Preparing for Extreme Heterogeneity in High Performance Computing

Jeffrey S. Vetter

With many contributions from ACSR Section and Colleagues

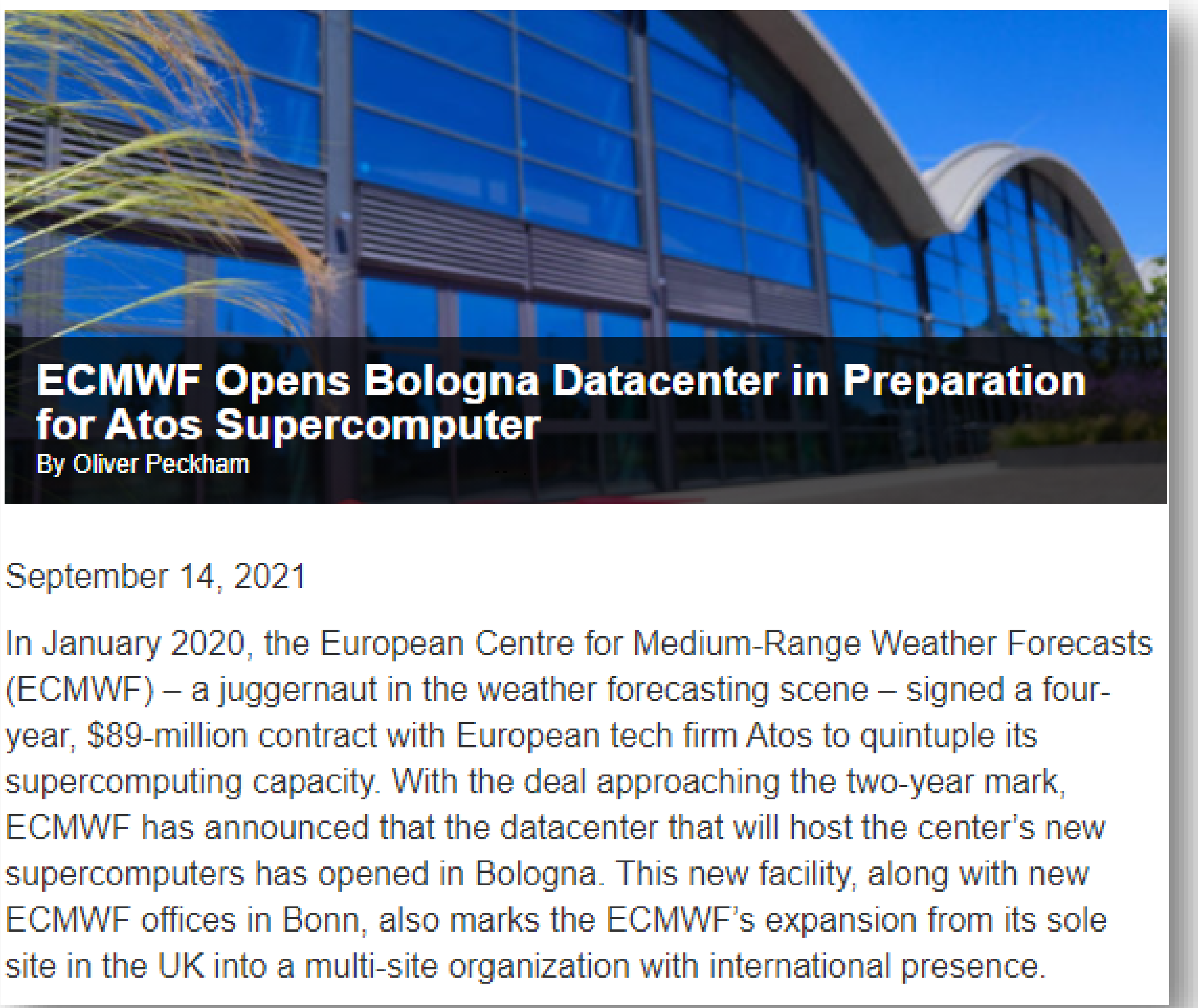
19th Workshop on HPC in Meteorology
21 Sep 2021
ECMWF (Virtual)

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

https://www.ornl.gov/section/advanced-computing-systems-research (https://j.mp/acsrs)
vetter@computer.org

Congratulations on your new data center!

Wish I could be there in person to chat and for a tour ☺



ECMWF Opens Bologna Datacenter in Preparation for Atos Supercomputer

By Oliver Peckham

September 14, 2021

In January 2020, the European Centre for Medium-Range Weather Forecasts (ECMWF) – a juggernaut in the weather forecasting scene – signed a four-year, $89-million contract with European tech firm Atos to quintuple its supercomputing capacity. With the deal approaching the two-year mark, ECMWF has announced that the datacenter that will host the center's new supercomputers has opened in Bologna. This new facility, along with new ECMWF offices in Bonn, also marks the ECMWF's expansion from its sole site in the UK into a multi-site organization with international presence.

Highlights

Recent trends in computing paint an ambiguous future for architectures

- Power constraints initially drove architectural changes
- Now, vendors are forced to use 2-3 foundries if they want access to leading-edge CMOS production
  - Forces vendors to add value with domain specific architectures by specializing processors, node design, memory systems, I/O
- Explosion of new architectures
  - Devices: GPUs, FPGAs, DSPs, SoCs
  - Deployment: HPC, AI, Edge, Cloud
  - OpenHW: RISC-V
- **Entering an era of Extreme Heterogeneity**

Extreme Heterogeneity 2018

PRODUCTIVE COMPUTATIONAL SCIENCE
IN THE ERA OF EXTREME HETEROGENEITY

Report for
DOE ASCR Workshop on Extreme Heterogeneity
January 23–25, 2018
Version August 27, 2018

https://doi.org/10.2172/1473756

As a result, applications and software systems are all reaching a state of crisis

- Proliferation of diverse and often immature programming ecosystems
  - In fact, programming and operating systems need major investment to address current and future architectural changes
- Applications will not be *functionally* or performance portable across architectures
- Additionally, procurements, acceptance testing, and operations of today's new platforms depend on performance prediction and benchmarking.
- **Complexity is our main challenge**
- **This is a crisis!**

Programming systems must provide performance portability (beyond functional portability)!!

- Ultimately, we should strive for '*Write once, perform satisfactorily anywhere*'
  - Descriptive models of parallelism and data movement
  - Introspective runtime systems
  - Layered, modular, open-source approaches required
  - Performance prediction tools for design, procurement, and operations
- **Examples**
  - ECP investments in LLVM
    - FORTRAN with GPU offloading
  - Introspective Runtime Systems
  - Programming FPGAs
    - Without Verilog

Time for a short poll…

History (circa 2010)

Q: Think back 10 years.

How many of you would have predicted that many of our top HPC systems would be heterogeneous (GPU-based) architectures?

Yes

No

Revisionists ☺

Future (circa 2030)

Q: Think forward 10 years.

How many of you predict that our top 100 HPC systems will have the following architectural features?

Assume general purpose multicore CPU

- GPU
- FPGA/Reconfigurable processor
- Neuromorphic processor
- Deep learning processor
- Quantum processor
- RISC-V processor
- Some new unknown processor
- All/some of the above in one SoC

Implications for Applications Teams

Q: Now, imagine you are building a new application with an expected ~3M LOC and 20 team members over the next 10 years.

What on-node programming model/system do you use to future-proof your app?

Assume C and C++

- Fortran XX
- CUDA, cu***, HIP, OpenCL
- Directives: OpenMP, OpenACC
- Python, Julia, Rust, R, Matlab, etc
- Metaprogramming, DSEL, etc (e.g., AMP, Kokkos, RAJA, SYCL)
- Domain Specific Language (e.g., Claw, Hallide, PySL) or Domain Specific Framework (e.g., PetSc, AMReX)
- Legion, Charm++, HPX, OmpSs, Star-P, etc
- Some new unknown programming approach
- Some combination of the above

Motivating Trends

News & Analysis

Foundries' Sales Show Hard Times Continuing

Peter Clarke

5/23/2016 09:33 P

2 comments

Like 6 4

N
1
L

LONDON--Taiwan … partial
semiconductor se … ales f
market slow down … yet ov

Both companies a … signi
those they achieve … were a
an annual basis in … ators
TSMC and UMC a … e of s

eetasia.com

SEMICONDUCTOR ENGINEERING

Home > Manufacturing, Design & Test > Uncertainty Grows For 5nm, 3nm

MANUFACTURING, DESIGN & TEST

Uncertainty Grows For 5nm, 3nm

797 74

Nanosheets and nanowire FETs under development, but costs are skyrocketing. New packaging options could provide an alternative.

DECEMBER 19TH, 2016 - BY: MARK LAPEDUS

As several chipmakers ramp up their 10nm finFET processes, with 7nm just around the corner, R&D has begun for 5nm and beyond. In fact, some are already moving full speed ahead in the arena.

TSMC recently announced plans to build a new fab in Taiwan

Samsung t… Foundry &… 2030

GlobalFoundries Forfeit 7nm Manufacturing - EE Times Asia

6-7 minu

SAN …
the bl…

Globa…
than …
subsi…

US Government's Aurora Supercomputer Delayed Due to Intel's 7nm Setback

By Anton Shilov 18 days ago

Intel-powered exascale supercomputer will come online later than expected.

Comments (7)

(Image credit: Argonne National Laboratory)

When Intel announced earlier this year that its 7nm process technology would be delayed, it brought implications for Aurora, the first Intel-based exascale supercomputer. There was no clear answer back in July, but an official for the U.S. Department of Energy's (DoE) Office of Science confirmed this week that the

Intel's 10nm Is Broken, Delayed Until 2019

by Paul Alcorn April 26, 2018 at 6:30 PM

DESIGNLINES | WIRELESS AND NETWORKING DESIGNLINE

GlobalFoundries Selling A… Business to Marvell

By Dylan McGrath, 05.20.19 1

…business to
…ree-phase

Another Step Toward the End of Moore's Law

Samsung and TSMC move to

Number of Semiconductor Manufacturers with a Cutting Edge Logic Fab

SilTerra										
X-FAB										
Dongbu HiTek										
ADI	ADI									
Atmel	Atmel									
Rohm	Rohm									
Sanyo	Sanyo									
Mitsubishi	Mitsubishi									
ON	ON									
Hitachi	Hitachi									
Cypress	Cypress	Cypress								
Sony	Sony	Sony								
Infineon	Infineon	Infineon								
Sharp	Sharp	Sharp								
Freescale	Freescale	Freescale								
		Renesas	Renesas	Renesas						
		Toshiba	Toshiba	Toshiba						
		Fujitsu	Fujitsu	Fujitsu						
		TI	TI	TI						
		Panasonic	Panasonic	Panasonic	Panasonic					
		STM	STM	STM	STM					
			HLMC	HLMC	HLMC					
		UMC	UMC	UMC	UMC		UMC			
		IBM	IBM	IBM	IBM	IBM				
		SMIC	SMIC	SMIC	SMIC		SMIC			
		AMD	GlobalFoundries	GF	GF	GF	GF			
		Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung
		TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC
		Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel
		90 nm	65 nm	45 nm/40 nm	32 nm/28 nm	22 nm/20 nm	16 nm/14 nm	10 nm	7 nm	5 nm

13 Dec 2019 | 20:20 GMT

TSMC's 5-Nanometer Process on Track for First Half of 2020

Devices are 15 percent faster, 30 percent more energy efficient

By Samuel K. Moore

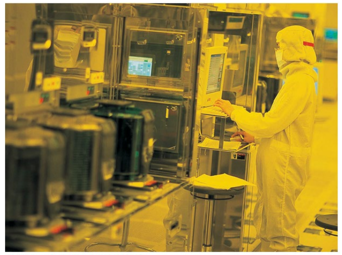
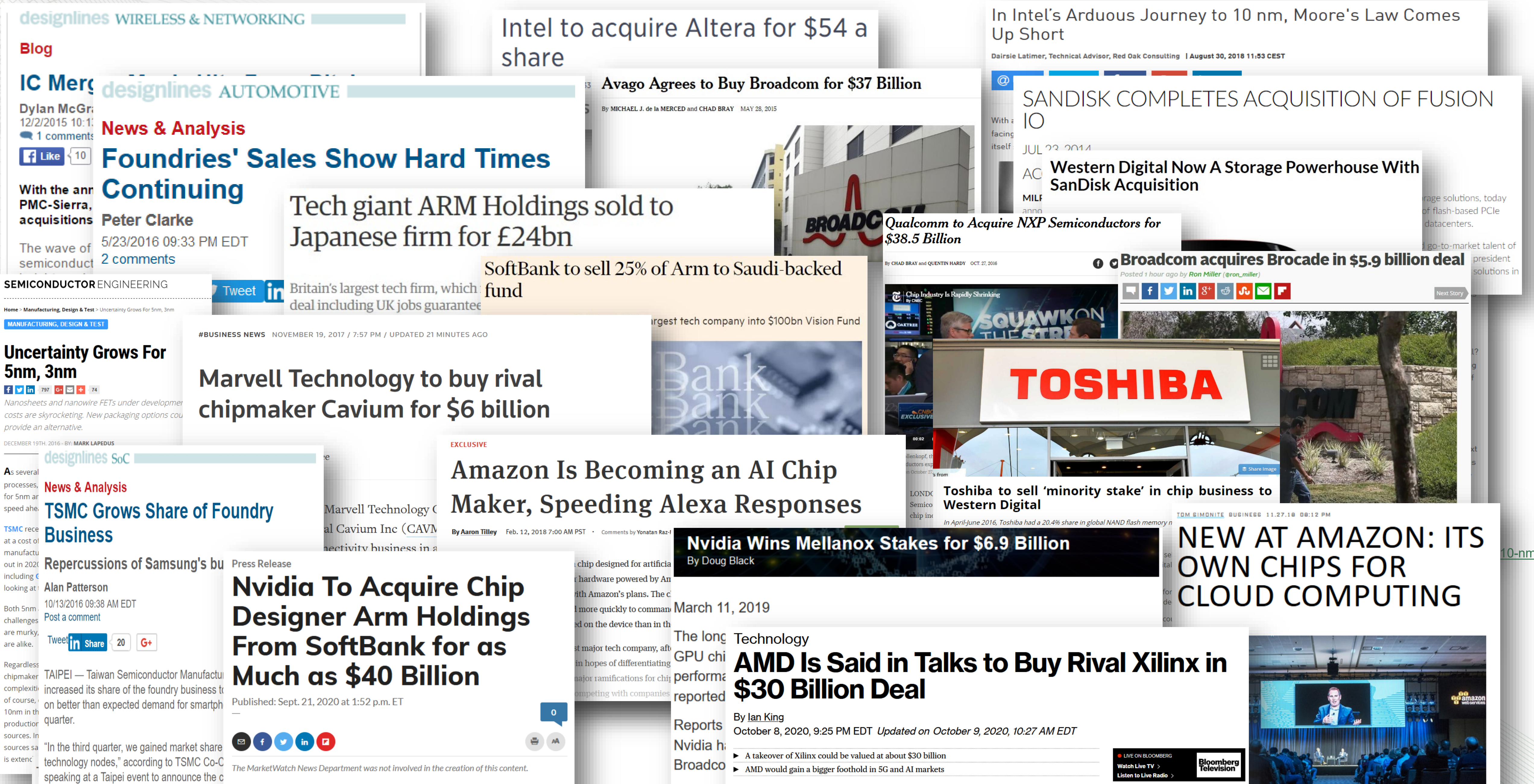


Photo: Taiwan Semiconductor Manufacturing Co.

The performance enhancement achieved by TSMC's new 5-nanometer process is partly due to the inclusion of a "high-mobility channel." How is it created? TSMC wouldn't reveal.

OAK RIDGE
National Laboratory

Business climate reflects this uncertainty, cost, complexity, consolidation



designlines WIRELESS & NETWORKING

Blog

IC Merg

Dylan McGr
12/2/2015 10:1
1 comments
Like 10

With the ann
PMC-Sierra,
acquisitions

The wave of
semiconduct

designlines AUTOMOTIVE

News & Analysis

Foundries' Sales Show Hard Times Continuing

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

Tweet

Intel to acquire Altera for $54 a share

Avago Agrees to Buy Broadcom for $37 Billion

By MICHAEL J. de la MERCED and CHAD BRAY MAY 28, 2015

In Intel's Arduous Journey to 10 nm, Moore's Law Comes Up Short

Dairsie Latimer, Technical Advisor, Red Oak Consulting | August 30, 2018 11:53 CEST

With a
facing
itself

SANDISK COMPLETES ACQUISITION OF FUSION IO

JUL 23, 2014

AC

MILP
anno

Western Digital Now A Storage Powerhouse With SanDisk Acquisition

rage solutions, today
of flash-based PCIe
datacenters.

go-to-market talent of
president
solutions in

Tech giant ARM Holdings sold to Japanese firm for £24bn

Britain's largest tech firm, which
deal including UK jobs guarantee

SoftBank to sell 25% of Arm to Saudi-backed fund

rgest tech company into $100bn Vision Fund

Qualcomm to Acquire NXP Semiconductors for $38.5 Billion

By CHAD BRAY and QUENTIN HARDY OCT. 27, 2016

Chip Industry Is Rapidly Shrinking
By CNBC

llenkopf, th
ductors exp
n October 27

LONDO
Semico
chip inc

Broadcom acquires Brocade in $5.9 billion deal

Posted 1 hour ago by Ron Miller (@ron_miller)

Next Story

Toshiba to sell 'minority stake' in chip business to Western Digital

In April-June 2016, Toshiba had a 20.4% share in global NAND flash memory m

SEMICONDUCTOR ENGINEERING

Home > Manufacturing, Design & Test > Uncertainty Grows For 5nm, 3nm

MANUFACTURING, DESIGN & TEST

Uncertainty Grows For 5nm, 3nm

797 74

Nanosheets and nanowire FETs under developmer costs are skyrocketing. New packaging options cou provide an alternative.

DECEMBER 19TH, 2016 - BY: MARK LAPEDUS

As several
processes,
for 5nm ar
speed ahe

TSMC rece
at a cost of
manufactu
out in 2020
including G
looking at t

Both 5nm
challenges
are murky,
are alike.

Regardless
chipmaker
complexiti
of course,
10nm in th
production
sources. In
sources sa
is extend

\#BUSINESS NEWS NOVEMBER 19, 2017 / 7:57 PM / UPDATED 21 MINUTES AGO

Marvell Technology to buy rival chipmaker Cavium for $6 billion

Marvell Technology
al Cavium Inc (CAVM
nectivity business in a

designlines SoC

News & Analysis

TSMC Grows Share of Foundry Business

Repercussions of Samsung's bu

Alan Patterson
10/13/2016 09:38 AM EDT
Post a comment

Tweet Share 20 G+

TAIPEI — Taiwan Semiconductor Manufactu increased its share of the foundry business t on better than expected demand for smartph quarter.

"In the third quarter, we gained market share technology nodes," according to TSMC Co-C speaking at a Taipei event to announce the c

Press Release

Nvidia To Acquire Chip Designer Arm Holdings From SoftBank for as Much as $40 Billion

Published: Sept. 21, 2020 at 1:52 p.m. ET

0

The MarketWatch News Department was not involved in the creation of this content.

EXCLUSIVE

Amazon Is Becoming an AI Chip Maker, Speeding Alexa Responses

By Aaron Tilley Feb. 12, 2018 7:00 AM PST · Comments by Yonatan Raz-F

chip designed for artificia
hardware powered by Am
ith Amazon's plans. The c
more quickly to comman
d on the device than in th

st major tech company, aft
in hopes of differentiating
major ramifications for chip
ompeting with companies

Nvidia Wins Mellanox Stakes for $6.9 Billion

By Doug Black

March 11, 2019

The long
GPU chi
performa
reported

Reports
Nvidia h
Broadco

TOM SIMONITE BUSINESS 11.27.18 08:12 PM

NEW AT AMAZON: ITS OWN CHIPS FOR CLOUD COMPUTING

Technology

AMD Is Said in Talks to Buy Rival Xilinx in $30 Billion Deal

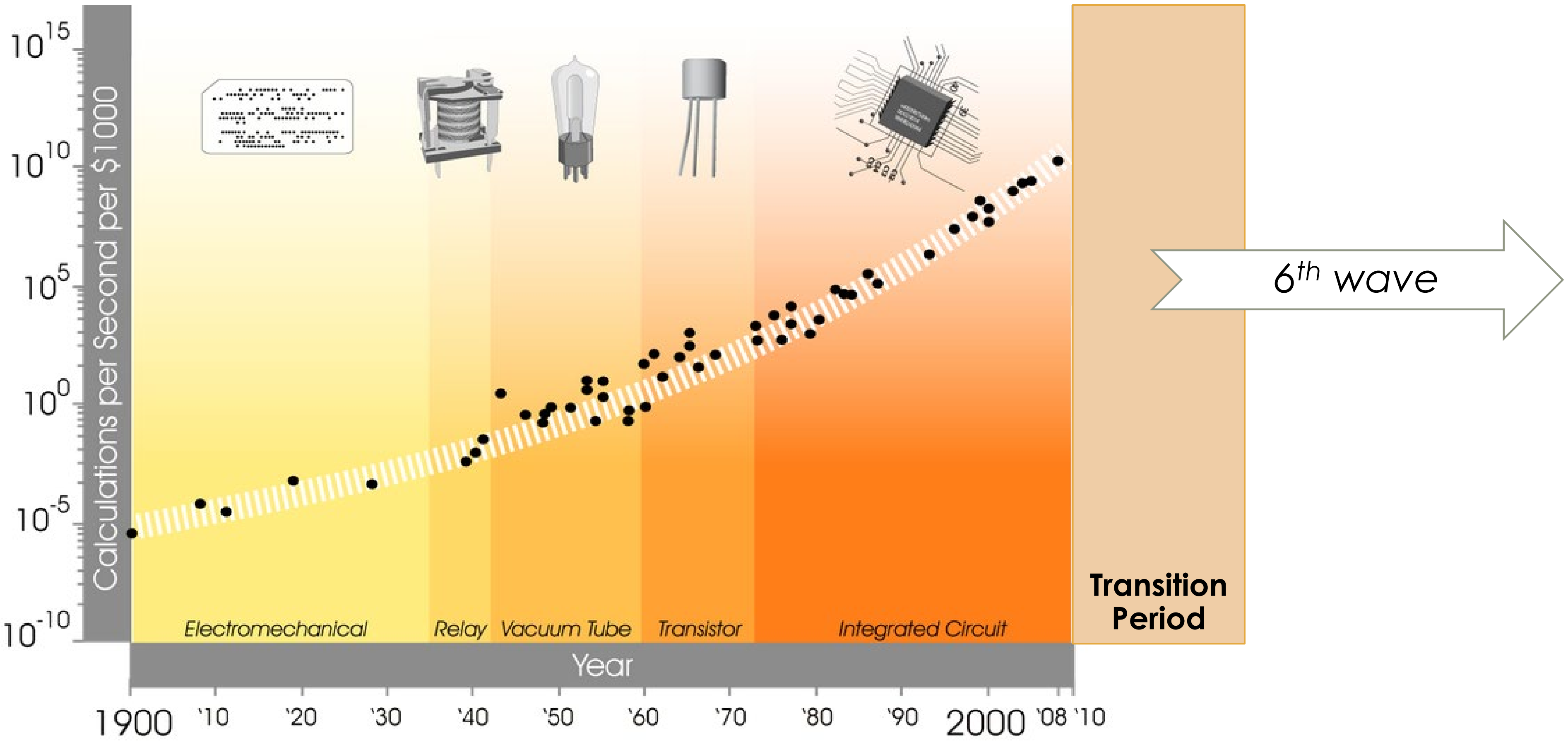
By Ian King
October 8, 2020, 9:25 PM EDT *Updated on October 9, 2020, 10:27 AM EDT*

- A takeover of Xilinx could be valued at about $30 billion
- AMD would gain a bigger foothold in 5G and AI markets

LIVE ON BLOOMBERG
Watch Live TV >
Listen to Live Radio >
Bloomberg Television

10-nm

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

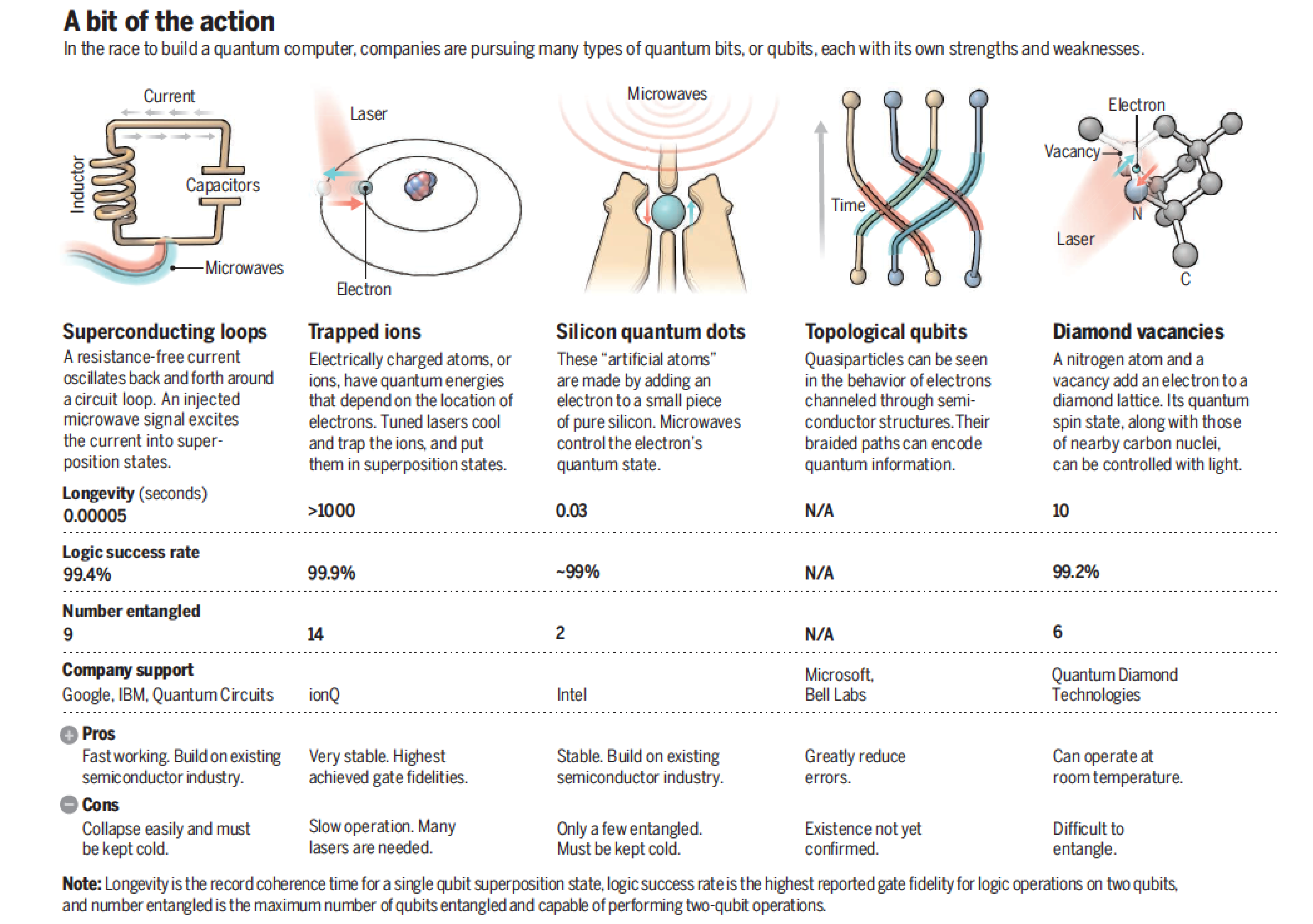
- Use CMOS more effectively for specific workloads
- Integrate components to boost performance and eliminate inefficiencies
- Workload specific memory+storage system design

Emerging Technologies

- Investigate new computational paradigms
    - Quantum
    - Neuromorphic
    - DNA Storage
    - Advanced Digital
    - Emerging Memory Devices

Quantum computing: Qubit design and fabrication have made recent progress but still face challenges

Science 354, 1091 (2016) – 2 December

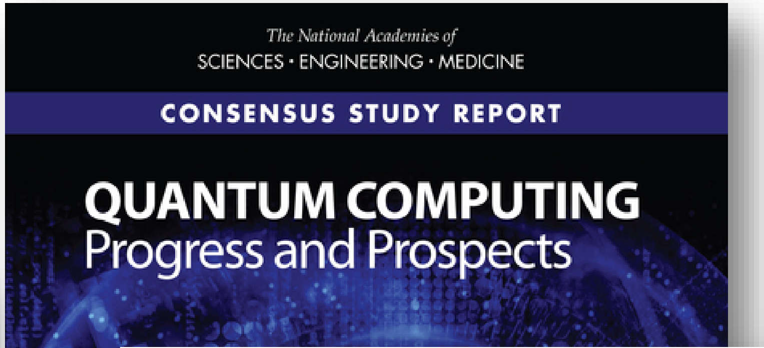

Current
Inductor
Capacitors
Microwaves
Laser
Electron
Microwaves
Time
Electron
Vacancy
Laser
N
C

A bit of the action

In the race to build a quantum computer, companies are pursuing many types of quantum bits, or qubits, each with its own strengths and weaknesses.

	Superconducting loops	Trapped ions	Silicon quantum dots	Topological qubits	Diamond vacancies
	A resistance-free current oscillates back and forth around a circuit loop. An injected microwave signal excites the current into superposition states.	Electrically charged atoms, or ions, have quantum energies that depend on the location of electrons. Tuned lasers cool and trap the ions, and put them in superposition states.	These "artificial atoms" are made by adding an electron to a small piece of pure silicon. Microwaves control the electron's quantum state.	Quasiparticles can be seen in the behavior of electrons channeled through semiconductor structures. Their braided paths can encode quantum information.	A nitrogen atom and a vacancy add an electron to a diamond lattice. Its quantum spin state, along with those of nearby carbon nuclei, can be controlled with light.
Longevity (seconds)	0.00005	>1000	0.03	N/A	10
Logic success rate	99.4%	99.9%	~99%	N/A	99.2%
Number entangled	9	14	2	N/A	6
Company support	Google, IBM, Quantum Circuits	ionQ	Intel	Microsoft, Bell Labs	Quantum Diamond Technologies
⊕ **Pros**	Fast working. Build on existing semiconductor industry.	Very stable. Highest achieved gate fidelities.	Stable. Build on existing semiconductor industry.	Greatly reduce errors.	Can operate at room temperature.
⊖ **Cons**	Collapse easily and must be kept cold.	Slow operation. Many lasers are needed.	Only a few entangled. Must be kept cold.	Existence not yet confirmed.	Difficult to entangle.

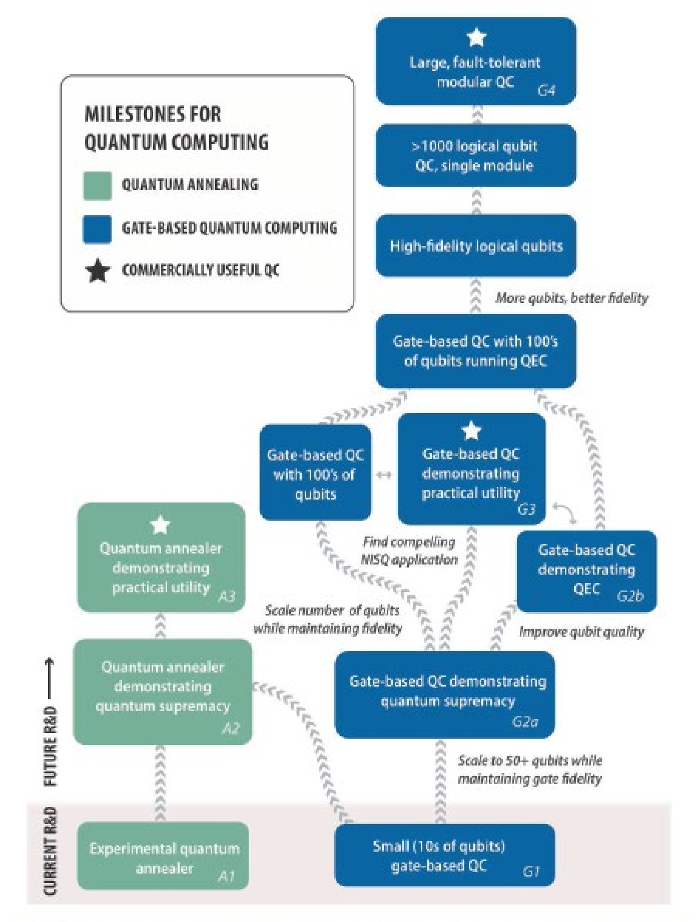
Note: Longevity is the record coherence time for a single qubit superposition state, logic success rate is the highest reported gate fidelity for logic operations on two qubits, and number entangled is the maximum number of qubits entangled and capable of performing two-qubit operations.



The National Academies of
SCIENCES · ENGINEERING · MEDICINE

CONSENSUS STUDY REPORT

QUANTUM COMPUTING
Progress and Prospects



MILESTONES FOR QUANTUM COMPUTING

🟩 QUANTUM ANNEALING

🟦 GATE-BASED QUANTUM COMPUTING

★ COMMERCIALLY USEFUL QC

★ Large, fault-tolerant modular QC G4

>1000 logical qubit QC, single module

High-fidelity logical qubits

More qubits, better fidelity

Gate-based QC with 100's of qubits running QEC

Gate-based QC with 100's of qubits

★ Gate-based QC demonstrating practical utility G3

★ Quantum annealer demonstrating practical utility A3

Find compelling NISQ application

Gate-based QC demonstrating QEC G2b

Scale number of qubits while maintaining fidelity

Improve qubit quality

Quantum annealer demonstrating quantum supremacy A2

Gate-based QC demonstrating quantum supremacy G2a

Scale to 50+ qubits while maintaining gate fidelity

FUTURE R&D

CURRENT R&D

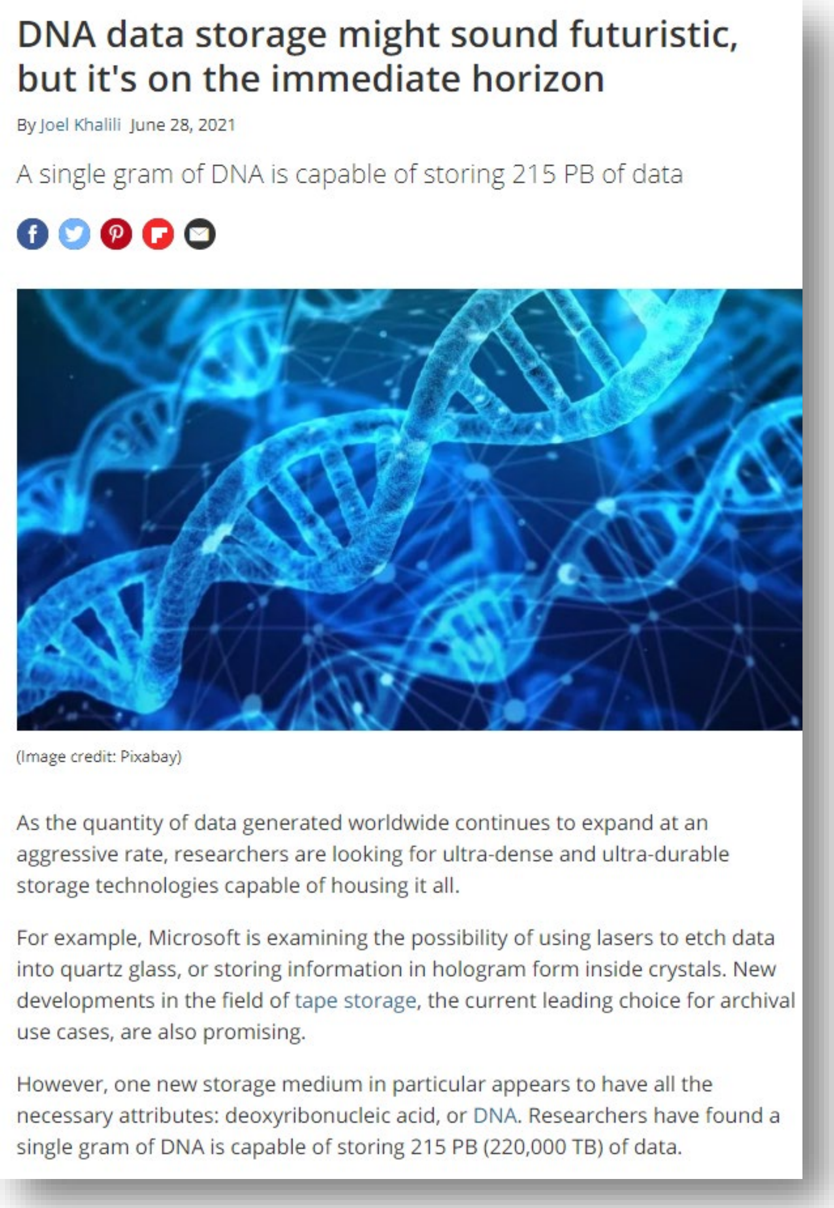
Experimental quantum annealer A1

Small (10s of qubits) gate-based QC G1

FIGURE 7.4 An illustration of potential milestones of progress in quantum computing. The arrangement of milestones corresponds to the order in which the committee thinks they are likely to be achieved; however, it is possible that some will not be achieved, or that they will not be achieved in the order indicated.

http://nap.edu/25196

DNA Storage



DNA data storage might sound futuristic, but it's on the immediate horizon

By Joel Khalili June 28, 2021

A single gram of DNA is capable of storing 215 PB of data

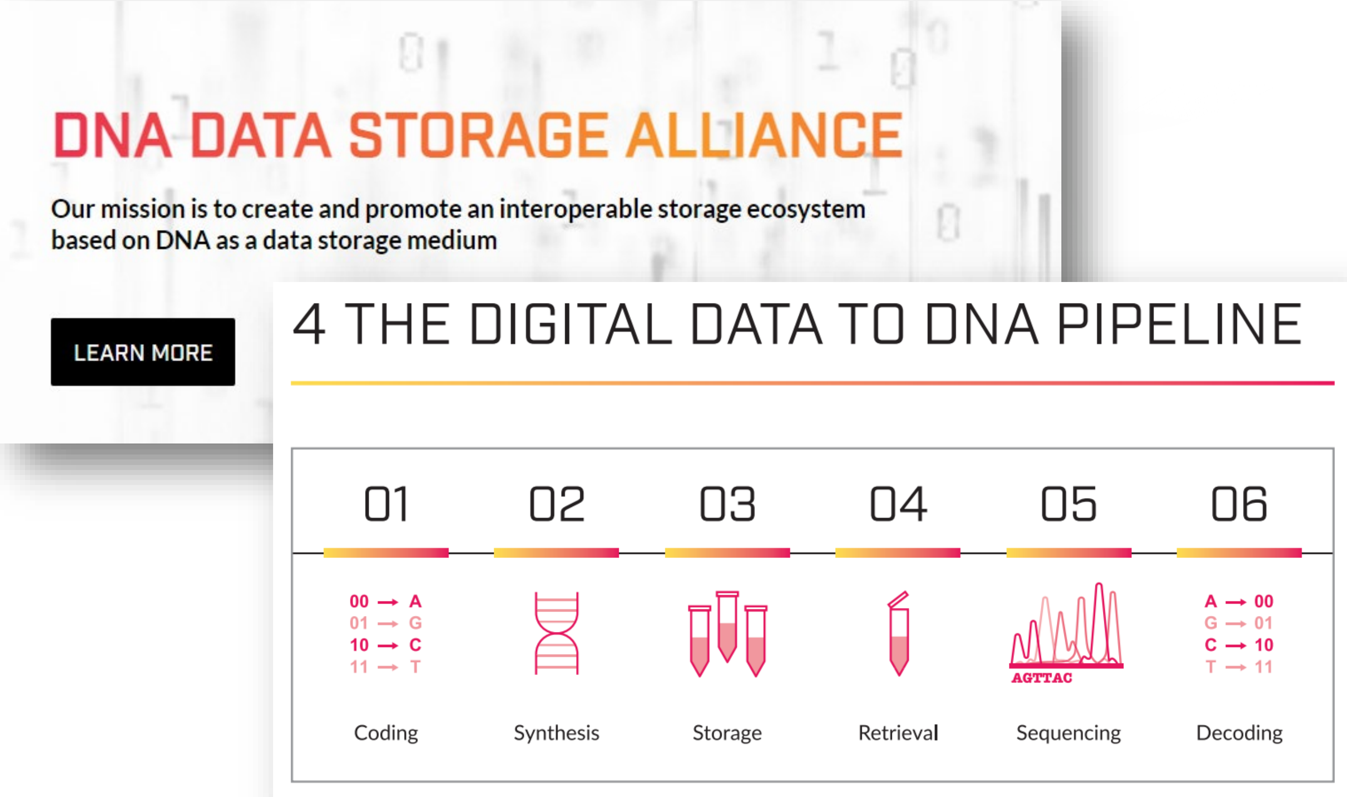
(Image credit: Pixabay)

As the quantity of data generated worldwide continues to expand at an aggressive rate, researchers are looking for ultra-dense and ultra-durable storage technologies capable of housing it all.

For example, Microsoft is examining the possibility of using lasers to etch data into quartz glass, or storing information in hologram form inside crystals. New developments in the field of tape storage, the current leading choice for archival use cases, are also promising.

However, one new storage medium in particular appears to have all the necessary attributes: deoxyribonucleic acid, or DNA. Researchers have found a single gram of DNA is capable of storing 215 PB (220,000 TB) of data.

- One gram of DNA can store 215 PB of data
- “Early adopters of DNA data storage are likely to be applications where they have
  - Write Once, Read Never (WORN) or
  - Write Once, Read Seldom if Ever (WORSE) data.”



DNA DATA STORAGE ALLIANCE

Our mission is to create and promote an interoperable storage ecosystem based on DNA as a data storage medium

LEARN MORE

4 THE DIGITAL DATA TO DNA PIPELINE

01 Coding

00 → A
01 → G
10 → C
11 → T

02 Synthesis

03 Storage

04 Retrieval

05 Sequencing

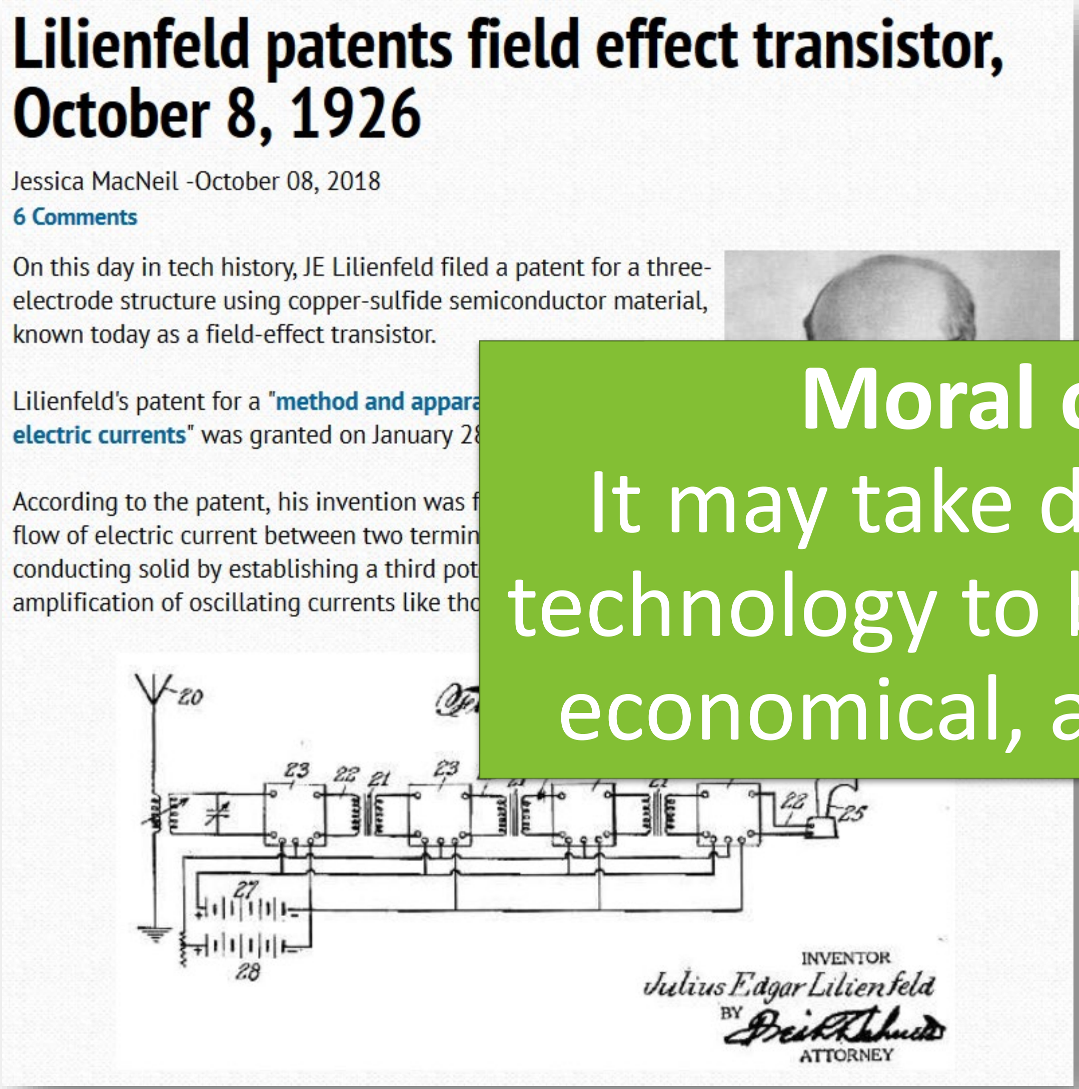
AGTTAC

06 Decoding

A → 00
G → 01
C → 10
T → 11

https://dnastoragealliance.org/dev/wp-content/uploads/2021/06/DNA-Data-Storage-Alliance-An-Introduction-to-DNA-Data-Storage.pdf

Fun Question about Future Technologies: when was the field effect transistor patented?



Lilienfeld patents field effect transistor, October 8, 1926

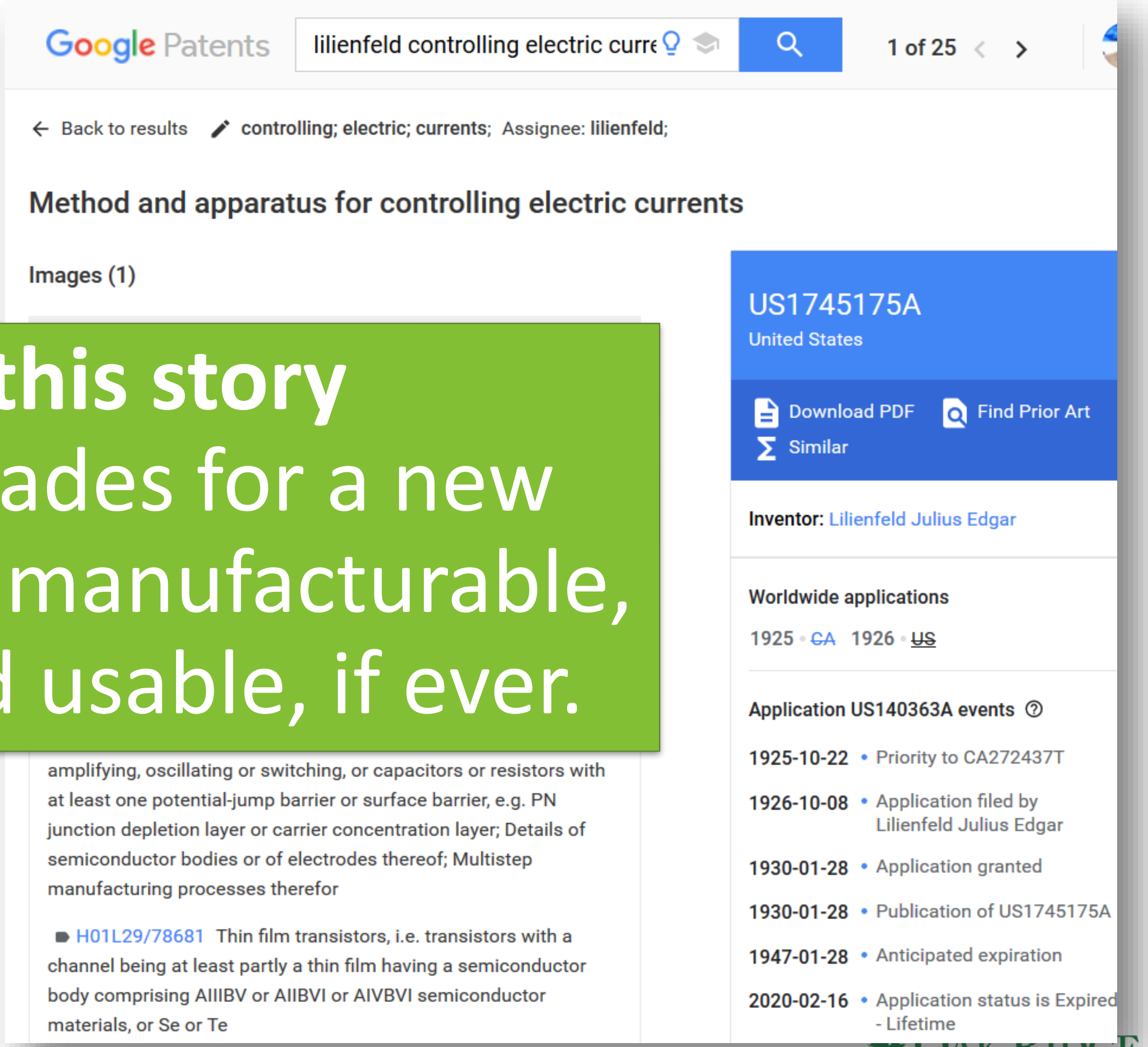
Jessica MacNeil -October 08, 2018

6 Comments

On this day in tech history, JE Lilienfeld filed a patent for a three-electrode structure using copper-sulfide semiconductor material, known today as a field-effect transistor.

Lilienfeld's patent for a "method and appara electric currents" was granted on January 28

According to the patent, his invention was f flow of electric current between two termin conducting solid by establishing a third pot amplification of oscillating currents like tho



Google Patents | lilienfeld controlling electric curre | 1 of 25

← Back to results | controlling; electric; currents; Assignee: lilienfeld;

Method and apparatus for controlling electric currents

Images (1)

amplifying, oscillating or switching, or capacitors or resistors with at least one potential-jump barrier or surface barrier, e.g. PN junction depletion layer or carrier concentration layer; Details of semiconductor bodies or of electrodes thereof; Multistep manufacturing processes therefor

H01L29/78681 Thin film transistors, i.e. transistors with a channel being at least partly a thin film having a semiconductor body comprising AIIIBV or AIIBVI or AIVBVI semiconductor materials, or Se or Te

US1745175A
United States

Download PDF | Find Prior Art | Similar

Inventor: Lilienfeld Julius Edgar

Worldwide applications

1925 • CA 1926 • US

Application US140363A events

1925-10-22	Priority to CA272437T
1926-10-08	Application filed by Lilienfeld Julius Edgar
1930-01-28	Application granted
1930-01-28	Publication of US1745175A
1947-01-28	Anticipated expiration
2020-02-16	Application status is Expired - Lifetime

Moral of this story
It may take decades for a new technology to be manufacturable, economical, and usable, if ever.

https://www.edn.com/electronics-blogs/edn-moments/4422371/Lilienfeld-patents-field-effect-transistor--October-8--1926

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

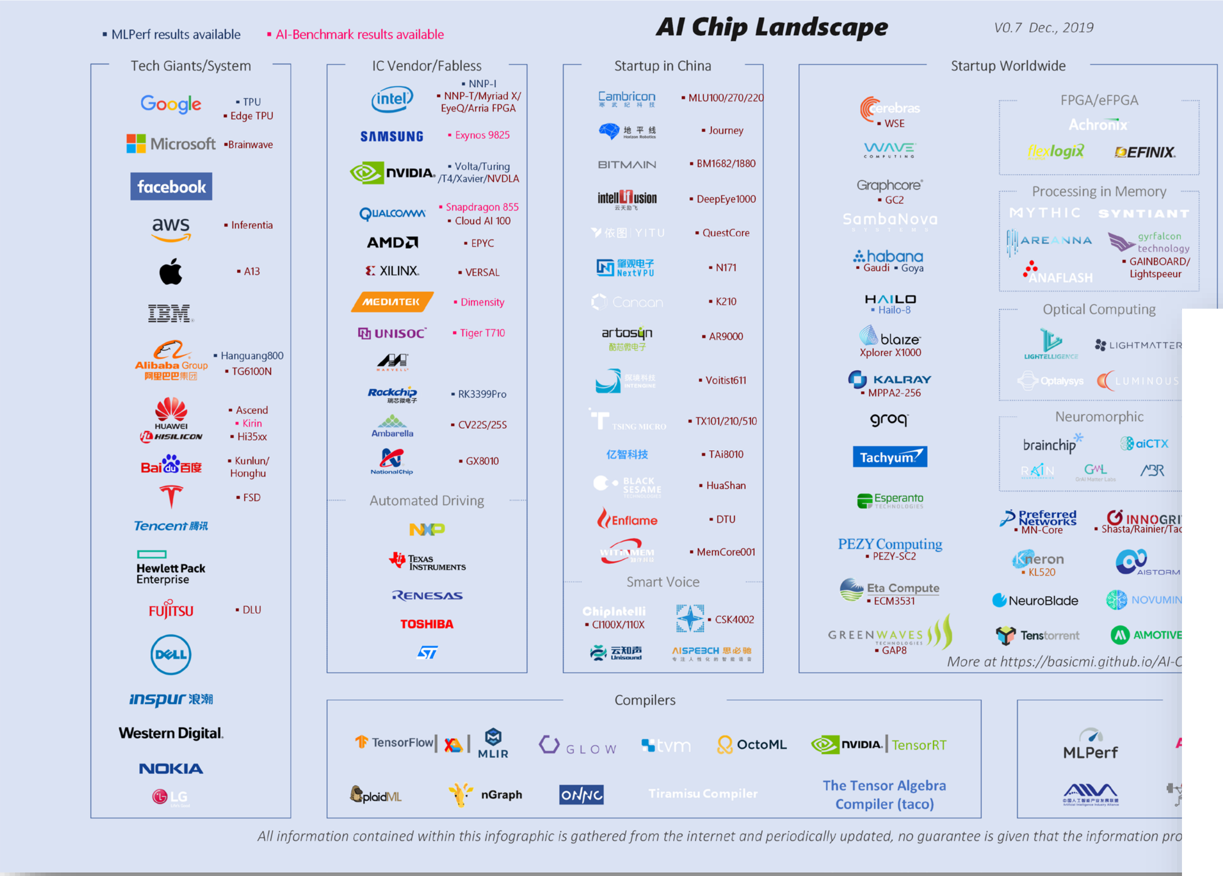
Architectural Specialization and Integration

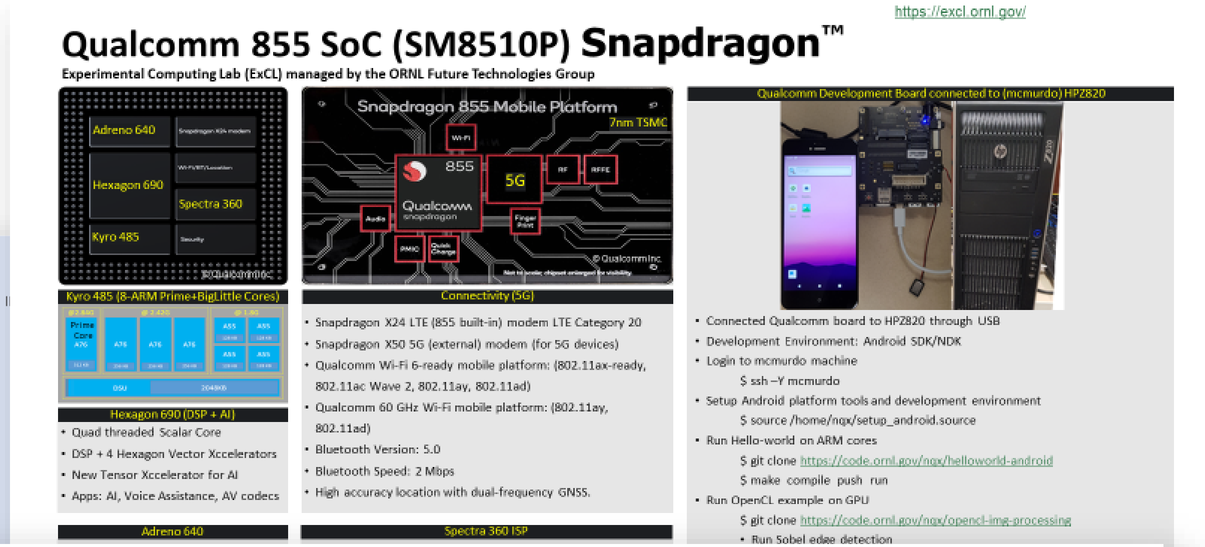
- Use CMOS (compatible) technology more effectively for specific workloads
- Integrate components to boost performance and eliminate inefficiencies
- Workload specific memory and storage system design

Emerging Technologies

- Investigate new computational paradigms
  - Quantum
  - Neuromorphic
  - Advanced Digital
  - Emerging Memory Devices

Various Markets are Already Specialized – HPC underway





https://excl.ornl.gov/

Qualcomm 855 SoC (SM8510P) Snapdragon™

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

Adreno 640
Hexagon 690
Spectra 360
Kyro 485
Security
Snapdragon 855 Mobile Platform
7nm TSMC
855
5G
Qualcomm snapdragon
Wi-Fi
RF
RFFE
Audio
PMIC
Quick Charge
Finger Print
© Qualcomm Inc.
Prime Core A76
A76
A55
DSU
Qualcomm Development Board connected to (mcmurdo) HPZ820

Kyro 485 (8-ARM Prime+BigLittle Cores)

Hexagon 690 (DSP + AI)

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

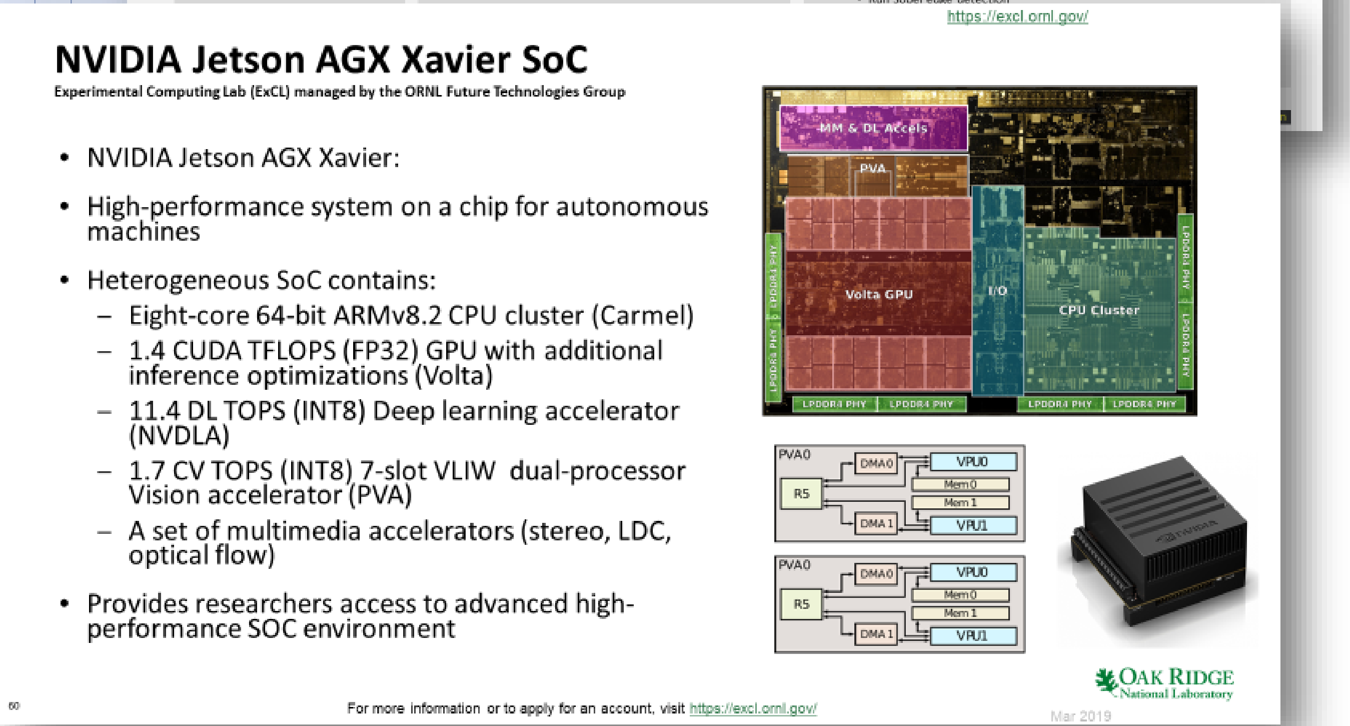
Connectivity (5G)

- 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.

- Connected Qualcomm board to HPZ820 through USB
- Development Environment: Android SDK/NDK
- Login to mcmurdo machine
  $ ssh –Y mcmurdo
- Setup Android platform tools and development environment
  $ source /home/ngx/setup_android.source
- Run Hello-world on ARM cores
  $ git clone https://code.ornl.gov/ngx/helloworld-android
  $ make compile push run
- Run OpenCL example on GPU
  $ git clone https://code.ornl.gov/ngx/opencl-img-processing
  - Run Sobel edge detection

Adreno 640

Spectra 360 ISP



https://excl.ornl.gov/

NVIDIA Jetson AGX Xavier SoC

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/

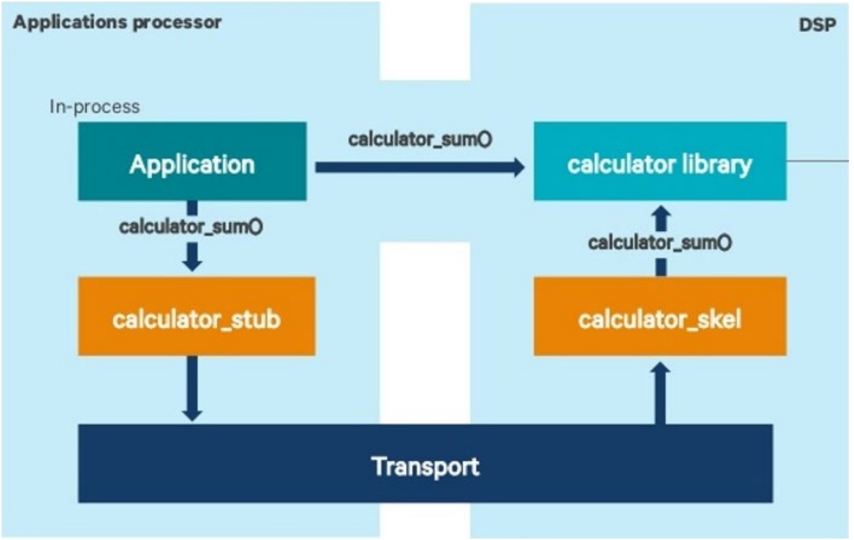
Mar 2019

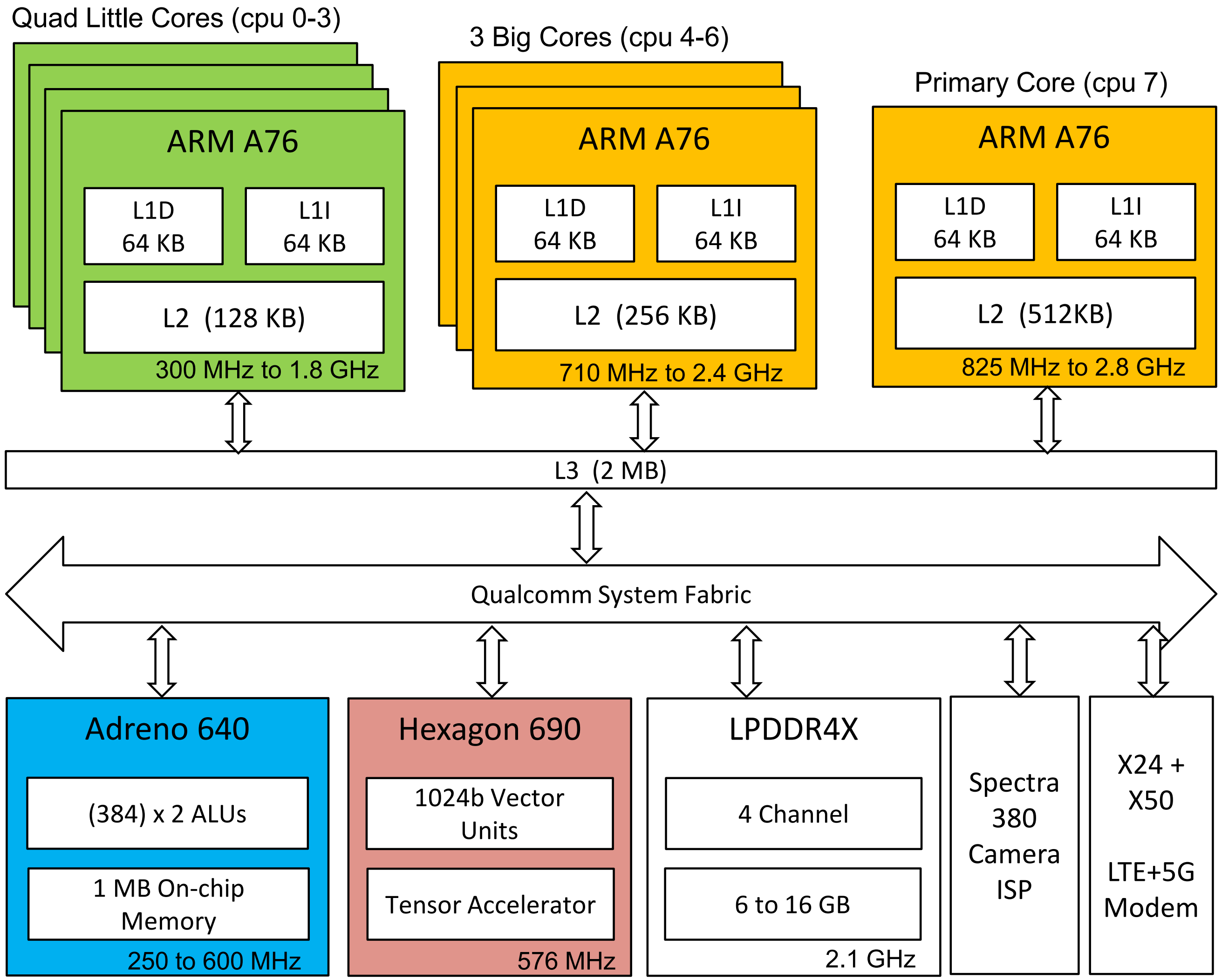
Example: Qualcomm Snapdragon Architecture and Programming

Snapdragon Programming Models

- ARM Kryo CPU: C/C++ with OpenMP
- Adreno GPU: OpenCL, OpenGL, Vulkan, and DirectX
- Hexagon DSP: C/C++ and Assembly language
- Spectra ISP, MDSP, NPU: custom Qualcomm API

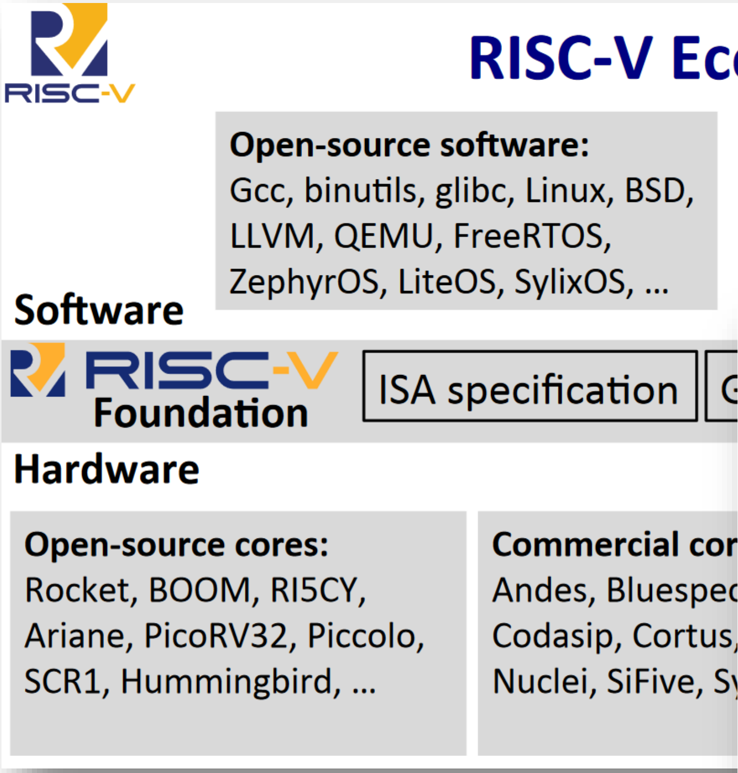
Hexagon DSP SDK





Qualcomm Snapdragon Architecture

Future -> Open Source Hardware Enables a Rapid Design of Specialized Chips and Effectively Mass Customization



RISC-V Ecosystem

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

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

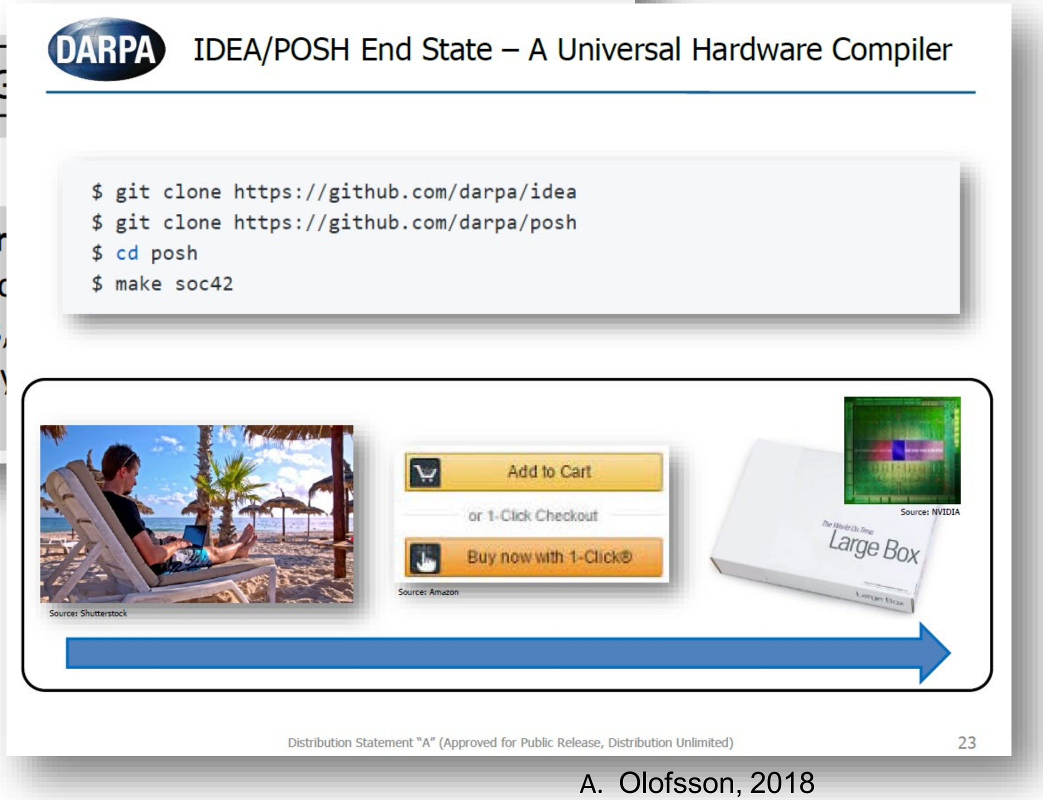
Software

RISC-V Foundation | ISA specification | G

Hardware

Open-source cores:
Rocket, BOOM, RI5CY, Ariane, PicoRV32, Piccolo, SCR1, Hummingbird, …

Commercial cor
Andes, Bluespe
Codasip, Cortus,
Nuclei, SiFive, Sy



DARPA IDEA/POSH End State – A Universal Hardware Compiler

```
$ git clone https://github.com/darpa/idea
$ git clone https://github.com/darpa/posh
$ cd posh
$ make soc42
```

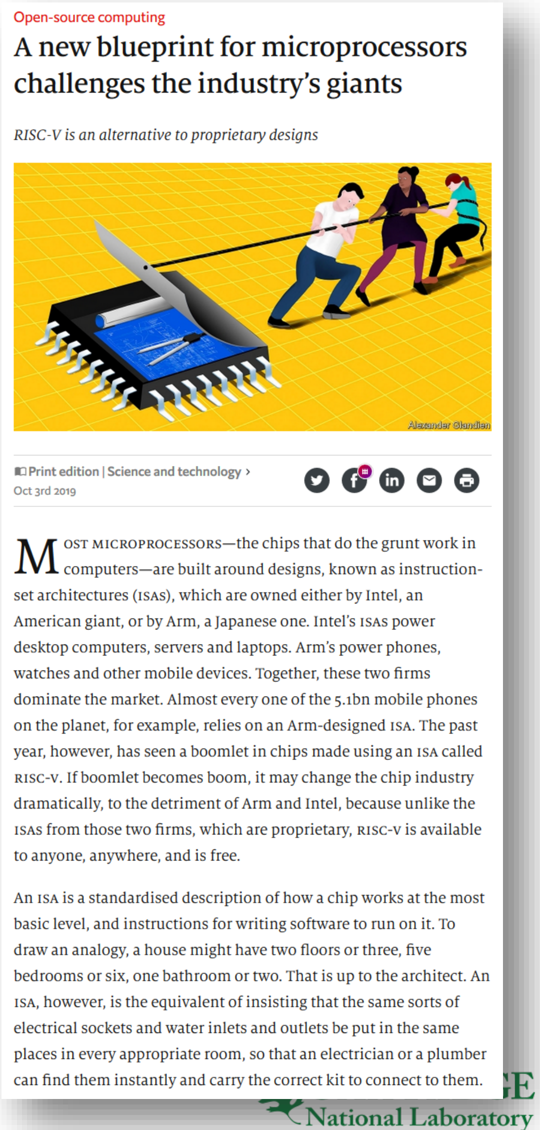
Source: Shutterstock

Source: Amazon

Source: NVIDIA

Distribution Statement "A" (Approved for Public Release, Distribution Unlimited) 23

A. Olofsson, 2018



Open-source computing

A new blueprint for microprocessors challenges the industry's giants

RISC-V is an alternative to proprietary designs

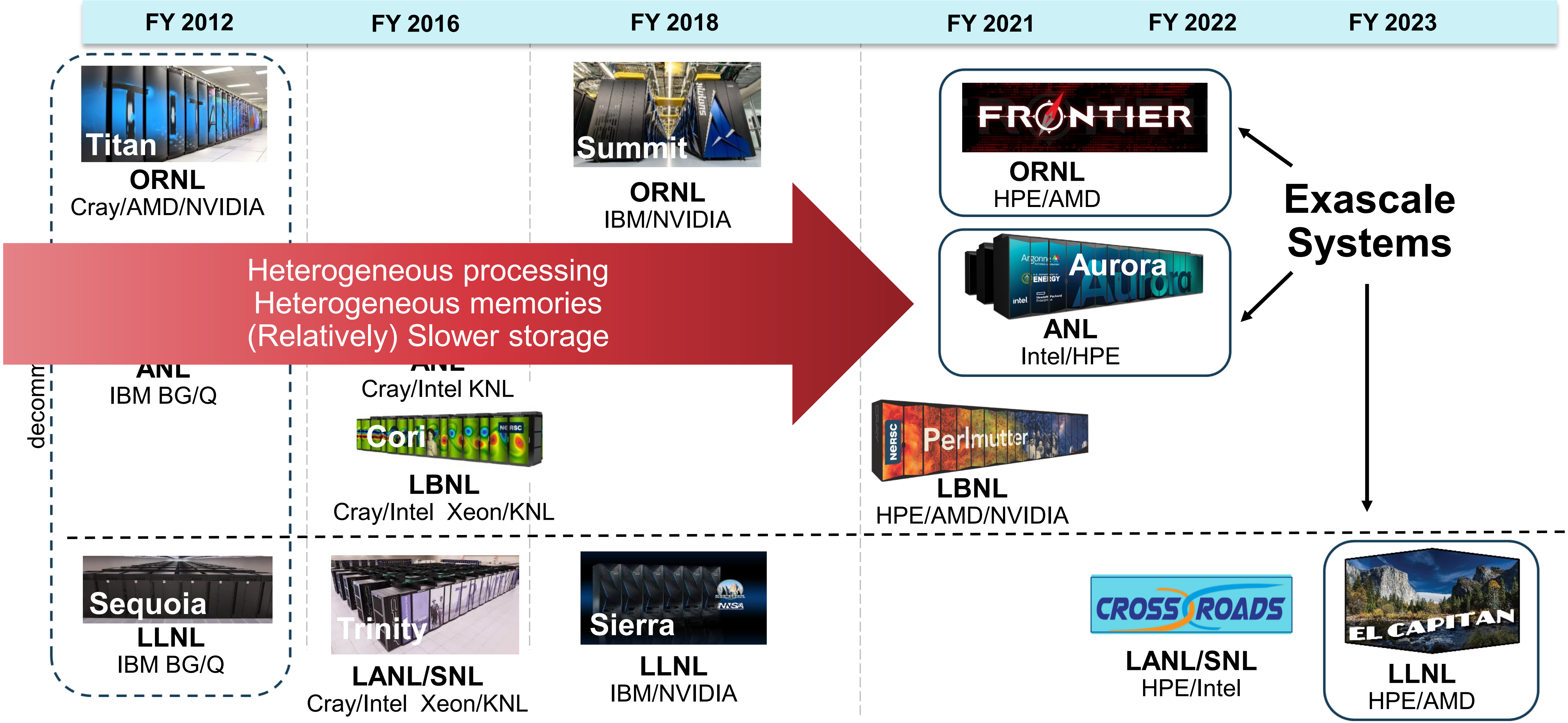
Alexander Glandien

Print edition | Science and technology ›
Oct 3rd 2019

MOST MICROPROCESSORS—the chips that do the grunt work in computers—are built around designs, known as instruction-set architectures (ISAs), which are owned either by Intel, an American giant, or by Arm, a Japanese one. Intel's ISAs power desktop computers, servers and laptops. Arm's power phones, watches and other mobile devices. Together, these two firms dominate the market. Almost every one of the 5.1bn mobile phones on the planet, for example, relies on an Arm-designed ISA. The past year, however, has seen a boomlet in chips made using an ISA called RISC-V. If boomlet becomes boom, it may change the chip industry dramatically, to the detriment of Arm and Intel, because unlike the ISAs from those two firms, which are proprietary, RISC-V is available to anyone, anywhere, and is free.

An ISA is a standardised description of how a chip works at the most basic level, and instructions for writing software to run on it. To draw an analogy, a house might have two floors or three, five bedrooms or six, one bathroom or two. That is up to the architect. An ISA, however, is the equivalent of insisting that the same sorts of electrical sockets and water inlets and outlets be put in the same places in every appropriate room, so that an electrician or a plumber can find them instantly and carry the correct kit to connect to them.

DOE HPC Roadmap to Exascale Systems



FY 2012 | FY 2016 | FY 2018 | FY 2021 | FY 2022 | FY 2023

Titan
ORNL
Cray/AMD/NVIDIA

ANL
IBM BG/Q

decomm

Sequoia
LLNL
IBM BG/Q

ANL
Cray/Intel KNL

Cori
LBNL
Cray/Intel Xeon/KNL

Trinity
LANL/SNL
Cray/Intel Xeon/KNL

Summit
ORNL
IBM/NVIDIA

Sierra
LLNL
IBM/NVIDIA

Heterogeneous processing
Heterogeneous memories
(Relatively) Slower storage

FRONTIER
ORNL
HPE/AMD

Aurora
ANL
Intel/HPE

Perlmutter
LBNL
HPE/AMD/NVIDIA

CROSSROADS
LANL/SNL
HPE/Intel

EL CAPITAN
LLNL
HPE/AMD

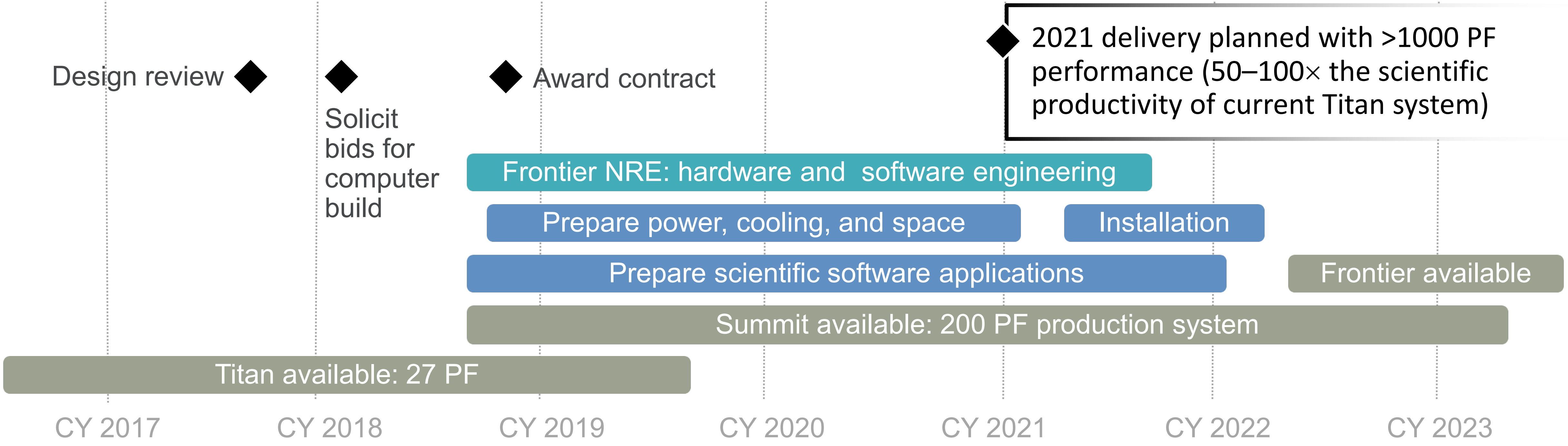
Exascale Systems

Frontier: The OLCF 2021 Exascale computer

Advanced simulations

High-performance data analytics

Artificial intelligence


Design review

Solicit bids for computer build

Award contract

2021 delivery planned with >1000 PF performance (50–100× the scientific productivity of current Titan system)

Frontier NRE: hardware and software engineering

Prepare power, cooling, and space

Installation

Prepare scientific software applications

Frontier available

Summit available: 200 PF production system

Titan available: 27 PF

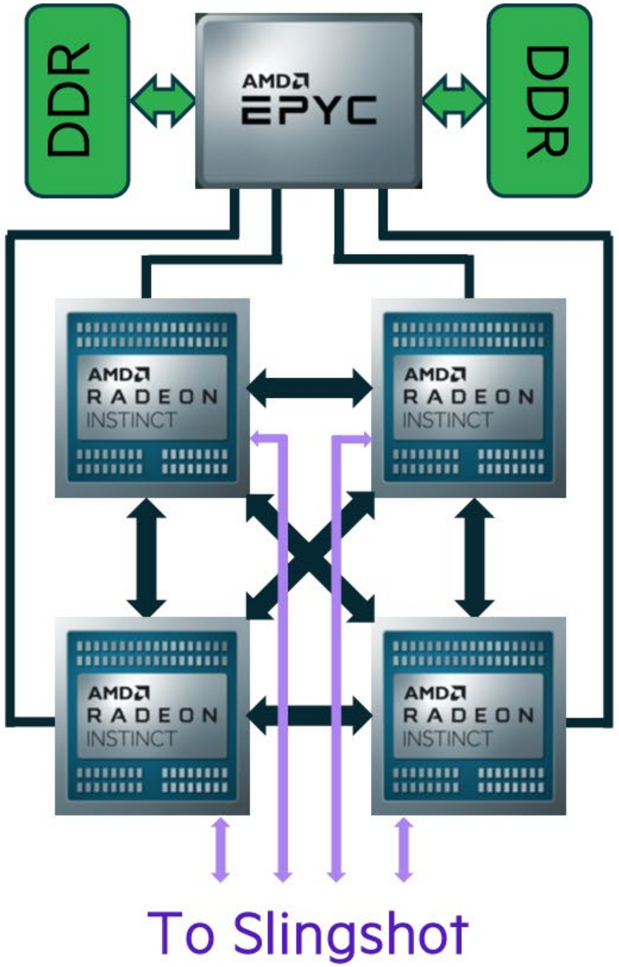
http://procurement.ornl.gov/rfp/CORAL2/

OAK RIDGE National Laboratory

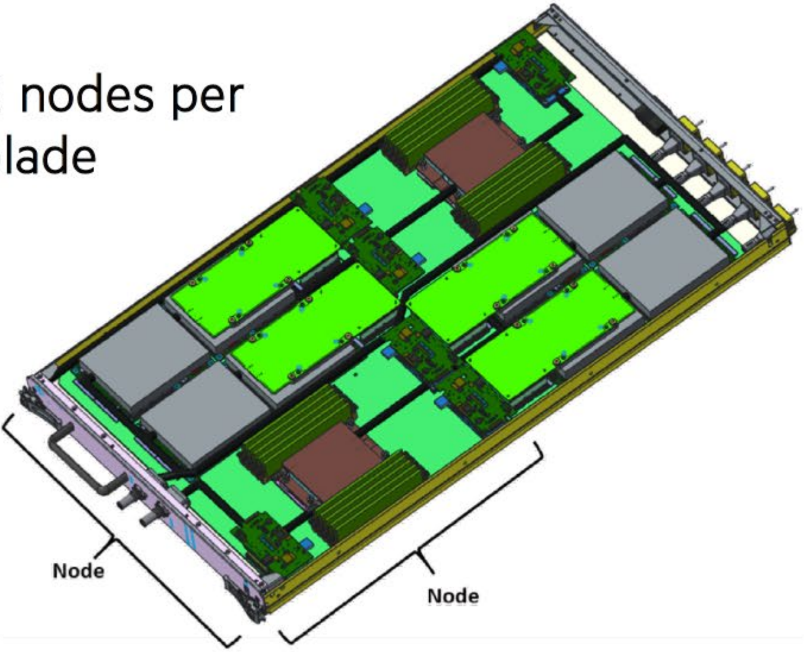
Frontier Node Architecture

AMD GPU

(ORNL)



2 nodes per blade



COPYRIGHT 2020 HPE

https://www.olcf.ornl.gov/wp-content/uploads/2019/05/frontier_specsheet.pdf

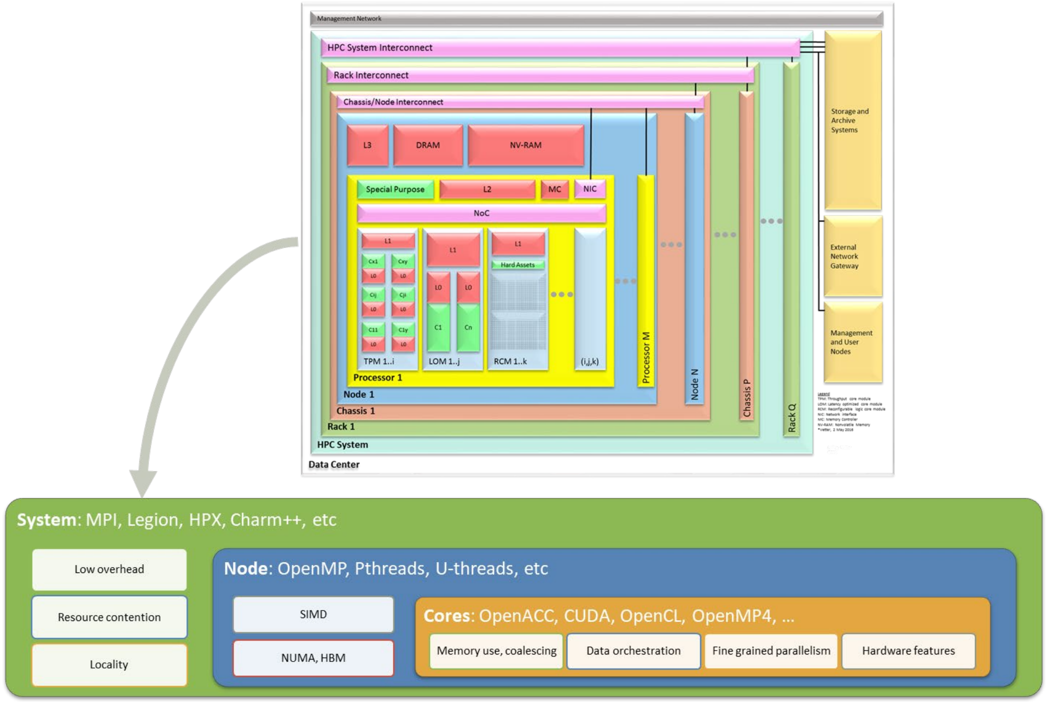
Take away message →
During this Sixth Wave transition, Complexity is our major challenge!

Architecture

- How do we design future systems so that they are better than current systems on important applications?
- Simulation and modeling are more difficult
- Entirely possible that the new system will be slower than the old system!
- Expect 'disaster' procurements

Programmabili
ty

- How do we design applications with some level of performance portability?
- Software lasts much longer than transient hardware platforms
- Proper abstractions for flexibility and efficiency
- Adapt or die



System: MPI, Legion, HPX, Charm++, etc

Low overhead

Resource contention

Locality

Node: OpenMP, Pthreads, U-threads, etc

SIMD

NUMA, HBM

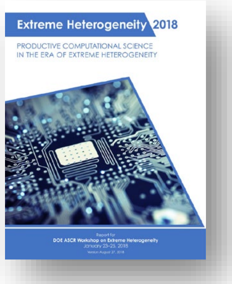
Cores: OpenACC, CUDA, OpenCL, OpenMP4, …

Memory use, coalescing

Data orchestration

Fine grained parallelism

Hardware features



Extreme Heterogeneity 2018
PRODUCTIVE COMPUTATIONAL SCIENCE IN THE ERA OF EXTREME HETEROGENEITY

Final Report on Workshop on Extreme Heterogeneity : https://doi.org/10.2172/1473756

Semiconductor Research Corp Decadal Plan: https://www.src.org/about/decadal-plan/

NITRD Software in the Era of Extreme Heterogeneity (Sep 2020) https://www.nitrd.gov/nitrdgroups/index.php?title=Software-Extreme-Heterogeneity

Oak Ridge National Laboratory

Software Strategies for Extreme Heterogeneity

Strategies for Programming Systems in this Era of Rapidly Designed, Diverse Architectures

Goals

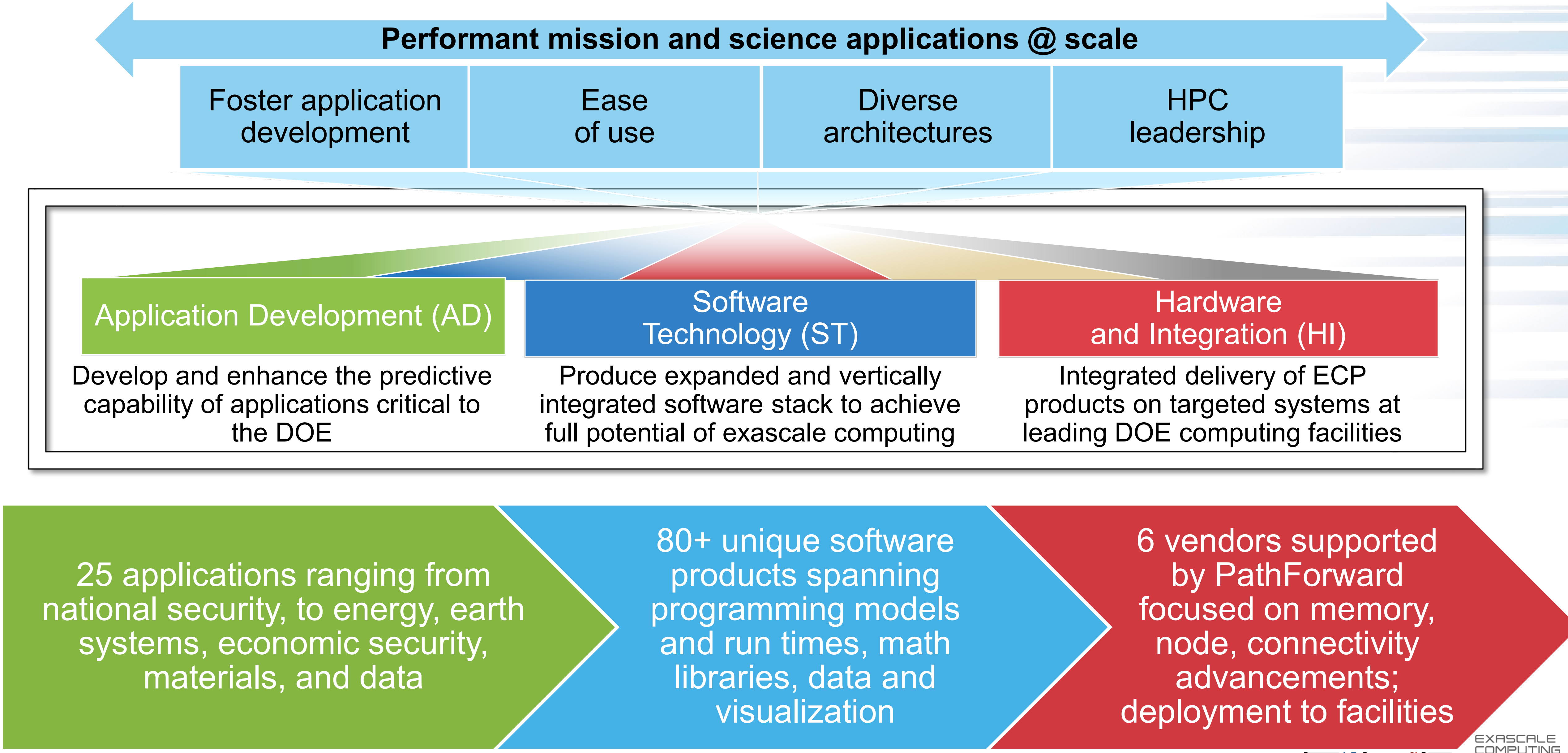
- Strive for 'Write one, run anywhere'
- Descriptive models of parallelism and data movement that enable effective code generation
- Introspective runtime systems
- Layered, modular, open source approaches required
  - One organization can't do it all

Examples

- Contributing to LLVM
  - FORTRAN with GPU offloading
- Programming FPGAs
  - Without Verilog
- Memory systems are changing too
  - Language support for NVM

The Open Source LLVM Compiler Ecosystem for Heterogeneous Computing

The three technical areas in ECP have the necessary components to meet national goals



Performant mission and science applications @ scale

Foster application development

Ease of use

Diverse architectures

HPC leadership

Application Development (AD)

Develop and enhance the predictive capability of applications critical to the DOE

Software Technology (ST)

Produce expanded and vertically integrated software stack to achieve full potential of exascale computing

Hardware and Integration (HI)

Integrated delivery of ECP products on targeted systems at leading DOE computing facilities

25 applications ranging from national security, to energy, earth systems, economic security, materials, and data

80+ unique software products spanning programming models and run times, math libraries, data and visualization

6 vendors supported by PathForward focused on memory, node, connectivity advancements; deployment to facilities

https://www.exascaleproject.org/

ECP is Improving the LLVM Compiler Ecosystem

LLVM COMPILER INFRASTRUCTURE

LLVM

- Very popular open-source compiler infrastructure
- Permissive license
- Modular, well-defined IR allows use by a lot of different languages, ML frameworks, etc.
- Backend infrastructure allowing the efficient creation of backends for new (heterogeneous) hardware.
- A state-of-the-art C++ frontend, CUDA support, scalable LTO, sanitizers and other debugging capabilities, and more.

+ SOLLVE

- Enhancing the implementation of OpenMP in LLVM
- Unified memory
- Prototype OMP features for LLVM
- OMP Optimizations
- OMP test suite
- Tracking OMP implementation quality
- Training

+ PROTEAS-TUNE

- Core optimization improvements to LLVM
  - OpenMP offload
- OpenACC capability for LLVM
  - Clacc
  - Flacc
- Autotuning for OpenACC and OpenMP in LLVM
- Integration with Tau performance tools
- SYCL characterizing and benchmarking
- Leading LLVM-DOE fork
- Training

+ FLANG

- Developing an open-source, production Fortran frontend
- Upstream to LLVM public release
- Support for OpenMP and OpenACC
- Recently approved by LLVM
- Initial implementation of serial F77 compiler for CPUs under review

+ HPCToolkit

- Improvements to OpenMP profiling interface OMPT
- OMPT specification improvements
- Refine HPCT for OMPT improvements

+ NNSA

- Enhancing LLVM to optimize template expansion for FlexCSI, Kokkos, RAJA, etc.
- Flang testing and evaluation
- Kitsune and Tapir

Vendors

- Increasing dependence on LLVM
- Many vendors import and redistribute LLVM
- Contributions and collaborations with many vendors through LLVM
  - AMD
  - ARM
  - Cray
  - HPE
  - IBM
  - Intel
  - NVIDIA

Active involvement with broad LLVM community: LLVM Dev, EuroLLVM
ECP personnel had 10+ presentations at the 2020 Dev Meeting

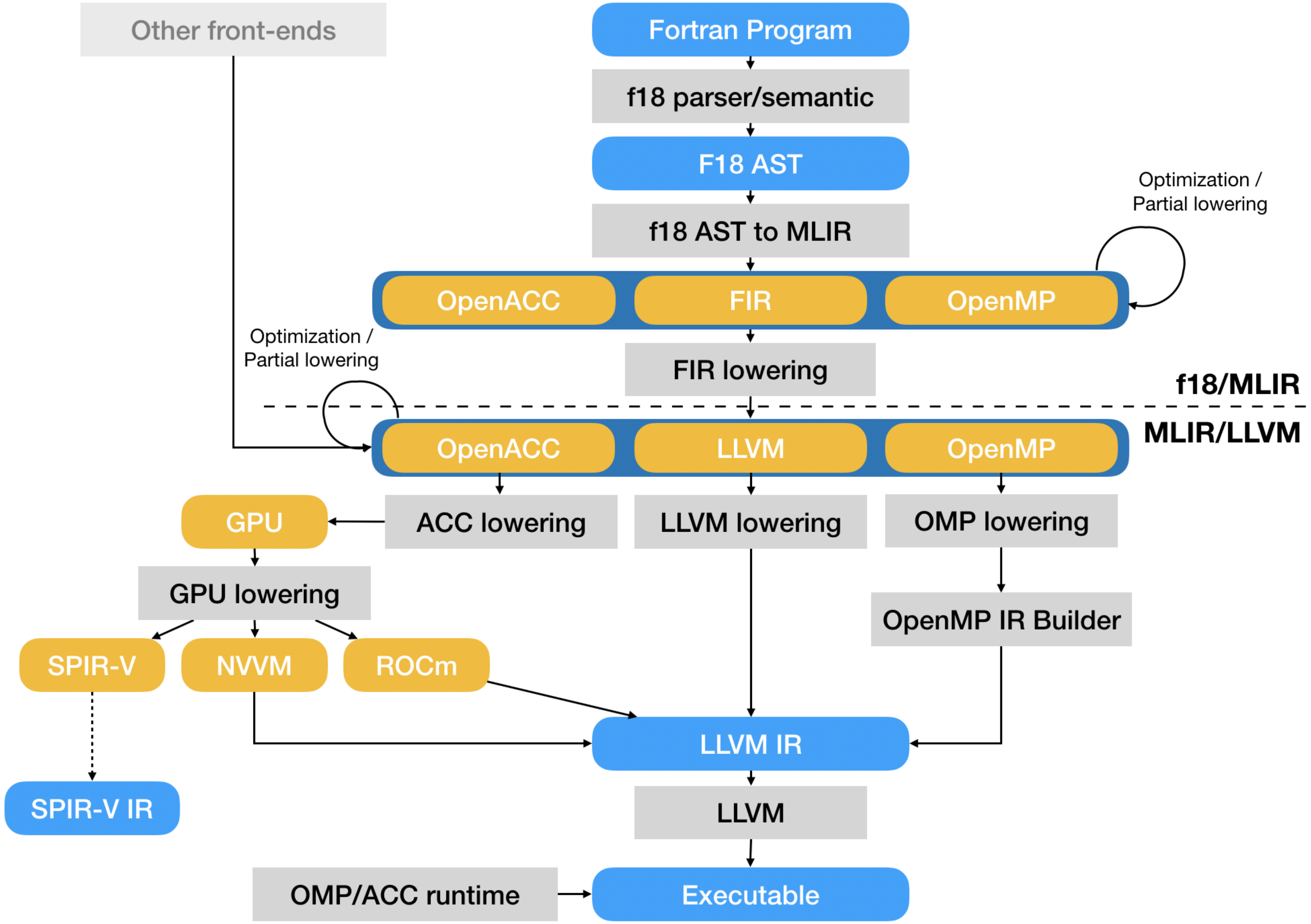
2020 LLVM VIRTUAL
DEVELOPERS' MEETING
OCTOBER 6-8

ECP EXASCALE COMPUTING PROJECT

ECP Projects: Flang, SOLLVE, PROTEAS-TUNE
Many other contributors: NNSA, NVIDIA, ARM, Google, …

Leveraging LLVM Ecosystem to Meet a Critical ECP (community) need : FORTRAN

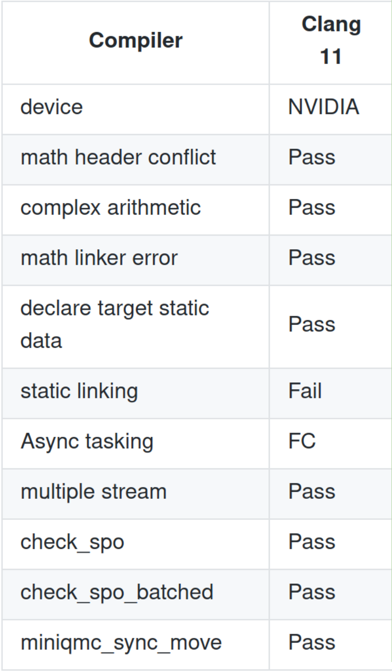
- Fortran support continues to be an ongoing requirement
- Flang project started in NNSA funding NVIDIA/PGI to open source compiler front-end into LLVM ecosystem
- SOLLVE is improving OpenMP dialect, implementation, and core optimizations
- PROTEAS-TUNE is creating OpenACC dialect and improving MLIR
- ECP projects are contributing many changes upstream to LLVM core, MLIR, etc
- Many others are contributing: backends for processors, optimizations in toolchain, …
    - Google contributed MLIR



V. Clement and J.S. Vetter, "Flacc: Towards OpenACC support for Fortran in the LLVM Ecosystem," in *Seventh Workshop on the LLVM Compiler Infrastructure in HPC SC21: St Louis, 2021*

Performance of OpenMP Offload in QMCPACK

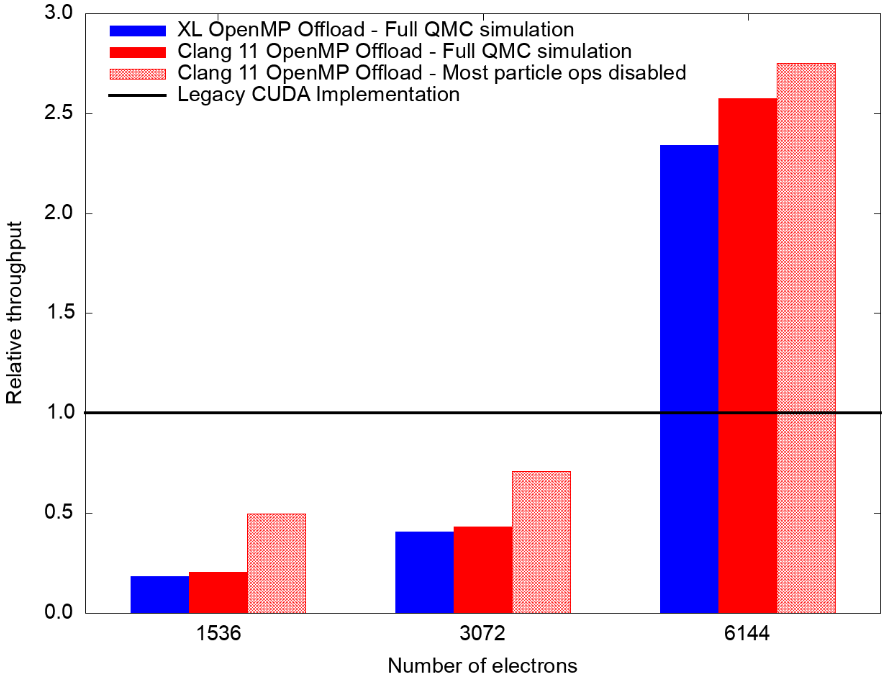
https://github.com/QMCPACK/miniqmc/wiki/OpenMP-offload#passfail-dashboard



Compiler	Clang 11
device	NVIDIA
math header conflict	Pass
complex arithmetic	Pass
math linker error	Pass
declare target static data	Pass
static linking	Fail
Async tasking	FC
multiple stream	Pass
check_spo	Pass
check_spo_batched	Pass
miniqmc_sync_move	Pass

Most compilers meet functionality needs.

Performance with IBM XL and LLVM clang OpenMP from Aug 2020 from QMCPACK Highlight



LLVM/Clang outperforms IBM XL on summit and is widely available.

→ QMCPACK tracks performance of latest OpenMP implementations available on ECP systems

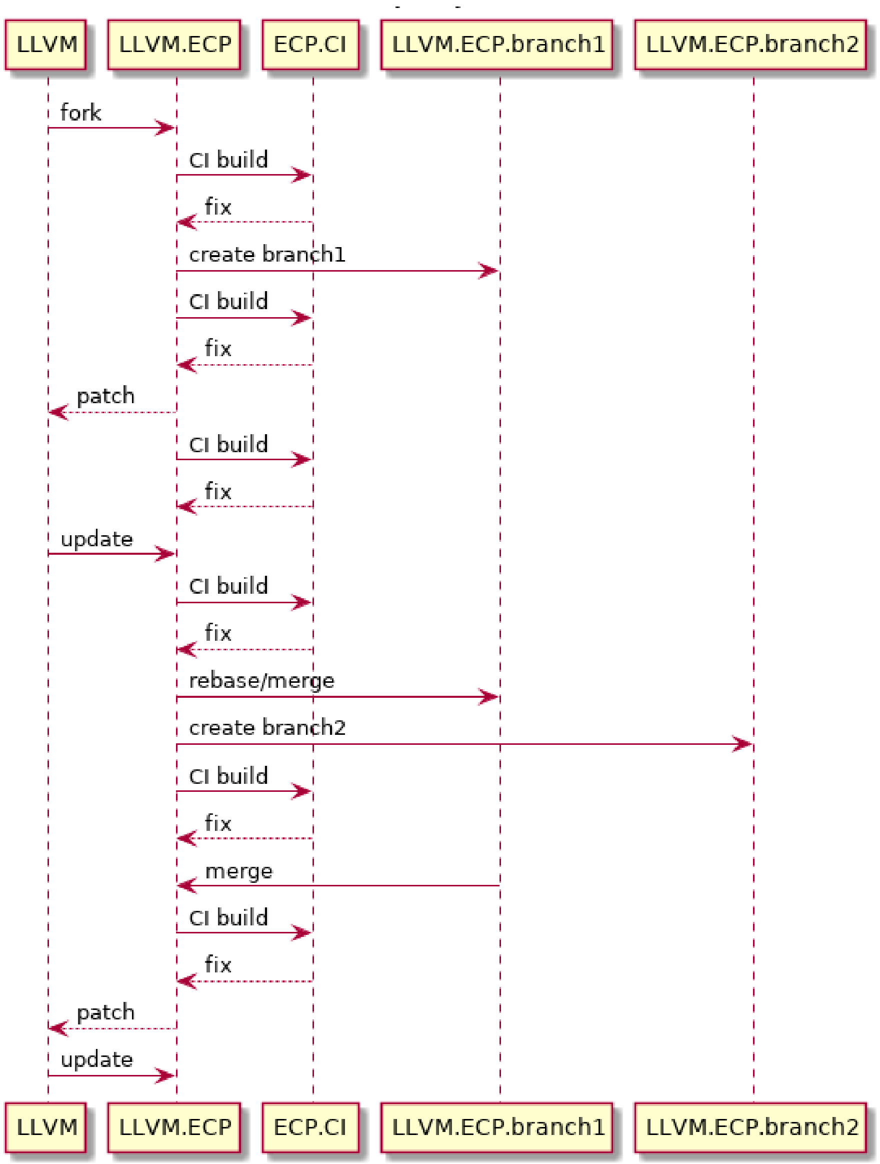
→ Rapid development of LLVM OpenMP has shown significant promise; better performance than IBM XL

Courtesy Paul Kent (ORNL) and Ye Luo (ANL) from QMCPACK team.

ECP LLVM Integration and Deployment

https://github.com/llvm-doe-org/llvm-project

- Projects contribute directly to LLVM monorepo
- Develop an integrated a DOE LLVM distribution
  - Integrating different ECP projects using LLVM
  - CI on target architectures
  - Shared vehicle for improvements in LLVM
  - Increased collaboration within ECP and DOE
- Operations
  - DOE LLVM distro will be closely maintained fork of LLVM mono repo
  - Multiple branches of individual ECP projects will exist at git branches
  - Branches will be integrated into DOE LLVM
- Periodic upstreaming and patching of major functionality to LLVM monorepo

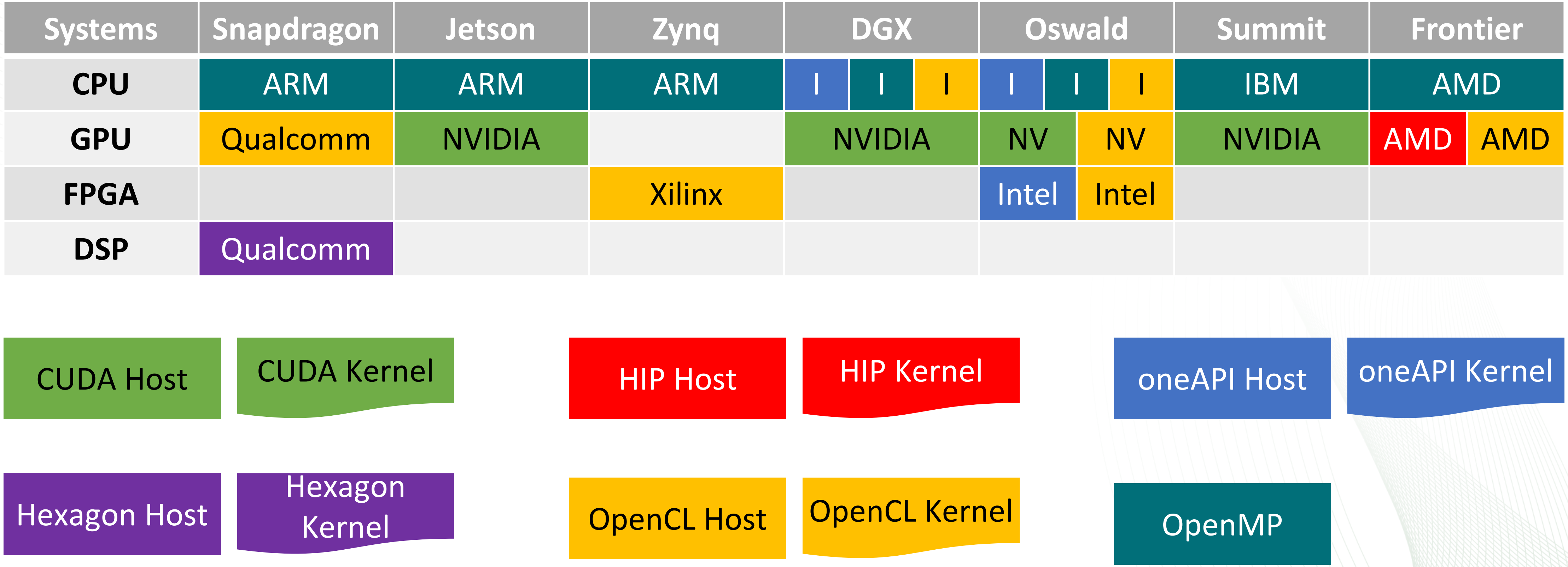


EXASCALE COMPUTING PROJECT

Introspective Runtime Systems

No De Facto Standard for Heterogeneous Programming

- ORNL Experimental Computing Laboratory (ExCL) systems*



Systems	Snapdragon	Jetson	Zynq	DGX	Oswald	Summit	Frontier
CPU	ARM	ARM	ARM	I I I	I I I	IBM	AMD
GPU	Qualcomm	NVIDIA		NVIDIA	NV NV	NVIDIA	AMD AMD
FPGA			Xilinx		Intel Intel		
DSP	Qualcomm						

CUDA Host

CUDA Kernel

HIP Host

HIP Kernel

oneAPI Host

oneAPI Kernel

Hexagon Host

Hexagon Kernel

OpenCL Host

OpenCL Kernel

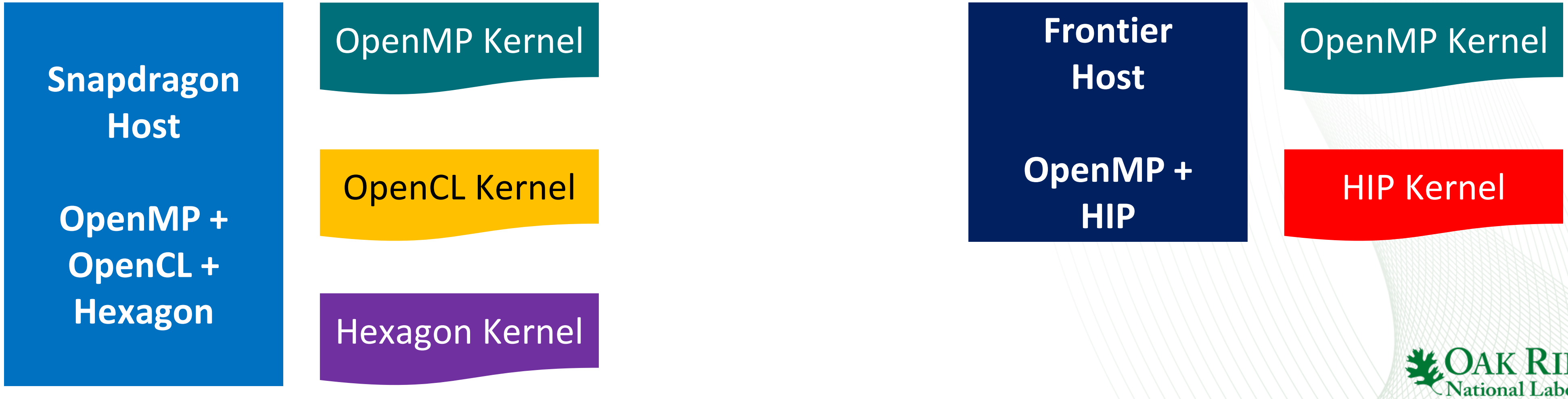
OpenMP

* ORNL ExCL: https://excl.ornl.gov/

We Need Portability in Heterogeneous Programming

- Not portable program across different HW configurations

<table>
<tr><th>Systems</th><th>Snapdragon</th><th>Jetson</th><th>Zynq</th><th colspan="3">DGX</th><th colspan="6">Oswald</th><th>Summit</th><th colspan="2">Frontier</th></tr>
<tr><td><b>CPU</b></td><td>ARM</td><td>ARM</td><td>ARM</td><td>I</td><td>I</td><td>I</td><td colspan="2">I</td><td colspan="2">I</td><td colspan="2">I</td><td>IBM</td><td colspan="2">AMD</td></tr>
<tr><td><b>GPU</b></td><td>Qualcomm</td><td>NVIDIA</td><td></td><td colspan="3">NVIDIA</td><td colspan="3">NV</td><td colspan="3">NV</td><td>NVIDIA</td><td>AMD</td><td>AMD</td></tr>
<tr><td><b>FPGA</b></td><td></td><td></td><td>Xilinx</td><td colspan="3"></td><td colspan="3">Intel</td><td colspan="3">Intel</td><td></td><td colspan="2"></td></tr>
<tr><td><b>DSP</b></td><td>Qualcomm</td><td></td><td></td><td colspan="3"></td><td colspan="6"></td><td></td><td colspan="2"></td></tr>
</table>



Snapdragon Host

OpenMP + OpenCL + Hexagon

OpenMP Kernel

OpenCL Kernel

Hexagon Kernel

Frontier Host

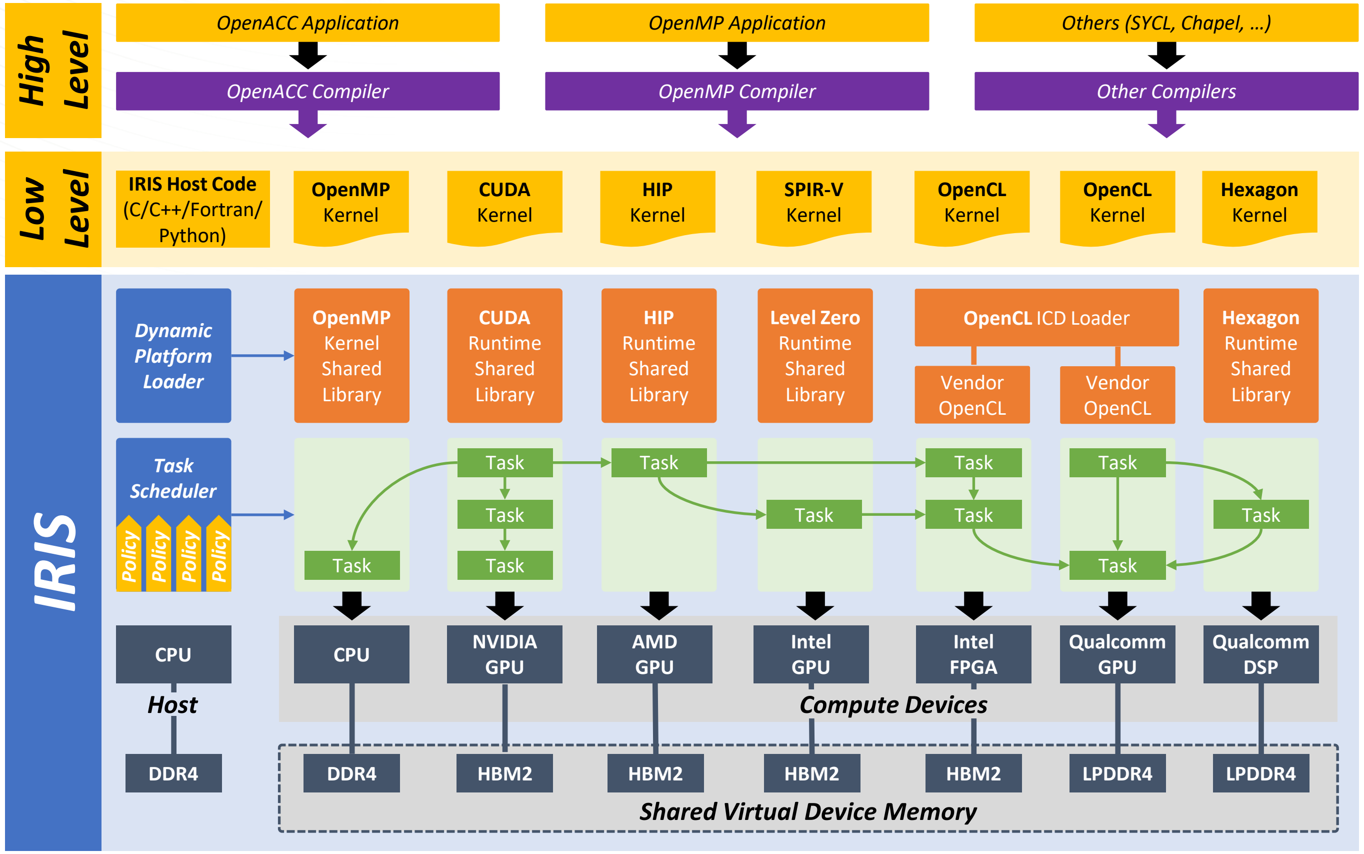
OpenMP + HIP

OpenMP Kernel

HIP Kernel

IRIS Runtime System: Orchestrating Multiple Architectures and Programming Systems

- The IRIS Architecture

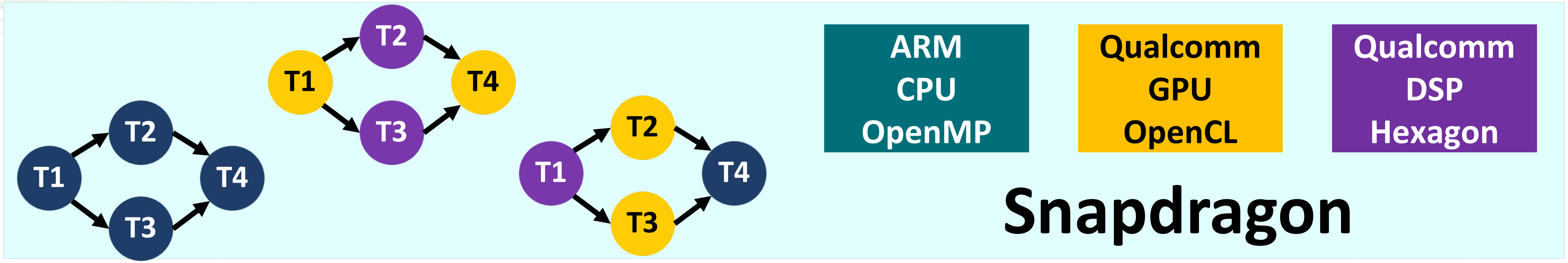


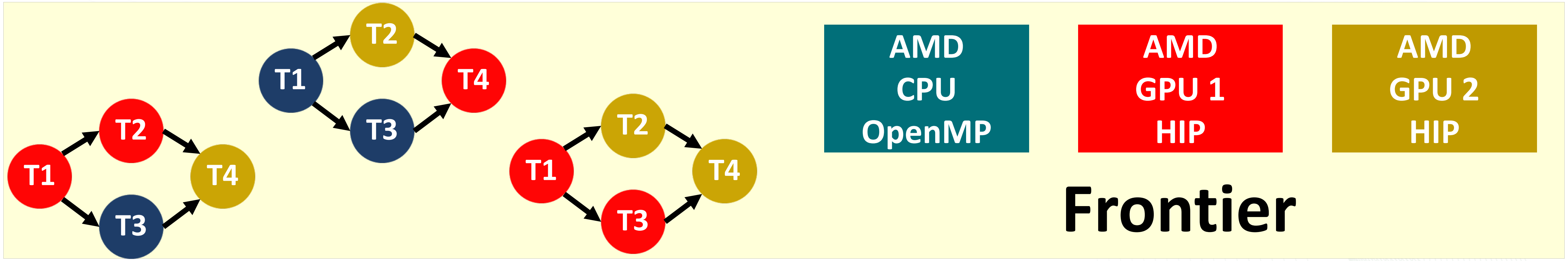
- Compilers
  - High level application → IRIS unified host code + native kernels
- Dynamic Platform Loader
  - Automatically discover all available accelerators and their programming systems
- Task Scheduler
  - Task: memory copy + kernel launch
  - DAG-style tasks graph across multiple devices
  - Device selection policies
- Shared Virtual Device Memory (SVDM)
  - An Illusion of single logical device memory across all physical device memories
  - Multiple local copies on multiple device memories (relaxed consistency model)

https://github.com/ORNL/iris

J. Kim, et al. “IRIS: A Portable Runtime System Exploiting Multiple Heterogeneous Programming Systems,” in *IEEE HPEC 2021, 2021*

Unified Host + Multiple Native Kernels + Shared VDM → *Flexible Task Scheduling & Portable Application*

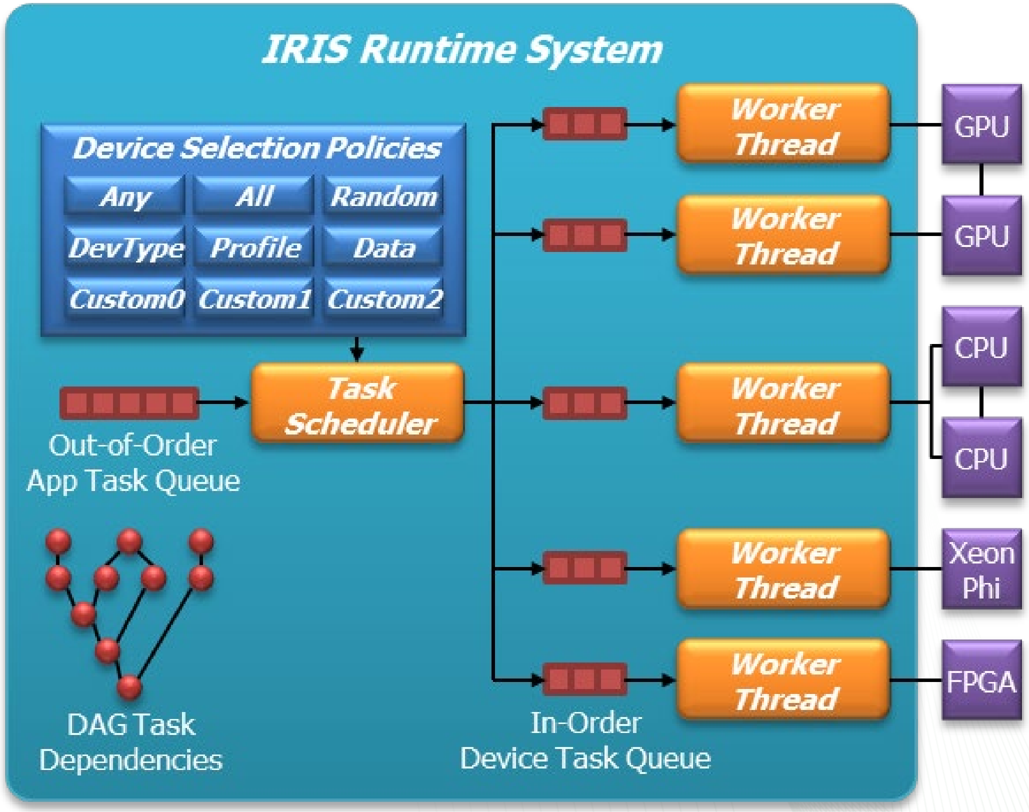




- A task can be freely scheduled and run on any device.
- An IRIS application is portable across different heterogeneous systems.

Task Scheduling in IRIS

- A task
  - A scheduling unit
  - Contains multiple in-order commands
    - Kernel launch command
    - Memory copy command (device-to-host, host-to-device)
  - May have DAG-style dependencies with other tasks
  - Enqueued to the application task queue with a device selection policy
    - Available device selection policies
      - Specific Device (compute device #)
      - Device Type (CPU, GPU, FPGA, XeonPhi)
      - Profile-based
      - Locality-aware
      - Ontology-base
      - Performance models (Aspen)
      - Any, All, Random, 3rd-party users' custom policies
- The task scheduler dispatches the tasks in the application task queue to available compute devices
  - Select the optimal target compute device according to task's device selection policy

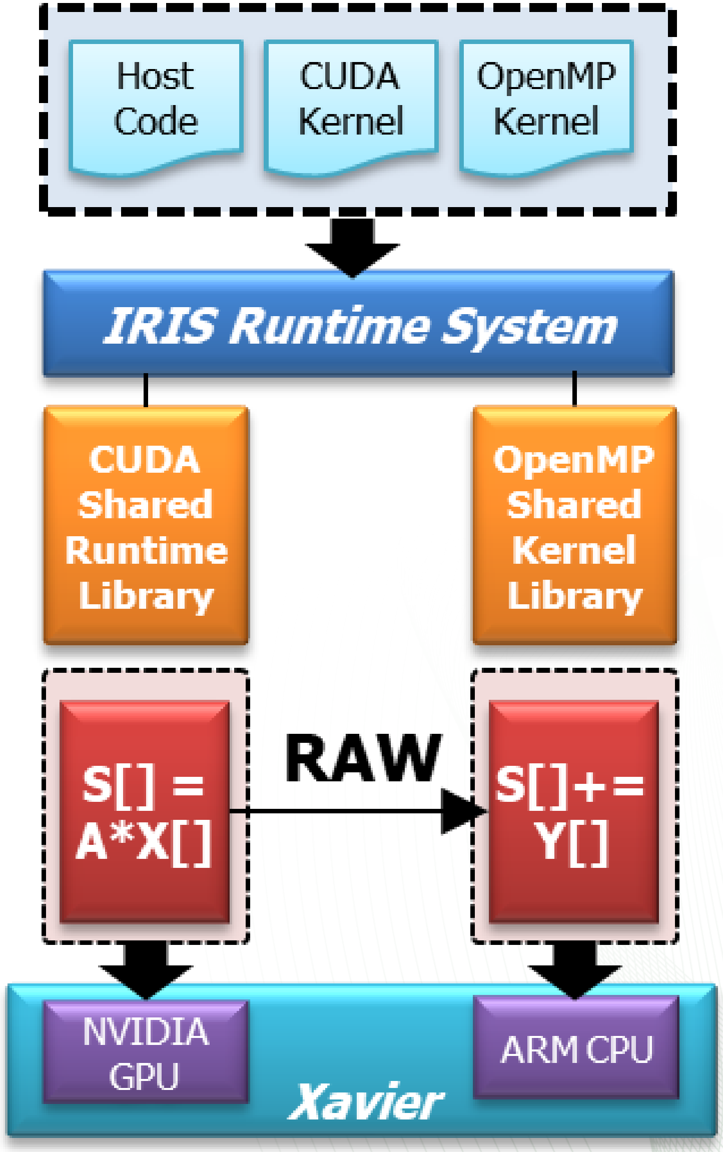


SAXPY Example on Xavier

- Computation
  - S[] = A * X[] + Y[]

- Two tasks
  - S[] = A * X[] on NVIDIA GPU (CUDA)
  - S[] += Y[] on ARM CPU (OpenMP)
    - S[] is shared between two tasks
    - Read-after-write (RAW), true dependency

- Low-level Python IRIS host code + CUDA/OpenMP kernels
  - saxpy.py
  - kernel.cu
  - kernel.openmp.h



SAXPY: Python host code & CUDA kernel code

saxpy.py (1/2)

```
#!/usr/bin/env python

import iris
import numpy as np
import sys

iris.init()

SIZE = 1024
A = 10.0

x = np.arange(SIZE, dtype=np.float32)
y = np.arange(SIZE, dtype=np.float32)
s = np.arange(SIZE, dtype=np.float32)

print 'X', x
print 'Y', y

mem_x = iris.mem(x.nbytes)
mem_y = iris.mem(y.nbytes)
mem_s = iris.mem(s.nbytes)
```

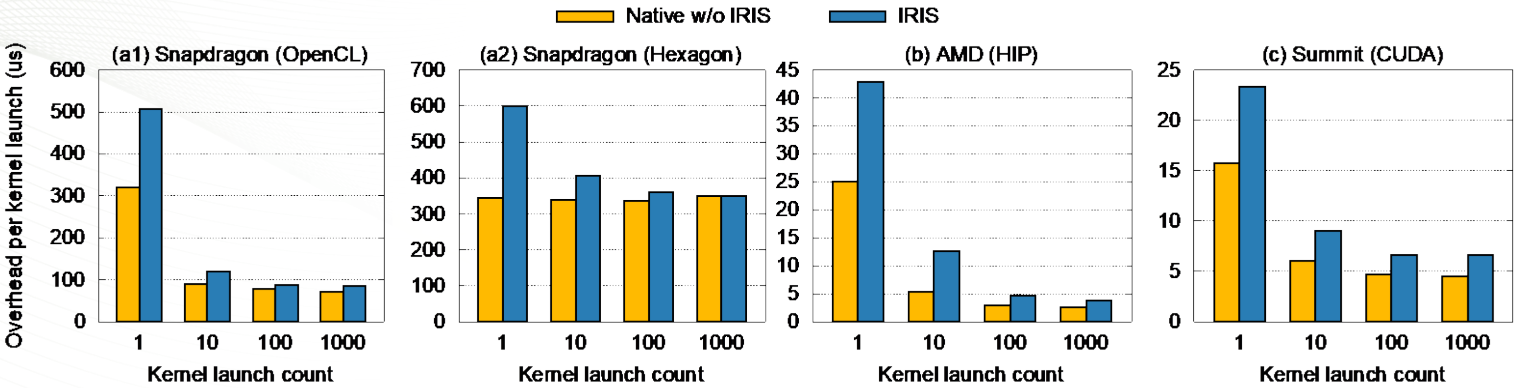
saxpy.py (2/2)

```
kernel0 = iris.kernel("saxpy0")
kernel0.setmem(0, mem_s, iris.iris_w)
kernel0.setint(1, A)
kernel0.setmem(2, mem_x, iris.iris_r)

off = [ 0 ]
ndr = [ SIZE ]

task0 = iris.task()
task0.h2d_full(mem_x, x)
task0.kernel(kernel0, 1, off, ndr)
task0.submit(iris.iris_gpu)

kernel1 = iris.kernel("saxpy1")
kernel1.setmem(0, mem_s, iris.iris_rw)
kernel1.setmem(1, mem_y, iris.iris_r)

task1 = iris.task()
task1.h2d_full(mem_y, y)
task1.kernel(kernel1, 1, off, ndr)
task1.d2h_full(mem_s, s)
task1.submit(iris.iris_cpu)

print 'S =', A, '* X + Y', s

iris.finalize()
```

kernel.cu (CUDA)

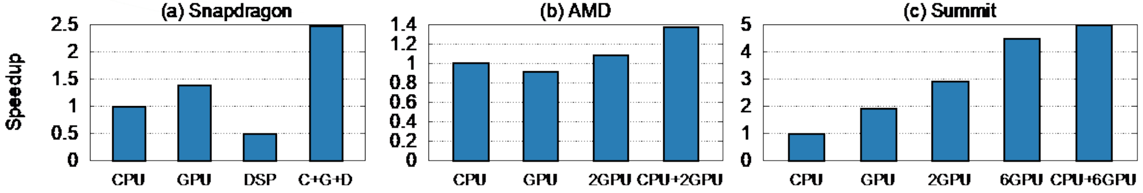
```
extern "C" __global__ void saxpy0(float* S, float A, float* X) {
  int id = blockIdx.x * blockDim.x + threadIdx.x;
  S[id] = A * X[id];
}

extern "C" __global__ void saxpy1(float* S, float* Y) {
  int id = blockIdx.x * blockDim.x + threadIdx.x;
  S[id] += Y[id];
}
```

Evaluation

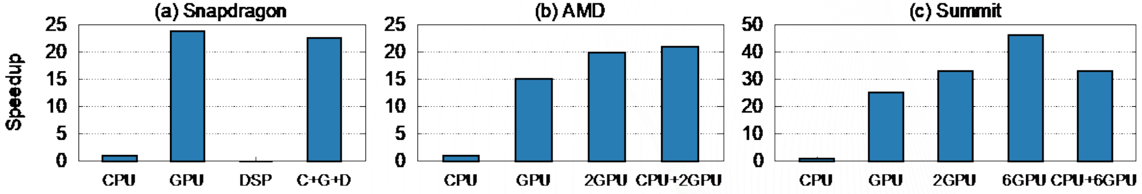


Kernel Launch Overhead

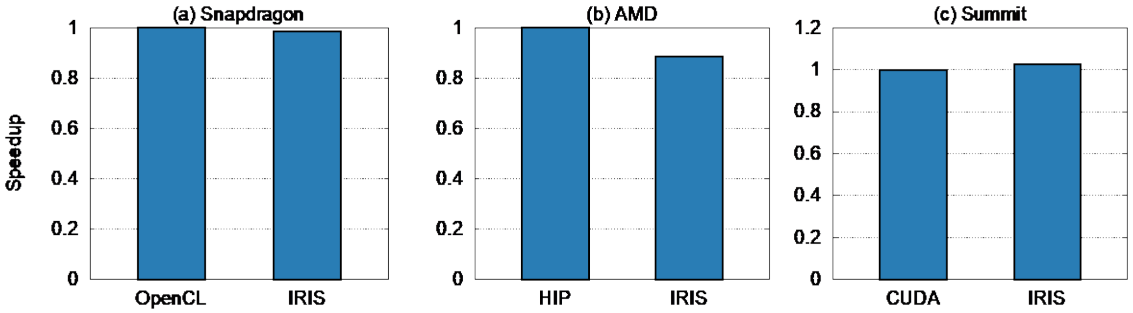
SAXPY



SGEMM



LULESH



Systems	Snapdragon	AMD	Summit
CPU	ARM OpenMP	AMD OpenMP	IBM OpenMP
GPU	Qualcomm OpenCL	AMD HIP	NVIDIA CUDA
DSP	Qualcomm Hexagon		

Reconfigurable Computing with FPGAs

FPGA Computing

Benefits

- Can provide excellent performance and power advantages
  - For specific workloads
  - For certain real-time and signal processing tasks, they are indispensable
- Can provide domains specific acceleration
  - Mass customization
  - 17b math ?

Challenges

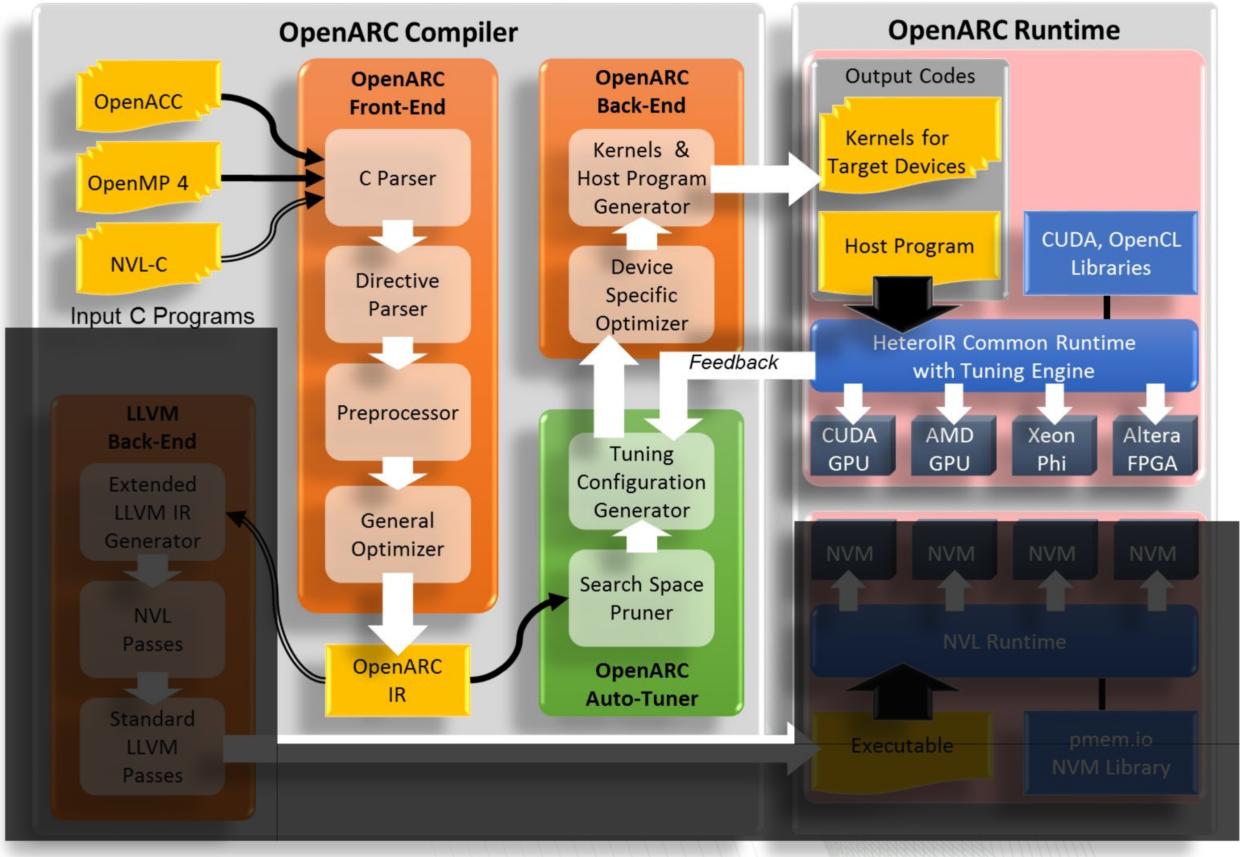
- Programmability and Portability Issues
  - Best performance for FPGAs requires writing Hardware Description Languages (HDLs) such as VHDL and Verilog; too complex and low-level
    - HDL requires substantial knowledge on hardware (digital circuits).
    - Programmers must think in terms of a state machine.
    - HDL programming is a kind of digital circuit design.
  - High-Level Synthesis (HLS) to provide better FPGA programmability
    - SRC platforms, Handel-C, Impulse C-to-FPGA compiler, Xilinx Vivado (AutoPilot), FCUDA, etc.
    - None of these use a portable, open standard.
- Reconfiguration time
- Cost (for high end FPGAs)

Standard, Portable Programming Models for Heterogeneous Computing

- OpenCL
  - Open standard portable across diverse heterogeneous platforms (e.g., CPUs, GPUs, DSPs, Xeon Phis, FPGAs, etc.)
  - Much higher than HDL, but still complex for typical programmers.
- Directive-based accelerator programming models
  - OpenACC, OpenMP4, etc.
  - Provide higher abstraction than OpenCL.
  - Most of existing OpenACC/OpenMP4 compilers target only specific architectures; none supports FPGAs.

Directive-based Strategy with OpenARC: Open Accelerator Research Compiler

- Open-Sourced, High-Level Intermediate Representation (HIR)-Based, Extensible Compiler Framework.
  - Perform source-to-source translation from OpenACC C to target accelerator models.
    - Support full features of OpenACC V1.0 (+ array reductions and function calls)
    - Support both CUDA and OpenCL as target accelerator models
  - Provide common runtime APIs for various back-ends
  - Can be used as a research framework for various study on directive-based accelerator computing.
    - Built on top of Cetus compiler framework, equipped with various advanced analysis/transformation passes and built-in tuning tools.
    - OpenARC's IR provides an AST-like syntactic view of the source program, easy to understand, access, and transform the input program.



S. Lee and J.S. Vetter, "OpenARC: Open Accelerator Research Compiler for Directive-Based, Efficient Heterogeneous Computing," in *ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC). Vancouver: ACM, 2014*

FPGAs|Approach

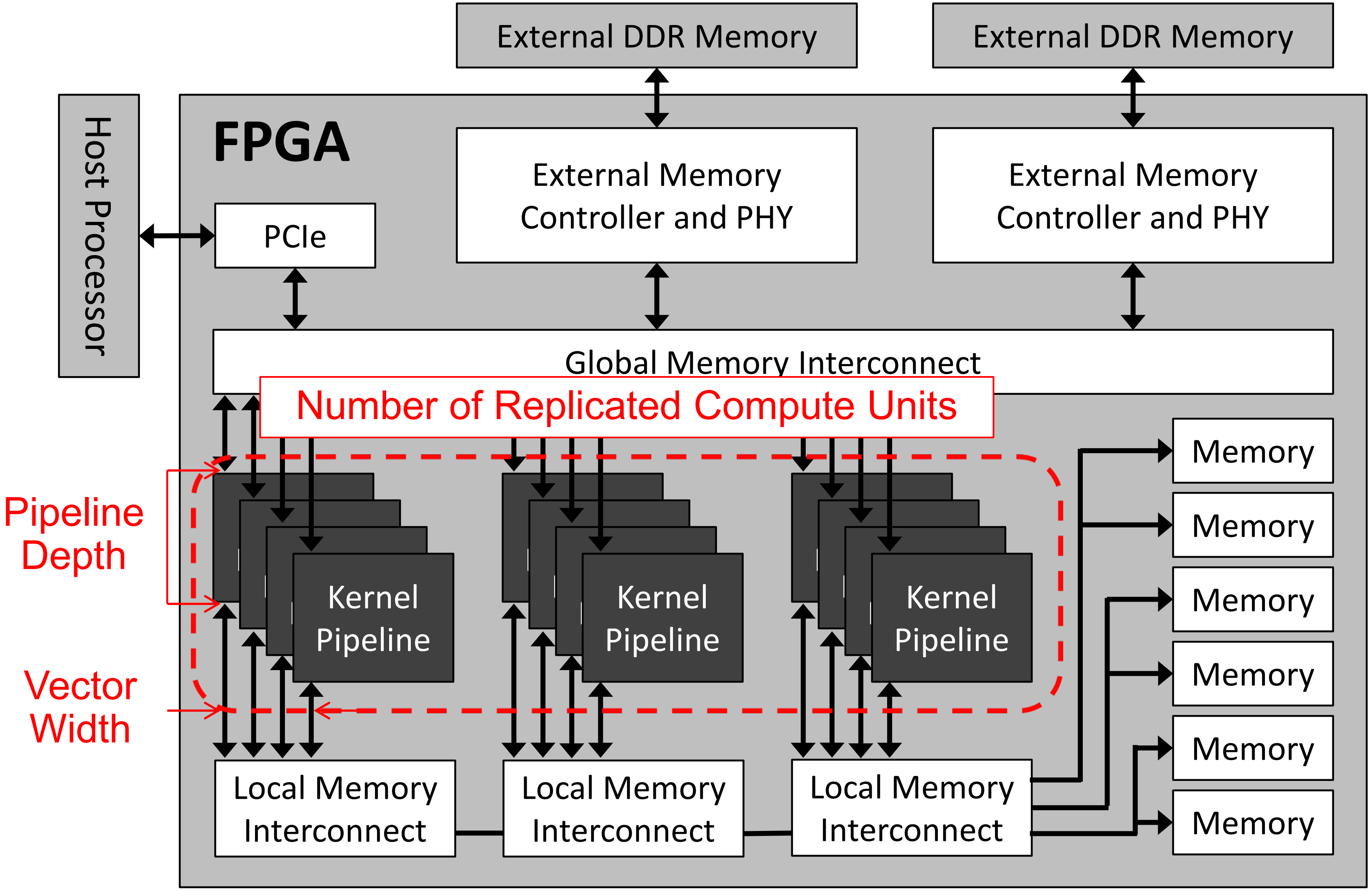
- Design and implement an OpenACC-to-FPGA translation framework, which is the first work to use a standard and portable directive-based, high-level programming system for FPGAs.
- Propose FPGA-specific optimizations and novel pragma extensions to improve performance.
- Evaluate the functional and performance portability of the framework across diverse architectures (Altera FPGA, NVIDIA GPU, AMD GPU, and Intel Xeon Phi).

S. Lee, J. Kim, and J.S. Vetter, "OpenACC to FPGA: A Framework for Directive-based High-Performance Reconfigurable Computing," Proc. IEEE International Parallel & Distributed Processing Symposium (IPDPS), 2016, 10.1109/IPDPS.2016.28.

Baseline Translation of OpenACC-to-FPGA

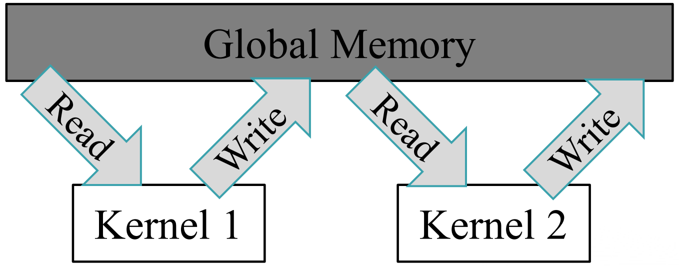
- Use OpenCL as the output model and the Altera Offline Compiler (AOC) as its backend compiler.
- Translates the input OpenACC program into a host code containing HeteroIR constructs and device-specific kernel codes.
  - Use the same HeteroIR runtime system of the existing OpenCL backends, except for the device initialization.
  - Reuse most of compiler passes for kernel generation.

FPGA OpenCL Architecture

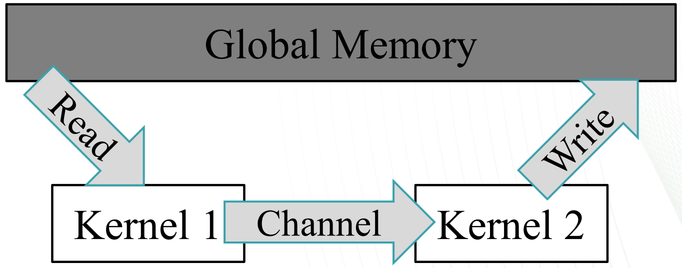

External DDR Memory
External DDR Memory
Host Processor
FPGA
PCIe
External Memory Controller and PHY
External Memory Controller and PHY
Global Memory Interconnect
Number of Replicated Compute Units
Pipeline Depth
Vector Width
Kernel Pipeline
Kernel Pipeline
Kernel Pipeline
Memory
Memory
Memory
Memory
Memory
Memory
Local Memory Interconnect
Local Memory Interconnect
Local Memory Interconnect

Kernel-Pipelining Transformation Optimization

- Kernel execution model in OpenACC
  - Device kernels can communicate with each other only through the device global memory.
  - Synchronizations between kernels are at the granularity of a kernel execution.
- Altera OpenCL channels
  - Allows passing data between kernels and synchronizing kernels with high efficiency and low latency



Kernel communications through global memory in OpenACC

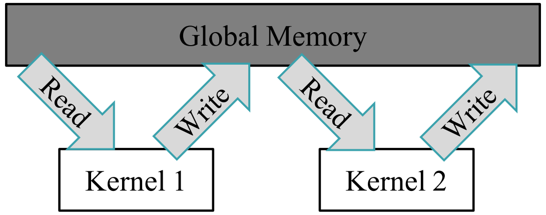


Kernel communications with Altera channels

Kernel-Pipelining Transformation Optimization (2)

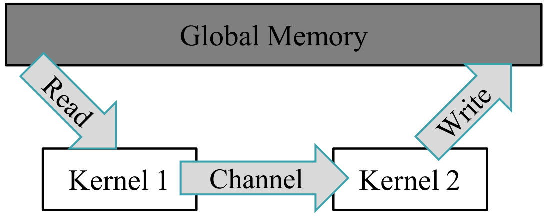
(a) Input OpenACC code

```
#pragma acc data copyin (a) create (b) copyout (c)
{
    #pragma acc kernels loop gang worker present (a, b)
    for(i=0; i<N; i++) { b[i] = a[i]*a[i]; }
    #pragma acc kernels loop gang worker present (b, c)
    for(i=0; i<N; i++) {c[i] = b[i]; }
}
```



(b) Altera OpenCL code with channels

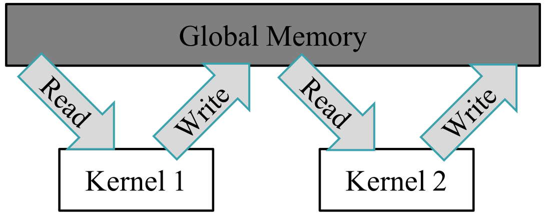
```
channel float pipe_b;
__kernel void kernel1(__global float* a) {
    int i = get_global_id(0);
    write_channel_altera(pipe_b, a[i]*a[i]);
}
__kernel void kernel2(__global float* c) {
    int i = get_global_id(0);
    c[i] = read_channel_altera(pipe_b);
}
```



Kernel-Pipelining Transformation Optimization (3)

(a) Input OpenACC code

```
#pragma acc data copyin (a) create (b) copyout (c)
{
    #pragma acc kernels loop gang worker present (a, b)
    for(i=0; i<N; i++) { b[i] = a[i]*a[i]; }
    #pragma acc kernels loop gang worker present (b, c)
    for(i=0; i<N; i++) {c[i] = b[i]; }
}
```

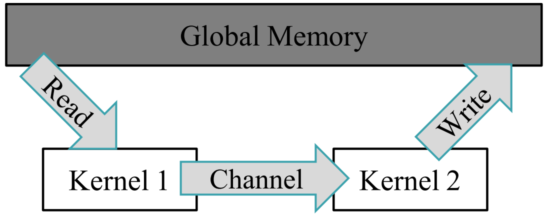


Kernel-pipelining transformation

Valid under specific conditions

(c) Modified OpenACC code for kernel-pipelining

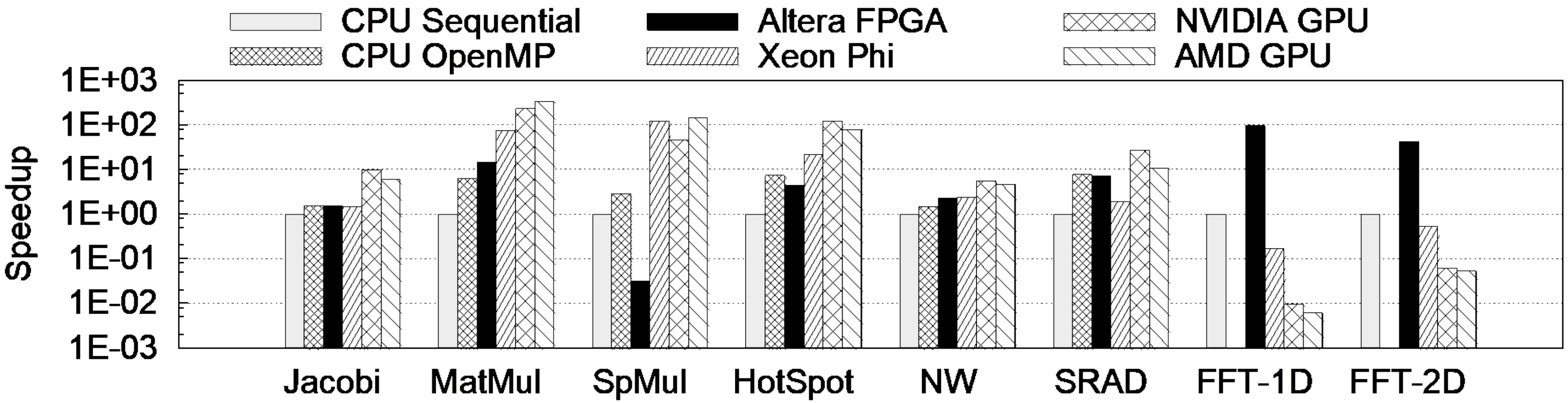
```
#pragma acc data copyin (a) pipe (b) copyout (c)
{
    #pragma acc kernels loop gang worker pipeout (b) present (a)
    For(i=0; i<N; i++) { b[i] = a[i]*a[i]; }
    #pragma acc kernels loop gang worker pipein (b) present (c)
    For(i=0; i<N; i++) {c[i] = b[i];}
}
```



FPGA-specific Optimizations

- Single work-item
- Collapse
- Reduction
- Sliding window
- (Branch-variant code motion)
- (Custom unrolling)

Overall Performance of OpenARC FPGA Evaluation



FPGAs prefer applications with deep execution pipelines (e.g., FFT-1D and FFT-2D), performing much higher than other accelerators.

For traditional HPC applications with abundant parallel floating-point operations, it seems to be difficult for FPGAs to beat the performance of other accelerators, even though FPGAs can be much more power-efficient.

- Tested FPGA does not contain dedicated, embedded floating-point cores, while others have fully-optimized floating-point computation units.

Current and upcoming high-end FPGAs are equipped with hardened floating-point operators, whose performance will be comparable to other accelerators, while remaining power-efficient.

Recent results from Arria 10 and Stratix 10

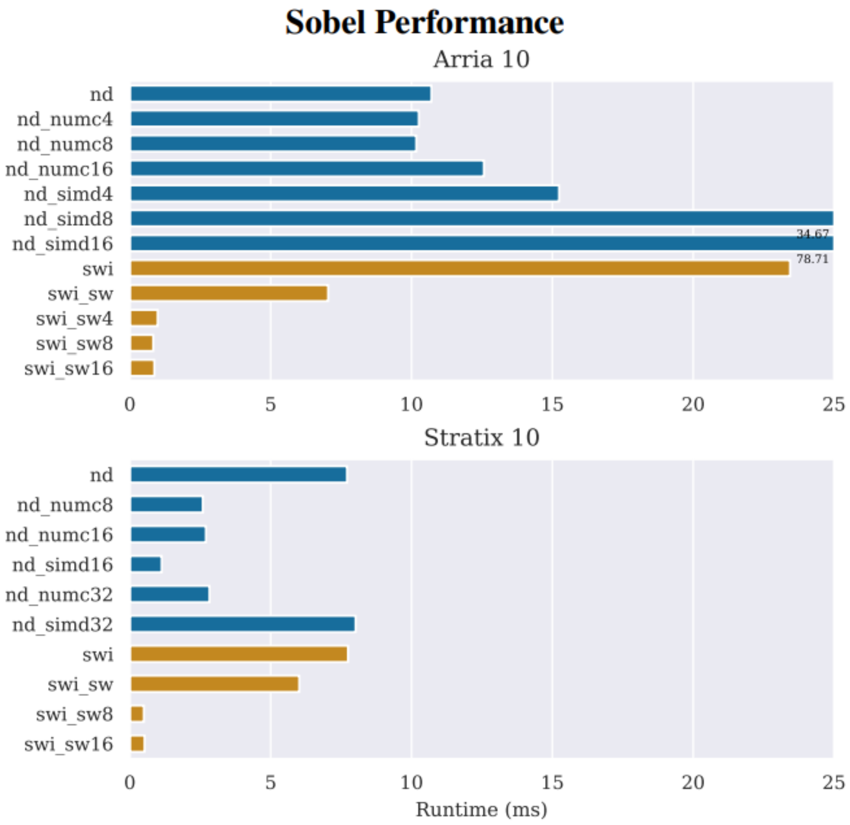


Figure 1: Runtime performance (in seconds) of Sobel with different FPGA-specific optimizations applied. black bars indicate the multi-threaded approach, and orange bars indicate the single work-item approach (smaller is better).

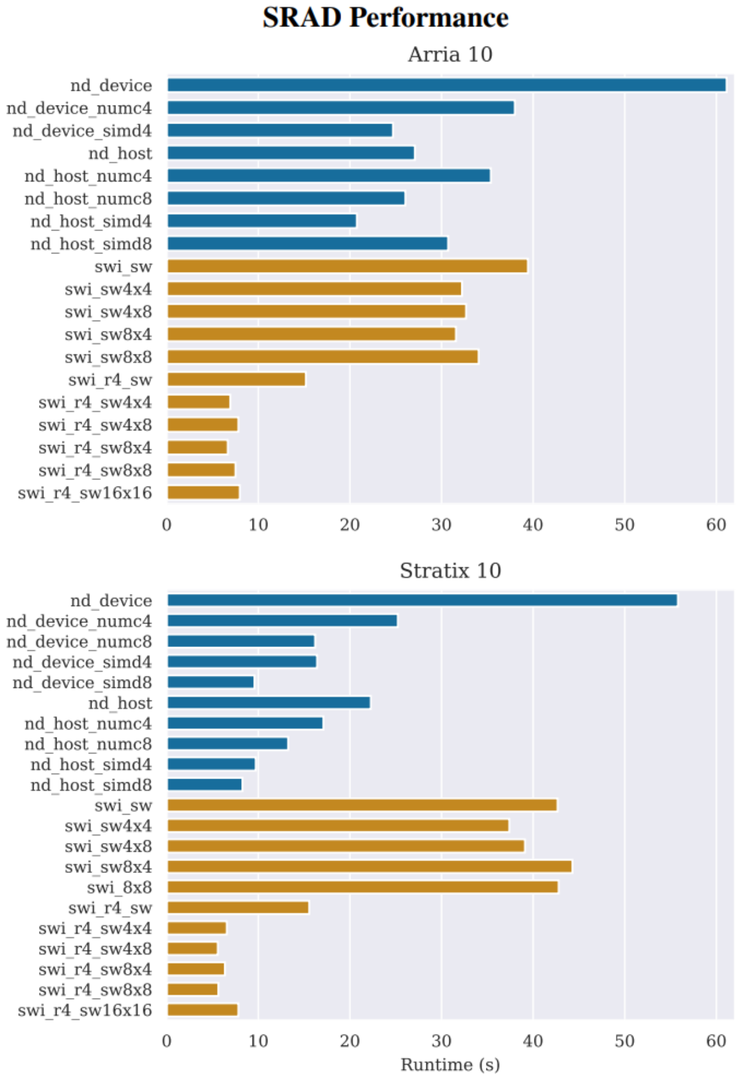


Figure 3: Runtime performance (in seconds) of SRAD with different FPGA-specific optimizations applied. Blue bars indicate the multi-threaded approach, and orange bars indicate the single work-item approach (smaller is better).

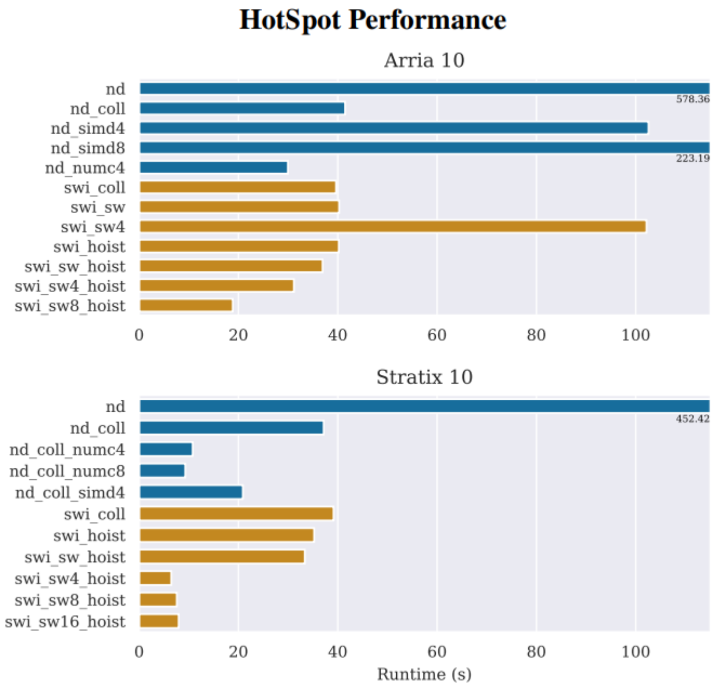


Figure 2: Runtime performance (in seconds) of HotSpot with different FPGA-specific optimizations applied. Blue bars indicate the multi-threaded approach, and orange bars indicate the single work-item approach (smaller is better).

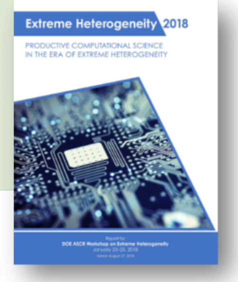
- `nd`: multi-threaded kernel
- `numcX`: number of compute units (X: replication factor)
- `simdX`: vectorization (X: replication factor)
- `elim`: kernel boundary elimination optimization
- `coll` collapse optimization
- `swi`: single work-item kernel
- `redX`: reduction optimization (X: unroll factor)
- `swX`: sliding window optimization (X: unroll factor)
- `hoist`: code motion optimization
- `flat`: 2D arrays manually flattened to 1D array.

J. Lambert, et al. "Optimization with the OpenACC-to-FPGA framework on the Arria 10 and Stratix 10 FPGAs," *Parallel Comput., 104-105:102784, 2021, doi:10.1016/j.parco.2021.102784.*

Recap

Recent trends in computing paint an ambiguous future for architectures

- Power constraints initially drove architectural changes
- Now, vendors are forced to use 2-3 foundries if they want access to leading-edge CMOS production
  - Forces vendors to add value with domain specific architectures by specializing processors, node design, memory systems, I/O
- Explosion of new architectures
  - Devices: GPUs, FPGAs, DSPs, SoCs
  - Deployment: HPC, AI, Edge, Cloud
  - OpenHW: RISC-V
- **Entering an era of Extreme Heterogeneity**



Extreme Heterogeneity 2018
PRODUCTIVE COMPUTATIONAL SCIENCE IN THE ERA OF EXTREME HETEROGENEITY

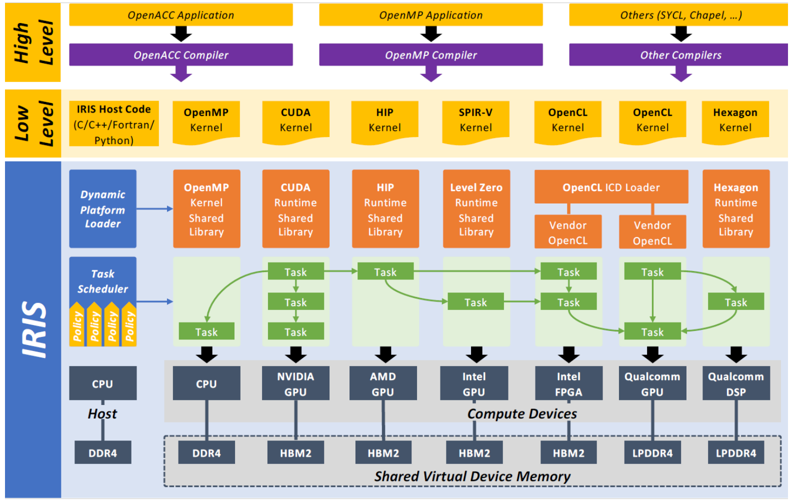
As a result, applications and software systems are all reaching a state of crisis

- Proliferation of diverse and often immature programming ecosystems
  - In fact, programming and operating systems need major investment to address current and future architectural changes
- Applications will not be *functionally* or performance portable across architectures
- Additionally, procurements, acceptance testing, and operations of today’s new platforms depend on performance prediction and benchmarking.
- **Complexity is our main challenge**
- **This is a crisis!**

Programming systems must provide performance portability (beyond functional portability)!!

- Ultimately, we should strive for *‘Write once, perform satisfactorily anywhere’*
  - Descriptive models of parallelism and data movement
  - Introspective runtime systems
  - Layered, modular, open-source approaches required
  - Performance prediction tools for design, procurement, and operations
- **Examples**
  - ECP investments in LLVM
    - FORTRAN with GPU offloading
  - Introspective Runtime Systems
  - Programming FPGAs
    - Without Verilog

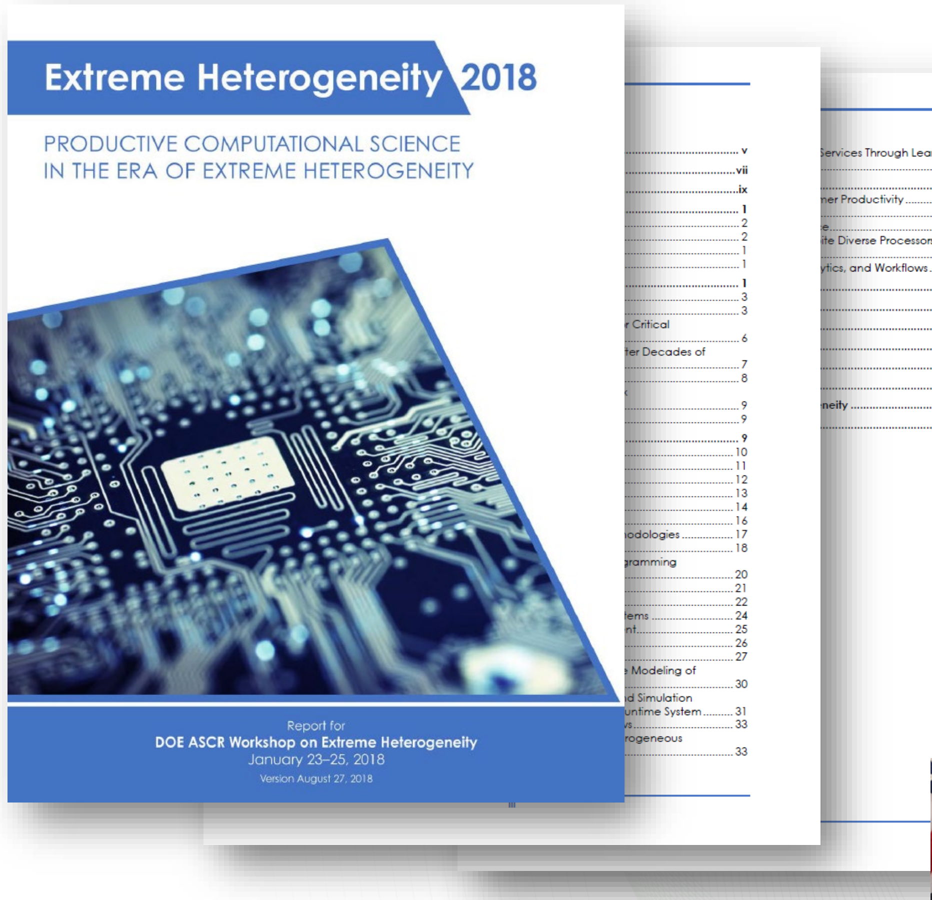
- Experimental Computing Lab
  - Lots of emerging archs
  - https://excl.ornl.gov
- Visit us (post COVID ☺)
  - We host interns and other visitors year round
    - Faculty, grad, undergrad, high school, industry
- Jobs at ORNL
  - Visit https://jobs.ornl.gov
- Contact me vetter@ornl.gov



OAK RIDGE National Laboratory

Final Report on Workshop on Extreme Heterogeneity

1. Maintaining and improving programmer productivity
   - Flexible, expressive, programming models and languages
   - Intelligent, domain-aware compilers and tools
   - Composition of disparate software components
- Managing resources intelligently
  - Automated methods using introspection and machine learning
  - Optimize for performance, energy efficiency, and availability
- Modeling & predicting performance
  - Evaluate impact of potential system designs and application mappings
  - Model-automated optimization of applications
- Enabling reproducible science despite non-determinism & asynchrony
  - Methods for validation on non-deterministic architectures
  - Detection and mitigation of pervasive faults and errors
- Facilitating Data Management, Analytics, and Workflows
  - Mapping of science workflows to heterogeneous hardware and software services
  - Adapting workflows and services to meet facility-level objectives through learning approaches



Extreme Heterogeneity 2018

PRODUCTIVE COMPUTATIONAL SCIENCE
IN THE ERA OF EXTREME HETEROGENEITY

Report for
DOE ASCR Workshop on Extreme Heterogeneity
January 23–25, 2018
Version August 27, 2018

https://orau.gov/exheterogeneity2018/

https://doi.org/10.2172/1473756