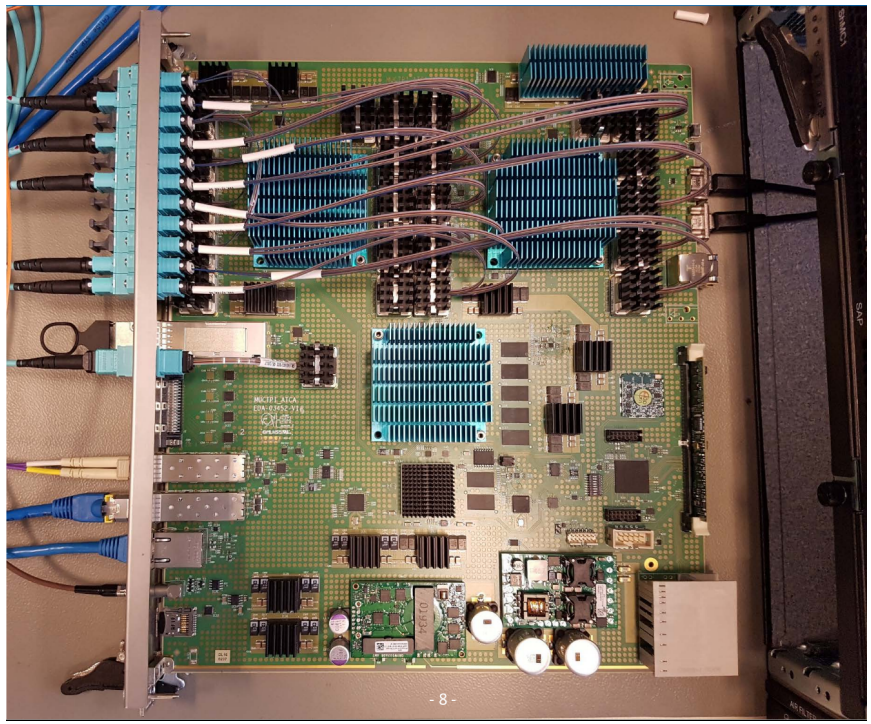
SoC workshop

CentOS on Xilinx Zynq Ultrascale+ MPSoC

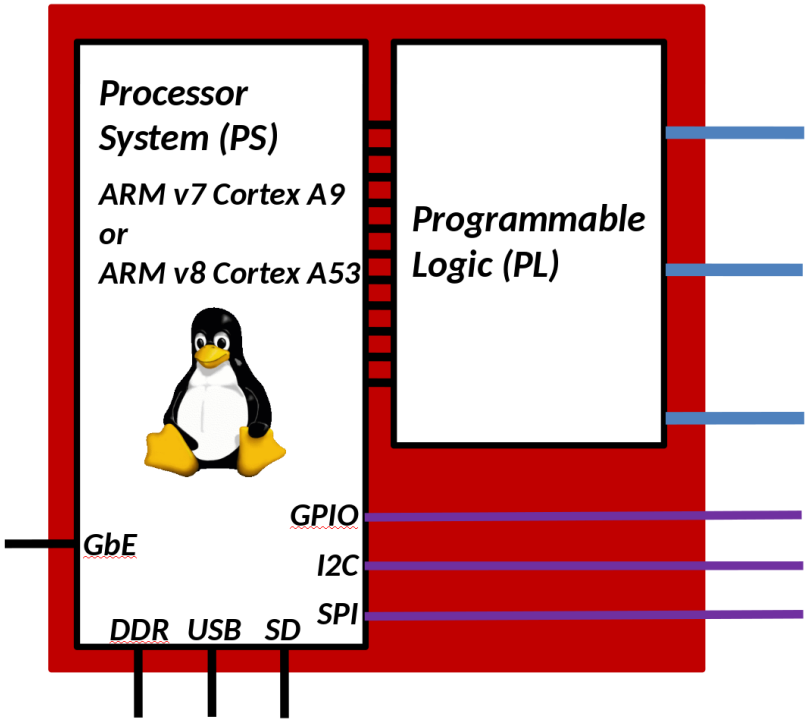
Panagiotis Papageorgiou

MUCTPI

- The Muon to Central Trigger Processor Interface (MUCTPI) is part of the ATLAS Level-1 Trigger System
- Single ATCA blade.
- High-density FPGAs are handling the data.
- We need System on Chip for Control
- See also presentation by R. Spiwoks, on Wednesday afternoon



System on Chip



- An SoC consists of a processor (CPU) part and a programmable logic part (FPGA).
- For the MUCTPI use Xilinx Zynq & Xilinx Zynq Ultrascale+ MPSoC (ZynqMP).

SoC Configuration

We need to provide the following:

- SoC PL Configuration:
  - Bitstream file
  - First-Stage Boot Loader (FSBL)
    - initialize hardware
    - loads bitstream
  - Xilinx/Vivado and Xilinx/SDK are used to prepare the configuration files and also provide the Device tree used with the kernel.
- SoC PS Booting:
  - a way to boot the PS,i.e. a bootloader.
  - an Operating System
    - Linux kernel
    - Device tree.
    - a root file system.

SoC Configuration

- The Linux Foundation Yocto project and the Xilinx meta layer provide:
  - A toolchain (cross-compiler and system libraries)
  - bootloader (U-Boot)
  - Linux kernel
  - Linux root filesystem
- Building of user application software can be achieved in the CMake environment using the filesystem and the toolchain produced by Yocto for Cross-Compilation.

Yocto

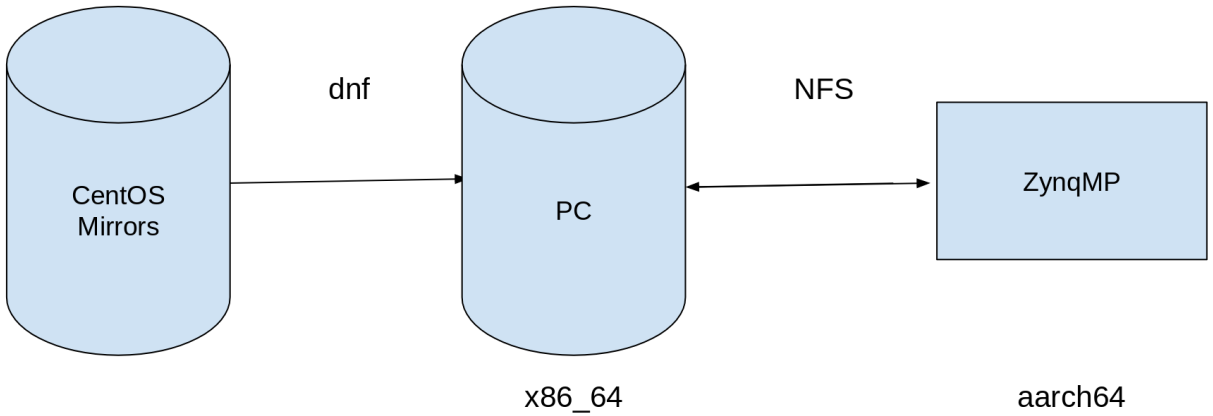
- Yocto and cross-compilation of user software work fine, but...
  - Even though each project could be using the same type of SoC and struggling with the same problems, it could end up with its own unique operating system. Each of them will be a non standard OS and thus will not be allowed into a CERN technical control network, e.g. ATLAS experiment, due to security requirements by CERN/IT and sysadmins.
  - We need a more standardized solution. So that we can all contribute towards the same goal, and have long term support.

Solution: CentOS/aarch64

A possible solution: Build a CentOS/aarch64 root file system and run it over a hardware-specific kernel (built by Yocto):

- CentOS is interesting because CERN supports its x86_64 version (CERN CentOS).
- Use dnf, the successor of yum
  - Original idea by Matthias Wittgen of SLAC, see his demo after this one, and also his presentation this afternoon.
- dnf has the `--forcearch` option that can cross install software to a different architecture filesystem. We use dnf on the host PC, to install CentOS/aarch64 packages.This is done by dnf using qemu.
- Script installs a minimal but fully functional Centos/aarch64 filesystem. Link here: https://twiki.cern.ch/twiki/bin/view/SystemOnChip/CentOSForZynqMP

Setup



- MUCTPI is being used so we are using a Xilinx Zynq UltraScale+ MPSoC ZCU102 evaluation board. It has the same chip and its less complicated to be brought for the demo.
- Host PC (x86_64) running CERN CentOS 7
- They are connected via a private network.
- The host is also connected to the internet.
- bitstream, bootloader and kernel are loaded via an SD-card.
- root filesystem mounted via NFS on ZynqMP from the host PC.

Results

We run such a system successfully on the SoC.

- The CentOS/aarch64 filesystem provides among others:
  - any software that is expected to be part of the CentOS distribution, e.g. ssh, NFS, NTP, iptables, etc.
- Is no different from any other PC running Linux. Can be used to install more software (e.g. nautilus).

Additional Software

Has been used to compile natively on the ZynqMP(ZCU102 evaluation board) because the build instructions for those packages where the same as in x86_64 builds. (e.g. make, cmake, autoconf)

- GCC version 8.2, because stock CentOS uses a really old one.
- ROOT
- a subset of ATLAS-TDAQ (∼50 TDAQ packages) necessary to run a RunControl application using ROOT.
- part of the ALTAS DCS software for hardware control.
  - see tutorial and presentation of S.Schlenker and P.Nikiel on Friday.

Additional Software

As mentioned before, we compile and install application software natively on the ZynqMP. In addition to install software packages we also have the following options:

- Use dnf on the host PC install from official CentOS repositories.
- Use yum on the ZynqMP, if connected to the internet.
- Cross-compile on host PC using the generated CentOS file system.

Results

Providing a CentOS filesystem was relatively easy and building natively on CentOS/aarch64 was straightforward.

- Most of the dependencies were installed from CentOS repositories.
  - Some dependencies were compiled natively to get a more recent version.

Results

- Building time can be sped up considerably using arm servers. Compiled software on a thunderx-01 arm server of CERN Techlab and then deployed on the ZynqMP.
  - ZynqMP 4 Cores@1.5GHz & 4GB RAM
    vs
    Thunderx-01 arm server 96 Cores @2.5GHz & 256GB RAM
  - Compiling GCC took 5h on the ZynqMP and 50m on the server.

CentOS/armv7hl

- A CentOS 32-bit ARMv7 is also available
- Required for the Zynq which is used by us for current prototype version of MUCTPI, and also by other projects who do not use the ZynqMP. Link below: https://twiki.cern.ch/twiki/bin/view/SystemOnChip/CentOSForZynqMP
- It is unclear if CentOS/armv7hl will have a long future

Common Platform: CentOS/aarch64

- CentOS/x86_64 is a de facto standard at CERN.
- CentOS/aarch64 is released by CentOS.org at the same time as CentOS/x86_64.
- Could be a platform for common software development:
  - Tools, e.g. Xilinx Virtual Cable, FPGA configuration
  - Drivers and libraries, e.g. DMA and contiguous memory segments
  - Compilers, e.g. gcc8
  - Eventually probably also TDAQ software
- Could eventually be maintained by system administrators and/or CERN-IT.
  - We intend to investigate the use of puppet for host configuration.
- Could possibly be allowed on a technical control network.

Questions?