Algorithms for Vertex QT-DSM Tree
RHIC 2018 Heavy Ion Run

Eleanor Judd

February 5, 2018

Change Log:

Date	Description
February 5, 2018	First version for 2018 heavy-ion data taking. BBC-small and BBC-large are using their traditional fastest-TAC algorithms. NOTE that the BBC-large detector itself will not be installed but its QT and DSM boards are still in place. VPD is using the mean-TAC algorithm. ZDC is in its heavy-ion mode. There are 3 new DSM boards in the Vertex DSM tree for EPD: 2 layer-0 boards feeding a single layer-1 board, which feeds the VT201 DSM. There is a new algorithm for VT201, which is based on the "a" version from 2016. The BBC-large and ZDC-UPC bits are being dropped from the output bit list in order to make room for 3 new EPD bits.

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 bits necessary for minimum bias triggers.

1. BBC small-tile 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. Please see the documentation provided by Chris Perkins for a detailed description of this algorithm 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. BBC small-tile 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

3. BBC large-tile 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

4. BBC large-tile 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

5. VPD QT Boards: BBQ_VP001:002

There are two VPD QT boards: one processes data from the East side of the detector and the other from the West side. The algorithm forms a truncated 12-bit ADC sum, a full-range 16-bit sum of slew-corrected TAC values and a “good hit” count. Only channels that satisfy the “good hit” requirements are included in the sums. 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. Please see the documentation for a detailed description of the algorithm at: http://www.star.bnl.gov/public/trg/TSL/Software/qt_v7_6_doc.pdf

6. VPD Layer 1 DSM Board: BBC_VP101

The VP101 DSM board processes data from the VPD detector. The algorithm receives ADC- and TAC sum data as well as hit counts from the QT boards. The ADC sums are compared to thresholds. The TAC sum and hit counts are combined to effectively cut on the difference of the means while avoiding the necessity of implementing a generic large integer division. The standard calculation is:

$$\left| \frac{\sum E}{N(E)} - \frac{\sum W}{N(W)} \right| < threshold$$

This can be re-written as:

$$\left| \left(N(W) \sum E\right) - \left(N(E) \sum W\right) \right| < threshold * N(E)N(W)$$

Performing multiplication in the FPGA is easier than division, so the 2nd equation is used. NOTE: This algorithm takes 4 extra ticks of the 4xRHIC clock, which is used by the FPGA, to complete all these calculations.

RBT File: bbc_vp101_2016_a.rbt

Users: VP101

Inputs: Ch0/3 = Unused
Ch4/5 = QT Board VP003 (East)

Ch6/7 = QT Board VP004 (West)

From each QT board:
bits 0:11 = ADC Sum
bits 12:15 = Hit Count
bits 16:31 = TAC Sum

LUT: 1:1

Registers:

Five registers, all thresholds can be set independently
R0: VPD-EastADCsum_th (12 bits)
R1: VPD-WestADCsum_th (12)
R2: VPD-MeanDiff-window1 (12)
R3: VPD-MeanDiff-window2 (12)
R4: VPD-MeanDiff-window3 (12)

Action:

1st	Latch input
2nd	Compare each ADC sum to its threshold Calculate: $N(W)\sum E$ $N(E)\sum W$ and $N(E)N(W)$
3rd	Delay the ADC sum threshold bits to the 8th step Calculate: $Diff = \lceil N(W)\sum E - N(E)\sum W \rceil$ Calculate: $Th1 = R2N(E)N(W)$ and the same for R3 and R4
4th	Compare the difference to the registers, i.e.: $Window1 = Diff < Th1$ Do the same for Windows 2 and 3.
5th – 7th	Extra time for calculations
8th	Latch output

Output to VT201:

(0)	VPD Mean-TAC difference inside Window-1
(1)	VPD Mean-TAC difference inside Window-2
(2)	VPD Mean-TAC difference inside Window-3
(3:13)	Unused
(14)	ADC-sum-E > th0
(15)	ADC-sum-W > th0

Scalers:

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

7. ZDC QT Board: BBQ_ZD001

The single ZDC QT board receives signals from both the East and West sides of the ZDC. 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 heavy-ion 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

8. 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. For Run 18 both sides are configured to be in heavy ion mode.

RBT File: bbc_zd101_2017_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 TAC
bits 20:25 = West sum/threshold bits (See table below for definition)
bits 26:31 = East sum/threshold bits (See table below for definition)

Bit #	Proton Mode	**Heavy-Ion Mode: Used for Run 18**
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	**Heavy-Ion Mode: Used for Run 18**
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

9. EPD QT Boards with Timing Capability

There are 14 EPD QT boards with TACS. 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 4-bit truncated Hit Count and a 12-bit Max TAC packed onto one output cable. In addition the full range (5-bit) Hit Count is driven on the other QT output cable. That data will not be used during this RHIC Run. 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.

10. EPD Layer 0 DSM Boards: BBC_EP001:2

The EP001:2 DSM boards process data from the East and West sides of the 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 an EPD 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

11. EPD Layer 1 DSM Board: BBC_EP101

The EP101 DSM board processes data from the 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. 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_2018_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:
Two registers, all thresholds can be set independently
R0: EPD-EastHitCnt_th (8 bits)
R1: EPD-WestHitCnt_th (8)

Action:

1st	Latch input
2nd	Compare each Hit Count to its threshold 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	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)	Unused
(14)	Hit-Count-E > th
(15)	Hit-Count-W > th

Scalers:

(0-13)	Unused
(14)	Hit-Count-E > th0
(15)	Hit-Count-W > th0

12. Layer 2 Vertex DSM Board: L1-VT201

All threshold bits of the Vertex tree from the small-tile BBC, the EPD the ZDC and the VPD are brought into the Vertex DSM. They are passed on to the TCU, some as individual bits and some in combinations. In parallel the TAC difference values from the BBC, EPD and ZDC are brought into the Vertex DSM. Windows are placed around each TAC difference, and the "inside window" bits get passed through to the TCU and the scaler system. A minimum bias bit, based on an OR of information from all 4 detectors is created. This bit is used to start a counter whose status can be used to provide preceded protection for subsequent triggers.

RBT File: l1_vt201_2018_a.rbt

Users: VT201

Inputs: Ch 0 = BB101
Ch 1 = Unused
Ch 2 = ZD101
Ch 3 = Unused
Ch 4 = VP101
Ch 5 = EP101
Ch 6:7 = 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

(0-7)	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 18 both boards have both East and West sides in the heavy ion mode. In this case the threshold bit definitions are:

Bit #	Definition
8/12	Good Analog sum > th0
9/13	Good Analog sum > th1
10/14	Good Analog sum > th2
11/15	Good Analog sum > th3

From VPD-DSM VP101

(0)	VPD Mean-TAC Difference inside Window-1
(1)	VPD Mean-TAC Difference inside Window-2
(2)	VPD Mean-TAC Difference inside Window-3
(3/13)	Unused
(14/15)	VPD ADC East/West> th0

From EPD-DSM EP101

(0-12)	EPD TAC-Difference
(13)	Unused
(14/15)	EPD East/West Hit Count > th0

LUT: 1-to-1

Registers:

R0: BBCsmall-TACdiff-Min (13 bits)
R1: BBCsmall-TACdiff-Max (13)
R2: EPD-TACdiff-Min (13)
R3: EPD-TACdiff-Max (13)
R4: ZDC-TACdiff-Min (8)
R5: ZDC-TACdiff-Max (8)
R6: Minimum-Bias-Select (4)
R7: Min_Bias_Protection_Time (9)

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 Make ZDC-COINC = E>th0 AND W>th0
Make ZDC-EW = E>th3 OR W>th3

Combine the results of the TAC difference comparisons to determine if each TAC difference is inside its specified window, e.g.:

ZDC-Tdiff = R4 < ZDC TAC difference < R5

Combine the results of the TAC difference comparisons and the ADC threshold bits to make the minimum bias bit, using R6 to turn each component on/off, i.e.:

```
MB = (R6(0) and BBC-S-Tdiff and BBC-S-E>th0 and BBC-S-W>th0) or
     (R6(1) and EPD-Tdiff and EPD-E>th0 and EPD-W>th0) or
     (R6(2) and ZDC-Tdiff) or
     (R6(3) and VPD-Win1)
```

The preceded logic is only enabled if R7 is set to a non-zero value. In this case, whenever the minimum bias bit is set a counter is initialized to R7-1. The counter then counts down to zero at a rate of one count per tick of the RHIC clock. If another minimum bias interaction occurs while the counter is counting, then the counter is re-initialized to R7-1 and counting continues. The preceded bit is true whenever the current counter value is non-zero, and false otherwise.

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	EPD-TAC	EPD TAC difference in window
Bit 4	EPD-E	EPD East Hit Count > threshold
Bit 5	EPD-W	EPD West Hit Count > threshold
Bit 6	ZDC-TAC	ZDC TAC difference in window
Bit 7	ZDC-E	ZDC East Good Analog sum > th0
Bit 8	ZDC-W	ZDC West Good Analog sum > th0
Bit 9	ZDC-EW	ZDC East Good Analog sum > th3 OR ZDC West Good Analog sum > th3
Bit 10	Minimum-Bias	At least one selected TAC difference in window
Bit 11	Preceded	Counter started by Minimum Bias bit still counting.
Bit 12	VPD-TAC2	VPD Mean-TAC difference in window-2
Bit 13	VPD-TAC	VPD Mean-TAC difference in window-1
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
Bits 1:4	Unused
Bit 5	EPD TAC difference in window
Bit 6	ZDC TAC difference in window
Bit 7	ZDC East Good Analog sum > th0 AND ZDC West Good Analog sum > th0
Bits 8:10	Unused
Bit 11	VPD Mean-TAC difference in window-1
Bit 12	VPD Mean-TAC difference in window-2
Bit 13	VPD Mean-TAC difference in window-3
Bits 14:15	Unused