Light Readout Electronics Current Status

Cayetano Santos

WA105 Collaboration Meeting - CERN - 22-23 March 2017

On behalf of the IN2P3 group - ACP/Omega/LAPP

Outline

1. Context
2. ASIC
3. uTCA Support
4. FMC Board
5. Summary

1. Context
2. ASIC
3. uTCA Support
4. FMC Board
5. Summary

IN2P3 Collaboration

Joint Effort

Joint effort between several in2p3 labs in France

Omega	Microelectronics Design Center for Physics and Medical Imaging - ASIC Development and testing
LAPP	Particle and Nuclear Physics - PCB layout and routing
APC	Cosmology and Astroparticle Physics - ASIC testing, PCB schematics
IPNL	Nuclear Phycis - General support, advice and firmware

IN2P3 Collaboration

Joint Effort

Joint effort between several in2p3 labs in France

Omega Microelectronics Design Center for Physics and Medical Imaging - ASIC Development and testing

LAPP Particle and Nuclear Physics - PCB layout and routing

APC Cosmology and Astroparticle Physics - ASIC testing, PCB schematics

IPNL Nuclear Phycis - General support, advice and firmware

IN2P3 Collaboration

Joint Effort

Joint effort between several in2p3 labs in France

Omega	Microelectronics Design Center for Physics and Medical Imaging - ASIC Development and testing
LAPP	Particle and Nuclear Physics - PCB layout and routing
APC	Cosmology and Astroparticle Physics - ASIC testing, PCB schematics
IPNL	Nuclear Phycis - General support, advice and firmware

IN2P3 Collaboration

Joint Effort

Joint effort between several in2p3 labs in France

Omega	Microelectronics Design Center for Physics and Medical Imaging - ASIC Development and testing
LAPP	Particle and Nuclear Physics - PCB layout and routing
APC	Cosmology and Astroparticle Physics - ASIC testing, PCB schematics
IPNL	Nuclear Phycis - General support, advice and firmware

IN2P3 Collaboration

Joint Effort

Joint effort between several in2p3 labs in France

Omega Microelectronics Design Center for Physics and Medical Imaging - ASIC Development and testing

LAPP Particle and Nuclear Physics - PCB layout and routing

APC Cosmology and Astroparticle Physics - ASIC testing, PCB schematics

IPNL Nuclear Phycis - General support, advice and firmware

IN2P3 Collaboration

Joint Effort

Joint effort between several in2p3 labs in France

Omega Microelectronics Design Center for Physics and Medical Imaging - ASIC Development and testing

LAPP Particle and Nuclear Physics - PCB layout and routing

APC Cosmology and Astroparticle Physics - ASIC testing, PCB schematics

IPNL Nuclear Phycis - General support, advice and firmware

IN2P3 Collaboration

Joint Effort

Joint effort between several in2p3 labs in France

Omega Microelectronics Design Center for Physics and Medical Imaging - ASIC Development and testing

LAPP Particle and Nuclear Physics - PCB layout and routing

APC Cosmology and Astroparticle Physics - ASIC testing, PCB schematics

IPNL Nuclear Phycis - General support, advice and firmware

(Micro)electronics front end for PMTs

Goals

Go beyond ASIC functionality

Integrate an state of the art, latest generation ASIC completed with a few FPGA advanced features

- Advanced: dead timeless monitoring system
- Digital event counting (not an ASIC feature)
- Endless (x-bits) time stamping

Implement Digital Pulse Processing

Perform advanced DSP on the samples with FPGA fabric

- Sampling of analog signals
- Compute falling tail, windowing, etc.
- Event rejection, pile up handling, etc.

Goals

Go beyond ASIC functionality

Integrate an state of the art, latest generation ASIC completed with a few FPGA advanced features

- Advanced: dead timeless monitoring system
- Digital event counting (not an ASIC feature)
- Endless (x-bits) time stamping

Implement Digital Pulse Processing

Perform advanced DSP on the samples with FPGA fabric

- Sampling of analog signals
- Compute falling tail, windowing, etc.
- Event rejection, pile up handling, etc.

Goals

Go beyond ASIC functionality

Integrate an state of the art, latest generation ASIC completed with a few FPGA advanced features

- Advanced: dead timeless monitoring system
- Digital event counting (not an ASIC feature)
- Endless (x-bits) time stamping

Implement Digital Pulse Processing

Perform advanced DSP on the samples with FPGA fabric

- Sampling of analog signals
- Compute falling tail, windowing, etc.
- Event rejection, pile up handling, etc.

Steps

First prototype developed in 2015

- Using former ASIC generation (ParisROC)

Second version under current development

- Newest ASIC generation (CatiROC)
- Bug fix release

Production release 2018

- Double width AMC, 32 channels, . . .

Steps

First prototype developed in 2015

- Using former ASIC generation (ParisROC)

Second version under current development

- Newest ASIC generation (CatiROC)
- Bug fix release

Production release 2018

- Double width AMC, 32 channels, . . .

Steps

First prototype developed in 2015

- Using former ASIC generation (ParisROC)

Second version under current development

- Newest ASIC generation (CatiROC)
- Bug fix release

Production release 2018

- Double width AMC, 32 channels, . . .

1 Context

2 **ASIC**

3 uTCA Support

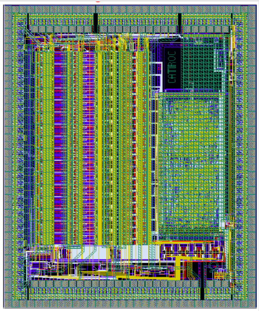
4 FMC Board

5 Summary

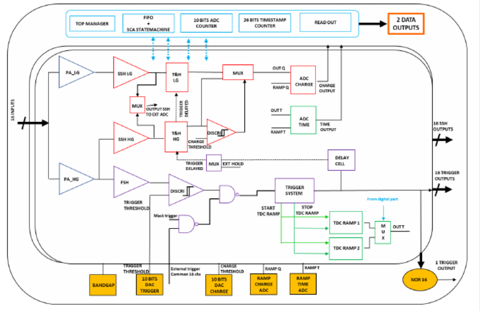
CatiROC

Fast and Slow Shapers - Two capacitors / channel AMS SiGe 0.35 um - 13.2 mm2 - TQFP208

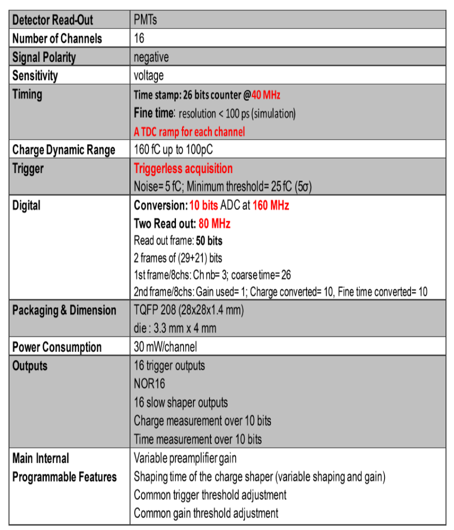
Architecture



Block Diagram

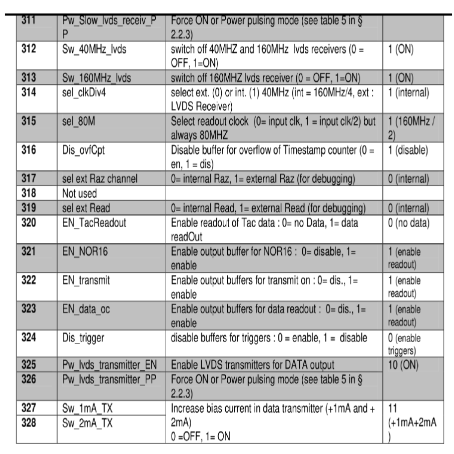


CatiROC features



Detector Read-Out	PMTs
Number of Channels	16
Signal Polarity	negative
Sensitivity	voltage
Timing	**Time stamp: 26 bits counter @40 MHz** **Fine time**: resolution < 100 ps (simulation) A TDC ramp for each channel
Charge Dynamic Range	160 fC up to 100pC
Trigger	Triggerless acquisition Noise= 5 fC; Minimum threshold= 25 fC (5σ)
Digital	**Conversion: 10 bits** ADC at **160 MHz** **Two Read out: 80 MHz** Read out frame: **50 bits** 2 frames of (29+21) bits 1st frame/8chs: Ch nb= 3; coarse time= 26 2nd frame/8chs: Gain used= 1; Charge converted= 10, Fine time converted= 10
Packaging & Dimension	TQFP 208 (28x28x1.4 mm) die : 3.3 mm x 4 mm
Power Consumption	30 mW/channel
Outputs	16 trigger outputs NOR16 16 slow shaper outputs Charge measurement over 10 bits Time measurement over 10 bits
Main Internal Programmable Features	Variable preamplifier gain Shaping time of the charge shaper (variable shaping and gain) Common trigger threshold adjustment Common gain threshold adjustment

Some specifications



311	Pw_Slow_lvds_receiv_PP	Force ON or Power pulsing mode (see table 5 in § 2.2.3)	
312	Sw_40MHz_lvds	switch off 40MHZ and 160MHz lvds receivers (0 = OFF, 1=ON)	1 (ON)
313	Sw_160MHz_lvds	switch off 160MHz lvds receiver (0 = OFF, 1=ON)	1 (ON)
314	sel_clkDiv4	select ext. (0) or int. (1) 40MHz (int = 160MHz/4, ext : LVDS Receiver)	1 (internal)
315	sel_80M	Select readout clock (0= input clk, 1 = input clk/2) but always 80MHZ	1 (160MHz / 2)
316	Dis_ovfCpt	Disable buffer for overflow of Timestamp counter (0 = en, 1 = dis)	1 (disable)
317	sel ext Raz channel	0= internal Raz, 1= external Raz (for debugging)	0 (internal)
318	Not used		
319	sel ext Read	0= internal Read, 1= external Read (for debugging)	0 (internal)
320	EN_TacReadout	Enable readout of Tac data : 0= no Data, 1= data readOut	0 (no data)
321	EN_NOR16	Enable output buffer for NOR16 : 0= disable, 1= enable	1 (enable readout)
322	EN_transmit	Enable output buffers for transmit on : 0= dis., 1= enable	1 (enable readout)
323	EN_data_oc	Enable output buffers for data readout : 0= dis., 1= enable	1 (enable readout)
324	Dis_trigger	disable buffers for triggers : 0 = enable, 1 = disable	0 (enable triggers)
325	Pw_lvds_transmitter_EN	Enable LVDS transmitters for DATA output	10 (ON)
326	Pw_lvds_transmitter_PP	Force ON or Power pulsing mode (see table 5 in § 2.2.3)	
327	Sw_1mA_TX	Increase bias current in data transmitter (+1mA and +2mA) 0 =OFF, 1= ON	11 (+1mA+2mA)
328	Sw_2mA_TX		

- I/O 1-bit shift register
- 328 bits to setup

CatiROC facts

16 channels readout chip for PMTs with fully independent charge and time measurements

16 negative inputs: each voltage input is sent to high/low noise amplifiers for small and large signals to ensure a good charge precision (30 fC)

Variable 8 bit gain / amplifier / channel

Charge: preamp followed by 2 variable slow shapers sent to analog memories to measure up to 50pC

Time: coarse + fine timing

10 bits Wilkinson ADC to convert charge and fine time @ 160 MHz

A fast shaper / channel followed by a discri for auto-trigger. One common 10 bit threshold

Digital section handles the acq, conversion and readout, providing a 26 bits coarse time measurement (TS)

CatiROC facts

16 channels readout chip for PMTs with fully independent charge and time measurements

16 negative inputs: each voltage input is sent to high/low noise amplifiers for small and large signals to ensure a good charge precision (30 fC)

Variable 8 bit gain / amplifier / channel

Charge: preamp followed by 2 variable slow shapers sent to analog memories to measure up to 50pC

Time: coarse + fine timing

10 bits Wilkinson ADC to convert charge and fine time @ 160 MHz

A fast shaper / channel followed by a discri for auto-trigger. One common 10 bit threshold

Digital section handles the acq, conversion and readout, providing a 26 bits coarse time measurement (TS)

CatiROC facts

16 channels readout chip for PMTs with fully independent charge and time measurements

16 negative inputs: each voltage input is sent to high/low noise amplifiers for small and large signals to ensure a good charge precision (30 fC)

Variable 8 bit gain / amplifier / channel

Charge: preamp followed by 2 variable slow shapers sent to analog memories to measure up to 50pC

Time: coarse + fine timing

10 bits Wilkinson ADC to convert charge and fine time @ 160 MHz

A fast shaper / channel followed by a discri for auto-trigger. One common 10 bit threshold

Digital section handles the acq, conversion and readout, providing a 26 bits coarse time measurement (TS)

CatiROC facts

16 channels readout chip for PMTs with fully independent charge and time measurements

16 negative inputs: each voltage input is sent to high/low noise amplifiers for small and large signals to ensure a good charge precision (30 fC)

Variable 8 bit gain / amplifier / channel

Charge: preamp followed by 2 variable slow shapers sent to analog memories to measure up to 50pC

Time: coarse + fine timing

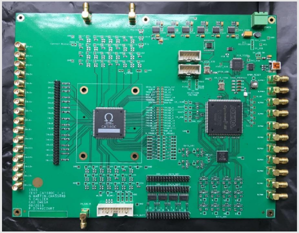
10 bits Wilkinson ADC to convert charge and fine time @ 160 MHz

A fast shaper / channel followed by a discri for auto-trigger. One common 10 bit threshold

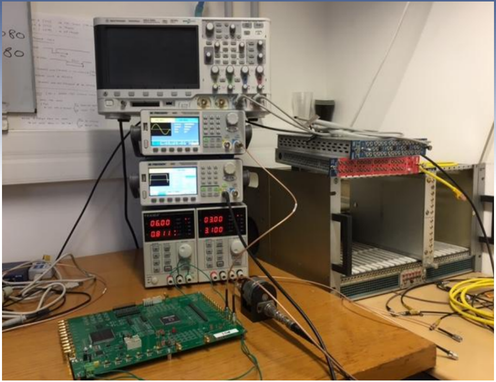
Digital section handles the acq, conversion and readout, providing a 26 bits coarse time measurement (TS)

CatiROC Checkout Testbench

Tests board



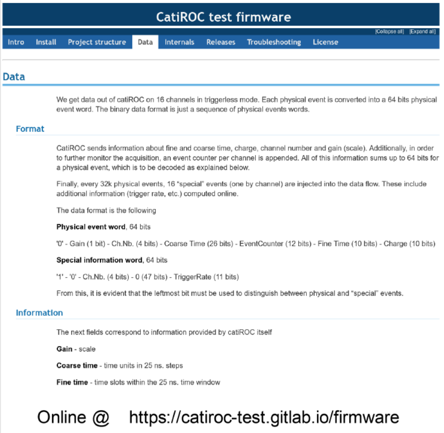
- USB / FPGA controlled



- Fanout board for timing, crosstalk, etc.
- Arbitrary Pulse Generator
- High gain (10^7) PMT
- Histogramming oscilloscope

Testbench coding

Firmware


CatiROC test firmware
[Collapse all] [Expand all]
Intro Install Project structure Data Internals Releases Troubleshooting License

Data

We get data out of catiROC on 16 channels in triggerless mode. Each physical event is converted into a 64 bits physical event word. The binary data format is just a sequence of physical events words.

Format

CatiROC sends information about fine and coarse time, charge, channel number and gain (scale). Additionally, in order to further monitor the acquisition, an event counter per channel is appended. All of this information sums up to 64 bits for a physical event, which is to be decoded as explained below.

Finally, every 32k physical events, 16 "special" events (one by channel) are injected into the data flow. These include additional information (trigger rate, etc.) computed online.

The data format is the following

Physical event word, 64 bits

'0' - Gain (1 bit) - Ch.Nb. (4 bits) - Coarse Time (26 bits) - EventCounter (12 bits) - Fine Time (10 bits) - Charge (10 bits)

Special information word, 64 bits

'1' - '0' - Ch.Nb. (4 bits) - 0 (47 bits) - TriggerRate (11 bits)

From this, it is evident that the leftmost bit must be used to distinguish between physical and "special" events.

Information

The next fields correspond to information provided by catiROC itself

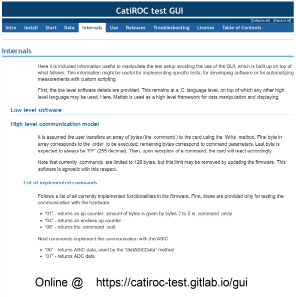
Gain - scale

Coarse time - time units in 25 ns. steps

Fine time - time slots within the 25 ns. time window

Online @ https://catiroc-test.gitlab.io/firmware

Software


CatiROC test GUI
[Collapse all] [Expand all]
Intro Install Start Data Internals Use Releases Troubleshooting License Table of Contents

Internals

Here it is included information useful to manipulate the test setup avoiding the use of the GUI, which is built up on top of what follows. This information might be useful for implementing specific tests, for developing software or for automatizing measurements with custom scripting.

First, the low level software details are provided. This remains at a `C` language level, on top of which any other high level language may be used. Here, Matlab is used as a high level framework for data manipulation and displaying.

Low level software

High level communication model

It is assumed the user transfers an array of bytes (the `command`) to the card using the `Write` method. First byte in array corresponds to the `order` to be executed; remaining bytes correspond to command parameters. Last byte is expected to always be "FF" (255 decimal). Then, upon reception of a command, the card will react accordingly.

Note that currently `commands` are limited to 128 bytes, but this limit may be removed by updating the firmware. This software is agnostic with this respect.

List of implemented commands

Follows a list of all currently implemented functionalities in the firmware. First, these are provided only for testing the communication with the hardware

- "01" - returns an up counter; amount of bytes is given by bytes 2 to 5 in `command` array
- "04" - returns an endless up counter
- "05" - returns the `command` sent

Next commands implement the communication with the ASIC

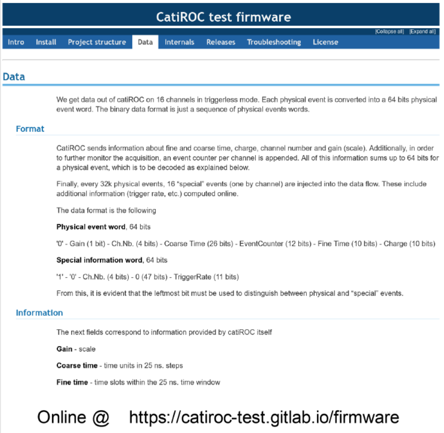
- "06" - returns ASIC data, used by the "GetASICData" method
- "07" - returns ADC data

Online @ https://catiroc-test.gitlab.io/gui

- Full stack
- From scratch

Testbench coding

Firmware



Data

We get data out of catiROC on 16 channels in triggerless mode. Each physical event is converted into a 64 bits physical event word. The binary data format is just a sequence of physical events words.

Format

CatiROC sends information about fine and coarse time, charge, channel number and gain (scale). Additionally, in order to further monitor the acquisition, an event counter per channel is appended. All of this information sums up to 64 bits for a physical event, which is to be decoded as explained below.

Finally, every 32k physical events, 16 "special" events (one by channel) are injected into the data flow. These include additional information (trigger rate, etc.) computed online.

The data format is the following

Physical event word, 64 bits

'0' - Gain (1 bit) - Ch.Nb. (4 bits) - Coarse Time (26 bits) - EventCounter (12 bits) - Fine Time (10 bits) - Charge (10 bits)

Special information word, 64 bits

'1' - '0' - Ch.Nb. (4 bits) - 0 (47 bits) - TriggerRate (11 bits)

From this, it is evident that the leftmost bit must be used to distinguish between physical and "special" events.

Information

The next fields correspond to information provided by catiROC itself

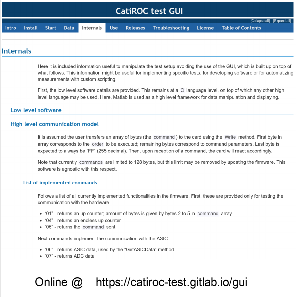
Gain - scale

Coarse time - time units in 25 ns. steps

Fine time - time slots within the 25 ns. time window

Online @ https://catiroc-test.gitlab.io/firmware

Software



Internals

Here it is included information useful to manipulate the test setup avoiding the use of the GUI, which is built up on top of what follows. This information might be useful for implementing specific tests, for developing software or for automatizing measurements with custom scripting.

First, the low level software details are provided. This remains at a `C` language level, on top of which any other high level language may be used. Here, Matlab is used as a high level framework for data manipulation and displaying.

Low level software

High level communication model

It is assumed the user transfers an array of bytes (the `command`) to the card using the `Write` method. First byte in array corresponds to the `order` to be executed; remaining bytes correspond to command parameters. Last byte is expected to always be "FF" (255 decimal). Then, upon reception of a command, the card will react accordingly.

Note that currently `commands` are limited to 128 bytes, but this limit may be removed by updating the firmware. This software is agnostic with this respect.

List of implemented commands

Follows a list of all currently implemented functionalities in the firmware. First, these are provided only for testing the communication with the hardware

- "01" - returns an up counter; amount of bytes is given by bytes 2 to 5 in `command` array
- "04" - returns an endless up counter
- "05" - returns the `command` sent

Next commands implement the communication with the ASIC

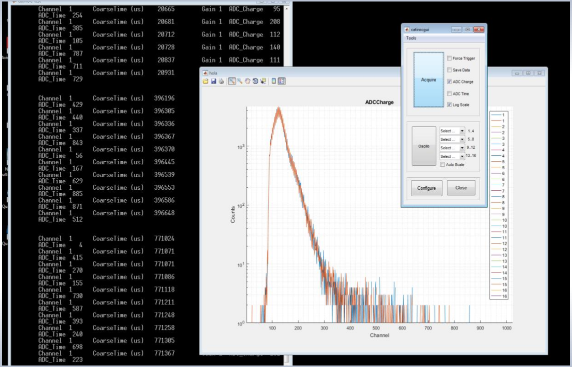
- "06" - returns ASIC data, used by the "GetASICData" method
- "07" - returns ADC data

Online @ https://catiroc-test.gitlab.io/gui

- Full stack
- From scratch

Testbench data taking

- Low level software in C++
- Gui / data processing in Matlab
- Windows / Linux
- Amplitude histogramming
- Timing histogramming
- Data recording
- Fitting, etc.



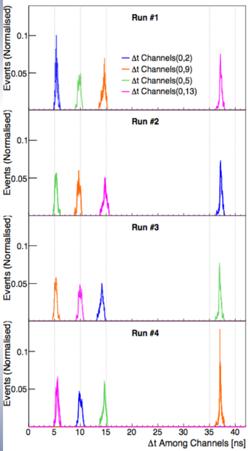
```
Channel  1      CoarseTime (us)    20665        Gain 1  ADC_Charge   95
ADC_Time  254
Channel  1      CoarseTime (us)    20681        Gain 1  ADC_Charge  208
ADC_Time  385
Channel  1      CoarseTime (us)    20712        Gain 1  ADC_Charge  112
ADC_Time  105
Channel  1      CoarseTime (us)    20728        Gain 1  ADC_Charge  140
ADC_Time  787
Channel  1      CoarseTime (us)    20837        Gain 1  ADC_Charge  111
ADC_Time  711
Channel  1      CoarseTime (us)    20931
ADC_Time  729


Channel  1      CoarseTime (us)   396196
ADC_Time  429
Channel  1      CoarseTime (us)   396305
ADC_Time  440
Channel  1      CoarseTime (us)   396336
ADC_Time  337
Channel  1      CoarseTime (us)   396367
ADC_Time  843
Channel  1      CoarseTime (us)   396370
ADC_Time   56
Channel  1      CoarseTime (us)   396445
ADC_Time  167
Channel  1      CoarseTime (us)   396539
ADC_Time  629
Channel  1      CoarseTime (us)   396553
ADC_Time  885
Channel  1      CoarseTime (us)   396586
ADC_Time  871
Channel  1      CoarseTime (us)   396648
ADC_Time  512


Channel  1      CoarseTime (us)   771024
ADC_Time    4
Channel  1      CoarseTime (us)   771071
ADC_Time  415
Channel  1      CoarseTime (us)   771071
ADC_Time  270
Channel  1      CoarseTime (us)   771086
ADC_Time  155
Channel  1      CoarseTime (us)   771118
ADC_Time  730
Channel  1      CoarseTime (us)   771211
ADC_Time  587
Channel  1      CoarseTime (us)   771248
ADC_Time  393
Channel  1      CoarseTime (us)   771258
ADC_Time  240
Channel  1      CoarseTime (us)   771305
ADC_Time  698
Channel  1      CoarseTime (us)   771367
ADC_Time  223
```

☐ Force Trigger
☐ Save Data
☑ ADC Charge
☐ ADC Time
☑ Log Scale
☐ Auto Scale

CatiROC Timing . . .



For each event, compute Δt between two channels experiencing different delays

Δt mean value is set by cable length
Δt RMS is an estimate of time resolution

Time difference between channels is:

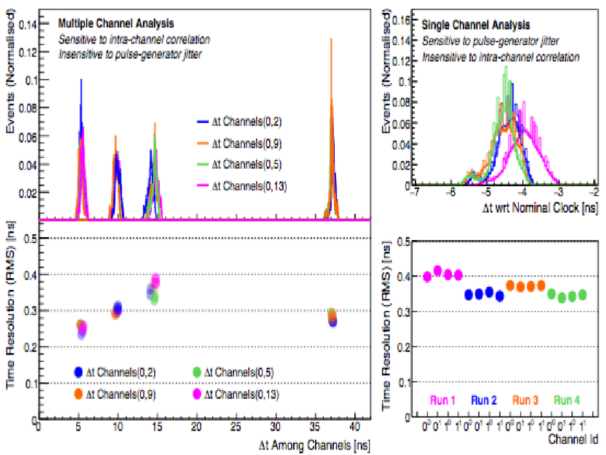
- insensitive to pulse generator jitter ✓
- insensitive to systematics common to all channels (intra-channel correlation) ✗

Assuming no correlation:
σT(single channel) = RMS /√2

Conservative estimation:
σT(single channel) ~ RMS

- Single rate digital pulse generator
- Signal is splitted to 4 channels
- Different cable lengths are used
- Raw data saving
- Computing time differences
- Cables are rotated: four data sets

. . . less than ~250 ps



- Coherent between data sets
- Timing independent of channel
- Independent of capacitor in ch.
- Method takes into account all non linearities
- Single ch. timing worst due to p.g. jitter
- Less than ~400 ps ch. to ch.

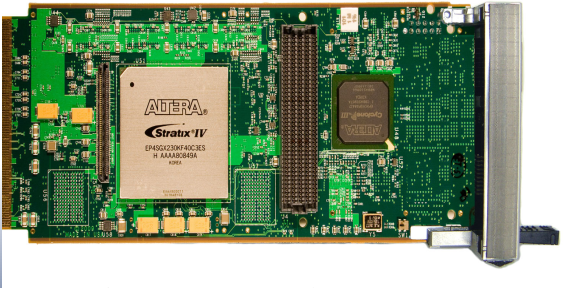
1. Context
2. ASIC
3. **uTCA Support**
4. FMC Board
5. Summary

Prototyping crate - Native C1

- Low cost, 1u 19' rack-mounted MTCA Chassis
- 1 MCH - 6 AdvancedMC (AMC) - 1 Power module (PM) slots
- Maximum 80W per AMC slot

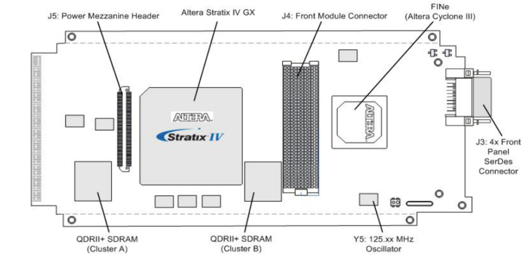
- 1 GbE routed to AMC Port 0 - kTCLKA, TCLKB, and FCLKA to each AMC
- Point-t-point SATA/SAS Port 2 & 3 - Fabrics D, E, F, G Port 4-7

Mother board - S4 AMC

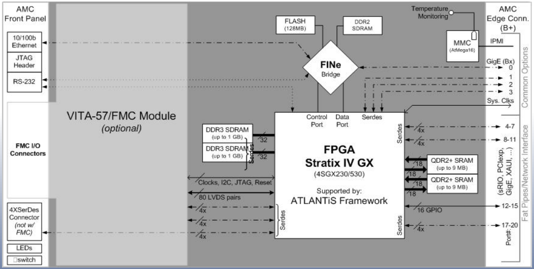


- Altera Stratix IV GX FPGA
- I BittWare's FINe™ Host/Control Bridge
- 10/100/1000 Ethernet, SerDes, LVDS, RS-232, and JTAG
- 2 GBytes of memory
- Fully connected to AMC (16 ports SerDes, 4 ports GPIO)
- I VITA 57 FMC site for I/O expansion
- I Six clocks

Mother board - S4 AMC - Block Diagram



J5: Power Mezzanine Header
Altera Stratix IV GX
J4: Front Module Connector
FINe (Altera Cyclone III)
ALTERA
Stratix IV
ALTERA
J3: 4x Front Panel SerDes Connector
QDRII+ SDRAM (Cluster A)
QDRII+ SDRAM (Cluster B)
Y5: 125.xx MHz Oscillator

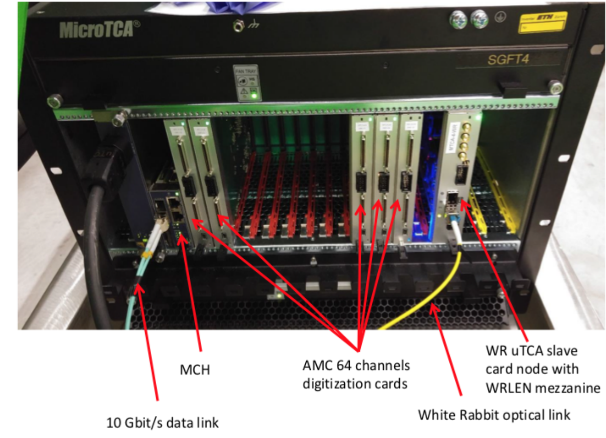


- Altera Stratix IV GX FPGA
- I BittWare's FINe™ Host/Control Bridge
- 10/100/1000 Ethernet, SerDes, LVDS, RS-232, and JTAG
- 2 GBytes of memory
- Fully connected to AMC (16 ports SerDes, 4 ports GPIO)
- I VITA 57 FMC site for I/O expansion
- I Six clocks

Production crate

Probably to be used during the experiment - Currently under discussion

How a crates was looking like before VHDCI signals cabling to the warm flange


MicroTCA
SGFT4
MCH
AMC 64 channels digitization cards
WR uTCA slave card node with WRLEN mezzanine
10 Gbit/s data link
White Rabbit optical link

- 12 double width slots
- 2 MCH slots
- Dedicated hardware for synchronizing
- Developed by / image from IPNL

1 Context

2 ASIC

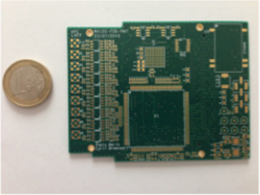
3 uTCA Support

4 **FMC Board**

5 Summary

First prototype of FMC board

Scale



Printed Circuit

Top view

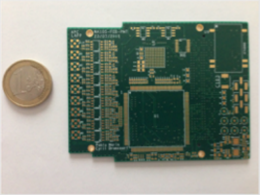
ParisROC ASIC + ADC

Bottom view

FMC connector

First prototype of FMC board

Scale



Printed Circuit

Top view

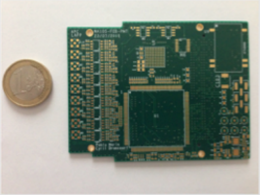
ParisROC ASIC + ADC

Bottom view

FMC connector

First prototype of FMC board

Scale



Printed Circuit

Top view

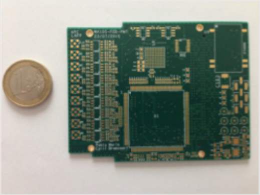
ParisROC ASIC + ADC

Bottom view

FMC connector

First prototype of FMC board

Scale



Printed Circuit

Top view

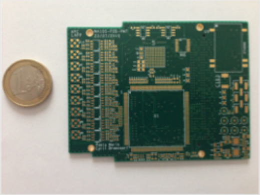
ParisROC ASIC + ADC

Bottom view

FMC connector

First prototype of FMC board

Scale



Printed Circuit

Top view

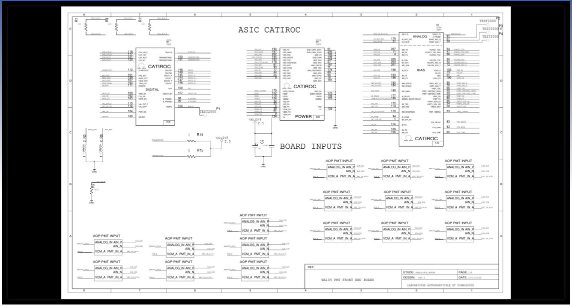
ParisROC ASIC + ADC

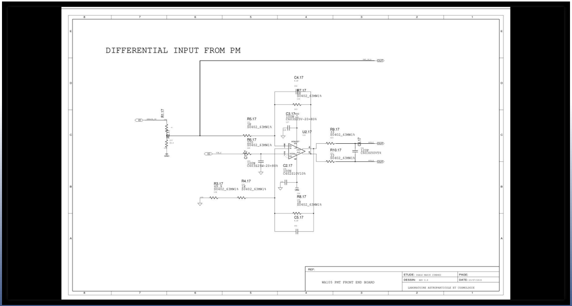
Bottom view

FMC connector

16 Channels ASIC + ADC

Current prototype - Schematics

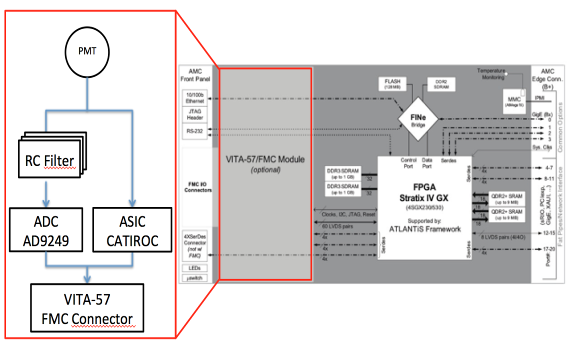




- Low pass filtering
- ADC9249, 65 MHz, 14 bits
- CatiROC ASIC - 16 ch.
- Power management
- 4 Spare signals
- VITA 57 FMC connector

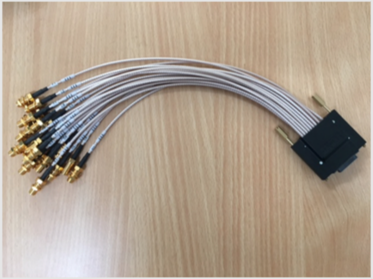
Block Diagram

- Splitting of analog inputs
- Anti aliasing filter
- Analog processing in ASIC
- Samples go to FPGA
- Readout of data from ASIC to FPGA
- Data is merged and processed in FPGA



How to bring 20 signals in a reduced space

Bunch of 20 cables



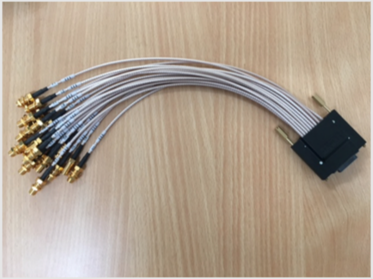
30 cm. length

SMA standard

SMA female

How to bring 20 signals in a reduced space

Bunch of 20 cables



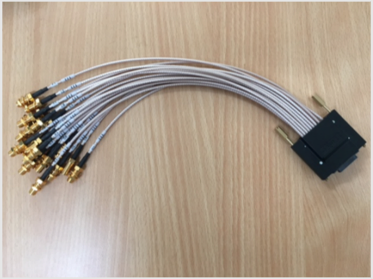
30 cm. length

SMA standard

SMA female

How to bring 20 signals in a reduced space

Bunch of 20 cables



30 cm. length

SMA standard

SMA female

To be plugged to RG58 cabling from splitter box

through a male SMA connector

1 Context

2 ASIC

3 uTCA Support

4 FMC Board

5 Summary

cnrs IN2P3 Les deux infinis

Conclusions

- Work in progress . . .
- ASIC fully characterized: good timing -> done
- Board schematics -> done
- End of routing by the end on March
- Board production during April
- ADC data capture firmware -> done
- ASIC control and data capture firmware -> done

But still an open issue . . .

- What kind of electrical standard for spare signals ?

Conclusions

- Work in progress . . .
- ASIC fully characterized: good timing -> done
- Board schematics -> done
- End of routing by the end on March
- Board production during April
- ADC data capture firmware -> done
- ASIC control and data capture firmware -> done

But still an open issue . . .

- What kind of electrical standard for spare signals ?

Conclusions

- Work in progress . . .
- ASIC fully characterized: good timing -> done
- Board schematics -> done
- End of routing by the end on March
- Board production during April
- ADC data capture firmware -> done
- ASIC control and data capture firmware -> done

But still an open issue . . .

- What kind of electrical standard for spare signals ?

Conclusions

- Work in progress . . .
- ASIC fully characterized: good timing -> done
- Board schematics -> done
- End of routing by the end on March
- Board production during April
- ADC data capture firmware -> done
- ASIC control and data capture firmware -> done

But still an open issue . . .

- What kind of electrical standard for spare signals ?

Conclusions

- Work in progress . . .
- ASIC fully characterized: good timing -> done
- Board schematics -> done
- End of routing by the end on March
- Board production during April
- ADC data capture firmware -> done
- ASIC control and data capture firmware -> done

But still an open issue . . .

- What kind of electrical standard for spare signals ?

Conclusions

- Work in progress . . .
- ASIC fully characterized: good timing -> done
- Board schematics -> done
- End of routing by the end on March
- Board production during April
- ADC data capture firmware -> done
- ASIC control and data capture firmware -> done

But still an open issue . . .

- What kind of electrical standard for spare signals ?

Conclusions

- Work in progress . . .
- ASIC fully characterized: good timing -> done
- Board schematics -> done
- End of routing by the end on March
- Board production during April
- ADC data capture firmware -> done
- ASIC control and data capture firmware -> done

But still an open issue . . .

- What kind of electrical standard for spare signals ?

Conclusions

- Work in progress . . .
- ASIC fully characterized: good timing -> done
- Board schematics -> done
- End of routing by the end on March
- Board production during April
- ADC data capture firmware -> done
- ASIC control and data capture firmware -> done

But still an open issue . . .

- What kind of electrical standard for spare signals ?

Conclusions

- Work in progress . . .
- ASIC fully characterized: good timing -> done
- Board schematics -> done
- End of routing by the end on March
- Board production during April
- ADC data capture firmware -> done
- ASIC control and data capture firmware -> done

But still an open issue . . .

- What kind of electrical standard for spare signals ?