# AXI4 Protocol over AIB Latency Reference Guide Revision 1.0

#### **24 FEBRUARY 2023**

## Contents

1.	Background	2
2.	Simulation Parameters	2
3.	AXI4-Stream Latency and FIFO Buffer	3
4.	AXI4 Latency and FIFO Buffer	4
5.	How to Specify RX FIFO Buffer Size	4
5.1.	AXI4-Stream	4
5.2.	AXI4-MM	5
6.	Latency Optimization	7
7.	Addendum: Example waveform capture for Parameters A and B (Config 2 for AXI4-ST)	8
7.1.	Latency A	8
7 2	Latency B	9

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice.

Intel, the Intel logo and Stratix are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Other names and brands may be claimed as the property of others.

© Intel Corporation.

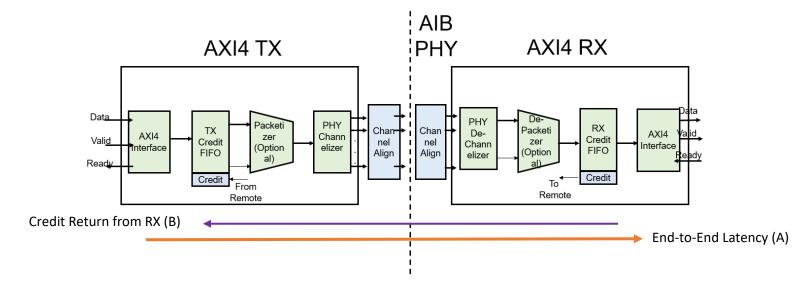
#### 1. Background

There are 2 critical latency paths in AXI4 protocol:

- A) End-to-End latency from TX to RX
- B) Credit return latency from RX to TX

For AXI4 without flow control (AXI4-Stream without ready) B is not relevant for other AXI4 protocols with flow control both A and B play an important role as discussed below.

AXI4 protocol over AIB uses credit mechanism to manage flow control between follower and leader chiplet. This credit mechanism also allows to manage the multiple clock cycle handshake between chiplets.



To avoid stalling in the traffic (bubbles) the recommendation is to size the RX credit FIFO to match A+B+k where k is an additional buffer. RX FIFO size is parameter for IP generation.

#### 2. Simulation Parameters

Below is a table of measured latency in simulation and RX FIFO buffer size recommendation for AXI4-Stream and AXI4 based on the following parameters.

AXI4 Mode	Freq (Ghz)
Half (Gen1)	0.5
Full (Gen1)	1
Quarter (Gen2)	0.5
Half (Gen2)	1
Full (Gen2)	2

# 3. AXI4-Stream Latency and FIFO Buffer

Confi	Leader AIB	Followe r AIB	FIFO Mode (TX→RX)	A (normalize	B (normalize	k	Freq Factor	RX FIFO Size =
No	Mode	Mode		d to AIB	d to AIB		betwee	Ceiling
110	IVIOGE	Mode		Fwd clk)	Fwd clk)		n AXI	((A+B+k)/F
				l			and AIB	)
							(F)	,
1	AIB2.0	AIB 1.0	Half→Half	44	40	2	2	43
	(Gen1)							
2	AIB2.0	AIB1.0	Full→Half	40	26	2	1	68
	(Gen1)							
3	AIB2.0	AIB1.0	Half→Full	30	40	2	1	72
	(Gen1)							
4	AIB2.0	AIB2.0	Half <del> →</del> Half	50	38	2	2	45
	(Gen2)	(Gen2)						
5	AIB2.0	AIB2.0	Full→Full	31	28	2	1	61
	(Gen2)	(Gen2)						
6	AIB2.0	AIB2.0	Quarter > Quarter	88	66	2	4	39
	(Gen2)	(Gen2)	_					
7	AIB2.0	AIB2.0	Full→Half	48	29	2	1	79
	(Gen2)	(Gen2)	_					
8	AIB2.0	AIB2.0	Full → Quarter	84	31	2	1	117
	(Gen2)	(Gen2)						
9	AIB2.0	AIB2.0	Half <del>→</del> Full	34	38	2	1	74
	(Gen2)	(Gen2)						
10	AIB2.0	AIB2.0	Half <del>→</del> Quarter	84	40	2	1	126
	(Gen2)	(Gen2)	_					
11	AIB2.0	AIB2.0	Quarter→Full	39	65	2	1	106
	(Gen2)	(Gen2)				_		
12	AIB2.0	AIB2.0	Quarter→Half	58	66	2	1	126
	(Gen2)	(Gen2)						

## 4. AXI4 Latency and FIFO Buffer

Si	Leader	Follower	FIFO Mode	Α	В	k	Freq	RX FIFO
No	AIB	AIB	(TX→RX)	(normalized	(normalized		Factor	Size =
	Mode	Mode		to AIB Fwd	to AIB Fwd		between	Ceiling
				clk)	clk)		AXI and	((A+B+k)/F)
							AIB (F)	
1	AIB2.0	AIB 1.0	Half→Half	50	32	2	2	42
	(Gen1)							
2	AIB2.0	AIB2.0	Half → Half	50	40	2	2	46
	(Gen2)	(Gen2)						
3	AIB2.0	AIB2.0	Full→Full	31	24	2	1	57
	(Gen2)	(Gen2)						
4	AIB2.0	AIB2.0	Quarter → Quarter	88	68	2	4	40
	(Gen2)	(Gen2)						

## 5. How to Specify RX FIFO Buffer Size

Can be specified as a configuration parameter in IP configuration file. Here are examples for AXI4-Stream and AXI4-MM

#### 5.1. AXI4-Stream

```
MODULE axi_st
// PHY and AIB Configuration
NUM_CHAN
                                           Gen1Only //Gen1Only, Gen2Only, Gen2, AIBO
CHAN_TYPE
// Channel Alignment Strobe Configuration
TX_ENABLE_STROBE
RX_ENABLE_STROBE
                                           True
                                                     // If False, all strobe functionality is removed.
                                                     // If False, all strobe functionality is removed.
                                           True
TX_PERSISTENT_STROBE
                                           True
                                                     // If True strobes are persistent (always there). If false, they are recoverable and can be
reused for data
RX_PERSISTENT_STROBE
                                                     // If True strobes are persistent (always there). If false, they are recoverable and can be
reused for data
TX_USER_STROBE
                                True
                                           // If True, then we input user generated signal
RX USER STROBE
                                True
                                           // If True, then we input user generated signal
TX_STROBE_GEN1_LOC
                                                     // Location of Strobe when in Gen1 Mode
RX_STROBE_GEN1_LOC
                                                     // Location of Strobe when in Gen1 Mode
// Word Marker Configuration
TX_ENABLE_MARKER
                                           True
                                                     // If False, all Marker functionality is removed.
RX_ENABLE_MARKER
                                           True
                                                     // If False, all Marker functionality is removed.
TX PERSISTENT MARKER
                                           True
                                                     // If True Markers are persistent (always there). If false, they are recoverable and can be
reused for data
RX_PERSISTENT_MARKER
                                           True
                                                     // If True Markers are persistent (always there). If false, they are recoverable and can be
reused for data//
TX_USER_MARKER
                                           True
RX_USER_MARKER
                                           True
TX_MARKER_GEN1_LOC
                                           39
                                                      // Location of Marker when in Gen1 Mode
RX_MARKER_GEN1_LOC
                                                     // Location of Marker when in Gen1 Mode
```

```
TX_REG_PHY
                    False
RX_REG_PHY
                    False
SUPPORT_ASYMMETRIC
                                         True
                                                   // Support Asymmetric Gearboxing (e.g. Full to Half)
llink ST
 TX_FIFO_DEPTH
 RX_FIFO_DEPTH
                     68 # for Full→Half
                  256
 output user_tdata
 input user_tready
                  ready
 output user_tvalid valid
```

#### 5.2. AXI4-MM

```
MODULE axi_mm
// PHY and AIB Configuration
NUM_CHAN
                                         2
CHAN_TYPE
                                         Gen1Only
                                                              //Gen1Only, Gen2Only, Gen2, AIBO
TX_RATE
                                         Half
                                                              // Full, Half, Quarter
RX_RATE
                                                              // Full, Half, Quarter
                                         Half
// Channel Alignment Strobe Configuration
TX_ENABLE_STROBE
                               True
                                         // If False, all strobe functionality is removed.
RX ENABLE STROBE
                               True
                                         // If False, all strobe functionality is removed.
TX PERSISTENT STROBE
                                                    // If True strobes are persistent (always there). If false, they are recoverable and can
be reused for data
RX_PERSISTENT_STROBE
                                                    // If True strobes are persistent (always there). If false, they are recoverable and can
                                         True
be reused for data
TX_USER_STROBE
                                         False
                                                    // If True, then we input user generated signal
RX USER STROBE
                                         False
                                                    // If True, then we input user generated signal
TX STROBE GEN1 LOC
                                         7
                                                    // Location of Strobe when in Gen1 Mode
RX_STROBE_GEN1_LOC
                                                    // Location of Strobe when in Gen1 Mode
// Word Marker Configuration
TX_ENABLE_MARKER
                               True
                                         // If False, all Marker functionality is removed.
RX_ENABLE_MARKER
                               True
                                         // If False, all Marker functionality is removed.
TX_PERSISTENT_MARKER
                                                    // If True Markers are persistent (always there). If false, they are recoverable and can
                                         True
be reused for data
RX_PERSISTENT_MARKER
                                         True
                                                    // If True Markers are persistent (always there). If false, they are recoverable and can
be reused for data//
TX USER MARKER
                                         False
RX_USER_MARKER
                                         False
TX_MARKER_GEN1_LOC
                                         39
                                                    // Location of Marker when in Gen1 Mode
RX_MARKER_GEN1_LOC
                                                    // Location of Marker when in Gen1 Mode
                                         39
TX_REG_PHY
                   False
RX_REG_PHY
                   False
// Packetization
TX_ENABLE_PACKETIZATION
                                                    True
RX_ENABLE_PACKETIZATION
                                                    True
TX_PACKET_MAX_SIZE
                                                    0
                                                              // Number of bits to packetize to. 0 means all available data.
RX_PACKET_MAX_SIZE
                                                              // Number of bits to packetize to. 0 means all available data.
PACKETIZATION_PACKING_EN
                                         False
                                                    // If True, enable packing which makes better use of the BW
llink AR
 TX_FIFO_DEPTH
```

```
RX_FIFO_DEPTH
 output user_arid 4
 output user_arsize 3
 output user_arlen 8
 output user_arburst 2
 output user_araddr 32
output user_arvalid valid
input user_arready ready
llink AW
TX_FIFO_DEPTH
 RX_FIFO_DEPTH
 output user_awid 4
 output user_awsize 3
 output user_awlen 8
 output user_awburst 2
output user_awaddr 32
output user_awvalid valid
input user_awready ready
llink W
TX_FIFO_DEPTH
RX_FIFO_DEPTH
 output user_wid 4
 output user_wdata 128
 output user_wstrb 16
 output user_wlast
output user_wvalid valid
input user_wready ready
llink AW
TX_FIFO_DEPTH
 RX_FIFO_DEPTH
 output user_awid 4
 output user_awsize 3
 output user_awlen 8
 output user_awburst 2
output user_awaddr 32
output user_awvalid valid
input user_awready ready
llink W
TX_FIFO_DEPTH
RX_FIFO_DEPTH
 output user_wid 4
 output user_wdata 128
 output user_wstrb 16
output user_wlast
 output user wvalid valid
 input user_wready ready
```

## 6. Latency Optimization

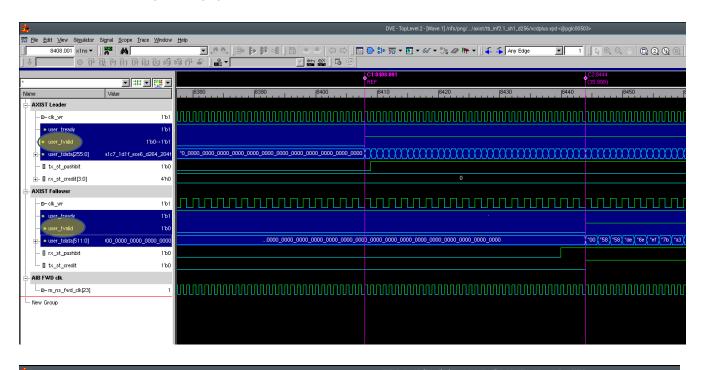
These latencies reported here are out of the box. Here are some tips to improve latency with system knowledge:

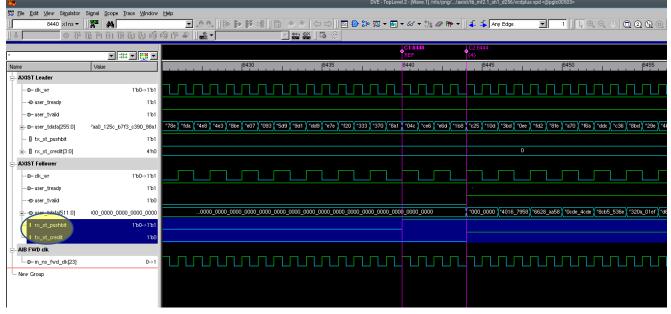
- a) Lane to lane to skew modeled in testbench as delay between AIB and CA. This delay is worst cased at 5 clock cycle this can be optimized for your interface and system once you have the system parameters extracted.
- b) Improve RX/TX phasecomp RD delay with the knowledge of AIB write vs read clock wander margin. These values can be programmed to AIB CSR 0x208/0x218.

# 7. Addendum: Example waveform capture for Parameters A and B (Config 2 for AXI4-ST)

#### 7.1. Latency A

Sum of valid to valid + push to pop on RX FIFO Buffer





# 7.2. Latency B

Credit return from follower to leader

