Corundum + White Rabbit

Alex Forencich
3/22/2024

UC San Diego

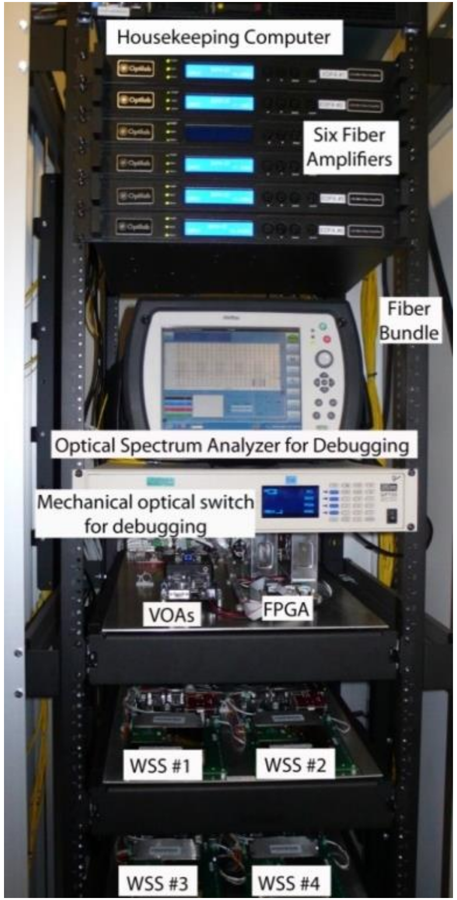
NSF

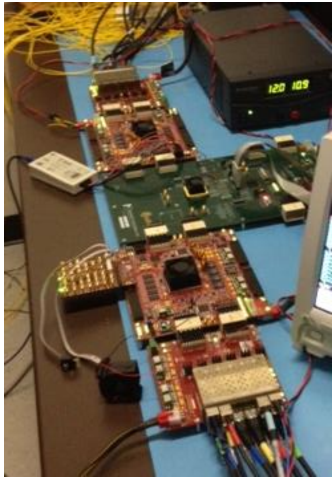
arpa·e
CHANGING WHAT'S POSSIBLE

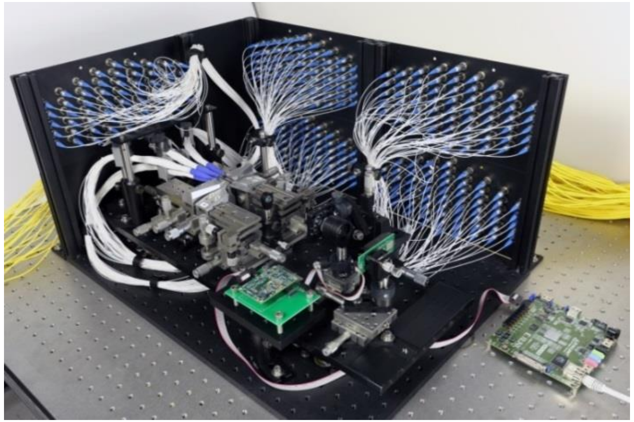
STATE OF CALIFORNIA
ENERGY COMMISSION

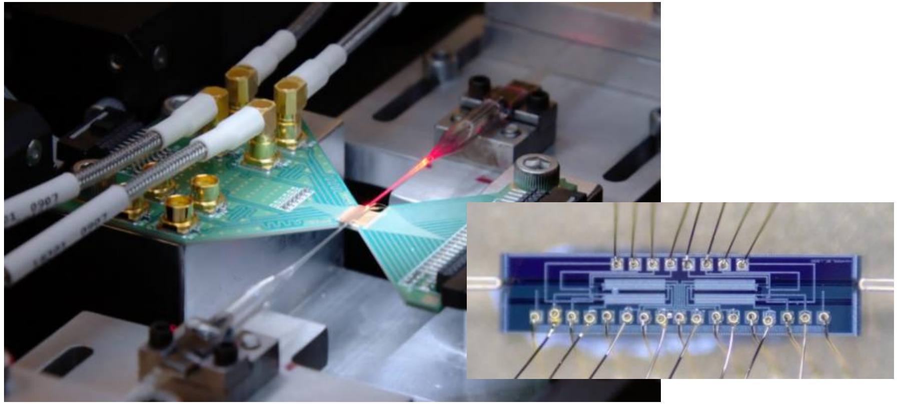
cns
Center for Networked Systems

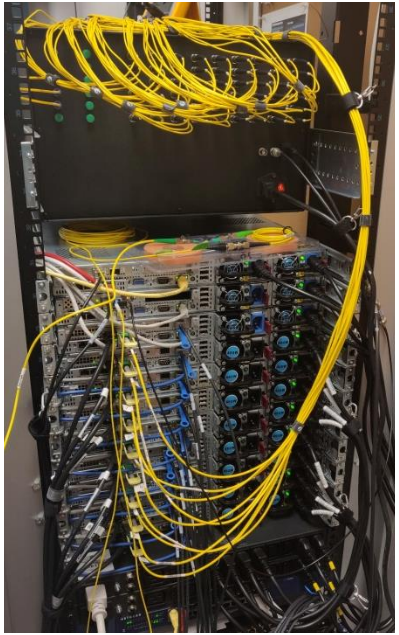
Circuit-Switching Research at UCSD





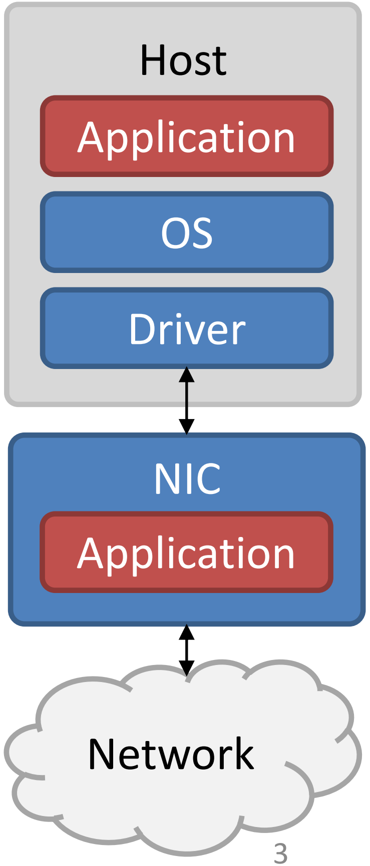






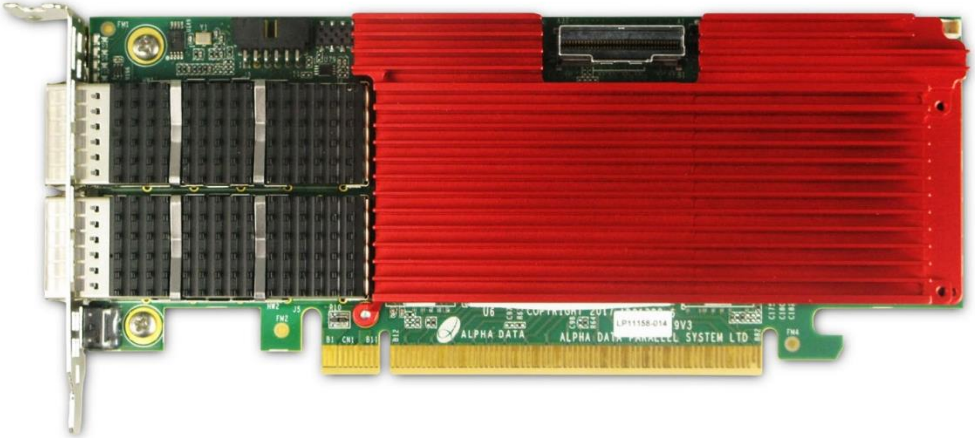
Introduction

- Network Interface Controller (NIC) connects software to the network
- NIC functionality is evolving
  - Line rate increases
  - Offload networking functions from CPU to NIC
- More general: in-network compute
  - Offload compute to programmable NICs, switches, etc.
  - Not limited to network stack



Corundum

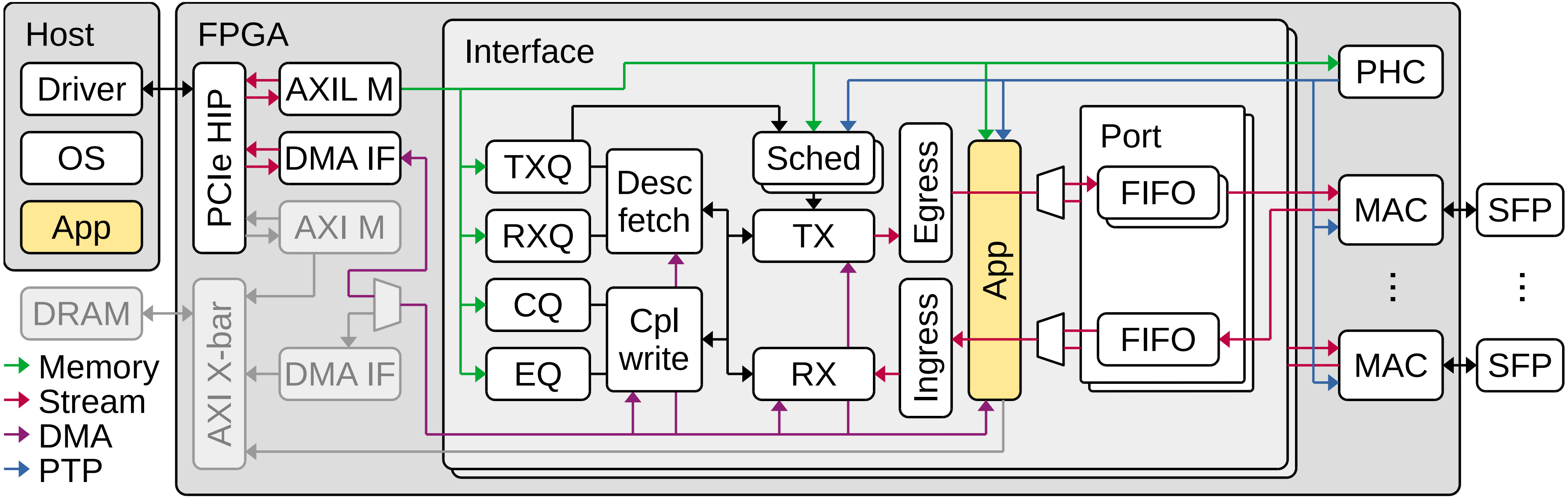
- Open-source, FPGA-based NIC and platform for in-network compute
  - A high performance "reference" NIC
  - Extensible: application block for implementation of custom features
  - Key applications: hardware prototyping of experimental networks/protocols, custom compute offload



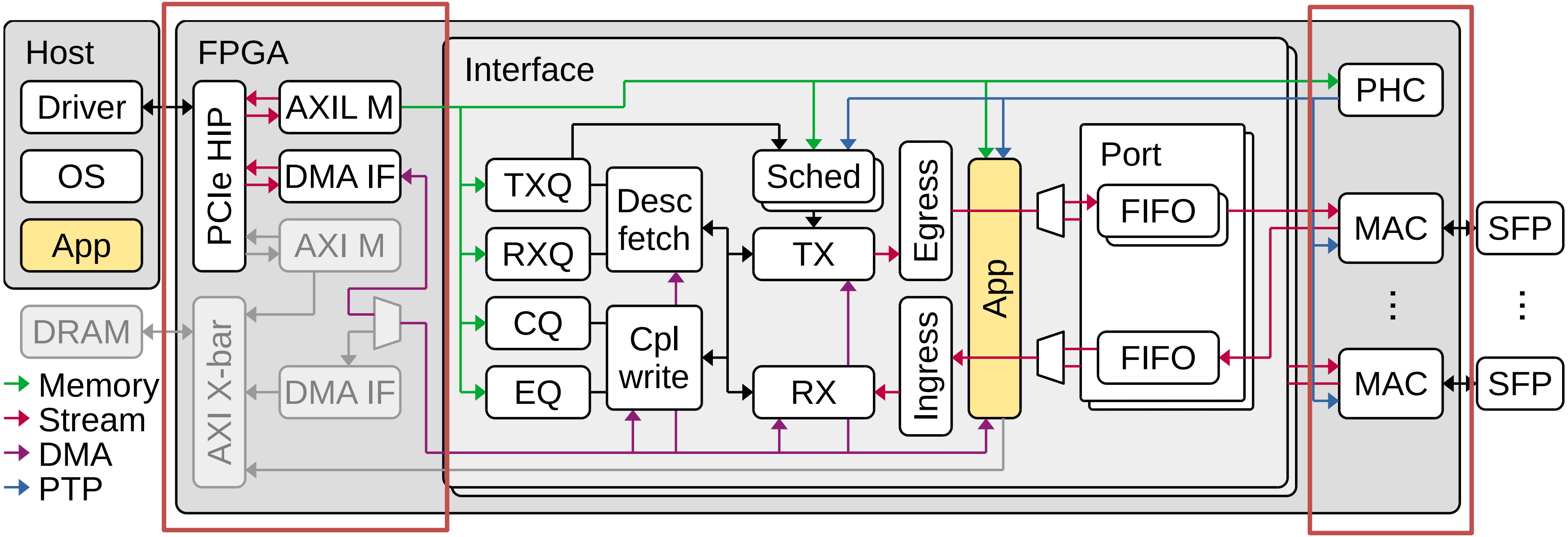
High-Level Features of Corundum

- Open-source, high-performance, FPGA-based NIC
  - PCIe gen 3 x16, multiple 10G/25G/100G Ethernet ports
  - Fully custom, high-performance DMA engine; Linux driver
- Application block for custom logic
  - Access to network traffic, DMA engine, on-card RAM, PTP time
- Fine-grained traffic control
  - 10,000+ hardware queues, customizable schedulers
- PTP timestamping and time synchronization
- Management features (FW update, etc.)
- Wide device support (AMD/Xilinx and Intel/Altera, PCIe and SoC)
- Source code: https://github.com/corundum/corundum

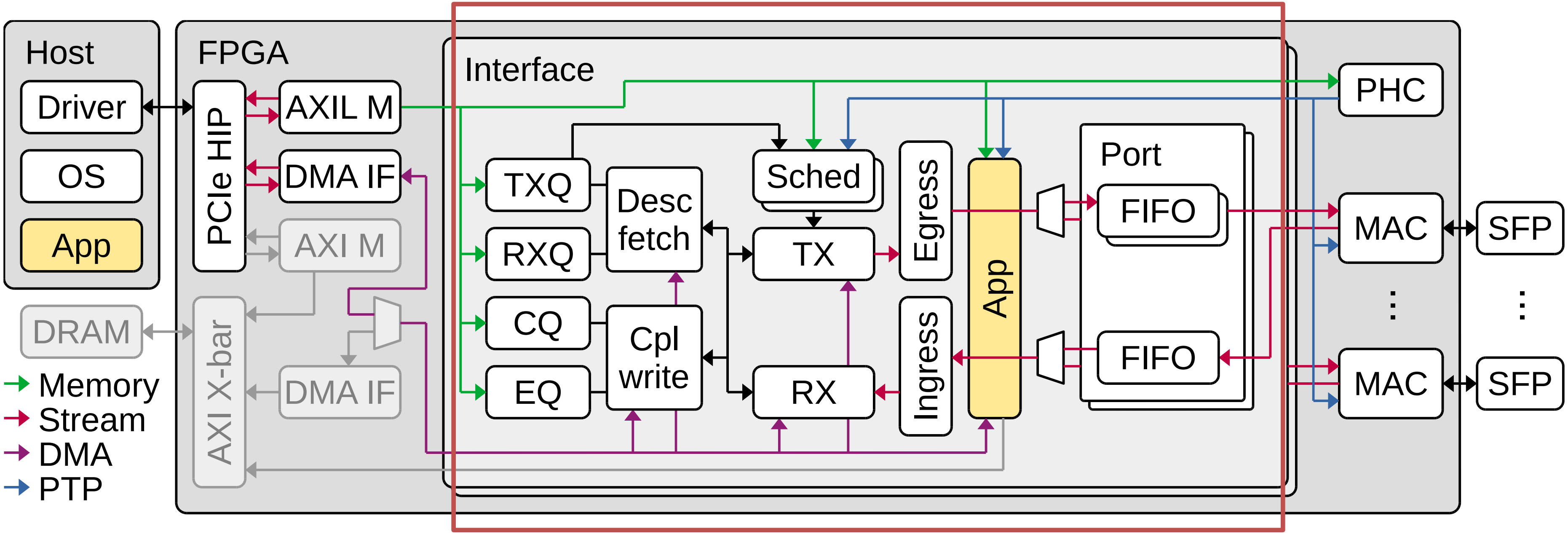
Corundum Block Diagram



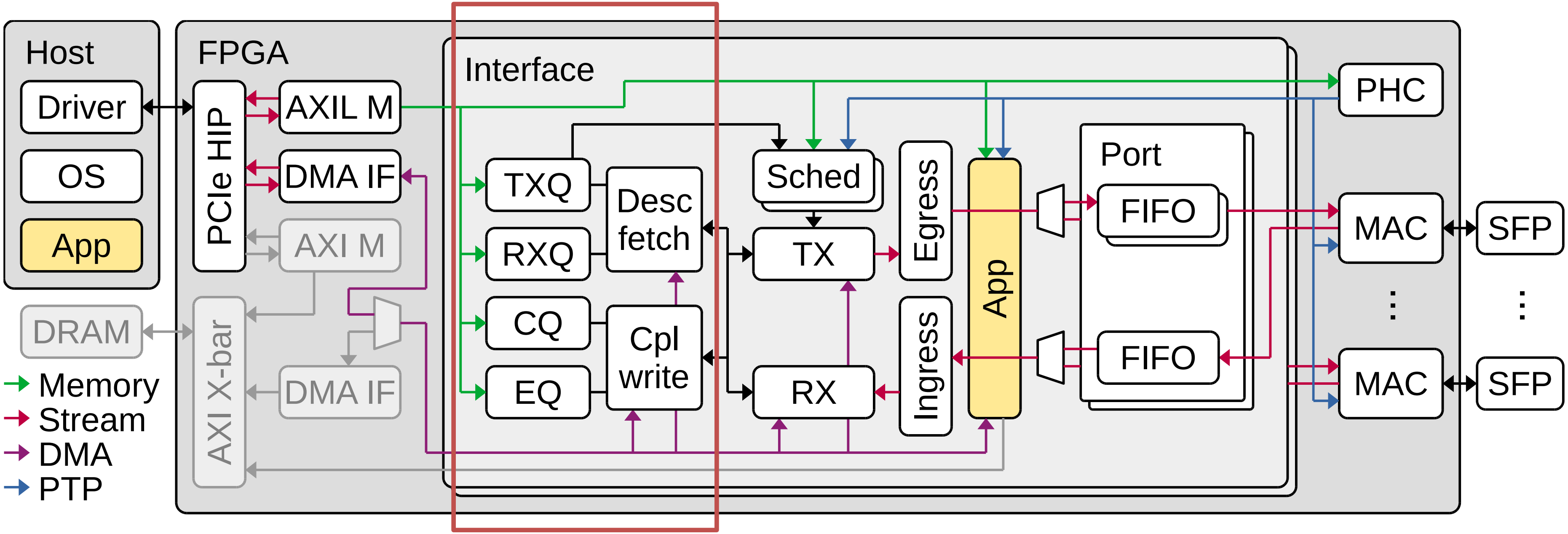
Corundum Block Diagram



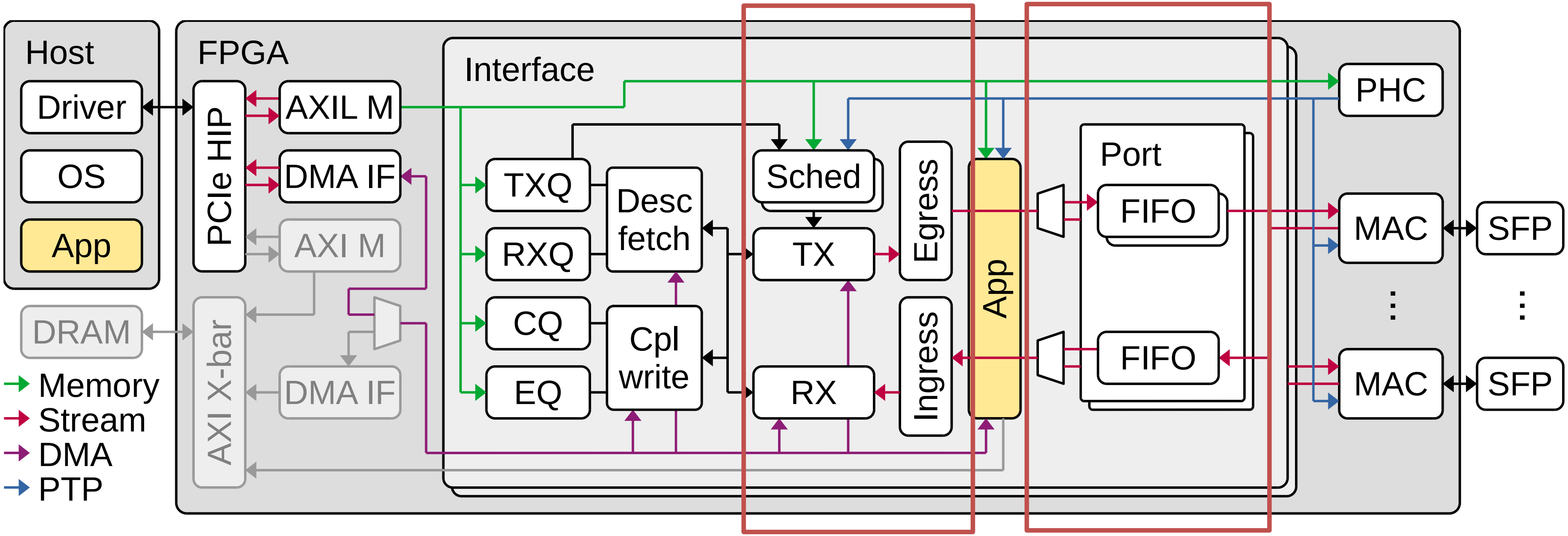
Corundum Block Diagram



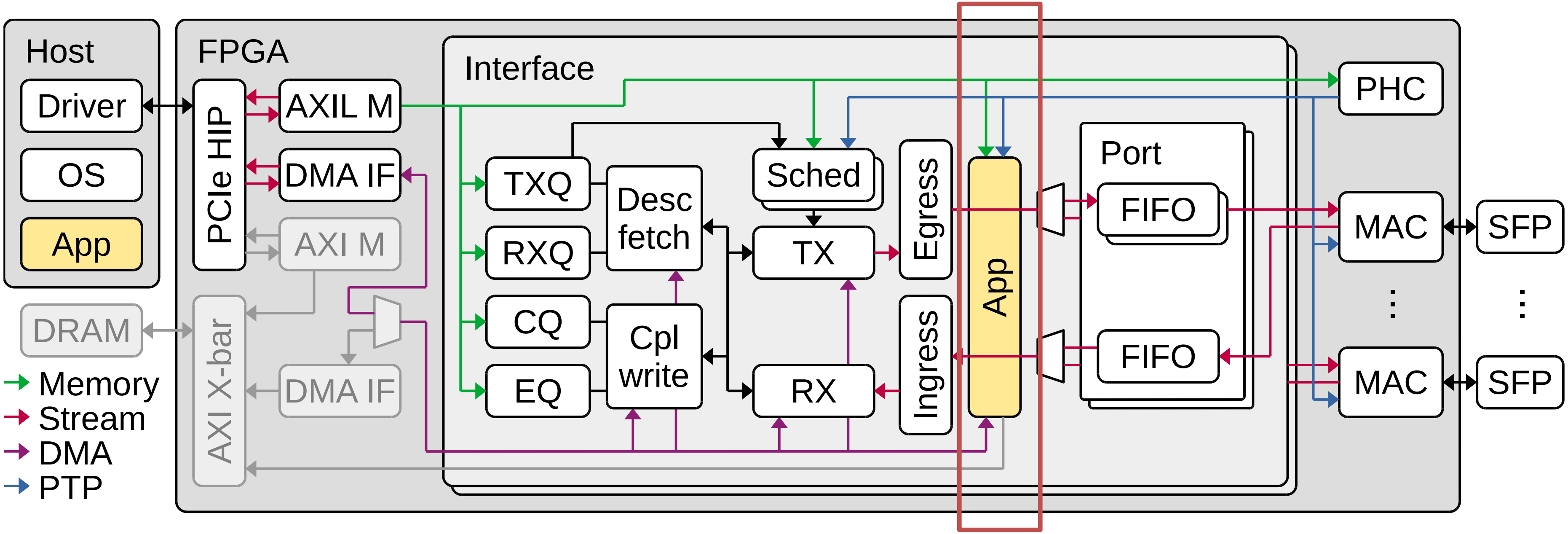
Corundum Block Diagram



Corundum Block Diagram



Corundum Block Diagram



→ Memory
→ Stream
→ DMA
→ PTP

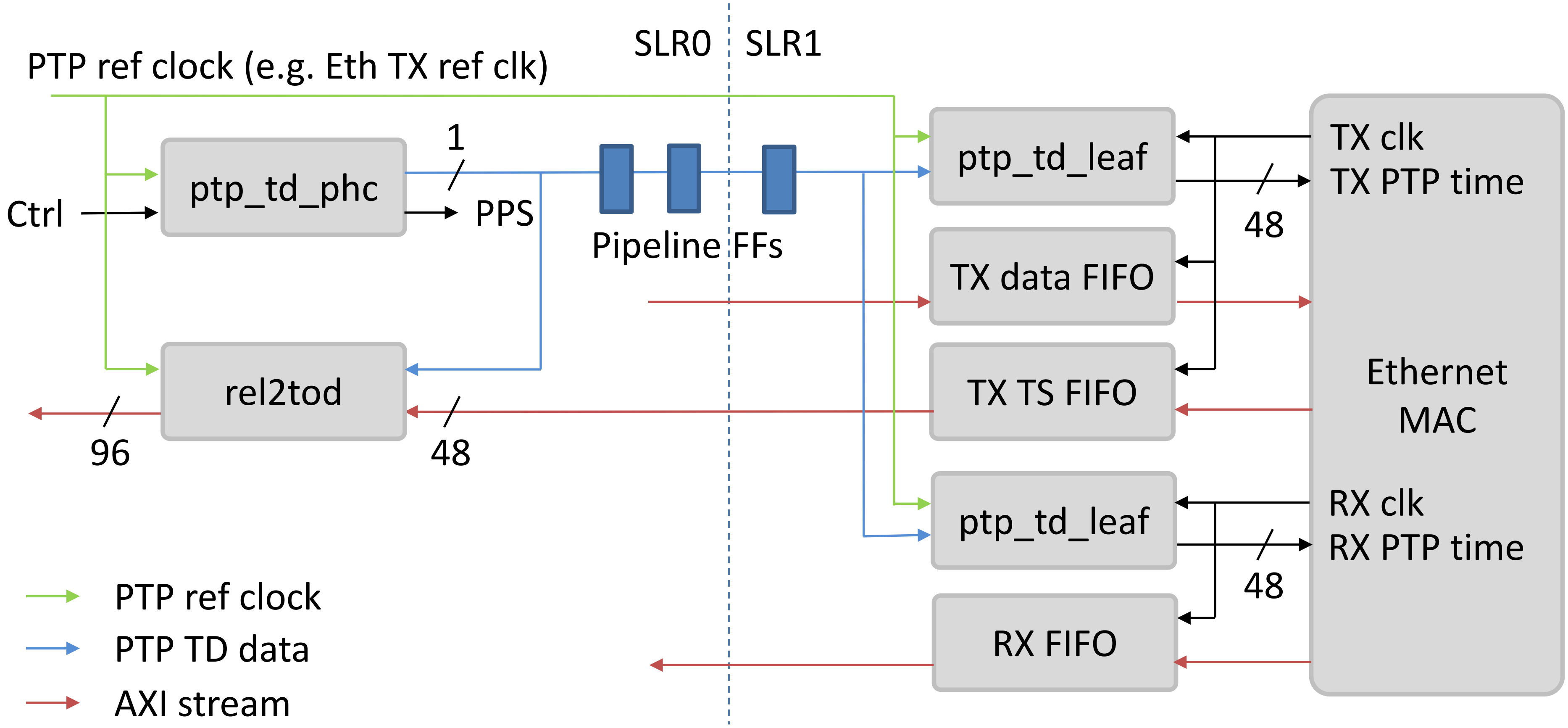
Fine-grained traffic control

- 10,000+ transmit queues
  - Each queue is an independent channel between SW and HW
  - Classify in SW, control in HW
  - Fine-grained, per-flow or per-destination control
  - 128 bits/queue -> 4096 queues in 2 URAM on US+
- Transmit scheduler
  - Determines which queue to transmit from
  - Default scheduler is round robin, but it can be replaced
  - Can be used to implement traffic shaping, rate limiting, etc.

PTP Time Distribution Subsystem

- Packet timestamping requires PTP time reference
  - Timestamping logic located near serdes and uses separate clock domains
  - Time from single PHC must be distributed across device to leaf clocks
- Serial protocol to distribute time from PHC
  - Single wire to reduce congestion
  - Protocol supports use of pipeline registers to cover long distances
- ToD timestamp derived from relative timestamp
  - Reduce logic resources by using truncated 32.16 relative timestamps

PTP Time Distribution Subsystem



Corundum and White Rabbit

- Integrate WR functionality directly into Corundum core logic
  - Likely will need to rework some of the PTP TD, MAC, and PCS logic
  - Also need some sort of timing I/O subsystem
- Should be able to “easily” add support for quite a few boards
  - Corundum currently supports ~30 boards spanning multiple board vendors and device families

WR device support

- Serdes, PHY, and MAC configuration is specific to device family
- WR requires deterministic latency and precision timestamping
  - Mitigate latency variance in serdes and gearbox/PCS/MAC/EMIB
  - Hard MAC timestamping must be correct (CMAC, E/F tiles, etc.)
- AMD/Xilinx GTX/GTH/GTY should work well
  - Used by current WR switch and other WR hardware
- Other hardware will require characterization

WR board support

- FPGA is part of the picture, board-level clocking is the rest
- White rabbit requires tunable Ethernet reference clock and "helper" clock with small offset for DDMTD
  - Original WR hardware uses two VCOs + DACs
- Helper clock can potentially be generated by (ab)using internal PLLs
- Ethernet reference clock can be provided by VCO, DCO, or Fractional-N PLL
  - DCOs and Frac-N PLLs are actually rather common (Si570, Si5341, etc.)

WR board support

- Corundum currently supports ~30 different FPGA boards
- Board clocking configurations fall into 3 general categories

ADM-PCIE-9V3	K35-S	K3P-S	K3P-Q	fb2CG@KU15P
fb4CGg3	SUME	250-SoC	XUPP3R	XUSP3S
520N-MX	IA-420F	S10MX DK	S10DX DK	Agilex F DK
Agilex I DK	DE10-Agilex	Alveo U45N	Alveo U50	Alveo U55C
Alveo U55N	Alveo U200	Alveo U250	Alveo U280	KR260
VCU108	VCU118	VCU1525	ZCU102	ZCU106

WR board support

- Boards with insufficiently tunable oscillator
  - Fixed osc, integer-N PLL, etc.

ADM-PCIE-9V3	K35-S	K3P-S	K3P-Q	fb2CG@KU15P
fb4CGg3	SUME	250-SoC	XUPP3R	XUSP3S
520N-MX	IA-420F	S10MX DK	S10DX DK	Agilex F DK
Agilex I DK	DE10-Agilex	Alveo U45N	Alveo U50	Alveo U55C
Alveo U55N	Alveo U200	Alveo U250	Alveo U280	KR260
VCU108	VCU118	VCU1525	ZCU102	ZCU106

WR board support

- Boards with tunable oscillator behind BMC
  - May need to modify BMC firmware to support tuning

ADM-PCIE-9V3	K35-S	K3P-S	K3P-Q	fb2CG@KU15P
fb4CGg3	SUME	250-SoC	XUPP3R	XUSP3S
520N-MX	IA-420F	S10MX DK	S10DX DK	Agilex F DK
Agilex I DK	DE10-Agilex	Alveo U45N	Alveo U50	Alveo U55C
Alveo U55N	Alveo U200	Alveo U250	Alveo U280	KR260
VCU108	VCU118	VCU1525	ZCU102	ZCU106

WR board support

- Boards with directly-connected tunable oscillator
  - Clocking network ready for white rabbit

ADM-PCIE-9V3	K35-S	K3P-S	K3P-Q	fb2CG@KU15P
fb4CGg3	SUME	250-SoC	XUPP3R	XUSP3S
520N-MX	IA-420F	S10MX DK	S10DX DK	Agilex F DK
Agilex I DK	DE10-Agilex	Alveo U45N	Alveo U50	Alveo U55C
Alveo U55N	Alveo U200	Alveo U250	Alveo U280	KR260
VCU108	VCU118	VCU1525	ZCU102	ZCU106

Corundum + WR Status

- Working on high-level architecture
  - Need to handle multiple PCS clock frequencies
  - Likely need to significantly rework PTP CDC logic and MAC+PCS logic
- OCP Time Appliances Project White Rabbit NIC
  - Goal is to build a relatively low-cost open-source white rabbit NIC
  - Custom PCIe form-factor carrier board for Xilinx Kria K26 SoM
  - Renesas/IDT 8A34002 PLL
  - “Stock” corundum operating on initial hardware
  - Eventual goal is to support WR + PTM

Thank you

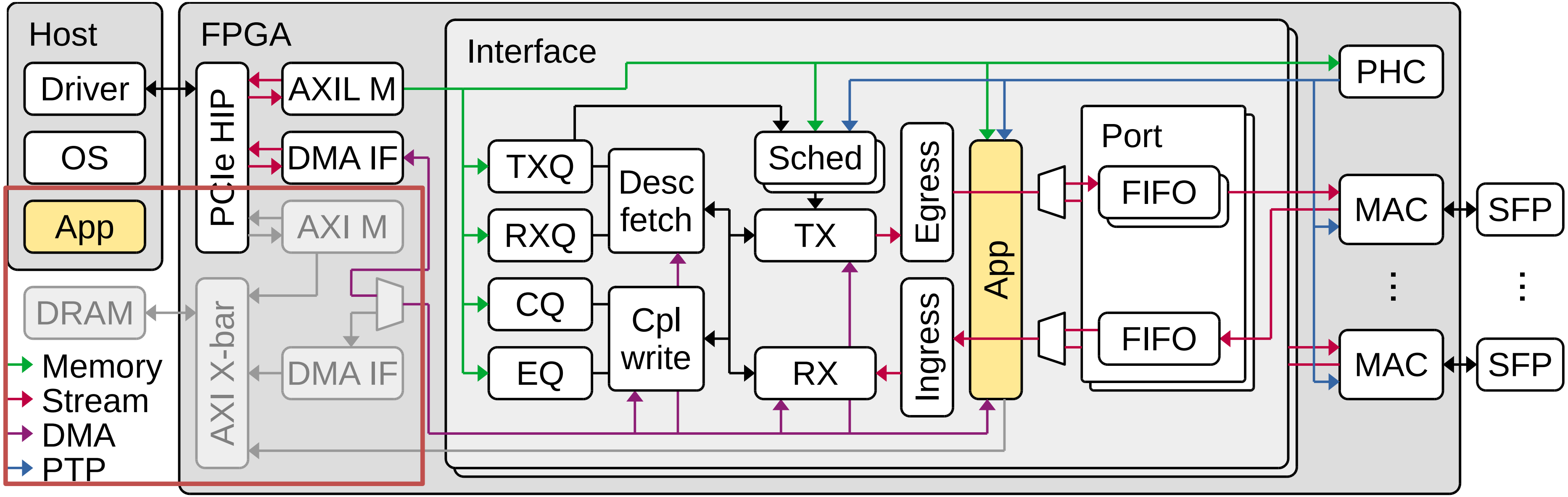
Corundum source code available on GitHub:

https://github.com/corundum/corundum

Contact me at: jforenci@ucsd.edu

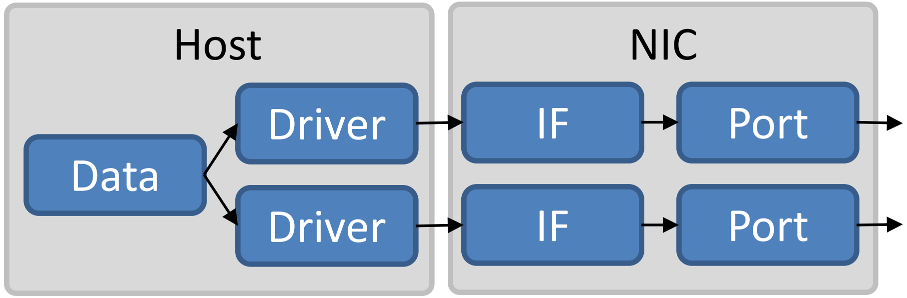
Backup slides

Corundum Block Diagram

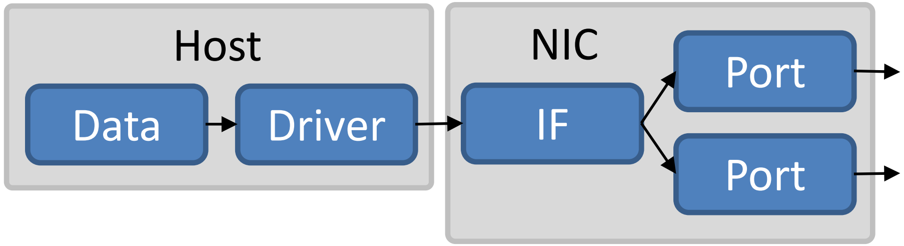


Ports and Interfaces

- Hardware support for multiple uplinks
- Multiple physical ports can appear as single OS-level interface
- Ports have separate schedulers
- Migrate or stripe flows across ports by changing scheduler settings



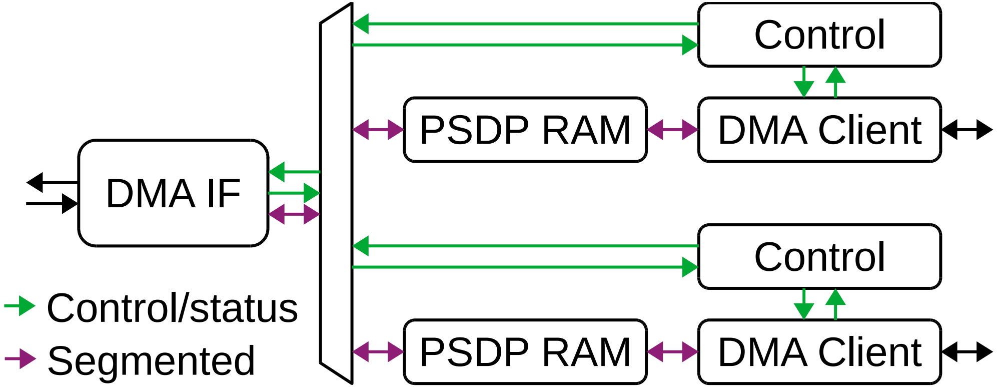
Traditional NIC: assignment in software



Corundum NIC: assignment in hardware

Modular DMA engine

- DMA engine split between interface and client modules
  - Interface connects to host – PCIe, AXI, etc.
  - Client modules form internal ports – AXI stream, memory-mapped AXI
- Clients connected to interface with dual port RAMs
- Support both servers (PCIe) and SoCs (AXI) with same core logic

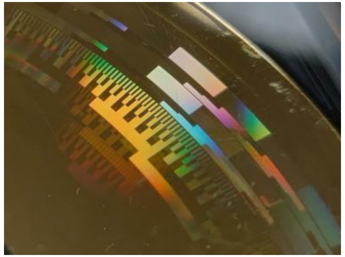


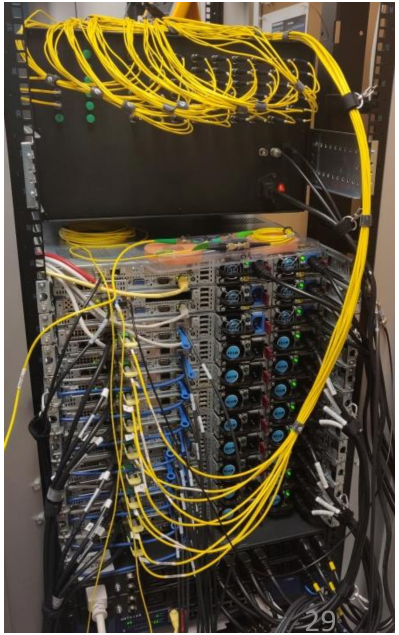
Applications

- Offload application-specific processing
- Datapath for novel transmit schedulers
- Instrument Corundum for performance measurements
- Direct transceiver access permits physical-layer measurements and development of new wire protocols
- Use core logic as a packet DMA engine in a larger system
- Discuss two applications:
  - TDMA for microsecond circuit switching
  - PHY layer BER measurement for link characterization

Application: TDMA

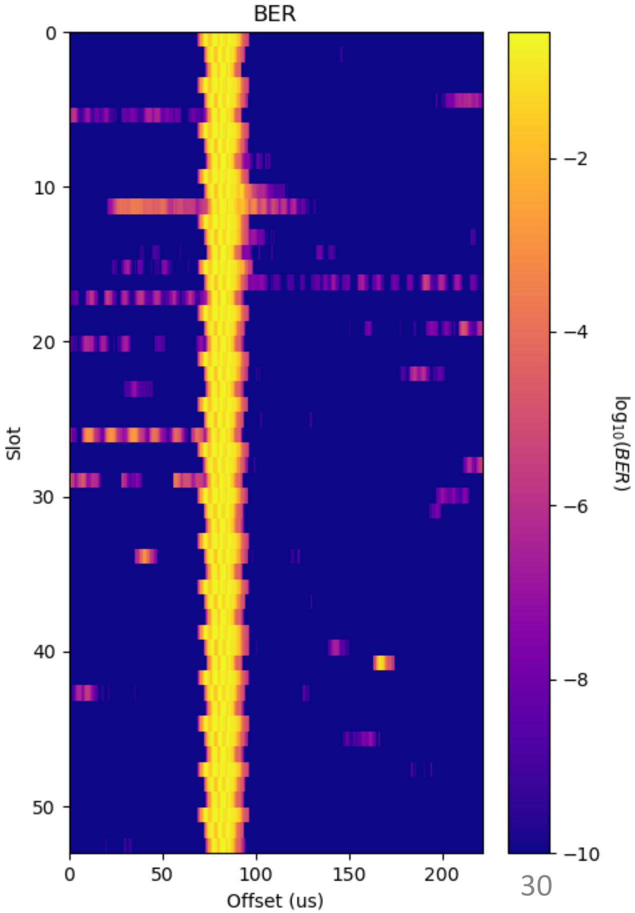
- Scheduler can control queues based on PTP time
  - Enables sub-microsecond-resolution TDMA
- TDMA off
  - 94 Gbps
- 200 us period, 50% duty cycle TDMA schedule
  - Guard time 2 us
- Using TDMA schedule, can run iperf through pinwheel switch
  - Initial test with old FW: 3 Gbps on 10 Gbps link with low packet loss





Application: link-level characterization

- *In situ* link-level measurements
- BER measurement capability integrated into NIC
- Measure link-level performance from vantage point of every NIC in datacenter
- Supports time-domain BER
  - Synchronized over network via PTP
  - Measure every path through switch
- Heat map represents signal at one receiver through pinwheel switch



Industry support and adoption

- Axbryd
  - Hardware eBPF offloading (hXDP) built on top of Corundum
- Missing Link Electronics
  - Building a product using Corundum
  - Ported Corundum to Zynq MPSoC
  - Working on Stratix 10 GX port
  - Developing DPDK driver

Axbryd
Building a lighter world

mle
missing link electronics

Long-term goals

- Core features
  - RDMA support in core datapath
  - Variable-length descriptor support
  - Unified DMA address space (on-card DRAM/HBM)
  - Embedded packet switch, SR-IOV, white rabbit
- Device and board support
  - Improve Intel device support
  - Move board-dependent code from driver to soft core
  - Simplify porting process
- Software
  - Improved interface to application logic
  - DPDK driver