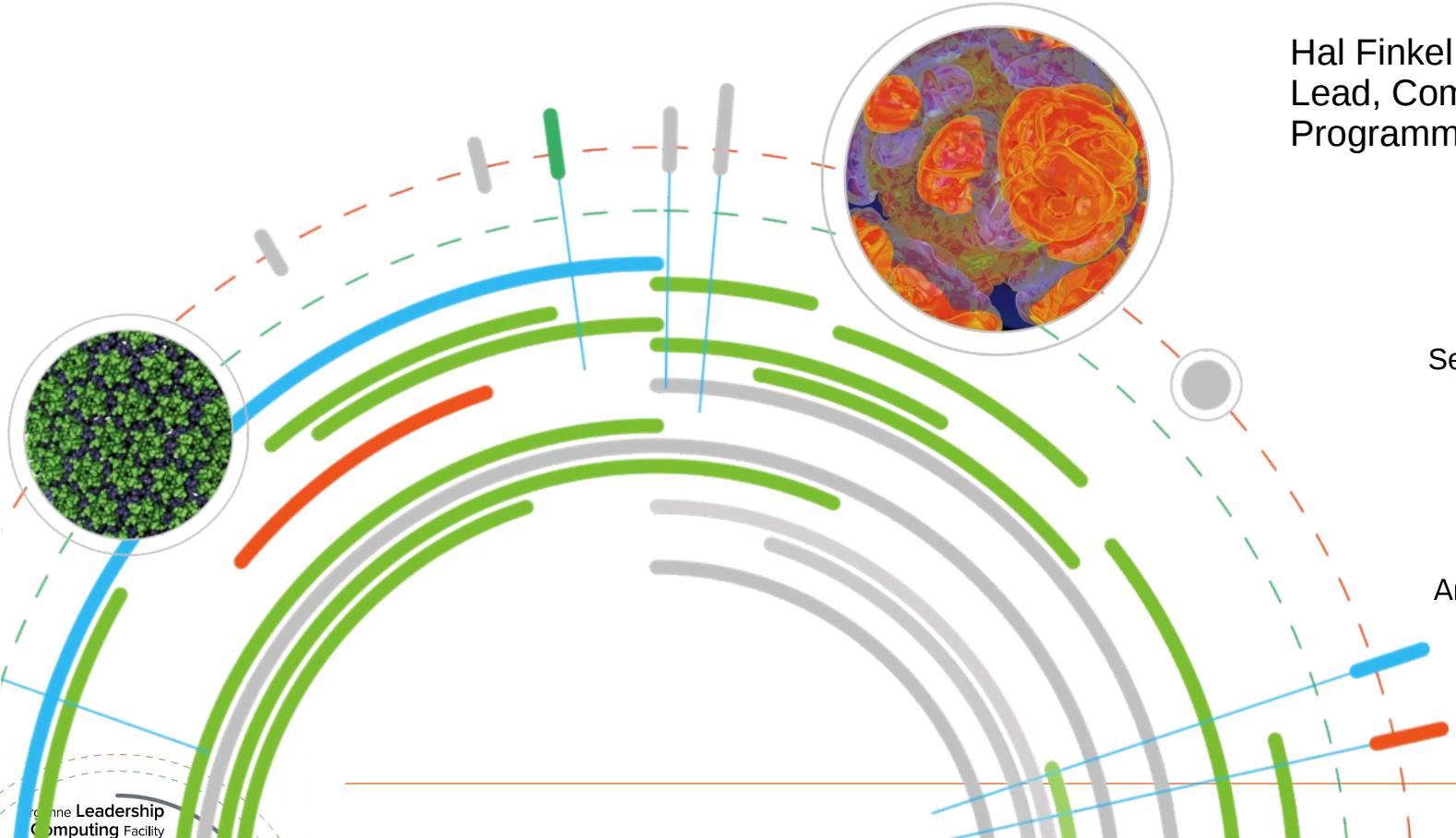
FPGAs and spatial architectures for HPC: Lessons Learned and Future Opportunities



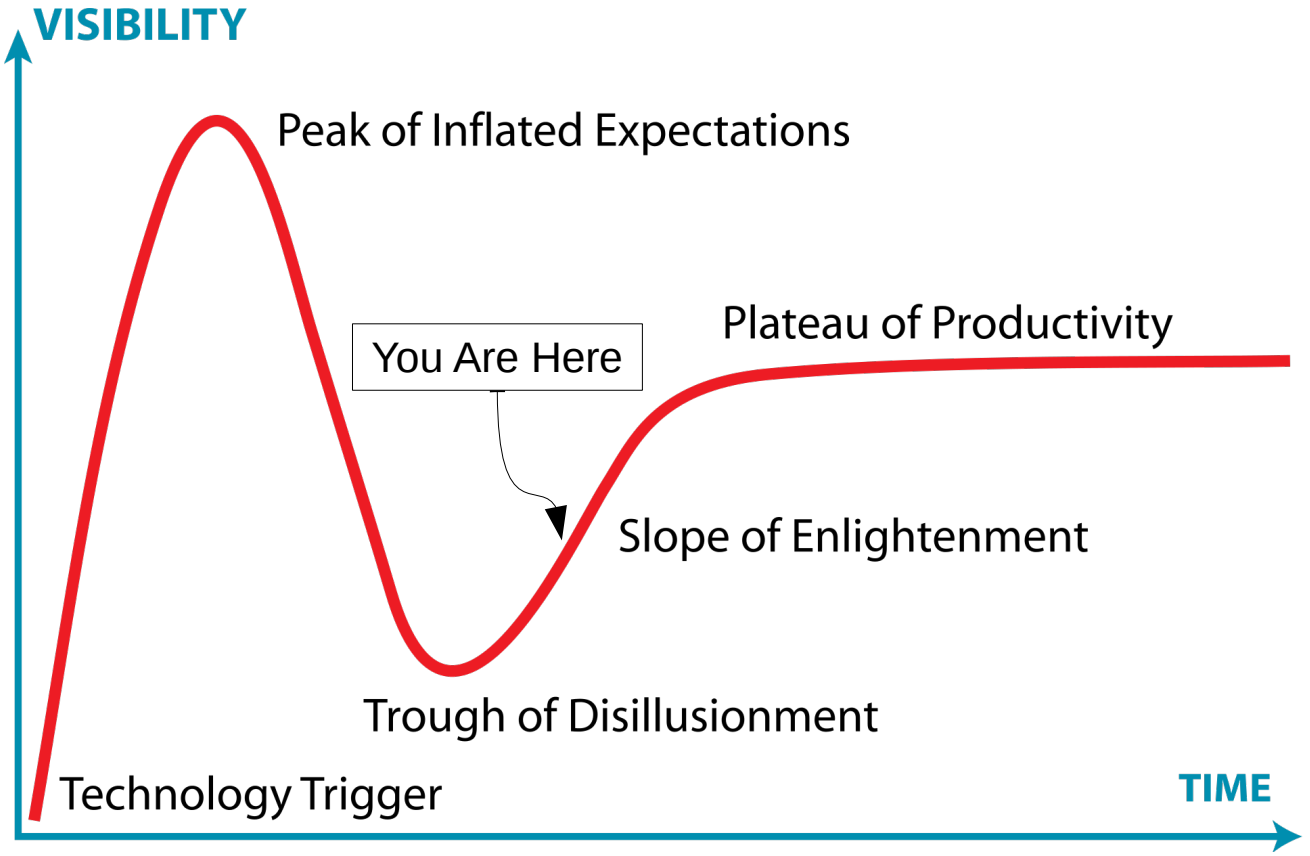
Hal Finkel <hfinkel@anl.gov>
Lead, Compiler Technology and Programming Languages

JLESC 2020
September 10, 2020

Argonne **Leadership Computing** Facility

Argonne NATIONAL LABORATORY

(Spatial) Architecture is Interesting Again...



https://en.wikipedia.org/wiki/Hype_cycle

Let's Talk About Headroom

Operation	Energy (pJ)
64-bit integer operation	1
64-bit floating-point operation	20
256 bit on-die SRAM access	50
256 bit bus transfer (short)	26
256 bit bus transfer (1/2 die)	256
Off-die link (efficient)	500
256 bit bus transfer (across die)	1,000
DRAM read/write (512 bits)	16,000
HDD read/write	$O(10^6)$

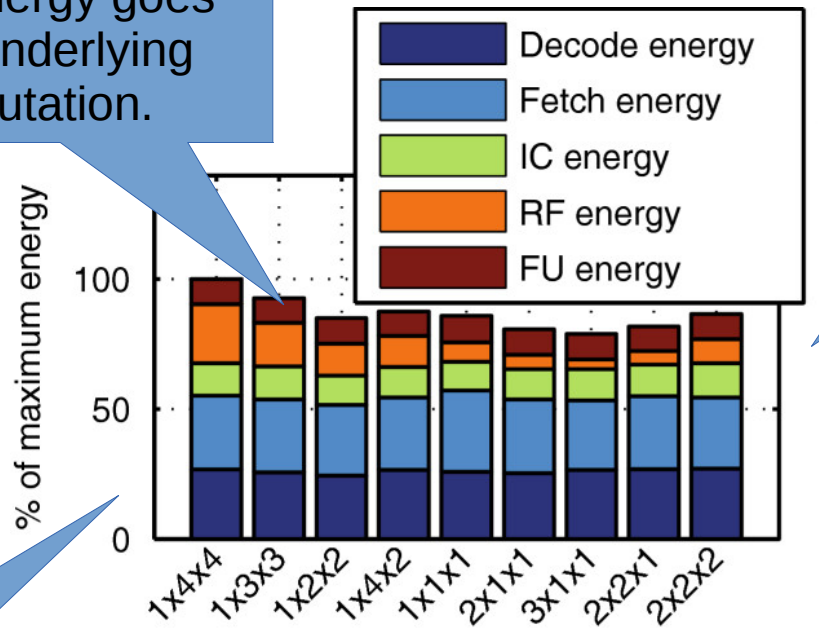
Courtesy Greg Asfalk (HPE) and Bill Dally (NVIDIA)

Do FPGAs/CGRAs perform less data movement per computation?

http://science.energy.gov/~/media/ascr/ascac/pdf/meetings/201604/McCormick-ASCAC.pdf

Let’s Talk About Headroom

Only a small portion of the energy goes to the underlying computation.



More centralized register files means more data movement which takes more power.

Model of CPU Energy Usage

Fetch and decode take most of the energy!

(Model with (# register files) x (read ports) x (write ports))

http://link.springer.com/article/10.1186/1687-3963-2013-9

See also: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2008-130.pdf

Some “Early” Evidence for Success

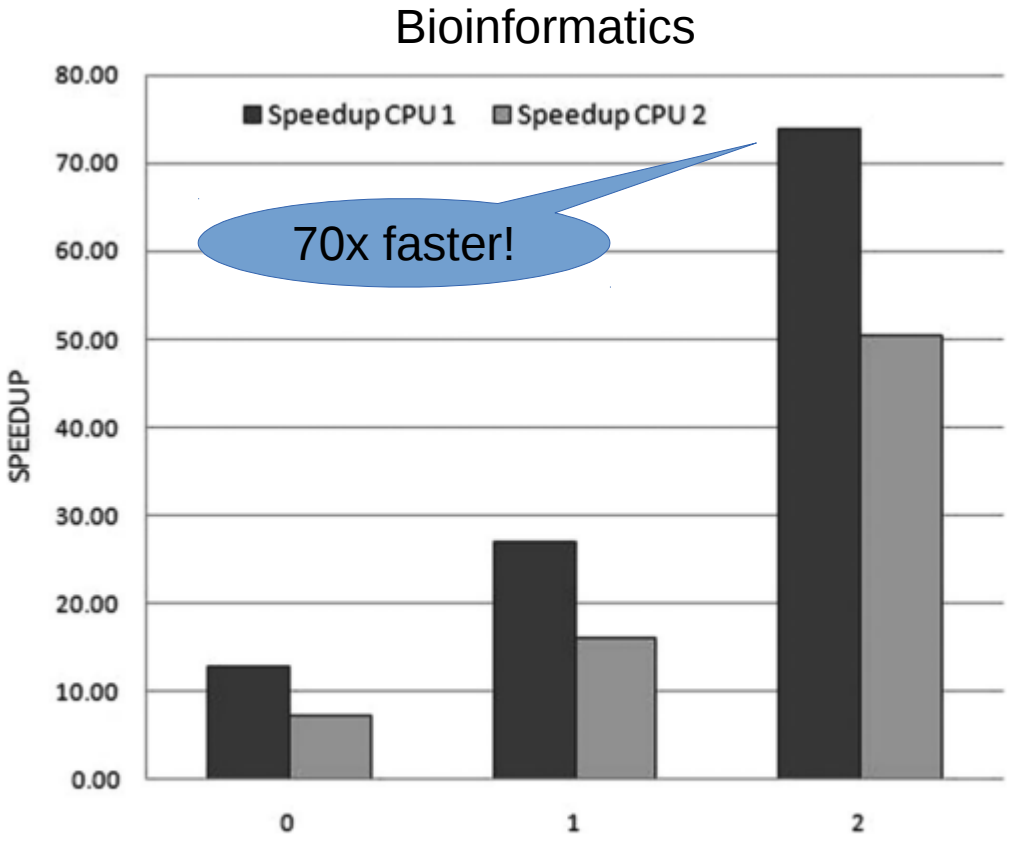


Fig. 9. Speed up of FHAST compared to BOWTIE for exact matches, one and two mismatches.

http://escholarship.org/uc/item/35x310n6

Some “Early” Evidence for Success

Machine learning and neural networks

FPGA is faster than both the CPU and GPU, 10x more power efficient, and a much higher percentage of peak!

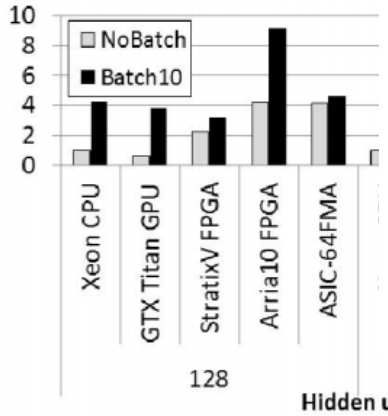


Fig. 5. Performance for all the accelerators under study, relative to CPU performance with no batching.

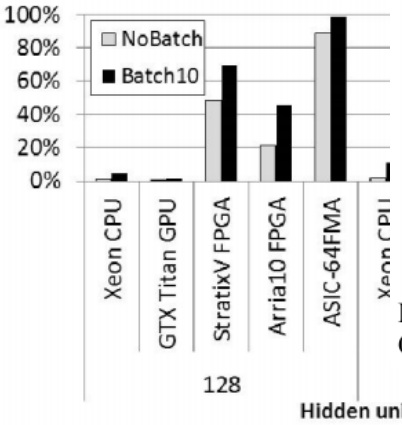


Fig. 6. Achieved performance relative to peak performance. E.g., 10% means the system is underutilized, where the achieved GFLOP/s is only at 10% of the available peak GFLOP/s. On the other hand, 100% means full utilization.

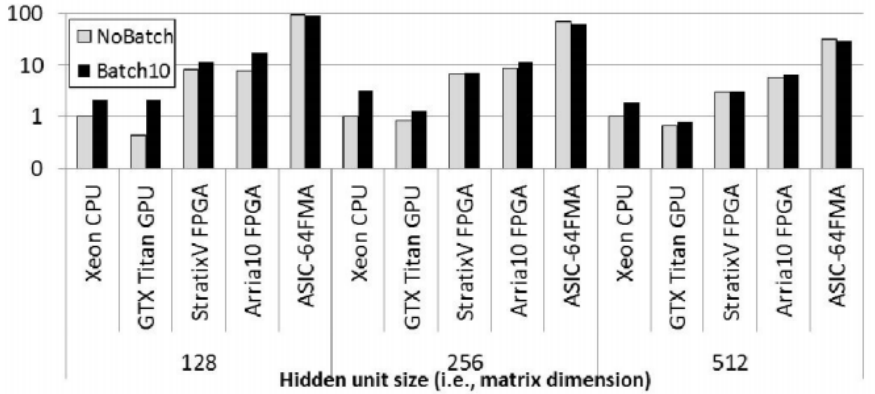
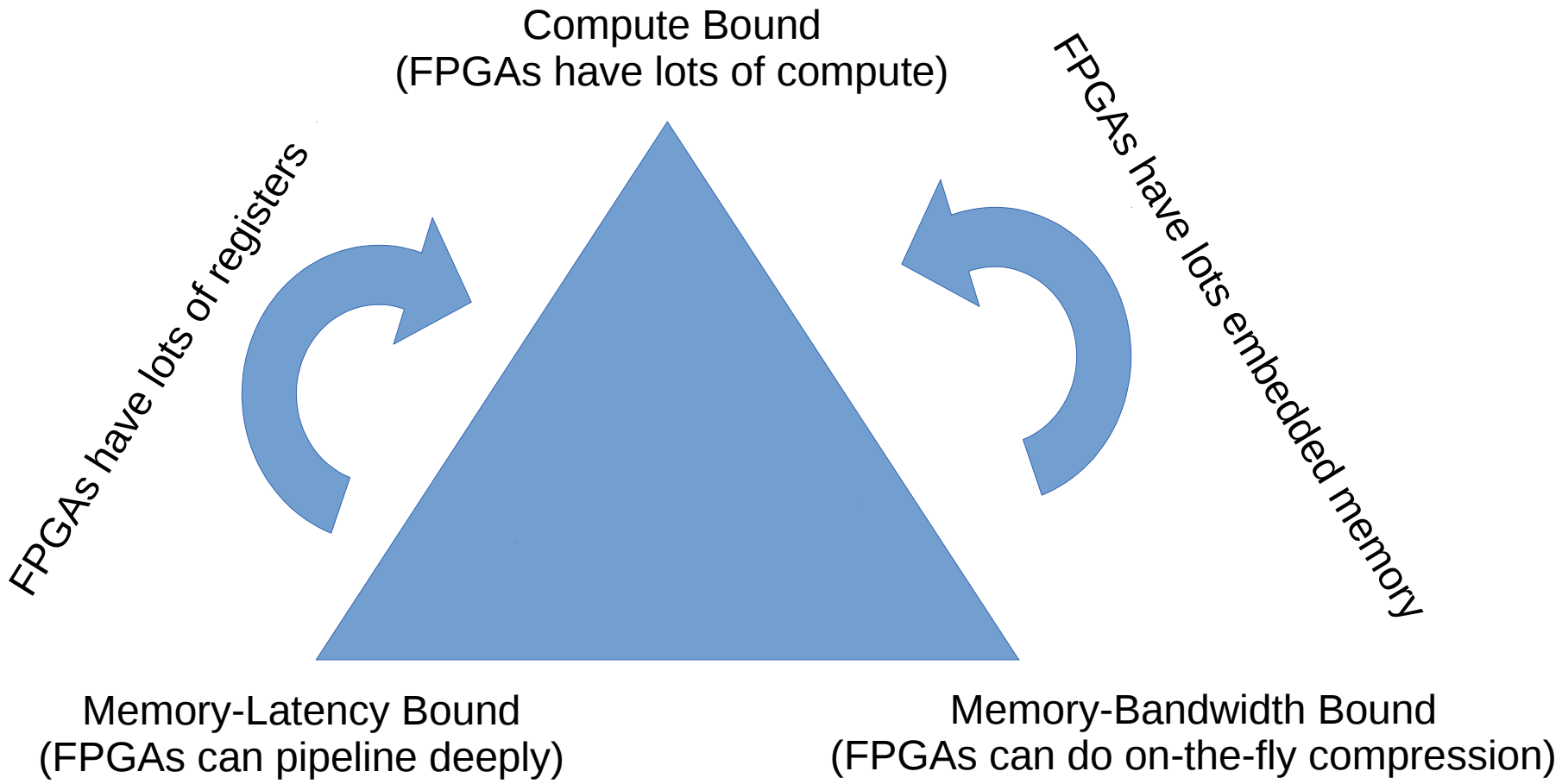


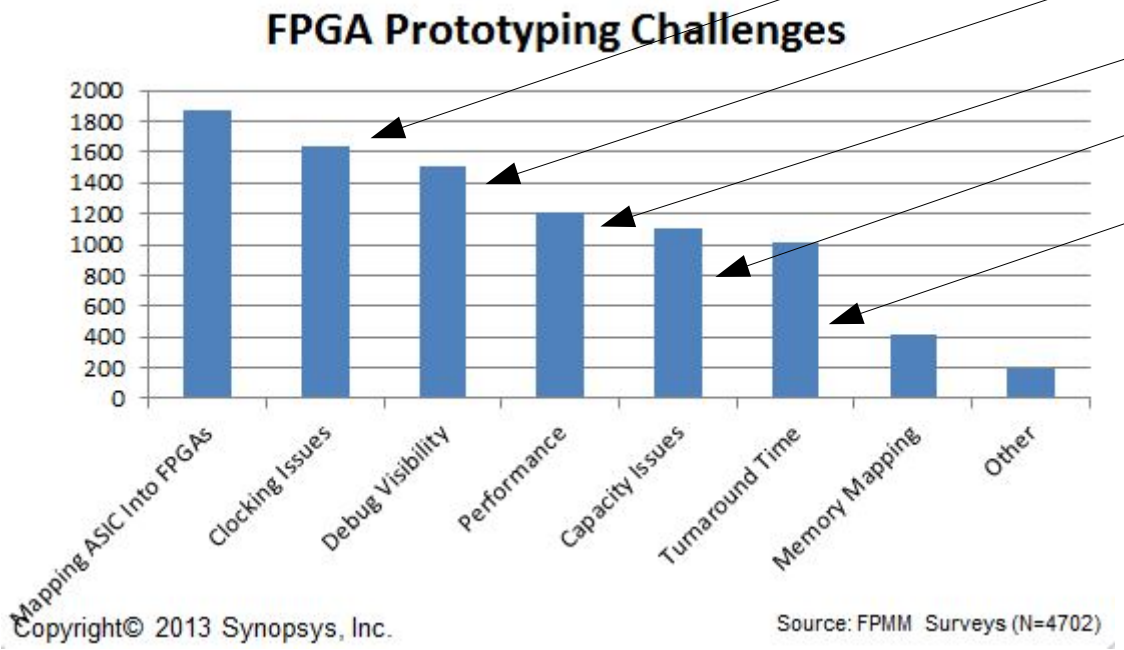
Fig. 7. Performance/Watt for all the accelerators under study, relative to the CPU performance with no batching. The y-axis is in log-scale.

http://ieeexplore.ieee.org/abstract/document/7577314/

What and where are the opportunities? Who are the stakeholders? (cont.)



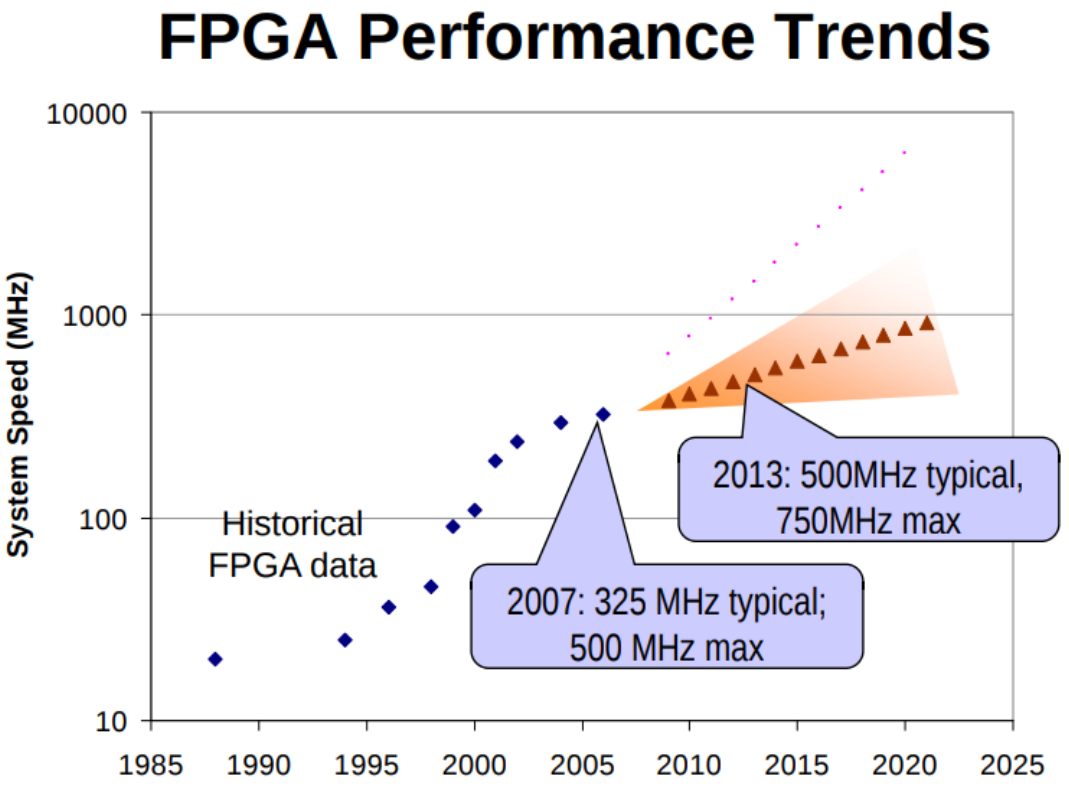
But There Are Challenges...



- It's hard to use all of the resources
- Debugging is hard
- Serial performance is low
- You might not have enough resources
- Compilation is slow

https://blogs.synopsys.com/breakingthethreelaws/2013/12/xilinx-fpga%E2%80%99s-for-fpga-based-prototyping/

A Note About Serial Performance...



f_{max} for the Intel Stratix 10, top speed grades (–E1V and –I1V)

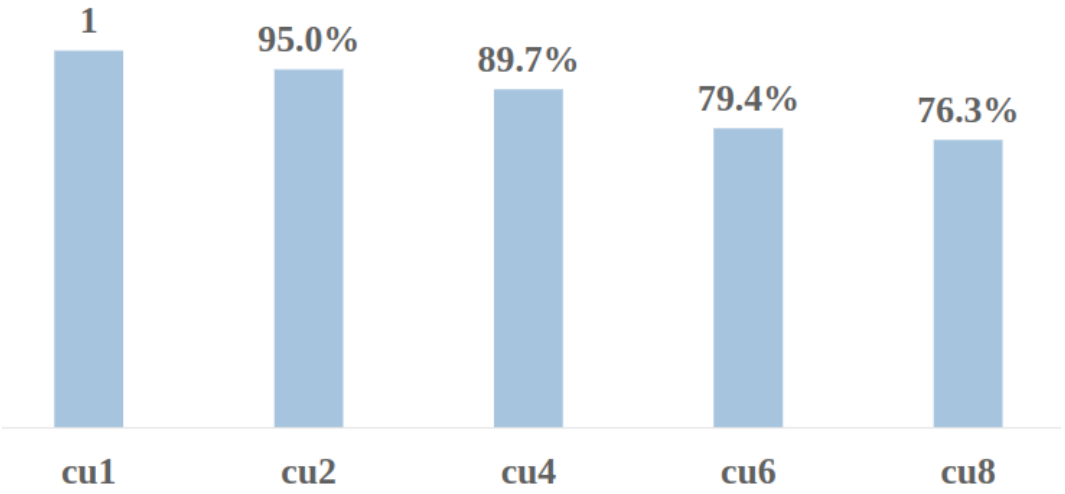
Programmable clock routing	1000 MHz
DSP block: Fixed-point multiplication modes	1000 MHz
DSP block: Floating-point modes	750 MHz
Memory (M20K) block: Simple dual-port modes	1000 MHz
Memory (M20K) block: True dual-port modes	600 MHz

https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/stratix-10/s10_datasheet.pdf

https://indico.cern.ch/event/21985/contributions/1522471/attachments/357002/497110/linderstruth.pdf

Keeping Up The Clock Rate Is Hard

Kernel Impl.	Logic	Memory Bits	RAM Blocks	#DSP	Fmax (MHz)
base	19%	5%	10%	2	276
cu1	19%	5%	11%	2	265
cu2	27%	5%	12%	4	251
cu4	43%	6%	13%	8	236
cu6	55%	7%	47%	12	209
cu8	70%	7%	60%	16	201



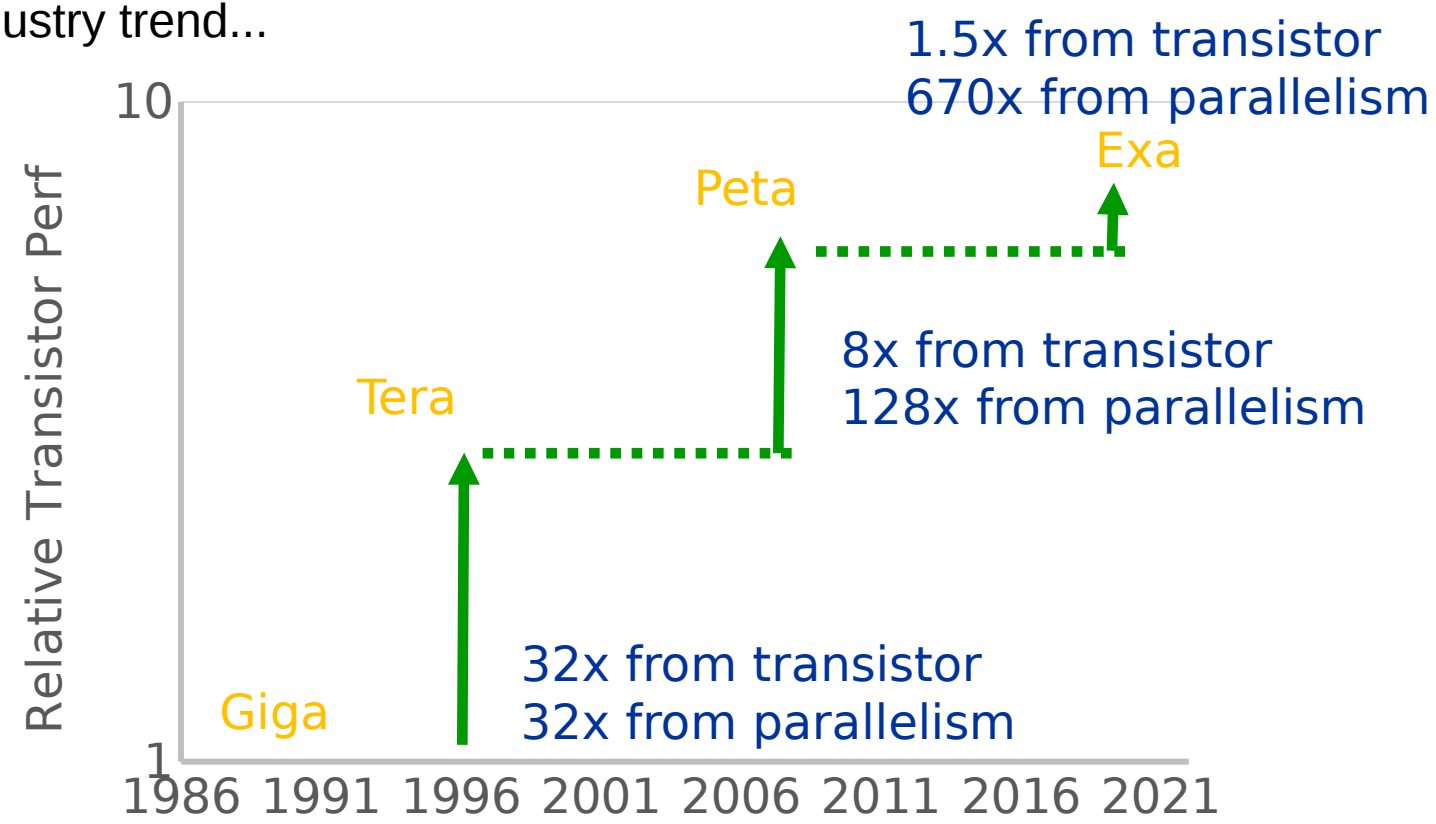
As the number of “compute units” is increased, and more logic is used…

The clock rate and parallel efficiency goes down.

This is an evaluation of an MD5Hash kernel using OpenCL on an Intel Arria 10 FPGA
(Zheming Jin and Hal Finkel, PDSEC 2018)

Low Serial Performance → Performance From Parallelism

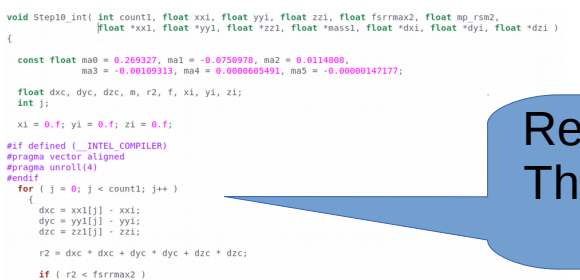
This is an industry trend...



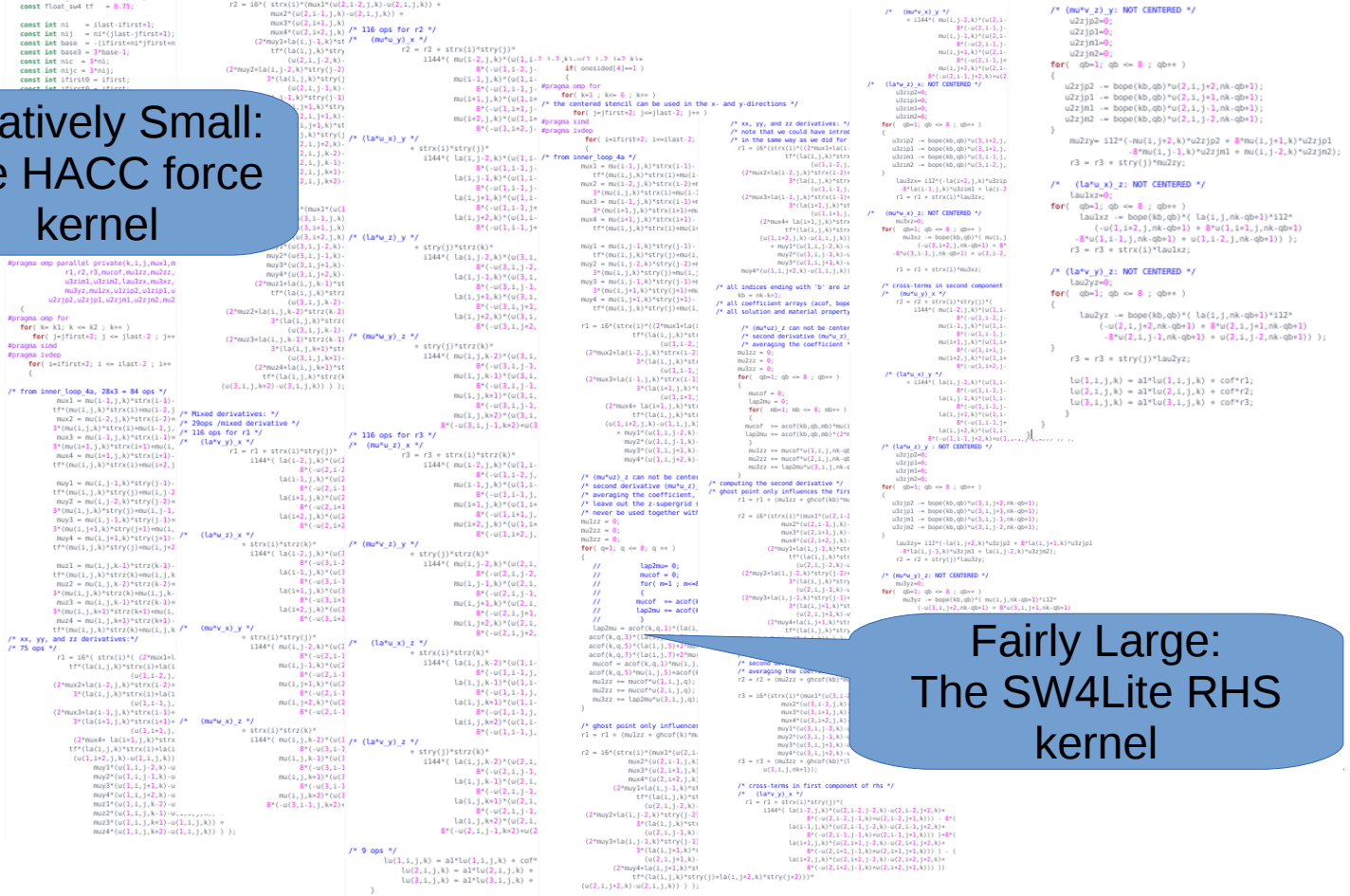
System performance from parallelism

http://www.socforhpc.org/wp-content/uploads/2015/06/SBorkar-SoC-WS-DAC-June-7-2015-v1.pptx

On The Size of HPC Kernels



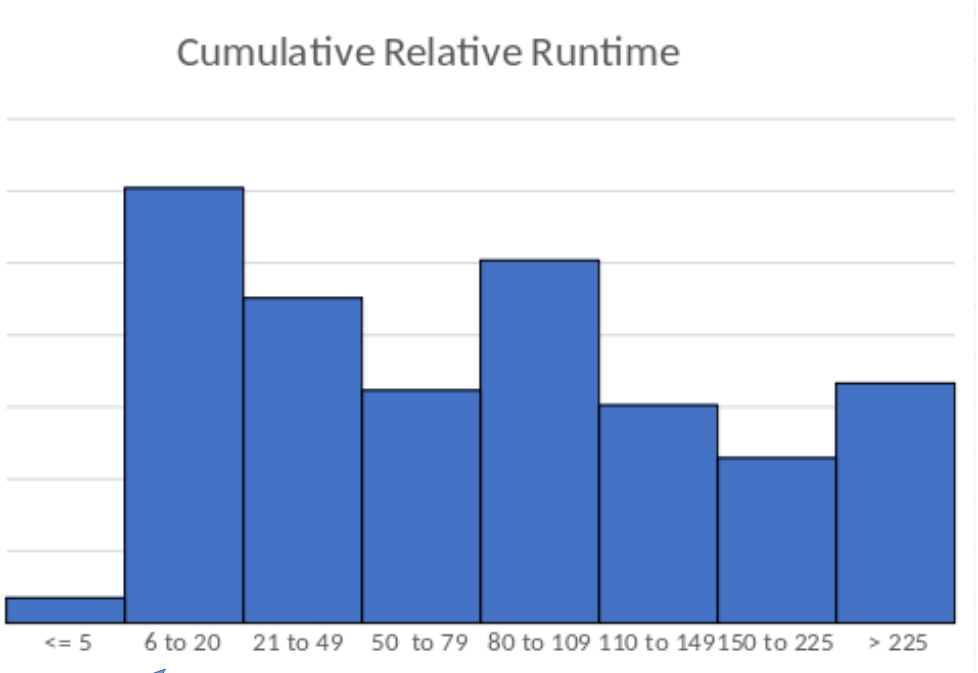
Relatively Small:
The HACC force kernel



Fairly Large:
The SW4Lite RHS kernel

https://github.com/geodynamics/sw4lite/blob/master/src/rhs4sg.C

On The Size of HPC Kernels



Lines of code in kernels in DOE proxy apps: ASPA, SNAP, SNBone, SW4lite, Nekbone, SWFFT, miniFE, Lulesh, miniAero, MiniGMG, CoMD, CoSP2, HPCCG, miniTri, PENNANT, Pathfinder, RSBench, SimpleMOC, XSBench – Data collected by Brian Homerding (ALCF).

Many kernels are small (6-20 lines)

But many kernels are large (200+ lines)

On The Size of HPC Kernels

Intel Stratix 10 GX/SX Device Name	Logic Elements (KLE)	M20K Blocks	M20K Mbits	MLAB Counts	MLAB Mbits	18x19 Multi-pliers (1)
GX 400/ SX 400	378	1,537	30	3,276	2	1,296
GX 650/ SX 650	612	2,489	49	5,364	3	2,304
GX 850/ SX 850	841	3,477	68	7,124	4	4,032
GX 1100/ SX 1100	1,325	5,461	107	11,556	7	5,184
GX 1650/ SX 1650	1,624	5,851	114	13,764	8	6,290
GX 2100/ SX 2100	2,005	6,501	127	17,316	11	7,488
GX 2500/ SX 2500	2,422	9,963	195	20,529	13	10,022
GX 2800/ SX 2800	2,753	11,721	229	23,796	15	11,520
GX 1660	1,679	6,162	120	14,230	9	6,652
GX 2110	2,073	6,847	134	17,856	11	7,920

The largest Stratix 10 has 5760 DSP blocks (FP32 multipliers)

If you didn't need any for integer ops, @750 Mhz that is 8.2 TF FP32 peak. A GV100 is 11.3 FP32 TF (16.3 TF boost).

https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/stratix-10/s10-overview.pdf

On The Size of HPC Kernels

Kernel	Logic utilization (%)	RAM blocks utilization (%)	DSP count
K0	33	60	192
K1	23	61	297
K2	20	56	489
K3	27	95	977
K4	27	95	977
K5	23	58	297
K6	21	55	489

CalcFBHourglassForce
LULESH kernel, K3 is SIMD2.

- Looking at LULESH on the Arria 10, which significantly smaller than a Stratix 10, but nevertheless, one compute unit with some SIMD and you're done...

EvalEOSForElems
LULESH Kernel, K4 is single-work-item SIMD8, on an Arria 10.

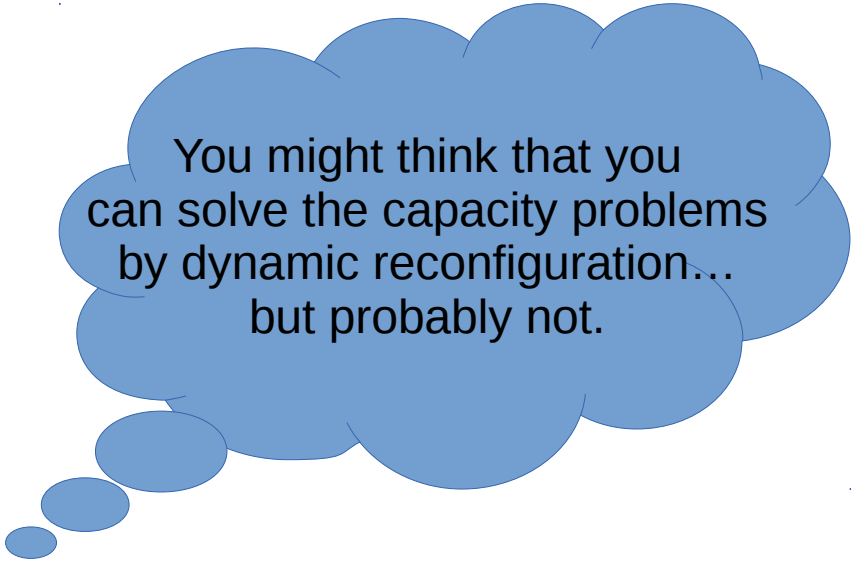
Kernel	Logic utilization (%)	RAM blocks utilization (%)	DSP count
K0	17	18	80
K1	16	21	80
K2	17	23	160
K3	22	27	320
K4	35	45	640

(Zheming Jin and Hal Finkel, 2018)

https://link.springer.com/chapter/10.1007/978-3-030-17227-5_15

The “Kernel Launch Overhead” Problem

Device	Time To Launch a New “Kernel”
Intel Stratix 10	200-400 ms (reconfiguration time)
NVIDIA GPU	50-100 µs



https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/stratix-10/s10_datasheet.pdf

On FPGA Performance

We've done a lot of evaluation work on Intel FPGAs using OpenCL, and…

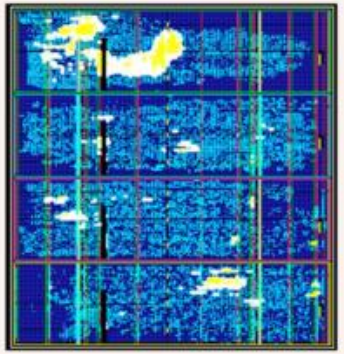
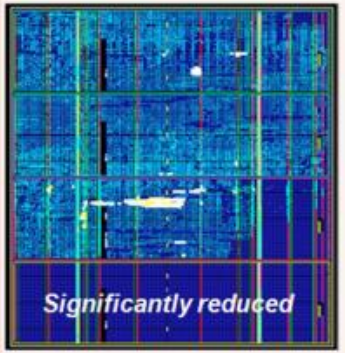
- Often, the GPU is better (even in performance / Watt)!
- FP64 performance is poor (lacks dedicated hardware).
- It's difficult to provision an FPGA to keep up with high memory bandwidth when the logic runs at low clock rate.
- The memory subsystems (at least those provided by OpenCL) are simple, and lack a cache. A fundamental tension in many cases:

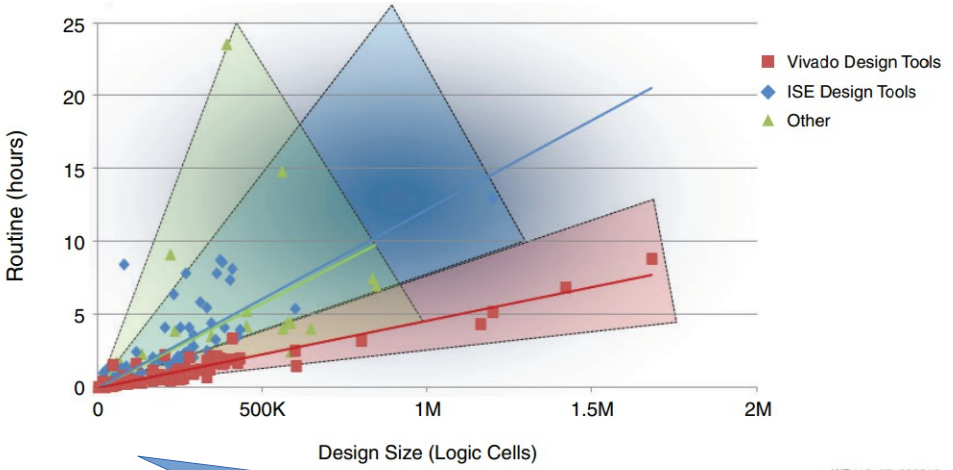
Taking advantage of fundamental "data movement" efficiency means simple, direct pipelines...

Simple, direct pipelines with non-trivial memory accesses stall a lot

Compile Time

FPGA place-and-route takes hours and a lot of memory!

	ISE	Vivado
P&R runtime	13 hrs.	**5 hrs**
Memory usage	16 GB	**9 GB**
Wire length and congestion		



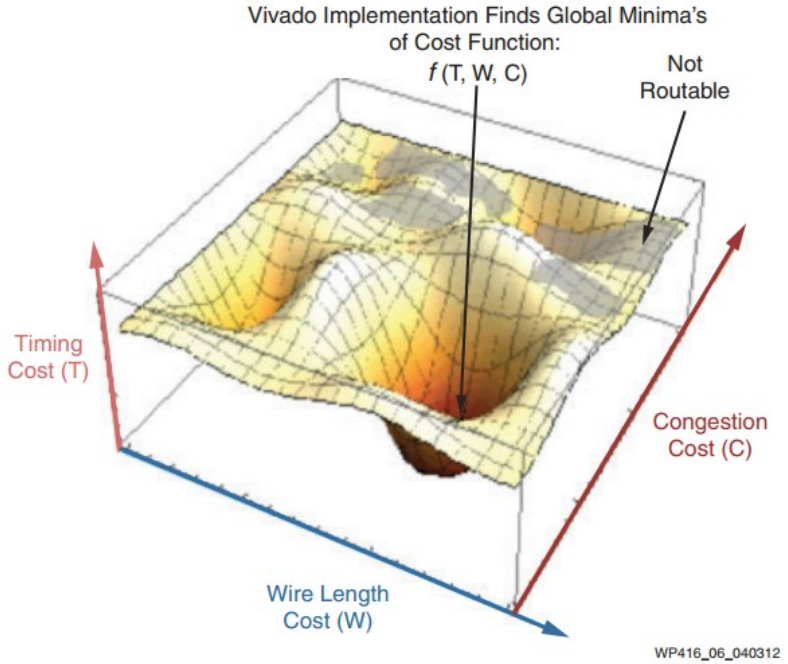
Past 100k elements, and we're already in trouble...

https://www.allaboutcircuits.com/news/Xilinx-vivado-design-suite-design-implementation/

https://www.eenewseurope.com/news/xilinx-unveils-low-cost-rtl-synthesis-tool-optimized-its-28nm-fpga-fabric/page/0/1

Compile Time

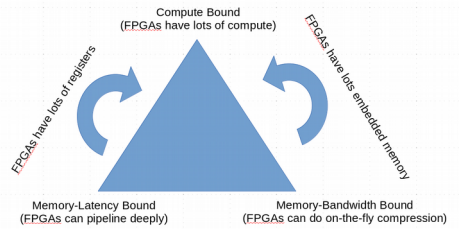
Why? It's a hard problem to route to all of the necessary resources, minimize the path lengths, and make sure all of the signals arrive at exactly the right time.



https://www.allaboutcircuits.com/news/Xilinx-vivado-design-suite-design-implementation/

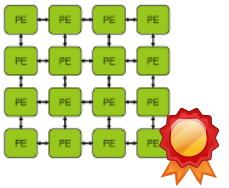
CGRA = Coarse-Grained Reconfigurable Architecture

Spatial architectures are fundamentally attractive, but:

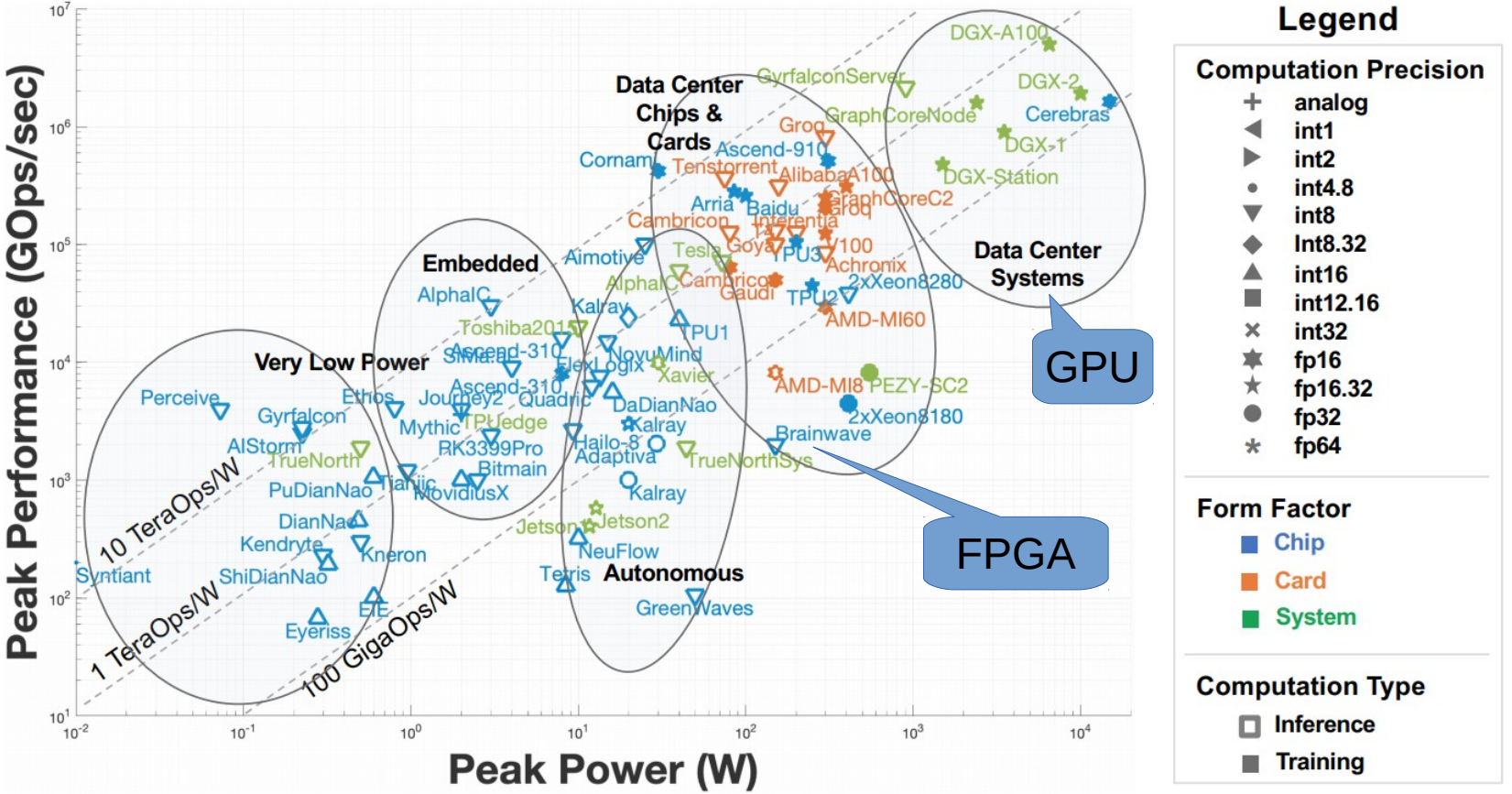


- Must run at a higher clock rate (in practice), to support higher bandwidths, and have builtin FP64 support.
- Probably needs to have a lower routing : logic area ratio.
- Must have a sophisticated memory subsystem, dealing with latency variance and providing for data reuse. Maybe even speculative execution.
- Must have many fewer units, and include sufficient asynchrony, to make place-and-route much easier (and faster)!

And, maybe then, we can beat the GPUs in HPC and ML...



In Case You Missed It... An Explosion in Architecutes for ML



Legend

Computation Precision
- analog
- int1
- int2
- int4.8
- int8
- Int8.32
- int16
- int12.16
- int32
- fp16
- fp16.32
- fp32
- fp64

Form Factor
- Chip
- Card
- System

Computation Type
- Inference
- Training

https://arxiv.org/pdf/2009.00993.pdf

For Specialized Workload, Specialized Architecutres are Ahead

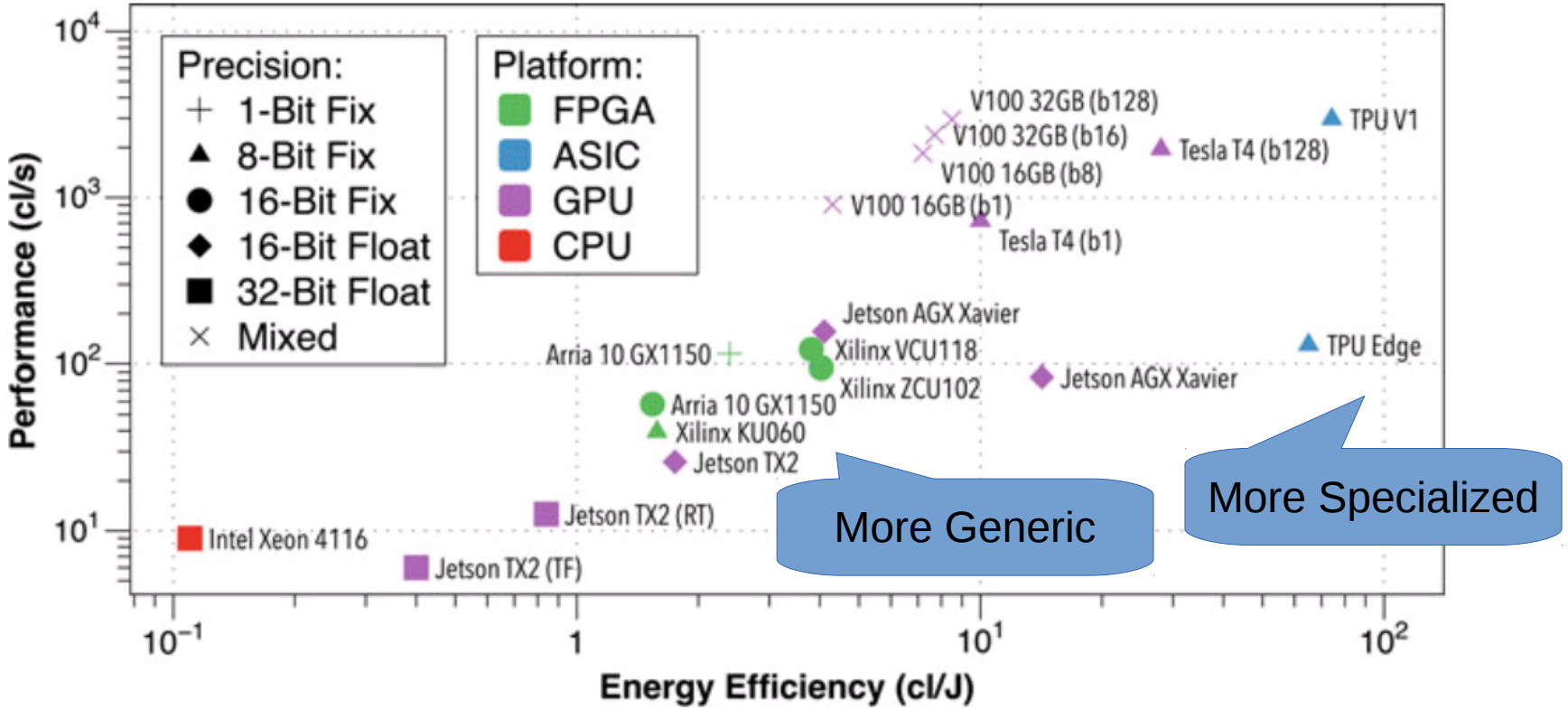


Fig. 12.10 Comparison of VGG-16 implementation results of selected accelerators

Rueckert. NANO-CHIPS 2030, 2020. https://link.springer.com/chapter/10.1007%2F978-3-030-18338-7_12

ML Architecures, Generically, Are Temporal / Spatial

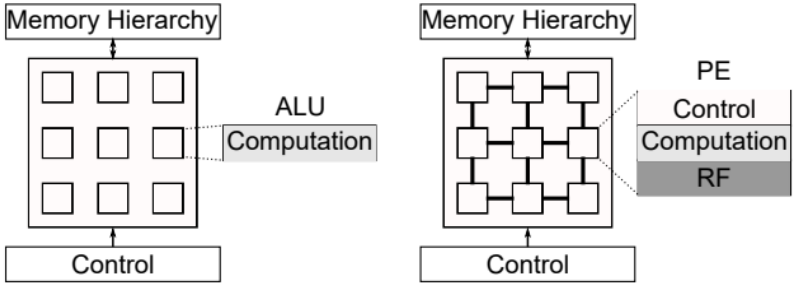


Figure 6. Comparison between the temporal and spatial architectures.

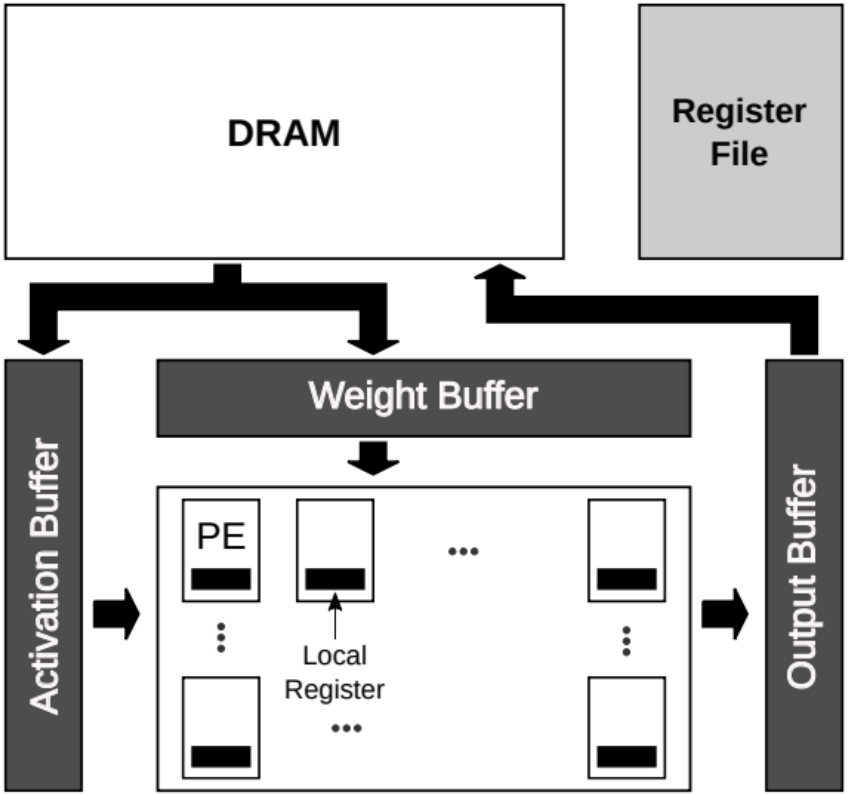


Figure 10. Generic memory hierarchy for a DNN accelerator.

Capra, et al. Future Internet 2020 - https://www.mdpi.com/1999-5903/12/7/113/pdf

And There Is A Lot of Variation in Design...

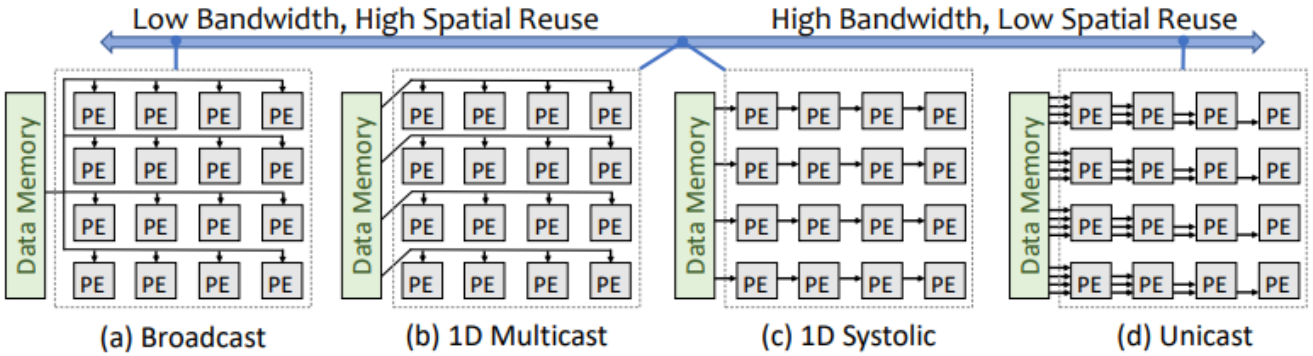


Fig. 13. Common NOC designs (Figure adopted from [63]).

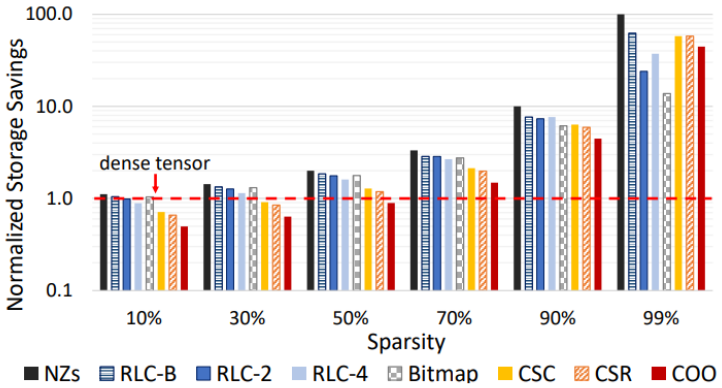


Fig. 6. Storage benefits for encoding a sparse tensor (512×2048 matrix with 16b elements) in different formats, normalized to the size of the fully dense tensor (Figure inspired by [144]).

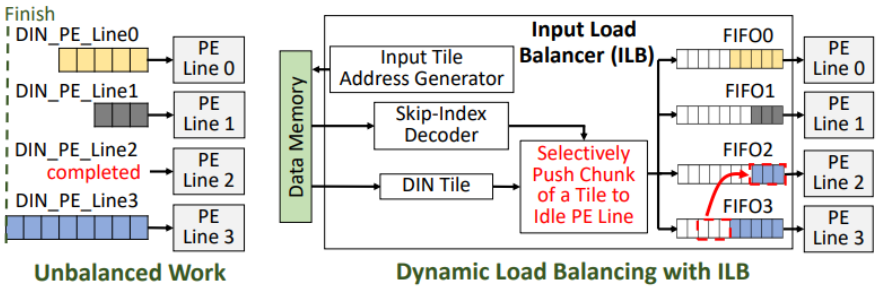


Fig. 24. Load balance mechanism in LNPU [97] (Figure adopted from [97]).

https://arxiv.org/pdf/2007.00864.pdf

And There Is A Lot of Variation in Design… Because Headroom Exists

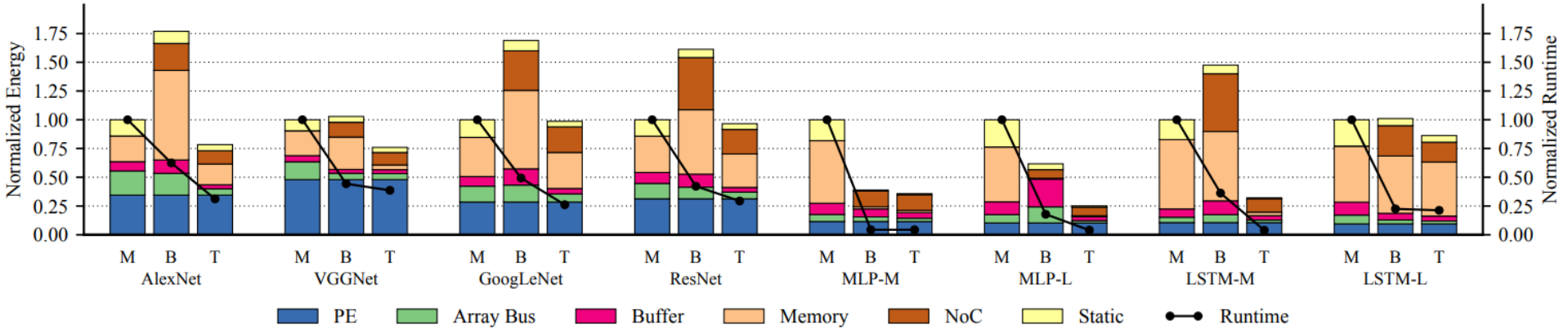


Figure 11. Comparison between monolithic architecture, baseline tiled architecture, and TANGRAM. All three designs use 16384 PEs and 8 MB buffers. In the two tiled architectures resources are organized into 256 tiled engines (16 × 16).

https://web.stanford.edu/~mgao12/pubs/tangram.asplos19.pdf

And The Real Archtectures Are Interesting...

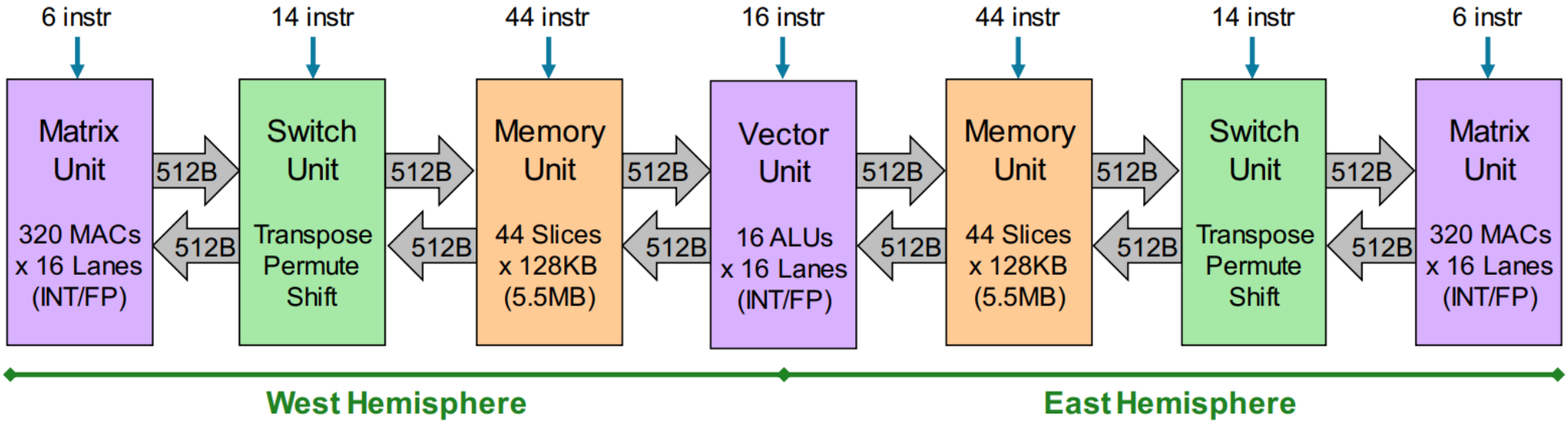


Figure 2. TSP superlane block diagram. Every superlane is bilaterally symmetric with an east side and a west side. It contains 16 lanes, each of which is 8 bits wide. Data flows from east to west and from west to east.

https://groq.com/wp-content/uploads/2020/04/Groq-Rocks-NNs-Linley-Group-MPR-2020Jan06.pdf

The Groq TSP "Superlane" Low-Level Pipeline

Trade Offs in Coarseness

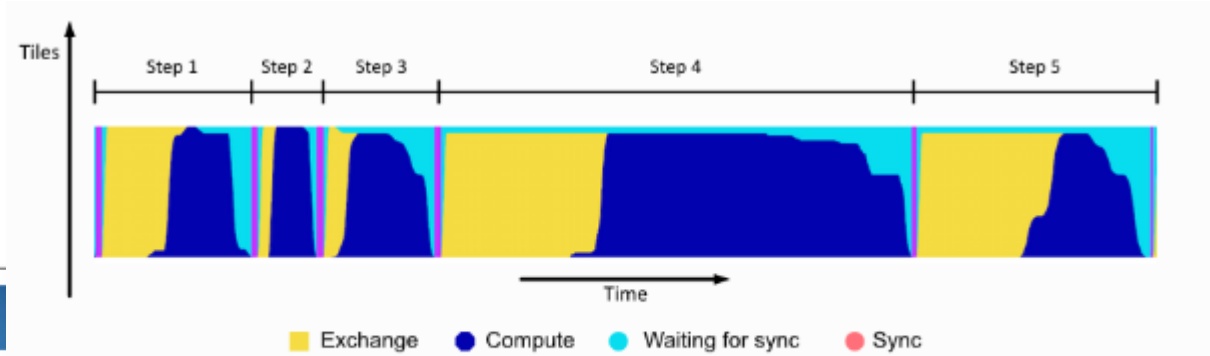


Fig. 3.6 Execution activity of all tiles in IPU

https://docs.graphcore.ai/projects/ipu-overview/en/latest/programming_model.html

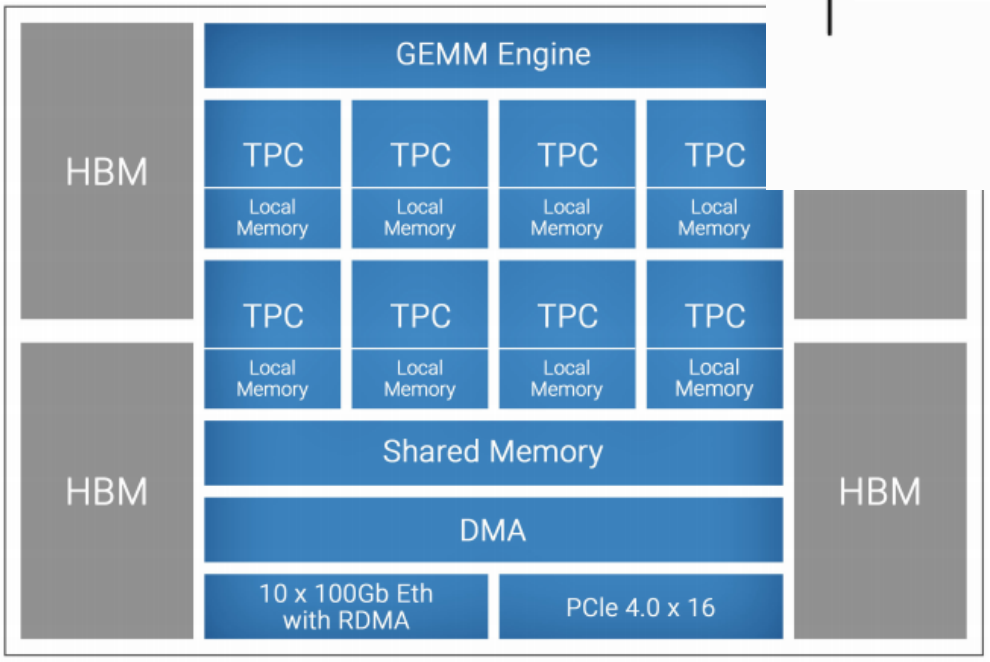


Figure 2: Gaudi High-level Architecture

https://habana.ai/wp-content/uploads/2019/06/Habana-Gaudi-Training-Platform-whitepaper.pdf

Coarseness in time

Coarseness in space

Combining Fine and Coarse – Intermixed, Close By, or Both

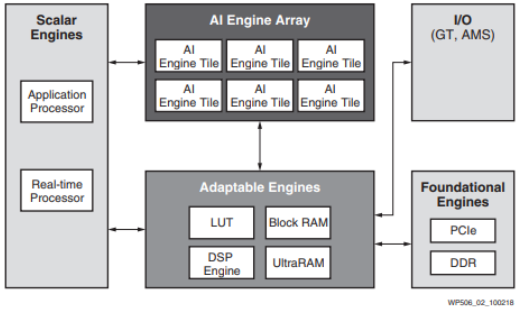


Figure 2: **Heterogeneous Compute**

Figure 3 illustrates the composition of AI Engine interface tiles into a 2D array.

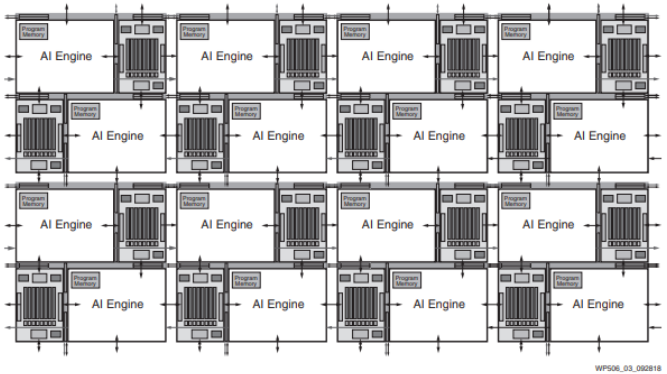


Figure 3: **AI Engine Array**

- Coarse PE tiles near the FPGA fabric
- DSP blocks intermixed in the FPGA fabric

Xilinx Versal

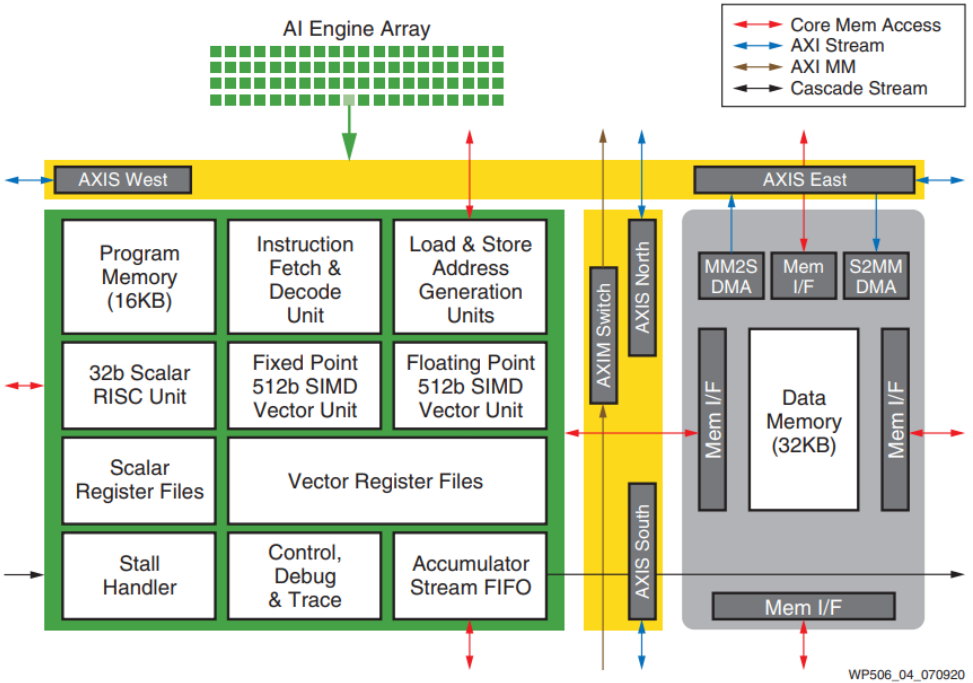
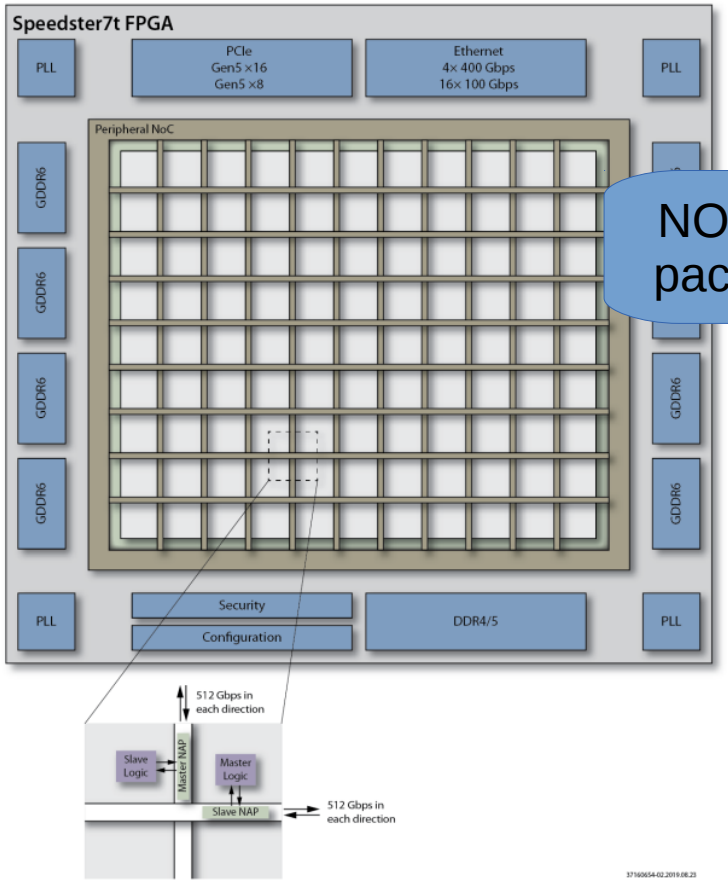


Figure 4: **Detail of AI Engine Tile**

https://www.xilinx.com/support/documentation/white_papers/wp506-ai-engine.pdf

Builtin NOCs, Memory, I/O, etc.



NOC supporting packet switching

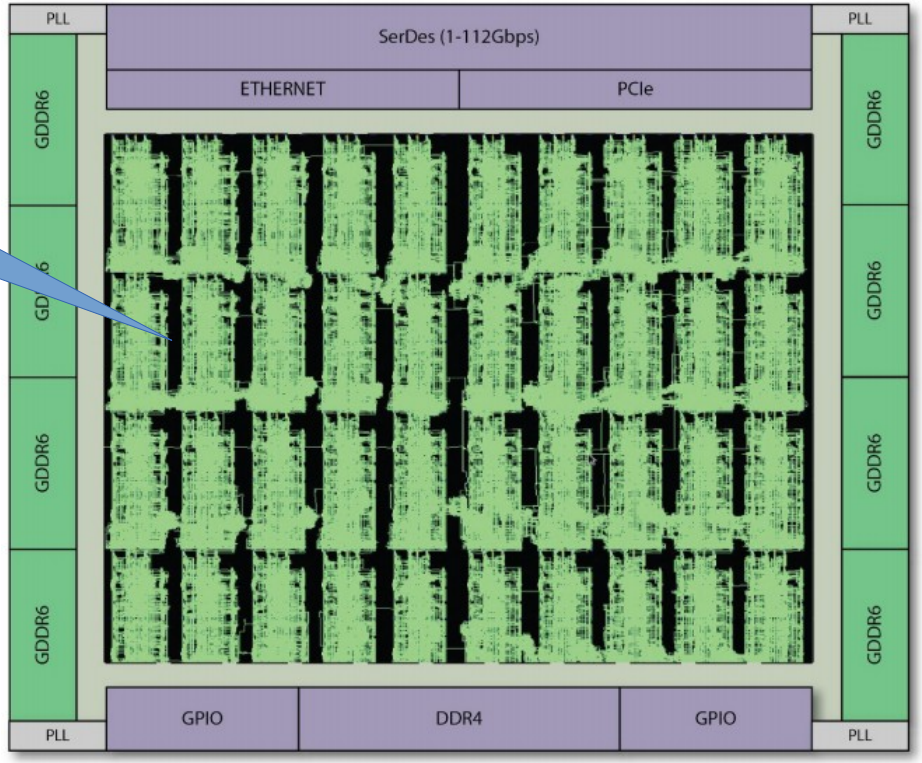


Figure 9: *A Speedster7t Device with 40 Instances of a 2D Convolution Block*

https://www.achronix.com/sites/default/files/docs/Eight_Benefits_of_Using_an_FPGA_with_an_On-chip_High-Speed_Network_WP020.pdf

What's Next?

- FPGAs and CGRAs are interesting and promising.
- ML accelerators have generated a new class of special architectures.
- To address scientific computing / HPC applications of the future, we should explore how these architectures can be used and enhanced.

➔ ALCF is supported by DOE/SC under contract DE-AC02-06CH11357