A DAQ Test System for the CMS Phase II Tracker Upgrade

General Concepts and Overview

Mykyta Haranko

4th MT student retreat, Berlin, 11th June 2018

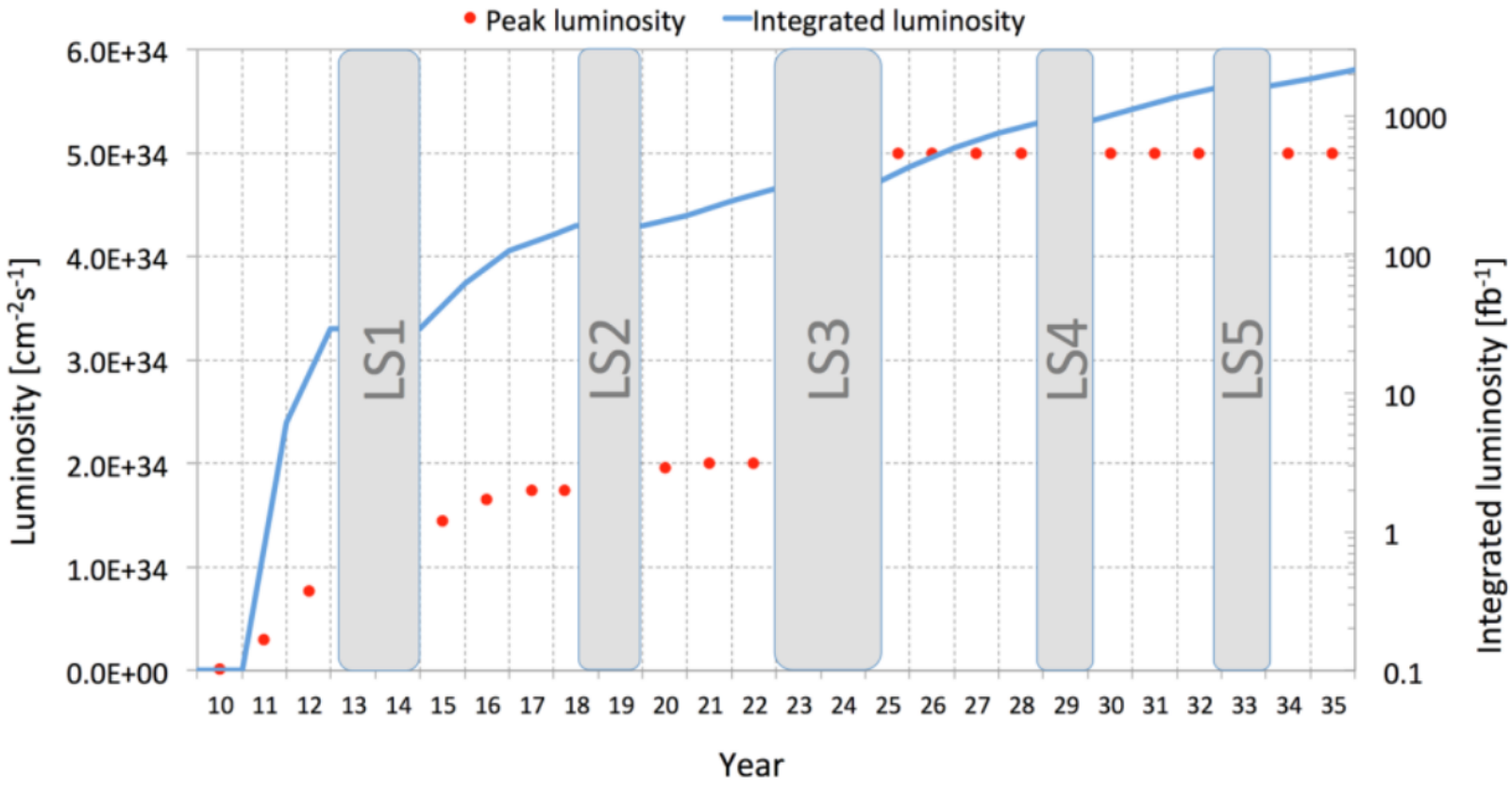
HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

LHC Schedule

Higher integrated luminosity – higher radiation dose = Radiation damage to the detector

!! Detector needs to be upgraded !!

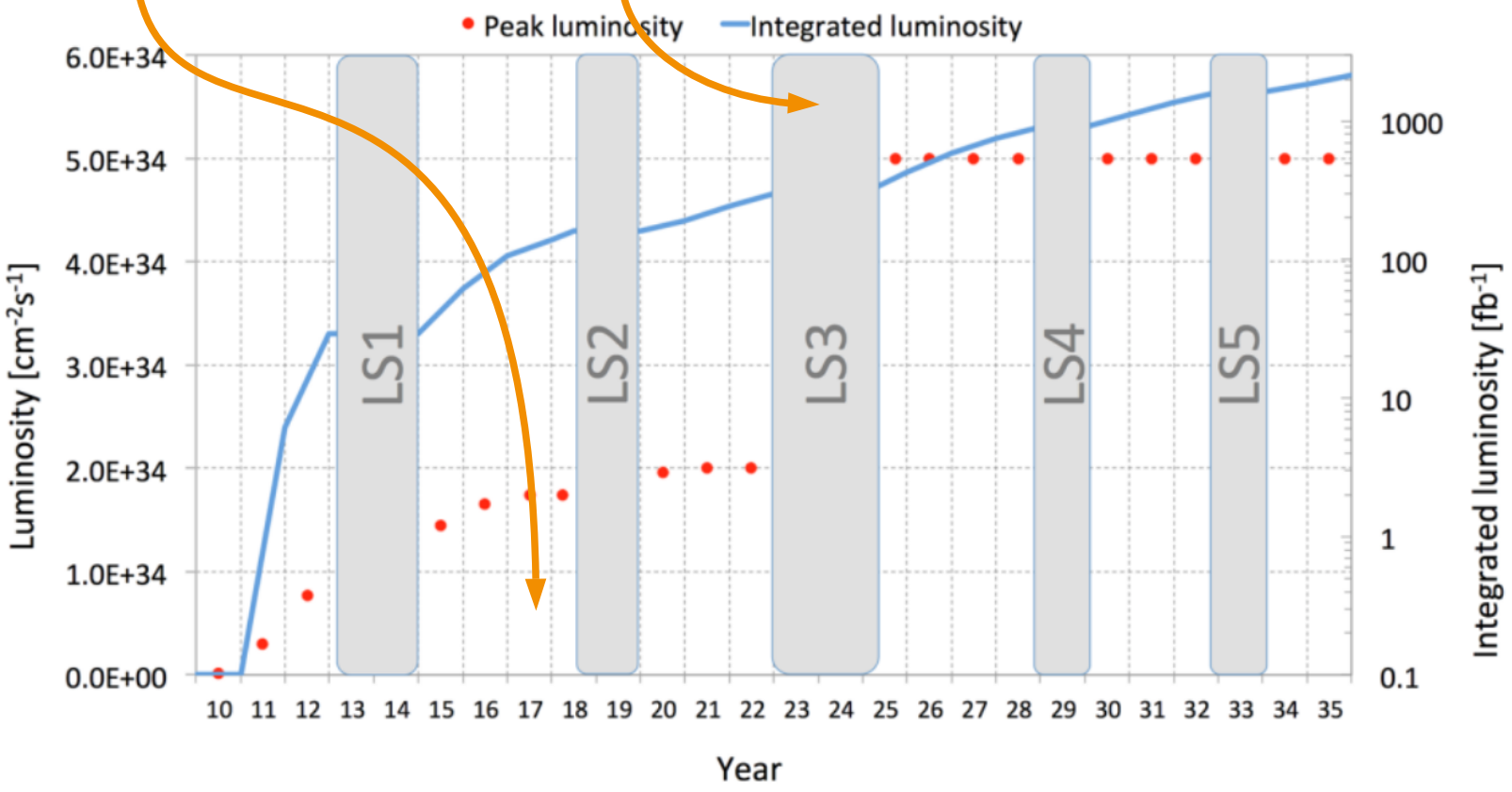
Higher instantaneous luminosity = High pileup, high trigger rate



CMS Tracker Upgrades

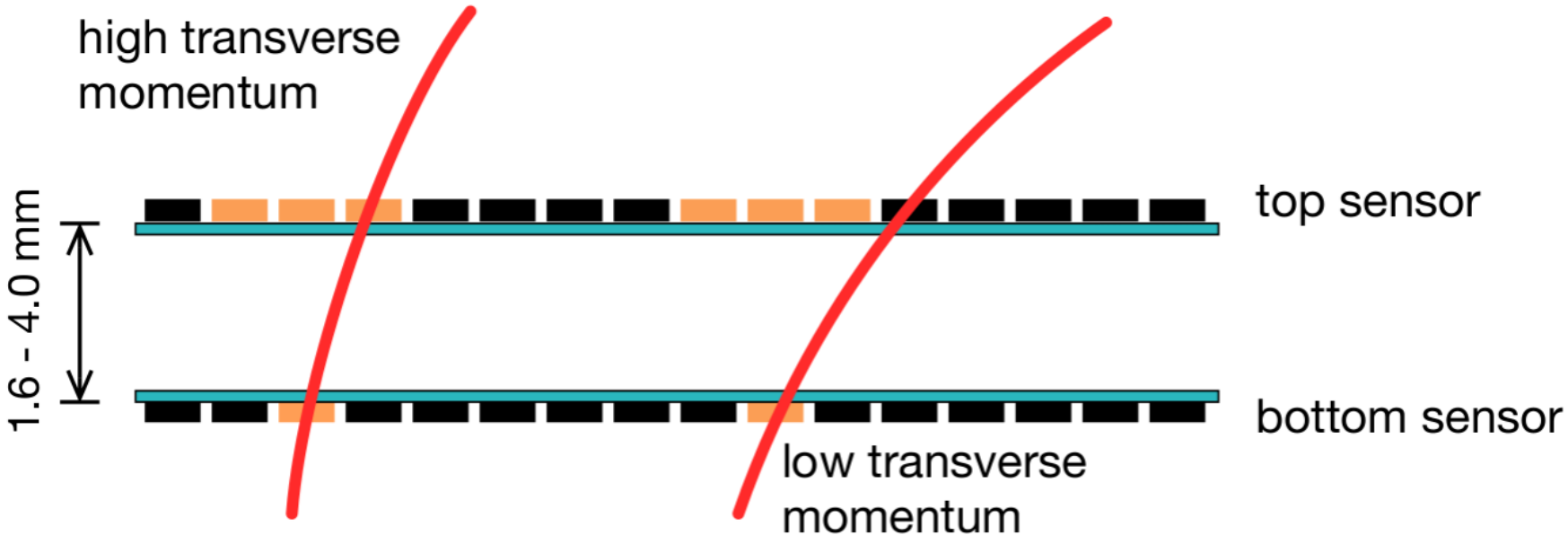
Extended technical stop:
- 4-layer pixel detector installation

New silicon tracker with trigger capability



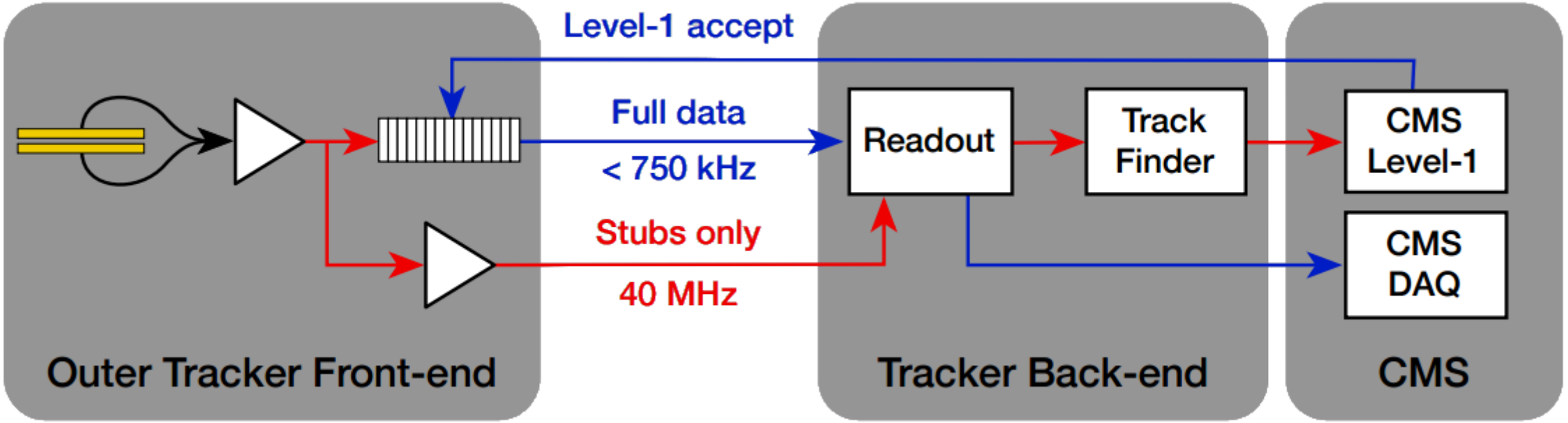
New Outer Tracker Module Concept

- **On-board pT discrimination:**
  - Signals from two closely spaced sensors are correlated.
  - Local rejection of low-pT tracks: so-called “stubs” formed and transmitted to the L1 trigger, when a high-pT event occurs.
  - The strong magnetic field of CMS makes it possible to measure pT locally with reasonable effort.



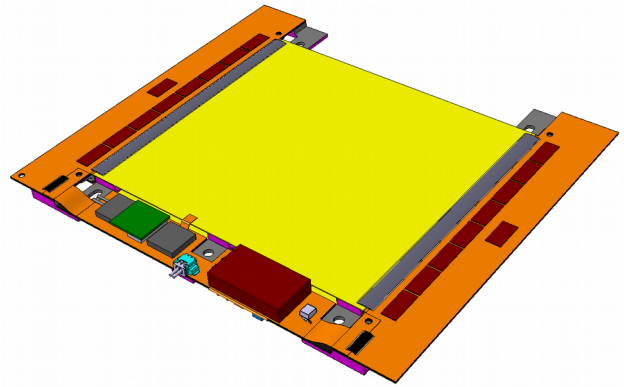
Data Flow

- **The stub data will be used to form Level-1 tracks**
  - "Stubs" will be sent at bunch crossing frequency (40 MHz)
  - Data readout at ~750 kHz

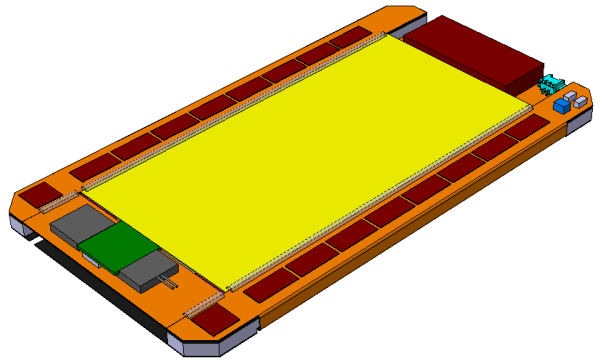


Phase 2 Outer Tracker Upgrade: 2S and PS Modules.

2S (“Two-Strip”) Module

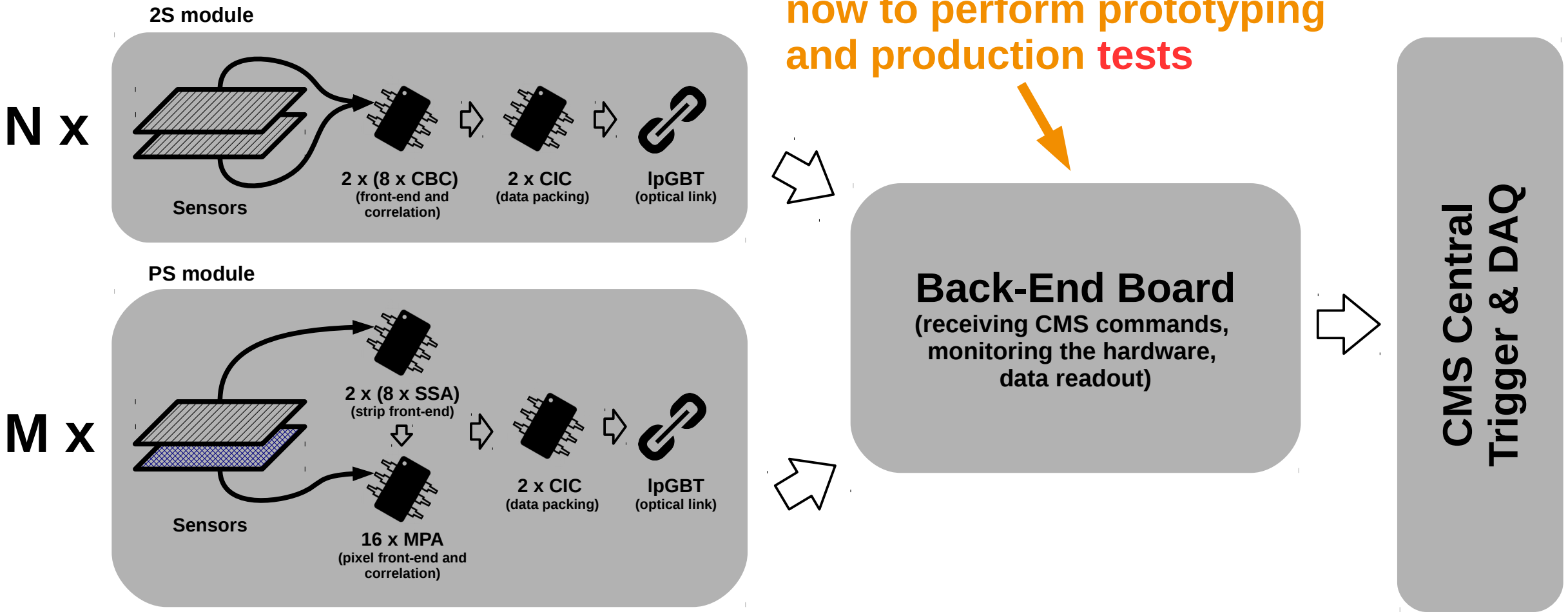


PS ("Pixel-Strip") Module



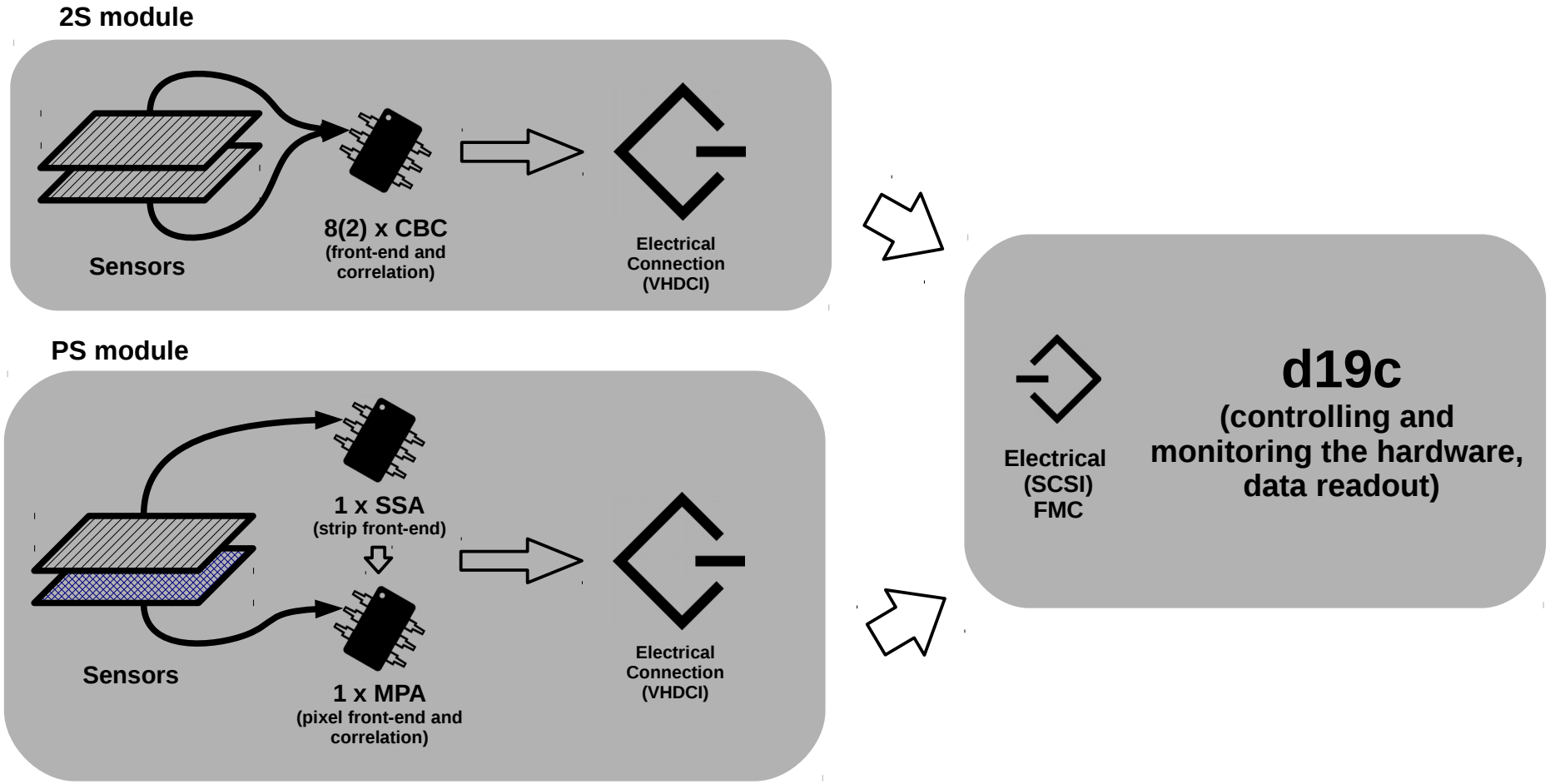
Data Flow - Closer Look

A test DAQ has to be built now to perform prototyping and production tests

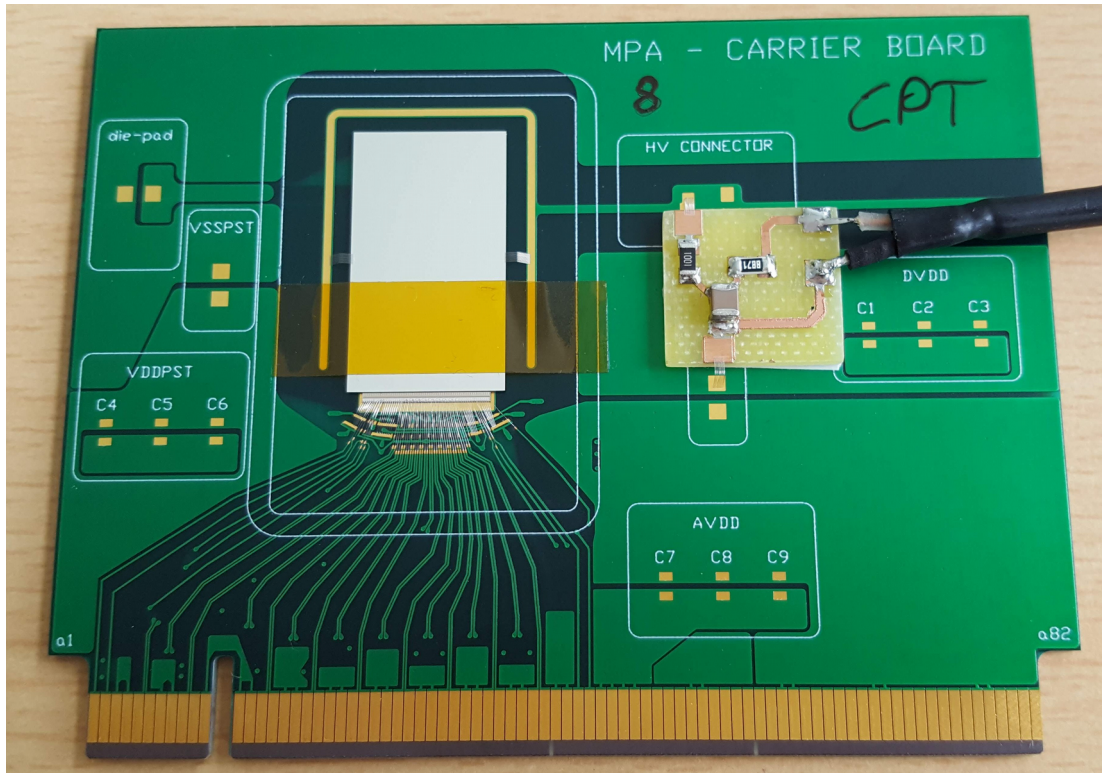


Data Flow – Electrical - Prototyping

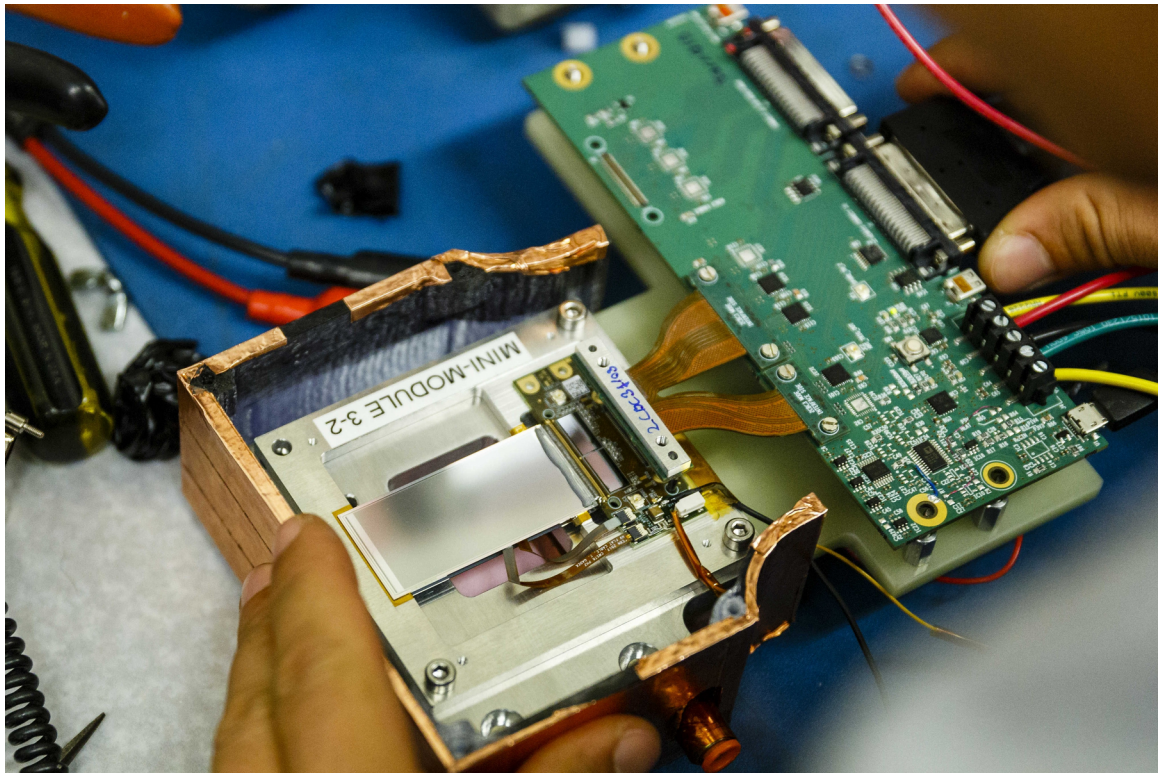
- **Building a final version of the tracker from the first try – might be a bit difficult. Prototyping is needed.**



Prototype Modules



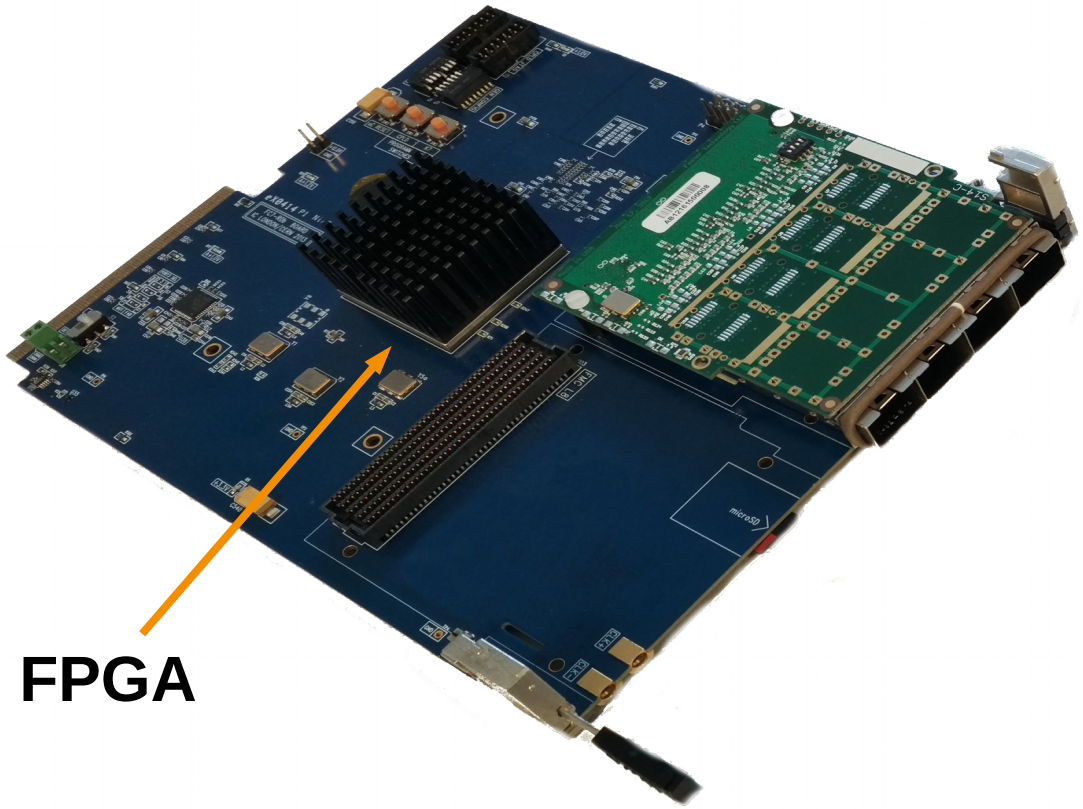
MaPSA (1xMPA) assembly

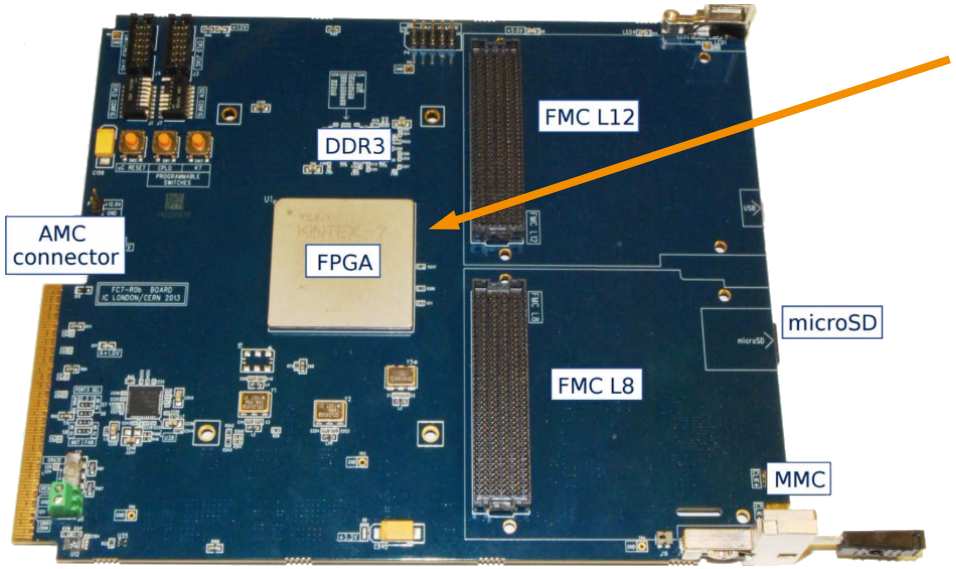


2xCBC3 module (2S prototype) at FNAL © Christian Dziwok

The FC7 Board

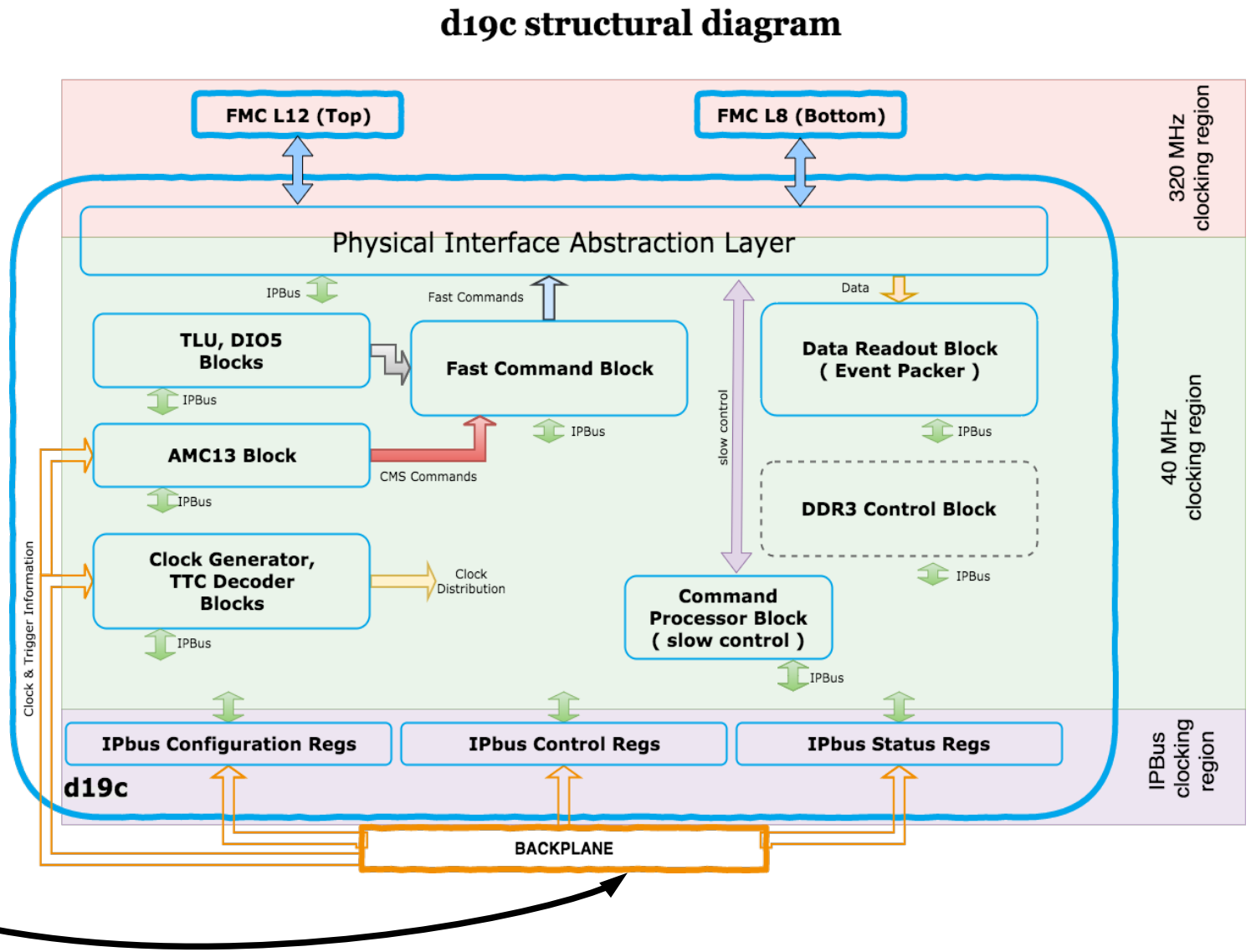
- **FMC Carrier – Xilinx Series 7 (FC7)** is a MicroTCA-compatible Advanced Mezzanine Card for generic data acquisition and control applications
  - AMC Connector
  - DDR3 Memory Interface
  - Programming from a microSD card
  - **Firmware has to be developed**





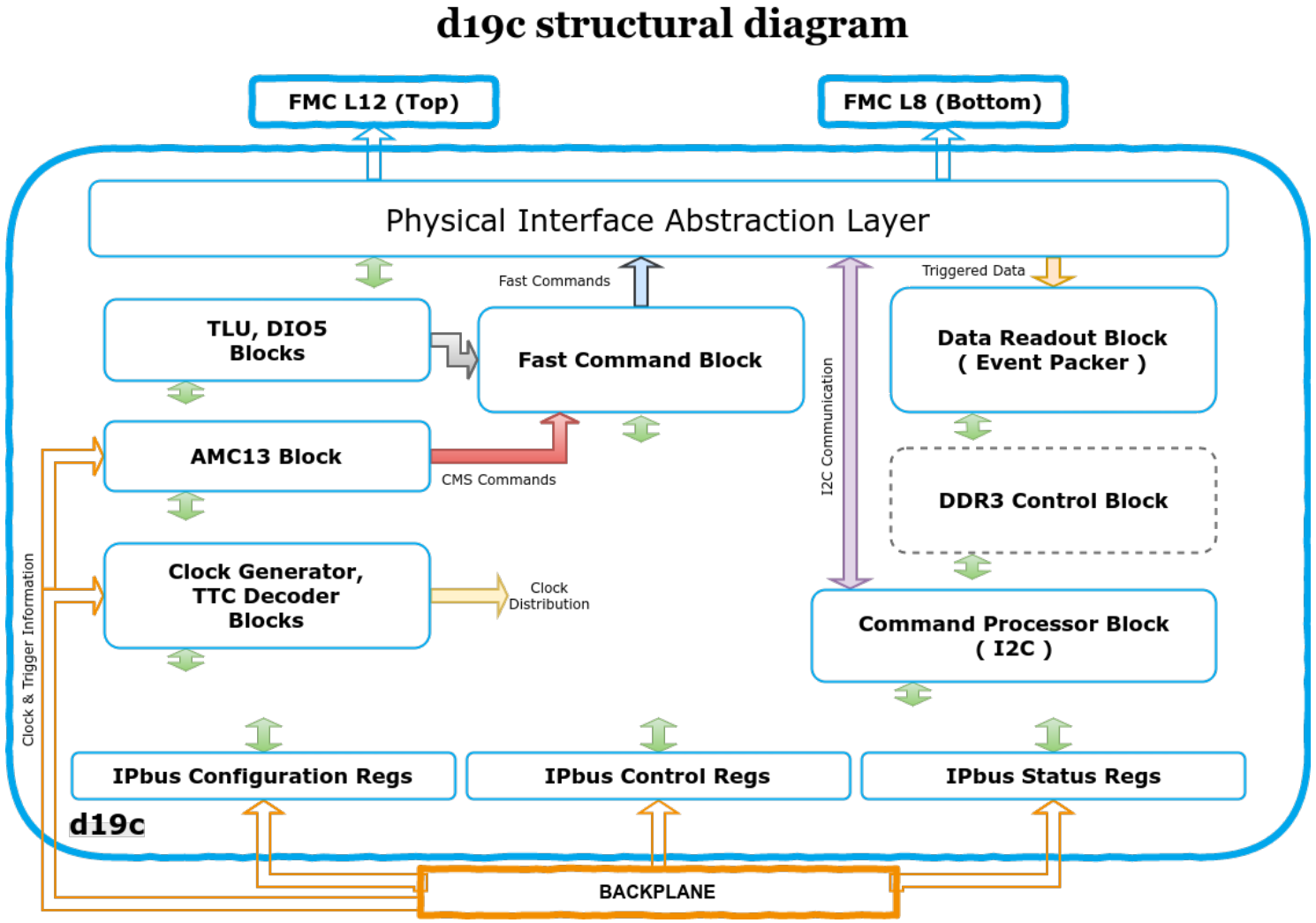
General Structure

- **d19c (µDTC) –** is the project, based on the FC7 AMC card and used for the system test purposes
- Is capable of reading all of the existing Phase II Outer Tracker hardware prototypes
- Has a block structure:
  - System Core (not shown) – responsible for all the basic AMC card functions and interaction with uTCA
  - User Core – all the logics responsible for the hardware readout and control



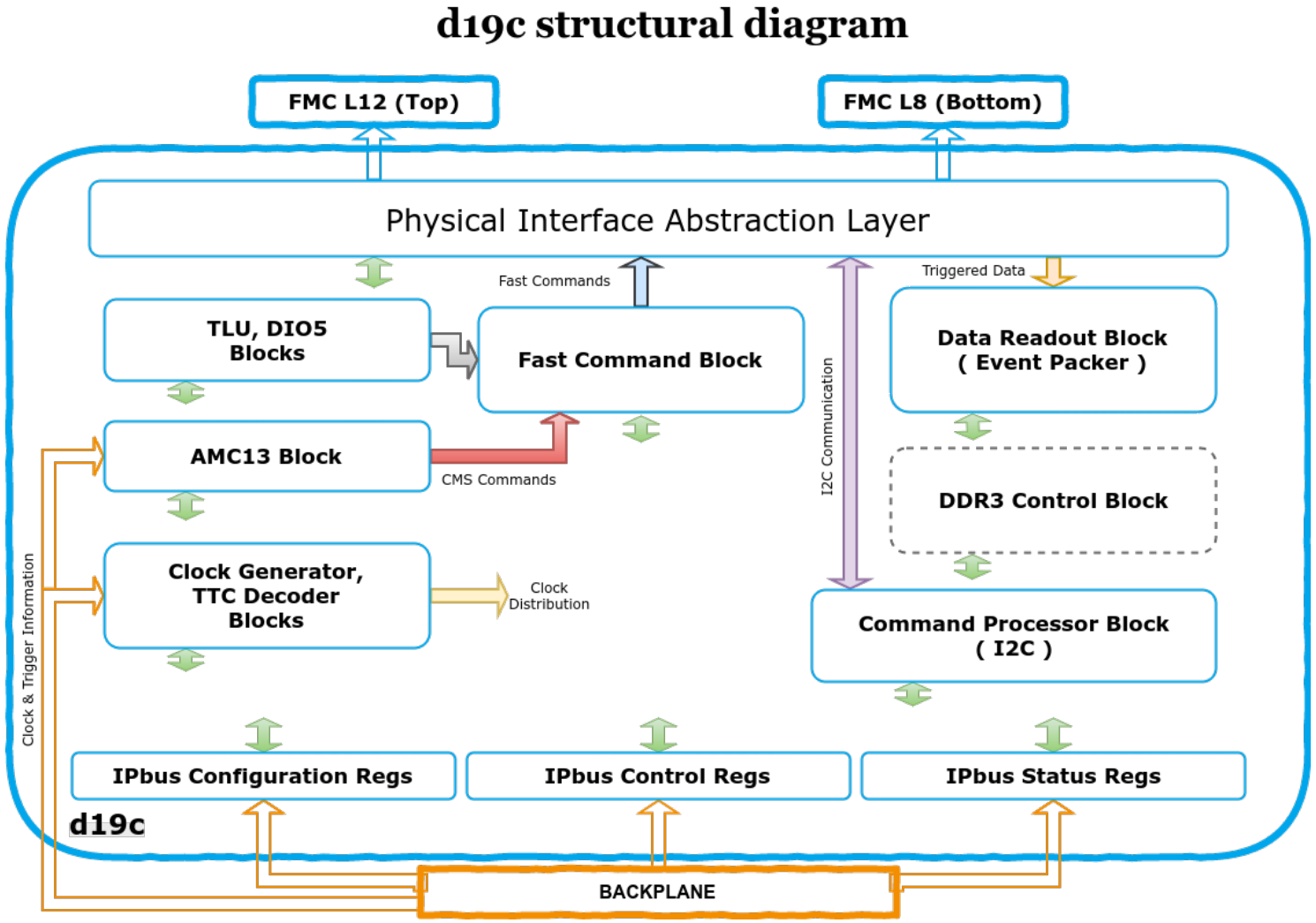
Common Blocks

- There is a set of blocks, commonly used for different hardware options. It includes such blocks, as:
  - **IPbus control/configuration/status slaves** – used for PC communication. Common interface for all blocks.
  - **AMC13/TTC/DIO5/TLU** blocks – used for clock and trigger recovery from external sources
  - **Clock Generator Block** – used for generation of the 40MHz, 160MHz, 200MHz, 320MHz clock
  - **Fast Command Block** – trigger distribution
  - **Command Processor Block** – I2C commands handling, sending them to the physical layer and handling replies
  - **IPbus DDR3 slave** – interfacing external memory chip. Input from Readout Block, output to IPbus



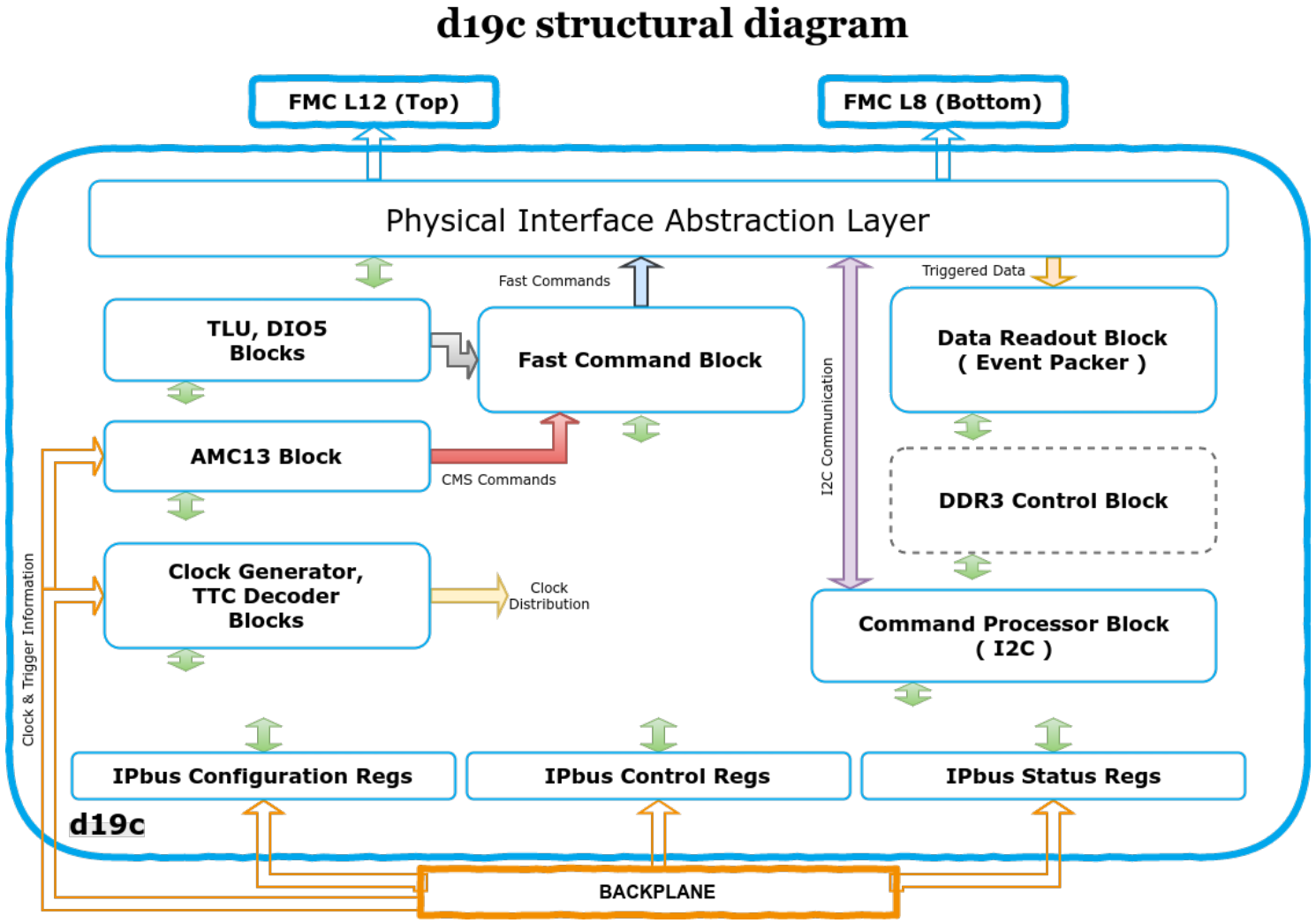
Physical Interface Abstraction Layer

- **Physical Interface Abstraction Layer**
  - Delivers the concept, where all the other blocks do not “know” about the change in hardware – all different low-level communications are hidden in this layer
  - All the other blocks are “informed” only about the amount of hybrids / chips connected to the board and their type



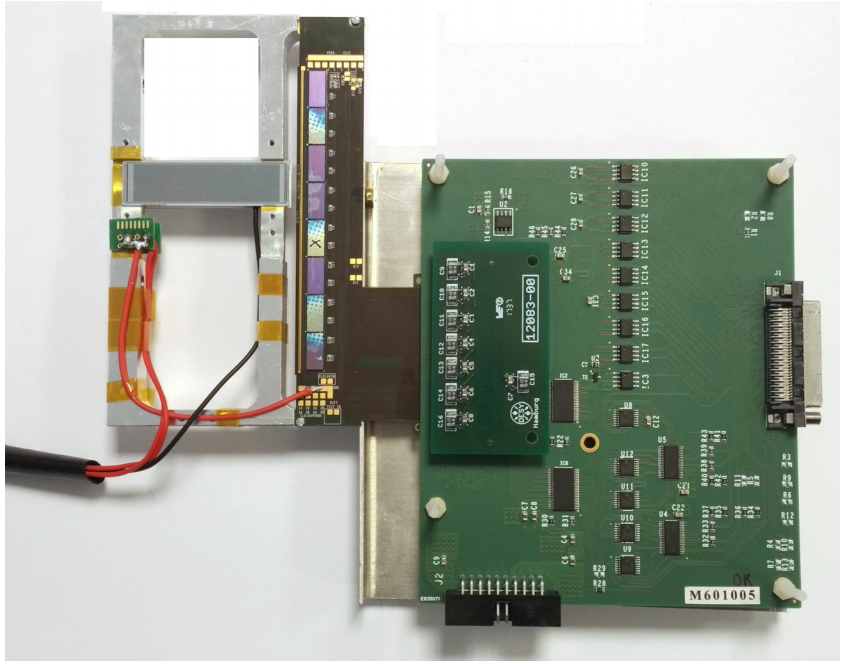
Readout Block

- **Back-End Processor** - does the event forming and packing them into the FIFO (DDR3):
  - 3 buffering stages
  - Different counters (BX, L1, TDC)
  - Two operation modes: Virgin Raw and Zero Suppressed

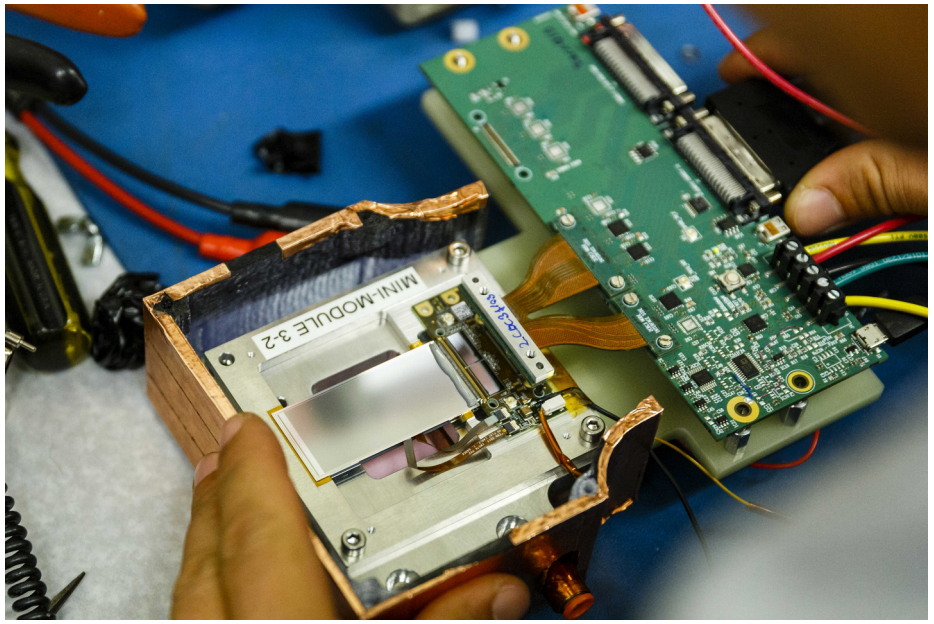


Firmware and Hardware Qualification

- **Four beam tests have already happened:**
  - 8xCBC2 @ DESY – November 2017
  - 2xCBC3 @ FNAL – November-December 2017
  - 1xMPA @ CERN – April 2018
  - 1xMPA (sensor layout studies) @ DESY – May 2018
- Also the **d19c** firmware is used for the PS (MPA, SSA) module chips qualification right now



8xCBC2 mini-module at DESY

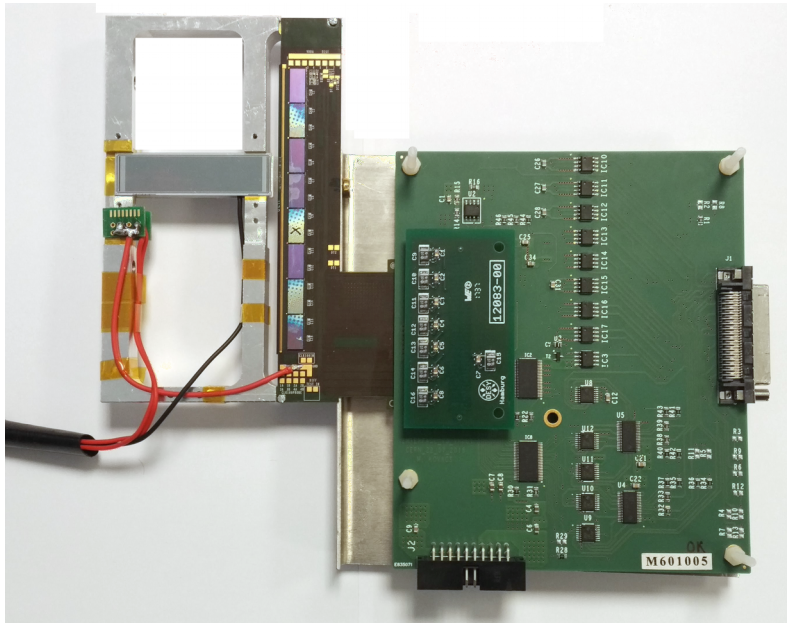


2xCBC3 module at FNAL @ Christian Dziwok

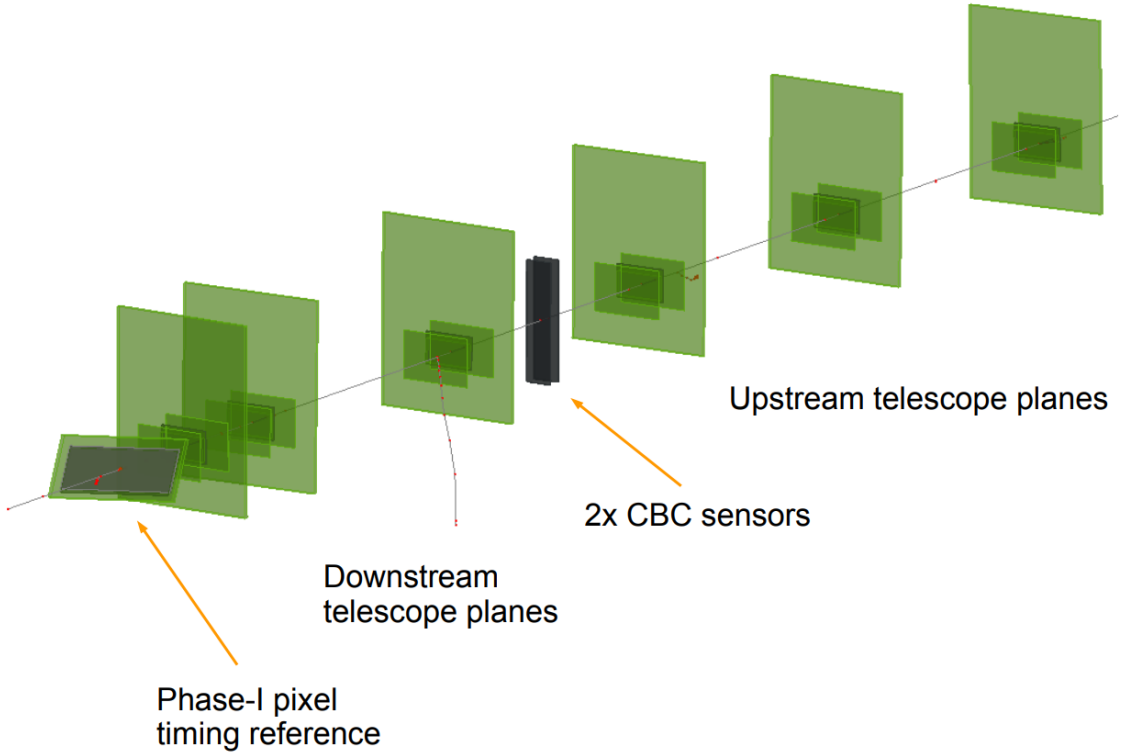
8xCBC2 Beam Test at DESY

Materials by T. Eichhorn

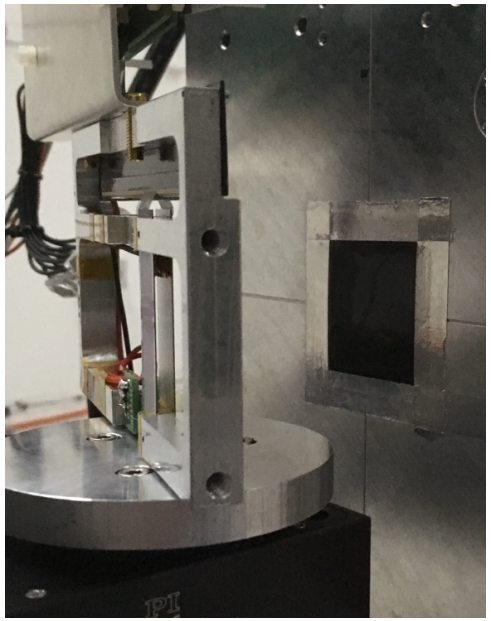
- **A module prototype was built at DESY:**
  - Based on 8xCBC2 front-end hybrid prototype
  - 2 baby sensors (127 channels each) were used
  - Each CBC has 254 channels (odd channels – top sensor, even channels – bottom sensor)
  - Only two half-chips were bonded
- **Sequence of measurements were performed at DESY II accelerator using the AIDA telescope.**



Module prototype



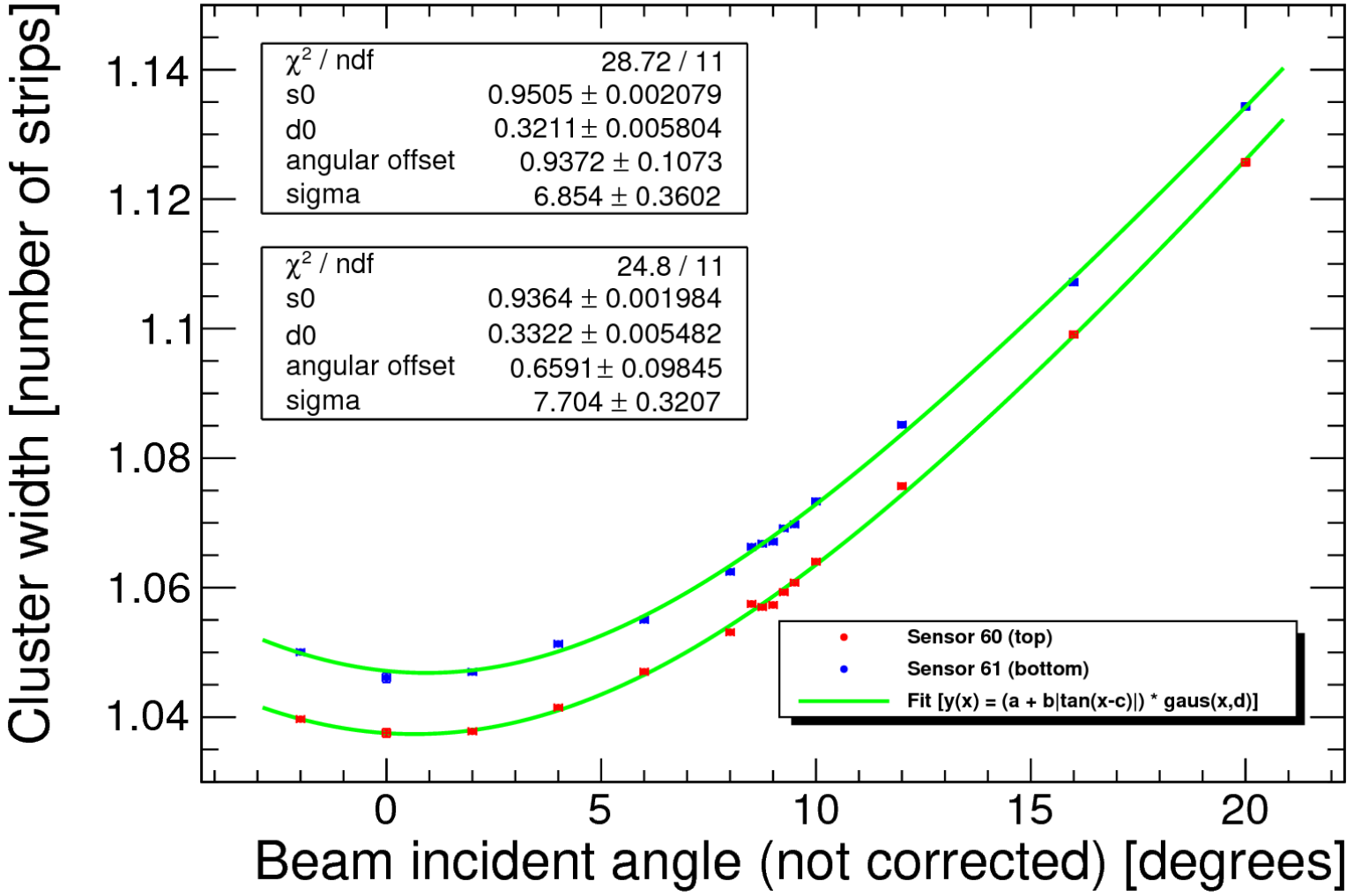
Experiment schematics



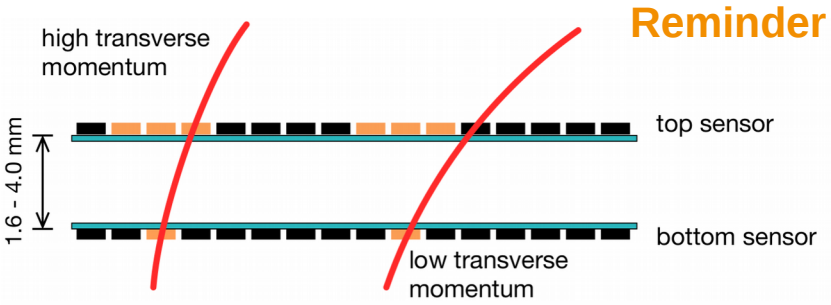
Module on the rotation stage

8xCBC2 Beam Test – Cluster Size

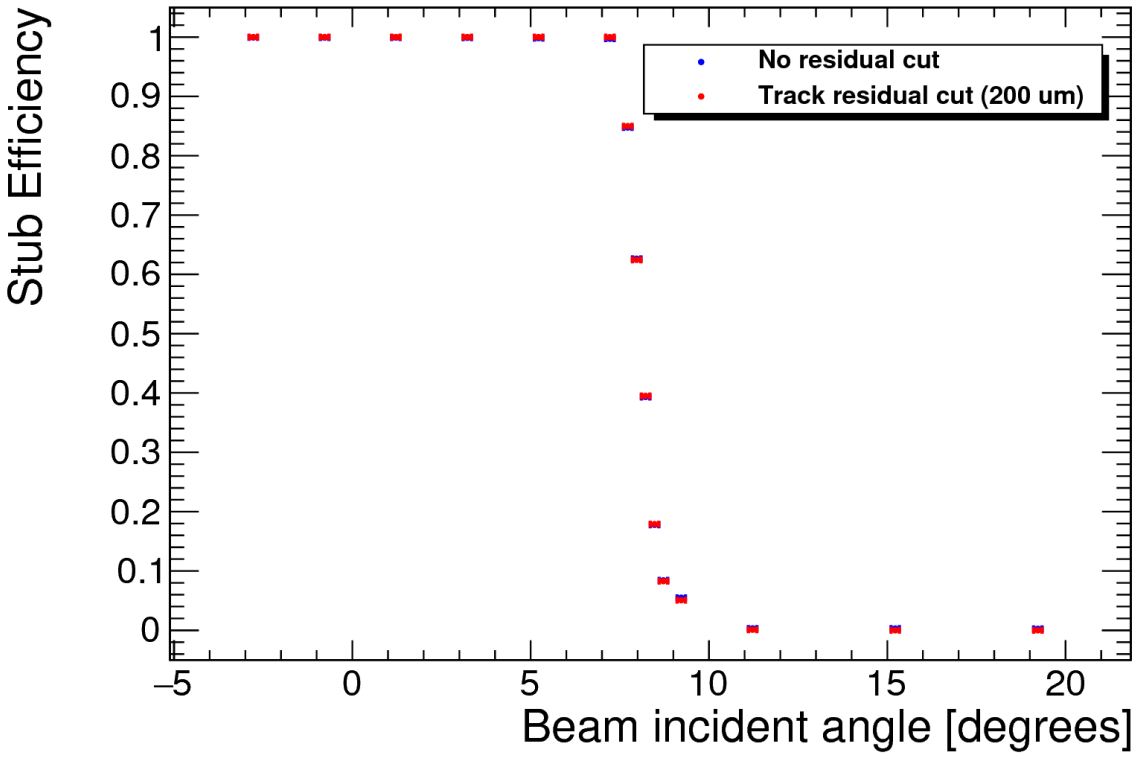
- **Cluster size distribution as the function of beam incident angle for both sensors**
- Fit using $y(x)=(s_0+d_0*|\tan(\theta-\theta_0)|)*gaus(\theta,\sigma)$
  - **s0** – cluster size at angle 0 degrees
  - **d0** – characterizes average depth of the cluster formation
  - **Θ0** – angular correction
  - **σ** – divergence of the beam
- **Several conclusions can be made:**
  - There is a different noise performance for odd and even channels
  - The beam is diverging within the spacing of two sensors (4.0 mm)
  - The correction for the angle can be extracted

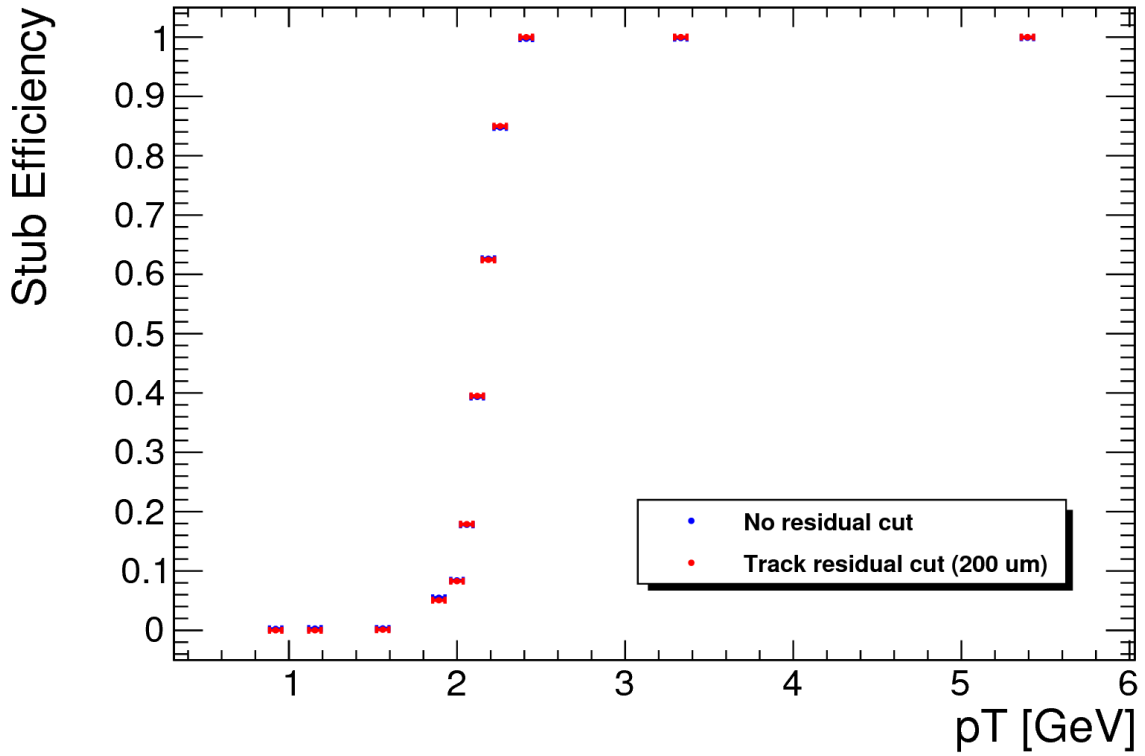


8xCBC2 Beam Test – Stub Efficiency



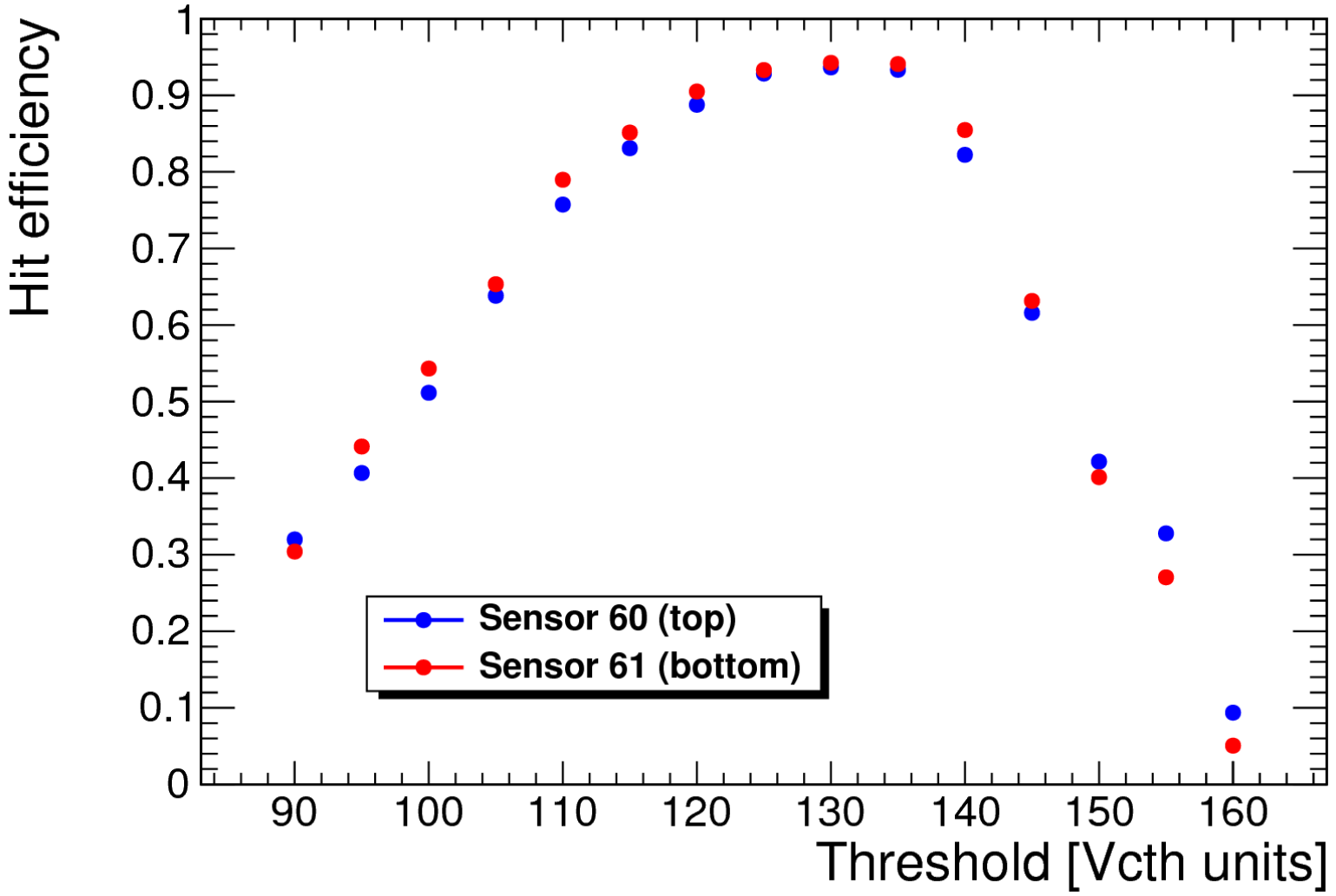
- **Stub efficiency for different incident angles, and the same plot converted into the pT space at R = 53.2 cm**
- $r_T[m] = \frac{p_T[GeV/c]}{0.3B[T]}$ ⇨ $\sin(\alpha) = \frac{R}{2 * r_T}$ ⇨ $p_T[Gev/c] = \frac{0.57\,R[m]}{\sin(\alpha)}$





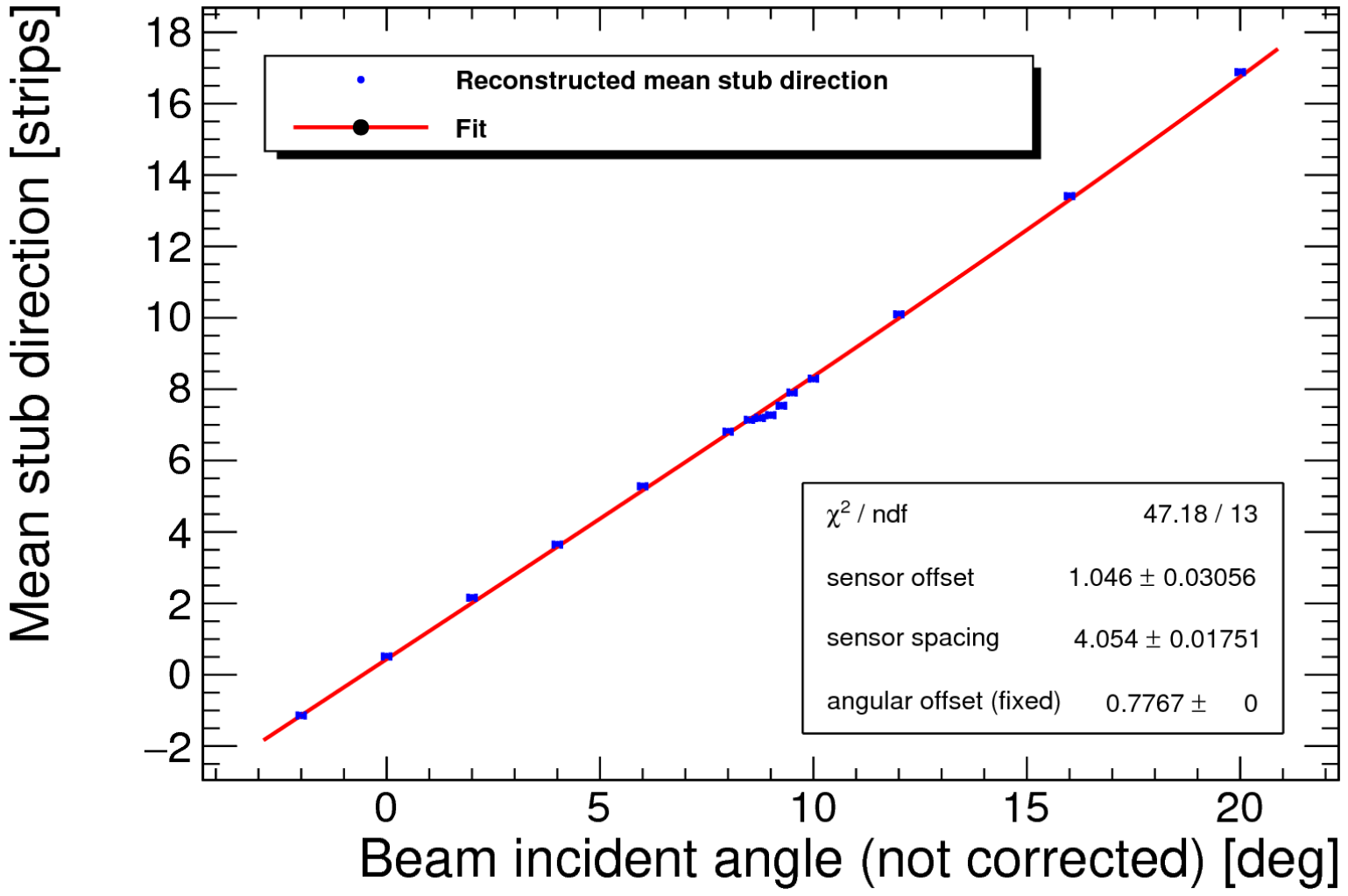
8xCBC2 Beam Test – Hit Efficiency

- **Hit efficiency as a function of the threshold**
  - Very similar for two sensors
  - The fact, that the detector is inefficient for some particle arrival phases should be taken into the account



8xCBC2 Beam Test – Mean Stub Direction

- **Mean Stub Direction as the function of the beam incident angle**
- Fit using $\Delta X = \Delta X_0 + \frac{d}{p}\tan(\theta_{rot} + \theta_0)$ [1]
  - **ΔX** – mean stub direction (distance between two cluster centers)
  - **ΔX0** – offset between two sensors
  - **d** – sensor spacing, **p** – sensor pitch
  - **Θ0** – angular missalignment
- **Several conclusions can be made:**
  - Due to absence of points for higher angles, the angular offset had to be taken from the cluster size fit
  - The sensor offset of 1.046 strip was extracted
  - The sensor spacing of 4.054 mm was extracted (matching with the expected value)



[1] Harb, Ali - DESY-THESIS-2017-031

Summary and Outlook

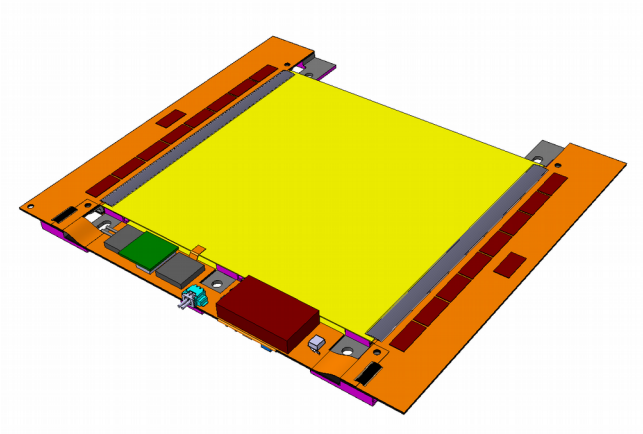
- **Since the beginning (October 2016), the project progressed significantly:**
  - Most of the groups within the collaboration have already switched to the d19c based module readout
  - Four beam tests were done using this DAQ system
  - The system is used now for the qualification of the new hardware (PS module chips)
  - The Inner Tracker team is planning to inherit some developments for their setup
- **Full firmware documentation is available – and being extended**
- **The collaboration-wide firmware review has happened in March 2018 with a highly positive outcome:**
  - "To conclude, the review committee agreed unanimously that the project is very well thought and partitioned and that significant progress has been made to date. The presentations were well structured and detailed enough to allow the committee to be able to provide constructive feedback."
- **Still lots of work has to be done:**
  - Test beam data analysis
  - Consolidation based on the review outcome
  - Soon DESY gets an MPA (pixel readout chip) assembly – new possibilities for testing
  - Getting ready to the new hardware arriving – CIC (data-packing) chip, Optical Readout

Questions?

BACKUP

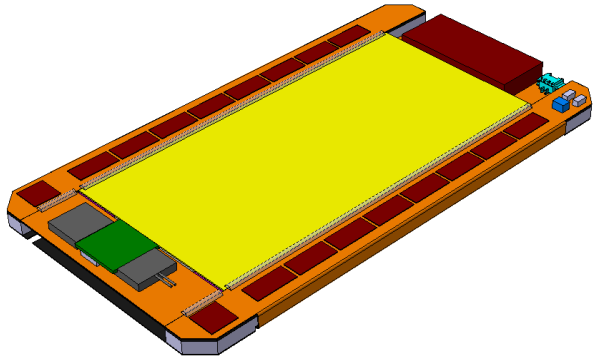
Modules – Closer Look

- **2S module:**



- **Consists of:**
  - Service hybrid (optical connection)
  - Power hybrid (biasing, powering electronics)
  - **Two** silicon **strip** sensors
  - **Two front-end hybrids**: each contains **8 x CBC** chips (strip readout and correlation between sensors) and **1 x CIC** chip (data packing)

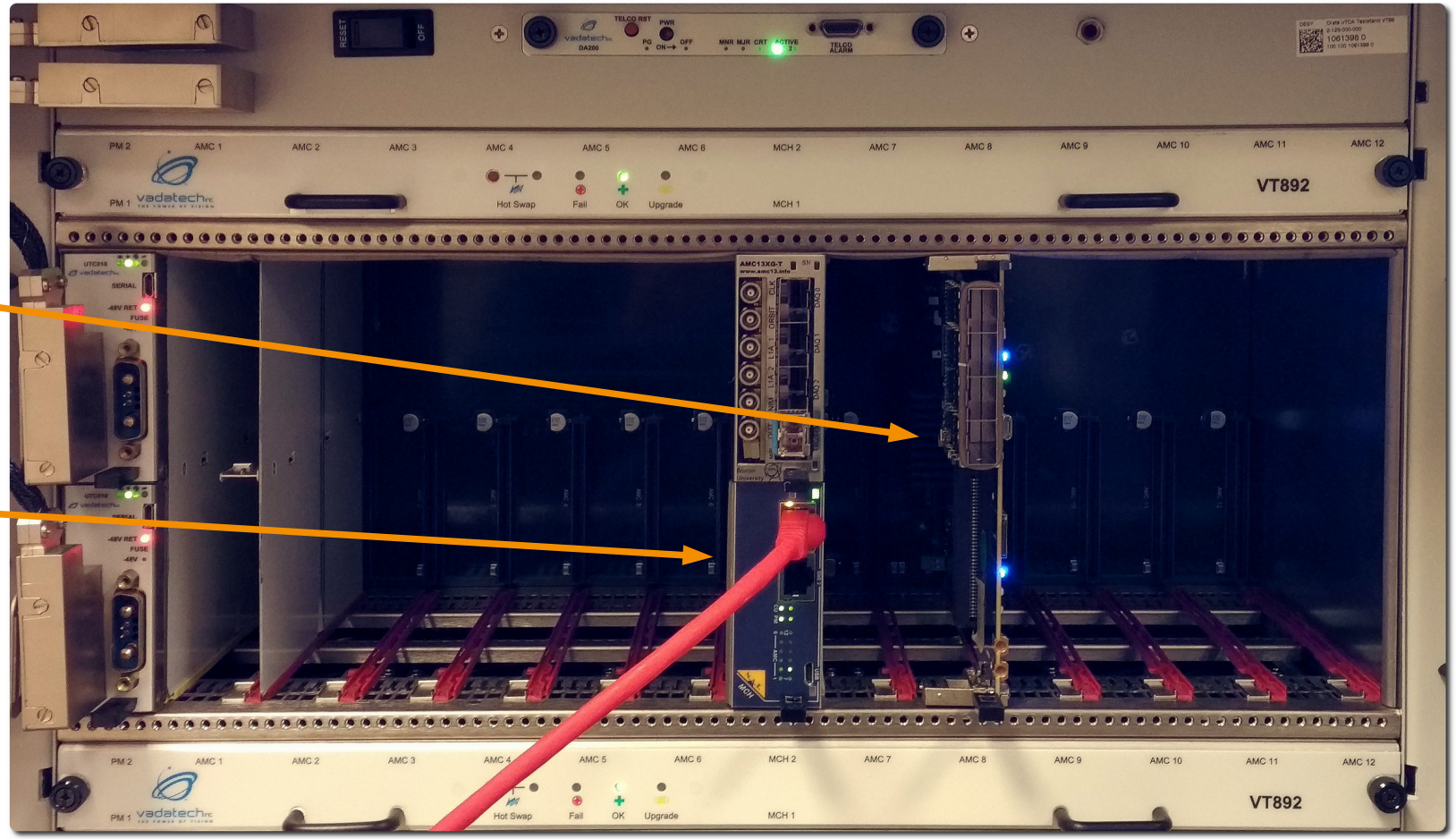
- **PS module:**



- **Consists of:**
  - Service hybrid (optical connection)
  - Power hybrid (biasing, powering electronics)
  - **One** silicon **strip** sensor
  - **One** silicon **pixel** sensor bump-bonded to the readout – **16 x MPA chips** (pixel readout and correlation between sensors)
  - **Two front-end hybrids**: each contains **8 x SSA** chips (strip readout) and **1 x CIC** chip (data packing)

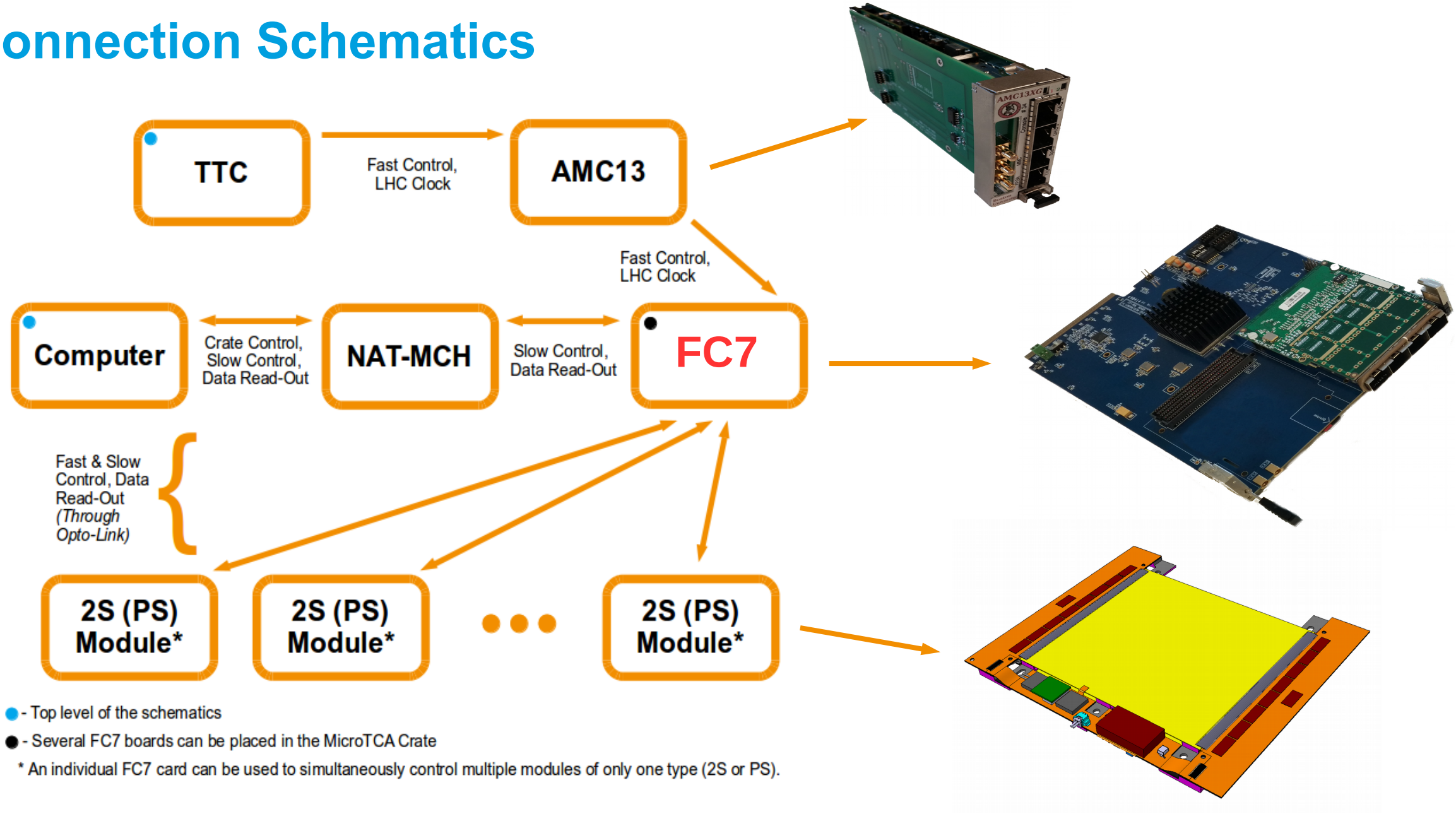
MicroTCA Standard

- **The MicroTCA standard is used as the basis for the test DAQ system**
- Scalable standard borrowed from the telecommunications
- **Advanced Mezzanine Card** (AMC) - printed circuit board (PCB), that can be hot-plugged into the MicroTCA crate
  - May contain CPU, **FPGA**, etc.
- **MicroTCA Carrier Hub** (MCH) – control and data switching unit



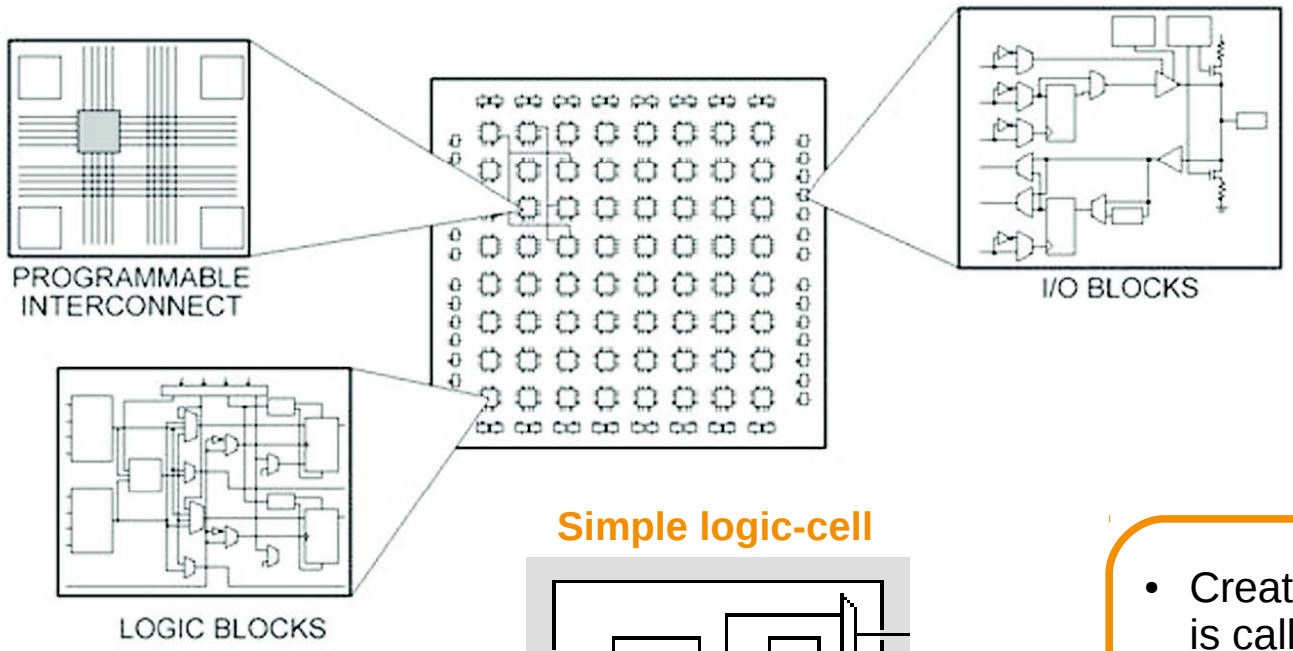
VADATECH MicroTCA Crate

Connection Schematics



What is a FPGA?

- **FPGA (Field-Programmable Gate Array) –** contains an array of programmable logic blocks, which can be “wired together” using reconfigurable interconnects
- **Very useful for huge and parallel data flows**

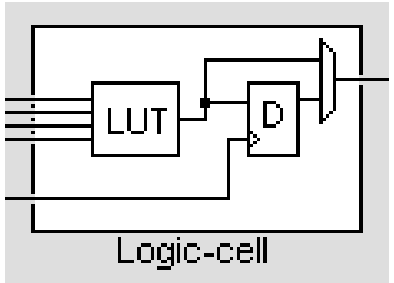


Simple logics

VHDL:
Y1 <= B1 nor B2;

B1	B2	Y1
0	0	1
0	1	0
1	0	0
1	1	0

Simple logic-cell



- Creation of the configuration file for FPGA is called “digital design” or “firmware development”.
- A user-specific **firmware has to be developed** to implement the desired logic.

Current Status

- The **d19c** firmware is already able to read & control (electrically) the following hardware options:
  - 2xCBC2 (up to 2 hybrids)
  - 8xCBC2 (up to 2 hybrids)
  - 2xCBC3 (up to 2 hybrids)
  - 8xCBC3 (up to 1 hybrid, ***not tested yet***)
  - 1 x SSA (up to 2 single-chip carriers)
  - 1 x MPA (up to 2 single-chip carriers)
  - 1 x (MPA+SSA) (up to 2 assemblies, ***not tested yet***)
- **This means, that we have one solid code block, which can be configured in order to control different types of hardware**

Middleware

- **Ph2 ACF (the d19c if fully implemented in the Ph2 ACF):**
  - A middleware API layer, implemented in C++, which wraps the firmware calls and handshakes into abstracted functions
  - A C++ object-based library describing the system components (CBCs, Hybrids, Boards) and their properties(values, status)
  - several utilities (like visitors to execute certain tasks for each item in the hierarchical Item description)
  - a tools/ directory with several utilities (currently: calibration, hybrid testing, common-mode analysis)
    - some applications: datatest, interfacetest, hybridtest, system, calibrate, commission, fpgaconfig

- **Python Scripts:**
  - Easy for debugging
  - And only for debugging
  - … Nevertheless, due to delay in the C++ implementation of the MPA readout, at last two beam tests the Python scripts were used for readout

What is d19c?

- A **d**ata **a**cquisition **c**ard, based on **FC7** with **e**lectrical or **o**ptical connectivity, able to readout **2**S or **P**S (prototype) modules, **w**ith **o**r **w**ith**o**ut the **C**IC

- To support:
  - **c**omponent prototyping and qualifications
  - **i**ntegration tests
  - **b**eam tests
  - **p**roduction **q**uality **c**ontrol

- Or **dacFC7eo2PwowoCcibpqc** in brief

Repository

https://gitlab.cern.ch/cms_tk_ph2/d19c-firmware

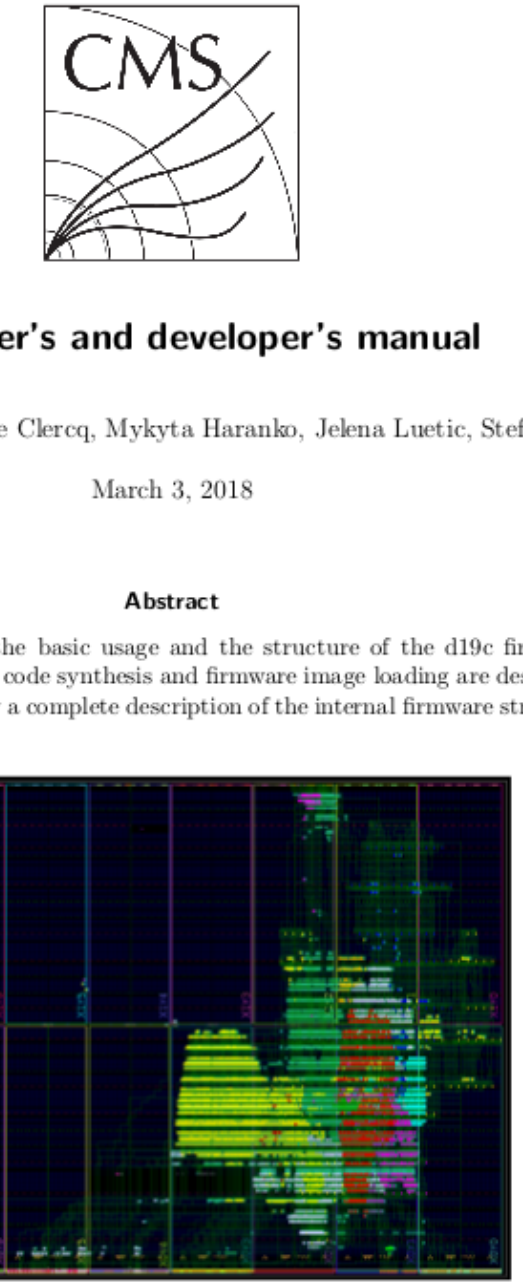
Stable branch – **master**

Use your CERN account

Manual

- The manual is available since the beginning of March 2018
- Contains getting started guidelines, the description of all firmware blocks and the overview of some basic concepts
- Can be accessed from the **GitLab** repository:

d19c user's and developer's manual



d19c user's and developer's manual

Laurent Charles, Jarne De Clercq, Mykyta Haranko, Jelena Luetic, Stefano Mersi

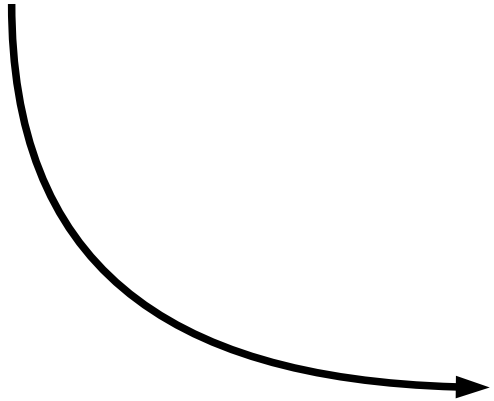
March 3, 2018

Abstract

This manual describes the basic usage and the structure of the d19c firmware. Vivado project creation, code synthesis and firmware image loading are described, and are supplemented by a complete description of the internal firmware structure.

Hardware Selection

The build configuration is done in the **user_package_basic.vhd** file



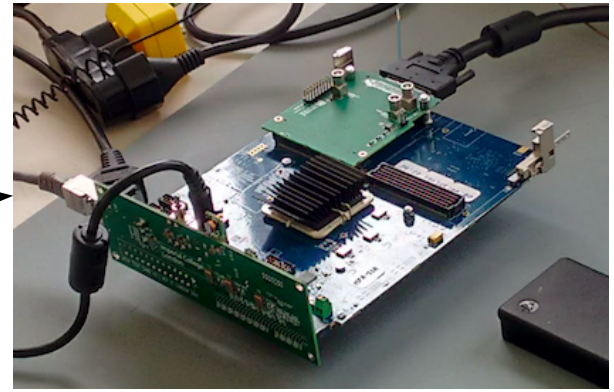
```
--====================================--
-- Important configs (have to correspond the FMC's configs)
--====================================--
-- total number of hybrids
constant NUM_HYBRIDS            : integer := 1;
-- number of chips per hybrid
constant NUM_CHIPS              : integer := 2;
--====================================--

    --====================================--
    -- Important types
type chip_type is (CBC2, CBC3, MPA, SSA);
-- implementation enum type
type implementation_type is (OPTICAL, ELECTRICAL, EMULATION);
-- fmc connection enum type (enumarates, which hardware can be connected to the FMC)
type fmc_hardware_type is (FMC_NONE, FMC_DIO5, FMC_2CBC2, FMC_8CBC2, FMC_1CBC3, FMC_2CBC3,
                           FMC_8CBC3_FMC1, FMC_8CBC3_FMC2, FMC_OPTO_QUAD, FMC_MPA_SSA_BOARD, FMC_FERMI_TRIGGER_BOARD);
-- readout type
type readout_type is (FIFO, DDR3);
--====================================--

--====================================--
-- Hardware Config
--====================================--
-- chip type
constant CHIP                   : chip_type := CBC3;
-- which implementation? electrical or optical? that is the question!
constant IMPLEMENTATION         : implementation_type := ELECTRICAL;
-- specifies, what's connected to the FMC1 (l12)
constant FMC1                   : fmc_hardware_type := FMC_2CBC3;
-- specifies, what's connected to the FMC2 (l8)
constant FMC2                   : fmc_hardware_type := FMC_NONE;

--====================================--
-- MISC Config
--====================================--
constant READOUT_BUFFER_TYPE        : readout_type := FIFO;
constant SLVS_DEBUG_ENABLED         : boolean := true;
-- valid options only "160MHz" or "240MHz"
constant sys_phase_mon_freq         : string   := "160MHz";
-- just needed for stats for middleware (version 1 appeared when merging cbc3 with mpa branches)
constant I2C_MASTER_VERSION         : natural := 1;
--====================================--
```

Selected constraint files have to match settings in this file

External Clocking & Triggering

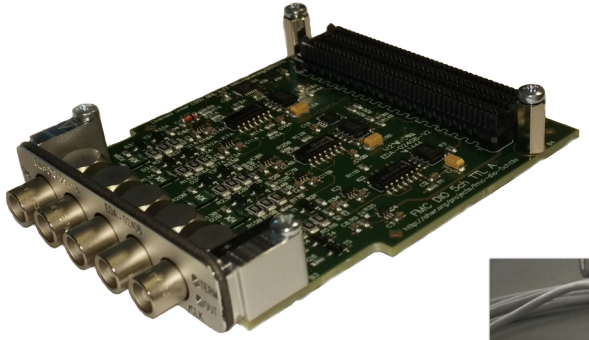
- **AMC13** Board (see user's manual)

- **Captain NIM+** (FNAL) – dedicated FMC has to be used



- **2xCBC3 Main FMC**:
  - SPARE1 – External Trigger
  - SPARE2 – External Clock

- **DIO5:**
  - Input 2 – External Trigger
  - Input 5 – External Clock



- **TLU:**
  - DIO5 is used
  - External LVDS - TTL Converter has to be built