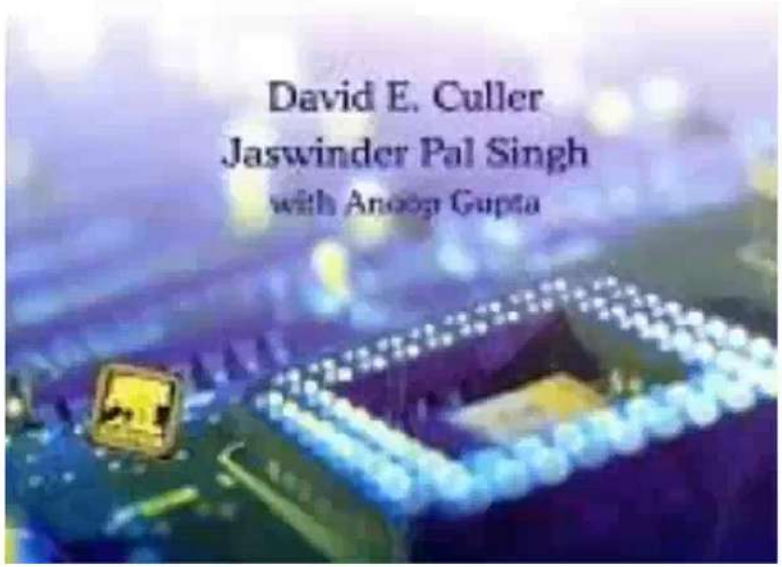
Why Parallel Computing?

Lecture-1

PARALLEL COMPUTER ARCHITECTURE

A Hardware/Software Approach



CS527: PCA

- Slot-B1: Thur, Fri [2:00 – 3:00 PM] ,
  Mon [3:00 – 4:00 PM]
- Room: MS-Teams
- Book: Parallel Computer Architecture: A Hardware/Software Approach
  - David Culler, J. P. Singh and A. Gupta
- Evaluation
  - Assignment
  - Quiz
  - Mid-sem
  - End-sem
  - seminar/term-paper/research paper based test

- TAs: PhD students, department of CSE
  - Arijit Nath
  - Aswathy
  - Imlijungla
  - Neeraj
  - Swati
  - Deep
  - Chetan
  - Nishant

Why we need ever increasing parallelism?

- Dramatic advancement in the field of science, internet, entertainment have increased computational power at the foundation/heart
- e.g. decoding the human genome, accurate medical imaging, fast and accurate web search, more realistic computer games
- But we cannot rest on these laurels. More open problems to solve ...

Open problems

- Climate modeling: includes interactions of ocean, atmosphere, land, ice-caps, etc.
- Protein folding: misfolded proteins leads to diseases: Huntington, Parksinson, Alzheimer, etc. Our ability to study configurations of complex molecules is severely limited by compute power
- Drug discovery: analysis of genome and drug effectiveness for personalised treatment of diseases depending on individuals
- Energy research: we will be able to program more detailed models of wind turbines, solar cells, batteries. Construct more efficient clean energy sources
- Data Analysis: we generate tremendous amounts of data, doubling every two years. Unless analysed, this data is useless
  - e.g. sequence of nuleotides in DNA is of little use unless we analyse it for its effect on diseases
  - Vast quantities of data are generated by particle colliders – Large Hadron Collider at CERN, medical imaging, astronomical research, web search engines

Open problems

- Climate modeling: includes interactions of ocean, atmosphere, land, ice-caps, etc.
- Protein folding: misfolded proteins leads to diseases: Huntington, Parksinson, Alzheimer, etc. Our ability to study configurations of complex molecules is severely limited by compute power
- Drug discovery: analysis of genome and drug effectiveness for personalised treatment of diseases depending on individuals
- Energy research: we will be able to program more detailed models of wind turbines, solar cells, batteries. Construct more efficient clean energy sources
- Data Analysis: we generate tremendous amounts of data, doubling every two years. Unless analysed, this data is useless
  - e.g. sequence of nuleotides in DNA is of little use unless we analyse it for its effect on diseases
  - Vast quantities of data are generated by particle colliders – Large Hadron Collider at CERN, medical imaging, astronomical research, web search engines

These and host of other problems won't be solved without vast increase in computational power

More appln: Science and Engg

- Examples:
  - Weather prediction
  - Evolution of galaxies
  - Oil reservoir simulation
  - Automobile crash tests
  - Drug development
  - VLSI CAD
  - Nuclear bomb simulation

- Typically model physical systems or phenomena
- Problems are 2D or 3D
- Usually require “number crunching”
- Involve “true” parallelism

More appln: Commercial

- Examples
  - On-line transaction processing (OLTP)
  - Decision support systems (DSS)
  - "Application servers" or "middleware" (WebSphere)
- Involves data movement, not much number crunching
  - OTLP has many small queries
  - DSS has fewer but larger queries
- Involves throughput parallelism
  - Inter-query parallelism for OLTP
  - Intra-query parallelism for DSS

More appln: multi-media

- Examples:
  - Speech recognition
  - Audio/video
  - Data compression/decompression
  - 3D graphics
  - Gaming!

- Involves everything (crunching, data movement, true parallelism, and throughput parallelism)

Hardware design
Constraint are driving towards
Parallelism

Topics

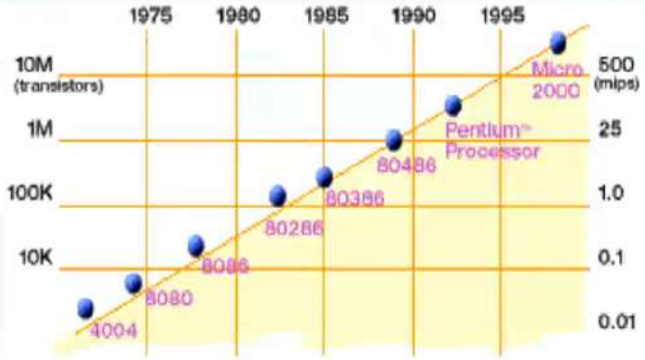
- Why are we building multicore systems?
  - Application requirement
  - Hardware design constraints
- Different parallel systems
- shared memory used for communication
- data consistency / coherence
- Coherence protocols
  - 3-4 types
  - protocol implementation details
  - protocol correctness issues
- Consistency
- Synchronisation

Multiprocessor Revolution

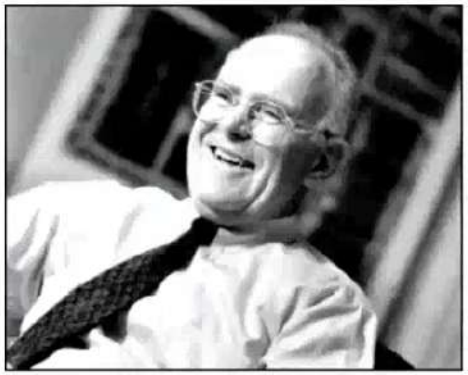
Hemangee K. Kapoor

Department of CSE, IIT Guwahati

Moore's law is Alive and Well

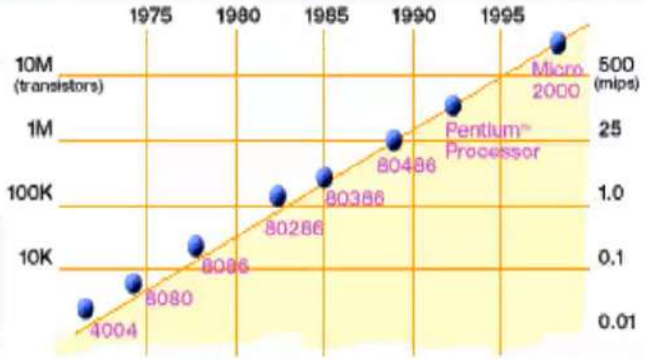


2X transistors/Chip Every 1.5 years
Called "Moore's Law"

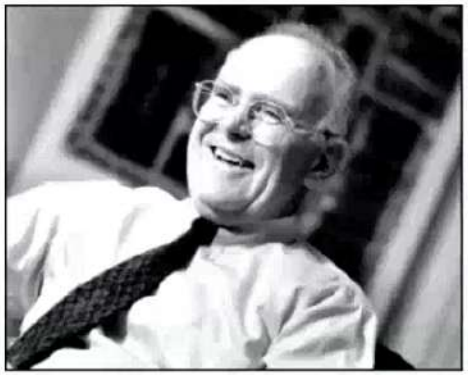


Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months.

Moore's law is Alive and Well



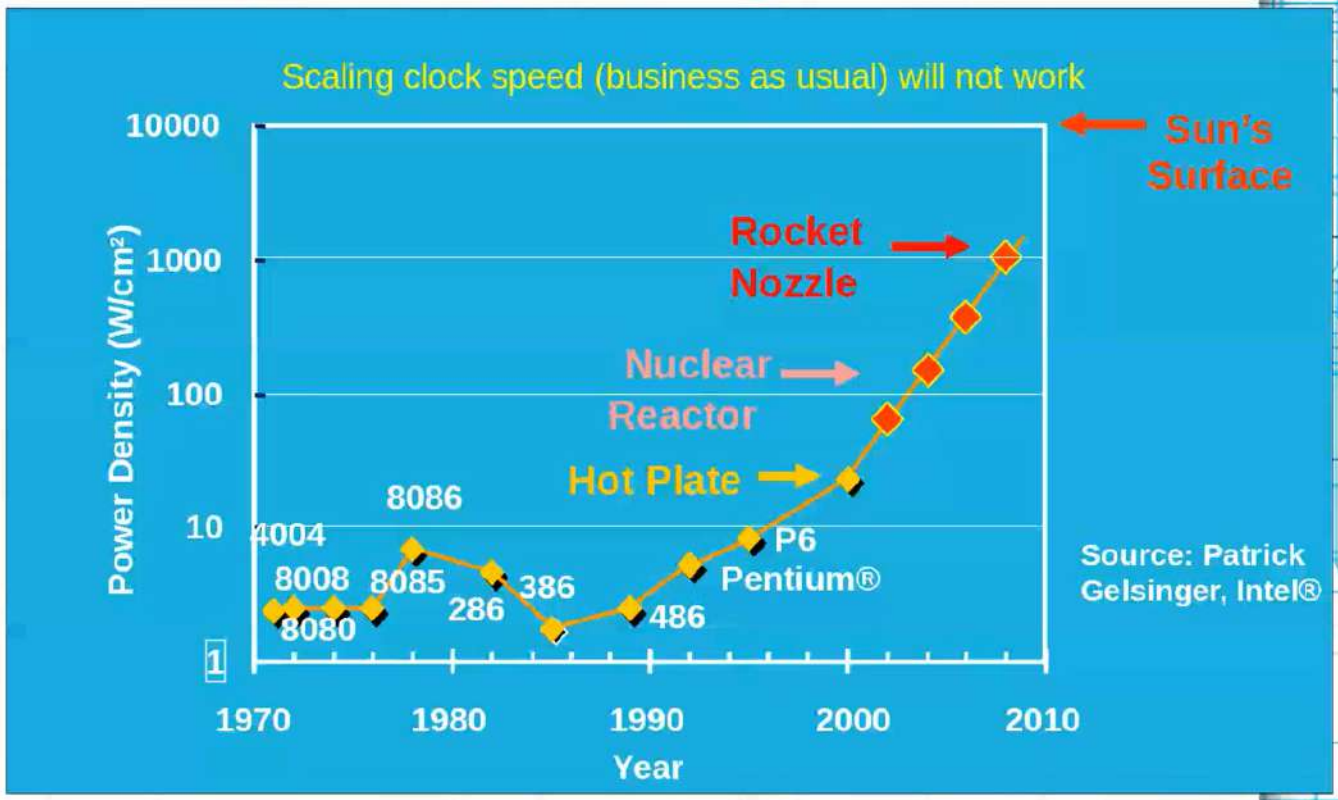
2X transistors/Chip Every 1.5 years
Called "Moore's Law"



Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months.

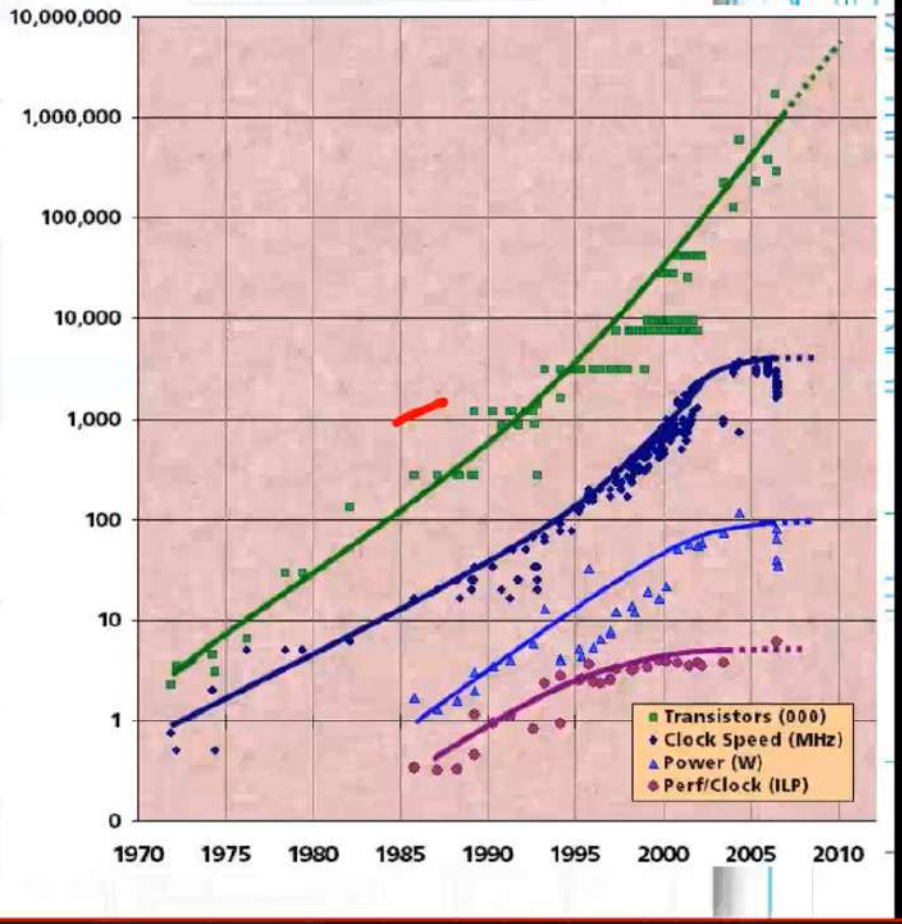
Microprocessors have become smaller, denser, and more powerful.

Clock scaling hits Power Density Wall



Revolution is happening

- Chip density is continuing to increase approx. 2 times every 2 years
  - Clock speed is not
- There is little or no hidden parallelism (ILP) to be found
- Parallelism must be exposed to and managed by software
- Number of processor cores may double instead

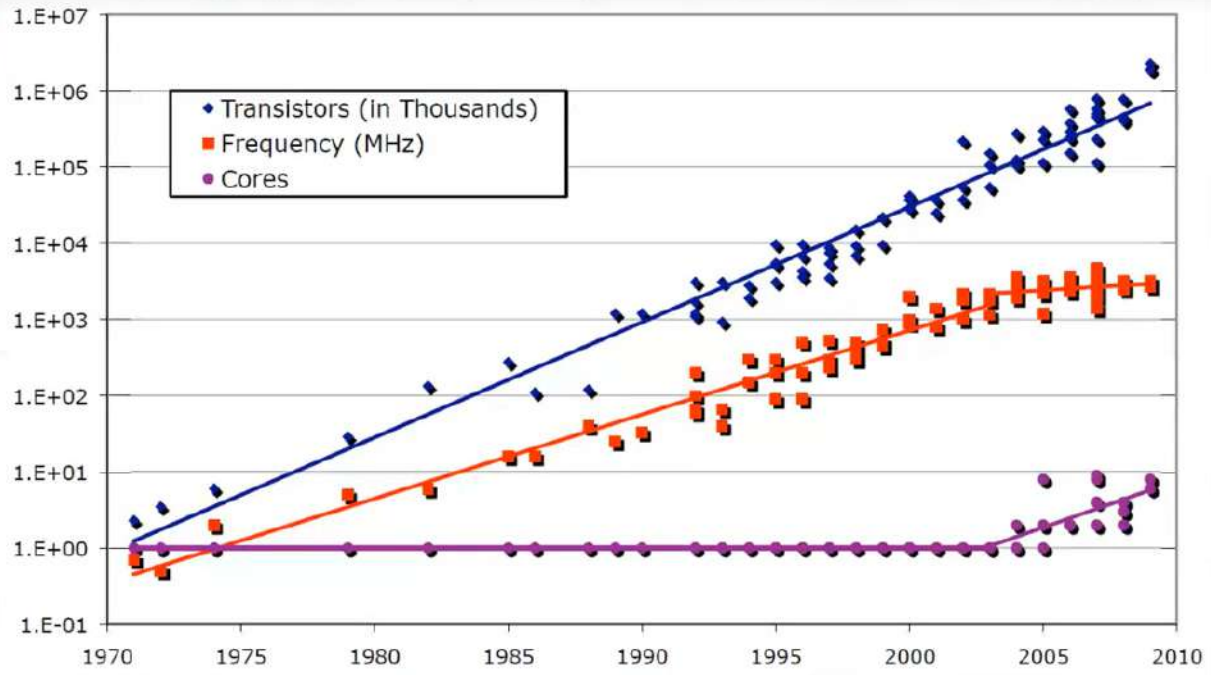


Source: Intel, Microsoft (Sutter) and Stanford (Olukotun, Hammond)

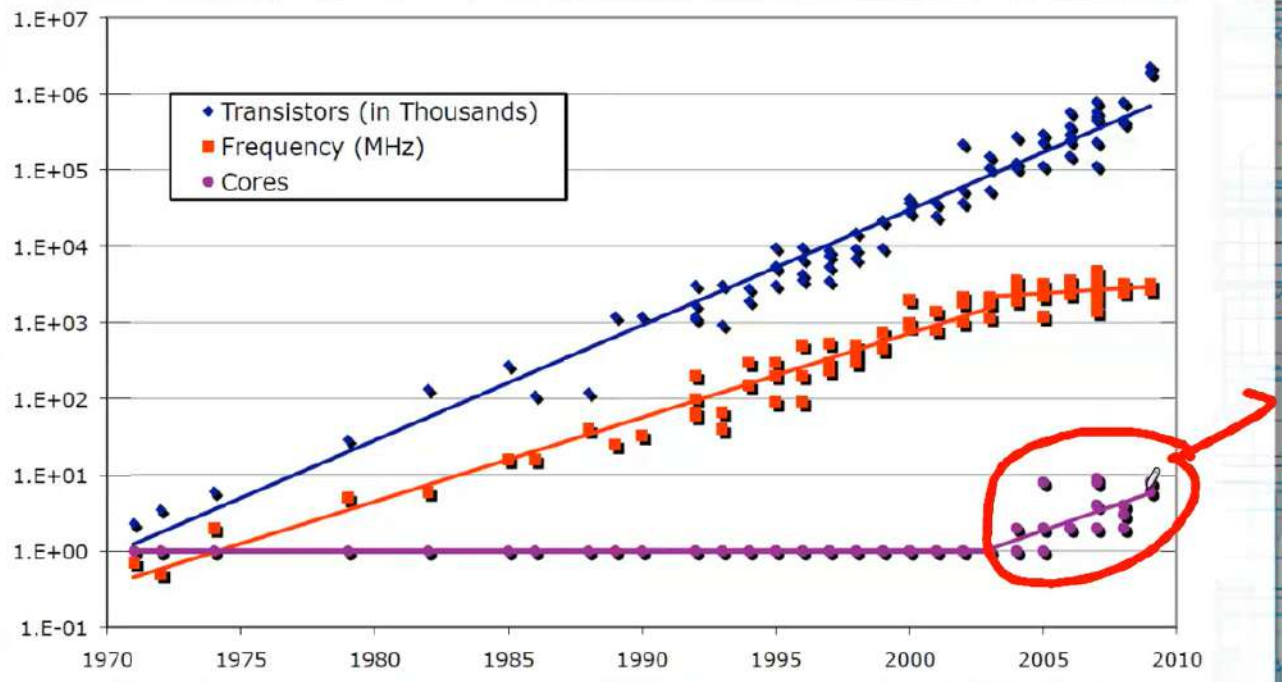
Processor designers forced to go “multicore”

- Heat density: faster clock means hotter chips
  - more cores with lower clock rates burn less power
- Declining benefits of “hidden” Instruction Level Parallelism (ILP)
  - Last generation of single core chips probably over-engineered
  - Lots of logic/power to find ILP parallelism, but it wasn’t in the applications
- Yield problems
  - Parallelism can also be used for redundancy
  - IBM Cell processor has 8 small cores; a blade system with all 8 sells for $20K, whereas a PS3 is about $600 and only uses 7

Moore's law



Moore's law

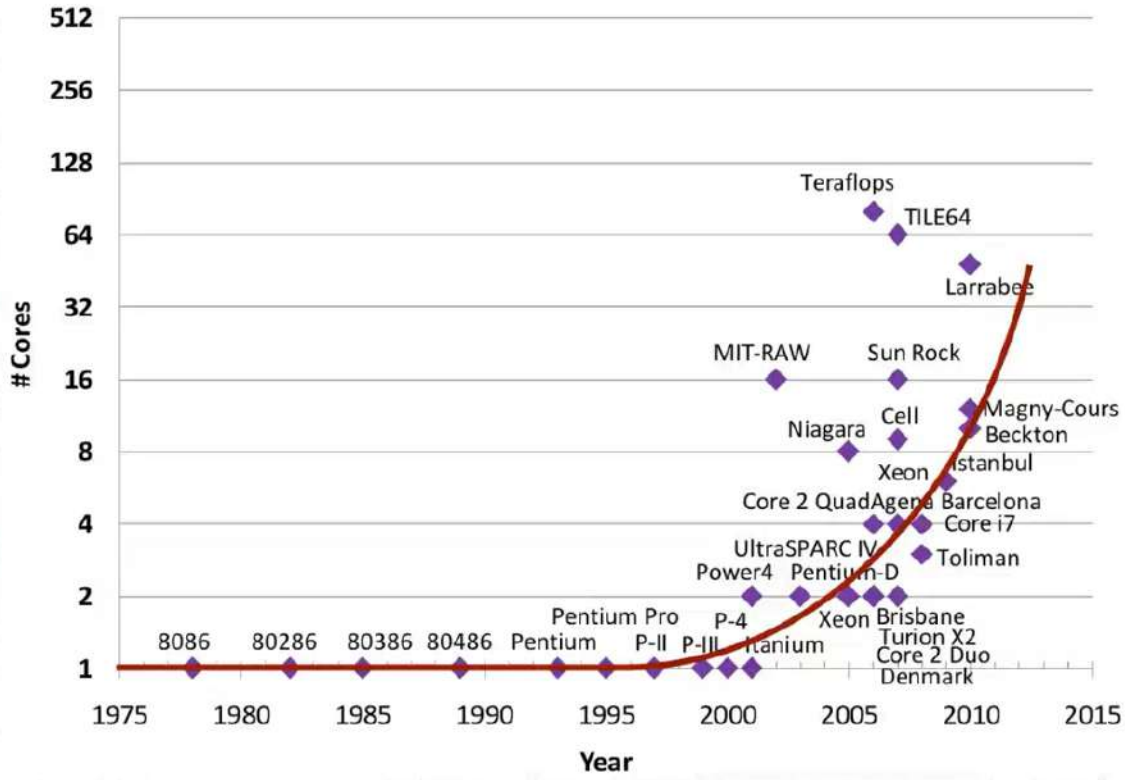


- "New" Moore's Law: 2x cores with every generation
- On-chip cache grows commensurately to supply all cores with data

Moore's law reinterpreted

- Number of cores per chip will double every two years
- Clock speed will not increase (possibly decrease)
- Need to deal with systems with millions of concurrent threads
- Need to deal with inter-chip parallelism as well as intra-chip parallelism

New Moore's law



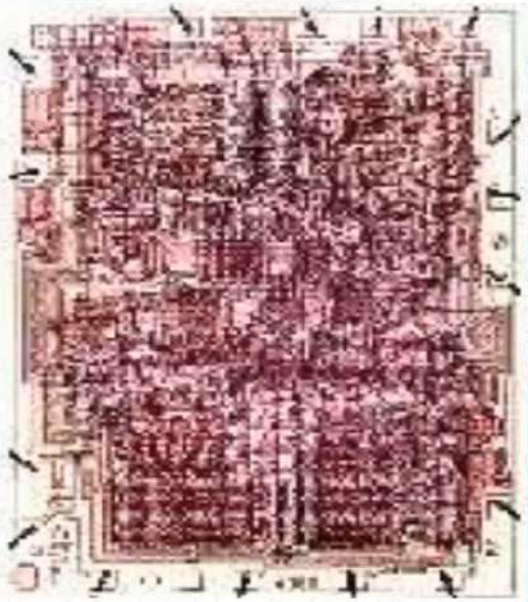
Number of cores/chip: Manycore

- "Multicore" 2X cores per generation: 2, 4, 8, …
- "Manycore" 100s of cores
- Multicore architectures & Programming Models good for 2 to 32 cores won't evolve to Manycore systems of 100's of processors
  ⇒ Desperately need HW/SW models that work for Manycore or will run out of steam
  (as ILP ran out of steam)
- We need revolution, not evolution

Multi-core examples

From “old” unicore

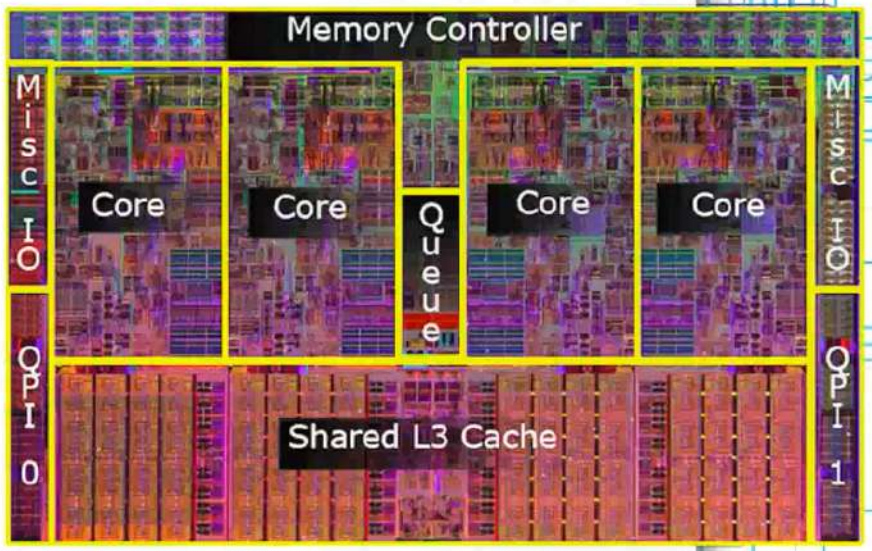
Intel 4004 (1971): 4-bit processor,
2312 transistors, ~100 KIPS,
10 micron PMOS, 11 mm^2 chip



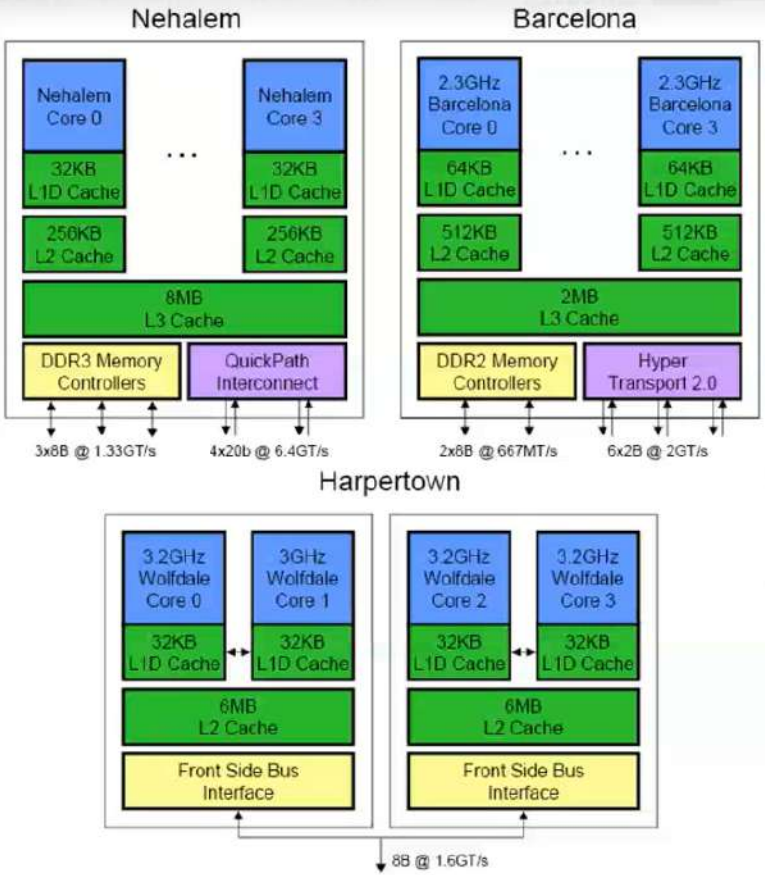
To Intel core i7

Nehalem: Quad core, 8 threaded, 64-bit,
4 issue superscalar, OoO, 16-stage pipeline,
48 bit virtual, 40 bit physical addressing

- 4 cores
- 731 million transistors, 263 mm^2 area, 45nm technology
- L3 is the last level cache, 8MB, 16-way set-assoc
- Each core has: 32KB, 8-way set-assoc, L1 (I and D), 256KB, 8-way set-assoc, L2
- Point-to-point interconnect called QuickPath

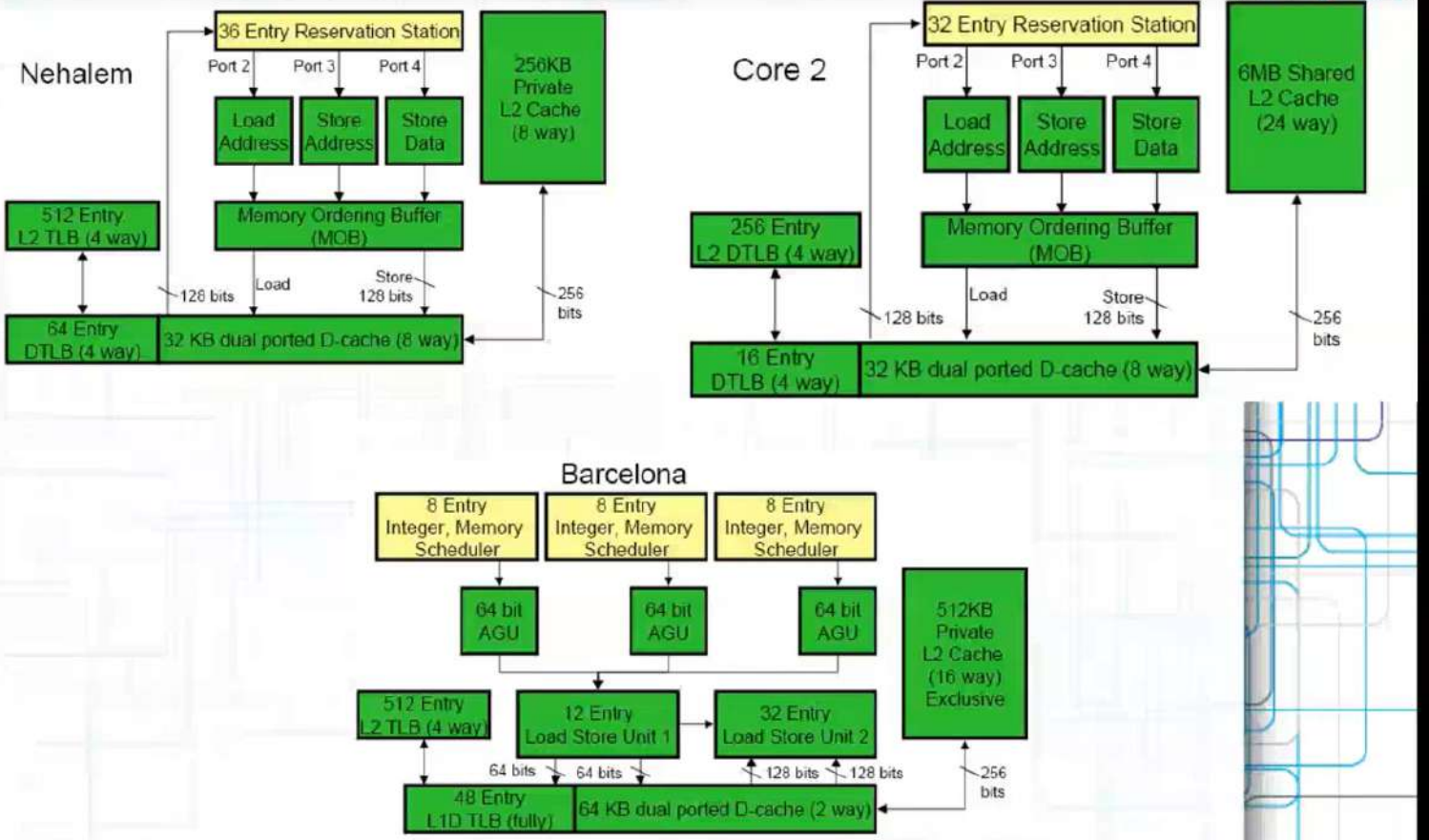


System Architecture comparison

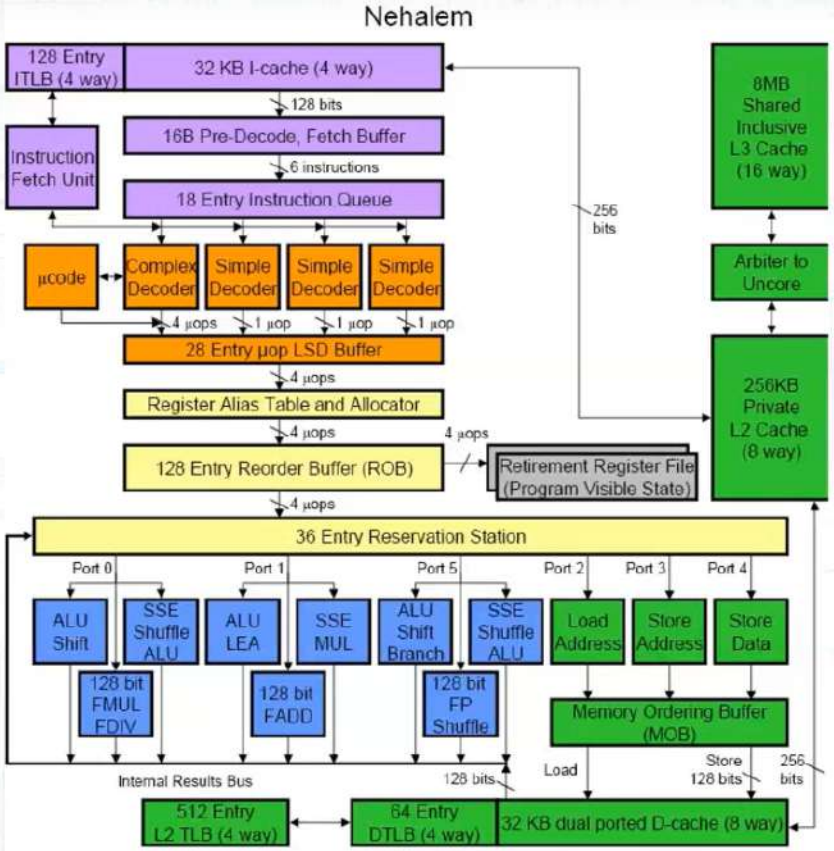


Nehalem, Harpentown: Intel
Barcelona: AMD

Memory subsystem comparison

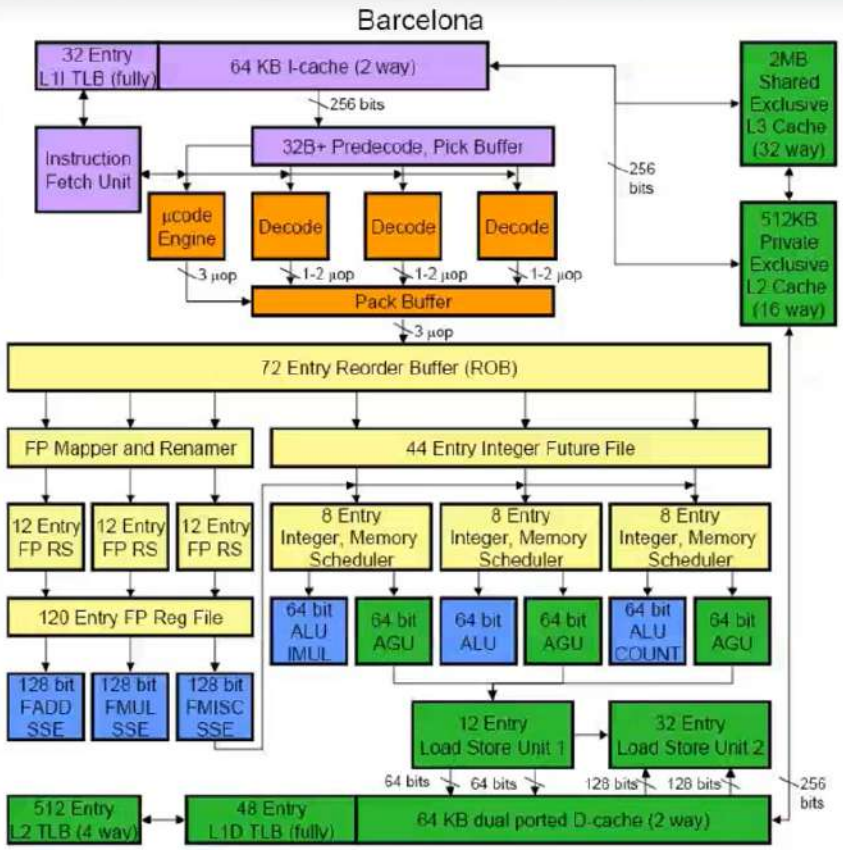


Nehalem architecture

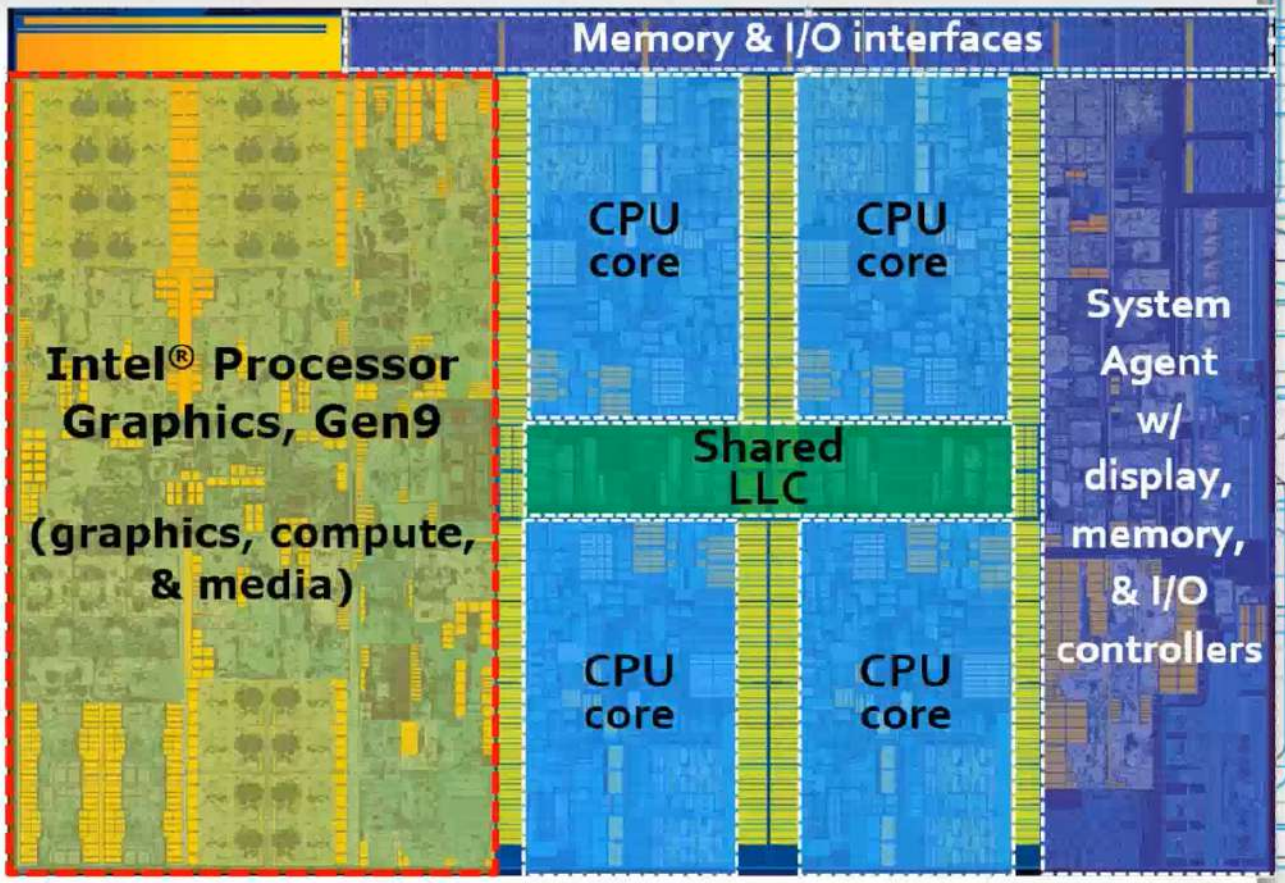


64-bit CPU, Register size = 64-bits
Virtual memory=2^{64}
Data bus=64 bits
64-bits can can be loaded/stored to memory

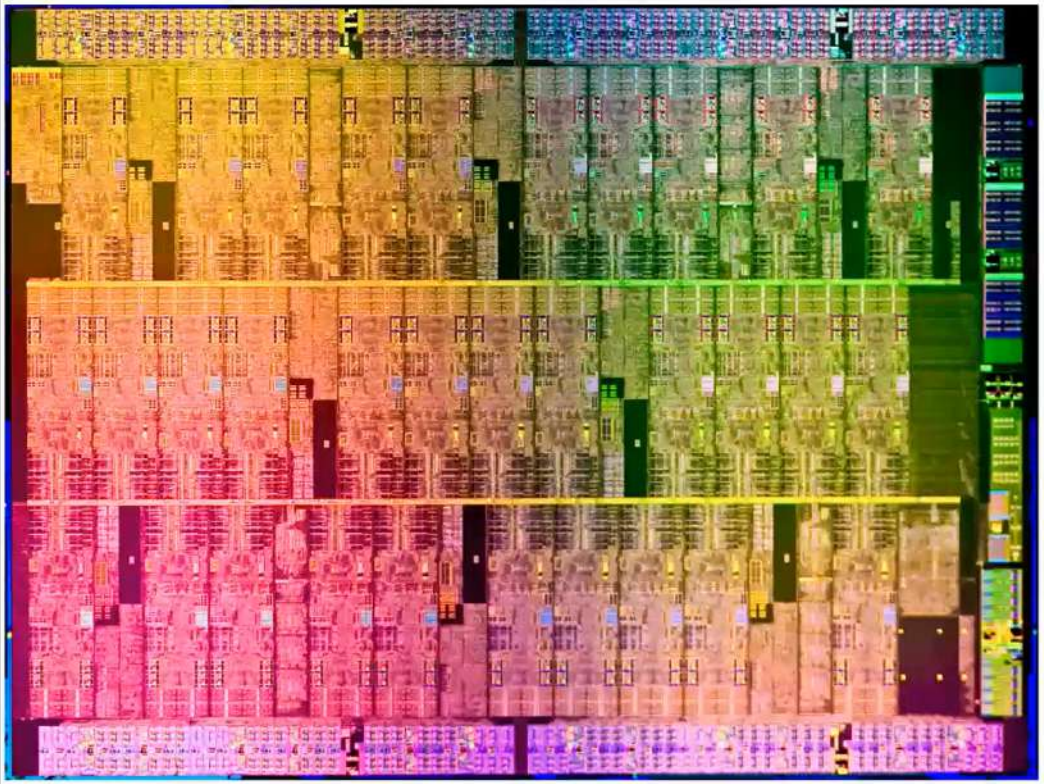
Barcelona architecture



Intel Skylake – core-i7 (6th gen)



Intel Xeon-Phi



61 simple x-86 cores, 1.3 GHz

Accelerator for supercomputing applications

TESLA V100

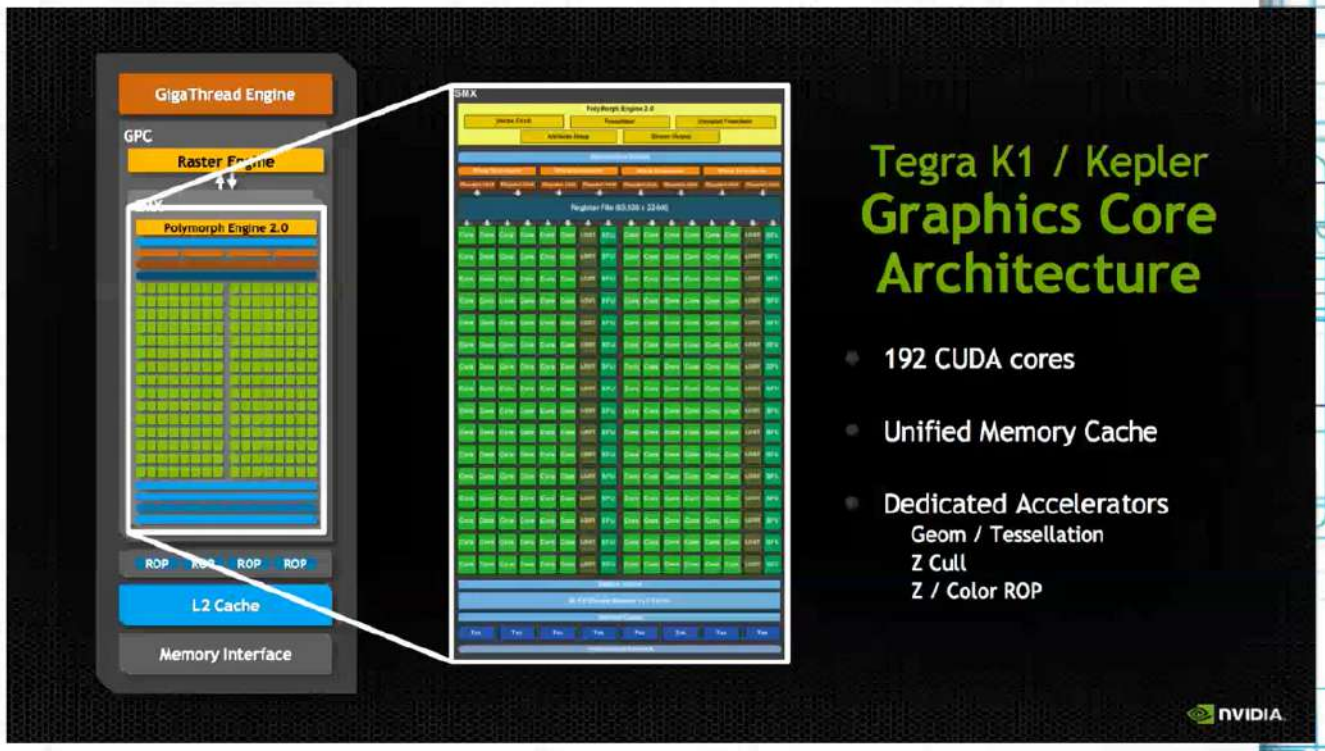
21B transistors
815 mm²

80 SM
5120 CUDA Cores
640 Tensor Cores

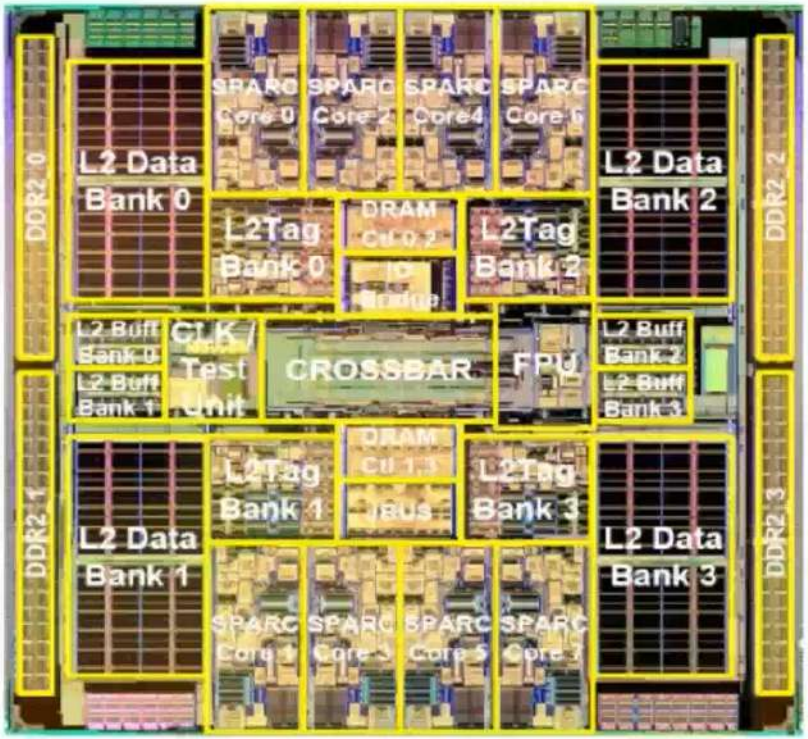
16 GB HBM2
900 GB/s HBM2
300 GB/s NVLink

*full GV100 chip contains 84 SMs

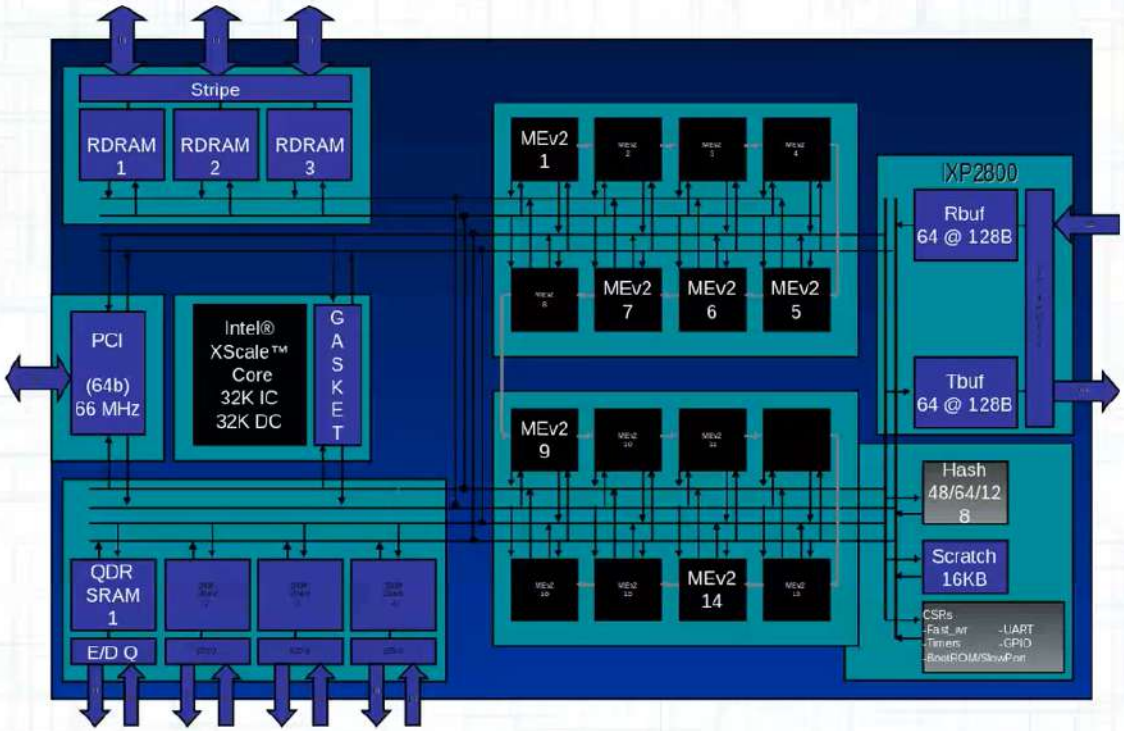
Mobile parallel processor



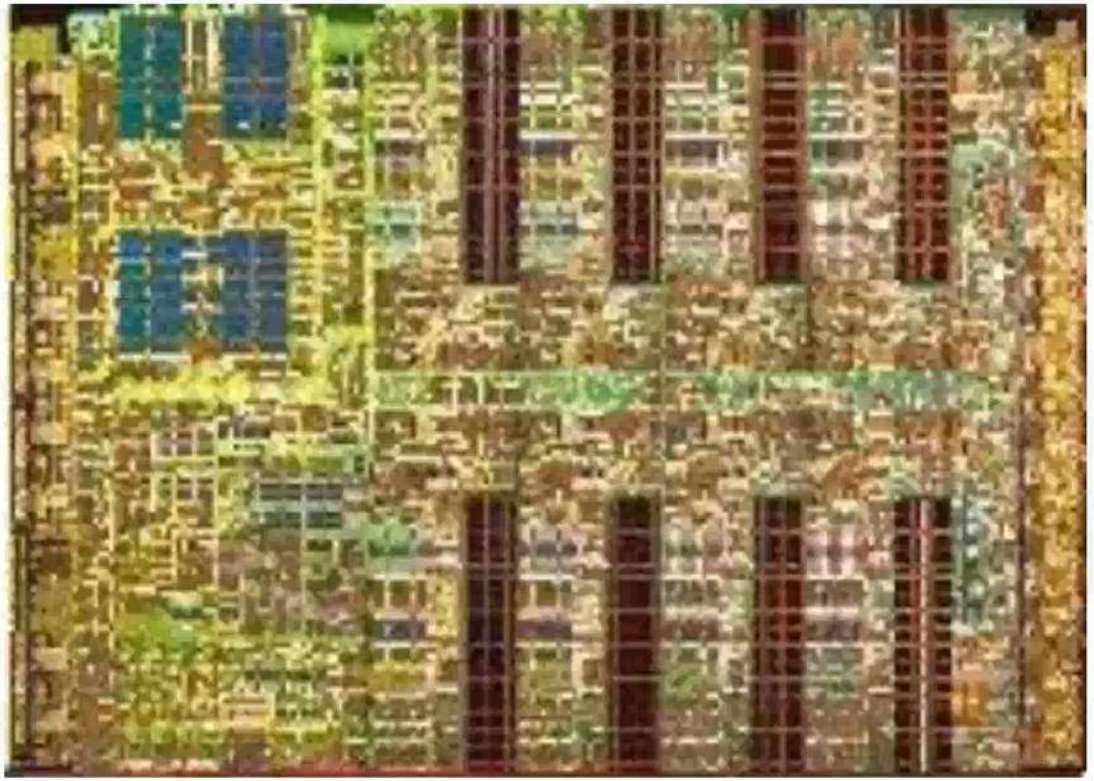
Sun Niagara
8 GPP cores (32 threads)



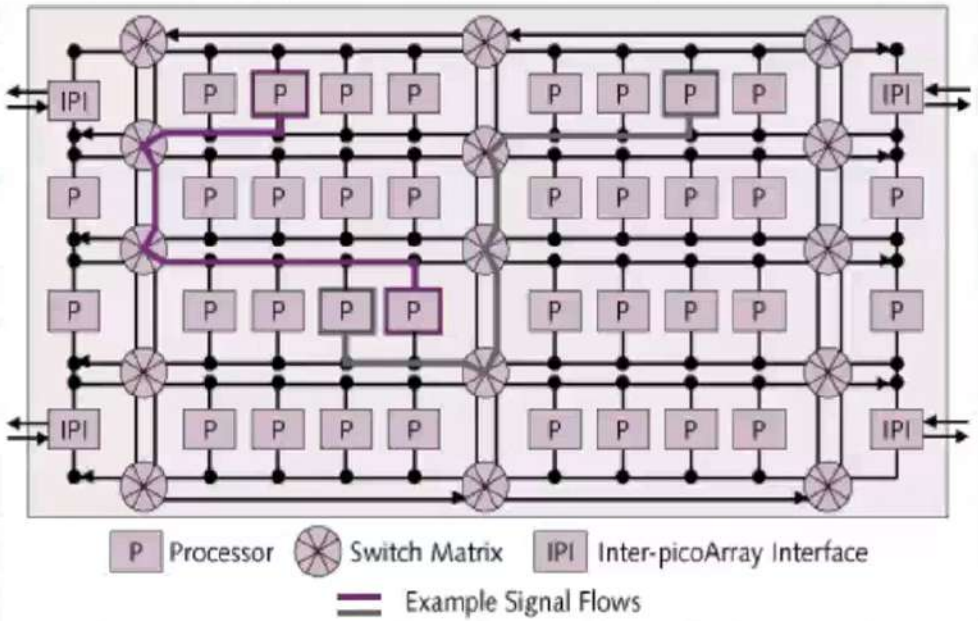
Intel Network Processor
1 GPP Core + 16 ASPs (128 threads)



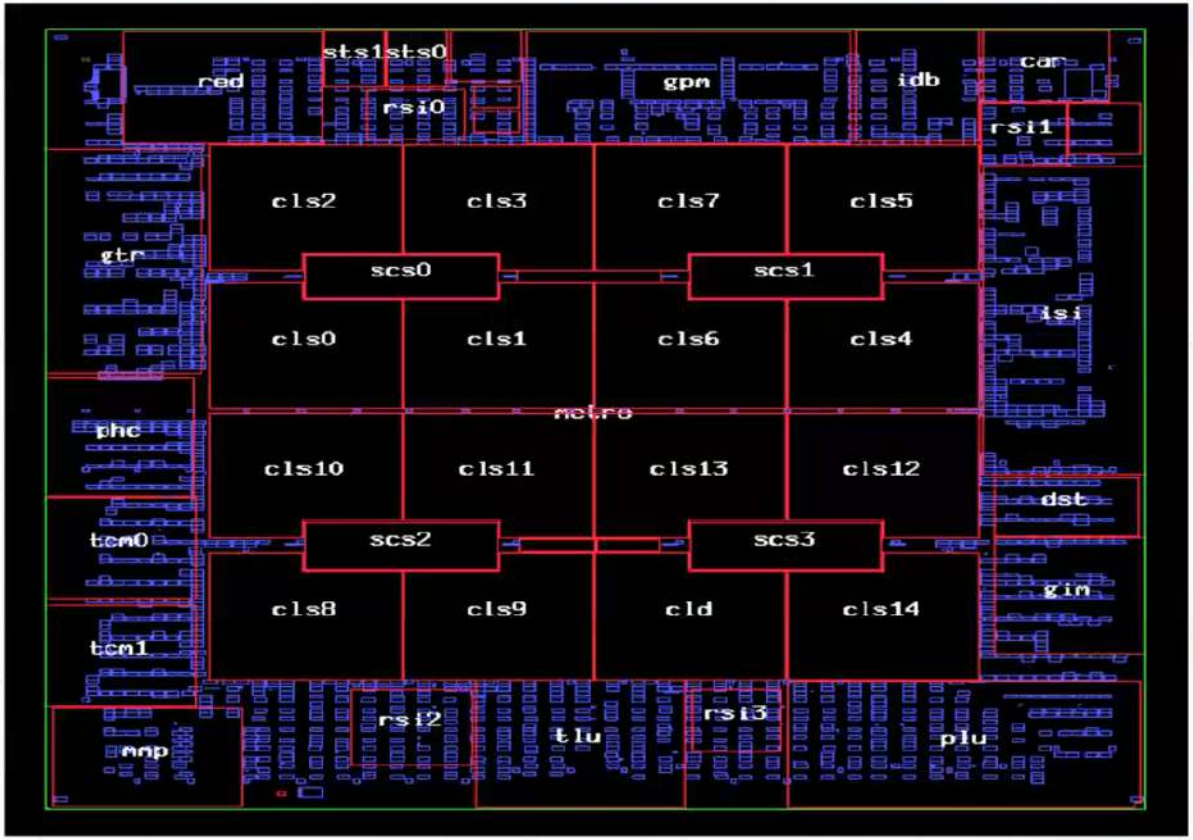
IBM Cell
1 GPP (2 threads) + 8 ASPs



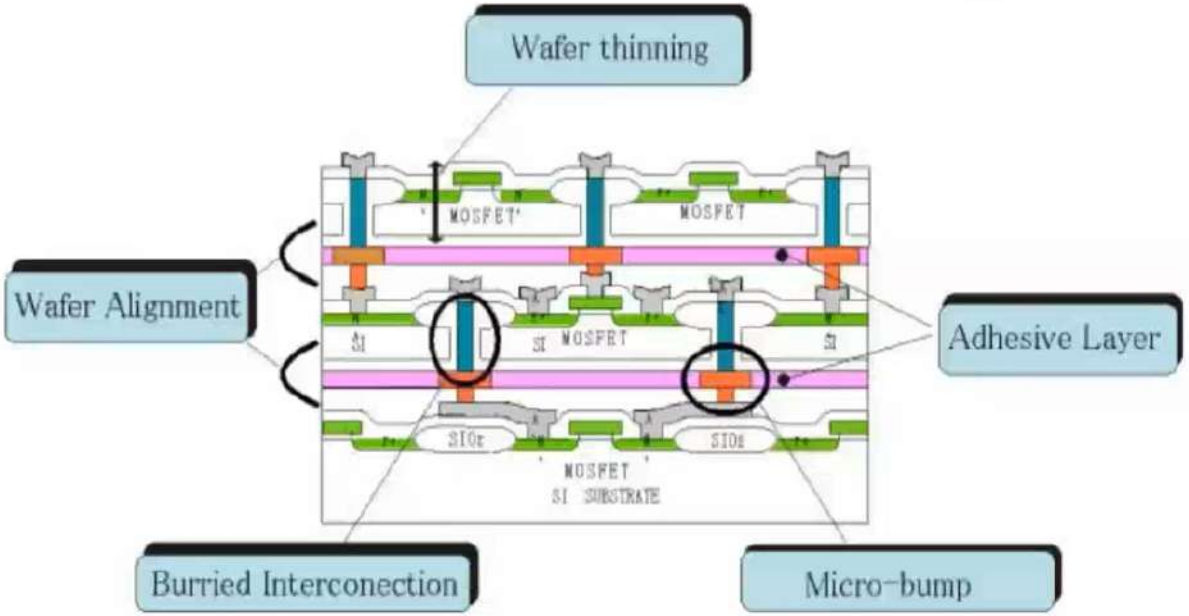
Picochip DSP
1 GPP core + 248 ASPs



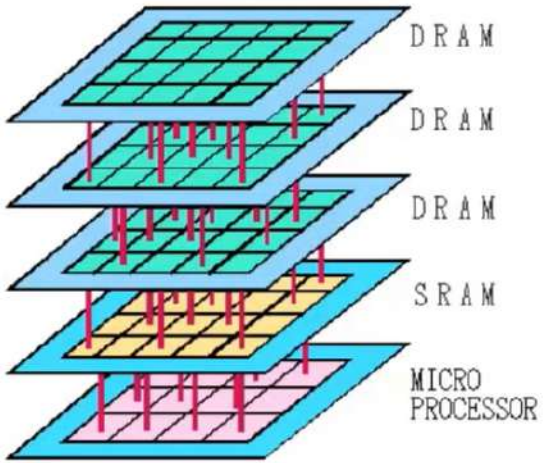
Cisco CRS-1
188 Tensilica GPPs

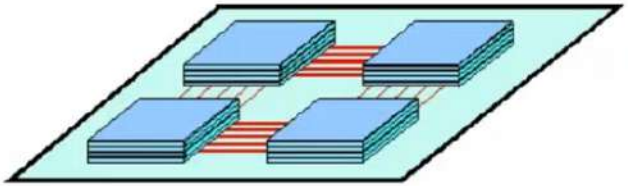


Key technology for 3D LSI



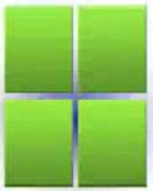
3D Computer Chip and 3D multi-chip Module



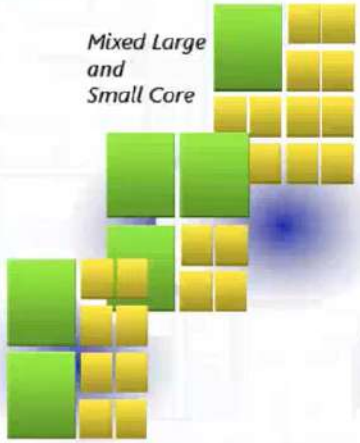


What's next

All Large Core



Mixed Large and Small Core

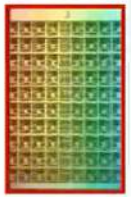


All Small Core

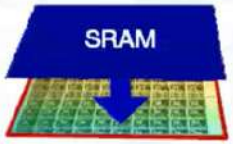
Many Small Cores

Different Classes of Chips
- Home
- Games / Graphics
- Business
- Scientific

Many Floating-Point Cores



+ 3D Stacked Memory



The question is not whether this will happen but whether we are ready