OSU status update 6/22

PSA

Electronics status

- DDA rev C, TDA rev A: boards in production (qty 15: enough for 3 stations + 3 spares)
  - TDA rev A: ~Monday/Tuesday
  - DDA rev C: July 5
  - Stencils are also in production, should arrive Monday/Tuesday
  - Mass production should be quick (2-3 days)
- ATRI rev B: at assembly house, pick’n’place programming done, board assembly in progress

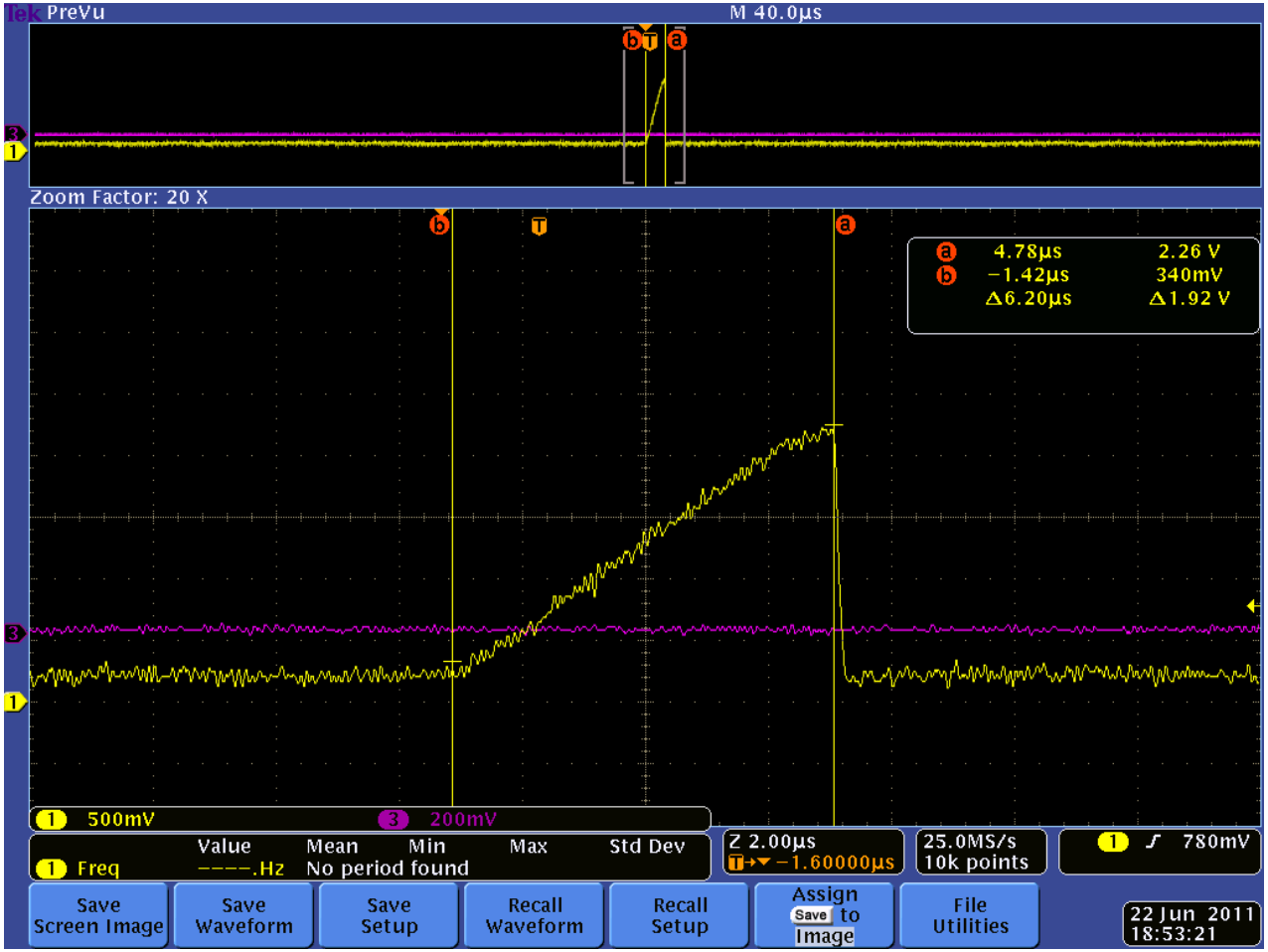
ATRI rev A testing

- Hacked-up ATRI rev A testing slowed due to incredible crosstalk from fly-wiring data bus (pickup was almost 1V!)
  - Had to rewire clock as clock+ground, cover with copper tape, etc.
- However, now have been able to talk from FPGA to USB
- Porting DDA_EVAL firmware to ATRI in progress
  - Just need to replace the ethernet_packet_interface with a "usb_packet_interface" module
  - Almost done: testing again slowed due to ridiculous timing constraints needed for FX2 USB interface
    - 18.7 ns setup time on a 20.83 ns clock period!
  - Had same problem with ICRR, however Spartan-6 has *worse* problems due to longer propagation times across the chip!
    - Newer isn't always faster…
  - Problems resolved, though: just had to mostly rework the logic to push the outputs into the IOBs and advance the clock enough so clock-to-out is the limiting time
    - Previously we just had the router work really hard to keep delays down

DDA Wilkinson ramp mods

- Default IRS_EVAL values way off
  - 100 pF Cramp, 20k Isel resistor
  - Gives ~2.5 us ramp time (6.2 us Wilkinson clock)
- Trial and error gave 100 pF Cramp, 33k Isel gives a little less than 6.2 us ramp

RampMon output (Sbbias = 50k to 2.5V)



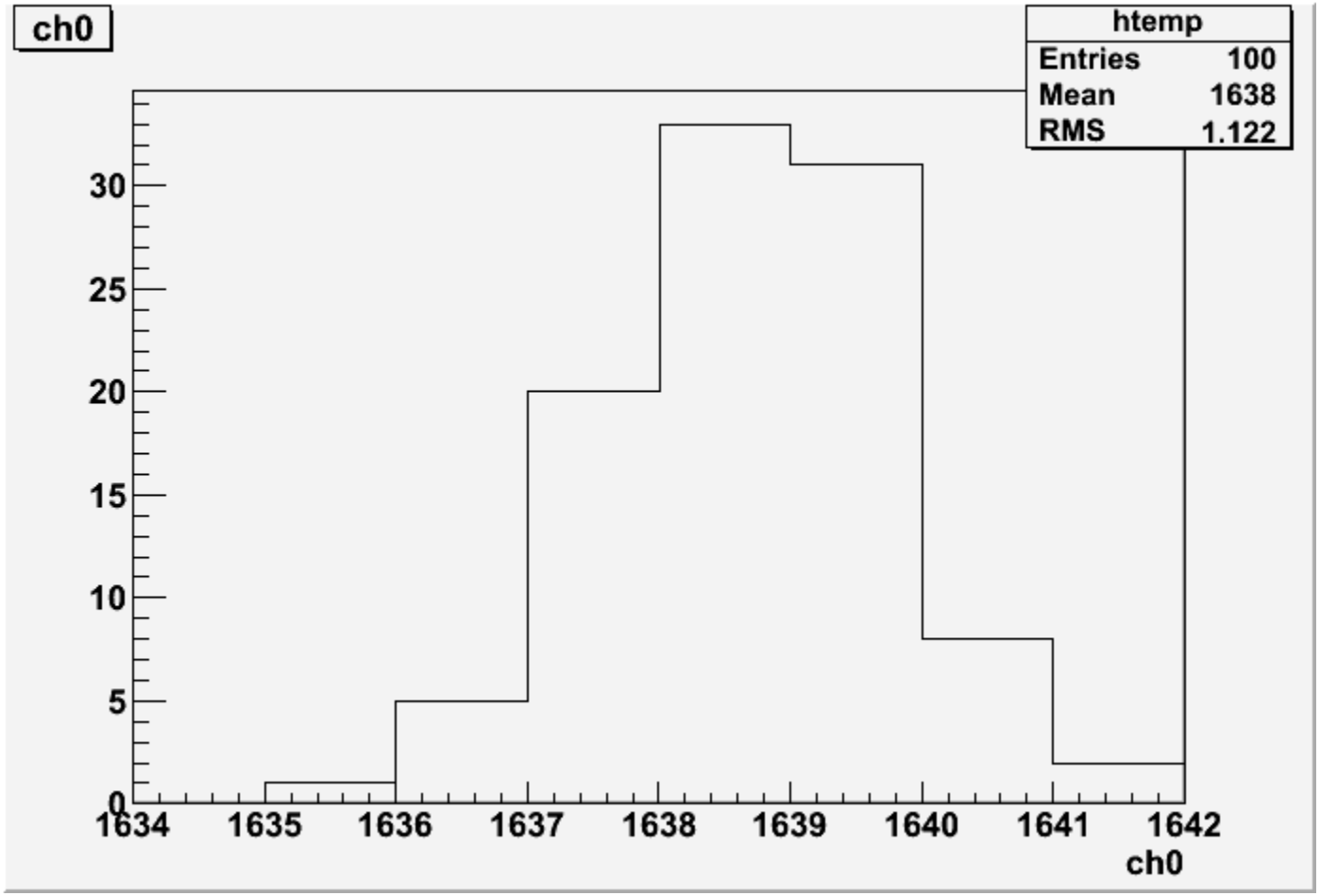
irs_block_manager improvements

- Previous block manager had no way to select a given block for readout
  - This makes getting pedestals very difficult
- Modify block manager to have a “pedestal mode”
  - In pedestal mode, sample a given address once (in the correct TSA phase) then set WR[9] to 0 (disabling writes)
  - Same pedestal can be read out again by asserting a pedmode_clear flag
  - Pedestal mode is basically a separate path to the normal block manager path
  - When pedestal mode entered, when the end of the normal 6-cycle (3 block) path is reached, transition to pedestal mode
  - When pedestal mode exited, transition back to the normal path (at end of 6-cycle pedestal path)
- Free block queue/active buffer not affected in pedestal mode, but lock register is, so the idea is:
  - 1: disable all triggers
  - 2: finish all readout
  - 3: switch to pedestal mode (specifying a given block address)
  - 4: read out as many times as you want
  - 5: when done, finish all readout
  - 6: end pedestal mode
  - 7: reenable triggers

Pedestal mode details

- irs2_block_manager has three new inputs
  - ped_mode_i: 1 if pedestal mode is on, 0 if not
  - ped_address_i: 9-bit block address
  - ped_clear_i: Take another sample (reassert WR[9] for one sample)
- In DDA_EVAL, accessible via IRS WISHBONE module
  - Address 0x0024: bit 0: ped_mode_i, bit 1 (self-clearing) ped_clear_i
  - Address 0x0025,0x0026: ped_address_i (little-endian)

Pedestal example



Procedure for pedestal readout

- To read out block 0x0123 100 times
  - Program FPGA
  - Start ddad
  - Run pedtrig 0x0123 100
- ddad will save the events as evdump#.dat, where # is the number of events received in the run so far (just a running ID in ddad)
- ddad doesn't need to be restarted for each block address, just run pedtrig (block addr) (ntimes)

“ddad” test program

- ‘ddad’ is somewhat of a prototype for ‘atrid’, an ATRI controller program
  - *Not* an acquisition program: just multiplexes access to the DDA_EVAL/ATRI
- Currently it accepts data for the control data path via a Unix domain socket at “atri_control”
- Any event data it receives it dumps to a file using its own internal event ID
- Event/control data paths are threaded, so event data is immediately dumped as soon as it’s available