ITS DCS Status

Paolo Martinengo
CERN

12th ALICE ITS Upgrade, MFT and O2 Asian Workshop
인하대학교, 인천, 대한민국

Outline

- Detector control chain
- ALPIDE monitor
- Detector calibration
- FLP/CRU/RU/Staves topology
- Outlook

Reminder & acknowledgments

- DCS meeting embedded in WP10 meeting (Tuesday, 3:00 PM Geneva time), quite useful and productive discussions

- Presentation by Peter Chochula: https://indico.cern.ch/event/766029/

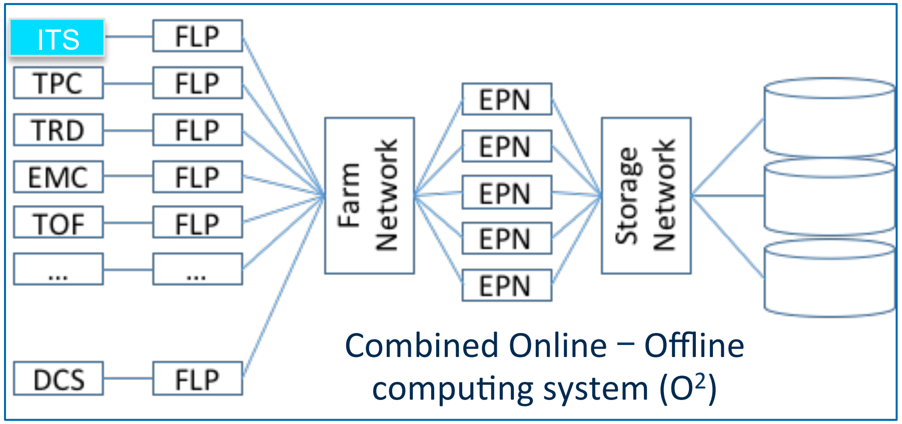
- Presentation by Markus Keil: https://indico.cern.ch/event/767536/

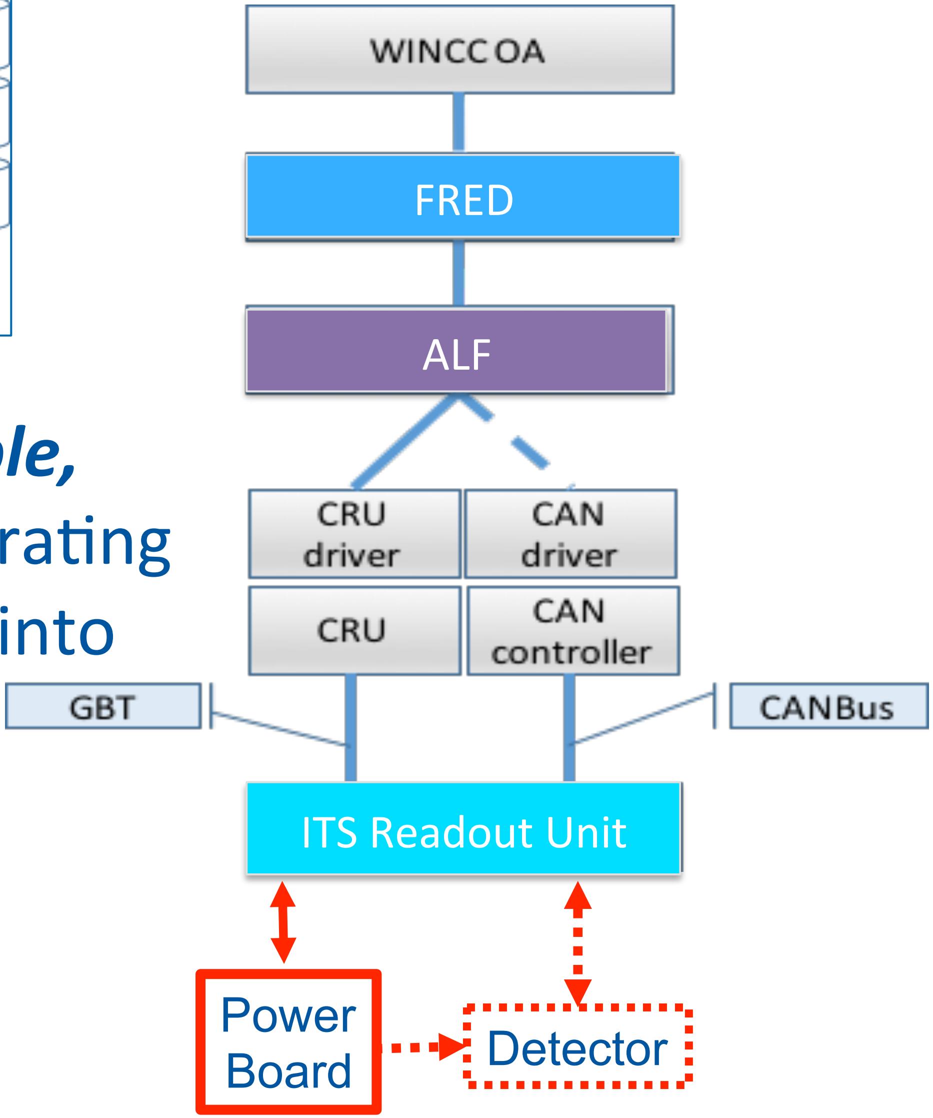
- Presentation by Johan Alme: https://indico.cern.ch/event/772143/

- Presentation by Simon Voigt Nesbø : https://indico.cern.ch/event/772143/

- Presentation by Sylvain Chapeland: https://indico.cern.ch/event/751993/

Detector Control chain



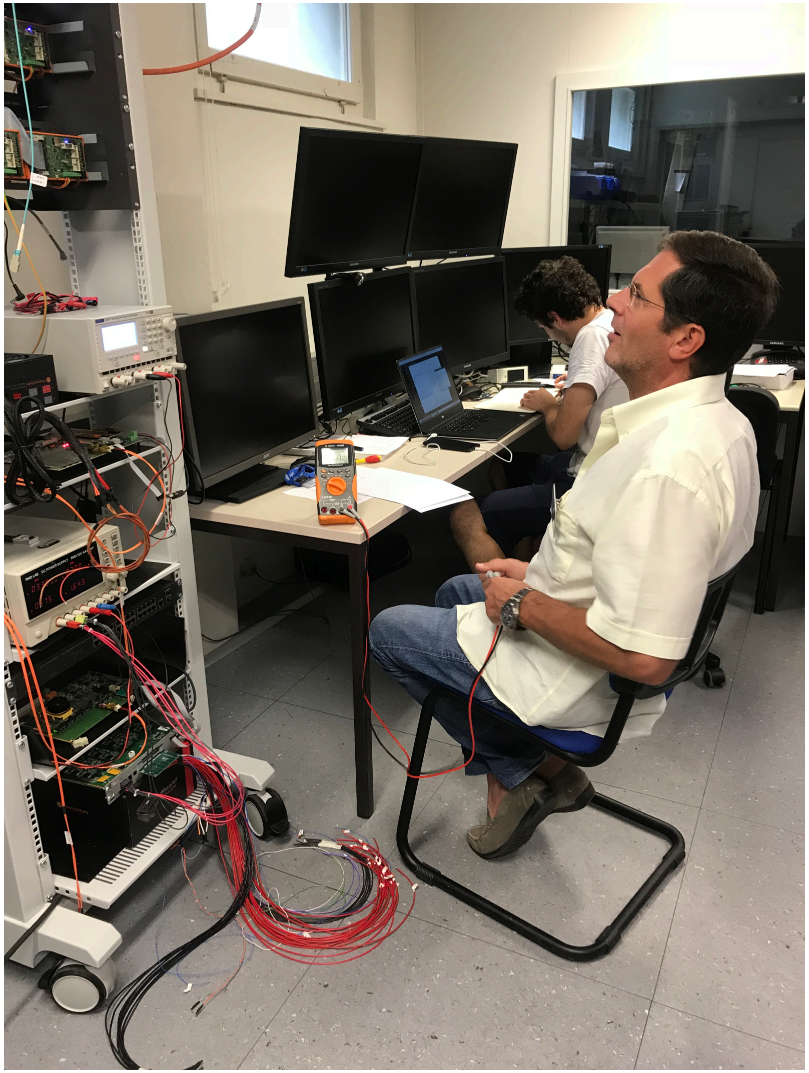


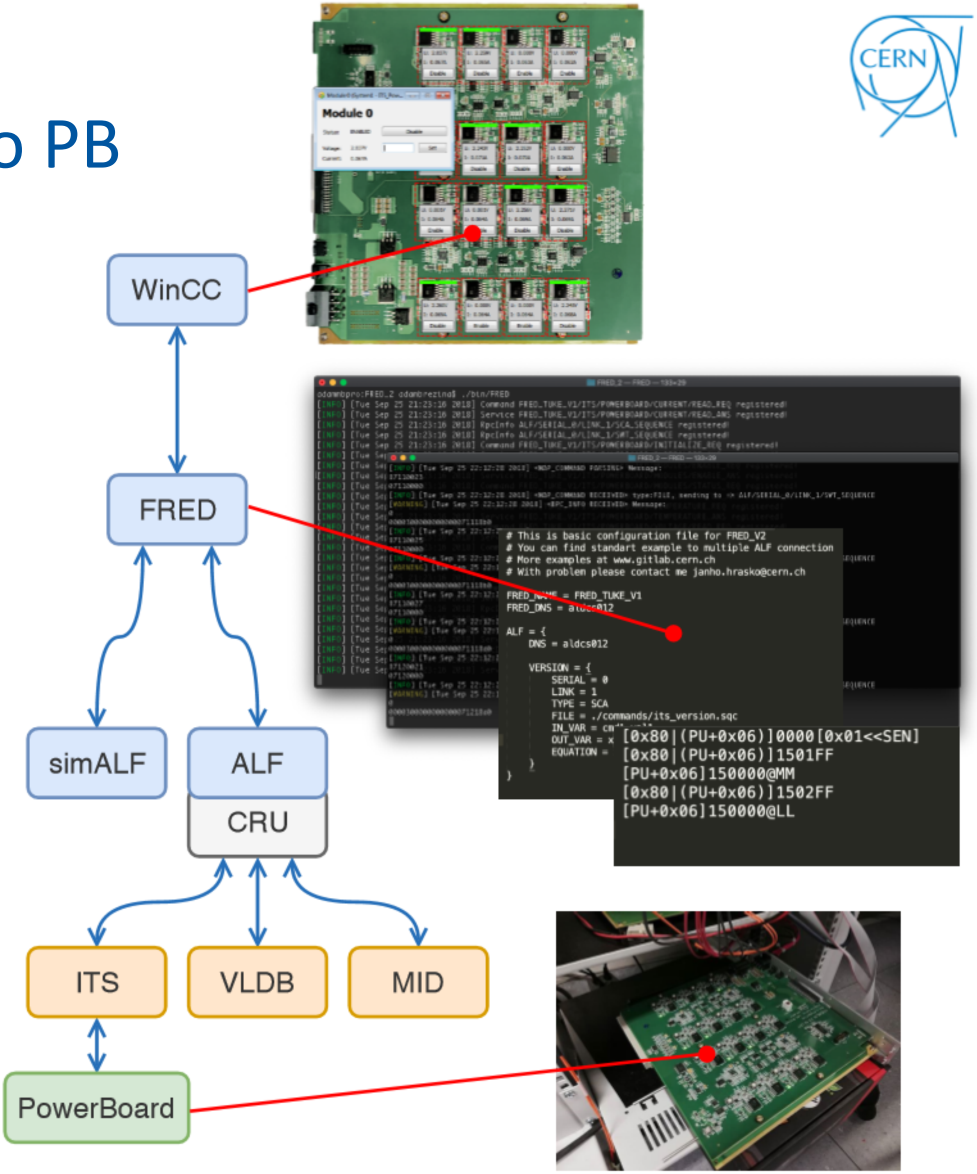
- ALF FRED is*: **universal, flexible, scalable*** framework for integrating the new detector front-ends into the DCS

Peter

ITS test bench

- Full chain from WINCC to PB





```
adambpro:FRED_2 adambrezina$ ./bin/FRED
[INFO] [Tue Sep 25 21:23:16 2018] Command FRED_TUKE_V1/ITS/POWERBOARD/CURRENT/READ_REQ registered!
[INFO] [Tue Sep 25 21:23:16 2018] Service FRED_TUKE_V1/ITS/POWERBOARD/CURRENT/READ_ANS registered!
[INFO] [Tue Sep 25 21:23:16 2018] RpcInfo ALF/SERIAL_0/LINK_1/SCA_SEQUENCE registered!
[INFO] [Tue Sep 25 21:23:16 2018] RpcInfo ALF/SERIAL_0/LINK_1/SWT_SEQUENCE registered!
[INFO] [Tue Sep 25 21:23:16 2018] Command FRED_TUKE_V1/ITS/POWERBOARD/INITIALIZE_REQ registered!
```

```
[INFO] [Tue Sep 25 22:12:28 2018] <WDP_COMMAND PARSING> Message:
07110023
07110000
[INFO] [Tue Sep 25 22:12:28 2018] <WDP_COMMAND RECEIVED> type:FILE, sending to -> ALF/SERIAL_0/LINK_1/SWT_SEQUENCE
[WARNING] [Tue Sep 25 22:12:28 2018] <RPC_INFO RECEIVED> Message:
0
000010000000000007111bb0
```

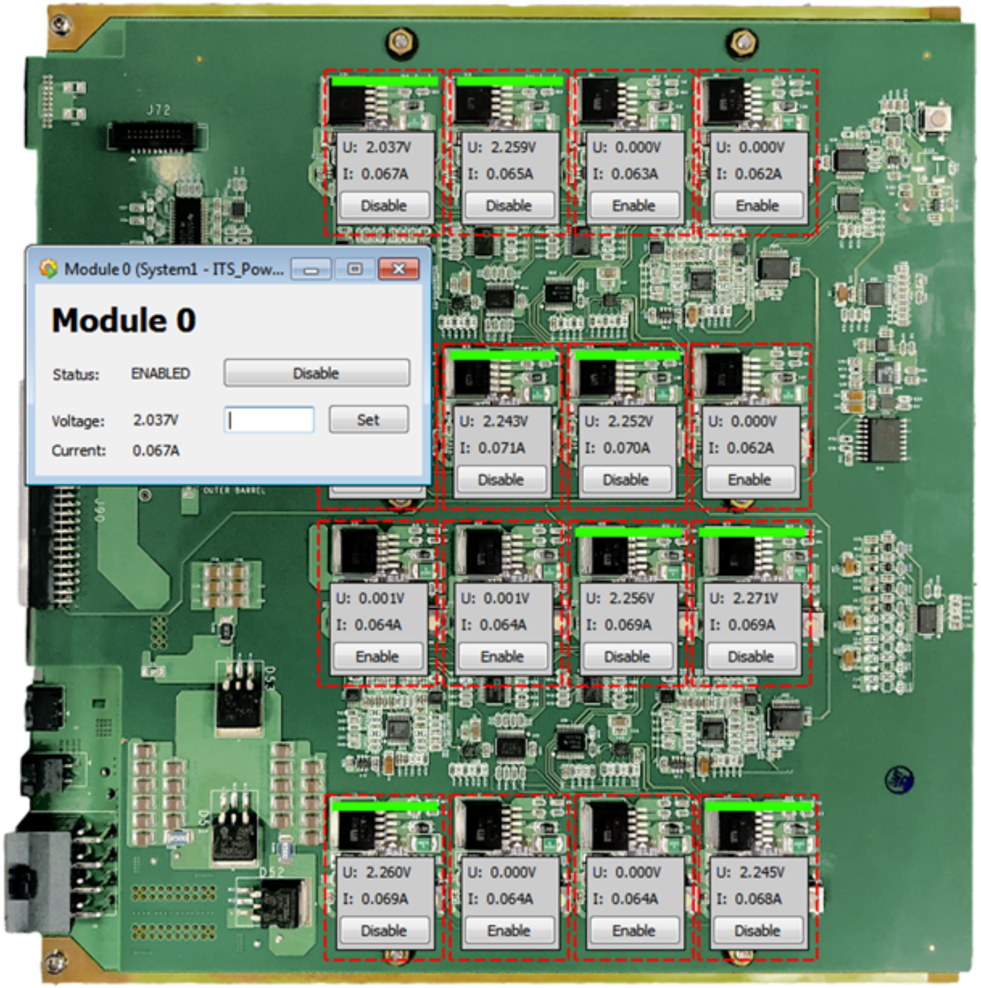
```
# This is basic configuration file for FRED_V2
# You can find standart example to multiple ALF connection
# More examples at www.gitlab.cern.ch
# With problem please contact me janho.hrasko@cern.ch

FRED_NAME = FRED_TUKE_V1
FRED_DNS = aldcs012

ALF = {
    DNS = aldcs012

    VERSION = {
        SERIAL = 0
        LINK = 1
        TYPE = SCA
        FILE = ./commands/its_version.sqc
        IN_VAR = cm
        OUT_VAR = x
        EQUATION =
    }
}
```

```
[0x80|(PU+0x06)]0000[0x01<<SEN]
[0x80|(PU+0x06)]1501FF
[PU+0x06]150000@MM
[0x80|(PU+0x06)]1502FF
[PU+0x06]150000@LL
```

Figure 1: Actual State of ALFRED server

Powerboard WINCC OA panel

- Control chain deployed also in 167
- Performance under test
- First version of PB control panel
- So far working with prototype version, production one already available (including manual)



Module 0 (System1 - ITS_Pow...

Module 0

Status: ENABLED

Voltage: 2.037V

Current: 0.067A

U: 2.037V I: 0.067A

U: 2.259V I: 0.065A

U: 0.000V I: 0.063A

U: 0.000V I: 0.062A

U: 2.243V I: 0.071A

U: 2.252V I: 0.070A

U: 0.000V I: 0.062A

U: 0.001V I: 0.064A

U: 0.001V I: 0.064A

U: 2.256V I: 0.069A

U: 2.271V I: 0.069A

U: 2.260V I: 0.069A

U: 0.000V I: 0.064A

U: 0.000V I: 0.064A

U: 2.245V I: 0.068A

ALPIDE monitor implementation

Johan & Simon

ALPIDE Registers of Interest

- 77 register in the ALPIDE in total
  - 22 ADC value registers
  - The rest is mostly control, and some status registers
- The most interesting data to monitor is probably:
  - Temperature
  - AVDD
  - DVDD
  - DVSS
  - Single Event Upset (SEU) counter
- But other registers may be interesting as well
- We need a flexible solution that can cater to future needs

Alpide Monitor Preliminary Specifications

- Readout of values on Alpide control bus, without colliding with trigger signals
  - Abort gap trigger input in triggered mode
  - Deterministic timing for Alpide control bus access
- Monitored values available via wishbone bus
  - Possible to adapt design for «push scheme» of Alpide monitor data to DCS, on CAN/SWT
- 1 Hz refresh rate of ADC values for all chips
- Configurable monitoring
  - Monitoring of all ADC values
  - Monitoring of user specified registers

Alpide Monitor Modes of Operation

- «Idle mode»
  - When there is no beam
  - Monitoring readout at will, or periodically

- Triggered mode
  - Perform monitoring readout on abort gap trigger
  - This is the most challenging mode to implement

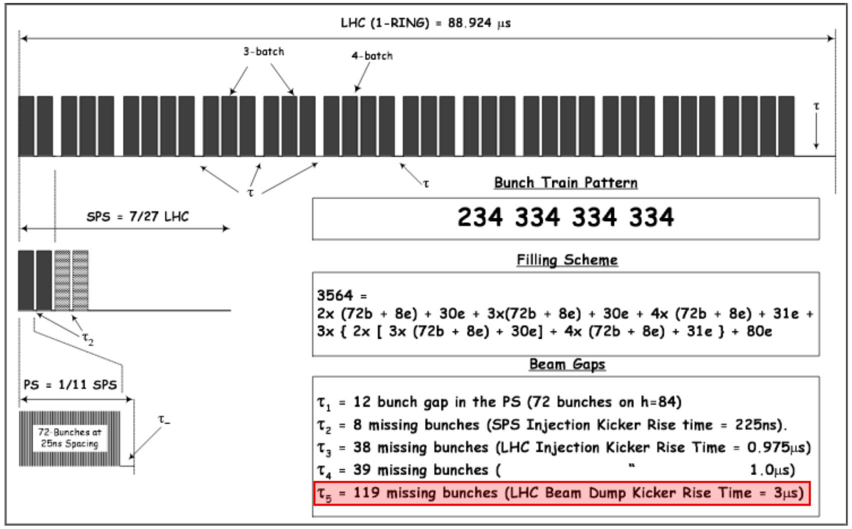
- Continuous mode
  - RU generates periodic triggers for the ALPIDEs
  - Perform monitoring readout after trigger

Abort Gap

- 119 bunch abort gap per orbit (this is the number mentioned by Paolo)
- 119 x 25 ns = 2975 ns
- This is sufficient to read out **one** register on the ALPIDE control bus (350 ns margin)

- LHC orbit is 11.2455 kHz
- There 2 abort gaps per orbit, we can read ALPIDE registers at a rate of 22.5 kHz

- There are some other gaps, but none over 2625 ns



Bunch Train Pattern

234 334 334 334

Filling Scheme

3564 =
2x (72b + 8e) + 30e + 3x(72b + 8e) + 30e + 4x (72b + 8e) + 31e +
3x { 2x [3x (72b + 8e) + 30e] + 4x (72b + 8e) + 31e } + 80e

Beam Gaps

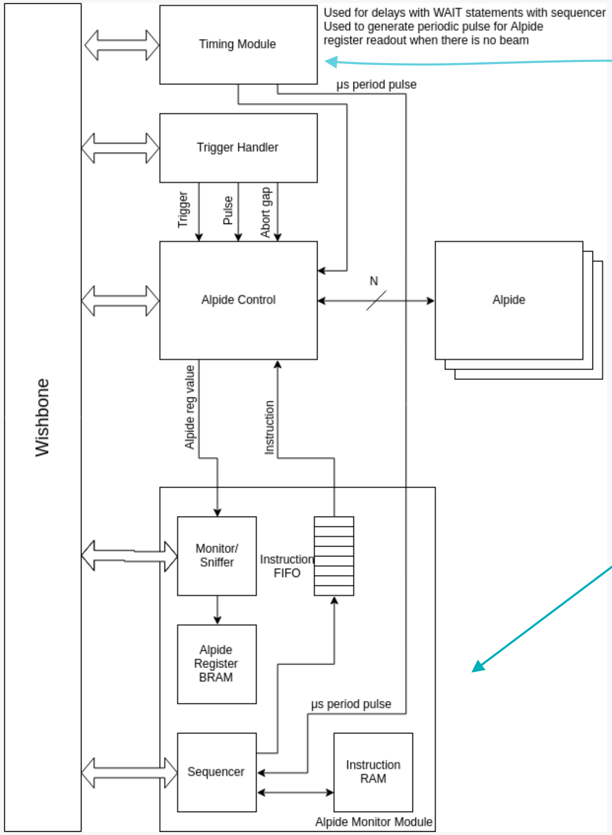
τ_1 = 12 bunch gap in the PS (72 bunches on h=84)
τ_2 = 8 missing bunches (SPS Injection Kicker Rise time = 225ns).
τ_3 = 38 missing bunches (LHC Injection Kicker Rise Time = 0.975μs)
τ_4 = 39 missing bunches (" 1.0μs)
τ_5 = 119 missing bunches (LHC Beam Dump Kicker Rise Time = 3μs)

Figure 1: Schematic of the Bunch Disposition around an LHC Ring for the 25ns Filling Scheme

LHC filling schemes (maybe old, 2003?) https://cds.cern.ch/record/691782/files/project-note-323.pdf

Alpide Monitoring Overview

Work in progress



New modules

- Modification to Alpide Control requried for broadcast write, abort gap input, and instruction input
- Modification to Trigger Handler required to accept abort gap trigger from CTP

Alpide Registers Refresh Rate

Outer barrel:

- With just one Alpide Control module:
- 22 x 3 writes + (32 reads x 7 x 7 x 4) = 6338 transactions

- Assumes that broadcast write can be performed on all links, but links must be read individually
- Assumes that ctrl link is used on every abort gap. If we wait for each conversion to complete, this number will be lower.

- Theoretical register refresh rate:
- 22.5 kHz / 6338 = ~3.5 Hz

Alpide Monitoring Overview

Work in progress

- Monitoring of Alpide Registers will be split into two submodules:
  - Sequencer
    - Has an instruction memory accessible from the wishbone bus
    - FIFO interface that connects to Alpide Control
    - Sequencer puts instructions on FIFO, Alpide Control reads and executes the instructions when it can (e.g. abort gap trigger)
  - Monitor/sniffer
    - Has a BRAM buffer with dedicated memory locations for each Alpide register, for up to 196 chips
    - BRAM buffer accessible from wishbone bus
    - Accepts data from any read transaction on Alpide CTRL bus
    - Puts the data into the BRAM according to register address and chip id
    - Completely decoupled from sequencer

Challenging but powerful and flexible

Detector calibration

Detector Calibration

Markus

Minimum set of calibration measurements:

- Digital Scan
- Threshold Tuning
- Threshold Scan
- Noise occupancy

Threshold Scan

Current Procedure:

Main loop:

- nChips configuration commands (set charge)
- 50 Pulse commands (Triggers generated on-chip)
- Readout

x 50 charge steps

x 512 rows

Full scan:

- **5M** configuration commands / stave
- **1.28M** triggers (pulse)
- Total raw data volume O(100 GB / stave), **typically 300 – 400 GB**

Requires continuous swapping/synchronization between DCS (change of configuration)
and
O2 (data readout & processing)
Can we avoid this ?

Possible Simplifications

1) Broadcast Write
   - Using broadcast write for the configuration commands
     nChips -> 4 / stave 5M -> 100k
   - Can be tested already now with new-alpide-software

2) Sequencer
   - Ideal use case for local sequencer
   - E.g. innermost loop one sequence does the scan of entire row
   - All scans could have same basic sequence with some configuration changes

```
for value = start_dac to stop_dac:
    WRITE_BC(dac_address, value)
    for i = 0 to npulse:
        SEND_OPCODE(PULSE)
        WAIT(n_wait_cycles)
```

Is it the bottleneck ?

   - Would reduce full threshold scan to 512 executions of the scan sequence
   - Each sequence would result in 2500 events, yielding the thresholds of the entire row
   - Only configuration overhead: initial configuration of chips and sequencer, stepping rows

Could ALF be an amphibian ?

- ALF eco-system is the FLP, traditionally O2 domain
- Developed in the DCS environment and considered part of it but can flourish in the O2 one as well

Sharing the control of ALF between FRED and O2 would avoid to continuously swap between DCS/O2

FLP/CRU/RU/Staves topology

Some numbers

The ALICE CRU has 24 optical inputs, 24 optical outputs
The RU has 3 optical outputs (data), 1 optical input (control)

We assume to connect 3 uplink fibers per RU, i.e. 8 RU/CRU
192 RU (48, 54, 90), 24 CRU (6, 6.75, 11.25)

1 optical link (up/down) has **3.2 Gb/s** bandwidth

	RU	CRU
IB	48	6
MB	54	6.75
OB	90	11.25
Total	192	24

192 RUs → 24 CRUs, 12 FLPs:
minimum HW to operate the ITS
(TDR scenario)

Do we have enough
computing power to clusterize the data ?

More numbers

Adam’s simulation, 20us frame duration

Layer	0	1	2	3	4	5	6
50kHz Pb-Pb:	1.86	1.11	0.76	0.51	0.40	0.52	0.48
100kHz Pb-Pb	3.47	2.06	1.41	0.81	0.59	0.69	0.61
200KHz Pb-Pb	6.59	3.89	2.63	1.43	0.99	1.03	0.89

(Gigabit/s/RU)

Strong indications that:

- Load balance desirable/needed
- Extra computing power, i.e. more FLPs (24), needed

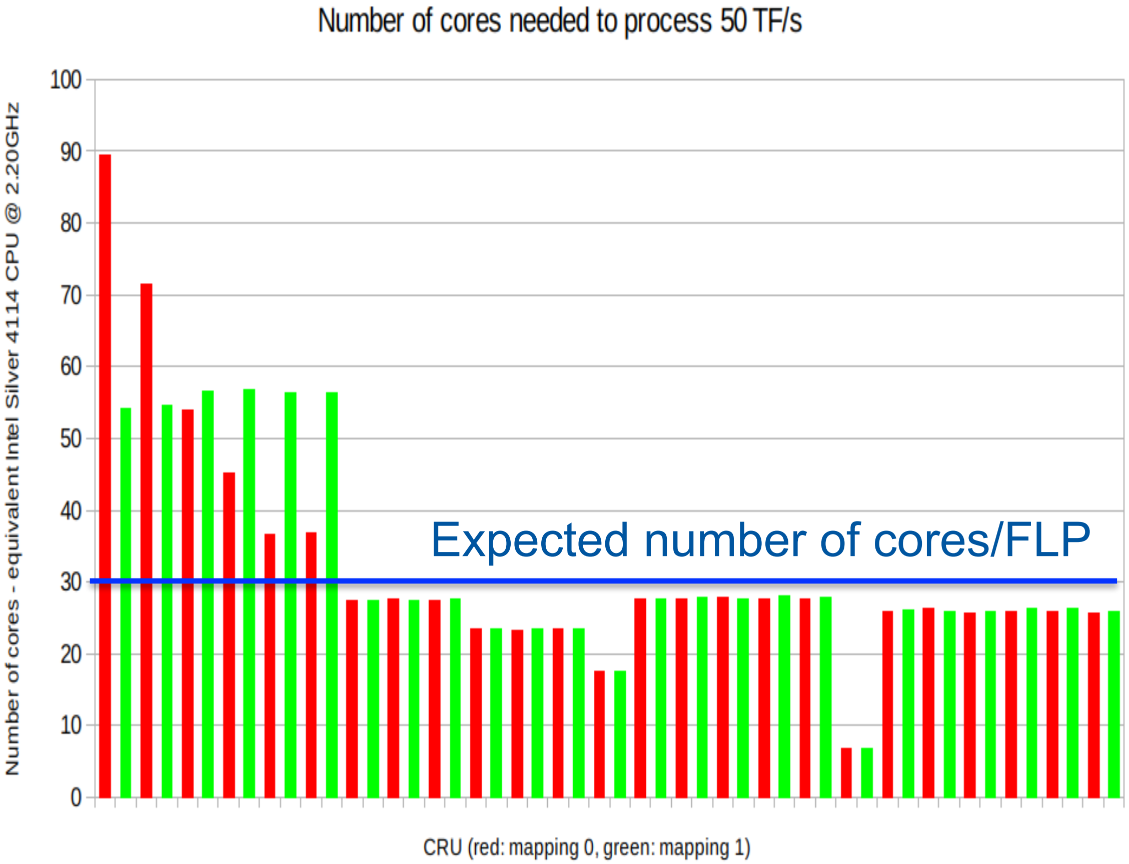
No balance

		CRU												Total STAVES (RU)
		0	1	2	3	4	5	6	7	8	9	10	11	192
IB	Layer 0	8	4											48
	Layer 1		4	8	4									
	Layer 2				4	8	8							
MB	Layer 3	8	8	8										54
	Layer 4				8	8	8	6						
OB	Layer 5	8	8	8	8	8	2							90
	Layer 6							8	8	8	8	8	8	

Mixing only in the IB

		CRU												Total STAVES (RU)
		0	1	2	3	4	5	6	7	8	9	10	11	192
IB	Layer 0	2	2	2	2	2	2							48
	Layer 1	2	2	3	3	3	3							
	Layer 2	4	4	3	3	3	3							
MB	Layer 3	8	8	8										54
	Layer 4				8	8	8	6						
OB	Layer 5	8	8	8	8	8	2							90
	Layer 6							8	8	8	8	8	8	

Sylvain



Extrapolated from measurements on a 2x10 cores Intel Silver 4114 CPU @ 2.20GHz

DELL R740 – as used for CRU readout in O2 lab

Executive summary

- Likely the FLPs will have 25 – 30 cores (50 cores machines uncommon)
- Balance by mapping helps but does not solve the problem unless we add more CRUs & FLPs (expensive)
- Balancing via network possible but does not help if the FLP has ~ 30 cores unless we add FLPs without CRUs
- Most promising solution is to perform/complete cluster finding in the EPN

Outlook

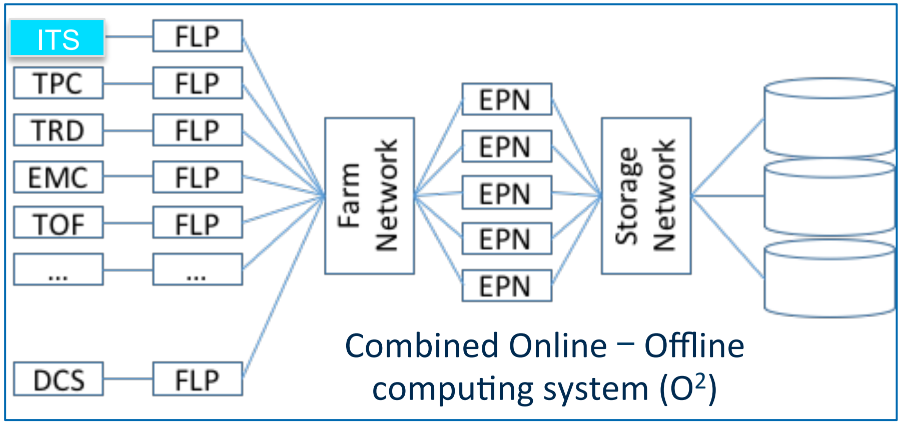
- All basic building blocks available

- Proof of principle of the full chain up to the detector
  Ready to include the detector (IB HIC) in Bergen & CERN

- Strategy to perform complex operations under definition
  (T monitoring, configuration, threshold scan etc.)
  Need experimental input (time), test bench(s) available

- Supervision/coordination layer needed to integrate DCS/O2

Back-up



Combined Online – Offline computing system (O^2)

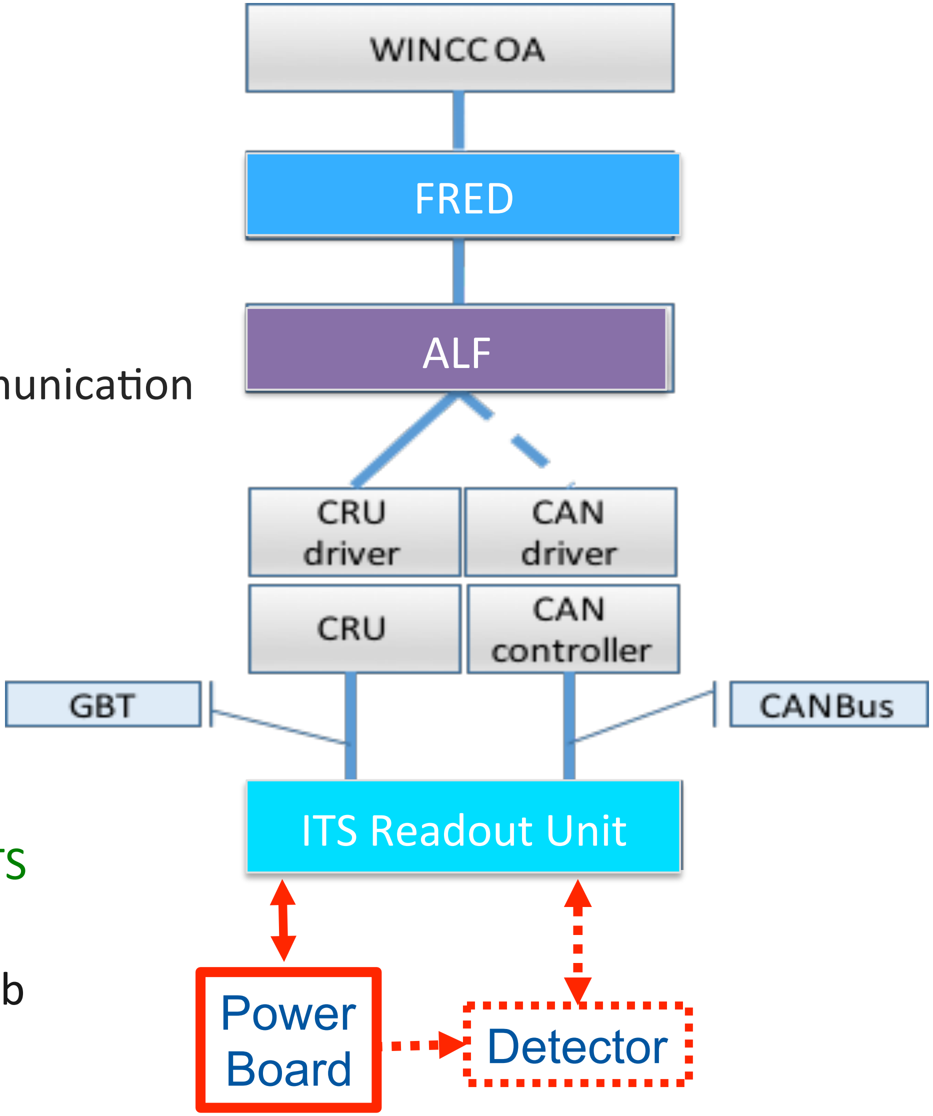
Architecture DCS

From the RU to WinCC

- **ALF** (Alice Low level Frontend)
  - General purpose for all detectors: communication interface to/from CRU (writing/reading)
- **FRED** (Front End Device)
  - Detector specific customization
  - Dedicated server
  - Commands from WinCC to ALF
  - Data from ALF to WinCC

➢ One RU board available to start working @ ITS communication protocols

  ➢ Setup under installation @ DCS group lab at CERN (Kosice)



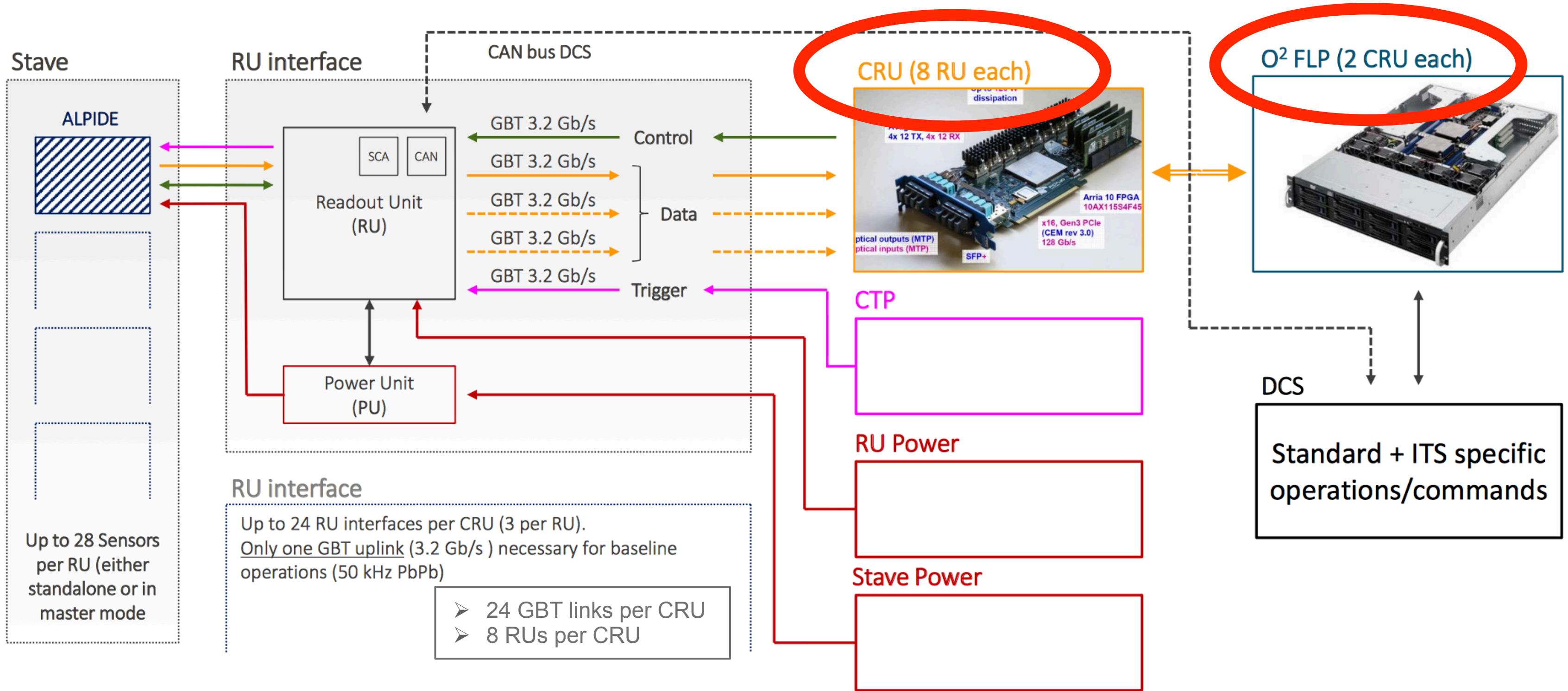
Architecture ITS

ITS readout overview

From ALPIDEs through RU to CRU …

- What we want to monitor/control?
  - Temperature, Voltages and Currents
- From?
  - ALPIDE / Power Unit / Readout Unit
- To?
  - CRU via GBT links and CAN bus (backup/emergency) → First Level Processing (FLP)

	ALPIDES	Staves	Readout Units	Power Boards
IB	9x48	48	48	24
OB	8x14x54 14x14x90	144	144	144
Total	24120	192	192	168

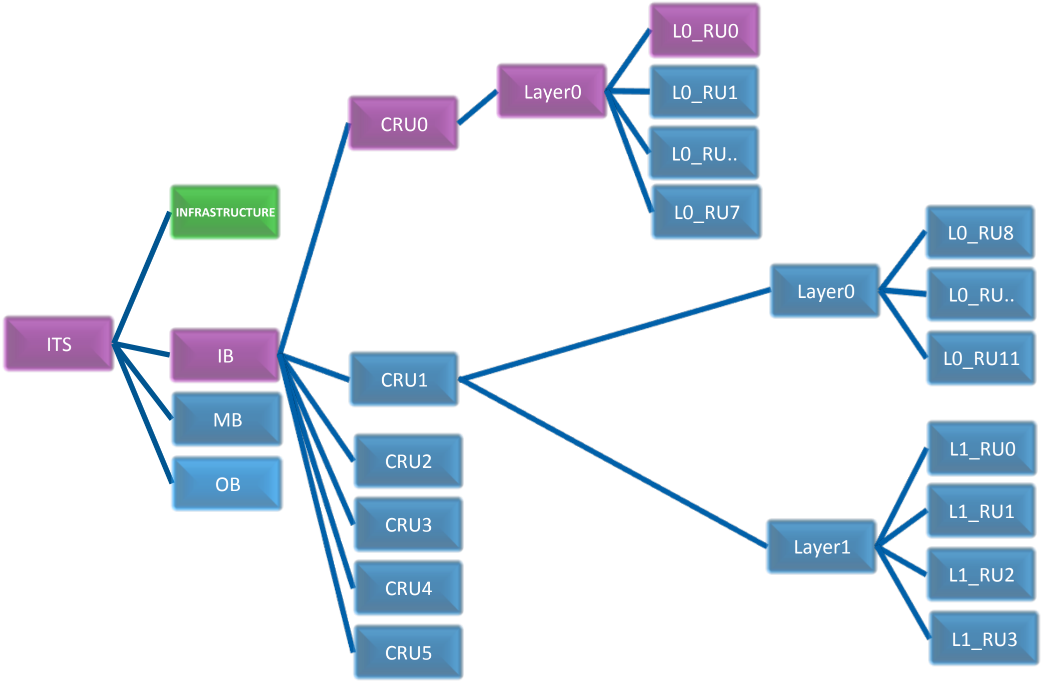


Up to 28 Sensors per RU (either standalone or in master mode

Up to 24 RU interfaces per CRU (3 per RU).
Only one GBT uplink (3.2 Gb/s) necessary for baseline operations (50 kHz PbPb)

- 24 GBT links per CRU
- 8 RUs per CRU

Detector Tree



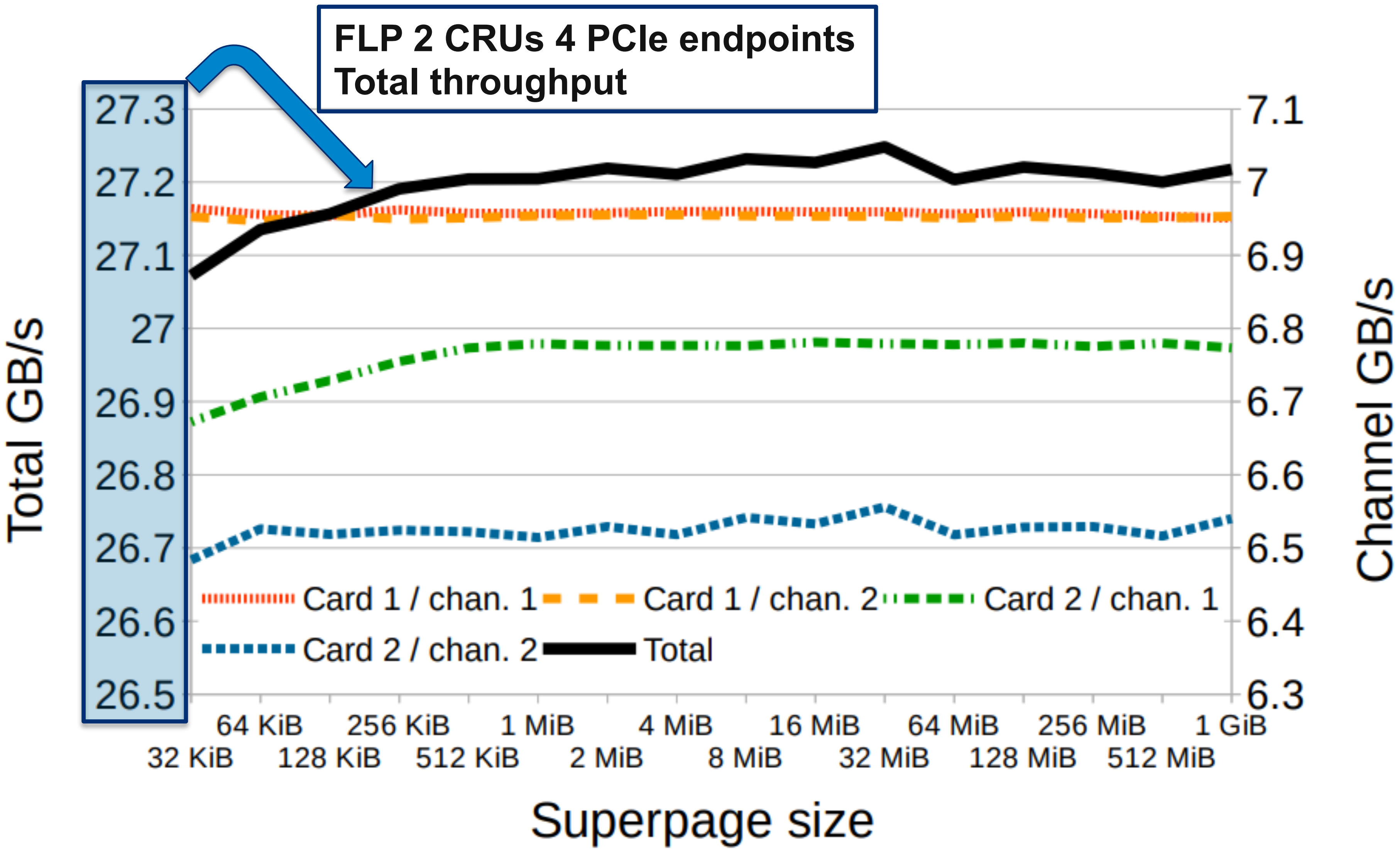
Divide per Layer/CRU
→ *Up to 8 RU/CRU*
→ RU enumeration per Layer

		CRU												Total STAVES (RU)
		0	1	2	3	4	5	6	7	8	9	10	11	192
IB	Layer 0	8	4											48
	Layer 1		4	8	4									
	Layer 2				4	8	8							
MB	Layer 3	8	8	8										54
	Layer 4				8	8	8	6						
OB	Layer 5	8	8	8	8	8	2							90
	Layer 6							8	8	8	8	8	8	

https://indico.cern.ch/event/678557/contributions/2813604/
(Filippo Costa)

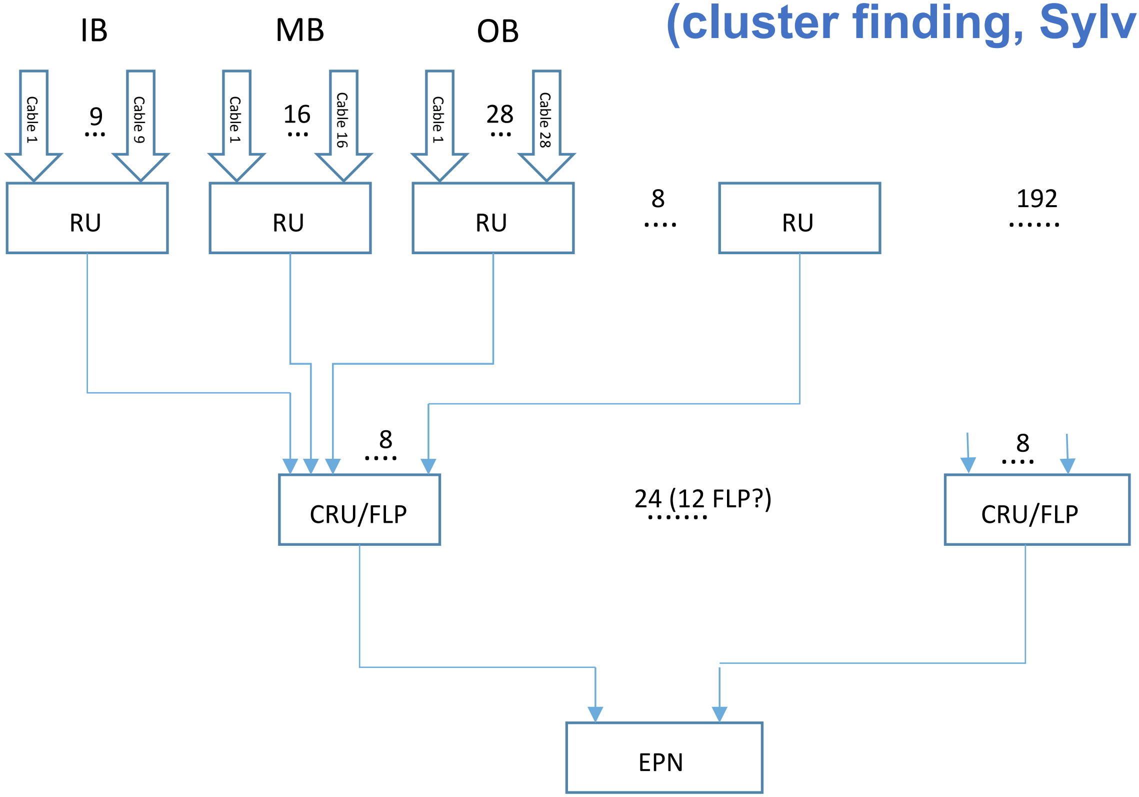
ROC bench tool performance

FLP 2 CRUs 4 PCIe endpoints
Total throughput



Data Hierarchy

1 CRU/FLP
IB, MB, OB mixed
load equalized, twice computing power
(cluster finding, Sylvain's simulation)



https://indico.cern.ch/event/698929/contributions/2866596/attachments/1620721/2596687/WP10_PRR_RU_Firmware_v0.pdf

Jo, RU PRR

Bandwidth is a (potential) issue only in the IB
If the number of FLPs is a free parameter (<= 24) in the fit then different optimizations are possible:

- keep 2 CRUs/FLP for the MB, OB, add machines only for the IB
- if we free also the number of CRUs we could further optimize connecting only 4 RU/CRU

N.B.
So far simulation suggests that computing power for cluster finding requires 24 FLPs (equally loaded)

(Personal) Conclusions

- (Too) many ways to optimize the layout

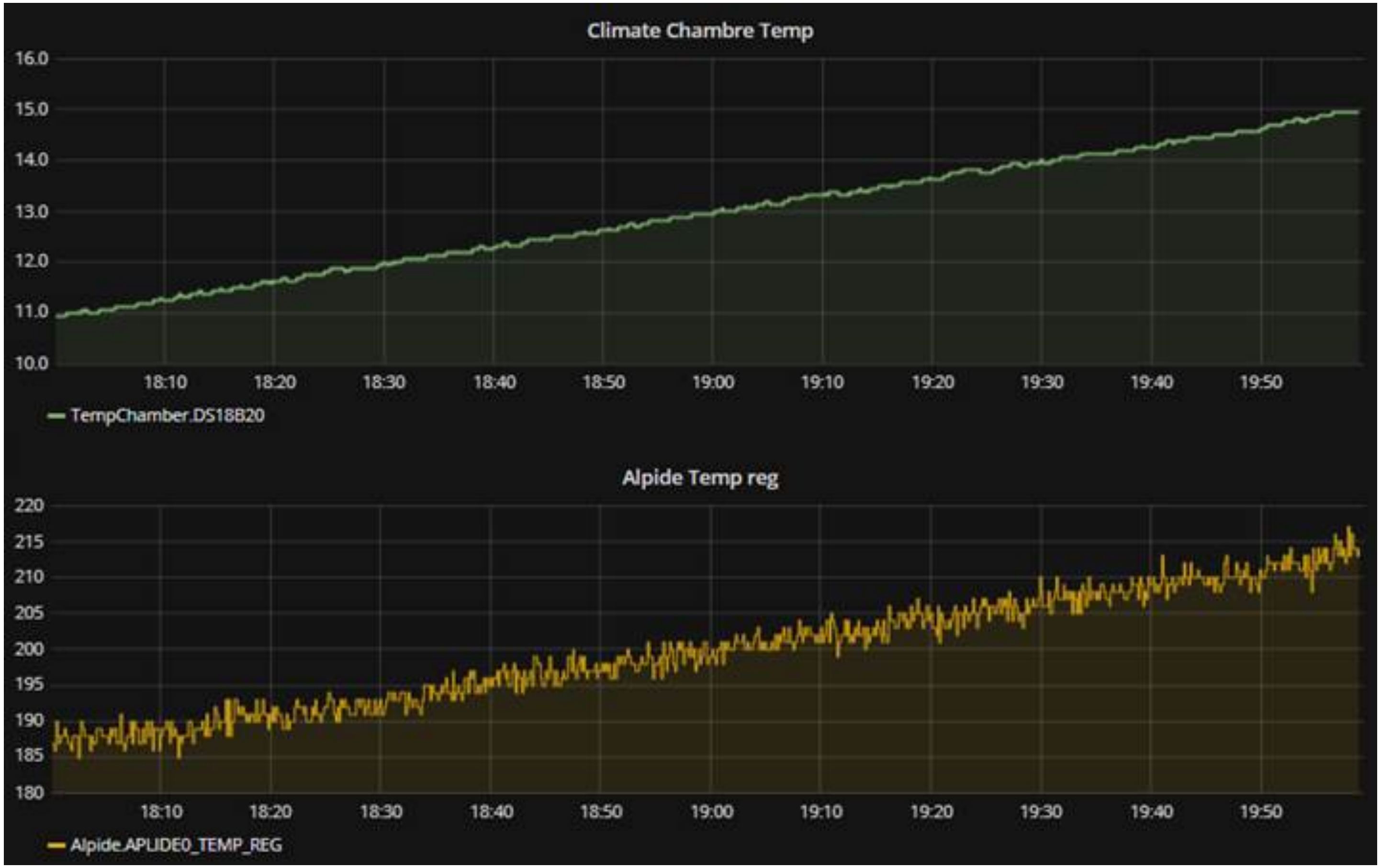
- Jo’s solution most elegant (all machines equal), optimization of computing power for cluster finding

- Layer-oriented layout more intuitive, matches power tree (which can change)

- Fibers from the RU will be connected to a patch panel in CR4, then to the CRUs
  Can any RU be connected to any CRU ?

- Can we design the patch panel to allow for “universal” configuration ?

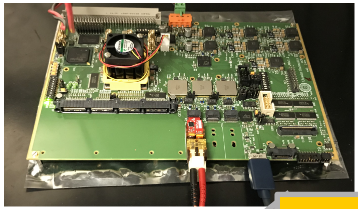
- Check cluster finding simulation (50 kHz vs 100 kHz)

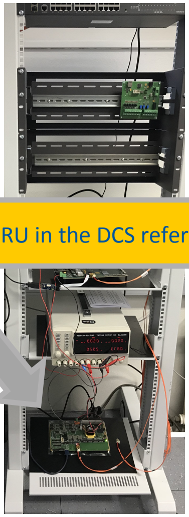


- In spite of its peculiar feature it seems possible to use the built-in sensor to monitor (variation of) the temperature.
- Optimal algorithm to be defined while refining the measurements
- <u>This applies to IB HIC, OB HIC to be checked</u>
- It would be good to give more HICs to Rune
- <u>Temperature reading concurrently to R/O to be checked</u>
- <u>Important to move to RU + PB</u>

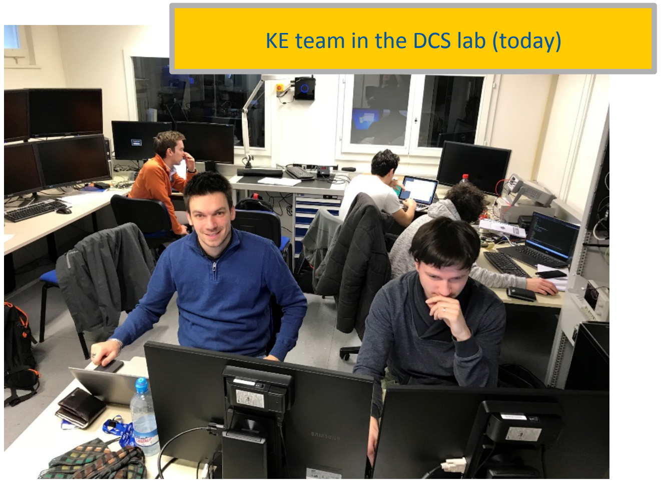
Frontend developments

- RU delivered to the DCS team
  (Matteo, Piero & Jo)
  - Card installed in the lab
  - Software prepared by DCS team
  (credits to Ombretta)
  can access all functionality of the SCA chip using WINCCOA and ALF
- Verified CANbus communication
- Software framework ready (FRED)
- SWT not yet implemented (needs also modification of ALF)



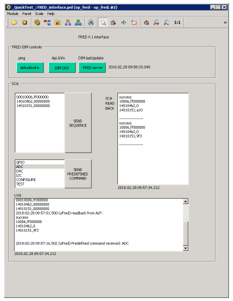


ITS RU in the DCS reference rack



KE team in the DCS lab (today)

ITS RU low-level access panels



```
2018.02.28 09:58:33.040
```

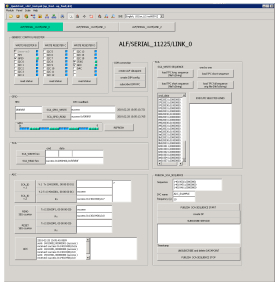
```
00010006,ff000000
140104b2,00000000
14010151,00000000
```

```
success
10006,ff000000
140104b2,0
14010151,a10
-----------------
success
10006,ff000000
140104b2,0
14010151,9f3
-----------------
```

2018.02.28 09:57:34.212

```
00010006,ff000000
140104b2,00000000
14010151,00000000
2018:02:28 09:57:01.500 (uFred) readback from ALF:
success
10006,ff000000
140104b2,0
14010151,9f3


2018:02:28 09:57:16.502 (uFred) Predefined command received: ADC
```

2018.02.28 09:57:34.212



ALF/SERIAL_11225/LINK_0

☐ (reserved) ☐ SPI ☑ GPIO ☐ I2C 0 ☐ I2C 1 ☐ I2C 2 ☐ I2C 3 ☐ I2C 4
☐ I2C 5 ☐ I2C 6 ☐ I2C 7 ☐ I2C 8 ☐ I2C 9 ☐ I2C 10 ☐ I2C 11 ☐ I2C 12
☐ I2C 13 ☐ I2C 14 ☐ I2C 15 ☑ JTAG ☑ ADC ☐ DAC

HEX: 0f0f0f0f
SCA_GPIO_WRITE: success 2018.02.28 10:05:10.721
SCA_GPIO_READ: success 0xf0f0f0f 2018.02.28 10:05:13.765
SCA_READ hex: success 0x2058400,0xf0f0f0f

V.1 Tx (14010091, 00 00 00 01)
V.2 Tx (140100D1, 00 00 00 01): success
Rx: success 0x14010400,0x7
READ SEU counter Tx (130000F1, 00 00 00 00): success
Rx: success 0x13010400,0x0
RESET SEU counter Tx (130000F0, 00 00 00 00)

```
2018-02-28 10:05:40.0889
sent: 14010002,00000001 (success )
received: success 0x14010400,0x2a
sent: 14010451,00000000 (success )
received: success 0x14010400,0x0
sent: 14010461,00000000 (success )
received: success 0x14010400,0x0
```

Sequence:
```
14010002,00000001
14010451,00000000
14010461,00000000
```
SVC name: ADC_EXAMPLE
frequency (s): 10

OUTLOOK

- DCS architecture taking shapes, although not yet frozen
- Temperature measurement to be clarified, next week ?
- 2-6 CRUs and 2 FLPs available from June for commissioning
- Urgent to provide the DCS colleagues with the latest version of the HW (RU + PB) and FW
- Define sequence/commands to operate the detector
- Dedicated slot in Tuesday WP10 meeting ?
- A WP10 dedicated person would also help

Alpide Monitor Spec

Johan

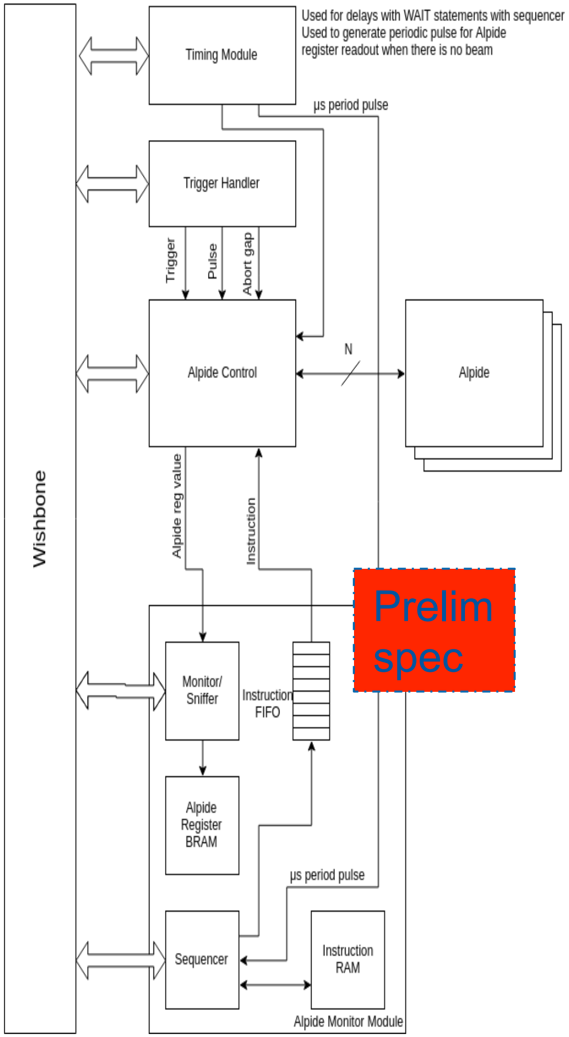
- Task: Read I, V, T ++ from ALPIDEs during operation
- Challenge: Using ALPIDE control bus - shared with trigger distribution

- Specs (short form):
  - Support 3 modes of operation:
    - Idle mode
    - Continous mode
    - Triggered mode (most challenging)
  - Deterministic timing for ALPIDE control bus access
  - Flexible solution with user specified ALPIDE registers to monitor
  - Monitored values available via Wishbone bus for *pulled* DCS access
    - Should also be easibly adaptable for push mechanisms

Proposed Architecture

Johan

- Using a sequencer to set up which registers to monitor and the actions needed to monitor them
  - Based on broad discussion in WP10 group to ensure flexibility for future needs
- A result memory for storage of the returned values.
  - Challenge:
    - Large memory for outer barrel (196 chips)
    - Need indirect wishbone addressing
- Note: The overall structure is more or less defined, but not all details are decided yet
- Continous mode:
  - Issue slow control command after strobetrigger
- Triggered mode:
  - Issue slow control command upon receiving abortgap trigger
  - Slowest: Worst case refresh cycle: ~3 Hz (outer layer with 196 ALPIDEs)
- Idle mode:
  - Running continously or upon command from software



Prelim spec

For more details see the DCS talk (Simon Voigt Nesbø

Values to be monitored per Power Boad (2 PU)

Status words (5):
5 * (2 * (8 + 1ack) + 1start + 1stop) = 120 CLK

V/I (68)
68 * (5 * (8 + 1ack) + 2start + 2stop) = 3332 CLK

T (6, 2 onboard + 4 external)
6 * (9 * (8 + 1ack) + 2start + 2stop) = 510 CLK

Total: 3962 clk, 80 ms @ 50kHz

Same exercise needed for ALPIDE & RU

Some (relevant) numbers

Parameters:
Bandwidth: data from RU to FLP, from FLP to EPN
Computing power available in each FLP node for cluster finding

The ALICE CRU has 24 optical inputs, 24 optical outputs
The RU has 3 optical outputs (data), 1 optical input (control)

We assume to connect 3 uplink fibers per RU, i.e. 8 RU/CRU
192 RU (48, 54, 90), 24 CRU (6, 6.75, 11.25)

1 optical link (up/down) has **3.2 Gb/s** bandwidth

	RU	**CRU**
IB	48	6
MB	54	6.75
OB	90	11.25
Total	192	24

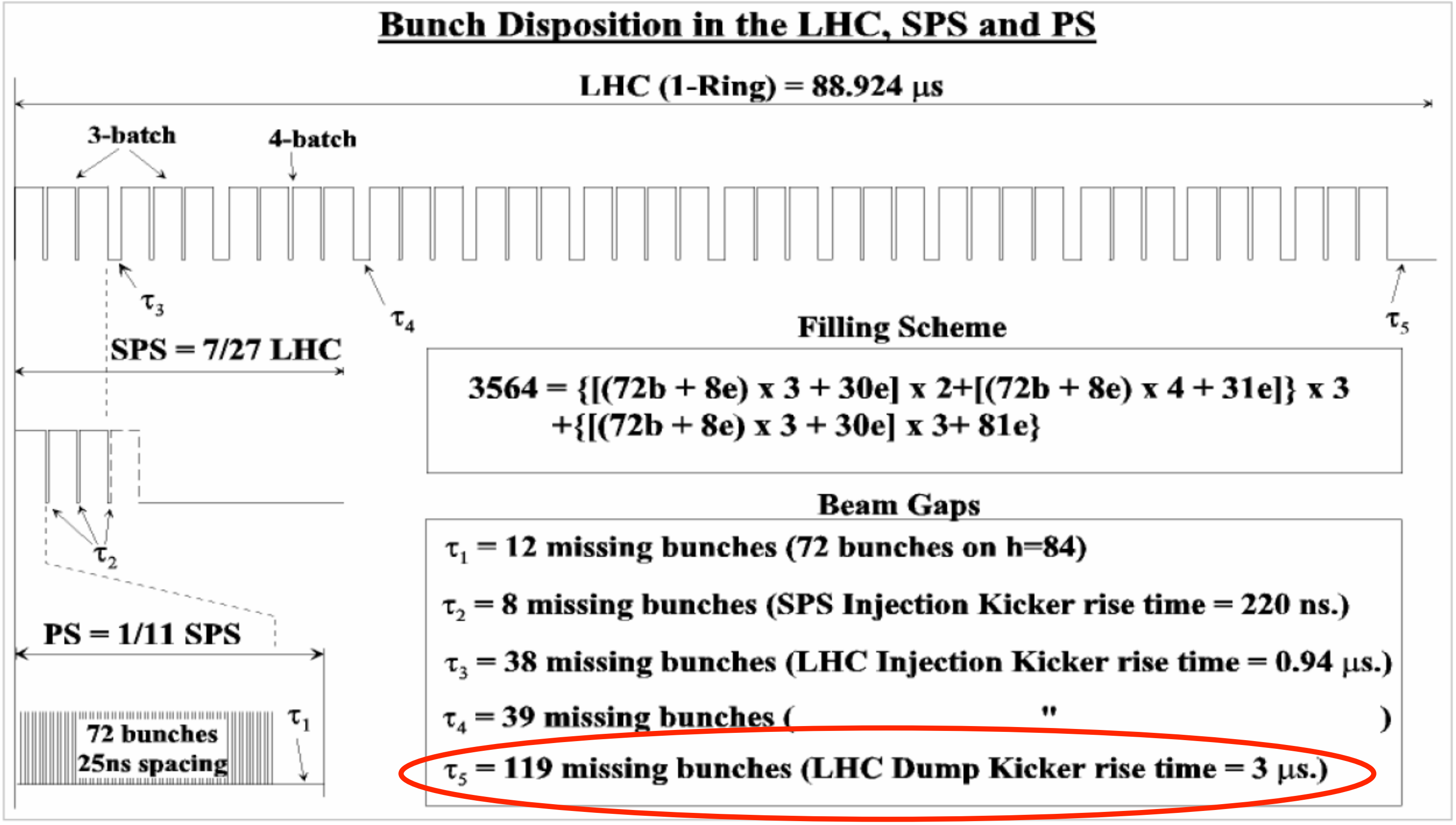
Reading T during collisions

- Temperature reading incompatible with the presence of trigger (share the same control line)
- The temperature can be readout only while the trigger is not issued
- To readout T (and AVDD) we need:

1. Set the ADCCtrlRegister 60 BC
2. Send the measurement command 60 BC
   Wait => 5 ms (ADC conversion time)
3. Read the output register 105 BC

No difference between IB/OB, Master/Slave

(calibration required after power-on, perhaps at SOR, omitted)



Bunch Disposition in the LHC, SPS and PS

Filling Scheme

$$3564 = \{[(72b + 8e) \times 3 + 30e] \times 2 + [(72b + 8e) \times 4 + 31e]\} \times 3 + \{[(72b + 8e) \times 3 + 30e] \times 3 + 81e\}$$

Beam Gaps

τ_1 = 12 missing bunches (72 bunches on h=84)

τ_2 = 8 missing bunches (SPS Injection Kicker rise time = 220 ns.)

τ_3 = 38 missing bunches (LHC Injection Kicker rise time = 0.94 μs.)

τ_4 = 39 missing bunches (")

τ_5 = 119 missing bunches (LHC Dump Kicker rise time = 3 μs.)

Figure 2.1: Proton bunches in the PS, SPS and one LHC ring. Note the partial filling of the SPS (3/11 or 4/11) and the voids due to kicker rise-time. One LHC ring is filled in ~3 min.

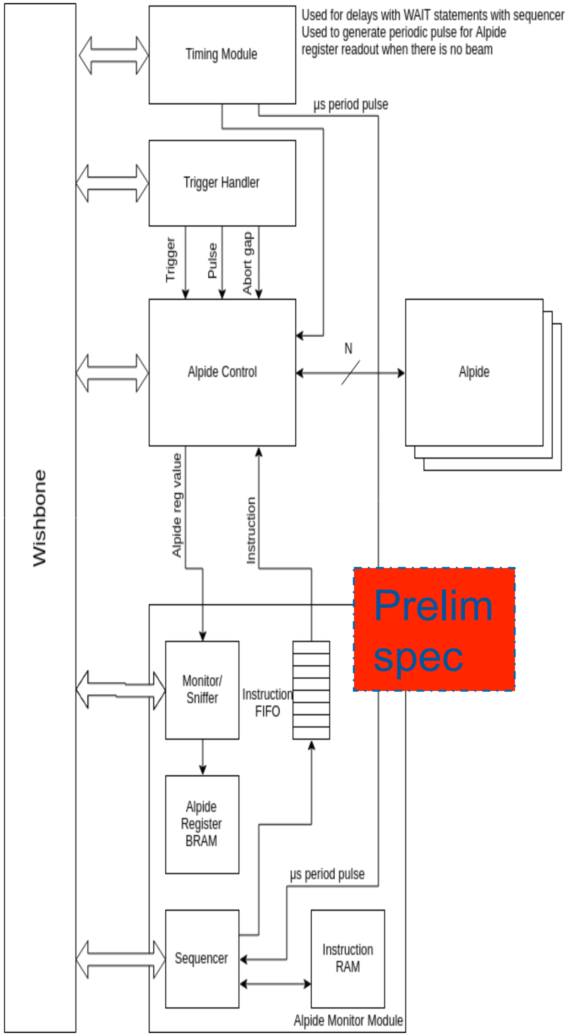
(https://edms.cern.ch/ui/file/445762/3/Vol3_Chap2_v4.pdf)

The LHC filling scheme has an interval of 119 BC w/o collisions (only B1 or B2 @ P2 during the same BC)

Proposed Architecture

Johan

- Using a sequencer to set up which registers to monitor and the actions needed to monitor them
  - Based on broad discussion in WP10 group to ensure flexibility for future needs
- A result memory for storage of the returned values.
  - Challenge:
    - Large memory for outer barrel (196 chips)
    - Need indirect wishbone addressing
- Note: The overall structure is more or less defined, but not all details are decided yet
- Continous mode:
  - Issue slow control command after strobetrigger
- Triggered mode:
  - Issue slow control command upon receiving abortgap trigger
  - Slowest: Worst case refresh cycle: ~3 Hz (outer layer with 196 ALPIDEs)
- Idle mode:
  - Running continously or upon command from software

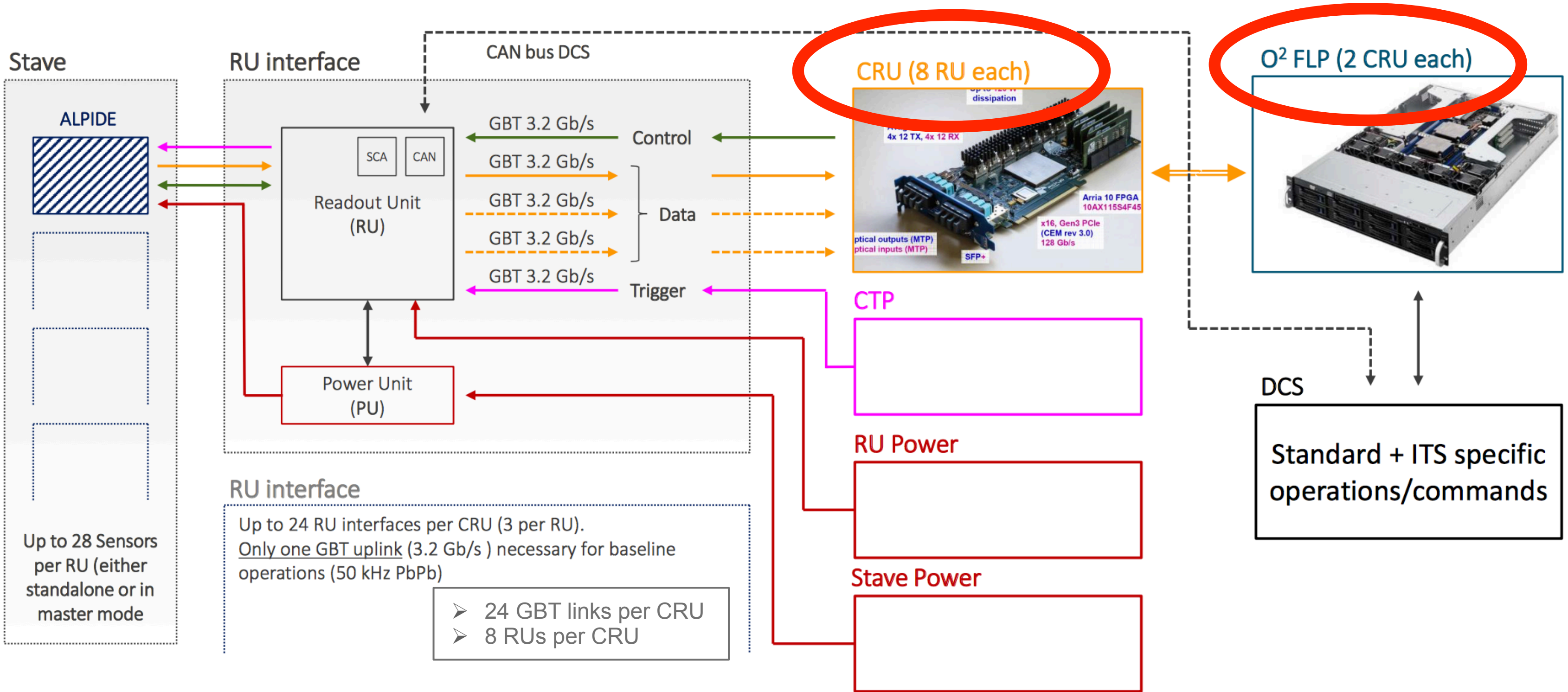


Prelim spec

For more details see the DCS talk (Simon Voigt Nesbø

ITS topology

	ALPIDES	Staves	Readout Units	Power Boards
IB	9x48	48	48	24
OB	8x14x54 14x14x90	144	144	144
Total	24120	192	192	168



Up to 28 Sensors per RU (either standalone or in master mode

RU interface

Up to 24 RU interfaces per CRU (3 per RU).
Only one GBT uplink (3.2 Gb/s) necessary for baseline operations (50 kHz PbPb)

- 24 GBT links per CRU
- 8 RUs per CRU

Standard + ITS specific operations/commands