AHB-Lite Protocol Verification

EE-599f SoC System Verilog

Submitted by:

Sawera Zahra

Reg ID:

2021-MSEE-37

Lab instructor:

Dr. Ubaid Ullah Fayyaz

|| DEPARTMENT OF ELECTRICAL ENGINEERING ||
UNIVERSITY OF ENGINEERING AND TECHNOLOGY, LAHORE.

Introduction to the Device-Under-Test (DUT)

AMBA3 AHB-lite Protocol

Introduction

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 a AHB-Lite slave, known as an APB bridge.

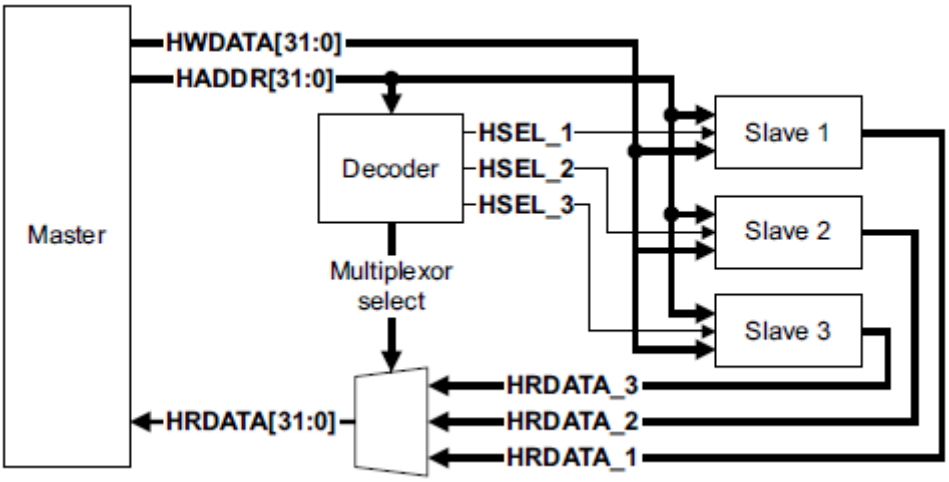

Figure 1 AHB lite block diagram

Components

1. **Master**

In order to perform read and write operations on slaves, the master gives the necessary information. When presenting the arbitration, each slave is chosen via a selection signal as explained below (HSEL). For example, the slave can now clearly distinguish between when it is being stimulated and when it is not. Additionally, it gets two response signals from the slaves, HRESP and HREADY, to determine whether or not a transfer has been completed successfully (HRESP) and whether or not an extension of the data phase is necessary (HREADY)

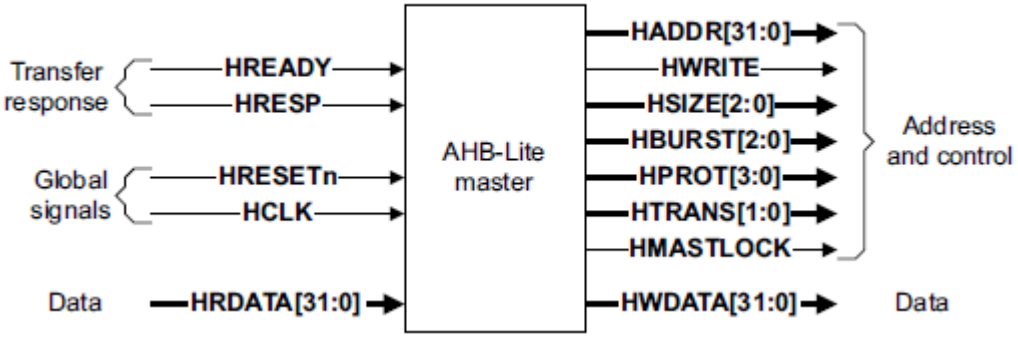

Figure 2 Master interface

2. **Slave**

Whenever a master in the system initiates a transfer, an AHB-Lite slave responds. HSELx select signals from the decoder are used to determine when a bus transfer occurs for the slave. The slave informs the master of the data transfer's success, failure, or delay.

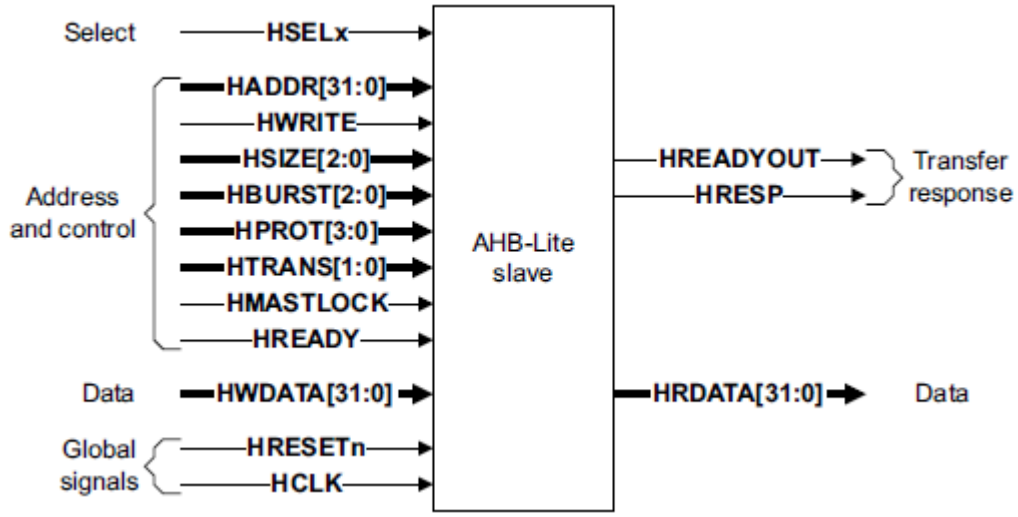

Figure 3 Slave interface

3. **Decoder**

The address of each transfer is decoded by this component, and a select signal is provided for the slave that is involved. The multiplexor receives a control signal from it as well. All AHB-Lite implementations that make use of two or more slaves must have a single centralised decoder.

4. **MUX**

There must be a multiplexer between the slaves and the master in order to multiplex the read data and response signals. The multiplexor is controlled by the decoder. All AHB-Lite implementations employing two or more slaves must make use of a single centralised multiplexor.

Working operation

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:

- single
- incrementing bursts that do not wrap at address boundaries

- wrapping bursts that wrap at particular address boundaries.

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:

- Address phase one address and control cycle
- 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 HREADY. 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 HRESP to indicate the success or failure of a transfer.

Signals

Global signals

Name	Source	Description
HCLK	Clock source	The bus clock times all bus transfers. All signal timings are related to the rising edge of HCLK.
HRESETn	Reset controller	The bus reset signal is active LOW and resets the system and the bus. This is the only active LOW AHB-Lite signal.

Master signals

Name	Destination	Description
HADDR [31:0]	Slave and decoder	The 32-bit system address bus.
HBURST [2:0]	Slave	The burst type indicates if the transfer is a single transfer or forms part of a burst. Fixed length bursts of 4, 8, and 16 beats are supported. The burst can be incrementing or wrapping. Incrementing bursts of undefined length are also supported.
HMASTLOCK	Slave	When HIGH, this signal indicates that the current transfer is part of a locked sequence. It has the same timing as the address and control signals.
HPROT [3:0]	Slave	The protection control signals provide additional information about a bus access and are primarily intended for use by any module that wants to implement some level of protection. The signals indicate if the transfer is an opcode fetch or data access, and if the transfer is a privileged mode access or user mode access. For masters with a memory management unit these signals also indicate whether the current access is cacheable or buffer able.
HSIZE [2:0]	Slave	Indicates the size of the transfer, that is typically byte, halfword, or word. The protocol allows for larger transfer sizes up to a maximum of 1024 bits.
HTRANS [1:0]	Slave	Indicates the transfer type of the current transfer. This can be: • IDLE • BUSY • NONSEQUENTIAL • SEQUENTIAL.
HWDATA [31:0]	Slave	The write data bus transfers data from the master to the slaves during write operations. A minimum data bus width of 32 bits is recommended. However, this can be extended to enable higher bandwidth operation.
HWRITE	Slave	Indicates the transfer direction. When HIGH this signal indicates a write transfer and when LOW a read transfer. It has the same timing as the address signals, however, it must remain constant throughout a burst transfer.

Slave signals

Name	Destination	Description
HRDATA [31:0]	MUX	During read operations, the read data bus transfers data from the selected slave to the multiplexor. The multiplexor then transfers the data to the master. A minimum data bus width of 32 bits is recommended. However, this can be extended to enable higher bandwidth operation.
HREADYOUT	MUX	When HIGH, the HREADYOUT signal indicates that a transfer has finished on the bus. This signal can be driven LOW to extend a transfer.
HRESP	MUX	The transfer response, after passing through the multiplexor, provides the master with additional information on the status of a transfer. When LOW, the HRESP signal indicates that the transfer status is OKAY. When HIGH, the HRESP signal indicates that the transfer status is ERROR.

Decoder signals

Name	Destination	Description
HSELx	Slave	Each AHB-Lite slave has its own slave select signal HSELx and this signal indicates that the current transfer is intended for the selected slave. When the slave is initially selected, it must also monitor the status of HREADY to ensure that the previous bus transfer has completed, before it responds to the current transfer. The HSELx signal is a combinatorial decode of the address bus.

MUX signals

Name	Destination	Description
HRDATA [31:0]	Master	Read data bus, selected by the decoder.
HREADY	Master and Slave	When HIGH, the HREADY signal indicates to the master and all slaves, that the previous transfer is complete.
HRESP	Master	Transfer response, selected by the decoder

Verification plan

Sr. No.	Feature	Test Description	Reference	Type	Expected Outcomes	Results	Comments
1	**Basic operation**	Send data A e.g., 0x24 to **HADDR [31:0]** and set **HWRITE** to *low*	3.1	TR	**HREADY** should be *high.* In the next HCLK, **HWDATA [31:0]** generates data A e.g., 0x24		
2		Send data A e.g., 0x24 to **HADDR [31:0**] and set **HWRITE** to *low*			**HREADY** should be *high.* In the next **HCLK, HRDATA [31:0]** generates data A e.g., 0x24		
3	**Wait state**	Send data A e.g., 0x24 to **HADDR [31:0]** and set **HWRITE** to *low.* On next rising edge, set HREADY to *low* Send data B e.g., 0x28 to **HADDR [31:0]**. On next edge of **HCLK**, set **HREADY** to *high*	3.1	TR	In the next **HCLK, HRDATA [31:0]** generates data A e.g., 0x24		
4		Send data A e.g., 0x24 to **HADDR [31:0]** and set **HWRITE** to *high.* On next edge of **HCLK**, set **HREADY** to *low* Send data B e.g., 0x28 to **HADDR [31:0]**. On next edge of **HCLK**, set **HREADY** to *high*			In the **HCLK** when **HREADY** is *low*, **HWDATA [31:0]** generates data A e.g., 0x24		

5	**Multiple transfers**	Send data A e.g., 0x24, to **HADDR [31:0]** and set **HWRITE** to *high.* On next rising edge, send data B e.g., 0x28 to **HADDR [31:0]** and set **HWRITE** to *low. S*et **HREADY** to *high.* For next two rising edges, send data C e.g., 0x2C to **HADDR [31:0]**, set **HWRITE** to *high* for one **HCLK** and *low* for other **HCLK.**	3.1	TR	In the **HCLK** when **HREADY** is low, **HWDATA [31:0]** generates data A e.g., 0x24. On next edge of **HCLK**, wait state will occur. On next edge of **HCLK**, data B e.g., 0x28 will be generated at **HRDATA [31:0]**		
6	**HREADY**	If **HREADY** is *low,* no data should be generated on **HRDATA [31:0]** and **HWDATA [31:0]**	3.1	A	No read or write transfer occurs		
7	**IDLE transfer**	Bursts transfer is occurring so next transfer cannot take place immediately	3.2	A	Slave gives wait with **HRESP** is OKAY, while the transfer is ignored by slave.		
8	**BUSY case**	When master uses BUSY, the address and control signals must reflect the next transfer in bursts	3.2	A	During this, no data would be read on respective **HCLK** and **HRDATA [31:0]** would generate data on next **HCLK**		
9	**SEQ case**	When a SEQ transfer type is applied the control information is identical to previous transfer while the address must be equal to previous transfer plus transfer size	3.2	A	For example, in INCR, SEQ data is 0x20 is generated on **HADDR [31:0]**, then on next positive edge of **HCLK, HDDR [31:0]** should generate 0x24		
10	**Address boundary**	**HSIZE [2:0]** must be used in conjunction with **HBURST [2:0]** and should remain constant throughout the transfer, to determine	3.4	A	Respective **HSIZE [2:0]** should be with burst type and be constant		

		address boundary for wrapping bursts					
11	**Burst**	In a burst, the actual address must be previous + offset size given by **HSIZE**. Each beat would be on new HCLK	3.5	A	For example, INCR4		
12	**Wrapping bursts**	Address boundary must be product of number of beats and size of transfer.	3.5	A	For example, a 4-beat wrapping bursts of word (4-byte) and have 16-byte boundaries.		
13	**HBURST [2:0] cases**	An **HBURST [2:0]** is generated on **HADDR [31:0]** such that on first **HCLK**, the transfer type is NONSEQ, while keeping the **HWRITE** to *low*. On all next **HCLK**, transfer type is SEQ. In this way all respective types of bursts are tested	3.5.3	TR	Data would be generated on **HRDATA [31:0]** with **HCLK** on 2nd positive edge. In this way, both data phase and address phase will work in parallel		All burst cases should be tested
14	**Fixed length bursts**	Each fixed length bursts must terminate with a SEQ type	3.5.1	A	Fixed length bursts are terminated otherwise **HRESP** gives ERROR		
15	**IDLE transfer**	During waited **IDLE** transfer is changed to **NONSEQ,** the address must be constant until **HREADY** is *high*	3.6.1	A	If the address is change changed when **HREADY** is low, assertion would fail		
16		If master changes from **SEQ** to **IDLE** transfer, slave must throw error by HRESP is ERROR			If slave does not responds to this transfer, assertion would fail		

17	**Busy transfer with fixed length**	Master changes from **BUSY** to **SEQ** transfer for fixed length bursts	3.6.1	A	Master keeps the transfer **HTRANS** constant until **HREADY** is *high.* Otherwise, assertion would fail.		
18	**Busy transfer**	Master changes from **BUSY** to **IDLE** or **NONSEQ** transfer, and undefined length burst is **terminated** but in case of **BUSY** to **SEQ** transfer and undefined length burst is **continued**.	3.6.1	A	If master has changed the transfer type and burst has violated the rule than assertion would fail		
19	**ERROR response**	ERROR must retain for two **HCLK** cycles. Such that ERROR would check if **HREADY** is low then on other **HCLK** edge if **HREADY** is high **HRESP** would be OKAY	3.6.2	A	If there is ERROR for single **HCLK** then assertion would fail		
20	**HPROT**	A burst transfer of 4-beat is generated with data A, B, C and D to the rising edge of **HCLK** respectively while **HWRITE** is *low*. Now for this input, set **HPROT [3:0]** to b0111.	3.7	TR	The interface must respond to non-cacheable, buffer able, privileged and data access		Similar cases should be tested on all HPROT cases
21	**Default slave case**	If a NONSEQ or SEQ transfer address location is attempted to a non-existent, **HRESP** must give an ERROR while **HREAD**Y is *low* and if IDLE or BUSY transfers to non-existent locations, **HRESP** must give an ERROR while **HREADY** is low	4.1.1	A	If none of the condition is satisfied, assertion would fail		

22	**Slave response**	A slave must provide a response that indicates the status of the transfer	5.1	A	If slave does not respond, assertion would fail		
23	**Transfer done**	Send data A e.g., 0x24 to **HADDR [31:0]** and set **HWRITE** to low. On next rising edge of **HCLK**, set **HREADY** to high	5.1	TR	**HRESP** is OKAY. The transfer has either completed successfully or additional cycles are required for the slave		
24	**Transfer pending**	Send data A e.g., 0x24 to **HADDR [31:0]** and set **HWRITE** to low. On 2nd rising edge of **HCLK**, set **HREADY** to low.	5.1.1	TR	**HRESP** is OKAY. Additional cycles are required for the slave		
25	**ERROR case**	If the slave requires more than two cycles to provide the ERROR response, then additional wait states should be inserted at the start of the transfer.	5.1.3	A			
26	**Endianness**	All masters and slaves must be on same route on same functioning	6.1.1	A	If **HSELx** selects different route, assertion would fail		
27	**Correct byte lane**	For read write transfers, the receiving module must select the data from the correct byte lane on the bus i.e., **HSELx** must select right slave	6.2	A	If read data selects data from wrong byte lane i.e., wrong slave is selected, assertion would fail		
28	**HCLK**	Send data A e.g., 0x24 and data B e.g., 0x28 to **HADDR [31:0]**. Set **HWRITE** and **HREADY** to *high.* Then, set **HWRITE** to low	7.1.1	TR	All inputs are sampled at rising edge of **HCLK** as well as all output are sampled after rising edge of **HCLK**		
29	**HRESETn case 1**	**HRESETn** should be the only active Low signal	7.1.2	A	All active signals must be *high,* otherwise assertion would fail		

30	**HRESETn case 2**	During reset all slaves must ensure **HREADYOUT** is *high*	7.1.2	A	If **HREADYOUT** is *low,* assertion would fail		

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.
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.
Expected Outcomes	The outputs expected when test is executed.
Result	Pass (P) or Fail (F).
Comments	Any other comments about the test or its results that you want to mention.