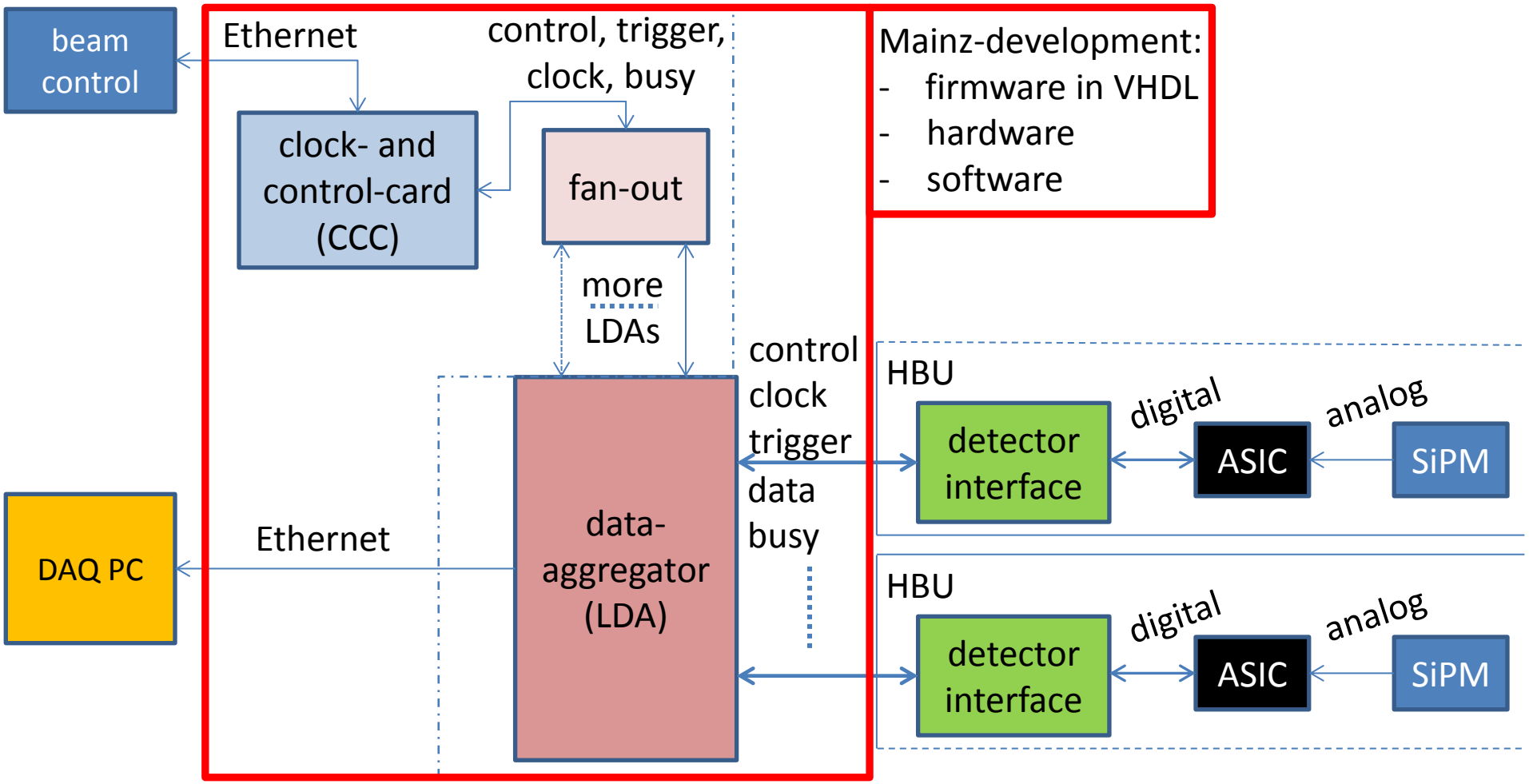
CCC and TLU

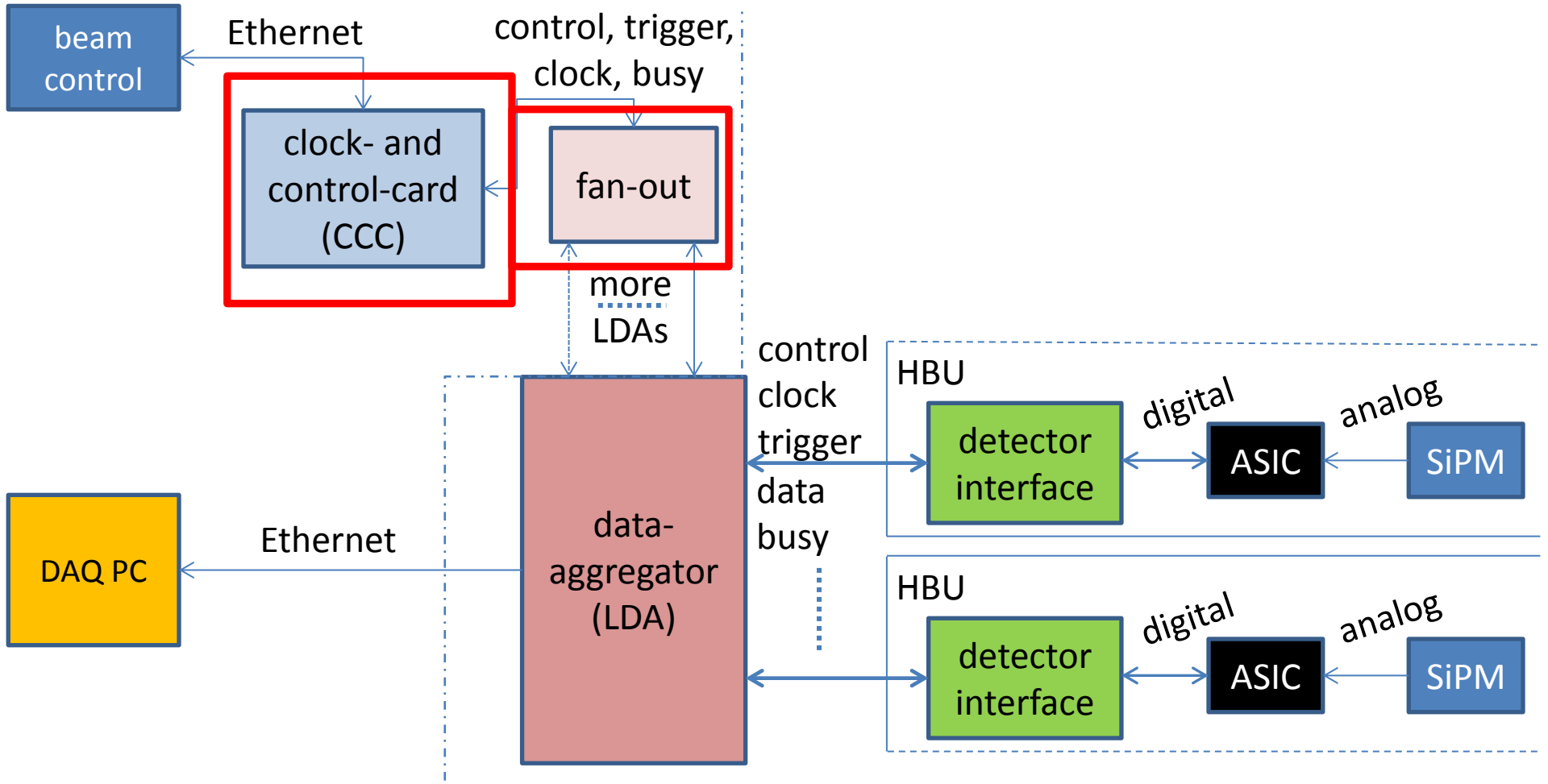
André Welker
Lennart Adam, Bruno Bauss, Volker Büscher, Reinhold Degele, Karl Heinz Geib, Sascha Krause, Yong Liu, Lucia Masetti, Phi Chau, Uli Schäfer, Rouven Spreckels, Stefan Tapprogge, Rainer Wanke

AIDA Meeting
Frascati, 11. April 2013

Read-out-chain



Read-out-chain



Clock and Control Card(CCC)

Requirements:

1. **High clock stability in the whole detector**

2. **Configuration interface over Ethernet**
   - manage different running modes
   - for feedback functions
   - for beam control

Status:

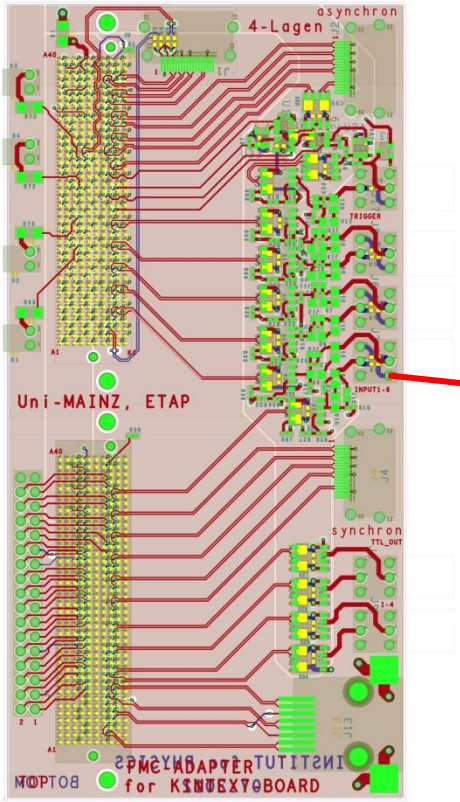
1. **Built and tested for a lab setup and used in test beams 2012**

2. **Next step: Upgrade for bigger test beams (almost finished)**

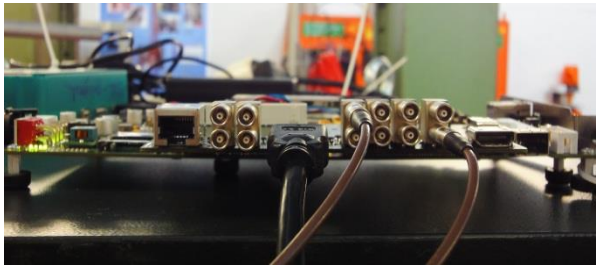
2012 Clock and Control Card

Mezzanine on an Kintex 7 Evaluation-Board:

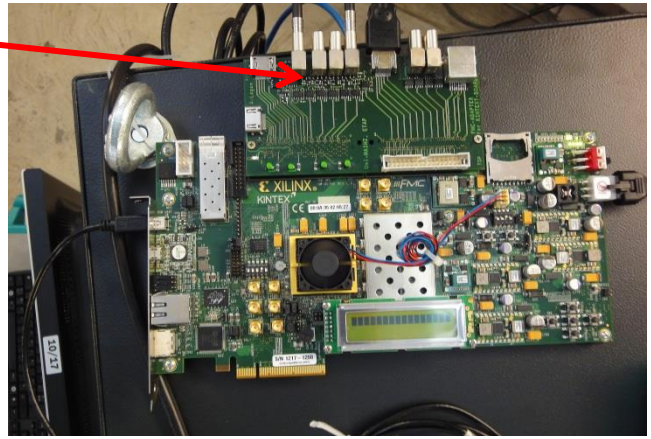
FPGA-controlled CCC board:



CCC at the testbeam (front):

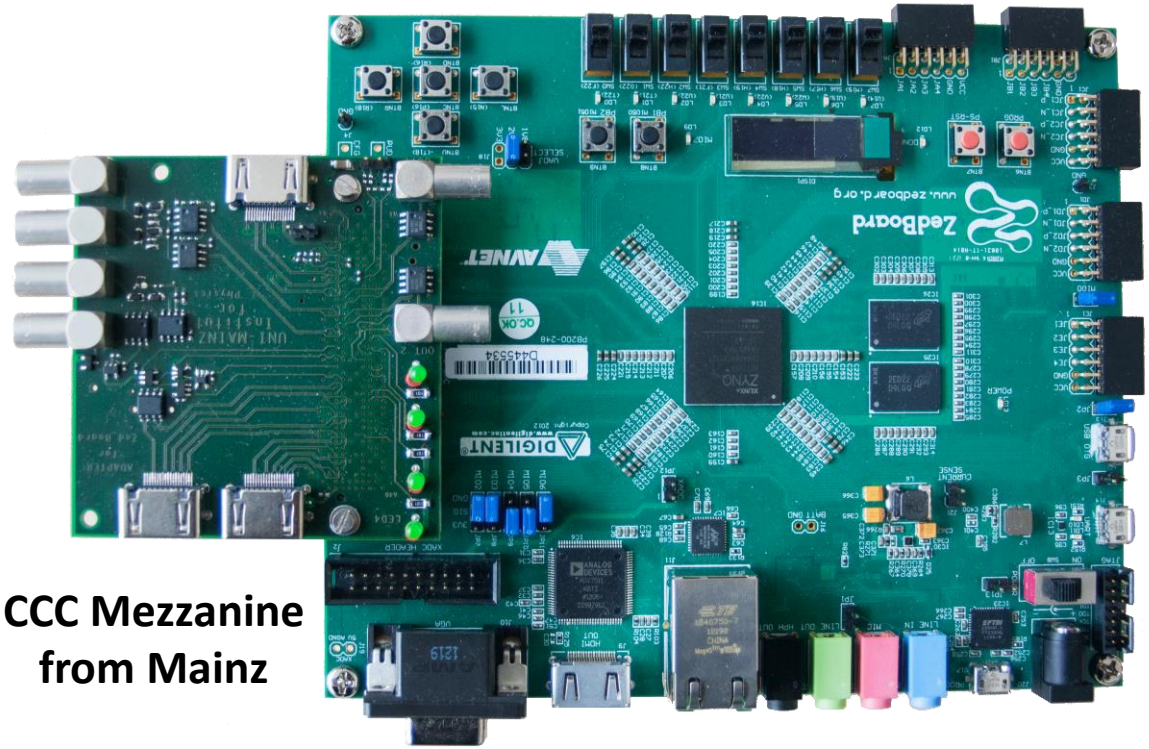


CCC at the testbeam (top):



New 2013 Clock and Control Card

First iteration with a Zynq-processor on a Zedboard for development:



CCC Mezzanine from Mainz

Zedboard:

- Evaluation Board from Digilent

Zynq:

- ARM9 dual core processor (Linux)

+

- FPGA

Clock and Control Card(CCC)

Final design for test beams:
6U-VME formfactor

front:

Ethernet

Micro-SD

USER

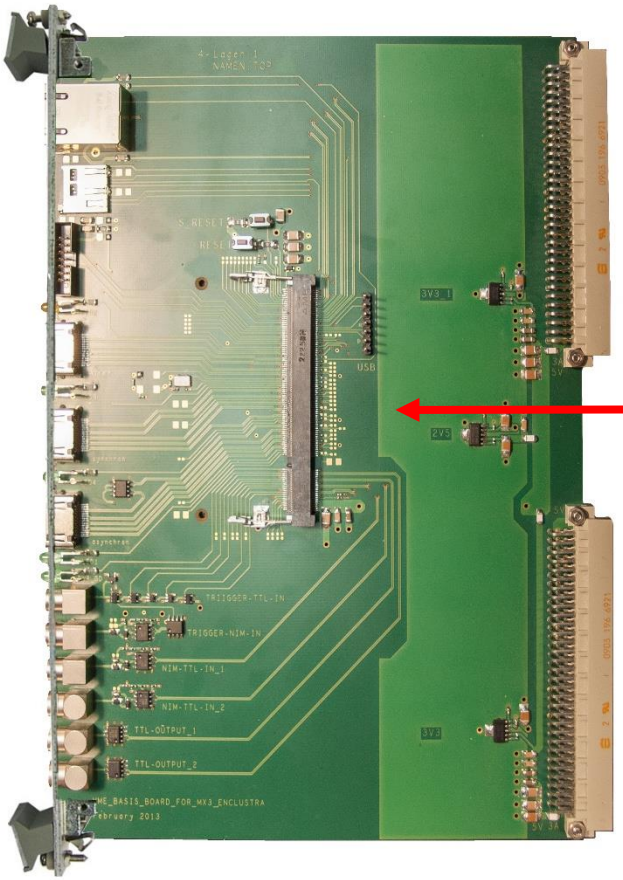
SYNC

ASYNC

TTL IN

TTL OUT

top:



Peculiarity of the Mars-Modul:

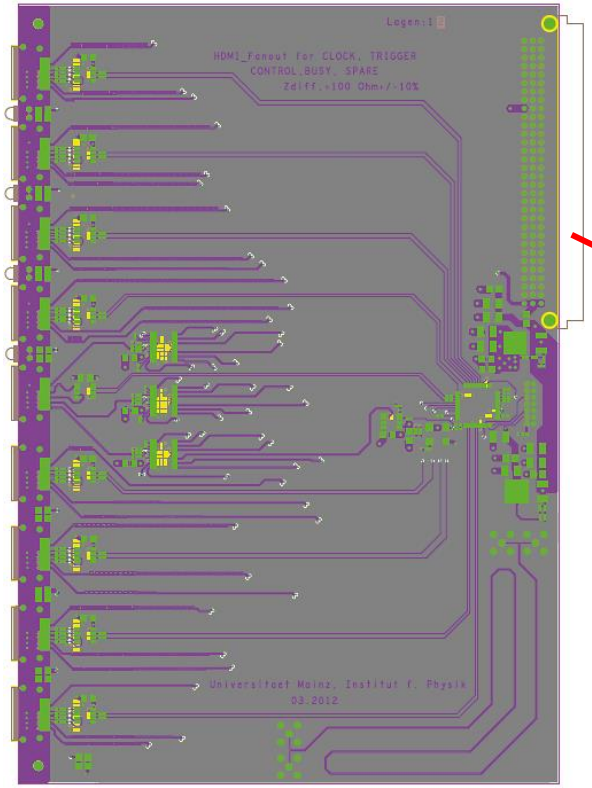
1. Zynq
   - ARM9 dual core
   - Linux + FPGA
2. 512MB NAND/RAM

Mars-ZX3 firm:Enclustra

CCC Fan-out same as in 2012

6U-VME Fan-out board:

Fan-out board layout:



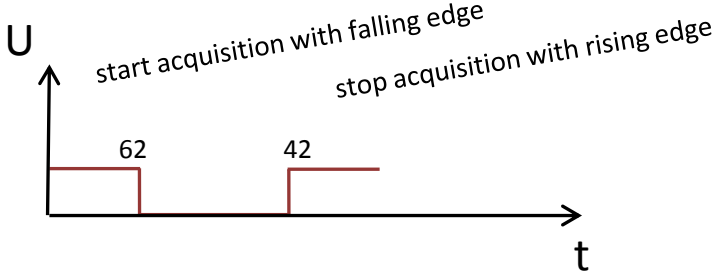
Fan-out board at the testbeam:

Running Modes

Four main modes and 13 commands:

signal:

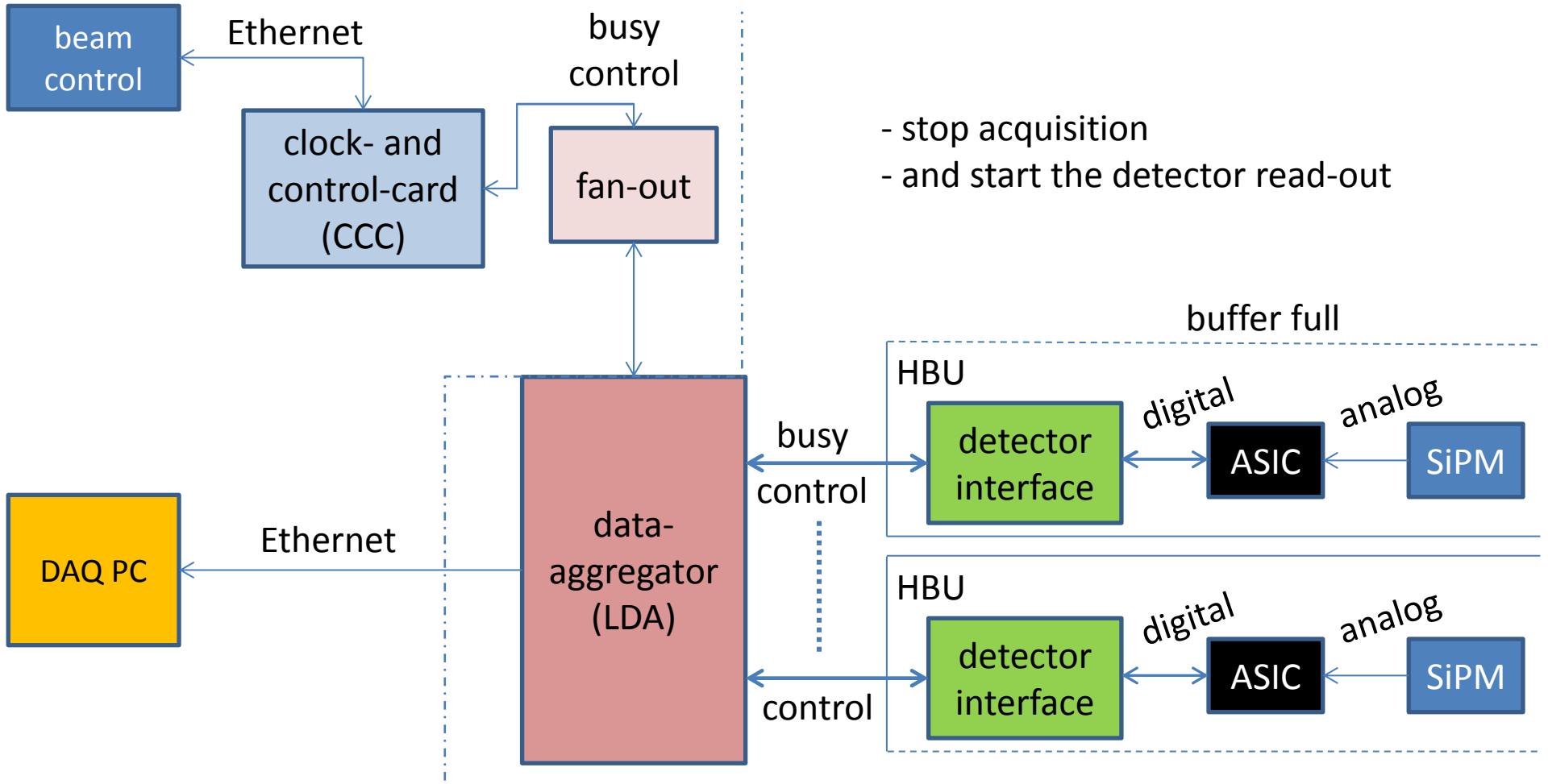
e.g. busy:



conditions:

ASIC buffer full

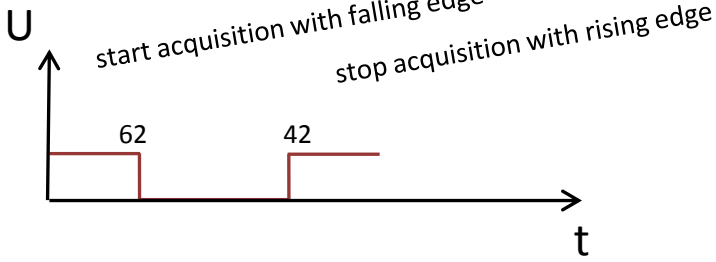
Read-out-chain



Running Modes

Four main modes and 13 commands:

signal:
e.g. busy:



conditions:
ASIC buffer full

extended commands: (backward compatible)

- manual trigger
- listen on falling or rising edge
- status
- hard- and softreset
- and many more

CCC and TLU Configuration

Processing Instruction Configurator (piconf) (by Rouven Spreckels)

- Configures all devices
- Configuration procedures are automated
- Everything stored in one or multiple XML files
- Self-explanatory, predefined XML tags
- Interface independent:
  - implemented: TCP/IP

piconf example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<piconf version="1.0.0">
        <!-- Only <pi> tags in this level are parsed and must only contain
             predefined tags. -->
        <pi>
                <conf>
                        <!-- Include some values. -->
                        <fpath>lda/some_values.xml</fpath>
                        <!-- Define another value. -->
                        <another_value>1</another_value>
                        <!-- piconf informs us if an endpoint rejected unknown
                             or invalid values. -->
                </conf>
                <!-- This <xpath> tag will be replaced by the <tcp> tags below,
                     since it is pointing to them. -->
                <xpath>../../some_devices/node()</xpath>
        </pi>
        <some_devices>
                <!-- Above <conf> tag will be sent to following devices. -->
                <tcp>
                        <host>lda1.physik.uni-mainz.de</host>
                        <service>3141</service>
                </tcp>
                <tcp>
                        <host>lda2.physik.uni-mainz.de</host>
                        <service>3141</service>
                </tcp>
        </some_devices>
</piconf>
```

CCC with a TLU

Two possibilities:

1. **Baseline: Integrate the current TLU VHDL code**
   Fallback: Integrate a hardware part for the Ethernet

2. **Baseline: Integrate a hardware part for the Ethernet**
   Fallback: Integrate an own TLU VHDL code

Both solutions depends on the timescale, less manpower until the May 2013 testbeam

Status of CCC

1. **One module physically exists**
2. **Hardware tested**
3. **Firmware almost done**

CCC should be ready to go for the test beam in May 2013

Thank you for your

attention!