

051013: HjC

Final Requirements for FMS digitizer (QT) boards – Modification II –

051013: change in clock input delay (3.d.), killer bits(6), interface to L0 (8a), latency (8f)removed HALT requirement(9),and change to odd/even for charge injection (14)

050926: change to read out raw data, not output of LUT.

The FMS is designed to measure the energy and position of neutral particles emitted in the forward direction at STAR. It consists of ~1500 PbGl crystals, each viewed by a photomultiplier tube. It is intended to act as a fast detector for STAR. We summarize the requirements this places on the electronics, and their justifications.

1. Dynamic range and sensitivity: 0-200 GeV, 0.05 GeV: The 200 GeV max and 50 MeV sensitivity require 12 bits of dynamic range for the digitized signal.

We need to measure energies in a single crystal up to the  $x_F=0.8$  for the 250 GeV proton beams because the minimum  $x_B$  for gluons is probed with maximum  $x_F$  for quarks, and because spin effects increase with  $x_F$  of the detected meson. We need sensitivity to low energy photons such as those produced in highly asymmetric electromagnetic decays so that we maintain good direct-photon capability.

For operation at  $\sqrt{s}=200$  GeV, 100 GeV full scale should be sufficient and the sensitivity would be 0.025 GeV/count. Assuming the average gain quoted by Cumalat for the XP-2202 and the E-831/E-731 voltage divider operated at -1200V ( $g=0.12 \times 10^6$ ) and 1000 p.e./GeV for photons incident on lead glass, 0.025 GeV / count implies  $(0.025 \text{ GeV/count}) \times (1000 \text{ p.e./GeV}) \times (0.12 \times 10^6) \times (1.6 \times 10^{-19} \text{ C})$  or  $\sim 0.5 \text{ pC/count}$  for the ADC.

The final sensitivity (pC/count) will be established after experience with the 6U prototype circuit.

2. Signal capture:  $\sim 80$  ns active capture time.

Shower development within the PbGl blocks leads to photon arrival time at the PMT cathode spread over as large as 50ns.

3. Clock

3a. Clock input: Receive STAR standard 9.4 MHz clock from RCF output via multi-drop cable.

The board needs to operate with a STAR standard clock. We use the RCF instead of the TCD because we want to synchronize local memories on each board using the RCF run/stop line.

3b. Local oscillator: Register selectable.

The board needs to operate on a local clock for testing.

3bb. Clock indicator: A Front panel LED will indicate which clock is active.

3c. ADC Gate: Register selectable gate start time and width with respect to clock leading edge. One gate setup per board. Sensitivity of 2ns, 6 bit range on delay (0-128 ns), 6 bit range for width (0-128 ns).

3d. Clock Input Delay: provide register selectable delay for clock leading edge at input. Steps of 5 ns. With a range of 105 ns. (5 bits).

This will allow 16 boards to share a single multi-drop cable from the RCF.

051013: Changed the sensitivity from 1ns to 5ns to match timing of ADC gate, and changed range from 25ns to 104ns to make sure we can align within a single experiment clock.

3e. PhaseLockedLoop: A PLL will be used to minimize effects of lost clock pulses.

4. Background suppression: time stamp “hits” with an accuracy of  $< \sim 5$  ns.

The detector is placed as near to the beam pipe as geometry will allow so that we can probe the highest  $x_F$  values, making it susceptible to the near-beam background radiation fields. At a collider, the background circulating with the beam is low and typically out-of-time compared to particles arriving from the intersection diamond. At least half the co-moving background is trivially suppressed with a gross timing cut. With the detector  $\sim 8$ m from the center of the diamond a hit may be out-of-time by  $> 50$ ns ( $2 \times 8\text{m} / 0.3\text{m/ns}$ ); such a hit will actually occur in a previous crossing's digits, suggesting a need for “killer bits”. Measurements at STAR at the location planned for the FMS indicate that much of the remaining background can be eliminated with a timing resolution of 5ns, while still allowing for variation in vertex location within the diamond.

5. Discriminator

5a. Discriminator thresholds: A single threshold will be used for a board. The range will be from 10mv to 100 mv.

5b. Discriminator signal length: There is no requirement on minimum length.

Discriminator outputs will be latched on leading edge of the input pulse over threshold. The latches are reset to 0 on the rising edge of each RS (RHIC Strobe  $\approx 105$ ns period). If a discriminator happens to still be over threshold at that time, the latch will not get set unless the input signal drops below threshold and then rises above the threshold again.

5c. Discriminator outputs: Each discriminator output will be driven on the outer rows of the p2 connector.

We can use these 32 bits as input to an auxiliary board for any fast multiplicity logic requirement that may arise later. The output standard will be selected for suitability.

5d. Discriminator test points: test point on front panel for each discriminator output.

6. Killer bits: Separate bit for each channel with a register to set length (measured in RSs). A single register for length should suffice for a board, with a maximum of 16 ticks.

In addition to the timing problem noted in 4 above, the PMTs may have significant after-pulsing which we want to eliminate from trigger considerations. The killer bit should cause suppression of the ADC and TDC but allow the discriminator to fire.

7. Rate capability: operate at 10 MHz to match the STAR system:

The STAR trigger is based on a fully pipelined dead-time-less operation allowing maximum use of the luminosity. This is similar to the clock requirement of 3 above.

8. Interface to STAR Trigger Level0 (L0)

8.a. Provide at least 32 bits of output from each QT board to a DSMI.

These 32 bits will be driven on p3 backplane to a new Digitizer Interface Board. We expect to provide high-tower ADC value (HT), tower number, and ADC board-sums to DSMI boards for input to DSM tree of Level0 trigger. We expect to use 3 layers for the FMS, just like for FPDW, with all information from the QT boards entering at layer0. Our  $\pi^0$  and  $J/\psi$  signals are based on reconstruction of energy and position of pairs of hits. DSMI input is taken in 16 bit quanta. Note that 32 channels of 12 bit ADC can lead to a sum requiring 17 bits, a tower ID requires 5 bits, and a high tower requires 12 bits, making 34 bits: this indicates we need a bit selection mechanism. The FPGA code will be reconfigurable via VME. It will load from an eeprom on power-up, but the bits in the eeprom can be rewritten via VME.

8b. Bit selection for L0 information: Need a register to select which contiguous bits of ADC sum and another register to select which contiguous bits of HT value are sent to L0.

8c. Defining bits to L0: Allow redefining the 32 bits going from QT to L0 via FPGA recoding,

It may evolve that a different set of bits are better suited to L0 triggering. Presumably we could code the FPGA to deliver a different set to the 32 lines that go to the DSMI.

8d. VME Reset: The board can be reset via standard VME sysreset command.

This will cause the FPGA to be reloaded from its eeprom. In this way we can reset a crate without power cycling. Using sysreset allows us to reset a full crate with a single command.

8e. Timing into L0: a register will be provided to specify delay of output from QT board to DSMI. Sensitivity of 5ns is sufficient.

8f. Maximum output delay from interaction: The board must present its values to L0 within 440 ns of the interaction.

This allows data to enter the decision tree in time to complete in 15 RS. The QT board requires a minimum of 3 RS to complete its digitization cycle.

## 9. Readout

9a. through VME backplane using either DMA or standard VME reads to intermediate memory for shipment to L2.

We may read using DMA or, for 0 suppressed data, standard VME reads. Either way we could read into the CPU memory, or we could read directly into the STP PMC memory. This has the advantage that the data would begin shipping to L2 as soon as the first bytes got to the STP PMC.

With 16 boards in a crate and 32 channels per board this means 512 channels per crate. Using 12 bit ADC and 5 bit TDC we would want to fill to 32 bits per channel (VME standard word length), or 2048 B per event per crate with no pre-post. Using DMA from each board, but not chain-block-transfers, the time is 5 mic-sec o'head per board + 55 ns/byte or  $5+128*0.055=12$  mic-sec/bd or 196 mic-sec/crate. The QT board is different from the CDB in that the data are not all shipped to a DSM for each crossing. Thus, we need local storage for data.

9b. Local memory: Require 8 MB of token addressed memory per QT board.

The easiest local storage is to use token-addressed memory leading to a requirement of 32 chn x 4 B/chn/xing x64k xing (7 ms worth) => 8 MB of local storage. Each channel would store its ADC and TDC value for each crossing.

We use STP cards to receive L0 commands into the crate CPU and to ship data to the L2 CPU for further trigger processing and data logging.

Any overflow condition in the board will be flagged by the TCU.

9c. Memory synchronization: Use an OR of (RUN/STOP signal from RCF) with (contents of a register) to set the board into run mode.

This will allow all QT board memories to be synchronized so that the "correct" crossing information can be selected for a given trigger. Allowing a register to put the board in run

mode makes debugging much easier. Because we want the VME register to be able to force two states: RUN and STOPPED (over-riding the RCF signal) two bits are required: one to say "obey RCF or local" and one to say RUN or STOP when in local mode.

#### 10. Zero-suppression: Register selectable option.

Most events have only a few hits, so data sparsification is reasonable. We would need 28 bits to describe a single cell: 12 bit ADC, 5 bit TDC, 5 bit cell-in-board addr, 6 bit board address. (EJ points out that VME uses 8 bit board address – so use 8 bits here). This suggests 4B per channel per event. Note that in the heavy-ion program we will have much higher occupancy.

Many possible methods of compaction can be considered; here is one: begin with two 16bit words that represent a bitmap of all channels contained in the event followed by the 4bytes from each of the nonzero channels (in order.) The board ID can also be included at the beginning .

#### 11. Pedestal: All pedestal values will be positive.

It is important to maintain slightly positive pedestal values so that we understand the offsets in ADC spectra. Analog adjustments are costly and unnecessary. We can design in enough pedestal offset to be sure ALL channels are above 0V. The lookup table will remove the small variations without losing more than a few counts (out of 4095) of dynamic range.

“The discriminator is the only place where a voltage offset is relevant and that depends on the quality of the chip; 5mV is a typical spread. The integrated charge as seen by the ADC is subject to offsets in the op-amps but even a worst case of all offsets adding up to 20mV is still only 1% of the dynamic range. The LUT will allow you to set all pedestals to X plus or minus 1 count and I will be sure the raw pedestal is never below zero. 32 trim pots on each of 50 modules would be a giant pain!” (FB)

#### 11a. Noise: The pedestal should have an RMS variation of <1 bit in normal operation.

#### 12. LUT for gain correction/ped subtraction: Look-Up-Table between digitizer and logic.

To make a good sum and to provide an accurate high-tower sample each channel must be gain matched. The simplest way to correct for differing pedestals and gains is through a LUT. This should be 12 bits per channel. We do not need an LUT for the TDC, only for the ADC.

The voltage divider networks (VDN) for the XP-2202 have Zener diodes to hold the voltage across the last four dynodes, roughly independent of the voltage applied to the VDN. Significant departure of the high voltage applied to the VDN may lead to non-linearity in the relation between the charge (or ADC count) from the phototube and incident light. To avoid these non-linearities, we may be forced to use software calibrations rather than high-voltage adjustments. The LUT would allow us to compensate for phototube gain variations at the trigger input.

NOTE: we do not need to read the LUT output for each event into the data stream, only for test purposes. Our main data path will send non-LUT corrected data to L2.

### 13. Monitoring:

#### 13.a. Front panel Test points for gates and signals.

It is useful to be able to see the relation of the signal and gate as they appear at the input to the digitizer. We want to monitor a sample of the analog signal on each input (suggest <10%) as well as the discriminator output for each channel. A separate monitor point for the ADC gate (one per board) and the TDC start or stop (1 per board) will be useful. This makes  $2 \times 32 + 2 = 66$  test points on the front panel.

#### 13.b. Readable ADC values.

We want to be able to read the values that are going into the data stream in a fashion that does not use the full board readout. Token memory for local storage allows VME reads.

#### 13.c. Readable TDC values. Again, token memory allows this.

### 14. Test input: Inject a fixed charge into each front end for testing.

This will simplify board testing. A register will be used to determine which channels (odd, even, or both) are injected. The odd/even selection allows us to check cross talk and makes the FPGA code much simpler than to use a bitmask for each channel as we do on the CDB.

### 15. Snapshot operation: Provide facility to suspend storage in local memory so that the board's memory can be read out via standard VME commands.

This has proven to be a valuable diagnostic tool for the DSMs. We take those snapshots by having the RCC put everyone in load mode and then reading the memory via VME reads.

### 16. Form factor: The board will be a 32 channel 9U VME board.

This sets the number of crates required for the full complement.

### 17. Data rate capability: We want to record every event for which the FMS can produce a trigger.

In current STAR running we are allowed to run "FAST" detector data when the TPC is BUSY, but not when it is LIVE. For our physics program we can take useful data consisting of the FMS, the BBC, and the bunch-crossing bits alone, regardless of the state of the TPC. Such events should impose no deadtime on the rest of STAR and should be run "invisibly", except for the overhead of event building. Rather than include a detailed

discussion here of any special data paths, I'd like us to discuss the various options for running within the STAR framework and see whether we really need a separate trigger and data path.

Brief description:

The requirement for widely adjustable integration times has led us to develop a dual integrator front end for the ADC. One integrator is active while the other is being reset in each RHIC clock cycle of 105ns. The integrator is alive only during a gate time whose leading edge and width are register selectable anywhere within the 105ns RHIC clock period. At the leading edge of the next clock cycle the integrators are switched and the last active integrator presents its signal to the 12 bit 40 MHz digitizer. Output from the digitizer is shipped to a Field Programmable Gate Array (FPGA) for packaging and LUT translation.

Output from the discriminator is used as input to a 5ns sensitivity time-to-digital-converter (TDC). The TDC is based on a counter operating at 200 MHz and counting the interval between the discriminator signal and the clock. This leads to a 5 bit TDC value which is stored and reset at the leading edge of each RHIC clock signal.

ADC and TDC values for each channel will be stored in a STAR-standard 64k deep token memory. The memories will be aligned to 0 when the boards are put into run mode, just like the current DSM boards are.

The FPGA will route each channel's digital signals to local memory for storage until receipt of a trigger. The CPU will then read the token-specified memory locations for each board and ship the data via STP fiber optic cable to a PCI receiver card in a linux CPU. The FPGA will also treat groups of 32 crystals as trigger patches: it will form sums of the 32 ADC values and it will select the highest of the 32 ADC values as a "high tower". It will then send up to 32 bits (e.g. 17 bit sum, 10 bit high tower, 5 bit tower ID) to an existing STAR Data Storage and Manipulation Interface (DSMI) board in the trigger.