Unlocking MicroTCA.4 for FPGA Developers: A Practical Guide

13th MicroTCA Workshop Tutorial

Cagil Gumus
Hamburg, 10 December 2024

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

DESY.

My background

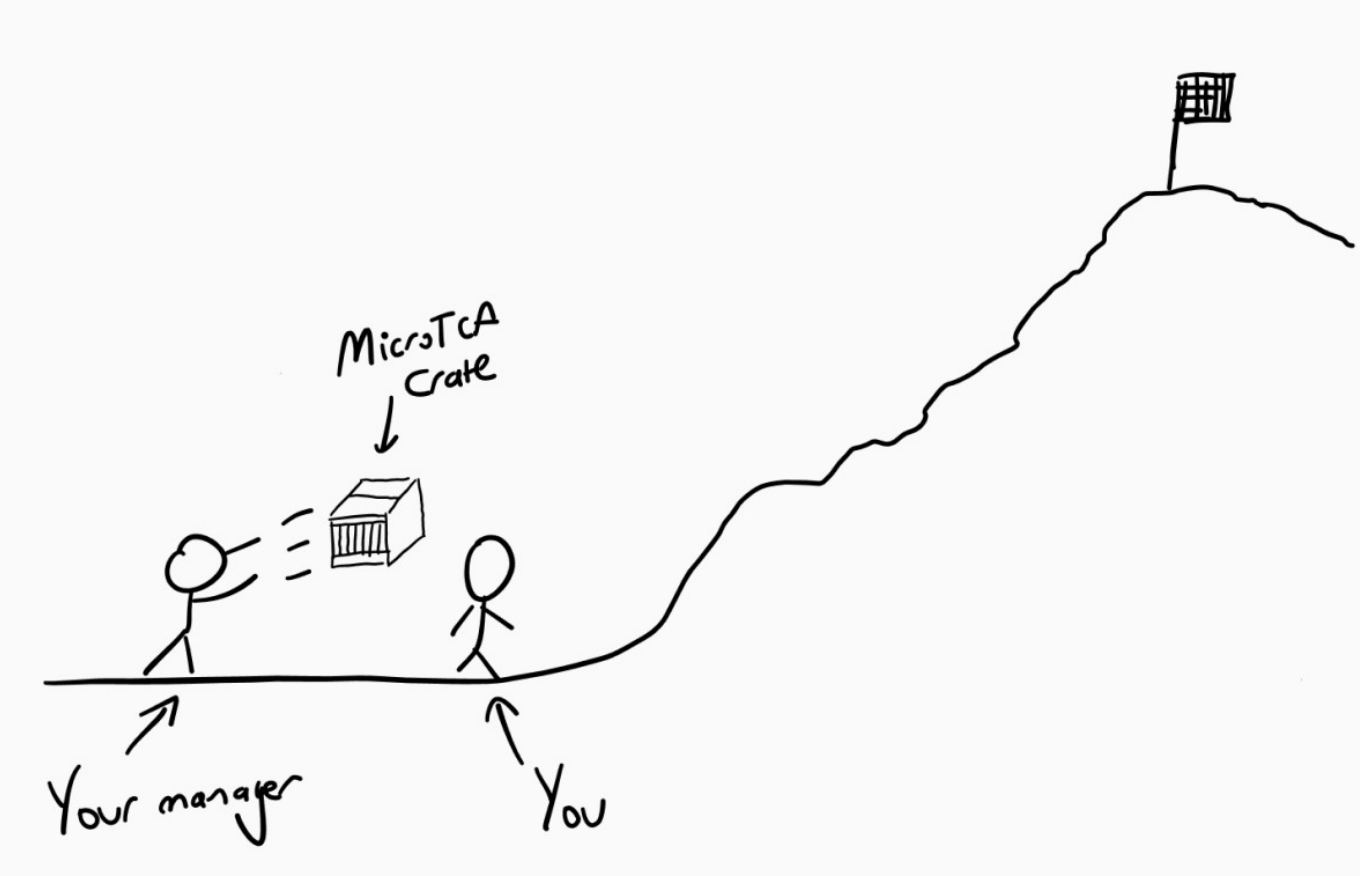
- Name: Çağıl (CJ)
- FPGA Dev | Systems Engineer
- Working with MicroTCA >7 years
- Focus Area:
  - Fast Feedback using RF
  - Synchronous Motion Control
  - DAQ Systems
- XFEL Operator (retired)

The typical scenario in MicroTCA world...

MicroTCA should not make your life harder.

The goal of this talk:

Show *some* of the important aspects of FPGA development inside the MicroTCA world.

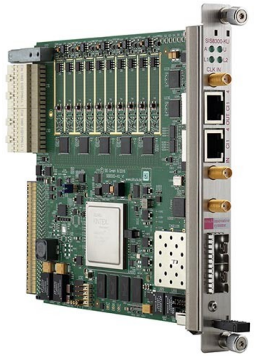


Where are the FPGAs?

What are they doing?

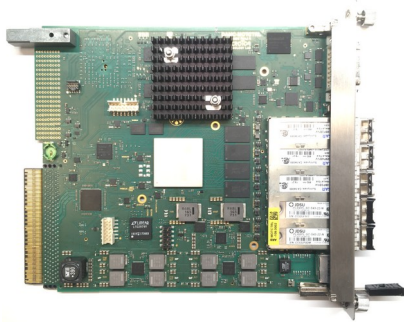
Types of Application for an FPGA in MicroTCA

Too many to count...



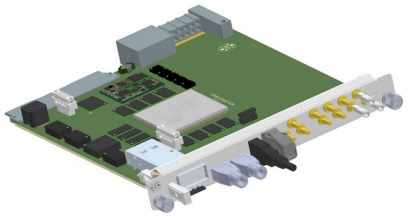
- Product:
  - SIS8300-KU
- Company:
  - Struck GmbH
- FPGA:
  - Kintex Ultrascale
  -
- Applications:
  - LLRF
  - Beam Diagnostics
  - DAQ

- Product:
  - DAMC-FMC2ZUP
- Company:
  - CaenELS
- FPGA:
  - MPSoC Ultrascale+
  - Spartan 7
- Applications:
  - Frame grabber for Plasma accelerators
  - Orbit feedback for Synchrotrons
  - Data Aggregation for Detectors



- Product:
  - DAMC-MOTCTRL
- FPGA:
  - Kintex-7
  - MPSoC Ultrascale+
- **Applications:**
  - Motion Controller Project for PETRA IV using DAMC-MOTCTRL

- Product:
  - DAMC-FMC1Z7IO
- Company:
  - d-tacq
- FPGA:
  - Zynq 7000
- Application:
  - Gravitational Wave Detector in Space

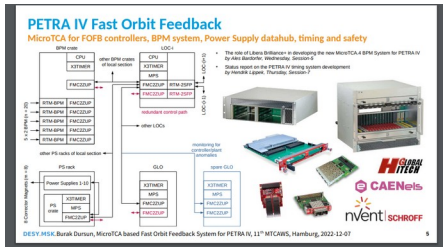


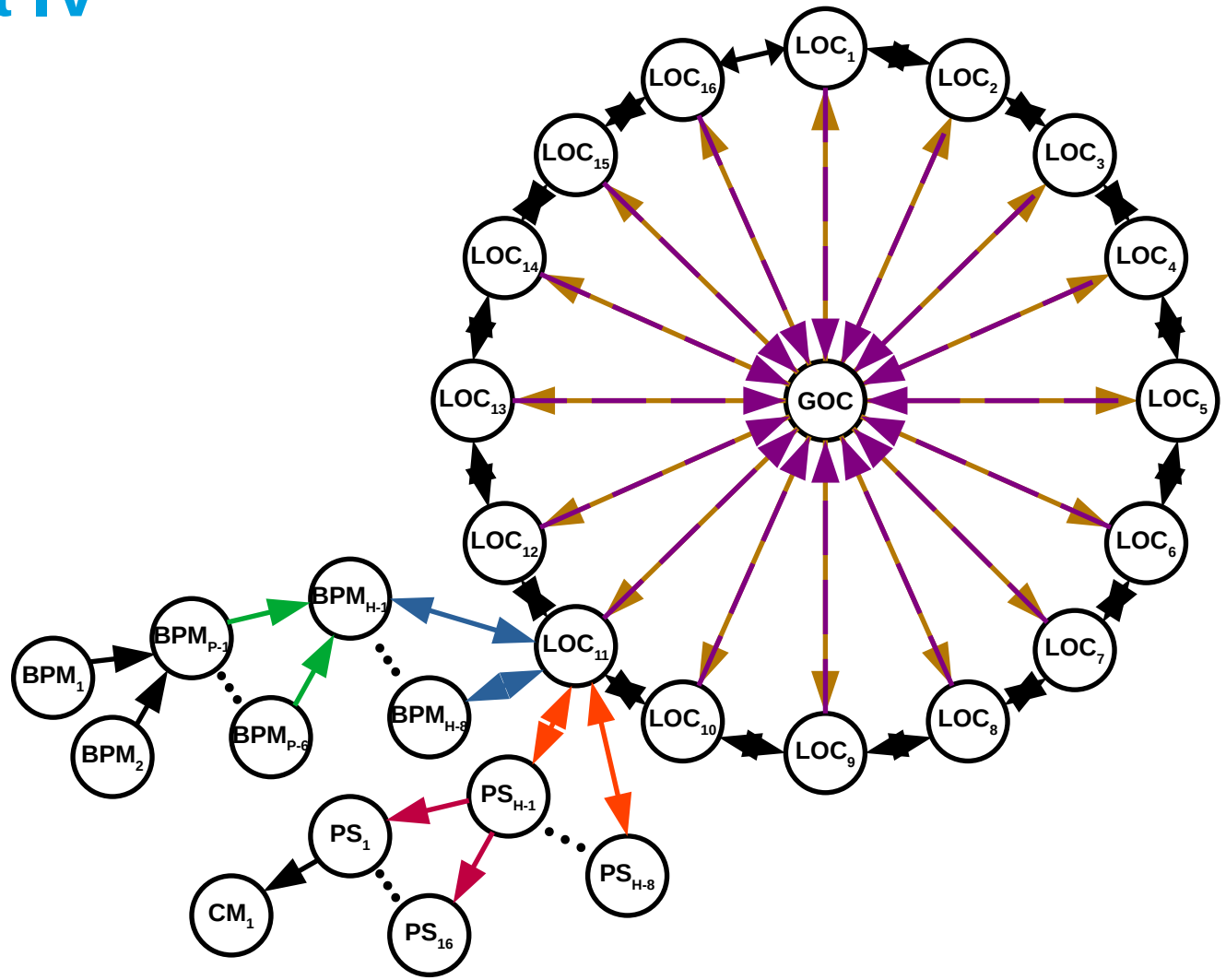
- Product:
  - DAMC-DS5014DR
- Under-design
- FPGA:
  - Zynq UltraScale+ RFSoC
- Application:
  - Multi-Bunch Feedback
  - LLRF
  - Quantum Computing

Fast Orbit Feedback at Petra IV

Things can get complicated very fast:

- Mission:
  - Stabilize the orbit of e- as it’s circling the synchrotron using magnets
  - Read bunch of beam position values and use magnets to correct it.
- > 100 MicroTCA Crates
- > 600 FPGAs working together
- Talk by Burak D. at 11th MicroTCA Workshop:





Common Tasks for an FPGA

Common Tasks of an FPGA developer

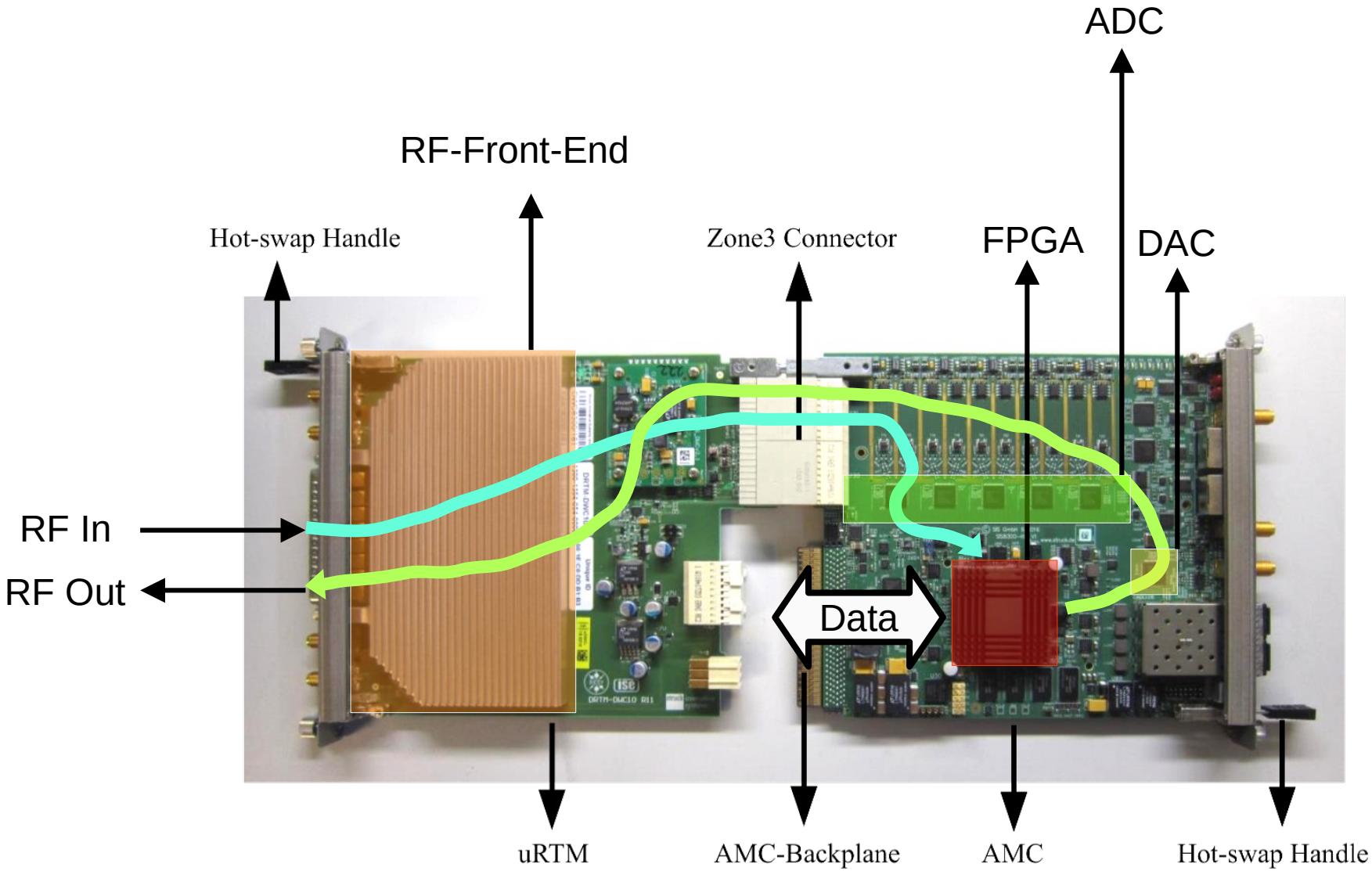
‘Things’ you’ll deal with:

- Do the ‘application’, Signal processing (eg. control a Particle Accelerator)
- **Move data around**
- **Get some signals from the backplane (Clock, Interrupt etc.)**
- **Interact with the MicroTCA Management (MMC)**
- **Remote programming (JTAG)**

Prompt: “show me an image of frustrated FPGA developer that is working with MicroTCA boards” Created with Gemini

Data Movement inside the AMC

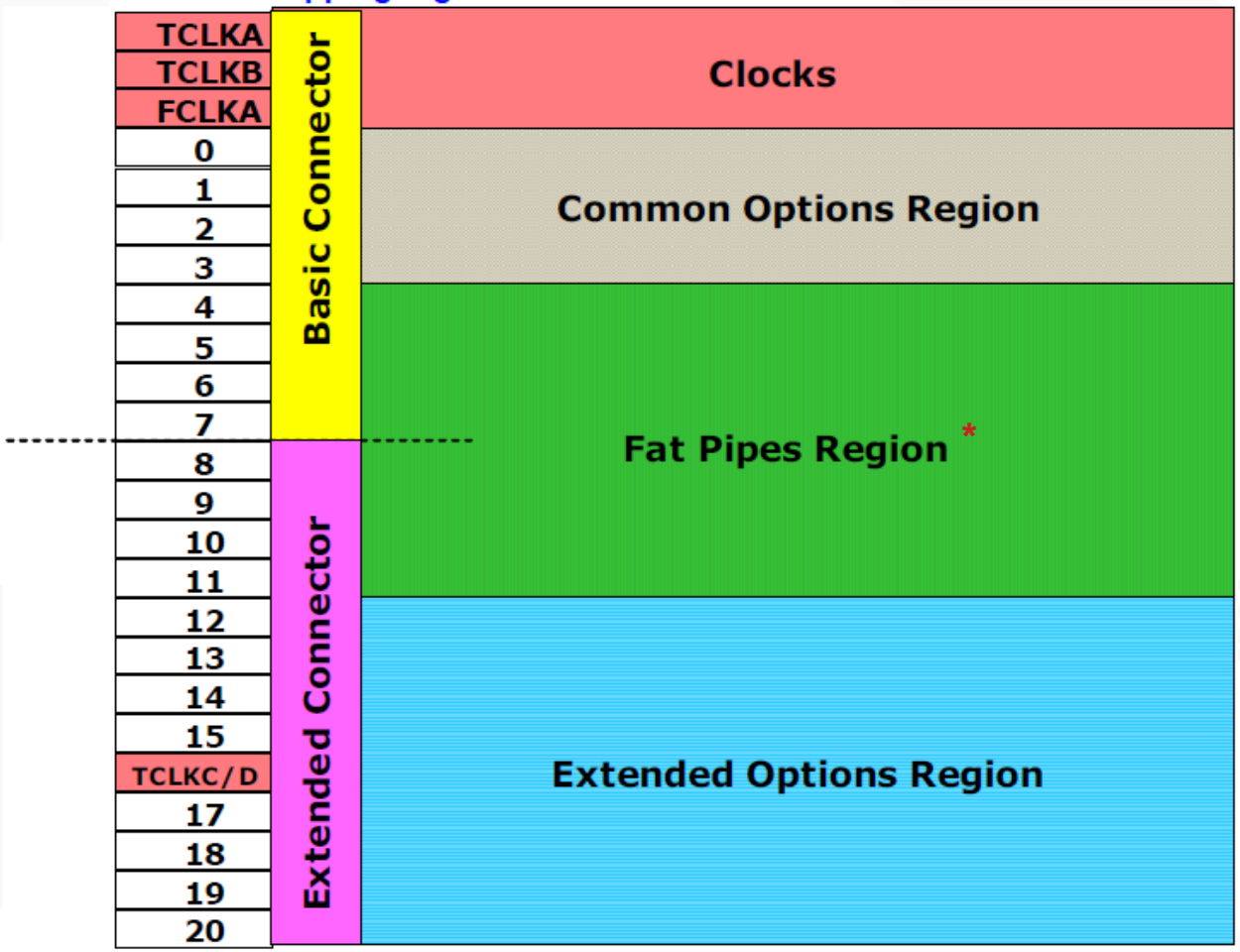
Example: Digitizing RF and Closing a Feedback



AMC Backplane

Grouping of the Ports

Figure 6-11 AMC Port mapping regions



1Gb Ethernet

SATA

PCI EXPRESS® | 1/10/40Gb Ethernet | Serial RapidIO®

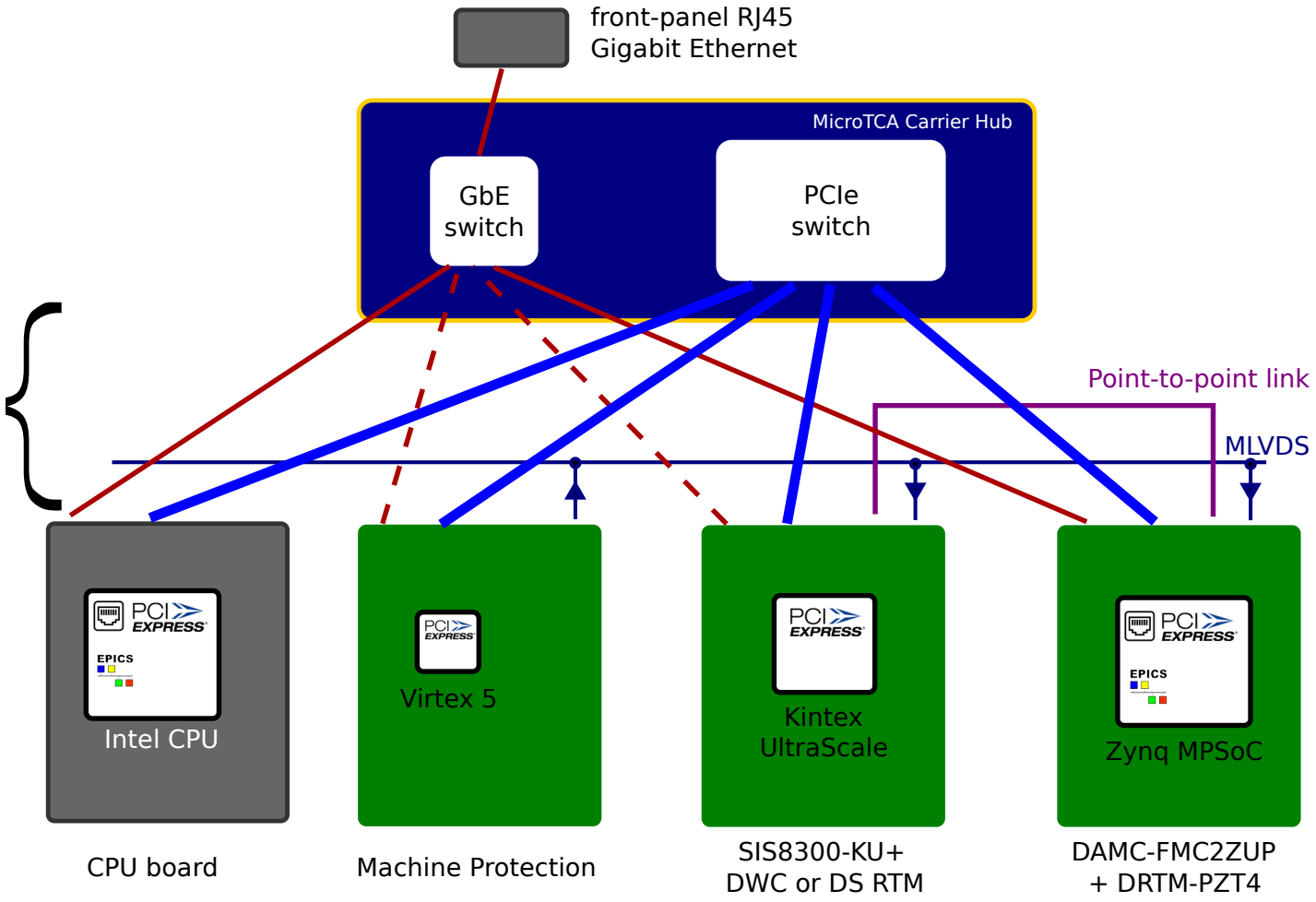
Point-to-Point Links

MLVDS
(Clocks, Triggers, Interlocks etc.)

* Configuration can change from different crates

Data Movement inside the MicroTCA Crate

Eg. Typical LLRF Crate Layout

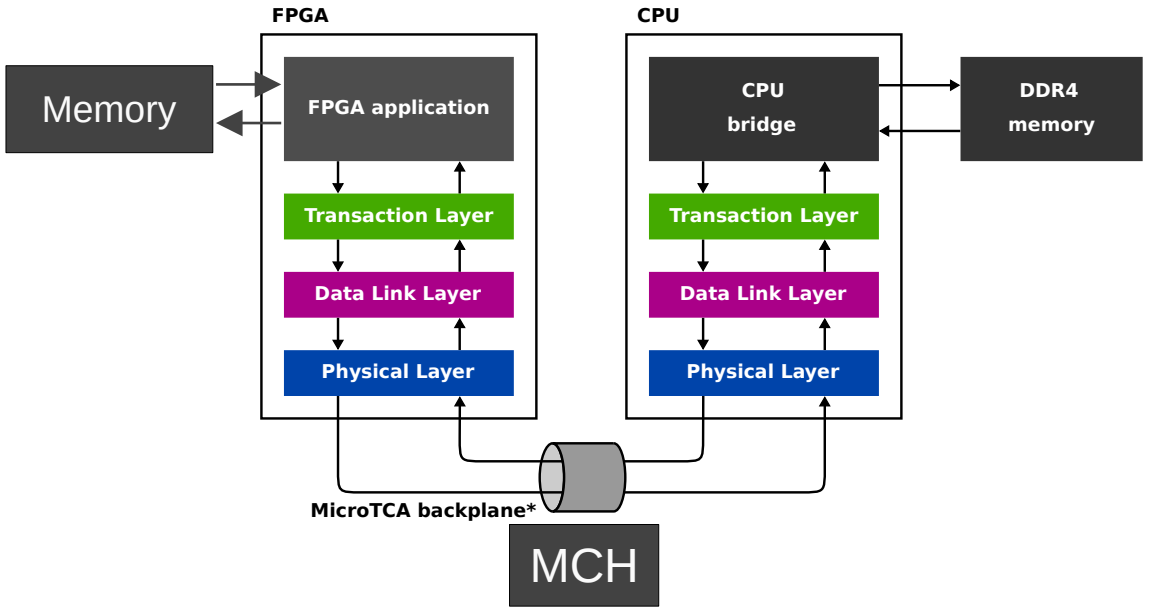


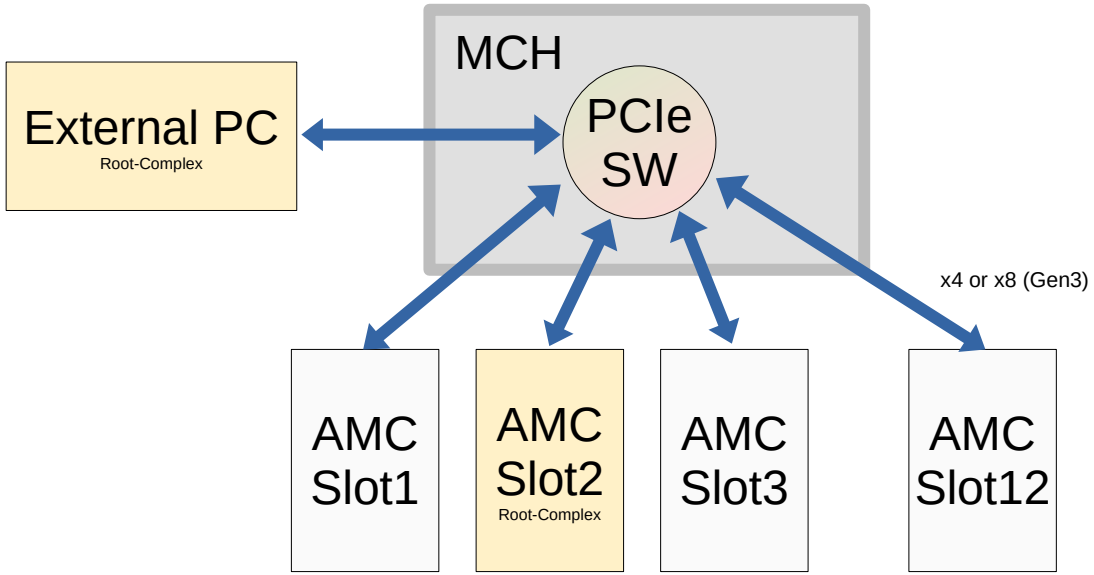
PCIe

PCIe

Introduction

- Serial-Interface
- High-speed data transfer using packets
- Widely adapted by the industry
- Backbone of CXL (Compute Express Link)
- Heavily used in Experimental Physics + MicroTCA



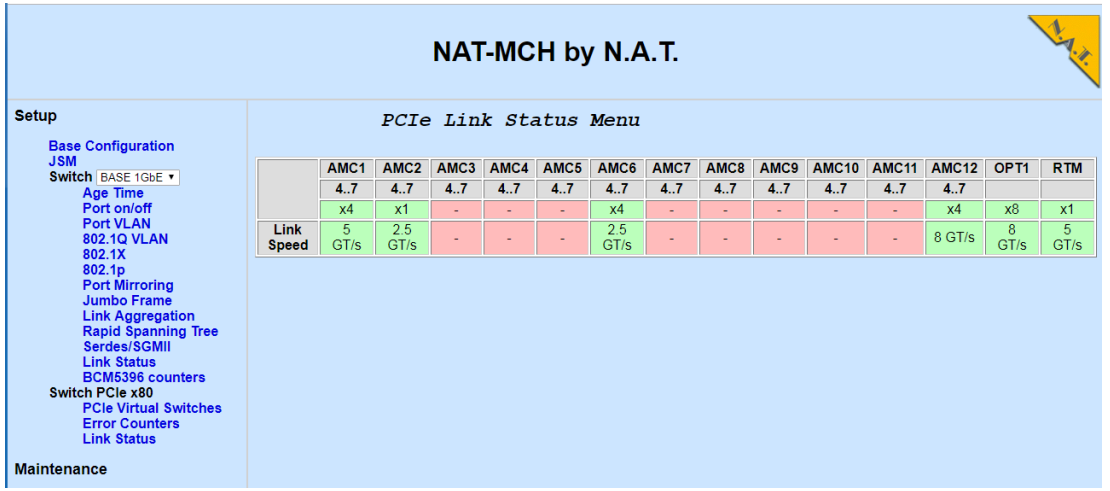


MCH's role on PCIe

Management Side

- PCIe switch for the AMC-Backplane sits on the MCH.
- Use Webserver to configure the switch:
  - Tell the switch where the CPU (Root-Complex) sits
    - External Uplink is possible **(and recommended)**
  - Limit speeds if necessary
  - Create an PCIe 'islands' where multiple Root-Complex co-exists inside a single MicroTCA crate.

See PCIe Link Status:



NAT-MCH by N.A.T.

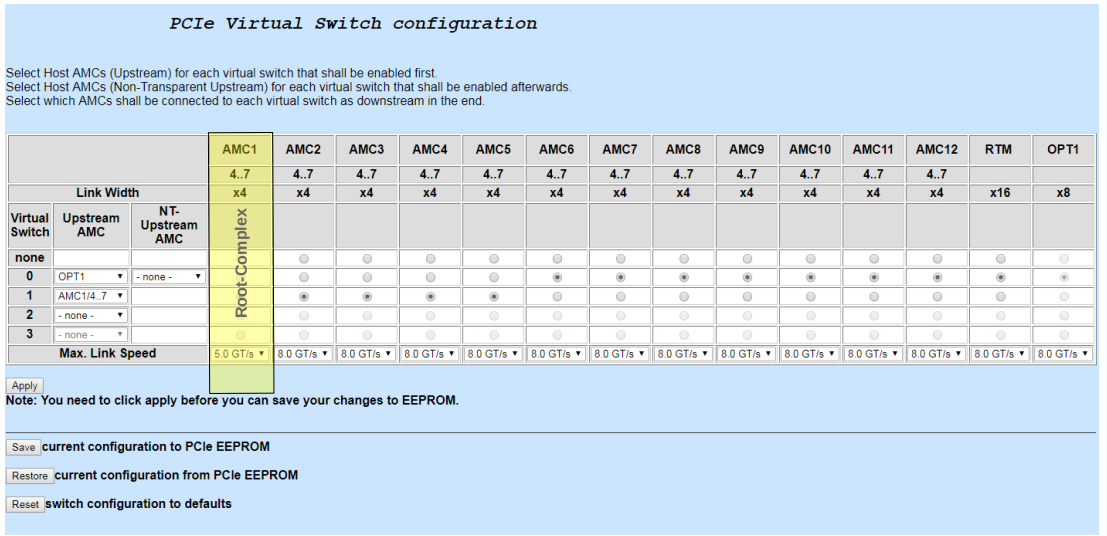
Setup
- Base Configuration
- JSM
- Switch BASE 1GbE
  - Age Time
  - Port on/off
  - Port VLAN
  - 802.1Q VLAN
  - 802.1X
  - 802.1p
  - Port Mirroring
  - Jumbo Frame
  - Link Aggregation
  - Rapid Spanning Tree
  - Serdes/SGMII
  - Link Status
  - BCM5396 counters
- Switch PCIe x80
  - PCIe Virtual Switches
  - Error Counters
  - Link Status

Maintenance

PCIe Link Status Menu

	AMC1	AMC2	AMC3	AMC4	AMC5	AMC6	AMC7	AMC8	AMC9	AMC10	AMC11	AMC12	OPT1	RTM
	4..7	4..7	4..7	4..7	4..7	4..7	4..7	4..7	4..7	4..7	4..7	4..7		
	x4	x1	-	-	-	x4	-	-	-	-	-	x4	x8	x1
Link Speed	5 GT/s	2.5 GT/s	-	-	-	2.5 GT/s	-	-	-	-	-	8 GT/s	8 GT/s	5 GT/s

Configure PCIe Switch



PCIe Virtual Switch configuration

Select Host AMCs (Upstream) for each virtual switch that shall be enabled first.
Select Host AMCs (Non-Transparent Upstream) for each virtual switch that shall be enabled afterwards.
Select which AMCs shall be connected to each virtual switch as downstream in the end.

			AMC1	AMC2	AMC3	AMC4	AMC5	AMC6	AMC7	AMC8	AMC9	AMC10	AMC11	AMC12	RTM	OPT1
			4..7	4..7	4..7	4..7	4..7	4..7	4..7	4..7	4..7	4..7	4..7	4..7		
Link Width			x4	x4	x4	x4	x4	x4	x4	x4	x4	x4	x4	x4	x16	x8
Virtual Switch	Upstream AMC	NT-Upstream AMC	Root-Complex													
none																
0	OPT1	- none -														
1	AMC1/4..7															
2	- none -															
3	- none -															
Max. Link Speed			5.0 GT/s	8.0 GT/s	8.0 GT/s	8.0 GT/s	8.0 GT/s	8.0 GT/s	8.0 GT/s	8.0 GT/s	8.0 GT/s	8.0 GT/s	8.0 GT/s	8.0 GT/s	8.0 GT/s	8.0 GT/s

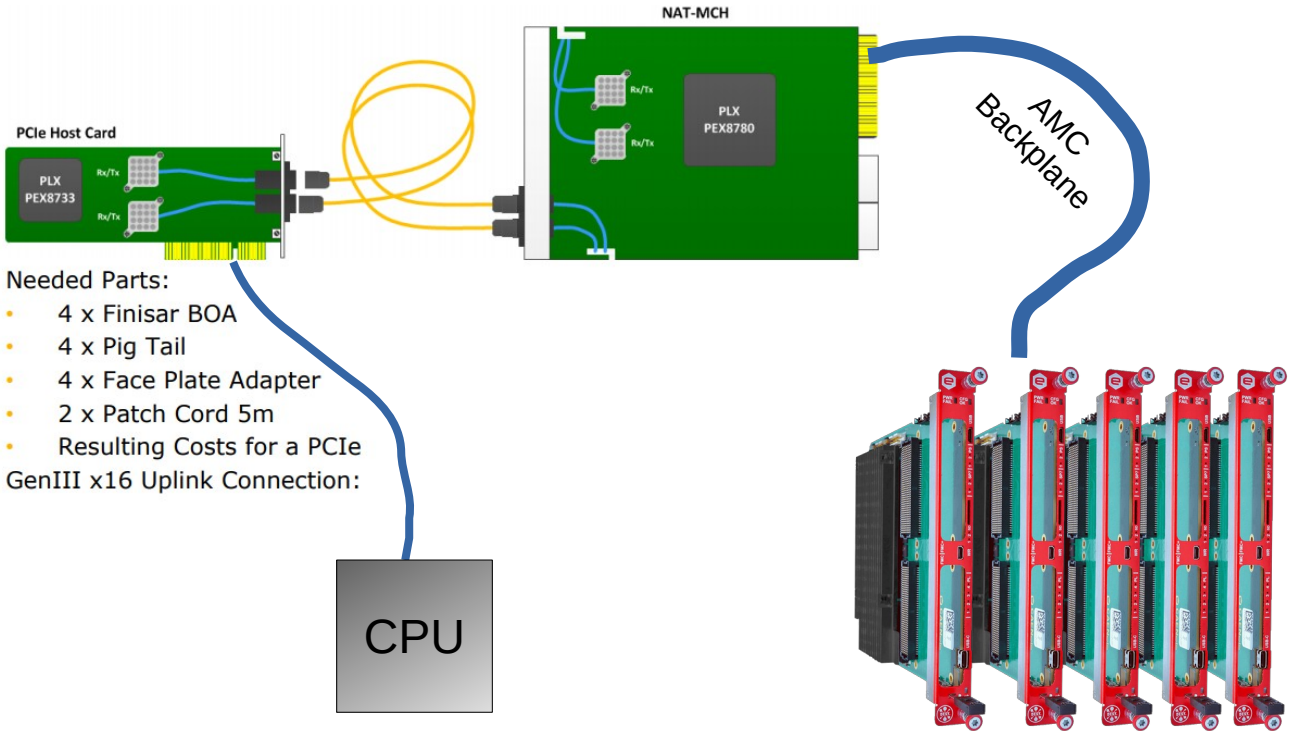
Apply
Note: You need to click apply before you can save your changes to EEPROM.

Save current configuration to PCIe EEPROM
Restore current configuration from PCIe EEPROM
Reset switch configuration to defaults

Optical Uplink for PCIe

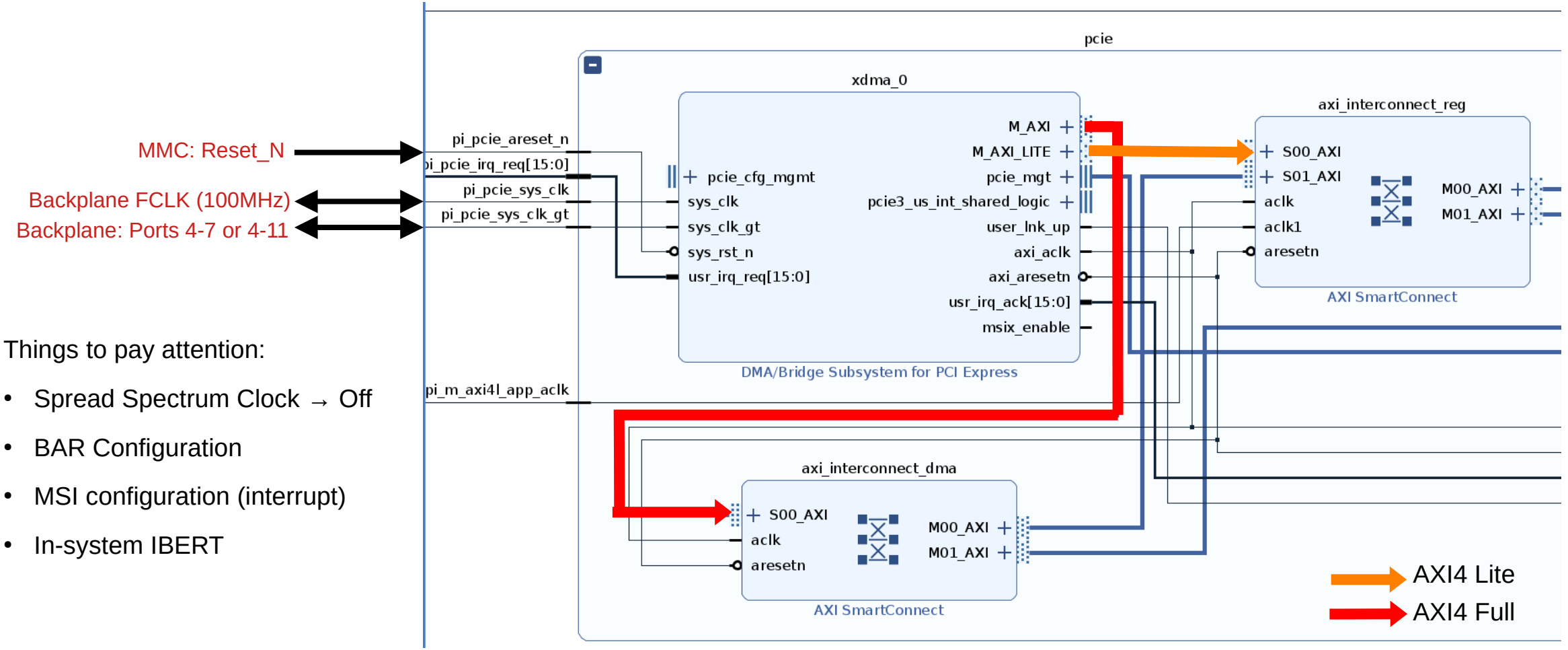
Optical Uplink

- MicroTCA.4 can only offer 80W (AMC + RTM)
  - Pretty weak for 2024 standards. Top TDP currently is at 500W TDP
- For designs that require high CPU load, Root-Complex should connect to PCIe Switch of the MCH through Front Panel



PCIe on PL Side

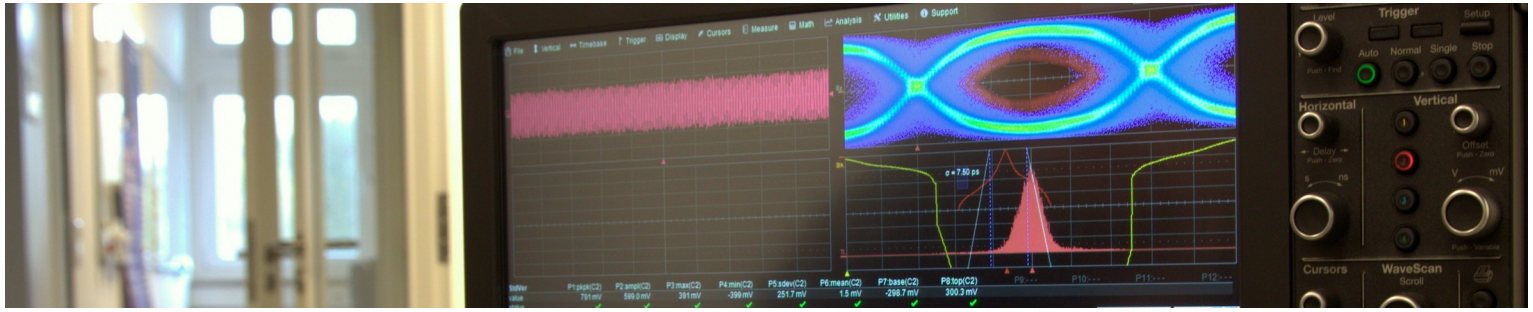
PCIe EP on AMD FPGAs



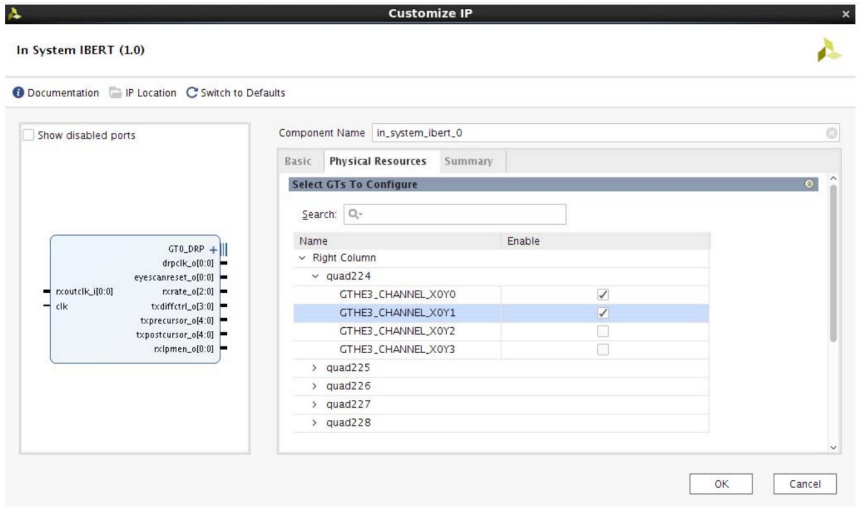
- Things to pay attention:
  - Spread Spectrum Clock → Off
  - BAR Configuration
  - MSI configuration (interrupt)
  - In-system IBERT

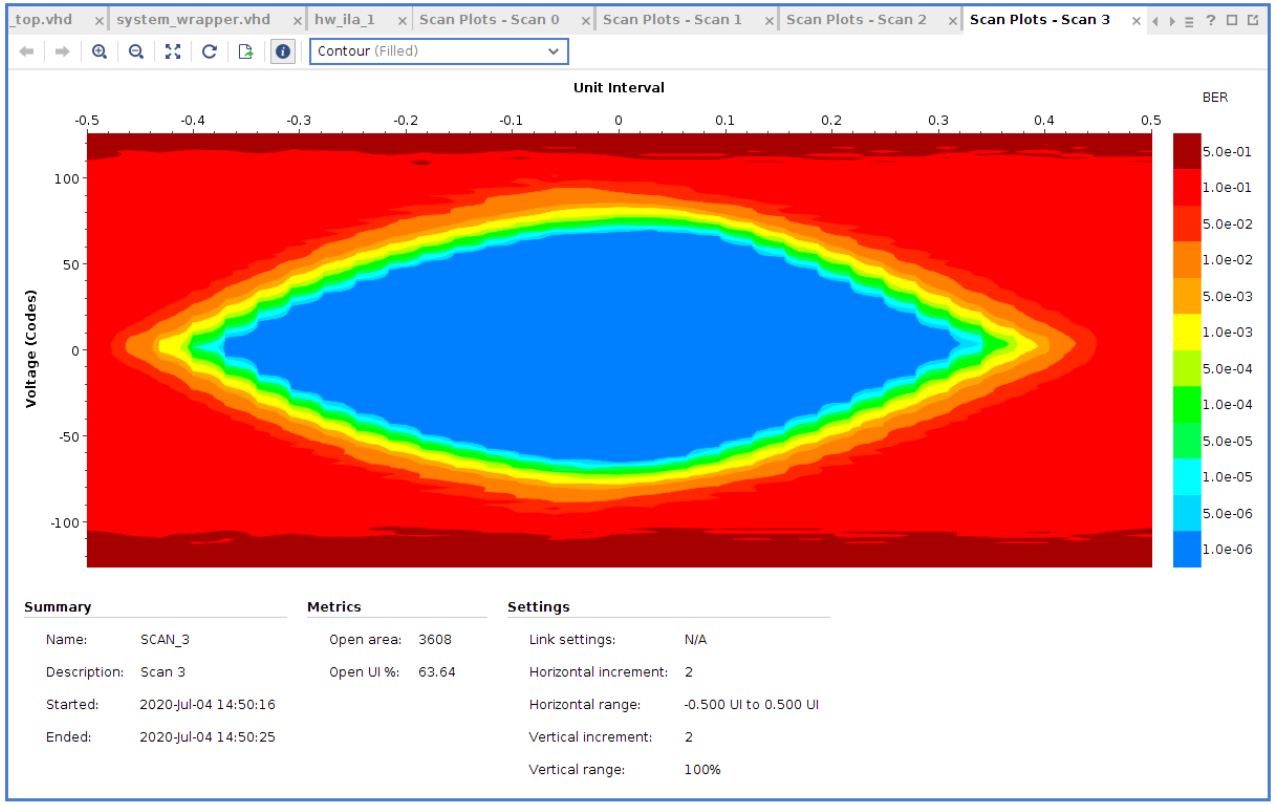
IBERT

See the eye directly from the FPGA



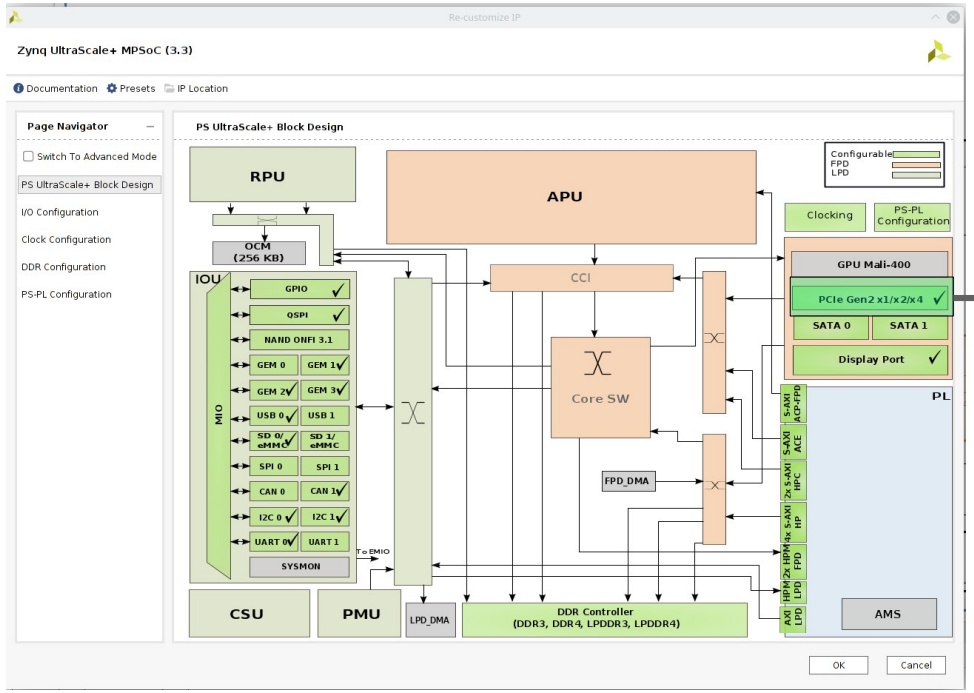
- It’s important to verify if the eye is open on each MGT.
- Not every backplane is equal. Some are worse than the other
- Use In-system IBERT to scan and get the ‘eye’

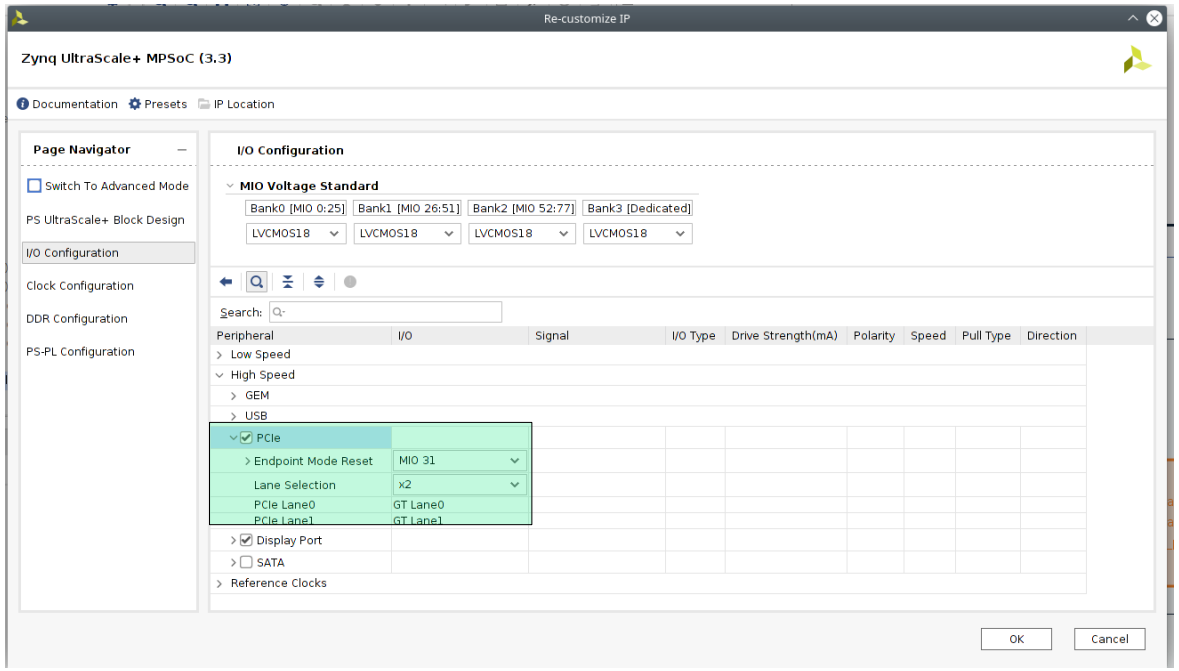




PCIe on PS Side

Connecting PCIe to MPSoC





FPGA can be either Root-Complex or Endpoint in this scenario.

PCIe

CPU side (GNU/Linux)

```
$: lspci -vvv
```

(use 'sudo' to see even more)

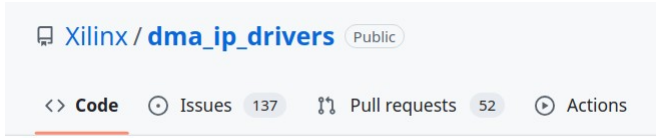
MicroTCA Slot number

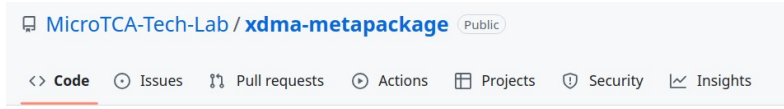
```
06:00.0 Serial controller: Xilinx Corporation Device 8024 (prog-if 01 [16450])
        Subsystem: Xilinx Corporation Device 0007
        Physical Slot: 4
        Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
        Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-
        Latency: 0
        IOMMU group: 2
        Region 0: Memory at c3000000 (32-bit, non-prefetchable) [size=16M]
        Region 1: Memory at c4000000 (32-bit, non-prefetchable) [size=64K]
        Capabilities: <access denied>
        Kernel driver in use: xdma-chr
        Kernel modules: xdma_chr
```

Features of Endpoint

BARs

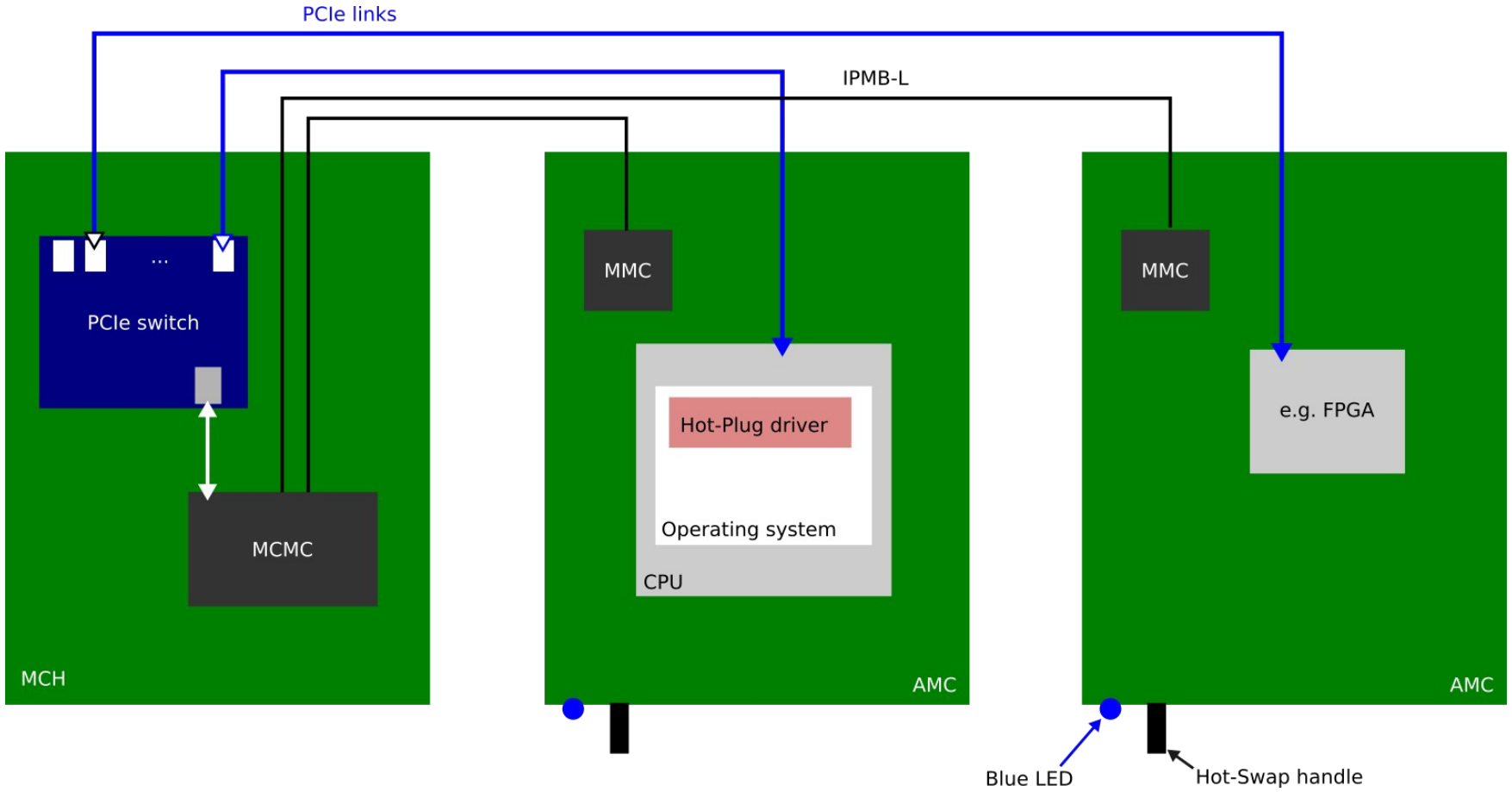
Loaded Driver





PCIe Hot-Plug Mechanism

How Management plays critical role in PCIe interrupts



Gigabit Ethernet

Gigabit Ethernet

Introduction

- 1000BASE-KX *(KX→ Electrical Backplanes)*
- 1.25 Gbaud
- 2 Differential Pairs: TX and RX
- Non-return-to-zero (NRZ) code
- 8b/10b encoding
- Clock embedded into the data stream
- No forward error correction

- AMC.2 E1 → Capable of using only Port-0 of Gbe
- AMC.2 E2 → Capable of using both Port-0 and Port-1 of Gbe (Redundant)

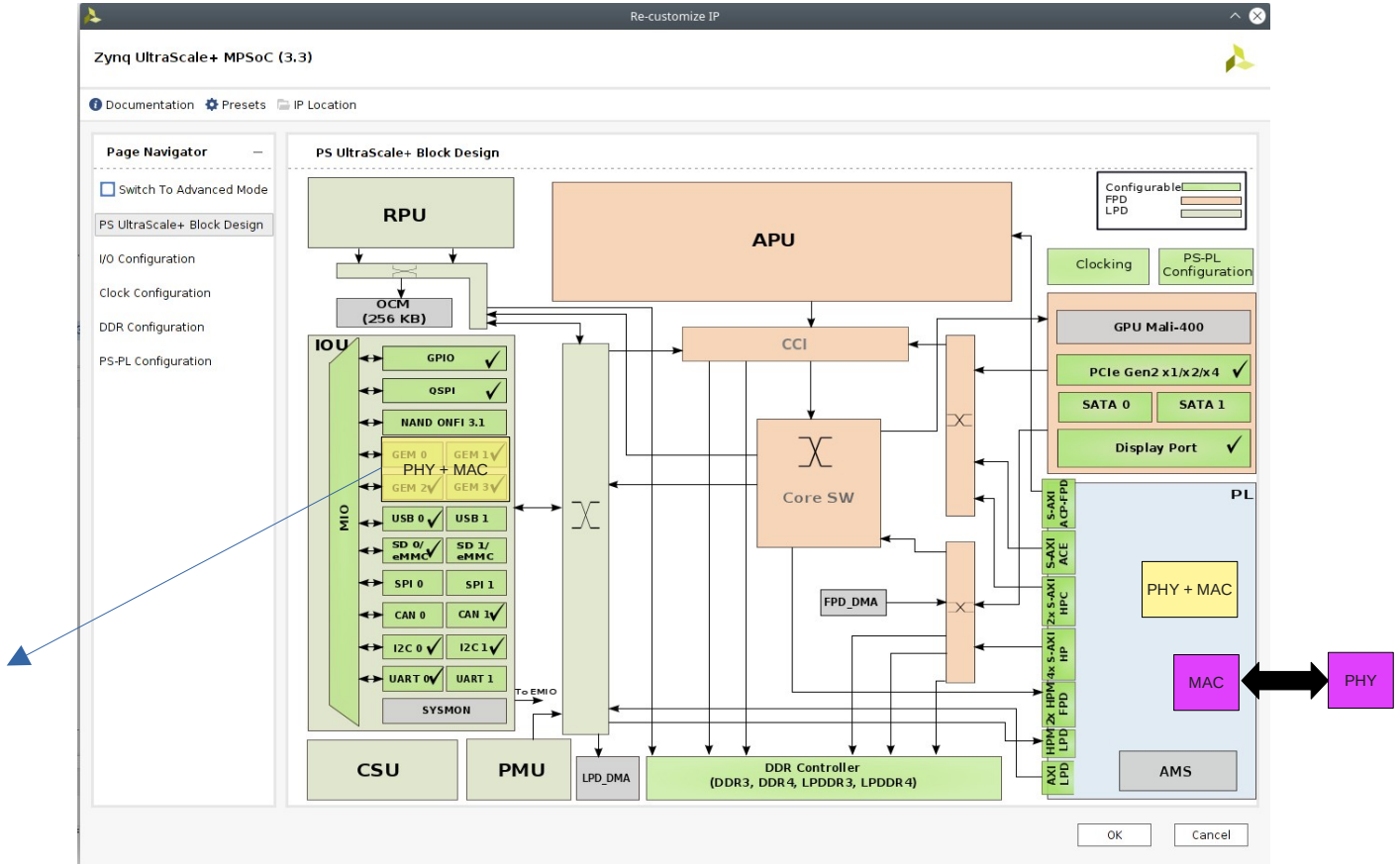
Gigabit Ethernet

Introduction

- AMC Backplane offers two ethernet ports. Port 0 and Port 1.
- Different possibilities for connection to the FPGA:
- Through PL Pins:
  - PHY + MAC Solution
  - MAC inside PL, PHY is outside
- Through PS Pins:
  - Use GEM
  - Add necessary info to the device-tree

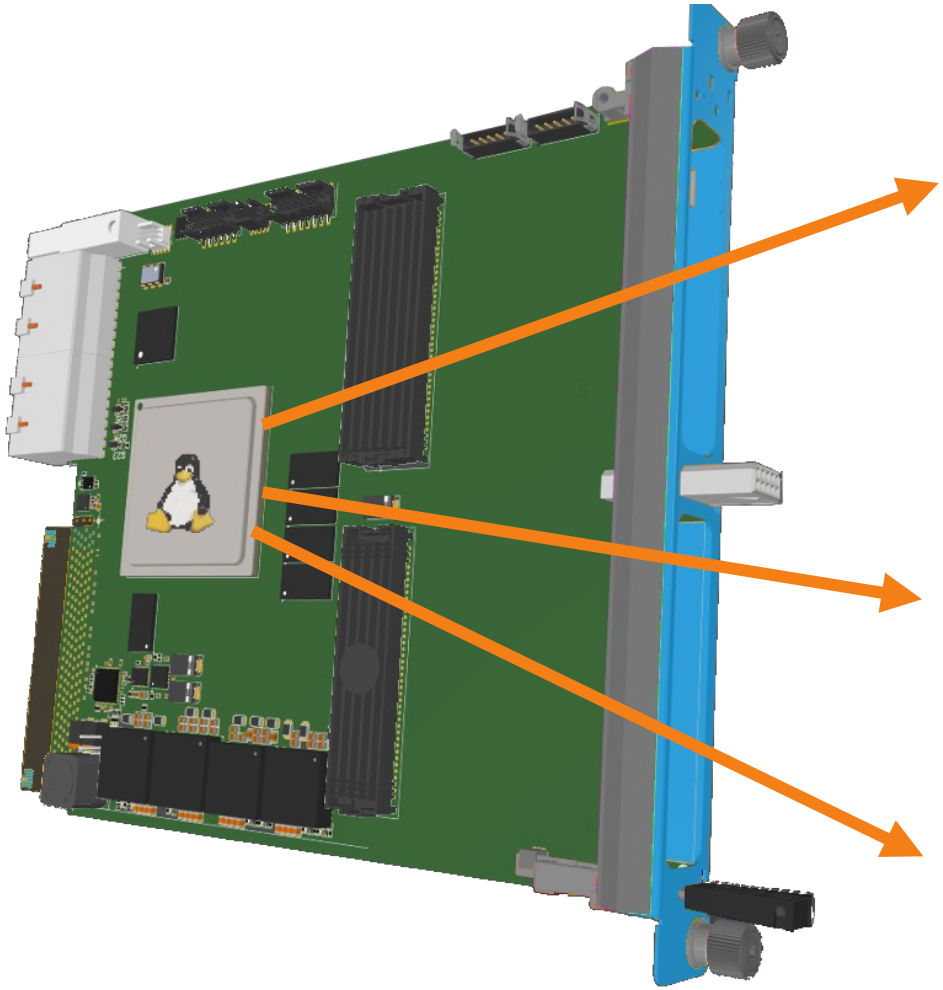
On the Embedded Linux side: <amc>.dtsi

```
&gem1 {
        phy-handle = <&phy0>;
        phy-mode = "rgmii-id";
        /* pinctrl-names = "default";
        pinctrl-0 = <&pinctrl_gem1_default>; */
        phy0: phy@f {
                reg = <0xf>;
                ti,rx-internal-delay = <0x8>;
                ti,tx-internal-delay = <0xa>;
                ti,fifo-depth = <0x1>;
                ti,dp83867-rxctrl-strap-quirk;
        };
};
```

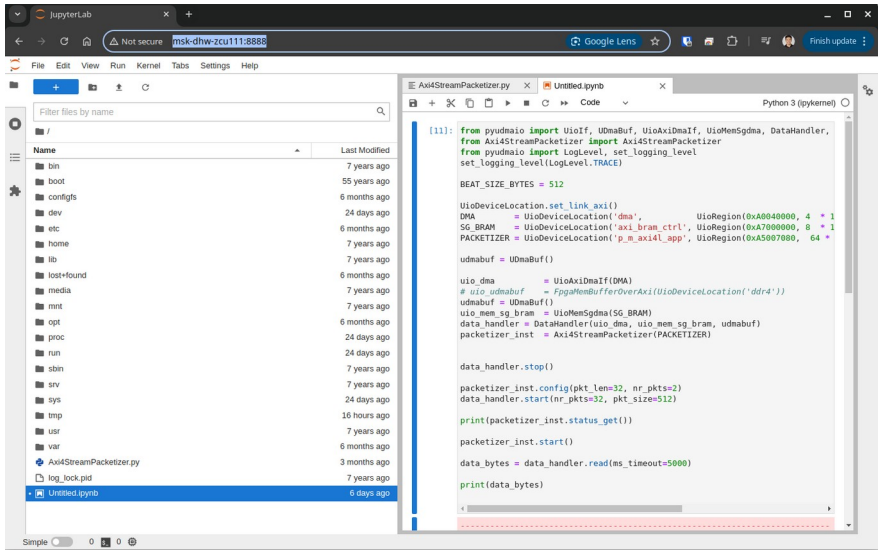


Connect to Embedded Linux via Ethernet

CPU side (GNU/Linux)



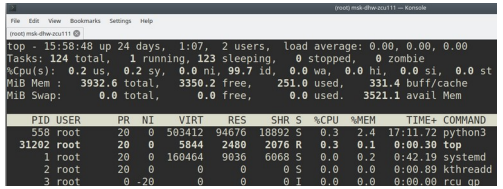
Connect to your Jupyter Notebooks



Run your servers directly

EPICS TANGO

SSH, Telnet

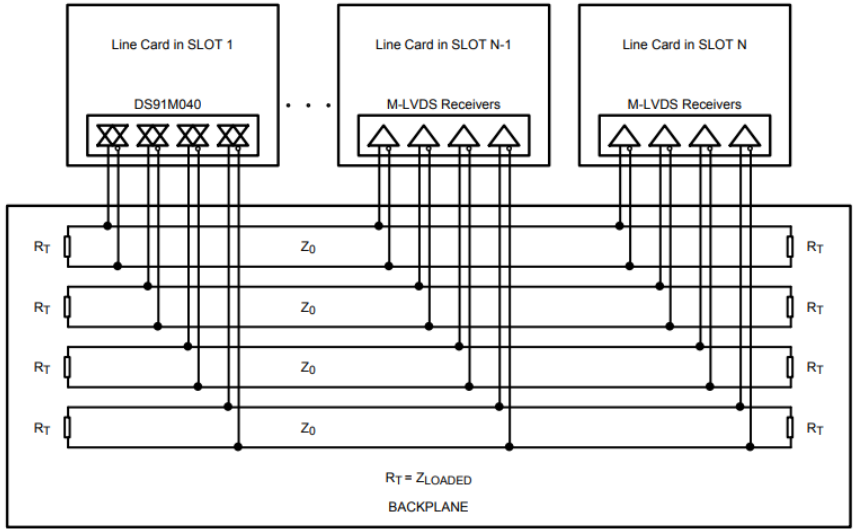


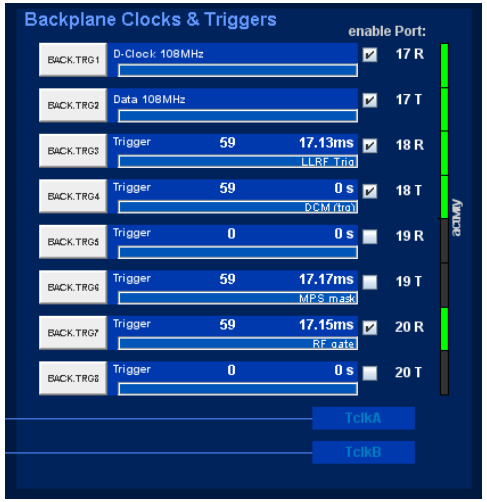
MLVDS

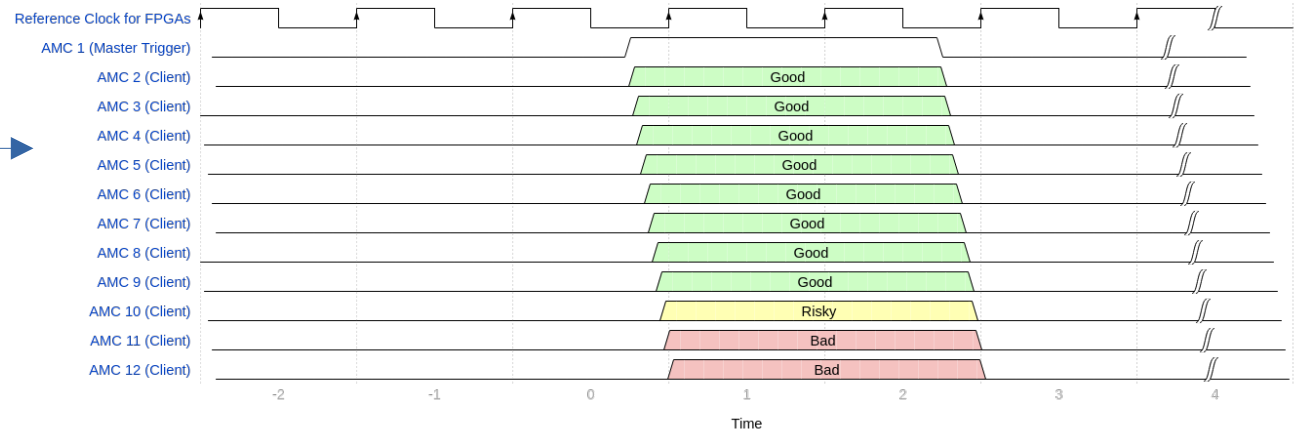
M-LVDS

Clocks, Triggers, Interlocks etc.

- Shared Bus
- Max 200Mbps
- Can be used to distribute:
  - Common clocks
  - Triggers
  - Interlocks
- Pay attention to arrival skew between slots!
  - Use IODELAY elements to match the skew



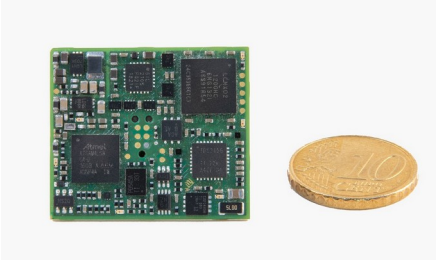


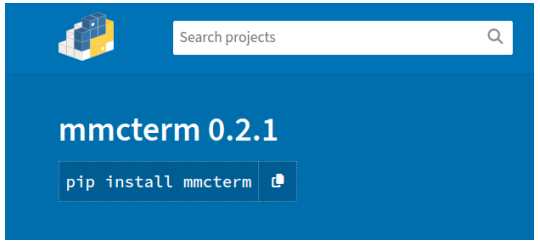


Interraction with Management (MMC)

Use mmcterm

Use all the possible paths!





- You can use IPMI bus to get a Serial Console interface to the MMC of the AMC
- mmcterm is a great tool for this.
  - Available on pip
  - Change Boot Mode
  - Read Sensors
  - Change JTAG Multiplexing
  - Change Boot FLASH pointer
  - **Serial Console of the Embedded Linux**

```
guemues@guemues-desktop:~/fpgafw/projects/petra_motion_control$ mmcterm mskmchref1 0x88
Press Ctrl-x to exit
DAMC-FMC1Z7IO@0x88 MMC>
DAMC-FMC1Z7IO@0x88 MMC>?
? / h / help: Show list of available commands
c ..... Clear screen
r ..... Reset MMC
v ..... Show firmware version
xm [0..n] ..... Start XMODEM update
sb ..... Start bootloader
vb [0..6] ..... Get/set verbosity
tm [smart|dumb|auto] ..... Get/set terminal mode
eefd ..... Set EEPROM factory defaults
s ..... Get status
lc [0..3] [on|off|blink] [on_ms] [off_ms] ..... Set LED
ser [addr] [lun] ..... Get/set event receiver
pu ..... Payload power up
pd ..... Payload power down
ppf [stop|retry|ignore] ..... Get/set payload power fail policy
pc ..... Toggle CPLD programming / JTAG forwarding mode
sj [con|bp|raw] [fpga(1|2|12)|rtm|fmc(1|2)] ..... Get/set JTAG multiplexing
st [0..15] ..... Get/set RTM temp sensor mask
rte [enable|override] ..... Get/set RTM e-keying policy
rto [calibrate] ..... Get/set I_RTM PP 12V calibration
cfu ..... CPLD force update
fru [0..n] ..... Dump FRU information
eth ..... Get backplane Ethernet information
rtp [auto|high|low] ..... Get/set RTM Power Good polarity
i2cd [ipmb|sens|rtm|pmbus|fmc|clk] ..... Detect I2C peripherals
i2cget [ipmb|sens|rtm|pmbus|fmc|clk] [addr] [reg] [nbytes] ..... Get I2C peripheral register
i2cset [ipmb|sens|rtm|pmbus|fmc|clk] [addr] [reg] [data..] ..... Set I2C peripheral register
fd [index] ..... Flash detect
bz [jtag|qspi|sd|raw] ..... Get/set Zynq-7000 boot mode
qz [qspi1|qspi2] ..... Get/set Zynq-7000 QSPI chip selection
rz ..... Re-configure Zynq-7000 (PS_POR)
cps [rpll|mpll] [tclka|tclkb] ..... Get/set TCLKA/B multiplexing
ci [out(0|1|2)|all] [rx19|tx19|rx20|fpga|override|disabled|bypass] Get/set interlock channel(s)
fma [8|16|auto] ..... Get/set FMC EEPROM address width
fmg [enable|disable] ..... Enable/disable fallback for swapped GA lines
fmv [0.95..1.9|auto] ..... Get/set FMC VCC_Vadj
DAMC-FMC1Z7IO@0x88 MMC>
```

Remote Programming FPGA on MicroTCA

Use all the possible paths!

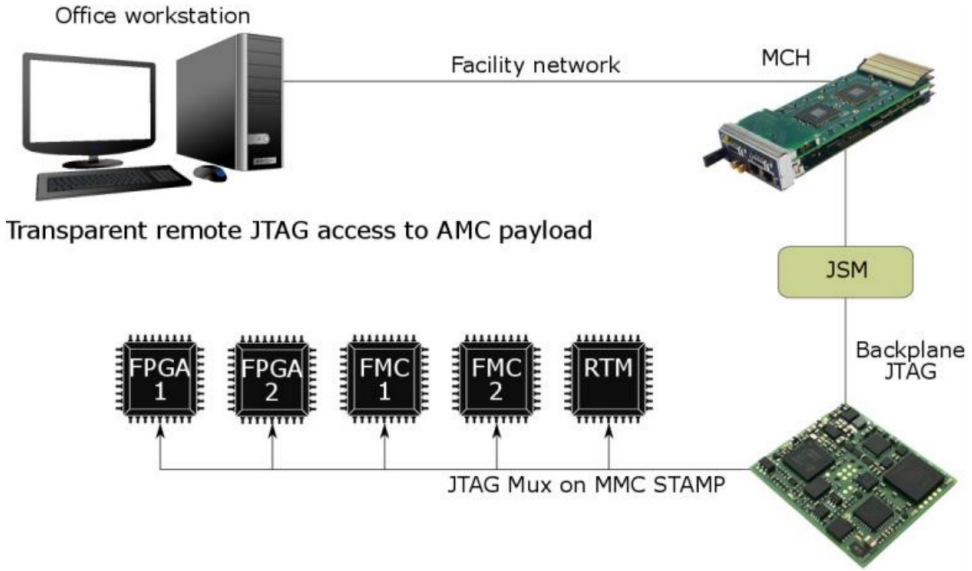
- JTAG Header
- Through Management
- Through Processing System (fpgautils)
- Through a Fat-Pipe using ICAP

Programming FPGA on using Management

Use all the possible paths!

- Modern AMC-Backplanes have JTAG for each AMC slot
- MMC is in charge of JTAG multiplexing inside the AMC

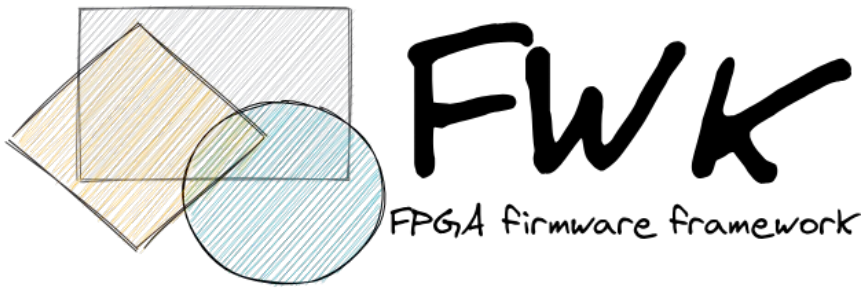
JSM



Use our Open-Source Example Projects!

Do not re-invent the wheel!

FPGA Team at MSK Group of DESY developed a framework for setting up build environment for MicroTCA Hardware:



FWK - Open Source FPGA Framework from DESY

Dec 11, 2024, 2:45 PM

15m

Session 5

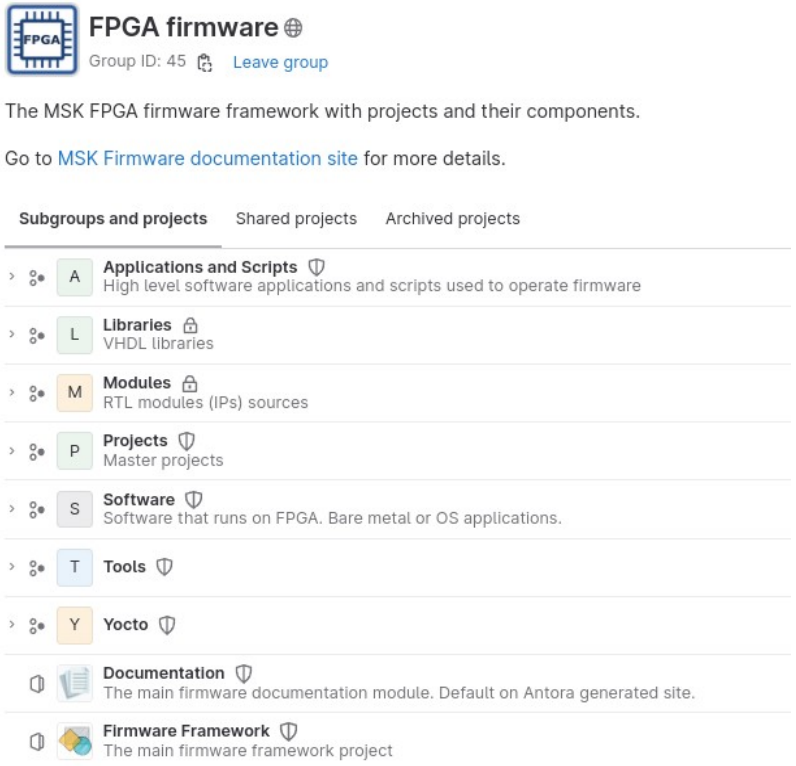
Speaker

Cagil Guemues (MSK (Strahlkontrollen))

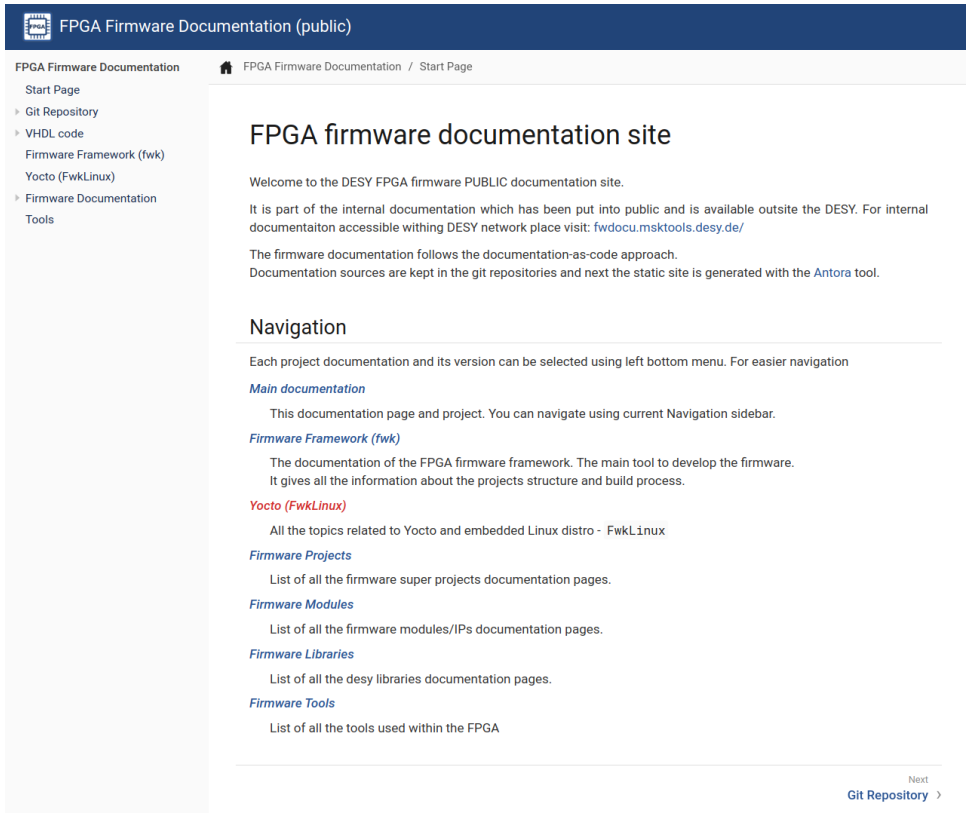
Description

In recent years, DESY has diligently developed an open-source FPGA framework, known as FWK, aimed at expediting FPGA development within the scientific community, particularly with MicroTCA hardware. The framework serves as an abstraction layer that simplifies the utilization of various FPGA vendor tools, facilitates IP integration, aids in documentation creation, and offers many other benefits. In addition to FWK, DESY also provides a wealth of Board Support Packages designed for a range of MicroTCA AMC boards. This presentation will provide an overview of the FWK framework and showcase examples of open-source board support packages, demonstrating their practical utility in accelerating FPGA development for MicroTCA hardware.

Visit our repository today!



gitlab.desy.de/fpgafw



fpgafw.pages.desy.de/docs-pub

Thank you!