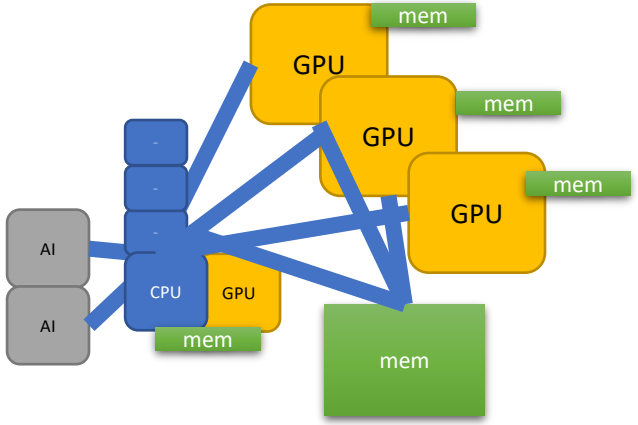
SYCL Academy

Parallelism & Heterogeneity



Why do we need to care?

James Reinders (Intel)

Computer trends: Parallel and Heterogeneous

- Our quest for more performance is eternal;
  how we obtain it adopts to the times.
- Once upon a time… a system was a CPU, and we made the CPUs faster and faster over time… 1950s, 1960s, 1970s, 1980s, 1990s, 2000s…

We feared it was unravelling in the mid-1980s, but we were off by two decades.

By 2006, any frequency scaling ended for all intensive purposes.

Computer trends: Parallel and Heterogeneous

Why Parallel?

Desire to get more work done, by having more workers.

Workers = compute units, devices, processing units, etc.

(e.g., CPU, GPU, FPGA, ASIC, AI chip)

Computer trends: Parallel and Heterogeneous

Why Parallel?

Desire to get more work done, by having more workers.

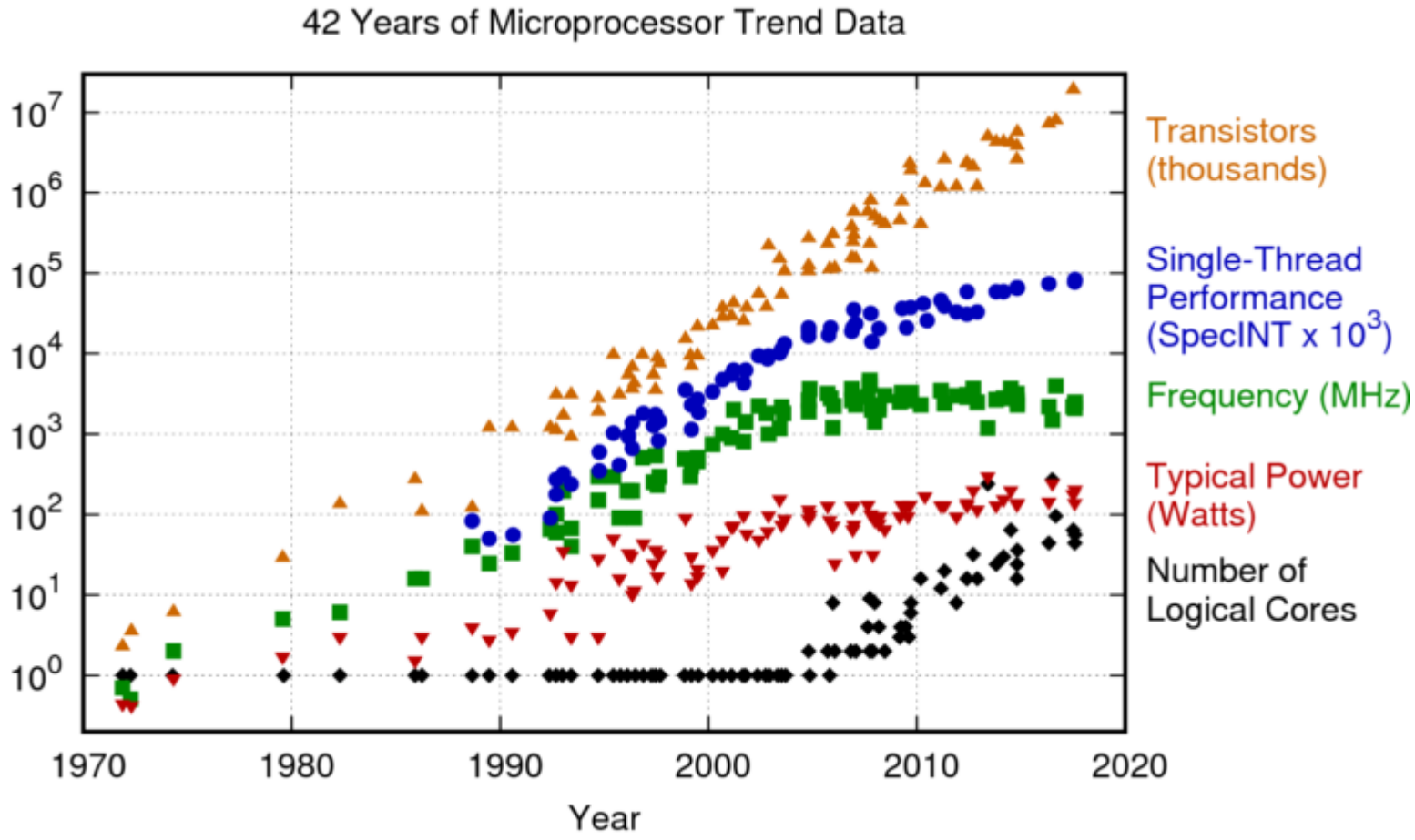
Why Heterogeneous?

Desire to get more work done, by having different types of workers.

Workers = compute units, devices, processing units, etc.

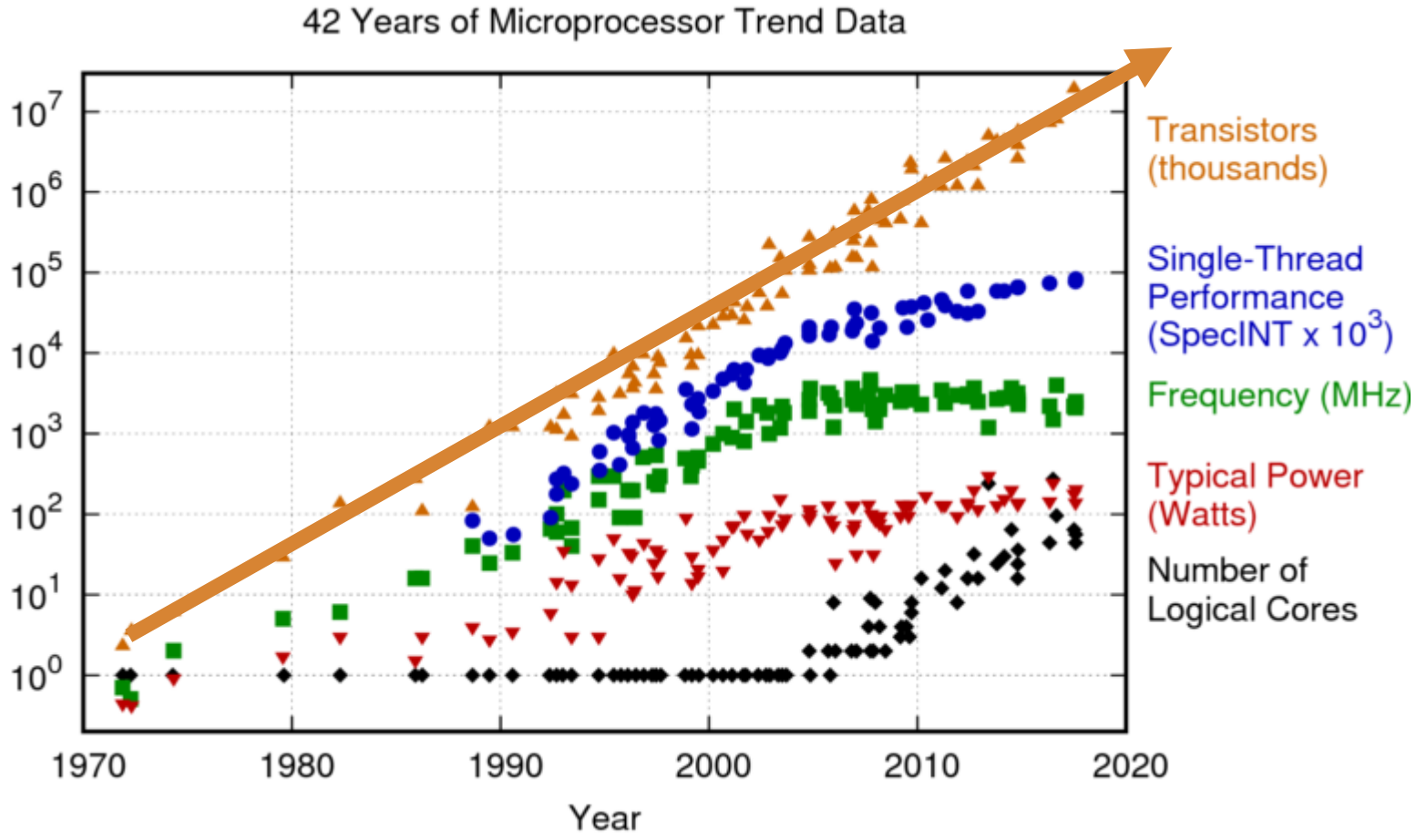
(e.g., CPU, GPU, FPGA, ASIC, AI chip)

Our quest for more performance is eternal; how we obtain it adopts to the times



Source: tinyurl.com/karlruppdata (CC BY 4.0 license)

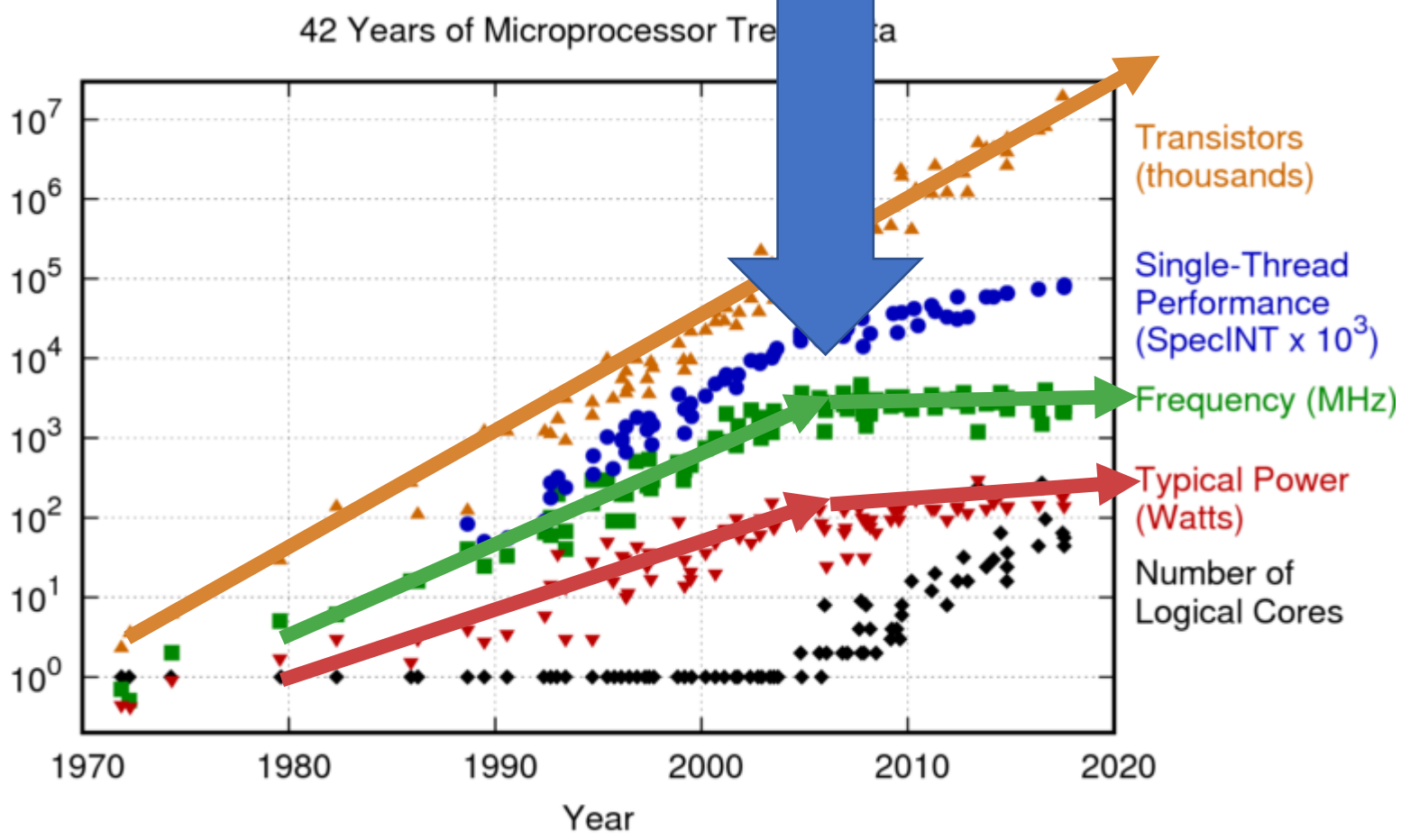
Our quest for more performance is eternal; how we obtain it adopts to the times



Source: tinyurl.com/karlruppdata (CC BY 4.0 license)

Our quest for nce is eternal; how we obta e times

The famous
"Hit the Power Wall"
~2006



Our quest for nce is eternal; how we obta e times

The famous

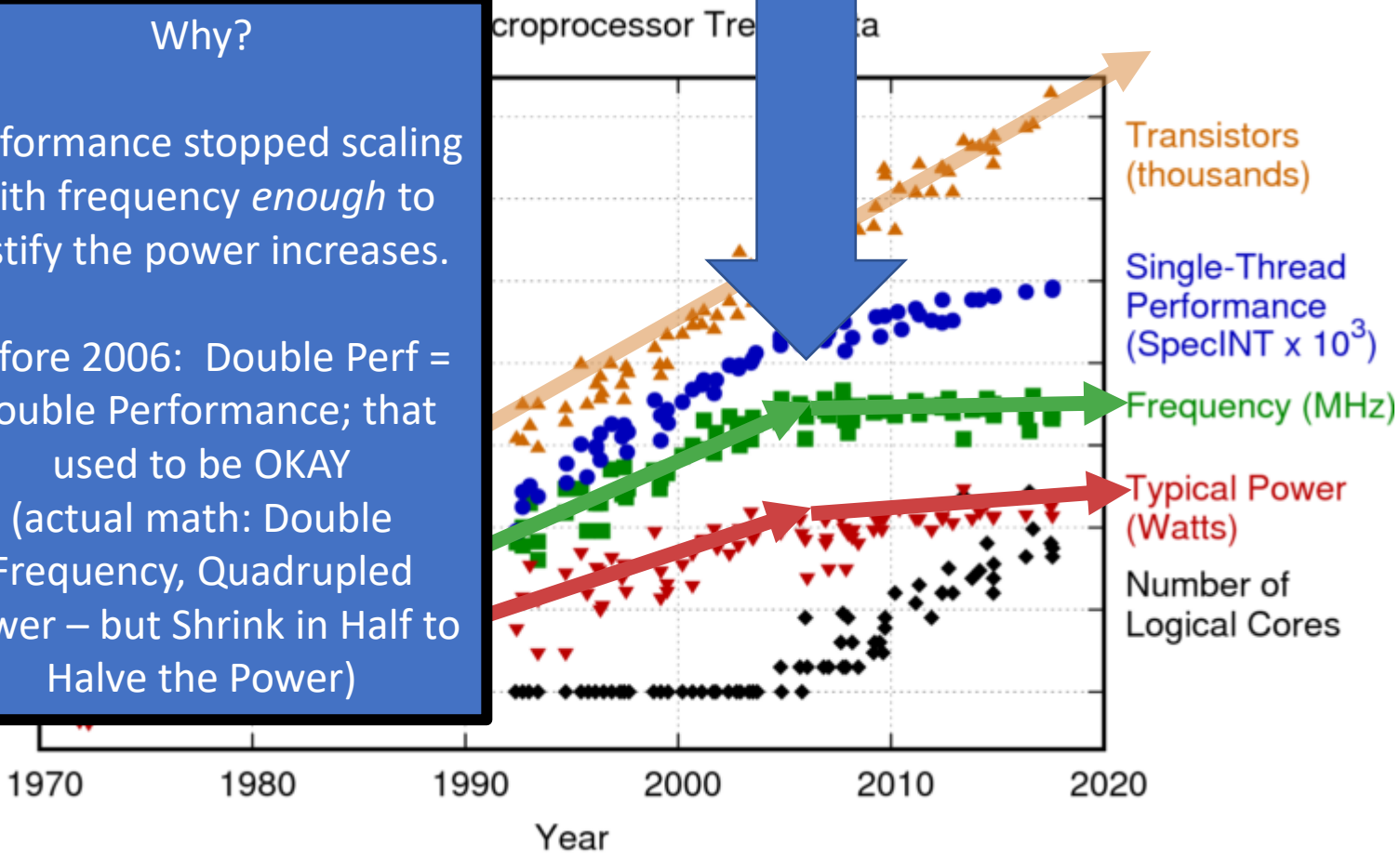
"Hit the Power Wall"

~2006

Why?

Performance stopped scaling with frequency *enough* to justify the power increases.

Before 2006: Double Perf = Double Performance; that used to be OKAY (actual math: Double Frequency, Quadrupled Power – but Shrink in Half to Halve the Power)



SYCL Academy

Our quest for nce is eternal; how we obta e times

The famous

"Hit the Power Wall"

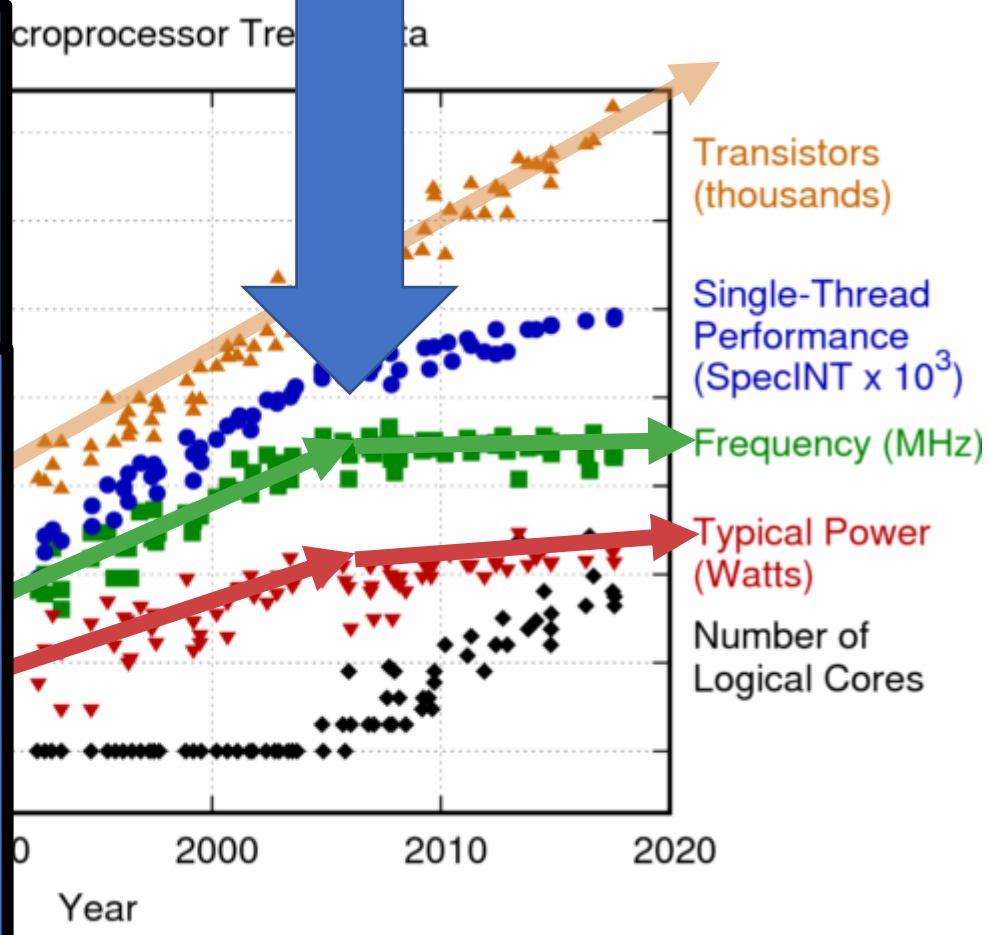
~2006

Why?

Performance stopped scaling with frequency *enough* to justify the power increases.

Around 2006 –
Three problems converged:

1. **Power Wall:**
   Thermal Constraints meant performance Increases no longer allowed to scale up power needs
2. **Memory Wall:**
   Frequency scaling of computation not scaling system performance, largely due to memory
3. **ILP Wall:**
   Not enough single-threaded parallelism to fuel performance gains.



SYCL Academy

Our quest for nce is eternal; how we obta e times

The famous

“Hit the Power Wall”

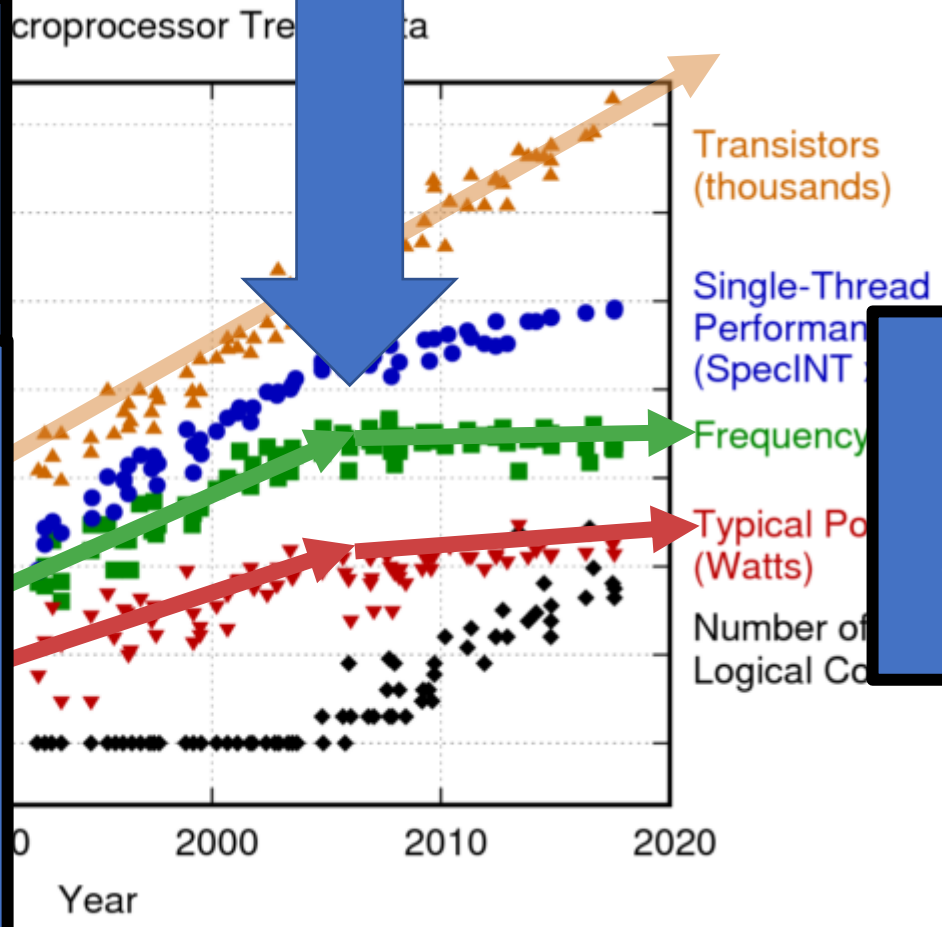
~2006

Why?

Performance stopped scaling with frequency *enough* to justify the power increases.

Around 2006 –
Three problems converged:

1. **Power Wall:**
   Thermal Constraints meant performance Increases no longer allowed to scale up power needs
2. **Memory Wall:**
   Frequency scaling of computation not scaling system performance, largely due to memory
3. **ILP Wall:**
   Not enough single-threaded parallelism to fuel performance gains.

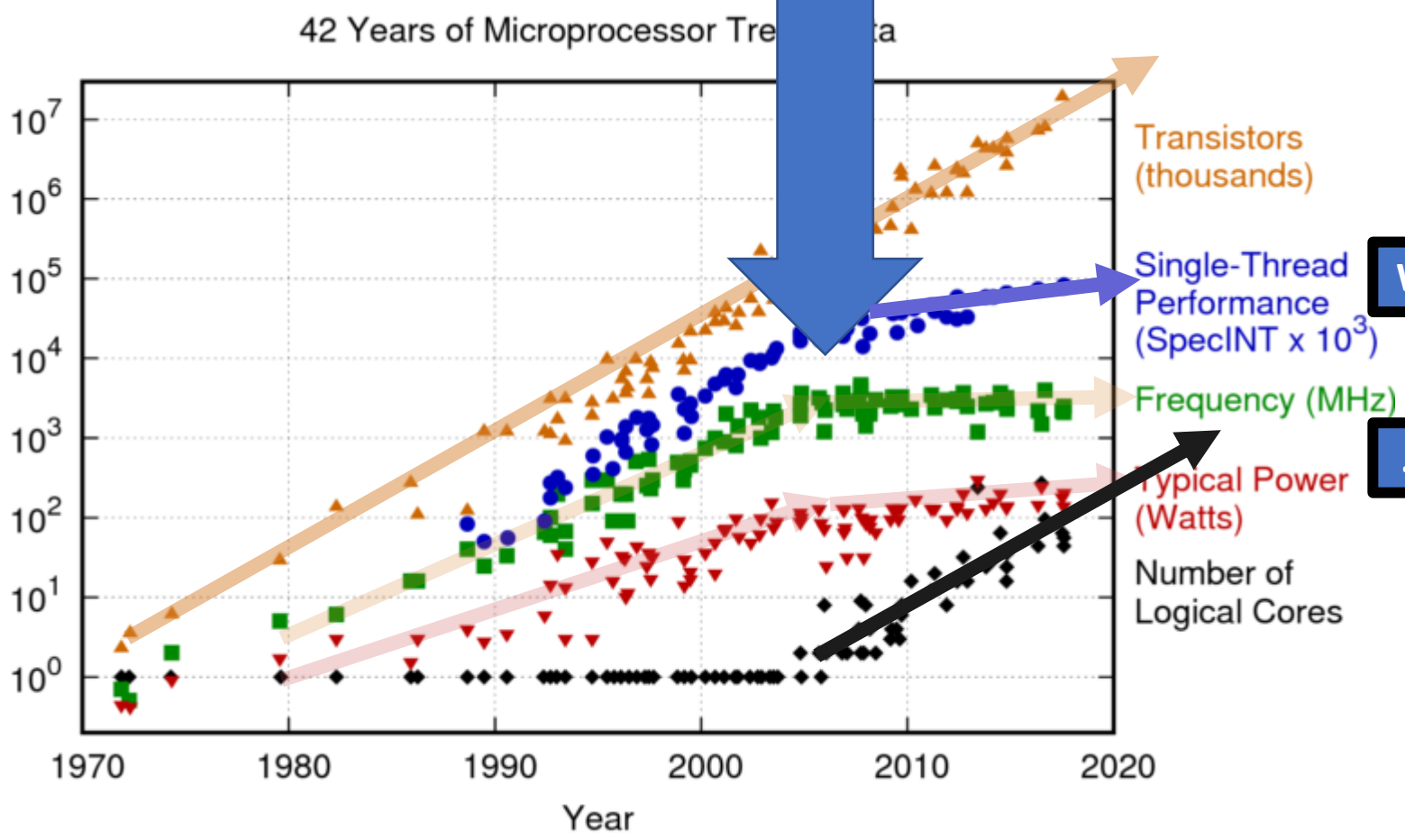


Frequency flattens, power consumption flattens.

The age of frequency scaling ends.

Our quest for nce is eternal; how we obta e times

The famous
"Hit the Power Wall"
~2006

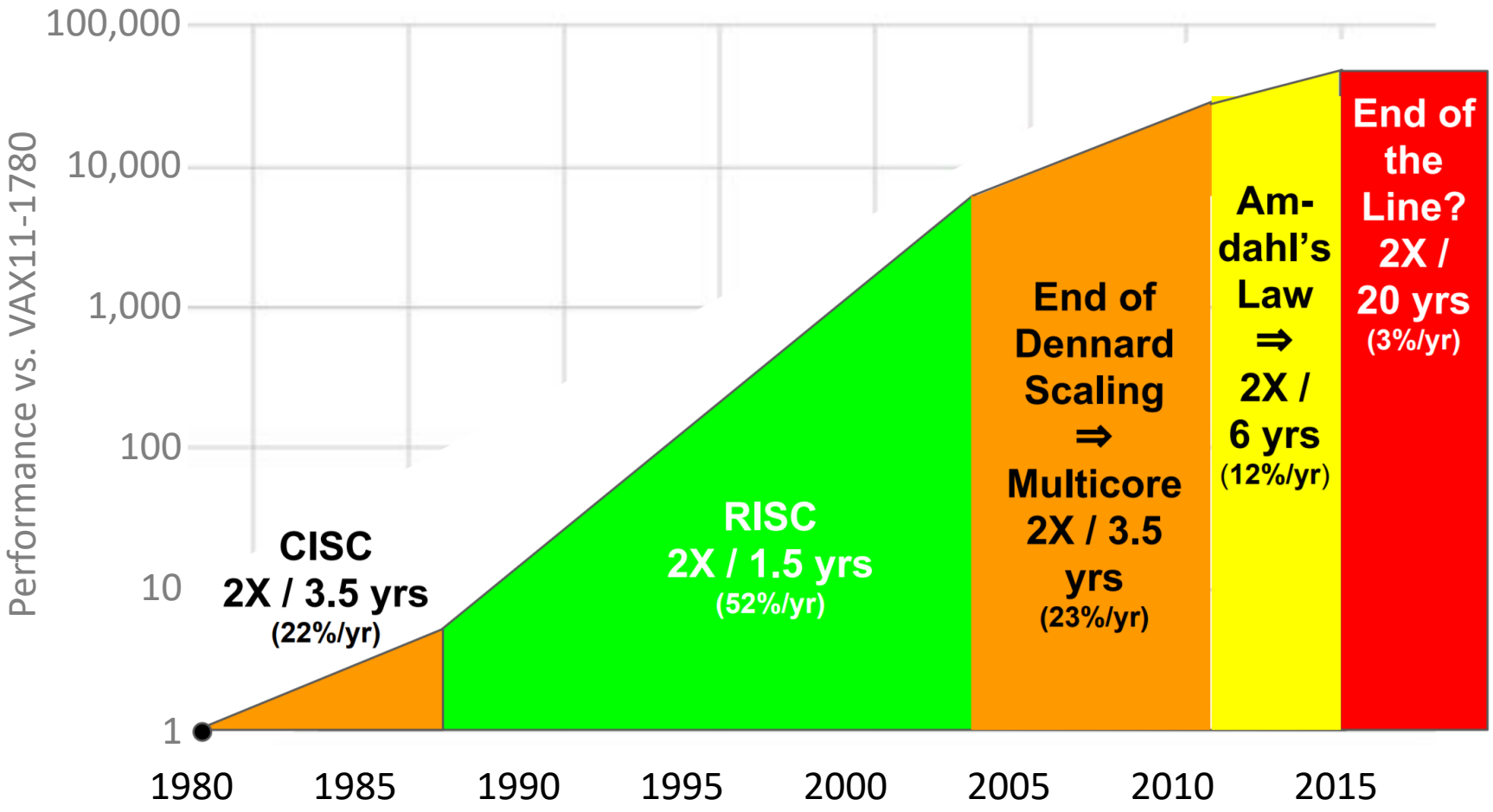


While this flattens…

…parallelism rises…

…because our quest for more performance never ends.

End of speed-ups, unless we go parallel ?



source: https://tinyurl.com/HPcambrian
(ISCA 2018 presentation by John Hennessy and David Patterson)

Parallelism need not be homogeneous

It’s easier sticking with homogeneous, but increasing that is not enough.

Remember our insatiable appetite for performance?

Architectures that “go parallel” are limited only by our imaginations (and various tolerances)

- SIMD on processors (e.g., MMX, SSE, AVX, DL-Boost, …)
- GPU – on and off die/package
- FPGA
- ASIC (e.g., TPU, many AI start-up chips)

Architectures that “go parallel” are limited only by our imaginations (and various tolerances)

- SIMD on processors (e.g., MMX, SSE, AVX, DL-Boost, …)
- GPU – on and off die/package
- FPGA
- ASIC (e.g., TPU, many AI start-up chips)

What does a machine look like in a heterogeneous world?

CPU

pet peeves

CPU mean “central” processing unit
every computer has one, or more CPUs, by definition

Maybe a better term is “*most* central processing unit” since in a heterogeneous world, we will have many (diverse)processing units

A sort of “who is in charge” (at least at first)

CPU does NOT mean microprocessor

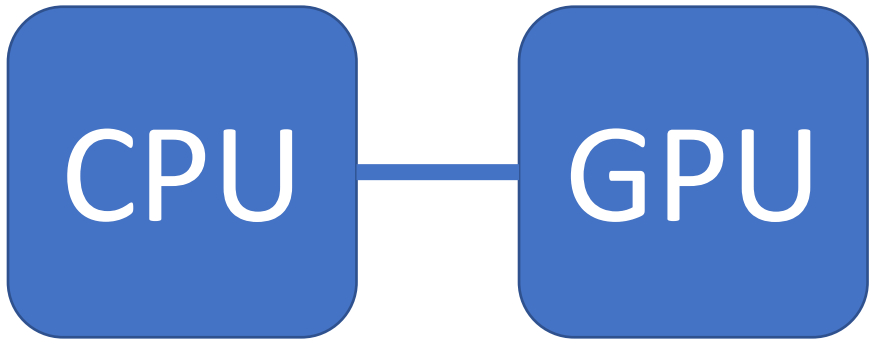
What does a machine look like in a heterogeneous world?

CPU

Yes!

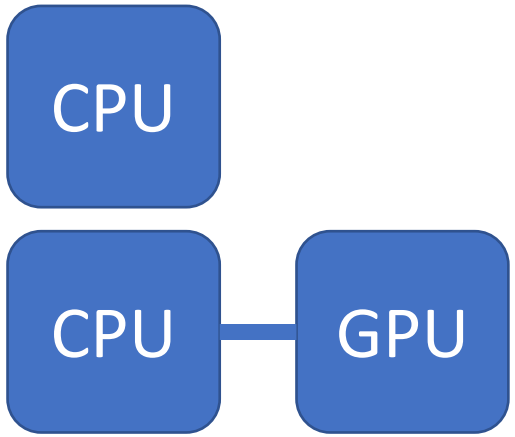
What does a machine look like in a heterogeneous world?

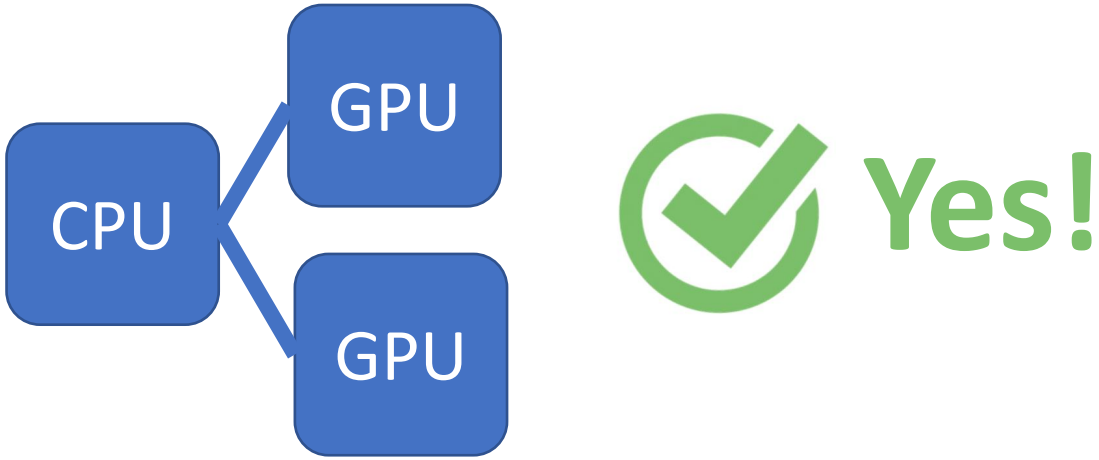
Key: Each processing unit (PU) can be doing work at the same time (in parallel)



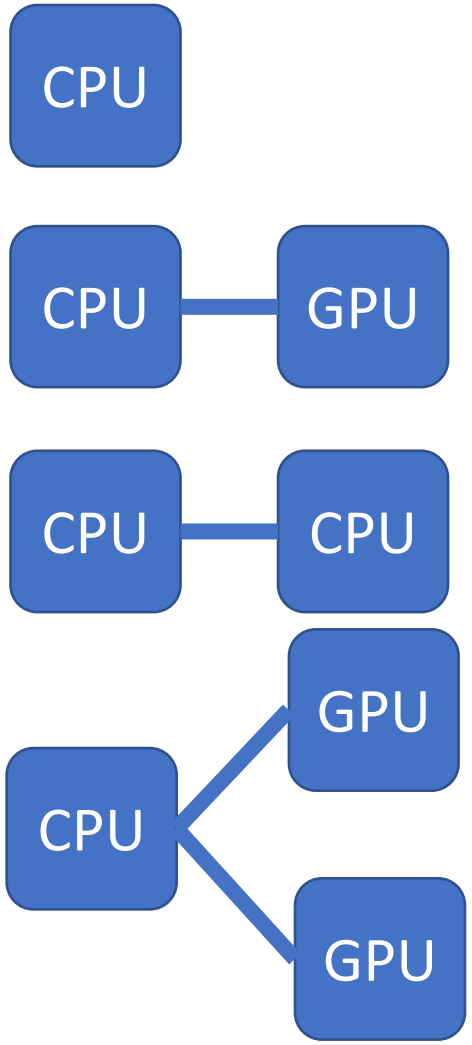
Of course!

What does a machine look like in a heterogeneous world?

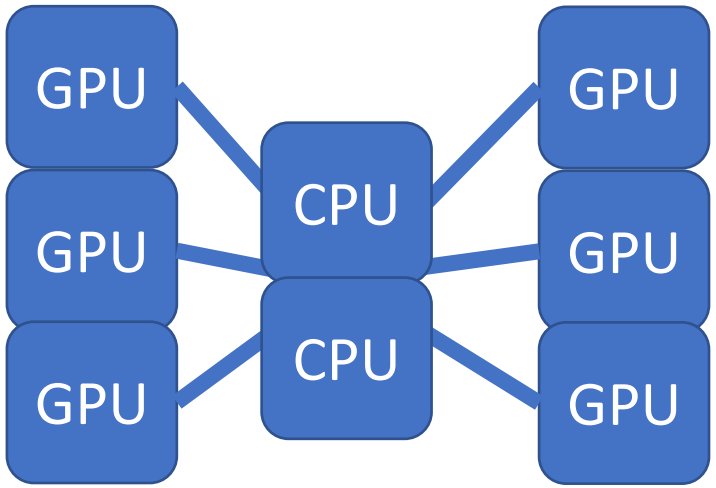




What does a machine look like in a heterogeneous world?

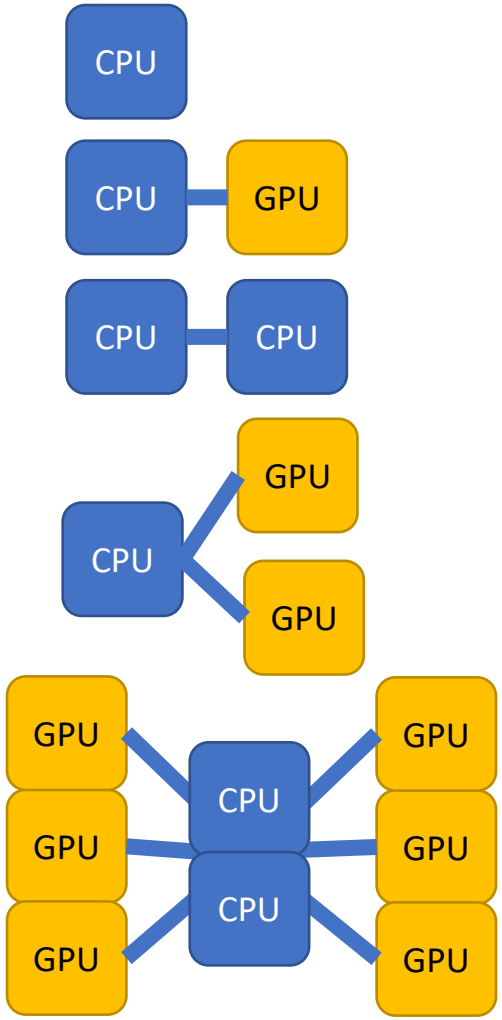


Every device can be doing work in parallel.

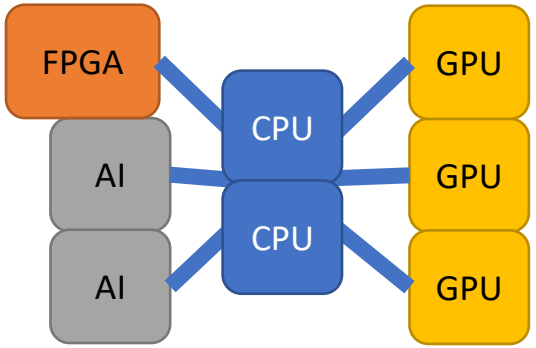


Yes!

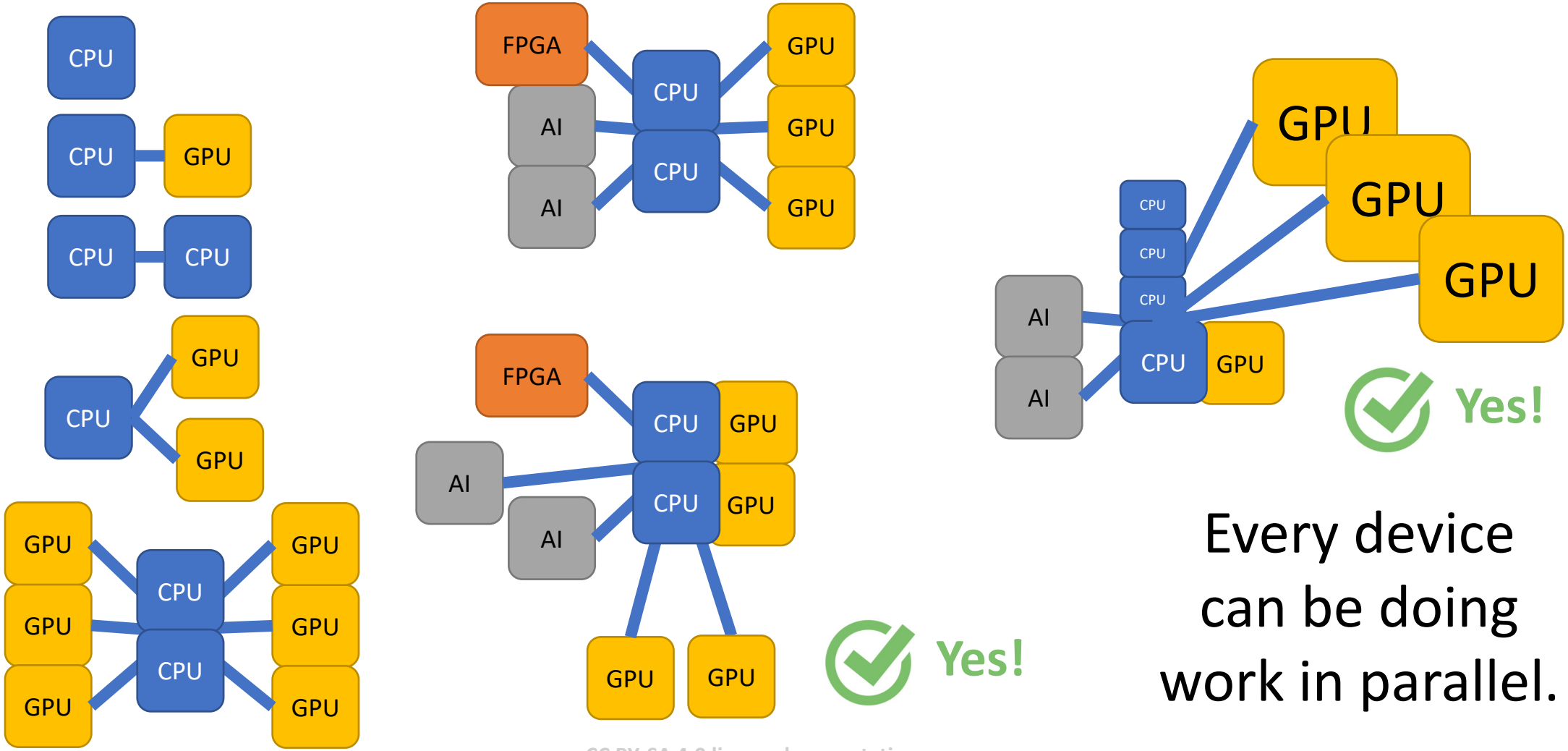
What does a machine look like in a heterogeneous world?



Every device can be doing work in parallel.

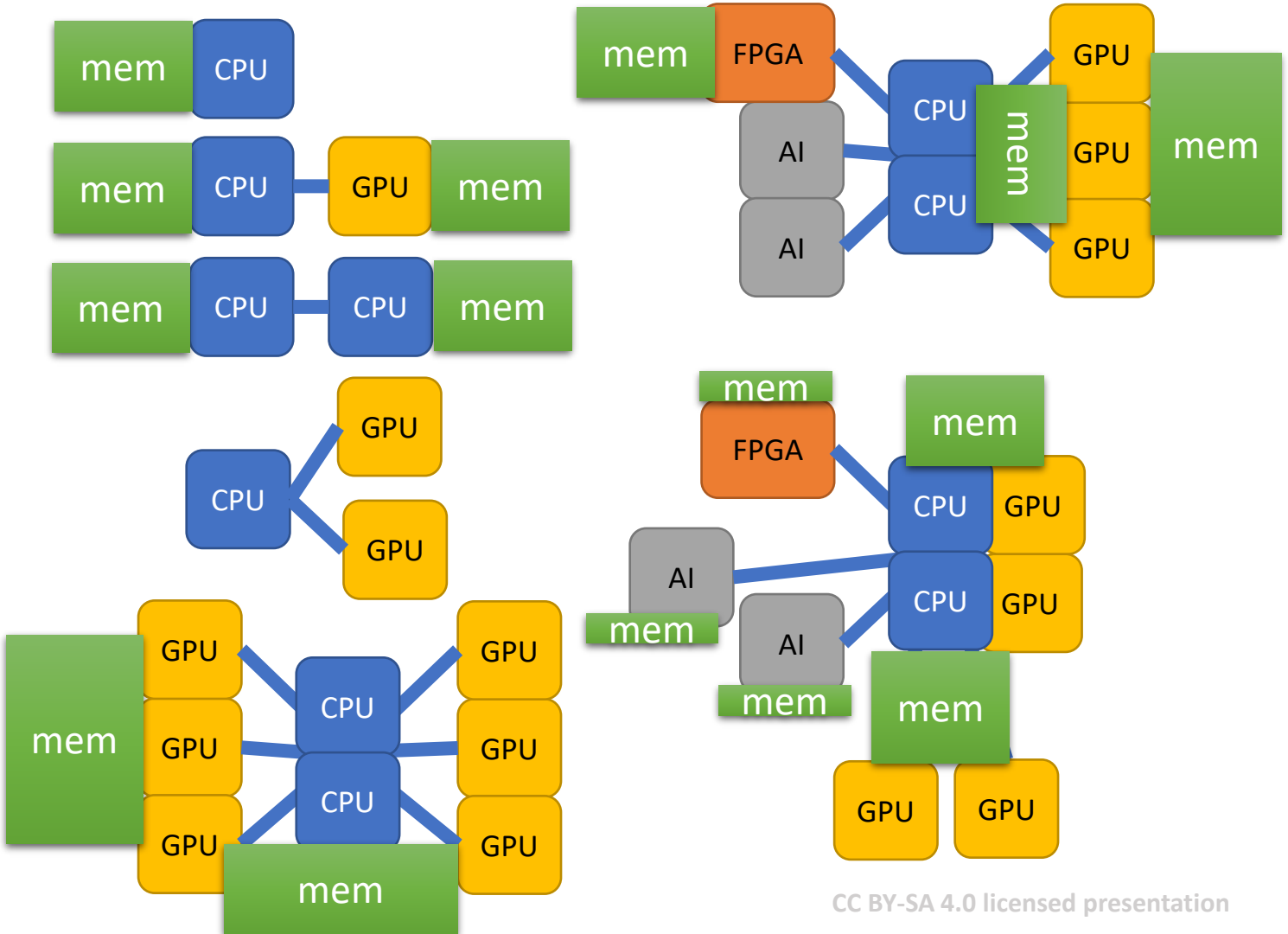


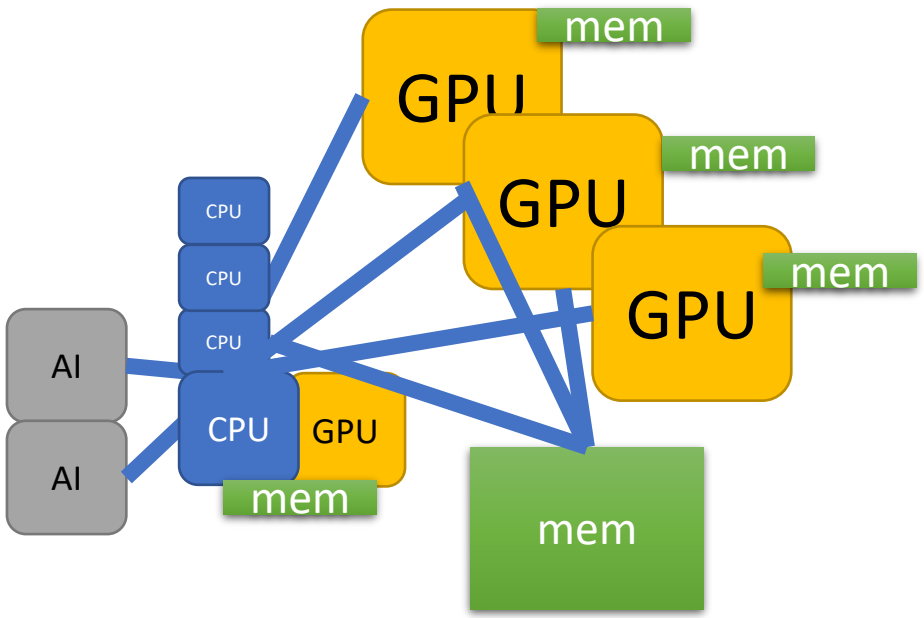
What does a machine look like in a heterogeneous world?



Every device can be doing work in parallel.

What does a machine look like in a heterogeneous world?





Memory adds to the fun.

Oh my!

What does a machine look like in a heterogeneous world?

Diversity in devices (capabilities)

Diversity in memory connectivity/coherence

Diversity in how they all connect

Desire to use them all concurrently in parallel

This is the world we need to support.

Memory adds to the fun.

We'll learn how SYCL helps with all these.

Architectures that "go parallel" are limited only by our imaginations (and various tolerances)

- SIMD on processors (e.g., MMX, SSE, AVX, DL-Boost, …)
- GPU – on and off die/package
- FPGA
- ASIC (e.g., TPU, many AI start-up chips)

- Why specialize?
  - More effective parallelism for a specific domain – *MD/*MT/VLIW/OoO/Pipelines
  - More effective use of memory bandwidth – less hardware managed caches
  - Eliminate unnecessary accuracy – increase performance by reducing pressures
  - Eliminate unnecessary generality – rely on more limited languages/frameworks

A New Golden Age for Computer Architecture

"The next decade will see a **Cambrian explosion of novel computer architectures**, meaning exciting times for computer architects in academia and industry."

ACM Turing Award laureates John Hennessy and David Patterson (CACM, Feb 2019, Vol 62, No 2, pp 48-60)

recommend: https://tinyurl.com/HPcambrian

Can we survive the diversity? Do we have a choice?

To survive, we need “open, multivendor, multiarchitecture”

Why?

- When a computer was homogeneous – we could program it with any tool, even if it was unique or proprietary.
- When a computer is heterogeneous – we need tools to work together.
  - If they don’t – will we be forced to use only single vendor machines?
  - Getting tools to work together is hard – and incomplete today.
  - The Cambrian explosion

Before we use SYCL, we need to

Think Parallel