Implementation of QT Algorithm for ZDC
Run 2011
qt32b_l0_v6_2.mcs

Chris Perkins

03/03/2011

Description:

This algorithm compares various ADC sums to thresholds, passes two separate partial TAC values, and passes two separate partial ADC sums.

Only channels that satisfy a "good hit" requirement are included in sums for threshold comparisons, TAC output, and sum output. A "good hit" is defined as one where the ADC value is greater than some threshold and the corresponding TAC value is greater than TAC_MIN and less then TAC_MAX. The channel mask register can be used but note that ADC and TAC channels must each be masked individually.

Note that only the first two ADC and TAC channels are used on each daughter card. The other channels will show up in the datastream but are not considered in the trigger decision.

The first sum considered is channel 0 + 1 on each daughter card. This is compared to Pair_Threshold and one bit per daughter card is output. The second sum considered is channel 0 + 1 on daughter A plus channel 0 + 1 on daughter B. A similar sum is calculated from channels 0 + 1 on daughter C plus channels 0 + 1 on daughter D. These sums are compared to Sum_Threshold and two bits total are output from each QT32.

There are two separate partial TAC values output: the upper nine bits (2-11) from the first TAC channel on daughter A and the first TAC channel on daughter C, both subject to the corresponding "good hit" requirement on the full TAC value.

The algorithm also outputs the upper three bits (11:13) from the $(\text{channel } 0+1)_{AB}$ sum and the $(\text{channel } 0+1)_{CD}$ sum.

Note that this algorithm uses the direct path from Daughter B to the L0 FPGA.

The default masks for Run 2011 are 0xEE on daughters A and C, and 0xCC on daughters B and D. This makes the Pair thresholds as follows:

Pair A: ZDC E1
Pair B: ZDC E2+E3
Pair C: ZDC W1
Pair D: ZDC W2+W3

And the Sum thresholds as follows:

Sum A+B: ZDC E1+E2+E3
Sum C+D: ZDC W1+W2+W3

Inputs:

QT8A: 2 PMT ADC (ch 0,1), 2 PMT TAC (ch 4,5)
QT8B: 2 PMT ADC (ch 8,9), 2 PMT TAC (ch 12,13)
QT8C: 2 PMT ADC (ch 16,17), 2 PMT TAC (ch 20,21)
QT8D: 2 PMT ADC (ch 24,25), 2 PMT TAC (ch 28,29)

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
Alg. Reg. 3 (Reg 16): Pair_Threshold
Alg. Reg. 4 (Reg 17): Sum_Threshold (only valid on daughters B,D)
Reg. 11: Channel Mask

LUT:
Timing adjustments/pedestal subtraction for each PMT

Algorithm Latch: 1

Action (21x RHIC Clock)**:**

1st: Mask channels and Latch inputs
If mask bit = 1, channel data = 0

2nd: For each PMT (2 per daughter board):
ADC above threshold: ADC > PMT_ADC_Thresh → Good_ADC
TAC above threshold: TAC > TAC_MIN → Good_TAC_MIN
TAC below threshold: TAC < TAC_MAX → Good_TAC_MAX

3rd: Make good_hits(0-1):
good_hit(i) = Good-ADC(i) && Good_TAC_MIN(i) && Good_TAC_MAX(i)

4th: Sum ADC channels 0+1 subject to good hit requirements → Int_sum_0
Delay TAC channel 4 subject to good hit requirement → TAC_ch4_del

5th: Compare Int_sum_0 to Pair_Threshold → Pair_Good
Add Int_sum_0 to sum from previous daughter (input bits0-12) → Int_sum_1
(Note: This result is ignored on daughters A,C)
Delay TAC value from previous daughter → TAC_in_del
(Note: This result is ignored on daughters A,C)

6th: Compare Int_sum_1 to Sum_Threshold → Sum_Good
(Note: This result is ignored on daughters A,C)

7th: Latch Output Bits to next daughter or L0 FPGA

if(daughter A)

(0-12) : Int_sum_0 (Pair ADC Sum A)
(13-16) : '0'
(17-28) : TAC_ch4_del (DA ch4 TAC)
(29-32) : '0'
(33) : Pair_Good (A)

else if(daughter B)

(0-12) : '0'
(13-16) : TAC_in_del (bits8-11) (DA ch4 TAC)
(17-28) : '0'
(29) : Int_sum_1 (bit 13) (A+B sum)
(30) : '0'
(31) : Sum_Good (A+B)
(32) : Pair_Good (B)
(33) : Pair_Good (A) (Passed from previous daughter)
Level0_Out (2-7): TAC_in_del (bits2-7) (DA ch4 TAC)
Level0_Out (0-1): Int_sum_1 (bits 11-12) (A+B sum)

else if(daughter C)

(0-12) : Int_sum_0 (Pair ADC Sum C)
(13-16) : DA ch4 TAC (bits8-11) (Passed from previous daughter)
(17-28) : TAC_ch4_del (DC ch4 TAC)
(29) : A+B Sum (bit 13) (Passed from previous daughter)
(30) : Pair Good (C)
(31) : Sum_Good (A+B) (Passed from previous daughter)
(32) : Pair_Good (B) (Passed from previous daughter)
(33) : Pair_Good (A) (Passed from previous daughter)

else if(daughter D)

(0-11) : TAC_in_del (DC ch4 TAC)
(12) : '0'
(13-16) : DA ch4 TAC (bits8-11) (Passed from previous daughter)
(17) : A+B Sum (bit 13) (Passed from previous daughter)
(18-20) : C+D Sum (bits 11-13)
(21-27) : '0'
(28) : Sum_Good (C+D)
(29) : Pair Good (D)
(30) : Pair Good (C) (Passed from previous daughter)
(31) : Sum_Good (A+B) (Passed from previous daughter)
(32) : Pair_Good (B) (Passed from previous daughter)
(33) : Pair_Good (A) (Passed from previous daughter)

L0 Output to DSM:

(0-9) : Daughter C ch4 TAC (bits 2-11)
(10-19) : Daughter A ch4 TAC (bits 2-11)
(20) : Pair Good (C)
(21) : Pair Good (D)
(22) : Sum Good (C+D)
(23) : Pair Good (A)
(24) : Pair Good (B)
(25) : Sum Good (A+B)
(26-28) : C+D Sum (bits 11-13)
(29-31) : A+B Sum (bits 11-13)