AMBA AHB-Lite Slave Protocol Verification Plan

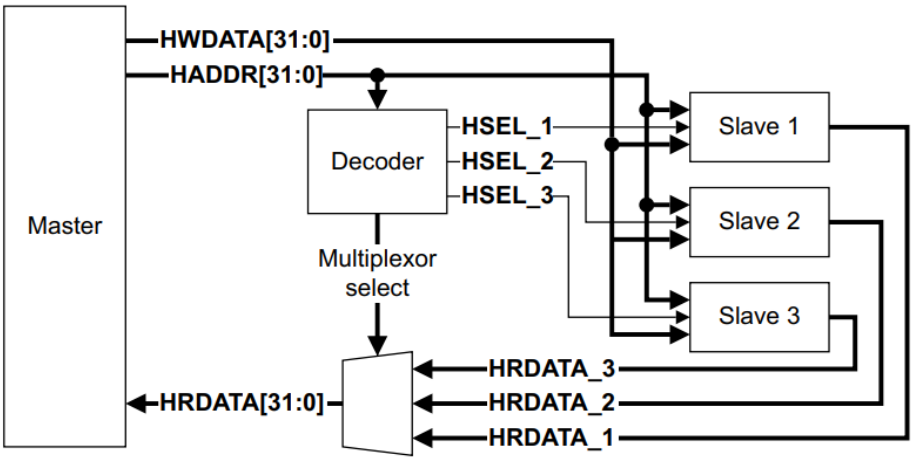
December 13, 2023.

Submitted By:

Wajahat Riaz
Zeeshan Khan
Muhammad Ahad
Sami Ullah Saqib

AHB-Lite Protocol:

Advance High Performance-Lite (AHB-lite) is a bus interface that supports a single bus master and provides high bandwidth operations. The most common slaves used for this protocol are internal memory devices, external memory interfaces, and high-bandwidth peripherals. The figure below illustrates an AHB-Lite system design with one master and multiple slaves.



The main components of the AHB-Lite system are as follows:
1) Master
2) Slave
3) Decoder
4) Multiplexor

An AHB-Lite master provides address and control information to initiate read and write operations. The slave responds to transfers initiated by masters in the system. The slave uses the select signal from the decoder to control when it responds to a bus transfer. The slave signals back to the master i.e., the success, failure, or waiting of the data transfer. This component decodes the address of each transfer and provides a select signal for the slave that is involved in the transfer. It also provides a control signal to the multiplexor. A slave-to-master multiplexor is required to multiplex the read data bus and response signals from the slaves to the master.

Working of Protocol:

The master starts a transfer by driving the address and control signals. These signals provide information about the address, direction, and width of the transfer, and indicate if the transfer forms part of a burst. Transfers can be of different types for instance single, incrementing bursts that do not wrap at address boundaries, wrapping bursts that wrap at particular address boundaries, etc. The write data bus moves data from the master to a slave, and the read data bus moves data from a slave to the master.
Every transfer consists of two phases:

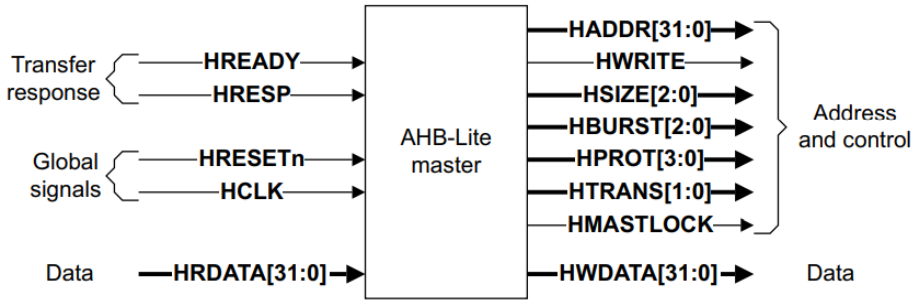
1) Address phase: one address and control cycle
2) Data phase: one or more cycles for the data.

A slave cannot request that the address phase is extended and therefore all slaves must be capable of sampling the address during this time. However, a slave can request that the master extends the data phase by using an HREADY signal. This signal, when LOW, causes wait states to be inserted into the transfer and enables the slave to have extra time to provide or sample data. The slave uses a response signal to indicate the success or failure of a transfer.

Global Signals:

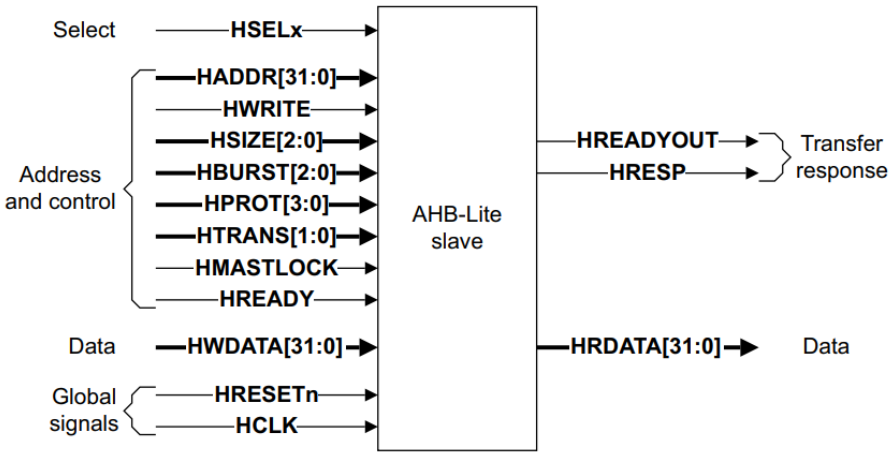
Name	Destination	Description
HCLK	Clock Source	The clock source for all operations on the protocol. Input signals are sampled at the rising edge and changes in output signals happen after the rising edge
HRESTn	Reset Controller	Asynchronous primary reset for all bus elements

Master Signals:



Name	Destination	Description
HADDR [31:0]	Slave and Decoder	Address bus of 32 bits
HBURST [2:0]	Slave	Indicates the type of burst signal including wrapping and incrementing bursts
HSIZE [2:0]	Slave	Indicates the size of transfer from 8 bits to 1024 bits
HPROT [3:0]	Slave	Protection control signal providing additional information like opcode fetch, data fetch, etc.

Slave Signals:



Name	Destination	Description
HRDATA [31:0]	Multiplexor	Read data bus to transfer the data from a Slave's location to the Master via multiplexor
HREADYOUT	Multiplexor	Indicates transfer has finished on the bus and is used to extend the data phase
HRESP	Multiplexor	Provides additional information on whether the transfer was successful or failed

Decoder Signals:

Name	Destination	Description
HSELx Note: x is a unique identifier for AHB lite slave	Slave	Indicates current transfer is intended for the selected slave

Multiplexor Signals:

Name	Destination	Description
HRDATA [31:0]	Master	Read data bus to rout to Master

HREADY	Master and Slave	Indicates completion of previous transfer
HRESP	Master	Transfer response

Software/Tools:

1) Cadence Xcelium
2) Cadence Incisive
3) Google Docs

Verification Plan:

No.	Feature	Test Description	Ref.	Type	Result	Expected Outcome	Comments
1	Read followed Write transfer at a fixed address	A random address, say A, is driven onto the bus for a write transaction followed by a read transaction.	3.1 5.1 5.1.2	TR	PASS	HRDATA must be equal to HWDATA.	Each transaction read or write must have a separate address and data phase.
2	Global Signal: HRESTn	Before continuous read transfers, an HRESTn is driven low asynchronously.	7.1.2	TR	PASS	All bus elements will initialize to valid levels. Read transfers will read initialized values from the slave memory	N/A
3	Slave response: HSEL = 0	Performing a read or write transfer given that slave is not selected during transactions	5.1.3	A	PASS	The effect of transfers has no effect on the DUT since it's not connected. No response is given	
4	Master Signal: IDLE HTRANS [1:0] =b00	Some addresses A, B, and C are driven onto the bus. An IDLE transfer is inserted to these addresses.	3.2	TR	PASS	No data transfers happen.	A master uses an IDLE transfer when it does not want to perform a data transfer.
5	Master Signal: BUSY HTRANS [1:0] =b01	The BUSY transfer type enables masters to insert idle cycles in the middle of a burst. This	3.2	TR	PASS	The master is unable to perform the transfer and delays the start of the current transaction.	N/A

		transfer type indicates that the master is continuing with a burst, but the next transfer cannot take place immediately.					
6	Transfer type changes from IDLE to NONSEQ	Address A and B are driven onto the bus. Half of the transactions are IDLE, and the other half is NONSEQ implemented in an interleaved manner	3.6.1	A	PASS	The slave will sample address A at the rising clock edge of the address phase. After a successful transfer to address A the slave will ignore the IDLE transfers i.e., transfers associated with address B will be neglected.	
7	Slave response: ● Transfer done ● Transfer pending	An address A is driven onto the bus with three waiting-for states. This is done by keeping the HREADY low during the data phase of address A. Then, an address B is driven onto the bus with zero wait states. The HRESP and HREADY signals are monitored	5.1.1 5.1.1	A	PASS	During the waiting states, the HRESP is low, and HREADY is also low, which indicates that the transfer is pending. After the wait states, the transfer of address A is signaled, i.e., HRESP is high, and HREADY is high.	These assertions were implicit in the previous tests.
8	Protection signals HPROT [3:0]	A read or write transaction is performed at an address A. Setting different values for the protection signal HPROT to test	3.7	C		The response is entirely dependent on how the design engineer implemented it.	Not enough information was provided in this regard.

		for all possible combinations which determine (not) cacheable (not) bufferable, (un) privileged access only.					Some implementations can be added to coverage items, therefore marked as C.
9	Four-beat incrementing burst, INCR4	Write transfers followed by read transfers four-beat incrementing burst,	3.5.3	TR	PASS	The address does not wrap at a 16-byte boundary.	
10	Eight-beat incrementing burst, INCR8	Write transfers followed by read transfers eight-beat incrementing burst,	3.5.3	TR	PASS	The transfers does not wrap around a 32-byte boundary.	
11	Four-beat wrapping burst WRAP4	Write transfers followed by read transfers four-beat wrapping	3.5.3	TR	PASS	The transfers wrap around a 16-byte boundary.	
12	Eight-beat wrapping burst, WRAP8	The burst is an eight-beat burst of word transfers; the address wraps at 32-byte boundaries,	3.5.3	TR	PASS	The transfers wrap around a 32-byte boundary.	

Explanation of Different Fields

No. The serial number of the test.

Feature: The feature that 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 is 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), a transaction (T), or possibly a coverage(C) type.

Result Indicates that the given module has passed or failed the test.

Comments :

Coverage Items:

No.	Feature	Covergroup Description
1	Covergroup for HSIZE coverage	Basic and corner cases
2	Covergroup for HTRANS coverage	Basic and corner cases
4	Covergroup for HBURST coverage	Basic and corner cases
5	Covergroup for Slave’s response signals	Basic and corner cases
6	Covergroup for address alignment	Basic and corner cases

Assertions:

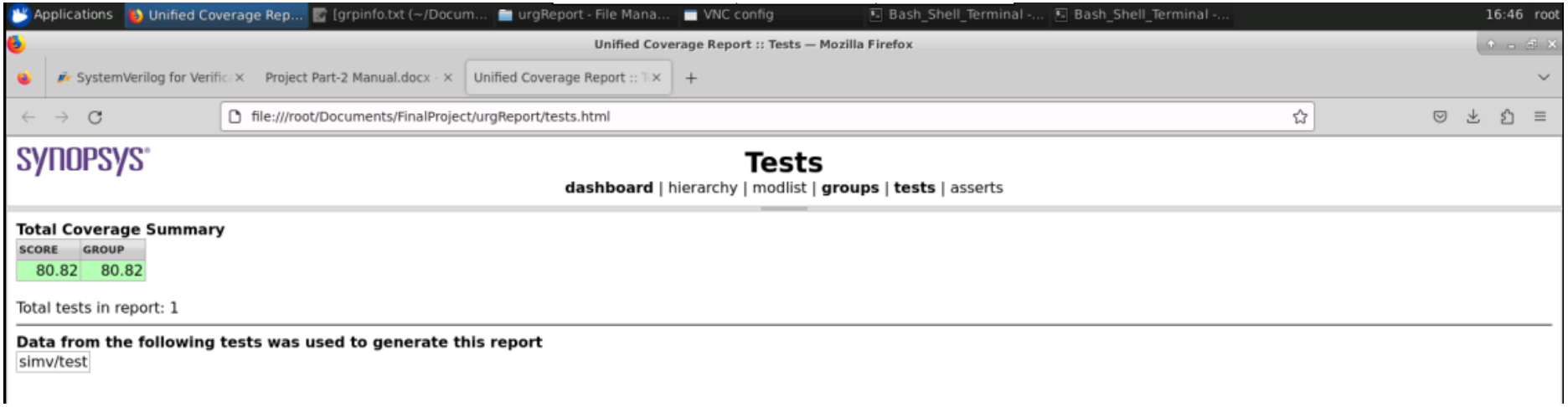
No.	Feature	Assertion Description
1	Assertion for HSIZE property	Basic and corner cases
2	Assertion for HTRANS property	Basic and corner cases
3	Assertion for HPROT property	Basic and corner cases
4	Assertion for HBURST property	Basic and corner cases
5	Assertion for Slave's response signals	Basic and corner cases
6	Assertion for address property	Basic and corner cases

Coverage Report:

Type the following command to generate a coverage report using Playground:

```
urg -lca -dir simv.vdb
```

For half of the test scenarios, I have achieved 80% coverage which includes toggle coverage as well.
Note: Since the license for Cadence-based tools ended, it was really hard to look into this further.


Applications
Unified Coverage Rep...
[grpinfo.txt (~/Docum...
urgReport - File Mana...
VNC config
Bash_Shell_Terminal -...
Bash_Shell_Terminal -...
16:46 root
Unified Coverage Report :: Tests — Mozilla Firefox
SystemVerilog for Verific
Project Part-2 Manual.docx
Unified Coverage Report ::
file:///root/Documents/FinalProject/urgReport/tests.html
SYNOPSYS
Tests
dashboard | hierarchy | modlist | groups | tests | asserts
Total Coverage Summary
SCORE GROUP
80.82 80.82
Total tests in report: 1
Data from the following tests was used to generate this report
simv/test