# Description and Instructions for the Firmware FADC250 V3Boards Version 0x4100

**May 2020**

**Hai Dong**

# 0.0 Differences between FADCV2 and FADCV3 Boards:

This section highlights the differences between FADCV2 and FADCV3 boards. Some of these differences necessitate changes in firmware. The differences are:

1) There is only 1 FPGA (XCKU060-2FFVA1517E) in FADCV3 board.
2) Pedestal (Bias) DAC for FADCV3.
   a) Requires a read command from Host to read back its registers.
   b) Requires to be initialize at power up. This is done automatically by the firmware at power up. The firmware also accommodate user to re-initialize the IC.
   c) The Bias DAC count has different effect on the ADC count. See Bias DAC Count vs ADC Count Table below
3) ADC ADS4229 for FADCV3
   a) ADS4229 has 2Vpp, instead of 1.2V as in FADC, hence the input voltage is -.8, -1, -2 Volt.
   b) Test patterns are: all 0, all 1, toggle pattern, digital RAMP.
   c) Data format is either Offset binary. Firmware initializes as Offset Binary.
   d) Does not have option to invert output bits. This is done in firmware.
4) Two external SRAM that are used to store read back data when the FIFO in FPGA is overflowed. The SRAM are replaced by DDR.
5) The FIFO connected between the 2 FPGAs in FADCV2 is moved inside the FPGA.

# 0.1 Differences between FADCV2 and FADCV3 firmware:

This section highlights the differences between FADCV2 and FADCV3 boards. Some of these differences necessitate changes in firmware. The differences are:

1) Pedestal (Bias) DAC
   a) Mapping VME's registers to DAC control lines:
      i) Reset Control (0x2C, bit 11) → Reset DAC (all channels)
      ii) DAC_CSR (Control/Status) (0x50:
         A. 31 (W) → DAC initialize
         B. 30 (R) → DAC Initialization complete
         C. 29 (W) → Clear latched stratus bits 18,19
         D. 19 (R) → DAC timeout on access since last status. Clear with bit 29.
         E. 18 (R) → DAC not ready on access since last status. Clear with bit 29.
         F. 17 (R) → Last DAC access successful
         G. 16 (R) → DAC ready current state
         H. 13..0 (R/W) DAC channel (0-15) to access with DAC DATA register
      iii) DAC DATA (0x54) See DAC AD5391 spec for explanation
         A. 21 (R) → DAC register A or B is read from
         B. 20 (R) → 1 DAC read is invalid
         C. 19 (R) → 0 DAC read
         D. 18 (R) → 0 DAC read
         E. 17..14 (R) → DAC channel accessed
         F. 13..12 (R) → DAC REG1, REG0
         G. 11..0 (R/W) → DAC data to write or read
      iv) Ask Ed Jastrzembski for C routines to read write DAC.
2) ADC ADS4229
   a) Mapping VME's registers to DAC control lines:
      i) Processing Config 4 (0x114) (W/R)
         A. 7 → rising edge execute ADC command.
         B. 6 → 1 write or read to all ADC. 0 write to or read from ADC specified by bit 3…0
         C. 5 → 0 write to ADC. 1 read from ADC
         D. 3..0 → Select ADC to write to when bit 6 is 0. Or read from regardless of bit 6
      ii) Processing Config 5 (0x118) (W/R)
         A. 15..8 → Select ADC's register to write to or read from
         B. 7..0 → Data to write to register
      iii) Processing Status0 (0x100) (R)

        A. 15 →1 command can be sent to ADC
   b) To invert ADC data, set Bit 6 of Config 1. This is useful when processing positive going puls
   c) Data format for host VME command to change ADC functionality is not changed. But the meaning of the bits are different. Look at TI ADS4229 data sheet for ADC registers' functions
3) FPGA system monitor Only provide alarm bits for :
   a) Mapping VME's registers to temperature read back:
      i) Processing Status4 (0x130) (R)
         A. 14 → Vcc Aux is not within range
         B. 13 → Vcc Int is not within range
         C. 12 → FPGA temp is above 85 degree C
         D. FPGA temp is above 125 degree C.
   b) Reset the FPGA when temperature is 85 degree C or above.
4) FPGA's Config ROM
   a) See VME_Program_FADCV3_Config_ROM.docx located at M:\FE\fADC-250V3\C_test_code_CPP2022
5) VME reboots FPGA from Config ROM
   a) Kintex Ultra-Scale Kintex FPGA has internal circuit to reboot the FPGA.
   b) See How_To_Reboot_FPGA_From_VME.docx located at M:\FE\fADC-250V3\Document
   c) VHDL code sends rebooting commands to Ultra-Scale Kintex FPGA's ICAP3 when a VME write a 1 to bit 11 of CONFIG ROM CONTROL 0 (VME base address (U104, U105 dip switches) + 0x58)

## Bias DAC Count vs ADC Count Table:

ADC counts is captured without input at LEMO connector

dac step 200

| Bias DAC Counts | CH0 | CH1 | CH2 | CH3 | CH4 | CH5 | CH6 | CH7 | CH8 | CH9 | CH10 | CH11 | CH12 | CH13 | CH14 | CH15 | Average | Max | Min | Difference | % Difference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 0 | 0.00 |
| 200 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 0 | 0.00 |
| 400 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 0 | 0.00 |
| 600 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 0 | 0.00 |
| 800 | 4095 | 4095 | 4093 | 4095 | 4095 | 4095 | 4079 | 4095 | 4095 | 4095 | 4088 | 4095 | 4087 | 4095 | 4084 | 4072 | 4090 | 4095 | 4072 | 23 | 0.56 |
| 1000 | 3765 | 3779 | 3736 | 3765 | 3765 | 3771 | 3723 | 3753 | 3774 | 3772 | 3731 | 3747 | 3729 | 3764 | 3729 | 3722 | 3751 | 3779 | 3722 | 57 | 1.52 |
| 1200 | 3398 | 3409 | 3375 | 3401 | 3400 | 3404 | 3363 | 3394 | 3403 | 3407 | 3370 | 3386 | 3368 | 3401 | 3367 | 3365 | 3388 | 3409 | 3363 | 46 | 1.36 |
| 1400 | 3034 | 3040 | 3016 | 3036 | 3037 | 3040 | 3007 | 3034 | 3037 | 3047 | 3012 | 3026 | 3006 | 3039 | 3008 | 3009 | 3026 | 3047 | 3006 | 41 | 1.35 |
| 1600 | 2672 | 2674 | 2658 | 2674 | 2679 | 2676 | 2649 | 2678 | 2671 | 2684 | 2657 | 2667 | 2649 | 2679 | 2651 | 2659 | 2667 | 2684 | 2649 | 35 | 1.31 |
| 1800 | 2310 | 2307 | 2302 | 2312 | 2318 | 2314 | 2294 | 2319 | 2306 | 2323 | 2299 | 2305 | 2289 | 2319 | 2292 | 2303 | 2307 | 2323 | 2289 | 34 | 1.47 |
| 2000 | 1949 | 1942 | 1944 | 1950 | 1958 | 1951 | 1939 | 1963 | 1941 | 1962 | 1944 | 1946 | 1931 | 1960 | 1936 | 1952 | 1948 | 1963 | 1931 | 32 | 1.64 |
| 2200 | 1586 | 1573 | 1587 | 1587 | 1598 | 1587 | 1583 | 1605 | 1575 | 1598 | 1588 | 1585 | 1572 | 1599 | 1578 | 1598 | 1587 | 1605 | 1572 | 33 | 2.08 |
| 2400 | 1226 | 1207 | 1231 | 1224 | 1237 | 1222 | 1227 | 1248 | 1210 | 1238 | 1230 | 1227 | 1214 | 1240 | 1222 | 1247 | 1228 | 1248 | 1207 | 41 | 3.34 |
| 2600 | 862 | 838 | 872 | 859 | 876 | 858 | 869 | 890 | 843 | 876 | 872 | 865 | 853 | 878 | 863 | 892 | 866 | 892 | 838 | 54 | 6.24 |
| 2800 | 499 | 471 | 513 | 497 | 516 | 493 | 513 | 531 | 476 | 512 | 515 | 505 | 494 | 517 | 505 | 539 | 506 | 539 | 471 | 68 | 13.44 |
| 3000 | 133 | 100 | 154 | 132 | 152 | 127 | 153 | 171 | 107 | 149 | 157 | 143 | 133 | 156 | 146 | 185 | 143 | 185 | 100 | 85 | 59.44 |
| 3200 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| 3400 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| 3600 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| 3800 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| 4000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |

## 1.0 Introduction

This document will describe the latest, upgraded firmware version of the processing FPGA on the FADC250 board. The new version of the firmware will provide more processing, include monitoring capabilities and reduces the processing time and data size. It has 6 unique functions as follows:

- Read Out Processing
- Trigger Processing
- Monitoring
- IC Configuration
- Miscellaneous
- Housekeeping.

The FADC250 board and this firmware are running nominally at 250MHz clock rate and all the time references listed in this document are Number of Samples * 4 nS. However the board and this firmware can also be ran at slower clock. In this case all the time references listed in this document are Number of Sample * 1/clock.

## 2.0 Read Out Processing

In Read Out Processing, the ADC samples are continuously stored in a 2048 sample Circular Raw Buffer (one per channel). When a trigger is received, a 12 bit Trigger Number counter is incremented and a number of ADC samples (Programmable Trigger Window) that are stored before (Programmable Lookback) the trigger is received is copied to the unprocessed Buffer. The Programmable Trigger Window (PTW) is user programmable from **5** to 511 (9 bits). The Programmable Lookback (PL) parameter is user programmable from 0 to 2047 (11 bits). After the PTW-number of ADC samples is copied to the unprocessed buffer, sample processing is based on the Processing Mode: Production Mode and Debug Mode.

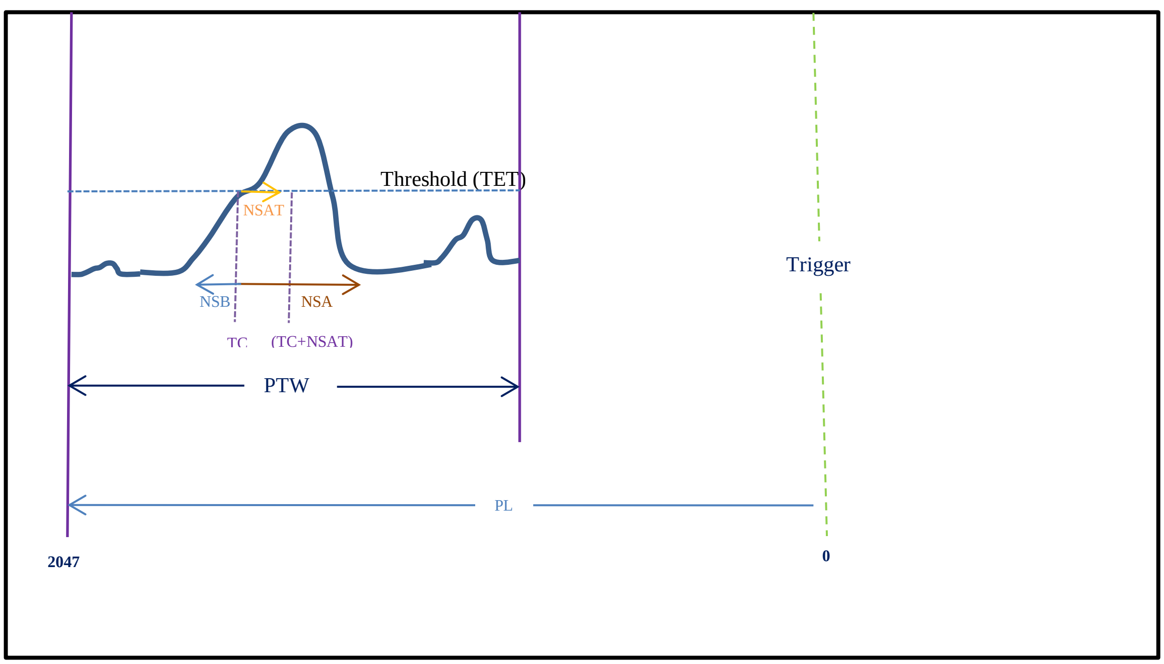



Figure 1 : Read Out Processing

### 1. Production Mode (9).

In this mode, the algorithm detects pulses within the PTW number of samples stored in the unprocessed buffer. The samples of pulse(s) are summed and the time at which the pulse occurred within the PTW is calculated.

a. **Pulse Definition**. Pulse identification is initiated if a number of consecutive samples (NSAT) are above a programmable threshold (TET). The pulse duration (data set) includes the number of

ADC samples before (NSB) and after (NSA) the first sample crossed threshold (TC). NSA includes TC. The data set that defines the pulse includes only samples within the trigger window (PTW), even if NSB and NSA would extend beyond the window boundaries.

i. Up to four distinct pulses may be detected within PTW samples. The Maximum Number of Pulses (MNoP) is programmable from 1 to 4. If PTW contains more than MNoP, only MNop pulses are reported.
ii. NSA is programmable from Min **2** to 511 (9bits).
iii. NSB is a 4 bit programmable parameter. When bit 3 is zero, bits 0-2 indicates the number of samples before TC to be included in Pulse Duration. When the bit 3 is one, bits 0-1 indicate the number of samples after TC to be excluded from Pulse Duration.
iv. **NSA has to be greater than NSB when bit 3 is one**. When bit 3 is one only bits 0-1 are effective (bit 2 is ignored).
v. Pulse duration = samples from MAX((TC-NSB),1) to MIN((TC+NSA-1), PTW) when NSB bit 3 = 0
vi. Pulse duration = samples from (TC+NSB) to MIN((TC+NSB+NSA-1), PTW) when NSB bit 3 = 1.
vii. Pulse number is counting from 1.
viii. Pulse has to have at least one sample below (less than) TET to be considered the end to allow detection of a next pulse.
ix. When NSB is positive, if the threshold crossing is within the last NSAT + 1 samples of the window, there is **NO** pulse assigned.
x. When NSB is negative, if the threshold crossing is within the last NSAT + [NSB] + 2 samples of the window, there is **NO** pulse assigned.
**xi. When NSB is negative, NSA – (NSB bits 1, 0) has to be greater than 3.**
**xii.**

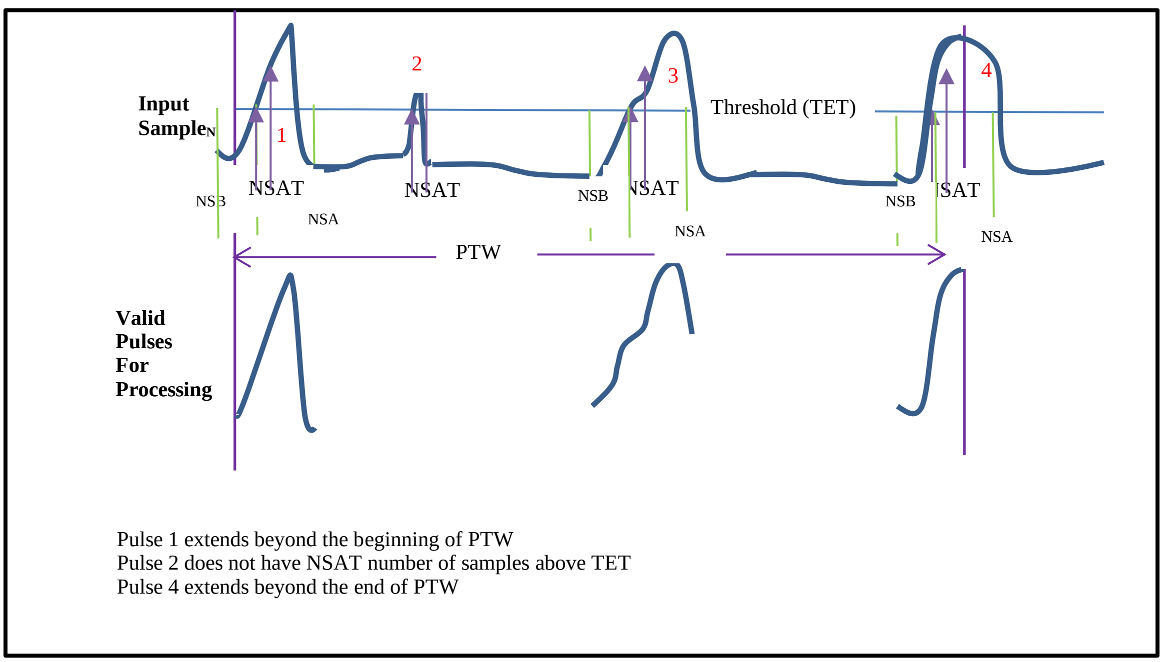



Figure 2 : Pulse Definition

b. **Pulse Sum** is sum of samples of the defined Pulse Duration. The resulting Sum is an 18 bit quantity. When a pulse falls outside PTW, the condition is reported in the Integral quality bits (3 bits). When the sum is overflowed, all sum bits are one..

i. When any sample within a pulse is underflow, integral quality bit 9 of integral word for that pulse will be set.
ii. When any sample within a pulse is overflow, integral quality bit 10 of integral word for that pulse will be set
iii. When NSA falls outside the ending of PTW, integral quality bit 11 of integral word will be set.
iv. Time over threshold counter: The number of samples where the FADC amplitude is larger than the readout threshold (TET) in the integration window will be reported in the data using 9 bits.

| Sum Quality Bit | Functions |
|---|---|
| 0 | Set when any sample within a pulse is underflow. |
| 1 | Set when any sample within a pulse is overflow |
| 2 | Set When NSA falls outside of window ending |

c. **Pulse Pedestal** is the sum of a number of samples (NPED) after the beginning of PTW. NPED is programmable from 4 to 15 samples. When any of NPED samples is greater than a programmable max pedestal threshold (MaxPed) or is underflow or over flow, the Pedestal Quality bit 14 (shown in Appendix A) of Pedestal Sum is set.

d. **Time to Digital Converter (TDC)**

   i. The time reported represents the time on the pulse's leading edge where half of its maximum sampled amplitude is reached. The algorithm for computing this time is described below. Exceptional cases where the algorithm cannot be applied are also discussed. Whenever the algorithm fails, the reported time is the threshold crossing time TC. Information is returned that identifies these cases to the user.

   ii. A baseline amplitude (VMIN) is determined for the entire trigger window by averaging the first 4 samples of the trigger window. A pulse with threshold crossing sample number TC is identified in the manner discussed in the section on pulse definition. The peak amplitude (VPEAK) is determined by finding a sample beyond TC for which the sample value first decreases. The algorithm will search for VPEAK beyond the expected end of the pulse (TC + NSA). Cases for which no VPEAK is detected are discussed below.

   iii. The half amplitude (VMID = (VPEAK + VMIN) / 2)) of the pulse is computed. The sample number N1 is found on the leading edge of the pulse that satisfies:
      1. $V(N1) \le VMID < V(N1+1)$
      2. where V(N1) and V(N+1) are the sample values of adjacent samples N1 and N1+1. N1 is reported as the coarse time.

   iv. The estimated time of occurrence of VMID between samples N1 and N1+1 is determined by a linear interpolation using their sample values V(N1) and V(N1+1). The time between samples (4 ns) is divided into 64 subsamples (62.5 ps each). In essence,
      1. $TF = 64*(VMID – V(N1)) / (V(N1+1) – V(N1))$.
      2. TF is reported as the fine time with values from 0 to 63. TF is reported in Fine Time bits 20-15 (Appendix A).
      3. Coarse Time is the time of V(N1) and it is reported in Coarse Time bits 29-21 (Appendix A).

   v. If any of the first 4 samples is greater than MaxPed but less than TET, the TDC algorithm will proceed and Time Quality Bit 0 will be set to 1.

   vi. If any any of the first 4 samples is greater than TET or ~~underflow~~, the TDC will NOT proceed.
      1. pulse time is set to TC,
      2. Pulse Peak is reported if found.
      3. Time Quality bits 0 (appendix A) is set to 1.
      4. Time Quality bits 1 (appendix A) is set to 1 if Vpeak cannot be found

   vii. A problem with the algorithm occurs if VPEAK is not found within the trigger window. In this case, the reported parameters are as followed:
      1. Pulse time is set to TC.
      2. Pulse Peak is set to zero.
      3. Time Quality bit 1 (appendix A) is set to 1.

   viii. Vpeak is found when the current sample is less than the previous sample. Moreover this condition has to be met 1 sample before end of window.

   ix. If the pulse extended beyond the window, TC will be reported and Sum quality bit 2 is set.

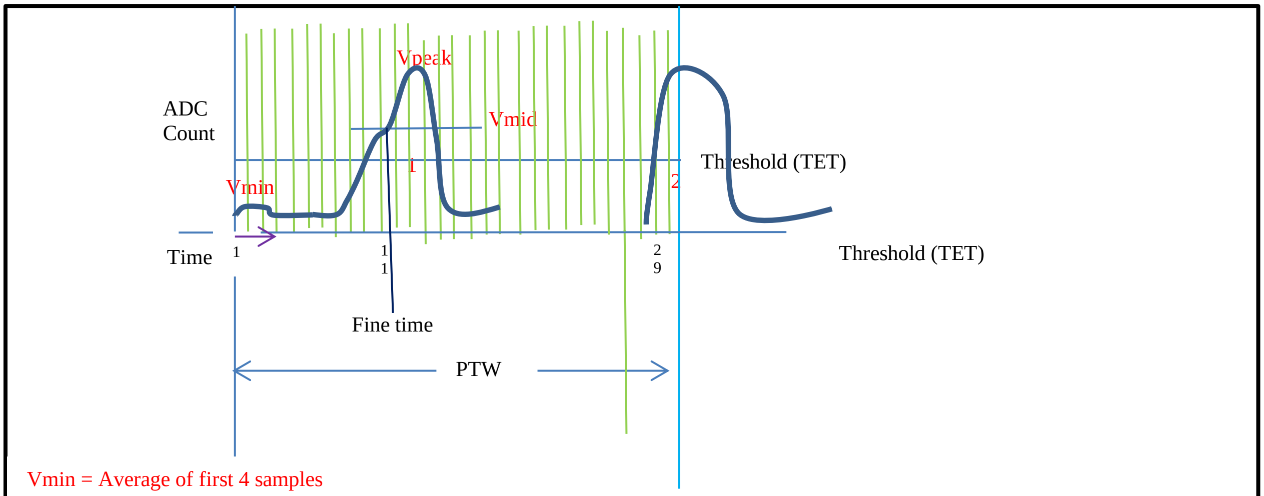
   x. When Vpeak is beyond NSA, bit 2 will be set. TDC time is reported if condition viii is met. If condition viii is not met, TC is report and Time quality bit 1 is also set. When a pulse occurred close to end of window and Vpeak cannot be found this bit will also set.

   xi. Time reported is 15 bits and has a time resolution of 62 pico-seconds. The upper 9-bits are called the coarse time and the lower 6-bits are called the fine time. The fine time of TC time is always zero. Both the coarse time of TC and TDC are starting from one. In other word the coarse time of the first sample of the PTW is one.

xii. "If any of the first 4 samples is greater than TET the TDC time quality bit 1 will be set. If If any of the first 4 samples is greater than MaxPed or greater than TET or is overflow or underflow, the TDC time quality bit 0 will be set"

**xiii. When coarse time is reported as one, the parameters for that pulse should be discarded.**

| Time Quality Bit | Functions |
|---|---|
| 0 | Set when any of the first 5 samples is greater than MaxPed or TET or is underflow or overflow |
| 1 | When Vpeak cannot be found ~~or when any of the first 4 samples is greater than TET~~. |
| 2 | Pulse Peak is beyond NSA or could be beyond end of window |




Vmin = Average of first 4 samples

| Pulse Time | | Coarse | Fine |
|---|---|---|---|
| Pulse #1 | = | 000001011 | 100000 |
| Pulse #2 | = | 000011101 | 000000 |

Pulse Time for pulse #1 is calculated as described above
Pulse Time for pulse #2 is set to TC because the pulse peak was not found within PTW.

Figure 3: TDC

e. **To run Production Mode**
   i. Set up Programmable Parameters
   ii. Write Config 1 as follow:
      1. Bit 3 = 1 Accept and Process Trigger
      2. Bit 5-4 = Max Number of Pulse in PTW to process
      3. Bit 7 = 1 to process data from ADC IC; 0 to process data from Play Back (see below)
      4. Bit 9-8 = 01

f. **To change to Debug Mode**
   i. Write Config 1 as follow:
      1. Bit 3 = 0 Stop Accepting trigger
      2. Bit 5-4 = Max Number of Pulse in PTW to process
      3. Bit 7 = 1 to process data from ADC IC; 0 to process data from Play Back (see below)
      4. Bit 9-8 = 01 Do not change mode yet
   ii. Wait For Done All Trigger Received by polling bit 15 of Status 1
   iii. Write Config 1 as follow:
      1. Bit 3 = 1 Accept and process trigger
      2. Bit 5-4 = Max Number of Pulse in PTW to process
      3. Bit 7 = 1 to process data from ADC IC; 0 to process data from Play Back (see below)
      4. Bit 9-8 = 10 change to Debug Mode

## 2. Debug Mode (10).

In addition to calculating Pulse Sum, Pulse Pedestal, TDC, Pulse Peak as described in Production Mode, this mode also reports all the samples in PTW if there is at least one pulse (NSAT number of samples greater than threshold) in the PTW.

a. **To run Debug Mode**
   i. Set up Programmable Parameters
   ii. Write Config 1 as follow:
      1. Bit 3 = 1 Accept and Process Trigger
      2. Bits 5-4 = Max Number of Pulse in PTW to process
      3. Bit 7 = 1 to process data from ADC IC; 0 to process data from Play Back (see below)
      4. Bits 9-8 = 10

b. **To change to Production Mode**
   i. Write Config 1 as follow:
      1. Bit 3 = 0 Stop Accepting trigger
      2. Bits 5-4 = Max Number of Pulse in PTW to process
      3. Bit 7 = 1 to process data from ADC IC; 0 to process data from Play Back (see below)
      4. Bits 9-8 = 10 Do not change mode yet
   ii. Wait For Done All Trigger Received by polling bit 15 of Status 1
   iii. Write Config 1 as follow:
      1. Bits 3 = 1 Accept and process trigger
      2. Bits 5-4 = Max Number of Pulse in PTW to process
      3. Bits 7 = 1 to process data from ADC IC; 0 to process data from Play Back (see below)
      4. Bits 9-8 = 01 change to Debug Mode

## 3. Read Out Programmable Parameters.

| Name | Number Of Bits | Functions |
|---|---|---|
| PTW | 9 | **PTW+1** number of ADC samples to be processed per trigger. **<u>Min is 6. Must be > NPED</u>** |
| PL | 10 | Number of ADC samples back from trigger point to beginning of PTW |
| NSB | 4 | When bit 3 = 0, bits 2-0 is the number of ADC samples from Threshold Crossing (TC) to be included in Pulse Sum. When bit 3 = 1, bits 1-0 is the number of ADC samples from TC to be excluded in Pulse Sum. |
| NSA | 9 | Number of ADC samples after TC to be included in Pulse Sum. NSA includes TC Sample. **<u>Min is 2</u>** |
| TET | 1 | ~~Trigger~~ Read Out Energy Threshold. A pulse is considered to be valid when a number of consecutive ADC samples (NSAT) are above TET |
| NSAT | 2 | Number of ADC samples that have to be above TET before a pulse is valid. 0-> 1 sample 1-> 2 samples 2->3samples 3->4samples |
| MNoP | 2 | Maximum number of pulses that will be processed. 0 → 1 1→2 2→3 3→4 |
| NPED | 4 | **NPED + 1** number of sample to sum up for Pulse Pedestal. **<u>Min is 4</u>** Max is 15. **<u>NPED has to be less than PTW.</u>** |
| MaxPed | 10 | ADC Samples have to be below this Max Pedestal value to be valid to be included in read back pedestal sum. |

**Failure to adhere to Min values can result in unpredictable results.**

### 4 Read Out Data Output.

Read Back data is written out to the External FiFo to be read by the Control FPGA. The data written out includes Trigger Number, Time at which the Trigger received, and Mode 9 or 10 data. The format is shown in Appendix A.

## 3.0 Trigger Processing

In Trigger Processing, the ADC samples are processed and sent to the Control FPGA to be sent to the Crate Trigger Processor (CTP). There are two functions: Trigger Sum and Trigger Hit Bits.

### 1. Trigger Pedestal Subtraction

Each ADC channel has a programmable Trigger Path Pedestal Subtract Value (TPSV). The TPSV is subtracted from every ADC sample. For ADC samples that are smaller than TPSV, the difference is set to zero. The differences are input to Trigger Sum.

### 2. Trigger Sum.

When a number of consecutive ADC Samples are greater than a programmable Trigger Path Threshold (TPT), three differences before and a TNSA number of differences after the 1st ADC Sample are sent to summing code. If at the end of TNSA samples, the immediate ADC sample is greater than TPT, another TNSA differences are sent. If at the end of TNSA samples, the immediate ADC sample is less than TPT, zeroes are sent. When 2 consecutive pulses are piling up, overlapping differences are not sent to summing code. In other words differences are only sent once. The summing code added differences and/or zeroes from all 16 ADC channels. Each ADC channel has individual TPSV values. One TPT value covers all ADC channels. Figure 3 show an example of differences "filtering"

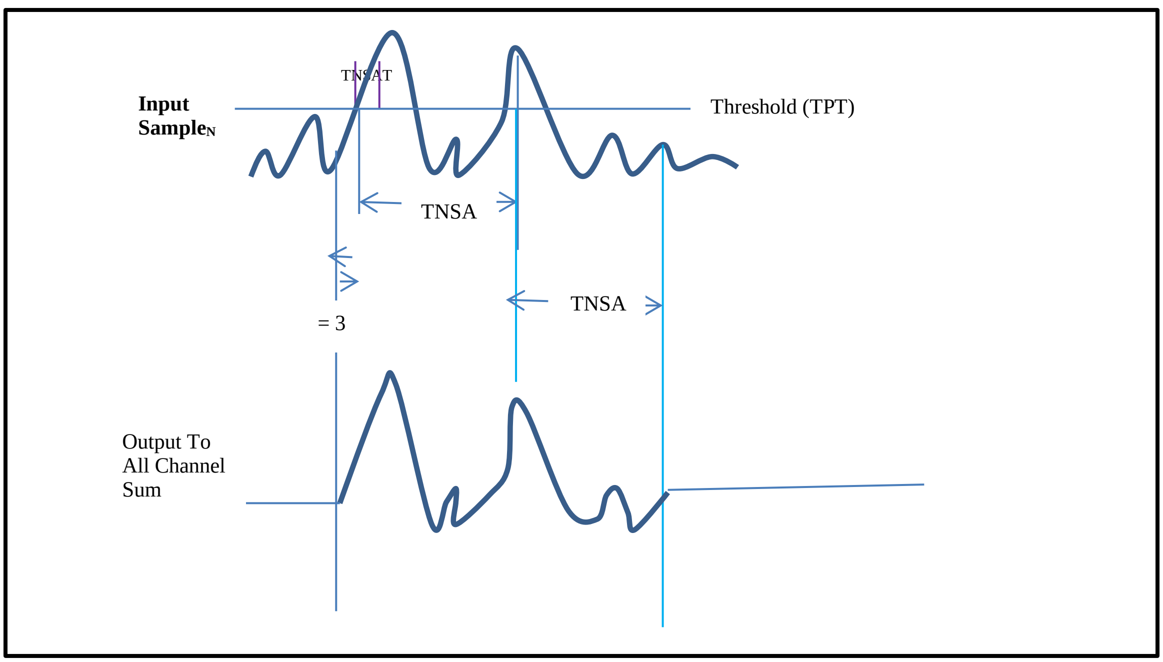



Figure 4. Example for Trigger Sum Sample Processing Code:

### 3. Trigger Hit Bit.

When the differences are greater than TPT, Hit Bit goes high. There is one hit bit per ADC channel

### 4. Trigger Processing Programmable Parameters.

| Names | Number Of Bits | Functions |
|---|---|---|
| TPSV | 12 | Trigger Path Pedestal Subtract Values are subtracted from ADC samples. The different ADC samples are sent to Trigger Sum and Hit Bits. Each ADC channel has TPSV (total 16 TPSV) |

| | | |
|---|---|---|
| TPT | 12 | When a number of consecutive ADC differences (TNSAT) are greater than Trigger Path Threshold (TPT), three before and TNSA differences ADC samples are sent to summing code. One TPT for all 16 ADC channels. |
| TNSAT | 2 | Number of consecutive ADC differences that have to be above TPT before a pulse is valid. 0-> 1 sample 1-> 2 samples 2->3samples 3->4samples |
| TNSA | 6 | Number of ADC differences after the first TPT crossing. |

# 4.0 Monitoring

## 1. Pedestal Sum

For each ADC channel, the sum of a programmable number unprocessed ADC samples can be read. When a read pedestal request is received from Control FPGA, it does the following for each ADC channel:

1. Capture a programmable number of unprocessed ADC samples (MNPED). These are not consecutive samples.
2. Sum up these samples. When any of MNPED samples is greater than a programmable max pedestal threshold (PMaxPed) or less than zero (ADC bit 13 is 1 and bit 0-12 are zeroes), bit 14 of the 14 bits (13-0) Sum will be set.
3. When sum is done, bit Monitored Pedestal Sum Done is set to 1. The time taken for summing is (MNPED + 3 + 2) * 1/Clock.
4. Host reads all 16 Sums at register STATUS2. All 16 sums are mapped to only one register. The first read will be sum of ADC channel 1, follow by sums of channels 2,3,...,16. Monitored Pedestal Sum Done bit will ReSet to 0.
5. The Pedestal Sum only monitors ADC Sample. It does not monitor PlayBack value.
6. **To Read Monitored Pedestal Sums**
   a. **Reset Monitored Pedestal Sum Request bit (Config1 Bit 15)**
   b. **Set Monitored Pedestal Sum Request bit**
   c. **Poll** Monitored Pedestal Sum Done **Bit (STATUS2 Bit 15) for a one**
   d. **Read STATUS2 16 times for Pedestal Sums of ADC channel 1,…, 16**

## 2. Monitoring Pedestal Sum Programmable Parameters

| Names | Number Of Bits | Functions |
|---|---|---|
| MNPED | 4 | **MNPED + 1** number of sample to sum up for Pulse Pedestal for Monitoring. **Min is 4** Max is 15. |
| PMaxPed | 10 | When an ADC Samples is greater than this, **bit 14 of the 14 bits (13-0) Sum will be set.** |

## 3. FPGA Die Temperature

The temperature of the FPGA die can be read at register STATUS3 (Die Temp). The Celsius temperature is calculated as follow:

DieTemp_C = ((float)(STATUS3 >> 6) * 503.975/1024) - 273.15;

## 4. Firmware Version Number

Firmware version number can be read at register STATUS1 bits 14 to 0. Bits 14-8 is 0x0C and bits 7-0 indicates the code revision

5. **Trigger Number 12 bits**

The present Trigger Number is also available at register STATUS1.

6. **STATUS Register**

| Names | Number Of Bits | Functions |
|---|---|---|
| STATUS 0 | 16 | Bits 14 to 0: Code Version<br>Bit 15: 1= Command can be sent to AD9230 |
| STATUS 1 | 16 | 15 → 1 Done process all Trigger received<br>11-0 →Trigger Number |
| STATUS 2 | 16 | Monitored Pedestal Sum.<br>Bit 15 → 1 Done computing Pedestal Sums.<br>Bit 14 → 0 Pedestal Sum O<br>1. 1 one or more ADC Sample is/are out of bound<br>Bit 13-0 → Sum of ADC Samples |
| STATUS 3 | 16 | FPGA Die Temperature |
| STATUS 4 | 16 | 7-0 Content of ADC register read back |

# 5.0 IC Configuration

## 1. ADC IC AD9230

The ADC AD9230 ICs are needed to be configured after power up.

a. To configure all ADC ICs at one time
    i. Poll bit 15 of Status 0 for a one. This indicates firmware is ready to accept command.
    ii. Write 0 to bit 7 of Config 4. Rising edge firmware sends data to AD9230
    iii. Select register of AD9230 to write to by writing to bits 15-8 of Config 5. Write data to be written to register of AD9230 by writing bits 7-0 of Config 5.
    iv. Set bits 7,6 and reset Bit 5 of Config 4. Bit 6 tells firmware to write to AD230. Bit 5 tells firmware to write to all AD9230
b. For Example to configure all ADC to convert negative going signal:
    i. Configure AD9230 delay clock
        1. Poll bit 15 of Status 0 for a one.
        2. Reset bit 7 of Config 4
        3. 0x17 to bits 15-8 of Config 5 to select AD9230 ADC_CLK_OUT_DELAY_REG
        4. 0x9E to bits 7-0 of Config 5. Data to write to ADC_CLK_OUT_DELAY_REG 0x9E. Delay clock b
        5. Set bit 7 and 6, reset bit 5 of Config 4. This tells firmware to write to AD9230
        6. Poll bit 15 of Status 0 for a one
        7. Reset bit 7 of Config 4
        8. 0xFF to bit 15-8 of Config 5 to select ADC_MASTER_TO_SLAVE_REG
        9. 0x01 to bit 7-0 of Config 5. Data to write to ADC_MASTER_TO_SLAVE_REG. Tell AD9230 to execute delay clock setting.
        10. Set bit 7 and 6, reset bit 5 of Config 4
    ii. Configure AD9230 to run in CML mode
        1. Poll bit 15 of Status 0 for a one.
        2. Reset bit 7 of Config 4
        3. 0x0F to bits 15-8 of Config 5 to select AD9230 ADC_AIN_CONFIG_REG
        4. 0x02 to bits 7-0 of Config 5. Data to write to ADC_AIN_CONFIG_REG. Run in CML mode
        5. Set bit 7 and 6, reset bit 5 of Config 4. This tells firmware to write to AD9230
        6. Poll bit 15 of Status 0 for a one
        7. Reset bit 7 of Config 4
        8. 0xFF to bit 15-8 of Config 5 to select ADC_MASTER_TO_SLAVE_REG
        9. 0x01 to bit 7-0 of Config 5. Data to write to ADC_MASTER_TO_SLAVE_REG. Tell AD9230 to execute delay clock setting.
        10. Set bit 7 and 6, reset bit 5 of Config 4

iii. Tell AD9230 to turn off test mode
1. Poll bit 15 of Status 0 for a one.
2. Reset bit 7 of Config 4
3. 0x0D to bits 15-8 of Config 5 to select AD9230 ADC_TEST_REG
4. 0x00 to bits 7-0 of Config 5. Data to write to ADC_TEST_REG. Turn off test mode
5. Set bit 7 and 6, reset bit 5 of Config 4. This tells firmware to write to AD9230
6. Poll bit 15 of Status 0 for a one
7. Reset bit 7 of Config 4
8. 0xFF to bit 15-8 of Config 5 to select ADC_MASTER_TO_SLAVE_REG
9. 0x01 to bit 7-0 of Config 5. Data to write to ADC_MASTER_TO_SLAVE_REG. Tell AD9230 to execute delay clock setting.
10. Set bits 7 and 6, reset bit 5 of Config 4

# 6.0 Miscellaneous Functions:

## 1. ADC Channel Disable

Each ADC channel has a bit that when set will zero the input to Trigger Sum and Trigger Hit Bits. Note, channels can be excluded from the readout if needed by setting readout thresholds (TET) to 4095. To disable (zeroes input to Trigger Sum and Hit Bit) an ADC channel, set the corresponding bit of Config 2 to 1.

## 2. Sync_reset

When Sync_reset signal from Control FPGA is high and bit 15 of Config 3 is 0, the following signals are in RESET (zeroes):

1. Time Stamp
2. Trigger Number
3. Not accepting Trigger signal from Control FPGA

## 3. Soft Reset

When Soft Reset signal from Control FPGA is asserted low, everything EXCEPT Configuration Registers and AD9230 is reset.

## 4. Hard Reset

When Hard Reset signal from Control FPGA is asserted low, everything is Set to values shown in Appendix C. All AD9230 ICs are reset to power state.

## 5. Play Back

User defines pulses maybe injected into the processing pipeline using a playback feature. Play Back stores 32, 13-bit ADC values in RAM and cycles through 32 ADC values when a Trigger_2 signal from Control FPGA goes from low to high. There are 16 Play Back, one per ADC Channel. All 512 ADC values are written into memory via Control Bus. Since all 512 ADC values occupy only one Control Bus address, all 512 values have to be written sequentially all at once The first 32 values are associated with ADC channel 0 and last 32 values are associated with ADC channel 15.
When **bit 7 of Config 1** is set, Play Back outputs (instead of ADC IC outputs) are applied to all processing functions of all ADC channels.

a. To write 510 ADC values to RAM
   i. Put ADC value on bits 12-0 of Test Wave Form register. Set Bit 15.
   ii. Write Test Wave Form register
   iii. Read Test Wave Form register to verify that value is stored to RAM
   iv. Repeat above three steps for 510 values
b. To write 511$^{th}$ and 512$^{th}$ ADC values to RAM
   i. Put 511$^{th}$ ADC value on bits 12-0 of Test Wave Form register. ReSet Bit 15.
   ii. Write Test Wave Form register
   iii. Read Test Wave Form register to verify that value is stored to RAM
   iv. Repeat above three steps for 512$^{th}$ values

# 7.0 House Keeping:

## 1. Control Bus

Registers and Status are accessed through Control Bus connected to FPGA. Control Bus is asynchronous 16 bits data bus. In the basic mode, Control Bus can access 65535 addresses. In extended mode the Control Bus can access up to 65,535 x 65,535 addresses by mean of secondary address feature (See FADC250 Program Manual)

a. The signals of Control Bus are:
   i. ADR_DAT = Address and Data
   ii. AS_N = Address strobe
   iii. AK_N = Address Acknowledge
   iv. DS_N = Data strobe
   v. DK_N = Data Acknowledge
   vi. RD_N = Low indicate Read cycle
   vii. MS = Mode select
   viii. SS = Slave Status.

b. To write to Register
   i. Control FPGA puts register's address on ADR_DAT bus. Drives MS low
   ii. Control FPGA brings AS_N low
   iii. Processing FPGA accept register's address. Drive SS low
   iv. Processing FPGA brings AK_N low.
   v. Control FPGA puts register's value on ADR_DAT bus
   vi. Control FPGA brings DS_N low
   vii. Processing FPGA accept register's value
   viii. Processing FPGA brings DK_N low.
   ix. Control FPGA drives DS_N high
   x. Processing FPGA drives DK_N high,
   xi. Control FPGA drives AS_N high
   xii. Processing FPGA drives AK_N high

**c.** To read Register **(Note Unused bits are read back as zeroes)**
   i. Control FPGA puts register's address on ADR_DAT bus. Drives MS low
   ii. Control FPGA brings AS_N low
   iii. Processing FPGA accept register's address. Drive SS low
   iv. Processing FPGA brings AK_N low.
   v. Control FPGA stops driving ADR_DAT bus
   vi. Control FPGA brings DS_N low.
   vii. Processing FPGA puts register's value on ADR_DAT bus
   viii. Processing FPGA brings DK_N low
   ix. Control FPGA accepts register's value.
   x. Control FPGA drives DS_N high
   xi. Processing FPGA drives DK_N high, Stop driving ADR_DAT bus
   xii. Control FPGA drives AS_N high
   xiii. Processing FPGA drives AK_N high

# Appendix A:  Data Format of FADC Processing

**Event Header** (2) – indicates the start an event.

(35 – 32) = 0001
(31) = 1
(30 – 27) = 2
(26 – 22) = 00000
(21 – 12) = trigger time (bits 9 – 0 (see below))
(11 – 0) = trigger number

**Trigger Time** (3) – time of trigger occurrence relative to the most recent global reset. Time in the ADC data processing chip is measured by a 48-bit counter that is clocked by the 250 MHz system clock. The six bytes of the trigger time

$$\text{Time} = T_A\ T_B\ T_C\ T_D\ T_E\ T_F$$

are reported in two words (Type Defining + Type Continuation).

Word 1:

(35 – 32) = 0000
(31) = 1
(30 – 27) = 3
(26 – 24) = $T_C$ bits 2 – 0 (duplicated in Word 2)
(23 – 16) = $T_D$
(15 – 8) = $T_E$
(7 – 0) = $T_F$

Word 2:

(35 – 32) = 0000
(31) = 0
(30 – 24) = reserved (read as 0)
(23 – 16) = $T_A$
(15 – 8) = $T_B$
(7 – 0) = $T_C$

**Window Raw Data** (4) – raw ADC data samples for the trigger window. The first word identifies the channel number and window width. Multiple continuation words contain two samples each. The earlier sample is stored in the most significant half of the continuation word. Strict time ordering of the samples is maintained in the order of the continuation words. A *sample not valid* flag bit 13 will be set when PTW+1 is odd.

Word 1:

(35 – 32) = 0000
(31) = 1
(30 – 27) = 4
(26 – 23) = channel number (0 – 15)
(22 – 12) = reserved (read as 0)
(8 – 0) = PTW + 1 (window width (in number of samples))

Words 2 - N:

(35 – 32) = 0000
(31) = 0
(30) = reserved (read as 0)
(29) = sample x not valid
(28 – 16) = ADC sample x (includes overflow bit)
(15 – 14) = reserved (read as 0)
(13) = sample x + 1 not valid
(12 – 0) = ADC sample x + 1 (includes overflow bit)

**Pulse Parameters** (9) – computed pulse parameters for detected pulses in a channel. The first word identifies the channel number, event number within the block, and pedestal information for the window. Multiple continuation word *pairs* contain information about the pulses detected. For a channel with hits detected:

Word 1: Channel ID and Pedestal information (reported *once* for a channel with hits)
(35 – 32) = 0000
(31) = 1
(30 – 27) = 9
(26 – 19) = event number within block (1 – 255)
(18 – 15) = channel number (0 – 15)
(14) = pedestal quality
(13 – 0) = pedestal sum

Word 2 : Integral of first pulse in window
(35 – 32) = 0000
(31) = 0
(30) = 1
(29 – 12) = 18-bit sum of raw samples that constitute the pulse data set
(11) = NSA extended beyond PTW
(10) = One or more samples is overflow = 0x1FFF
(9) = One or more sample is underflow = 0x1000
(8 – 0) = number of samples within NSA that the pulse is above threshold

Word 3 : Time of first pulse in window
(35 – 32) = 0000
(31) = 0
(30) = 0
(29 – 21) = coarse time (4 ns/count)
(20 – 15) = fine time (0.0625 ns/count)
(14 – 3) = pulse peak
(2) = Vpeak is beyond NSA or could be beyond window end
(1) = Vpeak cannot be found
(0) = 1 or more of first 4 samples is above either MaxPed or TET

Words 2 and 3 are repeated for *each additional pulse* found in the window for the channel.

**Event Trailer:** Indicate the end of an event.
EVENT_TRAILER = "0010" & X"E8000000";

Note: For maximum compression of data the Event Header (2) and Trigger Time (3) words may be suppressed from module readout in the Control FPGA (See Programming the FADC250)

# Appendix B: Example of Data Format for Production Mode 9

1st Trigger Occurred at Time 0x123456
Channel 1 and 15 has 1 good pulse each

**Event Header** (2) – indicates the start an event.
Word 1:
(35 – 32) = 0001
(31) = 1
(30 – 27) = 2
(26 – 22) = 00000
(21 – 12) = "00" & x"56" ( trigger time (bits 9 –))
(11 – 0) = x"0001"

**Trigger Time**
Word 2:
(35 – 32) = 0000
(31) = 1
(30 – 27) = 3
(26 – 24) = "011" $T_C$ bits 2 – 0 (duplicated in Word 2)
(23 – 16) = x"4"
(15 – 8) = x"5"
(7 – 0) = x"6"

Word 3:
(35 – 32) = 0000
(31) = 0
(30 – 24) = 0
(23 – 16) = x"1"
(15 – 8) = x"2"
(7 – 0) = x"3"

**Pulse Parameters**

**Channel 1 data**
Word 4: Channel ID and Pedestal information (reported *once* for a channel with hits)
(35 – 32) = 0000
(31) = 1
(30 – 27) = 9
(26 – 19) = 00000000 event number within block (0 – 255)
(18 – 15) = 0001
(14) = 0
(13 – 0) = pedestal sum
0xC800----

Word 5 : Integral of first pulse in window
(35 – 32) = 0000
(35 – 32) = 0000
(31) = 0
(30) = 1
(29 – 12) = 18-bit sum of raw samples that constitute the pulse data set
(11 – 9) = 000
(8 – 0) = number of samples within NSA that the pulse is above threshold

Word 6 : Time of first pulse in window
(35 – 32) = 0000
(35 – 32) = 0000
(31) = 0

(30) = 0
(29 – 21) = coarse time (4 ns/count)
(20 – 15) = fine time (0.0625 ns/count)
(14 – 3) = pulse peak
(2 – 0) = time quality

**Channel 15 data**

Word 7: Channel ID and Pedestal information (reported *once* for a channel with hits)
(35 – 32) = 0000
(31) = 1
(30 – 27) = 9
(26 – 19) = 00000000 event number within block (0 – 255)
(18 – 15) = 1111
(14) = 0
(13 – 0) = pedestal sum

Word 8 : Integral of first pulse in window
(35 – 32) = 0000
(31) = 0
(30) = 1
(29 – 12) = 18-bit sum of raw samples that constitute the pulse data set
(11 – 9) = 000
(8 – 0) = number of samples within NSA that the pulse is above threshold

Word 9 Time of first pulse in window
(35 – 32) = 0000
(31) = 0
(30) = 0
(29 – 21) = coarse time (4 ns/count)
(20 – 15) = fine time (0.0625 ns/count)
(14 – 3) = pulse peak
(2 – 0) = time quality

**Event Trailer:** Indicate the end of an event.

Word 10
EVENT_TRAILER = "0010" & X"E8000000";

# Appendix C: Control Bus Memory Map for Registers

Unused Bits are read back as zeroes

**Failure to adhere to Min values can result in unpredictable results**

| Name | Width (Bits) | Quantity | Access | Primary Address (Secondary Address) | Power Up Values (hex) | Function |
|---|---|---|---|---|---|---|
| STATUS0 | 16 | 1 | R | 0x0000<br>(---) | | Bits 14 to 0: Code Version<br>Bit 15: 1= Command can be sent to AD4229 |
| STATUS1 | 16 | 1 | R | 0x0001<br>(---) | | 15 : 1 Done all Trig received<br>11-0 : TRIGGER NUMBER |
| STATUS2 | 16 | 1 | R | 0x0002<br>(---) | | Monitored Pedestal<br>15 → Sum is valid<br>14 → 0 Sum OK. 1 One or more is out of bound<br>13-0 → Sum |
| CONFIG1 | 16 | 1 | R/W | 0x0003<br>(---) | 0040 | Bit 0-2 (old code process mode):<br><br>Bit 3: 1:Run<br>Bit 5-4 : Max Number of Pulses in Mode 10 and 9<br><br>Bit 6: 1 Inverts ADC data output<br>Bit 7: Test Mode (play Back).<br><br>Bit 9-8:<br>00 → mode 9 (pulse parameters)<br>01 → mode 10 (pulse parameters & raw)<br>11 → mode 11 (raw)<br><br>11-10 NSAT<br>13-12 TNSAT<br><br>15→ Request Sum of Pedestal for monitoring purpose |
| CONFIG2 | | | R/W | 0x0004<br>(---) | 0040 | When 1 ADC values = 0<br>Bit 0 → ADC 0<br>Bit 1 → ADC 1<br>Bit 2 → ADC 2<br>Bit 3 → ADC 3<br>Bit 4 → ADC 4<br>Bit 5 → ADC 5<br>Bit 6 → ADC 6<br>Bit 7 → ADC 7<br>Bit 8 → ADC 8<br>Bit 9 → ADC 9<br>Bit 10→ ADC 10<br>Bit 11→ ADC 11<br>Bit 12→ ADC 12<br>Bit 13→ ADC 13<br>Bit 14→ ADC 14<br>Bit 15→ ADC 15 |

| | | | | | | |
|---|---|---|---|---|---|---|
| CONFIG 4<br>**[0x114]** | 16 | 1 | R/W | 0x0005 | 0040 | ~~15..12=> Select which ADC receive IDELAY control bits and read back IDELAY comparator error~~<br>15..13 => Select which ADC Idelay Error Bits to monitor<br>~~11=> Idelay comparator reset~~<br>~~10=> Increment IDELAY N delay value~~<br>~~9 => Decrement IDELAY P delay value~~<br>~~8 => Reset IDELAY~~<br>12 => HostIdelayTuneEn<br>11 => IDELAYE3_EN_VTC_Host<br>10 => Clear_Clk_wiz_PhSh_psdone_latch<br>9 => Clk_wiz_PhSh_psen<br>8 => Clk_wiz_PhSh_psincdec<br>7 => rising edge execute ADC command<br>6 => 1 write to all ADC<br>5 => 0 write to ADC<br>1 read from ADC. Data is at Stat 4<br>4 => ~~1 Reset ADC~~<br>3..0 => Select ADC to write to or read from |
| CONFIG5 | 16 | 1 | | 0x0006 | 0040 | 15..8 => Register inside ADC<br>7..0 => Data to write to register. |
| PTW | 9 | 1 | R/W | 0x0007<br>(---) | 0010 | **PTW + 1** number of ADC sample to include in trigger window.<br>PTW = Trigger Window (ns) * 250 MHz.<br>**Minimum is 6**. |
| PL | 11 | 1 | R/W | 0x0008<br>(---) | 0000 | Number of sample back from trigger point.<br>PL = Trigger Window(ns) * 250MHz |
| NSB, | 4 | 1 | R/W | 0x0009<br>(---) | 0000 | 3..0: Read Back Path NSB<br>Number of sample before trigger point to include in data processing. This include the trigger Point. When NSB bit 3 is 1:<br>**NSA has to be > NSB bits 1,0 by at least 4 => NSA – (NSB bits 1,0) ≥ 3** |
| NSA | 15 | 1 | R/W | 0x000A<br>(---) | 0005 | 8..0: Read Back Path NSA<br>Number of sample after trigger point to include in data processing. **Minimum is 2**<br>14..9: Trigger Path NSA |
| TET | 12 | 16 | R/W | 0x000B -<br>0x001A | 0000 | ~~Trigger~~ Read Out Energy Thredhold. |
| CONFIG6<br>(Monitored Pedestal Sum) | 16 | 1 | R/W | 0x001B | 0000 | 13-10 MNPED : The number of ADC sample to sum up is MNPED + 1. **Min is 4**<br>9-0 PMaxPed : When an ADC Samples is greater than this, bit 14 of the 14 bits (13-0) Sum will be set. |
| CONFIG7<br>(Read Back Pedestal Sum) | 16 | 1 | R/W | 0x001C | 0000 | 13-10 NPED : The number of ADC sample is NPED + 1<br>9-0 MaxPed |
| Test Wave Form | 16 | 1 | R/W | 0x001D | 0000 | Write to PPG. Read should immediately follow write. |
| ADC Pedestal Subtract | 12 | 16 | R/W | 0x001E-<br>0x002D | 0000 | Subtract from ADC(0-15) Count before Summing |
| | | | | | | |
| Config 3 | 16 | 1 | R/W | 0x002E | 0000 | 15 : Sync Disable<br>(11..0) Trigger Path Processing Threshold |

| | | | | | | |
|---|---|---|---|---|---|---|
| STATUS 3<br>**[0x19C]** | 16 | 1 | R | 0x002F | | ~~FPGA core temp (DieTemp)~~<br>6 -<br>5 -<br>4 - Clk_wiz_PhSh_psdone<br>3 - Detect_ERROR_IDDR_P_CH8_15<br>2 - Detect_ERROR_IDDR_N_CH8_15<br>1 - Detect_ERROR_IDDR_P_CH0_7<br>0 - Detect_ERROR_IDDR_N_CH0_7 |
| STATUS 4 | 16 | 1 | R | 0x0030 | | 14 – Vcc Aux is out of range<br>13 – Vcc Int is out of range<br>12 – FPGA over 85 degree C<br>11 – FPGA over 125 degree C<br>9-0 Result of ADC AD4229 register read back |
| Roque_PTW_Fall_Back | 16 | 1 | R/W | 0x0032 | | When 1 enable send raw data when any of the first 4 samples is above threshold.<br>When 0 proceed to calculate SUM and TDC<br>Bit 0 → ADC 0<br>Bit 1 → ADC 1<br>Bit 2 → ADC 2<br>Bit 3 → ADC 3<br>Bit 4 → ADC 4<br>Bit 5 → ADC 5<br>Bit 6 → ADC 6<br>Bit 7 → ADC 7<br>Bit 8 → ADC 8<br>Bit 9 → ADC 9<br>Bit 10→ ADC 10<br>Bit 11→ ADC 11<br>Bit 12→ ADC 12<br>Bit 13→ ADC 13<br>Bit 14→ ADC 14<br>Bit 15→ ADC 15 |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

# Appendix B

| Bias DAC Count | ADC Count With No Input (in hex) |
|---|---|
| 4096 | 1000 |
| 4000 | 1000 |
| 3500 | 1000 |
| 3300 | 66 |
| 3100 | 173 |
| 3000 | 1FB |
| 2900 | 281 |
| 2800 | 30A |
| 2700 | 393 |
| 2600 | 419 |
| 2500 | 49E |
| 2400 | 526 |
| 2300 | 5ae |
| 2200 | 632 |
| 2100 | 6BA |
| 2000 | 741 |
| 1900 | 7C6 |
| 1800 | 850 |
| 1700 | 8D5 |
| 1600 | 95B |
| 1500 | 9E1 |
| 1400 | A6B |
| 1300 | AF0 |
| 1200 | B7B |
| 1100 | C00 |
| 1000 | C86 |
| 900 | D0E |
| 800 | D91 |
| 700 | E19 |
| 600 | EA1 |
| 500 | F29 |
| 400 | FAC |
| 300 | 1FFF |

To help checking simulation data to Control FPGA (shown in Hex):
190 ---- T A T R N ______ ---- → 0 0 TB1 TB 2 (Event Header)
098 T D T E T F ______ Trigger Time upper bytes
000 T A T B T C ______ Trigger Time lower bytes

Mode 9
0C8000 P D S ______ Pedestal Sum
0 ---- S U M M ---- S A ___ ---- → 1 S18 S17 S16 ---- → B O U SA8 B O U →NSA Beyond Window, Sample(s) Overflow, Sample(s) Underflow SA → Number Of Sample Above Threshold

0 ---- C ---- F ---- P P ---- ___ ---- → 0 0 C8 C7
C → Coarse Time
F → Fine Time
---- → F0 P11 P10 P9
P → Pulse Peak
---- → P0 B N Y --- B → Vpeak is beyond NSA or PTW
N → Vpeak can't be found
Y → 1 or more of first 4 samples is above either MaxPed or TET

Mode 10
Window Raw Data
First word
0A _ _ _ _ _ _ _
27 = 0
(26 – 23) = channel number (0 – 15)
(22 – 12) = reserved (read as 0)
(8 – 0) = PTW + 1 (window width (in number of samples))

Subsequent words

Words 2 - N:
(35 – 32) = 0000
(31) = 0
(30) = reserved (read as 0)
(29) = sample x not valid
(28 – 16) = ADC sample x (includes overflow bit)
(15 – 14) = reserved (read as 0)
(13) = sample x + 1 not valid
(12 – 0) = ADC sample x + 1 (includes overflow bit)