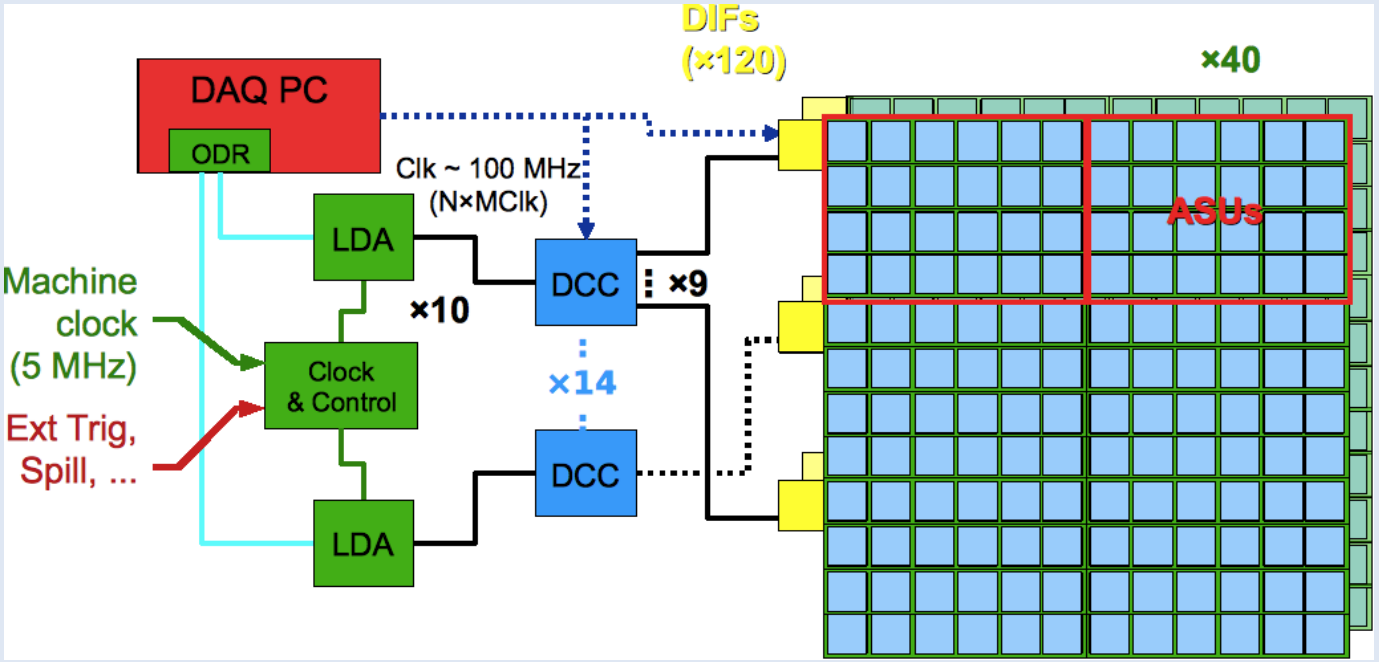
Proposal for a LCIO format for the DHCALs

Vincent Boudry
Rémi Cornat
David Decotigny
LLR
Gérald Grenier
Robert Kieffer
IPNL

EUDET annual meeting
U. of Geneva
19/10/2009

Raw data

- **Starting point**: store the EUDET DAQ2 raw data originating from a DHCAL using **ROC chips** providing **frame** *(≠ ROC ⇔ ≠ frame length)*
- Keep the frame order for each chip and DIF (and other intermediates cards) → counters, error & malfunctions tagging, full history
  - Train by train
  - 1 train ≡ a complete set of auto-triggered frames from one ASIC, readout on an external signal [trigger, DAQ stop, RAM full]
  - 2 Acq modes
    - ILC like
      - Trig = Spill start
    - Triggered
      - Ext. Trig
      - RAM full manag[t]



RAW data proposal (suitable for ROCs)

V. Boudry, R. Cornat,
D. Decotigny

LCIO format (source):
https://svn.in2p3.fr/calice/online-sw/trunk/daq/reassembler/lcio_dump.hpp?view=markup

Physical address (⊃ redundancy)

Event Counter/Tagging

Event Timing

Compacity vs Redundancy (debugging)

```
* Event:
*  - DetectorName <- runHdr::DetectorName
*  - RunNumber    <- runHdr::RunNumber
*  - EventNumber  <- ROEventId::toUnsigned()
*  - TimeStamp: default
*  - Weight: default
*  - LCCollection "rawdata_DIF:str(DIF_id)":
*         + param "id_ODR" (unsigned): from raw event
*         + param "id_LDA" (u32b)
*         + param "id_LDA_diflink" (unsigned)
*         + param "id_DIF" (unsigned)
*         + param "TrainNumber" (unsigned)      // since DAQ start
*         + param "BC_DIF" (unsigned)           // since RUN start
*         + param "DeltaCounter" (unsigned)     // 40 MHz counter
*         + param optional extra DIF info (type, temp, adc) ???
*         + elt[0]: LCGenericObject
*              + int[0]: id_roc_chain
*              + int[1]: id_roc
*              + int[2]: index_in_dif_dump (starts at 0)
*              + int[3]: chipType
*              + int[4]: acqMode
*              + For HR2 chips:
*                 + int[5]: nframes (< 128)
*                 + int[6 + i*5]: bcid for frame i (i in [0, nframes])
*                 + int[7 and 8 + i*5]: t0 vector (msb 7=t0_63, lsb 8=t0_0)
*                 + int[9 and 10 + i*5]: t1 vector (msb 9=t0_63, lsb 10=t0_0)
*                 + int[11 + i *5...]: additional data ??
*        + elt[1]: other LCGenericObject (ROC event)
*           ...
*  - other LCCollection "rawdata_DIF:str(DIF_id)":
*    ....
```

```
namespace IO
{
  class LCWriter;
};
/**
 *
 * Run Header:
 *   - RunNumber: user-specified
 *   - DetectorName: user-specified
 *   - Description: user-specified
 *   - ActiveSubdetectors: [ "DIF_rawdata" ]
 *
```

RAW Data ”in the tubes” format:
https://svn.in2p3.fr/calice/online-sw/trunk/daq/calice_packets/calice_raw_formats.h?view=markup

LCIO format for reconstructed DHCAL Header & DIF part (proposal v0.01)

back of envelop discussion
V. Boudry, G. Grenier, R. Kieffer
+ Lei Xia (US DHCAL)

```
EVENT::LCEvent
{
    int  _runNumber;          //
    int  _eventNumber;        // Global Trig Count == # evt
    EVENT::long64 _timeStamp; // Mean timestamp from Hit/DIF
                              // TB: Large BC
                              // For ILC: Large BC since run start.
    std::string   _detectorName; // Detector version ??
    LCCollectionMap    _colMap; // List of collections
    std::vector< std::string >  _colNames; // Name of collections
    LCParametersImpl   _params; //
    LCCollectionSet    _notOwned; //
}

EVENT::LCGenericObject
{
// ->DIF Collection
//    - int DIF_ID+Module_ID        // DIF_ID (48-144 ==> 7-8b)
//                                  // + Module_ID (40 barrel + 24 Endcap ==> 6b)
//    - int ATC                     // delta GTC / DTC
//    - int DTC                     // Diff Trig Counter == #evt vu par la DIF
//(    - float TimeDiff              // Time2Previous event )
//    - int LargeBC                 // BC from run start.(32b)
}
```

LCIO format for reconstructed DHCAL Calorimeter hits (proposal v0.01)

ILD and TB compatible

```
EVENT::RawCalorimeterHit
{
    int  _cellID0;          // Chan (64 ==> 6b)
                            // + Asic (max 420 ==> 9-10b)
                            // + Dif_Id (48-144 ==> 7-8b)
                            // + Module_Id (40 Barrel + 24 Endcap ==> 6b)
                            // == 28-30b (remain 2-4b)  [6+6+7 == 19b in TB]
     int _cellID1;          // Time2Previous (in BC) ==> 24b (remain 8b) (CHBIT_ID1 must be set)
    int  _amplitude;        // 3 Thr ==> 2b (remains 30)
    int  _timeStamp;        // Rec Time on 32b wrt (Spill start | Ext. Trigger)
}

EVENT::CalorimeterHit
// Recontructed Hits
{
    int  _cellID0;           // Idem RAW
    int  _cellID1;           // Idem RAW (CHBIT_ID1 must be set)
    float    _energy;        // Rec Energy
    float    _time;          // time from ref (in ns).(LCIO::RCHBIT_TIME must be set)
    (float   _position [3]);   // Position (unit not fixed) (LCIO.CHBIT_LONG must be set)
    int  _type;              // Deposit type (mip, EM, noise, ...)
    EVENT::LCObject *  _rawHit; // Link to RAW hit
}

One also needs the mapping functions:    int[3] GetIJK(cellID);
                                         float[3] GetXYZ(cellID);

Error on Energy => to be recalculated, or integrated to energy.
```

Comments ?

- Very first draft
  - to be discussed...
  - being implemented for SDHCAL offline storage
- to be implemented for the DAQ2
  - Needs to be finalised before data taking
  - Once validated → modification of LCIO ?
- Additional point for any DAQ:
  - Improvement of online writing
    - Experience from EUDAQ:
      - “Difficulty for keeping a LCIO file open for a long time”
  - Flush + easy crash recovery (?)