GPGPU: Challenges ahead

PPAM'15 conference

Krakow (Poland). September, 6th-9th, 2015

Manuel Ujaldón

A/Prof. @ Univ. of Malaga (Spain)

Conjoint Senior Lecturer @ Univ. of Newcastle (Australia)

CUDA Fellow @ Nvidia

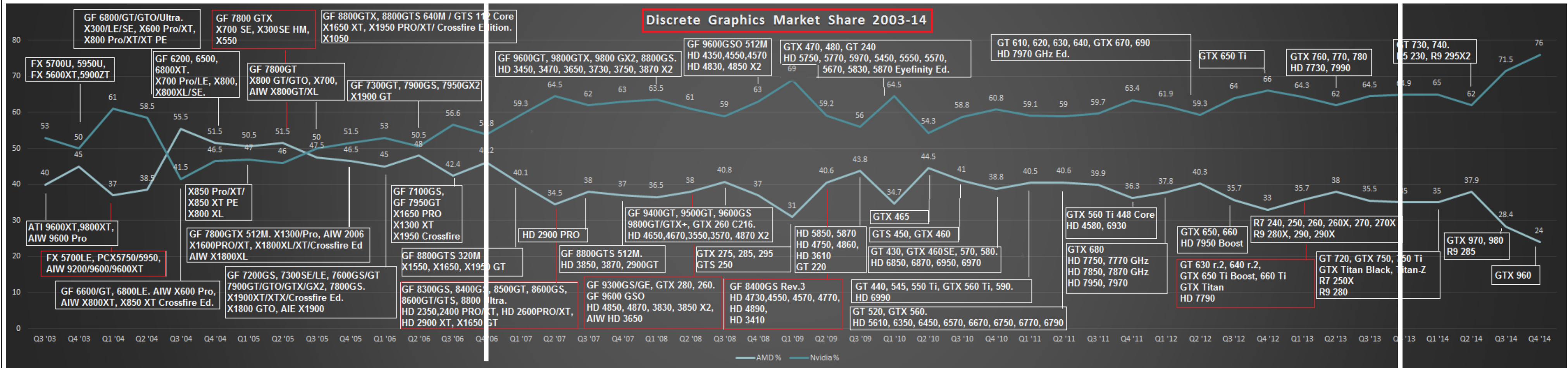
Talk contents [37 slides]

1. Past, present and future [5]
2. Transistors and memory improvements [19]
    1. New manufacturing processes [2]
    2. New memories [15]
3. Stacked DRAM [10]
    1. HMC (Hybrid Memory Cube) [6]
    2. HBM (High Bandwidth Memory) [3]
4. Impact on GPUs and concluding remarks [3]

I. Past, present and future

Past:
The GPU market share

Source: Jon Peddie Research consulting



	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	1Q15	2Q15
Nvidia	50%	53%	51%	55%	63%	63%	64%	61%	64%	66%	65%	76%	77%	81%
AMD	45%	46%	45%	46%	37%	37%	36%	39%	36%	33%	35%	24%	22%	18%

Pre-CUDA era: 1-1 | Stable period of 7 years: 2-1 | 3-1 | 4-1

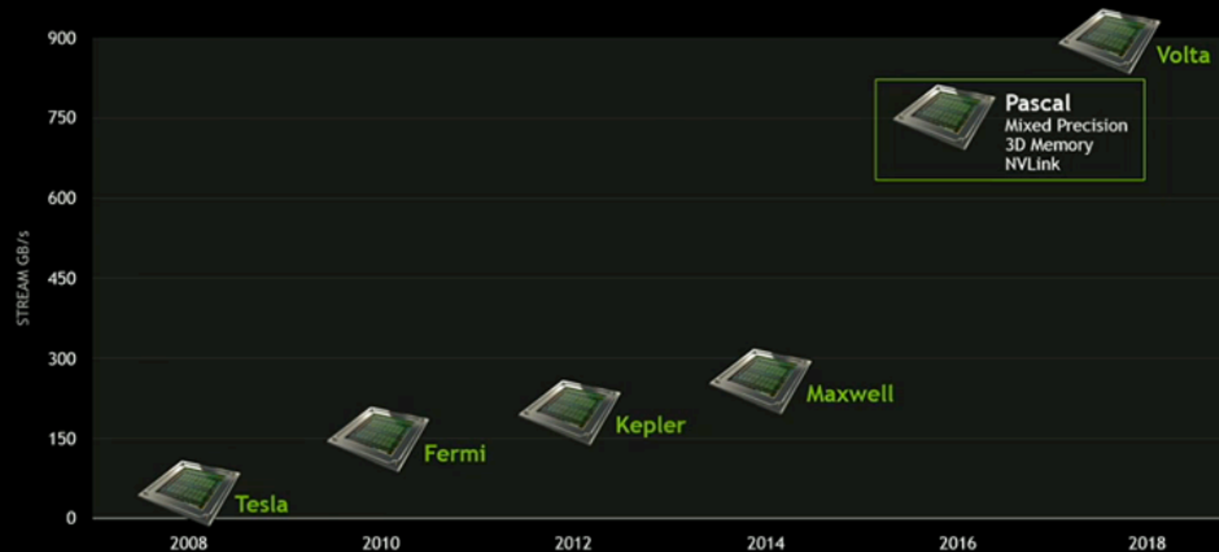
Present: Two hibernating movers wake up

CCC	Code names	Commercial series	Year range	Manufacturing process @ TSMC	Graphics memory
1.0	G80	8xxx	2006-07	90 nm.	DDR3
1.1	G84,6 G92,4,6,8	8xxx/9xxx	2007-09	80, 65, 55 nm.	DDR2/DDR3
1.2	GT215,6,8	2xx	2009-10	40 nm.	DDR2/DDR3
1.3	GT200	2xx	2008-09	65, 55 nm.	DDR3
2.0	GF100, GF110	4xx/5xx	2010-11	40 nm.	DDR3/DDR5
2.1	GF104,6,8, GF114,6,8,9	4xx/5xx/7xx	2010-13	40 nm.	DDR3/DDR5
3.0	GK104,6,7	6xx/7xx	2012-14	28 nm.	DDR3/DDR5
3.5	GK110, GK208	6xx/7xx/Titan	2013-14	28 nm.	DDR3/DDR5
3.7	GK210 (2xGK110)	Titan	2014	28 nm.	DDR3/DDR5
5.0	GM107,8	7xx	2014-15	28 nm.	DDR3/DDR5
5.2	GM200,4,6	9xx/Titan	2014-15	28 nm.	DDR5

Future: GTC'15 official announcements

GPU ROADMAP
Pascal 2.7x Memory Capacity

GPU ROADMAP
Pascal 3x Bandwidth



United States to build two flagship supercomputers

SUMMIT	SIERRA
150-300 PFLOPS Peak Performance	> 100 PFLOPS Peak Performance

IBM POWER9 CPU + NVIDIA Volta GPU

NVLink High Speed Interconnect

>40 TFLOPS/Node >3,400 Nodes

2017

Major Step Forward on the Path to Exascale

Past, present and future in numerical accuracy: Trade-off vs. performance

- [2010] Fermi: float (fp32) 2x faster than double (fp64).
- [2012] Kepler: fp32 3x fp64.
- [2014] Maxwell: fp32 32x fp64.
- [2016] Pascal: Introducing half-precision (fp16) 2x fp32.
- Half precision widely used in video-games and deep learning applications, so expect good scalability in future GPU generations.

II. Transistors and memory improvements

Benefits

- When you shrink the transistor gate, you get:
  - Faster switching:
    - Higher frequency.
  - Smaller units:
    - More transistors per chip.
    - Bigger designs.
  - Lower power:
    - Less heat.
    - Wider autonomy.

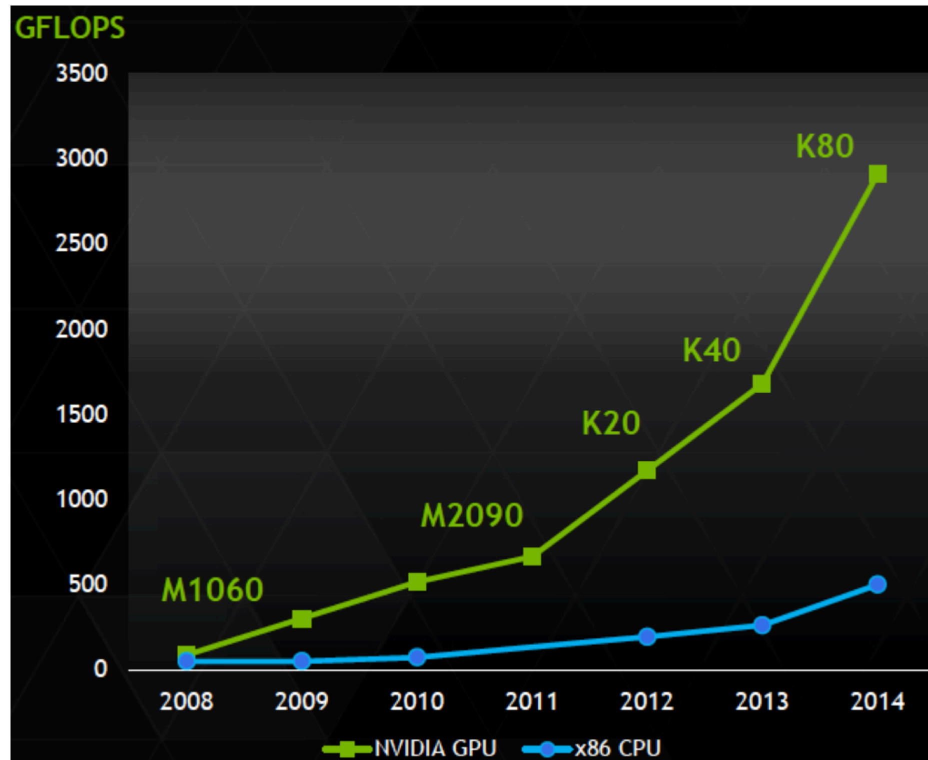
More GFLOPS/W

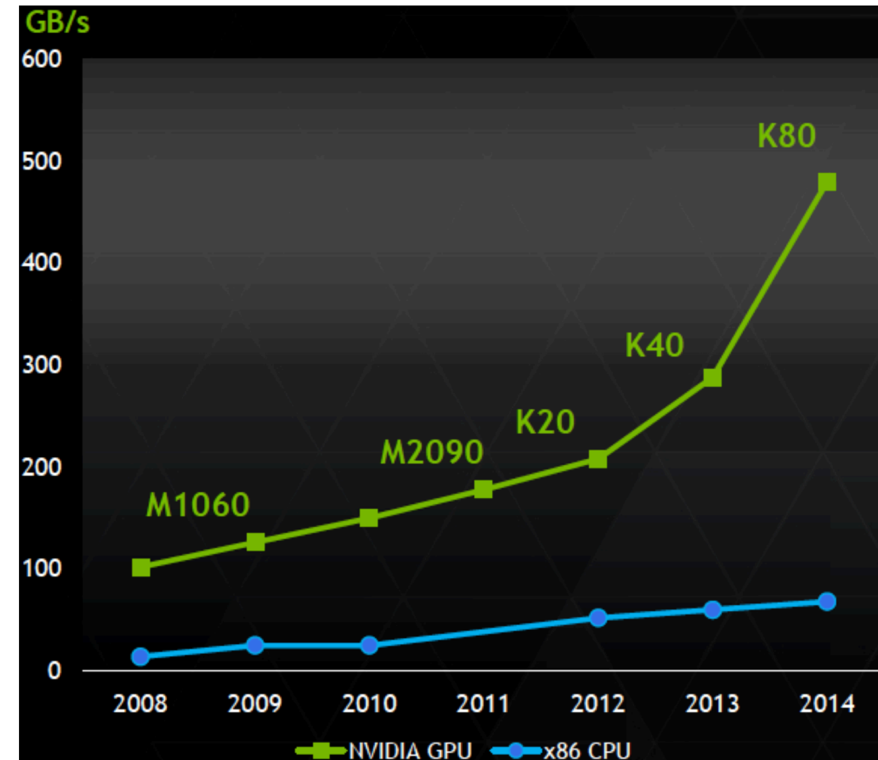
- When you adopt Stacked-DRAM, you get:
  - Faster response:
    - Higher frequency and bandwidth.
  - High density packaging:
    - More bytes per chip.
    - Bigger sizes.
  - Low power:
    - Less heat.
    - Wider autonomy.

More bandwidth

GPU peak performance vs. CPU

Peak GFLOPS (fp64)	Peak Memory Bandwidth





- GPU 6x faster on "`double`":
  - GPU: 3000 GFLOPS
  - CPU: 500 GFLOPS
- GPU 6x more bandwidth:
  - 7 GHz x 48 bytes = 336 GB/s.
  - 2 GHz x 32 bytes = 64 GB/s.

II.1. New manufacturing processes

Manufacturing process for a fabless company

- A loyal partner for more than 15 years has been TSMC.
- After many speculations, NVIDIA announced in Nov'14 to use TSMC's next-generation **16nm** FinFET process.
- They skip the **20nm** node. Intel & Samsung now in 14nm.
- Roadmap (already announced by TSMC):
  - Past: [4Q'11] They introduced **28nm.**
  - Present: **16nm** FinFET.
    - [4Q'15] Volume production.
    - [1Q'16] Commercial chips. Pascal will arrive shortly after this starting point.
  - Future: **10nm** 3D FinFET.
    - [4Q'16] Available to customers.
    - [1Q'17] Volume production.
  - Beyond: [4Q'17] **7nm** 3D FinFET.

Benefits of moving from the last 28nm node to the first 16nm node

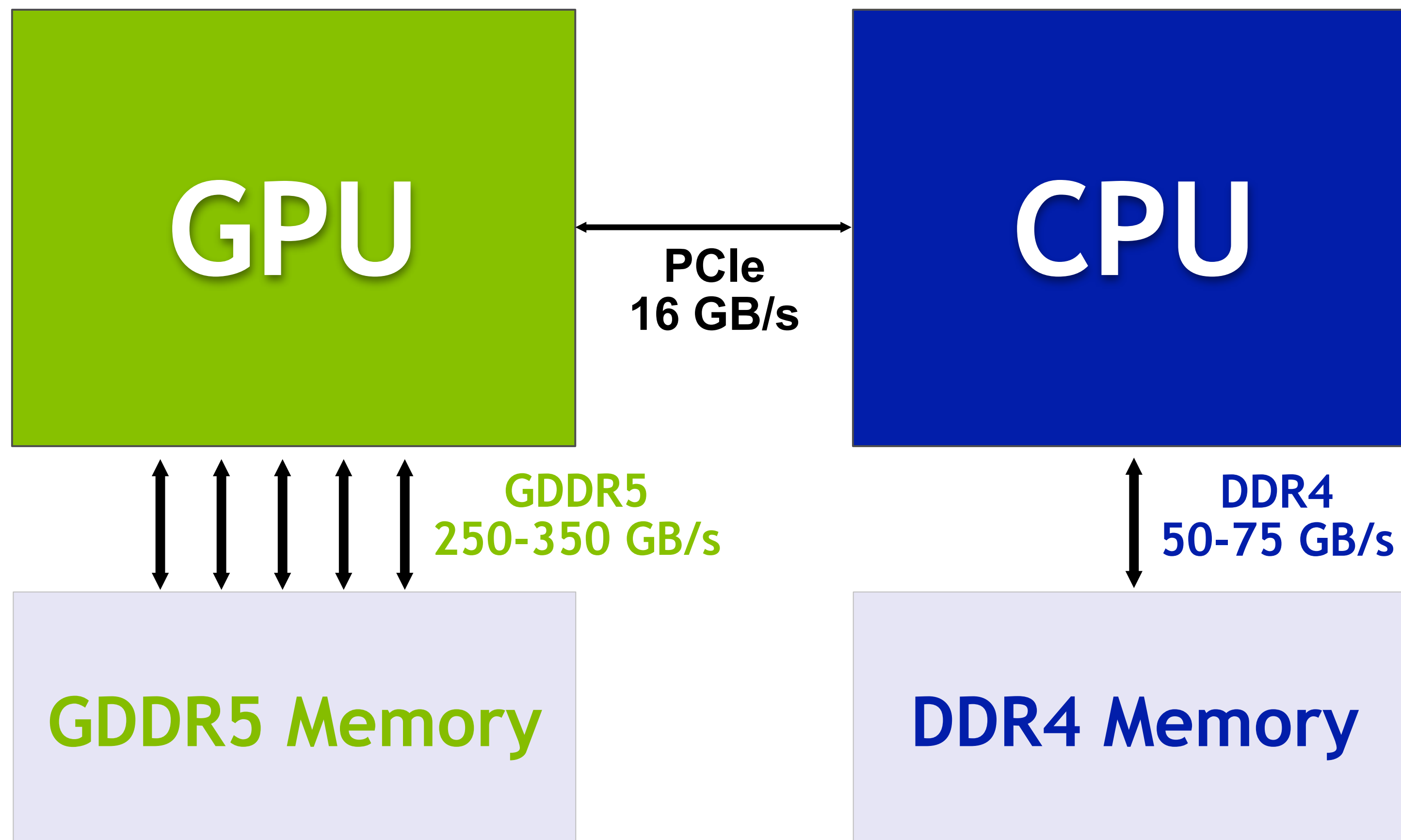
- 40% more performance at the same power draw.
- 50% less power at the same speed.

Source: Cadence (TSMC's partner)

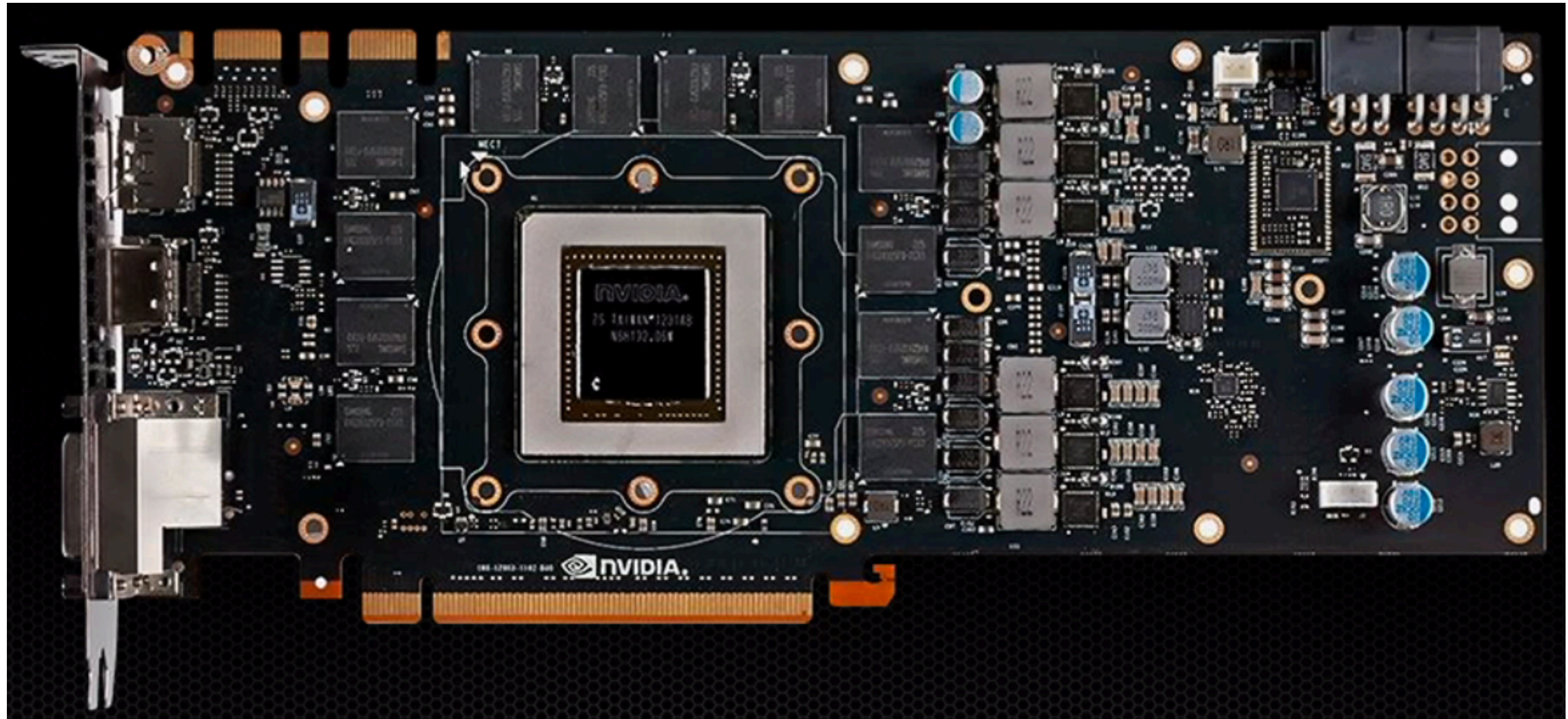
II.2. New memories

NVIDIA

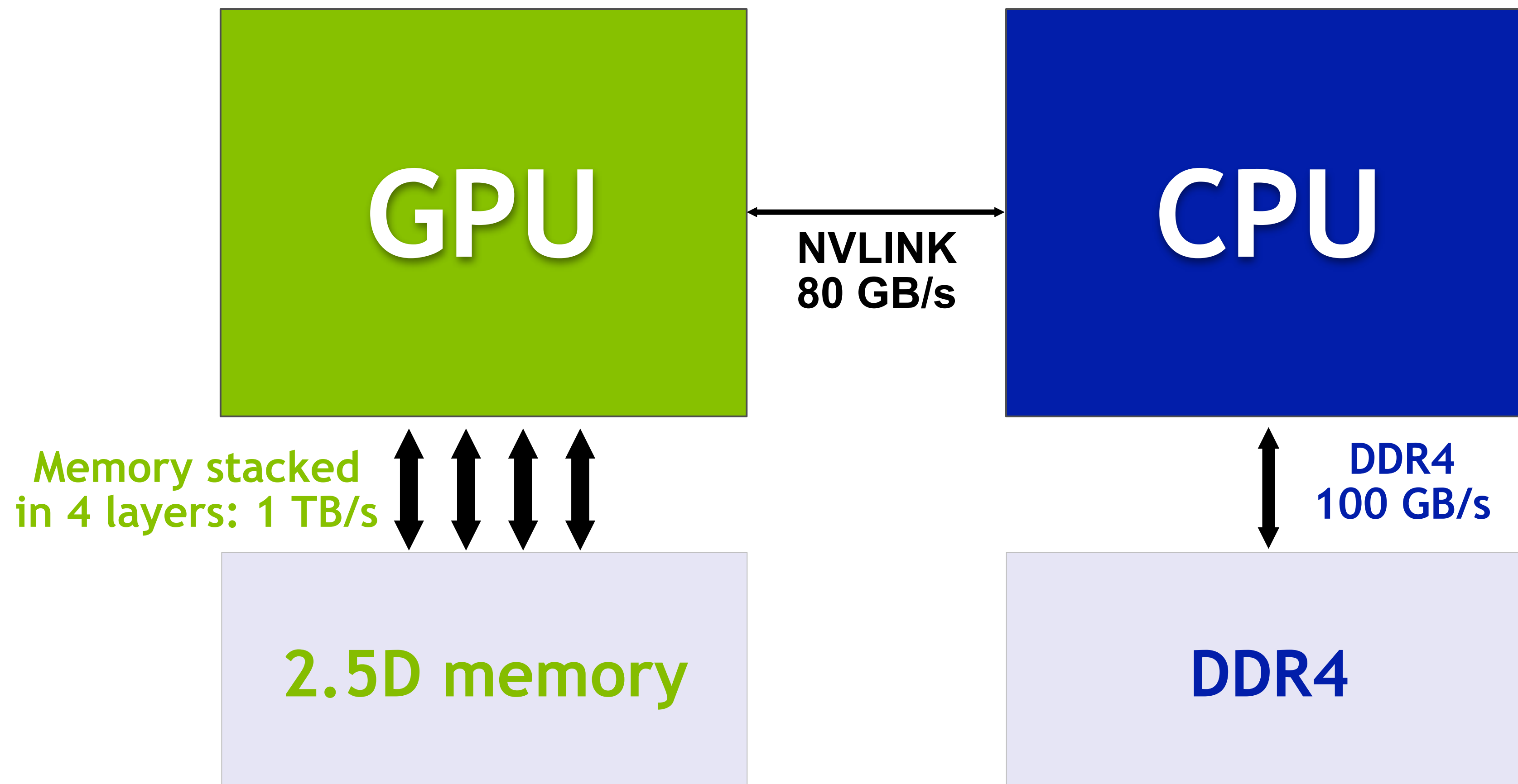
Today



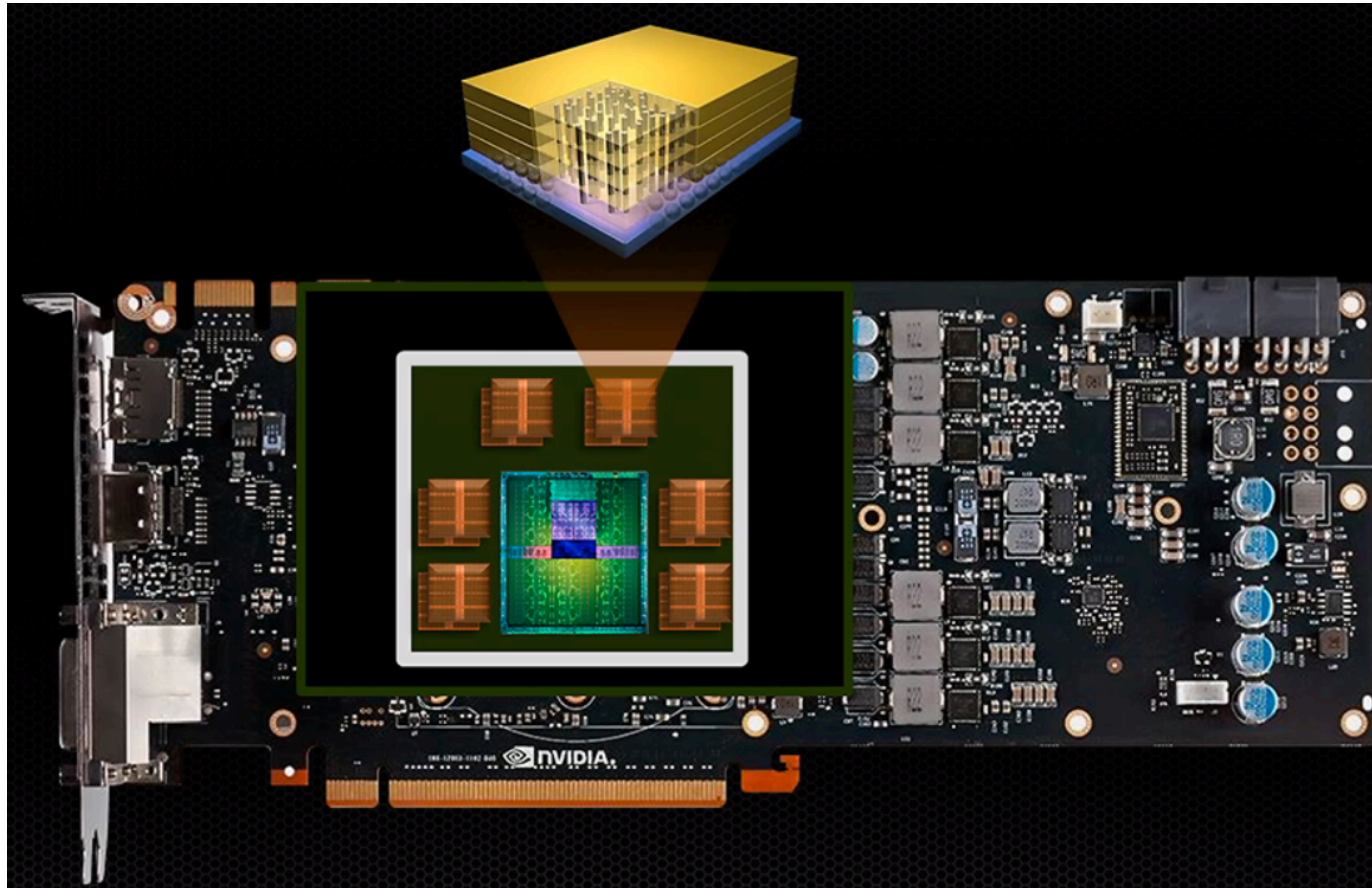
A 2014/15 graphics card: Kepler/Maxwell GPU with GDDR5 memory



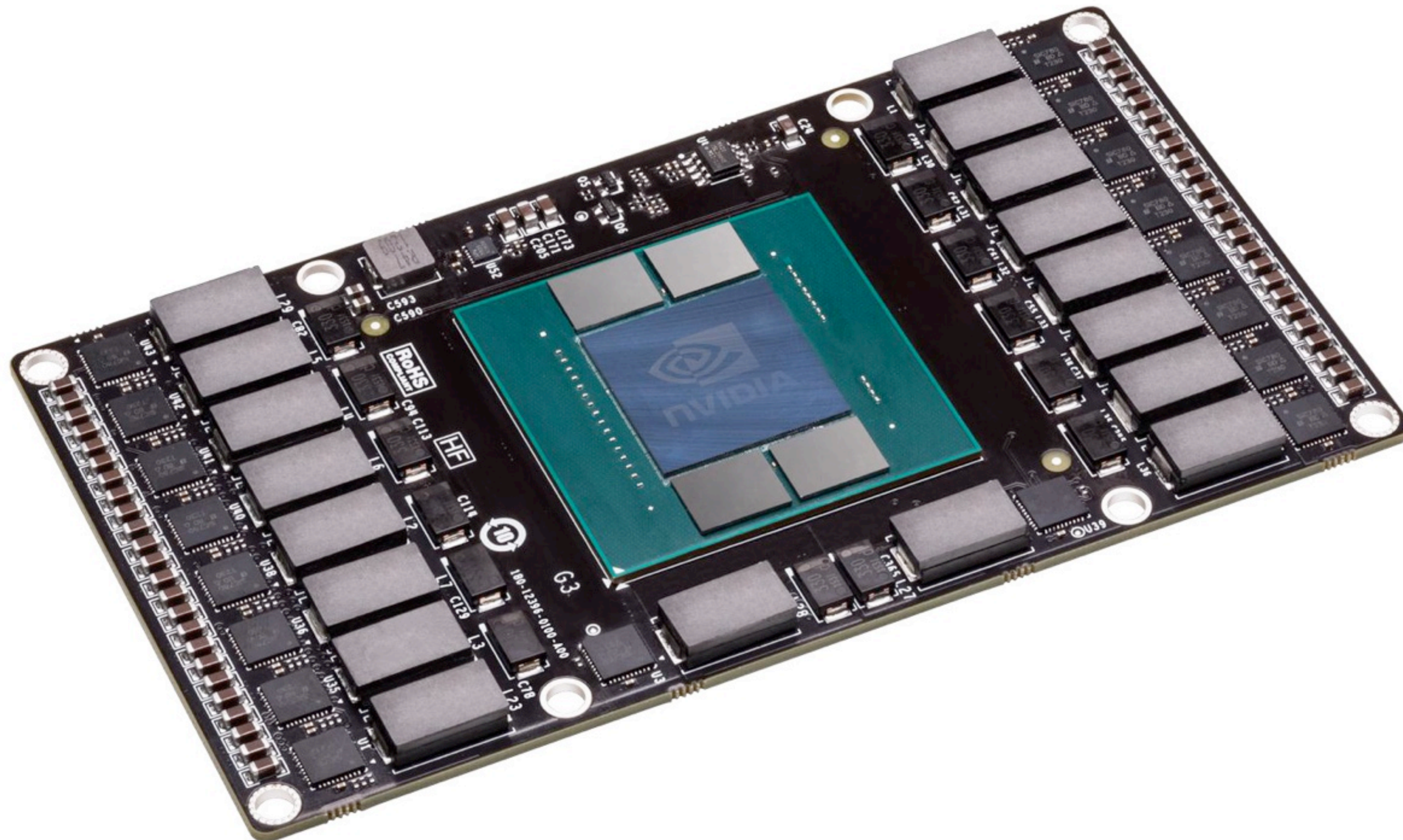
In 2016



A 2016 graphics card: Pascal GPU with Stacked DRAM

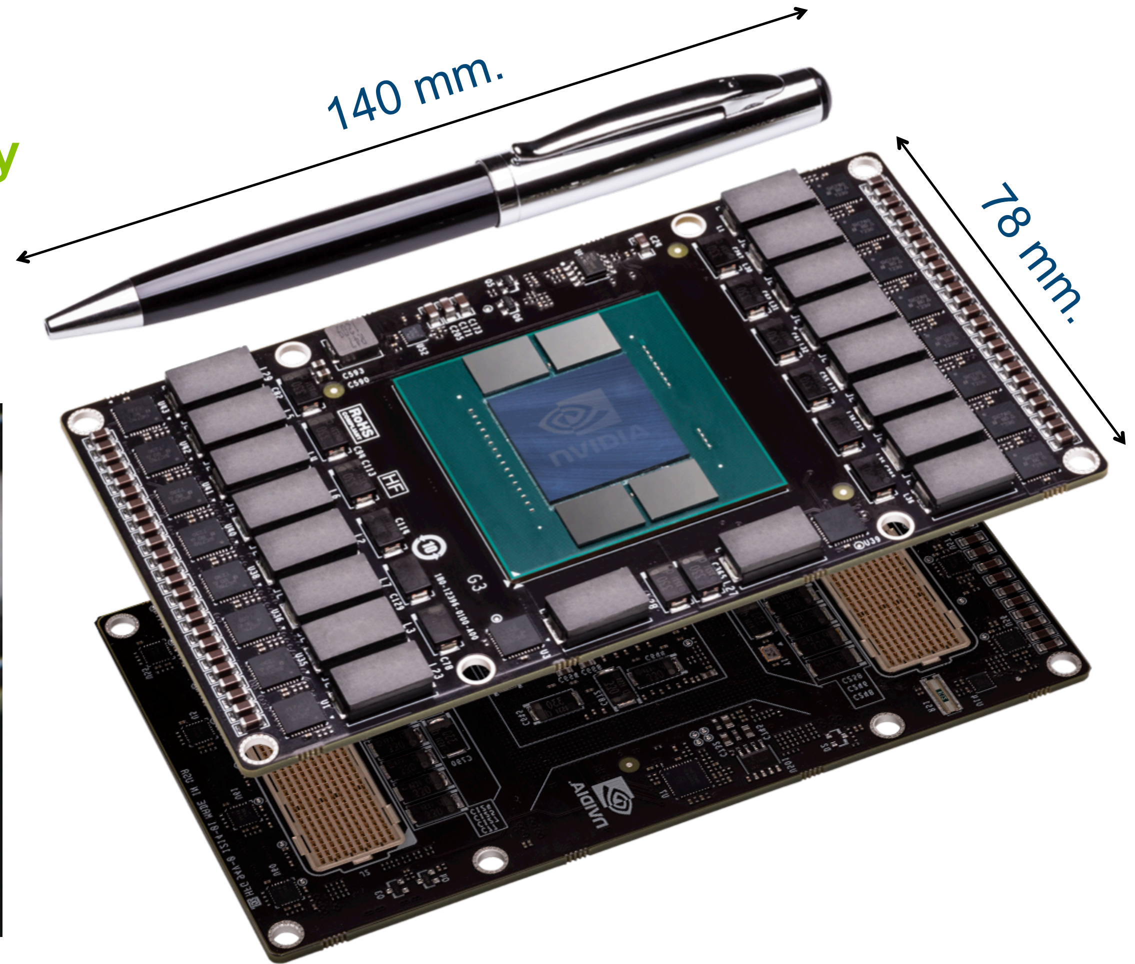


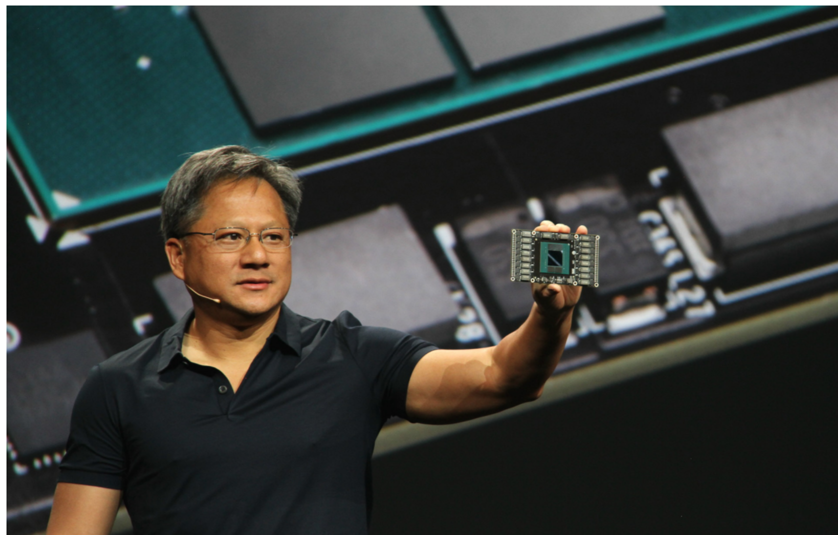
A Pascal GPU prototype



The Pascal GPU prototype: SXM 2.0 Form Factor

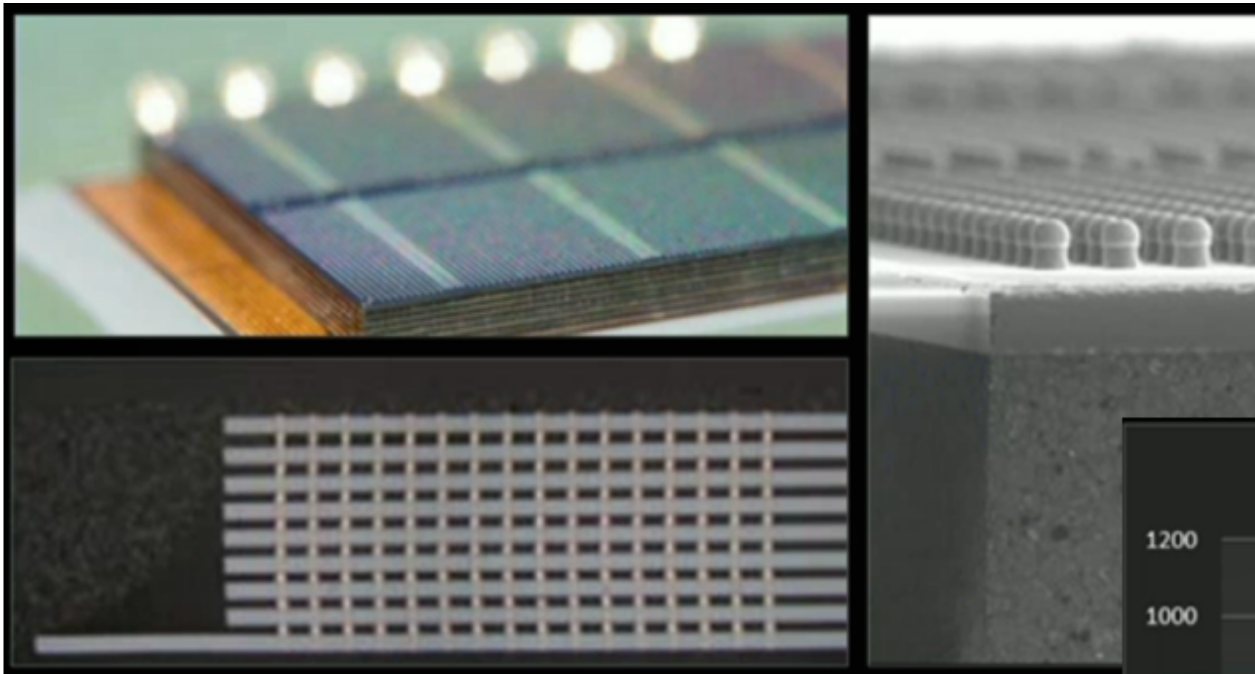
SMX 2.0 *:
3x Performance Density

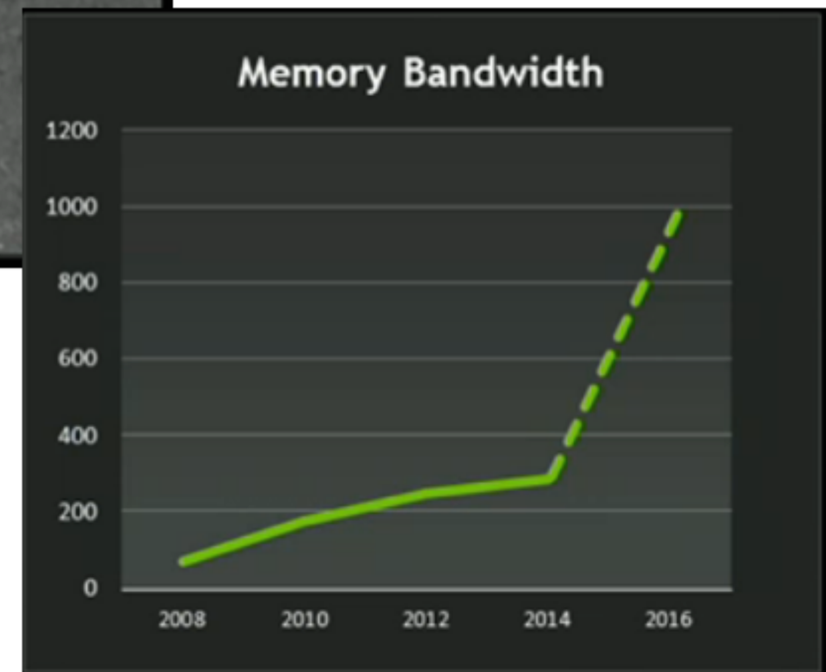




(* Marketing Code Name. Name is not final).

Pascal Stacked DRAM Memory

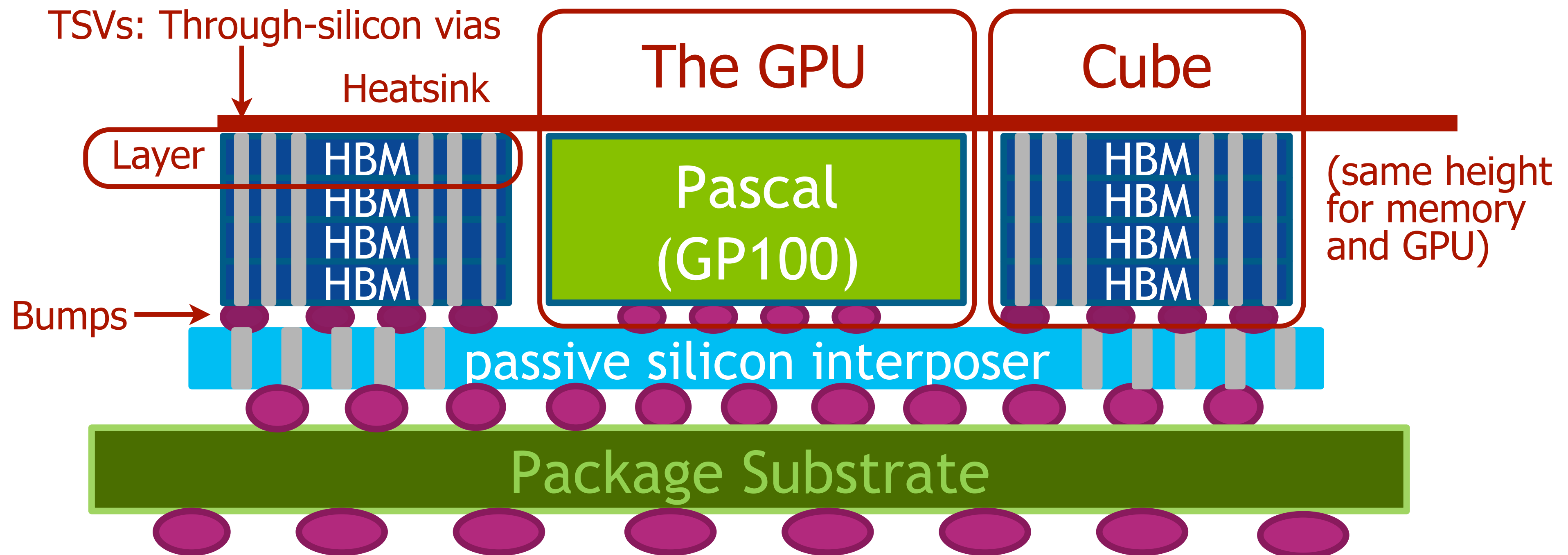




- 3D chip-on-wafer integration.
- 3x bandwidth vs. GDDR5.
- 2.7x capacity vs. GDDR5.
- 4x energy efficient per bit.

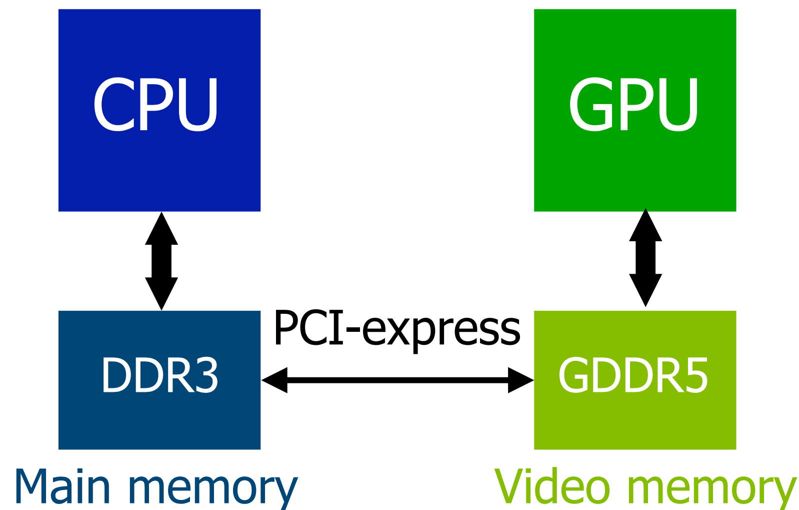
How to break the 1 TB/s bandwidth barrier with a 2x 500 MHz clock

- BW = frequency*width => 1 TB/s = 2x500MHz * width =>
- width = 8000 Gbits/s / 1 GHz = 8000 bits
- Width in Titan X: 384 bits. Max. in GPU history: 512 bits.

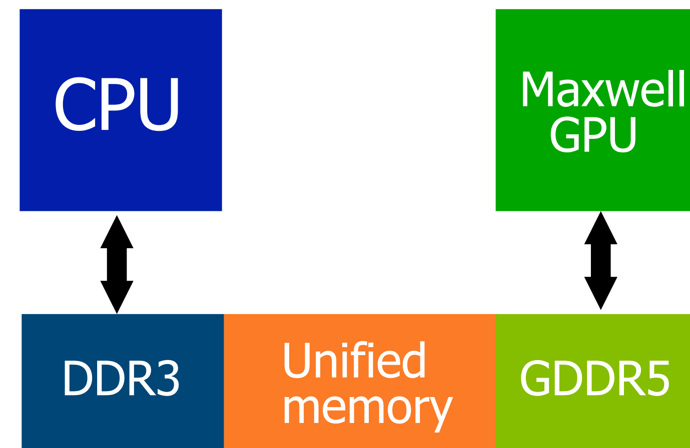


- There is an interconnection hierarchy!

Unified memory: Encourage the programmer NOW to see the FUTURE memory



The old hardware and software model: Different memories, performances and address spaces.

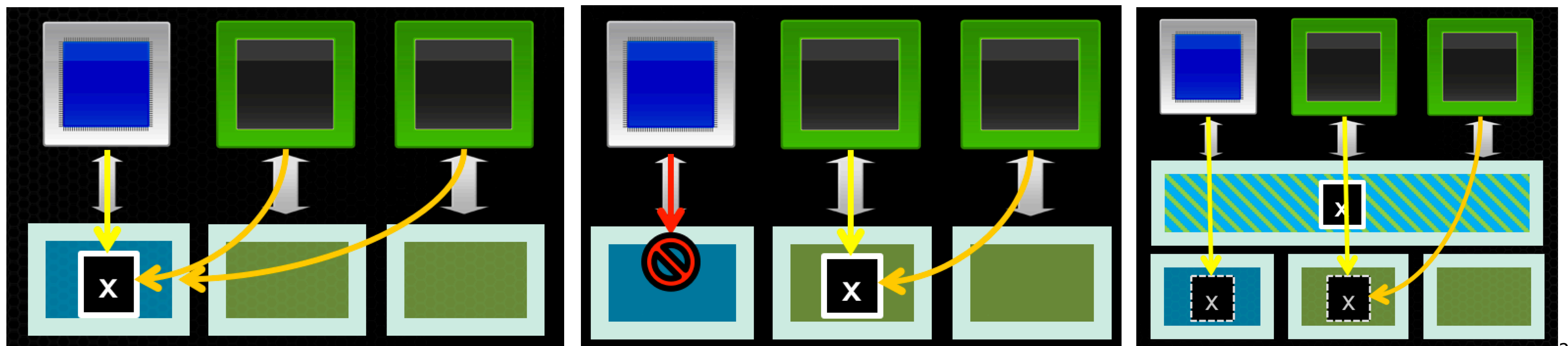


The new API: Same memory, a single global address space.

Performance is sensitive to data proximity.

CUDA memory types

	Zero-Copy (pinned memory)	Unified Virtual Addressing	Unified Memory
CUDA call	cudaMallocHost(&A, 4);	cudaMalloc(&A, 4);	cudaMallocManaged(&A, 4);
Allocation fixed in	Main memory (DDR3)	Video memory (GDDR5)	Both
Local access for	CPU	Home GPU	CPU and home GPU
PCI-e access for	All GPUs	Other GPUs	Other GPUs
Other features	Avoid swapping to disk	No CPU access	On access CPU/GPU migration
Coherency	At all times	Between GPUs	Only at launch & sync.
Full support in	CUDA 2.2	CUDA 1.0	CUDA 6.0



Example 1: Sorting elements from a file. The programming style converges with C

CPU code in C	GPU code in CUDA (v. 6.0 on)

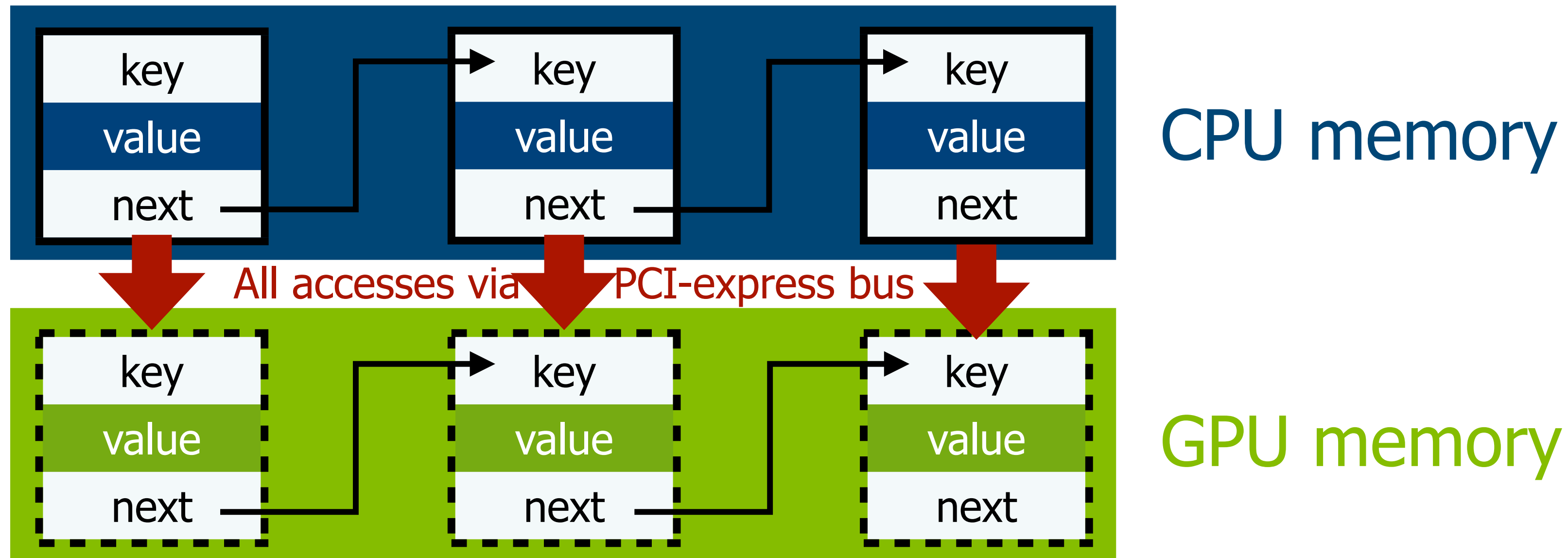
CPU code in C:

```
void sortfile (FILE *fp, int N)
{
  char *data;
  data = (char *) malloc(N);

  fread(data, 1, N, fp);

  qsort(data, N, 1, compare);

  use_data(data);

  free(data);
}
```

GPU code in CUDA (v. 6.0 on):

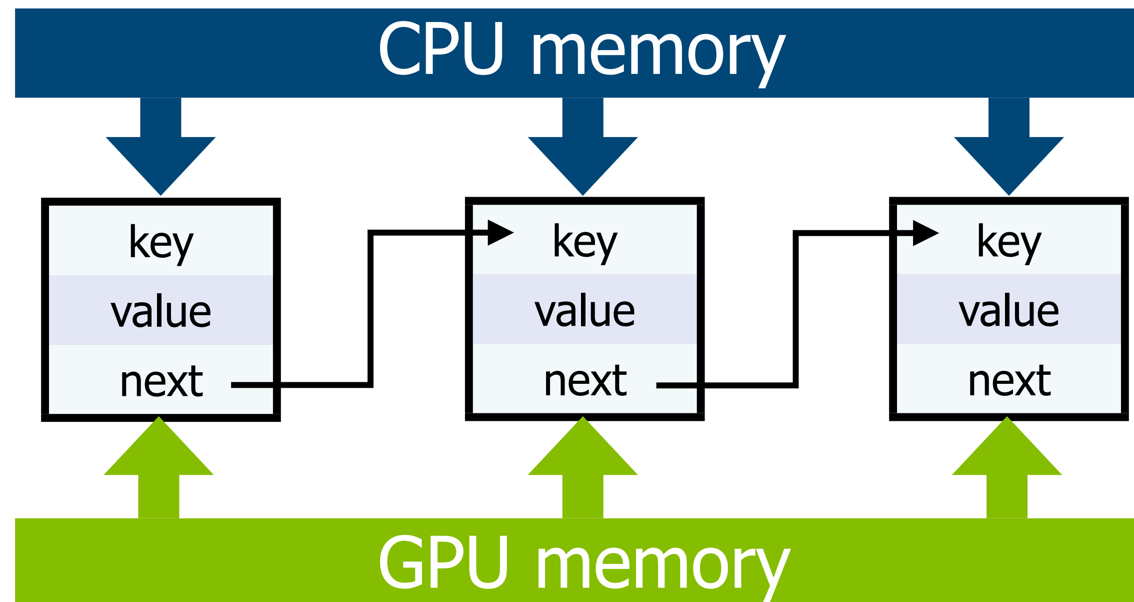
```
void sortfile (FILE *fp, int N)
{
  char *data;
  cudaMallocManaged(&data, N);

  fread(data, 1, N, fp);

  qsort<<<...>>>(data, N, 1, compare);
  cudaDeviceSynchronize();
  use_data(data);

  cudaFree(data);
}
```

Example 2: Linked lists



- Almost impossible to manage in the original CUDA API.
- The best you can do is use pinned memory:
  - Pointers are global: Just as unified memory pointers.
  - Performance is low: GPU suffers from PCI-e bandwidth.
  - GPU latency is very high, which is critical for linked lists because of the intrinsic pointer chasing.

Linked lists with unified memory



- Can pass list elements between CPU & GPU.
  - No need to move data back and forth between CPU and GPU.
- Can insert and delete elements from CPU or GPU.
  - But program must still ensure no race conditions (data is coherent between CPU & GPU at kernel launch only).

Unified memory: Summary

- Drop-in replacement for `cudaMalloc()` using `cudaMallocManaged()`.
  - `cudaMemcpy()` now optional.
- Greatly simplifies code porting.
  - Less Host-side memory management.
- Enables shared data structures between CPU & GPU
  - Single pointer to data = no change to data structures.
- Powerful for high-level languages like C++.

Unified memory: The roadmap. Contributions on every abstraction level

Abstraction level	Past: Consolidated in 2014	Present: On the way during 2015	Future: Available in coming years
High	Single pointer to data. No `cudaMemcpy()` is required	Prefetching mechanisms to anticipate data arrival in copies	System allocator unified
Medium	Coherence @ launch & synchronize	Migration hints	Stack memory unified
Low	Shared C/C++ data structures	Additional OS support	Hardware-accelerated coherence

III. Stacked DRAM (3D RAM)

NVIDIA

Stacked DRAM: A tale of two consortiums

- HMCC (Hybrid Memory Cube Consortium).
  - Mentors: Micron and Samsung.
  - http://www.hybridmemorycube.org (HMC 1.0, 1.1, 2.0 already available)
- HBM (High Bandwidth Memory).
  - Mentors: AMD and SK Hynix.
  - https://www.jedec.org/standards-documents/docs/jesd235 (access via JEDEC).
- Keep an eye on what the gurus predict at the end of this year (incoming report by the ITRS):
  - http://www.itrs.net

III.1 HMC (Hybrid Memory Cube)

Hybrid Memory Cube Consortium (HMCC)

HMCC achievements and milestones	Date
First papers published about Stacked DRAM (based on 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
Specification HMC 1.0 available	April, 2013
Production samples based on the standard	Second half of 2014
2.5 configuration available	End of 2014
Specification HMC 2.0 available	2015

Developer members of HMCC (at the time HMC 1.0 was available)

Altera Corporation

ARM

IBM

Micron Technology, Inc

Open-Silicon, Inc.

Samsung Electronics Co., Ltd

SK hynix

Xilinx, Inc.

Founders of the consortium

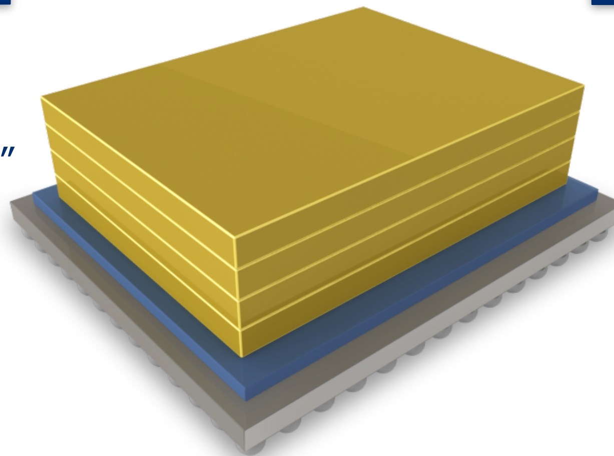
Hybrid Memory Cube at a glance

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.



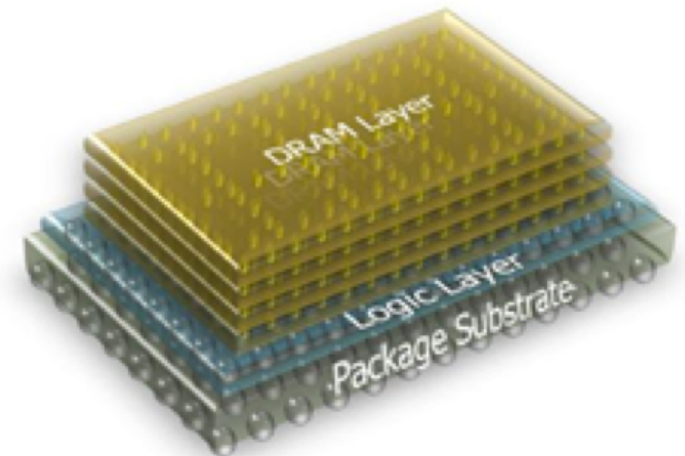
Unparalleled performance

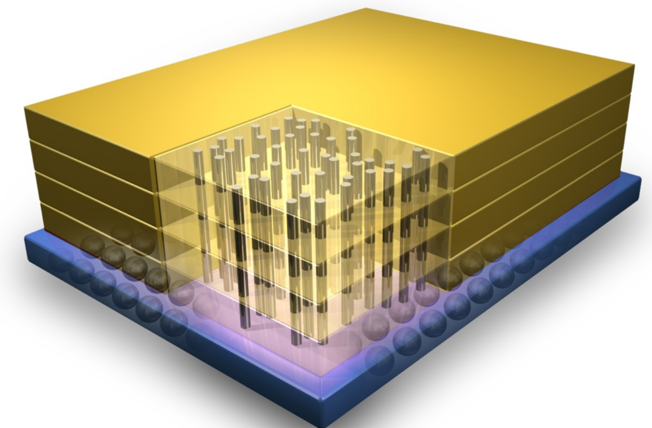
- Up to 15x the bandwidth of a DDR3 module [but just 2x vs. GDDR5].
- 70% less energy usage per bit than existing technologies [measured in number of active signals involved, power savings are 50% only].
- Occupying nearly 90% less space than today's RDIMMs [95% savings].

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 microns with a vertical latency of 12 picoseconds for a Stacked DRAM endowed with 20 layers.





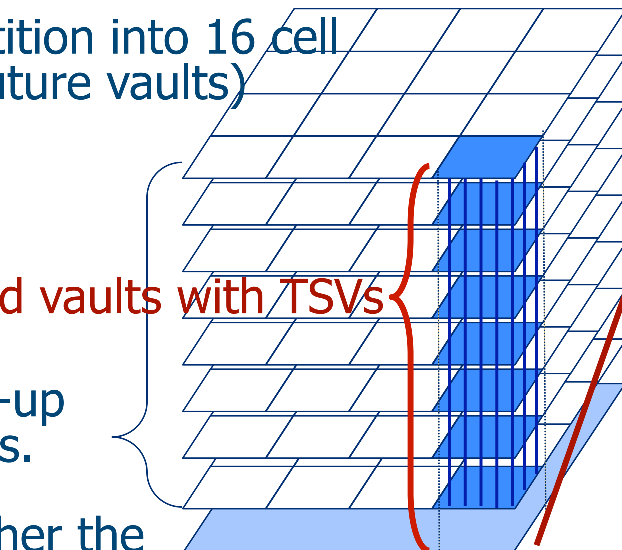
3D integration, side by side with the processor

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

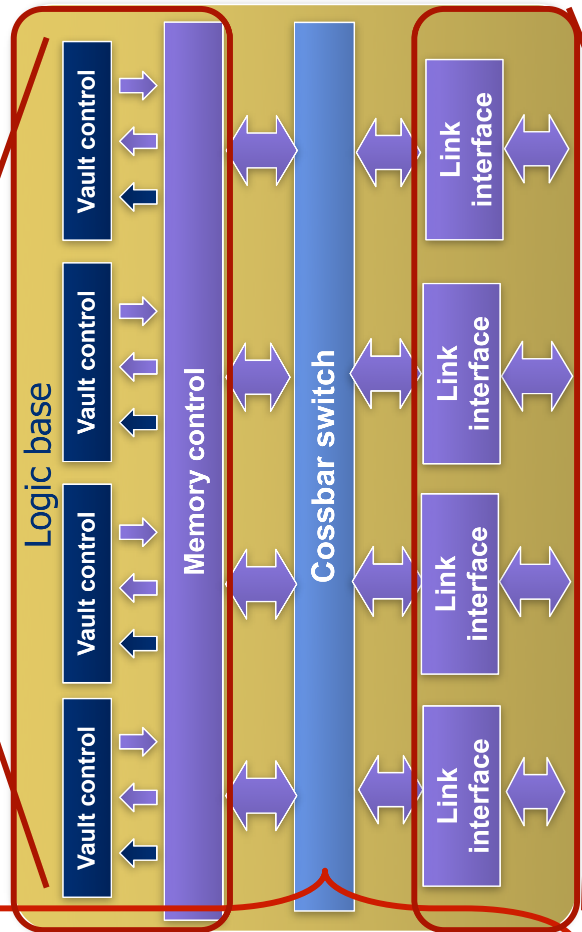
Step 4: Build vaults with TSVs

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.

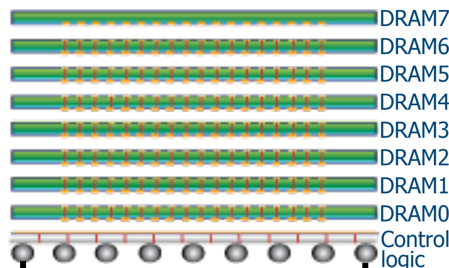


Links to processor(s), which can be another 3D chip, but more heterogeneous:
- Base: CPU and GPU.
- Layers: Cache (SRAM).

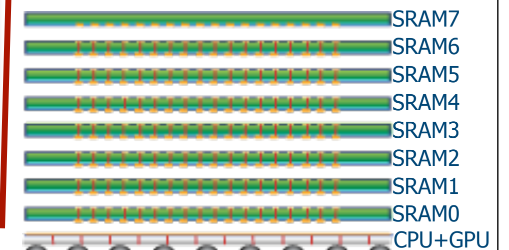
A typical multi-core die uses >50% for SRAM. And those transistors switch slower on lower voltage, so the cache will rely on interleaving over piled-up matrices, just the way DRAM does.

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

3D technology for DRAM memory



3D technology for processor(s)



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 **(70% savings)**

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. **(50% savings)**

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^2		1089 mm^2 per chip
Total area occupied for 640 GB/s.	825 cm^2	412.5 cm^2	43.5 cm^2 **(95% savings)**

III.2. HBM (High Bandwidth Memory)

Why GDDR5 is not enough

- Performance: Scaling has slowed down dramatically and grown exponentially more expensive in the last few years.
- Power:
  - Already in the non-efficient region of power/performance chart.
  - It requires much more energy to increase the BW that it used to.

Case study	Video memory	Bandwidth	Bandwidth per watt	Total power consumed
AMD Radeon R9 290X	GDDR5	320 GB/s	10 GB/s	32 W.
AMD Fiji	HBM	512 GB/s	35 GB/s	15 W.

- Space:
  - 4 chips of 256 MB occupy 672 mm^2.
  - Using HBM, 1 GB occupies only 35 mm^2 (5%).
- Silicon interposer is required to benefit from wire density.

The bandwidth battle: HBM vs. DDR3 and GDDR5

	DDR3	GDDR5	HBM1	HBM2
Pins for data	8 per chip	32 per chip	2 x 128 per layer	2 x 128 per layer
Prefetching (per pin)	8	8	2	2
Access granularity (product of the last two rows)	8 bytes per chip	32 bytes per chip	64 bytes per layer	64 bytes per layer
Bandwidth (per chip or layer)	2 GB/s (2 Gbps/pin)	28 GB/s (7 Gbps/pin)	32 GB/s (1 Gbps/pin)	64 GB/s (2 Gbps/pin)
Chips or layers	8 chips/module	12 chips/card	4 layers/cube	4 or 8 layers/cube
Cubes per GPU	-	-	4	4
Total GPU bandwidth	Typical CPU: 2 GB/s. * 8 chips * 4 channels = **64 GB/s**	Maxwell Titan X: 28 GB/s * 12 chips = **336 GB/s (the end)**	AMD's Fiji: 32 GB/s * 4 layers * 4 cubes = **512 GB/s (the beginning)**	64 GB/s * 4 or 8 layers * 4 cubes = **1 or 2 TB/s**

Pending challenges

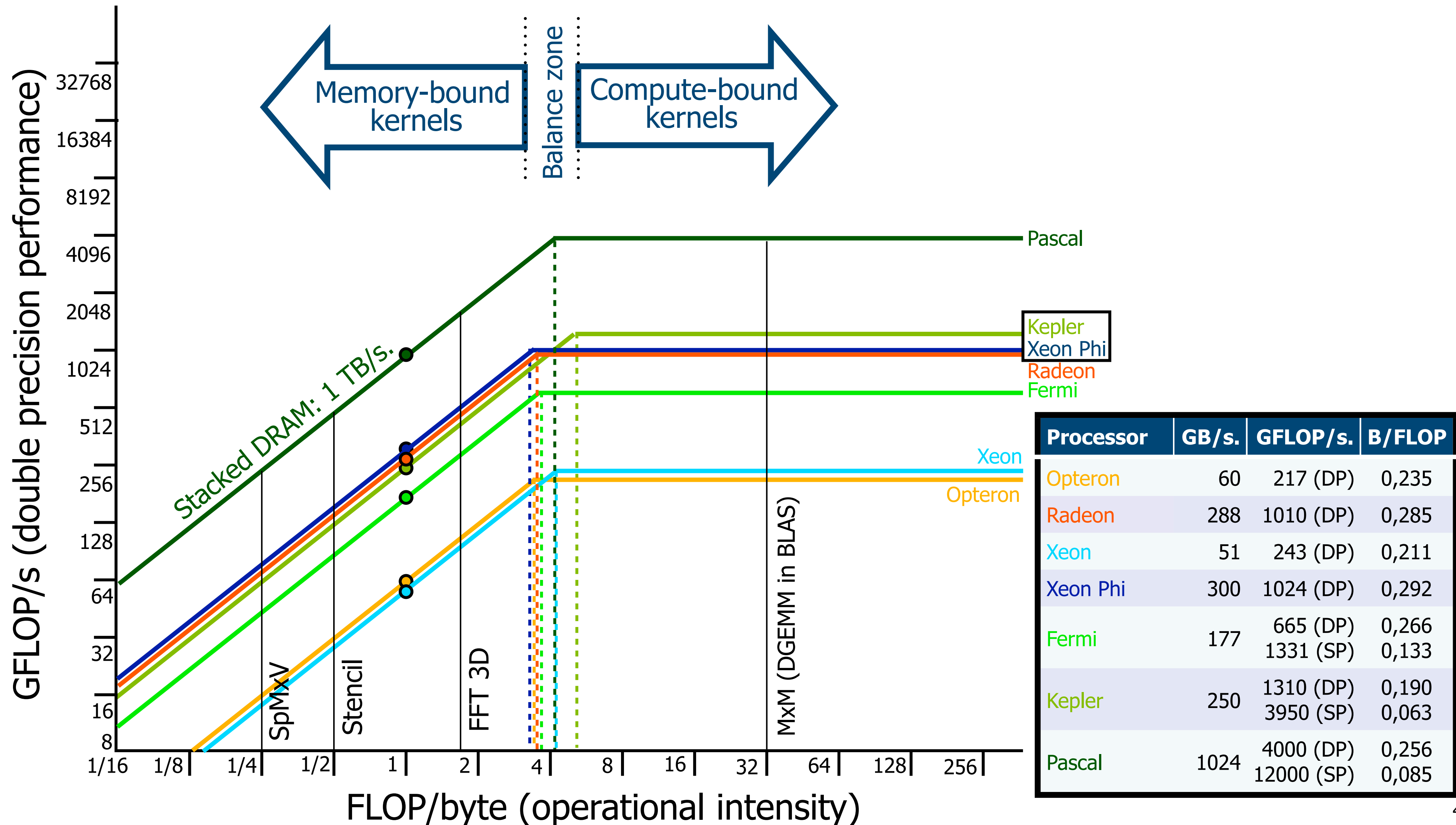
- Competitive cost (hopefully solved on massive sellings).
- Power density: One watt for every 35 GB/s is too much when your goal is to exceed the TB/s barrier.
- Capacity (hopefully solved when 16nm, 10nm and 7nm manufacturing processes contribute).

	HBM1	**HBM2**
Capacity per layer	2 Gbits	8 Gbits
Layers per cube	4	4-8
Capacity per cube	1 GB	4-8 GB
Cubes per GPU	4	4
Total capacity	4 GB	16-32 GB

IV. Impact on GPUs and concluding remarks

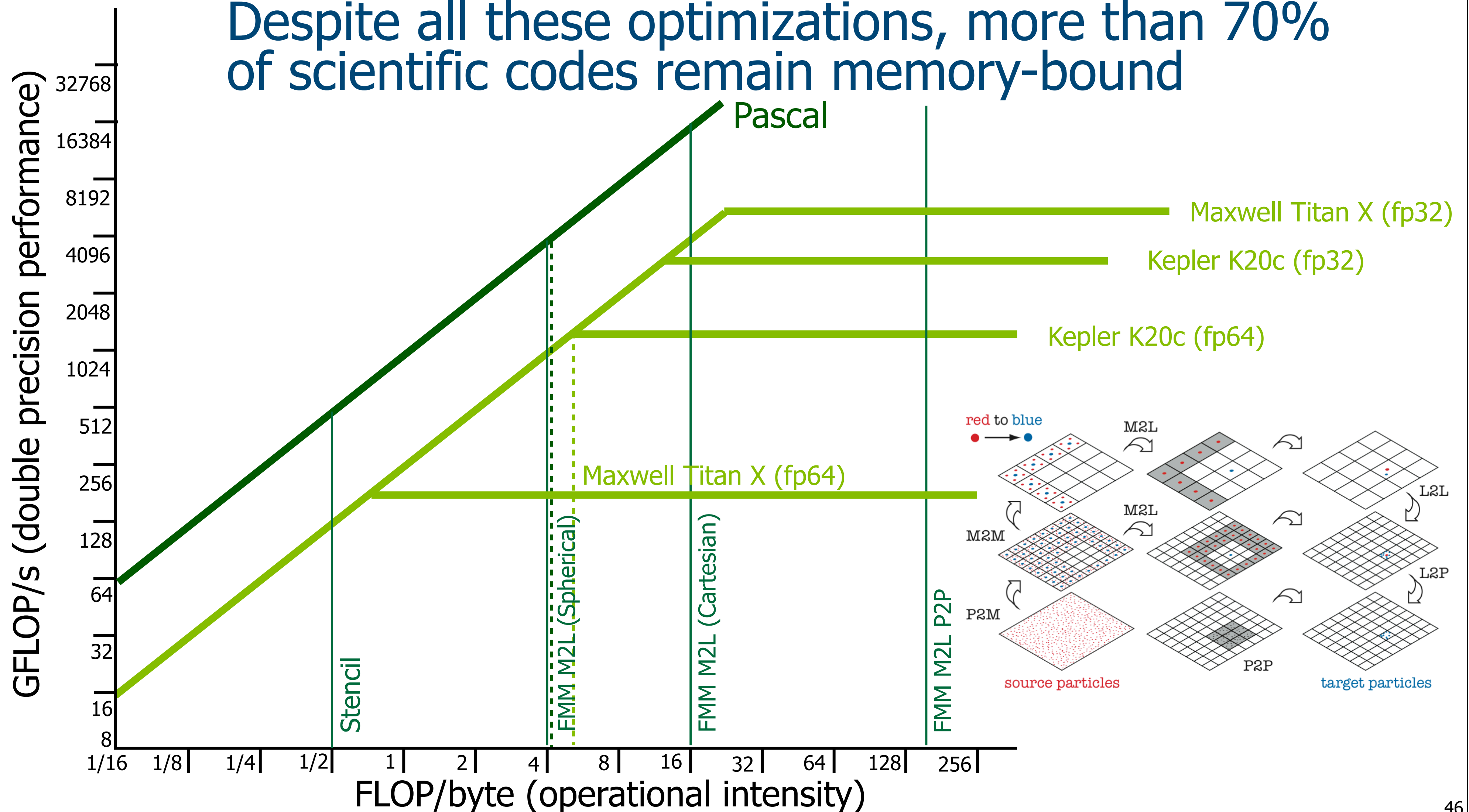
NVIDIA

The Roofline model: Hardware vs. Software



Processor	GB/s.	GFLOP/s.	B/FLOP
Opteron	60	217 (DP)	0,235
Radeon	288	1010 (DP)	0,285
Xeon	51	243 (DP)	0,211
Xeon Phi	300	1024 (DP)	0,292
Fermi	177	665 (DP) 1331 (SP)	0,266 0,133
Kepler	250	1310 (DP) 3950 (SP)	0,190 0,063
Pascal	1024	4000 (DP) 12000 (SP)	0,256 0,085

The Roofline model: Software evolution. Case study: FMM (Fast Multipole Method)



Concluding remarks

- We are facing the heterogeneous era in chips, with better integration of computing and capacity plus an emphasis on buses:
  - TSVs for communicating memory cells faster.
  - Silicon interposers for higher data volume and better scalability.
- GPU programmers can benefit from this technology by adopting unified memory and providing hints to compilers about the way they actually use data.
- HMC and HBM emerge to break the memory wall and promote more hierarchy on interconnections and less hierarchy on memory types.

Acknowledgments & Disclaimer

- To the people at Nvidia, for sharing ideas and slides. And to the company for the sponsorhip to bring me here.
- To Scott Stevens and Susan Platt (Micron) for providing me technical info from the HMC consortium, incorporated to this presentation under explicit permission.
- To Lorena Barba (CUDA Fellow), for her contribution to the FMM example using the roofline model.
- This talk shows my view of emerging technologies as a scientist. It is not an attempt to reflect future plans of Nvidia nor developments on the way (unless explicitly mentioned).