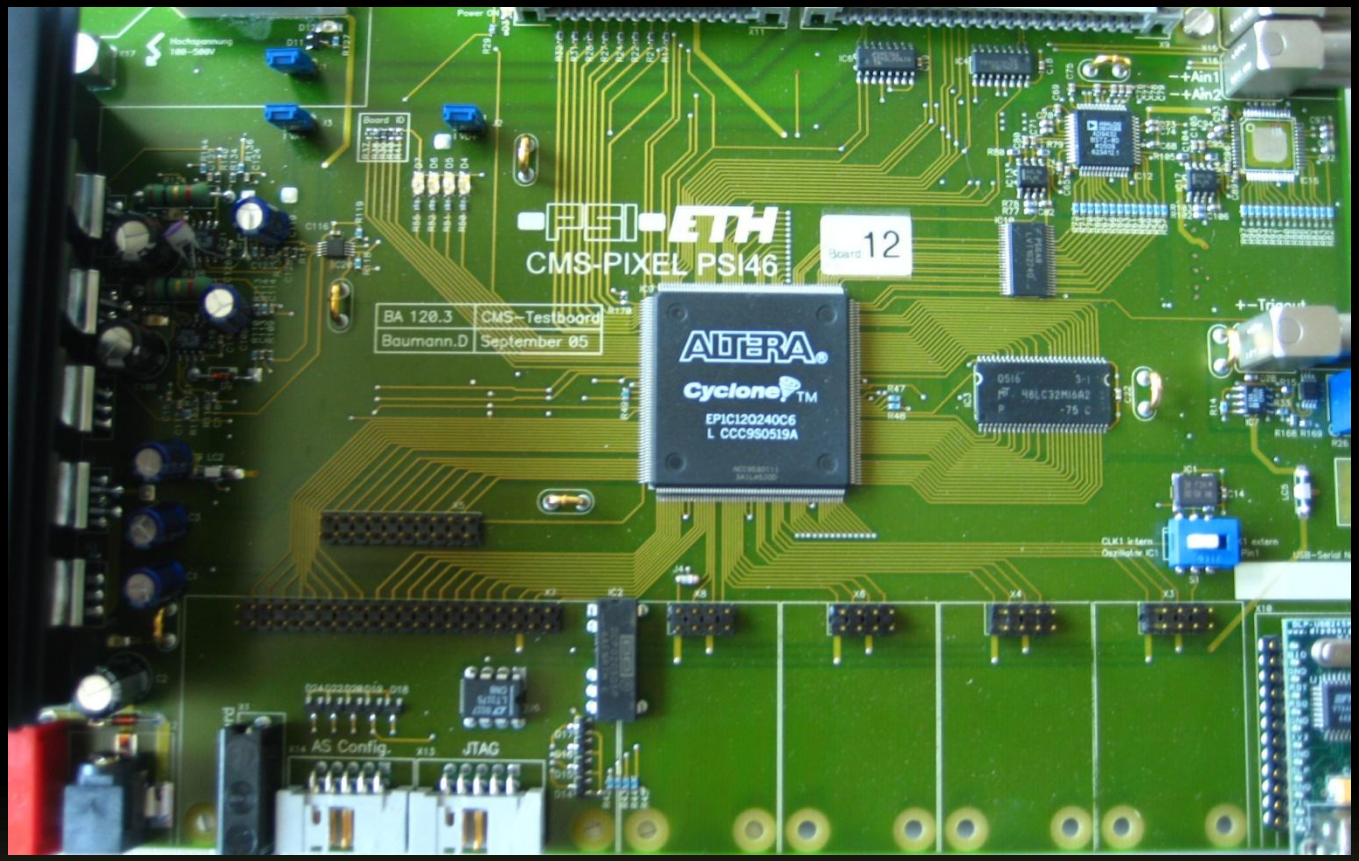
TBM EMULATOR PROJECT SUMMER 09



17 Sep 2009

Chris Martin

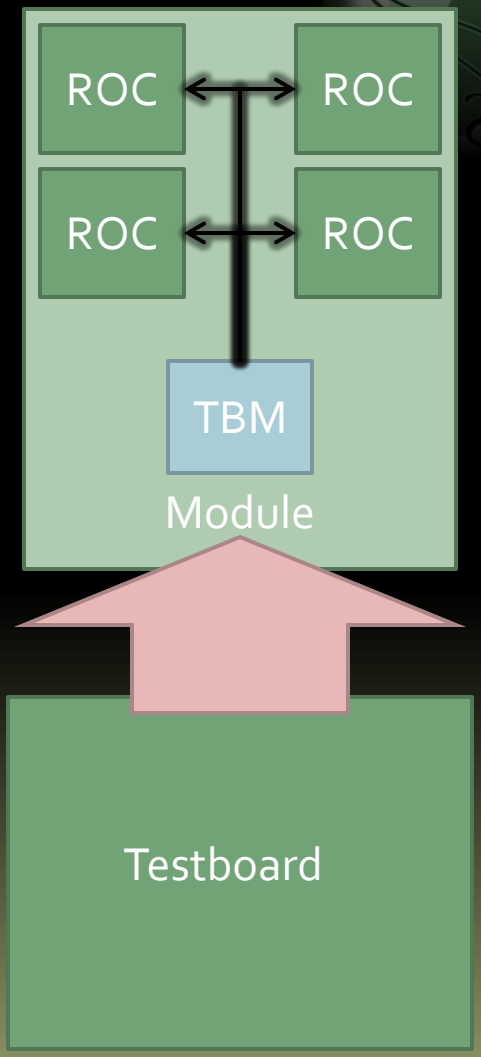
University of Kansas

Overview

- TBM Introduction
- Firmware changes
- Simulations
- TBM emulator
- Status of New Testboards

TBM Introduction

- Pixel modules are read out using a Token Bit Manager (TBM) chip
  - Organizes and deals with multiple events
- Interface between Read Out Chips(ROCs) and outside world.
  - Receives trigger then asks the ROCs to give data
  - Passes a 'Token' to each ROC to get its data
  - 'Wraps' in nice packages
  - Stops the outside world from confusing the ROCs by sending Tokens too fast



TBM Introduction cont'd

- The TBM speaks with the Firmware on our test boards
- Robert (student from Berlin who studied at ETH) incorporated some of the logic of the TBM into the Firmware
  - Trigger stacking
- 'Wrapper' signals were removed
  - TBM Header and Trailer no longer written to output
  - Causes problems when analyzing the output data

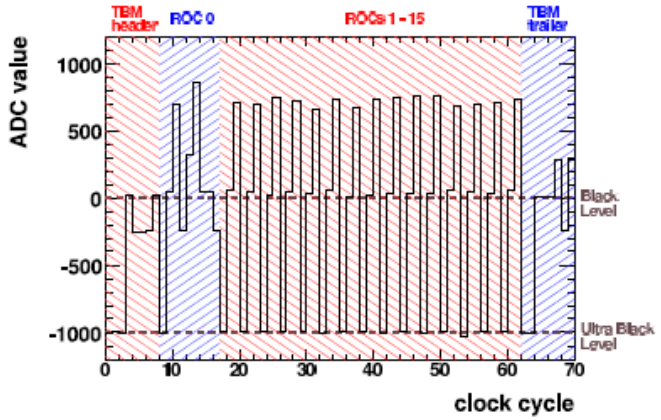
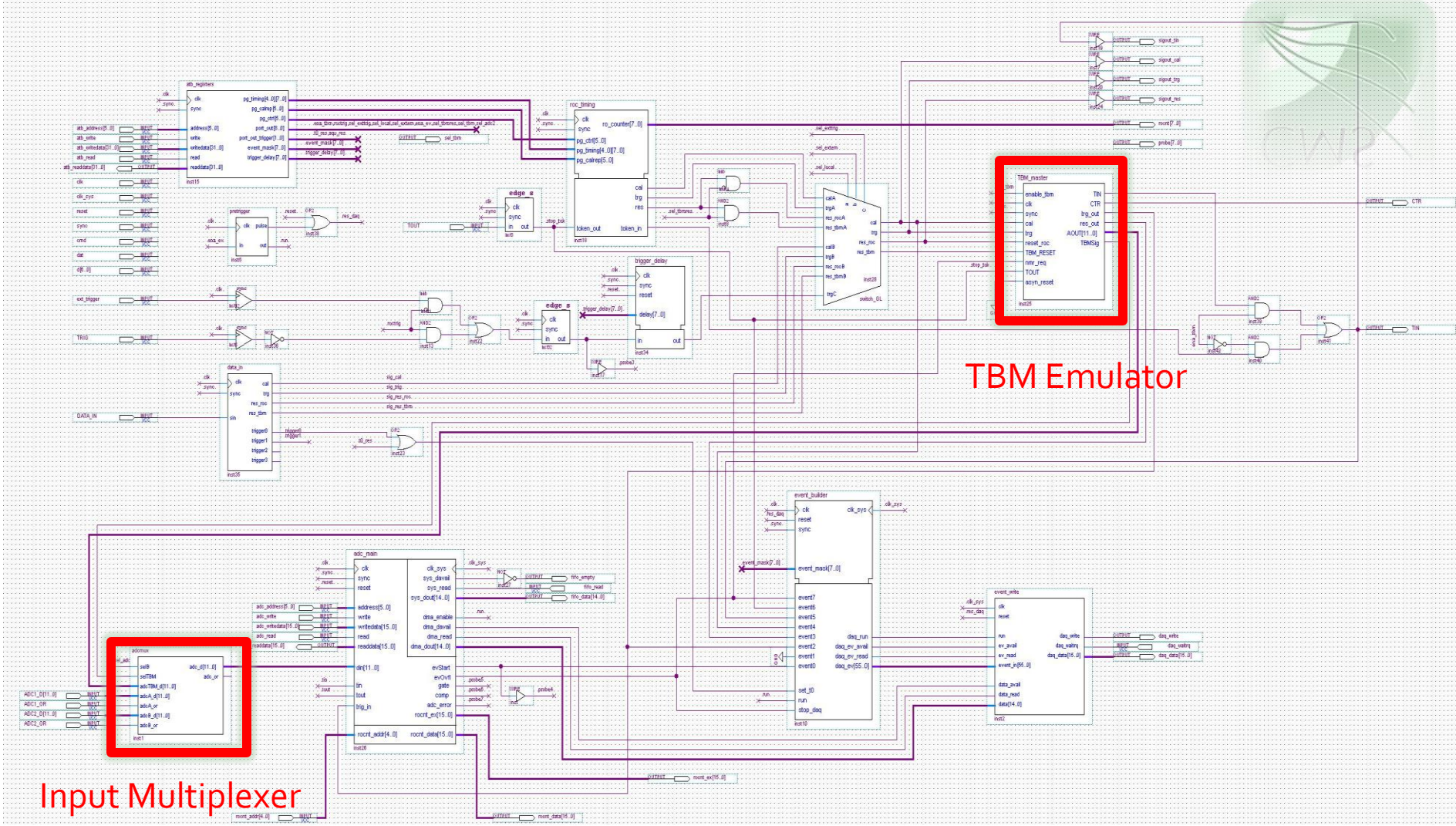


Figure 11. Output of a module with one pixel activated on ROC 0

Firmware Changes

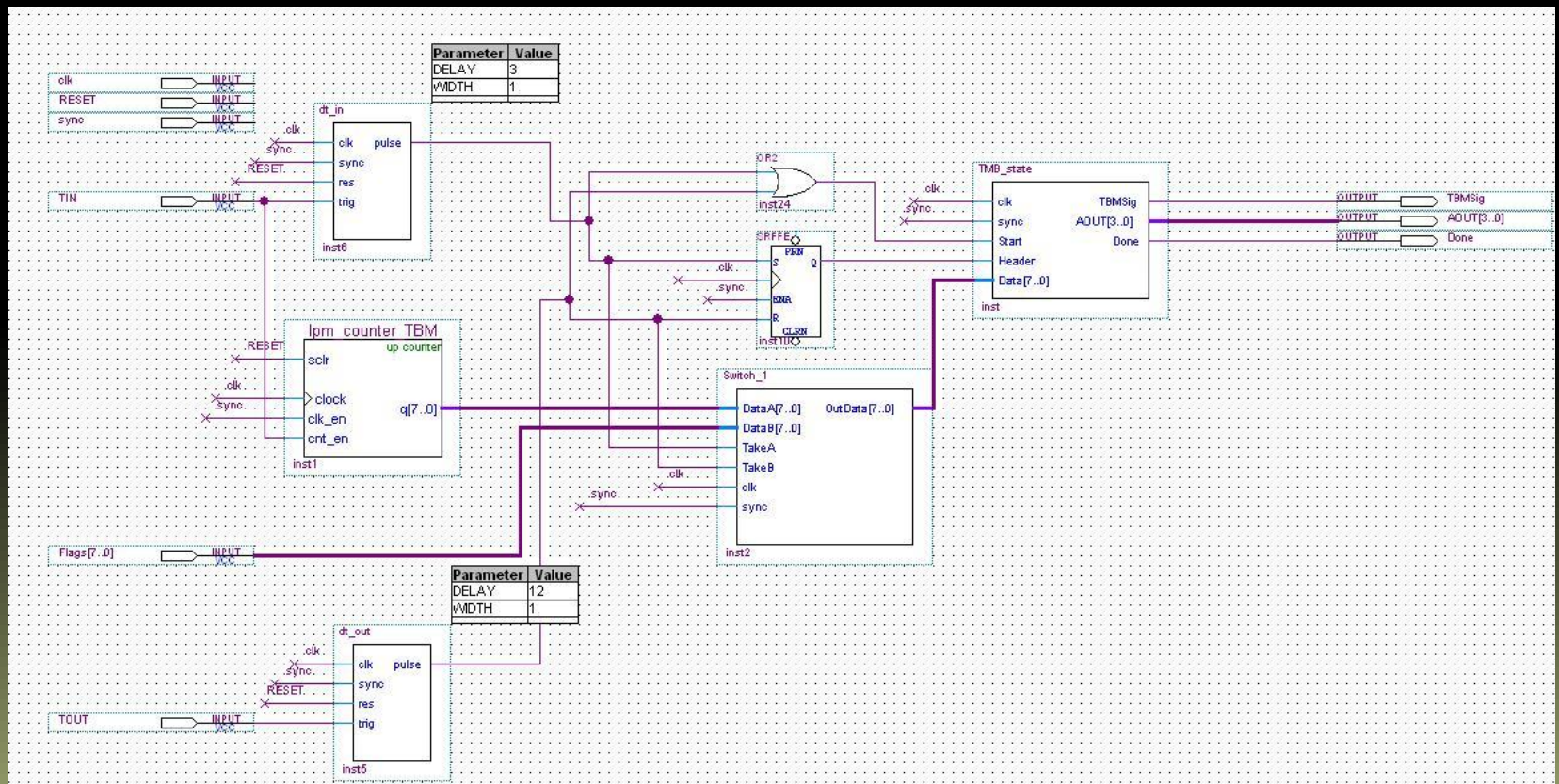
- Emulate the TBM on Testboard Firmware
- Two fold functionality
  - Trigger Stacking and management (Most of this was done by Robert (THANKS!!))
  - Signal Generation
- Must keep in mind…
  - Event counting
  - Data flags
  - Interaction of software with TBM (set registers)
  - Timing!!!! SUPER Important
  - Analog address levels

Firmware Changes



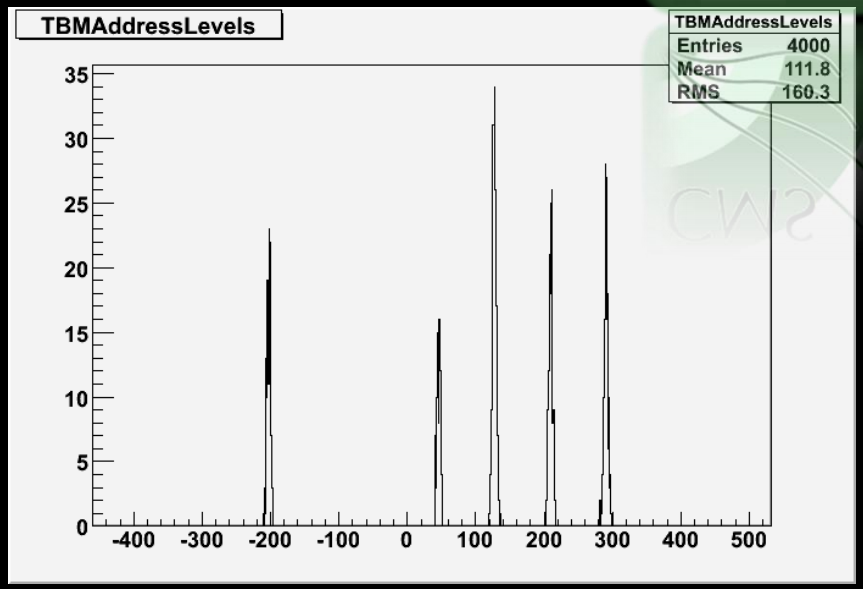
TBM Emulator: Signal Generation

- Simple State machine and Multiplier[not pictured] interfacing with Roberts Token Stack
- Produces event counter and flag signals



Signal Generation

- Software expects the TBM to output 5 distinct address levels that they use to decode a TBM Header and Trailer (right)
- Software expects a specific pattern of signals for reconstruction (below)



Header

1) Ultra Black
2) Ultra Black
3) Ultra Black
4) Black
5) 2 MSB's of Trigger Number
6) 2 Bits of Trigger Number
7) 2 Bits of Trigger Number
8) 2 LSB's of Trigger Number

Trailer

1) Ultra Black
2) Ultra Black
3) Black
4) Black
5) 2 MSB's of Trailer Status
6) Trailer Status Bits 5 +6
7) Trailer Status Bits 3 +4
8) 2 LSB's of Trailer Status

Input Multiplexer

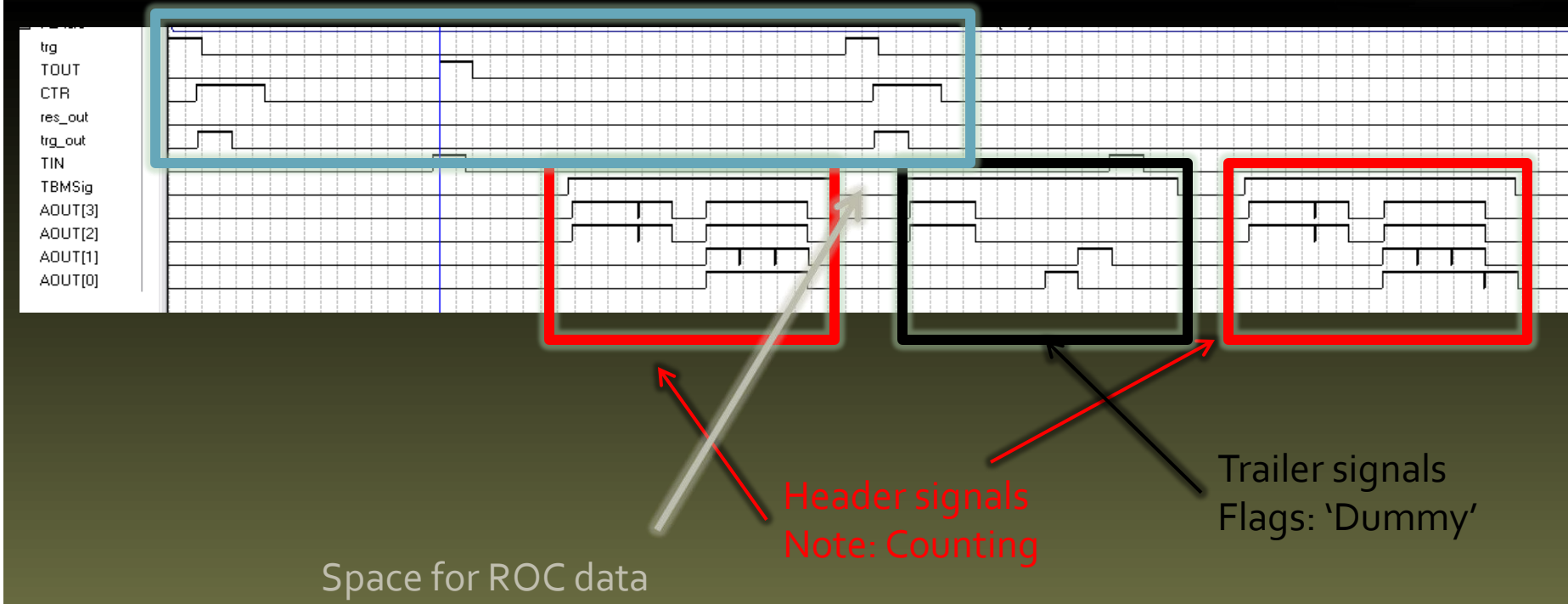
- Write TBM Emulator signals to output
  - Adding another channel to the ADC multiplexer
- Appropriate ROC output
  - Adding a multiplier to ROC input stream
    - Amplification done by TBM
  - Add offset to ROC input stream
    - Different "Zero" levels of testboards… This needs to be fine tuned for each board

Interfacing with the system

- Trigger and Token passing
  - Timing issues dealt with to ensure no data corruption
  - Interfacing with Roberts Token stack
- Enable/disable Registers
- Stop software from ‘adjusting’ levels

Simulations

System Interfacing



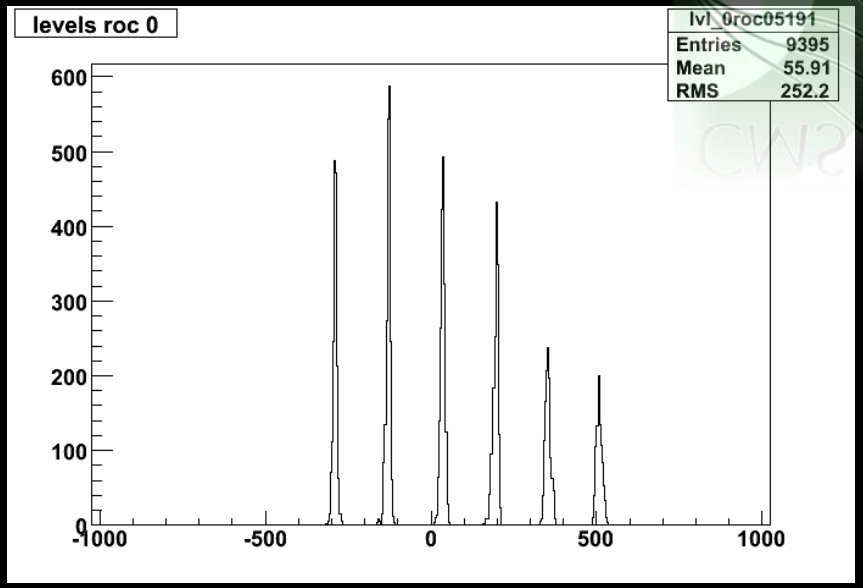
Real Testboard

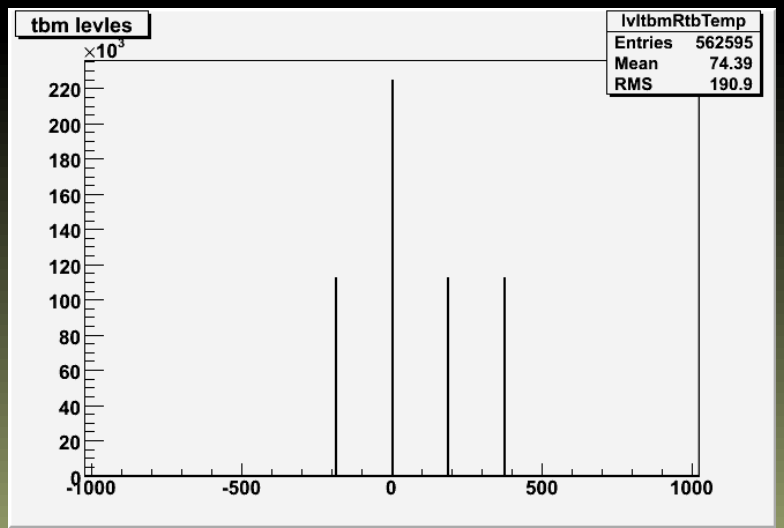
- ROC chip (no sensor)
  - Header and Trailer are both present
  - ROC Header immediately follows the TBM Header
  - Delay after ROC info before Trailer output
    - Old output, timing has been fixed
  - TBM and ROC levels don't match
    - Fixed as discussed above

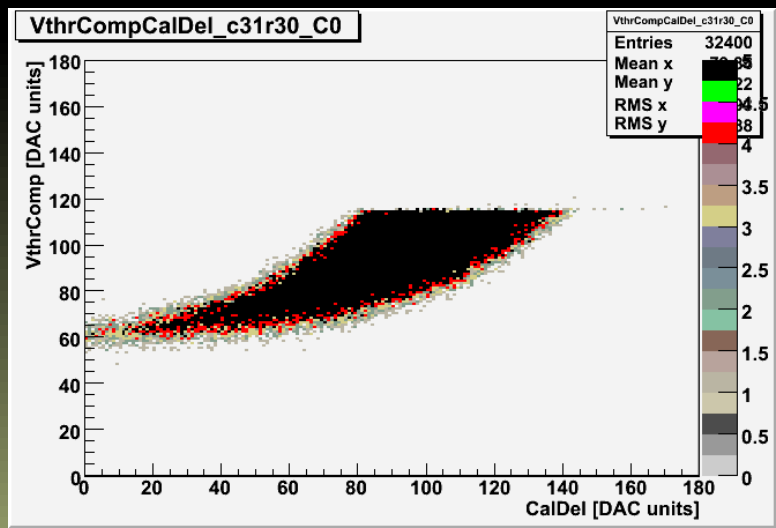
```
>dread
  60 samples
  41  42  41  41  40  41  41  40  40  42  41  41  40  42  40 -300 -300 -300   0 -75   0   0 150 -159  48
175  40
TRAILER:  41  41  41 -300 -300   0   0 150 -75  75 -75  42  41 41  41  41  40  41  41  41  40  42  41
40  41  41  40  42  40  40  40  41  42
```

TBM emulator with psi46expert

- Programmed 2 testboards and found the correct parameters to run the psi46expert calibration software!

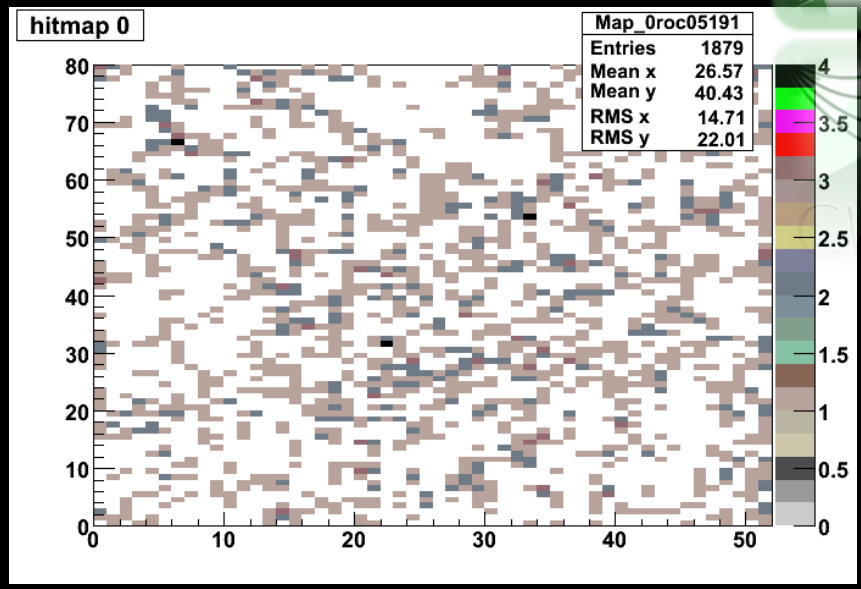


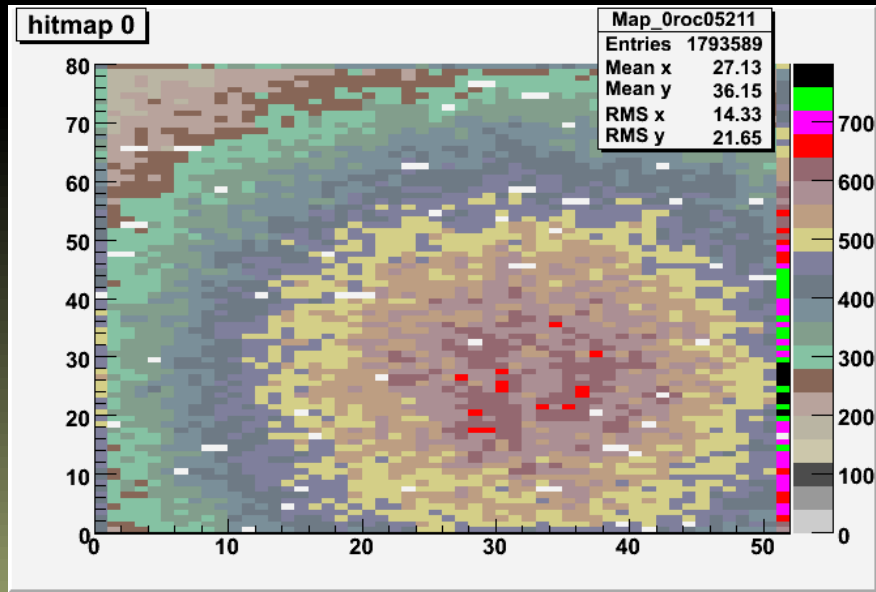




TBM emulator

- But wait…
  problem with TBM emulator and takeData program
- Look at the pixel map plots
  - See only a fraction of the events we saw before
- A WBC delay problem
- Jennifer found the correct values and everything worked!





PSI Linux software

- All of the above results are with a version of the psi46expert software that I have "retro-fitted" to not crash when it use the TBM emulator
- I believe a proper solution is in the "TBM_final" directory on pc5971
- However, all solutions now require very fine tuning of test board parameters… because of the sensitivity of the tests with the test board clock for each testboard that the new Firmware is put on a careful study of the timing parameters is needed.
- Even to run in the old mode with the new testboard now requires that the timing parameters are adjusted properly. This just simply takes time.

New Testboards

- Plan to produce
  - 10 Testboards (picture on title slide)
  - 10 Gatekeeper boards (right)
- Jose, Beat put together parts list
- UNL, UIC, KU purchased parts
- Jeff Worth (KU electronics expert) is populating boards

Status

- 1st Testboard almost done
  - Few missing parts (ordered)
- After completed I will test it
  - Check communication
  - Check Voltages and Currents
- After 1st Testboard is checked Jeff will make the rest
- Expect tests to start on 1st board next week