

# Front-end Electronics System Design Issues

*K. Einsweiler, LBNL*

**Collection of comments on FE system design details and concepts that need to be re-examined or modified for the final production versions (so-called “Phase II”)**

## Pinout Issues

### Comments on modifications to existing demonstrator pinout:

- Number of detector connections needed is one or two (now DGUARD and DGRID)
- Simplify present command interface (CCK, LD, DI, GA0-3) ? It seems to work well, and the wire-bonded GA is very robust against SEU, so would propose to leave it alone.
- Simplify reset interface (RST and SYNC generated by common pins with length encoding function ?). Would prefer old-fashioned RSTb pin, and separate SYNC pads which are differential as now.
- Simplify calibration interface ? Combined Strobe and VCal is very flexible, supporting digital and analog injection with internal and external “chopping”. Propose to leave it alone, keeping the Strobe asynchronous and differential.
- Eliminate all bias pins (I1 - I8, VTHR and VCOM) except for MON\_REF as a test point for the internal current source ?
- Simplify power connections ? Presently, have two sets of AGnd/AVdd/AVcc (needed to minimize internal voltage drops), and two sets of DGnd/DVdd (not needed for voltage drops, but good for bonding redundancy). Could replace digital pads with single set in middle of chip ?

## Command Decoder Issues

### Present protocol uses:

- CCK serves as lower frequency clock (5 MHz). The presence of this clock (as opposed to XCK system clock) signals that there is command decoding to be done. This clock could be dropped, and all command decoding could operate at 40 MHz. In this case, activity on DI would signal the presence of a command.
- LD serves as timing signal to separate command portion (LD low) and data portion (LD high). Differing implementations as far as “dataless” commands (0 or 1 CCK under LD), and in terms of readback of internal registers (n or n+1 CCK needed). This should become standard in the future. If the command decoder operates at 40 MHz, the timing of LD is more critical.
- Addressing provided by GA (hardwired pins on chip) or broadcast. Simple and robust scheme.
- Overall chip DO can be used for readback of all registers. Powerup state defined to provide useful diagnostics (command loopback mode).

## Reset Issues

### Different classes of reset useful:

- Hard Reset (implemented as pin RSTb): puts chip into a known state, including default values for the Global and DAC registers (most FF cleared to zero, some set to one). This ensures the chip does not enter strange high current states upon power up. We do not yet have a power-on reset circuit, but we probably need one.
- Soft Reset (implemented as command): clears all digital readout circuitry to a known state, but does not affect configuration data. This is necessary if MCC is to issue such a command to clear problem conditions, because the MCC does not know how to reconfigure the FE chips.
- SYNC (implemented as differential pin): will reset trigger number to zero and clear all pending events. If no pending events, there will be no lost data if SYNC is issued.

## Internal Registers

### Four basic internal registers in FE chip:

- Command Register: basic register which allows loading of other registers via generation of enables and strobes. Recognizes the GA field of an incoming command and checks against external GA pins. Will be simplified in new chips so all bits represent actions and not states (e.g. EnableAnalog enables analog charge injection and belongs in the Global Register)
- Global Register: register which includes shadowing to prevent asynchronous updates. Should contain all Global state information, including column mask.
- DAC Register: register which includes shadowing. Contains values for all global DACs in chip (FE bias adjusts, etc.)
- Select Register: long register, one bit per pixel, which controls loading of all pixel-level FF (Strobe Enable, Readout Enable for now)

### Reliability:

- Global and DAC register contain static information vital to chip operation. There is some concern about SEU.
- Suggestion from Peter Fischer: include checksum information when writing these registers, monitor continuously, and return error status in End-of-Event word. This provides information on whether given event had right configuration data.

## Some new register functions needed:

- Column pair masking for extra “robustness” (in case column pair fails catastrophically). This would control bypassing of Select Register around column pair as well as bypassing of sparse scan around EOC for column pair to prevent bad bits or stuck pixels from disturbing the operation of the whole chip.
- Ability to disable all non-command digital activity on FE chip using Global Register to allow studies of impact on module operation.

## Functions to support in MCC (presently on PCC)

### Calibration functions:

- VCal provided with dual range 8-bit DAC. It is assumed to drive a 10 fF injection capacitor in the pixel.
- The ranges should be: [0,100] mV or [0,6] Ke in steps of 25e to be used for threshold scans and [0,2000] mV or [0,120] Ke in 500e steps to be used for TOT calibration and cross-talk studies.
- Programmable Strobe Delay. It would be very useful to have a 7-bit delay with a 64 ns range (0.5 ns steps). This is very useful for timing studies, and in particular in-situ timewalk measurements.

### Monitoring functions:

- Provide HitBus support for self-triggered source testing of modules. The HitBus is programmable FastOR of pattern of pixels in a given chip, provided in differential form. FE chip could provide ORing function via “daisychained” input and output, but would add two pins.
- Not clear how else to implement this in a real module. Cannot multiplex with return data, and adding a separate differential pair to each FE is a major change in module connectivity.

## Global Issues

### Output Data:

- Do we want to continue to provide TOT data ? Almost certainly yes. It would be natural to limit the range to 7-bits (the maximum L1 latency), but shortening beyond this does not seem like a good idea.
- Present FE data format OK otherwise ? A single header bit seems to work well enough for communication with MCC.

### Architecture issues:

- Improve throughput of data from pixels to EOC to limit effects of column pair occupancy fluctuations.
- Present FE-B architecture has fairly hard “failure” when the column-pair transfer bandwidth is saturated - it can take quite some time to clock out all of the pixels, which are useless because they do not enter the EOC buffers in time to see their LVL1 trigger accept.
- Perhaps this requires some kind of “lifetime limit” for data in column, but this requires quite some circuitry to implement reliably.
- Possibilities also exist for clustering of data within the column pair to reduce data rate. This requires careful thought about the data format.
- Further discussions will take place in designers meeting.

## Event Handling and Error Conditions

### Present FE-B event handling:

- Internal 16-deep FIFO contains the 7-bit BCO plus error bits (OVFL only for now) latched at time of LVL1 trigger, for each pending trigger (event).
- The Write pointer is incremented by the LVL1 accept, and the Read pointer is incremented by the data-push readout controller in the FE chip.
- MCC flow-control of the LVL1 accept is critical to the successful operation of the FE readout controller block. Present FE-B chip can sometimes misbehave when sent too many triggers in short interval, leading to a state where the data transmission is stuck transmitting the same End-of-Event word over and over again, even in the absence of incoming LVL1. A more graceful failure mode needs to be designed in for the future.

### Overflow Handling:

- When last buffer in a given EOC block becomes occupied, the Full flag is set. This is stretched by the LVL1 latency, and latched into the readout FIFO sampled by the LVL1. Hence, it provides a stretched error flag whose presence indicates the possible loss of data due to buffer overflow. Presently, this flag is simply an OR of all EOC Full flags, and is inserted into the End-of-Event word. This scheme seems adequate.

- If no free locations are found in the EOC block, incoming hits are thrown away until a free location is available. The Full flag will remain set, so the presence of the condition will continue to be flagged.

## Synchronization:

- MCC uses the 4-bit trigger number included in each pixel hit word sent by the FE chip to control the merging of data from different MCCs into a common event structure.
- If the different FE chips on a module ever become “unsynced” due to a spike on the LVL1 line, etc., then all subsequent event building will be erroneous. This would normally lead to timeouts because the one chip that is “ahead” in trigger number will have skipped a valid trigger number and transmitted no End-of-Event word.
- The MCC generates SYNC pulses to the FE chips whenever it realizes that they have no pending events. Note this signal is bussed on the module, so all FE chips must simultaneously be empty before SYNC can be issued.
- Further simulations of this simple algorithm at the module level are probably needed. It is not obvious that it produces a self-recovering system, which is what we would prefer...
- At some stage, the MCC will have to withhold LVL1 triggers and issue either SYNC or SoftReset to clear the confusion. Again, simulations are needed...

## Timing Signal Conventions

- All critical timing signals (LVL1, SYNC, DO) are referenced to the leading edge of XCK.
- There is no critical requirement on the falling edge of XCK. However, the DORIC is designed to implement a very high quality 50% duty-cycle clock output. This will deteriorate somewhat during LVDS to CMOS and CMOS to LVDS conversions due to different currents, loads, and delays in buffers, but should remain acceptable.
- The spec for the demonstrator was that a variation of XCK high from 10ns to 15 ns was allowed (+/- 10% variation in duty cycle).