Dark Silicon & Modern Computer Architecture

Hung-Wei Tseng

Power v.s. Energy

- Power is the direct contributor of "heat"
  - Packaging of the chip
  - Heat dissipation cost
- Energy = P * ET
  - The electricity bill and battery life is related to energy!
  - Lower power does not necessary means better battery life if the processor slow down the application too much

Static/Leakage Power

- The power consumption due to leakage — transistors do not turn all the way off during no operation
- Becomes the **dominant** factor in the most advanced process technologies.

$$P_{leakage} \sim N \times V \times e^{-V_t}$$

  - N: number of transistors
  - V: voltage
  - V_t: threshold voltage where transistor conducts (begins to switch)

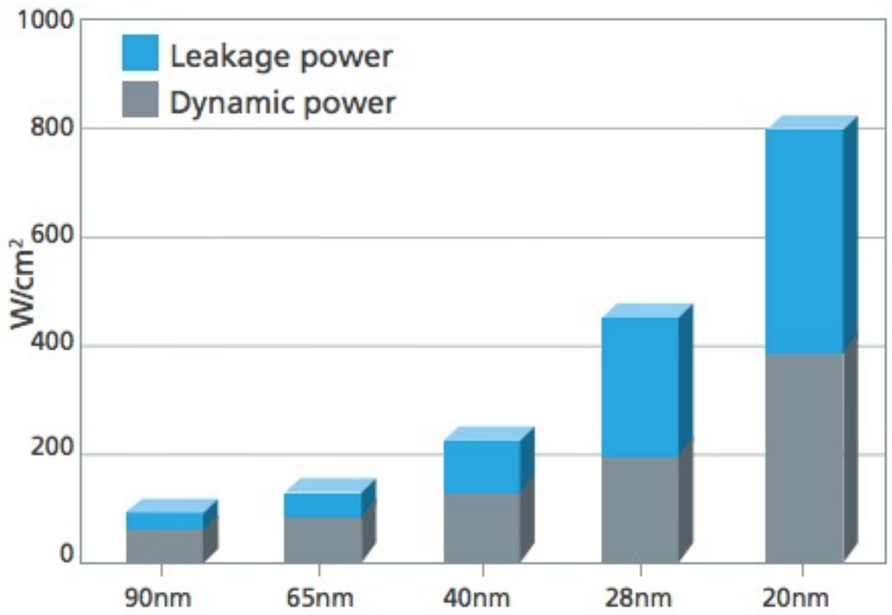


Figure 1: Leakage power becomes a growing problem as demands for more performance and functionality drive chipmakers to nanometer-scale process nodes (Source: IBS).

Dennardian Broken

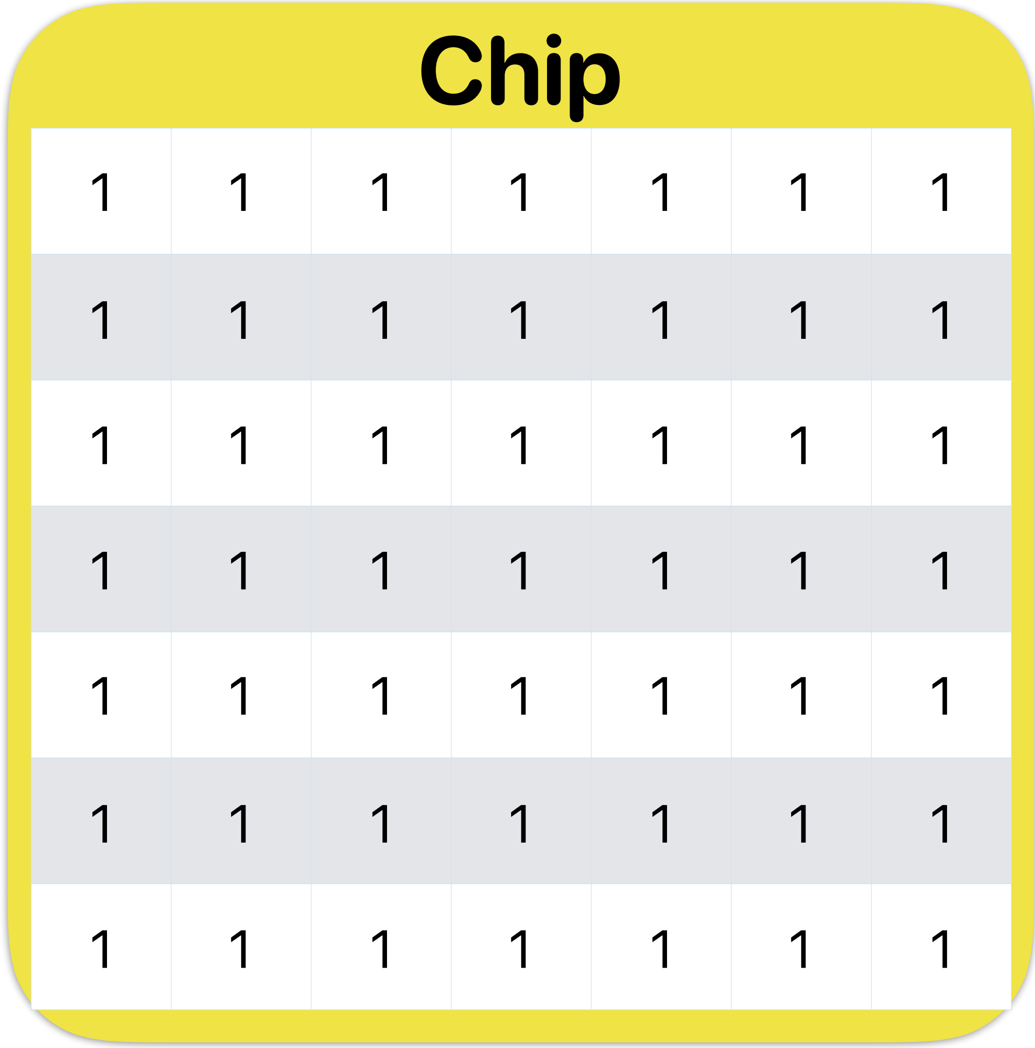
- Given a scaling factor S

Parameter	Relation	Classical Scaling	Leakage Limited
Power Budget		1	1
Chip Size		1	1
Vdd (Supply Voltage)		1/S	1
Vt (Threshold Voltage)	1/S	1/S	1
tex (oxide thickness)		1/S	1/S
W, L (transistor		1/S	1/S
Cgate (gate capacitance)	WL/tox	1/S	1/S
Isat (saturation current)	WVdd/tox	1/S	1
F (device frequency)	Isat/(CgateVdd)	S	S
D (Device/Area)	1/(WL)	S^2	S^2
p (device power)	IsatVdd	$1/S^2$	1
P (chip power)	Dp	1	S^2
U (utilization)	1/P	1	$1/S^2$

Dark Silicon and the End of Multicore Scaling

H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam and D. Burger
University of Washington, University of Wisconsin—Madison, University of Texas at Austin, Microsoft Research

Power consumption to light on all transistors

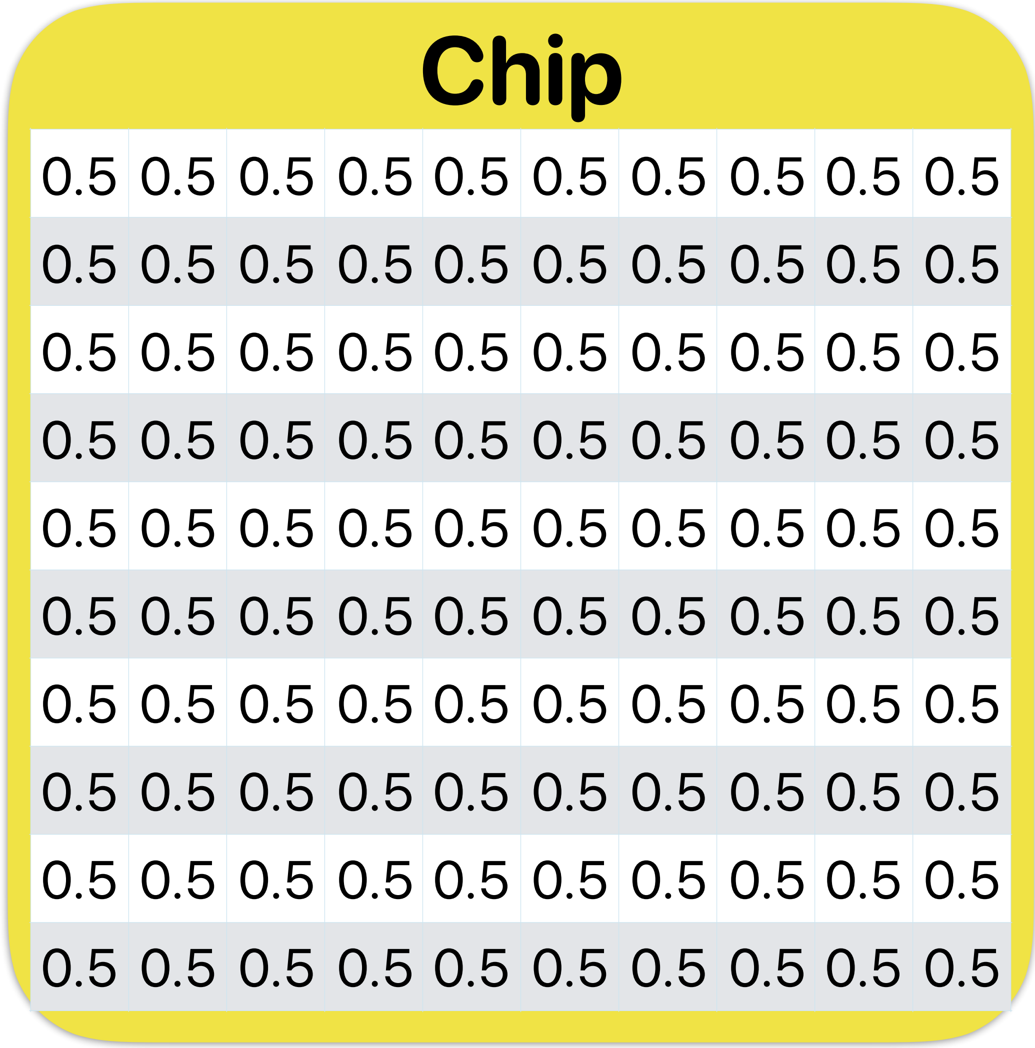


Chip

1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1

=49W

Dennardian Scaling

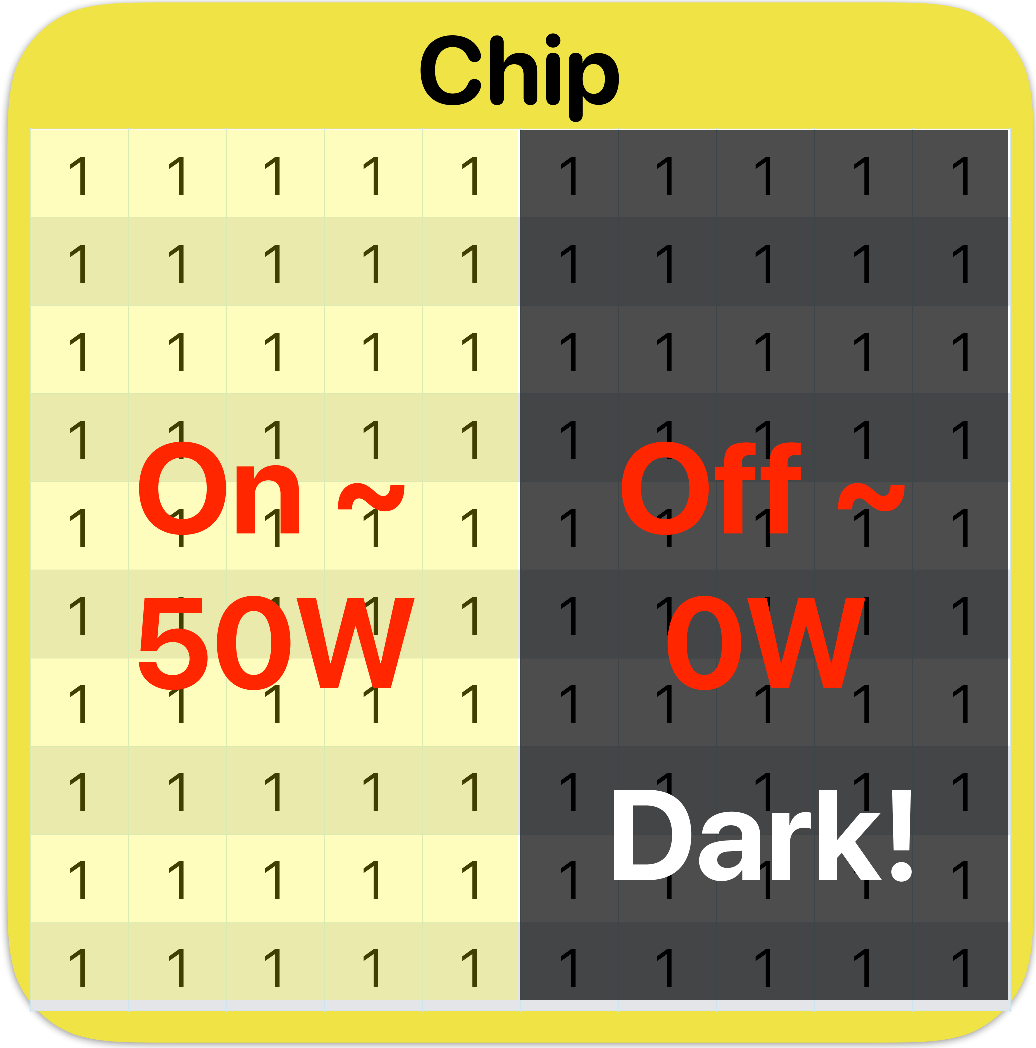


Chip

0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

=50W

Dennardian Broken



Chip

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

On ~ 50W

Off ~ 0W

Dark!

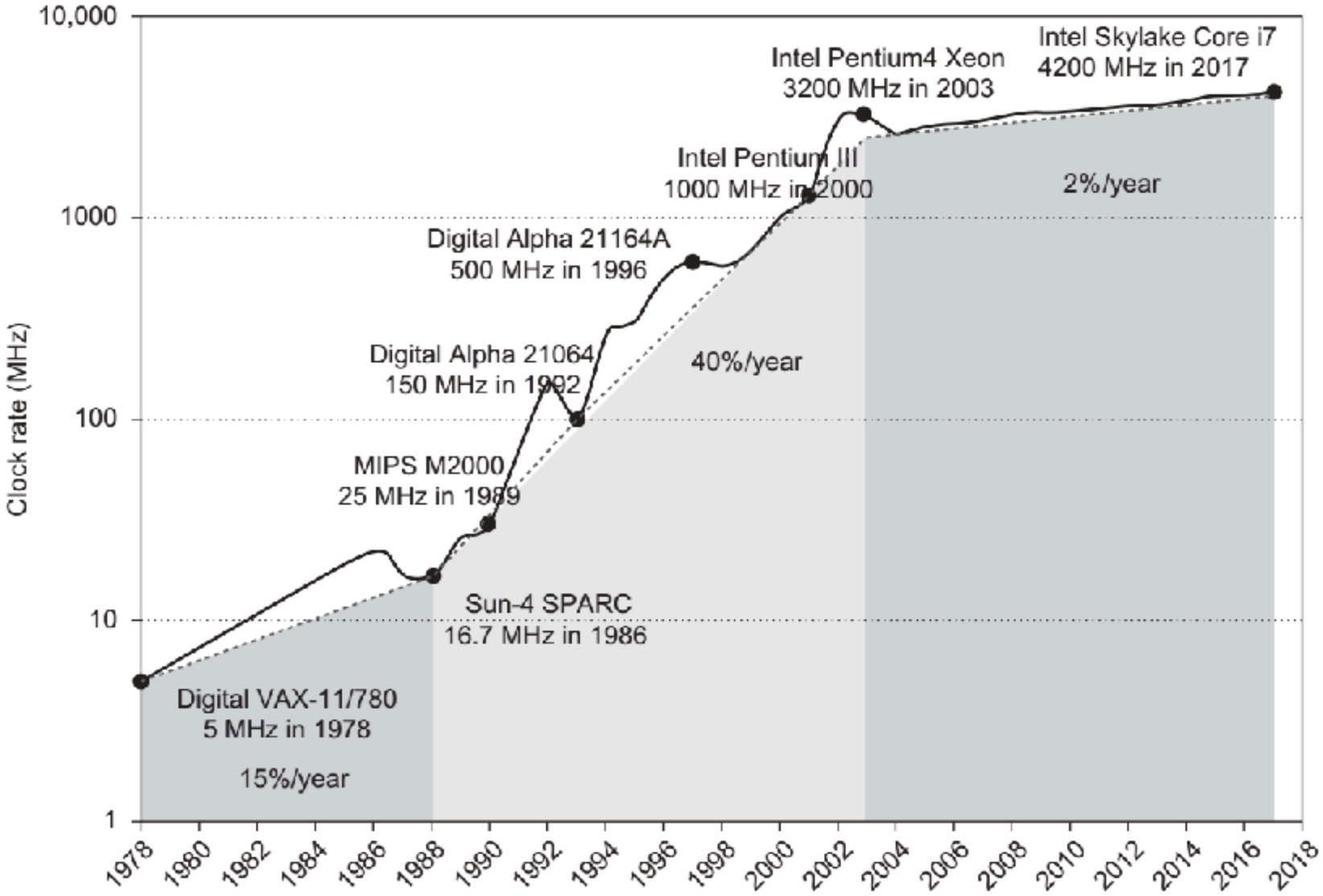
=100W!

What happens if power doesn't scale with process technologies?

- If we are able to cram more transistors within the same chip area (Moore's law continues), but the power consumption per transistor remains the same. Right now, if we power the chip with the same power consumption but put more transistors in the same area because the technology allows us to. How many of the following statements are true?
  - ① The power consumption per chip will increase
  - ② The power density of the chip will increase
  - ③ Given the same power budget, we may not able to power on all chip area if we maintain the same clock rate
  - ④ Given the same power budget, we may have to lower the clock rate of circuits to power on all chip area

  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Clock rate improvement is limited nowadays



Solutions/trends in dark silicon era

Aggressive dynamic frequency scailing

More cores per chip, slower per core

Products Solutions Support

	Intel® Xeon® Processor E7-8890 v4	Intel® Xeon® Processor E7-8893 v4	Intel® Xeon® Processor E7-8880 v4
Status	Launched	Launched	Launched
Launch Date	Q2'16	Q2'16	Q2'16
Lithography	14 nm	14 nm	14 nm
Performance			
# of Cores	24	4	22
# of Threads	48	8	44
Processor Base Frequency	2.20 GHz	3.20 GHz	2.20 GHz
Max Turbo Frequency	3.40 GHz	3.50 GHz	3.30 GHz
Cache	60 MB	60 MB	55 MB
Bus Speed	9.6 GT/s	9.6 GT/s	9.6 GT/s
# of QPI Links	3	3	3
TDP	165 W	140 W	150 W

Demo

- You may use cat /proc/cpuinfo to see all the details of your processors
- You may add "| grep MHz" to see the frequencies of your cores
- Only very few of them are on the boosted frequency

Slower, but more

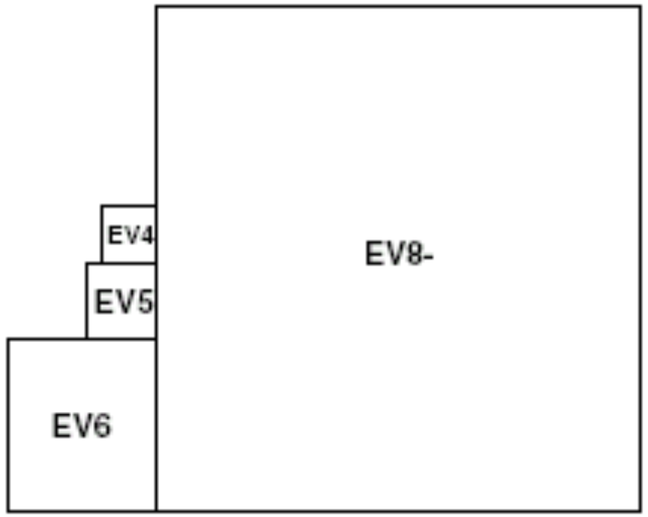
Single-ISA Heterogeneous Multi-Core Architectures: The Potential for Processor Power Reduction

Rakesh Kumar, Keith Farkas, Norm P. Jouppi, Partha Ranganathan, Dean M. Tullsen.
University of California, San Diego and HP Labs

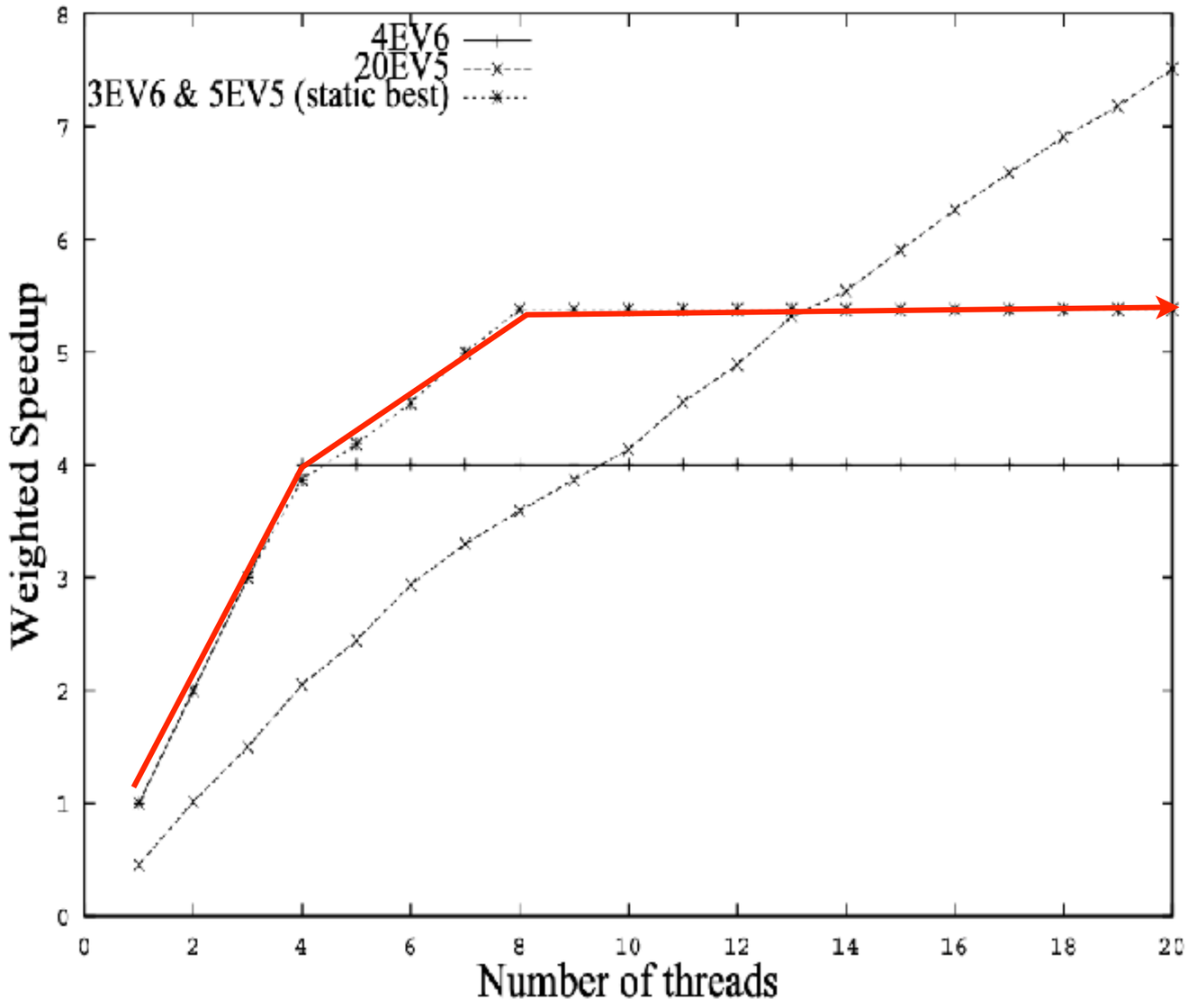
Areas of different processor generations

- You fit about 5 EV5 cores within the same area of an EV6
- If you build a quad-core EV6, you can use the same area to
  - build 20-core EV5
  - 3EV6+5EV5

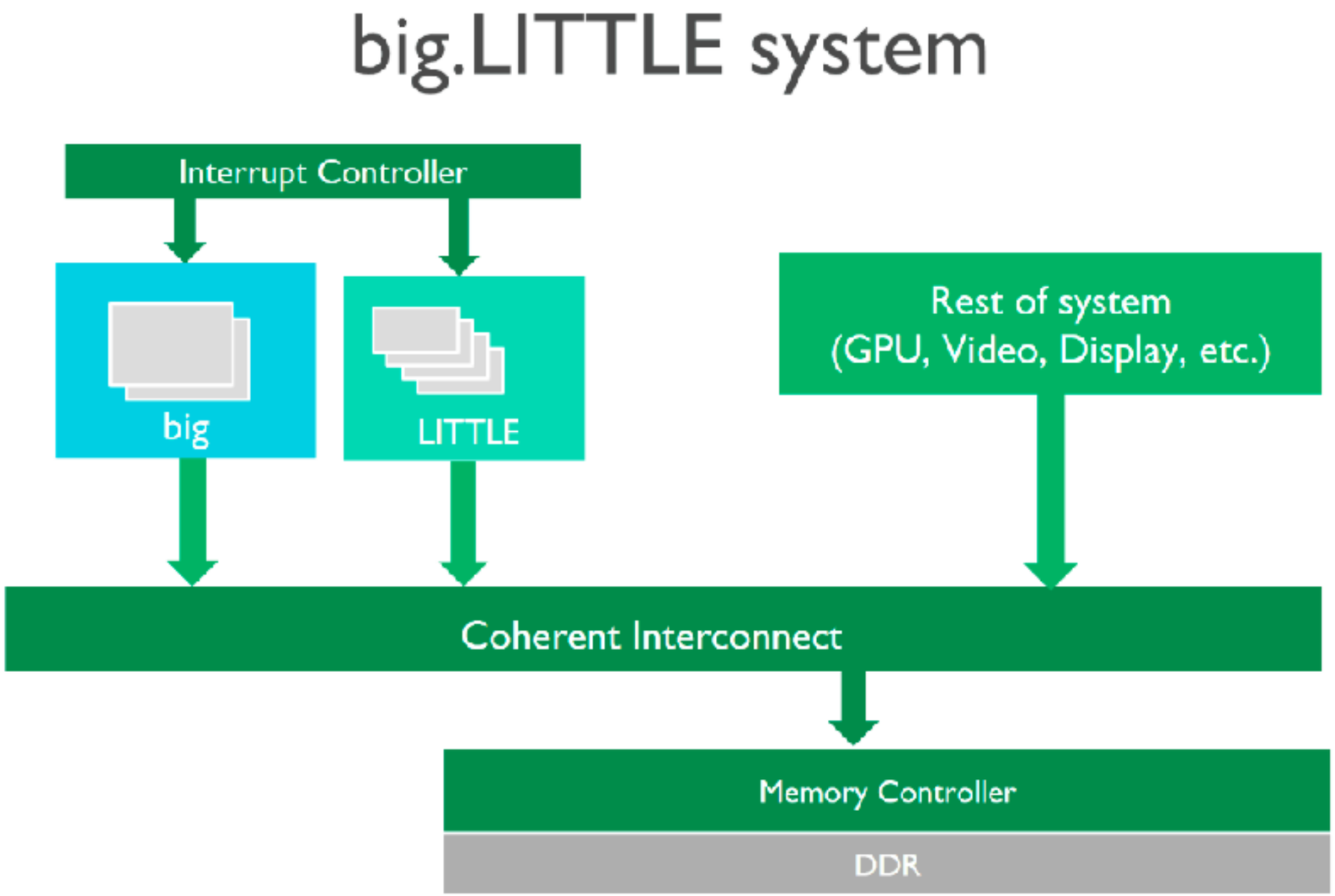
Processor	EV5	EV6	EV6+
Issue-width	4	6 (OOO)	6 (OOO)
I-Cache	8KB, DM	64KB, 2-way	64KB, 2-way
D-Cache	8KB, DM	64KB, 2-way	64KB, 2-way
Branch Pred.	2K-gshare	hybrid 2-level	hybrid 2-level
Number of MSHRs	4	8	16
Number of threads	1	1	4
Area (in mm^2)	5.06	24.5	29.9



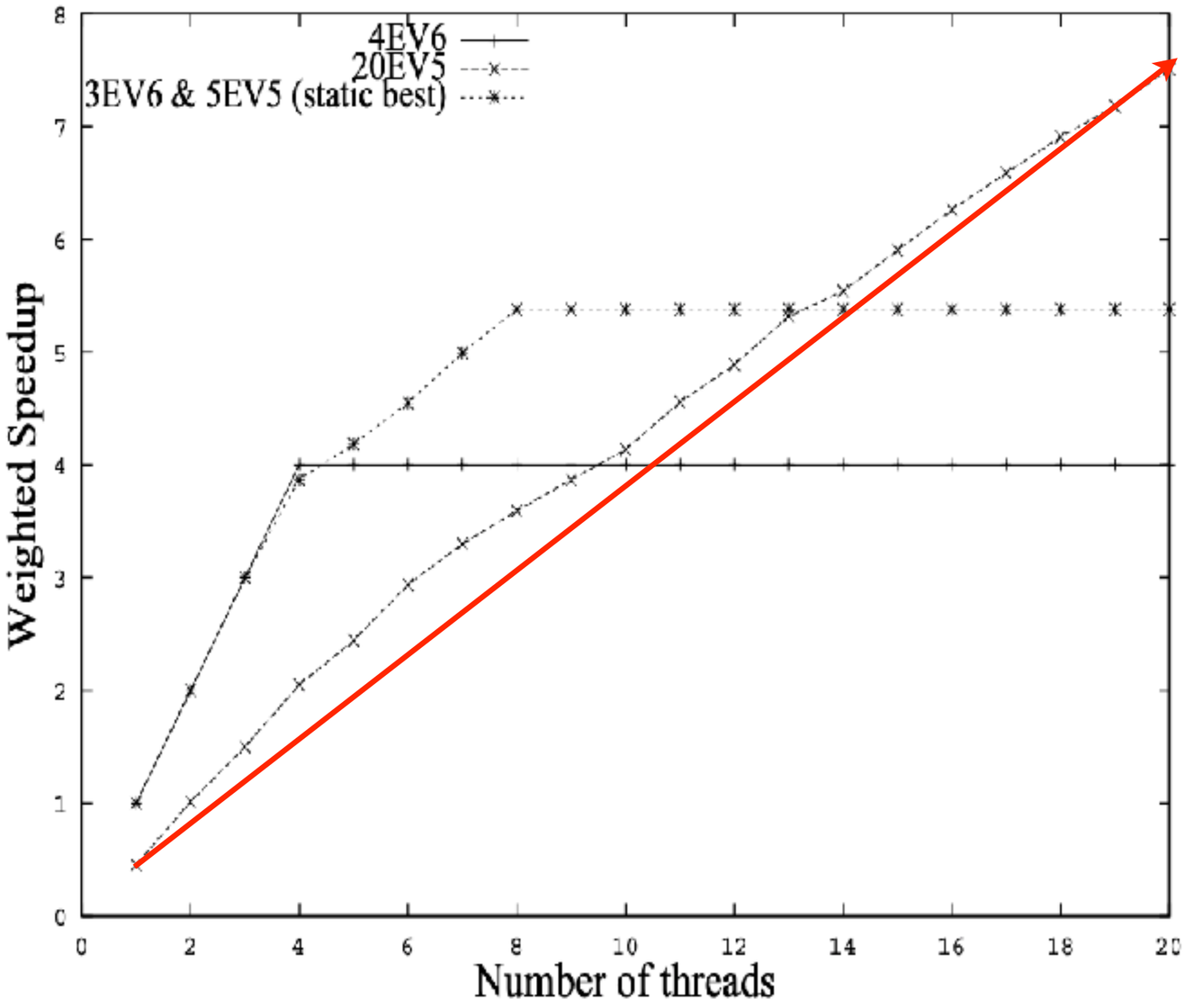
4EV6 v.s. 20 EV5 v.s. 3EV6+5EV5



ARM's big.LITTLE architecture



4EV6 v.s. 20 EV5 v.s. 3EV6+5EV5



Xeon Phi

Essentials

Product Collection	Intel® Xeon Phi™ 72x5 Processor Family
Code Name	Products formerly Knights Mill
Vertical Segment	Server
Processor Number	7295
Off Roadmap	No
Status	Launched
Launch Date	Q4'17
Lithography	14 nm

Performance

# of Cores	72
# of Threads	72
Processor Base Frequency	1.50 GHz
Max Turbo Frequency	1.50 GHz
Cache	36 MB L2 Cache
TDP	320 W

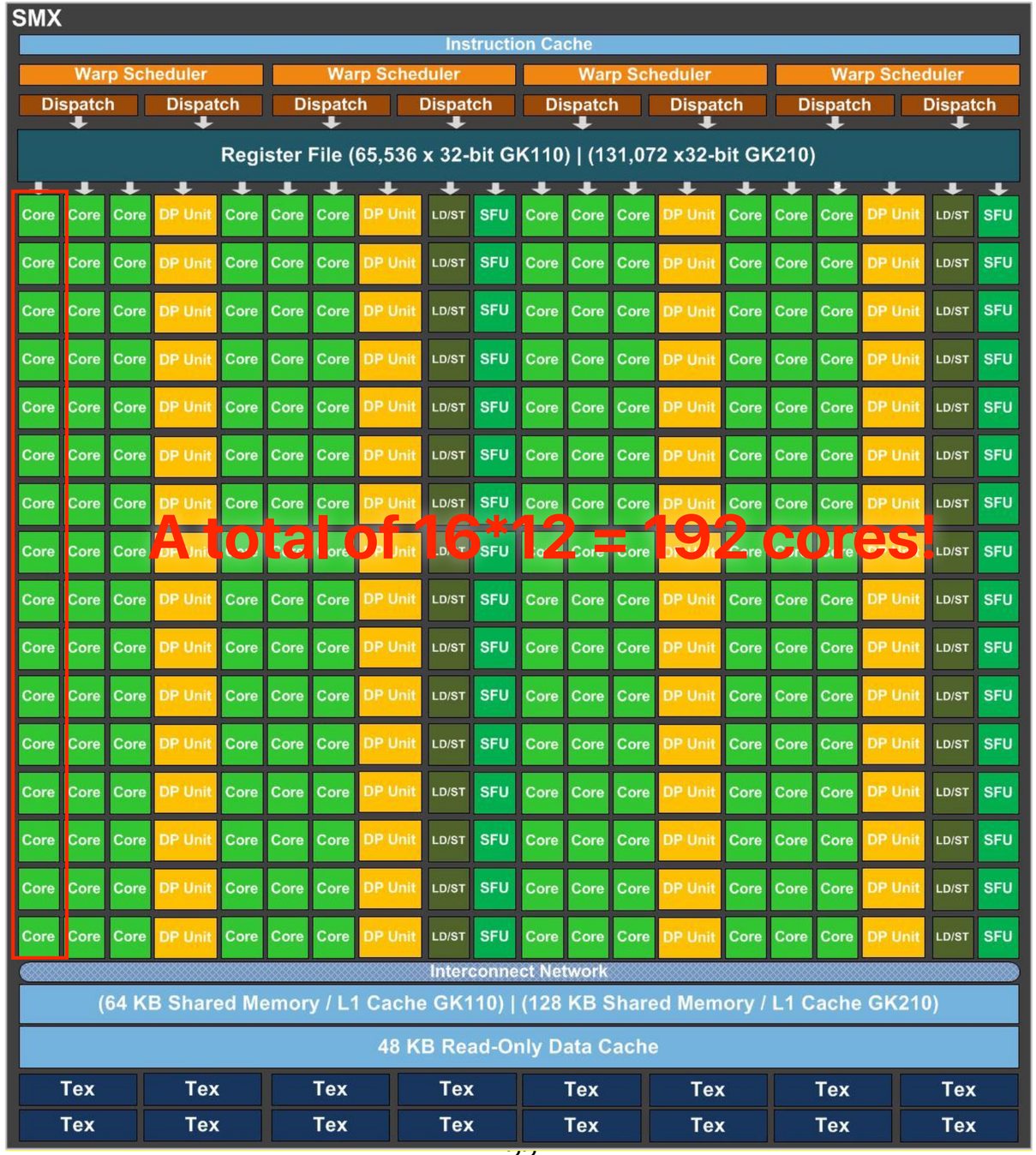
The rise of GPU

Thread scheduler

GPU global memory

High-bandwidth memory controllers

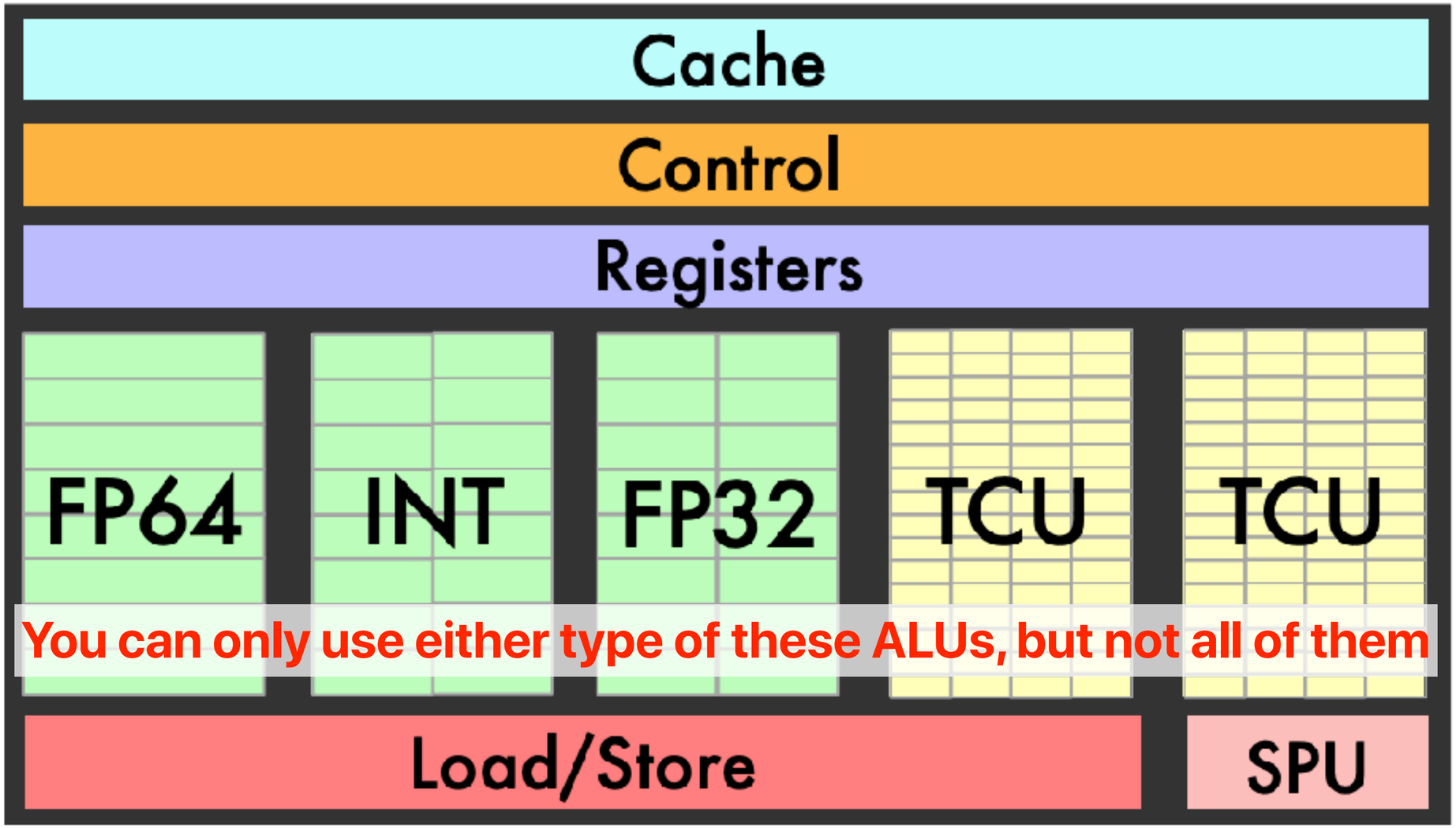
Each of these performs the same operation, but each of these is also a "thread"



A total of 16*12 = 192 cores!

Just let it dark

NVIDIA's Turing Architecture



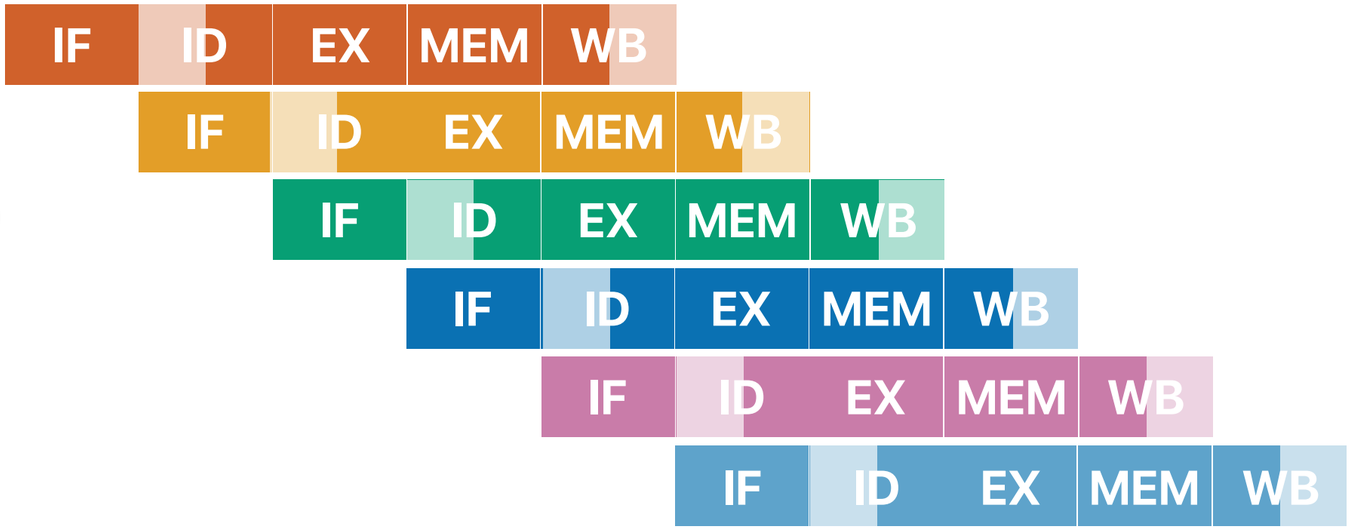
You can only use either type of these ALUs, but not all of them

The rise of ASICs

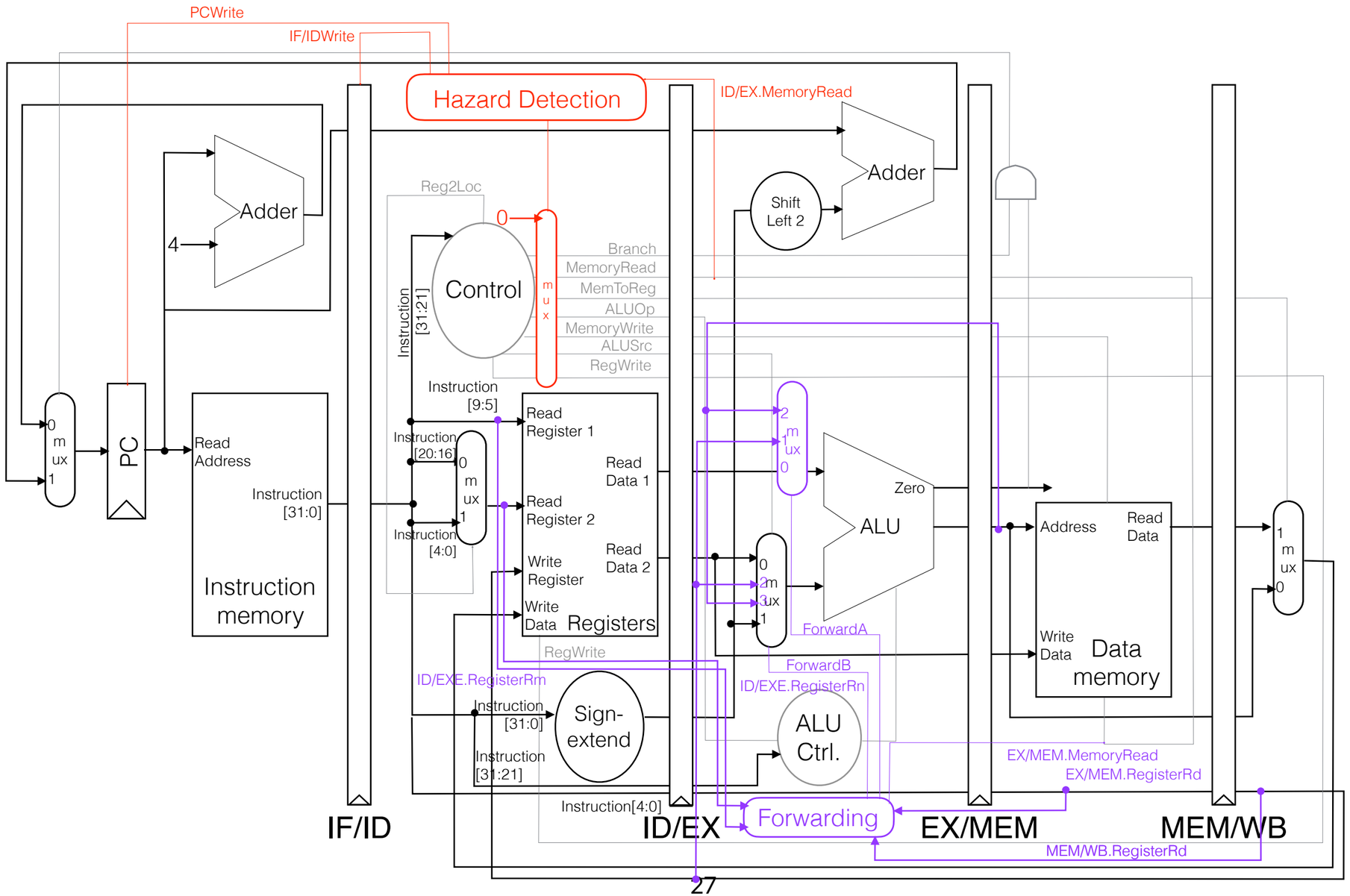
Say, we want to implement a[i] += a[i+1]*20

- This is what we need in RISC-V in each iteration

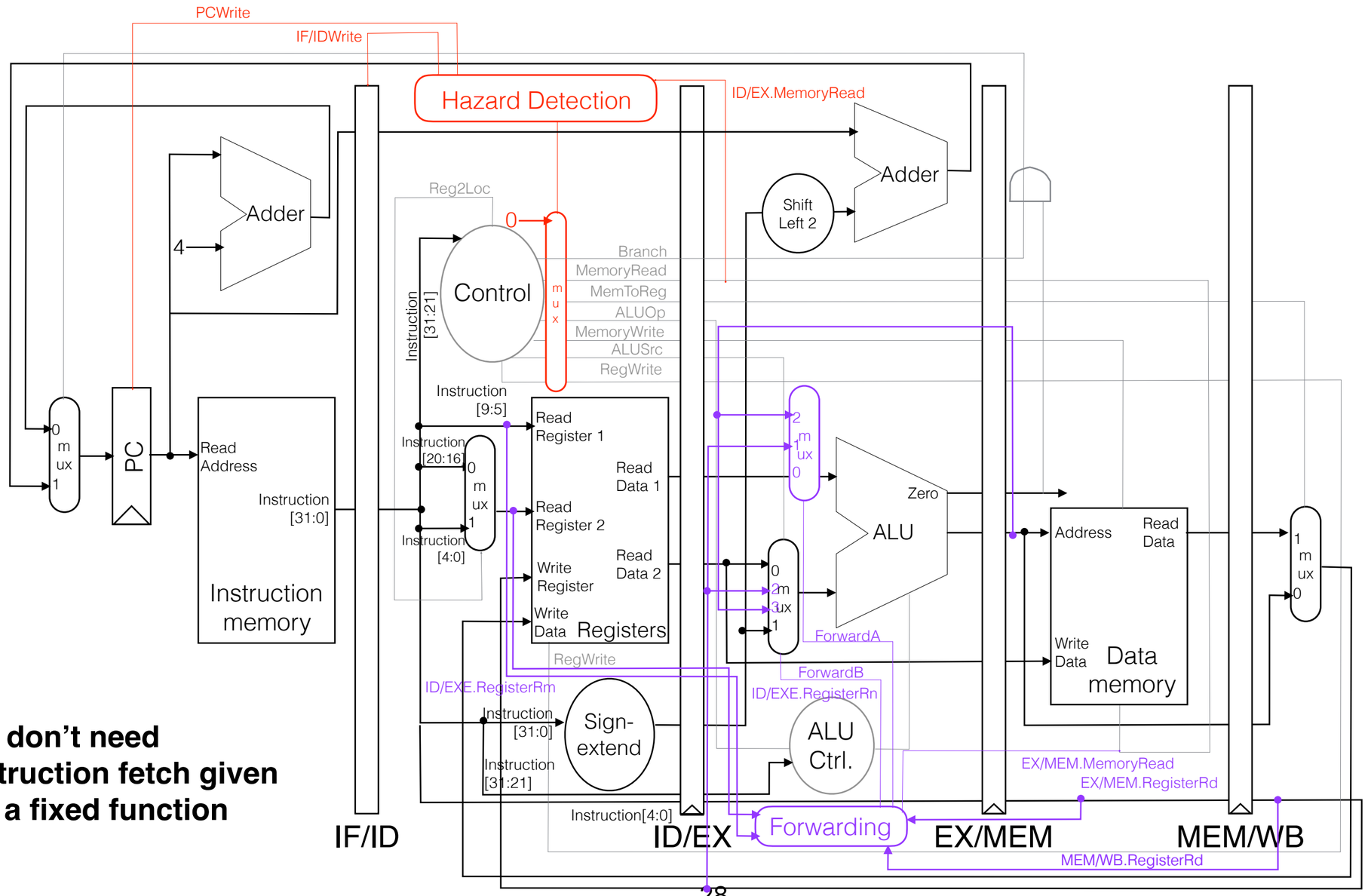
```
ld    X1, 0(X0)
ld    X2, 8(X0)
add   X3, X31, #20
mul   X2, X2, X3
add   X1, X1, X2
sd    X1, 0(X0)
```



This is what you need for these instructions

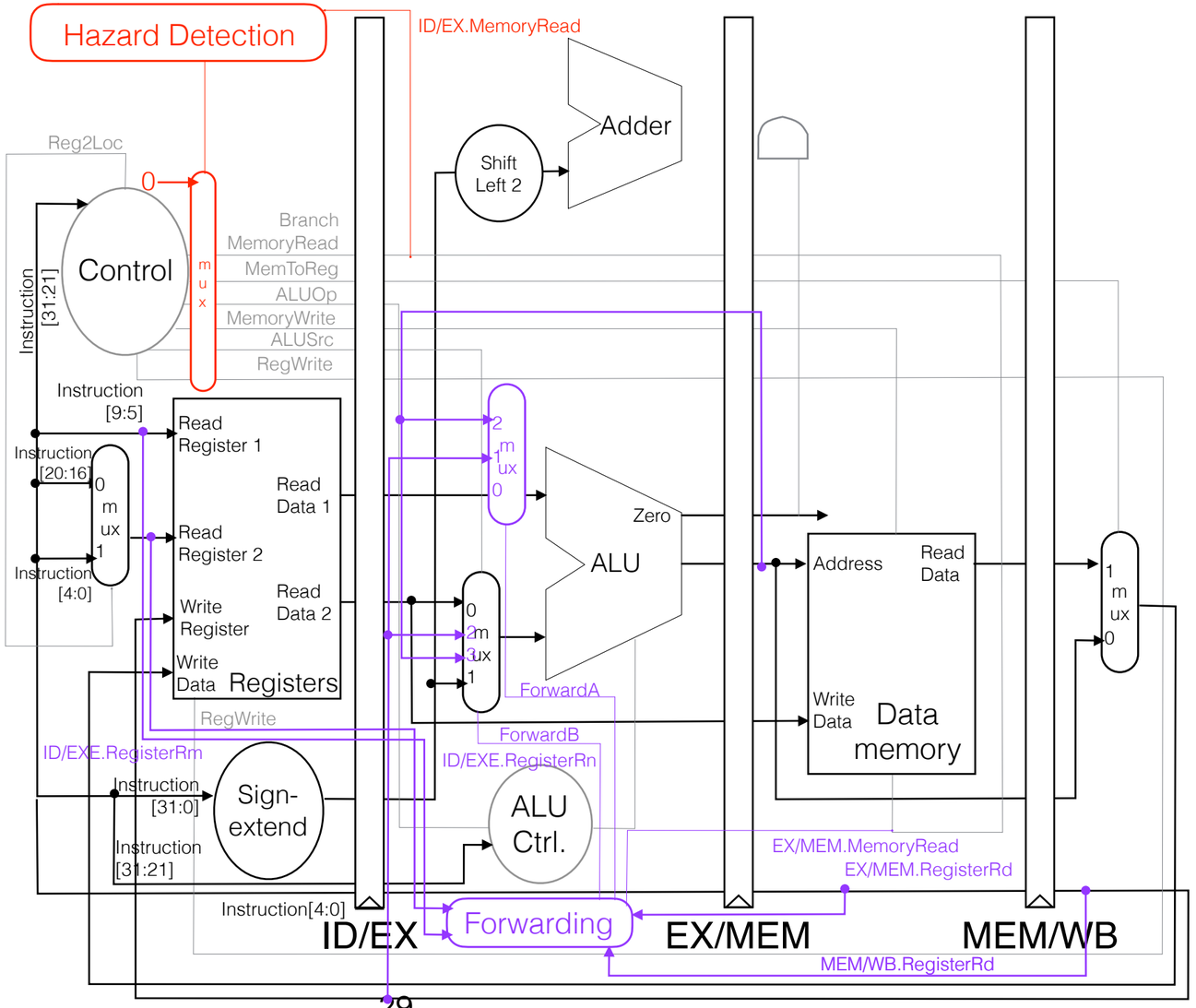


Specialize the circuit



We don't need instruction fetch given it's a fixed function

Specialize the circuit



We don’t need these many registers, complex control, decode

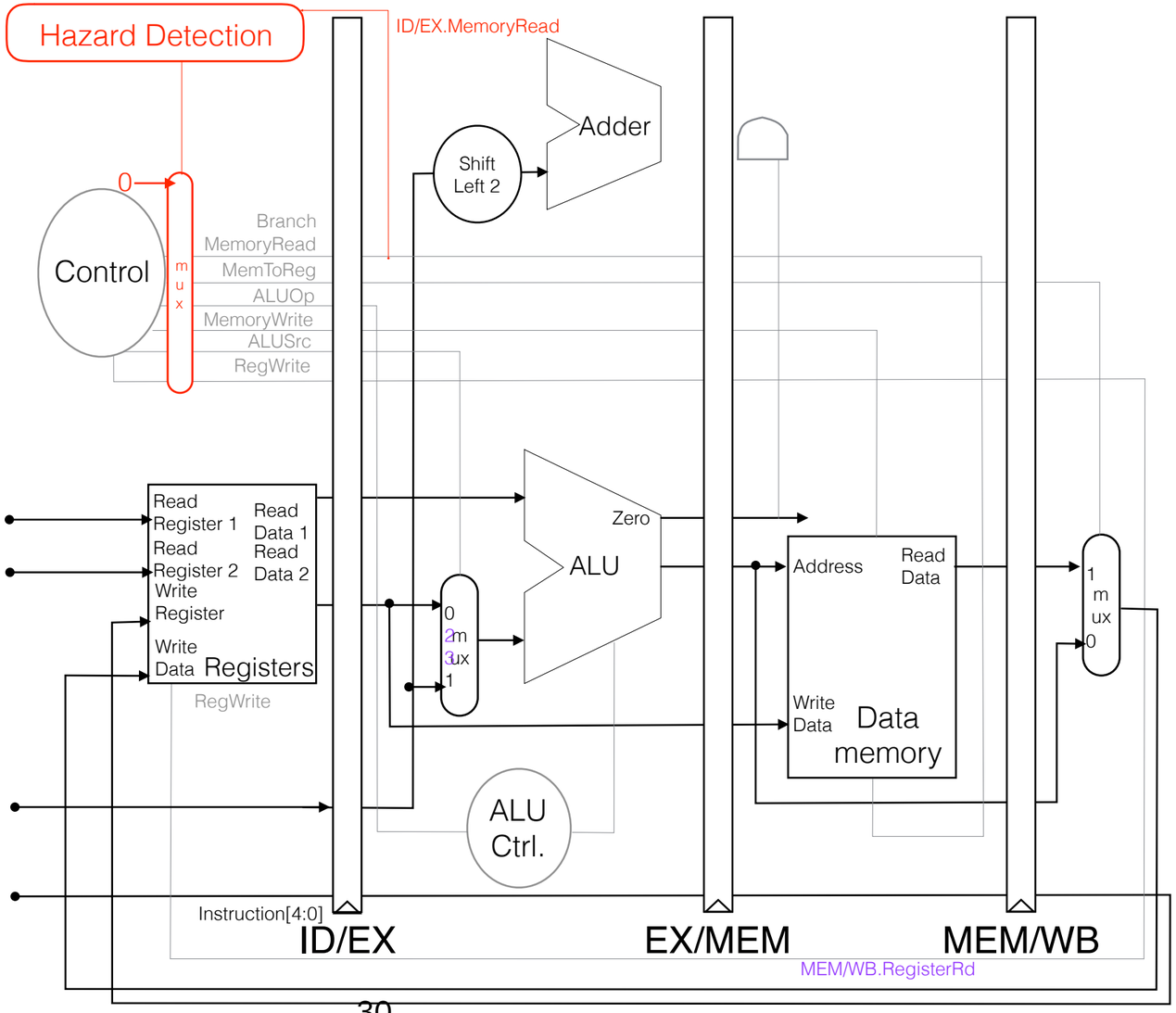
We don’t need instruction fetch given it’s a fixed function

Specialize the circuit

We don't need ALUs, branches, hazard detections…

We don't need these many registers, complex control, decode

We don't need instruction fetch given it's a fixed function

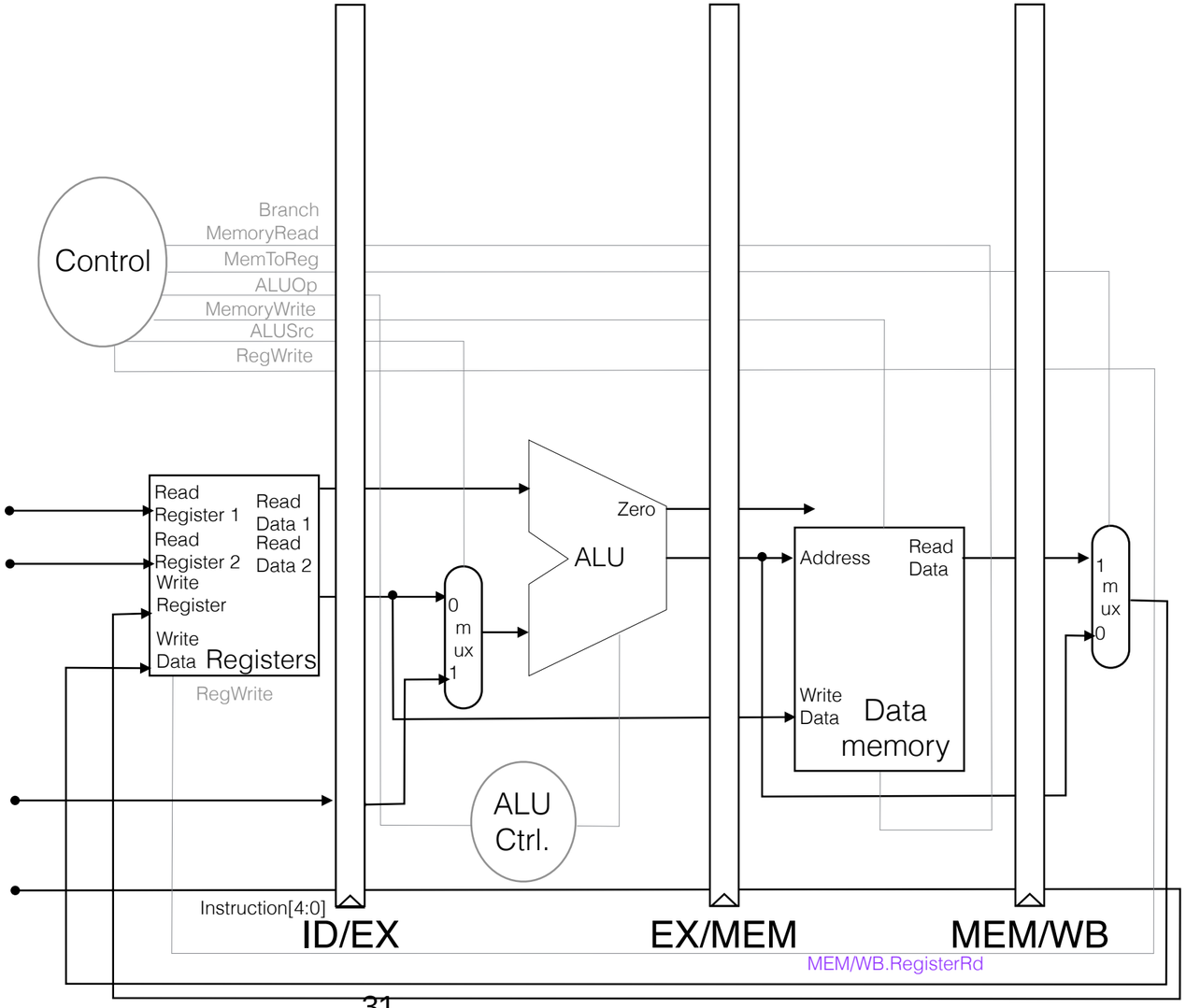


Specialize the circuit

We don't need big ALUs, branches, hazard detections…

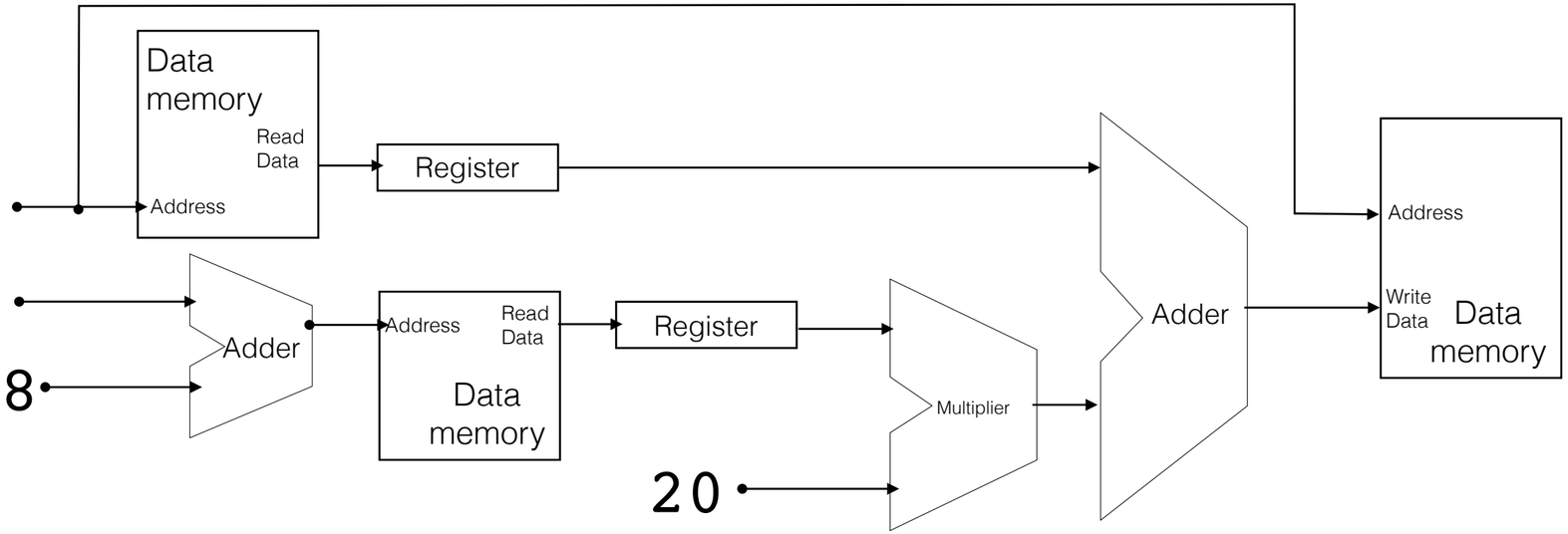
We don't need these many registers, complex control, decode

We don't need instruction fetch given it's a fixed function



Rearranging the datapath

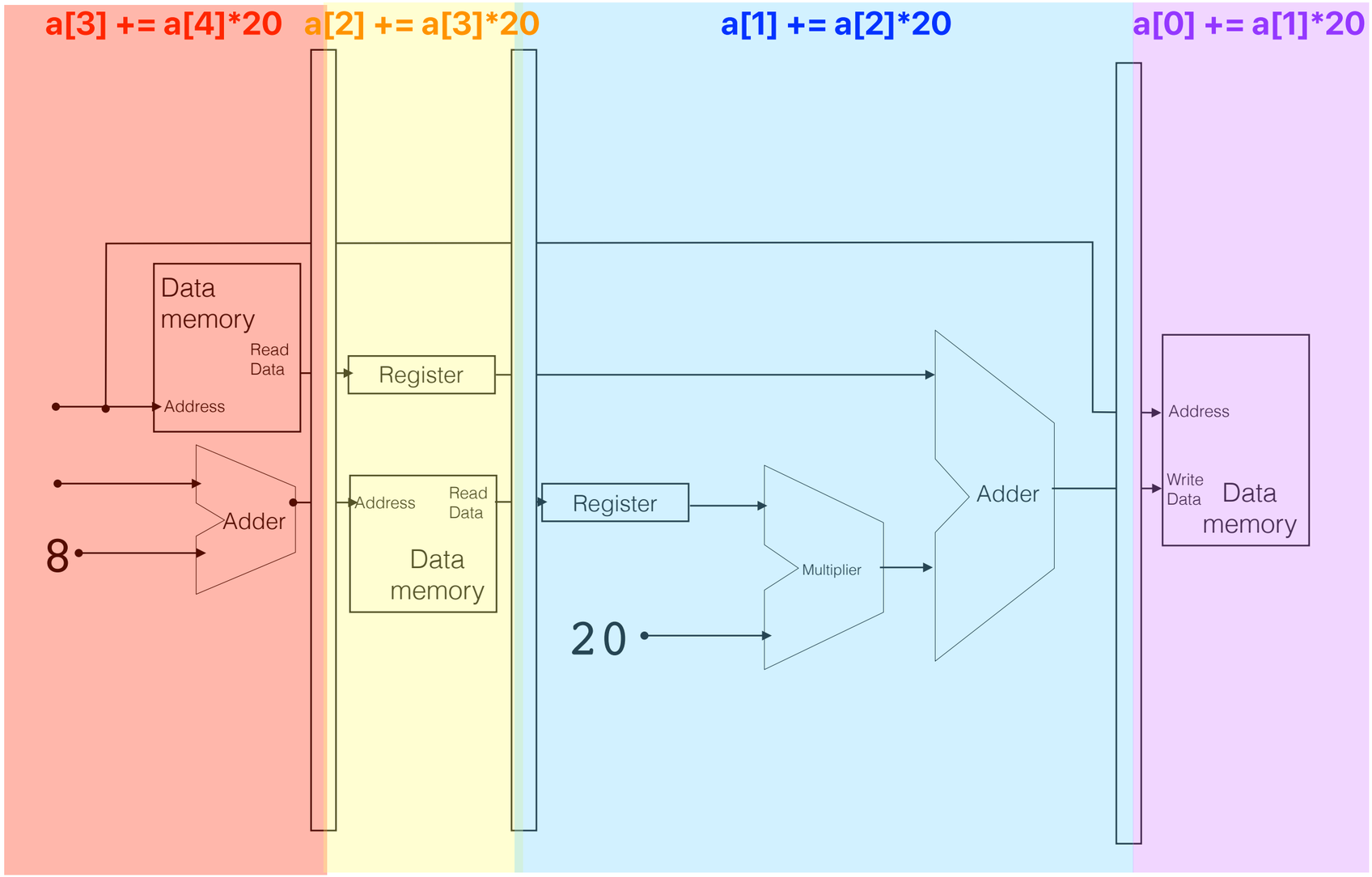
```
ld    X1, 0(X0)
ld    X2, 8(X0)
add   X3, X31, #20
mul   X2, X2, X3
add   X1, X1, X2
sd    X1, 0(X0)
```



The pipeline for a[i] += a[i+1]*20

Each stage can still be as fast as the pipelined processor

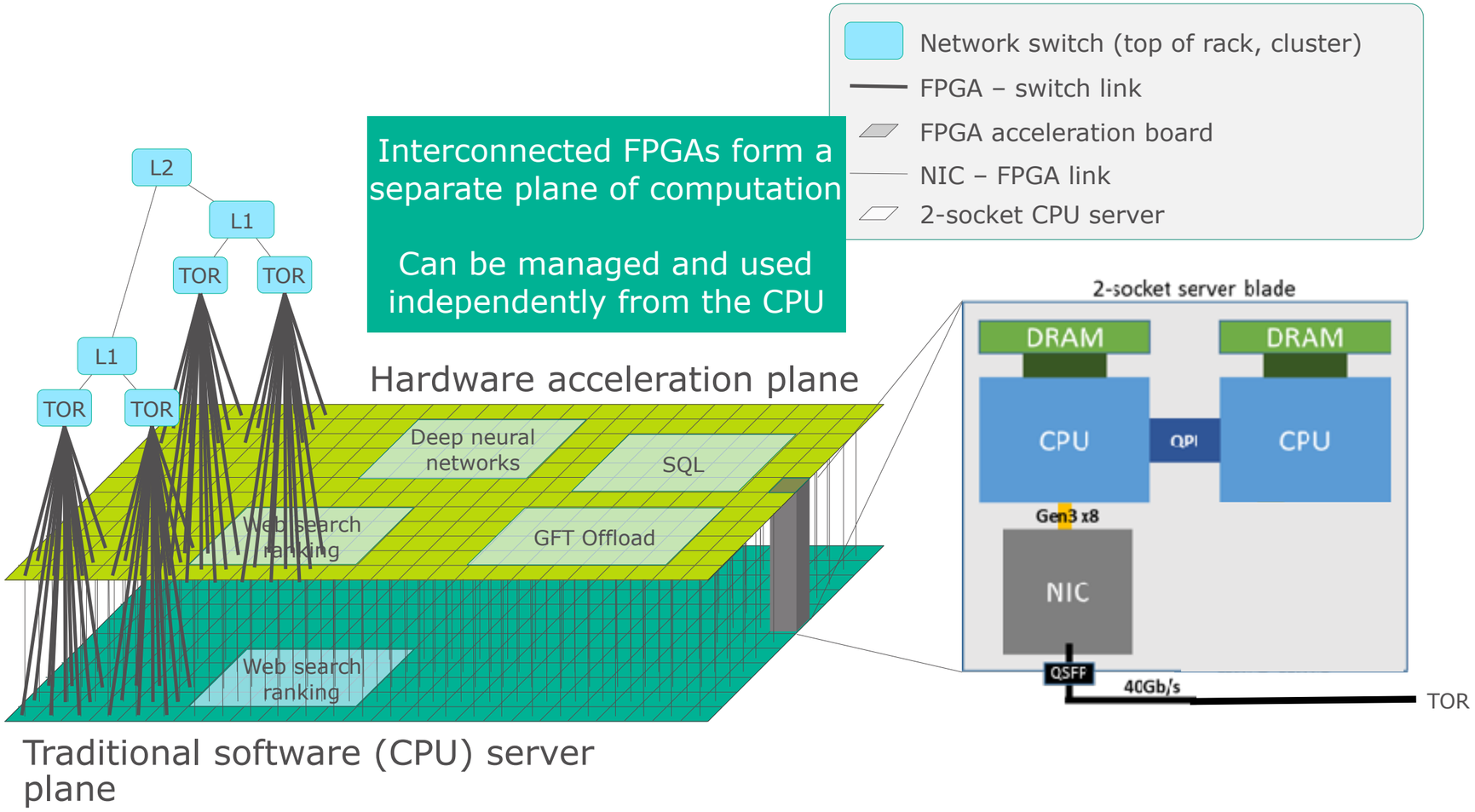
But each stage is now working on what the original 6 instructions would do



A Cloud-Scale Acceleration Architecture

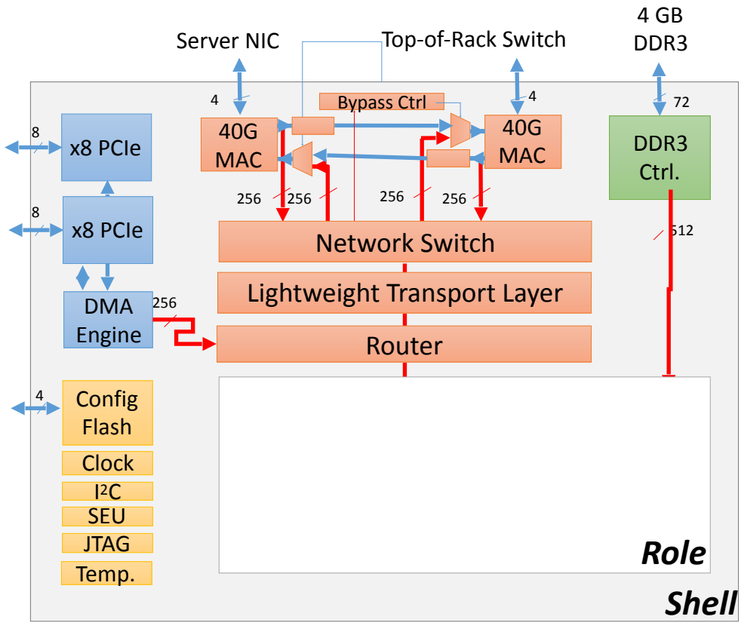
Adrian Caulfield, Eric Chung, Andrew Putnam, Hari Angepat, Jeremy Fowers, Michael Haselman, Stephen Heil, Matt Humphrey, Puneet Kaur, Joo-Young Kim, Daniel Lo, Todd Massengill, Kalin Ovtcharov, Michael Papamichael, Lisa Woods, Sitaram Lanka, Derek Chiou, Doug Burger
Microsoft

Configurable cloud



Gen2 shell

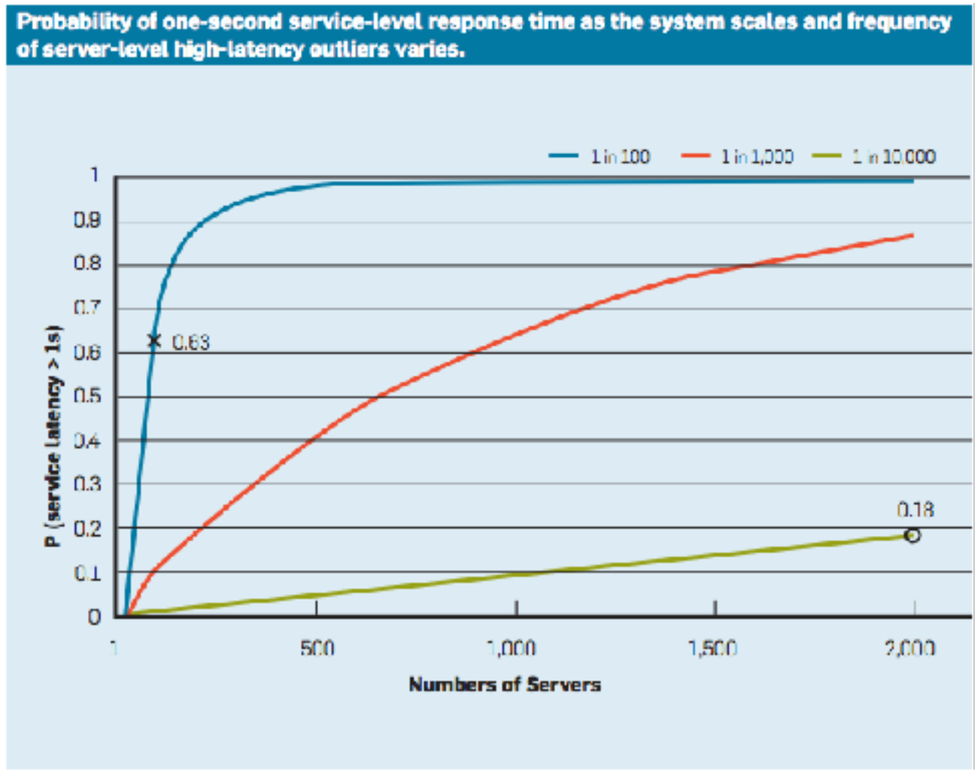
- Foundation for all accelerators
  - Includes PCIe, Networking and DDR IP
  - Common, well tested platform for development
- Lightweight Transport Layer
  - Reliable FPGA-to-FPGA Networking
  - Ack/Nack protocol, retransmit buffers
  - Optimized for lossless network
  - Minimized resource usage



Use cases

- Local: Great service acceleration
- Infrastructure: Fastest cloud network
- Remote: Reconfigurable app fabric (DNNs)

Tail latencies

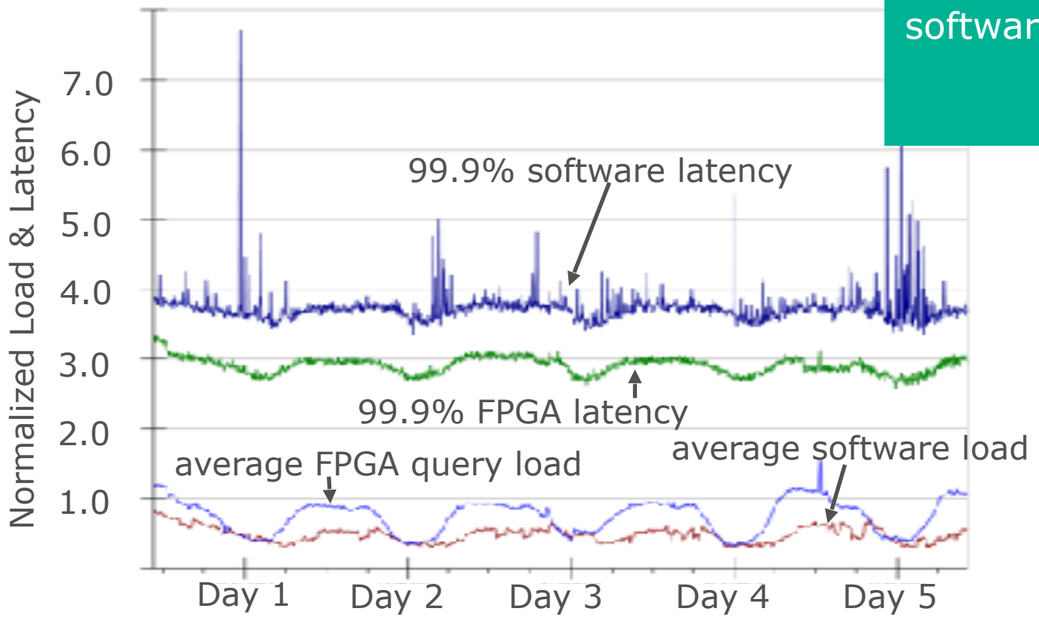


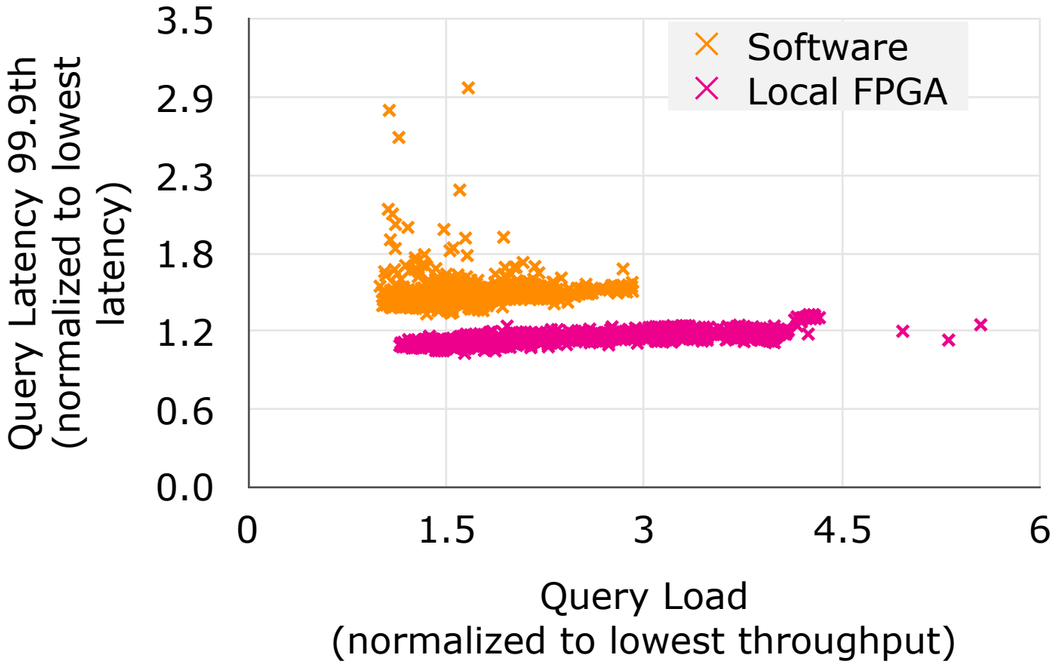
- Tail Latency == 1 in X servers being slow
- Why is this bad? – Each user request now needs several servers – Changes of experience tail is much higher
- If 99% of the server's response time is 10ms, but 1% of them take 1 second to response
  - If the user only needs one, the mean is OK
  - If the user needs 100 partitions from 100 servers, 63% of the requests takes more than 1 seconds.

5 day bed-level latency

- Lower & more consistent 99.9th tail latency
- In production for years

Even at 2× query load, accelerated ranking has lower latency than software **at any load**

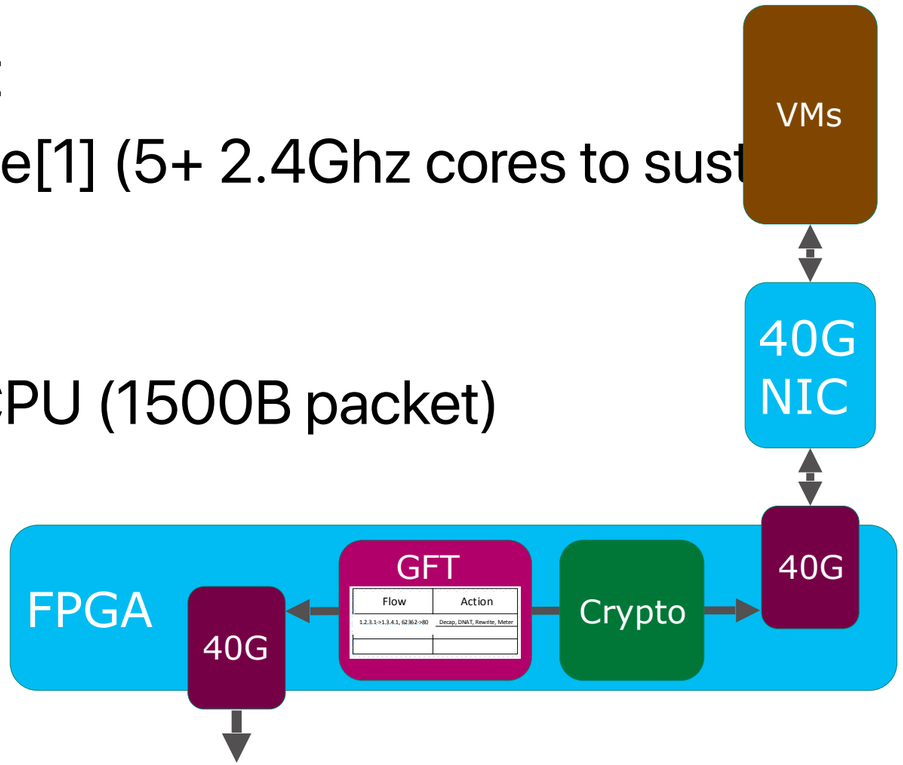




Accelerated networking

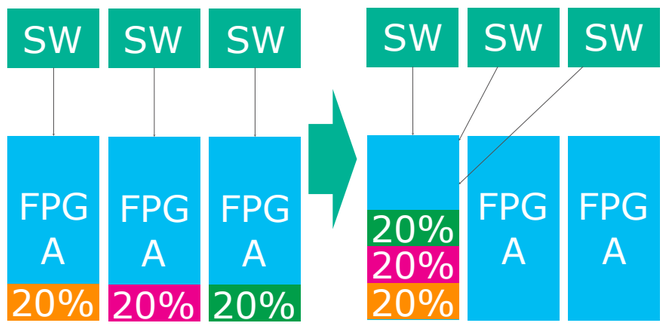
- Software defined networking
  - Generic Flow Table (GFT) rule based packet rewriting
  - 10x latency reduction vs software, CPU load now <1 core
  - 25Gb/s throughput at 25µs latency – the fastest cloud network

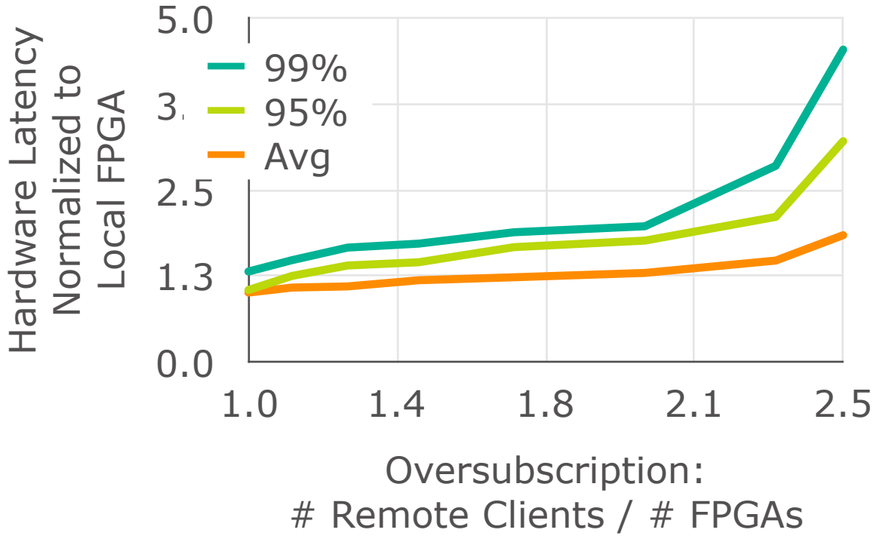
- Capable of 40 Gb line rate encrypt and decrypt
  - On Haswell, AES GCM-128 costs 1.26 cycles/byte[1] (5+ 2.4Ghz cores to sust 40Gb/s)
  - CBC and other algorithms are more expensive
  - AES CBC-128-SHA1 is 11µs in FPGA vs 4µs on CPU (1500B packet)
  - Higher latency, but significant CPU savings



Shared DNN

- Economics: consolidation
  - Most accelerators have more throughput than a single host requires
  - Share excess capacity, use fewer instances
  - Frees up FPGAs for other use services
- DNN accelerator
  - Sustains 2.5x busy clients in microbenchmark, before queuing delay drives latency up





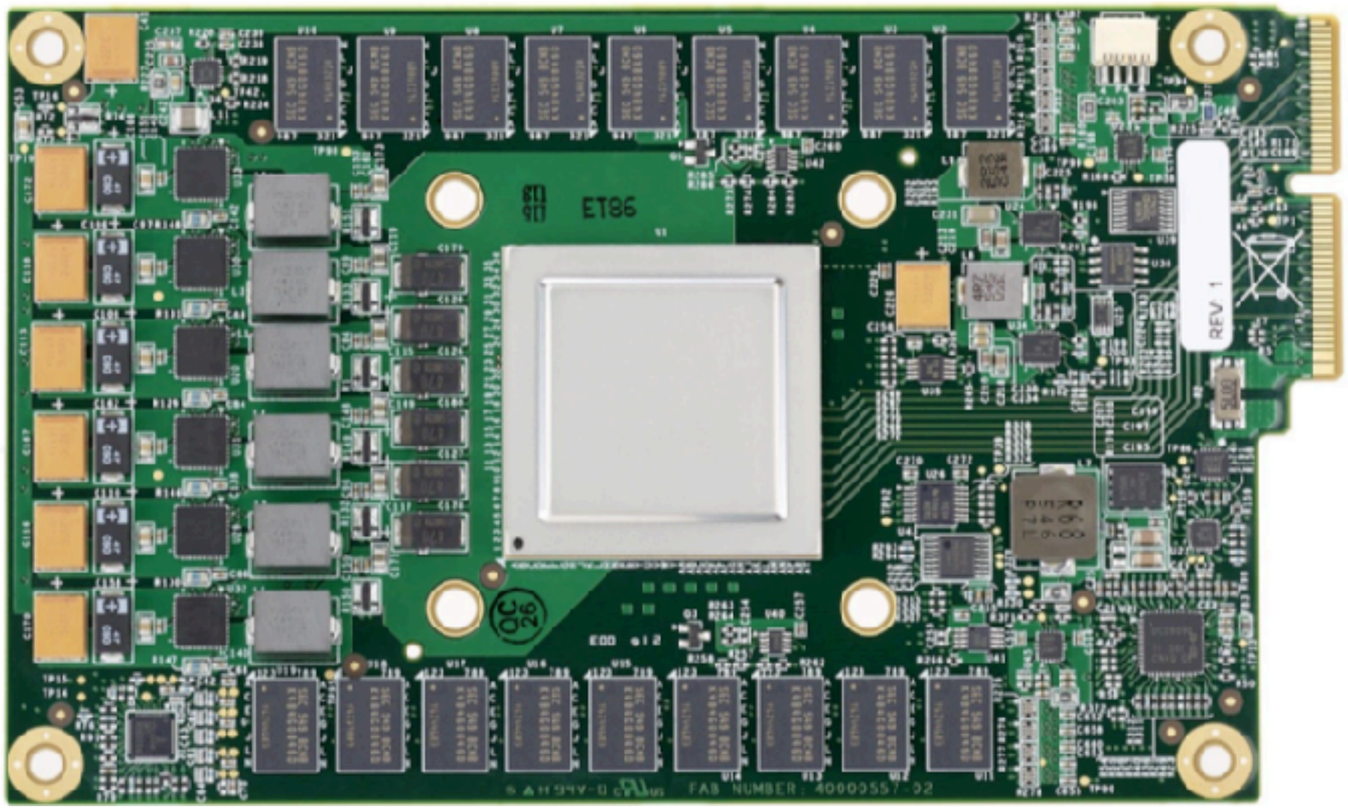
Summary: What makes a configurable cloud?

- Local, infrastructure and remote acceleration
  - Gen1 showed significant gains even for complex services (~2x for Bing)
  - Needs to have clear benefit for majority of servers: infrastructure
- Economics must work
  - What works at small scale doesn't always work at hyperscale and vice versa
  - Little tolerance for superfluous costs
  - Minimized complexity and risk in deployment and maintenance
- Must be flexible
  - Support simple, local accelerators and complex, shared accelerators at the same time
  - Rapid deployment of new protocols, algorithms and services across the cloud

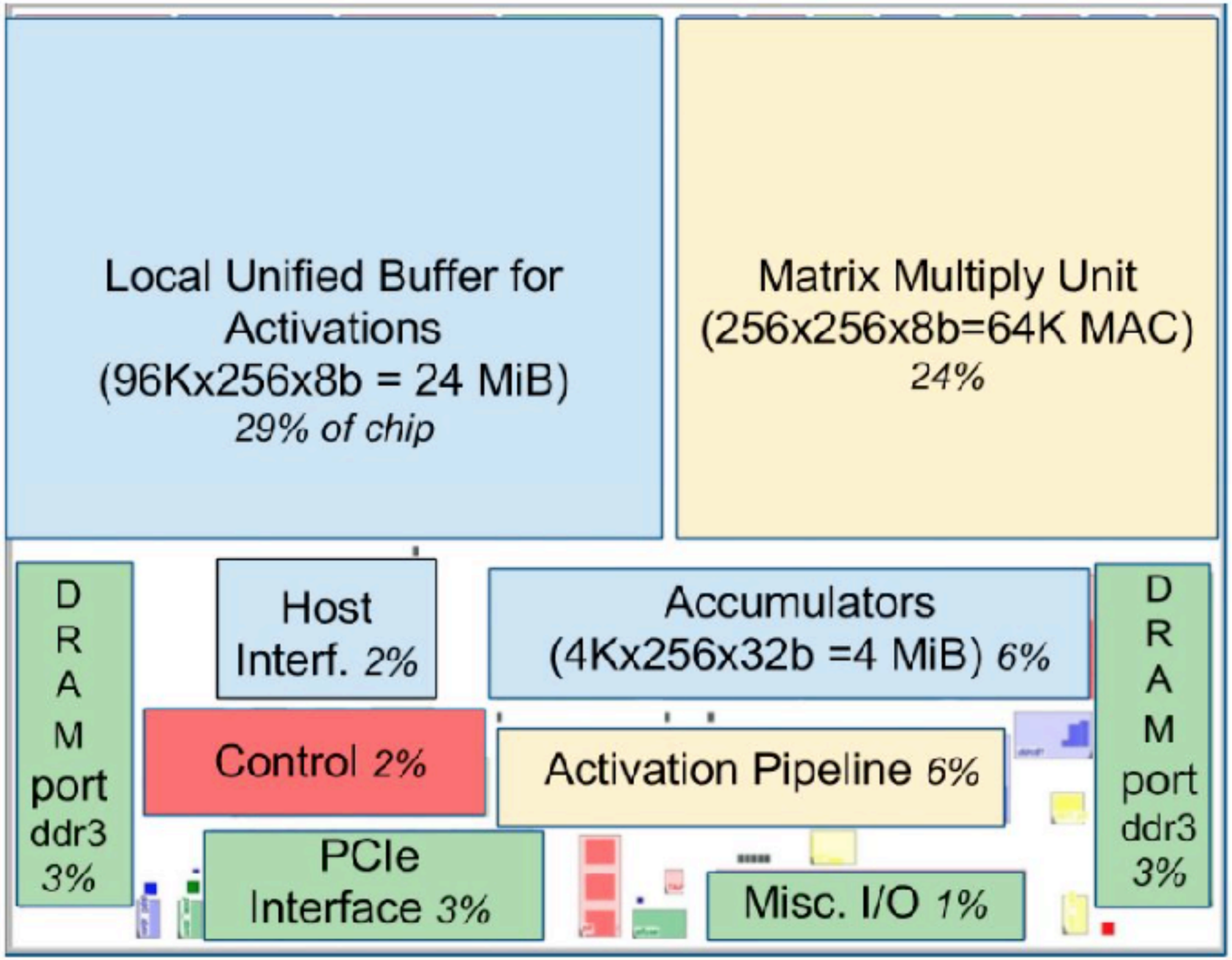
In-Datacenter Performance Analysis of a Tensor Processing Unit

N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P.-l. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T. V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C. R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, D. Killebrew, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin, G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Na- garajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn, G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Wal- ter, W. Wang, E. Wilcox, and D. H. Yoon
Google Inc.

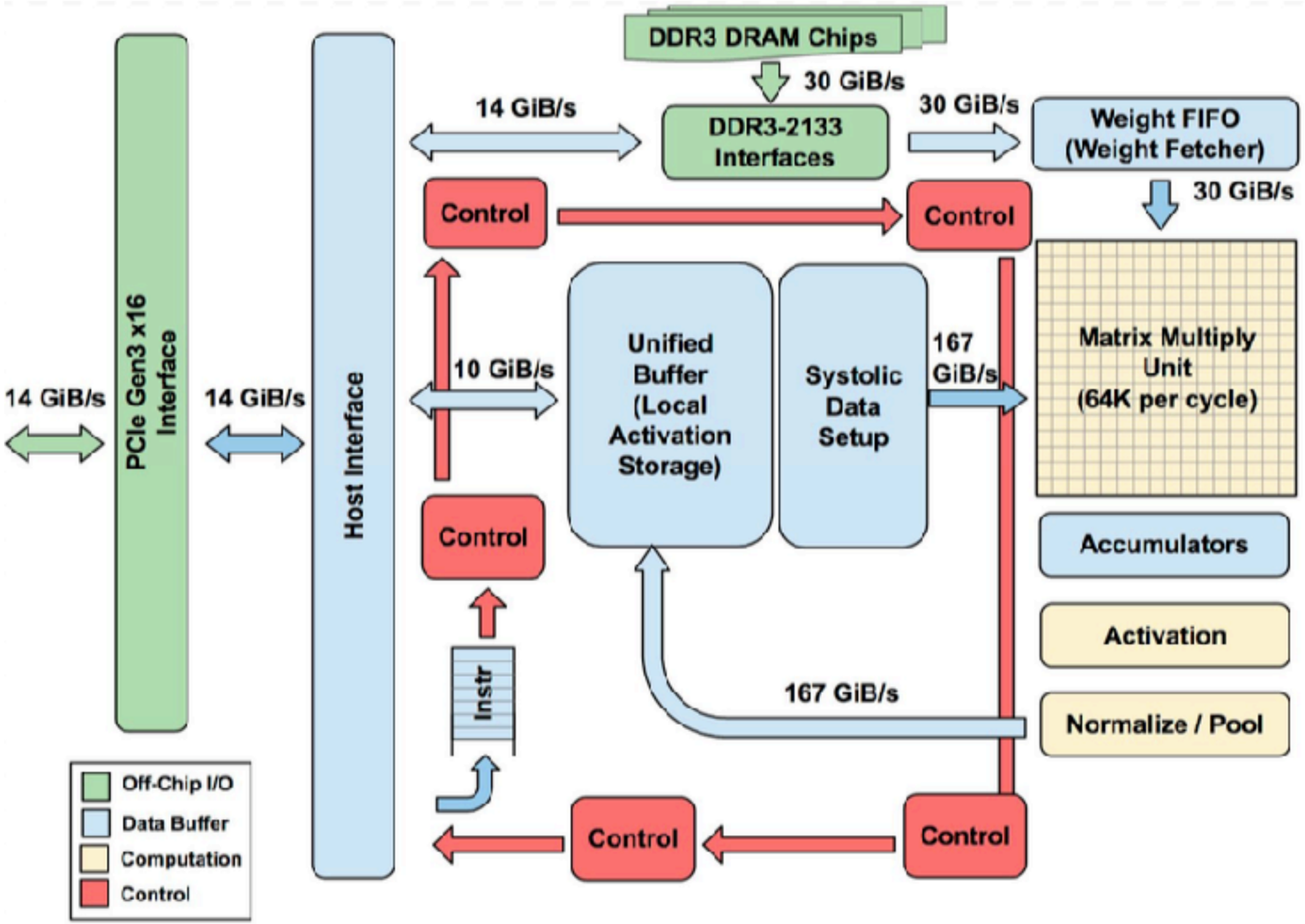
What TPU looks like



TPU Floorplan



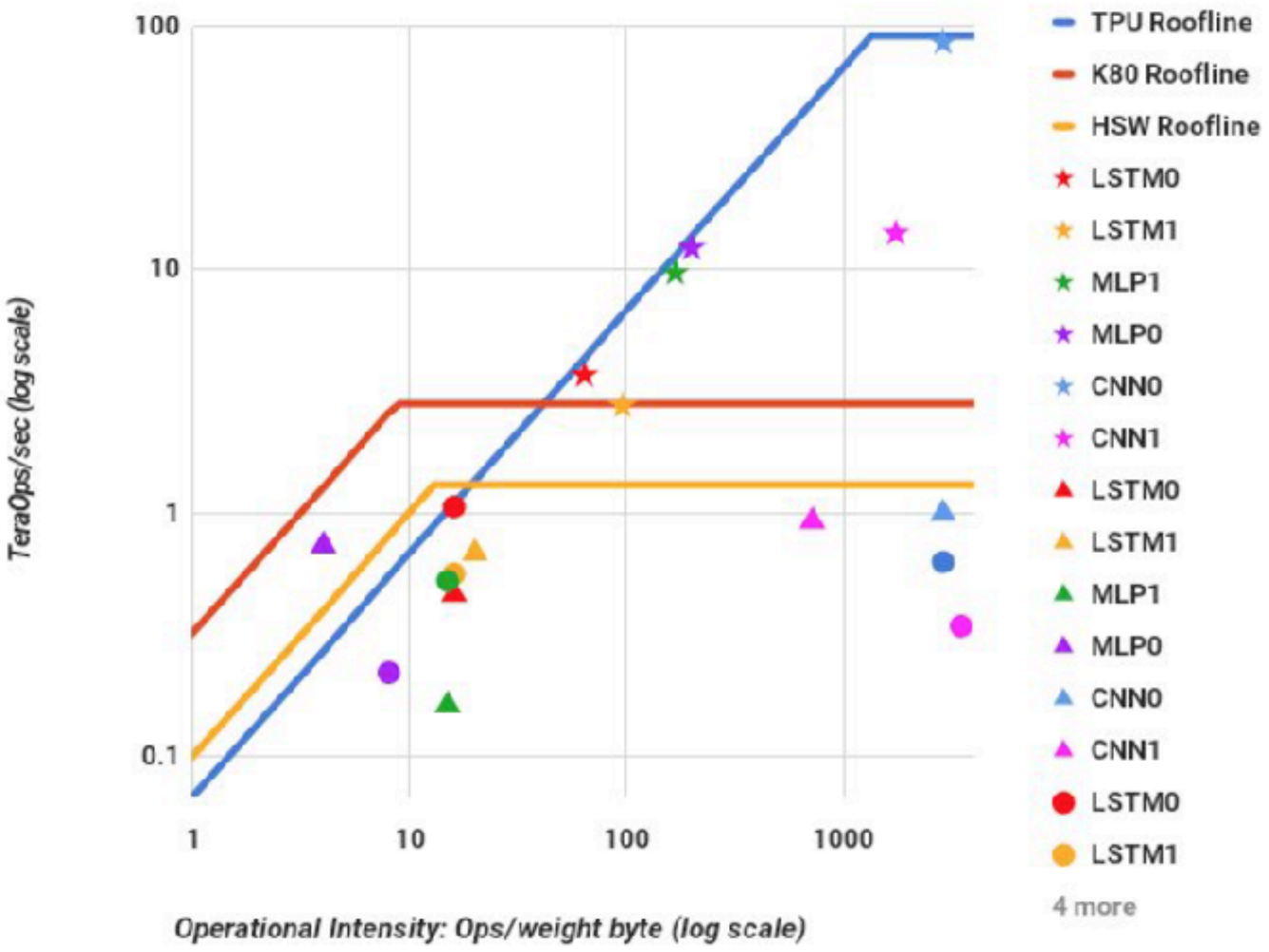
TPU Block diagram



Experimental setup

Model	Die										Benchmarked Servers				
	mm^2	nm	MHz	TDP	Measured		TOPS/s		GB/s	On-Chip Memory	Dies	DRAM Size	TDP	Measured	
					Idle	Busy	8b	FP						Idle	Busy
Haswell E5-2699 v3	662	22	2300	145W	41W	145W	2.6	1.3	51	51 MiB	2	256 GiB	504W	159W	455W
NVIDIA K80 (2 dies/card)	561	28	560	150W	25W	98W	--	2.8	160	8 MiB	8	256 GiB (host) + 12 GiB x 8	1838W	357W	991W
TPU	NA*	28	700	75W	28W	40W	92	--	34	28 MiB	4	256 GiB (host) + 8 GiB x 4	861W	290W	384W

Performance/Rooflines



Tail latency

Type	*Batch*	*99th% Response*	*Inf/s (IPS)*	*% Max IPS*
CPU	16	7.2 ms	5,482	42%
CPU	64	21.3 ms	13,194	100%
GPU	16	6.7 ms	13,461	37%
GPU	64	8.3 ms	36,465	100%
TPU	200	7.0 ms	225,000	80%
TPU	250	10.0 ms	280,000	100%

Table 4. 99-th% response time and per die throughput (IPS) for MLP0 as batch size varies for MLP0. The longest allowable latency is 7 ms. For the GPU and TPU, the maximum MLP0 throughput is limited by the host server overhead. Larger batch sizes increase throughput, but as the text explains, their longer response times exceed the limit, so CPUs and GPUs must use less-efficient, smaller batch sizes (16 vs. 200).

What NVIDIA says

https://blogs.nvidia.com/blog/2017/04/10/ai-drives-rise-accelerated-computing-datacenter/

	K80 2012	TPU 2015	P40 2016
Inferences/Sec <10ms latency	$^{1}/_{13}$X	1X	2X
Training TOPS	6 FP32	NA	12 FP32
Inference TOPS	6 FP32	90 INT8	48 INT8
On-chip Memory	16 MB	24 MB	11 MB
Power	300W	75W	250W
Bandwidth	320 GB/S		

While Google and NVIDIA chose different development paths, there were several themes common to both our approaches. Specifically:

- AI requires accelerated computing. Accelerators provide the significant data processing necessary to keep up with the growing demands of deep learning in an era when Moore's law is slowing.
- Tensor processing is at the core of delivering performance for deep learning training and inference.
- Tensor processing is a major new workload enterprises must consider when building modern data centers.
- Accelerating tensor processing can dramatically reduce the cost of building modern data centers.

7.10 Fallacies and Pitfalls

In these early days of both DSAs and DNNs, fallacies abound.

Fallacy *It costs $100 million to design a custom chip.*

Figure 7.51 shows a graph from an article that debunks the widely quoted $100-million myth that it was "only" $50 million, with most of the cost being salaries (Olofsson, 2011). Note that the author's estimate is for sophisticated processors that include features that DSAs by definition omit, so even if there were no improvement to the development process, you would expect the cost of a DSA design to be less.

Why are we more optimistic six years later, when, if anything, mask costs are even higher for the smaller process technologies?

First, software is the largest category, at almost a third of the cost. The availability of applications written in domain-specific languages allows the compilers to do most of the work of porting the applications to your DSA, as we saw for the TPU and Pixel Visual Core. The open RISC-V instruction set will also help reduce the cost of getting system software as well as cut the large IP costs.

Mask and fabrication costs can be saved by having multiple projects share a single reticle. As long as you have a small chip, amazingly enough, for $30,000 anyone can get 100 untested parts in 28-nm TSMC technology (Patterson and Nikolić, 2015).

Fallacies & Pitfalls

- Fallacy: NN inference applications in data centers value throughput as much as response time.
- Fallacy: The K80 GPU architecture is a good match to NN inference — GPU is throughput oriented
- Pitfall: For NN hardware, Inferences Per Second (IPS) is an inaccurate summary performance metric — it's simply the inverse of the complexity of the typical inference in the application (e.g., the number, size, and type of NN layers)
- Fallacy: The K80 GPU results would be much better if Boost mode were enabled — Boost mode increased the clock rate by a factor of up to 1.6—from 560 to 875 MHz—which increased performance by 1.4X, but it also raised power by 1.3X. The net gain in performance/Watt is 1.1X, and thus Boost mode would have a minor impact on LSTM1
- Fallacy: CPU and GPU results would be comparable to the TPU if we used them more efficiently or compared to newer versions.

Fallacies & Pitfalls

- Pitfall: Architects have neglected important NN tasks.
  - CNNs constitute only about 5% of the representative NN workload for Google. More attention should be paid to MLPs and LSTMs. Repeating history, it's similar to when many architects concentrated on floating- point performance when most mainstream workloads turned out to be dominated by integer operations.
- Pitfall: Performance counters added as an afterthought for NN hardware.
- Fallacy: After two years of software tuning, the only path left to increase TPU performance is hardware upgrades.
- Pitfall: Being ignorant of architecture history when designing a domain-specific architecture
  - Systolic arrays
  - Decoupled-access/execute
  - CISC instructions

Final words

Conclusion

- Computer architecture is more important than you can ever imagine
- Being a "programmer" is easy. You need to know architecture a lot to be a "performance programmer"
  - Branch prediction
  - Cache
- Multicore era — to get your multithreaded program correct and perform well, you need to take care of coherence and consistency
- We're now in the "dark silicon era"
  - Single-core isn't getting any faster
  - Multi-core doesn't scale anymore
  - We will see more and more ASICs
  - You need to write more "system-level" programs to use these new ASICs.