MICROELECTRONIC
SYSTEMS DESIGN
RESEARCH GROUP

Workshop Session on:

3D Emerging Memories and New Architecture Paradigms

3D Memories: Now and Then!

Hybrid, Cubes, Approximate, and Custom

Dr.-Ing. Christian Weis

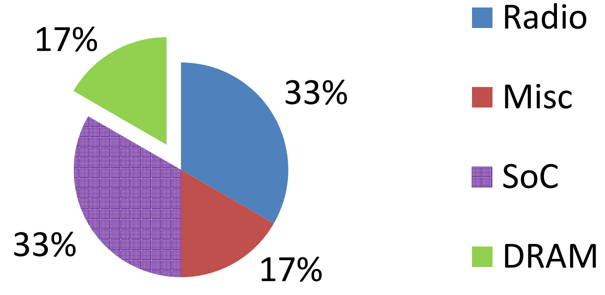
SPP1500

DFG Deutsche Forschungsgemeinschaft

Technische Universität Kaiserslautern

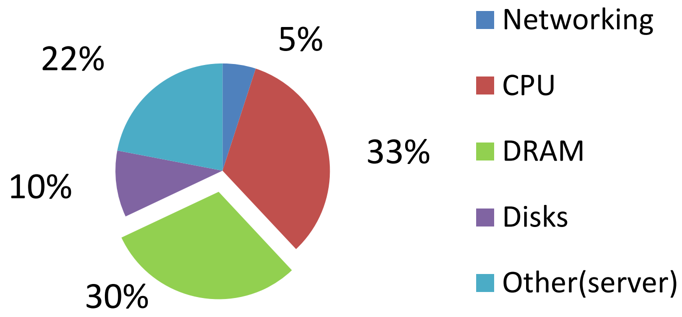
Why do we care about DRAM ?

Power Break-Down for Suspended 3G State



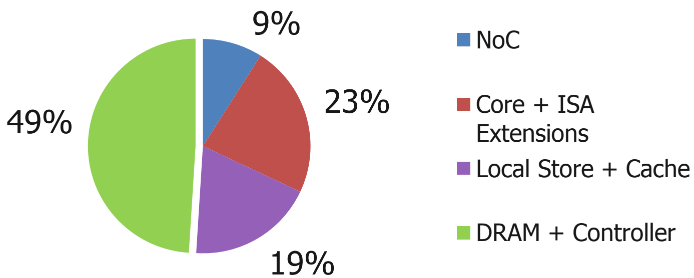
Source: The systems hackers guide to the galaxy: Energy usage in a modern smartphone

Power Break-Down Google Datacenter



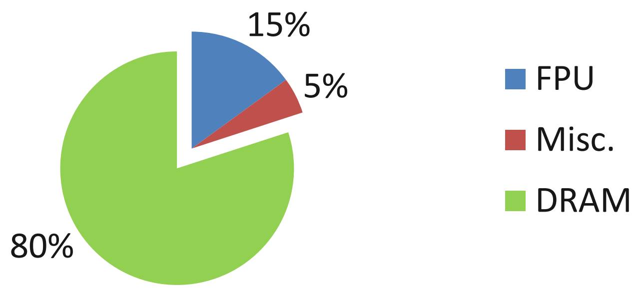
Source: The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines. 2009

Power Break-down for Big Data Application



Source: Power Consumption of Green Wave Architecture 2011

Power Break-Down eBrain

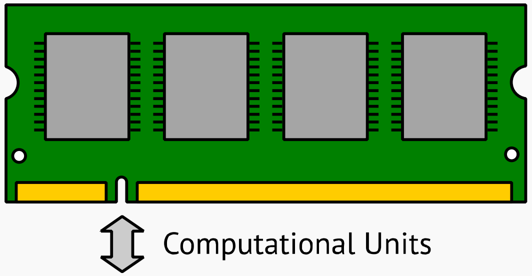


Source: A Scalable Custom Simulation Machine for the Bayesian Confidence Propagation Neural Network model of the Brain, 2014

Comparison of DRAM Subsystems

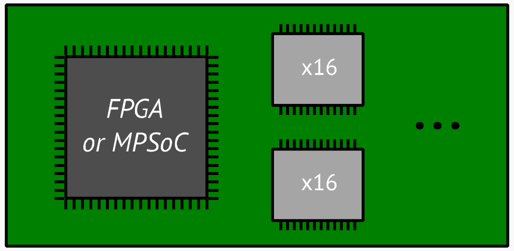
DIMM Based:

General Purpose Computers
e.g. DDR3, DDR4



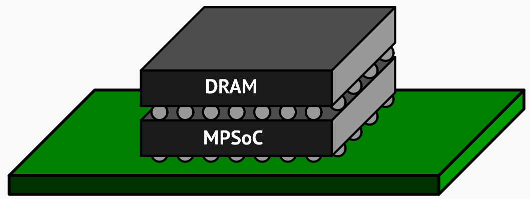
Device Based:

Embedded / Tablets / Graphic Cards
e.g. LPDDR3, GDDR5



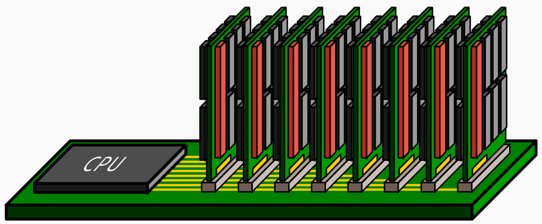
Package on Package (PoP):

Soldered on top of the MPSoC.
Smartphones
e.g. LPDDR3, LPDDR4



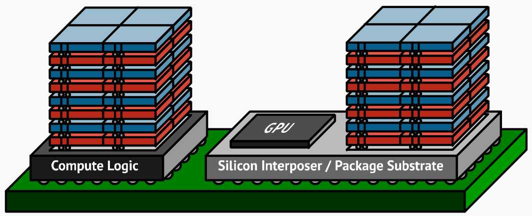
Buffer on Board:

Memory Controller on Buffer Chip,
Serial Connection
e.g. FBDIMM, IBM CDIMM, Intel SMI/SMB



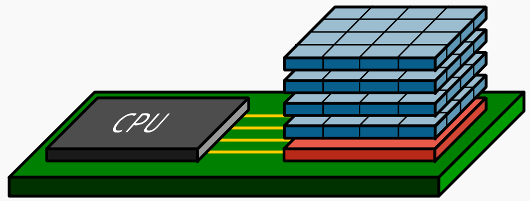
3D/2.5D-Integrated:

Stacked on Logic or Silicon Interposer
by means of TSVs
e.g. Wide I/O, HBM



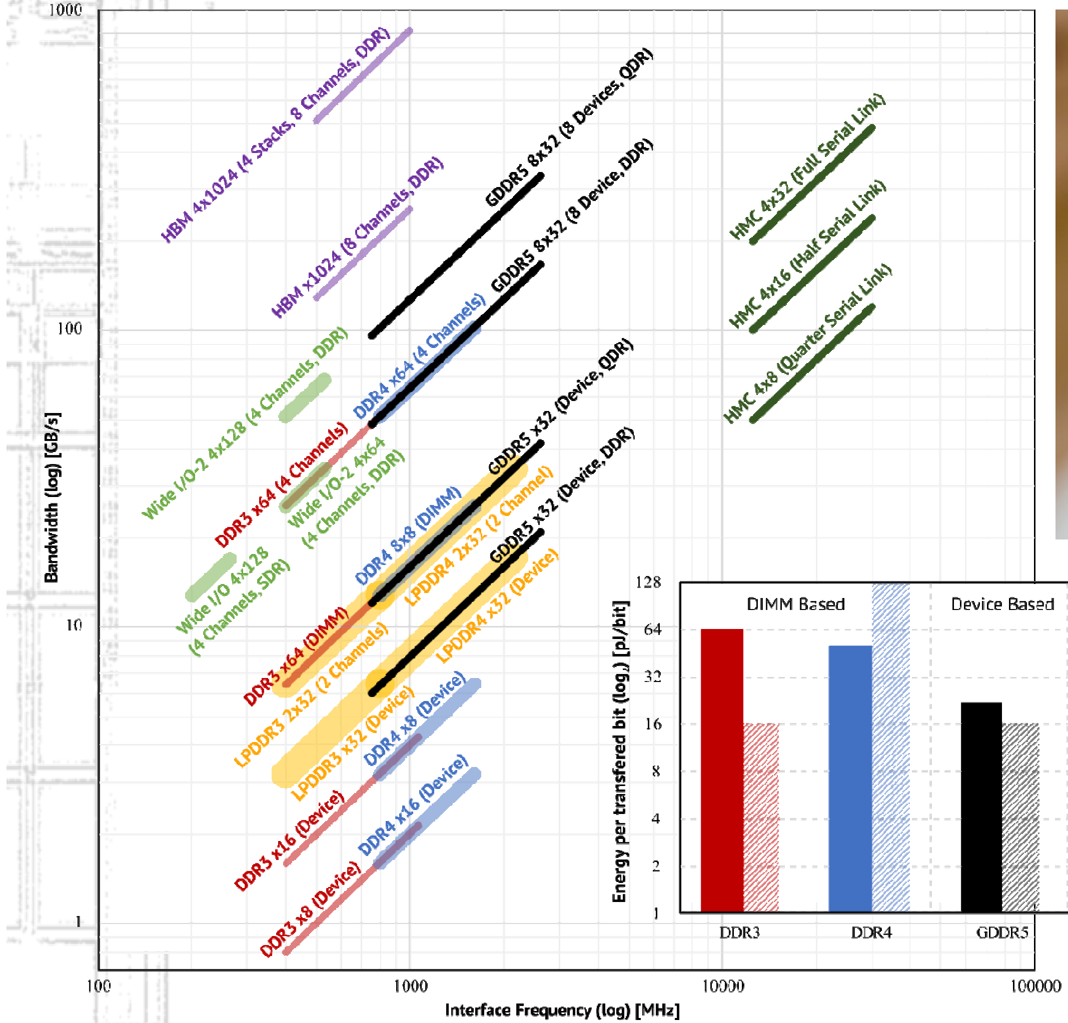
Memory Cube:

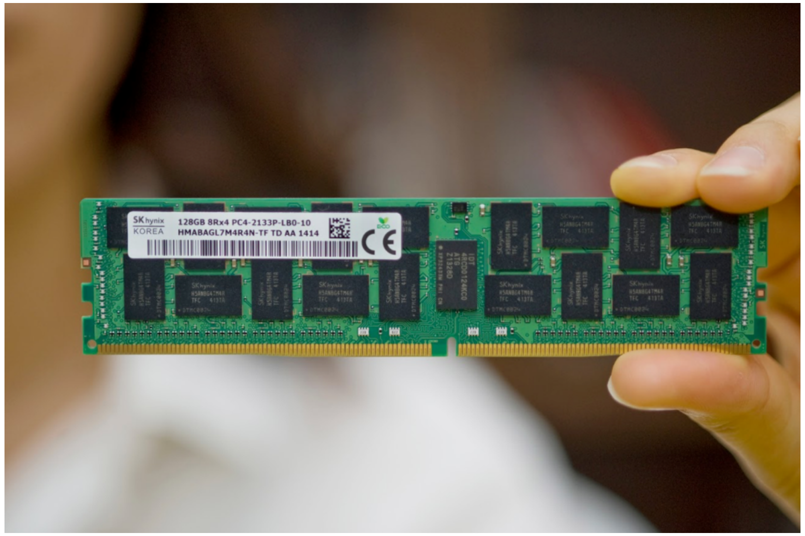
3D-Stacked, Memory Controller on
Bottom Layer, Serial Interconnect (SerDes)
e.g. HMC, SMC

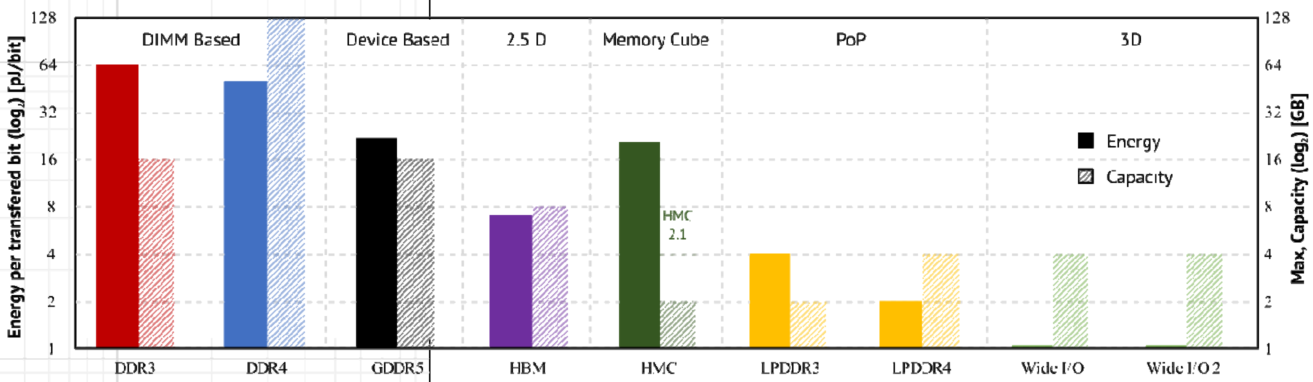


Source: Matthias Jung

Comparison of DRAM Subsystems







Source: Matthias Jung

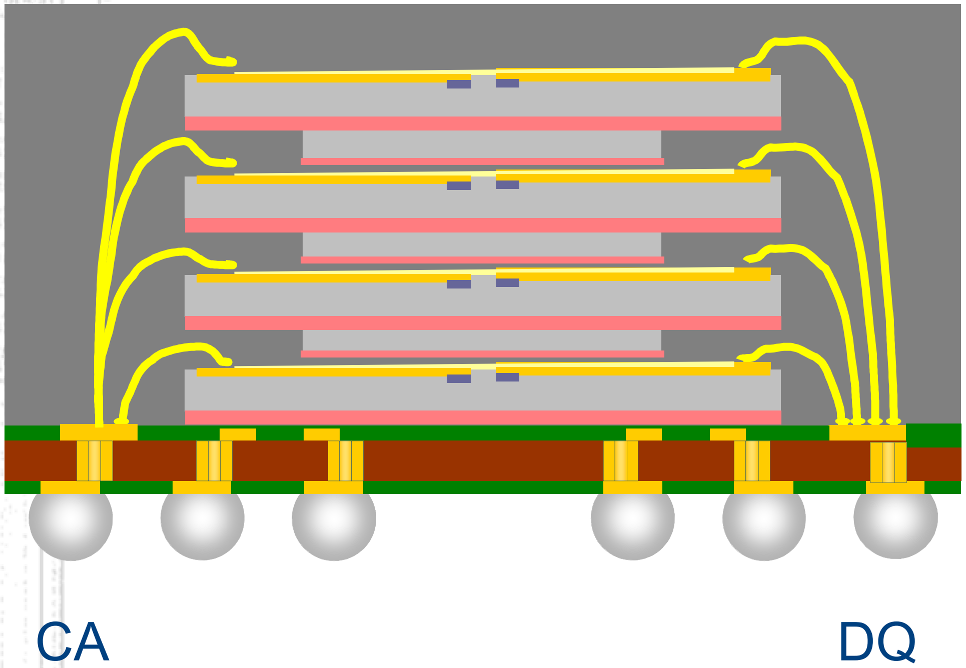
Best case - 100% usage of the available BW

3D DRAM's starting point …

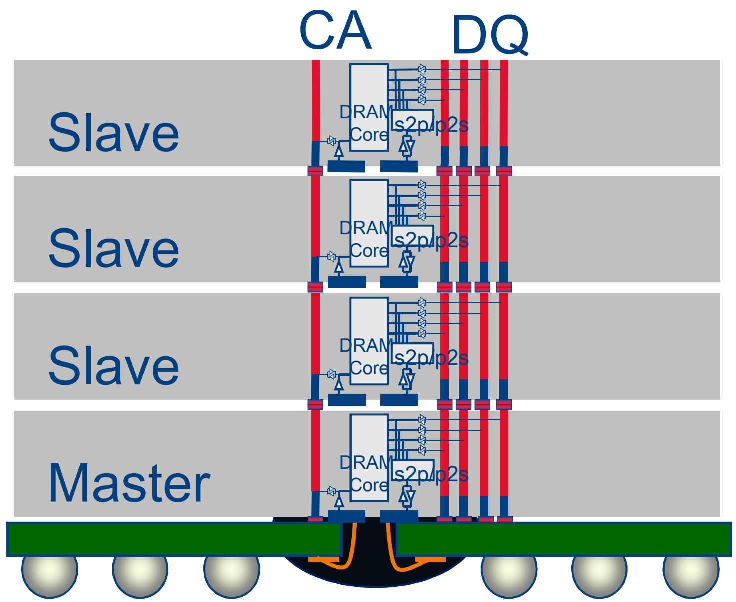
Reducing the I/O loads – a performance and power advantage:

- much smaller capacitances for a TSV stack than a Quad die

Conventional quad-die stack



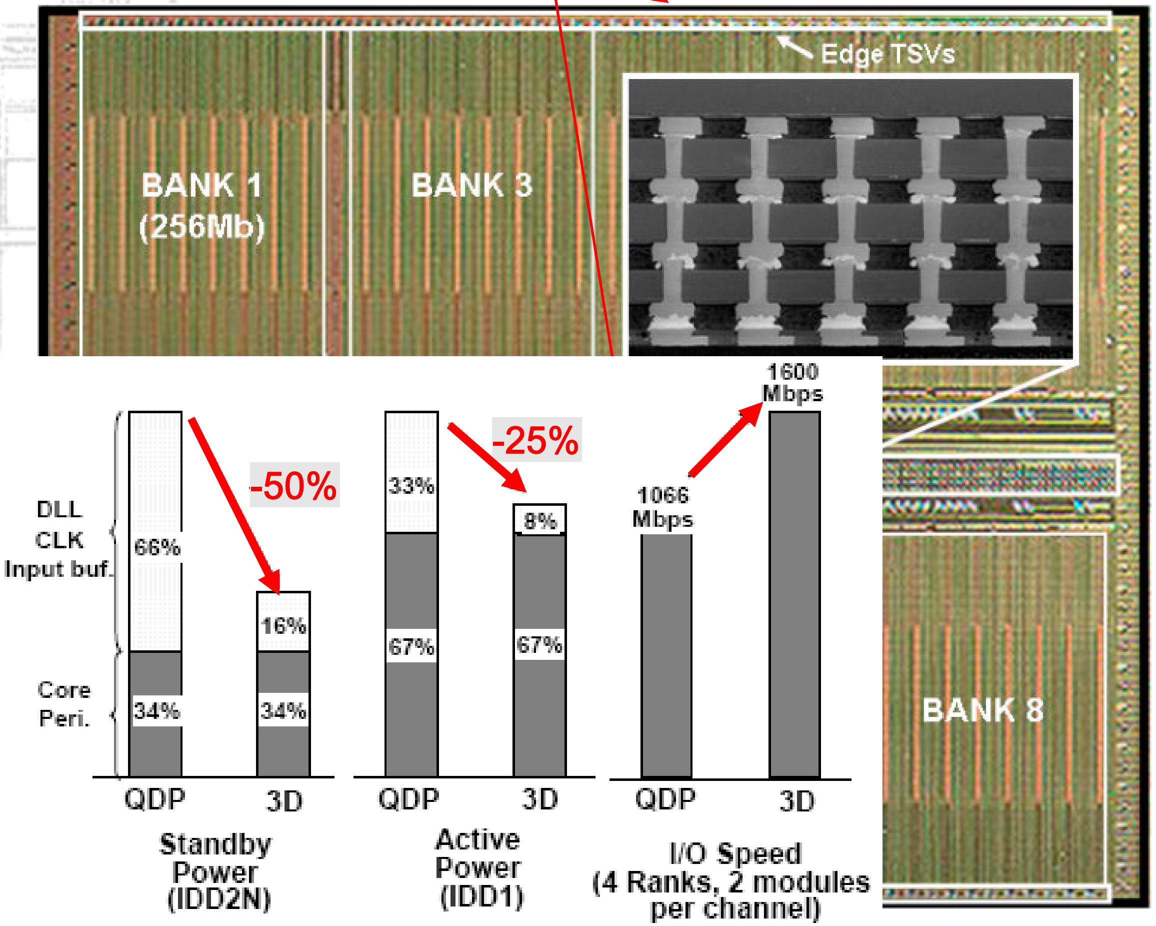
TSV stack:

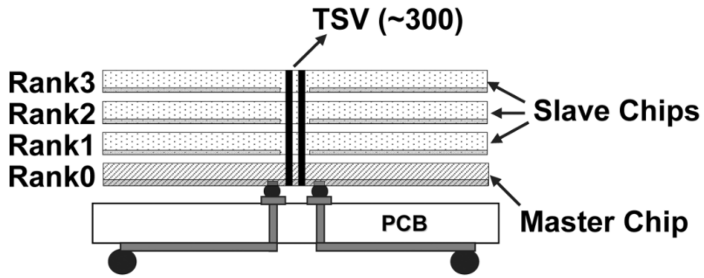


source: Qimonda, 2008

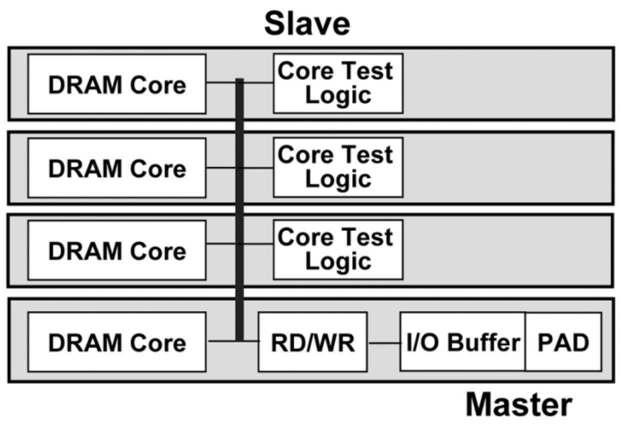
3D DRAM packaging example

- 3D Packaging with a commodity 2Gb DDR3 SDRAM chip (4x 2Gb = 8Gb)
- With areas reserved for TSVs



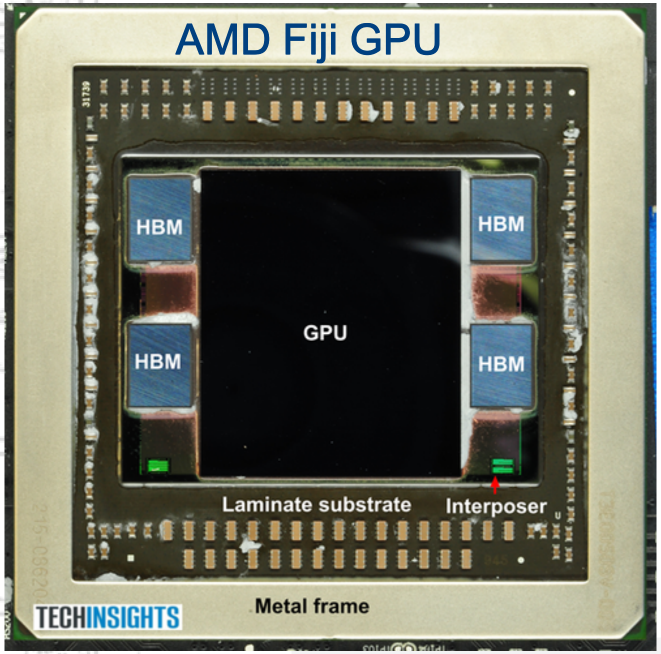


12.8 GB/s
DIMM Bandwidth

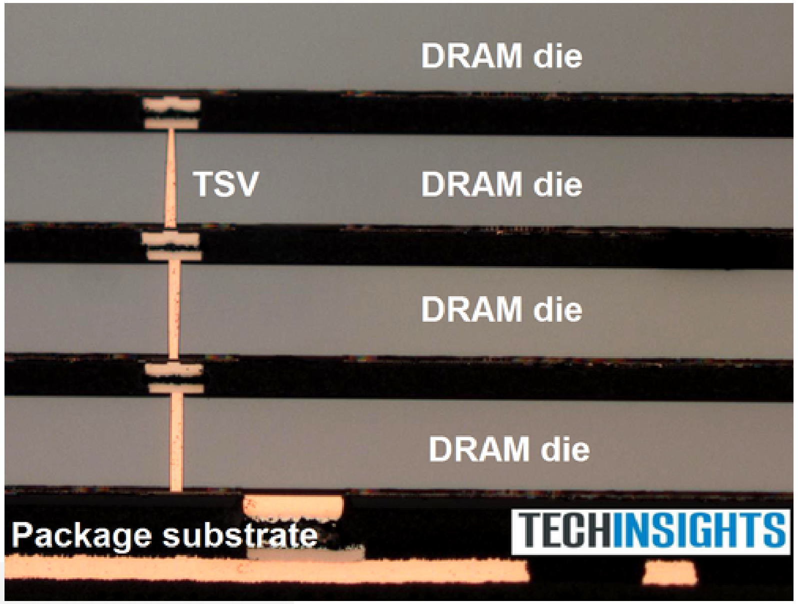


source: Samsung'09

3D Integration: State-of-the-art

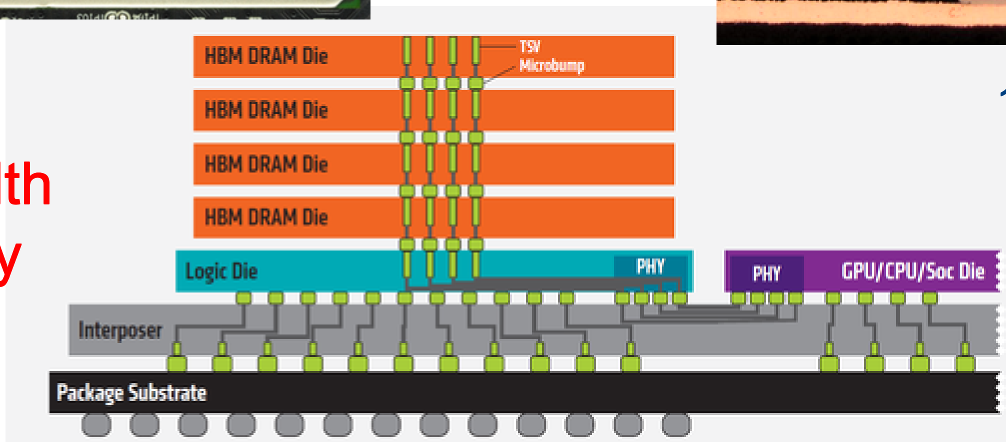


Graphics Memory



512 GB/s
Bandwidth

High
Bandwidth
Memory
(HBM)



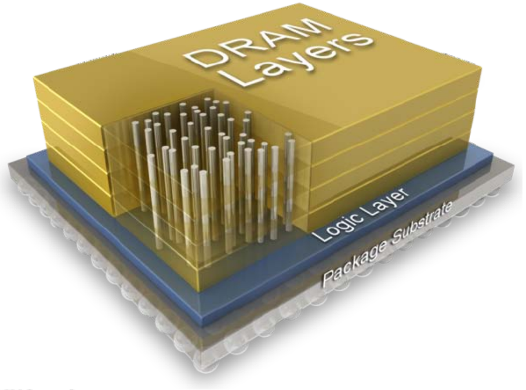
1024bit I/O per HBM cube

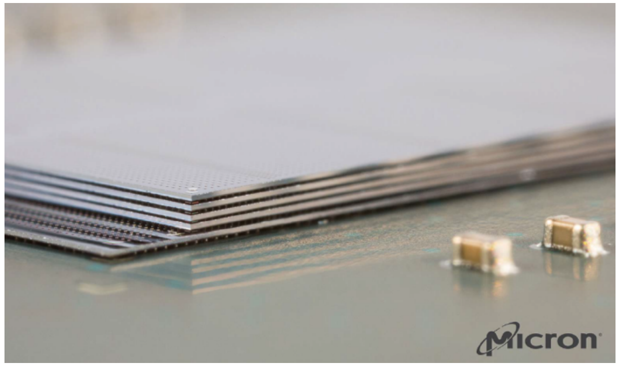
Light-weight Logic
layer interface
@500MHz DDR

source: AMD, June 2015

3D Integration: State-of-the-art

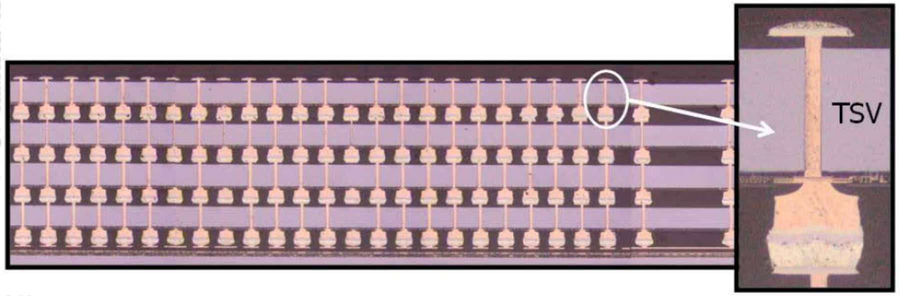
DRAM Cube with Abstracted Interface

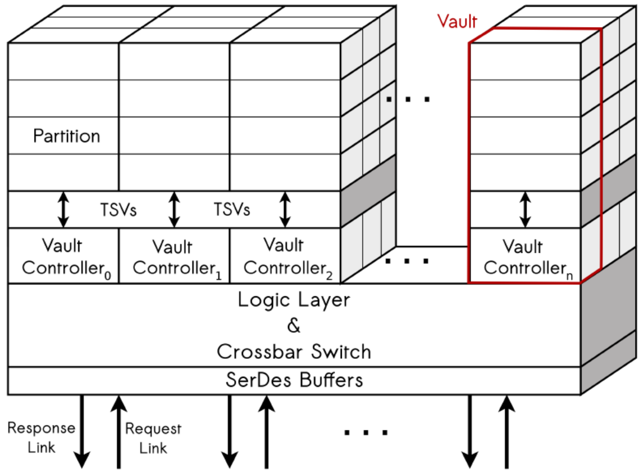




160+ GB/s Bandwidth

32 bit DDR I/O per Vault





Hybrid Memory Cube (HMC)

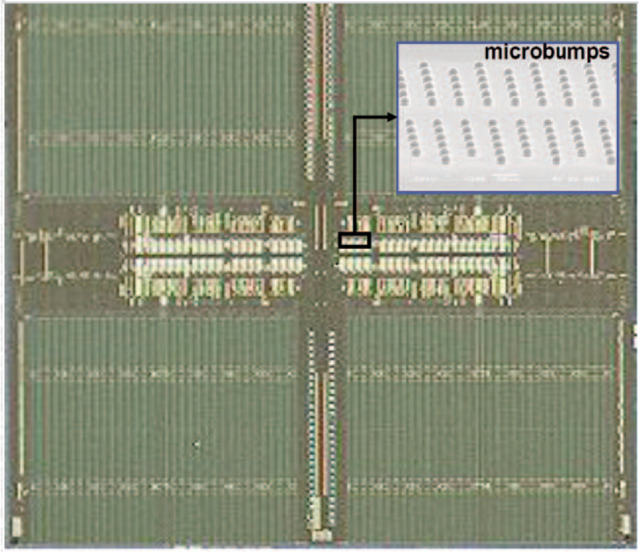
Source: Micron, 2014

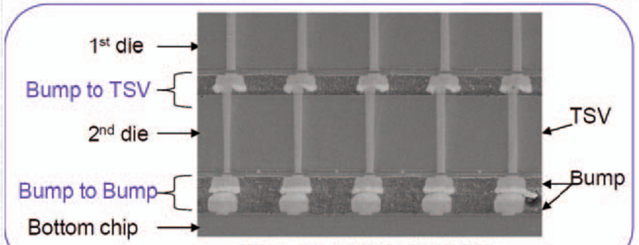
3D Integration: State-of-the-art

WIDE-IO DRAM 1st Gen

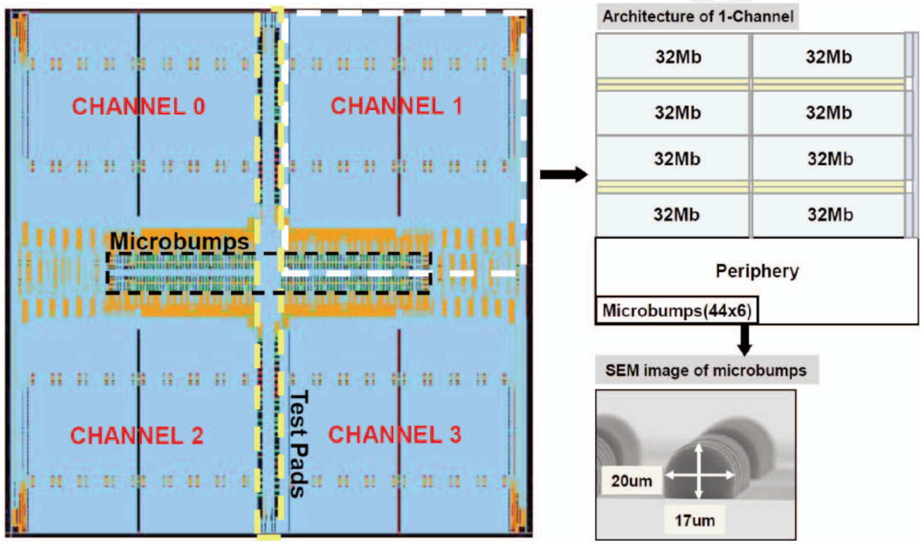
Chip architecture of 1Gb Wide-IO DRAM and SEM image of microbumps

Chip photograph





[2 stacked WIDE IO PKG]

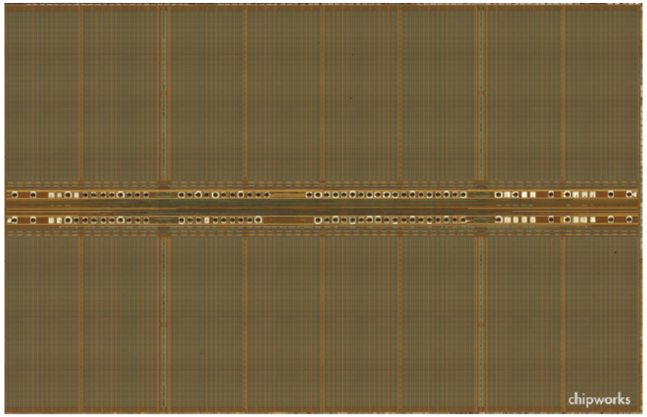


Device			MDDR	LPDDR2	Wide IO
Density			1Gb	1Gb	1Gb
Organization			4 Bank / x32	8 Bank / x32	16 Bank / x512
VDD [V]			1.8	1.2	1.2
Data Rate [MHz]			400	800	200
Data Bandwidth [GB/s]			1.6 (100%)	3.2 (200%)	12.8 (800%)
Meas. Power [mW]	Standby		0.32 (100%)	0.27 (83.3%)	0.27 (83.3%)
	Read	DQ	215.8 (100%)	221.2 (102.5%)	73.7 (34.2%)
		Total	322.3 (100%)	372.1 (115.4%)	330.6 (102.6%)
	I/O per pin [mW/Gbps]		17.33 (100%)	8.71 (50.3%	0.78 (4.5%)

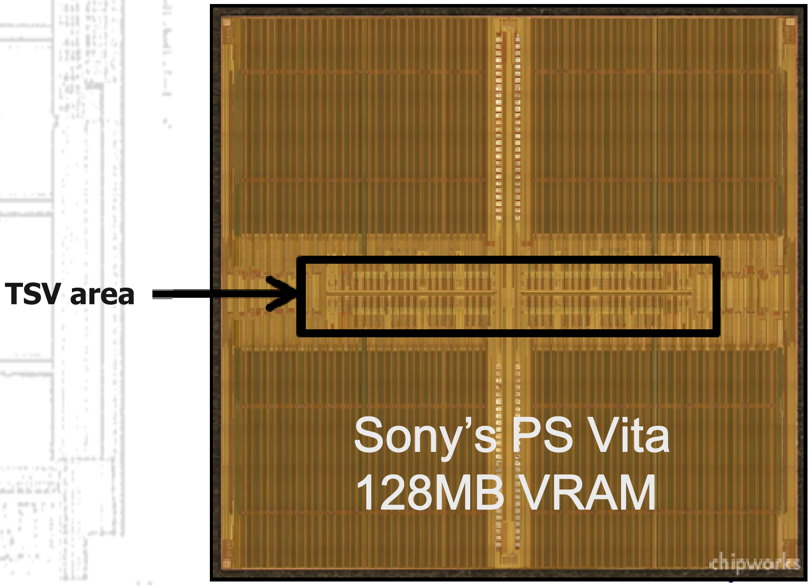
source : Samsung11

Different Die Flavors of DRAMs

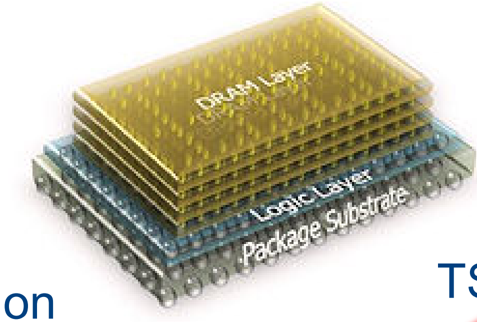
Commodity Samsung **2G DDR3** die:

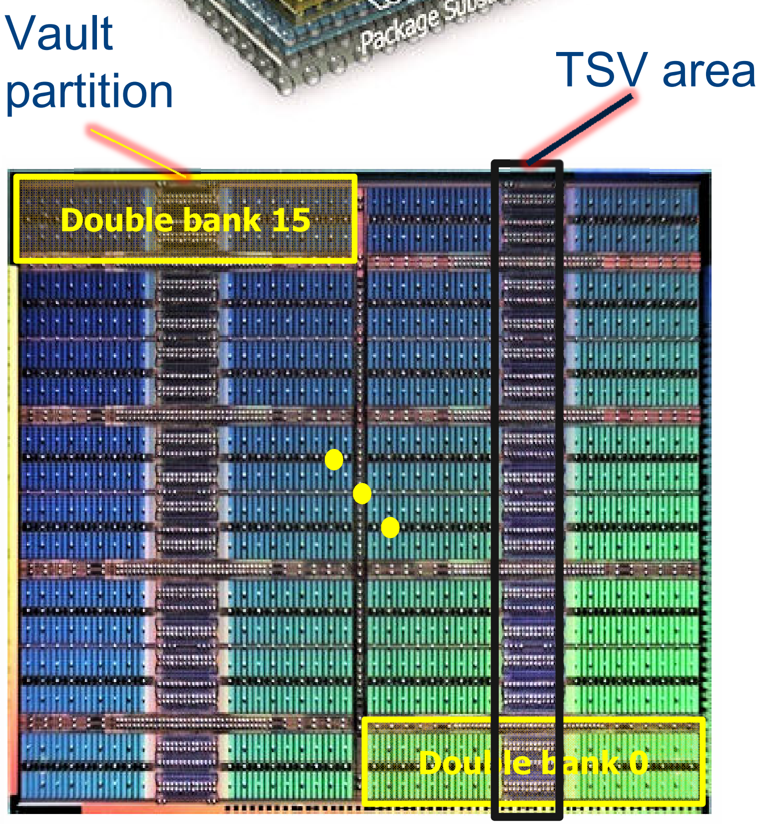


WIDE I/O 1Gb SDR JEDEC based die:



Micron´s **Hybrid Memory Cube** (HMC):





DRAM Layer

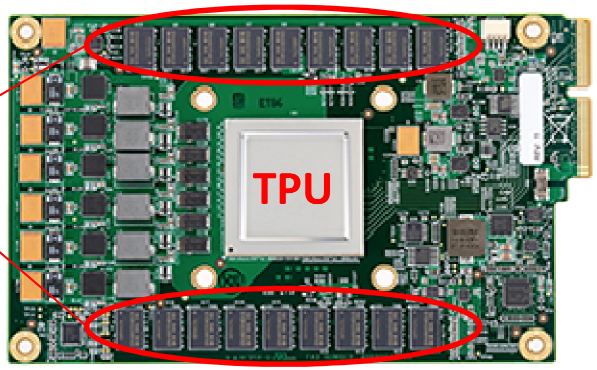
Does 3D help to do it better?

3 severe problems appeared during the last years:

1. **DRAMs don't like heat** → 2.5D integration or very good heat control in the underlying logic layer (uProc)
2. **When not** using direct **3D stacking** (on top of uProc), **how to get this huge bandwidth out** of the devices?
3. **Memory centric computing**, such as neuromorphic, NNs, or DL makes it even worse …

~ 30 GB/s
2 Channel Bandwidth

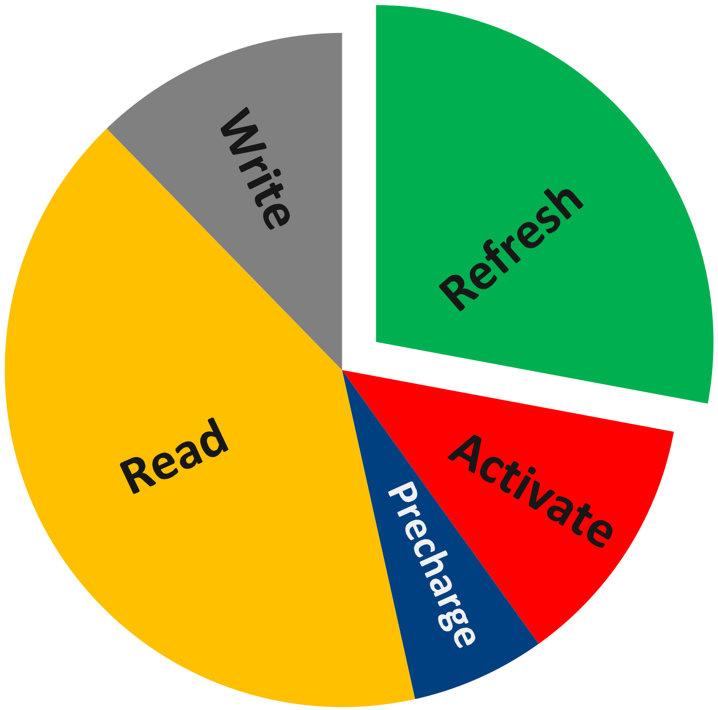
70 Gops/W – 2 Tops/W

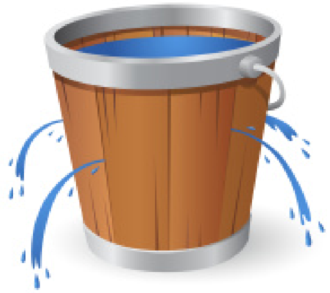


Google, 2017

DRAM Energy Distribution

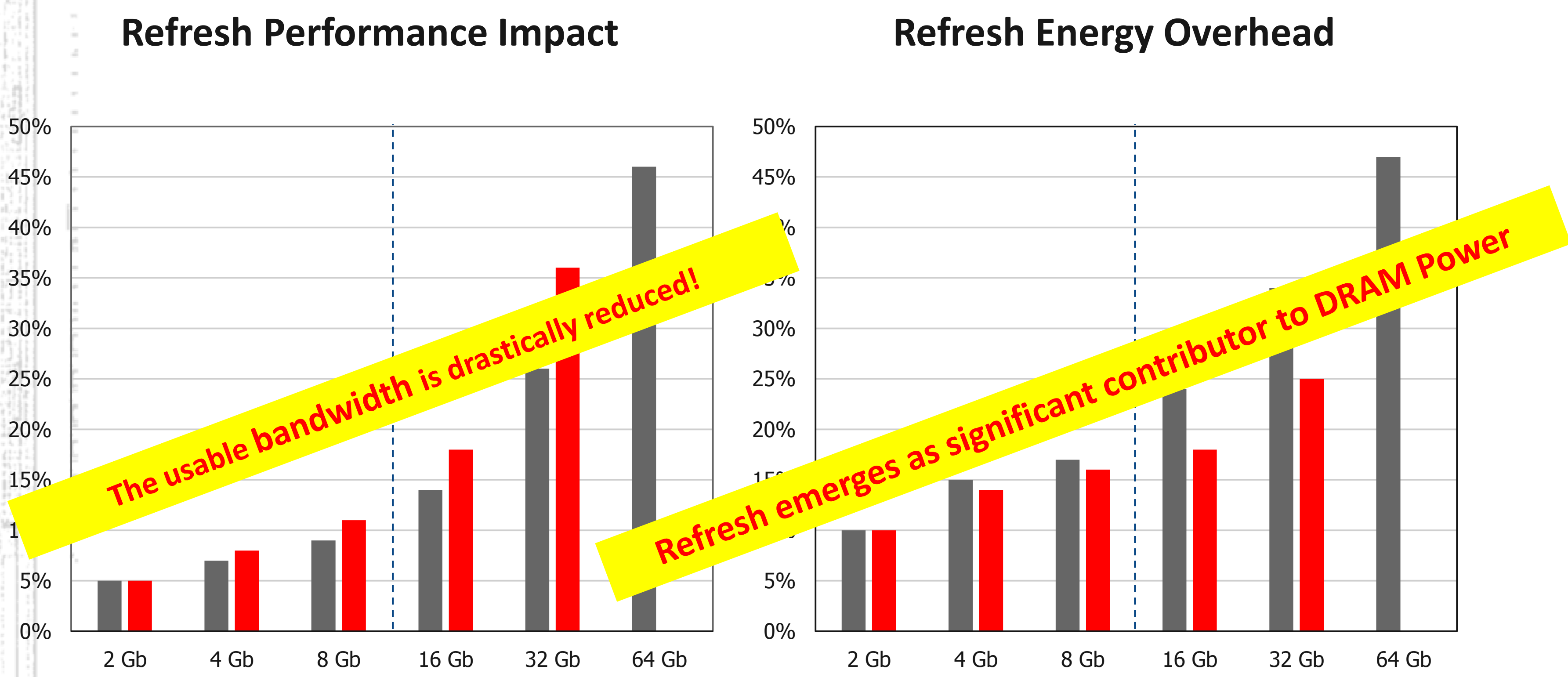
- DRAM Power Breakdown for Twitter Memcached Application*
- 2GB LPDDR3





A High-Level DRAM Timing, Power and Area Exploration Tool, O. Naji, A. Hansson, C. Weis, M. Jung, N. Wehn
IEEE International Conference on Embedded Computer Systems Architectures Modeling and Simulation (SAMOS), July 2015

Impact of Refresh for Future DRAMs

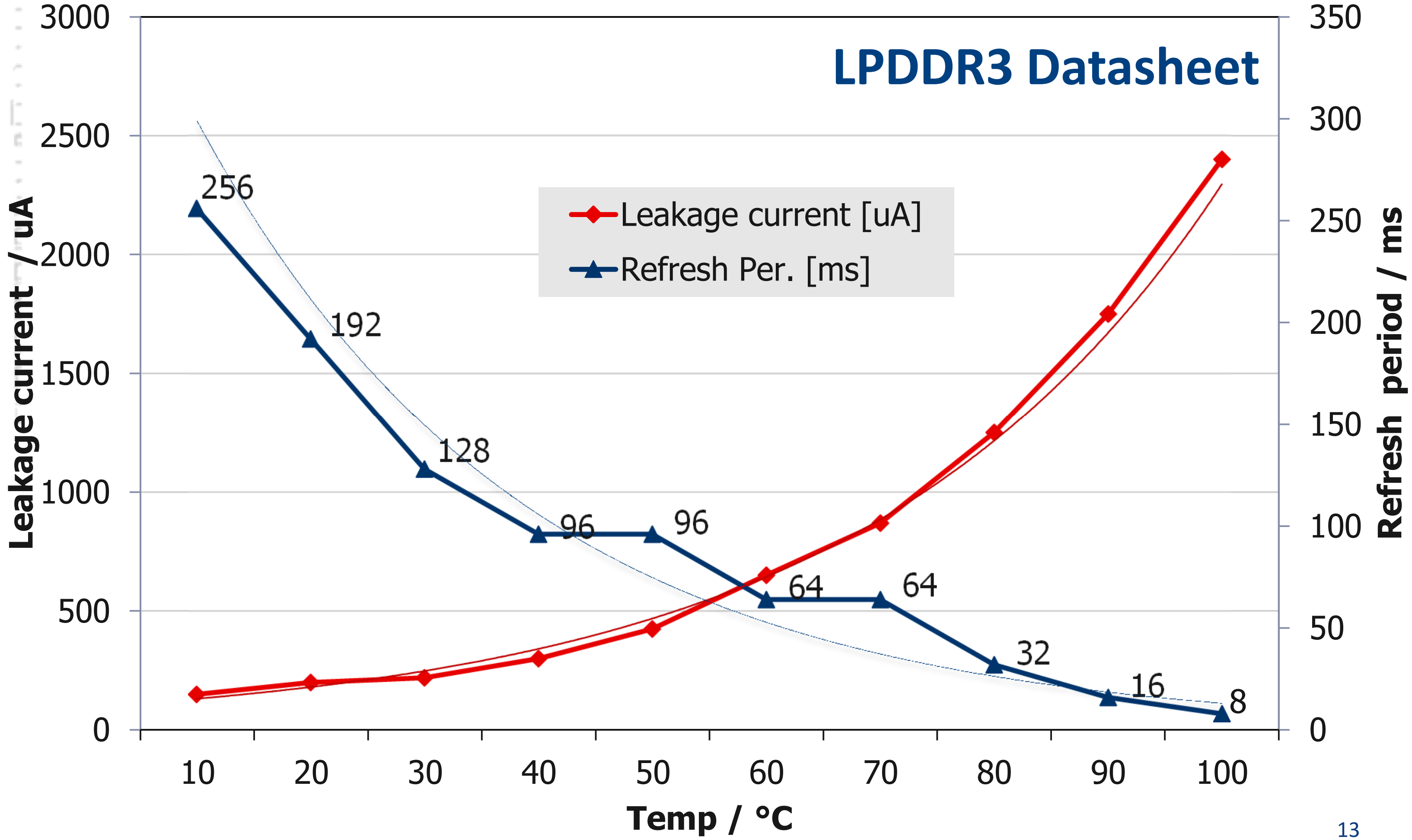


→ **High Temperatures Worsen The Behaviour**

- *J. Liu, et al. RAIDR: Retention-Aware Intelligent DRAM Refresh, ISCA 2012*
- *I. Bhati, et al. DRAM Refresh Mechanisms, Trade-offs and Penalties, IEEE Trans. 2015*

Refresh at High Temperatures

The exponential leakage current behavior must be counterbalanced by shorter refresh periods!



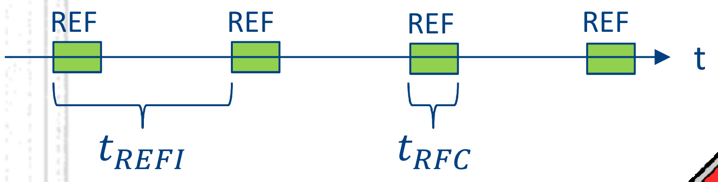
Refreshing WIDE I/O DRAM Stacks

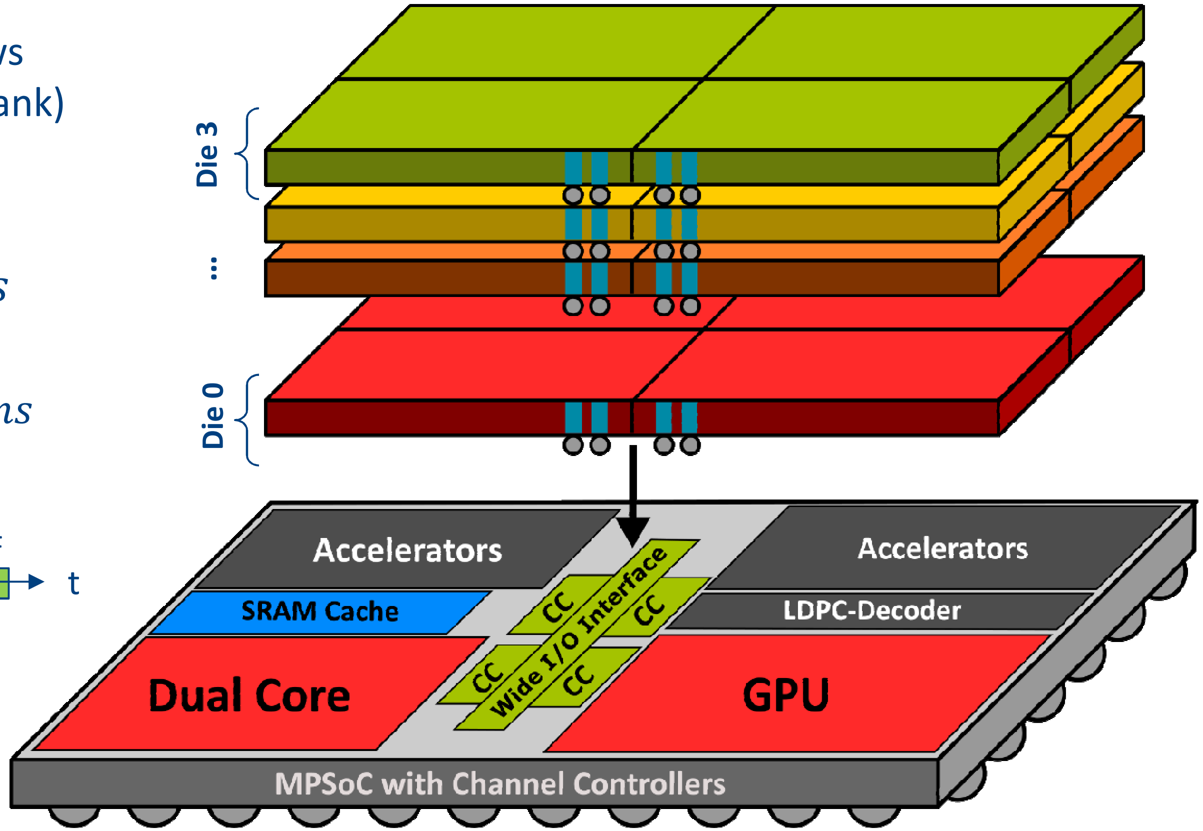
Worst Case Assumptions:

- Temperature = 100°C → t_{REF}= 8 ms
- Number of rows = 32768
- Bank parallel refresh (with 2 rows concurrently refreshed in one bank)
- Refresh command issued every:

$$t_{REFI} = \frac{8ms}{32768:2} = 488ns$$

- Refresh duration = $t_{RFC} = 130ns$

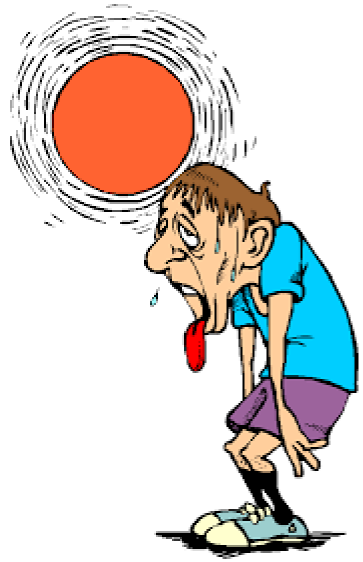




~25% of time spend in Refresh!

We can do better …

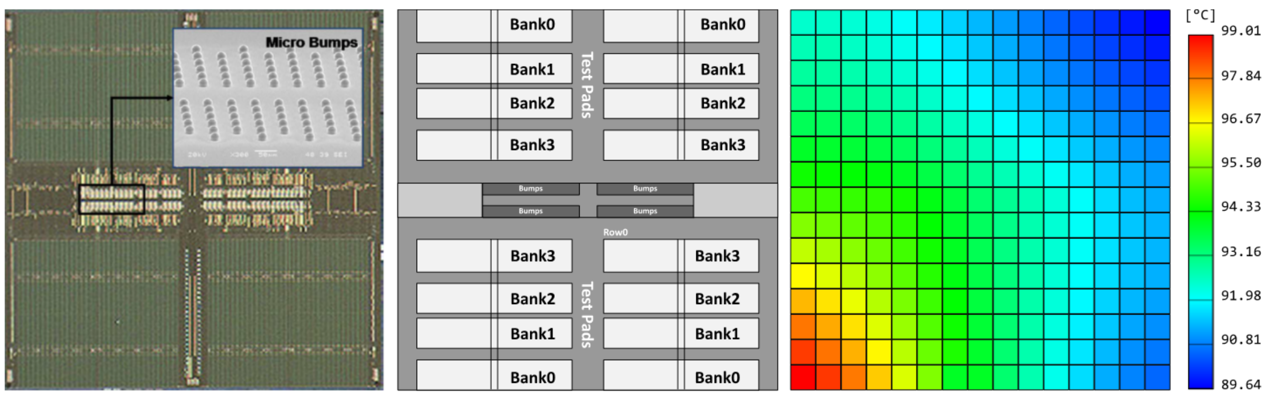
Response to the 1. problem: **DRAMs don't like heat**



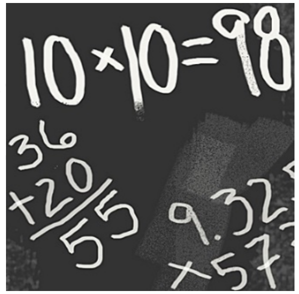
→ **Fine-granular refresh control**

&

→ **Approximate DRAM**



Approximate DRAM



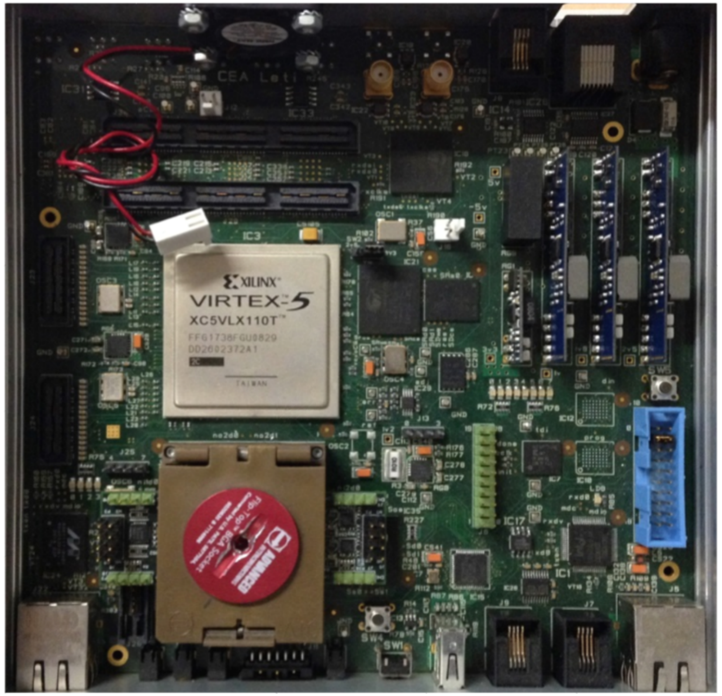
Reduce the number of Refreshes

- Lowering rate or completely switching off refresh
- Possible risk of data errors
- Example Case Studies:
  - *Flikker*[1]
    Lowers the refresh rate in a non-critical memory region
  - *Omitting Refresh*[2]
    Disables refresh completely for a specific memory region
  - …
- *Further Approaches:*
  - *RAPID, RAIDR, RIO, SECRET, ProactiveDRAM, AVATAR …*
  - *But: VRT, DPD, Temperature, Characterization Time, Storage …*
- Thorough analysis of retention errors mandatory

1: Song Liu, et al. 2011. Flikker: saving DRAM refresh-power through critical data partitioning.
2: Matthias Jung, et al. 2015. Omitting Refresh: A Case Study for Commodity and Wide I/O DRAMs.

Retention Error Analysis

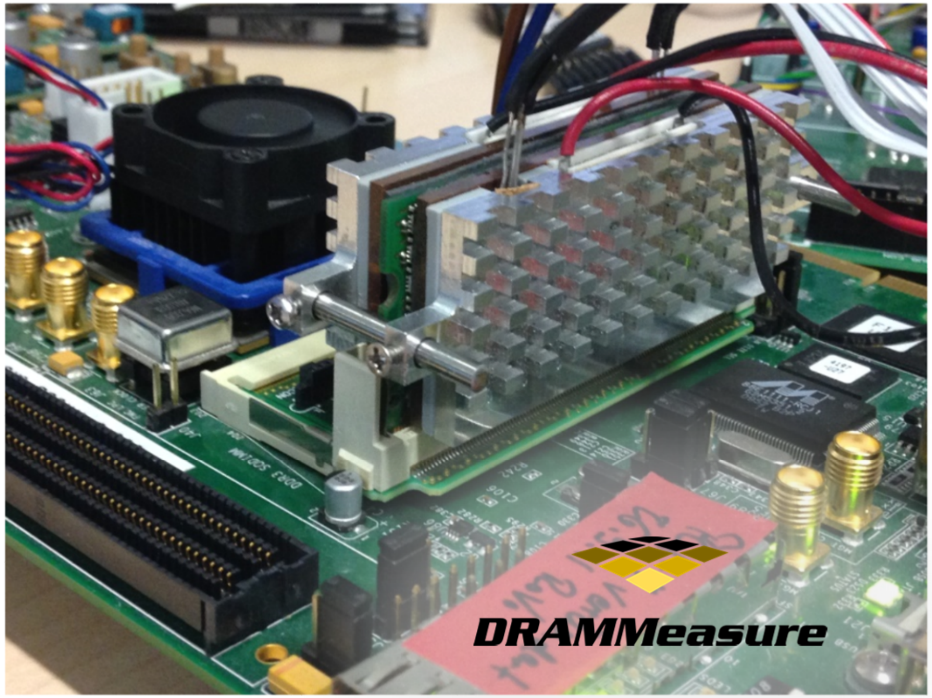
Wide I/O 3D-DRAM



WIOMING MPSoC:

- CMOS *65nm,* 72mm^2,1250 TSVs
- 4 Channels, 1Gb, 512 I/Os, *50nm*
- Heaters
- Temperature sensors

Commodity DDR3 DRAM

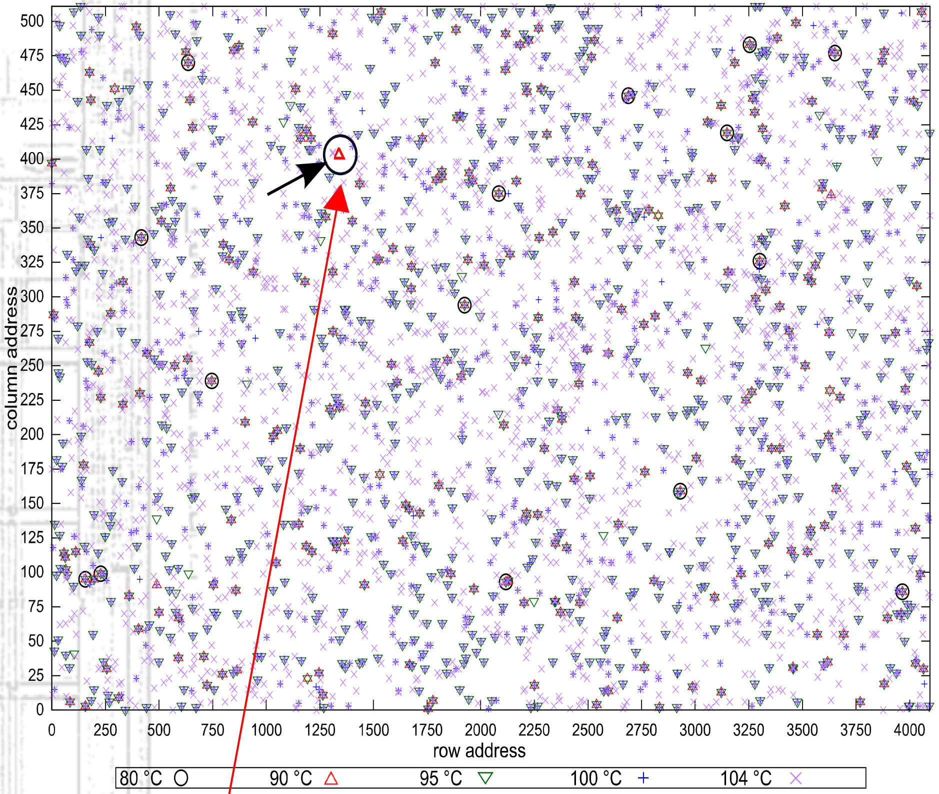


DRAMMeasure:

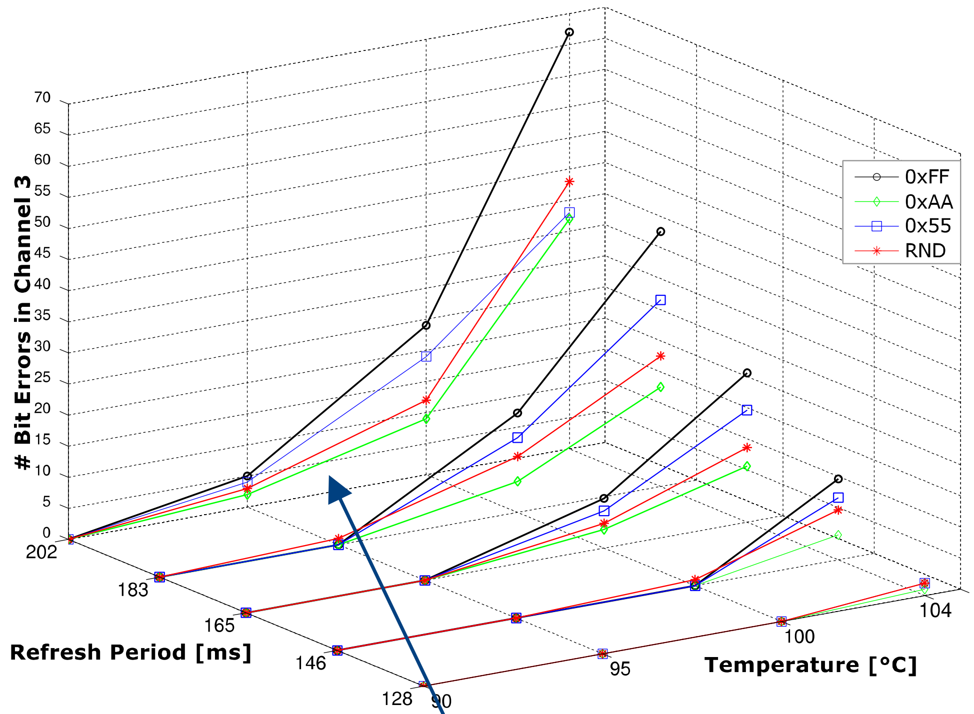
- Precise heating control of DDR3 SO-DIMMs
- Measuring currents and retention errors
- Applicable to any DDR3 SO-DIMM based platform (FPGAs, CPUs, …)

Wide I/O Retention Error Analysis

Observations : Variable Retention Times (VRT) & Data Pattern Dependency (DPD)

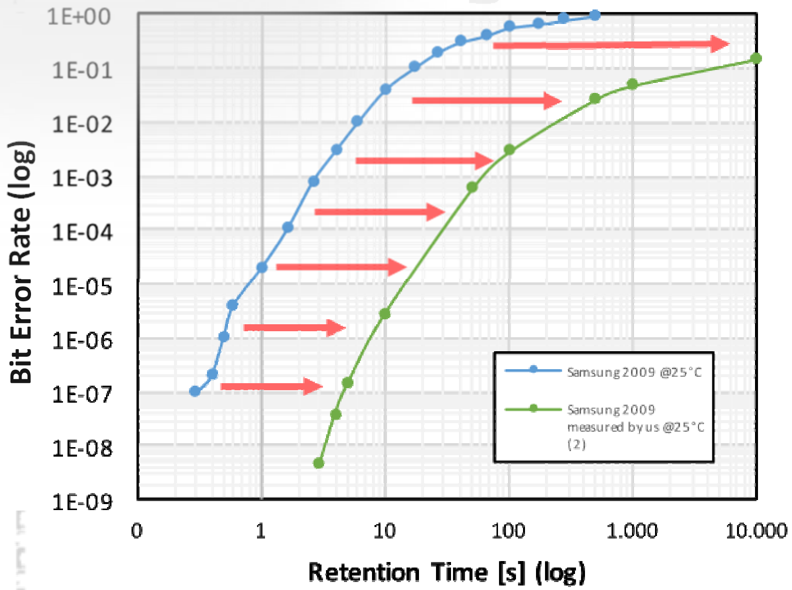


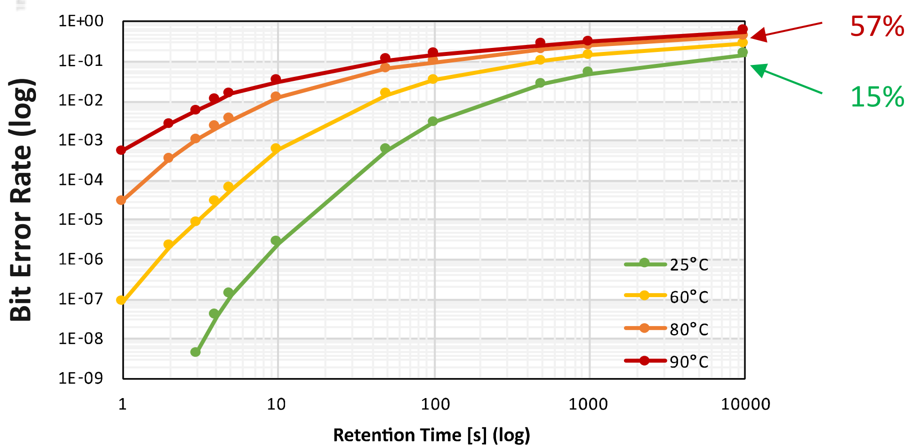
Unique bit error at 90°C



Different pattern cause different error rates!

Commodity DDR3 Measurements[1]





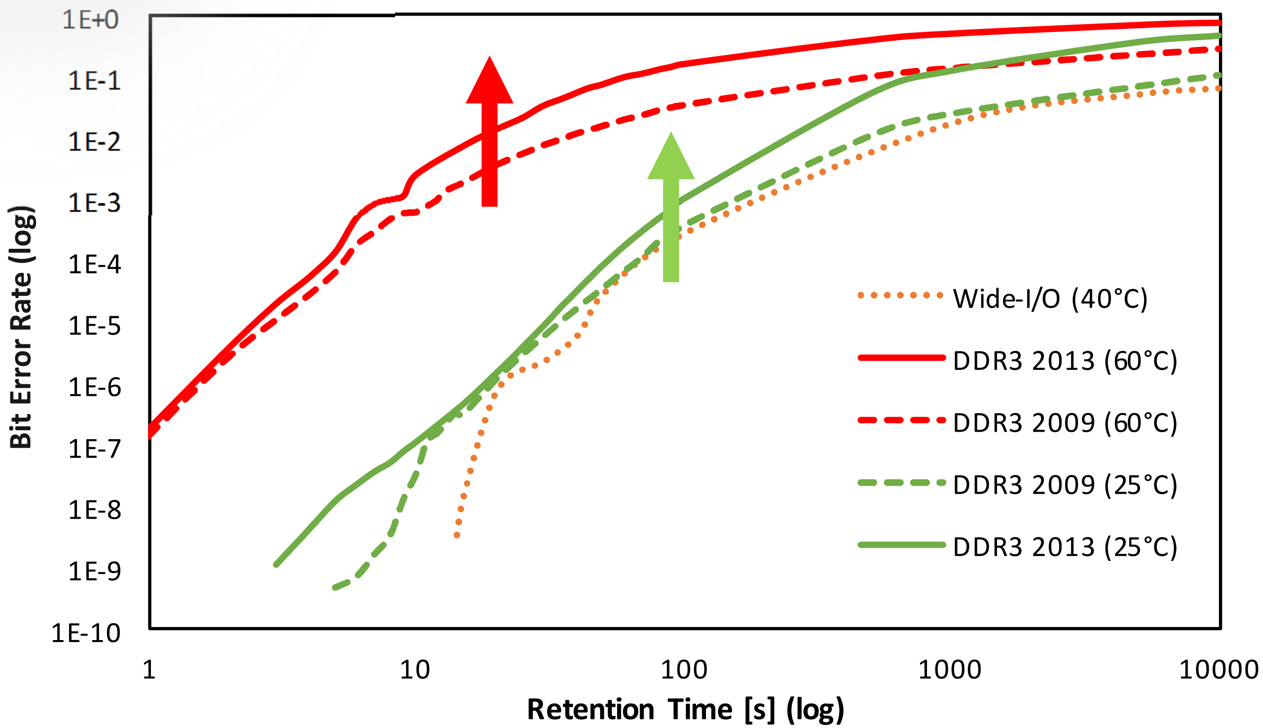
- Main reference in literature about retention errors published by Samsung[2]
- Measurements: 1-3 orders of magnitude better retention error behaviour
- DRAM can hold data much longer than specified, even at high temperatures.
- Can be exploited for Approximate Computing (DRAM)

[1] Values normalized to total DDR3 DRAM size: 512 MiB (Total number of DRAM cells: 4294967296)

[2] *Kim and Lee, A New Investigation of Data Retention Time in Truly Nanoscaled DRAMs, 2009*

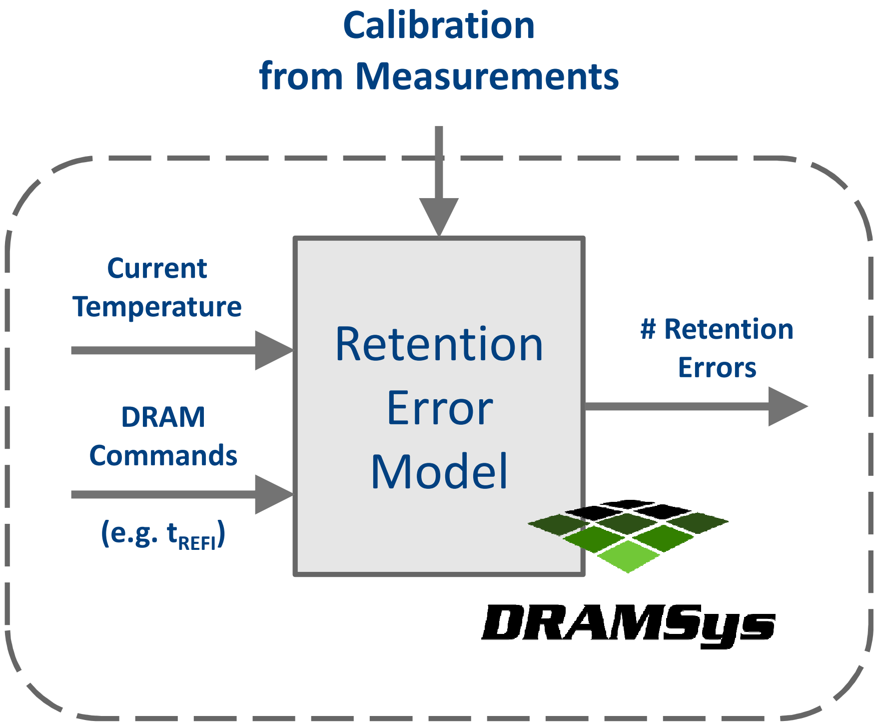
DRAMMeasure

Commodity DDR3 Scaling Trends



- A DRAM from 2009 (50nm) is compared with DRAM from 2013 (30nm)
- Scaling down DRAMs results in more errors
- We observe bends in the curves between 10 and 100 s

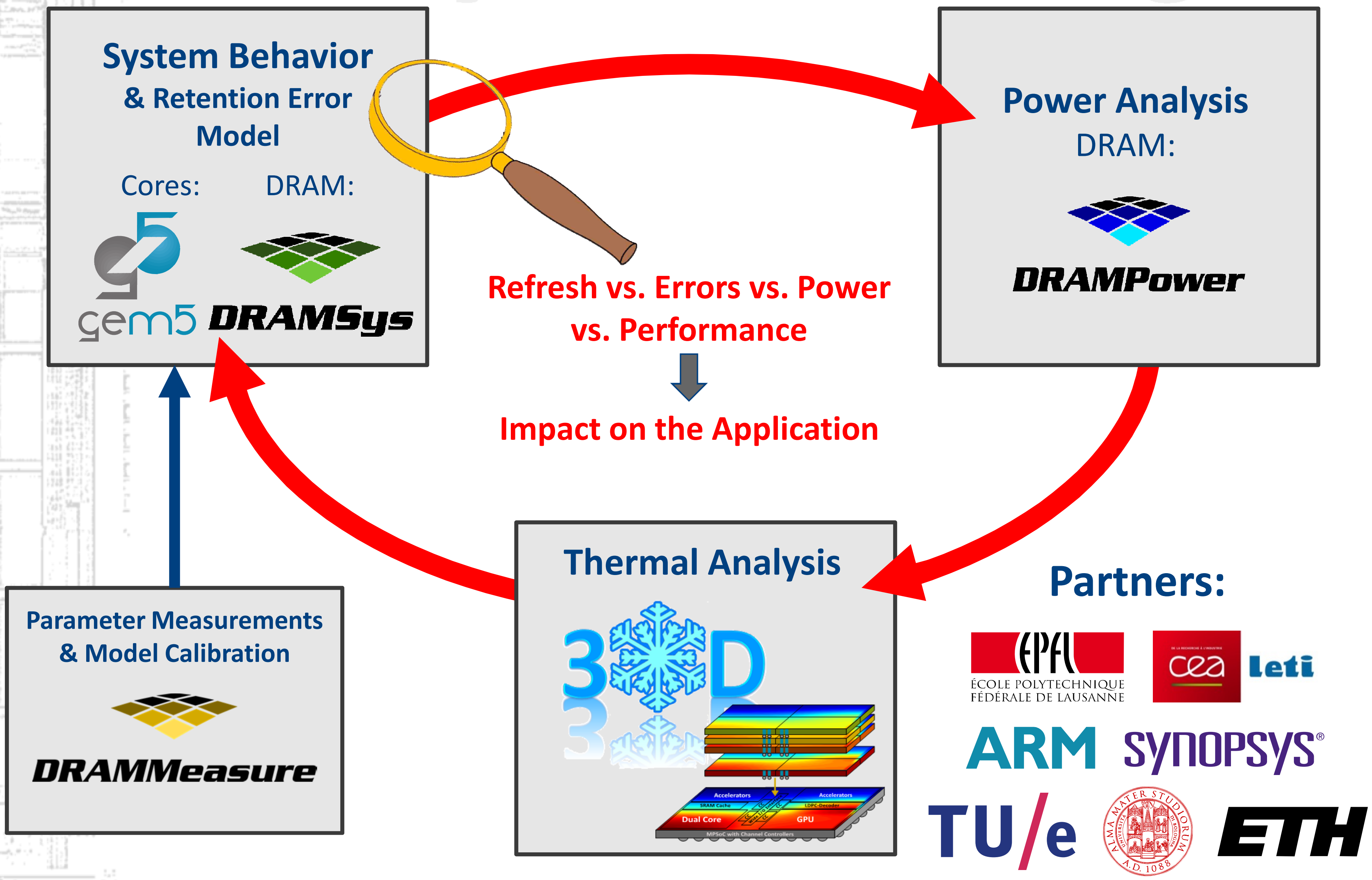
DRAM Retention Error Model



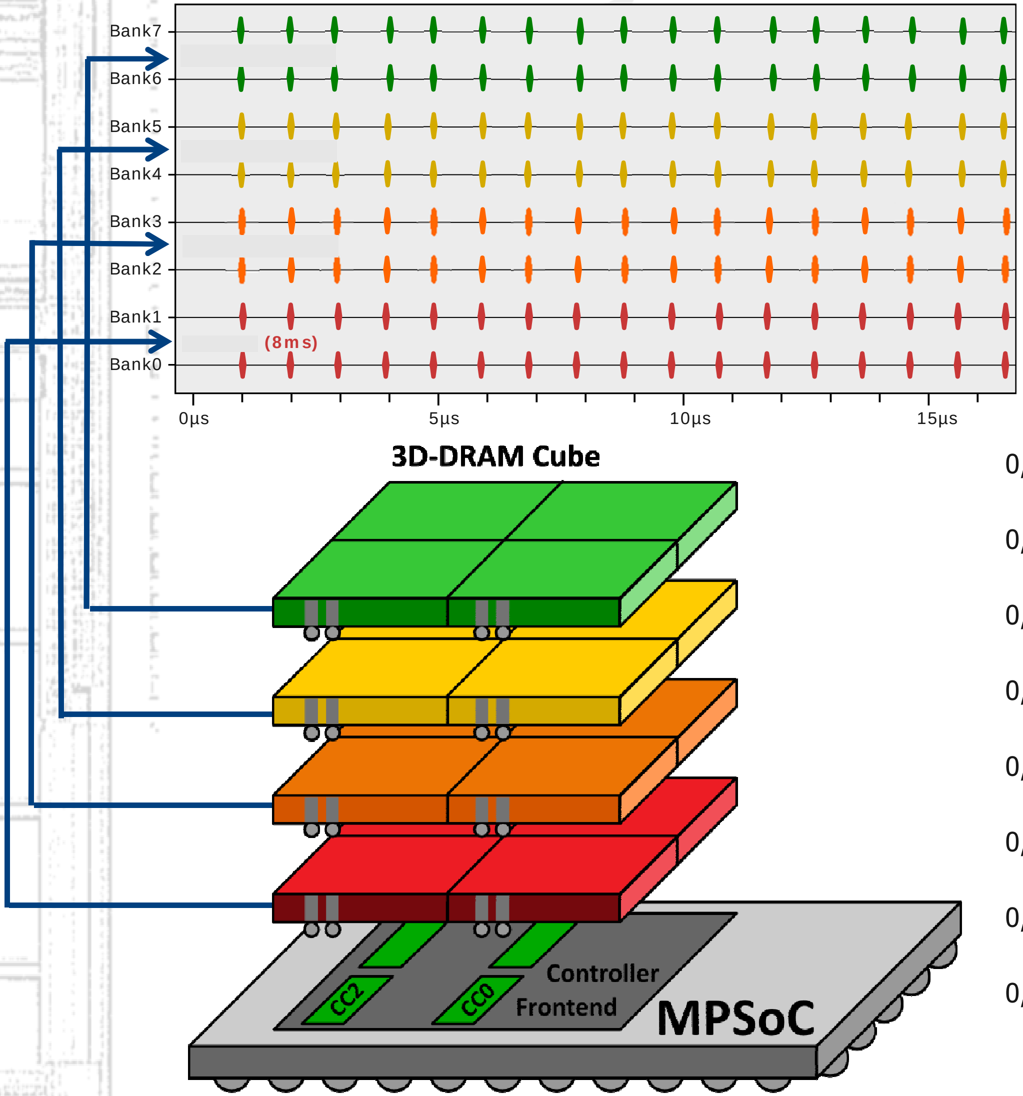
- Data Pattern Dependency (DPD)
- Variable Retention Times (VRT)
- Wide I/O and DDR3 DRAM
- Can be used in any C++ Simulator (e.g. gem5)

C. Weis, et al. Retention Time Measurements and Modelling of Bit Error Rates of WIDE-I/O DRAM in MPSoCs, DATE, 2015

Approximate DRAM Simulation Framework

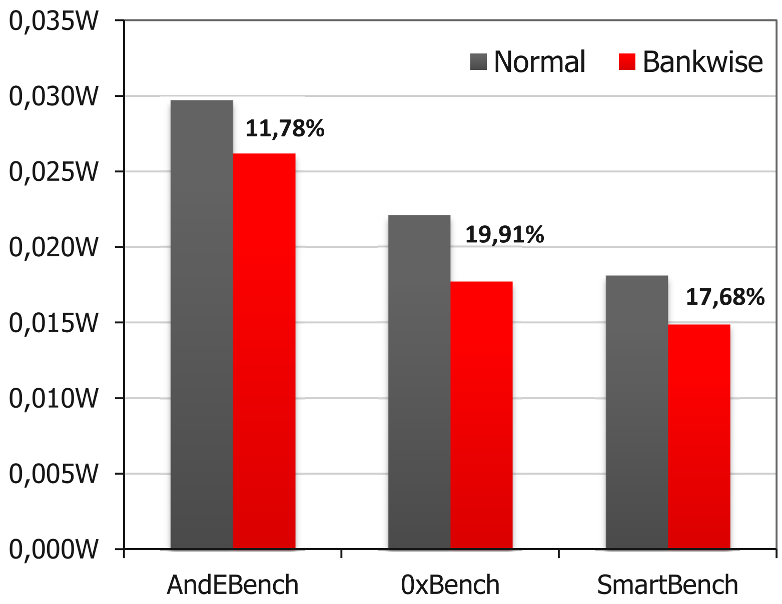


Temperature Variation Aware Bank-Wise Refresh

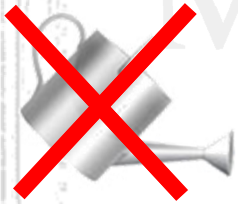


Different refresh rates on different dies (bank groups), according to the temperature of the die

Each bank was equipped with a TS

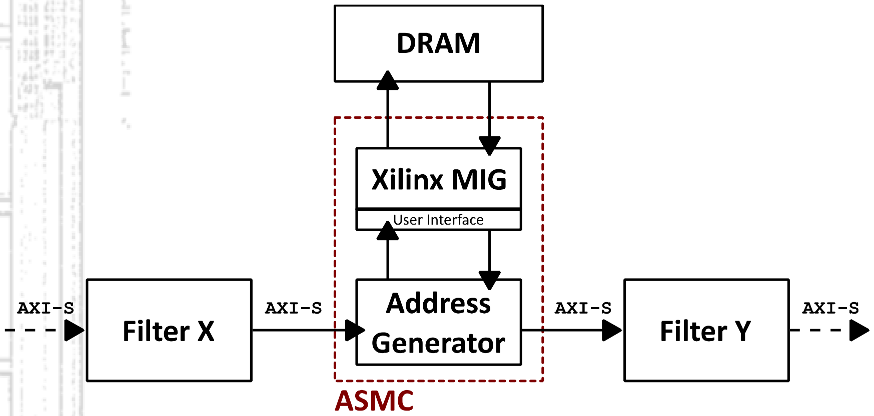


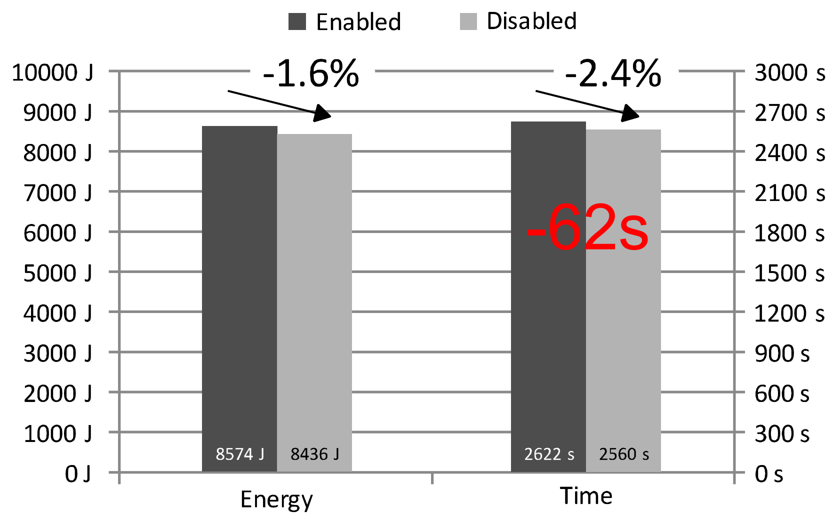
M. Sadri, et al. Energy Optimization in 3D MPSoCs with Wide-I/O DRAM Using Temperature Variation Aware Bank-wise Refresh, DATE 2014



Switch off Refresh: Image Processing

- Streamed image processing on Xilinx FPGA
- DDR3 SO-DIMM
- Application Specific Memory Controller (ASMC)
- Frame deadline = 9ms < t_{REF} = 64ms @ 25*C
- Refresh disabled in the memory controller
- No retention errors occur





600,000 Frames

A Per Layer Refresh Policy for 3D DRAMs

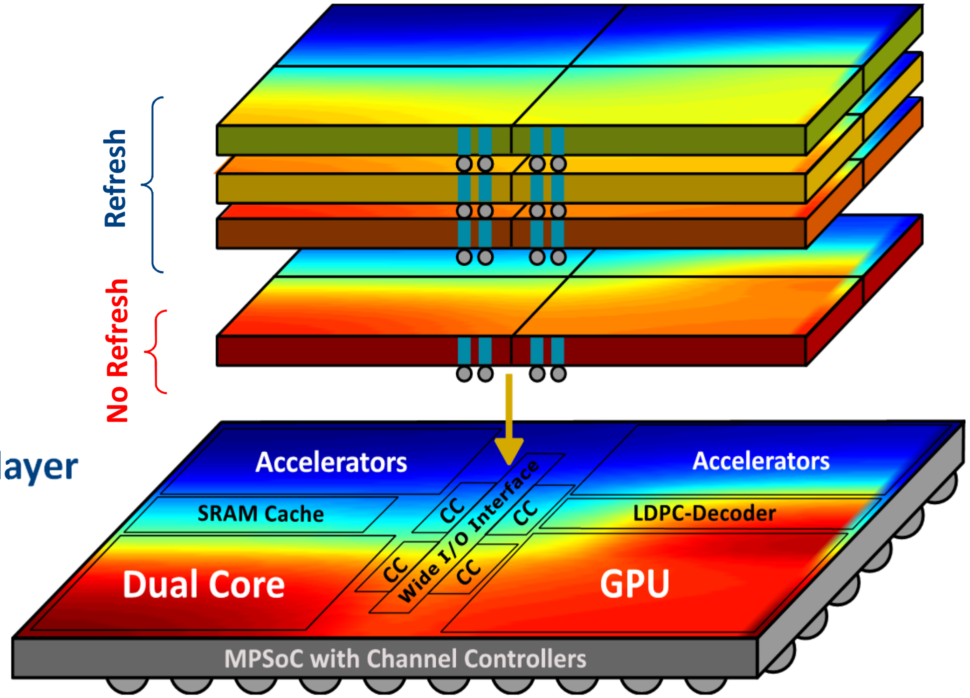
Separation of 3D DRAM Stack into **unreliable** and **reliable** regions

- Reliable regions: higher DRAM layers with temperature aware refresh
- Unreliable region: bottom DRAM layer with disabled refresh → **Omit Refresh (OR)**
- Access unreliable region while reliable region is refreshed

Example applications

- Graph processing
- Image processing
- Baseband processing

→ Saves 100% refresh power in the unr.-layer
→ Increases bandwidth

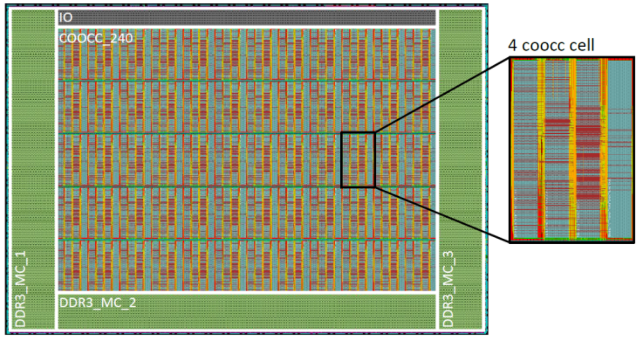


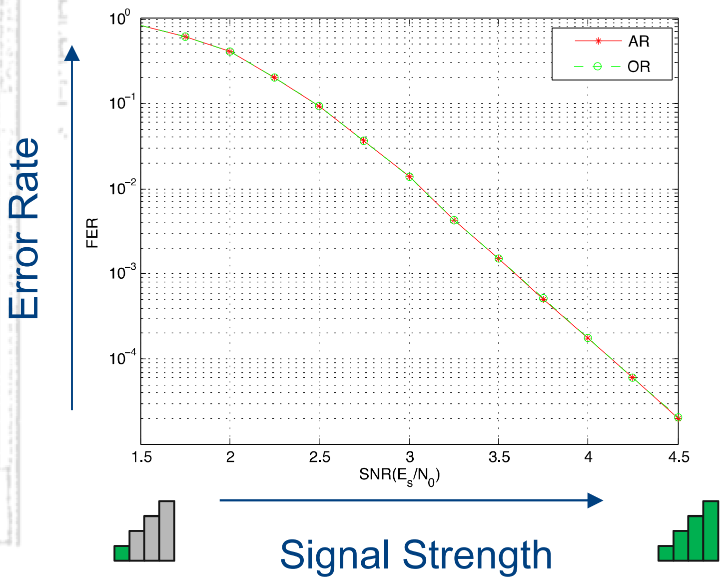
Simulation Results

Matthias Jung, et al. 2015. Omitting Refresh: A Case Study for Commodity and Wide I/O DRAMs.

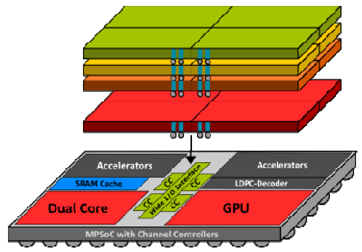
Example Applications

28 nm ASIC, 400 MHz, 51 mm²





Recommendation Systems:



- Netflix Dataset Graph:
  - 100,480,507 User Ratings
  - 480,189 Users
  - 17,700 Movies

- Graph is stored as matrix in unreliable region (sparse)
- Worst Case Assumptions: 90°C (actually required t_{REF}=16ms)

→ No noticeable loss in quality of recommendations

Baseband Processing:

- Simulation of Low-Density Parity Check Coding (LDPC)
- Channel data is stored in unreliable region Worst case assumptions: 100°C (actually required t_{REF}=8ms)
- Influence of retention errors much smaller than channel errors during transmission

→ No noticeable loss in communications performance

We can do better …

Response to the 2. problem: **How to get the huge bandwidth out of the device?**

→ More clever usage

&

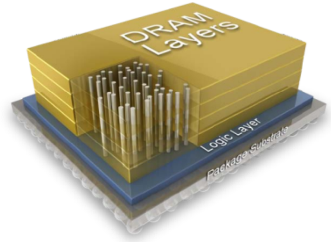
→ Maybe not needed at all

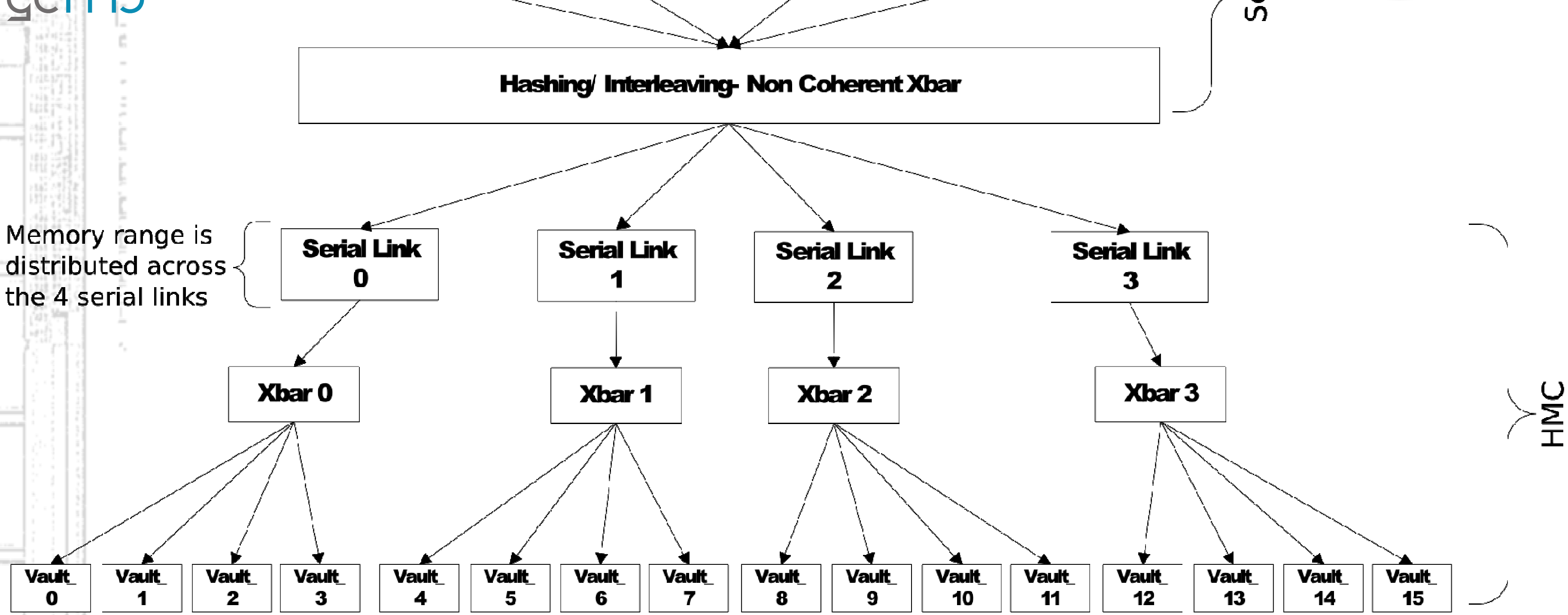
Is HMC a solution?

ARM

gem5

HMC Modeling (2nd Gen) in gem5:



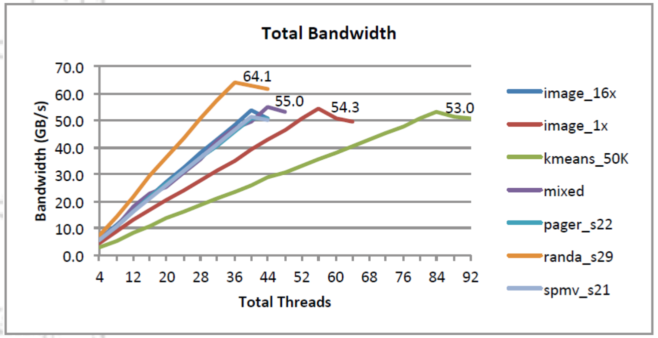


DRAMSpec: A High-Level DRAM Timing, Power and Area Exploration Tool (DOI)
C. Weis, A. Mutaal, O. Naji, M. Jung, A. Hansson, N. Wehn. *International Journal of Parallel Programming (IJPP)*, Springer, 2016.

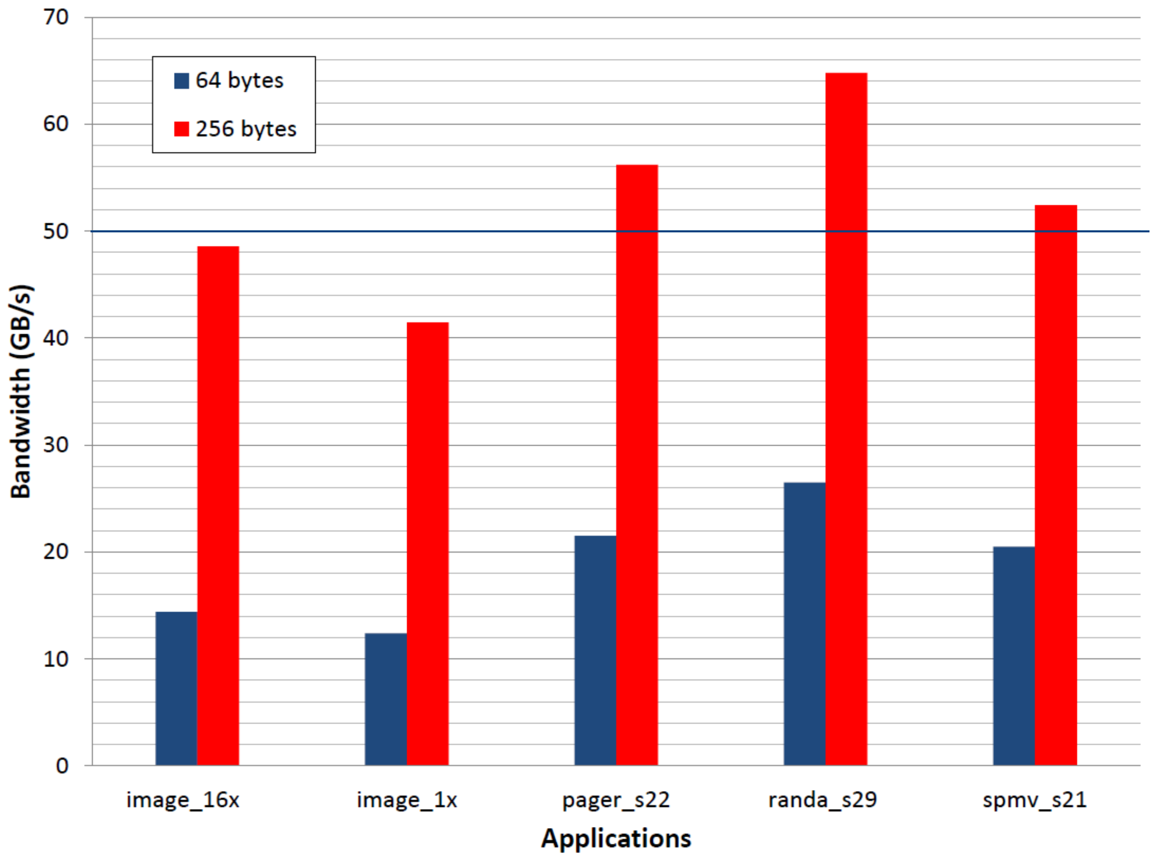
Applied bandwidth to the HMC

- Taken from M. Gokhale
- At the LLNL measured on a FPGA board the different response times of the HMC (round-trip ~24ns)

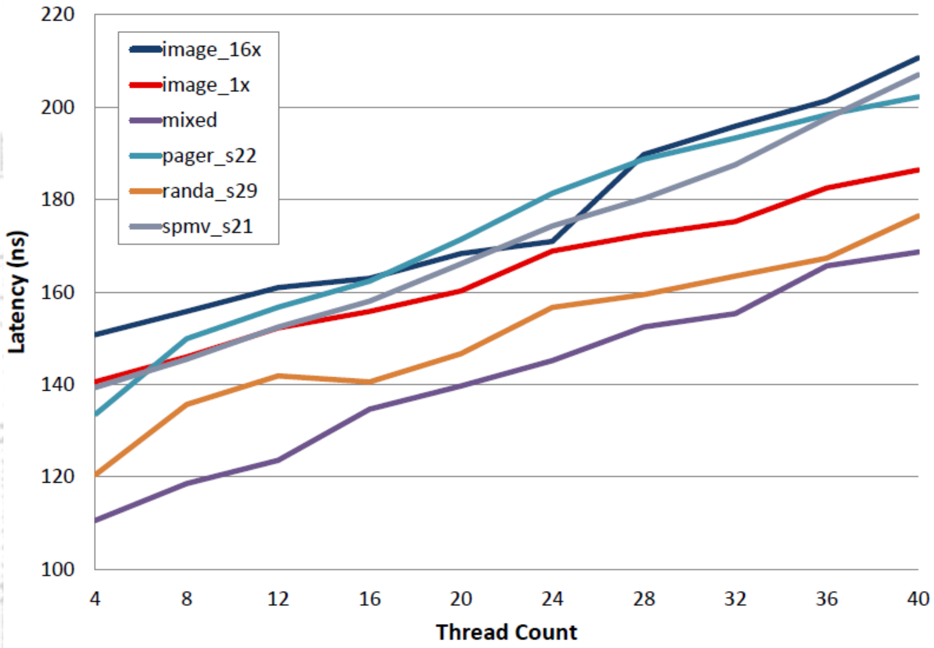
- Here we used 40 threads active with different data granularity (64 & 256B)
- BW was very similar to Mrs. Gokhale's results:

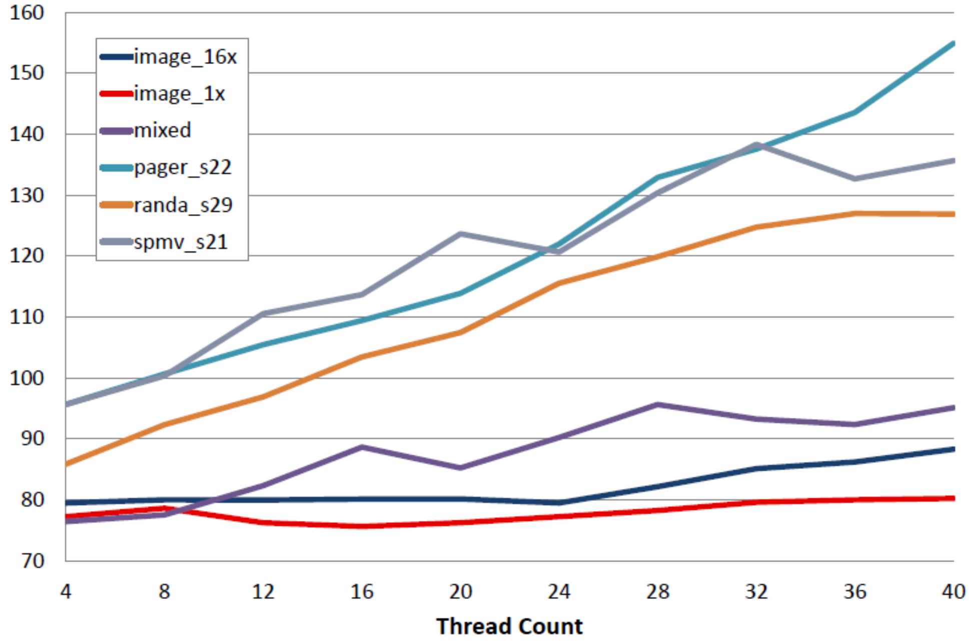


Workload	Short name	Description
Page Rank	pager_s22	A benchmark to rank web pages in popularity
Image Diff. (full)	image_x1	Pixel-wise diff-computation of two images (full)
Image Diff. (x16)	image_x16	Pixel-wise diff-computation of two images (x16 dec.)
Sparse Mat. Vec.	spmv_s21	Multiply a sparse matrix with a dense vector
Random Access	randa_s29	Read and updates random locations in a table
Mixed	mixed	A mix of all listed benchmarks



HMC Latency – not always predictable





Average access latency (a):
- 22nm DRAM HMC
- Page size = 256 Bytes and
- Packet size = 256 Bytes

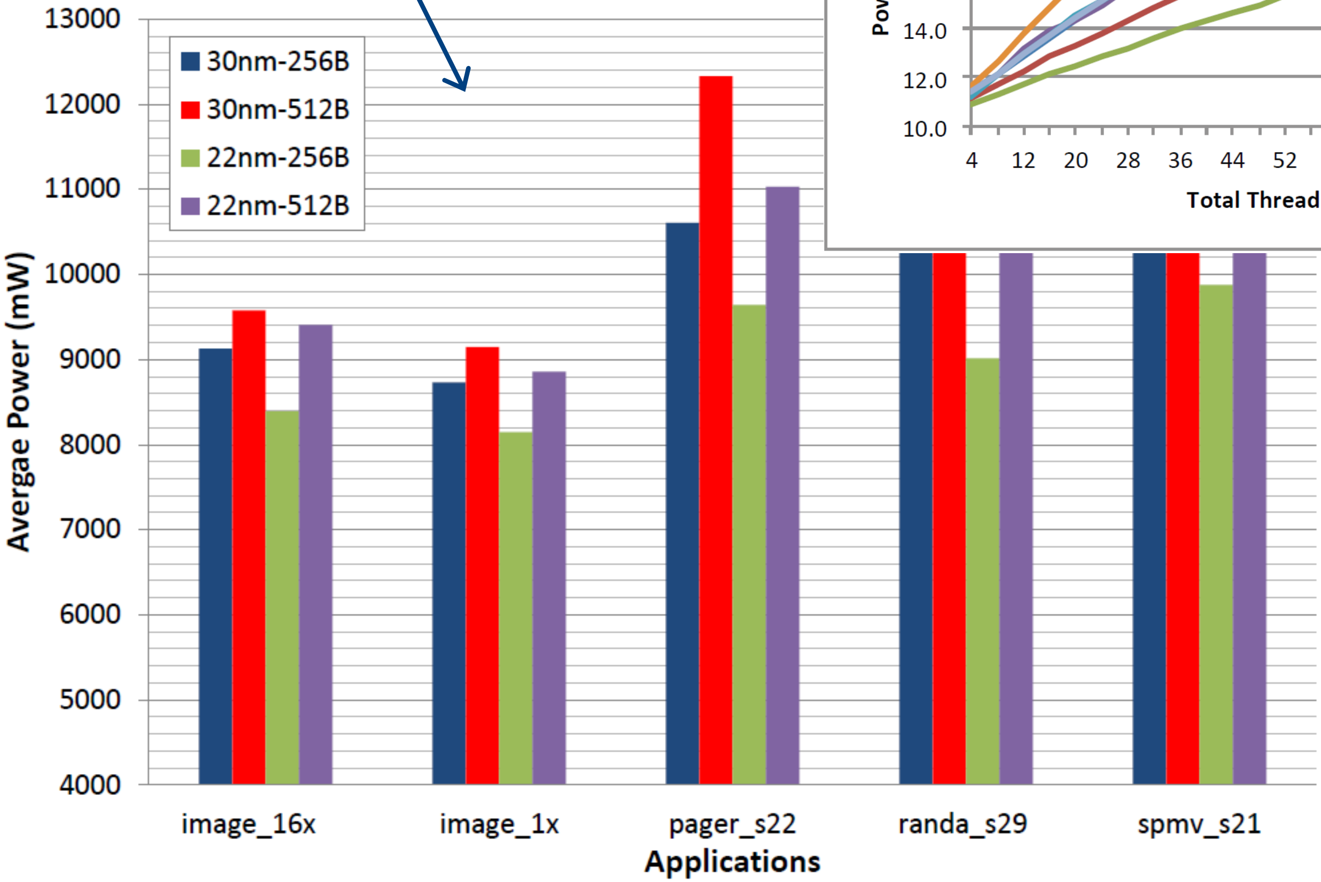
Average access latency (b):
- 22nm DRAM HMC
- Page size = 512 Bytes and
- Packet size = 64 Bytes

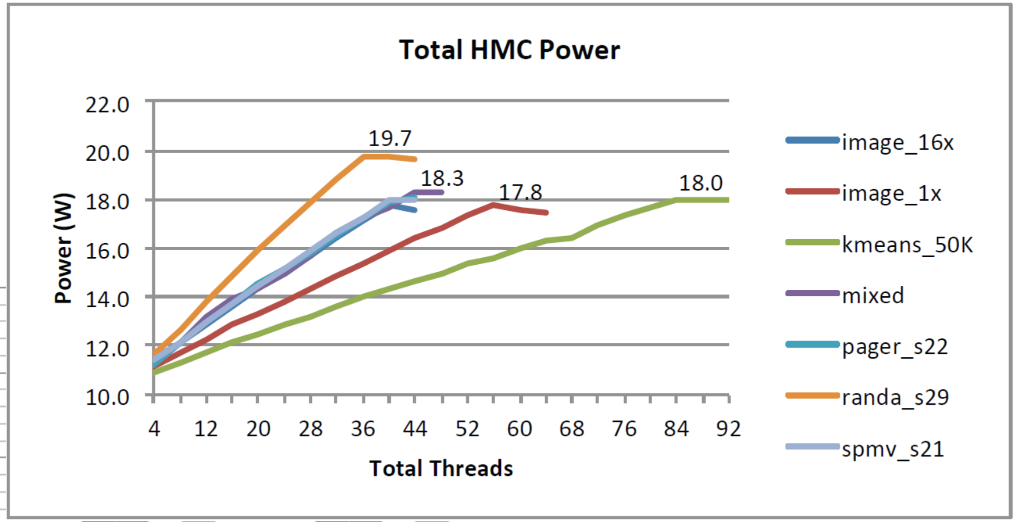
HMC Power – 11W ++

M. Gokhale et al., 2015

DRAM part only power:

for different page sizes and technologies





Link-power only is about 10-11W!

We can do better …

Response to the 3. problem: **Memory centric computing makes it worse…?**

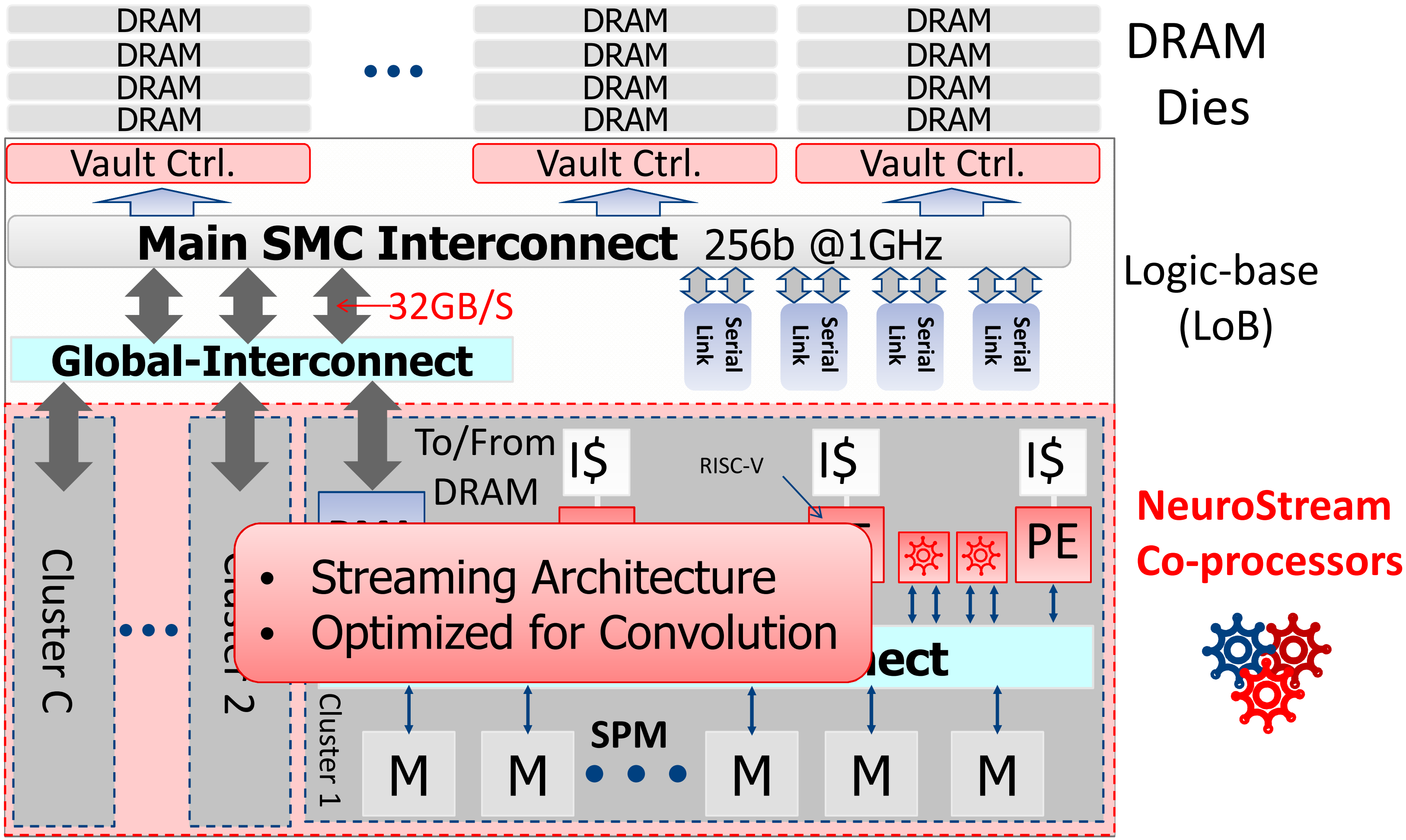
→ **New Architectures**

&

→ **Custom 3D-DRAMs**

The Smart Memory Cube (SMC)

Used for CNNs



~22 GFlops/W

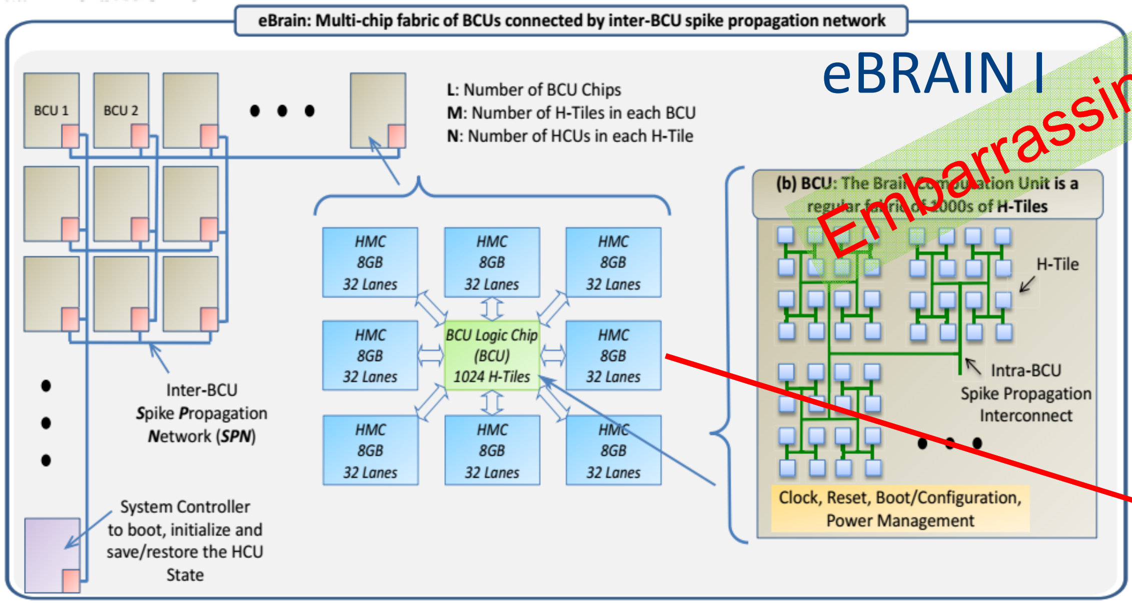
RISC-V Processors

Neurostream: Scalable and Energy Efficient Deep Learning with Smart Memory Cubes
Erfan Azarkhish et al., 2017

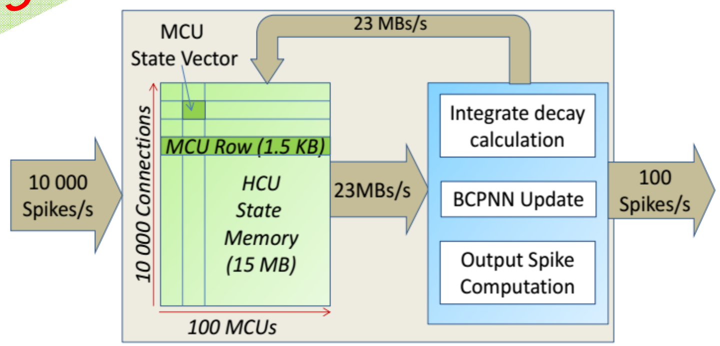
Custom 3D-DRAM for eBRAIN II

- A custom multi-chip design to simulate the human brain in real time using the spiking BCPNN (Bayesian Confidence Neural Network)
- The architecture for this algorithm is based on Hyper Columns Units (HCU) and Mini Columns units (MCU)
- The parallel computability of HCUs and MCUs makes this architecture hardware friendly
- Each HCU is an aggregation of 100 MCUs
- The hyper column unit has 10000 input connections and 100 output connections

Embarrassingly Parallel!



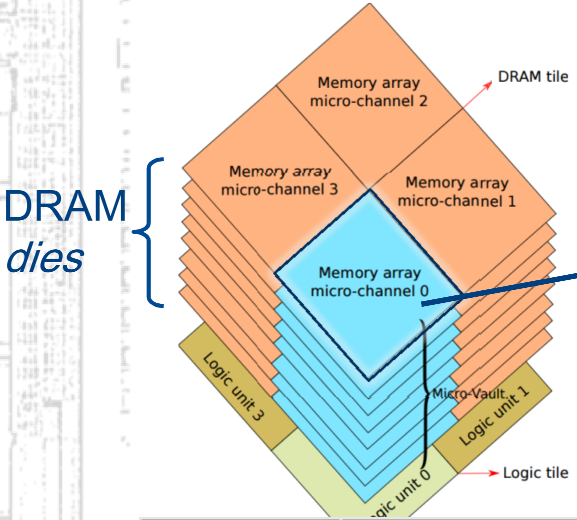
HCU updates:
Row-wise
Col-wise

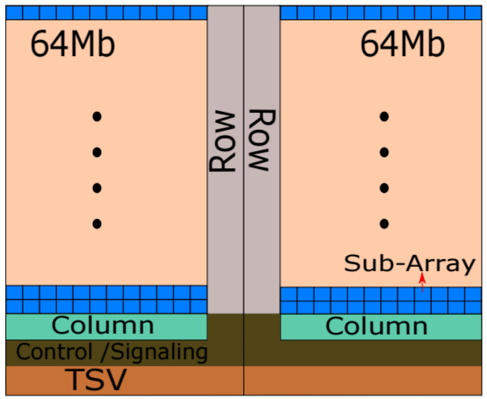


HMCs consume **40KW**

Custom 3D-DRAM for eBRAIN II

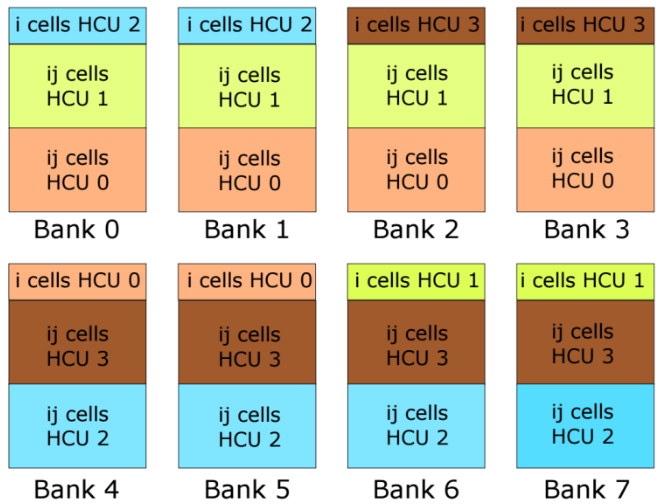
- Custom-optimized **3D-DRAM architecture** => 48 I/O DDR microChannel per HCU (1 – 2 mm^2 depending on the DRAM tech.) with 500MHz freq.
- Tailored **access** → using a technique called "**Row merge**", where we balanced the BW between Row-updates and Col-updates (from the HCUs).



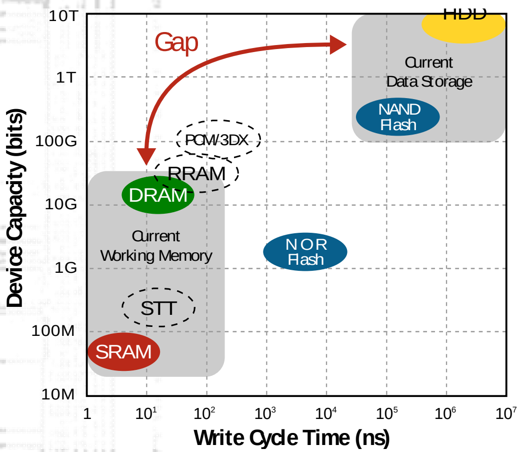


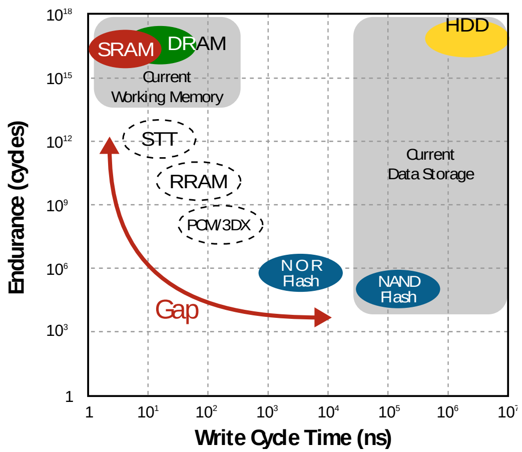
Species	# of HCUs	Average Power
Mouse	1.6×10^3	13 W
Rat	5.0×10^3	44 W
Cat	6.0×10^4	522 W
Macaque	2.0×10^5	1700 W
Human	2.0×10^6	**17 KW**

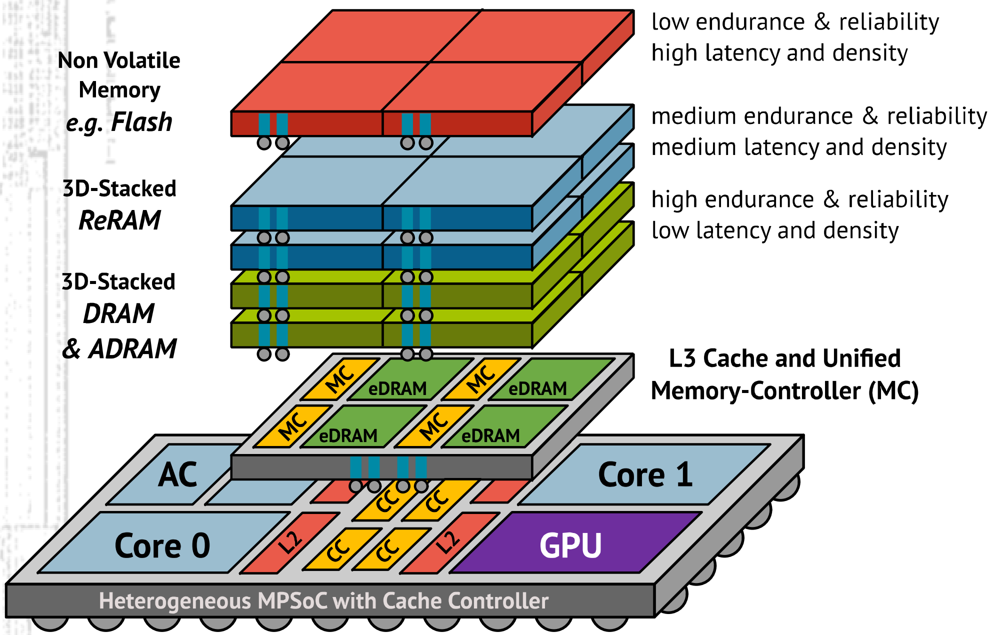
Matrix – Bank mapping of 4 HCUs:
→ *optimized data layout*



The Future is Heterogeneous





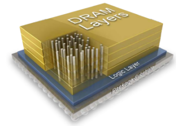


- New memory technologies:
  - PCM
  - 3DXPoint
  - STT-MRAM
  - RRAM
- DRAM **won't** be dead, but will change its role → maybe used as **Cache** …
- New memory **ECC** techniques
- Heterogeneous main memory systems:
  - **NVDIMM-P**
  - 3D MPSoCs / **3D Memory Stacks**
- New requirements on:
  - Compiler
  - OS
- Processing in memory (**PIM**)

Summary – Take-away messages

- Approximate DRAM can be used to trade-off BW vs. reliability
  - Fine-granular refresh control in 3D DRAM stacks is required

- HMC is good for high concurrency and highly distributed threads
  - Latency (contentions on the vault accesses) & Power are large drawbacks



- HBM, highest BW possible – but cost of a 1000mm^2 Si interposer

- Custom 3D-DRAMs have a large potential
- Hybrid architectures and Near/In-memory processing (e.g. NeuroStream or uPmem's processor) will be key

Thank you for Listening

For more information //ems.eit.uni-kl.de