SoC Integration and Usage in the gFEX Hardware Trigger in ATLAS

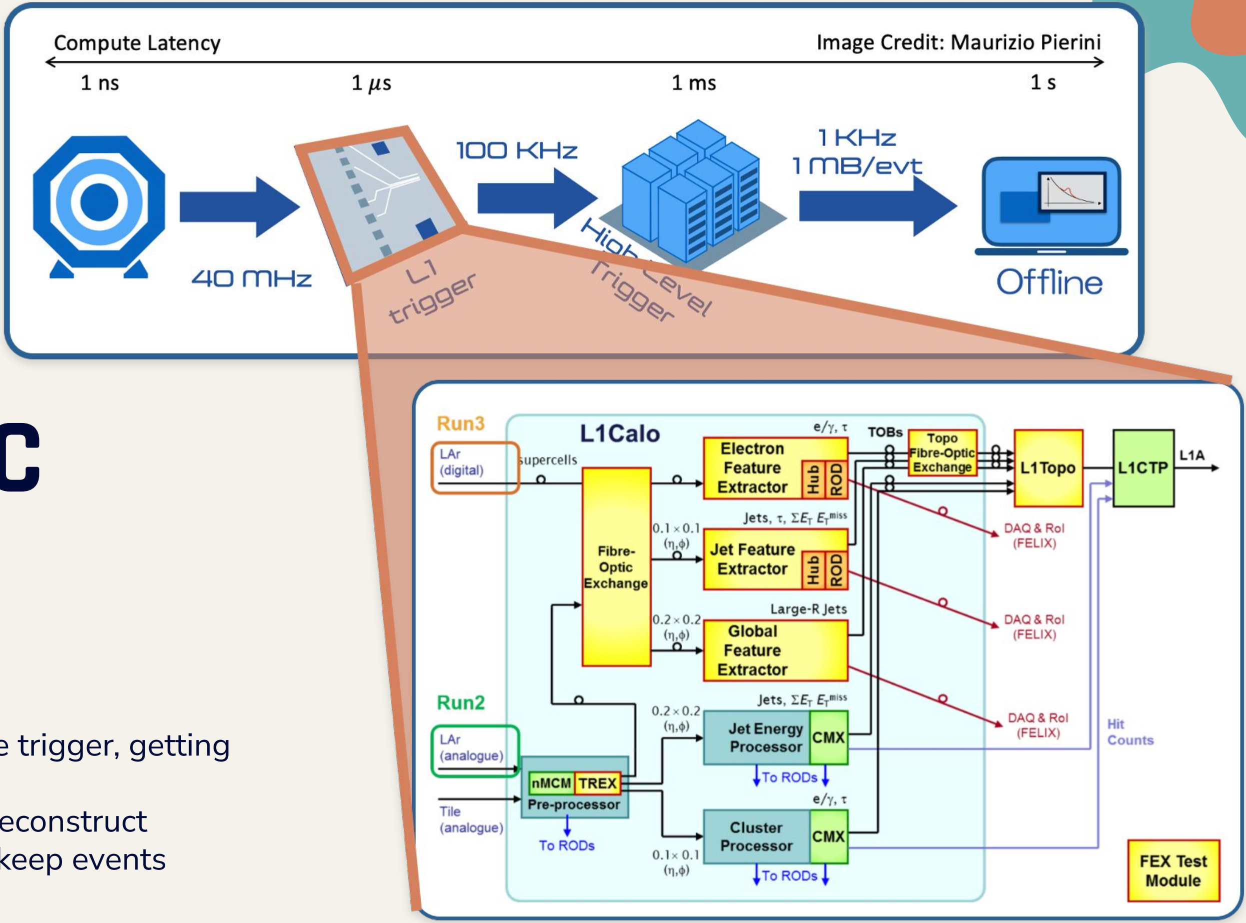
Kristin Dona on behalf of the gFEX team

3rd CERN System-on-Chip Workshop | October 4th, 2023

Table of contents

gFEX Introduction

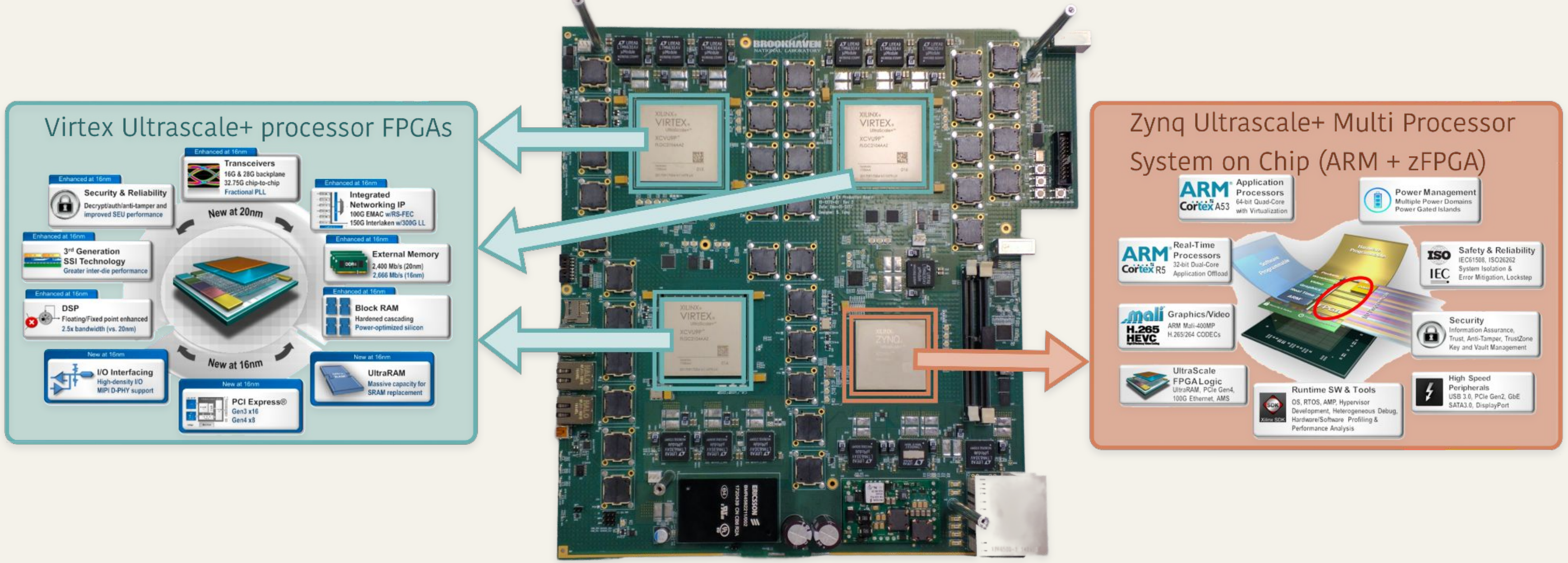
ATLAS Hardware Calorimeter Trigger for LHC Run 3



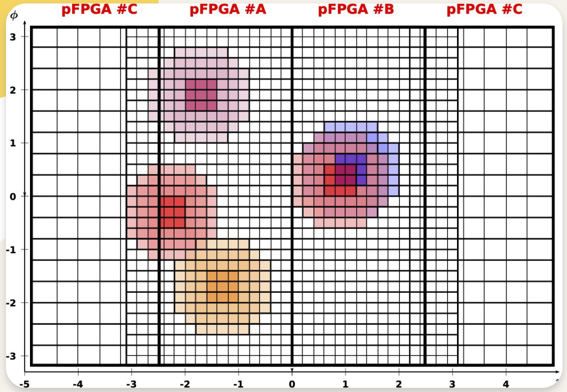
- gFEX is part of the ATLAS hardware trigger, getting input signals from the calorimeter
- Trigger systems are doing work to reconstruct physics objects and to decide if we keep events

gFEX hardware design

- gFEX is part of the hardware trigger, using FPGAs and SoC, with the goal of figuring out which data to save from the ATLAS detector
- The entire calorimeter is read out on one board for calculating global observables such as large-radius jets, missing transverse energy [MET], and pile-up estimation for triggers
- Board hardware
  - 3 Ultrascale+ (VU9P) processor FPGAs – where most data processing is done
  - Zynq Ultrascale+ MPSoC (ZU19EG) for monitoring, control and additional processes

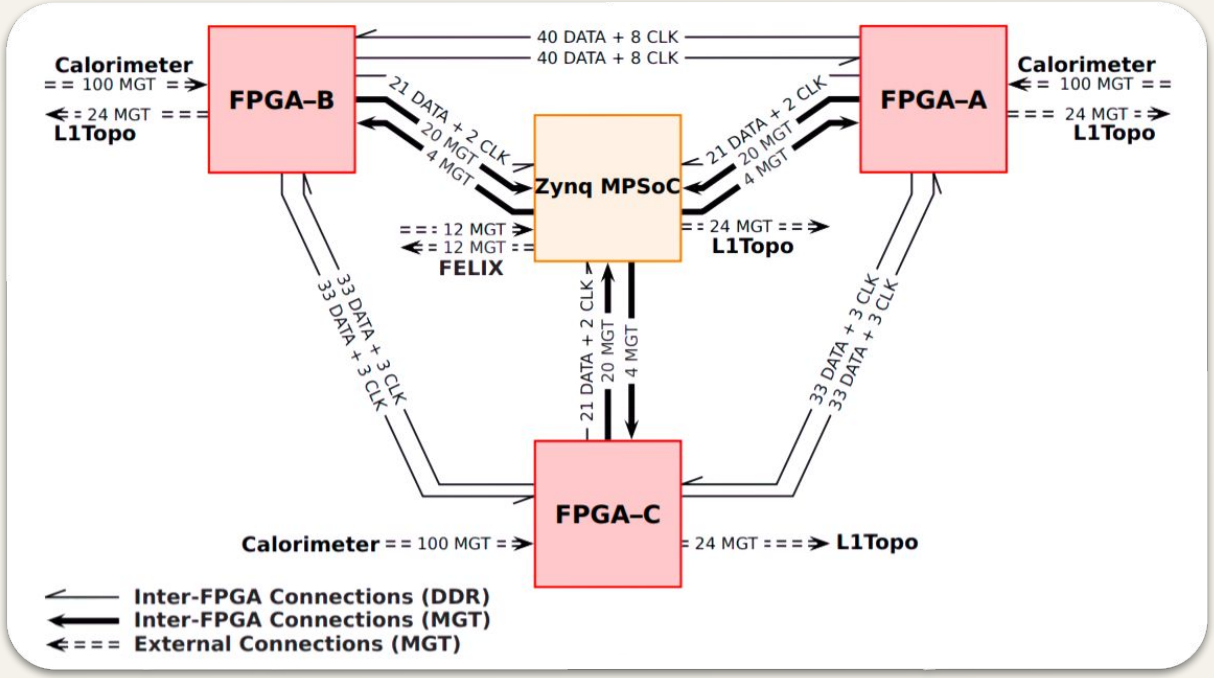


gFEX Zynq SoC Usage



Schematic illustrating the pFPGA limits of calorimeter data

- Many fibers going in and out of the gFEX system
  - 100 input fibers per FPGA
  - 11.2 Gb/s rate of input data per fiber
  - A few hundred nanoseconds of latency
- We have high-speed low latency system with custom firmware and support from the SoC
- All fiber control goes through the Zynq
- The zynq introduces increased flexibility and monitoring capabilities compared to the previous generation of L1Calo systems

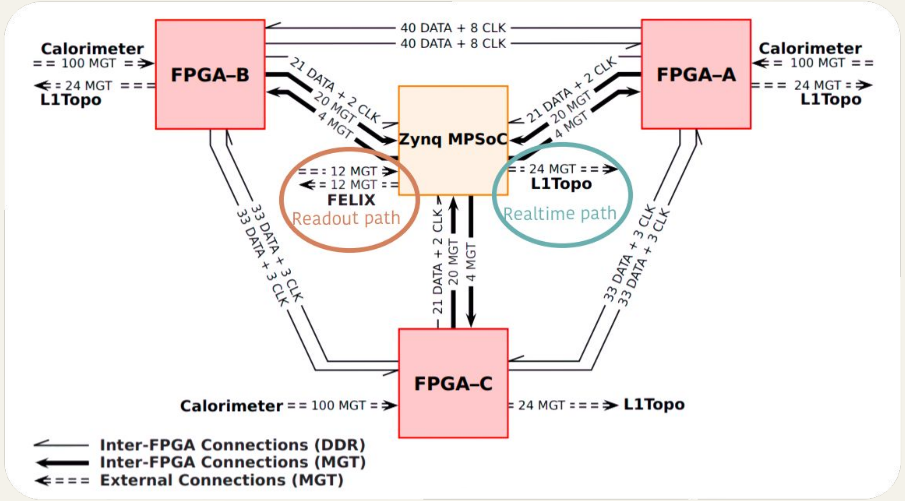


Schematic demonstrates dataflow through gFEX between the pFPGAs and the Zynq

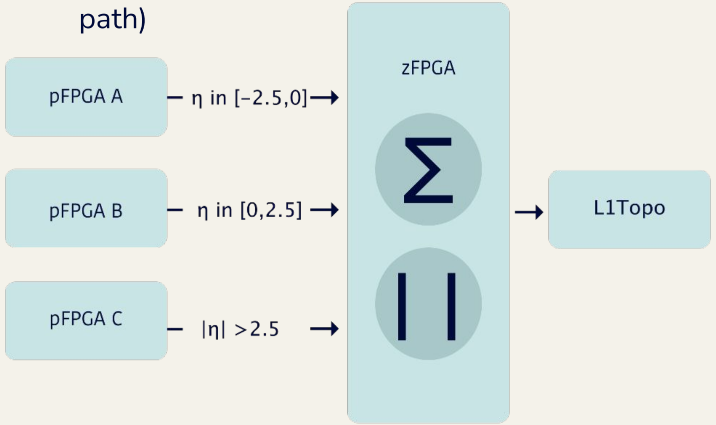
Zynq US+ MPSoC in the real time path

In addition to monitoring and control the Zynq provides real-time data processing functionality for global MET TOBs

- Zynq has to meet latency and timing
- pFPGA ⇄ zFPGA connection is GTH/GTY
  - UltraScale+ GTH (16.3 Gb/s)
  - UltraScale+ GTY (32.75 Gb/s)
- Zynq also receives algorithm output, does minimal processing, and then sends data downstream
  - Offers potential for additional processing here at a lower rate than that required by the real-time path (trigger path)



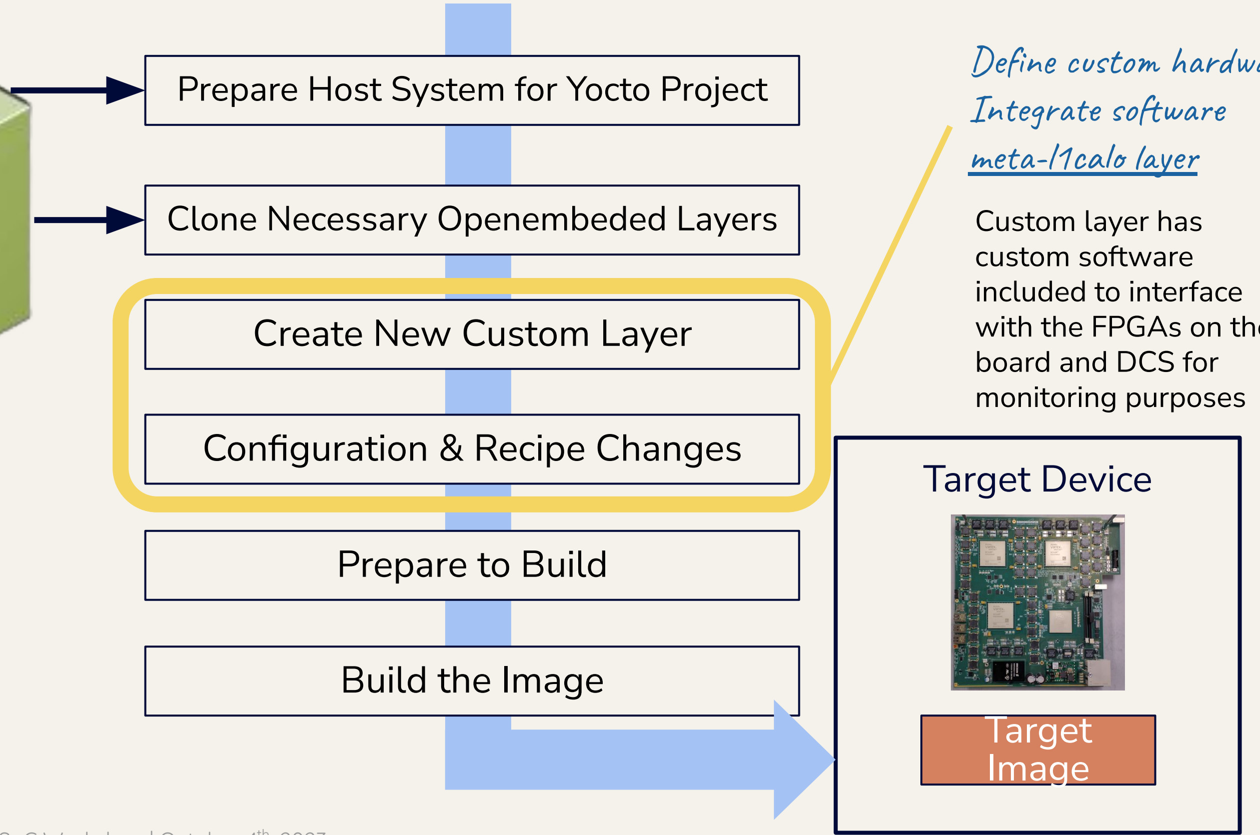
Schematic demonstrating that the FPGA on the zynq is part of the algorithmic realtime path for MET triggering



gFEX custom operating system

gFEX custom operating system

Yocto Project Docs


Prepare Host System for Yocto Project
Clone Necessary Openembeded Layers
Create New Custom Layer
Configuration & Recipe Changes
Prepare to Build
Build the Image
Target Device
Target Image

Define custom hardware

Integrate software

meta-l1calo layer

Custom layer has custom software included to interface with the FPGAs on the board and DCS for monitoring purposes

Current gFEX custom operating system

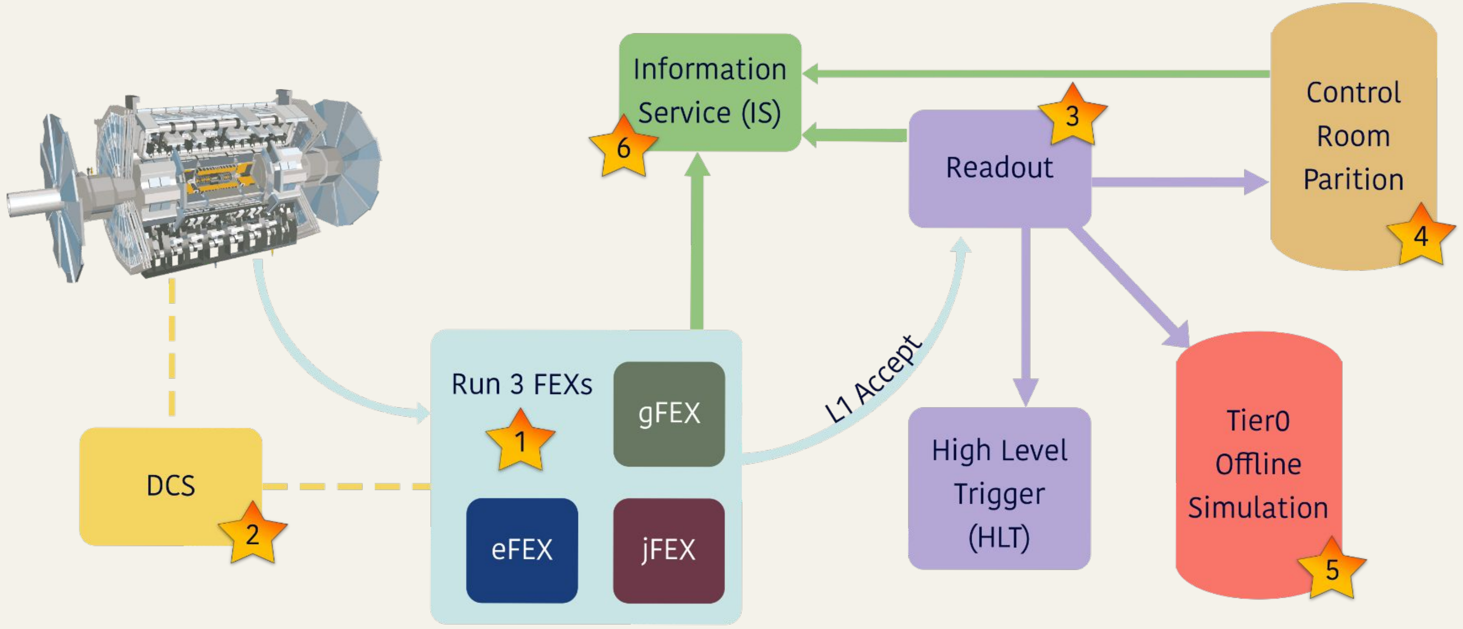
- We have a custom OS developed with Yocto/Openembedded build engine and bitbake
    - The meta-l1calo layer is where the "custom" design is done (link)
- Originally only used open-source yocto releases (rocko, sumo, zeus etc) but have now switched to slightly less open source yocto-manifests from AMD (previously known as Xilinx), currently working to upgrade to more recent releases (rel-v2023.1, rel-v2023.2)
    - Yocto-manifests benefits:
        - easier firmware integration: can pass in XSA file from firmware development to OS build
        - easier updating to newest Vivado
        - boot files built with OS (instead of externally with Vivado)
- Plans to update the system further
    - Implementation of CI on OS build in meta-l1calo repo
    - Update to a newer version version of yocto
        - current version in use requires Python 2, which is no longer supported
        - allow for upgrade to newer versions of the FPGA implementation tool
        - enable us to run the existing building process using Alma Linux 9

gFEX Zynq SoC monitor and control

gFEX Zynq SoC: monitoring and control

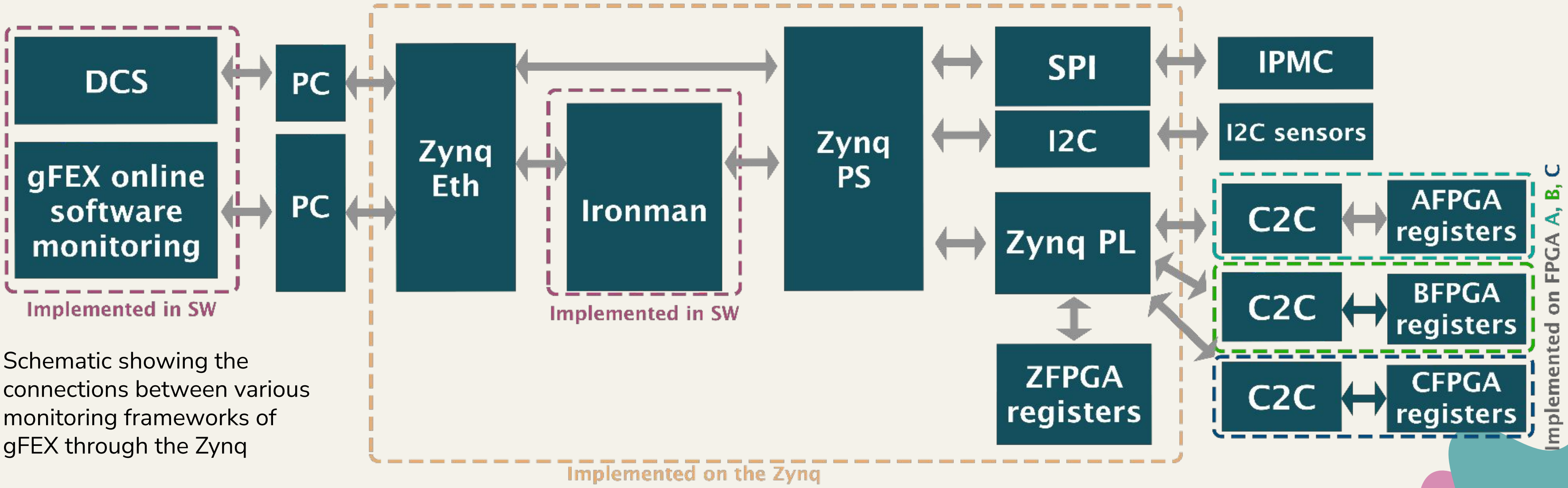
Monitoring Components

1. On-board (gFEX-Zynq)
2. Detector Control Systems
3. Readout monitoring
4. Control room partition
5. Offline simulation
6. Information service



Schematic outlining the various monitoring components that gFEX interacts with

gFEX Zynq SoC: monitoring and control



Schematic showing the connections between various monitoring frameworks of gFEX through the Zynq

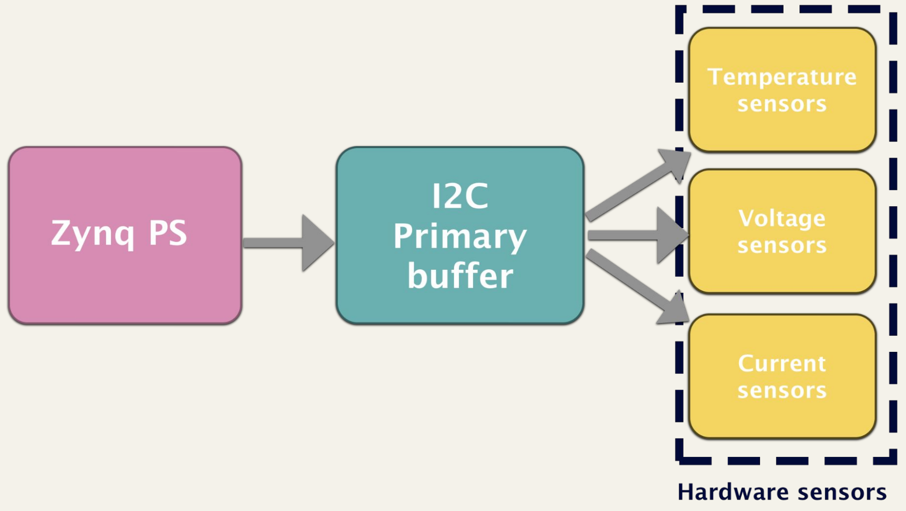
gFEX Zynq SoC: monitoring

The Zynq's primary task is monitoring and control.

The 3 main sources of this are

- IPMC (Intelligent Platform Management Controller)
- DCS
  - Via OPC-UA server
- ATLAS control room partition
  - Via IPBus packets

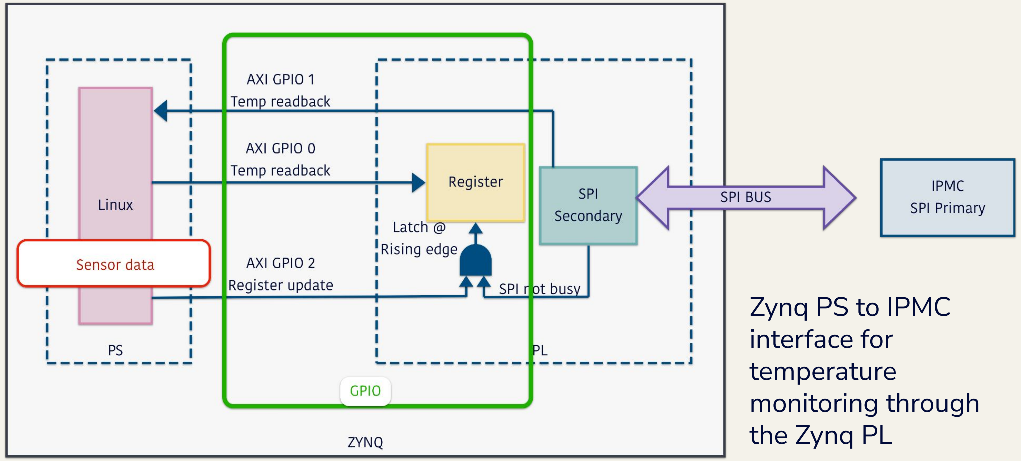
All of these are facilitated by reading hardware sensors over the I2C bus



I2C monitoring is done in the operating system using python modules to interface with I2Cbus and using a process that runs automatically on boot of the operating system

- Sensor values accessed over I2C using the periphery python module

gFEX Zynq SoC: Intelligent Platform Management Controller [IPMC]

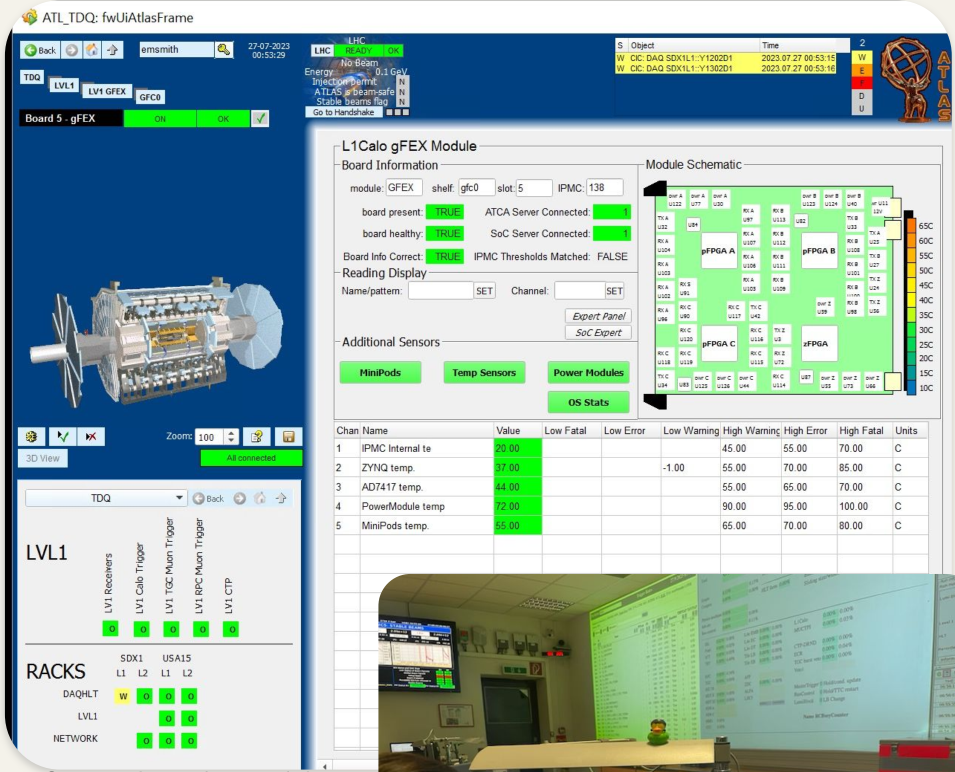


Zynq PS to IPMC interface for temperature monitoring through the Zynq PL

- IPMC = Intelligent Platform Management Controller
- Interface between board and the ATCA shelf that it's installed in
- SoC reads I2C sensors, and then updates the IPMC using the zFPGA as a go between
  - Temperature sensors on the board are read, the maximums found for different sensor types, and these values sent to the programmable logic (PL) using the GPIO. Then the IPMC can access the GPIO over the SPI Bus.
- Recently updated zFPGA firmware to revert to default values if there has been no update from the operating system

gFEX Zynq SoC: Detector Control Systems (DCS) Monitoring

- DCS is a graphical interface that is up in the control room so shifters can monitor the status of the detector
  - Used to monitor low level values like temperature
- Monitoring is done visa an on-board OPC UA server generated with a quasar framework (link)
- Developed and integrated into the Yocto OS build
- Python server using Milkyway by the quasar group at CERN
- Easy to add new elements without updating the entire OS!



S	Object	Time
W	CIC: DAQ SDX1L1::Y1202D1	2023.07.27 00:53:15
W	CIC: DAQ SDX1L1::Y1302D1	2023.07.27 00:53:16

L1Calo gFEX Module

Board Information

module: GFEX shelf: gfc0 slot: 5 IPMC: 138

board present: TRUE ATCA Server Connected: 1

board healthy: TRUE SoC Server Connected: 1

Board Info Correct: TRUE IPMC Thresholds Matched: FALSE

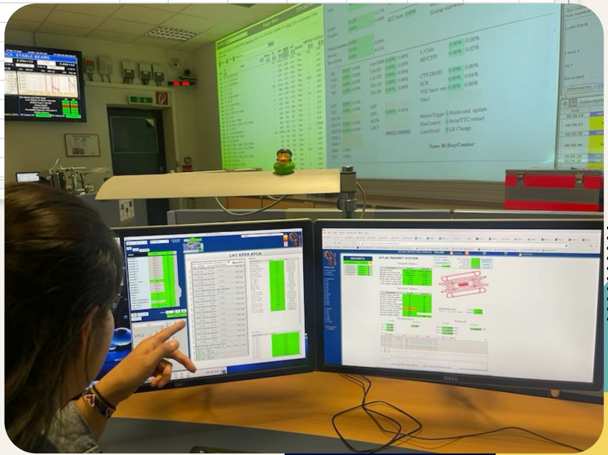
Reading Display

Name/pattern: SET Channel: SET

Additional Sensors

Chan	Name	Value	Low Fatal	Low Error	Low Warning	High Warning	High Error	High Fatal	Units
1	IPMC Internal te	20.00				45.00	55.00	70.00	C
2	ZYNQ temp.	37.00			-1.00	55.00	70.00	85.00	C
3	AD7417 temp.	44.00				55.00	65.00	70.00	C
4	PowerModule temp	72.00				90.00	95.00	100.00	C
5	MiniPods temp.	55.00				65.00	70.00	80.00	C

↑ Screenshot of one of the many DCS monitoring screens

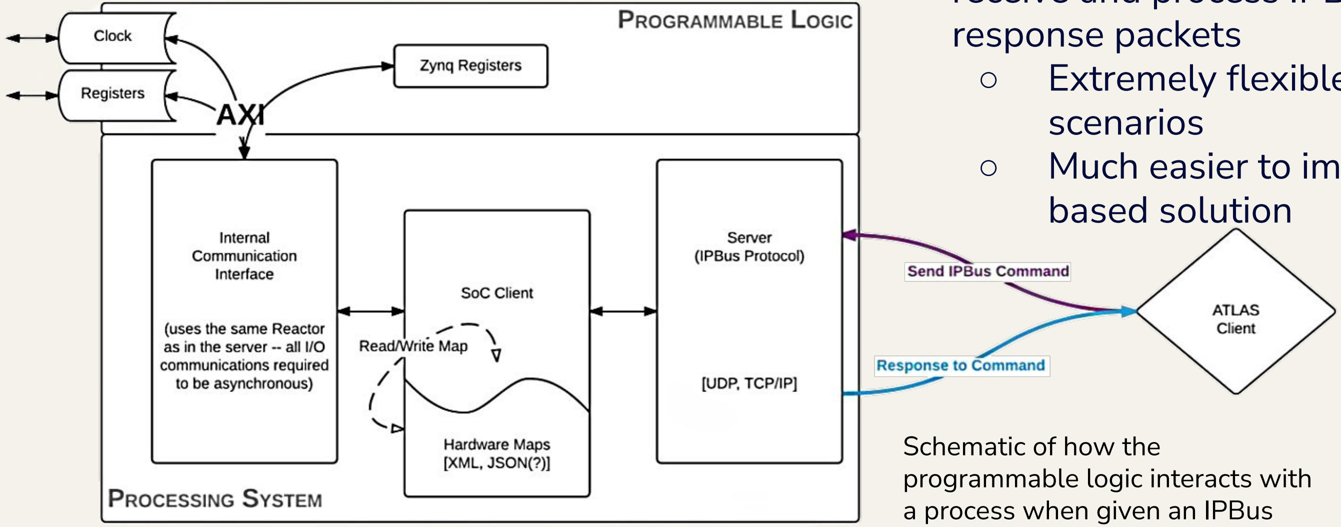


Real night shifter in the ACR monitoring the gFEX DCS (Not a paid actor)⟶

gFEX Zynq SoC: IPBus control and monitoring with Ironman

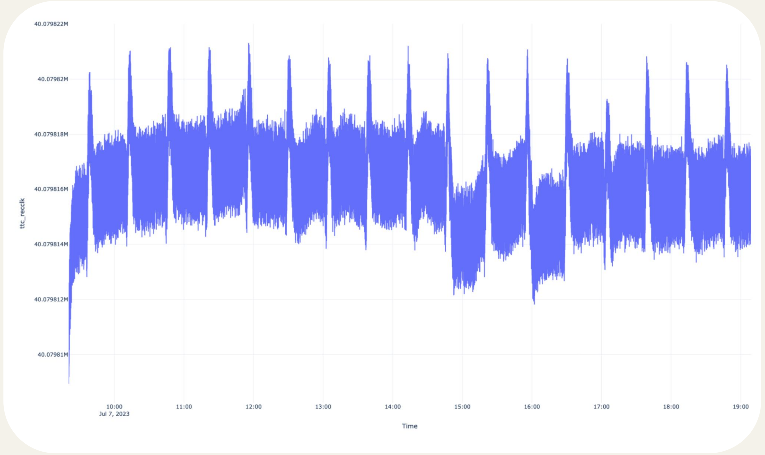
One of the greatest advantages of the Zynq SoC is that it performs many functions that were previously required by the firmware, but are much easier to handle in software

- Using custom python module Ironman we can receive and process IPBus packets, and send response packets
  - Extremely flexible, can be used in many scenarios
  - Much easier to implement than a firmware based solution



Schematic of how the programmable logic interacts with a process when given an IPBus packet

gFEX Zynq SoC: slow control monitoring plans

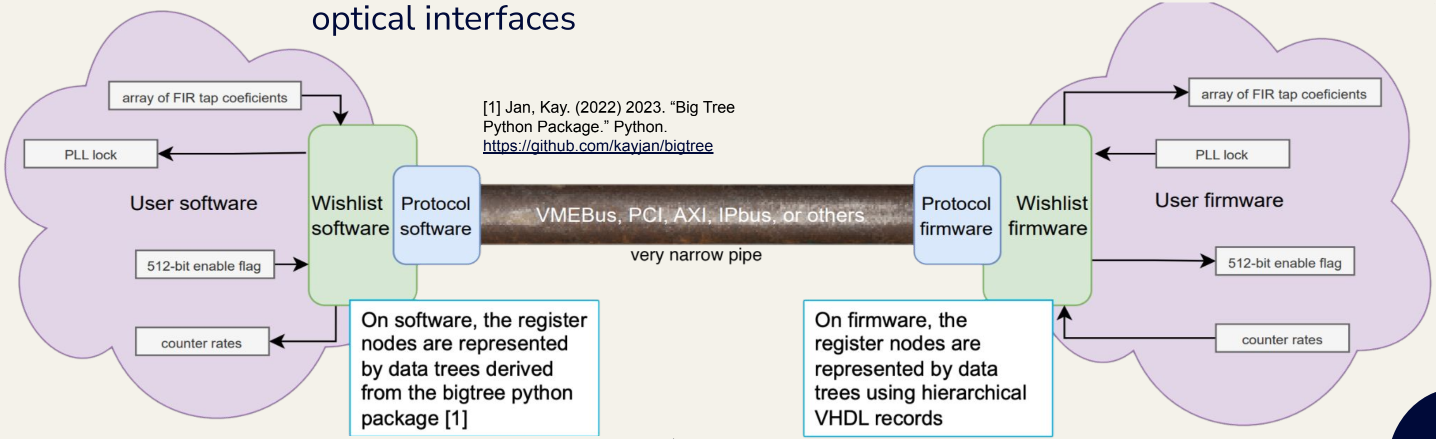


Monitoring plot of 40 MHz recovered clock from GBT-FPGA interface. Consistent low level values over time validate hardware is working (link)

- Recently updated slow control for online monitoring of firmware algorithms
- We also implemented wishlist software (gitlab, documentation) for producing firmware and software to ensure the control and monitoring register addressees and masks are consistent between software and firmware.
- We are implementing a monitoring system based on zero-deadtime counters to monitor
  - Frequency
  - PLL lock
  - FIFO busy
  - Ready
  - And more!
- Expansion to our other monitoring frameworks, running on the zynq and making plots automatically

gFEX Zynq SoC: slow control monitoring plans using wishlist

- Newly implemented wishlist software designed to simplify the process of requesting what information is shared between software and firmware
    - List of wishes are defined in the Input YAML file
    - Wishlist process this list and returns software and firmware representations of the resulting register tree
    - With this, we plan to add event metering
        - input rates from LAr and Tile, trigger object rates, and online monitoring of electrical and optical interfaces



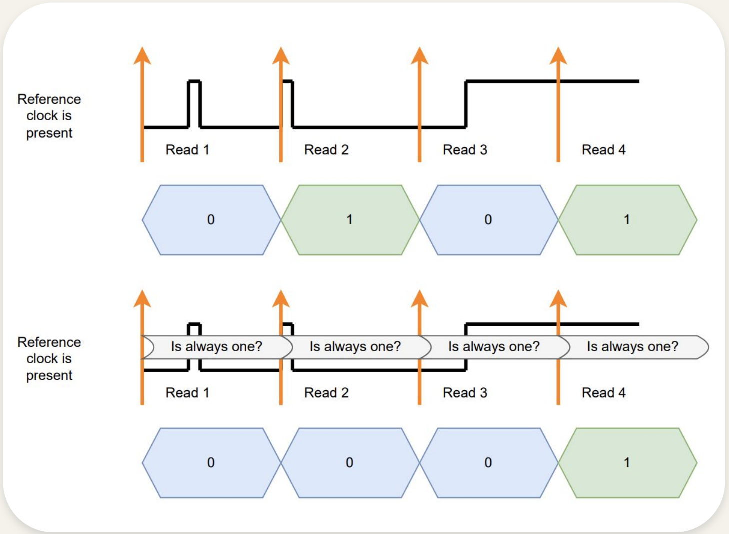
[1] Jan, Kay. (2022) 2023. "Big Tree Python Package." Python. https://github.com/kayjan/bigtree

On software, the register nodes are represented by data trees derived from the bigtree python package [1]

On firmware, the register nodes are represented by data trees using hierarchical VHDL records

Representation of how wishlist software acts as the go-between for software and firmware for configuring trigger and monitoring readout systems
From (link)

gFEX Zynq SoC: slow control monitoring with integration rather than reading single data points



- Newly implemented method of sampling for measuring to have zero-deadtime
- Using accumulators with overflow detection enables detection of transients
- Improves our ability to detect issues with the system

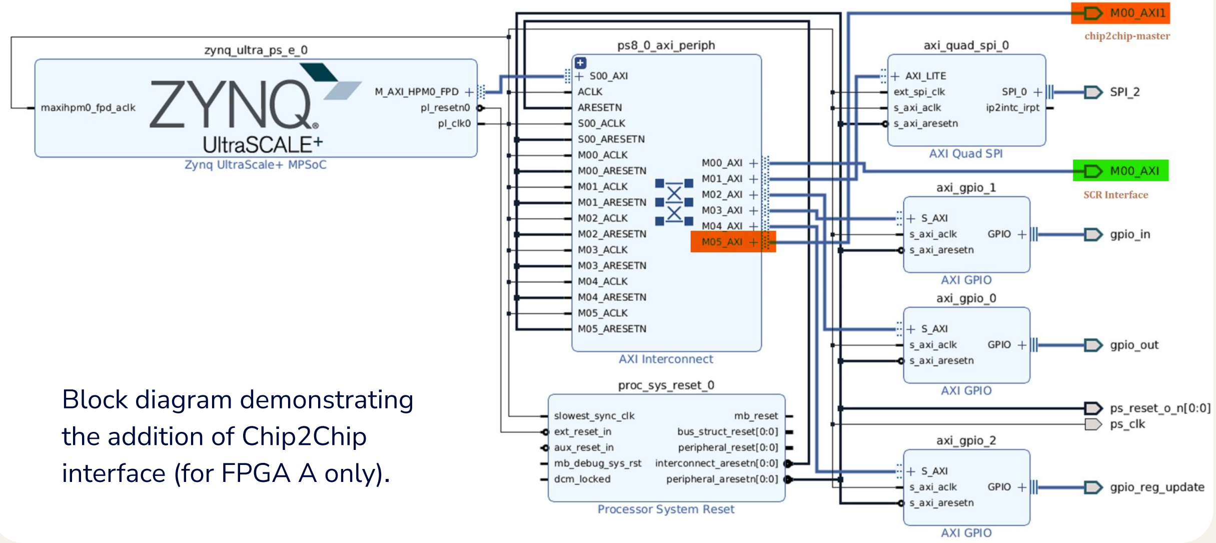
Schematic showing the integration based sampling method for deadtime checks in monitoring
From link

Processing system [PS] and Programmable Logic [PL] interfaces

Zynq SoC PS - PL Interfaces

- Processing system and programmable logic [PS-PL] interface is mostly driven by the firmware, and usually has to be taken into account when building the OS
    - Comes into play in the device tree, which defines hardware connections to the OS
    - Provided by the inclusion of the XSA file in the OS build
- Moving towards the use of AXI-DMA interface, "AXI DMA provides high-bandwidth direct memory access between memory and AXI4-Stream target peripherals"
    - Much quicker transfer of data between FPGAs and between PS and PL on the Zynq
    - Requires drivers to be implemented on the OS that match the implementation on the firmware
    - Before, we have used AXI-FIFO interfaces between Zynq FPGA and other FPGAs
- Currently working on implementing changes to our AXI crossbar that will add 3 new outputs for the Chip2Chip interface to fpgas A, B, and C in the PS part

Zynq SoC updated chip2chip PS



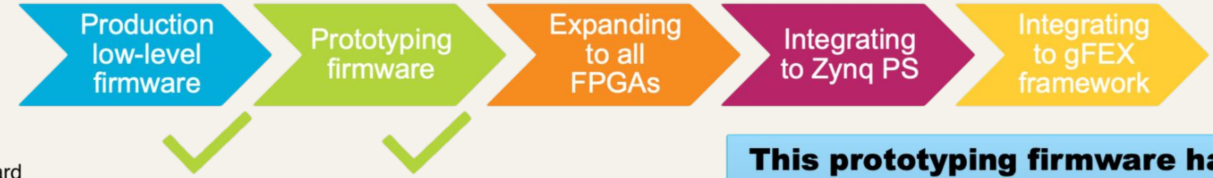
Block diagram demonstrating the addition of Chip2Chip interface (for FPGA A only).

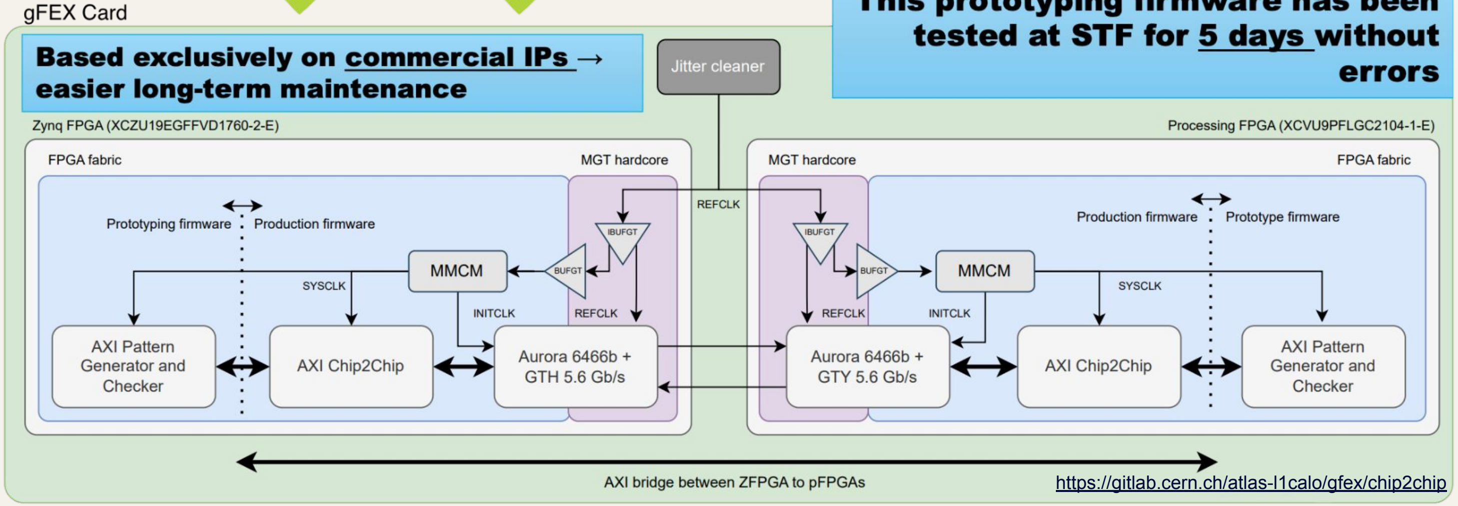
Recent updates to the AXI crossbar adds new outputs for the Chip2Chip interface to FPGAs A, B, and C in the PS part of our PS - PL interface

This will eventually replace the SCR interface (status control register interface) currently in place.

- Replacing SCR with a Chip2Chip will help in providing more robust solution as its AMD provided IP
- AXI C2C + Aurora 6466b IPs translates AXI transfers to high-speed serial lines and vice-versa.
  - This enable us to send/receive AXI transfers in our processing FPGAs (that are connected to the zynq using high-speed serial lines).

Zynq SoC updated chip2chip PS




gFEX Card
Jitter cleaner
Zynq FPGA (XCZU19EGFFVD1760-2-E)
Processing FPGA (XCVU9PFLGC2104-1-E)
FPGA fabric
MGT hardcore
REFCLK
Prototyping firmware
Production firmware
IBUFGT
BUFGT
MMCM
SYSCLK
INITCLK
AXI Pattern Generator and Checker
AXI Chip2Chip
Aurora 6466b + GTH 5.6 Gb/s
Aurora 6466b + GTY 5.6 Gb/s
Prototype firmware
AXI bridge between ZFPGA to pFPGAs

Based exclusively on <u>commercial IPs</u> → easier long-term maintenance

This prototyping firmware has been tested at STF for <u>5 days</u> without errors

https://gitlab.cern.ch/atlas-l1calo/gfex/chip2chip

Step-by-step plan to implement the chip-to-chip interface in gFEX

From link

CERN SoC petalinux template

CERN SoC petalinux template

- We are actively learning from the CERN SoC Interest Group how to adopt the CERN SoC petalinux template (link)
    - This would enable us to inherit a common approach and several components, such as DHCP client, DMA driver, Linux I/Os, and etc.
    - Also invites an opportunity for support from the CERN SoC Interest Group
    - Petalinux is a superset of Yocto, which should make it easier to use

yocto PROJECT · PetaLinux

Conclusions and outlook

Conclusions and outlook

The gFEX L1Calo Trigger hardware system uses a Zynq SoC with custom OS for monitoring and control of the board

- Monitoring and control methods are well established and function well and are being expanded
  - IPMC - SoC reads I2C sensors, and then updates the IPMC using the zFPGA as a go between
  - DCS - graphical interface for monitoring low level values like temperatures
  - ATLAS control room partition via IPBus packets
- PS - PL interfaces
  - Currently implementing changes to our AXI crossbar that will add 3 new outputs for the Chip2Chip interface to other fpgas on the board in the PS part
- Custom operating system
  - Current setup is functioning very well, it can be updated by a variety of people and has all necessary requirements
  - Considering moving to the petalinux system to inherit a common approach and support

Thank you!

On behalf of the gFEX team

PIs

- David Miller
- David Strom
- Lauren Tompkins
- Michael Begel
- Sabine Lammers
- Tae Min Hong

Engineers

- Daniel (Dan) Ulmes
- Emre Ercikti
- Fukun Tang
- Marcos Oliveira
- Michal Husejko
- Ramakanth (Raam) Desani
- Shaochun Tang

Postdocs and scientists

- Bastian Schlag
- Cecilia Torsciri
- Despoina Sampsonidou
- Jennifer Roloff
- Rajat Gupta
- Simone Sottocornola

Students

- Anthony Carroll
- David Guerra Nunez
- Emily Smith
- Greg Meyers
- Jennifer Lue
- Kristin Dona
- Rabia Omar
- Sergey Scoville

gFEX

Brookhaven National Laboratory

THE UNIVERSITY OF CHICAGO

INDIANA UNIVERSITY

University of Pittsburgh

UNIVERSITY OF OREGON

Stanford University

Backup

Acronym Glossary

- CI: Continuous integration
- CTP: central trigger processor
- DCS: Detector control systems
- gFEX: global Feature EXtractor
- GPIO: general purpose input/output
- HLT: High level trigger
- I2C: Inter integrated circuit - a bus interface connection protocol incorporated into devices for serial communication
- IPBus: Internet Protocol
- IPMC: Intelligent Platform Management Controller
- L1: Level 1 triggers
- MET: missing transverse energy
- OKS: object kernel support – database system specific to online software
- OPC-UA: open platform communication unified architecture - specific protocol for client-server communication
- PS: Processing system
- PL: Programmable logic
- PUC: pileup correction
- Soc: System on chip
- TOBs: Trigger objects
- TOPO: topological algorithms trigger
- Zynq US+ MPSoC - Zynq ultrascale + multiprocessor system on chip