



ClustraBus

Protocol Specification

Project Info	
Project Manager	Marc Oberholzer
Author(s)	Marc Oberholzer
Reviewer(s)	Martin Heimlicher, Christoph Glattfelder
Version	1.0
Date	05.02.2009

Copyright reminder

Copyright © 2009 by Enclustra GmbH, Switzerland. All rights are reserved.

Unauthorized duplication of this document, in whole or in part, by any means is prohibited without the prior written permission of Enclustra GmbH, Switzerland.

Although Enclustra GmbH believes that the information included in this publication is correct as of the date of publication, Enclustra GmbH reserves the right to make changes at any time without notice.

All information in this document is strictly confidential and may only be published by Enclustra GmbH, Switzerland.

All referenced trademarks are the property of their respective owners.

Document History

Version	Date	Author	Comment
1.0	05.02.2009	Marc Oberholzer	Release 1.0
0.4	12.11.2008	Marc Oberholzer	Slaves may now use the CmdLock signal to recognize that they are in a 'locked' state
0.3	11.08.2008	Marc Oberholzer	Fixed single read waveforms (CmdWrite polarity)
0.2	23.07.2008	Marc Oberholzer	Added external uC connectivity
0.1	01.04.2008	Marc Oberholzer	Preliminary

Table of Contents

1	Overview	4
2	Interface	5
2.1	Generics	5
2.2	Signals	5
3	Transfers	7
3.1	Single Word Transfers	7
3.1.1	Single Word Write Transfers	7
3.1.2	Single Word Read Transfers	8
3.2	Burst Transfers	9
3.2.1	Burst Write Transfers	9
3.2.2	Burst Read Transfers.....	10
4	Limitations	12
4.1	System Configuration	12
4.2	Functional	12
5	Verification Environment	13
5.1	Simulation-Only Verification Procedures and Components	13
5.1.1	ClustraBus Verification Simulation Package (cb_verif_sim_pkg)	13
5.1.2	ClustraBus Verification Read Data FIFO (cb_verif_rd_fifo)	13
5.2	Synthesizable Verification Components	13
5.2.1	ClustraBus Verification Master (cb_verif_master)	13
5.2.2	ClustraBus Verification Slave (cb_verif_slave)	13

1 Overview

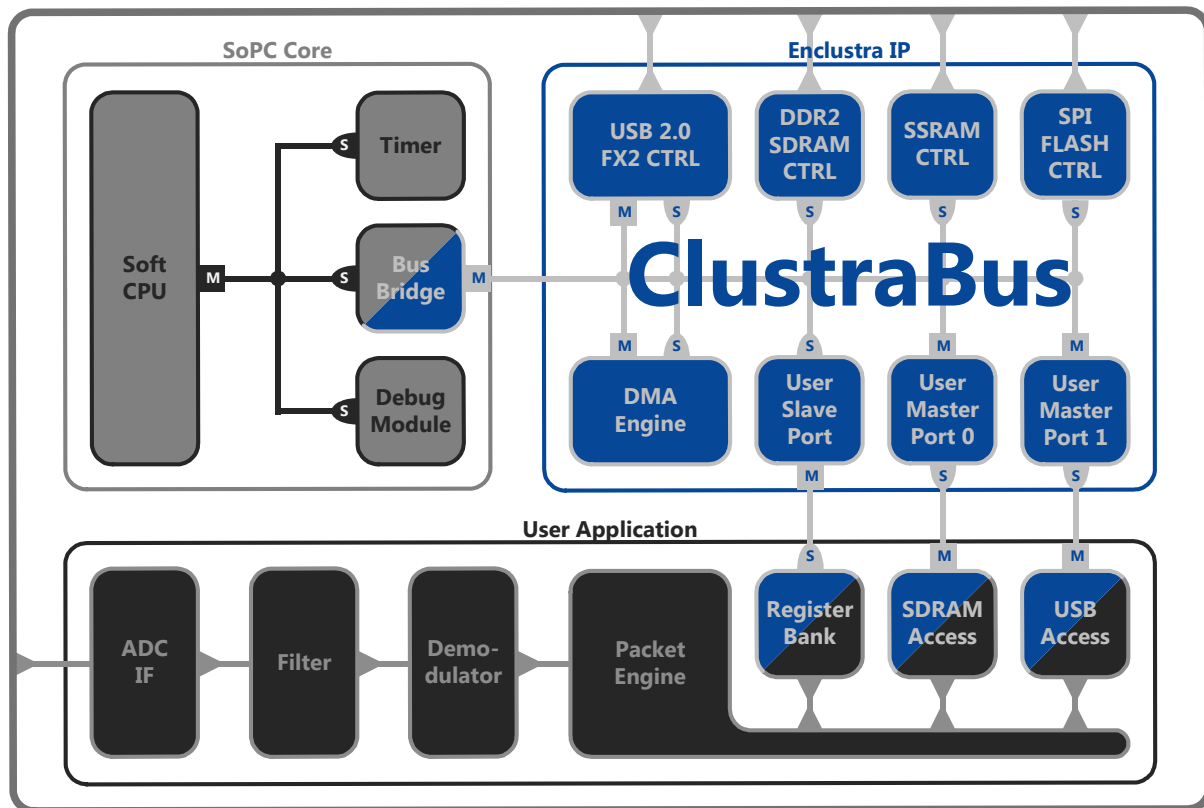


Figure 1: ClustraBus overview

ClustraBus is intended to be the standard interconnect for Enclustra's IP cores and VHDL projects, offering the following features and benefits:

- Common interface for Enclustra's IP cores
- Connectivity to the most popular SOPC bus systems (Avalon¹², PLB³⁴, Wishbone⁵)
- Connectivity to external microcontroller interfaces
- Multiple masters
- Multiple clock domains
- Multiple data bus widths
- Multiple open read transfers per slave
- Fully connected crossbar interconnect
- Single word and burst transfers
- Automated data width conversion
- Data prefetching for selected slaves (e.g. SDRAM controllers)

2 Interface

2.1 Generics

Generic Name	Type	Range	Description
AddrWidth_g	Positive	1...64	Specifies the width (in bits) of the CmdAddr interface signal.
DataWidth_g	Positive	1...144	Specifies the width (in bits) of the WrData and RdData interface signals.
ByteEnaWidth_g	Positive	1...16	Specifies the width (in bits) of the WrByteEna interface signal.

Table 1: Interface generics

Any ClustraBus slave or master must implement the generics listed in Table 1.

2.2 Signals

Signal Name	Width	Master	Slave	Description
Command Interface Signals				
CmdLock	1	Output	Input	Lock indicator. This signal may be used by a master to lock bus/slave access for multiple subsequent transfers.
CmdReq	1..16	Output	Input	Command request and slave select signal. A master uses this signal to request access to a particular slave and to indicate to the slave that a data transfer is requested. CmdWrite and CmdAddr must be valid whenever one bit of CmdReq is active. Only one bit of CmdReq may be high at a time.
CmdAck	1	Input	Output	Command acknowledge. This signal is used to indicate to the requesting master that its transfer request has been granted.
CmdWrite	1	Output	Input	Write/read indicator. This signal is used by a master to indicate whether a transfer is a read ('0') or a write ('1') transfer.
CmdAddr	1..64	Output	Input	Transfer start word address. Used to access a particular resource within the addressed slave's

				memory space.
Data Flow Interface Signals				
DataVld	1	Output	Input	Data Valid indicator. This signal is used by a master to indicate that WrData and DataLast are valid (write transfer) or that a read data word is requested (read transfer).
DataLast	1	Output	Input	Data last indicator. This signal is used by a master to mark the last write data word (write transfer) or the last requested read data word (read transfer).
DataRdy	1	Input	Output	Data ready indicator. This signal is used by a slave to indicate that it is ready to accept data (write transfer) or that it is ready to deliver data (read transfer).
Write Interface Signals				
WrData	1..144	Output	Input	Write data vector, qualified by DataVld.
WrByteEna	1...16	Output	Input	Write byte enable vector, qualified by DataVld.
Read Interface Signals				
RdData	1..144	Input	Output	Read data vector, qualified by RdVld.
RdVld	1	Input	Output	Read data valid indicator. This signal is used by a slave to indicate that it is presenting valid read data on RdData and that RdLast holds a valid value.
RdLast	1	Input	Output	Last read word indicator. This signal is used by a slave to indicate that the actually presented read data word is the last word of a transfer.

Table 2: Interface signals

Any ClustraBus slave or master must implement the signals listed in Table 2.

3 Transfers

3.1 Single Word Transfers

3.1.1 Single Word Write Transfers

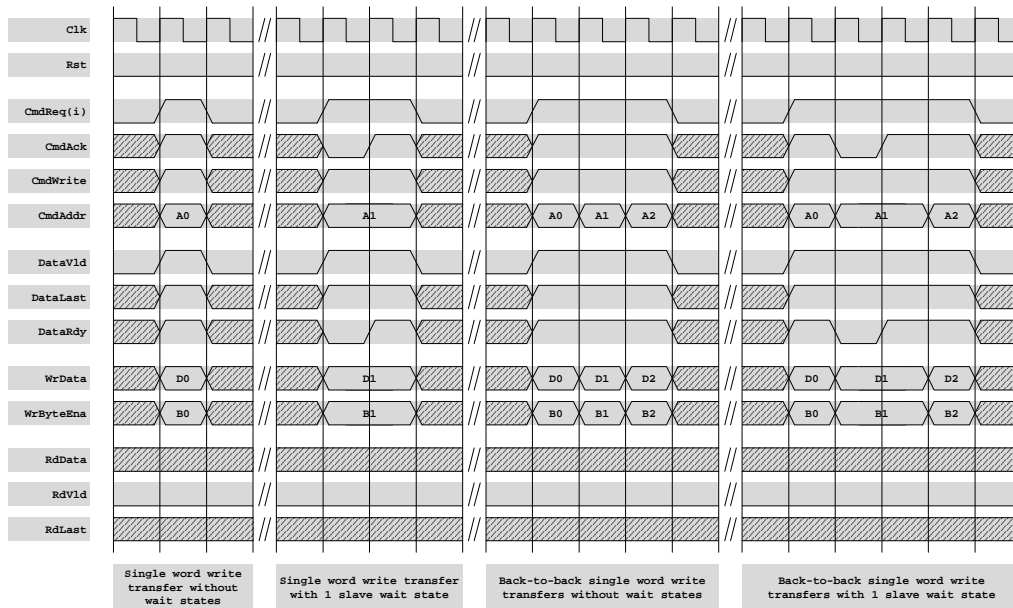


Figure 2: Single word write transfer waveforms

A single word write transfer is indicated by the master by presenting the following signal values to the slave:

CmdReq(i)	CmdWrite	DataVld	DataLast
'1'	'1'	'1'	'1'

Table 3: Single word write transfer master signal values

- A data transfer only takes place if CmdReq(i), CmdAck, DataVld and DataRdy are high at the same time.
- The slave is allowed to delay the data transfer by temporarily deasserting the CmdAck/DataRdy signals (slave wait states).
- The signals CmdAddr, DataLast, WrData and WrByteEna must always hold valid values during the time the signals CmdReq(i) and DataVld are high.
- The master must set/clear the CmdReq and DataVld signals simultaneously.
- The slave must set/clear the CmdAck and DataRdy signals simultaneously.

3.1.2 Single Word Read Transfers

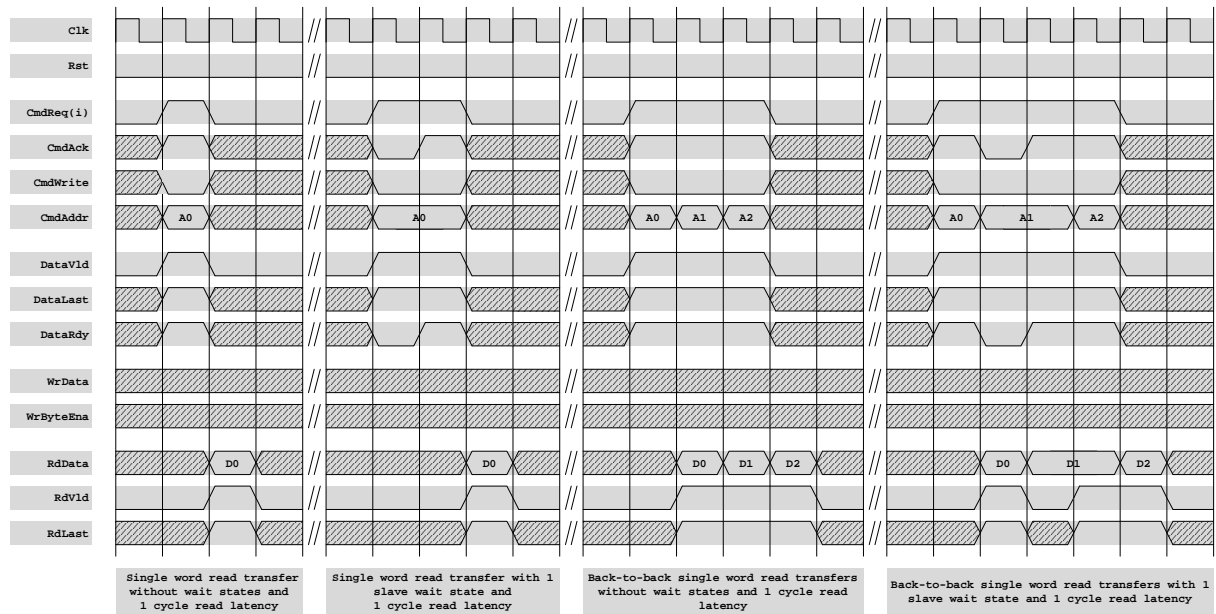


Figure 3: Single word read transfer waveforms

A single word read transfer is indicated by the master by presenting the following signal values to the slave:

CmdReq(i)	CmdWrite	DataVld	DataLast
'1'	'0'	'1'	'1'

Table 4: Single word read transfer master signal values

- The master must set and clear the CmdReq and DataVld signals simultaneously.
- A data transfer only takes place if CmdReq(i), CmdAck, DataVld and DataRdy are high at the same time.
- The slave is allowed to delay the data transfer by temporarily deasserting the CmdAck/DataRdy signals (slave wait states).
- The signals CmdAddr and DataLast must always hold valid values during the time the signals CmdReq(i) and DataVld are high.
- The master must set/clear the CmdReq and DataVld signals simultaneously.
- The slave must set/clear the CmdAck and DataRdy signals simultaneously.
- Read data latencies smaller than one cycle are not allowed.

3.2 Burst Transfers

3.2.1 Burst Write Transfers

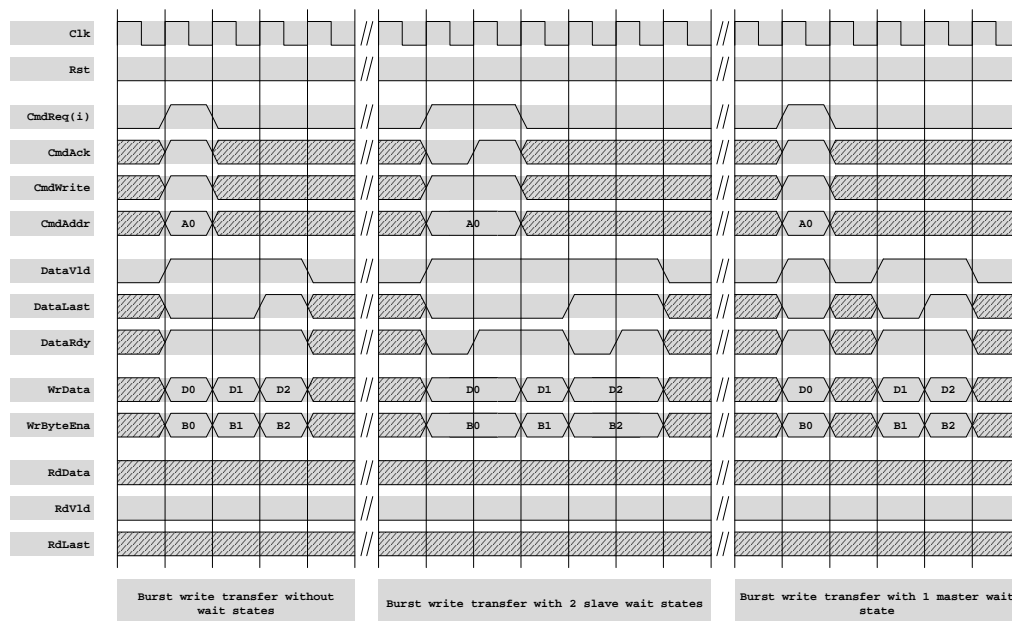


Figure 4: Burst write transfer waveforms

A burst write transfer start is indicated by the master by presenting the following signal values to the slave:

CmdReq(i)	CmdWrite	DataVld	DataLast
'1'	'1'	'1'	'0'

Table 5: Burst write transfer start master signal values

A burst write transfer intermediate data word is indicated by the master by presenting the following signal values to the slave (after a burst write transfer has been started):

CmdReq(i)	CmdWrite	DataVld	DataLast
'0'	Don't care	'1'	'0'

Table 6: Burst write transfer intermediate data master signal values

A burst write transfer last data word is indicated by the master by presenting the following signal values to the slave (after a burst write transfer has been started):

CmdReq(i)	CmdWrite	DataVld	DataLast
'0'	Don't care	'1'	'1'

Table 7: Burst write transfer last data master signal values

- A burst write start data transfer only takes place if CmdReq(i), CmdAck, DataVld and DataRdy are high at the same time.

- A burst write intermediate/last data transfer only takes place if DataVld and DataRdy are high at the same time.
- The slave is allowed to delay the burst write start data transfer by temporarily deasserting the CmdAck/DataRdy signals (slave wait states).
- The slave is allowed to delay burst write intermediate/last data transfers by temporarily deasserting the DataRdy signal (slave wait states).
- The master is allowed to delay burst write intermediate/last data transfers by temporarily deasserting the DataVld signal (master wait states).
- The signals CmdAddr, DataLast, WrData and WrByteEna must always hold valid values during the time the signals CmdReq(i) and DataVld are high (burst write start).
- The signals DataLast, WrData and WrByteEna must always hold valid values during the time the signal DataVld is high (burst write intermediate/last).
- The master must set the CmdReq and DataVld signals simultaneously (burst write transfer start).
- The slave must set the CmdAck and DataRdy signals simultaneously (burst write transfer start).

3.2.2 Burst Read Transfers

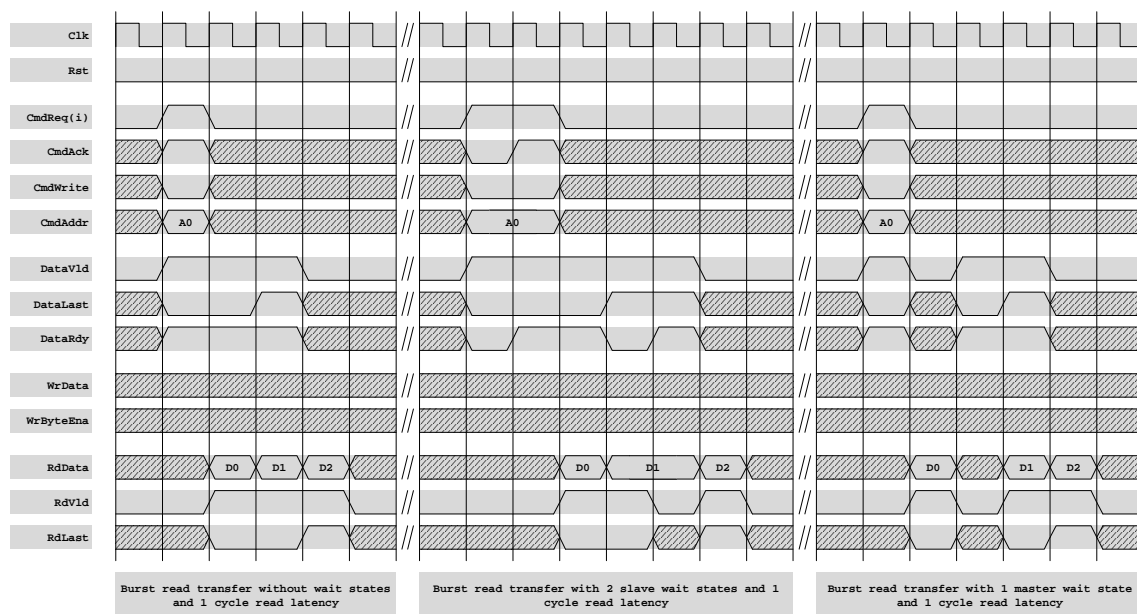


Figure 5: Burst read transfer waveforms

A burst read transfer start is indicated by the master by presenting the following signal values to the slave:

CmdReq(i)	CmdWrite	DataVld	DataLast
'1'	'0'	'1'	'0'

Table 8: Burst read transfer start master signal values

A burst read transfer intermediate data word is indicated by the master by presenting the following signal values to the slave (after a burst read transfer has been started):

CmdReq(i)	CmdWrite	DataVld	DataLast
'0'	Don't care	'1'	'0'

Table 9: Burst read transfer intermediate data master signal values

A burst read transfer last data word is indicated by the master by presenting the following signal values to the slave (after a burst read transfer has been started):

CmdReq(i)	CmdWrite	DataVld	DataLast
'0'	Don't care	'1'	'1'

Table 10: Burst read transfer last data master signal values

- A burst read start data request only takes place if CmdReq(i), CmdAck, DataVld and DataRdy are high at the same time.
- A burst read intermediate/last data request only takes place if DataVld and DataRdy are high at the same time.
- The slave is allowed to temporarily delay the burst read start data request by deasserting the CmdAck / DataRdy signals (slave wait states).
- The slave is allowed to temporarily delay burst read intermediate/last data requests by deasserting the DataRdy signal (slave wait states).
- The master is allowed to temporarily delay burst read intermediate/last data requests by deasserting the DataVld signal (master wait states).
- The signals CmdAddr and DataLast must always hold valid values during the time the signals CmdReq(i) and DataVld are high (burst read start).
- The signal DataLast must always hold a valid value during the time the signal DataVld is high (burst read intermediate/last).
- The RdData and RdLast signals are qualified by the RdVld signal. Therefore the RdData and RdLast signals must always hold valid values during the time the signal RdVld is high.
- Read data latencies smaller than one cycle are not allowed.
- There is no way for the master to delay the delivery of already requested read data (flow control has to be done by using master wait states while requesting read data).
- The master must set the CmdReq and DataVld signals simultaneously (burst read transfer start).
- The slave must set the CmdAck and DataRdy signals simultaneously (burst read transfer start).

4 Limitations

4.1 System Configuration

Parameter	Min	Max	Description
Master count	1	8	Total number of masters in a ClustraBus system
Slave count	1	16	Total number of slaves in a ClustraBus system
Address width [bit]	1	64	Total number of byte address bits (although ClustraBus uses word addresses, the maximum address bit width is specified as a byte address for convenience)
Data width [bit]	1	144	Total number of data bits (the maximum value equals to 16 bytes including a parity bit for each byte)
Byte enable width [bit]	1	16	Total number of byte enable bits

Table 11: System configuration limitations

4.2 Functional

Parameter	Min	Max	Description
Slave read latency	1	n.a.	Zero read latency (combinatorial) slaves are not supported.
Open read transfers	1	n.a.	The crossbar interconnect can be configured to support multiple open read transfers on an actually established master-slave connection. However, multiple open reads between one master and multiple slaves are not supported.

Table 12: Functional limitations

5 Verification Environment

5.1 Simulation-Only Verification Procedures and Components

5.1.1 ClustraBus Verification Simulation Package (cb_verif_sim_pkg)

This package provides basic ClustraBus protocol procedures for the use in Modelsim-based VHDL test benches.

5.1.2 ClustraBus Verification Read Data FIFO (cb_verif_rd_fifo)

This component must be instantiated in Modelsim-based VHDL test benches to enable the simulation of ClustraBus read transfers in conjunction with the ClustraBus protocol procedures.

5.2 Synthesizable Verification Components

5.2.1 ClustraBus Verification Master (cb_verif_master)

This synthesizable component consist of a memory bit error rate tester (BERT) and a randomized ClustraBus protocol wrapper. It is perfectly suited to verify ClustraBus memory interfaces but may also be used to verify other memory-mapped ClustraBus slaves.

5.2.2 ClustraBus Verification Slave (cb_verif_slave)

This synthesizable component consists of a storage memory and a randomized ClustraBus protocol wrapper. It is intended to act as a random slave in ClustraBus interconnect/bridge verification.

Figures

Figure 1: ClustraBus overview	4
Figure 2: Single word write transfer waveforms	7
Figure 3: Single word read transfer waveforms	8
Figure 4: Burst write transfer waveforms	9
Figure 5: Burst read transfer waveforms	10

Tables

Table 1: Interface generics.....	5
Table 2: Interface signals.....	6
Table 3: Single word write transfer master signal values	7
Table 4: Single word read transfer master signal values	8
Table 5: Burst write transfer start master signal values	9
Table 6: Burst write transfer intermediate data master signal values	9
Table 7: Burst write transfer last data master signal values	9
Table 8: Burst read transfer start master signal values	10
Table 9: Burst read transfer intermediate data master signal values	11
Table 10: Burst read transfer last data master signal values.....	11
Table 11: System configuration limitations	12
Table 12: Functional limitations.....	12

References

-
- ¹ *Avalon Interface Specifications, Version 1.00, Altera Corporation, March 2008*
 - ² *Avalon Memory-Mapped Interface Specification, Version 3.3, Altera Corporation, May 2007*
 - ³ *64 Bit Processor Local Bus Architecture Specification, Version 3.5, IBM Corporation, May 2001*
 - ⁴ *Processor Local Bus (PLB) v4.6 Product Specification DS531, v1.00a, Xilinx Inc., August 2007*
 - ⁵ *WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores, Revision B.3, September 2002*