Striving for SDR Performance Portability in the Era of Heterogeneous SoCs

Jeffrey S. Vetter
Seyong Lee
Mehmet Belviranli
Roberto Gioiosa
Richard Glassbrook
Abdel-Kareem Moadi

GNU Radio Conference 2019
Huntsville, Alabama
16 Sep 2019

http://ft.ornl.gov vetter@computer.org
Highlights

• Motivation: Recent trends in computing paint an ambiguous future
  – Contemporary systems provide evidence that power constraints are driving architectures to change rapidly
  – Multiple architectural dimensions are being (dramatically) redesigned: Processors, node design, memory systems, I/O
  – Complexity is our main challenge

• Applications and software systems across many areas are all reaching a state of crisis
  – Applications will not be functionally or performance portable across architectures
  – Programming and operating systems need major redesign to address these architectural changes
  – Procurements, acceptance testing, and operations of today’s new platforms depend on performance prediction and benchmarking.

• ORNL Cosmic project investigating design and programming challenges for these trends in SDR
  – Performance modeling and ontologies
  – Performance portable compilation to many different heterogeneous architectures/SoCs
  – Intelligent scheduling system to automate discovery, device selection, and data movement
  – Targeting wide variety of existing and future architectures (DSSoC and others)
Motivating Trends
Contemporary devices are approaching fundamental limits

Dennard scaling has already ended. Dennard observed that voltage and current should be proportional to the linear dimensions of a transistor: 2x transistor count implies 40% faster and 50% more efficient.


### News & Analysis

**Foundries' Sales Show Hard Times Continuing**

*Peter Clarke, 5/23/2016 09:33 PM EDT, 2 comments*

- **Semiconductor Engineering**

  Uncertainty Grows For 5nm, 3nm
  The outlook for the semiconductor manufacturing industry remains uncertain as companies struggle with the challenges of transitioning to 5-nanometer and 3-nanometer processes.

---

**Number of Foundries with a Cutting Edge Logic Fab**

<table>
<thead>
<tr>
<th>Site</th>
<th>Foundry</th>
</tr>
</thead>
<tbody>
<tr>
<td>STIeM</td>
<td>X-FAB</td>
</tr>
<tr>
<td>Dongbu HiTek</td>
<td></td>
</tr>
<tr>
<td>ADI</td>
<td>ADI</td>
</tr>
<tr>
<td>Atmel</td>
<td>Atmel</td>
</tr>
<tr>
<td>Rohm</td>
<td>Rohm</td>
</tr>
<tr>
<td>Sanyo</td>
<td>Sanyo</td>
</tr>
<tr>
<td>Mitsubishi</td>
<td>Mitsubishi</td>
</tr>
<tr>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>Hitachi</td>
<td>Hitachi</td>
</tr>
<tr>
<td>Cypress</td>
<td>Cypress</td>
</tr>
<tr>
<td>Sony</td>
<td>Sony</td>
</tr>
<tr>
<td>Infineon</td>
<td>Infineon</td>
</tr>
<tr>
<td>Sharp</td>
<td>Sharp</td>
</tr>
<tr>
<td>Freescale</td>
<td>Freescale</td>
</tr>
<tr>
<td>Renesas</td>
<td>Renesas</td>
</tr>
<tr>
<td>SMIC</td>
<td>SMIC</td>
</tr>
<tr>
<td>Toshiba</td>
<td>Toshiba</td>
</tr>
<tr>
<td>Fujitsu</td>
<td>Fujitsu</td>
</tr>
<tr>
<td>TI</td>
<td>TI</td>
</tr>
<tr>
<td>Panasonic</td>
<td>Panasonic</td>
</tr>
<tr>
<td>STM/SMIC</td>
<td>STM</td>
</tr>
<tr>
<td>UMC</td>
<td>UMC</td>
</tr>
<tr>
<td>IBM</td>
<td>IBM</td>
</tr>
<tr>
<td>AMD</td>
<td>AMD</td>
</tr>
<tr>
<td>Samsung</td>
<td>Samsung</td>
</tr>
<tr>
<td>TSMC</td>
<td>TSMC</td>
</tr>
<tr>
<td>GF</td>
<td>GF</td>
</tr>
<tr>
<td>GlobalFoundries</td>
<td>GlobalFoundries</td>
</tr>
</tbody>
</table>

---

**GlobalFoundries Forfeit 7nm Manufacturing - EE Times Asia**

GlobalFoundries has decided to drop its efforts to develop 7-nanometer manufacturing capabilities, focusing instead on other processes.

---

**Samsung to Invest $115 Billion in Foundry & Chip Businesses by 2030**

Samsung has announced plans to invest $115 billion over the next decade to expand its foundry and chip businesses, aiming to remain a leading player in the semiconductor industry.

---

**Intel's 10nm Is Broken, Delayed Until 2019**

*By Paul Alcorn, April 26, 2018 at 6:30 PM*

Intel's 10-nanometer process, intended for use in its first-generation Ice Lake processors, has been delayed and may not be ready until 2019.

---

**GlobalFoundries Selling ASIC Business to Marvell**

*By Dylan McGrath, 05/01/19, 1 comment*

GlobalFoundries has agreed to sell its ASIC business to Marvell Technology Group, marking the end of the company's efforts in the ASIC market.

---

**Another Step Toward the End of Moore's Law**

Samsung and TSMC have moved to 5-nanometer manufacturing, marking a significant milestone in the semiconductor industry and pushing the limits of Moore's Law.
Business climate reflects this uncertainty, cost, complexity, consolidation
Sixth Wave of Computing

http://www.kurzweilai.net/exponential-growth-of-computing
Predictions for Transition Period

Optimize Software and Expose New Hierarchical Parallelism
- Redesign software to boost performance on upcoming architectures
- Exploit new levels of parallelism and efficient data movement

Architectural Specialization and Integration
- Use CMOS more effectively for specific workloads
- Integrate components to boost performance and eliminate inefficiencies
- Workload specific memory+storage system design

Emerging Technologies
- Investigate new computational paradigms
  - Quantum
  - Neuromorphic
  - Advanced Digital
  - Emerging Memory Devices
Predictions for Transition Period

**Optimize Software and Expose New Hierarchical Parallelism**
- Redesign software to boost performance on upcoming architectures
- Exploit new levels of parallelism and efficient data movement

**Architectural Specialization and Integration**
- Use CMOS more effectively for specific workloads
- Integrate components to boost performance and eliminate inefficiencies
- Workload specific memory+storage system design

**Emerging Technologies**
- Investigate new computational paradigms
  - Quantum
  - Neuromorphic
  - Advanced Digital
  - Emerging Memory Devices
Predictions for Transition Period

Optimize Software and Expose New Hierarchical Parallelism

• Redesign software to boost performance on upcoming architectures
• Exploit new levels of parallelism and efficient data movement

Architectural Specialization and Integration

• Use CMOS more effectively for specific workloads
• Integrate components to boost performance and eliminate inefficiencies
• Workload specific memory+storage system design

Emerging Technologies

• Investigate new computational paradigms
  • Quantum
  • Neuromorphic
  • Advanced Digital
  • Emerging Memory Devices
Predictions for Transition Period

Optimize Software and Expose New Hierarchical Parallelism

• Redesign software to boost performance on upcoming architectures
• Exploit new levels of parallelism and efficient data movement

Architectural Specialization and Integration

• Use CMOS more effectively for specific workloads
• Integrate components to boost performance and eliminate inefficiencies
• Workload specific memory+storage system design

Emerging Technologies

• Investigate new computational paradigms
  • Quantum
  • Neuromorphic
  • Advanced Digital
  • Emerging Memory Devices
Pace of Architectural Specialization is Quickening

- Industry, lacking Moore’s Law, will need to continue to differentiate products (to stay in business)
  - Use the same transistors differently to enhance performance
- Architectural design will become extremely important, critical
  - Dark Silicon
  - Address new parameters for benefits/curse of Moore’s Law
- 50+ new companies focusing on hardware for Machine Learning

Analysis of Apple A-* SoCs

http://vlsiarch.eecs.harvard.edu/accelerators/die-photo-analysis
Growing Open Source Hardware Movement Enables Rapid Chip Design

RISC-V Ecosystem

Open-source software:
Gcc, binutils, glibc, Linux, BSD, LLVM, QEMU, FreeRTOS, ZephyrOS, LiteOS, SylixOS, ...

Commercial software:
Lauterbach, Segger, Micrium, ExpressLogic, ...

Software

RISC-V Foundation

ISA specification  Golden Model  Compliance

Hardware

Open-source cores:
Rocket, BOOM, RISCY, Ariane, PicoRV32, Piccolo, SCR1, Hummingbird, ...

Commercial core providers:
Andes, Bluespec, Cloudbear, Codasip, Cortus, C-Sky, Nuclei, SiFive, Syntacore, ...

Inhouse cores:
Nvidia, +others
Summary: Transition Period will be Disruptive – Opportunities and Pitfalls Abound

- New devices and architectures may not be hidden in traditional levels of abstraction

- Examples
  - A new type of CNT transistor may be completely hidden from higher levels
  - A new paradigm like quantum may require new architectures, programming models, and algorithmic approaches

<table>
<thead>
<tr>
<th>Layer</th>
<th>Switch, 3D</th>
<th>NVM</th>
<th>Approximate</th>
<th>Neuro</th>
<th>Quantum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Algorithm</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Language</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>API</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Arch</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>ISA</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Microarch</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>FU</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Logic</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Device</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Adapted from IEEE Rebooting Computing Chart
The Summit System @ ORNL
#1 on Top 500 since June 2018

System Performance
- Peak of 200 Petaflops \( (\text{FP}_{64}) \) for modeling & simulation
- Peak of 3.3 ExaOps \( (\text{FP}_{16}) \) for data analytics and artificial intelligence
- Max power 13 MW

The system includes
- 4,608 nodes
- Dual-rail Mellanox EDR InfiniBand network
- 250 PB IBM file system transferring data at 2.5 TB/s

Each node has
- 2 IBM POWER9 processors
- 6 NVIDIA Tesla V100 GPUs
- 608 GB of fast memory (96 GB HBM2 + 512 GB DDR4)
- 1.6 TB of NV memory
U.S. Department of Energy and Cray to Deliver Record-Setting Frontier Supercomputer at ORNL

Exascale system expected to be world’s most powerful computer for science and innovation

Topic: Supercomputing

May 7, 2019

OAK RIDGE, Tenn., May 7, 2019—The U.S. Department of Energy today announced a contract with Cray Inc. to build the Frontier supercomputer at Oak Ridge National Laboratory, which is anticipated to debut in 2021 as the world’s most powerful computer with performance of greater than 1.5 exaflops.

Scheduled for delivery in 2021, Frontier will accelerate innovation in science and technology and maintain U.S. leadership in high-performance computing and artificial intelligence. The total contract award is valued at more than $600 million for the system and technology development. The system will be based on Cray’s new Shasta architecture and Slingshot Interconnect and will feature high-performance AMD EPYC CPU and AMD Radeon Instinct GPU technology.

<table>
<thead>
<tr>
<th>Topic: Supercomputing</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 7, 2019</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th><strong>Peak Performance</strong></th>
<th>&gt;1.5 EF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Footprint</strong></td>
<td>&gt; 100 cabinets</td>
</tr>
</tbody>
</table>
| **Node**              | 1 HPC and AI Optimized AMD EPYC CPU  
4 Purpose-Built AMD Radeon Instinct GPU |
| **CPU-GPU Interconnect** | AMD Infinity Fabric  
Coherent memory across the node |
| **System Interconnect** | Multiple Slingshot NICs providing 100 GB/s network bandwidth  
Slingshot dragonfly network which provides adaptive routing, congestion management and quality of service. |
| **Storage**           | 2-4x performance and capacity of Summit's I/O subsystem. Frontier will have near node storage like Summit. |
Department of Energy (DOE) Roadmap to Exascale Systems

An impressive, productive lineup of *accelerated node* systems supporting DOE’s mission

**Pre-Exascale Systems** [Aggregate Linpack (Rmax) = 323 PF!]

<table>
<thead>
<tr>
<th>Year</th>
<th>2012</th>
<th>2016</th>
<th>2018</th>
<th>2020</th>
<th>2021-2023</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Titan (9)</strong></td>
<td>ORNL Cray/AMD/NVIDIA</td>
<td>ANL IBM BG/Q</td>
<td><strong>Mira (21)</strong></td>
<td>IBM/NVIDIA</td>
<td><strong>Sequoia (10)</strong></td>
</tr>
<tr>
<td><strong>Trinity (6)</strong></td>
<td>LANL/SNL Cray/Intel Xeon/KNL</td>
<td>LANL/SNL TBD</td>
<td><strong>Sierra (2)</strong></td>
<td>LLNL IBM/NVIDIA</td>
<td><strong>Cori (12)</strong></td>
</tr>
<tr>
<td><strong>Theta (24)</strong></td>
<td>ANL Cray/Intel Xeon/KNL</td>
<td><strong>Summit (1)</strong></td>
<td><strong>Perlmutter</strong></td>
<td>LBNL Cray/AMD/NVIDIA</td>
<td><strong>CrossRoads</strong></td>
</tr>
<tr>
<td><strong>Pre-Exascale Systems</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Heterogeneous Cores**
- **Deep Memory incl NVM**
- **Plateauing I/O Performance**

**First U.S. Exascale Systems**

- **ORNL**
  - Summit (1) IBM/NVIDIA
  - ORNL IBM/NVIDIA
- **ANL**
  - Mira (21) IBM BG/Q
  - Cori (12) LBNL Cray/Intel KNL
- **LANL/SNL**
  - Trinity (6) LANL/SNL Cray/Intel Xeon/KNL
  - Theta (24) ANL Cray/Intel Xeon/KNL
- **LLNL**
  - Sequoia (10) LLNL IBM BG/Q
  - Sierra (2) LLNL IBM/NVIDIA
  - LLNL TBD
- **ORNL**
  - Summit (1) IBM/NVIDIA
  - ORNL IBM/NVIDIA
- **ANL**
  - Mira (21) IBM BG/Q
  - Cori (12) LBNL Cray/Intel KNL
- **LANL/SNL**
  - Trinity (6) LANL/SNL Cray/Intel Xeon/KNL
  - Theta (24) ANL Cray/Intel Xeon/KNL
- **LLNL**
  - Sequoia (10) LLNL IBM BG/Q
  - Sierra (2) LLNL IBM/NVIDIA
  - LLNL TBD
- **ORNL**
  - Summit (1) IBM/NVIDIA
  - ORNL IBM/NVIDIA
- **ANL**
  - Mira (21) IBM BG/Q
  - Cori (12) LBNL Cray/Intel KNL
- **LANL/SNL**
  - Trinity (6) LANL/SNL Cray/Intel Xeon/KNL
  - Theta (24) ANL Cray/Intel Xeon/KNL
- **LLNL**
  - Sequoia (10) LLNL IBM BG/Q
  - Sierra (2) LLNL IBM/NVIDIA
  - LLNL TBD
- **ORNL**
  - Summit (1) IBM/NVIDIA
  - ORNL IBM/NVIDIA
- **ANL**
  - Mira (21) IBM BG/Q
  - Cori (12) LBNL Cray/Intel KNL
- **LANL/SNL**
  - Trinity (6) LANL/SNL Cray/Intel Xeon/KNL
  - Theta (24) ANL Cray/Intel Xeon/KNL
- **LLNL**
  - Sequoia (10) LLNL IBM BG/Q
  - Sierra (2) LLNL IBM/NVIDIA
  - LLNL TBD

**First U.S. Exascale Systems**

- **ORNL**
  - Summit (1) IBM/NVIDIA
  - ORNL IBM/NVIDIA
- **ANL**
  - Mira (21) IBM BG/Q
  - Cori (12) LBNL Cray/Intel KNL
- **LANL/SNL**
  - Trinity (6) LANL/SNL Cray/Intel Xeon/KNL
  - Theta (24) ANL Cray/Intel Xeon/KNL
- **LLNL**
  - Sequoia (10) LLNL IBM BG/Q
  - Sierra (2) LLNL IBM/NVIDIA
  - LLNL TBD

**Jan 2018**
Domain Specific System on Chip (DSSoC) Program to address these challenges
Performer domains and applications

IBM T. J. Watson Research Center
Pradip Bose
Columbia University, Harvard University, Univ. of Illinois at Urbana-Champaign

CV+SDR
- Multi-domain application
- Multi-spectral processing
- Communications

Arizona State University
Daniel W. Bliss
Univ. of Michigan, Carnegie Mellon University, General Dynamic Mission Systems, Arm Ltd., Episys Science

SDR
- Unmanned aerial
- Small robotic & leave-behind
- Universal soldier systems
- Multifunction systems

Raytheon
Tom Kazior

SDR
- Xilinx ACAP
- Visual system integrator
- Improved reconfigurability of processing

Raytheon

Computer Vision
- Still image and video processing
- Autonomous navigation
- Continuous surveillance
- Augmented reality

Stanford University
Mark Horowitz
Clark Barrett, Kayvon Fatahalian, Pat Hanrahan, Priyanka Raina

Oak Ridge National Laboratory
Jeffrey Vetter

SDR
- Communications and signal processing focused
- Up-front processing / data cutdown
- Improving understanding of processing systems

Stanford
Google/YouTube

ORNL

PlastyForma

Slide courtesy of Dr. Tom Rondeau, DARPA MTO
ORNL Cosmic Project
Cosmic Castle | Development Lifecycle

• Create scalable application Aspen models manually, with static or dynamic analysis, or using historical information

• Ontologies based on Aspen models using statistical and machine learning techniques

• Programming systems built to support ontologies
  • Query Aspen models and PFU for automatic code generation, optimization, etc.

• Intelligent runtime scheduling uses models and PFU to inform dynamic decisions
  • Dynamic resource discovery and monitoring

• DSSoC design quantitatively derived from application Aspen models
  • Early design space exploration with Aspen

• DSSoC design quantitatively derived from application Aspen models
  • Early design space exploration with Aspen

Dynamic Performance Feedback including profiling and configuration response

Precise configuration and benchmark data for static analysis, mapping, partitioning, code generation, etc.
Cosmic Castle | Project Overview

Applications
- Streaming (e.g., SW Radio)
- Sensing (e.g., SAR, vision)
- Deep learning (e.g., CNN)
- Analytics (e.g., graphs)
- Robotics (e.g., sense and react)
- Science and Engineering (e.g., CFD)

Ontologies include Motifs, Parallel Execution Patterns, Locality, Affinity, Sync, etc.

Programming Systems
- Compiler
- DSL
- JIT
- Libraries
- etc

Performance Models combine Ontologies and Machine Models to produce both static and dynamic cost estimates

Runtime and Operating Systems
- Task scheduling
- Memory Mgt
- IO
- Synchronization

Machine Models include architecture configuration information, microbenchmarks, and API for PFU

DSSoC Hardware
- Performance Parameters
- CPU
- GPU
- FPGA
- Accelerator
- DSP
- Deep Memory
- Neuromorphic
Cosmic Castle | Project Overview

**Applications**
- Streaming (e.g., SW Radio)
- Sensing (e.g., SAR, vision)
- Deep learning (e.g., CNN)
- Analytics (e.g., graphs)
- Robotics (e.g., sense and react)
- Science and Engineering (e.g., CFD)

**Ontologies**
- Include Motifs, Parallel Execution Patterns, Locality, Affinity, Sync, etc.

**Programming Systems**
- Compiler
- DSL
- JIT
- Libraries
- etc

**Parameterizable Performance Models**
- Concurrency
- Work
- Communication
- Capacities
- Synchronization

**Machine Model**
- Compute
- Memory
- Interconnects

**DSSoC Hardware**
- CPU
- GPU
- FPGA
- Accelerator
- DSP
- Deep Memory
- Neuromorphic

**Runtime and Operating Systems**
- Task scheduling
- Memory Mgt
- IO
- Synchronization

**PFU API**
Summit (IBM POWER9+NVIDIA Volta) Node installed
Experimental Computing Lab (ExCL) managed by the ORNL Future Technologies Group

- IBM Summit Node with 6 Nvidia Tesla V100 GPUs (8935-GTX)
  - Same CPU/GPU/Memory as nodes in OLCF Summit
    - 2 Power9 CPUs (IBM POWER9)
    - 22 Cores each, 4 threads/core
    - 896GB main memory
    - 6 Tesla V100 SM2 350GB GPU
  - Provides a development and evaluation environment for Power9/V100 GPUs
  - Tracks (as closely as possible) the software stack in use on Summit
  - Shared / Queued / Single User availability modes will be available

AMD Radeon VII Available
Experimental Computing Lab (ExCL) managed by the ORNL Future Technologies Group

- AMD Radeon VII, Vega 20 Architecture
  - 8Cn 5 nm TSMC 7TF process, 13.2B transistors
  - 50 Compute Units with 3.4 GF peak TF
  - 16 GB HBM2 with 4096-bit width (or ~1TBps bandwidth)
  - TDP 300W
  - PCIe 3.0 x16

- Intel Xeon Skylake Host
  - HP Z4 G4 Workstation w/ PCIe 3.0 x16
  - 2 X1235 / 2 X2620 host
  - 1 CPU / 4 cores / 2 threads/core
  - 512 GB SSD uncommitted/available

- Software
  - AMD ROCm development tools
  - HIP (homogeneous Compute Interface for Portability) available
    - OpenCL 2.1

- Additional Details
  - https://www.ossrsrc.org/about/13032/AMD-Radeon-VII
  - https://github.com/AMD/amd-ryzen-vii

NVIDIA DGX Workstation Available
Experimental Computing Lab (ExCL) managed by the ORNL Future Technologies Group

- 4X Tesla V100 GPUs
- TFLOPS (Mixed precision) 500
- GPU Memory 128 GB total system
- NVIDIA Tensor Cores 2,560
- NVIDIA CUDA® Cores 20,480
- CPU Intel Xeon E5-2698 v4 2.2 GHz (20-Core)
- System Memory 256 GB RDIMM DDR4
- Full NVIDIA stack
- Other compilers/tools installable on request

ARM ThunderX2 Node Available
Experimental Computing Lab (ExCL) managed by the ORNL Future Technologies Group

ThunderX2 Workstation
- Cavium (Marvell) ThunderX2 with ARMv8.1 instruction set.
  - 2 Cpus, each with 28 Cores with 4 threads/core
- 128 GiB Main Memory
- Gigabyte MT91-F81-00 motherboard
- Multiple access levels available to researchers investigating ARMv8 performance
- Traditional ARM/Linux software stack available
Intel Stratix 10 FPGA available
Experimental Computing Lab (ExCL) managed by the ORNL Future Technologies Group

- Intel Stratix 10 FPGA and four banks of DDR4 external memory
  - Board configuration: Nallatech 520 Network Acceleration Card

- Up to 10 TFLOPS of peak single precision performance

- 25MBytes of L1 cache @ up to 94 TBytes/s peak bandwidth

- 2X Core performance gains over Arria® 10

- Quartus and OpenCL software (Intel SDK v18.1) for using FPGA

- Provide researcher access to advanced FPGA/SOC environment

For more information or to apply for an account, visit https://excl.ornl.gov/
NVIDIA Jetson AGX Xavier SoC available
Experimental Computing Lab (ExCL) managed by the ORNL Future Technologies Group

NVIDIA Jetson AGX Xavier:

- High-performance system on a chip for autonomous machines
- Heterogeneous SoC contains:
  - Eight-core 64-bit ARMv8.2 CPU cluster (Carmel)
  - 1.4 CUDA TFLOPS (FP32) GPU with additional inference optimizations (Volta)
  - 11.4 DL TOPS (INT8) Deep learning accelerator (NVDLA)
  - 1.7 CV TOPS (INT8) 7-slot VLIW dual-processor Vision accelerator (PVA)
  - A set of multimedia accelerators (stereo, LDC, optical flow)
- Provides researchers access to advanced high-performance SOC environment

For more information or to apply for an account, visit https://excl.ornl.gov/
**Qualcomm 855 SoC (SM8510P)**

Experimental Computing Lab (ExCL) managed by the ORNL Future Technologies Group

---

### Hexagon 690 (DSP + AI)
- Quad threaded Scalar Core
- DSP + 4 Hexagon Vector Xccelerators
- New Tensor Xccelerator for AI
- Apps: AI, Voice Assistance, AV codecs

### Adreno 640
- Vulkan, OpenCL, OpenGL ES 3.1
- Apps: HDR10+, HEVC, Dolby, etc
- Enables 8k-360° VR video playback
- 20% faster compared to Adreno 630

### Kyro 485 (8-ARM Prime+BigLittle Cores)
- Snapdragon X24 LTE (855 built-in) modem LTE Category 20
- Snapdragon X50 5G (external) modem (for 5G devices)
- Qualcomm Wi-Fi 6-ready mobile platform: (802.11ax-ready, 802.11ac Wave 2, 802.11ay, 802.11ad)
- Qualcomm 60 GHz Wi-Fi mobile platform: (802.11ay, 802.11ad)
- Bluetooth Version: 5.0
- Bluetooth Speed: 2 Mbps
- High accuracy location with dual-frequency GNSS.

---

### Spectra 360 ISP
- New dedicated Image Signal Processor (ISP)
- Dual 14-bit CV-ISPs; 48MP @ 30fps single camera
- Hardware CV for object detection, tracking, stereo depth process
- 6DoF XR Body tracking, H265, 4K60 HDR video capture, etc.

### Connectivity (5G)
- Snapdragon X24 LTE (855 built-in) modem LTE Category 20
- Snapdragon X50 5G (external) modem (for 5G devices)
- 5G
- 7nm TSMC
- 20% faster compared to Adreno 630
- Quad threaded Scalar Core
- DSP + 4 Hexagon Vector Xccelerators
- New Tensor Xccelerator for AI
- Apps: AI, Voice Assistance, AV codecs

---

For more information or to apply for an account, visit [https://excl.ornl.gov/](https://excl.ornl.gov/)

**Created by Narasinga Rao Miniskar, Steve Moulton**
RISC-V

RISC-V Ecosystem

Open-source software:
Gcc, binutils, glibc, Linux, BSD, LLVM, QEMU, FreeRTOS, ZephyrOS, LiteOS, SylixOS, ...

Commercial software:
Lauterbach, Segger, Micrium, ExpressLogic, ...

Software

ISA specification  Golden Model  Compliance

RISC-V Foundation

Hardware

Open-source cores:
Rocket, BOOM, RISCY, Ariane, PicoRV32, Piccolo, SCR1, Hummingbird, ...

Commercial core providers:
Andes, Bluespec, Cloudbear, Codasip, Cortus, C-Sky, Nuclei, SiFive, Syntacore, ...

Inhouse cores:
Nvidia, +others
Applications

Ontologies include Motifs, Parallel Execution Patterns, Locality, Affinity, Sync, etc.

Programming Systems

Performance Models combine Ontologies and Machine Models to produce both static and dynamic cost estimates.

Machine Models include architecture configuration information, microbenchmarks, and API for PFU.

DSSoC Hardware

Performance Parameters: CPU, GPU, FPGA, Accelerator, DSP, Deep Memory, Neuromorphic
• Signal processing: An open-source implementation of IEEE-802.11 WIFI a/b/g with GNR OOT modules.
• Input / Output file support via Socket PDU (UDP server) blocks
• Image/Video transcoding with OpenCL/OpenCV
**Preliminary SDR Application Profiling:**
- Created fully automated GRC profiling toolkit
- Ran each of the 89 flowgraph for 30 seconds
- Profiled with performance counters
- Major overheads:
  - Python glue code (libpython), O/S threading & profiling (kernel.kallsyms, libpthread), libc, ld, Qt
- Runtime overhead:
  - Will require significant consideration when run on SoC
  - Cannot be executed in parallel
  - Hardware assisted scheduling is essential

### Applications Profiling Library

<table>
<thead>
<tr>
<th>Library</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>[kernel.kallsyms]</td>
<td>27.8547</td>
</tr>
<tr>
<td>libpython</td>
<td>18.6281</td>
</tr>
<tr>
<td>libgnuradio</td>
<td>11.7548</td>
</tr>
<tr>
<td>libc</td>
<td>7.7503</td>
</tr>
<tr>
<td>ld</td>
<td>3.8839</td>
</tr>
<tr>
<td>libvolk</td>
<td>3.7963</td>
</tr>
<tr>
<td>libperl</td>
<td>3.7837</td>
</tr>
<tr>
<td>[unknown]</td>
<td>3.6465</td>
</tr>
<tr>
<td>libQt5</td>
<td>2.9866</td>
</tr>
<tr>
<td>libpthread</td>
<td>2.1449</td>
</tr>
</tbody>
</table>

### libgnuradio CPU-time Breakdown

- libgnuradio-analog: 28%
- libgnuradio-blocks: 4%
- libgnuradio-channels: 13%
- libgnuradio-digital: 6%
- libgnuradio-dtv: 3%
- libgnuradio-fec: 2%
- libgnuradio-fft: 10%
- libgnuradio-frequencies: 8%
- [unknown]: 0%
- libgnuradio-frontend: 22%
• **GNR-Tools**
  
  • **PY1Q2: Three tools are released**
    
    • Block-level Ontologies [ontologyAnalysis]
      
      • Following properties are extracted from a batch of block definition files: Descriptions and IDs, source and sink ports (whether input/output is scalar, vector or multi-port), allowed data types, and additional algorithm-specific parameters
    
    • Flowgraph Characterization [workflowAnalysis]
      
      • Characterization of GNR workloads at the flowgraph level.
      
      • Scripts automatically run for for 30 seconds and reports a breakdown of high-level library module calls
    
    • Design-space Exploration [designSpaceCL]
      
      • Script to run 13 blocks included in gr-enabled
        - Both on a GPU and on a single CPU core
        - By using input sizes varying between 24 and 227 elements.
  
  • **PY1Q3: Two more tools are added**
    
    • cgran-scraper
    
    • GRC-analyzer

https://code.ornl.gov/fub/gnr-tools
GRC statistics: Block Proximity Analysis

Block proximity analysis

- Creates a graph:
  - **Nodes**: Unique block types
  - **Edges**: Blocks used in the same GRC file.
  - Every co-occurrence increases edge weight by 1.
- This example was run
  - With `--mode proximityGraph`
  - On randomly selected sub-set of GRC files
Integrating Modeling Across the Stack with Aspen

Applications
- Streaming (e.g., JW Radio)
- Sensing (e.g., SAR, vision)
- Deep learning (e.g., CNN)
- Analytics (e.g., graphs)
- Robotics (e.g., sense and react)
- Science and Engineering (e.g., CFD)

Ontologies include Motifs, Parallel Execution Patterns, Locality, Affinity, Sync, etc.

Parameterizable Performance Models
- Concurrency
- Work
- Communication
- Capacities
- Synchronization

Performance Models combine Ontologies and Machine Models to produce both static and dynamic cost estimates

Programming Systems
- Compiler
- DSL
- JIT
- Libraries
- etc

Runtime and Operating Systems
- Task scheduling
- Memory Mgt
- IO
- Synchronization

Machine Models include architecture configuration information, microbenchmarks, and API for PFU

DSSoC Hardware
- Performance Parameters
  - CPU
  - GPU
  - FPGA
  - Accelerator
  - DSP
  - Deep Memory
  - Neuromorphic
Aspen: Abstract Scalable Performance Engineering Notation

Aspen: Abstract Scalable Performance Engineering Notation

• Modular
• Sharable
• Composable
• Reflects program structure

E.g., MD, UHPC CP 1, Lulesh, 3D FFT, CoMD, VPFFT, …

Source code


Model Creation

• Static analysis via compiler, tools
• Empirical, Historical
• Manual (for future applications)

Model Uses

• Interactive tools for graphs, queries
• Design space exploration
• Workload Generation
• Feedback to Runtime Systems

Representation in Aspen

• Modular
• Sharable
• Composable
• Reflects program structure

Aspen code

E.g., MD, UHPC CP 1, Lulesh, 3D FFT, CoMD, VPFFT, …

Model Uses

Interactive tools for graphs, queries
Design space exploration
Workload Generation
Feedback to Runtime Systems
model demod-uhd-sync{
    import uhd from "gr-uhd.aspen"
    import blocks from "gr-blocks.aspen"
    import digital from "gr-digital.aspen"

    kernel main {
        call uhd.usrp_source()
        call digital.fill_band_edge_cc()
        call digital.correlate_and_sync_cc()

        parallel {
            sequence{
                parallel {
                    call blocks.complex_to_mag()
                    call blocks.complex_to_float()
                }
                call blocks_file_sink()
            }
            sequence{
                call digital.pfb_clock_sync_xxx()
                call digital.costas_loop_cc()
                parallel {
                    call blocks.file_sink()
                    call blocks.file_sink()
                }
            }
        }
    }
}

model blocks {
    kernel complex_to_mag {
        param aspen_param_default = 1
        param aspen_param_sizeof_float = 4
        param noutput_items = 8192
        param aspen_param_sizeof_FComplex = 8
        execute [noutput_items] "block_clComplexToMag_kernel14"
        flops [1] as integer
        execute "block_clComplexToMag_kernel14__intracommIN"
        intracomm [(aspen_param_sizeof_FComplex*noutput_items)] as copyin
        map [noutput_items] "mapblock_clComplexToMag_kernel14"
        execute "block_clComplexToMag_kernel17" {
            loads [(1*aspen_param_sizeof_FComplex)] as stride(1)
            loads [(1*aspen_param_sizeof_FComplex)] as stride(1)
            stores [(1*aspen_param_sizeof_float)] as stride(1)
            flops [4] as dp, simd
        }
        execute "block_clComplexToMag_kernel14__intracommOUT"
        intracomm [(aspen_param_sizeof_float*noutput_items)] as copyout
    }
    kernel to_float {
        ...
    }
}

GNURadio Flowgraph to Aspen Application Model Conversion
**Graph-Based Abstract Machine Model**

```plaintext
class Zynq::Board-ZCU102 : Aspen::CompoundNode {
    // Processing units
    Zynq::APU cpu;
    ARM::Mali400MP2 gpu;
    ARM::CortexR5 rpu;
    Xilinx::UltraScale<nFPUs=400M> fpga;

    // Memory
    Aspen::DDR3<freq=2000MHZ, CL=16> systemMemory;
    Aspen::Switch<bw=100GBs, latency= 25ns> lpSwitch;
    Aspen::Switch<bw=1TBs, latency= 35ns> centralSwitch;
    Aspen::PCIController<ver=3, totalLanes=24> pciController;

    // Define interconnects (edges)
    @add
    cpu -- cci_fp -- smmu;
    gpu -- cci_fp -- smmu;
    fpga -- cci_fp -- smmul
    systemMemory -- cci_fp[2] -- smmu; // Multiple links
    smmu -- cci_fp -- centralSwitch;
    smmu -- cci_fp -- pciController;
    fpga -- cci_fp[2] -- centralSwitch
    lpSwitch -- cci_fp[2] -- smmu; // Unidirectional link
    lpSwitch << cci_fp -- centralSwitch
    rpu -- cci_lp[2] -- lpSwitch;
    pciController -- pciBus --> Aspen::OUTPUT
    pciController << pciBus --> Aspen::INPUT
```

Belviranli, M E, et al, "FLAME: Graph-based Hardware Representations for Rapid and Precise Performance Modeling", DATE’19
Programming Systems

Aspen
- Ontologies
  - Structured
  - Composable
  - Interoperable
- Parameterizable Performance Models
  - Concurrency
  - Work
  - Communication
  - Capacities
  - Synchronization
- Machine Model
  - Compute
  - Memory
  - Interconnects
- PFU API

Applications
- Streaming (e.g., SW/Radio)
- Sensing (e.g., SAR, vision)
- Deep learning (e.g., CNN)
- Analytics (e.g., graphs)
- Robotics (e.g., sense and react)
- Science and Engineering (e.g., CFD)

Ontologies include Motifs, Parallel Execution Patterns, Locality, Affinity, Sync, etc.

Programming Systems
- Compiler
- DSL
- JIT
- Libraries
- etc

Programming Systems combine Ontologies and Machine Models to produce both static and dynamic cost estimates.

Runtime and Operating Systems
- Task scheduling
- Memory Mgt
- IO
- Synchronization

Runtime and Operating Systems include architecture configuration information, microbenchmarks, and API for PFU.

DSSoC Hardware
- Performance Parameters
  - CPU
  - GPU
  - FPGA
  - Accelerator
  - DSP
  - Deep Memory
  - Neuromorphic
OpenARC: Open Accelerator Research Compiler*

- OpenARC is the first open-sourced, OpenACC/OpenMP compiler supporting Altera FPGAs, in addition to NVIDIA/AMD GPUs and Intel Xeon Phis.
- OpenARC is a high-level intermediate representation based, extensible compiler framework, where various performance optimizations, traceability mechanisms, fault tolerance techniques, etc., can be built for the complex heterogeneous computing.

* OpenARC, Lee, HPDC ‘14.
Heterogeneous Architecture Support by OpenARC

OpenACC/OpenMP Program

Front End Compiler

Back End Compiler

CUDA ➔ Nvidia GPUs
OpenCL ➔ AMD GPUs
OpenCL ➔ Xeon Phis
Intel OpenCL ➔ Intel FPGAs
MCL ➔ DSSoCs
OpenMP ➔ CPUs/Xeon Phis

OpenACC/OpenMP P Interoperability
Initial Performance of OpenACC GNU Radio Blocks on Xavier

- Ported the OpenACC version of the GNU Radio blocks to OpenMP3 (CPU target), CUDA (GPU target), and OpenCL (GPU target) and compared against the reference CPU version.

- Tested Platform: NVIDIA Jetson Xavier (8-core ARM CPU, NVIDIA Volta GPU, two NVDLA Engines and VLIW Vision Processor)

Omitted MCL since JIT compilation not yet factored
Program Characteristics Prediction of GNU Radio Blocks using Aspen

- OpenARC automatically generates a structured Aspen performance model from the ported OpenACC code of the GNU Radio blocks.
- Aspen performance prediction tools digest the generated Aspen models and derive performance predictions for the target application.

COMPASS: A Framework for Automated Performance Modeling and Prediction
Programming Systems

Applications
- Streaming (e.g., IW Radio)
- Sensing (e.g., SAR, vision)
- Deep learning (e.g., CNN)
- Analytics (e.g., graphs)
- Robotics (e.g., sense and react)
- Science and Engineering (e.g., CFD)

Ontologies include Motifs, Parallel Execution Patterns, Locality, Affinity, Sync, etc.

Programming Systems
- Compiler
- DSL
- JIT
- Libraries
- etc

Runtime and Operating Systems
- Task scheduling
- Memory Mgt
- IO
- Synchronization

Machine Models include architecture configuration information, microbenchmarks, and API for PFU

DSSoC Hardware
- Performance Parameters
- CPU
- GPU
- FPGA
- Accelerator
- DSP
- Deep Memory
- Neuromorphic

Parameterizable Performance Models
- Concurrency
- Work
- Communication
- Capacities
- Synchronization

Aspen
- Ontologies
  - Structured
  - Composable
  - Interoperable

10. Parameterizable Performance Models
11. Machine Models
   - Compute
   - Memory
   - Interconnects

15. Programming Systems
16. Applications
17. DSSoC Hardware
18. Aspens
Cosmic Runtime and Scheduler

- Framework for programming extremely heterogeneous systems
  - Programming model and programming model runtime
  - Maximize resource utilizations
  - Abstract low-level architecture details from programmers
  - Dynamically schedule work to available resources

- Key programming features:
  - Scheduler dispatches application tasks to available computing resources
  - Asynchronous execution of runnable tasks
  - Devices are managed by the scheduler and presented as “Processing Elements” to users
  - Independent applications submit tasks without having to synchronize with each other
  - Simplified APIs and programming model (e.g., compared to OpenCL)

- Flexibility:
  - Provides a scheduling framework in which new scheduling algorithms can be plugged in
  - Multiple scheduling algorithms co-exist
  - Users don’t need to port code when running on different systems
  - Executing tasks on different PEs doesn’t require user intervention or code modification
  - Resources allocated at the last moment
Exploiting Parallelism in SDR

- **Task Parallelism**
- **Data Parallelism**
- **Application Parallelism**
- **Pipelining**

**Diagram Details**
- **Task**
- **Parallelism**
- **Application**
- **Data**
• ISR assumes that user application ...
  • Written in high level, performance portable programming model (e.g., OpenACC or OpenCL if nec)
    • OpenARC – presented in last section
  • Has target code versions generated appropriate versions of target system code
    • JIT is possible for OpenCL targets (except FPGA)

• ISR contains specific RT modules for each device

• ISR sets up dependencies as specified by compiler, user

• ISR creates catalog of data to orchestrate data movement across disparate device memories
During execution, ISR must:
- Discover available devices
- Pick the most appropriate device for the task
- Maintain dependencies
- Orchestrate data movement

Device selection uses any number of policies:
- Random, Round-robin, profiling, hints, ontology, performance models (Aspen)
- ISR must also monitor existing device work to make tradeoffs

Current support for:
- AMD GPU
- NVIDIA GPU
- CPU
- Xeon Phi
- Intel FPGA
OpenCL provides a good compatibility layer but doesn’t provide sufficient introspective feedback:

- Performance counters
- Interference counters
- Power/Energy metrics
- Temperature sensors
Recap

• Motivation: Recent trends in computing paint an ambiguous future
  – Multiple architectural dimensions are being (dramatically) redesigned: Processors, node design, memory systems, I/O
  – Complexity is our main challenge

• Applications and software systems across many areas are all reaching a state of crisis
  – Need a focus on performance portability

• ORNL Cosmic project investigating design and programming challenges for these trends in SDR
  – Performance modeling and ontologies
  – Performance portable compilation to many different heterogeneous architectures/SoCs
  – Intelligent scheduling system to automate discovery, device selection, and data movement
  – Targeting wide variety of existing and future architectures (DSSoC and others)

• Visit us
  – We host interns and other visitors year round
    • Faculty, grad, undergrad, high school, industry

• Jobs in FTG
  – Postdoctoral Research Associate in Computer Science
  – Software Engineer
  – Computer Scientist
  – Visit [https://jobs.ornl.gov](https://jobs.ornl.gov)

• Contact me vetter@ornl.gov
Bonus Material
ASCR Extreme Heterogeneity Workshop
January 23-25, 2018 Virtual Meeting

• Goal: Identify Priority Research Directions for Computer Science needed to make future supercomputers usable, useful and secure for science applications in the 2025-2040 timeframe
  – Note that quantum computing was defined as out of scope by ASCR.

• Primary focus on the software stack and programming models/environments/tools

• 150+ participants: DOE labs, academia, and industry

• White papers solicited (106 received!) to contribute to the FSD, identify potential participants, and help refine the agenda

• First ASCR workshop to use Basic Research Needs format (BES inspired)
  – Summit, Summit report, Factual Status Document, whitepapers, BRN/PRD result

• Organizing Committee
  – Jeffrey Vetter (ORNL), Lead Organizer and Program Committee Chair
  – Ron Brightwell (Sandia-NM), Pat McCormick (LANL), Rob Ross (ANL), John Shalf (LBNL)
  – Lucy Nowell, ASCR Program Manager

• Program Committee Members
  – Katy Antypas (LBNL, NERSC), David Donofrio (LBNL), Maya Gokhale (LLNL), Travis Humble (ORNL), Catherine Schuman (ORNL), Brian Van Essen (LLNL), Shinjae Yoo (BNL)
The Future Technologies Group performs research in core technologies for emerging generations of high-end computing architectures, including prototype computer architectures and experimental software systems. We investigate these technologies with the goal of improving the performance, energy efficiency, reliability, and productivity of these architectures for our sponsors and applications teams. See [http://ft.ornl.gov](http://ft.ornl.gov).

### Key Technical Areas
- Heterogeneous architectures
- Deep memory hierarchies including non-volatile memory
- Performance measurement, analysis, simulation, and modeling of emerging architectures.
- Programming systems to address emerging architectures
- Beyond Moore's Computing

### Software Artifacts
- Scalable Heterogeneous Computing Benchmarks (SHOC)
- mpip
- DESTINY
- Aspen
- OpenARC
- Papyrus
- NVL-C
- Oxbow
- LLVM Clacc and Parallel IR
- DRAGON
- RISC-V Extensions

### Sponsors
- DOE ASCR, BER
- DOE Exascale Computing Project
- DOE SciDAC
- DARPA
- ORNL LDRD
- National Science Foundation
- Department of Defense
- NIH

### Impact
- Publications in SC, ICS, HPDC, TPDS, DATE, PLDI, IPDPS, Trans VLSI, etc.
- Two Gordon Bell awards
- NSF Keeneland
- DOE Titan
- IEEE TCHPC Early Career
- IEEE Fellows
- ~100 interns
- ~130 FTG seminars

See [https://www.thebroadcastbridge.com/content/entry/1094/altera-arriva-10-2666mbps-ddr4-memory-fpga-interface.html](https://www.thebroadcastbridge.com/content/entry/1094/altera-arriva-10-2666mbps-ddr4-memory-fpga-interface.html)
Progression of Experimental Computing Technologies

**TRL 1-3 Basic Concepts**
- Examples: carbon-nanotube computing, memristor-based neuromorphic computing, chip-level silicon photonics, universal quantum computing

**TRL 4-6 Emerging**
- Examples: FPGAs in HPC, TrueNorth, SpiNNaker, D-Wave, Emu, many SoC-based systems, TPU, Gen-Z NoCs, near-memory computing

**TRL 7-9 Operational**
- Examples: Titan, Cori, Mira, Summit, BlueWaters, Keeneland, Stampede, Tsubame2.5

---

### Progression of Experimental Computing Technologies

<table>
<thead>
<tr>
<th>Programming</th>
<th>Assembly language, or less</th>
<th>Few, if any, development tools</th>
<th>Language support and compilers.</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS-R</td>
<td>Manual</td>
<td>Specialized programming environments and OSs</td>
<td>Commodity OS &amp; runtime systems</td>
</tr>
<tr>
<td>Scale</td>
<td>Small collections of devices</td>
<td>Single to hundreds of engineered processing elements</td>
<td>&gt;10,000 processing elements</td>
</tr>
<tr>
<td>Performance</td>
<td>Analytical projections based on device empirical evaluation.</td>
<td>Analytical projections or simulation based on component or pilot system empirical evaluation.</td>
<td>Empirical evaluation of prototype and final systems.</td>
</tr>
<tr>
<td>Apps</td>
<td>Small encoded kernels</td>
<td>Architecture-aware algorithms; Mini-apps; Small applications</td>
<td>Numerical libraries; Full scale applications</td>
</tr>
<tr>
<td>Example</td>
<td>GPUs invented in 1999</td>
<td>OpenGL in 2001; CUDA in 2007; OpenCL in 2008; OpenACC in 2010; DP in 2010; ECC in 2012</td>
<td>GPUs are a fully supported compute technology in the HPC ecosystem.</td>
</tr>
</tbody>
</table>
ORNL ExCL Model

https://excl.ornl.gov

ExCL Common Infrastructure

- Project and User management
  - Accounts
  - Projects and Proposals
  - Help

- Shared Login and Gateway Nodes
  - Gateway nodes
  - Data transfer nodes
  - Consistent and secure access to private network compartments

- Shared Filesystems and Databases
  - Secure access to filesystems across pillars

- Source Code and Data sets
  - Source Code repos
  - Performance databases for applications and architectures

- Community
  - Workshops
  - Online discussions forums and issues
  - Consolidated
  - News

- Authentication and Authorization
  - Secure operations
  - Partition access to specific compartments
  - System and account lifecycles
  - Experience with management of export controlled and proprietary systems

- Monitoring and control systems
  - Manage access to shared resources
  - Manage privileged access levels
  - Lights out operation

- Web
  - Educational and reference materials
  - Outreach
  - Both Open and Controlled access

- Shared Filesystems and Databases
  - Secure access to filesystems across pillars

- Web
  - Educational and reference materials
  - Outreach
  - Both Open and Controlled access

ExCL Technology Pillars

- GPU: NVIDIA PASCAL, VOLTA
- FPGA: Intel Arria 10, Stratix 10, Xilinx U250
- NVM: Intel Optane, Apache Pass
- Deep memory: HBM2
- SoC: ThunderX2, Zynq
- Data intensive: Emu
- Cloud: OpenStack Cluster
- Cryogenic devices: JJ memory cell
- Neuromorphic: TrueNorth
- Quantum: Rigetti, IBM, D-wave
- Deep Learning

This year’s hot item

Pillars refreshed annually

Per pillar expert collaboration