**UNIVERSITY OF ENGINEERING AND TECHNOLOGY  
 LAHORE, PAKISTAN**

****

**AMBA AHB-Lite Protocol Verification Plan**

March 31, 2022.

**Submitted By:**

Wajahat Riaz

2021-MS-EE-36

**Department of Electrical Engineering**

**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 given below illustrates an AHB-Lite system design having one master and multiple slaves.

HWDATA [31:0]

HADDR [31:0]

**Slave** 1

HSEL\_1

**Decoder**

HSEL\_2

**Master** **Slave** 2

HSEL\_3

**Slave** 3

HRDATA\_3

HRDATA\_2

HRDATA [31:0]

HRDATA\_1

**Multiplexer**

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, 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 a 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 signalto 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 |

**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) QuestaSim  
2) Github  
3) Microsoft Word

# **Verification Plan:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Feature** | **Test Description** | **Ref.** | **Type** | **Result** | **Expected Outcome** | **Comments** |
| 1 | Write Transfer from Master to Slave | When HWRITE is high then the Master will broadcast the data on the HWDATA [31:0] bus for individual burst types i.e., HBURST [2:0] including incrementing and wrapping types. | 3.1 | TR |  | Successful write. HRESP should be low and HREADY should be high |  |
| 2 | Read Transfer from Slave to Master | When HWRITE is low then the slave must generate the data on the HRDATA [31:0] bus for individual burst types i.e., HBURST [2:0] including incrementing and wrapping type. | 3.1 | TR |  | Successful read. HRESP should be low and HREADY should be high |  |
| 3 | Continuous writing to the same address location at the same clock edge | When HWRITE is high then the Master will broadcast the data packets on the HWDATA [31:0] bus. | 3.1 | TR |  | Successful write. HRESP should be low and HREADY should be high for the successive data packets |  |
| 4 | Continuous reading from the same address location at the same clock edge | When HWRITE is low then the slave must generate the data packets on the HRDATA [31:0] bus. | 3.1 | TR |  | Successful read. HRESP should be low and HREADY should be high for the successive data packets |  |
| 5 | Random Write transfers | When HWRITE is high then the Master will broadcast the data packets on the HWDATA [31:0] bus. | 3.1 | TR |  | Successful write. HRESP should be low and HREADY should be high for the successive data packets |  |
| 7 | Random Read Transfers | When HWRITE is low then the slave must generate the data packets on the HRDATA [31:0] bus. | 3.1 | TR |  | Successful read. HRESP should be low and HREADY should be high for the successive data packets |  |
| 8 | Write-Read Transfer | Write transfer followed by Read transfer at a particular address A. | 3.1 | TR |  | HRESP is low, HREADY is high and the address location must have the updated value |  |
| 9 | Read-Write Transfer | Read transfer followed by Write transfer at a particular address A. | 3.1 | TR |  | HRESP is low, HREADY is high and the slave must return the previous Data (A). |  |
| 10 | Global Signal: HCLK | A clock signal is generated in the top module | 7.1.1 | A |  | All input signals must be sampled at the rising edge of the clock and changes in the output signals must occur after the rising edge. |  |
| 11 | Global Signal:  HRESTn | Since this is an active low signal. When asserted then it must reset all bus elements. Note: Slaves must ensure that HREADYOUT is high. HTRANS [1:0] must indicate IDLE. | 7.1.2 | TR |  | All previous binary information in the bust elements will be lost. |  |
| 12 | Master Signal: IDLE HTRANS [1:0] =b00 | When the IDLE transfer is inserted to an address. | 3.2 | A |  | The transfer must be ignored by the slave. Slaves must provide a zero-wait OKAY response. |  |
|  | Master Signal: BUSY HTRANS [1:0] =b01 | When a BUSY transfer is inserted then the address and control signals must reflect the next burst transfer |  | A |  | Slaves must always provide a zero-wait state OKAY. The transfer must be ignored by the slave |  |
|  | Write transfer extended | If the transfer is extended  then the master must hold the data valid until the transfer completes, as indicated by HREADY high. |  | A |  |  |  |
|  | Ready transfer extended | If the slave extends  the read transfer by holding HREADY low then the slave only has to provide valid data in the final cycle of the transfer, as indicated by HREADY high |  | A |  |  |  |
| 13 | Transfer type changed during waited states: Scenario 1 | Transfer type changes from IDLE to NONSEQ during waited states. The HTRANS signal must be kept constant after the transition until HREADY is high | 3.6.1 | A |  | Successfully transfer type changed. Slaves must give the OKAY response. |  |
| 14 | Transfer type changed during waited states: Scenario 2 | Transfer type changes from BUSY to SEQ during waited states for fixed-length bursts. The HTRANS signal must be kept constant after the transition until HREADY is high | 3.6.1 | A |  | Successfully transfer type changed. Slaves must give the OKAY response. |  |
| 15 | Transfer type changed during waited states: Scenario 3 | Transfer type changes from BUSY to any other type during waited states for undefined length burst. The burst continues if an SEQ transfer is performed but terminates if an IDLE or NONSEQ transfer is performed. | 3.6.1 | A |  | Successfully transfer type changed. Slaves must give the OKAY response. |  |
| 17 | Bus Termination |  |  |  |  |  |  |
| 18 | Slave response:  Transfer done | A successful completed transfer is signalled when HREADY is high and HRESP is  OKAY. | 5.1.1 | A |  |  |  |
| 19 | Slave response:  Transfer pending | A typical slave uses HREADY to insert the appropriate number of wait states into the  data phase of the transfer. The transfer then completes with HREADY HIGH and an  OKAY response to indicate the successful completion of the transfer. | 5.1.2 | A |  | When a slave inserts a number of wait states prior to completing the response, it must  drive HRESP to OKAY. |  |
| 20 | Slave response:  Transfer failed | In the first cycle, to start the ERROR response, the slave drives HRESP high to indicate ERROR while driving HREADY low to extend the transfer for one extra  cycle.  In the next cycle HREADY is driven high to end the transfer and HRESP remains driven high to indicate ERROR. | 5.1.3 | A |  | The ERROR response requires two cycles. |  |

## 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 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) or a transaction (T) type. |
| **Result** | Indicates that the given module has passed (P) or failed (F) the test. |
| **Comments** | Any other comments about the test or its results that you want to mention. |