Implementation of the QT Algorithm for the STAR VPD: Run 2016

Chris Perkins
Eleanor Judd
06/14/2016

QT Code Version: 0x76
MCS File: qt32b_l0_v7_6.mcs

Description:

This algorithm forms a 12-bit ADC Sum, a 16-bit TAC Sum and a 4-bit Hit Count. Only channels that satisfy the "good hit" requirements are included in the Sums and Hit Count. A "good hit" is defined as one where the ADC value is greater than some threshold and the corresponding slew- and noise-corrected TAC value is greater than TAC_MIN and less than TAC_MAX. The channel mask register can be used but note that ADC and TAC channels must each be masked individually. Channels 0 (ADC) and 4 (TAC) are used to implement a noise monitor so that pair of channels is not included in either the sum or mean logic.

An outline of the steps followed by this algorithm is listed below, with details of each step described later in this section:

1. Slew and Noise Correct each TAC Channel
2. Apply channel masks
3. Check for overflow/underflow conditions
4. Apply "Good Hit" Requirement
5. Count good hits
6. Sum good ADC values and TAC values

A slew correction is applied to each TAC channel based on the value of the corresponding ADC channel. In the current implementation, there are a maximum of eight ADC bins. The ADC bin limits for each TAC channel can be defined independently. The ADC bin limits must cover the full available range of ADC values [0:4095] and must not overlap. Therefore any ADC value falls into exactly one ADC bin. The determination of which bin an ADC value falls into is done using the following logic:

$$Bin(X) = bin\_limit(X-1) < ADC <= bin\_limit(X)$$

Note that the lower limit of Bin(0) is hardwired to be 0, but the user has the ability to set all the other limits.

A slew correction offset is associated with each ADC bin of each channel. The slew correction offset is a signed integer with a range [-256:255]. The slew correction offset for this corresponding bin is then added to the raw TAC value. If the slew correction offsets are all set to 0 (the power-on default) then the slew correction is effectively turned off.

In parallel a noise correction is also applied to each TAC channel based on the behavior of the TAC monitor signal in channel 4. The noise correction is calculated from the current value of the TAC monitor minus a user-settable constant offset. The noise correction is then subtracted from each raw TAC value. The noise correction logic can be switched off entirely, via another user-settable register, if it is not needed.

The full correction logic therefore has the form:

Noise correction = monitor TAC - offset
Corrected TAC = raw TAC + slew correction (ADC) – Noise correction

If the result of applying these two corrections is negative, a corrected TAC value of '0' is used. If the result is greater than 4095, a corrected TAC value of '4095' is used. This ensures that the slew- and noise- corrected TAC values have the same range as the raw TAC values (i.e. [0:4095]).

The standard QT mask registers can be used for each channel to mask out that channel from the trigger but retain the data in the data-stream. The channel masks are applied AFTER the slew and noise correction. Separate masks must be used for ADC and TAC channels. Note that the mask for channel 0 is now ignored (even though the user can read/write that bit) because that channel is no longer used by the algorithm. Also the mask for channel 4 (the TAC monitor) MUST be set so that channel is included in the trigger and not masked out.

This algorithm then uses the standard "Good Hit" definition, which requires that the ADC value for a channel is greater than some **ADC_th** while the corresponding corrected TAC value is greater than some **TAC_Min** and less than some **TAC_Max**. The good hits are counted. The ADC values of the good hits are summed and separately so are the TAC values. The results are delayed appropriately so they can then be combined with the information that has been passed down from the preceding QT8 daughter card, and the final results are passed on to the next daughter card in the chain or the L0 FPGA on the mother board. The ADC sum is truncated to its 12 least significant bits to ensure there are always enough bits available in the chain for the hit count and TAC sum. If the full ADC sum exceeds 4095 then a truncated value of 4095 is used.

Inputs:

All daughter cards:

Ch. 0: Noise-monitor ADC
Ch. 1:3: PMT ADC
Ch. 4: Noise-monitor TAC
Ch. 5:7: PMT TAC

Registers (1 Set/Daughter Card, GUI Register Name in **bold**)**:**

(Reg. 11): **Channel_Mask** (8 bit mask)
Bit0 = channel 0 on (0) or off (1)
Bit1 = channel 1 on (0) or off (1)
…
Bit7 = channel 7 on (0) or off (1)

Alg. Reg. 0 (Reg. 13): "Good Hit" **ADC_Th** (12 bits, unsigned)
Alg. Reg. 1 (Reg. 14): "Good Hit" **TAC_Min** (12 bits, unsigned)
Alg. Reg. 2 (Reg. 15): "Good Hit" **TAC_Max** (12 bits, unsigned)
Alg. Reg. 3 (Reg. 16): **Noise_Control** (1 bit mask)
Bit 0 = Noise correction off (0) or on (1)
Alg. Reg. 4 (Reg. 17): **Noise_Offset** (12 bits, unsigned)

LUT:

TAC timing adjustment/ADC Pedestal subtraction for each channel

Slew Correction: 8 ADC Bins/Slew Correction Offsets per TAC channel

Algorithm Latch: 4

L0 Output to DSM:

(0-11): ADC Sum
(12-15): Hit Count
(16-31): TAC Sum

Actions:

Tick	QT8A	QT8B	QT8C	QT8D
1	Latch inputs	Same as QT8A	Same as QT8A	Same as QT8A
2	Find ADC bins for slew correction Calculate noise correction Delay ADC and TAC values	Same as QT8A	Same as QT8A	Same as QT8A
3	Calculate/Latch slew- and noise-corrected TAC values. Apply Channel_Mask	Same as QT8A	Same as QT8A	Same as QT8A
4	Overflow-Underflow mask corrected TAC values	Same as QT8A	Same as QT8A	Same as QT8A
5	ADC > R0 -> ADC_GOOD TAC > R1 -> TAC_MIN_GOOD TAC < R2 -> TAC_MAX_GOOD	Same as QT8A	Same as QT8A	Same as QT8A
6	Combine GOOD info -> GOOD hits Latch ADC/TAC for GOOD hits	Same as QT8A	Same as QT8A	Same as QT8A
7	Count GOOD hits Sum ADC: Ch2 + Ch 3 Sum TAC: Ch 6 + Ch 7 Delay Ch1 (ADC) and Ch 5 (TAC)	Same as QT8A	Same as QT8A	Same as QT8A
8	Sum ADC: Add in Ch1 Sum TAC: Add in Ch 5 Delay hit count	Same as QT8A	Same as QT8A	Same as QT8A
9	Latch hit count and ADC/TAC sums	Delay counts and sums	Delay counts and sums	Delay counts and sums
10	Delay final counts and TAC sum Truncate ADC Sum to 12 LSB	Delay	Delay	Delay
11	Latch out sums and hit count	Delay	Delay	Delay
12		Latch in sums and counts from upstream QT8 Latch local hit count and ADC/TAC sums	Delay	Delay
13		ADC Sum: Local + Upstream, truncated TAC Sum: Local + upstream Hit count: Local + upstream	Delay	Delay
14		Latch out sums and hit count	Delay	Delay
15			Latch in sums and counts from upstream QT8 Latch local hit count and ADC/TAC sums	Delay
16			ADC Sum: Local + Upstream, truncated TAC Sum: Local + upstream Hit count: Local + upstream	Delay
17			Latch out sums and hit count	Delay
18				Latch in sums and counts from upstream QT8 Latch local hit count and ADC/TAC sums
19				ADC Sum: Local + Upstream, truncated TAC Sum: Local + upstream Hit count: Local + upstream
20				Latch out sums and hit count