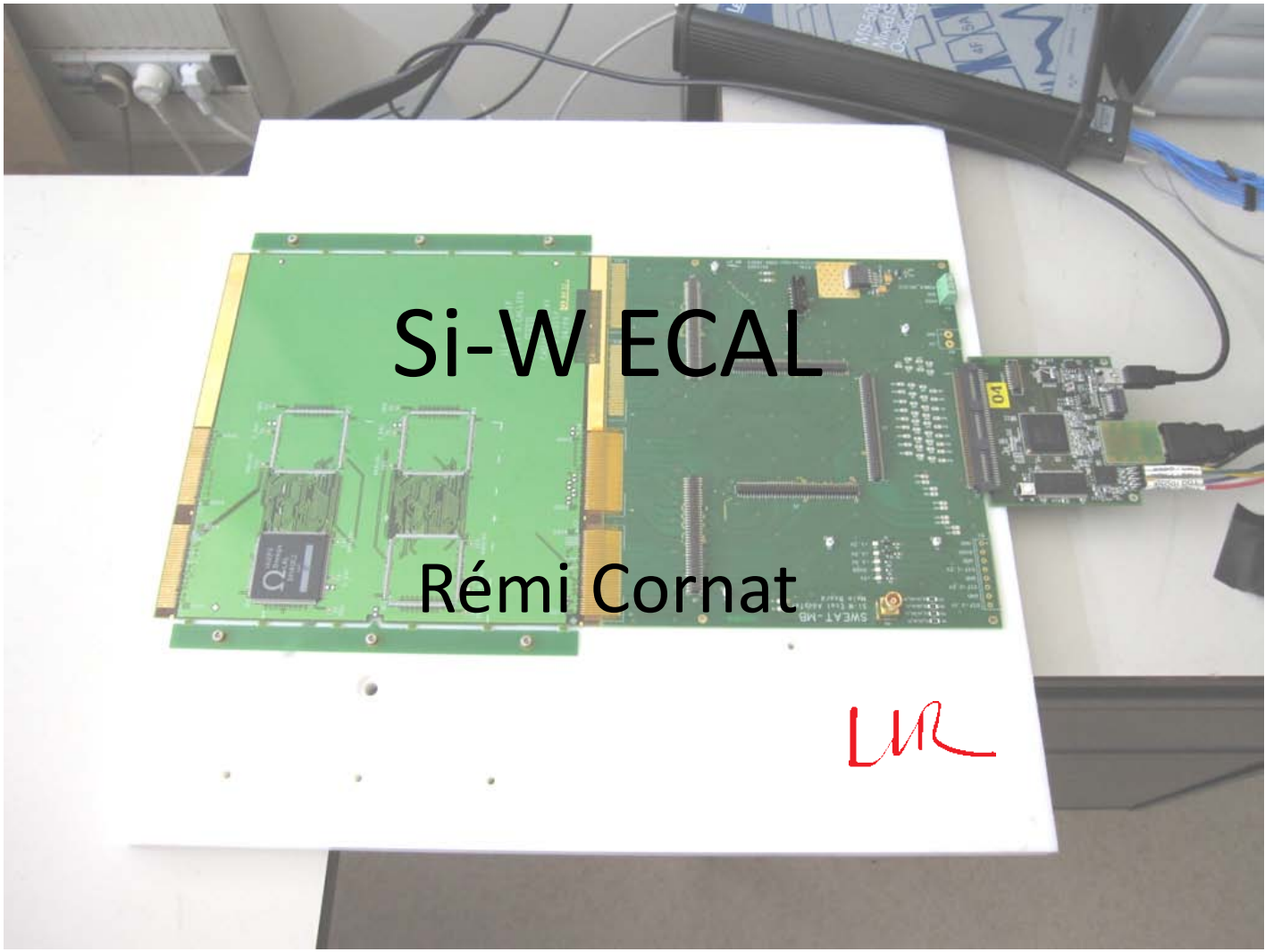

Si-W ECAL
Rémi Cornat

DIF firmware

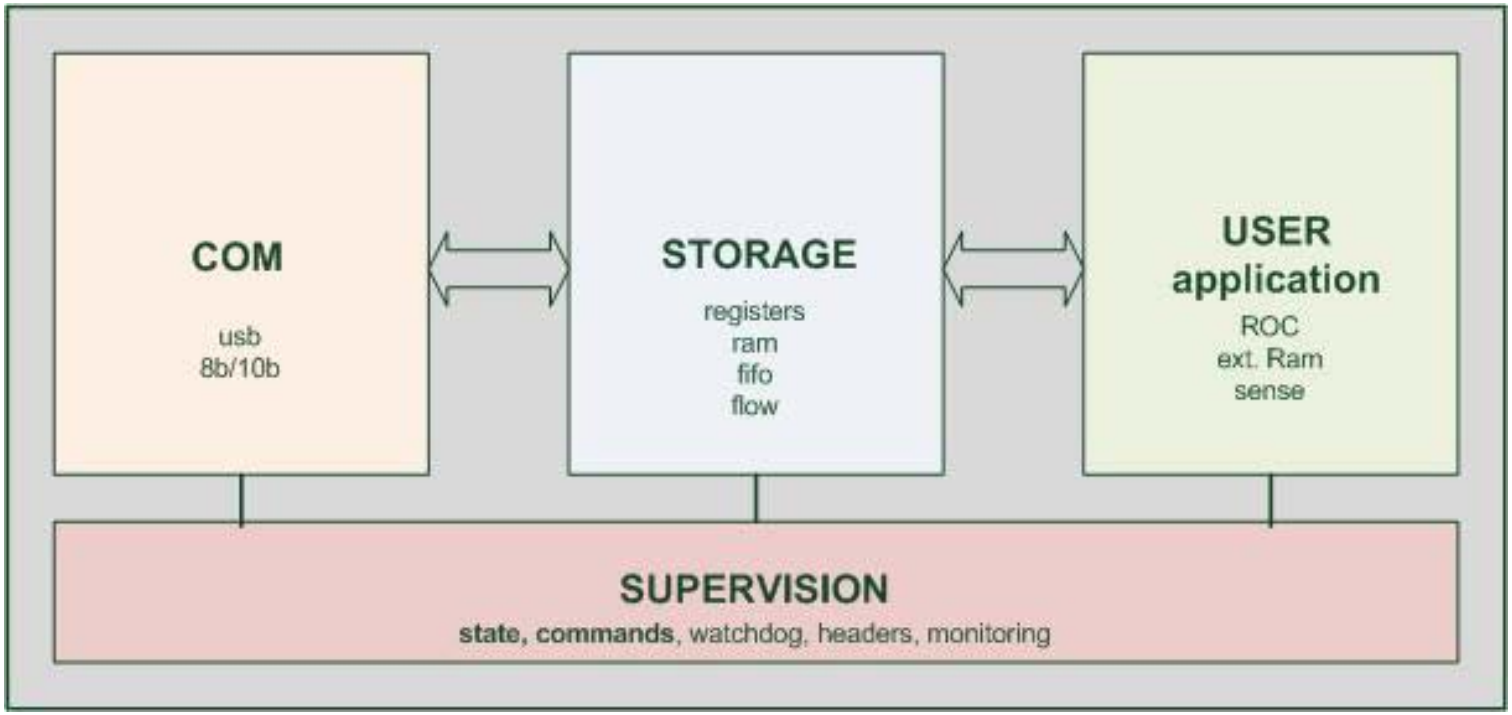
- Code exist in SVN repository
  - Includes firmware from UK
  - Structure to be reworked (not clear)
  - Configuration files to be provided
  - HUGE effort of maintenance (therefore not done)
  - (files do not include header with author citation etc…)
    - But should not be available to the entire world without formal agreement and authorization form authors (to my mind)
- Status : template code
  - Effort of normalization, standardization needed (almost no existing specifications are respected)
  - But principles are there and main mechanism are functioning
  - Code is improved progressively
- Generic : Altera (wrapper for macro), xilinx
  - Shared with Guillaume (LAPP)
- Lot of files
  - Some of them are obsolete
  - 15 MB total (30 pocket books 300 pages, 30 lines, 50 signs),
  - 2 MB (up to date code) needed for DIF (3 pocket books)

Tools

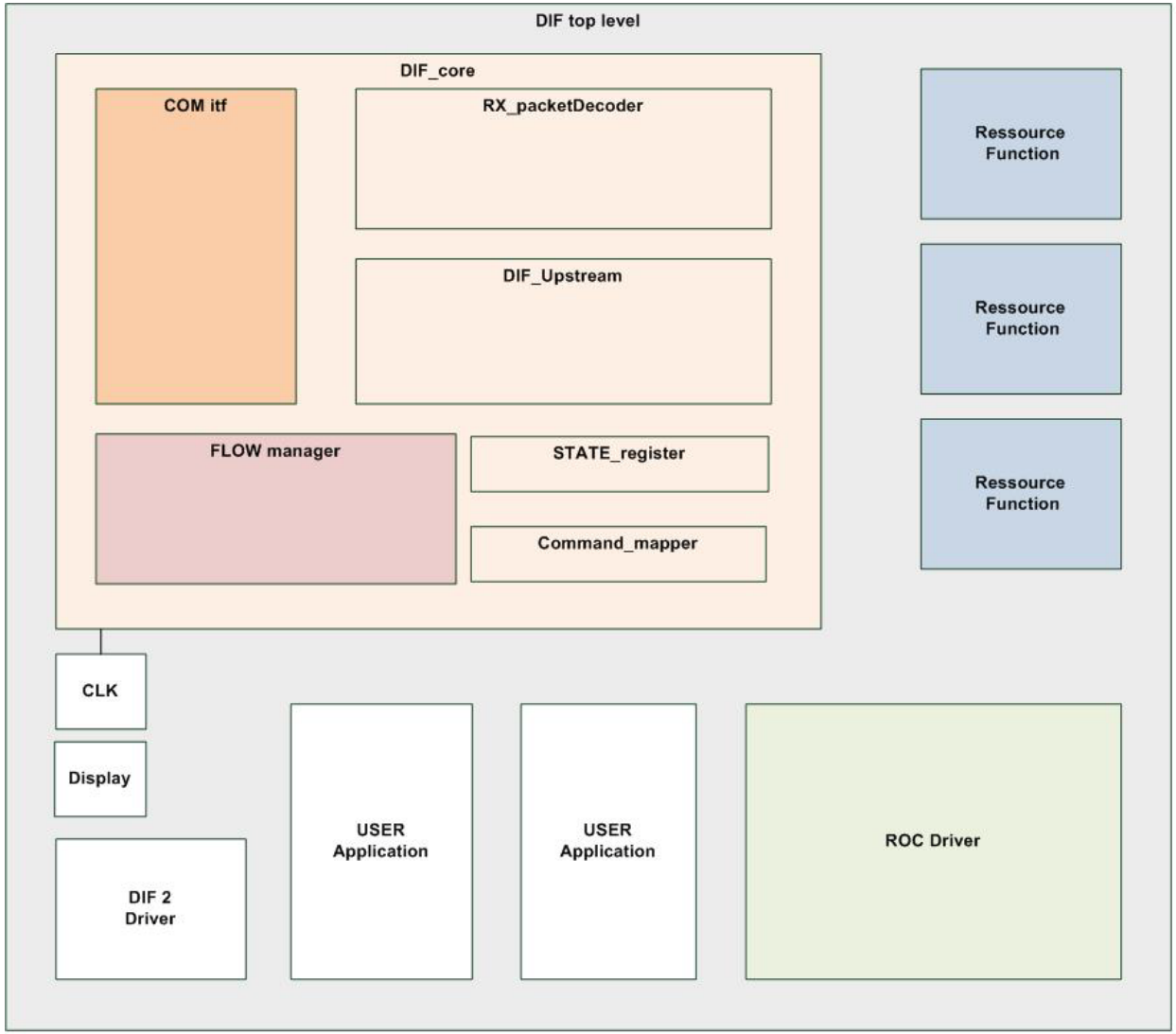
- CADENCE IUS 82 (can use modelsim also)
  - Eventually AMS license for HDL-analog model of ROC (preliminary)
- Synopsis Synplify e201009_SP1
- ISE 11.5 (xilinx)
  - WARNING ! 8b/10b cells are no more supported in version 12
  - And the GEthernet itf (“GEMAC” IP) as well (for LDA) !!!
    - Should buy new IP from xilinx (“TEMAC” IP) : 6 k€ (we accept intl transactions)
    - Some recipes given by xilinx to avoid this (but…)
    - (or keep the old one without using the new ISE versions)

Principle

- Do not mix user specific functions and DAQ/SC blocs
  - Adaptation of various data rates and protocols
  - Top level storage structures as interface (unless direct coupling “flow” if data rate are compatible)



Reality (simplified)



Functioning

- Fast commands (FCMD) and Bloc transfer commands (BTCMD)
  - Enable and trigger (corresponding bit set in one register) functions
  - Function execute and acknowledge the command (reset bit)

- Data transfer
  - To enabled functions (ram, …)
  - Exception : some function decodes directly the data transfer
    - Formally not a function but an application

- BTCMD & DATA : frame with header + data section (16b words)
  - Packet type (1=BTCMD, 2=data, others=tests or specific), DCC nibble 4 MSB
  - Packet ID : not used (supervisor function not implemented)
  - Type modifier : equivalent to an address (function id, register id, ram id)
  - Size (in 16b words)
  - Data : up to 500-508 words (most of the time : only 1=BTCMD or a few)

Configuration

- Code is generic as much as possible (not true everywhere)
- Most of the functionalities can be set-up in configuration files (vhdl packages) : eg. Number of registers, FCMD&BTCMD IDs, etc…
  - This is being generalized, huge effort to be done
  - Should ease implementation of new functions, uniformity among detectors
- DIF_pack package (example)

```
constant  DIF_BTCMD_CFG : DIF_BTCMD_CFG_array_t(DIF_BTCMDnb downto 1) :=
--    TM                REGCONF          RST     ACK    ECHO
((DIFBT_TM_GENFCMD    , "----------------P", X"0000", true ,  false), --16
(DIFBT_TM_READALL     , "----------------P", X"0000", true ,  false),
(DIFBT_TM_READSC      , "---------------CC", X"0000", false,  false),
(DIFBT_TM_POWER       , "----------------P", X"0000", true ,  false), --13
(DIFBT_TM_RESET       , "-------------CCCC", X"0000", true ,  false),
(DIFBT_TM_MODE        , "---------SSSCCCC", X"0000", true ,  false),
(DIFBT_TM_PWRPLS      , "----------PPPPPP", X"0000", true ,  false),
(DIFBT_TM_SC          , "---------------SC", X"0000", false,  false),
(DIFBT_TM_SCLOAD      , "SSSCSSSPC---CCCC", X"0000", false,  false),
[…]
 (DIFBT_TM_SPILL      , "----------------C", X"0000", false,  false)  --1 );
```

Documentation

- DIF firmware .doc (V16) maintained by Mathias
- DIF architecture (V1) maintained by Rémi
  - Attached in the agenda

- Twiki should be updated

Simulation environment

- (If I had time) full DAQ architecture
  - Beam Clock, Spill, trigger generator
  - Simplified models for LDA, (DCC), ROC chain
    - LDA : read scenari files
    - ROC : emulation (full SC, full digital), some AMS blocs
  - Quite uncomplete but architecture is there
  - Will look at coupling to MontéCarlo simulation

Scenario file (example)

```
31  --
32  -- load MODE bank
33  B 1 ./vectors/W_MODE_____1.vec
34  W 10.0 us
35  -- read MODE bank (auto ack)
36  C 1 11 0000000000000001
37  W 50.0 us
38  --
39  --
40  -- load RAM1
41  -- -- enable charegement RAM (BTC SC)
42  C 1 9 0000000000000001
43  W 15.0 us
44  B 1 ./vectors/DAT_CFG_RAM1.vec
45  W 100.0 us
46  -- -- read RAM
47  C 1 9 0000000000000001
48  W 15.0 us
49  -- read RAM1
50  C 1 7 0001000000000000
51  W 300.0 us
52  -- stop read RAM1
53  C 1 7 0000000000000000
54  W 20.0 us
55  -- stop RAM1
56  C 1 9 0000000000000000
57  W 20.0 us
58  --
59  --
60  -- SC bloc read ram
61  --C 1 14 0000000000000001
62  --W 800.0 us
63  -- -- generate ROC slow control
64  C 1 8 0000000000000001
65  W 500.0 us
66  R
```

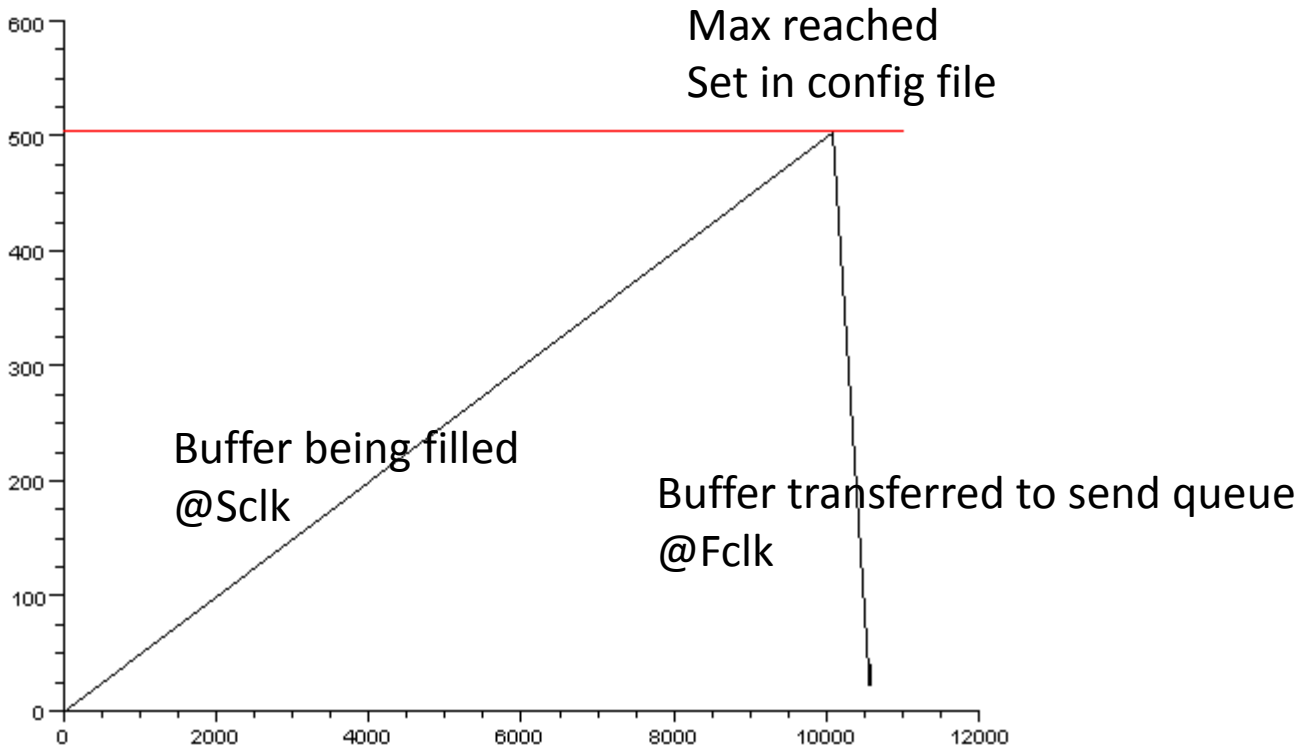
- Establishing procedures (sequence of orders) => should be common to SW developement

- Test and validation of DIF firmaware using the procedure

- Most of bugs found and solved without HW tests

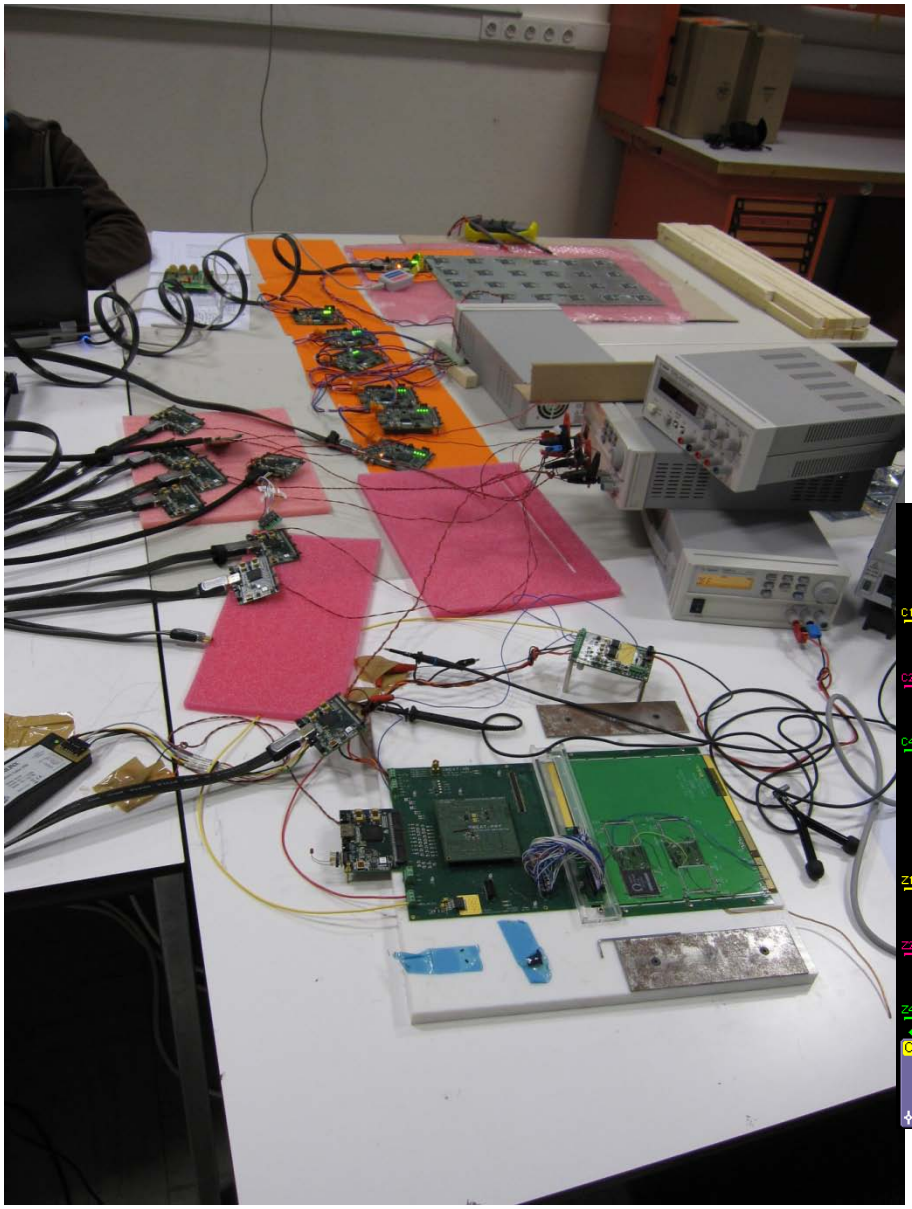
(very preliminary) system level sim.

- Example : buffer occupancy (readout of one SC frame)
  - To be generalized (MC stats)



- Funding available for a PhD
  - System level simulation
  - DAQ for 10^8 channels
  - Integration & tests of tech. prototype
  - Need candidates !

Conclusion : HW tests



SAME code used for both DHCAL & ECAL

Low level operations « by hands »

User friendly (similar to labview)
SW is mandatory
To be discussed as soon as possible