AFCK – Hardware and Software

Wojciech M. Zabołotny

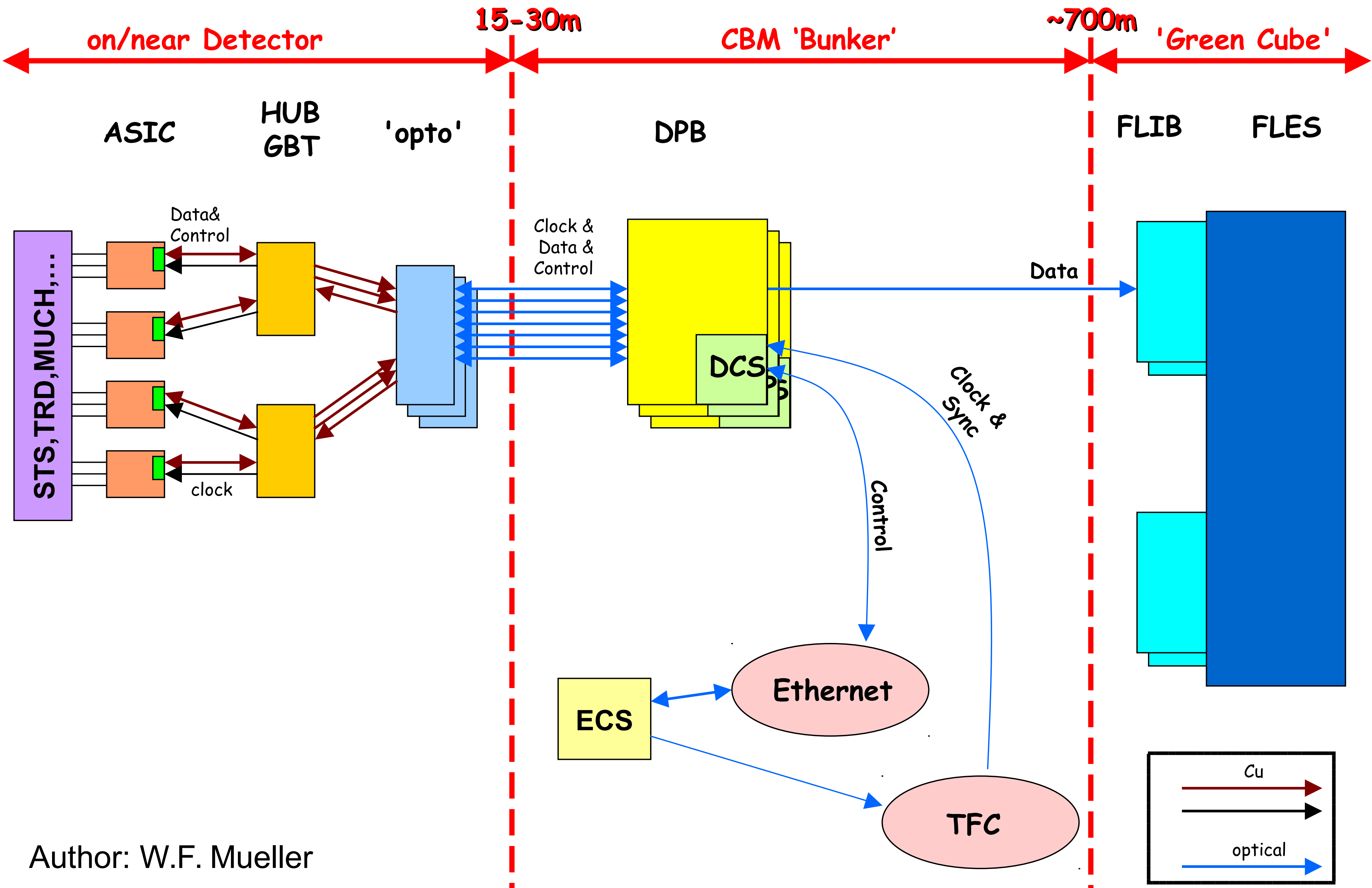
Institute of Electronic Systems, Warsaw University of Technology

(significantly based on materials provided by other members of CBM team)

Why AFCK?

- There was a need for a prototyping platform for DPB boards for CBM experiments
- What are requirements for the DPB boards?
  - The Data Processing Boards are intended to be an important component of the CBM readout chain and control system
  - They should concentrate and possibly preprocess data received from the front-end electronics, before sending them via long optical links to FLES
  - They should provide control (both fast and slow) for FE electronics
  - They should distribute reference clock and timing to the FE electronics

CBM Data Flow (ASIC based FEE)



Author: W.F. Mueller

Requirements for DPB boards

- The Data Processing Boards are intended to be an important component of the CBM readout chain and control system
- They should concentrate and possibly preprocess data received from the front-end electronics, before sending them via long optical links to FLES
- They should provide control (both fast and slow) for FE electronics
- They should distribute reference clock and timing to the FE electronics
- From the above requirements we can see, that the DPB boards should provide quite complex communication capabilities
- For prototyping we needed versatile board, allowing to check and verify different concepts, related to possible solutions of communication interfaces and associated firmware

MTCA.4 (for Physics) as DPB platform

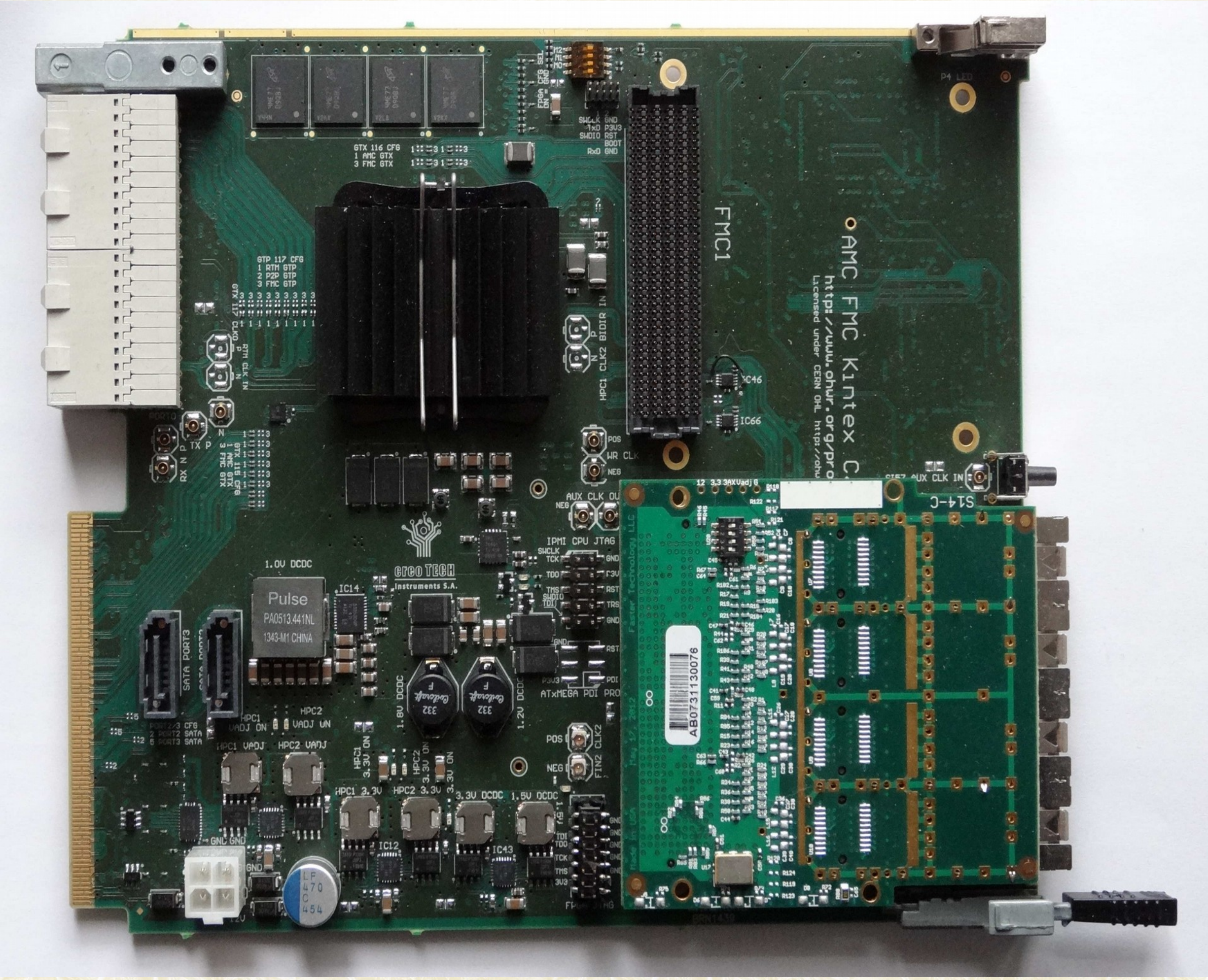
- Micro TCA Carrier Hub 1 (MCH1) with clocking, Ethernet switch is standard interface.
- 12x custom AMC with FPGA and 6 QSFP optical transceivers. On mid-height AMC.0 double-width board, one can fit 24 + 8 optical links into it. Optionally, one of AMC implements WR core and acts as a timing source. 8 GTX ports are routed to the RTM connector.
- Rear Transition Module (RTM) can be used to further extend number of optical links
- JTAG Switch Module. Used to communicate between MCH1, MCH2 and 12 AMCs for remote debugging (chipscope) and FPGA upgrade
- MCH2 - optional, redundant carrier hub with White Rabbit switch, crosspoint switch, low jitter clock distribution circuit and JTAG master port. WR management port can be connected to the general Ethernet network
- Optional AMC with 5 White Rabbit ports (SFPs). In this configuration, the crate may also perform function of WR switch
- MTCA.4, 8U crate with JTAG module (JSM), redundant power supply module and dual fan tray. VT811 from Vadatech is recommended.
- Optional AMC CPU with x86 Intel processor, connected do all AMCs using PCIe interface via MCH PCIe switch
- Optional RTM modules with SFP+ or QSFP connectors

Existing AFC board

- There was existing Open Hardware AMC FMC Carrier (AFC) board
- Artix based, flexible board in MTCA standard
- After replacement of Artix FPGA with bigger and faster Kintex, and adding even more flexible clocking and communication functionalities it was converted to AFCK, which may be used for development of DPB

AFCK board

- Available as Open Hardware solution on OHWR website
- Ready to work in the MTCA crate, but usable also in stand alone mode
- This is a prototyping platform, not a final DPB solution!



Programmable resources

- Module Management Controller (MMC) - LPC1764FBD100 – software may be modified by the user
- Xilinx Kintex 7 325T FFG900 FPGA
  - 326080 logic cells (50950 slices)
  - 840 DSP slices
  - 890 1kb BRAMs, 10 CMTs, 1 PCIe, 16 GTX

Memory resources

- 2 GB(16 Gb) of DDR3 SDRAM with 32-bit interface and 800 MHz clock – for huge amount of data requiring relatively slow access
- SPI Flash for FPGA configuration (accessible from MMC)
- SPI Flash for user data storage
- EEPROM with MAC and unique board ID
- 16020 kb of data in FPGAs BRAMs – for data requiring high speed parallel access

High speed communication capabilities

- 2 HPC connectors for 2 single width FMC or one dual width FMC
  - Up to 4 GTXs may be routed to each FMC
  - In the prototype: GTXs in FMC1 are proven to work at 10 Gbps, while GTXs in FMC2 up to 5 Gbps (in next revision higher speed in FMC2 should be achievable)
- GTXs available in FMC connectors may be optionally routed to the RTM connector
- Another 8 GTX transceivers are available at AMC FP ports

  (in MTCA backplane)

Examples of Ethernet connectivity

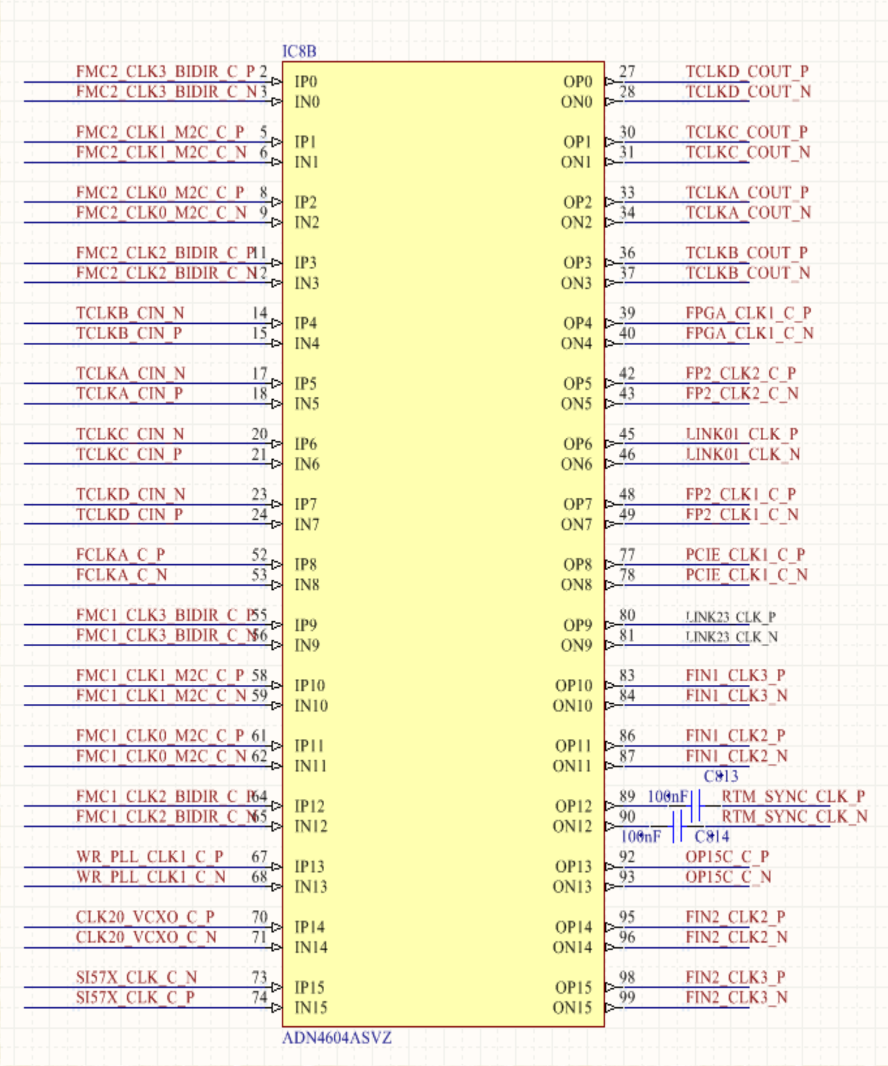
- FMC board with SFP+ cages (there are COTS boards with 4 SFP+ cages)
- Connection to the standard MTCA backplane – Ethernet lines or Ext. FP lines
- Connection to the mounted SATA connector (and further via COTS or self made SATA ↔ SFP adapter)

Other connectivity functions

- Mini USB connector connected to MMC
- Mini USB with UART converter connected either to FPGA or to MMC
- SATA connector connected to AMC PORT 2 and 3, which may be routed to FPGA GTX

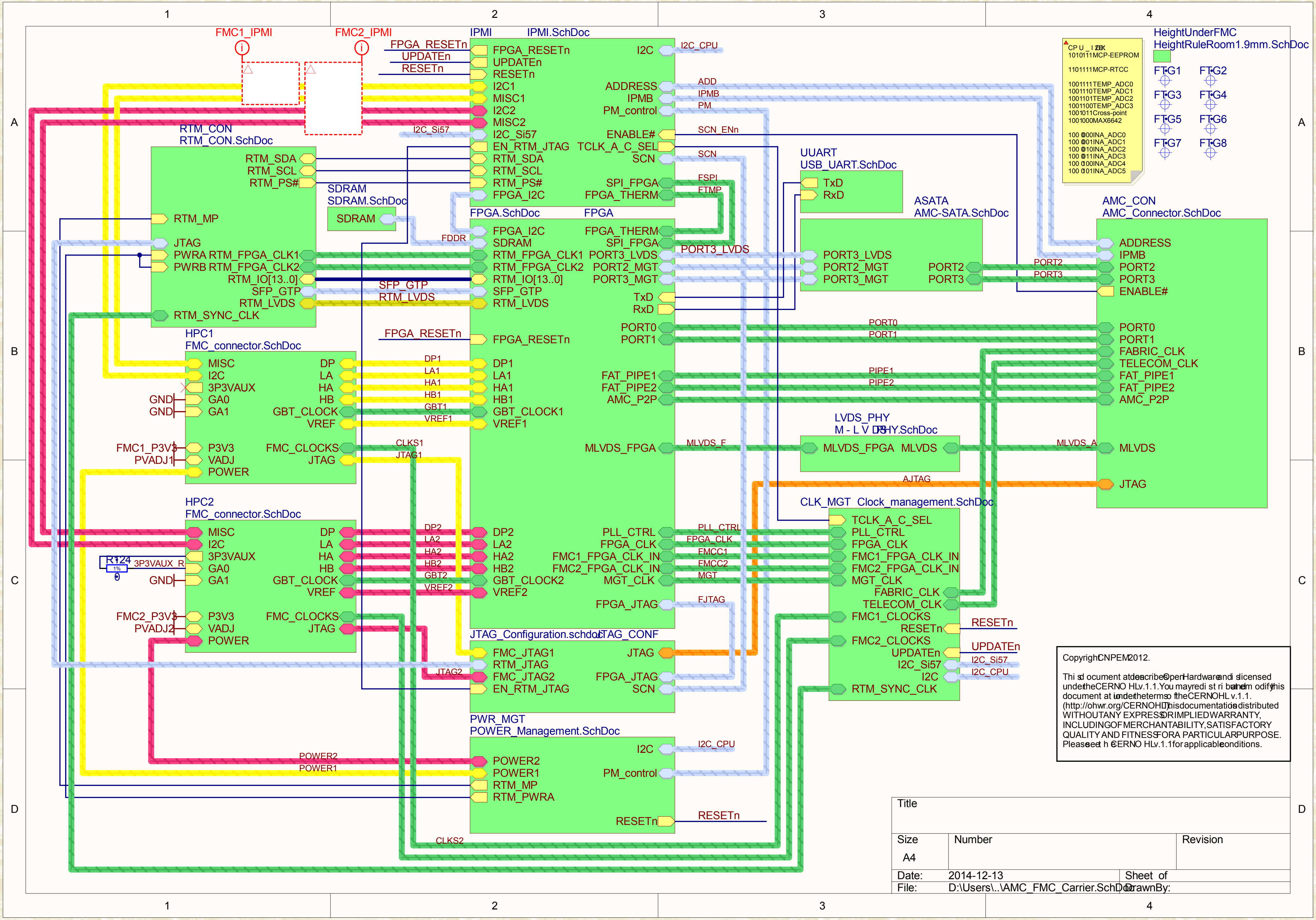
Flexible clocking

- Why do we need flexible clocking?
- Different clock domains:
  - GBT-FPGA – 120MHz or 40MHz
  - 1 Gbps Ethernet/WR – 125 MHz
  - 10 Gbps Ethernet – 156.25 MHz
- Clock distribution circuit compatible with White Rabbit
  - Based on CDCM61004RHBT and Si57X
- Jitter cleaner allowing to use clock recovered from GTX receiver to drive GTX transmitter
- Clock crossbar with 16 inputs and 16 outputs



Other functions

- Flexible JTAG
  - The JTAG chain uses the TI SCANSTA JTAG switch, allowing to access either main FPGA or FMC cards
- Monitoring capabilities
  - Temperature measurement: FMC1, FMC2, power supply, FPGA core, DDR memory
  - Monitoring of voltage and current in all FMC buses


FMC1_IPMI
FMC2_IPMI
IPMI IPMI.SchDoc
FPGA_RESETn
UPDATEn
RESETn
I2C1
MISC1
I2C2
MISC2
I2C_Si57
EN_RTM_JTAG
RTM_SDA
RTM_SCL
RTM_PS#
FPGA_I2C
I2C
ADDRESS
IPMB
PM_control
ENABLE#
TCLK_A_C_SEL
SCN
SPI_FPGA
FPGA_THERM
RTM_CON RTM_CON.SchDoc
RTM_MP
JTAG
PWRA RTM_FPGA_CLK1
PWRB RTM_FPGA_CLK2
RTM_IO[13..0]
SFP_GTP
RTM_LVDS
RTM_SYNC_CLK
SDRAM SDRAM.SchDoc
FPGA.SchDoc FPGA
UUART USB_UART.SchDoc
TxD
RxD
ASATA AMC-SATA.SchDoc
PORT3_LVDS
PORT2_MGT
PORT3_MGT
PORT2
PORT3
AMC_CON AMC_Connector.SchDoc
PORT0
PORT1
FABRIC_CLK
TELECOM_CLK
FAT_PIPE1
FAT_PIPE2
AMC_P2P
MLVDS
HeightUnderFMC HeightRuleRoom1.9mm.SchDoc
FT-G1 FT-G2 FT-G3 FT-G4 FT-G5 FT-G6 FT-G7 FT-G8
HPC1 FMC_connector.SchDoc
MISC
I2C
3P3VAUX
GA0
GA1
DP
LA
HA
HB
GBT_CLOCK
VREF
P3V3
VADJ
POWER
FMC_CLOCKS
FMC1_P3V3
PVADJ1
GND
DP1
LA1
HA1
HB1
GBT_CLOCK1
VREF1
LVDS_PHY M-LVDS_PHY.SchDoc
MLVDS_FPGA
AJTAG
HPC2 FMC_connector.SchDoc
FMC2_P3V3
PVADJ2
R124
DP2
LA2
HA2
HB2
GBT_CLOCK2
VREF2
PLL_CTRL
FPGA_CLK
FMC1_FPGA_CLK_IN
FMC2_FPGA_CLK_IN
MGT_CLK
FPGA_JTAG
CLK_MGT Clock_management.SchDoc
FMC1_CLOCKS
FMC2_CLOCKS
JTAG_Configuration.schdoc JTAG_CONF
FMC_JTAG1
RTM_JTAG
FMC_JTAG2
PWR_MGT POWER_Management.SchDoc
POWER2
POWER1
RTM_PWRA
CLKS1
CLKS2
JTAG1
JTAG2
Copyright CNPEM 2012.
This document describes Open Hardware and is licensed under the CERN OHL v.1.1. You may redistribute and modify this document under the terms of the CERN OHL v.1.1. (http://ohwr.org/CERNOHL). This documentation is distributed WITHOUT ANY EXPRESS OR IMPLIED WARRANTY, INCLUDING OF MERCHANTABILITY, SATISFACTORY QUALITY AND FITNESS FOR A PARTICULAR PURPOSE. Please see the CERN OHL v.1.1 for applicable conditions.
Title
Size
A4
Number
Revision
Date: 2014-12-13
Sheet of
File: D:\Users\..\AMC_FMC_Carrier.SchDoc
Drawn By:

Modes of operation

- The AFCK board may be used in two modes:
  - In the standard MTCA crate
  - In the stand alone mode. In this case only a single 12V power supply is necessary.

AFCK – software needed

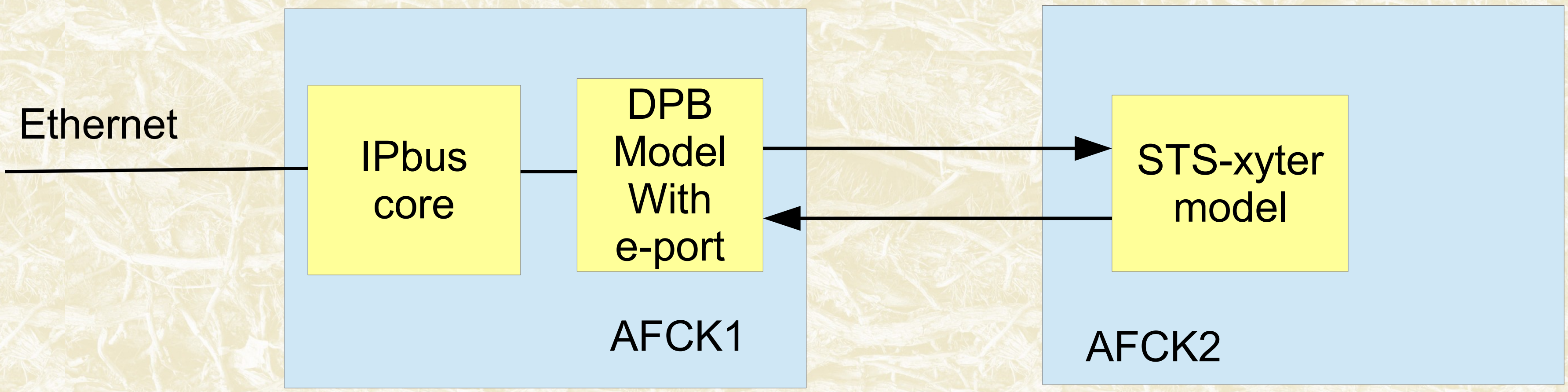
- Firmware development
  - Xilinx Vivado suite
- Configuration of the board
  - via JTAG interface (also limited control is possible)
- Communication with the board
  - Currently IPbus via Ethernet is used
  - PCIe control should be also possible
- High speed data transfer from AFCK
  - Currently the FADE protocol was tested
  - (Later on: FLES interface based solution should be added)

Usage of IPbus with AFCK

- IPbus – system for control of FPGA based electronics via 1Gbps Ethernet link https://svnweb.cern.ch/trac/cactus
- Free & Open source, but for Xilinx relies on non-free MAC core (may be replaced with simplified, open source implementation)
- Very convenient software prototyping in Python possible
- Final version of software may be implemented in C++

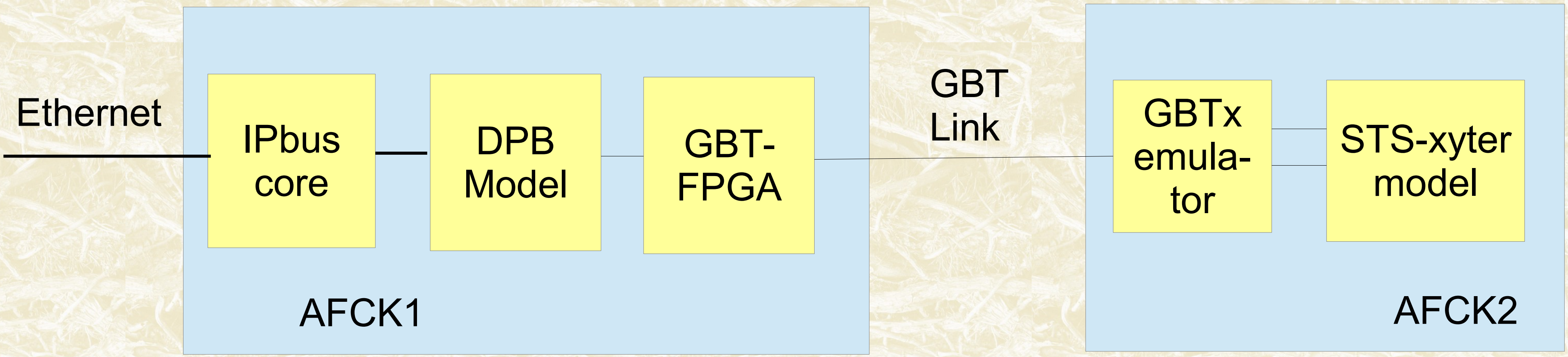
AFCK – use scenarios

- Testing of the communication with the STS-XYTER ASIC
  - Developed firmware and software for control of the STS-XYTER (eg. for link synchronization, for data transmission)



AFCK – use scenarios

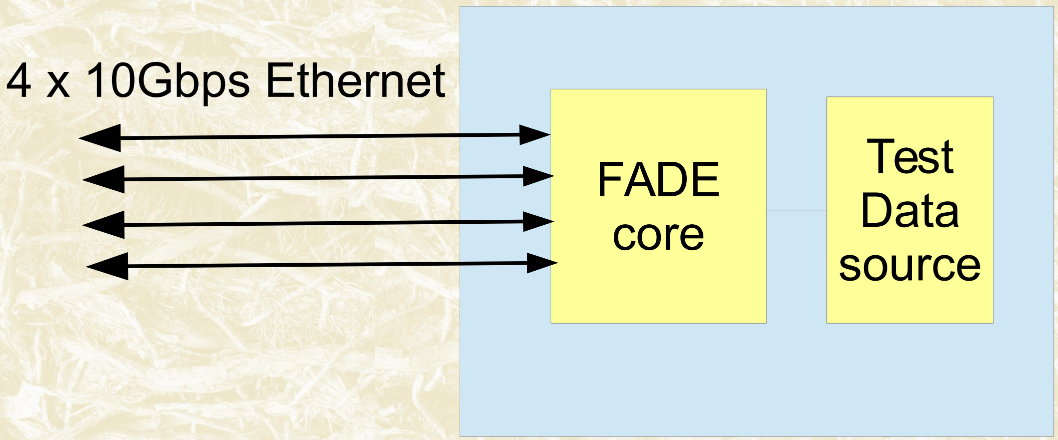
- Testing of the communication with the STS-XYTER ASIC via GBTx emulator
  - Tests of GBT-FPGA and GBTx emulator on AFCK platform (to be done in cooperation with colleagues from VECC)



AFCK – use scenarios

- Testing of high speed data transmission
  - Verification of data transmission and control via FADE protocol

    (http://opencores.org/project,fade_ether_protocol)

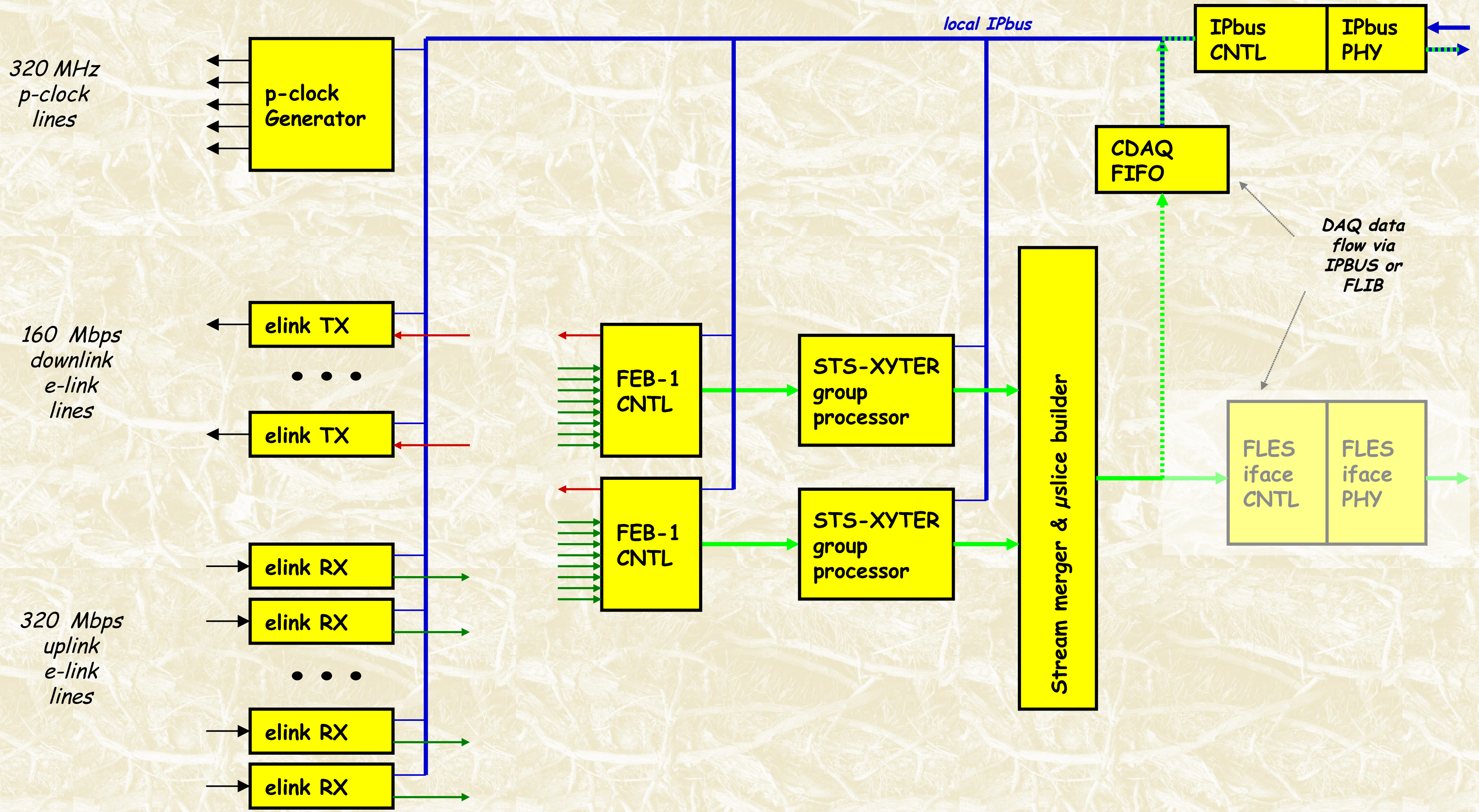


AFCK - use scenarios

- AFCK as a White Rabbit slave
  - AFCK was connected to the White Rabbit switch, working as a WR master
  - Correct operation of the White Rabbit core was confirmed
  - It is a proof of AFCK capability to provide reference clock and timing

AFCK – use scenarios

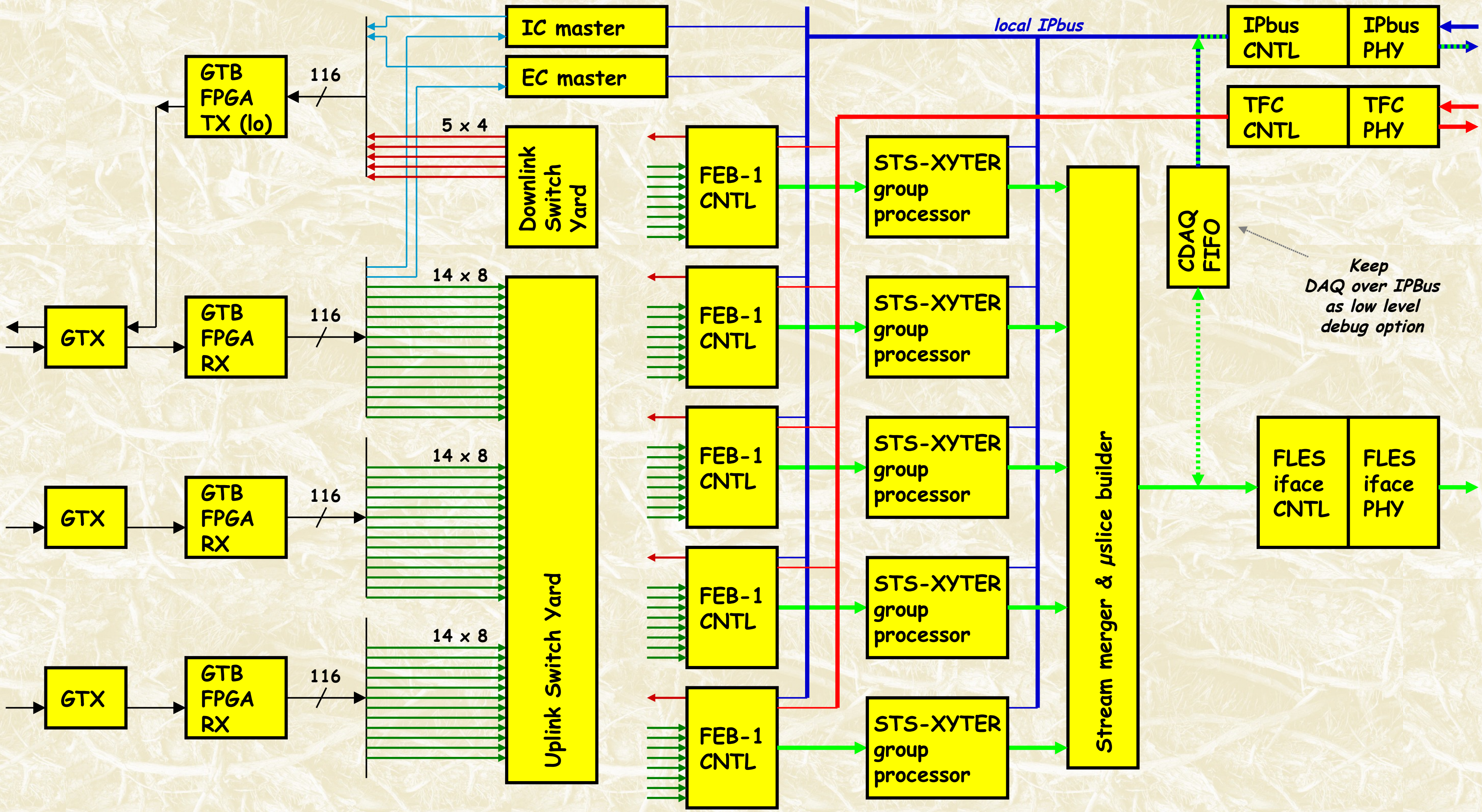
- AFCK used for minimalistic setup of real data acquisition (plan)



Slide prepared by Walter Müller for DAQ Meeting 25.03.2015

AFCK – use scenarios

- AFCK used for real data acquisition (plan)



Slide prepared by Walter Müller for DAQ Meeting 25.03.2015

What next for DPB ?

- Possible solutions should be tested on the AFCK platform
- The final, cost optimized version of the DPB board should be designed
  - Removal of unnecessary interfaces and functionalities
  - Implementation of cost optimized interfaces (e.g. multi-link transceivers like MiniPOD or MicroPOD from Avago)

Thank you for your attention!