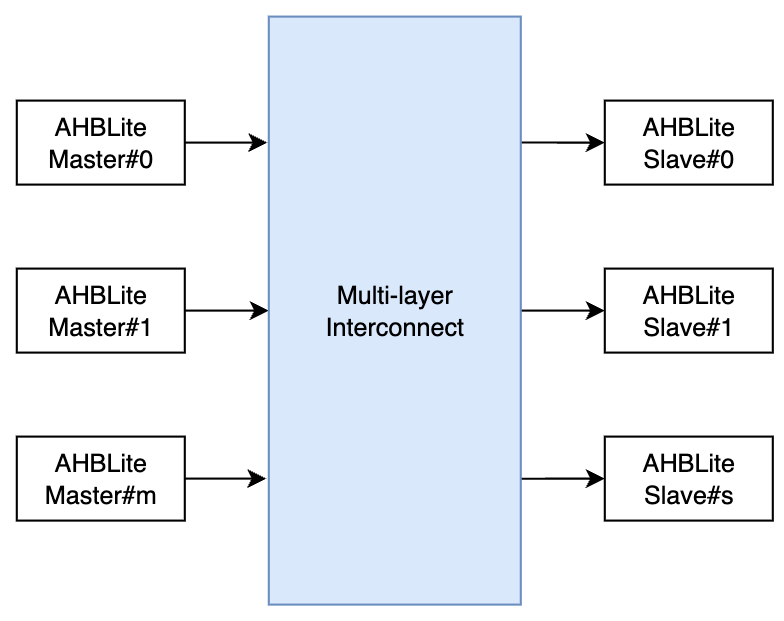


Figure 1-Example Multi-layer interconnect

The interconnect supports round-robin arbitration when multiple masters request access to the same slave port. Typically, arbitration completes within 1 cycle clock.

# Features

1. AMBA AHB-Lite compatible
2. Fully parameterized
3. Unlimited number of bus master and slave
4. Slave side round robin arbitration
5. Slave port address decoder

# Functional description

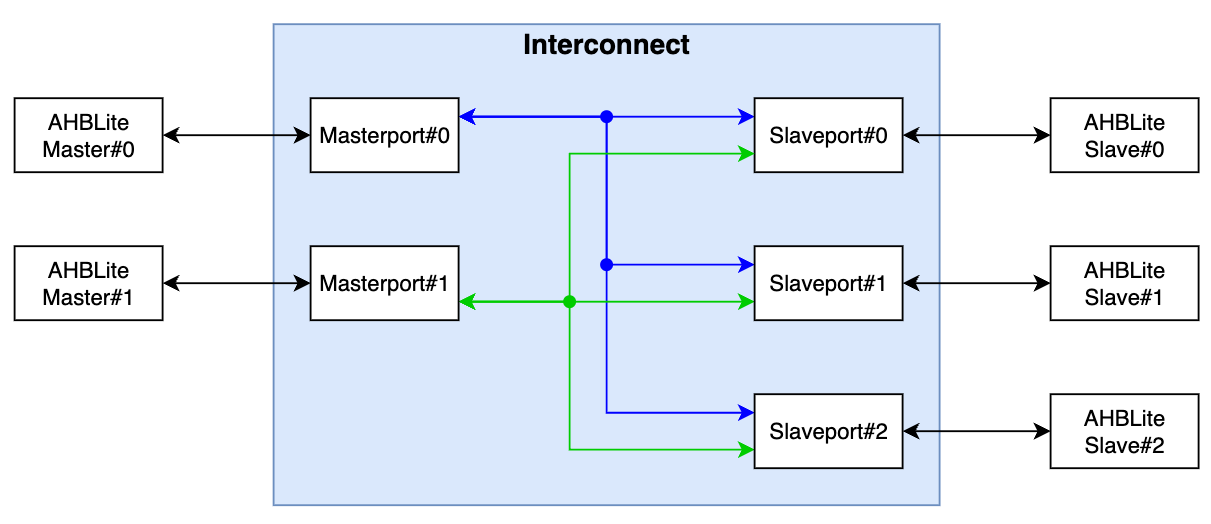


Figure 2-Example interconnect system with 2 masters and 3 slaves

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.

## Masterport

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.

To access a slave, master send to masterport the address of that slave. If the address is valid, masterport sends a select signal (mst\_HSEL\_o) to that slave to request access. After slave accept the request, master and slave can communicate without intervention from other masters. At the end of the command transaction, master will generate a pulse of swith signal (mst\_switch\_o) to notify slave, then slave can stop current grant, and generate new grant for other requests.

## Slaveport

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.

Each slave port has an address base (slv\_HADDR\_base) and address mask (slv\_HADDR\_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\_HADDR\_base AND slv\_HADDR\_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.

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

## HREADY and HREADYOUT routing

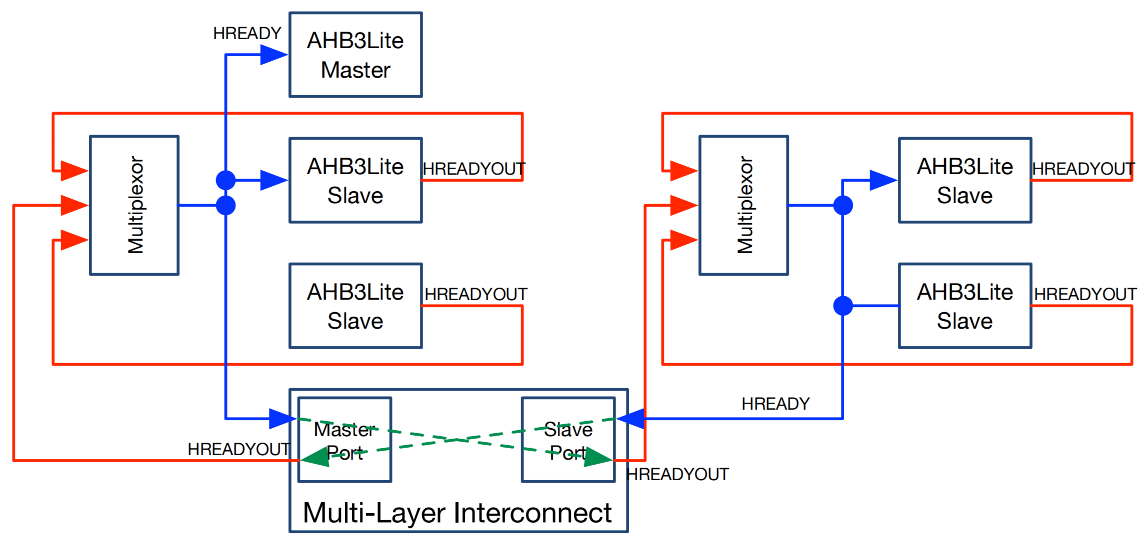


Figure 3-HREADYOUT and HREADY Routing

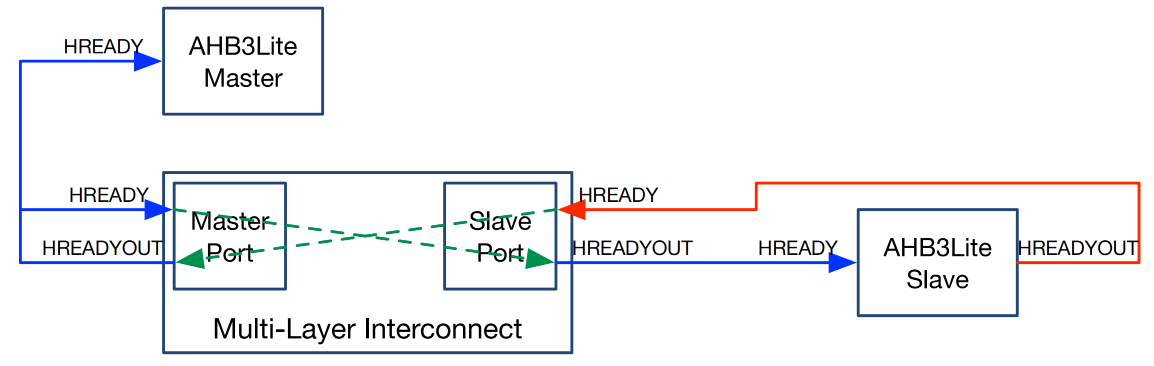


Figure 4-Single Master/Slave Routing

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 on the master local bus, drives the addressed slave’s HREADYOUT port.

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 figure 4.

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.

# Interface

## Parameter

* Parameter

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| MASTER | Number of masterport |
| SLAVE | Number of slaveport |
| HADDR\_WIDTH | Address width |
| HDATA\_WIDTH | Data width |

* Global signal

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal name** | **Type** | **Width** | **Description** |
| HCLK | input | 1 | Common clock |
| HRESETn | input | 1 | Reset active low-level |

* Connect with ahb master

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal name** | **Type** | **Width** | **Description** |
| HCLK | input | 1 | Common clock |
| HRESETn | input | 1 | Reset active low-level |
| mst\_HRDATA\_o | output | HDATA\_WIDTH | Read data from slave to master |
| mst\_HREADYOUT\_o | output | 1 | Ready feedback from selected slave |
| slv\_HTRANS\_i | input | 2 | Transfer type |
| slv\_HBURST\_i | input | 3 | Transfer burst |
| slv\_HSIZE\_i | input | 3 | Transfer size |
| slv\_HWRITE\_i | input | 1 | Write select |
| slv\_HADDR\_i | input | HADDR\_WIDTH | Address bus |
| slv\_HWDATA\_i | input | HDATA\_WIDTH | Write data bus |
| slv\_HMASTLOCK\_i | input | 1 | Transfer master lock (1) |
| slv\_HPROT\_i | input | 7 | Transfer protect (1) |
| slv\_HNONSEC\_i | input | 1 | Transfer non-secure (1) |
| slv\_HEXCL\_i | input | 1 | Transfer exclusive access monitor (1) |
| slv\_HMASTER\_i | input | 4 | Transfer exclusive access monitor(1) |
| slv\_HREADY\_i | input | 1 | Ready feedback to selected slave |

* Connect with ahb slave

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal name** | **Type** | **Width** | **Description** |
| slv\_HTRANS\_o | output | 2 | Transfer type |
| slv\_HBURST\_o | output | 3 | Transfer burst |
| slv\_HSIZE\_o | output | 3 | Transfer size |
| slv\_HWRITE\_o | output | 1 | Write select |
| slv\_HADDR\_o | output | HADDR\_WIDTH | Address bus |
| slv\_HWDATA\_o | output | HDATA\_WIDTH | Write data bus |
| slv\_HMASTLOCK\_o | output | 1 | Transfer master lock (1) |
| slv\_HPROT\_o | output | 7 | Transfer protect (1) |
| slv\_HNONSEC\_o | output | 1 | Transfer non-secure (1) |
| slv\_HEXCL\_o | output | 1 | Transfer exclusive access monitor (1) |
| slv\_HMASTER\_o | output | 1 | Transfer exclusive access monitor(1) |
| slv\_HREADYOUT\_o | output | 1 | Ready feedback from current master |
| slv\_HRDATA\_i | input | HDATA\_WIDTH | Read data bus |
| slv\_HREADY\_i | input | 1 | Ready signal |
| slv\_HRESP\_i | input | 1 | Response signal |
| slv\_HEXOKAY\_i | input | 1 | Exclusive okay |
| slv\_HADDR\_base\_i | input | HADDR\_WIDTH | Slave address base |
| slv\_HADDR\_mask\_i | input | HADDR\_WIDTH | Slave address mask |

* Note

1. Current version not support these features, just directly send from master to slave.

# Design

## Masterport

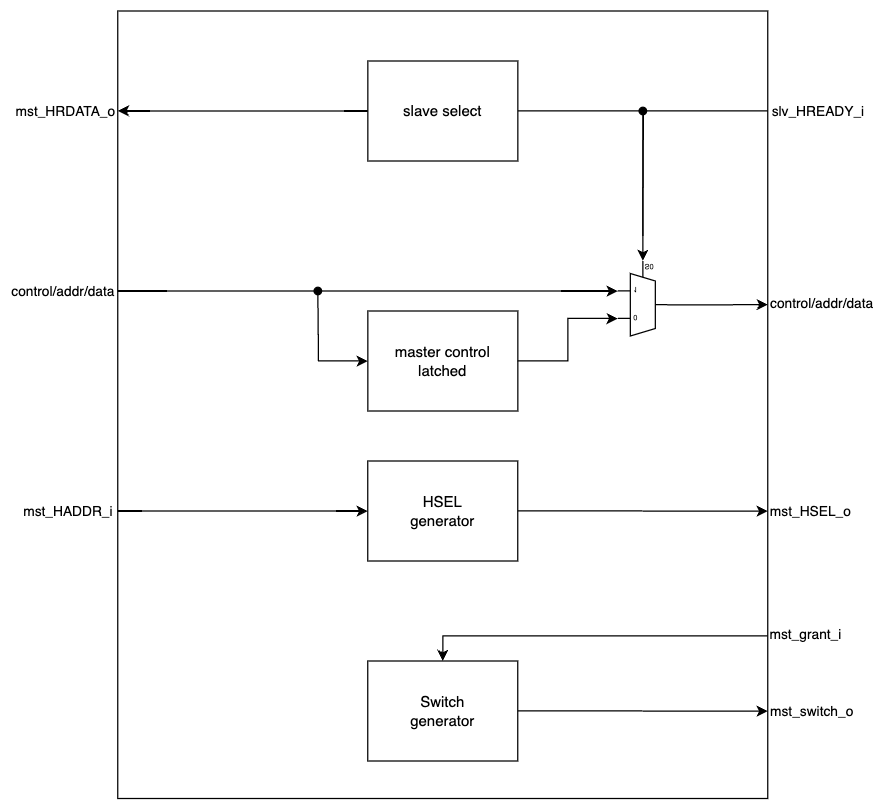


Figure 5-Block diagram of Masterport

* Master control latched

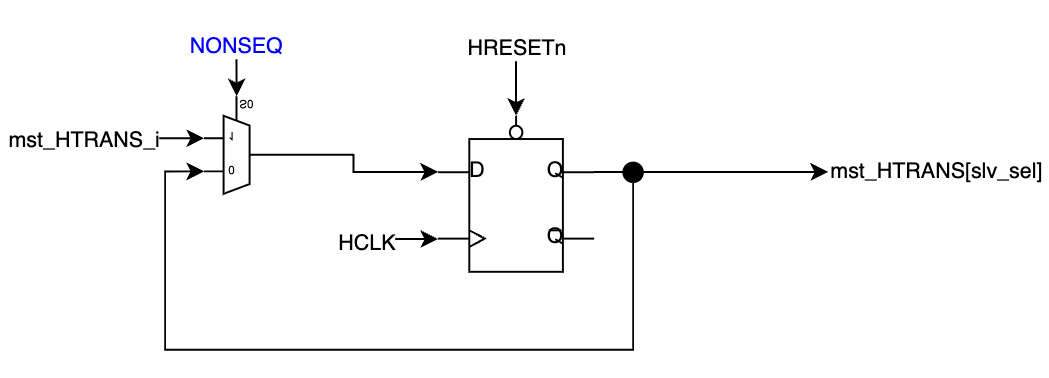


Figure 6-Master control latched

Masterport stores control/address/data when received NONSEQ command, the first transfer. These datas will be sent to slaveports if requested slave is busy.

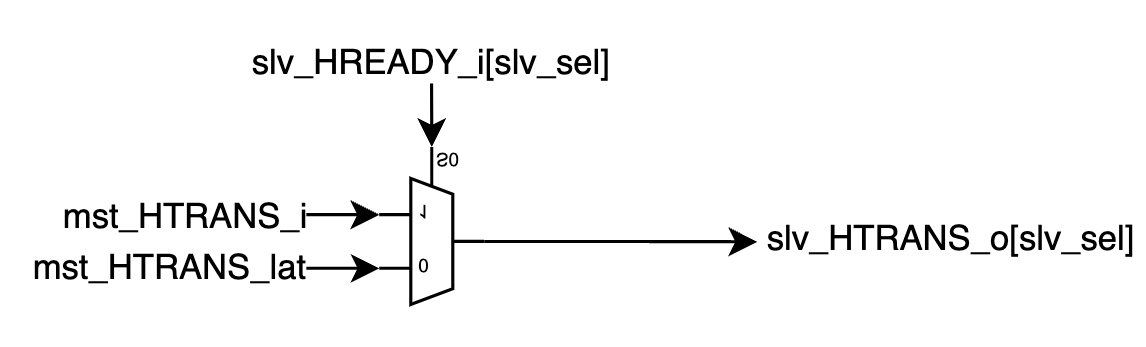


Figure 7-Output control/addr/data from master to slave

* HSEL generator

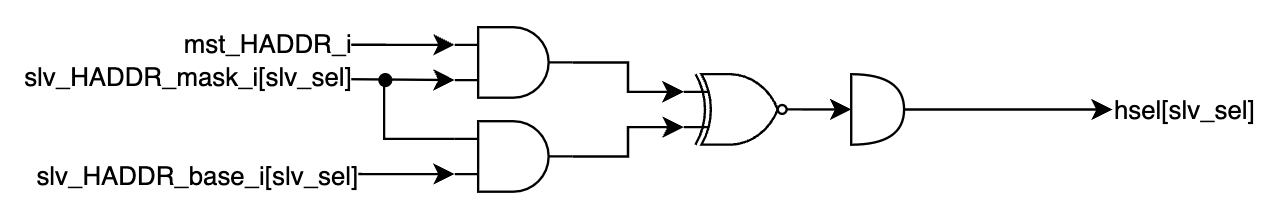


Figure 8-HSEL generator

* Master switch generator: Base on current slave HREADY and granted master to genera switch pulse at new transfer or end of transfer.

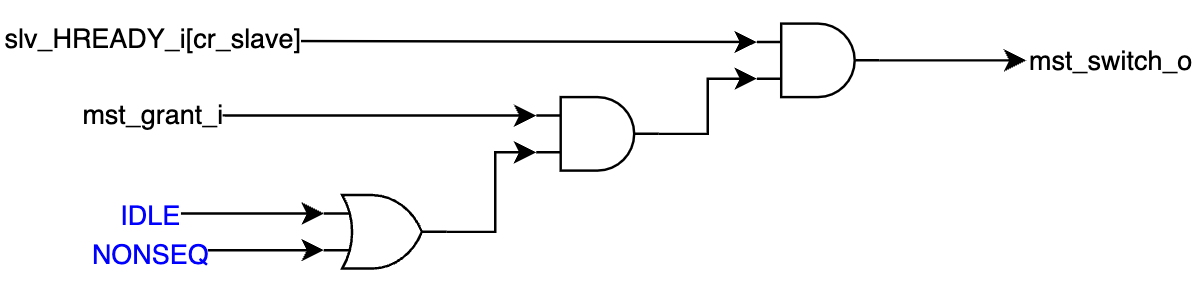


Figure 9-switch generator

## Slaveport

* Arbiter

