MCHPC'19

Workshop on Memory-Centric High-Performance Computing

Prospects for Memory

Keynote Presentation

J. Thomas Pawlowski

Current: Self-employed/consultant/entrepreneur, IEEE Fellow

Former: Chief Technologist and systems architect, Micron Fellow

Future: Make me an offer I can't refuse

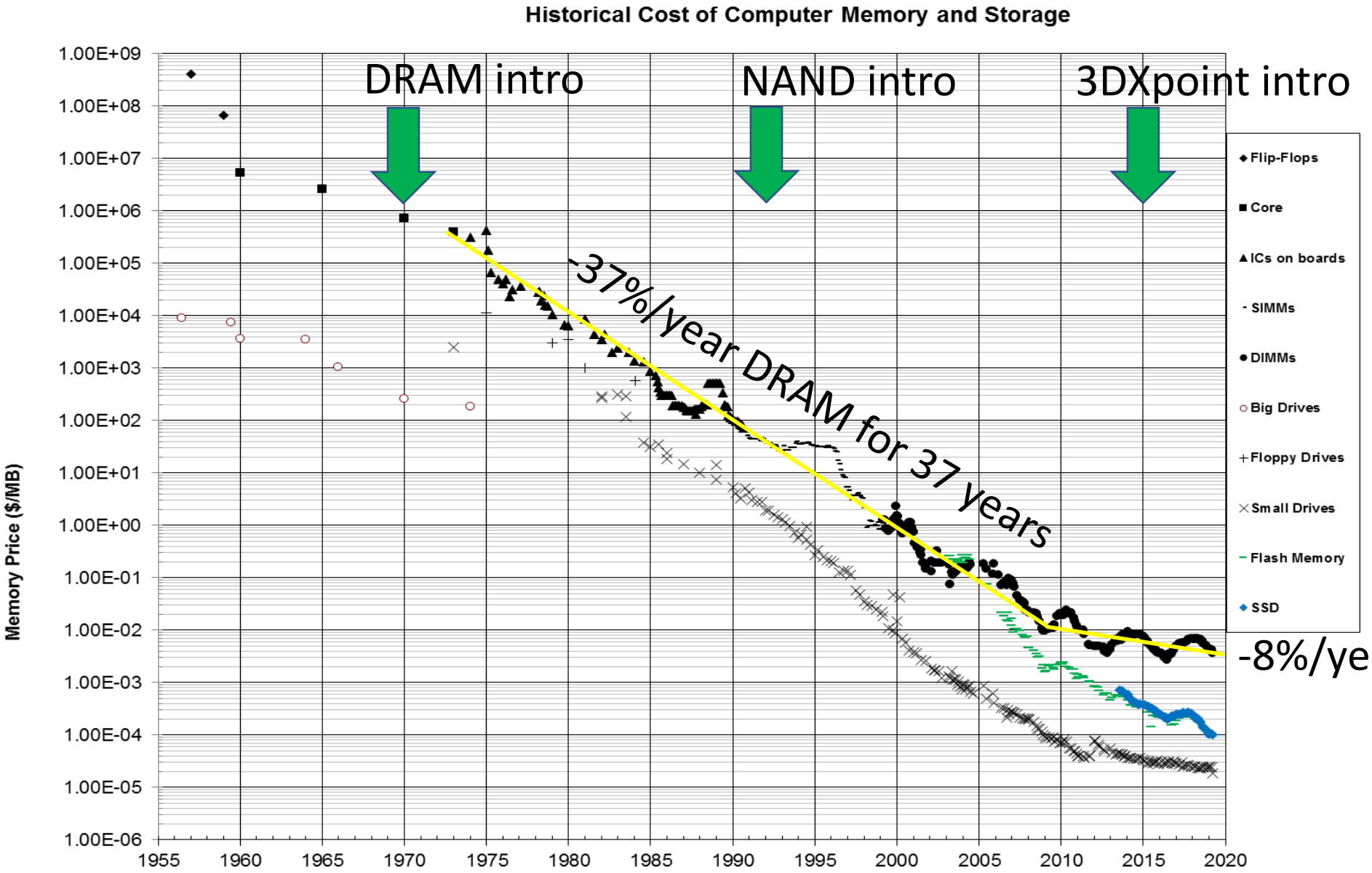
jtpawlowski@gmail.com

JTP

Outline

- Scaling, of everything
- Walls, of all sorts
- Abstraction
- Heterogeneity
- 3D CMOS
- Memory-Compute, of all sorts

Memory Price Scaling – 1957 to 2019 (market price, not cost)

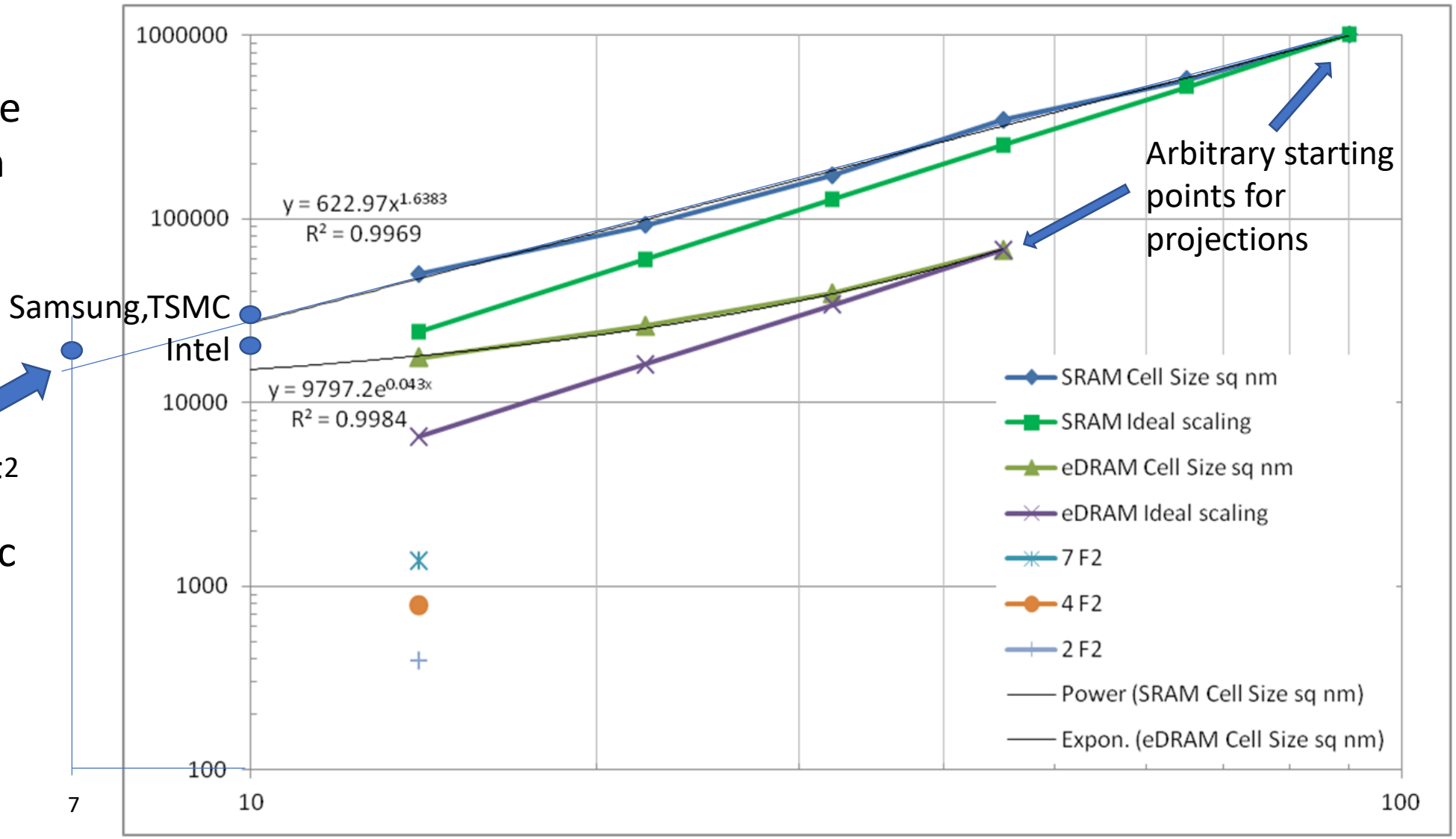


https://jcmit.net/memoryprice.htm &
J. T. Pawlowski - Where the Rubber Meets the Road, ISCA 2019

SRAM vs eDRAM Scaling

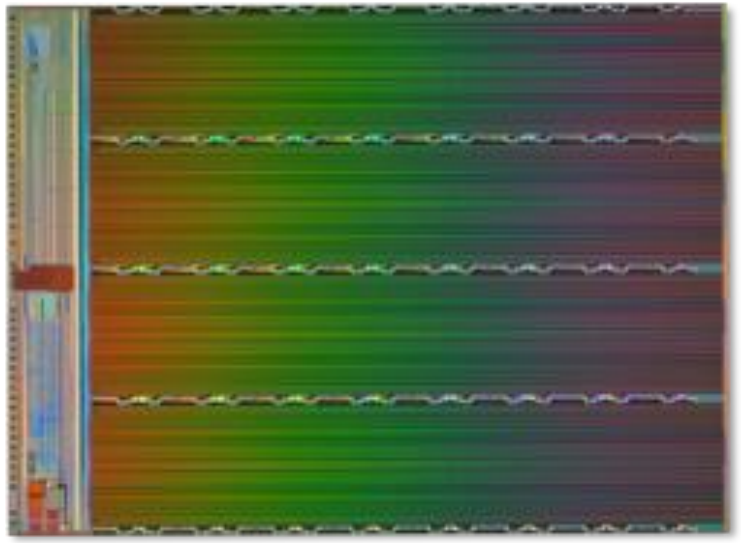
Intel High Capacity SRAM and IBM eDRAM cells shown

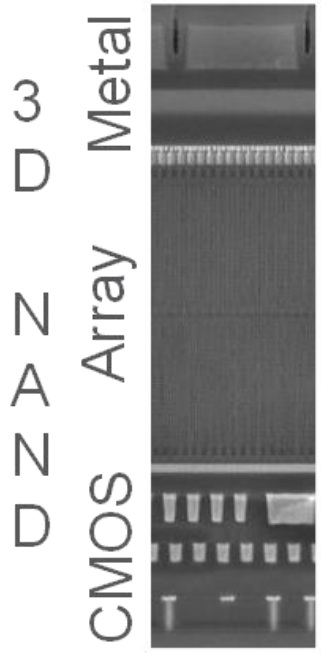
- eDRAM implemented in logic process
- SRAM 2x relative size at 14nm since 90nm
- eDRAM 2.7x since 45nm, macros scaling worse
- Samsung 7nm high capacity SRAM 531F^2
- Clearly NO economic case for any more eDRAM tech nodes

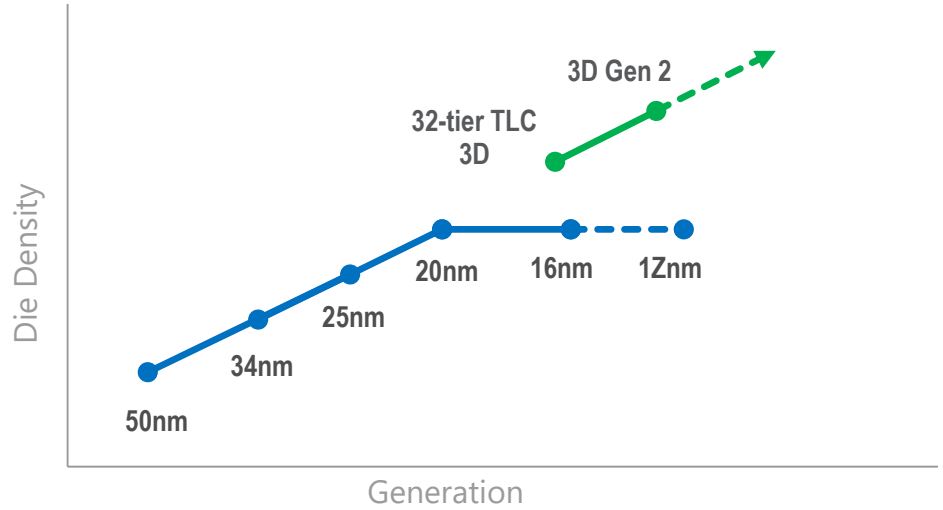


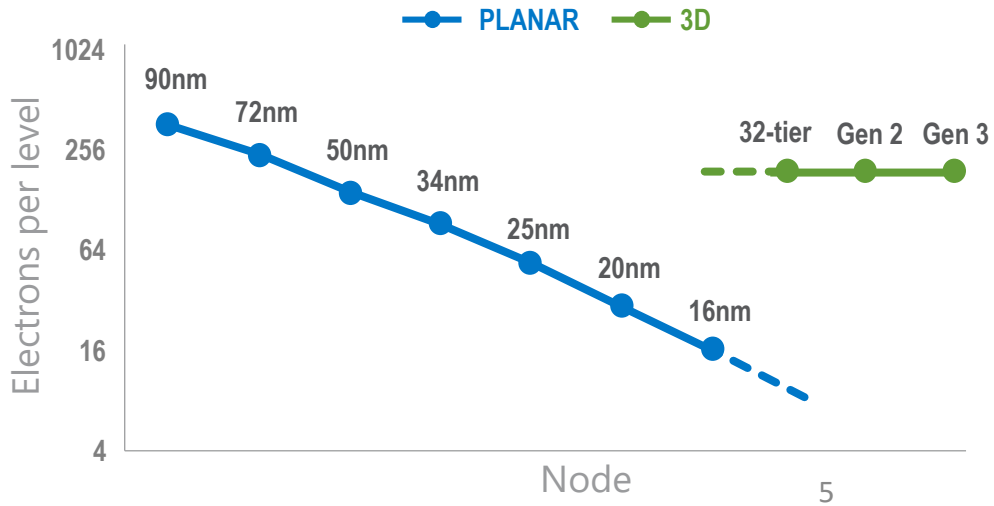
3D NAND – first monolithic 3D memory technology

- Planar NAND scaling
  - Diminished return below 16nm
  - Further shrink at TLC: could detect the loss of a single electron from the floating gate
- 3D NAND scaling
  - In 3D, the NAND chain is vertical instead of horizontal
  - Scalable and performance advantages

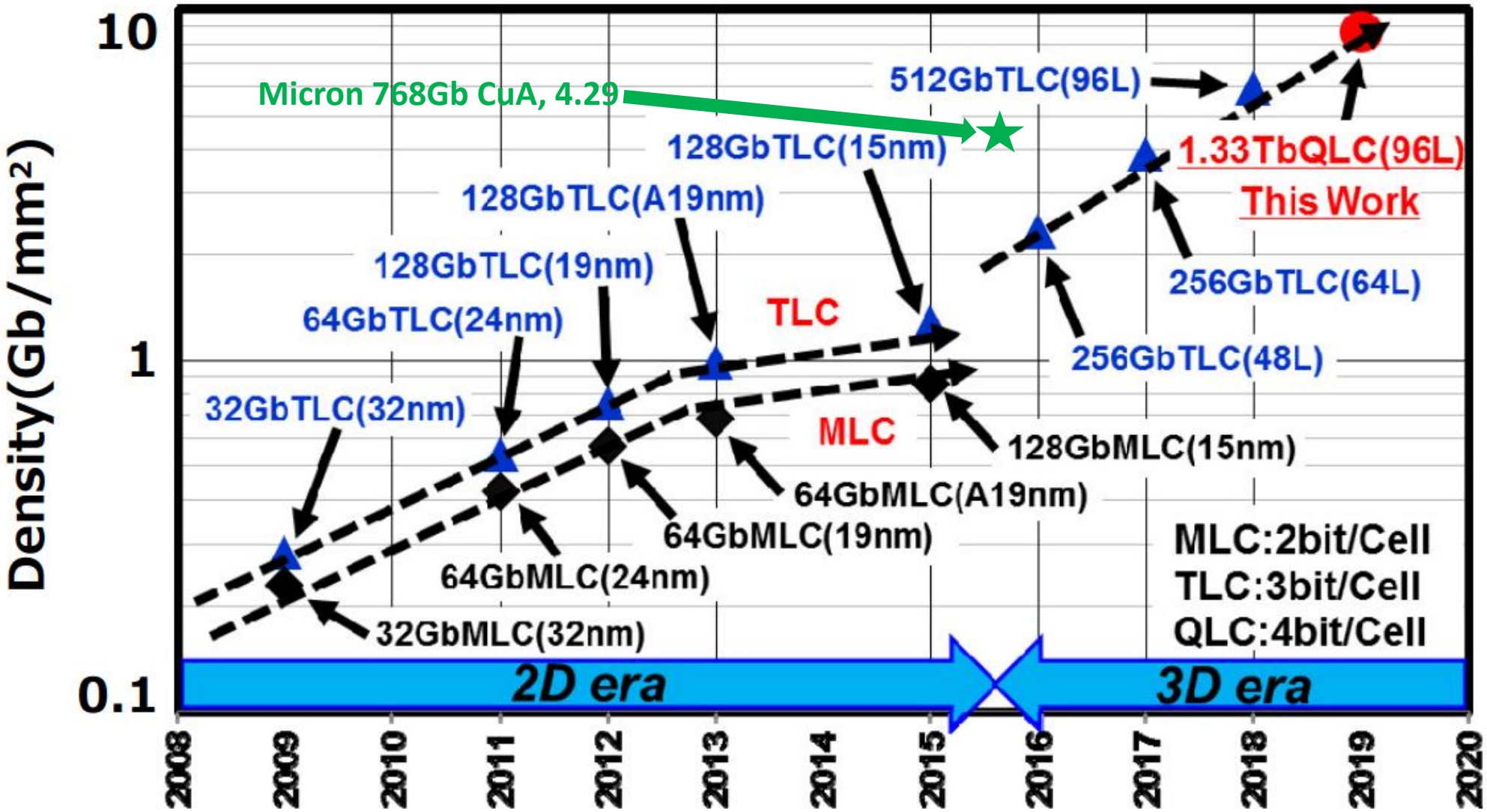






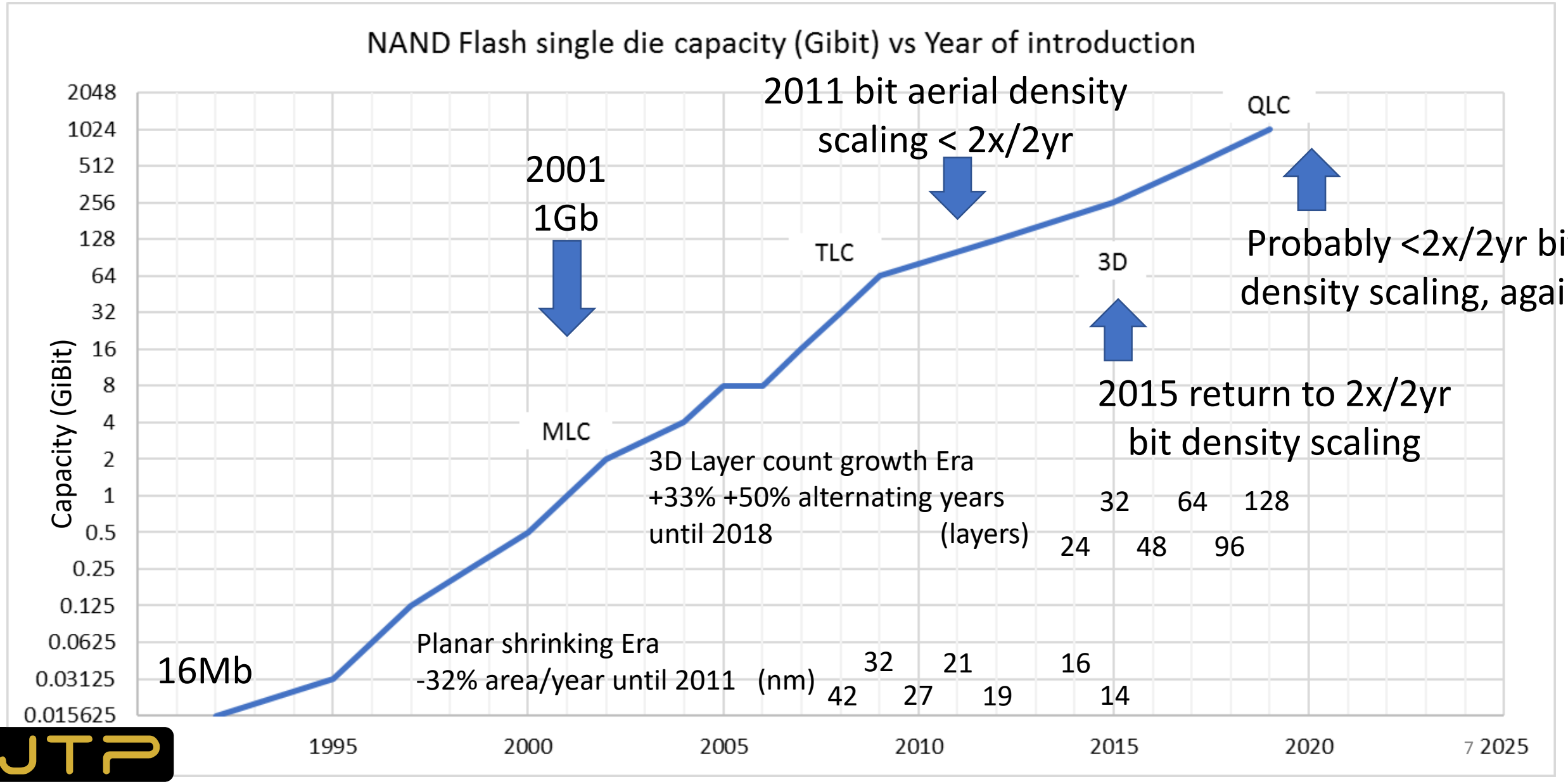


NAND Flash - Bit Density Scaling



Shibata et al, ISSCC, 2019 (Toshiba) *13.1: A 1.33Tb 4-bit/Cell 3D-Flash Memory on a 96-Word-Line-Layer Technology*

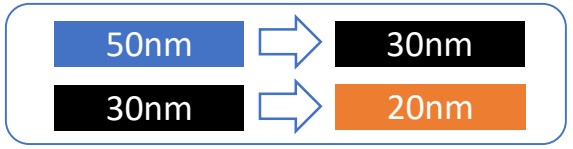
NAND Flash - Capacity Scaling

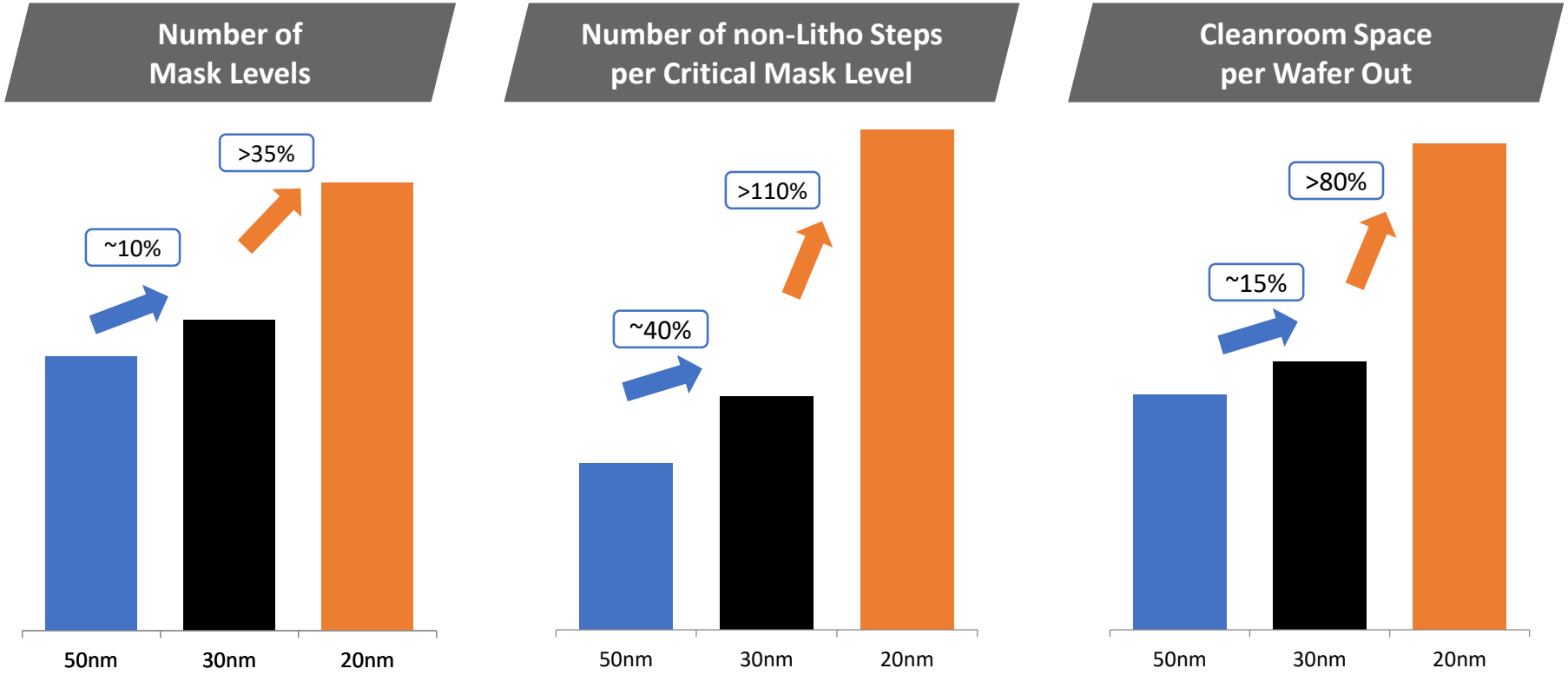


Impacts of DRAM Process Complexity

- Large increase in number of process steps to enable shrink
- Conversion Capital Expenditure scales with number of steps
- Significant reduction in wafer output per existing cleanroom area

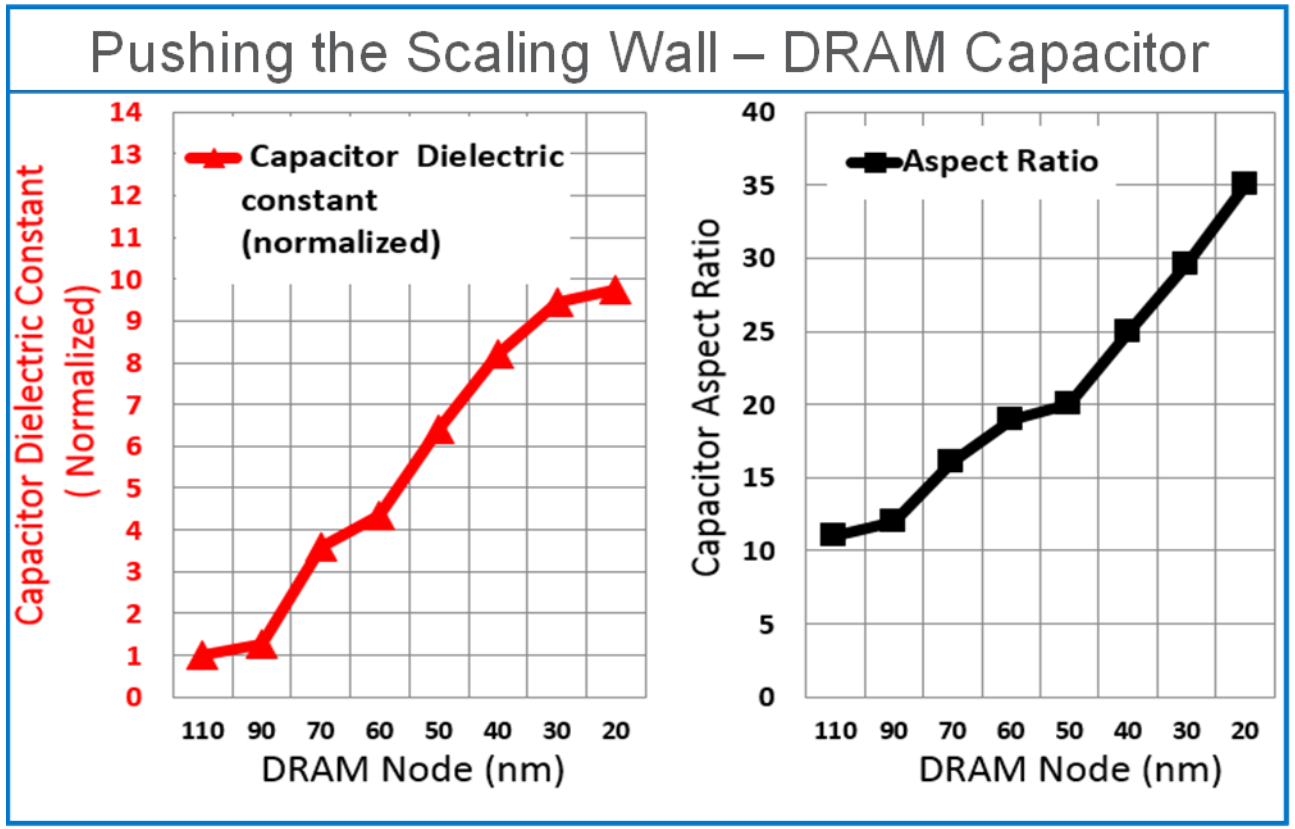
Complexity comparison for enablement of ~100% bits/wafer increase

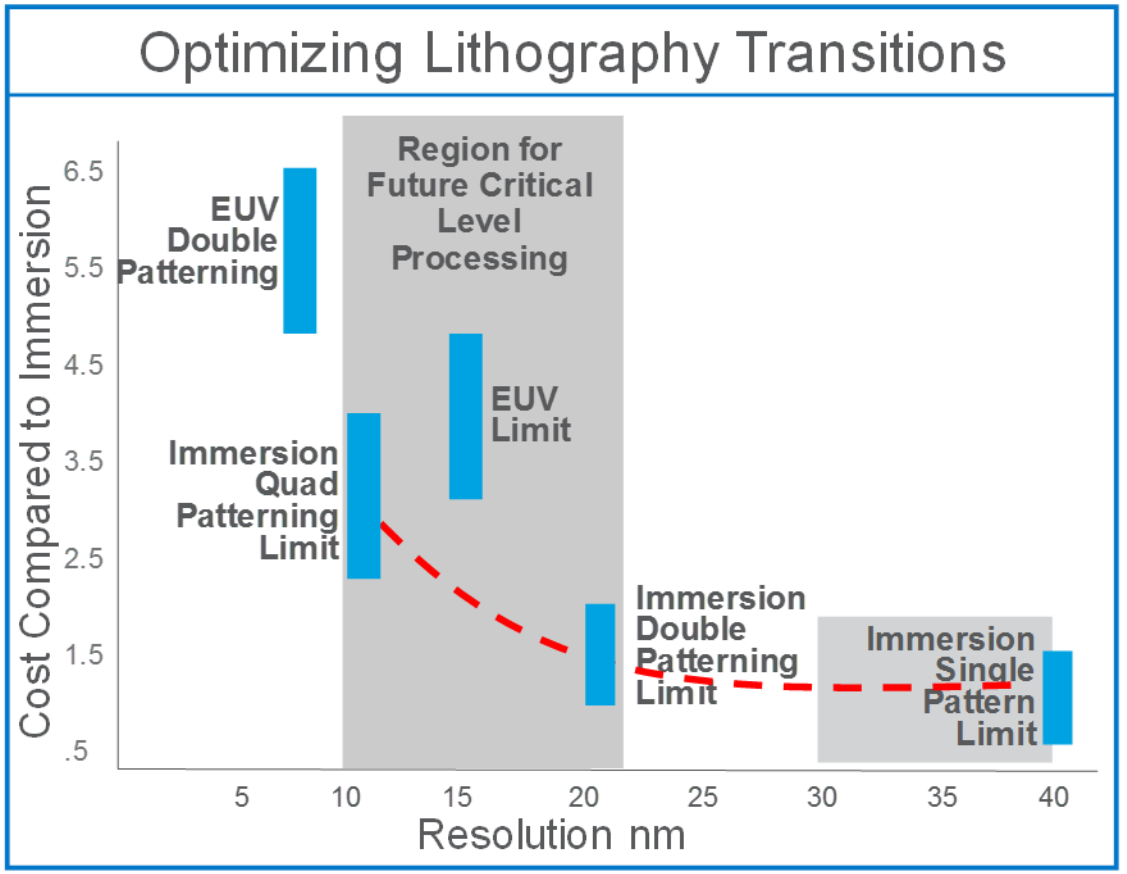




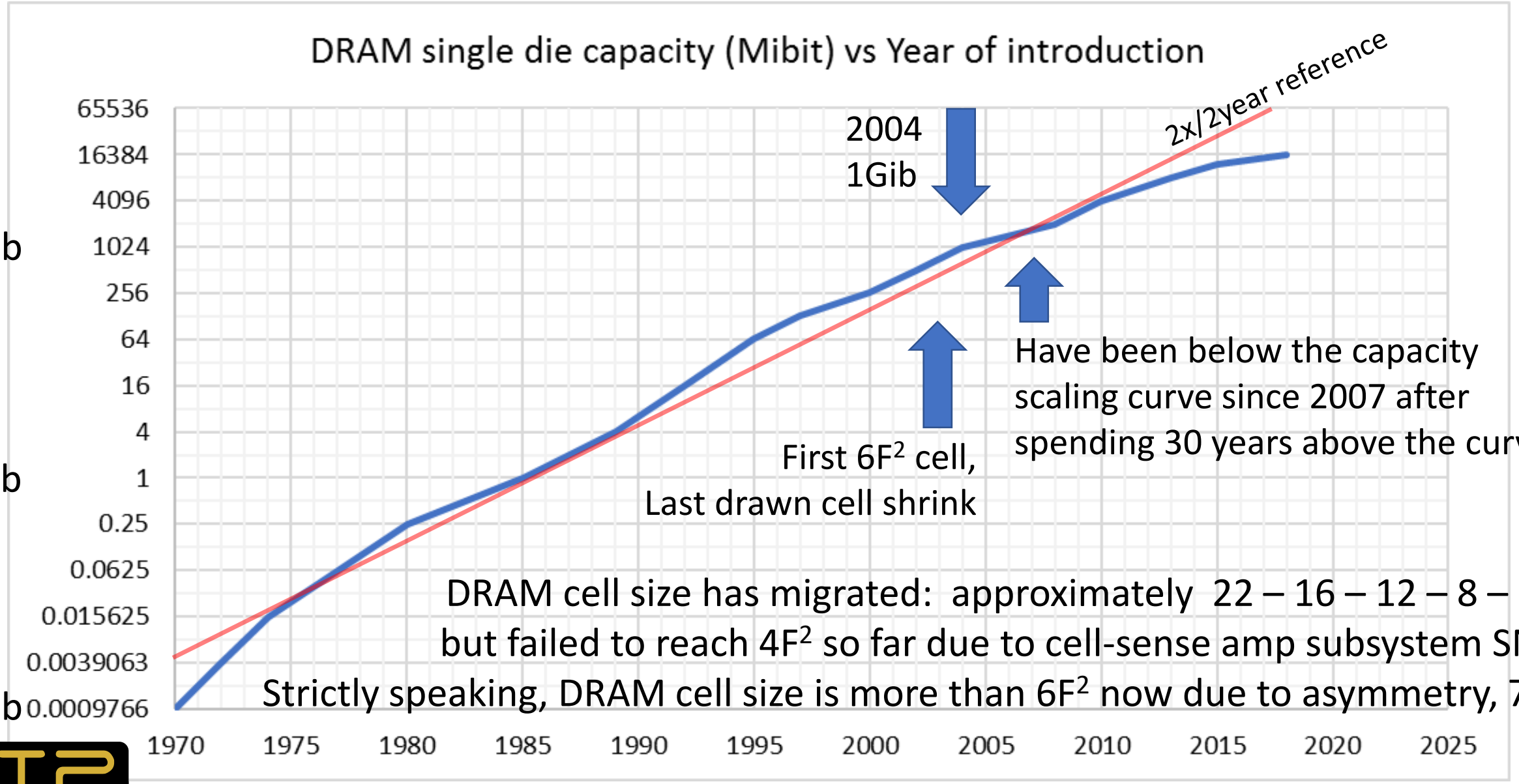
J. T. Pawlowski, IEEE MICRO keynote, 2015

DRAM Scaling is Slowing but Continuing





DRAM - Capacity Scaling

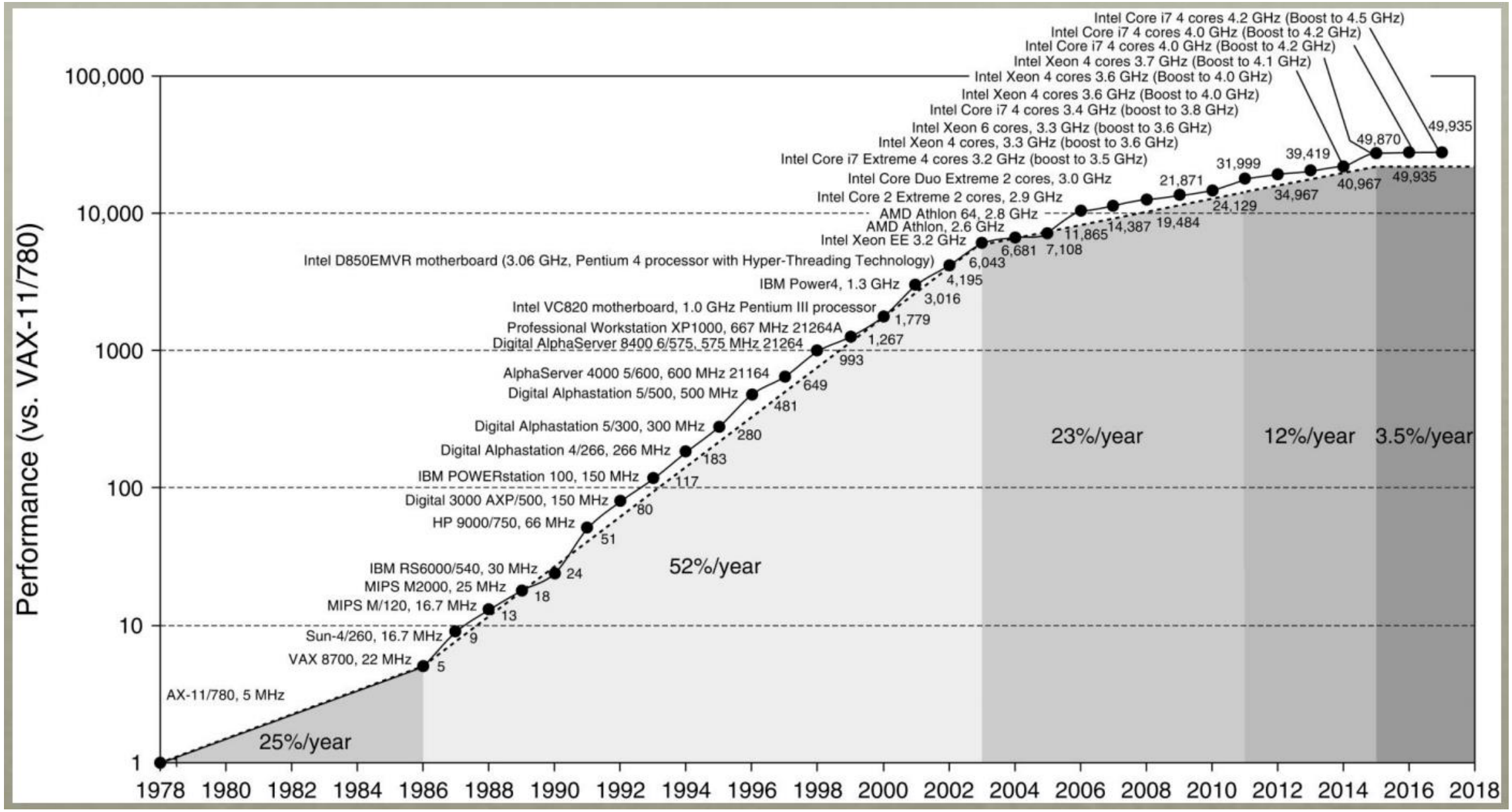


DRAM cell size has migrated: approximately 22 – 16 – 12 – 8 – $6F^2$
but failed to reach $4F^2$ so far due to cell-sense amp subsystem SNR.
Strictly speaking, DRAM cell size is more than $6F^2$ now due to asymmetry, $7F^2$.

Future of DRAM Technology

- **3D DRAM, obviously**

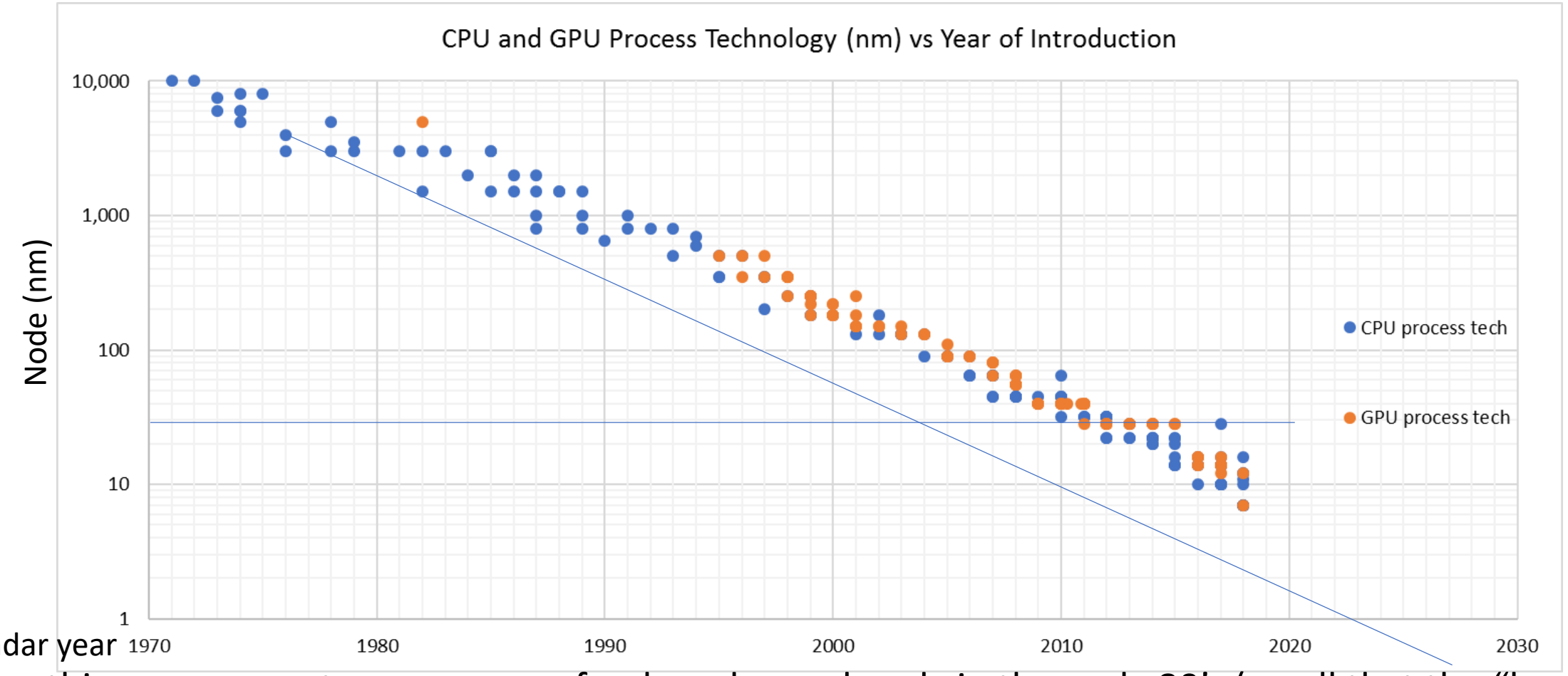
How does DRAM scaling compare to CMOS logic scaling?
Is Moore's "Law" failing catastrophically?



From Hennessy & Patterson, 2018, Single thread performance scaling

CMOS Process Technology Introduction Rate

This curve includes CPUs and GPUs



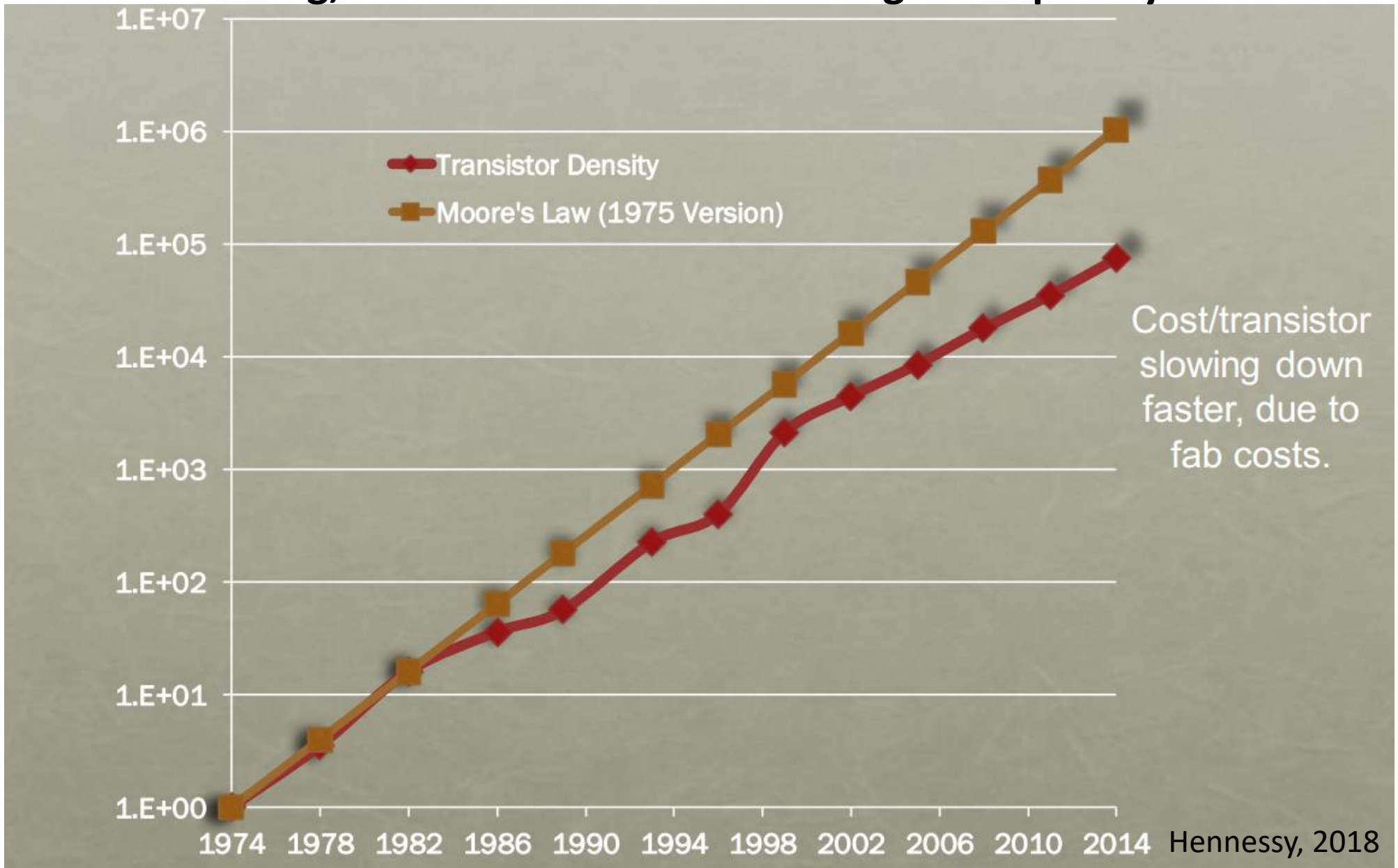
From this curve we get some sense of a slow-down already in the early 80's (recall that the "law" had to be reformulated in 1975), **BUT** down to ~28nm, nodes were named for FET gate length L. Now the naming is arbitrary for CMOS and any 3D technology including NAND; only meaningful for DRAM.

CMOS Transistor Density Growth is Slowing

And with wafer size not increasing, we are off the 2x transistor growth per 2 years now

Moore's Law, 1975 revision: economical # Tr/die doubles every 2 years.

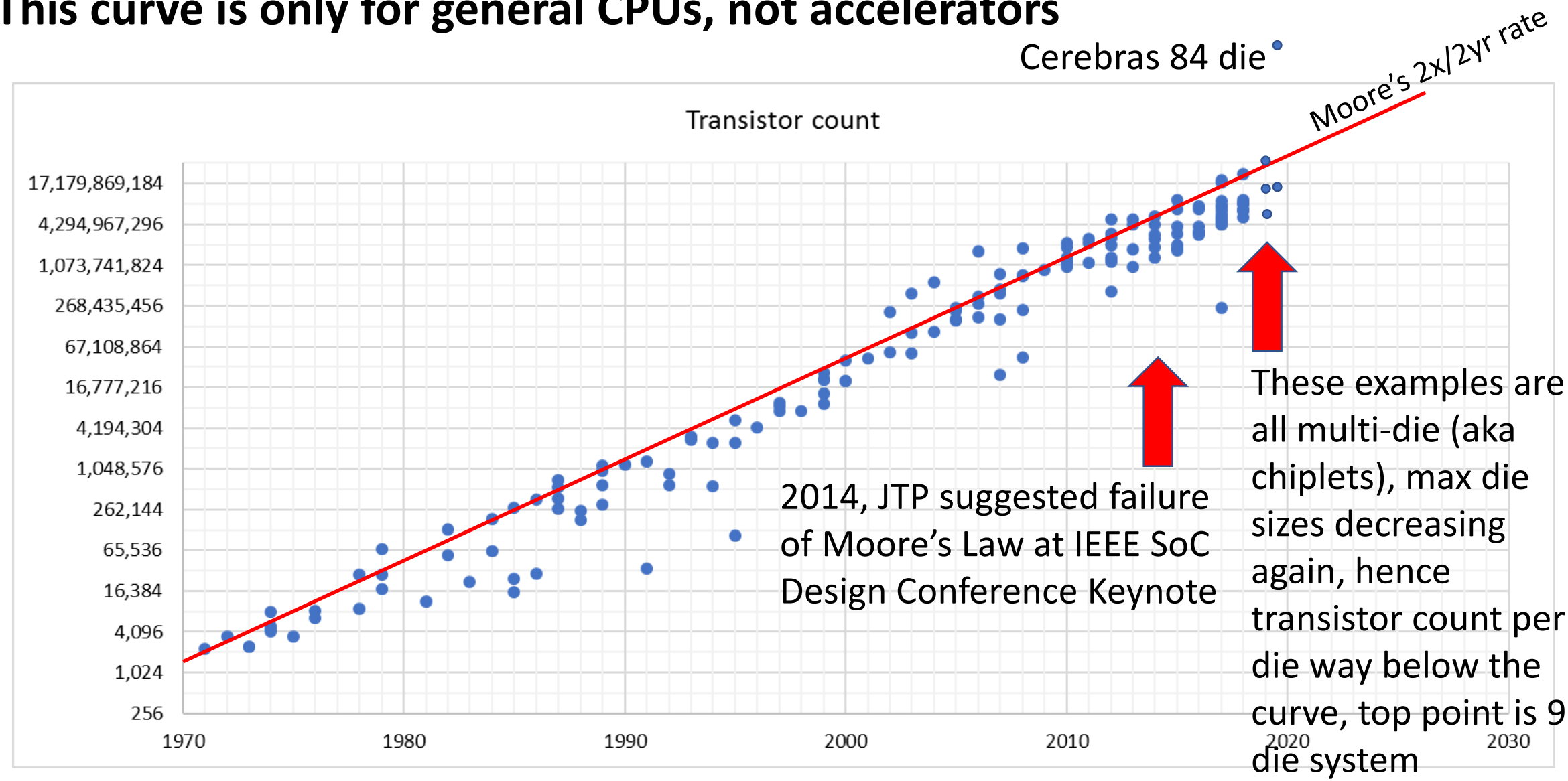
This graph does not directly address Moore's Law, does not account for die size increases.



Hennessy, 2018

CMOS Transistor Count per Die – started slow fall-off ~2012, again!

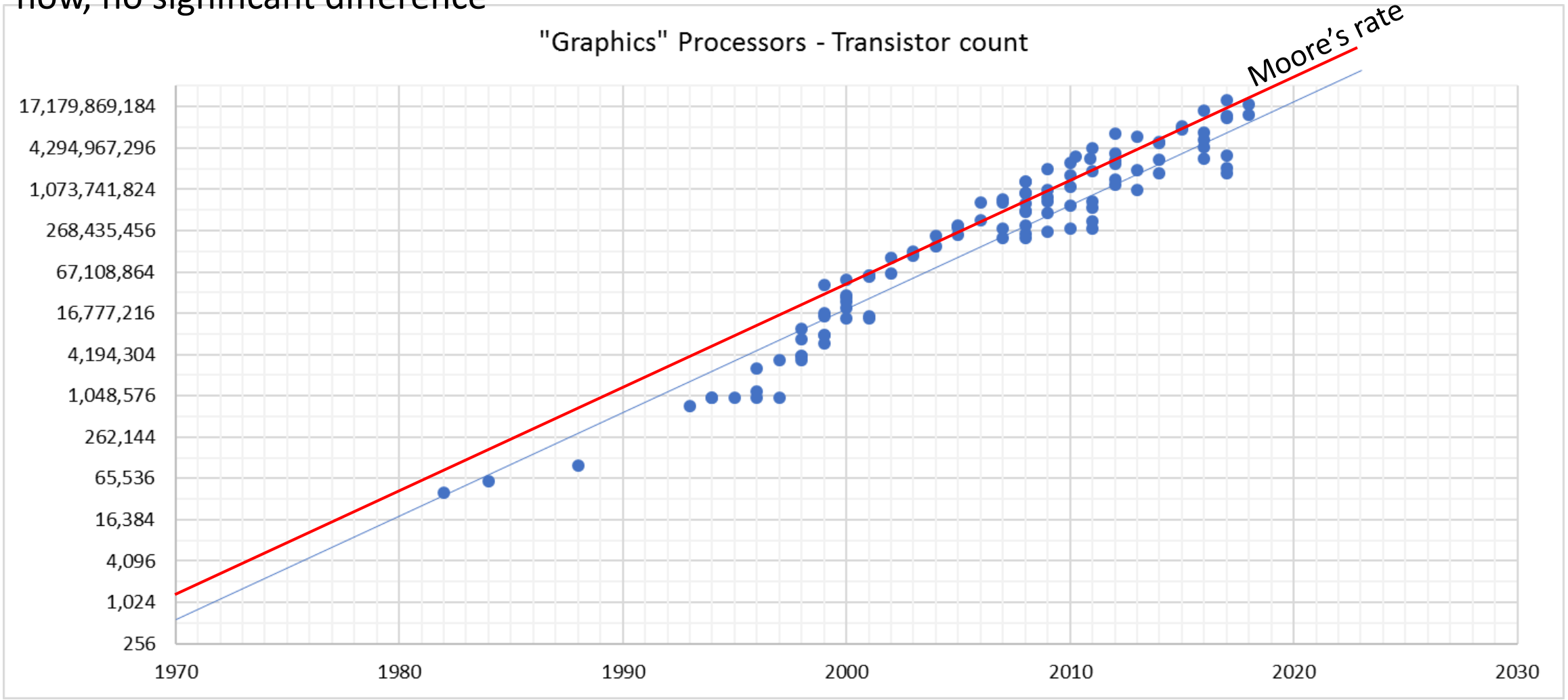
This curve is only for general CPUs, not accelerators



CMOS Transistor Count per Die – falling off after ~2013

This curve is only for GPUs

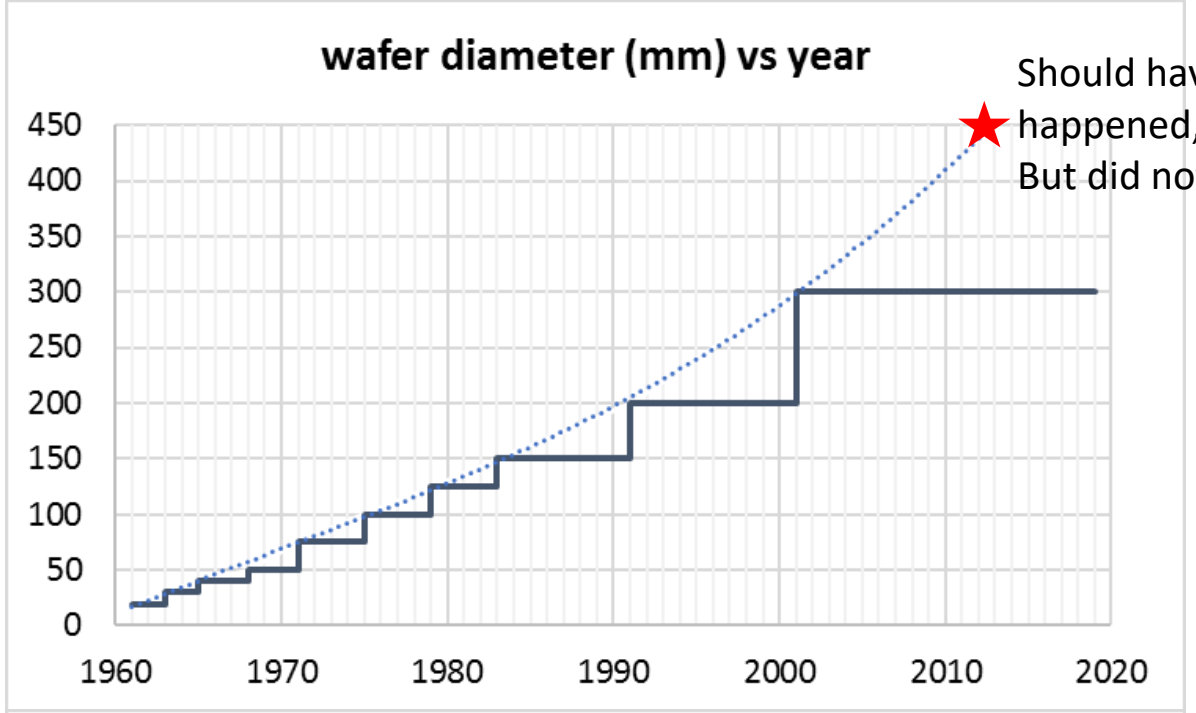
Insufficient public data available for FPGAs which used to be the densest of all CMOS logic, but now, no significant difference

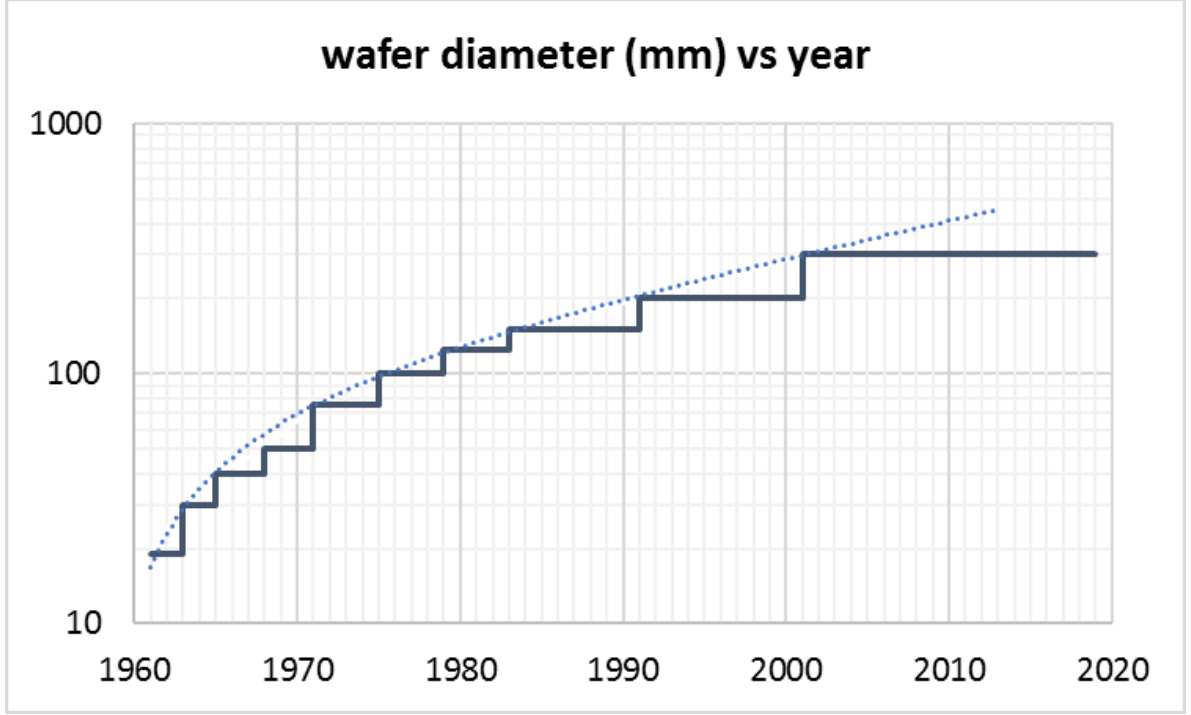


Wafer Size Scaling

Larger wafers
= Lower processing cost per unit area
& Larger reticle field
= Larger die possible
= More transistors/die
Contributed to Moore's scaling in the early days

J. T. Pawlowski, ISCA 2019

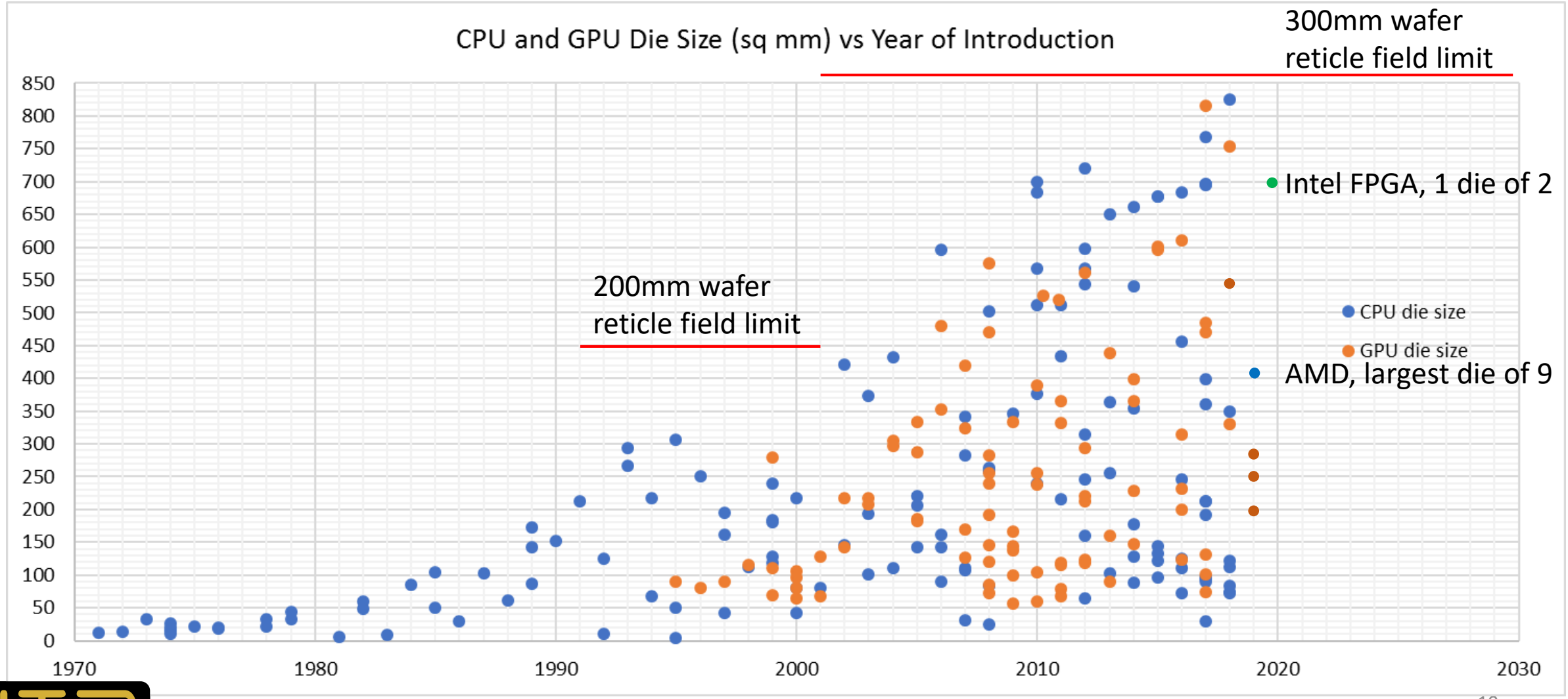




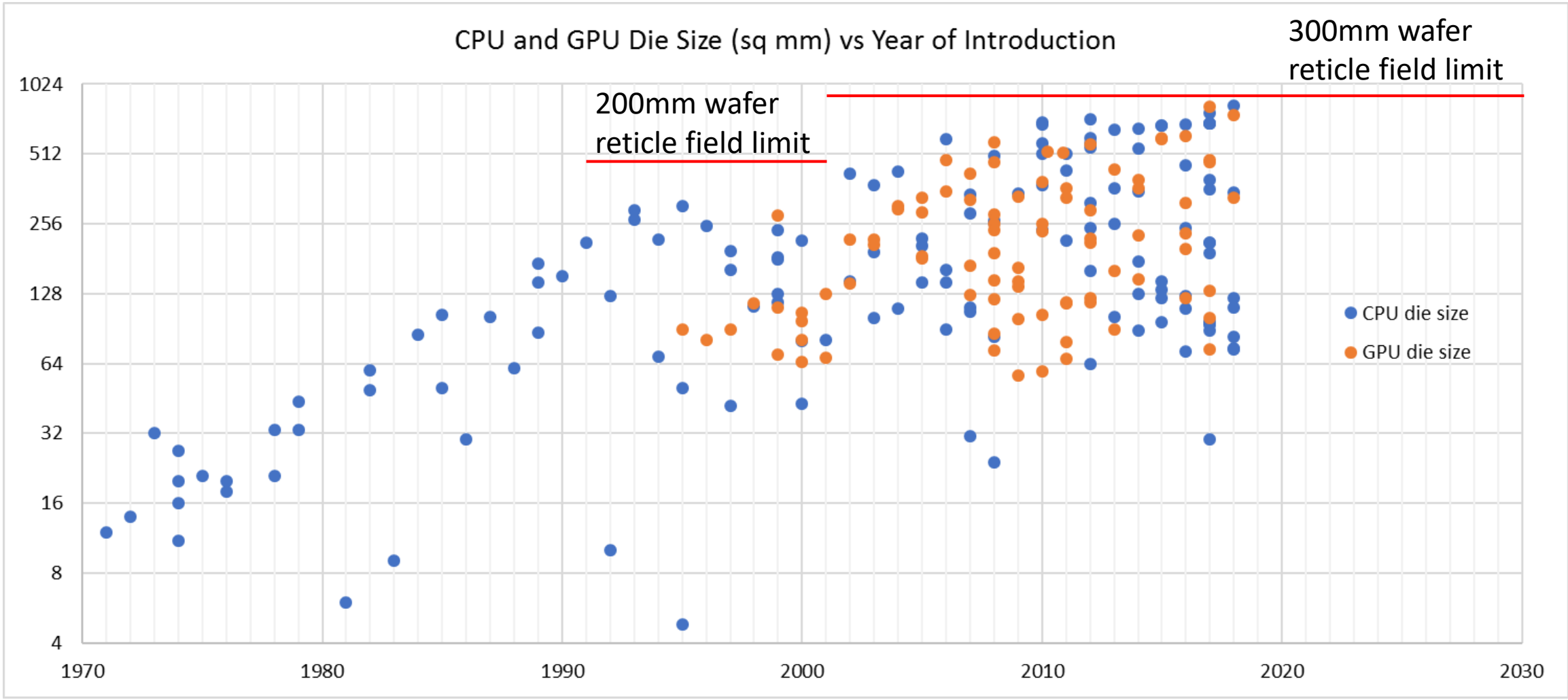
Die Size Growth has nowhere to go - reached the reticle limit

This curve includes CPUs and GPUs.

Future scaling will most likely be 3D and scale the die count both in 3D and planar



CMOS Die Size Growth on Log_2 Scale



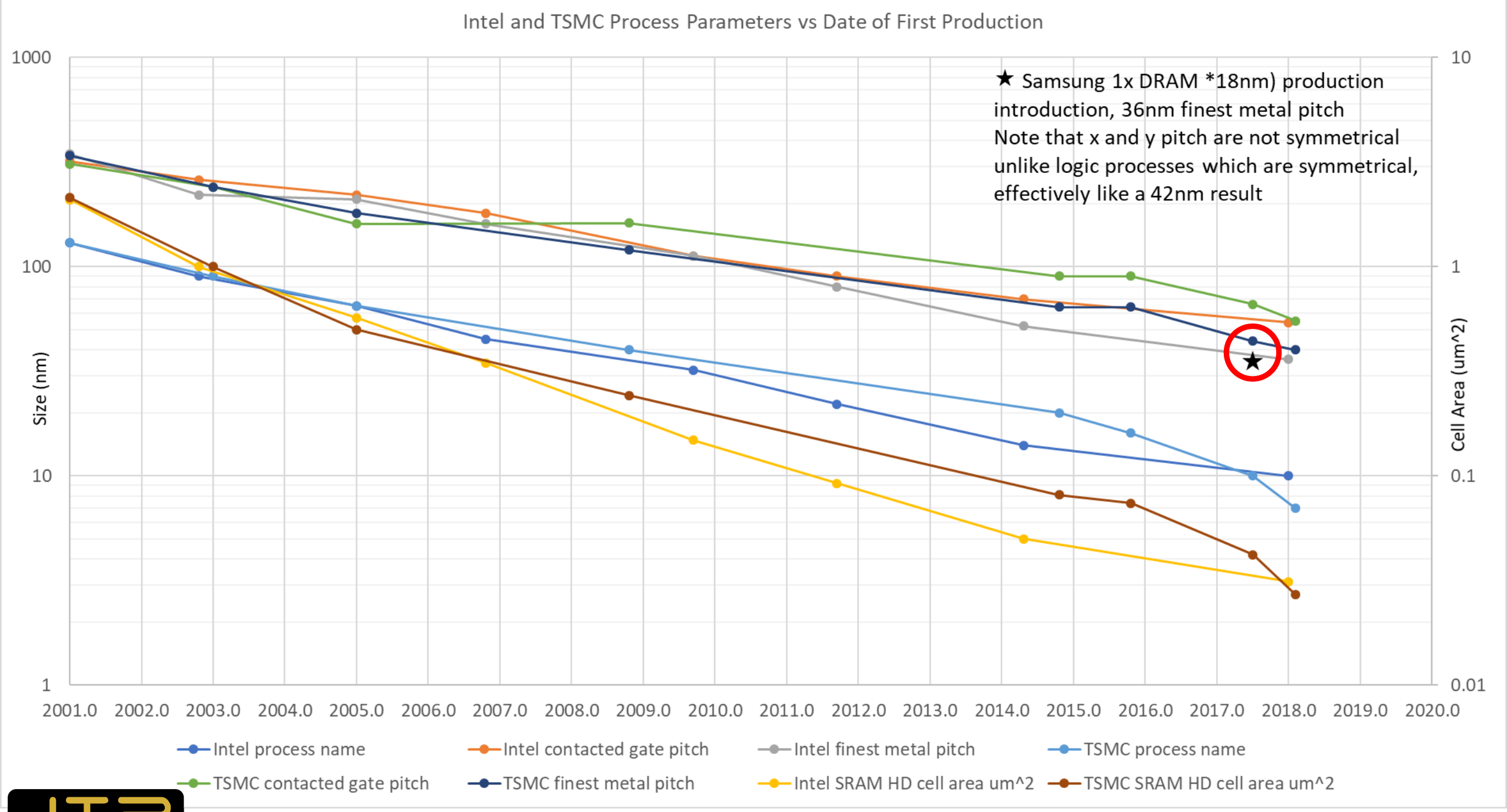
How do you relate DRAM and CMOS logic processes?

Good comparison is finest metal pitch on logic vs sqrt(xy) pitch on DRAM

DRAM processes are named by finest metal half pitch

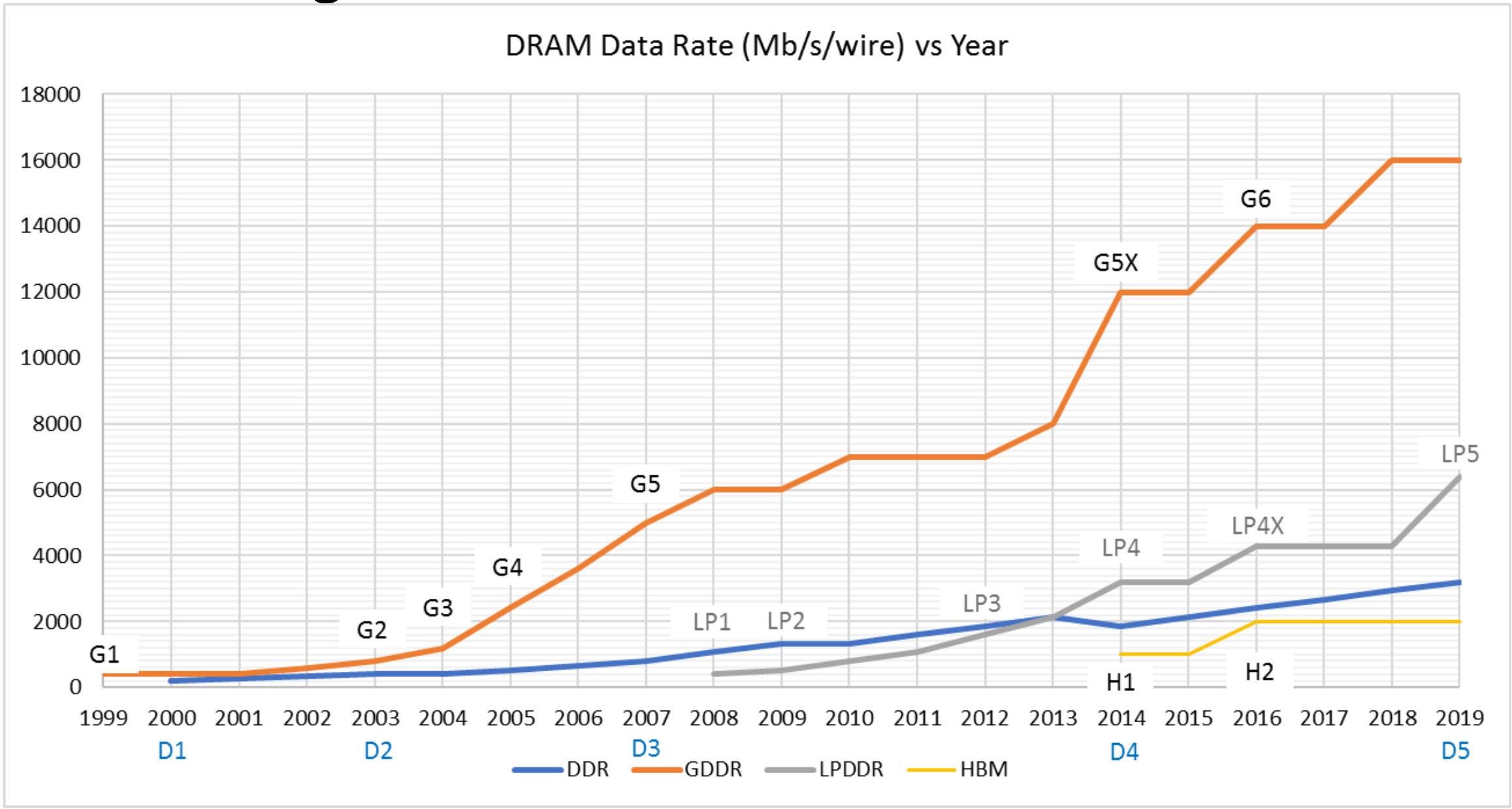
Logic process names are now marketing choices

Perhaps can show more at MemSys20

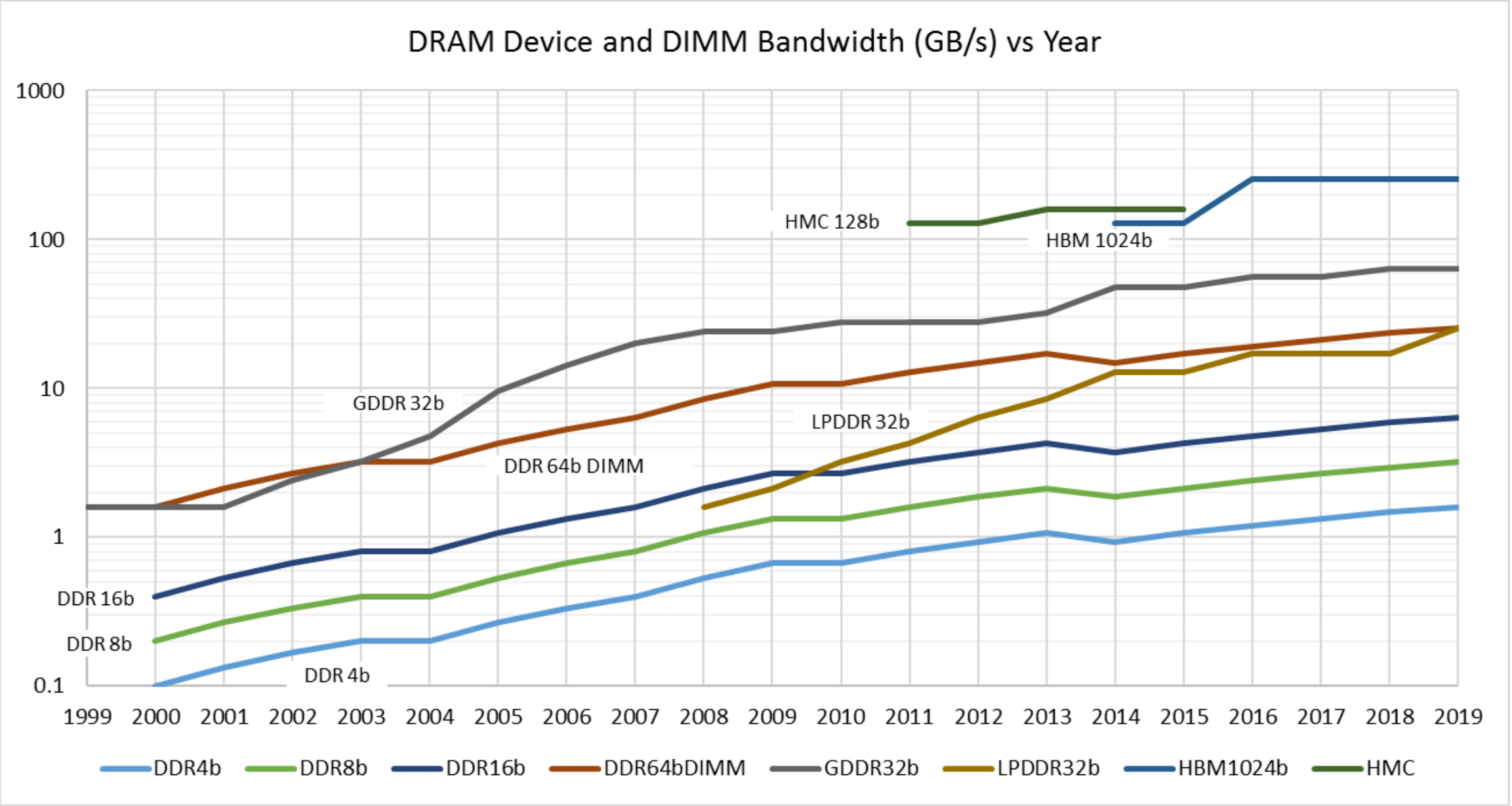

Intel and TSMC Process Parameters vs Date of First Production
★ Samsung 1x DRAM *18nm) production introduction, 36nm finest metal pitch
Note that x and y pitch are not symmetrical unlike logic processes which are symmetrical, effectively like a 42nm result
Size (nm)
Cell Area (um^2)
1000
100
10
1
10
1
0.1
0.01
2001.0
2002.0
2003.0
2004.0
2005.0
2006.0
2007.0
2008.0
2009.0
2010.0
2011.0
2012.0
2013.0
2014.0
2015.0
2016.0
2017.0
2018.0
2019.0
2020.0
Intel process name
Intel contacted gate pitch
Intel finest metal pitch
TSMC process name
TSMC contacted gate pitch
TSMC finest metal pitch
Intel SRAM HD cell area um^2
TSMC SRAM HD cell area um^2

DRAM Bandwidth Scaling

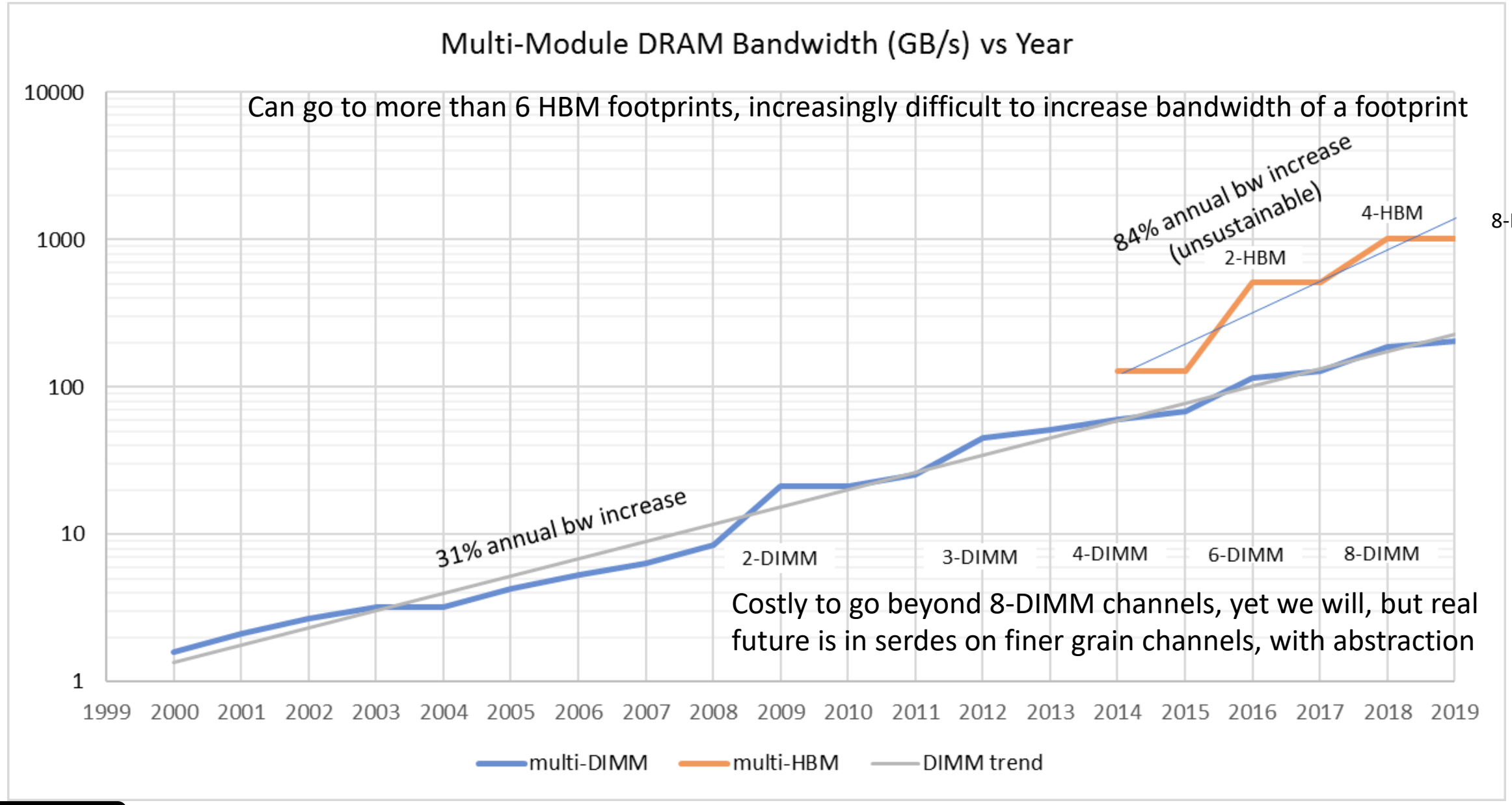
DRAM Data Signal Rate vs Year of Introduction



DRAM Single Die Footprint Peak Bandwidth vs Year of Introduction



DRAM Multi-DIMM, Multi-Footprint Peak Bandwidth vs Year of Introduction



Energy Scaling

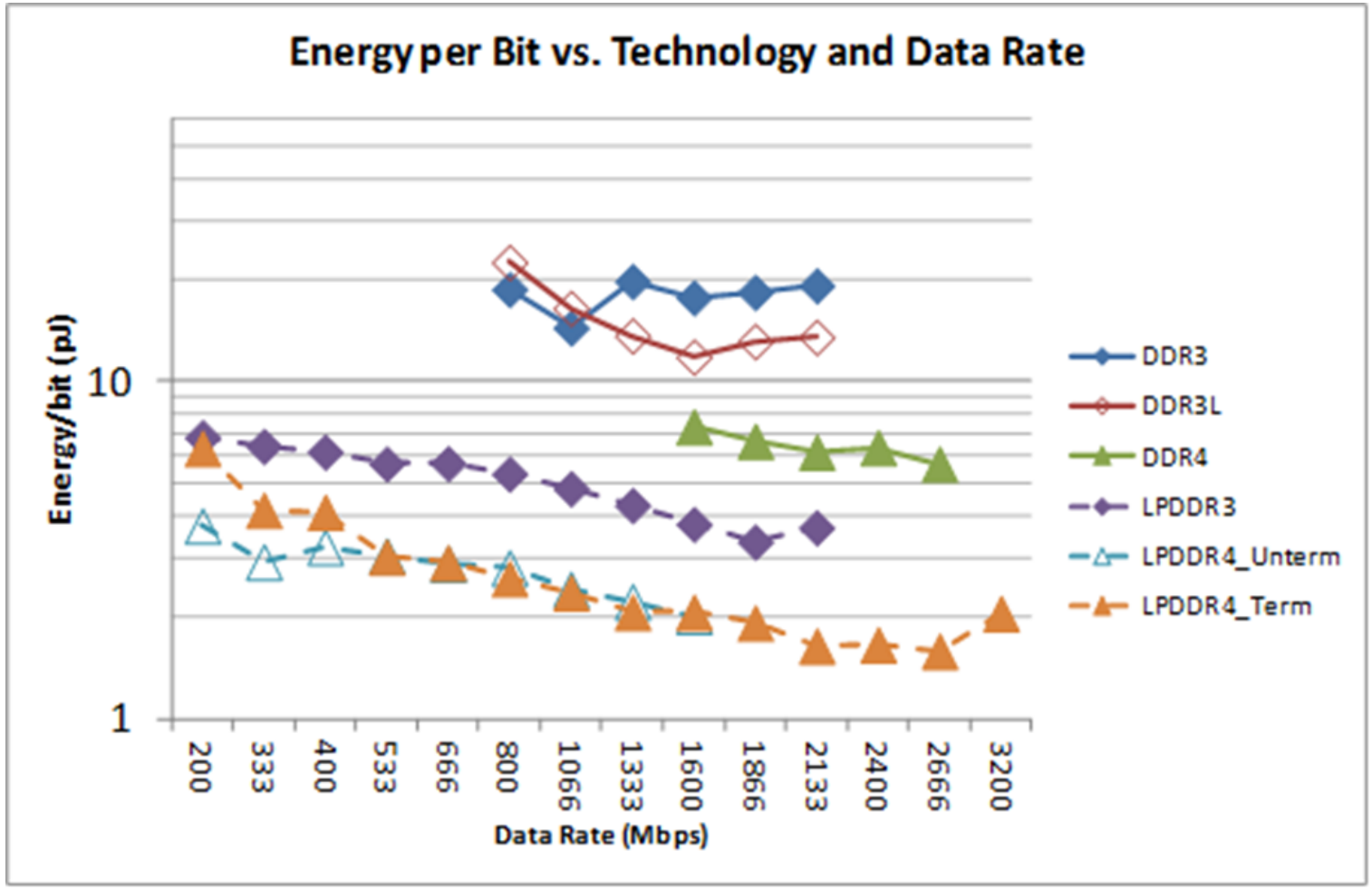
Processors and memory

Energy Observations

With the failure of Dennard scaling,
Energy is increasingly our largest issue
Data movement costs most
And don't even ask how many pJ/b for wireless !

Task (256b ops)	Energy (Without sequencing overhead)
Two 2-operand Double Precision Floating Point Operations	15 pJ (10nm logic)
Small L1 Cache SRAM Read	30 pJ (10nm logic)
10mm move on logic die	180 pJ (10nm logic)
Low-Power discrete DRAM off-chip read	1500pJ (same timeframe as 10nm logic, DRAM portion only, add ~600 for logic side)

Energy per Bit is Complex



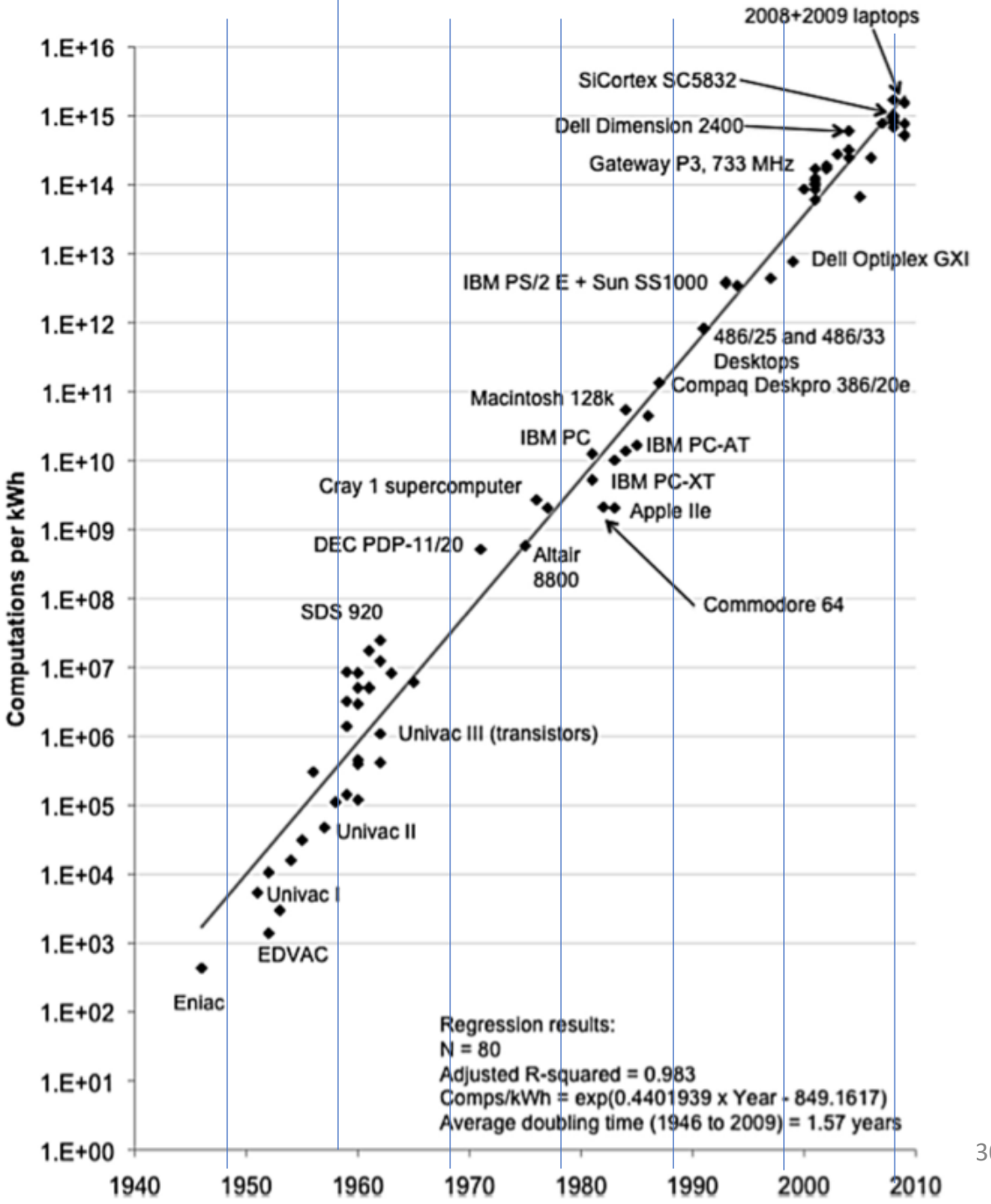
- For DRAM accesses must amortize ACT-PRE and account for CLK power
- Hence, the highest energy efficiency occurs when the whole page is transferred
- Energy/b increases as transported data decreases

- This shows just the IO energy of different architectures
- Each comprises a curve, not a point

Energy Scaling

- Energy scaling has been modest in DRAM
  - DDRx average -12.1% / year
  - LPDDRx average -9.9% / year (2008-2019)
  - Greatest reductions come with new architectures and voltage drops
- Other memory types are less energy efficient
- From the end of Dennard scaling ~2006, energy reduction has been through purposeful process technology innovations, circuit design, and chip architecture concepts
  - Now, little "natural" assistance, unlike before 2007
  - This figure shows an excellent 35.4% annual energy reduction "Koomey's Law", efficiency doubling every 2.3 years
  - In 2011 Koomey pronounced that the average doubling time since 2000 is 2.6 years

Is this true?



Still Good Evidence of Ops/W Scaling

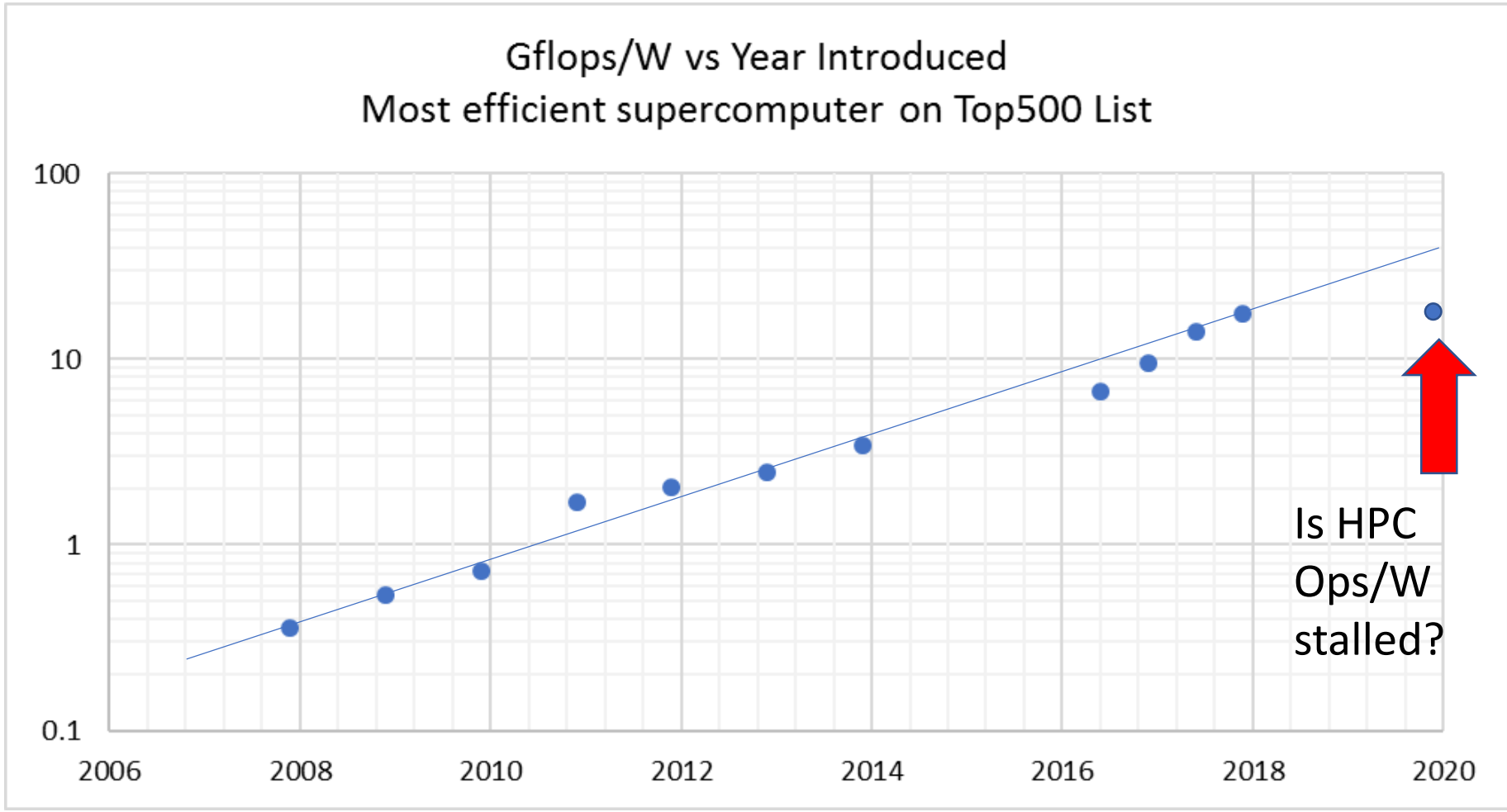
From analysis of the TOP500

Long duration job, average power is measured

Hence can be equated to compute per unit energy

Shows more aggressive Ops/W scaling than Koomey revised prediction but using unrealistic Linpack

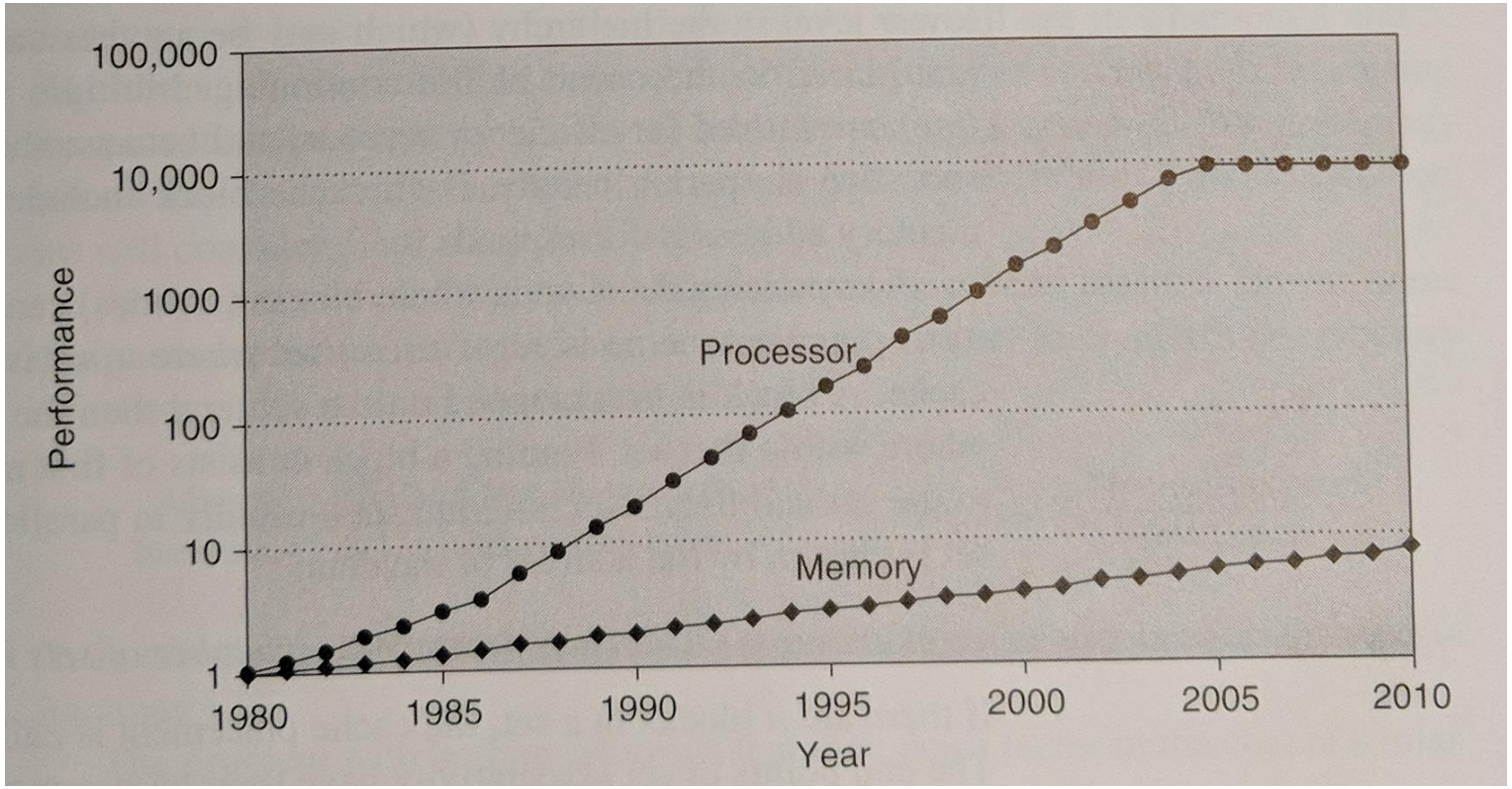
New Nov'19 Green #1 is not much improved, released today



Generally, Koomey's estimate seems reasonable

The Memory Wall

- Original paper on the Memory Wall compared processor latency with DRAM latency and showed a widening gap
- The often quoted Hennessey & Patterson figure e.g. *Computer Architecture 5th edition* page 73
  graphs the inverse of time between memory references as processor performance and the inverse of DRAM access latency as memory performance
- This is a legitimate comparison for a single core single memory system but **not useful in high core count multi-threaded systems which have many memory responders**

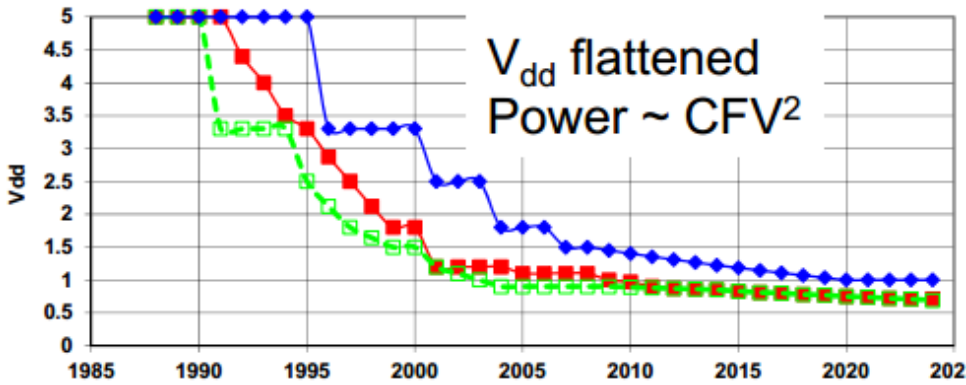


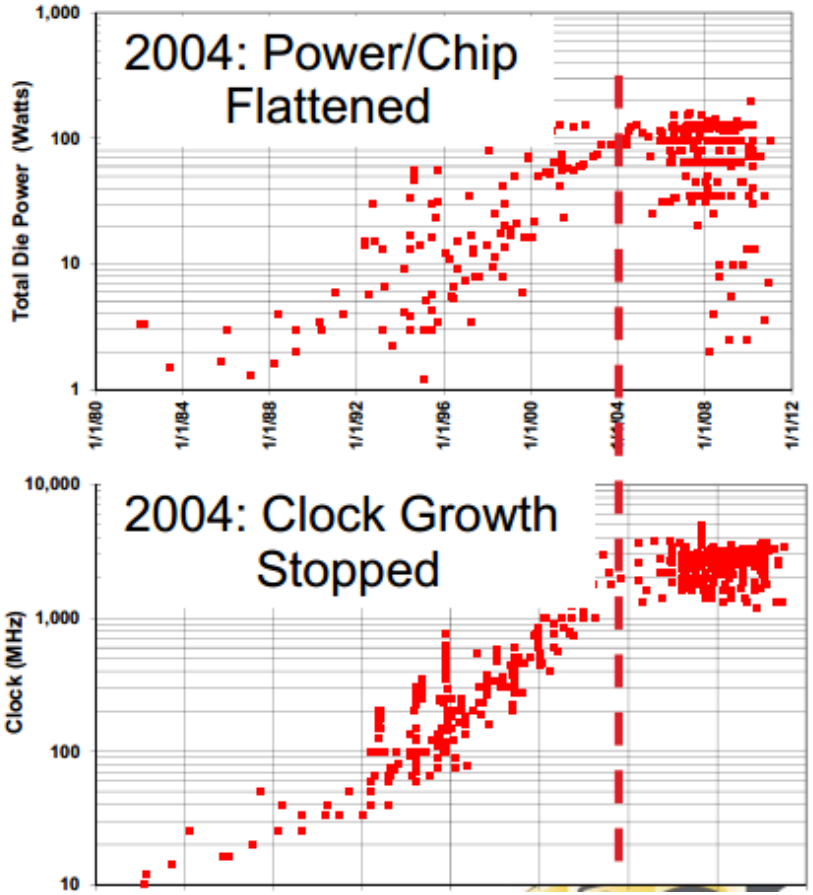
The Memory Wall

- In 1990's, began to devote transistors to SRAM cache
  - Provided more ILP, found more memory accesses, and more Flops/s
  - Enabled by applications which have plenty of locality
- Caches are here to stay
  - Even if memory suddenly became low latency and processors were radically redesigned there would still be an energy problem

The Power Wall (2004)

- Flattening V_{dd} increased power density
  - Bigger chips meant more logic to dissipate
- Result: at 120Watts, cooling uneconomical
- Breaking the wall:
  - ***Lower* the clock rate**
  - **Use *multiple* simpler cores**
  - **Increase *SIMD-style* parallelism**
- Side-effect: need more bandwidth
- Solution for dense apps: again **bigger caches**
- Is this the wall? Yes, A WALL. Not THE WALL.





P Kogge, MemSys 2017

The Wall is now the Locality Wall (PKogge) Energy Wall (JTP)

- Power is a problem, but breaks down as energy for each operation times the operation rate
- Energy increases as we lose locality and must traverse distance
- Horowitz, Asfalk, Dally, Horst, Kogge, JTP all presented similar data showing energy increase with larger structures and increasing communication distance
- The Great Wall is now Energy

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	$\mathcal{O}(10^6)$

28nm CMOS, DDR3 Greg Asfalk, HP

JTP

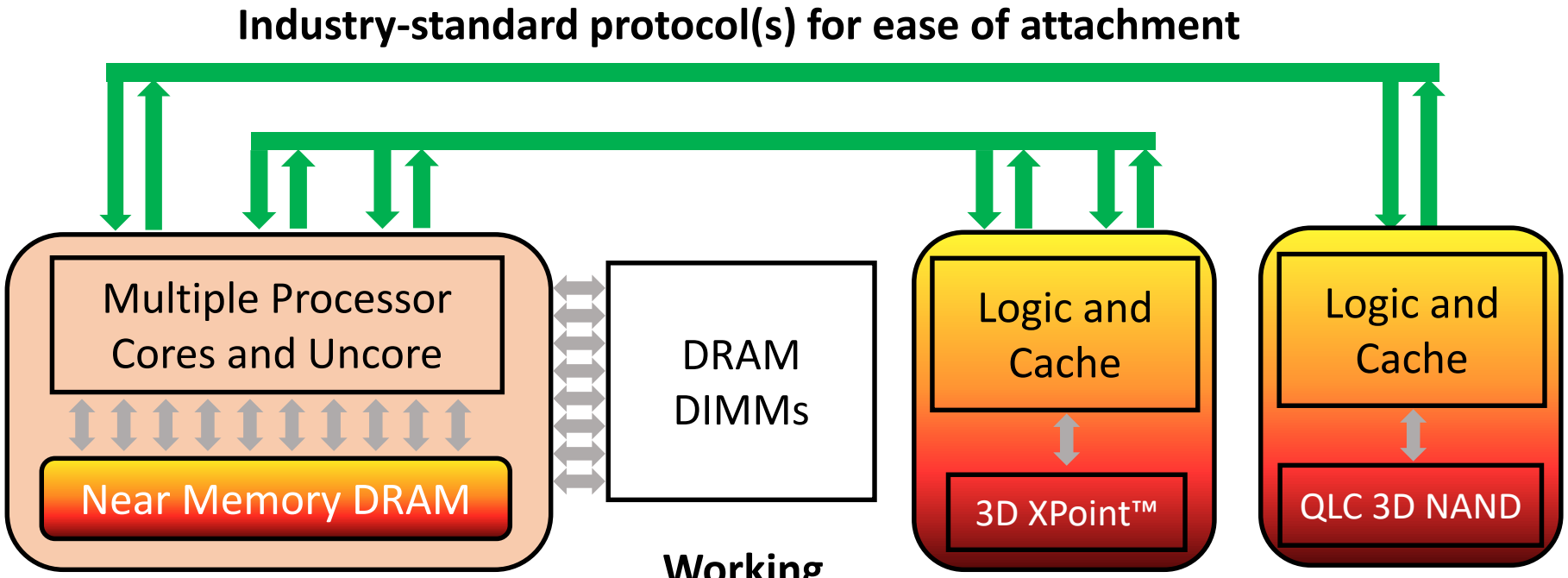
How Low Can Memory Access Energy Go?

- Today: 0.24pJ/b for a large capacity SRAM **256b** access vs. 7pJ/b for the most efficient memory (about a tie between LPDDR5 and HBM2E)
  - 4.5pJ/b for Hynix HBM2E full 1kiB page access
  - ~3.5pJ/b projected for Samsung HBM3 full page access
- By doing everything possible can get to sub-2pJ/b in high capacity DRAM
  - Potentially 1pJ/b with some major architecture improvements

Memory Abstraction

- Memory has been a second-class citizen in processor-memory systems for a long while
- There surely is potential for accelerated functions on the memory side
- Why not make it easier? Encourage innovation, nay, ENABLE IT? Mitigate the “warts”? Accommodate different technologies?
- HMC protocol inspired the world
- Gen-Z’s starting point was HMC protocol
- CCIX started shortly after Gen-Z
- Intel stymied abstraction initiatives but then finally released its response: CXL
- This is a great thing for the entire industry, enormous potential despite the PHY limitations
- Repeating my words from the first MemSys panel, 2015…
- **“shame on us if we propose another memory architecture that is not abstracted”**
- **“the degree of sensible abstraction is proportional to the latency of memory, low latency memory must be very lightly abstracted, high latency memory can be richly abstracted”**

Near-Term Future System Concept



Host with Fast Near Memory Such as HBM ~80ns Load-to-use latency (L2U)

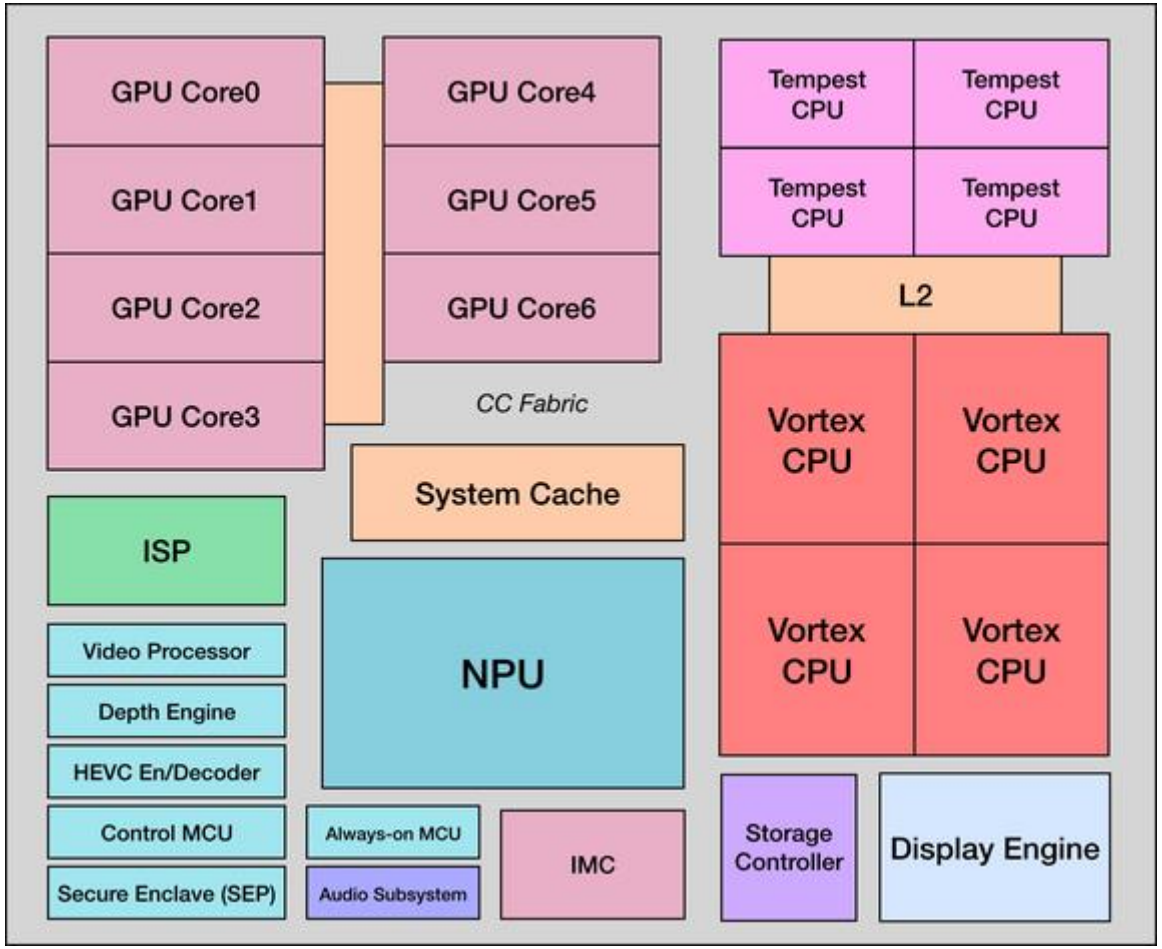
Working Memory Bulk DRAM ~80ns L2U

Far Memory 3D XPoint™? Memory Latency TBA

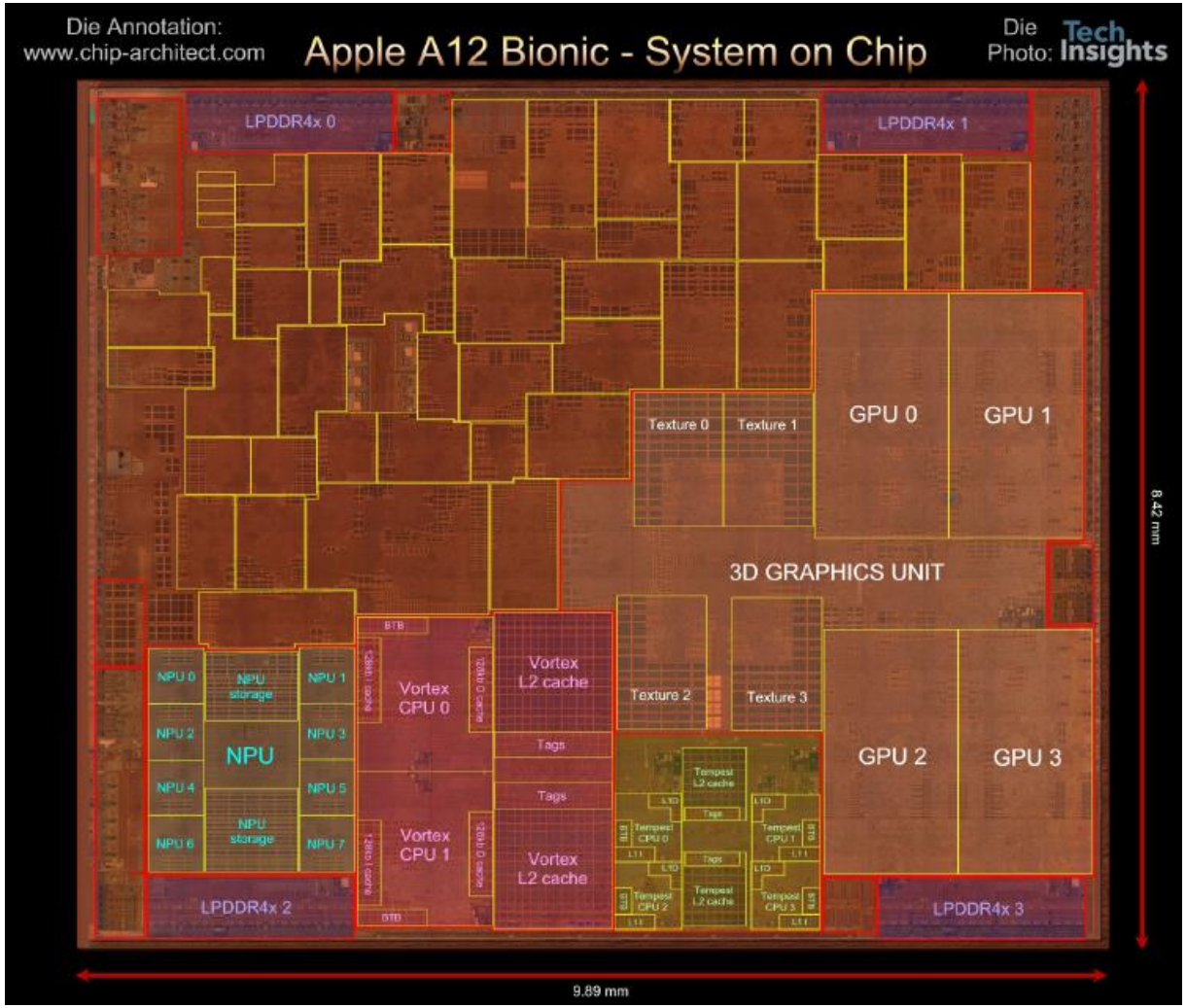
Storage SSD > 50,000 ns best L2U

Is 3D Xpoint really going to have an impact? Inspirational, but otherwise minor. Superior technologies will emerge with much greater potential, <100ns L2U system latency. Let's use 3DX only as a temporary proxy.

Systems Have Become Heterogeneous



- Apple A12X with 46 specialty accelerators
- 122mm2, >10B transistors, 7nm CMOS

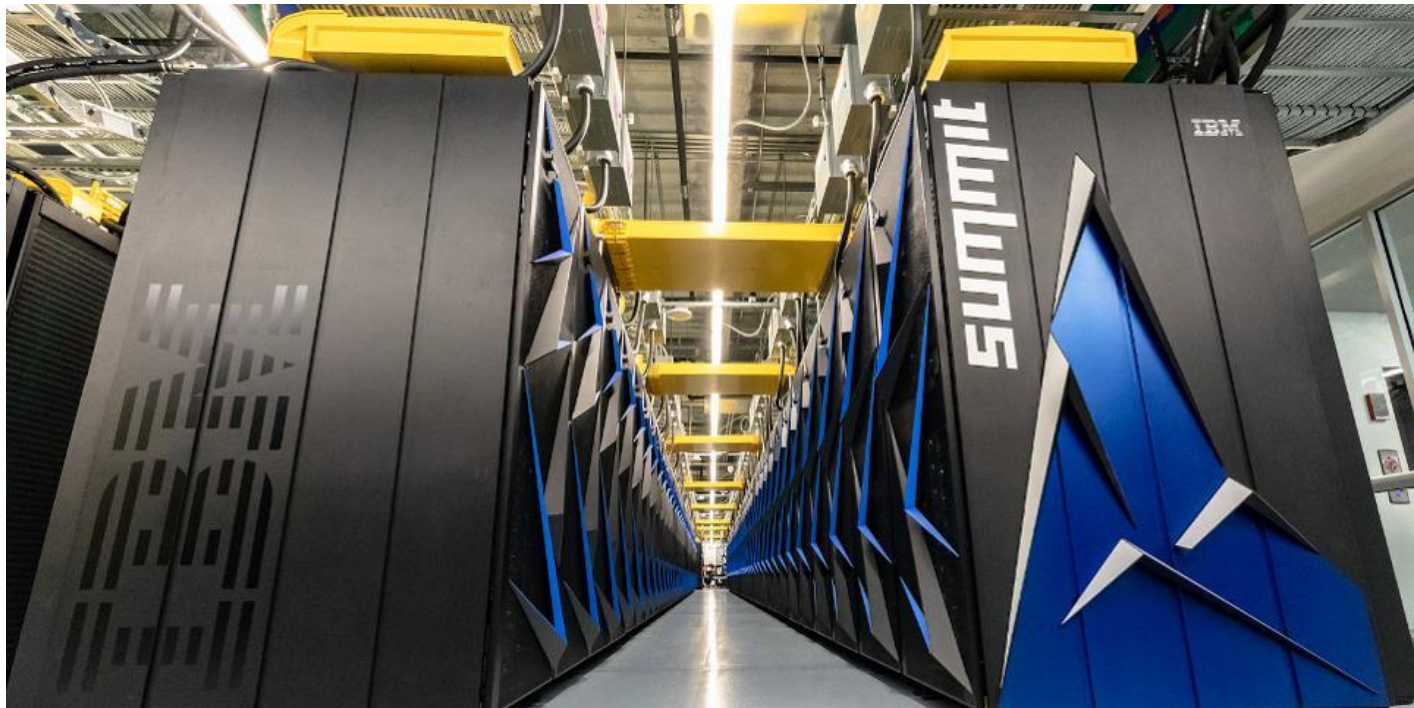


- Apple A12, 83.2mm2, 6.9B transistors, 7nm CMOS

All Top Supercomputers Are Heterogeneous

Both within their Processors/Accelerators and their Memories

- World's Top: Summit 200 PetaFlops at 14MW
- 4,608 nodes
  - 9,216 IBM POWER9 22-core CPUs and 27,648 Nvidia Tesla GPUs
  - > 95% of Flops comes from GPUs
- Over 600 GiB of coherent memory per node
  - 6×16 = 96 GiB HBM2
  - 2×8×32 = 512 GiB DDR4 SDRAM
  - Addressable by all CPUs and GPUs
- 800 GB of non-volatile RAM that can be used as a burst buffer or as extended memory
- Challenging to program system with heterogeneity
  - Not all software can be ported



Opportunity of Memory Heterogeneity

- Why have systems become heterogeneous?
  - Slowdown in frequency scaling, lithography scaling, interconnect scaling
  - Increase in specialization, problems that are embarrassingly parallel, higher intensity compute
  - Mixed compute requirements - variable access patterns, some very regular, others very irregular such as pointer chasing, indirection
  - High energy and latency of data movement - finer grain accelerators, data transformations
  - Advent of emerging memory technologies
  - Advances in packaging
- Challenges
  - Compute and Software - offload and communications overhead, variable performance, determinism, programming complexity, mapping increasingly varied workloads
  - Emerging memory explosion
    - Recent introduction of STT-MRAM, ReRAM into foundries, introduction of 3DXpoint by Intel/Micron, 5 classes of NAND Flash – fast SLC, SLC, MLC, TLC, QLC, with varied maturity and performance
  - Attaching the memory into hardware
  - Additional memory tiers presents challenge to app's software and operating system
  - Data movement between memory types
  - Widened variety of data access patterns

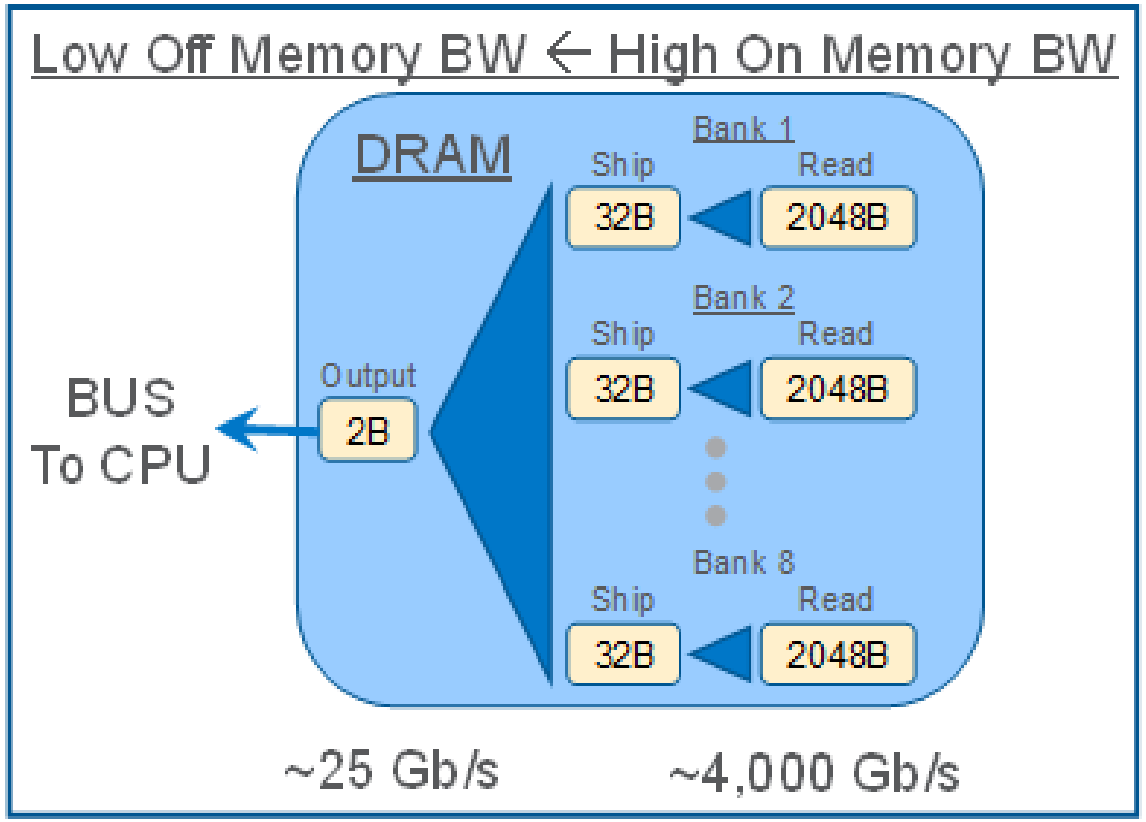
Processing In, At, or Near Memory

It is Time for 3D CMOS

- Process scaling is slowing, and we will observe more slowdown if we keep transistors in single plane
  - No Dennard's Law to save us, diminished power reductions
    - 10-20% per node now, pressure on circuit design and architecture for help
  - Reduced performance gains, domination by wires, diminishing cost reduction per process node
- Future technology improvements beyond TSVs
  - Small pitch for high-density, high-bandwidth, Cu-Cu bonding at 1um pitch without stress keep-out zones
  - Gains from system partitioning and het. integration for reduced cost and higher yields
  - Logic folding for additional energy benefits – reduced distances
- Monolithic 3D CMOS is possible
  - For what layer count?
- EDA tools will be a limiting factor
- Of course, many 3D integration concepts are in various stages of development
- The opportunity for Compute Memory is regulated by what is (or is not) done in CMOS

The Memory, the way we use Memory, and Memory Economics are all Problems

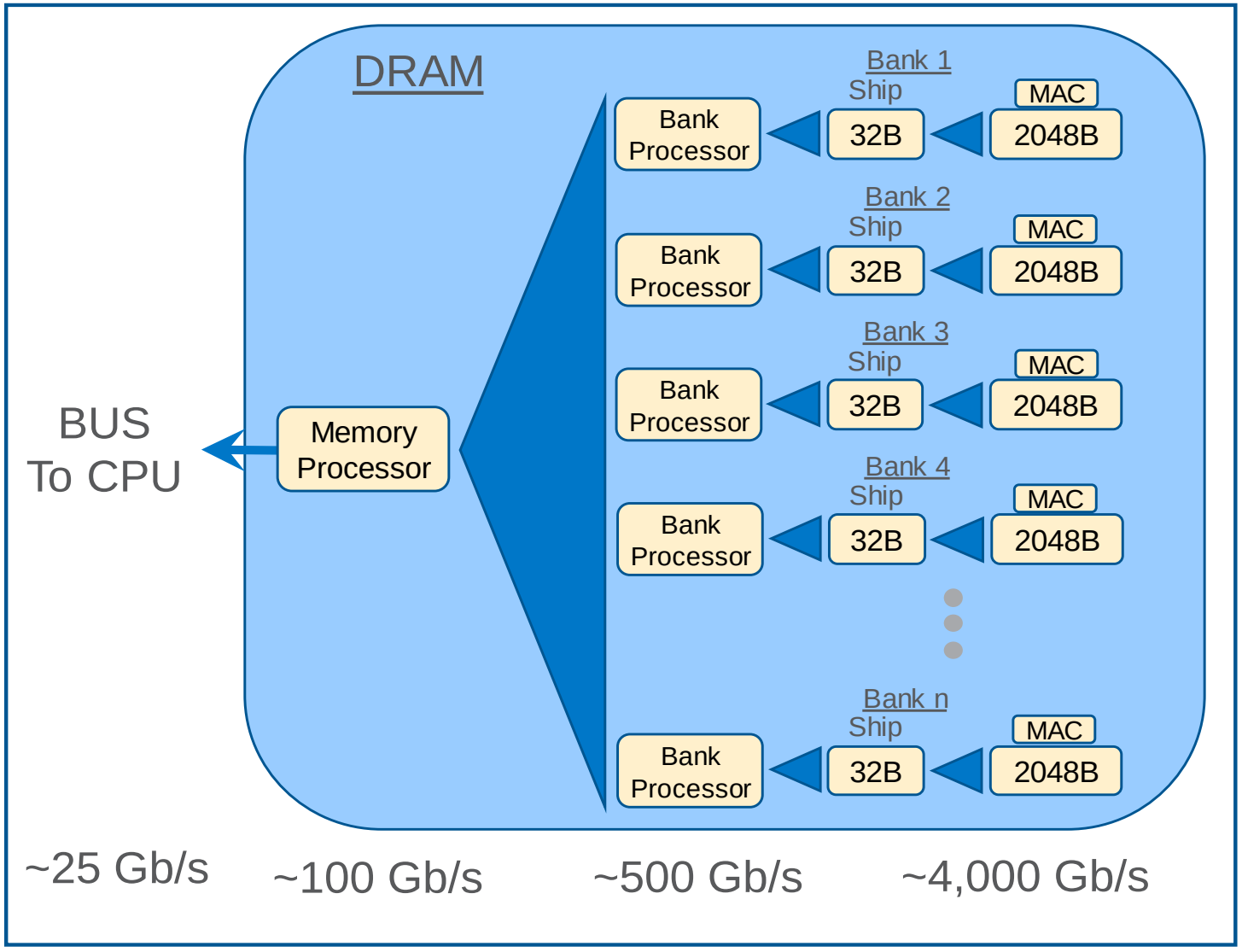
Intrinsic, on die, Memory BW is high, but is constrained by the off die system bus



- If we stay with today's paradigm, the memory bottleneck continues
  - Memory energy is interconnect dominated
  - Memory bandwidth is pin and locality constrained
- A tighter integration of compute and memory is a possible path forward
  - Reduce external bw requirement
  - Employ internal compute parallelism

Compute Deeper in Memory

Move Compute Primitives onto the Memory “Core”

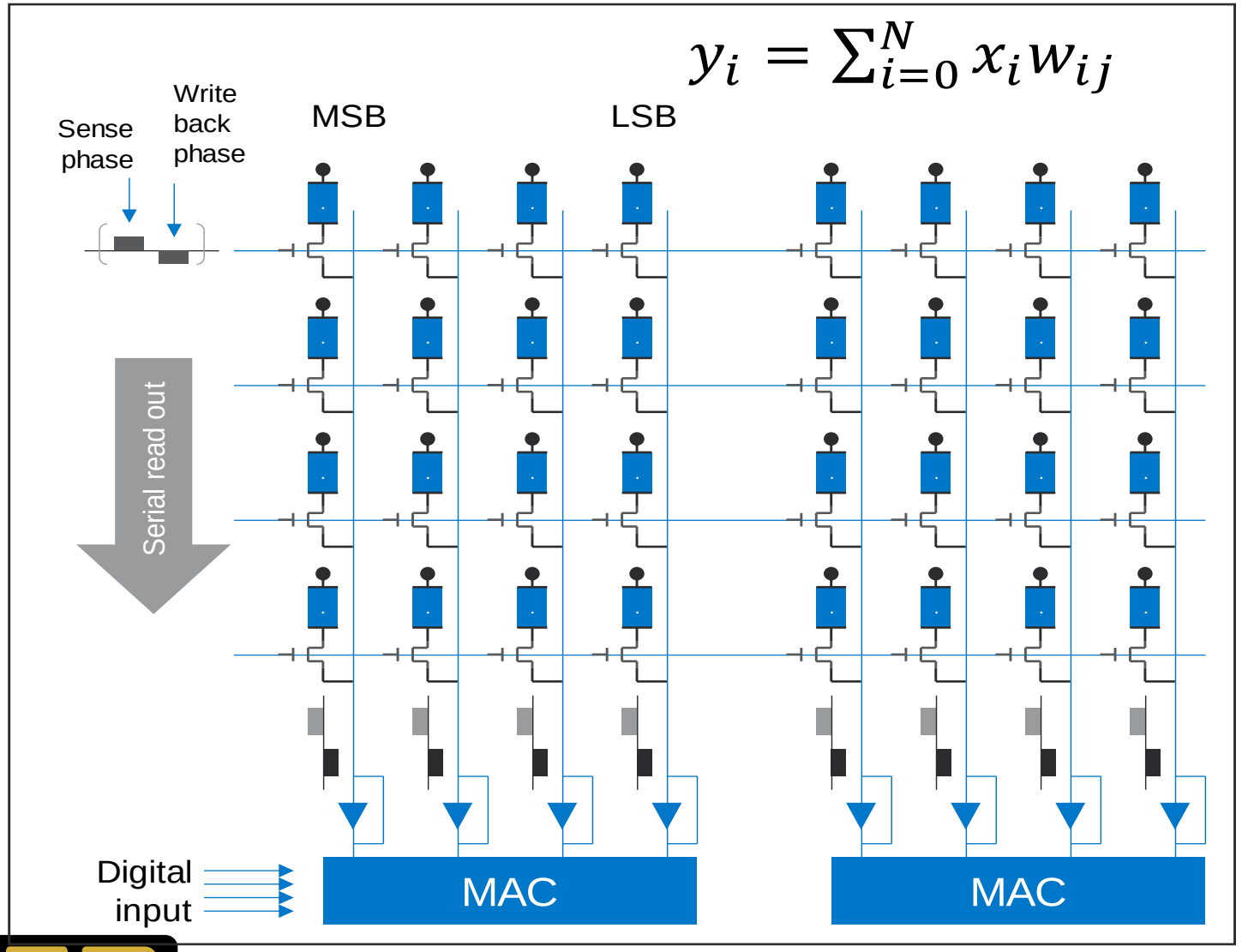


Consider putting some compute as part of the memory core

- Math and/or logical functions operating on an entire page
- Further increase in on memory die compute parallelism for specific tasks
- Operations would need to be vectors
- Favorable kernels would have high Bytes to Ops ratio

Merger of Compute and Memory for AI Accelerators

Possibly enabled by the storage Physics of some Emerging Memory Technologies

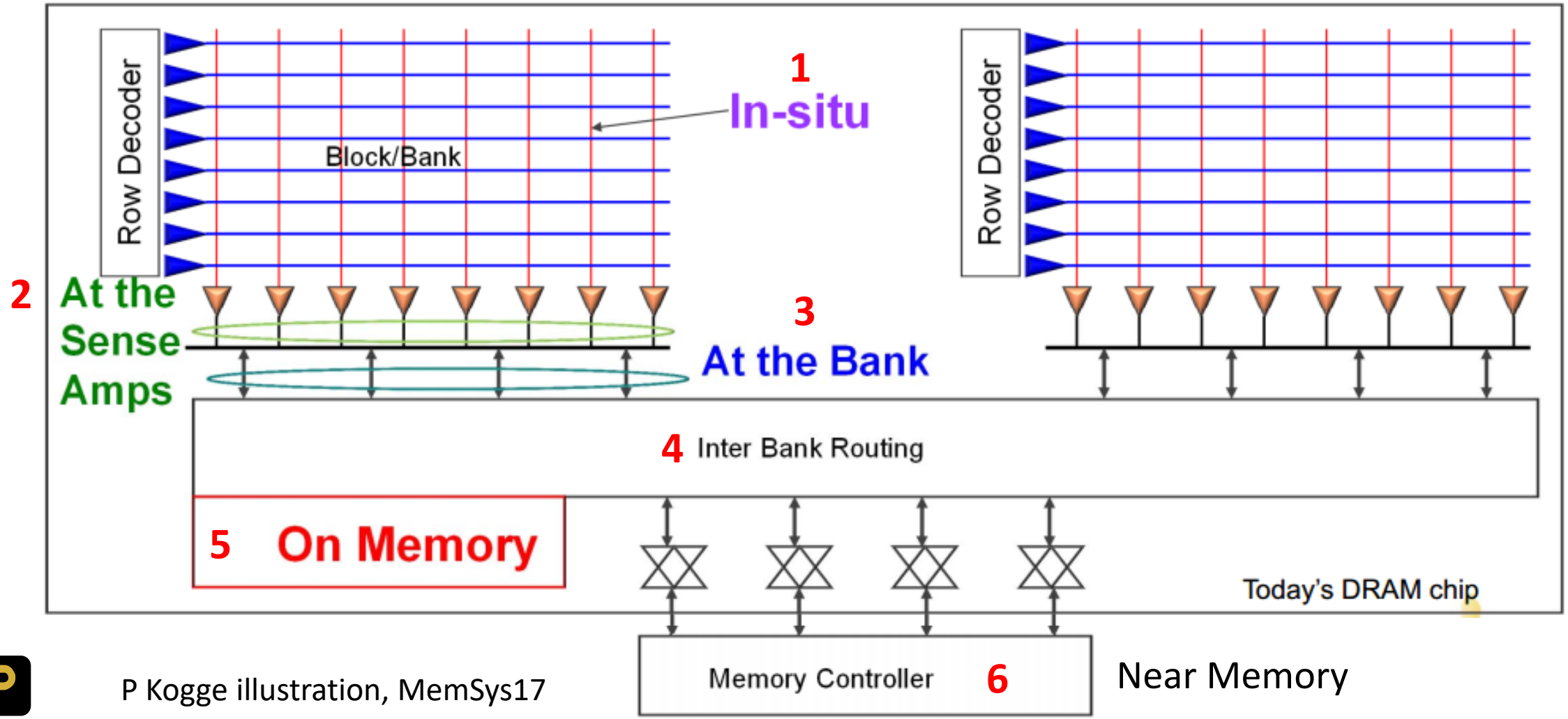


Exploit the unique physics of "emerging memory" technologies for in memory neural fabrics

- Summing (threshold) and sigmoid (triggering) behavior
- Analog "weight" storage to some degree
- Many recent papers based on resistive, magnetic, and floating gate technologies, with digital and analog processing techniques

Breaking the Energy Wall

- If memory-bound, can consider moving compute to the memory-side of the pinch-point
- 6 potential solution categories

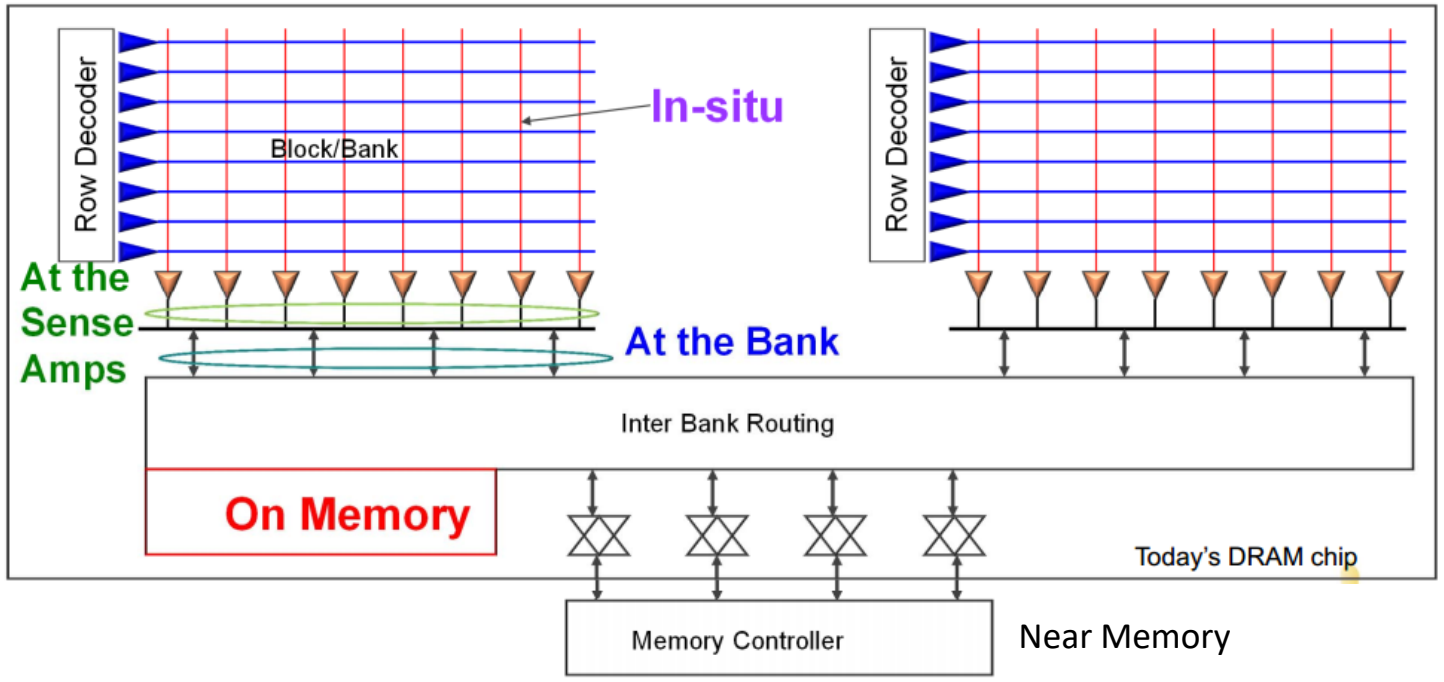


P Kogge illustration, MemSys17

Near Memory

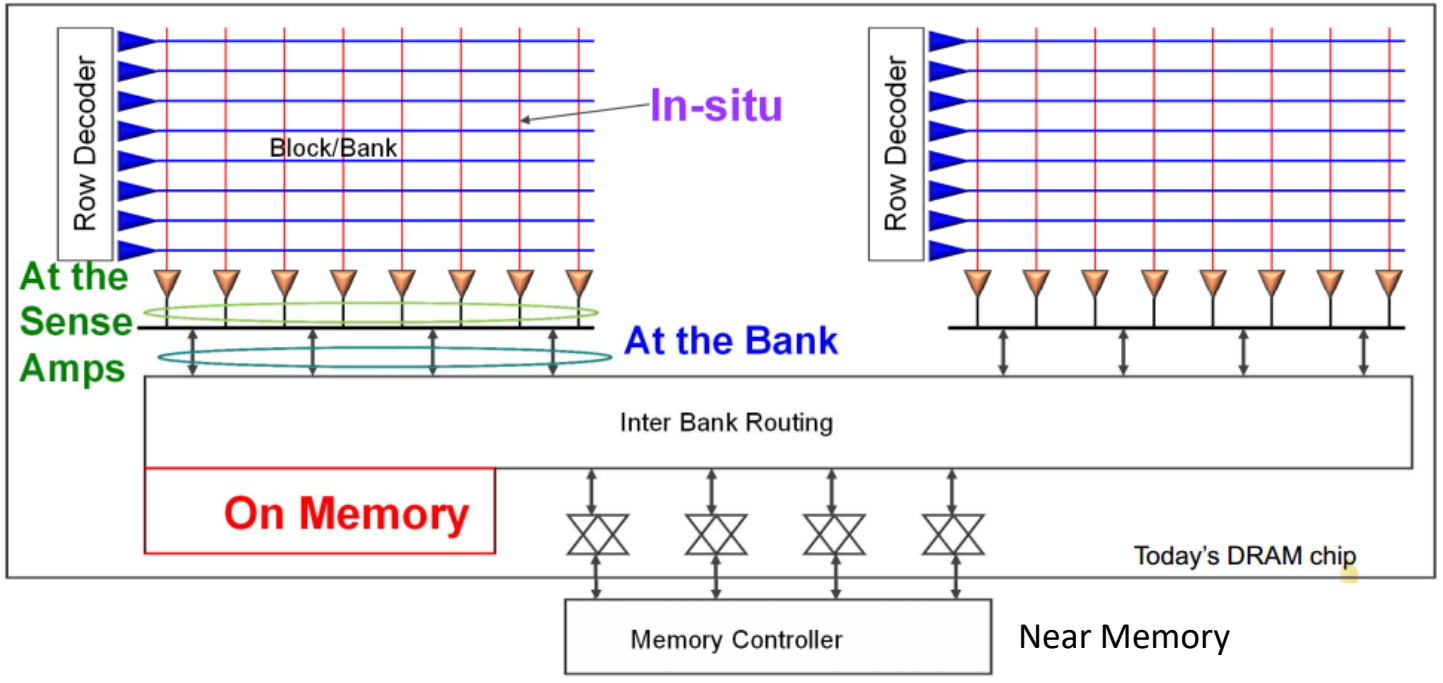
Is in-situ the Best Approach?

- Operations in the memory array itself
- E.g. enable more than a read or write in column by activating more than one row
  - Can easily construct simple Boolean operations this way
  - A big problem with this approach is the need to test a vast number of activation combinations, whereas currently there are no combinations, just single row activations
- Another problem is energy
  - Each activation adds to the energy requirement
  - Cannot keep arbitrarily increasing it without impacting power delivery and thermal management
- CAM is an example of in-situ
  - Also neuromorphic compute
  - And several recent papers at ISCA, VLSI, ISSCC, MemSys



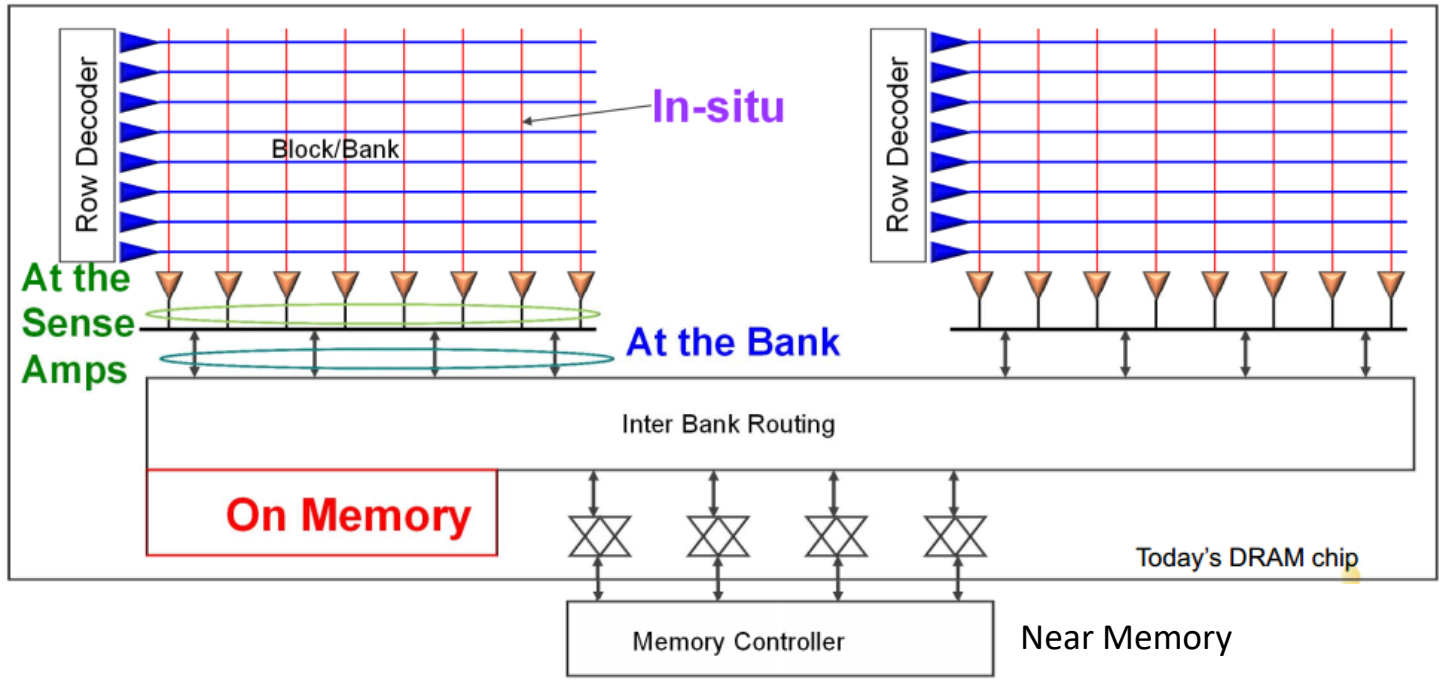
At the Sense Amps?

- At the point after where the digit lines become true Boolean signals
- Can be quite wide, and must use it at full width for highest efficiency
  - Amortize the control overhead energy
- Easy approach is 1b ops, e.g. 1ki vector of 1b ops
- With additional complexity, can make ops >1b
- More complex: perform bit-serial successive operations
- Also simple search functions
- More practical than in-situ
  - Eliminates the test permutations problem of the multiple rows activated in parallel
- Another example:
  Cray 2 Terasys SRAM chip
- Must be pitch-matched to sense amps



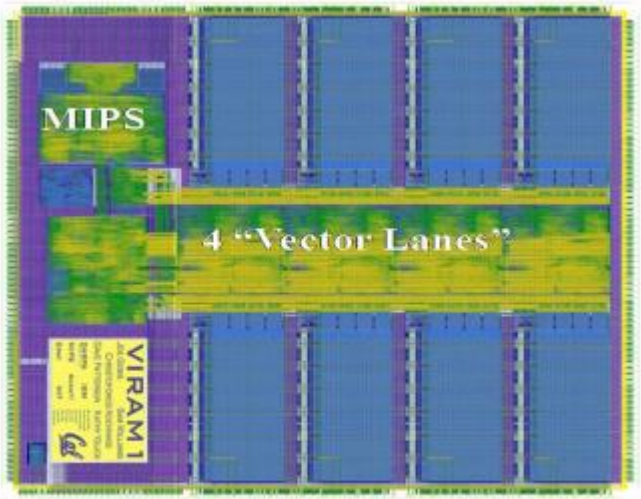
At the Bank? Or in the Inter-Bank Routing?

- After the column multiplexors
- There is less bandwidth, fewer bits, more space to add higher complexity circuitry, greater potential utilization of the added silicon
- Can range from ALUs to entire cores
- Examples
  - Execube 8-core (each core access to 1/8th memory)
  - Micron Yukon test vehicle with 256 8b ALUs in eDRAM technology, early 2000's was in the inter-bank routing area
  - Associative computing from Mikamonu, acquired by GSI
    - Can be implemented at bank or at inter-bank routing
  - Micron's HMC compute was at inter-bank routing, as was IBM's offshoot of it
    - Bottom of vault in 3D structure
    - Could also classify as On-Memory
  - UpMem – 1 Data Processor Unit at each bank, 14-stage pipe, etc.
- Complication: need to transpose row and column bits; data layout is sensitive issue

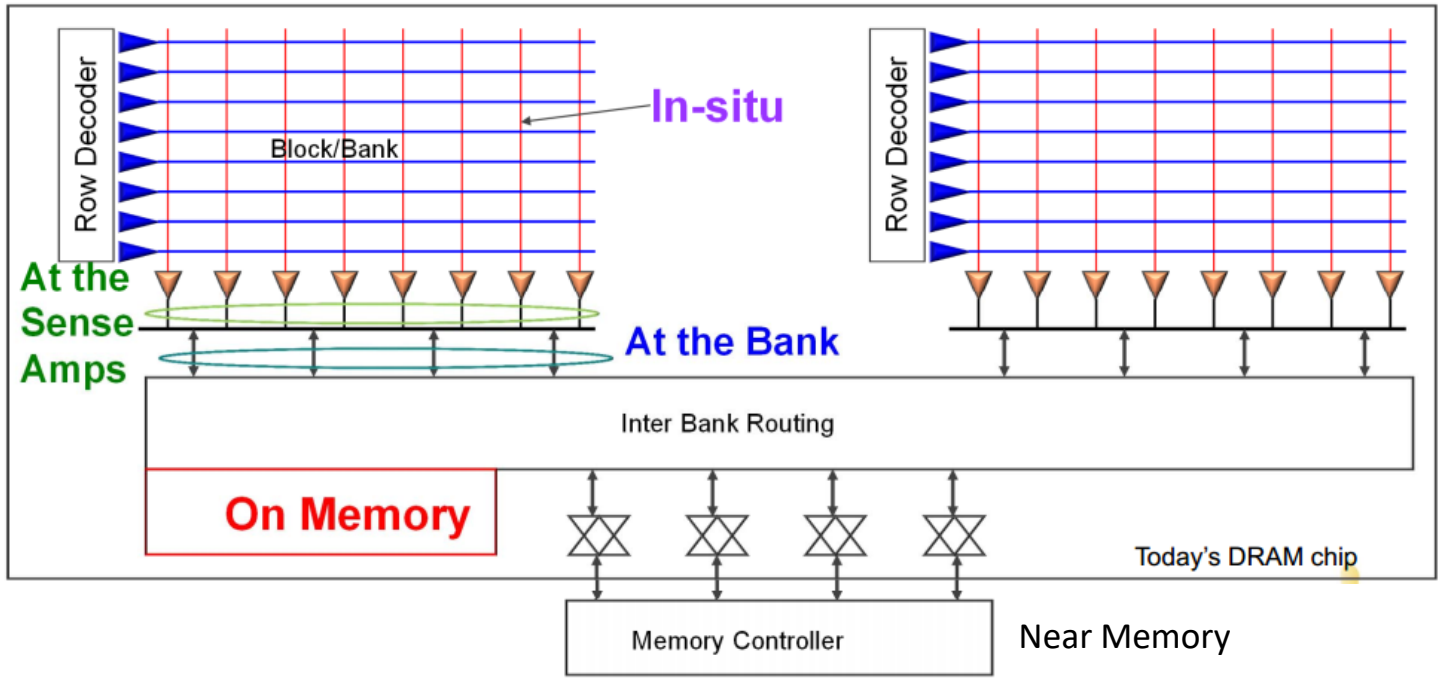


On Memory?

- On the same die as memory – after the inter-bank routing resources
- Typically a complete processor(s)
- Sees all the memory on die but nothing off-chip
- Chip interface may be memory-like
- E.g. Berkeley Vector IRAM (MIPS off to the side + Vector processors down the center spine); Mitsubishi M32R/D (processor core in the middle surrounded by 4 DRAMs)
- Both were eDRAM proposals
- Poor idea unless transistors and interconnect are improved

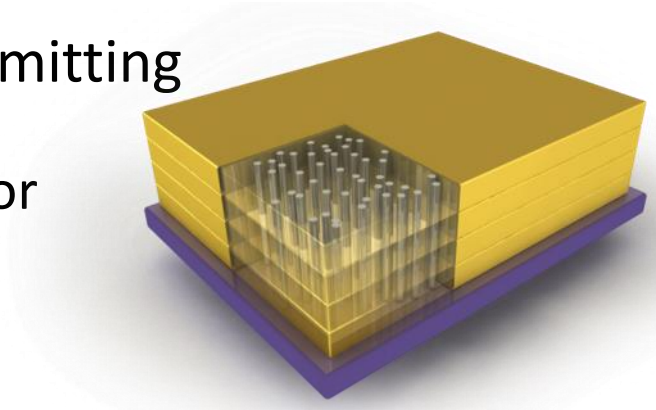


Berkeley VIRAM



Near Memory?

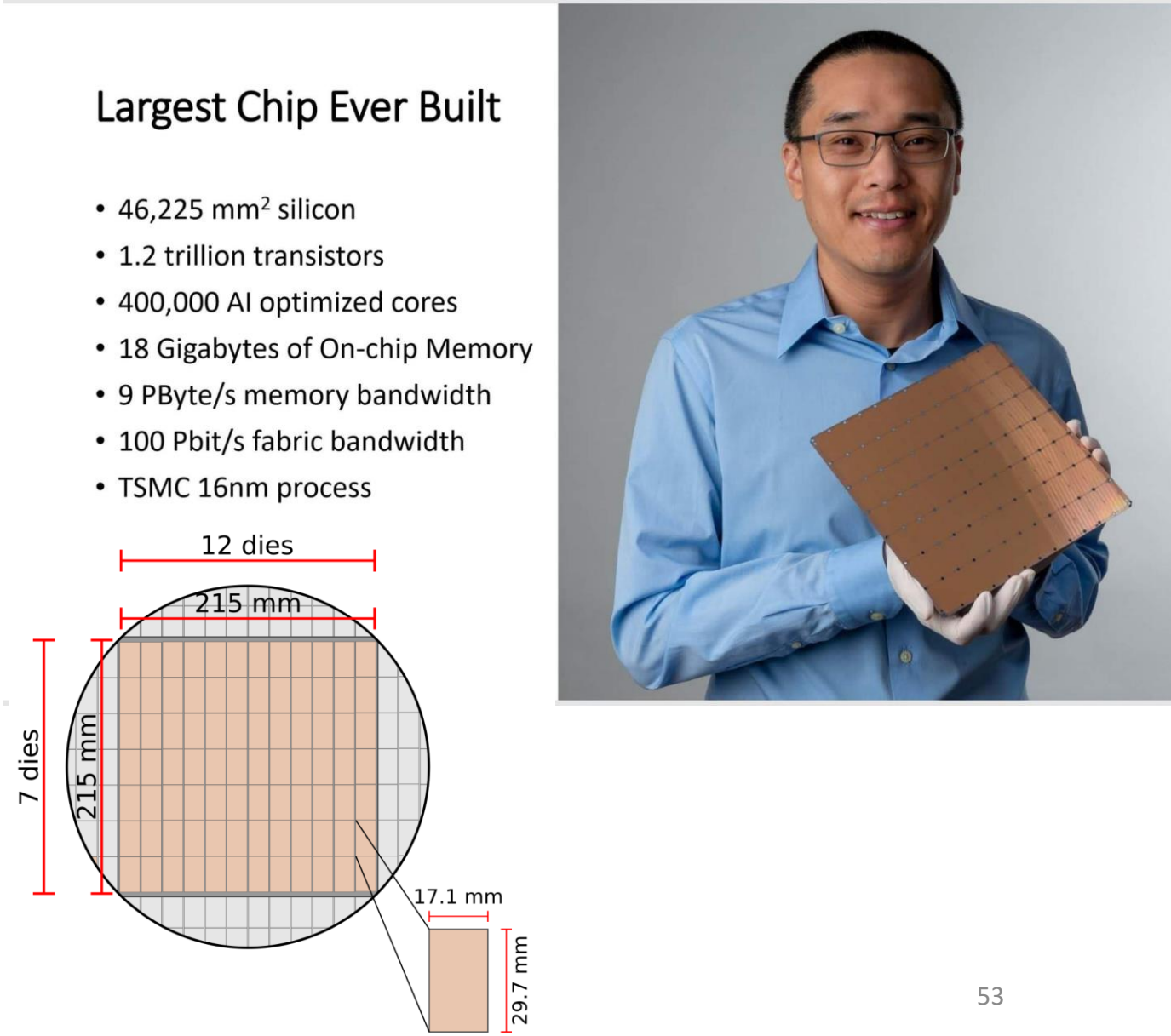
- Inside or in the vicinity of the memory controller
- Allows access to all memory unless some optimization deems that unwise
- Does the near-memory processor see logical or physical memory map?
  - A choice that can add tremendous complication
- In the case of a construction such as HMC where bottom die is foundry CMOS, processor could be as capable as any, power permitting
  - Production HMC did incorporate atomic Boolean, integer, and scatter/gather functions but did not yet incorporate a full processor
- In the case of an HBM-like construction where bottom die is in DRAM node, processor is much more limited in performance.
- Lucata (formerly known as Emu, Peter Kogge et al) has a brilliant approach – work moving via migratory threads



LUCATA

And the memory company's nightmare…processing without external memory. Memory and logic integrated in logic process only

- Cerebras example, eliminates the need for external memory, problems are mapped into the network of chips
  - 84 die stitched together on wafer, including IO circuitry between them
  - Works fine until the problem exceeds the chip capability
  - Begs for 3D solution instead of planar extension of chips
  - Lots of weaknesses – off-chip IO, cost, etc.
- Neuromorphic example is the opposite: memory and thin logic integrated into memory process



Common Challenges with Memory-Compute

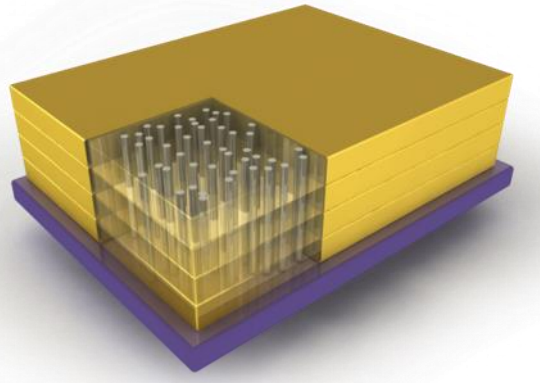
- A common problem with all of these in/at/near memory compute approaches is inherited from the DRAM system situation
  - E.g. if incorporated on DRAMs of a DIMM, data is interleaved across all DRAMs of a rank – no single DRAM sees the whole data variable
  - Optimizing data layout is preeminent issue
- Another common problem is disruption of host processes – conflict with local computations
  - Is it still a memory, or an accelerator with private memory?
- Ecosystem support is the elephant in the room – enabling simple access to the programmer

But, Opportunities Too

- Memory manufacturers have focused memory technology on inexpensive bits
- As the pressure to perform compute in memory mounts, performance can be economically addressed (increased)
  - Vastly improved interconnect and transistor performance is possible
  - This upsets all the analysis done to date – research anew

Final Thoughts

- HBM vs HMC concepts



- Must propagate and advantageously utilize abstraction
  - Enable selective memory-processing
- Must properly seize the advantages of 3D in both logic and memory
- Are chiplets a stop-gap measure only for planar scaling or does it have a future in 3D?