

# Algorithms for Vertex QT-DSM Tree RHIC 2014 $^3\text{He}+\text{Au}$ Run

Eleanor Judd

June 17, 2014

## Change Log:

Date	Description
June 9, 2014	First version for the 2014 $^3\text{He}+\text{Au}$ data taking. The ZD101 algorithm has been changed to include a “good TAC” cut in the ADC threshold bits. This is necessary to trigger on just ZDCE.
June 17, 2014	The ZD101 algorithm has been changed again. The OR of the two threshold-3 bits (used to define central triggers) has been replaced with just the East bit.

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 (for example) minimum bias or ultra-peripheral triggers.

## 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. The algorithm has been changed in 2014 to add a slewing correction to the original logic. Please see the documentation provided by Chris Perkins for a detailed description of the new algorithm at

[http://www.star.bnl.gov/public/trg/TSL/Software/qt\\_v6\\_d\\_doc.pdf](http://www.star.bnl.gov/public/trg/TSL/Software/qt_v6_d_doc.pdf)

The algorithm still forms a 16-bit ADC Sum and 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 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 old algorithm is documented at

[http://www.star.bnl.gov/public/trg/TSL/Software/qt\\_v5\\_6\\_doc.pdf](http://www.star.bnl.gov/public/trg/TSL/Software/qt_v5_6_doc.pdf)

The slow correction logic is the same logic that was added to the MTD QT boards in 2013, and is documented at [http://www.star.bnl.gov/public/trg/TSL/Software/qt\\_v6\\_c\\_doc.pdf](http://www.star.bnl.gov/public/trg/TSL/Software/qt_v6_c_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:

- 1<sup>st</sup> Latch input
- 2<sup>nd</sup> Compare each ADC-sum to its threshold  
Calculate: TAC difference =  $4096 + \text{TAC-E} - \text{TAC-W}$   
Define: Good-TAC-E =  $\text{TAC-E} > 0$ , 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
- 3<sup>rd</sup> 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.
- 4<sup>th</sup> 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](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:

- 1<sup>st</sup> Latch input
- 2<sup>nd</sup> Delay hit bits to 4<sup>th</sup> 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)
- 3<sup>rd</sup> Zero out TAC difference if either Good-TAC-E or Good-TAC-W is false, otherwise just delay TAC difference to the 4<sup>th</sup> 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.

4<sup>th</sup> 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 is used by 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 2 QT boards. 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**

For the 2014 full-energy heavy-ion data taking we are reverting to the QT algorithm created for the 2012 heavy-ion running. In the past we have used ZDC electronics that produced an East+West attenuated sum signal which was used for defining central triggers. That signal is no longer available. Instead, this algorithm adds an extra threshold to each of the East and West sides independently. Those thresholds can be used to detect large energy deposition in at least one side of the ZDC, and that condition will be used to define the central trigger. The algorithm was written by Chris Perkins and is documented at [http://www.star.bnl.gov/public/trg/TSL/Software/qt\\_v6\\_a\\_doc.pdf](http://www.star.bnl.gov/public/trg/TSL/Software/qt_v6_a_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 ZD001 QT board. 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 ZDC. A user-specified set of bits is then chosen to be passed on to VT201. In addition, the algorithm also receives the results of comparing sums from the East and West sides of the detector to thresholds. Each of those threshold bits is zeroed out if the relevant TAC value is also zero. Two thresholds, one from each side, are protected by a “preceded” signal in order to deal with after-pulsing in the ZDC. This allows the user to zero out those two thresholds for a certain number of RHIC clock ticks after a ZDC coincidence is detected.

RBT File: bbc\_zd101\_2014\_c.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

bit 20 = West attenuated sum > threshold-4 (Unused in this algorithm)

bit 21 = West attenuated sum > threshold-5 (Unused in this algorithm)

bit 22 = East attenuated sum > threshold-4 (Unused in this algorithm)

bit 23 = East attenuated sum > threshold-5 (Unused in this algorithm)

bit 24 = West sum > threshold-0

bit 25 = West sum > threshold-1

bit 26 = West sum > threshold-2

bit 27 = East sum > threshold-0

bit 28 = East sum > threshold-1

bit 29 = East sum > threshold-2

bit 30 = West sum > threshold-3 (Unused in this algorithm)

bit 31 = East sum > threshold-3

LUT: 1:1

Registers:

R0: ZDC-TACdiff-select (2 bits)

0 => select bits 0:8

1 => select bits 1:9

2 => select bits 2:10

R1: ZDC-EastTAC-select (3)

0 => select bits 0:4

1 => select bits 1:5

...

5 => select bits 5:9

R2: ZDC-WestTAC-select (3)

Same value definitions as for R1

R3: ZDC-deadtime (4 bits)

Action:

- 1<sup>st</sup> Latch input
- 2<sup>nd</sup> Delay threshold-0, -1, -2 and -3 bits to the 3<sup>rd</sup> step.  
Zero out a copy of the threshold-0 bits if the Preceded bit is set, i.e.  
Protected-West-th0 = West-th0 and NOT Preceded  
Protected-East-th0 = East-th0 and NOT Preceded  
Calculate: TAC difference = 1024 + TAC-E – TAC-W  
Define: Good-TAC-E = TAC-E > 0, same for West side  
For the scaler output, make all possible bit selections from TAC-E and TAC-W, including overflow logic. For example:  
TAC-E-overflow-0 = TAC-E(5), (6), (7), (8) or (9)  
If (TAC-E-overflow-0 = 1) then TAC-E-scaler-0 = 31  
Else TAC-E-scaler-0 = TAC-E(0:4)  
Same logic for all possible bit selections from TAC-E (see description of register R1) and TAC-W
- 3<sup>rd</sup> Combine the protected and unprotected threshold bits with the appropriate Good-TAC bit, e.g.:  
Sum-W-th1 = West-th1 and Good-TAC-W  
Sum-E-th3 = East-th3 and Good-TAC-E  
Etc.,,  
For the VT201 output use R0 to select the TAC difference bits, including overflow logic and the “good” TAC cut, i.e.:  
Diff-overflow-0 = TAC-diff(9) or (10)  
Diff-overflow-1 = TAC-diff(10)  
If (Good-TAC-E = 0 or Good-TAC-W = 0) then output = 0  
Else if (R0 = 0)  
If (Diff-overflow-0 = 1) then output = 511  
Else output = TAC-diff(0:8)  
Else if (R0 = 1)  
If (Diff-overflow-1 = 1) then output = 511  
Else output = TAC-diff(1:9)  
Else  
Output = TAC-diff(2:10)  
For the scaler output use R1 to select the TAC-E scaler bits:  
If (R1 = 0) then chose TAC-E-scaler-0  
Else if (R1 = 1) then chose TAC-E-scaler-1  
Etc...  
Do the same for the West side using R2 to control the selection.  
For the Preceded logic check for a ZDC coincidence:  
Coincidence = Protected-West-th0 and Protected-East-th0 and  
Good-TAC-W and Good-TAC-E  
If there is a coincidence then initialize a counter to R3-1. Allow it to count down to zero at a rate of one count per tick of the RHIC clock. Set the “Preceded” bit to one while the counter is counting.  
NOTE: If R3 = 0 then the Preceded logic is disabled.
- 4<sup>th</sup> Latch output

Output to VT201:

- (0-8) TAC difference
- (9) Protected-sum-W > th0
- (10) Sum-W > th1
- (11) Sum-W > th2
- (12) Protected-sum-E > th0
- (13) Sum-E > th1
- (14) Sum-E > th2
- (15) Sum-E > th3

Scalers:

- (0-4) selected bits of TAC-E
- (5-9) selected bits of TAC-W
- (10) Protected-sum-W > th0
- (11) Sum-W > th0
- (12) Protected-sum-E > th0
- (13) Sum-E > th0
- (14) ZDC coincidence
- (15) Sum-E > th3

## 8. Layer 2 Vertex DSM Board: L1-VT201

All threshold bits of the Vertex tree from the large and small-tile BBC, 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 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. Some of the TAC difference bits from the ZDC, the BBC small-tiles and the VPD are also in the scaler output. 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: 11\_vt201\_2014\_a.rbt

Users: VT201

Inputs: Ch 0 = BB101  
Ch 1 = 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 Large tile BBC-DSM BB102

- (0-12) Large tile TAC-Difference
- (13) Unused
- (14/15) East/West hit

From ZDC DSM ZD101

- (0-8) TAC difference
- (9) Protected-sum-W > th0
- (10) Sum-W > th1
- (11) Sum-W > th2
- (12) Protected-sum-E > th0
- (13) Sum-E > th1
- (14) Sum-E > th2
- (15) Sum-E > th3

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: BBClarge-TACdiff-Min (13)
- R3: BBClarge-TACdiff-Max (13)
- R4: ZDC-TACdiff-Min (9)
- R5: ZDC-TACdiff-Max (9)
- R6: VPD-TACdiff-Min (13)
- R7: VPD-TACdiff-Max (13)
- R8: Minimum-Bias-Select (4)
- R9: Min\_Bias\_Protection\_Time (9)
- R10: VPD-TACdiff2-Min (13)
- R11: VPD-TACdiff2-Max (13)

Action

- 1<sup>st</sup> Latch inputs
- 2<sup>nd</sup> Delay all the threshold bits that need to go to the TCU to the 4<sup>th</sup> step.  
Delay a 2<sup>nd</sup> copy of the ZDC-th0 bits and the threshold bits from BBC-small and BBC-large to the 3<sup>rd</sup> step.  
Delay a copy of the ZDC, BBC-small and VPD TAC difference to the 4<sup>th</sup> step.  
Combine the ZDC (un-protected) th1 and th2 bits to make windows on the East and West sides separately, i.e.:  
$$\text{ZDC-E-Window} = \text{Sum-E} > \text{th1} \text{ and not } \text{Sum-E} > \text{th2}$$
$$\text{ZDC-W-Window} = \text{Sum-W} > \text{th1} \text{ and not } \text{Sum-W} > \text{th2}$$
  
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. The VPD TAC difference is compared to two separate sets of min/max values defining two separate windows.
- 3<sup>rd</sup> Make the following combinations of ZDC and VPD bits:

ZDC-COINC = Protected-sum-W>th0 and Protected-sum-E>th0  
ZDC-UPC = ZDC-E-Window and ZDC-W-Window

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

$$\text{VPD-Tdiff} = R6 < \text{VPD TAC difference} < R7$$

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 BBC-L-Tdiff and BBC-L-E>th0 and BBC-L-W>th0) or  
(R8(2) and ZDC-Tdiff) or  
(R8(3) and VPD-Tdiff)

The preceded logic is only enabled if R9 is set to a non-zero value. In this case, whenever the minimum bias bit is set a counter is initialized to R9-1. 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 R9-1 and counting continues. The preceded bit is true whenever the current counter value is non-zero, and false otherwise.

4<sup>th</sup> 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	VPD-TAC2	VPD TAC difference in 2 <sup>nd</sup> window
Bit 4	BBC-L-E	BBC large-tile East hit
Bit 5	BBC-L-W	BBC large-tile West hit
Bit 6	ZDC-TAC	ZDC TAC difference in window
Bit 7	ZDC-E	ZDC Protected-sum-East > th0
Bit 8	ZDC-W	ZDC Protected-sum-West > th0
Bit 9	ZDC-UPC	ZDC East in window (th1 and th2) AND ZDC West in window (th1 and th2)
Bit 10	ZDC-E3	ZDC Sum-East > th3
Bit 11	Minimum-Bias	At least one selected TAC difference in window
Bit 12	Preceded	Counter started by Minimum Bias bit still counting.
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
Bits 1:4	4 MSB of BBC small-tile TAC difference
Bit 5	BBC large-tile TAC difference in window
Bit 6	ZDC TAC difference in window
Bit 7	ZDC Protected-sum-East > th0 AND ZDC Protected-Sum-West > th0
Bits 8:10	3 MSB of ZDC TAC difference
Bit 11	VPD TAC difference in window
Bits 12:15	4 MSB of VPD TAC difference