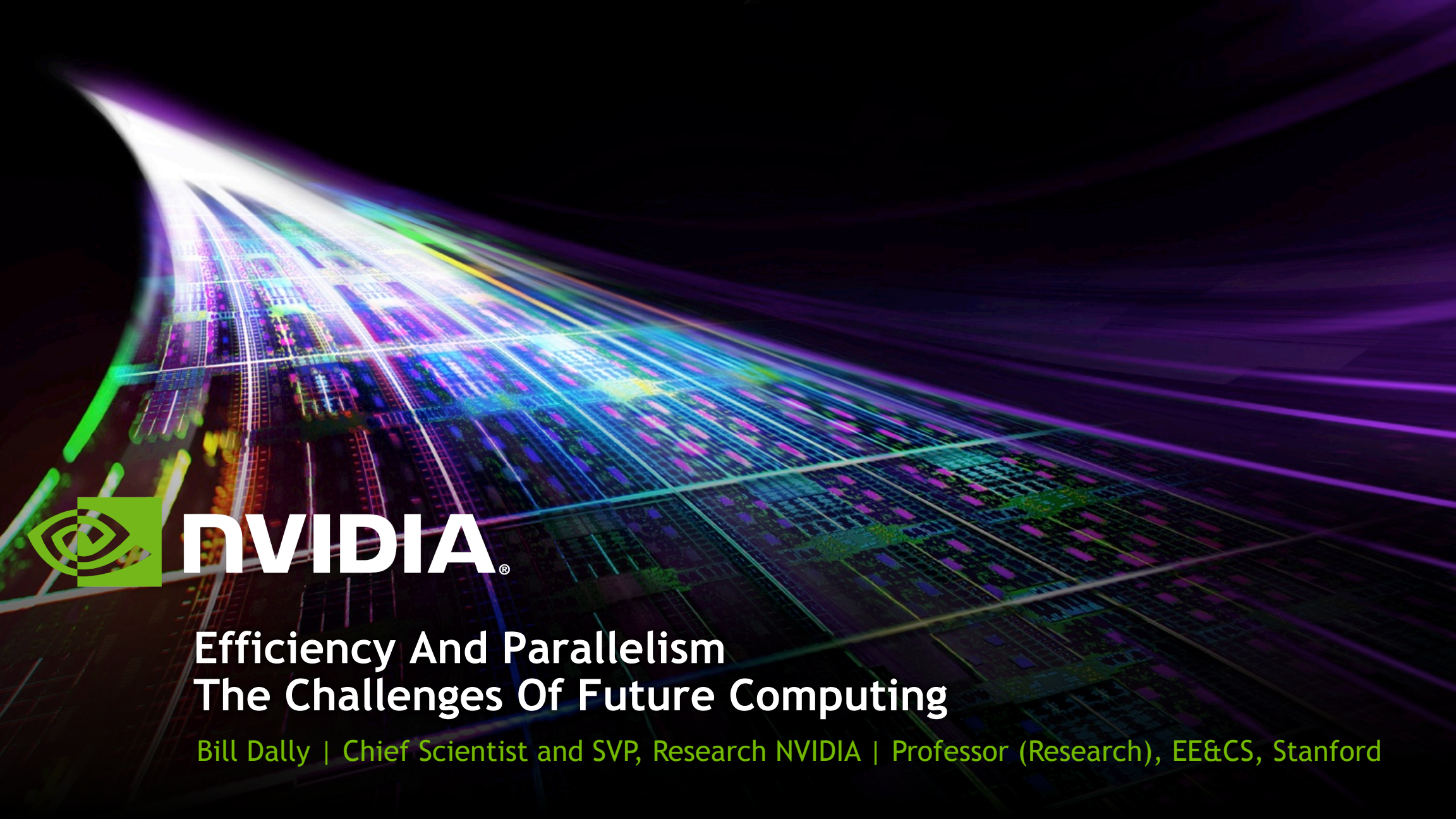


Efficiency And Parallelism
The Challenges Of Future Computing

Bill Dally | Chief Scientist and SVP, Research NVIDIA | Professor (Research), EE&CS, Stanford

Exascale Computing Will Enable Transformational Science

NVIDIA.

Exascale Computing Will Enable Transformational Science

High-Performance Computers are Scientific Instruments

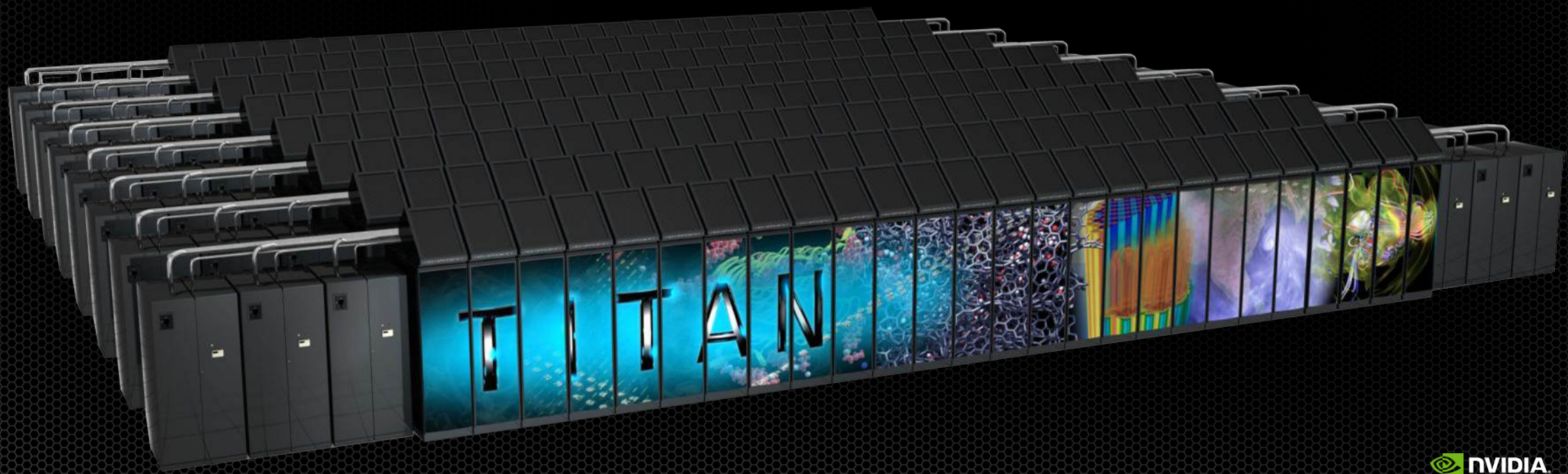
NVIDIA.

TITAN

18,688 NVIDIA Tesla K20X GPUs

27 Petaflops Peak: 90% of Performance from GPUs

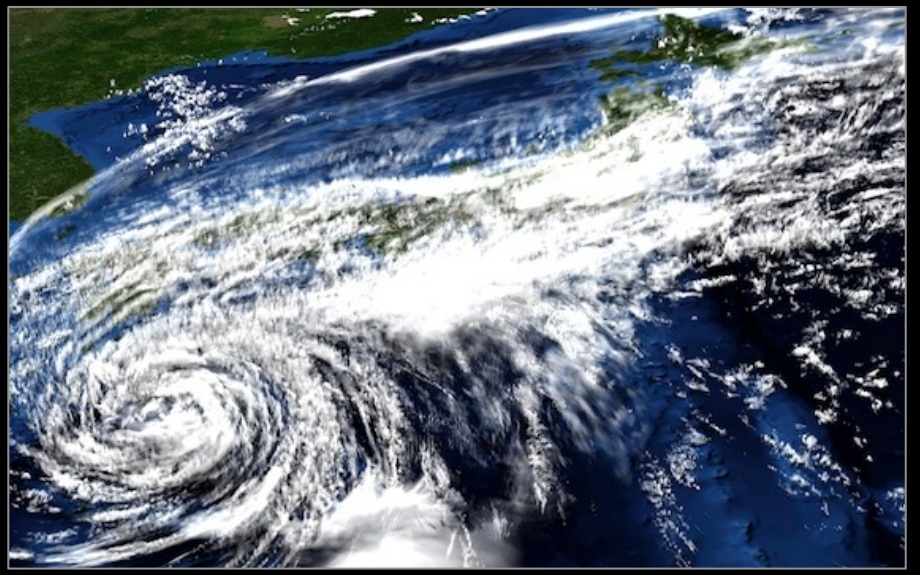
17.59 Petaflops Sustained Performance on Linpack

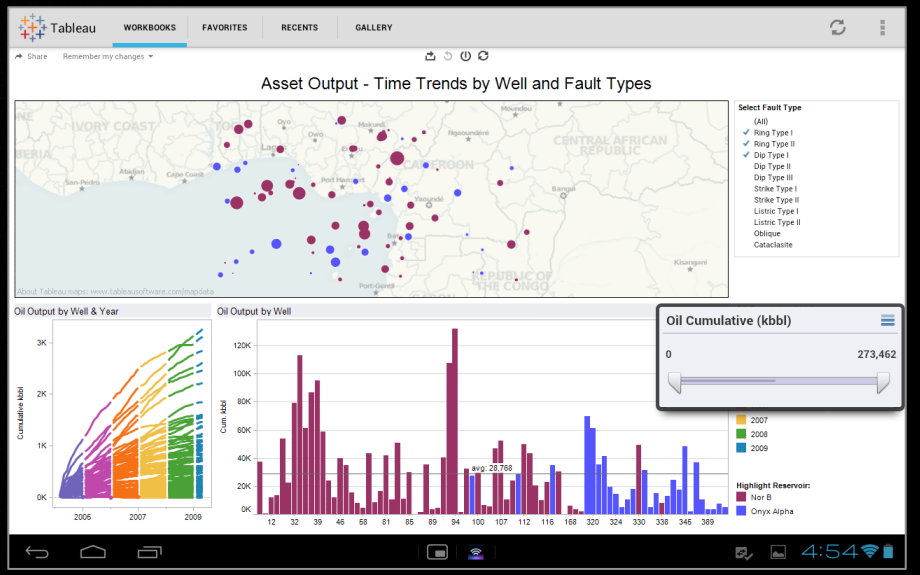


NVIDIA

Scientific Discovery and Business Analytics

Driving an Insatiable Demand for More Computing Performance





NVIDIA

3 Trends

The End of Dennard Scaling

Pervasive Parallelism

Cloud + Mobile

NVIDIA.

3 Trends

The End of Dennard Scaling

Pervasive Parallelism

Cloud + Mobile

NVIDIA.

Moore’s Law doubles processor performance every few years, right?

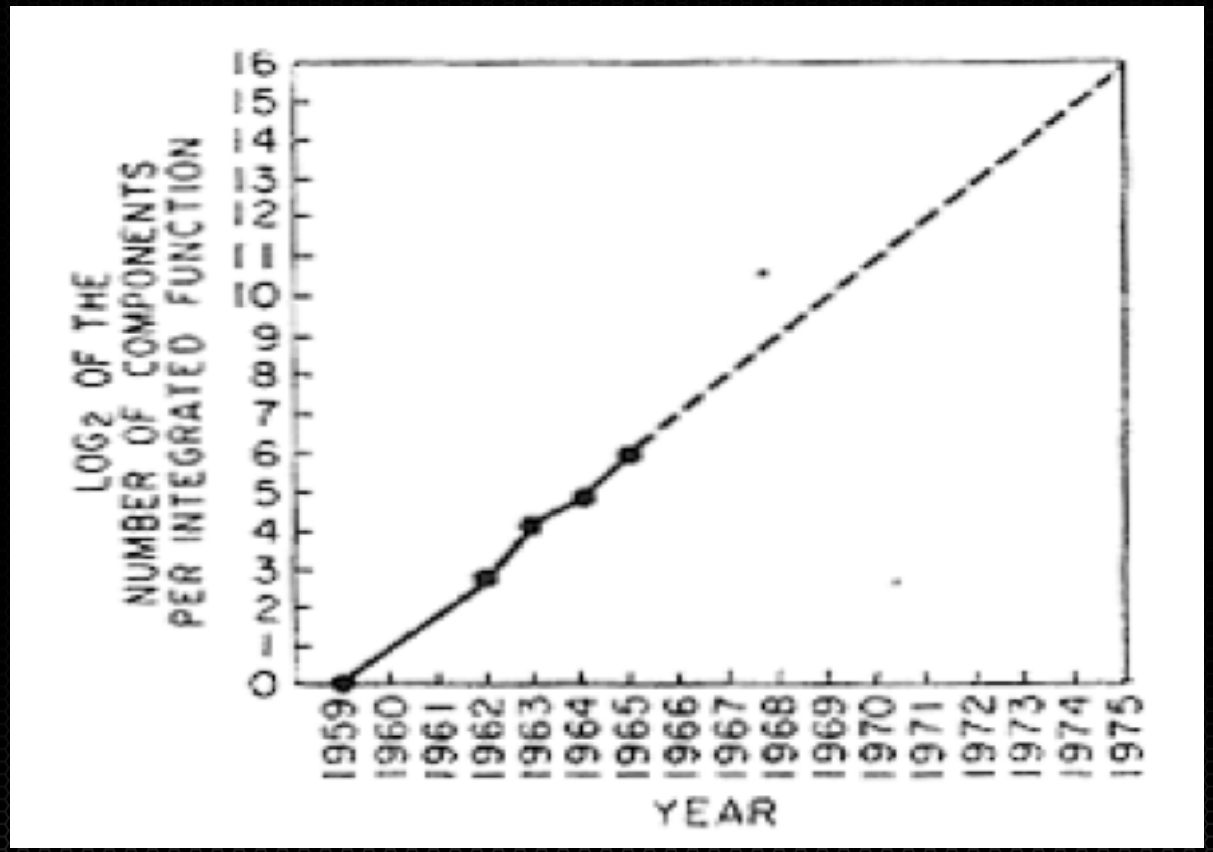
NVIDIA.

Moore’s Law doubles processor performance every few years, right?

Wrong!

NVIDIA.

Moore's Law gives us transistors Which we used to turn into scalar performance



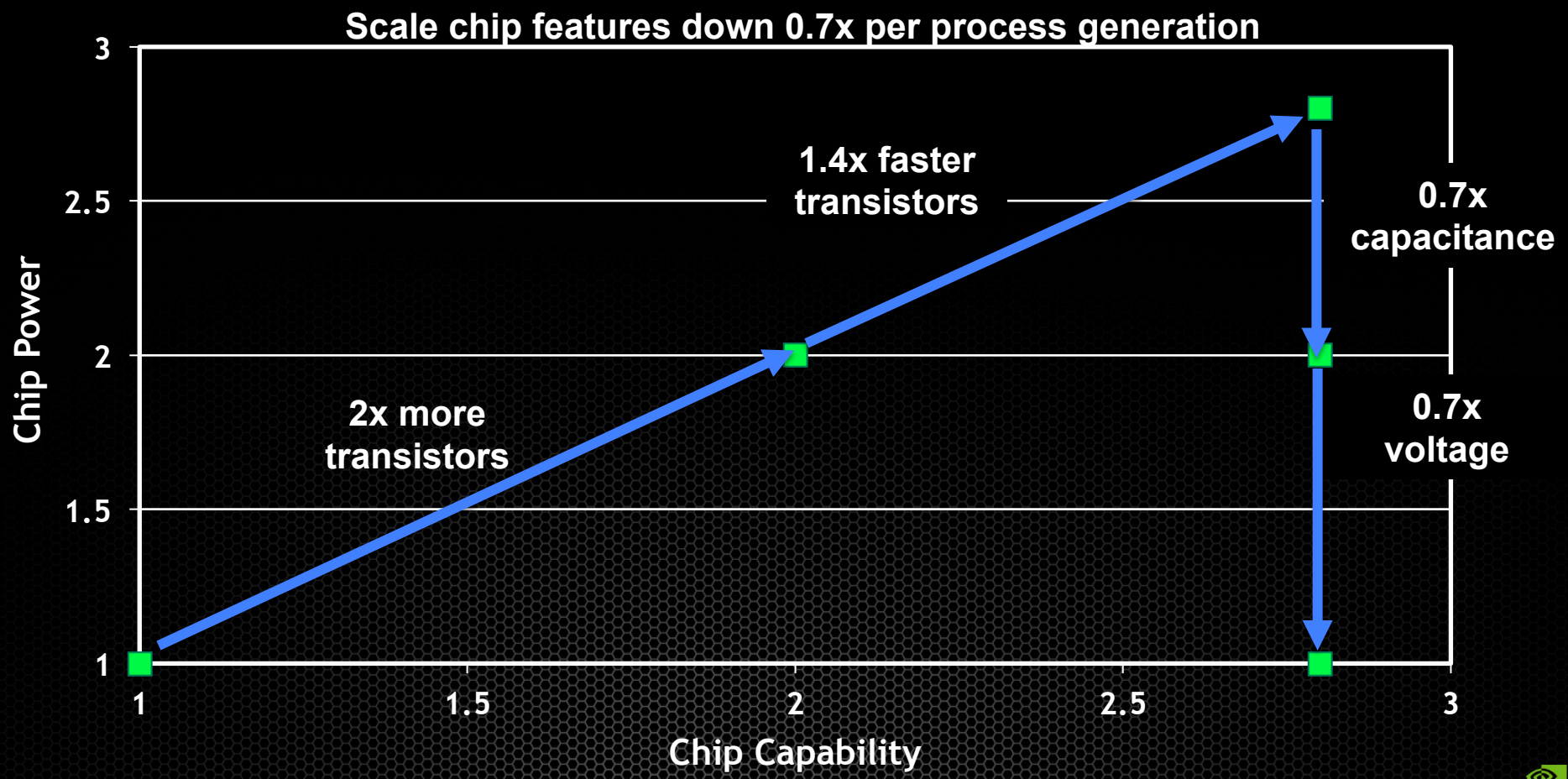
Moore, Electronics 38(8) April 19, 1965

NVIDIA

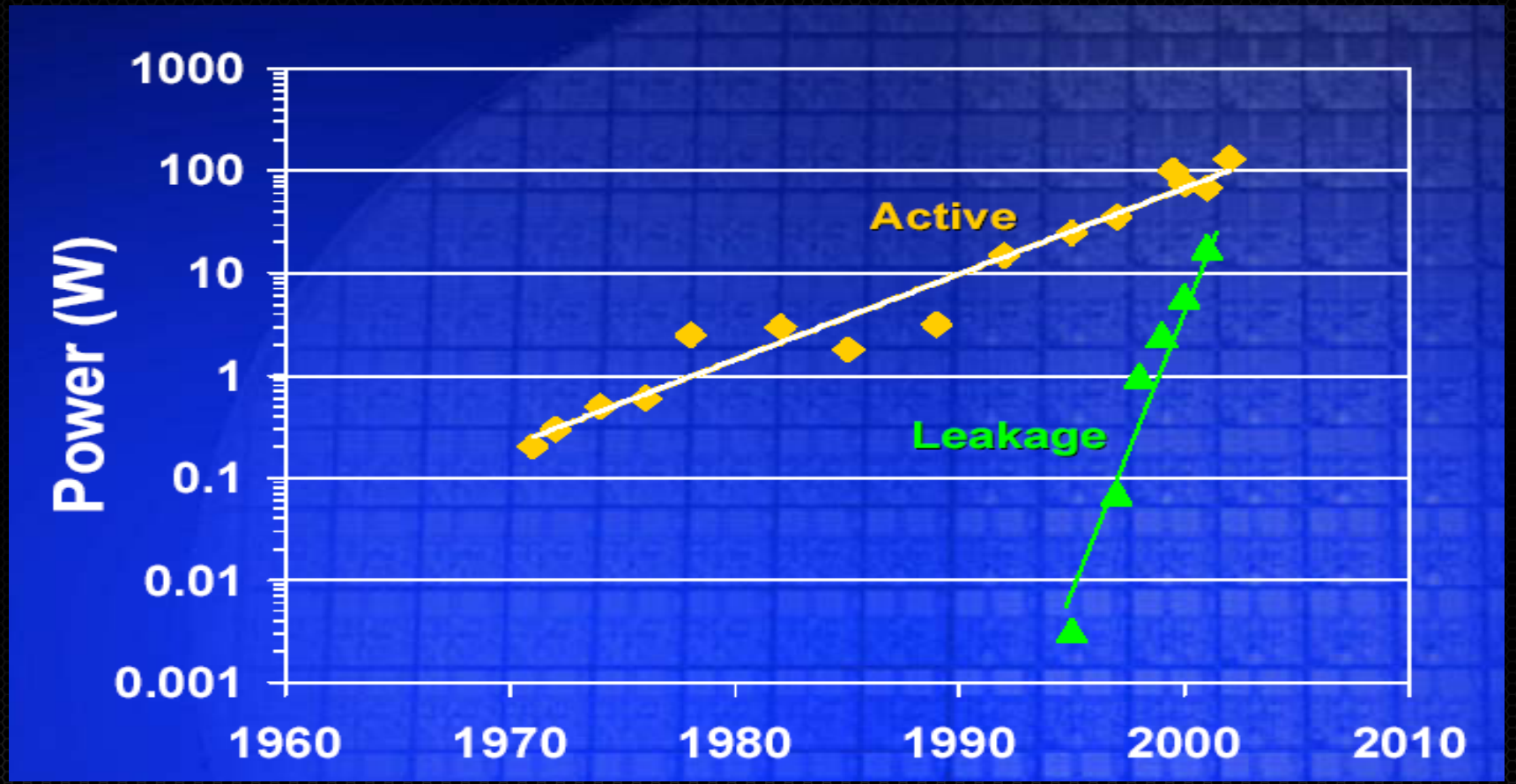
“Moore’s Law gives us more transistors…
Dennard scaling made them useful.”

Bob Colwell, DAC 2013, June 4, 2013

Classic Dennard Scaling
2.8x chip capability in same power



But L^3 energy scaling ended in 2005

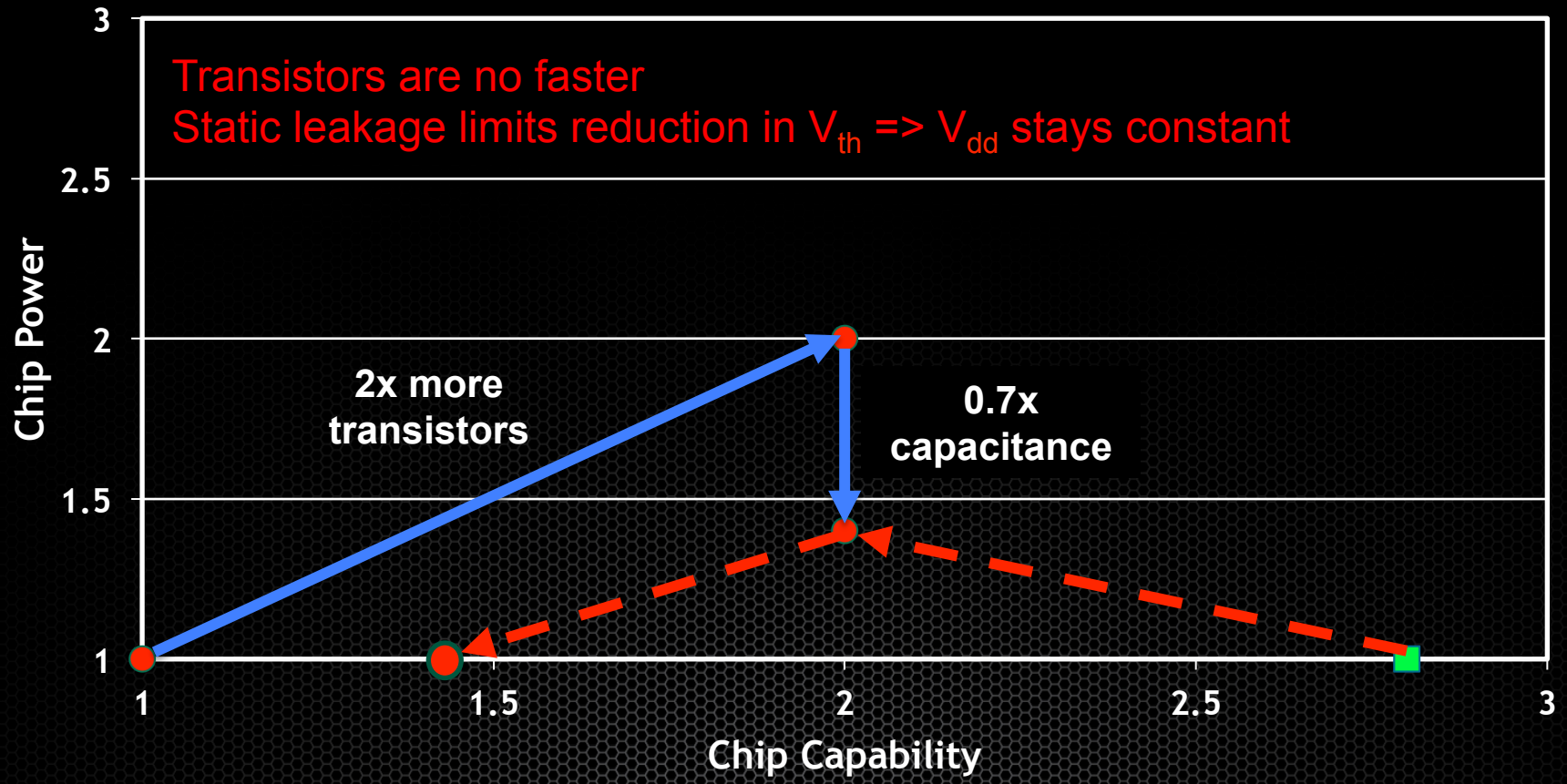


Moore, ISSCC Keynote, 2003

NVIDIA

Post Dennard Scaling

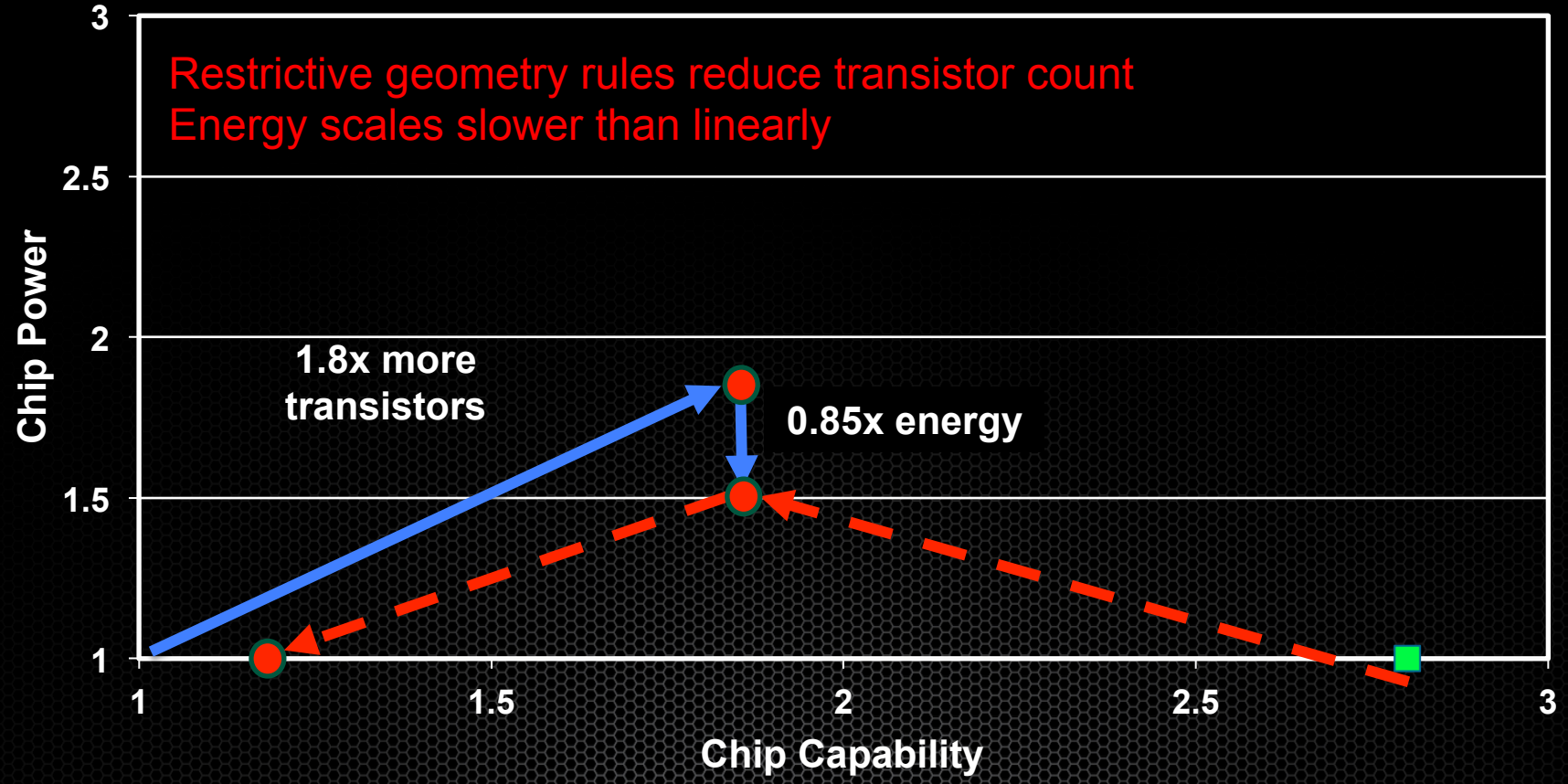
2x chip capability at 1.4x power
1.4x chip capability at same power



NVIDIA.

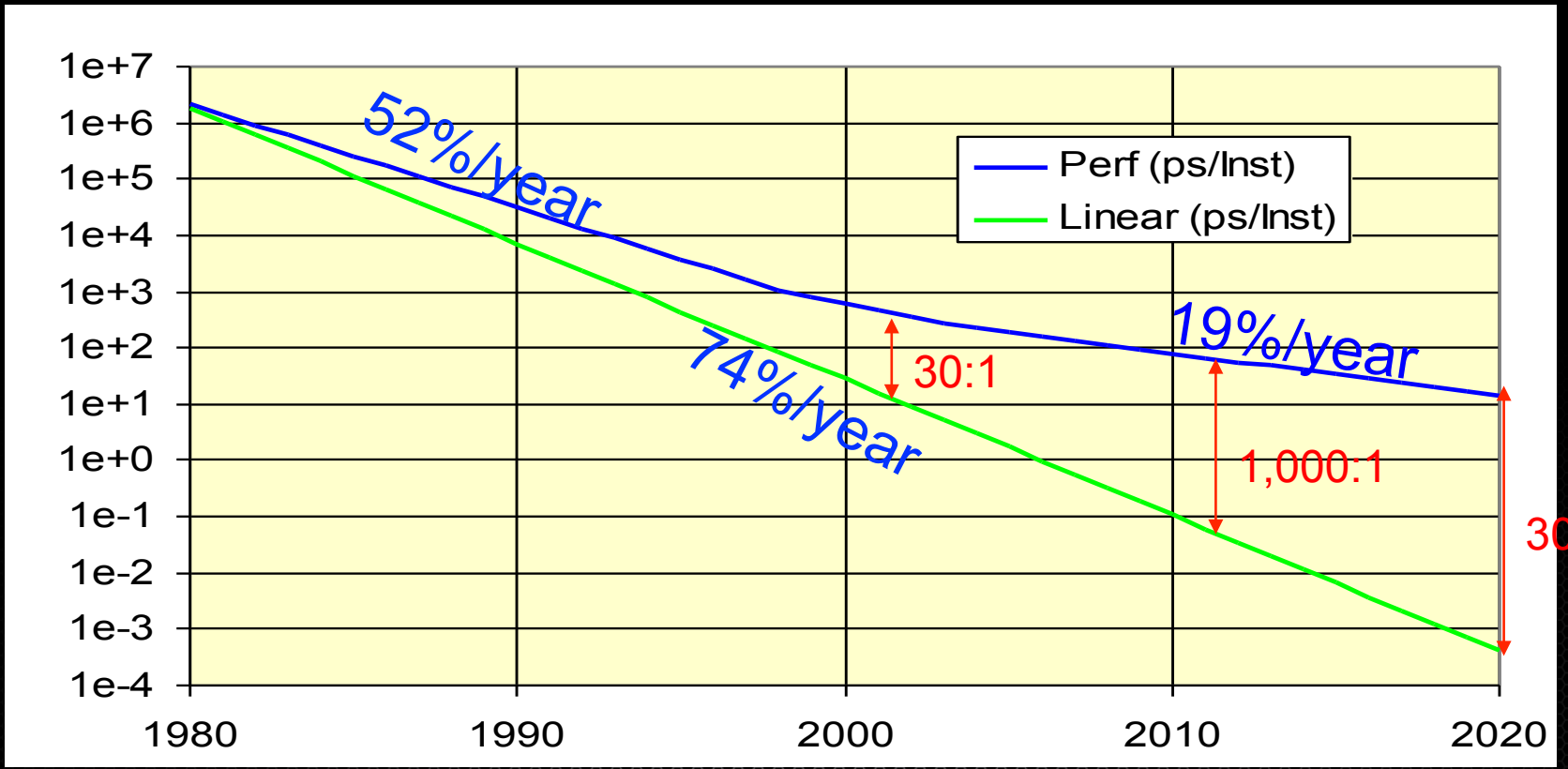
Reality isn't even this good

1.8x chip capability at 1.5x power
1.2x chip capability at same power



NVIDIA

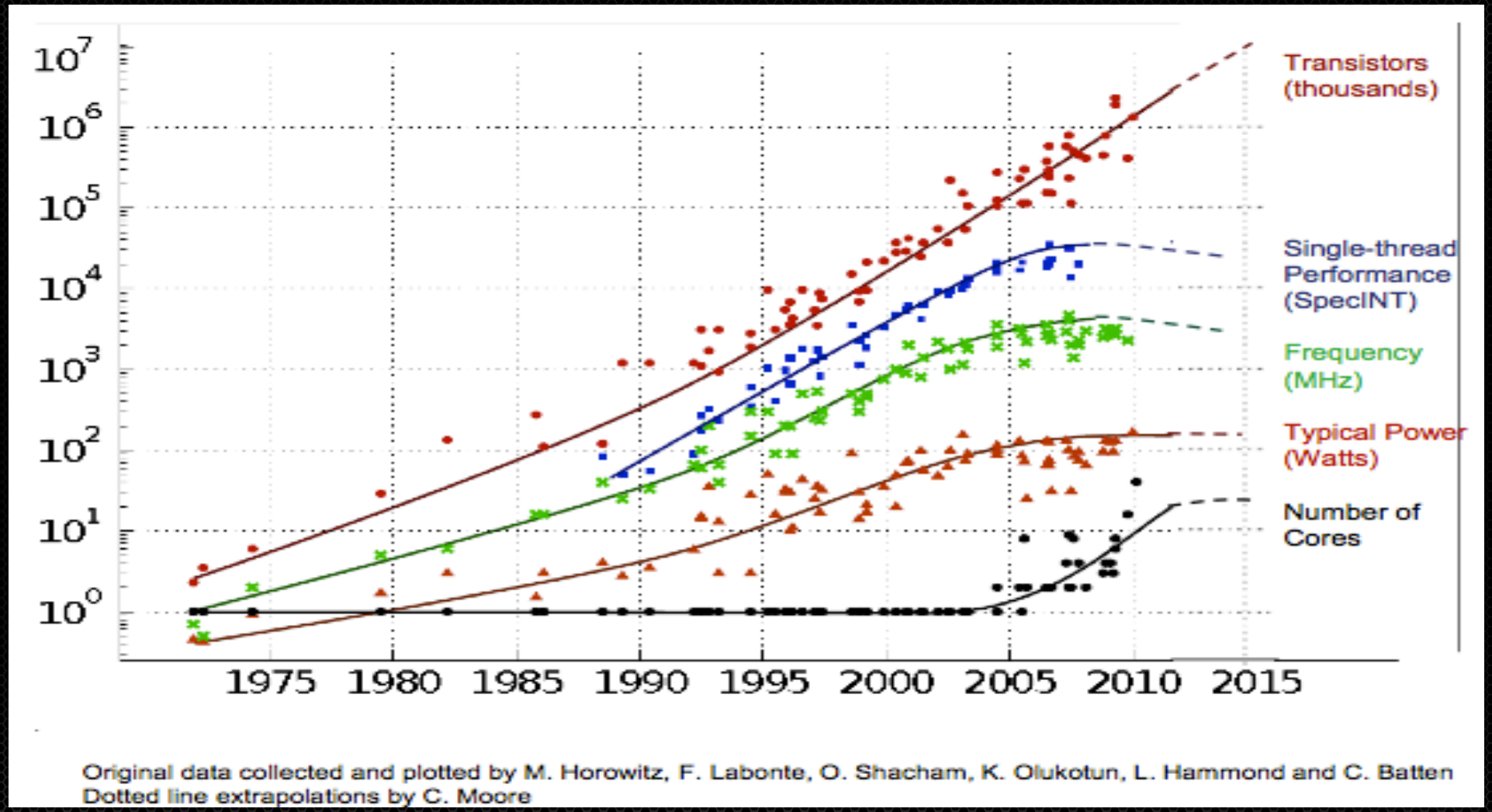
Also, ILP was 'mined out' in 2000



Dally et al. "The Last Classical Computer", ISAT Study, 2001

NVIDIA.

Result: The End of Historic Scaling



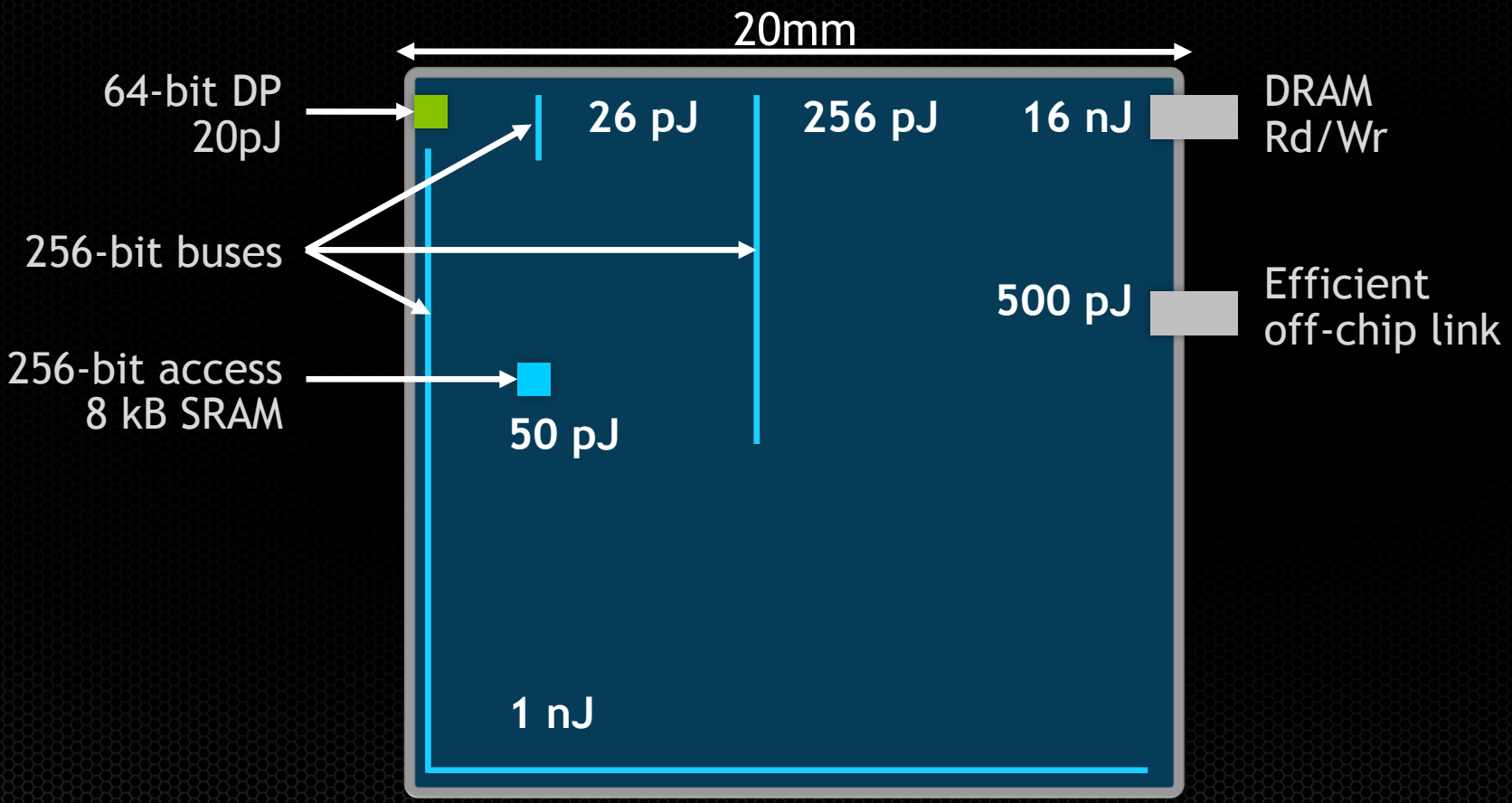
C Moore, *Data Processing in ExaScale-ClassComputer Systems,* Salishan, April 2011

The End of Dennard Scaling

- Processors aren't getting faster, just wider
  - Future gains in performance are from parallelism
- Future systems are energy limited
  - Efficiency ***IS*** Performance
- Process matters less
  - One generation is 1.2x, not 2.8x

NVIDIA.

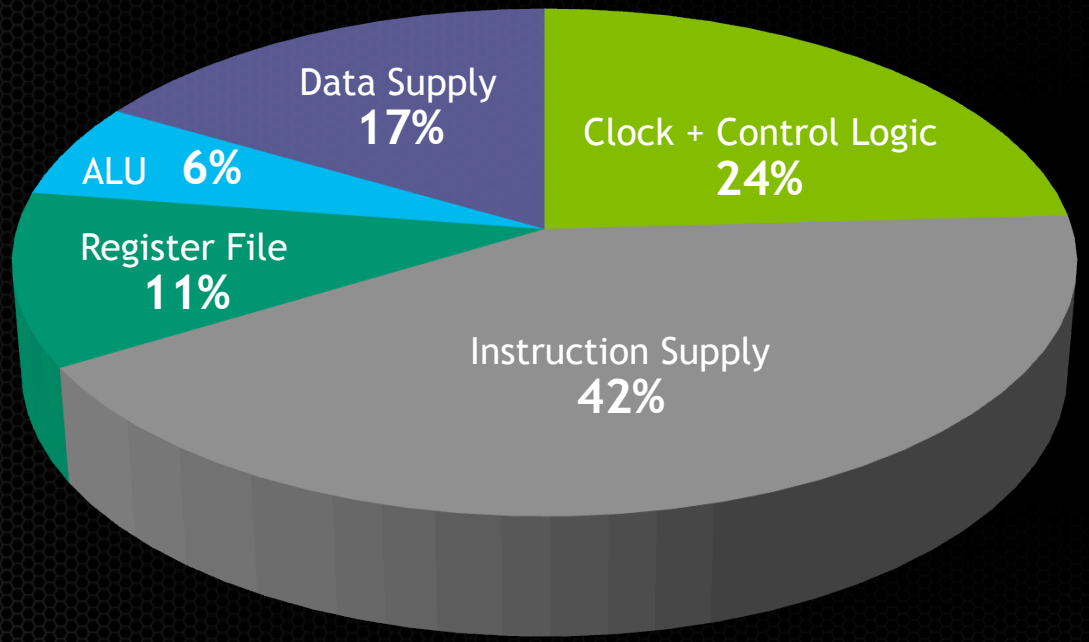
Overhead and Communication Dominate Energy



NVIDIA

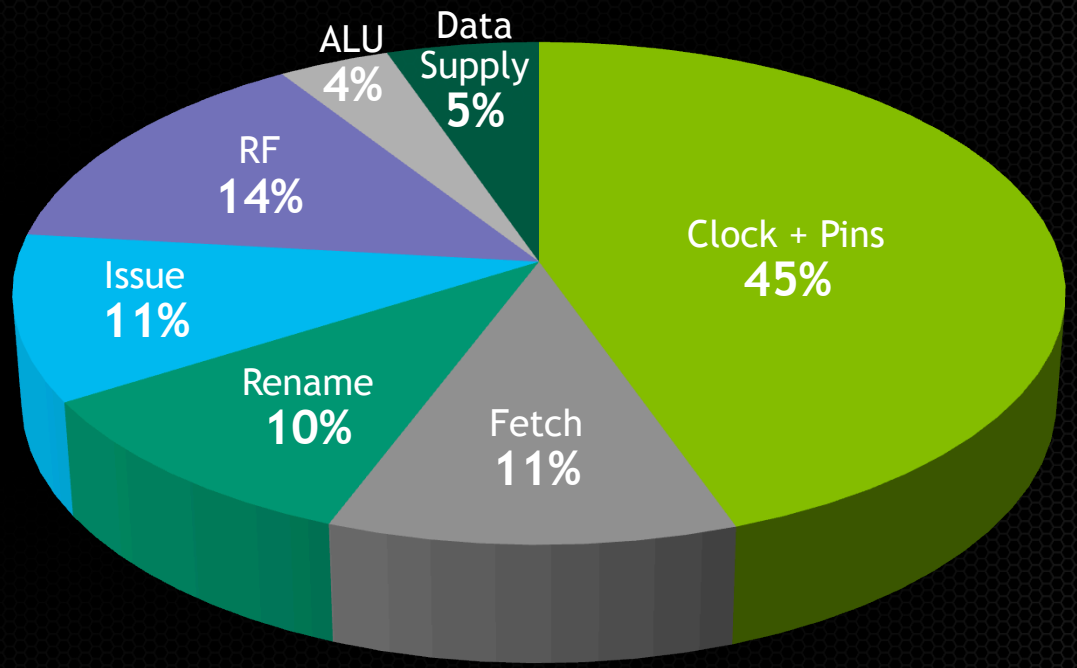
How is Power Spent in a CPU?

In-order Embedded



Dally [2008] (Embedded in-order CPU)

OOO Hi-perf



Natarajan [2003] (Alpha 21264)

NVIDIA.

Energy Shopping List

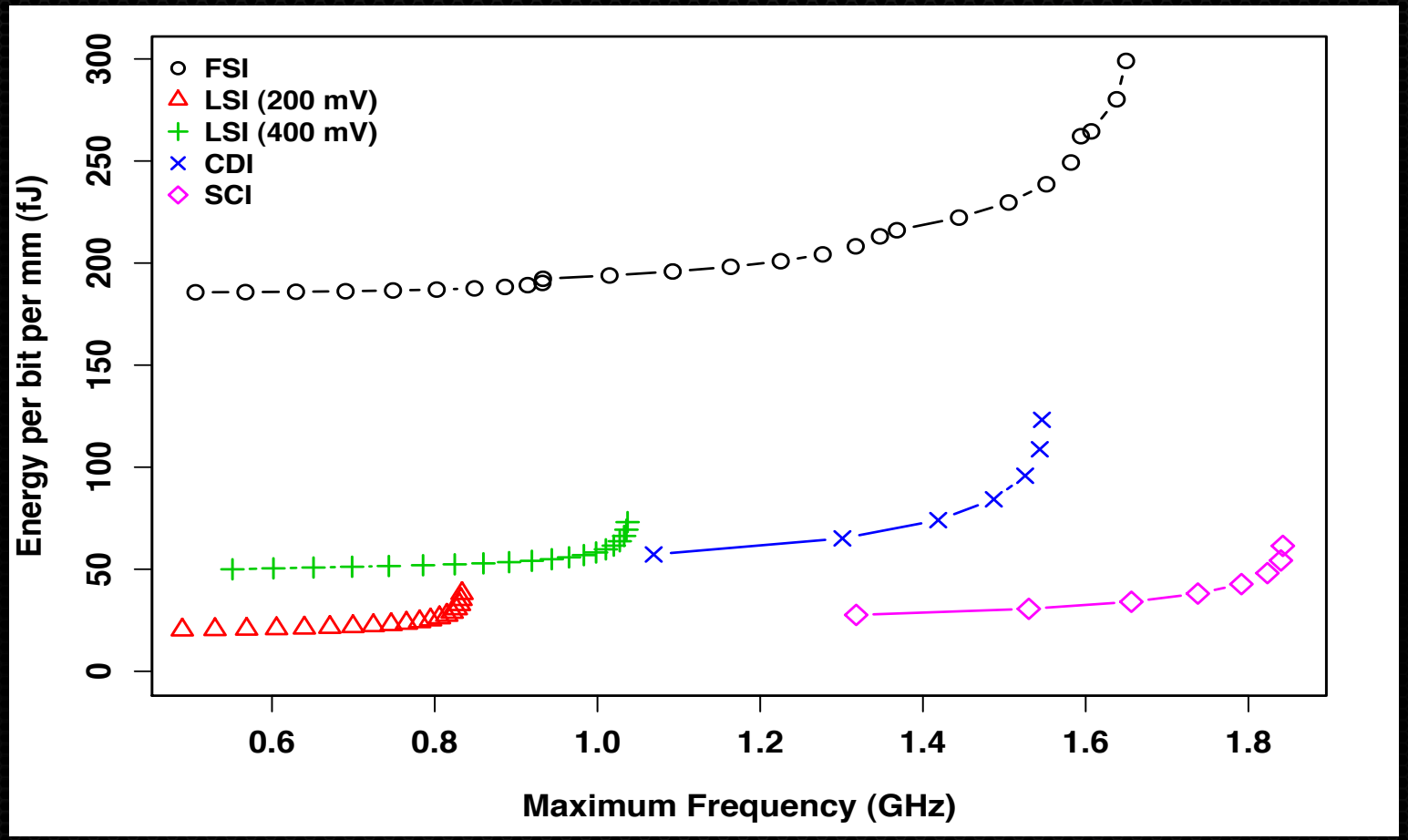
Processor Technology	40 nm	10nm
Vdd (nominal)	0.9 V	0.7 V
DFMA energy	50 pJ	7.6 pJ
64b 8 KB SRAM Rd	14 pJ	2.1 pJ
Wire energy (256 bits, 10mm)	310 pJ	174 pJ

Memory Technology	45 nm	16nm
DRAM interface pin bandwidth	4 Gbps	50 Gbps
DRAM interface energy	20-30 pJ/bit	2 pJ/bit
DRAM access energy	8-15 pJ/bit	2.5 pJ/bit

FP Op lower bound
=
4 pJ

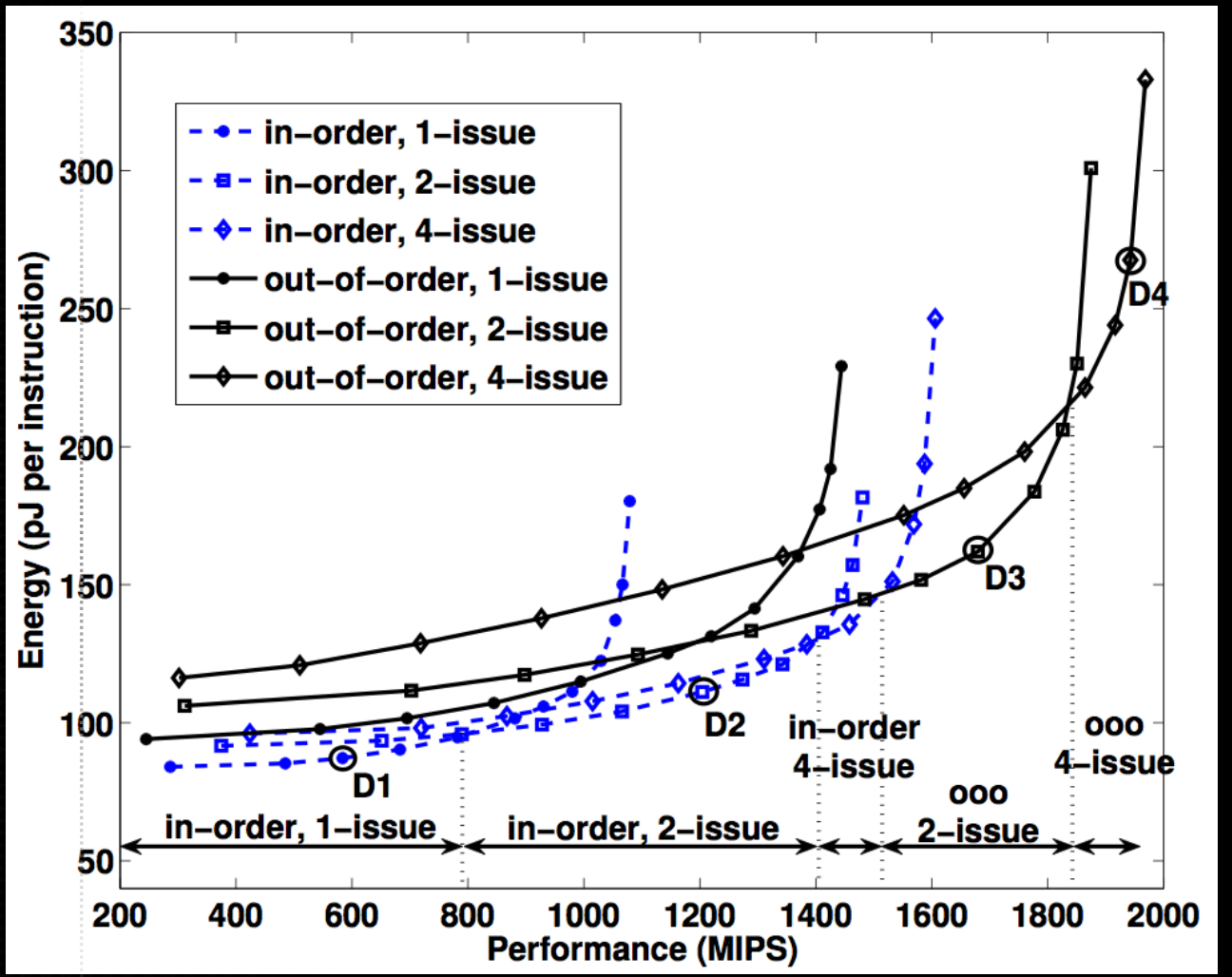
Keckler [Micro 2011], Vogelsang [Micro 2010]

NVIDIA.


FSI
LSI (200 mV)
LSI (400 mV)
CDI
SCI
Energy per bit per mm (fJ)
0
50
100
150
200
250
300
0.6
0.8
1.0
1.2
1.4
1.6
1.8
Maximum Frequency (GHz)

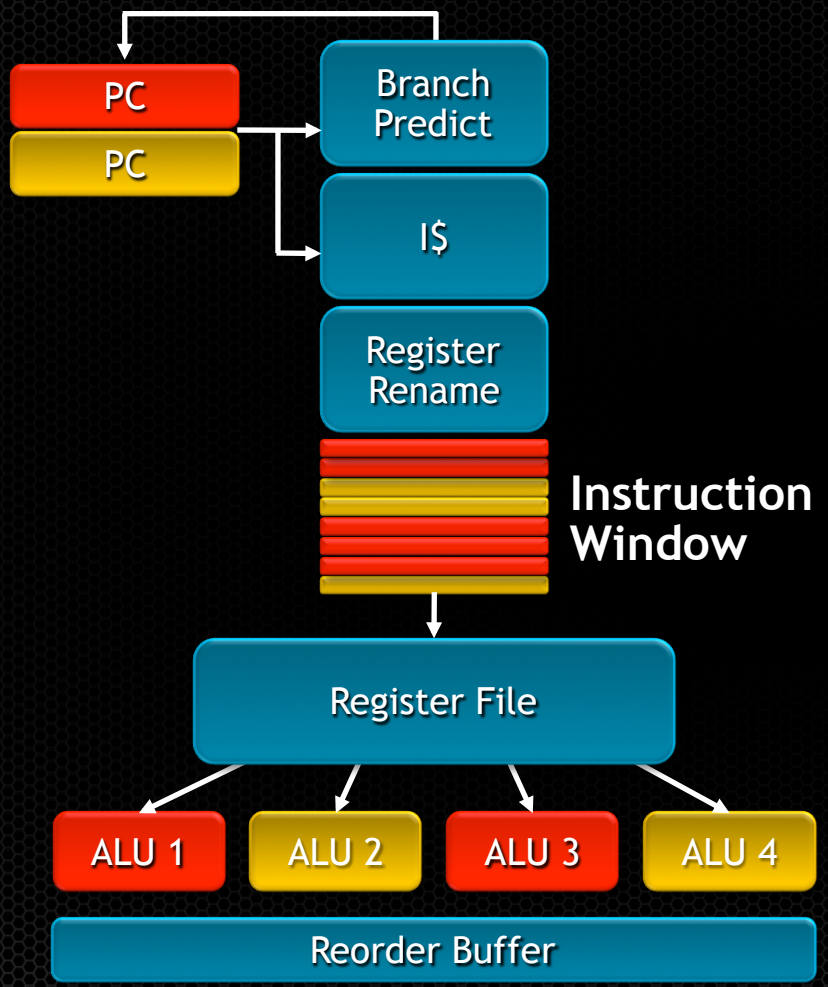
NVIDIA.

Simpler Cores = Energy Efficiency

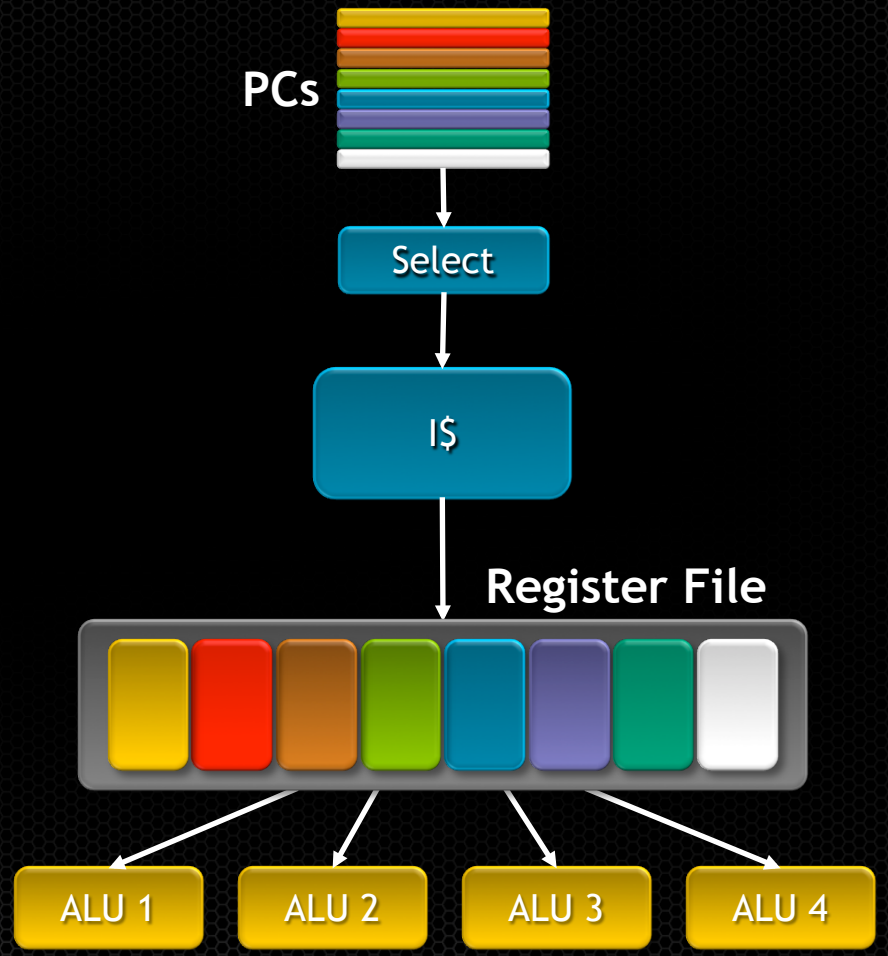


Source: Azizi [PhD 2010]

Latency-Optimized Core (LOC)



Throughput-Optimized Core (TOC)

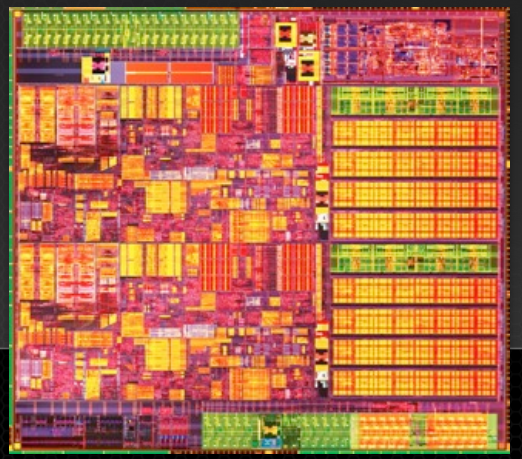


NVIDIA

CPU
1690 pJ/flop

Optimized for Latency

Caches

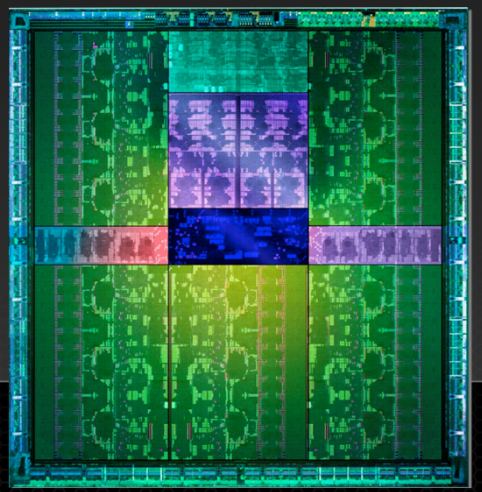


Westmere
32 nm

GPU
140 pJ/flop

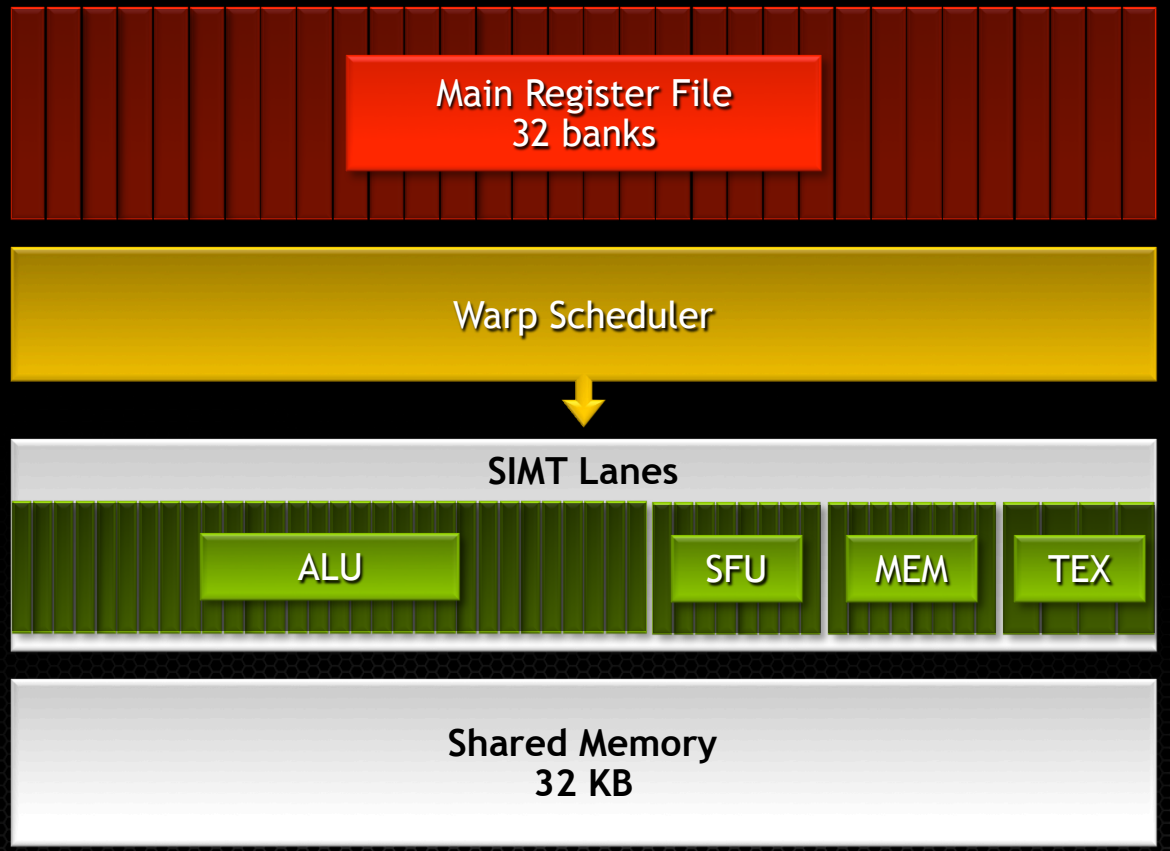
Optimized for Throughput

Explicit Management
of On-chip Memory



Kepler
28 nm

NVIDIA

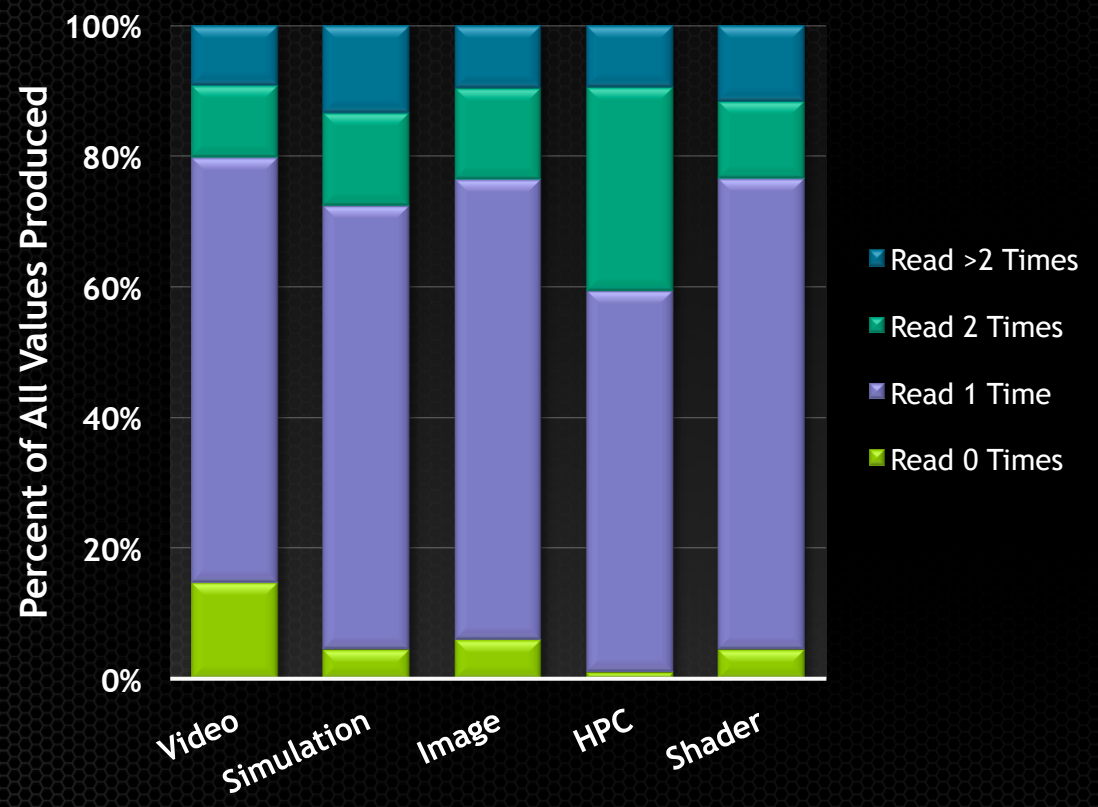


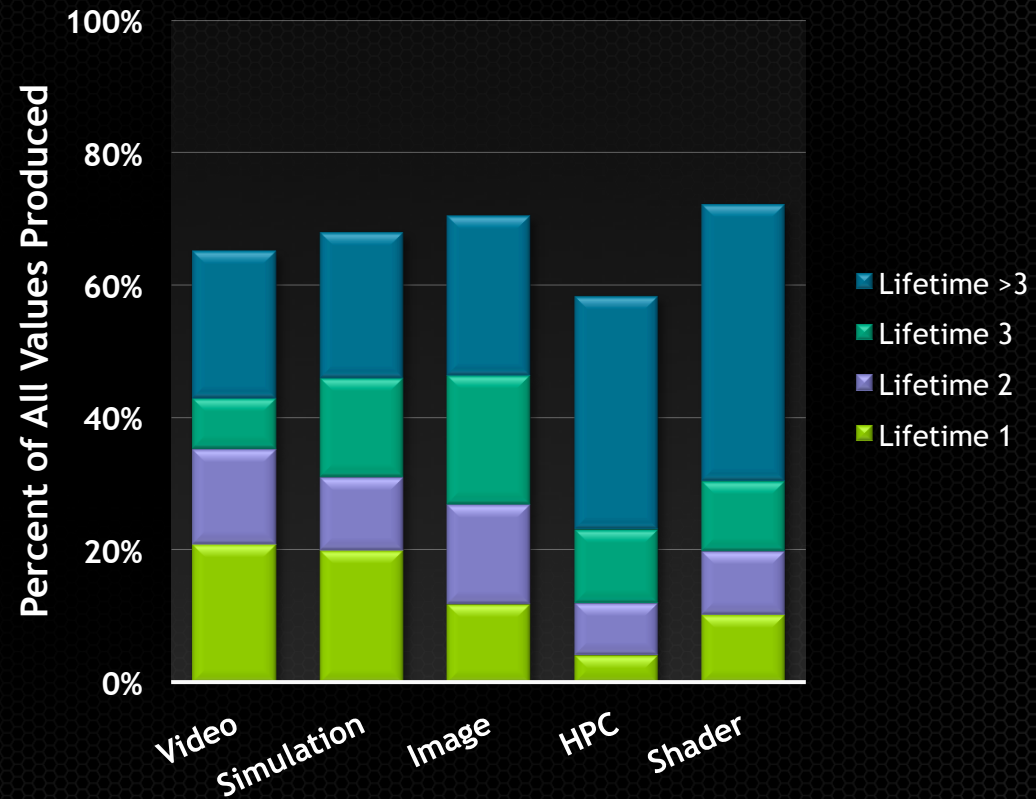
15% of SM Energy

Streaming Multiprocessor (SM)

NVIDIA.

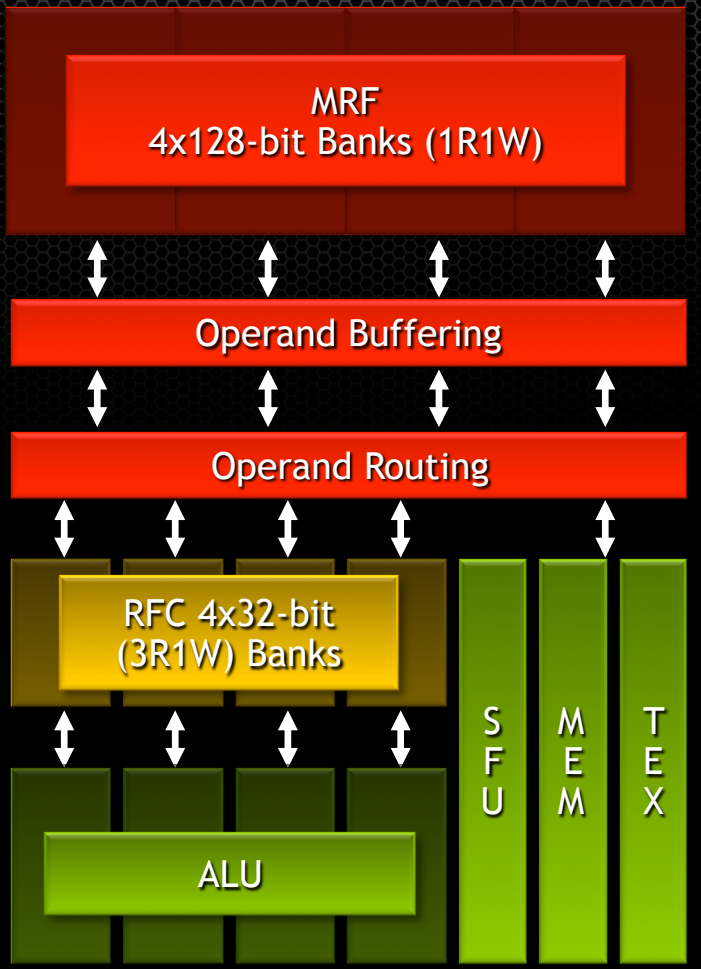
Hierarchical Register File





NVIDIA.

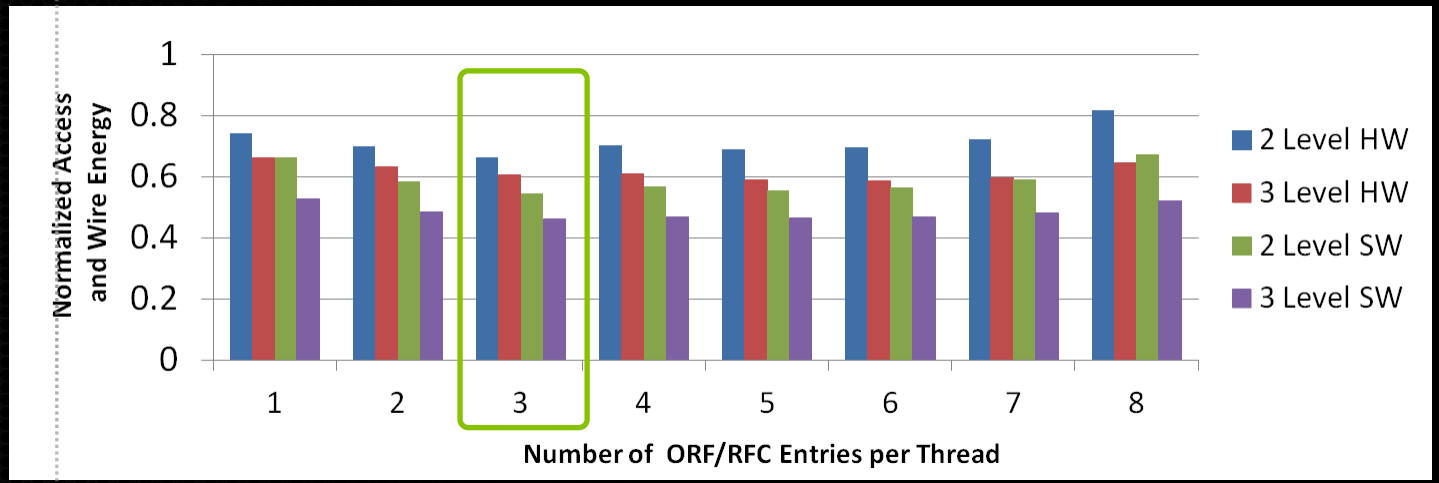
Register File Caching (RFC)



NVIDIA.

Energy Savings from RF Hierarchy

54% Energy Reduction



Source: Gebhart, et. al (Micro 2011)

3 Trends

The End of Dennard Scaling

Pervasive Parallelism

Cloud + Mobile

NVIDIA.

Processors aren’t getting faster just wider

NVIDIA.

Thread Count (CPU+GPU)

	2013 (28nm)	2020 (7nm)
Cell Phone	4+1,000	16+4,000
Server Node	12+10,000	32+40,000
Supercomputer	10^5+10^8	10^6+10^9

NVIDIA.

Parallel programming is not inherently any more difficult than serial programming

However, we can make it a lot more difficult

NVIDIA.

A simple parallel program

```
forall molecule in set { // launch a thread array
    forall neighbor in molecule.neighbors { // nested
        forall force in forces { // doubly nested
           molecule.force =
             reduce_sum(force(molecule, neighbor))
        }
    }
}
```

NVIDIA.

Why is this easy?

```
forall molecule in set { // launch a thread array
    forall neighbor in molecule.neighbors { // nested
        forall force in forces { // doubly nested
           molecule.force =
             reduce_sum(force(molecule, neighbor))
        }
    }
}
```

No machine details
All parallelism is expressed
Synchronization is semantic (in reduction)

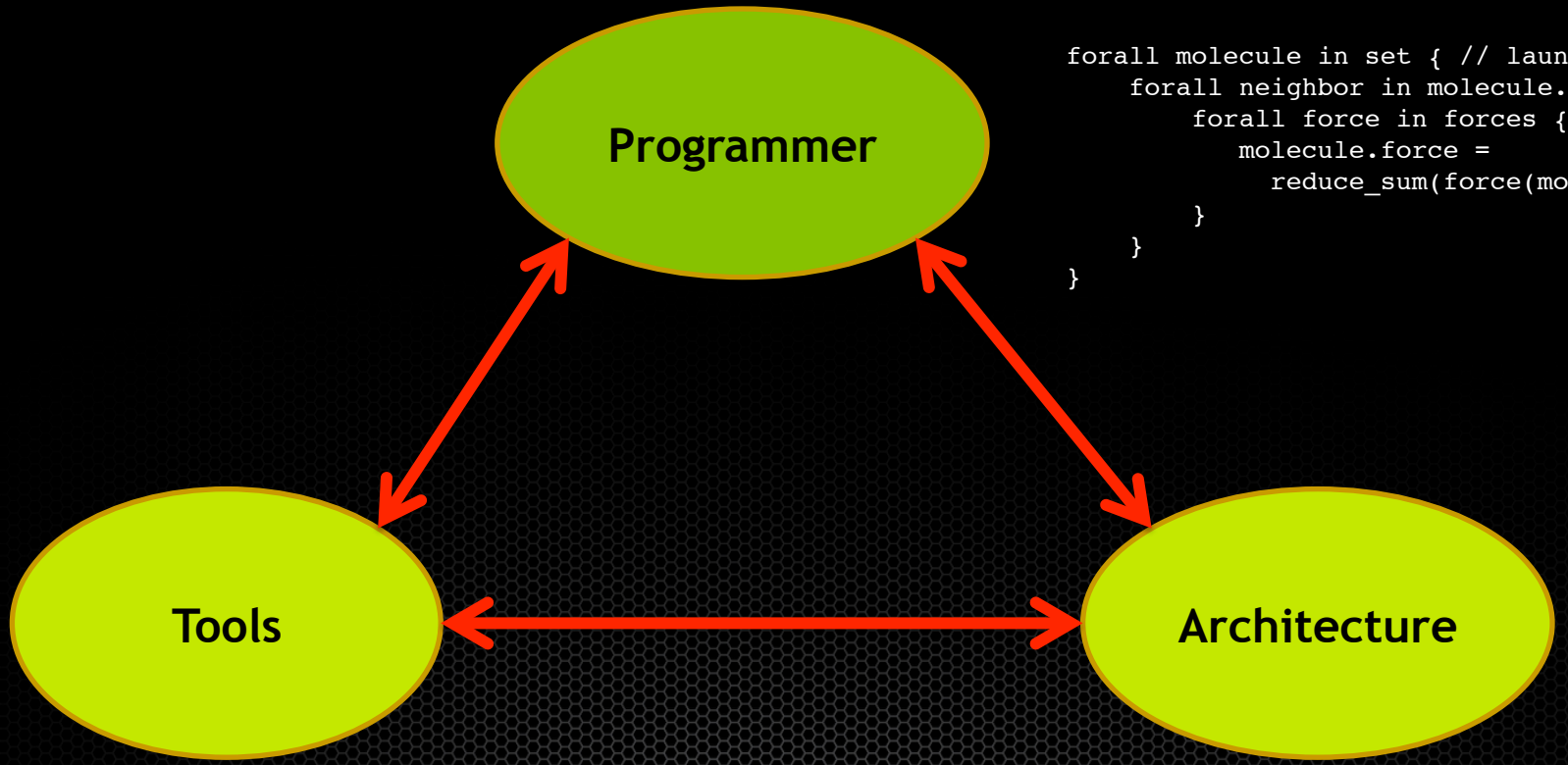
NVIDIA.

```
pid = fork() ; // explicitly managing threads

lock(struct.lock) ;  // complicated, error-prone synchronization
// manipulate struct
unlock(struct.lock) ;

code = send(pid, tag, &msg) ;  // partition across nodes
```

We could make it hard

NVIDIA.

Programmers, tools, and architecture Need to play their positions

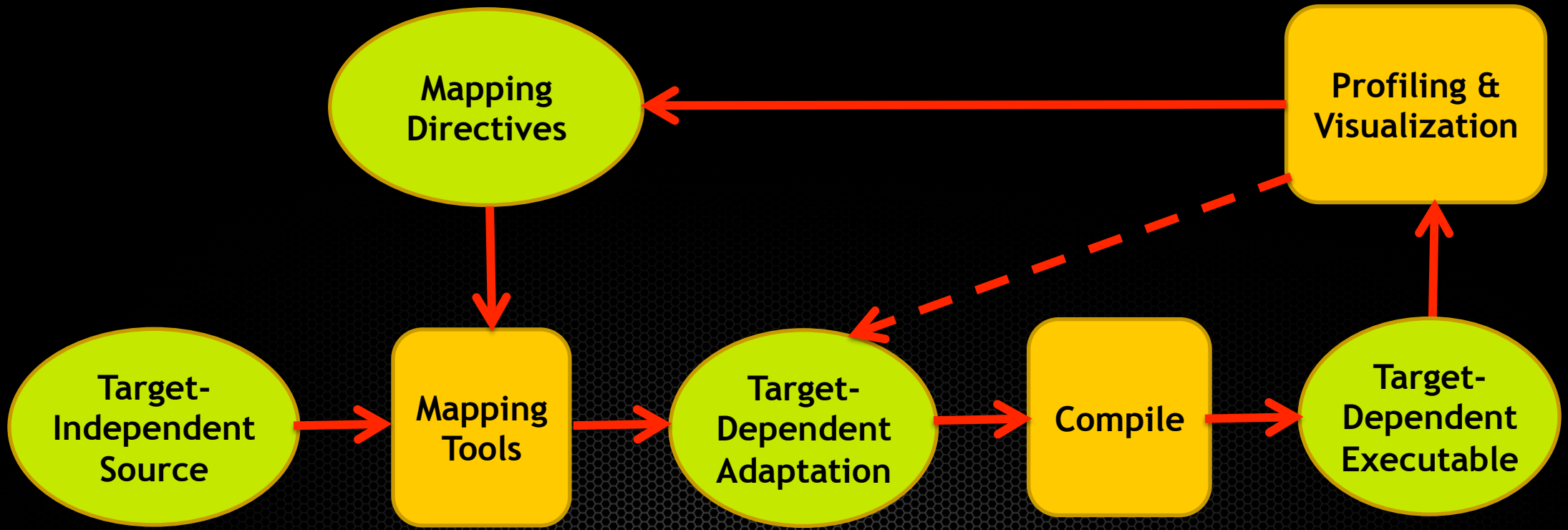



```
forall molecule in set { // launch a thread array
    forall neighbor in molecule.neighbors { //
        forall force in forces { // doubly nested
            molecule.force =
                reduce_sum(force(molecule, neighbor))
        }
    }
}
```

Map foralls in time and space
Map molecules across memories
Stage data up/down hierarchy
Select mechanisms

Exposed storage hierarchy
Fast comm/sync/thread mechanisms

NVIDIA.


Mapping Directives
Profiling & Visualization
Target-Independent Source
Mapping Tools
Target-Dependent Adaptation
Compile
Target-Dependent Executable

NVIDIA.

3 Trends

The End of Dennard Scaling

Pervasive Parallelism

Cloud + Mobile

NVIDIA.

Future Systems are Energy Limited

NVIDIA.

Future Systems are Energy Limited

But Remote Computation Allows a 1kW experience in a 1W handheld device

NVIDIA.

Future Systems are Energy Limited

But Remote Computation Allows a 1kW experience in a 1W handheld device

And Simplifies System Management

NVIDIA.

Future Systems are Energy Limited

But Remote Computation Allows a 1kW experience in a 1W handheld device

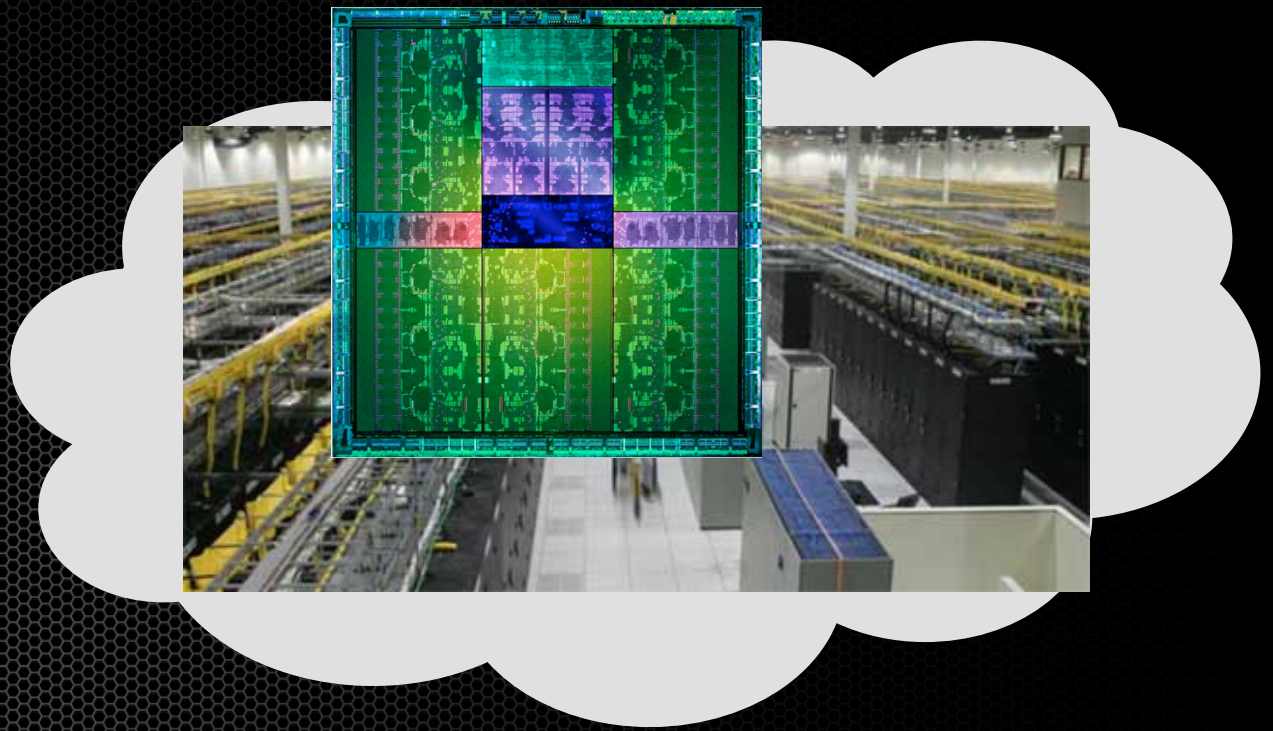
Simplifies System Management

And Enables Access to Large Volumes of Data

NVIDIA.

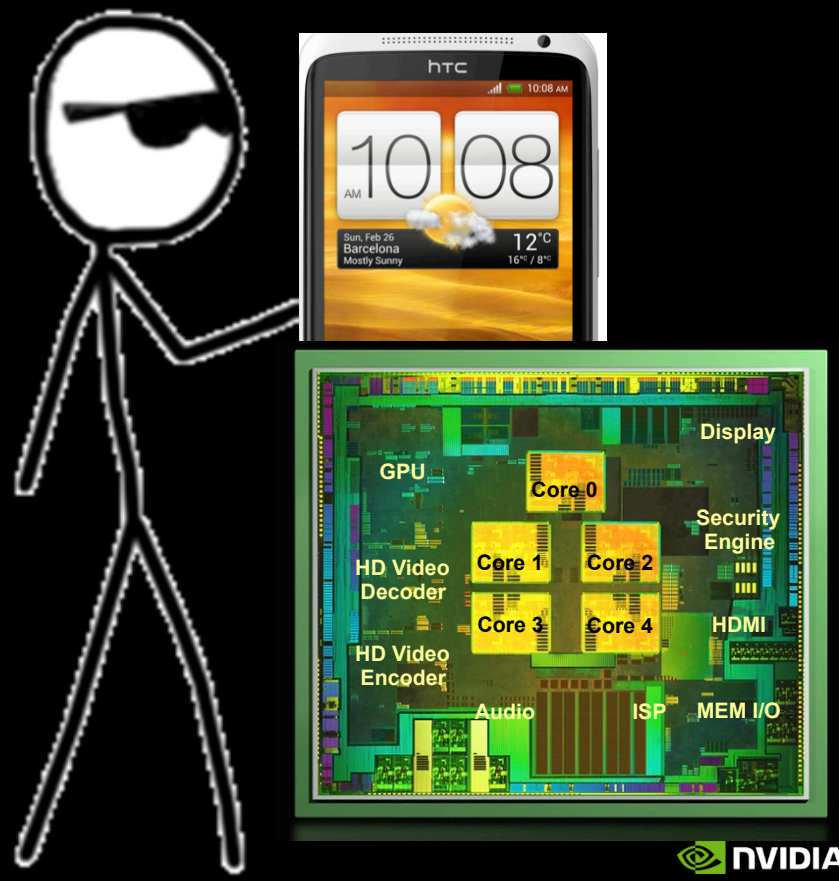
htc
10:08 AM
AM 10 08
Sun, Feb 26
Barcelona
Mostly Sunny
12°C
16°C / 8°C
Play Store
Favorites

NVIDIA.



Large, Highly Parallel Server
Optimized for perf/W

Small, Mobile Device
Optimized for User Experience



NVIDIA

Some Implications

NVIDIA.

Conclusion

- Large computers are scientific instruments
  - To study nature – or business
- Three key trends
  - The end of Dennard scaling – everything is energy limited
  - Pervasive parallelism – computers get wider, not faster
  - Cloud + Mobile
- Energy efficiency through architecture and circuits
- Parallel programming is simple – we can make it hard
  - Target-independent programming – mapping via tools

NVIDIA

“Super” Computing
From Super Computers to Super Phones