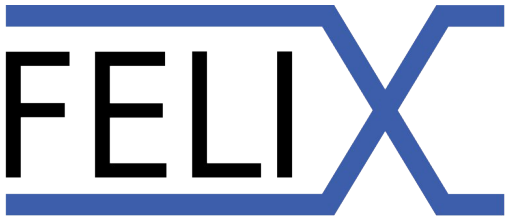
FELIX Phase II Run 4

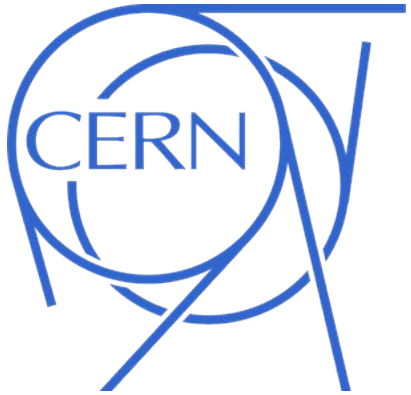
Based on a Xilinx Versal Prime ACAP Device

3rd CERN System-on-Chip Workshop

05-10-2023

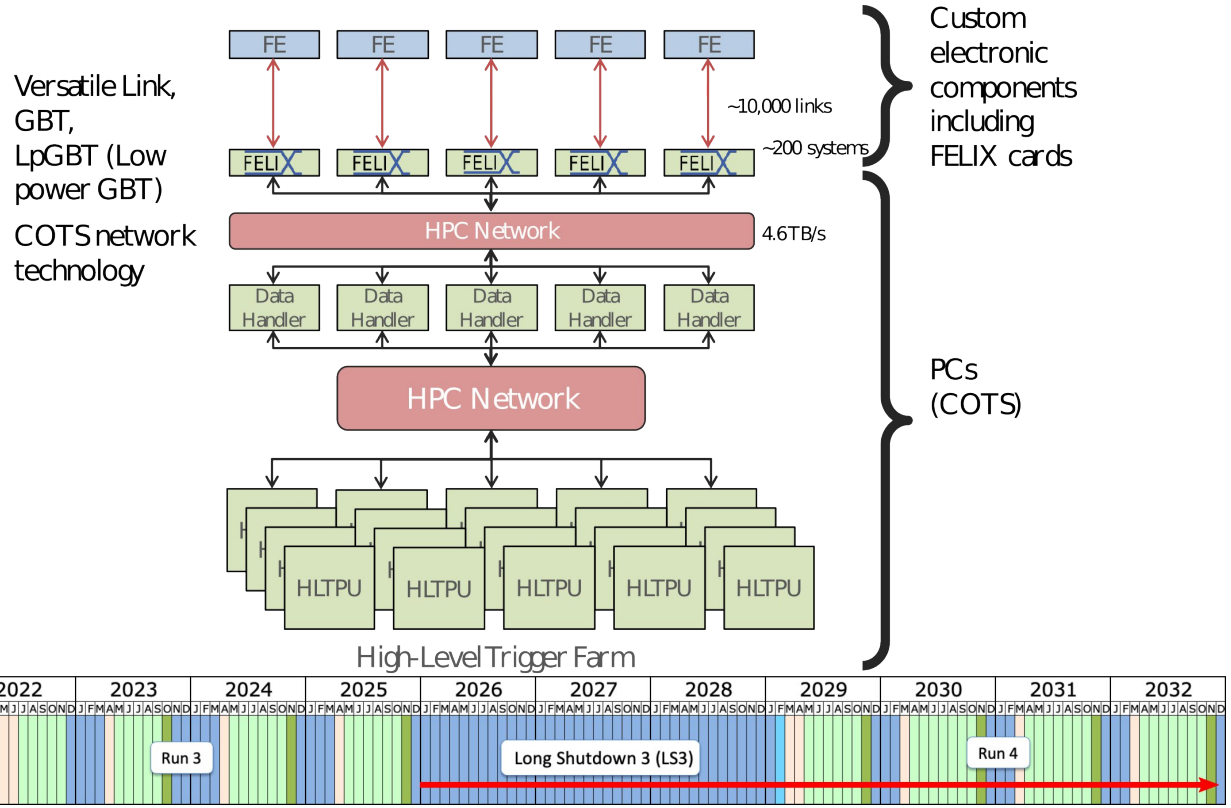
Nayib Boukadida





FELIX Phase II run 4 overview

- FELIX used by all subdetector systems
- ~10x higher trigger rate (1 MHz)
- ~20x higher data readout rate (4.6 TB/s)
- ~3x higher mean number of interactions per bunch crossing (200)



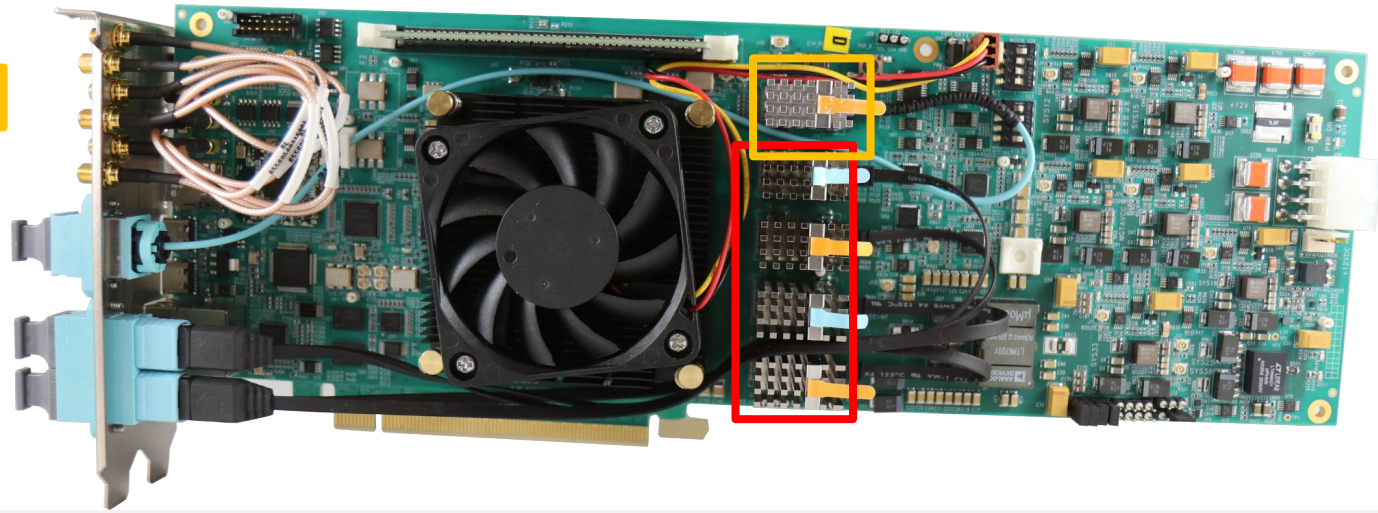
FELIX Phase II run 4 hardware

FLX-182 development card

- Based on a Xilinx Versal Prime VM1802
- 4 Samtec FireFly transceivers
  - 24 bidirectional optical links
  - 25 Gbps bandwidth per channel
- 1 Samtec FireFly for LTI/TTC link
  - Local Trigger Interface
  - Trigger, Timing and Control
- PCIe Gen4 x16 (240 Gbps)
  - 2 x8 lanes bifurcated

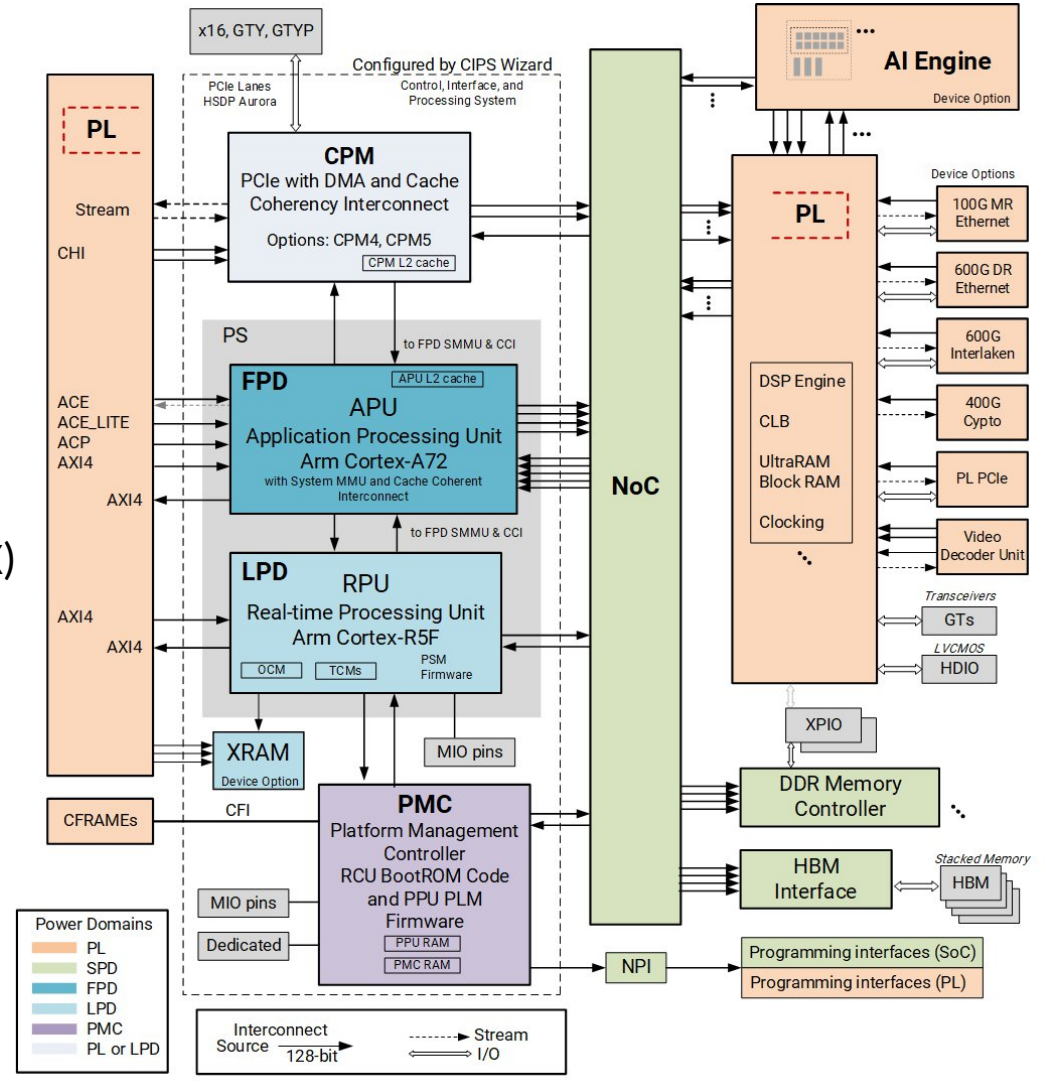
Server hardware

- AMD Epyc 9004 (Genoa)
- 96 GB DDR5
- 2x 200 Gbps Ethernet on PCIe Gen5



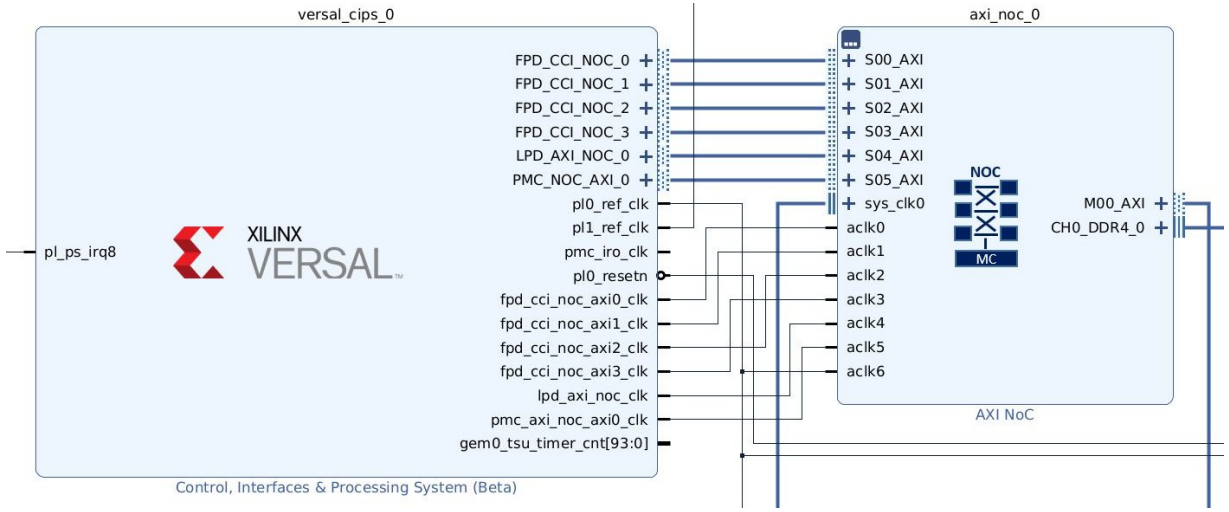
Versal Prime

- It is an Adaptive Compute Acceleration Platform (ACAP)
- Processing System (PS)
  - Dual-core ARM Cortex-A72 Application Processing Unit
  - Dual-core ARM Cortex-R5F Real-Time Processing Unit
- AI Engine
- CPM PCIe controllers capable of Cache Coherent Interconnect (CCIX)
- Programmable logic (PL)
- System started up by PMC (Platform Management Controller)
- Interconnections via NoC (Network on Chip)



Instantiating the CIPS

- Control, Interfaces & Processing System
- Starts with instantiating the CIPS in the PL design
  - Requires a Block Design in your project
- NoC used for AXI interconnection and DDR memory access
- Useful to include in Versal PL design
  - Even when not using the PS
  - NoC functionality
  - QSPI usage
  - Debug cores

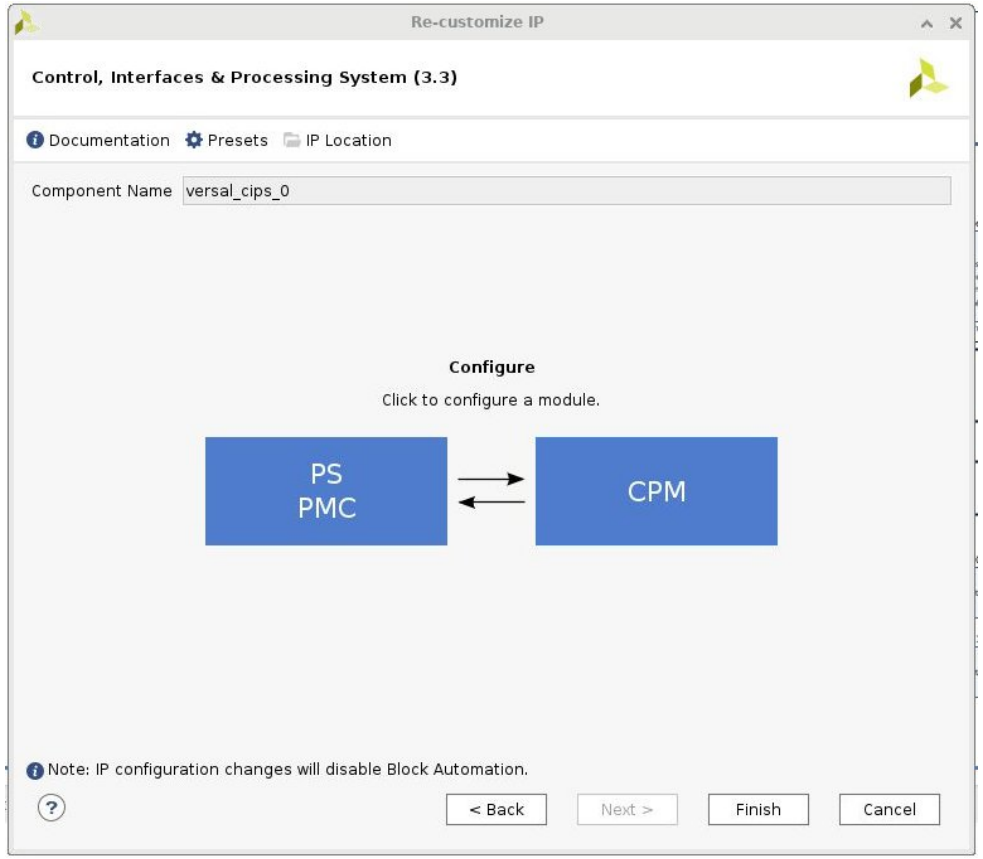


Configuring the CIPS

- Provides two modules to configure

- PS PMC (Platform Management Controller)
  - Boot and configuration of the PS
  - I/O peripherals
  - PS PL interfaces
  - NoC register initialization settings
  - Power management
  - Interrupts

- CPM (CCIX and PCIe Module)
  - Can be used for PCIe Gen4/5 connectivity (depends on the physical connection to the PCIe pins)
  - Operational quickly after boot, without the need to configure the PL



Why use the Processing System

- FELIX data flow is exclusively handled by the PL
- To provide control and insight in what our device is doing
- Test and verify correct behavior of implemented functionality
  - Such as transceivers and memory
- Monitor sensor data (through I2C, SPI etc.)
  - Read temperature and monitor power rails
- Update firmware
  - Let the PS update your firmware

Building PetaLinux for the FLX182

- Using PetaLinux v2022.2
- Docker image provided
  - Based on Ubuntu 20.04
  - Clean build environment with all required packages prepared
- Project is ready to build
- Can be written to a SD card or QSPI flash

https://gitlab.cern.ch/atlas-tdaq-felix/felix-versal-tools/flx-petalinux-2022.2

```
petalinux@e335d79a203f:~/build$ petalinux-build
[INFO] Sourcing buildtools
[INFO] Building project
[INFO] Generating Kconfig for project
[INFO] Silentconfig project
[INFO] Extracting yocto SDK to components/yocto. This may take time!
[INFO] Sourcing build environment
[INFO] Generating kconfig for Rootfs
[INFO] Silentconfig rootfs
[INFO] Generating plnxtool conf
[INFO] Adding user layers
[INFO] Generating workspace directory
INFO: bitbake petalinux-image-minimal
NOTE: Started PRServer with DBfile: /home/petalinux/build/build/cache/prserv.sqlite3, Address: 127.0.0.1:38339, P
ID: 2622
Loading cache: 100% |                                                | ETA:  --:--:--
Loaded 0 entries from dependency cache.
Parsing recipes: 100% |#########################################| Time: 0:02:01
Parsing of 4472 .bb files complete (0 cached, 4472 parsed). 6508 targets, 592 skipped, 1 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies
NOTE: Fetching uninative binary shim file:///home/petalinux/build/components/yocto/downloads/uninative/126f4f7f6f
21084ee140dac3eb4c536b963837826b7c38599db0b512c3377ba2/x86_64-nativesdk-libc-3.4.tar.xz;sha256sum=126f4f7f6f21084
ee140dac3eb4c536b963837826b7c38599db0b512c3377ba2 (will check PREMIRRORS first)
Initialising tasks: 100% |#########################################| Time: 0:00:11
Checking sstate mirror object availability: 100% |###############| Time: 0:00:54
Sstate summary: Wanted 2678 Local 0 Network 2130 Missed 548 Current 0 (79% match, 0% complete)
NOTE: Executing Tasks
NOTE: Tasks Summary: Attempted 6767 tasks of which 5645 didn't need to be rerun and all succeeded.
INFO: Failed to copy built images to tftp dir: /tftpboot
[INFO] Successfully built project
```

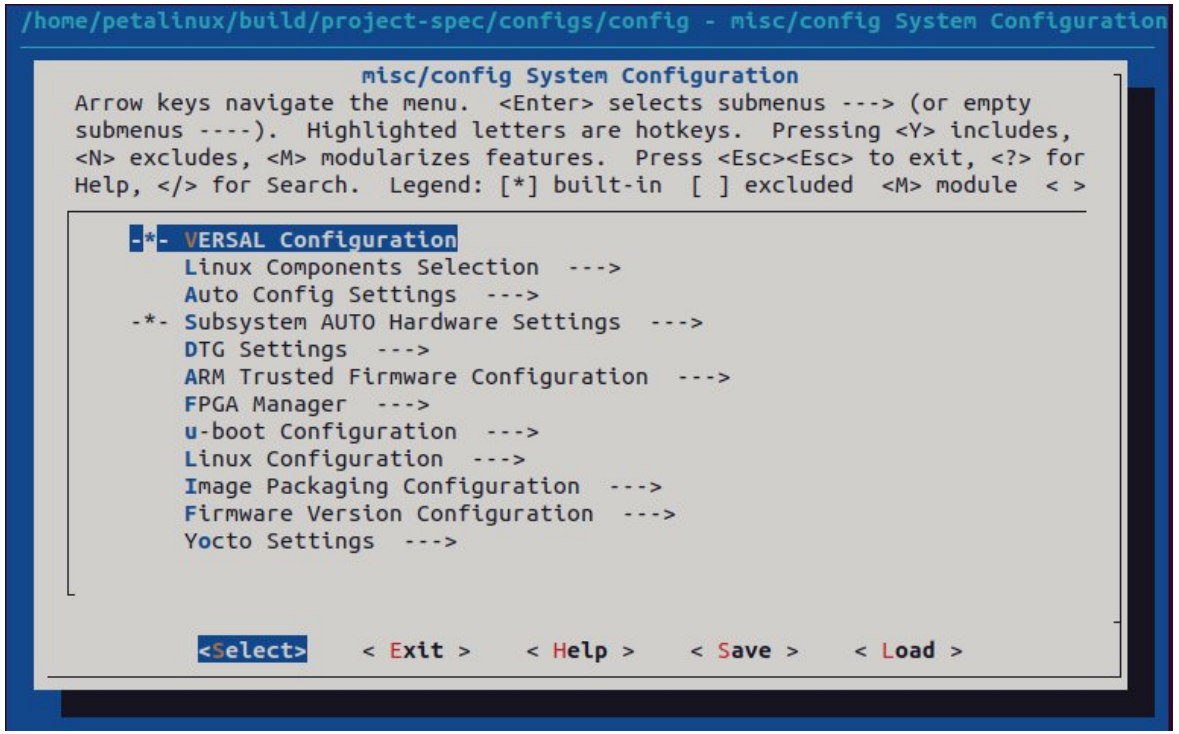
```
petalinux@e335d79a203f:~/build$ petalinux-package --boot --u-boot --force
[INFO] Sourcing buildtools
INFO: Getting system flash information...
INFO: File in BOOT BIN: "/home/petalinux/build/project-spec/hw-description/Versal_top_wrapper.pdi"
INFO: File in BOOT BIN: "/home/petalinux/build/images/linux/plm.elf"
INFO: File in BOOT BIN: "/home/petalinux/build/images/linux/psmfw.elf"
INFO: File in BOOT BIN: "/home/petalinux/build/images/linux/system.dtb"
INFO: File in BOOT BIN: "/home/petalinux/build/images/linux/bl31.elf"
INFO: File in BOOT BIN: "/home/petalinux/build/images/linux/u-boot.elf"
INFO: Generating versal binary package BOOT.BIN...


****** Xilinx Bootgen v2022.2
  **** Build date : Sep 26 2022-06:24:42
    ** Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.


[INFO]   : Bootimage generated successfully

INFO: Generating QEMU boot images...
INFO:  File in qemu_boot.img: /home/petalinux/build/images/linux/BOOT.BIN
INFO:  File in qemu_boot.img: /home/petalinux/build/images/linux/rootfs.cpio.gz.u-boot
INFO:  File in qemu_boot.img: /home/petalinux/build/images/linux/boot.scr
INFO: Binary is ready.
WARNING: Unable to access the TFTPBOOT folder /tftpboot!!!
WARNING: Skip file copy to TFTPBOOT folder!!!
```

Configuring Petalinux

- Provided instance is already pre-configured

- However it can be configured to your liking
  - By running the ‘petalinux-config’ command

```
petalinux@ffd5b0c3ecad:~/build$ petalinux-config
[INFO] Sourcing buildtools
[INFO] Menuconfig project


*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

[INFO] Sourcing build environment
[INFO] Generating kconfig for Rootfs
[INFO] Silentconfig rootfs
[INFO] Generating plnxtool conf
[INFO] Generating workspace directory
[INFO] Successfully configured project
```



Device tree & custom application

- Linux has to be informed about included devices

- For the FLX-182 all on-board chips and controllers have been defined in the device tree
  - project-spec/meta-user/recipes-bsp/device-tree/files/system-user.dtsi

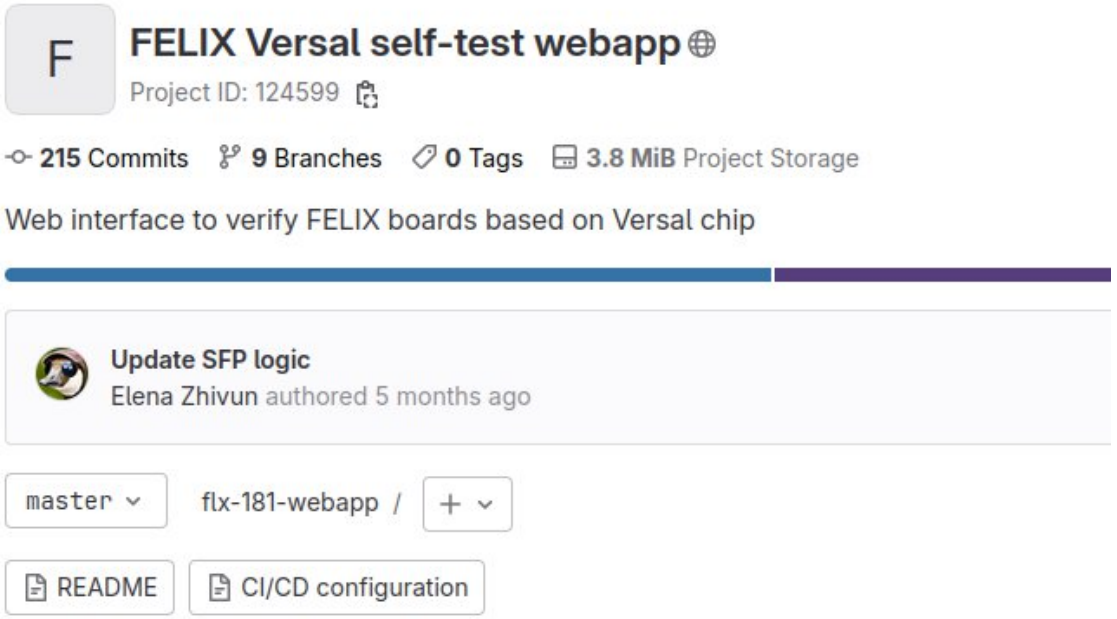
- It's possible to add your own custom application to the PetaLinux files
  - project-spec/meta-user/recipes-apps/

```
&gem0 { /* PMC_MIO_48, LPD_MIO0-11/24/25 */
        phy-handle = <&phy1>; /* u198 */
        phy-mode = "rgmii-id";
        local-mac-address = [ 00 0A 35 07 DE B2 ];
        mdio: mdio {
                #address-cells = <1>;
                #size-cells = <0>;

                phy1: ethernet-phy@1 {
                        #phy-cells = <1>;
                        compatible = "ethernet-phy-id2000.a231";
                        reg = <1>;
                        ti,rx-internal-delay = <0xb>;
                        ti,tx-internal-delay = <0xa>;
                        ti,fifo-depth = <1>;
                        ti,dp83867-rxctrl-strap-quirk;
                        reset-assert-us = <100>;
                        reset-deassert-us = <280>;
                        reset-gpios = <&gpio1 48 GPIO_ACTIVE_LOW>;
                };

        };
};
```

https://github.com/Xilinx/linux-xlnx/tree/master/Documentation/devicetree/bindings

FELIX Versal webapp

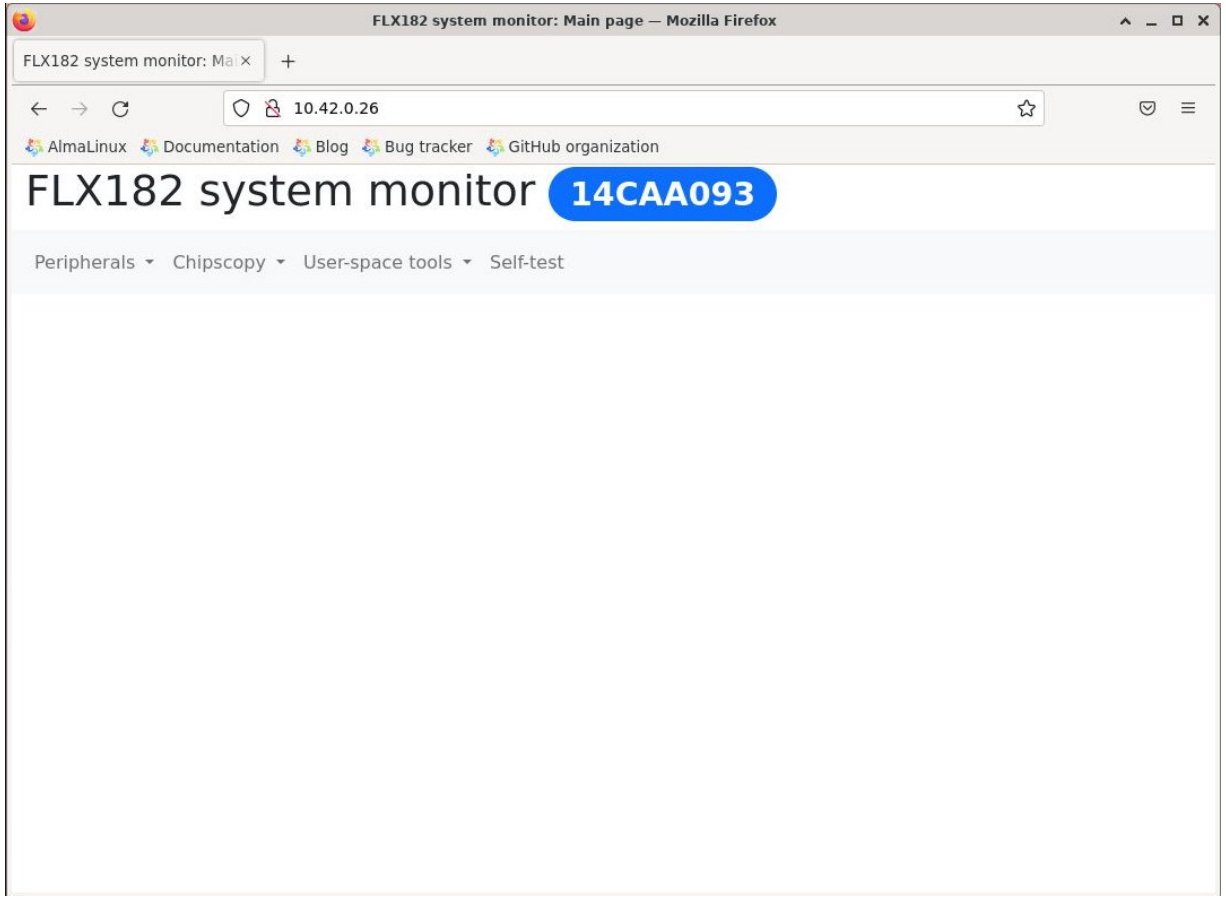
- Written in Python & Open-source on CERN gitlab
- Accompanied with good documentation
- Runs on the embedded platform or the development machine
  - You can even run the app standalone to get familiar with it
- Runs on the Versal Processing System
- Built-in self test (BIST) has been developed for the FLX-182
- Useful for testing and monitoring all peripherals on the card
  - Test transceiver links by generating an eye diagram (FireFly of PCIe links)
  - Supports I2C communication, check for faulty bus or power monitoring
  - A test report can be generated and automatically published in a database, useful to archive data



https://gitlab.cern.ch/atlas-tdaq-felix/felix-versal-tools/flx-181-webapp

Accessing the webapp

- Runs on embedded platform
- Navigate to the IP of the SoC in a preferred web-browser
- There is no IP assignment in our configuration, so you have to look up the IP
- Number in the blue field is the connected board IDCODE



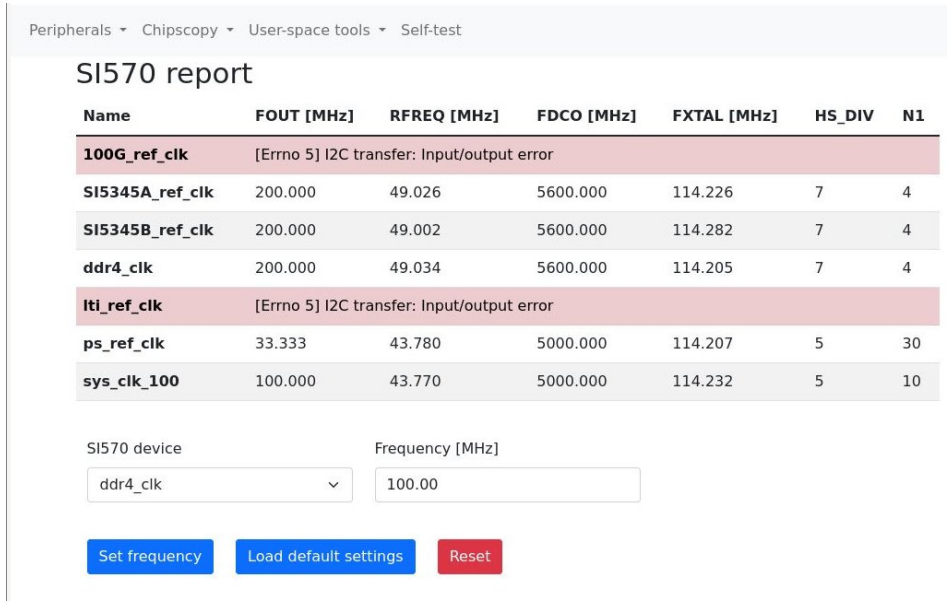
Peripherals

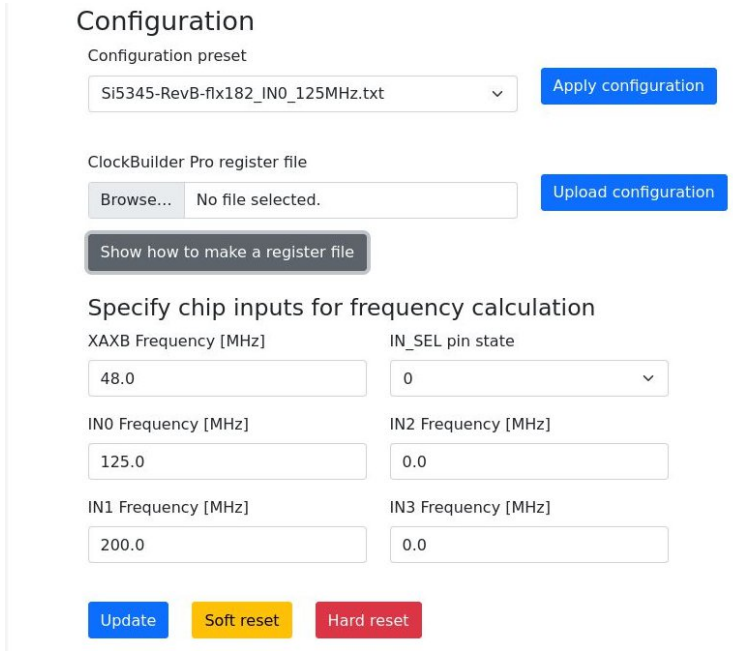
- Reading and configuring of peripherals on the FLX-182 board
- Monitoring voltages and temperatures
- Read and write GPIO
- Configure and verify clock related IC’s
- Read SFP, FireFly modules
- Access the I2C bus

Configuring clock chips

- All clock chips can be configured by the webapp

- Upload configuration file for more complex clock chips
  - SiLabs ClockBuilder Pro register file

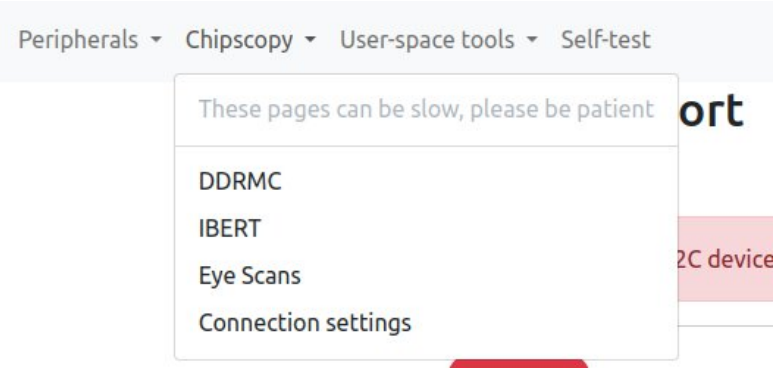


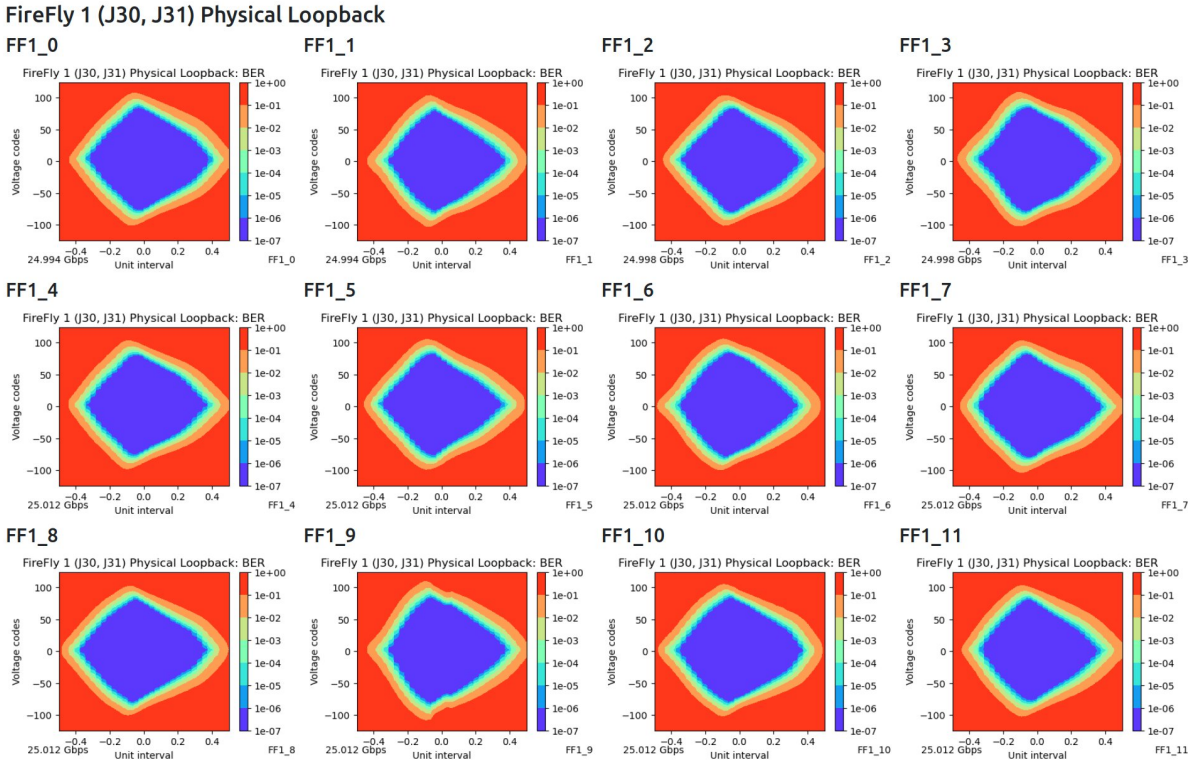


Chipscopy

- Support multiple tests
  - DDRMC (DDR Memory Controller)
  - IBERT (Transceiver Integrated Bit Error Ratio Tester)
  - Eye Scans (See how transceiver data propagates over a link)

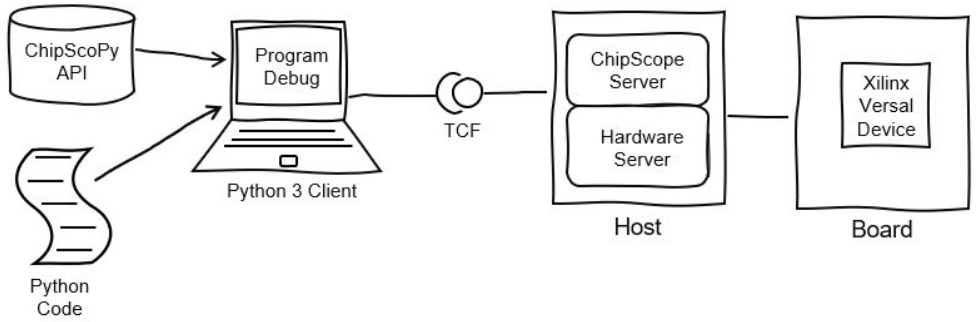
- Result of an eye diagram test

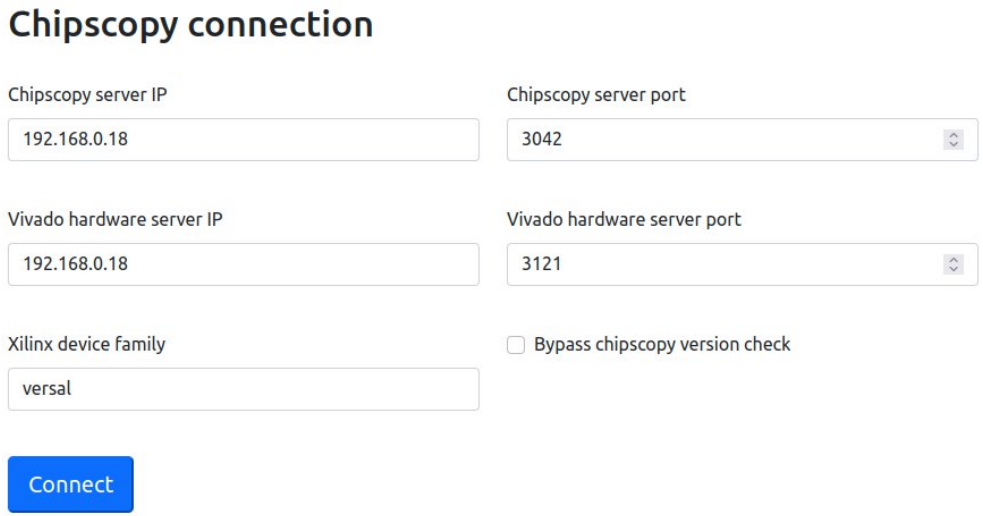




Chipscopy

- Tests which require Xilinx software and a JTAG link to access the hardware
- Uses the Xilinx ChipScoPy
  - Works only with Versal devices
- Connection ChipScoPy and Vivado HW server has to be configured

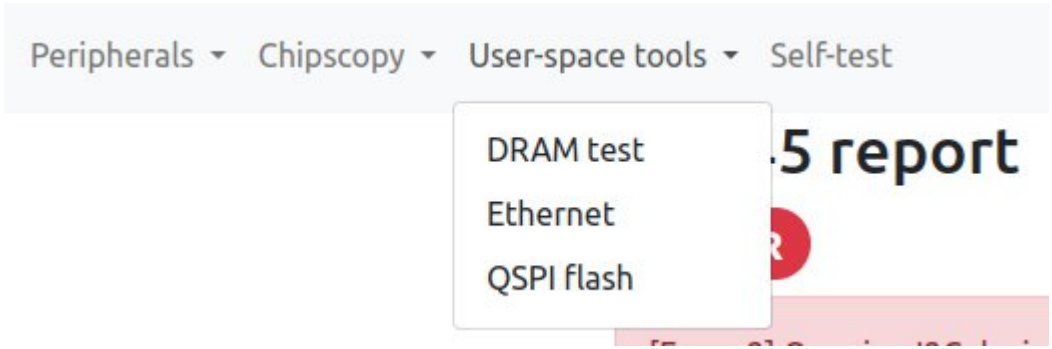


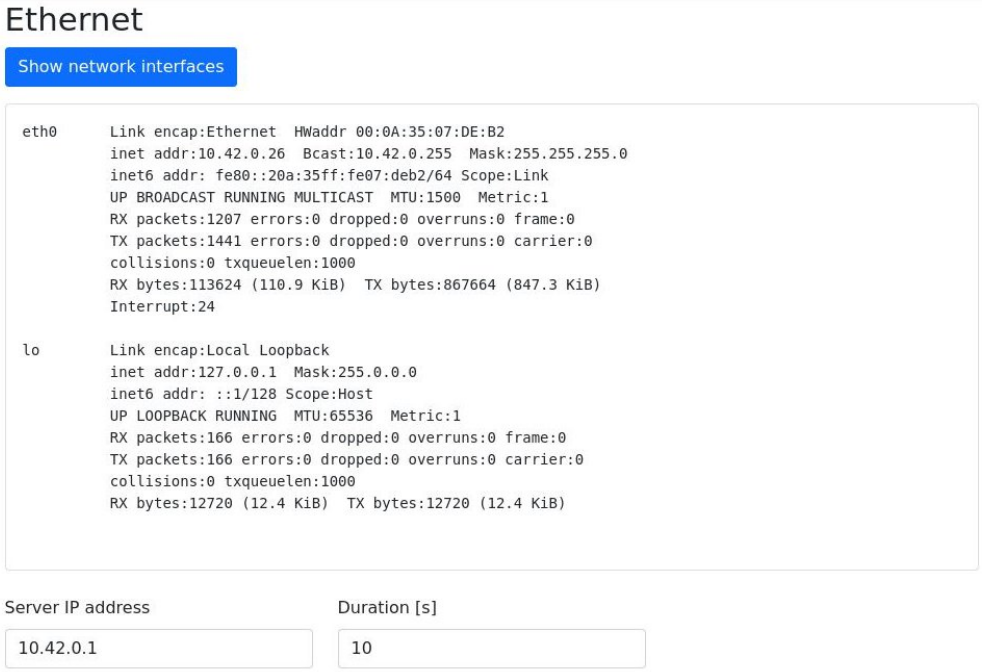


https://github.com/Xilinx/chipscopy

User space tools

- Testing of DRAM, Ethernet and QSPI flash
- Useful to verify correct implementation of Ethernet and DRAM
- Contains an iperf3 test to measure Ethernet performance





Built-In SelfTest

- Verify the functionality of the board
- Developer can select the desired tests to be run
- Test results are written to a .json file
  - Can be download and stored

Most recent test results **Download**

Test description	Result
Identify the board	Test results are missing
SI570 device readout	Error reading out lti_ref_clk, 100G_ref_clk
SFP and FireFly module readout	Readout completed, self-test criteria are not defined
HWMON device readout (SYSMON, TMP435, etc)	Readout completed, self-test criteria are not defined
INA226 chip readout	Readout completed, self-test criteria are not defined
GPIO readout	Readout completed
ADM106x chip readout	Readout completed, self-test criteria are not defined
Probe SI53156	Error reading out PCIe clk buffer (U37)

Board self-test summary

- Status: Done
- Completed: 8
- Incomplete: 0
- Skipped: 0
- Total: 8
- Start time: 2106-02-07T08:46:28.189847
- End time: 2106-02-07T08:46:28.592401
- Time elapsed: 0:00:00.402554
- Download full report

Please select the tests to run

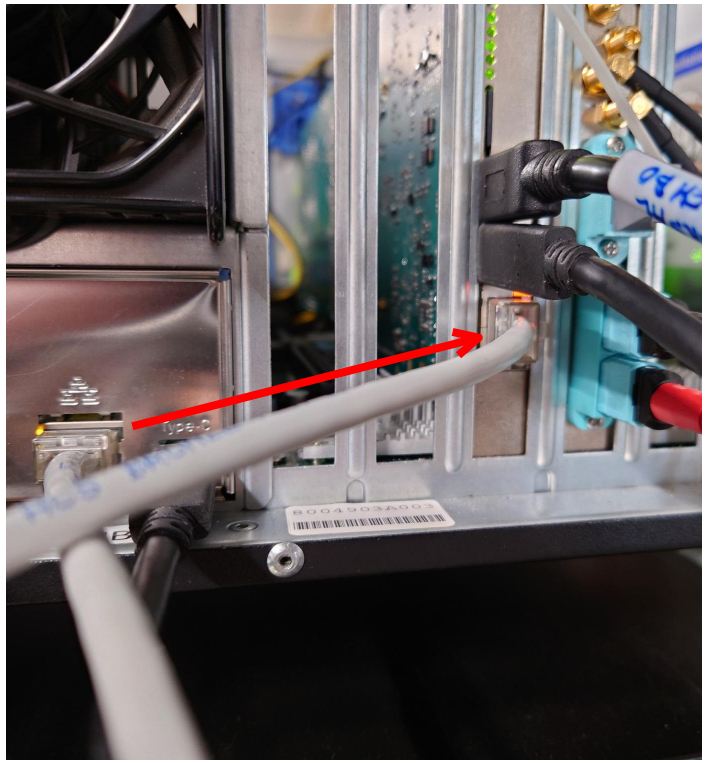
	Test description
☑	Identify the board
☑	SI570 device readout
☑	SFP and FireFly module readout
☑	HWMON device readout (SYSMON, TMP435, etc)
☑	INA226 chip readout
☑	GPIO readout
☑	ADM106x chip readout
☑	Probe SI53156

Start

Network connection

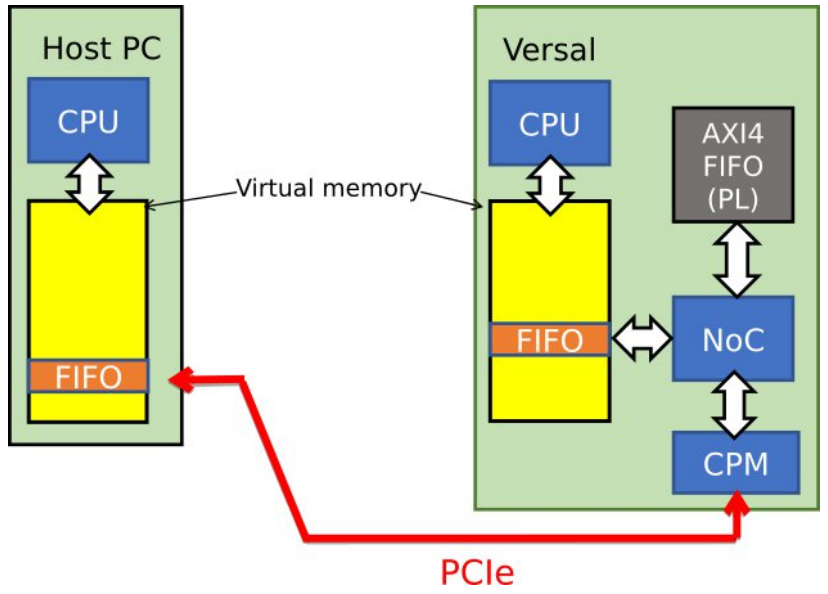
- Possible through physical Ethernet connection
  - FLX-182 provides a 1 GbE connector
  - This is a direct connection to the PS
  - Currently in use

- Using a virtual network connection over PCIe
  - Tunnel network traffic over the PCIe bus
  - Host PC sees a network card
  - No external network/cable required



Virtual network connection

- Using the PCIe bus to transfer network data
- The Host PC will see a new network device
- Network drivers for Linux developed
  - Consists of .c and .h files
  - Have to be compiled to kernel objects (.ko)
  - Makefile included
- PCIe communication can be done using the CPM or Wupper



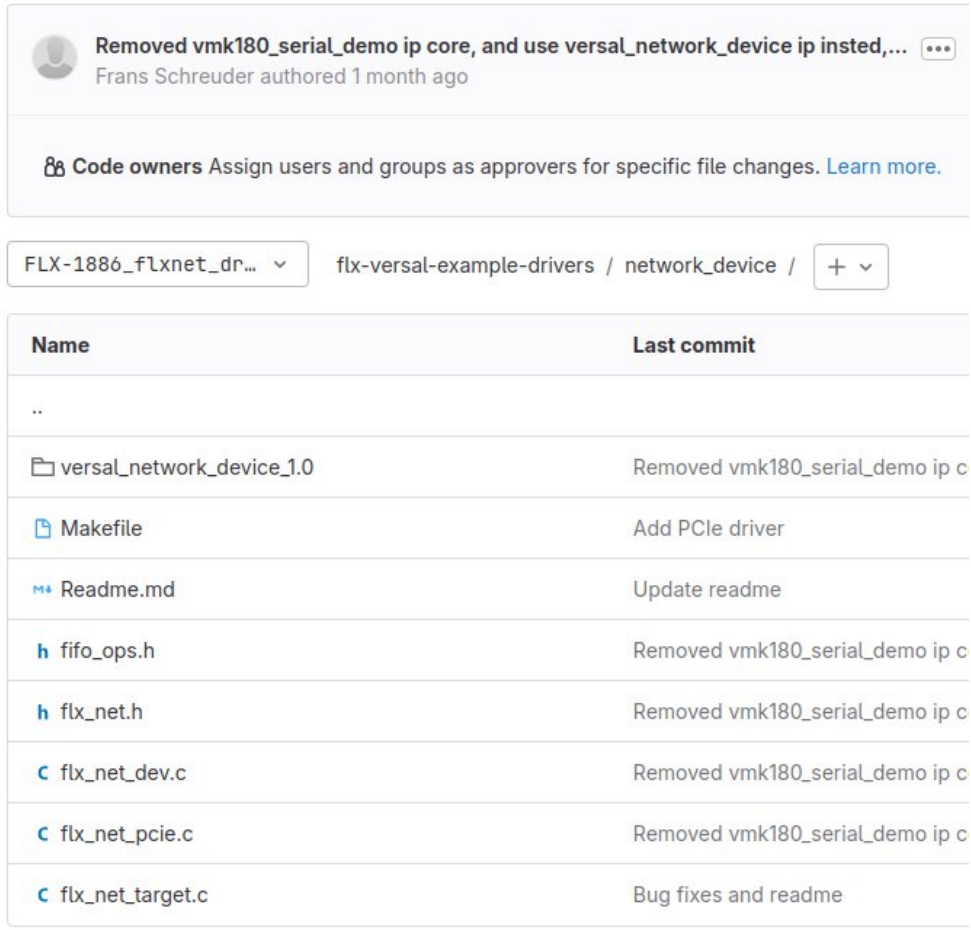
https://gitlab.cern.ch/atlas-tdaq-felix/firmware (Wupper is a part of FELIX)

Network drivers

- Linux drivers developed to establish a communication channel between the Versal PS and host PC over PCIe

- **flxnet_dev.ko (host PC & Versal)**
  - Network interface representation for the FIFO's
- **flxnet_target.ko (Versal)**
  - Obtains IP information from the device tree, maps it into Versal memory and registers it with flxnet_dev.ko
- **flxnet_pcie.ko (host PC)**
  - Finds the correct PCIe devices, maps the AXI Bridge BAR into host memory, registers it with flxnet_dev.ko

- Mapped to PCIe BAR
  - BAR0 for master branch using CPM
  - BAR3 for FLX-1886 branch using Wupper



https://gitlab.cern.ch/atlas-tdaq-felix/felix-versal-tools/flx-versal-example-drivers

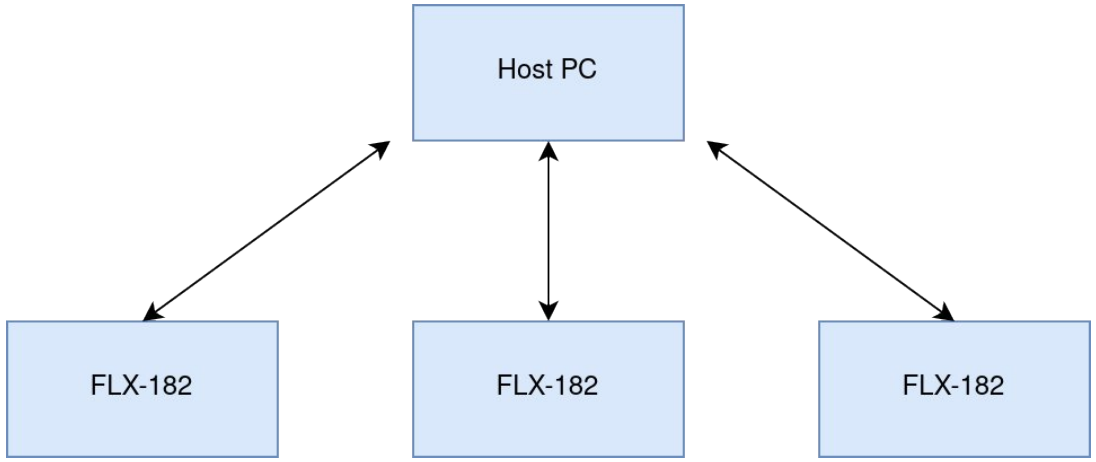
Using the drivers

- Insert modules into the kernel
- Stop network manager to assign static IP
- flxnet0 will be seen as a new network interface
  - Despite it being a virtual network interface over the PCIe bus

```
[felix@localhost ~]$ sudo insmod flxnet_dev.ko
[felix@localhost ~]$ sudo insmod flxnet_pcie.ko
[felix@localhost ~]$ sudo systemctl stop NetworkManager
[felix@localhost ~]$ sudo ifconfig flxnet0 192.168.10.3
[felix@localhost ~]$ ifconfig flxnet0
flxnet0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.10.3  netmask 255.255.255.0  broadcast 192.168.10.255
        inet6 fe80::bf04:4fca:6e7c:6b88  prefixlen 64  scopeid 0x20<link>
        ether 22:36:61:3d:94:97  txqueuelen 100  (Ethernet)
        RX packets 21386  bytes 1533756 (1.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 63099  bytes 95277048 (90.8 MiB)
        TX errors 0  dropped 1248 overruns 0  carrier 0  collisions 0
```

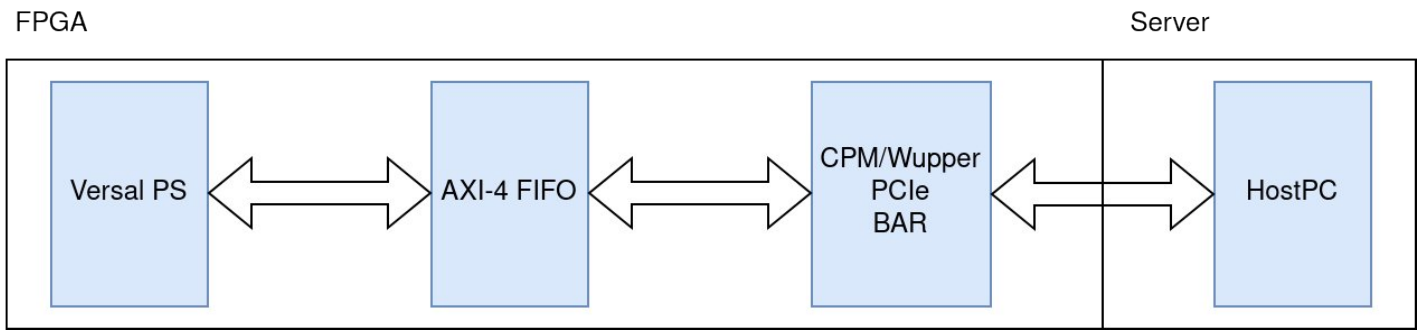
Network structure

- Each Versal card has a static MAC
  - Using its VERSION & IDCODE values
  - Unique for each chip
- Host PC gets a random MAC assigned
  - Can reach all Versal cards
- Versal cards and HostPC form a network
  - Multiple Versal cards can be on the same network
  - Other devices can also join this network



Exchanging network data

- An AXI-4 FIFO is created
  - Included in git repository as Xilinx IP to be instantiated
  - Device tree module provided in the repository
- This FIFO will be treated as a network interface
  - The OS will read/write the FIFO to receive/transmit network data
- Accessible for the PS and through PCIe for the Host PC
  - Reachable through using the NoC
- Mapped to PCIe BAR



Updating hardware description

- No need to completely rebuild petalinux
  - Only a kernel rebuild is required
  - This only works if you have build petalinux before
- Generate the .xsa file
  - In Vivado : file -> export -> export hardware
- Update the hardware description
  - petalinux-config --silentconfig --get-hw-description new_xsa_file.xsa
- Kernel can be rebuild using
  - petalinux-build -c kernel
  - petalinux-package --boot --u-boot --force
  - petalinux-package --wic

Updating firmware

- Firmware can be updated while PetaLinux is running
- Copy over the firmware files with SCP
- No JTAG/USB link required, just an available Ethernet connection

```
[felix@localhost ~]$ scp -r root@192.168.10.2:/media/sd-mmcblk0p1 ./boot
root@192.168.10.2's password:
BOOT.BIN                                                    100%  960KB   1.4MB/s   00:00
boot.scr                                                    100% 2594   145.6KB/s   00:00
image.ub                                                    100%   14MB   1.5MB/s   00:09

[felix@localhost ~]$ ls -la ./boot/
total 18100
drwxr-xr-x.  4 felix felix       85 Jun 29 17:34 .
drwx------. 25 felix felix     4096 Jun 29 17:34 ..
-rwxr-xr-x.  1 felix felix  4121600 Jun 29 17:34 BOOT.BIN
-rwxr-xr-x.  1 felix felix     2594 Jun 29 17:34 boot.scr
-rwxr-xr-x.  1 felix felix 14400404 Jun 29 17:34 image.ub
```

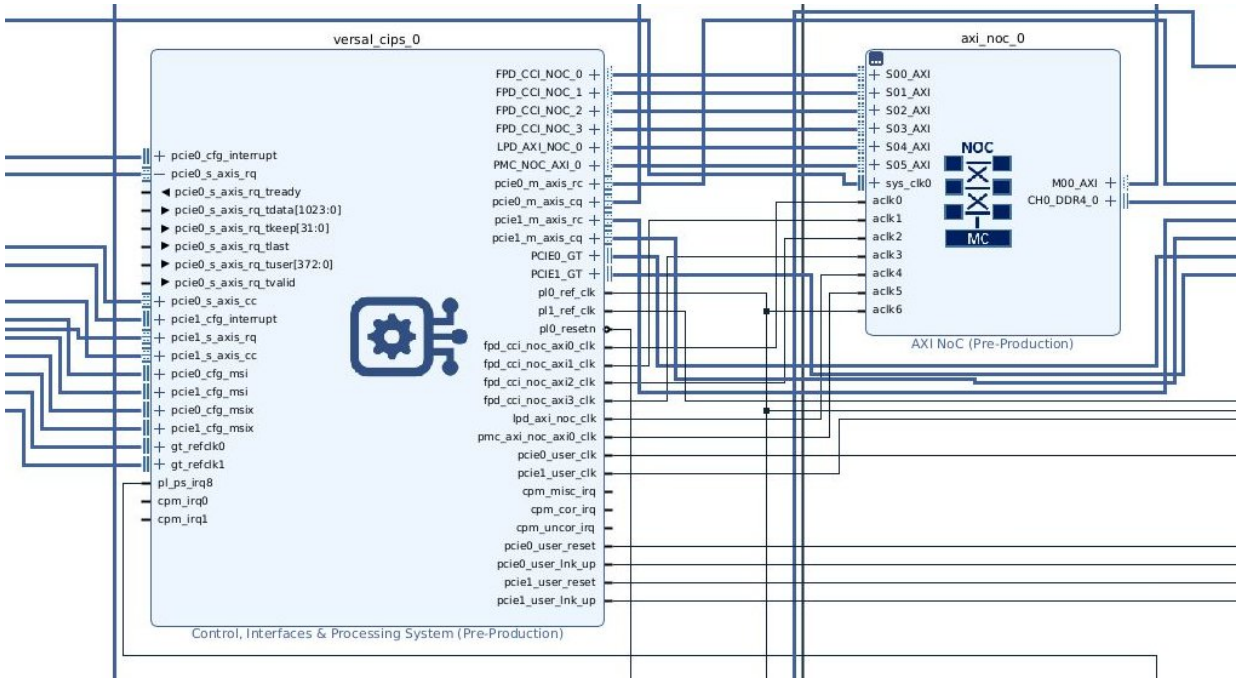
Versal Premium

- Ambitions for the next development card

- New FLX-155 development card
  - Based on a Xilinx Versal Premium VP1552 device

- Up to 48 duplex optical links

- PCIe Gen 5 x16 (512 Gbps)
  - 482 Gbps usable FELIX data bandwidth
  - 2 x8 lanes bifurcated
  - Now has to be routed through the CIPS
  - 1024 bit wide data bus



Conclusion

- Despite FELIX processing all data in the PL, the PS offers different QoL benefits
- The FELIX Versal webapp and Built-In SelfTest offer important monitoring and useful debugging options
- Further improvements will have to be made on the virtual Ethernet over PCIe connection
- Explore other useful features that could be implemented,using the PS

Repositories

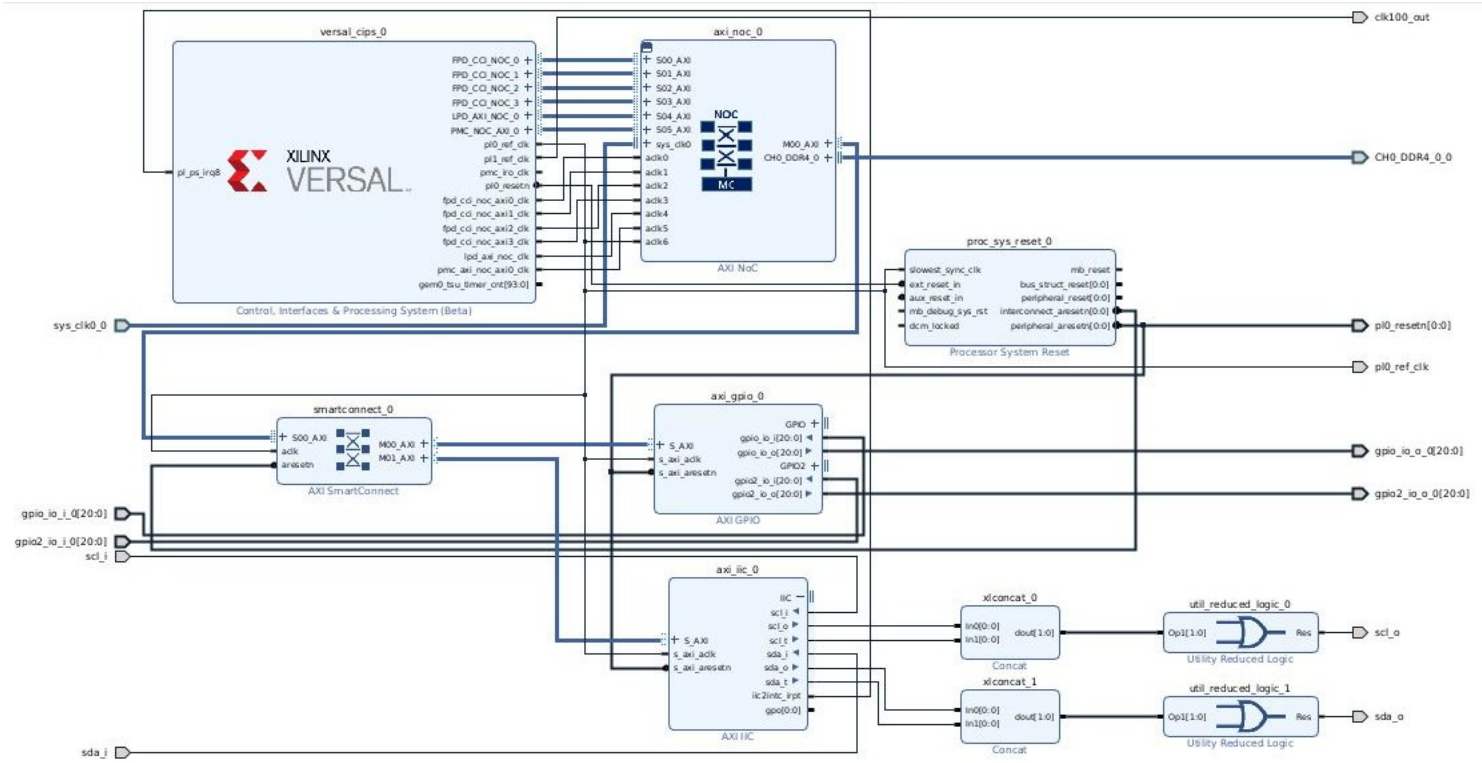
- FELIX Versal example drivers
  - https://gitlab.cern.ch/atlas-tdaq-felix/felix-versal-tools/flx-versal-example-drivers/
- flx182-petalinux-2022.2
  - https://gitlab.cern.ch/atlas-tdaq-felix/felix-versal-tools/flx-petalinux-2022.2
- FELIX Versal self-test webapp
  - https://gitlab.cern.ch/atlas-tdaq-felix/felix-versal-tools/flx-181-webapp
- Xilinx Chipscopy
  - https://github.com/Xilinx/chipscopy
  - https://xilinx.github.io/chipscopy/2022.2/
- Xilinx device tree
  - https://github.com/Xilinx/linux-xlnx/tree/master/Documentation/devicetree/bindings

Backup slides

Complete BD overview

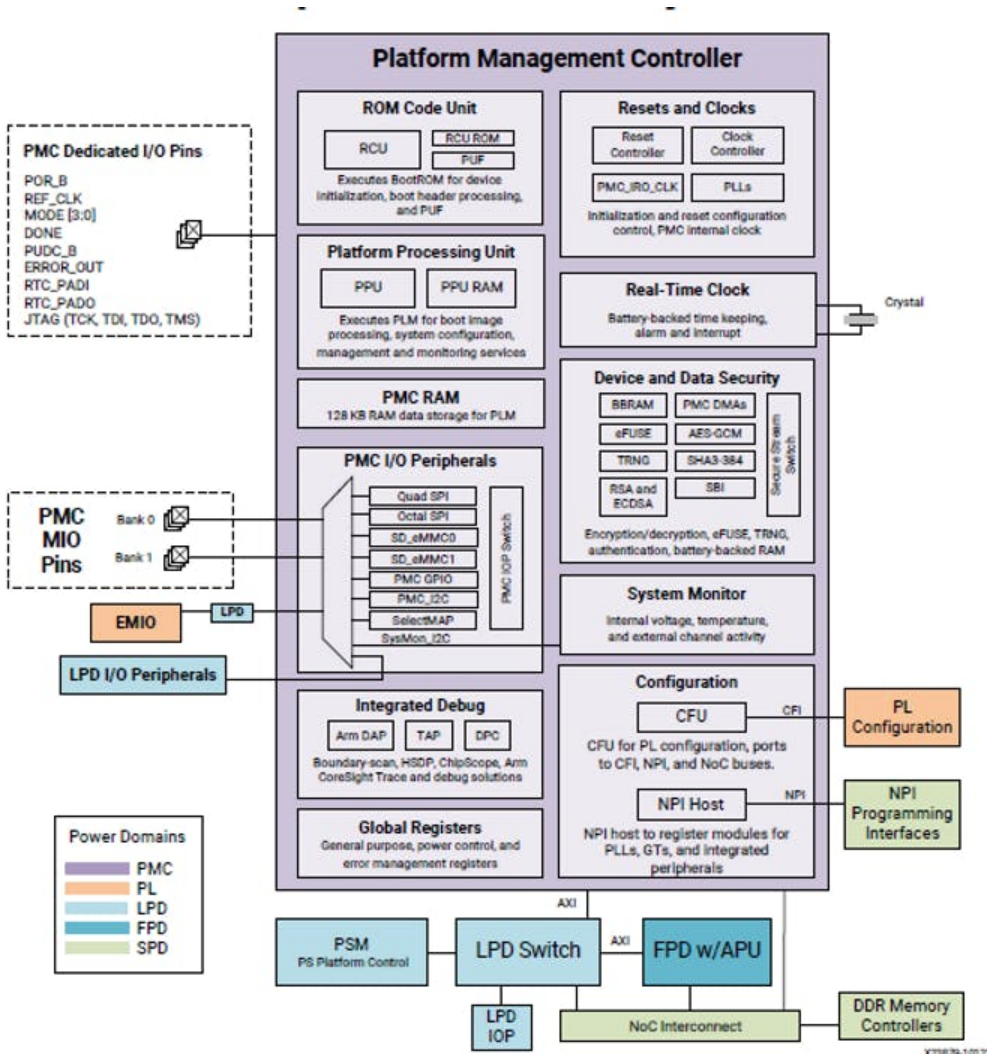
- Overview of the Block Diagram with all modules

- FLX-182 FELIX project



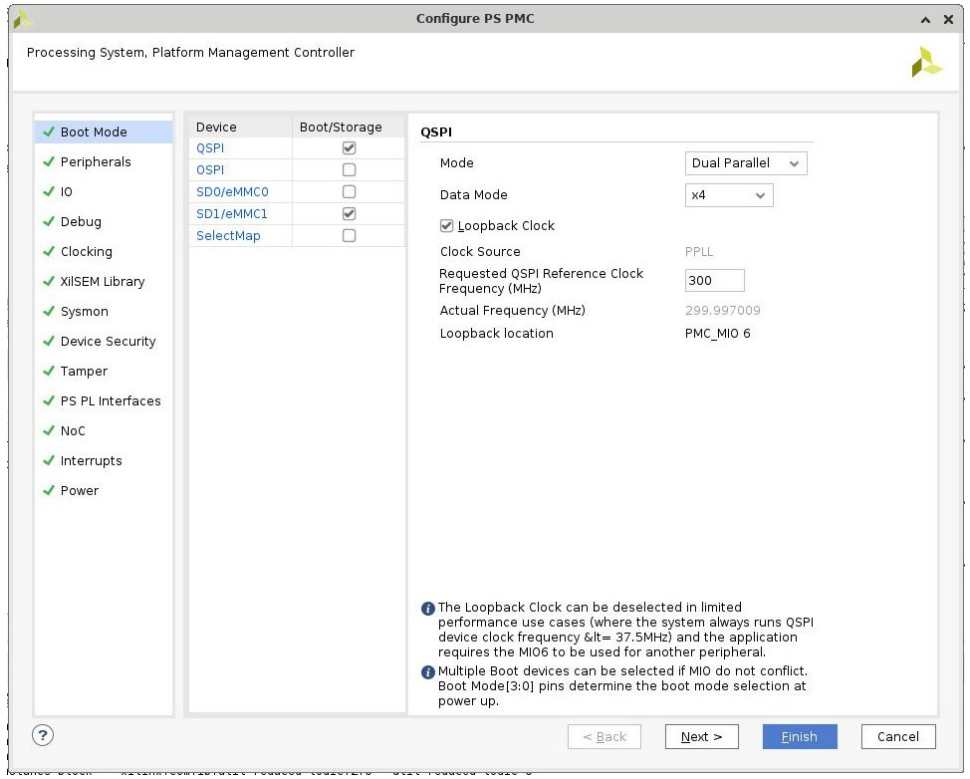
Platform Management Controller

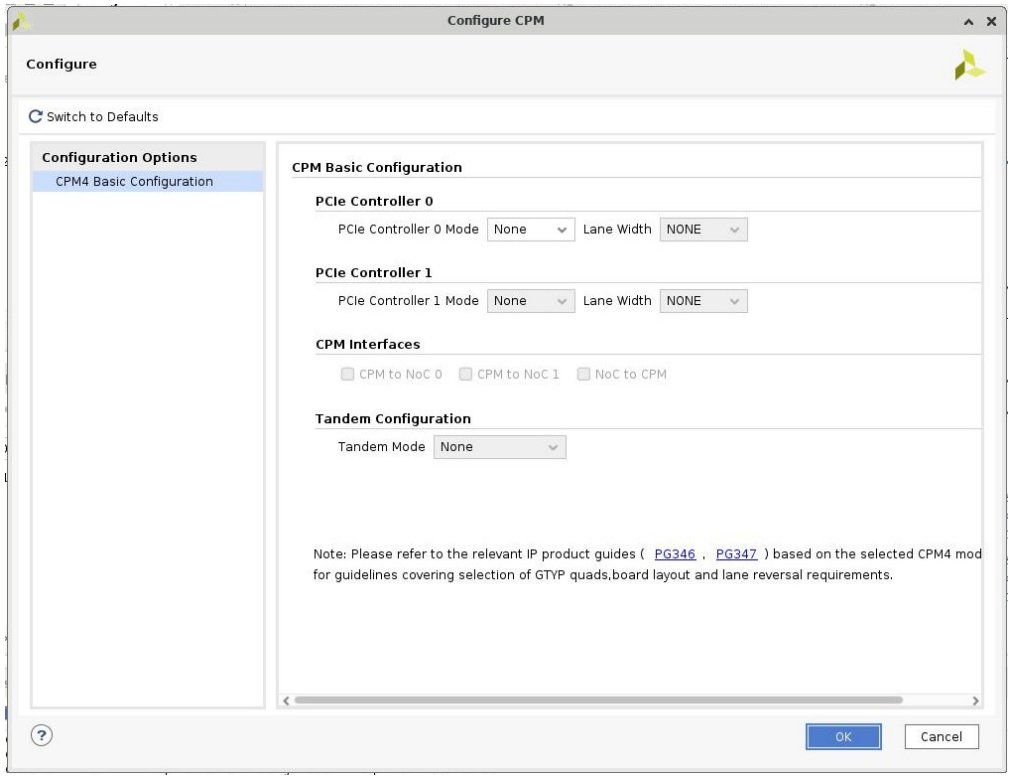
- More detailed overview of the PMC
- CFU part resposible for configuring the PL



Configuring the CIPS

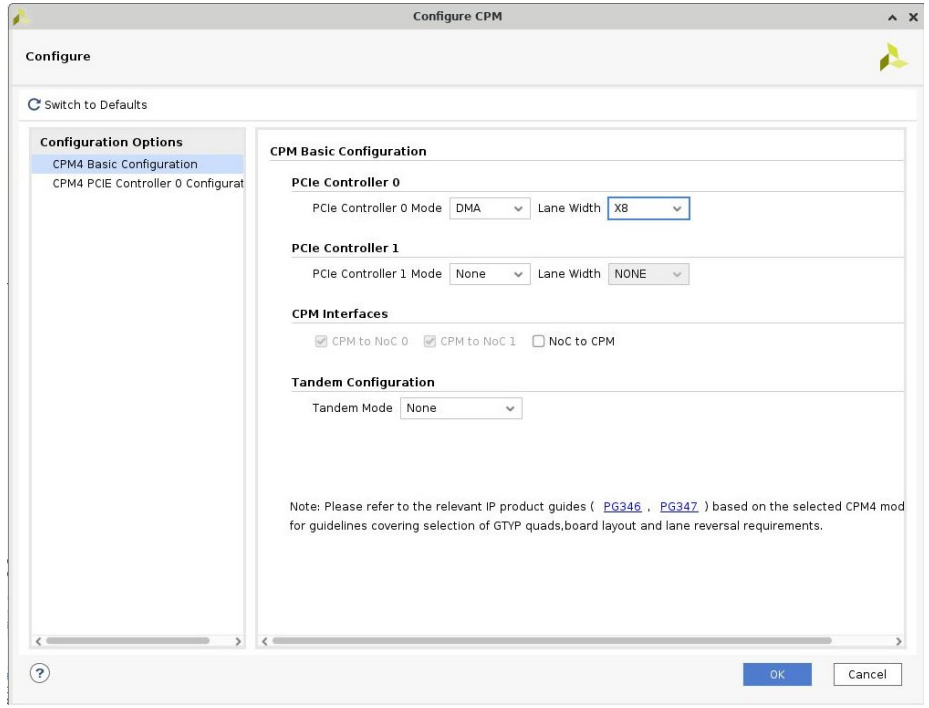
- PS PMC Plaform Management Controller
- CPM
  - Responsible for PCIe Gen4 connection
  - Operational without the need to configure the PL

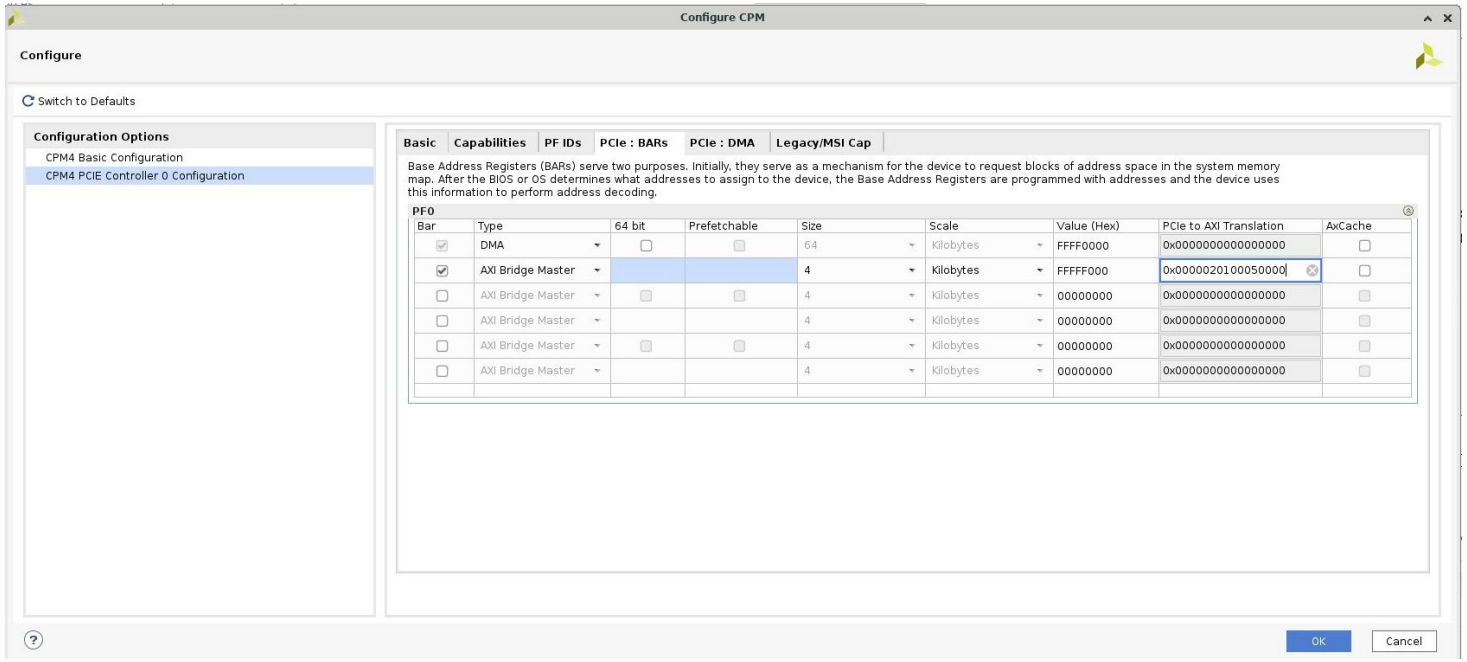




Virtual Network CPM

- CPM had to be configured in DMA mode
- AXI Bridge not supported with CPM mode





Running petalinux docker

- The docker image provides a pre-configured container with all required packages
  - Should be able to build petalinux without any issues
  - Immediately run petalinux commands

- cd flx-petalinux-2022.2/

- (sudo) docker run -it --rm -v $(pwd):/home/petalinux/build gitlab-registry.cern.ch/atlas-tdaq-felix/felix-versal-tools/petalinux-docker-ci

```
nayibb@nayibb-OptiPlex-3060:~/Documents/FPGA/flx-petalinux-2022.2$ sudo docker r
un -it --rm -v $(pwd):/home/petalinux/build gitlab-registry.cern.ch/atlas-tdaq-f
elix/felix-versal-tools/petalinux-docker-ci
[sudo] password for nayibb:
PetaLinux environment set to '/opt/petalinux'
WARNING: This is not a supported OS
INFO: Checking free disk space
INFO: Checking installed tools
INFO: Checking installed development libraries
INFO: Checking network and other services
WARNING: No tftp server found - please refer to "UG1144 2022.2 PetaLinux Tools D
ocumentation Reference Guide" for its impact and solution
petalinux@eb273cb36c55:~/build$
```

PetaLinux

- Set of tools to ease the development of embedded Linux on AMD devices
- Embeddded Linux SDK targeting FPGA-based SoC designs
- Based on Yocto