Preamble:
Use of ARM-Based Servers for HEP Computing

Use of ARM-based servers for experiments' offline computing as alternative or as complement to Intel-based servers

→ Examples:

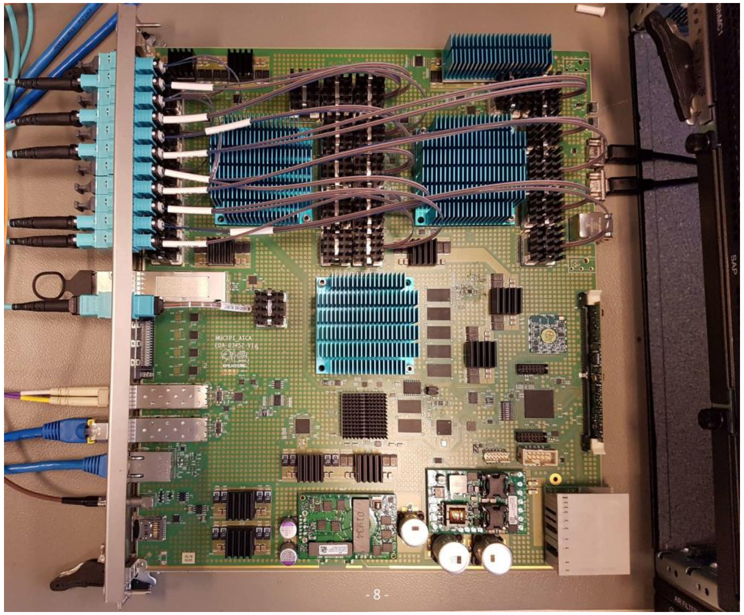
- **ATLAS software study on porting software to ARM64 servers (J. W. Smith):**
  Showed that performance with ARM64 servers can be competitive with traditional Intel machines while the efficiency in terms of power consumption is better
- **CERN's TechLab acquired four Cavium ThunderX2 Servers based on ARM64:**
  → https://cern.ch/techlab
- **University of Bristol is working on a high performance computing project using ARM64 servers: "Isambard" supercomputer**

⇒ Can HEP software be built to run on ARM, or other interesting computing architectures?

Embedded Linux for Run Control

R. Spiwoks, EP-ESE, ATLAS Level-1 Central Trigger Team

Example: ATLAS MUCTPI



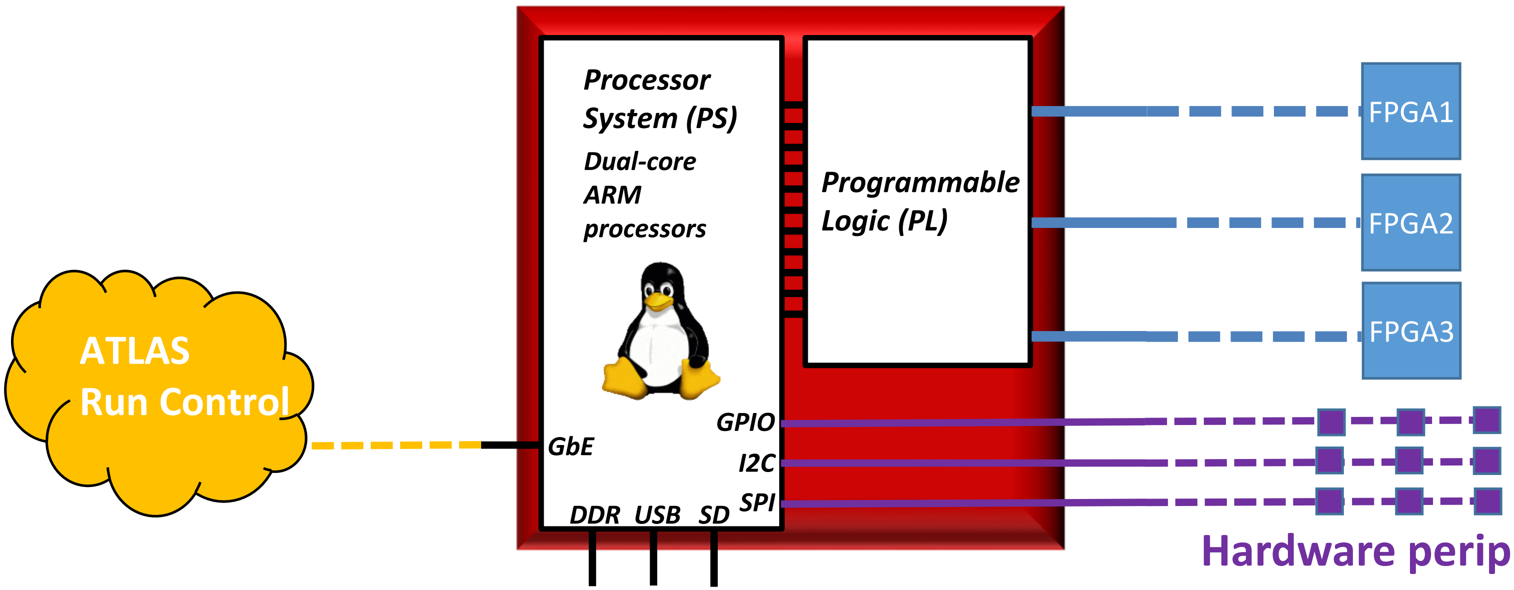
- **ATLAS MUCTPI = “*beautiful new trigger electronics module*”**

  The MUCTPI contains very powerful processing Field-Programmable Gate Arrays (FPGAs)

  The FPGAs process in a parallel and synchronous way the real-time trigger data (optical & electrical): o(100) GByte/s data in, o(10) GByte/s data out

- **How to control and monitor the FPGAs?**
  - **Configure the FPGAs with their firmware, load their look-up tables (depends on physics), monitor how well they are doing (depends on physics), could do some “analysis” on selected events**
- **How to control and monitor the hardware?**
  - **Power supplies, optical transceiver modules, clock chips, etc.**

System-on-Chip (SoC)



- **Xilinx Zynq SoC (ARMv7) or Zynq Ultrascale+ MPSoC (ARMv8):**
  - **Programmable Logic (PL) part:**
    - **Like FPGA, interfaces to the processing FPGAs**
  - **Processor System (PS):**
    - **Based on ARM cores, runs embedded Linux and user application software, interfaces to hardware peripherals, and communicates to the ATLAS run control (Ethernet)**
- **Embedded Linux:**
  - **Prepared using framework of the Linux Foundation's Yocto Project and Xilinx meta layer (Yocto)**
  - **For the user application software we have developed our own meta layer (Yocto)**
  - **We have ported an ATLAS TDAQ run control application (including some WLCG packages) to the SoC**

Other Examples

- **Other ATLAS trigger modules:**
  - L1 Calorimeter Trigger "*gFEX*" module
  - Muon Cathode Strip Chamber Readout Driver "*ROD*" module
- **ATLAS detector control system:**
  - Investigate a replacement for the Embedded Local Monitor Board (ELMB) and run OPC UA server on the processor system for hardware monitoring
- **CLICdp:**
  - Control and Readout Inner tracking Board (CaRIBOu), → A. Fiergolski
- **CMS calorimeter trigger:**
  - Calorimeter Trigger "*CTP7*" module
  - Embedded Linux Mezzanine "*ELM*" as embedded on-module computer
  - Intelligent Platform Management Controller "*IPMC*" for module hardware control

→ Will see more systems like this in the future, cf. Internet of Things

⇒ "push the intelligence into the electronics modules"

Research and Development

- **Build environment for embedded Linux:**
  - Other users of Yocto? Exchange experiences, recipes, code, e.g. drivers etc.?
- **Linux distribution:**
  - Can we use a common Linux distribution on the SoC? E.g. (CERN) CentOS?
    ⇒ Can CERN CentOS be built for a different CPU architecture, e.g. ARM?
      → *side aspect: Use of ARM-Based Servers for HEP Computing?*
- **HEP software:**
  - Can Root be used on the SoC?
      → *We (ATLAS L1CT) have ported ROOT to the SoC; could become part of the distribution)*
    ⇒ Can WLCG software be built for a different CPU architecture, e.g. ARM?
      Can it be cross-compiled?
      Can modularity be improved, i.e. chose more selectively only those packages required?
  - Can we run online physics monitoring on the SoC?
    Can experiment-specific software be built for ARM?
- **Security aspects:**
  - Any differences to other network nodes? Install patches in the running experiment?

⇒ Can we form a working group to provide a common platform for embedded systems?