Algorithms for Vertex QT-DSM Tree during Proton-Proton Collisions RHIC 2015 Run

Eleanor Judd
Chris Perkins

February 24, 2015

Change Log:

Date	Description
February 22. 2009	First version for PP Collisions where the original CTB and layer-0 DSM boards have been replaced with QT boards. Also, the VPD has been added into this branch of the DSM tree since it is designed to detect the same vertex as the BBC and ZDC.
March 31, 2010	BB102 algorithm changed. The BB102 QT board is now running an algorithm specific to the BBC large tiles. Its output is two TAC values and two hit flags, one pair for the East side and the other for the West side. The BB102 DSM algorithm is now a simplification of the BB101 algorithm, which calculates a TAC difference and selects bits to send to the scaler system.
January 11, 2011	First version for 2011 proton-proton data taking. The ZD101 algorithm has been changed. The size of the TAC values produced by the QT boards have been reduced from 12 to 10 bits, to make room for truncated sums from each of the East and West sides. The new ZD101 algorithm passes those sums through to the scaler board, instead of passing some of the TAC bits. The ZD101 output to VT201 has been kept the same as in 2009, so the VT201 algorithm from 2009 can still be used
March 14, 2012	Changed the ZD101 algorithm. The TAC overflow and underflow bits going to the scaler system have been combined. This frees up space for the AND of the threshold bits from the East and West sides.
March 5, 2013	Changed the VT201 algorithm for the 2013 running. The two ZDC East/West threshold bits are combined into one coincidence bit to make space for a minimum bias bit.
June 14, 2013	No algorithm changes, but I added links to Chris's documentation for the QT algorithms.
January 16, 2015	Changed the VT201 algorithm for the 2015 running. The BBC-large TAC bit has been dropped from the output. The VPD threshold bits have been combined into one coincidence bit. The ZDC East/West threshold bits have been separated out from their coincidence bit and a 2nd VPD TAC difference bit has been added.
February 24, 2015	Changed the VT201 algorithm. The 2 BBC-large threshold bits have been dropped from the output and replaced with the ZDC-West-Front bit and the 3rd VPD TAC window. The ZDC-West-Front bit was moved because it was not working in its original location. That non-working bit (11) is now unused.

The Vertex branch of the DSM tree is used to locate the primary vertex of the RHIC beam collisions at STAR. All three relevant trigger detectors connect to this branch: Zero Degree

Calorimeters (ZDC), Beam-Beam Counters (BBC) and the Vertex Position Detector (VPD). The raw detector signals are digitized and pre-processed in QT boards. The DSM tree is then used to calculate TAC differences and combine ADC information to produce the data needed for effectively triggering on proton-proton collisions.

Layer 0 QT Boards: BBQ_BB001:002

There are two BBC small-tile QT boards: one processes data from the East side of the detector and the other from the West side. This algorithm forms a 16bit ADC Sum and 12bit TAC Max. Only channels that satisfy a "good hit" requirement are included in the ADC Sum and TAC Max. 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 than TAC_MAX. The channel mask register can be used to force the algorithm to ignore certain channels, but note that ADC and TAC channels must each be masked individually. The algorithm was written by Chris Perkins and is documented at http://www.star.bnl.gov/public/trg/TSL/Software/qt_v5_6_doc.pdf

1. Layer 1 DSM Boards: BBC_BB101

The BB101 DSM board processes data from the BBC small-tile detector. The algorithm receives ADC-sum and fastest-TAC data from the QT boards. The ADC sums are compared to thresholds. A set of bits specified by the user is chosen from each incoming TAC value to send to the scaler system. In parallel, the TAC difference is calculated. The difference is set to zero if either of the two incoming TACS is zero, because a TAC value of zero implies there were no good hits on that side of the BBC.

RBT File: bbc_bb101_2009_a.rbt

Users: BB101

Inputs: Ch0/1 = QT Board BB001 (East)
Ch2/3 = QT Board BB002 (West)
Ch4/7 = Unused

From each QT board:
bits 0:15 = ADC-Sum
bits 16:27 = Max TAC (Value of zero implies NO good hits)

LUT: 1:1

Registers:
Four registers, all thresholds can be set independently
R0: BBCsmall-EastADCsum_th (16 bits)
R1: BBCsmall-WestADCsum_th (16)
R2: BBCsmall-EastTAC-select (3)
0 => select bits 0:6
1 => select bits 1:7
…
5 => select bits 5:11
R3: BBCsmall-WestTAC-select (3)

Same value definitions as for R2

Action:

1st Latch input

2nd Compare each ADC-sum to its threshold
Calculate: TAC difference = 4096 + TAC-E – TAC-W
Define: Good-TAC-E = TAC-E > 0, same for West side
Make all possible bit selections from TAC-E and TAC-W, including overflow logic. For example:
TAC-E-overflow-0 = TAC-E(7), (8), (9), (10) or (11)
If (TAC-E-overflow-0 = 1) then TAC-E-scaler-0 = 127
Else TAC-E-scaler-0 = TAC-E(0:6)
Same logic for all possible bit selections from TAC-E (see description of register R2) and TAC-W

3rd Delay ADC-sum threshold bits
Zero out TAC difference if either Good-TAC-E or Good-TAC-W is false, otherwise just delay TAC difference
Use R2 to select the TAC-E scaler bits:
If (R2 = 0) then chose TAC-E-scaler-0
Else if (R2 = 1) then chose TAC-E-scaler 1
Etc…
Do the same for West side, using R3 to control the selection.

4th Latch output

Output to VT201:

(0-12) TAC difference
(13) Unused
(14) ADC-sum-E > th0
(15) ADC-sum-W > th0

Scalers:

(0-6) selected bits of TAC-E
(7-13) selected bits of TAC-W
(14) ADC-sum-E > th0
(15) ADC-sum-W > th0

2. Layer 0 QT Boards: BBQ_BB003

There is just one BBC large-tile QT board and it receives data from both the East and West sides of the detector. The algorithm was written by Chris Perkins and is documented at http://www.star.bnl.gov/public/trg/TSL/Software/qt_v5_f_doc.pdf

3. Layer 1 DSM Board: BBC_BB102

The BB102 DSM board processes data from the BBC-large-tile detector. The algorithm receives a hit flag and fastest-TAC data for each of the East and West sides of the detector from the QT board. The hit flags indicate there was at least one good hit on each side, and they

are just passed through to the output. A set of bits specified by the user is chosen from each incoming TAC value to send to the scaler system. In parallel, the TAC difference is calculated. The difference is set to zero if either of the two incoming TACS is zero, because a TAC value of zero implies there were no good hits on that side of the BBC.

RBT File: bbc_bb102_2010_b.rbt

Users: BB102

Inputs: Ch0/1 = QT Board BB003 (East and West)
Ch2/7 = Unused

From the QT board:
bits 0:11 = MAX TAC East (value of zero implies no good hits)
bits 12:23 = MAX TAC West
bit 24 = East hit
bit 25 = West hit

LUT: 1:1

Registers:
R0: BBClarge-EastTAC-select (3)
0 => select bits 0:6
1 => select bits 1:7
…
5 => select bits 5:11
R1: BBClarge-WestTAC-select (3)
Same value definitions as for R0

Action:
1st Latch input

2nd Delay hit bits to 4th step
Calculate: TAC difference = 4096 + TAC-E – TAC-W
Define: Good-TAC-E = TAC-E > 0, same for West side
Make all possible bit selections from TAC-E and TAC-W, including overflow logic. For example:
TAC-E-overflow-0 = TAC-E(7), (8), (9), (10) or (11)
If (TAC-E-overflow-0 = 1) then TAC-E-scaler-0 = 127
Else TAC-E-scaler-0 = TAC-E(0:6)
Same logic for all possible bit selections from TAC-E (see description of register R0) and TAC-W (see register R1)

3rd Zero out TAC difference if either Good-TAC-E or Good-TAC-W is false, otherwise just delay TAC difference to the 4th step
Use R0 to select the TAC-E scaler bits:
If (R0 = 0) then chose TAC-E-scaler-0
Else if (R0 = 1) then chose TAC-E-scaler-1
Etc…
Do the same for West side, using R1 to control the selection.

4th Latch output

Output to VT201:

(0-12) TAC difference
(13) Unused
(14) East hit
(15) West hit

Scalers:

(0-6) selected bits of TAC-E
(7-13) selected bits of TAC-W
(14) East hit
(15) West hit

4. Layer 0 QT Boards: BBQ_VP001:002

The two VPD QT boards use the same algorithm as the two small-tile BBC QT boards. See documentation above for BBQ_BB001:002.

5. Layer 1 DSM Board: BBC_VP101

RBT File: bbc_vp101_2009_a.rbt

Users: VP101

Inputs: Ch0/3 = Unused
Ch4/5 = QT Board VP003 (East)
Ch6/7 = QT Board VP004 (West)

The VP101 DSM board receives VPD data from 4 QT boards. However, currently only 2 of those boards (VP003 and VP004) produce data that needs to be analyzed by the trigger system. The logic needed to do this analysis is the same as that used by the BB101 algorithm. The VP101 algorithm is therefore identical to the BB101 algorithm in every way, except for the input map. Please see the BBC_BB101 documentation above for details of the logic.

6. Layer 0 QT Board: BBQ_ZD001

There most recent QT algorithm for use with Proton-Proton collisions was created in 2011. The algorithm was written by Chris Perkins and is documented at
http://www.star.bnl.gov/public/trg/TSL/Software/qt_v6_2_doc.pdf

7. Layer 1 DSM Board: BBC_ZD101

The ZD101 DSM board processes data from the ZDC detector. The algorithm receives TAC data from the QT boards and calculates the TAC difference. The difference is set to zero if either of the two incoming TACS is zero, because a TAC value of zero implies there were no good hits on that side of the ZDC. A user-specified set of bits is then chosen to be passed on to VT201. In parallel, the algorithm also receives the results of comparing various sums to thresholds, and truncated total sums. These threshold bits are passed through to both VT201 and the scaler system. The truncated sums are just passed to the scaler system.

RBT File: bbc_zd101_2012_a.rbt

Users: ZD101

Inputs: Ch0/1 = QT Board ZD001
Ch2:7 = Unused

From the QT board:
bits 0:9 = West-1 TAC
bits 10:19 = East-1 TACEast
bit 20 = Front West ADC sum > threshold
bit 21 = Back West ADC sum > threshold
bit 22 = Total West ADC sum > threshold
bit 23 = Front East ADC sum > threshold
bit 24 = Back East ADC sum > threshold
bit 25 = Total East ADC sum > threshold
bits 26:28 = Truncated Total West sum
bits 29:31 = Truncated Total East sum

LUT: 1:1

Registers:
R0: ZDC-TACdiff-select (2 bits)
0 => select bits 0:9
1 => select bits 1:10

Action:
1st Latch input

2nd Delay threshold bits to the 4th step.
Delay truncated sums to the 4th step.
Make combination: ADC-sum-E > th0 AND ADC-sum-W > th0
Calculate: TAC difference = 1024 + TAC-E – TAC-W
Define: Good-TAC-E = TAC-E > 0, same for West side

3rd Delay the ADC combination to the 4th step.
Use R0 to select the TAC difference bits for VT201, including overflow logic and the "good" TAC cut. Also, set the out-of-range bit, i.e.:

```
If (Good-TAC-E = 0 or Good-TAC-W = 0) then
        output = out-of-range = 0
Else if (R0 = 0)
        out-of-range = TAC-diff(10)
        If (TAC-diff(10) = 1) then output = 1023
        Else output = TAC-diff(0:9)
Else
        out-of-range =  TAC-diff(0)
        output = TAC-diff(1:10)
Endif
```

4th Latch output

Output to VT201:

(0-9) TAC difference
(10) ADC-sum-E > th0
(11) ADC-sum-W > th0
(12) Front-ADC-E > th0
(13) Back-ADC-E > th0
(14) Front-ADC-W > th0
(15) Back-ADC-W > th0

Scalers:

(0-2) Truncated ADC-sum-E
(3-5) Truncated ADC-sum-W
(6) Good-TAC-E
(7) Good-TAC-W
(8) TAC-out-of-range
(9) ADC-sum-E > th0 AND ADC-sum-W > th0
(10) ADC-sum-E > th0
(11) ADC-sum-W > th0
(12) Front-ADC-E > th0
(13) Back-ADC-E > th0
(14) Front-ADC-W > th0
(15) Back-ADC-W > th0

8. Layer 2 Vertex DSM Board: L1-VT201

All information from the large and small-tile BBC, the ZDC and the VPD are brought into the Vertex DSM. The bits from the large BBC tiles are no longer used. The threshold bits from the other detectors are passed on to the TCU. In parallel windows are placed around each TAC difference, and the "inside window" bits get passed through to the TCU and the scaler system. Various TAC and threshold bits are also combined to make a minimum bias bit. The four MSB of the TAC difference from the BBC small-tiles, and the ZDC are also in the scaler output.

RBT File: l1_vt201_2015_c.rbt

Users: VT201

Inputs: Ch 0 = BB101
Ch 1 = Unused (BB102)
Ch 2 = ZD101
Ch 3 = Unused
Ch 4 = VP101
Ch 5:7 = Unused

From Small tile BBC-DSM BB101
(0-12) Small tile TAC-Difference
(13) Unused
(14/15) Small tile ADC East/West sum > th0

From ZDC DSM ZD101
(0-9) ZDC TAC-Difference
(10) ZDC East ADC sum > th0
(11) ZDC West ADC sum > th0
(12) ZDC East Front ADC > th0
(13) ZDC East Back ADC > th0
(14) ZDC West Front ADC > th0
(15) ZDC West Back ADC > th0

From VPD-DSM VP101
(0-12) VPD TAC-Difference
(13) Unused
(14/15) VPD ADC East/West> th0

LUT: Either 1-to-1 or TAC-difference range conversion

Registers:
R0: BBCsmall-TACdiff-Min (13 bits)
R1: BBCsmall-TACdiff-Max (13)
R2: VPD-TACdiff3-Min (13)
R3: VPD-TACdiff3-Max (13)
R4: ZDC-TACdiff-Min (10)
R5: ZDC-TACdiff-Max (10)
R6: VPD-TACdiff-Min (13)
R7: VPD-TACdiff-Max (13)
R8: Minimum-Bias-Select (3)
R9: VPD-TACdiff2-Min (13)
R10: VPD-TACdiff2-Max (13)

Action

1st Latch inputs

2nd Compare each of the 3 TAC differences to its minimum and maximum value, as specified in the relevant registers. The logic looks for the TAC difference to be greater than the minimum and less than the maximum.

3rd Combine the results of the TAC difference comparisons to determine if each TAC difference is inside its specified window, e.g.:
ZDC-TAC-diff-in-window = R4 < ZDC TAC difference < R5
Combine (AND) the VPD threshold bits to make the VPD coincidence bit.
Combine the results of the TAC difference comparisons and the ADC threshold bits to make the minimum bias bit, using R8 to turn each component on/off, i.e.:

MB = (R8(0) and BBC-S-Tdiff and BBC-S-E>th0 and BBC-S-W>th0) or
(R8(1) and ZDC-Tdiff) or
(R8(2) and VPD-Tdiff)

4th Latch Outputs

Output to TCU:

Bit	Name	Description
Bit 0	BBC-TAC	BBC small-tile TAC difference in window
Bit 1	BBC-E	BBC small-tile East ADC sum > threshold
Bit 2	BBC-W	BBC small-tile West ADC sum > threshold
Bit 3	ZDC-W	ZDC West ADC sum > threshold
Bit 4	ZDC-W-Front	ZDC West Front ADC sum > threshold
Bit 5	VPD-TAC3	VPD TAC difference in 3rd window
Bit 6	ZDC-TAC	ZDC TAC difference in window
Bit 7	ZDC-E	ZDC East ADC sum > threshold
Bit 8	Minimum-Bias	At least one selected minimum bias component satisfied.
Bit 9	ZDC-E-Front	ZDC East Front ADC sum > threshold
Bit 10	ZDC-E-Back	ZDC East Back ADC sum > threshold
Bit 11	Unused	BROKEN BIT – DO NOT USE
Bit 12	ZDC-W-Back	ZDC West Back ADC sum > threshold
Bit 13	VPD-TAC	VPD TAC difference in window
Bit 14	VPD-TAC2	VPD TAC difference in 2nd window
Bit 15	VPD-COINC	VPD East ADC sum > threshold AND VPD West ADC sum > threshold

Output to Scalers

Bit	Description
Bit 0	BBC small-tile TAC difference in window
Bits 1:4	4 MSB of BBC small-tile TAC difference
Bit 5	Unused
Bit 6	ZDC TAC difference in window
Bits 7:10	4 MSB of ZDC TAC difference
Bit 11	VPD TAC difference in window
Bit 12	VPD TAC difference in 2nd window
Bit 13	Minimum Bias
Bits 14:15	Unused