TCU-4 Counters

H.J. Crawford, J.M Engelage, E.G. Judd, J.M. Nelson, M. Ng, C. Perkins, J.M. Landgraf

October 23, 2013

All of the trigger-related logic on the fourth version of the TCU is implemented in one FPGA on a daughter card. That FPGA contains all the logic for actually issuing triggers and also the counter logic for monitoring how often each of the various trigger conditions is met. This document describes which counters are implemented, and how they are controlled and read out.

Change Log:

Date	Description
May 20, 2013	Original Version. This document is a modification of the counter description from the previous version of the TCU, in which the trigger and counter logic were split between 2 FPGAS.
October 23, 2013	Updated table to show correct gating of PHYS+CONT counter. Also updated description of data sent to external trigger system to include the DAQ10k bits.

Counter Description

The following set of counters has been implemented:

Name	Description	Number of Counters	Control Signals
PHYS	For each trigger, the number of bunch crossings where the physics conditions are satisfied.	64	Run/Stop
PHYS+CONT	For each trigger, the number of bunch crossings where the physics and contamination conditions are satisfied. The contamination condition is satisfied if either the event is not contaminated or the contamination logic is disabled.	64	Run/Stop
PHYS+CONT+ LIVE	For each trigger, the number of bunch crossings where the physics conditions, the contamination conditions and the detector-live conditions are satisfied.	64	Run/Stop Token FIFO Empty
PHYS+CONT+ LIVE+PS	For each trigger that is enabled, the number of bunch crossings where the physics conditions, the contamination conditions, the detector-live conditions and the prescale condition are all satisfied and there are tokens available so a trigger is issued.	64	Run/Stop Token FIFO Empty
DET-LIVE	For each detector, the number of bunch crossings where the detector is Live	16	Run/Stop
TCU-DEAD	The number of bunch crossings where the TCU cannot issue a trigger because the token FIFO is empty.	1	Run/Stop
CROSSINGS	The number of bunch crossings	1	Run/Stop
TRIGGERS	The number of bunch crossings where a trigger was issued	1	Run/Stop
RESPONSES	The number of bunch crossings where a response was issued	1	Run/Stop
	Total	276	

The first 8 triggers each have four physics components instead of just one. The satisfied conditions for all four components are OR'ed together in order to determine if the overall PHYS condition is satisfied. In order to avoid unused components contributing to the OR each component has a separate enable flag in addition to the overall trigger-enable flag. For these 8 triggers the PHYS, PHYS+CONT and PHYS+CONT+LIVE counters will only count if at least one component is enabled. In order for the PHYS+CONT+LIVE+PS counter to count the overall trigger-enable flag must be set in addition to enabling at least one component. The remaining 56 triggers have just one physics component which is always enabled. In this case the PHYS, PHYS+CONT and PHYS+CONT+LIVE counters can always count.

Each counter is 40 bits wide. All the counters are cleared in parallel during the configuration process (see VME Control and Readout section below). After that each counter is incremented by 1 during every RHIC clock tick where its input bit is set and the counter is enabled. The control signals listed in the table for each counter are used to enable/disable that counter.

In order to use the counters for real-time monitoring it is necessary for them to be read-out frequently (e.g. every few seconds) when a run is in progress. To this end, periodically, the current value of each counter is latched into a static register. This happens in parallel with the normal process of incrementing the counters, and therefore does not interfere with the incrementing procedure. Once it has been saved, the data in the static registers is read out over VME while the TCU is still running. The handshake between the TCU and the VME client is described in detail in the VME Control and Readout section below.

Even the bunch crossing counter, counting at RHIC clock rate, will take over a day to count up to 2^{40}. No STAR runs are that long so we should never have to deal with overflows. As a result, the counters do not need to be cleared after the current value has been read-out. The values read out during a run are therefore running totals. The values read at End-of-Run are the final run totals. No offline addition of intermediate counter values is needed to get the run totals.

VME Control and Readout

The scheme for the counter readout is one that is initiated by the VME client. At some point the VME client initiates the readout process, and all the counters simultaneously save their current value into static registers. Since the TCU uses the VME32 protocol, and the counters are all 40 bits wide, there are two static registers for each counter. With the current list of 276 counters, this results in 552 registers. These registers are then read out over VME. A handshake is necessary between the VME client and the TCU counter logic.

The readout procedure is as follows:

- The VME client initiates the procedure by sending a “latch counter values” command to the TCU.
- When it receives that command, the TCU latches the current values of all the counters into static registers and sets a “data ready” bit in a status CSR
- The VME client polls on that bit until it is set, and then reads out all the registers
- The VME client then sets a “reading done” bit in a control CSR.
- When the TCU detects that the “reading done” bit is set, it clears the “data ready” bit.
- If the TCU is ever instructed to latch the counter values, but the “data ready” bit has not been cleared, this indicates that old counter values have not been read out and are about to be over-written. In this case the TCU sets an “error” flag in the status CSR
- If the TCU ever detects that any one counter had overflowed it sets an “overflow” bit in the status CSR

- If the VME client ever finds the “error” or “overflow” bits have been set it will log those error messages appropriately.

The registers that are needed for counter control and status are:

Name	Address Offset	Definition
Counter Control	0x YY70 0000	Bit 0: Clear all counters Bit 1: Latch counter values Bit 2: Reading done
Counter Status	0x YY70 0004	Bit 0: Data ready Bit 1: Error Bit 2: Overflow

Even though the TCU currently has 64 triggers implemented, we do not necessarily use them all every run. If a trigger is not enabled during a particular run, its associated PHYS+CONT+LIVE+PS counter will never see a non-zero input bit, so the counter value will always be zero. There is no simple way to zero-suppress the register list in VHDL. So, there will be a fixed one-to-two counter-to-register mapping. The VME client can choose whether to read all registers every time, or make a run-specific list and read just those registers associated with enabled triggers.

External Scaler Data

In addition to the counters described here the TCU also sends data in real time to the separate STAR scaler system. It has access to 3 connectors on which to drive that data. It sends the following bits on each connector:

Connector	Data
DSMI Output Connector 0	PHYS[0:15]
DSMI Output Connector 1	PHYS[16:31]
TCUI Scaler Output Connector Bits 0:15 Bit 16 Bits 17:19 Bits 20:23	 Detector-Live Status Contamination Bit Unused DAQ10k Word

It should be noted that there are not enough bits on these connectors to pass all 64 PHYS bits. It was decided that it would be more useful to STAR to have access to just the first 32 PHYS bits and use the remaining space on the connectors to pass detector status information. The contamination bit is the signal that is used to veto triggers that have the contamination logic enabled. The DAQ10k Word does not actually go to the scaler system. It is used instead to select a TPC readout mask for the DAQ10k readout scheme.