Modular Chiplet Opportunities for HPC

Evaluating Chiplets for Post Exascale Supercomputing

John Shalf

Department Head for Computer Science

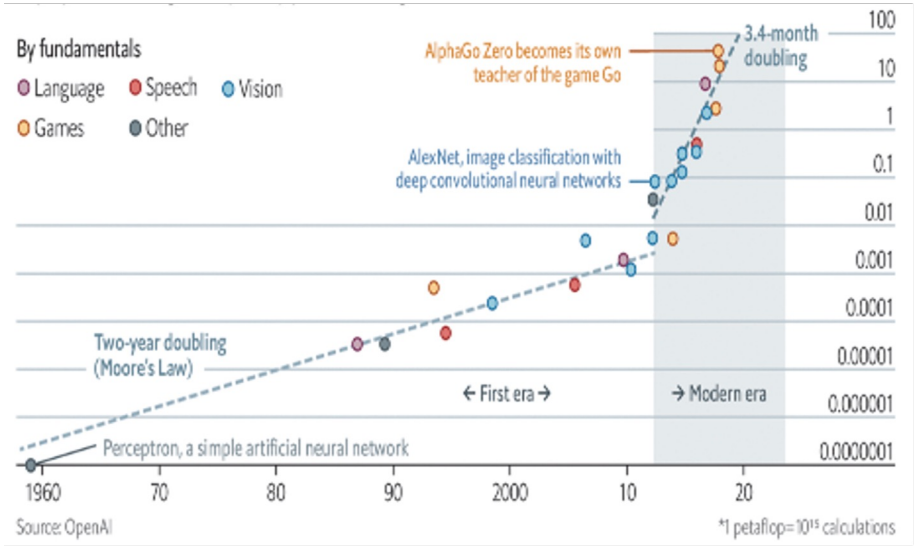
Lawrence Berkeley National Laboratory

ModSim 2024

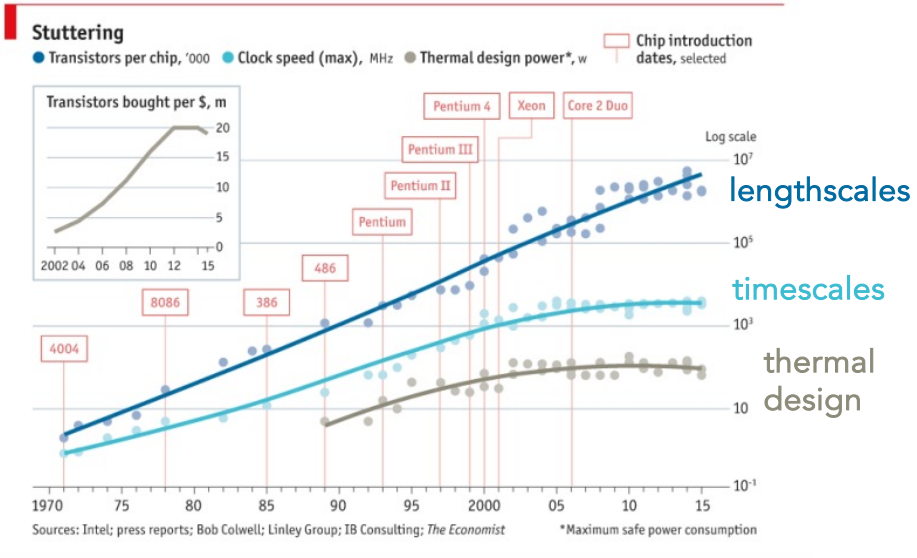
Seattle, Washington

Explosion of Computing Demand: Driving Need for Hyper-exponential Improvement in Performance, Energy Efficiency and Integration

Demand for Computing



Supply for Computing

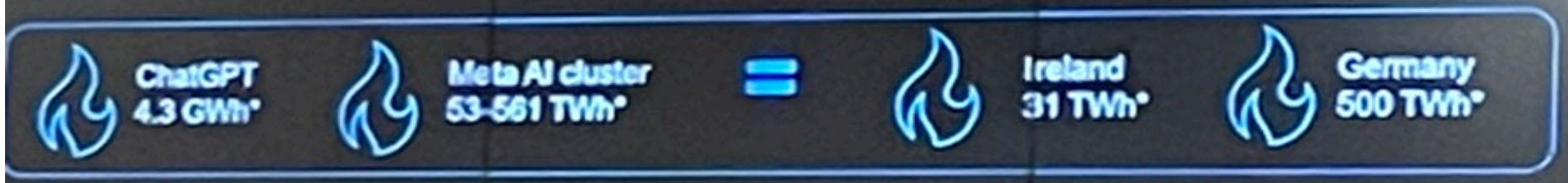


NVIDIAnomics

- **NVIDIA H100 SXM consumes 700W each**
- **Next Generation B100 is projected to consume 1400W each! (100% increase)**
- **Street price $20k-$30k**
  - That's an 800% profit
- **And still supply cannot keep up with demand**

ENERGY EFFICIENCY – UNSUSTAINABLE

AI Training Energy footprint on par with entire industrial nations



Performance Per Watt: A Critical Metric in the Age of AI

The carbon footprint of ChatGPT, An estimate of the carbon emissions... | by Chris Pointon | Dec, 2022 | Medium
https://www.semianalysis.com/p/the-disclosure-of-hardware-and-co
"Germany - Energy consumption in Germany (worldbank.org), April 2024
"Ireland - Energy consumption in Ireland (worldbank.org), Apr 1 2024
Yu Wang, Tsinghua University, Feb 2018 https://www.nvidia.com/blog/2022/05/24/ai-is-harming-our-planet/

AMD

Google Sustainability Page in Late 2023

Google Sustainability | Empowering individuals | Working together | Operating sustainably | Reports

Overview | Net-zero carbon | Water stewardship | Circular economy | Nature & biodiversity | Stories

Net-zero carbon

What a difference a year makes!

Google's AI ambitions threaten carbon neutrality pledge

Data center expansion leads to 48% Increase in emissions since 2019

Omer Kabir 09:53, 04.07.24

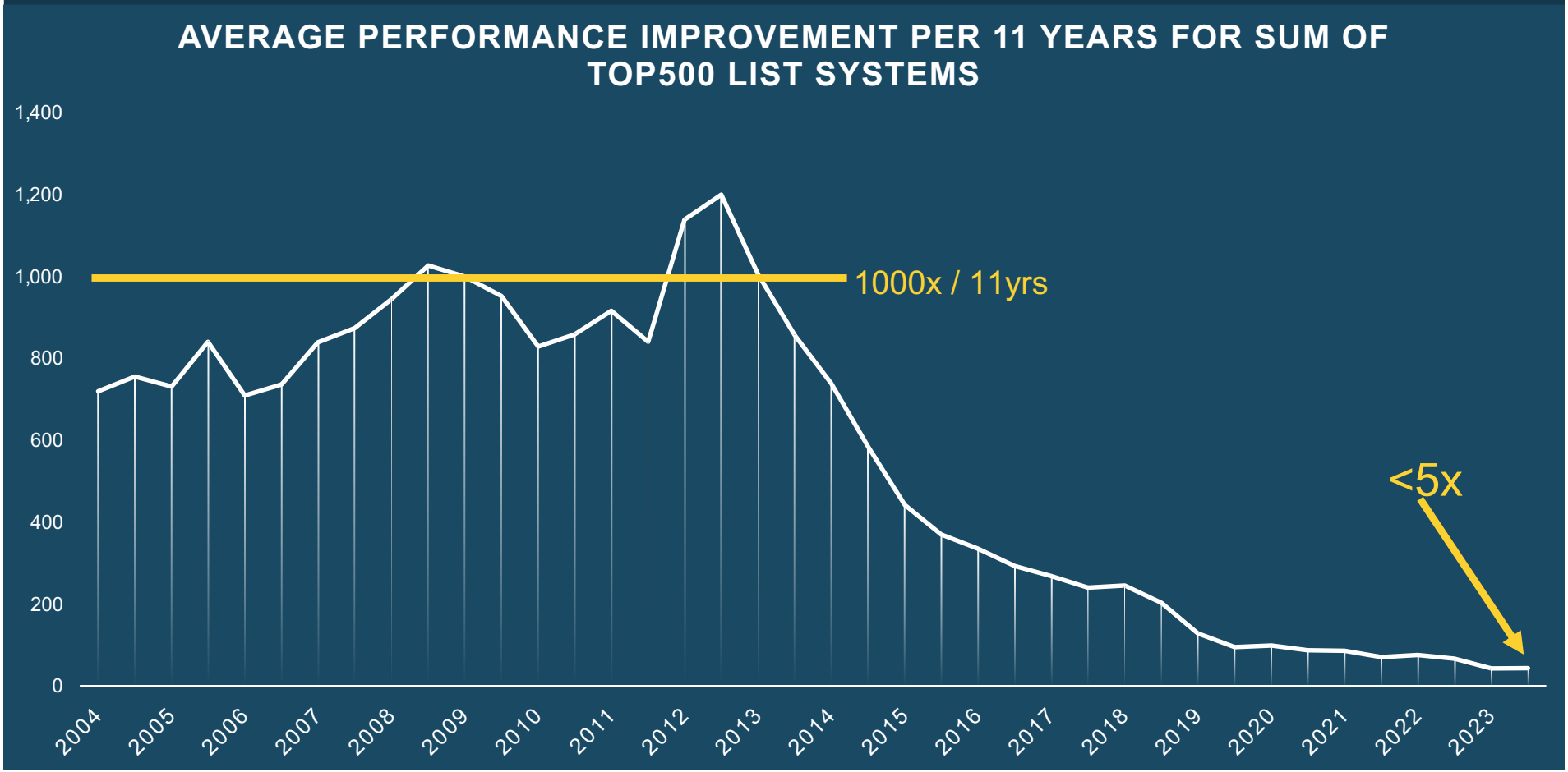
TAGS: Climate Change Data Center Google Emissions

Listen to this article now 4 min listen

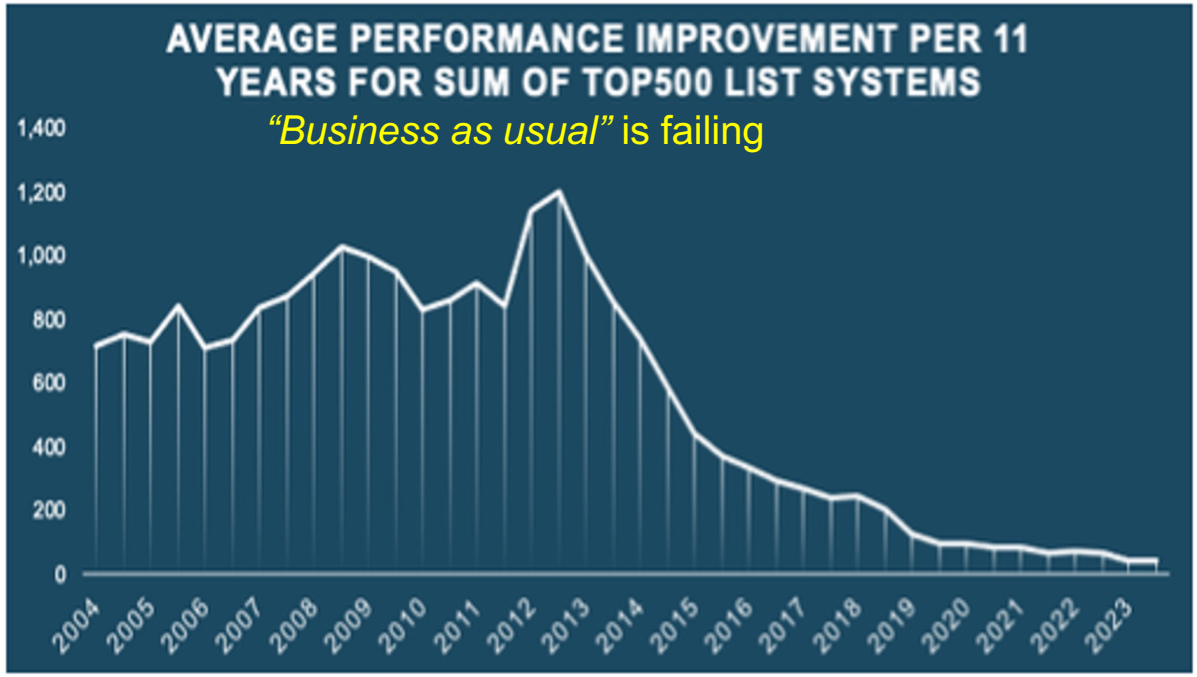
00:00 10 10 1.0× -03:48

This is HPCs future if we continue business as usual!

… and "scale" alone might not be the answer…



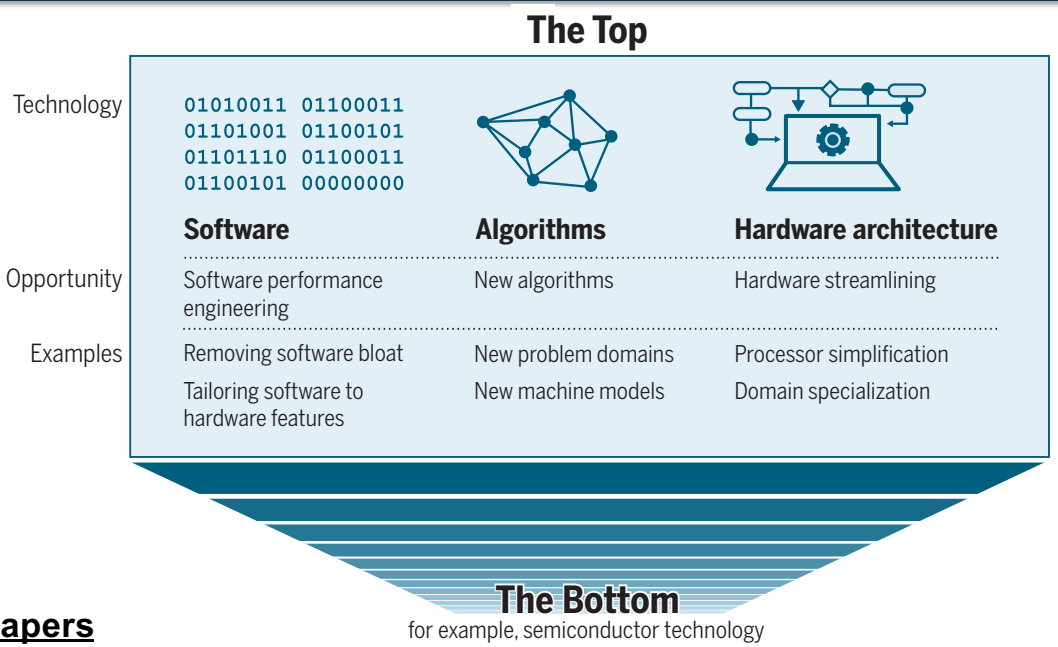
Algorithm-Driven Codesign of Specialized Architectures for Energy-Efficient HPC

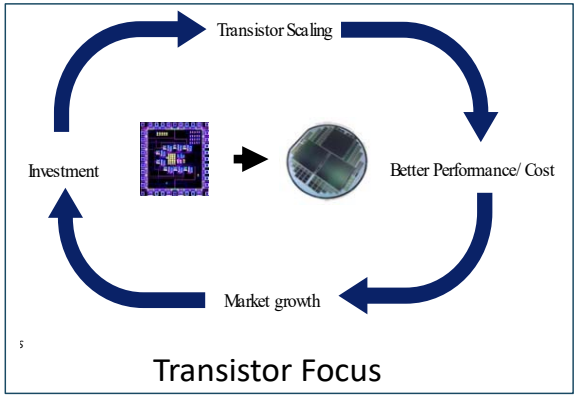


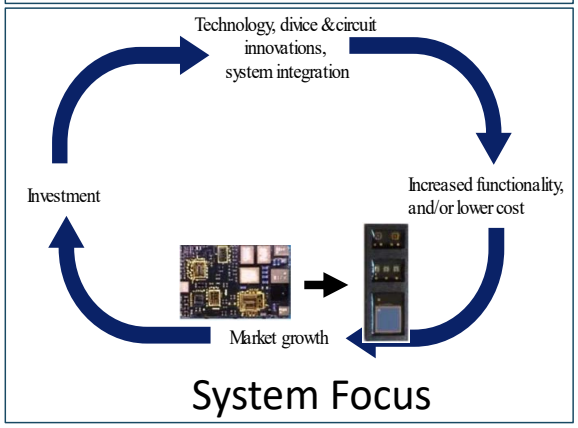
NASEM study on post-Exascale computing *"We must expand (and create where necessary) integrated teams that identify the key algorithmic and data access motifs in its applications and begin collaborative ab-initio hardware development of supporting accelerators,... a first principles approach that considers alternative mathematical models to account for the limitations of weak scaling."*

Neil Thompson: Economics of Post-Moore Electronics

http://neil-t.com, MIT CSAIL, MIT Sloan School





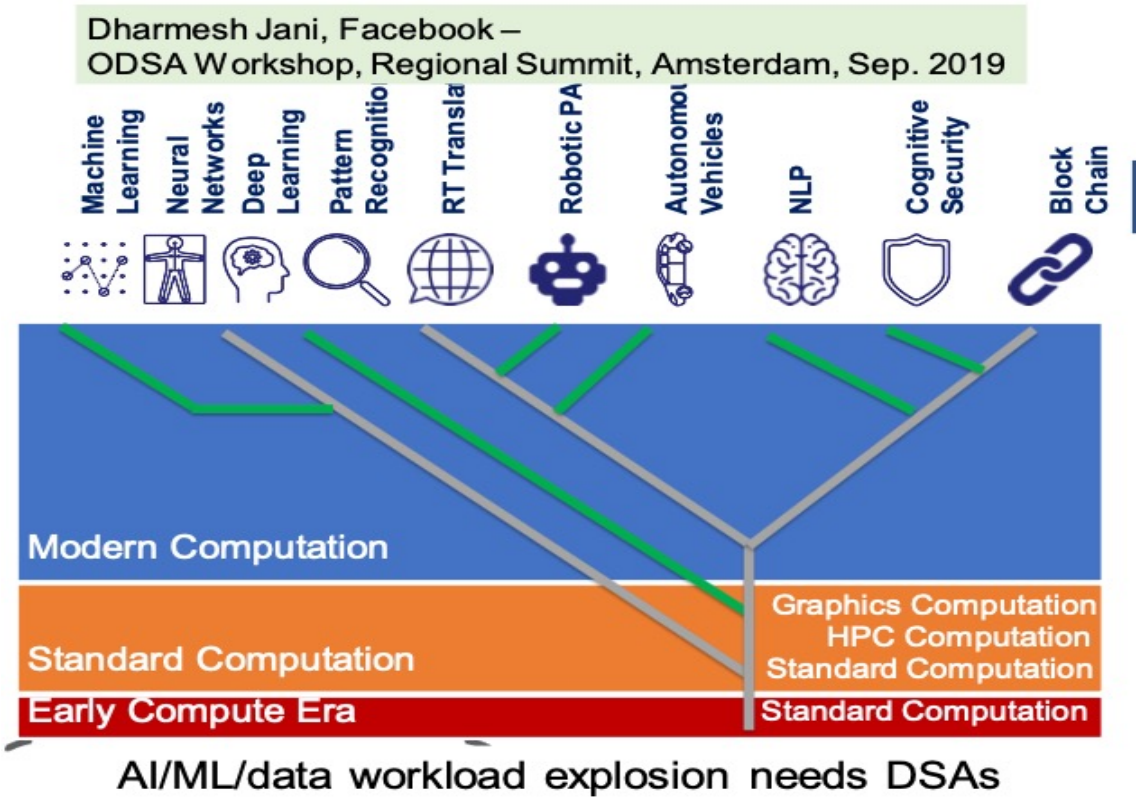


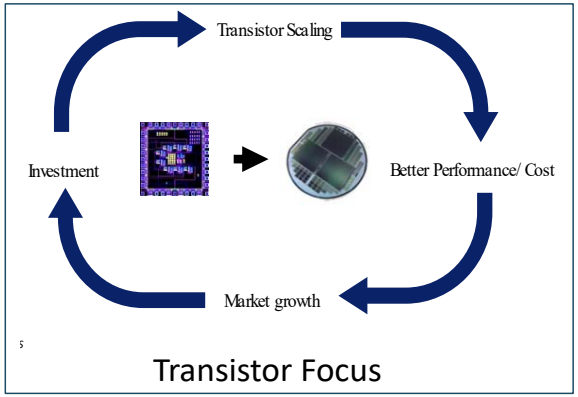
Papers

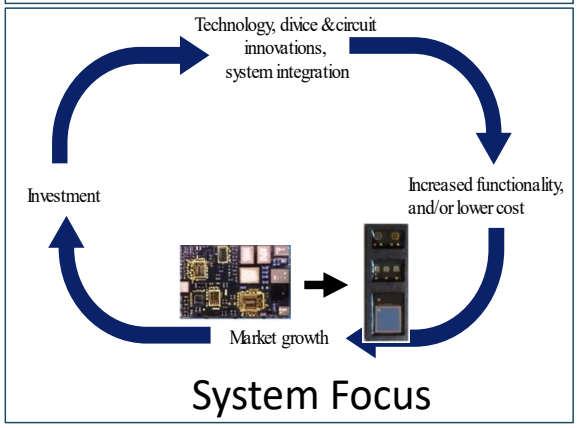
1. The Economic Impact of Moore's Law
2. There's Plenty of Room at the Top: What will drive computer performance after Moore's Law?
3. The Decline of Computers as a General Purpose Technology

Why? Domain specific Architectures driven by hyperscalers

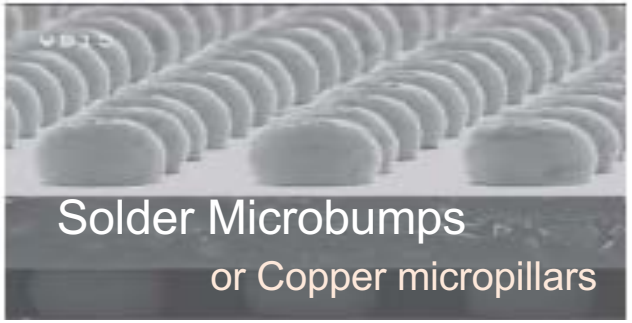
in response to slowing of Moore's Law (switch to systems focus for future scaling)

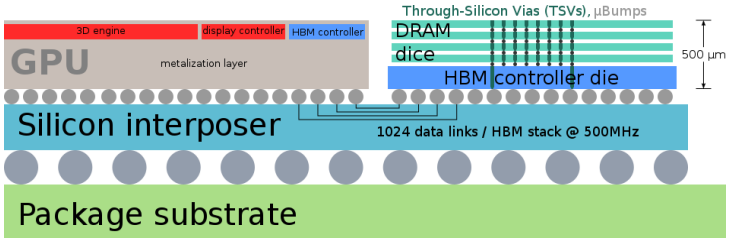


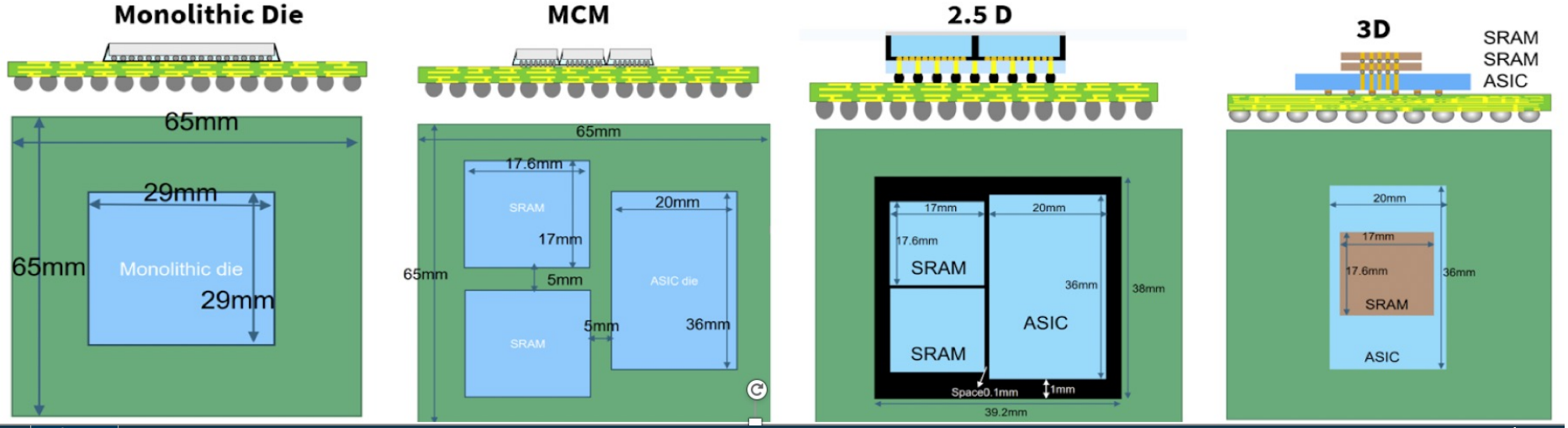




What is a Chiplet?

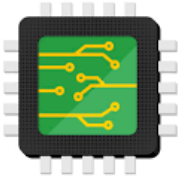


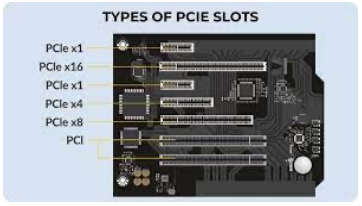




Chiplets Details

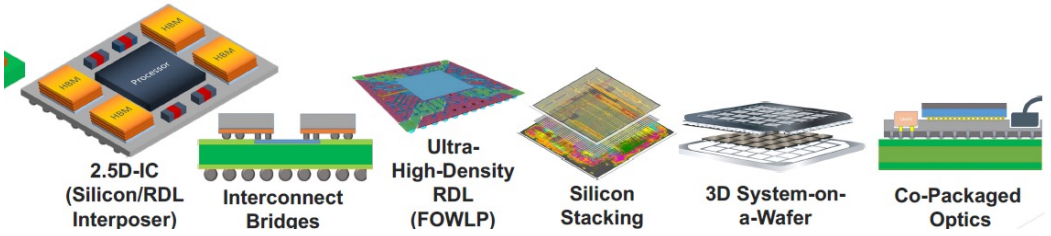
Chip: Homogenous logic, the one die in a package





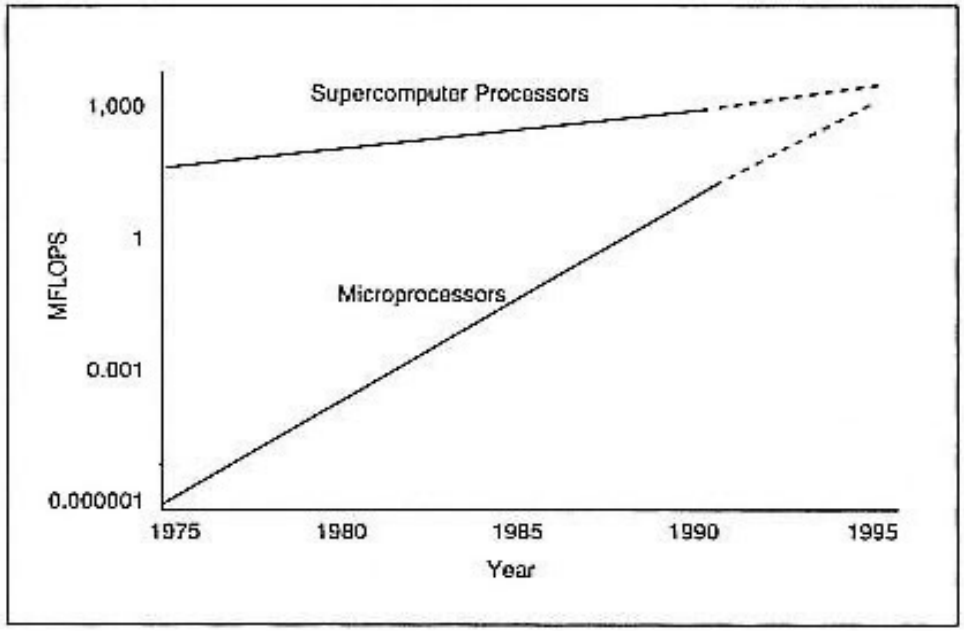
Off-package interconnect + simple package

Chiplet: Heterogeneous logic, Many die in a package



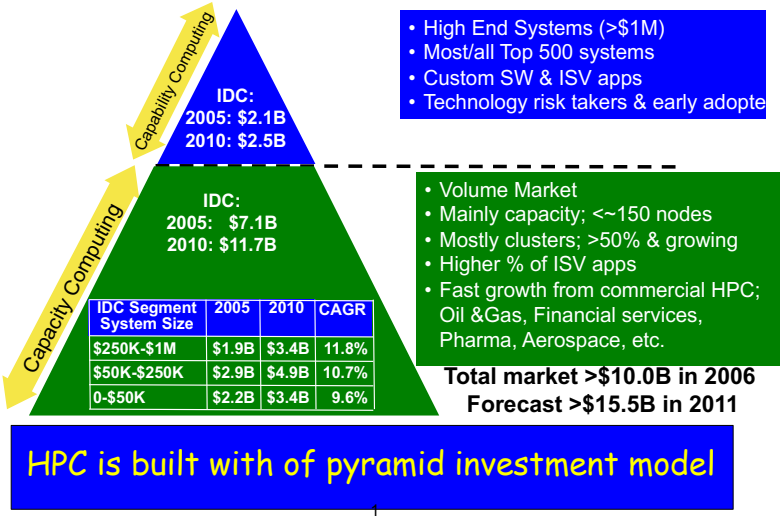
On-package interconnect + more complex packages

Learning from Attack of the Killer Micros



Attack of the killer micros
John Markoff, May 6, 1991

- **Was more about the economic model than technology alone**



Technology Insertion into Mainstream Platforms

AMD, Intel, Arm offer integration path for 3rd party accelerator "chiplets"

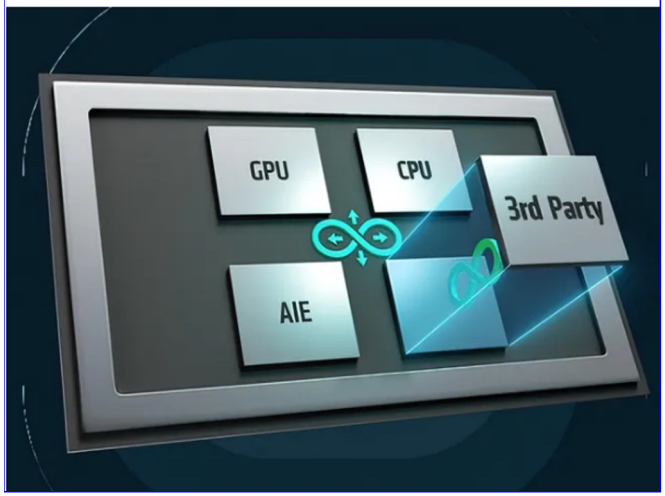
Modular AMD Chips to Embrace Custom 3rd Party Chiplets

News By Francisco Pires last updated June 20, 2022

Supercharging learnings - and earnings - from the console space.

Comments (2)

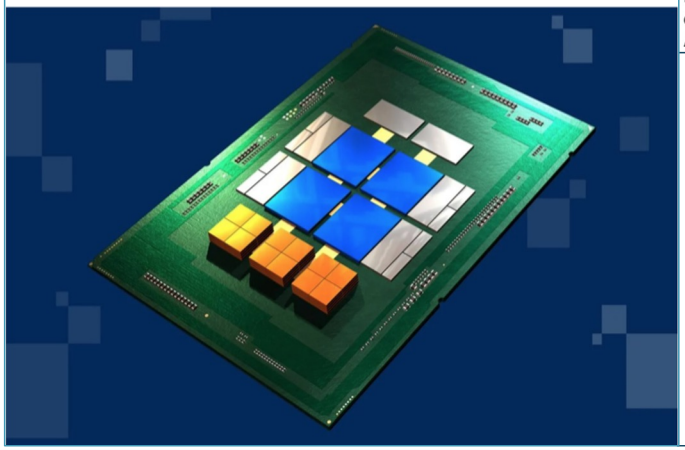
When you purchase through links on our site, we may earn an affiliate commission. Here's how it works.



To 'Meteor Lake' and Beyond: How Intel Plans a New Era of 'Chiplet'-Based CPUs

At the Hot Chips 2022 conference, Intel teased its upcoming 'Meteor Lake' and 'Arrow Lake' processor families, which will use multiple tiny tiles fused together in an attempt to break free of the limits of monolithic chip design. Here's why little tiles are a big deal.

By Michael Justin Allen Sexton August 24, 2022

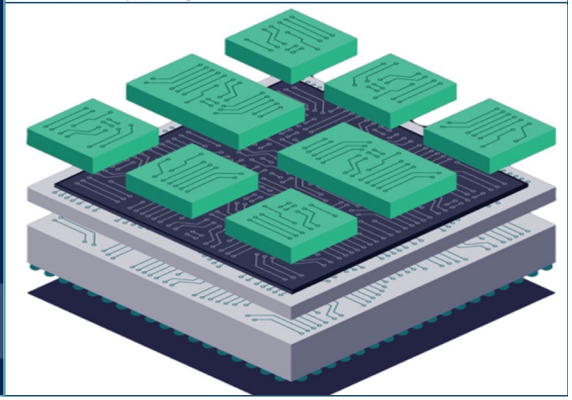


ARM Opens Door to Make Custom Chips for HPC, AI

By Doug Eadline

October 19, 2023

It is safe to say that ARM isn't a scrappy startup that was once the pride of the UK. The US-based IPO made the chip designer a big-game chip player, and the new capital is kickstarting some major initiatives to find more customers for its products. A new effort called Total Design aims at making it easier for companies looking to design chips in-house, an idea gaining ground with the AI boom and chip shortages.

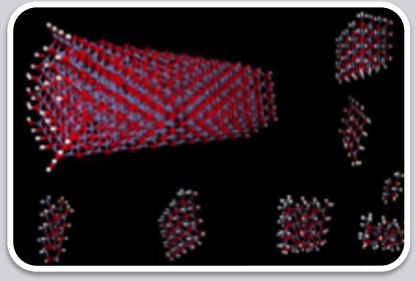


BERKELEY LAB

http://chiplets.lbl.gov/

Architecture Specialization for Science

(hardware is design around the algorithms) can't design effective hardware without math



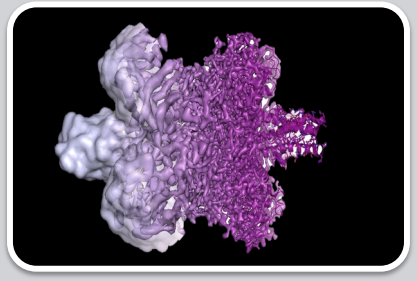
Materials

Density Functional Theory (DFT)

Use O(n) algorithm

Dominated by FFTs

FPGA or ASIC



CryoEM Accelerator

LBNL detector

750 GB / sec

Custom ASIC near detector

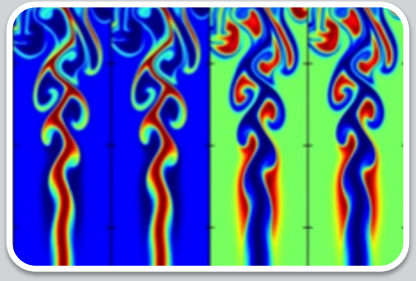
Genomics Accelerator

String matching

Hashing

2-8bit (ACTG)

FPGA solution



CFD Accelerator

3D integration

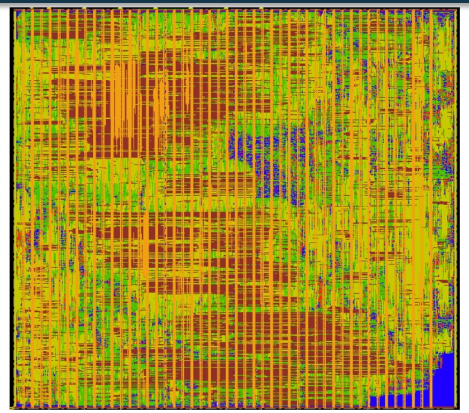
Petascale *chip*

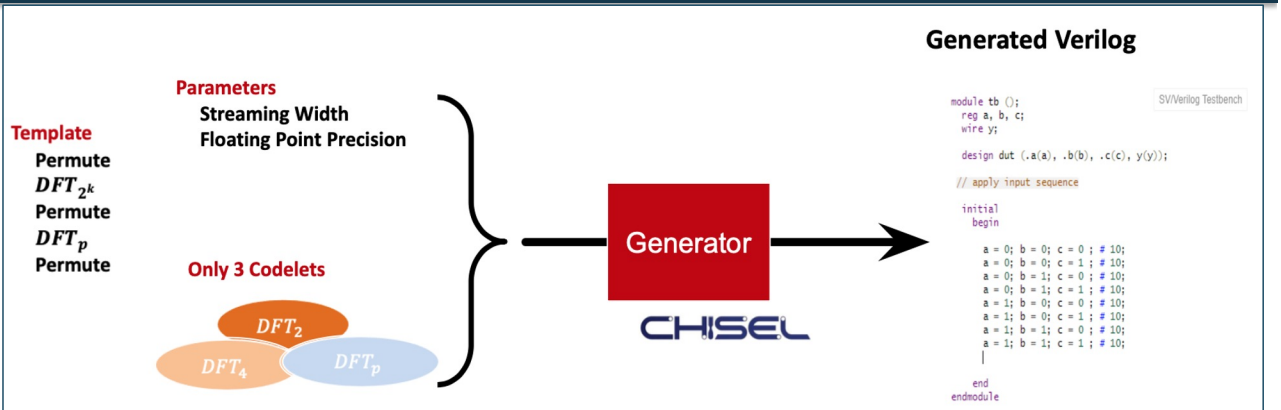
1024-layers

General / special HPC solution

Example of Mixed-Radix MultiDimensional FFT Accelerator

Initial steps towards a basis set of hardware accelerator primitives for science





FFT96 Accelerator Die: 16nm TSMC

Area eff.: 4.18 TF/mm²

Energy eff.: 4.8 TF/W

NVIDIA H100 in 4nm TSMC *(10x denser than 16nm)*

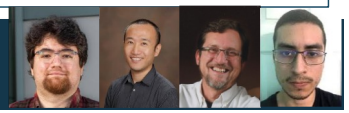
Area eff: 0.08 TF/mm²

Energy eff: 0.0957 TF/W

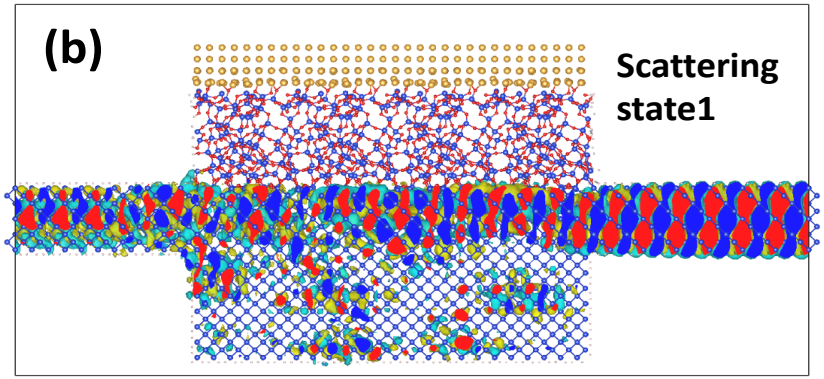
Accomplishment: Demonstrated general FFT Accelerator tile generator in 16nm TSMC that outperforms NVIDIA H100 by 50x in raw performance/area and 50x in energy efficiency/flop

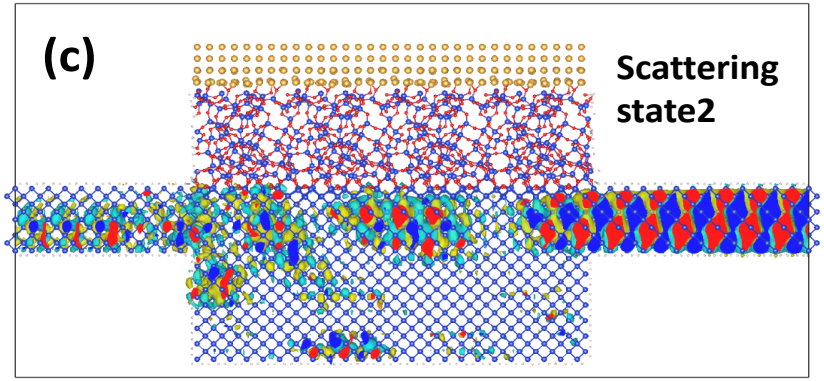
–Required only 3 codelet primitives

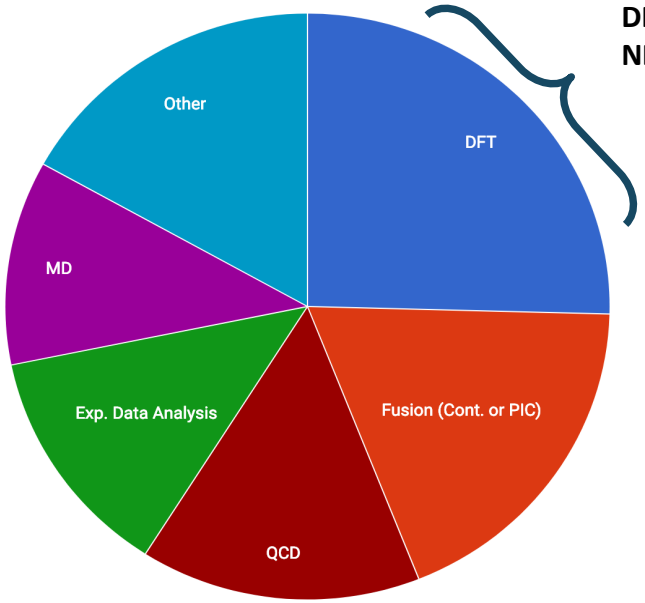
Goal: Generalize this approach to generators for accelerator hardware primitives that cover broad spectrum of algorithms *(e.g., FFT, Dense/Sparse Linear Algebra, particles and PDEs)*



The Importance for High Performance DFT





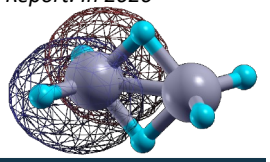


DFT: 25% of NERSC workload

B. Driscoll and Z. Zhao. 2020. Automation of NERSC Application Usage Report. In 2020 IEEE/ACM International Workshop on HPC User Support Tools (HUST)

binit

Qbox
First-Principles Molecular Dynamics



BERKELEY LAB

Circuit/Dataflow Reformulation of DFT

Eigenvalue Problem

Eigenvalue Problem:

$$h(i,j) = \langle \psi_i | H | \psi_j \rangle$$

Hpsi

$$P_i = H\psi_i - \epsilon_i \psi_i$$

Projection

$$P_i = A\left(P_i - \frac{\lambda_i}{\lambda_i^0} P_i^0\right)$$

$$P_i = P_i - \sum_{j=1,i} \langle P_i | \psi_j \rangle$$

nline

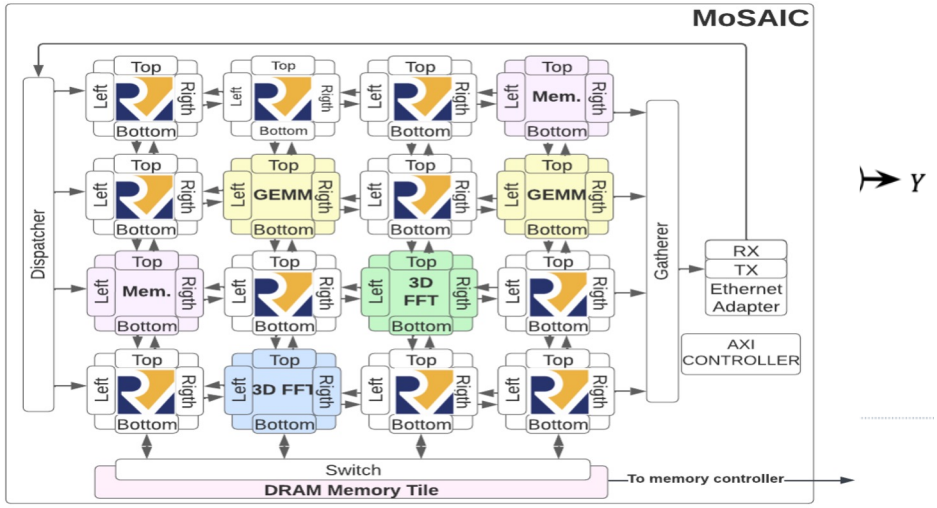
Orthogonalization

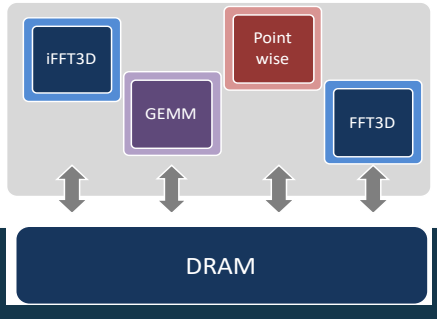
$$\psi_i = \psi_i \cos\theta_i + P_i \sin\theta_i$$

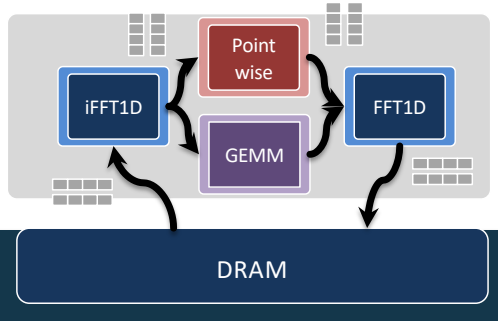
$$\psi_i = \psi_i - \sum_{j<i} \langle \psi_i | \psi_j \rangle$$

$$h(i,j) = \langle \psi_i | H | \psi_j \rangle$$

Dataflow/Circuit Algorithm Reformulation

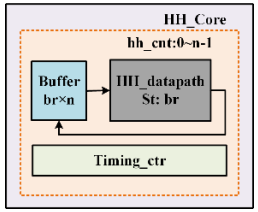






Building Flexible Accelerators

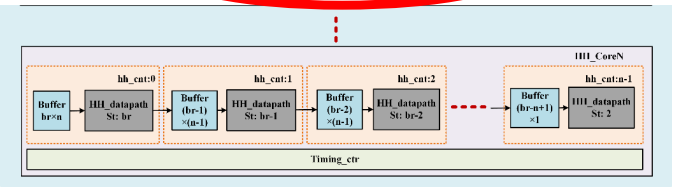
Parameterized Hardware Generation with CHISEL



A Single HH-core

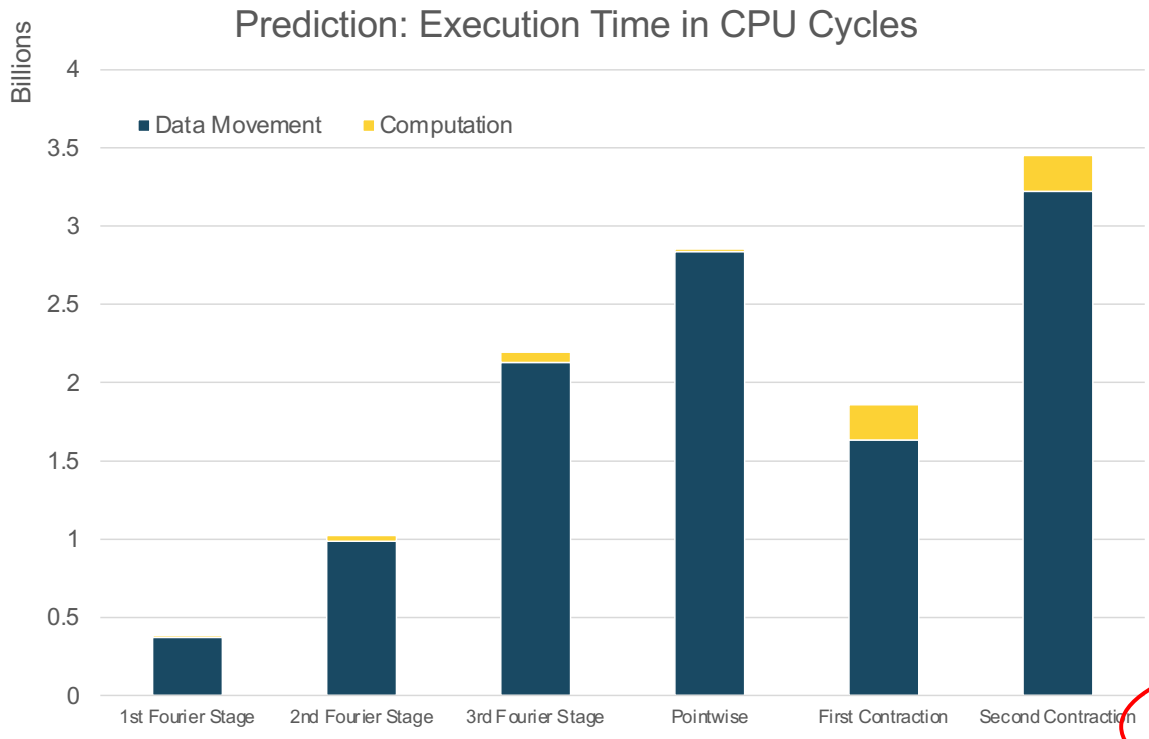
Table 1.3 Resource Cost Analysis

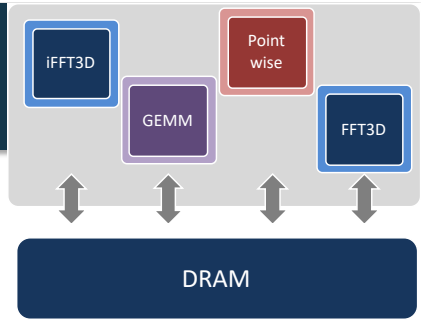
TSQR Designs		FP Operators			Register file (Words)		
Multi-core	Streaming Width	Iterative	Pipeline	Parallel (16 cores)	Iterative	Pipeline	Parallel (16 cores)
TSQR Designs	16	68	432	1,088	192	432	3,072
	32	132	1,632	2,112	768	3672	12,288
	64	260	6,336	4,160	3,072	28,336	49,152
	128	516	24,960	8,256	12,288	222,560	196,608
	256	1,028	99,072	16,448	49,152	1,764,032	786,432



Generate any level of concurrency or pipelining

Modeling the Baseline Computation
Limitations of the Offload Model for Accelerators





Assumptions:

A. Hardware

- CPU with 8 cores
- CPU frequency is 3.9 GHz
- DRAM bandwidth 22 GB/s

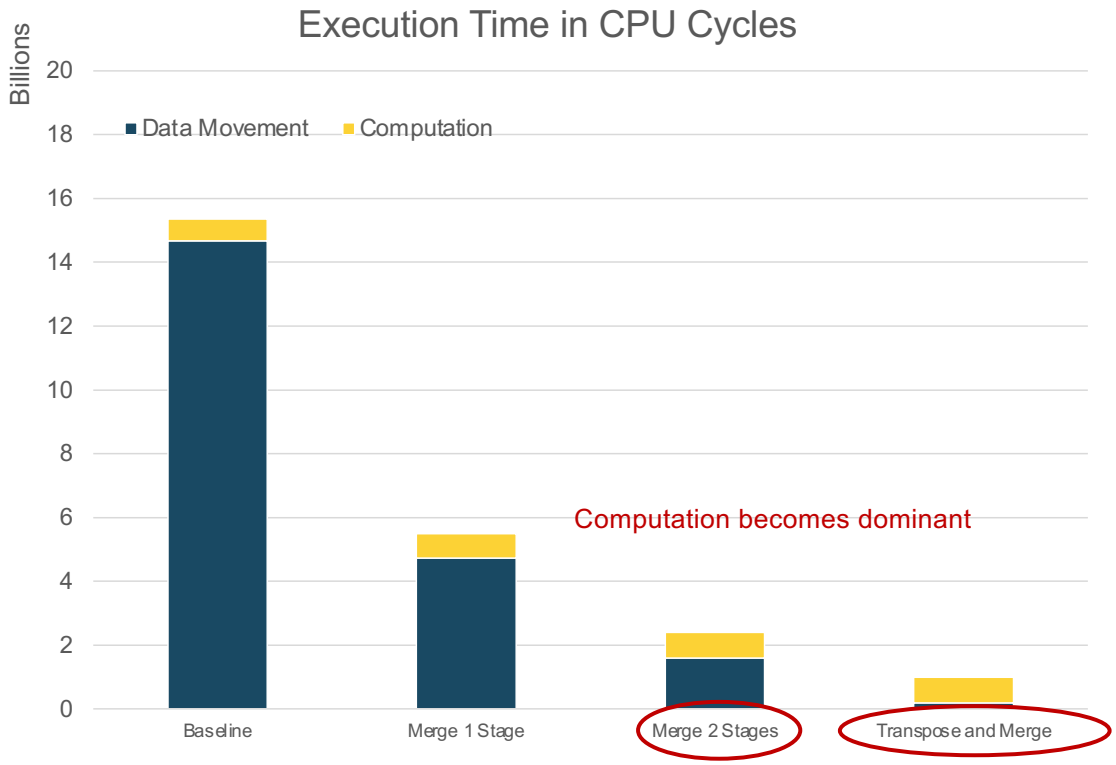
A. Problem Size

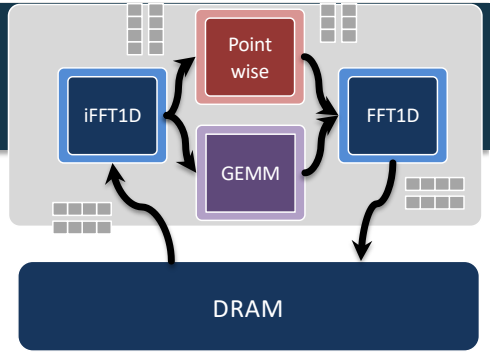
- Sphere diameter 64
- FFT size 128
- Number of bands 256
- Number of atoms 256

Prediction: **15.4 billion cycles**
Actual execution: **17.1 billion cycles**
Accuracy - 90%

Peer Accelerator Model (merged Kernels)

Convert Problem to Compute Bound





Assumptions:

A. Hardware

- CPU with 8 cores
- CPU frequency is 3.9 GHz
- DRAM bandwidth 22 GB/s

A. Problem Size

- Sphere diameter 64
- FFT size 128
- Number of bands 256
- Number of atoms 256

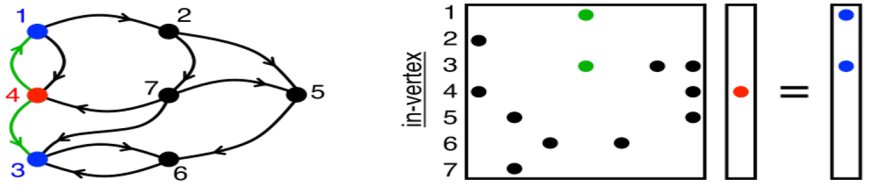
Merging different stages

The Transpose and Merge version requires a lot of on chip memory

GraphBLAS for Machine Learning and Graphs

http://graphblas.org

GRAPHBLAS



Higher-level machine learning tasks

Logistic Regression, Support Vector Machines	Dimensionality Reduction (NMF, CX, PCA)	Clustering (e.g., MCL, Spectral Clustering)	Partial Correlation Estimation (CONCORD)	Deep Learning (Neural Nets)

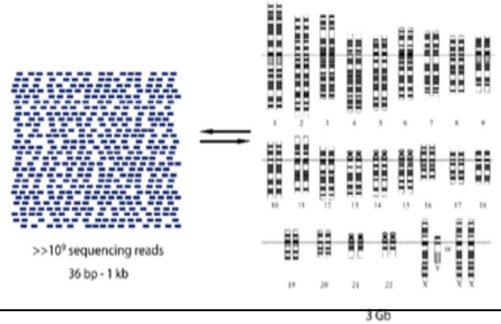
Sparse Matrix-Sparse Vector (SpMSpV)	Sparse Matrix-Dense Vector (SpMV)	Sparse Matrix-Multiple Dense Vectors (SpMM)	Sparse x Sparse Matrix (SpGEMM)	Dense Matrix-Vector (BLAS2)	Sparse x Dense Matrix (SpDM^3)	Dense Matrix-Matrix (BLAS3)

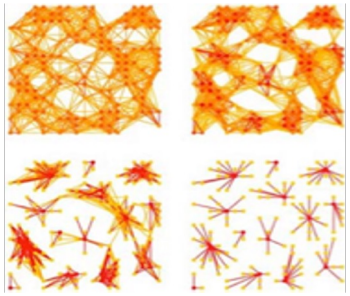
BERKELEY LAB

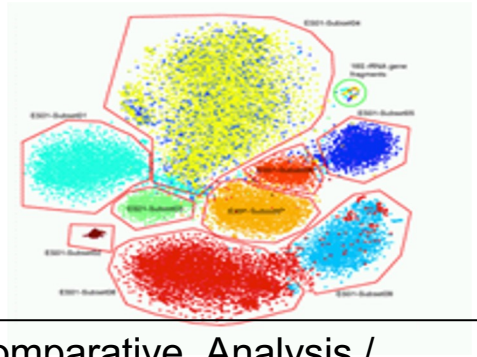
GraphBLAS and other existing BLAS functions (in increasing arithmetic intensity)

HPC Microbiome Analysis using GraphBLAS

- Use HPC algorithms and systems for orders of magnitude speedup and to solve previously intractable problems

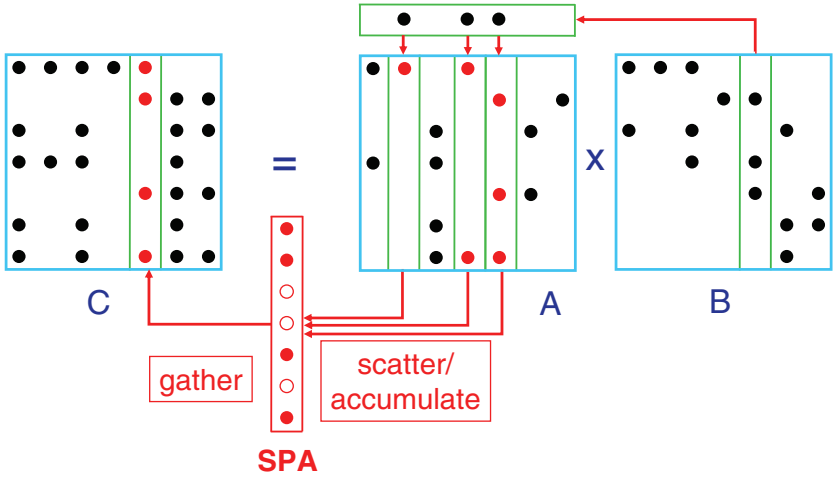




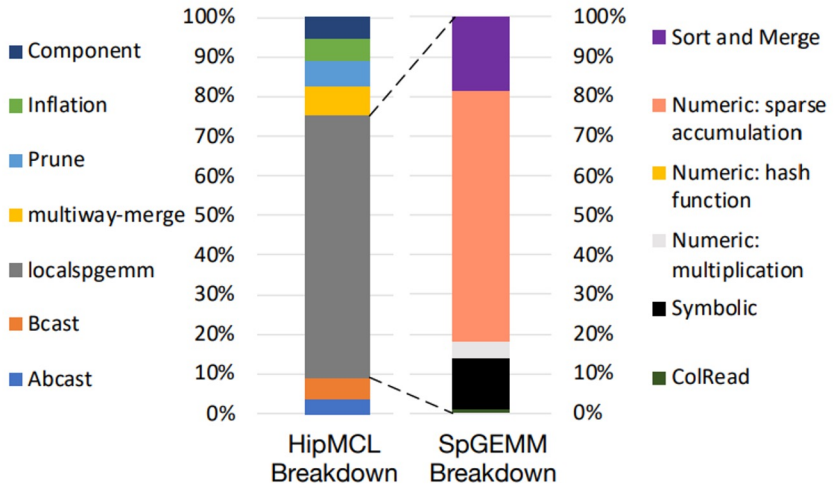


Metagenome Assembly	Protein Clustering	Comparative Analysis / Community Detection
Assemble millions of metagenomes based on incomplete data	Cluster billions of proteins	Use fast alignment and annotation for time-sensitive analyses.
Graph algorithms, De-Bruijin graphs, Hash Tables, alignment (Smith-Waterman)	Machine learning (clustering), sparse linear algebra / graphs	Alignment, Machine learning (dimensionality reduction), linear algebra

ASA: Accelerator for GraphBLAS Sparse Accumulation Primitives



Performance Breakdown of HipMCL and SpGEMM

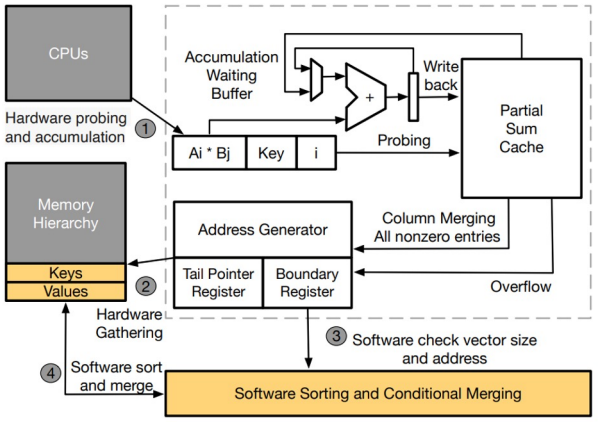


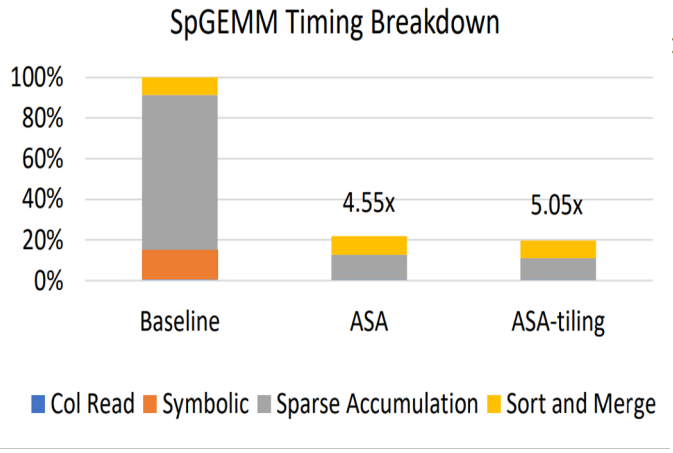
Majority of time goes to hypersparse accumulation

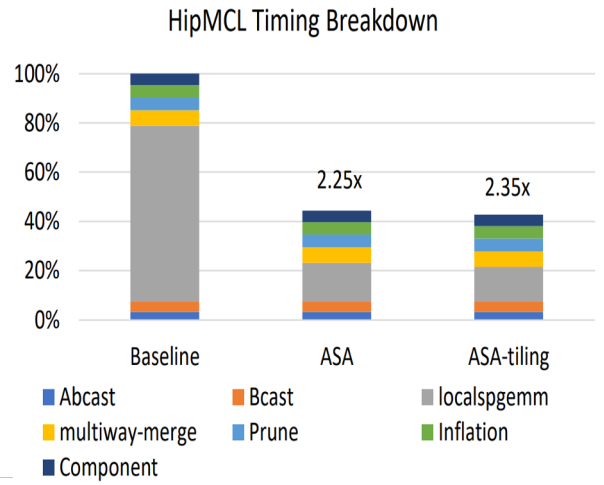
- Typically implemented as a hash table
- Replaced has lookup with an associative match
- Reaches maximum efficiency with 16k associative

SpGEMM and HipMCL Speedup

One Tile of ASA Architecture Diagram







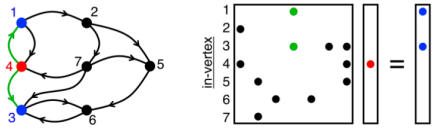
- ASA accelerator results in 5.05x speedup (average over test inputs) for SpGEMM kernel and 2.35x speedup for HipMCL application overall
- Reduced sparse accumulation cost and ***eliminated symbolic computation phase***

Accelerating GraphBLAS Primitives

GraphBLAS[1]

Graph algorithms are difficult to implement efficiently in hardware since it requires both **graph expertise** and **hardware expertise**

- GraphBLAS separates these concerns by expressing graph algorithms as linear algebraic operations



ASA: Accelerating Sparse Accumulation[2]

Hardware accelerator that extends the ISA to add dedicated instructions for partial sums and accumulates in sparse matrix matrix multiplication to avoid data dependent branches.

Can this accelerator targeting sparse general matrix-matrix multiplication (SpGEMM) be generalized for a wider range of graph algorithms?

GISA: GraphBLAS Instruction Set Architecture

GraphBLAS to GISA

GraphBLAS Operation	GISA Equivalent
Vector_build Matrix_build	GISA.malloc addr
Matrix_extractTuples	GISA.gather
reduce	GISA.gather_sum
$C_{[:,j]} = A \oplus.\otimes B_{[:,j]}$	for $B_{[k,j]}$ in $B_{[:,j]}$ for $A_{[i,k]}$ in $A_{[:,k]}$ GISA.insert_add i $A_{[i,k]} \otimes B_{[k,j]}$ $C_{[:,j]}$ = GISA.gather

The flexibility of the $\oplus$ and $\otimes$ operators enables the expression of a wide range of graph algorithms. GISA captures this by utilizing an ALU to perform the "plus" and "times" operations.

Microarchitecture

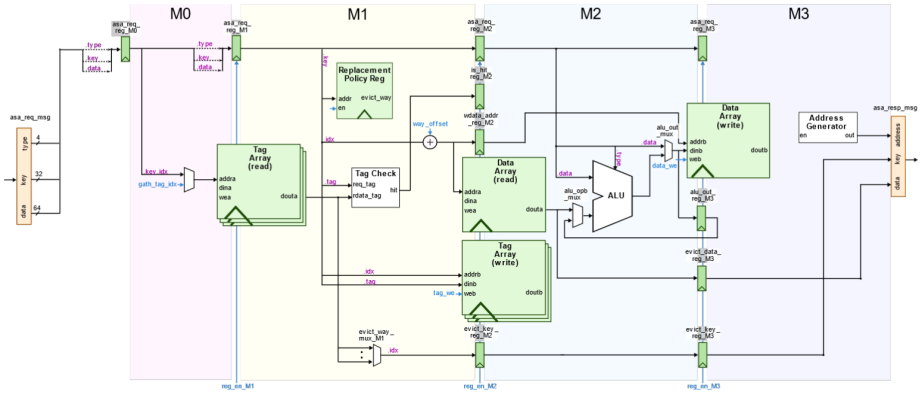
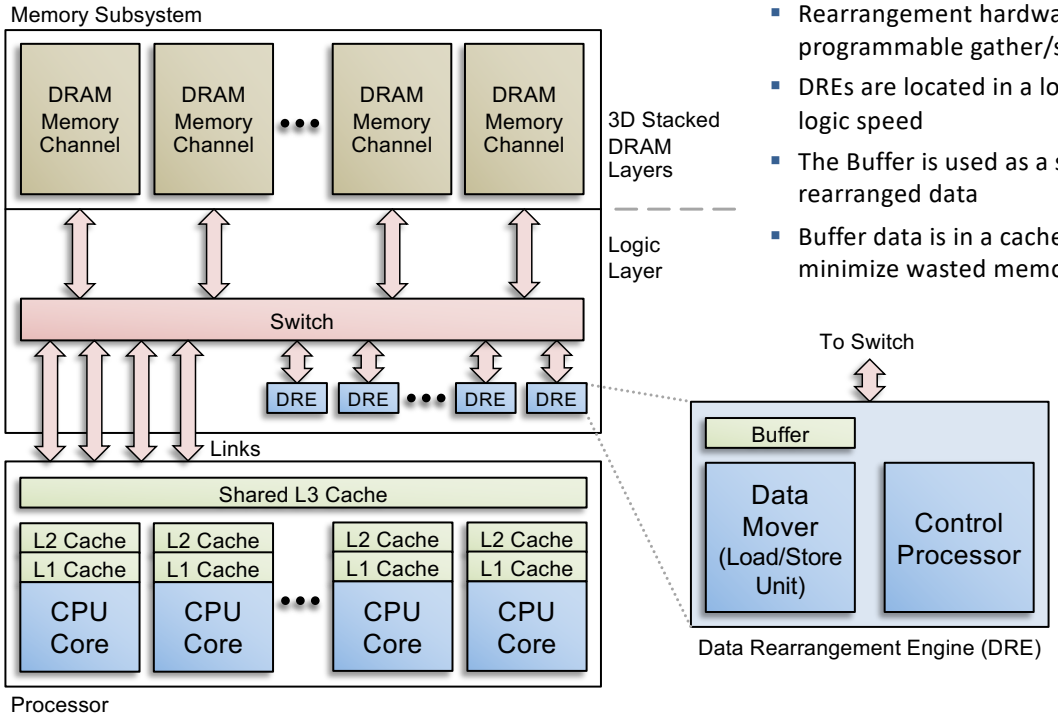


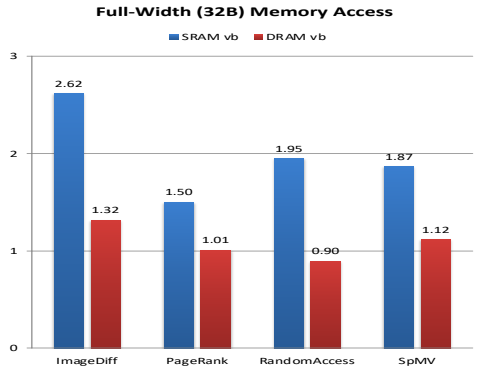
Figure 1: Datapath of GISA hardware accelerator. This has been written in SystemVerilog and is parameterizable by size and associativity

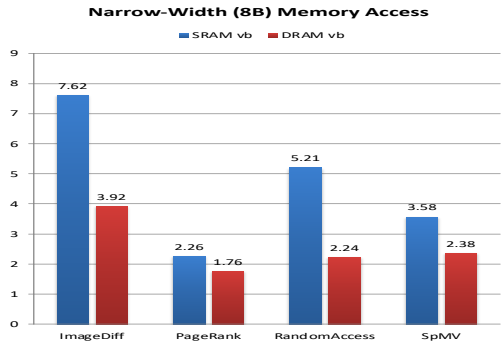
Data Rearrangement Engine (DRE) Near Memory

Xueyang Liu, Patricia Gonzalez-Guerrero, Ivy Peng, Ronald G. Minnich, Maya B. Gokhale: *Accelerator integration in a tile-based SoC: lessons learned with a hardware floating point compression engine*. SC 2023: 1662-1669



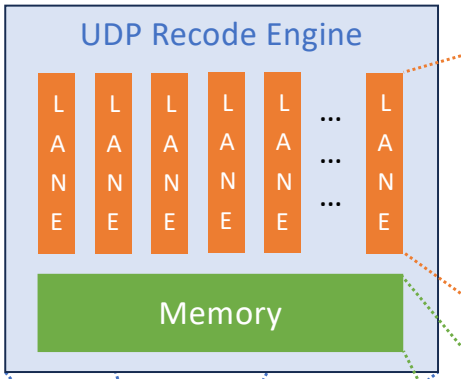
- Rearrangement hardware performs programmable gather/scatter
- DREs are located in a logic layer and run at logic speed
- The Buffer is used as a scratchpad for rearranged data
- Buffer data is in a cache-friendly layout to minimize wasted memory bandwidth





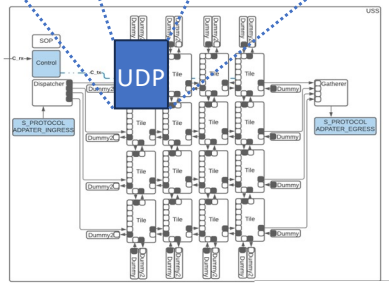
Recode Engine Accelerator

Unstructured Data Processor



UDP Lane

- Simple architecture designed for efficient encoding/transcoding
- Designed for fast symbol and branch processing
- ISA has support for variable symbol size operations
- ISA has native support for Automata transitions
- Hardware data streaming support



Local Memory

- Fast access from the different lanes
- Used as streaming buffer or scratchpad memory
- Holds encoded input and recoded output data

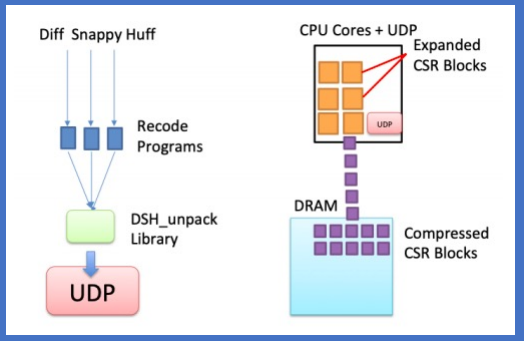
THE UNIVERSITY OF CHICAGO
Argonne NATIONAL LABORATORY

Andrew Chien, Valerie Taylor
Jose Monsalve Diaz, Dawson Fox

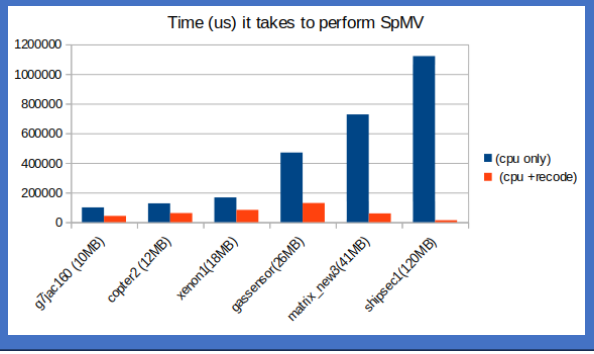
EXAMPLE

SpMV with compressed CSR Block

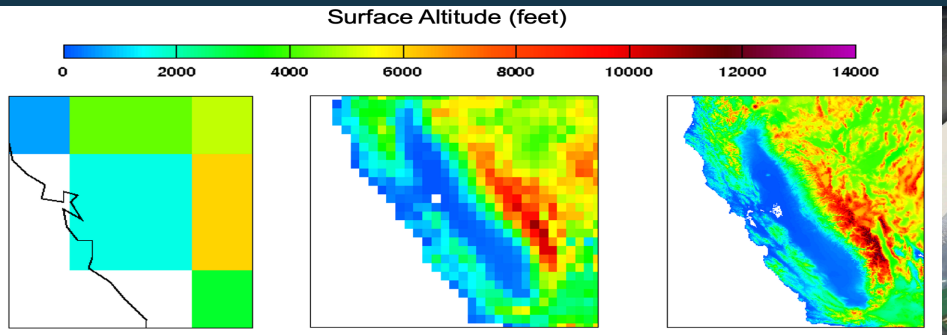
Data movement is expensive both in energy and bandwidth. Recode Engine is a high-performance accelerator that can be used to compress & decompress in an energy-efficient way.



In our case, Recode-Engine decompresses Sparse Matrices for the host to compute.



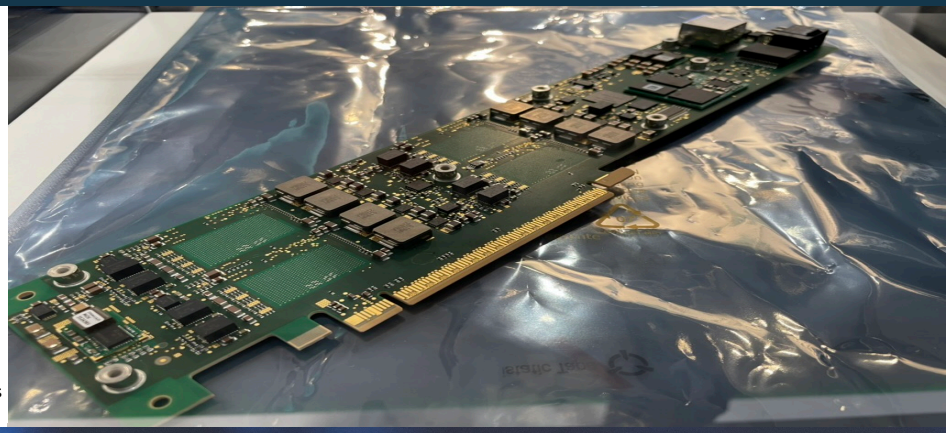
Science Chiplets Opportunity (matches hand-tuned H100 performance w/12nm tech)



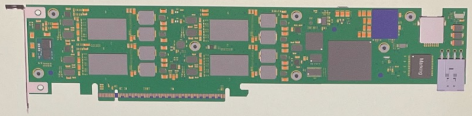
200km
Typical resolution of IPCC AR4 models

25km
Upper limit of climate models with cloud parameterizations

1km
Cloud system resolving models are a transformational change

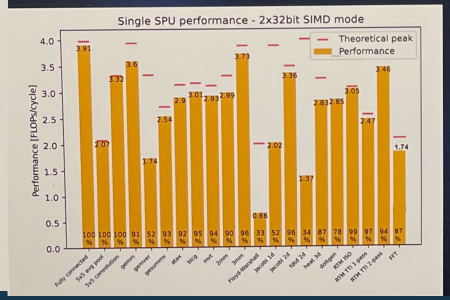


Hardware Acceleration for Stencil Codes



PCIe Gen 5 · x16 Card · 64 GB HBM memory
1,6 TB/s Memory Bandwitdh · 64 GB/s IO
180 Watt Power Consumption

Benchmark Results



RTM TTI Benchmark

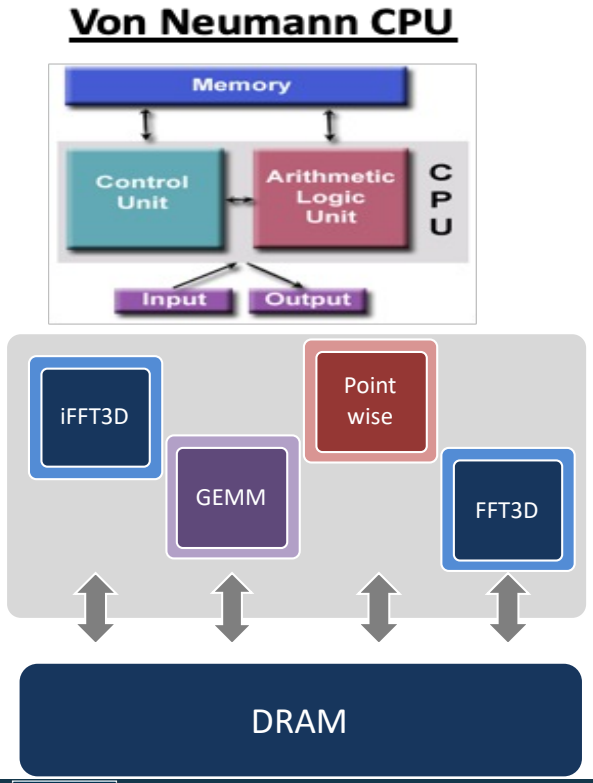
SYSTEM	POWER CONSUMPTION	MSTENCILS/S
NVIDIA H100 SXM	700 WATT	21.700
STX CARD	180 WATT	23.400

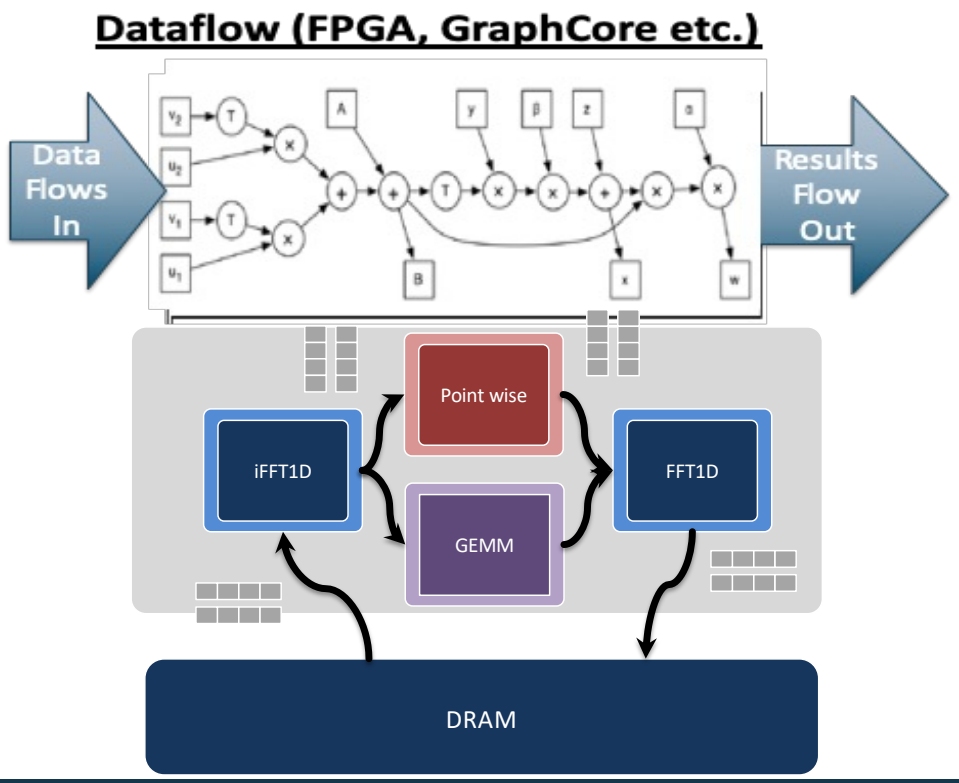
https://developer.nvidia.com/hpc-application-performance

BERKELEY LAB
Fraunhofer ITWM

Offload model is an unproductive way to use chiplets-base accel

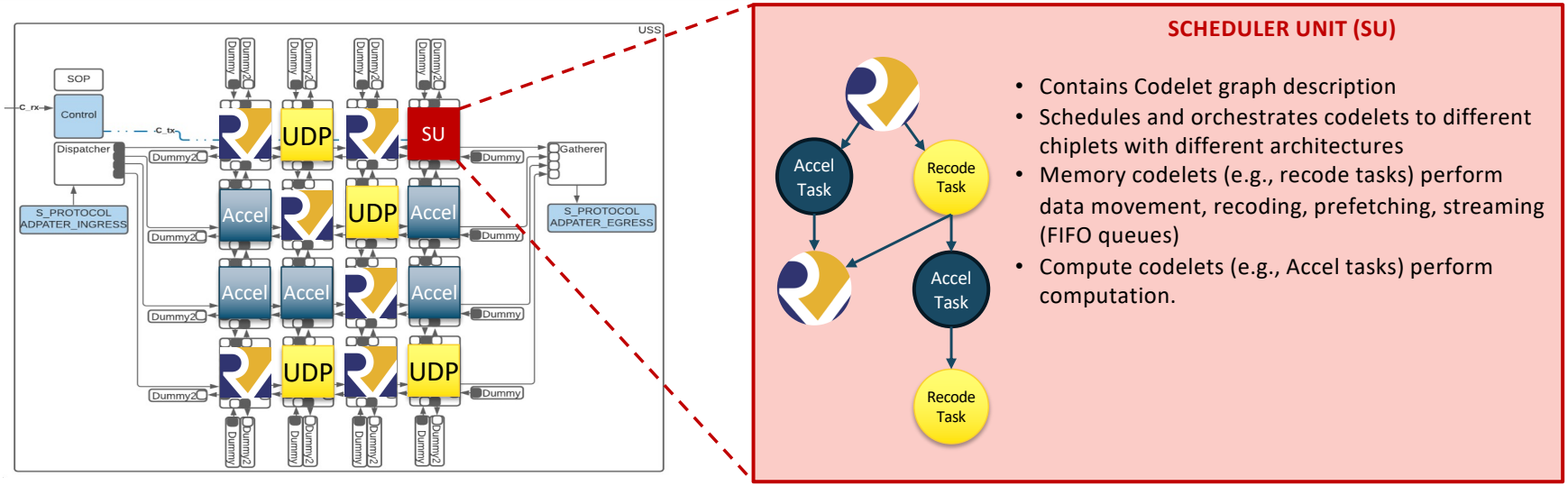
(redesign for static dataflow and deep flow-through pipelines?)





Codelet Program Execution Model for Chiplets?
We have to execute fast enough to run full codes

Guang Gao
Jose Diaz

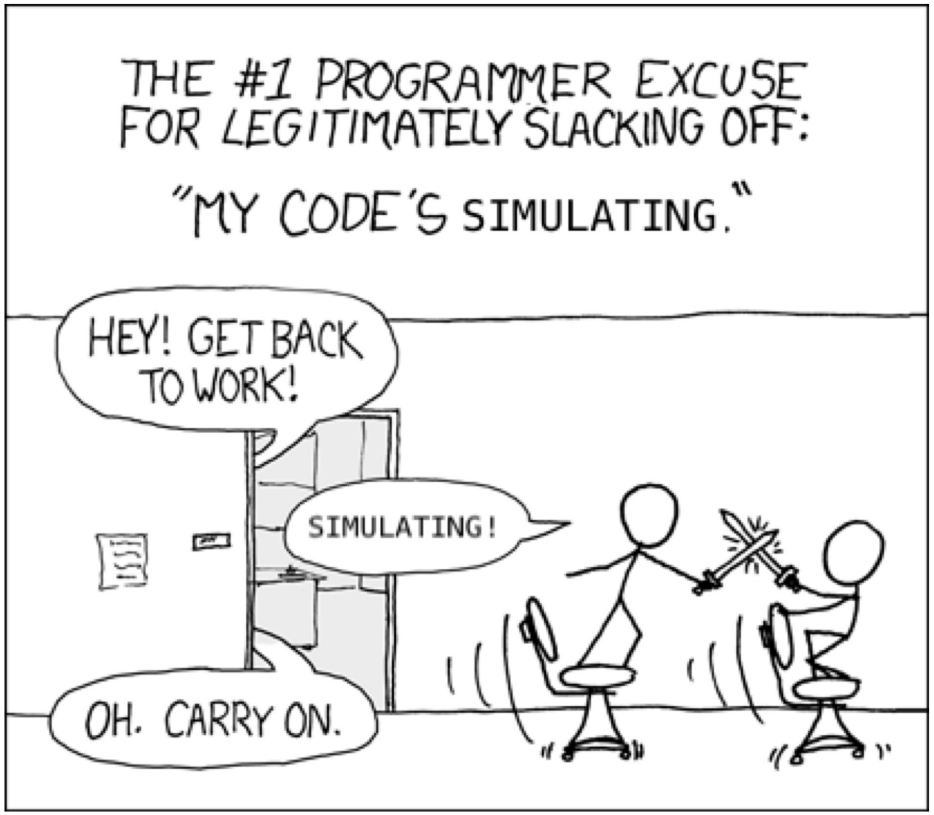


Design of a Codelet-based PXM with emphasis on memory orchestration

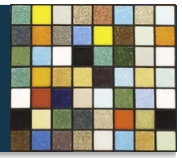
– Task and data orchestration of extremely heterogenous chipset-based systems.

– Orchestration of data movement and transformations (e.g., recoding) between tiles.

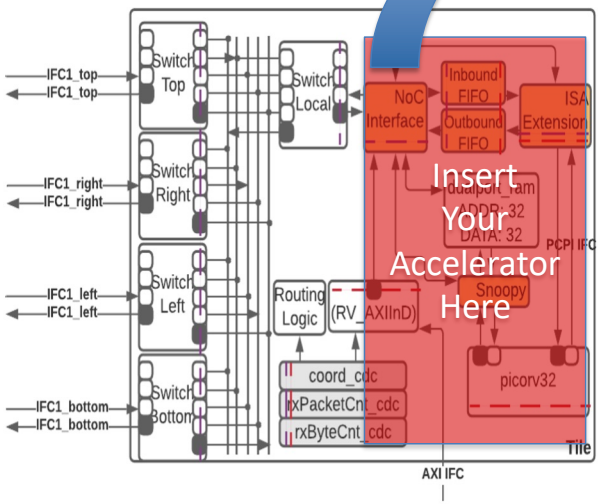
Hardware Simulation is Slow



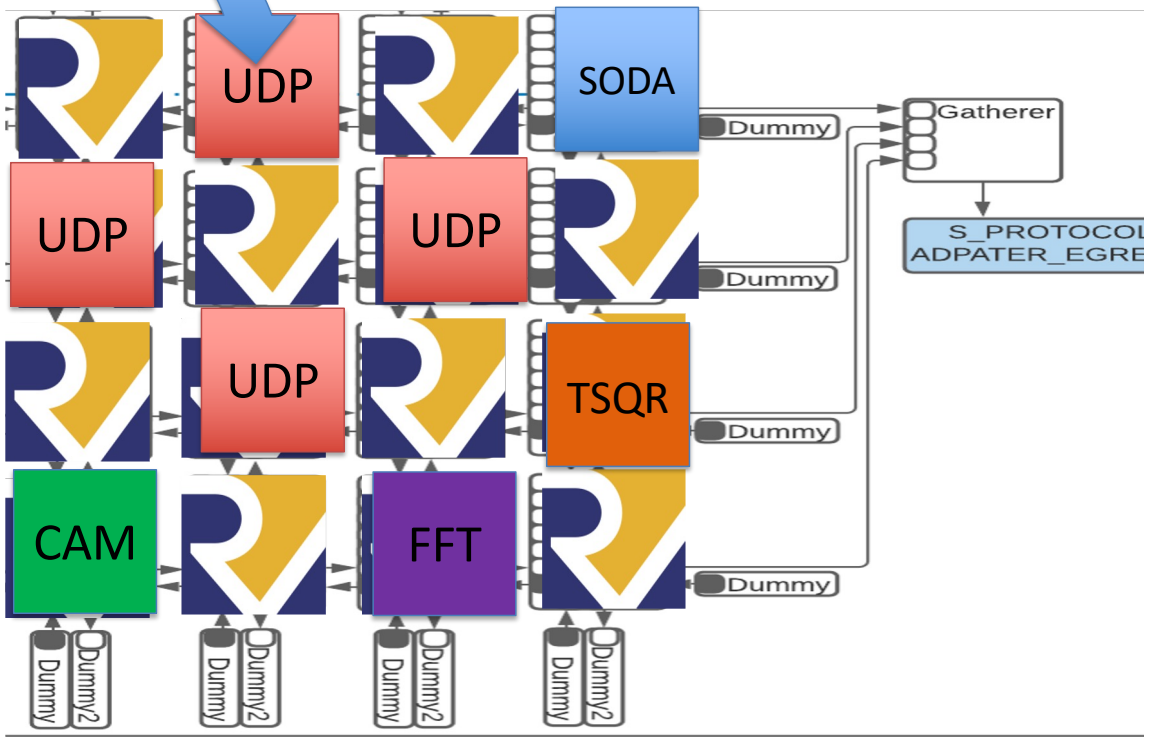
MoSAIC: Modular System for Accelerator Integration and Communication
Cross-USG Heterogeneous Integration Fabric



Tile

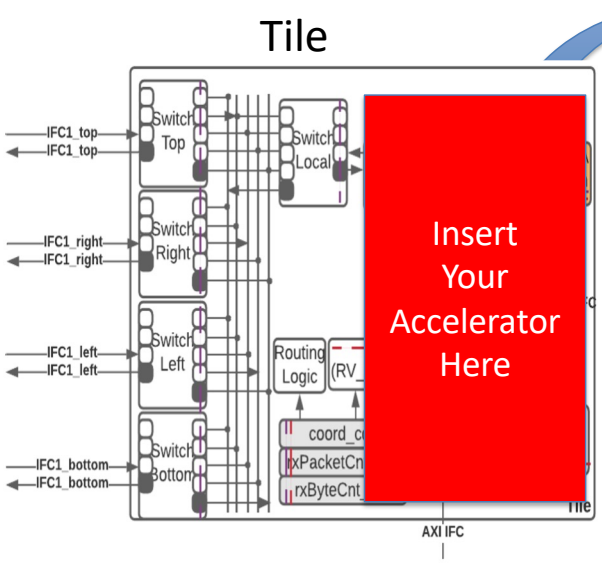


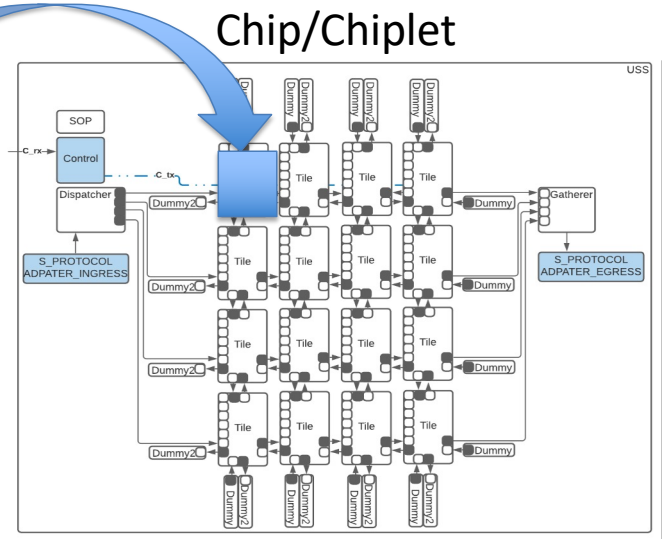
System on Chip

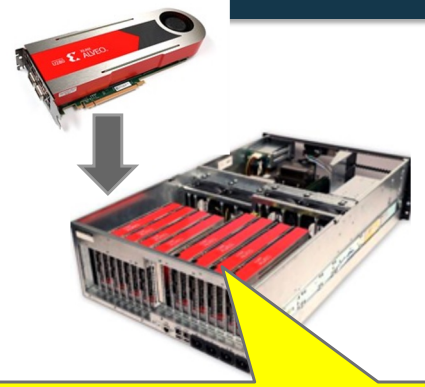


Lean and Mean
Operates at 250MHz (1/4 real-time)
Driverless inter-accelerator interaction
PGAS + MsgQs for communication
C++20 software stack

Customizable/Modular Open Architecture for FPGA







ESNet/Xilinx/CAG Emulation

- 256 RISC-V cores/card
- 8k RISC-V cores/cluster
- 1x HBM Memory/card
- 2x100Gigabit/card
- 8xCards/shelf
- 48 Shelfs in system

PGAS and MsgQ ISA Extensions for RISC-V ISA

	31 30 29 28 27 26 25	24 23 22 21 20	19 18 17 16 15	14 13 12	11 10 9 8 7	6 5 4 3 2 1 0
	funct 7	rs2	rs1	Funct 3	rd	Op Code
QPUT	NA	source register ID	destination QID	0 0 0	NA	0 1 0 1 0 1 1
QPOLL	NA	NA	destination QID	0 0 1	conditionRegister	0 1 0 1 0 1 1
QGET	NA	NA	destination QID	0 1 0	statusRegister	0 1 0 1 0 1 1
QWAIT	NA	NA	destination QID	0 1 1	destinationRegister	0 1 0 1 0 1 1
MGET	NA	Address (Dest)	Address (Source)	1 1 1	NA	0 1 0 1 0 1 1
MPUT	NA	Source	Address	1 1 0	NA	0 1 0 1 0 1 1

MsgQ ISA Extensions

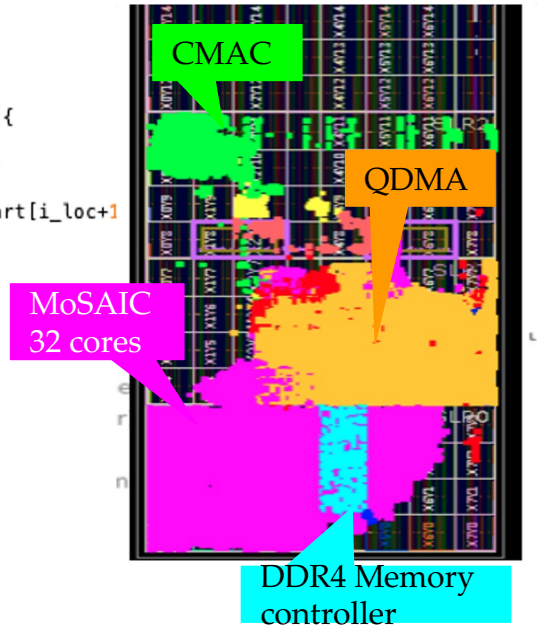
PGAS ISA Extensions

Open-Nic-Shell and C/C++ Software support

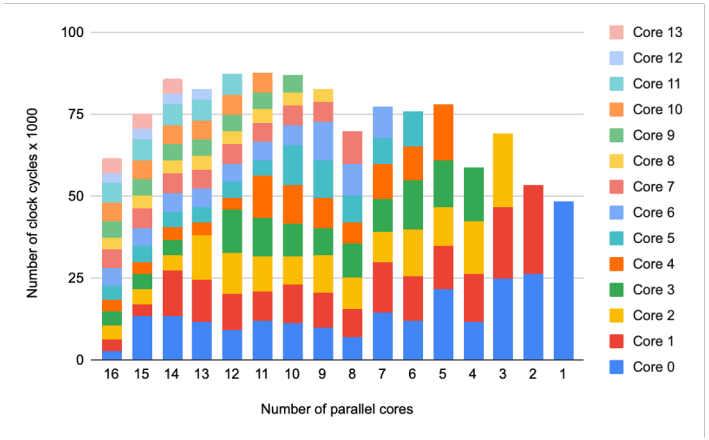
Non-Blocking *message queue instructions* similar to one-sided MPI functions: qPut(dest, data) ; qGet(src, data); qWait(); qPoll(); mPut(dest, data); mGet(src, data);

```
/****************************
 * Start Jacobi iterations *
 ****************************/
for (int iter = 0; iter < num_iters; iter++){
   for (int i_loc = 0; i_loc < n_loc; i_loc++){
      int i = loc_to_glob_row(i_loc, tid);
      int z = r_loc[i_loc];// / A.diag[i_loc];
      x_loc[i_loc] += z;
      for (int jj = A_start[i_loc]; jj < A_start[i_loc+1
         int j = A_j[jj];
         int y = A_data[jj] * z;
         if (i == j){
            r_loc[i_loc] -= y;
         }
         else {
            MsgQ_Put(j, y); // qPut()
            num_qPuts++;
         }
      }
```

Open-Nic-Shell provides support for PCIE and 100Gpbs Ethernet ports.



Clock cycles per core for the asynchronous triangular solver



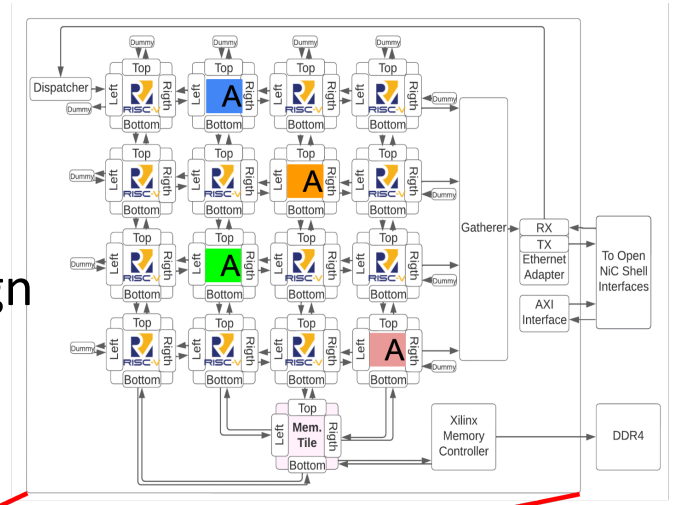
RegIO goal: host-side accessibility to the FPGA design

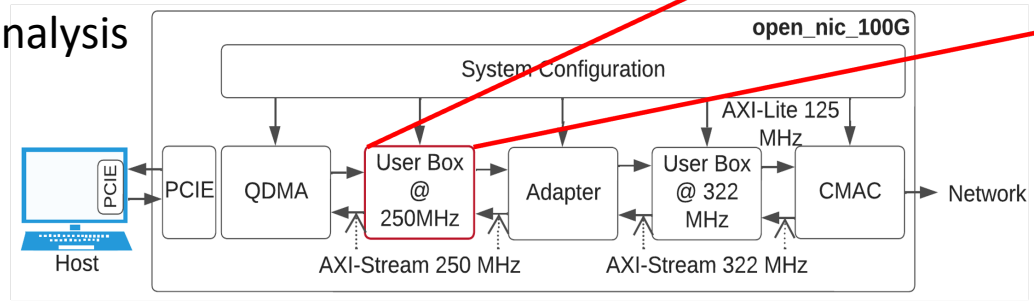
Baremetal-capable cores loaded on your FPGA

but what about control/monitoring?

RegIO provides visibility into your HW design

- **Host side access** to register space in the FPGA
  - initialization/configuration
  - performance analysis
  - debugging



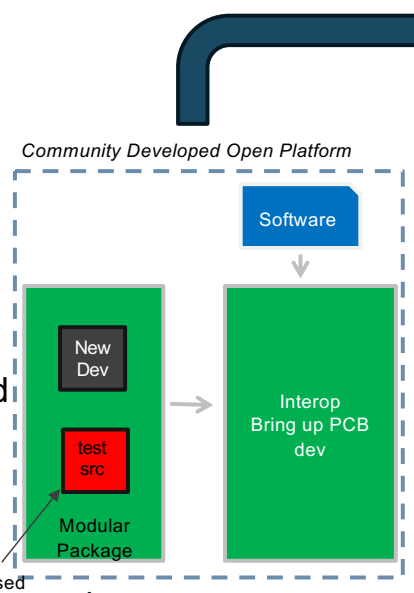


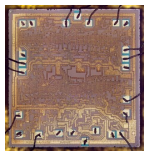
Integration with ChipYard/FireSim

- **Chipyard is Berkeley's SOC generator platform**
  - NoC Generator (many topologies and bit-widths)
  - Many different cores including 64-bit, OOO, and lightweight cores
  - Has all of the extra bits needed to boot Linux
  - Backed by 10 years of tape-outs
- **FireSim**
  - FPGA based simulation platform for ChipYard designs
  - GoldenGate inserts the delays to enable going from functional to cycle-accurate simulation
- **MoSAIC / FireSim Integration**
  - PGAS and MsgQ extensions implemented in RISC-V ROCC Interface
  - Modified Constellation NOC Generator
  - Enabled us to port MoSAIC Designs over to ChipYard and FireSim

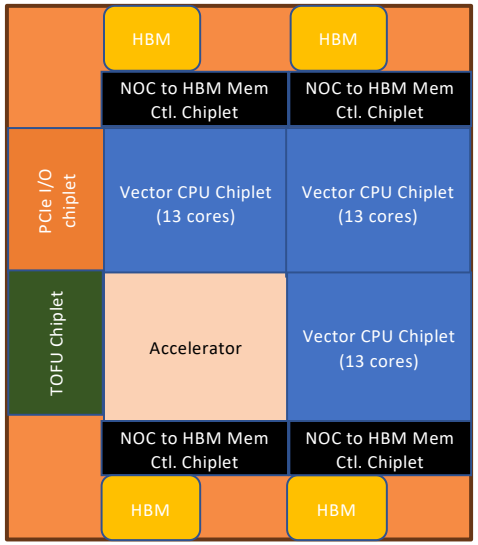
More Efficient Chiplet Development and Integration Path

Common Open Chiplet Eval Board for Bring-up

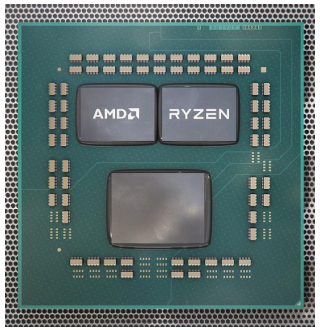




Chiplet



Prototype on Modular Architecture for function and performance integration



Commercial product integration

Chiplets at LBL

Home Register Demos Meeting Venue

http://chiplets.lbl.gov

LBNL/OCP Open Chiplet Economy Experience Center

Hosted by Lawrence Berkeley National Laboratory (LBNL)

Co-organized by the Open Compute Project (OCP)

Date: June 24, 2024

Time: 12:00pm to 5:00pm

Location: Berkeley National Lab, Wang Hall Bldg. 59, Room 59-3101

OPEN Compute Project®

2024 JEDEC and OCP Standards for Chiplet Design with 3DIC Packaging Workshop

JEDEC

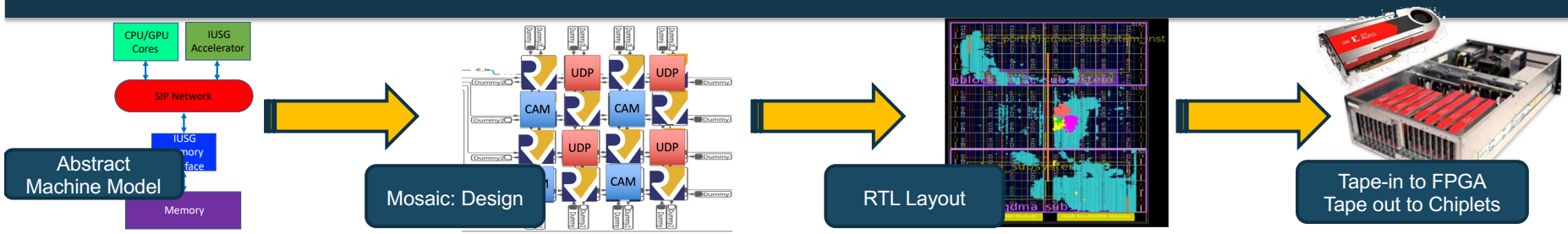
Final Thoughts: How to engage

MoSAIC Github Repository

OPEN Compute Project®

Launching new workstream for Modular AI and HPC. Join us!

P38 Prototyping Framework



- **Overall Prototyping Framework with MoSAIC integration platform takes us from Abstract Machine Model to functional model on FPGA running a full software stack**
  - MoSAIC is up-and-running on FPGA platform (192 RISC-V cores per FPGA + accelerator tiles co-integrated)
  - RTL to FPGA for accelerated hardware testing & software development
  - Synthesizable RTL tape-in for FY24 (setup for tape-out to chiplets)
- **GitHub Repository: https://github.com/PatriGonzalez/P38_Mosaic**
- **Tutorial: https://docs.google.com/document/d/1ZjCIGr6vOfTrv2l9z723LdF-E3mwk1DgInhbyacd5SA/edit**

How to Engage the Hyperscalers and Advanced Packaging Community

- **Open Compute Project : Hyperscalers Interoperable Standards**
  - Open Domain Specific Architectures & Open Chiplets Economy TWG
  - Modular AI/HPC Workstream (AMD, LBL, + many others)
  - Modularity is central to OCP's existence

- **SEMI APHI : Advanced Packaging & Hetro Integration**
  - Direct engagement with the foundries
  - OSATs (Outsourced Semiconductor Assembly and Test)
  - (see Samsung 2.5d pkg. service): They are an OEM, but also an OSAT
- **Heterogeneous Integration Roadmap: IEEE/EPS + Industry**
  - Everything you ever wanted to know about AdvPkg & use cases
- **CHIPS NAPMP: National Advanced Packaging and Mfr. Program**
  - Subu Ayar (formerly UCLA) and Bapi Vinnakota (formerly LBNL)

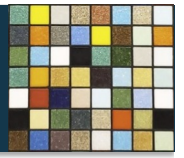
Final Thoughts

- **Conventional Wisdom: In the era of the "universal computer," *scale was the correct answer to deliver value to HPC's scientific customers.***
- ***New Wisdom: In this post-Moore/post-Exascale era, scale alone is not a viable approach to continuing to deliver value to the scientific community!***
  - It isn't system scale, system effectiveness for the workload targets (e.g. refocus on strong scaling)
  - This can be delivered by specialization, and the ***workflows OUTSIDE of the HPC system***

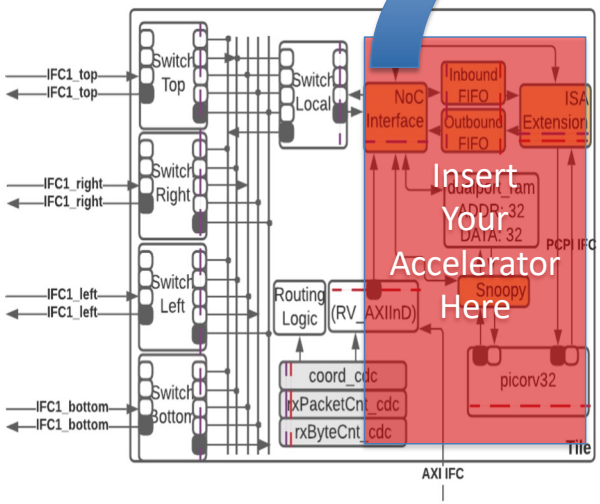
- **Conventional Wisdom: Buy off the shelf microprocessors**
- **New (old) Wisdom: Follow the money (e.g. follow what hyperscale is doing)**
  - *Join-em*: Buy AI machines and port everything to AI
  - *Beat-em* (really join-em): Work together with the hyperscalers to address capital cost challenge through ***modularity*** and ***specialization*** **(its not chip cost… platform cost!!!)**
  - *HPC punches above its weight because it engages in pre-competitive R&D with industry partners*

Start Talking about MoSAIC Platform

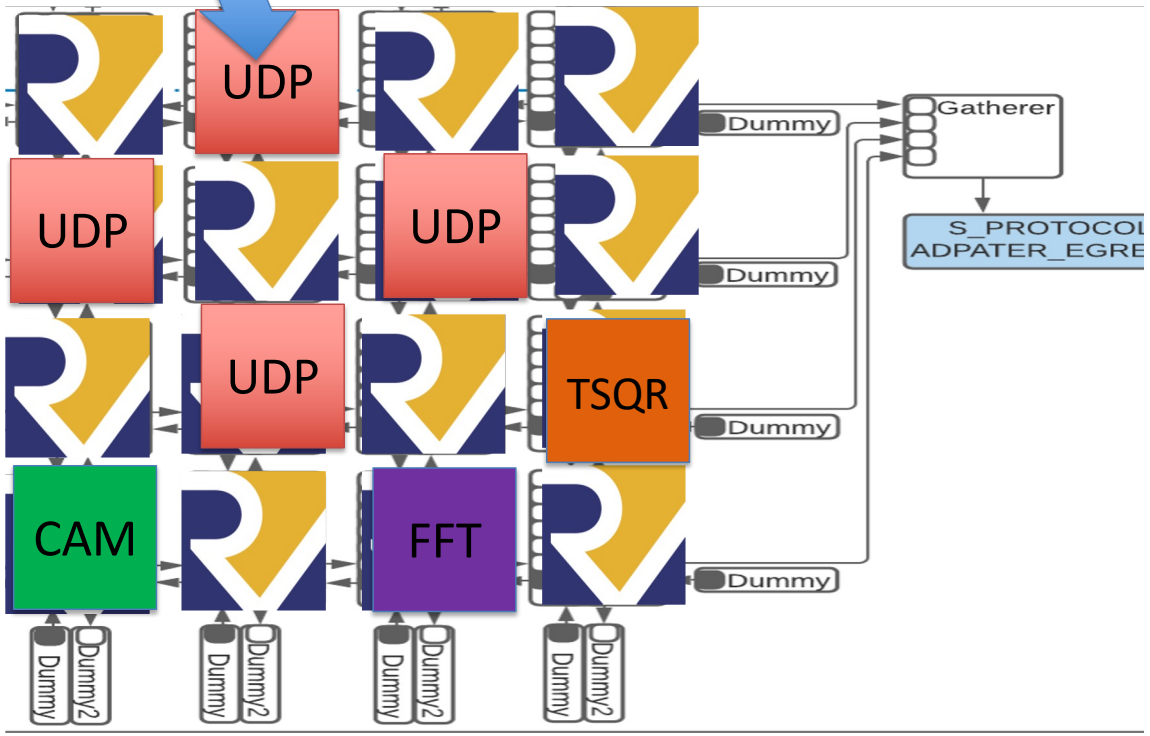
MoSAIC: Modular System for Accelerator Integration and Communication
Cross-USG Heterogeneous Integration Fabric



Tile



System on (FPGA) Chip

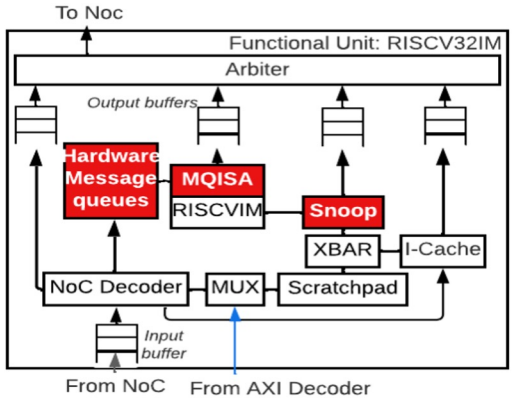


BERKELEY LAB

https://github.com/PatriGonzalez/P38_Mosaic/blob/main/mosaic_4k/doc/MoSAICTutJun2023.pdf

RISC-V ISA extensions for message driven and PGAS computing

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	funct 7							**rs2**					**rs1**					**Funct 3**			**rd**					**Op Code**						
QPUT	NA							source register ID					destination QID					0	0	0	NA					0	1	0	1	0	1	1
QPOLL	NA							NA					destination QID					0	0	1	conditionRegister					0	1	0	1	0	1	1
QGET	NA							NA					destination QID					0	1	0	statusRegister					0	1	0	1	0	1	1
QWAIT	NA							NA					destination QID					0	1	1	destinationRegister					0	1	0	1	0	1	1
MGET	NA							Address (Dest)					Address (Source)					1	1	1	NA					0	1	0	1	0	1	1
MPUT	NA							Source					Address					1	1	0	NA					0	1	0	1	0	1	1



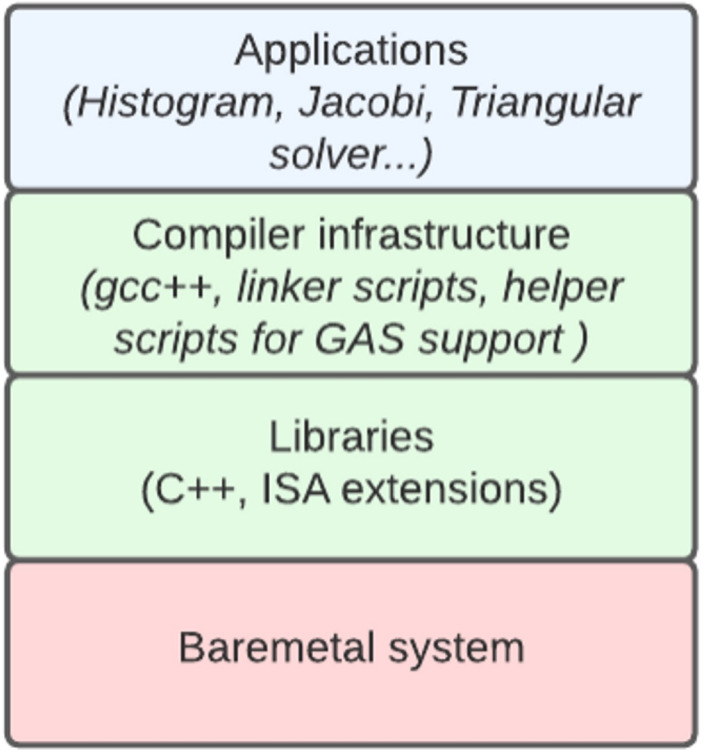
Message Queues

- *QPUT (destinationQID,sourceregisterID)*
- *QGET (destinationQID,destinationregisterID)*
- *QWAIT (destinationQID,statusregisterID)*
- *QPOLL (destinationQID, branchTarget)*

Non-Blocking 3rd party memory instructions

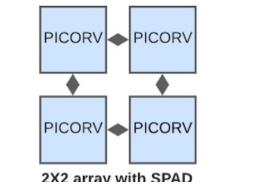
- *MPUT (Data, AddressDest)*
- *MGET (AddresSource, AddressDestination)*

MoSAIC supports C/C++ software stack (C++20)

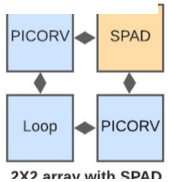



```
/****************************
 * Start Jacobi iterations *
 ****************************/
for (int iter = 0; iter < num_iters; iter++){
   for (int i_loc = 0; i_loc < n_loc; i_loc++){
      int i = loc_to_glob_row(i_loc, tid);
      int z = r_loc[i_loc];// / A.diag[i_loc];
      x_loc[i_loc] += z;
      for (int jj = A_start[i_loc]; jj < A_start[i_loc+1]; jj++){
         int j = A_j[jj];
         int y = A_data[jj] * z;
         if (i == j){
            r_loc[i_loc] -= y;
         }
         else {
            MsgQ_Put(j, y); // qPut()
            num_qPuts++;
         }
      }
```

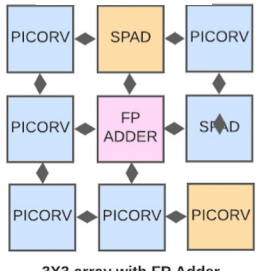
Examples of Scalable Tiled Arrays on a U250 FPGA



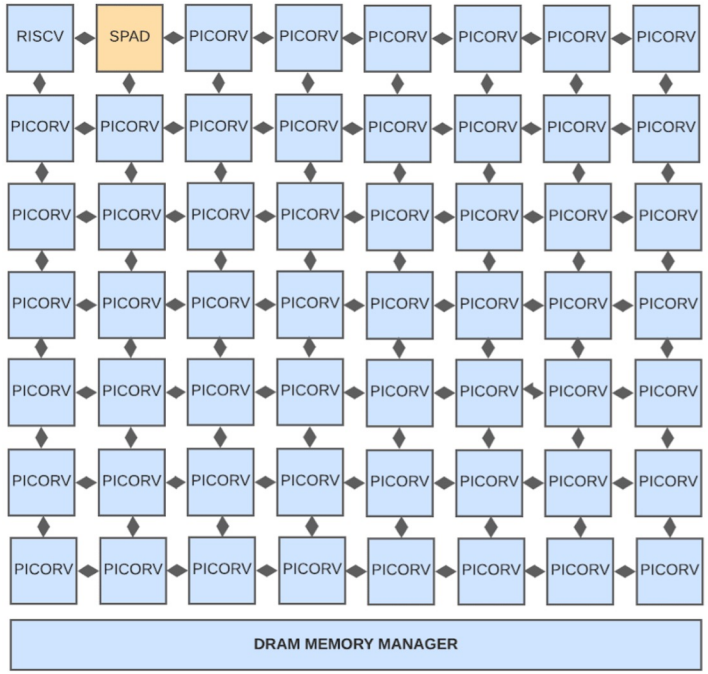
2x2 Array



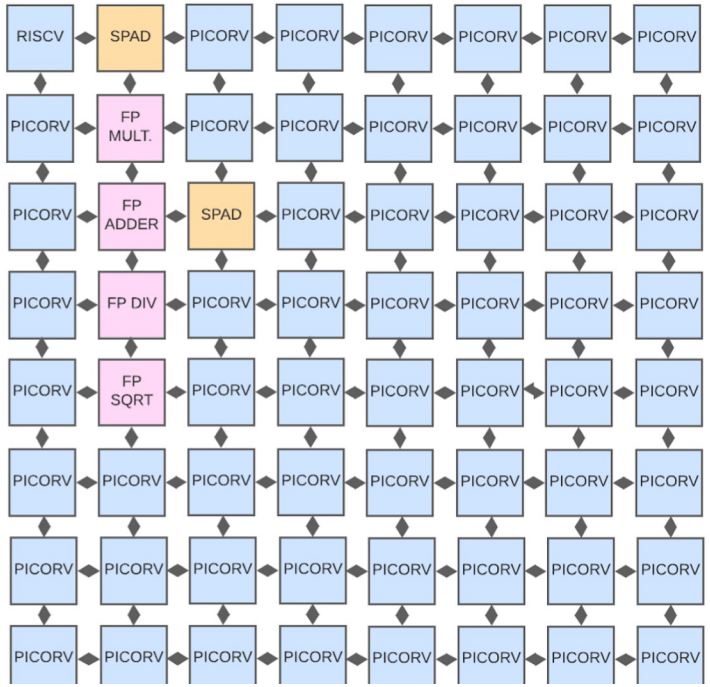
2x2 Array



3x3 Array

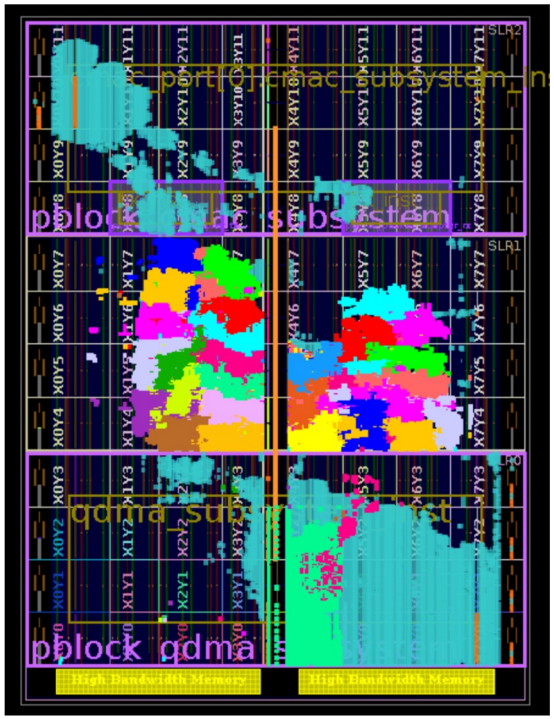


7 x 8 + DRAM

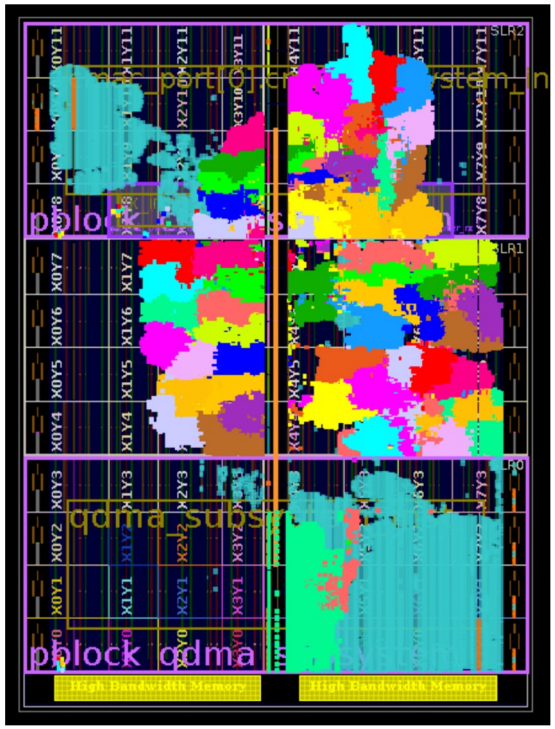


8 x 8 including FP adder, multiplier, divider, and square root

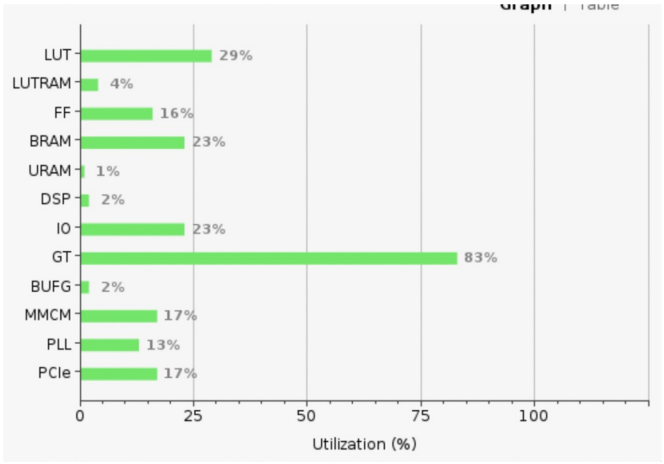
Examples of mapping these arrays to an FPGAs can fit up to 192 RISC-V tiles per U250 FPGA



7 x 4 + DRAM

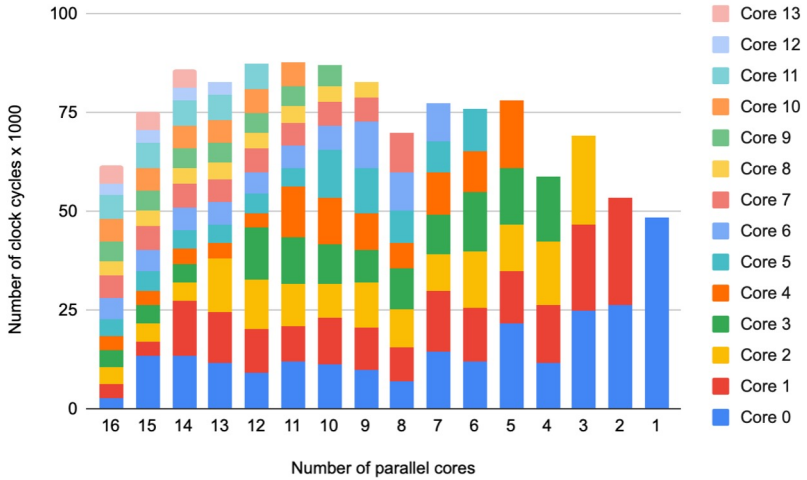


7 x 8 + DRAM



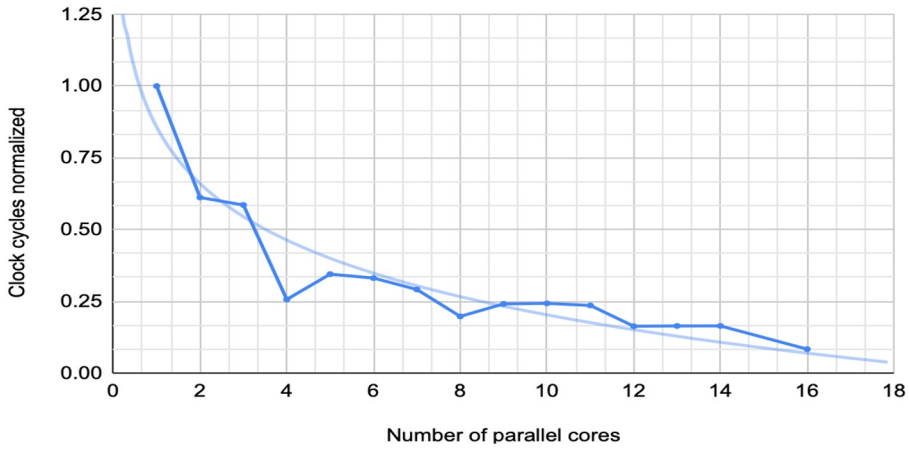
Examples of exploration for tri-solve
only 4x slower than real-time (operating at 250MHz)

Computing time per core



X axis: Number of parallel cores
Y axis: Number of clock cycles

Total computing time



X axis: Number of parallel cores
Y axis: Number of clock cycles normalized

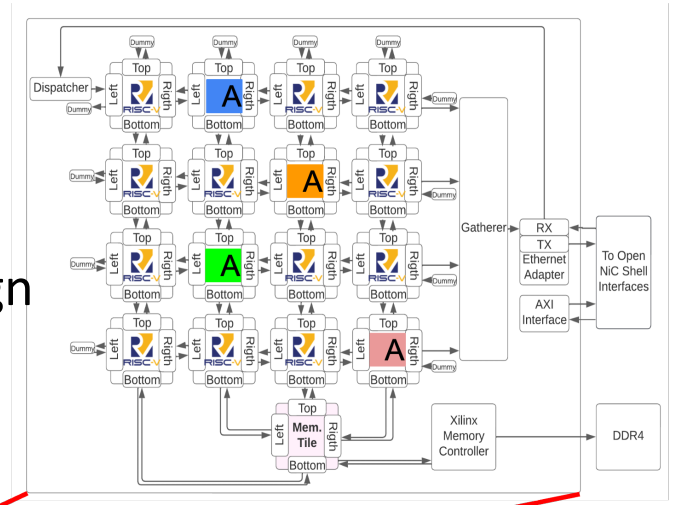
RegIO goal: host-side accessibility to the FPGA design

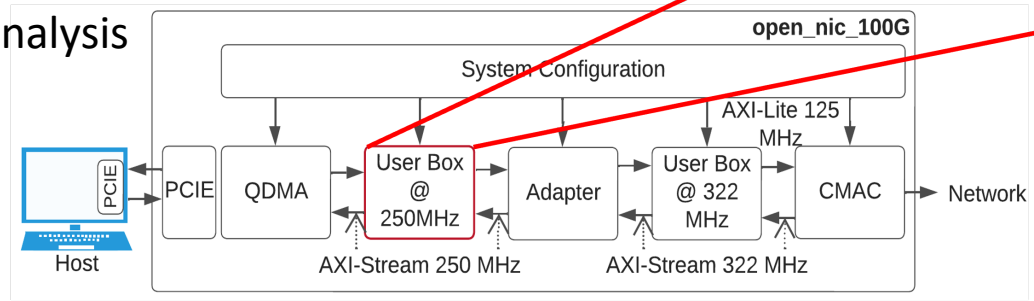
Baremetal-capable cores loaded on your FPGA

but what about control/monitoring?

RegIO provides visibility into your HW design

- **Host side access** to register space in the FPGA
  - initialization/configuration
  - performance analysis
  - debugging





Integration with ChipYard/FireSim

- **Chipyard is Berkeley's SOC generator platform**
  - NoC Generator (many topologies and bit-widths)
  - Many different cores including 64-bit, OOO, and lightweight cores
  - Has all of the extra bits needed to boot Linux
  - It's a heavy lift to pick it up, but its backed by 10 years of tape-outs
- **FireSim**
  - FPGA based simulation platform for ChipYard designs
  - GoldenGate inserts the delays to enable going from functional to cycle-accurate simulation
- **MoSAIC / FireSim Integration**
  - PGAS and MsgQ extensions implemented in RISC-V ROCC Interface
  - Modified Constellation NOC Generator
  - Enabled us to port MoSAIC Designs over to ChipYard and FireSim

Technology Insertion into Mainstream Platforms

AMD, Intel, Arm offer integration path for 3rd party accelerator "chiplets"

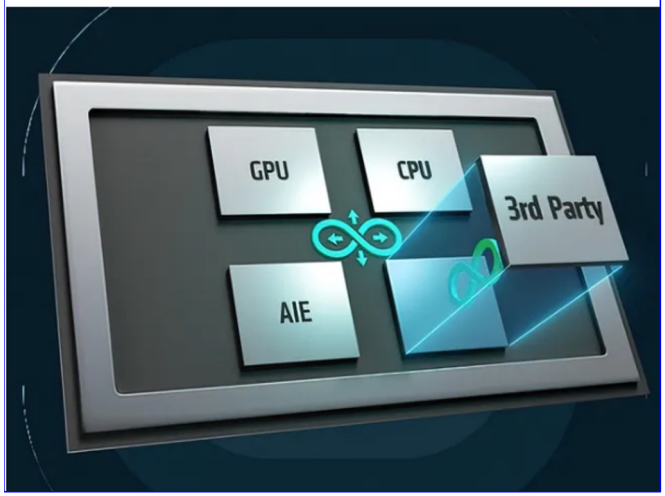
Modular AMD Chips to Embrace Custom 3rd Party Chiplets

News By Francisco Pires last updated June 20, 2022

Supercharging learnings - and earnings - from the console space.

Comments (2)

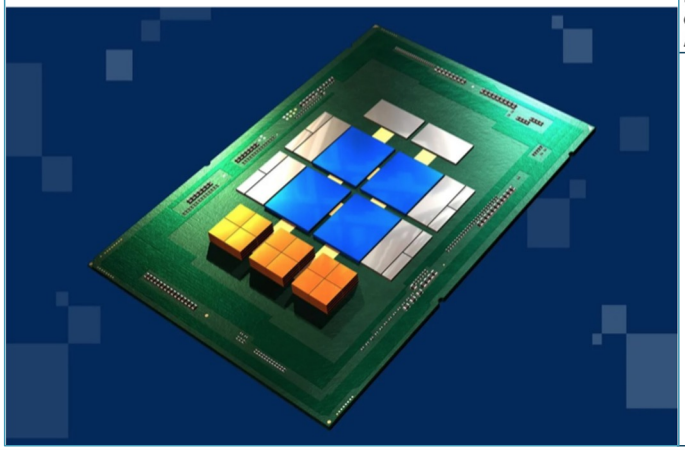
When you purchase through links on our site, we may earn an affiliate commission. Here's how it works.



To 'Meteor Lake' and Beyond: How Intel Plans a New Era of 'Chiplet'-Based CPUs

At the Hot Chips 2022 conference, Intel teased its upcoming 'Meteor Lake' and 'Arrow Lake' processor families, which will use multiple tiny tiles fused together in an attempt to break free of the limits of monolithic chip design. Here's why little tiles are a big deal.

By Michael Justin Allen Sexton August 24, 2022

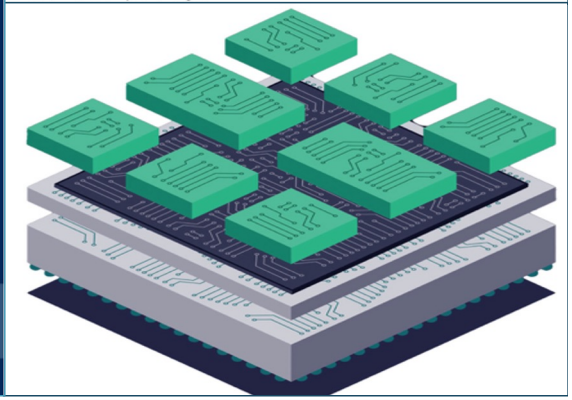


ARM Opens Door to Make Custom Chips for HPC, AI

By Doug Eadline

October 19, 2023

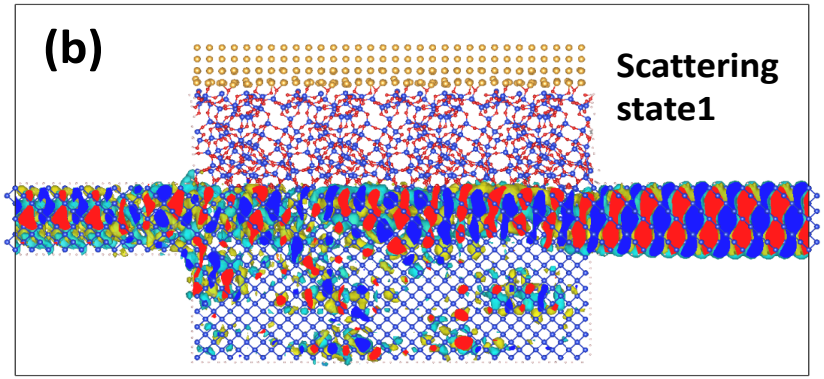
It is safe to say that ARM isn't a scrappy startup that was once the pride of the UK. The US-based IPO made the chip designer a big-game chip player, and the new capital is kickstarting some major initiatives to find more customers for its products. A new effort called Total Design aims at making it easier for companies looking to design chips in-house, an idea gaining ground with the AI boom and chip shortages.

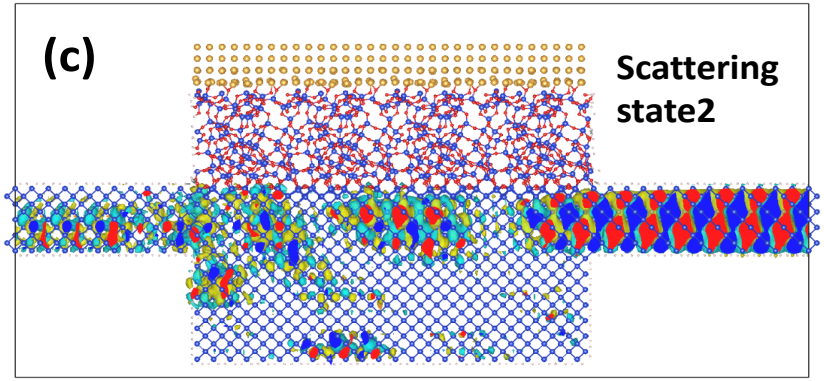


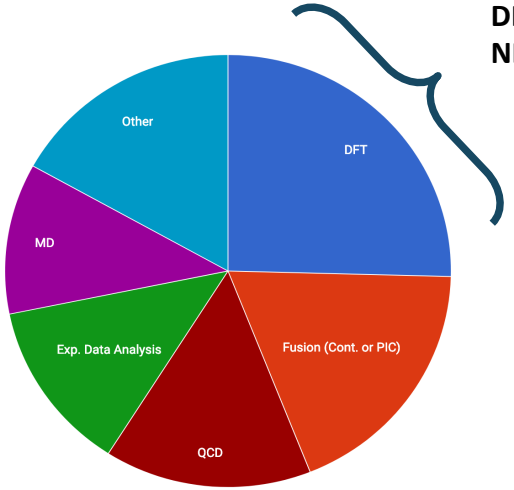
BERKELEY LAB

http://chiplets.lbl.gov/

The Importance for High Performance DFT





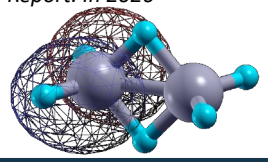


DFT: 25% of NERSC workload

B. Driscoll and Z. Zhao. 2020. Automation of NERSC Application Usage Report. In 2020 IEEE/ACM International Workshop on HPC User Support Tools (HUST)

binit

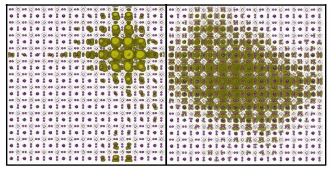
Qbox
First-Principles Molecular Dynamics

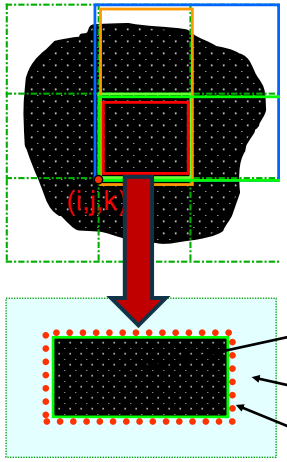


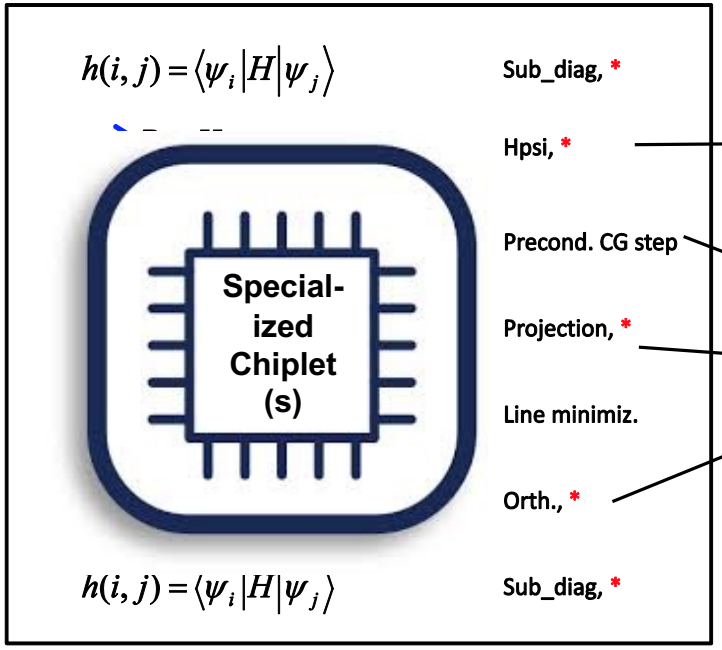
BERKELEY LAB

The DFT kernel for each fragment

Communication Avoiding LS3DF Formulation – Scales O(N)







$O(N^2 Log(N))$

Comm bound if non-local

3D parallel FFT

TSQR & Choelesky

ZGEMM

$O(N^3)$

Compute-bound

LS3DF O(N) Algorithm Formulation
Minimizes off-chip Communication

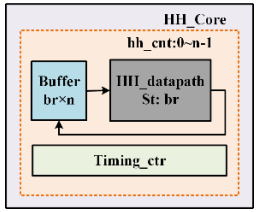
One patch per Chiplet
400 bands/patch

Compute Intensive Kernels
Targeted for HW Specializ

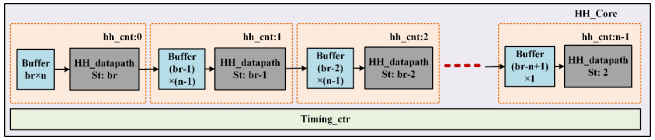
BERKELEY LAB

Building Flexible Accelerators

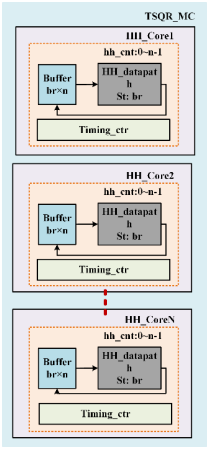
Parameterized Hardware Generation with CHISEL



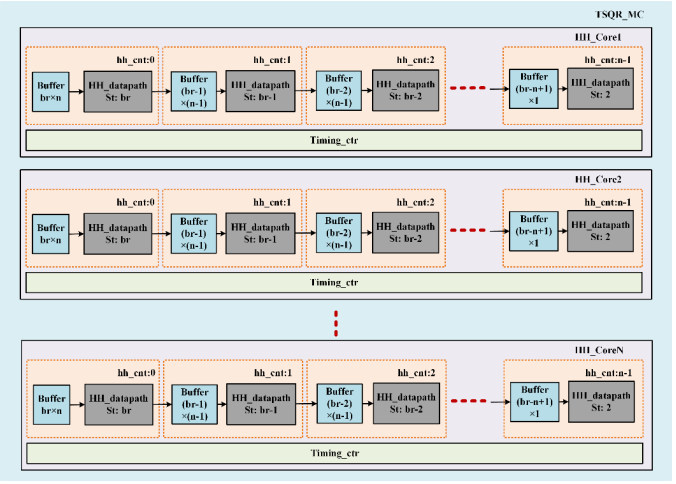
A Single HH-core



Deeply Pipelined HH-Core



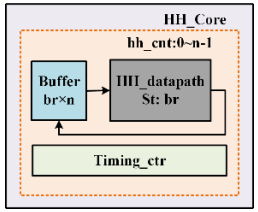
3-Way Parallel HH-core



Or Generate any level of concurrency or pipelining

Building Flexible Accelerators

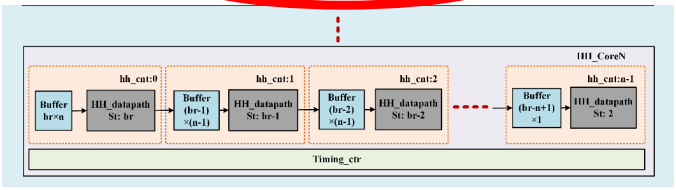
Parameterized Hardware Generation with CHISEL



A Single HH-core

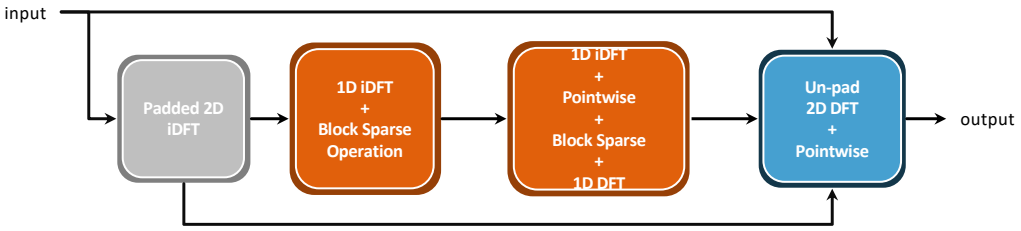
Table 1.3 Resource Cost Analysis

TSQR Designs		FP Operators			Register file (Words)		
Multi-core	Streaming Width	Iterative	Pipeline	Parallel (16 cores)	Iterative	Pipeline	Parallel (16 cores)
TSQR Designs	16	68	432	1,088	192	432	3,072
	32	132	1,632	2,112	768	3672	12,288
	64	260	6,336	4,160	3,072	28,336	49,152
	128	516	24,960	8,256	12,288	222,560	196,608
	256	1,028	99,072	16,448	49,152	1,764,032	786,432



Generate any level of concurrency or pipelining

Cross Domain Optimizations Using a Uniform Tensor Notation



Input:

$\Psi^{(b)(v_0 v_1 v_2)}$ - input planewaves

$\nabla^{(v_0 v_1 v_2)}$ - kinetic energy

$\Phi^{(Ir)(\mu_0\mu_1\mu_2)}$ - non - local projectors

$V^{(\mu_0\mu_1\mu_2)}$ - ionic potential

Output:

$\Psi^{(b)(v_0 v_1 v_2)}$ - output planewaves

begin

$T_0^{(b)(\mu_0'\mu_1'\mu_2')} = \Psi^{(b)(v_0 v_1 v_2)} \cdot P^{(v_0 v_1 v_2)(\mu_0'\mu_1'\mu_2')}$

$T_1^{(b)(\mu_0\mu_1\mu_2)} = T_0^{(b)(\mu_0'\mu_1'\mu_2')} \cdot iDFT_{3D}^{(\mu_0'\mu_1'\mu_2')(\mu_0\mu_1\mu_2)}$

$T_2^{(b)(Ir)} = T_1^{(b)(\mu_0\mu_1\mu_2)} \cdot \left(\Phi^{(Ir)(\mu_0\mu_1\mu_2)}\right)^T$

$T_1^{(b)(\mu_0\mu_1\mu_2)} = T_1^{(b)(\mu_0\mu_1\mu_2)} \odot V^{(\mu_0\mu_1\mu_2)}$

$T_1^{(b)(\mu_0\mu_1\mu_2)} += T_2^{(b)(Ir)} \cdot \Phi^{(Ir)(\mu_0\mu_1\mu_2)}$

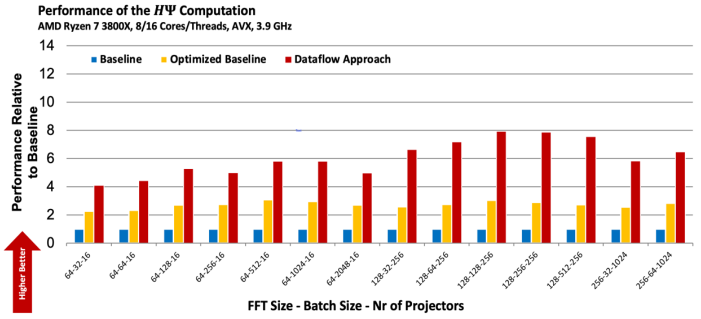
$T_0^{(b)(\mu_0'\mu_1'\mu_2')} = T_1^{(b)(\mu_0\mu_1\mu_2)} \cdot DFT_{3D}^{(\mu_0\mu_1\mu_2)(\mu_0'\mu_1'\mu_2')}$

$\Psi^{(b)(v_0 v_1 v_2)} = \Psi^{(b)(v_0 v_1 v_2)} \odot \nabla^{(v_0 v_1 v_2)}$

$\Psi^{(b)(v_0 v_1 v_2)} += T_0^{(b)(\mu_0'\mu_1'\mu_2')} \cdot U^{(\mu_0'\mu_1'\mu_2')(v_0 v_1 v_2)}$

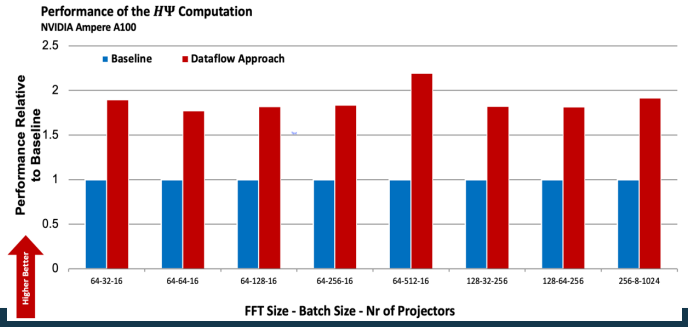
end

Pipelining Computation for Commodity Hardware



CPU implementation

GPU implementation



"A Systematic Approach to Improving Data Locality Across Fourier Transforms and Linear Algebra Operations", Doru Thom Popovici, Andrew Canning, Zhengji Zhao, Lin-Wang Wang, John Shalf, ICS 2021

Materials Science Accelerator

Preliminary Performance for Materials Science HΨ

Eigenvalue Problem

$$h(i,j) = \langle\psi_i|H|\psi_j\rangle$$

Hpsi

$$P_i = H\psi_i - \epsilon_i\psi_i$$

Projection

$$P_i = A\left(P_i - \frac{\lambda_i}{\lambda_i^0}P_i^0\right)$$

$$P_i = P_i - \sum_{j=1,i}\langle P_i|\psi_j\rangle$$

nline

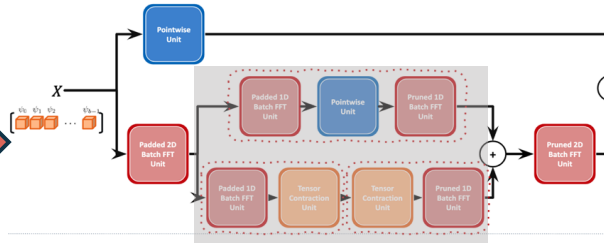
Orthogonalization

$$\psi_i = \psi_i\cos\theta_i + P_i\sin\theta_i$$

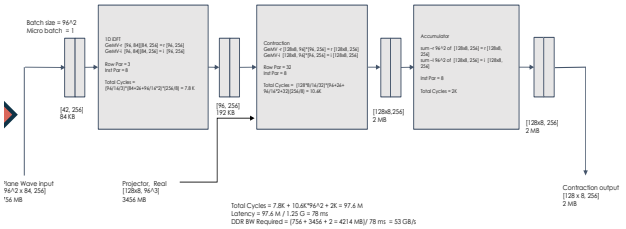
$$\psi_i = \psi_i - \sum_{j<i}\langle\psi_i|\psi_j\rangle$$

$$h(i,j) = \langle\psi_i|H|\psi_j\rangle$$

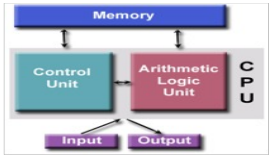
Dataflow Algorithm Reformulation



Mapping onto Custom Hardware

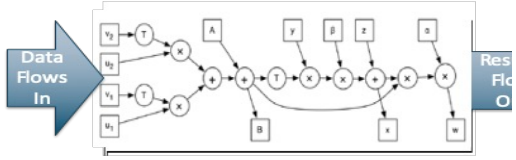


Von Neumann CPUor GPU




```
int main()
{
  int n = 0;
  while(n < 100)
  {
    n = n + 5;
    print("n = %d\n", n);
    pause(200);
    if(n == 50) break;
  }
  print("All done!");
}
```

Dataflow (FPGA, GraphCore etc.)



$R_{[t=n+1]}(0,0,0) = 0$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= 2{*}R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= R_{[t=n-1]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n+1]}(+1,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= C * 2 * R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n]}(-1,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n+1]}(0,+1,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= C * 2 * R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n]}(0,-1,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n+1]}(0,0,+1)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= C * 2 * R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n]}(0,0,-1)$
Rotate Registers

Results

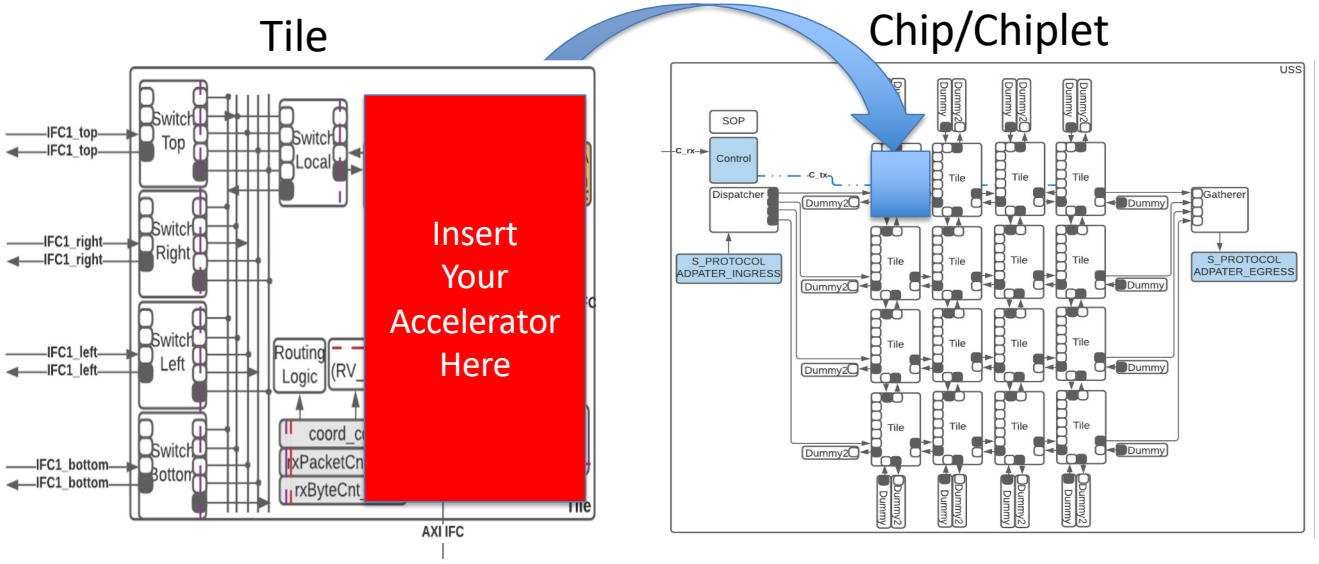
Platform	Time for Contraction	Speedup over CPU	Speedup over GPU
CPU (Haswell/Cori Phase 1) node	1.375	1	
GPU (NVIDIA 1080)	0.5	2.75	1
CGRA (Samba Nova) unoptimized	0.23	6	2.2
CGRA (Samba Nova) optimized	0.023	60	21.7

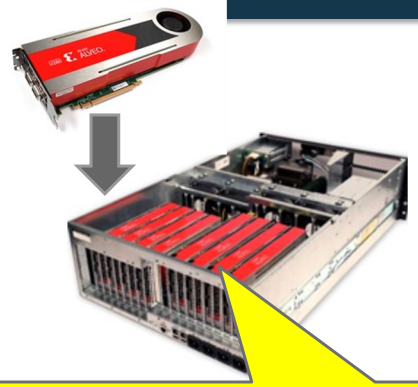
Delivered Speedups *(compared to optimized code)* of "custom" DFT

BERKELEY LAB

Thom Popovici, Andrew Canning (FFTx), Zhengji Zhang (NERSC)
Franz Francetti (CMU/FFTx)

Customizable/Modular Open Architecture for HPDA





ESNet/Xilinx/CAG Emulation

- 256 RISC-V cores/card
- 8k RISC-V cores/cluster
- 1x HBM Memory/card
- 2x100Gigabit/card
- 8xCards/shelf
- 48 Shelfs in system

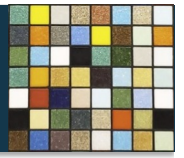
PGAS and MsgQ ISA Extensions for RISC-V ISA

	31 30 29 28 27 26 25	24 23 22 21 20	19 18 17 16 15	14 13 12	11 10 9 8 7	6 5 4 3 2 1 0
	funct 7	rs2	rs1	Funct 3	rd	Op Code
QPUT	NA	source register ID	destination QID	0 0 0	NA	0 1 0 1 0 1 1
QPOLL	NA	NA	destination QID	0 0 1	conditionRegister	0 1 0 1 0 1 1
QGET	NA	NA	destination QID	0 1 0	statusRegister	0 1 0 1 0 1 1
QWAIT	NA	NA	destination QID	0 1 1	destinationRegister	0 1 0 1 0 1 1
MGET	NA	Address (Dest)	Address (Source)	1 1 1	NA	0 1 0 1 0 1 1
MPUT	NA	Source	Address	1 1 0	NA	0 1 0 1 0 1 1

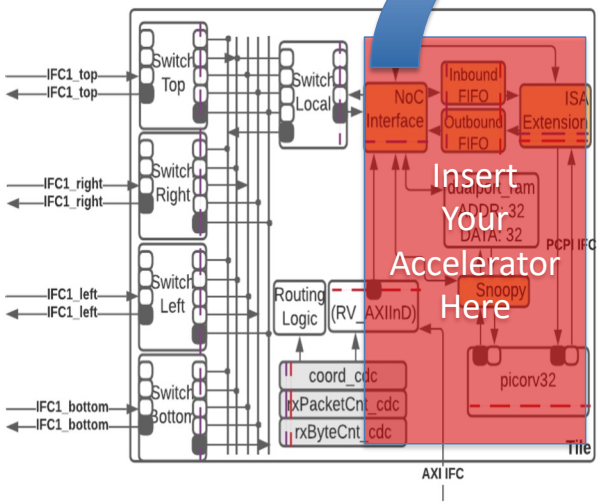
MsgQ ISA Extensions

PGAS ISA Extensions

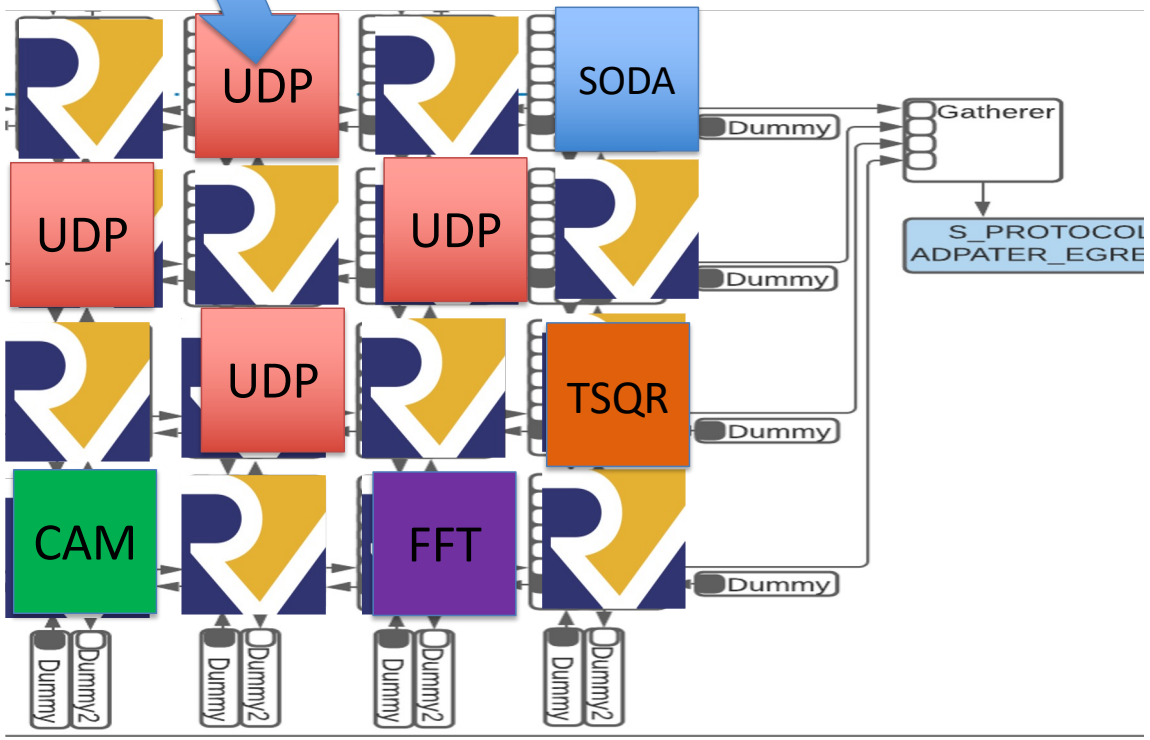
MoSAIC: Modular System for Accelerator Integration and Communication
Cross-USG Heterogeneous Integration Fabric



Tile



System on Chip

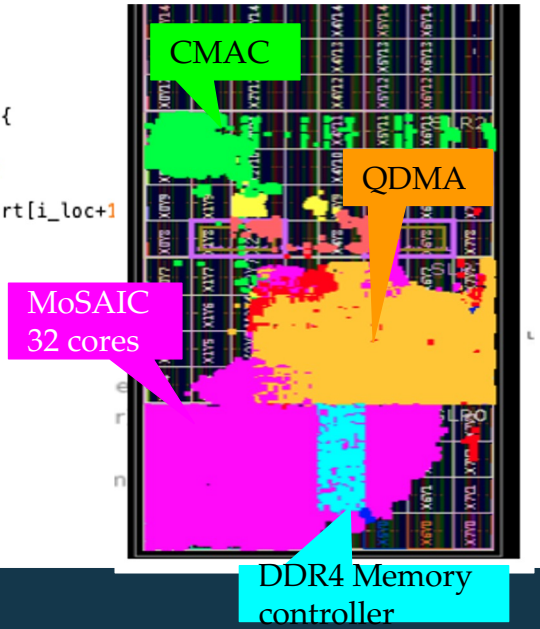


Open-Nic-Shell and C/C++ Software support

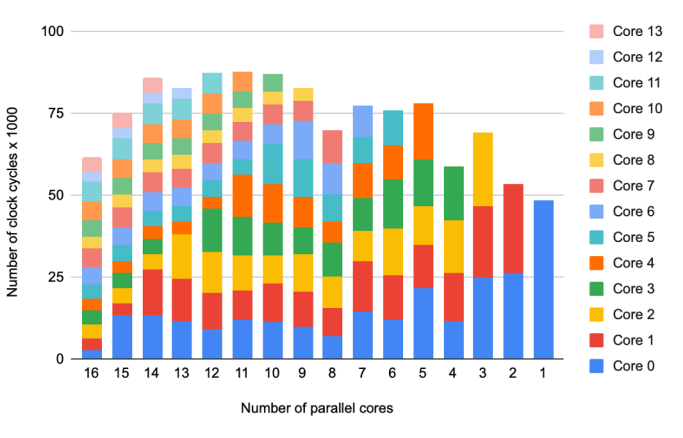
Non-Blocking *message queue instructions* similar to one-sided MPI functions: qPut(dest, data) ; qGet(src, data); qWait(); qPoll(); mPut(dest, data); mGet(src, data);

```
/*****************************
 * Start Jacobi iterations *
 *****************************/
for (int iter = 0; iter < num_iters; iter++){
   for (int i_loc = 0; i_loc < n_loc; i_loc++){
      int i = loc_to_glob_row(i_loc, tid);
      int z = r_loc[i_loc];// / A.diag[i_loc];
      x_loc[i_loc] += z;
      for (int jj = A_start[i_loc]; jj < A_start[i_loc+1
         int j = A_j[jj];
         int y = A_data[jj] * z;
         if (i == j){
            r_loc[i_loc] -= y;
         }
         else {
            MsgQ_Put(j, y); // qPut()
            num_qPuts++;
         }
      }
```

Open-Nic-Shell provides support for PCIE and 100Gpbs Ethernet ports.



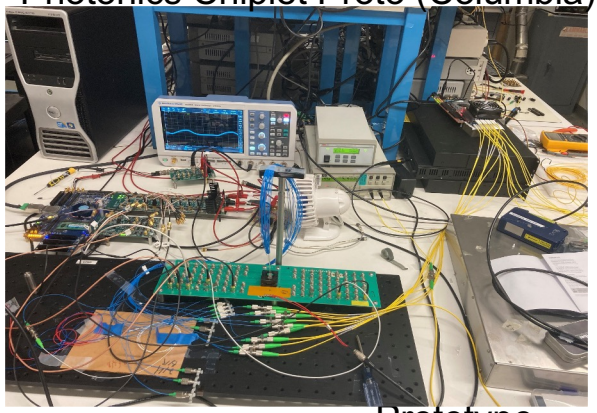
Clock cycles per core for the asynchronous triangular solver



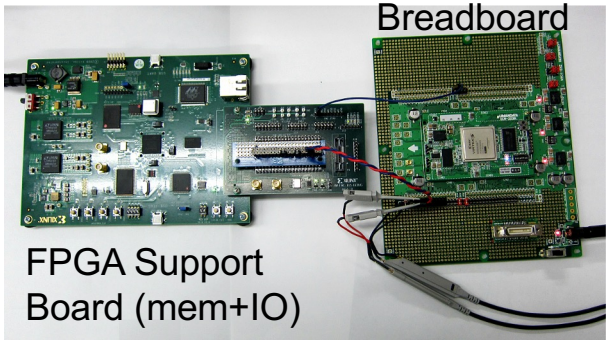
Limits of Current Practice

- **Typical MPW Prototype Chip Challenges**
  - Big chips have huge challenges
    - Complex clock trees
    - Power distribution and power sags
    - Yield / Known good die
    - Cost per area
  - MPW typically results in small chips
    - Higher yield (known good die)
    - Low bandwidth and low performance due to limited shoreline BW
    - Limited area available for peripheral support functions
  - ***Typically require FPGA support platform for memory control & PCIe***
- **There is a better way with chiplets**

Photonics Chiplet Proto (Columbia)

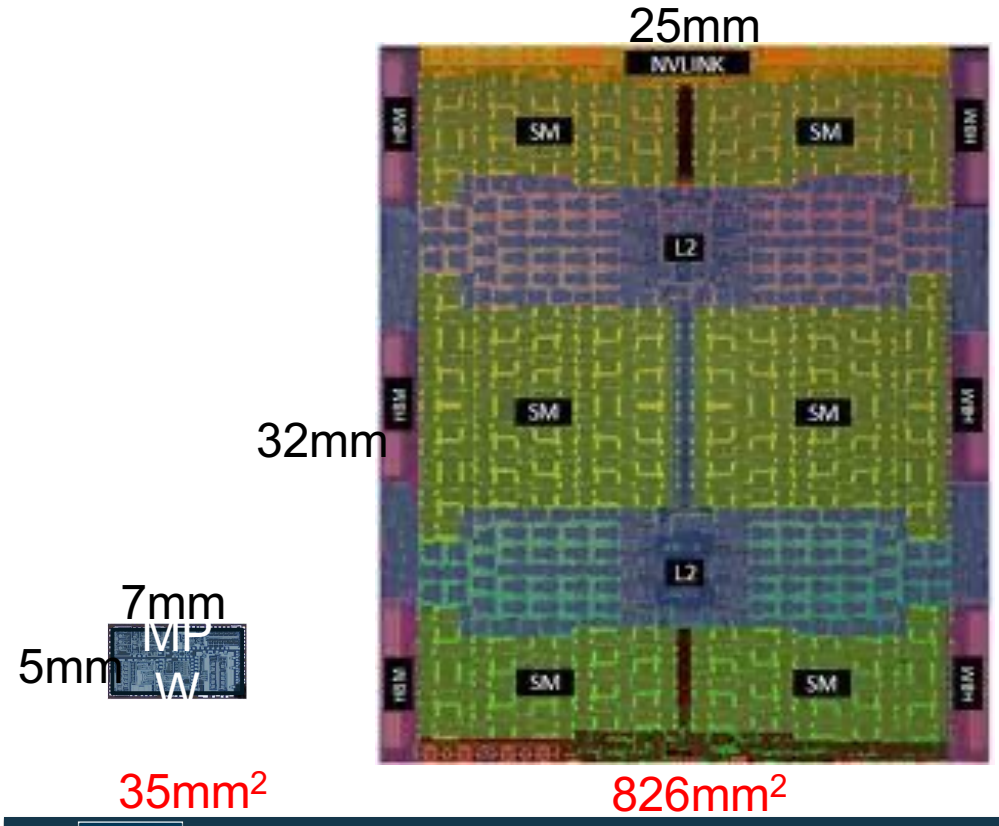


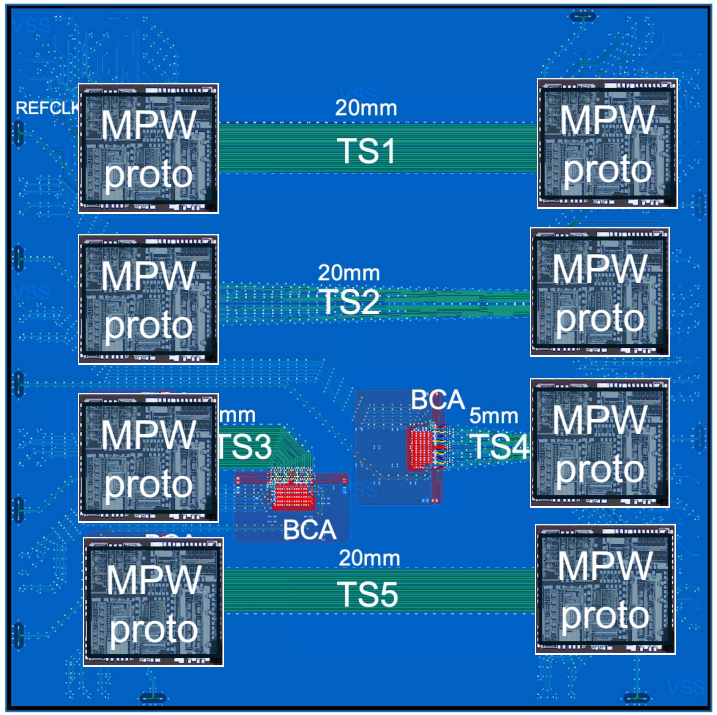
Prototype Breadboard



FPGA Support Board (mem+IO)

Chiplets for Scalable MPW Prototypes





Chiplets Demystification Tutorial

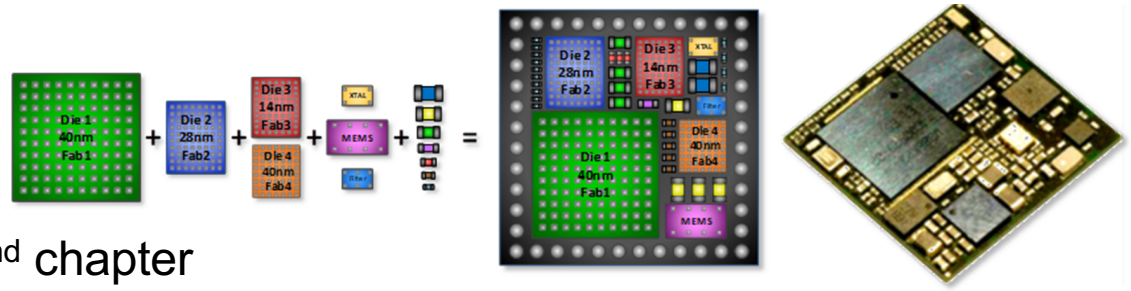
Industry: *Heterogeneous Integration Roadmap*

HETEROGENEOUS INTEGRATION ROADMAP

2019 Edition

http://eps.ieee.org/hir

All future applications will be further transformed through the power of AI, VR, and AR.



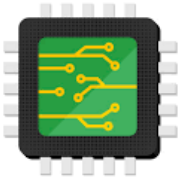
Die + Heterogeneous

System in Package (SiP)

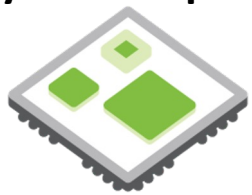
HPC and Mega-datacenters is 2nd chapter

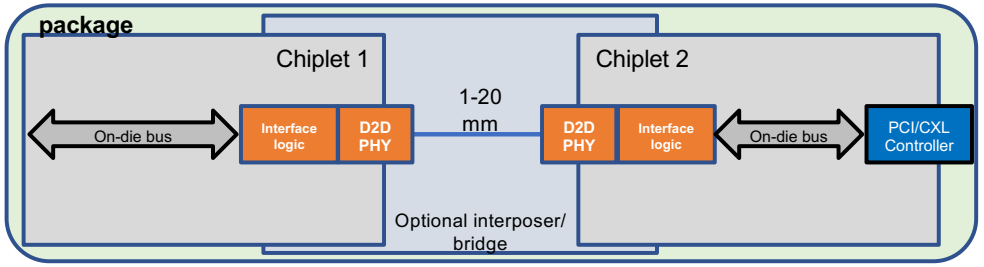
What is a Chiplet - Disaggregation

Chip: Homogenous logic, the one die in a package



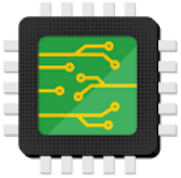
Chiplet: Heterogeneous logic, Many die in a package

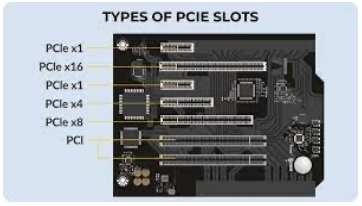




Chiplets Details

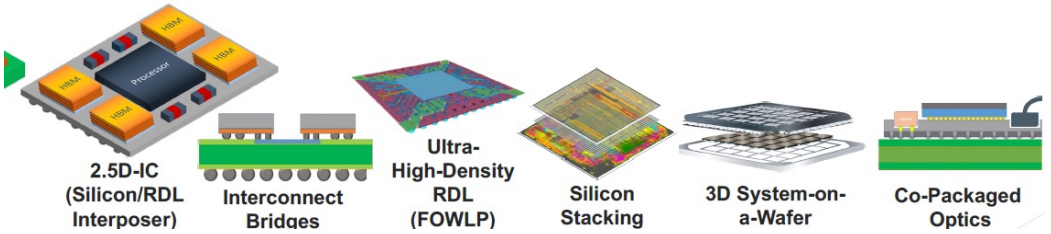
Chip: Homogenous logic, the one die in a package





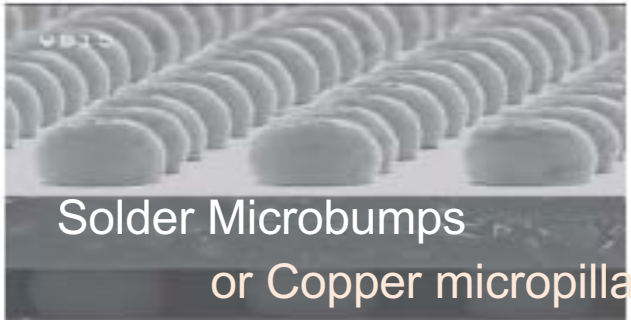
Off-package interconnect + simple package

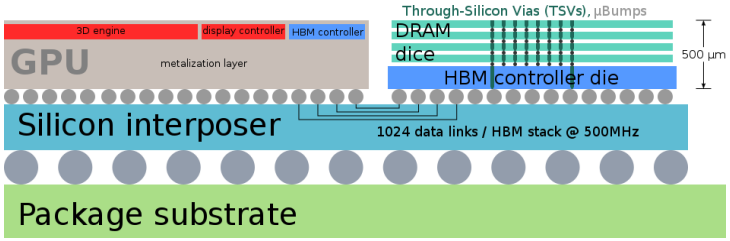
Chiplet: Heterogeneous logic, Many die in a package

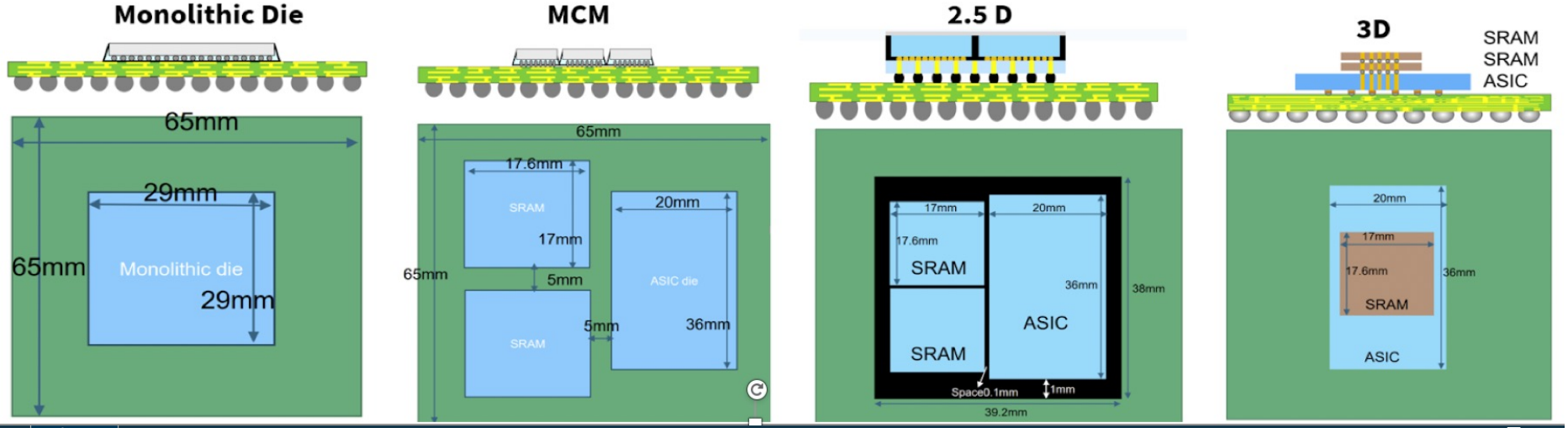


On-package interconnect + more complex packages

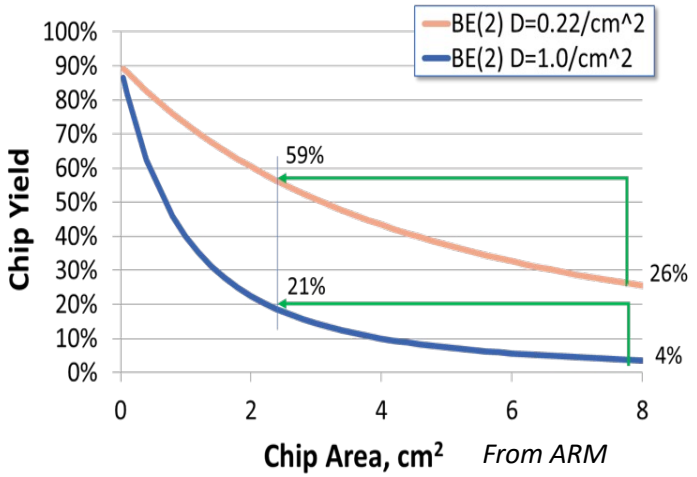
What is a Chiplet?



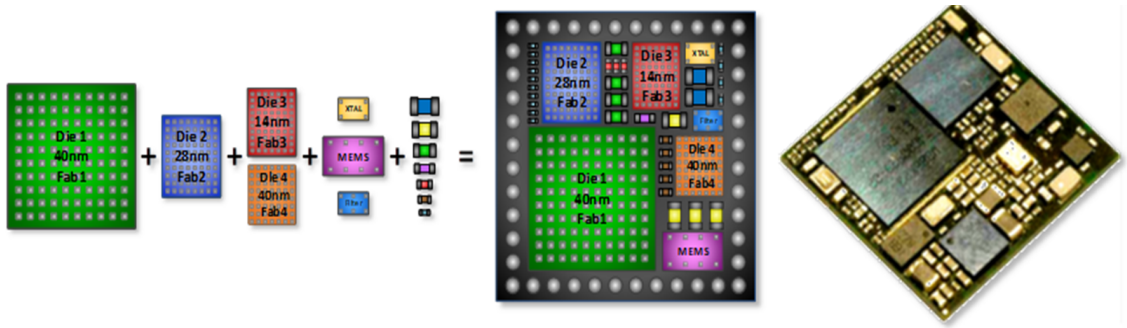




Why?: Chiplets Can Lower Manufacturing and Design Costs



From ARM

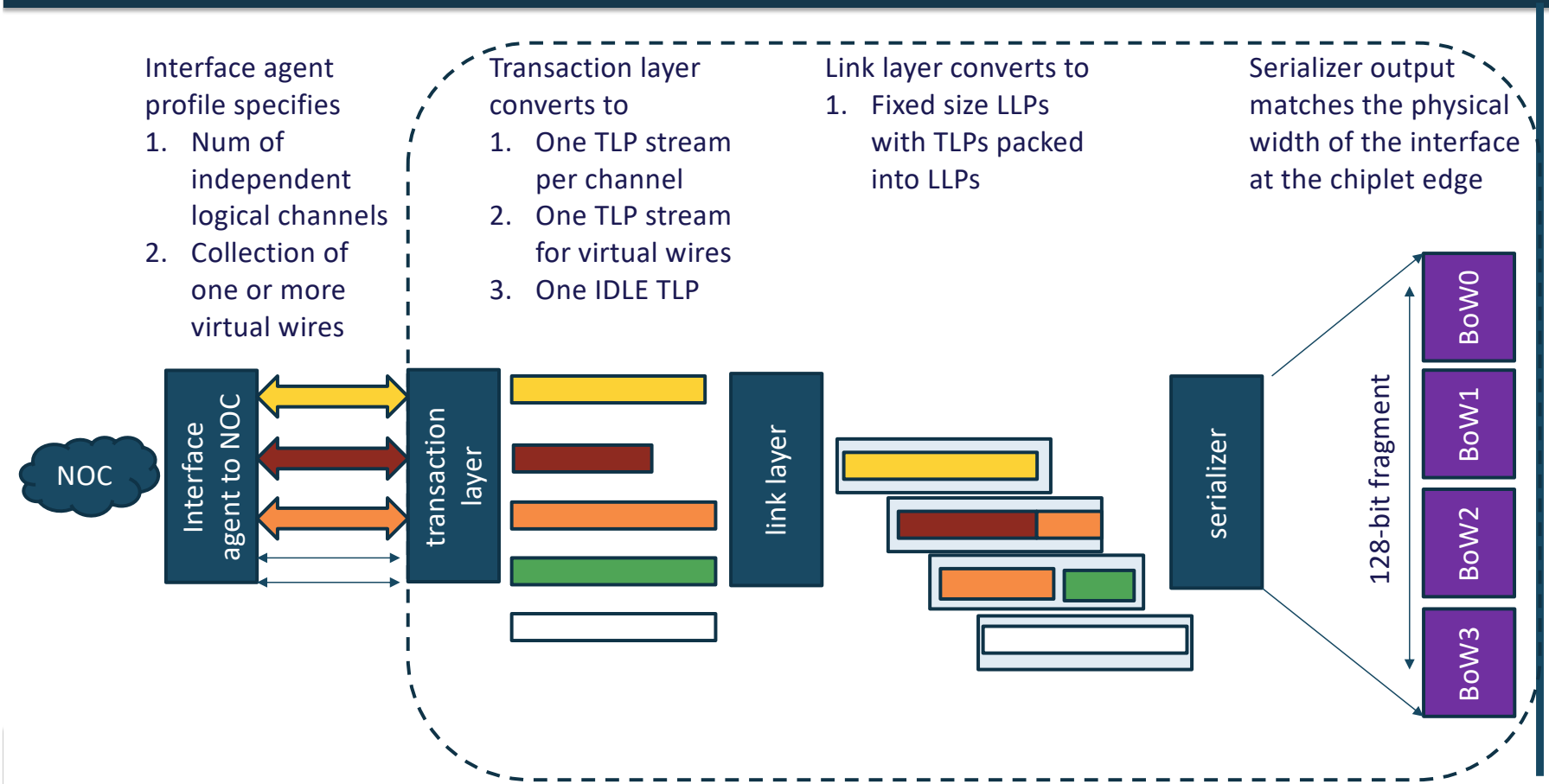


Die + Heterogeneous

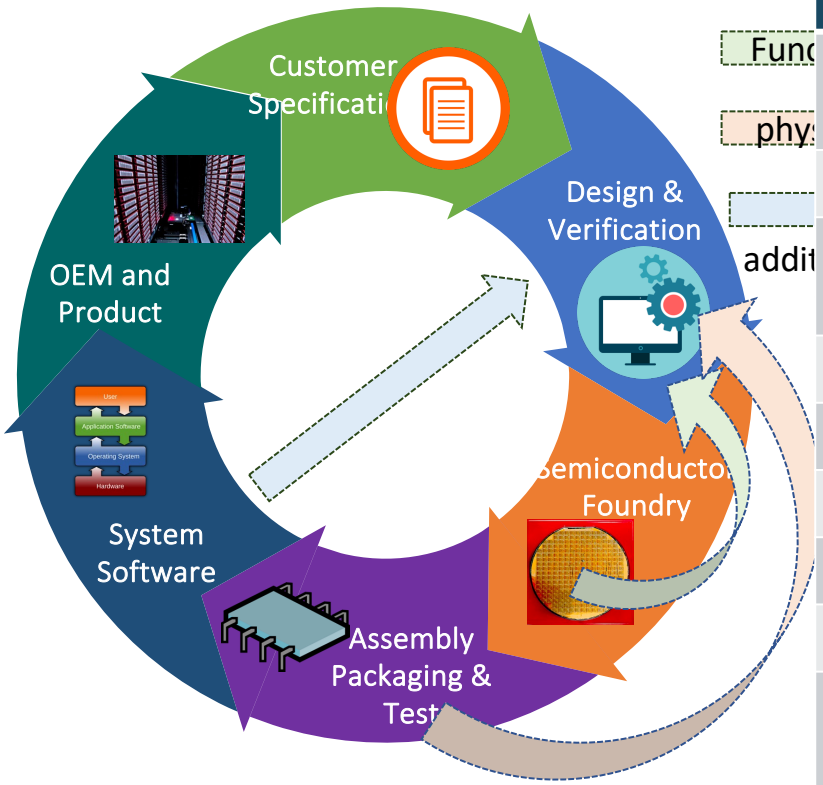
System in Package (SiP)

1. **Scale**: Build products larger than with single die
2. **Modularity**: Partition system into recognizable blocks, buy externally
3. **Reuse**: Create variants by changing a small part of a design, faster TTM
4. **Heterogeneous integration:** Die from multiple nodes, optimized for function
5. **Customize product by customizing chiplet**

Link and Transaction Layer for BoW



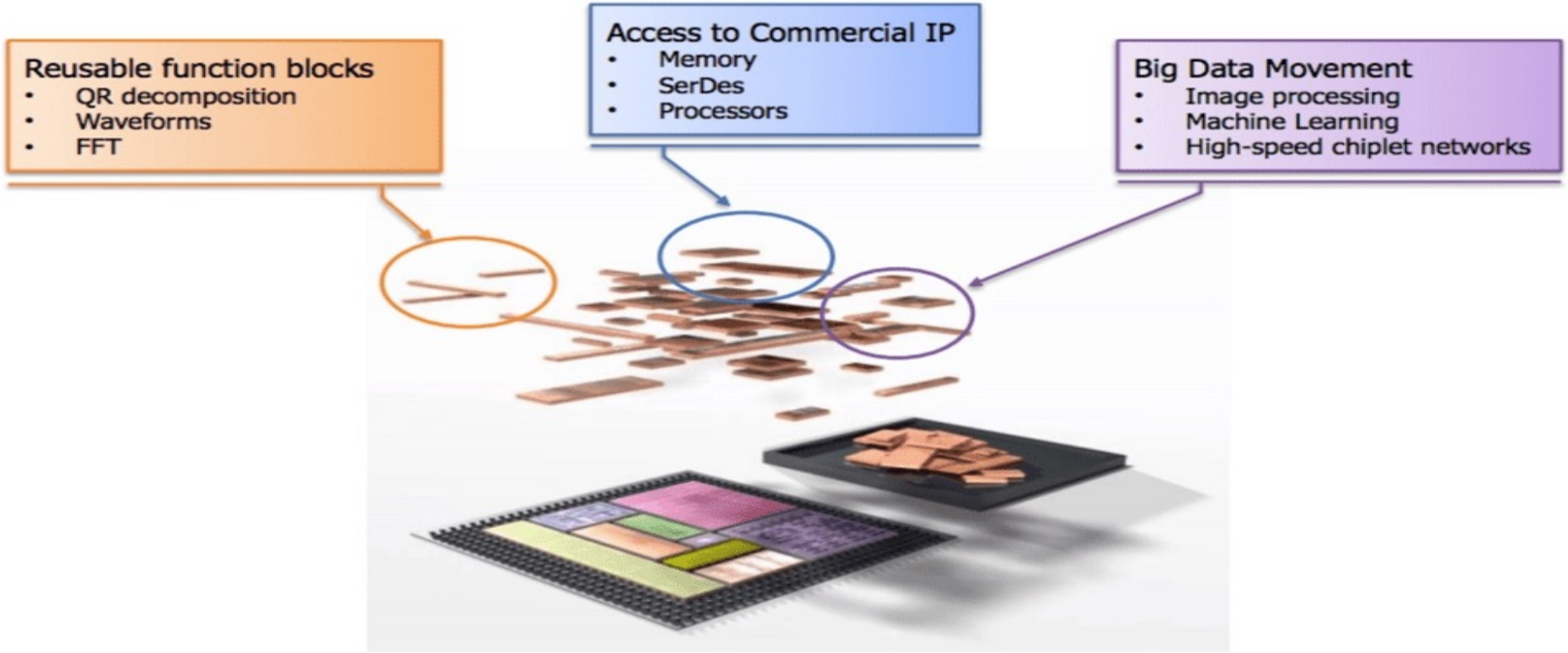
Not a Free Lunch:
Chiplets Make Workflow More Complex



Component	Status
D2D Interconnect (Huge growth/awareness here)	**UCIe**, BoW, AIB, XSR
Test	IEEE 1838, IEEE P3405
Chiplet description	JEDEC-OCP JEP 30 CDXML **(new in 2023)**
Size guardrails	✗
Power delivery guardrails	✗
Thermal guardrails	✗
Wiring density guardrails	✗
Mechanical guardrails	✗
Bump and assembly pitch guardrails	✗

How do chiplets enable domain specialization?

Lower cost barriers to co-integrating specialization From DARPA CHIPS

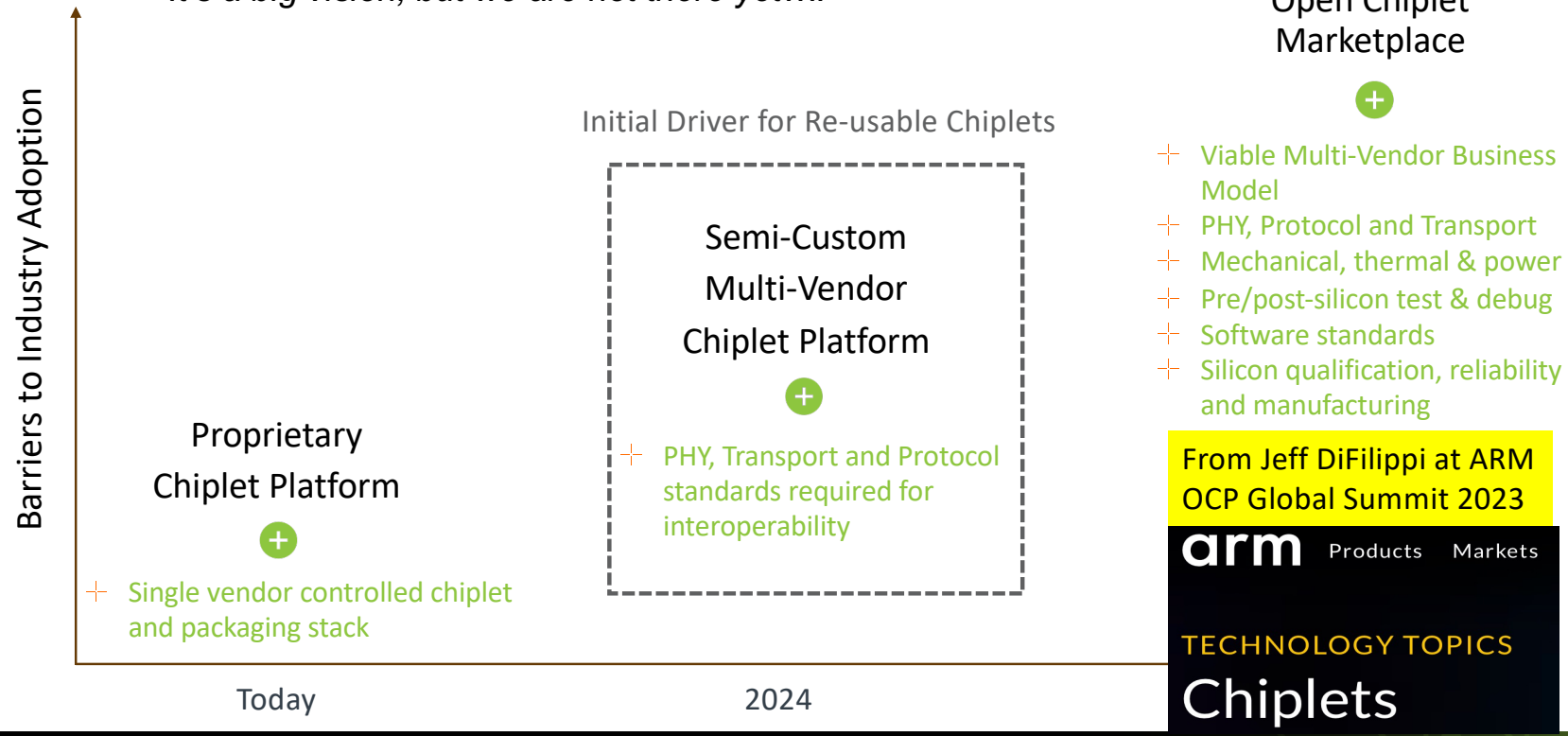


See the multi-agency chiplets workshop at https://sites.google.com/lbl.gov/chiplets-workshop-2023/home

CHIPS modularity targets the enabling of a wide range of custom solutions

The Road to an Open Chiplet Marketplace

It's a big vision, but we are not there yet….



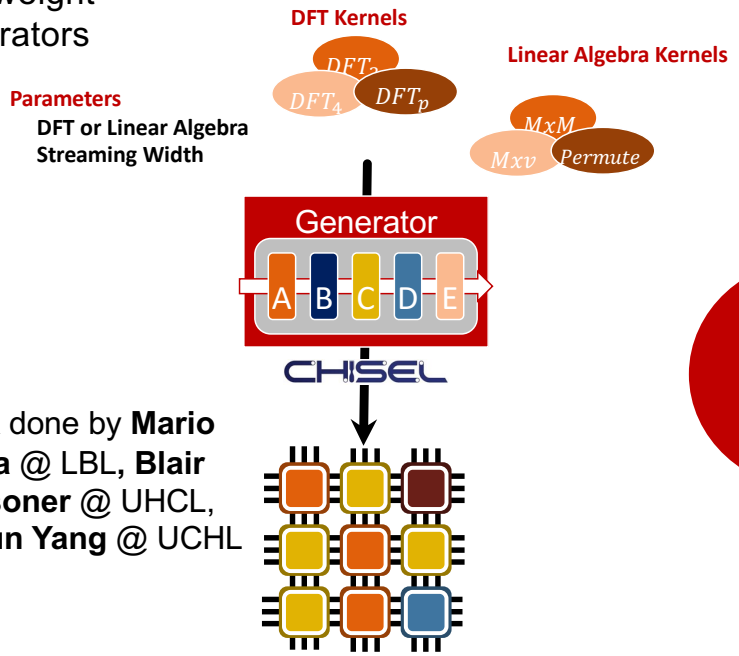
From Jeff DiFilippi at ARM
OCP Global Summit 2023

arm Products Markets

TECHNOLOGY TOPICS
Chiplets

From A Common Language to Hardware Design

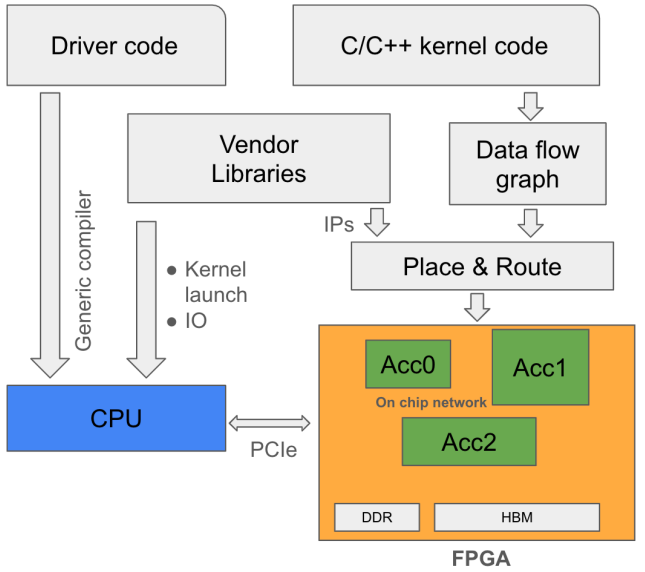
Lightweight generators



Work done by **Mario Vega** @ LBL, **Blair Reasoner** @ UHCL, **Xiaokun Yang** @ UCHL

Two-pronged approach for generating the hardware

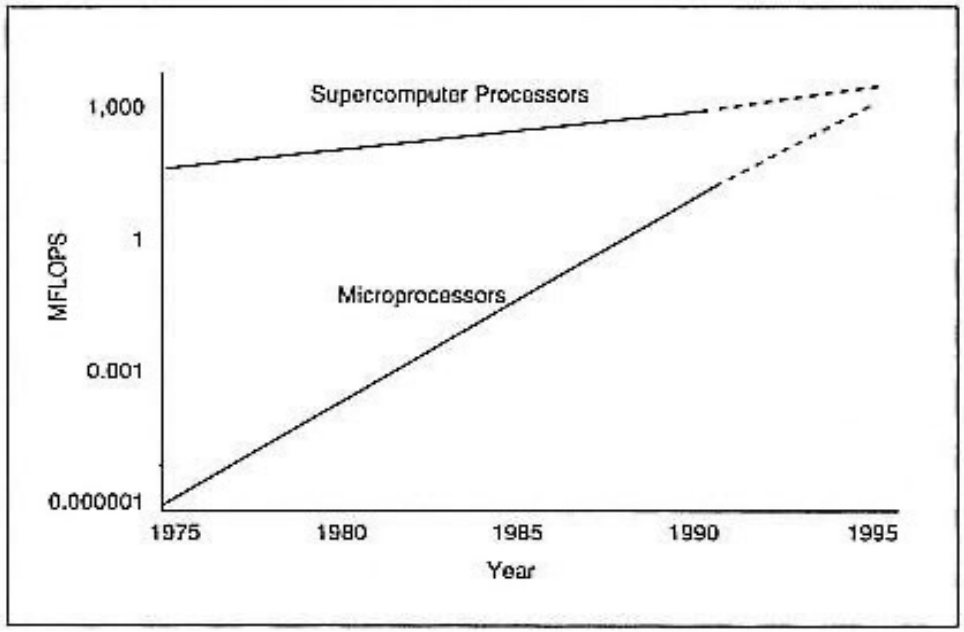
High Level Synthesis using Vivado HLS



"Towards a Flexible Hardware Implementation for Mixed Radix Fourier Transforms", Mario Vega, Xiaokun Yang, John Shalf, HPEC 2023

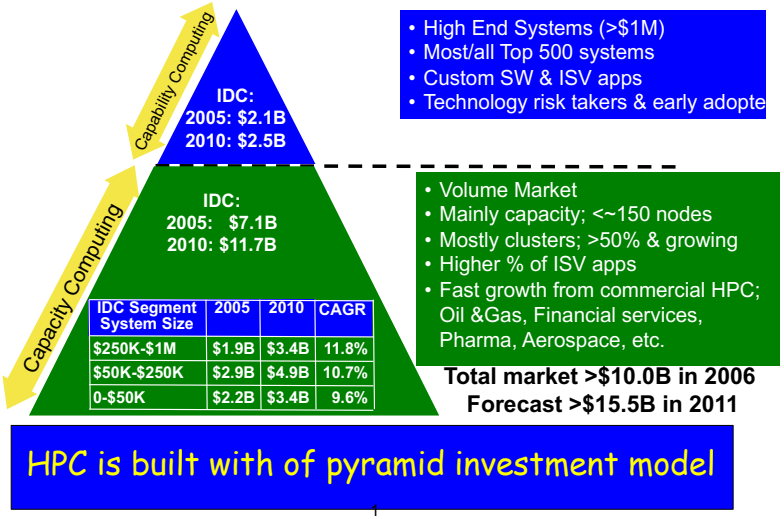
Work done by **Tan Nguyen** @ LBL

Lessons from Attack of the Killer Micros



Attack of the killer micros
John Markoff, May 6, 1991

- **Was more about the economic model than technology alone**

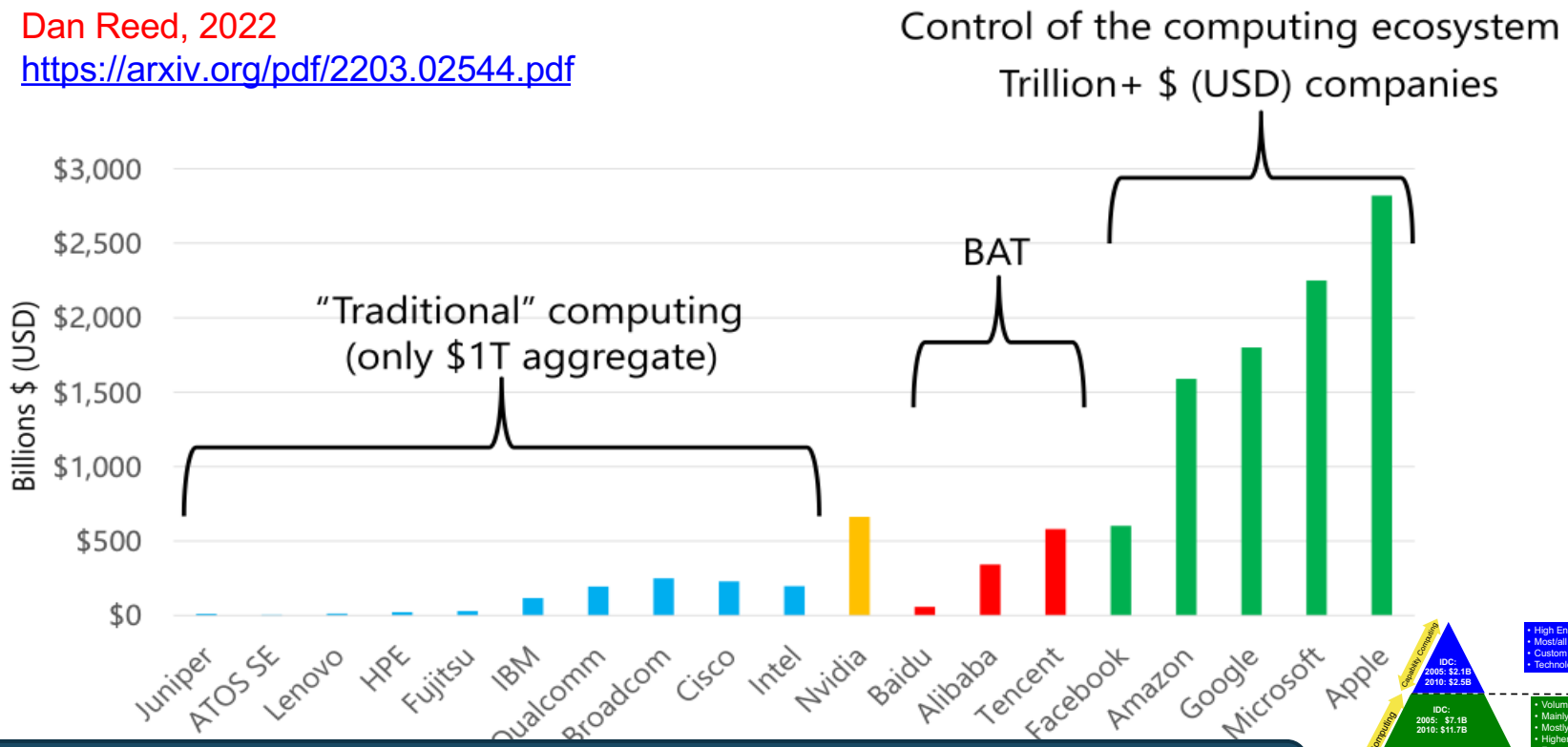


IDC Segment System Size	2005	2010	CAGR
$250K-$1M	$1.9B	$3.4B	11.8%
$50K-$250K	$2.9B	$4.9B	10.7%
0-$50K	$2.2B	$3.4B	9.6%

It is not good enough anymore to understand the technology

Now we must also understand the market context

Dan Reed, 2022
https://arxiv.org/pdf/2203.02544.pdf



Bottom Line: If you want to know the future of HPC and Hyperscalers.
***Follow the money!!!! (but that doesn't mean* quit!!)**