System-on-Chip (SoC)

How to bring intelligence to an electronics board!

Content:

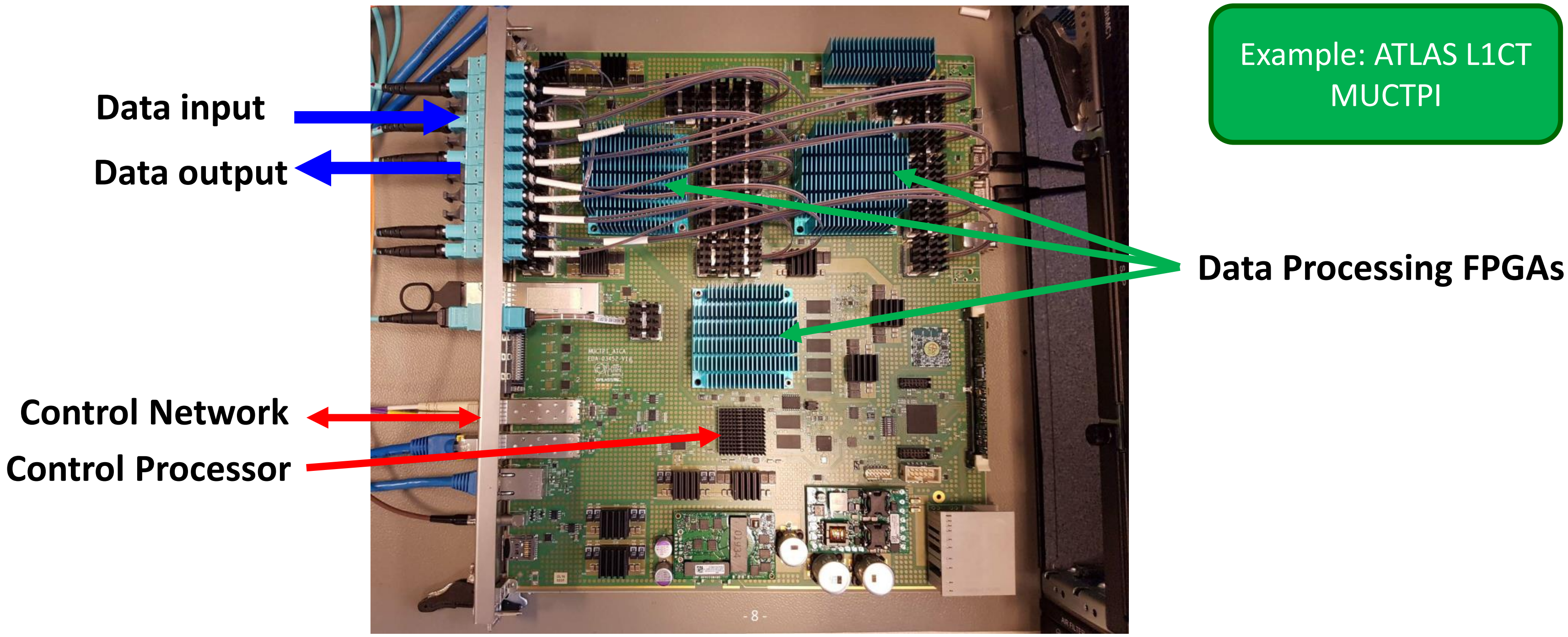
With a lot of material from the SoC Workshop …

Big thanks to all presenters!

Part 1: Introduction

System-on-Chip – Why? What? How?

Control of Electronics Modules (1)



Many electronics modules for trigger and readout in particle physics experiments have a similar structure:

- **Many high-speed links for data input and output, usually o(100) links of o(10) GBits/s**
- **Several high-end FPGAs for processing, usually a few (o(1)):**
  ⇒ workload FPGAs: implement massively parallel, low-latency algorithms for trigger and readout
- **Something for control …**

Control of Electronics Modules (2)

What is control:

- **Control (in a strict sense):**
  Sending of commands, e.g. start, stop, pause, reset, etc.
  Receiving of interrupts
- **Configuration:**
  Load configuration data, e.g. settings, look-up table contents, bit-stream files etc.
- **Monitoring:**
  Read monitoring data, e.g. temperatures, voltages, optical power, counter values, event monitoring etc.

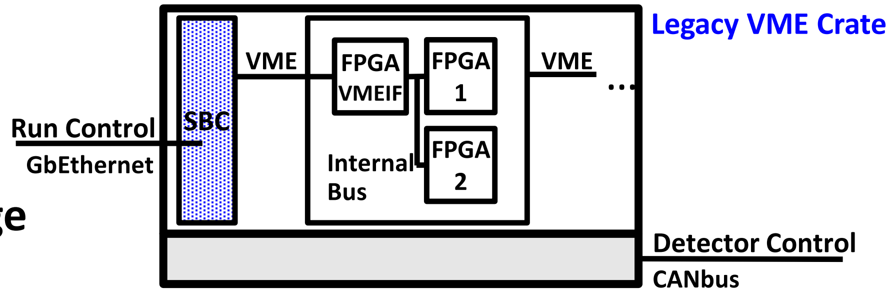
Generally two types of control:

- **Hardware control** *(or slow control)***:**
  Clock, power, and optical modules, and configuration of FPGAs, etc.
  Often using industry standards like I2C, SPI, JTAG etc.
- **Run control** *(or operational control)***:**
  Workload FPGAs: read/write status/control registers and memories/LUTs, read counter registers, data of selected physics events, etc.

Control of Electronics Modules (3)

In the past, very popular - VME:

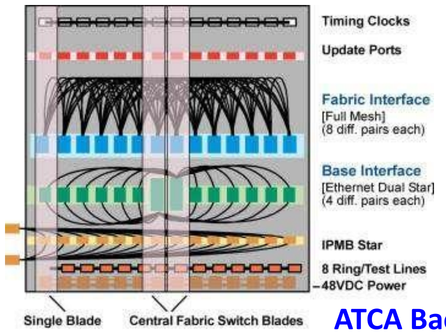
- Hardware control: CANbus
- Run control:
  Single-Board Computer (SBC) or VME bridge



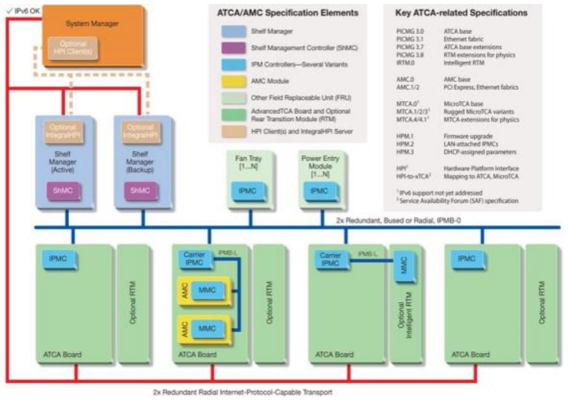
Today, many new projects use ATCA:

- Control is oriented towards GbEthernet:
- Hardware control:
  Blade → IPMC* → Shelf Manager → SCADA**
- Run control:
  Via base interface to hub module,
  or directly from ATCA blade to control network…

⇒ **Need a new control strategy!**



ATCA Backplane



✓ IPv6 OK
System Manager
Optional HPI Client(s)
ATCA/AMC Specification Elements
Shelf Manager
Shelf Management Controller (ShMC)
IPM Controllers—Several Variants
AMC Module
Other Field Replaceable Unit (FRU)
AdvancedTCA Board and Optional Rear Transition Module (RTM)
HPI Client(s) and IntegralHPI Server
Key ATCA-related Specifications
PICMG 3.0 ATCA base
PICMG 3.1 Ethernet fabric
PICMG 3.7 ATCA base extensions
PICMG 3.8 RTM extensions for physics
IRTM.0 Intelligent RTM
AMC.0 AMC base
AMC.1/2 PCI Express, Ethernet fabrics
MTCA.0¹ MicroTCA base
MTCA.1/2/3¹ Rugged MicroTCA variants
MTCA.4/4.1¹ MTCA extensions for physics
HPM.1 Firmware upgrade
HPM.2 LAN-attached IPMCs
HPM.3 DHCP-assigned parameters
HPI² Hardware Platform Interface
HPI-to-xTCA² Mapping to ATCA, MicroTCA
¹ IPv6 support not yet addressed
² Service Availability Forum (SAF) specification
Optional IntegralHPI
Shelf Manager (Active)
Shelf Manager (Backup)
ShMC
Fan Tray [1...N]
Power Entry Module [1...N]
IPMC
2x Redundant, Bused or Radial, IPMB-0
Carrier IPMC
IPMB-L
AMC
MMC
Optional RTM
Optional Intelligent RTM
ATCA Board
2x Redundant Radial Internet-Protocol-Capable Transport

ATCA Hardware Platform Management

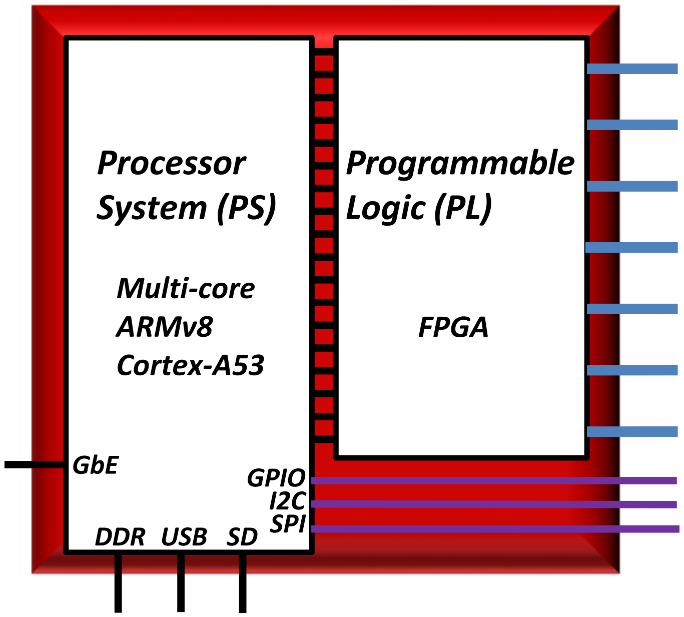
*   IPMC   = Intelligent Platform Management Controller
** SCADA = Supervisory Control And Data Acquisition

System-on-Chip

Programmable logic + processor system, *"FPGA and CPU"*

This definition of "SoC" is more restrictive than in Wikipedia

- **Programmable logic (PL)** = like FPGA:
  – Has logic cells, memory blocks, and I/O links (e.g. MGTs)
  – Can implement data processing or interface to other workload FPGAs, e.g. using Xilinx AXI Chip2Chip protocol
- **Processor system (PS)** = like CPU:
  - Currently all are multi-core ARM processors
  - Has peripherals, e.g. GbEthernet, I2C, SPI, GPIO, etc.
  - Runs software: "bare-metal" application or operating system, e.g. Linux



→ Very popular SoC: Xilinx Zynq

Zynq UltraScale+ MPSoC ["ZynqMP"]:	**ARMv8 Cortex-A53 (64-bit, 2-4 cores)**
Zynq 7000 SoC ["Zynq"]:	**ARMv7 Cortex-A9 (32-bit, 1-2 cores)**

… both with different extents of PL functionality

The Intel Stratix 10 SoC or Arria 10 SoC could also be alternatives …

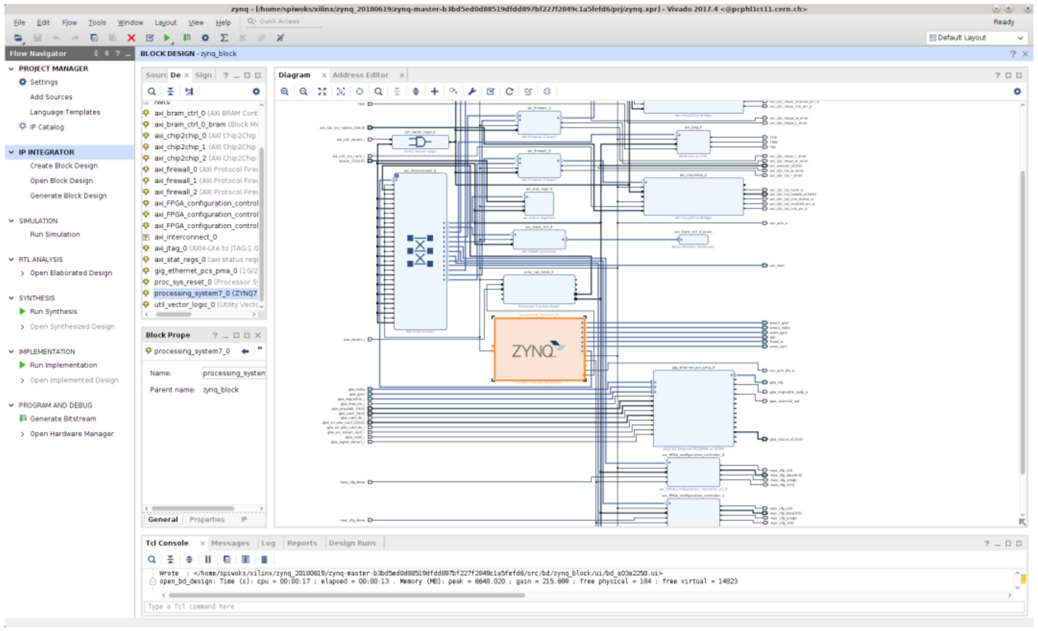
System-on-Chip: Firmware

For the SoC/PL: need firmware

This part if very well known to FPGA designers

Use FPGA design tool, e.g. Xilinx Vivado,
Design your logic using FPGA resources:

- **Multiple Gigabit Transceivers (MGT) and LVDS links**
- **Use them for AXI chip 2 chip to other FPGAs**
- **Ethernet, e.g. 10Gb/s**
- **Memory blocks**
- **Logic blocks for trigger/readout algorithms**
- **Etc.**



```
Wrote  : </home/spiwoks/xilinx/zynq_20180619/zynq-master-b3bd5ed0d88519dfdd897bf227f2849c1a5fefd6/src/bd/zynq_block/ui/bd_a03a2250.ui>
open_bd_design: Time (s): cpu = 00:00:17 ; elapsed = 00:00:13 . Memory (MB): peak = 6648.020 ; gain = 215.609 ; free physical = 184 ; free virtual = 14023
```

⇒ **Produce bit stream file (for FPGA configuration)**
+ hardware description file (.xsa file) for software development

System-on-Chip: Software (1)

This part if very well known to embedded system designers

For the SoC/PS: need software

- Architecture: current SoC use ARM-based processor cores
  - Tools required: toolchain = compiler + system libraries, several options:
    - Cross-compilation, on x86_64 for ARM
    - Native compilation, e.g. on an ARM Server or on the SoC itself
    - Using emulation, e.g. qemu
- Boot sequence: use a chain of boot loader(s)
  - First-Stage Boot Loader (FSBL): initialise SoC/PS and peripherals
  - Secondary program loader (SPL): for loading operating system or bare-metal application
  - Some other files for power management and ARM secure boot
  - Xilinx Software Development Kit (SDK) or PetaLinux provide all necessary files
- Operating system:
  - Without operating system:
    - Run bare-metal application
    - Use Xilinx board support package and develop user application program with it
  - With operating system:
    - System-level software: boot loader (U-Boot) + operating system (kernel + rootfs)
    - User-level software: application software, specific to module, and implementation in workload FPGAs

System-on-Chip: Software (2)

Operating Systems was largely discussed during 1st SoC Workshop

For the SoC/PS: operating system

- **Linux (very poplular):**
  - **PetaLinux: Xilinx provided distribution, convenient to start with**
  - **Yocto/OpenEmbedded: free tool to build your own Linux distribution, with Xilinx meta layer**
    Note: PetaLinux is based on Yocto
  - **Others …**
- **Real-time operating system if reaction time matters, e.g. FreeRTOS etc.**

Aspects for choice of operating system:

- **Network security:**
  In the experiment technical control networks at CERN, only CERN-IT certified systems are allowed. All other systems have to be put behind an isolating host/switch.
- **System integration:**
  Experiment-wide support for system administration requires common operating system.

⇒ Use CentOS:

- **Widely accepted**
- **Available for aarch64**
- **Some support from CERN-IT available**

⇒ Use cross installation (dnf), recipe available here
⇒ Many systems use it successfully

Situation before CentOS announced change in long-term support in DEC-2020

Part 2: SoC Workshop

System-on-Chip Interest Group

SoC Interest Group

- Interest group *"System-on-Chip for Electronics":*
  - Founded at ACES Workshop in 2018
  - Open for discussions on all aspects of SoC
  - Currently 152 members
    - → Mailing list*: system-on-chip@cern.ch*
    - → mattermost channel: ad-hoc discussions
    - → *twiki page*: documentation, tutorials, presentations
    - → gitlab group: software
- Organising committee:
  - M. Dobson (CMS), R. Kopeliansky (ATLAS), F. Meijers (CMS), D. A. Scannicchio (ATLAS), M. Shukla (BE-CEM), R. Spiwoks (EP-ESE & ATLAS)
  - Follow up on common issues and organize meetings
- *Meetings:*
  - 1st SoC Workshop June 12-14, 2019: 3 days, 160 people registered (link)
  - A number of SoC Interest Group Meetings: (link)

2nd CERN SoC Workshop

2nd CERN SoC Workshop took place last week, June 7-11:

- Follow up from the 1st workshop which took place in June 2019.
- Main goal was to report on progress since the 1st workshop.
- Due to COVID the workshop was completely virtual (zoom&mattermost)
- It took place over five afternoons:

MON, June 7	TUE, June 8	WED, June 9	THU, June 10	FRI, June 11
Vendor Presentations	Overviews Project Reports I	Project Reports II	Tutorials	Common Issues

- 125 people registered, and 40-70 attended at each session, with peaks of 90 on the first day.
- Indico page: all presentations, recordings, minutes (to come)

Part 2: SoC Workshop
Highlights

This is a selection of points – please find the full material on the indico page

Part 2: SoC Workshop

Vendor Presentations

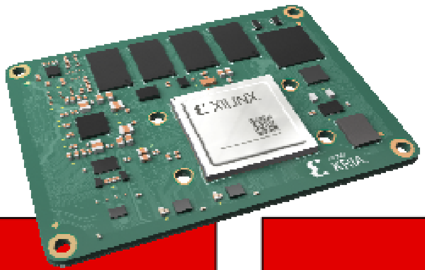
Note:
During 1st SoC Workshop some concern was raised about long-term maintainability of SoCs.
In order to overcome this problem, many projects are moving from SoC (directly on PCB) to System-on-Modules (SoM), i.e. using a mezzanine card with a SoC on it.

Xilinx System-on-Module = Kria Module

K26 SOM Overview

Based on the Zynq® UltraScale+™ MPSoC Architecture

Powerful SoC at interesting price



COMPUTE	
Application Processor	64-bit Quad-Core Arm® Cortex®-A53
Real-Time Processor	32-bit Dual-Core Arm Cortex-R5F
Graphics Processor	Arm Mali™-400MP2
Programmable Logic	256K System Logic Cells
Deep Learning Processor	4K INT8 (upgradable to INT4)
Video Codec (H.264/H.265)	Up to 32 Streams (total resolution ≤ 4Kp60)
Memory	26.6Mb On-Chip SRAM
Security	IEC62443 Security w/HW Root-of-Trust

INTERFACES	
Camera	11 x4 Full MIPI or sub-LVDS Interfaces 1 x4 SLVS-EC Interfaces
USB	4x USB 2.0 / 3.0
Multi-Media	DisplayPort, HDMI
Network	1Gb up to 40Gb Ethernet (w/GigE Vision)
Memory Interface	4GB 64-bit DDR4
Transceivers	4x 12.5Gb/s, 4x 6Gb/s
Mechanical	77 x 60 x 11mm w/ dual 240-pin connectors

1.4 TOPs

G. Donzel, Avnet/Silica: Xilinx

Xilinx Next Generation: Adaptive Compute Accelerator Platform (ACAP) – Versal families, 7nm

Versal Architecture Overview

Re-architectured hardware logic for 4x compute density

More compute power

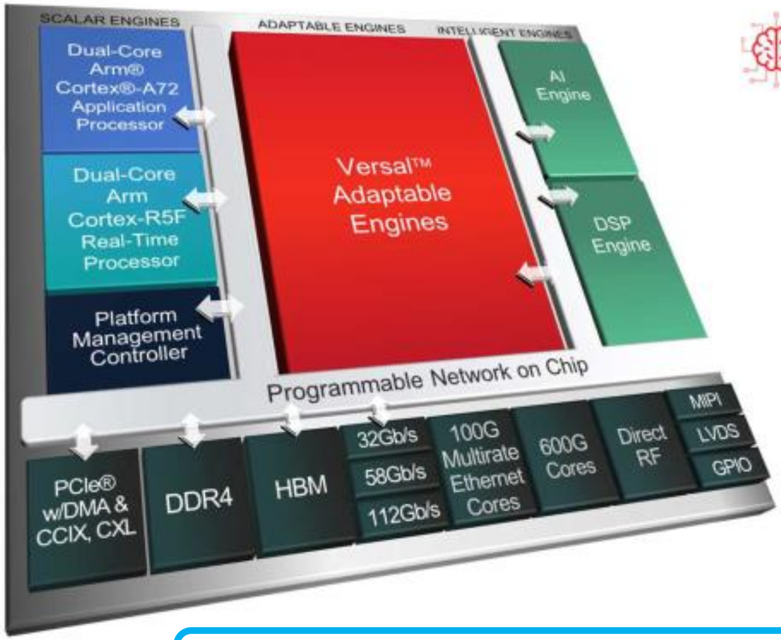
Adaptable Engines
- 4X compute density[1]
- Faster timing closure

Scalar Engines
- Platform Control
- Embedded Edge Compute

PCIe Gen5 & CCIX, CXL
- 2X PCIe & DMA bandwidth
- Cache-coherency

DDR4 Memory
- 3200-DDR4, 4266-LPDDR4
- 2X bandwidth/pin

Transceiver Leadership
- Broad range, 1G →112G
- 58G in mainstream devices



SCALAR ENGINES
ADAPTABLE ENGINES
INTELLIGENT ENGINES
Dual-Core Arm® Cortex®-A72 Application Processor
Dual-Core Arm Cortex-R5F Real-Time Processor
Platform Management Controller
Versal™ Adaptable Engines
AI Engine
DSP Engine
Programmable Network on Chip
PCIe® w/DMA & CCIX, CXL
DDR4
HBM
32Gb/s
58Gb/s
112Gb/s
100G Multirate Ethernet Cores
600G Cores
Direct RF
MIPI
LVDS
GPIO

Intelligent Engines (DSP)
- AI Compute
- Diverse DSP Workloads

Programmable NoC
- Guaranteed Bandwidth
- Enables SW Programmability

Protocol Engines
- 400G/600G cores
- Power-Optimized

Programmable I/O
- Any interface or sensor
- Includes 3.2Gb/s MIPI

Intelligent engines for diverse computing: AI and DSP

1: 4X logic resources per configurable logic block

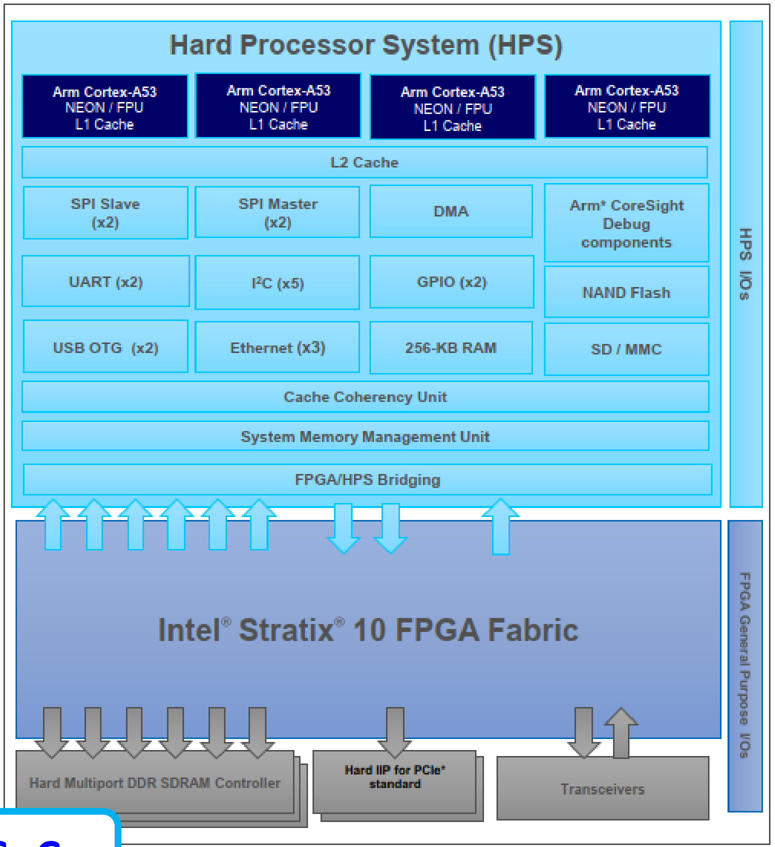
XILINX

G. Donzel, Avnet/Silica: Xilinx

Intel (Altera): SOC Families - Stratix 10 and Agilex, 10nm

Intel® Stratix® 10 and Agilex™ SoC FPGAs

- Processor
  - Quad-core Arm* Cortex*-A53 MPCore*
  - 64-bit ARMv8 architecture
- FPGA Features
  - 8 input Adaptive Logic Modules (ALM)
  - Variable precision DSP blocks
  - Hard IP for PCI Express*
  - High-Bandwidth on-chip interfaces


Hard Processor System (HPS)
Arm Cortex-A53 NEON / FPU L1 Cache
Arm Cortex-A53 NEON / FPU L1 Cache
Arm Cortex-A53 NEON / FPU L1 Cache
Arm Cortex-A53 NEON / FPU L1 Cache
L2 Cache
SPI Slave (x2)
SPI Master (x2)
DMA
Arm* CoreSight Debug components
UART (x2)
I²C (x5)
GPIO (x2)
NAND Flash
USB OTG (x2)
Ethernet (x3)
256-KB RAM
SD / MMC
Cache Coherency Unit
System Memory Management Unit
FPGA/HPS Bridging
HPS IOs
Intel® Stratix® 10 FPGA Fabric
FPGA General Purpose I/Os
Hard Multiport DDR SDRAM Controller
Hard IIP for PCIe* standard
Transceivers

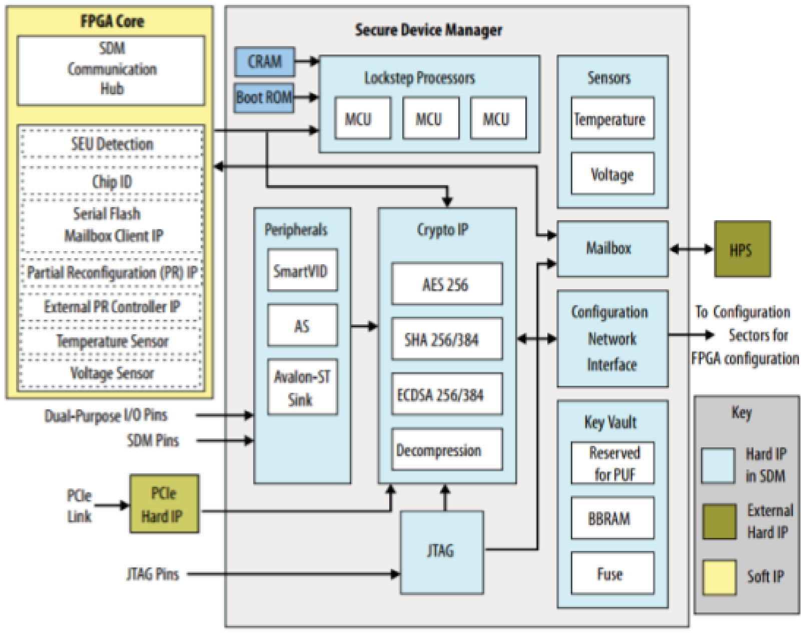
Similar architecture to Xilinx Zynq Ultrascale+ MPSoC

S. K. Ramegowda, Intel: Altera

Intel (Altera): Device Security

FPGA bitstream & boot image encryption and authentication/keys, etc.

The Secure Device Manager (SDM)



- The SDM is a triple-redundant processor-based module that manages configuration and security features
- The SDM interfaces with a configuration device (usually QSPI) which stores the FPGA bitstream and the First Stage Bootloader
- The SDM is also used as a peripheral to communicate with the configuration device later if needed

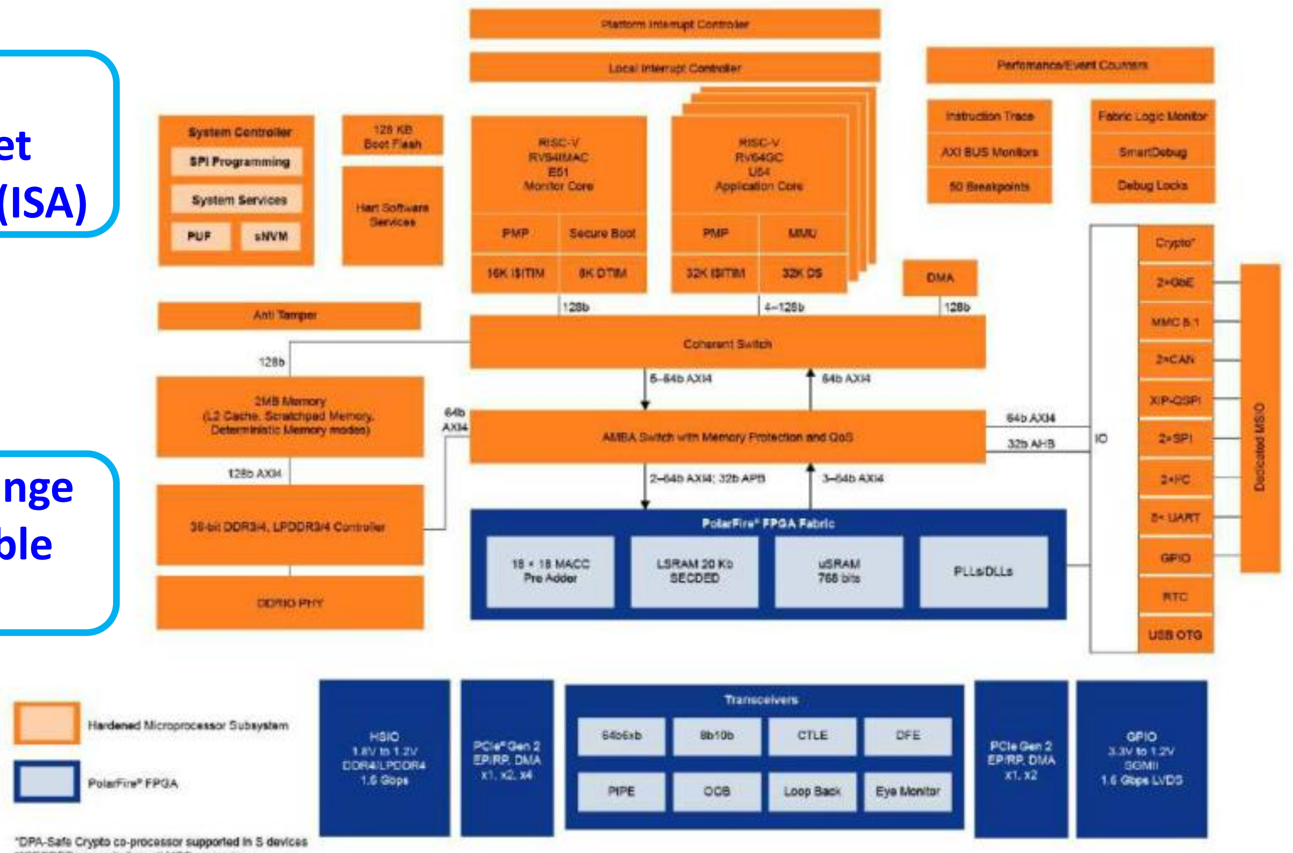
S. K. Ramegowda, Intel: Altera

Microchip: PolarFire SoC with RISC-V Processor Core

PolarFire® SoC - RISC-V Innovation Platform

Open-source Instruction-Set Architecture (ISA)

Low to mid-range FPGAs, available 2022



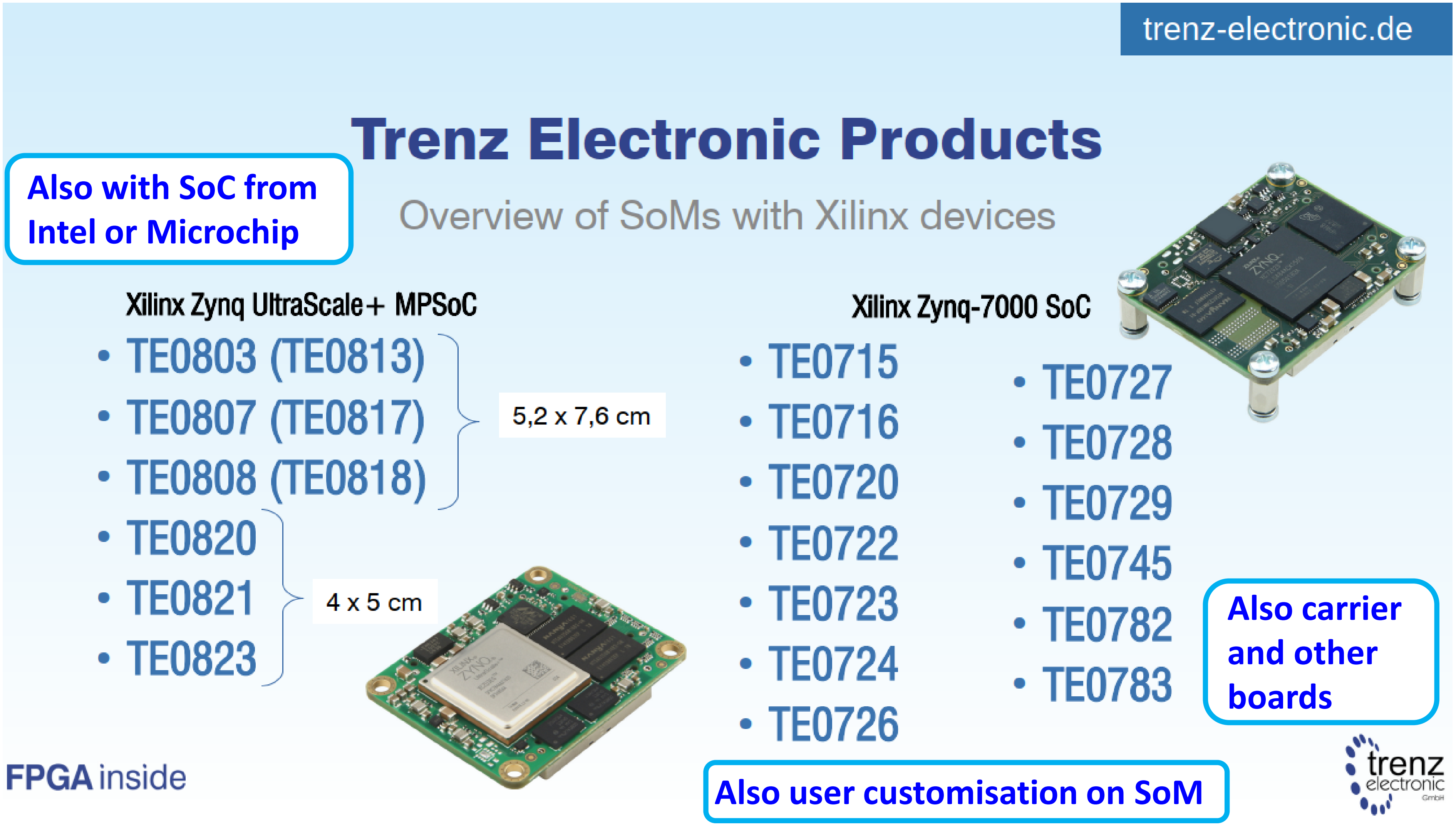
*DPA-Safe Crypto co-processor supported in S devices
**SECDED supported on all MSS memories

MICROCHIP

Something to be looked out for in future

G. Donzel, Avnet/Silica: Microchip

Trenz: Families of System-on-Module



trenz-electronic.de

Trenz Electronic Products

Overview of SoMs with Xilinx devices

Also with SoC from Intel or Microchip

Xilinx Zynq UltraScale+ MPSoC

- TE0803 (TE0813)
- TE0807 (TE0817)
- TE0808 (TE0818)

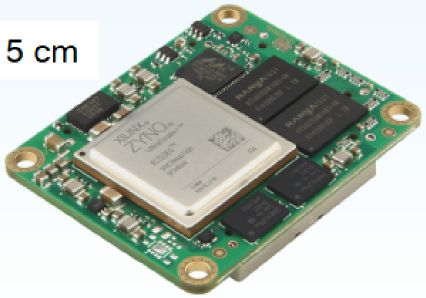
5,2 x 7,6 cm

- TE0820
- TE0821
- TE0823

4 x 5 cm

Xilinx Zynq-7000 SoC

- TE0715
- TE0716
- TE0720
- TE0722
- TE0723
- TE0724
- TE0726
- TE0727
- TE0728
- TE0729
- TE0745
- TE0782
- TE0783



Also carrier and other boards

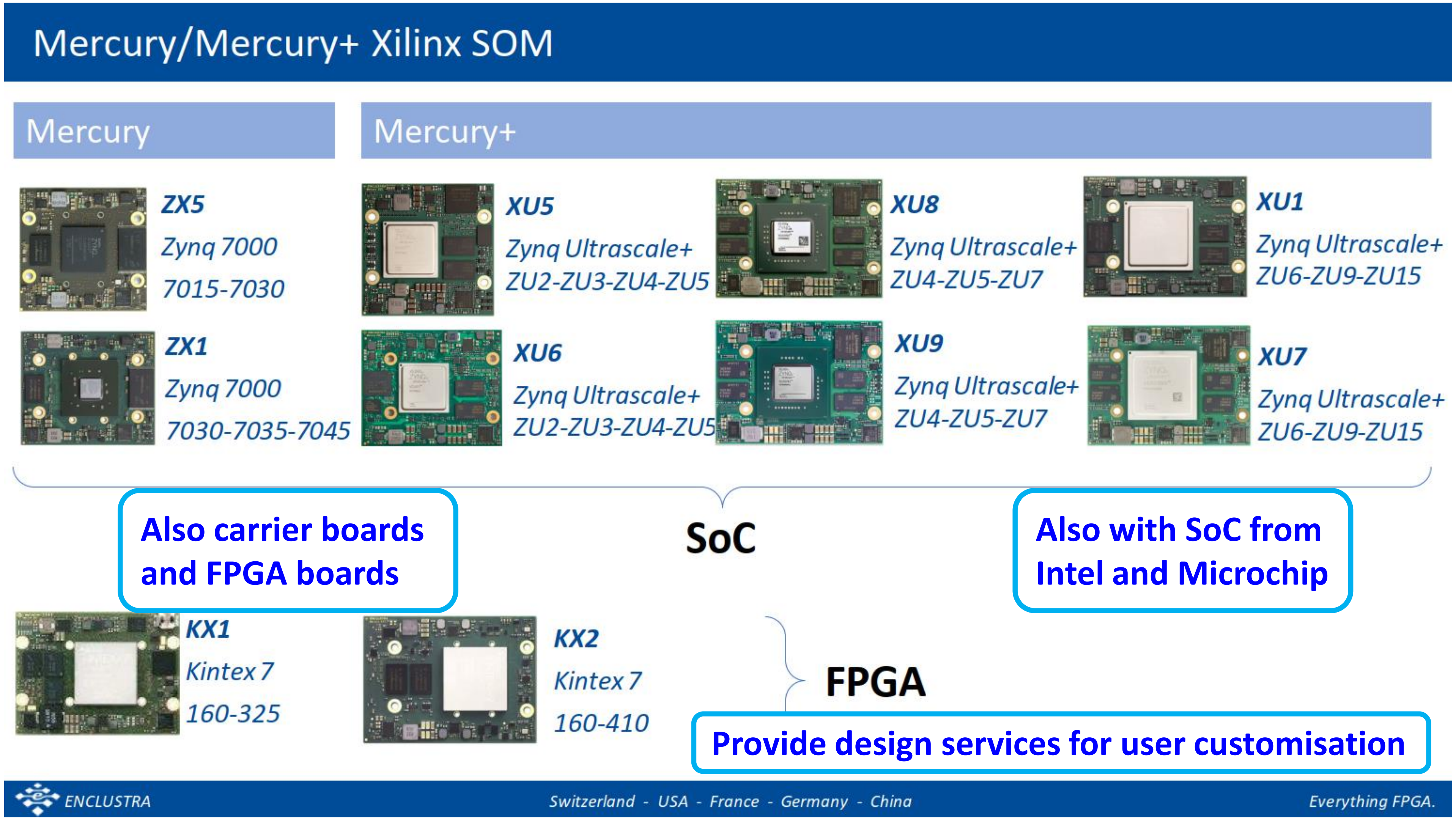
Also user customisation on SoM

FPGA inside

trenz electronic GmbH

M. Rohrmueller, Trenz Electronic GmbH: Trenz

Enclustra: Families of System-on-Module



Mercury/Mercury+ Xilinx SOM

Mercury		Mercury+					
ZX5 Zynq 7000 7015-7030		XU5 Zynq Ultrascale+ ZU2-ZU3-ZU4-ZU5		XU8 Zynq Ultrascale+ ZU4-ZU5-ZU7		XU1 Zynq Ultrascale+ ZU6-ZU9-ZU15	
ZX1 Zynq 7000 7030-7035-7045		XU6 Zynq Ultrascale+ ZU2-ZU3-ZU4-ZU5		XU9 Zynq Ultrascale+ ZU4-ZU5-ZU7		XU7 Zynq Ultrascale+ ZU6-ZU9-ZU15	

SoC

KX1
Kintex 7
160-325

KX2
Kintex 7
160-410

FPGA

ENCLUSTRA — Switzerland - USA - France - Germany - China — Everything FPGA.

Also carrier boards and FPGA boards

Also with SoC from Intel and Microchip

Provide design services for user customisation

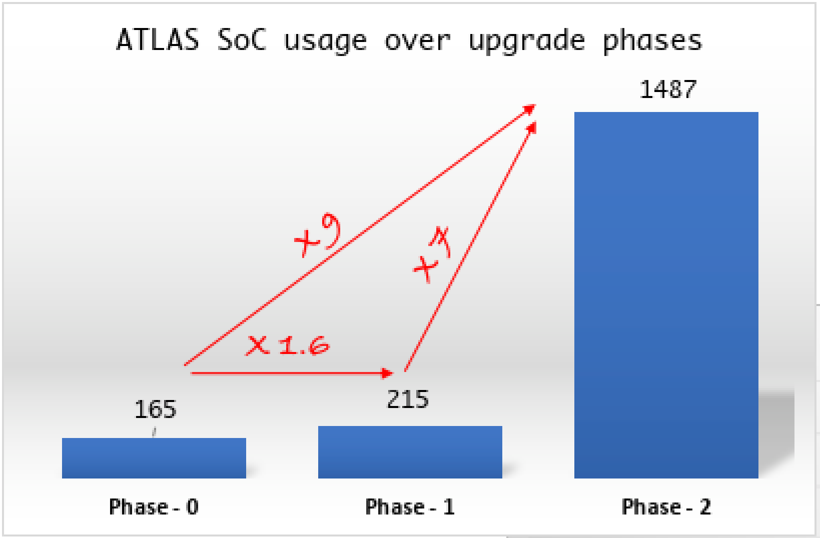
D. Ungureanu, Enclustra: Enclustra

Part 2: SoC Workshop

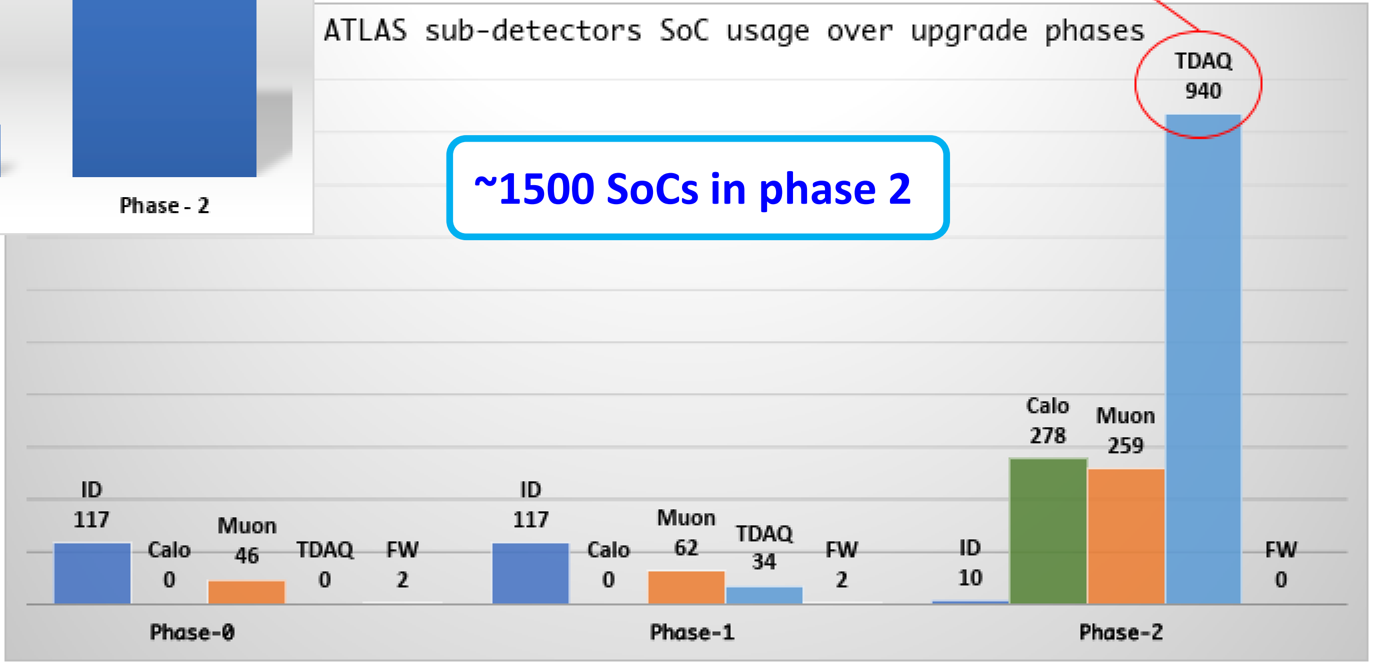
Overview Presentations

ATLAS: SoC Numbers

SoC usage evolution in ATLAS



Phase-II TDAQ has the highest (planned) SoC usage in ATLAS

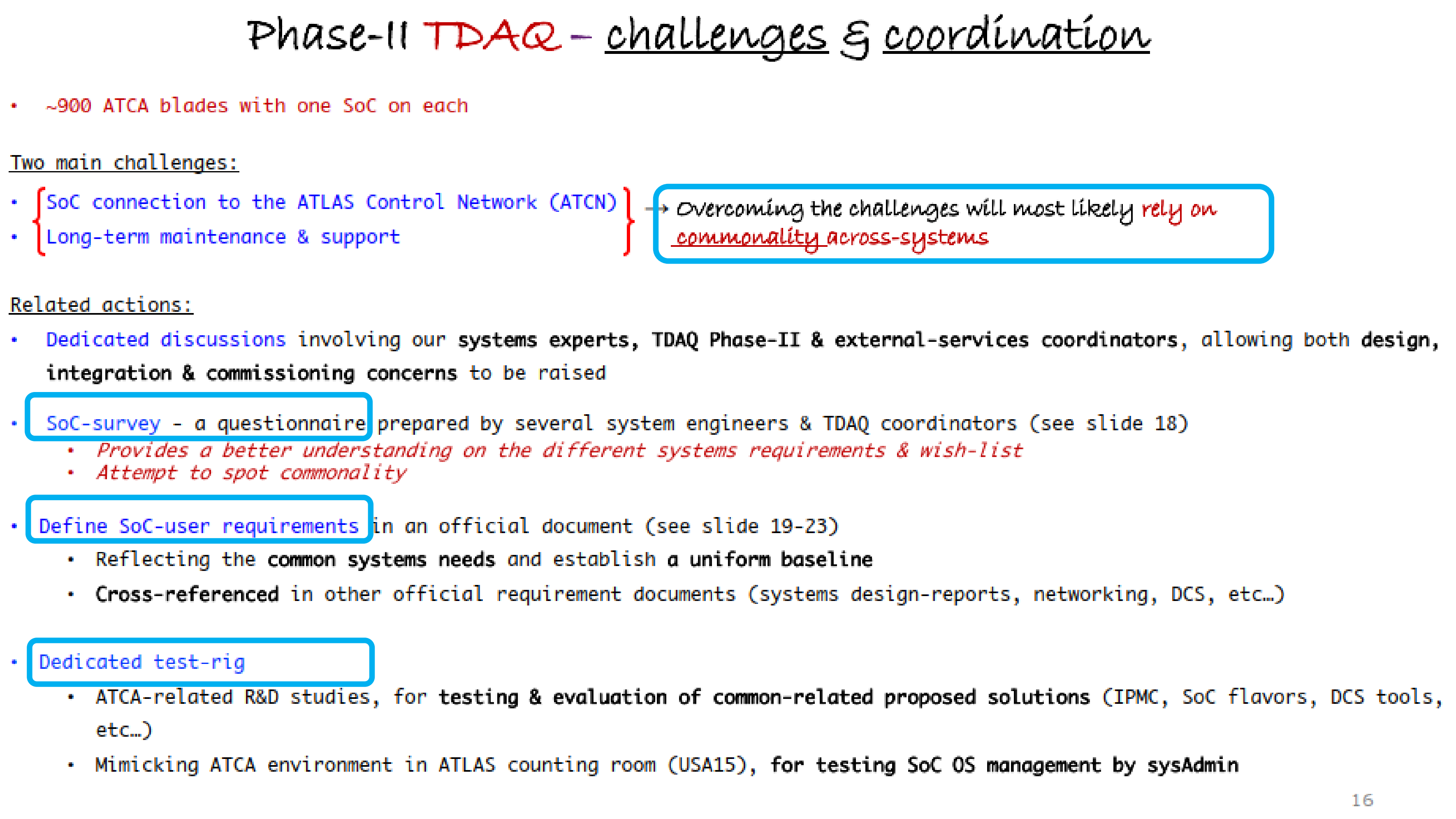


~1500 SoCs in phase 2

*Phase-2: As we know today, subject to change

R. Kopeliansky: ATLAS Overview

ATLAS: SoC Strategy



Phase-II TDAQ – challenges & coordination

- ~900 ATCA blades with one SoC on each

Two main challenges:

- SoC connection to the ATLAS Control Network (ATCN)
- Long-term maintenance & support

→ Overcoming the challenges will most likely rely on commonality across-systems

Related actions:

- Dedicated discussions involving our **systems experts, TDAQ Phase-II & external-services coordinators**, allowing both **design, integration & commissioning concerns** to be raised
- SoC-survey - a questionnaire prepared by several system engineers & TDAQ coordinators (see slide 18)
  - *Provides a better understanding on the different systems requirements & wish-list*
  - *Attempt to spot commonality*
- Define SoC-user requirements in an official document (see slide 19-23)
  - Reflecting the **common systems needs** and establish **a uniform baseline**
  - **Cross-referenced** in other official requirement documents (systems design-reports, networking, DCS, etc…)
- Dedicated test-rig
  - ATCA-related R&D studies, for **testing & evaluation of common-related proposed solutions** (IPMC, SoC flavors, DCS tools, etc…)
  - Mimicking ATCA environment in ATLAS counting room (USA15), **for testing SoC OS management by sysAdmin**

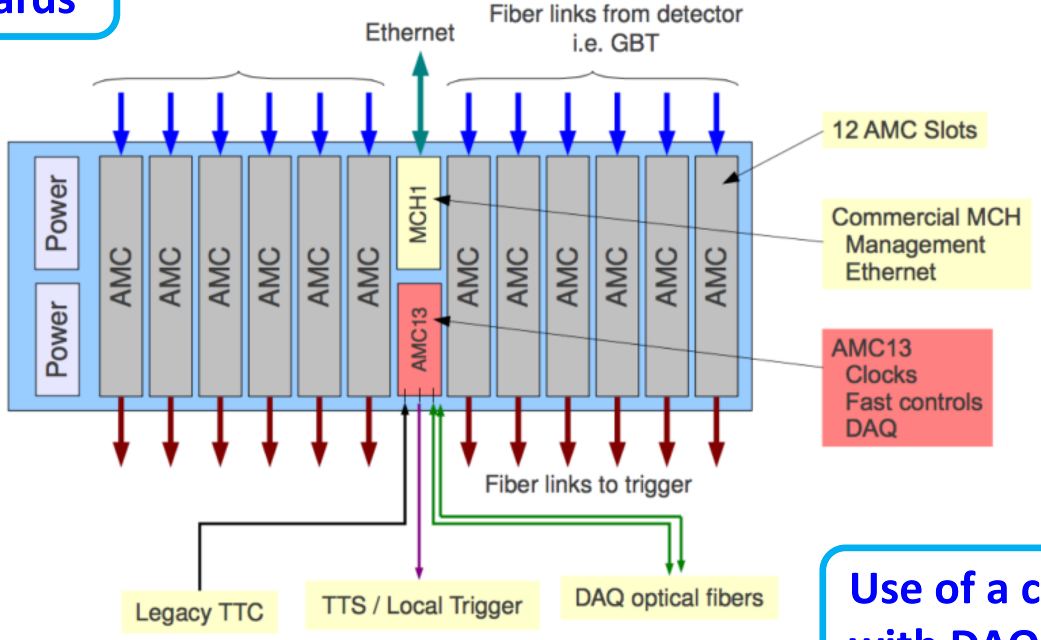
16

R. Kopeliansky: ATLAS Overview

CMS: Phase 1 (Run-2 & Run-3) - MicroTCA

MicroTCA is a spinoff from ATCA and AMC standards

Phase-1 typical MicroTCA crate



Use of a central hub card with DAQ functionality

- AMC13 in MCH2 slot
  - Fast Control+Timing HUB and DAQ concentrator (Boston University)
- Backplane
  - CMS specific allocation (clock, fast control, DAQ via fat pipes, ..)

F. Meijers: CMS Overview

CMS: Run 4 (Phase 2) - ATCA

Shelf = Hub (DTH) + Leaf Back-end Boards

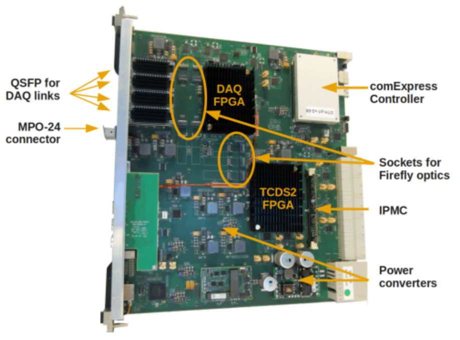
- **Hub slot 1**
  - Custom DTH (DAQ and Timing Hub)
    - Managed Ethernet switch (1 GbE to leaf boards, 10 GbE uplink)
    - TCDS (clock, TTC and TTS) using back plane serial links (10 Gbps)
    - DAQ with front panel links (eg FireFly), NOT over back-plane
- **Leaf Back-End boards**
  - About 5 platforms, re-purposed for different sub-dets, L1 trigger system
    - Apollo, APX, BCP, BMT, X2O, Serenity,
  - One or two main FPGAs (US+) and auxiliary small FPGA (for fast control)

DTH proto #2 uses a SoC

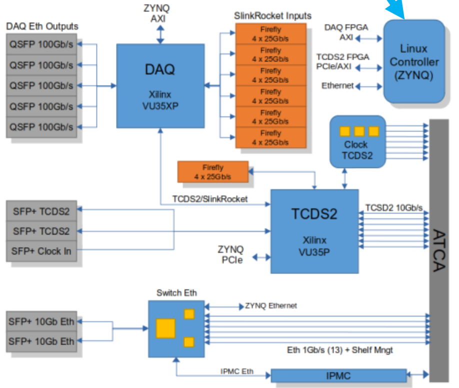
Use Case for On-board Controllers

- Two choices under consideration
  - Zynq
  - Com-express (option for Serenity)
- Scale: about 1500 boards
  - ~same size as HLT farm today
  - But much more diverse (about ~5 "platforms")

- DTH-Proto#1

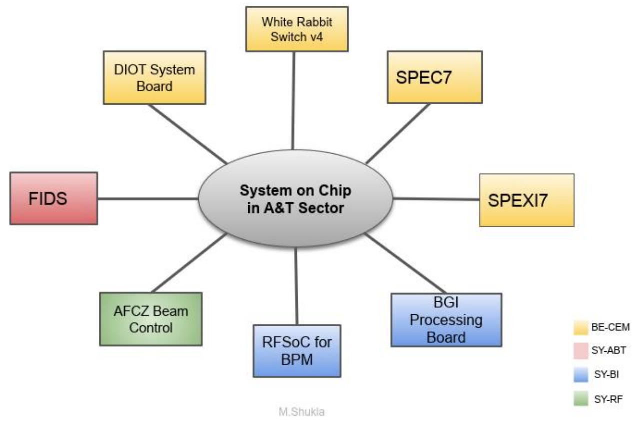


DTH-Proto#2 design



F. Meijers: CMS Overview

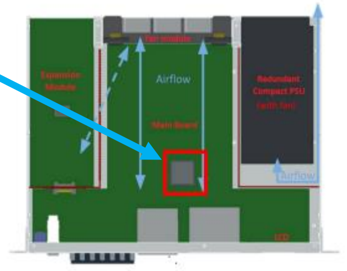
Accelerator&Technology Sector: Numerous SoC Projects (1)



Example: White Rabbit Switch

- Zynq Ultrascale+ SoC with support for up to 32.75 Gbps
- **PL**: SFP interface, White Rabbit PTP core supports upto 24 ports
- **PS**:
  - SNMP and diagnostics
  - Embedded Linux

SoC for precision-timing switching network



Courtesy: M.Lipinski

Example: Study of RFSoC for Beam Position Monitoring

- Evaluation board ZCU111 acquired as basis for prototype BPM processor
- RFSoC model XCZU28DR (Xilinx Zynq UltraScale+ architecture, gen. 1)

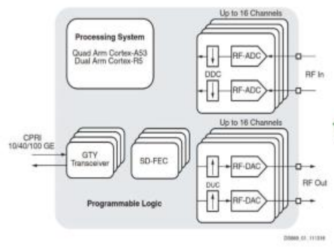


Figure 1: **Zynq UltraScale+ RFSoC**

Source: *Zynq UltraScale+ RFSoC Data Sheet: Overview (DS889)*

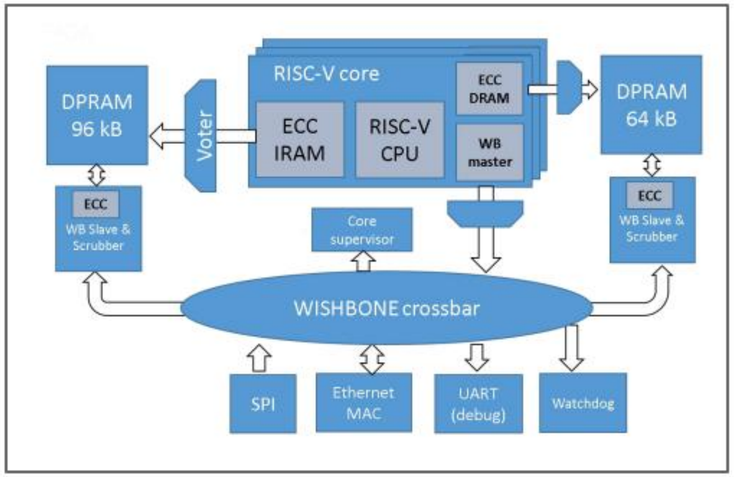
Courtesy: A.Boccardi, D.Bett, I.Degl'Innocenti

- Proof of Concept Implementation:
  - Acquire raw-data from Analog FE
  - Digitization of two planes pick-up signals
  - Data read-out from the memory for offline analysis
- Next, BPM Data-processing inside the FPGA
- **SoC:** Provides RF chain, DACs and ADCs on a single chip
- **PL**: Data Acquisition, Digitizing and Processing
- **PS**: Network Interface, Upstream data to FEC, RPU
- Simulation Framework for Study

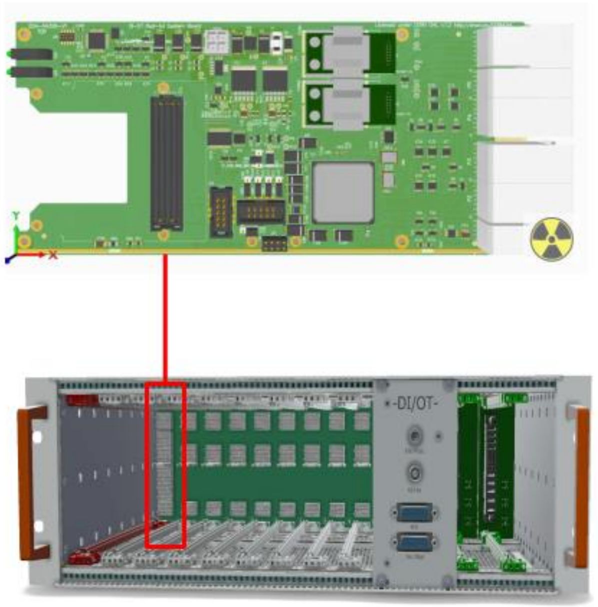
Accelerator&Technology Sector: A Number of SoC Projects (2)

Example: Hydra – SoC Architecture for Radiation-Tolerant System Board

SoC = FPGA design, contains soft RISC-V Cores



https://ohwr.org/project/hydra/wikis/home



- **System on Chip Study** conducted to know about the common needs and issues in A &T Sector
- With **FEC Linux Standardization project** - study the possible Linux distributions and recommend an option for the SoC platforms
- Study of **available tools** - Yocto, Buildroot, ELBE, Civil Infrastructure Project
- **CI/CD for gateware** - ongoing effort in BE-CEM and TE-EPC

Common approach for software framework

System-On-Chip Support Study, F.Vaga, BE-CO TM

https://gitlab.cern.ch/cce/cce

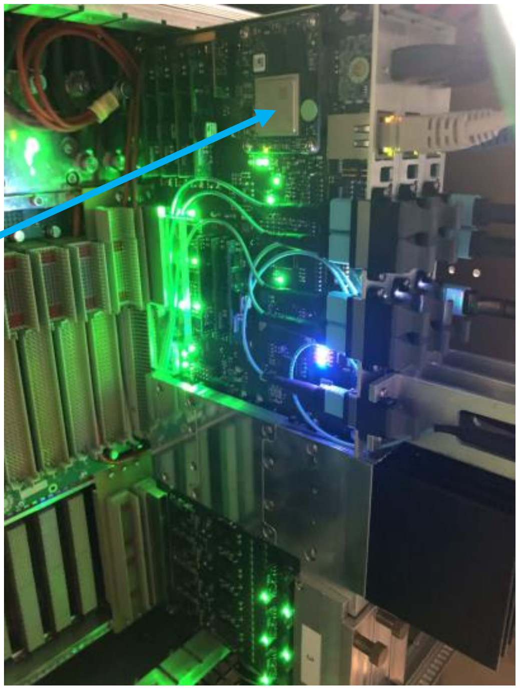
Courtesy:F.Vaga, K.Blantos

Part 2: SoC Workshop

Project Reports

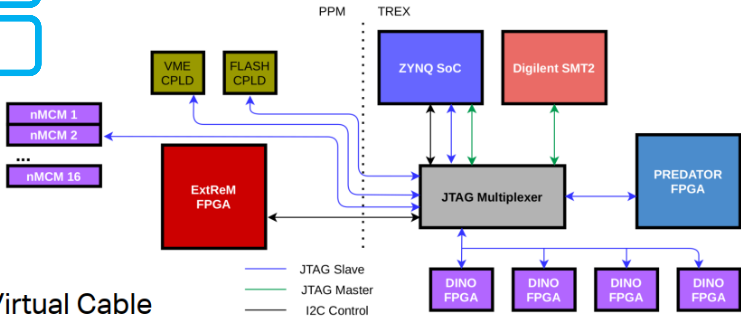
ATLAS: L1Calo TREX (phase 1) = 9U VME Rear-Transition Module

SoC on SoM= TE0820



Configuration

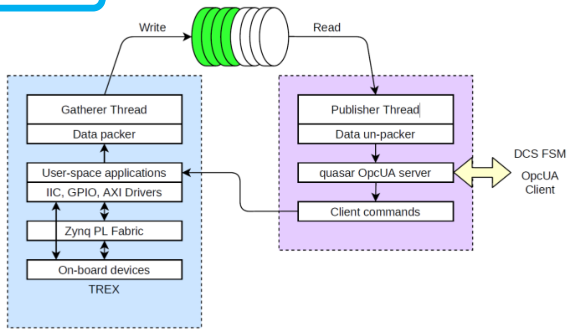
- Configuration of all on-board devices
- Programming of Power managers
- All FPGAs include I2C slaves
- Board power control



- Implementation of the Xilinx Virtual Cable
  - Programming of the TREX FPGAs
  - Bonus: allows to program CPLDs on the PPM side
  - Via Vivado or custom SVF
  - Programming of the FPGA flash memories (QSPI)
  - Heavily used for debugging

Slow-control

- Continuous gathering of environmental data
  - Voltage, Current and Temperature data
  - 17 devices
  - ~200 parameters in total
  - I2C is the main interface
  - Data stored in buffers

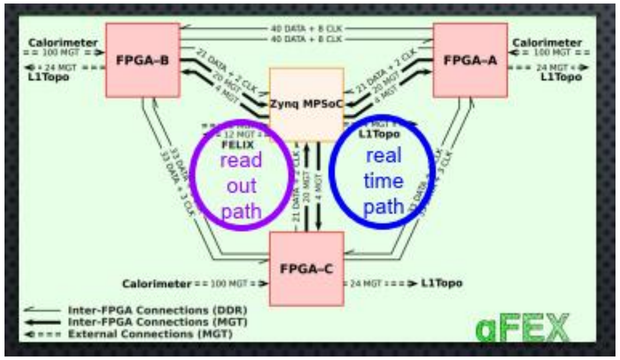


- Monitoring framework which interfaces with the quasar-based OpcUA server
  - Python3-based

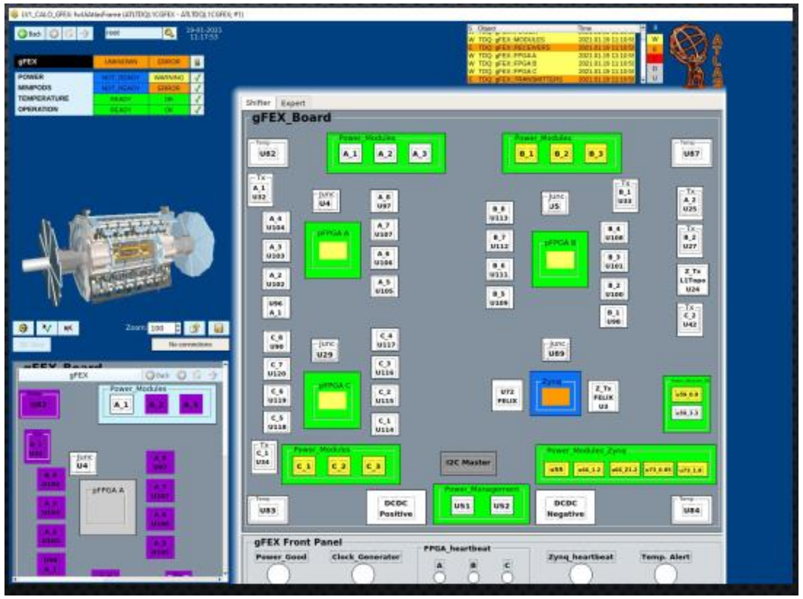
OPC-UA = OPC Unified Architecture: machine-to-machine communication protocol for industrial automation

T. Mkrtchyan: SoC-base Configuration and Monitoring of the ATLAS L1Calo TREX Module

ATLAS L1Calo Global Feature Extractor (gFEX) (phase 1)

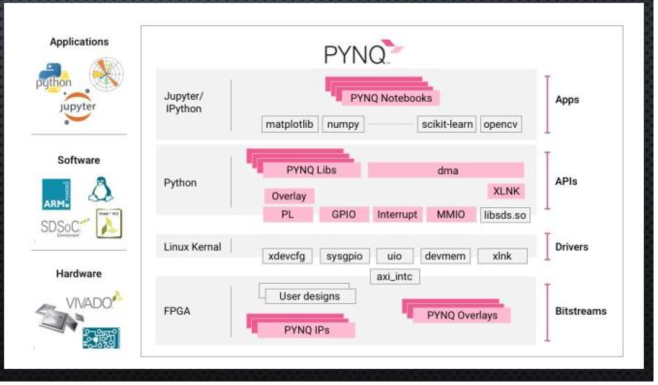


ZynqMP/PL provides triggering and readout



ZynqMP/PS provides slow control using quasar framework

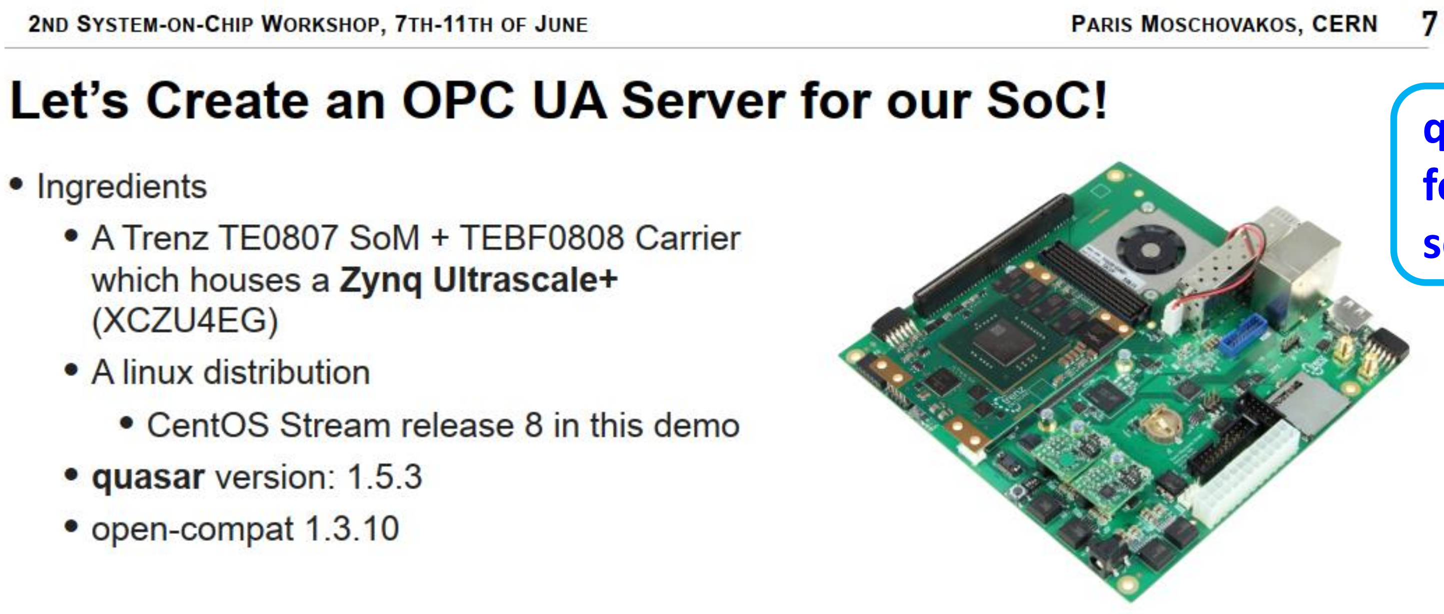
- PYNQ = **Py**thon Productivity for **Zynq**
- provides a Python interface which loads and processes bitstreams to create an API type reference to firmware objects, referred to as "overlays"
- Very nice way to utilize the AXI DMA setup as PYNQ already has DMA libraries
- Resulting code needed is extremely simple!



PYNQ = open-source framework using Python to read/write SoC/PL

E. Smith: The Zynq MPSoC in the gFEX Hardware Trigger of ATLAS

ATLAS Detector Control System (DCS) and SoC (1)



Let's Create an OPC UA Server for our SoC!

- Ingredients
  - A Trenz TE0807 SoM + TEBF0808 Carrier which houses a **Zynq Ultrascale+** (XCZU4EG)
  - A linux distribution
    - CentOS Stream release 8 in this demo
  - **quasar** version: 1.5.3
  - open-compat 1.3.10

quasar = framework for building OPC-UA servers

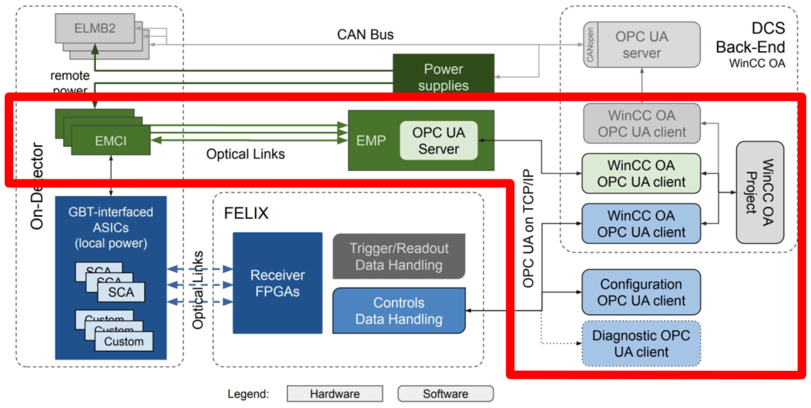
Demo (slides + recording) available in indico

News: MilkyWay = pure Python library for quasar

P. Moschovakos: SoC for Detector Control and their Applications

ATLAS Detector Control System (DCS) and SoC (2)

EMP – EMCI Path

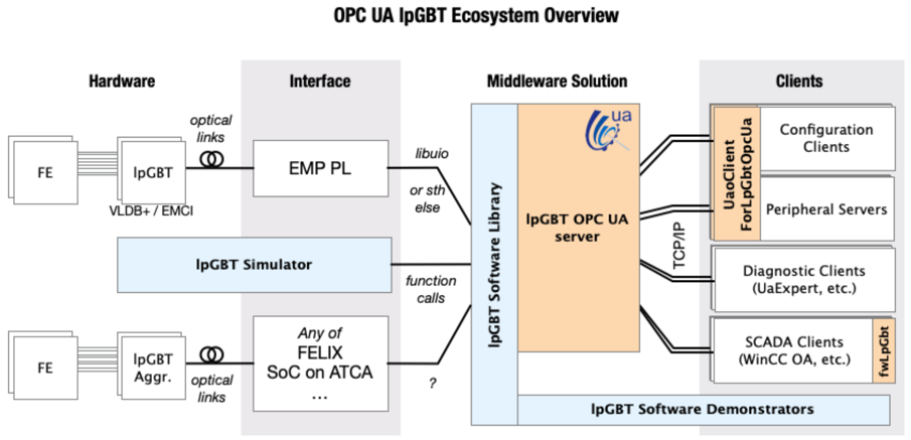


EMCI = Embedded Monitoring Control Interface

EMP = Embedded Monitoring Processor

In collaboration with EP-ESE-FE

EMP prototype: use TE0807 SoM (on TEBF0808 carrier)



Use lpGBT for communication with EMCI

+ Use CentOS Stream 8 as root file system

+ Extra: Running ARM Docker Containers on x86_64

P. Moschovakos: SoC for Detector Control and their Applications

Reconfigurable Cluster Element (RCE)

SLAC R&D project for generic SoC-Based DAQ/Trigger system

Successfully deployed for ATLAS Muon CSC Readout since 2014

RCE for Pixel RD53a/b readout

https://twiki.cern.ch/twiki/bin/viewauth/Atlas/RCEGen3SDK

- a new generation RCE system (prototyping)
  - Xilinx ZCU102 board with **UltraScale+ ZYNQ** MPSoC XCZU9EG
  - quad-core Arm Cortex-A53, 4GB of DDR4 memory, 64-bit
- ZCU102+FMC: a compact standalone bench-top DAQ
  - can scale up to ATCA form, like RCE gen-3: HSIO2 -> COB
  - miniDP (1 cmd + 4 data) for Rd53a/b readout with commercial DP cables
- Custom FMC adaptor to direct run RD53a/b
  - **SelectIO HP** pins for 160Mbps Cmd and 1.28Gbps data
    - demonstrates a compact high-density and economical solution to scale up to large number of channels
    - *MGT can not direct output 160Mbps Cmd
  - Retimer (160MHz clock, PLL, and flip-flop) on the FMC card to reduce Cmd stream random jitter to **2ps**, comparable to lpGBT e-link
  - adjustable Cmd pre-emphasis and data equalizer for lossy data transmission chain
  - FMC pin mapping is compatible with various FPGA boards

SLAC NATIONAL ACCELERATOR LABORATORY

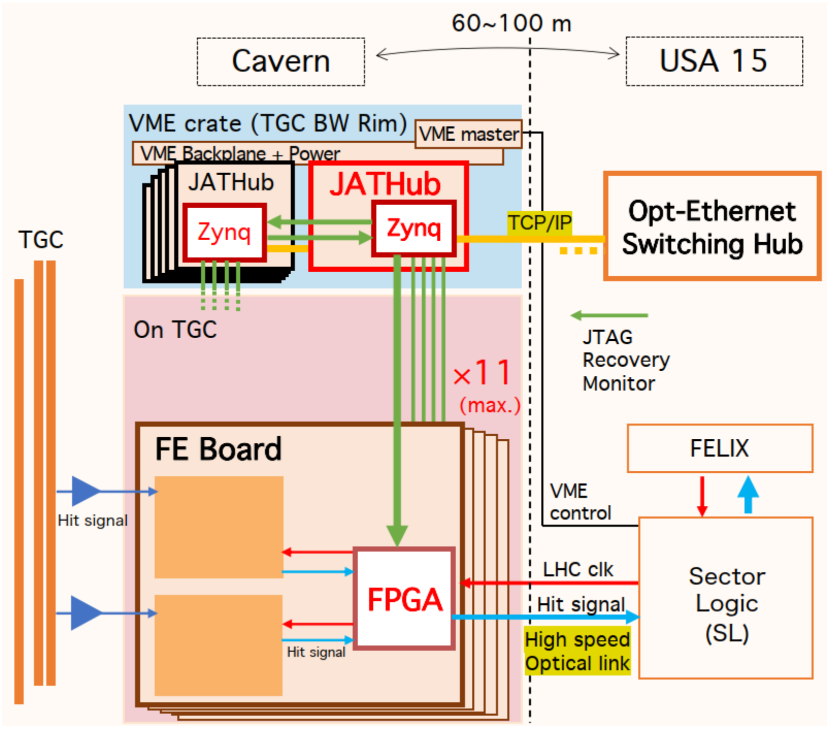
Z. Xu: SoC-based DAQ for the Pixel Readout Chip RD53a/b

ATLAS TGC JTAG Assistance Hub (JATHub) Module

- Program FPGA of FE board using JTAG
- Re-program FPGA when un-recoverable SEU occurred

- Is a VME Module x 148, controlling 1434 FE boards
- Use Zynq7000 as main device
- Redundant boot system
- Re-programming of JATHub

➢ Xilinx Virtual Cable (XVC) :
- Access to master JATHub with TCP/IP from Host PC and access to the slave module with JTAG, by running the XVC application in master JATHub (XVC is the TCP/IP based protocol accessing like JTAG)
- Enable to use the functions of Vivado running in a Host PC for remote FPGAs
  - Program FPGA and QSPI
  - Debug firmware logic
    - Integrated Logic Analyzer (ILA)
    - IBERT



- Additional functionality: LHC clock phase stability monitoring on FE board

ATLAS LAr Smart Rear-Transition Module (SRTM)

Functional Partitioning

Zynq Ultrascale+ MPSoC

- Arm Processor (PS in Xilinx terminology)

  Non-time critical system level functions: OS boot, login, board monitoring, GbE comm., configuration, infrastructure, logging, slow controls, …

- FPGA (PL in Xilinx terminology)

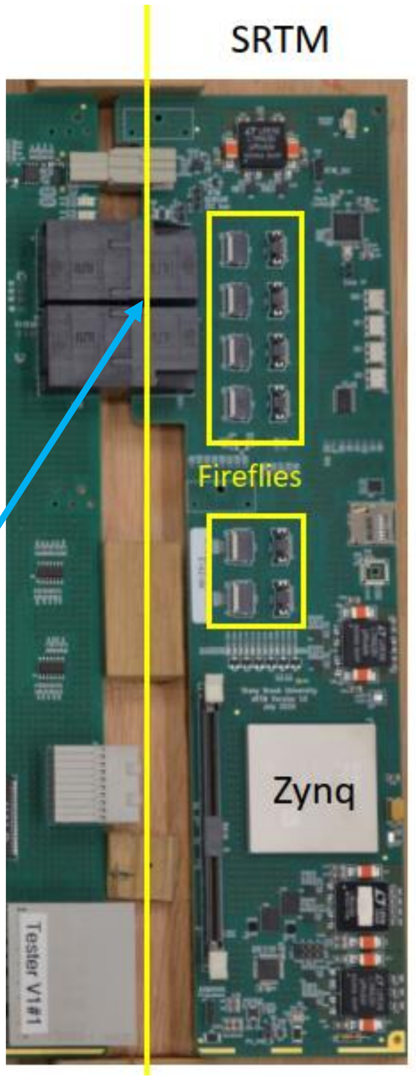
  Time critical data processing and handling: LHC clock recovery, data paths including any extra processing and reformatting; and higher-rate monitoring

SDK = Xilinx Software Development Kit

SDK for "User" code (defined as not in kernel or loadable modules)

- As for Vivado, script based builds
- Defined a simple directory structure and default locations
- Advantage is easy debugging via SDK.
- Some drawbacks: compilation by scripts and dependences

  Code examples:
  sensor monitoring (I2C),
  clock programming (I2C; GPIO),
  clock frequency checks (uio, AXI),
  firefly control (GPIO; I2C),
  hardware versioning (uio, AXI),
  LVDS prbs testing (uio, AXI),
  transceiver eyescan (uio, AXI),
  web interfaces to these



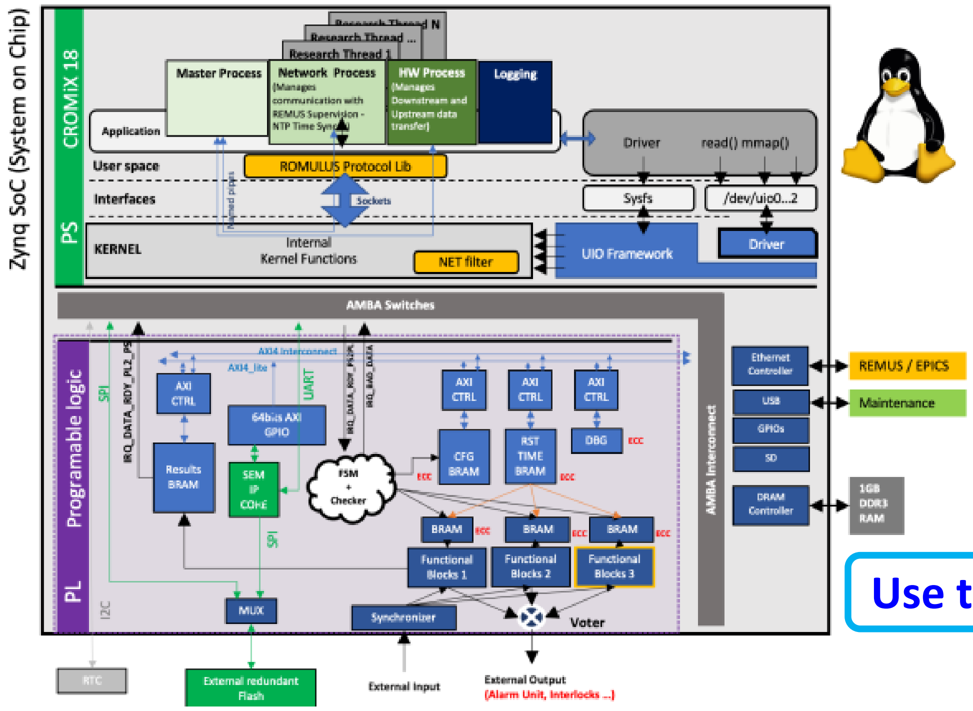
Side issue: 4x12-channel FireFly @ 25 Gbps Pass-Through

J. Hobbs: The ATLAS Smart Rear-Transition Module of the Liquid Argon Calorimeter

CERN Radiation Monitoring Electronics (CROME)

Almost 200 devices deployed at CERN

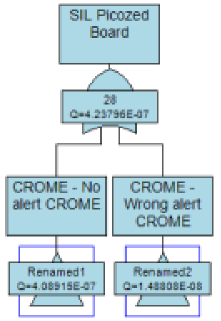
Zynq 7020: PS&PL architecture



High reliability:

Custom Zynq 7020 SoM developed with Avnet/Silica for both high reliability and cost efficiency. It is based on :

- Off the shelf PicoZED format (LT support)
- Reliability Availability and maintainability Analysis : quantified failure rate 1.10 fpmh
- We have ordered more than 200 SoMs.
- 170 SoM devices are used in operation (no failures so far)



Use triplicated logic

Also used at the European Spallation Source (ESS), where it is integrated with their EPICS control system

Demo available in indico

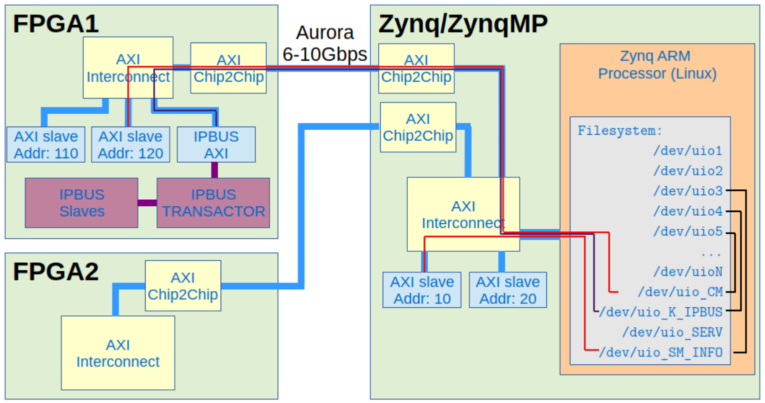
H. Boukabache, Y. Hast: Remote Management of SoC-based Radiation Monitors at CERN & ESS

Apollo Platform

Rev 2: commercial System-on-Module: Enclustra XU8

Selection: Automatic Generation of Code for Firmware/Software

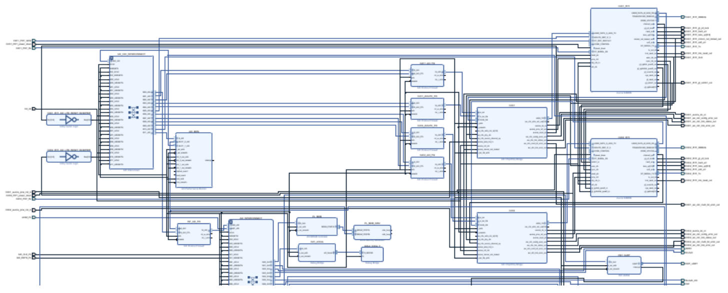
Typical architecture using Chip2Chip (C2C)



- Local and remote AXI slaves (remote via Xilinx Chip2Chip)
- Slaves memory-mapped using userspace UIO driver
- Modified μHAL for AXI access

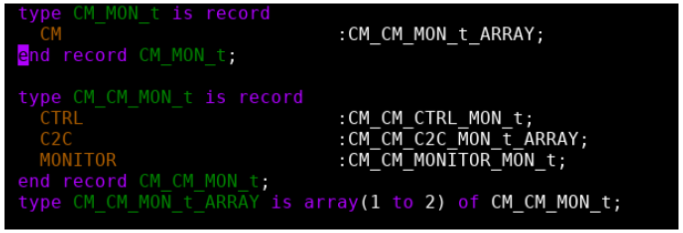
YAML slave file

- AXI_CONTROL_SETS
  - AXI interconnects to connect slaves to
  - Stores name and associated clk/resets
  - AXI slaves auto expand these interconnects
- AXI_SLAVES:
  - Each node is one AXI slave
  - TCL_CALL:
    tcl call to make for building slave
    AXI connection and addressing info to use
    Additional arguments for specific slave
  - XML:
    list of μHAL XML files for this slave
    First listed is the top for this slave
  - HDL:
    Controls automatic AXI register map decoder
    and package of records generation



YAML-driven Tcl:

- **creates AXI IP cores or PL AXI ports**
- **includes uHAL XML file and creates VHDL records and VHDL AXI ⇔ record decoder**



```
type CM_MON_t is record
  CM                                  :CM_CM_MON_t_ARRAY;
end record CM_MON_t;

type CM_CM_MON_t is record
  CTRL                                :CM_CM_CTRL_MON_t;
  C2C                                 :CM_CM_C2C_MON_t_ARRAY;
  MONITOR                             :CM_CM_MONITOR_MON_t;
end record CM_CM_MON_t;
type CM_CM_MON_t_ARRAY is array(1 to 2) of CM_CM_MON_t;
```

UIOuHAL access /dev/uioN to read/write

Use device tree overlays to add mappings at runtime

D. Gastler: Updates on SoC for the Apollo Platform

Serenity Platform: SoC for Board Management

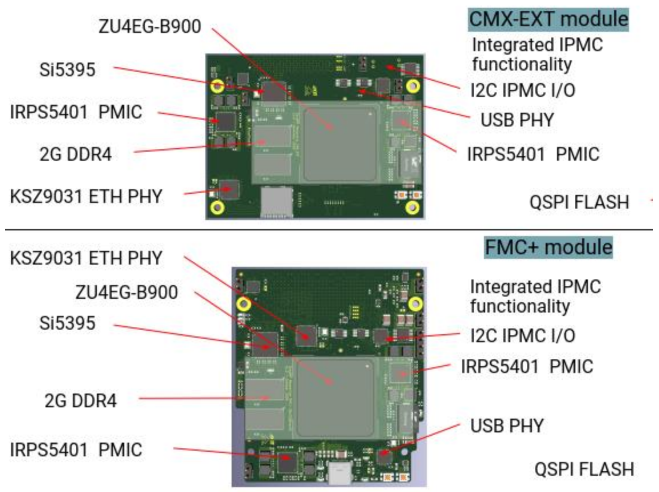
Custom System-on-Module

Technical requirements (Serenity specific)

- compatibility with CMX and FMC+ form factor
  - flexible choice between x86 CMX boards and the ZynqMP SoM
- specific requirements related to the integration of IPMC into ZynqMP
  - individual powering of the domains (LPD, FPD, PL)
  - IPMB circuitry
- availability of high speed transceivers limited on commercial boards
  - ZU4EG with 16 lanes not available

Soft requirements

- full control of the design sources
- long term availability
- designed using KiCAD, an open source EDA software



IPMC implementation on SoC

Pigeon Point IPMC software

- Based on VPX version for ZynqMP (BMR-ZNQ-VPX)
- Extension by KIT and Pigeon Point for ATCA compliance

OpenIPMC software

- hardware independent software implemented on 7 platforms so far (ZynqMP x4, ESP32, STM32 x2), proven portability and hardware independence (based on FreeRTOS)
- subset of PICMG specifications developed and tested
- implemented into two “production” platforms
  - ZynqMP mezzanines with integrated solution (this presentation)
  - OpenIPMC-HW designed with DIMM format using STM32 microcontrollers

Runs on ZynqMP R5 processor

Split-boot: minimal FSBL

- Idea: Separate configuration in two stages
- First stage (FSBL) loads basic configuration
  - **Minimal Vivado PS configuration** (e.g. DDR, ETH, UART)
  - allows to execute FSBL + ATF + uboot
- Second stage (uboot) loads full configuration
  - tftp load and apply **full application specific Vivado PS configuration**
  - SoC is mostly reconfigurable (exception are parts already in use like DDR)
  - Bitstream, linux kernel and rootfs are already tftp / nfs capable

Common boot strategy: full configuration from network

L. Ardila: ZynqMP-based Board Management Mezzanines for the Serenity ATCA Blades

N. Karcher:
SoCs for Readout Systems of Superconducting Circuits
$\longrightarrow$ use of RFSoC for controlling superconducting quantum bits

Slides available in indico

ATCA Processor (APx) Platform: Embedded Linux Developments

Custom System-on-Module

ELM1 (XC7Z045 based)

- A series of mezzanines with a single purpose and form factor, which vary in cost and power
  - ELM1: ZYNQ-7000, by U-Wisconsin
  - ELM2: Zynq UltraScale+, by U-Florida
- 84mm × 75mm form factor
- +12V Supply
- SD/QSPI primary/recovery boot options (IPMC-selectable)
- 512MB DDR3 Memory

Host configuration after boot: puppet apply

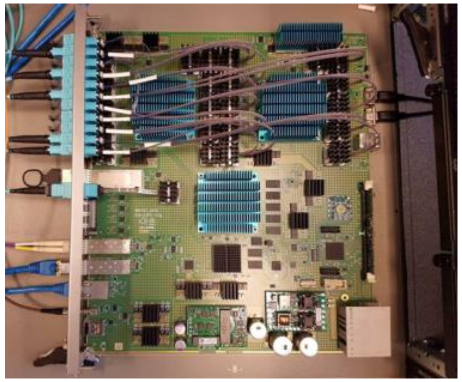
- We produced an APx puppet module which covers base requirements
  - Core basics (NTP, yum repos, requested packages)
  - Users & authentication (kerberos, nfs homedirs)
  - Base APx services (RPC, fpgaloader, sysmap, elmlink, etc)
- Primary configuration through Hiera: puppet mastery not required
- Layers for board-type and location allow easy, flexible configuration

+ Geographic Addressing

+ Sysmap for meaningful and consistent device naming and access

J. Tikalsky: APx Embedded Linux Developments

ATLAS MUCTPI (phase 1): System integration

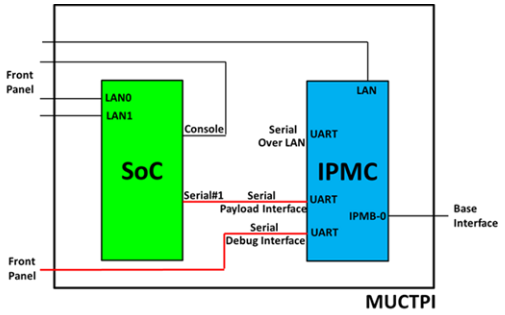


EP-ESE-BE in collaboration with EP-ADT-TR

Geographic addressing: SOC - IPMC

SoC has serial Payload Interface (PI) with CERN IPMC:

- SoC → IPMC: "Send Message" command to IPMC containing a "Get Shelf Address Info" to ShMgr ⇒ Shelf Address
- SoC →IPMC: "Get Address Info" commands to IPMC and to ShMgr using "SendMessage" command ⇒ Slot Number

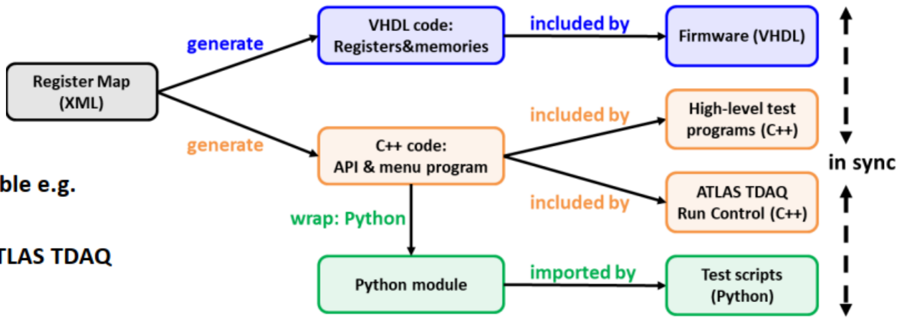


Implementation: FSBL, U-Boot, Linux/C++

Firmware/software generation

1) XML → VHDL and C++: hwcompiler (developed by L1CT)
2) C++ → Python wrapper: Simplified Wrapper and Interface Generator (SWIG), available e.g. from WLCG

For C++ and Python, calling of the hwcompiler and SWIG is fully integrated into the ATLAS TDAQ CMake environment:
⇒ automatic generation and wrapping, CMake rebuilds the full software



+ Use TDAQ/CMake and cross-compilation to build run control application

+ Full software work flow described and run in gitlab/CI pipelines

R. Spiwoks: Software Framework for the System-on-Chip of the ATLAS MUCTPI

CMS SoC System Aspects

DHCP/Client ID: use SoC – IPMC Interface

Client ID = Alternative to MAC address
Client ID = Shelf Address + Slot Location

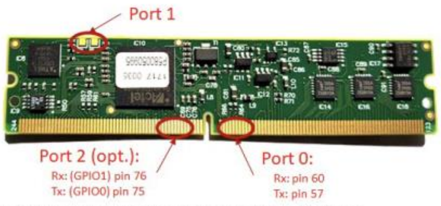
DHCP/Client ID at several stages:
U-Boot, kernel start-up, system services

- Using Trenz baseboard
  - TEBF0808-4A
  - Two UARTs on XMOD
    - One for ZYNQ PS
    - One for CPLD
      - For connecting to PS the CPLD's firmware had to be modified

UART from Trenz baseboard (XMOD)

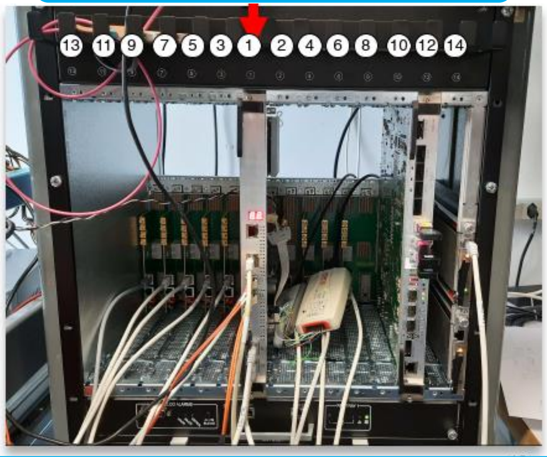
UART to IPMC in ATCA board

- Using edge pins on IPMC



UART availability described by Julian Mendez

Physical Slot Number



The slot numbering can be very confusing

Graceful Shutdown: use IPMC and SoC/PMU to shutdown kernel

- Extended to Zynq for graceful shutdown
  - Important e.g. un-mounting file systems
  - Two wires between Zynq and IPMC
    - **zynq_shutdown_request**
      - External interrupt triggers PS shutdown sequence
    - **zynq_shutdown_ack**
      - Set when shutdown completed, IPMC waits for this signal before shutting power
  - MIO pins routed to PMU
    - PMU input issues a shutdown request to the Linux kernel
    - PMU output changes its state after the Linux kernel is shut

Part 2: SoC Workshop
Tutorials

M. Husejko:
Setting up basic GitLab CI and CD Environment for Zynq-based Design

N. Dzemaili:
Reliable Booting and Upgrade System

M. Wyzlinski:
Continuous Integration for Building Software Infrastructure

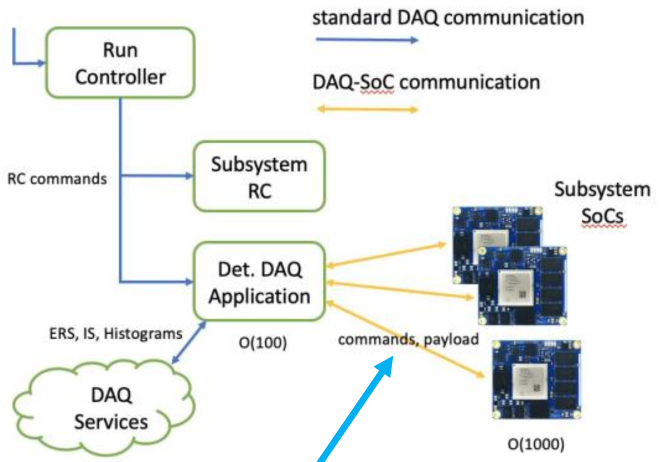
Material and recordings available in indico

Part 2: SoC Workshop

Common Issues

ATLAS SoC-DAQ Interface

Long-term maintainability of SoC:
Provide message-passing interface, which decouples TDAQ software from SoC



Client-Server Interface:
Client = DAQ/Run Control process
Server = SoC, execute cmd + sends async msg

Prerequisites and constraints

- It is assumed that SoC will host a standard, CERN/IT-certified Linux OS, like CentOS, and that a standard Ethernet-based network communication stack, including TCP/IP and HTTP libraries is available. A standard system gcc compiler and development kit is also available on SoC (or in a Linux cross-compilation environment).
- It is assumed that SoC is accessible from ATLAS Control Network (ATCN) and TDAQ Control networks and is addressable by its DNS name.
- It is assumed that a common middle layer s/w stack like LCG s/w, providing widely used libraries like Boost Intel TBB, ROOT etc) and standard TDAQ s/w **may be not supported** by either SOC s/w or h/w platform
- Implementation of SoC-DAQI is foreseen to be lightweight and on the SoC (server) side to be independent from TDAQ s/w and to rely only on s/w packages available in standard repositories for SoC OS, thus **not imposing any additional requirements or restrictions** on the TDAQ s/w design coming from the SoC architecture.
- Implementation should be **stable and supported for decades** of the experiment lifetime

Progress

- Having the requirements finalized, the next steps are:
  - technologies evaluation
    - HTTP ← we are here, looks flexible and promising
    - gRPC ?
    - …
  - a toy example
  - a prototype implementation (due to Oct 2021)
- Development environment for prototyping and validation:
  - Centos8 ARM running in QEMU on CC7 x86_64 host
  - can compile, run and connect from the host

A. Kazarov: ATLAS TDAQ Online Software Plans for SoC

CERN CentOS Stream 8 and ARM64 Linux Support

Change of CentOS Linux 8 EoL

- On the 8th of December 2020, the CentOS Project announced it was shifting its focus from CentOS Linux (C8) to CentOS Stream (CS)
- During the same announcement, the end-of-life for C8 was reduced from being a 10 year supported distribution to being 2 years ①
- Support for C8 will end at 31.12.2021
- End of Life for CERN CentOS 7 (CC7) has not changed (30.06.2024)

And after 2024?

③
- "Linux Future Committee": working group to assess the impact of current Linux usage at CERN
  - Meetings are invitation only, however all minutes/presentations/etc. are public at https://indico.cern.ch/category/13390/
- CERN is working with Fermilab and other facilities to decide on a path forward for the HEP community at large

Possible Options: CentOS Stream 9, RHEL "no/low cost", new Enterprise Linux Clones, e.g. Alma, Rocky

CentOS Stream 8 at CERN

- CentOS Stream 8 is a supported operating system
  - ② CERN controlled test/prod system updates **support until AUG-2024**
    - https://linux.web.cern.ch/updates/cs8/
  - OpenStack and Docker images
  - Supported with IT-related configuration management infrastructure
  - locmap is available for hosts not managed by IT centralised configuration management

What we provide *for ARM now*

④ **CentOS Stream 8 for aarch64**

- x86_64 and `aarch64` as first-class citizens
- Daily snapshots
  - http://linuxsoft.cern.ch/cern/centos/s8-snapshots/
- Daily Testing release: that day's snapshot
  - http://linuxsoft.cern.ch/cern/centos/s8-testing/
- Weekly Production release: updated on Wednesday to the previous Wednesday's snapshot
  - http://linuxsoft.cern.ch/cern/centos/s8/
- Docker image
  - `gitlab-registry.cern.ch/linuxsupport/cs8-base`
- Basic CERN RPMs
  - `CERN-CA-certs`, `cern-config-users`, `cern-get-keytab`, `cern-krb5-conf`, etc.

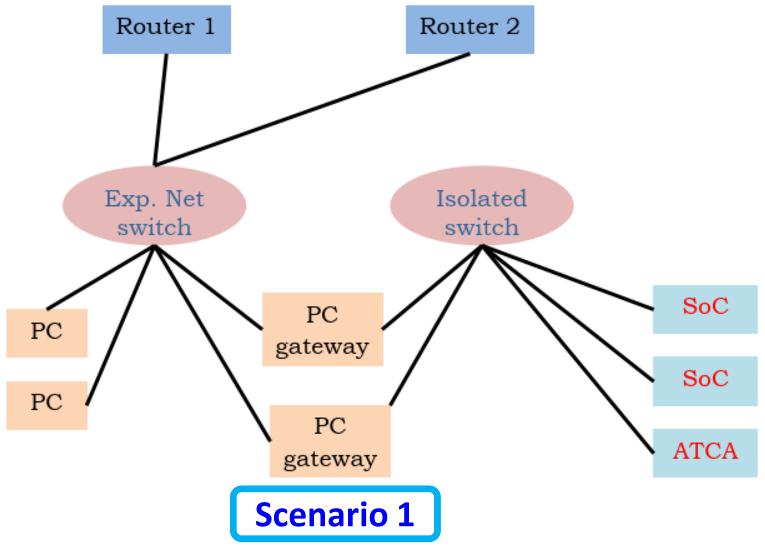
A. Iribarren: CentOS Stream 8 and ARM64 Linux Support

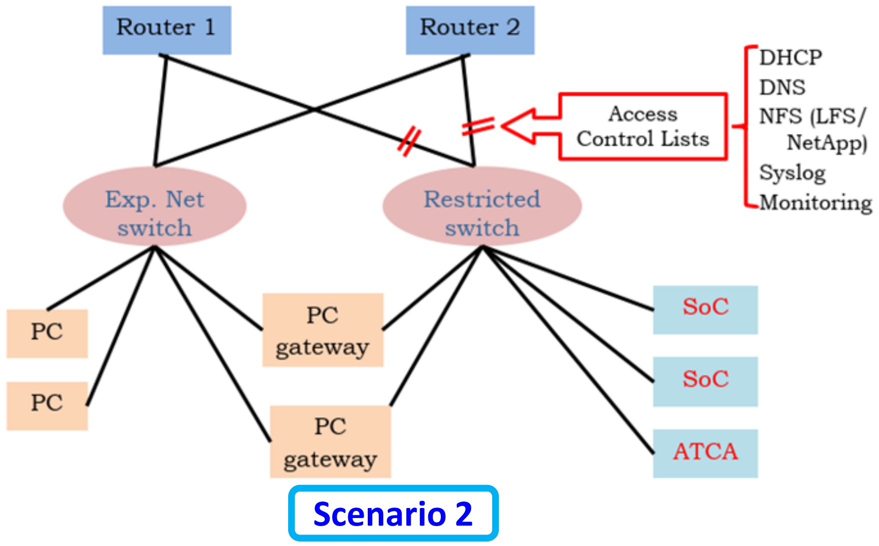
System and Network Administration (phase 1)

- All CERN users have agreed to Operational Circular N°5 (OC5) and the subsidiary rules and guidelines defining the use of CERN Computing facilities (includes the experiment facilities)
- The Computing and Network Infrastructure for Controls (CNIC) security policy defines the Experiment domain rules:
  - ★ https://edms.cern.ch/document/584092

…

- Embedded Linux devices are currently isolated
  - ★ Centrally maintained node acting as a Gateway, connected to the experiment network AND to an isolated "private" switch
  - ★ Or network equivalent (see later)





M. Dobson: System and Network Administration Aspects of Zynq MPSoC Devices in the Experiments

System and Network Administration (phase 2)

- Network isolation could be very cumbersome
  - ★ For all the reasons stated earlier, exacerbated by sheer number
  - ★ Gateway PCs could be a significant number
- Restricted Network are probably the viable alternative
  - ★ Restrictions but with many core services available
  - ★ Potentially getting rid of Gateway PC and allow selective connectivity
  - ★ Gateway PC may need to be replaced by "Buffer/Supervisor" PC (see later)

●●●

- ATCA spec. foresees usage of Client ID based DHCP
  - ★ Each Shelf is identified by a Shelf address (an arbitrary <=20 character long ASCII string):
    - ➢ Should be unique in a DHCP domain
    - ➢ For us it could be "Building", "Rack" and "U" (always unique): USC55-S1A10-10
    - ➢ Could drop "Building" in experiment networks
    - ➢ Configuration of shelf done when installed (script to run over serial line from laptop)
  - ★ IPMC gets this info (FRU data) from Shelf Manager
  - ★ ATCA boards are located in the shelf by the Physical Slot Number
    - ➢ Primary site type is 00h and primary site number is Physical slot number
  - ★ Secondary site type and secondary site number can be used to identify sub-elements of an ATCA board (e.g. AMC Modules):
    - ➢ Could be used to identify Zynq, FPGAs, Switches
    - ➢ Propose to use values from OEM range (see next slide)

Proposal: use DHCP/Client ID

Proposal:
Shelf Address = building/rack/unit

Proposal:
Slot location = primary(type,number)/secondary(type/number)

- Geographical
  - ★ Based on Shelf Address, Slot, Function and Index/Number
  - ★ Prepended by "ATCA"

●●●

  - ★ Controller (Zynq or Com-e):
    - ➢ ATCA-USC55-S1A10-10-XX-CTRL-1/2/3

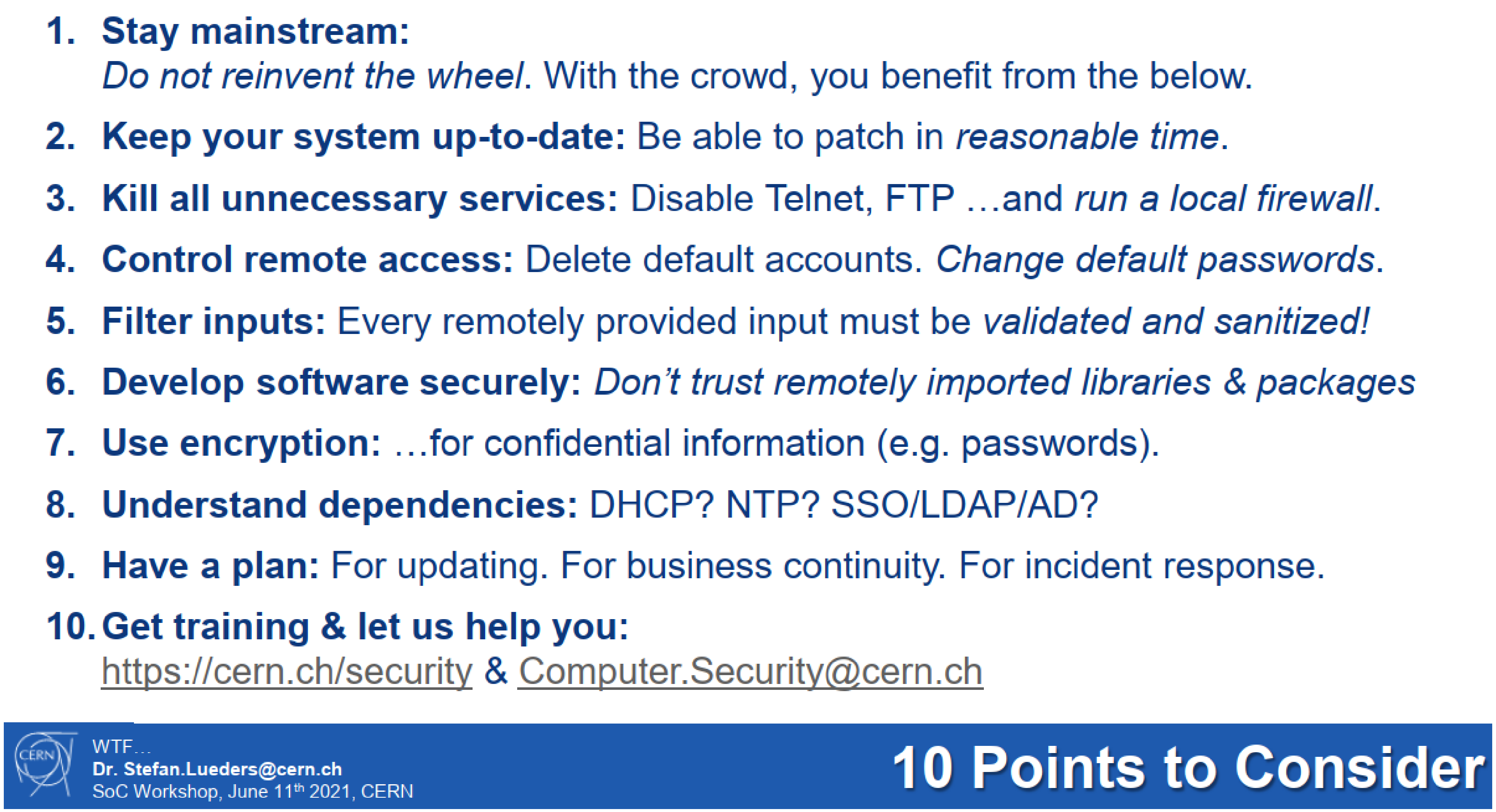
Proposal for DNS Names

●●●

- Need discussions with sub-detectors HW producers to understand network requirements to build infrastructure

M. Dobson: System and Network Administration Aspects of Zynq MPSoC Devices in the Experiments

The “golden” computer security rules



1. **Stay mainstream:**
   Do not reinvent the wheel. With the crowd, you benefit from the below.
2. **Keep your system up-to-date:** Be able to patch in *reasonable time*.
3. **Kill all unnecessary services:** Disable Telnet, FTP …and *run a local firewall*.
4. **Control remote access:** Delete default accounts. *Change default passwords*.
5. **Filter inputs:** Every remotely provided input must be *validated and sanitized!*
6. **Develop software securely:** *Don’t trust remotely imported libraries & packages*
7. **Use encryption:** …for confidential information (e.g. passwords).
8. **Understand dependencies:** DHCP? NTP? SSO/LDAP/AD?
9. **Have a plan:** For updating. For business continuity. For incident response.
10. **Get training & let us help you:**
    https://cern.ch/security & Computer.Security@cern.ch

WTF…
Dr. Stefan.Lueders@cern.ch
SoC Workshop, June 11th 2021, CERN

10 Points to Consider

S. Lueders: CERN IT Security

Part 2: SoC Workshop

Summary

Summary – Use of SoC

A lot of interest in System-on-Chips:

Across experiments, accelerators, and radiation protection

Wide range of uses: interactive access, protocol/converter, hardware/slow control, run/operational control, trigger, readout, event monitoring, etc.

SoCs are very powerful, flexible devices with a lot of potential

A prediction for ATLAS and CMS foresees about 1500 SoCs in phase 2 each

System aspects:

- How to organise SoCs into manageable systems?
- How to provide secure connection to the technical control network?
- How to provide long-term maintenance and support?

$\Rightarrow$ *"Overcoming the challenges will most likely rely on commonality across systems."*

Summary - Hardware

On the hardware side:

- **All systems are using the same hardware families:**
  Xilinx Zynq7000 and Zynq Ultrascale+ MPSoC
  Many Zynq7000 projects are moving to use Zynq Ultrascale+ MPSoC (but not all!)

- **Many systems are moving towards a System-on-Module (SoM) in order to overcome the problem of hardware obsolescence.**
- **There are a number of different SoMs being used:**
  Avnet, Enclustra, Trenz, Custom SoM, Xilinx Kria, …

Summary – Software

On the software side:

- **Operating system:**
  - There was large agreement to use CentOS; in Run 3, C7 and CS8 will be used; in the long-term, *we need to see what future Linux will be selected (support for ARM64 is recognised)*
- **User application software:**
  - Better understanding of eco system, i.e. building system and user software
    $\Rightarrow$ *sharing of software could be improved*
- **Network:**
  - Organisation: number of SoCs, network requirements, naming schemes, etc.
    $\Rightarrow$ *proposals for CMS & ATLAS are being put together*
- **Booting:**
  - Strategy to boot as little as possible from local SoC resources, and as much as possible from network; in addition, to boot as reliably as possible $\Rightarrow$ *mechanism under construction*

$\Rightarrow$ *Many elements in hand, need to put them together to build a framework for centrally managed and maintained SoCs in the experiments*

Conclusion & Lookout

- **The SoC Workshop was success: SoCs are very popular, versatile, and powerful devices.**
- **SoCs are already in use today, and will be used more in the future, in particular, in ATLAS&CMS, but also in beams and radioprotection.**
- **The SoC organising committee will follow up on common solutions; will continue to organise interest group meetings; plans to organise another SoC workshop in 1 or 2 years**

Thanks for you attention!
Questions?