

# Many-core GPUs: Achievements and perspectives

---

Manuel Ujaldón

Computer Architecture Department  
University of Malaga (Spain)

# Talk contents [30 slides]

---

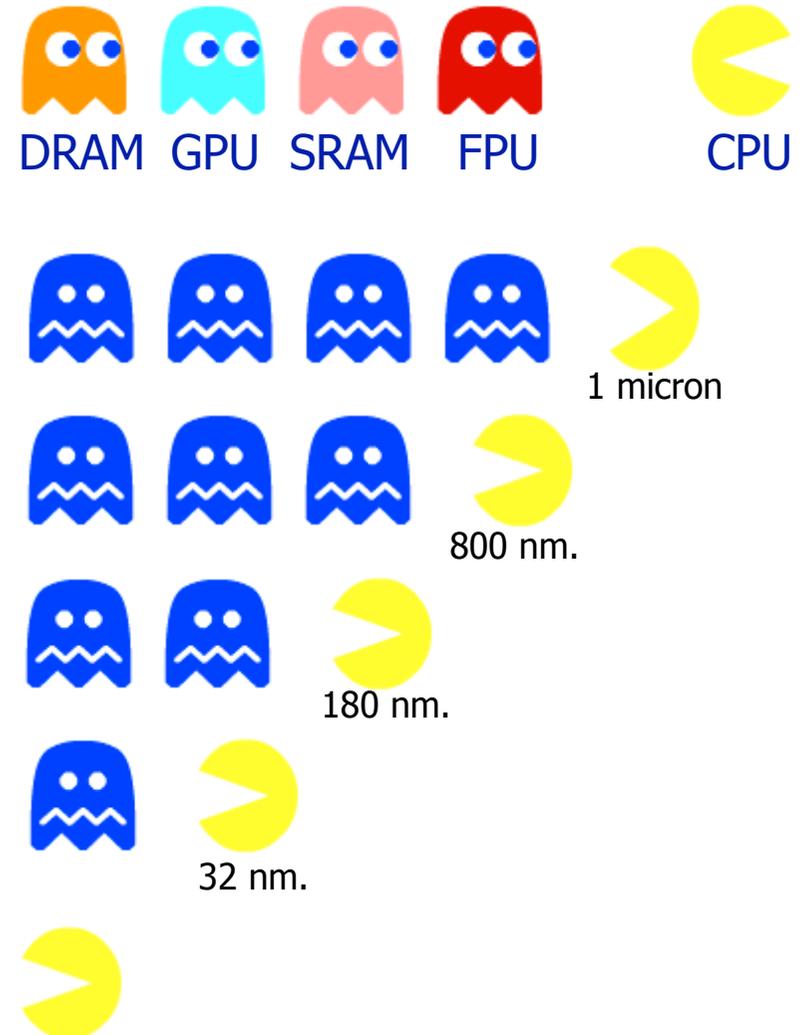
1. Evolution and achievements. [4]
2. SIMD execution and the warp size [8]
3. The future: Many-cores with stacked-DRAM. [10]
4. 3D DRAM gains versus 2D DRAM technology. [4]
5. Impact on GPUs. [3]
6. Conclusions.

# I. Evolution and achievements

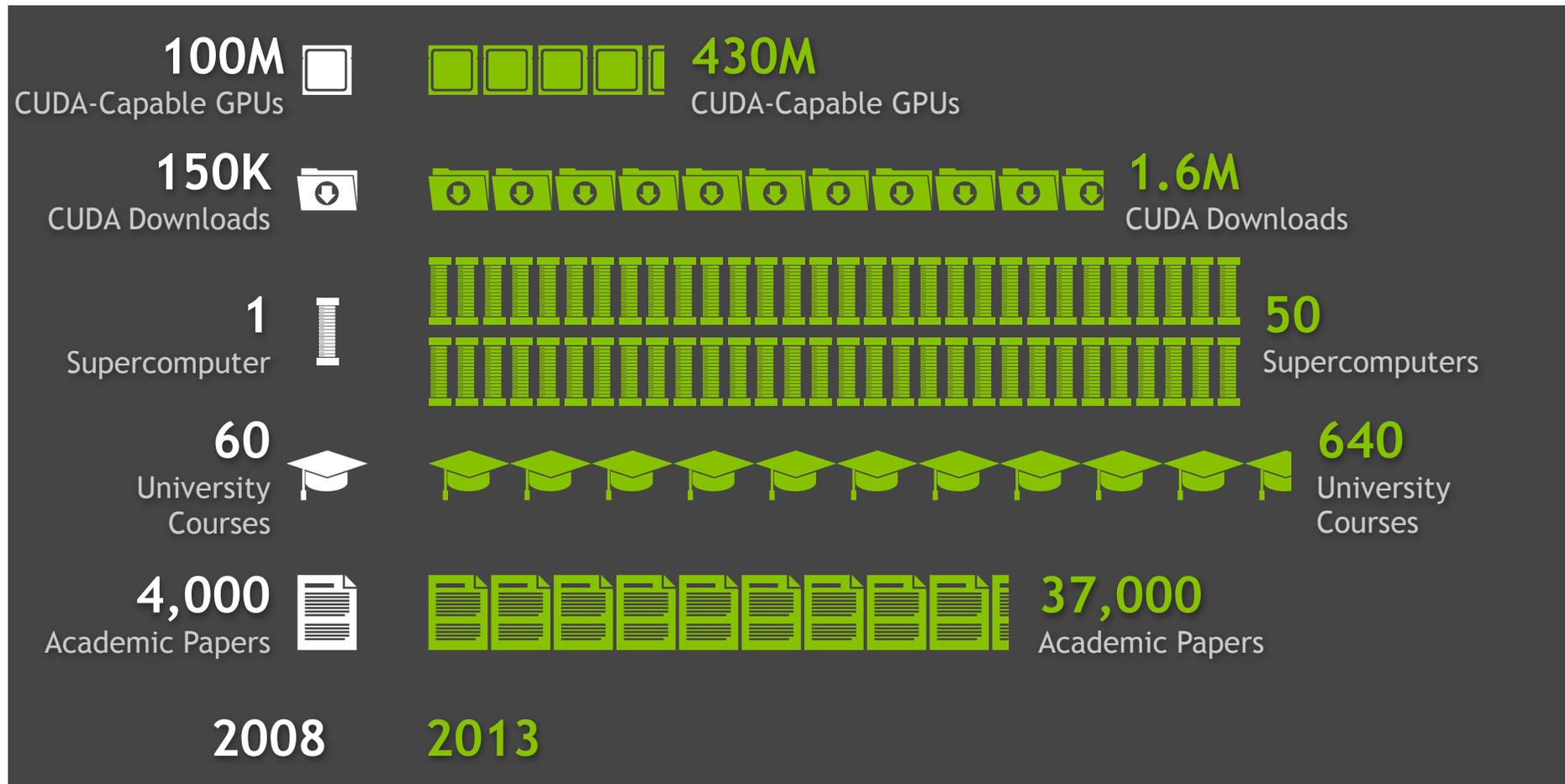


# Evolution over the last 30 years

- Initially, commodity PCs were decentralized systems.
- As chip manufacturing process shrank to less than a micron, they started to be integrated *on-die*:
  - 1989: FPU [Intel 80486DX].
  - 1999: SRAM [Intel Pentium III].
  - 2009: GPU [AMD Fusion].
  - 2016: DRAM [Nvidia Volta].
- The end of the story is SoC (System-on-Chip).



# GPU achievements: CUDA



There is a CUDA software download every minute.

# Three features for the GPU to become a unique processor

---

## ● Simplified control.

- The hardware control for a thread is amortized on 31 other threads (warp size = 32). This feature defines the personality for the processor and its affinity with vector and superscalar architectures.

## ● Scalability.

- Take advantage of the huge data volume handled by applications, to define a sustainable parallelization model.

## ● Productivity.

- Lots of mechanisms are defined so that when a thread starts processing slow instructions, others **hide its latency** taking over resources **immediately**.

# Three reason for feeling attracted to GPUs

---

## ● Power

- Those days of requiring 200 W. are over. Now, GPUs contribute to supercomputers which you easily find on top positions of the Green 500 list. Progression:
  - Fermi (2010): 5-6 GFLOPS/w.
  - Kepler (2012): 15-17 GFLOPS/w.
  - Maxwell (2014): 40 GFLOPS/w. (preliminary estimations).

## ● Cost

- Low price due to a massive selling marketplace.
- Three GPUs are sold for each CPU, and the ratio keeps growing.

## ● Ubiquitous

- Everybody has already owned a bunch of GPUs.
- And anyway, you can purchase one almost everywhere.

## II. SIMD execution and the warp size

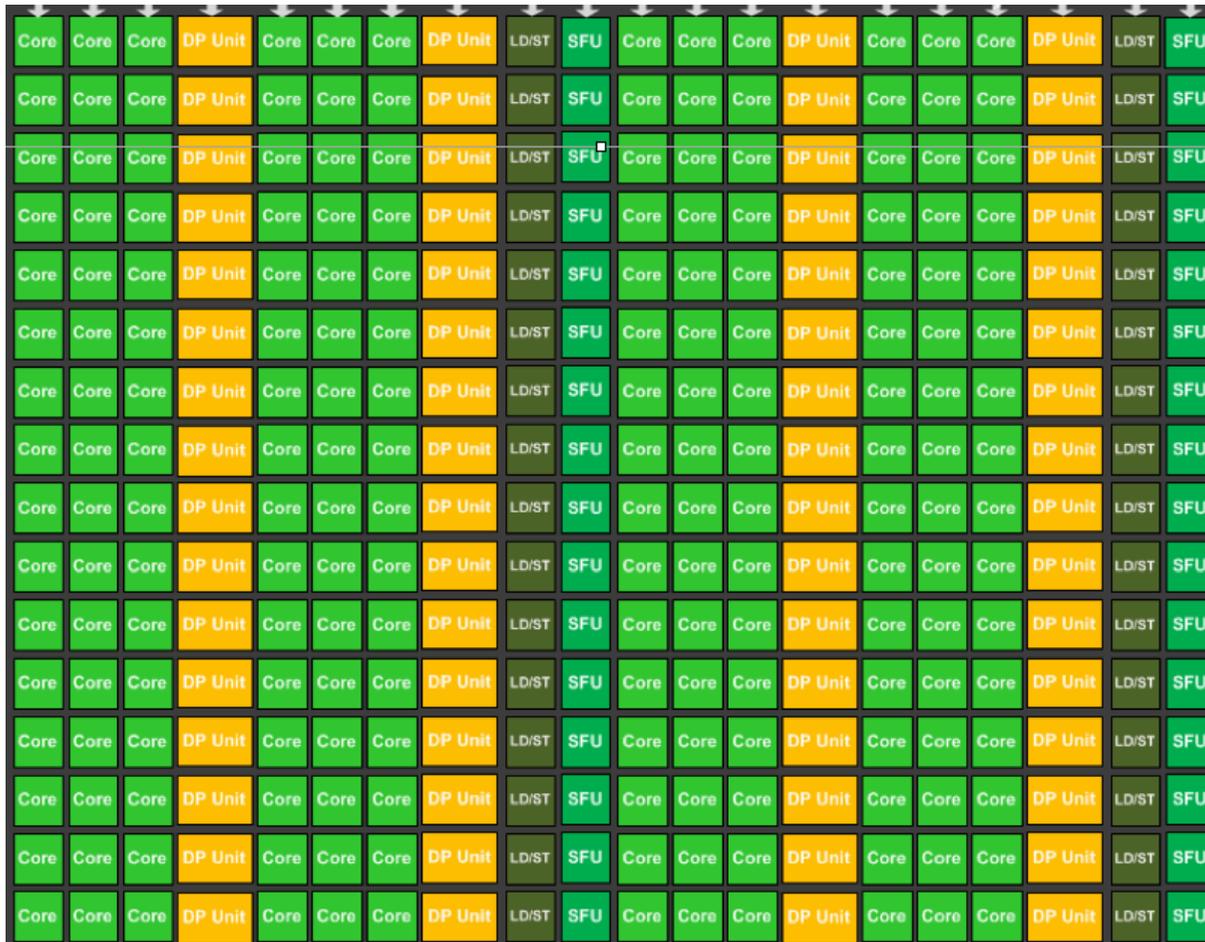


# Processing instructions on GPU

- The front-end dispatches and schedules instructions.
- The back-end executes instructions in parallel.
- Goal: Balance throughput on both sides. As the software is very diverse, we have to find a consensus in hardware.
- GPU multiprocessors contain different resources for this:

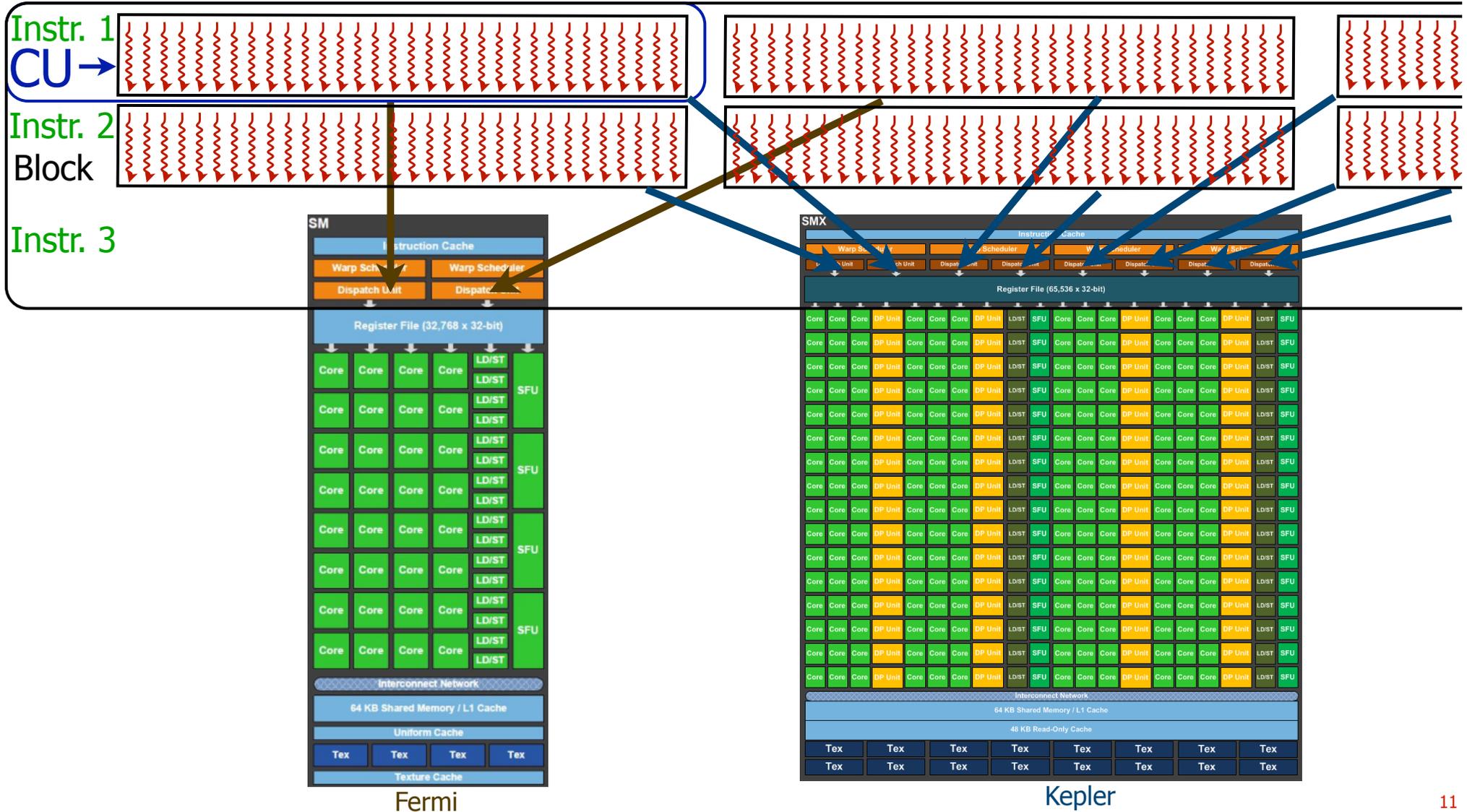
Instructions	Exec. Unit	Examples	Features	SM	SMX
Arithmetic	ALU	add, or, cmp	Swift operations	32	192 (37.5%)
Floating-point (SP)	FPU32	fadd32, fmul32	Share core with ALU	32	192 (37.5%)
Floating-point (DP)	FPU64	fsub64, fdiv64	Heavy workload	16	64 (12.5%)
Math functions	SFU	log, exp, sqrt	Unlikely used	16	32 (6.25%)
Load and store	DRAM	ld, st	High latency	4	32 (6.25%)

# Resources to execute instructions in parallel



Execution Unit	#	# warps
int 	192	6
fp32 	192	6
fp64 	64	2
load/store 	32	1
SFU 	32	1

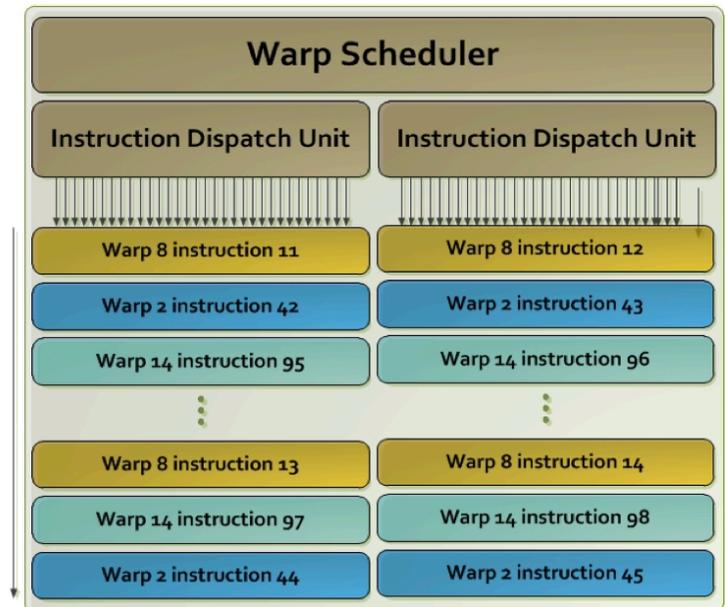
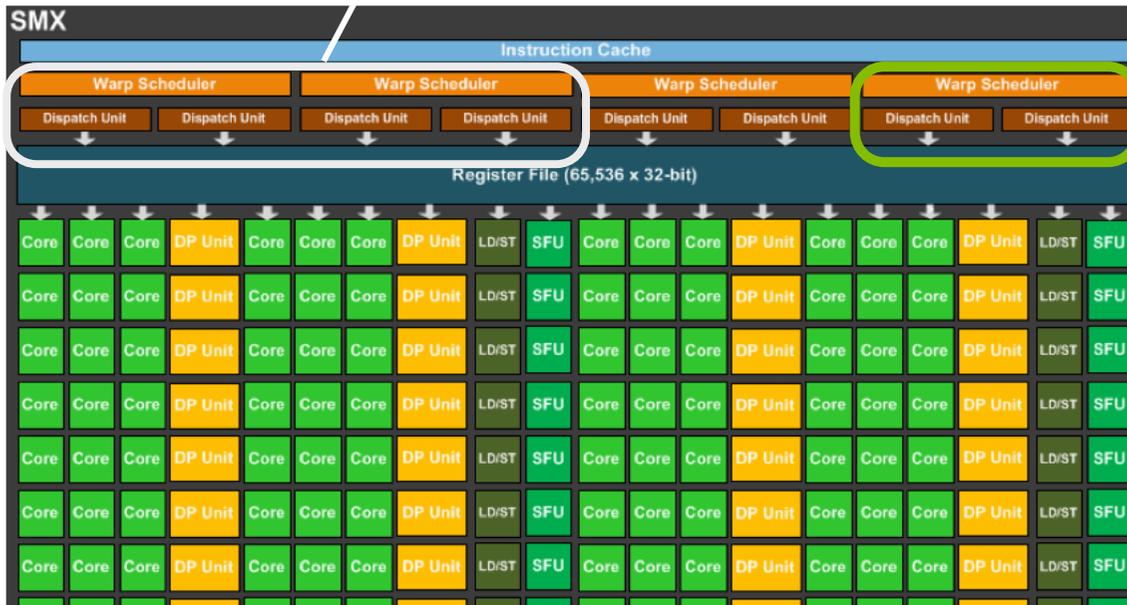
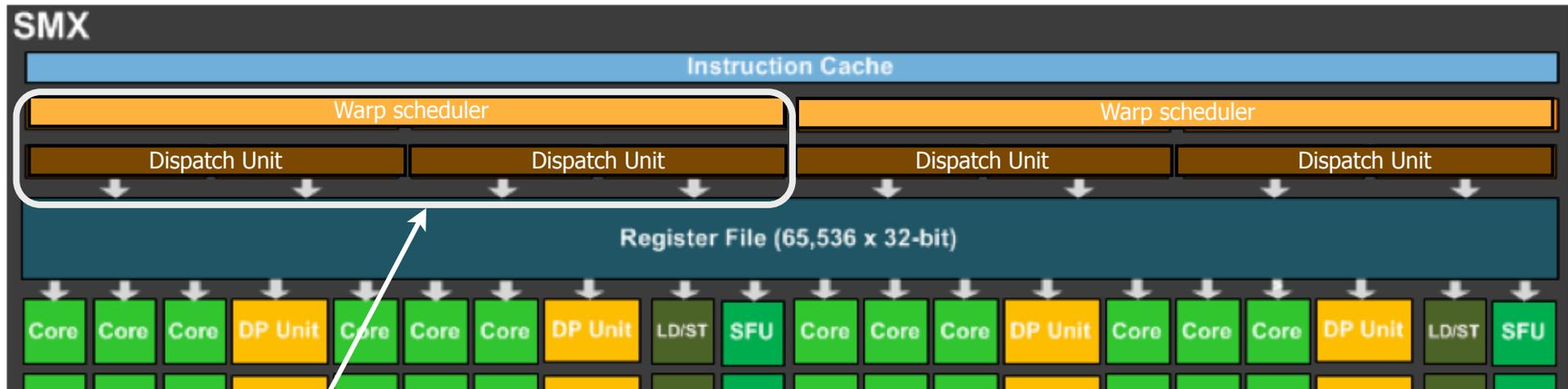
# The way each multiprocessor swallows SIMD instructions



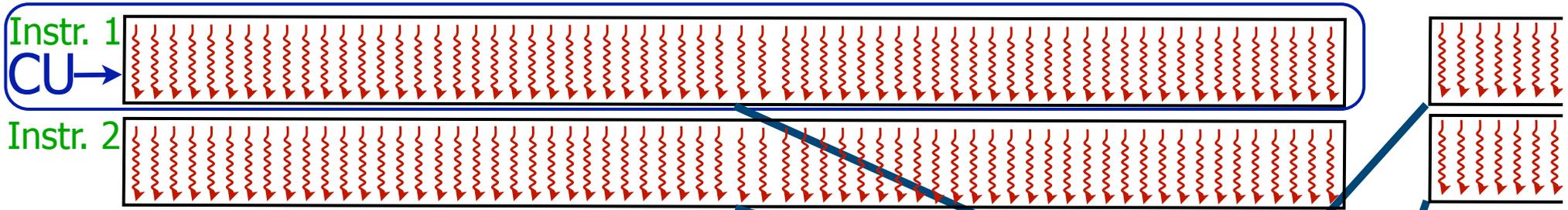
Fermi

Kepler

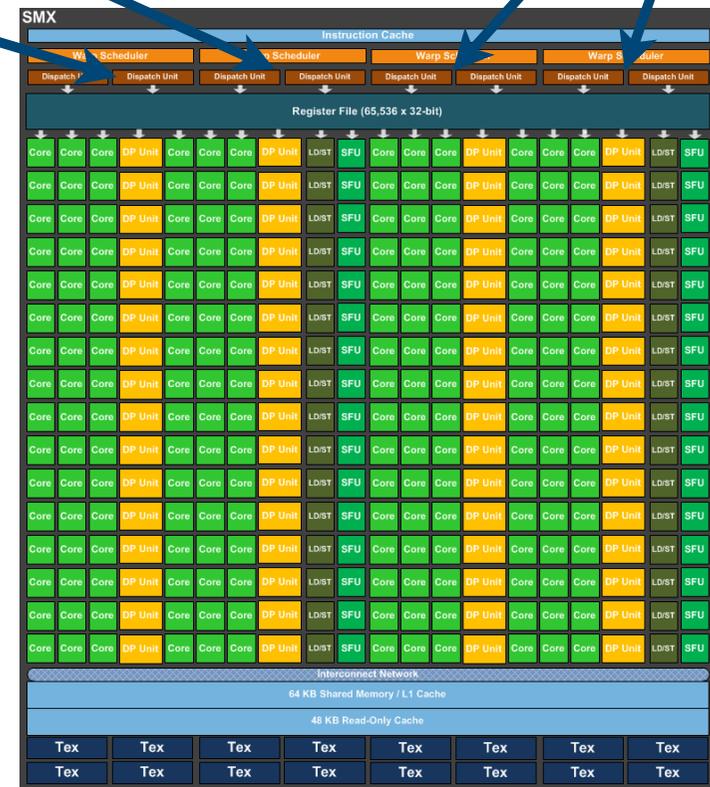
# A hypothetical GPU front-end with the warp size increased to 64



# The way each multiprocessor would swallow SIMD instructions using a warp size of 64

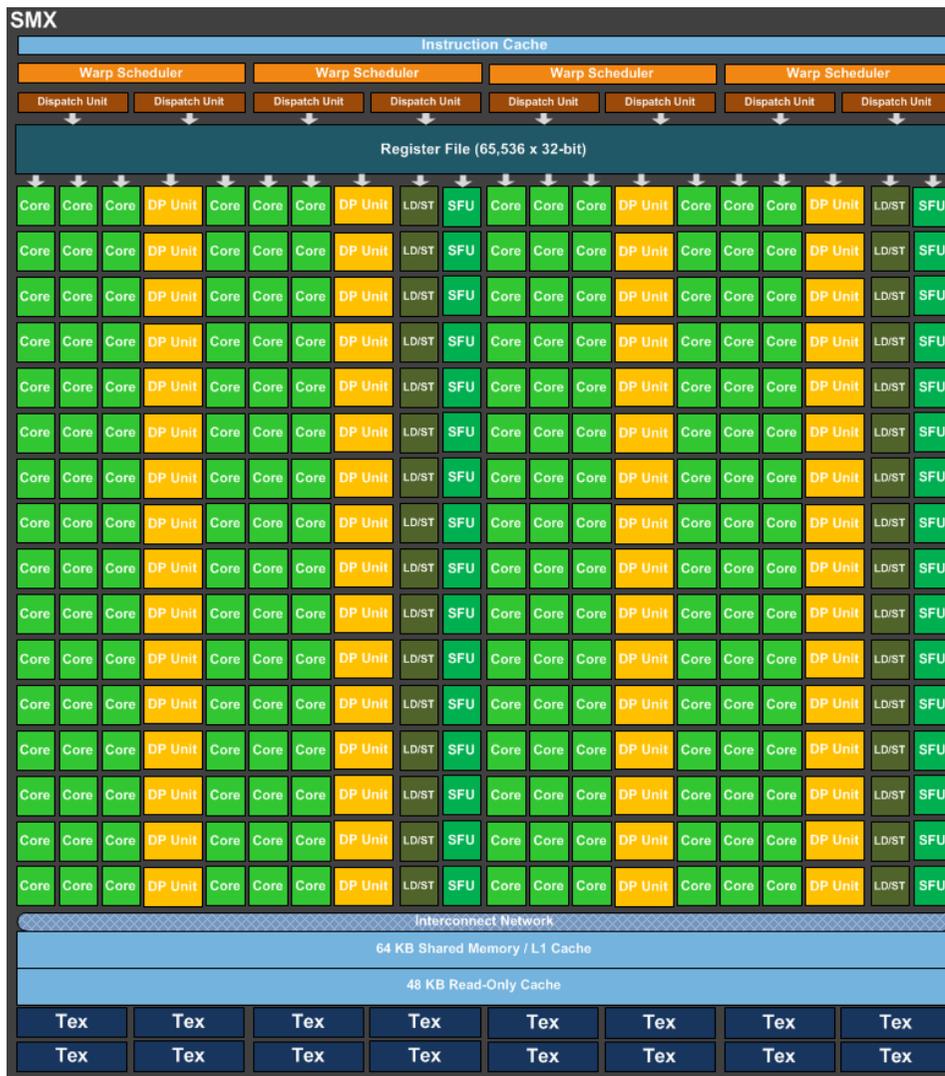


- The cost for the control unit is just the half.
- The penalty due to data dependencies is potentially lower, and the hardware is simplified.
- The penalty due to control dependencies is **higher**.



Kepler

# The GPU back-end: Transforming the SMX for a warp size of 64



Functional Unit	#	warp size = 32	warp size = 64
int/fp32 	192	6	3
fp64 	64	2	1
load/store 	32	1	1/2
SFU 	32	1	1/2

● The deficit lies in load/store and SFUs, but they were facing a tougher constraint during the Fermi generation, and they were able to recover from that.

# Other facts promoting the warp size to 64

---

- **Shared memory:** Concurrency attained through banks, and they were already increased from 16 (pre-Fermi) to 32.
- **Device memory:** Higher data bandwidth is required, but that is not the problem in the DDR saga (latency is).
- **Branching:** Techniques minimizing penalties on divergent branches are more mature and ready to face the challenge.
- **Scalability** in the number of cores: Simplicity in the control unit would allow to increase cores of every kind.
- Vendors are all moving in the same direction:
  - Ex: Graphics Core Next (AMD) is a 4 x 16-wide vector SIMD.

# To benefit from higher warp sizes

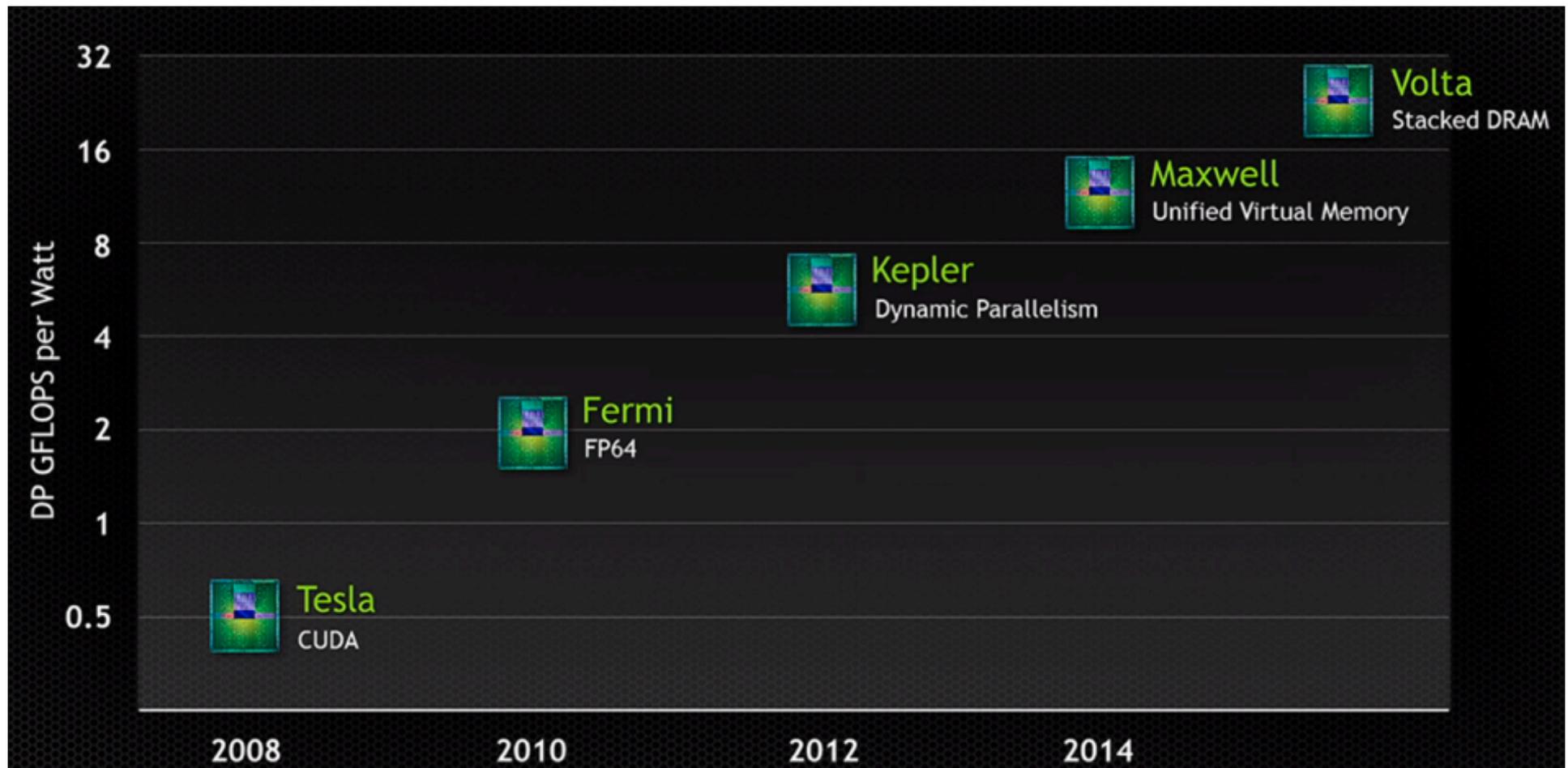
---

- Make blocks bigger:
  - Less than 64 threads per block is forbidden.
  - 256 would be the minimum required.
  - 384 gains momentum.
- Pay more attention to warp divergencies.
- Advantageous for regular computations. Sophistication of hardware scheduler (Hyper-Q, dynamic parallelism) lifts irregular applications.

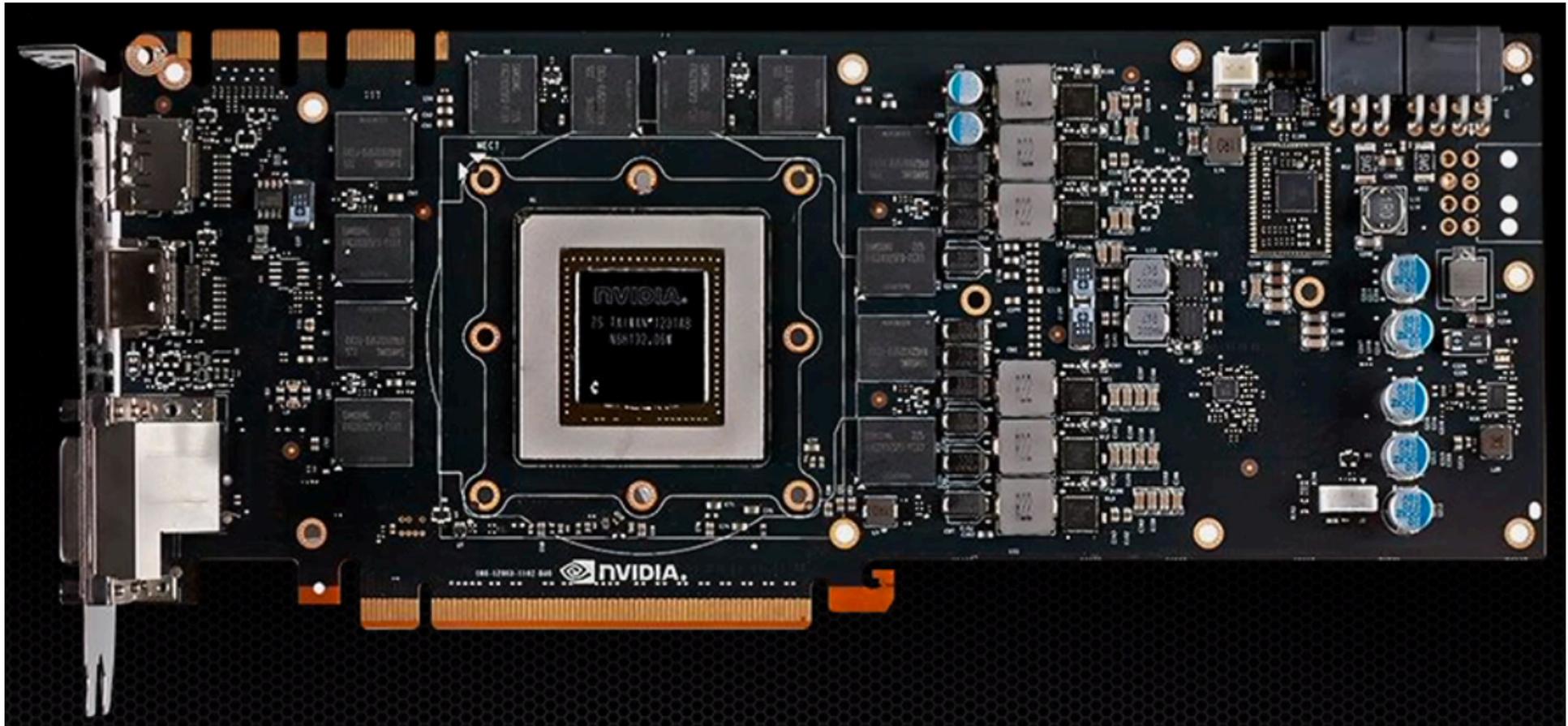
# III. The future: Many-cores with Stacked DRAM



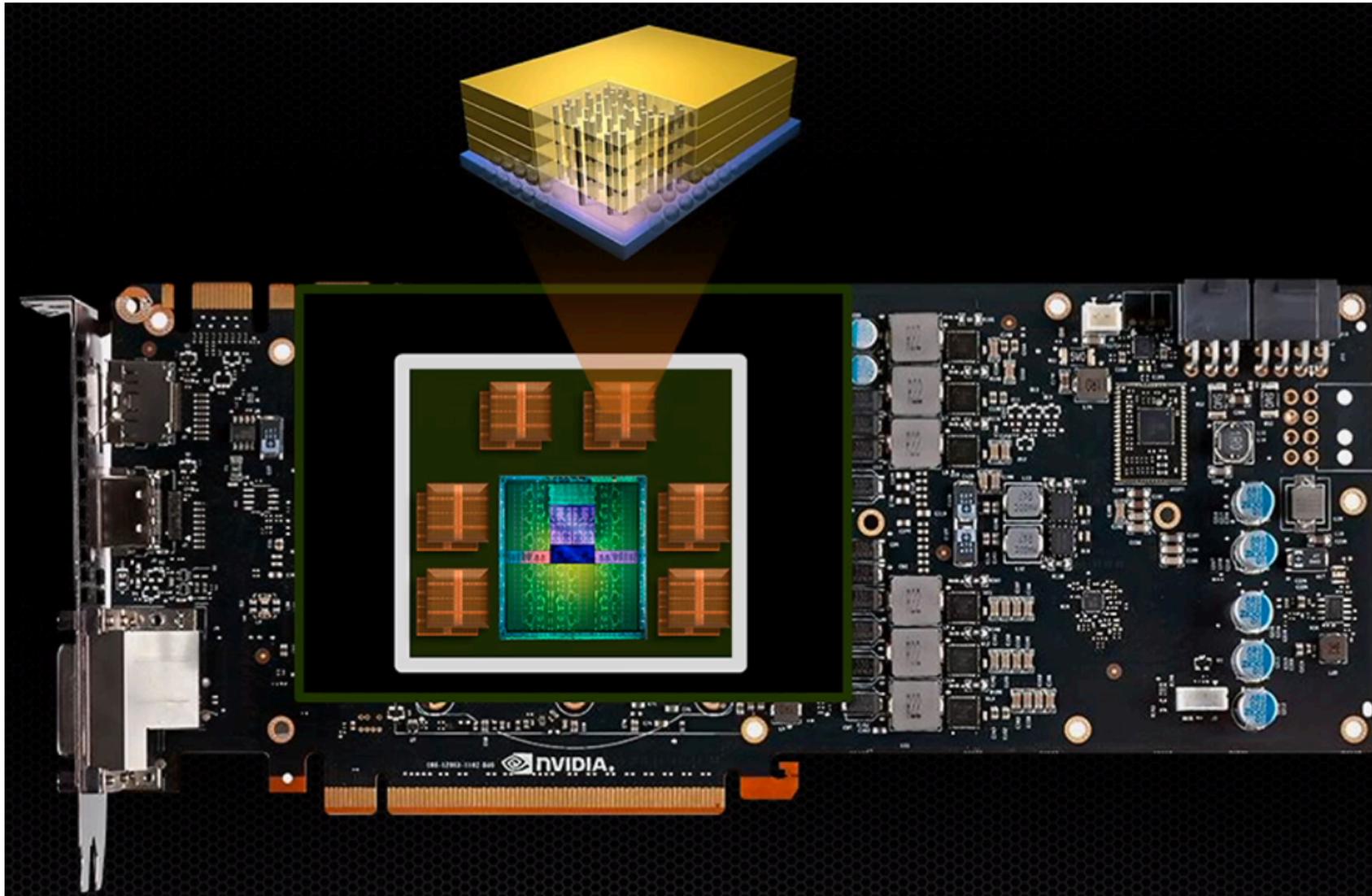
# A look ahead through Nvidia's GPU roadmap



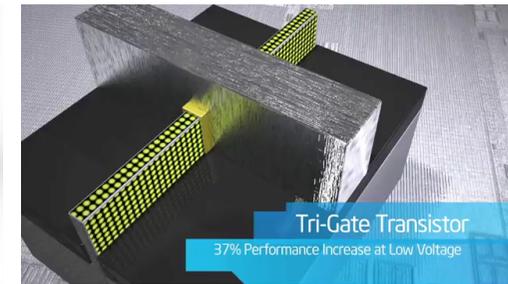
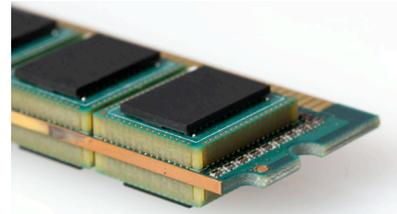
# A 2013 graphics card: Kepler GPU with GDDR5 video memory



# A 2016/17 graphics card: Volta GPU with Stacked (3D) DRAM



# Preliminary issues



- Our 3D analysis is not related to Intel 3D tri-gate transistors (but both are compatible).
- Our 3D chips are also compatible with existing 2D technologies (we do not sacrifice anything already attained).
- We focus on the processor, and overall, memory (but 3D manufacturing can be applied to CPU-GPU, SRAM-DRAM, ASICs, DSPs, ..., everywhere!).
- Heat sink remains on external layers (internal ones will likely be occupied by memory cells, technology favors SRAM-DRAM for once).

# A promising Stacked DRAM development: The Hybrid Memory Cube Consortium (HMCC)

HMCC achievements and milestones	Date
First papers published about Stacked DRAM (based of research projects)	2003-2006
First commercial announcement of the technology, by Tezzaron Semiconductors	January, 2005
HMC Consortium is launched by Micron Technologies and Samsung Electronics	October, 2011
Stacked DRAM announced for Volta GPU by Nvidia	March, 2013
Specification HMC 1.0 available	April, 2013
Production samples based on the standard	Second half of 2014 (estimated)
2.5 configuration available	End of 2014 (estimated)

# Hybrid Memory Cube at a glance

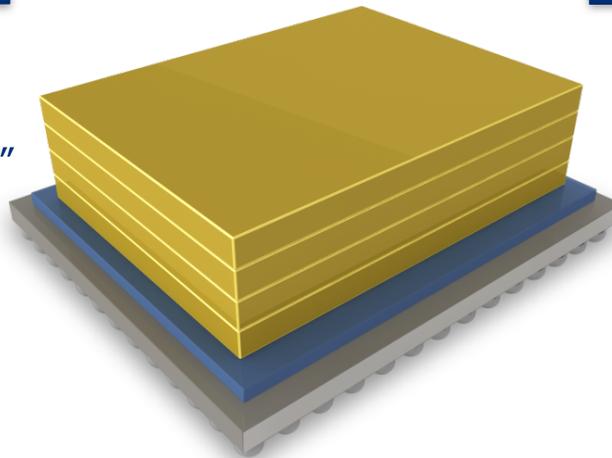
## Revolutionary Approach to Break Through the “Memory Wall”

- ▶ Evolutionary DRAM roadmaps hit limitations of bandwidth and power efficiency.
- ▶ Micron introduces a new class of memory: Hybrid Memory Cube.
- ▶ Unique combination of DRAMs on Logic.

### Key Features

- ▶ Micron-designed logic controller.
- ▶ High speed link to CPU.
- ▶ Massively parallel “Through Silicon Via” connection to DRAM.

Full silicon prototypes  
**TODAY**



### Unparalleled performance

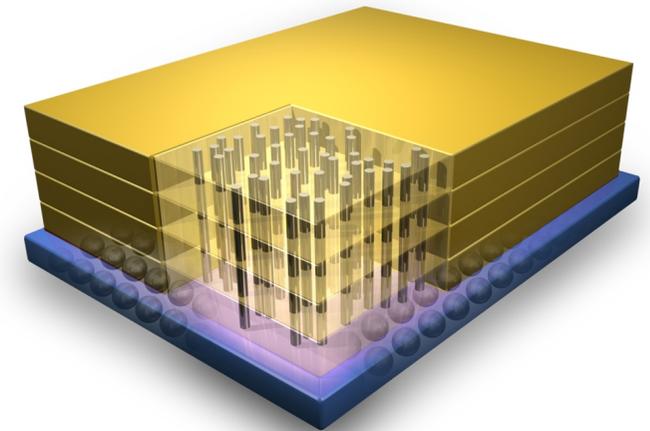
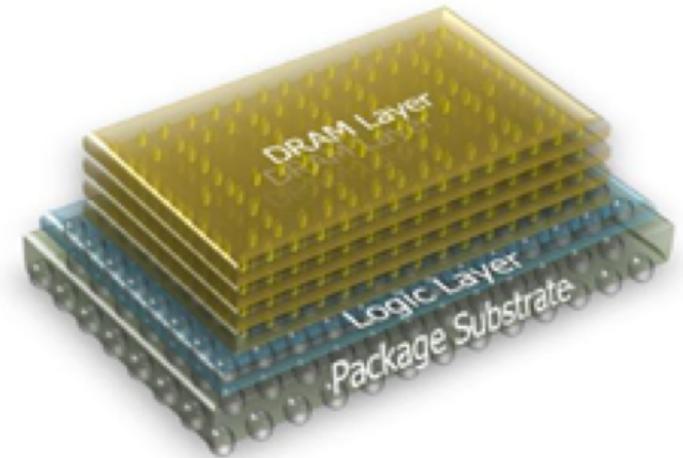
- ▶ Up to 15x the bandwidth of a DDR3 module [but just 2x vs. GDDR5].
- ▶ 70% less energy usage per bit than existing technologies [measured in number of active signals involved, power savings are 50% only].
- ▶ Occupying nearly 90% less space than today’s RDIMMs [95% savings].

[according to my own essay, which I will present here later]

Targeting high performance computing and networking, eventually migrating into computing and consumer

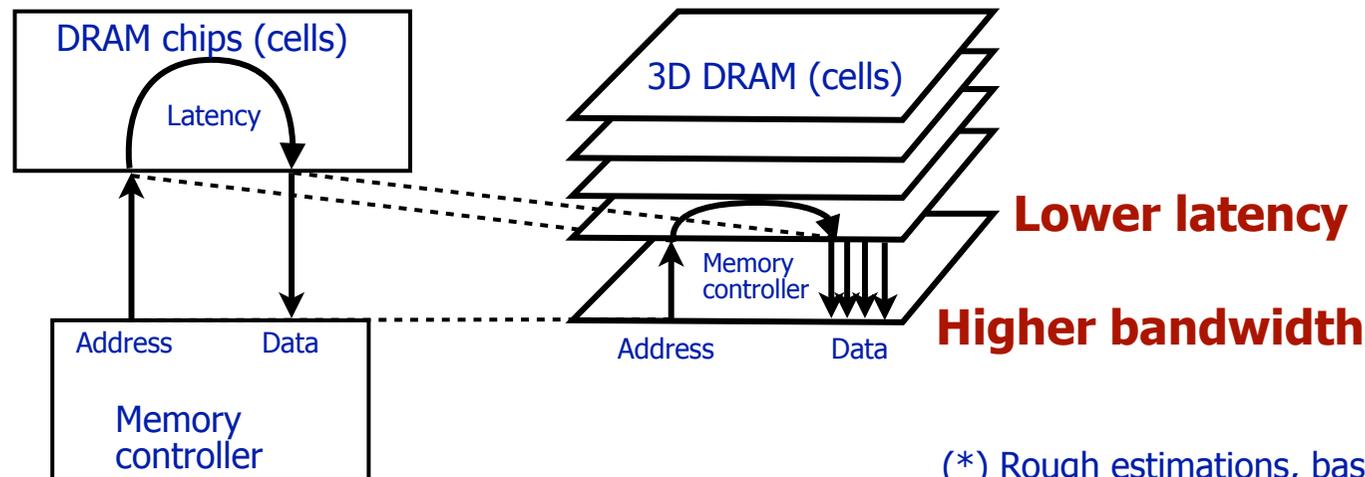
# Details on silicon integration

- DRAM cells are organized in **vaults**, which take borrowed the interleaved memory arrays from already existing DRAM chips.
- A logic controller is placed at the base of the DRAM **layers**, with data matrices on top.
- The assembly is connected with through-silicon vias, **TSVs**, which traverse vertically the stack using pitches between 4 and 50  $\mu\text{m}$ .
  - For a pitch of 10  $\mu\text{m}$ ., a 1024-bit bus (16 memory channels) requires a **die size** of 0.32  $\text{mm}^2$ , which barely represents 0.2% of a CPU die (160  $\text{mm}^2$ ).
  - Vertical **latency** to traverse the height of a Stacked DRAM endowed with 20 layers is only 12 picosecs.
- The final step is advanced package assembly of vaults, layers and TSVs. This prevents parasitic capacitances which reduce signal speed and increase power required to switch.



# What are the benefits for the DRAM chip?

- Speed doubles (\*), based on three benefits:
  - Shorter connections** between memory controller and DRAM cell matrices improve speed 1/3.
  - Wider buses up to 512 bits** thanks to higher wiring densities improve speed another 1/3.
  - Lower latencies** thanks to faster TSV connections and higher **interleaved factors** on a 3D geometry improve the remaining 1/3.



(\*) Rough estimations, based on simulations by G. Loh [ISCA'08], with improvement factors of 2.17x.

# Building the new DRAM chips in a 3D fashion

1. DRAM is partitioned into 16 vaults, similarly to the way the DDR saga did with banks or cell matrices to exploit spatial locality (legacy from initial designs coming from the old '80s).
2. Common logic is extracted from all those partitions, and placed at the logic base die.
3. DRAM is piled up in 4-high or 8-high configurations.
4. 16 vaults are built, and TSVs drilling holes in silicon through 4 or 8 layers. TSVs become the internal buses, and vaults the channels of the DDR saga, with outstanding interleaving factors and scalability.
5. A high speed bus (called a link) connects DRAM & processor, to be improved when moving from 2.5D to 3D. It is endowed with:
  1. Advanced switching.
  2. Optimized memory control.
  3. Simple interface.
  4. 16 transmits and receive lanes, each running at 10 GB/s.

These two buses are essential elements to preserve the legacy from the DDR saga.

# 3D integration, side by side with the processor

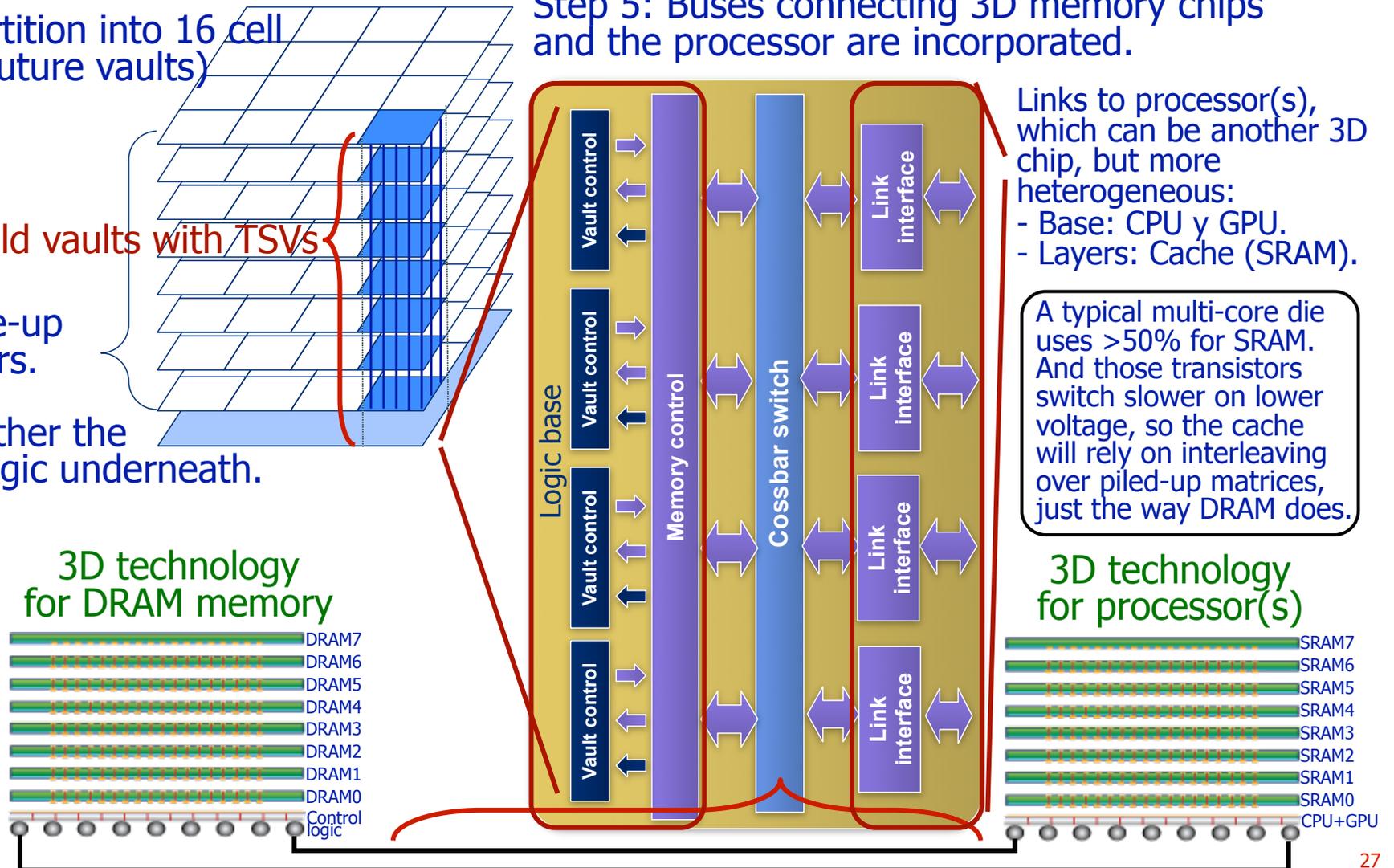
Step 1: Partition into 16 cell matrices (future vaults)

Step 4: Build vaults with TSVs

Step 3: Pile-up DRAM layers.

Step 2: Gather the common logic underneath.

Step 5: Buses connecting 3D memory chips and the processor are incorporated.



Typical DRAM chips use 74% of the silicon area for the cell matrices.

# IV. 3D DRAM gains versus DRAM technology



# Speed rates between memory and processor

<b>Bandwidth in 2013 between the memory controller and the processor (on each direction)</b>	<b>Short reach (for wirings between 20 and 25 cm. long)</b>	<b>Ultra Short reach (for printed circuit boards between 5-8 cm.)</b>
For every pin	15 Gbits/s.	10 Gbits/s.
For every HMC link (16 bits)	30 GBytes/s.	20 GBytes/s.
For every memory channel (64 bits)	120 GBytes/s.	80 GBytes/s.
For a CPU with 4 memory channels	Does not apply	320 GByte/s.
For a GPU 384-bits wide	Does not apply	480 GByte/s.

<b>Bandwidth in 2015 between the memory controller and the processor (on each direction)</b>	<b>Short reach (for wirings between 20 and 25 cm. long)</b>	<b>Ultra Short reach (for printed circuit boards between 5-8 cm)</b>
For every pin	28 Gbits/s.	15 Gbits/s.
For every HMC link (16 bits)	56 GBytes/s.	30 GBytes/s.
For every memory channel (64 bits)	224 GBytes/s.	120 GBytes/s.
For a CPU with 4 memory channels	Does not apply	480 GBytes/s.
For a GPU 284-bits wide	Does not apply	720 GBytes/s.

# A comparative in bandwidth with existing technologies

- On a CPU system (PC with a 4-channel motherboard, 256 bits):
  - [2013] DDR3 @ 4 GHz (2x 2000 MHz): 128 Gbytes/s.
  - [2014] A CPU with HMC 1.0 (first generation): 320 Gbytes/s. on each dir.
  - [2015] A CPU with HMC 2.0 (second generation): 448 Gbytes/s.
- On a GPU system (384-bits wide graphics card):
  - [2013] A GPU with GDDR5 @ 7 GHz (2x 3500 MHz): 336 Gbytes/s.
  - [2014] A GPU with 12 chips of 32 bits manuf. using near memory HMC 1.0 would reach **480 Gbytes/s.** (6 channels HMC 1.0 @ 80 GB/s. each).
  - [2015] A GPU using HMC 2.0 (112 GB/s.) would reach **672 Gbytes/s.,** which doubles the bandwidth with respect to the most advanced GDDR technology in 2013.

(\*) Taking the bandwidth estimations given by HMCC 1.0 y 2.0 (20 and 28 GB/s. respectively on each 16-bit link for each direction). Nvidia already confirmed in GTC'13 data bandwidths around 1 TB/s. for its Volta GPU.

# What it takes to each technology to reach 640 GB/s.

Circuitry required	DDR3L-1600	DDR4-3200	Stacked DRAM HMC 1.0
Data bandwidth (GB/s.)	12.8 per module	25.6 per module	20 per link of 16 bits
Items required to reach 640 GB/s.	50 modules	25 modules	32 links (8 3D chips)

Active signals	DDR3L-1600	DDR4-3200	Stacked DRAM HMC 1.0
Active pinout required	143 per module	148 per module	270 per chip
Total number of electrical lines	7150	3700	2160 <b>(70% savings)</b>

Energy consumed	DDR3L-1600	DDR4-3200	Stacked DRAM HMC 1.0
Watts (W.)	6.2 per module	8.4 per module	5 per link
Power consumed for 640 GB/s.	310 W.	210 W.	160 W. <b>(50% savings)</b>

Physical space on motherboard	DDR3L-1600	DDR4-3200	Stacked DRAM HMC 1.0
Module area (width x height)	165 mm. x 10 mm. = 1650 mm <sup>2</sup>		1089 mm <sup>2</sup> per chip
Total area occupied for 640 GB/s.	825 cm <sup>2</sup>	412.5 cm <sup>2</sup>	43.5 cm <sup>2</sup> <b>(95% savings)</b>

# Improvements published by other manufacturers

---

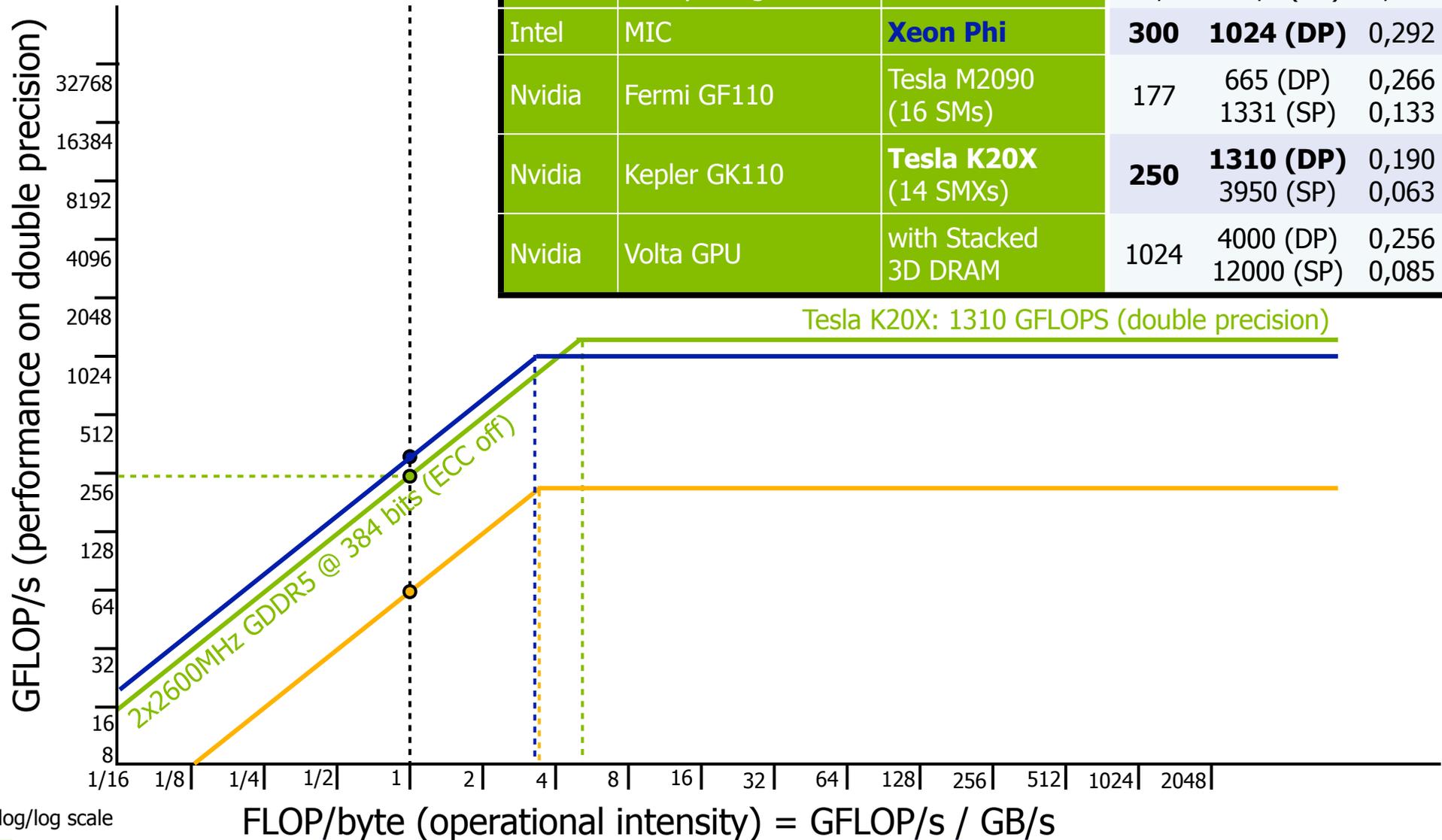
- [2008] A prototype of Tezzaron Semiconductors reduces 32.5% CL and RCD latencies, as compared to 2D technology of the same memory type.
- [2009] A 8 GB. 3D DDR3 chip by Samsung increments bandwidth from 1066 MB/s to 1600 MB/s, an additional 50%. Passive power is reduced 50%, active power is cut by 25%.
- [2012] IBM's implementation for Micron within the HMC 1.0 standard reaches bandwidths around 128 GB/s., consuming 10 watts (compared to 82 watts consumed by 15 DIMMs of equivalent DDR3-1333).

# V. Impact on GPUs and accelerators

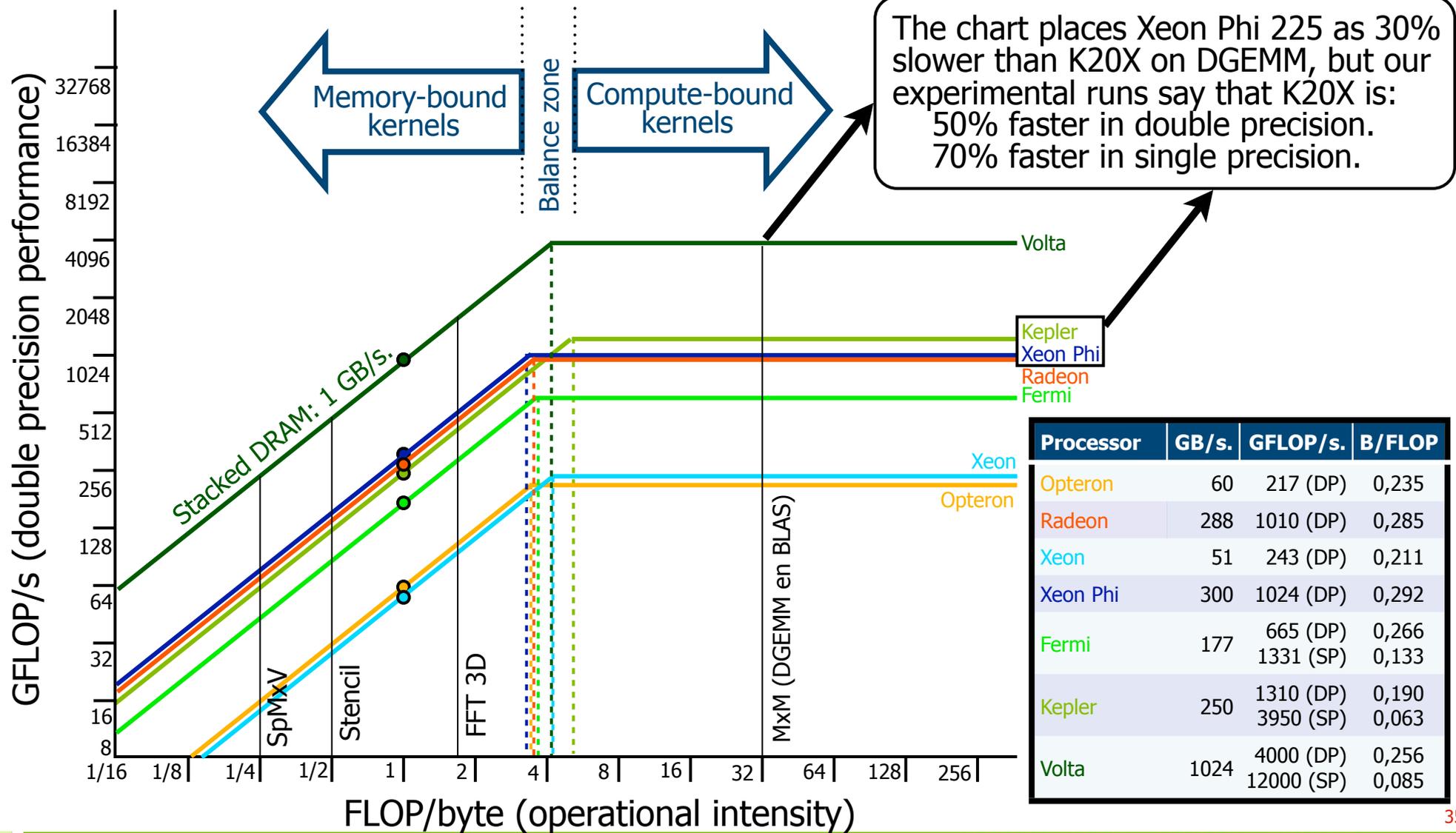


# Platforms to compare

Vendor	Microarchitecture	Model	GB/s.	GFLOP/s.	Byte/ FLOP
AMD	Bulldozer	<b>Opteron 6284</b>	<b>59,7</b>	<b>217,6 (DP)</b>	0,235
AMD	Souther Islands	Radeon HD7970	288	1010 (DP)	0,285
Intel	Sandy Bridge	Xeon E5-2690	51,2	243,2 (DP)	0,211
Intel	MIC	<b>Xeon Phi</b>	<b>300</b>	<b>1024 (DP)</b>	0,292
Nvidia	Fermi GF110	Tesla M2090 (16 SMs)	177	665 (DP) 1331 (SP)	0,266 0,133
Nvidia	Kepler GK110	<b>Tesla K20X</b> (14 SMXs)	<b>250</b>	<b>1310 (DP)</b> 3950 (SP)	0,190 0,063
Nvidia	Volta GPU	with Stacked 3D DRAM	1024	4000 (DP) 12000 (SP)	0,256 0,085

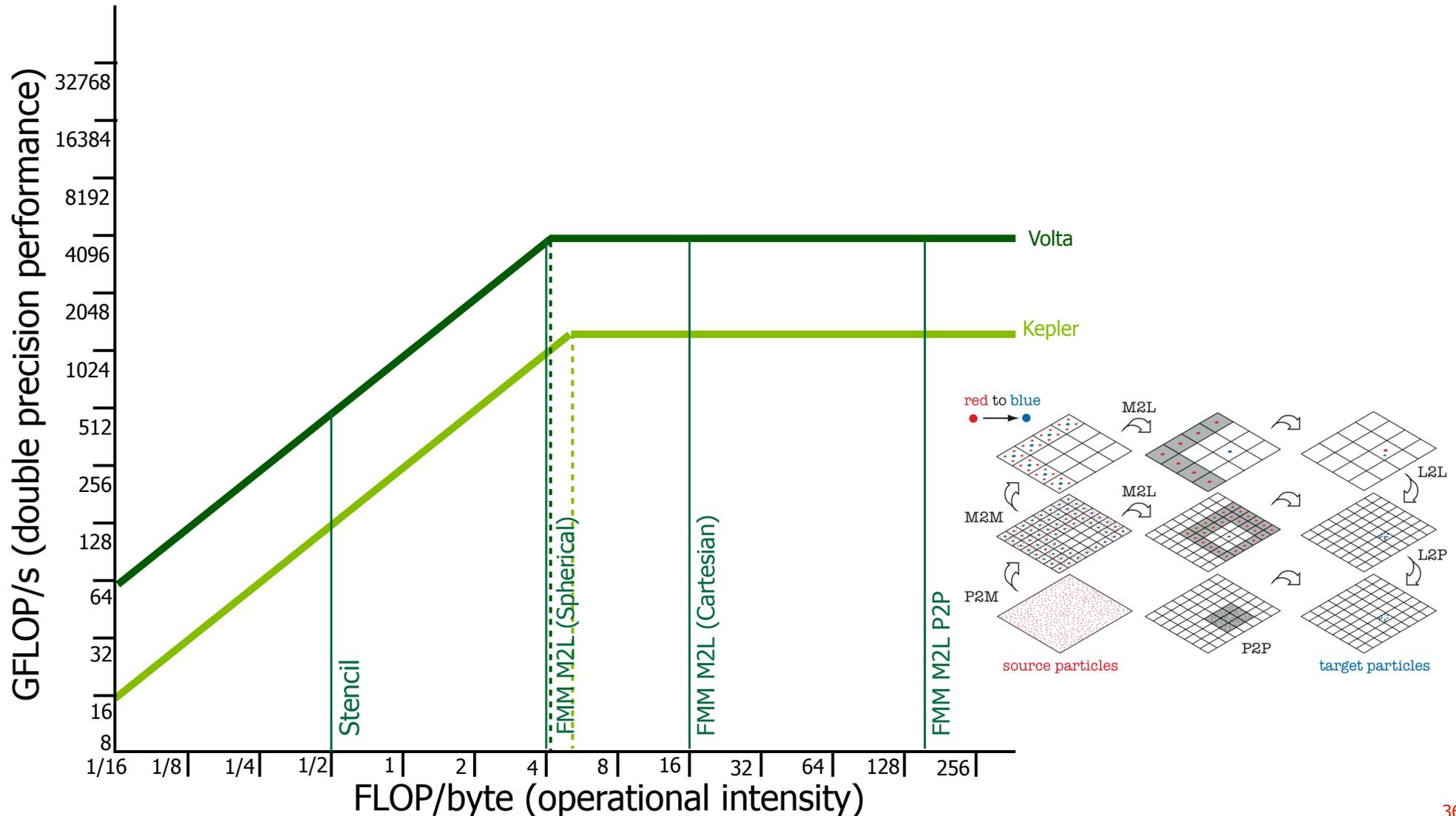


# The Roofline model: Hardware vs. Software



# The Roofline model: Software evolution.

## Case study: FMM (Fast Multipole Method)



# Conclusions

---

- The GPU SIMD model will benefit from vector processing.
- A wide number of GPU kernels are memory-bound, and to protect against it, they have evolved by raising FLOPS/byte.
- A new 3D integration favors memory chips, allowing to:
  - Reduce latency for basic memory cells, which was plain over the last two decades.
  - Improve the DDR# saga, increasing: Capacity (3D), communications (TSVs) and interleaving (vaults).
- The next generation of 3D chips will be more heterogeneous, and will converge to SoC (system-on-Chip).

# Acknowledgments

---

- To the great engineers at Nvidia, for sharing ideas, material, figures and presentations. And to the company for its support and sponsorship.
- To Lorena Barba (Nvidia CUDA Fellow), for her excellent contribution to the Roofline memory model.
- To Scott Stevens and Susan Platt (Micron) for providing me technical info from the HMCC, incorporated to this presentation under explicit permission.
- To Gabriel Loh (GaTech, now working for AMD) for allowing me to use the figures of his technical papers about previous memory technologies.

# Thanks so much for your attention

---

- You can always reach me in Spain at the Computer Architecture Department of the University of Malaga:
  - e-mail: [ujaldon@uma.es](mailto:ujaldon@uma.es)
  - Phone: +34 952 13 28 24.
  - Web page: <http://manuel.ujaldon.es> (in english and spanish).
- 40 CUDA/GPU courses taught around the world as Nvidia CUDA Fellow. In case you want to organize one in your University soon, please feel free to contact me here during the breaks (Nvidia will cover all expenses).

