ECAL Readout

Paul Dauncey

For the CALICE-UK electronics group

A. Baird, D. Bowerman, P. Dauncey, R. Halsall,
M. Postranecky, M.Warren, O. Zorba

CALICE Readout (ECAL) Card

- CERC $\rightarrow$ CRC as will be used for AHCAL also!
- Eight front ends (FE), 12 ADCs each; handles one full or two half-full VFE PCBs
- Back end (BE) routes data in and out; stores event data in 8 MByte memory
- Trigger logic and readout in one BE; distributed on custom backplane to others
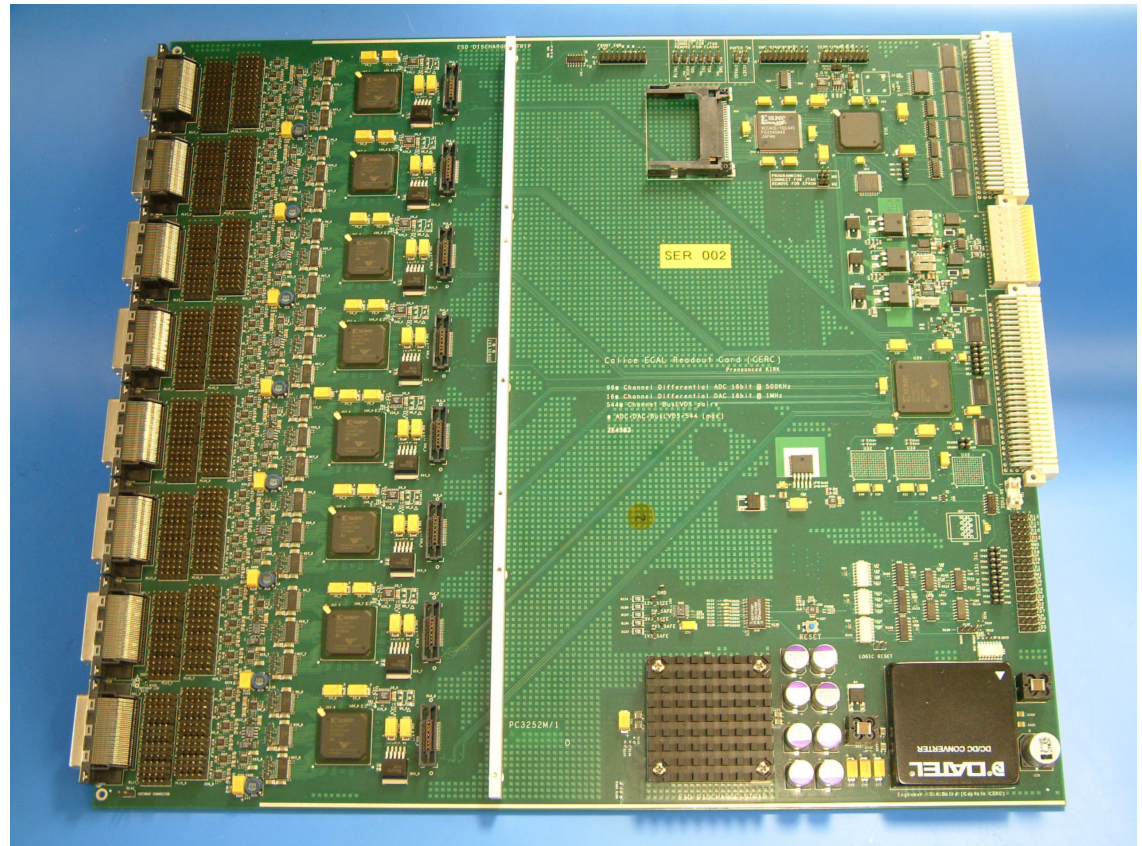
Production status

- Two production CRC boards returned on Nov 3; SER003,SER004
  - Seven more PCBs fabricated at the same time
  - Will be released for assembly when first two boards checked
  - Need six fully working boards for ECAL
- Several small assembly problems found
  - Missing resistor, shorted capacitor, bad solder joint to DAC, etc.
  - Board reworked to fix these at RAL and Imperial
  - Results shown here from after these repairs
- Now have firmware to fill and read out memory via VME
  - Orders of magnitude faster than RS232 readout used for prototype tests
- Rerun similar tests to prototypes to check boards
  - See my talk from CERN meeting, Jun 28, for prototype test results
- More details of production test results shown here available from

    http://www.hep.ph.imperial.ac.uk/calice/elecProductionTests/electronics.html
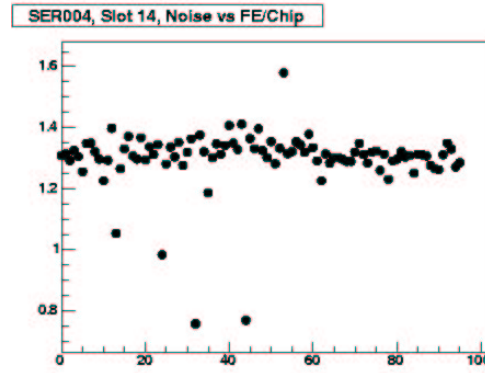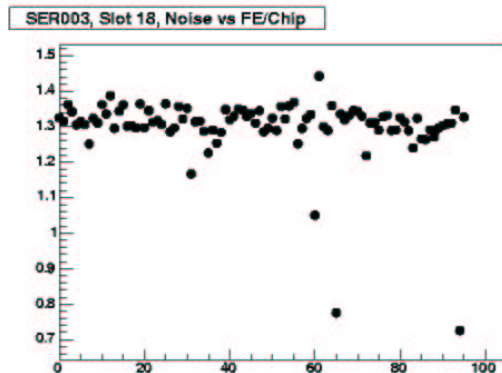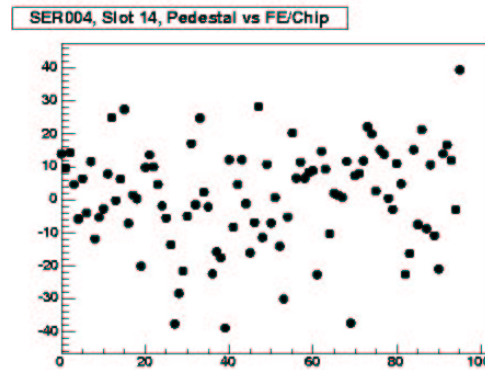
Intrinsic noise tests

- No VFE inputs; basic noise performance of ADCs; 96 channels
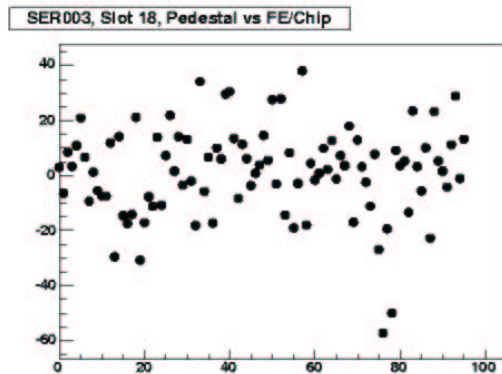
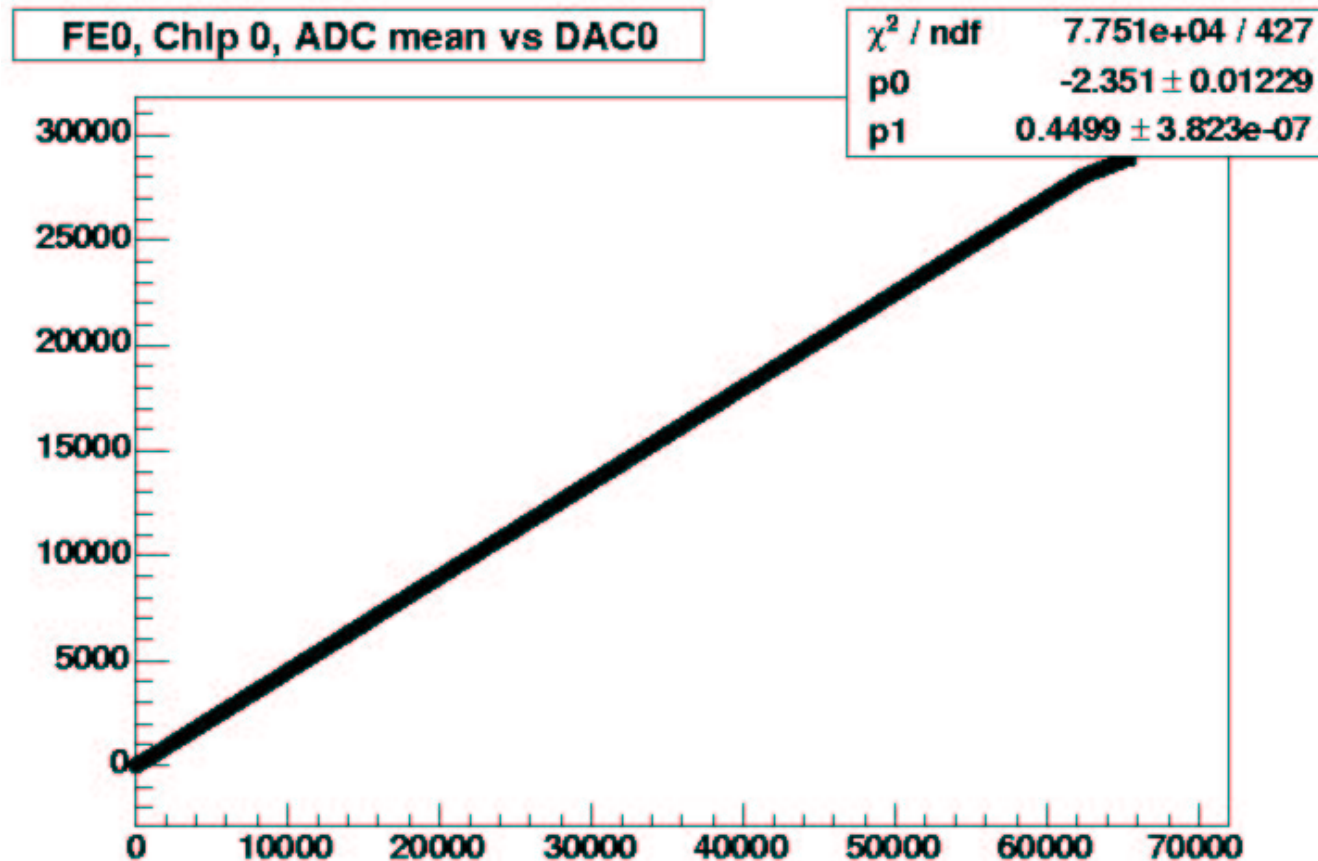SER003   SER004

- Pedestal within 50 counts of zero, average noise = 1.3 ADC counts
- 1 ADC count = 76μV diff (i.e. ±38μV). No obvious dead channels

Internal loopback calibration

- DAC looped back to ADC within dedicated path on board

FE0, ChIp 0, ADC mean vs DAC0

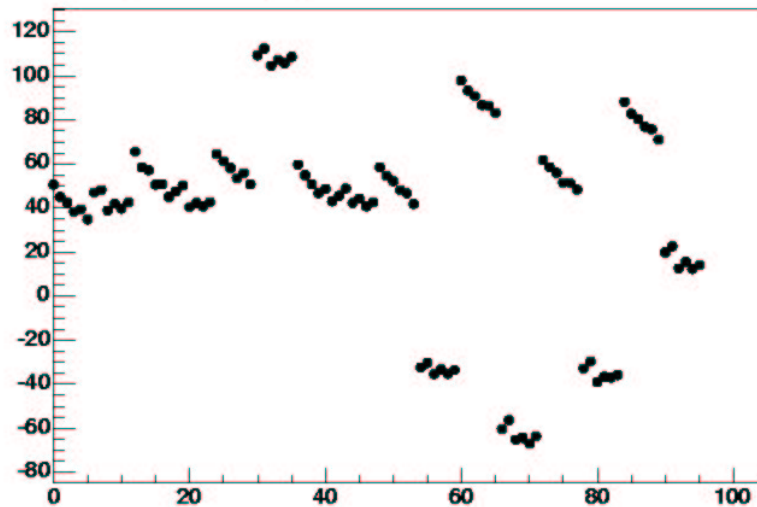| χ^2 / ndf	7.751e+04 / 427
p0	-2.351 ± 0.01229
p1	$0.4499 \pm 3.823\text{e-}07$

- Typical channel, linear response slope = 0.45 ADC/DAC counts
- High end saturation near max; no low end saturation any more
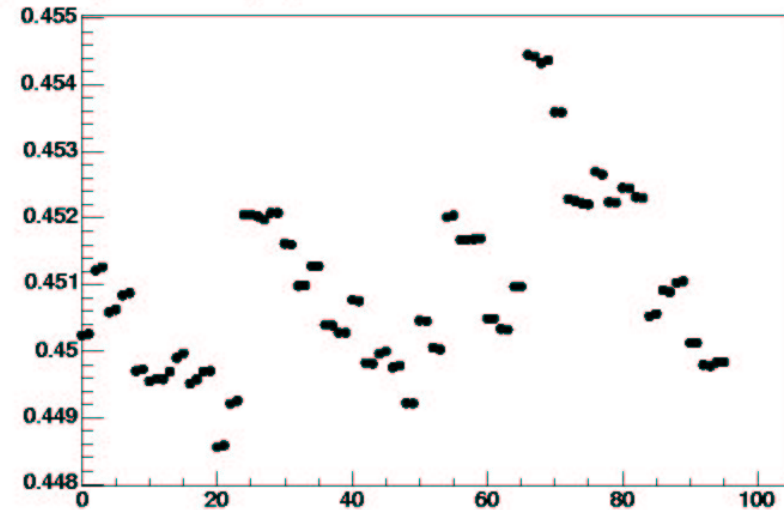
Internal loopback (cont)

- Interpolation and slope for all 96 channels

SER003

- SER004 similar
- Intercepts within 100 counts of zero; slope (gain) uniform to ±1%
- No bad channels on either board

External loopback calibration

- DAC looped back to ADC using wiring attached to front panel
  - Can invert the wiring to get either positive or negative voltages

FE0, Chip 0, ADC mean vs DAC

χ^2 / ndf	1.55e+05 / 62
p0	90.53 ± 0.008753
p1	0.2373 ± 2.546e-07

- Typical channel, linear response slope = 0.24 ADC/DAC counts
- Factor of two difference due to termination resistors

External loopback (cont)

- Interpolation and slope for all 96 channels
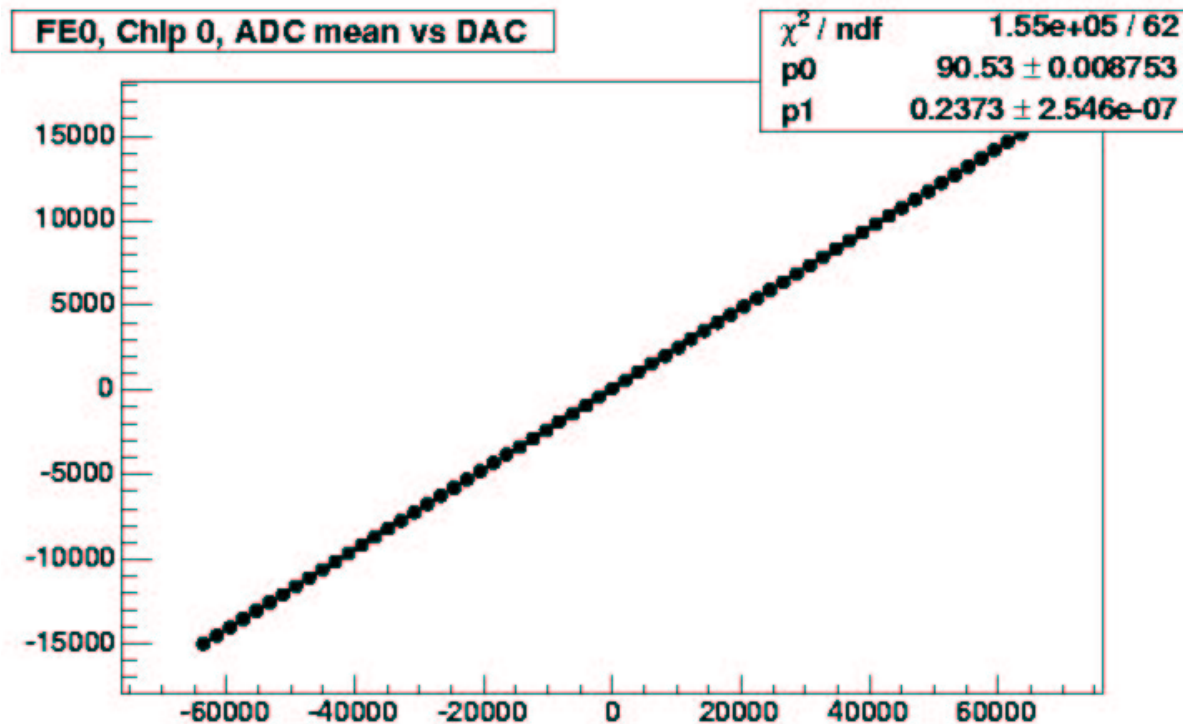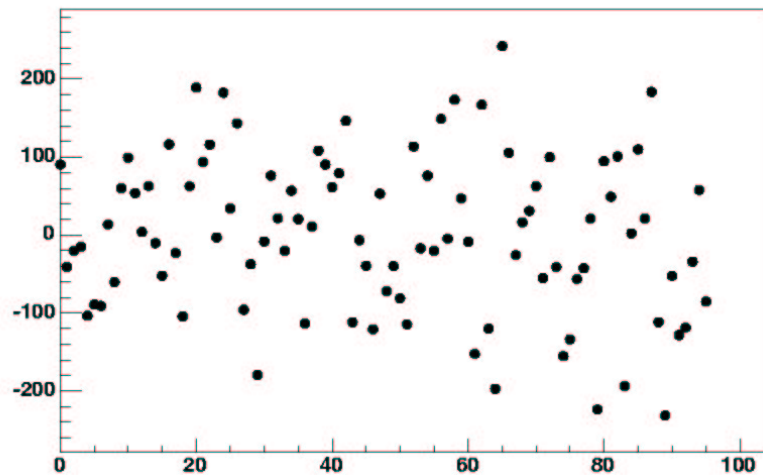
SER004
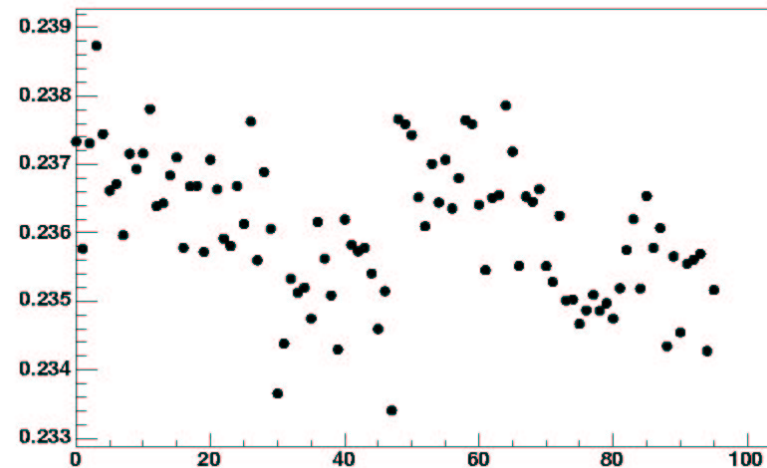
- SER003 similar
- Intercepts within 200 counts of zero; slope (gain) uniform to ±1%
- No bad channels on either board (again)

VFE calibration

- One full VFE PCB; 12 VFE chips, 216 channels (no wafers)
  - Calibration can be enabled to each group of 1/6 of all channels
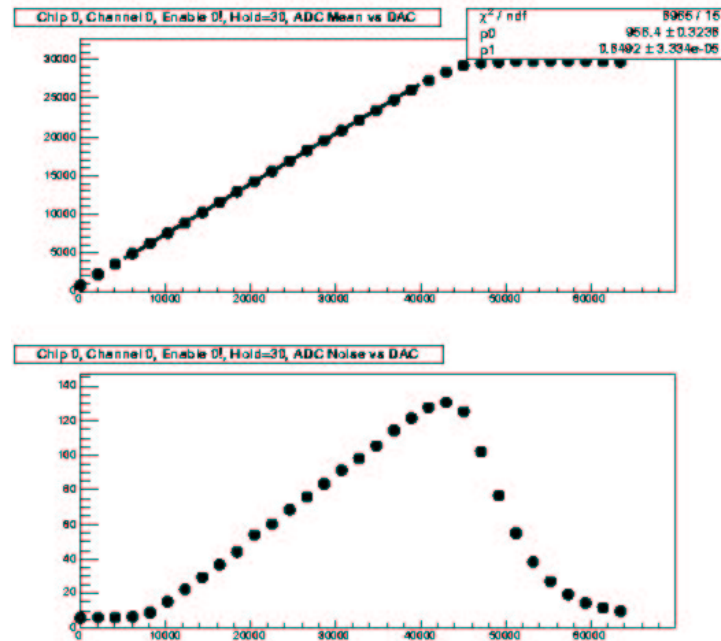
- Typical enabled channel, linear slope = 0.65 ADC/DAC counts
- High end saturation; limited by calibration chip as noise is similar
- Noise increases with DAC in unsaturated region; calibration chip?

VFE calibration (cont)

- Look at non-enabled channels to see crosstalk
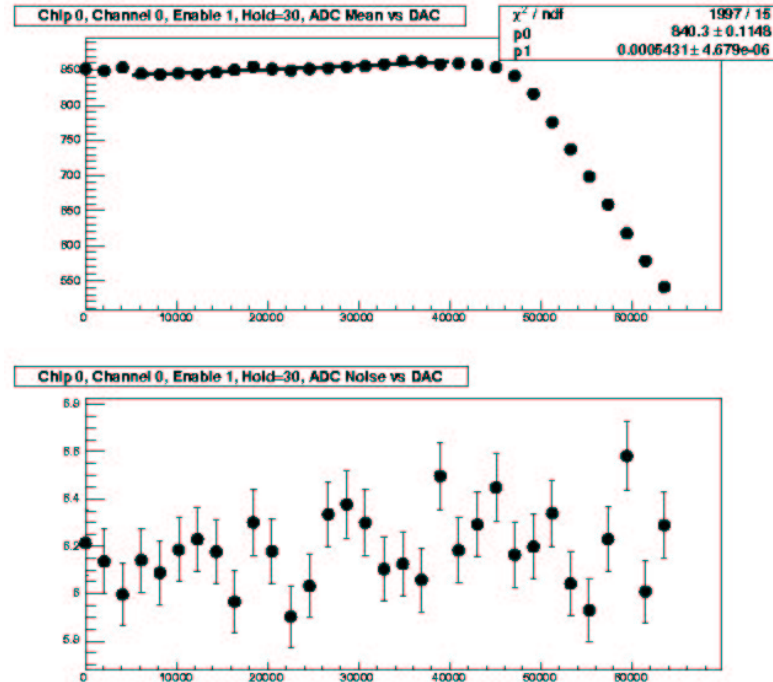
- Typical non-enabled channel, slope << 1% of enabled channel
- High end saturation; pulls down power line?
- Noise doesn't increase with DAC; stable at 6 ADC counts

VFE calibration (cont)

- Interpolation and slope for all 216 enabled channels
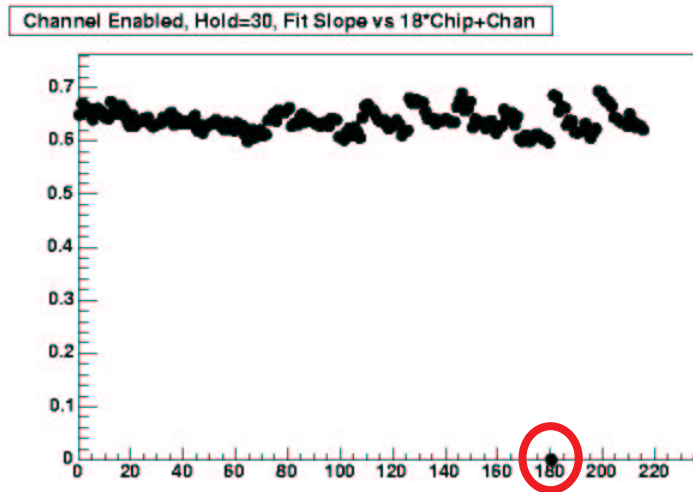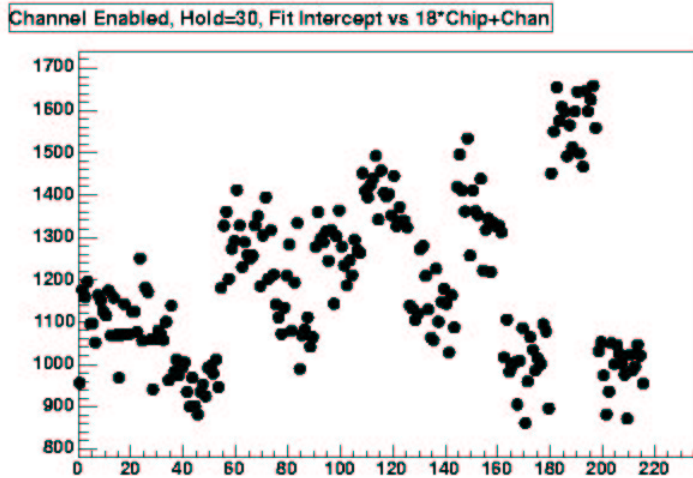
- Intercepts spread between 900 and 1700 counts
- Slope (gain) uniform to ±10%
  - Channel-to-channel variation dominated by VFE gain and/or calibration circuit
- One bad channel (Chip 10, Channel 0)

VFE calibration (cont)

- Look at the unresponsive channel when enabled
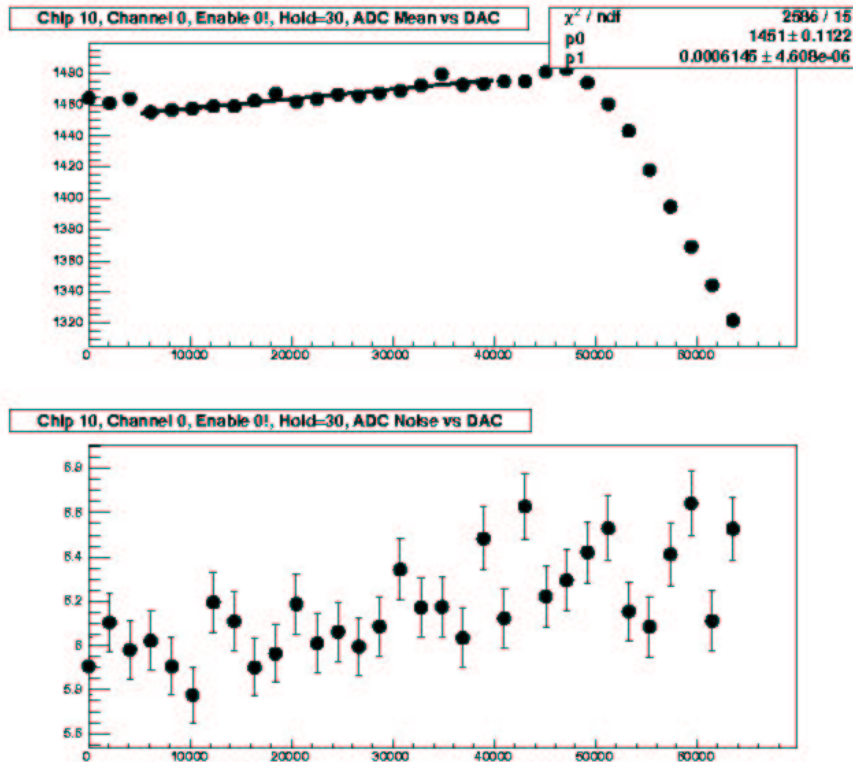
- Looks exactly like non-enabled channel
- Possible working channel but calibration circuit at fault?

VFE calibration (cont)

- Interpolation and slope for typical non-enabled channels

Disabled 1, Hold=30, Fit Intercept vs 18*Chip+Chan
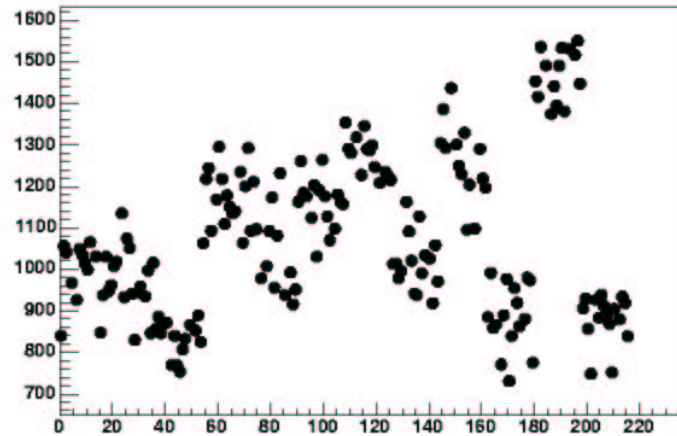
Disabled 1, Hold=30, Fit Slope vs 18*Chip+Chan
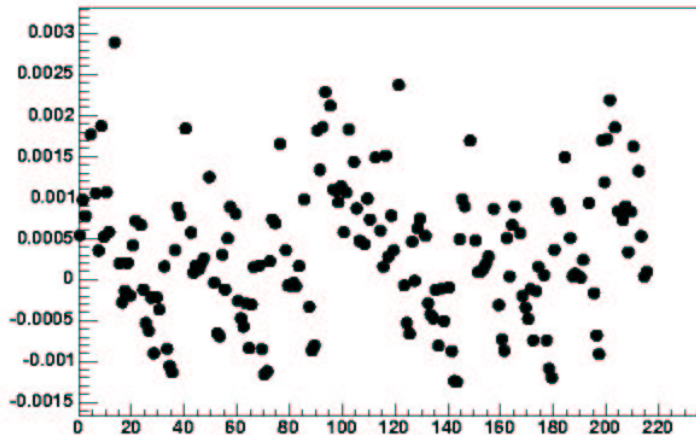
- Intercepts spread similarly
- Crosstalk below 0.5%
  - Greatly improved since VFE PCB preproduction boards (below)

Disabled 2, Hold=35, Fit Slope vs 18*Chip+Chan
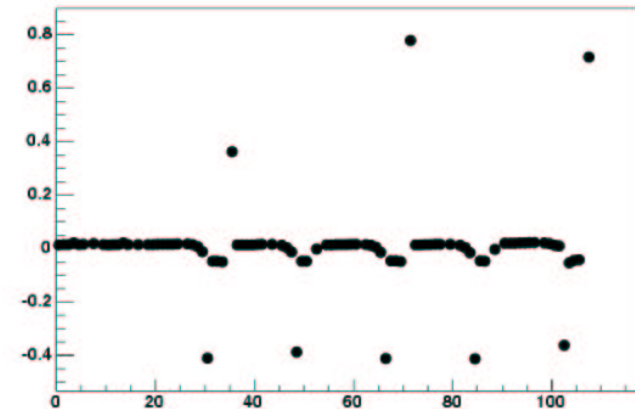
Outstanding issues with hardware

- Need to release the other seven PCBs for assembly
  - Review of performance soon; date not yet fixed
    - So far, no systematic assembly problems found
  - Production (and prototype) boards can run 32 full VFE PCBs, 20 layers
    - No urgent need for seven boards before end of Jan
    - Assembly will be ~ one month (slow to keep cost down)
- Latency is going to be tight
  - Measured delays
    - NIM-LVDS module and 5m cable ~35ns
    - Trigger section on CRC board ~30ns
    - Trigger input from backplane to minimum HOLD edge ~40ns
    - Cable (5m!) and level converter on VFE PCB ~45ns
  - Need trigger scintillators and discriminator to be ~30ns max
    - No further (NIM) logic needed; within CRC trigger section
  - Might be able to use shorter cable at NIM-LVDS, maybe save ~10ns
    - What does AHCAL need for bulk of cables? Could we swap?

Outstanding issues with firmware

- Several bugs still to be fixed
  - Reliability of event data capture in memory not perfect
    - Trigger is asynchronous; maybe need to be more careful in using it?
  - SER003/FE0 can stop functioning randomly within run
    - Temperature? Need to correlate with on-board temperature readout
- Only 512 events storable and 2MBytes of memory usable for a spill
  - Highest two address lines of memory not yet activated
- Trigger-related data not put into memory
  - Must be read for every event, i.e. during spill, through serial I/O (slow) path
  - Limits event rate; could be unread if data not considered essential initially
  - Not intrinsic to board; longer term aim to divert data into memory
- Period of clock for counter setting sample-and-hold delay not negligible
  - Currently 6.25ns; try for factor of two reduction
- Need timeout for trigger busy to get above ~ 100Hz
  - Currently reset by software only

DAQ (and other!) software

- Not analysis or Mokka, but still many aspects to this
  - Raw data format
  - Calibration, alignment, readout-to-physical mapping
  - Book-keeping for all these quantities
  - Production of LCIO analysis files
  - Slow control and readout
  - Integration with simulated data path
- Several of these not yet fixed
  - Some are getting urgent
- Those I would consider under control (for ECAL at least)
  - Raw data format – well defined (albeit dependent on firmware)
  - Calibration, etc - data handling set up
  - LCIO production software - exists; works for simulation also
- The others should be discussed here…
  - Need to think of ECAL+HCAL+tracking+PID+trigger as one system

Raw data format

- Whole online system works through use of records
  - Generalisation of events; mark logical transitions (runStart, etc.)
- Each record is very simple, with a header and trailing data
  - Header stores date/time (to µs) and record type; 12 bytes overhead
- In principle, trailing data format could be completely flexible
  - In practise, need to impose structure
- Sub-records; objects of classes known to DAQ system
  - Classes must be "flat"; no virtual functions/pointers, but can be extendable
  - Register class with DAQ to get unique integer
  - Allows type-safe storage and access to objects within record; every sub-record has integer written into it; 4 bytes overhead
  - Can insert objects to records, but can never remove them
  - "Schema migration" (i.e. non-backwards compatible changes to the classes) allowed through matching DAQ integer subfield and version number in class
- All results shown previously were produced using this system

Calibration, alignment, mapping for ECAL

- Online people must define readout-physical mapping for each run
  - Purely depends on cabling of VFE PCBs to boards; set once per run for ever
  - Class to handle mapping data exists
- Calibration and alignment more complex
  - Will need (many?) iterations on both to get optimal values
  - Will need to use raw data; who is ready to start studying this in detail?
  - First approximation for ECAL calibration is Ecole Polytechnique cosmic calibrations (see Jean-Claude's talk) but should be cross-checked from data
  - Classes to handle calibration and alignment data exist
  - Calibration allows for quadratic term also
- Very simple scheme currently used for book-keeping
  - One (flat) file of each type for every run
  - Using soft links to not have literal duplications of data
  - Not sufficient long term (?); does a database scheme for this exist already?
  - Will need to retrofit the data for these into the database system

LCIO conversion
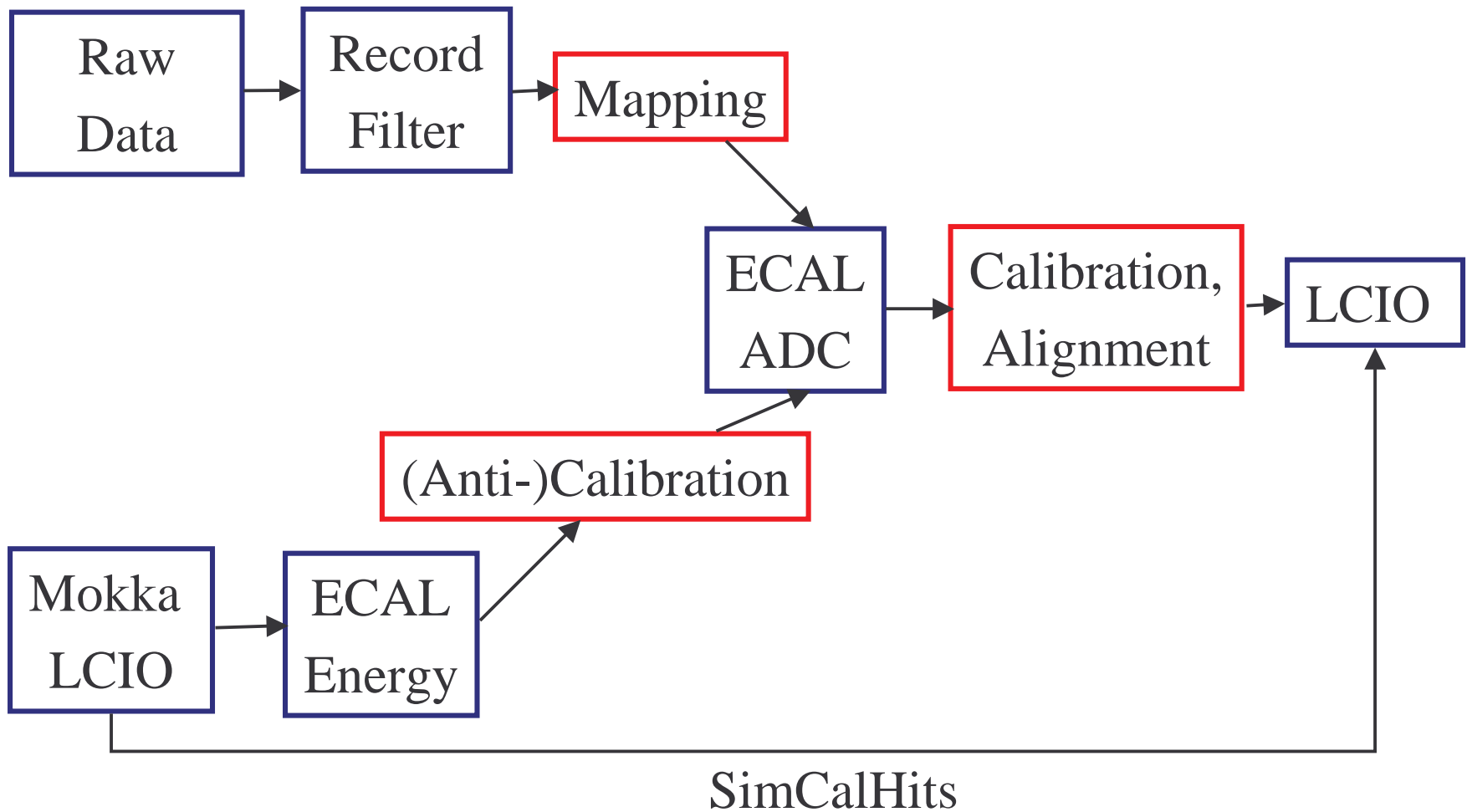
LCIO conversion (cont)

- Most analyses will use LCIO, not raw data
  - Need to convert raw ADC values to LCIO CalHit objects ("reconstruction")
    - Requires calibration, alignment and mapping to produce usable LCIO hits
  - Studies of CPU load for reconstruction only just starting; need Grid?

- Simulation must also be produced in LCIO format
  - Currently done in Mokka for truth information, SimCalHits
  - Need reconstructed version, CalHits, after simulating effects of noise, digitisation and threshold on all channels
    - Requires anti-calibration (to get ADC values) and calibration (to get back)
    - Requires knowing alignment used in simulation

- Code to do both exists
  - ADC to LCIO CalHits step is common to both; uses same class
  - Book-keeping does not exist; is done by hand at present
  - What is the solution for simulation alignment issue?
    - Write x,y,z of each pad (wafer?) into LCIO file?

Slow control and readout

- Two views
  - Set up and read out completely separate from DAQ readout

  OR

  - Integrate with DAQ system so slow readout appears in raw data records

- Former is much easier to implement
  - Can use separate path, active at all times, don't care about DAQ running
  - But CRC board voltage and temperature monitoring requires VME access so would disrupt data readout during run or would need buffering

- Latter is much easier to use
  - E.g. study/correct temperature dependence of pedestal; temperature data would be integrated into raw data files
  - Slow readout only outside spills after data flushed out; no impact on running
  - Would need DAQ system to swap into slow control/readout mode when "idle" between runs; build up complete time dependence

- Decision needed soon!

Outstanding issues on DAQ software

- Readout speed target of 100Hz ~ 3MBytes/s from VME
  - Extrapolate to full ECAL with current software, expect around ~ 50Hz
  - Currently CPU bound; improvements possible
    - E.g. inlining, preallocating memory, threading, etc; also faster PC!

- ECAL+AHCAL; how to read both at once?
  - Put all boards into one crate? Simple extension of current system
    - Maybe limited by VME bandwidth on one backplane
  - Two PCs? Coordinate using hardware trigger, software sockets
    - Doubles VME bandwidth
    - Record structure is designed to be used with sockets; software exists
    - Can be merged easily; sub-records can be appended to existing record
  - One PC with two PCI/VME interfaces? Hardware trigger, no sockets needed
    - Looks very like current system for software
    - No experience of realistic performance of multi-PCI card data readout; may be close to sum of VME bandwidth for two crates
  - Nice to use same system for ECAL+DHCAL; too many unknowns?

Future plans

- Boards are working; want to move to data taking!
- Move to Ecole Polytechnique to start cosmics runs with up to ten layers tomorrow
- Move to DESY for first beam test with central wafers for ten layers in Jan
- Add remaining boards and VFE PCBs; expand to full ECAL of thirty layers by Jun
- Buy faster, multi-CPU DAQ PC and disk array (>1TByte) before Mar; ~ 10-15k euros (?) remaining in budget
- AHCAL producing seven CRC boards for their use around Easter; can borrow an ECAL prototype from Jan
- Will help set up duplicate system at DESY to test these
- Integrate systems for ECAL+AHCAL operation later in 2005