Algorithms for Vertex QT-DSM Tree
RHIC 2021 BES-II Run

Eleanor Judd

March 31, 2021

Change Log:

Date	Description
February 20, 2019	First version for 2019. The BBC small-tile and EPD inner-tile systems are, as usual, using the fastest TAC logic. In order to maximize its efficiency, the VPD is also using the fastest-TAC logic, which uses all 16 channels on each side instead of just the 12 channels used by the mean-TAC logic. The ZDC is configured in its heavy-ion mode. The BBC large-tile system has been removed. The EPD outer-tile system has been added to allow STAR to trigger on asymmetries in the EPD hit pattern. The VT201 algorithm has been optimized for this year's requirements. The minimum bias and preceded bits have been deleted. They have been fully superseded by logic in the TCU and will never be used. The 2nd VPD TAC window and the high-threshold ZDC bits have been deleted because they are unnecessary for low-energy collisions. Those 4 bits have been replaced with 4 new bits from the EPD outer-tile system.
February 28, 2019	The EP102 DSM algorithm has been modified. The East-West multiplicity difference is zeroed out if either of the two multiplicity values is zero. This is the same logic that is used for all of the TAC difference calculations. In addition, the EPD outer-tile QT boards have reverted back to algorithm v64 (used in Run 18) because of a problem with algorithm v7b. As a result their inputs to EP005 and EP006 are nonsensical, and therefore the outer-tile data has been switched off at the input to EP102.
March 31, 2021	The EP101 algorithm has been modified to create an East-West bit for use in central triggers. As a result the VT201 algorithm has been modified to pass this bit through to the TCU. In order to make room the 4 EPD outer-tile bits have been dropped from VT201. One is replaced with the new EPD East-West bit. The others have been used to reinstate the ZDC East-West bit, the minimum bias bit and the preceded 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 data needed for effectively triggering on low-energy heavy-ion 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, 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 Layer 1 DSM Board: 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: BBC_small_EastADCsum-th (16 bits)
R1: BBC_small_WestADCsum-th (16)
R2: BBC_small_EastTAC-select (3)
0 => select bits 0:6
…
5 => select bits 5:11
R3: BBC_small_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 also algorithm 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 VP001 (East)
Ch6/7 = QT Board VP002 (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 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

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 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
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	**Heavy-Ion Mode: Used for Run 19**
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 19**
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. The full range (0-16, 5 bits) Hit Count is driven on the other QT 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 0 DSM Boards: BBC_EP003:4

The EP003:4 DSM boards process data from the East and West sides of the inner EPD detector respectively. The algorithm receives a full range Hit Count (0-16, 5 bits) from each of 7 QT boards. It will sum the hit counts in 2 groups to produce the total multiplicity in Ring Group 1 (channels 3,4, 5 and 7) and Ring Group 2 (channels 0, 1 and 2). NOTE that channel 6 is unused. Register 0 can be used to turn each input channel on or off in the logic.

RBT File: bbc_ep003_2019_a.rbt

Users: EP003:4

Inputs: Ch0/7 = 1st output cable from each EPD inner-tile QT board

NOTE: Channel 6 is unused

From each QT board:
bits 0:4 = Hit Count
bits 5:15 = Unused

LUT: 1:1

Registers:

R0: EPD_EP003_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 apply the mask specified in R0:
If R0(X) = 0 then hit count = 0
Else hit count = latched input value

3rd Sum the hit count values
Ring Group 2 = hit_count(0) + hit_count(1) + hit_count(2)
Ring Group 1 = hit_count(3) + hit_count(4) + hit_count(5) + hit_count(7)

4th Latch output

Output to EP102:

(0-6) Ring Group 1 Multiplicity
(7) Unused
(8-14) Ring Group 2 Multiplicity
(15) Unused

10. 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

11. EPD Outer-tile QT Boards: EQ1, EQ2 and EQ3 crates

There are 18 EPD QT boards for the outer tiles. 9 process data from the East side of the detector and the other 9 cover the West side. These boards are currently using an old FMS algorithm (v64) because of a problem with the new EPD-specific algorithm. 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_4_doc.pdf
The output consists a 17-bit total sum of all 32 12-bit ADC values. Only the 12 LSB are actually passed to the layer-0 DSM boards.

12. EPD Outer-tile Layer 0 DSM Boards: BBC_EP005:6

The EP005:6 DSM boards process data from the East and West sides of the outer EPD detector respectively. Each DSM receives its data through a TDSMI, so there are 10 12-bit input channels. The algorithm receives 2 hit counts (range 0-32, 6 bits) from each of 9 QT boards. Only one of each pair of hit counts is needed for the current algorithm, however both will be included in the logic with registers to turn off the unwanted values. The algorithm will sum the hit counts in 3 groups to produce the total multiplicity in Ring Group 3 (channels 6, 7 and 8), Ring Group 4 (channels 3, 4 and 5) and Ring Group 5 (channels 0, 1 and 2). NOTE that channel 9 is unused.

NOTE: The outer-tile QT boards are currently using an old algorithm which produces different output, so the data flowing through these boards is not useful. Their output is therefore being turned off at the input to EP102, the layer-1 DSM board.

RBT File: bbc_ep005_2019_a.rbt

Users: EP005:6

Inputs: Ch0/8 = 1st output cable from EPD outer-tile QT board
NOTE: Channel 9 is unused

From each QT board:
bits 0:5 = Hit Count-A
bits 6:11 = Hit Count-B

LUT: 1:1

Registers:

R0: EPD_EP005_Ch0_4_select (10 bits)
- Bit 0: Turn Hit-Count-A from Ch-0 on (1) or off (0)
- Bit 1: Turn Hit-Count-B from Ch-0 on (1) or off (0)
- Etc…

R0: EPD_EP005_Ch5_8_select (8 bits)
- Bit 0: Turn Hit-Count-A from Ch-5 on (1) or off (0)
- Bit 1: Turn Hit-Count-A from Ch-5 on (1) or off (0)
- Etc…

Action:

1st Latch input

2nd For each hit count X apply the mask specified in R0 or R1:
- If R(X) = 0 then hit count = 0
- Else hit count = latched input value

Add together the 2 hit counts from each channel to get a 7-bit channel count

3rd Sum the channel count values to make the Ring Group multiplicities
- Ring Group 5 = ch_count(0) + ch_count(1) + ch_count(2)
- Ring Group 4 = ch_count(3) + ch_count(4) + ch_count(5)

Ring Group 3 = ch_count(6) + ch_count(7) + ch_count(8)

Each Ring Group multiplicity is a 9-bit number, range 0 to 511. However, in reality the maximum physical Ring Group multiplicity is 96, which is a 7-bit number, so the 8th and 9th bits should always be zero (0). In order to simplify the data transfer to EP102 the 8th bit is dropped

4th Latch output

Output to EP102:

(0-7) Ring Group 5 Multiplicity
(8-15) Ring Group 4 Multiplicity
(16-23) Ring Group 3 Multiplicity
(24-31) Unused

13. EPD Outer-tile Layer 1 DSM Board: BBC_EP102

The EP102 DSM board processes data from the full EPD detector. The algorithm receives Ring Group hit counts for the East and West sides from 4 EPD Layer 0 DSM boards (EP003:6). The Ring Group Hit Counts are summed to get total counts for the East and West sides separately. The 2 sums are then added together to get a total multiplicity for the EPD detector, which is then truncated to 8 bits. In parallel the East and West totals are truncated, and then the difference between the 2 is calculated, resulting in another 8-bit number. The difference is set to zero if either of the two multiplicity values is zero. Registers allow the user to turn each Ring Group on or off in the logic.

RBT File: bbc_ep102_2019_b.rbt

Users: EP102

Inputs: Ch0 = DSM Board EP003 (East, inner-tiles, Ring Groups 1:2)
Ch1/2 = DSM Board EP005 (East, outer-tiles, Ring Groups 3:5)
Ch3 = DSM Board EP004 (West, inner-tiles, Ring Groups 1:2)
Ch 4/5 = DSM Board EP006 (West, outer-tiles, Ring Groups 3:5)

From EP003/4 (inner tiles)
(0-6) Ring Group 1 Multiplicity
(7) Unused
(8-14) Ring Group 2 Multiplicity
(15) Unused

From EP005/6 (outer tiles)
(0-7) Ring Group 5 Multiplicity
(8-15) Ring Group 4 Multiplicity
(16-23) Ring Group 3 Multiplicity
(24-31) Unused

LUT: 1:1

Registers:

R0: EPD-East-RG-onoff (5 bits)
R1: EPD-West-RG-onoff

Action:

1st Latch input

2nd For each Ring Group X apply the mask specified in R0 or R1:
If R(X) = 0 then hit count = 0
Else hit count = latched input value
Add together the 5 Ring Group hit counts from each side to get a 10-bit total multiplicity for the East and West sides separately.
Define: Good-E = Multiplicity-E > 0, same for West side

3rd Add together the East and West multiplicities to get an 11-bit EPD total multiplicity, then truncate to 8-bits by discarding the 3 LSB.
In parallel, truncate each of the East and West multiplicities to 7 bits by discarding the 3 MSB.
Calculate: East-West = 128 + Truncated-Mult-E – Truncated-Mult-W
Zero out multiplicity difference if either Good-E or Good-W is false.

4th Latch output

Output to VT201:

(0-7) Truncated EPD total multiplicity
(8-15) Difference between truncated East and West multiplicities

14. 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 both 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_2021_a.rbt

Users: VT201

Inputs: Ch 0 = BB101
Ch 1 = EP101 – EPD inner tiles, timing information
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) EPD TAC-Difference

(13) EPD inner tiles Total (E+W) Hit Count > th0
(14/15) EPD 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 21 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
(0-12) VPD TAC-Difference
(13) Unused
(14/15) VPD ADC East/West> th0

LUT: 1-to-1

Registers:
R0: BBC_small-TACdiff-Min (13 bits)
R1: BBC_small-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_vtx_TACdiff_Min (13)
R7: VPD_vtx_TACdiff_Max (13)
R8: XPERT_Minimum_Bias_Select (4)
R9: XPERT_Min_Bias_Protect_Time (9)

Action

1st Latch inputs

2nd Compare each of the TAC differences to its minimum and maximum values, as specified in the relevant registers. The logic looks for each difference value to be greater than the minimum and less than the maximum.
Delay all the incoming threshold bits to the 4th step. In parallel delay copies of the BBC, EPD and ZDC threshold bits to the 3rd step.

3rd Combine the results of the register comparisons to determine if each TAC is inside its specified window, e.g.:

ZDC-TAC-diff-in-window = R4 < ZDC TAC difference < R5

Combine those TAC-diff-in-window bits with 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 EPD-Tdiff and EPD-E>th0 and EPD-W>th0) or
(R8(2) and ZDC-Tdiff) or
(R8(3) and VPD-Tdiff)

Make ZDC-EW = E>th3 OR W>th3 for use in central triggers

Make ZDC-COINC = E>th0 AND W>th0 for the scaler system.

The preceded logic is only enabled if R9 is non-zero. 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.

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 inner-tile TAC difference in window
Bit 4	EPD-E	EPD East inner-tile Hit Count > threshold
Bit 5	EPD-W	EPD West inner-tile 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 countingi
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
Bits 1:4	Unused
Bits 5	EPD TAC difference in window
Bit 6	ZDC TAC difference in window
Bits 7	ZDC th0 Coincidence
Bits 8:10	Unused
Bit 11	VPD TAC difference in window
Bits 12:15	Unused