



National Energy Research Scientific Computing Center (NERSC)

Petascale Computing Application Challenges

More Processors
More Complexity
More Fear and Loathing

John Shalf
NERSC Center Division, LBNL



Supercomputing 2007
Reno, Nevada
November 13, 2007



Traditional Sources of Performance Improvement are Flat-Lining

- **New Constraints**
 - 15 years of *exponential* clock rate growth has ended
- **But Moore's Law continues!**
 - How do we use all of those transistors to keep performance increasing at historical rates?
 - Industry Response: #cores per chip doubles every 18 months *instead* of clock frequency!

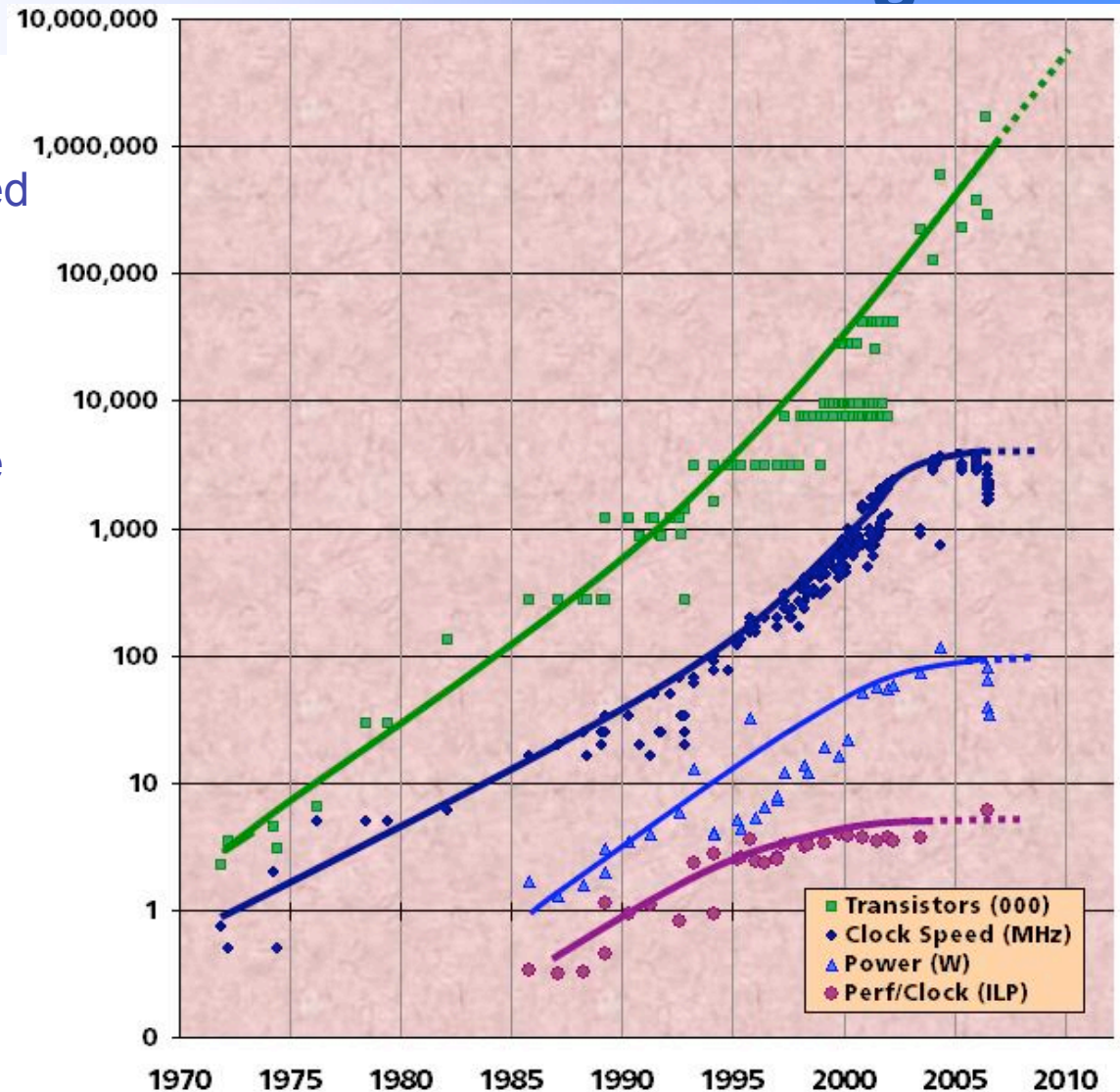


Figure courtesy of Kunle Olukotun, Lance Hammond, Herb Sutter, and Burton Smith



Is Multicore the Correct Response to New Lithography Constraints?

- Kurt Keutzer: “This shift toward increasing parallelism is not a triumphant stride forward based on breakthroughs in novel software and architectures for parallelism; instead, this plunge into parallelism is actually a retreat from even greater challenges that thwart efficient silicon implementation of traditional uniprocessor architectures.”
- David Patterson: “Industry has already thrown the hail-mary pass. . . But nobody is running yet.”

Scientists Ask Congress To Fund \$50 Billion Science Thing

SEPTEMBER 28, 2007 | ISSUE 43-39

Scientists from several major American universities appeared before a Congressional committee Monday to request \$50 billion for a science thing that would turner U.S. advancement science-wise and broaden human knowing.

ENLARGE IMAGE



"The [science thing] will make valuable inroads into our ultimate understanding of how [atoms and quarks move around and so on]."

David Kaminski,
Caltech Physicist

The scientists spoke for approximately three hours about the complicated science machine, which is expensive, and large, telling members of the House Committee on Science and Technology that the tubular, gamma-ray-using mechanism is vital in some big way. Yet the high price tag of the thing, which would be built on a 40-square-mile plot of land where the science would ultimately occur, remained a pressing question.

"While expense is something to consider, I think it's very important that we have this kind of scientific apparatus, because, in the end, I have always said that science is more important than it is unimportant," Committee chairman Rep. Bart Gordon (D-TN) said. "And it's essential we stay ahead of China, Japan, and Germany in science. We are ahead in space, with the NASA rockets going to other planets, so we should be ahead in science too."

According to the scientists, the electromagnetic science-maker will make atoms move and spin around very quickly, though spectators at the hearing said afterward they could not account for how one could get some atoms to move around faster than other ones if everything is made of atoms anyway. In addition, the scientists said that the device would be several miles in circumference, which puzzled onlookers who had long assumed that atoms were tiny. Despite these apparent inconsistencies, the scientists, in Rep. Gordon's words, appeared "very smart-sounding" and confident that their big spinner would solve some kind of problem they described.

The highlight of the scientists' testimony was a series of several colorful diagrams of how the big machine would work. One consisted of colored dots resembling Skittles banging into one another. Noting the motion lines behind the circle-ball things, committee members surmised that they were slamming together in a "fast, forceful manner." Yet some expressed doubts as to whether they justified the \$50 billion price tag.

"These scientists could trim \$10 million if they would just cut out some of the purple and blue spheres," said Rep. Roscoe Bartlett (R-MD), explaining that he understood the need for an abundance of reds and greens. "With all of those molecules and atoms going in every direction, the whole thing looks a bit unorganized, especially for science."

Another diagram presented to lawmakers contained several important squiggly lines, numbers, and letters. Despite not being numbers, the letters were reportedly meant to represent mathematics too. The scientists seemed to believe that correct math was what would help make the science thing go.

The scientists concluded their presentation by informing the committee that, if constructed correctly, the super science-flyer would be able to answer questions about many, many things, mainly stuff about the universe that

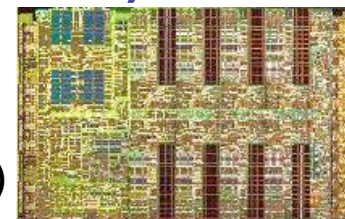
ENLARGE IMAGE



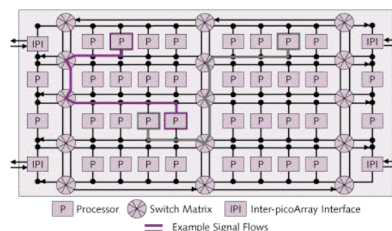


Convergence of Platforms

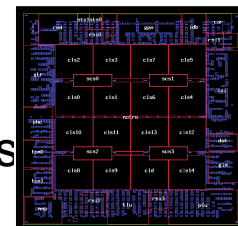
- Multiple parallel general-purpose processors (GPPs)
- Multiple application-specific processors (ASPs)



IBM Cell
1 GPP (2 threads)
8 ASPs

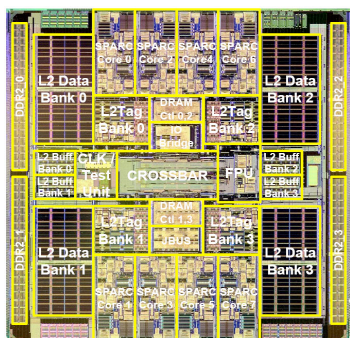
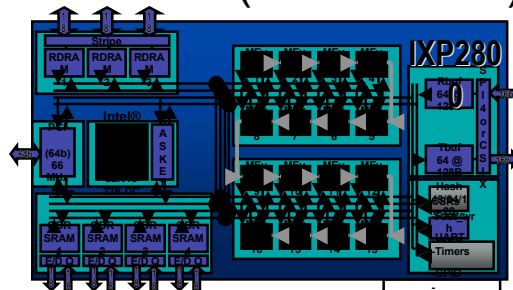


Picochip DSP
1 GPP core
248 ASPs

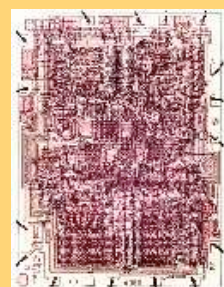


Cisco CRS-1
188 Tensilica GPPs

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



Sun Niagara
8 GPP cores (32 threads)



Intel 4004 (1971):
4-bit processor,
2312 transistors,
~100 KIPS,
10 micron PMOS,
11 mm² chip

1000s of
processor
cores per
die

***“The Processor is
the new Transistor”
[Rowen]***

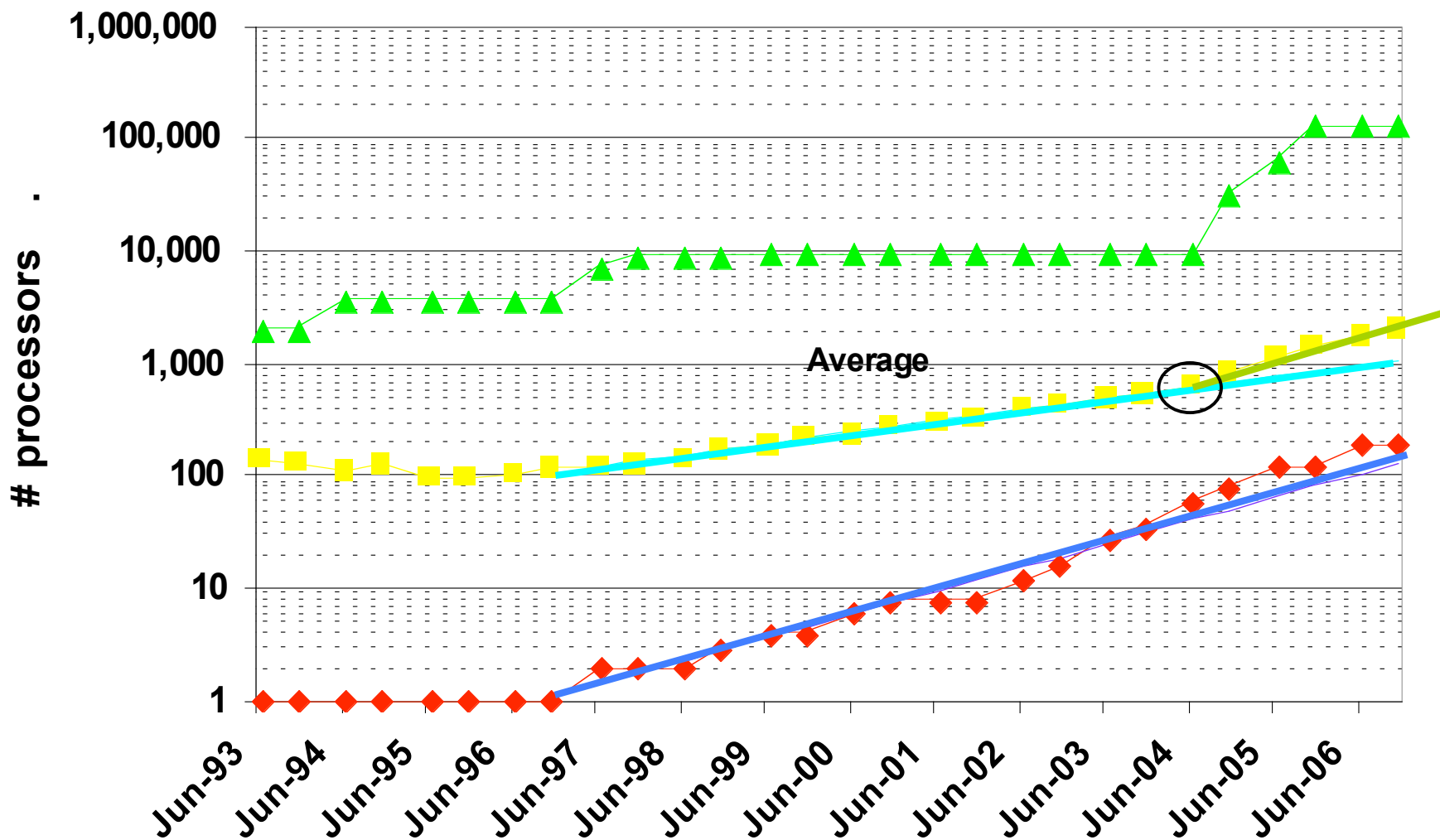


The *Entire* Computing Industry is Betting Its future on Parallelism

- **This transition is NOT just about HPC!**
 - Your Motorola Razor Cell Phone already has 8 Tensilica CPU cores in it (and will grow geometrically from there)
 - Cisco CRS-1 router has 188 tensilica CPU cores/socket (Metro) and scales to 400,000 cores! (more than HPC... runs an OS too!)
 - Your *toaster oven* is going be running parallel applications on manycore processors
- **Many key applications that motivate need for increased performance in consumer electronics are familiar scientific computing applications!**
- **Industry has already moved forward with parallelism without having a software solution in place (or even agreed upon)**



Concurrency Levels in TOP500



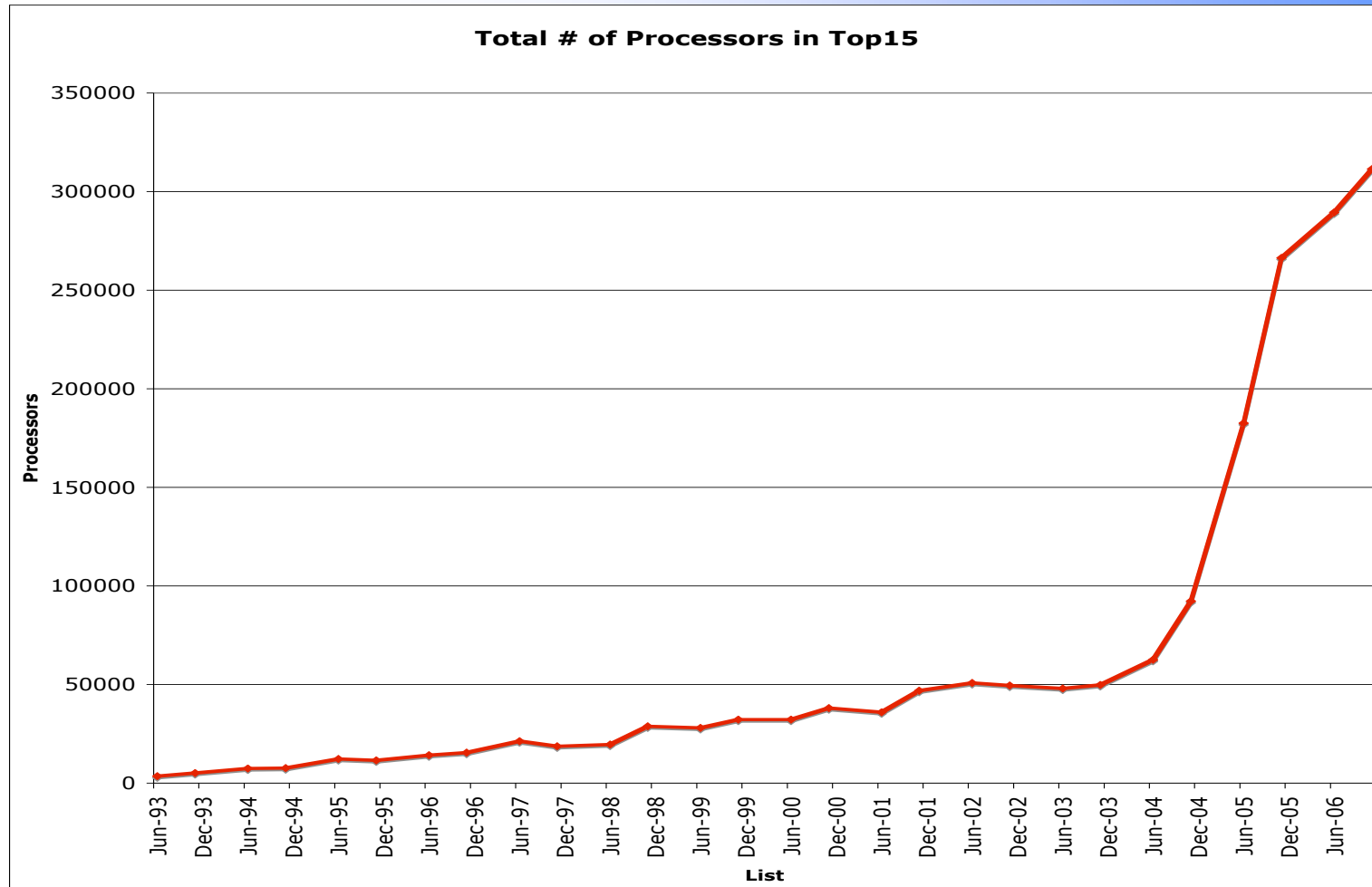


Top500 Trends

- **Teraflops Era**
 - **1997:** Teraflop/s system #1 on the list with 4,510 compute processors
 - **2004:** Requires 1 Teraflop just to enter Top500 List (hundreds of processors)
- **Petaflops Era**
 - **~2008:** Petaflop/s system #1 on the list with 40k-128k processors
 - **~2008 + 6-8 years:** Requires a petaflop just to enter Top500 list and it will still require 40k-128k processors
- **You cannot escape daunting concurrency!**



Humans Think in Terms of Linear Scales (hard to grok LOG() scale)



Must ride exponential wave of increasing concurrency for foreseeable future!

You will hit 1M cores sooner than you think!



Concerns about Multicore

(in the context of HPC)

- **System Balance:** Concern that memory and interconnect performance will ultimately cap multicore performance
- **Reliability:** More “moving parts” means more opportunity for failures
- **Programmability:** How can I possibly program 1M+ cores in an effective manner?



Memory Technology

- **Less Memory Bandwidth per core**
 - Balancing Little's Law is actually a bigger problem (and a much ignored problem)
 - Very much commodity price limited than technology limited
 - Bandwidth is going to force packaging changes (but they understand the technology to do so)
- **Less Memory Per core**
 - This is a cost issue
 - Currently, do not guarantee that we use all memory.
 - If memory is that costly, do we start sacrificing CPUs to get more memory? *(if they are equally expensive, then perhaps that's the right approach)*



Processor Technology

- **Multicore vs. Manycore**
 - Wait 3 years and you'll be at the same concurrency anyways
- **Deeper hierarchical architectures**
 - CMPs
 - NUMA effects on SMPs
 - Hierarchical interconnect fabrics (copper/optical)
 - *Locality Locality Locality*
- **Accelerators**
 - Please don't make me suffer Xilinx wire routing heuristics or hack on OpenGL (*Fortran is bad enough*)
 - Please don't make me write my program twice (*look at ISCA08 Kozyrakis*)
 - Please don't make me bounce memory around between accelerator and host (*it was our least-favorite feature of the CM-5 and its still no fun now*)



I/O For Massive Concurrency

- **Scalable I/O for massively concurrent systems!**
 - Many issues with coordinating access to disk within node (on chip or CMP)
 - OS will need to devote more attention to QoS for cores competing for finite resource (*mutex locks and greedy resource allocation policies will not do!*) (*it is rugby where device == the ball*)

nTasks	I/O Rate	
	16 Tasks/node	8 tasks per node
8	-	131 Mbytes/sec
16	7 Mbytes/sec	139 Mbytes/sec
32	11 Mbytes/sec	217 Mbytes/sec
64	11 Mbytes/sec	318 Mbytes/sec
128	25 Mbytes/sec	471 Mbytes/sec



Old OS Assumptions are Bogus for Hundreds of Cores!

- **Assumes limited number of CPUs that must be shared**
 - *Old OS: time-multiplexing (context switching and cache pollution!)*
 - *New OS: spatial partitioning*
- **Greedy allocation of finite I/O device interfaces (eg. 100 cores go after the network interface simultaneously)**
 - *Old OS: First process to acquire lock gets device (resource/lock contention! Nondeterm delay!)*
 - *New OS: QoS management for symmetric device access*
- **Background task handling via threads and signals**
 - *Old OS: Interrupts and threads (time-multiplexing) (inefficient!)*
 - *New OS: side-cores dedicated to DMA and async I/O*
- **Fault Isolation**
 - *Old OS: CPU failure --> Kernel Panic (will happen with increasing frequency in future silicon!)*
 - *New OS: CPU failure --> Partition Restart (partitioned device drivers)*
- **Inter-Processor Communication**
 - *Old OS: invoked for ANY interprocessor communication or scheduling*
 - *New OS: direct HW access mediated by hypervisor*



Concerns about Programmability

- Widespread panic regarding a programming model that can ride the “Tsunami of concurrency”
- “*Be afraid. . . Be Very Afraid.*” Ken Kennedy SC06



“The Processor is the new Transistor”

(Chris Rowen: Tensilica)

- NERSC’s 1999 flagship computing system, seaborg, contained as many processors as there are transistors in the original Intel 8080a implementation (6,000 transistors vs 6,000 processors)
 - Seaborg’s replacement, franklin, has 20,000 processors!
- BG/L at LLNL contains as many processors as there are transistors in the MC68000 (manufactured in 1980, the MC68000L was a 32-bit processor and contained 68,000 transistors).
- The next generation of BlueGene is likely to have more processors than there are logic gates in its constituent processing elements. (is that ironic or is it outrageous?)



The complexity of a Petascale system is exceeding the complexity of its components

- Applications developers today write programs that are as complex as describing where every single bit must move between the 6,000 transistors of the 8080a.
- We need to at **least** get to the “assembly language” level.
- We may need to reconsider our entire hardware/software programming model if this is indeed what the future holds for us.



Programmability

- **Widespread panic over programming model that can ride the “Tsunami of concurrency”**
- **Inter-dependent requirements for programming environment**
 - Productivity
 - Performance
 - Correctness
- **Approaches**
 - Abstracting single-chip parallelism
 - Focus of the Broader Consumer Electronics/Computing Industry
 - Even in HPC, observe that # chips growing much slower than # cores
 - Hiding complexity of global parallelism
 - Frameworks, Advanced compilers and programming languages, Auto-tuning
 - Nightmare Scenario: Microsoft solves in-socket programming model and we are stuck writing MPI between sockets that run C# code!
- **Competing Goals**
 - Productivity Layer: *Simplify specification of program/problem to solve*
 - Performance Layer: *Expose all hardware Capabilities to programmer*



Multicore is NOT a Familiar Programming Target

- **What about Message Passing on a chip?**
 - MPI buffers & datastructures growing $O(N)$ or $O(N^2)$ a problem for constrained memory
 - Redundant use of memory for shared variables and program image
 - *Flat view of parallelism doesn't make sense given hierarchical nature of multicore sys.*
- **What about SMP on a chip?**
 - Hybrid Model (MPI+OpenMP) : *Long and mostly unsuccessful history*
 - But it is NOT an SMP on a chip
 - 10-100x higher bandwidth on chip
 - 10-100x lower latency on chip
 - SMP model ignores potential for much tighter coupling of cores
 - *Failure to exploit hierarchical machine architecture will drastically inhibit ability to efficiently exploit concurrency! (requires code structure changes)*
- **Looking beyond SMP**
 - Cache Coherency: *necessary but not sufficient (and not efficient for manycore!)*
 - Fine-grained language elements difficult to build on top of CC protocol
 - Hardware Support for Fine-grained hardware synchronization
 - Message Queues: direct hardware support for messages
 - Transactions: Protect against incorrect reasoning about concurrency



Application Code Complexity

- **Application Complexity has Grown**
 - Big Science is a multi-disciplinary, multi-institutional, multi-national efforts! (and we are not just talking about particle accelerators and Tokamaks)
 - Looking more like science on atom-smashers
- **Advanced Parallel Languages Necessary, but NOT Sufficient!**
 - Need higher-level organizing constructs for teams of programmers



Application Code Complexity

Large-Scale Electronic Structure Calculations of High-Z Metals on the BlueGene/L Platform

Francois Gygi
Department of Applied Science
University of California, Davis
Davis, CA 95616
530-752-4042
fgygi@ucdavis.edu

Erik W. Draeger, Martin Schulz,
Bronis R. de Supinski
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
Livermore, CA 94551
{draeger1,schulz6,bronis}@llnl.gov

John A. Gunnels, Vernon Austel,
James C. Sexton
IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598
{gunnels,austel,sextonjc}@us.ibm.com

Franz Franchetti
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
franzf@ece.cmu.edu

Stefan Kral, Christoph W. Ueberhuber, Juergen Lorenz
Institute of Analysis and Scientific Computing
Vienna University of Technology, Vienna, Austria
skral@mips.complang.tuwien.ac.at, c.ueberhuber@tuwien.ac.at
juergen.lorenz@aurora.anum.tuwien.ac.at

- **QBox: Gordon Bell Paper title page**
 - Its just like particle physics papers!
 - *Looks like discovery of the Top Quark!*



Community Codes & Frameworks

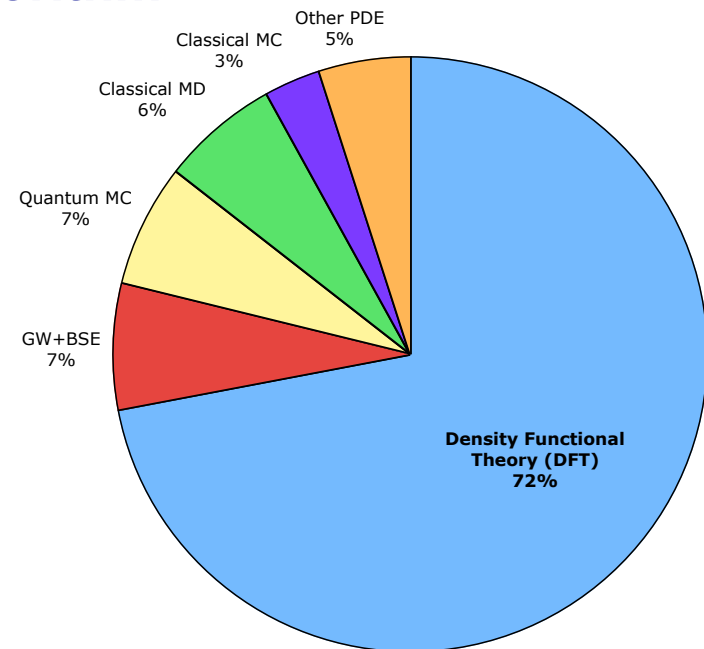
(hiding complexity using good SW engineering)

- **Frameworks (eg. Chombo, Cactus, SIERRA, UPIC, etc...)**
 - Clearly separate roles and responsibilities of your expert programmers from that of the domain experts/scientist/users (productivity layer vs. performance layer)
 - Define a *social* contract between the expert programmers and the domain scientists
 - Enforces and facilitates SW engineering style/discipline to ensure correctness
 - Hides complex domain-specific parallel abstractions from scientist/users to enable performance (hence, most effective when applied to community codes)
 - Allow scientists/users to code nominally serial plug-ins that are invoked by a parallel “driver” (either as DAG or constraint-based scheduler) to enable productivity
- **Properties of the “plug-ins” for successful frameworks (CSE07)**
 - Relinquish control of main(): invoke user module when framework thinks it is best
 - Module must be stateless
 - Module only operates on the data it is handed (no side-effects)
- **Frameworks can be thought of as driver for coarse-grained dataflow**
 - Very much like classic static dataflow, except coarse-grained objects written in declarative language (dataflow without the functional languages)
 - Broad flexibility to schedule Directed Graph of dataflow constraints
 - See *also* Jack Dongarra & Parry Husbands’ work on DAG-based scheduling



Density Functional Theory (DFT) Algorithm

- **Kohn-Sham formalism for computing electronic structure from first principles (DFT Method)**
 - Most common implementation is based on expanding the quantum wavefunction into plane-wave (fourier) components
 - This is the method employed by VASP, PARATEC, and Qbox
- **Dominant phases of planewave DFT algorithm**
 - **3D FFT**
 - transforming between real space and reciprocal space
 - $O(N_{\text{atoms}}^2)$ complexity
 - **Subspace Diagonalization**
 - $O(N_{\text{atoms}}^3)$ complexity
 - **Orthogonalization**
 - dominated by BLAS3
 - $\sim O(N_{\text{atoms}}^3)$ complexity
 - **Compute Non-local pseudopotential**
 - $O(N_{\text{atoms}}^3)$ complexity





Ramifications of DFT Algorithm Characteristics

- For smaller atomic systems (~600-1000 atoms)
 - BLAS dominates at lower concurrencies
 - 3D FFT tends to dominate the computation at high concurrency
 - Due to low computational intensity and small message size (NSF Track-2 bench)
 - Message size can be increased by expending more memory/processor
- For larger atomic systems (>1k atoms), the $O(N^3)$ complexity of orthogonalization and computing non-local pseudopotential will dominate
- For $O(N^3)$ complexity, moving from teraflops to petaflops only gets you from 1k atoms to 4k atoms.
 - not very impressive given the amount of hardware!
 - Good news is that FLOP rates will be very impressive given increased domination of highly localized BLAS3 operations (eg QBox example)
- *For this reason, we argue that DFT will be gradually supplanted by $O(N)$ methods as we move into Petaflop scale calculations!*



Anatomy of an $O(N)$ DFT method

(LS3DF as an example)

- **Total energy of a system can be decomposed into two parts**
 - **Quantum mechanical part:**
 - wavefunction kinetic energy and exchange correlation energy
 - Highly localized
 - Computationally expensive part to compute
 - **Classical electrostatic part:**
 - Coulomb energy
 - Involves long-range interactions
 - Solved efficiently using poisson equation even for million atom systems
 - **LS3DF takes advantage of localization of quantum mechanical part of calculation**
 - Divide computational domain into discrete tiles and solve quantum mechanical part
 - Solve global electrostatic part (no decomposition)
 - Very little interprocessor communication required! (almost embarrassingly parallel)
 - Result is $O(N_{\text{atoms}})$ complexity algorithm: enables exploration of larger atomic systems as we move to petaflop and beyond.



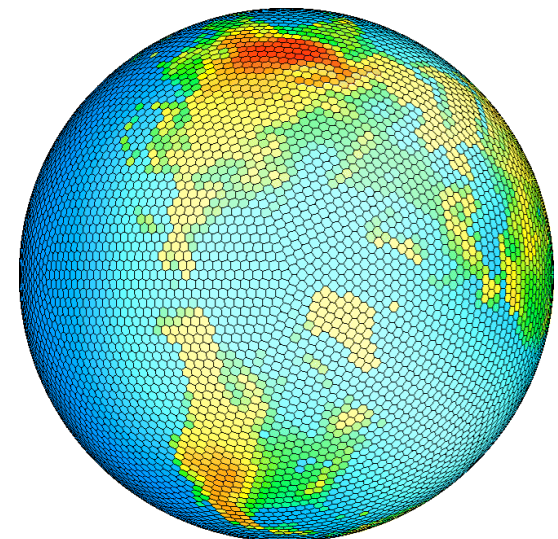
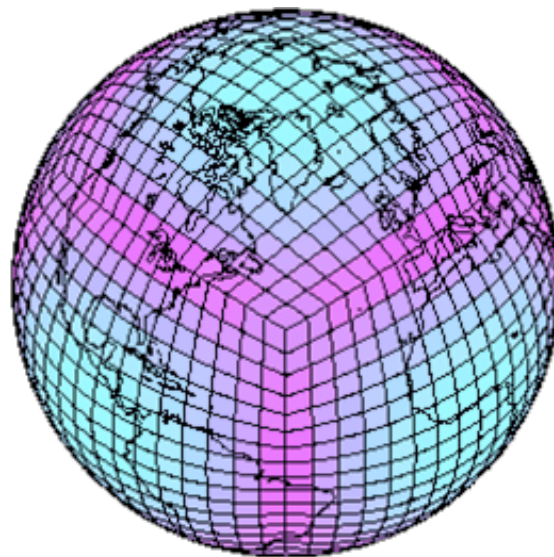
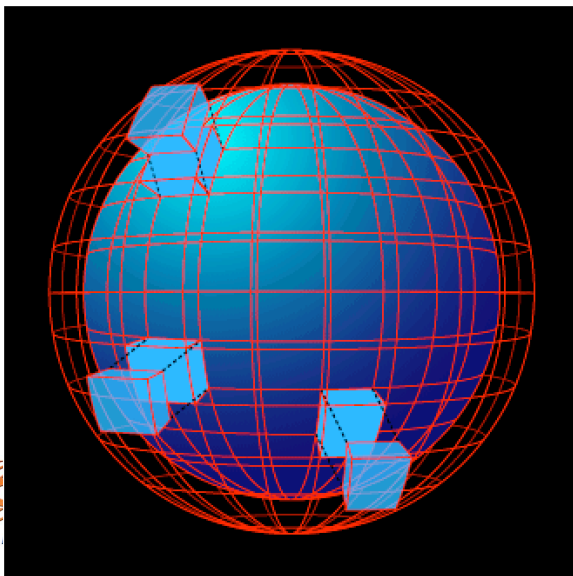
Conclusion for Materials Science

- **Density Functional Theory codes (particularly planewave DFT) dominates material science workload**
- **Petaflop machines will only enable exploration of modestly larger atomic systems**
 - due to $O(N^3)$ algorithmic complexity
 - Move from teraflops to petaflops gets from 1k atoms to 4k atoms
 - But with larger atomic systems, computational efficiency will look fantastic due as calculation is increasingly dominated by highly localized pBLAS3 operations (often hand-tuned vendor libraries)
- **Exploration of 10k or greater atomic systems at petaflop scale will ultimately require a move to new methods offering $O(N)$ complexity**
- **There will still be a use for conventional DFT for 1k atomic systems for time-domain DFT:**
 - Explore same size system, but for longer timescales
 - Will be extraordinarily demanding parallelization requirements for the concurrencies presented by Petaflop-scale systems



Cloud System Resolving Climate Simulation

- Computational Models Help answer question with multi-trillion dollar ramifications!!!
- A major source of errors in climate models is poor cloud simulation
- At ~ 1 km horizontal resolution, cloud systems can be resolved
- Requires, new discretizations, significant algorithm work and unprecedented concurrencies to maintain 1000x faster than realtime performance!





Petascale Architectural Exploration

Processor	Clock	Peak/ Core (Gflops)	Cores/ Socket	Mem/ BW (GB/s)	Network BW (GB/s)	Sockets	Power <i>(based on current generation technology)</i>
AMD Opteron	2.8GHz	5.6	2	6.4	4.5	890K	179 MW
IBM BG/L	700MHz	2.8	2	5.5	2.2	1.8M	27 MW
Semicustom Embedded	650MHz	2.7	32	51.2	34.5	120K	3 MW

- ❖ Software challenges (at all levels) are a tremendous obstacle for any of these approaches.
 - ❖ Unprecedented levels of concurrency are required.
 - ❖ Unprecedented levels of power are required if we adopt conventional route
 - ❖ Embedded route offers tractable power, but daunting concurrency!
- ❖ This only gets us to 10 Petaflops *peak* -
 - ❖ 200PF system to meet application sustained performance requirements
 - ❖ thus cost and power are likely to be 10x-20x more.



It's the MATH Stupid!!!

- **The broader industry is concerned about architecture and programming model**
- **HPC applications may also need to reformulate the numerical model at petaflop scale**
 - This takes more time than the other stuff!
 - It is more labor intensive than the other stuff!
 - The outcome even less predictable than the other stuff!
 - V&V isn't getting any easier!!!



Conclusions

- **Enormous transition is underway that affects all sectors of computing industry**
 - Motivated by power limits
 - Proceeding before emergence of the parallel programming model
- **Will lead to new era of architectural exploration given uncertainties about programming and execution model (and we MUST explore!)**
- **Need to get involved now**
 - 3-5 years for new hardware designs to emerge
 - 3-5 years lead for new software ideas necessary to support new hardware to emerge
 - 5+ MORE years to general adoption of new software



More Info

- **The Berkeley View**
 - <http://view.eecs.berkeley.edu>
- **NERSC Science Driven System Architecture Group**
 - <http://www.nersc.gov/projects/SDSA>

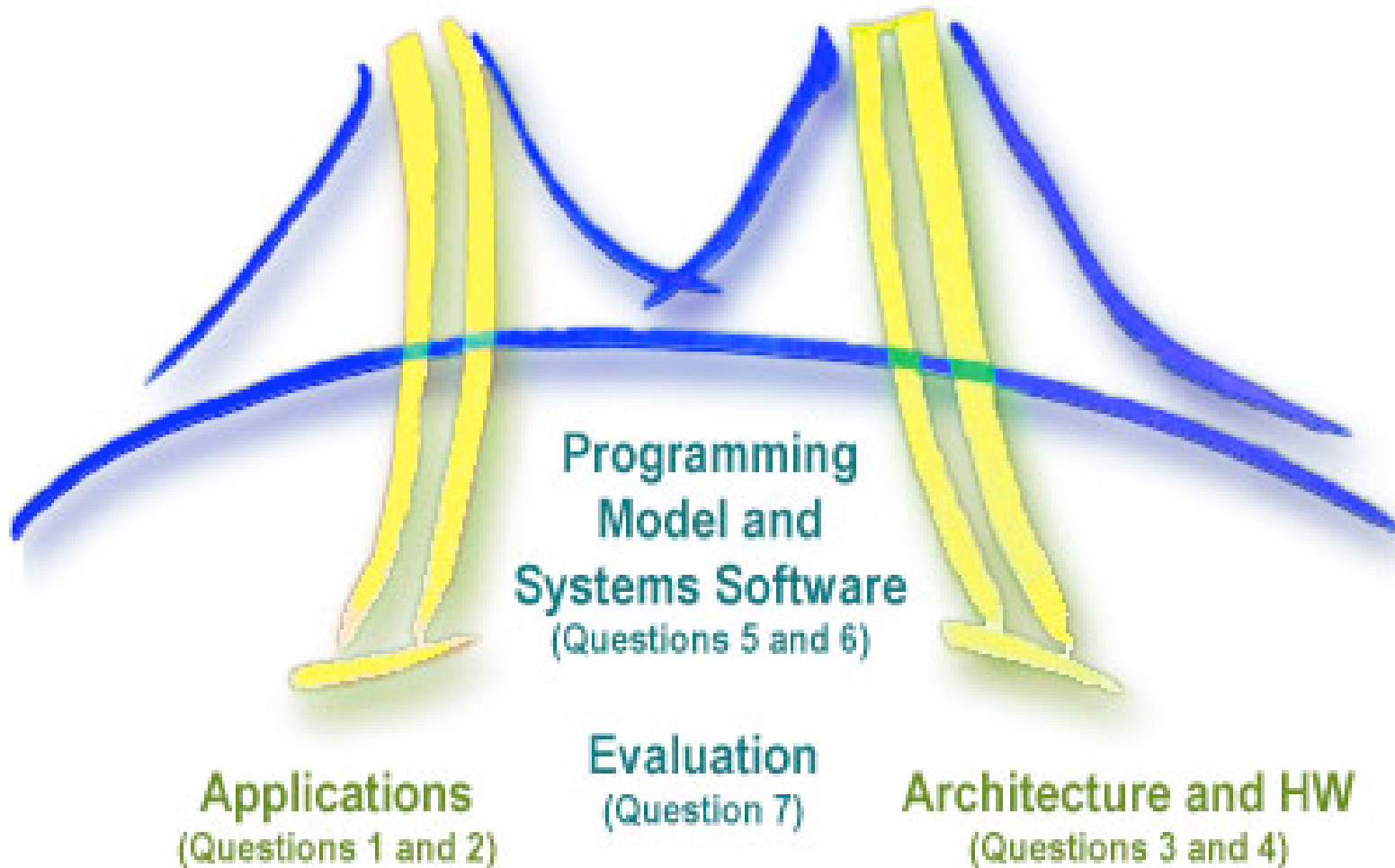




Extra Material



Landscape of Parallel Computing Architecture





Reliable System Design

- **The future is unreliable**
 - Silicon Lithography pushes towards the atomic scale, the opportunity for spurious hardware errors will increase dramatically
- **Reliability of a system is not necessarily proportional to the number of cores in the system**
 - Reliability is proportional to # of sockets in system (not #cores/chip)
 - At LLNL, BG/L has longer MTBF than Purple despite having 12x more processor cores
 - Integrating more peripheral devices onto a single chip (e.g. caches, memory controller, interconnect) can further reduce chip count and increase reliability (System-on-Chip/SOC)
- **A key limiting factor is software infrastructure**
 - Software was designed assuming perfect data integrity (but that is not a multicore issue)
 - Software written with implicit assumption of smaller concurrency (1M cores not part of original design assumptions)
 - Requires fundamental re-thinking of OS and math library design assumptions

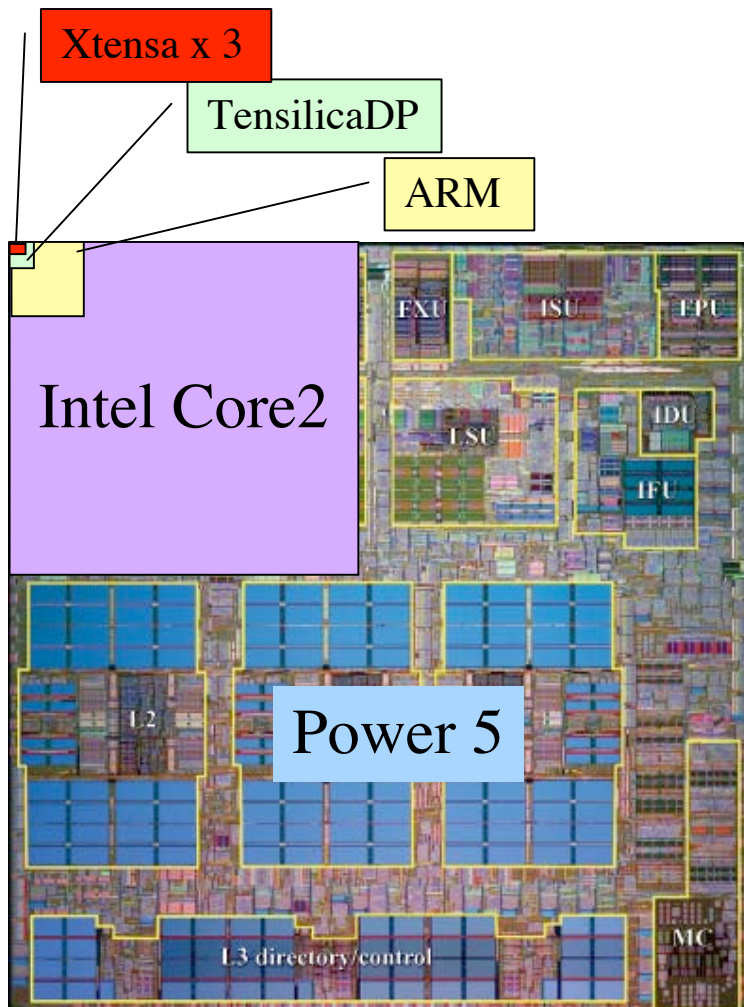


New Design Constraint: *POWER*

- **Transistors still getting smaller**
 - Moore's Law is alive and well
- **But Denard scaling is dead!**
 - No power efficiency improvements with smaller transistors
 - No clock frequency scaling with smaller transistors
 - All “magical improvement of silicon goodness” has ended
- **Traditional methods for extracting more performance are well-mined**
 - Cannot expect exotic architectures to save us from the “power wall”
 - As daunting as it is, we know more about how to program multicore than we do many of the exotic technologies!
 - Even resources of DARPA can only accelerate existing research prototypes (not “magic” new technology)!



How Small is “Small”



- **Power5 (Server)**
 - 389mm²
 - 120W@1900MHz
- **Intel Core2 sc (laptop)**
 - 130mm²
 - 15W@1000MHz
- **ARM Cortex A8 (automobiles)**
 - 5mm²
 - 0.8W@800MHz
- **Tensilica DP (cell phones / printers)**
 - 0.8mm²
 - 0.09W@600MHz
- **Tensilica Xtensa (Cisco router)**
 - 0.32mm² for 3!
 - 0.05W@600MHz

