

**FDAS DDRIF2 Module Design Specification for Intel Agilex F Implementation**

FDAS\_DDRIF2\_DS Revision 3 Draft C

Classification: UNRESTRICTED

Document type: DTE

Date: 2023-01-25

Status:

© Copyright 2023 SKA Observatory.

  This work is licensed under a [Creative Commons Attribution 4.0 International License](http://creativecommons.org/licenses/by/4.0/)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Role* | *Name* | *Designation* | *Affiliation* | *Signature* | *Date* |
| Author | Martin Droog | FPGA Design Engineer | SKAO |  | 25/01/2023 |
| Owner | Ben Stappers | Head of SKA PSS Team | SKAO |  |  |
| Approver | Ben Stappers | Head of SKA PSS Team | SKAO |  |  |
| Released by | Lina Levin Preston | SKA PSS Team Scrum Master | SKAO |  |  |

|  |  |  |
| --- | --- | --- |
| Document History | | |
| Issue | Date | Comments |
| Issue 2 Draft A | 05/04/2022 | First Issue for review.  Comments from Prabu Thiagaraj and Atul Ghalame, and responses from Martin Droog:-  1)If I am reading it right, the DDR write and read speeds are 300 ms and 130 ms, respectively. I am missing to recall why the read/write speeds are different?  **M. Droog Response: These times were based on initial theoretical analysis and are stating the time that each module/function within FDAS will need to have access to the DDR SDRAMs to complete the task. The numbers are not meant to infer that read and write accesses to DDR SDRAM take a different time to perform. More accurate analysis (with a single DDR SDRAM to store the FOP) shows the CONV module needs to have access to the DDR SDRAM for about 106ms to complete the FOP for a DM and the HSUM module needs to have access to the DDR SDRAM for about 335ms to read all the necessary FOP columns to perform the Harmonic summing of a DM. The nature of the HSUM module operation is that as it performs the harmonic summing across a range of possible pulsar frequencies and accelerations it may read a FOP column on multiple occasions thus accounting for the different CONV write access and HSUM read access times to the DDR SDRAM.**  **To make the DDRIF2 module specification more general the exact times shall be re-worded as a “period of time”.**  2) The read/write priority assigned is the lowest for the CONV. Although a clash may not happen, I am missing to recall why we have kept the lowest priority for CONV.    **M. Droog Response: The architecture is designed such that under normal operation there will be no clash between the CONV and HSUM modules trying to access the FOP DDR SDRAM at the same time. The PC/Computer knows what is happening because it triggers the activity in the FDAS FPGA and knows when activities have finished via the “DONE” signalling. Hence the PC/Computer access to the FOP DDR SDRAM via the PCIe has been given the highest priority just in case for debug purposes it wishes to grab access to the FOP DDR SDRAM midway through the processing of a DM.** |
| Issue 2 Draft B | 28/04/2022 | * The CONV and HSUM processing times have been re-worded as a “period of time” to make the DDRIF2 specification more general. |
| Issue 3 Draft A | 02/09/2022 | 1. Update the design to support up to three DDR Interfaces to be combined to act in unison to provide up to 1536-bit data buses to the CONV and HSUM modules to increase the bandwidth to the external memory storing the Filter Output Plane (FOP). 2. Remove the redundant feature whereby address bits [33:32] are used to select the DDRIF2 instance for PCIe access, since this feature is performed by the PCIe Hard IP Macro. 3. As a tidying exercise change the address busses to the PCIe Hard IP Macro to [31:0] as the upper [63:32] address bits are unused. 4. As a tidying exercise set the Byte Enable bus to the DDR Controller/Calibration block to 64 to cover only the data bytes that DDRIF2 controls. |
| Issue 3 Draft B | 02/09/2022 | Updated DDRIF2 instances to match the DDR Channel number. |
| Issue 3 Draft C | 25/01/2023 | * The Intel Agilex PCIe Hard Macro has a “wait request allowance” hard coded to 16 for the DMA Writes from the Host Computer to the FPGA. Hence the RX\_FIFO size has to be increased to take the data after DDRIF2 asserts Wait\_Request. The RX\_FIFO size has been increased from 16 to 64 locations. In addition the FIFO must have at least 48 spare locations to avoid RD\_DMA\_WAIT\_REQUEST being asserted for WRITE accesses from the PCIe Hard IP Macro, inferring the DDRIF2 module has a WaitRequest Allowance of 48 or WRITE accesses. |

**Table of Contents**

1 Introduction 5

2 Place in the System 6

3 Functional Specification 7

3.1 DDRIF2 Connectivity 7

3.1.1 DDRIF2 <> DDR Controller 9

3.1.2 PCIe Hard IP Macro <> DDRIF2 Interface 9

3.1.3 DDRIF2 > CLD and DDRIF2 > HSUM Interface 12

3.1.4 CONV > DDRIF2 Interface 13

3.2 DDRIF2 Architecture 14

3.2.1 Interface Select: IF\_SEL Sub-Module 15

3.2.2 Receive PCIE Interface: RX\_PCIE\_IF Sub-Module 15

3.2.3 Receive Processing Interface: RX\_PROC\_IF Sub-Module 16

3.2.4 RX\_FIFO , WAG\_IN & RAG\_IN Sub-Modules 17

3.2.5 Transmit PCIE Interface: TX\_PCIE\_IF Sub-Module 18

3.2.6 Transmit Processing Interface: TX\_PROC\_IF Sub-Module 19

3.2.7 DATA\_FIFO\_512, WAG\_OUT , RAG\_OUT\_512 and RAG\_OUT\_3\_512 Sub-Modules 20

4 Design Requirement Tags 21

5 Interface Specification 22

6 MCI Memory Mapped Interface 26

7 Design Parameterisation 27

8 References 28

9 Abbreviations and Acronyms 29

# Introduction

This document captures the requirements for the DDRIF2 module used in the FDAS FPGA implemented in the Intel Agilex F family.

For the first implementation of FDAS in Agilex two instantiations of the DDRIF2 module were required in FDAS. However for the second implementation of FDAS in Agilex four instantiations of the DDRIF2 module shall be required, with three of the instantiations combined to act in unison to provide the CONV and HSUM modules with a 1536-bit data bus to the external DDR4 SDRAM storing the FOP.

The DDRIF2 connection to the PCIE Hard IP Macro module shall support DMA transfers (both read and write) via 512-bit data buses @ 350MHz.

The DDRIF2 connection to the CLD module shall be via 512-bit data buses at 350MHz.

The DDRIF2 connection to the CONV and HSUM modules shall be via 512-bit data buses at 350MHz, with three DDRIF2 instances acting in unison to provide the CONV and HSUM modules with an effective 1536-bit data bus.

It is the responsibility of functions external to the DDRIF2 module to ensure that only one interface (from PCIe, CLD, CONV or HSUM) is requesting access to the DDRIF2 module at any one time. Hence the DDRIF2 module does not need any form of scheduling function to apportion bandwidth.

The DDRIF2 module shall use data FIFOs to adapt between the internal clock rates and the clock rate for the Intel Agilex DDR4 Controller. The internal clock rates shall be 350MHz for the PCIe Clock (CLK\_PCIE) and 350MHz for the System Clock (CLK\_SYS). The clock rate for the Intel Agilex DDR4 Controller (CLK\_DDR) shall be 1333.333MHz/4 = 333.333MHz.

**This version of DDRIF2 supports a WaitRequest Allowance of 48 on the WRITE accesses from the PCIe Hard IP Macro and hence is not compatible with the standard (classic) Avalon interface protocol. Only the RD\_DMA\_WRITE (= 1) signal is used to qualify existence of a new WRITE access. The WaitRequest Allowance feature allows for greater bandwidth as the data sender (i.e. Host) can continue transmitting for a specified time after the slave (i.e. Agent) as asserted its WaitRequest.**

**This version of DDRIF2 supports a WaitRequest Allowance of 0 on the READ accesses by PCIe Hard IP Macro, i.e. it follows the standard (classic) Avalon interface protocol. The WR\_DMA\_READ (= 1) and WR\_DMA\_WAIT\_REQUEST (= 0) signals are used to qualify existence of a new READ access.**

# Place in the System

The DDRIF2 Module’s place in the system is shown highlighted in the figure below:-

**DDRIF2 #2**

**DDRIF2 #2**

PCIe

**CLD**

**CTRL**

FDAS Control

CLD\_DONE

**CONV**

data[63:0]

ready

**HSUM**

**FDAS**

CONV\_DONE CONV\_FFT\_REDAY

valid

**PCIE HARD IP MACRO (INTEL IP)**

**MCI\_TOP**

CLD\_DM\_TRIGGER

CLD\_ENABLE

CLD\_PAGE[31:0]

CONV\_DM\_TRIGGER

CONV\_ENABLE

CONV\_PAGE[31:0]

IFFT\_LOOP\_NUM[5:0]

HSUM\_DM\_TRIGGER

HSUM\_ENABLE

HSUM\_PAGE[31:0]

HSUM\_DONE

**PCIF**

ADDR

DATA[511:0]

**DDRIF2 #1**

**DDRIF2 #0, #2 and #3 acting in unison to provide a 1536-bit data interface to CONV and HSUM**

ADDR

3xDATA[5111:0]

3xDATA[511:0]

ADDR

ADDR

DATA[511:0]

*Note: The PCIe Hard Macro can read and write to allExternal DDR memories for diagnostic purposes. Not shown in this figure for clarity.*

*Note CLD, CONV and HSUM are designed with paging of the DDR memory for a future implementation. This allows different regions of the DDR SDRAM memory to be used to store data if desired.*

**FDAS\_DDR\_CONTROLLER\_CALIBRATION\_HPS (INTEL IP)**

**provides the DDR Controllers for DDRIF2#0 and DDRIF#1**

**FDAS\_DDR\_CONTROLLER\_CALIBRATION (INTEL IP)**

**provides the DDR Controllers for DDRIF2#2 and DDRIF#3**

DMA Transfer

**External 4 Gibi-Byte**

**DDR4 SDRAM Memory #1**

**External 4 Gibi-Byte**

**DDR4 SDRAM Memory #0. #2 and #3**

MC

Interface

Via

PCIe

*Note: In this implementation CLD\_PAGE, CONV\_PAGE and HSUM\_PAGE are fixed (static) in this implementation. However with a future implementation with more DDR Interfaces to CONV/HSUM a Paging technique shall enable increased processing performance.*

MC Bus

OVERLAP\_SIZE[9:0]

FOP\_SAMPLE\_NUM[22:0]

**MSIX**

MSI-X

Figure ‑ – DDRIF2 Module Location in FDAS

# Functional Specification

## DDRIF2 Connectivity

The figure below shows the connectivity of the DDRIF2 modules the FDAS design.

PCIE

Hard IP

Macro

CLD

Module

CONV

Module

HSUM

Module

DDRIF2 #1 Module containing clock boundary FIFOs

DDR 4 SDRAM#1

**FDAS**

512-bit @ 350MHz access from PCIe to each DDRIF2

512-bit @ 350MHz

1536-bit @ 350MHz

1536-bit @ 350MHz

64-bit @ 1333MHz

C

A

L

DDR\_

CONT

HPS

Ch 1

FDAS\_DDR\_

CONTROLLER\_HPS

CALIBRATION

(INTEL IP) #1

DDR\_

CONT

Ch 0

DDRIF2 #0 Module containing clock boundary FIFOs

DDR 4 SDRAM#0

64-bit @ 1333MHz

DDRIF2 #2 Module containing clock boundary FIFOs

DDR 4 SDRAM#2

64-bit @ 1333MHz

C

A

L

DDR\_

CONT

Ch 2

FDAS\_DDR\_

CONTROLLER\_

CALIBRATION

(INTEL IP) #1

DDR\_

CONT

Ch 3

DDRIF2 #3 Module containing clock boundary FIFOs

DDR 4 SDRAM#3

64-bit @ 1333MHz

“FIFO\_READY”, “FIFO\_FULL” and “DATA\_AVAIL” signals transferred between DDRIF2 modules for CONV/ HSUM data paths

512 bits to/from each DDRIF2

Figure ‑ : DDRIF2 Connectivity in FDAS

System design external to the DDRIF2 module shall ensure that only one FDAS traffic module attempts to accesses a DDRIF2 module at any one time. Hence the DDRIF2 module does not need any internal scheduling or calendar.

The intended operational sequence of FDAS shall be as follows:-

* For a period of time the PCIe Hard Macro shall write to External DDR SDRAM memory via DDRIF2 #1 simultaneously whilst the HSUM module reads from External DDR SDRAM memory via DDRIF2 #0, #2 and #3.
* For a period of time the CLD module shall read from External DDR SDRAM memory via DDRIF2 #1 simultaneously whilst the CONV module writes to External DDR SDRAM memory via DDRIF2 #0, #2 and #3.

As a diagnostic facility the PCIe Hard IP Macro shall be able to read from or write to either any SDRAM via its associated DDRIF2 module.

However as a precaution the DDRIF2 modules shall prioritise any access requests as follows:-

DRRIF2 #1

|  |  |
| --- | --- |
| Access Request | Priority |
| PCIe Hard Macro Read from External DDR SDRAM memory | 1 - Highest |
| PCIe Hard Macro Write to External DDR SDRAM memory | 2 |
| CLD Read from External DDR SDRAM memory | 3 - Lowest |

Table ‑ : DDRIF2 #1 Access Request Priority

DRRIF2 #0, #2 and #3

|  |  |
| --- | --- |
| Access Request | Priority |
| PCIe Hard Macro Read from External DDR SDRAM memory | 1 - Highest |
| PCIe Hard Macro Write to External DDR SDRAM memory | 2 |
| HSUM Read from External DDR SDRAM memory | 3 |
| CONV Write to External DDR SDRAM memory | 4 - Lowest |

Table ‑: DDRIF2 #0, #2 and #3 Access Request Priority

### DDRIF2 <> DDR Controller

The DDRIF2 module shall provide an interface to the Intel Agilex DDR Controller [Ref: Agilex DDR User Guide] via a 512-bit data bus.

The Intel Agilex DDR Controller shall connect to the external DDR4 SDRAM.

It is expected that the SDRAM devices will be identical, each with 4 Gibi-byte of storage. Each address location of the SDRAM device shall store 64-bits of data (i.e. 8 Bytes).

### PCIe Hard IP Macro <> DDRIF2 Interface

The PCIe Hard IP Macro shall present separate read and write interfaces to the DDRIF2 modules with 512-bit data buses @ 350MHz. The PCIe Hard IP Macro shall perform DMA transfers to and from the DDRIF2 module.

#### PCIe Hard IP Macro Write to External DDR SDRAM memory

PC/Computer > External DDR SDRAM memory DMA transfer requests shall be indicated by assertion of the RD\_DMA\_WRITE signal by the PC/Computer via the PCIe Hard IP Macro.

The PCIe Hard IP Macro shall present a 512-bit data bus @ 350MHz “RD\_DMA\_WRITE\_DATA[511:0]” to the DDRIF2 module, with a 32-bit address bus “RD\_DMA\_ADDRESS[31:0]”. Each value of the address bus corresponds to a byte offset, however with a 512-bit (i.e. 64-byte) bus the only legal values for the 32-bit address are:-

0x0000,0000

0x0000,0040

0x0000,0080

etc

The 64-bit byte based address (RD\_DMA\_ADDRESS[31:0]) to the DDRIF2 module should be populated as:-

|  |  |
| --- | --- |
| PCIe Interface Address to DDRIF2 module | |
| bit 31 bit 6 | bit 5 bit 0 |
| 64 byte Word Address from  PCIe Hard IP Macro  i.e. [25:0] | All zeros |

Table ‑: Addressing of DDRIF2 with Address from the PCIe Hard IP Macro

To target a particular byte within the 512-bit data bus the RD\_DMA\_BYTE\_EN[63:0] signal shall be used:-

|  |  |  |
| --- | --- | --- |
| RD\_DMA\_BYTE\_EN[63:0]  / Hex | Valid RD\_DMA\_WRITE\_DATA[255:0]  bits | Comment |
| 0x0000\_0000\_0000\_0001 | [7:0] | Byte #1 valid |
| 0x0000\_0000\_0000\_0002 | [15:8] | Byte #2 valid |
| 0x0000\_0000\_0000\_0004 | [23:16] | Byte #3 valid |
| 0x0000\_0000\_0000\_0008 | [31:24] | Byte #4 valid |
| : | : |  |
| 0x4000\_0000\_0000\_0000 | [503:496] | Byte #63 valid |
| 0x8000\_0000\_0000\_0000 | [511:504] | Byte #64 valid |
| : | : |  |
| : | : |  |
| 0xFFFF\_FFFF\_FFFF\_FFFF | [511:0] | All Bytes valid |

Table ‑ : RD\_DMA\_BYTE\_EN Description

**However it shall be a requirement of the software running on the PC/Computer to ensure that RD\_DMA\_BYTE\_EN[63:0] is set to 0xFFFF\_FFFF\_FFFF\_FFFF to indicate that all bytes in the 512-bit word are valid.**

If the DDRIF2 module is unable to respond to the DMA request it shall assert the RD\_DMA\_WAIT\_REQUEST signal. This shall cause the PCIe Hard IP Macro to pause its operation and until RD\_DMA\_WAIT\_REQUEST is de-asserted.

**However it should be noted that PCIe Hard IP Macro may still send a further 16 write accesses after RD\_DMA\_WAIT\_REQUEST has been asserted, due to the fact it has a “wait request allowance” value hard coded to16.**

The size of the burst transfer of data from the PC/Computer to the DDRIF2 module in terms of the number of 512-bit words shall be indicated by the RD\_DMA\_BURST\_COUNT[3:0] signal.

#### PCIe Hard IP Macro Read from External DDR SDRAM memory

External DDR SDRAM memory > PC/Computer DMA transfer requests shall be indicated by assertion of the WR\_DMA\_READ signal by the PC/Computer via the PCIe Hard IP Macro.

The DDRIF2 module shall present a 512-bit data bus “WR\_DMA\_READ\_DATA[511:0]” @ 350MHz to the PCIe Hard IP Macro, along with a WR\_DMA\_READ\_DATA\_VALID signal which shall be asserted when the data is valid.

The PCIe Hard IP Macro shall present a 32-bit address bus “WR\_DMA\_ADDRESS[31:0]” to the DDRIF2 module. Each value of the address bus corresponds to a byte offset, however with a 512-bit (i.e. 64-byte) bus the only legal values for the 32-bit address are:-

0x0000,0000

0x0000,0040

0x0000,0080

etc

The population of the WR\_DMA\_ADDRESS[31:0] shall follow the same method as described in section 3.1.2.1 Table 3‑3.

If the DDRIF2 module is unable to respond to the DMA request it shall assert the WR\_DMA\_WAIT\_REQUEST signal. This shall cause the PCIe Hard IP Macro to pause its operation and until WR\_DMA\_WAIT\_REQUEST is de-asserted.

The size of the burst transfer of data from the DDRIF2 module to the PC/Computer in terms of the number of 512-bit words shall be indicated by the WR\_DMA\_BURST\_COUNT[3:0] signal.

### DDRIF2 > CLD and DDRIF2 > HSUM Interface

This interface shall support the reading of data from external DDR SDRAM memory to the CLD module (via DDRIF2 #1) or HSUM module (via DDRIF2 #0, #2 and #3) via a 512-bit data bus DDR\_RD\_DATA[511:0] from each DDRIF2 instance.

The traffic module (CLD/HSUM) shall assert a “DDR\_RD\_EN” signal when a read is required, along with an address signal DDR\_RD\_ADDR[31:0]. Each value of the address bus corresponds to a byte offset, however with a 512-bit (i.e. 64-byte) data bus the only legal values for the 32-bit address are:-

0x0000,0000

0x0000,0040

0x0000,0080

etc

The DDRIF2 module shall indicate when it is able to accept the address by controlling the DDR\_RD\_WAIT\_REQUEST signal to the CLD/HSUM module. In the case of DDRIF2 #0, #2 and #3 “FIFO\_READY” and “FIFO\_FULL” signals passed between the DDRIF2 module instances ensure that the DDR\_RD\_WAIT\_REQUEST is only de-asserted when all three DDRIF2 module instances have space to store the address.

When the DDRIF2 module supplies valid data to the CLD/HSUM module it shall assert the DDR\_RD\_DATA\_VALID signal.

In the case of DDRIF2 #0, #2 and #3 “DATA\_AVAIL” signals passed between the DDRIF2 module instances ensure that the DDR\_RD\_DATA\_VALID is only asserted when all three DDRIF2 module instances have data available.

### CONV > DDRIF2 Interface

This interface shall support the writing of data from the CONV module to DDRIF2#0, #2 and #3 via a 512-bit data bus DDR\_WR\_DATA[511:0] to each DDRIF2 instance.

The CONV module shall assert a “DDR\_WR\_EN” signal when a write is required, along with an address signal DDR\_WR\_ADDR[31:0]. Each value of the address bus corresponds to a byte offset, however with a 512-bit (i.e. 64-byte) data bus the only legal values for the 32-bit address are:-

0x0000,0000

0x0000,0040

0x0000,0080

etc

The DDRIF2 #0, #2 and #3 module instances shall indicate when it is able to accept data and address by controlling the DDR\_WR\_WAIT\_REQUEST signal to the CONV module. “FIFO\_READY” signals passed between the DDRIF2 module instances ensure that the DDR\_WR\_WAIT\_REQUEST is only de-asserted when all three DDRIF2 module instances have space to store the data and address.

## DDRIF2 Architecture

**DDRIF2**

RD\_DMA\_WRITE\_DATA[511:0]

RD\_DMA\_BURST\_COUNT[3:0]

RD\_DMA\_BYTE\_EN[63:0]

IF\_

SEL

RD\_DMA\_WAIT\_REQUEST

DDR\_WR\_EN

AMM\_ADDRESS[31:0]

RX\_PCIE\_IF

RX\_PROC\_IF

RD\_DMA\_WRITE

RD\_DMA\_ADDRESS[63:0]

DDR\_WR\_ADDR[31:0]

DDR\_WR\_DATA[511:0]

DDR\_WR\_WAIT\_REQUEST

AMM\_WRITE\_DATA[511:0]

RX\_ADDR [31:0]

AMM\_WRITE

AMM\_ WAIT\_REQUEST

AMM\_BURSTCOUNT[6:0]

= “0000001”

AMM\_BYTE\_EN[63:0]

= 0xFFFF,FFFF,FFFF,FFFF

TX\_PCIE\_IF

TX\_PROC\_IF

WR\_DMA\_ADDRESS[63:0]

DDR\_RD\_ADDR[31:0]

DDR\_RD\_DATA[511:0]

DDR\_RD\_DATA\_VALID

AMM\_READ\_DATA[511:0]

AMM\_READ\_DATA\_VALID

WR\_DMA\_READ\_DATA[511:0]

DDR\_RD\_EN

WR\_DMA\_READ

WR\_DMA\_BURST\_COUNT[3:0]

WR\_DMA\_READ\_DATA\_VALID

WR\_DMA\_WAIT\_REQUEST

AMM\_READ

FIFO\_READY

FIFO\_READY

RX\_FIFO

Store 512-bit data words and DDR address

RX\_DATA

\_PCIE

[511:0]

RX\_ADDR

\_PCIE

[31:0]

RX\_DATA

\_PROC

[511:0]

RX\_ADDR

\_PROC

[31:0]

VALID

WADDR

WAIT

WAG\_IN

RAG\_IN

RX\_DATA

\_PCIE

[511:0]

RX\_ADDR

\_PCIE

[31:0]

REQ

VALID

WAG

\_IN

RAG\_IN

RX\_FIFO

Store 512-bit data words and DDR address

RADDR

WADDR

RADDR

WAIT

REQ

RX\_DATA

\_PROC

[511:0]

RX\_ADDR

\_PROC

[31:0]

RD\_ADDR

\_PCIE

[31:0]

RD\_ADDR

\_PROC

[31:0]

VALID

VALID

TX\_DATA

[511:0]

DATA\_FIFO

\_512

Store

DDR Data

TX\_DATA

[511:0]

TX\_DATA

[511:0]

VALID\_

DATA

RADDR

RAG\_OUT512

WAG\_OUT

VALID\_DATA

RAG\_OUT\_3

\_512

WAG\_OUT

DATA\_

FIFO

\_512

Store

DDR Data

WADDR

RADDR

WADDR

TX\_DATA

[511:0]

DDR\_RD\_WAIT\_REQUEST

READ\_WRITE\_SEL

PCIE\_PROC\_SEL

CLK\_SYS

CLK\_PCIE

RST\_SYS\_N

CLK\_DDR

FIFO\_FULL

FIFO\_FULL

RX\_DATA [511:0]

WRITE

WAIT

READ

PROC\_READ\_DATA\_VALID

PCIE\_READ\_DATA\_VALID

VALID

WR\_DATA

\_PCIE

[511:0]

WR\_ADDR

\_PCIE

[31:0]

VALID

WR\_DATA

\_PROC

[511:0]

WR\_ADDR

\_PROC

[31:0]

RST\_DDR\_N

RST\_PCIE\_N

READ

WRITE

READ

WRITE

READ

WRITE

READ

WRITE

FIFO\_READY\_OUT

FIFO\_READY\_IN\_1

FIFO\_READY\_IN\_2

FIFO\_FULL\_IN\_1

FIFO\_FULL\_IN\_2

DATA\_AVAIL\_OUT

FIFO\_FULL\_OUT

DATA\_AVAIL\_IN\_1

DATA\_AVAIL\_IN\_2

Figure ‑ : DDRIF2 Architecture Block Diagram

### Interface Select: IF\_SEL Sub-Module

The IF\_SEL sub-module shall select which interface is to be given access to the External DDR SDRAM memory. Each of the input active high Enable signals shall be edge detected (both rising and falling edges detected) on the CLK\_DDR domain (333.33MHz). A priority shall be applied in the event that multiple input Enable signals are asserted simultaneously according to the table below:-

|  |  |  |
| --- | --- | --- |
| Enable Signal Requesting External DDR SDRAM memory access | Interface | Priority |
| WR\_DMA\_READ | PCIe Read from External DDR SDRAM memory | 1 - Highest |
| RD\_DMA\_WRITE | PCIe Write to External DDR SDRAM memory | 2 |
| DDR\_RD\_EN | CLD / HSUM Read from External DDR SDRAM memory | 3 |
| DDR\_WR\_EN | CONV Write to External DDR SDRAM memory | 4 - Lowest |

Table ‑ : Interface Priority Selection

In normal fault free operation only one interface shall be requesting External DDR SDRAM memory access at any one time.

The output of the IF\_SEL sub-module shall be used to control muxes to select interface to be granted access to the External DDR SDRAM memory.

### Receive PCIE Interface: RX\_PCIE\_IF Sub-Module

The RX\_PCIE\_IF sub-module shall terminate the data from the PCIe Hard IP Macro.

When the PCIe Hard IP Macro attempts to write to the External DDR SDRAM memory it shall assert the RD\_DMA\_WRITE signal (active high) which shall enable the RX\_PCIE\_IF sub-module.

The RX\_PCIE\_IF sub-module shall de-assert the RD\_DMA\_WAIT\_REQUEST if the associated RX\_FIFO sub-module has spare capacity as indicated by the “FIFO\_READY” signal from the RX\_FIFO sub-module’s Write Address Generator sub-module (WAG\_IN).

The transfer of data and address shall then commence with the RX\_PCIE\_IF sub-module asserting the VALID signal to the WAG\_IN sub-module when data is available and with the RX\_PCIE\_IF sub-module continually monitoring the fill level of the associated RX\_FIFO sub-module (via “FIFO\_READY”) and raising RD\_DMA\_WAIT\_REQUEST as and when necessary.

**To support the PCIe Hard IP Macro which has a WaitRequest Allowance of 16 this version of the RX\_PCIE\_IF sub-module only uses the RD\_DMA\_WRITE** **signal to qualify the existence of new data from the PCIe Hard IP Macro. The PCIe Hard IP Macro may still send a further 16 write accesses after RD\_DMA\_WAIT\_REQUEST has been asserted.**

The RD\_DMA\_BURST\_COUNT[3:0] signal and the RD\_DMA\_ADDRESS[31:0] signal shall be used to generate the series of 32-bit External DDR SDRAM memory addresses to accompany the data.

The PCIe Transaction Layer Packet is terminated in the PCIe Hard IP Macro and only the payload data is presented to the DDRIF2 module.

The incoming data RD\_DMA\_WRITE\_DATA[511:0] shall be stored in the RX\_FIFO sub-module with an associated 32-bit External DDR SDRAM memory address per 512-bit data word. It is a requirement on the PC/Computer that RD\_DMA\_BYTE\_EN[63:0] is always all 1’s (i.e. all bytes of RD\_DMA\_WRITE\_DATA[511:0] are valid).

### Receive Processing Interface: RX\_PROC\_IF Sub-Module

The RX\_PROC\_IF sub-module shall terminate the data from the CONV module.

When the CONV module attempts to write to the External DDR SDRAM memory it shall assert the DDR\_WR\_EN signal (active high) which shall enable the RX\_PROC\_IF sub-module.

The RX\_PROC\_IF sub-module shall de-assert the DDR\_WR\_WAIT\_REQUEST if the associated local RX\_FIFO sub-module has spare capacity as indicated by the “FIFO\_READY” signal from the RX\_FIFO sub-module’s Write Address Generator sub-module (WAG\_IN) and the “FIFO\_READY\_IN\_1” and “FIFO\_READY\_IN\_2” signals from the other DDRIF2 modules supporting the CONV module.

i.e.

DDR\_WR\_WAIT\_REQUEST <=

**NOT**(FIFO\_READY **AND** FIFO\_READY\_IN\_1 **AND** FIFO\_READY\_IN\_2)

The transfer of data and address shall then commence, with the RX\_PROC\_IF sub-module asserting the VALID signal to the WAG\_IN sub-module when data is available and with the RX\_PROC\_IF sub-module continually monitoring the fill level of the local RX\_FIFO sub-module and other DDRIF2 module (via “FIFO\_READY” “FIFO\_READY\_IN\_1” and “FIFO\_READY\_IN\_2”) and raising DDR\_WR\_WAIT\_REQUEST as and when necessary.

The RX\_PROC\_IF sub-module shall present the incoming address DDR\_WR\_ADDR[31:0]

and Data DDR\_WR\_DATA[511:0] to the RX\_FIFO sub-module so that the data for the External DDR SDRAM memory can be stored in the RX\_FIFO sub-module along with its External DDR SDRAM memory address.

### RX\_FIFO , WAG\_IN & RAG\_IN Sub-Modules

**These versions of RX\_FIFO, WAG\_IN and RAG\_IN sub-modules support a 64 location FIFO and require at least 48 free locations to be deemed to be capable of receiving more access commands from the PCIe Hard IP Macro. This infers DDRIF2 supports a WaitRequest Allowance of 48.**

The RX\_FIFO sub-module provides a rate adaption function between the CLK\_DDR domain and the CLK\_SYS or CLK\_PCIE domains.

DDR address, data and read/write information shall be stored in the RX\_FIFO sub-module. The RX\_FIFO submodule shall have a depth of 64 locations and a word width of 546-bits so that the 512-bit data for the External DDR SDRAM memory, the 32-bit Address for the External DDR SDRAM memory and the READ/WRITE signals can be stored in each RX\_FIFO sub-module location. (If desired the lower 6 bits of the address can be omitted from storage in the RX\_FIFO sub-module since they shall always be “000000” for a 512-bit data word).

The Write Address Generator sub-module (WAG\_IN) and Read Address Generator sub-module (RAG\_IN) shall pass their FIFO pointer positions between each other with meta-stable protection (for example using a gray code style). In this way the WAG\_IN sub-module shall be able to determine when there is space in the RX\_FIFO and RAG\_IN sub-module shall be able to determine when the RX\_FIFO sub-module is empty. A “FIFO\_READY” signal shall indicate if the FIFO is able to accept more information. The “FIFO\_READY” signal shall be asserted if the FIFO is able to accept at least 48 more access commands, and since this signal ultimately controls the RD\_DMA\_WAIT\_REQUEST to the PCIe Hard IP Macro, this infers the DDRIF2 module has a WaitRequest Allowance of 48.

The WAG\_IN sub-module shall be sensitive to the “VALID” signal from RX\_PCIE\_IF/RX\_PROC\_IF/ TX\_PCIE\_IF/TX\_PROC\_IF sub-modules indicating that data is available to be stored in the RX\_FIFO sub-module and shall increment each clock cycle that VALID is asserted.

The RAG\_IN sub-module shall be sensitive to the “WAIT” signal indicating that the downstream DDR Controller is not able to accept data and shall increment each clock cycle that WAIT is not asserted (until the Read pointer advances to the Write pointer, - indicating the RX\_FIFO sub-module is empty). When the RAG\_IN sub-module increments it shall assert the “REQ” signal which shall be used by the downstream to qualify the READ / WRITE signals from the RX\_FIFO sub-module to the DDR Controller to indicate a read or write request.

### Transmit PCIE Interface: TX\_PCIE\_IF Sub-Module

The TX\_PCIE\_IF sub-module shall terminate the address from the PCIe Hard IP Macro and source the data to the PCIe Hard IP Macro.

When the PCIe Hard IP Macro attempts to read the External DDR SDRAM memory it shall assert the WR\_DMA\_READ signal (active high) which shall enable the TX\_PCIE\_IF sub-module.

The TX\_PCIE\_IF sub-module shall de-assert the WR\_DMA\_WAIT\_REQUEST if the associated RX\_FIFO sub-module has spare capacity as indicated by the “FIFO\_READY” signal from the RX\_FIFO sub-module’s Write Address Generator sub-module (WAG\_IN), and the associated DATA\_FIFO\_512 sub-module is not getting too full as indicated by the “FIFO\_FULL” signal from the DATA\_FIFO\_512 sub-module’s Read Address Generator sub-module (RAG\_OUT\_512).

The transfer of the address information shall then commence with the TX\_PCIE\_IF sub-module asserting the VALID signal to the WAG\_IN sub-module when address information is available and with the TX\_PCIE\_IF sub-module continually monitoring the fill level of the associated RX\_FIFO sub-module (via “FIFO\_READY”) and the DATA\_FIFO\_512 sub-module (via “FIFO\_FULL”), raising WR\_DMA\_WAIT\_REQUEST as and when necessary.

The WR\_DMA\_BURST\_COUNT[3:0] signal and the WR\_DMA\_ADDRESS[63:0] signal shall be used to generate the series of 32-bit External DDR SDRAM memory addresses to be stored in the RX\_FIFO sub-module.

The outgoing data “WR\_DMA\_READ\_DATA[511:0]” back to the PCIe Hard IP Macro shall be validated by the WR\_DMA\_READ\_DATA\_VALID signal.

### Transmit Processing Interface: TX\_PROC\_IF Sub-Module

The TX\_PROC\_IF sub-module shall terminate the address from the CLD / HSUM module and source the data to the CLD/ HSUM module.

When the CLD/HSUM module attempts to read the External DDR SDRAM memory it shall assert the DDR\_RD\_EN signal (active high) which shall enable the TX\_PROC\_IF sub-module.

The TX\_PROC\_IF sub-module shall de-assert the DDR\_RD\_WAIT\_REQUEST if the associated local RX\_FIFO sub-module has spare capacity as indicated by:-

1. The “FIFO\_READY” signal from the RX\_FIFO sub-module’s Write Address Generator sub-module (WAG\_IN) and the “FIFO\_READY\_IN\_1” and “FIFO\_READY\_IN\_2” signals from the other DDRIF2 modules supporting the HSUM module (in the case of supporting the CLD module the “FIFO\_READY\_IN\_1” and “FIFO\_READY\_IN\_2” signals are tied to logic ‘1’)
2. The “FIFO\_FULL” signal from the DATA\_FIFO\_512 sub-module’s Read Address Generator sub-module (RAG\_OUT\_512) and the “FIFO\_FULL\_IN\_1” and “FIFO\_FULL\_IN\_2” signals from the other DDRIF2 modules supporting the HSUM module (in the case of supporting the CLD module the “FIFO\_FULL\_IN\_1” and “FIFO\_FULL\_IN\_2” signals are tied to logic ‘0’)

i.e.

DDR\_RD\_WAIT\_REQUEST <=

(**NOT**(FIFO\_READY **AND** FIFO\_READY\_IN\_1 **AND** FIFO\_READY\_IN\_2)

**OR**

(FIFO\_FULL **OR** FIFO\_FULL\_IN\_1 **OR** FIFO\_FULL\_IN\_2)

The transfer of data shall then commence, with the RX\_PROC\_IF sub-module asserting the VALID signal to the WAG\_IN sub-module when data is available and with continually monitoring the fill level of the local RX\_FIFO sub-module and other DDRIF2 module (via “FIFO\_READY” “FIFO\_READY\_IN\_1” and “FIFO\_READY\_IN\_2”) and raising DDR\_WR\_WAIT\_REQUEST as and when necessary.

The transfer of the address information “DDR\_RD\_ADDR[31:0]” shall then commence with the TX\_PROC\_IF sub-module asserting the VALID signal to the WAG\_IN sub-module when address information is available and with the TX\_PROC\_IF continually monitoring the fill level of the local RX\_FIFO sub-module and other DDRIF2 module (via “FIFO\_READY” “FIFO\_READY\_IN\_1” and “FIFO\_READY\_IN\_2”) and

and the DATA\_FIFO\_512 sub-module (via “FIFO\_FULL”, “FIFO\_FULL\_IN\_1” and “FIFO\_FULL\_IN\_2”)), raising DDR\_RD\_WAIT\_REQUEST as and when necessary.

The TX\_PROC\_IF sub-module supplies “DDR\_RD\_DATA[511:0]” valid data to the CLD/HSUM module with the DDR\_RD\_DATA\_VALID signal asserted if the VALID\_DATA signal from the RAG\_OUT\_3\_512 sub-module is asserted.

### DATA\_FIFO\_512, WAG\_OUT , RAG\_OUT\_512 and RAG\_OUT\_3\_512 Sub-Modules

The DATA\_FIFO\_512 sub-module provides a rate adaption function between the CLK\_DDR domain and the CLK\_SYS or CLK\_PCIE domains.

DDR data information shall be stored in the DATA\_FIFO\_512 sub-module when it is valid. The DATA\_FIFO\_512 sub-module shall have a depth of 512locations and a word width of 512-bits so that the 512-bit data from the External DDR SDRAM memory can be stored in each DATA\_FIFO\_512 sub-module location.

The reason for the large DATA\_FIFO\_512 sub-module depth is to take account of the latency of the external DDR memory. If the DATA\_FIFO\_512 sub-module starts to get too full (i.e. exceeds half full) the WAIT\_REQUEST to the FDAS module (CLD/HSUM) or PCIe Hard IP Macro shall be asserted which in turn shall result in the RX\_FIFO sub-module pausing thus stopping any further read requests to the external DDR memory. The external DDR memory shall continue to process read requests previously received which will need to be stored in the DATA\_FIFO\_512 sub-module. The largest read request shall be from the HSUM module when it requests 40 FOP columns (each of 85 rows with 32-bit values per location), which is equivalent to 212.5 DATA\_FIFO\_512 locations, - hence half the DATA\_FIFO\_512 sub-module capacity is set to 256 locations so that it exceeds 212.5 locations.

The Write Address Generator sub-module (WAG\_OUT) shall pass its FIFO pointer position to the Read Address Generator sub-module (RAG\_OUT\_512) with meta-stable protection (for example using a gray code style). In this way the RAG\_OUT\_512 sub-module shall be able to determine when the DATA\_FIFO\_512 sub-module is empty or when it is getting too full, raising the FIFO\_FULL signal when necessary to raise the WAIT signal to CLD/HSUM via the TX\_PROC\_IF sub-module or PCIE Hard IP Macro via the TX\_PCIE\_IF sub-module.

The RAG\_OUT\_512 and RAG\_OUT\_3\_512 sub-modules shall support addressing to the DATA\_FIFO\_512 sub-module based on a 512-bit read data.

The WAG\_OUT sub-module shall be sensitive to the “VALID” signal from the DDR Controller indicating that Data is available to be stored in the DATA\_FIFO\_512 sub-module and shall increment each clock cycle that VALID is asserted.

The RAG\_OUT\_512 sub-module shall increment its pointer on each clock cycle that the DATA\_FIFO\_512 sub-module is not empty. The RAG\_OUT\_512 sub-module shall indicate that the data is valid each time in increments and also indicate if the DATA\_FIFO\_512 sub-module is full.

The RAG\_OUT\_3\_512 sub-module is similar to the RAG\_OUT\_512 sub-module, however it will only increment its pointer each clock cycle if the DATA\_FIFO\_512 sub-module is not empty and the “DATA\_AVAIL\_IN\_1” and “DATA\_AVAIL\_IN\_2” from the other DDRIF2 modules are all asserted (in the case of supporting the CLD module the DATA\_AVAIL\_IN\_1” and “DATA\_AVAIL\_IN\_2” signals are tied to logic ‘1’).

# Design Requirement Tags

The relevant design requirement tags from the FDAS Design Specification Document for the DDRIF2 module are listed below.

| **Design Requirement Tag** | **Description** | **Comment** |
| --- | --- | --- |
| FDAS.DATAIN:040/A | The FFT sequence shall be stored by FDAS in external DDR SDRAM memory, with all values static for a known period of time. | The freq-bin samples from the PC/Computer via the PCIe shall be stored in External DDR SDRAM memory. |
| FDAS.DATAIN:060/A | The external memory containing the input data shall have two pages, to allow one page to be written to whilst the other page is being accessed by the FDAS core processing. | In this implementation with reduced complexity DDR interface only one page of storage shall be required. |
| FDAS.DATAIN:070/A | For a given page the allocation of freq-bins to memory locations shall be known and fixed, thus the memory address can be used to identify the freq-bin number and the real and imaginary value of each freq-bin. | The freq-bin locations within the External DDR SDRAM memory are fixed. |
| FDAS.DATAIN:080/A | The input data to FDAS (i.e. the observed data) shall be stored in external DDR SDRAM memory with the information shared equally across the available SDRAM devices. The **internal** FDAS data bus width shall be dependent on the number of external DDR SDRAM memory interfaces. A generic “ddr\_g” shall indicate the number of external DDR interfaces. | In this implementation with reduced complexity DDR interface only one External DDR SDRAM device shall be used to store the freq-bin samples, with a separate External DDR SDRAM device being used to store the FOP. |
| FDAS.DATAIN:090/A | External commands from Software via a Monitor and Control (MC) interface shall instruct FDAS to commence reading the FFT sequence from memory, and also which memory page to access. | In this implementation with reduced complexity DDR interface the commands shall be such that only one internal module is attempting access to an External DDR SDRAM device at any one time. |
| FDAS.DIAGNOSTIC:010/A | It shall be possible, via the PCIe, for the MC Interface to have read/write access to any location of the external DDR memory. Ideally block accesses shall possible so that a region of the DDR external memory can be accessed via the PCIe with a minimum number of read/write commands. | In this implementation with reduced complexity DDR interface, the PCIe shall access to the External DDR SDRAMs via two separate DDRIF2 modules. |

Table ‑ : DDRIF2 Design Requirement Tags

# Interface Specification

| Signal | Direction | Clock Domain | Description |
| --- | --- | --- | --- |
| **DDRIF2 Interface to PCIE Hard IP Read DMA Interface to Read Data from PC/Computer and write to External DDR SDRAM memory** |  |  |  |
| RD\_DMA\_WRITE | IN | CLK\_PCIE | When asserted, indicates that the PCIe Hard IP is ready to write data to External DDR SDRAM memory.  ‘1’ = Write |
| RD\_DMA\_ADDRESS[31:0] | IN | CLK\_PCIE | Address of an External DDR SDRAM memory location to which data from the PCIe Hard IP Macro is to be written.  [31] = msb |
| RD\_DMA\_WRITE\_DATA[511:0] | IN | CLK\_PCIE | Data from the PCIe Hard IP Macro which is to be written to External DDR SDRAM memory.  [511] = msb |
| RD\_DMA\_BURST\_COUNT[3:0] | IN | CLK\_PCIE | Specifies the burst count in 512-bit words. This bus is 4 bits for the 512-bit interface. |
| RD\_DMA\_BYTE\_EN[63:0] | IN | CLK\_PCIE | Specifies which bytes of a word or are valid. |
| RD\_DMA\_WAIT\_REQUEST | OUT | CLK\_PCIE | When asserted, indicates that the DDRIF2 module is not ready to receive data from the PCIe Hard IP Macro.  ‘1’ = Wait, DDRIF2 is not ready. |
|  |  |  |  |
| **DDRIF2 Interface to PCIE Hard IP Write DMA Interface to Read Data from External DDR SDRAM memory and write to PC/Computer** |  |  |  |
| WR\_DMA\_READ | IN | CLK\_PCIE | When asserted, indicates that the CLD/HSUM module is ready to read data from External DDR SDRAM memory.  ‘1’ = Read |
| WR\_DMA\_ADDRESS[31:0] | IN | CLK\_PCIE | Specifies the External DDR SDRAM memory address for the data to be read from.  [31] = msb |
| WR\_DMA\_READ\_DATA[511:0] | OUT | CLK\_PCIE | Data from External DDR SDRAM memory to the PCIe Hard IP Macro.  [511] = msb |
| WR\_DMA\_BURST\_COUNT[3:0] | IN | CLK\_PCIE | Specifies the burst count in 512-bit words. This bus is 4 bits for the 512-bit interface. |
| WR\_DMA\_READ\_DATA\_VALID | OUT | CLK\_PCIE | Indication that the WR\_DMA\_READ\_DATA from the DDRIF2 module to the PCIe Hard IP Macro is valid.  ‘1’ = Valid |
| WR\_DMA\_WAIT\_REQUEST | OUT | CLK\_PCIE | When asserted, indicates that the DDRIF2 module is not ready to process read requests from the PCIe Hard IP Macro.  ‘1’ = Wait, DDRIF2 is not ready. |
| **DDRIF2 Interfaces common to CONV and HSUM modules** |  |  |  |
| FIFO\_READY\_IN\_1 | IN | CLK\_SYS | Asserted when the RX FIFO of one of the other DDRIF2 module instances is able to accept Address from CONV or HSUM and Data from CONV to access the DDR4 SDRAM  ‘1’ = Able to accept Address & Data |
| FIFO\_READY\_IN\_2 | IN | CLK\_SYS | Asserted when the RX FIFO of one of the other DDRIF2 module instances is able to accept Address from CONV or HSUM and Data from CONV to access the DDR4 SDRAM  1’ = Able to accept Address & Data |
| FIFO\_READY\_OUT | OUT | CLK\_SYS | Asserted when the local RX FIFO is able to accept Address from CONV or HSUM and Data from CONV to access the DDR4 SDRAM  1’ = Able to accept Address & Data |
|  |  |  |  |
| **DDRIF2 Interface to CONV Module** |  |  |  |
| DDR\_WR\_EN | IN | CLK\_SYS | When asserted, indicates that the CONV module is ready to write data to External DDR SDRAM memory.  ‘1’ = Write |
| DDR\_WR\_ADDR[31:0] | IN | CLK\_SYS | Address of an External DDR SDRAM memory location to which data from the CONV module is to be written.  [31] = msb |
| DDR\_WR\_DATA[511:0] | IN | CLK\_SYS | Data from the CONV module which is to be written to External DDR SDRAM memory.  [511] = msb |
| DDR\_WR\_WAIT\_REQUEST | OUT | CLK\_SYS | When asserted, indicates that the DDRIF2 module is not ready to receive data from the CONV module.  ‘1’ = Wait, DDRIF2 is not ready. |
|  |  |  |  |
| **DDRIF2 Interface to CLD & HSUM Modules** |  |  |  |
| DDR\_RD\_EN | IN | CLK\_SYS | When asserted, indicates that the CLD/HSUM module is ready to read data from External DDR SDRAM memory.  ‘1’ = Read |
| DDR\_RD\_ADDR[31:0] | IN | CLK\_SYS | Specifies the External DDR SDRAM memory address for the data to be read from.  [31] = msb |
| DDR\_RD\_DATA[511:0] | OUT | CLK\_SYS | Data from External DDR SDRAM memory to the CLD/HSUM module.  [511] = msb |
| FIFO\_FULL\_IN\_1 | IN | CLK\_SYS | Asserted when the DATA\_FIFO \_512 of one of the other DDRIF2 module instances supporting HSUM access is full and hence no more Read requests from HSUM can be accepted.  ‘1’ = DATA\_FIFO\_512 is full |
| FIFO\_FULL\_IN\_2 | IN | CLK\_SYS | Asserted when the DATA\_FIFO \_512 of one of the other DDRIF2 module instances supporting HSUM access is full and hence no more Read requests from HSUM can be accepted.  ‘1’ = DATA\_FIFO\_512 is full |
| DATA\_AVAIL\_IN\_1 | IN | CLK\_SYS | Asserted when the DATA\_FIFO \_512 of one of the other DDRIF2 module instances supporting HSUM access has data available to supply to HSUM.  ‘1’ = Data is available |
| DATA\_AVAIL\_IN\_2 | IN | CLK\_SYS | Asserted when the DATA\_FIFO \_512 of one of the other DDRIF2 module instances supporting HSUM access has data available to supply to HSUM.  ‘1’ = Data is available |
| DDR\_RD\_DATA\_VALID | OUT | CLK\_SYS | Indication that the DDR\_RD\_DATA from DDRIF2 to the CLD/HSUM module is valid.  ‘1’ = Valid |
| DDR\_RD\_WAIT\_REQUEST | OUT | CLK\_SYS | When asserted, indicates that the DDRIF2 module is not ready to process read requests from the CLD / HSUM module.  ‘1’ = Wait, DDRIF2 is not ready. |
| FIFO\_FULL\_OUT | OUT | CLK\_SYS | Asserted when the local DATA\_FIFO \_512 is full and hence no more Read requests from HSUM can be accepted.  ‘1’ = DATA\_FIFO\_512 is full |
| DATA\_AVAIL\_IN\_1 | OUT | CLK\_SYS | Asserted when the the DATA\_FIFO \_512 has data available to supply to HSUM.  ‘1’ = Data is available |
|  |  |  |  |
| **Interface to the DDR Controller** |  |  |  |
| AMM\_WRITE | OUT | CLK\_DDR | When asserted, indicates that the DDRIF2 module is ready to write data to External DDR SDRAM memory.  ‘1’ = Write |
| AMM\_WRITE\_DATA[511:0] | OUT | CLK\_DDR | Data from the DDRIF2 module which is to be written to External DDR SDRAM memory.  [255] = msb |
| AMM\_BURSTCOUNT[6:0] | OUT | CLK\_DDR | Specifies the burst count. This is hardwired to a burst of one 512-bit word  AMM\_BURSTCOUNT[6:0] = “0000001” |
| AMM\_BYTE\_EN[63:0] | OUT | CLK\_DDR | Specifies which bytes of a word or are valid. This is hardwired to 0xFFFF,FFFF, FFFF,FFFF to indicate all bytes of the 512-bit word are valid  AMM\_BYTE\_EN[63:0]  = 0xFFFF,FFFF, FFFF,FFFF |
| AMM\_ADDRESS[31:0] | OUT | CLK\_DDR | Address of an External DDR SDRAM memory location to which data from the DDRIF2 module is to be written.  [31] = msb |
| AMM\_READ | OUT | CLK\_DDR | When asserted, indicates that the DDRIF2 module is ready to read data from External DDR SDRAM memory.  ‘1’ = Read |
| AMM\_READ\_DATA[511:0] | IN | CLK\_DDR | Data from External DDR SDRAM memory to the DDRIF2 module.  [255] = msb |
| AMM\_READ\_DATA\_VALID | IN | CLK\_DDR | Indication that the AMM\_READ\_DATA from the DDR Controller is valid.  ‘1’ = Valid |
| AMM \_WAIT\_REQUEST | IN | CLK\_DDR | When asserted, indicates that the DDR Controller is not ready to process requests from the DDRIF2 module.  ‘1’ = Wait, DDR Controller is not ready  . |
|  |  |  |  |
| **Global Clock/Resets** |  |  |  |
| CLK\_SYS | IN | - | 350MHz Core System Clock |
| CLK\_PCIE | IN | - | 350MHz clock from the PCIe Hard IP macro |
| CLK\_DDR | IN | - | 333.33MHz clock from the DDR Controller |
| RST\_SYS\_N | IN | - | Asynchronous Logic reset for the CLK\_SYS domain  ‘0’=Reset |
| RST\_PCIE\_N | IN | - | Asynchronous Logic reset for the CLK\_PCIE domain  ‘0’=Reset |
| RST\_DDR\_N | IN | - | Asynchronous Logic reset for the CLK\_DDR domain  ‘0’=Reset |

Table ‑ : DDRIF2 Pinlist

# MCI Memory Mapped Interface

The DDRIF2 module does not have an MCI memory mapped interface.

# Design Parameterisation

The DDRIF2 module does not have any generic parameterisation.

# References

|  |  |  |
| --- | --- | --- |
| **Bookmark** | **Reference** | **Description** |
| [Ref: Agilex DDR User Guide] | External Memory Interfaces Intel Agilex FPGA IP User Guide UG-20218 2021.07.09 | Intel DDR interface User Guide for the Agilex FPGA family. |

# Abbreviations and Acronyms

|  |  |
| --- | --- |
| DM | Dispersion Measure |
| DMA | Direct Memory Access |
| DDR | Double Data Rate |
| FDAS | Fourier Domain Acceleration Search |
| FFT | Fast Fourier Transform |
| FOP | Filter Output Plane |
| FPGA | Field Programmable Gate Array |
| IEEE | Institute of Electrical Engineers |
| IP | Intellectual Property |
| MC | Micro Controller Interface |
| MSI-X | Extended Message Signalled Interrupt |
| PCIe | Peripheral Component Interconnect Express |
| SDRAM | Synchronous Dynamic RAM |