Bristol University AstroFFT Project
FPGA FFT Processor
Driver Interface Specification

Alpha Data Parallel Systems

Revision: 1.11
Date: 17th September 2002

Repository Information

Project: AstroFFT

Revision History

Version	Date	Author	Description
1.1	25-06-2001	Tomas Whitlock	First issue.
1.2	02-07-2001	Tomas Whitlock	Incorporated feedback from BEAM on version 1.1: - Added FIFO underflow/overflow flags/interrupts - Added interrupts for FIFO levels - Added SUM_STAT2 register - Added ADC_AFL and SUM_AFL registers Added PLX9080 configuration information Added local bus frequency specification.
1.3	06-07-2001	Tomas Whitlock	Incorporated feedback from BEAM on version 1.2: - Simplified to present a single FIFO to the driver and removed superfluous registers from operational register section. - Clarified behaviour of modes. Invented notion of processing cycles and phases.
1.4	06-07-2001	Tomas Whitlock	Renamed to 'driver interface specification'. Rationalised PH_N so that it applies to all AstroFFT modes. Added section about data loss during mode change. Added IF_ID register and moved REV_ID and IF_ID to beginning of register map.
1.5	12-10-2001	Tomas Whitlock	Introduced command register. Clarified meaning of SUSP fields. Added section about write ordering and completion. Added section in driver notes about how to start and stop processing. Corrections to register fields: - PH_N.N reduced to 24 bits - MODE.SELECT now 7 bits & reset state now TBD - C_STAT.MODE reset state now 0 - INTSTAT bits now R/W1C - Gave most RAX/MBZ fields a determinate reset state. Removed item specifying that registers except FIFO_RD do not burst. Added note about prefetching to section 3. Removed FIFO_STAT.UDF/OVF fields as they are also in INTSTAT.

1.6	30-10-2001	Tomas Whitlock	Merged modes 3 and 4 as per current AstroFFT Specification (reference: SPEC7).
1.7	11-01-2002	Tomas Whitlock	Added C_ONCE and PH_ONCE to CMD register (allow one-shot of proc. phase/cycle). Added PH_ABT to CMD register (allow abort of proc. phase). Added MISCSTAT register. Polling now not required for RESET register. Renamed some registers and fields for better consistency and meaning.
1.8	15-01-2002	Tomas Whitlock	Deleted following fields because accurate FIFO level difficult to generate: • FIFOSTAT.LEVEL • FIFOSTAT.AFULL • FIFOSTAT.FULL • INTSTAT.AFULL, INTSTAT.FULL • INTMASK.AFULL, INTMASK.FULL Deleted FIFO_AFL register.
1.9	25-02-2002	Tomas Whitlock	Introduced CYC_N register, to allow a programmable number of processing cycles to be performed. Changed PHSTAT and CYCSTAT registers. Simplified CMD register to have only following fields: • RUNNING • START • ABORT • FLUSH Corrected typo in INTSTAT description : ”A bit in INTSTAT can transition to 1 only …” is now “A bit in INTSTAT can transition to 1 even …”. Added section about FIFO packet format. Added ADC_CTL register. Changed DMA channel used for AstroFFT from 0 to 1 because of PLX errata.
1.10	13-03-2002	Tomas Whitlock	Corrected CMD.ABORT description. Added OVR interrupt. Added MODE.RANGE field.
1.11	17-09-2002	Tomas Whitlock	Added ADC_CTL.NINTLV field. Updated section 6 to include data order for AstroFFT mode 2.

Document Approvals:

Version 1.1
Bill Blyth, Alpha Data, 26-06-2001

Version 1.2
Keith Baker, Alpha Data, 02-07-2001

Version 1.4
Terry Barnaby, BEAM, 11-07-2001

Table of contents

List of tables

1. Introduction

1.1. Purpose

This document is a register-level specification of the FPGA FFT processor in the AstroFFT system.

1.2. Terminology

1.2.1. Register field types

R/W	both readable and writeable
RO	read only; writes have no effect
WO	write only; reads return undefined data
R/W1C	readable and writing 1 clears a bit
R/W1S	readable and writing 1 sets a bit
R/W1M	readable and writing 1 momentarily sets a bit
RAX/MBZ	ignore value on reads and must be 0 on writes
RAX/MB1	ignore value on reads and must be 1 on writes

1.2.2. FFT processor terminology

Data FIFO	A buffer in the FFT processor that holds data for the driver to receive.
Driver	A component of software, running on the host CPU, that is responsible for controlling the FFT processor.
DWORD	A 32-bit (4 byte) integer.
Processing cycle	The unit of processing that results in a unit of data being produced for the driver to read.
Processing phase	A component of a processing cycle.
Sampler frontend	The subsystem that drives the sampler interface of the AstroFFT FPGA.

1.2.3. Mandatory and optional requirements

The word 'shall' indicates behaviour that is mandatory, whilst the word 'should' indicates behaviour that is optional but desirable. The word 'may' indicates behaviour that is optional and not necessarily desirable or undesirable.

1.3. Referenced documents

1. **SPEC7** - "Specification for Bristol University Radio Astronomy FFT Processor" – Beam Ltd., 10/8/2001

2. Overview

The FPGA FFT processor shall be responsible for performing a number of FFTs on sampled data input from an analogue to digital converter (ADC). A number of registers shall exist within the FPGA for the purpose of controlling the FFT processor.

The functionality of the AstroFFT system may be distributed over several different FPGA bitstreams. In order to simplify the driver, the register map defined in this document shall apply to all bitstreams.

2.1. Processing cycles and phases

The processing performed by the FFT processor is conceptually divided into *processing cycles*. A processing cycle is the unit of processing that results in a unit of data being placed in the *data FIFO* for the driver to read.

A processing cycle is divided into one or more *processing phases*. A processing phase is the smallest unit of processing that can be performed. In other words, the FFT processor shall permit processing to be started and stopped only at processing phase boundary.

In each of the AstroFFT modes, the number of processing phases per processing cycle and the data unit produced varies according to table 1 below:

AstroFFT mode	Name of mode	Processing phases per processing cycle	Data unit produced in data FIFO
0	FFT	Arbitrary	Block of integrated data
1	QFFT	Arbitrary	Block of integrated data
2	RFFT	Arbitrary	Block of integrated data
3	Analogue	1	Block of ADC samples plus packet containing number of clipped samples and RMS power.

Table 1. Number of processing phases in each AstroFFT mode

Thus:

- In AstroFFT modes 0, 1 and 2, each processing phase is an FFT. The result of each FFT is integrated over the course of a processing cycle.
- In AstroFFT mode 3, each processing phase places a block of ADC data into the data FIFO. Only one processing phase per processing cycle should be performed in order to prevent the data FIFO overflowing.

3. Register map

The FPGA FFT processor registers shall be accessible by the driver via space 0 of the PLX9080. The register map can be summarised as follows:

Name	Offset in space 0	Function
IF_ID	0x0	Interface version of FPGA
REV_ID	0x4	Internal revision level of FPGA
RESET	0x8	Reset of the FPGA
CMD	0xC	Starts/stops processing
CONTROL	0x10	Miscellaneous control
MODE	0x14	Sets operating mode
INTMASK	0x18	Sets interrupt mask
INTSTAT	0x1C	Interrupt status
	0x20	*Reserved for operational registers*
FIFOSTAT PH_N	0x24 0x28	Data FIFO status Sets number of processing phases per processing cycle
CYCSTAT	0x2C	Processing cycle status
PHSTAT	0x30	Processing phase status
MISCSTAT	0x34	Miscellaneous status

CYC_N	0x38	Number of processing cycles to perform
ADC_CTL	0x3C	Controls the sampler frontend
	0x40 – 0x7C	*Reserved for operational registers*
FIFO_RD	0x80	Returns integrated data
	0x84 – 0xFC	*Reserved for operational registers*
	0x100 – 0x3FC	*Reserved for debug registers*

Table 2. Register map of AstroFFT FPGA

The defined range in space 0 of the FPGA registers shall be locations 0x0 to 0x3FC. All unused locations within this range shall be ignored by software. Locations outside the range 0x0 to 0x3FC may contain aliases of the register map.

Locations 0x0 to 0xFC shall contain operational registers for use by the driver in normal operation.

Locations 0x100 to 0x3FC shall contain debug registers. The driver shall not rely on their presence in order to provide user accessible functionality.

Note that the PLX9080 should be configured to access these registers in burst mode. FIFO_RD shall be capable of sustaining burst throughput on the local bus.

Prefetching on the PCI bus or PLX local bus is undesirable when accessing the AstroFFT registers, since prefetching up to the FIFO_RD register will result in spurious reads of the data FIFO and hence loss of data. If prefetching occurs but is limited to a maximum of 0x40 bytes on from the original target address, data loss cannot occur.

3.1. IF_ID register (0x0)

The IF_ID register shall contain a code indicating to the driver the format of the remainder of the register map.

A new code shall be issued whenever a change is made to the bitstream that is not fully transparent, behaviourally and at a register level, to the driver.

Field	Bit range	Reset state	Type	Function
ID	31:0	TBD	RO	Interface version code

The ID field may be split into sub-fields in a future version of this document.

Table 3. IF_ID register fields

3.2. REV_ID register (0x4)

The REV_ID register shall contain an internal revision level of the bitstream. The REV_ID.REV field shall be incremented with each new release of the bitstream.

Field	Bit range	Reset state	Type	Function
REV	31:0	TBD	RO	Internal revision level of bitstream.

These fields may be split into sub-fields in a future version of this document.

Table 4. REV_ID register fields

3.3. RESET Register (0x8)

The RESET register shall provide a means of initialising the FFT processor to a known state.

Field	Bit range	Reset state	Type	Function
ALL	0	n/a	WO	Write 1 to reset the FPGA.
	31:1	0	RAX/MBZ	*Reserved*

Table 5. RESET register fields

Writing a 1 to ALL shall cause the entire FFT processor to undergo software reset, discarding state and returning all registers to their reset state. The FPGA can be reset in this way at any time. After the host writes 1 to RESET.ALL, the FPGA shall hold off the host for a short time while reset is being performed. This will ensure that the host sees reset as an atomic operation.

A software reset also sends a reset pulse to the sampler frontend.

3.4. CMD Register (0xC)

The CMD register shall provide a means of starting and stopping the FFT processor. Only one command shall be issued by the host per write to the CMD register. In other words, at most one bit shall be written with 1.

Field	Bit range	Reset state	Type	Function
RUNNING	0	0	RO	1 => Processing is in progress
	3:1	n/a	RAX/MBZ	*Reserved*
START	4	0	WO	Write 1 to begin processing.
	6:5	n/a	RAX/MBZ	*Reserved*
ABORT	7	0	WO	Write 1 to abort processing.
	11:8	0	RAX/MBZ	*Reserved*
FLUSH	12	0	WO	Flushes the data FIFO.
	31:13	0	RAX/MBZ	*Reserved*

Table 6. CMD register fields

The CMD.RUNNING field returns 1 if the FPGA is processing, otherwise 0. The driver can rely on consistency between INTSTAT and CMD.RUNNING. When the last processing cycle is completed, CMD.RUNNING, INTSTAT.CYCLE and INTSTAT.PHASE will transition at the same instant to 0, 1 and 1 respectively.

The CMD.START field should be written with 1 in order to begin processing. A number of processing cycles equal to CYC_N+1 will be performed. Each processing cycle will consist of PH_N+1 processing phases. Writing to 1 CMD.START will cause CMD.RUNNING to transition to 1.

The CMD.ABORT field should be written with 1 in order to abort processing. The current processing phase and processing cycles will be completed (abnormally) and the amount of data entered into the data FIFO will be unpredictable. Writing 1 to CMD.ABORT will cause CMD.RUNNING to transition to 0.

The CMD.FLUSH field should be written with 1 in order to discard any data currently in the data FIFO.

3.5. CONTROL register (0x10)

The CONTROL register provides miscellaneous control functions for the FFT processor.

Field	Bit range	Reset state	Type	Function
EN_DDMA	0	0	R/W	1 => Use demand-mode DMA.
	31:1	n/a	RAX/MBZ	*Reserved*

Table 7. CONTROL register fields

When the EN_DDMA bit is 1, the FFT processor shall be allowed to use demand-mode DMA for transferring data to the driver. The FFT processor shall request demand-mode DMA when the data FIFO is not empty and EN_DDMA is 1.

3.6. MODE register (0x14)

The MODE register shall determine which mode the FFT processor operates, as defined by **SPEC7**.

Field	Bit range	Reset state	Type	Function
SELECT	6:0	n/a	RO	AstroFFT mode as per **SPEC7**: 0x0 => FFT 0x1 => QFFT 0x2 => RFFT 0x3 => Analogue 0x4 to 0x7F => *reserved*
	7	0	RAX/MBZ	*Reserved*
RANGE (mode 0, 1, 2)	9:8	0	R/W	Selects the integrator range returned in the data FIFO: 0x0 => top 32 bits 0x1 => 32 bits below top 8 bits 0x2 => 32 bits below top 16 bits 0x3 => 32 bits below top 24 bits
(mode 3)	9:8	0	RAX/MBZ	*Reserved*
	31:10	0	RAX/MBZ	*Reserved*

Table 8. MODE register fields

The MODE.SELECT field is hardcoded into each AstroFFT bitstream, and its value corresponds to the AstroFFT modes defined by **SPEC7**.

The MODE.RANGE field applies only to AstroFFT modes 0, 1 and 2 and selects the 32-bit range of the integrated data that is returned to the host via the data FIFO.

3.7. INTMASK register (0x18)

The INTMASK register shall determine which sources of interrupt in the FFT processor can generate an interrupt on the host CPU.

Field	Bit range	Reset	Type	Function

		state		
CYCLE	0	1	R/W	0 => generate an interrupt when a processing cycle is completed.
PHASE	1	1	R/W	0 => generate an interrupt when a processing phase is completed.
NEMPTY	2	1	R/W	0 => generate an interrupt when the FIFO becomes not empty.
	4:3	All 1's	RAX/MB1	*Reserved*
UDF	5	1	R/W	0 => generate an interrupt when the FIFO underflows.
OVF	6	1	R/W	0 => generate an interrupt when the FIFO overflows.
	7	All 1's	RAX/MB1	*Reserved*
OVR	8	1	R/W	0 => generate an interrupt when integrator goes over range
	31:9	All 1's	RAX/MB1	*Reserved*

Table 9. INTMASK register fields

3.8. INTSTAT register (0x1C)

The INTSTAT register provides a means for the driver to determine why an interrupt occurred.

Field	Bit range	Reset state	Type	Function
CYCLE	0	0	R/W1C	1 => a processing cycle was completed
PHASE	1	0	R/W1C	1 => a processing phase was completed
NEMPTY	2	0	R/W1C	1 => the FIFO became non-empty
	4:3	0	RAX/MBZ	*Reserved*
UDF	5	0	R/W1C	1 => the FIFO underflowed
OVF	6	0	R/W1C	1 => the FIFO overflowed
	7	0	RAX/MBZ	*Reserved*
OVR	8	0	R/W1C	1 => the integrator was over range
	31:9	0	RAX/MBZ	*Reserved*

Table 10. INTSTAT register fields

A bit in INTSTAT can transition to 1 even if the corresponding bit in INTMASK is 1. Once set, the bits in INTSTAT can be cleared only by issuing a reset via RESET.ALL or by writing 1s to INTSTAT.

Each time the INTSTAT register is written, the FPGA interrupt line FINTI_L is momentarily forced high in order to 'rearm' the edge-triggered interrupt pin on the CPLD of the ADM-XRC.

The FINTI_L line will be asserted whenever there is at least one bit in INTSTAT that is 1 and whose corresponding bit in INTMASK is 0.

The driver can rely on INTSTAT.CYCLE as an indicator of FIFO readiness. INTSTAT.CYCLE will not transition to 1 until the last word of data for a processing cycle has been placed in the data FIFO.

3.9. FIFOSTAT register (0x24)

The FIFOSTAT register shall return status information about the data FIFO.

Field	Bit range	Reset state	Type	Function
	15:0	0	RAX/MBZ	*Reserved*
EMPTY	16	1	RO	1 => data FIFO is empty
	31:17	0	RAX/MBZ	*Reserved*

Table 11. FIFOSTAT register fields

3.10. PH_N register (0x28)

The PH_N register shall specify how many processing phases are performed per processing cycle.

Field	Bit range	Reset state	Type	Function
N	23:0	0	R/W	Should be programmed with 1 less than the number of processing phases to perform in each processing cycle.
	31:24	0	RAX/MBZ	*Reserved*

Table 12. PH_N register fields

The PH_N register can be set to an arbitrary value in AstroFFT modes 0, 1 and 2. In mode 3 it must be set to 0 in order to prevent overflow of the data FIFO, as explained in section 7.1.

Changes to PH_N take effect at the beginning of the next processing cycle.

3.11. CYCSTAT register (0x2C)

The CYCSTAT register shall return information about the current processing cycle.

Field	Bit range	Reset state	Type	Function
N	23:0	0	RO	Returns one less than the number of processing cycles that remain to be performed.
MODE	30:24	TBD	RO	The AstroFFT mode being used for the current processing cycle
	31	0	RAX/MBZ	*Reserved*

Table 13. CYCSTAT register fields

Both the CYCSTAT.N and CYCSTAT.MODE fields do not contain valid values unless CYCSTAT.SUSP is 0.

The CYCSTAT.MODE field shall reflect the value of the MODE.SELECT field as it was at the beginning of the current processing cycle, rather than the current value of MODE.SELECT.

3.12. PHSTAT register (0x30)

The PHSTAT register shall return information about the current processing phase. Its format is TBD and will depend upon the current value of CYCSTAT.MODE.

Field	Bit range	Reset	Type	Function

		state		
N	23:0	0	RO	Returns one less than the number of processing phases that remain to be performed in the current processing cycle.
	31:24	0	RAX/MBZ	*Reserved*

Table 14. PHSTAT register fields

3.13. MISCSTAT register (0x34)

Field	Bit range	Reset state	Type	Function
LCLK	0	n/a	RO	1 => LCLK is ok
PCLK	1	n/a	RO	1 => PCLK is ok
SCLK	2	n/a	RO	1 => SCLK is ok
	31:3	0	RAX/MBZ	*Reserved*

Table 15. MISCSTAT register fields

3.14. CYC_N register (0x38)

The CYC_N register shall specify how many processing cycles are to be performed when the START command is issued via the CMD register.

Field	Bit range	Reset state	Type	Function
N	23:0	0	R/W	Should be programmed with 1 less than the number of processing cycles to perform.
	31:24	0	RAX/MBZ	*Reserved*

Table 16. CYC_N register fields

The CYC_N register can be set to an arbitrary value in AstroFFT modes 0, 1 and 2. In mode 3 it must be set to 0 in order to prevent overflow of the data FIFO, as explained in section 6.1.

3.15. ADC_CTL register (0x3C)

The ADC_CTL register allows control over the sampler frontend.

Field	Bit range	Reset state	Type	Function
SOURCE	0	1	R/W	Selects sampling clock source 0 => Use external clock input 1 => Use internal oscillator
DIVIDE	3:1	0x7	R/W	Selects division factor for selected clock source: 7 => Divide by 1 (no division) 6 => Divide by 2 5 => Divide by 4 4 => Divide by 8 3 => Divide by 16 Other values are reserved.
	4	1	RAX/MBZ	*Reserved*

NINTLV	5	1	R/W	Selects interleaved or non-interleaved sampler mode: 0 => Interleaved mode 1 => Non-interleaved mode For AstroFFT modes 0, 1 and 3, this field must be 1. For AstroFFT mode 2, this field must be 0.
	6	1	RAX/MBZ	*Reserved*
RESET	7	0	R/W	0 => Sampler reset not asserted 1 => Sampler reset asserted
	31:8	0	RAX/MBZ	*Reserved*

Table 17. ADC_CTL register fields

A change to the SOURCE or DIVIDE fields of ADC_CTL must be accompanied by a sampler reset pulse.

When software reset is applied, via the RESET.ALL field, a sampler reset pulse is automatically applied to the sampler frontend.

3.16. FIFO_RD register (0x80)

The FIFO_RD register is the port by which the driver can sequentially read the data from the data FIFO.

Field	Bit range	Reset state	Type	Function
DATA	31:0	n/a	RO	Read port of data FIFO.

Table 18. FIFO_RD register fields

The FIFO_RD register can be read by the driver in one of three ways, all functionally equivalent:

- Direct slave, using programmed I/O
- DMA using PLX9080 DMA engines in non-demand-mode
- DMA using PLX9080 DMA engines in demand-mode

In order to use demand-mode DMA, the driver shall set CONTROL.DDMA to 1.

The format of FIFO_RD.DATA and the number of words transferred for each processing cycle is to be decided at this version of this document, but shall depend upon the AstroFFT mode under which the data was generated.

3.17. DBG registers (0x100-0x3FC)

A set of registers shall exist to assist debug and test. These registers shall be firmed up later in the development cycle in a future revision of this document.

4. Required ADM-XRC configuration

The registers defined in this document exist in PLX9080 direct slave space 0.

The configuration of the PLX9080 to enable direct slave register access is detailed in section 4.1. Configuration for non-demand mode DMA is detailed in section 4.2, while demand mode DMA configuration is detailed in section 4.3.

4.1. PLX9080 direct slave configuration

The driver should access the AstroFFT registers via direct slave space 0. Table 19 below details the values that must be programmed into certain fields of the PLX9080 local bus configuration registers. Register fields that are not specified with a particular value below should be left in their default state, or whatever state is appropriate for the driver.

Offset in PCI space	PLX register field	Value (Verilog convention)
0x04	LAS0BA[0]	1'b1
0x04	LAS0BA[31:4]	28'b0
0x08	MARBR[20:19]	2'b00
0x08	MARBR[24]	1'b1
0x18	LBRD0[1:0]	2'b11
0x18	LBRD0[6]	1'b1
0x18	LBRD0[7]	1'b1
0x18	LBRD0[8]	1'b1

Table 19. Required PLX local bus configuration

4.2. PLX9080 non-demand mode DMA configuration

Register fields that are not specified with a particular value below should be left in their default state, or whatever state is appropriate for the driver. The values in table 20 should be written into the DMA control registers of the PLX9080:

Offset in PCI space	PLX register field	Value (Verilog convention)	Notes
0x80 & 0x94	DMAMODEx[1:0]	2'b11	
0x80 & 0x94	DMAMODEx[5:2]	4'b0	
0x80 & 0x94	DMAMODEx[6]	1'b1	
0x80 & 0x94	DMAMODEx[7]	1'b1	
0x80 & 0x94	DMAMODEx[8]	1'b1	
0x80 & 0x94	DMAMODEx[11]	1'b1	
0x80 & 0x94	DMAMODEx[12]	1'b0	
0x80 & 0x94 0x80 & 0x94	DMAMODEx[14] DMAMODEx[15]	1'b0 1'b0	
0x80 & 0x94	DMAMODEx[16]	1'b0	
0x80 & 0x94	DMAMODEx[17]	1'b1	
0x88 & 0x9C	DMALADRx[31:0]	32'h80	
0x8C & 0xA0	DMASIZx[1:0]	2'b00	
0x90 & 0xA4	DMADPRx[0]	1'b1	1,2
0x90 & 0xA4	DMADPRx[3]	1'b1	1

Table 20. Required PLX9080 non-demand mode DMA configuration

Notes

1. Must be written to the DMADPRx registers on each DMA transfer to ensure correct operation.
2. Need not be set if chained mode DMA is not used.

4.3. PLX9080 configuration for demand mode DMA

If demand mode DMA is used, the DMA channels of the PLX9080 shall be used as follows by the driver:

- DMA channel 1 - used to read the FIFO_RD register
- DMA channel 0 - not used

Register fields that are not specified with a particular value below should be left in their default state, or whatever state is appropriate for the driver. The values in table 21 should be written into the DMA control registers of the PLX9080:

Offset in PCI space	PLX register field	Value (Verilog convention)	Notes
0x94	DMAMODE1[1:0]	2'b11	
0x94	DMAMODE1[5:2]	4'b0	
0x94	DMAMODE1[6]	1'b1	
0x94	DMAMODE1[7]	1'b1	
0x94	DMAMODE1[8]	1'b1	
0x94	DMAMODE1[11]	1'b1	
0x94	DMAMODE1[12]	1'b1	
0x94	DMAMODE1[14]	1'b0	
0x94	DMAMODE1[15]	1'b0	
0x94	DMAMODE1[16]	1'b0	
0x94	DMAMODE1[17]	1'b1	
0x9C	DMALADR1[31:0]	32'h80	
0xA0	DMASIZ1[1:0]	2'b00	
0xA4	DMADPR1[0]	1'b1	1,2
0xA4	DMADPR1[3]	1'b1	1

Table 21. Required PLX9080 demand mode DMA configuration

Notes
1. Must be written to the DMADPR1 register on each DMA transfer to ensure correct operation.
2. Need not be set if chained mode DMA is not used.

5. Local bus operating frequency

The AstroFFT FPGA's local bus interface shall be clocked at any local bus clock frequency up to 33MHz. The lower bound on clock frequency is imposed in order to avoid overflow of the data FIFO.

6. Data FIFO packet formats

This section details the format of data read from the FIFO_RD register for each AstroFFT mode.

6.1. AstroFFT modes 0 and 1

In AstroFFT modes 0 and 1, a packet consisting of 4096 DWORDs shall be placed in the data FIFO at completion of each processing cycle.

Each DWORD shall correspond to one bin in the integrated power spectrum, read out in natural order (as opposed to bit-reversed order). The first DWORD (index 0) read from the data FIFO shall correspond to the lowest frequency bin and the 4096th DWORD (index 4095) read from the data FIFO shall correspond to the highest frequency bin.

Note that an arbitrary scaling factor may be included in the values read from the data FIFO.

6.2. AstroFFT mode 2

In AstroFFT mode 2, a packet consisting of 4096 DWORDs shall be placed in the data FIFO at completion of each processing cycle.

Each DWORD shall correspond to one bin in the integrated power spectrum. The correspondence of each DWORD to a bin is unlike modes 0 and 1 and is best described by the following tables:

Data word from FIFO # (zero-indexed)	FFT bin # (zero-indexed)
0	0
1, 3, 5, …, 4093	1, 2, 3, …, 2047
2, 4, 6, …, 4094	4095, 4094, 4093, …, 2049
4095	2048

Table 22. AstroFFT mode 2 FIFO-to-FFT ordering

FFT bin # (zero-indexed)	Data word from FIFO # (zero-indexed)
0	0
1, 2, 3, …, 2047	1, 3, 5, …, 4093
2048	4095
2049, 2050, 2051 …, 4095	4094, 4092, 4090, …, 2

Table 23. AstroFFT mode 2 FFT-to-FIFO ordering

Note that an arbitrary scaling factor may be included in the values read from the data FIFO.

6.3. AstroFFT mode 3

In AstroFFT modes 0, 1 and 2, a packet consisting of 5004 DWORDs shall be placed in the data FIFO at completion of each processing cycle.

The first 4096 DWORDs of the packet (indexed 0 to 4095) shall be a block of dual-channel samples as received from the A/D card. Each DWORD shall consist of two 16-bit fields as follows:

Field	Bit range	Function
REAL	15:0	Analogue data, real channel
IMAG	31:16	Analogue data, imaginary channel

Table 24. AstroFFT mode 3 data words

REAL and IMAG shall be zero padded from the right hand side such that regardless of the A/D precision, the full 16 bit range of REAL and IMAG shall be occupied by the data.

The last 8 DWORDs of the packet (indexed 4096 to 5003) shall consist of statistics about the data, as follows:

DWORD index	Function
4096	Low 32 bits of sum of squares, real channel
4097	High 32 bits of sum of squares, real channel
4098	Low 32 bits of sum of squares, imaginary channel
4099	High 32 bits of sum of squares, imaginary channel
5000	Low 32 bits of number of clipped samples, real channel
5001	High 32 bits of number of clipped samples, real channel
5002	Low 32 bits of number of clipped samples, imaginary channel
5003	High 32 bits of number of clipped samples, imaginary channel

Table 25. AstroFFT mode 3 statistic words

7. Driver notes

7.1. Capturing data in AstroFFT mode 3

The average data throughput to the host in AstroFFT mode 3 (200MB/s) exceeds the bandwidth of a 32-bit, 33MHz PCI bus. Therefore, unless the host can guarantee that the throughput requirements of the data FIFO in this mode can be met, the host should ensure that only a single processing cycle of a single processing phase is performed at a time.

If processing cycles of more than one processing phase are performed, and required throughput to the host cannot be achieved, the data FIFO will overflow and data will be lost.

7.2. Capturing data in AstroFFT modes 0, 1, 2

When the number of processing phases per processing cycle is set to a reasonable value, such as 100 or more, the average data throughput to the host in AstroFFT modes 0, 1 and 2 is one to two orders of magnitude less than the achievable PCI bandwidth in a modern PC. Therefore, the FFT processor can run continuously in these modes.

If the number of processing phases per processing cycle is set to a low value, and the FFT processor runs continuously, the required data throughput to the host may exceed the achievable PCI bandwidth in the system. Host software should not operate the FFT processor in such a manner.