**PEPPo Helicity Trigger with the FADC250** (3/21/12 – EJ)

The FADC250 firmware has been modified for use with the helicity change signal (‘T\_settle’) supplied by the accelerator. As described in an earlier note by Hai Dong, the input channel signals are integrated while T\_settle is low (helicity state valid). This note focuses on how to set up and coordinate the FADC and trigger interface (TI).

The T\_settle signal is always active. It is applied to the FADC through the front panel signal distribution module (SD). The differential ECL trigger input on the SD should be used. The FADC will ignore this signal until the trigger to the FADC module is enabled (FADC CTRL2(1) = 1). When the FADC trigger is enabled, the control logic waits until the *next* valid helicity interval (i.e. T\_settle low) before it applies it to the FADC processing chip. While T\_settle is low, the processing chip sums the 4 ns samples of each FADC input channel. When T\_settle goes high, the sums (+ header, time information) are stored in a buffer. The FADC control logic then generates a trigger out pulse to indicate that data is ready for readout. This differential ECL pulse appears at the MT1 output pins of the SD (assuming that port 1 of the SD is connected to the FADC). The MT1 output should be connected to a trigger input of the TI. Upon receipt of the trigger pulse, the TI generates an interrupt that initiates readout of the FADC (and other modules).

To end the run the user disables the trigger to the FADC module (FADC CTRL2(1) = 0). Processing will continue until the *end* of the current helicity interval, when the last trigger out pulse is generated. Depending on the helicity change frequency, this may take up to 33 ms. A status bit (FADC CSR(7)) is asserted while processing (i.e. integration) is active. The user should not disable the TI until this bit is negated. Or the user can simply wait the appropriate time after disabling the FADC front panel input.

The above scheme guarantees that each event represents a complete helicity interval. It also assures that no data will remain on the FADC board at the end of a run, and that no improper (e.g. too short) integration interval can be applied to the FADC processing chip that may adversely affect its behavior.

Sequence of Software Operations

1. Configuration of FADC (e.g. clock source, trigger source, block size, DAC, soft reset).
2. Configuration of TI; enable TI trigger input.
3. Start run - write FADC CTRL2 = 0x3 (enable FADC trigger, enable internal data transfer).
4. End run -
5. Write FADC CTRL2 = 0x1 (disable FADC trigger, keep internal data transfer enabled).
6. Poll for FADC CSR(7) = 0, or WAIT > 100 ms.
7. Write FADC CTRL2 = 0x0 (disable internal data transfer).
8. Disable TI trigger input.

**Note**: The actual integration signal applied to the FADC processing chip is driven out of the busy output of the FADC. It is available as differential ECL signal ‘busy’ on the SD. (Write SD register busy\_ena = 1).