DA4

Digital to Analog

Function Module

MODULE MANUAL

Revision History

Module Manual - DA4 Revision History		
Revision	**Revision Date**	**Description**
C	2022-11-03	ECO C09774, initial release of module manual.

Module Manual - Status and Interrupts Revision History		
Revision	**Revision Date**	**Description**
C	2021-11-30	C08896; Transition manual to docbuilder format - no technical info change.

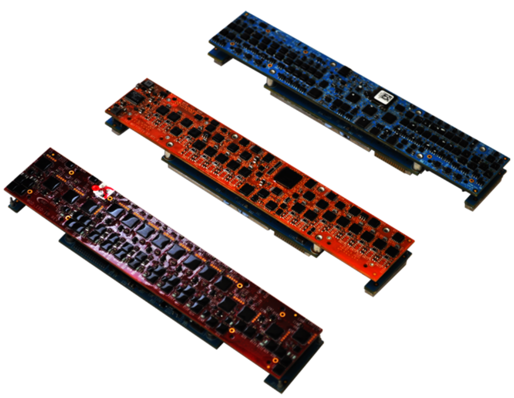
Module Manual - User Watchdog Timer Revision History		
Revision	**Revision Date**	**Description**
C	2021-11-30	C08896; Transition manual to docbuilder format - no technical info change.

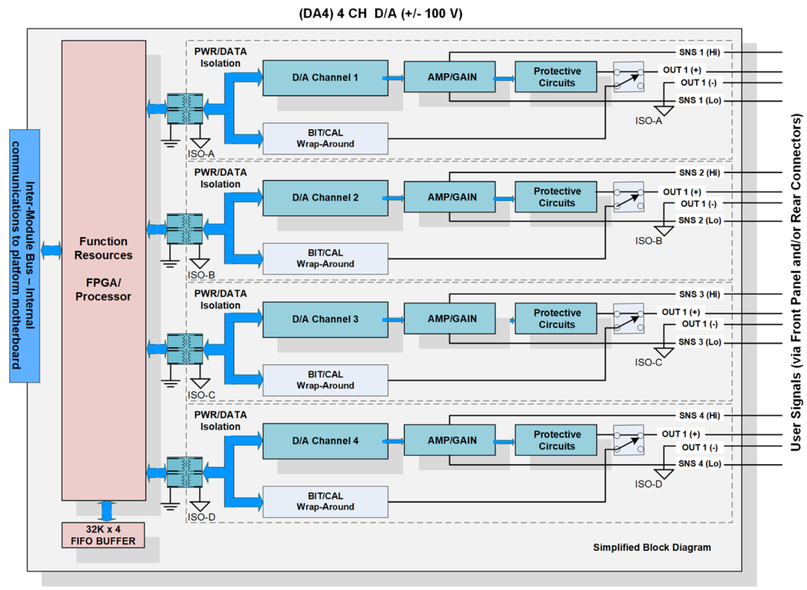
DA4 I/O Modules Digital-to-Analog Function Modules

4 (High Voltage) D/A Outputs (±100 VDC)

The Digital-to-Analog (D/A) module DA4 provides 4 high-voltage independent D/A output channels with a maximum full-scale range of ±100 VDC at ±10 mA (max). Linearity/accuracy is ±0.15% FS range over temperature. The DA4 provides voltage control loop mode, which is programmable for the application.

The DA4 module includes extensive Background Built-in-Test (BIT). In addition to output signal read-back (wrap) capabilities, overloaded outputs are detected with automatic channel shut-down protection, and the results are displayed in a status word. DA4 also include D/A FIFO buffering for greater control of the output voltage and signal data. Once enabled and triggered, the D/A FIFO buffer accepts, stores, and outputs the voltage commands 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.





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**

Specifications

Resolution	16-bit/channel for voltage command mode.
Output Format	Single-ended
Output Range	±100 VDC (unipolar or bipolar); ranges are programmable in pairs.
Output Impedance	< 1 Ω
System Protection	Output is set to 0 V at reset or Power-On.
Linearity Error	±0.15% FS range over temperature
Offset Error	±100 mV
Gain Error	±0.08% FS range
Settling Time	350 µs (max.)
Data Buffer	See Operations Manual for 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 any channel, for > 50 ms, that channel is set to 0 V and a flag is set.
Update Rate	5 µs per channel
ESD Protection	Designed to meet the testing requirements of IEC 801-2 Level 2. (4 kV transient with a peak current of 7.5 A and a time constant of approximately 60 ns).
Power	5 VDC @ 300 mA typical (est.); ±12 VDC @ 200 mA (est. quiescent) Add 2 mA per 1 mA load per channel
Ground	Channel pair returns (Ch-Lo (-)) are common but are isolated (250 Vpeak) from system ground.
Weight	1.5 oz. (42 g)

Architected for Versatility

NAI’s Configurable Open Systems Architecture™ (COSA®) offers a choice of over 100 smart I/O, communications, or Ethernet switch functions, providing the highest packaging density and greatest flexibility of ruggedized embedded product solutions in the industry. Preexisting, fully-tested functions can be combined in an unlimited number of ways quickly and easily.

One-Source Efficiencies

Eliminate man-months of integration with a configured, field-proven system from NAI. Specification to deployment is a seamless experience as all design, state-of-the-art manufacturing, assembly and test are performed - by one trusted source. All facilities are located within the U.S. and optimized for high-mix/low volume production runs and extended lifecycle support.

Product Lifecycle Management

From design to production and beyond, NAI’s product lifecycle management strategy ensures the long-term availability of COTS products through configuration management, technology refresh and obsolescence component purchase and storage.

INTRODUCTION

This module manual provides information about the North Atlantic Industries, Inc. (NAI) Digital-to-Analog Function Module: DA4. This module is compatible with all NAI Generation 5 motherboards.

Digital-to-Analog (D/A) module DA4 provides 4 independent D/A output channels with a full-scale range ±100 VDC @ ±10 mA (max). Linearity/accuracy is ±0.15% FS range over temperature. The DA4 provides voltage control loop mode, which is programmable for the application.

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

PRINCIPLE OF OPERATION

In addition to the functions and features already described, this 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 module 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.

Built-In Test (BIT)/Diagnostic Capability

The DA4 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.

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.

Continuous Background Built-In Test (CBIT)

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.

Initiated Built-In Test (IBIT)

The DA4 module supports 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. The 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.

Voltage Feedback Sense Lines

The DA4 includes two voltage feedback sense lines for each output channel. It is always recommended that the sense lines be utilized and connected at the external load to provide maximum measurement accuracy (which ensures that the commanded voltage is applied at the load). Sense (Hi) should be connected to Output (Hi), and Sense (Lo) should be connected to Output (Lo).

If additional wiring is prohibitive in the application, the sense lines may alternately be terminated at the connector.

D/A FIFO Buffering/Pattern Buffer

The DA4 module provides the ability to use memory buffers either as a Pattern buffer, (addressable RAM used for creating an output pattern (or cycling) or as a FIFO buffer. These buffers provide greater control of the output voltage and signal data. The D/A buffers 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 ranges.

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.

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)*
- *DAC Value (Voltage (Volts))**
- *FIFO Buffer Data/Pattern RAM Buffer**

**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 interior 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/ Pattern RAM Buffer*

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.

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.

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.

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.

D/A Control Registers

The D/A control registers provide the ability to specify how the D/A channel is outputting voltage, polarity and *voltage range*, *update rate*, and the enabling or disabling of the power to the D/A channel. 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. The D/A *Overcurrent Value* register provides that ability to programmatically change the threshold of the overcurrent detection.

Voltage Range

Function: Sets voltage polarity and range for each channel. The value written to the *DAC Value* register (or values coming from the memory buffer) 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***.

Type: unsigned binary word (32-Bit)

Data Range: See table below.

Read/Write: R/W

Initialized Value: 0 (Unipolar: 0-50 V)

Operational Settings: Write to the register with a value from the table to select the range**.** Ex: for the 0-40V unipolar range write a 0x0.

Reg Value	*Voltage Range*
0x0	Unipolar: 0 – 50 V
0x1	Unipolar: 0 – 100 V
0x2	Bipolar: ± 25 V
0x3	Bipolar: ± 50 V
0x4	Bipolar: ± 100 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**

Power Enable

Function: Enables the DAC's channel power.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 000F

Read/Write: R/W

Initialized Value: 0x0000 000F (Channel power is enabled)

Operational Settings: Set bit to **1 to** enable the power. Set bit to **0** to disable the power.

Power 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
0	0	0	0	0	0	0	0	0	0	0	0	**Ch4**	**Ch3**	**Ch2**	**Ch1**

Update Rate

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

Type: unsigned binary word (32bit)

Data Range: 0x001E 8480 to 0x0003 0D40; 500ns (2MHz) to 5µs (200kHz).

Read/Write: R/W

Initialized Value: 0x001E 8480 (500ns) (2MHz)

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

Overcurrent Reset

Function: Resets over loaded channels based on the *Overcurrent Status* 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**

Overcurrent Value

Function: Sets the *overcurrent value* to be used to determine the overcurrent condition.

Type: unsigned binary word (32-bit)

Data Range: 0x0001 to 0x000C; 1 mA to 12 mA.

Read/Write: R/W

Initialized Value: 0x000C (12 mA); Note - steady state current rating for each channel is still 10 mA

Operational Settings: LSB = 1 mA. This setting is the value that is used to determine the overcurrent condition for each channel. This value is applied to both positive and negative currents. Note, the maximum value for the *overcurrent value* is 12 mA; values greater than this will be forced to the maximum value.

D/A Measurement Registers

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

Wrap Voltage

Function: *Wrap voltage* reading from the channel's output. Also used in conjunction with BIT to verify that the output voltage is within range of the user set *DAC value* (voltage-control mode). Accuracy is 0.2% FS.

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 (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: 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 0-50V then the LSB would have value (50-0)/2^16 = 0.763 mV. Sign bit = **D17** for bipolar ranges.

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

Wrap Current

Function: *Wrap current* reading from the channel's output. Reads current values of D/A outputs being delivered per channel. Accuracy is 0.2% FS.

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 (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: LSB = 305 nA. To calculate the LSB subtract the minimum range (-.010) from the maximum range (.010) then divide by 2^16. Sign bit = **D17** for bipolar ranges.

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**

D/A Test Registers

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

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

FIFO/RAM Registers

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 000F

Read/Write: R/W

Initialized Value: 0 (The output will reflect the *DAC Value* register value)

Operational Settings: Write a **1** to use the memory buffer. Bit-mapped per channel.

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
0	0	0	0	0	0	0	0	0	0	0	0	**Ch4**	**Ch3**	**Ch2**	**Ch1**

Buffer Mode

Function: Selects how the memory buffer will be used; either as a Pattern buffer (addressable RAM used for creating an output pattern (or cycling)) or FIFO buffer.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 000F

Read/Write: R/W

Initialized Value: 0 (RAM Mode)

Operational Settings: Write a **1** to use the buffer for the channel as a FIFO. Write a **0** to use the buffer as Pattern RAM. To use the memory buffer, ensure that the *Data Mode* register is set properly. Bit-mapped per channel.

Buffer 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
0	0	0	0	0	0	0	0	0	0	0	0	**Ch4**	**Ch3**	**Ch2**	**Ch1**

FIFO Registers

FIFO Buffer Data

Function: Data in the form of *DAC value*s are written to this register one word at a time (32-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-bits) 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: N/A

Operational Settings: Data is held in FIFO until triggered. FIFO size is 32767 words per channel (each channel has its own buffer).

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 0x0000 7FFF (empty to 32767)

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 32767 words.

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.

FIFO Almost Empty

Function: This register enables the user to set the limit for the “almost empty” status.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 7FFF

Read/Write: R/W

Initialized Value: 0x400 (1024)

Operational Settings: When the *FIFO Word Count* 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 *FIFO Count* is greater than the value stored in the register, the “almost empty” bit (**D1**) of the *FIFO Status* register will be cleared.

FIFO Low Watermark

Function: The *FIFO low watermark* threshold enables the user to set the limit for the “low watermark” status.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 7FFF

Read/Write: R/W

Initialized Value: 0x2000 (8192)

Operational Settings: When the *FIFO Count* is less than or equal than 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 Count* is greater than the value stored in the register, the “low watermark” bit (**D2**) of the *FIFO Status* register will be cleared.

FIFO High Watermark

Function: The *FIFO high watermark* threshold enables the user to set the limit for the “high watermark” status.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 7FFF

Read/Write: R/W

Initialized Value: 0x6000 (24576)

Operational Settings: When the *FIFO Count* 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 Count* counter is less than the value stored in the register, the “high watermark” bit (**D3**) of the *FIFO Status* register will be cleared.

FIFO Almost Full

Function: This register enables the user to set the limits for the “almost full” status.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 7FFF

Read/Write: R/W

Initialized Value: 0x7C00 (31744)

Operational Settings: When the *FIFO 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 *FIFO Count* is less than the value stored in the *FIFO Almost Full Value register*, the “almost full” bit (**D4**) of the *FIFO Status* register will be cleared

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

FIFO 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 Data* register the *Use FIFO* register must be enabled. Then write a **1** to the *FIFO Control* register to begin outputting data. The **1** will clear once the FIFO empties.

FIFO 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

Pattern Registers

Pattern Control

Function: Control the RAM data output.

Type: unsigned binary word (32-bit)

Data Range: N/A

Read/Write: R/W

Initialized Value: 0

Operational Settings: The *Pattern Start Address* register determines where the Pattern output will begin when enabled. The *Pattern End Address* register determines where the *Pattern* output will end when enabled. After the pattern at the *Pattern End Address* is outputted, it will loop back to the start address.

Description	
D0	Enable. This bit will enable or disable continuous pattern looping. Write **0x1** to enable and **0x0** to disable the pattern.
D1	Burst Mode: Writing **0x3** (**D0** and **D1**) will burst the pattern from the start address to the end address for N number of times. N is determined by the value written in the *Pattern Number of Cycles* register.
D2	Pause: Set this bit to pause the pattern when enabled

Pattern RAM Control															
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**

Pattern Start Address

Function: Programs the starting address for the *Pattern* output buffer. There are 32767 address locations.

Type: unsigned binary word (32-bit)

Data Range: Ch 1: 0x0002 0000 to 0x0003 FFFC; Ch 2: 0x0004 0000 to 0x0005 FFFC; Ch 3: 0x0006 0000 to 0x0007 FFFC; Ch 4: 0x0008 0000 to 0x0009 FFFC

Read/Write: R/W

Initialized Value: Ch 1: 0x0002 0000; Ch 2: 0x0004 0000; Ch 3: 0x0006 0000; Ch 4: 0x0008 0000;

Operational Settings: Address where the Pattern buffer will start when it is enabled.

Pattern End Address

Function: Programs the ending address for the *Pattern* output buffer. There are 32767 address locations.

Type: unsigned binary word (32-bit)

Data Range: Ch 1: 0x0002 0000 to 0x0003 FFFC; Ch 2: 0x0004 0000 to 0x0005 FFFC; Ch 3: 0x0006 0000 to 0x0007 FFFC; Ch 4: 0x0008 0000 to 0x0009 FFFC

Read/Write: R/W

Initialized Value: Ch 1: 0x0002 0000; Ch 2: 0x0004 0000; Ch 3: 0x0006 0000; Ch 4: 0x0008 0000;

Operational Settings: Address where the Pattern buffer will end when it is enabled. After the *DAC value* that is stored at this address is outputted, the buffer will jump back to the *Pattern Start Address.*

Pattern Number of Cycles

Function: Set the number of *Pattern* cycles for a channel.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0001 to 0xFFFF FFFF

Read/Write: R/W

Initialized Value: 0

Operational Settings: When the Pattern buffer is enabled in burst mode, it will loop from the *Pattern Start Address* to the *Pattern End Address* the number of times that is set in this register.

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.

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.

Floating Point Offset

Function: This register sets the floating-point offset to add to DA 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.

Floating Point Scale

Function: This register sets the floating-point scale to multiple to the DA 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* or *Current 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.

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 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**

Background BIT Threshold 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.

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: 0x5 (5ms)

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.

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 000F

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
0	0	0	0	0	0	0	0	0	0	0	0	**Ch4**	**Ch3**	**Ch2**	**Ch1**

Watchdog Timer Registers

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

Status and Interrupt Registers

The DA4 Module provides status registers for BIT, Overcurrent, External Power Under Voltage, Inter-FPGA Failure, and FIFO.

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 000F (Channel Status)

Read/Write: R/W

Initialized Value: 0x0000 000F

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**
0	0	0	0	0	0	0	0	0	0	0	0	**Ch4**	**Ch3**	**Ch2**	**Ch1**

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**
0	0	0	0	0	0	0	0	0	0	0	0	**Ch4**	**Ch3**	**Ch2**	**Ch1**

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

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 000F

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.

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**
0	0	0	0	0	0	0	0	0	0	0	0	**Ch4**	**Ch3**	**Ch2**	**Ch1**

Function: Sets the corresponding bit associated with the channel's Overcurrent error.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 000F

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

Initialized Value: 0

External Power Under Voltage

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: Sets 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

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; **0x000F** = 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**
0	0	0	0	0	0	0	0	0	0	0	0	**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 000F

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

Initialized Value: 0

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**
0	0	0	0	0	0	0	0	0	0	0	0	**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 000F

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

Initialized Value: 0

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 = 32767 Words (0x0000 7FFF)	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)

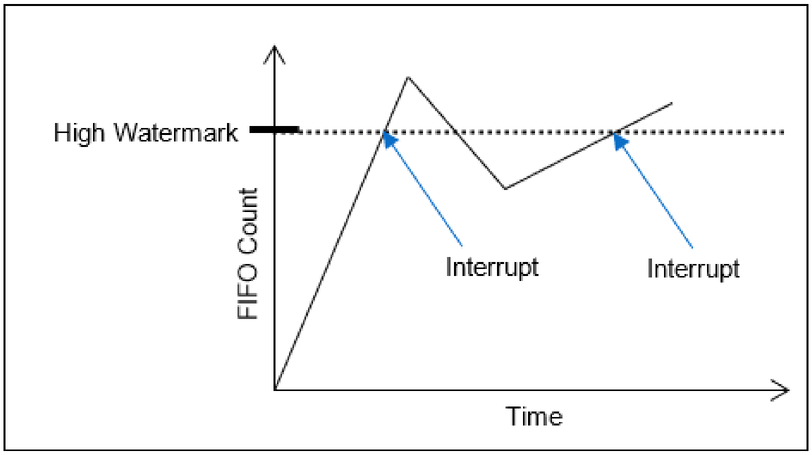
Data Range: 0x0000 0000 to 0x0000 003F

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.



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).

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.

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

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.

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

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

0x0250	***Power Enable Ch 1-4***	R/W
0x100C	***Update Rate Ch 1-4***	R/W
0x1010	***Overcurrent Reset Ch 1-4***	R/W

0x2048	***Overcurrent Value Ch 1***	R/W
0x2148	***Overcurrent Value Ch 2***	R/W
0x2248	***Overcurrent Value Ch 3***	R/W
0x2348	***Overcurrent Value Ch 4***	R/W

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

0x200C	**Wrap Current Ch 1****	R
0x210C	**Wrap Current Ch 2****	R
0x220C	**Wrap Current Ch 3****	R
0x230C	**Wrap Current Ch 4****	R

FIFO/RAM Registers

FIFO/RAM Controls

0x1004	***Data Mode Ch 1-4***	R/W
0x1008	***RAM/FIFO Mode Ch 1-4***	R/W

FIFO

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

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

0x2010	***Clear FIFO Ch 1***	W
0x2110	***Clear FIFO Ch 2***	W
0x2210	***Clear FIFO Ch 3***	W
0x2310	***Clear FIFO Ch 4***	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

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

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

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

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

RAM

0x2030	*Pattern Control Ch 1*	R/W
0x2130	*Pattern Control Ch 2*	R/W
0x2230	*Pattern Control Ch 3*	R/W
0x2330	*Pattern Control Ch 4*	R/W

0x2034	*Pattern Start Address Ch 1*	R/W
0x2134	*Pattern Start Address Ch 2*	R/W
0x2234	*Pattern Start Address Ch 3*	R/W
0x2334	*Pattern Start Address Ch 4*	R/W

0x2038	*Pattern End Address Ch 1*	R/W
0x2138	*Pattern End Address Ch 2*	R/W
0x2238	*Pattern End Address Ch 3*	R/W
0x2338	*Pattern End Address Ch 4*	R/W

0x203C	*Pattern Number of Cycles Ch 1*	R/W
0x213C	*Pattern Number of Cycles Ch 2*	R/W
0x223C	*Pattern Number of Cycles Ch 3*	R/W
0x233C	*Pattern Number of Cycles Ch 4*	R/W

0x0002 0000 to 0x0003 FFFC	*Pattern RAM Data Space Ch 1***	W
0x0004 0000 to 0x0005 FFFC	*Pattern RAM Data Space Ch 2***	W
0x0006 0000 to 0x0007 FFFC	*Pattern RAM Data Space Ch 3***	W
0x0008 0000 to 0x0009 FFFC	*Pattern RAM Data Space Ch 4***	W

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

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

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.

D/A Status Registers

0x02B0	*Channel Status Enabled*	R/W

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*.

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
0x0939	*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 3	0x0830	Dynamic Status	R
	0x0814	Latched Status*	R/W		0x0834	Latched Status*	R/W
	0x0818	*Interrupt Enable*	R/W		0x0838	*Interrupt Enable*	R/W
	0x081C	*Set Edge/Level Interrupt*	R/W		0x083C	*Set Edge/Level Interrupt*	R/W
Ch 2	0x0820	Dynamic Status	R	Ch 4	0x0840	Dynamic Status	R
	0x0824	Latched Status*	R/W		0x0844	Latched Status*	R/W
	0x0828	*Interrupt Enable*	R/W		0x0848	*Interrupt Enable*	R/W
	0x082C	*Set Edge/Level Interrupt*	R/W		0x084C	*Set Edge/Level Interrupt*	R/W

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 to 0x0540	*Module 1 Interrupt Vector 6-17 - Reserved*	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 to 0x0640	*Module 1 Interrupt Steering 6-17 - Reserved*	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 to 0x0740	*Module 2 Interrupt Vector 6-17 - Reserved*	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 to 0x0840	*Module 2 Interrupt Steering 6-17 - Reserved*	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 to 0x0940	*Module 3 Interrupt Vector 6-17 - Reserved*	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 to 0x0A40	*Module 3 Interrupt Steering 6-17 - Reserved*	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 to 0x0B40	*Module 4 Interrupt Vector 6-17 - Reserved*	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 to 0x0C40	*Module 4 Interrupt Steering 6-17 - Reserved*	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 to 0x0D40	***Module 5 Interrupt Vector 6-17 - Reserved***	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 to 0x0E40	***Module 5 Interrupt Steering 6-17 - Reserved***	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 to 0x0F40	***Module 6 Interrupt Vector 6-17 - Reserved***	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 to 0x1040	***Module 6 Interrupt Steering 6-17 - Reserved***	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

APPENDIX A: INTEGER/FLOATING POINT MODE PROGRAMMING

Integer Mode Programming

The following registers should be configured as follows:

Register	Value	Description
Voltage Range	0x3	±50 volts
Enable Floating Point	0	Disable for Floating Point Mode

Note: LSB for Bipolar ±50-volt range:

LSB = 40/0x00007FFF

= 40/32767=1525uV

DAC Value (Integer)	DAC Voltage Output
1.000 of FS = 0x0000 7FFF	32767 * LSB = 50.0 volts
0.5 of FS = 0x0000 4000	16384 * LSB = 25.0 volts
0.0 of FS = 0x0000 0000	0 * LSB = 0.0 volts
-0.5 of FS = 0xFFFF C000	-16384 * LSB = -25.0 volts
-1.0 of FS = 0xFFFF 8000	-32768 * LSB = -50.0 volts

Floating Point Mode Voltage Programming

The following registers should be configured as follows:

Register	Value	Description
Voltage Range	0x3	±50 volts
Enable Floating Point	1	Enable for Floating Point Mode
Floating Point Scale	0.05	Scale = 1 / (Full Range) = 1 / 50.0 = 0.02
Floating Point Offset	0.0	No Offset

Note: LSB for Bipolar ±50-volt range:

LSB = 50/0x00007FFF

= 50/32767=1525uV

DAC Value (volts) (Floating Point)	DAC Value (Calculated by Module)	DAC Value (Integer)	DAC Voltage Output
50.0	(50.0 + 0.0) * 0.02 = 1 (FS)	FS = 0x0000 7FFF	32767 * LSB = 50.0 volts
25.0	(25.0 + 0.0) * 0.02 = 0.5 of FS	0.5 of FS = 0x0000 4000	16384 * LSB = 25.0 volts
0.0	(0.0 + 0.0) * 0.02 = 0.0 of FS	0.0 of FS = 0x0000 0000	0 * LSB = 0.0 volts
-25.0	(-25.0 + 0.0) * 0.02 = -0.5 of FS	-0.5 of FS = 0xFFFF C000	-16384 * LSB = -25.0 volts
-50.0	(-50.0 + 0.0) * 0.02 = -1 (-FS)	-1.0 of FS = 0xFFFF 8000	-32768 * LSB = -50.0 volts

Floating Point Mode Engineering Units Programming

Example #1:

An application wants to associate -50 to 50 volts to -5 to 5 inches.

The following registers should be configured as follows:

Register	Value	Description
Voltage Range	0x3	±50 volts
Enable Floating Point	1	Enable for Floating Point Mode
Floating Point Scale	0.2	Scale = 1 / inches range = 1 / 5 inches = 0.2
Floating Point Offset	0.0	No Offset

Note: LSB for Bipolar ±50-volt range:

LSB = 50/0x00007FFF

= 50/32767=1220uV

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 = 50.0 volts
2.5	(2.5 + 0.0) * 0.2 = 0.5 FS	0.5 of FS = 0x0000 4000	16384 * LSB = 25.0 volts
0.0	(0.0 + 0.0) * 0.2 = 0.0 FS	0.0 of FS = 0x0000 0000	0 * LSB = 0.0 volts
-2.5	(-2.5 + 0.0) * 0.2 = -0.5 FS	-0.5 of FS = 0xFFFF C000	-16384 * LSB = -25.0 volts
-5.0	(-5.0 + 0.0) * 0.2 = -1 (-FS)	-FS = 0xFFFF 8000	-32768 * LSB = -50.0 volts

Example #2:

An application wants to associate 0 to 50 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	0x0	Unipolar 0-50 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 50-volt range:

LSB = 50/0x0000FFFF

= 50/65535=763uV

DAC Value (ft) (Floating Point)	DAC Value (Calculated by Module)	DAC Value (Integer)	DAC Voltage Output
50.00	(50.00 - 0.50) * 0.02 = 0.99 of FS	0.99 of FS = 0x0000 FD70	64880 * LSB = 49.50 volts
25.00	(25.00 - 0.50) * 0.02 = 0.49 of FS	0.49 of FS = 0x0000 7D70	32112 * LSB = 24.50 volts
5.50	(5.50 - 0.50) * 0.02 = 0.10 of FS	0.10 of FS = 0x0000 199A	6554 * LSB = 5.00 volts
0.50	(0.50 - 0.50) * 0.02 = 0.00 of FS	0.00 of FS = 0x0000 0000	0 * LSB = 0.0 volts

APPENDIX B: PIN-OUT DETAILS

Pin-out details (for reference) are shown below, with respect to DATAIO. Additional information on pin-outs can be found in the Motherboard Operational Manuals.

Module Signal (Ref Only)	DA-HI-VOLT (DA4)
DATIO1	
DATIO2	
DATIO3	SENSE-H_CH1
DATIO4	SENSE-L_CH1
DATIO5	OUT-H_CH1
DATIO6	OUT-L_CH1
DATIO7	EXT-SYNC_P*
DATIO8	EXT-SYNC_N*
DATIO9	SENSE-H_CH2
DATIO10	SENSE-L_CH2
DATIO11	OUT-H_CH2
DATIO12	OUT-L_CH2
DATIO13	
DATIO14	
DATIO15	SENSE-H_CH3
DATIO16	SENSE-L_CH3
DATIO17	OUT-H_CH3
DATIO18	OUT-L_CH3
DATIO19	
DATIO20	
DATIO21	SENSE-H_CH4
DATIO22	SENSE-L_CH4
DATIO23	OUT-H_CH4
DATIO24	OUT-L_CH4
DATIO25	
DATIO26	
DATIO27	
DATIO28	
DATIO29	
DATIO30	
DATIO31	
DATIO32	
DATIO33	
DATIO34	
DATIO35	
DATIO36	
DATIO37	
DATIO38	
DATIO39	
DATIO40	
N/A	

Notes

*	Pending external clock/trigger mode feature – contact factory for application use and availability

Status and Interrupts

MODULE MANUAL

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 below).

- *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.

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.

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.

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.

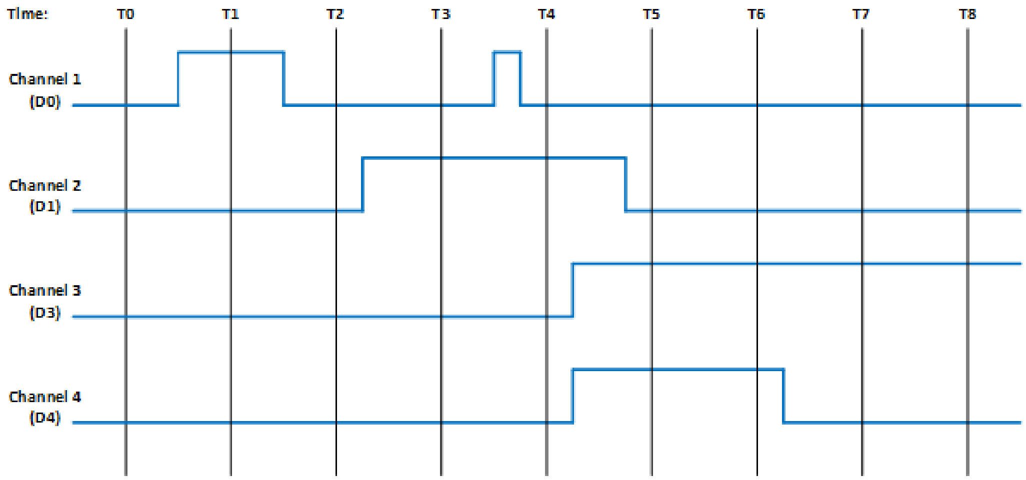


Figure 1. Example of Module's Channel-Mapped Dynamic and Latched Status States

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

Interrupt Examples

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

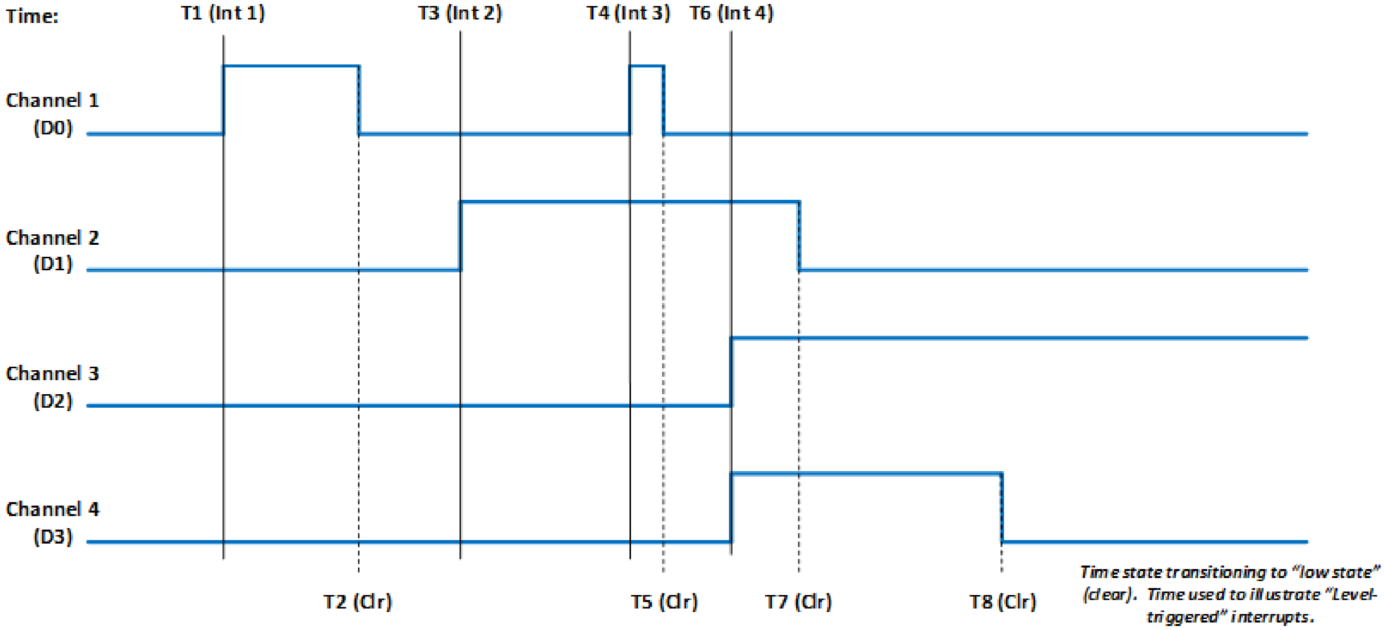


Figure 2. Illustration of Latched Status State for Module with 4-Channels with Interrupt Enabled

Time	Latched Status (Edge-Triggered – Clear Multi-Channel)		Latched Status (Edge-Triggered – Clear Single Channel)		Latched Status (Level-Triggered – Clear Multi-Channel)	
	Action	**Latched**	**Action**	**Latched**	**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*

<table>
<tr><th rowspan="2">Time</th><th colspan="2">Latched Status
(Edge-Triggered –
Clear Multi-Channel)</th><th colspan="2">Latched Status
(Edge-Triggered –
Clear Single Channel)</th><th colspan="2">Latched Status
(Level-Triggered –
Clear Multi-Channel)</th></tr>
<tr><th>Action</th><th>Latched</th><th>Action</th><th>Latched</th><th>Action</th><th>Latched</th></tr>
<tr><td rowspan="5">T6
(Int 4)</td><td>Interrupt Generated
Read Latched Registers</td><td>0xC</td><td>Interrupt Generated
Read Latched Registers</td><td>0xC</td><td>Interrupt Generated
Read Latched Registers</td><td>0xE</td></tr>
<tr><td>Write 0xC to Latched Register</td><td></td><td>Write 0x4 to Latched Register</td><td></td><td>Write 0xE to Latched Register</td><td></td></tr>
<tr><td rowspan="3"></td><td rowspan="3">0x0</td><td>Interrupt re-triggers
Write 0x8 to Latched Register</td><td>0x8</td><td>Interrupt re-triggers
Note, interrupt re-triggers after each clear and 0xE is reported in Latched Register until T7.</td><td>0xE</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">0x0</td><td>Interrupt re-triggers
Note, interrupt re-triggers after each clear and 0xC is reported in Latched Register until T8.</td><td>0xC</td></tr>
<tr><td>Interrupt re-triggers
Note, interrupt re-triggers after each clear and 0x4 is reported in Latched Register always.</td><td>0x4</td></tr>
</table>




User Watchdog Timer

MODULE MANUAL

USER WATCHDOG TIMER MODULE MANUAL

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.

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.

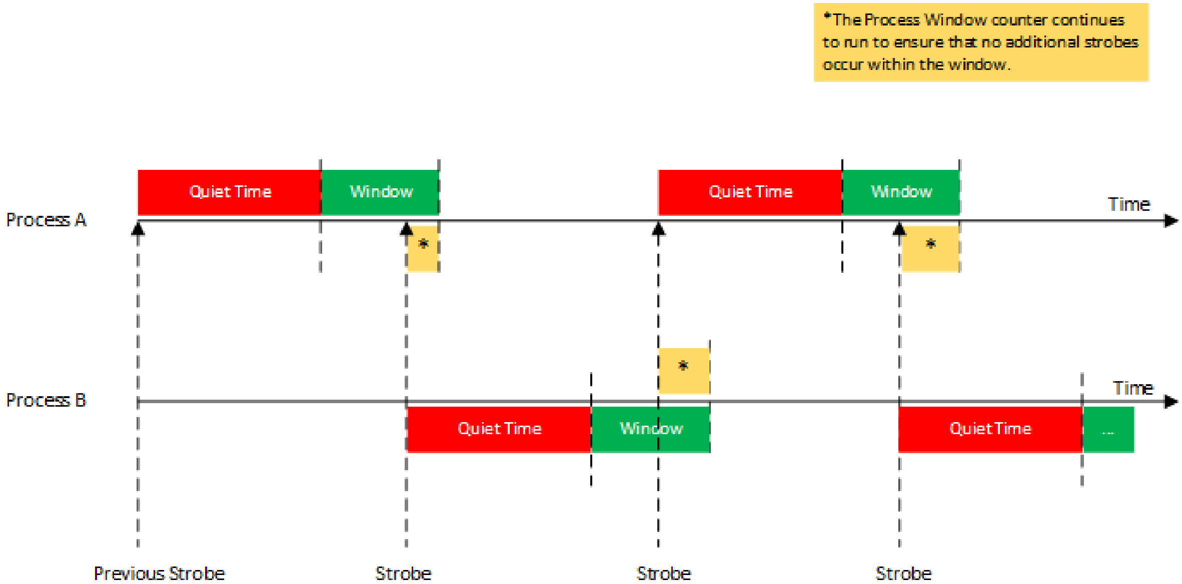


Figure 1. User Watchdog Timer Overview

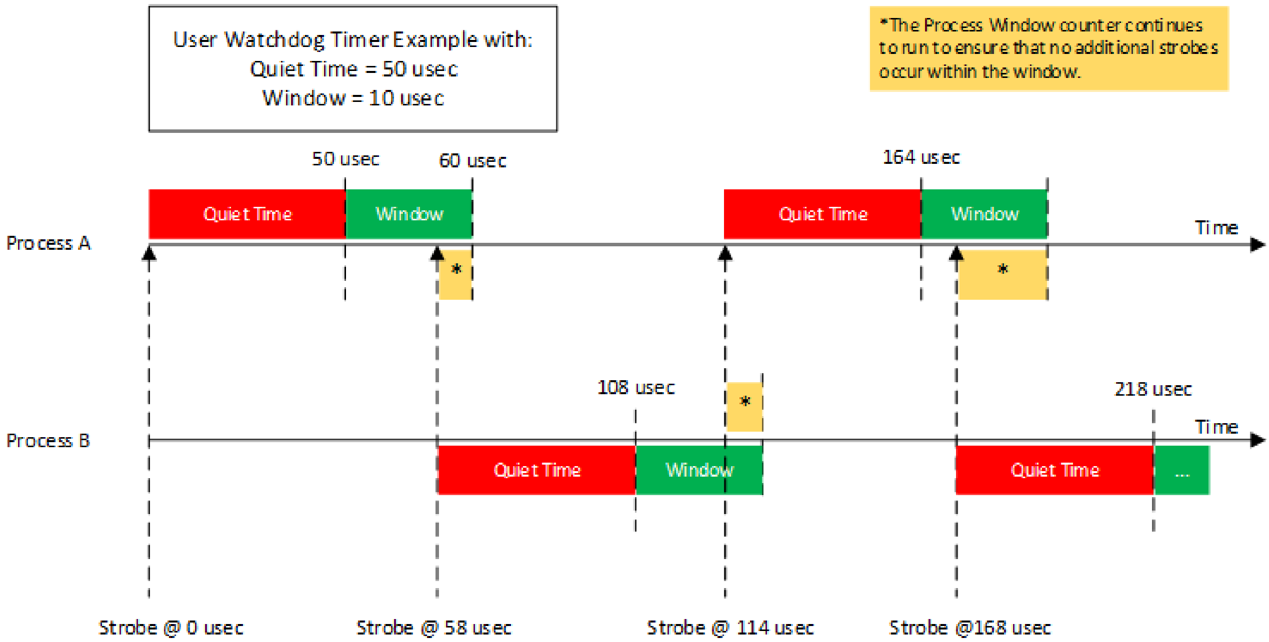


Figure 2. User Watchdog Timer Example

Figure 3 illustrates examples of User Watchdog Timer failures.

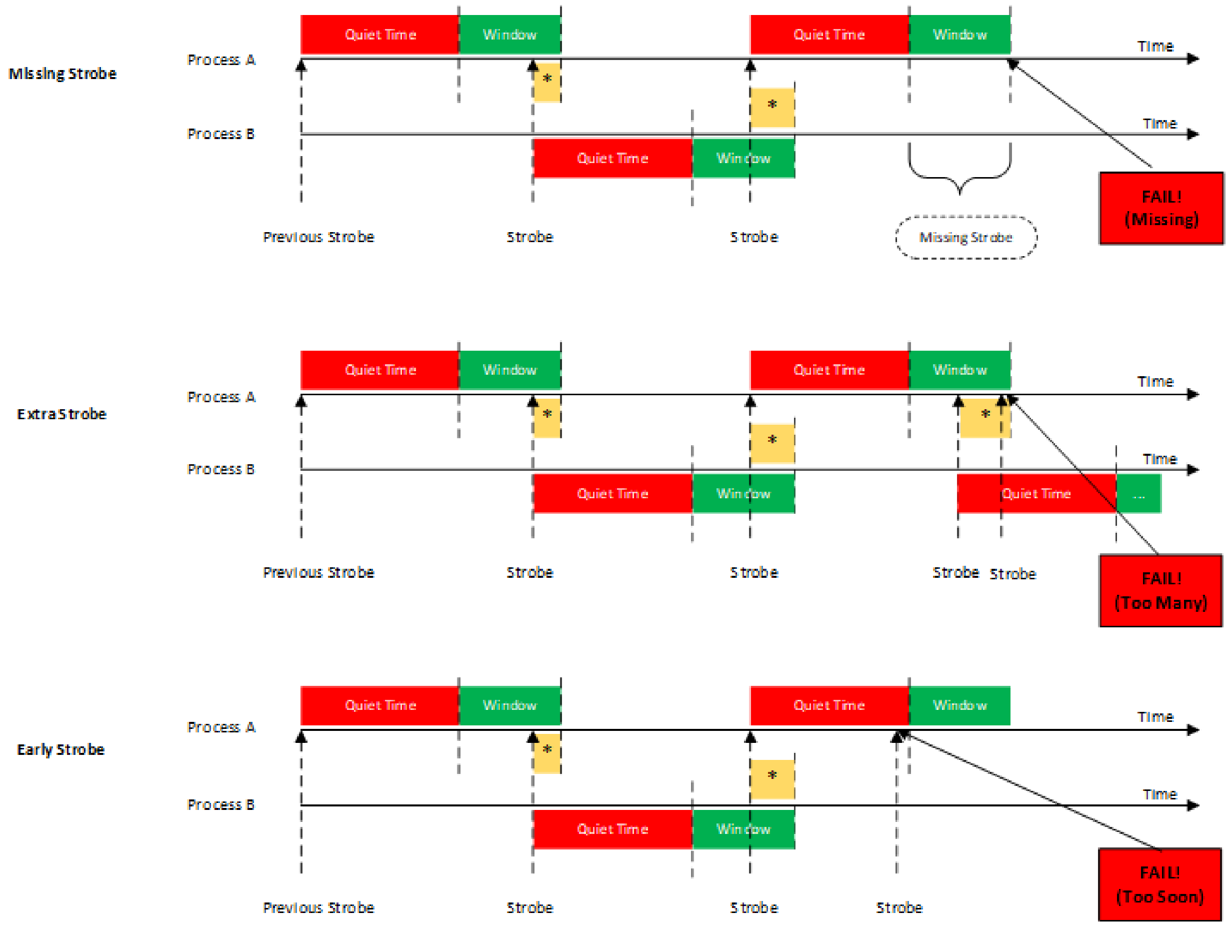


Figure 3. User Watchdog Timer Failures

Register Descriptions

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

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.

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.

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.

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.

Status and Interrupt

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

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

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).

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.

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

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').

User Watchdog Timer Registers

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

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

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
0x076C	***Module 2 Interrupt Vector 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
0x0B6C	***Module 4 Interrupt Vector 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
0x0F6C	***Module 6 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
0x086C	***Module 2 Interrupt Steering 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
0x0C6C	***Module 4 Interrupt Steering 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
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 Configurable Open Systems 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/faqs

Application Notes
http://www.naii.com/applicationnotes

Calibration and Repairs
http://www.naii.com/calibrationrepairs

Call Us
(631) 567-1100

Accelerate Your Time-to-Mission™