Data transmitted

- **Now from FEPGA to SEQPGA to CROC**
  - **12bits of ADC +8 bits trigger+1 bit antiparity**
  - **36 serialized data :**
    - **2 zero words+ 1Header word BXID EvtID**
    - **1 control word (derandomiser empty, clock parity, calibration data bit …)**
    - **32 words 21 bits (ADC data + trigger + antiparity)**
- **New Card from FEPGA to GBT**
  - **12 bits ADC X32**
  - **Suggest to keep only the result of TrigPGA**
    - **8 bits Max value**
    - **5 bits address of Max value**
    - **8 bits sum of 32 channels x8bits**
    - **8bits BXID**
  - **Result 32X12 +29 = 413 + some antiparity bits?**

Cost vs number of optical fiber link

- **200 euros per fiber link**
- **A Tell 40 evaluation is 20Keuros for about 100-120 fibers from J-P Cachemiche**
  - **http://indico.cern.ch/getFile.py/access?contribId=6&resId=0&materialId=slides&confId=48992**
  - **Electronic upgrade meeting February 2nd 2009**
  - **Optimistic?**
- **Total cost about 400 euros /fiber**
- **If enough fibers and enough data in one Tell40, could one do 2D zero suppress in Tell 40: 2 crates => about 28 cards =>84-112 fibers???**

Packing in Tell1

- **August 21 2005 presentation by Bolek**
- **CALO readout note EDMS 527942 May 9th 2006**

Control word (9b)	Crate (5b)	Card (4b)	Length ADC (7b)	Length trigger (7b)
Trigger bit pattern (32b)				
Zero padding	Trigger (8b)	Trigger (8b)	Trigger (8b)	
ADC bit pattern (32b)				
ADC low	ADC long (12b)	ADC long (12b)	ADC (4b)	
Zero padding at the end	ADC long (12b)	ADC high (8b)		

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.

Present Decision 4-12 bits in Tell1

- **Part of packing is for trigger data dropped in the upgrade.**
- **Before packing average ADC value is 256.5**
- **If the ADC value is between 248 and 263 =>subtract 248 and get 4bit value**
- **Otherwise give original 12bits**

Scheme without compression

- **384 +29 bits minimum =413 need more than 5X80 need 6 fibers =>480bits**
- **Use extra space for Card and crate address + antiparity? =10 bits**
- **Still space left => optimize correspondence 4FPGA => 6Fibers each FPGA => 6 Fibers like 2bits/12bitADC in each fiber => a bit ugly!!!**
- **Use BXID in each Fiber? Use crate and Fiber address in each Fiber? Antiparity in each Fiber?**

II)With Compression

- **Minimum data per card = 181 bits +BXID at least 3 Fibers**
  - **32 bits for 4-12 bits pattern**
  - **21 bits trigger information + BXID**
  - **32 x 4 bits minimum of ADC**
- **Maximum average data/card in 3 Fibers= 240 bits**
  - **181 bits (minimum above)**
  - **6 out of 32 ,long ADC data =>6X8 =48 (How ?6 is not divided by 4FPGA)**
  - **11 bits left 3 antiparity + 8 bits BXID? Card&Crate address?**
  - **Or 4/32 long ADC data 4X8 bits =32**
  - **27 bits for address BXID antiparity**

Choice of scheme in fibers (I)

- **"general LHCb scheme" (as I understand it)**
  - **Store N event data in derandomiser buffer (for each fiber?)**
  - **For each event => in each fiber send header with: BXID +length of data+data quality.**
  - **Then send data of variable length (address+data in zero suppress VELO)**
  - **if data buffer becomes full => warn by data quality**
  - **Every LHC cycle (3564 X 25 ns) at the end of empty BX check that data buffer is empty or zero it.**
  - **Exploit possibility to centrally block data input to data buffer? (channel-B?)**
- **Implementation in calo**
  - **split 21 bits (+xx) trigger-info in the 4 FPGA add to header**
  - **One FPGA sends data to one fiber => need 4Fibers/card => more confortable. But more expensive by 100Keuro!**

Choice of scheme in fibers (II)

- **In previous scheme the 4 fibers data are "independent" and can be desynchronized by an arbitrary amount (resynchronization every 3564 BX)**
- **Another possibility would be to complete a certain buffer length (256?512?) with zeros so that events are synchronized periodically**
- **Another event architecture, in a similar spirit could be to have a synchro part of event only 8bits for large ADC are non-Synchro: in each FPGA (IF 4FPGA=>3 Fibers one FPGA=>60 bits)**
  - **8bits =1/4 of trigger**
  - **8bits map short long ADC**
  - **Xbits BXID (4?)**
  - **8X4=32 bits minimum of ADC**
  - **8bits one long ADC (desynchro but resynchro every 256-512 BX)**
  - **if 4fibers 20 more bits available =>16bits two long ADC 4 bit address?**

Conclusion:choice

- One choice is 3 fibers vs 4 fibers
  - 3 fibers may be impossible at 10**33 or 2X10**33
  - Need a Monte Carlo test (test also a switch short long with 5bits-12bits)
- Even if the choice is 4 fibers on needs to choose between "semi permanent" pattern with only 4X8 or 4X16 bits per event desynchronized the system must resynchro every 256 or 512
- or the "VELO-like" pattern with serialized data . Here also it is possible to fill with 0 and resynchronize every 256 or 512 events
- So 3 choices: 3-4fibers, resynchro or not, "semi-permanent" or VELO-like" but not all 8 combinations are possible (only 4?)
- Some implementation (fixed format) seem easier than other in FPGA! =>example at blackboard!!!