Summary of FADC250 Operating Modes

(E. Jastrzembski, H. Dong – **2/18/14**)

We briefly describe the current operating modes of the FADC250 and identify the data word types that each produces. Since several modes report pulse data, we begin by discussing how pulses are defined, and how times are assigned to pulses.

Pulse Definition

Several modes of operation of the FADC250 detect a pulse from the raw samples and report data associated with this pulse. We describe the pulse detection algorithm that is currently implemented in the firmware. Exceptions are noted below.

The trigger window consists of NW samples of the ADC. Samples are 4 ns apart in time. Samples within the trigger window are numbered from 1 to NW.

A pulse window width that would contain the entire expected detector pulse is specified by the user. The pulse window width (in number of samples) is defined from two programmed values (NSB, NSA): width is NSB + NSA.

Pulse identification is initiated if at least one sample in the trigger window is above the programmed threshold. The sample number of the first threshold crossing (TC) in the trigger window is determined. If the first sample of the trigger window is above threshold, a pulse is declared and TC = 1 for this pulse. Otherwise, the transition above threshold must be explicitly visible: at least one sample before TC must be at or below threshold.

The parameter NSB represents the number of samples before the threshold crossing sample (TC) that are included in the pulse data set. When TC is found and TC – NSB < 1, the expected pulse started before the trigger window. In this case, the first sample of the pulse data set is declared to be the first sample of the trigger window.

The parameter NSA represents the number of samples after the threshold crossing to include in the pulse data set. Sample number TC is included in this count. If the computed last sample number (TC + NSA – 1) of the pulse data set is greater than NW, the expected pulse extends past the trigger window. In this case, the last sample of the pulse data set is declared to be the last sample of the trigger window.

In summary, the pulse data set consists of sample numbers:

$$\mathrm{MAX}(\,(TC - NSB), 1\,) \quad \text{to} \quad \mathrm{MIN}(\,(TC + NSA - 1), NW\,).$$

Up to three distinct pulses may be detected within the trigger window. The maximum number of pulses is programmable. The earliest pulses found, up to the programmable limit, are reported. Any additional activity in the trigger window is ignored.

After a pulse is detected, a search for a subsequent pulse begins at the sample number after the pulse (TC + NSA). Another pulse is identified only if there is a clear transition above threshold after the previous pulse: at least one sample before TC must be at or below threshold. Consecutive pulses may overlap (i.e. share samples). The magnitude of possible overlap is determined by NSB.

Pulse Time

When operating in a pulse finding mode, a time is assigned to each pulse found. The simplest reportable time is TC, the number of the first sample above threshold for the pulse. ADC modes 2 and 3 report TC as pulse time.

A procedure for calculating a high-resolution time has also been implemented in the firmware. 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.

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.

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:

$$V(N1) \leq VMID < V(N1+1)$$

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.

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,

$$TF = 64*(VMID - V(N1)) / (V(N1+1) - V(N1)).$$

TF is reported as the fine time with values from 0 to 63. In addition to the time data word (type 8) consisting of the coarse and fine times, a pulse parameter data word (type 10) reports the values of VMIN and VPEAK used in the computation of the high resolution time.

A problem in the computation of the high resolution time will occur when VMIN is greater than VPEAK. In the current implementation of the algorithm, the simplest way to protect against this situation is to require that all 4 samples that determine the VMIN must be at or below threshold for the high resolution timing algorithm to be used. If this condition is not satisfied, the reported pulse time is TC, and both VMIN and VPEAK are reported as 0 in the pulse parameter data word (type 10) to identify the condition.

A problem with the algorithm occurs if VPEAK is not found within the trigger window. In this case, the reported pulse time is TC. To identify this condition, the pulse parameter data word (type 10) reports VPEAK = 0 and VMIN as measured.

In the current implementation of the algorithm, a technical difficulty arises when TC is near the end of the trigger window. If (NW – TC) < 5, the reported pulse time is TC, and the pulse parameter data word (type 10) reports VPEAK = 0 and VMIN as measured.

To standardize reporting of the pulse time, a 15-bit time value is always formed. The coarse time is reported in bits 14 through 6, while the fine time is reported in bits 5 through 0. Each count represents 62.5 ps. In situations where only the threshold crossing TC is reported, TC is set to be the coarse time and the fine time is 0.

ADC Modes

ALL operating modes produce data types 0 (block header), 1 (block trailer), 2 (event header), 3 (trigger time), 12 (scaler – optional), 14 (data not valid), 15 (filler) (see **Appendix 1**).

1 – <u>Raw ADC data samples</u>. If any sample in the trigger window is above the programmed threshold, all samples of the trigger window are reported. (Data type 4).

2 – <u>Pulse Raw ADC data samples</u>. A pulse is identified in the manner described above. Raw samples that constitute a detected pulse are reported. If the programmed width (NSB + NSA) is odd, an extra sample is reported. The threshold crossing sample number (TC) is also reported so that the pulse samples can be referenced to the start of the trigger window. Up to 3 pulses can be identified within the trigger window. (Data type 6.)

3 – <u>Pulse integral</u>. A pulse is identified in the manner described above. The sum of raw samples that constitute the pulse data set is reported. The threshold crossing sample number TC is reported as the pulse time. Up to 3 pulses can be identified within the trigger window. (Data types 7, 8).

4 – <u>High-resolution time</u>. A high-resolution time for the pulse is reported. Parameters used to compute this time are also reported. Up to 3 pulses can be identified within the trigger window. (Data types 8, 10).

7 – Pulse integral + high-resolution time. Equivalent to the processes of mode 3 and 4. Note that the time in mode 3 is replaced by the high-resolution time of mode 4. (Data types 7, 8, 10).

8 – Raw ADC data samples + high-resolution time. Equivalent to the processes of mode 1 and 4. (Data types 4, 8, 10.) (This mode is used to validate the implementation of the high-resolution time algorithm.)

Recommended modes of operation are 1, 2, 3, 7.

Firmware Implementation

As in most complex projects, the FADC250 firmware was developed in stages. Pulse detection mode 3, reporting the pulse integral and threshold crossing time TC, was delivered early. The high resolution timing algorithm was developed later and added as an independent process so it would not disrupt the functioning modes of operation.

As a consequence of having separate processes for pulse identification and timing, we have observed rare occasions where there is a mismatch in the number of pulses identified (pulse integral) and the number of pulse times reported. To be safe, we suggest that the user eliminate this data from consideration.

A future version of the firmware is planned that combines the pulse finding and high resolution algorithms into a single process. This will eliminate the pulse and time mismatches described above, and soften some of the restrictions we have imposed on when the high resolution timing mode can be applied. It will also increase trigger rate capability of the module.

Appendix 1. FADC250 Data Format (10/8/14) (CTRL FPGA ver > 23)

Data Type List

0 – block header
1 – block trailer
2 – event header
3 – trigger time
4 – window raw data
5 – (reserved)
6 – pulse raw data
7 – pulse integral
8 – pulse time
9 – (reserved)
10 – pulse parameters (pedestal)
11 – (reserved)
12 – scaler data
13 – (reserved)
14 – data not valid (empty module)
15 – filler (non-data) word

Data Word Categories

Data words from the module are divided into two categories: Data Type Defining (bit 31 = 1) and Data Type Continuation (bit 31 = 0). Data Type Defining words contain a 4-bit data type tag (bits 30 - 27) along with a type dependent data payload (bits 26 - 0). Data Type Continuation words provide additional data payload (bits 30 – 0) for the *last defined data type*. Continuation words permit data payloads to span multiple words and allow for efficient packing of raw ADC samples. Any number of Data Type Continuation words may follow a Data Type Defining word. The scaler data type is an exception. It specifies the number of 32-bit data words that follow.

Data Types

Block Header (0) – Word 1 indicates the beginning of a block of events. Optional continuation Word 2 contains ADC processing parameters.

Word 1:

(31) = 1
(30 – 27) = 0
(26 – 22) = slot number (set by VME64x backplane)
(21 – 18) = module ID (‘1’ for FADC250)
(17 – 8) = event block number
(7 – 0) = number of events in block

Word 2:

(31) = 0
(30 – 29) = 0
(28 – 18) = PL (# samples before trigger point for processing to begin)
(17 – 9) = NSB (# samples before threshold crossing to include in processing) (pulse modes)
(8 – 0) = NSA (# samples after threshold crossing to include in processing) (pulse modes)

Block Trailer (1) – indicates the end of a block of events.

(31) = 1
(30 – 27) = 1
(26 – 22) = slot number (set by VME64x backplane)
(21 – 0) = total number of words in block of events

Event Header (2) – indicates the start an event; includes the trigger number.

(31) = 1
(30 – 27) = 2
(26 – 22) = slot number (set by VME64x backplane)
(21 – 0) = event number (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). (Both Words or Word 2 alone may be suppressed from readout by the user.)

Word 1:

(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:

(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 may be set for any sample; e.g. the last reported sample is not valid when the window consists of an odd number of samples.

<u>Word 1</u>:

(31) = 1
(30 – 27) = 4
(26 – 23) = channel number (0 – 15)
(22 – 12) = reserved (read as 0)
(11 – 0) = window width (in number of samples)

<u>Words 2 - N</u>:

(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 Raw Data (6) – raw ADC data samples for an identified pulse. Raw data from an interval of the trigger window that includes the pulse is provided. The first word indicates the channel number, pulse number, and sample number of the threshold crossing (TC). Up to 3 pulses may be identified for each channel. Multiple continuation words contain two raw 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 may be set for any sample; for example, the last reported sample is tagged as not valid when the pulse interval consists of an odd number of samples.

<u>Word 1</u>:

(31) = 1
(30 – 27) = 6
(26 – 23) = channel number (0 – 15)
(22 – 21) = pulse number (0 – 3)
(20 – 10) = reserved (read as 0)
(9 – 0) = sample number of threshold crossing (TC)

<u>Words 2 - N</u>:

(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 Integral (7) – sum of raw samples that constitute the pulse data set.
(31) = 1
(30 – 27) = 7
(26 – 23) = channel number (0 – 15)
(22 – 21) = pulse number (0 – 3)
(20 – 19) = measurement quality factor (0 – 3)
(18 – 0) = pulse integral

Pulse Time (8) – time associated with an identified pulse.
(31) = 1
(30 – 27) = 8
(26 – 23) = channel number (0 – 15)
(22 – 21) = pulse number (0 – 3)
(20 – 19) = measurement quality factor (0 – 3)
(18 - 15) = reserved (read as 0)
(14 – 6) = pulse coarse time
(5 – 0) = pulse fine time

Pulse Pedestal (10) – pedestal and peak value associated with an identified pulse.
(31) = 1
(30 – 27) = 10
(26 – 23) = channel number (0 – 15)
(22 – 21) = pulse number (0 – 3)
(20 – 12) = pedestal (average of first 4 samples of window)
(11 – 0) = pulse peak value

Scaler Header (12) – indicates the beginning of a block of scaler data words. The number of scaler data words that will immediately follow it is provided in the header. The scaler data words are 32 bits wide and so have no bits available to identify them. Currently there are 18 scaler words reported: 16 from individual channels, a timer, and a trigger count. The scalers and time represent values recorded at the indicated trigger count. Scaler data must be enabled into the data stream by the user.

(31) = 1
(30 – 27) = 12
(26 – 6) = reserved (read as 0)
(5 – 0) = number of scaler data words to follow (18 = current)

Data Not Valid (14) – module has no valid data available for read out.

(31) = 1
(30 – 27) = 14
(26 – 22) = slot number (set by VME64x backplane)
(21 – 0) = undefined

Filler Word (15) – non-data word appended to the block of events. Forces the total number of 32-bit words read out of a module to be a multiple of 2 or 4 when 64-bit VME transfers are used. **This word should be ignored**.

(31) = 1
(30 – 27) = 15
(26 – 22) = slot number (set by VME64x backplane)
(21 – 0) = undefined