Using the FADC250 Module (V29 – 3/20/15)

1.1 Controlling the Module

Communication with the module is by standard VME bus protocols. All registers and memory locations are defined to be 4-byte entities. The VME slave module has three distinct address ranges.

A24 – The base address of this range is set by a 12-element DIP switch on the board. It occupies 4 Kbytes of VME address space, organized in 1 K 32-bit words. Relative to the base address, this space is divided as follows:

000-0FF – Register space to control and monitor the module (64 long words)

100-1FF – ADC processing registers (64 long words)

200-2FF – HITSUM processing registers (64 long words)

300-3FF – SCALER registers (64 long words)

400-4FF – SYSTEM TEST registers (64 long words)

500-FFF – Reserved (704 long words)

In addition to registers that are directly mapped to a VME address (Primary Address), the module supports Secondary Addressing in the A24 address space. These registers are accessed through an address mapping register (Secondary Address Register). Each secondary address is associated with a primary address. A Primary Address may have up to 64 K secondary addresses associated with it. A VME cycle loads the mapping register with data which is the internal (secondary) address of the target register. A VME cycle with the associated primary address accesses (read/write) the chosen internal register. Important registers are assigned primary addresses, allowing them to be directly accessible in a single VME cycle. Setup tables are assigned secondary addresses. This allows for a large internal address space, while maintaining a small VME footprint.

A32 - The base address of this range is programmed into register ADR32. It occupies 8 Mbytes of VME address space, organized in 2 M 32-bit words. A read of any address in this range will yield the next FADC data word from the module. Even though the module is logically a FIFO, the expanded address range allows the VME master to increment the address during block transfers. This address range can participate in single cycle, 32-bit block, and 64-bit block reads. The only valid write to this address range is the data value 0x80000000 which re-enables the module to generate interrupts (after one has occurred). The address range must be enabled by setting ADR32[0] = 1.

A32 - The lower and upper limits of this address range are programmed into register ADR_MB. This common address range for a set of FADC modules in the crate is used to implement the Multiblock protocol. By means of token passing FADC data may be read out from multiple FADC modules using a single logical block read. The board possessing the token will respond to a read cycle in this address range with the next FADC data word from that module. The token is passed along a private daisy chain line to the next module when it has transferred all data from a programmed number of events (register BLOCK SIZE). The address range must be enabled: set ADR_MB[0] = 1.

1.3 Module Registers

VERSION – board/firmware revision (0x0)

[7…0] – (R) – firmware revision

[15…8] – (R) – board revision

[31…16] – (R) – board type (“FADC”)

CSR – Control/Status (0x4)

0 – (R) – Event Accepted

1 – (R) – Block of Events Accepted

2 – (R) – Block of Events ready for readout

3 – (R) – BERR Status (1 = BERR asserted)

4 – (R) – Token Status (1 = module has token)

[5…10] – (reserved)

11 – (R) – Data FIFO Empty Flag Asserted

12 – (R) – Data FIFO Almost Empty Flag Asserted

13 – (R) – Data FIFO Half Full Flag Asserted

14 – (R) – Data FIFO Almost Full Flag Asserted

15 – (R) – Data FIFO Full Flag Asserted

16 – (R) – ADC Processing FPGA high temperature alarm flag

17 – (R) – CTRL FPGA high temperature alarm flag

[18…19] – (reserved)

20 – (W) – Pulse Soft **Trigger 2** (if CTRL[7] = 1 and CTRL[6..4] = 5)
(delayed **Trigger 1** follows; delay in TRIG21_DELAY register)
(R) – Trigger 2 -> Trigger 1 sequence active

21 – (W) – Pulse Clear Module – soft reset + clear data pipelines
(R) – Clear Module process active

22 – (W) – ENABLE SCALERS INTO DATA STREAM with FORCED BLOCK TRAILER INSERTION (write '1' to bits 22, 23)

23 – (W) – FORCE BLOCK TRAILER INSERTION – will be successful only if there are NO triggers waiting to be processed

24 – (R) – Last FORCE BLOCK TRAILER INSERTION Successful

25 – (R) – Last FORCE BLOCK TRAILER INSERTION Failed

26 – (R) – Local Bus Time Out – target AK or DK timed out (5 us);

27 – (R/W) – Local Bus Error – target protocol violation;
(write '1' clears latched bits 26, 27)

28 – (W) – Pulse Soft **Sync Reset** (if CTRL[11] = 1 and CTRL[10..8] = 6)

29 – (W) – Pulse Soft **Trigger 1** (if CTRL[7] = 1 and CTRL[6..4] = 6)

30 – (W) – Pulse Soft Reset – initialize counters, state machines, memory

31 – (W) – Pulse Hard Reset – initialize module to power-up state

CTRL1 – Control 1 (0x8)

[1…0] – (R/W) – Sampling Clock Source Select
0 = Internal Clock
1 = Front Panel connector
2 = P0 connector (VXS)
3 = P0 connector (VXS)

2 – (not used)

3 – (R/W) – Enable Internal Clock

[6…4] – (R/W) – Trigger Source Select

0 = Front Panel Connector (**Trigger 1**)
1 = Front Panel Connector (**Trigger 1**; synchronized)
2 = P0 Connector (VXS) (**Trigger1**, **Trigger 2**)
3 = P0 Connector (VXS) (**Trigger1**, **Trigger 2**; synchronized)
4 – (not used)
5 – Software Generated (**Trigger 2** + delayed **Trigger 1**)
6 = Software Generated (**Trigger 1**)
7 = Module Internal Logic

7 – (R/W) – Enable Soft Trigger

[10…8] – (R/W) – Sync Reset Source Select

0 = Front Panel Connector
1 = Front Panel Connector (synchronized)
2 = P0 Connector (VXS)
3 = P0 Connector (VXS) (synchronized)
4 – (not used)
5 – (not used)
6 = Software Generated
7 = no source

11 – (R/W) – Enable Soft Sync Reset

12 – (R/W) – Select Live Internal Trigger to Output (otherwise, Module Trigger).

13 – (R/W) – Enable Front Panel Trigger Output

14 – (R/W) – Enable P0 (VXS) Trigger Output

15 – (R/W) – Insert ADC parameter word into data stream. The data word appears as a block header continuation word and has the following format:

[31…29] – 0
[28…18] – PL (see ADC processing FPGA address map below)
[17…9] – NSB
[8…0] – NSA

16 – (R/W) – Suppress both trigger time words from data stream

17 – (R/W) – Suppress trigger time word 2 from data stream
(most significant bytes)

18 – (R/W) – Enable Event Level Interrupt

19 – (reserved)

20 – (R/W) – Enable BERR response

21 – (R/W) – Enable Multiblock protocol

22 – (R/W) – FIRST board in Multiblock system

23 – (R/W) – LAST board in Multiblock system

24 – (reserved)

25 – (R/W) – Enable Debug Mode

[26…27] – (reserved)

28 – (R/W) – Multiblock Token passed on P0

29 – (R/W) – Multiblock Token passed on P2

30 – (reserved)

31 – (R/W) – System Test Mode (0 = normal, 1 = test mode enabled)

CTRL2 – Control 2 (0xC)

0 – (R/W) – GO (allow data transfer from external FIFOs to input FIFOs)

1 – (R/W) – Enable **Trigger** (**1** & **2**) to Module (source = CTRL1[6…4])

2 – (R/W) – Enable **Sync Reset** to Module (source = CTRL1[10…8])

3 – (R/W) – Enable Internal Trigger Logic

4 – (R/W) – Enable Streaming mode (NO event build)

5 – (R/W) – Use pulse derived from leading edge of **Sync Reset** signal as module Sync Reset

[6…7] – (reserved)

8 – (R/W) – Enable Test Event Generation (for debug)

[9…15] – (reserved)

Bits 16 – 31 are functional only in Debug Mode (CTRL1[25] = 1)

16 – (reserved)

17 – (R/W) – Transfer data: build FIFO → output FIFO

[18…31] – (reserved)

BLOCK SIZE (0x10)

[15…0] – (R/W) – number of events in a BLOCK.
Stored Event Count ≥ BLOCK SIZE → CSR[3] = 1.

[31…16] – (reserved)

INTERRUPT (0x14)

[7…0] – (R/W) – Interrupt ID (vector)

[10…8] – (R/W) – Interrupt Level [2..0]. Valid values = 1,..,7.

11 - 15 – (reserved)

[20…16] – (R) – Geographic Address (slot number) in VME64x chassis.

21 – 22 – (reserved)

23 – (R) – Parity Error in Geographic Address.

24 – 31 – (reserved)

ADR32 – Address for data access (0x18)

0 – (R/W) – Enable 32-bit address decoding

1 – 6 – (reserved – read as 0)

[15…7] – (R/W) – Base Address for 32-bit addressing mode (8 Mbyte total)

ADR_MB – Multiblock Address for data access (0x1C)

0 – (R/W) – Enable Multiblock address decoding

1 – 6 – (reserved – read as 0)

[15…7] – (R/W) – Lower Limit address (ADR_MIN) for Multiblock access

16 – 22 – (reserved – read as 0)

[31…23] – (R/W) – Upper Limit address (ADR_MAX) for Multiblock access

The board that has the TOKEN will respond with data when the VME address satisfies the following condition:

$$ADR\_MIN \leq Address < ADR\_MAX.$$

SEC_ADR – Secondary Address (0x20)

[15…0] – (R/W) – Secondary Address for 24-bit addressing mode

16 – (R/W) – Enable auto-increment mode (secondary address increments by 1 after each access of the associated primary address)

DELAY – Trigger/Sync_Reset Delay (0x24) (NOT USED)

[21…16] – (R/W) – Sync reset delay

[5…0] – (R/W) – Trigger delay

INTERNAL TRIGGER CONTROL (0x28)

[23…16] – (R/W) – trigger width (4 ns per count; max = 1024 ns)

[11…0] – (R/W) – trigger hold off delay (4 ns per count; max = 16,384 ns)

RESET CONTROL (0x2C)

0 – (W) – Hard reset – Control FPGA

1 – (W) – Hard reset – ADC processing FPGA

[2…3] – (reserved)

4 – (W) – Soft reset – Control FPGA

5 – (W) – Soft reset – ADC processing FPGA

[6…7] – (reserved)

8 – (W) – Reset – ADC data FIFO

[9…10] – (reserved)

10 – (W) – Reset – HITSUM FIFO

11 – (W) – Reset – DAC (all channels)

12 – (W) – Reset – EXTERNAL RAM Read & Write Address Pointers

[13…15] – (reserved)

16 – (W) – Take Token – return token to 1st board of multiboard set

[17…31] – (reserved)

<u>TRIGGER COUNT</u> (0x30)

[31…0] – (R) – total trigger count

31 – (W) – reset count

<u>EVENT COUNT</u> (0x34)

[23…0] – (R) – number of events on board (non-zero → CSR[0] = 1).

[31…24] – (reserved)

<u>BLOCK COUNT</u> – (0x38)

[31…20] – reserved

[19…0] – (R) - number of event BLOCKS on board (non-zero → CSR[2] = 1).

<u>BLOCK FIFO COUNT</u> – (0x3C)

[31…6] – reserved

[5…0] – (R) - number of entries in BLOCK WORD COUNT FIFO

BLOCK WORD COUNT FIFO – (64 deep FIFO) (0x40)

[31…25] – reserved (read as ‘0’)

24 – (R) – count not valid (word count FIFO empty)

[23…20] – reserved (read as ‘0’)

[19…0] – (R) - number of words in next event BLOCK

INTERNAL TRIGGER COUNT (0x44)

[31…0] – (R) – internal live trigger count

31 – (W) – reset count

EXTERNAL RAM WORD COUNT (0x48)

[31…22] – reserved (read as ‘0’)

21 – (R) – RAM empty

20 – (R) – RAM full (1,048,576 eight byte words)

[19…0] – (R) – data word count (eight byte words)

DATA FLOW STATUS (0x4C) (for debug)

DAC 1_2 – DAC channels 1,2 (0x50)

31 – (R) – DAC channel 1 write timeout error

[30…28] – (reserved – read as 0)

[27...16] – (R/W) – DAC value channel 1

15 – (R) – DAC channel 2 write timeout error

[14…12] – (reserved – read as 0)

[11...0] – (R/W) – DAC value channel 2

DAC 3_4 – DAC channels 3,4 (0x54)

31 – (R) – DAC channel 3 write timeout error

[30…28] – (reserved – read as 0)

[27...16]– (R/W) – DAC value channel 3

15 – (R) – DAC channel 4 write timeout error

[14…12] – (reserved – read as 0)

[11...0] – (R/W) – DAC value channel 4

DAC 5_6 – DAC channels 5,6 (0x58)

31 – (R) – DAC channel 5 write timeout error

[30…28] – (reserved – read as 0)

[27...16]– (R/W) – DAC value channel 5

15 – (R) – DAC channel 6 write timeout error

[14…12] – (reserved – read as 0)

[11...0] – (R/W) – DAC value channel 6

DAC 7_8 – DAC channels 7,8 (0x5C)

31 – (R) – DAC channel 7 write timeout error

[30…28] – (reserved – read as 0)

[27...16]– (R/W) – DAC value channel 7

15 – (R) – DAC channel 8 write timeout error

[14…12] – (reserved – read as 0)

[11...0] – (R/W) – DAC value channel 8

DAC 9_10 – DAC channels 9,10 (0x60)

31 – (R) – DAC channel 9 write timeout error

[30…28] – (reserved – read as 0)

[27...16]– (R/W) – DAC value channel 9

15 – (R) – DAC channel 10 write timeout error

[14…12] – (reserved – read as 0)

[11...0] – (R/W) – DAC value channel 10

DAC 11_12 – DAC channels 11,12 (0x64)

31 – (R) – DAC channel 11 write timeout error

[30…28] – (reserved – read as 0)

[27...16]– (R/W) – DAC value channel 11

15 – (R) – DAC channel 12 write timeout error

[14…12] – (reserved – read as 0)

[11...0] – (R/W) – DAC value channel 12

DAC 13_14 – DAC channels 13,14 (0x68)

31 – (R) – DAC channel 13 write timeout error

[30…28] – (reserved – read as 0)

[27...18]– (R/W) – DAC value channel 13

15 – (R) – DAC channel 14 write timeout error

[14…12] – (reserved – read as 0)

[11...0] – (R/W) – DAC value channel 14

DAC 15_16 – DAC channels 15,16 (0x6C)

31 – (R) – DAC channel 15 write timeout error

[30…28] – (reserved – read as 0)

[27...16] – (R/W) – DAC value channel 15

15 – (R) – DAC channel 16 write timeout error

[14…12] – (reserved – read as 0)

[11...0] – (R/W) – DAC value channel 16

STATUS 1 – Input Buffer Status (0x70)

31 – (R) – data buffer ready for input

30 – (R) – data buffer input paused

29 – (R) – reserved (read as ‘0’)

28 – (R) – data buffer empty

27 – (R) – data buffer full

[26…16] – (R) – data buffer word count

[15…0] – (reserved)

STATUS 2 – Build Buffer Status (0x74)

[31…29] – reserved (read as ‘0’)

28 – (R) – data buffer ‘A’ empty

27 – (R) – data buffer ‘A’ full

[26…16] – (R) – data buffer ‘A’ word count

[15…13] – reserved (read as ‘0’)

12 – (R) – data buffer ‘B’ empty
11 – (R) – data buffer ‘B’ full

[10…0] – (R) – data buffer ‘B’ word count

STATUS 3 – Output Buffer Status (0x78)

[31…30] – reserved (read as ‘0’)

29 – (R) – data buffer ‘A’ empty

28 – (R) – data buffer ‘A’ full

[27…16] – (R) – data buffer ‘A’ word count

[15…14] – reserved (read as ‘0’)

13 – (R) – data buffer ‘B’ empty

12 – (R) – data buffer ‘B’ full

[11…0] – (R) – data buffer ‘B’ word count

STATUS 4 – (spare) (0x7C)

[31…0] – reserved

AUXILIARY 1 – (spare) (0x80)

[31…0] – reserved

TRIGGER CONTROL (0x84) – Under normal conditions, a trigger received by the module is counted and sent to the ADC Processing FPGA. A trigger is considered acknowledged when data associated with it has been transferred to the Control FPGA. To ensure that data buffers in the ADC Processing FPGA are not overrun, the number of unacknowledged triggers is continuously compared to levels (MAX1, MAX2) set by the user. These levels are based on the parameters (processing mode, # samples in window) loaded into the ADC Processing FPGA. Two module based methods to avert buffer overflow and the resulting data corruption are available to the user.

In the first method, module BUSY is asserted when the number of unacknowledged triggers ≥ MAX1. BUSY propagates back to the Trigger Supervisor. The Trigger Supervisor does not deliver triggers to the system while BUSY is asserted. Level MAX1 is based on the BUSY progagation delay, ADC Processing parameters, and the minimum trigger interval parameter of the Trigger Supervisor.
In the second method, triggers received by the module are not sent to the ADC Processing FPGA when the number of unacknowledged triggers ≥ MAX2. Level MAX2 is based on the ADC Processing parameters. Not sending triggers to the FPGA Processing FPGA can cause a loss of synchronization among system components, but will prevent data corruption within the module. (See Auxiliary Scaler 2 for a count of lost triggers.)
Both methods can be applied together, with MAX2 ≥ MAX1.

31 – (R/W) – enable trigger stop when number of unacknowledged triggers ≥ MAX2

[23...16] – (R/W) – level MAX2

15 – (R/W) – enable module busy assertion when number of unacknowledged triggers ≥ MAX1

[7...0] – (R/W) – level MAX1

TRIG21 DELAY (0x88)

[31…12] – reserved

[11…0] – (R/W) – Delay from soft TRIG2 to generated TRIG1 (4 ns/count)

RAM Address Register (0x8C) – The RAM is organized as two 36-bit words with a common address. Auxiliary VME access (R/W) to the RAM is provided through a pair of 32 bit data registers (RAM 1, RAM 2). Note that bits 35 – 32 of each RAM word are not accessible through VME. During data flow operations, these bits carry event marker tags (header, trailer).

31 – increment address after access (R/W) of RAM 1 Data Register

30 – increment address after access (R/W) of RAM 2 Data Register

[29…21] – reserved (read as 0)

[19…0] – RAM address

RAM 1 Data Register (0x90)

[31…0] – RAM data word bits 67 – 36 (32 bits)

RAM 2 Data Register (0x94)

[31…0] – RAM data word bits 31 – 0 (32 bits)

(PROM Registers 1 and 2 are used for FPGA configuration over VME.)

PROM Register 1 (0x98)

31 – READY – (R) – configuration state machine is available to accept command (i.e. no configuration process is currently executing).

[30…8] – reserved (read as 0)

[7…0] – configuration OPCODE

PROM Register 2 (0x9C)

[31…0] – PROM ID – (R) response to specific OPCODE write to PROM reg 1.

BERR Module Count (0xA0)

[31…0] – BERR count (driven by module to terminate data transmission)

BERR Total Count (0xA4)

[31…0] – BERR count (as detected on bus)

Auxiliary Scaler 1 (0xA8)

[31…0] – Total word count from ADC Processing FPGA

Auxiliary Scaler 2 (0xAC)

[31…0] – Count of triggers lost because number of unacknowledged triggers ≥ MAX2. (See TRIGGER CONTROL register.)

Auxiliary Scaler 3 (0xB0)

[31…0] – Event header word count from ADC Processing FPGA

TRIGGER 2 SCALER (0xB4)

[31…0] – (R) – **Trigger 2** count

31 – (W) – write '1' to reset count

Auxiliary Scaler 5 (0xB8)

[31…0] – Event trailer word count from ADC Processing FPGA

SYNC RESET SCALER (0xBC)

[31…0] – (R) – **Sync Reset** count

31 – (W) – write '1' to reset count

Module Busy Level (0xC0)

[31] – Force module busy

[30…20] – reserved

[19…0] – Busy Level (eight-byte words)
(External RAM word count > Busy Level → module busy = 1)

NOTE: At the system level, an asserted module busy signal should be used to prevent further triggers from being sent to the modules. By setting the Busy Level well below the memory capacity (1,048,576 eight-byte words), triggers already in the distribution pipeline can still be accepted by the module.
To prevent data corruption in the module if this global trigger control is NOT in place, the module itself will BLOCK input triggers when the number of 8-byte words in memory is within 12K (1.2%) of memory capacity (i.e. 1,036,288).
When global trigger control IS in place, it is important to set Busy Level significantly LESS than 1,036,288 so that local trigger blocking will NEVER occur. (Local trigger blocking will result in a loss of system synchronization.) The safe maximum Busy Level depends on data size per trigger (i.e. mode of

ADC processing), but a Busy Level equal to ~87.5% of memory capacity (i.e. 917,504 = 0xE0000) should be adequate for all circumstances.

Generate Event Header Word (0xC4) (for debug)

[31…0] – (W) – Event Header Word

Generate Event Data Word (0xC8) (for debug)

[31...0] – (W) – Event Data Word

Generate Event Trailer Word (0xCC) (for debug)

[31...0] – (W) – Event Trailer Word

MGT STATUS (0xD0)

0 – (R) – lane 1 up (GTX1)

1 – (R) – lane 2 up (GTX1)

2 – (R) – channel up (GTX1)

3 – (R) – hard error (GTX1)

4 – (R) – soft error (GTX1)

5 – (R) – lane 1 up (GTX2)

6 – (R) – lane 2 up (GTX2)

7 – (R) – channel up (GTX2)

8 – (R) – hard error (GTX2)

9 – (R) – soft error (GTX2)

10 – (R) – SUM DATA VALID

11 – (R) – MGT RESET ASSERTED

[31...12] – (R) - Reserved

MGT CONTROL (0xD4)

0 – **RELEASE MGT RESET** (0 = reset MGT, 1 = release reset)

1 – Data Type to CTP (0 = counting sequence, 1 = front-end data)

2 – Enable Data Alignment on Sync Reset occurrence
[31...3] – Reserved

RESERVED (2 registers) (0xD8 – 0xDC)

SCALER CONTROL (0xE0) – See SCALERS (0x300 – 0x340)

0 – (R/W) – Enable all scalers to count (1 = enable, 0 = disable)

1 – (W) – Latch all scalers. Write ‘1’ to simultaneously transfer all 17 scaler counts to registers for readout.

2 – (W) – Reset all scalers. Write ‘1’ to simultaneously reset all 17 scaler counts to zero.

[3 – 31] – (reserved)

BOARD SERIAL NUMBER 0 (0xE4)

[31…24] – (R) – board serial number byte 0

[23…16] – (R) – board serial number byte 1

[15…8] – (R) – board serial number byte 2

[7…0] – (R) – board serial number byte 3

BOARD SERIAL NUMBER 1 (0xE8)

[31…24] – (R) – board serial number byte 4

[23…16] – (R) – board serial number byte 5

[15…8] – (R) – board serial number byte 6

[7…0] – (R) – board serial number byte 7

BOARD SERIAL NUMBER 2 (0xEC)

[31…24] – (R) – board serial number byte 8

[23…16] – (R) – board serial number byte 9

[15…8] – (R) – board serial number byte 10

[7…0] – (R) – board serial number byte 11

SCALER INSERTION INTERVAL (0xF0) - Data from the SCALERS defined below (0x300 – 0x340) may be inserted into the readout data stream at regular event count intervals. The interval is specified in multiples of the event BLOCK SIZE. When the interval is ZERO (the default condition), there is NO insertion of scaler data into the data stream. When programmed for a non-zero interval, the current scaler values are appended to the last event of the appropriate BLOCK of events. The current Trigger 1 count is also inserted as the 18^{th} scaler. Note that the scalers are **NOT** reset after their values are captured.

Example: Interval = 10 means that every 10^{th} block of events will have the integrated scaler data appended to it.

(See the document FADC V2 Data Format for information on identifying scaler data words in an event.)

The scalers may ALSO be inserted into the data stream when a FORCE BLOCK TRAILER is done by the user. A simultaneous write of '1' to bit 22 and bit 23 of the CSR (0x4) accomplishes this. The scaler values are those at the time of the last trigger's occurrence.

[15…0] - (R/W) – N (in BLOCKS of events); every N^{th} block of events has integrated scaler data appended to the last event in the block.

[31…16] – (reserved)

SUM THRESHOLD (History buffer) (0xF4)

31 – (R) – sum data READY for readout if value = '1'

[30…16] – (R/W) – reserved (read as '0')

[15…0] – (R/W) – sum threshold value for data capture

SUM DATA (History buffer) (0xF8)

31 – (W) – writing a ‘1’ARMs History buffer for data capture

[30…16] – (R) – reserved (read as ‘0’)

[15…0] – (R) – sum data sample

SYSTEM MONITOR (0xFC)

[31…22] – (R) – FPGA auxiliary voltage (2.5V) (common)

```
vaux = (((float)((reg_value >> 22) & 0x3FF))/1024.0) * 3.0;
```

21 – (R) – reserved (read as ‘0’)

[20…11] – (R) – FPGA core voltage (1.0V) (common)

```
vint = (((float)((reg_value >> 11) & 0x3FF))/1024.0) * 3.0;
```

10 – (R) – reserved (read as ‘0’)

[9…0] – (R) – CTRL_FPGA temperature (^{o}C)

```
temp_ctrl = (((float)(reg_value & 0x3FF)) * 503.975/1024.0) - 273.15;
```

SCALER Registers (0x300 – 0x340) (R)

SCALER[0] – (0x300) - input channel 0 count

SCALER[1] – (0x304) - input channel 1 count

SCALER[2] – (0x308) - input channel 2 count

SCALER[3] – (0x30C) - input channel 3 count

SCALER[4] – (0x310) - input channel 4 count

SCALER[5] – (0x314) - input channel 5 count

SCALER[6] – (0x318) - input channel 6 count

SCALER[7] – (0x31C) - input channel 7 count

SCALER[8] – (0x320) - input channel 8 count

SCALER[9] – (0x324) - input channel 9 count

SCALER[10] – (0x328) - input channel 10 count

SCALER[11] – (0x32C) - input channel 11 count

SCALER[12] – (0x330) - input channel 12 count

SCALER[13] – (0x334) - input channel 13 count

SCALER[14] – (0x338) - input channel 14 count

SCALER[15] – (0x33C) - input channel 15 count

TIME COUNT – (0x340) - timer (each count represents 2048 ns)

<u>System Test Registers</u> (0x400 – 0x410)

<u>TEST BIT REGISTER</u> (0x400)

0 – (R/W) – trigger_out_p0 (1 = asserted, 0 = not asserted)

1 – (R/W) – busy_out_p0 (1 = asserted, 0 = not asserted)

2 – (R/W) – sdlink_out_p0 (1 = asserted, 0 = not asserted)

3 – (R/W) – token_out_p0 (1 = asserted, 0 = not asserted)

[4 – 7] – (R/W) – spare out test bits

8 – (R) – status_b_in_p0 state (1 = asserted, 0 = not asserted)

9 – (R) – token_in_p0 state (1 = asserted, 0 = not asserted)

[10 - 14] – (R) – reserved (read as ‘0’)

15 – (R) – clock_250 counter status (1 = counting, 0 = not counting)

[16 - 31] – (R) – reserved (read as ‘0’)

<u>CLOCK_250 COUNT REGISTER</u> (0x404)

0 – (W) – Write ‘0’ resets the counter. Write ‘1’ initiates 20us counting interval.

[31 - 0] – (R) – CLK_250 counter value. (Should be 5000 after count interval.)

<u>SYNC_IN_P0 COUNT REGISTER</u> (0x408)

0 – (W) – Write ‘0’ resets the counter.

[31 - 0] – (R) – SYNC_IN_P0 counter value.

<u>TRIG1_IN_P0 COUNT REGISTER</u> (0x40C)

0 – (W) – Write ‘0’ resets the counter.

[31 - 0] – (R) – TRIG1_IN_P0 counter value.

TRIG2_IN_P0 COUNT REGISTER (0x410)

0 – (W) – Write ‘0’ resets the counter.

[31 - 0] – (R) – TRIG2_IN_P0 counter value.

ADC PROCESSING FPGA ADDRESS MAP:

Control Bus Memory Map for FADC FPGA

Name **[VME ADDRESS]**	Width (Bits)	Quantity	Access	Primary Address (Secondary Address)	Function
STATUS0 **[0x100]**	16	1	R	0x0000 (---)	Bits 14 to 0: Code Version Bit 15: 1= Command can be sent to AD9230
STATUS1 **[0x104]**	16	1	R	0x0001 (---)	TRIGGER NUMBER BIT 15 to 0
STATUS2 **[0x108]**	16	1	R	0x0002 (---)	Monitor ADC Data for Debugging Purpose
CONFIG 1 **[0x10C]**	16	1	R/W	0x0003 (---)	Bit 0-2 (process mode): 000 → Select Mode 0 001 → Select Mode 1 010 → Select Mode 2 011 → Select Mode 3 111 → Run Mode 0 then Mode 3 for each trigger Bit 3: 1:Run Bit 5-4 : Number of Pulses in Mode 1 and 2 Bit 7: Test Mode (play Back). Bit 11 to 8: Select which ADC Data channel to be read at Status 2.
CONFIG 2 **[0x110]**	16		R/W	0x0004 (---)	When 1 ADC values = 0 Bit 0 → ADC 0 Bit 1 → ADC 1 Bit 2 → ADC 2 Bit 3 → ADC 3 Bit 4 → ADC 4 Bit 5 → ADC 5 Bit 6 → ADC 6 Bit 7 → ADC 7

					Bit 8 → ADC 8 Bit 9 → ADC 9 Bit 10→ ADC 10 Bit 11→ ADC 11 Bit 12→ ADC 12 Bit 13→ ADC 13 Bit 14→ ADC 14 Bit 15→ ADC 15
CONFIG 4 **[0x114]**	16	1		0x0005 (---)	7 => rising edge write to AD9230 ADC 6 => 1 write to all ADC 5 => 0 write to AD9230 1 read from AD9230 3..0 => Select ADC to write to
CONFIG 5 **[0x118]**	16	1		0x0006 (---)	15..8 => Registers inside AD9230 7..0 => Data to write to register.
PTW **[0x11C]**	9	1	R/W	0x0007 (---)	Number of ADC sample to include in trigger window. PTW = Trigger Window (ns) * 250 MHz. **Minimum is 6**. **Always report Even Number. For odd PTW number, discard the last sample reported.**
PL **[0x120]**	11	1		0x0008 (---)	Number of sample back from trigger point. PL = Trigger Window(ns) * 250MHz
NSB **[0x124]**	13	1		0x0009 (---)	8..0: Read Back Path NSB Number of sample before trigger point to include in data processing. This include the trigger Point. **Minimum is 2 in all modes.** 12..9: Trigger Path NSB

NSA **[0x128]**	15	1		0x000A (---)	8..0: Read Back Path NSA Number of sample after trigger point to include in data processing. **Minimum is (6 in mode 2)and (3 in mode 0 and 1). Number of sample report is 1 more for odd and 2 more for even NSA number.** 14..9: Trigger Path NSA
TET **[0x12C – 0x148]**	12	16		0x000B - 0x001A (---)	Trigger Threshold.
PTW DAT BUF LAST ADR **[0x14C]**	12	1		0x001B (---)	Last Address of the Secondary Buffer. See calculation below
PTW MAX BUF **[0x150]**	8	1		0x001C (---)	The maximum number of unprocessed PTW blocks that can be stored in Secondary Buffer. See Calculation below.
Test Waveform **[0x154]**	16	1		0x001D (---)	Write to PPG. Read should immediately follow write.
ADC0 Pedestal Subtract **[0x158]**	16	1	R/W	0x001E	Subtract from ADC0 Count before Summing
ADC1 Pedestal Subtract **[0x15C]**	16	1	R/W	0x001F	Subtract from ADC1 Count before Summing
ADC2 Pedestal Subtract **[0x160]**	16	1	R/W	0x0020	Subtract from ADC2 Count before Summing
ADC3 Pedestal Subtract **[0x164]**	16	1	R/W	0x0021	Subtract from ADC3 Count before Summing
ADC4 Pedestal Subtract **[0x168]**	16	1	R/W	0x0022	Subtract from ADC4 Count before Summing

ADC5 Pedestal Subtract **[0x16C]**	16	1	R/W	0x0023	Subtract from ADC5 Count before Summing
ADC6 Pedestal Subtract **[0x170]**	16	1	R/W	0x0024	Subtract from ADC6 Count before Summing
ADC7 Pedestal Subtract **[0x174]**	16	1	R/W	0x0025	Subtract from ADC7 Count before Summing
ADC8 Pedestal Subtract **[0x178]**	16	1	R/W	0x0026	Subtract from ADC8 Count before Summing
ADC9 Pedestal Subtract **[0x17C]**	16	1	R/W	0x0027	Subtract from ADC9 Count before Summing
ADC10 Pedestal Subtract **[0x180]**	16	1	R/W	0x0028	Subtract from ADC10 Count before Summing
ADC11 Pedestal Subtract **[0x184]**	16	1	R/W	0x0029	Subtract from ADC11 Count before Summing
ADC12 Pedestal Subtract **[0x188]**	16	1	R/W	0x002A	Subtract from ADC12 Count before Summing
ADC13 Pedestal Subtract **[0x18C]**	16	1	R/W	0x002B	Subtract from ADC13 Count before Summing
ADC14 Pedestal Subtract **[0x190]**	16	1	R/W	0x002C	Subtract from ADC14 Count before Summing
ADC15 Pedestal Subtract **[0x194]**	16	1	R/W	0x002D	Subtract from ADC15 Count before Summing
CONFIG 3 **[0x198]**	16	1	R/W	0x002E	(11..0) Trigger Path Processing Threshold
STATUS3 **[0x19C]**	16	1	R/W	0x002F	FPGA core temperature (^{o}C) See calculation below.

```
temp = (((float)(status3 & 0x3FF)) * 503.975/1024.0) - 273.15;
```

PTW MAX BUF = INT(2016 / (PTW + 8))

Where:

2016 → Number of address of Secondary Buffer

PTW → Trigger Window width in nanoseconds

PTW DAT BUF LAST ADR = PTW MAX BUF * (PTW + 8) - 1;

Where:

6 → 4 address for Time Stamp and 2 address for Trigger Number

NumberOfBytePerTrigger → PTW * 250 MHz.

TrIgger Path NSB + Trigger Path NSA has to be less then 63.

NOTE 1: Trigger Energy Threshold (TET)

0x12C – Channel 1 & Channel 2

[31…28] – not used
[27…16] – channel 1 threshold
[15…12] – not used
[27…16] – channel 2 threshold

0x130 – Channel 3 & Channel 4

[31…28] – not used
[27…16] – channel 3 threshold
[15…12] – not used
[27…16] – channel 4 threshold

. . . .

0x148 – Channel 15 & Channel 16

[31…28] – not used
[27…16] – channel 15 threshold
[15…12] – not used
[27…16] – channel 16 threshold

HITSUM PROCESSING ADDRESS MAP:

Name **[VME ADDRESS]**	Width (Bits)	Quantity	Access	Secondary Address	Function
STATUS **[0x200]**	16	1	R	(---)	(not used)
CONFIGURATION **[0x204]**	16	1	R/W	(---)	00→ Table mode 10→ Window mode 01→ Boolean Overlap 11→ undefined Bit 2: 0→ HIT trigger, HIT pattern to FIFO 1→ SUM trigger, SUM value to FIFO Bit4: 0→ Enable Table overlap and Trigger mode 1→ Disable Table overlap and Trigger mode. Read back hit pattern selection table.
HITBITS_WIDTH **[0x208]**	16	16	R/W	(0 – 0xF)	Hit Bits One Shot Pulse Width and Delay for input channels 0-15. Bits 7-0: width Bits 12-8: delay (Actual width and delay is one clock period longer.)
Live Trig Delay **[0x20C]**	16	1	R/W	(---)	(not used)
Live Trig WIDTH **[0x210]**	16	1	R/W	(---)	Pulse width of Live Trig Output. (Actual width is 2 clock periods longer.)
TRIGGER HITBITS **[0x214]**	16	1	R/W	(---)	In Window Mode selects Hit Bits that can generate window.

WINDOW WIDTH **[0x218]**	16	1	R/W	(---)	In Window Mode selects the duration of window. Width is 2 clock periods longer.
BOOLEAN OVERLAP QUALIFIED BITS **[0x21C]**	16	1	R/W	(---)	In Boolean Overlap Mode select required Hit Bits to be active for trigger to result.
HIT PATTERN SELECTION TABLE DATA **[0x220]**	16	65536	R/W	(0 – 0xFFFF)	65536 x 1 Hit Pattern Selection Table. Each word contains data for 1 location. The address can be automatically incremented. (See Secondary Address Register.)
SUM/HITBIT FIFO **[0x224]**	16	1	R	(---)	Bits 15-0: HIT pattern or SUM value (depending on mode selected).
SUM Threshold **[0x228]**	16	1	R/W	(---)	SUM threshold register. SUM > register value results in trigger when SUM mode is selected.