"Toto, I've a feeling we're not on Moore's Law anymore."

Dileep Bhandarkar, Ph. D.

IEEE Life Fellow

Random Access @ The Salishan Conference

27 April 2016

Disclaimer

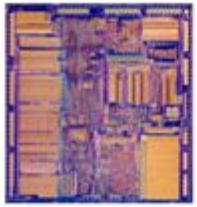
This presentation is based on personal
Experiences over the last 40+ years in industry
As a Computer Architect
and
Is not presented on behalf of
current or past employers.

Directions | Contact Us

COMPUTER HISTORY MUSEUM

The Silicon Engine:
A Timeline of Semiconductors in Computers

MOORE'S LAW "Transistor density on integrated circuits doubles about every two years." *

1950s	1960s	1970s	1980s	1990s	2000s
Silicon Transistor	TTL Quad Gate	8-bit Microprocessor	32-bit Microprocessor	32-bit Microprocessor	64-bit Microprocessor
					
1 Transistor	**16 Transistors**	**4500 Transistors**	**275,000 Transistors**	**3,100,000 Transistors**	**592,000,000 Transistors**

Microelectronic silicon computer "chips" have grown in capability from a single transistor in the 1950s to hundreds of millions of transistors per chip on today's microprocessor and memory devices. From the first documented semiconductor effect in 1833 to the transition from transistors to integrated circuits in the 1960s and 70s, this website explores key milestones in the development of these extraordinary engines that power the computing and communications revolution of the information age.

*Source: "Moore's Law: Raising the Bar" (Intel Corporation 2005)

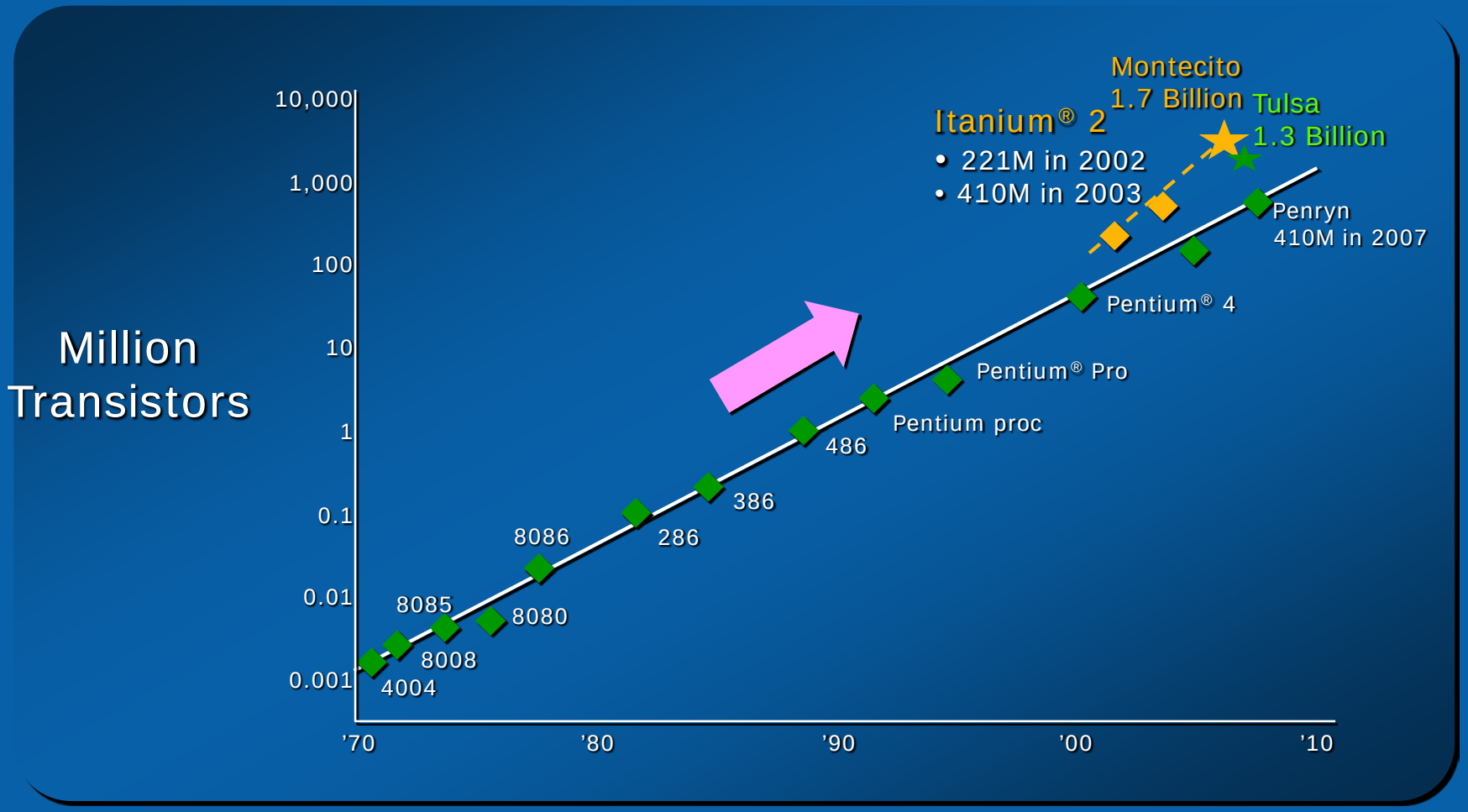
Photo credits: Fairchild Camera and Instrument Corporation, Intel Corporation (Note that images are not to scale)

1958: Jack Kilby's Integrated Circuit

SSI -> MSI -> LSI -> VLSI -> OMGWLSI

From 2300 to >1Billion Transistors

In < 40 Years of Moore's Law



More than 1 Billion Transistors in 2006!

Dennard Scaling

Device or Circuit Parameter	Scaling Factor
Device dimension tox, L, W	1/K
Doping concentration Na	K
Voltage V	1/K
Current I	1/K
Capacitance eA/t	1/K
Delay time per circuit VC/I	1/K
Power dissipation per circuit VI	$1/K^2$
Power density VI/A	1

Dennard's 1974 paper summarizes transistor or circuit parameter changes under ideal MOSFET device scaling conditions, where K is the unitless scaling constant.

The benefits of scaling : as transistors get smaller, they can switch faster and use less power. Each new generation of process technology was expected to reduce minimum feature size by approximately 0.7x (K ~1.4). A 0.7x reduction in linear features size provided roughly a 2x increase in transistor density.

Dennard scaling broke down around 2004 with unscaled interconnect delays and our inability to scale the voltage and the current due to reliability concerns.

But our the ability to etch smaller transistors has continued spawning multicore designs.

Post Dennard Scaling

- Moore's Law continued for 10 more years!
- Instruction Level Parallelism harder to find
- Increasing single-stream scalar performance often requires non-linear increase in design complexity, area, and power
- Vectorization for increasing floating point performance

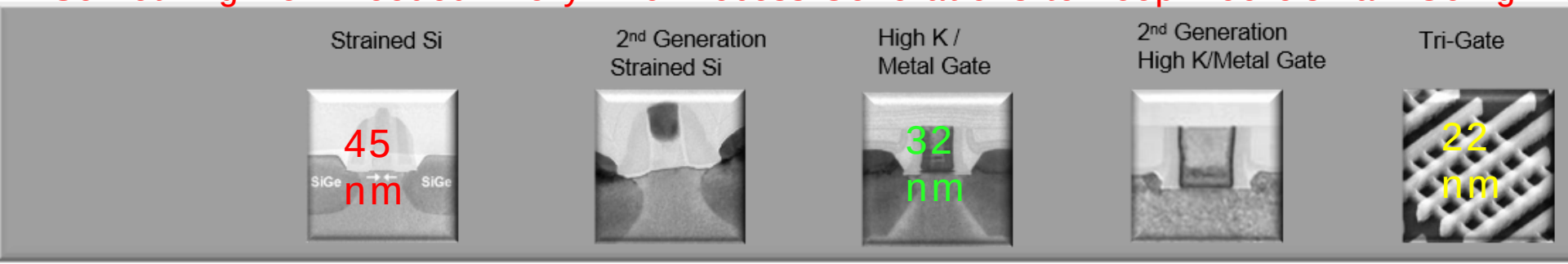
THE MULTICORE ERA

NEW DEVICE STRUCTURES & MATERIALS

ENERGY EFFICIENCY WITH POWER CONSTRAINTS

2001 2003 2005 2007 2009 2011

Something New Needed Every Two Process Generations to Keep Moore's Law Going



Multi-Core Era

Who Has The Most Cores?

4 is Better Than 2!

And

8 is Even Better!

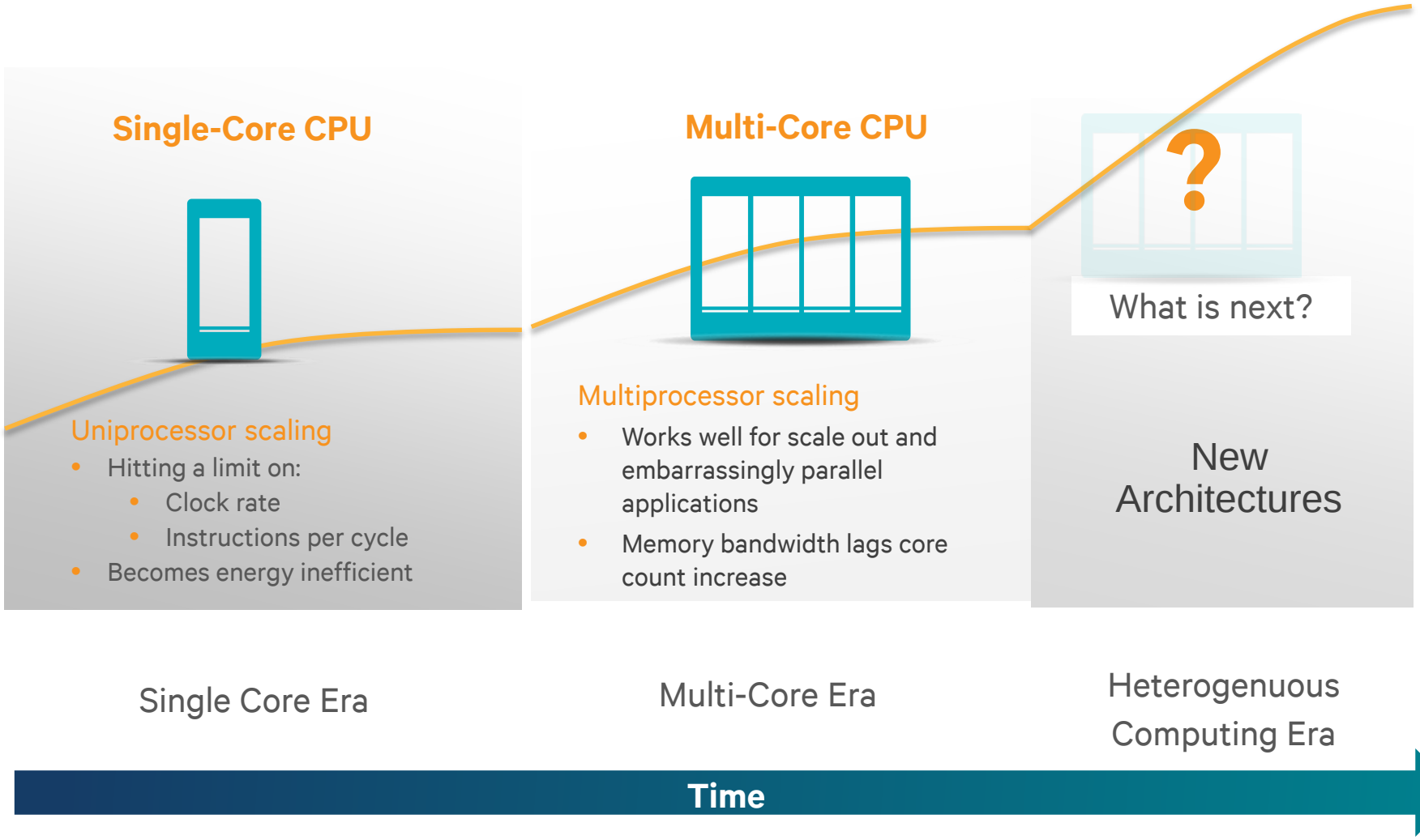
22 nm Intel Ivy Bridge Xeon E5/E7 had 15 cores in 525 mm2

22 nm Intel Haswell Xeon E5/E7 had 18 cores in 662 mm2

14 nm Intel Broadwell Xeon E5/E7 has 24 cores in 456 mm2

FLOPS per core also doubled with each generation

CPU scaling is reaching diminishing returns



Thoughts about the Future?

- 14 nm is in production but ramping slower than previous generations
  - Future Generations will be even harder!
- Costs per wafer increasing
  - Capital, more process steps, increased mask costs, EUV cost
  - Cost per transistor decreasing, but at a slower rate
- Moore's Law is slowing down beyond 14 nm
  - New process generation every 30 months
  - Economics, Physics, Materials, Power, Lithography
  - What is the best use for increased transistor density?
  - Other architectures?
  - Heterogenuous Processing Engines?
- Is vectorized floating point sufficient?
- Can we truly exploit higher levels of parallelism in large "traditional" systems effectively & efficiently?

Thank You

dbhandarkar@outlook.com