Procedure for timing-in a new branch of the STAR Level-0 Trigger System

H.J. Crawford , E.G. Judd, J.M. Engelage, C.W. Perkins, J. M. Nelson, S. Valentino

September 15, 2016

Timing-in a new branch of the STAR Level-0 Trigger System takes several steps:

- Adjusting the ADC gate timing on the QT boards so that PMT signals fall inside the gate
- Adjusting the TAC-stop timing (if TACS are being used) so that all PMT signals produce a valid TAC value.
- Setting the data start address on the QT boards so that data is stored in the correct place, as defined by the TCU, for readout.
- Adjusting the trickle-down timing between QT daughter cards so data are combined correctly
- Adjusting the QT output delay setting so that QT data is correctly transferred from QT to DSM boards.
- Adjusting the clock phase setting and the output delay (FIFO blanks) for all DSM boards so they correctly transfer data from one layer to the next, and to the TCU.
- Setting the address increment on the DSM boards so that data is stored in the correct place, as defined by the TCU, for readout.

Any new branch must fit within the total transit time defined by the existing branches otherwise we have to change the timing of the whole system. The endpoint, the TCU, is therefore considered to be fixed for all branches. In addition, the timing of the front end gates must match the arrival time of the PMT signals so that timing is highly constrained for each detector. There is however a significant amount of flexibility built in to the inter-layer timing, and that is where adjustments are made for individual branches.

In this document the new FCAL detector will be used as the example. It will use QT boards (FC001:26) and 2 layers of DSM boards (FC101:4 and FC201), just like the existing BBC and VPD subsystems. FCAL will not use TAC information, just ADC values, so the TAC timing section will be based on the experience gained from the Run 16 QT8-TAC tests.

The procedure for timing in the FCAL detector will be described based on the assumption that all the necessary hardware and software are available before collisions start. The basic philosophy is to do as much work as possible before collisions start in order to minimize the time needed to set up the system once collisions are available. The first steps will involve using an initial guess at all the DSM and QT timing parameters based on the values used by similar existing systems. The FCAL system will then be timed in using pedestal data. The phase of the RCC2 clock that drives each DSM layer, and the QT ADC gate settings, will be scanned through a large enough range to determine how much slack is available for future adjustments. The remaining steps will be accomplished after collisions start. This second part of the procedure will involve adjusting the QT clock so that the ADC gate covers all the PMT signals, determining the necessary QT data start address so the data can be readout and then adjusting the QT output delay and DSM settings so that the data is correctly transferred through the tree to the TCU. The final step is to time in the TAC-stop signal.

Step 0 - Preparation

The following pieces of software should be ready before the timing process begins.

- The QT FPGA configuration files (MCS)
- All DSM algorithm FPGA configuration files (RBT)
- A bit checker that will simulate the effect of the algorithm on the data read from each layer, predict the data that should be read from the next layer and compare that prediction to the actual data read.
- A histogram package that can operate on raw trigger data blocks, including pre- and post-crossing data.

Before collisions start an oscilloscope with Ethernet capability should be set up next to one of the QT crates. It should be connected to a PMT signal cable and the ADC gate monitor. If the new QT daughter cards with built-in TAC circuitry are being used then the oscilloscope also has access to the discriminator signal and the flip-lop output from Channel 4 of each daughter card. If the older external TAC boards are still being used then the available TAC signals are the discriminator signal from Channel 16 and the flip-flop reset signal from that external board.

The initial guesses for the values of all the timing parameters in the new branch should be based on the values used by existing boards at the same layer in the decision tree, and ideally in the same VME crate, e.g.

- FC201 is a layer-2 DSM board that will be located in the L1 crate and will feed the TCU. It will receive its data from layer-1 DSM boards in the BBC crate. FC201 is therefore very similar to VT201 so it should start with the same RCC2 phase (0x98), address increment (4) and FIFO blanks (0) as are used by VT201.
- FC101:4 are layer-1 DSM boards that will be located in the BBC crate and will feed FC201. There are 3 different DSM configurations currently in that crate:
    - BB101 receives its data early and takes one tick of the RHIC clock. Its timing settings are RCC2 phase = 0x6a, address increment = 8 and FIFO blanks = 2
    - VP101 receives its data at the same time as BB101, so it uses the same RCC2 phase (0x6a) and the same address increment (8). However the VP101 algorithm takes an extra tick of the RHIC clock so VP101 uses 1 FIFO blank instead of 2.
    - ZD101 receives its data one full RHIC clock tick later than BB101 and VP101. It therefore uses the same clock phase (0x6a), but an address increment of 7 instead of 8. Since the data arrives one tick late with respect to BB101 the results are produced one tick late, so again ZD101 uses 1 FIFO blank instead of 2.

Any one of these 3 configurations could work for FCAL. One should be selected as the initial guess for the FC101:4 settings, based on the expected arrival time of the data and the known transit time of the FC101 algorithm.

- FQ001:26 are QT boards that will be located in crates in the STAR hall, not on the platform.
    - These QT boards will be in new crates with dedicated RCC2 clock drivers. The clock phase that drives the QT boards can be set to zero. NOTE: In the case where new QT boards are being added in to an existing crate the new QT boards will need to make do with whatever RCC2 clock setting is used by that existing system. If TACS are being used then the separate RCC2 clock phase that drives the TAC-stop signal can also be set to zero.
    - The QT output timing settings (QT8D Algorithm_Latch and motherboard Output_Latch_Delay) will need to be timed in. The motherboard Output_Latch_Delay should be set to zero. On daughter card QT8D Algorithm_Latch should be set to 1. NOTE: on daughter cards A, B and C Algorithm_Latch must be set to 20.
    - The ADC gate settings will also need to be timed in, so they should be set to the minimal values needed to get an appropriately wide gate, e,g. Gate_Start_Delay = 1 and Gate_End_Delay = 16, as used by the BBC.
    - The most similar existing system is the FMS so the data address settings used in the 4 FMS QT crates would probably be a good starting point, i.e. Data_Start_Address = 10

Step 1 – DSMS-to-TCU

- Make a Tier1 file with all the correct DSM algorithms and all the initial starting guesses listed above. For FC101:4 set DSM_INMEM_PLAY_REC = 1 and set the 4 DSM_INMEM files to DSM_8bit_Ramps.dat. This will cause those DSM boards to ignore any input from the QT boards and drive the ramp data through the DSM tree. Make sure any DSM algorithm registers are set to appropriate values so that the output bits change.
- Take data using a Random trigger. Use the bit checker to make sure the same ramp data is read from all of FC101:4 (they should be synchronized) and that the algorithm results are passed correctly to FC201 and the TCU.
- Repeat this process several times, changing just the RCC2 clock phase of the FC101:4 DSMS at each step. Take data at enough steps to scan through the RCC2 clock phase and find the range that works. This information may be necessary if the real data arrives much earlier or later than expected and it is necessary to adjust the timing inside the DSM tree. NOTE: the RCC2 phase settings are not available in the Run Control GUI so a new Tier1 file will be needed for each step.

- Repeat the process again, but this time leave the FC101:4 RCC2 clock phase at its starting value and scan through values of the FC201 clock phase.

Step 2 – QT8D to DSM

- Load the correct algorithm into the FQ001:26 QT boards.
- Make a Tier1 file with the QT8-D Algorithm Latch, the Data Start Address, ADC Gate Start and End Delays, the motherboard Output Latch Delay and the Channel Mask registers available in the GUI. Set the QT LUT configuration to be 1-to-1 maps so the QT algorithm will be operating on pedestal data. Make sure any algorithm thresholds are set to appropriate values so that the output bits change.
- Turn off all channels in QT8A, B and C, but turn on all channels in QT8D. Take pedestal data using a zero bias trigger. Take short runs where you scan through the QT8D Algorithm Latch using 7 values from 1 to 7. For each value scan through the Output

those in QT8D, and check that their combined data still flows correctly through the system.

- Repeat with QT8B added in.
- Repeat with QT8A added in.

At this point the system should be fully timed in using all the correct algorithms and the initial guess as to the QT timing. The remaining steps must wait until collisions start.

Step 4 – ADC Gate Timing

- The timing of the ADC gate with respect to the PMT pulses depends on the phase of the RCC2 clock that drives the QT boards, and the delay settings for the gate. The RCC2 clock phase is not available from the Run Control GUI however the gate delay settings are available. The gate timing is therefore accomplished by adjusting the gate delay settings, not the RCC2 clock phase.
- Configure the system using the initial guesses for the ADC Gate Start/End Delays. Look at the gate monitor and the incoming PMT pulses on the oscilloscope. Increase the gate delay settings as necessary to move the gate so it covers all of the pulses. This is the coarse adjustment.
- Take data using a trigger based on other detectors that should result in particles hitting FCAL. The trigger system should be set up to read out at least 1 pre- and post-crossing. Take a series of runs and scan through the QT Data_Start_Address values. Select the Data_Start_Address value that puts the ADC values into the triggered crossing.
- Next do the fine adjustment of the gate. Take a series of runs, scan through the Gate Start/End Delays in 1ns steps and make a histogram of the ADC data for each channel separately. If the gate is too late, and the leading edges of some PMT pulses are being missed, then the ADC distribution should show an enhancement at low values and suppression at high values. As the gate is shifted to earlier times and starts to pick up those missed signals the shape of the distribution will shift to higher values. Once the gate is early enough that all PMT signals are covered then shifting the gate to even earlier times should have no effect on the ADC distribution. It should be possible to determine which gate settings are too late and which gate settings are OK. For each QT board choose the gate settings that result in all pulses on all channels being comfortably inside the gate.
- Now it is necessary to check the flow of data through the QT and DSM tree again. Use the information gained during the last part of Step 2 to set the QT8D Algorithm_Latch, the motherboard Output_Latch_Delay, the DSM (FC101:4) address increment and FIFO Blanks to the most likely values so that the data flows correctly through the whole system to the TCU. Take data using a trigger based on other detectors that should result in particles hitting FCAL. Check the flow of bits all the way through to the TCU. If necessary adjust the timing at any stage using the method described in Step 2. **NOTE**: If

there was not enough time to do Step 2 in full before collisions started then that process should just be done now, using the optimal settings for the Gate_Start_Delay and Gate_End_Delay.

Step 5 – TAC-stop Timing

- The TAC-stop signal is a 50% duty cycle clock that is used to reset a D-flip-flop. The TAC circuits are therefore live for exactly half of every RHIC clock tick. The ideal configuration is one where the TAC circuit becomes live just before the arrival time of the earliest pulse inside the ADC gate.
- The RCC2 is the source of the TAC-stop clock, and its clock phases are not available from the Run Control GUI. The phases can however be adjusted manually at any time. The TAC-stop timing is therefore accomplished by adjusting the phase of the relevant RCC2 channel manually at the beginning of each run.
- Configure the system using the optimal values for the ADC Gate Start/End Delays. When collisions are present look at the gate monitor, the TAC-stop monitor and the incoming PMT pulses on the oscilloscope. Adjust the TAC-stop phase setting as necessary to move the TAC-stop so it becomes live as soon as possible after the ADC gate has opened. This is the coarse adjustment.
- Next do the fine adjustment of the TAC-stop. Take a series of runs and scan through the TAC-stop phase setting in 1ns steps. For each channel separately make a 1-D histogram of the TAC data, and a 2-D histogram plotting ADC vs TAC for each hit. It should be possible to determine the optimal TAC-stop phase setting for each board:
    - If the TAC flip-flop becomes live (i.e. not being reset) too late then some early pulses will have normal ADC values but no TAC value, and the TAC values of the remaining pulses may be shifted to higher values. In the 1-D histogram this would show up as an enhancement at high values, with a cut-off at the highest values because the earliest pulses were missed. In the 2-D histogram this would show up as a line of data at TAC=0.
    - If the TAC flip-flop becomes live too early then some late pulses will have normal ADC values but no TAC value, and the TAC values of the remaining pulses may be shifted to lower values. In the 1-D histogram this would show up as an enhancement at low values, with a cut-off at the lowest values because the earliest pulses were missed. In the 2-D histogram this would again show up as a line of data at TAC=0.
    - If the TAC flip-flop becomes live SO early that the ADC gate has not yet opened then it is possible for really early hits to produce a maximal TAC value with no ADC value. In this situation the PMT pulse fires the discriminator and sets the flip-flop before the ADC gate opens. In the 1-D histogram this would show up as a spike at the maximum TAC value. In the 2-D histogram this would show up as a line of data at the maximum TAC value.

- Once the optimal value for the TAC-stop phase delay has been selected it can be put into the Tier1 file with the other RCC2 clock phase settings.

At this point the system should be fully configured!