

# Implementation of QT Algorithm for the EPD Inner Tiles qt32c\_10\_v7\_8.mcs

Chris Perkins

Eleanor Judd

01/10/2018

## Description:

This algorithm forms a 12-bit TAC Max and a Hit Count. Only channels that satisfy a “good hit” requirement are included in the logic. A “good hit” is defined as one where the ADC value is greater than some threshold and the corresponding slew-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.

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

1. Slew Correct each TAC Channel
2. Apply channel masks
3. Check for overflow/underflow conditions
4. Apply “Good Hit” Requirement
5. Count good hits and select largest TAC value

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:

$$\text{Bin}(X) = \text{bin\_limit}(X-1) < \text{ADC} \leq \text{bin\_limit}(X)$$

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.

If the result of applying the correction 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-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 correction. Separate masks must be used for ADC and TAC channels.

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 corrected TAC values of the good hits are sorted to select the largest. 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. Those final results include a truncated hit count as well as the full hit count. The truncated hit count is calculated from “full hit count – 1”. It is set to zero if the full hit count is itself zero. This reduces the integer size from 5 bits (range 0:16) down to 4 bits (range 0:15). The truncated hit count is ignored by all the daughter cards, but it is used by the L0 FPGA. That FPGA packs the 4-bit truncated hit count onto one output cable with the 12-bit TAC Max value. The data can go to one DSM, which can reconstruct the full hit count by using the fact that a non-zero TAC Max implies there was at least one good hit. The 5-bit full hit count can be driven on another cable to another DSM.

**Inputs:**

QT8A: 4 PMT ADC, 4 PMT TAC

QT8B: 4 PMT ADC, 4 PMT TAC

QT8C: 4 PMT ADC, 4 PMT TAC

QT8D: 4 PMT ADC, 4 PMT TAC

**Registers (1 Set Per Daughter Card):**

Alg. Reg. 0 (Reg 13): ADC\_Threshold

Alg. Reg. 1 (Reg 14): TAC\_MIN

Alg. Reg. 2 (Reg 15): TAC\_MAX

Reg. 11: Channel Mask

**LUT:**

Timing adjustments/pedestal subtraction for each PMT

**Algorithm Latch:** 1 or 2

**L0 Output to DSM:**

(0-4) : Full Hit Count

(5-15) : Unused

(16-27) : TAC Max

(28-31) : Truncated Hit Count

### 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 Delay ADC and TAC values	Same as QT8A	Same as QT8A	Same as QT8A
3	Calculate/Latch slew-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 TAC for GOOD hits	Same as QT8A	Same as QT8A	Same as QT8A
7	Select TAC: Ch 4 vs Ch 5 Select TAC: Ch 6 vs Ch 7 Count Hits: Ch0/Ch1 Count Hits: Ch2/Ch3	Same as QT8A	Same as QT8A	Same as QT8A
8	Select TAC: Select largest TAC Count Hits: Add intermediate counts	Same as QT8A	Same as QT8A	Same as QT8A
9	Latch hit count and largest TAC	Delay hit count and TAC	Delay count, sum and TAC	Delay count, sum and TAC
10	Truncate hit count Delay full hit count and largest TAC	Delay	Delay	Delay
11	Latch out both counts and TAC info	Delay	Delay	Delay
12		Latch in full hit count and TAC from upstream QT8 Latch local info	Delay	Delay
13		Select TAC: Local + upstream Count Hits: Local + upstream Truncate full hit count	Delay	Delay
14		Latch out both counts and TAC info	Delay	Delay
15			Latch in full hit count and TAC from upstream QT8 Latch local info	Delay
16			Select TAC: Local + upstream Count Hits: Local + upstream Truncate full hit count	Delay
17			Latch out both counts and TAC info	Delay
18				Latch in full hit count and TAC from upstream QT8 Latch local info
19				Select TAC: Local + upstream Count Hits: Local + upstream Truncate full hit count
20				Latch out both counts and TAC info