Specialized Processing for Edge Computing

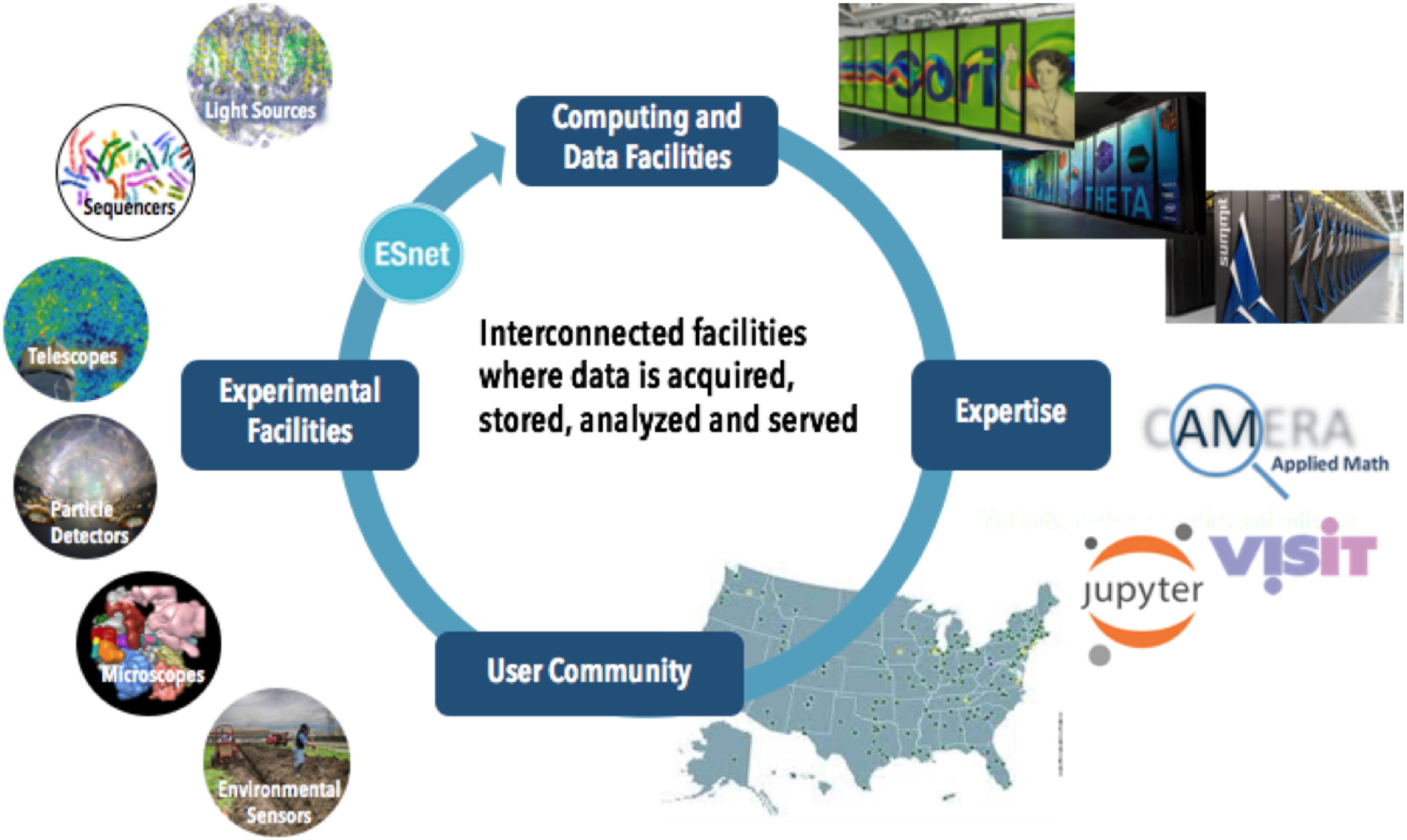
Bringing the computation to the data

David Donofrio
presented by John Shalf

June 2019

Coupling of Data and Experiments

Opportunities for specialized processing throughout the Superfacility Model



Computing and Data Facilities

ESnet

Experimental Facilities

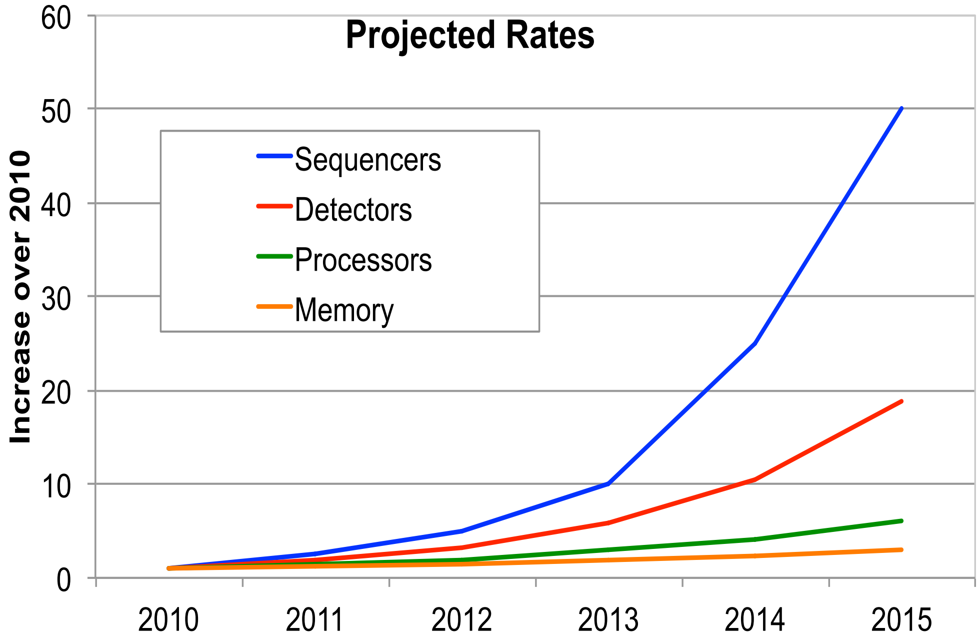
Interconnected facilities where data is acquired, stored, analyzed and served

Expertise

User Community

On-detector processing

Putting hardware specialization to work to augment existing HPC resources

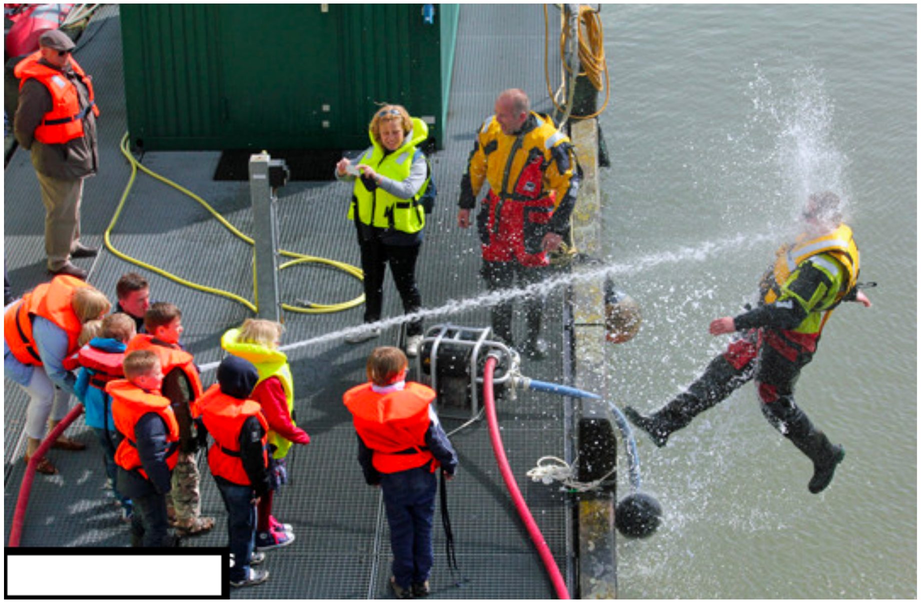


The Problem:

- Future detectors threaten to overwhelm data transfer and computing capabilities w/ data rates exceeding 1 Tb/s
- Data processing experiment driven

Proposed solution:

- Process the data *before it leaves the sensor*
- Application-tailored, programmable processing allows data reduction to occur on-sensor
- Programmability allows data reduction techniques to be tailored to the experiment – even *after* the sensor is built!



Bringing the Processing to the Data: Hardware for Data Analysis and Reduction

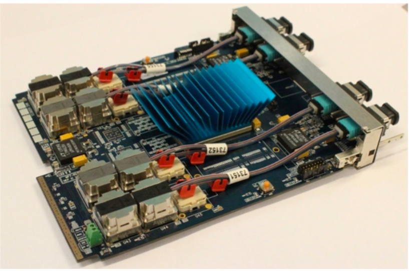
Augment HPC facilities to handle increases in EOS

On Sensor / Field Deployable Processing

Leverage our architectural exploration and design tools to design custom, programmable logic to be integrated on existing sensors or to act as independent, field deployable computing.

Can be used in a facility or act as standalone, low-power field deployed unit

Near Sensor and Real Time Processing



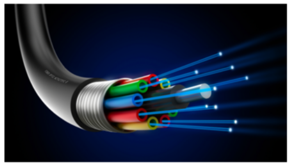
Custom logic FPGAs or ASIC) can be placed near the sensor to analyze and reduce data in real time as it is produced.

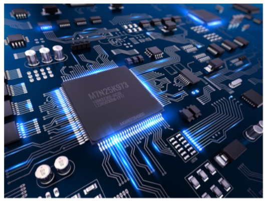
The same logic can be used for automated, real time control of instruments

Example: LHC Triggers or control processor for Quantum Processor

ESnet

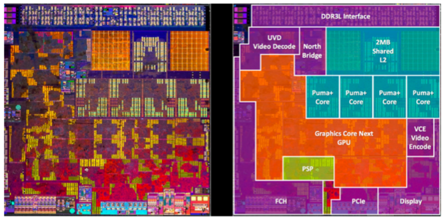
Smart HPC Interconnects





Embedding hardware within the network may allow better utilization of existing HPC interconnects

HPC Specialized Accelerators



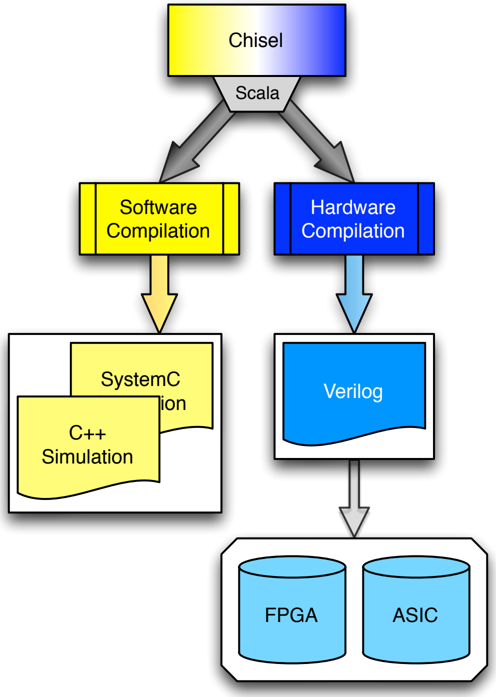
Specialized accelerators incorporated into HPC systems. Could be integrated onto an SoC or discrete compute elements

Require advances in programming and execution models

Hardware Generators: *Enabling Technology for Exploring Design Space Together with Close Collaborations with Applied Mathematics*

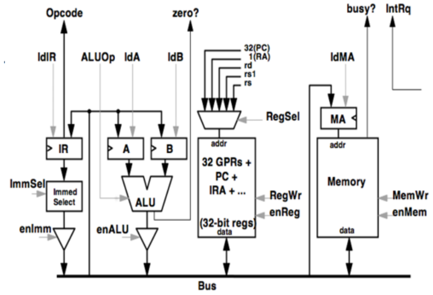
Chisel

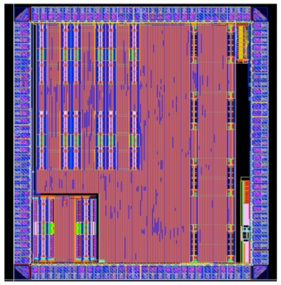
DSL for rapid prototyping of circuits, systems, and arch simulator components

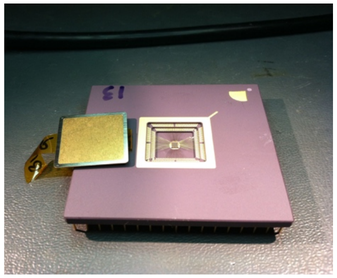


RISC-V

Open Source Extensible ISA/Cores

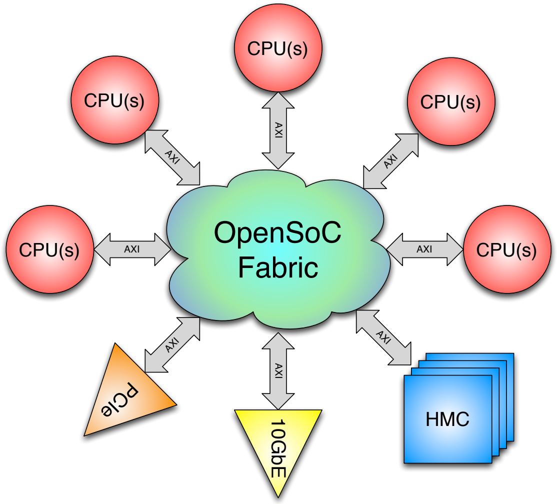






OpenSOC

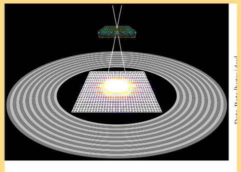
Open Source fabric
To integrate accelerators
And logic into SOC



SuperTools
Superconducting
RISC-V

QUASAR
Quantum
ISA

Multiagency
Architecture
Exploration

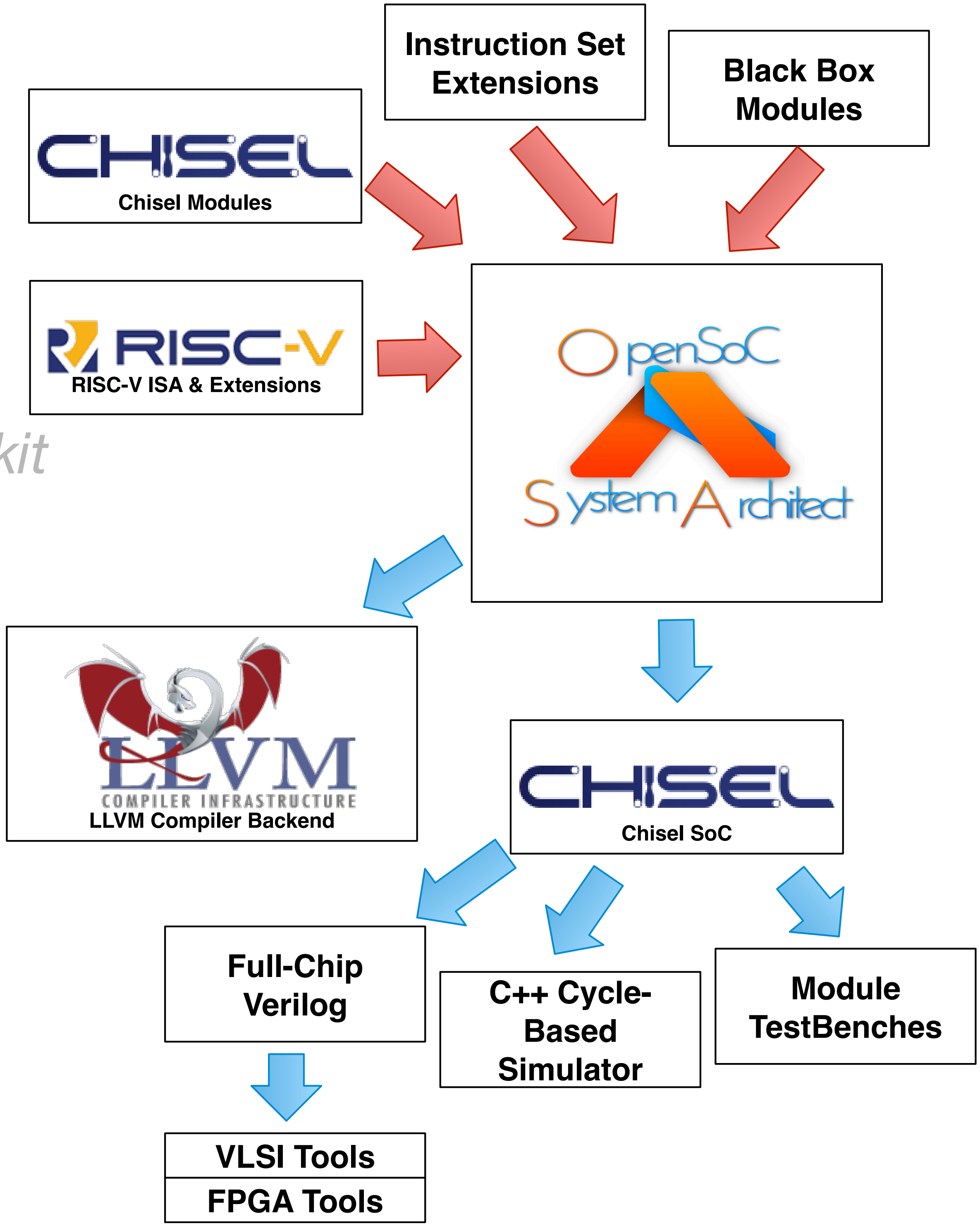


Active
Sensors
(CryoEM)

Existing projects using LBL Open Source Hardware Generators

OpenSoC System Architect

A complete hardware and software development toolkit



Chisel Modules

Instruction Set Extensions

Black Box Modules

RISC-V ISA & Extensions

LLVM Compiler Backend

Chisel SoC

Full-Chip Verilog

C++ Cycle-Based Simulator

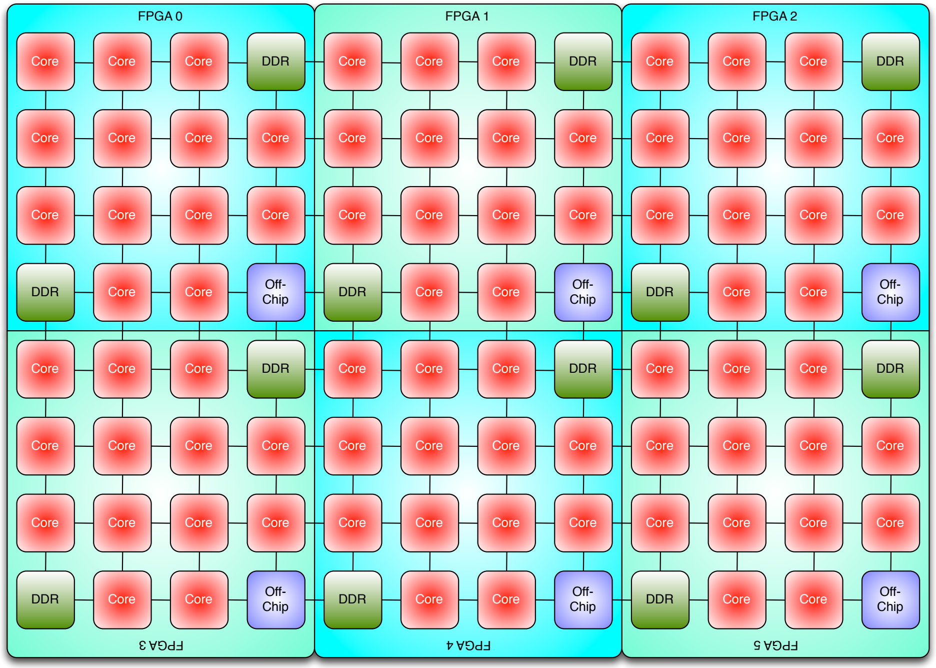
Module TestBenches

VLSI Tools

FPGA Tools

Demo: 96 Core SoC Design for HPC

SC2016 Demo

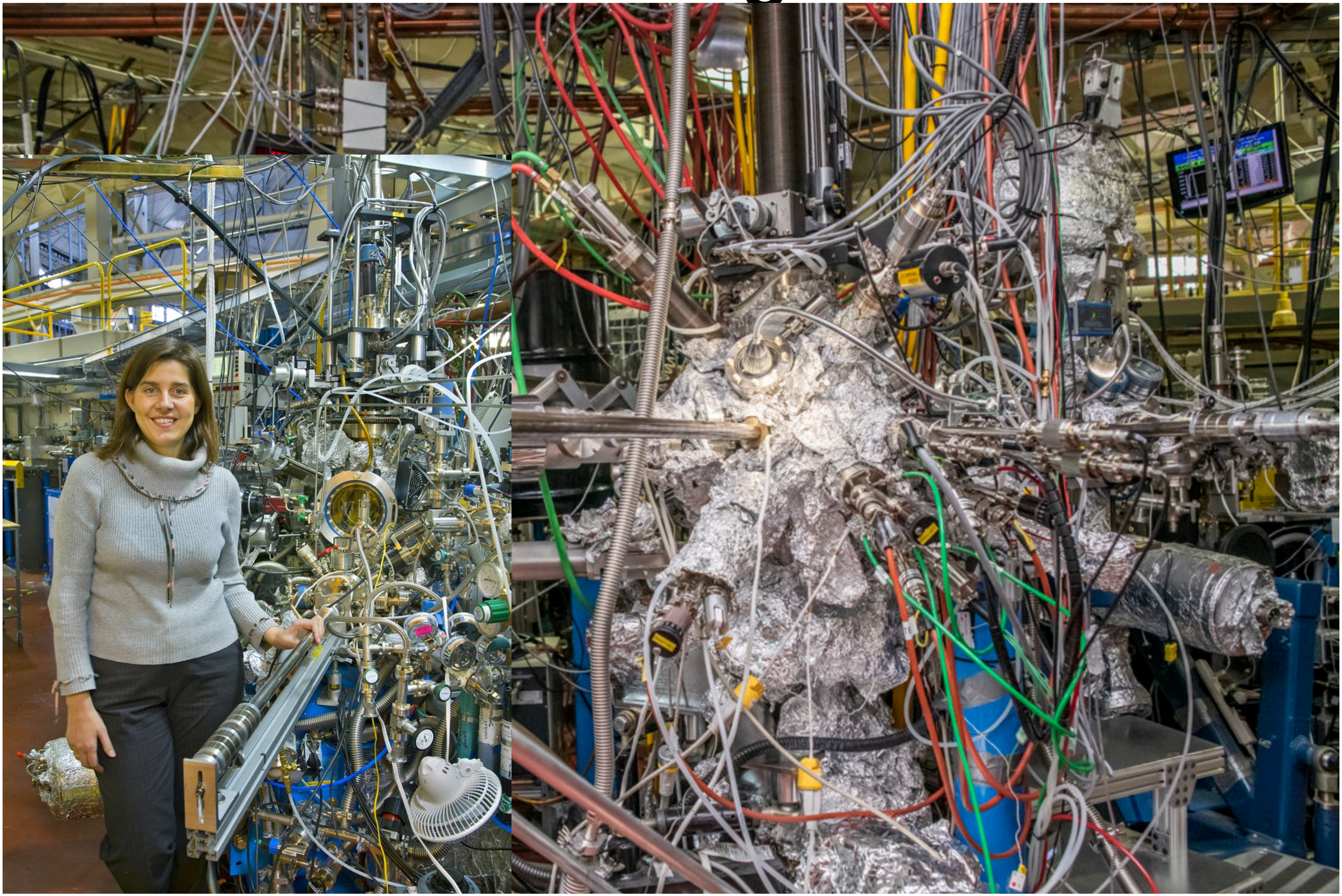

FPGA 0
FPGA 1
FPGA 2
Core Core Core DDR Core Core Core DDR Core Core Core DDR
Core Core Core Core Core Core Core Core Core Core Core Core
Core Core Core Core Core Core Core Core Core Core Core Core
DDR Core Core Off-Chip DDR Core Core Off-Chip DDR Core Core Off-Chip
Core Core Core DDR Core Core Core DDR Core Core Core DDR
Core Core Core Core Core Core Core Core Core Core Core Core
Core Core Core Core Core Core Core Core Core Core Core Core
DDR Core Core Off-Chip DDR Core Core Off-Chip DDR Core Core Off-Chip
FPGA 3
FPGA 4
FPGA 5

2 people spent 2 months to create

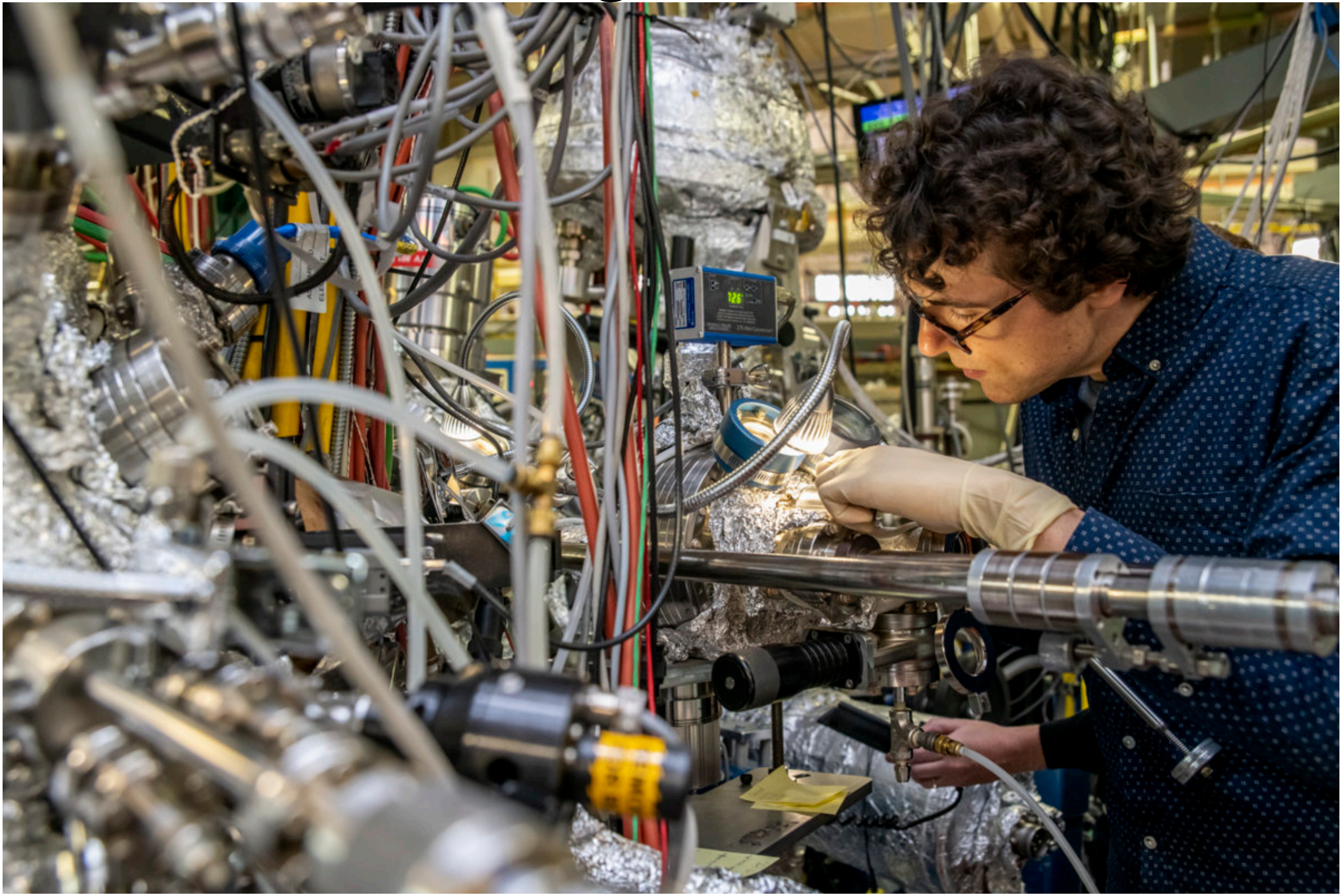
- **Z-Scale processors connected in a Concentrated Mesh**
- **4 Z-scale processors**
- **2x2 Concentrated mesh with 2 virtual channels**
- **Micron HMC Memory**

http://www.codexhpc.org/?p=367

This is What the Edge Looks Like

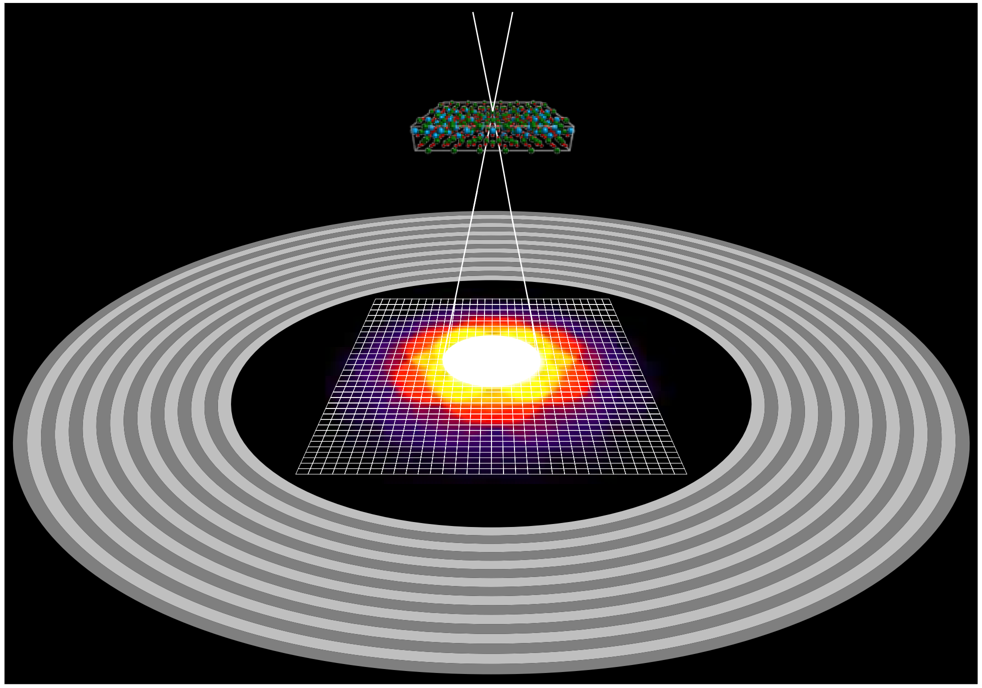


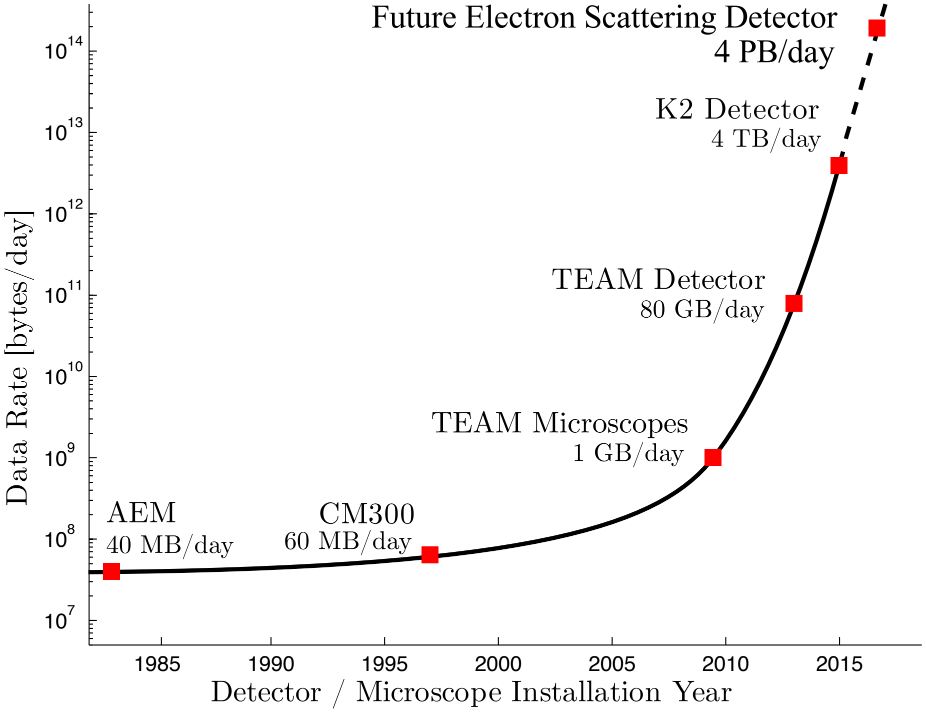
This is “The Edge”



National Center for Electron Microscopy (NCEM)

Future Electron Scattering Detector

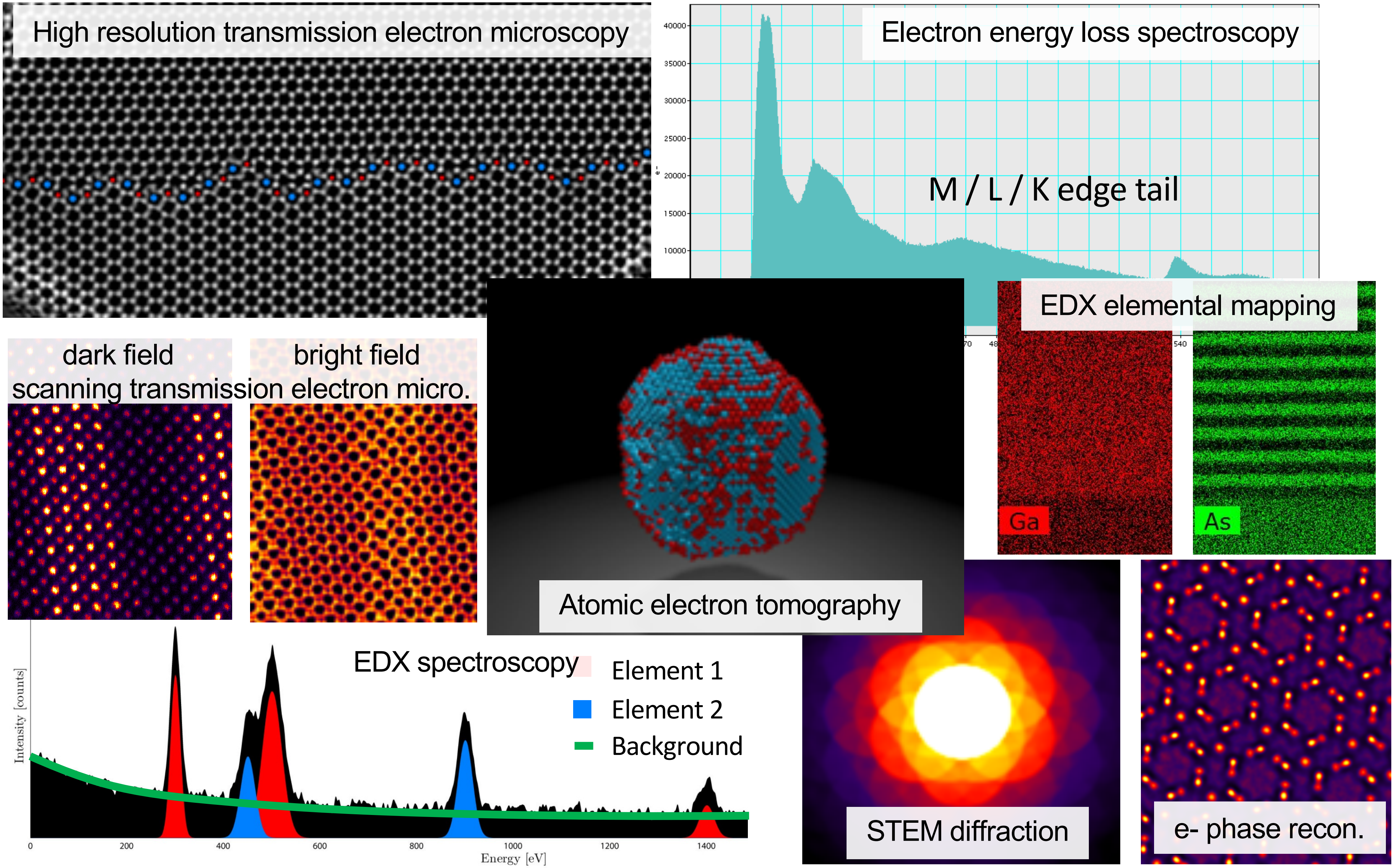




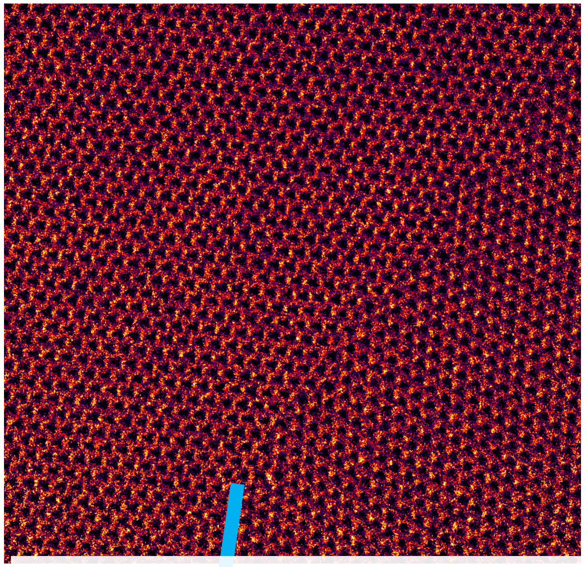
- 100,000 fps pixel detector
  - 576 x 576 x 10 μm
- Segmented silicon HAADF
- Fabricate detectors Q4CY2016

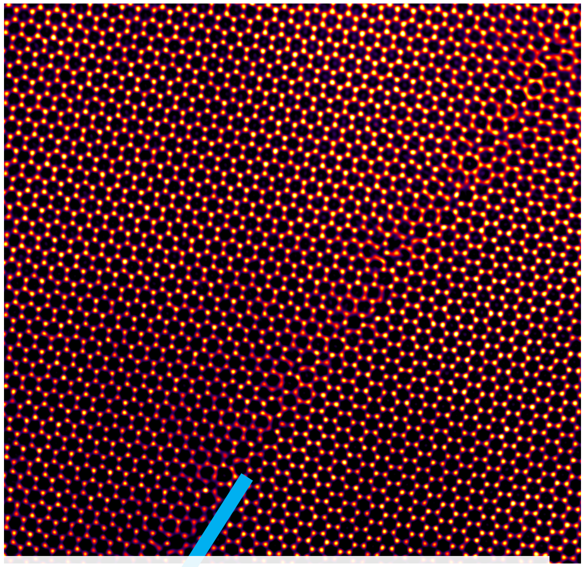
- Dedicated (donated)
  400 Gbs link to NERSC
  - Link testing underway
- Stream events to processors on Cori
- Future goal: firmware processing
  (reduce data rate)

Data Diversity makes National Center for Electron Microscopy (NCEM) perfect use case for pilot study

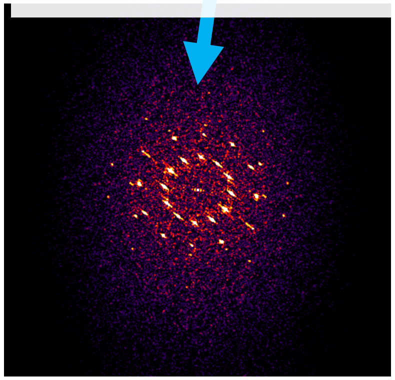


Online data quality metrics and high sparsity of Fourier Transform





Can an algorithm tell the image on the right is better?
Yes! Take Fourier transforms:



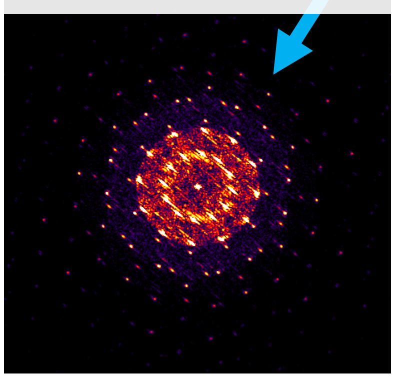


Image on the left has a lot more noise in background, less spots.

Below is an example of how the hardware aberration correction software works – can evaluate defocus, astigmatism, etc. but far too slowly to be used live.

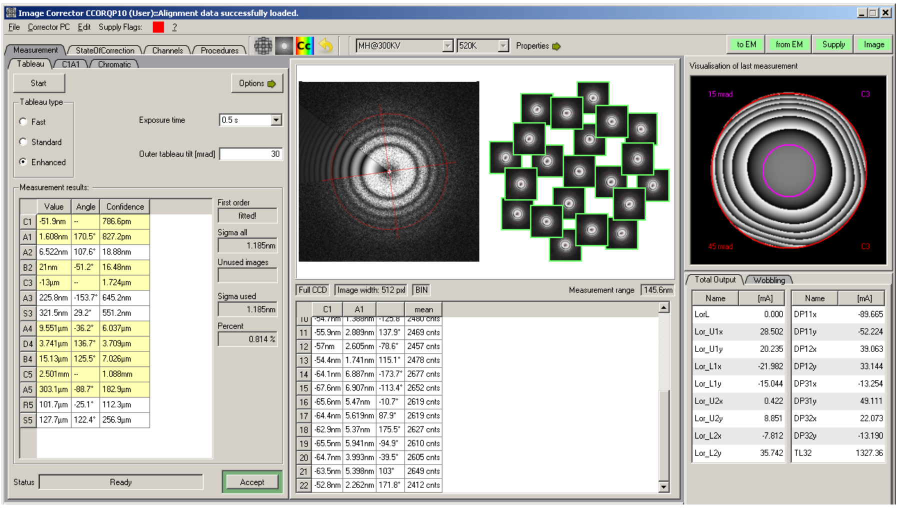


Image Corrector CCORQP10 (User)::Alignment data successfully loaded.

MH@300KV
520K

Tableau type

○ Fast
○ Standard
● Enhanced

Exposure time: 0.5 s

Outer tableau tilt [mrad]: 30

Measurement results:

	Value	Angle	Confidence
C1	-51.9nm	--	786.6pm
A1	1.608nm	170.5°	827.2pm
A2	6.522nm	107.6°	18.88nm
B2	21nm	-51.2°	16.48nm
C3	-13µm	--	1.724µm
A3	225.8nm	-153.7°	645.2nm
S3	321.5nm	29.2°	551.2nm
A4	9.551µm	-36.2°	6.037µm
D4	3.741µm	136.7°	3.709µm
B4	15.13µm	125.5°	7.026µm
C5	2.501mm	--	1.088mm
A5	303.1µm	-88.7°	182.9µm
R5	101.7µm	-25.1°	112.3µm
S5	127.7µm	122.4°	256.9µm

First order: fitted!

Sigma all: 1.185nm

Unused images:

Sigma used: 1.185nm

Percent: 0.814 %

Status: Ready

Full CCD | Image width: 512 pxl | BIN

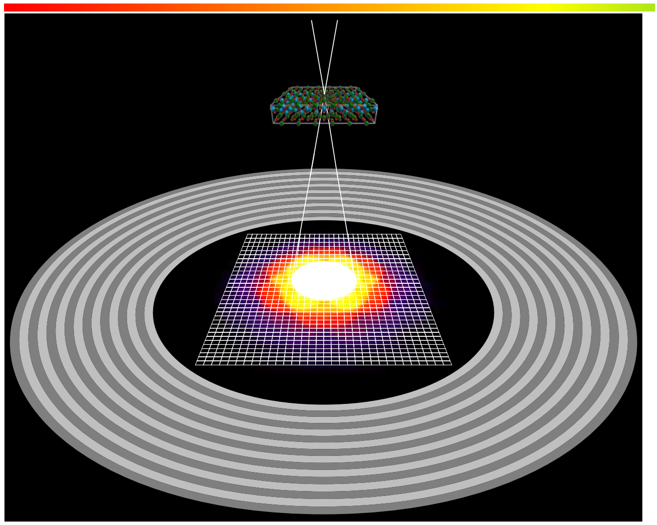
Measurement range: 145.6nm

	C1	A1		mean
10	-54.7nm	1.388nm	-125.8°	2480 cnts
11	-55.9nm	2.889nm	137.9°	2469 cnts
12	-57nm	2.605nm	-78.6°	2457 cnts
13	-54.4nm	1.741nm	115.1°	2478 cnts
14	-64.1nm	6.887nm	-173.7°	2677 cnts
15	-67.6nm	6.907nm	-113.4°	2652 cnts
16	-65.6nm	5.47nm	-10.7°	2619 cnts
17	-64.4nm	5.619nm	87.9°	2619 cnts
18	-62.9nm	5.37nm	175.5°	2627 cnts
19	-65.5nm	5.941nm	-94.9°	2610 cnts
20	-64.7nm	3.993nm	-39.5°	2605 cnts
21	-63.5nm	5.398nm	103°	2649 cnts
22	-52.8nm	2.262nm	171.8°	2412 cnts

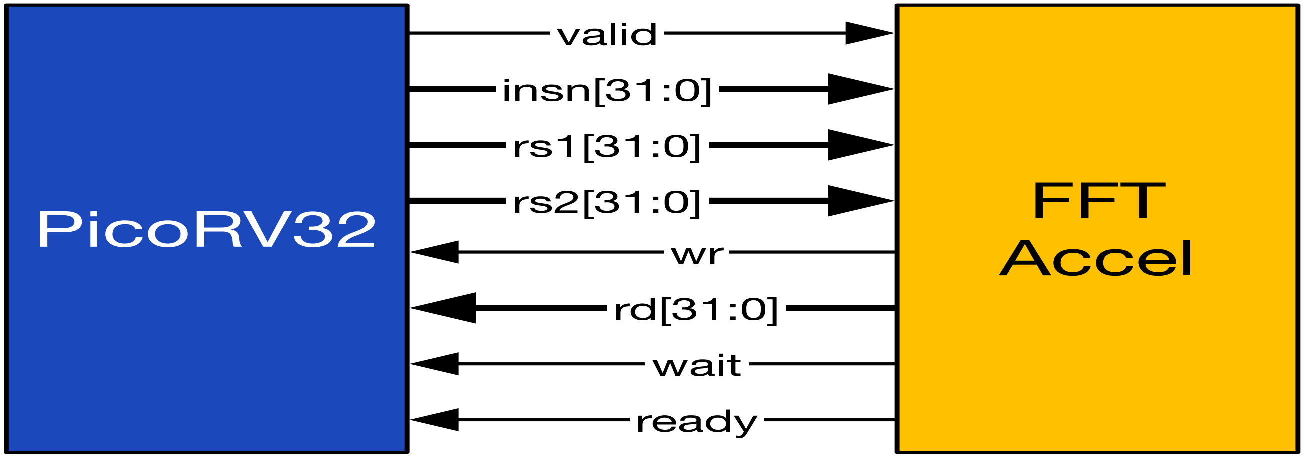
Name	[mA]	Name	[mA]
LorL	0.000	DP11x	-89.665
Lor_U1x	28.502	DP11y	-52.224
Lor_U1y	20.235	DP12x	39.063
Lor_L1x	-21.982	DP12y	33.144
Lor_L1y	-15.044	DP31x	-13.254
Lor_U2x	0.422	DP31y	49.111
Lor_U2y	8.851	DP32x	22.073
Lor_L2x	-7.812	DP32y	-13.190
Lor_L2y	35.742	TL32	1327.36

Results for RISC-V FFT Accelerator for CryoEM

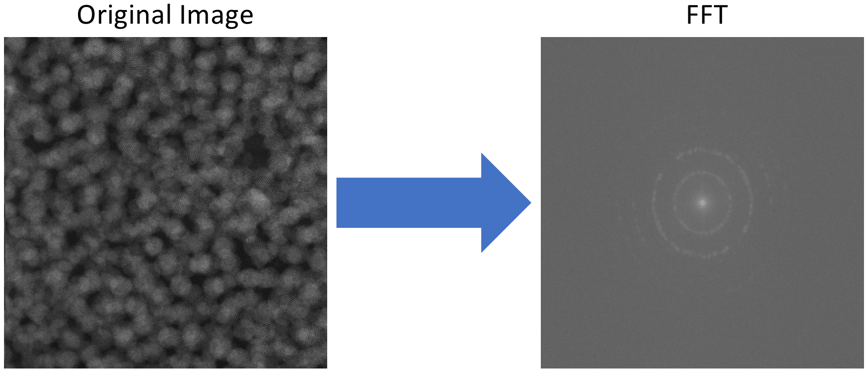
Benchmarking FFT Accelerator for image analysis



Instruction	opcode[3:2]	Description
fft_config	10b	Configures FFT parameters
fft_status	01b	Reads FFTAccel status registers
fft_start	11b	Starts FFT processing
fft_stop	00b	Stops FFT processing



- 100,000 fps pixel detector
  - 576 x 576 x 10 μm
- Segmented silicon HAADF
- Fabricate detectors Q4CY2016



Created RISC-V Core with FFT ISA Extension

RISC-V+FFT Accel 126x faster than x86 host

- FFT on Intel Core i7-5930K @ 3.50GHz: **~265ms**
- FFTAccel (Floating): **~2.10ms**

BERKELEY LAB

U.S. DEPARTMENT OF ENERGY | Office of Science

Quantum Computing Infrastructure: Challenges / Opportunities

- Control System / Test Electronics Complexity

- Control response time limits QC performance

- Complexity of Circuits/ Programs to implement quantum algorithms

Test Electronics

Immersion Probe and LHe Dewar

Quantum Instruction Set Architecture (QUASAR)

PIs: Carter, Siddiqi, DuBois, Advanced Quantum-Enabled Simulation Testbed, July, 2017
Anastasiia Butko and Dubois: QUASAR Developers

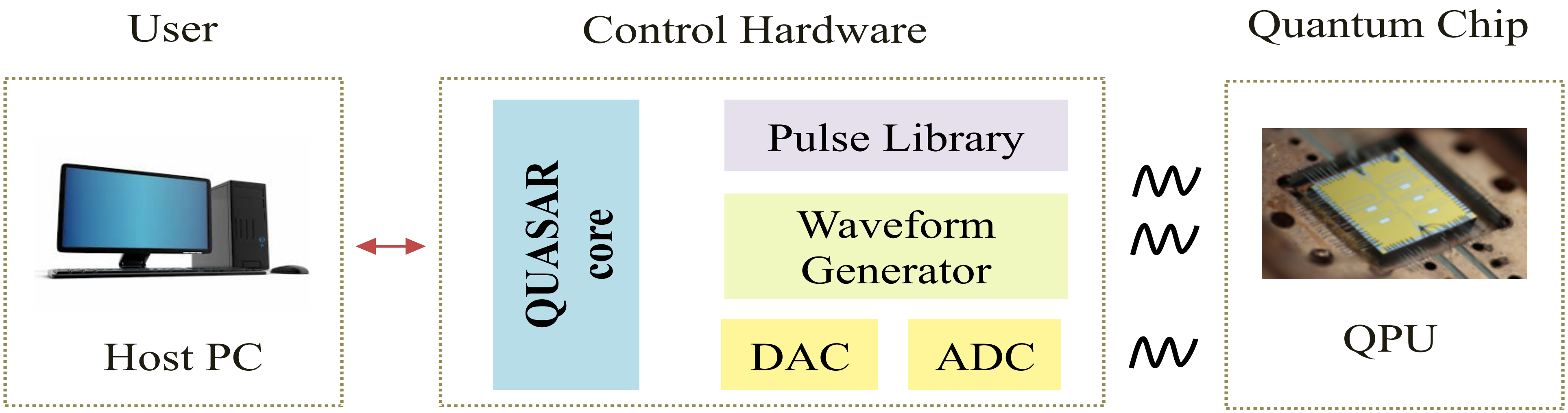
New Concept

We have developed the Quantum instruction set architecture (QUASAR) to provide a compact and efficient method to program complex sequences of operations for the interface to a quantum computer. We have also developed a QUASAR microcontroller core that enables fast in-situ programming and control of qubits for a quantum computer.

Significance and Impact

The QUASAR ISA significantly improves quantum computer programmability thereby facilitating their integration into future extremely heterogeneous systems. The QUASAR core (operating in-situ with the qubits using superconducting logic) reduces signal propagation and decision-making delays enabling more efficient usage of limited qubit lifetimes..

Research Details

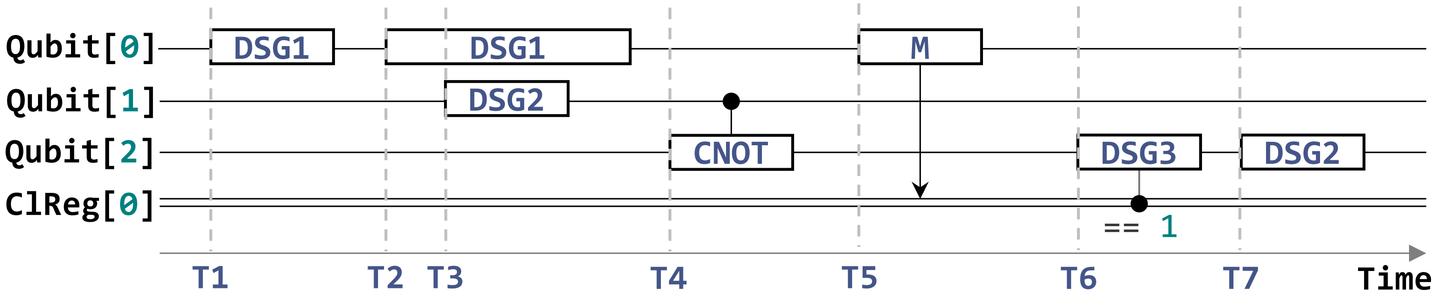


Software-to-hardware quantum system : (from left to right) user host computer loads quantum program into QUASAR core memory. QUASAR core executes program generating commands to low-level control hardware, i.e. waveform generator, analog-digital converter (ADC), etc. Low-level analog devices communicate with quantum chip via waveforms. Quasar Microcontroller is implemented using the Chisel Hardware Description Language to enable future optimization and technology integration.

Extending RISC-V ISA :
Because some companies don't like you messing with their ISA

Quantum ISA program

- Program example from LvlX



```
Time          Gate          QubitA        QubitB        Phase         ClReg         Duration
T1            DSG1          0             -             -             -             -
T2            DSG1          0             -             -             -             20ns
T3            DSG2          1             -             -             -             -
T4            CNOT          1             2             -             -             -
T5            M             0             -             -             0             -
if ClReg[0] == 1:
T6            DSG3          2             -             0.392699081698724           -
```

Encoding capabilities

- 32-bit instruction length
- up-to 48 gates (single- and dual-qubit)
- 128-qubit addressing
- 32-bit arbitrary phase

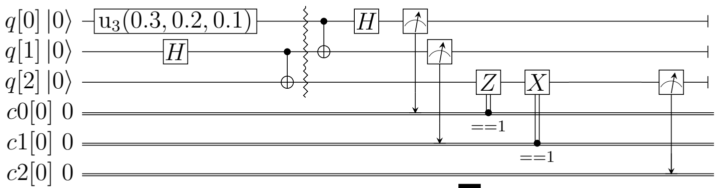
- precise timing control (5ns cycle)
- parallel gate application
- conditional branching
- extensions: pulse shaping

format	0-6		7	8-14	15-16	17-23	24-25	26-30	31
D	opcode		maskBit	targetA	imm	targetB	imm	imm	DurBit
	type	fun	single/mult	qubit/maskReg	offsetA	qubit/maskReg	offsetB	Phase R	custom
	3	4	1	7	2	7	2	5	1
CNOT	011	0000	false/true	0-127	0-3	0-127	0-3	0-31	false/true

Quantum Circuit: compilation flow

- Multiple levels of details/compilation

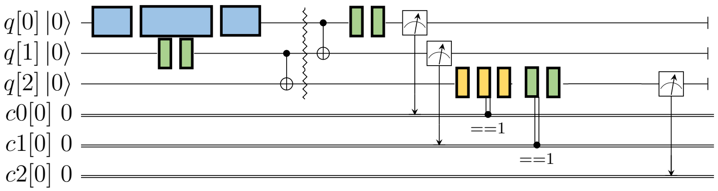
Lvl1

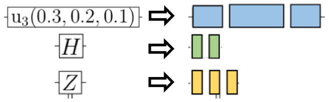


High level of abstraction (e.g. QASM)

Decompose into device supported gates

Lvl2





- Decomposition problem, may have multiple solution
- Device-specific and requires a library of supported gat
- This level defines the *"Pulse Library"*

Place gates in respect of device connectivity

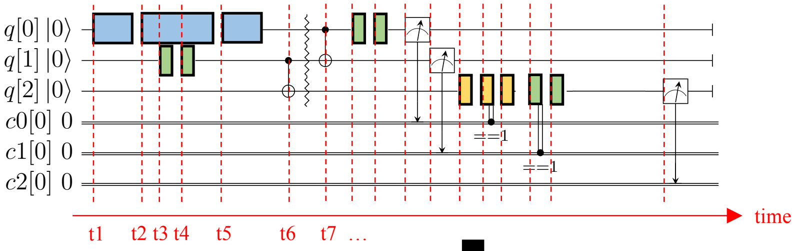
Lvl3

- Optimization problem (different gate-fidelity/etc. options)
- Device-specific and requires the knowledge of qubit characteristics and topology

Timing adjustments

…

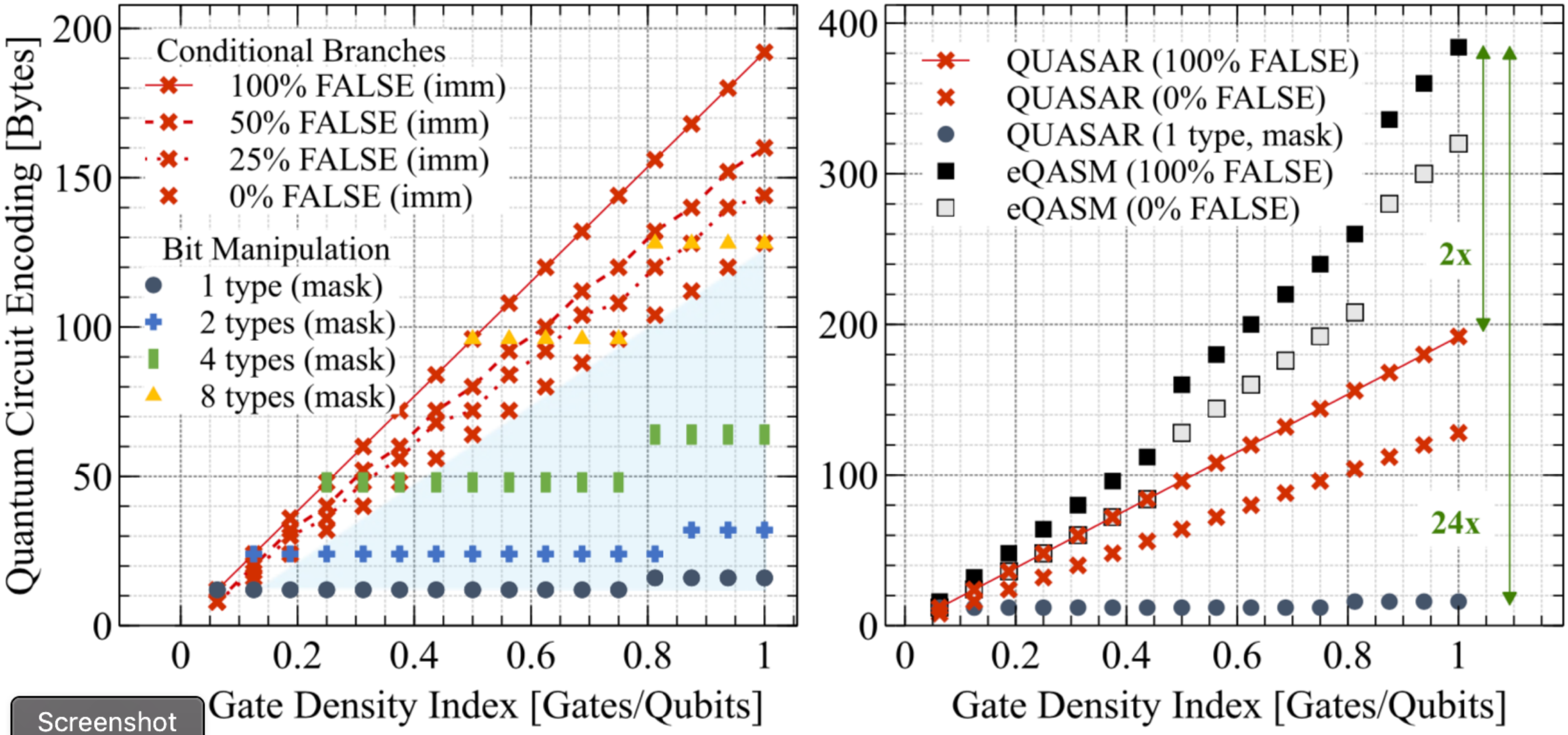
LvlX



- May have different timings
- Requires the knowledge of gate duration, device-specific constraints

Program can be expressed in quantum ISA

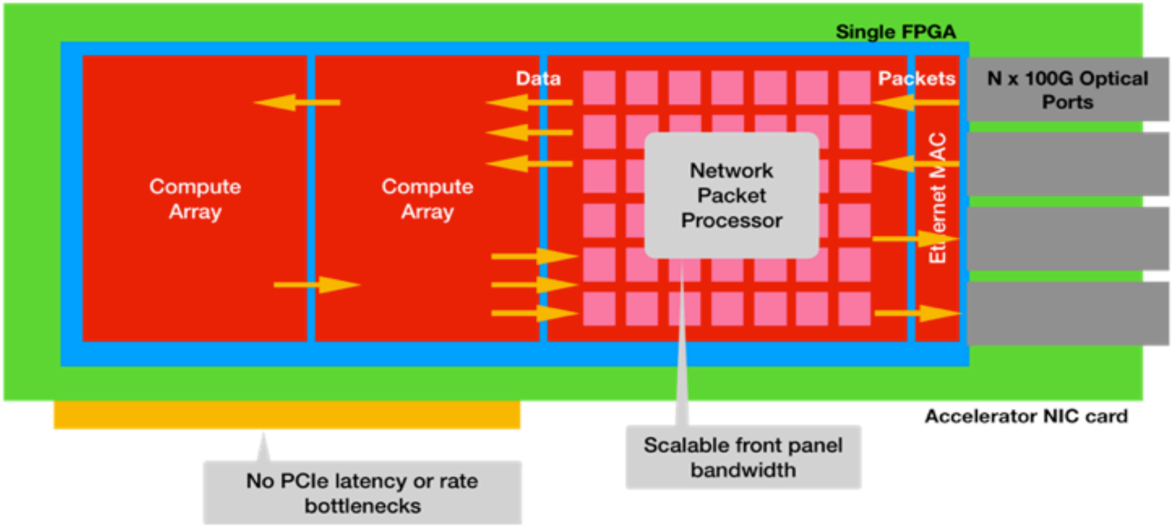
QUASAR Performance Results

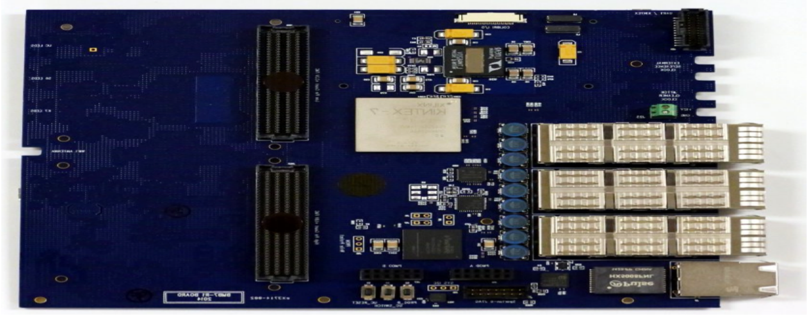


(a) QUASAR: conditional branches vs. bit manipulation (b) QUASAR (this work) vs. eQASM [10]

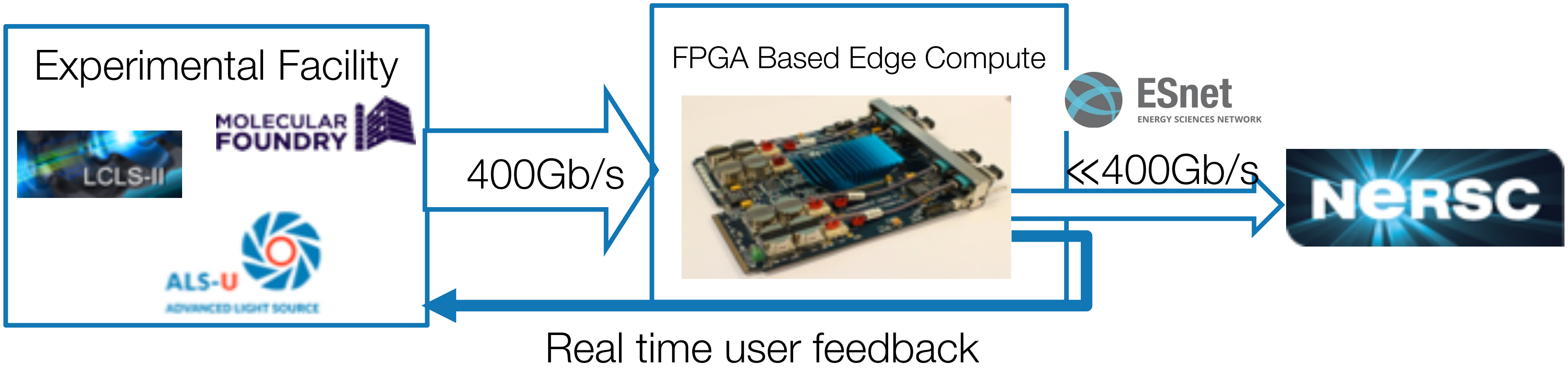
In-Transit (streaming) Data Processing with Emerging FPGA Devices

Emerging Acceleration and Networking Hardware



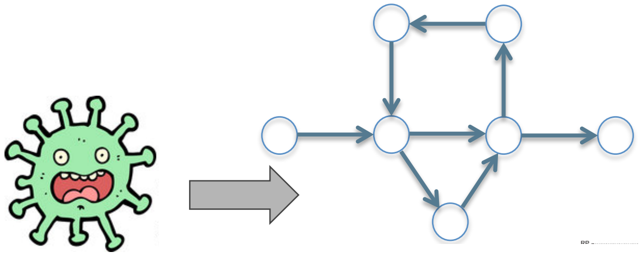


- **FPGAs well suited to streaming processing**
  - IO Bandwidth > 400Gb/s; 1.5TFlops per device – HBM at 460 GB/s
- **Algorithms for data reduction and analysis scale on FPGAs**



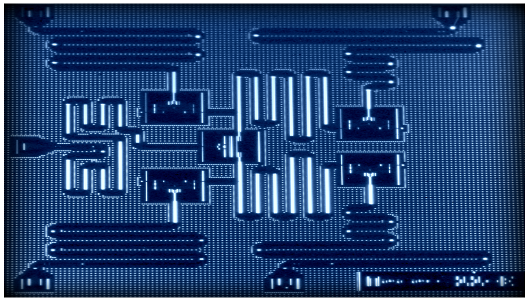
More opportunities for specialization

Custom computing could be applied and incorporated in many areas



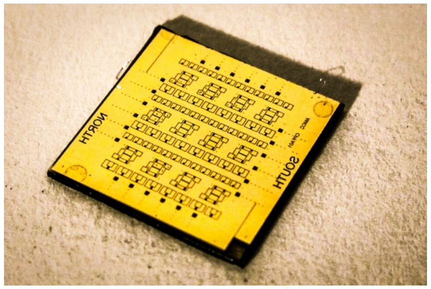
JGI – Accelerating sequence alignment

- Traditional processors not optimal for bioinformatics codes
- FPGAs provide potential solution



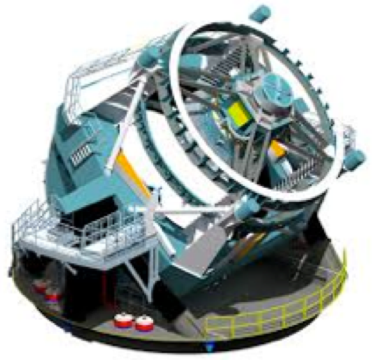
Quantum – Control Processor

- Future Quantum accelerators limited by not just number of qubits
- Leverage our work with superconducting logic design tools



LSST – Transient detection

- Power constrained on-site, custom, low-power computing could reduce need for lossy compression or high-bandwidth, long-haul networks



Questions?