

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

FDAS\_CTRL\_DS Revision 2 Draft B

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 CTRL for the Intel Agilex implementation of FDAS with 19-bit address to allow address decode space for the space for the MSIX module. |
| Issue 3 Draft B | 13/06/2023 | Added Counts for the CONV and HSUM DDR Accesses |

**Table of Contents**

1 Introduction 4

2 Place in the System 5

3 Functional Specification 6

3.1 Control MC Interface: CTRLMCI Sub-Module 7

3.1.1 Write to CTRLMCI 7

3.1.2 Read from CTRLMCI 8

3.2 Control Function: CTRL\_FUNC Sub-Module 10

3.2.1 Automatic Processing 10

3.2.2 Manual Processing 10

3.2.3 Processing Timers 11

3.2.4 CONV and HSUM DDR SDRAM Access Counters 11

4 Design Requirement Tags 12

5 Interface Specification 14

6 MCI Memory Mapped Interface 17

6.1 DM Trigger 17

6.2 Page Selection 17

6.3 Convolution Overlap 17

6.4 FOP Dimensions 18

6.5 Manual Override 19

6.6 Manual Trigger 19

6.7 Manual Enable 19

6.8 Manual Pause Enable 20

6.9 Manual Pause Reset 20

6.10 Manual Pause Time 21

6.11 Done Indications 22

6.12 Pause Indications 22

6.13 CONV FFT Ready Indication 23

6.14 Processing Times 23

6.15 CONV and HSUM DDR SDRAM Access Counts 24

7 Design Parameterisation 26

8 Abbreviations and Acronyms 27

# Introduction

This document captures the requirements for the CTRL Module used in the FDAS FPGA.

The CTRL Module is required to control the CLD, CONV and HSUM processing. Under normal operation CTRL shall receive an indication from the PC/Computer via the PCIe that a new set of samples are available to be processed. CTRL shall then automatically enable the CLD, CONV and HSUM modules at the appropriate times to carry out the necessary processing.

As a diagnostic aid CTRL can be configured by the PC/Computer via the PCIe to operate in a “Manual” mode, where each processing module only processes when enabled by an instruction from the PC/Computer via the PCIe, i.e. under “Manual” mode MC configuration data may be written into the memory mapped interface for CTRL (via the PCIe interface) to selectively control the functions within FDAS. In Manual mode each Processing module (i.e. CLD, CONV and HSUM) can be individually triggered, enabled and paused after an individually configurable time.

This version of the CTRL module has the address reduced from 20 bits to 19 bits to provide address decode space for the MSIX module. This is acceptable as the CTRL module memory map does not use the address bit that has been removed.

# Place in the System

The CTRL 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 ‑ – CTRL Location in FDAS

# Functional Specification

The CTRL Module architecture is shown in the figure below:-

MCI\_TOP

Module

CTRLMCI

CLD\_TRIGGER

CLD\_PAGE[31:0]

**CTRL**

CLD\_DONE

CLD\_ENABLE

**CTRL\_FUNC**

DM\_TRIG

CLK\_SYS

RST\_SYS\_N

MCADDR[18:0]

MCRWN

MCDATAIN[31:0]

MCDATAOUT[31:0]

CONV\_TRIGGER

CONV\_PAGE[31:0]

CONV\_ENABLE

HSUM\_TRIGGER

HSUM\_PAGE[31:0]

HSUM\_ENABLE

CONV\_DONE

HSUM\_DONE

PAGE

MAN\_OVERRIDE

MAN\_CLD\_TRIG

MAN\_CONV\_TRIG

MAN\_HSUM\_TRIG

MAN\_CLD\_EN

MAN\_CONV\_EN

MAN\_HSUM\_EN

MAN\_CLD\_PAUSE\_RST

MAN\_CONV\_PAUSE\_RST

MAN\_HSUM\_PAUSE\_RST

MAN\_CLD\_PAUSE\_CNT[31:0]

MAN\_CONV\_PAUSE\_CNT[31:0]

MAN\_HSUM\_PAUSE\_CNT[31:0]

LATCHED\_CLD\_DONE

LATCHED\_CONV\_DONE

LATCHED\_HSUM\_DONE

CONV

Module

CLD

Module

HSUM

Module

MCMS

MAN\_CLD\_PAUSE\_EN

MAN\_CONV\_PAUSE\_EN

MAN\_HSUM\_PAUSE\_EN

CLD\_PAUSED

CONV\_PAUSED

HSUM\_PAUSED

OVERLAP\_SIZE[9:0]

FOP\_SAMPLE\_NUM[22:0]

IFFT\_LOOP\_NUM[5:0]

CLK\_MC

RST\_MC\_N

CONV\_FFT\_READY

CLD\_PROC\_TIME[31:0]

CONV\_PROC\_TIME[31:0]

HSUM\_PROC\_TIME[31:0]

CONV\_WR\_EN

CONV\_WAITREQ

HSUM\_RD\_EN

HSUM\_WAITREQ

HSUM\_VALID

HSUM\_REQ\_CNT[31:0]

HSUM\_REC\_CNT[31:0]

CONV\_REQ\_CNT[31:0]

Figure ‑ : CTRL Architecture Block Diagram

## Control MC Interface: CTRLMCI Sub-Module

The CTRLMCI sub-module shall operate at the CLK\_MC domain with transfer of signals to the CLK\_SYS frequency of 200MHz.

The CTRLMCI sub-module shall connect to the MCI\_TOP module to support the writing of configuration information into the CTRLMCI sub-module by the PC/Computer via the PCIe interface (via MCDATAIN[31:0]), and read-back of configuration and status information from the CTRLMCI sub-module to the PC/Computer via the PCIe interface (via MCDATAOUT[31:0]).

The configuration information shall be stored in D-type registers in the CTRLMCI sub-module with the configuration values being made available to the CTRL\_FUNC sub-module. The status information from the CTRL\_FUNC sub-module shall be sampled by the CTRLMCI sub-module when it is read-back to the PC/Computer via the PCIe interface.

To minimise the address decoding within the CTRLMCI sub-module the MCI\_TOP module shall supply and MCMS signal (active high) to indicate that the CTRLMCI sub-module has been selected.

Back-to-back accesses without an intermediate “idle” cycle shall be supported.

### Write to CTRLMCI

When MCMS =1 and MCRWN = 0, if MCADDR[18:0] is a valid address of the memory map the value of MCDATAIN[31:0] shall be written. This operation shall be synchronous to CLK\_MC, with the D-type storage identified by MCADDR[18:0] being loaded with MCDATAIN[31:0] each clock cycle at which MCMS =1 and MCRWN = 0.

All internal stored values in memory shall be set to 0 if the asynchronous reset “RST\_MC\_N” is asserted (active low).

The figure below shows a write access with the minimum access time that is required to store the data.

Figure ‑ : CTRLMCI Write Access Timing Diagram showing the minimum access time

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | A |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | D |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | D |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

MCDATAIN[31:0]

MCADDR[18:0]

MCMS

Internal Storage at Location “A”

CLK\_MC

**DATAIN “D” stored at MCADDR “A” here, hence the access can complete on this cycle**

MCRWN

### Read from CTRLMCI

When MCMS =1, if MCADDR[18:0] is a valid address of the memory map the value of MCDATAOUT[31;0] shall be set to the value of the targeted location. This operation shall be synchronous to CLK\_MC, with the targeted location identified by MCADDR[18:0] driving MCDATAOUT[31:0] each clock cycle at which MCMS =1 and MCRWN = 0.

If MCMS = 0 or MCADDR[18:0] is not a valid address of the memory map MCDATAOUT[31:0] shall be set to 0x0000,0000

The figure below shows a Read access with the minimum access time that is required to read the data.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | |  |  | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | |  |  | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | |  |  | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | |  |  | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | Don’t care for Read Access | | | | | | | | |  |  |  |  |  |  |  |
|  |  |  |  | |  |  | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | A | |  |  | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | |  |  | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | D |  |  |  |  |  |  |  |  |
|  |  |  |  | |  |  | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | |  |  | |  |  |  |  |  |  |  |  |  |  |  |

MCDATAOUT[31:0]

MCADDR[18:0]

MCMS

CLK\_MC

**MCDATAOUT “D” from MCADDR “A” available here to be sampled.**

MCRWN

Figure ‑ : CTRLMCI Read Access Timing Diagram showing the minimum access time

## Control Function: CTRL\_FUNC Sub-Module

The CTRL\_FUNC sub-module shall operate at the CLK\_SYS frequency of 200MHz.

The CTRL\_FUNC sub-module shall control the processing of the CLD, CONV and HSUM modules based on instructions from the PC/Computer via the PCIe Interface.

### Automatic Processing

If MAN\_OVERRIDE configuration is not asserted (active high) the CTRL\_FUNC sub-module shall automatically control the processing of the CLD, CONV and HSUM modules when triggered by a rising edge of the DM\_TRIG configuration. The CTRL\_FUNC sub-module shall sequence the processing of the CLD, CONV and HSUM modules.

In this implementation the DM\_TRIG shall start the CLD / CONV module processing. When convolution is complete as indicated by the CONV\_DONE signal the HSUM module processing shall be automatically triggered.

A rising edge of the DM\_TRIG shall clear down the Latched “Done” signals (LATCHED\_CLD\_DONE, LATCHED\_CONV\_DONE and LATCHED\_HSUM\_DONE) to the CTRLMCI sub-module.

The Page of external DDR memory that should be used to process the page is set by the PAGE configuration from the PC/Computer via the PCIe interface. **Note that in this implementation the PAGE configuration has no effect, since there is only a single page.**

### Manual Processing

If MAN\_OVERRIDE configuration is asserted (active high) the CTRL\_FUNC sub-module shall manually control the processing of the CLD, CONV and HSUM modules via configuration from the PC/Computer via the PCIe interface. Each processing module shall have a manual enable configuration signal (MAN\_\*\_EN) that shall only allow processing (including trigger detection) if enabled (active high). The processing module shall halt at its current state if the Manual enable is de-asserted by the PC/Computer via the PCIe interface.

Each processing module shall also have a pause feature configurable by the PC/Computer via the PCIe (MAN\_\*\_PAUSE\_EN) to allow the processing module to run for a configurable number of clock cycles (MAN\_\*\_PAUSE\_CNT[31:0]) at which point it shall pause. The processing shall re-commence from the paused position when reset by the PC/Computer via the PCIe interface (MAN\_\*\_PAUSE\_EN), so that it can run for a further MAN\_\*\_PAUSE\_CNT[31:0] cycles. A \*\_PAUSED indication for each module shall indicate to the PC/Computer when the relevant module has paused.

Each processing module shall have individually configurable triggering in manual mode by the PC/Computer via the PCIe (MAN\_\*\_TRIG).

A rising edge of a MAN\_\*\_TRIG shall clear down the associated Latched “Done” signal (LATCHED\_\*\_DONE) to the CTRLMCI module.

The Page of external DDR memory that should be used to process the page is set by the PAGE configuration from the PC/Computer via the PCIe interface.

### Processing Timers

As a diagnostic aid timers shall be provided to measure the processing times of the CLD, CONV and HSUM modules. The processing time shall measure the time from when a module is triggered to when it has finished processing in terms of CLK\_SYS clock cycles.

The timers shall hold at the maximum value (0xFFFF,FFFF) and shall not roll-over.

### CONV and HSUM DDR SDRAM Access Counters

As a diagnostic aid timers shall be provided to count the number of DDR SDRAM accesses by the CONV and HSUM modules and the number of words returned to the HSUM module from the DDR SDRAMs

The counters are set to All 1’s after reset and are set to 0 when the associated module is triggered to perform processing of a DM. The counters are 32-bit and they will roll-over.

# Design Requirement Tags

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

| **Design Requirement Tag** | **Description** | **Comment** |
| --- | --- | --- |
| 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 | The Software commands shall instruct the CTRL module to commence processing a DM. The CTRL module shall ensure the CLD, CONV and HSUM modules operate in the correct sequence. |
| 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 32-bit little-endian. | The MC Data bus interface is 32-bit |
| 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 | CTRL Module within FDAS  Configuration:-  DM\_TRIG  PAGE  MAN\_OVERRIDE  MAN\_CLD\_TRIG  MAN\_CLD\_EN  MAN\_CLD\_PAUSE\_EN  MAN\_CLD\_PAUSE\_CNT[31:0]  MAU\_CLD\_PAUSE\_RST  MAN\_CONV\_TRIG  MAN\_CONV\_EN  MAN\_CONV\_PAUSE\_EN  MAN\_CONV\_PAUSE\_CNT[31:0]  MAN\_CONV\_PAUSE\_RST  MAN\_HSUM\_TRIG  MAN\_HSUM\_EN  MAN\_HSUM\_PAUSE\_EN  MAN\_HSUM\_PAUSE\_CNT[31:0]  MAN\_HSUM\_PAUSE\_RST  Status  LATCHED\_CLD\_DONE  LATCHED\_CONV\_DONE  LATCHED\_HSUM\_DONE |
| FDAS.DIAGNOSTIC:050/A | It shall be possible to trigger each main function within FDAS individually if desired via the MC interface. | This is a diagnostic aid to allow each function to be triggered individually to study the effect of the function. |
| FDAS.DIAGNOSTIC:060/A | It shall be possible to run each main function within FDAS for a configurable number of clock cycles, at which point processing will pause. Processing can be subsequently re-commenced for another configurable number of clock cycles. | This is a diagnostic aid to allow a function to run for a set period, thus allowing the function’s output to be observed before the end of a DM. |
| FDAS.DIAGNOSTIC:070/A | Processing Timers shall be provided for each of the main functions (CLD, CONV and HSUM), measuring the time from when a function is triggered to when it finishes processing. The time shall be measured in terms of system clock cycles. Each timer shall be 32-bits. The timer shall not roll-over if it reaches its maximum value (0xFFFF,FFFF). | This allows the processing time of FDAS to be accurately measured. |

Table ‑ : CTRL Design Requirement Tags

# Interface Specification

| Signal | Direction | Clock Domain | Description |
| --- | --- | --- | --- |
| **Interface to MCI\_TOP** |  |  |  |
| MCMS | IN | CLK\_MC | Module Select. When MCMS =1 the CTRL module is selected by the MCI\_TOP module. |
| MCADDR[18:0] | IN | CLK\_MC | Micro Configuration Address.  [18] = msb |
| MCRWN | IN | CLK\_MC | Micro Configuration Read Not Write.  ‘0’ = Write to CTRL.  A read access from CTRL 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. |
| MCDATAIN[31:0] | IN | CLK\_MC | Micro Configuration Data In from MCI\_TOP. This data will be written to address location MCADDR if MCCSN =0 and MCRWN = 0 .  [31] = msb |
| MCDATAOUT[31:0] | OUT | CLK\_MC | Micro Configuration Data Out to MCI\_TOP. This data will be valid from a valid MCADDR if MCMS =1  If MCMS = 0 or MCADDR is invalid then MCDATAOUT will be 0x00000000  [31] = msb |
| **CTRL Trigger/Enable/Page to CLD Module and Done Return** |  |  |  |
| CLD\_TRIGGER | OUT | CLK\_SYS | A 0 > 1 transition triggers the CLD module to commence processing a DM. |
| CLD\_ENABLE | OUT | CLK\_SYS | Enable the CLD module processing. The CLD module only processes when CLD\_ENABLE = 1. If CLD\_ENABLE is set to 0 the processing in CLD halts and only re-commences from its current position when CLD\_ENABLE is set to 1. |
| CLD\_PAGE[31:0] | OUT | CLK\_SYS | Indication of which page of external DDR SDRAM to read the freq-bin samples from. The value is the DDR SDRAM address for the start of the page. |
| CLD\_DONE | IN | CLK\_SYS | Indication that the CLD module has completed processing the DM. CLD\_DONE shall pulse for a single cycle when the processing has completed. |
|  |  |  |  |
| **CTRL Trigger/Enable/Page to CONV Module and Done Return** |  |  |  |
| CONV\_TRIGGER | OUT | CLK\_SYS | A 0 > 1 transition triggers the CONV module to commence processing a DM. |
| CONV\_ENABLE | OUT | CLK\_SYS | Enable the CONV module processing. The CONV module only processes when CONV\_ENABLE = 1. If CONV\_ENABLE is set to 0 the processing in CONV halts and only re-commences from its current position when CONV\_ENABLE is set to 1. |
| CONV\_PAGE[31:0] | OUT | CLK\_SYS | Indication of which page of external DDR SDRAM to write the FOP to. The value is the DDR SDRAM address for the start of the page. |
| CONV\_DONE | IN | CLK\_SYS | Indication that the CONV module has completed processing the DM. CONV\_DONE shall pulse for a single cycle when the processing has completed. |
| CONV\_FFT\_READY | IN | CLK\_SYS | Indication that the output of the first FFT in the CONV module is ready for diagnostic inspection after the enable to the CONV module has been de-asserted. |
|  |  |  |  |
| **CTRL Trigger/Enable/Page to HSUM Module and Done Return** |  |  |  |
| HSUM\_TRIGGER | OUT | CLK\_SYS | A 0 > 1 transition triggers the HSUM module to commence processing a DM. |
| HSUM\_ENABLE | OUT | CLK\_SYS | Enable the HSUM module processing. The HSUM module only processes when HSUM\_ENABLE = 1. If HSUM\_ENABLE is set to 0 the processing in HSUM halts and only re-commences from its current position when HSUM\_ENABLE is set to 1. |
| HSUM\_PAGE[31:0] | OUT | CLK\_SYS | Indication of which page of external DDR SDRAM to read the FOP from. The value is the DDR SDRAM address for the start of the page. |
| HSUM\_DONE | IN | CLK\_SYS | Indication that the HSUM module has completed processing the DM. HSUM\_DONE shall pulse for a single cycle when the processing has completed. |
|  |  |  |  |
| **CONV and HSUM DDR Access Counts** |  |  |  |
| CONV\_WR\_EN | IN | CLK\_SYS | Write Enable from the CONV module to the DDRIF2 modules |
| CONV\_WAITREQ | IN | CLK\_SYS | Wait Request from the DDRIF2 modules to the CONV module |
| HSUM\_RD\_EN | IN | CLK\_SYS | Read Enable from the HSUM module to the DDRIF2 modules |
| HSUM\_WAITREQ | IN | CLK\_SYS | Wait Request from the DDRIF2 modules to the HSUM module |
| HSUM\_VAILD | IN | CLK\_SYS | Valid from the DDRIF2 modules to the HSUM module indicatin there is a valid word |
|  |  |  |  |
| **FOP Configuration Interface** |  |  |  |
| FOP\_SAMPLE\_NUM[22:0] | OUT | CLK\_SYS | The number of samples for the FOP to the CLD module. This defines the “x-axis” (frequency) dimension of the FOP. |
| OVERLAP\_SIZE[9:0] | OUT | CLK\_SYS | The size of the overlap in the “Overlap-Save” convolution. |
| IFFT\_LOOP\_NUM[5:0] | OUT | CLK\_SYS | The number of times the CONV module loops round its instantiated IFFTs.  This defines the “y-axis” (frequency derivative) of the FOP.  Number of filters in FOP y-axis =  (IFFT\_LOOP\_NUM x ifft\_g x 2) + 1  Where ifft\_g is the generic applied to the CONV module to define the number if instantiated IFFTs. |
| **Global Clock/Resets** |  |  |  |
| CLK\_SYS | IN | - | 200MHz Core System Clock |
| RST\_SYS\_N | IN | - | Asynchronous Logic reset for the CLK\_SYS domain  ‘0’=Reset |
| CLK\_MC | IN | - | Micro Configuration Clock. 200MHz Max |
| RST\_MC\_N | IN | - | Asynchronous Logic reset for the CLK\_MC domain  ‘0’=Reset |

Table ‑ : CTRL Pinlist

# MCI Memory Mapped Interface

Note that the addresses in the Memory Map are byte based.

## DM Trigger

DM Trigger

Register: DM\_TRIG (default: 0x00000000)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0x000000 | RW |  |  |  |  |  |  |  | DM\_TRIG |

DM\_TRIG: A 0>1 transition of DM\_TRIG causes FDAS to process a DM. All FDAS modules are simultaneously triggered. Default: 0.

## Page Selection

Page Selection

Register: PAGE (default: 0x00000000)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0x000020 | RW |  |  |  |  |  |  |  | PAGE |

PAGE: Indication of which page the processing modules should use as the source of data.  
CLD shall use PAGE  
CONV shall use PAGE  
HSUM shall use NOT(PAGE) Default: 0.

## Convolution Overlap

Convolution Overlap

Register: OVERLAP\_SIZE (default: 0x00000000)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 0x000040 | RW |  |  |  |  |  |  | OVERLAP\_SIZE[9:8] | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OVERLAP\_SIZE[7:0] | | | | | | | |

OVERLAP\_SIZE[9:0]: Size of the Convolution Overlap  
"0000000000" = No overlap  
"0000000001" = Overlap of 1 sample  
etc Default: 0.

## FOP Dimensions

FOP Dimensions

Register: FOP\_SAMPLE\_NUM (default: 0x00000000)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 0x000060 | RW |  | FOP\_SAMPLE\_NUM[22:16] | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FOP\_SAMPLE\_NUM[15:8] | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FOP\_SAMPLE\_NUM[7:0] | | | | | | | |

Register: IFFT\_LOOP\_NUM (default: 0x00000000)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0x000068 | RW |  |  | IFFT\_LOOP\_NUM | | | | | |

FOP\_SAMPLE\_NUM[22:0]: The number of 64-bit samples of the DM to be processed to form the FOP.  
This determines the "frequency" dimension of the FOP (i.e. the x-axis).  
  
The number of samples processed is one more than the configuration value.  
  
FDAS processes an integral number of FFTs during the convolution process.   
  
To calculate the FOP\_SAMPLE\_NUMBER the size of the FFT and the OVERLAP\_SIZE configuration must be taken into account.  
  
For example if the FFT size is 1,024 points and the OVERLAP\_SIZE = 500, this means that 524 new samples are processed by each FFT. Hence if we wish to process 2,097,152 samples we will need to perform ROUNDUP(2097152/524) = 4003 FFTs, which equates to 4003 x 524 = 2,097,572 samples.  
  
Hence in this example the value entered into FOP\_SAMPLE\_NUMBER = 2,097,572 - 1 = 2,097,571 decimal = 0x2001A3 Default: 0.

IFFT\_LOOP\_NUM[5:0]: The number of times the CONV module loops through its Inverse Fourier Transforms.   
The number of Inverse Fourier Transforms (IFFTs) processed in each loop is set by the "ifft\_g" generic. Each IFFT generates two rows of the FOP (i.e. the y-axis), -one for positive acceleration "+[p]" and the compliment for negative acceleration -"[p]". The total number of rows of the FOP is given by:-  
  
Total FOP Rows = 1 + ifft\_g\* IFFT\_LOOP\_NUM  
  
Default: 0.

## Manual Override

Manual Override

Register: MAN\_OVERRIDE (default: 0x00000000)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0x000080 | RW |  |  |  |  |  |  |  | MAN\_OVERRIDE |

MAN\_OVERRIDE: 1 = Enable individual manual control of the triggering of the FDAS modules. Default: 0.

## Manual Trigger

Manual Trigger

Register: MAN\_CLD\_TRIG (default: 0x00000000)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0x0000A0 | RW |  |  |  |  |  | MAN\_HSUM\_TRIG | MAN\_CONV\_TRIG | MAN\_CLD\_TRIG |

MAN\_CLD\_TRIG: If MAN\_OVERRIDE = 1, then a 0>1 transition on MAN\_CLD\_TRIG shall trigger the CLD module to commence processing a DM. Default: 0.

MAN\_CONV\_TRIG: If MAN\_OVERRIDE = 1, then a 0>1 transition on MAN\_CONV\_TRIG shall trigger the CONV module to commence processing a DM. Default: 0.

MAN\_HSUM\_TRIG: If MAN\_OVERRIDE = 1, then a 0>1 transition on MAN\_HSUM\_TRIG shall trigger the HSUM module to commence processing a DM. Default: 0.

## Manual Enable

Manual Enable

Register: MAN\_CLD\_EN (default: 0x00000000)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0x0000C0 | RW |  |  |  |  |  | MAN\_HSUM\_EN | MAN\_CONV\_EN | MAN\_CLD\_EN |

MAN\_CLD\_EN: If MAN\_OVERRIDE = 1, then if MAN\_CLD\_EN =1 the CLD module operates, however if MAN\_CLD\_EN = 0 the CLD module halts at its current state. Default: 0.

MAN\_CONV\_EN: If MAN\_OVERRIDE = 1, then if MAN\_CONV\_EN =1 the CONV module operates, however if MAN\_CONV\_EN = 0 the CONV module halts at its current state. Default: 0.

MAN\_HSUM\_EN: If MAN\_OVERRIDE = 1, then if MAN\_HSUM\_EN =1 the HSUM module operates, however if MAN\_HSUM\_EN = 0 the HSUM module halts at its current state. Default: 0.

## Manual Pause Enable

Manual Pause Enable

Register: MAN\_CLD\_PAUSE\_EN (default: 0x00000000)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0x0000E0 | RW |  |  |  |  |  | MAN\_HSUM\_PAUSE\_EN | MAN\_CONV\_PAUSE\_EN | MAN\_CLD\_PAUSE\_EN |

MAN\_CLD\_PAUSE\_EN: If MAN\_OVERRIDE = 1, then if MAN\_CLD\_PAUSE\_EN =1 the CLD module automatic pause feature is enabled which will pause the CLD module after MAN\_CLD\_PAUSE\_CNT clock cycles. Toggling MAN\_CLD\_PAUSE\_RST from 0>1 shall recommence CLD operation to the next pause. Default: 0.

MAN\_CONV\_PAUSE\_EN: If MAN\_OVERRIDE = 1, then if MAN\_CONV\_PAUSE\_EN =1 the CONV module automatic pause feature is enabled which will pause the CONV module after MAN\_CONV\_PAUSE\_CNT clock cycles. Toggling MAN\_CONV\_PAUSE\_RST from 0>1 shall recommence CONV operation to the next pause. Default: 0.

MAN\_HSUM\_PAUSE\_EN: If MAN\_OVERRIDE = 1, then if MAN\_HSUM\_PAUSE\_EN =1 the HSUM module automatic pause feature is enabled which will pause the HSUM module after MAN\_HSUM\_PAUSE\_CNT clock cycles. Toggling MAN\_HSUM\_PAUSE\_RST from 0>1 shall recommence HSUM operation to the next pause. Default: 0.

## Manual Pause Reset

Manual Pause Reset

Register: MAN\_CLD\_PAUSE\_RST (default: 0x00000000)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0x000100 | RW |  |  |  |  |  | MAN\_HSUM\_PAUSE\_RST | MAN\_CONV\_PAUSE\_RST | MAN\_CLD\_PAUSE\_RST |

MAN\_CLD\_PAUSE\_RST: If MAN\_OVERRIDE = 1 a 0>1 transition of MAN\_CLD\_PAUSE\_RST shall recommence CLD operation if it is paused. Default: 0.

MAN\_CONV\_PAUSE\_RST: If MAN\_OVERRIDE = 1 a 0>1 transition of MAN\_CONV\_PAUSE\_RST shall recommence CONV operation if it is paused. Default: 0.

MAN\_HSUM\_PAUSE\_RST: If MAN\_OVERRIDE = 1 a 0>1 transition of MAN\_HSUM\_PAUSE\_RST shall recommence HSUM operation if it is paused. Default: 0.

## Manual Pause Time

Manual Pause Time

Register: MAN\_CLD\_PAUSE\_CNT (default: 0x00000000)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| 0x000120 | RW | MAN\_CLD\_PAUSE\_CNT[31:24] | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MAN\_CLD\_PAUSE\_CNT[23:16] | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MAN\_CLD\_PAUSE\_CNT[15:8] | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAN\_CLD\_PAUSE\_CNT[7:0] | | | | | | | |

Register: MAN\_CONV\_PAUSE\_CNT (default: 0x00000000)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| 0x000124 | RW | MAN\_CONV\_PAUSE\_CNT[31:24] | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MAN\_CONV\_PAUSE\_CNT[23:16] | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MAN\_CONV\_PAUSE\_CNT[15:8] | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAN\_CONV\_PAUSE\_CNT[7:0] | | | | | | | |

Register: MAN\_HSUM\_PAUSE\_CNT (default: 0x00000000)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| 0x000128 | RW | MAN\_HSUM\_PAUSE\_CNT[31:24] | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MAN\_HSUM\_PAUSE\_CNT[23:16] | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MAN\_HSUM\_PAUSE\_CNT[15:8] | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAN\_HSUM\_PAUSE\_CNT[7:0] | | | | | | | |

MAN\_CLD\_PAUSE\_CNT[31:0]: If MAN\_OVERRIDE = 1 and MAN\_CLD\_PAUSE\_EN =1 then the CLD module will pause after MAN\_CLD\_PAUSE\_CNT clock cycles. Default: 0.

MAN\_CONV\_PAUSE\_CNT[31:0]: If MAN\_OVERRIDE = 1 and MAN\_CONV\_PAUSE\_EN =1 then the CONV module will pause after MAN\_CONV\_PAUSE\_CNT clock cycles. Default: 0.

MAN\_HSUM\_PAUSE\_CNT[31:0]: If MAN\_OVERRIDE = 1 and MAN\_HSUM\_PAUSE\_EN =1 then the HSUM module will pause after MAN\_HSUM\_PAUSE\_CNT clock cycles. Default: 0.

## Done Indications

Done Indications

Register: LATCHED\_CLD\_DONE

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0x000140 | RO |  |  |  |  |  | LATCHED\_HSUM\_DONE | LATCHED\_CONV\_DONE | LATCHED\_CLD\_DONE |

LATCHED\_CLD\_DONE: Latched Indication that CLD has finished processing the DM. 1 = finished.

LATCHED\_CONV\_DONE: Latched Indication that CONV has finished processing the DM. 1 = finished.

LATCHED\_HSUM\_DONE: Latched Indication that HSUM has finished processing the DM. 1 = finished.

## Pause Indications

Pause Indications

Register: CLD\_PAUSED

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0x000160 | RO |  |  |  |  |  | HSUM\_PAUSED | CONV\_PAUSED | CLD\_PAUSED |

CLD\_PAUSED: '1' = Latched Indication CLD has paused. If MAN\_OVERRIDE = 1 a 0> 1 Transition of MAN\_CLD\_PAUSE\_RST shall clear CLD\_PAUSED.

CONV\_PAUSED: '1' = Latched Indication CONV has paused. If MAN\_OVERRIDE = 1 a 0> 1 Transition of MAN\_CONV\_PAUSE\_RST shall clear CONV\_PAUSED.

HSUM\_PAUSED: '1' = Latched Indication HSUM has paused. If MAN\_OVERRIDE = 1 a 0> 1 Transition of MAN\_HSUM\_PAUSE\_RST shall clear HSUM\_PAUSED.

## CONV FFT Ready Indication

CONV FFT Ready Indication

Register: CONV\_FFT\_READY

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0x000180 | RO |  |  |  |  |  |  |  | CONV\_FFT\_READY |

CONV\_FFT\_READY: '1' = Indication that the first FFT in the CONV module has completed its processing and is ready for the FFT output to be observed for diagnostic purposes. This signal is asserted when the enable to CONV has been de-asserted and the first FFT has completed its processing.

## Processing Times

Processing Times

Register: CLD\_PROC\_TIME

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| 0x000188 | RO | CLD\_PROC\_TIME[31:24] | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CLD\_PROC\_TIME[23:16] | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CLD\_PROC\_TIME[15:8] | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLD\_PROC\_TIME[7:0] | | | | | | | |

Register: CONV\_PROC\_TIME

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| 0x00018C | RO | CONV\_PROC\_TIME[31:24] | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CONV\_PROC\_TIME[23:16] | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CONV\_PROC\_TIME[15:8] | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CONV\_PROC\_TIME[7:0] | | | | | | | |

Register: HSUM\_PROC\_TIME

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| 0x000190 | RO | HSUM\_PROC\_TIME[31:24] | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| HSUM\_PROC\_TIME[23:16] | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| HSUM\_PROC\_TIME[15:8] | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSUM\_PROC\_TIME[7:0] | | | | | | | |

CLD\_PROC\_TIME[31:0]: CLD Processing Time.  
The time from the triggering of CLD to CLD\_DONE measured in CLK\_SYS cycles.

CONV\_PROC\_TIME[31:0]: CONV Processing Time.  
The time from the triggering of CONV to CONV\_DONE measured in CLK\_SYS cycles.

HSUM\_PROC\_TIME[31:0]: HSUM Processing Time.  
The time from the triggering of HSUM to HSUM\_DONE measured in CLK\_SYS cycles.

## CONV and HSUM DDR SDRAM Access Counts

Register: CONV\_REQ\_CNT

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| 0x0001A0 | RO | CONV\_REQ\_CNT[31:24] | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CONV\_REQ\_CNT[23:16] | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CONV\_REQ\_CNT[15:8] | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CONV\_REQ\_CNT[7:0] | | | | | | | |

Register: HSUM\_REQ\_CNT

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| 0x0001A4 | RO | HSUM\_REQ\_CNT[31:24] | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| HSUM\_REQ\_CNT[23:16] | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| HSUM\_REQ\_CNT[15:8] | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSUM\_REQ\_CNT[7:0] | | | | | | | |

Register: HSUM\_REC\_CNT

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Mode | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| 0x0001A8 | RO | HSUM\_REC\_CNT[31:24] | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| HSUM\_REC\_CNT[23:16] | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| HSUM\_REC\_CNT[15:8] | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSUM\_REC\_CNT[7:0] | | | | | | | |

CONV\_REQ\_CNT[31:0]: Number of CONV DDR SDRAM Requests

HSUM\_REQ\_CNT[31:0]: Number of HSUM DDR SDRAM Requests

HSUM\_REC\_CNT[31:0]: Number of HSUM DDR SDRAM Received Words

# Design Parameterisation

The CTRL module has no generic parameters:

# 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 |