DM35425HR

PCI Express Data Acquisition Board

User's Manual

BDM-610010047 Rev D

Revision History

Rev A	Initial Release
Rev B	Add Parallel Bus set up note Add Power Consumption Updated IDAN Dimensions Picture Added a Channel Delay Section Added Equation 1
Rev C	Corrected Signal Names in IDAN Pin Out
Rev D	Update Register Map to FPGA Rev C i. ADC Function Block changed to 0x01031000 ii. DAC Function Block changed to 0x01032000 iii. ADIO Function Block changed to 0x01003001 iv. Add CH_FIFO_ACCESS register to ADC, DAC and ADIO Add Section 5.6 External Clocking Add Clock Source description

RTD Embedded Technologies, Inc.
103 Innovation Boulevard
State College, PA 16803 USA
Telephone: 814-234-8087
Fax: 814-234-5218

www.rtd.com

sales@rtd.com
techsupport@rtd.com

Table of Contents

Table of Figures

Table of Tables

1 Introduction

1.1 Product Overview

The DM35425 is a software configurable high-speed, 12-bit data acquisition module in the PCIe/104 format. This module provides 16 differential or 32 single-ended analog input channels, with programmable gain and input ranges. The DM35425 also features four 12-bit high-speed analog outputs with programmable output ranges, and a 32-bit port of digital I/O.

1.2 Board features

- PCIe x 1 Interface
  - Universal Board can be used with a PCIe/104 Type 1 or Type 2 host
  - Dedicated DMA channel per I/O for maximum efficiency
- Analog inputs:
  - 16 Differential or 32 Single-ended analog input channels
  - 1.25 MSPS maximum input sampling rate
  - 12 bits resolution
  - Programmable single ended or differential inputs per channels
  - Threshold detection can generate an interrupt, or be used as a start or stop trigger
  - Configurable IIR filter on each channel
- Analog outputs:
  - 4 channels high-speed
  - 12 bit D/A converters
  - ±5, +5, ±10, & +10V output ranges
  - 7 µs full-scale settling time
- Advanced Digital I/O
  - 32-bit port of digital I/O
  - Bit programmable direction
  - Advanced digital interrupts
  - Parallel Bus Mode
- External Clocking
  - Provides 6 external clocking pins that can be used as inputs or outputs
  - Provides external triggering
  - External gate for each clock pin

1.3 Ordering Information

The DM35425 is available with the following options:

Table 1: Ordering Options

Part Number	Description
DM35425HR	PCIe/104 Analog I/O dataModule
IDAN-DM35425HR-62S	PCIe/104 Analog I/O dataModule in IDAN enclosure with 62-pin D-Sub Connector
IDAN-DM35425HR-68S	PCIe/104 Analog I/O dataModule in IDAN enclosure with 68-pin High-Density Connector

The Intelligent Data Acquisition Node (IDAN™) building block can be used in just about any combination with other IDAN building blocks to create a simple but rugged 104™ stack. This module can also be incorporated in a custom-built RTD HiDAN™ or HiDANplus High Reliability Intelligent Data Acquisition Node. Contact RTD sales for more information on our high reliability systems.

1.4 Contact Information

1.4.1 Sales Support

For sales inquiries, you can contact RTD Embedded Technologies sales via the following methods:

Phone: 1-814-234-8087 Monday through Friday, 8:00am to 5:00pm (EST).
E-Mail: sales@rtd.com

1.4.2 TECHNICAL SUPPORT

If you are having problems with you system, please try the steps in the Troubleshooting section of this manual on page 57.

For help with this product, or any other product made by RTD, you can contact RTD Embedded Technologies technical support via the following methods:

Phone: 1-814-234-8087 Monday through Friday, 8:00am to 5:00pm (EST).
E-Mail: techsupport@rtd.com

2 Specifications

2.1 Operating Conditions

Table 2: Operating Conditions

Symbol	Parameter	Test Condition	Min	Max	Unit
V_{cc5}	5V Supply Voltage		4.75	5.25	V
V_{cc3}	3.3V Supply Voltage		n/a	n/a	V
V_{cc12}	12V Supply Voltage		n/a	n/a	V
V_{cc-12}	-12V Supply Voltage		n/a	n/a	V
T_a	Operating Temperature		-40	+70	C
T_s	Storage Temperature		-55	+125	C
RH	Relative Humidity	Non-Condensing	0	90%	%
MTBF	Mean Time Before Failure	Telcordia Issue 2 30°C, Ground benign, controlled		TBD	Hours

2.2 Electrical Characteristics

Table 3: Electrical Characteristics

Symbol	Parameter	Test Condition	Min	Typical	Max	Unit
P	Power Consumption	V_{cc5} = 5.0V		4.35		W
I_{cc5}	5V Input Supply Current	Active		0.87		A
		PCIe/104 Bus				
	Differential Output Voltage		0.8		1.2	V
	DC Differential TX Impedance		80		120	Ω
	Differential Input Voltage		0.175		1.2	V
	DC Differential RX Impedance		80		120	Ω
	Electrical Idle Detect Threshold		65		175	mV
		Analog to Digital Converter				
	Linear Input Voltage	IN+ or IN-	-10		+10	V
FSR	Full-Scale Differential Input Voltage	V_{IN}=(IN+ -IN-) G = PGA Gain	$\frac{-5}{G}$		$\frac{4.99878}{G}$	V
	Resolution				12	Bits
	Data Rate				1.25	MSPS
	Input Impedance				>6	MΩ
	ENOB	Inputs= 0.8dBFS @ 10Khz Single-Ended, ±5V		11.55		Bits
	SNR	Inputs= 0.8dBFS @ 10Khz Single-Ended, ±5V		70.54		dB
	SINAD	Inputs= 0.8dBFS @ 10Khz Single-Ended, ±5V		70.44		dB
	THD	Inputs= 0.8dBFS @ 10Khz Single-Ended, ±5V		-86.78		dB
	SFDR	Inputs= 0.8dBFS @ 10Khz Single-Ended, ±5V		89.50		dB
	Third Order Intermodulation	F_{IN1}=0.8dBFS @ 10.0708Khz F_{IN2}=0.8dBFS @ 39.0625Khz		83.4		dB
	Noise Free Bits	Inputs Grounded Differential Inputs, ±5V		11		Bits
	Channel to Channel Cross Talk	No Channel Delay		-20.46		dB
		½ Sample Delay		-41.68		dB
		Sample Delay		-59.54		dB
		2 Sample Delay		-66.41		dB
G	Gains				0.5,1,2,4,8	
		Digital to Analog Converter				
	Full-Scale Analog Output Voltage	G = 1	-5		4.99756	V

Table 3: Electrical Characteristics

Symbol	Parameter	Test Condition	Min	Typical	Max	Unit
	Resolution				12	Bits
	Relative Accuracy			±1		LSB
	Gain Error			±2		LSB
	Settling Time			5		µs
	Output Current			5		mA
	Slew Rate			2		V/µs
G	Gains				1,2	
	Output Impedance			45		Ω
Digital I/O						
V_{IL}	Input High Voltage		2		5.5	V
V_{IH}	Input Low Voltage		-0.5		0.8	V
V_{OL}	Output Low Voltage	I_O = 12mA	0.0		0.4	V
V_{OH}	Output High Voltage	I_O = -12mA	2.6		3.3	V
	5V Output	CN3, CN4			200	mA

2.2.1 ANALOG INPUT FFT PLOTS

In Figure 1, a coherent 10 kHz sine wave signal was attached to input Channel 0 in the +/-5V, Single-ended mode. The FFT absolute value was calculated using 8192 data sample.

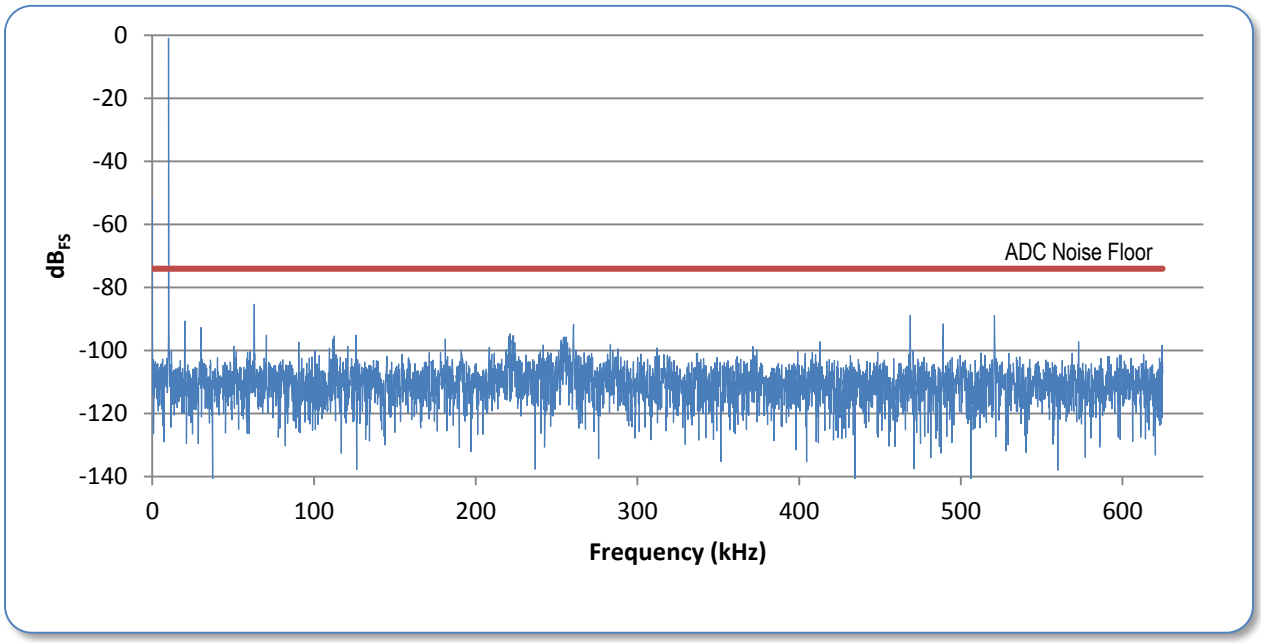


Figure 1: Channel FFT

2.2.2 ANALOG INPUT HISTOGRAMS

In Figure 2 you can see a histogram of samples from sampling a grounded input in +/-10 V, differential input range. The number of samples is 32768.

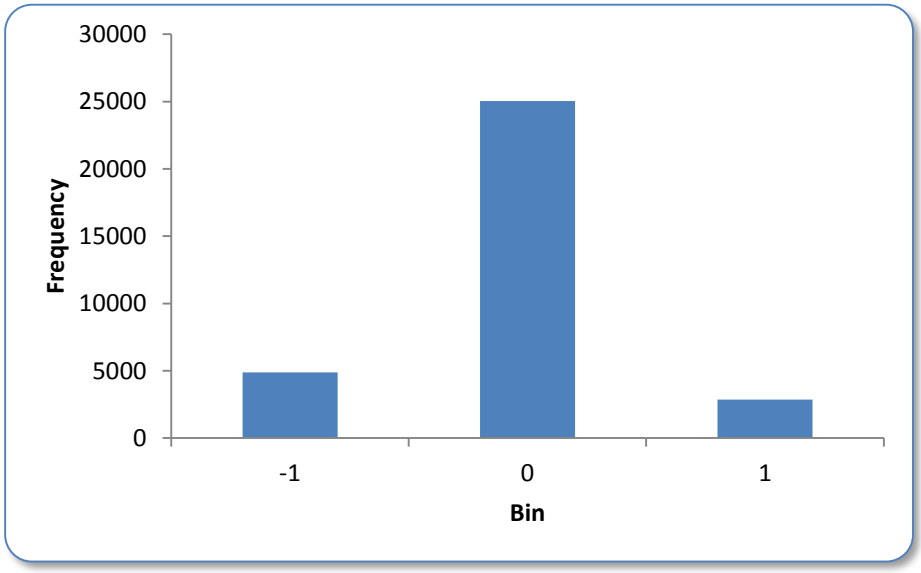


Figure 2: Histogram

3 Board Connection

3.1 Board Handling Precautions

To prevent damage due to Electrostatic Discharge (ESD), keep your board in its antistatic bag until you are ready to install it into your system. When removing it from the bag, hold the board at the edges, and do not touch the components or connectors. Handle the board in an antistatic environment, and use a grounded workbench for testing and handling of your hardware.

3.2 Physical Characteristics

- Weight: Approximately 55 g (0.12 lbs.)
- Dimensions: 90.17 mm L x 95.89 mm W (3.550 in L x 3.775 in W)

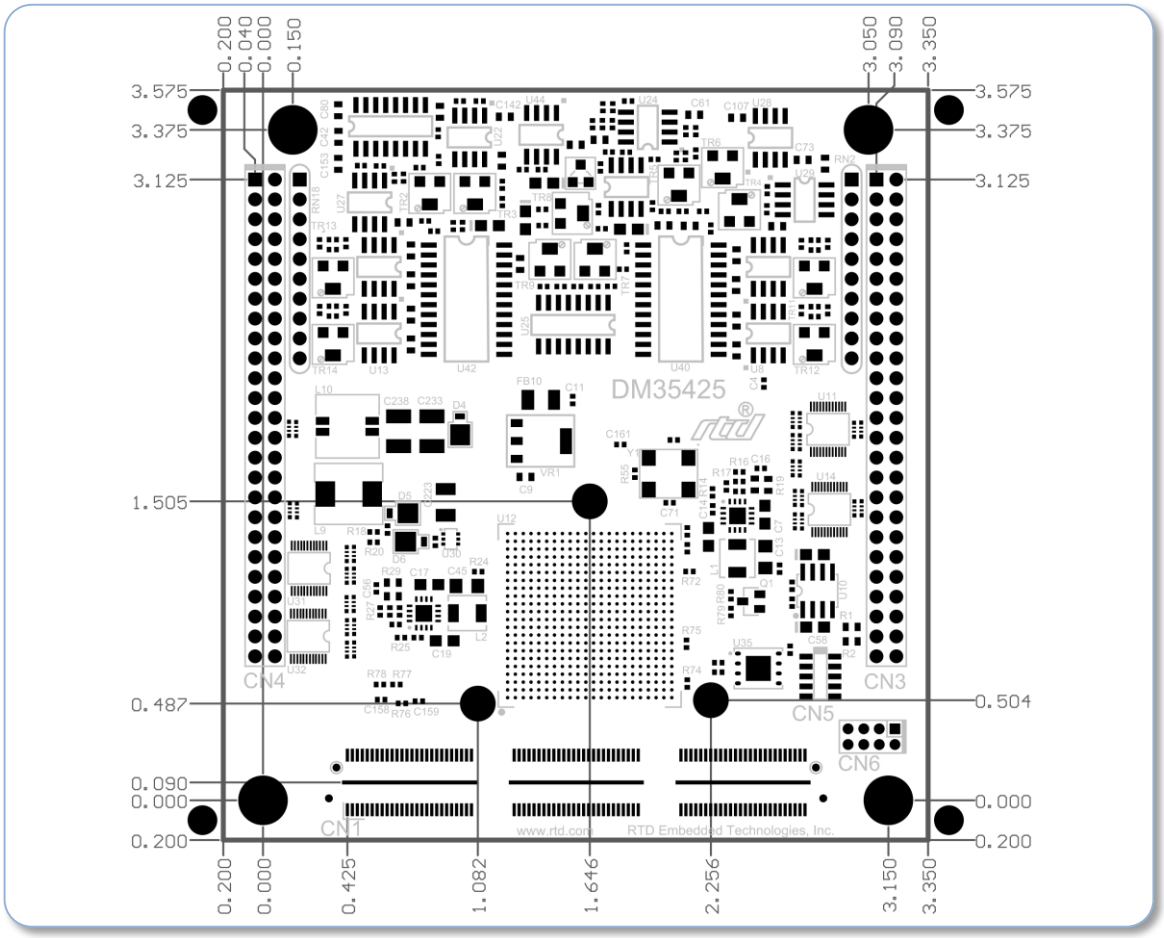


Figure 3: Board Dimensions

3.3 Connectors and Jumpers

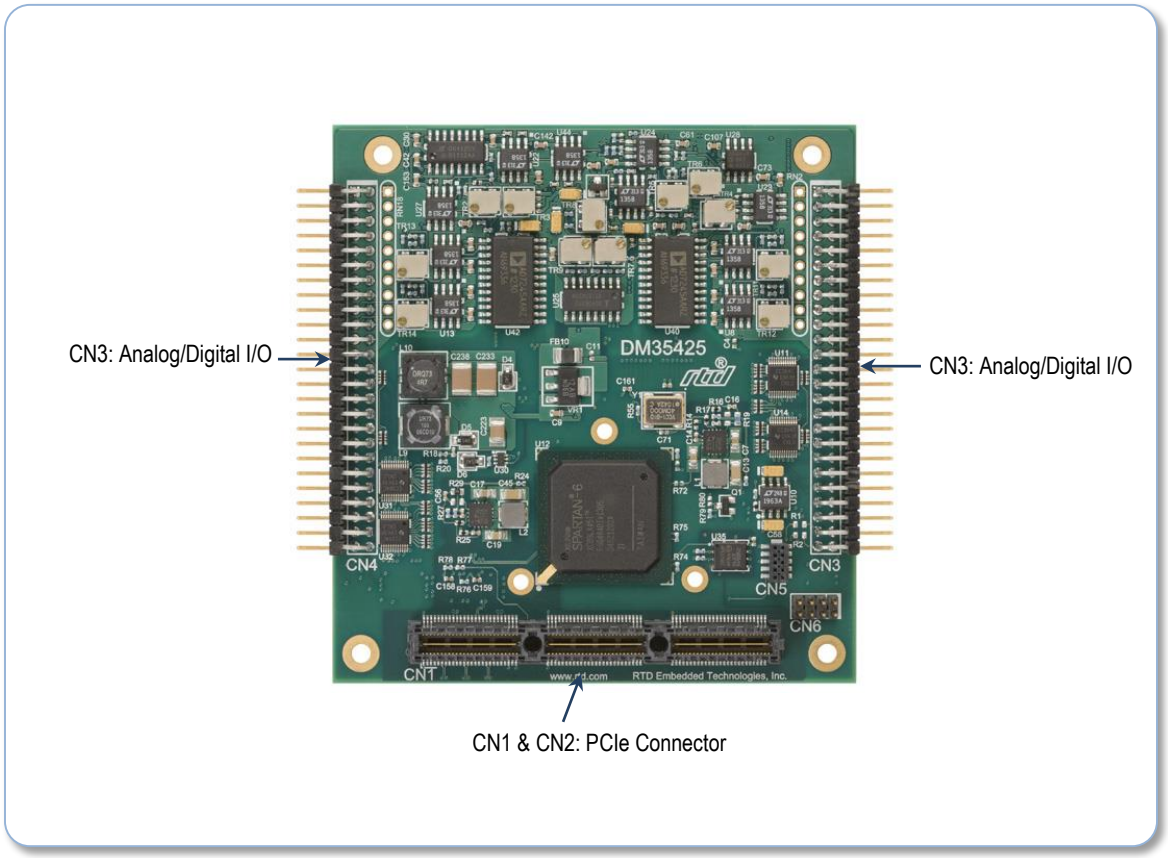


Figure 4: Board Connections

3.3.1 Bus Connectors

CN1(Top) & CN2(Bottom): PCIe Connector

The PCIe connector is the connection to the system CPU. The position and pin assignments are compliant with the PCI/104-Express Specification. (See PC/104 Specifications on page 58)

The DM35425 is a "Universal" board, and can connect to either a Type 1 or Type 2 PCIe/104 connector.

3.3.2 DM35425 EXTERNAL I/O CONNECTORS

CN3 & CN4: Analog/Digital I/O Connector

The Digital I/O Connector is a 2 x 25, 0.1" spacing right-angle connector. The pin assignments are shown in Tables below.

Table 4: CN3 Differential Mode Pin-out

AIN0+	1	2	AIN0-
AIN1+	3	4	AIN1-
AIN2+	5	6	AIN2-
AIN3+	7	8	AIN3-
AIN4+	9	10	AIN4-
AIN5+	11	12	AIN5-
AIN6+	13	14	AIN6-
AIN7+	15	16	AIN7-
AOUT0	17	18	AGND
AOUT1	19	20	AGND
AGND	21	22	AGND
DIO7	23	24	DIO15
DIO6	25	26	DIO14
DIO5	27	28	DIO13
DIO4	29	30	DIO12
DIO3	31	32	DIO11
DIO2	33	34	DIO10
DIO1	35	36	DIO9
DIO0	37	38	DIO8
EXT_CLK_2	39	40	GND
EXT_CLK_3	41	42	EXT_CLK_4
EXT_CLK_5	43	44	EXT_CLK_6
EXT_CLK_7	45	46	Reserved
Reserved	47	48	+5V
Reserved	49	50	GND

Table 5: CN3 Single-Ended Mode Pin-out

AIN0	1	2	AIN8
AIN1	3	4	AIN9
AIN2	5	6	AIN10
AIN3	7	8	AIN11
AIN4	9	10	AIN12
AIN5	11	12	AIN13
AIN6	13	14	AIN14
AIN7	15	16	AIN15
AOUT0	17	18	AGND
AOUT1	19	20	AGND
AGND	21	22	AGND
DIO7	23	24	DIO15
DIO6	25	26	DIO14
DIO5	27	28	DIO13
DIO4	29	30	DIO12
DIO3	31	32	DIO11
DIO2	33	34	DIO10
DIO1	35	36	DIO9
DIO0	37	38	DIO8
EXT_CLK_2	39	40	GND
EXT_CLK_3	41	42	EXT_CLK_4
EXT_CLK_5	43	44	EXT_CLK_6
EXT_CLK_7	45	46	Reserved
Reserved	47	48	+5V
Reserved	49	50	GND

Table 6: CN4 Differential Mode Pin-out

AIN8+	1	2	AIN8-
AIN9+	3	4	AIN9-
AIN10+	5	6	AIN10-
AIN11+	7	8	AIN11-
AIN12+	9	10	AIN12-
AIN13+	11	12	AIN13-
AIN14+	13	14	AIN14-
AIN15+	15	16	AIN15-
AOUT2	17	18	AGND
AOUT3	19	20	AGND
AGND	21	22	AGND
DIO23	23	24	DIO31
DIO22	25	26	DIO30
DIO21	27	28	DIO29
DIO20	29	30	DIO28
DIO19	31	32	DIO27
DIO18	33	34	DIO26
DIO17	35	36	DIO25
DIO16	37	38	DIO24
EXT_CLK_GATE2	39	40	GND
EXT_CLK_GATE3	41	42	EXT_CLK_GATE4
EXT_CLK_GATE5	43	44	EXT_CLK_GATE6
EXT_CLK_GATE7	45	46	Reserved
Reserved	47	48	+5V
Reserved	49	50	GND

Table 7: CN4 Single-Ended Mode Pin-out

AIN16	1	2	AIN24
AIN17	3	4	AIN25
AIN18	5	6	AIN26
AIN19	7	8	AIN27
AIN20	9	10	AIN28
AIN21	11	12	AIN29
AIN22	13	14	AIN30
AIN23	15	16	AIN31
AOUT2	17	18	AGND
AOUT3	19	20	AGND
AGND	21	22	AGND
DIO23	23	24	DIO31
DIO22	25	26	DIO30
DIO21	27	28	DIO29
DIO20	29	30	DIO28
DIO19	31	32	DIO27
DIO18	33	34	DIO26
DIO17	35	36	DIO25
DIO16	37	38	DIO24
EXT_CLK_GATE2	39	40	GND
EXT_CLK_GATE3	41	42	EXT_CLK_GATE4
EXT_CLK_GATE5	43	44	EXT_CLK_GATE6
EXT_CLK_GATE7	45	46	Reserved
Reserved	47	48	+5V
Reserved	49	50	GND

Other Connectors

CN5 and CN6 are for Factory Use only

3.3.3 JUMPERS

There are no jumpers on the board.

3.4 Steps for Installing

1. Always work at an ESD protected workstation, and wear a grounded wrist-strap.
2. Turn off power to the PC/104 system or stack.
3. Select and install stand-offs to properly position the module on the stack.
4. Remove the module from its anti-static bag.
5. Check that pins of the bus connector are properly positioned.
6. Check the stacking order; make sure all of the busses used by the peripheral cards are connected to the cpuModule.
7. Hold the module by its edges and orient it so the bus connector pins line up with the matching connector on the stack.
8. Gently and evenly press the module onto the PC/104 stack.
9. If any boards are to be stacked above this module, install them.
10. Attach any necessary cables to the PC/104 stack.
11. Re-connect the power cord and apply power to the stack.
12. Boot the system and verify that all of the hardware is working properly.

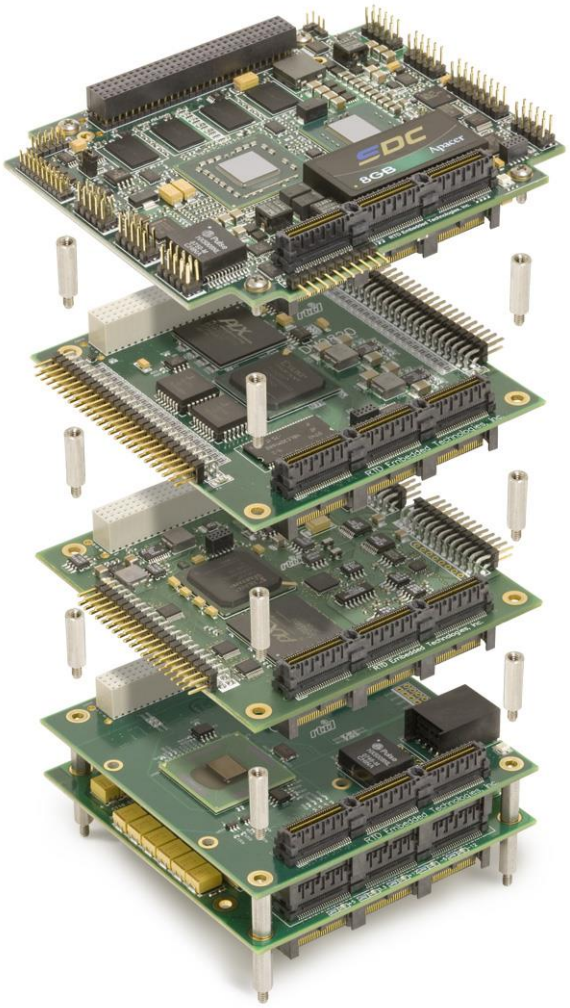


Figure 5: Example 104™ Stack

4 IDAN Connections

4.1 Module Handling Precautions

To prevent damage due to Electrostatic Discharge (ESD), keep your module in its antistatic bag until you are ready to install it into your system. When removing it from the bag, hold the module by the aluminum enclosure, and do not touch the components or connectors. Handle the module in an antistatic environment, and use a grounded workbench for testing and handling of your hardware.

4.2 Physical Characteristics

- Weight: Approximately 0.33 Kg (0.72 lbs.)
- Dimensions: 151.972 mm L x 129.978 mm W x 34.011 mm H (5.983 in L x 5.117 in W x 1.339 in H)

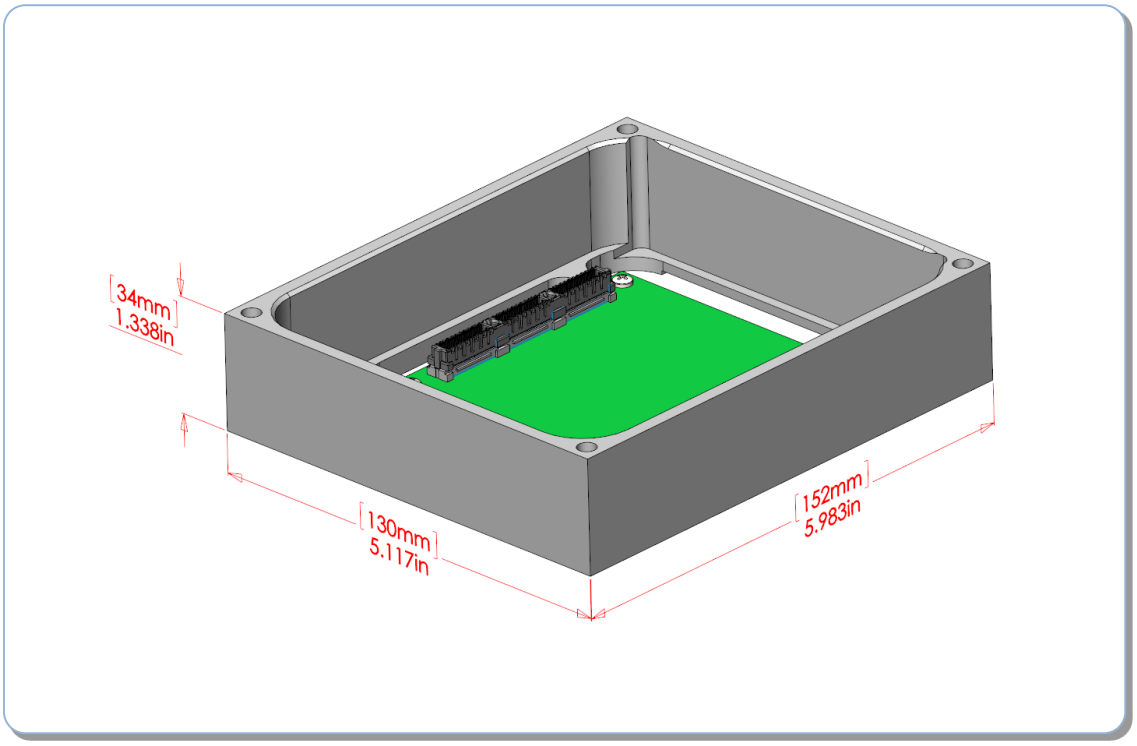


Figure 6: IDAN Dimensions

4.3 Connectors

4.3.1 Bus Connectors

CN1(Top) & CN2(Bottom): PCIe Connector

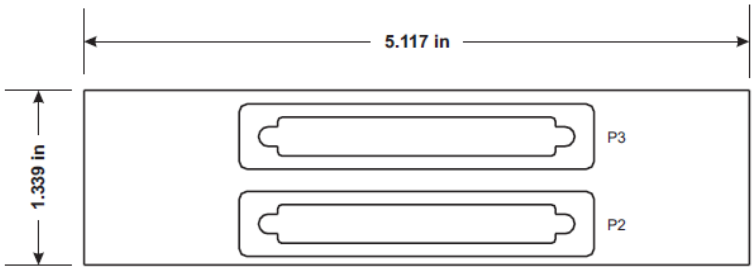
The PCIe connector is the connection to the system CPU. The position and pin assignments are compliant with the *PCI/104-Express Specification*. (See PC/104 Specifications on page 58)

The DM35425 is a "Universal" board, and can connect to either a Type 1 or Type 2 PCIe/104 connector.

4.4 Connectors

4.4.1 External I/O Connectors

P2 Connector - 68-pin Subminiature “D” Female Connector



Connector Part #: Amp 749070-7 **Sample Mating Connector:** Amp 786090-7(IDC Crimp)

Table 8: IDAN- DM35425 68-Pin Subminiature "D" Connector

IDAN Pin#	Signal	DM35425 Pin #	
1	AIN0+/AIN0	CN3	1
2	AIN0-/AIN8	CN3	2
3	AIN1+/AIN1	CN3	3
4	AIN1-/AIN9	CN3	4
5	AIN2+/AIN2	CN3	5
6	AIN2-/AIN10	CN3	6
7	AIN3+/AIN3	CN3	7
8	AIN3-/AIN11	CN3	8
9	AIN4+/AIN4	CN3	9
10	AIN4-/AIN12	CN3	10
11	AIN5+/AIN5	CN3	11
12	AIN5-/AIN13	CN3	12
13	AIN6+/AIN6	CN3	13
4	AIN6-/AIN14	CN3	14
15	AIN7+/AIN7	CN3	15
16	AIN7-/AIN15	CN3	16
17	AOUT0	CN3	17
18	AGND	CN3	18
19	AOUT1	CN3	19
20	AGND	CN3	20
21	AGND	CN3	21
22	AGND	CN3	22
23	DIO7	CN3	23
24	DIO15	CN3	24
25	DIO6	CN3	25
26	DIO14	CN3	26
27	DIO5	CN3	27
28	DIO13	CN3	28
29	DIO4	CN3	29
30	DIO12	CN3	30
31	DIO3	CN3	31
32	DIO11	CN3	32
33	DIO2	CN3	33
34	DIO10	CN3	34

Table 8: IDAN- DM35425 68-Pin Subminiature "D" Connector

IDAN Pin#	Signal	DM35425 Pin #	
35	DIO1	CN3	35
36	DIO9	CN3	36
37	DIO0	CN3	37
38	DIO8	CN3	38
39	EXT_CLK_2	CN3	39
40	GND	CN3	40
41	EXT_CLK_3	CN3	41
42	EXT_CLK_4	CN3	42
43	EXT_CLK_5	CN3	43
44	EXT_CLK_6	CN3	44
45	EXT_CLK_7	CN3	45
46	Reserved	CN3	46
47	Reserved	CN3	47
48	+5V	CN3	48
49	Reserved	CN3	49
50	GND	CN3	50
51	N/C		
52	N/C		
53	N/C		
54	N/C		
55	N/C		
56	N/C		
57	N/C		
58	N/C		
59	N/C		
60	N/C		
61	N/C		
62	N/C		
63	N/C		
64	N/C		
65	N/C		
66	N/C		
67	N/C		
68	N/C		

P3 Connector - 68-pin Subminiature “D” Female Connector

Connector Part #: Amp 749070-7 **Sample Mating Connector:** Amp 786090-7(IDC Crimp)

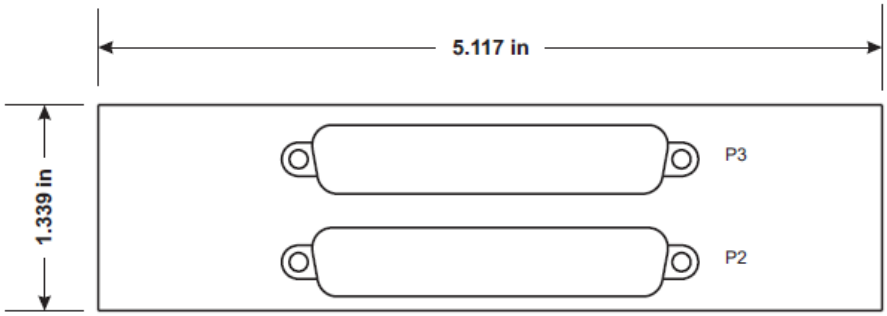
Table 9: IDAN- DM35425 68-Pin Subminiature "D" Connector

IDAN Pin#	Signal	DM35425 Pin #	
1	AIN8+/AIN16	CN4	1
2	AIN8-/AIN24	CN4	2
3	AIN9+/AIN17	CN4	3
4	AIN9-/AIN25	CN4	4
5	AIN10+/AIN18	CN4	5
6	AIN10-/AIN26	CN4	6
7	AIN11+/AIN19	CN4	7
8	AIN11-/AIN27	CN4	8
9	AIN12+/AIN20	CN4	9
10	AIN12-/AIN28	CN4	10
11	AIN13+/AIN21	CN4	11
12	AIN13-/AIN29	CN4	12
13	AIN14+/AIN22	CN4	13
4	AIN14-/AIN30	CN4	14
15	AIN15+/AIN23	CN4	15
16	AIN15-/AIN31	CN4	16
17	AOUT2	CN4	17
18	AGND	CN4	18
19	AOUT3	CN4	19
20	AGND	CN4	20
21	AGND	CN4	21
22	AGND	CN4	22
23	DIO23	CN4	23
24	DIO31	CN4	24
25	DIO22	CN4	25
26	DIO30	CN4	26
27	DIO21	CN4	27
28	DIO29	CN4	28
29	DIO20	CN4	29
30	DIO28	CN4	30
31	DIO19	CN4	31
32	DIO27	CN4	32
33	DIO18	CN4	33
34	DIO26	CN4	34

Table 8: IDAN- DM35425 68-Pin Subminiature "D" Connector

IDAN Pin#	Signal	DM35425 Pin #	
35	DIO17	CN4	35
36	DIO25	CN4	36
37	DIO16	CN4	37
38	DIO24	CN4	38
39	EXT_CLK_GATE2	CN4	39
40	GND	CN4	40
41	EXT_CLK_GATE3	CN4	41
42	EXT_CLK_GATE4	CN4	42
43	EXT_CLK_GATE5	CN4	43
44	EXT_CLK_GATE6	CN4	44
45	EXT_CLK_GATE7	CN4	45
46	Reserved	CN4	46
47	Reserved	CN4	47
48	+5V	CN4	48
49	Reserved	CN4	49
50	GND	CN4	50
51	N/C		
52	N/C		
53	N/C		
54	N/C		
55	N/C		
56	N/C		
57	N/C		
58	N/C		
59	N/C		
60	N/C		
61	N/C		
62	N/C		
63	N/C		
64	N/C		
65	N/C		
66	N/C		
67	N/C		
68	N/C		

P2 Connector - 62-pin High Density “D” Female Connector



Connector Part #: VALCONN HDB-62S **Sample Mating Connector:** VALCONN HDB-62P

Table 10: IDAN- DM35425 62-Pin High Density "D" Connector

IDAN Pin#	Signal	DM35425 Pin #	
1	AIN0+/AIN0	CN3	1
2	AIN1-/AIN9	CN3	4
3	AIN3+/AIN3	CN3	7
4	AIN4-/AIN12	CN3	10
5	AIN6+/AIN6	CN3	13
6	AIN7-/AIN15	CN3	16
7	AOUT1	CN3	19
8	AGND	CN3	22
9	DIO6	CN3	25
10	DIO13	CN3	28
11	DIO3	CN3	31
12	DIO10	CN3	34
13	DIO0	CN3	37
14	GND	CN3	40
15	EXT_CLK_5	CN3	43
16	Reserved	CN3	46
17	Reserved	CN3	49
18	Reserved		
19	Reserved		
20	Reserved		
21	Reserved		
22	AIN0-/AIN8	CN3	2
23	AIN2+/AIN2	CN3	5
24	AIN3-/AIN11	CN3	8
25	AIN5+/AIN5	CN3	11
26	AIN6-/AIN14	CN3	14
27	AOUT0	CN3	17
28	AGND	CN3	20
29	DIO7	CN3	23
30	DIO14	CN3	26
31	DIO4	CN3	29

Table 10: IDAN- DM35425 62-Pin High Density "D" Connector

IDAN Pin#	Signal	DM35425 Pin #	
32	DIO11	CN3	32
33	DIO1	CN3	35
34	DIO8	CN3	38
35	EXT_CLK_3	CN3	41
36	EXT_CLK_6	CN3	44
37	Reserved	CN3	47
38	GND	CN3	50
39	Reserved		
40	Reserved		
41	Reserved		
42	Reserved		
43	AIN1+/AIN1	CN3	3
44	AIN2-/AIN10	CN3	6
45	AIN4+/AIN4	CN3	9
46	AIN5-/AIN13	CN3	12
47	AIN7+/AIN7	CN3	15
48	AGND	CN3	18
49	AGND	CN3	21
50	DIO15	CN3	24
51	DIO5	CN3	27
52	DIO12	CN3	30
53	DIO2	CN3	33
54	DIO9	CN3	36
55	EXT_CLK_2	CN3	39
56	EXT_CLK_4	CN3	42
57	EXT_CLK_7	CN3	45
58	+5V	CN3	48
59	Reserved		
60	Reserved		
61	Reserved		
62	Reserved		

P3 Connector - 62-pin High Density “D” Female Connector

Connector Part #: VALCONN HDB-62S **Sample Mating Connector:** VALCONN HDB-62P

Table 11: IDAN- DM35425 62-Pin High Density "D" Connector

IDAN Pin#	Signal	DM35425 Pin #	
1	AIN8+/AIN16	CN4	1
2	AIN9-/AIN25	CN4	4
3	AIN11+/AIN19	CN4	7
4	AIN12-/AIN28	CN4	10
5	AIN14+/AIN22	CN4	13
6	AIN15-/AIN31	CN4	16
7	AOUT3	CN4	19
8	AGND	CN4	22
9	DIO22	CN4	25
10	DIO29	CN4	28
11	DIO19	CN4	31
12	DIO26	CN4	34
13	DIO16	CN4	37
14	GND	CN4	40
15	EXT_CLK_GATE5	CN4	43
16	Reserved	CN4	46
17	Reserved	CN4	49
18	Reserved		
19	Reserved		
20	Reserved		
21	Reserved		
22	AIN8-/AIN24	CN4	2
23	AIN10+/AIN18	CN4	5
24	AIN11-/AIN27	CN4	8
25	AIN13+/AIN21	CN4	11
26	AIN14-/AIN30	CN4	14
27	AOUT2	CN4	17
28	AGND	CN4	20
29	DIO23	CN4	23
30	DIO30	CN4	26
31	DIO20	CN4	29

Table 10: IDAN- DM35425 62-Pin High Density "D" Connector

IDAN Pin#	Signal	DM35425 Pin #	
32	DIO27	CN4	32
33	DIO17	CN4	35
34	DIO24	CN4	38
35	EXT_CLK_GATE3	CN4	41
36	EXT_CLK_GATE6	CN4	44
37	Reserved	CN4	47
38	GND	CN4	50
39	Reserved		
40	Reserved		
41	Reserved		
42	Reserved		
43	AIN9+/AIN17	CN4	3
44	AIN10-/AIN26	CN4	6
45	AIN12+/AIN20	CN4	9
46	AIN13-/AIN29	CN4	12
47	AIN15+/AIN23	CN4	15
48	AGND	CN4	18
49	AGND	CN4	21
50	DIO31	CN4	24
51	DIO21	CN4	27
52	DIO28	CN4	30
53	DIO18	CN4	33
54	DIO25	CN4	36
55	EXT_CLK_GATE2	CN4	39
56	EXT_CLK_GATE4	CN4	42
57	EXT_CLK_GATE7	CN4	45
58	+5V	CN4	48
59	Reserved		
60	Reserved		
61	Reserved		
62	Reserved		

4.5 Steps for Installing

1. Always work at an ESD protected workstation, and wear a grounded wrist-strap.
2. Turn off power to the IDAN system.
3. Remove the module from its anti-static bag.
4. Check that pins of the bus connector are properly positioned.
5. Check the stacking order; make sure all of the busses used by the peripheral cards are connected to the cpuModule.
6. Hold the module by its edges and orient it so the bus connector pins line up with the matching connector on the stack.
7. Gently and evenly press the module onto the IDAN system.
8. If any boards are to be stacked above this module, install them.
9. Finish assembling the IDAN stack by installing screws of an appropriate length.
10. Attach any necessary cables to the IDAN system.
11. Re-connect the power cord and apply power to the stack.
12. Boot the system and verify that all of the hardware is working properly.

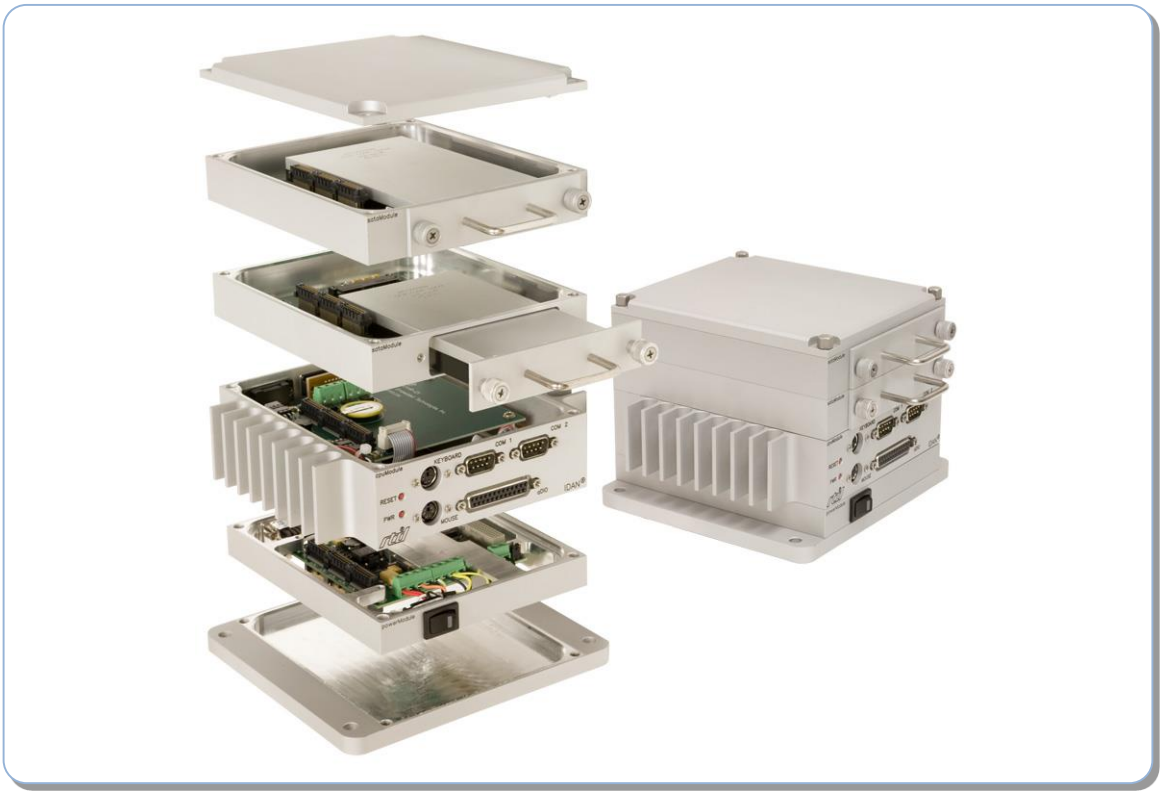


Figure 7: Example IDAN System

5 Functional Description

5.1 Block Diagram

The Figure below shows the functional block diagram of the DM35425. The various parts of the block diagram are discussed in the following sections.

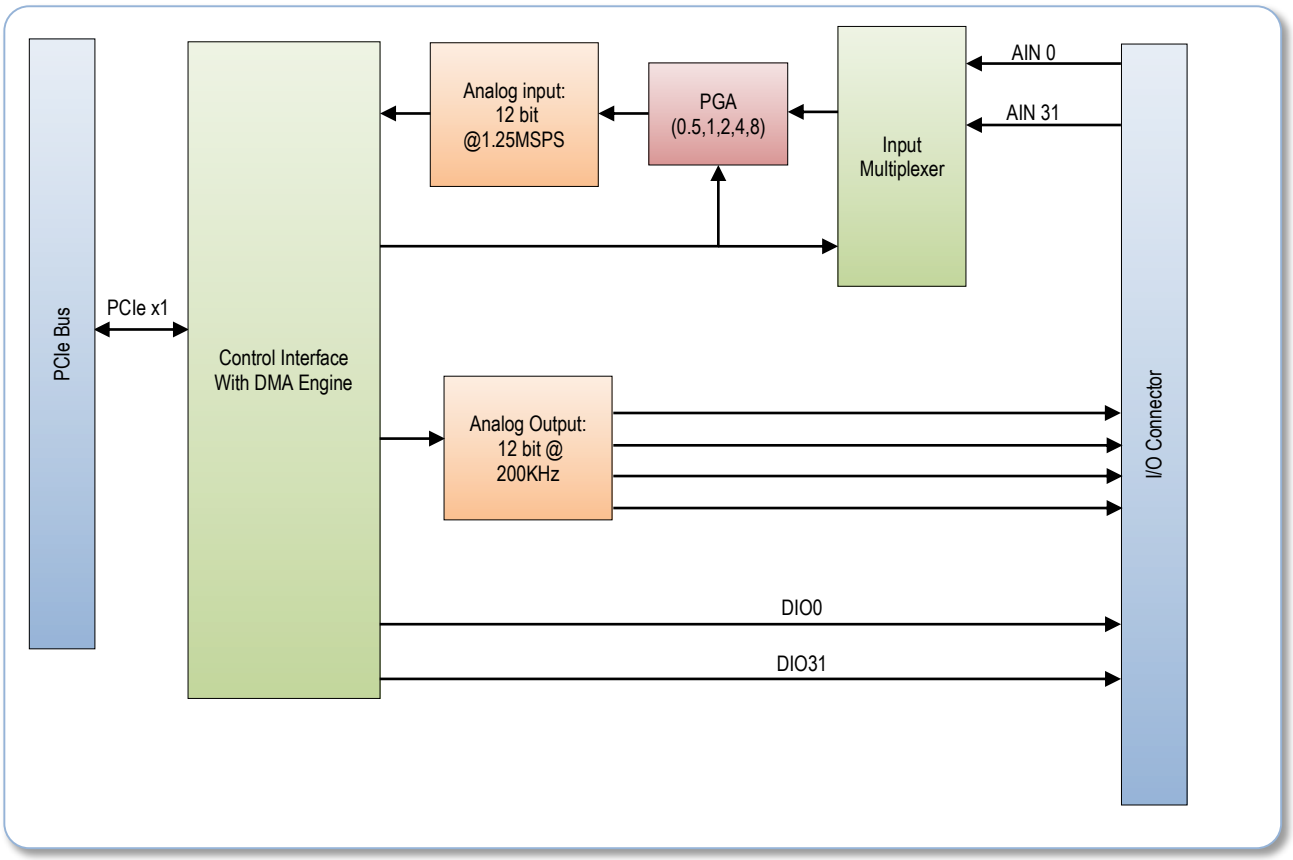


Figure 8: DM35425 Block Diagram

5.2 Control Interface with DMA Engine

The DM35425 features a FPGA with a built in PCI Express interface and DMA engine. The FPGA controls all communication between the bus and the control logic on the board. The FPGA also features small FIFOs for use with DMA, which is needed for continuous data transfer. Each DAC and ADC is provided with its own FIFO and DMA channel, allowing them to transfer data independent of one another.

Each DMA channel can be programmed to transfer data from FPGA to PCIe bus or from the PCIe bus to the FPGA. Each DMA channel also features a 64-bit PCI addressing and can access a maximum 16MB of memory for buffers.

5.3 Analog input

The DM35425 has 16 differential or 32 single-end channel inputs muxed to a single 12 bit SAR ADC converter, to provide high speed and high digital resolution of the analog input. The ADC converter has a max throughput of 1.25MHz (1 Channel). Refer to Equation 1 below to calculate max sampling rate. The high input impedance, low distortion, low noise operation design give you accurate results. The DM35425 also provides ±12V overvoltage input protection to the analog connector.

The DM35425 has a programmable input. This provides the user the ability to select single-ended/differential input, full-scale input range, unipolar/bipolar input, and channel sampling delay.

Each ADC channel supports a 511 sample FIFO for DMA. Each sample packed into a 32 bits word.

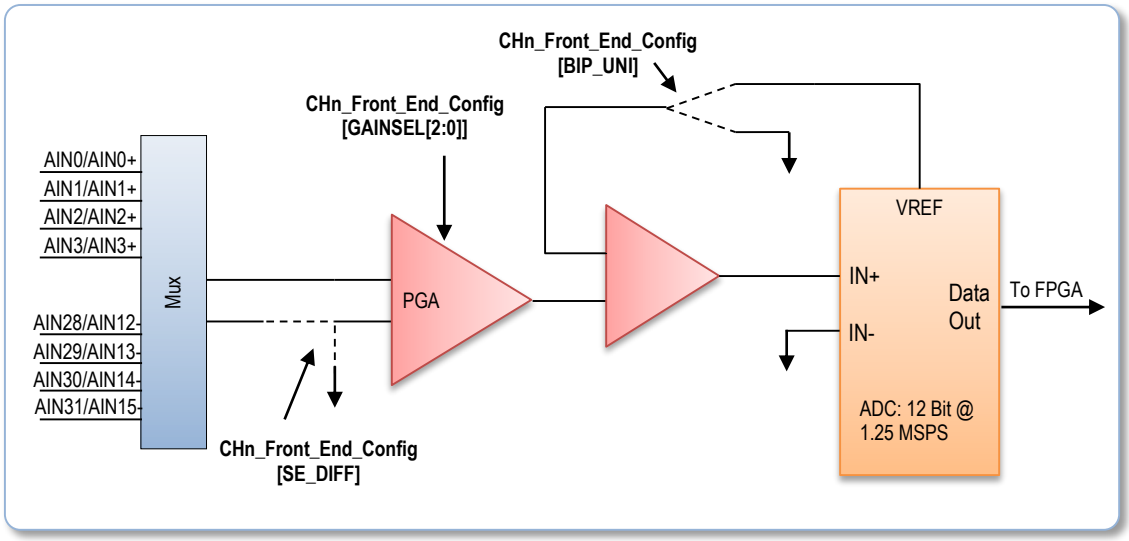
5.3.1 INITIALIZATION

There are several steps to initialize the Analog to Digital converter. The initialization prepares the converter and the front-end to capture samples. Following the example programs and using the drivers provided by RTD will ensure that these steps are followed in the correct order. Initialization of the ADC is performed as follows:

1. Set the ADC to the Uninitialized state (MODE = Uninitialized)
2. Setup the DMA for the channel
3. Set the input mode (CH_FRONT_END_CONFIG)
4. Set the start and stop triggers (START_TRIG, STOP_TRIG)
5. Set the clock source (CLK_SOURCE)
6. Set the sample rate (CLK_DIV_CNTR)
7. Set the Pre and/or Post Capture counters (PRE_TRIGGER_CAPTURE, POST_STOP_CAPTURE)
8. Set the ADC to the Reset state (MODE = Reset)
9. Start the DMA
10. Start the ADC (MODE = Go)

5.3.2 SIMPLIFIED BLOCK DIAGRAM OF ANALOG INPUT

The following figure shows the front end circuit for the DM35425. It also shows the names of the FPGA registers in bold and the different ways the front-end can be configured for different modes of operation. Refer to section 6.3.16 for more information about FPGA registers.



Single-Ended Input Mode

In single-ended mode, the input signal is measured in reference to the boards GND. In this mode the input signal is connected to input AIN0 through AIN31, and the low side to any of the GND pins available on the Analog Connector.

Differential Input Mode

In this mode your signal source may or may not have a separate ground reference. In differential mode, the high side input is measured in reference to the low side input. In this mode you connect the high side of the input signal to the analog input, AIN0+ through AIN16+, and connect the low side to corresponding ADC **-** pin. In most cases, the board ground must still be attached to the device that is generating the input signal.

When using the differential mode, you should install a 10 kΩ resistor pack at locations **RN2** and **RN18** on the DM35425HR to provide a reference to ground for signal sources without a separate ground reference.

Full-Scale Input Range

The DM35425 has a programmable gain input per channel. This gain can be programed for 1,2,4,8 to achieve input ranges ±5V, ±2.5V, ±1.25V, ±0.625V.

The DM35425 also features an attenuator in the front-end. This allows for additional input range of ±10V. This feature can only be used in bipolar mode.

Refer to Table 18 on page 39 for full-scale ranges for both unipolar and bipolar modes.

Bipolar/Unipolar Mode

In bipolar mode the data is collected in two's complement format. In this format the MSB of the data is the sign bit. If the MSB is "0" the output data is positive value. If the MSB is "1" the output code is negative. The 12-bits of data are sign-extended into a 32-bit word before transferring to the FIFO.

In unipolar mode there is no sign bit, since all values are positive. When the output code is all "0" the input voltage is 0 volts. When the output code is all "1" the input voltage is 10 volts when the gain is set to 1.

Voltage values for each bit will vary depending on input range. For example, if the input is set for ±5V, the formula for calculating voltage is as follows:

$$Voltage = \frac{Input\ Range}{2^{12}} \times Conversion\ Data$$

$$Voltage = \frac{10}{1\ \times 2^{12}} \times Conversion\ Data$$

$$Voltage = 2.44mV \times Conversion\ Data$$

Table 12: ADC Bipolar Code (±5 V Input Range)

Input Voltages	Sign	Output Code
+4.998 V	0	MSB 0111 1111 1111 LSB
+2.500 V	0	MSB 0100 0000 000 LSB
0.000 V	0	MSB 0000 0000 0000 LSB
-0.00244 V	1	MSB 1111 111 1111 LSB
-5.000 V	1	MSB 1000 0000 0000 LSB

Table 13: ADC Bipolar Code (±10 V Input Range)

Input Voltages	Sign	Output Code
+9.995 V	0	MSB 0111 1111 1111 LSB
+5.000 V	0	MSB 0100 0000 000 LSB
0.000 V	0	MSB 0000 0000 0000 LSB
-0.00488 V	1	MSB 1111 111 1111 LSB
-10.000 V	1	MSB 1000 0000 0000 LSB

Table 14: ADC Unipolar Code (0 to 10 V Input Range)

Input Voltages	Sign	Output Code
+9.99756 V	0	MSB 1111 1111 1111 LSB
+5.000 V	0	MSB 1000 0000 000 LSB
0.000 V	0	MSB 0000 0000 0000 LSB

Channel Delay

The DM35425 uses a 32 single-ended/16 differential channel multiplexed input, when ADC is sampling, which iterates through each enabled channels after the pervious channel has finished sampling. By default, there is no delay between channel sampling. The channel delay bits of the CHn_FRONT_END_CONFIG (Maskable Read/Write) register on page 39 provide a way to delay sampling between channels. This is used to reduce cross talk between channels. Refer to Electrical Characteristics on page 10 to see typical channel to channel cross talk.

Effective Sampling Rate

The DM35425 uses a multiplexed input to allow a max of 32 single-end/16-differential channels using one ADC. This module burst samples all enabled channels after each pacer clock pulse, starting at the first enabled channel and sequentially every enabled channel afterwards. Due to

the multiplexed input the sampling rate is limited to the number of channels enabled and summation of channel delay. Refer to the equation below to calculate the max sample rate per channel.

Equation 1: Max Sampling Rate

$$Max\ Sample\ Rate\ = \frac{1.25MHz}{Number\ of\ Channels\ Enabled + \sum Channel\ Delay}$$

5.4 Analog output

The DM35425 feature 4 independent 12-bit analog output channels with individually programmable output ranges of ±5V, and ±10V. Each channel supports a maximum update rate 200 kHz and a maximum operating load of 5mA.

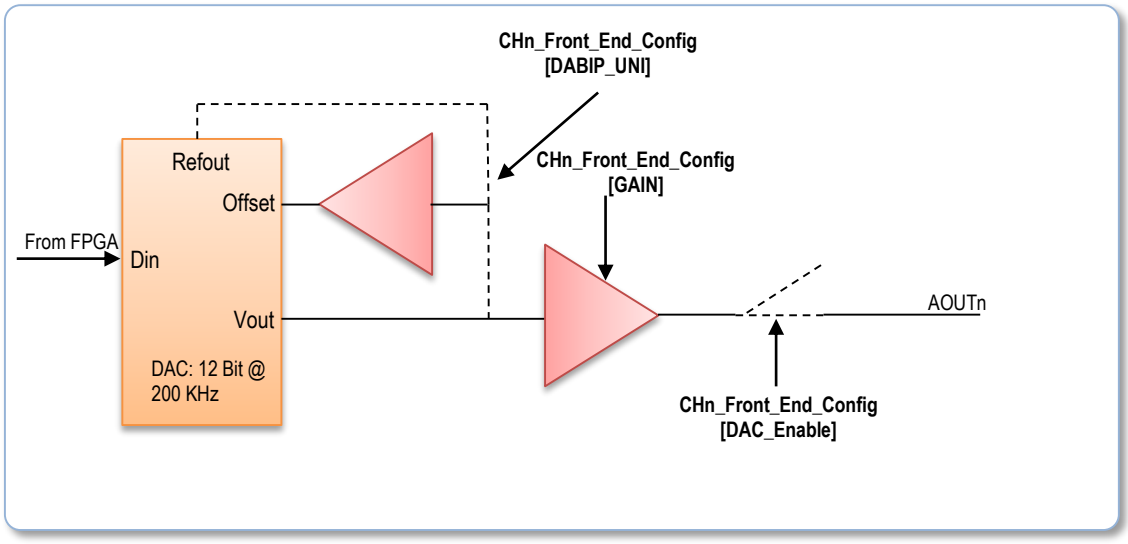
5.4.1 Initializing the DAC Converter

The following is a list of the typical steps needed to initialize the DAC converter and begin sampling

1. Set the DAC to the Uninitialized state (MODE = Uninitialized)
2. Setup the DMA for the channel
3. Set the input mode (CH_FRONT_END_CONFIG)
4. Set the start and stop triggers (START_TRIG, STOP_TRIG)
5. Set the clock source (CLK_SOURCE)
6. Set the sample rate (CLK_DIV_CNTR)
7. Set the Post Capture counter (POST_STOP_CAPTURE)
8. Set the DAC to the Reset state (MODE = Reset)
9. Start the DMA
10. Start the DAC (MODE = Go)

5.4.2 Simplified Block Diagram of Analog Output

The following figure shows the front end circuit for the DM35425. It also shows the names of the FPGA registers in bold and the different ways the front-end can be configured for different modes of operation. Refer to section 6.4.15 for more information about FPGA registers.



The following table list the key digital codes and corresponding output voltages for the DAC converters.

DAC Bit Weight	Ideal Output Voltages (mV)			
	-5 to +5 V	0 to +5 V	-10 to +10 V	0 to +10 V
1111 1111 1111	+4997.56	+4998.78	+9995.12	+9997.56
1000 0000 0000	0	+2500.00	0000.00	+5000.00
0100 0000 0000	-2500.00	+1250.00	-5000.00	+2500.00
0010 0000 0000	-3750.00	+625.00	-7500.00	+1250.00
0001 0000 0000	-4375.00	+312.50	-8750.00	+625.00
0000 1000 0000	-4687.50	+156.25	-9375.00	+312.50
0000 0100 0000	-4843.75	+78.13	-9687.50	+156.25
0000 0010 0000	-4921.88	+39.06	-9843.75	+78.13
0000 0001 0000	-4960.94	+19.53	-9921.88	+39.06
0000 0000 1000	-4980.47	+9.77	-9960.94	+19.53
0000 0000 0100	-4990.24	+4.88	-9980.47	+9.77
0000 0000 0010	-4995.12	+2.44	-9990.23	+4.88
0000 0000 0001	-4997.56	+1.22	-9995.12	+2.44
0000 0000 0000	-5000.00	0.00	-10000.00	0.00

Voltage values for each bit will vary depending mode and gain. The formula for calculating count value as follows:

Bipolar Range

$$Count = \left(\frac{\frac{Voltage}{Gain} + 5}{10}\right) \times 2^{12}$$

Unipolar Range

$$Count = \left(\frac{\frac{Voltage}{Gain}}{10}\right) \times 2^{12}$$

Refer to Table 20 on page 44 for full-scale ranges for both unipolar and bipolar modes.

Each DAC converter has a 511 sample FIFO for DMA. Each sample is packed into 32 bits, right justified and sign-extended.

5.5 Advanced Digital I/O

The DM35425 features 32 digital I/O line with DMA, parallel bus mode, and advance interrupts.

DMA

The DM35425 has three DMA channels for Digital I/O: input, output, and direction. Each channel has a 511 sample FIFO for DMA. Each sample is packed into 32 bits.

Advanced Interrupts

The DM35425 has an advanced interrupt block that can generate an interrupt on a match or event. The interrupts are across all 32 digital I/O. The bits can be individually selected.

When an interrupt is generated, the data on all of the ports is latched into the Capture registers. Bits are tested regardless of if a pin is an input or output.

A Match interrupt is generated when all un-masked bits in the Compare register match the input value of the port. An Event interrupt is generated when any un-masked input port bit changes.

Parallel Bus Mode

The DM35425 also features parallel bus mode for the digital I/O lines. In this mode, the 3 MSB lines of the Digital I/O are switched to control signals: Ready, Valid, and Clock. The remaining lines are used as a data bus.

When the clock pin is set to output, a high clock pulse (50ns) will be sent every Digital I/O pacer clock. Also, at the rising edge of each clock pulse, data in the OUT FIFO will be outputted on all digital I/O set to output.

The Valid pin is set high when the valid pin is set to an output and the Out FIFO is not empty.

When the clock pin is set to input, when the clock pin receives a rising edge and the valid pin is high all 32 bits on data will be written to the IN FIFO.

The Ready pin is set high when the Ready pin is set to output and the IN FIFO is not full.

5.6 External Clocking

The DM35425 features an external clocking function block. This feature allows the user to input a clock to drive a CLK_SRC_GLBn signal or output a CLK_SRC_GLBn signal. The CLK_SRC_GLBn are used to drive the CLK_GLBn signal which are part of the FPGA function block Clock Source. Clock sources are used as either sample clocks for function blocks or triggers for starting and stopping them.

There are 6 available CLK_GBLn, each is associated with a pin on CN3. Each pin can be configured to be either an input or an output. As an input to a CLK_SRC_GLBn, the max input clock frequency is ½ system clock frequency. This value can be found in GBC_SYS_CLK_FREQ (Read Only) on page 32. As an output CLK_SRC_GLBn will generated on the associated pin. By default this signal will be a pulse that is high for 25ns when the CLK_SRC_GLBn signal goes high. The width of this pulse can be increased using EXT_CLK_PWn (Read/Write) on page 52.

The following example show to capture ADC samples using an external and the external clocking function block:

- External Clocking Function Block Setup
    1. Set the CLK_SRC_GLB2 as input (EXT_CLK_DIR)
    2. Set the edge detect of CLK_SRC_GLB2 (EXT_CLK_EDGE)
    3. Set CLK_SRC_GLB2 clocking method (EXT_CLK2_CFG)
    4. Provide clock on CN3 pin 39.
- ADC Function Block Setup
    1. Set the ADC to the Uninitialized state (MODE = Uninitialized)
    2. Setup the DMA for the channel
    3. Set the input mode (CH_FRONT_END_CONFIG)
    4. Set the start and stop triggers (START_TRIG, STOP_TRIG)
    5. Set the clock source (CLK_SOURCE = CLK_GBL2)
    6. Set the sample rate (CLK_DIV_CNTR)
    7. Set the Pre and/or Post Capture counters (PRE_TRIGGER_CAPTURE, POST_STOP_CAPTURE)
    8. Set the ADC to the Reset state (MODE = Reset)
    9. Start the DMA
    10. Start the ADC (MODE = Go)

6 Register Address Space

The DM35425 FPGA code was built as a modular design, which allows each board function to have its own Functional Block (FB). Each functional block was designed to work independent of each other. For this reason, we provide individual DMA channels, interrupts, clocks, and FIFOs to each functional block.

The registers are described by their PCIe Base Address Register (BAR), which is defined in the PCI configuration space for this board. The configuration space is generally handled by the operating system. For more information on how to use the configuration space, consult the *PCI Local Bus Specification, Revision 3.0* from the PCI-SIG.

Register Types

There are several different types of registers that are referred to in this section. A description of each type is below.

- Read/Write Registers: The value that is written to this register can also be read back.
- Maskable Registers: This is a 32 bit register that consists of 16-bit data field in the upper word and a 16-bit mask value in the lower word. For each bit in the data field, it is only written to the register if the corresponding bit in the mask field is '1'.
- Sticky Registers: This is a status read register. When bit in this register has a value of '1', a '1' needs written to that bit to reset the register to '0'. This is typically used for interrupt status registers.
- Read Only: This register can only be read.

NOTE: Writing to Read-Only registers may have unexpected results.

Clock Source

Clock sources can serve as either sample clocks for function blocks, or triggers for starting and stopping them. Function blocks can drive a CLK_GBLn with a CLK_SRC_GBLn (see the register descriptions for details on the possible values for CLK_SRC_GBLn), and other function blocks then trigger from that clock. This is what lets multiple function blocks start at the same time, or stop on the same trigger.

For example, to have all function blocks start the same time as ADC, you would set ADC to drive CLK_GBL2 with its start trigger. You would then set all other function blocks to use CLK_GBL2 as their start trigger, and then start them. They will wait for the start trigger on CLK_GBL2 before they actually start. Start ADC, and all of the function blocks will start with it.

Below is the list of clock sources and the register value needed to select the source:

0x00: System clock/immediate
0x01: Never
0x02: CLK_GBL2
0x03: CLK_GBL3
0x04: CLK_GBL4
0x05: CLK_GBL5
0x06: CLK_GBL6
0x07: CLK_GBL7
0x08: Channel Threshold – One of the channels has exceeded the High or Low threshold.
0x09: Channel Threshold Inverted– All of the channels are within the High and Low threshold.
0x0A: CLK_GBL2 Inverted
0x0B: CLK_GBL3 Inverted
0x0C: CLK_GBL4 Inverted
0x0D: CLK_GBL5 Inverted
0x0E: CLK_GBL6 Inverted
0x0F: CLK_GBL7 Inverted

6.1 BAR0 – General Board Control

The BAR0 region is a Memory Mapped register space which contains some global registers. It also contains a table describing the different Function Blocks of the board, and the offsets into BAR2 of the registers for that Function Block. For maximum flexibility, the driver must read the table in BAR0 to calculate the offset to each Function Block in BAR2.

Table 15: BAR0 Registers

Offset	0x03	0x02	0x01	0x00
0x00	GBC_BRD_RST	GBC_EOI	GBC_REV	GBC_FMT
0x04	GBC_PDP			
0x08	GBC_BUILD			
0x0C	*Reserved*		GBC_SYS_CLK_FREQ	
0x10	GBC_IRQ_STATUS			
0x14				
0x18	GBC_DIRQ_STATUS			
0x1C				
0x20	FB0_ID			
0x24	FB0_OFFSET			
0x28	FB0_OFFSET_DMA			
0x2C	*Reserved*			
0x30	FB1_ID			
0x34	FB1_OFFSET			
0x38	FB1_OFFSET_DMA			
0x3C	*Reserved*			
…				
0x20+0x10*n	FBn_ID			
0x24+0x10*n	FBn_OFFSET			
0x28+0x10*n	FBn_OFFSET_DMA			
0x2C+0x10*n	*Reserved*			
…				
0xA0	FB8_ID			
0xA4	FB8_OFFSET			
0xA8	FB8_OFFSET_DMA			
0xAC	*Reserved*			

6.1.1 GBC_BRD_RST (READ/WRITE)

This register is used to send a reset command to the board. Write 0xAA to this register to reset the board.

6.1.2 GBC_EOI (READ/CLEAR)

This register is used to acknowledge an interrupt. It is used to safeguard against missing an interrupt. At the end of the Interrupt Service Routines (ISR), write a 0x01 to this register. If there is another interrupt pending in the status registers, the interrupt line is toggled (Legacy Mode), or another interrupt is sent (MSI Mode).

6.1.3 GBC_REV (READ-ONLY)

This register contains the FPGA revision for this board. A=1, B=2, etc.

6.1.4 GBC_FMT (READ-ONLY)

This register contains the format ID that is used in this board. The current value is 0x01.

6.1.5 GBC_PDP (READ-ONLY)

This register contains the PDP number for this board.

6.1.6 GBC_BUILD (READ-ONLY)

This register contains a unique 32-bit build number for the FPGA code.

6.1.7 GBC_SYS_CLK_FREQ (READ ONLY)

This register contains the measured frequency of the system clock. Units are 10 kHz, i.e. (Frequency in Hertz) = (GBC_SYS_CLK_FREQ * 10 kHz). This value is not available (will read 0) until 100us after a Board Reset, and is continually updated.

6.1.8 GBC_IRQ_STATUS (READ/CLEAR)

This is a 64-bit interrupt status register for non-DMA interrupts. Each bit in this register corresponds to one of the Function Blocks; bit 0 corresponds to FB0 (whose ID and OFFSET are at 0x020), etc. Bits 60 through 63 are reserved. This is a Sticky Register, so the driver clears it by writing a '1' to the appropriate bit.

6.1.9 GBC_DIRQ_STATUS (READ/CLEAR)

This is a 64-bit interrupt status register for DMA interrupts. Each bit in this register corresponds to one of the Function Blocks; bit 0 corresponds to FB0 (whose ID and OFFSET are at 0x020), etc. Bits 60 through 63 are reserved. This is a "sticky" register, and the driver clears it by writing a '1' to the appropriate bit.

6.1.10 FBN_ID (READ-ONLY)

This is a 32-bit value that identifies the type of Function Block in slot 'n'.

0x01031000 – ADC
0x01032000 – DAC
0x01003001 – Digital I/O
0x00010001 – External Clocking

6.1.11 FBN_OFFSET (READ-ONLY)

This is the offset from the beginning of BAR2 that this Functional Block resides in.

6.1.12 FBN_OFFSET_DMA (READ-ONLY)

This is the offset from the beginning of BAR2 that the Functional Block DMA Registers reside in.

6.2 BAR2 – Functional Block Standard DMA

This section describes a standard DMA implementation is used by the Functional Blocks. There is a single DMA engine that services all of the DMA channels used by the Function Block. Each DMA channel has a block of registers associated with it to configure the DMA channel, as well as set up the descriptors for the buffers in system memory. In the sections below, "m" is used to enumerate the DMA channels, and "n" is used to enumerate the buffer descriptors within a channel.

Table 16: DMA Registers

Offset	0x03	0x02	0x01	0x00
D + 0x00	FB_DMAm_Stat_Underflow	FB_DMAm_Stat_ Overflow	FB_DMAm_Setup	FB_DMAm_Action
D + 0x04	FB_DMAm_Current_Buffer	FB_DMAm_Count		
D + 0x08	FB_DMAm_RD_FIFO_CNT		FB_DMAm_WR_FIFO_CNT	
D + 0x0C	FB_DMAm_Last_Action	FB_DMAm_Stat_ Complete	FB_DMAm_Stat_ Invalid	FB_DMAm_Stat_ Used
D + 0x10	FB_DMAm_CTRL0	FB_DMAm_STAT0	Reserved	
D + 0x14	Reserved	FB_DMAm_SIZE0		
D + 0x18	FB_DMAm_ADDRESS0			
D + 0x1C				
D + 0x20	FB_DMAm_CTRL1	FB_DMAm_STAT1	Reserved	
D + 0x24	Reserved	FB_DMAm_SIZE1		
D + 0x28	FB_DMAm_ADDRESS1			
D + 0x2C				
…				
D + 0x10 + (0x10 * n)	FB_DMAm_CTRLn	FB_DMAm_STATn	Reserved	
D + 0x14 + (0x10 * n)	Reserved	FB_DMAm_SIZEn		
D + 0x18 + (0x10 * n)	FB_DMAm_ADDRESSn			
D + 0x1C + (0x10 * n)				

6.2.1 FB_DMAm_ACTION (READ/WRITE)

This register is the overall control for this DMA channel. After writing to the Action register, the driver should poll the Last_Action register (below) until it reads the same value. This shows that the action has been performed by the DMA state machine. This is especially important when entering and exiting the Clear state.

0x00 = Clear: Clear the Current Buffer field, the internal offset counters, and the FIFO. DMA is stopped.

0x01 = Go: Starts DMA

0x02 = Pause: DMA transfers are stopped, but all internal registers maintain their state. During PAUSE you will still receive Stat_Underflow and Stat_Overflow interrupts. After PAUSE, you may transition to GO or CLEAR.

0x03 = Halt: Buffer has been filled that has the HALT bit set, attempted to use a buffer with the Valid bit cleared. After HALT, you must transition to CLEAR.

NOTE: The DMA engine also writes to this register when a buffer is completed with the HALT bit set, or it encounters an invalid buffer. When changing this register from the Go to the Clear state, be sure to read it back to make sure the DMA engine did not change it to the Halt state.

6.2.2 FB_DMAm_LAST_ACTION (READ/WRITE)

The DMA Engine writes the value of FB_DMAm_Action to this register after it has completed the action. This indicates to the user that the last command has been processed. It specifically aids the transition to the Clear state. When transitioning to Clear, the user should wait until FB_DMAm_Last_Action indicates that the Clear has been processed before initiating any other Action changes.

The user may also write a value to this register and then poll the register to see when the value changes. This method can be used to detect when the DMA engine services the channel without an Action change.

6.2.3 FB_DMAM_SETUP (READ/WRITE)

B0: IntEna: Set to '1' to enable the DMA engine to generate interrupts on completion of a buffer.

B1: ErrIntEna: Set to '1' to enable the DMA engine to generate interrupts on error.

B2: Direction: Set to '1' to transfer from the board to the PCI bus. Clear to '0' to transfer from the PCI bus to the board. Note that although the DMA channel always supports both directions, the Function Block that the channel is associated with may only support one direction.

B3: IgnoreUsed: Set to '1' to prevent an error condition when accessing a buffer with the Used bit set. Examples are continuous output from a DAC, or very large Pre-trigger buffering using system memory.

6.2.4 FB_DMAM_STAT_USED (READ/WRITE)

This register is used to determine the source of a DMA interrupt. The bits are cleared by writing 0x00 to the byte. Stat_Used will be set regardless of having ErrIntEna set to '1'.

B0: Used_Desc. Set to '1' by the DMA engine if it attempting to use a descriptor with the Used bit set.

6.2.5 FB_DMAM_STAT_INVALID (READ/WRITE)

This register is used to determine the source of a DMA interrupt. The bits are cleared by writing 0x00 to the byte. Stat_Invalid will be set regardless of having ErrIntEna set to '1'.

B0: Invalid_Desc. Set to '1' by the DMA engine if it attempting to use a descriptor with the Valid bit cleared.

6.2.6 FB_DMAM_STAT_OVERFLOW (READ/WRITE)

This register is used to determine the source of a DMA interrupt. The bits are cleared by writing 0x00 to the byte. Stat_Overflow will be set regardless of having ErrIntEna set to '1'. If an overflow occurs the DMA engine will PAUSE.

B0: Overflow (R/C). Set to '1' by the DMA engine if an overflow occurred on the FIFO.

6.2.7 FB_DMAM_STAT_UNDERFLOW (READ/WRITE)

This register is used to determine the source of a DMA interrupt. The bits are cleared by writing 0x00 to the byte. Stat_Underflow will be set regardless of having ErrIntEna set to '1'. If an underflow occurs the DMA engine will PAUSE.

B0: Underflow (R/C). Set to '1' by the DMA engine if an underflow occurred on the FIFO.

6.2.8 FB_DMAM_STAT_COMPLETE (READ/WRITE)

This register is used to determine the source of a DMA interrupt. The bits are cleared by writing 0x00 to the byte.

B0: Buffer_Complete (R/C). Set to '1' by the DMA engine when a buffer is filled that has the Interrupt bit set.

6.2.9 FB_DMAM_CURRENT_BUFFER (READ-ONLY)

This is the ID for the buffer that will be used for the next access. The driver may use this to track the progress of the DMA activity. This value is displayed in Bytes.

6.2.10 FB_DMAM_COUNT (READ-ONLY)

This is the offset in the DMA buffer for the next access. The driver may use this to track the progress of the DMA activity. This value is displayed in Bytes.

6.2.11 FB_DMAM_RD_FIFO_CNT (READ-ONLY)

B[9:0] This is the amount of data available in the read FIFO in bytes. Software can use this to determine when the FIFO is empty. A value of 0x3FC indicates that there are 1020 or more bytes of data available.

B15: RD_EMPTY- '1' indicates that the read FIFO is empty

6.2.12 FB_DMAm_WR_FIFO_CNT (Read-Only)

B[9:0] This is the amount of space available in the write FIFO in bytes. Software can use this to determine when the FIFO is full. A value of 0x3FC indicated that there are 1020 or more bytes of space available.

B15: WR_FULL- '1' indicates that the write FIFO is full

6.2.13 FB_DMAm_ADDRESSn (Read/Write)

This is the 64-bit PCI address for DMA Channel m, buffer n. It must be double-word aligned (i.e. b[1:0] are reserved).

6.2.14 FB_DMAm_SIZEn (Read/Write)

This is the size in bytes of the buffer for DMA Channel m, buffer n. It must be an integer number of double-words (i.e. b[1:0] are reserved). The actual size is FB_DMAm_SIZEn + 4 Bytes. The maximum buffer size is 16MB.

6.2.15 FB_DMAm_CTRLn (Read/Write)

B0: Valid: Driver sets to '1' to indicate that this contains valid information. The DMA engine will set the error bit and halt if it is ready to use this descriptor and it is not valid.

B1: Halt: Driver sets to '1' to halt the DMA engine after this buffer is full.

B2: Loop: Driver sets to '1' to start back at descriptor 0 after this buffer is full. This has a higher priority than the HALT bit.

B3: Interrupt: Driver sets to '1' to generate an interrupt after this buffer is full.

If the last buffer is reached, and the HALT and LOOP bits are both '0', the DMA engine will loop.

If the last buffer is reached, and the HALT and LOOP bits are both '1', the DMA engine will halt and the Current_Buffer will be set to 0.

6.2.16 FB_DMAm_STATn (Read/Clear)

B0: Used (R/C): DMA engine sets to '1' to indicate that it has completely used this descriptor. The driver must clear this bit when it is ready to be used again. The DMA engine will set the error bit and PAUSE if it is ready to use this descriptor and the Used bit is set, unless the IgnoreUsed bit is set. The bits are cleared by writing 0x00 to the byte.

6.3 BAR2 – ADC Functional Block

This Function Block is for an Analog to Digital converter. This ADC Function block has multiple channels. There are 32 channels in this functional block, however all channels must use the same pacer clock. Each channel has its own FIFO and DMA channel. In the sections below "n" is used to enumerate the channels of the ADC function block.

Table 17: Multi-Channel ADC Functional Block

	Offset	0x03	0x02	0x01	0x00
Header	FB + 0x00	FB_ID			
	FB + 0x04	FB_DMA_BUFFERS	FB_DMA_CHANNELS	Reserved	*Reserved*
ADC Control	FB + 0x08	STOP_TRIG	START_TRIG	CLK_SRC	MODE_STATUS
	FB + 0x0C	CLK_DIV			
	FB + 0x10	CLK_DIV_CNTR			
	FB + 0x14	PRE_TRIGGER_CAPTURE (limited by FIFO size)			
	FB + 0x18	POST_STOP_CAPTURE			
	FB + 0x1C	SAMPLE_CNT			
	FB + 0x20	INT_ENA (Sample, Start, Stop, Threshold, Pacer Tick, etc)			
	FB + 0x24	INT_STAT		*Reserved*	
	FB + 0x28	CLK_SRC_GBL3	CLK_SRC_GBL2		*Reserved*
	FB + 0x2C	CLK_SRC_GBL7	CLK_SRC_GBL6	CLK_SRC_GBL5	CLK_SRC_GBL4
	FB + 0x30	*Reserved*			
ADC Channel 0	FB + 0x34	CH0_FRONT_END_CONFIG (Maskable register – 16-bit)			
	FB + 0x38	CH0_FIFO_DATA_CNT			
	FB + 0x3C	CH0_THRESH_INT_ENA	CH0_THRESH_INT_STAT	CH0_FILTER	*Reserved*
	FB + 0x40	CH0_THRESH_LOW			
	FB + 0x44	CH0_THRESH_HIGH			
	FB + 0x48	CH0_LAST_SAMPLE			
ADC Channel 1	FB + 0x4C	CH1_FRONT_END_CONFIG (Maskable register – 16-bit)			
	FB + 0x50	CH1_FIFO_DATA_CNT			
	FB + 0x54	CH1_THRESH_INT_ENA	CH1_THRESH_INT_STAT	CH1_FILTER	*Reserved*
	FB + 0x58	CH1_THRESH_LOW			
	FB + 0x5C	CH1_THRESH_HIGH			
	FB + 0x60	CH1_LAST_SAMPLE			
		…			
ADC Channel n	FB + 0x34 + (0x18 * n)	CHn_FRONT_END_CONFIG (Maskable register – 16-bit)			
	FB + 0x38 + (0x18 * n)	CHn_FIFO_DATA_CNT			
	FB + 0x3C + (0x18 * n)	CHn_THRESH_INT_ENA	CHn_THRESH_INT_STAT	CHn_FILTER	*Reserved*
	FB + 0x40 + (0x18 * n)	CHn_THRESH_LOW			
	FB + 0x44 + (0x18 * n)	CHn_THRESH_HIGH			
	FB + 0x48 + (0x18 * n)	CHn_LAST_SAMPLE			
ADC Channel FIFO	FB + 0x334	CH_FIFO_ACCESS (ADC Channel 0)			
	FB + 0x338	CH_FIFO_ACCESS (ADC Channel 1)			
		…			
	FB + 0x334 + (0x04 * n)	CH_FIFO_ACCESS (ADC Channel n)			

6.3.1 FB_ID (READ-ONLY)

This is the functional block ID. This register should read 0x01031000 for the ADC functional block.

6.3.2 FB_DMA_CHANNELS (READ -ONLY)

This register contains the number of DMA Channels in this Function Block. Each Channel contains a control register, and a set of Buffer Descriptor Registers.

6.3.3 FB_DMA_BUFFERS (READ-ONLY)

This register contains the number of Buffer Descriptors in each DMA Channel.

6.3.4 MODE_STATUS (READ/WRITE, READ-ONLY)

Selects the current mode of operation and indicates its triggering status.

B[3:0]: Mode

- 0x04: Uninitialized. This is the power-on state. No converter initialization has taken place. Sampling is stopped, and all counters are reset and the triggering state machine is reset. Transition to any of the other Modes will start converter initialization (sampling will not start until initialization is complete).
- 0x00: Reset. Sampling is stopped. All counters are reset and the triggering state machine is reset.
- 0x01: Paused. Sampling is stopped, but the counters and triggering state machine maintain their state.
- 0x02: Go, Single-Shot. After filling the buffer with the Post-Stop samples, capturing stops. The Mode must be set back to RESET in order to capture more samples.
- 0x03: Go, Re-arm. After filling the buffer with the Post-Stop samples and the FIFO is empty, the triggering state machine is restarted, i.e. FIFO is filled with Pre-Start samples and waits for a start trigger.

B[7:4]: Status

- 0x08: Uninitialized – The status when in the "Uninitialized" mode and the converter requires initialization.
- 0x09: Initializing
- 0x00: Stopped – The status when in the "Reset" mode, or in the "Uninitialized" mode and the converter does not require initialization.
- 0x01: Filling Pre-Trigger buffer
- 0x02: Waiting for start trigger
- 0x03: Sampling/Waiting for stop trigger
- 0x04: Filling Post-Stop buffer
- 0x05: Wait to re-arm – Waiting until local FIFO is empty so the pre-trigger buffer can be filled.
- 0x07: Done capturing

6.3.5 CLK_SRC (READ/WRITE)

Selects the source for CLK_DIV from the clock bus.

Refer to Clock Source on page 30 for list of valid values.

6.3.6 START_TRIG (READ/WRITE)

Selects the start trigger from the clock bus. CLK_DIV will start counting after the start trigger, unless PRE_TRIGGER_CAPTURE is non-zero in which case CLK_DIV will start counting immediately.

Refer to Clock Source on page 30 for list of valid values.

6.3.7 STOP_TRIG (READ/WRITE)

Selects the stop trigger from the clock bus.

Refer to Clock Source on page 30 for list of valid values.

6.3.8 CLK_DIV (READ/WRITE)

Divider for the pacer clock. Pacer Clock Frequency = (Clk_Src_Frequency) / (1 + CLK_DIV). If synchronizing with the pacer clock from another Function Block (by using one of the CLK_GBL signals), this is typically set to 0.

NOTE: Pacer Clock Frequency should equal desired sampling rate of enabled channels. The max sampling rate can be calculated by using Equation 1 on page 27.

6.3.9 CLK_DIV_CNTR (READ ONLY)

The current value of the Clock Divide Counter. This counter starts at a value of CLK_DIV, and counts down. When it reaches zero, a sample is taken. This is useful when using a slow sample clock.

6.3.10 PRE_TRIGGER_CAPTURE (READ/WRITE)

Number of samples to collect before the Start Trigger. The length is limited by the FIFO size – writing a value larger than the FIFO size will have indeterminate results.

6.3.11 POST_STOP_CAPTURE (READ/WRITE)

Number of samples to collect after the Stop Trigger.

6.3.12 SAMPLE_CNT (READ ONLY)

Total number of samples collected. This only increment while in the "Filling Pre-Trigger buffer", "Sampling/Waiting for stop trigger" and "Filling Post-Stop buffer" state. It also continues counting after a Re-Arm.

6.3.13 INT_ENA (MASKABLE READ/WRITE)

Each bit corresponds to an interrupt source. A value of '1' enables the source, and a value of '0' disables it. See below for a description of the sources.

6.3.14 INT_STAT (READ/CLEAR)

Each bit corresponds to an interrupt source. Reading a value of '1' indicates that an event has occurred. Reading a value of '0' indicates that the event has not occurred. Writing a '1' will clear that bit.

B0: Sample – A sample has been taken.

B1: Channel Threshold – One of the channels has exceeded the High or Low threshold. Check the CH_THRESH_STAT registers.

B2: Pre-Start Buffer Filled

B3: Start Trigger

B4: Stop Trigger

B5:Post-Stop Buffer Filled

B6: Sampling has completed and the FIFO is empty (all data transferred to host)

B7: Pacer – The pacer clock has ticked.

6.3.15 CLK_SRC_GBLN

NOTE: If a CLK_SRC_GBL is unassigned in all function blocks, it defaults to System Clock/Immediate.

Selects the source to drive onto Clock Bus signal N.

B[7:0]:

0x00:	Disables Clock Source
0x80:	Sample – A sample has been taken.
0x81:	Channel Threshold – One of the channels has exceeded the High or Low threshold. Check the CH_THRESH_STAT registers.
0x82:	Pre-Start Buffer Filled

0x83: Start Trigger
0x84: Stop Trigger
0x85: Post-Stop Buffer Filled
0x86: Sampling has completed and the FIFO is empty (all data transferred to host)
0x87: Pacer – The pacer clock has ticked.

6.3.16 CHn_FRONT_END_CONFIG (MASKABLE READ/WRITE)

Refer to Analog input on page 24 for more information about the front end circuit.

This provides up to 16 bits to configure the Front End for this ADC Channel, to allow adjustment of gains, ranges.

B[7:6]: CH_DELAY
- CH_DELAY [1:0] = 00: No Channel to Channel Delay
- CH_DELAY [1:0] = 01: Half Sample Clock Channel to Channel Delay
- CH_DELAY [1:0] = 10: Full Sample Clock Channel to Channel Delay
- CH_DELAY [1:0] = 11: 2 Full Sample Clock Channel to Channel Delay

B[5]: CH_ENABLE 0 = Channel Disabled 1 = Channel Enabled

B[4:2]: GAINSEL
- GAINSEL [2:0] = 000: Gain of 1
- GAINSEL [2:0] = 001: Gain of 2
- GAINSEL [2:0] = 010: Gain of 4
- GAINSEL [2:0] = 011: Gain of 8
- GAINSEL [2:0] = 100: Gain of 0.5

B[1]: BIP_UNI 0 = Bipolar operation 1 = Unipolar operation

B[0]: SE_ DIFF 0 = Single-Ended Input 1 = Differential Input

Table 18: ADC Full-Scale Settings

GAINSEL[2:0]	Signal Path Gain	Unipolar Mode	Bipolar Mode
100	0.5	n/a	±10V
000	1	0-10V	±5V
001	2	0-5V	±2.5V
010	4	0-2.5V	±1.25V
011	8	0-1.25V	±0.625V

NOTE: The Front End may take up to 800ns to settle after writing to this register.

6.3.17 CHn_FIFO_DATA_CNT (READ)

This register shows the current sample count that is available in the ADC channel FIFO.

6.3.18 CHn_FILTER (READ/WRITE)

The programmable digital filter provides a single pole Infinite Impulse Response (IIR) filter on each channel. This a unity-gain filter. The filtered data has a value of:

$$D_n = \frac{D_{n-1} \times (2^{ORDER} - 1) + NewSample}{2^{ORDER}}$$

The response of the filter is shown in the Figure 9below. The Table below shows the -3dB cutoff for each of the filter settings. Both the figure and the table are relative to the sample rate (f_s).

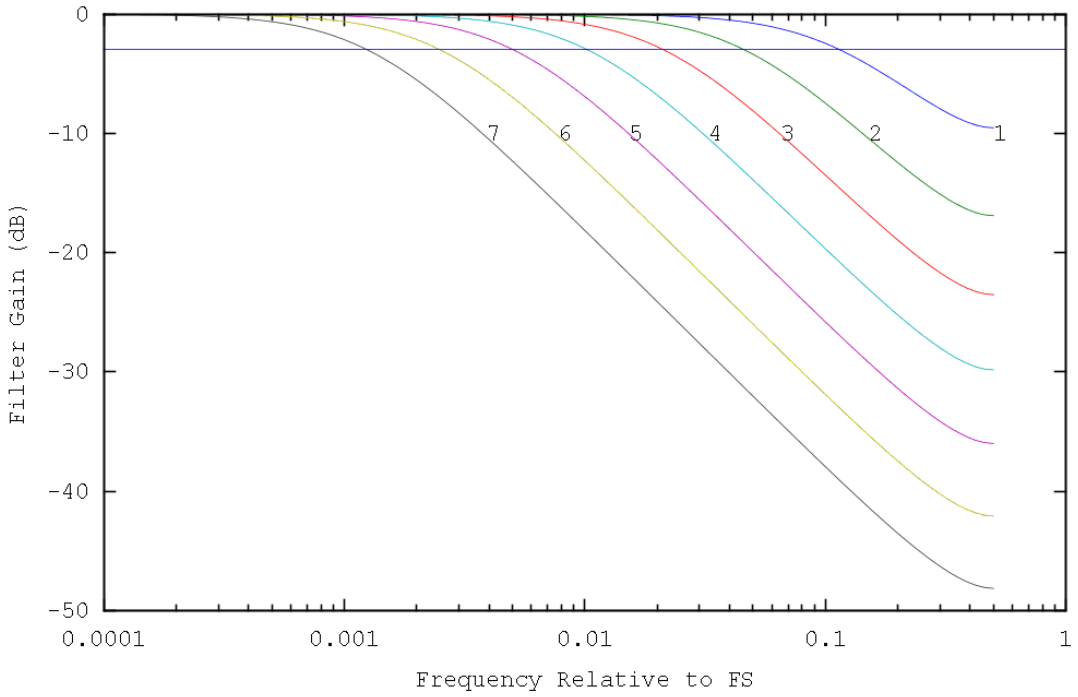


Figure 9: Filter Response with each ORDER Value

ORDER	-3 dB Cutoff
0	n/a
1	0.114791 * f_s
2	0.045995 * f_s
3	0.021236 * f_s
4	0.010255 * f_s
5	0.005042 * f_s
6	0.002501 * f_s
7	0.001246 * f_s

6.3.19 CHN_THRESH_STAT(READ/CLEAR)

This is the status register for the Threshold Detection. Reading a '1' indicates that the threshold has been crossed. Writing a '1' will clear the bit.

B0: '1' = Low Threshold has been crossed

B1: '1' = High Threshold has been crossed

6.3.20 CHN_THRESH_ENA (READ/WRITE)

This is the interrupts enable for the threshold detection. Bit defines are above. An interrupt is generated (if not already generated) each time a sample is taken and the value is above the high threshold or below the low threshold.

6.3.21 CHN_THRESH_LOW (READ/WRITE)

Signed 32-bit value indicating the low threshold. If the input signal drops below this value, an interrupt or clock can be generated until the signal goes above this value. The 3 least significant bits are ignored from the actual threshold value.

NOTE: The threshold value should not exceed the ADC range. If the threshold value exceeds the ADC range unexpected results will occur.

6.3.22 CHN_THRESH_HIGH (READ/WRITE)

Signed 32-bit value indicating the high threshold. If the input signal goes above this value, an interrupt or clock can be generated until the signal goes below this value. The 3 least significant bits are ignored from the actual threshold value.

NOTE: The threshold value should not exceed the ADC range. If the threshold value exceeds the ADC range unexpected results will occur.

6.3.23 CHN_LAST_SAMPLE (READ-ONLY)

The last sample read from the ADC Converter, after filtering. This is the same value that is written to the DMA FIFO.

6.3.24 CH_FIFO_ACCESS (READ/WRITE)

This register provides direct access to the DMA FIFO. It can be used to access the data without the use of the DMA engine. The DMA engine for this channel must be set to "Pause." Each register access advances to the next sample.

6.4 BAR2 – DAC Functional Block

This Function Block is for a Digital to Analog converter. This DAC Function block has multiple channels. There are 4 channels in this functional block, however all channels must use the same pacer clock. Each channel has its own FIFO and DMA channel.

Table 19: Multi-Channel DAC Functional Block

	Offset	0x03	0x02	0x01	0x00
Header	FB + 0x00	FB_ID			
	FB + 0x04	FB_DMA_BUFFERS	FB_DMA_CHANNELS	*Reserved*	*Reserved*
DAC Control	FB + 0x08	STOP_TRIG	START_TRIG	CLK_SRC	MODE_STATUS
	FB + 0x0C	CLK_DIV			
	FB + 0x10	CLK_DIV_CNTR			
	FB + 0x14	Reserved			
	FB + 0x18	POST_STOP_CONVERSIONS			
	FB + 0x1C	CONVERSION_CNT			
	FB + 0x20	INT_ENA (Conversion, Start, Stop, Error, Channel)			
	FB + 0x24	INT_STAT		*Reserved*	
	FB + 0x28	CLK_SRC_GBL3	CLK_SRC_GBL2		BOOKMARK_TRIG
	FB + 0x2C	CLK_SRC_GBL7	CLK_SRC_GBL6	CLK_SRC_GBL5	CLK_SRC_GBL4
	FB + 0x30	*Reserved*			
DAC Channel 0	FB + 0x34	CH0_FRONT_END_CONFIG (Maskable register – 16-bit)			
	FB + 0x38	CH0_FIFO_DATA_CNT			
	FB + 0x3C	CH0_MARK_INT_ENA	CH0_MARK_INT_STAT	*Reserved*	*Reserved*
	FB + 0x40	*Reserved*			
	FB + 0x44	CH0_LAST_CONVERSION			
DAC Channel 1	FB + 0x48	CH1_FRONT_END_CONFIG (Maskable register – 16-bit)			
	FB + 0x4C	CH1_FIFO_DATA_CNT			
	FB + 0x50	CH1_ MARK_INT_ENA	CH1_ MARK_INT_STAT	*Reserved*	*Reserved*
	FB + 0x54	*Reserved*			
	FB + 0x58	CH1_LAST_ CONVERSION			
DAC Channel 2	FB + 0x5C	CH2_FRONT_END_CONFIG (Maskable register – 16-bit)			
	FB + 0x60	CH2_FIFO_DATA_CNT			
	FB + 0x64	CH2_ MARK_INT_ENA	CH2_ MARK_INT_STAT	*Reserved*	*Reserved*
	FB + 0x68	*Reserved*			
	FB + 0x6C	CH2_LAST_ CONVERSION			
DAC Channel 3	FB + 0x70	CH3_FRONT_END_CONFIG (Maskable register – 16-bit)			
	FB + 0x74	CH3_FIFO_DATA_CNT			
	FB + 0x78	CH3_ MARK_INT_ENA	CH3_ MARK_INT_STAT	*Reserved*	*Reserved*
	FB + 0x7C	*Reserved*			
	FB + 0x80	CH3_LAST_ CONVERSION			
DAC FIFO	FB + 0x84	CH_FIFO_ACCESS (DAC Channel 0)			
	FB + 0x88	CH_FIFO_ACCESS (DAC Channel 1)			
	FB + 0x8C	CH_FIFO_ACCESS (DAC Channel 2)			
	FB + 0x90	CH_FIFO_ACCESS (DAC Channel 3)			

6.4.1 FB_ID (READ-ONLY)

This is the functional block ID. This register should read 0x01032000 for the D\A functional block.

6.4.2 FB_DMA_CHANNELS (READ -ONLY)

This register contains the number of DMA Channels in this Function Block. Each Channel contains a control register, and a set of Buffer Descriptor Registers.

6.4.3 FB_DMA_BUFFERS (READ-ONLY)

This register contains the number of Buffer Descriptors in each DMA Channel.

6.4.4 MODE_STATUS (READ/WRITE, READ-ONLY)

Selects the current mode of operation and indicates its triggering status.

B[3:0]: Mode

- 0x04: Uninitialized. This is the power-on state. No converter initialization has taken place. Sampling is stopped, and all counters are reset and the triggering state machine is reset. Transition to any of the other Modes will start converter initialization (sampling will not start until initialization is complete).
- 0x00: Reset. Sampling is stopped. All counters are reset and the triggering state machine is reset.
- 0x01: Paused. Sampling is stopped, but the counters and triggering state machine maintain their state.
- 0x02: Go, Single-Shot. After converting the Post-Stop number of values, converting stops. The Mode must be set back to RESET in order to convert more values.
- 0x03: Go, Re-arm. After converting the Post-Stop number of values, the triggering state machine is restarted. DAC data is resumed from that last value sent.

B[7:4]: Status

- 0x08: Uninitialized – The status when in the "Uninitialized" mode and the converter requires initialization.
- 0x09: Initializing
- 0x00: Stopped – The status when in the "Reset" mode, or in the "Uninitialized" mode and the converter does not require initialization.
- 0x01: *Reserved*
- 0x02: Waiting for start trigger
- 0x03: Converting/Waiting for stop trigger
- 0x04: Output Post-Stop buffer
- 0x05: Wait to re-arm
- 0x07: Done capturing

6.4.5 CLK_SRC (READ/WRITE)

Selects the source for CLK_DIV from the clock bus.

Refer to Clock Source on page 30 for list of valid values.

6.4.6 START_TRIG (READ/WRITE)

Selects the start trigger from the clock bus. CLK_DIV will start counting after the start trigger.

Refer to Clock Source on page 30 for list of valid values.

6.4.7 STOP_TRIG (READ/WRITE)

Selects the stop trigger from the clock bus.

Refer to Clock Source on page 30 for list of valid values.

6.4.8 CLK_DIV (READ/WRITE)

Divider for the pacer clock. Pacer Clock Frequency = (Clk_Src_Frequency) / (1 + CLK_DIV). If synchronizing with the pacer clock from another Function Block (by using one of the CLK_GBL signals), this is typically set to 0.

6.4.9 CLK_DIV_CNTR (READ ONLY)

The current value of the Clock Divide Counter. This counter starts at a value of CLK_DIV, and counts down. When it reaches zero, a sample is taken. This is useful when using a slow sample clock.

6.4.10 POST_STOP_CONVERSIONS (READ/WRITE)

Number of conversions to send after the Stop Trigger.

6.4.11 CONVERSION_CNT (READ ONLY)

Total number of conversions. This only increment in while in "Converting/Waiting for stop trigger" and "Output Post-Stop buffer" state. It also continues counting after a Re-Arm.

6.4.12 INT_ENA (MASKABLE READ/WRITE)

Each bit corresponds to an interrupt source. A value of '1' enables the source, and a value of '0' disables it. See below for a description of the sources.

6.4.13 INT_STAT (READ/CLEAR)

Each bit corresponds to an interrupt source. Reading a value of '1' indicates that an event has occurred. Reading a value of '0' indicates that the event has not occurred. Writing a '1' will clear that bit.

B[0]: Conversion – A value has been sent.

B[1]: Channel Marker – One of the channels has an enabled marker.

B[2]: Reserved

B[3]: Start Trigger

B[4]: Stop Trigger

B[5]: Post-Stop Conversions Completed

6.4.14 CLK_SRC_GBLN

NOTE: If a CLK_SRC_GBL is unassigned in all function blocks, it defaults to System Clock/Immediate.

Selects the source to drive onto Clock Bus signal N. Values are:

B[7:0] Clock Source Select

0x00: Disable Clock Source
0x80: Conversion – A value has been sent.
0x81: Channel Marker – One of the channels has an enabled marker.
0x82: Reserved
0x83: Start Trigger
0x84: Stop Trigger
0x85: Post-Stop Conversions Completed

6.4.15 CH_FRONT_END_CONFIG (MASKABLE READ/WRITE)

This provides configuration to the Front End for this DAC Channel, to allow adjustment of gains, ranges.

B[2]: DAC_ENABLE	0 =Output Disabled	1 = Output Enabled
B[1]: DABIP_UNI	0 = Unipolar operation	1 = Bipolar operation
B[0]: GAIN	0 = Gain of 1	1 = Gain of 2

Table 20: DAC Full-Scale Settings

GAIN	Unipolar Mode	Bipolar Mode
0	0-5V	±5V
1	0-10V	±10V

NOTE: The Front End may take up to 100us to settle after writing to this register.

6.4.16 CHN_FIFO_DATA_CNT (READ)

This register shows the current sample count that is available in the DAC channel FIFO.

6.4.17 CH_MARKER_STAT(READ/CLEAR)

This is the status register for the Data Markers. Reading a '1' indicates that the Data Marker has been asserted. Writing a '1' will clear the bit.

The upper eight bits of the DAC value can be used for Markers. These Markers can be used to generate an interrupt when a certain part of the waveform is sent to the DAC. This allows an automated indication to the application software as to the state of the data being sent to the DAC. Marker bit 7 corresponds to bit 31 of the DAC data, and Marker bit 0 corresponds to bit 24 of the DAC data.

6.4.18 CH_MARKER_ENA (READ/WRITE)

These are interrupts enables for the Data Markers. Bit defines are above.

6.4.19 CH_LAST_CONVERSION (READ/WRITE)

The last value sent to the DAC Converter.

B[31:24]: DAC Markers

B[12:0]: DAC Data

If the current Mode is "Reset" or the associated DMA engine is set to "Clear", a write to this register will immediately update the DAC Converter.

6.4.20 CH_FIFO_ACCESS (READ/WRITE)

This register provides direct access to the DMA FIFO. It can be used to access the data without the use of the DMA engine. The DMA engine for this channel must be set to "Pause." Each register access advances to the next sample.

6.5 BAR2 – Advanced Digital I/O Functional Block

This function block is for 32 bi-directional digital I/O. The Advanced Digital I/O (ADIO) Function block has multiple channels. There are 3 channels in this functional block: IN, OUT and DIRECTION. Each channel has its own FIFO and DMA channel.

Table 21: Digital I/O Functional Block

	Offset	0x03	0x02	0x01	0x00
Header	FB + 0x00	FB_ID			
	FB + 0x04	FB_DMA_BUFFERS	FB_DMA_CHANNELS	*Reserved*	*Reserved*
Digital I/O Control	FB + 0x08	STOP_TRIG	START_TRIG	CLK_SRC	MODE_STATUS
	FB + 0x0C	CLK_DIV			
	FB + 0x10	CLK_DIV_CNTR			
	FB + 0x14	PRE_TRIGGER_CAPTURE (limited by FIFO size)			
	FB + 0x18	POST_STOP_CAPTURE			
	FB + 0x1C	SAMPLE_CNT			
	FB + 0x20	INT_ENA (Conversion, Start, Stop, Error, Channel)			
	FB + 0x24	INT_STAT		*Reserved*	
	FB + 0x28	CLK_SRC_GBL3	CLK_SRC_GBL2		*Reserved*
	FB + 0x2C	CLK_SRC_GBL7	CLK_SRC_GBL6	CLK_SRC_GBL5	CLK_SRC_GBL4
Digital I/O Channels	FB + 0x30	DIO_IN			
	FB + 0x34	DIO_OUT			
	FB + 0x38	DIO_DIR			
	FB + 0x3C	*Reserved*			ADV_INT_MODE
	FB + 0x40	ADV_INT_MASK			
	FB + 0x44	ADV_INT_COMP			
	FB + 0x48	ADV_INT_CAPT			
	FB + 0x4C	*Reserved*		P_BUS_READY_EN	P_BUS_EN
Digital I/O FIFO	FB + 0x50	CH_FIFO_ACCESS (DIO_IN)			
	FB + 0x54	CH_FIFO_ACCESS (DIO_OUT)			
	FB + 0x58	CH_FIFO_ACCESS (DIO_DIR)			

6.5.1 FB_ID (READ-ONLY)

This is the functional block ID. This register should read 0x01003001 for the Advanced Digital I/O functional block.

6.5.2 FB_DMA_CHANNELS (READ -ONLY)

This register contains the number of DMA Channels in this Function Block. Each Channel contains a control register, and a set of Buffer Descriptor Registers.

6.5.3 FB_DMA_BUFFERS (READ-ONLY)

This register contains the number of Buffer Descriptors in each DMA Channel.

6.5.4 MODE_STATUS (READ/WRITE, READ-ONLY)

Selects the current mode of operation and indicates its triggering status.

B[3:0]: Mode

- 0x04: Uninitialized. This is the power-on state. No converter initialization has taken place. Sampling is stopped, and all counters are reset and the triggering state machine is reset. Transition to any of the other Modes will start converter initialization (sampling will not start until initialization is complete).
- 0x00: Reset. Sampling is stopped. All counters are reset and the triggering state machine is reset.
- 0x01: Paused. Sampling is stopped, but the counters and triggering state machine maintain their state.
- 0x02: Go, Single-Shot. After converting the Post-Stop number of values, converting stops. The Mode must be set back to RESET in order to convert more values.

- 0x03: Go, Re-arm. After converting the Post-Stop number of values, the triggering state machine is restarted. ADIO data is resumed from that last value sent.

NOTE: In Parallel Bus Mode, unexpected results may occur when setting Mode to GO before setting the VALID, CLK and READY bits on both the Transmitter and Receiver

B[7:4]: Status

- 0x08: Uninitialized – The status when in the "Uninitialized" mode and the converter requires initialization.
- 0x09: Initializing
- 0x00: Stopped – The status when in the "Reset" mode, or in the "Uninitialized" mode and the converter does not require initialization.
- 0x01: Filling Pre-Trigger buffer
- 0x02: Waiting for start trigger
- 0x03: Sampling/Waiting for stop trigger
- 0x04: Filling Post-Stop buffer
- 0x05: Wait to re-arm – Waiting until local FIFO is empty so the pre-trigger buffer can be filled.
- 0x07: Done capturing

6.5.5 CLK_SRC (READ/WRITE)

Selects the source for CLK_DIV from the clock bus.

Refer to Clock Source on page 30 for list of valid values.

6.5.6 START_TRIG (READ/WRITE)

Selects the start trigger from the clock bus. CLK_DIV will start counting after the start trigger.

Refer to Clock Source on page 30 for list of valid values.

6.5.7 STOP_TRIG (READ/WRITE)

Selects the stop trigger from the clock bus.

Refer to Clock Source on page 30 for list of valid values.

6.5.8 CLK_DIV (READ/WRITE)

Divider for the pacer clock. Pacer Clock Frequency = (Clk_Src_Frequency) / (1 + CLK_DIV). If synchronizing with the pacer clock from another Function Block (by using one of the CLK_GBL signals), this is typically set to 0.

NOTE: The max clock frequency for the ADIO FB is 4 MHz, CLK_DIV needs set to a minimum of 9 for this FB to work properly.

6.5.9 CLK_DIV_CNTR (READ ONLY)

The current value of the Clock Divide Counter. This counter starts at a value of CLK_DIV, and counts down. When it reaches zero, a sample is taken. This is useful when using a slow sample clock.

6.5.10 PRE_TRIGGER_CAPTURE (READ/WRITE)

Number of samples to collect before the Start Trigger. The length is limited by the FIFO size – writing a value larger than the FIFO size will have indeterminate results.

6.5.11 POST_STOP_CAPTURE (READ/WRITE)

Number of samples to collect after the Stop Trigger.

6.5.12 SAMPLE_CNT (READ ONLY)

Total number of samples collected. This only increment while in the "Filling Pre-Trigger buffer", "Sampling/Waiting for stop trigger" and "Filling Post-Stop buffer" state. It also continues counting after a Re-Arm.

While in Parallel Mode, samples collected is only accurate when P_BUS_CLK is low.

6.5.13 INT_ENA (MASKABLE READ/WRITE)

Each bit corresponds to an interrupt source. A value of '1' enables the source, and a value of '0' disables it. See below for a description of the sources.

6.5.14 INT_STAT (READ/CLEAR)

Each bit corresponds to an interrupt source. Reading a value of '1' indicates that an event has occurred. Reading a value of '0' indicates that the event has not occurred. Writing a '1' will clear that bit.

B0: Sample – A sample has been taken.
B1: Advanced Int
B2: Pre-Start Buffer Filled
B3: Start Trigger
B4: Stop Trigger
B5:Post-Stop Buffer Filled
B6: Sampling has completed and the FIFO is empty (all data transferred to host)
B7: Pacer – The pacer clock has ticked.
B8: CN3 5V Over Current
B9: CN4 5V Over Current

6.5.15 CLK_SRC_GBLN

NOTE: If a CLK_SRC_GBL is unassigned in all function blocks, it defaults to System Clock/Immediate.

Selects the source to drive onto Clock Bus signal N.

B[7:0]:

0x00:	Disables Clock Source
0x80:	Sample – A sample has been taken.
0x81:	Advanced Int
0x82:	Pre-Start Buffer Filled
0x83:	Start Trigger
0x84:	Stop Trigger
0x85:	Post-Stop Buffer Filled
0x86:	Sampling has completed and the FIFO is empty (all data transferred to host)

6.5.16 DIO_INPUT (READ ONLY)

This register provides the current value on the Digital I/O lines regardless of pin direction. The bits in the register correspond with the Digital I/O pins as follows:

This is the same value that is written to the DMA INPUT FIFO.

When P_BUS_EN is enabled and the P_BUS_CLK pin direction is set to high this Register and DMA INPUT FIFO will only be updated when receives a high clock pulse and P_BUS_VALID is high.

Bit	CN4 Pin Number	Signal Name	
		P_BUS_EN = 0	P_BUS_EN = 1
31	24	DIO31	P_BUS_CLK
30	26	DIO30	P_BUS_READY
29	28	DIO29	P_BUS_VALID
28	30	DIO28	P_BUS_DATA28
27	32	DIO27	P_BUS_DATA27
26	34	DIO26	P_BUS_DATA26
25	36	DIO25	P_BUS_DATA25
24	38	DIO24	P_BUS_DATA24
23	23	DIO23	P_BUS_DATA23
22	25	DIO22	P_BUS_DATA22
21	27	DIO21	P_BUS_DATA21
20	29	DIO20	P_BUS_DATA20
19	31	DIO19	P_BUS_DATA19
18	33	DIO18	P_BUS_DATA18
17	35	DIO17	P_BUS_DATA17
16	37	DIO16	P_BUS_DATA16

Bit	CN3 Pin Number	Signal Name	
		P_BUS_EN = 0	P_BUS_EN = 1
15	24	DIO15	P_BUS_DATA15
14	26	DIO14	P_BUS_DATA14
13	28	DIO13	P_BUS_DATA13
12	30	DIO12	P_BUS_DATA12
11	32	DIO11	P_BUS_DATA11
10	34	DIO10	P_BUS_DATA10
9	36	DIO9	P_BUS_DATA9
8	38	DIO8	P_BUS_DATA8
7	23	DIO7	P_BUS_DATA7
6	25	DIO6	P_BUS_DATA6
5	27	DIO5	P_BUS_DATA5
4	29	DIO4	P_BUS_DATA4
3	31	DIO3	P_BUS_DATA3
2	33	DIO2	P_BUS_DATA2
1	35	DIO1	P_BUS_DATA1
0	37	DIO0	P_BUS_DATA0

6.5.17 DIO_OUTPUT (READ/WRITE)

The last value sent to the Digital I/O Output.

If the current Mode is "Reset" or the associated DMA engine is set to "Clear", a write to this register will immediately update the Digital I/O Output. Bit assignments are the same as above.

When P_BUS_EN is enabled

B[31:29]: Reserved
B[28:0]: P_BUS_DATA

6.5.18 DIO_DIRECTION (READ/WRITE)

The last value sent to the Digital I/O Direction.

If the current Mode is "Reset" or the associated DMA engine is set to "Clear", a write to this register will immediately update the Digital I/O Direction. Bit assignments are the same as above.

Selects the direction of the I/O bit. 0=input, 1=output.

All pins default to inputs at power-up.

When P_BUS_EN is enabled

B[31]: P_BUS_CLK: When set high, data in the Digital I/O DMA OUTPUT FIFO and high clock pulse (50ns) will be sent every Digital I/O pacer clock. When set low, DIO_INPUT Register and DMA INPUT FIFO will be updated every high clock pulse received.

B[30]: P_BUS_READY: When set high, this will output high ready signal once the Digital I/O DMA INPUT FIFO is setup and started. When using the module to send data, set this bit low to receive the ready signal.

B[29]: P_BUS_VALID: When set high, this will output high valid signal once the Digital I/O DMA OUTPUT FIFO is setup and started. When using the module to receive data, set this bit low to receive the valid signal.

B[28:0]: P_BUS_DATA: Sets the direction of the I/O bit. 0=input, 1=output.

6.5.19 ADV_INT_MODE (READ/WRITE)

Set the current mode for the advance interrupts.

B[1:0]: Mode

- 0x0: Disabled. This is the power-on state. In this mode no advanced interrupts will occur.
- 0x1: Match. A Match interrupt is generated when all un-masked bits in the Compare register match the input value of the port. This is when the following expression is true:

  ((DIO_INPUT xor ADV_INT_COMP) and not ADV_INT_MASK) = '0'
- 0x2: Event Mode. An Event interrupt is generated when any un-masked input port bit changes. This is when the following expression is:

  ((DIO_INPUT xor ADV_INT_ CAPT) and not ADV_INT_MASK) = '1'

  The ADV_INT_CAPT register is updated at every advanced interrupt or event.

6.5.20 ADV_INT_MASK (READ/WRITE)

This register determines if a bit is checked for the advanced interrupts.

0 = Bit is used for match/event 1 = Bit is ignored

6.5.21 ADV_INT_COMP (READ/WRITE)

The compare register is used for the Match interrupt. When all selected bits in this register match all selected bits on the DIO_INPUT register, an interrupt is generated.

6.5.22 ADV_INT_CAPT (READ/WRITE)

The Capture register latches the input ports when an interrupt is generated. All values are latched, regardless of the ADV_INT_MASK register, or DIO_DIRECTION.

This register can be written to when ADV_INT_MODE is set to Disabled.

6.5.23 P_BUS_EN (READ/WRITE)

The P_BUS_EN register is used to enable the parallel bus feature of the digital I/O.

0 = Disabled 1 = Enabled

6.5.24 P_BUS_READY_EN (READ/WRITE)

The P_BUS_READY_EN register is used to enable the parallel bus ready signal check of the digital I/O. When this bit is enabled the P_BUS_CLK will not be outputted until P_BUS_READY is high.

0 = Disabled 1 = Enabled

6.5.25 CH_FIFO_ACCESS (READ/WRITE)

This register provides direct access to the DMA FIFO. It can be used to access the data without the use of the DMA engine. The DMA engine for this channel must be set to "Pause." Each register access advances to the next sample.

6.6 BAR2 – External Clocking Functional Block

This function block provides an interface to the External Clocking. It is used to input or output the source of the CLK_SRC_GLBn.

Table 22:External Clocking Functional Block

	Offset	0x03	0x02	0x01	0x00
Header	FB + 0x00	FB_ID			
	FB + 0x04	FB_DMA_BUFFERS	FB_DMA_CHANNELS	*Reserved*	*Reserved*
	FB + 0x08	EXT_CLK_EDGE	EXT_CLK_DIR	EXT_CLK_GATE_IN	EXT_CLK_IN
	FB + 0x0C	EXT_CLK_PW5	EXT_CLK_PW4	EXT_CLK_PW3	EXT_CLK_PW2
	FB + 0x10	EXT_CLK3_CFG	EXT_CLK2_CFG	EXT_CLK_PW7	EXT_CLK_PW6
	FB + 0x14	EXT_CLK7_CFG	EXT_CLK6_CFG	EXT_CLK5_CFG	EXT_CLK4_CFG

6.6.1 FB_ID (READ-ONLY)

This is the functional block ID. This register should read 0x00000002 for the External Clocking functional block.

6.6.2 FB_DMA_CHANNELS (READ -ONLY)

Has no DMA channels, reads 0

6.6.3 FB_DMA_BUFFERS (READ-ONLY)

Has no DMA buffers, reads 0

6.6.4 EXT_CLK_IN (READ-ONLY)

This register provides the current value on the External Clocking lines. The bits in the register correspond with the External Clocking pins as follows:

Bit	CN3 Pin Number	CLK_SRC_GLBn	Signal
5	45	7	EXT_CLK_7
4	44	6	EXT_CLK_6
3	43	5	EXT_CLK_5
2	42	4	EXT_CLK_4
1	41	3	EXT_CLK_3
0	39	2	EXT_CLK_2

6.6.5 EXT_CLK_GATE_IN (READ ONLY)

This register provides the current value on the External Clocking Gates lines. External Clocking Gates can only be used when inputting an external clock. The bits in the register correspond with the External Clocking Gates pins as follows:

Bit	CN4 Pin Number	CLK_SRC_GLBn	Signal
5	45	7	EXT_CLK_GATE7
4	44	6	EXT_CLK_GATE6
3	43	5	EXT_CLK_GATE5
2	42	4	EXT_CLK_GATE4
1	41	3	EXT_CLK_GATE3
0	39	2	EXT_CLK_GATE2

6.6.6 EXT_CLK_DIR (READ/WRITE)

Selects the direction of the External Clocking bits. 0=input, 1=output.

All pins default to inputs at power-up.

6.6.7 EXT_CLK_EDGE (READ/WRITE)

Selects which edge detect to trigger on. This is a bit settable register.

0= Rising Edge Detect, 1= Fall Edge Detect.

6.6.8 EXT_CLK_PWN (READ/WRITE)

This register is used to increase the pulse width of the clock. When set to 0x00 the pulse width is high for 25ns. By incrementing this register by 0x01 the pulse width stays high for additional 25ns.

NOTE: If EXT_CLK_PWn is set to be wider than the EXT_CLKn_CFG Clock Frequency the signal will just stay high.

6.6.9 EXT_CLKN_CFG (READ/WRITE)

Selects clocking method.

B[7:0]:

0x00:	Disables External Clocking.
0x80:	Not Gated: CLK_SRC_GLBn will be inputted/outputted independent of the CLK_GBLn corresponding gate value.
0x81:	Clock Gated (High): CLK_SRC_GLBn will be inputted when the CLK_GBLn corresponding gate value is high, this doesn't affect when outputting a clock.
0x82:	Clock Gated (Low): CLK_SRC_GLBn will be inputted when the CLK_GBLn corresponding gate value is low, this doesn't affect when outputting a clock.

7 Calibration

This section describes how to calibrate the DM35425HR using the trimpots on the module. The trimpots calibrate the A/D converter gain and offset.

The D/A converter does not need to be calibrated. The offset and full-scale performance of the module's A/D converter is factory- calibrated. Any time you suspect inaccurate readings, you can check the accuracy of your conversions using the procedure in this section, and make adjustments as necessary.

Calibration is done with the module installed in your system. Power up the system and let the DM35425HR circuitry stabilize for 15 minutes before calibration.

7.1 Required Equipment

The following equipment is required for calibration:

- Precision voltage source: -10 to +10 V
- Digital voltmeter: 5 ½ digits
- Small screwdriver (for trimpot adjustment)

The figure below shows the DM35425HR layout with the trimpots located along the top.

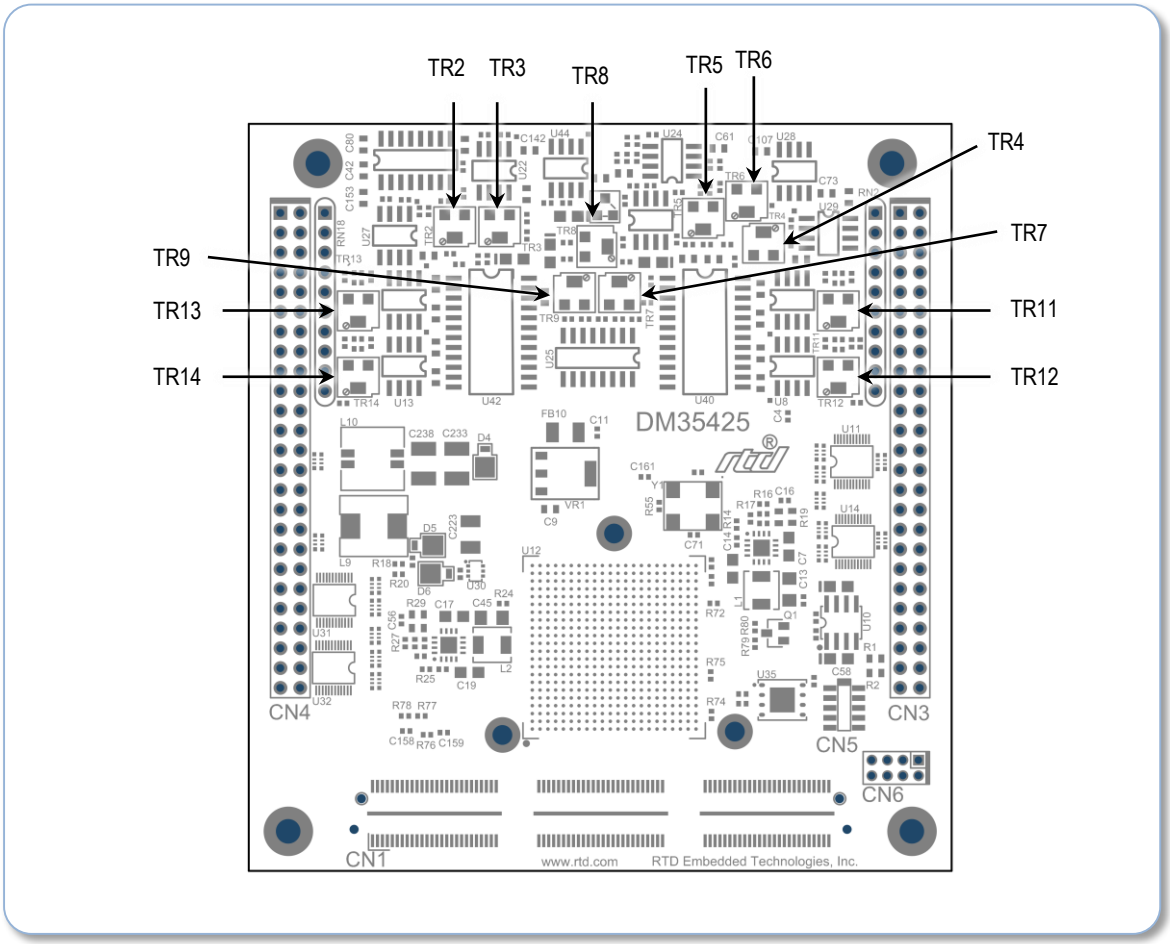


Figure 10: DM35425HR Trimpots

7.2 ADC Calibration

Two procedures are used to calibrate the ADC for all input voltage ranges. The first procedure calibrates the converter for the bipolar ranges (±5, ±10 V), and the second procedure calibrates the unipolar range (0 to +10 V). Table 25 shows the ideal input voltage for each bit weight for the bipolar ranges, and Table 27 shows the ideal voltage for each bit weight for the unipolar ranges.

7.2.1 Bipolar Calibration

Bipolar Range Adjustment: -5 to +5 V

Two adjustments are made to calibrate the ADC for the bipolar range of -5 to +5 V. One is the offset adjustment, and the other is the full scale, or gain, adjustment. Trimpot **TR4** is used to make the offset adjustment, and trimpot **TR5** is used for gain adjustment. Before making these adjustments, make sure that the board is programmed for a range of ±5 V.

Use AIN0 and set it for a gain of 1 while calibrating the board. Connect you precision voltage source to AIN0. Set the voltage source to -1.22070 mV, start a conversion and read the resulting data. Adjust trimpot **TR4** until the reading flickers between the values listed in the table below. Next, set the voltage to -4.99878 V, and repeat the procedure, this time adjusting **TR5** until the data flickers between the values in the table below.

Table 23: Data Values for Calibrating Bipolar 10 V Range

	Offset (TR4) Input Voltage = -1.22mV	Converter Gain (TR5) Input Voltage = -4.99878V
ADC Converted Data	0000 0000 0000	1000 0000 0000
	1111 1111 1111	1000 0000 0001

Bipolar Range Adjustment: -10 to +10 V

To adjust the bipolar 20 V range (-10 to +10 V), program the board for ± 10 V input range. Then, set the input voltage to +5.0000 V and adjust **TR2** until the output matches the data in the table below.

Table 24: Data Values for Calibrating Bipolar 20 V Range

	Offset (TR4) Input Voltage = +5.0000 V
ADC Converted Data	0100 0000 0000

Bipolar Range Ideal Bit Weight

Below is a table listing the ideal input voltage for each bit weight for the bipolar ranges.

Table 25: ADC Bit Weights, Bipolar

Sign	ADC Bit Weight	Ideal Output Voltage (mV)	
		-5 to +5 V	-10 to +10 V
1	1111 1111 1111	-2.44	-4.88
1	1000 0000 0000	-5000.00	-10000.00
0	0100 0000 0000	+2500.00	+5000.00
0	0010 0000 0000	+1250.00	+2500.00
0	0001 0000 0000	+625.00	+1250.00
0	0000 1000 0000	+312.50	+625.00
0	0000 0100 0000	+156.25	+312.50
0	0000 0010 0000	+78.13	+156.25
0	0000 0001 0000	+39.06	+78.13
0	0000 0000 1000	+19.53	+39.06
0	0000 0000 0100	+9.77	+19.53
0	0000 0000 0010	+4.88	+9.77
0	0000 0000 0001	+2.44	+4.88
0	0000 0000 0000	0.00	0.00

7.2.2 Unipolar Calibration

One adjustment is made to calibrate the ADC for the unipolar range of 0 to +10 V. Trimpot **TR6** is used to make the offset adjustment. This calibration procedure is performed with the module programmed for a 0 to +10 V input range. Before making these adjustments, make sure that the module is programmed properly and has been calibrated for bipolar ranges.

Use AIN0 and set it for a gain of 1 while calibrating the board. Connect your precision voltage source to AIN0. Set the voltage source to +1.22070 mV, start a conversion, and read the resulting data. Adjust trimpot **TR6** until the data flickers between the values listed in the table below.

Table 26: Data Values for Calibrating Bipolar 20 V Range

	Offset (TR6) Input Voltage = +1.22070 mV
ADC Converted Data	0000 0000 0000
	0000 0000 0001

Unipolar Range Ideal Bit Weight

Below is a table listing the ideal input voltage for each bit weight for the unipolar ranges.

Table 27: ADC Bit Weights, Unipolar

Sign	ADC Bit Weight	Ideal Output Voltage (mV)
		0 to +10 V
0	1111 1111 1111	+9997.60
0	1000 0000 0000	+5000.00
0	0100 0000 0000	+2500.00
0	0010 0000 0000	+1250.00
0	0001 0000 0000	+625.00
0	0000 1000 0000	+312.50
0	0000 0100 0000	+156.25
0	0000 0010 0000	+78.13
0	0000 0001 0000	+39.06
0	0000 0000 1000	+19.53
0	0000 0000 0100	+9.77
0	0000 0000 0010	+4.88
0	0000 0000 0001	+2.44
0	0000 0000 0000	0.00

7.2.3 GAIN ADJUSTMENT

Should you find it necessary to check any of the programmable gain settings, the following table will show the proper trimpot to adjust.

Table 28: Trimpots for Calibrating ADC Gain

Gain	Trimpot
X2	TR7
X4	TR8
X8	TR9

7.3 DAC Calibration

The DAC circuit requires gain calibration. TR11-TR14 adjust the DAC gain. The table below provides, for your reference, a list of the input bits and their corresponding ideal output voltages for each of the three output ranges.

Table 29: DAC Bit Weights

DAC Bit Weight	Ideal Output Voltages (mV)			
	-5 to +5 V	0 to +5 V	-10 to +10 V	0 to +10 V
1111 1111 1111	+4997.56	+4998.78	+9995.12	+9997.56
1000 0000 0000	0	+2500.00	0000.00	+5000.00
0100 0000 0000	-2500.00	+1250.00	-5000.00	+2500.00
0010 0000 0000	-3750.00	+625.00	-7500.00	+1250.00
0001 0000 0000	-4375.00	+312.50	-8750.00	+625.00
0000 1000 0000	-4687.50	+156.25	-9375.00	+312.50
0000 0100 0000	-4843.75	+78.13	-9687.50	+156.25
0000 0010 0000	-4921.88	+39.06	-9843.75	+78.13
0000 0001 0000	-4960.94	+19.53	-9921.88	+39.06
0000 0000 1000	-4980.47	+9.77	-9960.94	+19.53
0000 0000 0100	-4990.24	+4.88	-9980.47	+9.77
0000 0000 0010	-4995.12	+2.44	-9990.23	+4.88
0000 0000 0001	-4997.56	+1.22	-9995.12	+2.44
0000 0000 0000	-5000.00	0.00	-10000.00	0.00

Table 30: Trimpots for Calibrating DAC Gain

DAC	Trimpot
AOUT0	TR11
AOUT1	TR12
AOUT2	TR13
AOUT3	TR14

8 Troubleshooting

If you are having problems with your system, please try the following initial steps:

- **Simplify the System** – Remove modules one at a time from your system to see if there is a specific module that is causing a problem. Perform you troubleshooting with the least number of modules in the system possible.

- **Swap Components** – Try replacing parts in the system one at a time with similar parts to determine if a part is faulty or if a type of part is configured incorrectly.

If problems persist, or you have questions about configuring this product, contact RTD Embedded Technologies via the following methods:

Phone: +1-814-234-8087
E-Mail: techsupport@rtd.com

Be sure to check the RTD web site (http://www.rtd.com) frequently for product updates, including newer versions of the board manual and application software.

9 Additional Information

9.1 PC/104 Specifications

A copy of the latest PC/104 specifications can be found on the webpage for the PC/104 Embedded Consortium:

www.pc104.org

9.2 PCI and PCI Express Specification

A copy of the latest PCI and PCI Express specifications can be found on the webpage for the PCI Special Interest Group:

www.pcisig.com

10 Limited Warranty

RTD Embedded Technologies, Inc. warrants the hardware and software products it manufactures and produces to be free from defects in materials and workmanship for one year following the date of shipment from RTD Embedded Technologies, Inc. This warranty is limited to the original purchaser of product and is not transferable.

During the one year warranty period, RTD Embedded Technologies will repair or replace, at its option, any defective products or parts at no additional charge, provided that the product is returned, shipping prepaid, to RTD Embedded Technologies. All replaced parts and products become the property of RTD Embedded Technologies. Before returning any product for repair, customers are required to contact the factory for a Return Material Authorization (RMA) number.

This limited warranty does not extend to any products which have been damaged as a result of accident, misuse, abuse (such as: use of incorrect input voltages, improper or insufficient ventilation, failure to follow the operating instructions that are provided by RTD Embedded Technologies, “acts of God” or other contingencies beyond the control of RTD Embedded Technologies), or as a result of service or modification by anyone other than RTD Embedded Technologies. Except as expressly set forth above, no other warranties are expressed or implied, including, but not limited to, any implied warranties of merchantability and fitness for a particular purpose, and RTD Embedded Technologies expressly disclaims all warranties not stated herein. All implied warranties, including implied warranties for merchantability and fitness for a particular purpose, are limited to the duration of this warranty. In the event the product is not free from defects as warranted above, the purchaser's sole remedy shall be repair or replacement as provided above. Under no circumstances will RTD Embedded Technologies be liable to the purchaser or any user for any damages, including any incidental or consequential damages, expenses, lost profits, lost savings, or other damages arising out of the use or inability to use the product.

Some states do not allow the exclusion or limitation of incidental or consequential damages for consumer products, and some states do not allow limitations on how long an implied warranty lasts, so the above limitations or exclusions may not apply to you.

This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

RTD Embedded Technologies, Inc.
103 Innovation Boulevard
State College, PA 16803 USA
Telephone: 814-234-8087
Fax: 814-234-5218

www.rtd.com

sales@rtd.com
techsupport@rtd.com