Digital to Analog
DA2
Function Module

MODULE MANUAL

Revision History

Revision	Revision Date	Description
A	2/1/2018	Initial release
A1	2/28/2018	ECO C05384: Fixes to AD1-3 module manual and fixes to consistency issues across all module manuals
B	8/5/2019	- Reformatted to include numbers in headings. - Added register grouping headers in Register Description section and Function Register Map section. - Added the following Pending functionality: - Engineering Scaling Conversion - Background BIT Threshold - Watchdog Timer Capability - Synchronous Sample Clock - Channel Status Enabled - Summary Status - Added Hardware and Processing Block Diagrams
B1	4/2/2020	-Changed Conversion Busy to Floating Point State -Added Firmware Revision and Release Dates for Synchronous Sample Clock Pending feature.
B2	1/21/2021	ECO C08204 - Removed 'Pending' functionality: - Updated Floating Point Scale initialized value to 0.0 in Section 5.6.3 - Updated Background BIT Threshold max value to 2^32 ms in Section 5.7.1 - Updated Floating Point Scale & DAC Value calculations in Appendix A

Table of Contents

1 Introduction

This module manual provides information about the North Atlantic Industries, Inc. (NAI) Digital-to-Analog Function Module: DA2. This module is compatible with all NAI Generation 5 motherboards.

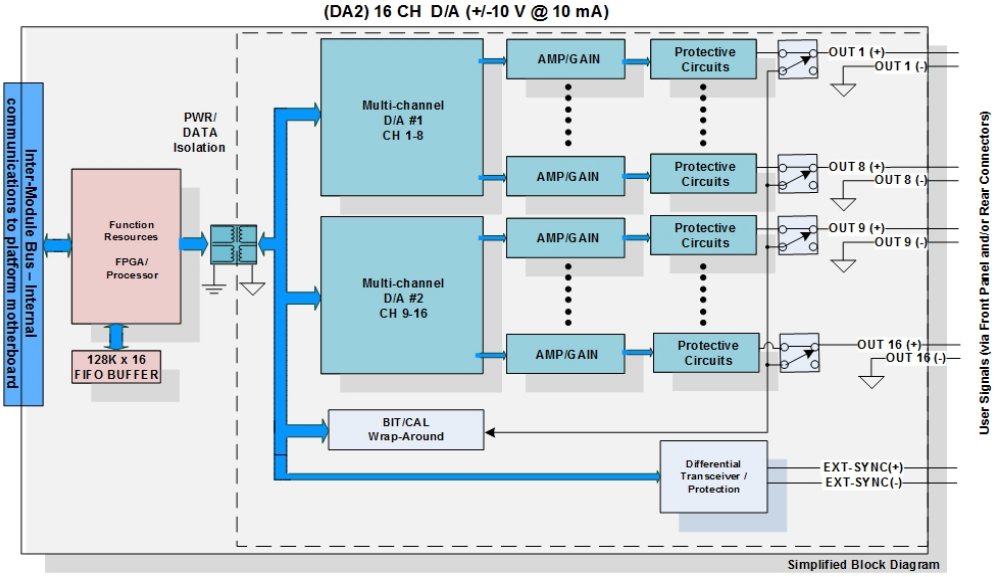
Digital-to-Analog (D/A) module DA2 provides 16 independent D/A output channels with a full-scale range of 0 to ±10 VDC. Linearity/accuracy is ±0.10% FS range over temperature. The DA2 provides a FIFO output buffer, which is programmable for the application.

2 Features

- High-quality D/A conversion, 16-Bit/channel
- Designed to meet the testing requirements of IEC 801-2 Level 2
- Continuous background BIT
- Automatic shutdown protection with the results displayed in a status word
- Extended D/A FIFO buffering capabilities

3 Specifications

3.1 Module DA2 – 16-Channel Digital-to-Analog (±10.0 VDC @ ±10mA)



Resolution:	16-bit/channel for voltage command modes.
Output Format:	Single-ended
Output Range:	±10 VDC, ±5 VDC, ±2.5 VDC, 0 to 10 VDC, 0 to 5 VDC; range programmable for each channel; 10 mA max per channel.
Output and Ground Impedance:	< 1 Ω for each output
System Protection:	Output is set to open circuit at reset or power on
Linearity Error:	±0.10% FS range (FSR) over temperature
Offset Error:	±3 mV
Gain Error:	±0.10% FSR ± 0.02% x (maximum current in load (mA)), per channel Example: The channel is expected to operate @ 2.5 mA maximum, therefore the effective gain error calculates to: (±0.10%) ± (0.02% x 2.5) = ±0.15% FSR
Crosstalk Error (common return):	±0.063% / FSR (V) x (the cumulative maximum Δ current in the other 15 channel loads (mA)) Example: Seven channels @ 2.5 mA, eight channels @ 5 mA, max., @ ±10 V range, so, the error calculates to: (±0.063 / 20) * (±57.5) = ±0.18% maximum (@ FSR of ±10 V, this specific example)
Settling Time:	15 µs typical (25 µs maximum)
Data Buffer:	FIFO, 1 M x 16 channels elements deep. See Operations Manual for functional details.
Load:	Can drive a capacitive load of 0.1 µF, 10 mA/CH max. (Source or Sink). Short circuit protected. When current exceeds 10 mA for I^2T calculation, that channel is set to open circuit and a flag is set.
***Update Rate*:**	40 µs (25 kHz max.) per channel,
ESD Protection:	Designed to meet the testing requirements of IEC 801-2 Level 2. (4k V transient with a peak current of 7.5 A and a time constant of approximately 60 ns).
Power:	5 VDC @ 350 mA typical (est.); ±12 VDC @ 100 mA (est. quiescent) Add 1 mA per 1 mA load per channel in ±12 VDC power supplies.
Ground:	All channel returns (CH-Lo, (-)) are common, but are isolated (250 Vpeak) from system ground.
Weight:	1.5 oz. (42 g)

Specifications are subject to change without notice.

4 Principle of Operation

In addition to the functions and features already described, each module includes extensive background BIT/diagnostics that run in the background in normal operation without user intervention. In addition to output signal read-back (wrap) capabilities, overloaded outputs will be detected with automatic channel shutdown protection, with the results displayed in a status word. The modules also include D/A FIFO Buffering for greater control of the output voltage and signal data. The FIFO D/A buffer will accept, store and output the voltage commands, once enabled and triggered, for applications requiring simulation of waveform generation; single or periodic. The output data command word is formatted as a percentage of the full scale (FS) range selection, which allows maximum resolution and accuracy at lower *voltage range*s.

4.1 Built-In Test (BIT)/Diagnostic Capability

The DA2 module supports three types of built-in tests: Power-On, Continuous Background and Initiated. The results of these tests are logically OR'd together and stored in the *BIT Dynamic Status* and *BIT Latched Status* registers.

4.1.1 Power-On Self-Test (POST) / Power-on BIT (PBIT) / Start-up BIT(SBIT)

The power-on self-test is performed on each channel automatically when power is applied and report the results in the *BIT Status* register when complete. After power-on, the *Power-on BIT Complete* register should be checked to ensure that POST/PBIT/SBIT test is complete before reading the *BIT Dynamic Status* and *BIT Latched Status* registers.

4.1.2 Continuous Background Built-In Test

The background Built-In-Test or Continuous BIT (**CBIT**) *("D2")* runs in the background where each channel is checked to a test accuracy of 0.2% FS. The testing is totally transparent to the user, requires no external programming, and has no effect on the operation of the module or card.

The technique used by the continuous background BIT (**CBIT**) test consists of an "add-2, subtract-1" counting scheme. The BIT counter is incremented by 2 when a BIT-fault is detected and decremented by 1 when there is no BIT fault detected and the BIT counter is greater than 0. When the BIT counter exceeds the (programmed) Background BIT Threshold value, the specific channel's fault bit in the BIT status register will be set. Note, the interval at which BIT is performed is dependent and differs between module types. Rather than specifying the BIT Threshold as a "count", the BIT Threshold is specified as a time in milliseconds. The module will convert the time specified to the BIT Threshold "count" based on the BIT interval for that module. The "add-2, subtract-1" counting scheme effectively filters momentary or intermittent anomalies by allowing them to "come and go" before a BIT fault status or indication is flagged (e.g. BIT faults would register when sustained; i.e. at a ten second interval, not a 10-millisecond interval). This prevents spurious faults from registering valid such as those caused by EMI and/or dirty power causing false BIT faults. Putting more "weight" on errors ("add-2") and less "weight" on subsequent passing results (subtract-1) will result in a BIT failure indication even if a channel "oscillates" between a pass and fail state.

4.1.3 Initiated Built-In Test

The DA2 module support an off-line Initiated Built-in Test (IBIT) *("D3").*

The **IBIT** test uses an internal A/D that measures all D/A channels while they remain connected to the I/O and cycle through 16 signal levels from -FS to +FS. Each channel will be checked to a test accuracy of 0.2% FS. Test cycle is completed within 45 seconds (depending on *update rate*) and results can be read from the Status registers when **IBIT** bit changes from **1** to **0**. This test requires no user programming and can be enabled via the bus.

4.2 D/A FIFO Buffering

The Digital-to-Analog modules include D/A FIFO Buffering for greater control of the output voltage and signal data. The D/A FIFO buffers will accept, store and output the voltage (and/or current) commands, once enabled and triggered, for applications requiring simulation of waveform generation; single or periodic. The output data command word is formatted as a percentage of the full scale (FS) range selection, which allows maximum resolution and accuracy at lower voltage ranges.

4.3 Status and Interrupts

The D/A Module provides registers that indicate faults or events. Refer to “Status and Interrupts Module Manual” for the Principle of Operation description.

4.4 Engineering Scaling Conversions

The D/A Module Data, Voltage and Current Measurement registers can be programmed to be utilized as single precision floating point values (IEEE-754) or as a 32-bit integer value.

When the *Enable Floating Point* Mode register is set to 1 (Floating Point Mode) the following registers are formatted as Single Precision Floating Point Value (IEEE-754):

- *Wrap Voltage (Volts)*
- *Wrap Current (mA)*
- *Internal Voltage (Volts)*
- *DAC Value (Voltage (Volts))**
- *FIFO Buffer Data**

When the Enable Floating Point Mode register is set to 1, it is important that these registers are updated with the Single Precision Floating Point (IEEE-754) representation of the value for proper operation of the channel. Conversely, when the Enable Floating Point Mode register is set to 0, these registers must be updated with the Integer 32-bit representation of the value.

Note, when changing the *Enable Floating Point Mode* from Integer Mode to Floating Point Mode or vice versa, the following step should be followed to avoid faults from falsely being generated because internal registers have an incorrect binary representation of the values:

1. Set the *Enable Floating Point Mode* register to the desired mode (Integer or Floating Point).
2. Wait for the *Floating Point State* register to match the value for the requested Floating Point Mode (Integer = 0, Floating Point = 1); this indicates that the module’s conversion of the register values and internal values is complete. Data registers will be converted to the units specified and can be read in that specified format.
3. Initialize configuration and control registers with the values in the units specified (Integer or Floating Point).

It is very often necessary to relate D/A voltage and current to other engineering units such as PSI (Pounds per Square Inch). When the *Enable Floating Point Mode* register is set to 1, the values entered for the *Floating Point Offset* register and the *Floating-Point Scale* register will be used to convert the D/A data from engineering units to voltage or current values. The purpose of this is to offload the processing that is normally performed by the mission processor to convert the physical quantity to voltage or current values for the *DAC Value* register and the *FIFO Buffer Data* register. When enabled, the module will compute the D/A data as follows:

D/A Value as Volts/Current (Floating Point) =

(D/A Value in Engineering Units (Floating Point) + ***Floating Point Offset****) ** ***Floating Point Scale***

Note:

When *Enable Floating Point* Mode is set to 1 (Floating Point Mode) the listed registers below are formatted as Single Precision Floating Point Value (IEEE-754) and the values specified in the *Floating Point Offset* register and the *Float Point Scale register* ***are*** applied:

- *DAC Value*
- *FIFO Buffer Data – any data left in the FIFO prior to changing the Floating Point Mode will be invalid.*

4.5 Watchdog Timer Capability

The Digital-to-Analog Modules provide support for Watchdog Timer capability. Refer to “Watchdog Timer Module Manual” for the Principle of Operation description.

5 Register Descriptions

The register descriptions provide the register name, Type, Data Range, Read or Write information, Initialized Value, a description of the function and, in most cases, a data table.

5.1 D/A Output Registers

The D/A output is normally in terms of voltage. When the *Enable Floating Point* Mode is enabled, the register value is formatted as a Single Precision Floating Point Value (IEEE-754). In addition, the D/A output value can be specified in engineering units rather than voltage by setting the *Floating Point Scale* and *Floating Point Offset* register values to reflect the conversion algorithm.

5.1.1 DAC Value

Function: Sets the output voltage for the channel.

Type: signed binary word (32-bit) or Single Precision Floating Point Value (IEEE-754) (Floating Point Mode)

Data Range: *DAC value*s are dependent on *Voltage Range* setting for the channel

Enable Floating Point Mode: 0 (Integer Mode)

Unipolar: 0x0000 0000 to 0x0000 FFFF

Bipolar (2's compliment. 16-bit value sign extended to 32 bits): 0xFFFF 8000 to 0x0000 7FFF

Enable Floating Point Mode: 1 (Floating Point Mode)

Single Precision Floating Point Value (IEEE-754)

Read/Write: R/W

Initialized Value: 0

Operational Settings: Refer to section Appendix A – Integer/Floating Point Mode Programming for Integer and Floating Point Mode examples.

5.2 D/A Control Registers

The D/A control registers provide the ability to specify the polarity and *voltage range*, *update rate*, and the enabling or disabling of the D/A outputs. The D/A channels are monitored to detect overcurrent conditions and will automatically disable the D/A output. In the event of an overcurrent condition, the D/A channel needs to be "reset" by writing to the *Overcurrent Reset* register.

5.2.1 Voltage Range

Function: Sets voltage polarity and range for each channel. The value written to the *DAC Value* registers will correlate to the *voltage range* set in this register. *Note*, if the *Enable Floating Point Mode* register is set to **1,** the *Floating Point Scale* register must be set to the **reciprocal of *Voltage Range*** for direct voltage output.

Type: unsigned binary word (32-bit)

Data Range: See table below

Read/Write: R/W

Initialized Value: 0 (Unipolar: 0-5 V)

Operational Settings: Write to the register with a value from the table to select the range. For example, for a ±10V bipolar range write a 0x4 to the register.

Reg Value	*Voltage Range*
0x0	Unipolar: 0 – 5 V
0x1	Unipolar: 0 – 10 V
0x2	Bipolar: ± 2.5 V
0x3	Bipolar: ± 5 V
0x4	Bipolar: ± 10 V

Voltage Range															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	0	0	0	0	0	**D**	**D**	**D**

5.2.2 Output Enable

Function: Enables the voltage to appear on the output.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 FFFF

Read/Write: R/W

Initialized Value: 0 (Channel outputs are disabled)

Operational Settings: Set bit to **1 to** activate the output. Set bit to **0** to disable the output.

Output Enable															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Ch16	**Ch15**	**Ch14**	**Ch13**	**Ch12**	**Ch11**	**Ch10**	**Ch9**	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

5.2.3 Update Rate

Function: Sets the output rate for the DAC output and FIFO Data Output

Type: unsigned binary word (32bit)

Data Range: 0x0000 09C4 to 0x0000 61A8; 400µs (2.5kHz) to 40µs (25kHz).

Read/Write: R/W

Initialized Value: 0x0000 61A8 (40µs) (25kHz)

Operational Settings: This setting is the output rate for each DAC. One *update rate* applies to all channels.

5.2.4 Overcurrent Reset

Function: Resets over loaded channels based on the current value read in the *Wrap Current* register.

Type: unsigned binary word (32-bit)

Data Range: 0 or 1

Read/Write: W

Initialized Value: 0

Operational Settings: Set to **1** to reset over loaded channels. Writing a **1** to this register will re-enable over loaded channels.

Overcurrent Reset															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**D**

5.3 D/A Measurement Registers

The measured voltage and current for the D/A output can be read from the *Wrap Voltage*, *Wrap Current* and *Internal Voltage* registers.

5.3.1 Wrap Voltage

Function: Wrap voltage reading from the channel's output. Used in conjunction with BIT to verify that the output voltage is within range.

Type: signed binary word (32-bit) or Single Precision Floating Point Value (IEEE-754) (Floating Point Mode)

Data Range:

Enable Floating Point Mode: 0 (Integer Mode)

Unipolar: 0x0000 0000 to 0x0003 FFFF; Bipolar: 0x0002 0000 – 0x0001 FFFF

Bipolar (2's compliment. 18-bit value sign extended to 32 bits): 0xFFFE 0000 to 0x0001 7FFF

Enable Floating Point Mode: 1 (Floating Point Mode)

Single Precision Floating Point Value (IEEE-754)

Read/Write: R

Initialized Value: 0

Operational Settings:

Integer Mode: To calculate the LSB subtract the minimum *voltage range* from the maximum *voltage range* then divide by 2^16. For example, if the value in the *Voltage Range* register is range 0-10V then the LSB would have value (10-0)/2^16 = .153 mV. Sign bit = **D17** for bipolar ranges.

Floating Point Mode: Convert the IEEE-754 Single Precision 32-bit value to a floating point value. For example, if the register value is *0x4020 0000*, this is equivalent to 2.5, which represents 2.5V.

Wrap Voltage (Enable Floating Point Mode: Integer Mode)															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	**D**	**D**
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
D	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**

Wrap Voltage (Enable Floating Point Mode: Floating Point Mode)															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
D	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
D	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**

5.3.2 Wrap Current

Function: *Wrap current* reading from the channel's output. Reads current values of D/A outputs being delivered per channel.

Type: signed binary word (32-bit) or Single Precision Floating Point Value (IEEE-754) (Floating Point Mode)

Data Range:

Enable Floating Point Mode: 0 (Integer Mode)

Bipolar (2's compliment. 18-bit value sign extended to 32 bits): 0xFFFE 0000 to 0x0000 7FFF

Enable Floating Point Mode: 1 (Floating Point Mode)

Single Precision Floating Point Value (IEEE-754)

Read/Write: R

Initialized Value: 0

Operational Settings:

Integer Mode: LSB = 305 nA. Sign bit = **D17.**

Floating Point Mode: Convert the IEEE-754 Single Precision 32-bit value to a floating point value.

Wrap Current (Enable Floating Point Mode: Integer Mode)															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	**D**	**D**
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
D	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**

Wrap Current (Enable Floating Point Mode: Floating Point Mode)															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
D	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
D	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**

5.3.3 Internal Voltage

Function: Read the internal voltage value. The interval voltage reading is the voltage before the output switch. If the *Output Enable* register is configured to be disabled, the *Internal Voltage* register will contain the voltage reading that would be outputted.

Data Type: signed binary word (32-bit) or Single Precision Floating Point Value (IEEE-754) (Floating Point Mode)

Data Range:

Enable Floating Point Mode: 0 (Integer Mode)

Unipolar: 0x0000 0000 – 0x0003 FFFF

Bipolar (2's compliment. 18-bit value sign extended to 32 bits): 0xFFFE 0000 to 0x0001 7FFF

Enable Floating Point Mode: 1 (Floating Point Mode)

Single Precision Floating Point Value (IEEE-754)

Read/Write: R

Initialized Value: 0

Operational Settings:

Integer Mode: To calculate the LSB subtract the minimum voltage range from the maximum voltage range then divide by 2^16. Sign bit = **D17** for bipolar ranges.

Floating Point Mode: Read as Single Precision Floating Point Value (IEEE-754) value.

Internal Voltage (Enable Floating Point Mode: Integer Mode)															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	**D**	**D**
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
D	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**

Internal Voltage (Enable Floating Point Mode: Floating Point Mode)															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
D	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
D	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**

5.4 D/A Test Registers

Two different tests, one on-line (**CBIT**) and one off-line (**IBIT**), can be selected.

5.4.1 Test Enabled

Function: Sets bit to enable the associated CBIT ("*D2*") or IBIT ("*D3*"). Note, CBIT cannot be disabled.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 000C

Read/Write: R/W

Initialized Value: 0x4 (CBIT *Test Enabled*)

Operational Settings: BIT tests include an on-line CBIT and an off-line IBIT tests. Failures in the BIT test are reflected in the *BIT Status* registers for the corresponding channels that fail. In addition, an interrupt (if enabled in the *BIT Interrupt Enable* register) can be triggered when the BIT testing detects failures.

Test Enabled															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	0	0	0	0	**IBIT Test D**	**CBIT Test 1**	0	0

5.5 FIFO Registers

The FIFO registers are configurable for each channel.

5.5.1 Data Mode

Function: Sets the data mode of the channel. The output can be based on either the DAC Value register or the RAM Buffer.

Type: unsigned binary word (32-bit**)**

Data Range: 0x0000 0000 to 0x0000 FFFF

Read/Write: R/W

Initialized Value: 0

Operational Settings: Write a **0** to set the data source to the DAC Value Register. Write a **1** to set the data source to the FIFO Buffer.

Data Mode															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Ch16	**Ch15**	**Ch14**	**Ch13**	**Ch12**	**Ch11**	**Ch10**	**Ch9**	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

5.5.2 FIFO Buffer Data

Function: Data in the form of *DAC value*s are written to this register one word at a time (16 bits) and will be outputted to the channel's output once triggered. Buffer will be emptied one value at a time when triggered.

Type: signed binary word (32-bit) or Single Precision Floating Point Value (IEEE-754) (Floating Point Mode)

Data Range:

Enable Floating Point Mode: 0 (Integer Mode)

Unipolar: 0x0000 0000 to 0x0000 FFFF

Bipolar (2's compliment. 16-bit value sign extended to 32 bits): 0xFFFF 8000 to 0x0000 7FFF

Enable Floating Point Mode: 1 (Floating Point Mode)

Single Precision Floating Point Value (IEEE-754)

Read/Write: W

Initialized Value: 0

Operational Settings: Data is held in FIFO until triggered. FIFO size is 1 mega words per channel (each channel has its own buffer). Refer to section Appendix A – Integer/Floating Point Mode Programming for Integer and Floating Point Mode examples.

FIFO Buffer Data (Enable Floating Point Mode: Integer Mode)															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
D	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**

FIFO Buffer Data (Enable Floating Point Mode: Floating Point Mode)															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
D	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
D	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**

5.5.3 FIFO Word Count

Function: Reports the number of words stored in the FIFO buffer.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x000F FFFF

Read/Write: R

Initialized Value: 0 (FIFO is empty)

Operational Settings: Each time a value is written to the FIFO buffer this count is incremented by 1. Once the FIFO is triggered, after each value is outputted to the DAC, this count will be decremented by 1. Watermarks and threshold values can be setup to trigger interrupts when this count crosses user defined values. The maximum number of words that can be stored in the FIFO is 1 mega words.

5.5.4 FIFO Thresholds

The *FIFO Almost Empty*, *FIFO Low Watermark, FIFO High Watermark, and FIFO Almost Full* sets the threshold limits that are used to set the bits in the *FIFO Status* register.

5.5.4.1 FIFO Almost Empty

Function: The *FIFO Almost Empty* is used to set the limits for the “almost empty” status.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x000F FFFF

Read/Write: R/W

Initialized Value: 0x400 (1024)

Operational Settings: When the *FIFO Word Count* register is less than or equal to the value stored in the *FIFO Almost Empty Value* register, the “almost empty” bit **(D1)** of the *FIFO Status* register will be set. When the *Words in FIFO* counter is greater than the value stored in the register, the “almost empty” bit **(D1)** of the *FIFO Status* register will be reset.

5.5.4.2 FIFO Low Watermark

Function: The *FIFO Low Watermark* (low-threshold level) is used to set the limits for the “low watermark” status.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x000F FFFF

Read/Write: R/W

Initialized Value: 0x2000 (8192)

Operational Settings: When the *FIFO Word Count* register is less than or equal to the value stored in the *FIFO Low Watermark Value* register, the “low watermark” bit (**D2**) of the *FIFO Status* register will be set. When the *FIFO Word Count* counter is greater than or equal to the value stored in the register, the “low watermark” bit (**D2**) of the *FIFO Status* register will be reset.

5.5.4.3 FIFO High Watermark

Function: The *FIFO High Watermark* (high-threshold level) is used to set the limits for the “high watermark” status.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x000F FFFF

Read/Write: R/W

Initialized Value: 0x6000 (24576)

Operational Settings: When the *FIFO Word Count* register is greater than or equal to the value stored in the *FIFO High Watermark Value* register, the “high watermark” bit (**D3**) of the *FIFO Status* register will be set. When the *FIFO Word Count* register is less than the value stored in the *FIFO High Watermark Value*, the “high watermark” bit (**D3**) of the *FIFO Status* register will be reset.

5.5.4.4 FIFO Almost Full

Function: The *FIFO Almost Full* is used to set the limits for the "almost full" status.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x000F FFFF

Read/Write: R/W

Initialized Value: 0x7C00 (31744)

Operational Settings: When the *FIFO Word Count* register is greater than or equal to the value stored in the *FIFO Almost Full Value* register, the "almost full" bit (**D4**) of the *FIFO Status* register will be set. When the *Words in FIFO* counter is less than the value stored in the register, the "almost full" bit (**D4**) of the *FIFO Status* register will be reset

5.5.5 Clear FIFO

Function: Clears the FIFO buffer.

Type: unsigned binary word (32-bit)

Data Range: 0 or 1

Read/Write: W

Initialized Value: 0

Operational Settings: Writing a **1** will clear the FIFO buffer and reset the count in the *FIFO Word Count* register.

Clear FIFO															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**D**

5.5.6 Pattern Control

Function: Enable to output all the values written to the *FIFO Buffer* and then repeat.

Note: Pattern Control for the DA2 supports Loop Mode only.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 FFFF

Read/Write: R/W

Initialized Value: 0

Operational Settings: To activate *FIFO Loop Mode* in the *Pattern Control* Register, set the *Data Mode* register to **1**. Then set the bit for the specific channel in the *Pattern Control* register to **1**. Finally, write a **1** to the *Software Trigger* register. The FIFO will output all values written to the *FIFO Data* register and then repeat. To stop looping write a **0** to the *Software Trigger* register.

Pattern Control (FIFO Loop Mode)															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Ch16	**Ch15**	**Ch14**	**Ch13**	**Ch12**	**Ch11**	**Ch10**	**Ch9**	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

5.5.7 Software Trigger

Function: If the memory buffer is enabled writing the trigger value to this register will start the output. Values stored in the FIFO will be output at the set *update rate* until the FIFO is empty.

Type: unsigned binary word (32-bit)

Data Range: 0 or 1

Read/Write: R/W

Initialized Value: 0

Operational Settings: To initiate output from the *FIFO Buffer,* the *Data Mode* register must be set to *FIFO Mode*. Then write a **1** to the *Software Trigger* register to begin outputting data. The **1** will clear once the FIFO empties.

Software Trigger															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**D**

5.6 Engineering Scaling Conversion Registers

The D/A Module Data, Voltage and Current Measurement registers can be programmed to be utilized as an IEEE 754 single-precision floating-point value or as a 32-bit integer value.

5.6.1 Enable Floating Point Mode

Function: Sets all channels for floating point mode or integer module.

Type: unsigned binary word (32-bit)

Data Range: 0 or 1

Read/Write: R/W

Initialized Value: 0 (Integer mode)

Operational Settings: Set bit to **1** to *enable Floating Point* Mode and **0** for Integer Mode.

5.6.2 Floating Point Offset

Function: Single 32-bit register that sets the floating-point offset to add to D/A output.

Type: Single Precision Floating Point Value (IEEE-754)

Data Range: N/A

Read/Write: R/W

Initialized Value: 0.0

Operational Settings: Refer to section Appendix A – Integer/Floating Point Mode Programming for Integer and Floating Point examples.

5.6.3 Floating Point Scale

Function: Single 32-bit register that sets the floating-point scale to multiple to the D/A output.

Type: Single Precision Floating Point Value (IEEE-754)

Data Range: N/A

Read/Write: R/W

Initialized Value: 0.0

Operational Settings: When changing the *Voltage Range,* the *Floating Point Scale* needs to be adjusted in order for the *Wrap Voltage* and *Wrap Current* floating point representation to be scaled correctly.

5.6.4 Floating Point State

Function: Indicates whether the module's internal processing is converting the register values and internal values to the binary representation of the mode selected (Integer or Floating Point).

Type: unsigned binary word (32-bit)

Data Range: 0 to 1

Read/Write: R

Initialized Value: 0

Operational Settings: Indicates the whether the module registers are in Integer (**0**) or Floating Point Mode (**1**). When the *Enable Floating Point Mode* is modified, the application must wait until this register's value matches the requested mode before changing the values of the configuration and control registers with the values in the units specified (Integer or Floating Point).

Floating Point State															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**D**

5.7 Background BIT Threshold Programming Registers

The *Background BIT Threshold* register provides the ability to specify the ***minimum*** time before the BIT fault is reported in the *BIT Status* registers. The *Reset BIT* register provides the ability to reset the BIT counter used in CBIT.

5.7.1 Background BIT Threshold

Function: Sets BIT Threshold value (in milliseconds) to use for all channels for BIT failure indication.

Type: unsigned binary word (32-bit)

Data Range: 1 ms to 2^32 ms

Read/Write: R/W

Initialized Value: 5 (5 ms)

Operational Settings: The interval at which BIT is performed is dependent and differs between module types. Rather than specifying the BIT Threshold as a "count", the BIT Threshold is specified as a time in milliseconds. The module will convert the time specified to the BIT Threshold "count" based on the BIT interval for that module.

5.7.2 Reset BIT

Function: Resets the CBIT internal circuitry and count mechanism. Set the bit corresponding to the channel you want to clear.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 FFFF

Read/Write: W

Initialized Value: 0

Operational Settings: Set bit to **1** for channel to resets the CBIT mechanisms. Bit is self-clearing.

Reset BIT															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Ch16	**Ch15**	**Ch14**	**Ch13**	**Ch12**	**Ch11**	**Ch10**	**Ch9**	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

5.8 Watchdog Timer Registers

Refer to "Watchdog Timer Module Manual" for the Watchdog Timer Register Descriptions.

5.9 Status and Interrupt Registers

The DA2 Module provides status registers for BIT, Overcurrent, External Power Under Voltage, Inter-FPGA Failure, and FIFO.

5.9.1 Channel Status Enabled

Function: Determines whether to update the status for the channels. This feature can be used to "mask" status bits of unused channels in status registers that are bitmapped by channel.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 FFFF (Channel Status)

Read/Write: R/W

Initialized Value: 0x0000 FFFF

Operational Settings: When the bit corresponding to a given channel in the Channel Status Enabled register is not enabled (**0**) the status will be masked and report "0" or "no failure". This applies to all statuses that are bitmapped by channel (BIT Status, Overcurrent Status and Summary Status). Note, Background BIT will continue to run even if the Channel Status Enabled is set to '0'.

Channel Status Enabled															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Ch16	**Ch15**	**Ch14**	**Ch13**	**Ch12**	**Ch11**	**Ch10**	**Ch9**	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

5.9.2 BIT Status

There are four registers associated with the BIT Status: *Dynamic, Latched, Interrupt Enable,* and *Set Edge/Level Interrupt.* The *BIT Status* register will indicate an error when the D/A conversion is outside 0.2% FS accuracy spec.

BIT Dynamic Status															
BIT Latched Status															
BIT Interrupt Enable															
BIT Set Edge/Level Interrupt															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Ch16	**Ch15**	**Ch14**	**Ch13**	**Ch12**	**Ch11**	**Ch10**	**Ch9**	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

Function: Reports the corresponding bit associated with the channel's BIT error.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 FFFF

Read/Write: R (*Dynamic*), R/W (*Latched*, *Interrupt Enable*, *Edge/Level Interrupt*)

Initialized Value: 0

Notes: BIT Status is part of background testing and the status register may be checked or polled at any given time.

5.9.3 Overcurrent Status

There are four registers associated with the Overcurrent Status: *Dynamic, Latched, Interrupt Enable,* and *Set Edge/Level Interrupt*.

Overcurrent Dynamic Status															
Overcurrent Latched Status															
Overcurrent Interrupt Enable															
Overcurrent Set Edge/Level Interrupt															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Ch16	**Ch15**	**Ch14**	**Ch13**	**Ch12**	**Ch11**	**Ch10**	**Ch9**	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

Function: Reports the corresponding bit associated with the channel's Overcurrent error.
Type: unsigned binary word (32-bit)
Data Range: 0x0000 0000 to 0x0000 FFFF
Read/Write: R (*Dynamic*), R/W (*Latched*, *Interrupt Enable*, *Edge/Level Interrupt*)
Initialized Value: 0

5.9.4 External Power Under Voltage Status

There are four registers associated with the External Power Under Voltage Status: *Dynamic, Latched, Interrupt Enable,* and *Set Edge/Level Interrupt*.
D0 = +12V External Power Under Voltage
D1 = -12V External Power Under Voltage

External Power Under Voltage Dynamic Status															
External Power Under Voltage Latched Status															
External Power Under Voltage Interrupt Enable															
External Power Under Voltage Set Edge/Level Interrupt															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	**-12V**	**+12V**

Function: Reports the corresponding bit associated with the channel's External Power Under Voltage error.
Type: unsigned binary word (32-bit)
Data Range: 0x0000 0000 to 0x0000 0003
Read/Write: R (*Dynamic*), R/W (*Latched*, *Interrupt Enable*, *Edge/Level Interrupt*)
Initialized Value: 0

5.9.5 *Inter-FPGA Failure Status/Watchdog Timer Fault*

Data is periodically transferred between the processing module and functional module within the FPGA. A CRC value is calculated and verified with each data transfer. In order to recover from an Inter-FPGA Failure, the module needs to be reset and re-initialized.

There are four registers associated with the Inter-FPGA Status/Watchdog Timer Fault: *Dynamic, Latched, Interrupt Enable,* and *Set Edge/Level Interrupt*.

The lower 16-bits represent the Inter-FPGA Failure Status: **0** = Normal; **0xFFFF** = Inter-FPGA Communication Failure. The status represents the status for all channels on the module.

Bit 31 represents the Watchdog Timer Fault: **0** = Normal; **1** = Watchdog Timer Fault.

Inter-FPGA Failure/Watchdog Timer Fault Dynamic Status															
Inter-FPGA Failure/Watchdog Timer Fault Latched Status															
Inter-FPGA Failure/Watchdog Timer Fault Interrupt Enable															
Inter-FPGA Failure/Watchdog Timer Fault Set Edge/Level Interrupt															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
D	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**	**D**

Function: Sets the corresponding bit associated with the channel's Inter-FPGA Failure and Watchdog Timer Fault error.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x8000 FFFF

Read/Write: R (*Dynamic*), R/W (*Latched*, *Interrupt Enable*, *Edge/Level Interrupt*)

Initialized Value: 0

5.9.6 Summary Status

There are four registers associated with the Summary Status: *Dynamic, Latched, Interrupt Enable,* and *Set Edge/Level Interrupt.*

Summary Status Dynamic Status															
Summary Status Latched Status															
Summary Status Interrupt Enable															
Summary Status Set Edge/Level Interrupt															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Ch16	**Ch15**	**Ch14**	**Ch13**	**Ch12**	**Ch11**	**Ch10**	**Ch9**	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

Function: Sets the corresponding bit when a fault is detected for BIT, Overcurrent, External Power Under Voltage or Inter-FPGA Failure on that channel.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 FFFF

Read/Write: R (*Dynamic*), R/W (*Latched*, *Interrupt Enable*, *Edge/Level Interrupt*)

Initialized Value: 0

Summary Events Table

Module	BIT	Overcurrent	External Power Loss	Open Line	External Power Under Volt	External Power Over Volt	Over Temp	Surge Suppressor Fault
DA1/2/3/4	X	X	X					

5.9.7 FIFO Status

There are four registers associated with the FIFO Status: *Dynamic, Latched, Interrupt Enable,* and *Set Edge/Level Interrupt.* **D0**-**D5** is used to show the different conditions of the buffer.

Description		Configurable?
D0	Empty; 1 when *FIFO Word Count* = 0	No
D1	Almost Empty; 1 when *FIFO Word Count* <= "*FIFO Almost Empty*" register	Yes
D2	Low Watermark; 1 when *FIFO Word Count* <= "*FIFO Low Watermark*" register	Yes
D3	High Watermark; 1 when *FIFO Word Count* >= "*FIFO High Watermark*" register	Yes
D4	Almost Full; 1 when *FIFO Word Count* >= "*FIFO Almost Full*" register	Yes
D5	Full; 1 when *FIFO Word Count* = 1 Mega Words (0x000F FFFF)	No

FIFO Dynamic Status															
FIFO Latched Status															
FIFO Interrupt Enable															
FIFO Set Edge/Level Interrupt															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	0	0	**D**	**D**	**D**	**D**	**D**	**D**

Function: Sets the corresponding bit associated with the FIFO status type; there are separate registers for each channel.

Type: unsigned binary word (32-bit)

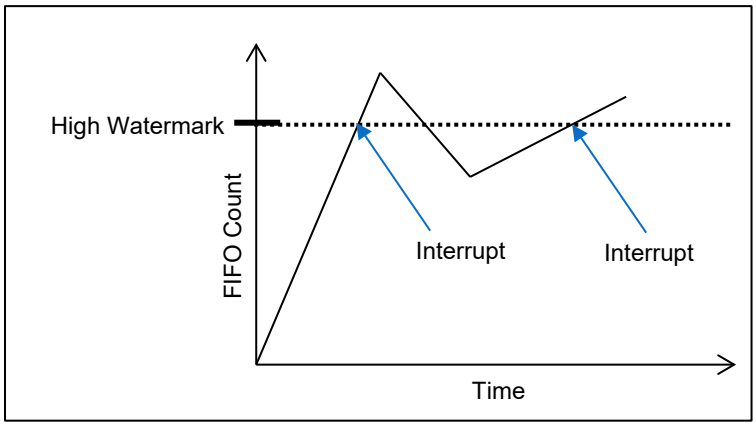
Data Range: 0x0000 0000 to 0x0000 003F

Read/Write: R (*Dynamic*), R/W (*Latched*, *Interrupt Enable*, *Edge/Level Interrupt*)

Initialized Value: 1 (Empty)

Notes:

- Shown below is an example of interrupts generated for the High Watermark. As shown, the interrupt is generated as the *FIFO Word Count* crosses the High Watermark. The interrupt will not be generated a second time until the count goes below the watermark and then above it again.



5.9.8 Interrupt Vector and Steering

When interrupts are enabled, the interrupt vector associated with the specific interrupt can be programmed (typically with a unique number/identifier) such that it can be utilized in the Interrupt Service Routine (ISR) to identify the type of interrupt. When an interrupt occurs, the contents of the Interrupt Vector registers is reported as part of the interrupt mechanism.

In addition to specifying the interrupt vector, the interrupt can be directed ("steered") to the native bus or to the application running on the onboard ARM processor.

Note, the Interrupt Vector and Interrupt Steering registers are mapped to the Motherboard Common Memory and these registers are associated with the Module Slot position (refer to Function Register Map).

5.9.8.1 Interrupt Vector

Function: Set an identifier for the interrupt.

Type: unsigned binary word (32-bit)

Data Range: 0 to 0xFFFF FFFF

Read/Write: R/W

Initialized Value: 0

Operational Settings: When an interrupt occurs, this value is reported as part of the interrupt mechanism.

5.9.8.2 Interrupt Steering

Function: Sets where to direct the interrupt.

Type: unsigned binary word (32-bit)

Data Range: See table

Read/Write: R/W

Initialized Value: 0

Operational Settings: When an interrupt occurs, the interrupt is sent as specified:

Direct Interrupt to VME	1
Direct Interrupt to ARM Processor (via SerDes) *(Custom App on ARM or NAI Ethernet Listener App)*	2
Direct Interrupt to PCIe Bus	5
Direct Interrupt to cPCI Bus	6

6 Function Register Map

Key: ***Bold Italic = Configuration/Control***

Bold Underline = Measurement/Status

*When an event is detected, the bit associated with the event is set in this register and will remain set until the user clears the event bit. Clearing the bit requires writing a 1 back to the specific bit that was set when read (i.e. write-1-to-clear, writing a '1' to a bit set to '1' will set the bit to '0').

** Data is available in Floating Point if *Enable Floating Point Mode* register is set to Floating Point Mode.

~ Data is always in Floating Point.

6.1 D/A Output Registers

0x2004	***DAC Value Ch 1****	R/W
0x2104	***DAC Value Ch 2****	R/W
0x2204	***DAC Value Ch 3****	R/W
0x2304	***DAC Value Ch 4****	R/W
0x2404	***DAC Value Ch 5****	R/W
0x2504	***DAC Value Ch 6****	R/W
0x2604	***DAC Value Ch 7****	R/W
0x2704	***DAC Value Ch 8****	R/W
0x2804	***DAC Value Ch 9****	R/W
0x2904	***DAC Value Ch 10****	R/W
0x2A04	***DAC Value Ch 11****	R/W
0x2B04	***DAC Value Ch 12****	R/W
0x2C04	***DAC Value Ch 13****	R/W
0x2D04	***DAC Value Ch 14****	R/W
0x2E04	***DAC Value Ch 15****	R/W
0x2F04	***DAC Value Ch 16****	R/W

6.2 D/A Control Registers

0x2000	***Voltage Range Ch 1***	R/W
0x2100	***Voltage Range Ch 2***	R/W
0x2200	***Voltage Range Ch 3***	R/W
0x2300	***Voltage Range Ch 4***	R/W
0x2400	***Voltage Range Ch 5***	R/W
0x2500	***Voltage Range Ch 6***	R/W
0x2600	***Voltage Range Ch 7***	R/W
0x2700	***Voltage Range Ch 8***	R/W
0x2800	***Voltage Range Ch 9***	R/W
0x2900	***Voltage Range Ch 10***	R/W
0x2A00	***Voltage Range Ch 11***	R/W
0x2B00	***Voltage Range Ch 12***	R/W
0x2C00	***Voltage Range Ch 13***	R/W
0x2D00	***Voltage Range Ch 14***	R/W
0x2E00	***Voltage Range Ch 15***	R/W
0x2F00	***Voltage Range Ch 16***	R/W

0x100C	***Update Rate Ch 1-16***	R/W
0x1010	***Overcurrent Reset Ch 1-16***	W
0x1014	***Output Enable Ch 1-16***	R/W

6.3 D/A Measurement Registers

0x2008	Wrap Voltage Ch 1**	R
0x2108	Wrap Voltage Ch 2**	R
0x2208	Wrap Voltage Ch 3**	R
0x2308	Wrap Voltage Ch 4**	R
0x2408	Wrap Voltage Ch 5**	R
0x2508	Wrap Voltage Ch 6**	R
0x2608	Wrap Voltage Ch 7**	R
0x2708	Wrap Voltage Ch 8**	R
0x2808	Wrap Voltage Ch 9**	R
0x2908	Wrap Voltage Ch 10**	R
0x2A08	Wrap Voltage Ch 11**	R
0x2B08	Wrap Voltage Ch 12**	R
0x2C08	Wrap Voltage Ch 13**	R
0x2D08	Wrap Voltage Ch 14**	R
0x2E08	Wrap Voltage Ch 15**	R
0x2F08	Wrap Voltage Ch 16**	R

0x200C	Wrap Current Ch 1**	R
0x210C	Wrap Current Ch 2**	R
0x220C	Wrap Current Ch 3**	R
0x230C	Wrap Current Ch 4**	R
0x240C	Wrap Current Ch 5**	R
0x250C	Wrap Current Ch 6**	R
0x260C	Wrap Current Ch 7**	R
0x270C	Wrap Current Ch 8**	R
0x280C	Wrap Current Ch 9**	R
0x290C	Wrap Current Ch 10**	R
0x2A0C	Wrap Current Ch 11**	R
0x2B0C	Wrap Current Ch 12**	R
0x2C0C	Wrap Current Ch 13**	R
0x2D0C	Wrap Current Ch 14**	R
0x2E0C	Wrap Current Ch 15**	R
0x2F0C	Wrap Current Ch 16**	R

0x2044	Internal Voltage Ch 1**	R
0x2144	Internal Voltage Ch 2**	R
0x2244	Internal Voltage Ch 3**	R
0x2344	Internal Voltage Ch 4**	R
0x2444	Internal Voltage Ch 5**	R
0x2544	Internal Voltage Ch 6**	R
0x2644	Internal Voltage Ch 7**	R
0x2744	Internal Voltage Ch 8**	R
0x2844	Internal Voltage Ch 9**	R
0x2944	Internal Voltage Ch 10**	R
0x2A44	Internal Voltage Ch 11**	R
0x2B44	Internal Voltage Ch 12**	R
0x2C44	Internal Voltage Ch 13**	R
0x2D44	Internal Voltage Ch 14**	R
0x2E44	Internal Voltage Ch 15**	R
0x2F44	Internal Voltage Ch 16**	R

6.4 FIFO Registers

0x2018	*FIFO Buffer Data Ch 1***	W
0x2118	*FIFO Buffer Data Ch 2***	W
0x2218	*FIFO Buffer Data Ch 3***	W
0x2318	*FIFO Buffer Data Ch 4***	W
0x2418	*FIFO Buffer Data Ch 5***	W
0x2518	*FIFO Buffer Data Ch 6***	W
0x2618	*FIFO Buffer Data Ch 7***	W
0x2718	*FIFO Buffer Data Ch 8***	W
0x2818	*FIFO Buffer Data Ch 9***	W
0x2918	*FIFO Buffer Data Ch 10***	W
0x2A18	*FIFO Buffer Data Ch 11***	W
0x2B18	*FIFO Buffer Data Ch 12***	W
0x2C18	*FIFO Buffer Data Ch 13***	W
0x2D18	*FIFO Buffer Data Ch 14***	W
0x2E18	*FIFO Buffer Data Ch 15***	W
0x2F18	*FIFO Buffer Data Ch 16***	W

0x201C	FIFO Word Count Ch 1	R
0x211C	FIFO Word Count Ch 2	R
0x221C	FIFO Word Count Ch 3	R
0x231C	FIFO Word Count Ch 4	R
0x241C	FIFO Word Count Ch 5	R
0x251C	FIFO Word Count Ch 6	R
0x261C	FIFO Word Count Ch 7	R
0x271C	FIFO Word Count Ch 8	R
0x281C	FIFO Word Count Ch 9	R
0x291C	FIFO Word Count Ch 10	R
0x2A1C	FIFO Word Count Ch 11	R
0x2B1C	FIFO Word Count Ch 12	R
0x2C1C	FIFO Word Count Ch 13	R
0x2D1C	FIFO Word Count Ch 14	R
0x2E1C	FIFO Word Count Ch 15	R
0x2F1C	FIFO Word Count Ch 16	R

0x2010	*Clear FIFO Ch 1*	W
0x2110	*Clear FIFO Ch 2*	W
0x2210	*Clear FIFO Ch 3*	W
0x2310	*Clear FIFO Ch 4*	W
0x2410	*Clear FIFO Ch 5*	W
0x2510	*Clear FIFO Ch 6*	W
0x2610	*Clear FIFO Ch 7*	W
0x2710	*Clear FIFO Ch 8*	W
0x2810	*Clear FIFO Ch 9*	W
0x2910	*Clear FIFO Ch 10*	W
0x2A10	*Clear FIFO Ch 11*	W
0x2B10	*Clear FIFO Ch 12*	W
0x2C10	*Clear FIFO Ch 13*	W
0x2D10	*Clear FIFO Ch 14*	W
0x2E10	*Clear FIFO Ch 15*	W
0x2F10	*Clear FIFO Ch 16*	W

0x2014	*FIFO Software Trigger Ch 1*	W
0x2114	*FIFO Software Trigger Ch 2*	W
0x2214	*FIFO Software Trigger Ch 3*	W
0x2314	*FIFO Software Trigger Ch 4*	W
0x2414	*FIFO Software Trigger Ch 5*	W
0x2514	*FIFO Software Trigger Ch 6*	W
0x2614	*FIFO Software Trigger Ch 7*	W
0x2714	*FIFO Software Trigger Ch 8*	W
0x2814	*FIFO Software Trigger Ch 9*	W
0x2914	*FIFO Software Trigger Ch 10*	W
0x2A14	*FIFO Software Trigger Ch 11*	W
0x2B14	*FIFO Software Trigger Ch 12*	W
0x2C14	*FIFO Software Trigger Ch 13*	W
0x2D14	*FIFO Software Trigger Ch 14*	W
0x2E14	*FIFO Software Trigger Ch 15*	W
0x2F14	*FIFO Software Trigger Ch 16*	W

0x1004	*Data Mode Ch 1-16*	R/W
0x1008	*Pattern Control Ch 1-16*	R/W

FIFO Thresholds

0x2020	***FIFO Almost Empty Value Ch 1***	R/W
0x2120	***FIFO Almost Empty Value Ch 2***	R/W
0x2220	***FIFO Almost Empty Value Ch 3***	R/W
0x2320	***FIFO Almost Empty Value Ch 4***	R/W
0x2420	***FIFO Almost Empty Value Ch 5***	R/W
0x2520	***FIFO Almost Empty Value Ch 6***	R/W
0x2620	***FIFO Almost Empty Value Ch 7***	R/W
0x2720	***FIFO Almost Empty Value Ch 8***	R/W
0x2820	***FIFO Almost Empty Value Ch 9***	R/W
0x2920	***FIFO Almost Empty Value Ch 10***	R/W
0x2A20	***FIFO Almost Empty Value Ch 11***	R/W
0x2B20	***FIFO Almost Empty Value Ch 12***	R/W
0x2C20	***FIFO Almost Empty Value Ch 13***	R/W
0x2D20	***FIFO Almost Empty Value Ch 14***	R/W
0x2E20	***FIFO Almost Empty Value Ch 15***	R/W
0x2F20	***FIFO Almost Empty Value Ch 16***	R/W

0x2024	***FIFO Low Watermark Value Ch 1***	R/W
0x2124	***FIFO Low Watermark Value Ch 2***	R/W
0x2224	***FIFO Low Watermark Value Ch 3***	R/W
0x2324	***FIFO Low Watermark Value Ch 4***	R/W
0x2424	***FIFO Low Watermark Value Ch 5***	R/W
0x2524	***FIFO Low Watermark Value Ch 6***	R/W
0x2624	***FIFO Low Watermark Value Ch 7***	R/W
0x2724	***FIFO Low Watermark Value Ch 8***	R/W
0x2824	***FIFO Low Watermark Value Ch 9***	R/W
0x2924	***FIFO Low Watermark Value Ch 10***	R/W
0x2A24	***FIFO Low Watermark Value Ch 11***	R/W
0x2B24	***FIFO Low Watermark Value Ch 12***	R/W
0x2C24	***FIFO Low Watermark Value Ch 13***	R/W
0x2D24	***FIFO Low Watermark Value Ch 14***	R/W
0x2E24	***FIFO Low Watermark Value Ch 15***	R/W
0x2F24	***FIFO Low Watermark Value Ch 16***	R/W

0x2028	***FIFO High Watermark Value Ch 1***	R/W
0x2128	***FIFO High Watermark Value Ch 2***	R/W
0x2228	***FIFO High Watermark Value Ch 3***	R/W
0x2328	***FIFO High Watermark Value Ch 4***	R/W
0x2428	***FIFO High Watermark Value Ch 5***	R/W
0x2528	***FIFO High Watermark Value Ch 6***	R/W
0x2628	***FIFO High Watermark Value Ch 7***	R/W
0x2728	***FIFO High Watermark Value Ch 8***	R/W
0x2828	***FIFO High Watermark Value Ch 9***	R/W
0x2928	***FIFO High Watermark Value Ch 10***	R/W
0x2A28	***FIFO High Watermark Value Ch 11***	R/W
0x2B28	***FIFO High Watermark Value Ch 12***	R/W
0x2C28	***FIFO High Watermark Value Ch 13***	R/W
0x2D28	***FIFO High Watermark Value Ch 14***	R/W
0x2E28	***FIFO High Watermark Value Ch 15***	R/W
0x2F28	***FIFO High Watermark Value Ch 16***	R/W

0x202C	***FIFO Almost Full Value Ch 1***	R/W
0x212C	***FIFO Almost Full Value Ch 2***	R/W
0x222C	***FIFO Almost Full Value Ch 3***	R/W
0x232C	***FIFO Almost Full Value Ch 4***	R/W
0x242C	***FIFO Almost Full Value Ch 5***	R/W
0x252C	***FIFO Almost Full Value Ch 6***	R/W
0x262C	***FIFO Almost Full Value Ch 7***	R/W
0x272C	***FIFO Almost Full Value Ch 8***	R/W
0x282C	***FIFO Almost Full Value Ch 9***	R/W
0x292C	***FIFO Almost Full Value Ch 10***	R/W
0x2A2C	***FIFO Almost Full Value Ch 11***	R/W
0x2B2C	***FIFO Almost Full Value Ch 12***	R/W
0x2C2C	***FIFO Almost Full Value Ch 13***	R/W
0x2D2C	***FIFO Almost Full Value Ch 14***	R/W
0x2E2C	***FIFO Almost Full Value Ch 15***	R/W
0x2F2C	***FIFO Almost Full Value Ch 16***	R/W

6.5 Engineering Scaling Conversion Registers

0x02B4	***Enable Floating Point***	R/W
0x0264	**Floating Point State**	R

0x2050	***Floating Point Offset Ch 1~***	R/W
0x2150	***Floating Point Offset Ch 2~***	R/W
0x2250	***Floating Point Offset Ch 3~***	R/W
0x2350	***Floating Point Offset Ch 4~***	R/W
0x2450	***Floating Point Offset Ch 5~***	R/W
0x2550	***Floating Point Offset Ch 6~***	R/W
0x2650	***Floating Point Offset Ch 7~***	R/W
0x2750	***Floating Point Offset Ch 8~***	R/W
0x2850	***Floating Point Offset Ch 9~***	R/W
0x2950	***Floating Point Offset Ch 10~***	R/W
0x2A50	***Floating Point Offset Ch 11~***	R/W
0x2B50	***Floating Point Offset Ch 12~***	R/W
0x2C50	***Floating Point Offset Ch 13~***	R/W
0x2D50	***Floating Point Offset Ch 14~***	R/W
0x2E50	***Floating Point Offset Ch 15~***	R/W
0x2F50	***Floating Point Offset Ch 16~***	R/W

0x2054	***Floating Point Scale Ch 1~***	R/W
0x2154	***Floating Point Scale Ch 2~***	R/W
0x2254	***Floating Point Scale Ch 3~***	R/W
0x2354	***Floating Point Scale Ch 4~***	R/W
0x2454	***Floating Point Scale Ch 5~***	R/W
0x2554	***Floating Point Scale Ch 6~***	R/W
0x2654	***Floating Point Scale Ch 7~***	R/W
0x2754	***Floating Point Scale Ch 8~***	R/W
0x2854	***Floating Point Scale Ch 9~***	R/W
0x2954	***Floating Point Scale Ch 10~***	R/W
0x2A54	***Floating Point Scale Ch 11~***	R/W
0x2B54	***Floating Point Scale Ch 12~***	R/W
0x2C54	***Floating Point Scale Ch 13~***	R/W
0x2D54	***Floating Point Scale Ch 14~***	R/W
0x2E54	***Floating Point Scale Ch 15~***	R/W
0x2F54	***Floating Point Scale Ch 16~***	R/W

6.6 Watchdog Timer Registers

The D/A Modules provide registers that support Watchdog Timer capability. Refer to “Watchdog Timer Module Manual” for the Watchdog Timer Function Register Map.

6.7 D/A Status Registers

0x02B0	***Channel Status Enabled***	R/W

6.7.1 BIT Registers

BIT

0x0800	**Dynamic Status**	R
0x0804	**Latched Status***	R/W
0x0808	***Interrupt Enable***	R/W
0x080C	***Set Edge/Level Interrupt***	R/W

0x0248	***Test Enabled***	R/W

0x02B8	***Background BIT Threshold***	R/W
0x02BC	***Reset BIT***	W

0x02AC	***Power-on BIT Complete***++	R

++After *power-on*, *Power-on* BIT Complete should be checked before reading the BIT *Latched Status*.

6.7.2 Status Registers

Overcurrent

0x0910	**Dynamic Status**	R
0x0914	**Latched Status***	R/W
0x0918	***Interrupt Enable***	R/W
0x091C	***Set Edge/Level Interrupt***	R/W

External Power Under Voltage

0x0930	**Dynamic Status**	R
0x0934	**Latched Status***	R/W
0x0938	***Interrupt Enable***	R/W
0x093C	***Set Edge/Level Interrupt***	R/W

Watchdog Timer Fault/Inter-FPGA Failure

0x09B0	**Dynamic Status**	R
0x09B4	**Latched Status***	R/W
0x09B8	***Interrupt Enable***	R/W
0x09BC	***Set Edge/Level Interrupt***	R/W

Summary

0x09A0	Dynamic Status	R
0x09A4	Latched Status*	R/W
0x09A8	*Interrupt Enable*	R/W
0x09AC	*Set Edge/Level Interrupt*	R/W

FIFO Status

Ch 1	0x0810	Dynamic Status	R	**Ch 9**	0x0890	Dynamic Status	R
	0x0814	Latched Status*	R/W		0x0894	Latched Status*	R/W
	0x0818	*Interrupt Enable*	R/W		0x0898	*Interrupt Enable*	R/W
	0x081C	*Set Edge/Level Interrupt*	R/W		0x089C	*Set Edge/Level Interrupt*	R/W
Ch 2	0x0820	Dynamic Status	R	**Ch 10**	0x08A0	Dynamic Status	R
	0x0824	Latched Status*	R/W		0x08A4	Latched Status*	R/W
	0x0828	*Interrupt Enable*	R/W		0x08A8	*Interrupt Enable*	R/W
	0x082C	*Set Edge/Level Interrupt*	R/W		0x08AC	*Set Edge/Level Interrupt*	R/W
Ch 3	0x0830	Dynamic Status	R	**Ch 11**	0x08B0	Dynamic Status	R
	0x0834	Latched Status*	R/W		0x08B4	Latched Status*	R/W
	0x0838	*Interrupt Enable*	R/W		0x08B8	*Interrupt Enable*	R/W
	0x083C	*Set Edge/Level Interrupt*	R/W		0x08BC	*Set Edge/Level Interrupt*	R/W
Ch 4	0x0840	Dynamic Status	R	**Ch 12**	0x08C0	Dynamic Status	R
	0x0844	Latched Status*	R/W		0x08C4	Latched Status*	R/W
	0x0848	*Interrupt Enable*	R/W		0x08C8	*Interrupt Enable*	R/W
	0x084C	*Set Edge/Level Interrupt*	R/W		0x08CC	*Set Edge/Level Interrupt*	R/W
Ch 5	0x0850	Dynamic Status	R	**Ch 13**	0x08D0	Dynamic Status	R
	0x0854	Latched Status*	R/W		0x08D4	Latched Status*	R/W
	0x0858	*Interrupt Enable*	R/W		0x08D8	*Interrupt Enable*	R/W
	0x085C	*Set Edge/Level Interrupt*	R/W		0x08DC	*Set Edge/Level Interrupt*	R/W
Ch 6	0x0860	Dynamic Status	R	**Ch 14**	0x08E0	Dynamic Status	R
	0x0864	Latched Status*	R/W		0x08E4	Latched Status*	R/W
	0x0868	*Interrupt Enable*	R/W		0x08E8	*Interrupt Enable*	R/W
	0x086C	*Set Edge/Level Interrupt*	R/W		0x08EC	*Set Edge/Level Interrupt*	R/W
Ch 7	0x0870	Dynamic Status	R	**Ch 15**	0x08F0	Dynamic Status	R
	0x0874	Latched Status*	R/W		0x08F4	Latched Status*	R/W
	0x0878	*Interrupt Enable*	R/W		0x08F8	*Interrupt Enable*	R/W
	0x087C	*Set Edge/Level Interrupt*	R/W		0x08FC	*Set Edge/Level Interrupt*	R/W
Ch 8	0x0880	Dynamic Status	R	**Ch 16**	0x0900	Dynamic Status	R
	0x0884	Latched Status*	R/W		0x0904	Latched Status*	R/W
	0x0888	*Interrupt Enable*	R/W		0x0908	*Interrupt Enable*	R/W
	0x088C	*Set Edge/Level Interrupt*	R/W		0x090C	*Set Edge/Level Interrupt*	R/W

6.8 Interrupt Registers

The Interrupt Vector and Interrupt Steering registers are located on the Motherboard Memory Space and do not require any Module Address Offsets. These registers are accessed using the absolute addresses listed in the table below.

0x0500	***Module 1 Interrupt Vector 1 - BIT***	R/W
0x0504	***Module 1 Interrupt Vector 2 - FIFO Ch 1***	R/W
0x0508	***Module 1 Interrupt Vector 3 - FIFO Ch 2***	R/W
0x050C	***Module 1 Interrupt Vector 4 - FIFO Ch 3***	R/W
0x0510	***Module 1 Interrupt Vector 5 - FIFO Ch 4***	R/W
0x0514	***Module 1 Interrupt Vector 6 - FIFO Ch 5***	R/W
0x0518	***Module 1 Interrupt Vector 7 - FIFO Ch 6***	R/W
0x051C	***Module 1 Interrupt Vector 8 - FIFO Ch 7***	R/W
0x0520	***Module 1 Interrupt Vector 9 - FIFO Ch 8***	R/W
0x0524	***Module 1 Interrupt Vector 10 - FIFO Ch 9***	R/W
0x0528	***Module 1 Interrupt Vector 11 - FIFO Ch 10***	R/W
0x052C	***Module 1 Interrupt Vector 12 - FIFO Ch 11***	R/W
0x0530	***Module 1 Interrupt Vector 13 - FIFO Ch 12***	R/W
0x0534	***Module 1 Interrupt Vector 14 - FIFO Ch 13***	R/W
0x0538	***Module 1 Interrupt Vector 15 - FIFO Ch 14***	R/W
0x053C	***Module 1 Interrupt Vector 16 - FIFO Ch 15***	R/W
0x0540	***Module 1 Interrupt Vector 17 - FIFO Ch 16***	R/W
0x0544	***Module 1 Interrupt Vector 18 - Overcurrent***	R/W
0x0548	***Module 1 Interrupt Vector 19 - Reserved***	R/W
0x054C	***Module 1 Interrupt Vector 20 - External Power Under Voltage***	R/W
0x0550 to 0x0564	***Module 1 Interrupt Vector 21-26 - Reserved***	R/W
0x0568	***Module 1 Interrupt Vector 27 - Summary***	R/W
0x056C	***Module 1 Interrupt Vector 28 – Watchdog Timer/Inter-FPGA***	R/W
0x0570 to 0x057C	***Module 1 Interrupt Vector 29-32 - Reserved***	R/W

0x0600	***Module 1 Interrupt Steering 1 - BIT***	R/W
0x0604	***Module 1 Interrupt Steering 2 - FIFO Ch 1***	R/W
0x0608	***Module 1 Interrupt Steering 3 - FIFO Ch 2***	R/W
0x060C	***Module 1 Interrupt Steering 4 - FIFO Ch 3***	R/W
0x0610	***Module 1 Interrupt Steering 5 - FIFO Ch 4***	R/W
0x0614	***Module 1 Interrupt Steering 6 - FIFO Ch 5***	R/W
0x0618	***Module 1 Interrupt Steering 7 - FIFO Ch 6***	R/W
0x061C	***Module 1 Interrupt Steering 8 - FIFO Ch 7***	R/W
0x0620	***Module 1 Interrupt Steering 9 - FIFO Ch 8***	R/W
0x0624	***Module 1 Interrupt Steering 10 - FIFO Ch 9***	R/W
0x0628	***Module 1 Interrupt Steering 11 - FIFO Ch 10***	R/W
0x062C	***Module 1 Interrupt Steering 12 - FIFO Ch 11***	R/W
0x0630	***Module 1 Interrupt Steering 13 - FIFO Ch 12***	R/W
0x0634	***Module 1 Interrupt Steering 14 - FIFO Ch 13***	R/W
0x0638	***Module 1 Interrupt Steering 15 - FIFO Ch 14***	R/W
0x063C	***Module 1 Interrupt Steering 16 - FIFO Ch 15***	R/W
0x0640	***Module 1 Interrupt Steering 17 - FIFO Ch 16***	R/W
0x0644	***Module 1 Interrupt Steering 18 - Overcurrent***	R/W
0x0648	***Module 1 Interrupt Steering 19 - Reserved***	R/W
0x064C	***Module 1 Interrupt Steering 20 - External Power Under Voltage***	R/W
0x0650 to 0x0664	***Module 1 Interrupt Steering 21-26 - Reserved***	R/W
0x0668	***Module 1 Interrupt Steering 27 - Summary***	R/W
0x066C	***Module 1 Interrupt Steering 28 – Watchdog Timer/Inter-FPGA***	R/W
0x0670 to 0x067C	***Module 1 Interrupt Steering 29-32 - Reserved***	R/W

0x0700	*Module 2 Interrupt Vector 1 - BIT*	R/W
0x0704	*Module 2 Interrupt Vector 2 - FIFO Ch 1*	R/W
0x0708	*Module 2 Interrupt Vector 3 - FIFO Ch 2*	R/W
0x070C	*Module 2 Interrupt Vector 4 - FIFO Ch 3*	R/W
0x0710	*Module 2 Interrupt Vector 5 - FIFO Ch 4*	R/W
0x0714	*Module 2 Interrupt Vector 6 - FIFO Ch 5*	R/W
0x0718	*Module 2 Interrupt Vector 7 - FIFO Ch 6*	R/W
0x071C	*Module 2 Interrupt Vector 8 - FIFO Ch 7*	R/W
0x0720	*Module 2 Interrupt Vector 9 - FIFO Ch 8*	R/W
0x0724	*Module 2 Interrupt Vector 10 - FIFO Ch 9*	R/W
0x0728	*Module 2 Interrupt Vector 11 - FIFO Ch 10*	R/W
0x072C	*Module 2 Interrupt Vector 12 - FIFO Ch 11*	R/W
0x0730	*Module 2 Interrupt Vector 13 - FIFO Ch 12*	R/W
0x0734	*Module 2 Interrupt Vector 14 - FIFO Ch 13*	R/W
0x0738	*Module 2 Interrupt Vector 15 - FIFO Ch 14*	R/W
0x073C	*Module 2 Interrupt Vector 16 - FIFO Ch 15*	R/W
0x0740	*Module 2 Interrupt Vector 17 - FIFO Ch 16*	R/W
0x0744	*Module 2 Interrupt Vector 18 - Overcurrent*	R/W
0x0748	*Module 2 Interrupt Vector 19 - Reserved*	R/W
0x074C	*Module 2 Interrupt Vector 20 - External Power Under Voltage*	R/W
0x0750 to 0x0764	*Module 2 Interrupt Vector 21-26 - Reserved*	R/W
0x0768	*Module 2 Interrupt Vector 27 - Summary*	R/W
0x076C	*Module 2 Interrupt Vector 28 – Watchdog Timer/Inter-FPGA*	R/W
0x0770 to 0x077C	*Module 2 Interrupt Vector 29-32 - Reserved*	R/W

0x0800	*Module 2 Interrupt Steering 1 - BIT*	R/W
0x0804	*Module 2 Interrupt Steering 2 - FIFO Ch 1*	R/W
0x0808	*Module 2 Interrupt Steering 3 - FIFO Ch 2*	R/W
0x080C	*Module 2 Interrupt Steering 4 - FIFO Ch 3*	R/W
0x0810	*Module 2 Interrupt Steering 5 - FIFO Ch 4*	R/W
0x0814	*Module 2 Interrupt Steering 6 - FIFO Ch 5*	R/W
0x0818	*Module 2 Interrupt Steering 7 - FIFO Ch 6*	R/W
0x081C	*Module 2 Interrupt Steering 8 - FIFO Ch 7*	R/W
0x0820	*Module 2 Interrupt Steering 9 - FIFO Ch 8*	R/W
0x0824	*Module 2 Interrupt Steering 10 - FIFO Ch 9*	R/W
0x0828	*Module 2 Interrupt Steering 11 - FIFO Ch 10*	R/W
0x082C	*Module 2 Interrupt Steering 12 - FIFO Ch 11*	R/W
0x0830	*Module 2 Interrupt Steering 13 - FIFO Ch 12*	R/W
0x0834	*Module 2 Interrupt Steering 14 - FIFO Ch 13*	R/W
0x0838	*Module 2 Interrupt Steering 15 - FIFO Ch 14*	R/W
0x083C	*Module 2 Interrupt Steering 16 - FIFO Ch 15*	R/W
0x0840	*Module 2 Interrupt Steering 17 - FIFO Ch 16*	R/W
0x0844	*Module 2 Interrupt Steering 18 - Overcurrent*	R/W
0x0848	*Module 2 Interrupt Steering 19 - Reserved*	R/W
0x084C	*Module 2 Interrupt Steering 20 – External Power Under Voltage*	R/W
0x0850 to 0x0864	*Module 2 Interrupt Steering 21-26 - Reserved*	R/W
0x0868	*Module 2 Interrupt Steering 27 - Summary*	R/W
0x086C	*Module 2 Interrupt Steering 28 – Watchdog Timer/Inter-FPGA*	R/W
0x0870 to 0x087C	*Module 2 Interrupt Steering 29-32 - Reserved*	R/W

0x0900	***Module 3 Interrupt Vector 1 - BIT***	R/W
0x0904	***Module 3 Interrupt Vector 2 - FIFO Ch 1***	R/W
0x0908	***Module 3 Interrupt Vector 3 - FIFO Ch 2***	R/W
0x090C	***Module 3 Interrupt Vector 4 - FIFO Ch 3***	R/W
0x0910	***Module 3 Interrupt Vector 5 - FIFO Ch 4***	R/W
0x0914	***Module 3 Interrupt Vector 6 - FIFO Ch 5***	R/W
0x0918	***Module 3 Interrupt Vector 7 - FIFO Ch 6***	R/W
0x091C	***Module 3 Interrupt Vector 8 - FIFO Ch 7***	R/W
0x0920	***Module 3 Interrupt Vector 9 - FIFO Ch 8***	R/W
0x0924	***Module 3 Interrupt Vector 10 - FIFO Ch 9***	R/W
0x0928	***Module 3 Interrupt Vector 11 - FIFO Ch 10***	R/W
0x092C	***Module 3 Interrupt Vector 12 - FIFO Ch 11***	R/W
0x0930	***Module 3 Interrupt Vector 13 - FIFO Ch 12***	R/W
0x0934	***Module 3 Interrupt Vector 14 - FIFO Ch 13***	R/W
0x0938	***Module 3 Interrupt Vector 15 - FIFO Ch 14***	R/W
0x093C	***Module 3 Interrupt Vector 16 - FIFO Ch 15***	R/W
0x0940	***Module 3 Interrupt Vector 17 - FIFO Ch 16***	R/W
0x0944	***Module 3 Interrupt Vector 18 - Overcurrent***	R/W
0x0948	***Module 3 Interrupt Vector 19 - Reserved***	R/W
0x094C	***Module 3 Interrupt Vector 20 – External Power Under Voltage***	R/W
0x0950 to 0x0964	***Module 3 Interrupt Vector 21-26 - Reserved***	R/W
0x0968	***Module 3 Interrupt Vector 27 - Summary***	R/W
0x096C	***Module 3 Interrupt Vector 28 – Watchdog Timer/Inter-FPGA***	R/W
0x0970 to 0x097C	***Module 3 Interrupt Vector 29-32 - Reserved***	R/W

0x0A00	***Module 3 Interrupt Steering 1 - BIT***	R/W
0x0A04	***Module 3 Interrupt Steering 2 - FIFO Ch 1***	R/W
0x0A08	***Module 3 Interrupt Steering 3 - FIFO Ch 2***	R/W
0x0A0C	***Module 3 Interrupt Steering 4 - FIFO Ch 3***	R/W
0x0A10	***Module 3 Interrupt Steering 5 - FIFO Ch 4***	R/W
0x0A14	***Module 3 Interrupt Steering 6 - FIFO Ch 5***	R/W
0x0A18	***Module 3 Interrupt Steering 7 - FIFO Ch 6***	R/W
0x0A1C	***Module 3 Interrupt Steering 8 - FIFO Ch 7***	R/W
0x0A20	***Module 3 Interrupt Steering 9 - FIFO Ch 8***	R/W
0x0A24	***Module 3 Interrupt Steering 10 - FIFO Ch 9***	R/W
0x0A28	***Module 3 Interrupt Steering 11 - FIFO Ch 10***	R/W
0x0A2C	***Module 3 Interrupt Steering 12 - FIFO Ch 11***	R/W
0x0A30	***Module 3 Interrupt Steering 13 - FIFO Ch 12***	R/W
0x0A34	***Module 3 Interrupt Steering 14 - FIFO Ch 13***	R/W
0x0A38	***Module 3 Interrupt Steering 15 - FIFO Ch 14***	R/W
0x0A3C	***Module 3 Interrupt Steering 16 - FIFO Ch 15***	R/W
0x0A40	***Module 3 Interrupt Steering 17 - FIFO Ch 16***	R/W
0x0A44	***Module 3 Interrupt Steering 18 - Overcurrent***	R/W
0x0A48	***Module 3 Interrupt Steering 19 - Reserved***	R/W
0x0A4C	***Module 3 Interrupt Steering 20 – External Power Under Voltage***	R/W
0x0A50 to 0x0A64	***Module 3 Interrupt Steering 21-26 - Reserved***	R/W
0x0A68	***Module 3 Interrupt Steering 27 - Summary***	R/W
0x0A6C	***Module 3 Interrupt Steering 28 – Watchdog Timer/Inter-FPGA***	R/W
0x0A70 to 0x0A7C	***Module 3 Interrupt Steering 29-32 - Reserved***	R/W

0x0B00	***Module 4 Interrupt Vector 1 - BIT***	R/W
0x0B04	***Module 4 Interrupt Vector 2 - FIFO Ch 1***	R/W
0x0B08	***Module 4 Interrupt Vector 3 - FIFO Ch 2***	R/W
0x0B0C	***Module 4 Interrupt Vector 4 - FIFO Ch 3***	R/W
0x0B10	***Module 4 Interrupt Vector 5 - FIFO Ch 4***	R/W
0x0B14	***Module 4 Interrupt Vector 6 - FIFO Ch 5***	R/W
0x0B18	***Module 4 Interrupt Vector 7 - FIFO Ch 6***	R/W
0x0B1C	***Module 4 Interrupt Vector 8 - FIFO Ch 7***	R/W
0x0B20	***Module 4 Interrupt Vector 9 - FIFO Ch 8***	R/W
0x0B24	***Module 4 Interrupt Vector 10 - FIFO Ch 9***	R/W
0x0B28	***Module 4 Interrupt Vector 11 - FIFO Ch 10***	R/W
0x0B2C	***Module 4 Interrupt Vector 12 - FIFO Ch 11***	R/W
0x0B30	***Module 4 Interrupt Vector 13 - FIFO Ch 12***	R/W
0x0B34	***Module 4 Interrupt Vector 14 - FIFO Ch 13***	R/W
0x0B38	***Module 4 Interrupt Vector 15 - FIFO Ch 14***	R/W
0x0B3C	***Module 4 Interrupt Vector 16 - FIFO Ch 15***	R/W
0x0B40	***Module 4 Interrupt Vector 17 - FIFO Ch 16***	R/W
0x0B44	***Module 4 Interrupt Vector 18 - Overcurrent***	R/W
0x0B48	***Module 4 Interrupt Vector 19 - Reserved***	R/W
0x0B4C	***Module 4 Interrupt Vector 20 – External Power Under Voltage***	R/W
0x0B50 to 0x0B64	***Module 4 Interrupt Vector 21-26 - Reserved***	R/W
0x0B68	***Module 4 Interrupt Vector 27 - Summary***	R/W
0x0B6C	***Module 4 Interrupt Vector 28 – Watchdog Timer/Inter-FPGA***	R/W
0x0B70 to 0x0B7C	***Module 4 Interrupt Vector 29-32 - Reserved***	R/W

0x0C00	***Module 4 Interrupt Steering 1 - BIT***	R/W
0x0C04	***Module 4 Interrupt Steering 2 - FIFO Ch 1***	R/W
0x0C08	***Module 4 Interrupt Steering 3 - FIFO Ch 2***	R/W
0x0C0C	***Module 4 Interrupt Steering 4 - FIFO Ch 3***	R/W
0x0C10	***Module 4 Interrupt Steering 5 - FIFO Ch 4***	R/W
0x0C14	***Module 4 Interrupt Steering 6 - FIFO Ch 5***	R/W
0x0C18	***Module 4 Interrupt Steering 7 - FIFO Ch 6***	R/W
0x0C1C	***Module 4 Interrupt Steering 8 - FIFO Ch 7***	R/W
0x0C20	***Module 4 Interrupt Steering 9 - FIFO Ch 8***	R/W
0x0C24	***Module 4 Interrupt Steering 10 - FIFO Ch 9***	R/W
0x0C28	***Module 4 Interrupt Steering 11 - FIFO Ch 10***	R/W
0x0C2C	***Module 4 Interrupt Steering 12 - FIFO Ch 11***	R/W
0x0C30	***Module 4 Interrupt Steering 13 - FIFO Ch 12***	R/W
0x0C34	***Module 4 Interrupt Steering 14 - FIFO Ch 13***	R/W
0x0C38	***Module 4 Interrupt Steering 15 - FIFO Ch 14***	R/W
0x0C3C	***Module 4 Interrupt Steering 16 - FIFO Ch 15***	R/W
0x0C40	***Module 4 Interrupt Steering 17 - FIFO Ch 16***	R/W
0x0C44	***Module 4 Interrupt Steering 18 - Overcurrent***	R/W
0x0C48	***Module 4 Interrupt Steering 19 - Reserved***	R/W
0x0C4C	***Module 4 Interrupt Steering 20 – External Power Under Voltage***	R/W
0x0C50 to 0x0C64	***Module 4 Interrupt Steering 21-26 - Reserved***	R/W
0x0C68	***Module 4 Interrupt Steering 27 - Summary***	R/W
0x0C6C	***Module 4 Interrupt Steering 28 – Watchdog Timer/Inter-FPGA***	R/W
0x0C70 to 0x0C7C	***Module 4 Interrupt Steering 29-32 - Reserved***	R/W

0x0D00	***Module 5 Interrupt Vector 1 - BIT***	R/W
0x0D04	***Module 5 Interrupt Vector 2 - FIFO Ch 1***	R/W
0x0D08	***Module 5 Interrupt Vector 3 - FIFO Ch 2***	R/W
0x0D0C	***Module 5 Interrupt Vector 4 - FIFO Ch 3***	R/W
0x0D10	***Module 5 Interrupt Vector 5 - FIFO Ch 4***	R/W
0x0D14	***Module 5 Interrupt Vector 6 - FIFO Ch 5***	R/W
0x0D18	***Module 5 Interrupt Vector 7 - FIFO Ch 6***	R/W
0x0D1C	***Module 5 Interrupt Vector 8 - FIFO Ch 7***	R/W
0x0D20	***Module 5 Interrupt Vector 9 - FIFO Ch 8***	R/W
0x0D24	***Module 5 Interrupt Vector 10 - FIFO Ch 9***	R/W
0x0D28	***Module 5 Interrupt Vector 11 - FIFO Ch 10***	R/W
0x0D2C	***Module 5 Interrupt Vector 12 - FIFO Ch 11***	R/W
0x0D30	***Module 5 Interrupt Vector 13 - FIFO Ch 12***	R/W
0x0D34	***Module 5 Interrupt Vector 14 - FIFO Ch 13***	R/W
0x0D38	***Module 5 Interrupt Vector 15 - FIFO Ch 14***	R/W
0x0D3C	***Module 5 Interrupt Vector 16 - FIFO Ch 15***	R/W
0x0D40	***Module 5 Interrupt Vector 17 - FIFO Ch 16***	R/W
0x0D44	***Module 5 Interrupt Vector 18 - Overcurrent***	R/W
0x0D48	***Module 5 Interrupt Vector 19 - Reserved***	R/W
0x0D4C	***Module 5 Interrupt Vector 20 – External Power Under Voltage***	R/W
0x0D50 to 0x0D64	***Module 5 Interrupt Vector 21-26 - Reserved***	R/W
0x0D68	***Module 5 Interrupt Vector 27 - Summary***	R/W
0x0D6C	***Module 5 Interrupt Vector 28 – Watchdog Timer/Inter-FPGA***	R/W
0x0D70 to 0x0D7C	***Module 5 Interrupt Vector 29-32 - Reserved***	R/W

0x0E00	***Module 5 Interrupt Steering 1 - BIT***	R/W
0x0E04	***Module 5 Interrupt Steering 2 - FIFO Ch 1***	R/W
0x0E08	***Module 5 Interrupt Steering 3 - FIFO Ch 2***	R/W
0x0E0C	***Module 5 Interrupt Steering 4 - FIFO Ch 3***	R/W
0x0E10	***Module 5 Interrupt Steering 5 - FIFO Ch 4***	R/W
0x0E14	***Module 5 Interrupt Steering 6 - FIFO Ch 5***	R/W
0x0E18	***Module 5 Interrupt Steering 7 - FIFO Ch 6***	R/W
0x0E1C	***Module 5 Interrupt Steering 8 - FIFO Ch 7***	R/W
0x0E20	***Module 5 Interrupt Steering 9 - FIFO Ch 8***	R/W
0x0E24	***Module 5 Interrupt Steering 10 - FIFO Ch 9***	R/W
0x0E28	***Module 5 Interrupt Steering 11 - FIFO Ch 10***	R/W
0x0E2C	***Module 5 Interrupt Steering 12 - FIFO Ch 11***	R/W
0x0E30	***Module 5 Interrupt Steering 13 - FIFO Ch 12***	R/W
0x0E34	***Module 5 Interrupt Steering 14 - FIFO Ch 13***	R/W
0x0E38	***Module 5 Interrupt Steering 15 - FIFO Ch 14***	R/W
0x0E3C	***Module 5 Interrupt Steering 16 - FIFO Ch 15***	R/W
0x0E40	***Module 5 Interrupt Steering 17 - FIFO Ch 16***	R/W
0x0E44	***Module 5 Interrupt Steering 18 - Overcurrent***	R/W
0x0E48	***Module 5 Interrupt Steering 19 - Reserved***	R/W
0x0E4C	***Module 5 Interrupt Steering 20 – External Power Under Voltage***	R/W
0x0E50 to 0x0E64	***Module 5 Interrupt Steering 21-26 - Reserved***	R/W
0x0E68	***Module 5 Interrupt Steering 27 - Summary***	R/W
0x0E6C	***Module 5 Interrupt Steering 28 – Watchdog Timer/Inter-FPGA***	R/W
0x0E70 to 0x0E7C	***Module 5 Interrupt Steering 29-32 - Reserved***	R/W

0x0F00	***Module 6 Interrupt Vector 1 - BIT***	R/W
0x0F04	***Module 6 Interrupt Vector 2 - FIFO Ch 1***	R/W
0x0F08	***Module 6 Interrupt Vector 3 - FIFO Ch 2***	R/W
0x0F0C	***Module 6 Interrupt Vector 4 - FIFO Ch 3***	R/W
0x0F10	***Module 6 Interrupt Vector 5 - FIFO Ch 4***	R/W
0x0F14	***Module 6 Interrupt Vector 6 - FIFO Ch 5***	R/W
0x0F18	***Module 6 Interrupt Vector 7 - FIFO Ch 6***	R/W
0x0F1C	***Module 6 Interrupt Vector 8 - FIFO Ch 7***	R/W
0x0F20	***Module 6 Interrupt Vector 9 - FIFO Ch 8***	R/W
0x0F24	***Module 6 Interrupt Vector 10 - FIFO Ch 9***	R/W
0x0F28	***Module 6 Interrupt Vector 11 - FIFO Ch 10***	R/W
0x0F2C	***Module 6 Interrupt Vector 12 - FIFO Ch 11***	R/W
0x0F30	***Module 6 Interrupt Vector 13 - FIFO Ch 12***	R/W
0x0F34	***Module 6 Interrupt Vector 14 - FIFO Ch 13***	R/W
0x0F38	***Module 6 Interrupt Vector 15 - FIFO Ch 14***	R/W
0x0F3C	***Module 6 Interrupt Vector 16 - FIFO Ch 15***	R/W
0x0F40	***Module 6 Interrupt Vector 17 - FIFO Ch 16***	R/W
0x0F44	***Module 6 Interrupt Vector 18 - Overcurrent***	R/W
0x0F48	***Module 6 Interrupt Vector 19 - Reserved***	R/W
0x0F4C	***Module 6 Interrupt Vector 20 – External Power Under Voltage***	R/W
0x0F50 to 0x0F64	***Module 6 Interrupt Vector 21-26 - Reserved***	R/W
0x0F68	***Module 6 Interrupt Vector 27 - Summary***	R/W
0x0F6C	***Module 6 Interrupt Vector 28 – Watchdog Timer/Inter-FPGA***	R/W
0x0F70 to 0x0F7C	***Module 6 Interrupt Vector 29-32 - Reserved***	R/W

0x1000	***Module 6 Interrupt Steering 1 - BIT***	R/W
0x1004	***Module 6 Interrupt Steering 2 - FIFO Ch 1***	R/W
0x1008	***Module 6 Interrupt Steering 3 - FIFO Ch 2***	R/W
0x100C	***Module 6 Interrupt Steering 4 - FIFO Ch 3***	R/W
0x1010	***Module 6 Interrupt Steering 5 - FIFO Ch 4***	R/W
0x1014	***Module 6 Interrupt Steering 6 - FIFO Ch 5***	R/W
0x1018	***Module 6 Interrupt Steering 7 - FIFO Ch 6***	R/W
0x101C	***Module 6 Interrupt Steering 8 - FIFO Ch 7***	R/W
0x1020	***Module 6 Interrupt Steering 9 - FIFO Ch 8***	R/W
0x1024	***Module 6 Interrupt Steering 10 - FIFO Ch 9***	R/W
0x1028	***Module 6 Interrupt Steering 11 - FIFO Ch 10***	R/W
0x102C	***Module 6 Interrupt Steering 12 - FIFO Ch 11***	R/W
0x1030	***Module 6 Interrupt Steering 13 - FIFO Ch 12***	R/W
0x1034	***Module 6 Interrupt Steering 14 - FIFO Ch 13***	R/W
0x1038	***Module 6 Interrupt Steering 15 - FIFO Ch 14***	R/W
0x103C	***Module 6 Interrupt Steering 16 - FIFO Ch 15***	R/W
0x1040	***Module 6 Interrupt Steering 17 - FIFO Ch 16***	R/W
0x1044	***Module 6 Interrupt Steering 18 - Overcurrent***	R/W
0x1048	***Module 6 Interrupt Steering 19 - Reserved***	R/W
0x104C	***Module 6 Interrupt Steering 20 – External Power Under Voltage***	R/W
0x1050 to 0x1064	***Module 6 Interrupt Steering 21-26 - Reserved***	R/W
0x1068	***Module 6 Interrupt Steering 27 - Summary***	R/W
0x106C	***Module 6 Interrupt Steering 28 – Watchdog Timer/Inter-FPGA***	R/W
0x1070 to 0x107C	***Module 6 Interrupt Steering 29-32 - Reserved***	R/W

7 Appendix A – Integer/Floating Point Mode Programming

7.1 Integer Mode Programming

The following registers should be configured as follows:

Register	Value	Description
Voltage Range	0x4	±10 volts
Enable Floating Point	0	Disable for Floating Point Mode

Note: LSB for Bipolar ±10-volt range:
LSB = 10/0x00007FFF
= 10/32767=305uV

DAC Value (Integer)	DAC Voltage Output
1.000 of FS = 0x0000 7FFF	32767 * LSB = 10.0 volts
0.5 of FS = 0x0000 4000	16384 * LSB = 5.0 volts
0.0 of FS = 0x0000 0000	0 * LSB = 0.0 volts
-0.5 of FS = 0xFFFF C000	-16384 * LSB = -5.0 volts
-1.0 of FS = 0xFFFF 8000	-32768 * LSB = -10. volts

7.2 Floating Point Mode Voltage Programming

The following registers should be configured as follows:

Register	Value	Description
Voltage Range	0x4	±10 volts
Enable Floating Point	1	Enable for Floating Point Mode
Floating Point Scale	0.1	Scale = 1 / (Full Range) = 1 / 10.0 = 0.1
Floating Point Offset	0.0	No Offset

Note: LSB for Bipolar ±10-volt range:
LSB = 10/0x00007FFF
= 10/32767=305uV

DAC Value (volts) (Floating Point)	DAC Value (Calculated by Module)	DAC Value (Integer)	DAC Voltage Output
10.0	(10.0 + 0.0) *0.1 = 1 (FS)	FS = 0x0000 7FFF	32767 * LSB = 10.0 volts
5.0	(5.0 + 0.0) *0.1 = 0.5 of FS	0.5 of FS = 0x0000 4000	16384 * LSB = 5.0 volts
0.0	(0.0 + 0.0) *0.1 = 0.0 of FS	0.0 of FS = 0x0000 0000	0 * LSB = 0.0 volts
-5.0	(-5.0 + 0.0) *0.1 = -0.5 of FS	-0.5 of FS = 0xFFFF C000	-16384 * LSB = -5.0 volts
-10.0	(-10.0 + 0.0) *0.1 = -1 (-FS)	-FS = 0xFFFF 8000	-32768 * LSB = -10. volts

7.3 Floating Point Mode Engineering Units Programming

Example #1:

An application wants to associate -10 to 10 volts to -5 to 5 inches.

The following registers should be configured as follows:

Register	Value	Description
Voltage Range	0x4	±10 volts
Enable Floating Point	1	Enable for Floating Point Mode
Floating Point Scale	0.2	Scale = 1 / inches range = 1 / 5 = 0.2
Floating Point Offset	0.0	No Offset

Note: LSB for Bipolar ±10-volt range:

LSB = 10/0x00007FFF

= 10/32767=305uV

DAC Value (in) (Floating Point)	DAC Value (Calculated by Module)	DAC Value (Integer)	DAC Voltage Output
5.0	(5.0 + 0.0) * 0.2 = 1 (FS)	FS = 0x0000 7FFF	32767 * LSB = 10.0 volts
2.5	(2.5 + 0.0) * 0.2 = .5 of FS	0.5 of FS = 0x0000 4000	16384 * LSB = 5.0 volts
0.0	(0.0 + 0.0) * 0.2 = 0	0.0 of FS = 0x0000 0000	0 * LSB = 0.0 volts
-2.5	(-2.5 + 0.0) * 0.2 = -.5 of FS	-0.5 of FS = 0xFFFF C000	-16384 * LSB = -5.0 volts
-5.0	(-5.0 + 0.0) * 0.2 = -1 (-FS)	-FS = 0xFFFF 8000	-32768 * LSB = -10. volts

Example #2:

An application wants to associate 0 to 10 volts to 0 to 50 feet with a bias of 0.5 feet (in other words 0.5 feet is equivalent to 0 volts).

The following registers should be configured as follows:

Register	Value	Description
Voltage Range	0x1	Unipolar 0-10 volts
Enable Floating Point	1	Enable for Floating Point Mode
Floating Point Scale	0.02	Scale = 1 / feet range = 1 / 50 = 0.02
Floating Point Offset	-0.50	Bias (0.5 feet) that is equivalent to 0 volts

The following are sample outputs:

Note: LSB for Unipolar 10-volt range:

LSB = 10/0x0000FFFF

= 10/65535

DAC Value (ft) (Floating Point)	DAC Value (Calculated by Module)	DAC Value (Integer)	DAC Voltage Output
50.00	(50.0 - 0.50) * 0.02 = 0.99 of FS	0.99 of FS = 0x0000 FD70	64880 * LSB = 9.90 volts
25.00	(25.00 - 0.50) * 0.02 = 0.49 of FS	0.49 of FS = 0x0000 7D70	32112 * LSB = 4.90 volts
5.50	(5.50 - 0.50) * 0.02 = 0.10 of FS	0.10 of FS = 0x0000 199A	6554 * LSB = 1.00 volts
0.50	(0.50 - 0.50) * 0.02 = 0.00 of FS	0.00 of FS = 0x0000 0000	0 * LSB = 0.0 volts

8 Appendix B – Register Name Changes From Previous Releases

This section provides a mapping of the register names used in this document against register names used in previous releases.

Rev B - Register Names	Rev A - Register Names
D/A Output Registers	
DAC Value	DAC Value
D/A Control Registers	
Voltage Range	Voltage Range
Output Enable	Enable Output
Update Rate	Update Rate
Overcurrent Reset	Overcurrent Reset
D/A Measurement Registers	
Wrap Voltage	Wrap Voltage
Wrap Current	Wrap Current
Internal Voltage	Internal Voltage
D/A Test Registers	
Test Enabled	Test Enable
FIFO Registers	
Data Mode	Use FIFO
FIFO Buffer Data	FIFO Buffer Data
FIFO Word Count	FIFO Word Count
FIFO Almost Empty	FIFO Empty Mark
FIFO Low Watermark	FIFO Low Mark
FIFO High Watermark	FIFO High Mark
FIFO Almost Full	FIFO Full Mark
Clear FIFO	FIFO Reset
Pattern Control	FIFO Loop Mode
Software Trigger	Software Trigger
Engineering Scaling Conversion Registers	
Enable Floating Point Mode	
Floating Point Offset	
Floating Point Scale	
Floating Point State	
Background BIT Threshold Programming Registers	
Background BIT Threshold	
Reset BIT	
Status and Interrupt Registers	
Channel Status Enabled	
BIT Dynamic Status	BIT Dynamic Status
BIT Latched Status	BIT Latched Status
BIT Interrupt Enable	BIT Interrupt Enable
BIT Set Edge/Level Interrupt	BIT Set Edge/Level Interrupt

Overcurrent Dynamic Status	Overcurrent Dynamic Status
Overcurrent Latched Status	Overcurrent Latched Status
Overcurrent Interrupt Enable	Overcurrent Interrupt Enable
Overcurrent Set Edge/Level Interrupt	Overcurrent Set Edge/Level Interrupt
External Power Under Voltage Dynamic Status	External Power Loss Dynamic Status
External Power Under Voltage Latched Status	External Power Loss Latched Status
External Power Under Voltage Interrupt Enable	External Power Loss Interrupt Enable
External Power Under Voltage Set Edge/Level Interrupt	External Power Loss Set Edge/Level Interrupt
Inter-FPGA Failure Dynamic Status	
Inter-FPGA Failure Latched Status	
Inter-FPGA Failure Interrupt Enable	
Inter-FPGA Failure Set Edge/Level Interrupt	
Summary Dynamic Status	
Summary Latched Status	
Summary Interrupt Enable	
Summary Set Edge/Level Interrupt	
FIFO Dynamic Status	FIFO Dynamic Status
FIFO Latched Status	FIFO Latched Status
FIFO Interrupt Enable	FIFO Interrupt Enable
FIFO Set Edge/Level Interrupt	FIFO Set Edge/Level Interrupt
Interrupt Vector	
Interrupt Steering	

9 D/A Hardware Block Diagram

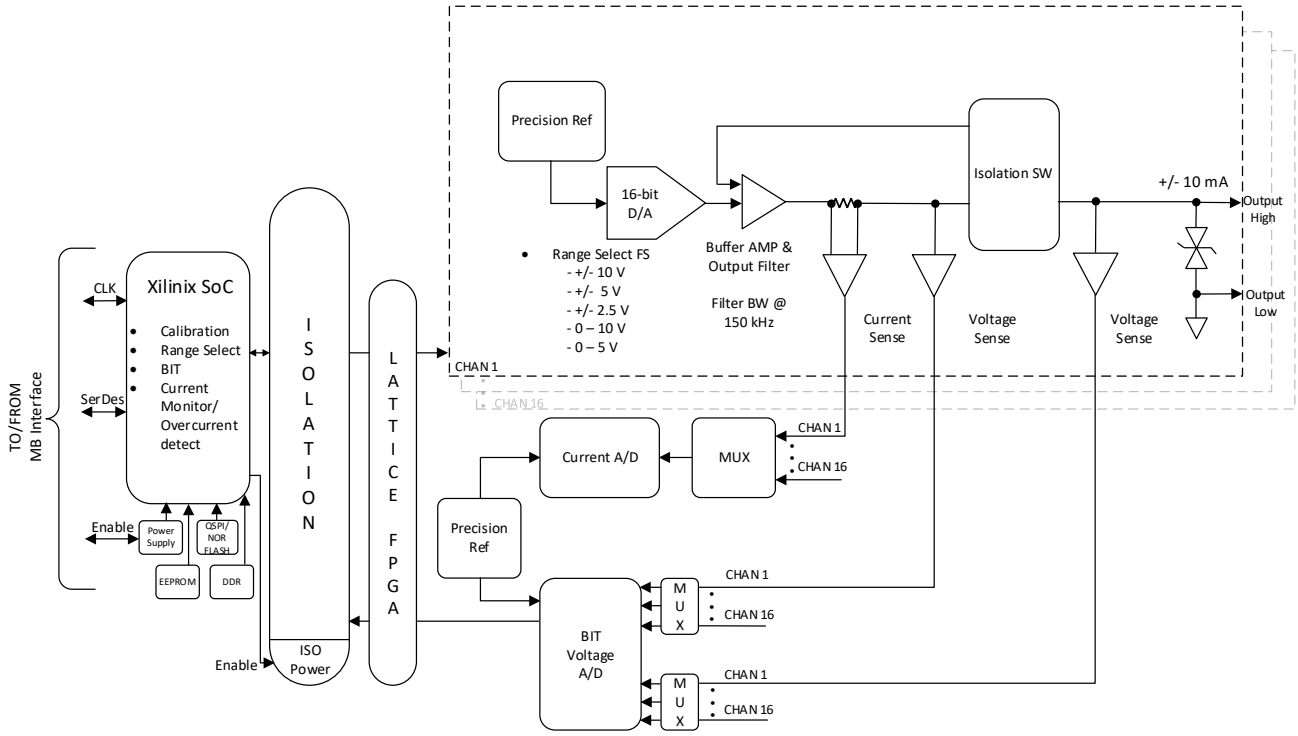


Figure 1 - DA2 Hardware Block Diagram

10 Firmware Revision Notes

This section identifies the firmware revision in which specific features were released. Prior revisions of the firmware would not support the features listed.

Feature	FPGA		Bare Metal (BM)	
	Firmware Revision	Release Date	Firmware Revision	Release Date
Power-On Self-Test (POST) / Power-on BIT (PBIT) / Start-up BIT(SBIT)	2.6 and later	10/08/2019	3.4 and later	01/07/2020
Engineering Scaling Conversions	2.6 and later	10/08/2019	3.4 and later	01/07/2020
Background BIT Threshold Programming	2.6 and later	10/08/2019	3.4 and later	01/07/2020
Watchdog Timer Capability	2.6 and later	10/08/2019	3.4 and later	01/07/2020
Channel Status Enabled and Summary Status	2.6 and later	10/08/2019	3.4 and later	01/07/2020

NAI Cares

North Atlantic Industries (NAI) is a leading independent supplier of Embedded I/O Boards, Single Board Computers, Rugged Power Supplies, Embedded Systems and Motion Simulation and Measurement Instruments for the Military, Aerospace and Industrial Industries. We accelerate our clients' time-to-mission with a unique approach based on a Configurable Open System Architecture™ (COSA®) that delivers the best of both worlds: custom solutions from standard COTS components.

We have built a reputation by listening to our customers, understanding their needs, and designing, testing and delivering board and system-level products for their most demanding air, land and sea requirements. If you are experiencing any problems with our products, we urge you to contact us as soon as possible.

Please visit us at: www.naii.com or select one of the following for immediate assistance:

FAQ

http://www.naii.com/faq/

Application Notes

http://www.naii.com/appNotes/

Calibration and Repairs

http://www.naii.com/calibration/

Call Us

(631) 567-1100

Status and Interrupts

MODULE MANUAL

Revision History

Revision	Revision Date	Description	Author
A	6/17/2019	Initial release	GC
A1	4/22/2020	ECO C07519: Module manuals updated for formatting consistency. No technical or specification updates.	MC

Table of Contents

1 Status and Interrupts

Status registers indicate the detection of faults or events. The status registers can be channel bit-mapped or event bit-mapped. An example of a channel bit-mapped register is the BIT status register, and an example of an event bit-mapped register is the FIFO status register.

For those status registers that allow interrupts to be generated upon the detection of the fault or the event, there are four registers associated with each status: *Dynamic, Latched, Interrupt Enabled*, and *Set Edge/Level Interrupt*.

Dynamic Status: The *Dynamic Status* register indicates the *current* condition of the fault or the event. If the fault or the event is momentary, the contents in this register will be clear when the fault or the event goes away. The *Dynamic Status* register can be polled, however, if the fault or the event is sporadic, it is possible for the indication of the fault or the event to be missed.

Latched Status: The *Latched Status* register indicates whether the fault or the event *has occurred* and keeps the state until it is cleared by the user. Reading the *Latched Status* register is a better alternative to polling the *Dynamic Status* register because the contents of this register will not clear until the user commands to clear the specific bit(s) associated with the fault or the event in the *Latched Status* register. Once the status register has been read, the act of writing a **1** back to the applicable status register to any specific bit (channel/event) location will "clear" the bit (set the bit to **0**). When clearing the channel/event bits, it is strongly recommended to write back the same bit pattern as read from the *Latched Status* register. For example, if the channel bit-mapped *Latched Status* register contains the value 0x0000 0005, which indicates fault/event detection on channel 1 and 3, write the value 0x0000 0005 to the *Latched Status* register to clear the fault/event status for channel 1 and 3. Writing a "1" to other channels that are not set (example 0x0000 000F) may result in incorrectly "clearing" incoming faults/events for those channels (example, channel 2 and 4).

Interrupt Enable: If interrupts are preferred upon the detection of a fault or an event, enable the specific channel/event interrupt in the *Interrupt Enable* register. The bits in *Interrupt Enable* register map to the same bits in the *Latched Status* register. When a fault or event occurs, an interrupt will be fired. Subsequent interrupts will not trigger until the application acknowledges the fired interrupt by clearing the associated channel/event bit in the *Latched Status* register. If the interruptible condition is still persistent after clearing the bit, this may retrigger the interrupt depending on the *Edge/Level* setting.

Set Edge/Level Interrupt: When interrupts are enabled, the condition on retriggering the interrupt after the Latch Register is "cleared" can be specified as "edge" triggered or "level" triggered. Note, the Edge/Level Trigger also affects how the Latched Register value is adjusted after it is "cleared" (see 1.1.1).

- *Edge triggered*: An interrupt will be retriggered when the Latched Status register change from low (0) to high (1) state. Uses for edge-triggered interrupts would include transition detections (Low-to-High transitions, High-to-Low transitions) or fault detections. After "clearing" an interrupt, another interrupt will not occur until the next transition or the re-occurrence of the fault again.
- *Level triggered*: An interrupt will be generated when the Latched Status register remains at the high (1) state. Level-triggered interrupts are used to indicate that something needs attention.

1.1 Interrupt Vector and Steering

When interrupts are enabled, the interrupt vector associated with the specific interrupt can be programmed with a unique number/identifier defined by the user such that it can be utilized in the Interrupt Service Routine (ISR) to identify the type of interrupt. When an interrupt occurs, the contents of the Interrupt Vector registers is reported as part of the interrupt mechanism. In addition to specifying the interrupt vector, the interrupt can be directed ("steered") to the native bus or to the application running on the onboard ARM processor.

1.2 Interrupt Trigger Types

In most applications, limiting the number of interrupts generated is preferred as interrupts are costly, thus choosing the correct Edge/Level interrupt trigger to use is important.

Example 1: Fault detection

This example illustrates interrupt considerations when detecting a fault like an “open” on a line. When an “open” is detected, the system will receive an interrupt. If the “open” on the line is persistent and the trigger is set to “edge”, upon “clearing” the interrupt, the system will not re-generate another interrupt. If, instead, the trigger is set to “level”, upon “clearing” the interrupt, the system will re-generate another interrupt. Thus, in this case, it will be better to set the trigger type to “edge”.

Example 2: Threshold detection

This example illustrates interrupt considerations when detecting an event like reaching or exceeding the “high watermark” threshold value. In a communication device, when the number of elements received in the FIFO reaches the high-watermark threshold, an interrupt will be generated. Normally, the application would read the count of the number of elements in the FIFO and read this number of elements from the FIFO. After reading the FIFO data, the application would “clear” the interrupt. If the trigger type is set to “edge”, another interrupt will be generated only if the number of elements in FIFO goes below the “high watermark” after the “clearing” the interrupt and then fills up to reach the “high watermark” threshold value. Since receiving communication data is inherently asynchronous, it is possible that data can continue to fill the FIFO as the application is pulling data off the FIFO. If, at the time the interrupt is “cleared”, the number of elements in the FIFO is at or above the “high watermark”, no interrupts will be generated. In this case, it will be better to set the trigger type to “level”, as the purpose here is to make sure that the FIFO is serviced when the number of elements exceeds the high watermark threshold value. Thus, upon “clearing” the interrupt, if the number of elements in the FIFO is at or above the “high watermark” threshold value, another interrupt will be generated indicating that the FIFO needs to be serviced.

1.3 Dynamic and Latched Status Registers Examples

The examples in this section illustrate the differences in behavior of the Dynamic Status and Latched Status registers as well as the differences in behavior of Edge/Level Trigger when the Latched Status register is cleared.

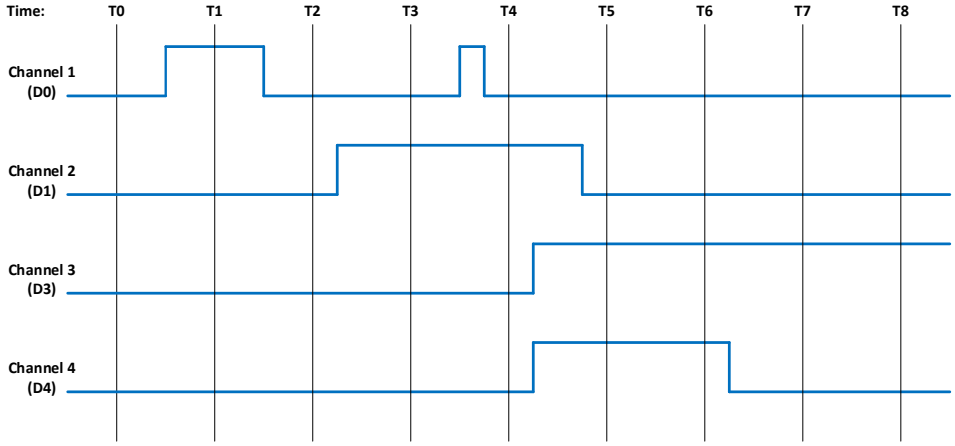


Figure 1 - Example of Module's Channel-Mapped Dynamic and Latched Status States

Time	Dynamic Status	No Clearing of Latched Status	Clearing of Latched Status (Edge-Triggered)		Clearing of Latched Status (Level-Triggered)	
		Latched Status	Action	Latched Status	Action	Latched
T0	0x0	0x0	Read Latched Register	0x0	Read Latched Register	0x0
T1	0x1	0x1	Read Latched Register	0x1		0x1
			Write 0x1 to Latched Register		Write 0x1 to Latched Register	
				0x0		*0x1*
T2	0x0	0x1	Read Latched Register	0x0	Read Latched Register	0x1
					Write 0x1 to Latched Register	
						0x0
T3	0x2	0x3	Read Latched Register	0x2	Read Latched Register	0x2
			Write 0x2 to Latched Register		Write 0x2 to Latched Register	
				0x0		*0x2*
T4	0x2	0x3	Read Latched Register	0x1	Read Latched Register	0x3
			Write 0x1 to Latched Register		Write 0x3 to Latched Register	
				0x0		*0x2*
T5	0xC	0xF	Read Latched Register	0xC	Read Latched Register	0xE
			Write 0xC to Latched Register		Write 0xE to Latched Register	
				0x0		*0xC*
T6	0xC	0xF	Read Latched Register	0x0	Read Latched Register	0xC
					Write 0xC to Latched Register	
						0xC
T7	0x4	0xF	Read Latched Register	0x0	Read Latched Register	0xC
					Write 0xC to Latched Register	
						0x4
T8	0x4	0xF	Read Latched Register	0x0	Read Latched Register	0x4

1.4 Interrupt Examples

The examples in this section illustrate the interrupt behavior with Edge/Level Trigger.

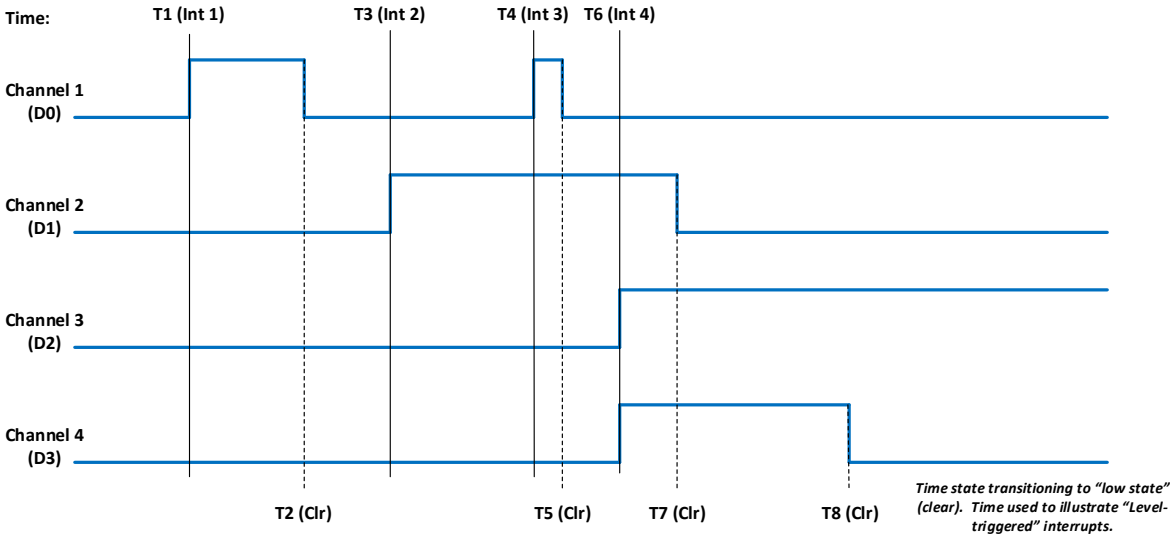


Figure 2 - Illustration of Latched Status State for Module with 4-Channels with Interrupt Enabled

Time	Latched Status (Edge-Triggered – Clear Multi-Channel) Action	Latched	Latched Status (Edge-Triggered – Clear Single Channel) Action	Latched	Latched Status (Level-Triggered – Clear Multi-Channel) Action	Latched
T1 (Int 1)	**Interrupt Generated** Read Latched Registers	0x1	**Interrupt Generated** Read Latched Registers	0x1	**Interrupt Generated** Read Latched Registers	0x1
	Write 0x1 to Latched Register		Write 0x1 to Latched Register		Write 0x1 to Latched Register	
		0x0		*0x0*	***Interrupt re-triggers*** Note, interrupt re-triggers after each clear until T2.	*0x1*
T3 (Int 2)	**Interrupt Generated** Read Latched Registers	0x2	**Interrupt Generated** Read Latched Registers	0x2	**Interrupt Generated** Read Latched Registers	0x2
	Write 0x2 to Latched Register		Write 0x2 to Latched Register		Write 0x2 to Latched Register	
		0x0		*0x0*	***Interrupt re-triggers*** Note, interrupt re-triggers after each clear until T7.	*0x2*
T4 (Int 3)	**Interrupt Generated** Read Latched Registers	0x1	**Interrupt Generated** Read Latched Registers	0x1	**Interrupt Generated** Read Latched Registers	0x3
	Write 0x1 to Latched Register		Write 0x1 to Latched Register		Write 0x3 to Latched Register	
		0x0		*0x0*	***Interrupt re-triggers*** Note, interrupt re-triggers after each clear and 0x3 is reported in Latched Register until T5.	*0x3*
					Interrupt re-triggers Note, interrupt re-triggers after each clear until T7.	*0x2*

T6 (Int 4)	**Interrupt Generated** Read Latched Registers	0xC	**Interrupt Generated** Read Latched Registers	0xC	**Interrupt Generated** Read Latched Registers	0xE
	Write 0xC to Latched Register		Write 0x4 to Latched Register		Write 0xE to Latched Register	
		0x0	***Interrupt re-triggers*** Write 0x8 to Latched Register	*0x8*	***Interrupt re-triggers*** Note, interrupt re-triggers after each clear and 0xE is reported in Latched Register until T7.	*0xE*
				0x0	***Interrupt re-triggers*** Note, interrupt re-triggers after each clear and 0xC is reported in Latched Register until T8.	*0xC*
					Interrupt re-triggers Note, interrupt re-triggers after each clear and 0x4 is reported in Latched Register always.	*0x4*

NAI Cares

North Atlantic Industries (NAI) is a leading independent supplier of Embedded I/O Boards, Single Board Computers, Rugged Power Supplies, Embedded Systems and Motion Simulation and Measurement Instruments for the Military, Aerospace and Industrial Industries. We accelerate our clients' time-to-mission with a unique approach based on a Custom-on-Standard Architecture™ (COSA®) that delivers the best of both worlds: custom solutions from standard COTS components.

We have built a reputation by listening to our customers, understanding their needs, and designing, testing and delivering board and system-level products for their most demanding air, land and sea requirements. If you have any applications or questions regarding the use of our products, please contact us for an expedient solution.

Please visit us at: www.naii.com or select one of the following for immediate assistance:

FAQ

http://www.naii.com/faq/

Application Notes

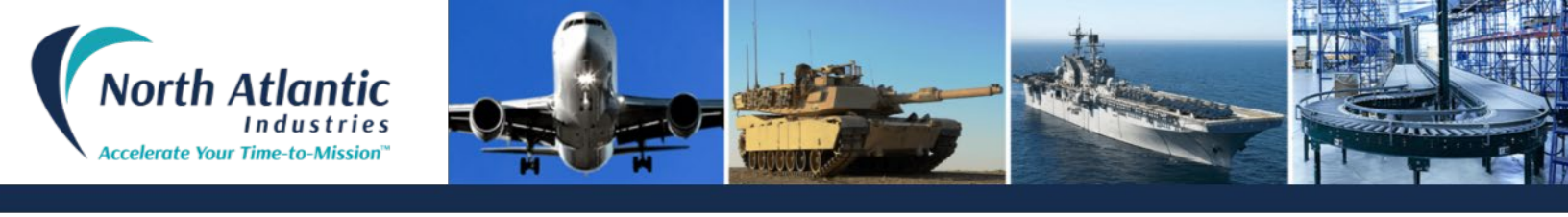
http://www.naii.com/appNotes/

Calibration and Repairs

http://www.naii.com/calibration/

Call Us

(631) 567-1100



User Watchdog Timer

MODULE MANUAL
APPENDIX

Revision History

Revision	Revision Date	Description	Draft/Apprv.
A	6/25/2019	Initial release	GC
A1	10/7/2019	Appended “User” to Watchdog Timer to indicate that the Watchdog Timer is controlled by the user’s application.	GC
A2	4/22/2020	ECO C07519: Module manuals updated for formatting consistency. No technical or specification updates.	MC
B	3/29/2021	ECO C08381: Re-identified Digital-to-Synchro/Resolver (D/S) or Digital-to-L(R)VDT (D/LV) Modules are currently unsupported (at this time).	ARS

Table of Contents

1 User Watchdog Timer Module Manual

1.1 User Watchdog Timer Capability

The User Watchdog Timer (UWDT) Capability is available on the following modules:

- AC Reference Source Modules
  - AC1 – 1 Channel, 2-115 Vrms, 47 Hz – 20kHz
  - AC2 – 2 Channels, 2-28 Vrms, 47 Hz – 20kHz
  - AC3 – 1 Channel, 28-115 Vrms, 47 Hz – 2.5 kHz
- Differential Transceiver Modules
  - DF1/DF2 – 16 Channels Differential I/O
- Digital-to-Analog (D/A) Modules
  - DA1 – 12 Channels, ±10 VDC @ 25 mA, Voltage or Current Control Modes
  - DA2 – 16 Channels, ±10 VDC @ 10 mA
  - DA3 – 4 Channels, ±40 VDC @ ±100 mA, Voltage or Current Control Modes
  - DA4 – 4 Channels, ±80 VDC @ 10 mA
  - DA5 - 4 Channels, ±65 VDC or ±2 A, Voltage or Current Control Modes
- Digital-to-Synchro/Resolver (D/S) or Digital-to-L(R)VDT (D/LV) Modules

  (Not supported)
- Discrete I/O Modules
  - DT1/DT4 – 24 Channels, Programmable for either input or output, output up to 500 mA per channel from an applied external 3 – 60 VCC source.
  - DT2/DT5 – 16 Channels, Programmable for either input voltage measurements (±80 V) or as a bi-directional current switch (up to 500 mA per channel).
  - DT3/DT6 – 4 Channels, Programmable for either input voltage measurements (±100 V) or as a bi-directional current switch (up to 3 A per channel).
- TTL/CMOS Modules
  - TL1-TL8 – 24 Channels, Programmable for either input or output.

1.2 Principle of Operation

The User Watchdog Timer is optionally activated by the applications that require the module's outputs to be disabled as a failsafe in the event of an application failure or crash. The circuit is designed such that a specific periodic write strobe pattern must be executed by the software to maintain operation and prevent the disablement from taking place.

The User Watchdog Timer is inactive until the application sends an initial strobe by writing the value 0x55AA to the *UWDT Strobe* register. After activating the User Watchdog Timer, the application must continually strobe the timer within the intervals specified with the configurable *UWDT Quiet Time* and *UWDT Window* registers. The timing of the strobes must be consistent with the following rules:

- The application must not strobe during the Quiet time.
- The application must strobe within the Window time.
- The application must not strobe more than once in a single window time.

A violation of any of these rules will trigger a User Watchdog Timer fault and result in shutting down any isolated power supplies and/or disabling any active drive outputs, as applicable for the specific module. Upon a User Watchdog Timer event, recovery to the module shutting down will require the module to be reset.

The Figure 1 and Figure 2 provides an overview and an example with actual values for the User Watchdog Timer Strobes, Quiet Time and Window. As depicted in the diagrams, there are two processes that run in parallel. The Strobe event starts the timer for the beginning of the "Quiet Time". The timer for the Previous Strobe event continues to run to ensure that no additional Strobes are received within the "Window" associated with the Previous Strobe.

The optimal target for the user watchdog strobes should be at the interval of [Quiet time + ½ Window time] after the previous strobe, which will place the strobe in the center of the window. This affords the greatest margin of safety against unintended disablement in critical operations.

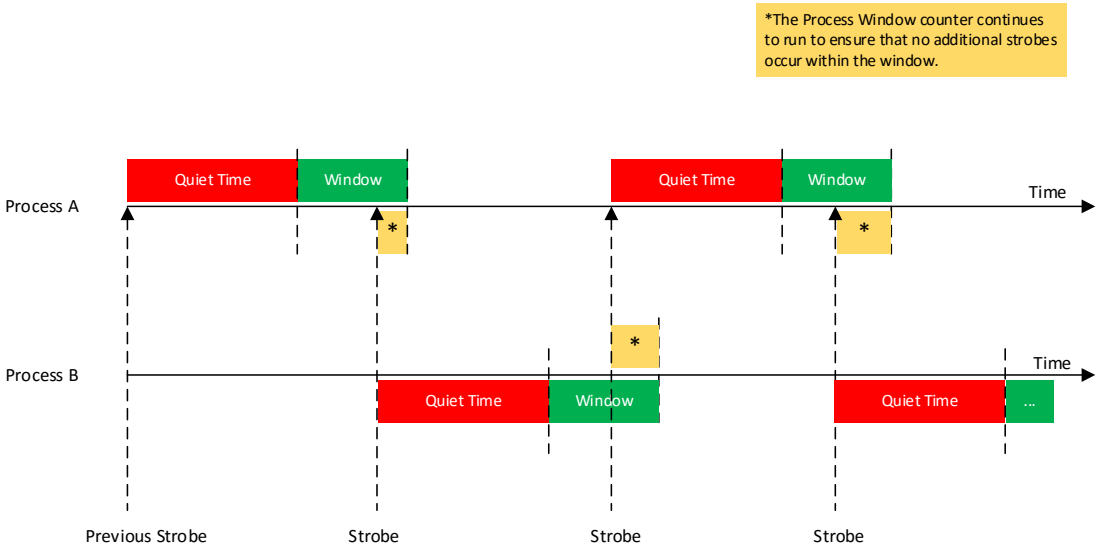


Figure 1 – User Watchdog Timer Overview

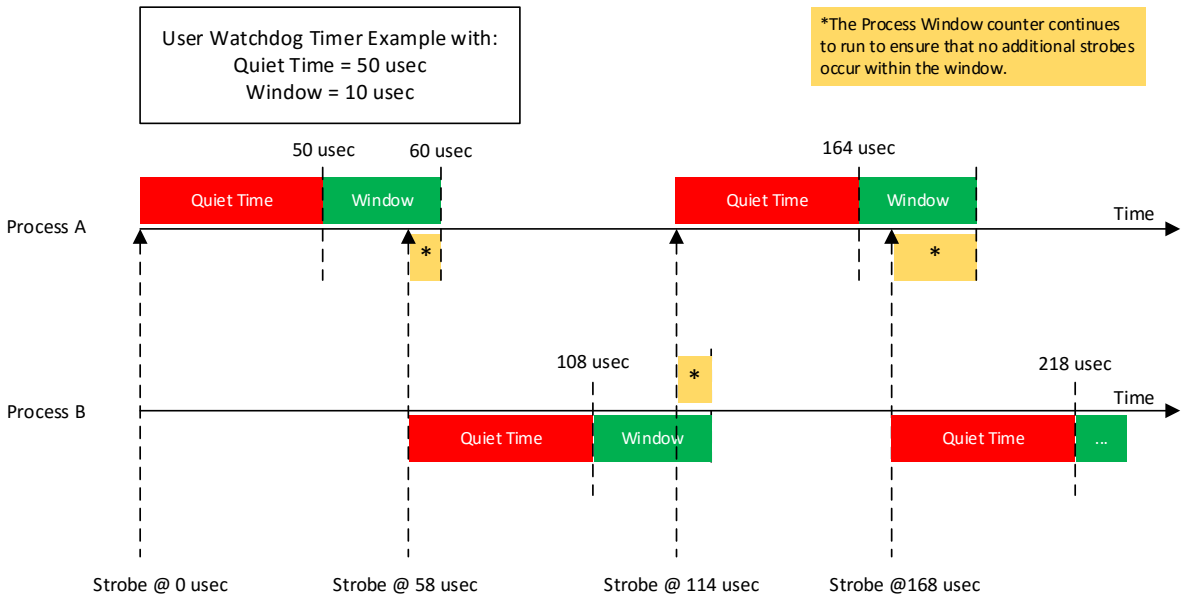


Figure 2 – User Watchdog Timer Example

Figure 3 illustrates examples of User Watchdog Timer failures.

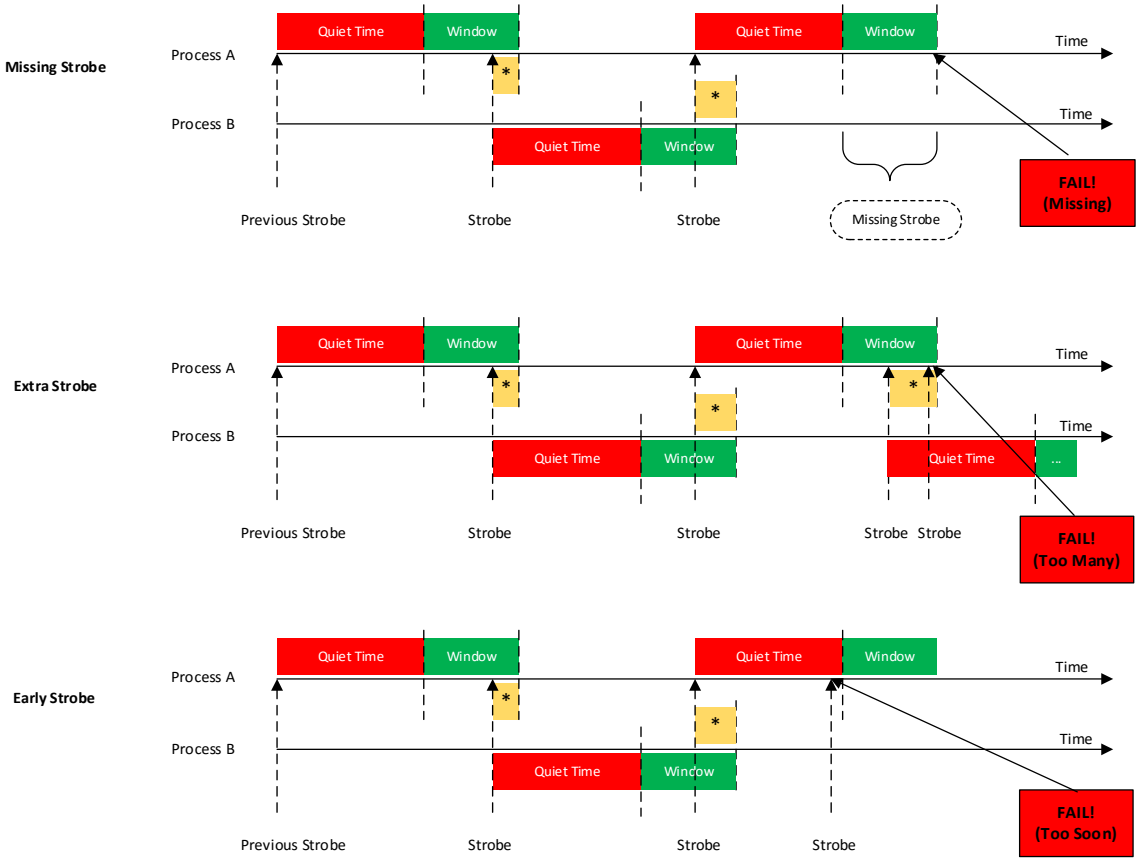


Figure 3 – User Watchdog Timer Failures

1.3 Register Descriptions

The register descriptions provide the register name, Type, Data Range, Read or Write information, Initialized Value, and a description of the function.

1.3.1 User Watchdog Timer Registers

The registers associated with the User Watchdog Timer provide the ability to specify the *UWDT Quiet Time* and the *UWDT Window* that will be monitored to ensure that EXACTLY ONE User Watchdog Timer (UWDT) Strobe is written within the window.

1.3.1.1 UWDT Quiet Time

Function: Sets Quiet Time value (in microseconds) to use for the User Watchdog Timer Frame.

Type: unsigned binary word (32-bit)

Data Range: 0 µsec to 2^32 µsec (0x0 to 0xFFFFFFFF)

Read/Write: R/W

Initialized Value: 0x0

Operational Settings: LSB = 1 µsec. The application must NOT write a strobe in the time between the previous strobe and the end of the Quiet time interval. In addition, the application must write in the *UWDT Window* EXACTLY ONCE.

1.3.1.2 UWDT Window

Function: Sets Window value (in microseconds) to use for the User Watchdog Timer Frame.

Type: unsigned binary word (32-bit)

Data Range: 0 µsec to 2^32 µsec (0x0 to 0xFFFFFFFF)

Read/Write: R/W

Initialized Value: 0x0

Operational Settings: LSB = 1 µsec. The application must write the strobe once within the Window time after the end of the Quiet time interval. The application must write in the *UWDT Window* EXACTLY ONCE.

This setting must be initialized to a non-zero value for operation and should allow sufficient tolerance for strobe timing by the application.

1.3.1.3 UWDT Strobe

Function: Writes the strobe value to be use for the User Watchdog Timer Frame.

Type: unsigned binary word (32-bit)

Data Range: 0x55AA

Read/Write: W

Initialized Value: 0x0

Operational Settings: At startup, the user watchdog is disabled. Write the value of 0x55AA to this register to start the user watchdog timer monitoring after initial power on or a reset. To prevent a disablement, the application must periodically write the strobe based on the user watchdog timer rules.

1.3.2 Status and Interrupt

The modules that are capable of User Watchdog Timer support provide status registers for the User Watchdog Timer.

1.3.2.1 User Watchdog Timer Status

The status register that contains the User Watchdog Timer Fault information is also used to indicate channel Inter-FPGA failures on modules that have communication between FPGA components. There are four registers associated with the User Watchdog Timer Fault/Inter-FPGA Failure Status: *Dynamic, Latched, Interrupt Enable,* and *Set Edge/Level Interrupt*.

User Watchdog Timer Fault/Inter-FPGA Failure Dynamic Status		
User Watchdog Timer Fault/Inter-FPGA Failure Latched Status		
User Watchdog Timer Fault/Inter-FPGA Failure Interrupt Enable		
User Watchdog Timer Fault/Inter-FPGA Failure Set Edge/Level Interrupt		
Bit(s)	**Status**	**Description**
D31	User Watchdog Timer Fault Status	**0** = No Fault **1** = User Watchdog Timer Fault
D30:D0	Reserved for Inter-FPGA Failure Status	Channel bit-mapped indicating channel inter-FPGA communication failure detection.

Function: Sets the corresponding bit (D31) associated with the channel's User Watchdog Timer Fault error.
Type: unsigned binary word (32-bit)
Data Range: 0x0000 0000 to 0xFFFF FFFF
Read/Write: R (*Dynamic*), R/W (*Latched*, *Interrupt Enable*, *Edge/Level Interrupt*)
Initialized Value: 0

1.3.2.2 Interrupt Vector and Steering

When interrupts are enabled, the interrupt vector associated with the specific interrupt can be programmed (typically with a unique number/identifier) such that it can be utilized in the Interrupt Service Routine (ISR) to identify the type of interrupt. When an interrupt occurs, the contents of the Interrupt Vector registers is reported as part of the interrupt mechanism.

In addition to specifying the interrupt vector, the interrupt can be directed ("steered") to the native bus or to the application running on the onboard ARM processor.

Note, the Interrupt Vector and Interrupt Steering registers are mapped to the Motherboard Common Memory and these registers are associated with the Module Slot position (refer to Function Register Map).

1.3.2.3 Interrupt Vector

Function: Set an identifier for the interrupt.
Type: unsigned binary word (32-bit)
Data Range: 0x0000 0000 to 0xFFFF FFFF
Read/Write: R/W
Initialized Value: 0
Operational Settings: When an interrupt occurs, this value is reported as part of the interrupt mechanism.

1.3.2.4 Interrupt Steering

Function: Sets where to direct the interrupt.
Type: unsigned binary word (32-bit)
Data Range: See table
Read/Write: R/W
Initialized Value: 0
Operational Settings: When an interrupt occurs, the interrupt is sent as specified:

Direct Interrupt to VME	1
Direct Interrupt to ARM Processor (via SerDes) *(Custom App on ARM or NAI Ethernet Listener App)*	2
Direct Interrupt to PCIe Bus	5
Direct Interrupt to cPCI Bus	6

1.3.3 Function Register Map

Key: ***Bold Italic = Configuration/Control***

Bold Underline = Status

*When an event is detected, the bit associated with the event is set in this register and will remain set until the user clears the event bit. Clearing the bit requires writing a 1 back to the specific bit that was set when read (i.e. write-1-to-clear, writing a '1' to a bit set to '1' will set the bit to '0').

1.3.3.1 User Watchdog Timer Registers

0x01C0	***UWDT Quiet Time***	R/W
0x01C4	***UWDT Window***	R/W
0x01C8	***UWDT Strobe***	W

1.3.3.2 Status Registers

User Watchdog Timer Fault/Inter-FPGA Failure

0x09B0	**Dynamic Status**	R
0x09B4	**Latched Status***	R/W
0x09B8	***Interrupt Enable***	R/W
0x09BC	***Set Edge/Level Interrupt***	R/W

1.3.3.3 Interrupt Register

The Interrupt Vector and Interrupt Steering registers are mapped to the Motherboard Memory Space and these addresses are absolute based on the module slot position. In other words, do not apply the Module Address offset to these addresses.

0x056C	***Module 1 Interrupt Vector 28 – User Watchdog Timer Fault/Inter-FPGA Failure***	R/W
0x066C	***Module 1 Interrupt Steering 28 – User Watchdog Timer Fault/Inter-FPGA Failure***	R/W
0x076C	***Module 2 Interrupt Vector 28 – User Watchdog Timer Fault/Inter-FPGA Failure***	R/W
0x086C	***Module 2 Interrupt Steering 28 – User Watchdog Timer Fault/Inter-FPGA Failure***	R/W
0x096C	***Module 3 Interrupt Vector 28 – User Watchdog Timer Fault/Inter-FPGA Failure***	R/W
0x0A6C	***Module 3 Interrupt Steering 28 – User Watchdog Timer Fault/Inter-FPGA Failure***	R/W
0x0B6C	***Module 4 Interrupt Vector 28 – User Watchdog Timer Fault/Inter-FPGA Failure***	R/W
0x0C6C	***Module 4 Interrupt Steering 28 – User Watchdog Timer Fault/Inter-FPGA Failure***	R/W
0x0D6C	***Module 5 Interrupt Vector 28 – User Watchdog Timer Fault/Inter-FPGA Failure***	R/W
0x0E6C	***Module 5 Interrupt Steering 28 – User Watchdog Timer Fault/Inter-FPGA Failure***	R/W
0x0F6C	***Module 6 Interrupt Vector 28 – User Watchdog Timer Fault/Inter-FPGA Failure***	R/W
0x106C	***Module 6 Interrupt Steering 28 – User Watchdog Timer Fault/Inter-FPGA Failure***	R/W

NAI Cares

North Atlantic Industries (NAI) is a leading independent supplier of Embedded I/O Boards, Single Board Computers, Rugged Power Supplies, Embedded Systems and Motion Simulation and Measurement Instruments for the Military, Aerospace and Industrial Industries. We accelerate our clients' time-to-mission with a unique approach based on a Custom-on-Standard Architecture™ (COSA®) that delivers the best of both worlds: custom solutions from standard COTS components.

We have built a reputation by listening to our customers, understanding their needs, and designing, testing and delivering board and system-level products for their most demanding air, land and sea requirements. If you have any applications or questions regarding the use of our products, please contact us for an expedient solution.

Please visit us at: www.naii.com or select one of the following for immediate assistance:

FAQ

http://www.naii.com/faq/

Application Notes

http://www.naii.com/appNotes/

Calibration and Repairs

http://www.naii.com/calibration/

Call Us

(631) 567-1100