Algorithms for Vertex QT-DSM Tree during Proton-Proton Collisions RHIC 2022 Run

Eleanor Judd
Chris Perkins

December 20, 2021

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.
February 7, 2017	First changes for 2017. The ZD101 algorithm has been upgraded so it is consistent with the hybrid ZDC QT algorithm and it can deal with pp, pA and Ap running, using a register to switch between different modes.

	The VT201 algorithm has been modified for this year's requirements. The minimum bias bit and the 3rd VPD TAC window have been dropped. The BBC-large threshold bits have been restored and an EPD bit has been added.
February 24, 2017	The VT201 algorithm has been modified: some unused scaler output bits have been removed, and the remaining scaler bits are now gated with the board mode signal. This fixes an internal problem with the timing of the VPD coincidence bit.
November 15, 2021	The VT201 algorithm has been modified: the BBC-L and ZDC-Front/Back bits have been removed. The full set of EPD bits has been added and the VPD coincidence bit is replaced with the separate East and West thresholds. In addition the full set of EPD algorithm documentation has been added and the BBC Large-Tile documentation has been removed.
December 13, 2021	The VT201 algorithm has been modified: the 2nd EPD TAC window has been removed. The ZDC Front/Back bits have been restored as coincidences.
December 16, 2021	No logic changes. The ZDC SMD QT input to ZD101 has been listed.
December 20, 2021	The VT201 algorithm has been modified again. The remaining TCU output bits have been added into the scaler output.

The Vertex branch of the DSM tree is used to locate the primary vertex of the RHIC beam collisions at STAR. All four relevant trigger detectors connect to this branch: Zero Degree Calorimeters (ZDC), Beam-Beam Counters (BBC), Event Plane Detector (EPD) 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 and hit information to produce the data needed for effectively triggering on proton-proton collisions.

1. BBC QT Boards: BBQ_BB001:002

There are two BBC QT boards: one processes data from the East side of the detector and the other from the West side.
For a detailed description of this algorithm please see the documentation at
http://www.star.bnl.gov/public/trg/TSL/Software/qt_v6_d_doc.pdf
The algorithm forms a 16-bit ADC Sum and a 12-bit 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 slew-corrected 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.

2. 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
…
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

3. VPD QT Boards: BBQ_VP001:002

The two VPD QT boards in the Vertex branch of the DSM tree use algorithm as 6d, which is documented at http://www.star.bnl.gov/public/trg/TSL/Software/qt_v6_d_doc.pdf .

4. VPD 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)

LUT: 1:1

Registers:

Four registers, all thresholds can be set independently
R0: VPD_vtx_EastADCsum-th (16 bits)
R1: VPD_vtx_WestADCsum-th (16)
R2: VPD_vtx_EastTAC-select (3)
0 => select bits 0:6
…
5 => select bits 5:11
R3: VPD_vtx_WestTAC-select (3)
Same value definitions as for R2

The VP101 DSM board receives VPD data from 2 QT boards: VP001 and VP002. 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.

5. ZDC QT Board: BBQ_ZD001

The single ZDC QT board receives signals from both the East and West sides of the ZDC. For each side separately the algorithm can be configured to compare the ZDC-Front and ZDC-Back ADC values, and their digital sum, to a threshold (the "proton" logic) or to compare the analog sum to multiple thresholds (the "heavy ion" logic). It should be noted that the proton logic uses the "good hit" requirement for all hits (ADC > threshold and associated TAC in window) but the heavy ion logic does not. In the current setup both sides are configured to use the proton logic. The algorithm was written by Chris Perkins and is documented at http://www.star.bnl.gov/public/trg/TSL/Software/qt_v6_f_doc.pdf

6. ZDC Layer 1 DSM Board: BBC_ZD101

The ZD101 DSM board processes data from the ZDC detector. The algorithm receives TAC data from the QT board and calculates the TAC difference. It 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 ADC sums to thresholds. If either side (East or West) is in heavy-ion mode then those threshold bits are zeroed out if the relevant TAC value is zero. This is not necessary in proton mode because the full good-hit logic was applied in the QT board. The resulting threshold bits are passed through to both VT201 and the scaler system. In the current setup both sides are configured to be in proton mode.
NOTE: The ZD101 DSM board also receives a copy of scaler bits that are generated by the ZDC SMD QT board. These bits are not used by the algorithm; they are stored and read out from ZD101 to enable bit-checking of the ZDC SMD QT algorithm.

RBT File: bbc_zd101_2017_a.rbt

Users: ZD101

Inputs: Ch0/1 = QT Board ZD001
Ch2 = QT Board PX001 (ZDC SMD, unused in this algorithm)
Ch3:7 = Unused

From the ZD001 QT board:
bits 0:9 = West-1 TAC
bits 10:19 = East-1 TAC
bits 20:25 = West sum/threshold bits
bits 26:31 = East sum/threshold bits

The definition of the threshold bits depends on which mode was used by the QT logic.

Bit #	**Proton Mode: Used for Run22**	Heavy-Ion Mode
20	**Truncated West digital sum**	West analog sum > th0
21		West analog sum > th1
22		West analog sum > th2
23	**Front West digital sum > th**	West analog sum > th3
24	**Back West digital sum > th**	West attenuated analog sum > th4
25	**Total West digital sum > th**	West attenuated analog sum > th5
26	**Truncated East digital sum**	East analog sum > th0
27		East analog sum > th1
28		East analog sum > th2
29	**Front East digital sum > th**	East analog sum > th3
30	**Back East digital sum > th**	East attenuated analog sum > th4
31	**Total East digital sum > th**	East attenuated analog sum > th5

LUT: 1:1

Registers:
R0: ZDC-TACdiff-select (2 bits)
0 => select bits 0:7
1 => select bits 1:8
2 => select bits 2:9
3 => select bits 3:10
R1: ZDC_EW_Mode_Select (2 bits)
Bit 0: East side - Proton mode (0) or Heavy Ion mode (1)
Bit 1: West side – Proton mode (0) or Heavy Ion mode (1)

Action:
1st Latch input

2nd Delay all threshold/sum bits to the 3rd step.
Calculate: TAC difference = 1024 + TAC-E – TAC-W
Define: Good-TAC-E = TAC-E > 0, same for West side

3rd Use R0 to select the TAC difference bits for VT201, including overflow logic and the “good” TAC cut. Also, set the overflow bit, i.e.:
The output is 0 if either Good-TAC bit is 0

The output is 255 (maximum) if any higher order bits above the maximum selected bit are set. In this case the overflow bit is also set to 1.

Otherwise the output is set to the selected bits.

Delay the Good-TAC bits to the 4th step.

Use R1 to select which threshold bits are passed to VT201 and the Scaler system. In heavy-ion mode the bits are masked with the relevant Good-TAC bit.

Bits to VT201	**Proton Mode: Used in Run 22**	Heavy-Ion Mode
1st	**0**	Good Analog sum > th0
2nd	**Front digital sum > th**	Good Analog sum > th1
3rd	**Back digital sum > th**	Good Analog sum > th2
4th	**Total digital sum > th**	Good Analog sum > th3
Bits to Scalers		
1st to 6th	**All 6 input bits**	All 6 input bits

4th Latch output

Output to VT201:

(0-7) TAC difference
(8-11) West threshold bits
(12-15) East threshold bits

Scalers:

(0) Good-TAC-W
(1) Good-TAC-E
(2) TAC-overflow
(3) 0 (Unused)
(4-9) West sum/threshold bits
(10-15) East sum/threshold bits

7. EPD Inner-tile QT Boards: EQ1, EQ2 and EQ3 crates

There are 14 EPD QT boards for the inner tiles. 7 process data from the East side of the detector and the other 7 cover the West side. The algorithm is a variation of the traditional fastest TAC algorithm used by the BBC small-tile detector. It calculates a hit count instead of an ADC sum. Please see the documentation provided by Eleanor Judd for a detailed description of this algorithm at
http://www.star.bnl.gov/public/trg/TSL/Software/qt_v7_8_doc.pdf
The output consists of a truncated Hit Count (range 0-15, 4 bits) and a 12-bit Max TAC packed onto one output cable. Only channels that satisfy a "good hit" requirement are included in the Hit Count and Max TAC. 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 to force the algorithm to ignore certain channels, but note that ADC and TAC channels must each be masked individually.

8. EPD Inner-tile Layer 0 DSM Boards: BBC_EP001:2

The EP001:2 DSM boards process data from the East and West sides of the inner EPD detector respectively. The algorithm receives a truncated Hit Count and fastest-TAC data from each of 7 QT boards. It will combine the data to produce a total (not truncated) hit count, the fastest TAC value from channels 0:3 and from channel 4:7. Register 0 can be used to turn each input channel on or off in the logic because only 7 of the 8 input channels are used.

RBT File: bbc_ep001_2018_a.rbt

Users: EP001:2

Inputs: Ch0/7 = 2nd output cable from each EPD inner-tile QT board
NOTE: Only 7-of-8 input channels will be used

From each QT board:
bits 0:11 = Max TAC (Value of zero implies NO good hits)
bits 12:15 = Truncated Hit Count

LUT: 1:1

Registers:
R0: EPD_EP001_ChSelect (8 bits)
Bit 0: Turn Ch-0 on (1) or off (0)
Bit 1: Turn Ch-1 on (1) or off (0)
Etc…

Action:
1st Latch input

2nd For each channel (X) calculate the masked un-truncated hit count:
If R0(X) = 0 then hit count = 0
Else if TAC = 0 then hit count = 0
Else hit count = truncated hit count + 1
Mask out the TAC values from those channels that are turned off by R0.

3rd Sum the hit count values
Select the fastest (largest) masked TAC value from channels 0:3
Select the fastest (largest) masked TAC value from channels 4:7

4th Latch output

Output to EP101:
(0-11) Max TAC from channels 0:3
(12-23) Max TAC from channels 4:7
(24-31) Total Hit Count

9. EPD Inner-tile Layer 1 DSM Board: BBC_EP101

The EP101 DSM board processes data from the inner EPD detector. The algorithm receives total Hit Count and fastest-TAC data for the East and West sides from the 2 EPD Layer 0 DSM

boards. The Hit Counts are compared to thresholds. They are also summed, and the sum is compared to its own threshold. In parallel, the 2 fastest TAC values from each side are compared to find the fastest TAC from that side. The TAC difference between the two sides is then 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 EPD.

RBT File: bbc_ep101_2021_a.rbt

Users: EP101

Inputs: Ch0/1 = DSM Board EP001 (East)
Ch2/3 = DSM Board EP002 (West)
Ch4/7 = Unused

From each DSM board:
(0-11) Max TAC from channels 0:3
(12-23) Max TAC from channels 4:7
(24-31) Total Hit Count

LUT: 1:1

Registers:
Three registers, all thresholds can be set independently
R0: EPD_EastHitCnt_th (8 bits)
R1: EPD_WestHitCnt_th (8)
R2: EPD_EWHitCnt_th (9)

Action:

1st Latch input

2nd Compare each Hit Count to its threshold
Sum the 2 Hit Counts to get the Total (E+W) Hit Count
Select the largest TAC (TAC-E) value from EP001
Select the largest TAC (TAC-W) value from EP002
Define: Good-TAC-E = TAC-E > 0, same for West side

3rd Compare Total (E+W) Hit Count to the EW threshold
Calculate: TAC difference = 4096 + TAC-E – TAC-W
Zero out TAC difference if either Good-TAC-E or Good-TAC-W is false.

4th Latch output

Output to VT201:
(0-12) TAC difference
(13) Total-Hit-Count > th
(14) Hit-Count-E > th
(15) Hit-Count-W > th

Scalers:

(0-12) Unused
(13) Total-Hit-Count > th
(14) Hit-Count-E > th
(15) Hit-Count-W > th

10. Layer 2 Vertex DSM Board: L1-VT201

All information from the BBC, EPD, ZDC and the VPD detectors are brought into the Vertex DSM. The threshold bits are passed on to the TCU. In addition windows are placed around each TAC difference value. The "inside window" bits get passed through to the TCU and the scaler system.

RBT File: l1_vt201_2022_c.rbt

Users: VT201

Inputs: Ch 0 = BB101
Ch 1 = EP101
Ch 2 = ZD101
Ch 3 = Unused
Ch 4 = VP101
Ch 5:7 = Unused

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

From EPD-DSM EP101
(0-12) Inner-tile TAC-Difference
(13) Inner tiles Total (E+W) Hit Count > th0
(14/15) Inner tiles East/West Hit Count > th0

From ZDC DSM ZD101
(0-7) ZDC TAC-Difference
(8-11) West threshold bits
(12-15) East threshold bits

The definition of these threshold bits depends on whether the upstream QT (ZD001) and DSM (ZD101) boards were in proton mode or heavy ion mode. For Run 22 both boards have both East and West sides in the proton mode. In this case the threshold bit definitions are:

Bit #	Definition
8/12	Unused (0)
9/13	Front digital sum > th
10/14	Back digital sum > th
11/15	Total digital sum > th

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

LUT: 1-to-1

Registers:

R0: BBC-TACdiff-Min (13 bits)
R1: BBC-TACdiff-Max (13)
R2: EPD-TACdiff-Min (13)
R3: EPD-TACdiff-Max (13)
R4: ZDC-TACdiff-Min (8)
R5: ZDC-TACdiff-Max (8)
R6: VPD-TACdiff-Min (13)
R7: VPD-TACdiff-Max (13)
R8: VPD-TACdiff2-Min (13)
R9: VPD-TACdiff2-Max (13)

Action

1st Latch inputs

2nd Compare each of the 4 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.
Combine the ZDC Front and Back bits:
ZDC_FB_East = ZDC East Front AND ZDC East Back
ZDC_FB_West = ZDC West Front AND ZDC West Back
Delay all the remaining threshold bits to the 4th step.

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
Make a copy of each of these bits that is gated with the system Run/Stop signal for the scaler output.
Delay the ZDC_FB bits to the 4th step.

4th Latch Outputs

Output to TCU:

Bit	Name	Description
Bit 0	BBC-TAC	BBC TAC difference in window
Bit 1	BBC-E	BBC East ADC sum > threshold
Bit 2	BBC-W	BBC West ADC sum > threshold
Bit 3	EPD-TAC	EPD TAC difference in window
Bit 4	EPD-E	EPD East ADC sum > threshold
Bit 5	EPD-W	EPD West ADC sum > threshold
Bit 6	ZDC-TAC	ZDC TAC difference in window
Bit 7	ZDC-E	ZDC East digital ADC sum > threshold
Bit 8	ZDC-W	ZDC West digital ADC sum > threshold
Bit 9	ZDC-FB-E	ZDC East Front > threshold AND Back > threshold
Bit 10	VPD-TAC2	VPD TAC difference in 2nd window
Bit 11	ZDC-FB-W	ZDC West Front > threshold AND Back > threshold
Bit 12	EPD-EW	EPD inner-tile Total (E+W) Hit Count > threshold
Bit 13	VPD-TAC	VPD TAC difference in window
Bit 14	VPD-E	VPD East ADC sum > threshold
Bit 15	VPD-W	VPD West ADC sum > threshold

Output to Scalers

Bit	Description
Bit 0	BBC small-tile TAC difference in window
Bit 1	BBC East ADC sum > threshold
Bit 2	BBC West ADC sum > threshold
Bit 3	EPD East ADC sum > threshold
Bit 4	EPD West ADC sum > threshold
Bit 5	EPD TAC difference in window
Bit 6	ZDC TAC difference in window
Bit 7	ZDC East Front > threshold AND Back > threshold
Bit 8	ZDC West Front > threshold AND Back > threshold
Bit 9	ZDC East digital ADC sum > threshold
Bit 10	ZDC West digital ADC sum > threshold
Bit 11	VPD TAC difference in window
Bit 12	EPD inner-tile Total (E+W) Hit Count > threshold
Bit 13	VPD TAC difference in 2nd window
Bit 14	VPD East ADC sum > threshold
Bit 15	VPD West ADC sum > threshold