FP-DAQ1616

FeaturePak™ I/O Module with Analog Data Acquisition

Rev A3: May 2012

Revision	Date	Comment
A	1/26/2011	Initial version
A1	4/29/11	D/A Waveform Generator information added
A2	9/22/2011	Specifications updated
A3	5/4/12	Resolution information added, minor edits

FOR TECHNICAL SUPPORT PLEASE CONTACT:

support@diamondsystems.com

Diamond Systems Corporation
555 Ellis Street
Mountain View, CA 94043 USA
Tel 1-650-810-2500
Fax 1-650-810-2525
www.diamondsystems.com

CONTENTS

IMPORTANT SAFE HANDLING INFORMATION

WARNING!

ESD-Sensitive Electronic Equipment

Observe ESD-safe handling procedures when working with this product.

Always use this product in a properly grounded work area and wear appropriate ESD-preventive clothing and/or accessories.

Always store this product in ESD-protective packaging when not in use.

Safe Handling Precautions

The FP-DAQ1616 board contains a high density connector with many connections to sensitive electronic components. This creates many opportunities for accidental damage during handling, installation and connection to other equipment. The list here describes common causes of failure found on boards returned to Diamond Systems for repair. This information is provided as a source of advice to help you prevent damaging your Diamond (or any vendor's) embedded computer boards.

ESD damage – This type of damage is usually almost impossible to detect, because there is no visual sign of failure or damage. The symptom is that the board eventually simply stops working, because some component becomes defective. Usually the failure can be identified and the chip can be replaced. To prevent ESD damage, always follow proper ESD-prevention practices when handling computer boards.

Damage during handling or storage – On some boards we have noticed physical damage from mishandling. A common observation is that a screwdriver slipped while installing the board, causing a gouge in the PCB surface and cutting signal traces or damaging components.

Another common observation is damaged board corners, indicating the board was dropped. This may or may not cause damage to the circuitry, depending on what is near the corner. Most of our boards are designed with at least 25 mils clearance between the board edge and any component pad, and ground / power planes are at least 20 mils from the edge to avoid possible shorting from this type of damage. However these design rules are not sufficient to prevent damage in all situations.

A third cause of failure is when a metal screwdriver tip slips, or a screw drops onto the board while it is powered on, causing a short between a power pin and a signal pin on a component. This can cause overvoltage / power supply problems described below. To avoid this type of failure, only perform assembly operations when the system is powered off.

Sometimes boards are stored in racks with slots that grip the edge of the board. This is a common practice for board manufacturers. However our boards are generally very dense, and if the board has components very close to the board edge, they can be damaged or even knocked off the board when the board tilts back in the rack. Diamond recommends that all our boards be stored only in individual ESD-safe packaging. If multiple boards are stored together, they should be contained in bins with dividers between boards. Do not pile boards on top of each other or cram too many boards into a small location. This can cause damage to connector pins or fragile components.

Power supply wired backwards – Our power supplies and boards are not designed to withstand a reverse power supply connection. This will destroy each IC that is connected to the power supply (i.e. almost all ICs). In this case the board will most likely will be unrepairable and must be replaced. A chip destroyed by reverse power or by excessive power will often have a visible hole on the top or show some deformation on the top surface due to vaporization inside the package. **Check twice before applying power!**

Overvoltage on analog input – If a voltage applied to an analog input exceeds the design specification of the board, the input multiplexor and/or parts behind it can be damaged. Most of our boards will withstand an erroneous connection of up to ±35V on the analog inputs, even when the board is powered off, but not all boards, and not in all conditions.

Overvoltage on analog output – If an analog output is accidentally connected to another output signal or a power supply voltage, the output can be damaged. On most of our boards, a short circuit to ground on an analog output will not cause trouble.

Overvoltage on digital I/O line – If a digital I/O signal is connected to a voltage above the maximum specified voltage, the digital circuitry can be damaged. On most of our boards the acceptable range of voltages connected to digital I/O signals is 0-5V, and they can withstand about 0.5V beyond that (-0.5 to 5.5V) before being damaged. However logic signals at 12V and even 24V are common, and if one of these is connected to a 5V logic chip, the chip will be damaged, and the damage could even extend past that chip to others in the circuit

1. INTRODUCTION

1.1 Description

The FP-DAQ1616 is a FeaturePak-format data acquisition board with a full set of analog and digital I/O features.

It offers 16 single-ended or 8 differential analog voltage inputs with 12-bit resolution and programmable input range; 2.0MHz maximum aggregate A/D sampling rate with 16K FIFO operation; 16 analog voltage outputs with 16-bit resolution, user-selectable analog output ranges; 56 total programmable digital I/O bits, 3.3V logic compatible; one 32-bit counter/timer for A/D sampling rate control and one 32-bit counter/timer for user counting and timing functions; and four 24-bit PWM generators.

The FP-DAQ1616 has a PCI Express host interface brought out through the FeaturePak connector. All analog and digital I/O connects through this connector.

1.2 Features

Analog Inputs

- 16 12-bit or 16-bit analog inputs with up to 2MHz aggregate maximum sampling rate; 1MHz for multi-channel sampling
- Programmable input ranges, polarity, and mode
- 16K sample A/D FIFO with programmable threshold
- Autocalibration circuit with precision reference voltages

Analog Outputs

- 16 16-bit analog outputs with programmable range and polarity
- D/A waveform generator

Digital I/O

- 56 programmable digital I/O

Counter/Timers and A/D Triggering

- 2 32-bit counter/timers for A/D timing and general purpose use
- 4 24-bit pulse-width modulator circuits

Miscellaneous

- One PCI Express x1 lane host interface
- FeaturePak form-factor compliant
- Zero height expansion module
- -40°C to +85°C operating temperature
- Universal Driver software support

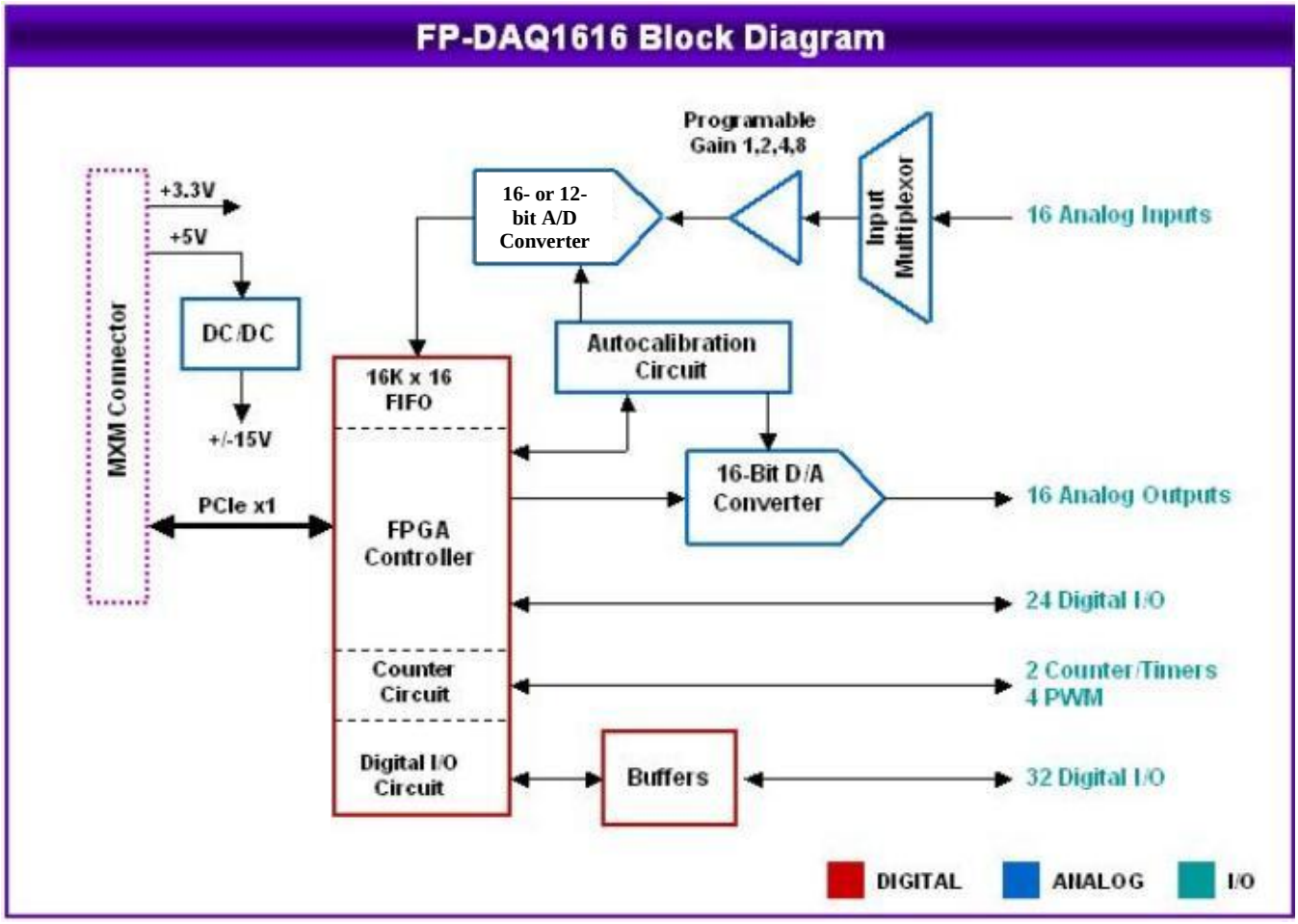
1.3 FeaturePak Resources Used By FP-DAQ1616

The following table identifies which resources among those defined by the FeaturePak specification that are used in this product:

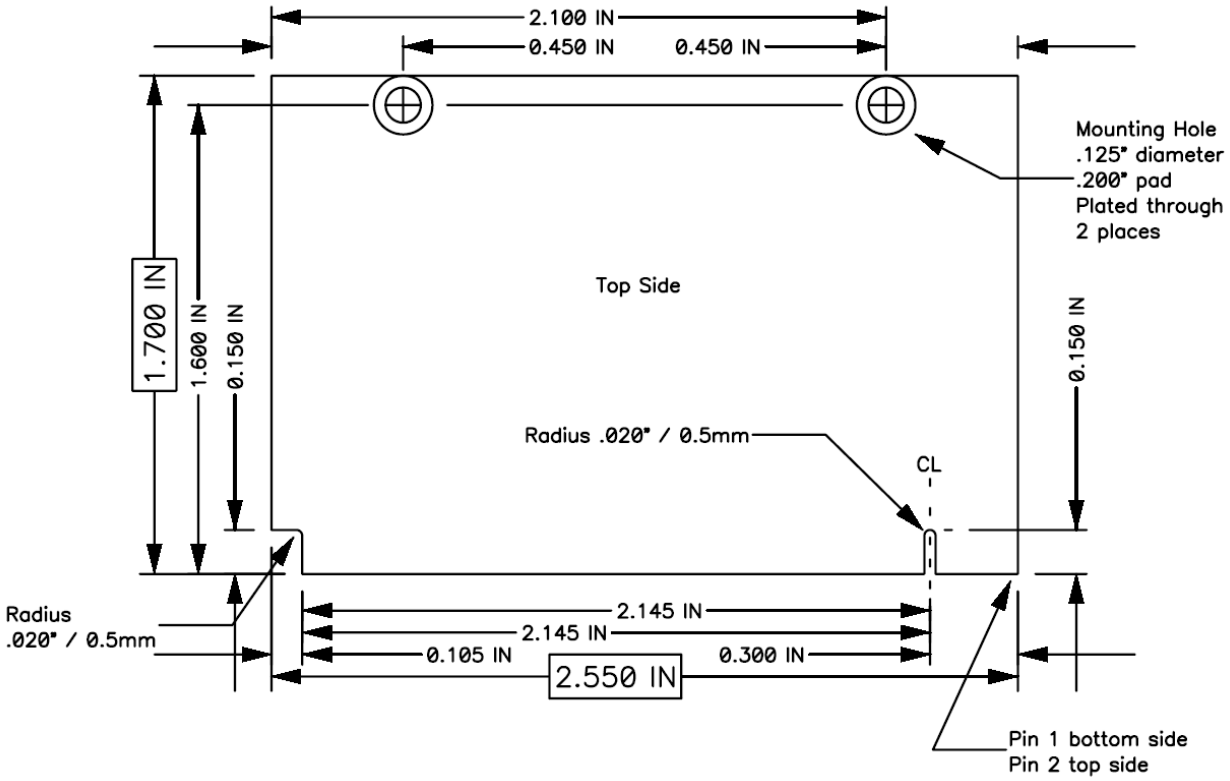
FeaturePak™ Module Resources	
Company: Diamond Systems Corp.	
Product: FP-DAQ1616	
Host Interface Resources Supported	
PCIe x1 links	1
USB channels	–
Serial port	–
SMBus	–
PCIe Reset	√
Sys Reset	√
JTAG	√
+3.3V	√
+5V	√
+12V	–
Notes: www.featurepak.org/label	

2. FUNCTIONAL OVERVIEW

2.1 Functional Block Diagram



2.2 FP-DAQ1616 Board Drawing



3. CONNECTOR PINOUT AND PIN DESCRIPTION

3.1 FeaturePak Connector

The FP-DAQ1616 uses the FeaturePak standard, which provides a card form-factor and connector suitable for high-density, compact applications. See http://featurepak.org/ for further details of the FeaturePak embedded I/O modules standard, including the specification.

In this document we will use the terms FeaturePak edge card connector and MXM connector interchangeably.

The FP-DAQ1616 uses an MXM 230-pin connector for all I/O, both to the main CPU for user I/O connections. The board contains gold-plated fingers conforming to the MXM physical standard for interfacing to the MXM connector.

The system interface signals on the MXM connector are predefined for all FeaturePak modules and a portion of pins are reserved for individual module I/O and are freely assignable for each module. Signals are grouped into several categories:

- System interface (PCIe)
- Slot/card management signals
- Dedicated user I/O
- Power/ground
- Reserved for future expansion

The module connects to 5 reserved pins to provide compatibility with possible future expansion of the FeaturePak standard.

The user I/O consists of 100 signals organized as 2 50-pin groups. On a general purpose baseboard, these signals directly drive 2 50-pin I/O connectors. The 50-pin connectors may then be connected to application-specific cabling, or an I/O adapter board may be plugged onto them to provide a custom I/O connector configuration for the system enclosure. A custom or application-specific main board may elect to use these signals in any way desired without conflicting with the standard.

A special feature of the FeaturePak I/O pin assignment is the electrical isolation provided on 32 of the module I/O pins on the primary 50-pin connector. These pins are separated from each other and from other connector pins by 34 unassigned pins (marked NC) which are to be left unconnected on the main board (no pads or traces leading to these pins). The module should omit the corresponding fingers on the edge connector.

3.1.1 PCI Express

The FP-DAQ1616 uses a single PCI Express x1 link for its host interface and does not use the USB or serial interfaces allowed in the FeaturePak specification. It is reset by the PCI-Reset- input. The module routes the Slot ID signals to a register for readback by the host, and it drives the Present- output line low.

3.1.2 System Interface Signals

Note: not all functionalities that are defined in the FeaturePak socket standard are supported by the FP-DAQ1616 module. For example, the USB and serial port interfaces are not utilized on this card.

3.2 FeaturePak Connector Pinout By Pin

This section indicates standard FeaturePak connector signal assignments. The I/OA and I/OB pin groups are mapped to specific I/O functions implemented by the FP-DAQ1616 board, as indicated in following sections.

	Pins 1 – 114		
+3.3V	1	2	+12V
+3.3V	3	4	PS-Current
Ground	5	6	Ground
PCIe-TX1+	7	8	PCIe-RX1+
PCIe-TX1-	9	10	PCIe-RX1-
Ground	11	12	Ground
PCIe-CLK1+	13	14	PCIe-CLK2+
PCIe-CLK1-	15	16	PCIe-CLK2-
Ground	17	18	Ground
PCIe-TX2+	19	20	PCIe-RX2+
PCIe-TX2-	21	22	PCIe-RX2-
Ground	23	24	Ground
PCIe-Reset-	25	26	Reserved
Reserved	27	28	Reserved
Reserved	29	30	Reserved
Reserved	31	32	Reserved
Reserved	33	34	Reserved
Ground	35	36	Ground
USB-Ch1+	37	38	USB-Ch2+
USB-Ch1-	39	40	USB-Ch2-
Ground	41	42	Ground
+3.3V	43	44	USB-OC1/2-
+3.3V	45	46	Serial-RX1
Serial-TX1	47	48	Serial-CTS1
Serial-RTS1	49	50	SMBclk
SMBalert#	51	52	SMBdata
Slot ID 2	53	54	Slot ID 1
Slot ID 0	55	56	Present-
JTAG-TDI	57	58	JTAG-TDO
JTAG-CLK	59	60	JTAG-TMS
Sys-Reset-	61	62	Reserved
+3.3V	63	64	Ground
+3.3V	65	66	Ground
Reserved	67	68	Reserved
Reserved	69	70	Reserved
+3.3V	71	72	Ground
Reserved	73	74	Reserved
Reserved	75	76	Reserved
+3.3V	77	78	Ground
Reserved	79	80	Reserved
Reserved	81	82	Reserved
Reserved	83	84	Reserved
Reserved	85	86	Reserved
+5V	87	88	Ground
+5V	89	90	Ground
I/OB-50	91	92	I/OB-49
I/OB-48	93	94	I/OB-47
I/OB-46	95	96	I/OB-45
I/OB-44	97	98	I/OB-43
I/OB-42	99	100	I/OB-41
I/OB-40	101	102	I/OB-39
I/OB-38	103	104	I/OB-37
I/OB-36	105	106	I/OB-35
I/OB-34	107	108	I/OB-33
I/OB-32	109	110	I/OB-31
I/OB-30	111	112	I/OB-29
I/OB-28	113	114	I/OB-27

	Pin 115 – 230		
I/OB-26	115	116	I/OB-25
I/OB-24	117	118	I/OB-23
I/OB-22	119	120	I/OB-21
I/OB-20	121	122	I/OB-19
I/OB-18	123	124	I/OB-17
I/OB-16	125	126	I/OB-15
I/OB-14	127	128	I/OB-13
I/OB-12	129	130	I/OB-11
I/OB-10	131	132	I/OB-9
I/OB-8	133	134	I/OB-7
I/OB-6	135	136	I/OB-5
I/OB-4	137	138	I/OB-3
I/OB-2	139	140	I/OB-1
+5V	141	142	Ground
+5V	143	144	Ground
+5V	145	146	Ground
I/OA-50	147	148	I/OA-49
I/OA-48	149	150	I/OA-47
I/OA-46	151	152	I/OA-45
I/OA-44	153	154	I/OA-43
I/OA-42	155	156	I/OA-41
I/OA-40	157	158	I/OA-39
I/OA-38	159	160	I/OA-37
I/OA-36	161	162	I/OA-35
(NC)	163	164	(NC)
I/OA-34	165	166	I/OA-33
(NC)	167	168	(NC)
I/OA-32	169	170	I/OA-31
(NC)	171	172	(NC)
I/OA-30	173	174	I/OA-29
(NC)	175	176	(NC)
I/OA-28	177	178	I/OA-27
(NC)	179	180	(NC)
I/OA-26	181	182	I/OA-25
(NC)	183	184	(NC)
I/OA-24	185	186	I/OA-23
(NC)	187	188	(NC)
I/OA-22	189	190	I/OA-21
(NC)	191	192	(NC)
I/OA-20	193	194	I/OA-19
(NC)	195	196	(NC)
I/OA-18	197	198	I/OA-17
(NC)	199	200	(NC)
I/OA-16	201	202	I/OA-15
(NC)	203	204	(NC)
I/OA-14	205	206	I/OA-13
(NC)	207	208	(NC)
I/OA-12	209	210	I/OA-11
(NC)	211	212	(NC)
I/OA-10	213	214	I/OA-9
(NC)	215	216	(NC)
I/OA-8	217	218	I/OA-7
(NC)	219	220	(NC)
I/OA-6	221	222	I/OA-5
(NC)	223	224	(NC)
I/OA-4	225	226	I/OA-3
(NC)	227	228	(NC)
I/OA-2	229	230	I/OA-1

3.3 Analog I/O Signals

The FP-DAQ1616 module's analog I/O signals are brought out on FeaturePak connector pins 147 through 230 as shown here. The FeaturePak standard defines a Primary Group of pins for I/O; the mapping of those pins to the FP-DAQ1616 pins is in the table below. The FeaturePak specification allocates (up to) 34 pins for use either in isolated analog signal pairs for increased noise immunity, or as separate lines.

FeaturePak connector pin no.		Typical I/O connector pin no.		FeaturePak connector pin no.	
230	Vin 0/0+	1	2	Vin 8/0-	229
226	Vin 1/1+	3	4	Vin 9/1-	225
222	Vin 2/2+	5	6	Vin 10/2-	221
218	Vin 3/3+	7	8	Vin 11/3-	217
214	Vin 4/4+	9	10	Vin 12/4-	213
210	Vin 5/5+	11	12	Vin 13/5-	209
206	Vin 6/6+	13	14	Vin 14/6-	205
202	Vin 7/7+	15	16	Vin 15/7-	201
198	Aground (Vin)	17	18	Aground (Vin)	197
194	Vout 0	19	20	Vout 8	193
190	Vout 1	21	22	Vout 9	189
186	Vout 2	23	24	Vout 10	185
182	Vout 3	25	26	Vout 11	181
178	Vout 4	27	28	Vout 12	177
174	Vout 5	29	30	Vout 13	173
170	Vout 6	31	32	Vout 14	169
166	Vout 7	33	34	Vout 15	165
162	Aground (Vout)	35	36	Aground (Vout)	161
160	ADTrig	37	38	Ctr0Clk	159
158	Ctr1Clk	39	40	Ctr1Out	157
156	Aux0	41	42	Aux1	155
154	Aux2	43	44	Aux3	153
152	ADGate / Aux4	45	46	Aux5 / Ctr1Gate	151
150	WDTOut / Aux6	47	48	Aux7 / WDTIn	149
148	+3.3V	49	50	Dground	147

Signal Name	Definition
Vin 7/7+ ~ Vin 0/0+	Analog input channels 7 – 0 in single-ended mode; High side of input channels 7 – 0 in differential mode
Vin 15/7- ~ Vin 8/0-	Analog input channels 15 – 8 in both single-ended mode; Low side of input channels 7 – 0 in differential mode
Vout 0-15	Analog output channels 0 – 15
Aground (Vout), (Vin)	Analog ground; used for analog signals only Vout pin is for the analog outputs; Vin pin is for the analog inputs
Dground	Digital ground reference for digital input signals
AD Gate	Enables A/D sampling when high or open, disables when pulled low
AD Trig	External A/D trigger or clock input
Ctr0Clk, Ctr1Clk	Counter 0/1 optional external clock inputs
Ctr1Out	Counter 1 output signal
AUX0-7	Auxiliary digital I/O port; bit direction programmable; Line 4-7 have auxiliary functions for A/D gating, counter gating, and watchdog timer.

3.4 Digital I/O Signals

The FP-DAQ1616's digital I/O signals are brought out on FeaturePak connector pins 91 through 140, as shown below.

FeaturePak connector pin no.		Typical I/O connector pin no.		FeaturePak connector pin no.	
140	DIO A0	1	2	DIO A1	139
138	DIO A2	3	4	DIO A3	137
136	DIO A4	5	6	DIO A5	135
134	DIO A6	7	8	DIO A7	133
132	DIO B0	9	10	DIO B1	131
130	DIO B2	11	12	DIO B3	129
128	DIO B4	13	14	DIO B5	127
126	DIO B6	15	16	DIO B7	125
124	DIO C0	17	18	DIO C1	123
122	DIO C2	19	20	DIO C3	121
120	DIO C4	21	22	DIO C5	119
118	DIO C6	23	24	DIO C7	117
116	DIO D0	25	26	DIO D1	115
114	DIO D2	27	28	DIO D3	113
112	DIO D4	29	30	DIO D5	111
110	DIO D6	31	32	DIO D7	109
108	DIO E0	33	34	DIO E1	107
106	DIO E2	35	36	DIO E3	105
104	DIO E4	37	38	DIO E5	103
102	DIO E6	39	40	DIO E7	101
100	PWM0 / DIO F0	41	42	DIO F1 / PWM1	99
98	PWM2 / DIO F2	43	44	DIO F3 / PWM3	97
96	DIO F4	45	46	DIO F5	95
94	Latch- / DIO F6	47	48	DIO F7 / Ack-	93
92	+3.3V	49	50	Dground	91

Signal Name	Definition
DIO A7-A0	Digital I/O port A; byte direction programmable
DIO B7-B0	Digital I/O port B; byte direction programmable
DIO C7-C0	Digital I/O port C; byte direction programmable
DIO D7-D0	Digital I/O port D; byte direction programmable
DIO E7-E0	Digital I/O port E; bit direction programmable
DIO F7-F0	Digital I/O port F; bit direction programmable Port F signals have auxiliary functions enabled with control registers:
PWM0-3	32-bit programmable pulse width modulation outputs
Latch- / Ack-	DIO latch and acknowledge signals to enable DIO with handshaking; Latch- also serves as a digital input signal to drive a user-controlled interrupt.
Dground	Digital ground

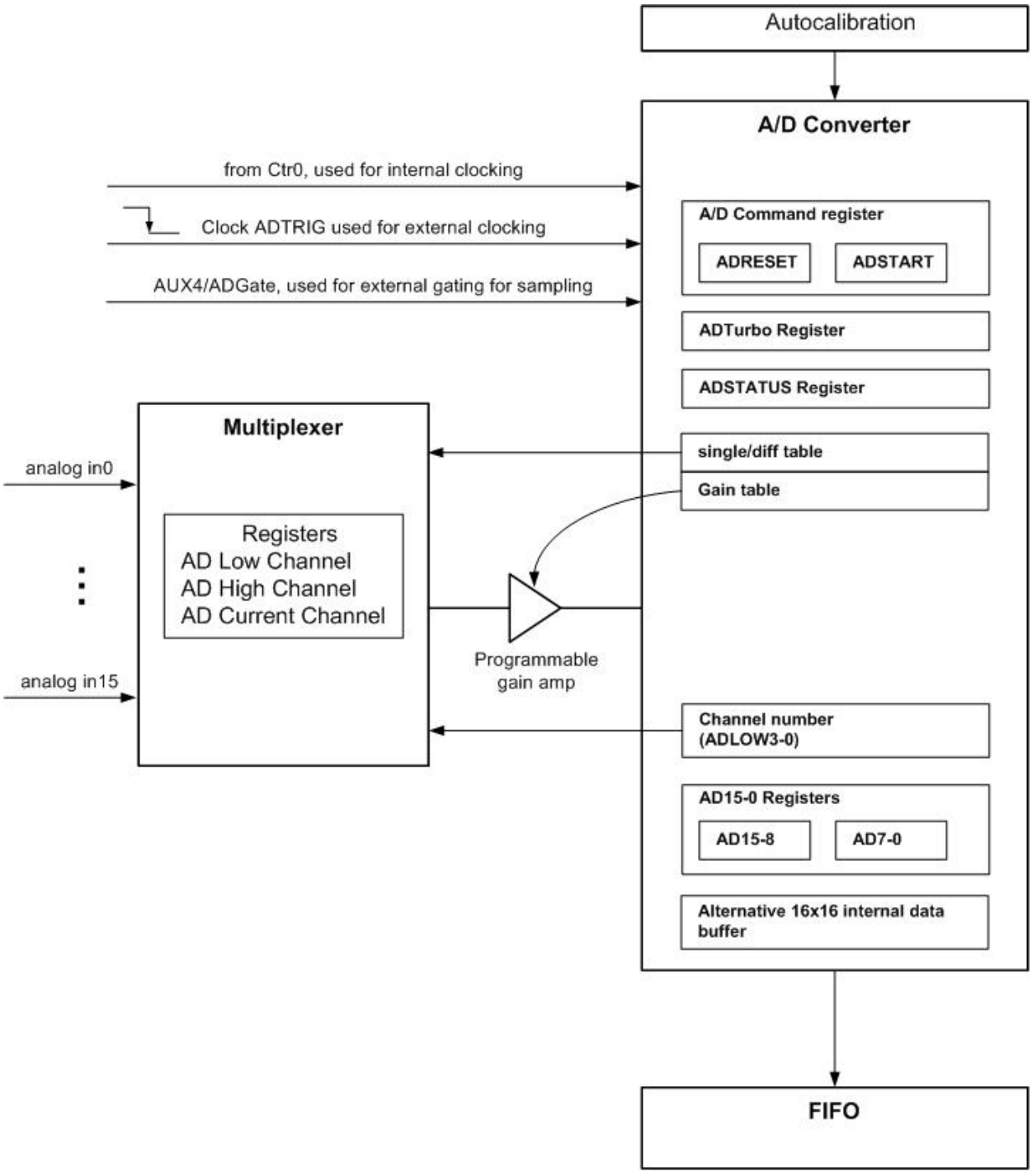
4. THEORY OF OPERATION

This chapter provides an orientation to the functional architecture of blocks on the board. Additional details are in the register programming section and specific chapters on various blocks.

4.1 A/D Block

The diagram below gives a quick overview of this block's key inputs, outputs, and internals.

A/D Block I/O and Internals



Channels and Their Properties

Depending which operating mode is selected using the A/D Advanced Configuration Register, the input multiplexer organizes inputs as either 8 differential channels numbered Channel 0 through Channel 7 or 16 single-ended channels numbered Channel 0 through Channel 15. The A/D converter block can automatically read channels in sequence from the Low Channel Number (set with A/D Low Channel Register) to the High Channel Number (set with A/D High Channel Register). This is referred to as channel scanning.

Channel Scanning

The A/D block contains an auto-incrementing channel counter. The channel counter is loaded with the value of the A/D Low Channel register bit field ADLOW3-0 whenever that register is written. On each A/D conversion start, the channel counter increments by 1. When the counter is equal to the value of A/D High Channel register bit field ADHIGH3-0, on the next A/D conversion start it will reload with ADLOW3-0. On power-up/reset, the channel counter, ADLOW3-0, and ADHIGH3-0 are cleared to 0. When A/D Advanced Configuration Register bit field ADSEDI=1, the block configuration is set to handle 8 differential inputs and the channel counter is limited to the range 0-7, and register bit fields ADLOW3 and ADHIGH3 are cleared to 0.

Channel Range (Gain) Setting

The A/D block contains an internal 16x4 bit channel/range table. The table can be programmed with an individual 4-bit A/D range for each of the 16 channels. The range value determines gain for the programmable gain amplifier. When writing to the A/D Configuration Register, if RTLOAD=1, the table entry selected by ADLOW3-0 will be programmed with the values in fields ADPOL, ADRANGE, and ADG1 and ADG0. This thus dynamically programs the characteristics when a channel is selected.

4.2 D/A Block

Capabilities and architecture

The D/A functional block provides sixteen 16-bit analog outputs with programmable range and polarity

The D/A block contains a single D/A converter for each of the 16 output channels. The converter internally divides the 16 channels into two sets of 8 channels. The individual channels are brought out at the FeaturePak connector on signal lines Vout0 through 15. The lines are physically spaced with empty adjacent lines in order to improve noise immunity.

Autocalibration

The Diamond FP-DAQ1616 board provides autocalibration capability as follows:

The channels are fed into the calibration multiplexer where TrimDAC channels are used to calibrate them. Both coarse and fine adjustments are used for the low as well as high reference. No potentiometers, which are subject to tampering, vibration, and maladjustment, are used. Instead, all calibration adjustments are performed using a 12-channel TrimDAC and precision, low-drift reference voltages on the board.

The optimum TrimDAC values for each output range are stored in the on-board EEPROM where they are retained through power-off and recalled automatically on power up. The board always boots up from factory calibration in 10V BIPOLAR mode on D/A channels.

Further details of calibration are given in Chapter 13.

4.3 DIO Block

The FP-DAQ1616 has seven 8-bit bidirectional digital I/O ports, named A-F and AUX. The DIO block is controlled and configured using registers at BAR0+32 to 47.

Ports A, B, C, and D are 8-bit ports with direction programmable byte by byte. Register bits DIRA-D control the direction of these ports and also the direction of the port pins where a value of 0 = input and 1 = output. In output mode, the values in these registers drive their associated I/O pins. The logic levels on the I/O pins may be read back in both input and output modes. These ports reset to 0 and input mode during power-up, reset, FPGARST=1, or BRDRST=1. If a port is in input mode, its output register may still be written to. When the port is switched to output mode, the value of the output register will drive the corresponding I/O pins.

Ports A, B, C, and D may operate in latched mode. This mode enables handshaking signals to control the transfer of data between the board and an external device. Latch mode is enabled for a port by setting its MODEn field = 1 (where n = A, B, C, or D). All ports with their MODEn = 1 operate in the same manner. To avoid undefined or undesired behavior, all DIO ports operating in latch mode should have the same direction setting.

Ports E-F are 8-bit ports with direction programmable bit by bit according to register bits DIRE7-0 and DIRF7-0 where a value of 0 = input, 1 = output. I/O pins DIOF3-0 may be reassigned as PWM outputs; see the PWM block description.

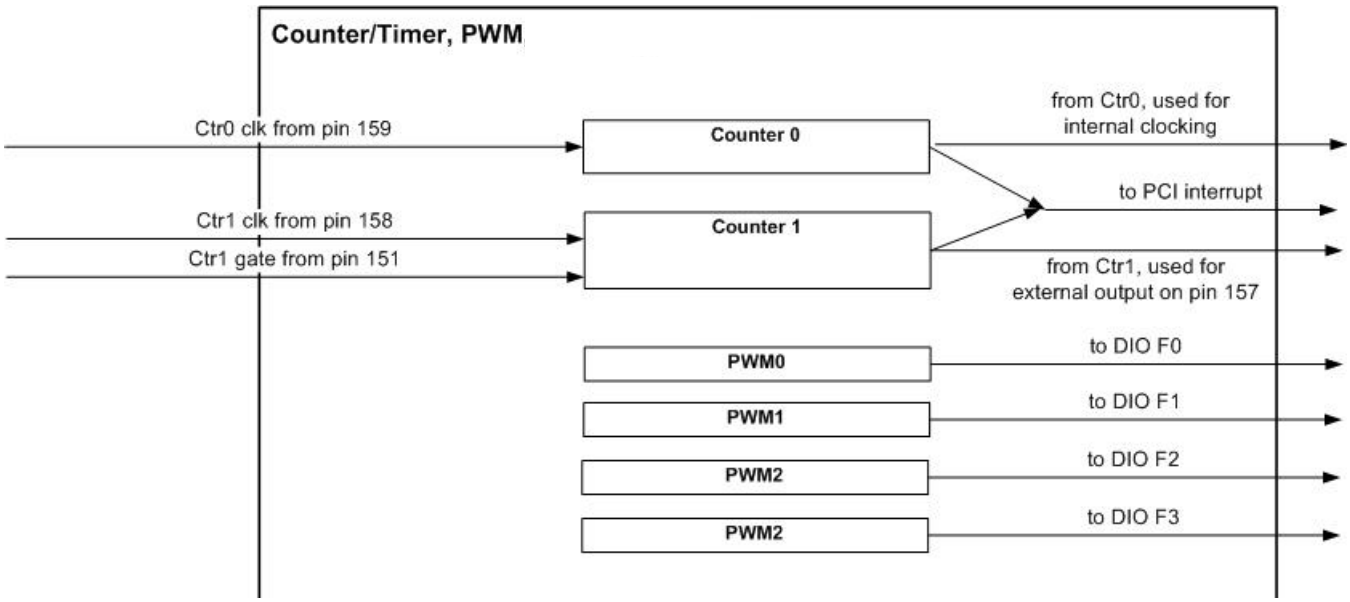
I/O pins DIOF6-7 may be reassigned as latch and acknowledge signals. When any MODEn bit is 1, signal line DIOF6 is forced to input mode and operates as a latch signal LATCH-, and register bit DIOF6 reads as 0. Signal line DIOF7 is forced to output mode and operates as an acknowledge signal ACK-, and register bit DIOF7 reads as 0.

Port AUX is an 8 bit port with direction programmable bit by bit. Register bits AUXDIR7-0 control the direction of bits AUX7-0 (0 = input and 1 = output). These bits reset to 0 and input mode during power-up, reset, FPGARST=1, or BRDRST=1. If a bit is in input mode, its output register may still be written to. When the port is switched to output mode, the value of the output register will drive the output pins.

AUX7-4 have dual functions defined by additional control register bits. When these bits are 0, the corresponding I/O pins are normal I/O pins and behave as described above. When these bits are 1, the corresponding I/O pins change to their alternate functions: AUX7 = WDTIn, AUX6 = WDTOut, AUX5 = Ctr1Gate, AUX4 = ADGate.

Details of register programming for the DIO block are given in section 6.2.3 and Chapter 14.

4.4 Counter Timer & PWM Block



Counters

The block contains two 32-bit up/down counter timers with programmable functions set by three registers in the register set at BAR0+48 for this block. The counters can be used to generate interrupts on the PCI Express bus. For details see sections on Interrupt registers and Chapter 15.

PWM

There are four available independent 24-bit PWM channels. Period and duty cycle may be set for each one.

4.5 Interrupts

Interrupts to the PCI Express Bus can be generated by any of these blocks: A/D, DIO, or the two counter timers.

Register control is provided for enabling/disabling interrupts

4.6 FIFO

The FIFO theory of operation is described in Chapter 9.

5. BOARD CONFIGURATION

The jumperless FP-DAQ1616 FeaturePak module is configured by software. The board must first be initialized, then configured. These operations can be done either using Diamond's Universal Driver (version 7.0 or higher) or by an independent set of equivalent register operations.

5.1 Configuring Using Universal Driver

Diamond Systems provides a device driver which will enable access to the board functionalities via an easy to use API set. This driver is called the Universal Driver and is available in Windows XP and Linux 2.6.xx operating systems. The details on the Universal driver can be found in the Universal Driver manual and can be accessed online at http://docs.diamondsystems.com/dscud/manual_Main+Page.html.

5.2 Configuring Using Register Operations

The board can also be controlled using simple register read/write commands if you are willing to write your own driver. In typical modern operating systems, the user level applications cannot directly access the low level system information and don't have register level access. In order to communicate with any PCI device, a device driver is required.

The Universal Driver mentioned above can be also be used to do register-level control, and a programmer could develop his own driver functionality that uses simple register read/write command after performing a PCI scan using the Universal Driver. Users of this type of access need to understand the board register map which is defined in later sections of this manual. This type of approach is suitable for someone who is very aware of the nature of low-level operations of hardware.

5.2.1 Interrupt level

Interrupts are used for hardware I/O operations that are independent of normal program flow. The Diamond FP-DAQ1616 can be set up to generate interrupts under several circumstances. The most common use of interrupts is to transfer A/D data from the board to system memory during high-speed A/D sampling. The board can also generate interrupts to transfer digital data into the board, as well as at regular intervals according to a programmable timer on the board. Individual control bits are used to enable each type of interrupt.

Since the FP-DAQ1616 board works on PCI Express bus architecture, the interrupt level is obtained as a result of a PCI scan performed by the device driver. To obtain the interrupt level used by the board, DSC provides a default device driver which can perform low level PCI commands and provide user level access to the board. The driver is from a third-party driver developer and is called WinDriver.

If you do not wish to use this driver and would like to develop your own driver, you need to be knowledgeable on the PCI / PCI express system architecture as well as the device driver model and architecture details for your chosen operating system.

5.2.2 Single-Ended / Differential A/D Channels

The input channels on the FP-DAQ1616 can be configured as 16 single-ended or 8 differential channels. A **single-ended input** is a single-wire input (plus ground) that is measured with reference to the board's analog ground. In order for the measurement to be accurate, the board's ground must be at the same potential as the source signal's ground. Usually this is accomplished by connecting the two grounds together at some point, for example by connecting to one of the analog ground pins on the FeaturePak connector.

A **differential input** is a two-wire input (plus ground) that is measured by subtracting the low input from the high input. This type of connection offers two advantages: It allows for greater noise immunity, because the noise, which is present in equal amounts and equal phase on both the high and low inputs, is subtracted out when the low input is subtracted from the high input; and it allows for the signal to float away from ground. Normally the ground of the signal source is still connected to the ground on the A/D board in order to keep the signal from straying out of the common mode range of the A/D board's input circuitry.

Configuration is done by programming the appropriate register bits.

5.2.3 D/A Configuration

The FP-DAQ1616 provides 16 analog outputs. They exist in two 8-output groups. Each group has its own reference inputs and range selection capability. Offset and gain can be adjusted for each group as a whole.

The outputs can be set individually or all at once to operate in bipolar (both + and –) or unipolar (+ only) output voltage ranges with the full-scale output range set to 5V, 10V, or programmable. The maximum output current on any channel is 5mA. Current outputs such as 0-20mA outputs are not supported.

On power-up, the DACs are configured to reset to mid-scale (0V in bipolar mode) or zero scale (0V in unipolar mode) so that the DACs power up to 0V. Default autocalbration settings assume they are in 10V bipolar mode.

6. I/O REGISTERS

The FP-DAQ1616 register map consists of 256 bytes, divided into 16 16-byte blocks in the system I/O address space. Direct register access is not required if you are using Diamond's Universal Driver software that ships with the board. The driver handles all board access and provides a high-level set of functions to simplify programming. The information presented here and in the next chapter is intended to provide a detailed description of the board's features and operation, as well as for programmers who are not using the Universal Driver software.

6.1 I/O Map Summary

Block (Dec)	Range (Dec)	Range (Hex)	Function
0	0-15	0-F	A/D control and data
1	16-31	10-1F	D/A control and data
2	32-47	20-2F	Digital I/O
3	48-63	30-3F	Counters, PWM
4	64-79	40-4F	FIFO
5	80-95	50-5F	EEPROM / Calibration
6	96-111	60-6F	do not use
7	112-127	70-7F	Interrupts, Misc., and ID
8	128-143	80-8F	do not use
9	144-159	90-9F	do not use
10	160-175	A0-AF	do not use
11	176-191	B0-BF	do not use
12	192-207	C0-CF	do not use
13	208-223	D0-DF	do not use
14	224-239	E0-EF	SPI Flash Interface
15	240-255	F0-FF	Capabilities / Page Control

The FP-DAQ1616 board is a PCI Express based design and has 255 bytes of addressable registers as shown in the table above. Since in PCI architecture, the I/O base address is obtained as a result of a PCI scan on the Vendor ID and Device ID of the PCI device, the base address is not shown in the table; instead, offsets from the base address BAR0 are shown for individual functional blocks.

The FP-DAQ1616 Vendor ID is 0x1C0E and the Device ID is 0x0800. When a PCI scan is performed using these two IDs, the board is detected and the register BAR0 gets the base address for the start of where registers are mapped. For the user application, the address information of BAR0 serves as the base address of the board and the rest of the registers are all accessed as an offset from the BAR0 address.

In the following material, register block addresses are shown in a format as in this example:

BAR0 + 0 (0x00) Read/Write/Command A/D Block Registers

Where the address is given as BAR0 + offset amount with the number in decimal, and the hex equivalent in parentheses. This is followed by the register block group name.

Registers are Read, Read/Write, Write, or Command, where Command registers can be written but cannot be read from.

6.2 I/O Register Details

This section describes the location and general behavior of specific bits in each I/O map register. In all register definitions below, any bit named X is not defined and serves no function.

6.2.1 BAR0 + 0 (0x00) A/D Block Registers

Write

Offset from Block Base (Dec)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	ADRESET							ADSTART
1							ADRES	ADTURBO
2								
3								
4					ADLOW3	ADLOW2	ADLOW1	ADLOW0
5					ADHIGH3	ADHIGH2	ADHIGH1	ADHIGH0
6								
7	Reserved for A/D channel expansion							
8	RTEN	RTLOAD			ADPOL	ADRANGE	ADG1	ADG0
9								ADSEDI
10			ADGATEN	ADCLK1	ADCLK0	SCANINT1	SCANINT0	SCANEN
11	PSCN7	PSCN6	PSCN5	PSCN4	PSCN3	PSCN2	PSCN1	PSCN0
12	AUX7-0							
13	AUXDIR7-0							
14								
15								

Read

Offset from Block Base (Dec)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 2	Bit 0
0	AD7-0							
1	AD15-8							
2	Reserved for AD23-16							
3	ADBUSY	ADWAIT						
4					ADLOW3	ADLOW2	ADLOW1	ADLOW0
5					ADHIGH3	ADHIGH2	ADHIGH1	ADHIGH0
6					ADCH3	ADCH2	ADCH1	ADCH0
7	Reserved for A/D channel expansion							
8	RTEN				ADPOL	ADRANGE	ADG1	ADG0
9								ADSEDI
10			ADGATEN	ADCLK1	ADCLK0	SCANINT1	SCANINT0	SCANEN
11								
12	AUX7 or WDTin	AUX6 or WDTOut	AUX5 or Ctr1Gate	AUX4 or ADGate	AUX3	AUX2	AUX1	AUX0
13	AUXDIR7-0							
14								
15								

BAR0 + 0 (0x00) Write A/D Command Register

Bit No.	7	6	5	4	3	2	1	0
Name	ADRESET	X	X	X	X	X	X	ADSTART
Reset value	0	X	X	X	X	X	X	0

ADRESET

This bit allows the A/D circuit to be reset. When the A/D circuit is reset, the FPGA controller resets the entire AD related hardware and internal registers. A reset is issued to the A/D chip. The FPGA also resets the A/D mode to the default power up A/D mode and resets the A/D FIFO as well as FIFO flags.

1 = Perform AD circuit reset as described above.

0 = NO ACTION. (Default setting)

ADSTART

This bit provides the A/D conversion signal to the A/D chip. This bit when set to 1, initiates an A/D conversion and the results are stored in the on board FIFO registers. If the FIFO is disabled, only the last conversion is stored in the internal memory which can be read back by software as explained later. This bit is a command bit and needs to be set every time an AD conversion is required.

1 = Initiate A/D conversion.

0 = NO ACTION. (Default setting)

The FPGA can trigger the A/D conversion in multiple ways, one of which is using the ADSTART bit. This is the default setting of the FPGA. The A/D conversion clock can be selected by the register at offset 9 in the A/D block and the values ADCLK1-0 determine the clock that the FPGA will use to perform the A/D conversion. The default power up value of ADCLK1-0 is “00” which indicates that the clock is provided by setting the ADSTART bit to 1. This is explained in details in the next sections.

BAR0 + 0 (0x00) Read A/D LSB

Bit No.	7	6	5	4	3	2	1	0
Name	AD7-0							
Reset value	0x00							

AD7-0

These bits represent the lower 8 bits of the A/D value. If the FIFO is not empty, this register returns the LSB of the A/D value stored at the current FIFO pointer. If the FIFO is empty, reading from this register returns 0.

BAR0 + 1 (0x01) Write A/D Resolution and A/D Turbo Registers

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	X	ADRES	ADTURBO
Reset value							0	0

ADRES

This is a command bit and cannot be read-back. This bit sets the resolution of the A/D converter to either 12-bit or 16-bit resolution.

1 = When this bit is set to 1, the A/D circuit operates with a resolution of 16-bits.

0 = When this bit is set to 0 (Default setting), the A/D circuit operates with a resolution of 12-bits.

ADTURBO

This is a command bit and cannot be read-back. This bit sets the overall bandwidth of the A/D conversion. The FP-DAQ1616 can be set to operate at either 2.0MHz or at 1.67MHz maximum sampling rate.

1 = When this bit is set to 1, the A/D circuit operates at a maximum of 2.0MHz sample rate.

0 = When this bit is set to 0 (Default setting), the A/D circuit operates at a maximum of 1.67Hz sample rate.

BAR0 + 1 (0x01) Read A/D MSB

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	AD11-8			
Reset value	0x00							

AD11-8 If the FIFO is not empty, this register returns the MSB (upper 4 bits) of the A/D value stored at the current FIFO pointer and decrements the FIFO depth value by one sample. If the FIFO is empty, reading from this register returns 0.

BAR0 + 3 (0x03) Read A/D Status Register

Bit No.	7	6	5	4	3	2	1	0
Name	ADBUSY	ADWAIT	X	X	X	X	X	X
Reset value	0	0						

ADBUSY This bit provides the status of the A/D conversion. The user is expected to read this bit to learn that the A/D chip is IDLE or busy. After starting a conversion in software, the program must monitor ADBUSY and wait for it to become 0 prior to reading A/D values. If SCANEN = 0 (single conversion mode), ADBUSY goes high when an A/D conversion is started and stays high until the conversion is finished. If SCANEN = 1 (scan mode enabled), ADBUSY stays high during the entire scan.

1 = A/D Conversion or scan in progress.

0 = A/D is IDLE.

ADWAIT This bit provides A/D input circuit status. ADWAIT goes high after the channel register (offset 4 or 5) or the gain register (offset 8&9) is changed. It stays high for the time interval specified by SCANINT1-0 bits. The program should monitor this bit after writing to either register, and wait for it to become 0 prior to starting an A/D conversion.

1 = A/D circuit is settling on a new value.

0 = A/D is idle.

BAR0 + 4 (0x04) Read/Write A/D Low Channel Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	ADLOW3-0			
Reset value					0	0	0	0

ADLOW3-0 This register holds the low channel number of scan range for A/D conversion. When this register is written to, the ADWAIT bit goes high for the SCANINT1-0 time interval to indicate that the A/D circuit is settling with the new channel value.

The A/D circuit is designed to automatically increment the A/D channel every time an A/D conversion is triggered. This enables the user to avoid having to write the A/D channel each time. The A/D channel will rotate through the values between ADLOW3-0 and ADHIGH3-0. When channel ADHIGH3-0 is sampled, the register resets to ADLOW3-0.

Reading from this register returns the value previously written to it.

Valid values: 0 – 15.

NOTE: Whenever this register is written to, the ADWAIT bit goes high for a time period defined by the bits SCANINT1-0.

BAR0 + 5 (0x05) Read/Write A/D High Channel Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	ADHIGH3-0			
Reset value					0	0	0	0

ADHIGH3-0

Similar to the previous register, this register holds the high A/D channel value. The A/D block rotates through the channels up to the value held in this register. Note that even though the A/D block rotates through the channels from the ADLOW3-0 value to the ADHIGH3-0 value, it is not required that the value in this register be greater than the value in the ADLOW3-0 register.

If ADLOW3-0 < ADHIGH3-0 :
In this case, the A/D block will rotate through the channel number specified by ADLOW3-0 up to the channel number specified in ADHIGH3-0 and then stops.

For example: ADLOW3-0 = 0 and ADHIGH3-0 = 4 then the channels 0, 1, 2, 3 and 4 will be sampled and the conversions will stop.

If ADLOW3-0 > ADHIGH 3-0:

In this case, the A/D block counts up to 16 and then wraps around from channel 0 until it reaches the channel number specified by bit field ADHIGH3-0.

For example: ADLOW3-0 = 13 and ADHIGH3-0 = 4 then the channels 13, 14, 15, 0, 1, 2, 3, and 4 will be sampled in the order specified.

Valid values for this register are 0 – 15.

NOTE: Whenever this register is written to, the ADWAIT bit goes high for a time period defined by the bits SCANINT1-0.

BAR0 + 6 (0x05) Read A/D Current Channel Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	ADCH3	ADCH2	ADCH1	ADCH0
Reset value					0	0	0	0

ADCH3-0

This register provides the value of the current A/D channel counter. This value is incremented at the start of each A/D conversion. Because of this increment, this value always represents the channel value of the next A/D conversion channel.

Valid values are from 0-15.

BAR0 + 8 (0x08) Write A/D Configuration Register

Bit No.	7	6	5	4	3	2	1	0
Name	RTEN	RTLOAD	X	X	ADPOL	ADRANGE	ADG1	ADG0
Reset value	0	0			0	0	0	0

This register controls the parameters used to configure analog input channels, and some channel scanning characteristics using the A/D Low Channel and A/D High Channel registers.

RTEN

The RTEN bit provides access to an A/D block table which is 4 bit wide and 16 bytes long. This buffer holds unique polarity/range and gain values (just as the bits 3-0 in this register) for each of the 16 A/D channels. Each of the 16 bytes represents a channel number starting from channel 0 at location 0 and so on. The channel number for using the RTEN is specified in the ADLOW3-0 register.

1 = Enables the A/D circuit to load the polarity, range and gain values for the current channel from the internal channel/range table memory. The internal range/gain table memory buffer is explained on page 21 and in Chapter 7 of this document.

0 = Disables the internal table usage. All A/D channels have the same setting for AD polarity, range and gain as specified by the bits 0 to 3 in this register.

RTLOAD

This is a command bit which enables loading of the internal 16X4 bit A/D settings table as described above.

1 = When set to 1, the values of ADPOL, ADRANGE, ADG1 and ADG0 will be stored in the internal table at the offset which will be specified in the ADLOW3-0 register.

0 = The internal A/D settings table is not addressed or accessed. This is the default mode of operation.

ADPOL

This bit sets the polarity of the A/D conversion. The FP-DAQ1616 board's A/D channels can be set to operate in either Bipolar or Unipolar modes.

1 = Unipolar operation.
0 = Bipolar operation (Default setting).

ADRANGE

Provides the input range for the A/D circuit.

1 = 10V range.
0 = 5V range (Default setting).

ADG1-0

Analog input gain. The gain is the ratio of the voltage seen by the A/D converter and the voltage applied to the input pin. The gain setting is the same for all input channels when RTEN=0. When RTEN=1, the gain value and the other values like ADPOL and ADRANGE are read from the corresponding entry in the internal A/D settings table.

The following table describes the different gain values.

ADG 1	ADG 0	Gain
0	0	1 (Default setting)
0	1	2
1	0	4
1	1	8

This table lists the available analog input ranges selectable by the four configuration bits.

ADPOL	ADRANGE	ADG1	ADG0	DESCRIPTION
0	0	0	0	Bipolar input, +/- 5V range (Default setting)
0	0	0	1	Bipolar input, +/- 2.5V range
0	0	1	0	Bipolar input, +/- 1.25V range
0	0	1	1	Bipolar input, +/- 0.625V range
0	1	0	0	Bipolar input, +/- 10V range
0	1	0	1	Bipolar input, +/- 5V range
0	1	1	0	Bipolar input, +/- 2.5V range
0	1	1	1	Bipolar input, +/- 1.25V range
1	0	0	0	Unipolar input, 5V range
1	0	0	1	Unipolar input, 2.5V range
1	0	1	0	Unipolar input, 1.25V range
1	0	1	1	Unipolar input, 0.625V range
1	1	0	0	Unipolar input, 10V range
1	1	0	1	Unipolar input, 5V range
1	1	1	0	Unipolar input, 2.5V range
1	1	1	1	Unipolar input, 1.25V range

NOTE: Whenever this register is written to, the ADWAIT bit goes high for a time period defined by the bits SCANINT1-0.

BAR0 + 8 (0x08) Read A/D Configuration Register

Bit No.	7	6	5	4	3	2	1	0
Name	RTEN	X	X	X	ADPOL	ADRANGE	ADG1	ADG0
Reset value	0				0	0	0	0

RTEN — See description under Write mode above.

1 = Read the ADPOL, ADRANGE, ADG1 and ADG0 values from the internal A/D settings table and load the values on the A/D chip.

0 = The A/D settings of ADPOL, ADRANGE, ADG1 and ADG0 values are configured on the A/D chip based on the values of bits 0-3 in the register.

ADPOL — Read back of the A/D polarity setting.

1 = A/D is set in UNIPOLAR mode

0 = A/D is set in BIPOLAR mode. (Default setting)

ADRANGE — Read back of the A/D range setting.

1 = A/D is operating in 10V range.

0 = A/D is operating in 5V range. (Default setting)

ADG1-0 — Read back of the current gain setting of the A/D circuit.

ADG1	ADG0	Gain
0	0	1 (Default Setting)
0	1	2
1	0	4
1	1	8

BAR0 + 9 (0x09) Read/Write A/D Advanced Configuration Register

Bit No.	7	6	5	4	3	2	1	0
Name								ADSEDI
Reset value								0

ADSEDI — A/D single ended / differential configuration.

1 = Differential configuration (# of A/D channels = 8, numbered from 0 to 7.

0 = Single ended configuration (# of A/D channels = 16, numbered from 0 to 15. (Default configuration)

NOTE: Whenever this register is written to, the ADWAIT bit goes high for a time period defined by the bits SCANINT1-0.

BAR0 + 10 (0x0A) Read/Write A/D Scan Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	ADGATEN	ADCLK1	ADCLK0	SCANINT1	SCANINT0	SCANEN
Reset value			0	0	0	0	0	0

ADGATEN This bit controls the A/D gate feature of the board. When this bit is set to 1, A/D samples will only occur when bit AUX4 (bit 4 of DIO port F) is high.

1 = Enabled A/D gating.

0 = Disable A/D gating. (Default setting)

ADCLK1-0 A/D clock setting. The settings in these bits define how the on-board logic clocks the A/D circuit for performing A/D conversions. The table below describes different A/D clocking mechanisms available on the FP-DAQ1616 board.

ADCLK1	ADCLK0	A/D clock source
0	0	Command bit ADSTART at offset 0 (Default Setting)
0	1	Falling edge of External trigger (ADGATE/AUX4 signal line, pin 160 of FeaturePak connector)
1	0	Rising edge of output of counter 0 (Used for Interrupts)
1	1	Rising edge of output of counter 1

SCANINT1-0 Scan interval setting. The time interval specified by these bits is the time between A/D samples when performing an A/D scan (SCANEN = 1). The scan interval time is as in the table below.

SCANINT1	SCANINT0	Scan Interval	Resulting Max Sample Rate
0	0	500nS	2 MHz (Default Setting)
0	1	1000nS	1.00MHz
1	0	1333nS	750KHz
1	1	Programmable	(1/Resulting Scan Rate) – See description of BAR0 + 11

SCANEN Enables/Disables SCAN mode of A/D sampling.

1 = An A/D conversion sequence will occur once for each channel in the range ADLOW3-0 to ADHIGH3-0 with a time interval between conversions defined by SCANINT1-0. The ADBUSY bit stays high during the entire scan.

0 = Scan mode disabled. Each A/D trigger causes the board to generate a single A/D conversion on the current channel. The internal channel pointer will increment to the next channel in the range LOW – HIGH or reset to LOW if the current channel is HIGH. The ADBUSY bit stays high during each A/D conversion.

When the SCANINT1 and SCANINT0 are both ‘1’, the scan interval is not a fixed value and is user programmable. The value used is determined by the register BAR0+11. Users are encouraged to use the programmable scan interval timings to determine the best possible interval for their application.

BAR0 + 11 (0x0B) Read/Write, as noted A/D Programmable Scan Interval Configuration

Bit No.	7	6	5	4	3	2	1	0
Name (R/W)	PSCN7	PSCN6	PSCN5	PSCN4	PSCN3	PSCN2	PSCN1	PSCN0
Reset value	0	0	0	0	0	0	0	0

This register determines the scan interval that will be used when the SCANINT1-SCANINT0 bits in the register BAR0+10 are set to 11 marking the scan interval as user programmable.

When the user programmable mode is used, the value in this register is used in the formula below to calculate the resulting scan interval.

Scan Interval = (Register value) * 20nS.

The table below provides a few examples of the scan interval values with respect to the register values.

Register Value	Scan Interval Value	Resulting Max Sample Rate
0	INVALID	INVALID
25 (0x19)	500 nS	2.00 MHz
50 (0x32)	1000 nS	1.00 MHz
128 (0x80)	2560 nS	390 KHz
255 (0xFF)	5100 nS	196 KHz

NOTE: When programmable scan interval is used, the register value should be always greater than 25 as with the value of 25, the resulting scan rate will come out to be the maximum that the board can support. Any value below 25 will result in unpredictable A/D readings.

BAR0 + 12 (0x0C) Read/Write, as noted A/D Auxiliary Register

Bit No.	7	6	5	4	3	2	1	0
Name (R/W)	AUX7	AUX6	AUX5	AUX4	AUX3	AUX2	AUX1	AUX0
Reset value	0	0	0	0	0	0	0	0
Name (R)	WDTIn	WDTOut	CtrlGate	ADGate				
Reset value								

Some bits in this register have dual functionality and whose readability/writeability depends on the overall mode setting for the register.

AUX7-0

This controls the 8 bit DIO port F when it is used as the AUX port with each bit having programmable direction control using the A/D Auxiliary Control Register at BAR0+13 and bits AUXDIR7-0.

On power up, the port is an input port. The data read from this register represents the state of the lines on the AUX port.

The bits AUX4-7 have dual functions as defined by additional control bits. When these bits are 0, the corresponding I/O pins act as regular DIO pins. When any of these bits are 1, they have an alternate function which is controlled by other control bits as defined by table at the top of the next page.

AUX Bit	Alternate Signal	Function	Direction	Control Bit
7	WDTIN	Watchdog timer in	In	WDTIEN
6	WDTOUT	Watchdog timer out	Out	WDTOEN
5	CTR1GATE	Counter 1 gate input	In	CTRCMD3-0=0011 & CCD0=1
4	ADGATE	A/D gate input	In	ADGATEN

BAR0 + 12 (0x0C) Read A/D Auxiliary Register

Bit No.	7	6	5	4	3	2	1	0
Name	AUX7	AUX6	AUX5	AUX4	AUX3	AUX2	AUX1	AUX0
Reset value	0	0	0	0	0	0	0	0

AUX7-0 When this register is read, it provides the read back of the AUX port bits 0-7. If the corresponding bit on the port is LOW, the bit would be 0 where as 1 represents HIGH level on the AUX pin.

BAR0 + 13 (0x0D) Read/Write A/D Auxiliary Control Register

Bit No.	7	6	5	4	3	2	1	0
Name	AUXDIR7	AUXDIR6	AUXDIR5	AUXDIR4	AUXDIR3	AUXDIR2	AUXDIR1	AUXDIR0
Reset value	0	0	0	0	0	0	0	0

AUXDIR7-0 Provides direction control of the AUX DIO port. Each bit provides direction control for the corresponding AUX port bit.

1 = Sets the corresponding AUX port bit as an output bit.

0 = Sets the corresponding AUX port bit as an input bit. (Default setting)

6.2.2 BAR0 + 16 (0x10) D/A Block Registers

Offset from Block Base (Dec)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 2	Bit 0
0	DA7-0							
1	DA15-8							
2								
3								
4 (W)	DACMD7-0							
5					DABG1	DABG0	DAAG1	DAAG0
6							DAB2C	DASIM
7 (C)	DARESET					DAUPDT	DACLR1	DACLR0
7 (R)	DABUSY	DAOVF						
8								
9	DACA7	DACA6	DACA5	DACA4	DACA3	DACA2	DACA1	DACA0
10	DAWCH3	DAWCH2	DAWCH1	DAWCH0			DACA9	DACA8
11	DEPTH3	DEPTH2	DEPTH1	DEPTH0	WGCH1	WGCH0	WGSRC1	WGSRC0
12					WGINC	WGRST	WGPS	WGSTRT
13								
14								
15								

Note: (W) means write-only, (R) means read-only, (C) means write-only command bits

BAR0 + 16 (0x10) Write D/A LSB Register

Bit No.	7	6	5	4	3	2	1	0
Name	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
Reset value	0	0	0	0	0	0	0	0

DA7-0 D/A data bits 7 - 0. This register stores the DA LSB of the 16 bit D/A data.

BAR0 + 17 (0x11) Write D/A MSB Register

Bit No.	7	6	5	4	3	2	1	0
Name	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
Reset value	0	0	0	0	0	0	0	0

DA15-8 D/A data bits 15 - 8. This register stores the DA MSB of the 16 bit D/A data.

BAR0 + 20 (0x14) Write D/A Command Register

Bit No.	7	6	5	4	3	2	1	0
Name	DACMD7	DACMD6	DACMD5	DACMD4	DACMD3	DACMD2	DACMD1	DACMD1
Reset value	0	0	0	0	0	0	0	0

DACMD7-0 D/A command register. The value written to this register is sent to the Analog Devices AD5360 D/A converter. The commands are described in detail in the AD5360 device manual available from that vendor. This register must be written to with the appropriate command for every operation to be performed on the D/A converter. This register along with the two data registers above provide complete access to the D/A converter.

The D/A commands are as below.

DACMD7-0	Command Description
0x02	Set Offset to Channel Group 1
0x03	Set Offset on Channel Group 2
0xC8-0xCF	Output D/A code on D/A channel 0 to 7
0xD0-0xD7	Output D/A code on D/A channel 8 to 15

BAR0 + 21 (0x15) Read/Write D/A Gain Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	DABG1	DABG0	DAAG1	DAAG0
Reset value					0	0	0	0

DABG1-0 D/A gain settings for D/A channel group 2. The values in these bits apply to all the D/A channels in the D/A channel group 2 (D/A channels 8-15).

DAAG1-0 D/A gain settings for D/A channel group 1. The values in these bits apply to all the D/A channels in the D/A channel group 1 (DA channels 0-7).

The following table describes the different gain values (x=A or B).

DAxG1	DAxG0	Gain
0	0	1 (Default Setting)
0	1	2
1	0	4
1	1	8

When this register is read, the bits DAAG1-0 and DABG1-0 return the gain values previously written.

BAR0 + 22 (0x16) Read/Write D/A format register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	X	DAB2C	DASIM
Reset value							0	0

DAB2C Binary or 2's complement data. This bit tells the D/A converter how to treat the data provided to it via the DA0-15 register. The data can be either treated as pure binary data or as 2's complement data.

NOTE: It is the responsibility of the user to make sure that the data is provided in the required format.

1 = 2's complement format

0 = Binary format. (Default setting)

DASIM — D/A simultaneous update control. This bit determines when the D/A output is updated.

1 = DASIM enabled. D/A channel will not be updated until the bit DAUPDT in the register at offset 23 is set to 1. Until the DAUPDT bit is set to 1, the D/A data is maintained in an internal buffer and not transferred to the D/A channels. Once the DAUPDT bit is set to1, the data is simultaneously transferred to all the 16 channels.

0 = DASIM is disabled. (Default setting) The D/A channel will be updated with the data in DA0-15 as soon as the DACMD register is updated with the appropriate command as described in the register above.

When this register is read, the bits DAB2C and DASIM return the value previously set.

BAR0 + 23 (0x17) Write D/A control register

Bit No.	7	6	5	4	3	2	1	0
Name	DARESET	X	X	X	X	DAUPDT	DACLR1	DACLR0
Reset value	0					0	0	0

This is a command register and none of the values are available for read back. When any of the bits in this register are written to, the board's FPGA takes action to perform the required commands as described below.

DARESET — D/A Reset bit.

1 = Perform a reset of the D/A block. When D/A block is reset, all the D/A registers are defaulted to their power up state. All the D/A outputs are reset to the power up state which is at 0V.

0 = NO ACTION

DAUPDT — D/A update control. This bit is required when DASIM is enabled in the register at offset 22 as explained above.

1 = Update all D/A channels simultaneously. This command affects the output of the D/A channels only when DASIM = 1.

0 = NO ACTION.

DACLR1 — D/A channel group 1 clear command

1 = When set to 1, this bit clears the outputs of all the D/A channels in channel group 2. Thus D/A channels 8-15 are cleared to 0V output when this bit is set.

0 = NO ACTION.

DACLR0 — D/A channel group 0 clear command

1 = When set to 1, this bit clears the outputs of all the D/A channels in channel group 1. Thus D/A channels 0-7 are cleared to 0V output when this bit is set.

0 = NO ACTION.

BAR0 + 24 (0x18) Read D/A Status Register

Bit No.	7	6	5	4	3	2	1	0
Name	DABUSY	DAOVF	X	X	X	X	X	X
Reset value	0	0						

This register provides status information on the D/A block.

DABUSY D/A busy indicator. Indicates that the D/A block is busy performing the previous command. Any new D/A command will be ignored when this bit is high. This bit should always be checked in the application program before writing any command to the D/A registers.

1 = DAC is busy.

0 = DAC is IDLE. D/A command can be executed.

DAOVF D/A overflow indicator. When a D/A command is sent to the D/A block when DABUSY = 1, this bit will be set to 1 indicating that the new command is being ignored.

1 = DABUSY = 1 and the last command is ignored. When read, the bit goes back to 0.

0 = D/A is IDLE.

BAR0 + 25 (0x19) Write D/A Waveform Address LSB

Bit No.	7	6	5	4	3	2	1	0
Name	DACA7	DACA6	DACA5	DACA4	DACA3	DACA2	DACA1	DACA0
Reset value	0	0	0	0	0	0	0	0

This register has the LSB of the DA waveform generator address.

DACA7-0 LSB of the 10 bit address to store D/A code in D/A waveform buffer.

BAR0 + 26 (0x20) Write D/A Waveform Address MSB

Bit No.	7	6	5	4	3	2	1	0
Name	DAWCH3	DAWCH2	DAWCH1	DAWCH0	X	X	DACA9	DACA8
Reset value	0	0	0	0	X	X	0	0

This register has the MSbits of the DA waveform generator address and the D/A channel on which the current addressed packet will be used.

DACA9-8 MSbits of the 10 bit address to store D/A code in D/A waveform buffer.

DAWCH3-0 D/A waveform channel on which the D/A code addressed by the address bits DACA9-0 will be applied to.

BAR0 + 27 (0x21) Write D/A Waveform Configuration

Bit No.	7	6	5	4	3	2	1	0
Name	DEPTH3	DEPTH2	DEPTH1	DEPTH0	WGCH1	WGCH0	WGSRC1	WGSRC0
Reset value	0	0	0	0	X	X	0	0

This register provides the configuration options to use the D/A waveform generator.

WGSRC1-0 The bits WGSRC1-0 combine to provide a choice for the trigger source that is used to increment the waveform by one frame. The bits are defined as below.

WGSRC1	WGSRC0	**Description**
0	0	Manual (using WGINC)
0	1	Counter 0 output
1	0	Counter 1 output
1	1	External trigger (AUX0/WGTrig)

WGCH1-0 The bits WGCH1-0 combine to provide a selection on how many codes are output on each frame as described below.

WGCH1	WHCH0	**Description**
0	0	1 code per frame
0	1	2 codes per frame
1	X	4 codes per frame

DEPTH3-0 These bits define the size of the D/A waveform buffer. The depth is based on this equation:

Depth = [(DEPTH3-0) + 1] * 64

This allows valid depth values from 64 to 1024 samples.

The waveform generator frame pointer will return to 0 whenever it hits either 1024 or the depth value indicated above.

NOTE: A frame means one instance of the data packet stored in the internal DA waveform buffer memory. Each frame consists of a 16 bit D/A code and the channel on which the D/A code is sent out. Increment to the frame means increment to the waveform buffer memory location. Upon each increment, the number of codes assigned for the frame will be sent out. If the configuration is to send 1 code per frame, only one location will be sent out where as for 4 codes per frame, the logic will send out codes for the current location and also the next 3 locations in the waveform buffer.

BAR0 + 28 (0x22) Write D/A Waveform Command

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	WGINC	WGRST	WGPS	WGSTRT
Reset value	X	X	X	X	X	X	0	0

This register has various commands to control the waveform generator circuit. Only one bit can be set to '1' at once. Bits are processed MSB to LSB. The first '1' determines which command is carried out

WGSTRT Begin or resume the waveform generator.

1 = Begin or resume the waveform generator.

0 = NO ACTION

WGPS Pause/stop the waveform generator. The current position in memory is saved for the next begin/resume, or can be reset using WGRST.

1 = Pause or stop the waveform generator.

0 = NO ACTION

WGRST Reset the waveform generator to output from the beginning of the D/A code

Buffer.

1 = Reset the output

0 = NO ACTION

WGINC Force the waveform generator to increment one frame.

1 = Increment the waveform generator buffer by one frame with software command.

0 = NO ACTION

6.2.3 BAR0 + 32 (0x20) DIO Block Registers

These registers provide control of digital I/O ports. To use the ports, they must first be configured using Mode and Dir fields, and then can be read from and written to.

Offset from Block Base (Dec)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 2	Bit 0
0	DIOA7-0							
1	DIOB7-0							
2	DIOC7-0							
3	DIOD7-0							
4	DIOE7-0							
5	DIOF7-0							
6								
7								
8 (W)							MODEA	DIRA
8 (R)	LATCHA						MODEA	DIRA
9 (W)							MODEB	DIRB
9 (R)	LATCHB						MODEB	DIRB
10 (W)							MODEC	DIRC
10 (R)	LATCHC						MODEC	DIRC
11 (W)							MODED	DIRD
11 (R)	LATCHD						MODED	DIRD
12	DIRE7-0							
13	DIRF7-0							
14								
15	ACK							

BAR0 + 32 (0x20) Read/Write Digital I/O port A

Bit No.	7	6	5	4	3	2	1	0
Name	DIOA7-0							
Reset value	0	0	0	0	0	0	0	0

DIOA7-0 This register is used for digital I/O on byte addressable port A. When port A is in output mode, the output signal lines DIO A7-0 (see Section 3.4) will be set to the values in this register, and reading this register will read back the programmed value. When port A is in input mode, this register will read back the logic levels on signal lines DIO A7-0, and writing to this register will have no effect. The direction of port A is controlled by the bit DIRA in the DIO control register at BAR0+40.

The DIO port A defaults to input mode on power up and all the register bits reset to 0.

BAR0 + 33 (0x21) Read/Write Digital I/O port B

Bit No.	7	6	5	4	3	2	1	0
Name	DIOB7-0							
Reset value	0	0	0	0	0	0	0	0

DIOB7-0 This register is used for digital I/O on byte addressable port B. When port B is in output mode, the output signal lines DIO B7-0 (see Section 3.4) will be set to the values in this register, and reading this register will read back the programmed value. When port B is in input mode, this register will read back the logic levels on signal lines DIO B7-0, and writing to this register will have no effect. The direction of port B is controlled by bit DIRB in the DIO control register at BAR0+41.

The DIO port B defaults to input mode on power up and all the register bits reset to 0.

BAR0 + 34 (0x22) Read/Write Digital I/O port C

Bit No.	7	6	5	4	3	2	1	0
Name	DIOC7-0							
Reset value	0	0	0	0	0	0	0	0

DIOC7-0 This register is used for digital I/O on byte addressable port C. When port C is in output mode, the output signal lines DIO C7-0 (see Section 3.4) will be set to the values in this register, and reading this register will read back the programmed value. When port C is in input mode, this register will read back the logic levels on signal lines DIO C7-0, and writing to this register will have no effect. The direction of port C is controlled by bit DIRC in the DIO control register at BAR0+42.

The DIO port C defaults to input mode on power up and all the register bits reset to 0.

BAR0 + 35 (0x23) Read/Write Digital I/O port D

Bit No.	7	6	5	4	3	2	1	0
Name	DIOD7-0							
Reset value	0	0	0	0	0	0	0	0

DIOD7-0 This register is used for digital I/O on byte addressable port D. When port D is in output mode, the output signal lines DIO D7-0 (see Section 3.4) will be set to the values in this register, and reading this register will read back the programmed value. When port D is in input mode, this register will read back the logic levels on signal lines DIO D7-0, and writing to this register will have no effect. The direction of port D is controlled by bit DIRD in the DIO control register at BAR0+43.

The DIO port D defaults to input mode on power up and all the register bits reset to 0.

BAR0 + 36 (0x24) Read/Write Digital I/O port E

Bit No.	7	6	5	4	3	2	1	0
Name	DIOE7-0							
Reset value	0	0	0	0	0	0	0	0

DIOE7-0 This register is used for digital I/O on bit addressable port E. When port E is in output mode, the output signal lines DIO E7-0 on (see Section 3.4) will be set to the values in this register, and reading this register will read back the programmed value. When port E is in input mode, this register will read back the logic levels on signal lines DIO E7-0, and writing to this register will have no effect. The direction of port E is controlled by DIRE7-0 bits in the register at BAR0+44. Each bit in the direction register controls the direction of the corresponding port bit.

The DIO port E defaults to input mode on power up and all the register bits reset to 0.

BAR0 + 37 (0x25) Read/Write Digital I/O port F

Bit No.	7	6	5	4	3	2	1	0
Name	DIOF7-0							
Reset value	0	0	0	0	0	0	0	0

DIOF7-0 This register is used for digital I/O on bit addressable port F. When port F is in output mode, the output signal lines DIO F7-0 (see Section 3.4) will be set to the values in this register, and reading this register will read back the programmed value. When port F is in input mode, this register will read back the logic levels on signal lines DIO F7-0, and writing to this register will have no effect. The direction of port F is controlled by DIRF7-0 bits in the register at BAR0+45. Each bit in the direction register controls the direction of the corresponding port bit.

The DIO port F defaults to input mode on power up and all the register bits reset to 0.

BAR0 + 38 (0x26) Write DIO Port A Configuration

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	X	MODEA	DIRA
Reset value							0	0

This register is used to configure the direction and mode control of the DIO port A. The DIO port A can be configured in either input or output direction and be operated in two different modes: mode 0 and mode 1. In mode 0, the ports operate as a regular DIO port and no handshake signals are involved. The mode 1 operation of the DIO port enables a Latch/ACK handshake mechanism with a device which can implement similar functionality. This is explained in details in the DIO port usage in Chapter 14 of this document.

MODEA Mode control bit for DIO port A.

1 = Set the DIO port A in Latched mode of operation with handshake.

0 = Normal DIO operation for DIO port A without handshake. (Default setting)

DIRA Direction control bit for DIO port A.

1 = DIO Port A is configured as an output port.

0 = DIO Port A is configured as an input port. (Default setting)

BAR0 + 38 (0x26) Read DIO Port A Configuration

Bit No.	7	6	5	4	3	2	1	0
Name	LATCHA	X	X	X	X	X	MODEA	DIRA
Reset value	0						0	0

This register provides a read back of the configuration of DIO port A.

LATCHA Latch signal indication. When the DIO port A is configured to operate in Mode 1 (Latched operation with handshake), if this bit is detected going high, the user program should send out an ACK to the remote device by writing 1 to the ACK bit in register at BAR0+47.

1 = This value has a different meaning depending on the direction setting of the port as explained in table below.

LATCHA	Port Direction	Meaning
1	Input	Indicates that the data is available to read from the port
1	Output	Indicates that the remote entity has acknowledged the receipt of data

0 = If MODEA = 1, then this indicates that the remote device has not taken the action as shown in the table above. The software should continue to monitor the bit. If MODEA=0, then this bit has no meaning.

MODEA — Read back of the mode bit.

1 = DIO port A is in Latched mode of operation with handshake.

0 = DIO port A is operating as a regular DIO port without handshake.

DIRA — Read back of the direction setting of DIO port A.

1 = DIO port A is configured as an output.

0 = DIO port A is configured as an input. (Default setting)

BAR0 + 39 (0x27) Write DIO Port B Configuration

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	X	MODEB	DIRB
Reset value							0	0

This register is used to configure the direction and mode control of the DIO port B. The DIO port B can be configured in either input or output direction and be operated in two different modes: mode 0 and mode 1. In mode 0, the ports operate as a regular DIO port and no handshake signals are involved. The mode 1 operation of the DIO port enables a Latch/ACK handshake mechanism with a device which can implement similar functionality. This is explained in the details in the DIO port usage section in Chapter 14 of this document.

MODEB — Mode control bit for DIO port B.

1 = Set the DIO port B in Latched mode of operation with handshake.

0 = Normal DIO operation for DIO port A without handshake. (Default setting)

DIRB — Read back of the direction setting of DIO port B.

1 = DIO port B is configured as an output.

0 = DIO Port B is configured as an input port. (Default setting)

BAR0 + 39 (0x28) Read DIO Port B Configuration

Bit No.	7	6	5	4	3	2	1	0
Name	LATCHB						MODEB	DIRB
Reset value								

This register provides a read back of the configuration of DIO port B.

LATCHB — Latch signal indication. When the DIO port B is configured to operate in Mode 1 (Latched operation with handshake), if this bit is detected going high, the user program should send out an ACK to the remote device by writing 1 to the ACK bit in register at BAR0+47.

1 = This value has a different meaning depending on the direction setting of the port as explained in the table below.

LATCHB	Port Direction	Meaning
1	Input	Indicates that the data is available to read from the port
1	Output	Indicates that the remote entity has acknowledged the receipt of data

0 = If MODEB = 1, then this indicates that the remote device has not taken the action as shown in the table above. The software should continue to monitor the bit.

If MODEB=0, then this bit has no meaning.

MODEB — Read back of the mode bit.

1 = DIO port B is in Latched mode of operation with handshake.

0 = DIO port B is operating as a regular DIO port without handshake.

DIRB — Read back of the direction setting of DIO port B.

1 = DIO port B is configured as an output.

0 = DIO port B is configured as an input. (Default setting)

BAR0 + 40 (0x29) Write DIO Port C Configuration

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	X	MODEC	DIRC
Reset value							0	0

This register is used to configure the direction and mode control of the DIO port C. The DIO port C can be configured in either input or output direction and be operated in two different modes: mode 0 and mode 1. In mode 0, the ports operate as a regular DIO port and no handshake signals are involved. The mode 1 operation of the DIO port enables a Latch/ACK handshake mechanism with a device which can implement similar functionality. This is explained in the details in the DIO port usage section in Chapter 14 of this document.

MODEC — Mode control bit for DIO port C.

1 = Set the DIO port C in Latched mode of operation with handshake.

0 = Normal DIO operation for DIO port C without handshake. (Default setting)

DIRC — Direction control bit for DIO port C.

1 = DIO Port C is configured as an output port.

0 = DIO Port C is configured as an input port. (Default setting)

BAR0 + 40 (0x29) Read DIO Port C Configuration

Bit No.	7	6	5	4	3	2	1	0
Name	LATCHC	X	X	X	X	X	MODEC	DIRC
Reset value	0						0	0

This register provides a read back of the configuration of DIO port C.

LATCHC — Latch signal indication. When the DIO port C is configured to operate in Mode 1 (Latched operation with handshake), if this bit is detected going high, the user program should send out an ACK to the remote device by writing 1 to the ACK bit in register at BAR0+47.

1 = This value has a different meaning depending on the direction setting of the port as explained in table below.

LATCHC	Port Direction	Meaning
1	Input	Indicates that the data is available to read from the port
1	Output	Indicates that the remote entity has acknowledged the receipt of data

0 = If MODEC = 1, then this indicates that the remote device has not taken the action as shown in the table above. The software should continue to monitor the bit.

If MODEC=0, then this bit has no meaning.

MODEC — Read back of the mode bit.

1 = DIO port C is in Latched mode of operation with handshake.

0 = DIO port C is operating as a regular DIO port without handshake.

DIRC — Read back of the direction setting of DIO port C.

1 = DIO port C is configured as an output.

0 = DIO port C is configured as an input. (Default setting)

BAR0 + 41 (0x2A) Write DIO Port D Configuration

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	X	MODED	DIRD
Reset value							0	0

This register is used to configure the direction and mode control of the DIO port D. The DIO port D can be configured in either input or output direction and be operated in two different modes: mode 0 and mode 1. In mode 0, the ports operate as a regular DIO port and no handshake signals are involved. The mode 1 operation of the DIO port enables a Latch/ACK handshake mechanism with a device which can implement similar functionality. This is explained in the details in the DIO port usage section in Chapter 14 of this document.

MODED — Mode control bit for DIO port D.

1 = Set the DIO port D in Latched mode of operation with handshake.

0 = Normal DIO operation for DIO port D without handshake. (Default setting)

DIRD — Direction control bit for DIO port D.

1 = DIO Port D is configured as an output port.

0 = DIO Port D is configured as an input port. (Default setting)

BAR0 + 41 (0x2A) Read DIO Port D Configuration

Bit No.	7	6	5	4	3	2	1	0
Name	LATCHD	X	X	X	X	X	MODED	DIRD
Reset value	0						0	0

This register provides a read back of the configuration of DIO port D.

LATCHD — Latch signal indication. When the DIO port D is configured to operate in Mode 1 (Latched operation with handshake), if this bit is detected going high, the user program should send out an ACK to the remote device by writing 1 to the ACK bit in register at BAR0+47.

1 = This value has a different meaning depending on the direction setting of the port as explained in table below.

LATCHD	Port Direction	Meaning
1	Input	Indicates that the data is available to read from the port
1	Output	Indicates that the remote entity has acknowledged the receipt of data

0 = If MODED = 1, then this indicates that the remote device has not taken the action as shown in the table above. The software should continue to monitor the bit. If MODED=0, then this bit has no meaning.

MODED — Read back of the mode bit.

1 = DIO port D is in Latched mode of operation with handshake.

0 = DIO port D is operating as a regular DIO port without handshake.

DIRD — Read back of the direction setting of DIO port D.

1 = DIO port D is configured as an output.

0 = DIO port D is configured as an input. (Default setting)

BAR0 + 42 (0x2C) Read/Write DIO Port E Configuration

Bit No.	7	6	5	4	3	2	1	0
Name	DIRE7-0							
Reset value	0	0	0	0	0	0	0	0

This register controls the direction for DIO port E. Each of the DIO port E bits is controllable individually using corresponding bit in this register as either input or output.

DIRE7-0 — These bits provide direction configuration of the DIO port E. When any bit is 0, the corresponding DIO port E bit is configured as an input while the port bit acts as an output when the corresponding bit in this register is set to 1.

The default value of this register is 0 which means that all bits of DIO port E are configured as inputs. Below are a few examples of various configurations.

DIRE7-0 = 0x0F - Port E bits 7, 6, 5 and 4 are input while bits 3, 2, 1 and 0 are output.
DIRE7-0 = 0xC0 - Port E bits 7 and 6 are output and all other bits are inputs.

BAR0 + 43 (0x2D) Read/Write DIO Port F Configuration

Bit No.	7	6	5	4	3	2	1	0
Name	DIRF7-0							
Reset value	0	0	0	0	0	0	0	0

This register controls the direction for DIO port F. Each of the DIO port F bits is controllable individually using corresponding bit in this register as either input or output.

DIRF7-0 — These bits provide direction configuration of the DIO port F. When any bit is 0, the corresponding DIO port F bit is configured as an input while the port bit acts as an output when the corresponding bit in this register is set to 1.

The default value of this register is 0 which means that all bits of DIO port F are configured as inputs.

BAR0 + 45 (0x2F) Read/Write ACK

Bit No.	7	6	5	4	3	2	1	0
Name	ACK							
Reset value	0	0	0	0	0	0	0	0

This register provides the ability for the software to pulse the Acknowledge line for latched mode of operation of the DIO ports A-D.

ACK — ACK signal to output to the device operating in mode 1.

1 = Causes the ACK pulse output to the remote device along with clearing of the LATCH# bit.

0 = NO ACTION.

6.2.4 BAR0 + 48 (0x30) Counters & PWM Block Registers

Offset from Block Base (Dec)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 2	Bit 0
0	CTRD7-0							
1	CTRD15-8							
2	CTRD23-16							
3	CTRD31-24							
4 (W)	CTRN7-0							
5 (C)	CTRCMD3-0						CCD1-0	
6								
7								
8	PWMD7-0							
9	PWMD15-8							
10	PWMD23-16							
11 (W)	PWCMD3-0				PWMCD	PWM2-0		
12								
13								
14								
15								

BAR0 +48 (0x30) Read/Write Counter Byte 0 (LSB)

Bit No.	7	6	5	4	3	2	1	0
Name	CTRD7-0							
Reset value	0	0	0	0	0	0	0	0

This register is used for both the Counter 0 and Counter 1. The counter holds the LSB of the counter data for either counter indicated by the counter number register.

When writing to this register, the value is written to an internal register first and when a command is issued at the Counter command register at BAR0+53, the value is actually reflected in the selected counter.

When reading from this register, the value returned by the register is the value that was latched on the last latch command issued to the counter. Thus the value returned is not the value written to the counter.

BAR0 + 49 (0x01) Read/Write Counter Byte 1

Bit No.	7	6	5	4	3	2	1	0
Name	CTRD15-8							
Reset value								

This register is used for both the Counter 0 and Counter 1. The counter holds byte 1 of the counter data for either counter indicated by the counter number register.

When writing to this register, the value is written to an internal register first and when a command is issued at the Counter command register at BAR0+53, the value is actually reflected in the selected counter.

When reading from this register, the value returned by the register is the value that was latched on the last latch command issued to the counter. Thus the value returned is not the value written to the counter.

BAR0 + 50 (0x02) Read/Write Counter Byte 2

Bit No.	7	6	5	4	3	2	1	0
Name	CTRD23-16							
Reset value								

This register is used for both the Counter 0 and Counter 1. The counter holds byte 2 of the counter data for either counter indicated by the counter number register.

When writing to this register, the value is written to an internal register first and when a command is issued at the Counter command register at BAR0+53, the value is actually reflected in the selected counter.

When reading from this register, the value returned by the register is the value that was latched on the last latch command issued to the counter. Thus the value returned is not the value written to the counter.

BAR0 + 51 (0x03) Read/Write Counter Byte 3(MSB)

Bit No.	7	6	5	4	3	2	1	0
Name	CTRD31-24							
Reset value								

This register is used for both the Counter 0 and Counter 1. The counter holds byte 3 (<MSB) of the counter data for either counter indicated by the counter number register.

When writing to this register, the value is written to an internal register first and when a command is issued at the Counter command register at BAR0+53, the value is actually reflected in the selected counter.

When reading from this register, the value returned by the register is the value that was latched on the last latch command issued to the counter. Thus the value returned is not the value written to the counter.

BAR0 + 52 (0x34) Write Counter Number Register

Bit No.	7	6	5	4	3	2	1	0
Name	CTRN7-0							
Reset value	0	0	0	0	0	0	0	0

CTRN7-0 Counter number register. FP-DAQ1616 has only 2 counters, thus only values 1 and 2 are valid values for this register.

BAR0 + 53 (0x35) Write Counter Command Register

Bit No.	7	6	5	4	3	2	1	0
Name	CTRCMD3-0				X	X	CCD1	CCD0
Reset value	0	0	0	0			0	0

This register has various commands to control the behavior of the counter selected by the value in the Counter Number register.

CTRCMD3-0 These bits provide the following control commands for the counter/timer: clear, load, enable, disable, reset, latch and select counter clock source.

CCD1-0 These bits are additional control bits and operate with the CTRCMD3-0 bits to provide additional ability to control the counter behavior. Their use is explained in the following table.

CTRCMD3-0				Command	Control Bits		Action	Register Value
3	2	1	0		CCD 1	CCD 0		
0	0	0	0	Clear Counter.	X	X	-	0x00
0	0	0	1	Load the selected counter with data in CTRD32-0.	X	X	-	0x10
0	0	1	0	Select Count Direction.	X	1	Count up	0x21
					X	0	Count down	0x20
0	0	1	1	Enable / Disable External gate. When this command is selected for Counter 1, DIO pin AUX5 is reconfigured as an input and used for Counter 1 gate.	X	1	Enable Gating	0x31
					X	0	Disable Gating	0x30
0	1	0	0	Enable/Disable counting.	X	1	Enable Counting	0x41
					X	0	Disable Counting	0x40
0	1	0	1	Latch Selected counter.	X	X	-	0x50
0	1	1	0	Select Counter clock source.	0	X	Counter Input pin	0x60
					1	0	Internal CLK 50MHz	0x62
					1	1	Internal CLK 5MHz	0x63
1	1	1	1	Reset selected or both counters.	X	0	Reset selected counter	0xF0
					X	1	Reset both counters	0xF1

BAR0 + 54 (0x38) Read/Write PWM Data Register 0 (LSB)

Bit No.	7	6	5	4	3	2	1	0
Name	PWMD7-0							
Reset value	0	0	0	0	0	0	0	0

PWMD7-0 This register holds the LSB of the 24 bit PWM data value.

BAR0 + 55 (0x39) Read/Write PWM Data Register 1

Bit No.	7	6	5	4	3	2	1	0
Name	PWMD15-8							
Reset value	0	0	0	0	0	0	0	0

PWMD15-8 This register holds the second byte of the PWM data value.

BAR0 + 56 (0x40) Read/Write PWM Data Register 2 (MSB)

Bit No.	7	6	5	4	3	2	1	0
Name	PWMD23-16							
Reset value								

PWMD23-16 This register holds the MSB of the 23 bit PWM data value.

BAR0 + 57 (0x41) Write PWM Control Register

Bit No.	7	6	5	4	3	2	1	0
Name	PWCMD3-0				PWMCD	PWM2-0		
Reset value	0	0	0	0	0	0	0	0

This register provides PWM command information for various PWM operations. The different combinations of the register bits invoke the PWM command as shown in the table below.

PWCMD3-0 These bits contain the PWM command to execute on the PWM block.

The details of commands are as below.

0000 Stop all / selected PWM as indicated by PWMCD
0 = stop all PWMs (opposite polarity for "all" compared to other commands)
1 = stop PWM selected with PWM2-0
Command 0x00 = stop all PWMs

0001 Load counter C0 or C1 selected by PWMCD:
0 = load C0 / period counter
1 = load C1 = duty cycle counter

0010 Set polarity for output according to PWMCD. The pulse occurs at the start of the period.
0 = pulse high
1 = pulse low

0011 Enable/disable pulse output as indicated by PWMCD
0 = disable pulse output; output = opposite of polarity setting from command 0010
1 = enable pulse output

0100 Clear all / selected PWM as indicated by PWMCD
0 = clear PWM selected with PWM2-0
1 = clear all PWMs

0101 Enable/disable PWM outputs on DIO port F according to PWMCD
0 = disable output
1 = enable output on DIOFn where n = PWM number; this forces DIOFn to output mode

0110 Select clock source for PWM indicated by PWM2-0 according to PWMCD (both counters C0 and C1 use the same clock source):
0 = 50MHz
1 = 1MHz

0111 Start all / selected PWM as indicated by PWMCD
0 = start PWM selected with PWM2-0
1 = start all PWMs
Command 0x7F = start all PWMs

PWMCD Additional control bit that performs a selection on the command indicated by PWMCMD3-0.

The functionality of this bit is described above with PWMCMD3-0 bits.

PWM2-0 PWM circuit number. The command indicated by PWMCMD3-0 is applied to the PWM circuit indicated by these bits.

6.2.5 BAR0 + 64 (0x40) FIFO Block Registers

Offset from Block Base (Dec)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 2	Bit 0
0	FT7-0							
1	FT15-8							
2	FT23-16							
3	FT31-24							
4 (R)	FD7-0							
5 (R)	FD15-8							
6 (R)	FD23-16							
7 (R)	FD31-24							
8 (R)	FSIZE7-0							
9 (R)	FSIZE15-8							
10 (R)	FSIZE23-16							
11 (R)	FSIZE31-24							
12								FIFOEN
13 (C)	FIFORST							
13 (R)					OF	FF	TF	EF
14								
15								

BAR0 + 64 (0x40) Read/Write FIFO Threshold Register (LSB)

Bit No.	7	6	5	4	3	2	1	0
Name	FT7-0							
Reset value	0	0	0	0	0	0	0	0

FT7-0 This register holds the LSB of the FIFO threshold value. The FPGA uses the 24 bit FIFO threshold value to generate an interrupt when the number of samples in the FIFO is equal to the FIFO threshold value.

When read, this register returns the LSB of the FIFO threshold value previously programmed.

BAR0 +65 (0x41) Read/Write FIFO Threshold Register

Bit No.	7	6	5	4	3	2	1	0
Name	FT15-8							
Reset value	0	0	0	0	0	1	0	0

FT15-8 This register is holds the second byte of the FIFO threshold register. The default FIFO threshold value is 1024 samples thus this register powers up to a value of 0x04.

When read, this register returns the middle byte of the FIFO threshold value previously programmed.

BAR0+ 66 (0x42) Read/Write FIFO Threshold Register (MSB)

Bit No.	7	6	5	4	3	2	1	0
Name	FT23-16							
Reset value	0	0	0	0	0	0	0	0

FT23-16 This register holds the MSB of the FIFO threshold value. The FP-DAQ1616 board has a 16K deep sample FIFO.

When read, this register returns the MSB of the FIFO threshold value previously programmed.

BAR0 + 67 (0x43) Read/Write For Future Expansion

Bit No.	7	6	5	4	3	2	1	0
Name	FT31-24							
Reset value								

FT31-24 This register is reserved for future expansion and is not used in the current design.

BAR0 + 68 (0x44) Read FIFO Depth LSB

Bit No.	7	6	5	4	3	2	1	0
Name	FD7-0							
Reset value	0	0	0	0	0	0	0	0

FD7-0 This register provides the LSB of the current FIFO depth. FIFO depth is defined as the total number of sample currently available for reading in the FIFO.

The default value of this register is 0 since there are no samples available in the FIFO. For example when the FIFO contains 255 samples, this register would read 0xFF.

This register will be reset to zero upon board reset or FIFO reset commands.

BAR0 + 69 (0x45) Read FIFO Depth Middle byte

Bit No.	7	6	5	4	3	2	1	0
Name	FD15-8							
Reset value	0	0	0	0	0	0	0	0

FD15-8 This register provides the middle byte of the current FIFO depth. This register will be reset to zero upon board reset or FIFO reset commands.

BAR0 + 70 (0x46) Read FIFO Depth MSB

Bit No.	7	6	5	4	3	2	1	0
Name	FD23-16							
Reset value	0	0	0	0	0	0	0	0

FD23-16 This register provides the MSB of the current FIFO depth. This register will be reset to zero upon board reset or FIFO reset commands.

BAR0 + 71 (0x47) Read Reserved for Future Expansion

Bit No.	7	6	5	4	3	2	1	0
Name	FD31-24							
Reset value								

FD31-24 This register is reserved for future expansion and is not used in the current design.

BAR0 + 72 (0x48) Read FIFO Size LSB

Bit No.	7	6	5	4	3	2	1	0
Name	FSIZE7-0							
Reset value	0	0	0	0	0	0	0	0

FSIZE7-0 This register will always read 0x00 in the current design since the available FIFO buffer on the board is 0x4000 (16384 samples).

BAR0 + 73 (0x49) Read FIFO Size middle byte

Bit No.	7	6	5	4	3	2	1	0
Name	FSIZE15-8							
Reset value	0	1	0	0	0	0	0	0

FSIZE15-8 This register will always read 0x40 in the current design since the available FIFO buffer on the board is 0x4000 (16384 samples).

BAR0 + 74 (0x4A) Read Reserved For Future Expansion

Bit No.	7	6	5	4	3	2	1	0
Name	FSIZE23-16							
Reset value								

FSIZE23-16 This register is reserved for future expansion and is not used in the current design.

BAR0 + 75 (0x4B) Read Reserved For Future Expansion

Bit No.	7	6	5	4	3	2	1	0
Name	FSIZE31-24							
Reset value								

FSIZE31-24 This register is reserved for future expansion and is not used in the current design.

BAR0 + 76 (0x4C) Read/Write FIFO Control Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	X	X	FIFOEN
Reset value								0

FIFOEN Enable/Disable FIFO. When FIFO is enabled, the FPGA stores the samples in the FIFO buffer and increments the FIFO depth value. This is explained in detail in Chapter 9 of this document.
1 = Enable FIFO.
0 = Disable FIFO (Default setting).

BAR0 +77 (0x4D) Command (Write) FIFO Control Register

Bit No.	7	6	5	4	3	2	1	0
Name	FIFORST	X	X	X	X	X	X	X
Reset value	0							

FIFORST Enables FIFO reset. When FIFO is reset, all the available samples in the FIFO buffer are lost and the FIFO depth register is reset to 0. The FIFO flags are also reset to their default state with FIFO Empty Flag set to 1.
1 = Reset the FIFO.
0 = NO ACTION.

BAR0 + 78 (0x4D) Read FIFO Flags Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	OF	FF	TF	EF
Reset value					0	0	0	0

This register provides the FIFO status using 4 flag bits called Overflow, FIFO Full, Threshold and Empty.

OF — This indicates that the FIFO has overflowed and unless the user empties the FIFO by reading the sample values, all subsequent AD conversions will be lost and not stored in the FIFO buffer.

1 = FIFO has overflowed.

0 = FIFO may be FULL or have space for samples.

FF — This bit indicates that the FIFO buffer is full and the next A/D conversion will lead to a FIFO overflow. The user is expected to start emptying the FIFO by reading samples from the buffer.

1 = FIFO buffer is full.

0 = FIFO still has space for storing at least one more A/D sample.

TF — This bit indicates that the number of samples in the FIFO buffer is equal to the FIFO threshold.

1 = FIFO has at least FIFO threshold number of samples in the buffer.

0 = FIFO has less than FIFO threshold number of samples in the buffer.

EF — This bit indicates that the FIFO buffer is empty.

1 = Indicates that the FIFO is empty.

0 = This means that there is at least one sample stored in the FIFO buffer.

6.2.6 BAR0 + 80 (0x50) EEPROM/Calibration Block Registers

Offset from Block Base (Dec)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 2	Bit 0
0	ECA7-0							
1	ECA15-8							
2								
3						EEDEV2-0		
4	ECD7-0							
5								
6						REFSEL1	REFSEL0	CMUXEN
7 (C)	EEEN	EERW	LOADCAL		TDACWR			
7 (R)	EEBUSY				TDBUSY			
8	Unlock code A							
9	Reserved							
10								
11								
12	(Reserved for auto-autocal)							
13	(Reserved for auto-autocal)							
14	(Reserved for auto-autocal)							
15	(Reserved for auto-autocal)							

BAR0 + 80 (0x50) Read/Write EEPROM Address LSB

Bit No.	7	6	5	4	3	2	1	0
Name	ECA7-0							
Reset value	0	0	0	0	0	0	0	0

ECA7-0 LSB of the EEPROM address or the TrimDAC number in case of TrimDAC access.

BAR0 + 81 (0x51) Read/Write EEPROM Address MSB

Bit No.	7	6	5	4	3	2	1	0
Name	ECA15-8							
Reset value	0	0	0	0	0	0	0	0

ECA15-8 MSB of the EEPROM address. This register is not used for TrimDAC access.

BAR0 + 83 (0x53) Read/Write EEPROM Select

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	EEDEV2	EEDEV1	EEDEV0
Reset value						0	0	0

EEDEV2-0 This register always reads 0. The user should not change the value of this register as this could result in EEPROM malfunction.

BAR0 + 84 (0x54) Read/Write EEPROM/TrimDAC Data Register

Bit No.	7	6	5	4	3	2	1	0
Name	ECD7-0							
Reset value	0	0	0	0	0	0	0	0

During EEPROM or TrimDAC write operations, the data written to this register will be written to the selected device. During EEPROM read operations this register contains the data read from the EEPROM and is valid after EEBUSY = 0. The TrimDAC data cannot be read back.

ECD7-0 Calibration data to be read or written to EEPROM. In case of TrimDAC access, TrimDAC data to be written to the selected TrimDAC.

BAR0 + 86 (0x56) Read/Write Calibration Reference Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	REFSEL1	REFSEL0	CMUXEN
Reset value						0	0	0

This register provides the ability to measure the on-board precision reference voltages so that the board can be calibrated to those values.

REFSEL1-0 Provide different reference voltages on the A/D channels. When CMUXEN=1, different reference voltages are available on the A/D channels based on the values of these bits.

CMUXEN Calibration multiplexor enable. The cal mux is used to read precision on-board reference voltages that are used in the auto-calibration process. It also can be used to read back the value of the D/A temperature sensor and D/A monitor output.

1 = Enable cal mux and disable user inputs.

0 = Disable cal mux and enable user inputs.

BAR0 + 87 (0x57) Command (Write) EEPROM/TrimDAC Command Register

Bit No.	7	6	5	4	3	2	1	0
Name	EEEN	EERW	LOADCAL	X	TDACWR	X	X	X
Reset value	0	0	0		0			

This is a command register to perform various actions on either EEPROM or the TrimDACs.

EEEN EEPROM Enable bit.

1 = When this bit is set to 1, the EEPROM can be accessed either to write or read data to/from the EEPROM.

0 = NO ACTION.

EERW EEPROM operation bit. This bit enables either read or write operation on the EEPROM. For this bit to be active, the EEEN MUST be set to 1.

1 = EEPROM Write operation.

0 = EEPROM Read operation.

LOADCAL Load calibration values from the EEPROM on to TrimDACs.

1 = Causes the board to reload calibration settings from EEPROM. All other bits are ignored when this bit is set to 1 but the CALMUXEN setting is preserved.

0 = NO ACTION.

TDACWR Trim DAC Write command.

1 = Performs an update of the TrimDAC with the TrimDAC data on the TrimDAC selected by the TrimDAC address register.

0 = NO ACTION.

BAR0 + 87 (0x57) Read EEPROM/TrimDAC Status Register

Bit No.	7	6	5	4	3	2	1	0
Name	EEBUSY	X	X	X	TDBUSY	X	X	X
Reset value	0				0			

This register provides the status on the EEPROM and TrimDAC devices. It is required that the software polls the corresponding status bits before performing next operation on either of the devices.

EEBUSY EEPROM Busy/IDLE indication.

1 = The EEPROM is busy performing the previous operation. No new EEPROM command will be accepted.

0 = EEPROM is IDLE and ready to accept a new command.

TDBUSY TrimDAC Busy/IDLE indication.

1 = The TrimDAC is busy performing the previous operation. No new TrimDAC command will be accepted.

0 = TrimDAC is IDLE and ready to accept a new command.

BAR0 + 88 (0x58) Read/Write EEPROM Unlock Code1 Register

Bit No.	7	6	5	4	3	2	1	0
Name	Unlock code A							
Reset value	0	0	0	0	0	0	0	0

The EEPROM access is protected to prevent accidental access and over-write of the EEPROM data. In order to gain access to the EEPROM, a series of codes MUST be written to the EEPROM register. This register MUST be written with a value of 0xA5 followed by a value of 0x24 to unlock the EEPROM. If the values are not written in this order, the EEPROM will not be unlocked.

Once the EEPROM is unlocked, the EEPROM remains accessible while the board is powered up. The EEPROM gets locked following a Board Reset or an FPGA Reset command.

6.2.7 BAR0 + 112 (0x70) Interrupts, Miscellaneous, and ID Block Registers

Offset from Block Base (Dec)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 2	Bit 0
0					T1INTEN	T0INTEN	DINTEN	AINTEN
1 (C)					T1INTCLR	T0INTCLR	DINTCLR	AINTCLR
1 (R)					T1INT	T0INT	DINT	AINT
2								
3								
4								LED
5 (R)						SLOT2-0		
6							CFG1-0	
7 (R)	SMBus address							
8 (R)	Board ID minor							
8 (R)	Board ID major							
10 (R)	Board revision minor							
11 (R)	Board revision major							
12 (R)	FPGA ID minor							
13 (R)	FPGA ID major							
14 (R)	FPGA revision minor							
15 (C)	BRDRST	FPGARST						
15 (R)	FPGA revision major							

BAR0 + 112 (0x70) Read/Write Interrupt Enable Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	T1INTEN	T0INTEN	DINTEN	AINTEN
Reset value					0	0	0	0

This register provides interrupt enable bits for various interrupts that the FP-DAQ1616 board can generate. The on-board FPGA will not generate an interrupt unless the corresponding enable bit in this register is set to 1.

T1INTEN Timer 1 interrupt enable.

1 = Timer 1 Interrupt is enabled. When the counter 1 reaches zero, the interrupt will be generated.

0 = Timer 1 interrupt disabled. (Default setting)

T0INTEN Timer 0 interrupt enable.

1 = Timer 0 Interrupt is enabled. When the counter 0 reaches zero, the interrupt will be generated.

0 = Timer 0 interrupt disabled. (Default setting)

DINTEN DIO interrupt enable.

1 = DIO Interrupt is enabled. The FPGA will generate an interrupt when any of the DIO ports is in Mode1 of operation. When DINTEN=1, DIOF6-7 are reassigned as follows: DIOF6 is a latch signal with input direction and default value high. DIOF7 is an acknowledge signal with output direction and default value high.

0 = DIO interrupt is disabled.

AINTEN A/D interrupt enable.

1 = A/D interrupt is enabled. The FPGA will generate an interrupt when the number of samples in the FIFO buffer is equal to the FIFO threshold value set in the FIFO Threshold register when FIFO is enabled. If FIFO is disabled, an interrupt will be generated on every A/D conversion.

AINT is set to 1 and an interrupt occurs when AINTE=1 and one of the following occurs:

FIFOEN	SCANEN	Action
0	0	Interrupt occurs after each A/D conversion completes (ADBUSY goes low). Only most recent A/D value is available. A/D value is stored in sample storage block.
0	1	Interrupt occurs after each A/D scan completes (ADBUSY goes low). Only most recent A/D scan is available. A/D values are stored in sample storage block.
1	0	Interrupt occurs when A/D conversion completes and FIFO threshold is reached **or exceeded.** A/D data is stored in the FIFO.

0 = A/D interrupt is disabled.

BAR0 + 113 (0x71) Command (Write) Interrupt Flip-Flop Reset Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	T1INTCLR	T0INTCLR	DINTCLR	AINTCLR
Reset value					0	0	0	0

This register provides different command bits for resetting various interrupt flip-flop bits. For every interrupt type, a clear bit is provided which is required to be set to 1 by the interrupt service routine after every execution. If the corresponding clear bit is not set in the ISR, the FPGA will not generate any more interrupts.

This register accepts only one command at a time.

T1INTCLR Timer 1 interrupt request flip-flop reset bit.

1 = Causes the timer 1 interrupt request flip-flop to be reset.
0 = NO ACTION.

T0INTCLR Timer 0 interrupt request flip-flop reset bit.

1 = Causes the timer 0 interrupt request flip-flop to be reset.
0 = NO ACTION.

DINTCLR DIO interrupt request flip-flop reset bit,

1 = Causes the DIO interrupt request flip-flop to be reset.
0 = NO ACTION.

AINTCLR A/D interrupt request flip flop reset bit.

1 = Causes the A/D interrupt request flip-flop to be reset.
0 = NO ACTION.

BAR0 + 114 (0x72) Read Interrupt Status Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	T1INT	T0INT	DINT	AINT
Reset value					0	0	0	0

This register provides status information on the interrupts generated by the FPGA. The user program can monitor the respective bits in this register to obtain the status on the interrupt that was configured.

T1INT Timer 1 interrupt status.

1 = Timer 1 interrupt is pending.
0 = Timer 1 interrupt is not pending.

T0INT Timer 0 interrupt status.

1 = Timer 0 interrupt is pending.
0 = Timer 0 interrupt is not pending.

DINT DIO interrupt status

1 = DIO interrupt is pending.
0 = DIO interrupt is not pending.

AINT A/D interrupt status.

1 = A/D interrupt is pending.
0 = A/D interrupt is not pending.

BAR0 + 116 (0x74) Read/Write LED Control Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	X	X	LED
Reset value								0

LED On board LED indicator control register. This register controls the on board LED and the value of the LED is inverted to the bit value in the register.

1 = Turn the LED OFF.
0 = Turn the LED ON. (default power-up value)

When this register is read, it returns the state of the LED in inverted mode. If the LED is ON, the value returned is 0 and 1 otherwise.

BAR0 + 117 (0x75) Read Reserved for Future Expansion

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	SLOT2-0		
Reset value						0	0	0

SLOT2-0 FP Slot indicator. This is not used at present and is reserved for future expansion.

BAR0 + 118 (0x76) Read/Write Board Configuration Settings

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	X	CFG1-0	
Reset value							0	0

CFG1-0 These bits provide information on the NVRAM on the board which holds the board configuration data. This is not used at present. The register should always be 0.

BAR0 + 119 (0x77) Read SMBus Address Register

Bit No.	7	6	5	4	3	2	1	0
Name	SMBus address							
Reset value	X	X	X	X	X	X	X	X

SMBus address This register is reserved for future expansion and is not used at present.

BAR0 + 120 (0x78) Read Board ID Minor Register

Bit No.	7	6	5	4	3	2	1	0
Name	Board ID minor							
Reset value	0	0	0	0	0	0	0	1

Board ID minor This register provides information on the Board ID. This register will read 0x01.

BAR0 + 121 (0x79) Read Board ID Major Register

Bit No.	7	6	5	4	3	2	1	0
Name	Board ID major							
Reset value	0	0	0	0	0	0	0	0

Board ID major This register will return the major revision number of the board. At present it will read 0x00.

BAR0 + 122 (0x7A) Read Board Revision Register

Bit No.	7	6	5	4	3	2	1	0
Name	Board revision minor							
Reset value	0	0	0	0	0	0	0	0

Board revision minor Returns a value of 0x00

BAR0 + 123 (0x7B) Read Board Revision Register

Bit No.	7	6	5	4	3	2	1	0
Name	Board revision major							
Reset value	0	0	0	0	0	0	0	0

Board revision major Returns a value of 0x00

BAR0 + 124 (0x7C) Read FPGA ID Minor Register

Bit No.	7	6	5	4	3	2	1	0
Name	FPGA ID minor							
Reset value	0	0	0	0	0	0	0	1

FPGA ID minor This register holds the FPGA ID minor (LSB) for the product. Every DSC product has an FPGA ID. FP-DAQ1616 has an FPGA ID of 0x0801.

This register always returns 0x01.

BAR0 + 125 (0x7D) Read FPGA ID Major Register

Bit No.	7	6	5	4	3	2	1	0
Name	FPGA ID major							
Reset value	0	0	0	0	1	0	0	0

FPGA ID major This register holds the FPGA ID major (MSB) for the product. This register always returns 0x08.

BAR0 + 126 (0x7E) Read FPGA Firmware Minor Revision Number

Bit No.	7	6	5	4	3	2	1	0
Name	FPGA revision minor							
Reset value	0	0	0	0	0	1	0	0

FPGA revision minor This register has the minor revision number for the FPGA firmware. Every time the FPGA firmware is updated, this register gets a new value.

This register will return a value of 0x04.

BAR0 + 127 (0x7F) Command(Write) Reset Register

Bit No.	7	6	5	4	3	2	1	0
Name	BRDRST	FPGARST	X	X	X	X	X	X
Reset value	0	0						

This register provides different control bits to either reset the entire board or reset the FPGA and leaving the other peripherals in their current state.

BRDRST Board Reset bit.

1 = Causes the entire board to be reset. This includes all the peripherals and the internal FPGA registers.

0 = NO ACTION.

FPGARST FPGA Reset bit.

1 = Causes the FPGA to reset to its default power up state. The FPGA registers are reset to their default state as defined by individual registers. All the contents of the FIFO are lost. None of the peripherals are reset when this command is issued.

0 = NO ACTION.

BAR0 + 127 (0x7F) Read FPGA Firmware Major Revision Number

Bit No.	7	6	5	4	3	2	1	0
Name	FPGA revision major							
Reset value	0	0	0	0	0	0	0	1

FPGA revision major This register has the major revision number of the FPGA firmware.

6.2.8 BAR0 + 224 (0xE0) SPI Flash Interface Block Registers

Offset from Block Base (Dec)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 2	Bit 0
0	SPITXD7-0							
1	SPIRXD7-0							
2	SPICMD7-0							
3 (R)						SPIRXRDY	SPITXRDY	SPIBUSY
4	SPIA7-0							
5	SPIA15-8							
6	SPIA23-16							
7								
8								
9								
10								
11								
12								
13								
14								
15								

BAR0 + 224 (0xE0) Read/Write SPI Transmit Data Register

Bit No.	7	6	5	4	3	2	1	0
Name	SPITXD7-0							
Reset value	0	0	0	0	0	0	0	0

SPITXD7-0 This register holds the data to be written to the SPI flash memory on the board. The FP-DAQ1616 board's firmware is held in this memory. If invalid data pattern is written to the device, the board will become unusable.

BAR0 + 225 (0xE1) Read/Write SPI Receive Data Register

Bit No.	7	6	5	4	3	2	1	0
Name	SPIRXD7-0							
Reset value	0	0	0	0	0	0	0	0

SPIRXD7-0 These bits provide the data read from the SPI flash device.

BAR0 + 226 (0xE2) Write SPI Command Register

Bit No.	7	6	5	4	3	2	1	0
Name	SPICMD7-0							
Reset value	0	0	0	0	0	0	0	0

SPICMD7-0 These bits indicate the type of command to send to the SPI flash device. The commands are as below.

The following SPI commands are available:

0x01	WRSR	write status register
0x02	PP	page program
0x03	RD	read data
0x04	WRDI	write disable
0x05	RDSR	read status register
0x06	WREN	write enable
0x0B	FRD	fast read data
0xAB	RES	read signature
0xC7	BE	bulk erase
0xB9	DP	deep power down
0xD8	SE	sector erase
0xFF	NOP	no command to execute / end current command

BAR0 + 227 (0xE3) Read SPI Status Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	SPIRXRDY	SPITXRDY	SPIBUSY
Reset value						0	0	0

This register provides status information on the SPI communication interface between the FPGA and the SPI flash device.

SPIRXRDY SPI flash ready to receive status bit.

1 = When the SPI receive buffer has at least one byte available for reading (SPI RX FIFO not empty).

0 = SPI receive buffer is empty.

SPITXRDY SPI flash ready to transmit status bit.

1 = When the SPI transmit buffer has room for at least one byte of transmit data (SPI TX FIFO not full).

0 = SPI transmit buffer is empty.

SPIBUSY SPI busy status bit. Application software must check this bit before performing any operation on the SPI bus.

1 = SPI circuit is busy.

0 = SPI circuit is IDLE.

BAR0 + 228 (0xE4) Read/Write SPI Address Register LSB

Bit No.	7	6	5	4	3	2	1	0
Name	SPIA7-0							
Reset value	0	0	0	0	0	0	0	0

SPIA7-0 This register holds the LSB of the 24 bit SPI address. The SPI address is required for every read/write operation.

BAR0 + 229 (0xE5) Read/Write SPI Address Register Byte 2

Bit No.	7	6	5	4	3	2	1	0
Name	SPIA15-8							
Reset value	0	0	0	0	0	0	0	0

SPIA15-8 This register holds the second byte of the 24 bit SPI address. The SPI address is required for every read/write operation.

BAR0 + 230 (0xE6) Read/Write SPI Address Register Byte 3

Bit No.	7	6	5	4	3	2	1	0
Name	SPIA23-16							
Reset value	0	0	0	0	0	0	0	0

SPIA23-16 This register holds the MSB of the 24 bit SPI address. The SPI address is required for every read/write operation.

6.2.9 BAR0 + 240 (0xF0) Capabilities/Page Control Block Registers

Offset	Write	Read	Programmed Value
0		A/D channels	All bits 0 except bits 4-3
1		A/D resolution	0x10
2		D/A channels	All bits 0 except bit 4
3		D/A resolution	0x10
4		DIO type A channels	0x30
5		DIO type B channels	0x00
6		Counter configuration	0x02
7		PWM configuration	0x34
8			
9			
10			
11			
12			
13			
14	Reserved		
15			

The registers BAR0+240 to BAR0+247 all return values as indicated in the column named Programmed Value next to it.

7. ANALOG INPUT RANGES AND RESOLUTION

The Diamond-FP-DAQ1616 features a user configurable 16-bit or 12-bit A/D converter. The default is set to 12-bits. This means that the analog input voltage can be measured to the precision of a 16- or 12-bit binary number. In the default setting, the maximum value of a 12-bit binary number is 2^{162} - 1, or 4096, so the full range of numerical values that you can get from a Diamond-FP-DAQ1616 analog input channel is 0 - 4096.

The smallest change in input voltage that can be detected is 1/(2^{12}), or 1/4096, of the full-scale input range. This smallest change results in an increase or decrease of 1 in the A/D code, and so this change is referred to as 1 LSB, or 1 least significant bit.

7.1.1 Unipolar and Bipolar Inputs

The Diamond-FP-DAQ1616 can measure both unipolar (positive only) and bipolar (positive and negative) analog voltages. The full-scale input voltage range depends on the Gain, Range, and Polarity bit settings in the A/D Configuration Register at BAR0 + 8. Preceding the A/D converter is a programmable gain amplifier that multiplies the input signal before it reaches the A/D. This gain circuit has the effect of scaling the input voltage range to match the A/D converter for better resolution. In general you should select the highest gain you can that will allow the A/D converter to read the full range of voltages over which your input signals will vary. If you pick too high a gain, then the A/D converter will clip at either the high end or low end, and you will not be able to read the full range of voltages on your input signals.

7.1.2 Input Ranges and Resolution

The table below lists the full-scale input range for each valid analog input configuration. The parameters Polarity, Range, and Gain are combined to create the value “Code”, which is the value that you must write to the A/D Configuration Register at BAR0 + 8 to get the input range shown. A total of 16 different input ranges are possible. Note that some of the modes are duplicates of other modes.

Diamond-FP-DAQ1616 Analog Input Ranges

Polarity	Range	Gain	Code	Full-Scale Rang	12-bit Resolution (1 LSB)	16-bit Resolution (1 LSB)
Bipolar	5V	1	0	±5V	2.44mV	153µV
Bipolar	5V	2	1	±2.5V	1.22mV	76µV
Bipolar	5V	4	2	±1.25V	0.61mV	38µV
Bipolar	5V	8	3	±0.625V	0.305mV	19µV
Bipolar	10V	1	4	±10V	4.88mV	305µV
Bipolar	10V	2	5	±5V	2.44mV	153µV
Bipolar	10V	4	6	±2.5V	1.22mV	76µV
Bipolar	10V	8	7	±1.25V	0.61mV	38µV
Unipolar	5V	1	8	0 - 5V	1.22V	76µV
Unipolar	5V	2	9	0-2.5V	0.61mV	38µV
Unipolar	5V	4	10	0-1.25V	0.305mV	19µV
Unipolar	5V	8	11	0-0.625V	0.153mV	9µV
Unipolar	10V	1	12	0 - 10V	2.44mV	153µV
Unipolar	10V	2	13	0 - 5V	1.22mV	76µV
Unipolar	10V	4	14	0 - 2.5V	0.61mV	38µV
Unipolar	10V	8	15	0 - 1.25V	0.305mV	19µV

7.1.3 A/D Conversion Formulas

The 12-bit value returned by the A/D converter is always a number ranging from 0 to 4095, regardless of the input range. This is because the input range of the A/D is fixed at +5V. The input signal is actually magnified and shifted to match this range before it reaches the A/D. For example, for an input range of 0-10V, the signal is first shifted down by 5V to 0-5V and then sent to the A/D circuit.

Therefore, two different formulas are needed to convert the A/D value back to a voltage, one for bipolar ranges, and one for unipolar ranges. Tables showing the correlation between A/D code and input voltage are shown on the following page.

NOTE: In all cases, the A/D circuit will generate an A/D code with values ranging from 0 to 4095 regardless of the mode/range the board is set to via the registers. It is advisable that the A/D codes be used as 0 to 4095 for all unipolar ranges while for BIPOLAR ranges, it is advisable to subtract 2048 from the resulting code to obtain an A/D value of -2047 to +2048 to represent –ve max to +ve max value of the bipolar range.

For Bipolar Input Ranges

FS = full-scale voltage (e.g. 5V for ±5V range)

If using a 16-bit signed integer in C:

A/D code = (A/D code read from FPGA)- 2048

Input voltage = ((A/D code – 2048) / 2048) x FS

Example: ±5V range selected, A/D code = **3154** (Hex 0C52)

A/D code for conversion = 3154 – 2048= 1106

Input voltage = (1106 / 2048) x 5V = **+2.7103V**

Example: ±5V range selected, A/D code = **1110** (Hex456)

A/D code for conversion = 1110 – 2048= --938

Input voltage = (-938 / 2048) x 5V = **-2.2900V**

For Unipolar Input Ranges

FS = full-scale voltage (e.g. 10 for 0 - 10V range)

Input voltage = (A/D code / 4096) x FS

Example: 0 - 10V range selected, A/D code = **1579** (Hex 62B)

Input voltage = (1579 / 2048) x 10V = **+7.7103V**

Note that this is simply the result for the ±5V range shifted up by 5V.

7.1.4 Correlation Between A/D Code and Input Voltage

The following two tables illustrate the correlation between the A/D code and the corresponding input voltage. Use these tables as guides to convert between the voltage domain and the A/D code domain.

Bipolar Input Ranges

A/D Code	Input voltage formula	Input voltage (±5V range)
0	$-V_{FS}$	-5.0000V
1	$-V_{FS}$ + 1 LSB	-4.9975V
2047	-1 LSB	-48.83mV
2048	0V	0.0000V
2049	+1 LSB	48.83mV
4095	V_{FS} - 1 LSB	4.9998V

Unipolar Input Ranges

A/D Code	Input voltage formula	Input voltage (0 - 10V range)
0	0V	0.0000V
1	1 LSB (V_{FS} / 4096)	2.44mV
2047	V_{FS} / 2 - 1 LSB	4.9975V
2048	V_{FS} / 2	5.0000V
2049	V_{FS} / 2 + 1 LSB	5.002V
4095	V_{FS} - 1 LSB	9.9975V

8. PERFORMING AN A/D CONVERSION

This chapter describes the steps involved in performing an A/D conversion on a selected input channel using direct programming (not with the driver software).

The A/D block can be controlled using the registers at BAR0+ 0 to 15 as shown in the table below.

Offset from BAR0	7	6	5	4	3	2	1	0
0 (W)	ADRESET							ADSTART
1(W)								ADTURBO
2								
3								
4(R/W)					ADLOW3	ADLOW2	ADLOW1	ADLOW0
5(R/W)					ADHIGH3	ADHIGH2	ADHIGH1	ADHIGH0
6								
7	Reserved for A/D channel expansion							
8(R/W)	RTEN	RTLOAD			ADPOL	ADRANGE	ADG1	ADG0
9(R/W)								ADSEDI
10(R/W)			ADGATEN	ADCLK1	ADCLK0	SCANINT1	SCANINT0	SCANEN
11								
12(R/W)	AUX7	AUX6	AUX5	AUX4	AUX3	AUX2	AUX1	AUX0
13(R/W)	AUXDIR7	AUXDIR6	AUXDIR5	AUXDIR4	AUXDIR3	AUXDIR2	AUXDIR1	AUXDIR0
14								
15								

ADRESET Reset the A/D block

ADSTART Perform an A/D sample or A/D scan using software command

ADLOW3-0 Low channel number in A/D scan

ADHIGH3-0 High channel number in A/D scan

RTEN Gain table enable

RTLOAD Load gain table entry for the channel number indicated by ADLOW3-0

ADPOL A/D polarity. 0 = Bipolar, 1 = Unipolar

ADRANGE A/D range. 0 = 5V, 1 = 10V

ADG1-0 A/D block gain settings. 1, 2, 4, or 8

ADSEDI A/D channels inputs are single ended (0) or differential (1)

ADGATEN A/D block uses external gating to perform A/D conversion

ADCLK1-0 A/D block clock source.

SCANINT1-0 Scan interval between channels in an A/D scan

SCANEN Scan enable bit

AUX7-0 Auxiliary DIO port bits

AUXDIR7-0 Direction control bits for the auxiliary DIO port bits

8.1 The A/D FIFO

The A/D FIFO is controlled by the registers at offsets 64-79 as shown in the table below

Offset from BAR0	7	6	5	4	3	2	1	0
64	FT7-0							
65	FT15-8							
66	FT23-16							
67								
68 (R)	FD7-0							
69 (R)	FD15-8							
70 (R)	FD23-16							
71 (R)								
72 (R)	FSIZE7-0							
73 (R)	FSIZE15-8							
74 (R)	FSIZE23-16							
75 (R)								
76								FIFOEN
77 (C)	FIFORST							
77 (R)					OF	FF	TF	EF
78								
79								

FT23-0	FIFO Threshold value for generating interrupts
FD23-0	FIFO depth. Indicates the total number of samples available in the FIFO
FSIZE23-0	Total size of the FIFO buffer memory. (0x4000 = 16384)
FIFOEN	FIFO Enable/Disable bit
FIFORST	FIFO reset
OF, FF, TF, EF FIFO	Overflow, full, threshold and empty flag status bits.

All A/D conversions are stored in an on-board FIFO (first in first out memory). The FIFO can hold up to 16384 (16K) samples.

Each time an A/D conversion is finished, the data is stored in the FIFO and the FIFO counter increments by 1. Each time you read A/D data, you are actually reading it out of the FIFO and the FIFO counter decrements by 1. When the FIFO is empty the data read from it is undefined – you may continue to read the last sample, or you may read all 1s.

You can read each A/D sample as soon as it is ready, or you can wait until you take a collection of samples (up to 16384 maximum) and then read them all out at once.

To be sure that you are getting only current A/D data, be sure to reset the FIFO each time before you start any A/D operation. This will prevent errors caused by leaving data in from a previous operation. To reset the FIFO, write 1 to the FIFORST bit at BAR0+77 (write 0x80 to the register) in the register map. This bit is not a real register bit; instead it triggers a command in the board's controller. Therefore you do not need to write a 1 and then a 0, just write a 1.

There are seven steps involved in performing an A/D conversion:

1. Select the input channel or input channel range
2. Select the analog input range (Range, Polarity, and Gain codes)
3. Wait for analog input circuit to settle
4. Start an A/D conversion on the current channel
5. Wait for the conversion to finish
6. Read the A/D data
7. Convert the numerical data to a meaningful value

If you are going to sample the same channel multiple times or sample multiple consecutive channels with the same input range, you only need to perform steps 1-3 once, and then you can repeat steps 4-6 or 4-7 as many times as desired. The seven steps are described below.

STEP1 : Select the Input Channel

To select the input channel range, write the low channel number to the low channel register at offset 4 and high channel number to the high channel register at offset 5. When you write any value to these registers, the current A/D channel register at offset 6 is set to the low channel.

If you want to sample only one channel, then both the low channel and high channel registers should be written with the same channel number.

For example, to program the board to sample channel 4 only, write the following..

```
outp ( base + 4 , 0x04 ) ;          // set low channel
outp ( base + 5 , 0x04 ) ;          // set high channel
```

where base is the address pointed to by BAR0.

To program the board to program all the 16 channels:

```
outp ( base + 4 , 0x00 ) ;          // set low channel to 0
outp ( base + 4 , 0x0F ) ;          // set high channel to 15
```

After setting the channel registers, it is required to wait till the ADWAIT bit in register at offset 3 goes low as detected by polling.

```
while ( inp ( base + 3 ) & 0x40 ) ;     // wait for bit 6 ADWAIT to go low.
```

STEP 2 : Select the Analog Input Range and Type

To select the desired input range, the register at offset 8 should be written to. This register lets you configure the range (5V or 10V), polarity (bipolar/unipolar) and gain (1, 2, 4 or 8) for the A/D block. If only one of the values needs to be changed, the register should be read first and then the required bits should be masked off for the operation.

- To set the gain and preserve range/polarity

```
value = inp ( base + 8 ) ;          // current value of the register
value = value | gain ;              // where gain can be 1,2,4 or 8.
outp ( base + 8 , value );
```

- To set the polarity and preserve range/gain

```
value = inp ( base + 8 ) ;          // current value of the register
value = value | ( polarity << 3 ) ; // polarity is either 0 (bipolar) or 1 (unipolar)
outp ( base + 8 , value ) ;
```

- To set the range and preserve polarity/gain

```
value = inp ( base + 8 ) ;          // current value of the register
value = value | ( range << 2 ) ;    // range is either 0 (5V) or 1 (10V)
outp ( base + 8 , value ) ;
```

Apart from the above settings, the board can be configured to operate with either single ended inputs or differential inputs. This is programmed in the register at offset 9.

- To set the board in 16 single ended channels mode

```
outp ( base + 9 , 0x00 ) ;          // set to single ended mode
```

- To set the board in 8 differential channels mode

```
outp ( base + 9 , 0x01 ) ;          // set to differential mode
```

STEP 3 : Wait for Analog Block to Settle

After writing to the channel and analog settings register as explained in the sections above, the analog block should be given time to settle before performing A/D sampling. To do that, you should wait for the ADWAIT bit in the register at BAR0+3, bit 6 to go low. When the bit value is 1, the circuitry is actively settling on the input signal. When the value is 0, the board is ready to perform A/D conversions. The below code will work but does not account for possible hardware problems which could cause the program to freeze. A better approach is to enable a timeout by using a loop that will exit after a fixed number of tests.

```
while (inp(base+3) & 0x40); // wait for ADWAIT to go low, base+3 bit 6
```

STEP 4 : Perform an A/D Conversion on the Current Channel

After the above steps are completed, start the A/D conversion by writing to the ADSTART bit at BAR0+0. This write operation only triggers the A/D if AINTEN = 0 (interrupts are disabled). When AINTEN = 1, the A/D can only be triggered by the on-board counter/timer (Counter 0) or an external signal. This protects against accidental triggering by software during a long-running interrupt-based acquisition process.

```
outp(base,0x01);                    // trigger an A/D conversion using software
```

This method can also be used to trigger a SCAN of A/D conversions if SCANEN=1.

STEP 5 : Wait for the Conversion to Finish

The A/D converter takes up to 500ns to complete one A/D conversion. Most processors and software can operate fast enough so that if you try to read the A/D converter immediately after starting the conversion, the read will occur before the A/D conversion completes and return invalid data. Therefore, the data acquisition block provides a status bit ADBUSY to indicate whether the A/D block is busy or idle. This bit can be read back from the status register at BAR0+3, bit 7. When the A/D converter is busy (performing an A/D conversion), ADBUSY=1 and the program must wait. When the A/D converter is idle (conversion is done and data is available), ADBUSY=0 and the program may read the data.

```
while (inp(base+3) & 0x80); // wait for ADBUSY to go low, base+3 bit 7
```

The above example could hang your program if there is a hardware fault and the bit is stuck at 1. A better solution is to use a loop with a timeout, as shown below.

```
int check_AD_status() // returns 0 if ok, -1 if error
{
int i;
for (i = 0; i < 20000; i++)
    if (!(inp(base+3) & 0x80)) return(0); // conversion completed
return(-1); // conversion did not complete
}
```

STEP 6 : Read the Data from the Board

Once the conversion is complete, you can read the data back from the A/D converter. The data is a 16-bit value and is read back in two 8-bit bytes. The LSB must be read from the board before the MSB because the data is inserted into the board's FIFO in that order. Unlike data in other registers on the board, the A/D data may only be read one time, because each time an A/D sample is read from the FIFO, the internal FIFO pointer advances and that sample is no longer available. Reading data from an empty FIFO returns unpredictable results.

The following code illustrates how to read and construct the 16-bit A/D value.

```
LSB = inp(base);
MSB = inp(base+1);
ADdata = MSB * 256 + LSB; // combine the 2 bytes into a 16-bit value
```

The final value is always in the range of 0 - 65535. This numerical value is then converted to a voltage depending on the selected input range as explained below.

In scan mode, the behavior is the same except when the program initiates a conversion, all channels in the programmed channel range will be sampled once and the data will be stored in the FIFO. The FIFO depth register increments by the scan size. When STS goes low, the program should read out the data for all channels.

The following code provides an example of reading the data after a scan of all the channels.

```
Create a function to read the AD data as shown below…
WORD readADData ( void )
{
        WORD ADdata ;
        BYTE LSB,MSB ;

        LSB = inp(base);
        MSB = inp(base+1);
        ADdata = MSB * 256 + LSB; // combine the 2 bytes into a 16-bit value

        return ADdata ;
}
……
for ( channel = 0 ; channel < 16 ; channel ++ )
{
        ADdata [channel] = readData () ; // store the data in an array of WORD type
}
```

STEP 7 : Convert the Data to Volts or Engineering Units

Once the A/D value is read, it needs to be converted to a meaningful value. The first step is to convert it back to the actual measured voltage. Afterwards, you may need to convert the voltage to some other engineering units. For example, the voltage may come from a temperature sensor and the voltage would then need to be converted to the corresponding temperature, according to the temperature sensor's characteristics.

Since there are many possible formulas for converting the input voltage to engineering values, this secondary step is not included here. Only conversion to input voltage is described. However, you can combine both transformations into a single formula if desired.

To convert the A/D value to the corresponding input voltage, use the formulas as described in Sections 7.1.3 and 7.1.4 of this document.

9. A/D SAMPLING METHODS

9.1 Sampling Modes

There are several different A/D sampling modes available on the Diamond FP-DAQ1616.

The A/D sampling is performed by the board using an A/D clock input. The clock used for the A/D conversion decides the sampling mechanism. The A/D clock can be selected by the ADCLK1-0 bits in the register at BAR0+10 of the A/D block. As described in the register definition, there are 4 different possibilities for providing the A/D clock to the A/D circuit.

- A/D start bit

  Used for software triggering for individual samples and scan operations.

- External trigger

  Used for triggering using an external source (ADGATE/AUX4 signal line, pin 152 of the FeaturePak connector).

- Rising edge of counter 0

  Can be used for interrupt based A/D conversions using precise timing from counter.

- Rising edge of counter 1

  Can be used for achieving precise timing from counter.

In each of the specified modes, the A/D samples are stored in an internal FIFO memory which is explained in the section below.

9.1.1 FIFO Description

The Diamond-FP-DAQ1616 uses a 16384-sample FIFO (First In First Out) memory buffer to manage A/D conversion data. It is used to store A/D data between the time it is generated by the A/D converter and the time it is read by the user program. The FIFO may be enabled and disabled under software control.

In single-conversion mode, the FIFO features are not generally needed so FIFO use should not be selected. However the FIFO is still actually being used. Each A/D sample is stored in the FIFO, and when the software reads the data, it reads it out of the FIFO. In low-speed sampling, each time a conversion occurs, the program reads the data, so there is always a one-to-one correspondence between sampling and reading. Thus the FIFO contents never exceed one sample. When the FIFO is enabled, the available FIFO buffer is 1 sample deep in A/D sample mode and 16 samples deep in A/D scan mode.

For high-speed sampling or interrupt operation, the FIFO substantially reduces the amount of software overhead in responding to A/D conversions as well as the interrupt rate on the bus, since it enables the program to read a number of samples all at once rather than one at a time. In addition, the FIFO is required for sampling rates in excess of the maximum interrupt rate possible on the bus. Since the Diamond-FP-DAQ1616 can sample up to 2,000,000 times per second, the FIFO is needed to reduce the interrupt rate at high speeds. When the interrupt routine runs, it reads multiple samples from the FIFO. The interrupt rate is equal to the sample rate divided by the number of samples read in each interrupt. For the FP-DAQ1616, this number, called the FIFO threshold, is programmable using the registers at BAR0+64 to 63. The FIFO threshold should be programmed to a value which will reduce the interrupt rate to as low as possible while still providing access to new data in sufficiently short time for the application. The general goal should be to keep the interrupt rate below 400Hx at all times in order to avoid overloading Windows and Linux OS with interrupts. As an example, if the threshold of 8192 samples is programmed for sampling speed of 2500000 Hz. If these values are followed, the interrupt rate reduces to around 305 interrupts per second which should be easier to handle by any OS. The default value of the FIFO threshold is 1024 samples.

IMPORTANT NOTE: If both Scan and FIFO operation are enabled, then the interrupt will still occur at the programmed FIFO threshold, and the interrupt routine should read the indicated number or samples and then exit.

9.1.2 Scan Sampling

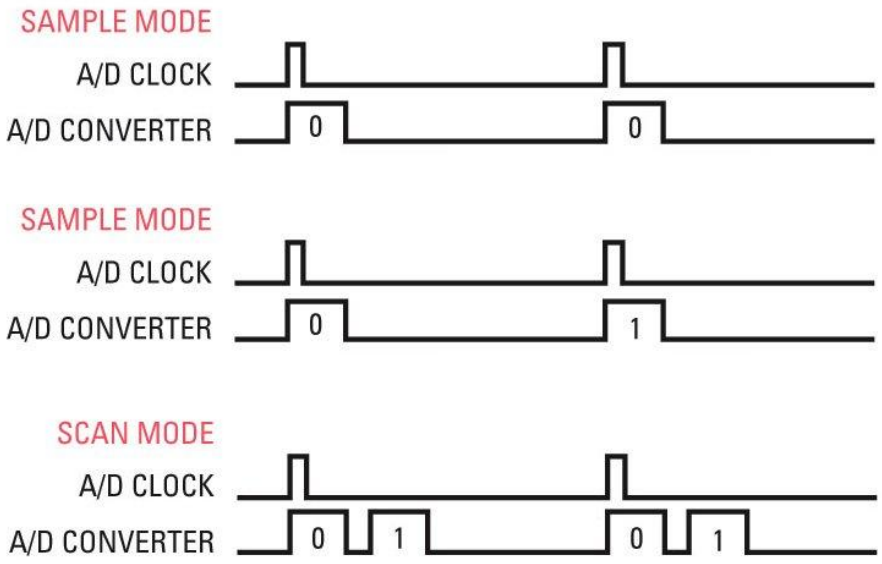
A scan is defined as a quick burst of samples of multiple consecutive channels. For example, you may want to sample channels 0-15 all at once, and repeat the operation each second. This would be a scan at a frequency of 1Hz. Each time the A/D clock occurs (software command, timer, or external trigger), all 16 channels are sampled in high-speed succession. There is a short delay of 500 to 2000 nanoseconds between each sample in the scan. Since each clock pulse causes all channels to be sampled, the effective sampling rate for each channel is the same as the programmed rate, and the total sampling rate is the programmed sampling rate times the number of channels in the scan range.

Scan sampling is independent of FIFO operation. Either or both can be enabled independently.

See following diagram illustrating Scan Sampling and Sequential Sampling.

9.1.3 Sequential Sampling

In sequential sampling, each clock pulse results in a single A/D conversion on the current channel. If the channel range is set to a single channel (high channel = low channel), each conversion is performed on the same input channel. If the channel range is set to more than one channel (high channel > low channel), then the channel counter increments to the next channel in the range, and the next conversion is performed on that channel. When a conversion is performed on the high channel, the channel counter resets to the low channel for the next conversion. The intervals between all samples are equal. Since each clock pulse results in only one channel being sampled, the effective sampling rate is the programmed sampling rate divided by the number of channels in the channel range.



10. HOW TO PERFORM A/D CONVERSIONS USING INTERRUPTS

The FP-DAQ1616 can generate hardware interrupts to manage A/D conversions. Interrupt-based A/D conversions are used in several situations:

- High-speed sampling
- Applications where the sampling rate must be precise
- Applications where the sampling rate is based on an external clock

Diamond Systems Universal Driver functions **dscADSampleInt()** and **dscADSetSettings()** manage all of the required parameters to generate interrupt-based A/D conversions. For more details on how to use the Universal Driver and the related functions, please refer to the Universal Driver manual.

1. **A/D channel range (low channel, high channel)**

   On the FP-DAQ1616, the channel numbers range from 0 to 15. During interrupt-based A/D conversions, the channels being sampled must be consecutive in number. To sample only a single channel, set the low channel and high channel to the same channel number. To sample a range of channels, set the low and high channels accordingly.

2. **Input voltage range**

   During interrupt-based A/D conversions, the input voltage range must be the same for all channels. Select the input range from the list of codes in Chapter 6.

3. **A/D Clock source, internal or external**

   For internal clocking, the on-board 32-bit counter/timer is programmed to the desired sample rate. For external clocking, the signal on external clock pin ADTRIG (which is on pin 160 of the FeaturePak connector) controls sampling. Falling edges on this pin will generate A/D conversions. The signal is edge sensitive, so holding it low will generate only one conversion.

4. **A/D conversion rate, if using internal clock**

   If internal clocking is selected, provide the desired sample rate in Hz as a floating value. The maximum sample rate is 2,000,000 per second (maximum A/D operating speed), and the slowest rate is .00116 Hz (5MHz input / 0xFFFFFFFF), or approximately 1 sample every 858 seconds.

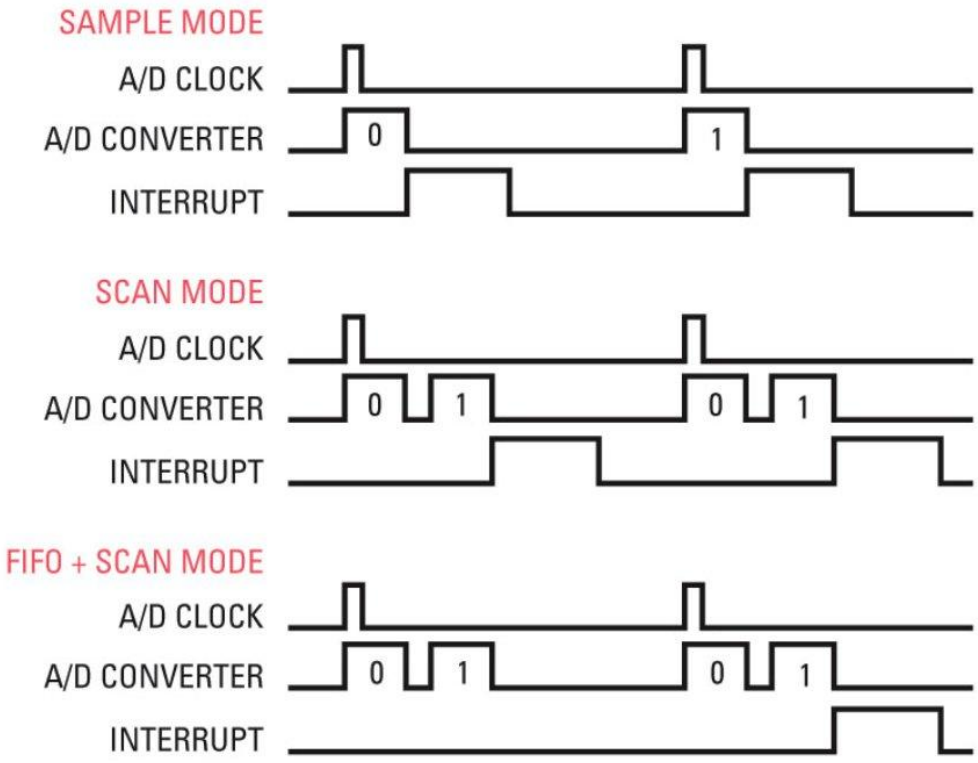
5. **External gating enable**

   When external gating is enabled, the A/D block will only sample the A/D channels when pin 152 AUX4 (ADGATE) is high. The external gating needs to be enabled by setting the ADGATEN bit to 1.

The table below summarizes the FIFO operation with interrupts.

FIFOEN	SCANEN	Action
0	0	Interrupt occurs after each A/D conversion completes (ADBUSY goes low). Only most recent A/D value is available. A/D value is stored in sample storage block.
0	1	Interrupt occurs after each A/D scan completes (ADBUSY goes low). Only most recent A/D scan is available. A/D values are stored in sample storage block.
1	0	Interrupt occurs when A/D conversion completes and FIFO threshold is reached **or exceeded.** A/D data is stored in the FIFO.
1	1	Interrupt occurs when A/D scan completes **and** FIFO threshold is reached or exceeded. A/D data is stored in the FIFO.

The following illustration shows timing when using interrupts with conversions:



11. ANALOG OUTPUT RANGES AND RESOLUTION

11.1 Description

The FP-DAQ1616 uses a 16-channel 16-bit D/A converter (DAC) to provide 16 analog outputs. A 16-bit DAC can generate output voltages with the precision of a 16-bit binary number. The maximum value of a 16-bit binary number is 2^{16} - 1, or 65535, so the full range of numerical values that you can write to the analog outputs on FP-DAQ1616 is 0 – 65535.

Note: In this manual, the terms analog output, D/A, and DAC are all used interchangeably to mean the same thing.

11.2 Resolution

The *resolution* is the smallest possible change in output voltage. For a 16-bit DAC the resolution is 1/(2^{16}), or 1/65536, of the full-scale output range. This smallest change results from an increase or decrease of 1 in the D/A code, and so this change is referred to as 1 LSB, or 1 least significant bit. The value of this LSB is calculated as follows:

1 LSB = Maximum voltage swing / 65536 (16-Bit)
The maximum voltage swing is defined as the difference between the highest nominal output voltage and the lowest output voltage. For an output range of 0-10V or +/-5V, the maximum voltage swing is 10V.

Example:

Output range = +/-5V

Maximum voltage swing = 10V

1 LSB = 10V / 65536 = 152.6uV (16-Bit)

11.2.1 Full-Scale Range Selection

The D/A converter on the FP-DAQ1616 requires two references, one for the low end and one for the high end of the range. The high end can be set to 5V or 10V, and the low end can be either 0V (for unipolar output ranges) or minus the high-end voltage (-5V or -10V). On power up, the D/A automatically resets to the range and polarity set by the D/A configuration registers at BAR0+20 and 21 and with the output voltage set to 0V.

The D/A converter is divided into a set of 2 separate D/A channel groups. Each channel group has 8 D/A channels in it. When the D/A converter is addressed, the values for the configuration settings and the D/A command are applied to the channel group selected.

12. GENERATING AN ANALOG OUTPUT

This chapter describes the steps involved in generating an analog output (also called performing a D/A conversion) on a selected output channel using direct programming (not with the driver software).

The D/A block allows you to configure the output range of the D/A operation. The D/A configurations such as polarity and gain can be set using registers in the D/A block at offset 21 as shown in the table below.

Offset from BAR0	7	6	5	4	3	2	1	0
16	DA7-0							
17	DA15-8							
18								
19								
20	DACMD7-0							
21					DABG1	DABG0	DAAG1	DAAG0
22							DAB2C	DASIM
23	DARESET					DAUPDT	DACLR1	DACLR0
24	DABUSY	DAOVF						
25	DACA7	DACA6	DACA5	DACA4	DACA3	DACA2	DACA1	DACA0
26	DAWCH3	DAWCH2	DAWCH1	DAWCH0			DACA9	DACA8
27	DEPTH3	DEPTH2	DEPTH1	DEPTH0	WGCH1	WGCH0	WGSRC1	WGSRC0
28					WGINC	WGRST	WGPS	WGSTRT
29								
30								
31								

The D/A block can be configured to operate in either Unipolar or Bipolar modes of operation. The mechanism to change the operational mode is described in this section.

The various modes that can be set in the D/A converter are determined by the DABG1-0/DAAG1-0 bits in conjunction with the polarity value. Available configurations are shown in the following table.

Polarity	DA#G1	DA#G0	Description
Unipolar	0	0	*INVALID*
Unipolar	0	1	0-10V
Unipolar	1	0	0-5V
Unipolar	1	1	0-2.5V
Bipolar	0	0	+/- 10V (Default Power up mode)
Bipolar	0	1	+/- 5V
Bipolar	1	0	+/- 2.5V
Bipolar	1	1	+/- 1.25V

WARNING	As noted in the table above, when the gain values are 0 and the selected mode is Unipolar, the D/A converter goes into an Invalid mode. If the D/A registers containing the DA Code are programmed to 0xFFFF, then this mode will drive output on the selected channels to go beyond the 10V limit, up to 15V. If this will cause an issue with the connected circuitry, proper care MUST be taken by the software to prevent this condition from happening.

The D/A block is divided in two channel groups called Channel Group1 and Channel Group2. Group1 provides access to D/A channels 0-7 while Group2 provides access to D/A channels 8-15. In order to change the settings for operation, the desired channel group needs to be addressed in the command.

In general, for any operation to be performed on the D/A converter, the D/A data registers and the D/A command registers need to be used. The sequence of operations is:

- Write data to DA15-0 registers at BAR0+ 16 and 17.
- Write the command to send to the D/A chip at the D/A command register at BAR0+20.

There are four steps involved in performing a D/A conversion:

1. Configure the desired output range
2. Set D/A Simultaneous Update bit (DASIM in BAR0+22)
3. Send the command to update D/A value
4. Monitor the DACBUSY status bit

12.1 Configure the Desired Output Range

To change the polarity of the D/A operation, the offset registers of the D/A block should be changed. The table below describes the different values to write for different modes of operations.

D/A Polarity	Channel Group	DACMD7-0	Data
Unipolar	CH Group1 (CH0-7)	0x02	0x0000
	CH Group2 (CH8-15)	0x03	
Bipolar	CH Group1 (CH0-7)	0x02	0x2000
	CH Group2 (CH8-15)	0x03	

The following code samples describe the different commands and data pattern to write for configuring the D/A block.

- To set D/A channel group1 in Unipolar mode, use the following commands

```
outp ( base + 16 , 0x00 ) ;        // Data LSB = 0x00
outp ( base + 17 , 0x00 ) ;        // Data MSB = 0x00
outp ( base + 20 , 0x02 ) ;        // Data MSB = 0x00
```

- To set D/A channel group2 in Unipolar mode, use the following commands

```
outp ( base + 16 , 0x00 ) ;        // Data LSB = 0x00
outp ( base + 17 , 0x00 ) ;        // Data MSB = 0x00
outp ( base + 20 , 0x03 ) ;        // Data MSB = 0x00
```

- To set D/A channel group1 in Bipolar mode, use the following commands

```
outp ( base + 16 , 0x00 ) ;        // Data LSB = 0x00
outp ( base + 17 , 0x20 ) ;        // Data MSB = 0x20
outp ( base + 20 , 0x02 ) ;        // Data MSB = 0x00
```

- To set D/A channel group2 in Bipolar mode, use the following commands

```
outp ( base + 16 , 0x00 ) ;        // Data LSB = 0x00
outp ( base + 17 , 0x20 ) ;        // Data MSB = 0x00
outp ( base + 20 , 0x03 ) ;        // Data MSB = 0x00
```

Apart from polarity, the gain value can be changed for the specific channel group also. The gain value can be changed by writing to the DABG1-0 and DAAG1-0 bits in the D/A gain register at offset 21.

The gain value can be set to a value of 1, 2, 4, or 8 which correspond to the bits programmed as 0-3 where 0 represents a gain of 1 and 3 represents the gain of 8.

- To set the gain value on channel group1

```
gain_value = inp ( base + 21 ) ;  // read the current settings of the register
gain_value = gain_value | ch_group1_gain ;        // ch_group1_gain is either 0-3
outp ( base + 21 , gain_value ) ; // preserve gain settings for ch_group2
```

- To set the gain value on channel group2

```
gain_value = inp ( base + 21 ) ;  // read the current settings of the register
gain_value = gain_value | ( ch_group2_gain << 3 ) ;        // ch_group2_gain is either 0-3
outp ( base + 21 , gain_value ) ; // preserve gain settings for ch_group1
```

Care should be taken while changing the gain values for a channel group so that the gain settings for the other channel group are not overwritten. The above example describes how to achieve that.

12.2 Set D/A Simultaneous Bit

If the application requires that all the D/A channels be updated at the same time, the D/A Simultaneous Update bit DASIM should be set to 1. If it is set to 1 and the D/A conversion command is sent to the register, the D/A channel does not get updated until the bit DAUPDT is set to 1.

To set the D/A simultaneous bit:

```
current_value  = inp ( base + 22 ) ;        // store the rest of the register settings
outp ( base + 22 , (current_value | 0x01 ) ); // set the DASIM bit to 1
```

12.3 Send Command to Update D/A Value

The D/A value to be output on a channel needs to be converted into an LSB and an MSB and written to the D/A data registers at offset 16 and 17.

Use the following formulas to compute the LSB and MSB values from the D/A code:

LSB = D/A Code & 0x00FF (keep only the low 8 bits)

MSB = (D/A code & 0xFF00) >> 8

Output code = 1776 stored in a variable called da_code

```
LSB = da_code & 255 = 240 (0xF0)
MSB = ( da_code & 0xFF00 ) >> 8  = 0x06
outp ( base + 16 , LSB ) ;
outp ( base + 17 , MSB ) ;
```

Once the data is written to the DA15-0 registers, a D/A command should be executed based on the channel number and channel group being addressed.

The command to update a D/A channel in channel group1 is 0xC8 – 0xCF for channel number 0 – 7. The command to update channel 0 is 0xC8. For every channel the command number increases by 1. Thus the command to update any channel is 0xC8 + channel number (0 to 7)

Similarly, the command to update a D/A channel in channel group2 is 0xD0 – 0xD7 for channel number 8 – 15. The command to update channel 0 is 0xD0.For every channel the command number increases by 1. Thus the command to update any channel is 0xD0 + channel number (8 to 15) - 8.

For example:

```
To output a value to D/A channel 0:
outp ( base + 21 , 0xC8 ) ;        // 0xC8 – CH# group1 command offset
To output a value to D/A channel 8:
outp ( base + 21 , 0xD0 ) ;        // 0xD0 – CH# group2 command offset
```

If the DASIM bit is enabled, then sending the D/A value to a D/A channel will not reflect in the output voltage on the selected D/A channel. In this case, the bit DAUPDT must be set to 1 to update all the channels simultaneously.

```
outp ( base + 23 , 0x04 ) ;          // DAUPDT is bit 2 in the register at BAR0+23.
```

12.4 Monitor DABUSY Signal

The D/A convert operation results in the block to generate a DABUSY signal available in register at BAR0+24. When this bit is high, it indicates that the block is busy. This bit should be monitored before sending any command to the D/A chip.

```
while ( inp ( base + 24 ) & 0x80 ) ;          // wait till DABUSY is 1.
```

12.5 D/A Waveform Generator

12.5.1 Description

The registers BAR0+25 – BAR0+28 provide control for the D/A waveform generator. The D/A waveform generator uses an in-FPGA memory block of 1024 words to store D/A codes. The FPGA parses through this memory at a user-programmable speed (or through manual/external trigger) while sending codes to the D/A converter. The generator automatically stops if enhanced features are disabled.

The generator works in frames. A new frame is triggered from a programmable source (manual, counters, external, etc.) For each frame, the FPGA sends a programmable (1, 2 or 4) number of D/A codes from the generator's memory bank straight to the DAC. This transfer is done in latched mode, and the DAC is updated after all codes in a frame are sent. The generator continues this process, incrementing through the memory until it reaches the end of the buffer, or hits a programmable depth – at which point it will wrap back to the beginning of the buffer and continue operation. The generator can be paused, resumed or reset to the beginning of the memory bank at any time.

With the use of the memory block, the D/A waveform generator can output consistent waveforms at a maximum frequency of 100KHz. There are four different input sources available for the D/A waveform generator: manual software trigger, counter 0 output, counters 1 output, and external trigger. The memory block also allows a programmable depth which when hit will wrap and return to the beginning. The threshold ranges from 64 to 1024 and is programmable in multiples of 64.

12.5.2 Programming the D/A Waveform Generator

This section details how to program the D/A waveform generator through direct I/O without using the driver software.

There are 4 steps to programming the D/A waveform generator

1. **Reset D/A waveform pointer**
2. **Store D/A values into buffer**
3. **Setup D/A waveform settings**
4. **Start D/A waveform generator**

12.5.3 Reset D/A Waveform Pointer

Reset the D/A waveform pointer by accessing, register BAR0 + 28, bit 2. Writing a 1 to this bit and cause the pointer to start at the beginning, address 0.

12.5.4 Store D/A Values into Buffer

The D/A value needs to be stored in the FPGA waveform memory. The D/A code to be output should be programmed in the D/A data registers at BAR0+16 and BAR0 + 17 before writing the D/A waveform channel and MSbit register at BAR0+26. When the BAR0 + 26 register is written, the D/A code that is present in the D/A data register bits DA15-0, will be stored in the D/A waveform memory.

12.5.5 Setup D/A Waveform Settings

D/A waveform settings include input source, number of code per frame, and threshold. Each can be set individually and in any combination.

There are four different input sources to choose from: manual/software trigger, counter 0 output, counters 1 output, and external trigger. Manual trigger should be used when the rate is slow or inconsistent and needs be controlled in software. Counter 0 output should be used when a consistent rate is desired. External trigger should be used when an external signal is desired to generate D/A waveform. Input source is set by bits 0 and 1 on BAR0 + 27.

Number of code per frame determines the number of buffer values that will be output per frame. Each code is determined by the value set at its address.

For example, if the codes per frame option is set at 2, the first frame will output the codes at address 0 and 1, then 2 and 3, then 4 and 5 and so on. Number of code per frame is set by bits 2 and 3 on BAR0 + 27.

Threshold determines the number of code to output before the pointer starts over. The threshold must be set in multiples of 64 up to 1024. When the threshold is hit, the pointer wraps around and starts at the beginning. Threshold is set by bits 4, 5, 6, and 7 at BAR0 + 27.

12.5.6 Start D/A Waveform Generator

Initialize D/A waveform output by writing 1 to bit 0 at BAR0 + 28. The generator will continue to output the periodic waveform until you disable it.

13. AUTOCALIBRATION

The FP-DAQ1616 features automatic calibration of both analog inputs and outputs. No potentiometers, which are subject to tampering, vibration, and maladjustment, are used. Instead, all calibration adjustments are performed using an octal 8-bit TrimDAC and precision, low-drift reference voltages on the board. The optimum TrimDAC values for each input range are stored in an EEPROM and recalled automatically on power up.

The calibration and EEPROM/TrimDAC related access can be achieved using registers at BAR0+ 80 to 95 as shown in the table below.

Offset from BAR0	7	6	5	4	3	2	1	0
80	ECA7-0							
81	ECA15-8							
82								
83						EEDEV2	EEDEV1	EEDEV0
84	ECD7-0							
85								
86						REFSEL1	REFSEL0	CMUXEN
87 (C)	EEEN	EERW	LOADCAL	-	TDACWR	-	-	-
87 (R)	EEBUSY	-	-	-	TDBUSY	-	-	-
88	Unlock code A							
89								
90								
91								
92	(Reserved for auto-autocal)							
93	(Reserved for auto-autocal)							
94	(Reserved for auto-autocal)							
95	(Reserved for auto-autocal)							

ECA15-0	EEPROM address to read/write or TrimDAC number depending on the device selected
EEDEV2-0	Reserved -- Must be 0
ECD7-0	EEPROM or TrimDAC data to be read/written on the selected device
REFSEL1-0	Selection of on-board reference voltages for various ranges
CMUXEN	Enable on-board multiplexer for routing reference voltages on A/D channels
EEEN	EEPROM enable bit
EERW	EEPROM read/write action
LOADCAL	Reload the calibration values from EEPROM to TrimDACs
TDACWR	Write command to update the selected TrimDAC with the Trim DAC data
EEBUSY	EEPROM busy indicator
TDBUSY	Trim DAC busy indicator
Unlock Code A	Provide write access to the EEPROM

To calibrate the board through software, Diamond provides a calibration utility program and software driver function that enables you to calibrate the analog inputs and outputs at any time for any range and store the settings in the EEPROM. This feature dramatically improves the accuracy and reliability of the board, since you can calibrate the board as often as desired without worrying about temperature or time drift.

13.1 How Autocalibration Works

The autocalibration circuit uses an octal 8-bit TrimDAC IC to provide small adjustments to the offset and gain at various points in the circuit. Four of the DACs are used for the A/D calibration, and the other eight are used for the D/A. For the D/A circuit there are two sets of 4 TrimDACs for each channel group. The TrimDACs 4-7 are for Channel group1 (D/A channels 0-7) where as TrimDACs 8-11 are for channel group2 (D/A channels 8-15). The 8-bit TrimDAC values are stored in an on-board EEPROM and are recalled automatically on power-up.

An on-board ultra-stable +5V reference chip with 5ppm offset drift is used as the voltage reference for all calibration operations. From this reference several intermediate values are derived that are used for the calibration. One is just under +5V and one is just above 0V. These values are measured at the factory, and their values are stored in the on-board EEPROM for use by the calibration program. Note that the actual values of the reference signals does not matter, as long as they are stable, since the calibration routine knows the values and can adjust the calibration circuit to achieve them. An extra input multiplexer is used to feed the calibration voltages into the A/D circuit during the process.

For bipolar A/D calibration, first 0V is measured, and the TrimDAC is adjusted until the target A/D reading is achieved. For unipolar calibration, the voltage just above 0 is used as the first measurement value. Two TrimDAC channels are used for the offset. The first channel provides a coarse adjustment to bring the A/D readings into range, and then the second channel provides a fine adjustment for maximum accuracy. The use of both coarse and fine adjustments provides a wider range of total adjustment capability. The range of the fine adjustment exceeds the smallest change in the coarse adjustment, so there is no gap in the adjustment range.

After the offset is adjusted, the full-scale is adjusted in a similar manner. The reference value just under 5V is fed into the A/D, and two additional TrimDACs provide coarse and fine adjustments to achieve the target A/D near-full-scale reading.

Once the A/D is completely calibrated, the 16-bit D/A channels can be calibrated. Unlike the A/D circuit, which uses a single A/D for all input channels, the D/A circuit actually contains a single D/A converter for each of the 16 output channels. The D/A converter internally divides the 16 channels into two sets of 8 channels These channels are fed into the calibration multiplexor and the remaining 8 TrimDAC channels are used to calibrate them in a similar manner to the A/D. Both coarse and fine adjustments are used for the low as well as high reference.

The entire process takes about one second for each input range. Once it is complete, the board is ready to run. All 12 TrimDAC values are stored in the EEPROM so that the next time power is cycled to the board, the values will be loaded automatically.

The board always boots up from factory calibration in 10V BIPOLAR mode on A/D and D/A modes. Thus the results obtained would be best for 10V BIPOLAR mode for both A/D and D/A circuits.

13.2 How to Perform Autocalibration With Software

Diamond's Universal Driver software provides two functions, dscADAutocal() and dscDAAutocal(), that can be called from within a user program to calibrate the board at any time.

The API function dscADAutoCal provides A/D calibration on all the 16 A/D modes whereas the function dscDAAutoCal provides D/A calibration functionality on all the available D/A modes.

For details on how to use the functions, please refer to the Universal Driver user manual available online at **http://docs.diamondsystems.com/dscud/manual_Main+Page.html**.

14. DIGITAL I/O OPERATION

The FP-DAQ1616 has seven 8-bit bidirectional digital I/O ports, named A-F and AUX. The DIO block is controlled and configured using registers at BAR0+32 to 47 as shown in the table below.

Offset from BAR0	7	6	5	4	3	2	1	0
32	DIOA7-0							
33	DIOB7-0							
34	DIOC7-0							
35	DIOD7-0							
36	DIOE7-0							
37	DIOF7-0							
38								
39								
40 (W)							MODEA	DIRA
40 (R)	LATCHA						MODEA	DIRA
41 (W)							MODEB	DIRB
41 (R)	LATCHB						MODEB	DIRB
50 (W)							MODEC	DIRC
50 (R)	LATCHC						MODEC	DIRC
51 (W)							MODED	DIRD
51 (R)	LATCHD						MODED	DIRD
52	DIRE7	DIRE6	DIRE5	DIRE4	DIRE3	DIRE2	DIRE1	DIRE0
53	DIRF7	DIRF6	DIRF5	DIRF4	DIRF3	DIRF2	DIRF1	DIRF0
54								
55	ACK							

DIOn7-0 DIO port n data where n = A, B, C, D, E or F

DIRA-D Direction control bits for DIO ports A-D

0 = Port is Input

1 = Port is an Output

MODEA-D Mode control bits for DIO ports A-D

0 = port is normal DIO port

1 = port is in Latched mode

LATCHA-D Latch status bit for ports A-D when the port is configured in Mode 1 of operation

DIRE7-0 Direction control bits for DIO port E

0 = Respective bit is input

1 = Bit is output

DIRF7-0 Direction control bits for DIO port F

0 = Respective bit is input

1 = Bit is output

ACK ACK signal control for latched mode of operation in mode 1

Ports A-D are 8-bit ports with direction programmable byte by byte. Register bits DIRA-D control the direction of these ports and also the direction of the port pins where a value of 0 = input and 1 = output. In output mode, the values in these registers drive their associated I/O pins. The logic levels on the I/O pins may be read back in both input and output modes. These ports reset to 0 and input mode during power-up, reset, FPGARST=1, or BRDRST=1. If a port is in input mode, its output register may still be written to. When the port is switched to output mode, the value of the output register will drive the corresponding I/O pins.

Ports E-F are 8-bit ports with direction programmable bit by bit according to register bits DIRE7-0 and DIRF7-0 where a value of 0 = input, 1 = output. I/O pins DIOF3-0 may be reassigned as PWM outputs; see the PWM circuit description. I/O pins DIOF6 and DIOF7 may be reassigned as latch and acknowledge signals when MODEn = 1. See latched mode behavior below and the Interrupt section.

Port AUX is an 8 bit port with direction programmable bit by bit. Register bits AUXDIR7-0 control the direction of bits AUX7-0. 0 = input and 1 = output. These bits reset to 0 and input mode during power-up, reset, FPGARST=1, or BRDRST=1. If a bit is in input mode, its output register may still be written to. When the port is switched to output mode, the value of the output register will drive the output pins.

AUX7-4 have dual functions defined by additional control register bits. When these bits are 0, the corresponding I/O pins are normal I/O pins and behave as described above. When these bits are 1, the corresponding I/O pins change to other functions as defined below.

AUX Bit	Alternate Signal	Function	Direction	Control Bit
7	WDTIN	Watchdog timer in	In	WDTIEN
6	WDTOUT	Watchdog timer out	Out	WDTOEN
5	CTR1GATE	Counter 1 gate input	In	CTRCMD3-0=0011 & CCD0=1
4	ADGATE	A/D gate input	In	ADGATEN

14.1 Digital I/O Configuration Register

The direction control register is programmed by writing to the registers at offset 40-45. Once you have set the port directions with this register, you can read and write to the ports as desired. These registers allow you to set the direction as well as the mode of operation for the respective ports.

Below are several examples on how to configure various DIO ports.

To configure DIO port A as output port in Mode 1

```
outp( base + 40 , 0x03 ) ;// MODEA = 1 (bit 1) , DIRA=1 (bit 0)
```

To configure DIO port D as input port in Mode 0

```
outp ( base + 43 , 0x00 ); // MODED = 0 (bit 1) , DIRD=0 (bit 0)
```

To configure DIO port F bits 4-7 as input and bits 0-3 as output

```
Outp ( base + 45, 0x0F);
```

14.2 Mode 0 Digital I/O

This is the simpler of the two I/O modes and works well for most uses. In mode 0, the handshaking signals Latch and Ack are not used. When reading any port in input mode, the data at the I/O pins at the time of the read command will be returned. Only ports A-D need to be configured in mode 0 of operation. Ports E & F can only operate in mode 0.

14.3 Latched Mode Behavior and Interrupts on DIO

DIO ports A, B, C, and D may operate in latched mode. This mode enables handshaking signals to control the transfer of data between the board and an external device. Latch mode is enabled by setting MODEn = 1 where n = A, B, C, or D. All ports with MODEn = 1 operate in the same manner. To avoid undefined or undesired behavior, all DIO ports operating in latch mode should have the same direction setting.

When any MODEn bit is 1, pin DIOF6 is forced to input mode and operates as a latch signal and register bit DIOF6 reads as 0. Pin DIOF7 is forced to output mode and operates as an acknowledge signal ACK-, and register bit DIOF7 reads as 0.

(See the table on the following page)

MODEn	DINTEN	Action
0	0	General purpose digital I/O without latch function Ports A, B, C, and D operate as described in the basic DIO functional description.
0	1	Invalid setting; DINTEN is ignored unless at least one DIO port is set for Mode 1.

1	0	Ad hoc latched digital I/O operation
		When the Direction of the port# n = input: (where n = A,B,C or D)
		Falling edge on DIOF6 latches data on all ports where MODEn = 1. This sets LATCHn = 1 for all ports where MODEn = 1.
		No IRQ is generated.
		Software should monitor LATCHn status bits to determine when data is available.
		After software reads data, it should pulse DIOF7 low by writing 1 to the ACK bit. This also clears LATCHn status bits.
		outp (base + 47 , 0x80) ; // ACK=1
		When the Direction of the port #n = output: (where n = A,B,C or D)
		After software writes data, it should pulse DIOF7- low by writing 1 to the ACK bit. This also clears LATCHn status bits.
		External device should pulse DIOF6 low to acknowledge receipt. This sets LATCHn = 1 for all ports where MODEn = 1.

1	1	Interrupt-driven latched digital I/O operation When the Direction of Port # n = input: (where n = A,B,C or D) DIOF6 falling edge latches data on all ports where MODEn = 1. This sets LATCHn = 1 for all ports where MODEn = 1. This also sets DINT=1 and generates an IRQ. After ISR reads data, it should pulse DIOF7 low with DINTCLR = 1. This also clears LATCHn status bits and clears DINT. When the Direction of Port # n = output: (where n = A,B,C or D) After ISR writes data, it pulses P_ACK- low with DINTCLR = 1. This also clears LATCHn status bits and clears DINT. External device pulses DIOF6 low to acknowledge receipt. This sets LATCHn = 1 for all ports where MODEn = 1. This also sets DINT=1 and generates an IRQ. During setup, software writes initial values to port(s) n and should pulse DIOF7 low one time with DINTCLR = 1 or ACK = 1.

14.3.1 Auxiliary Digital I/O

FP-DAQ1616 contains 8 bits of auxiliary digital I/O. The auxiliary port is a bit by bit configurable port just like DIO ports E & F. By default this port operates as a normal mode 0 digital I/O port. When in special function mode, the bits 4-7 of this port operate in their dual function mode as described in the previous sections. The bits 0-3 can still be used as normal DIO bits.

15. COUNTER/TIMER OPERATION

The FP-DAQ1616 provides two 32-bit counter/timers. The counter/timers can be used using the registers at offset 48-52 as shown in the table below.

Offset from BAR0	7	6	5	4	3	2	1	0
48	CTRD7-0							
49	CTRD15-8							
50	CTRD23-16							
51	CTRD31-24							
52	CTRN7-0							
53	CTRCMD3	CTRCMD2	CTRCMD1	CTRCMD0			CCD1	CCD0

CTRD31-0 Counter data to provide the counter duration

CTRN7-0 Counter number register (0 or 1)

CTRCMD3-0 Counter command bits

CCD1-0 Additional control bits for counter behavior

15.1 Counter/Timer Features and Configuration Options

Counter 0 can be used as a programmable A/D sampling clock. If not being used for A/D sampling, these counter/timers may be used for other functions. Counter/timer 1 is always available for user applications.

The inputs of the counter/timers are programmable, and the outputs may be routed to the I/O header under software control. The table below lists the key features of each counter/timer:

Counter/Timer Configuration Options

Counter	Input	Gate	Output
0	50MHz on-board 5MHz on-board External clock on pin 159	NOT AVAILABLE	NOT AVAILABLE
1	50MHz on-board 5MHz on-board External clock on pin 158	Gate EN on pin 151	Available on pin 157 of FeaturePak connector

15.2 Counter/Timer Configuration

The counter/timer configuration is determined by the control register at BAR0 + 53 in the Counter Command Register as described in the register description. Note that the outputs of counter 1 can be routed to pin 157 under software control rather than being hardwired.

Configuring the A/D sampling clock is done with the control register at BAR0+ 10 in the A/D scan register described in the register description section. Bits ADCLK1-0 select whether the A/D hardware clocking is enabled, and if so, bits ADCLK1-0 select whether it is the output of counter/timer 0 or the external clock input at Extclk.

15.3 Counter/Timer Access and Programming

All the programming information regarding using the counter/timer of the FP-DAQ1616 board is given in the Counter block register descriptions. The counter programming registers are available from offset 48 to 53 from the base address of BAR0.

To program the counter timer, the following sequence of operations needs to be performed.

1. Write the counter number to the counter number register.
2. Write the count value to the counter data register. This value should be calculated based on the desired counter value as a divisor from the clock source used.

   For example if the clock source for the counter is 50MHz clock and the desired counter value is 10KHz, the value that should be written to the counter data registers is 50,000,000/10,000 = 5000 decimal or 0x1388.

   ```
   outp ( base + 30 , 0x88 ) ;        // LSB of counter data
   outp ( base + 31 , 0x13 ) ;        // MSB of counter data
   ```

3. Write the counter command to the command register to execute the desired function.

   For example, to enable counter 0

   ```
   outp ( base + 34 , 0x41 ) ;        // enable and start counting on the selected counter.
   ```

   More information on individual commands is provided in the sections below.

15.3.1 Counter/Timer commands

The counter/timers can be controlled using various commands shown in the table below.

CTRCMD3-0				Command	Control Bits		Action	Register Value
3	2	1	0		CCD 1	CCD 0		
0	0	0	0	Clear Counter.	X	X	-	0x00
0	0	0	1	Load the selected counter with data in CTRD32-0.	X	X	-	0x10
0	0	1	0	Select Count Direction.	X	1	Count up	0x21
					X	0	Count down	0x20
0	0	1	1	Enable / Disable External gate. When this command is selected for Counter 1, DIO pin AUX5 is reconfigured as an input and used for Counter 1 gate.	X	1	Enable Gating	0x31
					X	0	Disable Gating	0x30
0	1	0	0	Enable/Disable counting.	X	1	Enable Counting	0x41
					X	0	Disable Counting	0x40
0	1	0	1	Latch Selected counter.	X	X	-	0x50
0	1	1	0	Select Counter clock source.	0	X	Counter Input pin	0x60
					1	0	Internal CLK 50MHz	0x62
					1	1	Internal CLK 5MHz	0x63
1	1	1	1	Reset selected or both counters.	X	0	Reset selected counter	0xF0
					X	1	Reset both counters	0xF1

The programming details for each of the commands are as follows.

15.3.1.1 CLEAR COUNTER

To clear the counter, the following sequence should be used.

```
BYTE counter_number , counter_command ;


outp ( base + 34 , counter_number ) ;     // counter_number = 0 or 1
counter_command = 0x00 ;                  // clear the selected counter.
outp ( base + 35 , counter_command ) ; // issue the clear command
```

15.3.1.2 LOAD COUNTER

To load the counter, the command 0x10 should be used as below.

```
BYTE counter_number, counter_command ;
// again using an example of loading the value of 0x1388 in counter 0.
counter_number = 0 ;                      // or 1
outp ( base + 34 , counter_number ) ;
outp ( base + 30 , 0x88 ) ;               // counter LSB
outp ( base + 31 , 0x13 ) ;               // counter MSB
outp ( base + 35 , 0x10 ) ;               // Load the counter selected
```

15.3.1.3 SELECT COUNT DIRECTION

The count direction for the selected counter can be set using the control bits CCD1-CCD0 as explained in the table above. The command to use is either 0x20 or 0x21 and the sequence is as below.

- To set the counter 0 in down count mode.

```
outp ( base + 34 , 0 );                   // select counter 0
outp ( base + 35 , 0x20 ) ;               // down count direction
```

- To set the counter 1 in up count mode.

```
outp ( base + 34 , 1 );                   // select counter 1
outp ( base + 35 , 0x21 ) ;               // down count direction
```

15.3.1.4 ENABLE/DISABLE GATING

Gating can be enabled on counter 1 only and the sequence to enable/disable gating is as below.

```
outp ( base + 34 , 1 ) ;                  // select counter 1
outp ( base + 35 , 0x31 ) ;               // enable gating
alternatively to disable gating…
outp ( base + 35 , 0x30 ) ;               // disable gating
```

15.3.1.5 ENABLE/DISABLE COUNTING

After the counter data is loaded in the counter register for the selected counter as shown in the section 14.4.2 of this document, the counter MUST be enabled to start the counter. After the counter has been started, it can be disabled using a different command as below.

// load the counter as in section 14.4.2. Do not change the counter select register in base + 34.

```
outp ( base + 35 , 0x41 ) ;               // start the selected counter.
```

Alternatively to stop the counter…

```
outp ( base + 35 , 0x40 ) ;               // stop the selected counter.
```

15.3.1.6 LATCH AND READ COUNTER

To read the counter value, it MUST be latched first otherwise the contents of the counter registers would not represent the current value of the counter.

The sequence to latch and read the counter 0 is as below.

```
BYTE ctr_byte0, ctr_byte1, ctr_byte2, ctr_byte3 ;


outp (  base + 34 , 0 ) ;              // select counter 0
outp ( base + 35 , 0x50 ) ;            // issue latch command
ctr_byte0 = inp ( base + 30 );         // First byte of counter data
ctr_byte1 = inp ( base + 31 );         // Second byte of counter data
ctr_byte2 = inp ( base + 32 );         // Third byte of counter data
ctr_byte3 = inp ( base + 33 );         // Fourth byte of counter data
```

15.3.1.7 SELECT COUNTER SOURCE

The counter source needs to be selected before actually starting a counter. As it can be seen from the table, there can be three different clock sources which the counter can use to performing counting operations. The clock source selection for the selected counter can be performed using only the corresponding command write to the counter command register.

```
outp ( base + 35 , 0x62 ) ;            // use 50 Mhz clock as source for the selected counter
```

Alternatively

```
outp ( base + 35 , 0x63 ) ;            // use 5 Mhz clock as source for the selected counter
```

Or

```
outp ( base + 35 , 0x60 ) ;            // use external clock source for counter clock
```

15.3.1.8 RESET COUNTER

The register map allows either resetting an individual counter or both the counters on the board. When the reset command is executed on a counter, the counter data resets to a value of 0.

To reset an individual counter (for example , counter 0)

```
outp ( base + 34 , 0 ) ;               // select counter 0
outp ( base + 35 , 0xF0 ) ;            // issue reset selected counter command
```

Alternatively, to reset both the counters, the counter selection register at Base + 34 is ignored and the command to write is

```
outp (  base + 35 , 0xF1 ) ;           // issue reset counter command to both counters.
```

15.4 Timer interrupts

The FP-DAQ1616 counters can be used to generate interrupts on the PCI bus at a regular programmable rate. To use the interrupts, the following should be done.

- Select the counter to use.
- Write the count value in the counter data register for the timing that is required.
- Enable the respective Timer interrupt enable bit. For timer 0 use T0INTEN bit and set it to 1. For timer 1, use T1INTEN bit and set it to 1. Please note that both the timers cannot be used at the same time to generate interrupts.
- Enable the respective counter/timer.

- Upon interrupt, check for the respective timer interrupt by reading the status bit T0INT for timer 0 and T1INT for timer 1.
- To generate the next interrupt, reset the respective timer interrupt request flip-flop by writing 1 to either T0INTCLR or T1INTCLR bits. If this bit is not reset in the interrupt service routine, the board will not generate any more interrupts.

16. PWM OPERATION

The FP-DAQ1616 board has four PWM timers, each with a 24-bit resolution. The PWM circuit works very similarly to the counter/timer circuit and has similar register architecture.

The PWM circuit is controlled using the registers at offset 56 to 60 as shown in the table below.

Offset	7	6	5	4	3	2	1	0
56	PWMD7-0							
57	PWMD15-8							
59	PWMD23-16							
60	PWCMD3	PWCMD2	PWCMD1	PWCMD0	PWMCD	PWM2	PWM1	PWM0

PWMD23-0 24-bit PWM data. This is the PWM data for period as well as duty cycle

PWMCMD3-0 PWM Command byte

PWMCD Additional control bit for use by certain commands along with PWMCMD3-0

PWM2-0 PWM circuit to use

Each PWM consists of a pair of 24-bit down counters named C0 and C1. The C1 counter defines the duty cycle (active portion of the signal), and the C0 counter defines the period of the signal. When the PWM is enabled, both counters start to count down from their initial values, and the output, if enabled, is driven to its active state. When C1 reaches 0, it stops counting, and the output, if enabled, returns to its inactive state. When C0 reaches 0, both counters reload to their initial values and the cycle repeats. If C1 = 0 then duty cycle = 0. If C1 = C0, then duty cycle = 100% (the output should be glitch free).

The PWM command register has two fields, namely PWM Command in bits PWCMD3-0, and PWM circuit number in bits PWM2-0. The bit PWMCD is additional data for use by certain commands. The default setting for all parameters is 0 since the default / reset value for all registers in this circuit is 0.

If a PWM output is not enabled, its output is forced to the inactive state, which is defined as the opposite of the value selected with command 0010 (in PWMCMD3-0). The PWM may continue to run even though its output is disabled.

PWM outputs may be made available on Auxiliary DIO port's I/O pins DIOF0 to DIOF3 by writing a command value of 0101 to the PWM command register. When a PWM output is enabled, the corresponding pin on DIOFn is forced to output mode regardless of the DIRFn direction control bit. To make the pulse appear on the output pin, command 0011 must additionally be executed, otherwise the output will be held in inactive mode (the opposite of the selected polarity for the PWM output).

17. SPECIFICATIONS

Host Interface	
Interface type	PCI Express x1
Analog Inputs	
Number of input channels	16 single-ended or 8 differential voltage inputs (software selectable)
Resolution	16 bits or 12 bits (1/4096 of full scale), user selectable
Input ranges	Bipolar: ±10V, ±5V, ±2.5V, ±1.25V, ±0.625V Unipolar: 0-10V, 0-5V, 0-2.5V, 0-1.25V, 0-0.625V
Input bias current	40nA max
Input impedance	10^13 ohms
Maximum input voltage	±10V for linear operation
Over-voltage protection	-25V to +40V on any input without damage
Drift	±10ppm/^{o}C typical
A/D conversion rate	2,000,000 maximum aggregate samples/sec max 1,000,000 samples/sec for multi-channel operation
Conversion trigger	Software trigger Internal pacer clock External digital signal
FIFO	16K (16384) 16-bit samples with programmable interrupt threshold
Accuracy	< ±2 LSB, after calibration
Nonlinearity	< ±3 LSB, no missing codes
Calibration	Automatic; values stored in EEPROM
Analog Outputs	
Number of output channels	16 voltage outputs
D/A resolution	16 bits (1/65,536 of full scale)
Output ranges	Unipolar: 0-10V, 0-5V, 0-2.5V Bipolar: ±10V, ±5V, ±2.5V, ±1.25V
Output load current	±1mA max per channel
Output short circuit current	15mA max per channel
Transition time	1V/us typical
Relative accuracy	±1 LSB channel to channel
Nonlinearity	±1 LSB, monotonic
Reset	All channels reset to 0V
Calibration	Automatic; values stored in EEPROM
D/A Waveform Generator	Output consistent waveforms at up to 100KHz

Digital I/O	**Ports A, B, C, D**	
Number of I/O lines	32	
Compatibility	LVTTL and TTL	
Pull-up / pull-down	Programmable 10Kohm	
Input voltage	Low: 0.0V min, 0.8V max High: 2.0V min, 5.5V max	
Input current	+/-340µA max	
Output voltage	Low: 0.0V min, 0.4V max High: 2.4V min, 3.3V max	
Output current	Low: 16mA max High: -16mA max	
Digital I/O	**Ports E, F, AUX**	
Number of I/O lines	24	2
Compatibility	LVTTL	LVTTL
Pull-up / pull-down	Programmable 10Kohm for port E	
Input voltage	Low: -0.3V min, 1.0V max High: 2.0V min, 3.6V max	Low: -0.3V min, 1.0V max High: 2.0V min, 3.6V max
Input current	+/-340µA max	+/-340µA max
Output voltage	Low: 0.0V min, 0.4V max High: 2.7V min, 3.3V max	Low: 0.0V min, 0.4V max High: 2.7V min, 3.3V max
Output current	Low: 12mA max High: -12mA max	Low: 12mA max High: -12mA max
Counters/Timers		
A/D pacer clock	32-bit up/down counter 50MHz, 5MHz, or external clock input	
General purpose counter	32-bit up/down counter 50MHz, 5MHz, or external clock input	
PWMs		
PWMs	Four independent 24-bit PWMs with user selectable clock at either 1MHz or 50MHz	
General		
FeaturePak	Compliant, zero height expansion	
Operating temperature	-40°C to +85°C	
Power Consumption	840mW (typical without external load)	
Weight	0.5 oz (14.2 g)	
RoHS Compliant	Yes	