**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 | An address A is driven onto the bus. The slave will sample the address A on the next rising clock edge.  Afterward, the slave will drive the HREADY response. This response is sampled on the next rising edge of HCLK. | 3.1 | TR |  | Address phase should not be more than one cycle.  The slave must only sample address when HREADY is high.  The Data(A) must be written at the address A and a completed transfer is signaled i.e., HRESP should be low and HREADY should be high. | HWRITE is high, indicating a write transfer and the master broadcasts data on the write data bus, HWDATA [31:0]. |
| 2 | Read transfer from slave to master | An address A is driven onto the bus. The slave will sample the address A on the next rising clock edge.  Afterward, the slave will drive the HREADY response. This response is sampled on the next rising edge of HCLK. | 3.1 | TR |  | The address phase should not be more than one cycle.  The slave must only sample address on when HREADY is high.  The Data(A) must be read from the address A and completed transfer is signaled i.e., HRESP should be low and HREADY should be high. | HWRITE is low, a read transfer is performed and the slave must generate the data on the read data bus, HRDATA [31:0]. |
| 3 | Random transfers | Random addresses A, B, C and D with zero wait states are driven onto the bus.  The slave will sample the addresses A, B, C and D on rising clock edged of their address phase. | 3.1 | TR |  | Just like in test 1 and test 2 the slave must only sample the address A, B, C and D when HREADY is high and completion of transfer must be signaled by the slave i.e., HRESP should be low and HREADY should be high.  Based on the basic transfer type i.e., write or read Data(A), Data(B), Data(C) and Data(D) will be driven on the HWDATA [31:0] bus or HRDATA [31:0] bus respectively. | Based on the type of transfer i.e., read transfer or write transfer HWRITE will be set low and high respectively. |
| 4 | Read or Write transfer with wait states | An address A is driven onto the bus. The slave will sample the address on the rising edge of the clock provided that HREADY is high.  After sampling the address. Wait states are added in the data phase by keeping HREADY low for two cycles after we have sampled the address | 3.1  5.1  5.1.2 | A |  | During the wait state, the slave must provide transfer pending response i.e., HREADY and HRESP must be low before completion.  Afterward, a successful complete transfer is signaled when HREADY is high and HRESP is low. | Adding wait states causes latency in the read or write transfer. The master can not cancel the transfer. |
| 5 | Multiple transfers extended | Three addresses A, B, and C are driven onto the bus. The addresses are sampled on rising clock edges during their address phases.  Wait states are added using HREADY. Transfer to address A is one wait state. Transfer to address B is two wait states. | 3.1 | TR |  | Since the data phase of address, A is extended the address phase of B is extended by one cycle.  The address phase of C is extended by three cycles. | When a transfer is extended it has side effects of extending the address phase of the next transfer. |
| 6 | Write followed by Read transfer | An address A is driven onto the bus twice. The slave will sample the address A on the first and second rising clock edge for write and read transfer respectively.  Firstly, HWRITE is high, indicating a write transfer and the master broadcasts data on the write data bus, HWDATA [31:0].  Lastly, HWRITE is set low, a read transfer is performed and the slave must generate the data on the read data bus, HRDATA [31:0]. | 3.1 | TR |  | Based on the specifications of the memory. Read transfer should produce the updated Data(A). | During the Write transfer the completed transfer is signaled i.e., HRESP should be low and HREADY should be high.  During the Read transfer the completed transfer is signaled i.e., HRESP should be low and HREADY should be high. |
| 7 | Read followed by Write transfer | An address A is driven onto the bus twice. The slave will sample the address A on the first and second rising clock edge for read and write transfer respectively.  Firstly, HWRITE is set low, a read transfer is performed and the slave must generate the data on the read data bus, HRDATA [31:0].  Lastly, HWRITE is high, indicating a write transfer and the master broadcasts data on the write data bus, HWDATA [31:0]. | 3.1 | TR |  | Based on the specifications of the memory. Read transfer should produce the Data(A) which was stored before Write transfer.  The Write transfer will update Data(A). | During the Read transfer the completed transfer is signaled i.e., HRESP should be low and HREADY should be high.  During the Write transfer the completed transfer is signaled i.e., HRESP should be low and HREADY should be high. |
| 8 | Wrapping burst types:   * WRAP4 * WRAP8 * WRAP16 | We have transfer size of 4-byte (32-bit) which is a word.  In WRAP4, firstly address A+4, address A+8 and address A+12 is driven onto the bus which are sampled by the slave on the rising clock edges of their address phases. After the transfer at address A+12, we have reached the address boundary therefore next transfer is wrapped to address A.  Similarly, we drive addresses on the bus for WRAP8 and WRAP16 to check if they wrap at the address boundaries. | 3.5  3.5.3 | A |  | In WRAP4 the burst is a four-beat burst of word transfers; the addresses wrap at 16-byte boundary.  In WRAP8 the burst is an eight-beat burst of word transfers; the addresses wrap at 32-byte boundary.  In WRAP16 the burst is a sixteen-beat burst of word transfers; the addresses wrap at 64-byte boundary.  For all the above scenarios the slave will provide a completed transfer signal. | Wrapping bursts wrap when they cross an address boundary.   Address boundary = HBURST x HSIZE  Note: Different combinations of read and write transfers can be used which were implicitly checked in previous tests. |
| 9 | Incrementing burst type:   * INCR4 * INCR8 * INCR16 | We have transfer size of 4-byte (32-bit) which is a word.  Firstly, address A+4, A+8 and A+12 is driven onto the bus which are sampled by the slave on the rising clock edges of their address phases. After the transfer at address A+12, we have reached the address boundary.  Since we are using incrementing burst type. Instead of wrapping around it will transfer to the next location which is A+16.  Similarly, we drive addresses on the bus for INCR8, INCR 16 to check if they increment at the address boundaries. |  |  |  | In INCR4, the transfers are incremented by 4.  In INCR8, the transfers are incremented by 8.  In INCR16, the transfers are incremented by 16.  For all the above scenarios the slave will provide a completed transfer signal. | Incrementing bursts access sequential locations. The addresses of each transfer in the burst are an increment of the previous address. |
| 10 | Incrementing burst type:  INCR and undefined length burst | First burst is driven on the bus which consists of two halfword transfers at an address A.  The second burst is read consisting of three word read transfers starting at address B |  |  |  | In first burst, the transfer address is incremented by two.  In the second burst, the transfer address is incremented by four.  For all the above scenarios the slave will provide a completed transfer signal. |  |
| 11 | Protection signals HPROT [3:0]:   * 4’b0000 * 4’b1111 | An address A is driven on the bus. The timing of HPROT and address bus must be same. The must remain constant throughout the burst transfer.  The protection signal HPROT [3:0] = 4’b0000 corresponds to non-cacheable, non-bufferable, unprivileged opcode fetch.  The protection signal HPROT [3:0] = 4’b1111 corresponds to cacheable, bufferable, privileged data access. | 3.7 | A |  | The protection signal basically gives extra information which can be used to determine an exception for instance illegal instruction, illegal access and etc.  For instance, data(A) can’t be accessed because only a privileged level can access that information.  The response is entirely dependent how the design engineer implemented it. | The test is dependent on master’s ISA (Instruction Set Architecture) and design.  Used by a module that wants to implement some level of protection. |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 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 |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 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. |  |
|  |  |  |  |  |  |  |  |
| 18 | Slave response:  Transfer done | A successfully 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. |