Verification Report

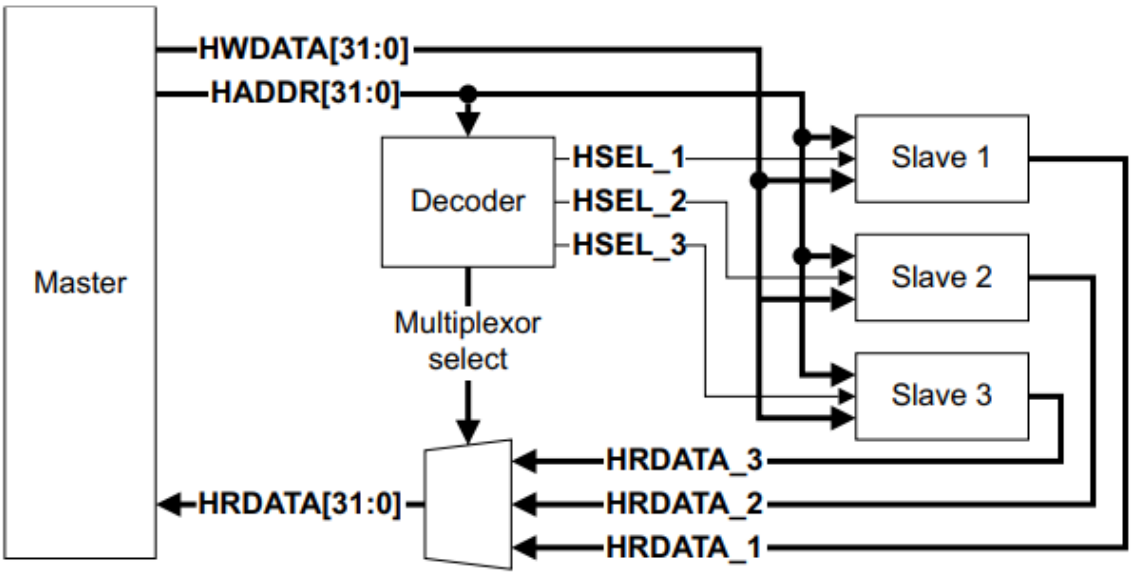
AHB-Lite Protocol

Introduction to the Device-Under-Test (DUT)

AMBA AHB-Lite addresses the requirements of high-performance synthesizable designs. It is a bus interface that supports a single bus master and provides high-bandwidth operation.
AHB-Lite implements the features required for high-performance, high clock frequency systems including:

- burst transfers
- Single-clock edge operation
- Non-tristate implementation
- Wide data bus configurations, 64, 128, 256, 512, and 1024 bits.

The most common AHB-Lite slaves are internal memory devices, external memory interfaces, and high bandwidth peripherals. Although low-bandwidth peripherals can be included as AHB-Lite slaves, for system performance reasons they typically reside on the AMBA Advanced Peripheral Bus (APB). Bridging between this higher level of bus and APB is done using an AHB-Lite slave, known as an APB bridge.
The figure shown below describes a single master AHB-Lite system design with one AHB-Lite master and three AHB-Lite slaves. The bus interconnect logic consists of one address decoder and a slave-to-master multiplexor. The decoder monitors the address from the master so that the appropriate slave is selected and the multiplexor routes the corresponding slave output data back to the master.



Verification Plan

No.	Feature	Test Description	Ref.	Type	Result	Comments
1.	Enable the value of **HWRITE**	**HWRITE** controls the direction of data transfer to or from the master	IHI0033.pdf/Sec3.1 /pg3-2	T	• **HWRITE** is HIGH, it indicates a write transfer and the master broadcasts data on the write data bus, **HWDATA[31:0]**	The simplest transfer is one with no wait states, so the transfer consists of one address cycle and one data cycle
2.	Disable the value of **HWRITE**	**HWRITE** controls the direction of data transfer to or from the master	IHI0033.pdf/Sec3.1 /pg3-2	T	**HWRITE** is LOW, a read transfer is performed and the slave must generate the data on the read data bus, **HRDATA [31:0]**.	The simplest transfer is one with no wait states, so the transfer consists of one address cycle and one data cycle
3.	Enable **HMASTLOCK** signal	If the master requires locked accesses then it must assert the **HMASTLOCK** signal	IHI0033.pdf/Sec3.3 /pg3-7	T	This signal indicates to any slave that the current transfer sequence is indivisible and must therefore be processed before any other transactions are processed	Typically the locked transfer is used to maintain the integrity of a semaphore, by ensuring that the slave does not perform other operations between the read and write phases of a microprocessor SWP instruction
4.	Bus termination after a BUSY transfer for an undefined length	After a burst has started, the master uses BUSY transfers if it requires more time before continuing with the next transfer in the burst.	IHI0033.pdf/Sec3.5 .1/pg3-10		During an undefined length burst, INCR, the master might insert BUSY transfers and then decide that no more data transfers are required. Under these circumstances, it is acceptable for the master to then perform a NONSEQ or IDLE transfer that then effectively terminates the undefined length burst	The protocol does not permit a master to end a burst with a BUSY transfer for fixed length bursts of type incrementing or wrapping
5.	Bus termination after a BUSY	After a burst has started, the master uses BUSY transfers if	IHI0033.pdf/Sec3.5 .1/pg3-10	T	The fixed length burst types must terminate with a SEQ transfer	The master is not permitted to perform a BUSY transfer immediately after a SINGLE burst.

	transfer for a SINGLE length	it requires more time before continuing with the next transfer in the burst.				SINGLE bursts must be followed by an IDLE transfer or a NONSEQ transfer.
6.	Change in transfer type during **IDLE** Transfer	Slaves use **HREADY** to insert wait states if they require more time to provide or sample the data. During a waited transfer, the master is restricted to what changes it can make to the transfer type and address. During a waited transfer, the master is permitted to change the transfer type from IDLE to NONSEQ	IHI0033.pdf/Sec3.6.1/pg3-16	T	When the **HTRANS** transfer type changes to NONSEQ the master must keep **HTRANS** constant, until **HREADY** is HIGH	-
7.	Change in transfer type during **BUSY** Transfer, Fixed Length Burst	During a waited transfer for a fixed length burst, the master is permitted to change the transfer type from BUSY to SEQ	IHI0033.pdf/Sec3.6.1/pg3-17	T	When the **HTRANS** transfer type changes to SEQ the master must keep **HTRANS** constant, until **HREADY** is HIGH	Because **BUSY** transfers must only be inserted between successive beats of a burst, this does not apply to **SINGLE** bursts
8.	Change in transfer type during **BUSY** Transfer, Undefined Length Burst	During a waited transfer for an undefined length burst, INCR, the master is permitted to change from BUSY	IHI0033.pdf/Sec3.6.1/pg3-18	T	The burst continues if a SEQ transfer is performed but terminates if an IDLE or NONSEQ transfer is performed	-

		to any other transfer type, when **HREADY** is LOW				
9.	Change in address during **IDLE** Transfer	When the slave is requesting wait states, the master can only change the address once except in case of an IDLE Transfer. During a waited transfer, the master is permitted to change the address for IDLE transfers	IHI0033.pdf/Sec3.6.2/pg3-19	T	When the **HTRANS** transfer type changes to NONSEQ the master must keep the address constant, until **HREADY** is HIGH	-
10.	Change in address after an **ERROR** response	When the slave is requesting wait states, the master can only change the address once except after an **ERROR** response	IHI0033.pdf/Sec3.6.2/pg3-20	T	During a waited transfer, if the slave responds with an **ERROR** response then the master is permitted to change the address when **HREADY** is LOW	-
11.	Transfer done	After a master has started a transfer, the slave controls how the transfer progresses. A master cannot cancel a transfer after it has commenced. A slave must provide a response that indicates the status	IHI0033.pdf/Sec5.1.1/pg5-3	T	A successful completed transfer is signalled when **HREADY** is HIGH and **HRESP** is OKAY (0)	-

		of the transfer when it is accessed. The transfer status is provided by the **HRESP** signal				
12.	Transfer pending	A slave uses **HREADY** to insert the appropriate number of wait states into the data phase of the transfer.	IHI0033.pdf/Sec5.1.2/pg5-3	T	In pending transfer, **HREADY** is LOW and **HRESP** is OKAY (0). A transfer then completes with **HREADY** HIGH and an OKAY response	-
13.	ERROR response	A slave uses the **ERROR** response to indicate some form of error condition with the associated transfer. Usually this denotes a protection error such as an attempt to write to a read-only memory location	IHI0033.pdf/Sec5.1.3/pg5-3	T	The slave will give an ERROR response in the first cycle with **HREADY** LOW and **HRESP** ERROR (1). In the next cycle, **HREADY** is driven HIGH and **HRESP** remains HIGH to indicate error	The two-cycle response is required because of the pipelined nature of the bus. By the time a slave starts to issue an ERROR response then the address for the following transfer has already been broadcast onto the bus. The two-cycle response provides sufficient time for the master to cancel this next access and drive **HTRANS**[1:0] to IDLE before the start of the next transfer

Explanation of Different Fields

No. The serial number of the test.
Feature The feature which the current test is verifying in full or partially. The feature is usually on the abstraction level of a user.
Test Description A detailed description of the test case being performed. You can be as verbose as you want.
Ref. Reference to the section in the related standard document. The section number as well as page numbers should be described here.

Type	Type of the test. Whether the test is an assertion (A) or a transaction (T) type.
Result	Pass (P) or Fail (F).
Comments	Any other comments about the test or its results that you want to mention.