Developments in EdgeML Inference in FPGAs, and their Applications in PET

Ryan Herbst, Ryan Coffee, Larry Ruckman
SLAC National Accelerator Laboratory

03/17/2023

U.S. DEPARTMENT OF ENERGY | Stanford University

SLAC NATIONAL ACCELERATOR LABORATORY

SNL (SLAC Neural network Library) Framework Goals

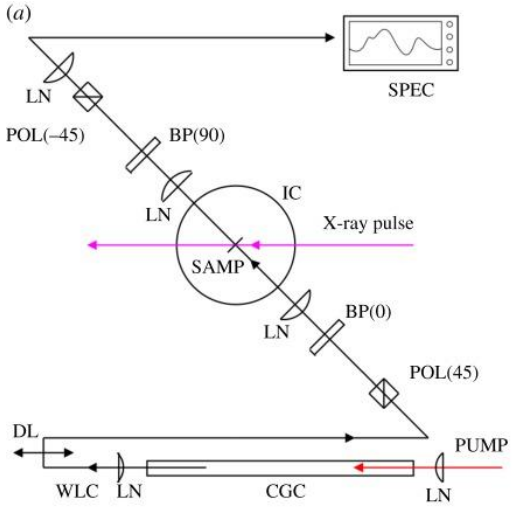
- Provide a **set of libraries** to synthesize AI inference networks into FPGAs
- Networks of a medium size,
  - **10 - 20 layers**, ~100s of thousand trainable parameters
  - Total end to end latency of **1uS** to 10ms
  - Deep networks call for AI-specific ASICs, e.g. IPU or Groq, but **FPGAs still ubiquitous** at sensor
- Pipelined implementations targeting a frame rate of **100kHz - 1MHz**
- **Dynamic reloading of weights** and biases that avoids re-synthesis
- Support a **Keras-like API** for layer definition and configuration
- Allow for a standard interface such as **HDF5** for loading weights and biases
- Allow for a modular approach with the ability to implement new and **custom layers**

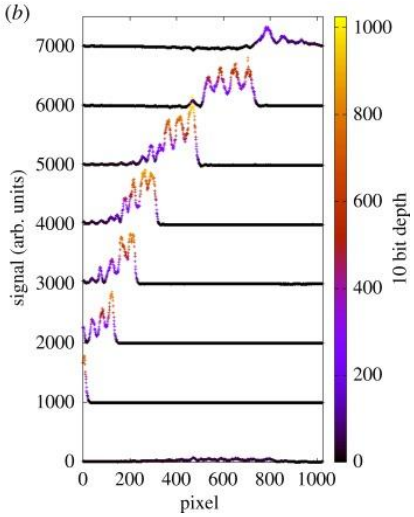
Why Dynamic Weights & Biases?

- Our SNL implementation is targeting **scientific instruments** which will continuously adapt to new data and **changing environments**
  - **High speed training** to supports this goal
  - Bias and weight **updates in real time**
- Some AI-to-FPGA frameworks take the weights and biases as an input, pruning portions of the network structure to save resources
  - **Re-synthesis is required** for each new training set
  - **Risk of the FPGA implementation failing** due to increase resources usage, timing failures or massive change in internal interconnect structure
- **Large FPGA designs can take hours to days** for the HLS -> synthesis -> place and route cycle to complete
- This is a trade off between **robustness vs. latency** and resource usage

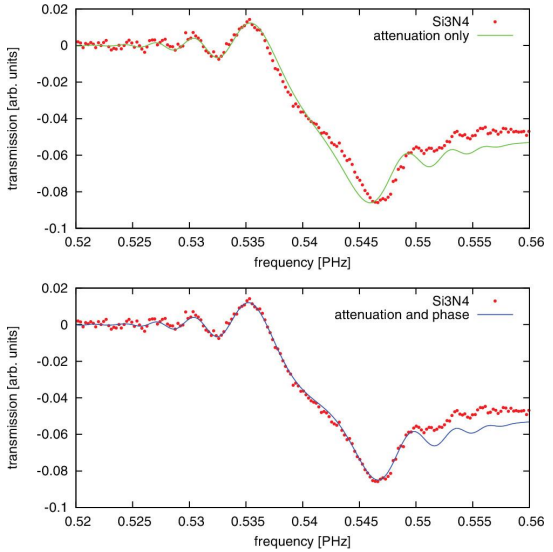
Scheme to Measure the X-ray Timing in LCLS

- Concept borrowed from DOE-funded LCLS (Linac Coherent Light Source) X-ray timing project (“TimeTool”)
- Use frequency chirped broadband laser pulse to create a spectrum-to-time mapping
- Result is the relative timing between FEL (Free-electron Laser) X-ray pulse and the experimental optical laser





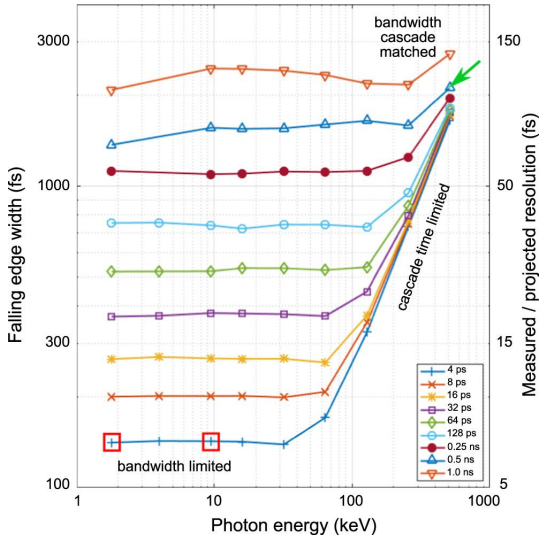
Bionta *et al.,* Opt Express. (2011) **19** (22) 21855-65. doi: 10.1364/OE.19.021855. PMID: 22109037.
Bionta *et al.,* Rev Sci Instrum. (2014) **85** (8) 083116. doi: 10.1063/1.4893657. PMID: 25173255.

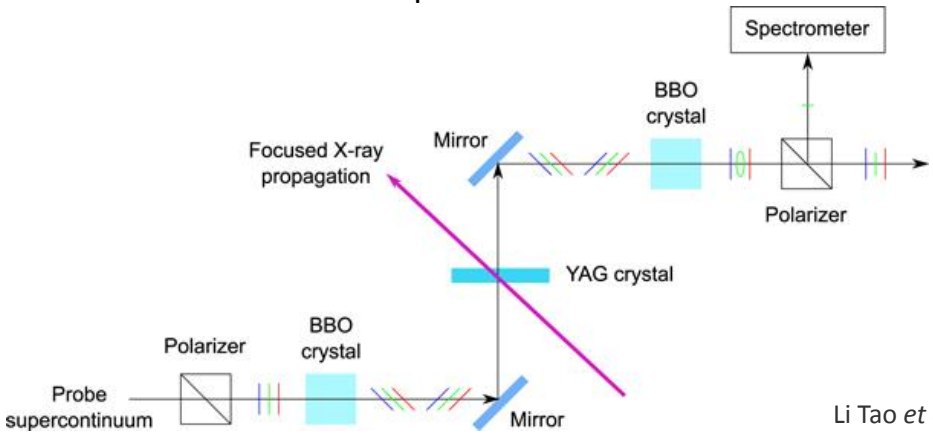


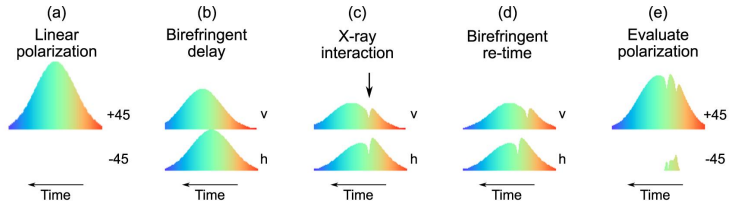
Coffee *et al.,* Philo. Trans. A Math Phys. Eng. Sci. (2019) **377** (2145):20180386. doi: 10.1098/rsta.2018.0386.

Translating Method to Medical Imaging

- The annihilation photons in PET creates a similar cascade in bulk materials as x-ray photons from LCLS, but much higher energy/photon
- Measure (with the chirped laser) the induced refractive index change directly (-45deg) rather than wait for scintillation process.
- Birefringence measured with interference for sensitivity to single photon events
- 511 keV photons produce ps-scale cascades, sets optimal chirp for resolution below 0.2ps



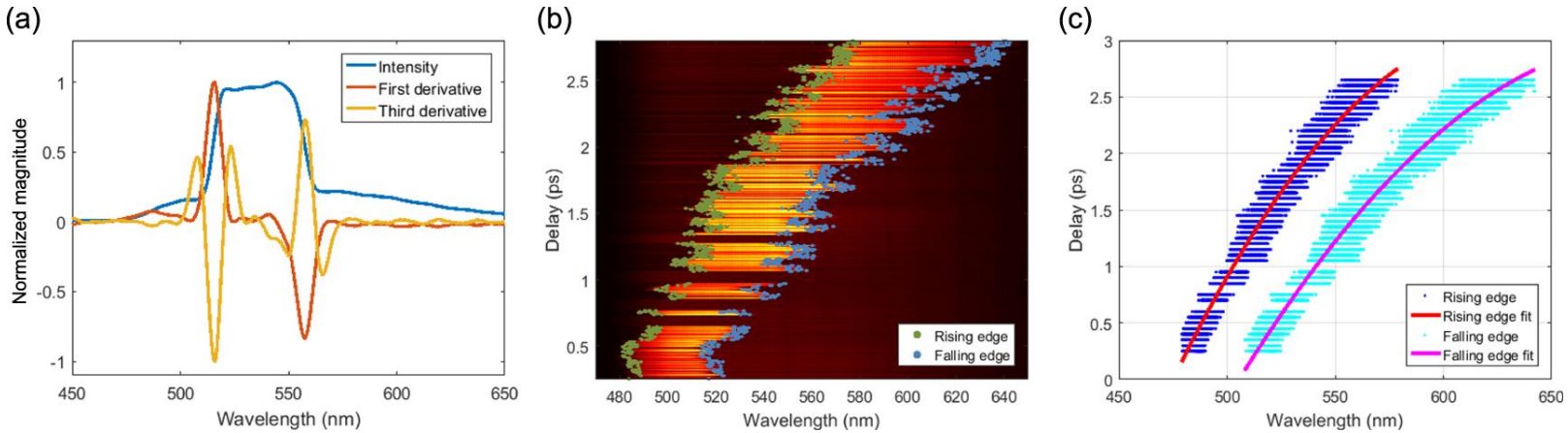




Li Tao *et al* (2021) *Phys. Med. Biol.* 66 045032, DOI 10.1088/1361-6560/abd951

Running the PET algorithm with LCLS Light Source

- Edge identification via conventional edge-finding or CNN algorithms
- Results for PET will be rare events (kHz – MHz) individual rows in image that show rise and fall edges
- Rare event edge locations are the only relevant information
- Can we produce that information at the camera, then feature extract the framegrabber before software receives it (“lossy” compression)



Li Tao *et al* 2021 *Phys. Med. Biol.* 66 045032, DOI 10.1088/1361-6560/abd951

CoaXPress over Fiber (CXPoF) Framegrabber

SLAC

- Array of optical cameras for parallel readout
- Each camera hosts ~1000 of detection elements (as signal rows)
- Streaming images are evaluated at CXPoF receiving FPGA for “live” rows
- Results from “live” rows are then used for direct PET reconstruction

CXP over Fiber Protocol

Vision Research S991 camera

Up to 937 FPS at 4096 x 2304 (8 bit)

Up to 52,080 FPS at reduced resolutions

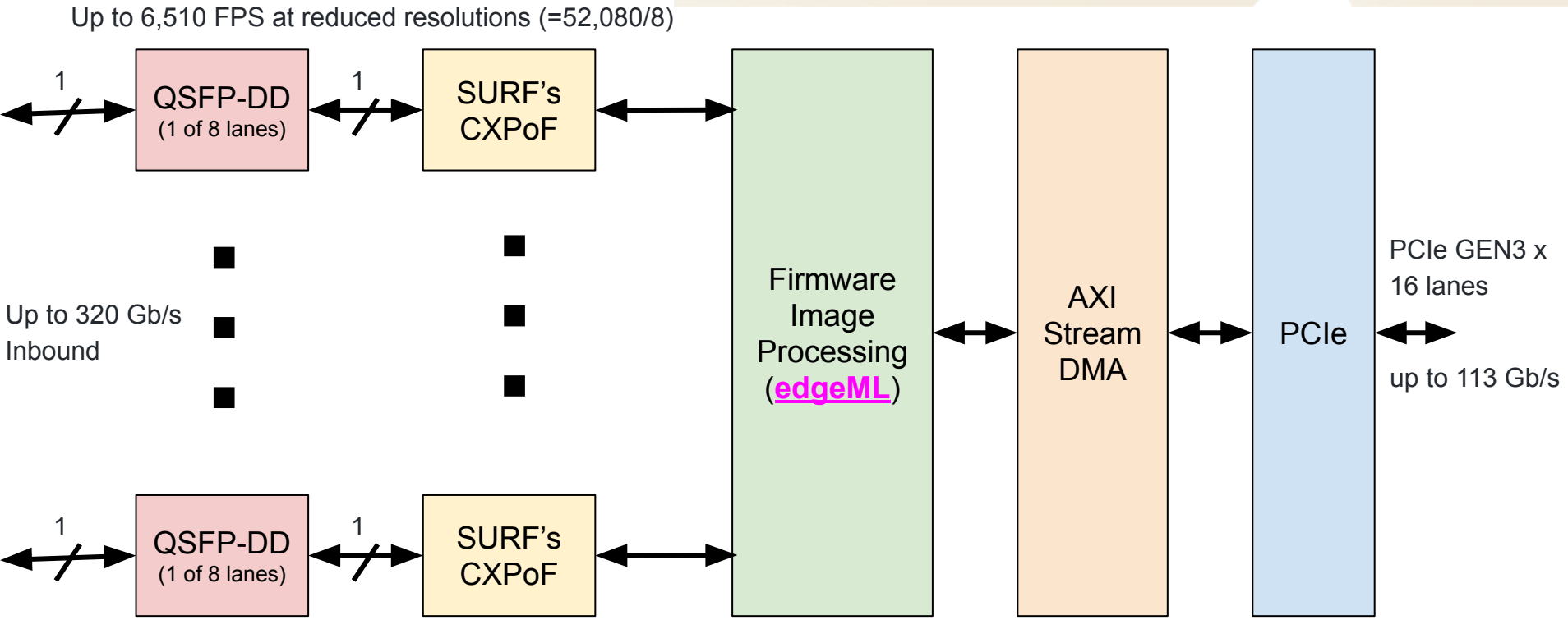
BittWare XUP-VV8

32 TX/RX Optical channels (Front Panel)

On-board 64 GB DDR4 Memory

PCIe GEN3 x 16 Lane

BittWare Card could support up to 32 units of S991 camera at 1 lane per Camera (1/8 rate per camera)



Our CXPoF firmware is open source
(refer to backup slide about SURF)

Our DMA firmware and kernel driver is open source
(refer to backup slide about "PCIe Framework")

Acknowledgement

- National Institute of Health
  - Time-of-Flight PET: (PI Levin) Proj. # 5R01EB023903-04

- Department of Energy, Office of Basic Energy Science
  - CookieBox detector: FWP 100498 (PI Coffee) “Enabling long wavelength Streaking for Attosecond X-ray Science”
  - EdgeAI for CookieBox: FWP 100643 (PI Thayer) “Actionable Information from Sensor to Data Center”
  - DOE High Energy Physics Detector R&D

- Department of Energy, Office of Fusion Energy Science
  - EdgeAI for Fusion Control: FWP 100636 (PIs Koleman, Coffee, Smith, Boyer, Schneider) “Machine Learning for Real-time Fusion Plasma Behavior Prediction and Manipulation.”

- SLAC-LCLS Program Development (PI Kling) “LCLS Growth”

- SLAC TID LDRD 21-007 (PI Herbst) “Edge ML for acquisition and analysis of data generated by ultra high rate detectors”

BACKUP SLIDES

BittWare Solution: COTS FPGA PCIe DAQ Card: XUP-VV8

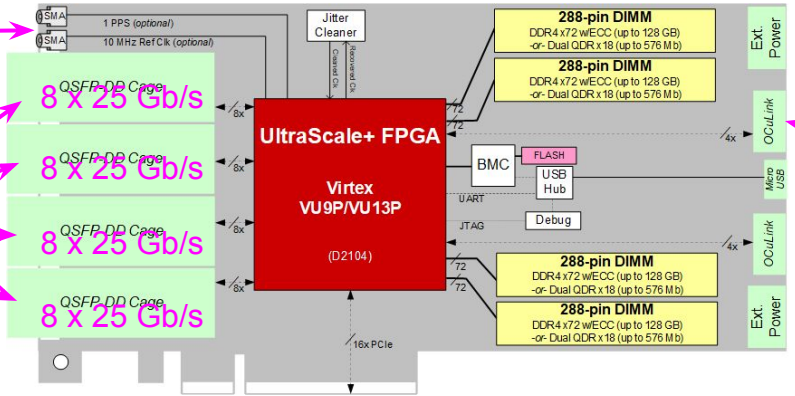
Passive Cooling ONLY

Device Name	VU13P
System Logic Cells (K)	3,780
CLB Flip-Flops (K)	3,456
CLB LUTs (K)	1,728
Max. Dist. RAM (Mb)	48.3
Total Block RAM (Mb)	94.5
UltraRAM (Mb)	360.0
DSP Slices	12,288

Type	Size	Speed (MTPS)
DDR4 RDIMM VLP	16GB	2400
DDR4 RDIMM	32GB	1866
DDR4 LRDIMM	32GB	2400
DDR4 LRDIMM	64GB	2133
DDR4 RDIMM	128GB	1600
QDR-II+ Q2XDM2	2x 288Mbit	550 MHz

GPS

800 Gb/s
total bandwidth

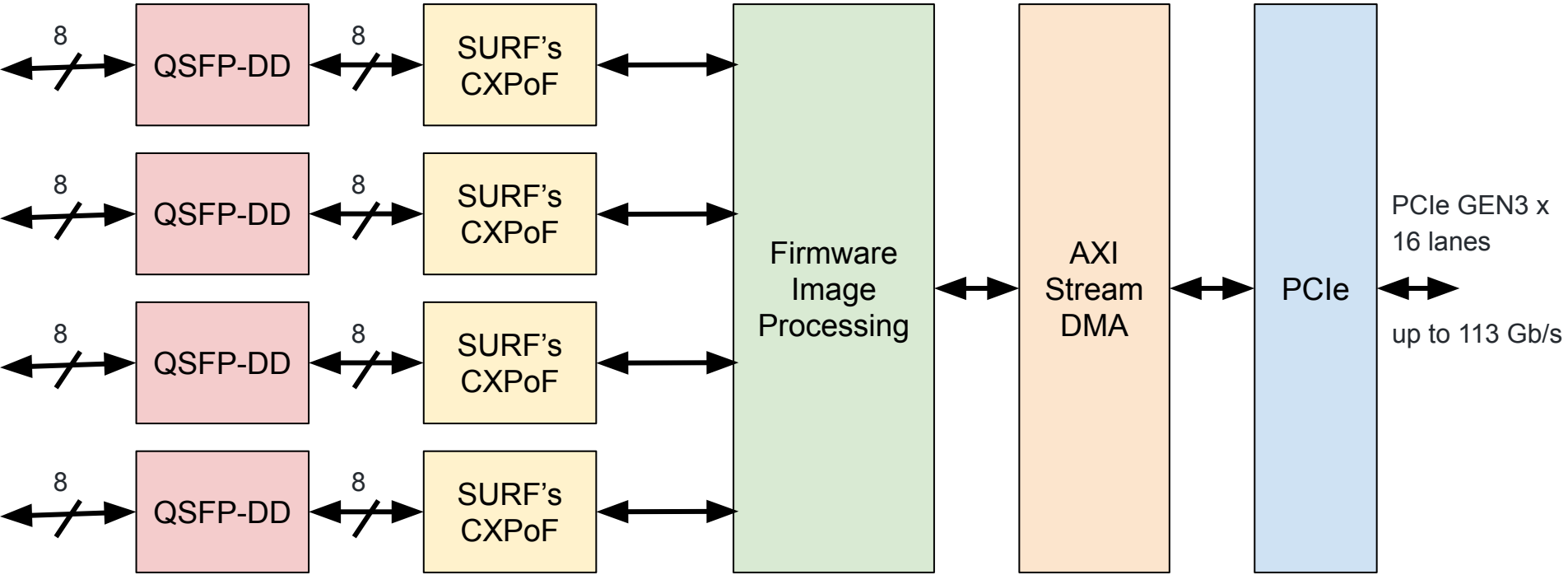


Rear Serial Expansion Ports:
200 Gb/s total bandwidth

PCIe GEN3 x 16 lanes

https://www.bittware.com/fpga/xup-vv8/

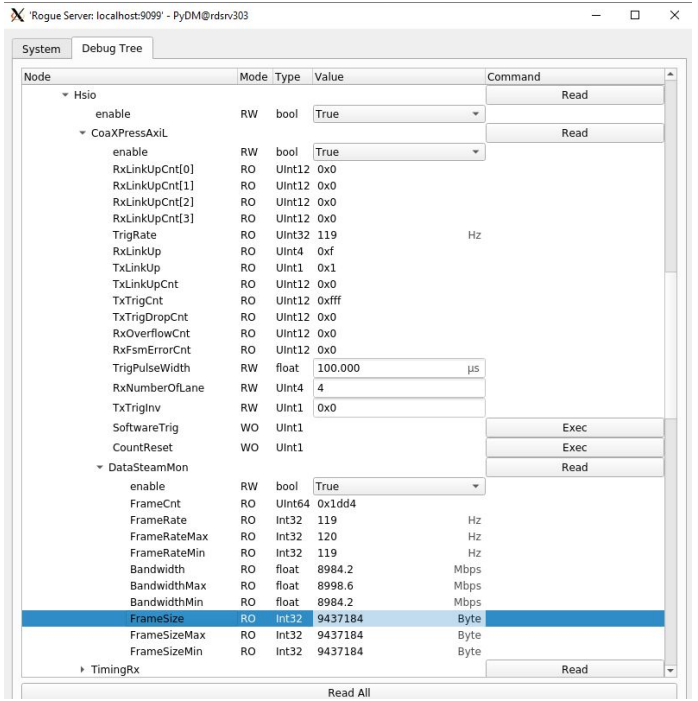
BittWare Card could support up to four S991 camera at 8 lane per Camera at FULL rate



Demonstration of custom CXPoF Framegrabber + S991 camera

- Demonstrated 120 Hz LCLS timing fiber triggering with the S991 camera
- 4096 pixels wide x 2304 pixels high in 8-bit pixel mode (9,437,184 Byte camera image)
- Tag Release (public domain):
  - https://github.com/slaclab/lcls2-coaxpress-over-fiber-apps/releases/tag/v1.0.0

Rogue PyQT GUI

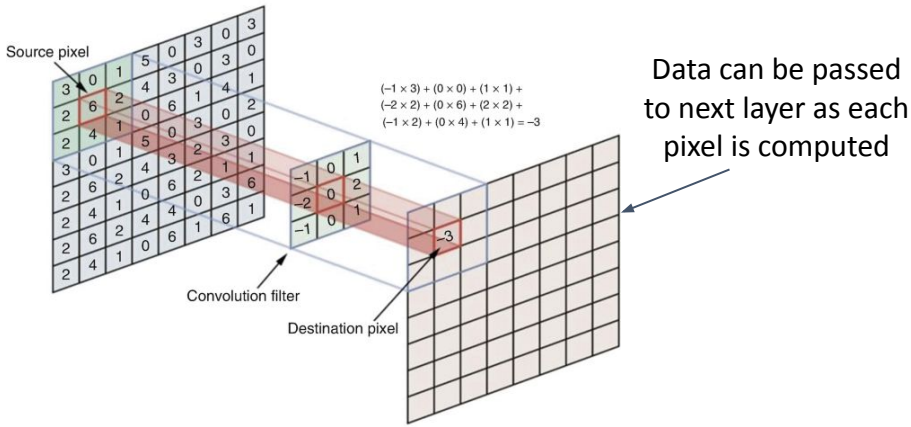


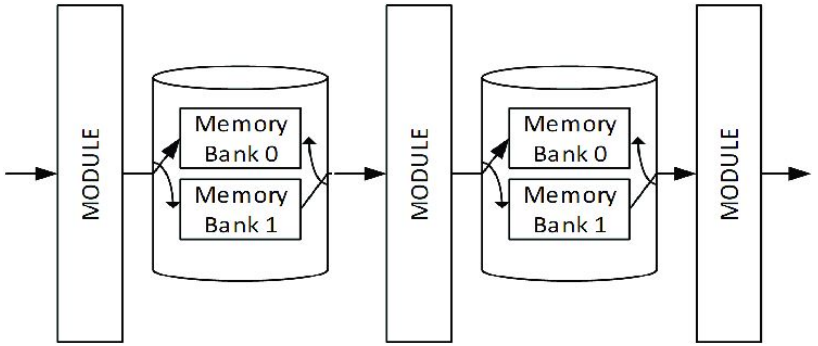
SNL Usage

- User defines layers using a **collection of C++ templates** that define each layer type and the associated activator for each layer
- Current layer types:
  - Conv2D
  - MaxPooling
  - AveragePooling
  - Dense
  - Reservoir
- Current activators:
  - LeakyRelu
  - Relu
- Each layer interface matches closely to the Keras model
  - Allows user to use **Keras documentation as a reference**
- Additional configurations are required to control the **FPGA specific configurations**

Why Streaming & pipelining?

- Certain layer types work better with streaming based upon their **data flow** model
  - Convolution (2D & 3D) and pooling layers can process **data as it arrives**
  - **Partial data** is ready as each region of the layer completes
  - Pass to next layer can begin when indices reach roughly ½ the kernel dimensions
- Other layers such as **Dense fully connected** need all of the input data to have arrived before output can begin… this suffers a **high latency penalty**





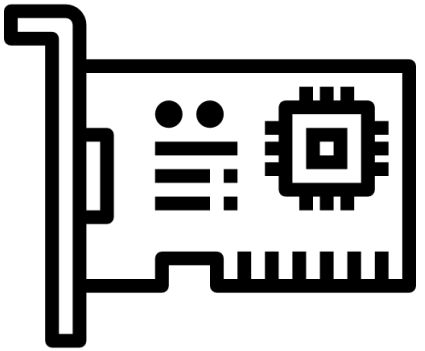
SLAC Ultimate RTL Framework (SURF) Firmware: Open source

- https://github.com/slaclab/surf
- HUGE VHDL library for FPGA development
- Used in Xilinx FPGAs, Intel FPGAs and ASIC digital designs
- VHDL-based IPs for commonly implemented modules
  - Ethernet Library: (IPv4, ARP, DHCP, ICMP, UDP)
  - AXI4 Library: (Crossbar, DMA, FIFO, etc.)
  - AXI4-Lite Library: (Crossbar, AXI4-to-AXI4-Lite bridge, etc.)
  - AXI4 stream Library: (DMA, MUX, FIFO, etc)
  - Device Library: (ADI, Micron, SiliconLabs, TI, etc.)
  - Synchronization Library: (Synchronize bits, buses, vectors, resets, etc)
  - Wrapped Xilinx Library: (clock managers, SEM, DNA, IPROG)
  - Serial Protocols Library: (I2C, SPI, UART, line-code, JESD204B, etc)
  - CXPoF Protocol: https://github.com/slaclab/surf/tree/master/protocols/coaxpress
- SURF is managed and maintained by TID-ID Electronics Systems Department
- New features and bug fixes on a weekly basis

Rogue Software (Open source)

- https://github.com/slaclab/rogue, https://slaclab.github.io/rogue/
- Software tools for both rapid prototyping and experiment deployment
- “Rogue” is not an acronym
- Operates either in a python/C++ hybrid or C++ only mode
  - Higher level python for connecting the high performance C++ modules togeth
- x86-64, ARM32 and ARM64 support
- Able to run on Linux, MAC or windows
  - Windows requires either WSL2 or Linux virtual machine
- Lots of ways to run the rogue software
  - 100% C++ only code
  - python script (non-GUI)
  - EPICS
  - Python GUI (e.g. PyQT, PyDM, etc)
- Rogue is managed and maintained by TID-ID Electronics Systems Department
- New features and bug fixes on a weekly basis

PCIe FW/SW Framework (Open source)



- https://github.com/slaclab/axi-pcie-core (Firmware)
- https://github.com/slaclab/aes-stream-drivers (Linux Kernel Driver)
- Firmware framework for BAR0 AXI-Lite interface with up to 8 DMA lanes
  - 256 TDEST per DMA lane (up to 2048 destinations total)
- Provide a “common platform” firmware frame and software kernel driver for any Xilinx PCIe card
  - Common Xilinx Dev board support
    - AC701, KC705, KCU105, KCU116, VCU128
  - Common Xilinx Data Center Card support
    - U50, U55C, U200, U250, U280, C1100
  - Much more card support can be added as needed
- Demonstrate up to 113 Gb/s for large frames (PCIe GEN3 x 16, 1MB frames)
- Demonstrate > 1MHz frame rate for small frame (<128B) without frame batching
- Managed and maintained by TID-ID Electronics Systems Department