Principles of Programming Languages

New Processor Architectures and Programming Paradigms

Need for Concurrency

Multi-Core Architectures

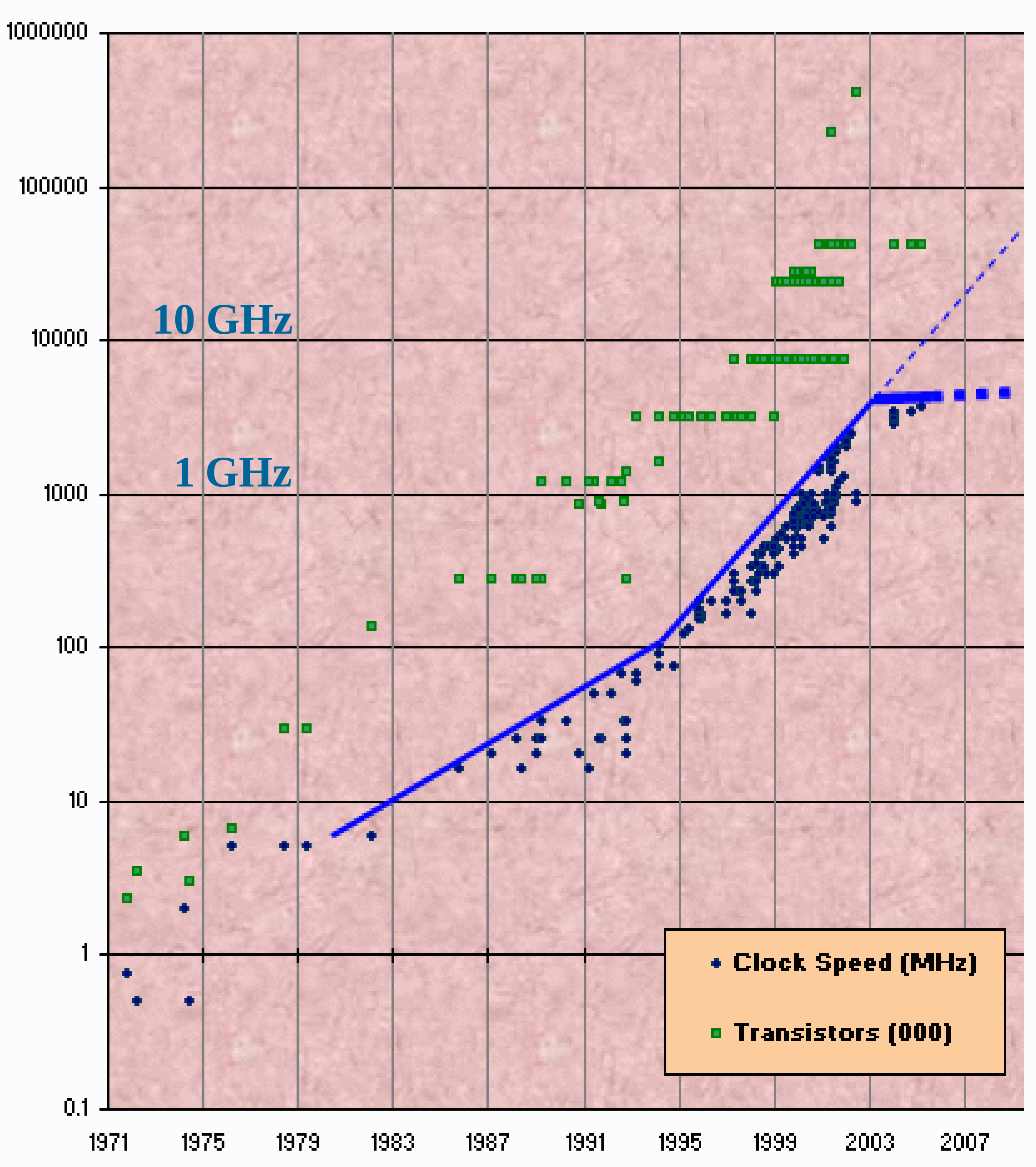
Need for New Languages

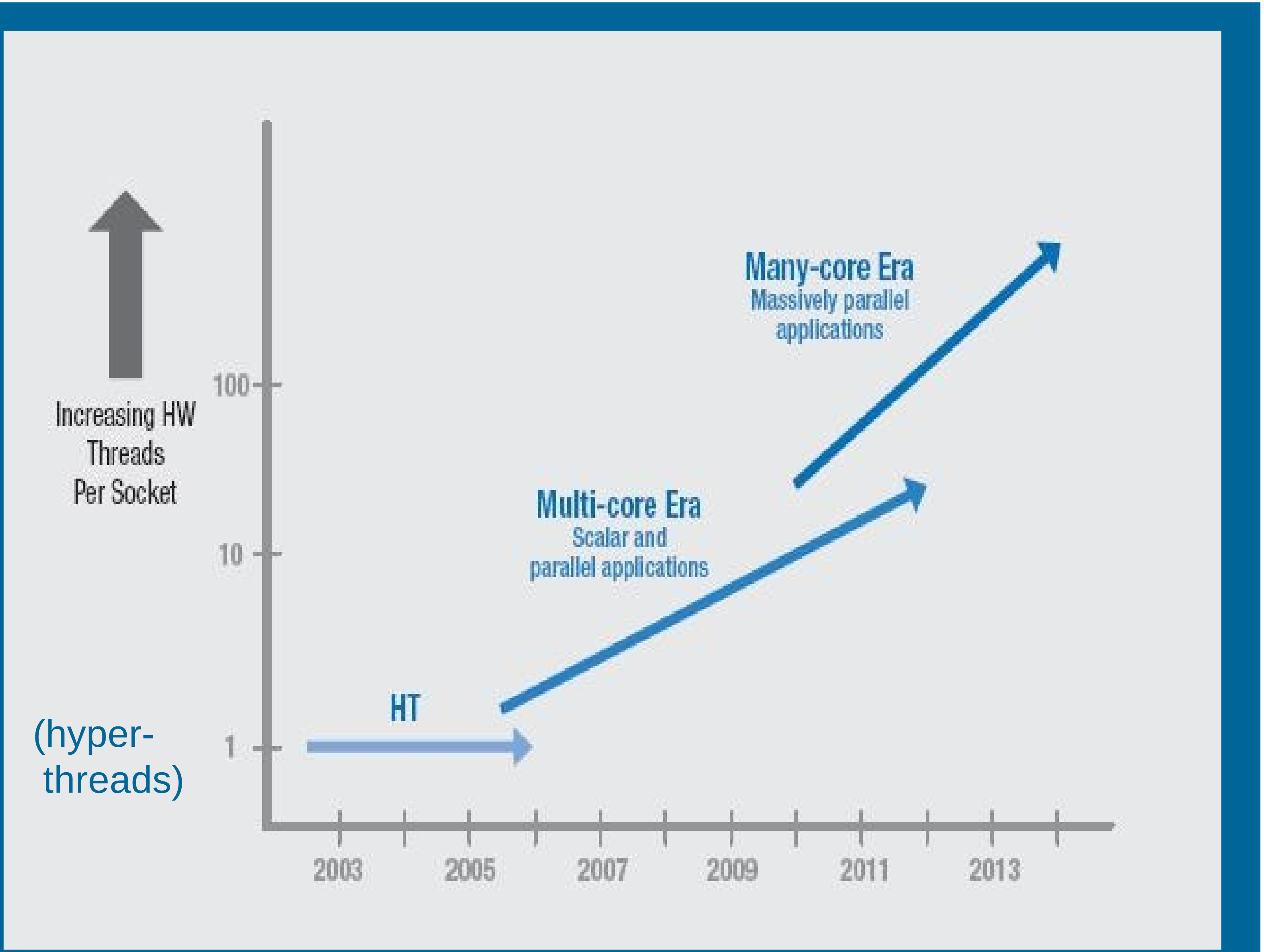
"Leveraging the full power of multicore processors demands new tools and new thinking from the software industry."

"Our parallel future has finally arrived: new machines will be parallel machines, and this will require major changes in the way we develop software."

"Programming languages and systems will increasingly be forced to deal well with concurrency."

Sutter: *The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software*, 2005





(hyper-
threads)

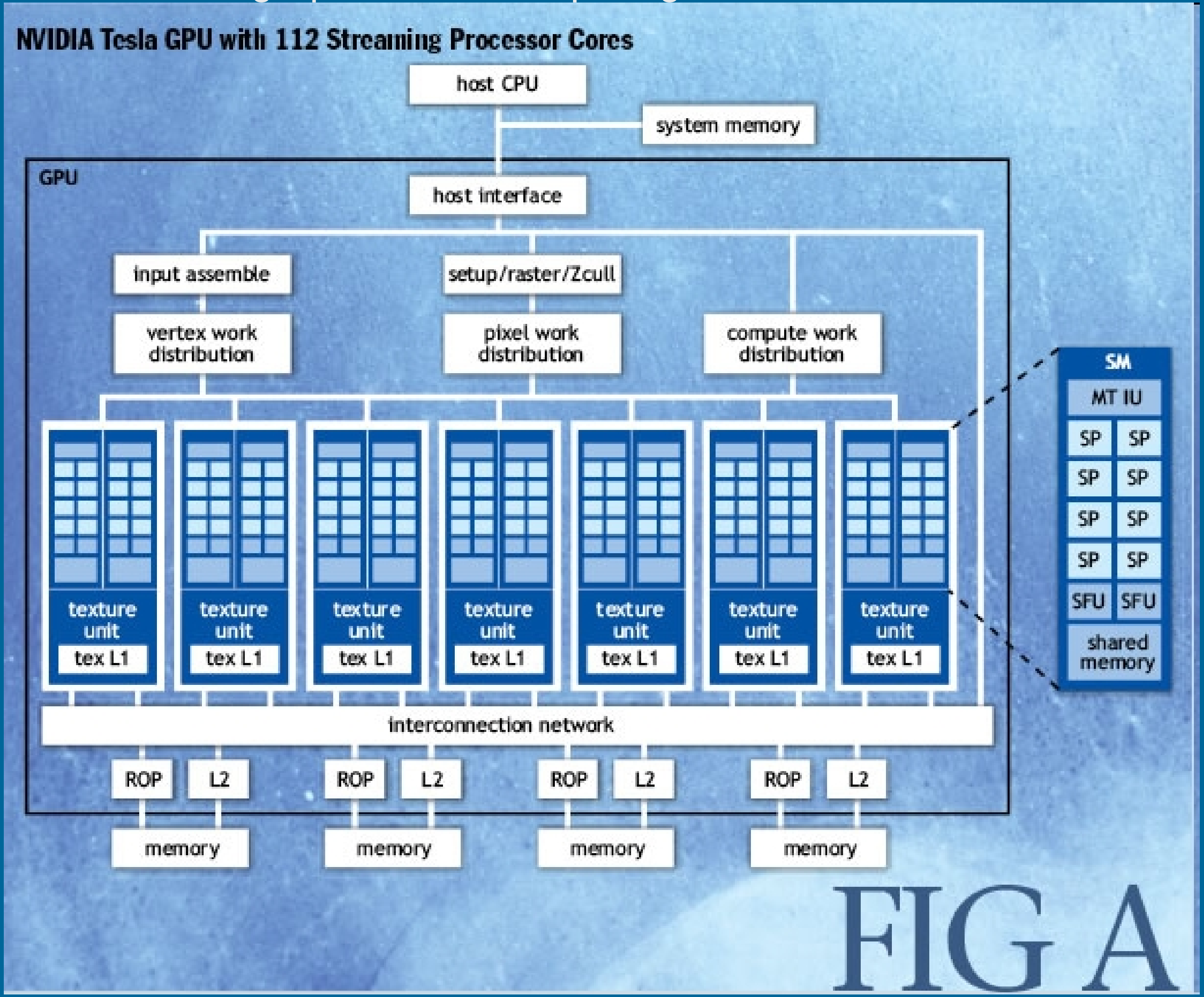
Borkar, Dubey, Kahn, et al. “Platform 2015.” Intel White Paper, 2005.

http://www.cs.helsinki.fi/u/kerola/rio/papers/borkar_2015.pdf

Moore’s Law Reinterpreted

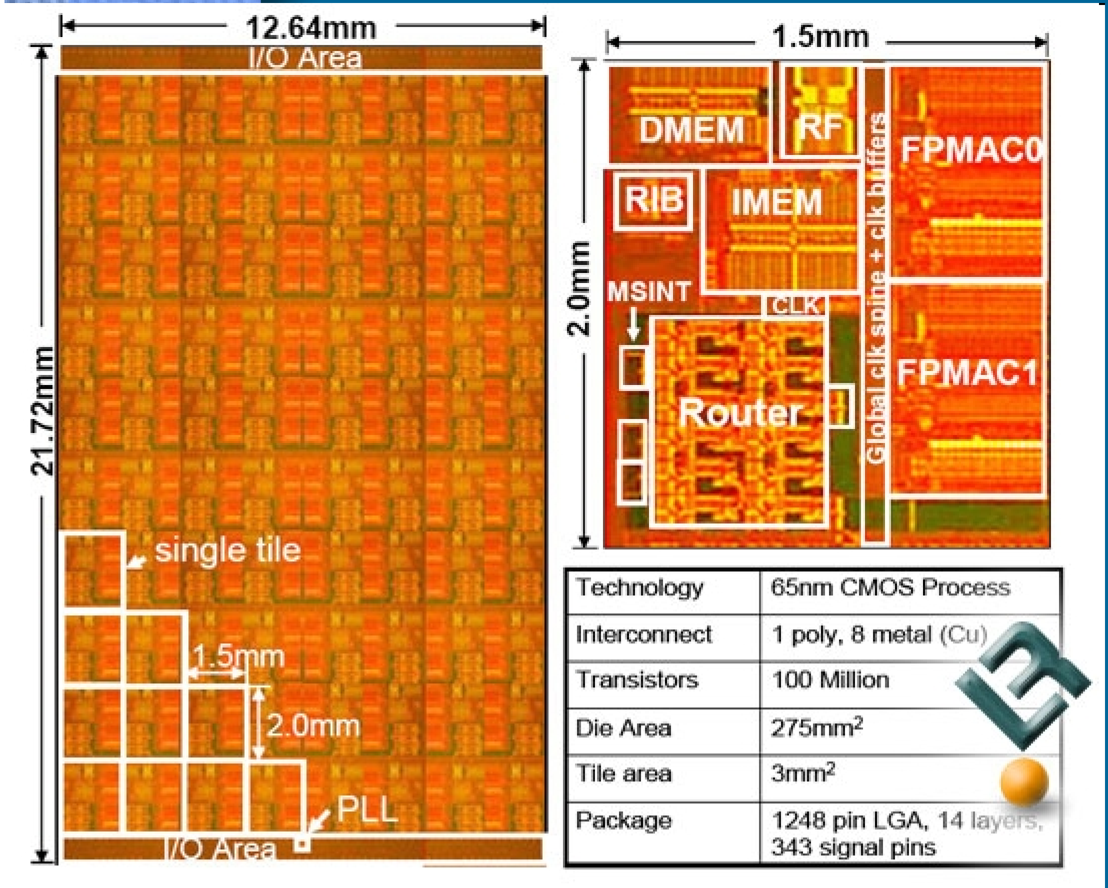
- Number of cores per chip doubles every two years, while clock speed decreases
  - Need to utilize systems with hundreds or thousands of cores
  - Need to handle systems with millions (billions?) of concurrent threads
  - Need to emphasize scalability – not best performance for fixed number of cores.
  - Need to be able to easily replace inter-chip parallelism with intra-chip parallelism

Tesla unified graphics and computing architecture



http://www.acmqueue.org/modules.php?name=Content&pa=showpage&pid=532&page=6

http://www.legitreviews.com/article/460/1/



Technology	65nm CMOS Process
Interconnect	1 poly, 8 metal (Cu)
Transistors	100 Million
Die Area	275mm^2
Tile area	3mm^2
Package	1248 pin LGA, 14 layers, 343 signal pins

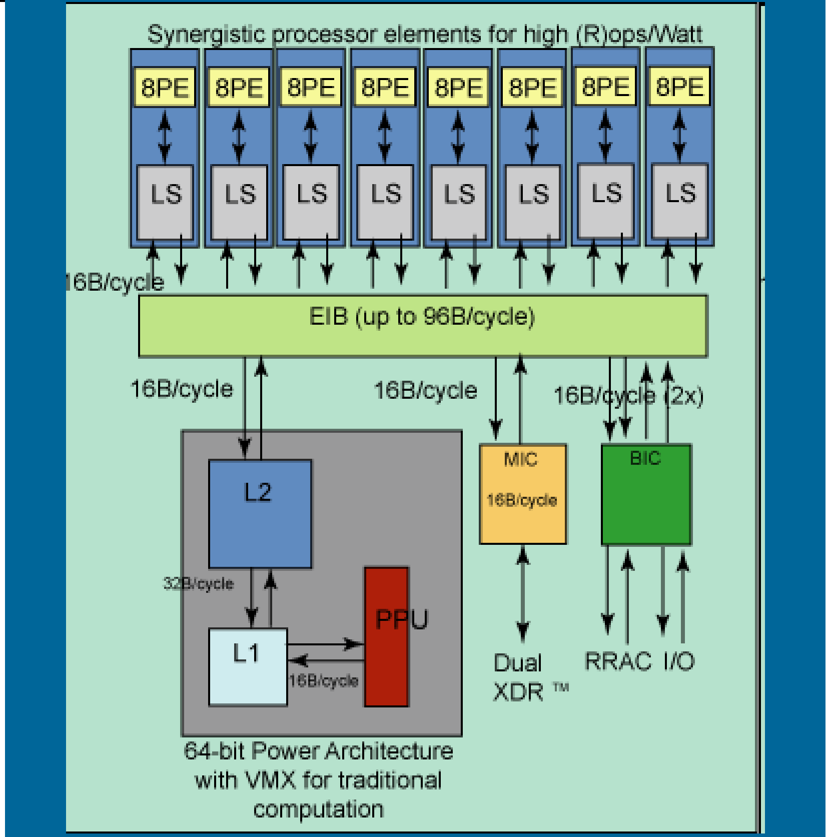
Intel Teraflops Research Chip wafer

Cool 80-core chip:

- block matrix operations

  1 TFLOPS at 1.0V at 110 °C

http://www.ibm.com/developerworks/power/library/pa-celltips1/



The Cell processor

Fast Roadrunner system

- 12 960 Cells, 1 PFLOPS, 2.3 MW
- 6 480 dual-core Opteron I/O
- total 116 640 cores
- 90 km fiber-optic cable, 560m^2

Multi-core SW Development

- Multi-core architectures: an inflection point in mainstream SW development
- Writing parallel SW is hard
  - Mainstream developers (currently) not used to thinking in parallel
  - Mainstream languages (currently) force the use of (existing) low-level concurrency features
  - Must have parallel SW with new systems
- Need better concurrency abstractions

New Software Paradigm

- The overarching goal should be to make it easy to write programs that execute efficiently on highly parallel computing systems
- The target should be 1000s of cores per chip
  - Shared memory or distributed memory
- Programming models should be independent of the number of processors
- Programming models should support a wide range of data types and successful models of parallelism: task-level, word-level, and bit-level parallelism
- We need a programming model, system software, and a supporting architecture that are naturally parallel