AndeShape™ ATCTLC2AXI500 Data Sheet

Document Number DS182-10
Date Issued 2020-05-04

Copyright Notice

Contact Information

Should you have any problems with the information contained herein, you may contact Andes Technology Corporation through:

Email — support@andestech.com
Website — https://es.andestech.com/eservice/

Please include the following information in your inquiries:

- the document title
- the document number
- the page number(s) to which your comments apply
- a concise explanation of the problem.

General suggestions for improvements are welcome.

Revision History

Rev.	Rev. Date	Revised Content
1.0	2020-05-04	Initial release

Contents

List of Figures

List of Tables

1 Introduction

ATCTLC2AXI500 is a bus bridge, which connect a TileLink master to an AXI slave. It handles the bus protocol translation between TileLink and AMBA4 AXI protocols.

1.1 Features

- Compliant with the TileLink Cached (TL-C) and AMBA4 AXI protocols
- Configurable asynchronous and synchronous(N:1) clock domain between TileLink and AXI bus
- Configurable 24–64 bits address width
- Configurable 32/64/128/256 bits data width
- Configurable 4–8 ID width
- Configurable 8/16/32 outstanding requests
- Supports up to 128 bytes burst size
- Supports beat interleaving in TileLink channel D
- Supports AXI exclusive access
- Supports AXI cacheable characteristics information

1.2 Block Diagram

Figure 1 illustrates the top-level block diagram of the ATCTLC2AXI500 design. The upstream TileLink ports are slave interfaces which interact with the connected master interfaces. The downstream TileLink ports are master interfaces which interact with the connected slave interfaces.

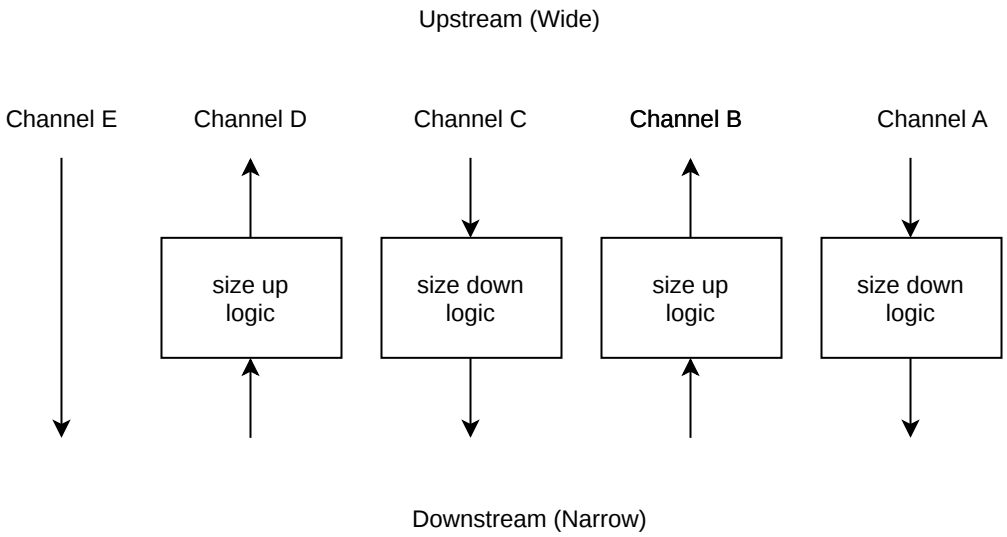

Figure 1: ATCTLC2AXI500 Block Diagram

1.3 Functional Description

1.3.1 Beat Interleave

ATCTLC2AXI500 allows interleaving beats of different messages on Channel D from the slave device through the downstream interface to the master device through the upstream interface. It is, however, forbidden to interleave beats on Channel A, Channel B, and Channel C. For supporting beat interleaving, the connected master device must support the following features when the connected slave device, such as TileLink-to-AXI bridge, responds beats with interleaving.

- Each ID of `a_source` with the request type message of `a_opcode`, such as `PutFullData`, `PutPartialData` and `AcquireBlock`, can only be issued once until the correspond response returned from Channel D.
- Each ID of `c_source` with the request type message of `c_opcode`, such as `Release` and `ReleaseData`, can only be issued once until the correspond response returned from Channel D.
- Each ID of `a_source` and `c_source` with the write request type message can only be issued once until the correspond response returned from Channel D.

2 Hardware Configuration Option

Table 1 lists the configuration parameters of ATCTLC2AXI500. These descriptions can be applied to both upstream side and downstream side unless otherwise stated.

Table 1: Configuration Parameters

Parameter Name	Legal Value	Default Value	Description
ACLK_DOMAIN	sync, async	sync	Define the axi clock domain, sync(N:1) or async
ADDR_WIDTH	32 - 128	32	Define the bit width of the address field in tilelink and axi channels
DATA_WIDTH	32, 64, 128, 256	64	Define the bit width of the data field in tilelink/axi channels
ID_WIDTH	4 - 8	4	Define the bit width of source/id field in tilelink/axi channels
MAX_OUTSTANDING	4 - 16	16	Define the maximum number of outstanding requests supported

3 Signal Description

Figure 2 shows the I/O signals of ATCTLC2AXI500, including TileLink signals for upstream and downstream interfaces respectively. Table 2 shows the detailed descriptions of both upstream and downstream interface signals.

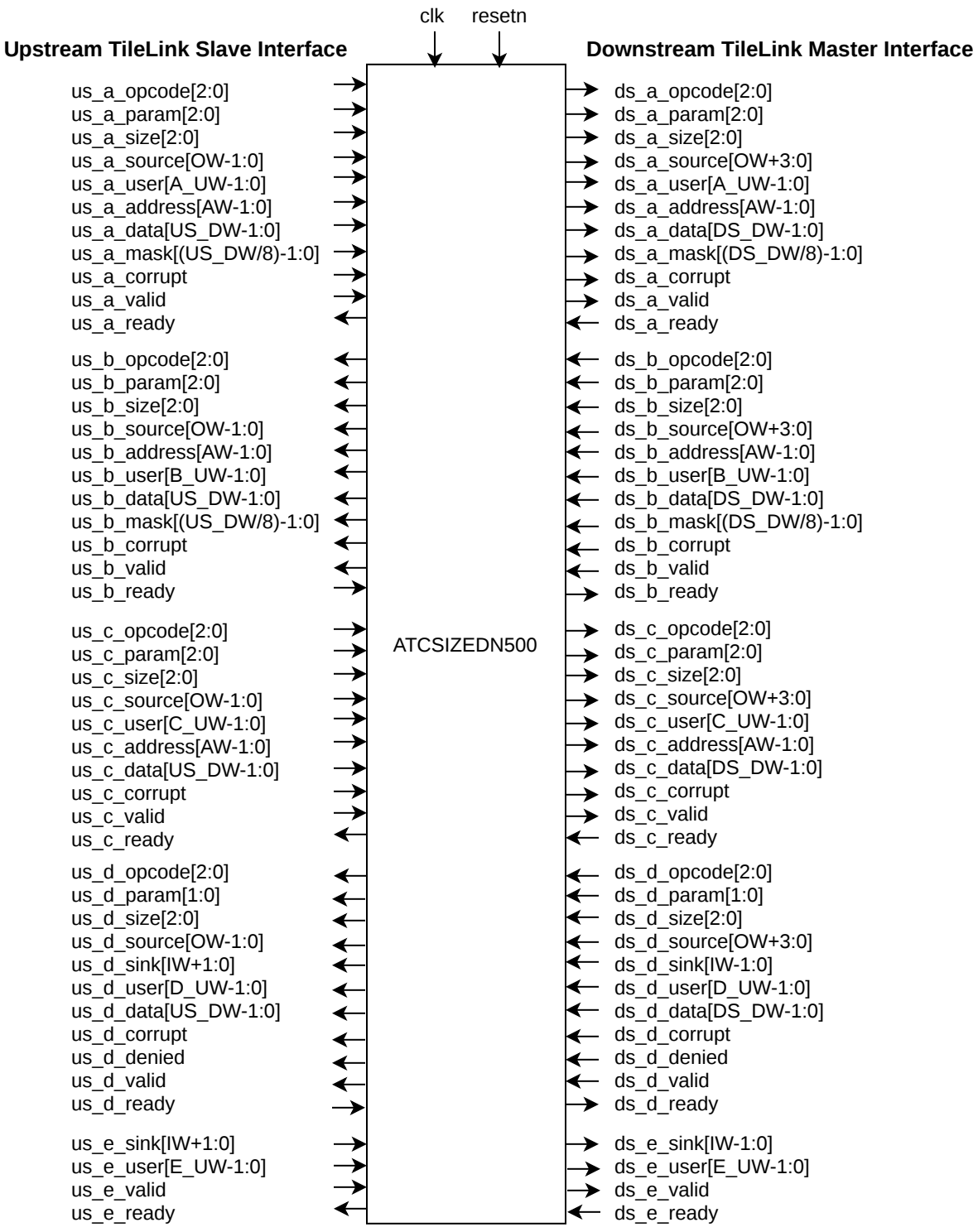

Figure 2: ATCTLC2AXI500 I/O Signals

Table 2: Interface Signal Descriptions

Signal Name	IO Type	Description
clk	I	Bus clock
resetn	I	Bus reset (Active-Low)
aclk	I	AXI clock (async)
aclk_en	I	AXI clock enable (sync)
TileLink Interface		
Channel A		
a_opcode[2:0]	I	Operation code
a_param[2:0]	I	Parameter code
a_size[2:0]	I	Operation size
a_source[ID_WIDTH-1:0]	I	Master source identifier
a_address[ADDR_WIDTH-1:0]	I	Target byte address
a_user[7:0]	I	Sideband for a*x*lock, a*x*cache, a*x*lock
a_data[DATA_WIDTH-1:0]	I	Data payload
a_mask[(DATA_WIDTH/8)-1:0]	I	Byte lane select signal
a_corrupt	I	The data in this beat is corrupt.
a_valid	I	Valid signal.
a_ready	O	Ready signal.
Channel B		
b_valid	O	Valid signal, hard-wired to 0.
Channel C		
c_opcode[2:0]	I	Operation code
c_param[2:0]	I	Parameter code
c_size[2:0]	I	Operation size
c_source[ID_WIDTH-1:0]	I	Master source identifier
c_user[7:0]	I	Sideband for a*x*lock, a*x*cache, a*x*lock
c_address[ADDR_WIDTH-1:0]	I	Target byte address
c_data[DATA_WIDTH-1:0]	I	Data payload
c_corrupt	I	The data in this beat is corrupt.
c_valid	I	Valid signal.
c_ready	O	Ready signal.
Channel D		
d_opcode[2:0]	O	Operation code
d_param[1:0]	O	Parameter code

Continued on next page...

Table 2: (continued)

Signal Name	IO Type	Description
d_size[2:0]	O	Operation size
d_source[ID_WIDTH-1:0]	O	Master source identifier
d_sink	O	Slave sink identifier
d_user[1:0]	O	Sideband to transfer rresp and bresp
d_data[DATA_WIDTH-1:0]	O	Data payload
d_denied	O	The slave was unable to service the request
d_corrupt	O	The data in this beat is corrupt.
d_valid	O	Valid signal.
d_ready	I	Ready signal.
Channel E		
e_ready	O	Ready signal, hard-wired to 1.
AXI Interface		
Write Address Channel		
awid[ID_WIDTH-1:0]	O	Write address ID
awaddr[ADDR_WIDTH-1:0]	O	Write address
awlen[7:0]	O	Write burst length
awsize[2:0]	O	Write burst size
awburst[1:0]	O	Write burst type
awlock	O	Write lock type
awcache[3:0]	O	Write memory type
awprot[2:0]	O	Write protection type
awvalid	O	Write address valid
awready	I	Write address ready
Write Data Channel		
wdata[DATA_WIDTH-1:0]	O	Write data
wstrb[(DATA_WIDTH/8)-1:0]	O	Write strobes
wlast	O	Write last
wvalid	O	Write data valid
wready	I	Write data ready
Write Response Channel		
bid[ID_WIDTH-1:0]	I	Write response ID
bresp[1:0]	I	Write response

Continued on next page...

Table 2: (continued)

Signal Name	IO Type	Description
bvalid	I	Write response valid
bready	O	Write response ready
Read Address Channel		
arid[ID_WIDTH-1:0]	O	Read address ID
araddr[ADDR_WIDTH-1:0]	O	Read address
arlen[7:0]	O	Read burst length
arsize[2:0]	O	Read burst size
arburst[1:0]	O	Read burst type
arlock	O	Read lock type
arcache[3:0]	O	Read memory type
arprot[2:0]	O	Read protection type
arvalid	O	Read address valid
arready	I	Read address ready
Read Data Channel		
rid[ID_WIDTH-1:0]	I	Read response ID
rdata[DATA_WIDTH-1:0]	I	Read data
rresp[1:0]	I	Read response
rlast	I	Read last
rvalid	I	Read data valid
rready	O	Read data ready

3.1 Side-band Signals

There are two side-band signals: a_user, c_user and d_user for encapsulating AXI signals that cannot be converted to TileLink protocol.

- a_user and c_user is used to propagate the **AxLOCK**, **AxCACHE**, and **AxPROT** from AXI AR and AW channel.
- d_user is used to indicate the **RRESP**, **BRESP**, and to the AXI R and B channel.

4 Access Latencies

Table 3 summarizes the channel latencies of ATCTLC2AXI500, which represent the beat transfer delay from source side to destination side.

Table 3: Latency between Both Sides for Each Channel

Source Side	Destination Side	Latency
TileLink A channel	AXI AW/AR/W channel	1
TileLink B channel	-	-
TileLink C channel	AXI AW/AR/W channel	1
AXI R/B channel	TileLink D channel	1
TileLink E channel	-	0