# Common SiECAL-SDHCAL beamtest efforts

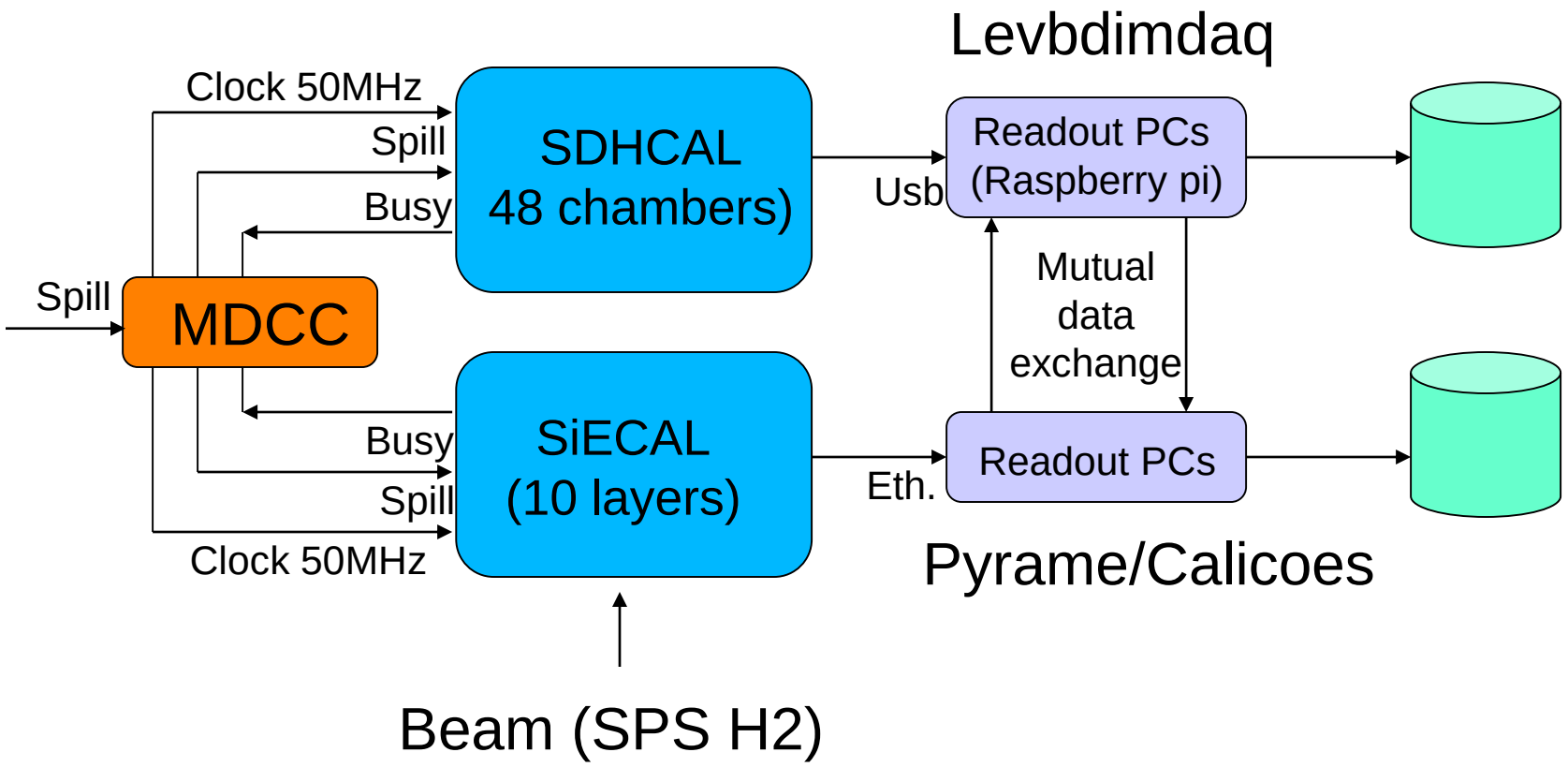
C. Combaret, R. Cornat, F. Magniette, L. Mirabito

# Global system constraints

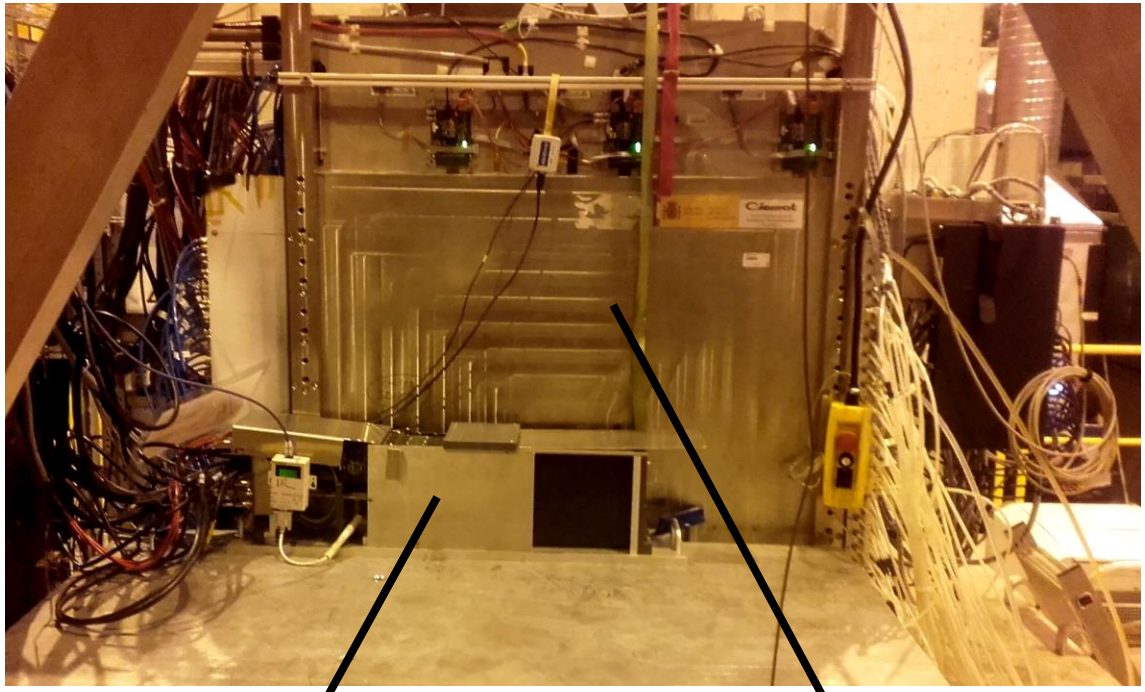
**We had to try to keep both systems as much as possible as they have been designed and debugged**

- No modification in hardware architecture of both SDHCAL and SiECAL
- No deep modification of firmwares of both SDHCAL and SiECAL (especially in acquisition cycles)
- No deep modification of daq softwares of both SDHCAL and SiECAL

# Global system architecture (1)



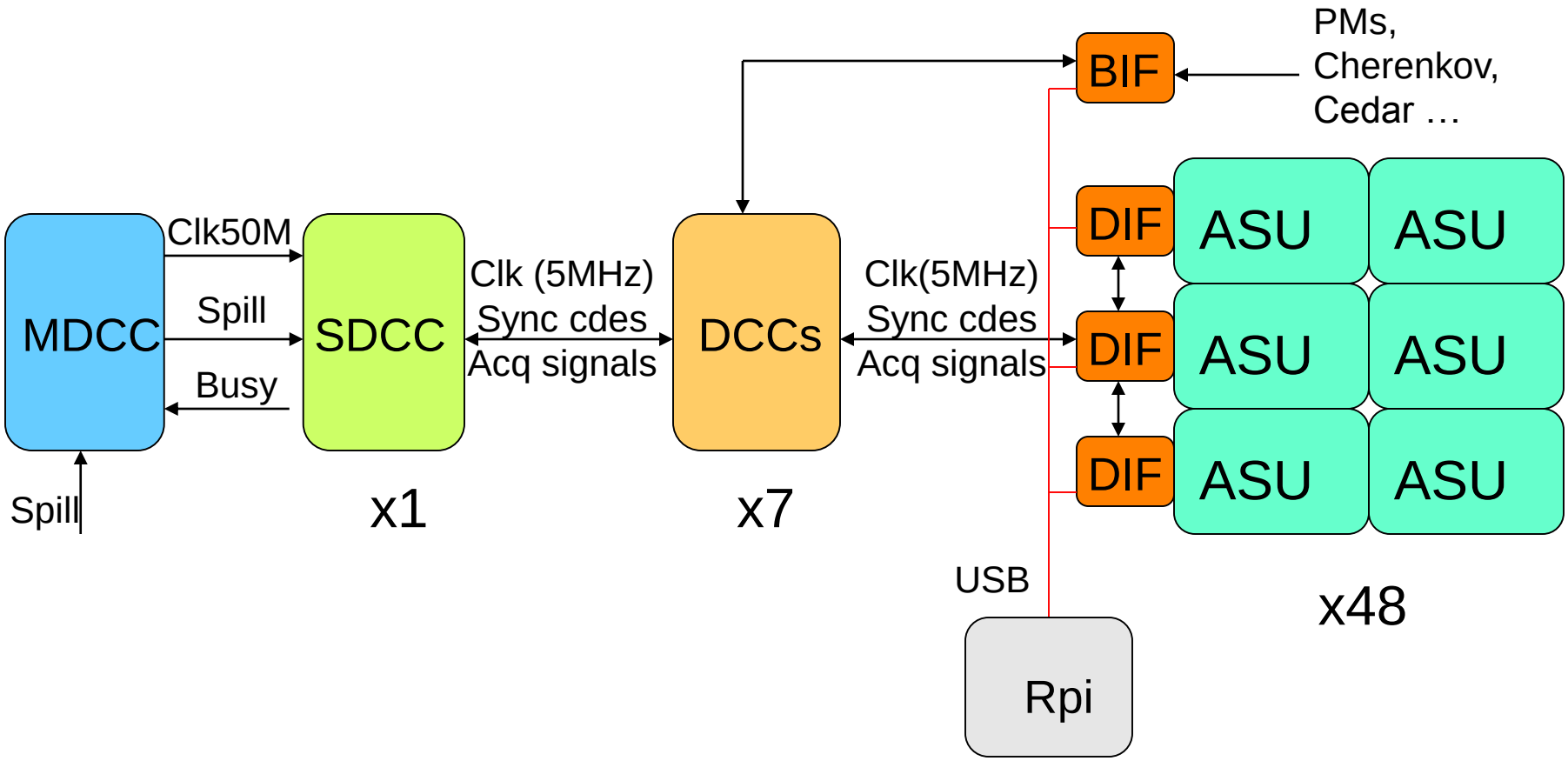
# Global system architecture (2)



SiECAL

SDHCAL

# SDHCAL


PMs,
Cherenkov,
Cedar …
BIF
MDCC
Clk50M
Spill
Busy
SDCC
Clk (5MHz)
Sync cdes
Acq signals
DCCs
Clk(5MHz)
Sync cdes
Acq signals
DIF
DIF
DIF
ASU
ASU
ASU
ASU
ASU
ASU
Spill
x1
x7
USB
Rpi
x48

# SDHCAL/SiECAL synchronization constraints

**SDHCAL**

DIFs need a 5MHz clock
Synchronous commands are sent with the 5MHz clock
The spill can be any lenght, several acquisitions per spill
No data readout at end of spill (last data in memory are lost)

**SiECAL**

DIFs need a 50MHz clock
Synchronous data are sent with the 50MHz clock
1 spill = 1 acquisition
Data are always read at end of spill

# SDHCAL/SiECAL synchronization solutions

**SDHCAL**

DIFs need a 5MHz clock

→ nothing to do, just live with it.

Synchronous commands are sent with the 5MHz clock

→ AC link to DCCs removed, Idle commands removed, so SDCC commands get resynchronized on 5MHz clock → jitter with SiECAL is 1 slow clock (5MHz) ie 1 BCID

The spill can be any length, several acquisitions per spill

→ MDCC firmware to handle both detectors spills definition (see next slide)

No data readout at end of spill (last data in memory are not transferred to DAQ PCs)

→ MDCC firmware to handle both detectors spills definition (see next slide)

**SiECAL**

DIFs need a 50MHz clock

→ nothing to do, just live with it.

Synchronous data are sent with the 50MHz clock

→ nothing to do

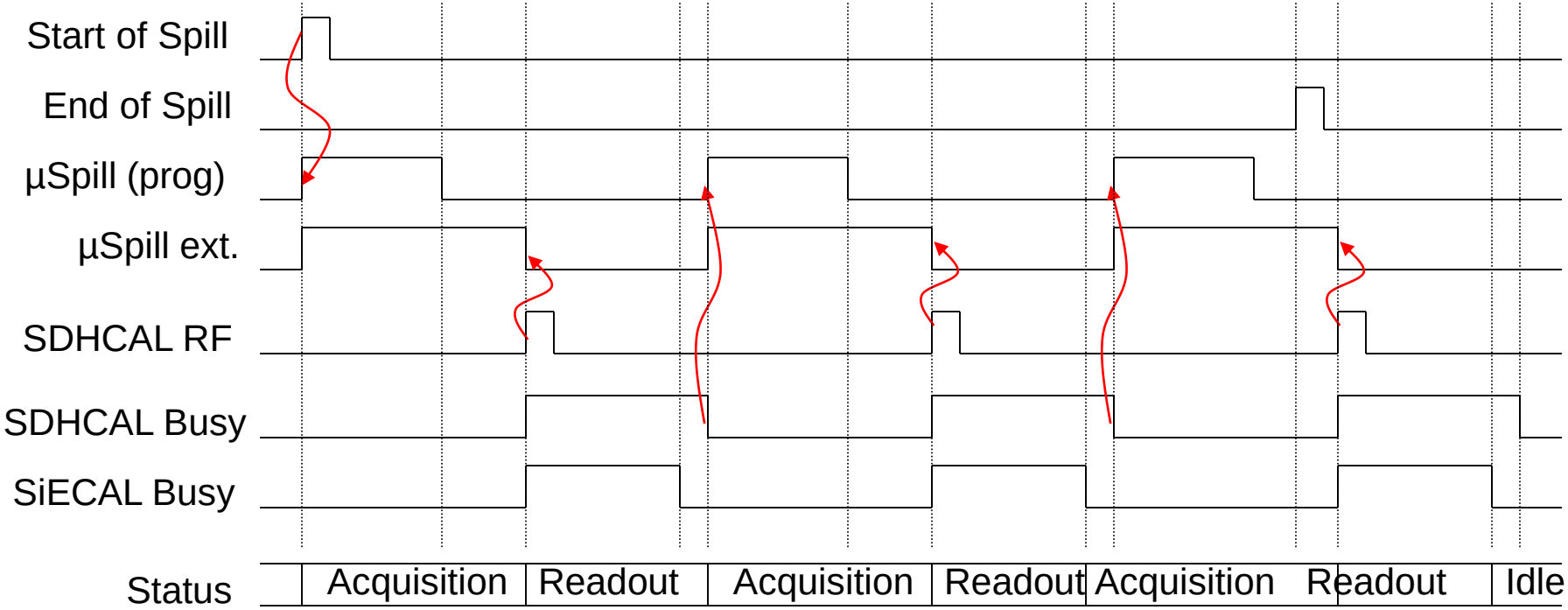
1 spill = 1 acquisition

→ MDCC firmware to handle both detectors spills definition (see next slide)

Data are always read at end of spill

→ nothing can be done, adapt SDHCAL to handle this

# MDCC




- µSpills generated to accommodate the SiECAL firmware (duration programmed in MDCC registers)
- µSpills artificially extended until the next SDHCAL Ramfull to get SDHCAL data of every µSpill
- Assume Readout of the SiECAL is shorter than the Readout of the SDHCAL (always the case)

# Synchronization

**Alignment of acquisition windows**

SPILL triggers the power-up then “start-acquisition” then “val_event”

Alignment of the “val_event” windows is mandatory, ECAL is about 1.4 ms late wrt. SDHCAL

SDHCAL “val_event” can be delayed (programmable in SDHCAL SDCC)

Alignment within one slow clock period done (< 1 BCID on each side)

**Synchronization of SPILL ID**

Runs start with the ECAL and SDHCAL counters set to 0

**Use of BUSY to optimize the dead-time**

SPILL restarted immediately after end of busy: OK on SDHCAL, OK on ECAL

**Central clock**

50MHz common clock, feeding the ECAL CCC and the SDHCAL

**Software**

Software Veto from SDHCAL and ECAL sent to the MDC

# Conclusion

## Results are satisfactory

Successfully configured and acquired data for both detectors from central SW (PYRAME or XDAQ)

Seen issues fixed (ECAL BUSY ; SDHCAL jitter ; logic levels ; supervisor logic)

Run starts with time_stamp counters properly reset

Acquisition windows (and BCIDs) aligned within 0 … +1 : checked using several GDCCs (ECAL)

SW controlled veto added on Main HW supervisor for SW/HW sync.

## On going tasks

Use of common clocks (50MHz system & BXID clock): existing features, add few cables

Fix data sharing after recent SW updates at CERN

## Remarks have been sent to EUDAQ community

Together with a demand for support and an invitation to come & help during our tests