Integration of the SAMPA in the SRS frontend

Status of the project for the SAMPA ASIC in the SRS ecosystem

Marco Bregant
IFUSP – São Paulo

RD51 MiniWeek

WG5 - Electronics for MPGD

10th February, 2020

SAMPA Design Specifications Summary

SAMPA is an ASIC developed for the readout of ALICE TPC and MCH detectors:

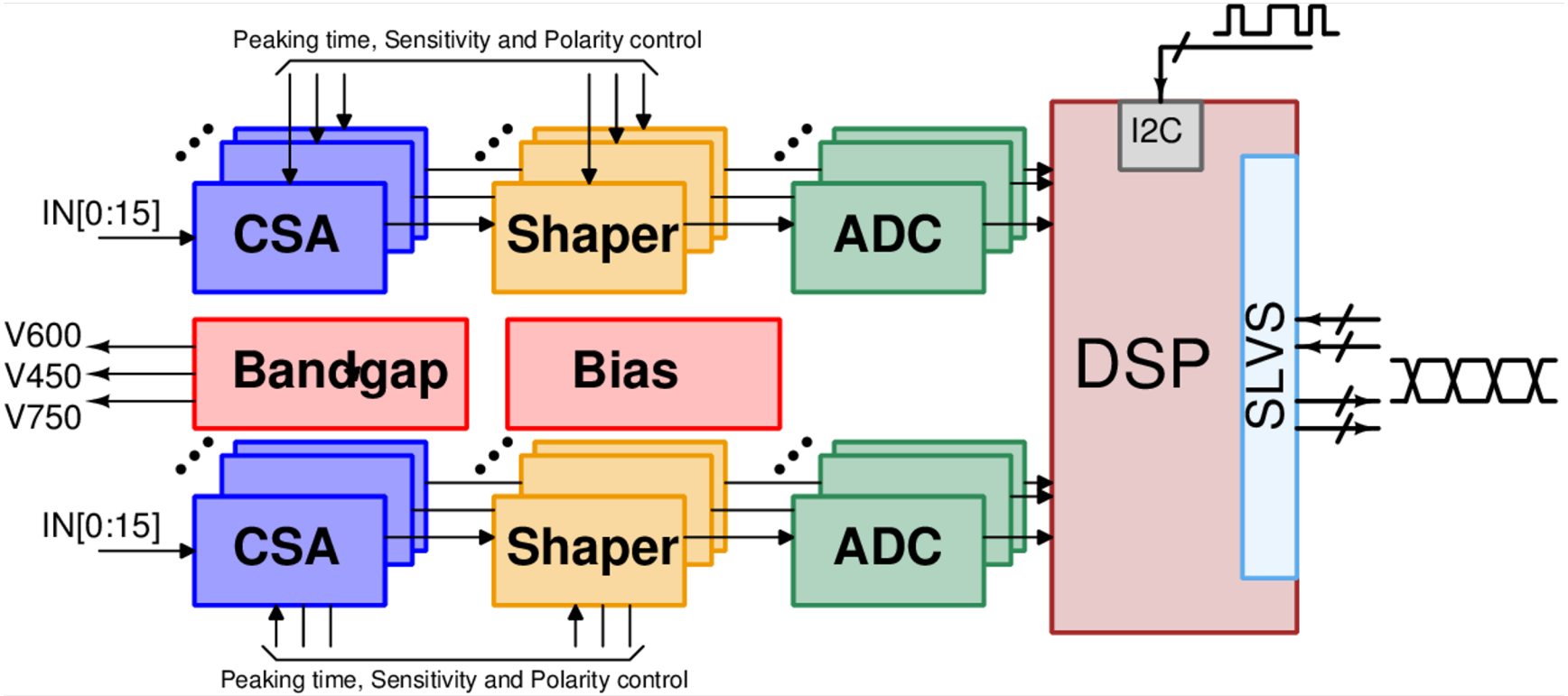
GEM MWPC

- TSMC CMOS 130 nm, 1.25V technology
- 32 channels, Front-end + ADC + DSP
- package size ≤15x15mm^2 (total footprint)
- ADC: 10-bit resolution, 10MS/s, ENOB>9.2
  (Alice TPC is eventually using: 5MS/s, to keep BW requirement in the readout chain lower)
- DSP functions: pedestal removal, baseline shift corrections, zero-suppression
- Data transmission: up to 11 e-link at 320 Mbps to GBTx, SLVS I/O
- Power < 32 mW/channel (Front End + ADC) V4, typical configuration, usually 20mW/ch or less.

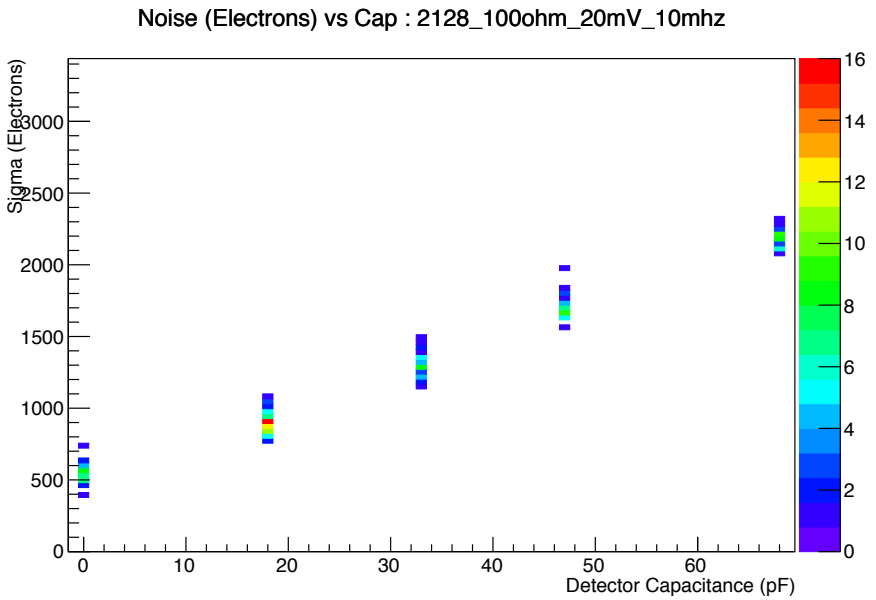
TPC Mode	MCH Mode
▪ Negative Input charge	▪ Positive input charge
▪ Sensor capacitance: 12 – 25 pF	▪ Sensor capacitance: 40–80 pF
▪ Sensitivity: 20mV/fC & 30mV/fC	▪ Sensitivity: 4mV/fC
▪ Noise: ENC ≤ 580 e$^-$ @ 18.5pF	▪ Noise: ENC ≤ 950 e- @ 40pF 1600 e- @80pF
▪ Peaking time: ~160 ns	▪ Peaking time: ~300 ns
▪ Baseline return: <500 ns	▪ Baseline return: <550 ns

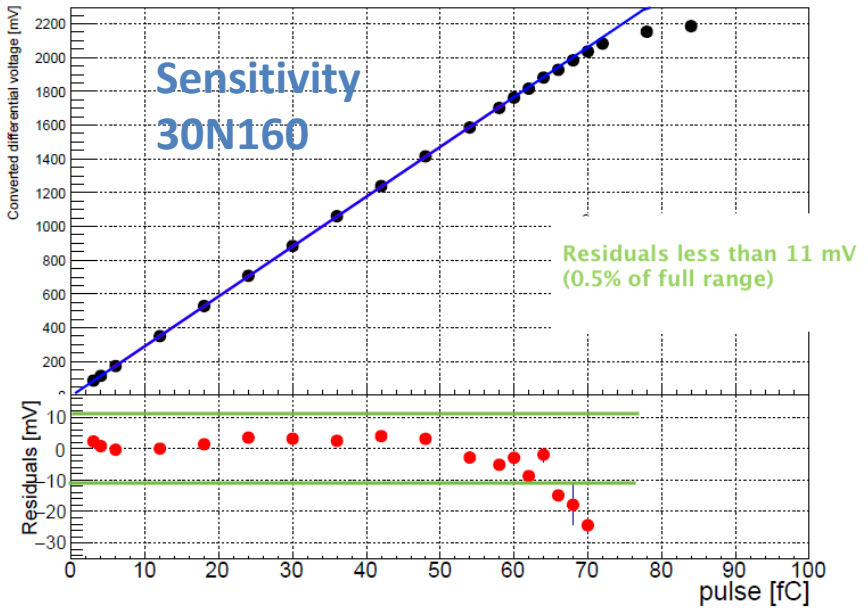
A modified version with 80/160 ns shaping, 20/30 mV/fC gain, 20MSps ADC, has been designed. First full-size prototype fabricated, we will receive and start testing it in a few weeks from now

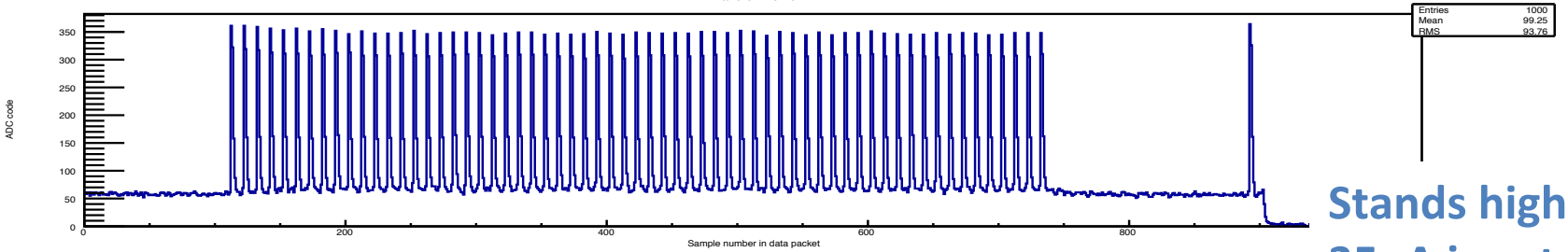
SAMPA Block Diagram

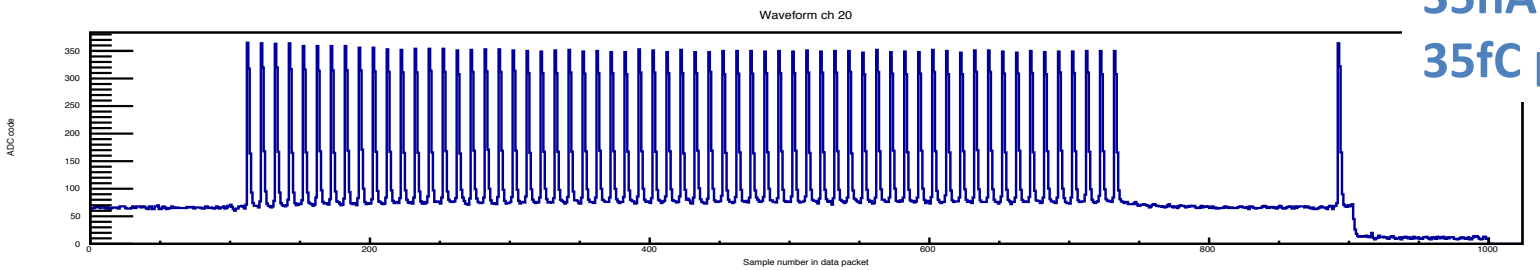


SAMPA works well









Stands high rate
35nA input current,
35fC pulses@1MHz

Functionalities overview

DSP

Top Level Functionality

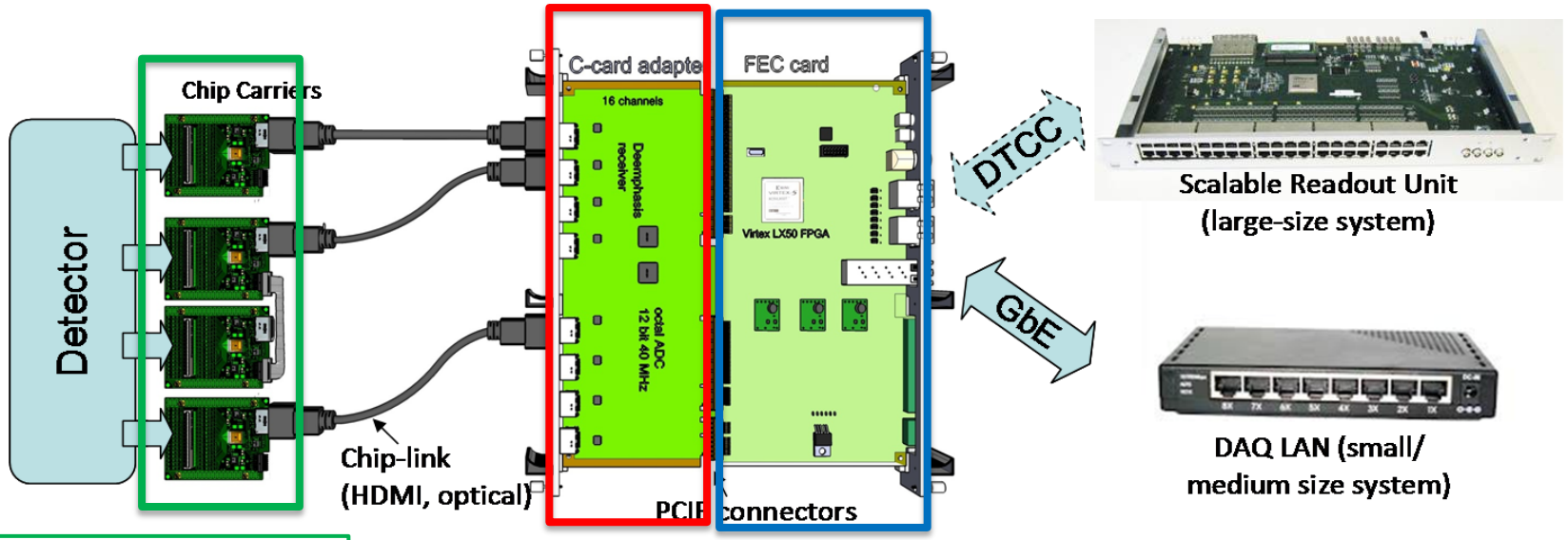
- 4 primary filter blocks
  - Individual correction per channel
  - Baseline correction
    - 1 FIR filter
    - 1 Slope based filter
    - 1 IIR filter
    - Lookup table correction(Pedestal Memory) f(t);f(din)
    - Conversion f(din)
    - Fixed correction
  - Tail cancellation
    - 1 IIR filter
- Configuration
  - Configurable through I2C
  - 1 global register unit, 32 sets of channel registers
- Radiation tolerant
  - TMR on almost all flip-flops
    - except on part of data path
  - Hamming protected headers
- Compression
  - Zero suppression with run length encoding
    - Forward linked list for easier decoding
  - Cluster sum
    - Uses zero suppression with run length encoding , but sums cluster into 20bit word
  - Huffman
    - Differential encoded data
    - Programmable table of codes for +17 to -17
    - Values outside table have special Huffman code prepended to raw 10bit value
- Design for test
  - JTAG boundary scan
  - Built in memory tester
  - Scan chain (on >98% of digital block flops)

Readout

- Selectable number of serial links up to 11
  - 320/160/80Mbps
  - Channels divided among links, no load sharing
  - Which channel goes to which link and in which order can be selected
  - Data is packet based (header + payload)
    - One packet per channel per event
- Event modes
  - Triggered
  - Continuous
  - Selectable event length up to 1024 samples
  - 192 pre-trigger samples
- Event buffer per channel
  - 6144(6K) words of compressed samples
  - 256 words of headers
  - Header still created if data memory goes full
- Daisy chain
  - Multiple devices can share a single serial link to readout unit
  - 2K word buffer in the receiving side
- Direct ADC serialization
  - Data serialized directly from ADC at 32xADC speed over 10 links
  - Raw data, no filtering, no headers
  - Sync pattern on startup, receiver should maintain sync after that
  - 2 modes
    - 10 bits is sent consecutively for channel 0-31 each 32xADC cycle
    - 5 lower bits, then 5 higher bits consecutively for channel 0-15 is sent on link 0-4 and for channel 16-31 on link 5-9
  - Clockgate the rest of the system to save power

The SRS idea

crate close to the experimental apparatus

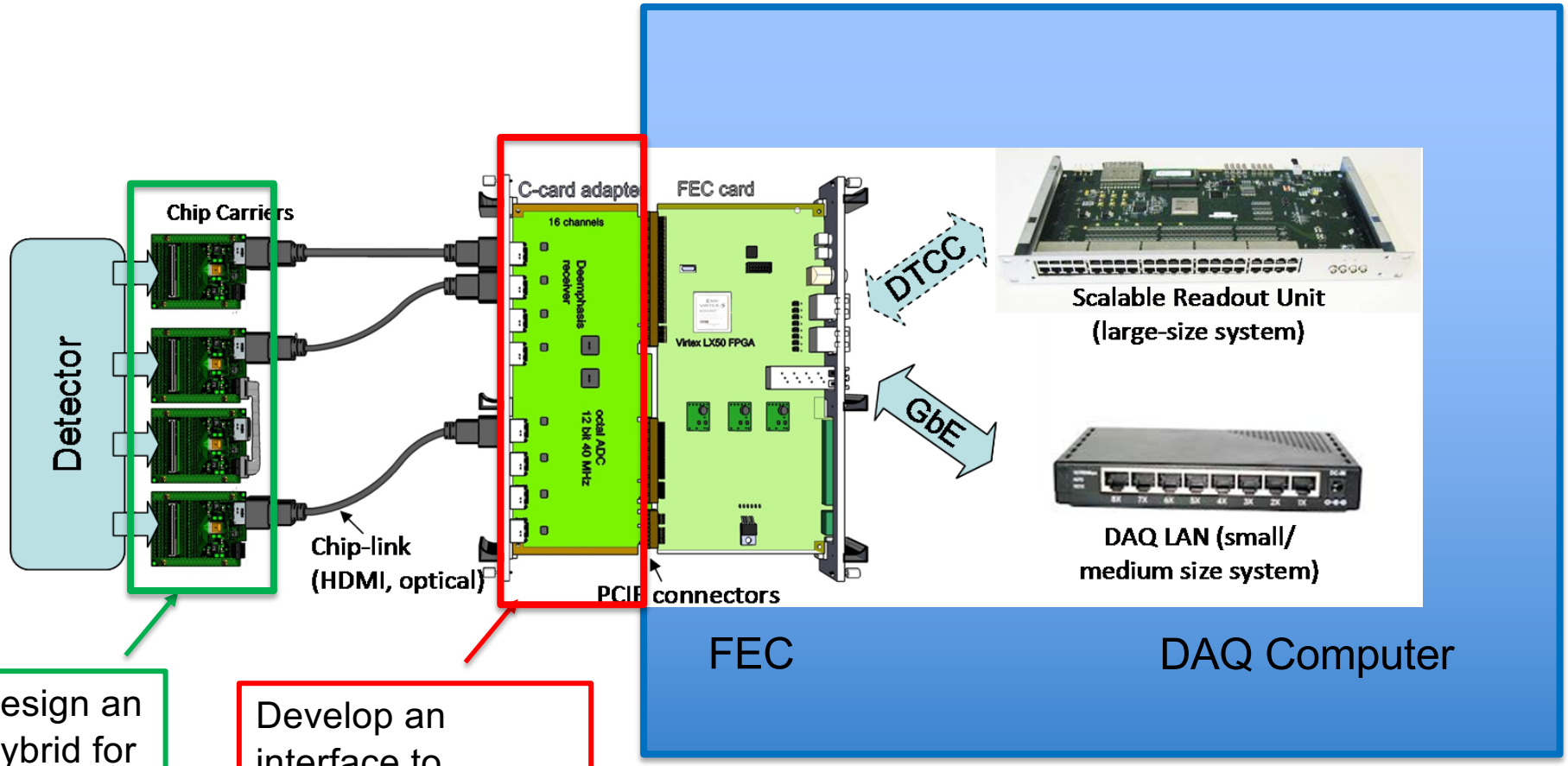


Front-end ASIC (close to detector)

An interface between ASIC and FEC card

FEC: handle the data taking of a set of ASIC. Handle the communication with the Readout Unit / DAQ

What's needed for SAMPA in SRS



Design an Hybrid for SAMPA

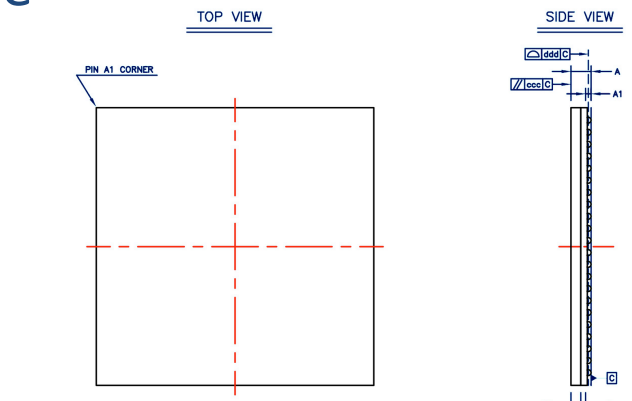
Develop an interface to manage the communication with SAMPA

KEEP/reuse, but FEC FPGA firmware

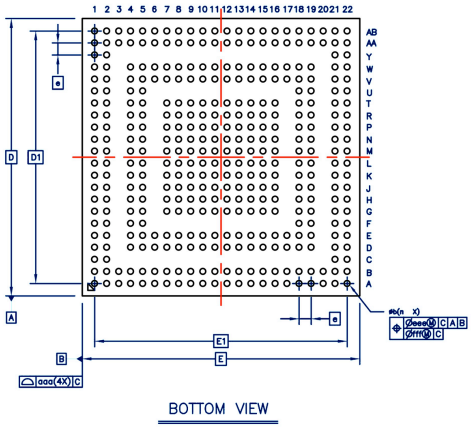
Acquisition software needs to be adapted?

SAMPA Package

- TFBGA package
- 15 mm x 15 mm body size
- 1.2 mm thickness
- 0.65 mm ball pitch.
- 372 balls
  - 4-substrate layers



		Symbol	Common Dimensions MIN.	NOM.	MAX.
Package :				TFBGA	
Body Size:	X	E		15.000	
	Y	D		15.000	
Ball Pitch :		e		0.650	
Total Thickness :		A	–	–	1.200
Mold Thickness :		M		0.530 Ref.	
Substrate Thickness :		S		0.360 Ref.	
Ball Diameter :				0.300	
Stand Off :		A1	0.160	–	0.260
Ball Width :		b	0.270	–	0.370
Package Edge Tolerance :		aaa		0.100	
Mold Parallelism :		ccc		0.100	
Coplanarity:		ddd		0.150	
Ball Offset (Package) :		eee		0.150	
Ball Offset (Ball) :		fff		0.080	
Ball Count :		n		372	
Edge Ball Center to Center :	X	E1		13.650	
	Y	D1		13.650	



ASE

TITLE: PACKAGE OUTLINE

372 L TFBGA 15.000x15.000x1.200

SCALE | PROJ.

DWG. NO. AAA14628 | REV. A

SHEET 1 OF 2 | SIZE A4

UNIT: MM | TOLERANCE: DIMENSION, ANGLE | REFERENCE DOCUMENT

(2019)

Our Planning for an SRS frontend

Slide from one year ago

- SAMPA designed locally, good knowledge, quick and full access to the original designers (important here, the digital ones)
- Two mains challenges
  1. Firmware for the FPGA linking SAMPA to SRS backend
  2. Design of high density, multilayer, low crosstalk (analog & analog-digital), etc., board to host 4 SAMPAs (to provide 128chs hybrid)
- First steps
  - Study both in simulations and with an assembly "SAMPA_testboard" <-> "FPGA developing board" the coupling and the communication btw SAMPA and target FPGA (w/o hybrid)
  - Start already design of hybrid board

FPGA firmware engineer already contracted;

Board design specialist candidate being evaluated right now.

Design strategy

- Hybrid compatible with present SRS standards
  - HRS input connector
  - Reading 128 Chs => 4 SAMPAs
  - "Slim" (two hybrids should be mounted side by side on the detector)
    - SAMPA is packaged in 15x15 mm^2 BGA, max 2 SAMPA side by side
    - Design the hybrid "almost" specular double face, top side reads even channels, bottom side reads odd channels
- Handling, at least partially, the SAMPA output bandwidth
- Avoiding FPGA in the hybrid
- Single "input" power connector, then each power domain served by dedicated LDO
    - SAMPA-analog
    - SAMPA-digital
    - SAMPA-ADC-reference
    - auxiliary circuits on the hybrid

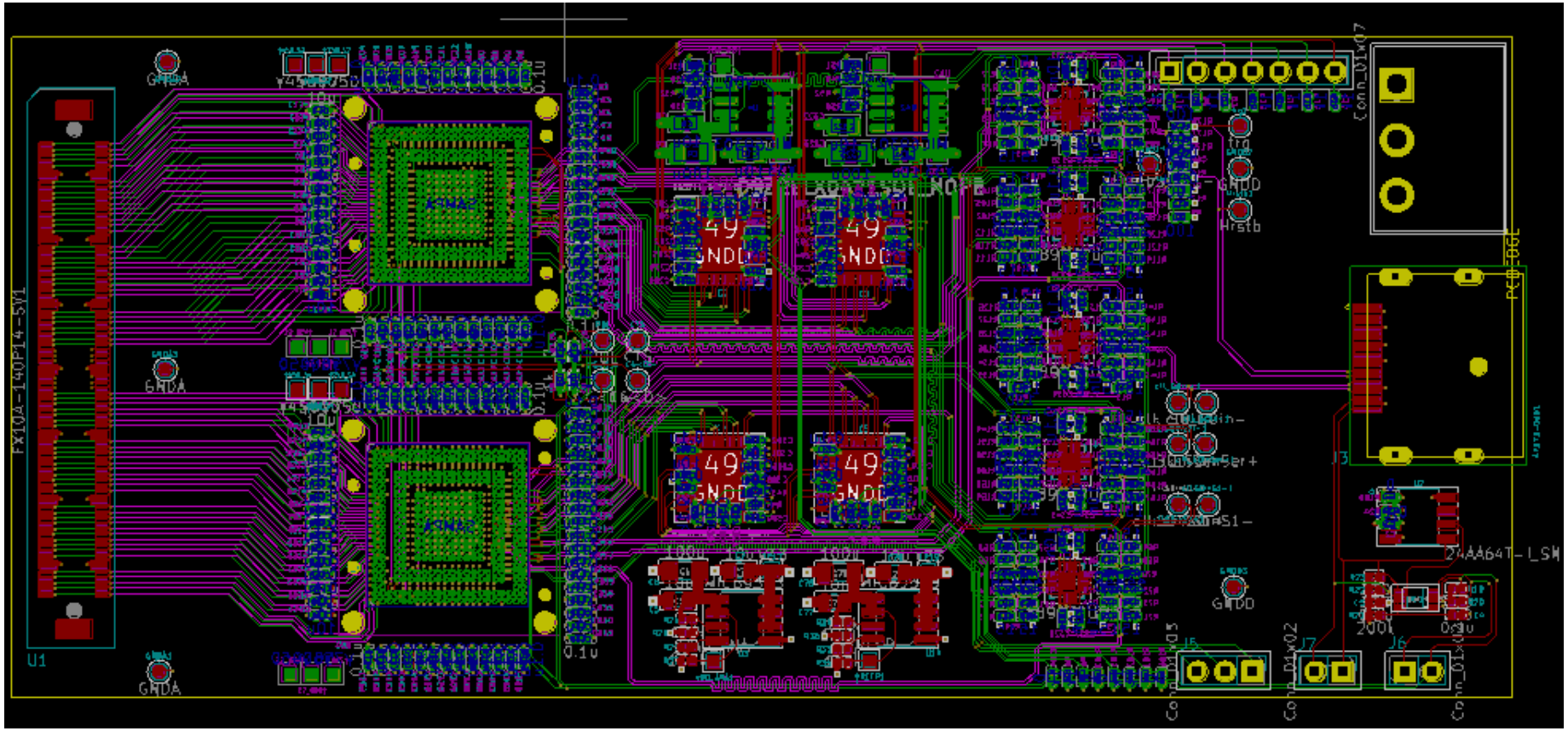
Output connector

- SRS uses HDMI cables, 19 pins
- Hybrid connects with adapter card in the crate, which is also ASIC specific
- Choose to go for DP port (20 pins), to route so all needed signals
- 4 DP main lanes used to transport output data
  - DP supports >2Gbps/lane already in it version 1.0 (so any DP cable does)
  - Each SAMPA has a dedicated lane
  - SAMPA uses up to 11 SLVS link at 320 Mbps
    - This signal cannot travel over long distances… should be converted in any case
  - Use a Serializer to convert 4 SLVS @320 Mbps to 1 LVDS@1280 Mbps
- Remaining DP pins used to transfer triggers and I2C signals

Development prototype physical layout

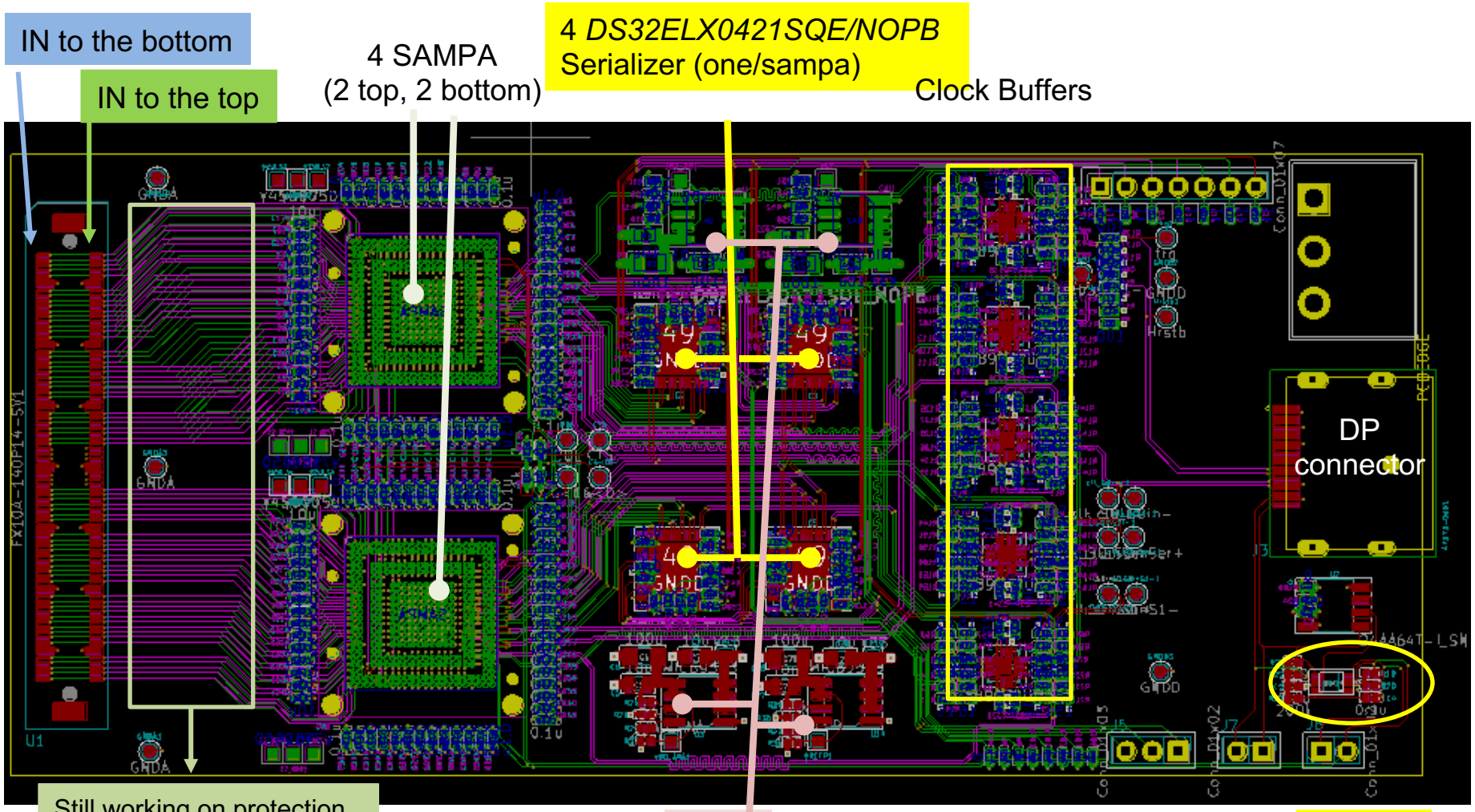
(almost final)

Compatible with the use of a socket to simplify the board production… and to allow “mistakes”



Development prototype physical layout

(almost final)



Present Status

- Hybrid card design, full prototype, almost ready
- Schematic of Interface card ready, physical layout work in progress
- FEE_v6 FPGA code studied in depth
- SAMPA testboard + adaption patch-board used to make a test-connection between one SAMPA and the FPGA.
  - Successfully provided control signals to SAMPA and sent the SAMPA data stream up to FPGA
- FEC FPGA firmware under development

Conclusions

- Designed a set “Hybrid + Adapter_card” to integrate SAMPA into SRS
- Fabrication of a prototype of the hybrid planed in some weeks from now.
- Prototype design quite close to the final goal (few minor compromises)
  - 128 channels hybrid
  - All control signals already routed via I2C (but back-up pins are still provided)
  - Not yet 50 mm wide to allow the placing of some extra test points (either pins or pads), and the use of a socket during the first debugging
  - Protection diodes network not yet included
- Debugging and extensive test of FPGA firmware with real object

I look forward to coming back with *(good!)* news in some months from now.