Inflammatory title here

John L. Henning
Performance Engineer, Sun Microsystems
Vice-Chair, CPU Subcommittee

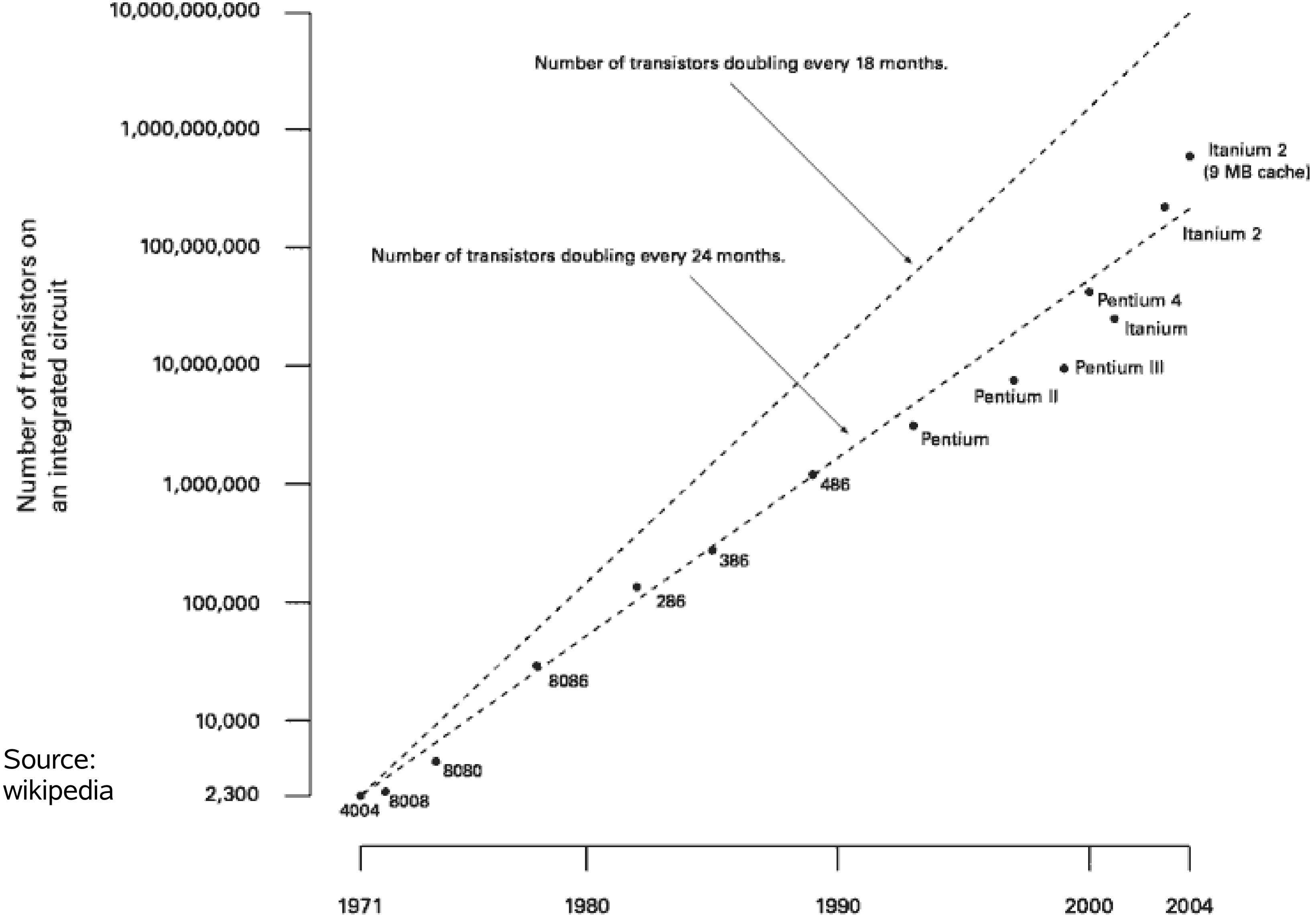
Is CPU2006 the last of SPEC's CPU benchmarks?

John L. Henning
Performance Engineer, Sun Microsystems
Vice-Chair, CPU Subcommittee

Resolved: SPEC CPU2006 is the last of SPEC's CPU benchmarks because:

- The end of Moore's Law is in sight
- CPUs aren't so important anymore
- CPU2006 has enough in it to keep people busy indefinitely
- It's not maintainable
- There's no particular need for a follow-on suite
- The investment in suite development is not sustainable

The end of Moore's Law is in sight

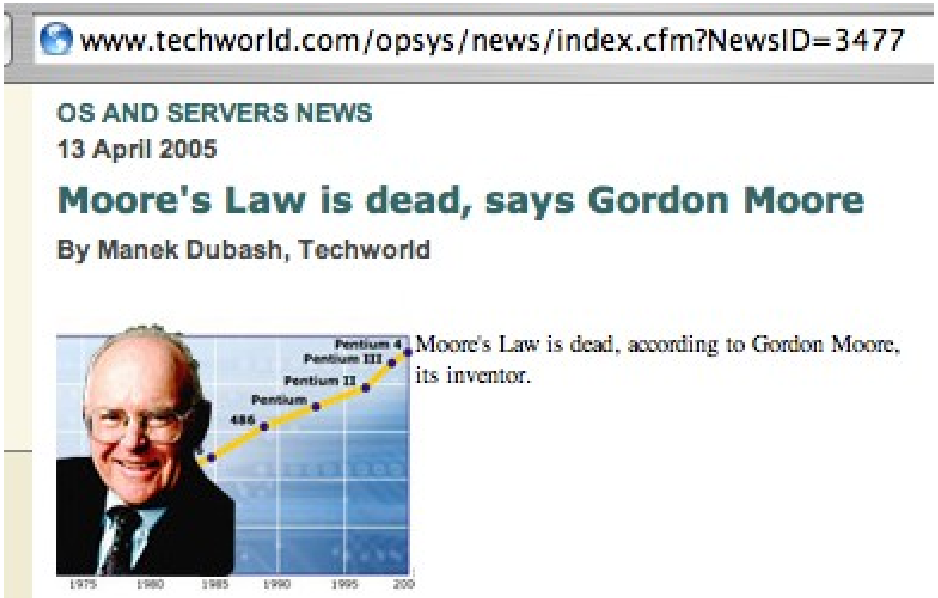


Source: wikipedia

J.Henning Sun Microsystems 21 Jan 2007

At least, according to <u>some</u> certain parties…

- "It can't continue forever. The nature of exponentials is that you push them out and eventually disaster happens."



www.techworld.com/opsys/news/index.cfm?NewsID=3477

OS AND SERVERS NEWS

13 April 2005

Moore's Law is dead, says Gordon Moore

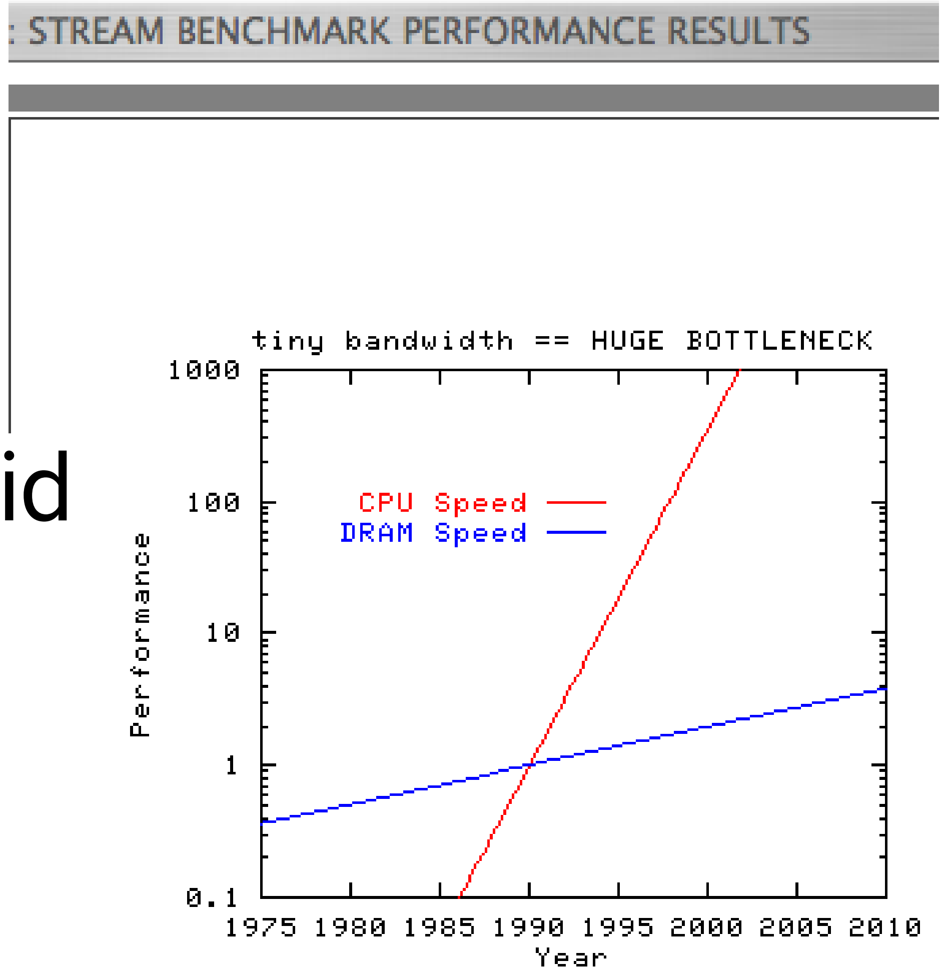
By Manek Dubash, Techworld

Moore's Law is dead, according to Gordon Moore, its inventor.

- Intel's Grove warns of the end of Moore's Law: Feeling the heat http://www.theinquirer.net/default.aspx?article=6677

CPUs aren't so important anymore

It's the memory, stupid


: STREAM BENCHMARK PERFORMANCE RESULTS

- Or the disks,
- or the network,
- or the graphics card,
- or the WoW Authentication Server
- ...

The VAX 11/780 Day Is Long Past

- Once upon a time, a ¼ million dollar system was cheap

- But the VAX 11/780 day was 9571 days ago
  - at least, according to Excel when asked to compute
  `=DATEVALUE("1/21/2007")-DATEVALUE("11/7/80")`

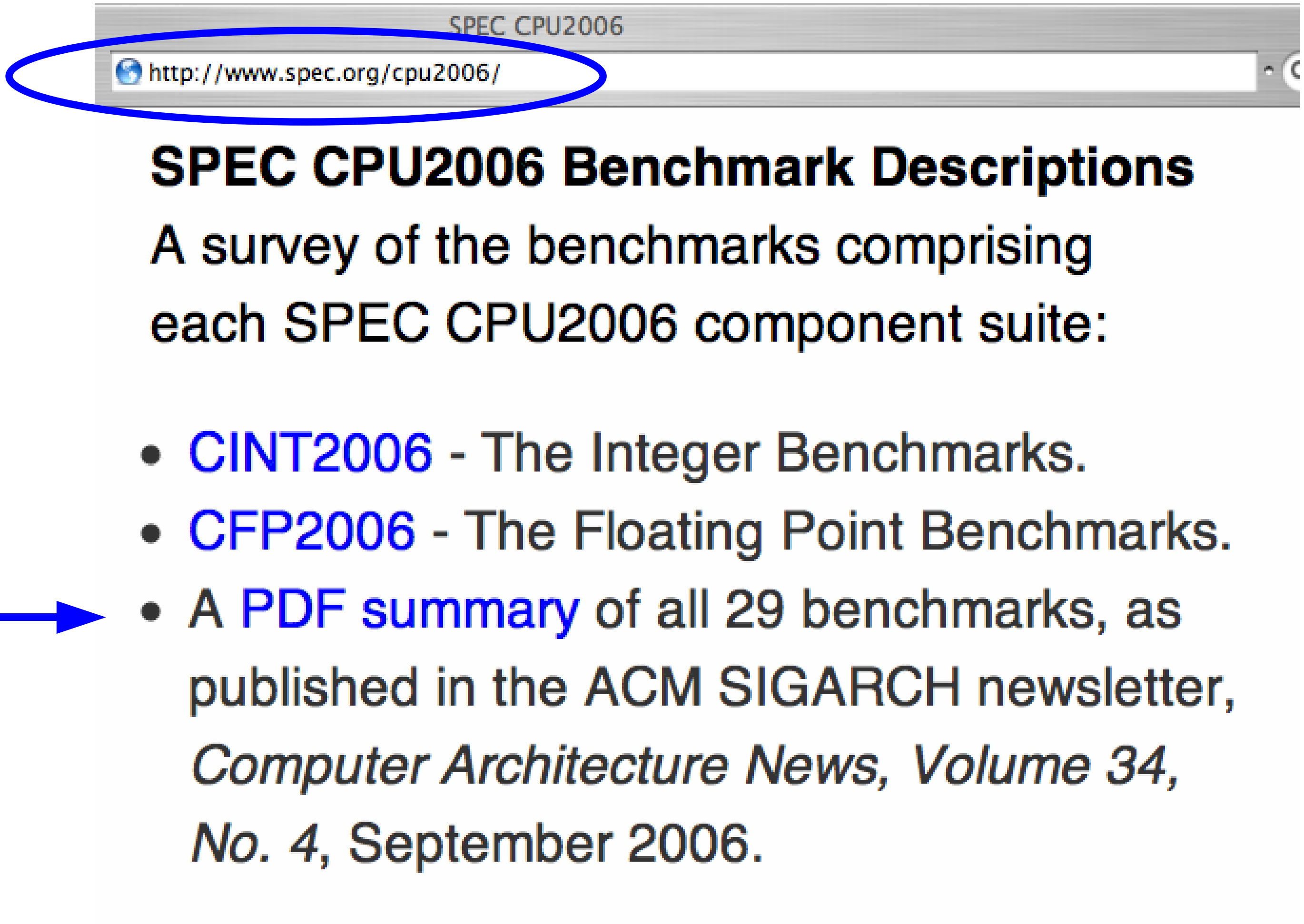
CPU2006 has enough in it to keep people busy indefinitely

CINT2006 (Integer Component of SPEC CPU2006):

Benchmark	Language	Application Area	Brief Description
400.perlbench	C	Programming Language	Derived from Perl V5.8.7. The workload includes SpamAssassin, MHonArc (an email indexer), and specdiff (SPEC's tool that checks benchmark outputs).
401.bzip2	C	Compression	Julian Seward's bzip2 version 1.0.3, modified to do most work in memory, rather than doing I/O.
403.gcc	C	C Compiler	Based on gcc Version 3.2, generates code for Opteron.
429.mcf	C	Combinatorial Optimization	Vehicle scheduling. Uses a network simplex algorithm (which is also used in commercial products) to schedule public transport.
445.gobmk	C	Artificial Intelligence: Go	Plays the game of Go, a simply described but deeply complex game.
456.hmmer	C	Search Gene Sequence	Protein sequence analysis using profile hidden Markov models (profile HMMs)
458.sjeng	C	Artificial Intelligence: chess	A highly-ranked chess program that also plays several chess variants.
462.libquantum	C	Physics / Quantum Computing	Simulates a quantum computer, running Shor's polynomial-time factorization algorithm.
464.h264ref	C	Video Compression	A reference implementation of H.264/AVC, encodes a videostream using 2 parameter sets. The H.264/AVC standard is expected to replace MPEG2
471.omnetpp	C++	Discrete Event Simulation	Uses the OMNet++ discrete event simulator to model a large Ethernet campus network.
473.astar	C++	Path-finding Algorithms	Pathfinding library for 2D maps, including the well known A* algorithm.
483.xalancbmk	C++	XML Processing	A modified version of Xalan-C++, which transforms XML documents to other document types.

CFP2006 (Floating Point Component of SPEC CPU2006):

Benchmark	Language	Application Area	Brief Description
410.bwaves	Fortran	Fluid Dynamics	Computes 3D transonic transient laminar viscous flow.
416.gamess	Fortran	Quantum Chemistry.	Implements a wide range of quantum chemical computations. The SPEC workload does self-consistent field calculations using the Restricted Hartree Fock method, Restricted open-shell Hartree-Fock, and Multi-Configuration Self-Consistent Field
433.milc	C	Physics/Quantum Chromodyn	A gauge field generating program for lattice gauge theory with dynamical quarks.
434.zeusmp	Fortran	Physics / CFD	ZEUS-MP is a computational fluid dynamics code developed at the Laboratory for Computational Astrophysics (NCSA, University of Illinois at Urbana-Champaign) for the simulation of astrophysical phenomena.
435.gromacs	C, Fortran	Biochemistry / Molecular Dynamics	Molecular dynamics, i.e. simulate Newtonian equations of motion for hundreds to millions of particles. The test case simulates protein Lysozyme in a solution.
436.cactusADM	C,Fortran	Physics / General Relativity	Solves the Einstein evolution equations using a staggered-leapfrog numerical method
437.leslie3d	Fortran	Fluid Dynamics	Computational Fluid Dynamics (CFD) using Large-Eddy Simulations with Linear-Eddy Model in 3D. Uses MacCormack Predictor-Corrector time integration
444.namd	C++	Biology / Molecular Dynamics	Simulates biomolecular systems. Test case has 92,224 atoms of apolipoprotein A-I.
447.dealII	C++	Finite Element Analysis	deal.II is a C++ library targeted at adaptive finite elements and error estimation. The testcase solves a Helmholtz-type equation with non-constant coefficients.
450.soplex	C++	Linear Programming, Optimization	Solves a linear program using a simplex algorithm and sparse linear algebra. Test cases include railroad planning and military airlift models.
453.povray	C++	Image Ray-tracing	Image rendering. The testcase is a 1280x1024 anti-aliased image of a landscape with some abstract objects with textures using a Perlin noise function.
454.calculix	C,Fortran	Structural Mechanics	Finite element code for 3D structural applications. Uses the SPOOLES solver library.
459.GemsFDTD	Fortran	Electromagnetics	Solves Maxwell equations in 3D using finite-difference time-domain (FDTD) method.
465.tonto	Fortran	Quantum Chemistry	An open source quantum chemistry package, using an object-oriented design in Fortran 95. The test case places a constraint on a molecular Hartree-Fock wavefunction calculation to better match experimental X-ray diffraction data.
470.lbm	C	Fluid Dynamics	Implements the "Lattice-Boltzmann Method" to simulate incompressible fluids in 3D
481.wrf	C,Fortran	Weather	Weather modeling from scales of meters to thousands of kilometers. The test case is from a 30km area over 2 days.
482.sphinx3	C	Speech recognition	A widely-known speech recognition system from Carnegie Mellon University



SPEC CPU2006 Benchmark Descriptions

A survey of the benchmarks comprising each SPEC CPU2006 component suite:

- CINT2006 - The Integer Benchmarks.
- CFP2006 - The Floating Point Benchmarks.
- A PDF summary of all 29 benchmarks, as published in the ACM SIGARCH newsletter, *Computer Architecture News, Volume 34, No. 4*, September 2006.

Handy PDF summary, airplane-reading-size, 20 copies in the back of the room

Run Rule Changes

- Base does not allow feedback directed optimization (still legal in peak)
- An unlimited number of flags may be set in base,
  - Why? Because flag counting is not worth arguing about. For example, is

    ```
    -fast:np27
    ```

    one flag, two, or three? Prove it. What if it's

    ```
    -fast_np27
    ```
- But they must be set consistently, and rules for consistency have been tightened

Run-Time Dynamic Optimization Is Allowed, Subject to the Usual Rules

- Must be generally available, documented, supported
- HW and SW used by RDO must be disclosed
- You can't tweak the RDO system during the run

RDO v. Benchmarking

Challenge:

- SPEC tests: expected to be repeatable
- But RDO systems learn as they go
- An RDO system might learn too much and "carry over" to the next run of same benchmark
  - Value prediction is too easy when the inputs are always the same
  - Risk: reduce entire benchmark to a print statement
  - If you use RDO, you must have a method to prevent carryover

RDO Is Allowed in Base

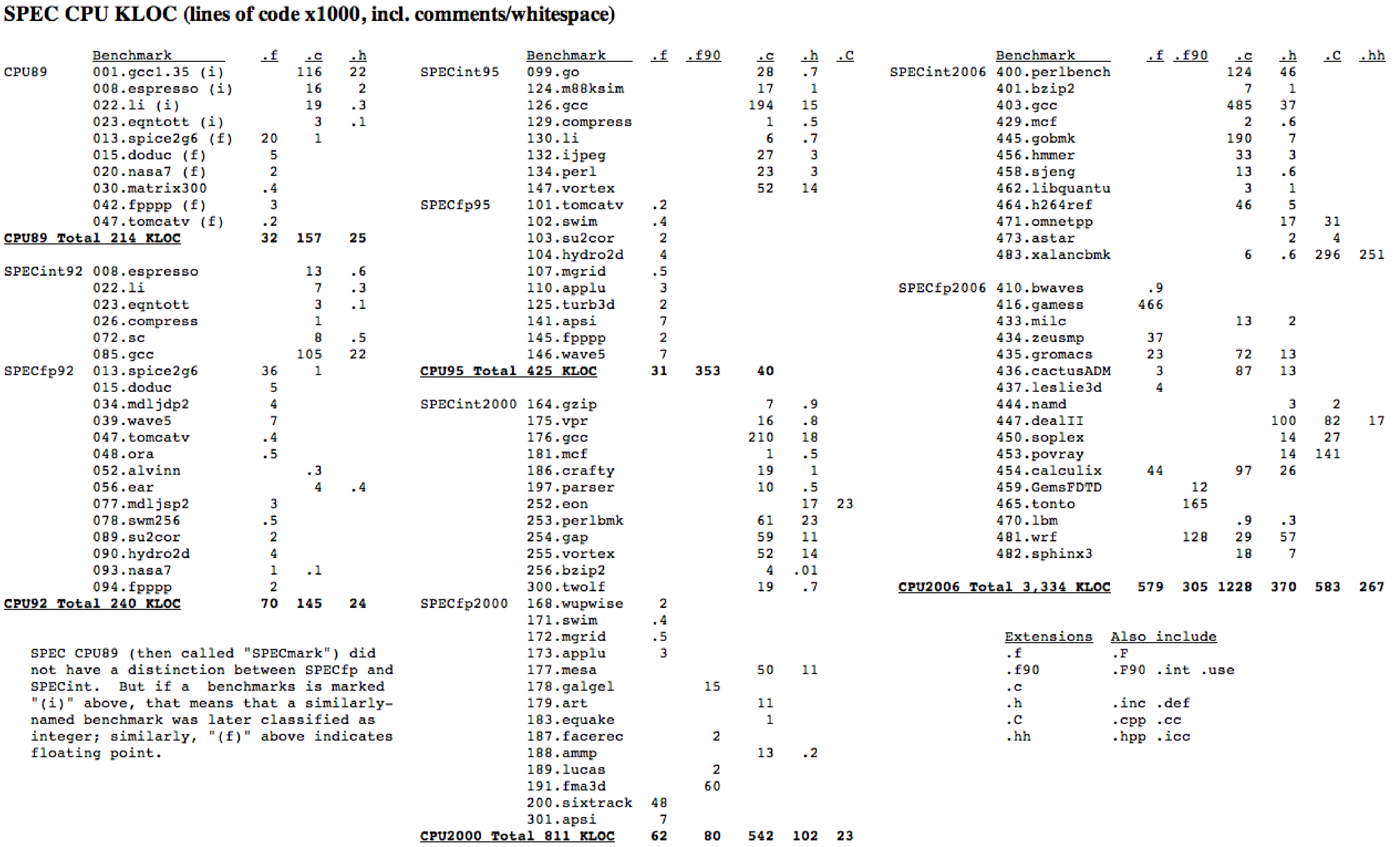
- Must be “safe”
- RDO can assume program meets the Standard
- But can't assume it uses a subset of the standard
- You can't make it work only for the SPEC-supplied inputs

CPU2006 has enough in it to keep people busy indefinitely

Summarizing last few slides:

- Lots of new benchmarks for optimizers to chew on
- Lots to do in base: improve optimizer heuristics in the absence of feedback
- Lots to do for run-time dynamic optimizers

It's Not Maintainable



SPEC CPU KLOC (lines of code x1000, incl. comments/whitespace)

	Benchmark	.f	.c	.h
CPU89	001.gcc1.35 (i)		116	22
	008.espresso (i)		16	2
	022.li (i)		19	.3
	023.eqntott (i)		3	.1
	013.spice2g6 (f)	20	1	
	015.doduc (f)	5		
	020.nasa7 (f)	2		
	030.matrix300	.4		
	042.fpppp (f)	3		
	047.tomcatv (f)	.2		
CPU89 Total 214 KLOC		**32**	**157**	**25**
SPECint92	008.espresso		13	.6
	022.li		7	.3
	023.eqntott		3	.1
	026.compress		1	
	072.sc		8	.5
	085.gcc		105	22
SPECfp92	013.spice2g6	36	1	
	015.doduc	5		
	034.mdljdp2	4		
	039.wave5	7		
	047.tomcatv	.4		
	048.ora	.5		
	052.alvinn		.3	
	056.ear		4	.4
	077.mdljsp2	3		
	078.swm256	.5		
	089.su2cor	2		
	090.hydro2d	4		
	093.nasa7	1	.1	
	094.fpppp	2		
CPU92 Total 240 KLOC		**70**	**145**	**24**

SPEC CPU89 (then called "SPECmark") did not have a distinction between SPECfp and SPECint. But if a benchmarks is marked "(i)" above, that means that a similarly-named benchmark was later classified as integer; similarly, "(f)" above indicates floating point.

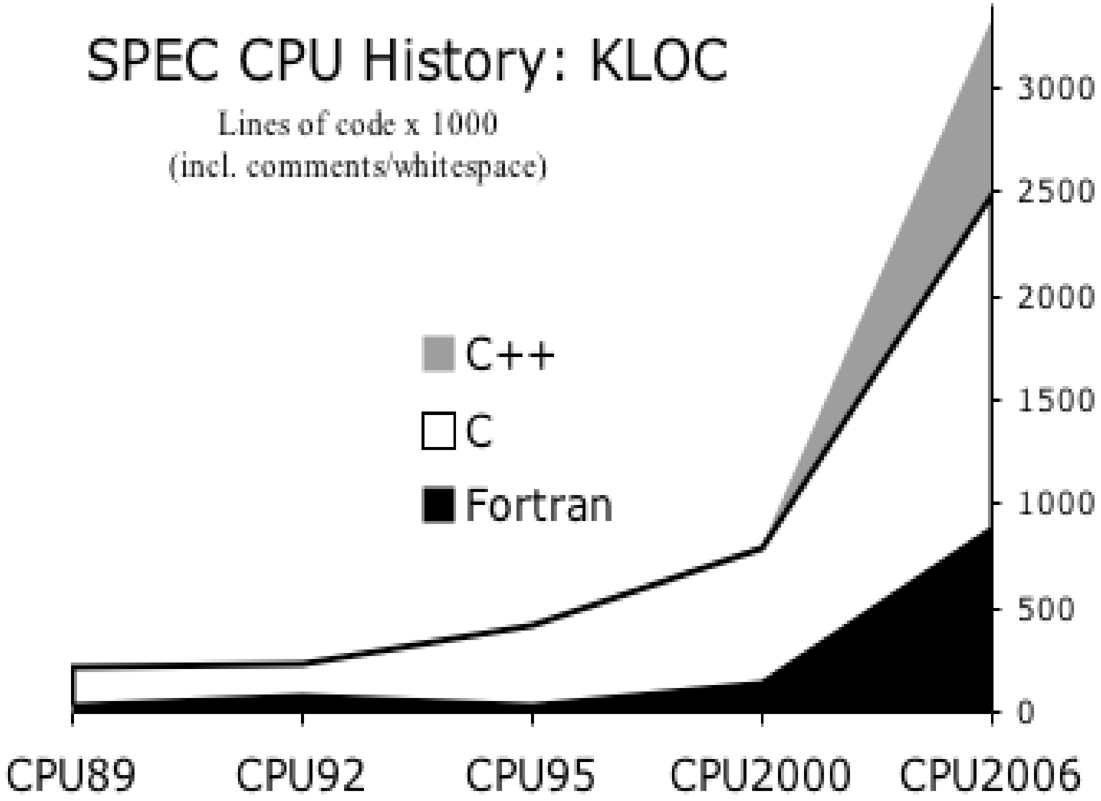
	Benchmark	.f	.f90	.c	.h	.C
SPECint95	099.go			28	.7	
	124.m88ksim			17	1	
	126.gcc			194	15	
	129.compress			1	.5	
	130.li			6	.7	
	132.ijpeg			27	3	
	134.perl			23	3	
	147.vortex			52	14	
SPECfp95	101.tomcatv	.2				
	102.swim	.4				
	103.su2cor	2				
	104.hydro2d	4				
	107.mgrid	.5				
	110.applu	3				
	125.turb3d	2				
	141.apsi	7				
	145.fpppp	2				
	146.wave5	7				
CPU95 Total 425 KLOC		**31**	**353**	**40**		
SPECint2000	164.gzip			7	.9	
	175.vpr			16	.8	
	176.gcc			210	18	
	181.mcf			1	.5	
	186.crafty			19	1	
	197.parser			10	.5	
	252.eon				17	23
	253.perlbmk			61	23	
	254.gap			59	11	
	255.vortex			52	14	
	256.bzip2			4	.01	
	300.twolf			19	.7	
SPECfp2000	168.wupwise	2				
	171.swim	.4				
	172.mgrid	.5				
	173.applu	3				
	177.mesa			50	11	
	178.galgel		15			
	179.art			11		
	183.equake			1		
	187.facerec		2			
	188.ammp			13	.2	
	189.lucas		2			
	191.fma3d		60			
	200.sixtrack	48				
	301.apsi	7				
CPU2000 Total 811 KLOC		**62**	**80**	**542**	**102**	**23**

	Benchmark	.f	.f90	.c	.h	.C	.hh
SPECint2006	400.perlbench			124	46		
	401.bzip2			7	1		
	403.gcc			485	37		
	429.mcf			2	.6		
	445.gobmk			190	7		
	456.hmmer			33	3		
	458.sjeng			13	.6		
	462.libquantu			3	1		
	464.h264ref			46	5		
	471.omnetpp				17	31	
	473.astar				2	4	
	483.xalancbmk			6	.6	296	251
SPECfp2006	410.bwaves	.9					
	416.gamess	466					
	433.milc			13	2		
	434.zeusmp	37					
	435.gromacs	23		72	13		
	436.cactusADM	3		87	13		
	437.leslie3d	4					
	444.namd				3	2	
	447.dealII				100	82	17
	450.soplex				14	27	
	453.povray				14	141	
	454.calculix	44		97	26		
	459.GemsFDTD		12				
	465.tonto		165				
	470.lbm			.9	.3		
	481.wrf		128	29	57		
	482.sphinx3			18	7		
CPU2006 Total 3,334 KLOC		**579**	**305**	**1228**	**370**	**583**	**267**

Extensions	Also include
.f	.F
.f90	.F90 .int .use
.c	
.h	.inc .def
.C	.cpp .cc
.hh	.hpp .icc

It's Not Maintainable

- Any code set this size has bugs
- No one person can understand all of it
- SPEC, as a mostly volunteer organization, is not prepared to handle exploding support calls
- Certainly not if growth continues on this curve



There's no particular need for a follow-on suite

One committee member has repeatedly claimed that CPU2006 is so much better than CPU2000 that there just won't be a need to update it.

There's no particular need for a follow-on suite

One committee member has repeatedly claimed that CPU2006 is so much better than CPU2000 that there just won't be a need to update it.

"Everything that can be invented
has been invented."

There's no particular need for a follow-on suite

One committee member has repeatedly claimed that CPU2006 is so much better than CPU2000 that there just won't be a need to update it.

"Everything that can be invented has been invented."

- U.Legend

http://tinyurl.com/2vkqlc aka

http://www.findarticles.com/p/articles/mi_m2843/is_3_27/ai_100755224/print

The investment in suite development is not sustainable

- www.spec.org/cpu2006/docs/credits.html names 116 people (well, 82 `uniq` people) who worked on CPU2006
- SPEC is largely a volunteer organization
- In these days of tightening budgets and greater accountability, people won't get away with investing so much time in the future

Just a moment...

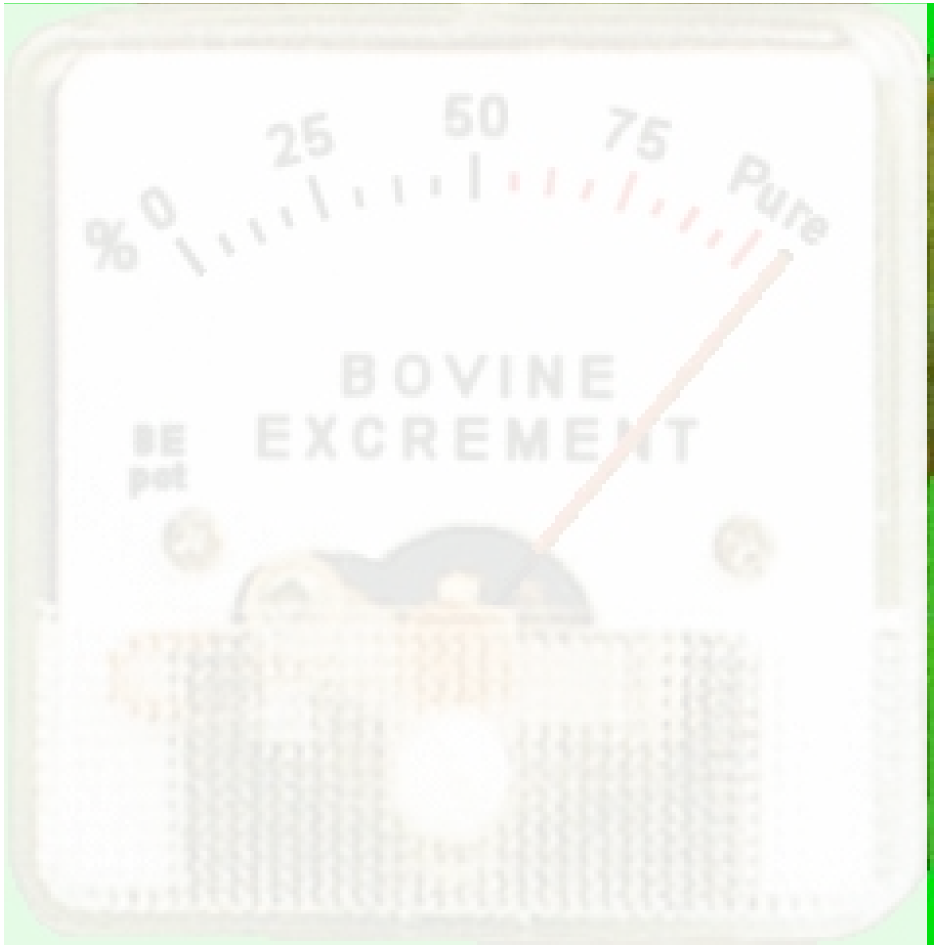
Just a moment...

Just a moment

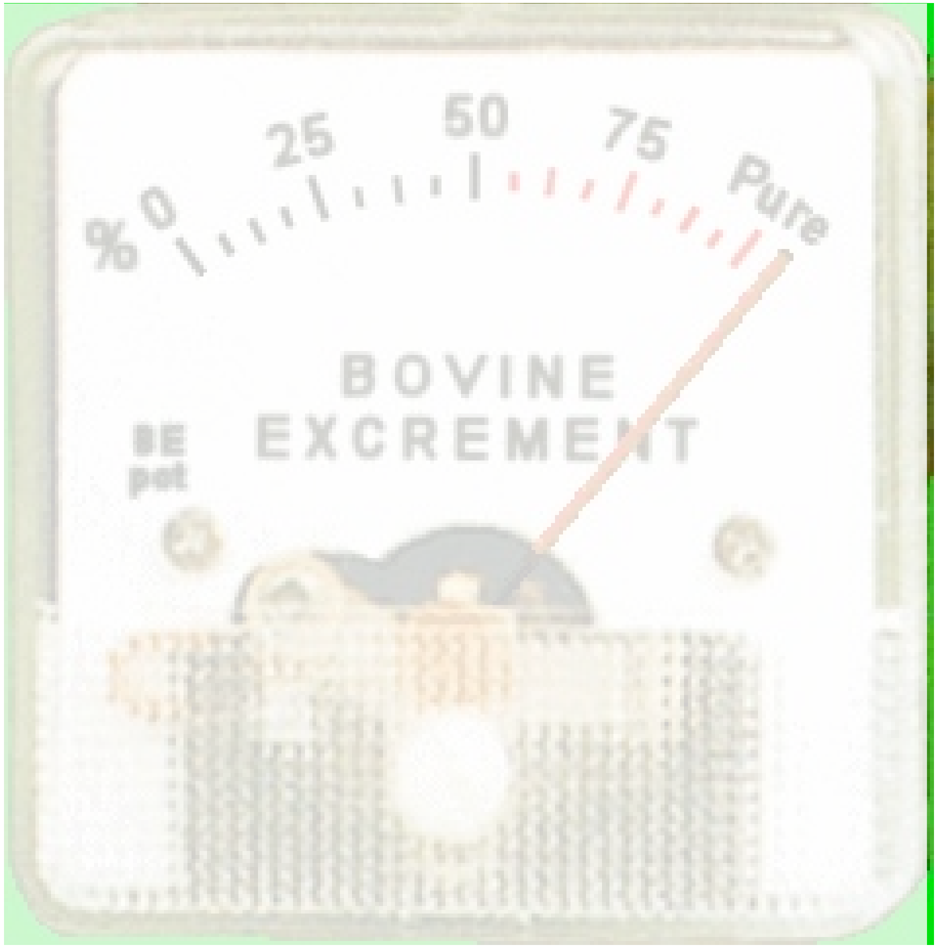
Just a moment...

We appear to be
receiving some measurements
regarding the talk
up to this point

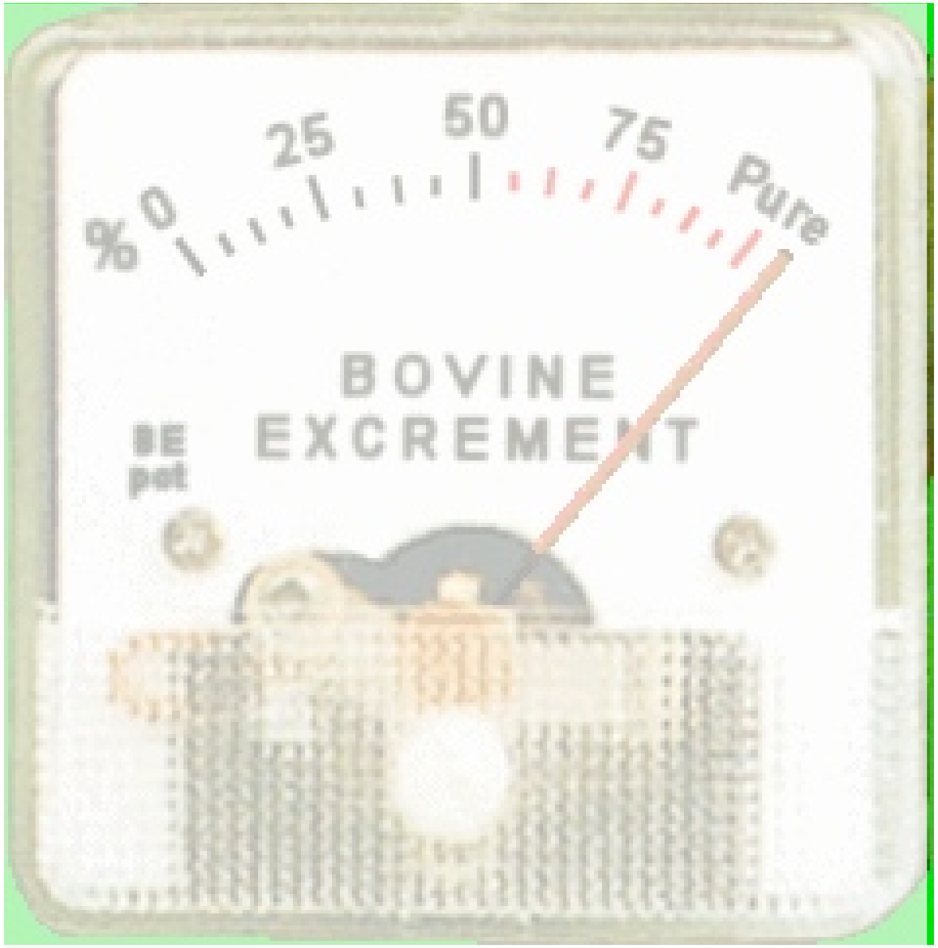
Just a moment...



Just a moment...



Just a moment...

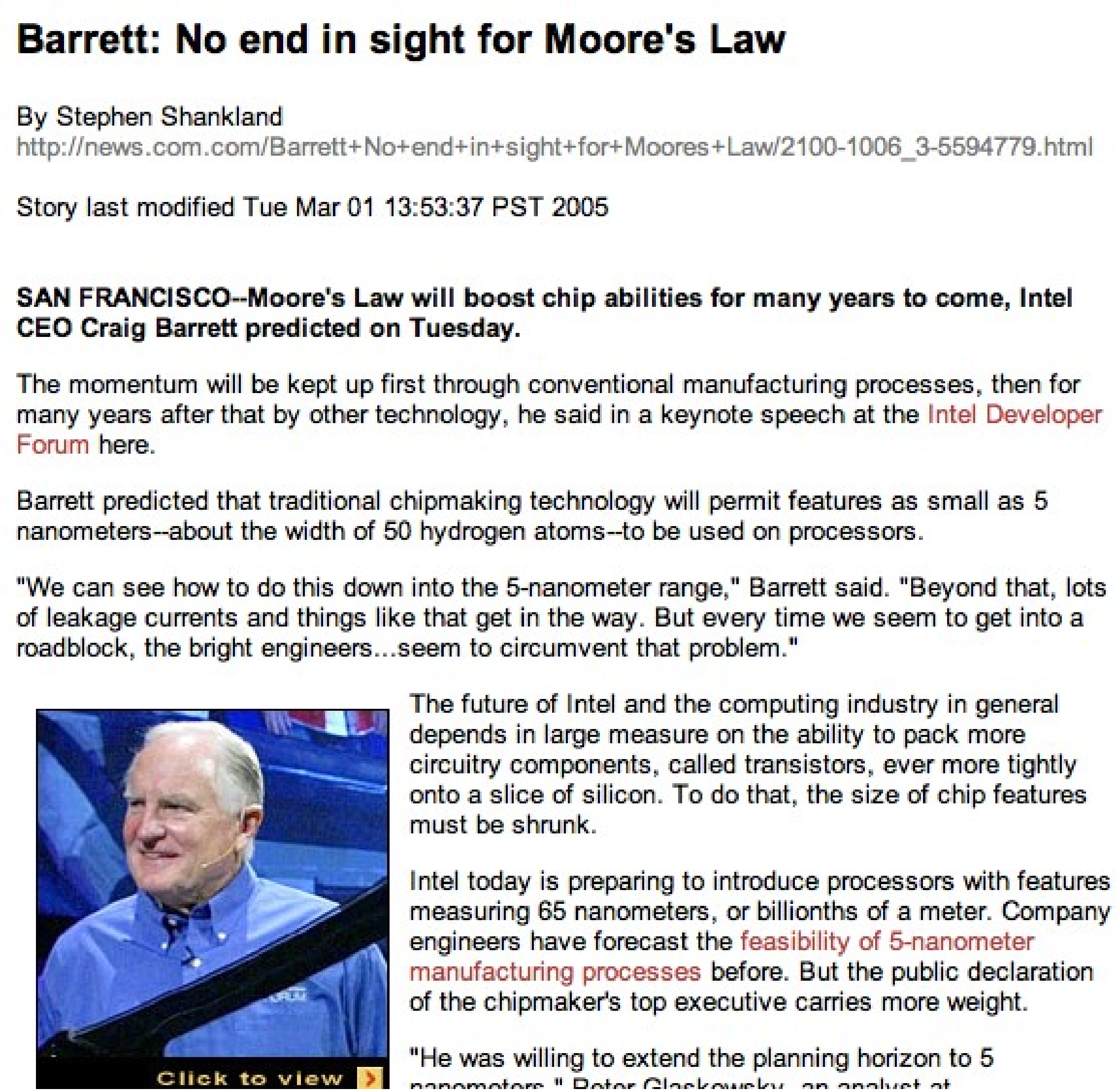


Just a moment...

Just a moment...

Perhaps there's another side to this story

~~The end of Moore's Law is in sight~~



Barrett: No end in sight for Moore's Law

By Stephen Shankland
http://news.com.com/Barrett+No+end+in+sight+for+Moores+Law/2100-1006_3-5594779.html

Story last modified Tue Mar 01 13:53:37 PST 2005

SAN FRANCISCO--Moore's Law will boost chip abilities for many years to come, Intel CEO Craig Barrett predicted on Tuesday.

The momentum will be kept up first through conventional manufacturing processes, then for many years after that by other technology, he said in a keynote speech at the Intel Developer Forum here.

Barrett predicted that traditional chipmaking technology will permit features as small as 5 nanometers--about the width of 50 hydrogen atoms--to be used on processors.

"We can see how to do this down into the 5-nanometer range," Barrett said. "Beyond that, lots of leakage currents and things like that get in the way. But every time we seem to get into a roadblock, the bright engineers...seem to circumvent that problem."

The future of Intel and the computing industry in general depends in large measure on the ability to pack more circuitry components, called transistors, ever more tightly onto a slice of silicon. To do that, the size of chip features must be shrunk.

Intel today is preparing to introduce processors with features measuring 65 nanometers, or billionths of a meter. Company engineers have forecast the feasibility of 5-nanometer manufacturing processes before. But the public declaration of the chipmaker's top executive carries more weight.

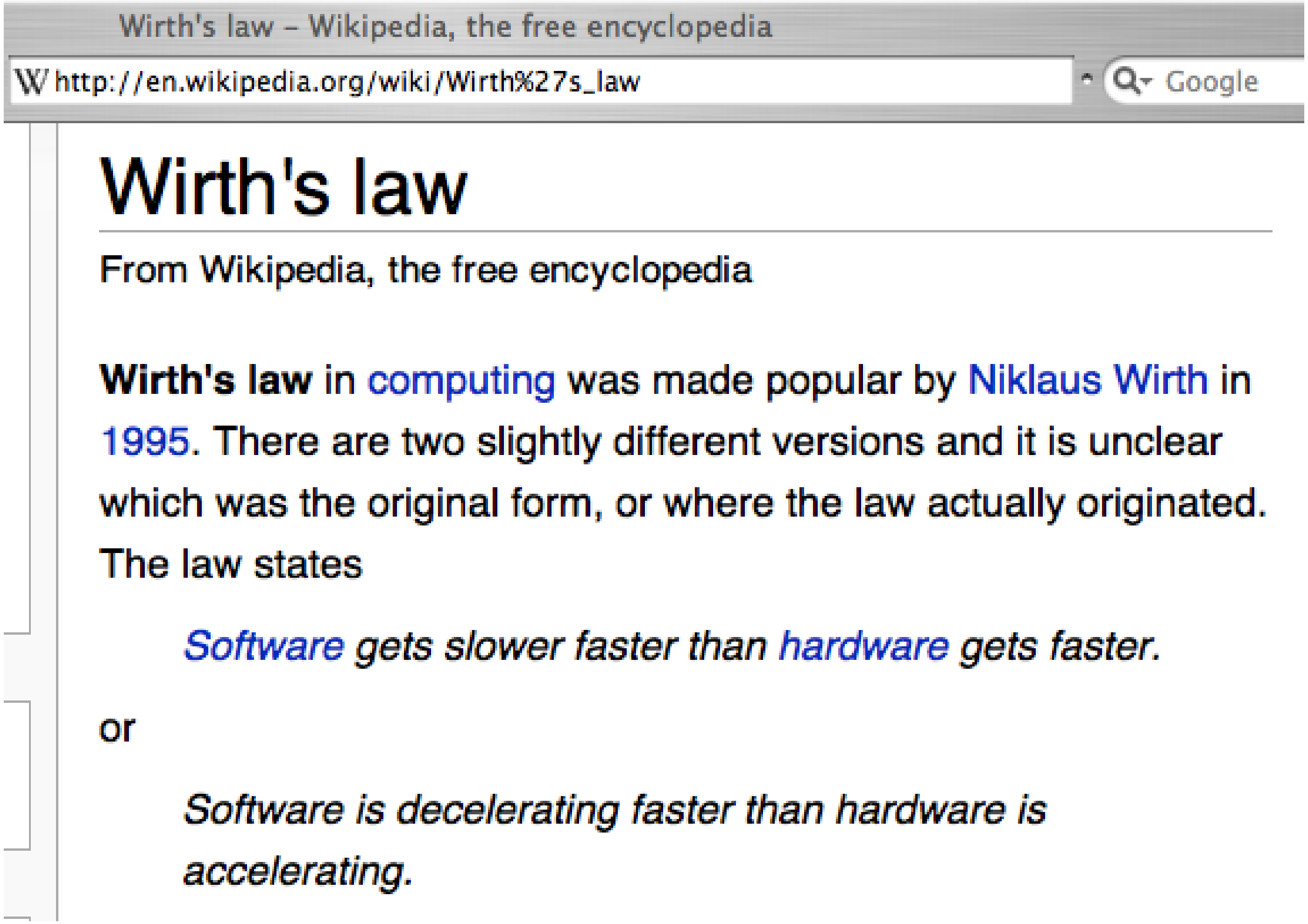
"He was willing to extend the planning horizon to 5 nanometers," Peter Glaskowsky, an analyst at

"Predictions are dangerous, especially when they're about the future"
- Y. Berra

"Moore's Law is a violation of Murphy's Law. Everything gets better and better."
- G. Moore, 2005,
in The Economist,
quoted @ wikipedia

~~CPUs aren't so important anymore~~

- SPEC CPU remains ½ of the traffic to www.spec.org
- Wirth's law:



Wirth's law - Wikipedia, the free encyclopedia

http://en.wikipedia.org/wiki/Wirth%27s_law

Wirth's law

From Wikipedia, the free encyclopedia

Wirth's law in computing was made popular by Niklaus Wirth in 1995. There are two slightly different versions and it is unclear which was the original form, or where the law actually originated. The law states

> *Software gets slower faster than hardware gets faster.*

or

> *Software is decelerating faster than hardware is accelerating.*

~~CPU2006 has enough in it to keep people busy indefinitely~~

- That may depend on who you're trying to keep busy
- (story: Milford diner at 5am)

SPEC CPU Development Contributes to Optimizer Quality

- ISVs and users may not be strongly motivated to report optimizer bugs.

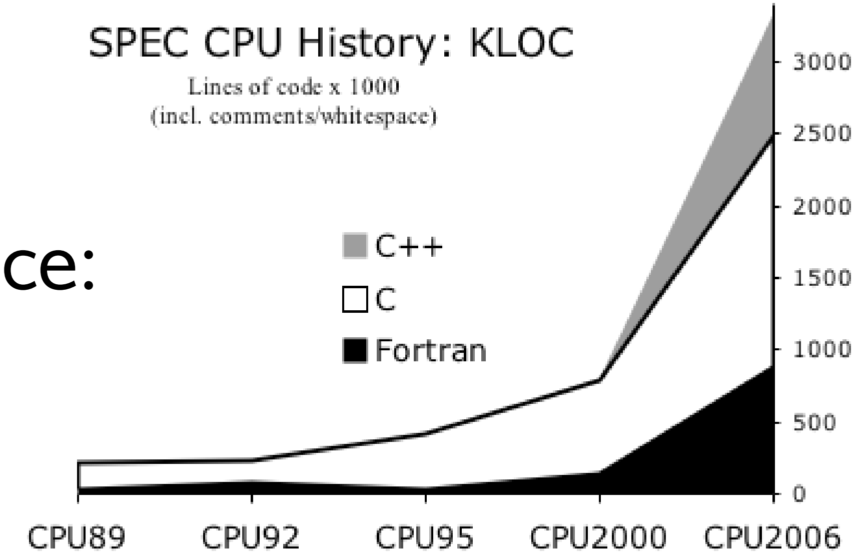
"Oh, it fails with -O5?
What happens if you turn it down to -O3?"

- May not feel a need to find root cause: program bug, standards violation, or actual compiler bug.
- SPEC CPU benchmark development provides a continuing stream of test cases to chew on, with strong motivation to find root causes

Porter's Progress

- SPEC maintains a list of tested platforms and benchmark candidate status
- On 28 Feb 2005: 52 benchmarks x 33 platforms
- 135 unsuccessful tests tentatively assigned to the platform under test, not to the benchmark candidate.

~~It's not maintainable~~



Yes, there *is* a risk

There is also reason for confidence:

- Extensive pre-release testing
- Much of the growth has come from well-exercised open source apps
- The rate of changes due to *benchmark* problems has been low since March 2006.
  - 6 changes for portability to additional compilers
  - 1 actual problem (which did not prevent running the benchmark)

~~There's no particular need for a follow-on suite~~

- 64-bit
- Other languages?
- Cache growth
- Low-hanging fruit
- The “lag”
  - Hard to get very best ISV algorithms into current suite
  - Not so much of a problem with OSS

The investment in suite development is not sustainable

?

The investment in suite development is not sustainable

The investment in suite development is not sustainable

- www.spec.org/cpu2006/docs/credits.html names 116 people (well, 82 `uniq` people) who worked on CPU2006
- SPEC is largely a volunteer organization
- In these days of tightening budgets and greater accountability, people won't get away with investing so much time in the future

The investment in suite development is not sustainable

- www.spec.org/cpu2006/docs/credits.html names 116 people (well, 82 `uniq` people) who worked on CPU2006

- **SPEC is largely a volunteer organization**

- In these days of tightening budgets and greater accountability, people won't get away with investing so much time in the future

SPEC is largely a volunteer organization

- Yes, we all have employers with divergent interests
- But do you have the courage to say: "*Fair Benchmarks Are Important*"
  - To customers
  - To the industry
  - To my own institution
  - Distinction: long-term v. short term interests