System-on-Chip (SoC) for Control of ATCA Boards

- Introduction
- Overview of the Use of SoC
- Summary

With contributions from

... many thanks!
**Control of Electronics Modules**

**Control** - send control commands:
- e.g. start, stop, pause, reset, etc.

**Configuration** - load configuration data:
- **Hardware related:**
  - settings of clock, power, and optical modules, etc.,
  - FPGA (firmware) configuration,
  - usually using industry standards like I2C, SPI, JTAG, etc.
- **Run control related, i.e. related to physics/run (which is the functionality implemented in firmware):**
  - Control register and/or FPGA memories, e.g. processing settings, look-up tables etc.

**Monitoring** - collect monitoring data:
- **Hardware related:**
  - Operational values of hardware, e.g. temperatures, voltages, optical power, etc.
- **Run control related:**
  - Status registers, counters/rates of physics events, data of some selected physics events, etc.

**Example: ATLAS**

**ACES**

**26 APR 2018**

**R. Spiwoks**
What changes with ATCA?

Legacy systems use **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 → IPMI → shelf manager → SCADA  
**Run control**:  
Via hub module and base interface, or directly to ATCA blade ...  
⇒ **Need a new control strategy!**
IPbus over UDP with FPGA

- IPbus is used with μTCA-based systems developed for the CMS phase-1 upgrade, when there was no real alternative to FPGAs, successfully deployed for L1 Trigger, Pixel, HCAL:
  - Originally developed by University of Minnesota, and now maintained by CMS: based on firmware and software
  - IPbus is a protocol based on UDP. It defines a transaction as a request and response; several transactions can be concatenated in the same UDP packet
  - The FPGA runs firmware which receives IPbus requests, performs read/write transactions with the other (processing) FPGAs, and sends back responses
  - The PC runs run control software which uses a software library (uHAL) with a user-friendly register map
  - The ControlHub software service provides a transparent reliable transport layer over UDP

- IPbus has evolved, and today is a protocol not exclusive to FPGAs and which can run over different physical media, including UDP+ControlHub, TCP, SPI, PCIe
Alternative to FPGA: System-on-Chip

**System-on-Chip:**
Processor system + programmable logic, “CPU and FPGA”

→ **Xilinx Zynq:**
- **Zynq SoC:** ARM v7 (32-bit, 1-2 cores)
- **Zynq UltraScale+ MPSoC:** ARM v8 (64-bit, 2-4 cores)

- **Processor system (PS):** like CPU, multi-core ARM processors with memory and peripherals, e.g. GbEthernet, I2C, etc., can be run “bare-metal” with Xilinx software or using an operating system, e.g. Linux

- **Programmable logic (PL):** like FPGA, can interface to the processor FPGAs, using Xilinx AXI Chip2Chip protocol
Using a System-on-Chip

- Use a System-on-Chip (SoC):
  - The **processor part runs software**, communicates with the experiment run control system and performs read/write transactions with the peripherals, i.e. for hardware control,
  - the **programmable logic provides** read/write transactions *with the other FPGAs*, i.e. for run control
- The SoC can be used in connection with a PC running all the application software *and the SoC is “relatively dumb”*, or it can be use like the PC itself or like the SBC in VME *and be “relatively intelligent”*
Overview of Use of SoC

Different existing and future, i.e. phase I and phase II, projects for electronics modules using System-on-Chip and ATCA:

• What is the project?
• What is the SoC being used for?
• What SoC is being used?
• What software is being used?
Readout Driver (ROD) for the Cathode Strip Chambers (CSC) of the ATLAS muon system:

- Readout for 32 CSCs
- One ATCA shelf with 6 Cluster On Board (COBs) and Rear Transition Modules (RTMs)
- Each COB is equipped with 5 Reconfigurable Cluster Elements (RCE): 4 Data Processing Modules (DPM) and 1 Data Transmission Module (DTM)
- Each DPM has 2 Xilinx Zynq Z045 and each DTM 1 Xilinx Zynq Z030, each COB has a 24-port 10 GbE Fulcrum Ethernet switch for network connection
- The CSC RODs were installed in ATLAS in 2014. They have been running very stably since.

**COB**
- DPM: FEX and sending to SLINK
- DTM: manage trigger and busy signals

**RTM**
- GLINKs from CSC chamber front-end cards and SLINKs
Single ATCA board: 1 COB with 2 RCE.
ATLAS TTC interface in the RTM.
SFP connections to VLDB (frontends) and S-Link
16 ATLAS-IBL-like tracker modules.
2 Time-of-flight systems.

In 2016 a HSIO-based system (High Speed Input Output) was installed, it is supposed to be replaced by the COB-based system during a technical stop soon or during LS2.

The same setup has been used for the Inner Tracker Pixel Test Readout.
SLAC COB Software Environment

- **AFP + CSC/DTM:** ArchLinux
  - Rolling updates challenging, no stable platform for common middleware
- **Originally ATLAS TDAQ ported for AFP, but later abandoned for RCF**
  - TDAQ packages originally ported to ArchLinux/Zynq: IS, IPC, OH, PMG
  - Port used crosstools-ng and CMT:
    - Should be much simpler with CMAKE
  - **Remote Call Framework (RCF):**
    - Class serializing (programmatically, no IDL compiler)
    - Network patterns (based on BOOST ASIO)
- **CSC/DPM runs RTEMS:**
  - CSC feature extraction, data formatting, S-Link transmission, and control program
- **CSC/DPM and CSC/DTM use SUN/RPC and JSON configuration data**
- Successfully booted Zynq with **Centos 7/ARM:**
  - Considering to migrate from ArchLinux soon
  - Fast bug fixes provided by community
  - Release based Linux distribution more suitable to run common middleware (TDAQ)
  - Does not reinvent the concept of a Linux distribution (in contrast to buildroot and YOCTO)
  - Still requires **board-specific kernel** builds
uTCA-based Calorimeter Trigger Processor

First CMS ZYNQ SoC card (Phase-1 upgrade)

- 22 cards (incl. spares) deployed in production in 2015 and running stably since as part of calorimeter L1 trigger chain
- Also in widespread use as a Phase-2 demonstrator platforms in several CMS upgrade projects and multiple institutions*

Key designs points:

- Virtex-7 690T FPGA with 80 MGTs for data processing
  - 69RX/48TX 10Gbps optical links, 12RX/TX backplane MGT links
- ZYNQ XC7045 SoC FPGA with dual ARM Cortex-A9 CPU used as the board control endpoint
- Embedded Xilinx PetaLinux running on the ZYNQ

Use of the ZYNQ SoC:

- Provides board configuration support (booting FPGA, configuring memories, clocks, optics, etc.)
- Acts as primary network-connected TCP endpoint on the board during online operation
- AXI Chip-to-Chip employed to seamlessly extend the memory space of the ZYNQ SoC to the Virtex-7 FPGA.
- Xilinx Virtual Cable (XVC) service running on the ZYNQ replaces a physical cable with a TCP/IP connection to Xilinx toolset
- Eye Scan Capability of IBERT ported to ZYNQ → Integrated Eye Scan (IES)
  - (see backup slide for more details)
- Application software based on Remote Procedure Call (RPC)

*CMS Phase 2 Demonstrators:
Track Trigger: CERN, Cornell, Rutgers
Correlator Trigger: CERN, FNAL, Wisconsin
EMU, HB/EB Cal Readout and Trigger: CERN, FNAL, Princeton, Texas A&M, UCLA, Virginia
“ELM”: Embedded Linux Mezzanine (Phase-2 upgrade)

Key design points:

• Control point for **ATCA in mezzanine** form factor (84mm x 75mm)
  – **stretched version of COMe mini, with a second MGT connector** added and PC/AT pins repurposed to ZYNQ PL IOs

• Adopted by multiple CMS Phase-2 upgrade back end subsystems

• Sub-project of APx consortium (see backup slide)

• ELM1: 7-Series ZYNQ, XC7Z045-2 (8 GTX MGTs)

• Embedded Xilinx PetaLinux running on the ZYNQ

• Programmable IO: >24@3.3V, 74@1.8V, 8x 10Gbps MGTs

• ELM2: ZYNQ Ultrascale+ (2019)

**Use of the ZYNQ SoC:**

• Extending the concept successfully pioneered with CTP7 design

• Low latency access point tightly integrated with workhorse FPGA(s)

• **MGT-based AXI Chip-to-chip bridge to processing FPGA(s)**

• **PL-based 10GbE TCP/IP Offload Engine**
  – fast data transfer to processing FPGA(s) for large memory config IO and test stand DAQ applications

• Interface to PCIe/SSD

• **Application software based on Remote Procedure Call (RPC), using Google Protocol Buffers**

---

**Present CMS Phase 2 Intended Use:**
Calorimeter, Correlator and Muon Triggers
ECAL Barrel, CSC and GEM readouts.
CMS – Barrel Calorimeter Processor

Barrel Calorimeter Processor: A common hardware platform targeted for the electromagnetic (EB) and hadron (HB) calorimeters in the barrel region of CMS

- **Functionality of the BCP**
  - Control of the on-detector electronics of EB or HB
  - Data concentration from 24 Front-End towers in EB or two wedge in HB per BCP
  - Digital Spike (noise) suppression and Trigger Primitive Generation
  - Provides single crystal trigger primitives to the trigger system

- **Use of the SoC:**
  - Configuration, control and monitoring of the board:
    - Hardware control of clock, optical modules + configuration of FPGAs
    - Run control of processing FPGAs: read/write status/control registers and memories/LUTs
    - BCP demonstrator board is in development and will use the Embedded Linux Mezzanine (ELM) board with Xilinx Zync SoC.
    - Future BCP prototype will be developed to host two FPGAs
    - Each FPGA will be controlled by the ELM using one bi-directional SERDES lane
    - Interface between user and ELM is via 1 GbE switch
    - A second lane will be used via the ELM as 10 GbE (not shown) and will be useful for stand-alone data collection for test beams and other development tasks

- **Operating System = embedded Linux:**
  Specific distribution to be decided

Interactive access:
- ssh, mpeek & mpoke, Xilinx Virtual Cable, FPGA configuration, IBERT test programs, etc.

Application software:
- Private implementation similar to Remote Procedure Call (RPC) flexible & extendible (C++)
ZYNQ-IPMC

Key designs points:

- **Open-source non-NDA HW/SW/FW** ATCA IPMI controller—IPMC (Phase-2 upgrade)
- Developed at U. of Wisconsin
- **Adopted by multiple CMS Phase-2 upgrade back end subsystems**
- Sub-project of APx consortium (see backup slide)
- Real-time monitoring, diagnostic and control of ATCA board services
- **244-pin MiniDIMM form factor, ~100 3.3V configurable IOs in ZYNQ PL**
- ZYNQ XC7Z020 SoC
- 16 ADC channels (16-bit) for fast response signal monitoring

**Use of the ZYNQ SoC:**

- Cortex-A9 @ 600+ MHz backed by 256MB of RAM is a radical upgrade in resources compared to typical IPMC hardware platforms
- **C++/FreeRTOS-based runtime framework**
- Employing a true object-oriented approach to the design without performance penalty
- **HW design to be fully published**, schematics, PCB, artwork, BOM, for independent production
- **SW design to be released under GPL**, no NDA required to use/reference
ATLAS Level0 Muon MDT Trigger Processor

64 Blades for 16 sectors, barrel/forward, +/-Z

- Main FPGA (e.g. KU15P)
  - all Fiber I/O at \( \leq 10 \)Gbps
- 3 Processing Mezzanines
- CERN IPMC – Xilinx XVC, back-up path for FPGA config and board communications
- Zynq (likely mezzanine)
  - Linux OS (Petalinux tools)
  - "IPBus-like" communications for register access, downloading
  - "BUTool" CLI scripting tool (based on CMS "AMC13Tool"
- Ethernet switch: 1Gbps to Zynq, 100 Mbps to IPMC
Pixel DTC

SoC and infrastructure very similar to ATLAS L0MDT

Not yet clear if this blade will belong to a family or consortium!
CMS HL-LHC Track Trigger

Cornell is designing the FPGA-based ATCA processing board for the Track Trigger system. The μTCA “YUGE” (Your Ultrascale Gigabit Evaluator) board allows early evaluation of planned technology, including 25 Gbit links, UltraScale+ FPGAs, and the Zynq SOC module.

- Zynq SOC – YUGE has 7020 chip on an Enclustra Mars ZX2 module, using a 200-pin SODIMM connector
  - first ATCA will use Zynq UltraScale+ MPSoC, also on a pluggable module
- Evaluating both Petalinux and FreeRTOS for either “smart” or “dumb” functionality
- The Zynq interfaces to all hardware: some control is autonomous, other via network commands
  - Xilinx AXI Chip2Chip to main FPGA for register and memory configuration and monitoring
  - Xilinx Virtual Cable (Virtual JTAG) to main FPGA for programming and debugging
  - I2C bus to control and configure the optical interfaces and clock distribution
  - Zynq PL to control power supply sequencing for the main FPGA

CMS HL - LHC Track Trigger

Charlie Strohman crs5@cornell.edu
CMS Serenity

*(Not using Xilinx Zynq but similar approach)*

Prototype for studying control and monitoring on ATCA blades, and for evaluating different FPGAs

To be used for Outer Tracker DTC, HGCAL DAQ & TPG, and Level-1 Trigger

**• COM Express Card:**
  - Pluggable embedded CPU
  - COM Express Type 10 Mini: 84mm x 55mm
  - Quad-core Atom, up to 2.2 GHz, 8 GB RAM, an SSD can be attached
  - PXE (net boot) and BMC (Board Management Control)

**• PCIe interface is boundary:**
  - Raw PCIe or IPBus over PCIe
  - FPGA vendor neutral
  - If COM Express Type 7 Mini was available it would provide even more connectivity, e.g. more PCIe, 4 x 10 GbE, etc.
  - Running uHAL (software library for IPBus) over PCIe

**• Advantages of COM-Express:**
  - A non-proprietary industry standard – many vendors, originates from the same standards’ body (PICMG) as ATCA
  - Supports non-proprietary operating systems
  - Provides a clean separation of hardware, firmware and software functionality
  - More powerful processor – could totally eliminate any need for separate rack-PCs, run DAQ directly on the ATCA blade in test stands
**gFEX - Global Feature EXtractor**

Part of the phase 1 Upgrade of the ATLAS Level-1 Calorimeter Trigger

- **Board architecture**
  - 3x UltraScale+ Virtex Processor FPGAs - U9P
    - Full calorimeter sent over 300 input fibers to a single gFEX board
    - Run processing algorithms to identify jets, calculate global event observables (Large-R jets, MET), provide trigger objects
  - 1x UltraScale+ Zynq MPSoC - ZU19
    - Configuration, Slow-control and Monitoring
  - ATCA platform with IPMC fully control and monitoring capability

- **Custom Operating System (OS)**
  - Using OpenEmbedded/Yocto toolchain to cross-compile Linux kernel; maintain official meta-l1calo layer registered with OpenEmbedded
  - Gone through multiple iterations of boards with both pre-UltraScale+ Zynq and current UltraScale+ Zynq MPSoC
  - Kernel patches and custom recipes written for python packages to be included in custom-developed OS on top of mainline Linux
  - Undergoing implementation of ATLAS DCS OPC-UA project, as well as a recipe for cross-compiling ROOT6

- **Functionality of ZU19**
  - Configuration of clocks, FPGAs, golden image, etc. through a well-defined user-facing register map
  - Slow-control to enable read-out fibers through read-out path, put board in Run-Mode state for detector operation, activate/deactivate input/output fibers on a link-by-link basis, read/write QSPI flash, and other registers in defined user-facing register map
  - Slow-control/monitoring via IPBus protocol and any other communication protocol allowed through software developed in-house, “ironman”, a single-threaded, reactor-based event-driven callback Python framework; also supports websockets and HTTP streaming
  - Monitoring of the onboard hardware components through I2C
gFEX - Future applications of SoCs for ATLAS/HEP

● **On-board trigger-level analysis**
  ○ PL firmware and on-board Linux OS can increase the DAQ flexibility and capacity in data collection and processing
  ○ High-speed LPDDR4 memory (2.4 Gbps) can provide significant capacity for buffering
  ○ **Could do “real” analysis** (e.g. invariant mass calculations) **as well as calibration-style computations** (e.g. di-jet balance). Depending on the proposed application, these could provide additional information to L1Topo, for example

● **Multi-processor system opportunities**
  ○ Low-power ARM Mali-400 **GPU** may aid in vector calculations (e.g. matrix multiplication for calorimeter image processing)
  ○ **Real-time processor** for trigger-level analysis (see above), real-time calibration, and object-level monitoring
Muon-to-Central-Trigger-Processor Interface

- **Functionality of the MUCTPI:**
  - Data concentration of 208 muon trigger sector inputs
  - Overlap removal, i.e. avoid double counting of single muons
  - Provide muon trigger objects to topological trigger and muon trigger object multiplicity to Central Trigger Processor

- **Use of the SoC:**
  Configuration, control and monitoring of the board:
  - Hardware control of clock, power, optical modules + configuration of FPGAs
  - Run control of processing FPGAs: read/write status/control registers and memories/LUTs
  - Current prototype with Zynq SoC, build a new prototype with Zynq Ultrascale+ MPSoC

- **Operating System = embedded Linux:**
  Built using Yocto project and Xilinx meta layer (+ sources)

- **Interactive access:**
  ssh, peek&poke, Xilinx Virtual Cable, FPGA configuration, IBERT test programs, etc.

- **Application software:**
  Two alternative approaches:
  1. RPC-like private implementation: flexible/extendible (C++)
  2. Port of ATLAS TDAQ to Zynq: run control application + ROOT on Zynq
    → Investigating the use of CentOS ...
ATLAS – TREX (Tile Rear-Extension Module)

• Functionality of the TREX:
  - Rear-transition module in the L1Calo PreProcessor VME crate
  - Sends the real-time data from the front PreProcessor Module (PPM) to the FEXs (optical), and to the legacy L1Calo processors (electrical)
  - Formats & sends the PPM event data to FELIX & to legacy ROD (optical)

• Use of the SoC:
  Configuration, control and monitoring of the board via VME and the PPM:
  - Config. & control of clock and power modules + configuration of FPGAs
  - Monitor the operating conditions of TREX (temps, voltages, currents)
  - Slow-control communication with the DCS (via Ethernet)
  - No SoC on current prototype, but use commercial module with Zynq Ultrascale+ MPSoC on pre-production modules

• Trenz Electronics Module ZU2CG-1E:
  – Dual-core ARM® Cortex-A53 MPCore up to 1.3GHz
  – 1 GByte of 32-Bit DDR4 SDRAM
  – 128 MByte SPI boot flash + 4 GByte eMMC
  – 260 pin board-to-board connector with 14 MIO, 132 I/O, and 4-lane GTR
  – 1 GbEth PHY, 1 USB 2.0 OTG PHY, etc.

• Operating System = embedded Linux:
  Not yet decided, but Yocto-based project preferred

• Application software:
  OPC UA server for the communication with the DCS
  Service apps (C/C++/Python) to configure, control & monitor the board
**Detector Control System** – monitor and control ATCA back-end electronics:
- Control of shelves via shelf manager SNMP interface (Pigeon Point)
- SoCs for independent and flexible blade interface:
  Monitor board health, power control and reset signals
  No IPMB or shelf manager bottlenecks, no dependency on proprietary software

**Software – OPC UA* based interface:**
- OPC UA server on blade SoC
  Industry standard with many commercial and open source implementations on various platforms
  Run on embedded Linux
  Network integration using Ethernet/TCP
- Build servers with **quasar** framework
  Makes OPC UA transparent to developer
  Reduces sw development (C++, python) significantly by generating code from xml device model and providing useful components
  Build environment for several platforms (x64 Win+Linux, ARM)
  Client software components to ease SCADA or run control integration

---

*Open Platform Communications Unified Architecture
**Quick OPC UA Server generAtion fRamework*
There is a lot of commonality for the hardware:

Almost all projects use or are intending to use
- Xilinx Zynq SoC, based on ARMv7 (32-bit) processor architecture
- Xilinx Zynq UltraScale+ MPSoC, based ARMv8 (64-bit) processor architecture

Advantages of Xilinx Zynq (MP)SoC are:
- Provide an efficient, low-latency way to interface hardware and software:
  - Peripherals, like I2C, SPI, etc.
  - Other FPGAs via Xilinx AXI Chip2Chip Links
- Allows one to run a set of low-level test tools, e.g. Xilinx Virtual Cable (XVC), IBERT test programs, FPGA configuration, etc.
Summary – Software

Commonality for the software?

- **Operating system:**
  - Provided by Xilinx: embedded Linux built using Yocto/meta-xilinx or Petalinux which uses Yocto
  - Linux distribution: ArchLinux, CentOS
  - Real-time operating system: freeRTOS, RTEMS

→ *All are open-source!*

→ *It might help if we can agree on common platform, e.g. CentOS, at least for those that do not need real-time features*

→ *CentOS is gaining speed in the ARM user community, it is being used at CERN’s techlab*

→ *For the Zynq, one could deploy CentOS (or a stripped down version):*
  - *The kernel is hardware-specific* (for the Xilinx Zynq), because of the drivers and the device tree required by Xilinx and/or custom IP cores in the PL and/or required by different Vivado versions
  - *The decoupling is achieved at the level of the system libraries*, i.e. glibc etc.
  - *The application software is built on top of CentOS* (and system libraries)

⇒ *ATLAS AFP ROD (SLAC) have started investigating this approach, ATLAS MUCTPI (CERN) are also interested*
Summary – Software

Commonality for the software?

• User application software – a variety of approaches:
  
  – **IPbus:**
    • Fully decouple implementation (on Zynq) and application (on PC)
    • Runs over a variety of physical media, UDP+ControlHub, TCP, PCIe, SPI
    • Provides a software library and a user-facing hierarchical register map
  
  – **RPC-like approach:**
    • Fully decouple implementation (on Zynq) and application (on PC)
    • A variety of middleware is being used: Google protocol buffers, Remote Call Framework with JSON, simple home-built C++ RPC-like software, ....
  
  – **Port of TDAQ software:**
    • Run control application runs on Zynq
    • Could become much simpler if used with CentOS, and even more so if LCG and TDAQ are built for ARM
  
  – **Run OPC UA server on the SoC**
    • For hardware control, complementary to IPMI → shelf manager → SCADA

⇒*Share our experiences ... and learn from each other*
Summary – Software

• **Generally, having an operating system provides a lot of flexibility:**
  – Makes the SoC look like the SBC or like the PC+VME-bridge in the legacy VME system
  – Possibility to run monitoring and/or calibration applications
  – Possibility to run stand-alone in testbeds
  – Potential to improve long-term maintenance of control and configuration

• **For development and debugging it has already proven to be a big advantage to be able to “log into” the electronics module:**
  – Interactive running of test programs, e.g. Xilinx IBERT eye scans, or I2Cdump, etc.
Summary – Other Issues

Long-term support?
– Large community of embedded systems and commercial interest!
– Linux, gcc, ... are all open-source software!
– We are becoming a non-negligible community of SoC users ourselves ...

System administration aspects?
– Network administration, e.g. assignment of MAC and IP addresses, DHCP, private network, HPM.3 (DUID)?
– Network boot services, e.g. could the SoCs be booted from system administration PCs? How would the boot images be provided, in particular, if using CentOS?
– Kernel patches due to security updates?
⇒ We might need to define new protocols/procedures together with system administrators!
Outlook

I suggest to create an interest group:

“System-on-Chip for Electronics”

⇒ Mailing list: system-on-chip@cern.ch

⇒ Exchange information, suggestions, questions, ...
⇒ Could organise regular meetings (1-2 times per year?) ...
---- BACKUP ----
CMS – CTP7

ZYNQ-based Integrated Eye Scan (IES)

- Eye Scan Capability of IBERT ported to ZYNQ
- Non-invasive scans taken on live operational data (not PRBS test patterns)
- Parallel multi-channel capability for faster results

**Programmable engine can be used for a wide range of applications**
- Quantitative link characterization and parameter optimization
- Operational troubleshooting
- Trend analysis and forecasting for preventive maintenance

**ECAL link @ 4.8 Gbps**

**HCAL link @ 6.4 Gbps**

Eyescans taken at CMS during physics fill with stable beams in the LHC
US CMS ATCA APx Consortium

- Pooling of efforts in ATCA Processor hardware, firmware and software development
- Multiple ATCA processors and mezzanine board types
- Modular design philosophy, emphasis on platform solutions with flexibility and expandability
- Reusable circuit, firmware and software elements
gFEX – ZYNQ MPSoC EG Devices
Block Diagram

Zynq® UltraScale+™ MPSoCs: EG Block Diagram

Processing System
Application Processing Unit
- ARM Cortex™-A53
- NEON™
- Floating Point Unit
- 32KB iCache w/ECC
- 32KB D-Cache w/ECC
- Memory Management Unit
- Embedded Trace Macrocell

Memory
- DDR4/3L LPDDR4/4
- 32/64-Bit w/ECC
- 256KB OCM with ECC
- 64KB L2 Cache

Graphics Processing Unit
- ARM Mali™-400 MP2
- Geometry Processor
- Pixel Processor

High-Speed Connectivity
- DisplayPort v1.2a
- USB 3.0
- SATA 3.1
- PCIe® 3.0 / 2.0
- PS-BTR

General Connectivity
- GigE
- USB 2.0
- CAN
- UART
- SPI
- Quad SPI NOR
- NAND
- SD/eMMC

Real-Time Processing Unit
- ARM Cortex™-R5
- Vector Floating Point Unit
- Memory Protection Unit

Platform Management Unit
- System Management
- Power Management
- Functional Safety

Configuration and Security Unit
- Config AES Encryption, Authentication, Secure Boot
- Voltage/Temp Monitor
- TrustZone

System Functions
- Timers, WDT, Resets, Clocking & Debug

Programmable Logic
- Storage & Signal Processing
  - Block RAM
  - UltraRAM
  - DSP
- System Monitor
  - General-Purpose I/O
  - High-Performance HP I/O
  - High-Density HD I/O
- High-Speed Connectivity
  - Interlaken
  - GTH
  - GTY
  - 100G EMAC
  - PCIe Gen4
Monitoring by 3 sources:
- Onboard power-management chips
- IPMC – onboard & Shelf-manager
- Zynq - MPSoC

Strategy:
- **Power management chips:** Emergency actions
  - Cases: Over-voltage/Current
  - DCS only for intended actions and general monitoring
- **Zynq:**
  - DCS monitoring - using OPC UA through TCP/IP
  - Hardware components monitored: (sensors, FPGAs, etc...)
  - Measured parameters: Voltages, Currents, Temperatures
  - Other: pFPGAs information, health signal, CRC counter
  - **DCS actions:** Board reset or shutdown, miniPODs reset
- **Shelf-manager:**
  - DCS monitoring - through SNMP using fwATCA
  - Fans: status & speed
  - IPMC – status ON/OFF
  - gFEX board status - connection to the chassis and active
  - **DCS actions:** Board shutdown