ECAL electronics upgrade meeting

Date: 8th-9th April 2009

Place: LAL - Orsay

Present: Jacques Lefrançois, Frederic Machefert, Christophe Beigbeder, Olivier Duarte, Dominique Breton, David Gascón, Carlos Abellan, Eduardo Picatoste.

1.1. Analog section

1.2. Specifications

In SLHCb, with an increased luminosity PMT gain has to be decreased by a factor 4-5 in order to avoid ageing (limit is 100 C). Main implication is that preamp input equivalent noise must be decreased.

A list of specs was presented, after correction they are summarized in the following table:

	Value	Comments
Energy range	0-10 GeV/c (ECAL) Transverse energy	1-3 Kphe / GeV Total energy
Calibration	4 fC /2.5 MeV / ADC cnt	4 fC input of FE card: assuming 25 Ω clipping at PMT base 12 fC / ADC count if no clipping
Dynamic range	4096-256=3840 cnts :12 bit	Enough? New physic req.? Pedestal variation? Should be enough
Noise	<≈1 ADC cnt or ENC < 5 -6 fC	**< 1 nV/√Hz ?**
Termination	50 ± 5 Ω	Passive vs. active
AC coupling	Needed	Low freq. (pick-up) noise
Baseline shift Prevention	Dynamic pedestal subtraction (also needed for LF pick-up)	How to compute baseline? Number of samples needed?
Max. peak current	4-5 mA over 25 Ω	50 pC in charge
Spill-over correction	Clipping	Residue level: 2 % ± 1 % ?
Spill-over noise	<< ADC cnt	Relevant after clipping?
Linearity	< 1%	
Crosstalk	< 0.5 %	
Timing	Individual (per channel)	PMT dependent

Definition of calibration in charge vs ADC count has been discussed:

- If no clipping maximal charge both at PMT output and at FE input is 50 pC. Therefore, 1 ADC count is 12,2 fC.
- With 25 Ω clipping only 1/3 of the current flows into FE card, thus 1 ADC count is 4 fC.

1.3.Ideas on low noise preamp with controlled input impedance

Some schemes of possible preamplifier options with controlled input impedance have been presented. The goal is achieving low noise (< 1 ADC count) keeping 25 Ω clipping impedance in PMT base. The way to achieve it consists on using a current mode amplifier (to be shown that it is better than a voltage mode input) with input impedance which is carefully determined without using 50 Ω resistor for termination (avoid thermal noise).

Preliminary simulations of 0T cable coupled amplifier (ATLAS LAr style) have been shown, using SiGe 0.35 μm CMOS AMS technology. It is a common gate amplifier with a double feedback to minimize noise and to control input impedance. The “super common gate” input configuration has been used also on the LHCb PS chip (without the second feedback loop).

Main concern is effect of parameter fluctuations:

- Local mismatch has been simulated and the effect is at 1% level, should not be a problem.
- Effect of parameter variations (gradients) across the wafer can not be simulated, although experience with LHCb Calo chips indicates that it is < 3% (for instance, tolerance of subtractor of SPD chip).
- Process variation (lot ot lot): has been simulated and the variations are large ± 15 %. Compensation to be studied:
    - Bias parameter. Tune a bias current or voltage controlling input impedance (1 value for the full production).
    - Use an external resistor (low value in series or large value in parallel: noise!)

Linearity has to be checked also for input impedance.

The circuit has to be simulated with a more realistic signal.

A differential version has to be studied. *Important in a complex card with reduced gain (input is more sensitive!) to prevent pick-up noise.*

Cable effect should be included in simulations (reflection).

Jacques pointed out how to compute the input current as function of the cable impedance (Rz) and the preamplifier input impedance (Ri). The current flowing into the preamplifier is:

$$I_o = \frac{2R_z I_{PMT}}{R_Z + R_i}$$

The voltage pulse travelling in the cable should be $V_C = I_{PMT} R_Z$, as the pulse entering the cable only “knows” about the cable impedance. Of course if clipping is used $V_C = I_{PMT} \frac{R_Z}{R_Z + R_{CLIP}}$. When the pulse arrives at the end of the cable it sees the load impedance (amplifier input impedance) and the output voltage depends on the reflection coefficient:

$$V_O = V_C + \Gamma V_C = V_C + \frac{R_i - R_Z}{R_Z + R_i} V_C = 2\frac{R_i}{R_Z + R_i} V_C$$

Then,

$$I_O = \frac{V_O}{R_i} = 2\frac{1}{R_Z + R_i} V_C = \frac{2 I_{PMT} R_Z}{R_Z + R_i}$$

Using that input impedance is $R_i \square \frac{1/g_{m1}}{G} + R_{C1}\frac{R_1}{R_1+R_2} \square R_{C1}\frac{R_1}{R_1+R_2}$, the output of the amplifier is,

$$V_{O\_T0} = \frac{2I_{PMT}R_Z R_{C1}}{R_Z + R_{C1}\frac{R_1}{R_1+R_2}}$$

1.4. Discrete component solution

Carlos has presented Pspice simulation of the analogue solution based on Op Amps and first room computations.

Some issues to be improved:

- Input stage should be modified (capacitor in inverting input).
- Integrator time constant should be increased to 5 µs at least.
- Delayed signal is connected to the non-inverted input of the integrator Op Amp, and thus it is also seen to the output. An additional or a fully differential Op Amp is needed.
- Noise simulations should be added.

Anatoli also sent a talk describing a discrete component solution. Simulation show good behavior, including noise computations. A 50 Ω resistor in PMT base helps to minimize the effect of reflected pulses, although it is not realistic to inlclude it (difficult to add), nevertheless this solution should work without this resistor.

1.5.Effect of cable on noise

Edu presented basic concepts on the analysis of the effect of cable on noise. The skin effect resistor should be < 1 Ω/m even for a center frequency 50 MHz. Thus, direct thermal noise of the cable does not seem to be problematic.

However, complete evaluation of the effect should take into account the cable transfer function and the load and source impedance. Calculations are complicated. To be understood.

1.6. Semi-Gaussian Shaping

Semi-Gaussian shaping might be an alternative for integrated implementation if clipping is removed from PMT base. Pole-zero cancellation is needed to avoid undershoot. Fast amplifiers are needed to achieve few ns time constant.

Simulations should be repeated with more realistic signal shape.

1.7.Offset and pedestal

Extrapolating the offset of the SPD chip (0.8 mV rms referred to the input) to the gain of the ECAL for SLHCb the random offset (device mismatch) should be about 30 mV rms, 6σ are about 180 mV (12 % dynamic range assuming 1,5 V).

Offset in PS chip is probably smaller. To be studied if a current mode input could help to minimize offset.

Offset should be minimized:

- Technology and layout: perhaps possible to improve by factor 2.
- Circuit design (current mode input a la PS might help?)
- AC coupling, between last amplifier and ADC.
- Digital trimming.

No effect of integrator switches in offset has been observed.

1.8.Tasks for next meeting

Next meeting will probably take place in Barcelona on mid May. Some tasks have to be performed, I attach some tentative names:

a) Noise analysis of current and voltage mode input amplifiers (David)

b) Effect of cable on noise (Edu)

c) Discrete solution with OpAmps (Carlos)

d) Gaussian shaping of a non-clipped PM signal (David)

e) ADC interface: dynamic range and offset (Edu/David)

1.9. Digital section

1.10. First tests of implementing present AX firmware into a ProAsic3 (Olivier)

Olivier recalled the present functionalities implemented in the AX fpga of the present version of the front-end boards. There are mainly 4 such functionalities :

- ADC data resynchronisation and pedestal subtraction

- Trigger data calculation from the 12 bits ADC values and taking into account the desired calibration constant

- The third block consists of a latency and a derandomizer in order to wait for the L0-decision and to send the data channel by channel at 40MHz to the sequencer fpga in case of L0-yes decision.

- The last block allows to send test patterns in the front-end pga (FEPGA) processing or to control the injection of pulses at the input of the front-end board.

The software used to make the firmware of the fpga is ACTEL Libero which is dedicated and fully integrated (synthesis, simulation and routing).

The debugging of the code with the ProAsic family is done by the Identify tool. This is not so performant as the ACTEL Silicon Explorer which allows to “see” inside the chip at almost any position without modification of its firmware (this is an antifuse component, so the code cannot be changed). The identify tool permits to do some debugging inside the flash devices, but it is more limited in the sense that the fimware has to be recompiled and the re-routed at least partially. Identify uses the logic and RAM blocks of the chip. This is a severe limitation and we may expect not to be able to debug some problems with it. Still, it could be worthwill to have it.

Problems linked to the routing may not be understood thanks to identify. A good usage of the timing analysis tools could help in debugging.

A comparison of the architectures of the AX and the ProAsic was shown. Here is the present usage of the AX on the front-end boards :

- 89% of the R-Cells (Sequentials)
- 77% of the C-Cells (Combinatoric)
- 81% of the Logic (R+C-Cells)
- All the RAM have been used (12 banks)
- 148 I/O out of 248 are used
- 3 routed clocks have been cabled and 2 PLL out of 8 are used.

The combinatorial operations possible with the "versatile" basic cells of the ProAsic3 are less powerful than the combinatoric cells of the AX. Nevertheless, the implementation of the AX firmware into the ProAsic3 A3PE600 gives the following utilisation :

- 39% of the versatile cells used
- 147 I/O out of 270
- 2 PLL out of 6 are used as for the AX
- 16 RAM banks are used out of 24 available

But, the plan is to have 8 channels per fpga and not only 4 as was done up to now. Going to 8 channels leads to the following estimation of the needed resources :

	Current FEPGA (4 channels) AX FPGA	Futur FEPGA (8 channels) PA3 family ?	Futur FEPGA (8 channels) AX : backup solution
I/Os	148	~260 (see below)	317 (FG 484) 336 (FG 676)
I/O banks	8	8 (4 seems too few considering the different bank usages: GBT, ADC, slow control, etc...)	8
RAM Blocks	12 (Latency+Derandomizer)	~26	16
Cells	3430 (R+C cells)	~11000 versatiles	8064 (R+C cells)

PLL	8 : 2 are used	1 ?	8

The I/O needed are :

- 96 ADC inputs
- 32 trigger outputs (8 channels times 8 bits at 80MHz)
- 48 I/O for the neighbours (6 neighbours times 8 bits)
- less than 60 outputs for the GBT (depends on the data packing)
- 21 I/O for slow control, clock, channelB, etc...

and makes a total of less than 260 I/O.

The packing requires fraction of the RAM to create the packed events lists. We will have to share the RAM between the packing and the test pattern capabilities.

Among the foreseen ProAsic target,

- the A3P1000 has too few I/O banks (only 4)
- the A3PE3000 looks oversized
    - the possibility to implement 16 channels in it was thought of but would lead to PCB conception difficulties. Moreover, the number of I/O being almost twice bigger, the mentioned problem of bit flipping and current comsumption could be more acute.
- The A3PE1500 model looks to be a good candidat:
    - 38400 versatile cells (apart from packing, ~12000 cells should be necessary)
    - 60 block of 4.6kbits
    - 8 I/O banks
    - from 280 up to 444 I/O

There are two possible types of classes of ProAsic : the E class and the L one. The L ProAsic are low-power components. Unfortunately, all the E-type ProAsic cannot always found amoung the L-types. Hence, there is A3PL1000 and an A3PL3000 but no A3PL1500.

The compilation inside an A3PE1500 was done and leads to the following occupancy (4 channels !) :

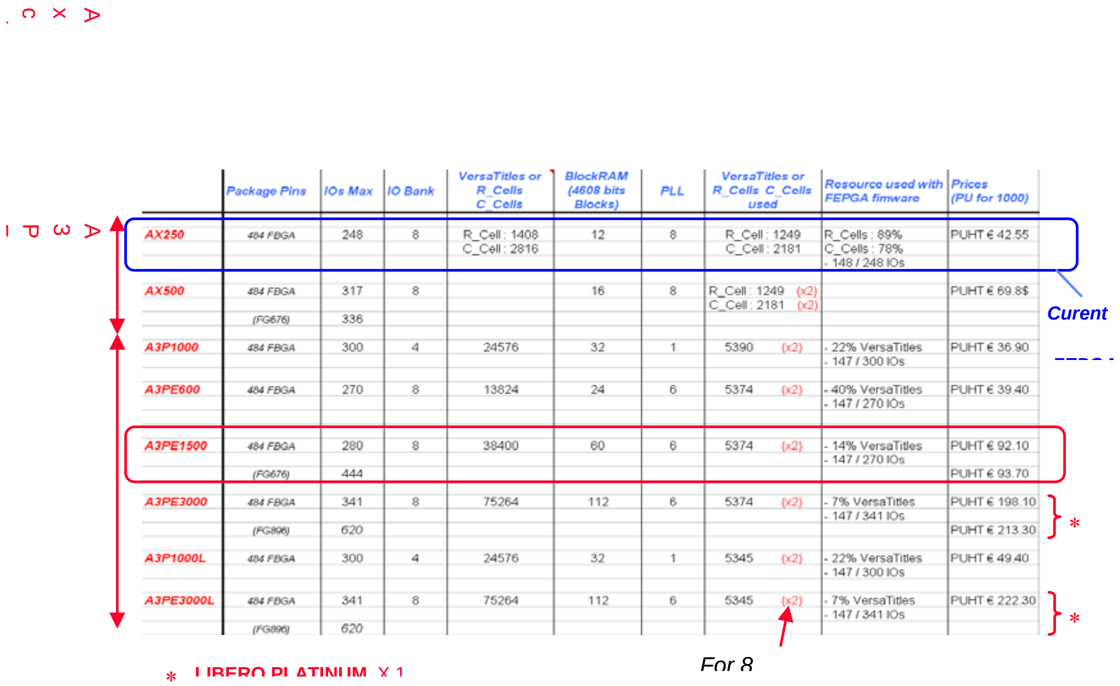
1. 14 % versatile cells

2. 147 I/O (~52%)

3. 16 RAM out of 60

Migration between components inside the ProAsic3 family was looked at. This seems possible from higher to lower density chips. Moreover, migration between A3PE and A3PL is feasible apart from a flash freeze pin.

The figure below presents a summary of the resources available, needed and of the component prices.


A x c .
A 3 P l
Curent
FEPGA
For 8

	Package Pins	IOs Max	IO Bank	VersaTitles or R_Cells C_Cells	BlockRAM (4608 bits Blocks)	PLL	VersaTitles or R_Cells C_Cells used	Resource used with FEPGA fimware	Prices (PU for 1000)
AX250	484 FBGA	248	8	R_Cell : 1408 C_Cell : 2816	12	8	R_Cell : 1249 C_Cell : 2181	R_Cells : 89% C_Cells : 78% - 148 / 248 IOs	PUHT € 42.55
AX500	484 FBGA	317	8		16	8	R_Cell : 1249 (x2) C_Cell : 2181 (x2)		PUHT € 69.8$
	(FG676)	336							
A3P1000	484 FBGA	300	4	24576	32	1	5390 (x2)	- 22% VersaTitles - 147 / 300 IOs	PUHT € 36.90
A3PE600	484 FBGA	270	8	13824	24	6	5374 (x2)	- 40% VersaTitles - 147 / 270 IOs	PUHT € 39.40
A3PE1500	484 FBGA	280	8	38400	60	6	5374 (x2)	- 14% VersaTitles - 147 / 270 IOs	PUHT € 92.10
	(FG676)	444							PUHT € 93.70
A3PE3000	484 FBGA	341	8	75264	112	6	5374 (x2)	- 7% VersaTitles - 147 / 341 IOs	PUHT € 198.10
	(FG896)	620							PUHT € 213.30
A3P1000L	484 FBGA	300	4	24576	32	1	5345 (x2)	- 22% VersaTitles - 147 / 300 IOs	PUHT € 49.40
A3PE3000L	484 FBGA	341	8	75264	112	6	5345 (x2)	- 7% VersaTitles - 147 / 341 IOs	PUHT € 222.30
	(FG896)	620							

* LIBERO PLATINUM X 1

1.11. Data Compression (Jacques)

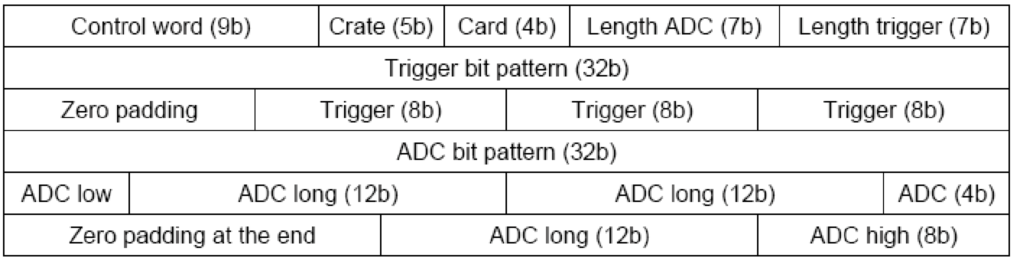
The second topic of the digital session concerned the data transmission .

The data to be sent were currently 20 bits (12 ADC, 8 trigger and the anti-parity) on 34 words (Header, control word and 32 channels) followed by two compulsory empty words (event separation) out of the 36 word budget.

In the future, only the 12 ADC bit values will be kept for the 32 channels. The result of the trigger calculations (8 bit maximum, 5 bit address of the maximum, sum of the channels on 8 bits) will be sent to the DAQ. Adding the BXID (8 bits), a total of 29 bits have to be added to the 12x32 others. 413 bits should be sent to which some (anti-)parity bits to be included.

The requested bandwidth has a price : roughly 200 € per optical link (emitter). At the reception, the new TELL40 could cost up to 20k€. Considering that it supports up to 120 fibers (see J-P Cachemiche presentation in electronics upgrade meeting in february 2009), the total price of a link is rather 400€. Notice that gathering so many fibers to a single board looks very ambitious although we could now think of perform 2D zero suppression at the level of the TELL40.

The interest in reducing the number of fibers looks clear having those figures in mind. The idea is thus to perform a data compression or packing of the data in order to reduce the needed bandwidth. This is what is done on the TELL1 to send the data to the present DAQ. The following graph shows what is done on the TELL1.



<table>
<tr><td colspan="3">Control word (9b)</td><td colspan="2">Crate (5b)</td><td colspan="2">Card (4b)</td><td colspan="2">Length ADC (7b)</td><td colspan="2">Length trigger (7b)</td></tr>
<tr><td colspan="11">Trigger bit pattern (32b)</td></tr>
<tr><td colspan="2">Zero padding</td><td colspan="4">Trigger (8b)</td><td colspan="2">Trigger (8b)</td><td colspan="3">Trigger (8b)</td></tr>
<tr><td colspan="11">ADC bit pattern (32b)</td></tr>
<tr><td>ADC low</td><td colspan="5">ADC long (12b)</td><td colspan="4">ADC long (12b)</td><td>ADC (4b)</td></tr>
<tr><td colspan="4">Zero padding at the end</td><td colspan="4">ADC long (12b)</td><td colspan="3">ADC high (8b)</td></tr>
</table>

Figure 14 : ECAL and HCAL data format

The minimal length in byte is then 4 (header) + 0 (trigger) + 4 (ADC pattern) + 16 (32*4 bits ADC) = 24 bytes. The maximal length is 4 + 36 (trigger) + 52 (ADC) = 92 bytes. The ADC length field varies from 20 to 52, the trigger length field from 0 to 36.

The idea is to keep the original ADC value on 12 bits if it is between 248 and 263. Otherwise, 248 is subtracted to the value which is sent on a 4 bit format. The format used per channel is indicated in the “ADC bit pattern” made of 32 bits (one per channel).

A fiber sends 80 bits@40MHz. 413 bits being sent, 6 fibers are needed, which would break the nice correspondence between a FEPGA and an optical link. With the margin some new parameters could be sent (crate or card Id, anti-parity, etc...) possibly on each link.

Taking into account a data compression analog to the present TELL1’s, the minimum number of bits to be sent is 181 bits (32 for bit pattern, 4 bits per ADC, 21 bits of trigger) + BXID, i.e. 3 fibers. In those three fibers a maximum of 6 ADC values could be send with the large format (12 bits) and a few remaining bits could be used for the extra info (BXId, crate and card id). Above that threshold of 6 channels, data would be lost/corrupted. Another scheme could consist in reducing the number of long format channels to 4, extending the number of extra info. But, the risk of losing data would be increased.

Having 4 fibers instead of 3 would limit this risk and restore the FEPGA – optical link correspondence... at a total cost of 100k€ !

Packing is clearly an advantage but is difficult to perform in an FPGA at 40MHz. Several schemes have been mentioned :

- Sending serialised data containing for each event a header with the BXID, the data length and the data quality. The data are stored in a derandomizer whose status (empty or full) is recorded in the data quality bit. The data are sent with variable length after the header and sequentially. Periodically, the buffer should be empty (LHC cycle), which can be checked. Moreover, the buffer may also be cleared at that time.

- Taking the hypothesis of 4 fibers, each one would send the data of a specific FEPGA. Hence they are independent of each other and optical links could be desynchronised. A resynchronisation could be done periodically (LHC cycle).

- The synchonisation could be alternatively done by filling a buffer with zeros for “missing” data. The buffer would be synchronized automatically when the buffer is full.

- Another possibility imagined by Jacques would be to store the data in two parallel tables. The first one would contain the fixed format data (header, bit pattern), the second and parallel table would be filled sequentially and contain variable format data. The two tables could become de-synchronised along with time. When the maximum table length is reached, the data are sent. In the meantime, the storage and compression would be done in a second group of two parallel tables.

Several possibilities appeared but no clear solution emerged. The technical implementation of the previous schemes looked either difficult or impossible in our FEPGA. This should be clarified at the next meeting.

1.12. Tasks for next meeting

Here is a list of things to be done for mid-May (next meeting)

- Detailed list of tests to be implemented on the digital PCB prototype
  - FPGA data exchanges
  - Interface between Digital prototype and analog mezzanine
- Understanding of the data compression

The schematics of the PCB should be provided to the LAL-CAO by the end of June.