

**FDAS MCI\_TOP Module Design Specification for Intel Agilex F Implementation**

FDAS\_MCI\_TOP\_DS Revision 2 Draft C

Classification: UNRESTRICTED

Document type: DTE

Date: 2023-06-13

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 |  | 13/06/2023 |
| 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 | 29/04/2022 | First Issue of MCI\_TOP for the Intel Agilex implementation of FDAS with address decode space for the MSIX module. |
| Issue 2 Draft B | 12/04/2023 | Now have separate Reset Control and reporting for four DDR SDRAM interfaces. Also the sense of the Resets from the MCI registers is inverted so that at powerup no resets are applied. |
| Issue 2 Draft C | 13/06/2023 | Updated the Bitamp to clarify which circuits are reset by which MCI Reset Config bit |

**Table of Contents**

1 Introduction 4

2 Place in the System 5

3 Functional Specification 6

3.1 MCI\_TOP\_DECODEMCI Sub-Module 7

3.2 MCI\_TOPMCI Sub-Module 8

3.3 MCI\_TOP\_DATAOR Sub-Module 9

3.4 MCI\_TOP\_RESET Sub-Module 9

4 Design Requirement Tags 10

5 Interface Specification 11

6 MCI Memory Mapped Interface 14

6.1 Memory map 14

6.2 Generic values 14

6.3 Module MCI\_TOP 14

6.3.1 Inventory 14

6.3.2 Monitor and Reset Control 16

7 Design Parameterisation 19

8 Abbreviations and Acronyms 20

# Introduction

This document captures the requirements for the MCI\_TOP module used in the FDAS FPGA.

The MCI\_TOP module shall be required to:-

* Provide the read only registers containing the Device, Version and Revision information for the FPGA to the user via the MCI Interface.
* Provide the DDR Calibration Pass / Fail signals from four DDR\_CONTROLLER modules to the host computer via the MC/PCIe interface. If calibration fails the DDR\_CONTROLLER(s) can be individually reset by the host computer via the MC/PCIe interface.
* Provide the manual Reset signals to allow the user to manually reset the main processing modules within FDAS via the MCI Interface. These are logically ANDed (i.e. using AND gates) with the main system reset. Note that the manual resets are configured active high (1 = reset) in the MCI memory map and they are then inverted. This is to ensure that at power up no resets are applied as the MCI reset registers power up at logic ‘0’.
* Decode the Address from the PCIF module so that the correct module (CTRL, MSIX, CONV, HSUM and TOPMCI with MCI\_TOP) within FDAS can be targeted for Micro-configuration access by the PC/Computer via the PCIe interface.
* Logically “OR” the MCDATA signals from CTRL, MSIX, CONV, HSUM modules and TOPMCI sub-module within MCI\_TOP to form the MCDATA to PCIF. The decoding of the address by the MCI\_TOP module ensures that only the targeted module within FDAS actively drives the data bus, with all other FDAS modules outputting zeros. Hence the data buses from the FDAS modules can be bit-wise OR’ed to provide a single data bus to the PCIF module.

# Place in the System

One example of the MCI\_TOP module’s place in the system is shown highlighted in the figure below:-

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 #2**

ADDR

DATA[511:0]

DATA[511:0]

ADDR

ADDR

DATA[511:0]

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

*Note: CLD, CONV and HSUM are designed with generically sized data width interfaces to DDRIF2 for a future implementation, but in this implementation the DDRIF2 data width is fixed. CLD, CONV and HSUM are also designed with paging of the DDR memory for a future implementation.*

**FDAS\_DDR\_CONTROLLER\_**

**CALIBRATION (INTEL IP) #1**

**FDAS\_DDR\_CONTROLLER\_**

**CALIBRATION (INTEL IP) #2**

DMA Transfer

**External 4 Gibi-Byte**

**DDR4 SDRAM Memory #1**

**External 4 Gibi-Byte**

**DDR4 SDRAM Memory #2**

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 ‑ – MCI\_TOP Location in FDAS

# Functional Specification

The MCI\_TOP module architecture is shown in the figure below:-

PCIF

Module

MCADDR\_PCIF[21:0]

**MCI\_TOP**

MCMS\_CTRL

MCCS\_PCIF

MCMS\_CONV

MCMS\_HSUM

FDAS Core

**MCI\_TOP\_**

**DATAOR**

MCDATAOUT\_CTRL[31:0]

MCDATAOUT\_CONV[31:0]

MCDATAOUT\_HSUM[31:0]

MCDATAOUT\_PCIF[31:0]

MCRWN\_PCIF

**MCI\_TOPMCI**

**(Inventory and DDR Monitor/ Control)**

PRODUCT\_ID[15:0]

TOP\_VERSION [15:0]

TOP\_REVISION [15:0]

MCMS\_TOPMCI

Hard

coded values

CLK\_MC

RST\_MC\_N

Bus Ripper

[19:0]

MCDATAIN\_PCIF[31:0]

DDR\_0\_CAL\_FAIL

DDR\_0\_CAL\_PASS

CTRL\_RESETN

DDR\_1\_CAL\_FAIL

DDR\_1\_CAL\_PASS

HSUM\_RESETN

CLD\_RESETN

CONV\_RESETN

DDRIF\_3\_RESETN

DDRIF\_2\_RESETN

DDR\_0\_RESETN

DDRIF\_PCIE\_RESETN

RESETN

DDR\_1\_RESETN

DDR\_1\_RESET\_DONE

DDR\_0\_RESET\_DONE

MCMS\_MSIX

**MCI\_TOP**

**\_DECODEMCI**

**(Address Decode)**

MCDATAOUT\_MSIX[31:0]

MSIX\_RESETN

CORE\_VERSION [15:0]

CORE\_REVISION [15:0]

DDR\_2\_CAL\_FAIL

DDR\_2\_CAL\_PASS

DDR\_3\_CAL\_FAIL

DDR\_3\_CAL\_PASS

DDR\_2\_RESET\_DONE

DDR\_3\_RESET\_DONE

DDRIF\_0\_RESETN

DDRIF\_1\_RESETN

DDR\_2\_RESETN

DDR\_3\_RESETN

**MCI\_TOP\_RESET**

Figure ‑ : MCI\_TOP Architecture Block Diagram

## MCI\_TOP\_DECODEMCI Sub-Module

Bits [21:0] of the address from PCIF shall connect to [21:0] of the address to the MCI\_TOP module, since the address from PCIF is for word data and the MCI Interface uses 32-bit data words.

The MCI\_TOP\_DECODEMCI sub-module shall decode the address bits to determine which internal module (MCI\_TOPMCI, CTRL, MSIX, CONV or HSUM) to select by controlling the modules select signals (MCMS\_TOPMCI, MCMS\_CTRL, MCMS\_MSIX, MCMS\_CONV and MCMS\_HSUM).

The address decode shall only occur whilst MCCS\_PCIF from the PCIF module is asserted (active high).

The address decode is shown in the table below:-

|  |  |  |  |
| --- | --- | --- | --- |
| Module | Size  /Hex | MCADDR\_PCIF[21:0] | |
| Base Address  (Hex) | End Address  (Hex) |
| TOPMCI | 0100000 | 00,0000 | 0F,FFFF |
| CTRL | 0080000 | 10,0000 | 17,FFFF |
| MSIX | 0000010 | 18,0000 | 18,000F |
| CONV | 0100000 | 20,0000 | 2F,FFFF |
| HSUM | 0040000 | 30,0000 | 33,FFFF |

Table ‑ : FDAS MC Address Decode

When a module is selected the appropriate MCMS\_\* signal shall be asserted (active high).

The address decoding shall use combinational logic without any retime stages.

## MCI\_TOPMCI Sub-Module

The MCI\_TOPMCI sub-module provides:-

* The read only inventory for the FDAS device via hard-coded values applied to PRODUCT\_ID[15:0], VERSION\_NUMBER[15:0] and REVISION\_NUMBER[15:0] pins of the MCI\_TOP module. The intention is that these fields shall be set via top level generics of the FDAS design prior to the design being compiled. This information allows the particular build to be identified. This is a clocked process requiring CLK\_MC (up to 200MHz) and RST\_MC\_N (active low).
* The ability to monitor the Calibration pass/fail signals from the DDR\_CONTROLLER modules
* The ability to manually reset the following:-
  + **CTRL\_RESETN**: CTRL Module CLK\_SYS clock domain circuits
  + **MSIX\_RESETN**: MSIX Module CLK\_SYS clock domain circuits
  + **CLD\_RESETN**: CLD Module CLK\_SYS clock domain circuits
  + **CONV\_RESETN**: CONV Module CLK\_SYS clock domain circuits
  + **HSUM\_RESETN**: HSUM Module CLK\_SYS clock domain circuits
  + **DDRIF\_0\_RESETN**: DDRIF2 #0 Module CLK\_SYS clock domain circuits
  + **DDRIF\_1\_RESETN**: DDRIF2 #1 Module CLK\_SYS clock domain circuits
  + **DDRIF\_2\_RESETN**: DDRIF2 #2 Module CLK\_SYS clock domain circuits
  + **DDRIF\_3\_RESETN**: DDRIF2 #3 Module CLK\_SYS clock domain circuits
  + **DDRIF\_PCIE\_RESETN**: DDRIF2#0, DDRIF2#1 DDRIF2#2 & DDRIF#3 Module CLK\_PCIE clock domain circuits
  + **DDR\_0\_RESETN**: DDR\_CONTROLLER #0
  + **DDR\_1\_RESETN**: DDR\_CONTROLLER #1
  + **DDR\_2\_RESETN**: DDR\_CONTROLLER #2
  + **DDR\_3\_RESETN**: DDR\_CONTROLLER #3

Reset Procedure:-

1. Apply all resets in the MCI\_TOP module.
2. Remove the DDR\_0\_RESETN, DDR\_1\_RESETN, DDR\_2\_RESETN and DDR\_3\_RESETN in the MCI\_TOP module.
3. Wait for DDR\_0\_RESET\_DONE, DDR\_1\_RESET\_DONE, DDR\_2\_RESET\_DONE, DDR\_3\_RESET\_DONE, DDR\_0\_CAL\_PASS, DDR\_1\_CAL\_PASS, DDR\_2\_CAL\_PASS and DDR\_3\_CAL\_PASS to go high
4. Remove remaining resets in the MCI\_TOP module.

## MCI\_TOP\_DATAOR Sub-Module

The MCI\_TOP\_DATAOR sub-module shall perform a bitwise logical “OR” of the Inventory data from the MCI\_TOPMCI sub-module, MCDATAOUT\_CTRL[31:0], MCDATAOUT\_MSIX[31:0], MCDATAOUT\_CONV[31:0] and MCDATAOUT\_HSUM[31:0].

The OR’ed result shall be passed to the PCIF module as MCDATAOUT\_PCIF[31:0].

The OR’ing shall use combinational logic without any retime stages.

## MCI\_TOP\_RESET Sub-Module

The MCI\_TOP\_RESET sub-module shall invert and logically AND each of the individual module resets from the MCI\_TOPMCI sub-module with the main system reset “RESETN”. This shall allow either the main system reset or the MC configured reset to action a reset to each processing module within FDAS.

# Design Requirement Tags

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

| **Design Tag** | **Description** | **Comment** |
| --- | --- | --- |
| FDAS.MC:010/A | FDAS shall be configured, monitored and controlled by a host computer via a Monitor and Control (MC) interface. This interface shall be little-endian, with byte assignments within a 32-bit word shown below for an IEEE 754 value as an example | 32-bit data Bus |
| FDAS.MC:020/A | The MC Interface shall be supported by the PCIe interface to the FPGA | MCI\_TOP connects to PCIF |
| FDAS.MC:030/A | The MC register map for the FDAS | Define Address Ranges for Modules. |
| FDAS.DIAGNOSTIC:032/A | Each of the processing modules within FDAS shall have an MC configurable asynchronous reset. The individual resets are for the following modules:-   * CTRL * CLD * CONV * HSUM * DDRIF2#1 * DDRIF2#2 * DDR Controller #1 (Altera IP) * DDR Controller#2 (Altera IP)   This reset shall be ANDed with the main system reset from a pin on the FPGA. | This allows control of resets via Software for diagnostic purposes. |
| FDAS.DIAGNOSTIC:035/A | The DDR Controllers shall be monitored to check the calibration of the interface to the external DDR SDRAM after a reset to the DDR Controllers. | The DDR Controller outputs a “Calibration Pass” and “Calibration Fail” signal. After a reset these signals are both set to logic 0, and after calibration either the “Calibration Pass” signal or “Calibration Fail” signal is asserted. |

Table ‑ : MCI\_TOP Design Requirement Tags

# Interface Specification

| Signal | Direction | Clock Domain | Description |
| --- | --- | --- | --- |
| **Interface to PCIF Module** |  |  |  |
| MCADDR\_PCIF[21:0] | IN | CLK\_MC | Micro Configuration Address from the PCIF module.  [21] = msb. |
| MCCS\_PCIF | IN | CLK\_MC | Micro Configuration Chip Select from the PCIF module.  ‘1’ = select. |
| MCRWN\_PCIF | IN | CLK\_MC | Micro Read not Write Signal from the PCIF module.  ‘0’ = Write  A read access from MCI\_TOP does not care about the state of MCRWN. The PCIF module generates MCRWN and controls the direction of the data bus to the PCIe Hard IP Macro. |
| MCDATAOUT\_PCIF[31:0] | IN | CLK\_MC | Micro Configuration Data from the PCIF module.  [31] = msb |
| MCDATAOUT\_PCIF[31:0] | OUT | CLK\_MC | Logical OR of data from CTRL, CONV and HSUM modules to the PCIF module.  [31] = msb |
|  |  |  |  |
| **MC Interface to CTRL, MSIX, CONV and HSUM Modules** |  |  |  |
| MCMS\_CTRL | OUT | CLK\_MC | Micro Configuration Module Select to the CTRL module.  ‘1’ = CTRL selected. |
| MCMS\_MSIX | OUT | CLK\_MC | Micro Configuration Module Select to the MSIX module.  ‘1’ = HSUM selected. |
| MCMS\_CONV | OUT | CLK\_MC | Micro Configuration Module Select to the CONV module.  ‘1’ = CONV selected. |
| MCMS\_HSUM | OUT | CLK\_MC | Micro Configuration Module Select to the HSUM module.  ‘1’ = HSUM selected. |
| MCDATAOUT\_CTRL[31:0] | IN | CLK\_MC | Micro Data from the CTRL module.  [31] = msb. |
| MCDATAOUT\_MSIX[31:0] | IN | CLK\_MC | Micro Data from the MSIX module.  [31] = msb. |
| MCDATAOUT\_CONV[31:0] | IN | CLK\_MC | Micro Data from the CONV module.  [31] = msb. |
| MCDATAOUT\_HSUM[31:0] | IN | CLK\_MC | Micro Data from the HSUM module.  [31] = msb. |
|  |  |  |  |
| **Manual Reset Control** |  |  |  |
| RESETN | IN | CLK\_SYS | Main System Reset.  ‘0’ = Reset |
| CTRL\_RESETN | OUT | CLK\_MC | Manual Reset to the CTRL module.  ‘0’ = Reset |
| MSIX\_RESETN | OUT | CLK\_MC | Manual Reset to the MSIX module.  ‘0’ = Reset |
| CLD\_RESETN | OUT | CLK\_MC | Manual Reset to the CLD module.  ‘0’ = Reset |
| CONV\_RESETN | OUT | CLK\_MC | Manual Reset to the CONV module.  ‘0’ = Reset |
| HSUM\_RESETN | OUT | CLK\_MC | Manual Reset to the HSUM module.  ‘0’ = Reset |
| DDRIF\_0\_RESETN | OUT | CLK\_MC | Manual Reset to the DDRIF2 #0 module  ‘0’ = Reset |
| DDRIF\_1\_RESETN | OUT | CLK\_MC | Manual Reset to the DDRIF2 #1 module  ‘0’ = Reset |
| DDRIF\_2\_RESETN | OUT | CLK\_MC | Manual Reset to the DDRIF2 #2 module  ‘0’ = Reset |
| DDRIF\_3\_RESETN | OUT | CLK\_MC | Manual Reset to the DDRIF2 #3 module  ‘0’ = Reset |
| DDR\_0\_RESETN | OUT | CLK\_MC | Manual Reset to the DDR\_CONTROLLER #0  ‘0’ = Reset |
| DDR\_1\_RESETN | OUT | CLK\_MC | Manual Reset to the DDR\_CONTROLLER #1  ‘0’ = Reset |
| DDR\_2\_RESETN | OUT | CLK\_MC | Manual Reset to the DDR\_CONTROLLER #2  ‘0’ = Reset |
| DDR\_3\_RESETN | OUT | CLK\_MC | Manual Reset to the DDR\_CONTROLLER #3  ‘0’ = Reset |
| DDRIF\_PCIE\_RESETN | OUT | CLK\_MC | Manual Reset to the DDRIF2 Module circuits on the CLK\_PCIE clock domain.  ‘0’ = Reset |
|  |  |  |  |
| **DDR\_CONTROLLER Monitor Signals** |  |  |  |
| DDR\_0\_CAL\_FAIL | IN | CLK\_DDR | DDR\_CONTROLLER #0 has failed calibration.  1 = Fail |
| DDR\_0\_CAL\_PASS | IN | CLK\_DDR | DDR\_CONTROLLER #0 has passed calibration.  1 = Pass |
| DDR\_1\_CAL\_FAIL | IN | CLK\_DDR | DDR\_CONTROLLER #1 has failed calibration.  1 = Fail |
| DDR\_1\_CAL\_PASS | IN | CLK\_DDR | DDR\_CONTROLLER #1 has passed calibration.  1 = Pass |
| DDR\_2\_CAL\_FAIL | IN | CLK\_DDR | DDR\_CONTROLLER #2 has failed calibration.  1 = Fail |
| DDR\_2\_CAL\_PASS | IN | CLK\_DDR | DDR\_CONTROLLER #2 has passed calibration.  1 = Pass |
| DDR\_3\_CAL\_FAIL | IN | CLK\_DDR | DDR\_CONTROLLER #3 has failed calibration.  1 = Fail |
| DDR\_3\_CAL\_PASS | IN | CLK\_DDR | DDR\_CONTROLLER #3 has passed calibration.  1 = Pass |
| DDR\_0\_RESET\_DONE | IN | CLK\_DDR | Indication that DDR Controller #0 has completed its reset.  1 = Reset Done |
| DDR\_1\_RESET\_DONE | IN | CLK\_DDR | Indication that DDR Controller #1 has completed its reset.  1 = Reset Done |
| DDR\_2\_RESET\_DONE | IN | CLK\_DDR | Indication that DDR Controller #2 has completed its reset.  1 = Reset Done |
| DDR\_3\_RESET\_DONE | IN | CLK\_DDR | Indication that DDR Controller #3 has completed its reset.  1 = Reset Done |
|  |  |  |  |
| **Inventory Interface** |  |  |  |
| PRODUCT\_ID[15:0] | IN | - | Product ID. Hardcoded value. |
| CORE\_VERSION [15:0] | IN | - | Core Version Number. Hardcoded value. |
| CORE\_REVISION [15:0] | IN | - | Core Revision Number. Hardcoded value. |
| TOP\_VERSION [15:0] | IN | - | Top Version Number. Hardcoded value. |
| TOP\_REVISION [15:0] | IN | - | Top Revision Number. Hardcoded value. |
|  |  |  |  |
| **Clocks & Resets** |  |  |  |
| CLK\_MC | IN | - | Micro Configuration Clock |
| RST\_MC\_N | IN | - | Asynchronous Reset.  Active Low. |

Table ‑ : MCI\_TOP Pinlist

# MCI Memory Mapped Interface

Note the addresses in the Memory Map are byte based.

## Memory map

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instance | Module | Size | Base Address | End Address | Mapped To |
| TOPMCI\_1 | MCI\_TOP | 0x400000 | 0x000000 | 0x3FFFFF |  |
| CTRL\_1 | CTRL | 0x200000 | 0x400000 | 0x5FFFFF |  |
| MSIX\_1 | MSIX | 0x000040 | 0x600000 | 0x60003F |  |
| CONV\_1 | CONV | 0x400000 | 0x800000 | 0xBFFFFF |  |
| HSUM\_1 | HSUM | 0x100000 | 0xC00000 | 0xCFFFFF |  |

## Generic values

|  |  |  |  |
| --- | --- | --- | --- |
| Instance | Generic | Value | Default |
| HSUM\_1 | summer\_g | 1 | 1 |

FDAS Address Decode

## Module MCI\_TOP

Top Level MCI

### Inventory

Register: PRODUCT\_ID

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 0x000000 | RO | PRODUCT\_ID[15:8] | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PRODUCT\_ID[7:0] | | | | | | | |

Register: CORE\_VERSION

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 0x000004 | RO | CORE\_VERSION[15:8] | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CORE\_VERSION[7:0] | | | | | | | |

Register: CORE\_REVISION

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 0x000008 | RO | CORE\_REVISION[15:8] | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CORE\_REVISION[7:0] | | | | | | | |

Register: TOP\_VERSION

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 0x00000C | RO | TOP\_VERSION[15:8] | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TOP\_VERSION[7:0] | | | | | | | |

Register: TOP\_REVISION

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 0x000010 | RO | TOP\_REVISION[15:8] | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TOP\_REVISION[7:0] | | | | | | | |

PRODUCT\_ID[15:0]: FDAS FPGA Product Number. Hard Coded Value.

CORE\_VERSION[15:0]: FDAS FPGA Core level version number. Hard Coded Value.

CORE\_REVISION[15:0]: FDAS FPGA Core level revision number. Hard Coded Value.

TOP\_VERSION[15:0]: FDAS FPGA Top level version number. Hard Coded Value.

TOP\_REVISION[15:0]: FDAS FPGA Top level revision number. Hard Coded Value.

### Monitor and Reset Control

Register: CTRL\_RESET (default: 0x00000000)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 0x000040 | RW |  |  | MSIX\_RESET | DDR\_3\_RESET | DDR\_2\_RESET | DDR\_1\_RESET | DDR\_0\_RESET | DDRIF\_PCIE\_RESET |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DDRIF\_3\_RESET | DDRIF\_2\_RESET | DDRIF\_1\_RESET | DDRIF\_0\_RESET | HSUM\_RESET | CONV\_RESET | CLD\_RESET | CTRL\_RESET |

Register: DDR\_0\_CAL\_FAIL

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 0x000044 | RO |  |  |  |  | DDR\_3\_RESET\_DONE | DDR\_2\_RESET\_DONE | DDR\_1\_RESET\_DONE | DDR\_0\_RESET\_DONE |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DDR\_3\_CAL\_PASS | DDR\_3\_CAL\_FAIL | DDR\_2\_CAL\_PASS | DDR\_2\_CAL\_FAIL | DDR\_1\_CAL\_PASS | DDR\_1\_CAL\_FAIL | DDR\_0\_CAL\_PASS | DDR\_0\_CAL\_FAIL |

CTRL\_RESET: Reset CTRL Module  
Logic '1' = Reset Default: 0.

CLD\_RESET: Reset CLD Module  
Logic '1' = Reset Default: 0.

CONV\_RESET: Reset CONV Module  
Logic '1' = Reset Default: 0.

HSUM\_RESET: Reset HSUM Module  
Logic '1' = Reset Default: 0.

DDRIF\_0\_RESET: Reset DDRIF2 Module #0 system clock domain and DDR clock domain  
Logic '1' = Reset Default: 0.

DDRIF\_1\_RESET: Reset DDRIF2 Module #1 system clock domain and DDR clock domain  
Logic '1' = Reset Default: 0.

DDRIF\_2\_RESET: Reset DDRIF2 Module #2 system clock domain and DDR clock domain  
Logic '1' = Reset Default: 0.

DDRIF\_3\_RESET: Reset DDRIF2 Module #3 system clock domain and DD clock domain  
Logic '1' = Reset Default: 0.

DDRIF\_PCIE\_RESET: Resets DDRIF2 Modules #0, #1, #2 & #3 PCIE clock domain, and also the MSIX module PCIE clock domain  
Logic '1' = Reset Default: 0.

DDR\_0\_RESET: Reset DDR Controller#0  
Logic '1' = Reset Default: 0.

DDR\_1\_RESET: Reset DDR Controller#1   
Logic '1' = Reset Default: 0.

DDR\_2\_RESET: Reset DDR Controller#2   
Logic '1' = Reset Default: 0.

DDR\_3\_RESET: Reset DDR Controller#3  
Logic '1' = Reset Default: 0.

MSIX\_RESET: Reset MSIX Module system clock domain  
Logic '1' = Reset Default: 0.

DDR\_0\_CAL\_FAIL: DDR Controller #0 has failed Calibration  
Logic '1' = Fail

DDR\_0\_CAL\_PASS: DDR Controller #0 has passed Calibration  
Logic '1' = Pass

DDR\_1\_CAL\_FAIL: DDR Controller #1 has failed Calibration   
Logic '1' = Fail

DDR\_1\_CAL\_PASS: DDR Controller #1 has passed Calibration   
Logic '1' = Pass

DDR\_2\_CAL\_FAIL: DDR Controller #2 has failed Calibration  
Logic '1' = Fail

DDR\_2\_CAL\_PASS: DDR Controller #2 has passed Calibration  
Logic '1' = Pass

DDR\_3\_CAL\_FAIL: DDR Controller #3 has failed Calibration  
Logic '1' = Fail

DDR\_3\_CAL\_PASS: DDR Controller #3 has passed Calibration  
Logic '1' = Pass

DDR\_0\_RESET\_DONE: DDR Controller #0 has reset  
Logic '1' = Reset Done

DDR\_1\_RESET\_DONE: DDR Controller #1 has reset  
Logic '1' = Reset Done

DDR\_2\_RESET\_DONE: DDR Controller #2 has reset  
Logic '1' = Reset Done

DDR\_3\_RESET\_DONE: DDR Controller #3 has reset  
Logic '1' = Reset Done

Reset procedure:  
1) Apply all resets.  
2) Remove DDR\_0\_RESETN, DDR\_1\_RESETN, DDR\_2\_RESETN and DDR\_3\_RESETN  
3) Wait for DDR\_0\_RESET\_DONE, DDR\_1\_RESET\_DONE, DDR\_2\_RESET\_DONE, DDR\_3\_RESET\_DONE, DDR\_0\_CAL\_PASS, DDR\_1\_CAL\_PASS, DDR\_2\_CAL\_PASS and   
DDR\_3\_CAL\_PASS to go high  
4) Remove remaining resets.

# Design Parameterisation

The MCI\_TOP module does not have any parameterised generics.

# Abbreviations and Acronyms

|  |  |
| --- | --- |
| DMA | Direct Memory Access |
| DDR | Double Data Rate |
| FDAS | Fourier Domain Acceleration Search |
| 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 |