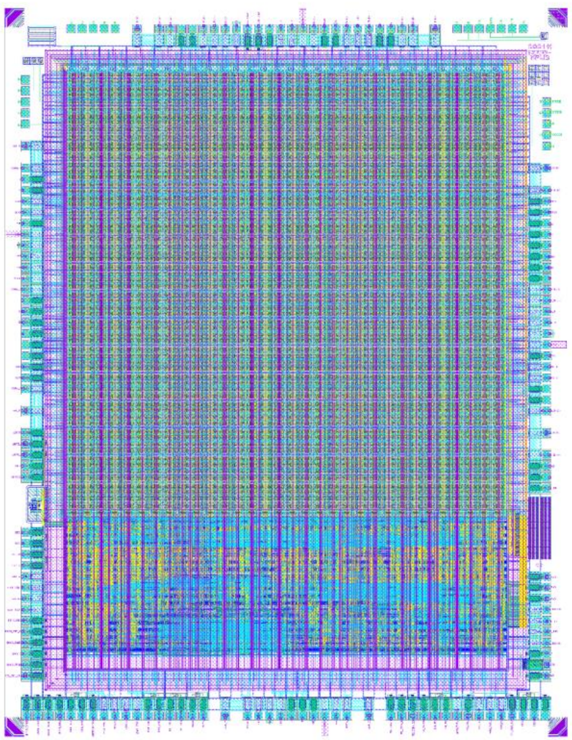


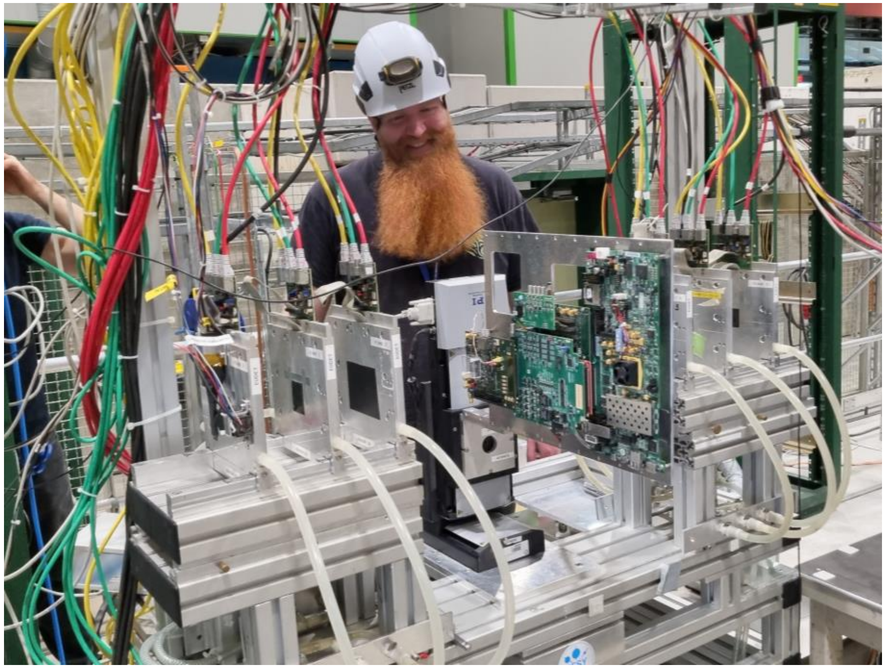
Testbeam Analysis

Patrick Sieberer, on behalf of the RD50 CMOS working group

REFERENCE DETECTORS

AIDA-Telescope

- 6 mimosa26 planes
  - Only working ones (end of lifetime)
  - 2 in front of MPW3
  - 3 at the back of MPW3
- 3D reference tracks reconstructed
- Event definition: M26 record data for 230us, roughly +/- 100 us around the trigger



- AIDA TLU for Trigger handling + Timestamping

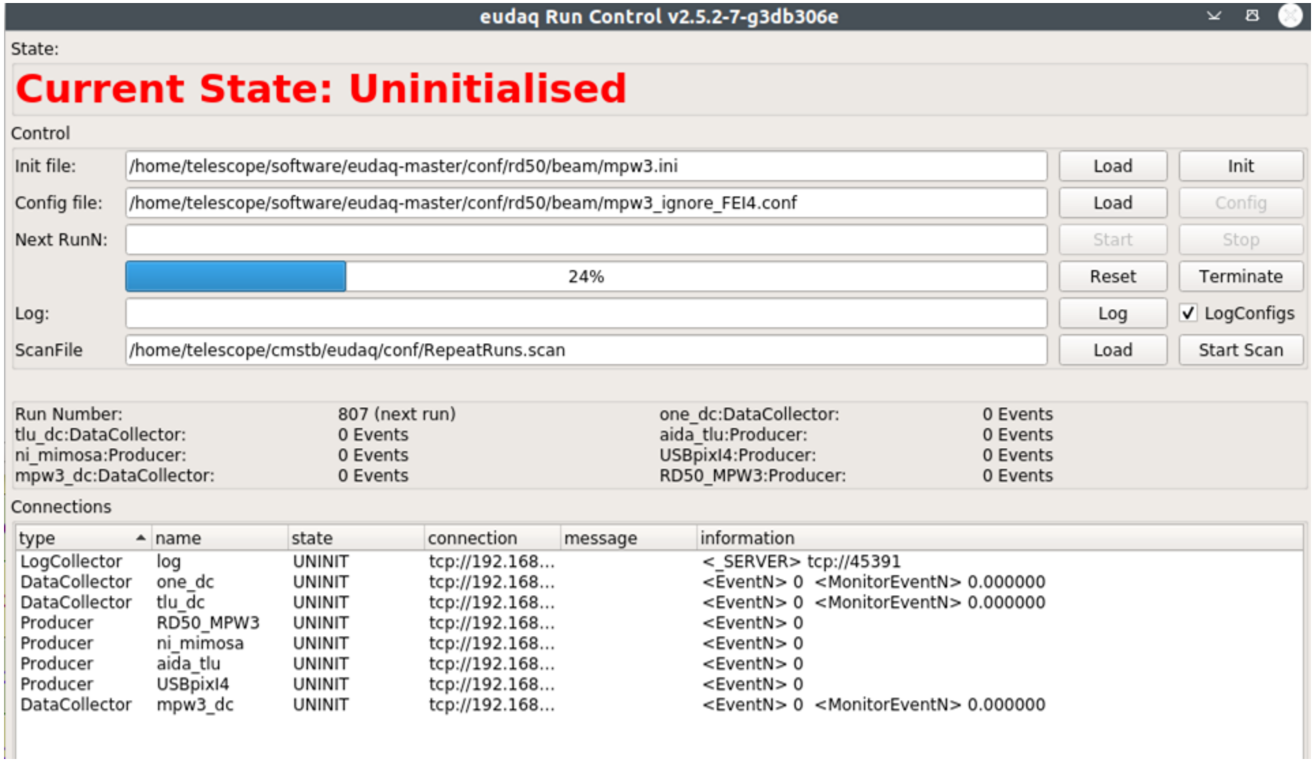
Timing Layers

- We tried an FEI4 and Timepix3 plane as timing reference layer
- FEI4 setup by CERN, worked with some issues
  - Crashes every ~30min (know bug, but not to us at the beginning) -> Huge data loss for 1 night
    - Solution: Sam scripted some magic to automatically Stop, configure and Start a run again
  - Overwrites data from telescopes -> Data loss for ~2 days (as we didn‘t check this)
- Timepix3 setup by Uwe
  - We did not manage to get it working due to more serious issues with other detectors

- Why do we need timing?
  - We expect around 50 tracks per event in the mimosa26 planes -> Hard to find the correct track with 3D tracking only
    - 4D tracking let‘s us easily distinguish between them
  - Problem not known to us before, so we were not prepared

DATA TAKING

EUDAQ2



State:

Current State: Uninitialised

Control

Init file: /home/telescope/software/eudaq-master/conf/rd50/beam/mpw3.ini

Config file: /home/telescope/software/eudaq-master/conf/rd50/beam/mpw3_ignore_FEI4.conf

Next RunN:

24%

Log:

☑ LogConfigs

ScanFile: /home/telescope/cmstb/eudaq/conf/RepeatRuns.scan

Run Number:	807 (next run)	one_dc:DataCollector:	0 Events
tlu_dc:DataCollector:	0 Events	aida_tlu:Producer:	0 Events
ni_mimosa:Producer:	0 Events	USBpixI4:Producer:	0 Events
mpw3_dc:DataCollector:	0 Events	RD50_MPW3:Producer:	0 Events

Connections

type	name	state	connection	message	information
LogCollector	log	UNINIT	tcp://192.168...		<_SERVER> tcp://45391
DataCollector	one_dc	UNINIT	tcp://192.168...		<EventN> 0 <MonitorEventN> 0.000000
DataCollector	tlu_dc	UNINIT	tcp://192.168...		<EventN> 0 <MonitorEventN> 0.000000
Producer	RD50_MPW3	UNINIT	tcp://192.168...		<EventN> 0
Producer	ni_mimosa	UNINIT	tcp://192.168...		<EventN> 0
Producer	aida_tlu	UNINIT	tcp://192.168...		<EventN> 0
Producer	USBpixI4	UNINIT	tcp://192.168...		<EventN> 0
DataCollector	mpw3_dc	UNINIT	tcp://192.168...		<EventN> 0 <MonitorEventN> 0.000000

Producer: Slow Control of detectors aka „what happens if you press Init, Config, Start“

DataCollector: Stores data from detectors to disk

Monitor: Live plotting of some fraction of the data. Monitor for MPW3 available (not shown)

Data Flow

- DataCollectors for TLU, M26 and FEI4 store data to external HDD (on CERN PC)
  - rsync command to copy to EOS (CERN Fileserver)
- MPW3 stores to internal disk of (our) NUC
  - Automatic CERNbox sync to EOS
- This way, we store each piece of data twice
  - Automatic backup ;)

Timing MPW3

- Due to low data-rate: hits sent out immediately from MPW3
- Firmware buffers them unteil a certain amount of data is stored before sending to as a UDP Jumbo frame
- Causes issue with calculating global timestamp (TS): too long waiting time, we do not see all TS overflows
  - Solution: Attach a 64 bit TS counter to UDP packet -> Change of firmware during testbeam!
    - Did cost us 3 days to find the issue

Other MPW3 issues

- We did not see a reset of all timestamp (in firmware + ASIC) every time
  - Crucial to know „when time starts“ = t0 for synchronization
- Problem: We slowed down the TLU clock, but this does not slow down the whole interface (which we expected) -> Firmware didn‘t capture every (still fast) t0 signal anymore
  - Solution: Another fix in the firmware, during the testbeam ;)
  - Did cost us another ~1 day to find out
- We did not have an online quality monitor
  - Lack of time
  - Could not verify with time synchronization actually works, so we had no clue during the testbeam if stuff is good or not.
- All together: Just 1 day + 1 night of useful data…

ANALYSIS

Corryvreckan

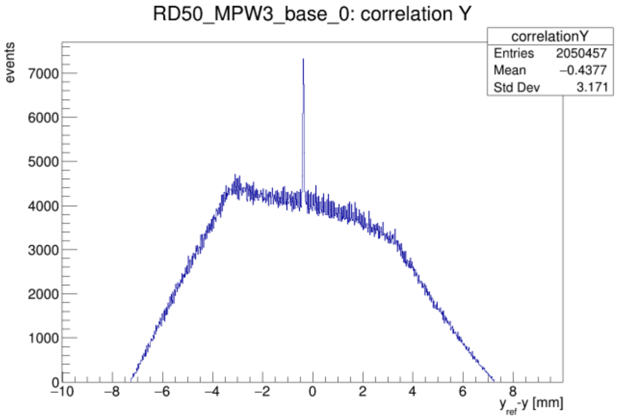
- No Tutorial here, but have a look at https://project-corryvreckan.web.cern.ch/project-corryvreckan/
  - Tutorial, workshop, detailed documentation available
- Eventloading and Eventdefinition is working
  - Uwe managed to setup M26 part
  - Bojan managed to load FEI4 Files
  - Bernhard managed to load MPW3
    - MPW3 needs preprocessing -> Next slide
- Check of synchronization: Define a (M26) plane as reference and plot correlations (x-x_ref etc…)
  - Did work for M26 planes immediately during testbeam
  - Did not work for FEI4 and MPW3
    - Bojan fixed FEI4 shortly after testbeam, looks fine

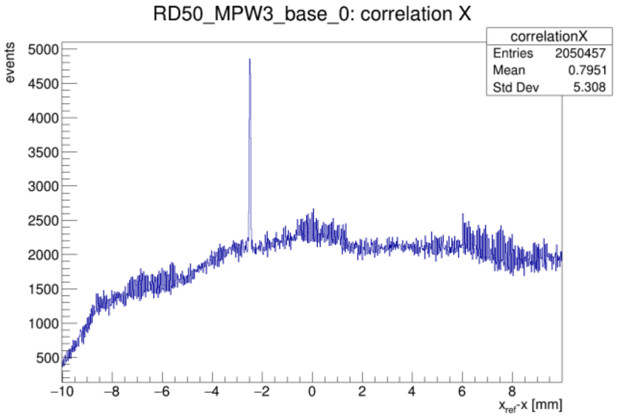
MPW3 Analysis

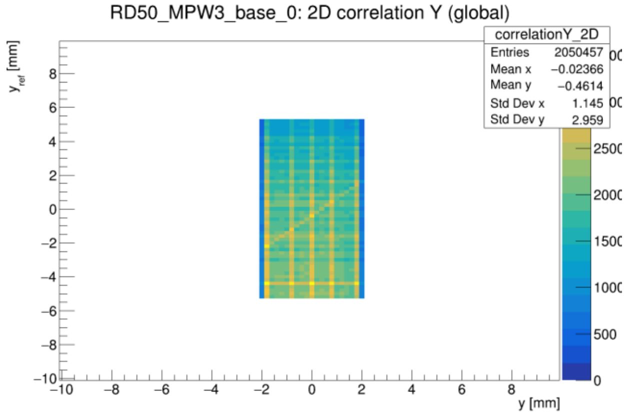
- MPW3 need a preprocessing step before loading it to Corryvreckan
  - Assigns global TS to the pixel hits
  - Done within EUDAQ framework
  - *raw event* = Data stored to disk (online, during data taking)
    - 0s in UDP stream are thrown away
  - *preprocessed event* = including global timestamp calculation after preprocessing
  - Both types are stored in EUDAQ binary format
- Calculating a global TS is very difficult (have a look at backup)
  - We did not manage during testbeam
  - We did not see correlations…

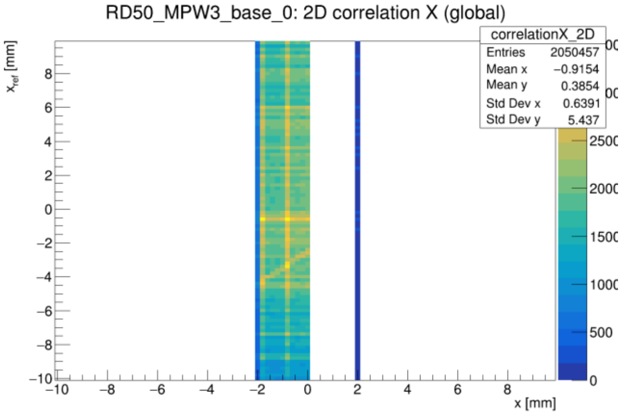
Correlations !!!

- After 1 week of analysis, ….









- *Very preliminary, needs some checks*
- *we do not fully understand why we see them (we apply a „random“ time shift)*
  - *We do not know if this is constant for every run*

BEAM HAZARD
EVACUATE AREA
DANGER FAISCEAU
EVACUEZ LA ZONE
Evacuation immediately
Evacuation immédiate

WHERE TO GO?

RD50-MPW3

- Shall we design a new Chipboard?
- Shall we test the piggy board?

RD50-MPW4

Another MPW run is a reasonable option. Current design is not yet ready for engineering run.

- Fixing current issues of RD50-MPW3:
  - Interface between matrix and periphery
    - We know the problem and have a solution
  - Easy generation of global timestamp
    - We know the problem and have a solution
  - High noise in lower half of matrix
    - We have a hint where the problem is
- Adding more features for the future
  - Depends on where we want to go. Telescope? Future detectors (FCC)?
  - Thinning, backside processing
  - Scaling of the sensor (Does require detailed design study to identify limits)
- Can we already start designing? Who would be in?
  - I‘m not available for the digital design anymore
    - (unless somebody has money for a postdoc ;))
- Need strategy for transition from RD50 to ECFA DRD scheme from 2024 onwards

BACKUP

Event Definition

- M26 Event Definition
- 230us long event frame
  - Between ~ +/-100us before trigger recorded in TLU
- Event matching (for correlations):
  - Every other detector, which has a timestamp in this window in correlated with every event from M26 reference plane
    - Trigger number irrelevant

MPW3 Readout

- Assuming a low trigger-rate of a few 100Hz to 1 kHz
- Event read out instantly: SOF + 1 Pixel-event + EOF
  - No buffer in ASIC used
- Time information from chip:
  - 8 bit Pixel event
  - 8 bit in SOF (written when SOF received by FPGA)
  - 8 bit in EOF (written when EOF received by FPGA)
  - Due to instant readout: SOF and EOF bit are usually the same or differ by 1 if you're unlucky
  - Give 16bit at 20MHz, thus an overflow every 2^16*50ns = 3.276.800 ns or every 3.2ms

Firmware readout

- UDP buffer of depth 4096 implemented
  - 1 Word = 1 Pixel-event or 1 SOF or 1 EOF (32 bits each)
  - Due to low event rate/instant readout the buffer looks like this: SOF, 1 PixEv, EOF, SOF, 2 PixEv, EOF ,…. (Thus, 7 of the FIFO spots are written here)
- Fifo read-out when either half full (2048) or after ~1ms (Query by software)
  - Usually: **1ms** is the case.
- UDP packet is always 2048 words long -> Rest filled with 0
  - Usually, a lot of 0, only few frames.
  - Time needed to send 1 UDP packet: 2048 * 32 * 1ns (=1GHz) = **66.048us**
  - At the end of a UDP packet, a 64bit counter (at 20MHz) is sent
- 64bit counter, SOF + EOF counter and ASIC are reset at t0

Software readout

- Online: A full UDP packet is loaded into the buffer
  - Packet counter (to check for missing packets) checked
  - Discard 0 events
  - Store blocks between SOF and EOF as *raw events*, attach 64bit counter to each *raw event*
- **Offline (Preprocessing):**
  - **Take 8 bit from pixel event + 8 bit from SOF (3.2ms OF)**
  - **Check for SOF Overflow**
  - **Replace last 16 bits of the 64 bit counter (queried every 1 ms) to generate *preprocessed Event*.**
    - **If SOF Overflow x times subtract x times 3.2ms to 64 Bit counter (last SOF closest to time when 64 bit counter is written)**
- Analysis:
  - Possibility to subtract us from MPW3 event