technische universität
dortmund

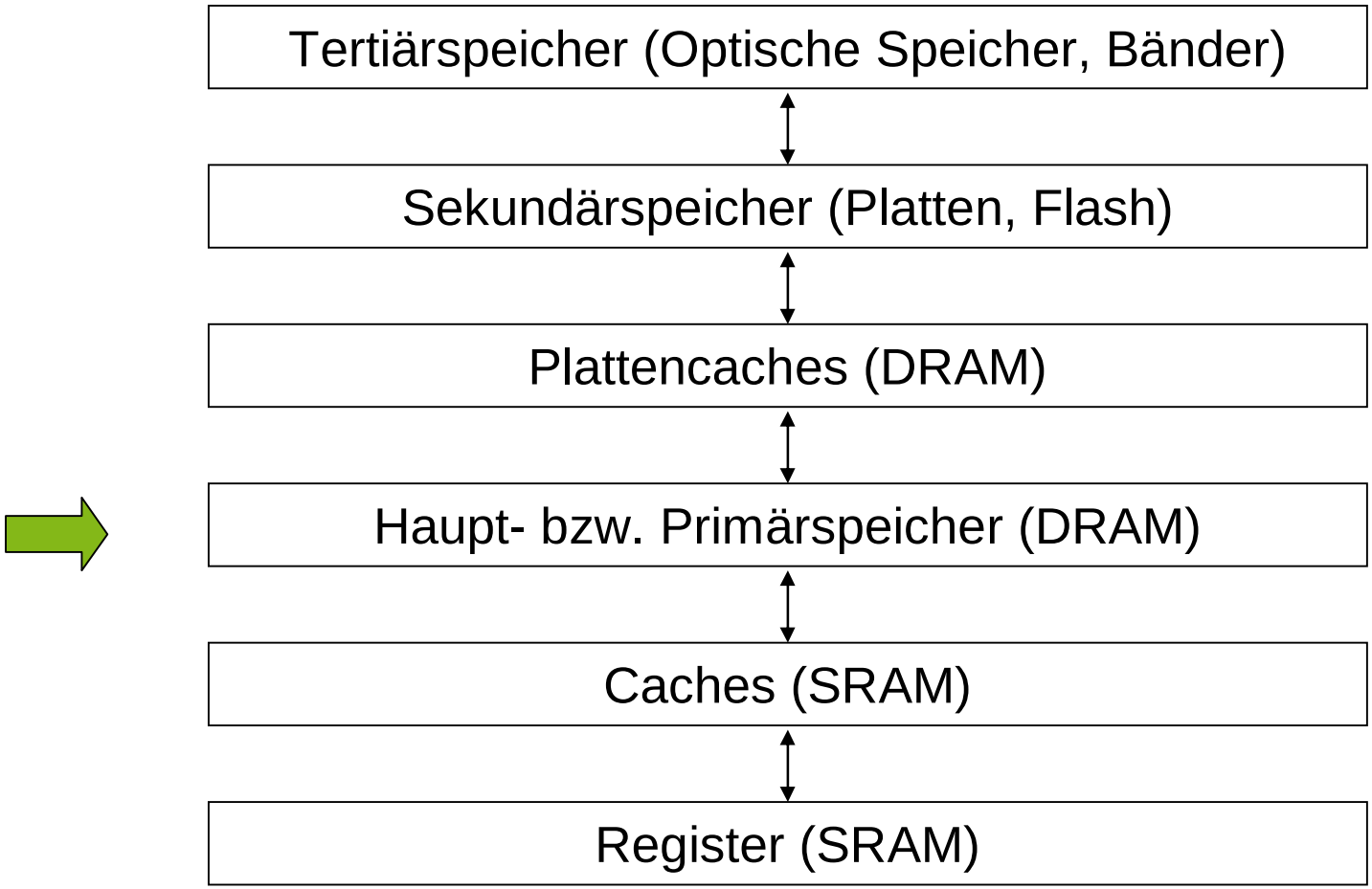
fakultät für informatik
informatik 12

Die Speicherhierarchie

Peter Marwedel
Informatik 12
TU Dortmund

2012/05/24

Mögliche Stufen der Speicherhierarchie
und derzeit eingesetzte Technologien



Hautspeicherorganisation

Hauptspeicher ist weitere Ebene der Speicherhierarchie

Für Leistung wichtig: Latenz und Bandbreite

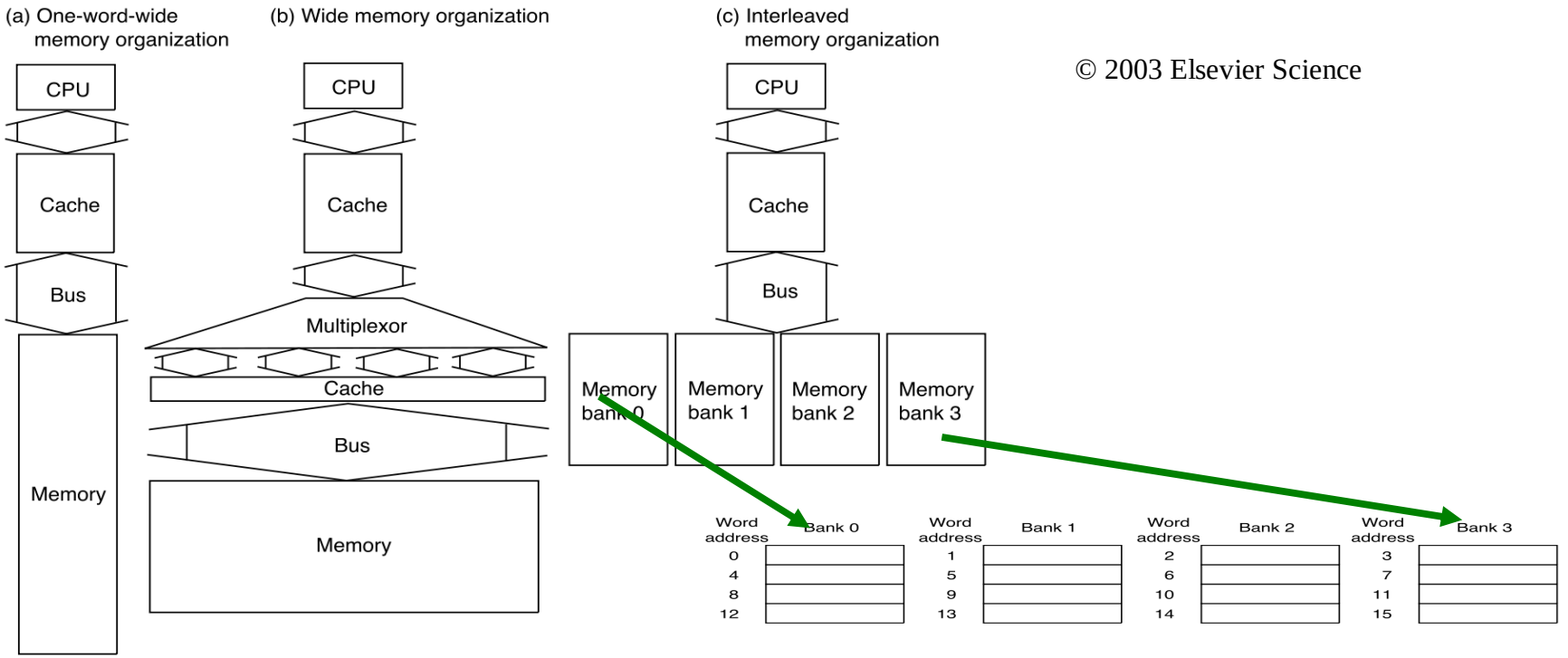
- Latenz relevant für Kosten eines Fehlzugriffs auf *Cache*
- Bandbreite wichtig in Kombination mit ...
  - großem L2-Cache ...
  - der große Cache-Blöcke verwendet.

☞ Große Transfereinheiten zwischen *Cache* und Hauptspeicher, kein wortweiser Zugriff!

Verringerung der Latenz aufwendig

Verbesserung der Bandbreite durch geeignete Organisation des Speichers (relativ leicht) möglich

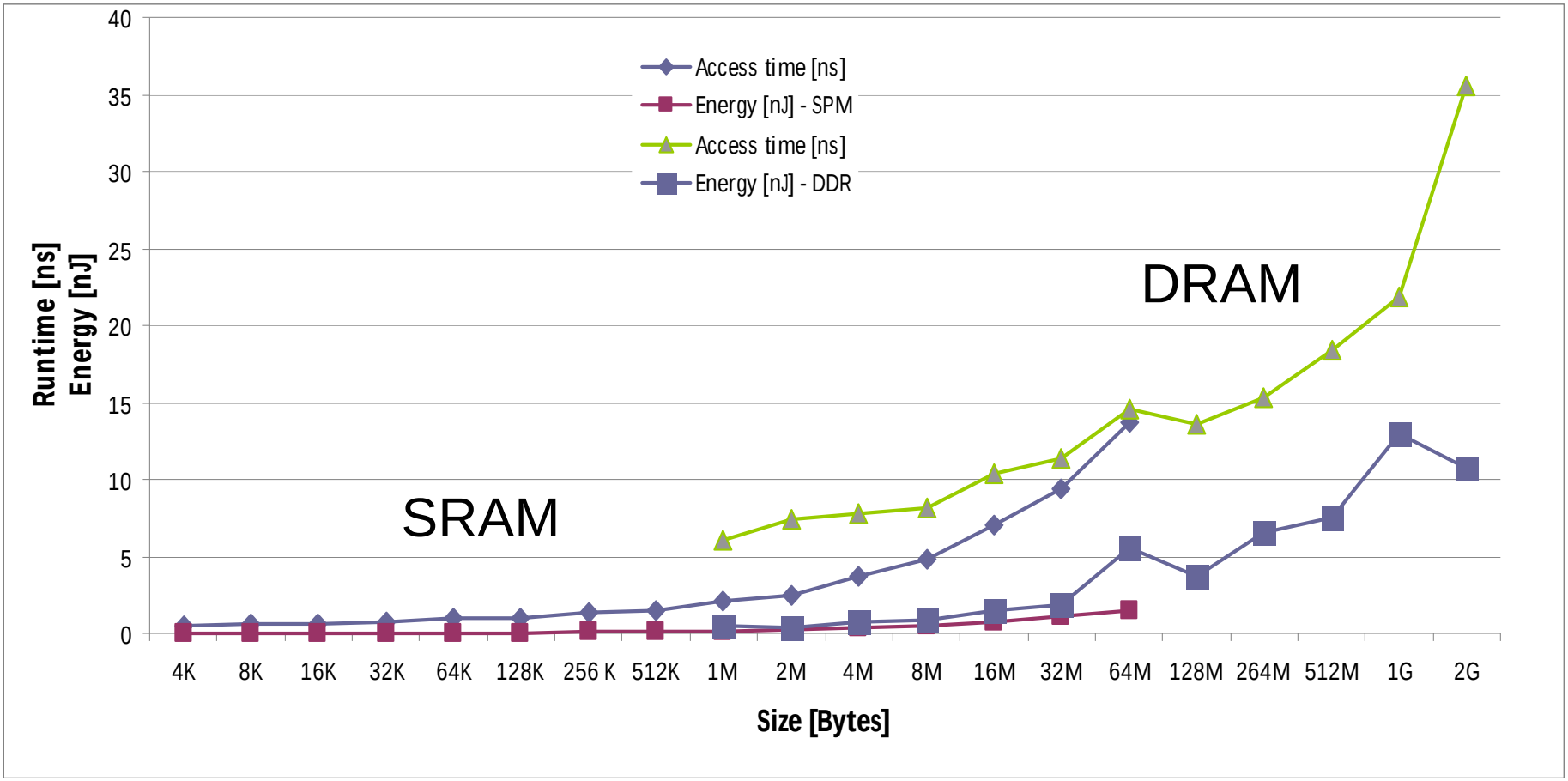
Hautspeicherorganisation (2)



- „Breiter" Hauptspeicher: Höhere Bandbreite durch Parallele Zugriffe
- „Verschränkte" Organisation (*interleaved*): logisch wie „breiter" Speicher, Zugriff auf Bänke zeitlich sequentiell

Energy consumption of memories

Scratchpad (SRAM) vs. DRAM (DDR2)

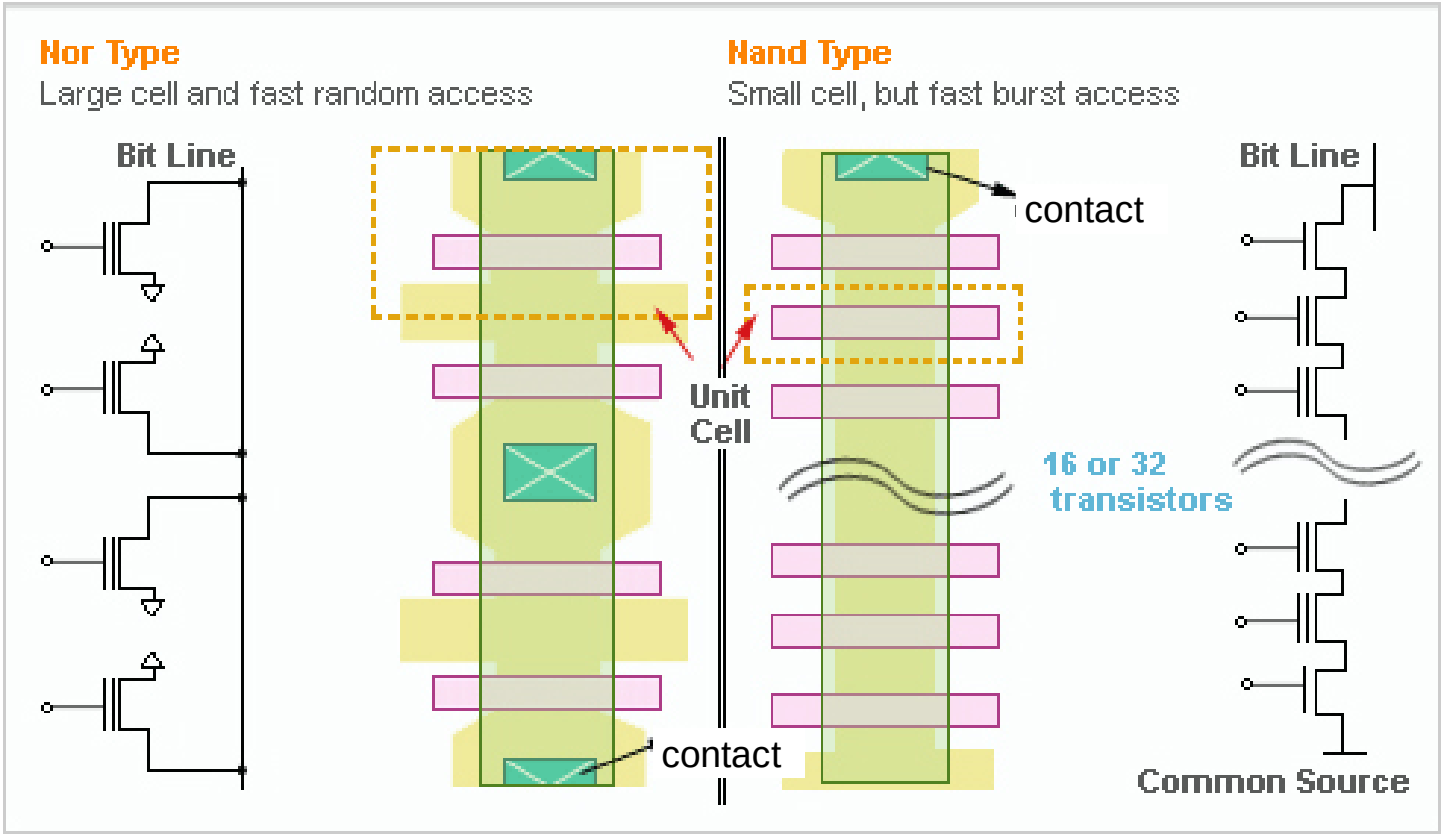


CACTI, 64 bit read; banks: 16;
65 nm for SRAM, 80 nm for DRAM

Source: Olivera Jovanovic, TU Dortmund, 2012

NOR- und NAND-Flash

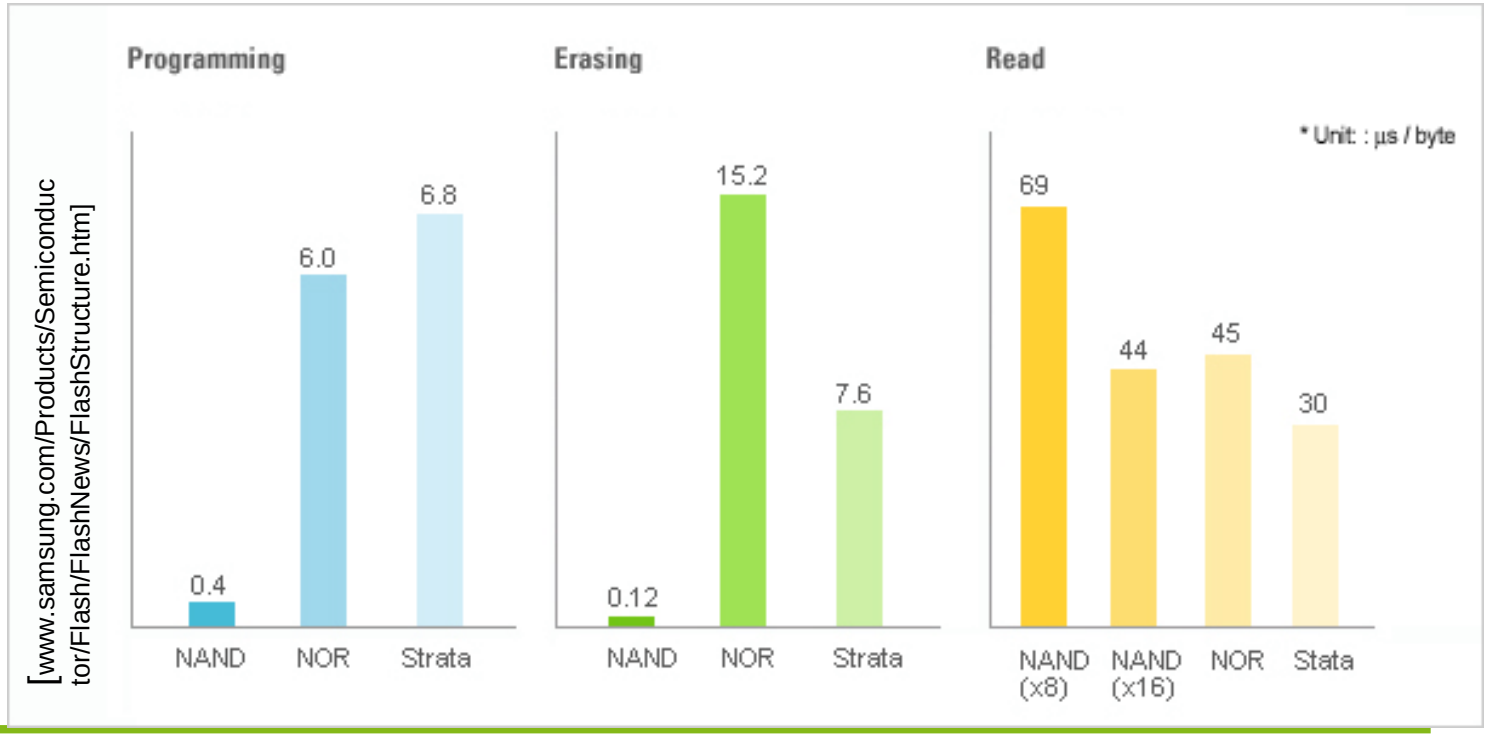
NOR: 1 Transistor zwischen Bitleitung und Masse
NAND: >1 Transistor zwischen Bitleitung und Masse



was at [www.samsung.com/Products/ Semicon-ductor/ Flash/FlashNews/ FlashStructure. htm] (2007)

Eigen-schaften von NOR- und NAND-Flash-Speichern

Type/Eigenschaft	NOR	NAND
Wahlfreier Zugriff	Ja ☺	Nein ☹
Block löschen	Langsam ☹	Schnell ☺
Zellgröße	Groß ☹	Klein ☺
Zuverlässigkeit	Größer ☺	Kleiner ☹
Direktes Ausführen	Yes ☺	No ☹
Anwendungen	Codespeicherung, *boot flash, set top box*	Datenspeicher, USB *Sticks*, Speicherkarten



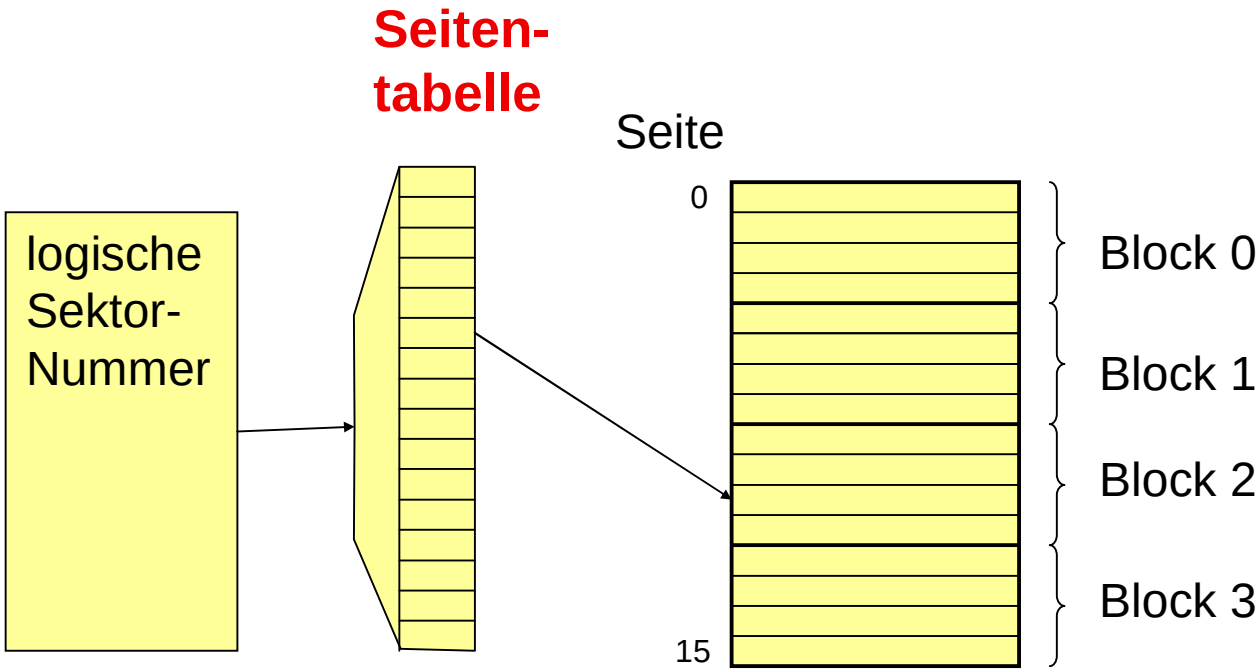
Charakteristische Eigenschaften von NAND Flash Speicher

Speicher aufgeteilt in Blöcke (typ. 16-256 KB),
Blöcke unterteilt in Seite (typ. 0.5-5 KB).
Schreib-/Lesevorgänge jeweils auf Seiten

	1 Bit/Zelle (SLC)	>1 Bit/Zelle (MLC)
Lesen (Seite)	25 µs	≫ 25 µs
Schreiben (Seite)	300 µs	≫ 300 µs
Löschen (Block)	2 ms	1.5 ms

J. Lee, S. Kim, H. Kwin, C. Hyun, S, Ahn, J. Choi, D. Lee, S.Noh: Block Recycling Schemes and Their Cost-based Optimization in NAND Flash Memory Based Storage System, EMSOFT'07, Sept. 2007

Seiten-/bzw. Sektorabbildung mit *Flash transaction layer* (FTL)

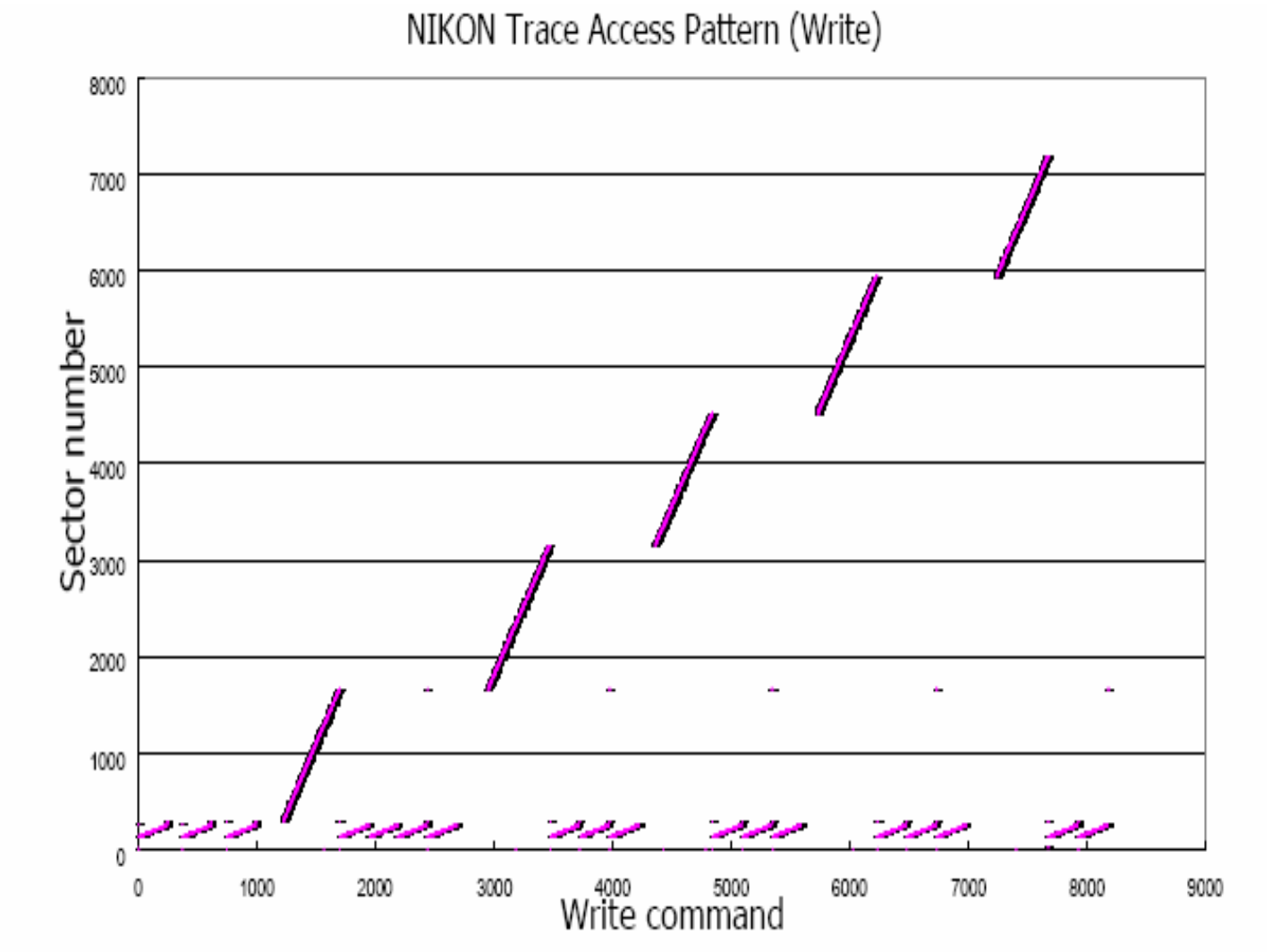


Invertierte Seitentabelle im Flashspeicher gespeichert (Extra Bits); “Normale Seitentabelle” während der Initialisierung erzeugt; Seitentabelle kann sehr groß werden; Wird in kleinen NOR Flash-Speichern benutzt.

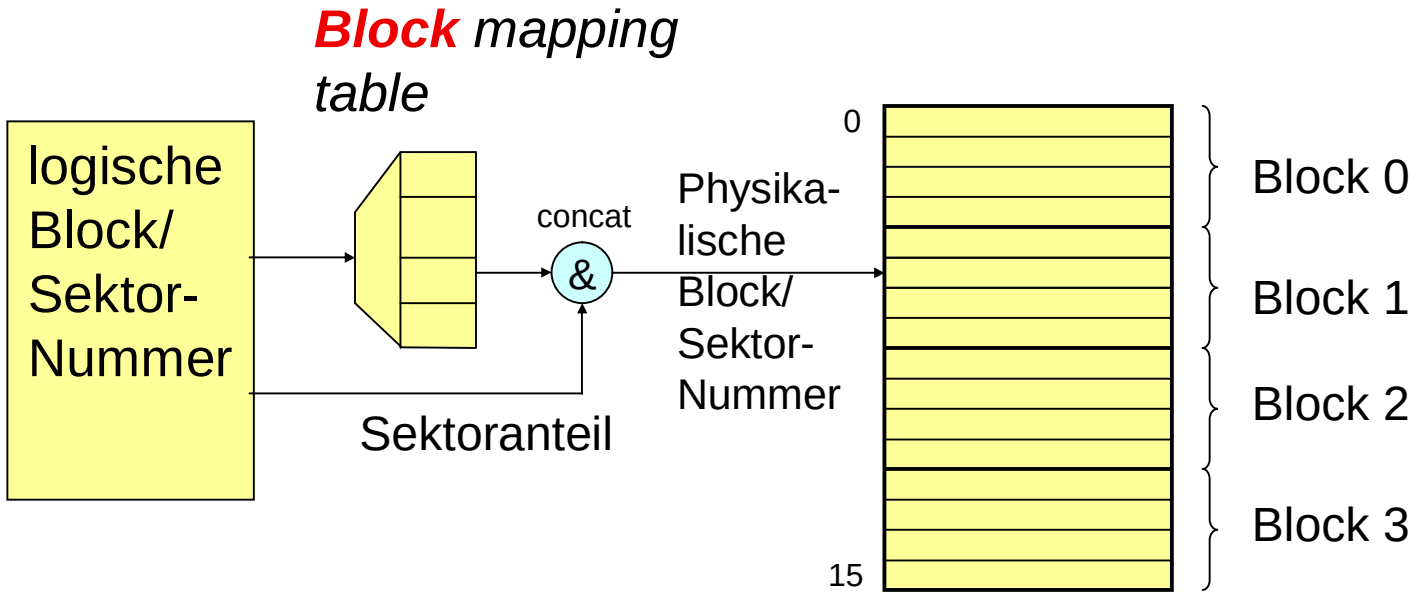
Sektor ≈ Seite
+ Extra Bits

Ausnutzung von Regularität

Häufig lange Sequenzen von sequentiellen Schreib-vorgängen



Block mapping flash transaction layer (FTL)

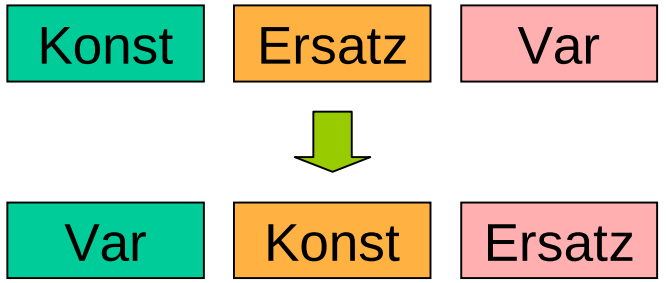


- Abbildungstabellen kleiner als bei Seiten-basierten FTLs
- ☞ In großen NAND Flash-Speichern benutzt
- ☞ Einfache Realisierung,
- Wiederholtes Schreiben erfordert Kopieren auf einen neuen Block
- Schlechte *Performance* bei wiederholtem und zufälligem Schreiben
- Hybride Verfahren

Ausgleich der Abnutzung (*wear levelling*)

Beispiel (Lofgren et al., 2000, 2003):

- Jede *erase unit* (Block) besitzt Löschzähler
- 1 Block wird als Ersatz vorgehalten
- Wenn ein häufig genutzter Block frei wird, wird der Zähler gegen den des am wenigsten benutzten Blocks verglichen. Wenn der Unterschied groß ist:
  - Inhalt wenig genutzten Blocks (≈ Konstanten) → Ersatz
  - Inhalt häufig genutzten Blocks → am wenigsten genutzter Block
  - Häufig genutzter Block wird zum Ersatzblock



Source: Gal, Toledo, *ACM Computing Surveys*, June 2005

Flash memory as main memory

Approach published (Wu, Zwaenepoel, 1994):

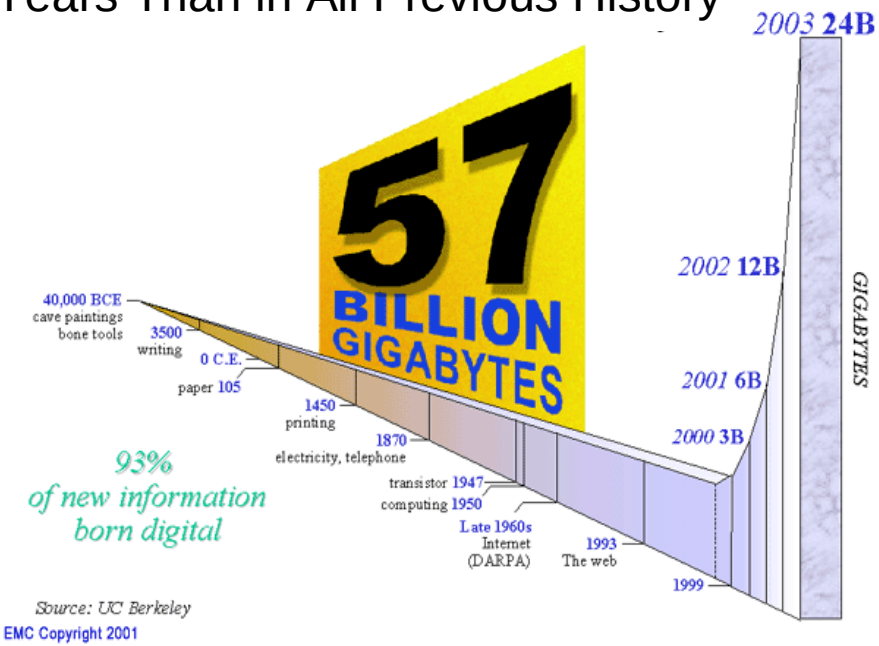
- Uses MMU
- RAM + Flash mapped to memory map
- Reads from Flash read single words from Flash
- Writes copy block of data into RAM,
  all updates done in RAM
- If the RAM is full, a block is copied back to Flash
- Crucial issue: Speed of writes.
  Proposal based on wide bus between Flash and RAM, so that writes are sufficiently fast
  ☞ Larger erase units, increased wear-out feasible.

M. Wu, W. Zwaenepoel: eNVy: A nonvolatile, main memory storage system. In *Proceedings of the 6th International Conference on Architectural Support for Programming Languages and Operating Systems*. 1994, p. 86–97.

Memory hierarchies beyond main memory

- Massive datasets are being collected everywhere
- Storage management software is billion-$ industry

More New Information Over Next 2 Years Than in All Previous History



Examples (already in 2002):

Phone: AT&T 20TB phone call database, wireless tracking

Consumer: WalMart 70TB database, buying patterns

WEB: Web crawl of 200M pages and 2000M links, Akamai stores 7 billion clicks per day

Geography: NASA satellites generate 1.2TB per day

[© Larse Arge, I/O-Algorithms, http://www.daimi.au.dk/~large/ioS07/]

Vergleich *Harddisc/Flash*-Speicher (2011)

	Flash	HDD
Zugriffszeit (random) [ms]	~0.1	5-10
Kosten [$/GB]	1,2-2 (Fixanteil gering)	0,05-0,1, Fixanteil !
Kapazität [GB]	Tyo. < 120	1000-3000
Leistungsaufnahme [W]	Ca. 1/3-1/2 der HDD-Werte	Typ. 12-18, laptops: ~2
Defragmentierung	unwichtig	Zu beachten
Zeitverhalten	Relativ deterministisch	Von Kopfbewegung abhängig
Anzahl der Schreibvorgänge	begrenzt	unbegrenzt
Verschlüsselung	Überschreiben unver-schlüsselter Info schwierig	Überschreiben einfach
Mechanische Empfindlichk.	Gering	Stoßempfindlich
Wiederbenutzung von Blöcken	Erfordert Extra-Löschen	Überschreiben

Flash-spezifische Dateisysteme

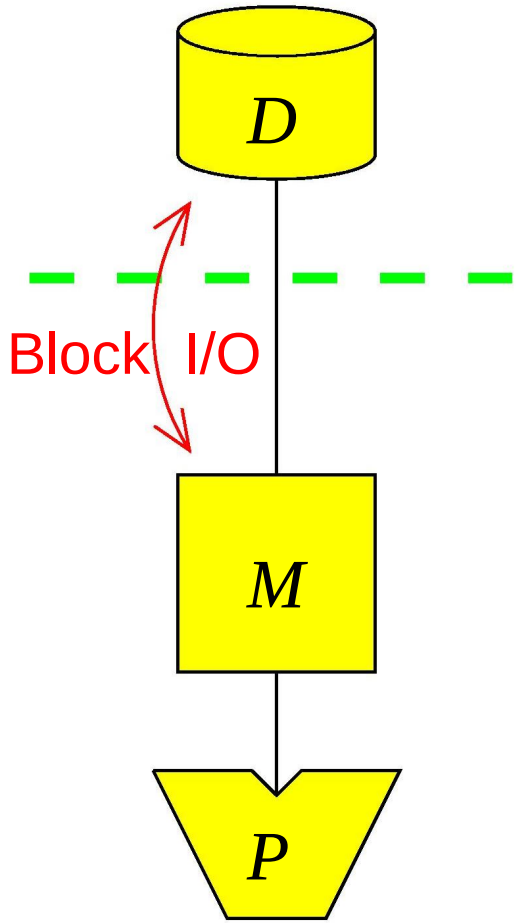
- Zwei Ebenen können ineffizient sein:
  - FTL bildet Magnetplatte nach
  - Standard-Dateisystem basiert auf Magnetplatten

  Beispiel: Gelöschte Sektoren nicht markiert
  ☞ nicht wieder verwendet
- Log-strukturierte Dateisysteme fügen nur neue Informationen zu
  - Für Magnetplatten
    - Schnelle Schreibvorgänge
    - Langsames Lesen (Kopfbewegungen für verteilte Daten)
  - Ideal für Flash-basiertes Dateisystem:
    - Schreibvorgänge in leere Sektoren
    - Lesen nicht langsam, da keine Köpfe bewegt werden

☞ Spezifische log-basierte *Flash*-Dateisysteme
- JFFS2 (NOR)
- YAFFS (NAND)

Source: Gal, Toledo, *ACM Computing Surveys*, June 2005

External Memory Model



N= # of items in the problem instance
B = # of items per disk block
M = # of items that fit in main memory

T = # of items in output

I/O: Move block between memory and disk

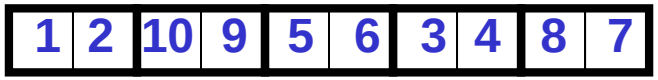
We assume (for convenience) that $M > B^2$

[© Larse Arge, I/O-Algorithms, http://www.daimi.au.dk/~large/ioS07/]

Scalability Problems: Block Access Matters

- Example: Reading an array from disk
  - Array size N = 10 elements
  - Disk block size B = 2 elements
  - Main memory size M = 4 elements (2 blocks)

Algorithm 1: N=10 I/Os



Algorithm 2: N/B=5 I/Os

- Difference between N and N/B large since block size is large
  - Example: $N = 256 \times 10^6$, $B = 8000$, *1ms* disk access time

    $\Rightarrow$ N I/Os take 256×10^3 *sec* = *4266 min* = *71 hr*

    $\Rightarrow$ N/B I/Os take *256/8 sec* = *32 sec*

[© Larse Arge, I/O-Algorithms, http://www.daimi.au.dk/~large/ioS07/]

Re-writing algorithms for memory hierarchies

Analysis of algorithm complexity mostly using the *RAM* (random access machine; const. mem. acc. times) model outdated

☞ take memory hierarchies explicitly into account.

Example:

- Usually, divide-&-conquer algorithms are good.
- "Cache"-oblivious algorithms (are good for any size of the faster memory and any block size). Assuming
  - Optimal replacement (Belady's algorithm)
  - 2 Memory levels considered (there can be more)
  - Full associativity
  - Automatic replacement

Unlikely to be ever automatic

[Piyush Kumar: Cache Oblivious Algorithms, in: U. Meyer et al. (eds.): Algorithms for Memory Hierarchies, *Lecture Notes in Computer Science, Volume 2625*, 2003, pp. 193-212]

[Naila Rahman: Algorithms for Hardware Caches and TLB, in: U. Meyer et al. (eds.): Algorithms for Memory Hierarchies, *Lecture Notes in Computer Science, Volume 2625*, 2003, pp. 171-192]

Zusammenfassung

Betrachtung der gesamten Speicherhierarchie

- *Interface* zu Caches erfordert breite Zugänge zum Hauptspeicher
- *Flash*-Speicher erfordern Anpassung an technologische Eigenheiten
  - Erfordern in der Regel Abbildung logische→reale Blockadressen, und dementsprechend FTL/MMU
  - Nur eingeschränkt als Hauptspeicher geeignet
  - Als Sekundärspeicher am besten mit speziellem Dateisystem zu kombinieren
- Bei großen Datenmengen sind „Sekundärspeicher“ hinsichtlich der *Performance* die entscheidenden Komponenten

Das war die Pflicht, nun zur Kür!

Die weitere Entwicklung

Basis:

- ITRS
- Babak Falsafi: Dark Silicon & Its Implications on Server Chip Design, Microsoft Research, Nov. 2010
  Siehe auch *publications* unter http://parsa.epfl.ch/~falsafi/
- Hadi Esmaeilzadeh: Dark Silicon and the End of Multicore Scaling, International Symposium on Computer Architecture (ISCA ’11)

Predicted number of functions per chip

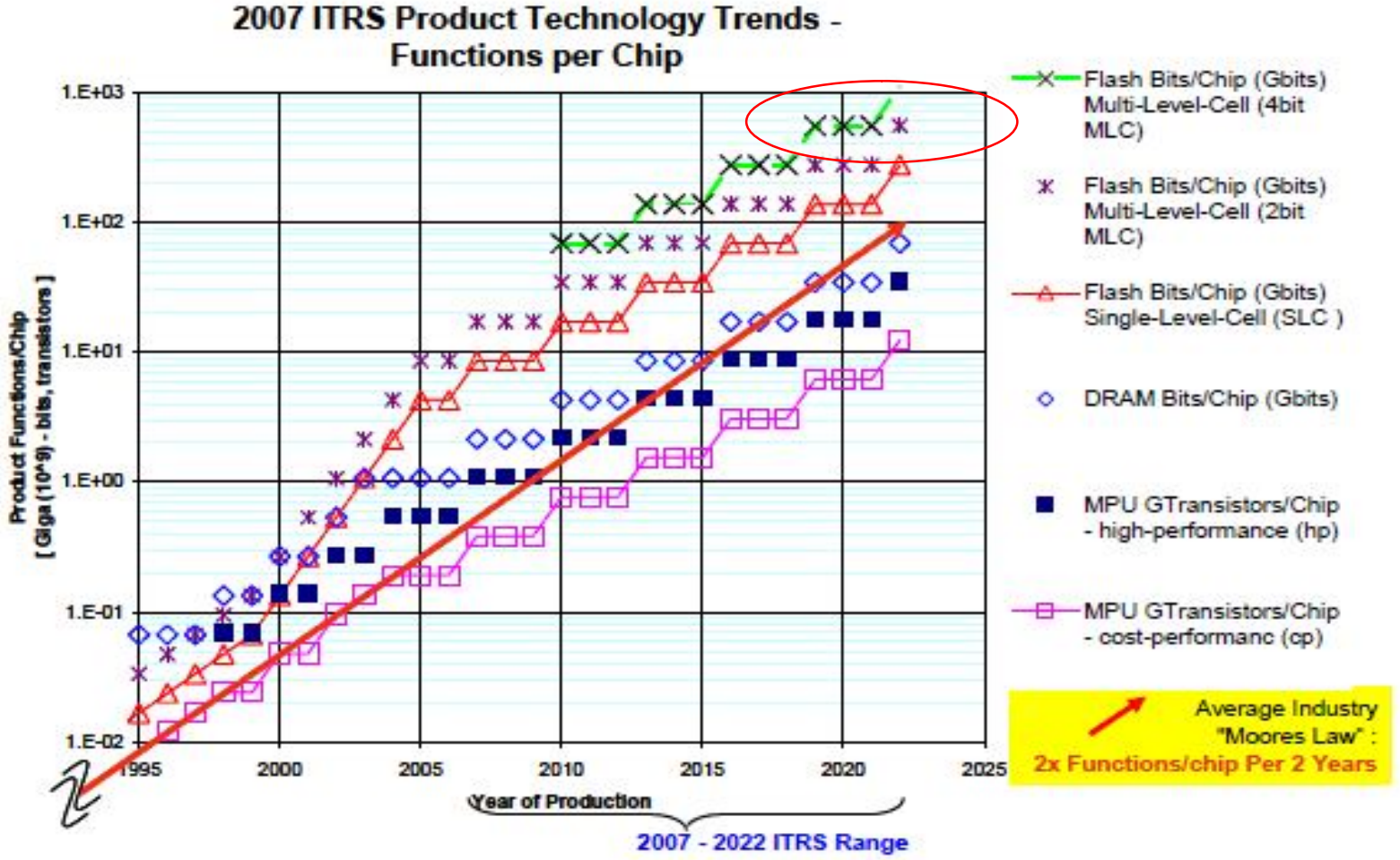
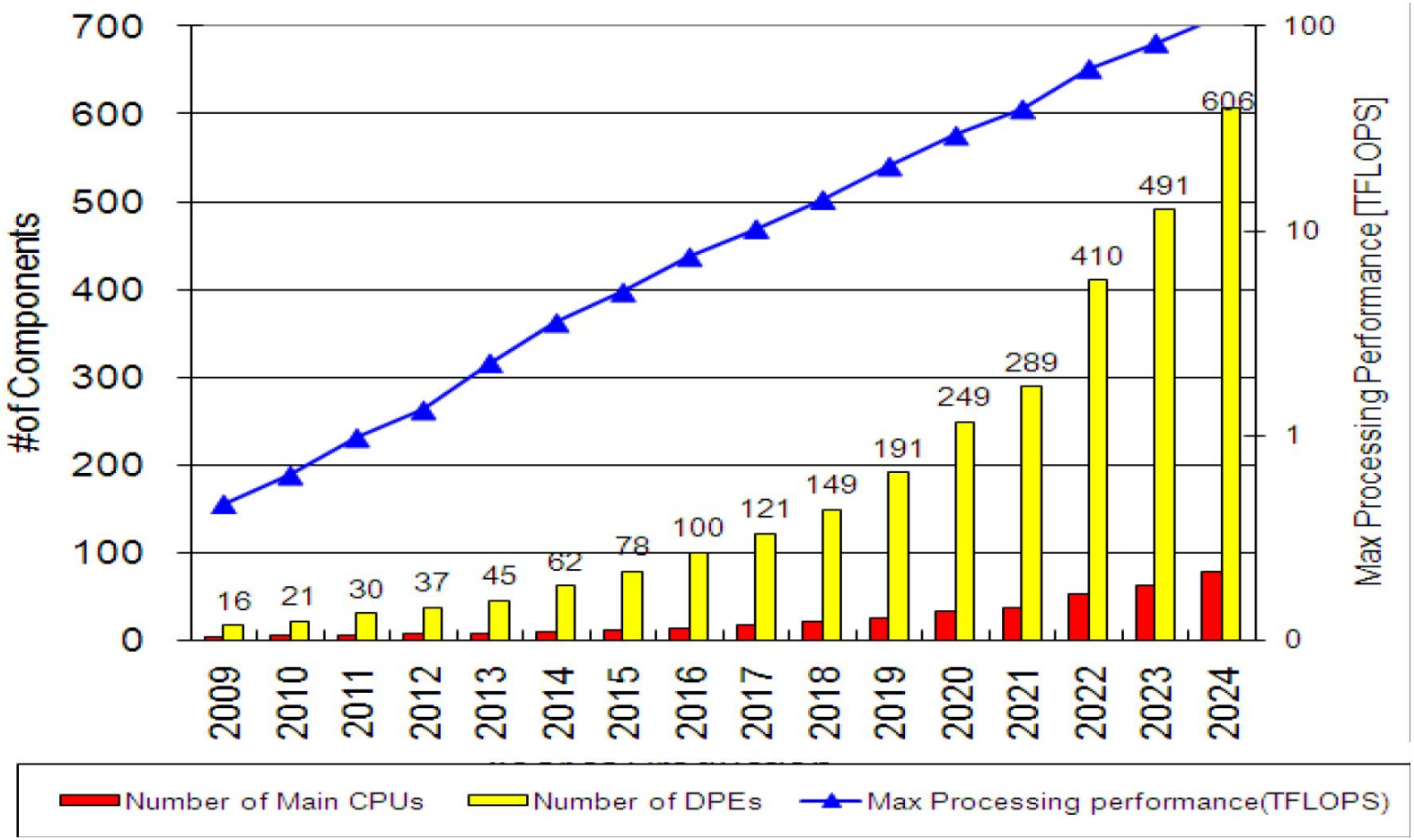


Figure ORTC2 ITRS Product Function Size Trends: MPU Logic Gate Size (4-transistor); Memory Cell Size [SRAM (6-transistor); Flash (SLC and MLC), and DRAM (transistor + capacitor)]--Updated

[ITRS Update 2008]

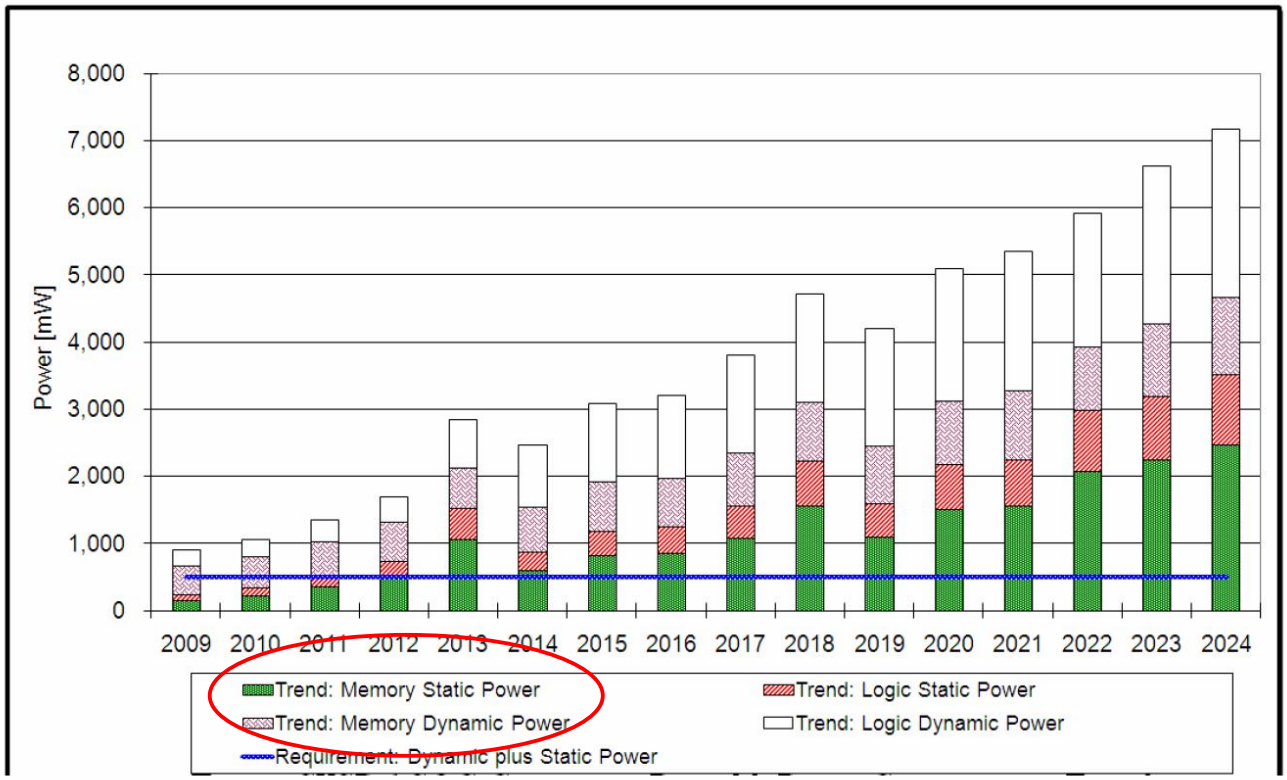
Reason for increased power consumption: trend toward higher performances

For stationary systems:



Where is the power consumed?
- Consumer portable systems -

- According to *International Technology Roadmap for Semi-conductors* (ITRS), 2010 update, [www.itrs.net]
- Based on current trends

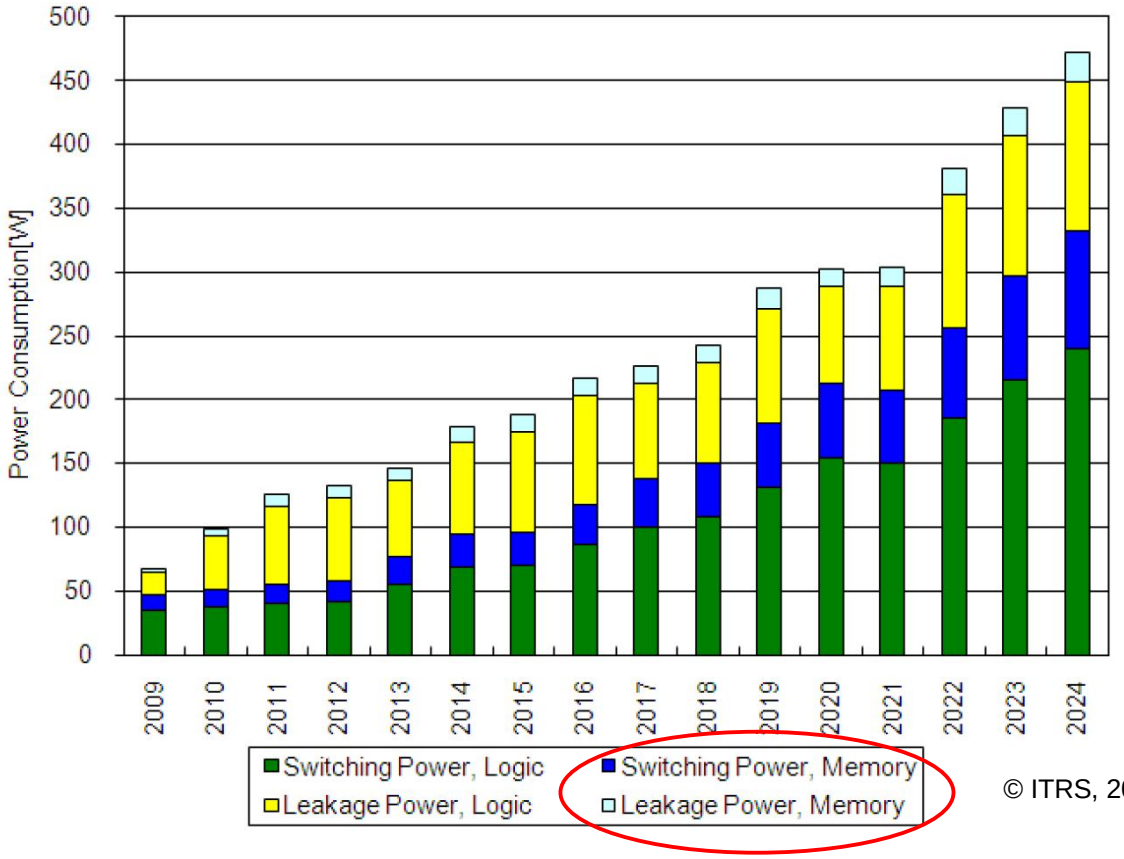


© ITRS, 2010

- Memory and logic, static and dynamic relevant
- Following current trends will violate maximum power constraint (0.5-1 W).

Where is the power consumed?
- Stationary systems -

- According to *International Technology Roadmap for Semi-conductors* (ITRS), 2010 update, [www.itrs.net]



- Switching power, logic dominating
- Overall power consumption a nightmare for environmentalists

Shift towards Cloud Computing Helps

- Ubiquitous connectivity & access to data
- Consolidate servers ➔ Amortize energy costs

But, the Cloud has hit a wall!

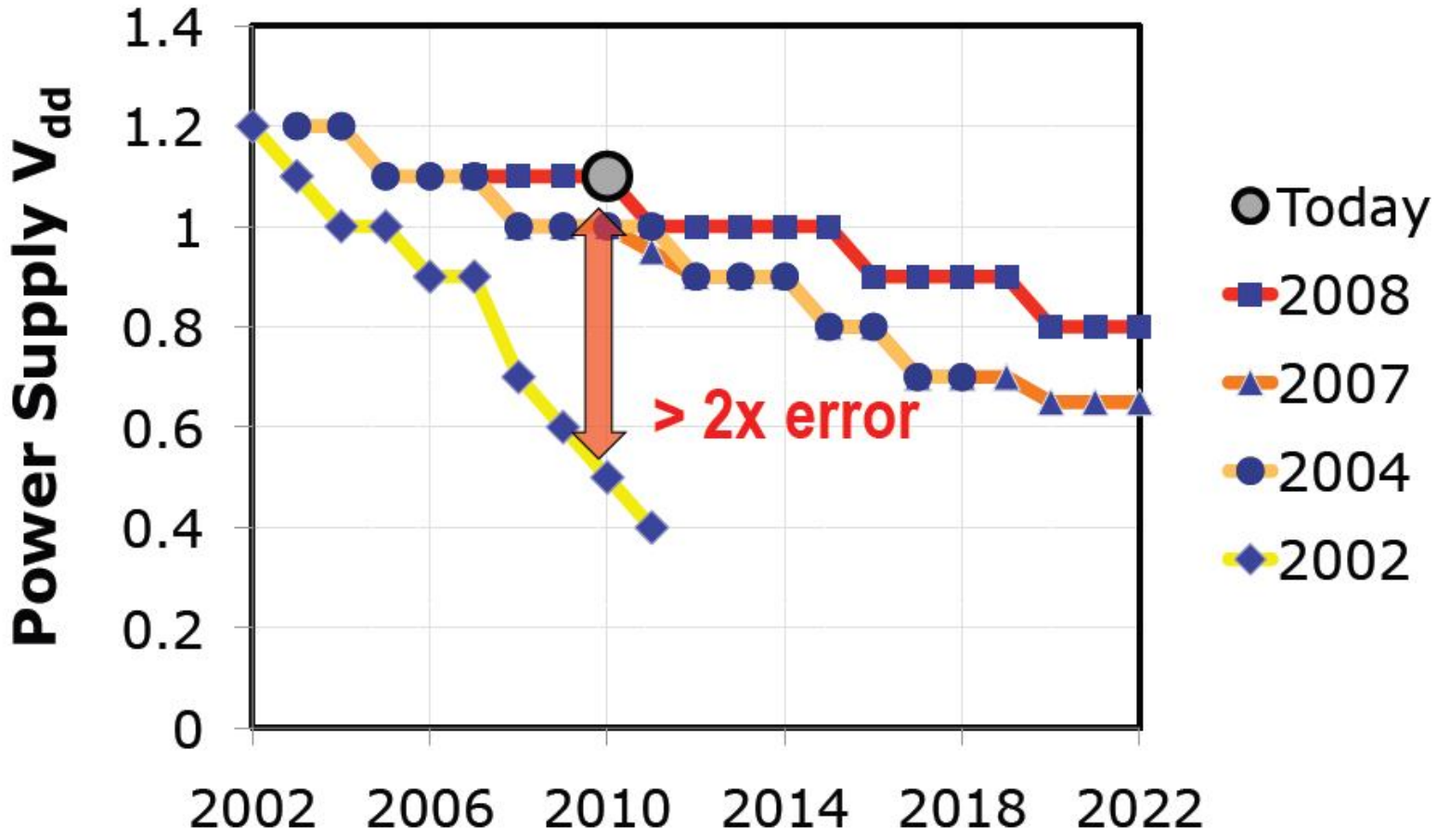
Trends:

- Moore's law continues
- Server density is increasing
- But, voltage scaling has slowed

→It's too expensive to buy/cool servers

A 1,000m² datacenter is 1.5MW!
(carbon footprint of airlines in 2012)

Voltages have already leveled off



ITRS estimates for today were off by > 2x

A few words about our model

Physical char. modeled after Niagara

Area: cores/caches (72% die)

- scaled across tech. nodes

Power:

- Active: projected V_{dd}/ITRS
  - Core=scaled, cache=f(miss), crossbar=f(hops)
- Leakage: projected V_{th}/ITRS, f(area), 62C

Performance:

- Parameters from real server workloads (DB2, Oracle, Apache, Zeus)
- Cache miss rate model (validated)
- CPI model based on miss rate

Caveat: Simple Parallelizable Workloads

Workloads are assumed parallel

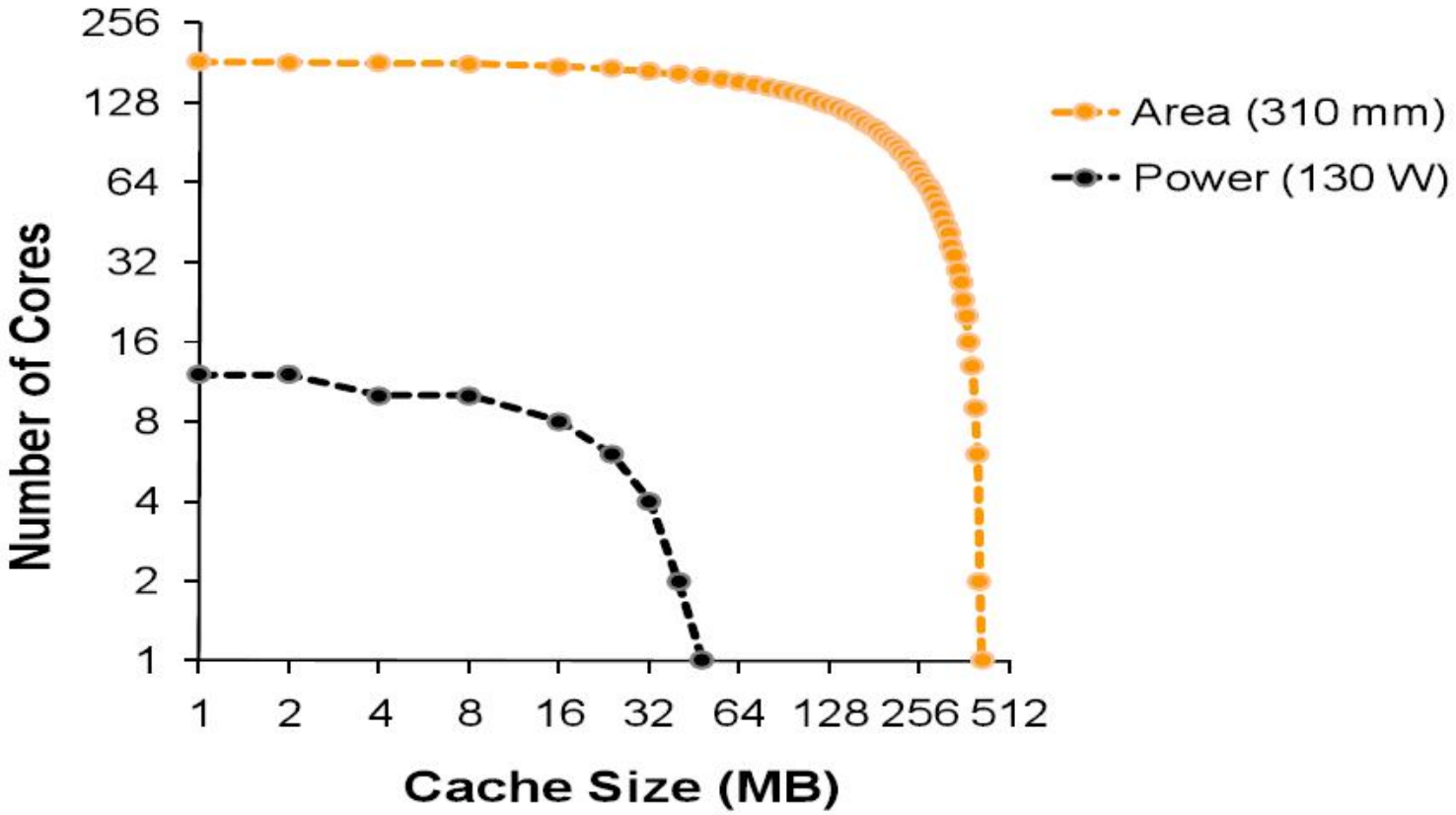
- Scaling server workloads is reasonable

CPI model:

- Works well for workloads with low MLP
- OLTP, Web & DSS are mostly memory-latency dependent

Future servers will run a mix of workloads

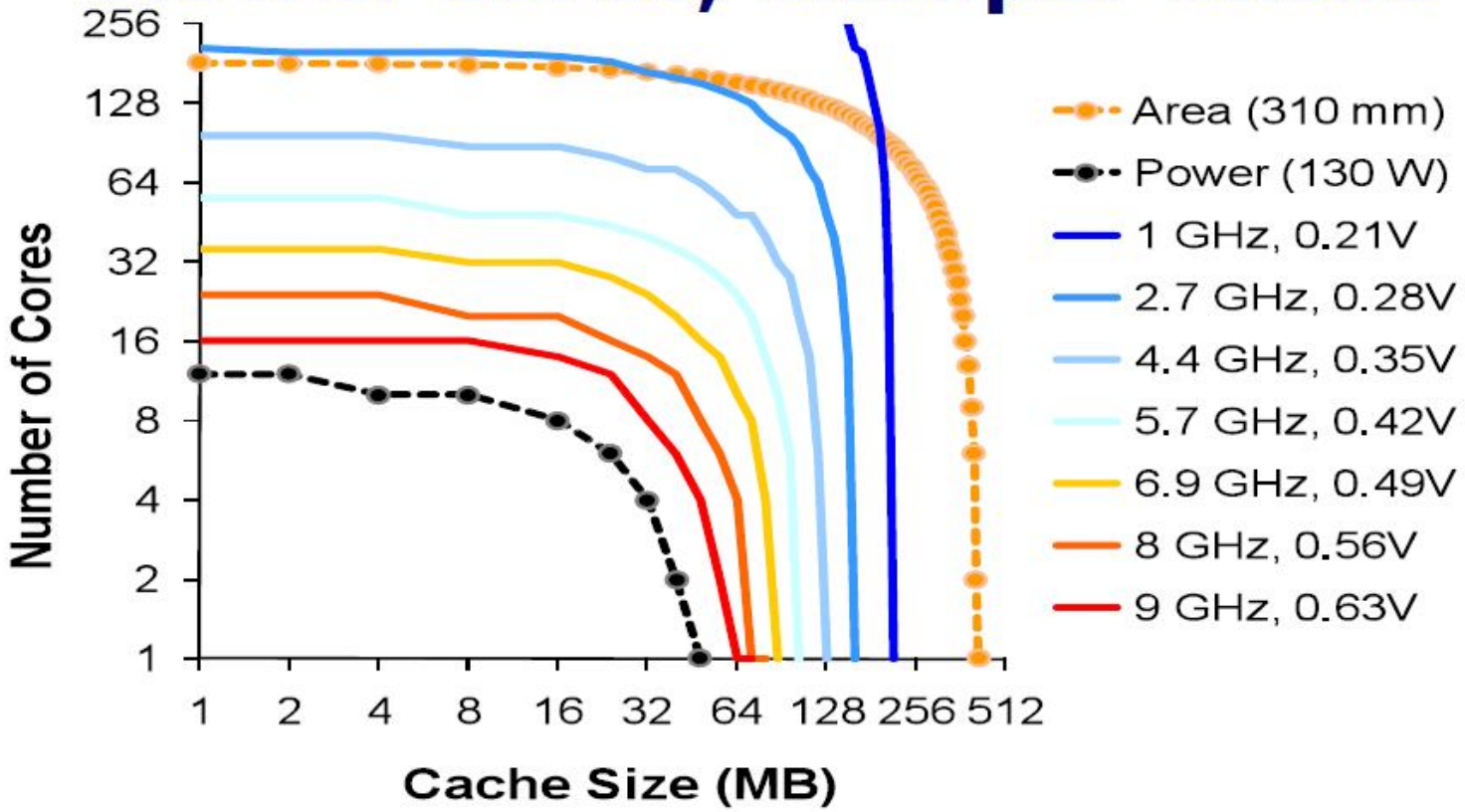
Area vs. Power Envelope (22nm)



✓ Good news: can fit hundreds of cores

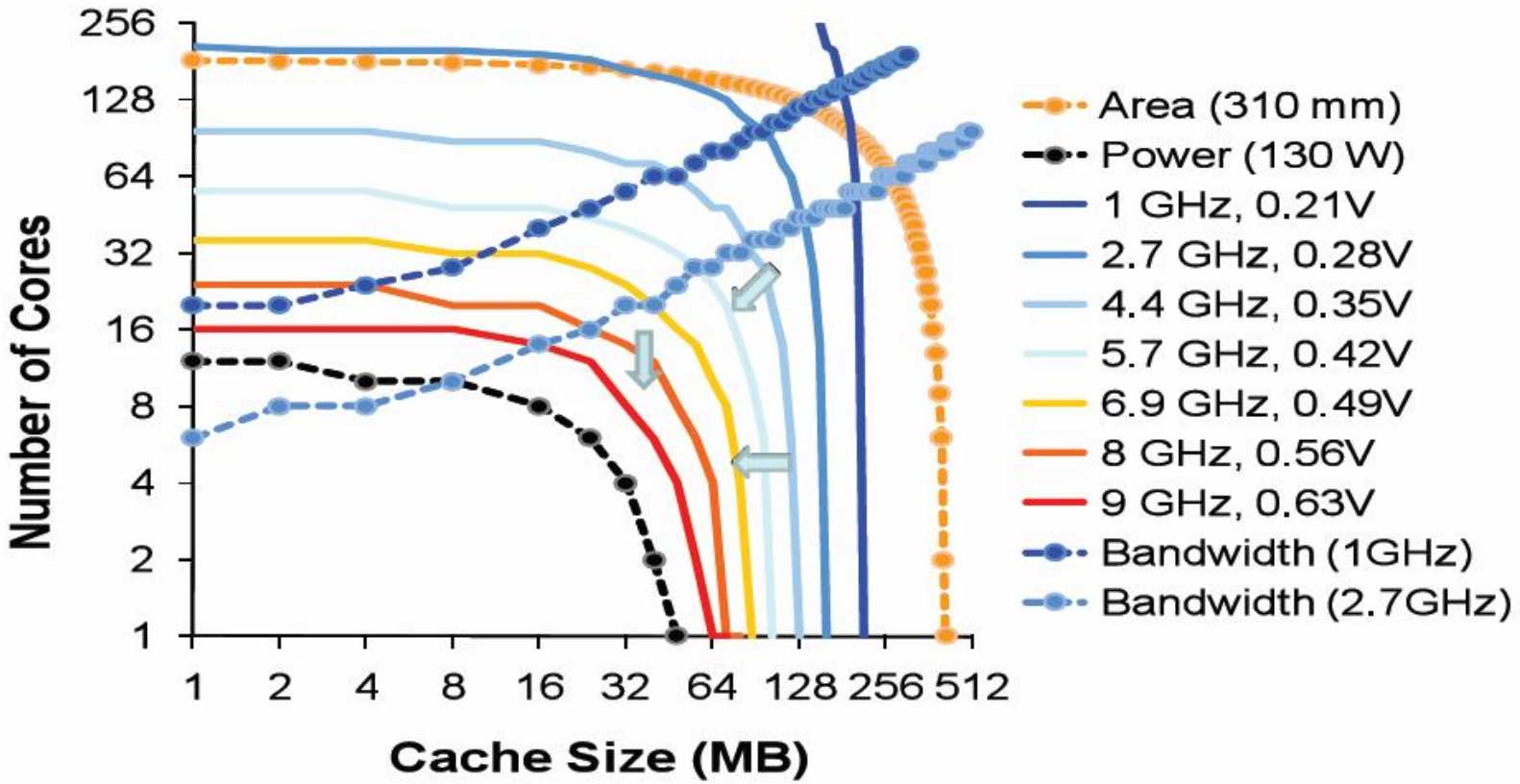
× Can not use them all at highest speed

Of course one could pack more slower cores, cheaper cache



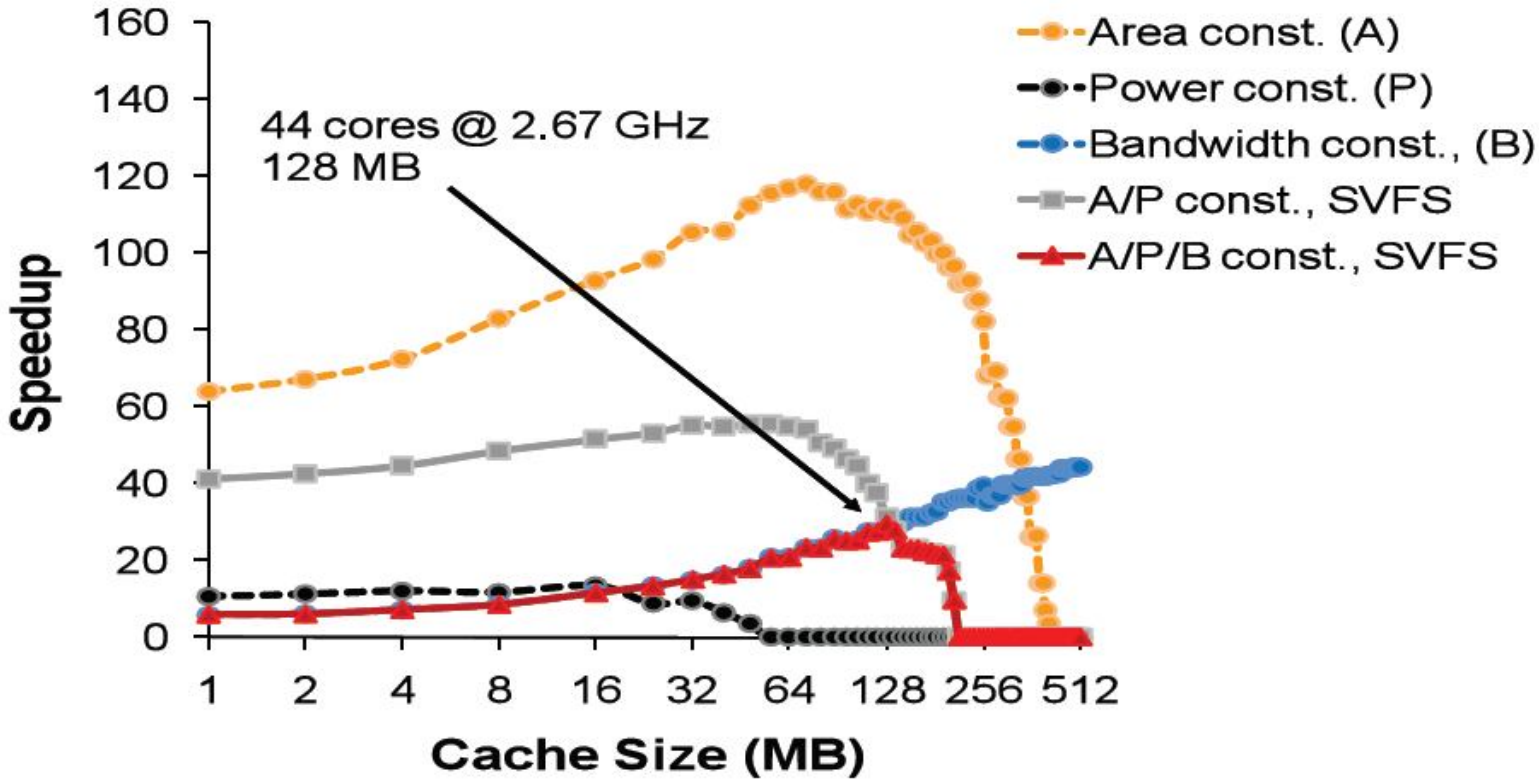
- Result: a performance/power trade-off
- Assuming bandwidth is unlimited

But, limited pin b/w favors fewer cores + more cache



- For clarity, only showing two bandwidth lines
- Where would the best performance be?

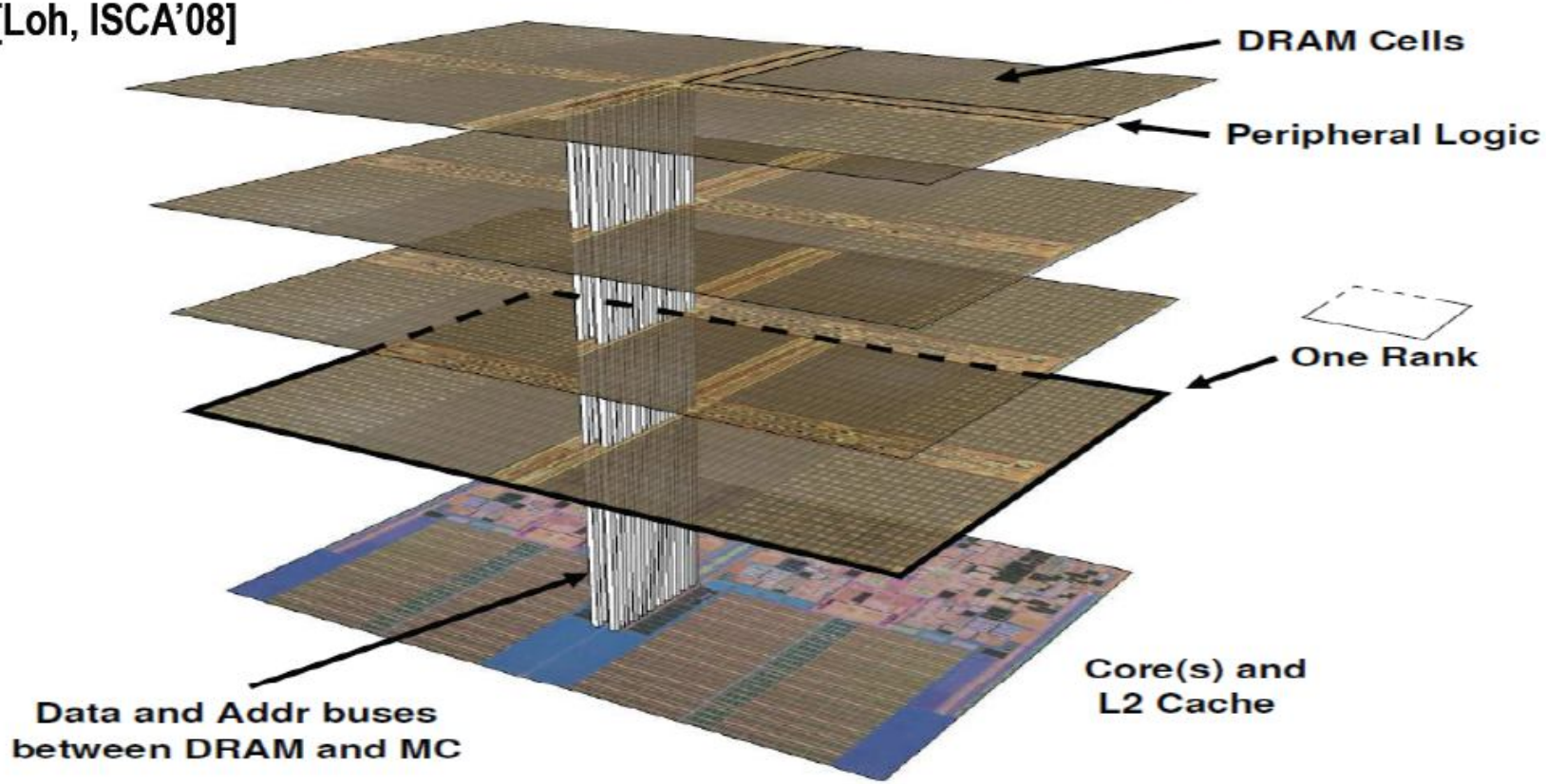
Peak Performing with Conventional Memory



- B/W constrained, then power constrained
- Fewer slower cores, lots of cache

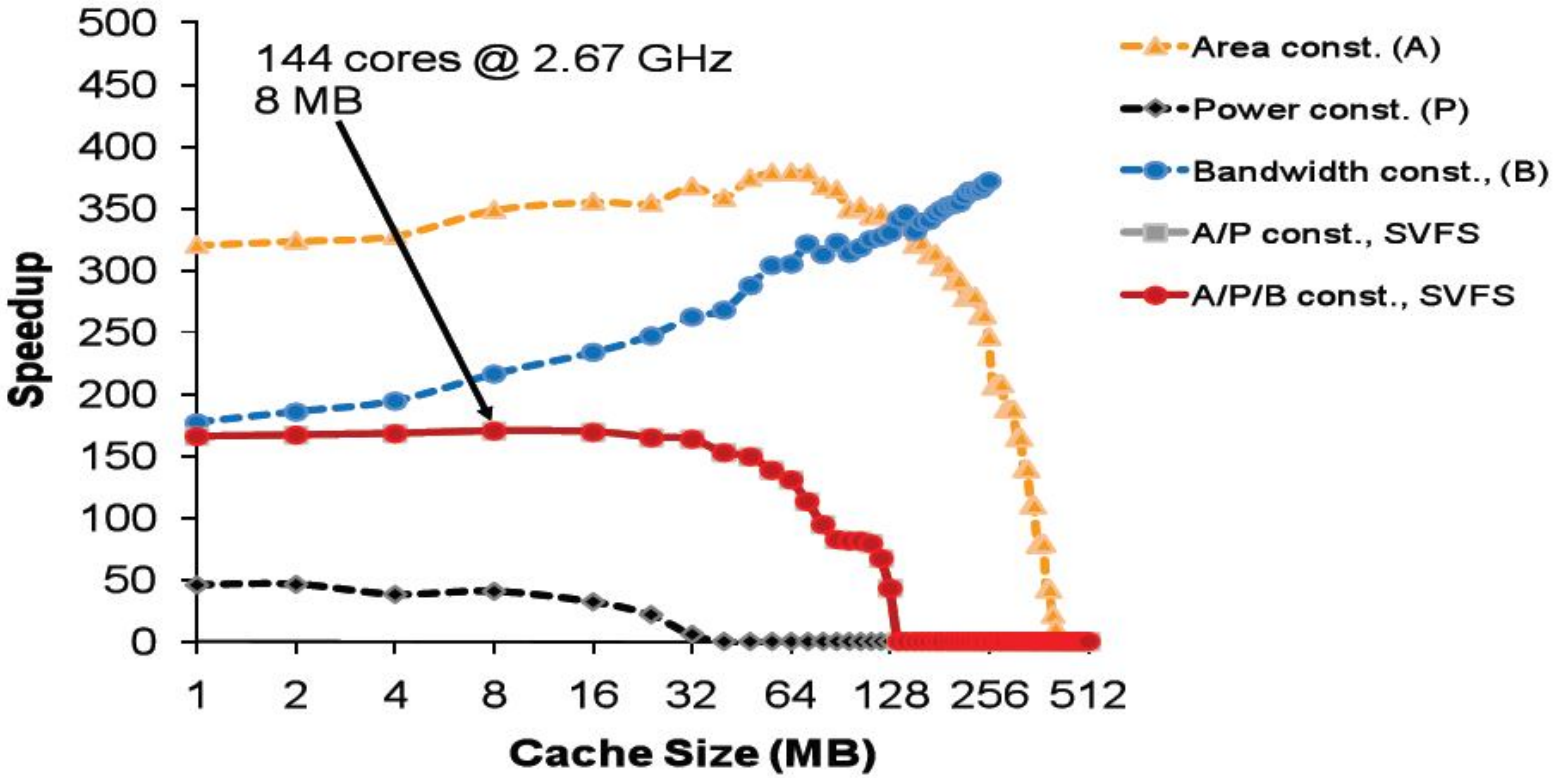
Mitigating B/W Limitations: 3D-stacked Memory

[Loh, ISCA'08]



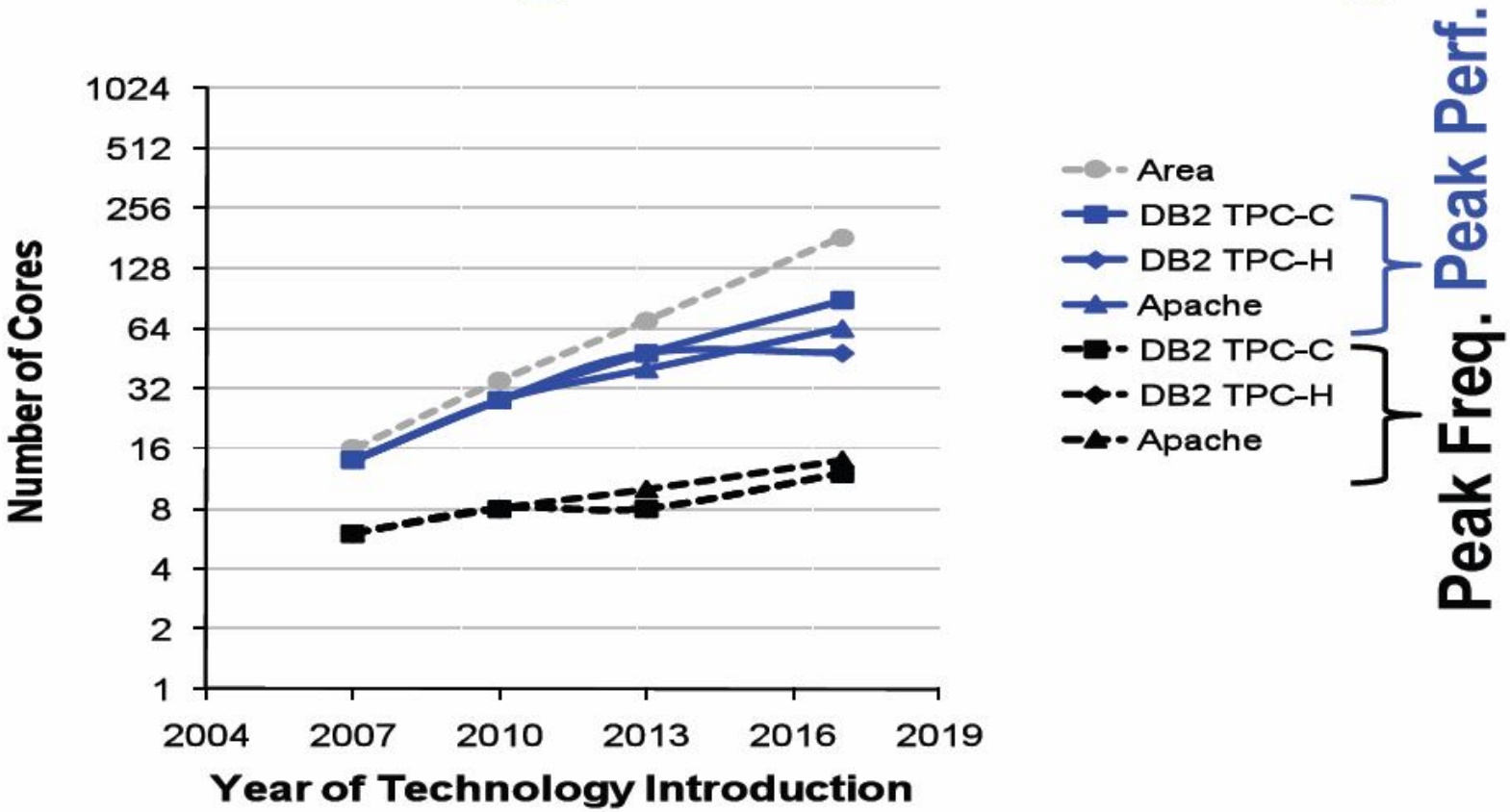
- Delivers TB/sec of bandwidth

Peak Performing w/ 3D-stacked Memory



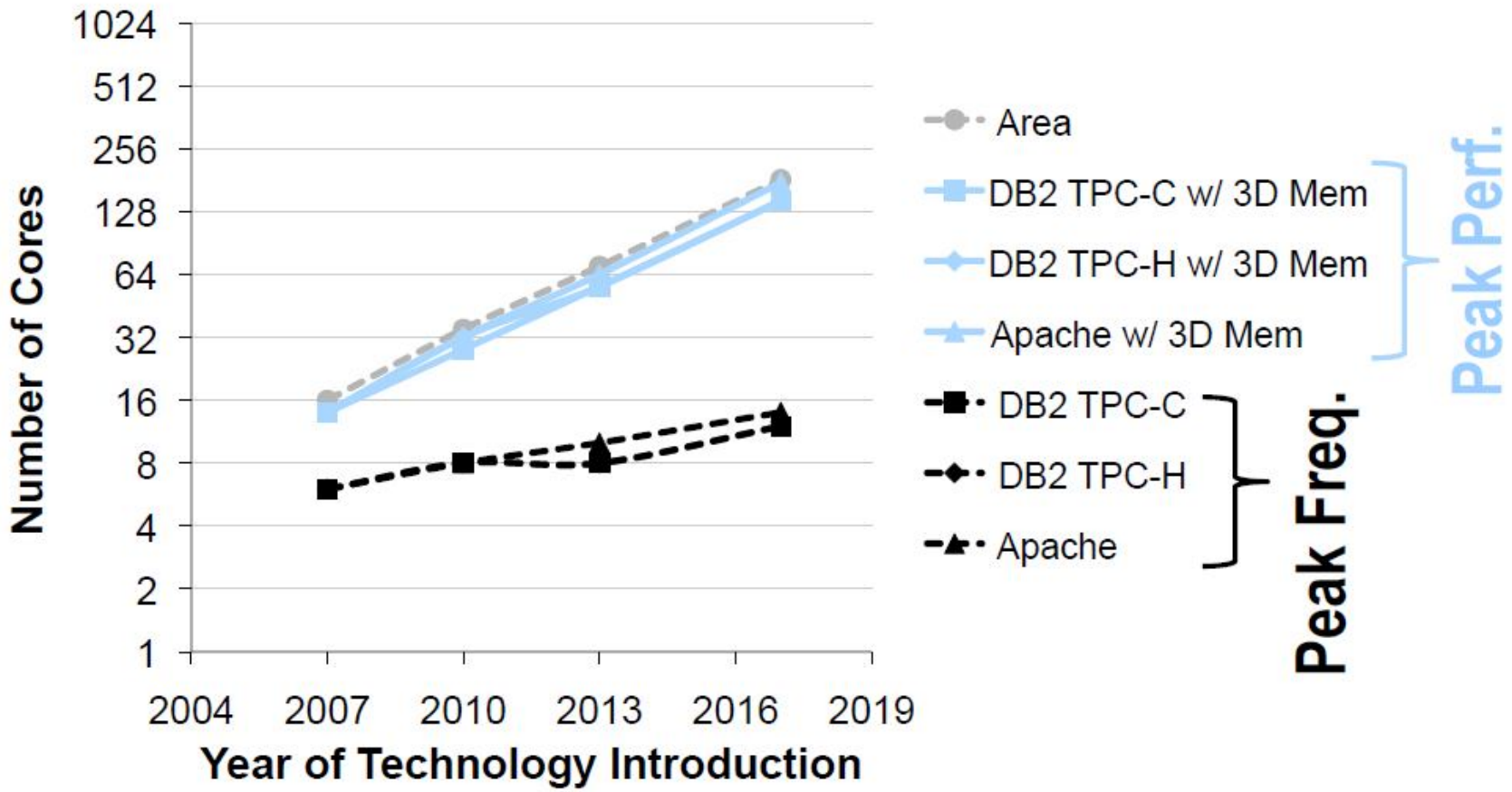
- Only power-constrained
- Virtually eliminates on-chip cache

Core Scaling across Technologies



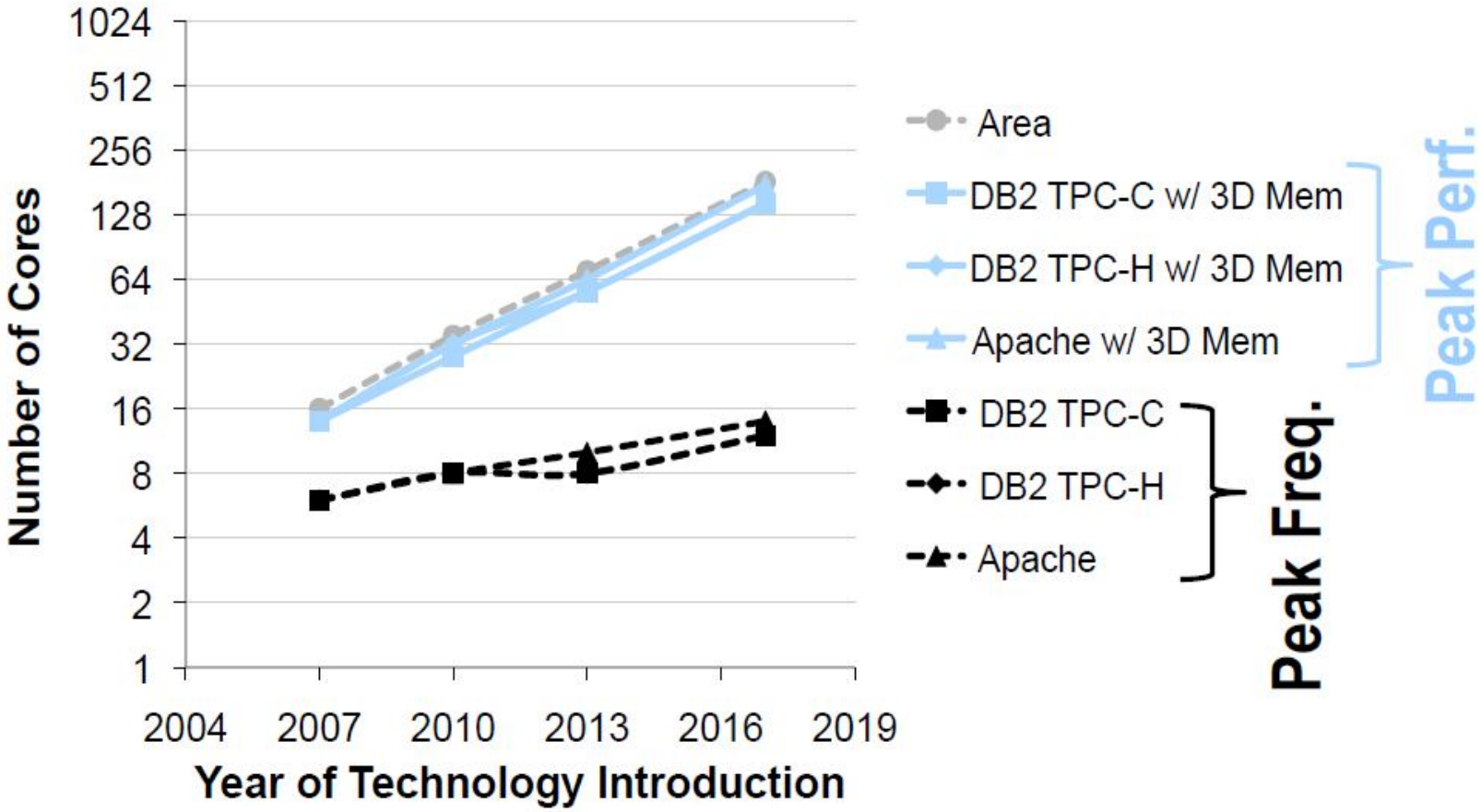
- Assumes a 130-Watt chip envelope
- Pin b/w keeps Niagara from scaling

Niagara + 3D-stacked Memory



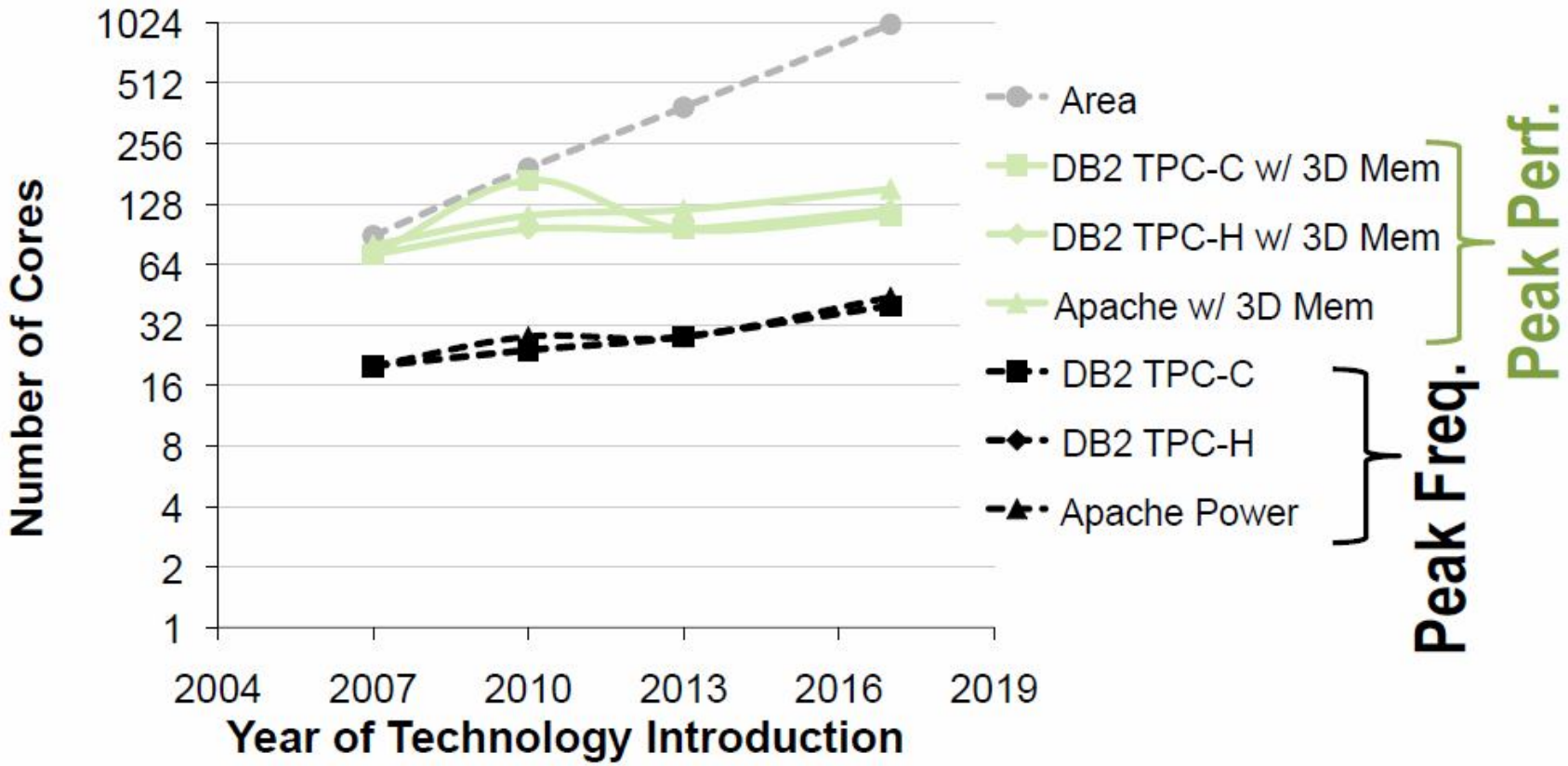
- Power limits Niagara to 75% area!

Niagara + 3D-stacked Memory



- Power limits Niagara to 75% area!

ARM9 + 3D-stacked Memory



- Can not scale with a 130-Watt envelope!!!
- On-chip hierarchy + interconnect not scalable

Long-term: Where to go from here?

1. Redo SW stack
   - Minimize joules/work (algo. down to HW)
   - Program for locality + heterogeneity

2. Pray for technology
   - Energy-scalable silicon devices
   - Emerging nanoscale technologies?

3. Infrastructure technology
   - Renewable/carbon-neutral energy
   - Scalable cooling + power delivery

Short-term Scaling Implications

- Caches are getting huge
  - Need cache architectures to deal with >> MB
  - E.g., Reactive NUCA [ISCA'09]
- Interconnect + cache hierarchy power
  - Need lean on-chip communication/storage
  - Eurocloud chip: ARM+3D [ACLD'10]
- Dark Silicon
  - Specialized processors
  - Use only parts of the chip at a time

Zusammenfassung

- Der Trend zur weiteren Miniaturisierung von Schaltungen hält vermutlich noch für einige Jahre weiterhin an (wie viele?)
- Allerdings trifft man neben der memory *wall* jetzt auch auf die *power wall*
- Man kann mehr Transistoren auf die Chips integrieren, aber sie nicht mehr alle gleichzeitig mit Strom versorgen (☞ ***dark silicon***)

- Partielle Abhilfen:
  - *Embedded* Prozessoren (z.B. ARM) als Server?
  - 3D-Integration von Speicher?
  - Über die Energieeffizienz von Software nachdenken!