Table of Contents

Authors ........................................................................................................................................ 2
Copyright & License .................................................................................................................. 2
Revision No. ............................................................................................................................... 2
Disclaimer .................................................................................................................................... 2
Introduction .................................................................................................................................. 3
Limitations of Design .................................................................................................................. 3
Synthesizability .......................................................................................................................... 3
Port Definitions ........................................................................................................................... 4
Black Box .................................................................................................................................... 5
Reset operation ........................................................................................................................... 6
Write Cycle .................................................................................................................................... 6
Read Cycle ...................................................................................................................................... 7
Write with Wait states ................................................................................................................... 8
  Inserted by AHB Slave .............................................................................................................. 8
  Inserted by Wishbone Master .................................................................................................. 9
Read with wait states .................................................................................................................... 10
  Inserted by AHB Slave ............................................................................................................ 10
  Inserted by Wishbone Master ............................................................................................... 11
Read after Write : ......................................................................................................................... 12
Write after Read ............................................................................................................................ 13
References ..................................................................................................................................... 15
Authors

TooMuch Semiconductor Solutions (info@toomuchsemi.com)

Copyright & License

Copyright (c) 2007 TooMuch Semiconductor Solutions Pvt Ltd.

Revision No.

1.0

Disclaimer

This source file may be used and distributed without restriction provided that this copyright statement is not removed from the file and that any derivative work contains the original copyright notice and the associated disclaimer.

This software is provided “as is” and without any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. In no event shall the author or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.
Introduction

This is a BRIDGE that is used to interconnect WISHBONE Master and AHB Slave

Limitations of Design

- This design will not work for Sequential transfers (i.e., for wrap and incremental type of transfers).
- No error, retry, and split cases have been considered in the design.
- Word size is 32 bits. It can be modified as per requirement.
- All transfers have been considered as non-sequential type.

Synthesizability

This code is completely synthesizable. The only care that has to be taken is that the AHB clock (hclk) and reset (hresetn) signals have been left unconnected inside the design. The whole design is working on Wishbone clock and reset signals that should be provided by Wishbone master.
<table>
<thead>
<tr>
<th>Port Name</th>
<th>Direction</th>
<th>Size</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>haddr</td>
<td>out</td>
<td>32</td>
<td>AHB Master</td>
<td>Address Bus</td>
</tr>
<tr>
<td>htrans</td>
<td>out</td>
<td>2</td>
<td>AHB Master</td>
<td>Transfer Type</td>
</tr>
<tr>
<td>hwrite</td>
<td>out</td>
<td>1</td>
<td>AHB Master</td>
<td>Read/Write enable</td>
</tr>
<tr>
<td>hsize</td>
<td>out</td>
<td>3</td>
<td>AHB Master</td>
<td>not used</td>
</tr>
<tr>
<td>hbust</td>
<td>out</td>
<td>3</td>
<td>AHB Master</td>
<td>Burst Type</td>
</tr>
<tr>
<td>hwdata</td>
<td>out</td>
<td>32</td>
<td>AHB Master</td>
<td>Write data bus</td>
</tr>
<tr>
<td>hrdata</td>
<td>out</td>
<td>32</td>
<td>AHB Master</td>
<td>Read data bus</td>
</tr>
<tr>
<td>hready</td>
<td>in</td>
<td>1</td>
<td>AHB Slave</td>
<td>To indicate bridge is ready</td>
</tr>
<tr>
<td>hresp</td>
<td>in</td>
<td>2</td>
<td>AHB Slave</td>
<td>Response</td>
</tr>
<tr>
<td>data_o</td>
<td>out</td>
<td>32</td>
<td>WISHBONE Slave</td>
<td>Read data bus</td>
</tr>
<tr>
<td>data_i</td>
<td>in</td>
<td>32</td>
<td>WISHBONE Master</td>
<td>Write data bus</td>
</tr>
<tr>
<td>addr_i</td>
<td>in</td>
<td>32</td>
<td>WISHBONE Master</td>
<td>Address Bus</td>
</tr>
<tr>
<td>clk_i</td>
<td>in</td>
<td>1</td>
<td>WISHBONE Master</td>
<td>Clock</td>
</tr>
<tr>
<td>rst_i</td>
<td>in</td>
<td>1</td>
<td>WISHBONE Master</td>
<td>Active Hign Sync. Reset</td>
</tr>
<tr>
<td>cyc_i</td>
<td>in</td>
<td>1</td>
<td>WISHBONE Master</td>
<td>To indicate valid bus cycle</td>
</tr>
<tr>
<td>stb_i</td>
<td>in</td>
<td>1</td>
<td>WISHBONE Master</td>
<td>To indicate valid data transfer cycle</td>
</tr>
<tr>
<td>sel_i</td>
<td>in</td>
<td>4</td>
<td>WISHBONE Decoder</td>
<td>Selection</td>
</tr>
<tr>
<td>we_i</td>
<td>in</td>
<td>1</td>
<td>WISHBONE Master</td>
<td>Read/Write enable</td>
</tr>
<tr>
<td>ack_o</td>
<td>out</td>
<td>1</td>
<td>WISHBONE Slave</td>
<td>To indicate current transfer status</td>
</tr>
<tr>
<td>hclk</td>
<td>in</td>
<td>1</td>
<td>AHB Master</td>
<td>Unconnected</td>
</tr>
<tr>
<td>hresetn</td>
<td>in</td>
<td>1</td>
<td>AHB Master</td>
<td>Unconnected</td>
</tr>
</tbody>
</table>
Black Box

WISHBONE
AHB
BRIDGE

clk_i  →  hclk
rst_i  →  hresetn
data_i →  hwrite
addr_i →  haddr

cyc_i  →  hwrite
stb_i  →  hwrite
we_i   →  hwrite

ack_o  ←  hready
data_o  ←  hrdma

htrans  ←  htrans
hsel    ←  hsel
hburst  ←  hburst
hresp   ←  hresp
Reset operation

The WISHBONE reset signal is active HIGH and is used to reset the system and the bus. This reset has been asserted and deasserted synchronously after the rising edge of clk_i.

![Fig. 1: Reset operation](image)

Write Cycle

When we_i is HIGH, this signal indicates a write transfer and the master will broadcast data on the write data bus(data_i) when cyc_i and stb_i signal is HIGH.

1) At 1st clock edge the address(A1) and data (D1) is sampled by the WISHBONE slave on addr_i and data_i bus respectively.

2) At 1st clock edge ack_o is also asserted.

3) This address sampled is transfered asynchronously to the AHB master on haddr.

4) At 2nd clock edge Data(D1) for this address is sampled synchronously by AHB master on to hwdata bus irrespective of hready status.
5) This data is written into the AHB slave synchronously on hwdata bus.

6) Htrans is asserted to “Non – Seq (10)” transfer type at clock edge when cyc_i and stb_i are HIGH.

Fig 2: Write Cycle

Read Cycle

When we_i is LOW, this signal indicates a read transfer and the master will receive data on the read data bus (data_o) provided the ack_o signal is HIGH.

1) At 1st clock edge the address(A1) is sampled by the WISHBONE slave on addr_i.

2) This address will be transferred asynchronously to AHB master on haddr provided hready is HIGH.

3) At 2nd clock edge Data(D1) for this address is sampled synchronously by AHB on hrdata.

4) Data (D1) is latched on hrdata bus asynchronously when hready is asserted.
5) Data (D1) is latched on data_o asynchronously at the same time.

6) At 2\(^{nd}\) clock edge ack_o is asserted.

7) Htrans is asserted to “Non – Seq (10)” transfer type at clock edge when cyc_i and stb_i are HIGH.

---

**Fig. 3**: Read Cycle

**Write with Wait states**

*Inserted by AHB Slave*

1) At 1\(^{st}\) clock edge the address(A1) and data (D1) is sampled by the WISHBONE slave on addr_i and data_i bus respectively.

2) At 1\(^{st}\) clock edge ack_o is also asserted.

3) This address sampled is transferred asynchronously to the AHB master on haddr.
4) At 2\textsuperscript{nd} clock edge Data(D1) for this address is sampled synchronously by AHB master on to hwdata bus irrespective of hready status.
5) This data is written into the AHB slave synchronously on hwdata bus.
6) At 2\textsuperscript{nd} clock edge if hready is LOW then asynchronously even the WISHBONE slave is also made inactive by deasserting ack_o, so no new address is sampled.
7) At 3\textsuperscript{rd} clock edge if hready is HIGH then address (A2) and data (D2) is sampled by the WISHBONE on addr_i and data_i bus.
8) At 3\textsuperscript{rd} clock edge ack_o is made HIGH asynchronously.

\textbf{Inserted by Wishbone Master}

1) 1) At 1\textsuperscript{st} clock edge the address(A1) and data (D1) is sampled by the WISHBONE slave on addr_i and data_i bus respectively.
2) At 1\textsuperscript{st} clock edge ack_o is also asserted.
3) This address sampled is transfered asynchronously to the AHB master on haddr.
4) At 2\textsuperscript{nd} clock edge if stb_i is LOW then asynchronously even the ack_o as well as hready is made Low, so no new address is sampled.
5) At 2\textsuperscript{nd} clock edge Data(D1) for this address is sampled synchronously by AHB master on to hwdata bus irrespective of hready status.
6) This data is written into the AHB slave synchronously on hwdata bus.
7) At 3\textsuperscript{rd} clock edge if stb_i is HIGH then address (A2) and data (D2) is sampled by the
WISHBONE on addr_i and data_i bus.

9) At 3rd clock edge ack_o is made HIGH asynchronously along with hready.

![Fig. 5: Write cycle with wait state inserted by WB Master](image)

**Read with wait states**

*Inserted by AHB Slave*

1) At 1st clock edge the address(A1) is sampled by the WISHBONE slave on addr_i.
2) This address will be transferred asynchronously to AHB master on haddr.
3) At 2nd clock edge Data(D1) for this address is sampled synchronously by AHB on hrdata provided hready is HIGH.
4) At 2nd clock edge if hready is LOW then asynchronously even the WISHBONE slave is also made inactive by deasserting ack_o, so no new address is sampled.
5) Data (D1) is latched on hrdata bus asynchronously when hready is asserted.
6) Data (D1) is latched on data_o asynchronously at the same time.
7) When data (D1) appears on data_o, ack_o is asserted same time.

**Fig. 6:** Read Cycle with wait state inserted by AHB slave

**Inserted by Wishbone Master**

1) At 1\textsuperscript{st} clock edge the address(A1) is sampled by the WISHBONE slave on addr_i.
2) This address will be transferred asynchronously to AHB master on haddr.
3) At 2\textsuperscript{nd} clock edge if stb_i is Low, asynchronously even the ack_o and hready are made Low.
4) At 3\textsuperscript{rd} Clock edge if stb_i is HIGH then Data(D1) for this address is sampled synchronously by AHB on hrdata provided hready is HIGH.
5) Data (D1) is latched on hrdata bus asynchronously when hready is asserted.
6) Data (D1) is latched on data_o asynchronously at the same time.
7) When data (D1) appears on data_o, ack_o is asserted same time.
**Read after Write:**

1) At 1\textsuperscript{st} clock edge the address(A1) and data(D1) are sampled by the WISHBONE Slave on addr\textsubscript{i} and data\textsubscript{i} along with we\textsubscript{i} (Active High) signal and other control signals.

2) This address is transferred asynchronously to the AHB Master on haddr.

3) At 2\textsuperscript{nd} clock edge address(A2) for next transfer is sampled by WISHBONE Slave on addr\textsubscript{i} along with we\textsubscript{i} signal (Active Low) and is latched asynchronously on haddr bus. Also at 2\textsuperscript{nd} clock edge data(D1) for address(A1) is sampled synchronously by AHB Master on hwdata.

4) hwrite signal is pulled LOW asynchronously at 2\textsuperscript{nd} clock edge.

5) At 3\textsuperscript{rd} clock edge data for address (A2) is latched synchronously on to hrdata bus.

6) Data(D2) for this address is sampled asynchronously by WISHBONE Slave on data\textsubscript{o}.
Write after Read:

1) At 1\textsuperscript{st} clock edge the address(A1) is sampled by the WISHBONE Slave on addr\_i along with we\_i (Active Low) signal and other control signals.
2) This address will be transferred asynchronously to AHB Master on haddr.
3) At 2\textsuperscript{nd} Clock edge data(D1) for this address is sampled synchronously by AHB Master on hrdata.
4) D1 is transferred asynchronously to WISHBONE Slave on data\_o and ack\_o is asserted as well.
5) At 3\textsuperscript{rd} Clock edge we\_i signal is pulled to HIGH logic(for write operation).
6) we\_i signal is transferred asynchronously on hwrite.
7) At 3\textsuperscript{rd} clock edge address(A2) and data(D2) is sampled by WISHBONE Slave on addr\_i and data\_i busses.
8) A2 is transferred asynchronously from addr\_i to haddr.
9) At 4\textsuperscript{th} Clock edge data(D2) for address(A2) is sampled synchronously by AHB Master on hwdata.
Fig. 9: Write after Read Cycle
References

AMBA

- http://www.arm.com
- AMBA Specification (Rev. 2.0)

Wishbone

- http://www.opencores.org/wishbone
- WISHBONE specification Rev. B.3