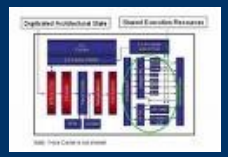
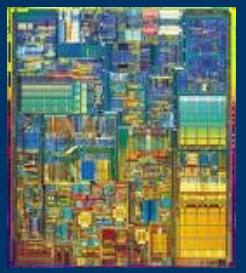
Adam Fitzgerald and Nick Black present...
...an Eric Sprangle and Doug Carmean joint...

Increasing Processor Performance by Implementing Deeper Pipelines [2002]

A DISQUISITION INTO the Theories, Conjectures and Magicks of Mssrs. Carmean and Sprangle, both FELLOWS Most LAUDED of Intel (an Auric Bear State ENCHARTERMENT), whereby claims will be Judged, heresies will be Illuminated, rebuttals Made, and information Sprayed, yea, as if it was going out of Style.





Presented for 8803MCA 2009-01-30

“They told me Satan would be attractive.”
-Ned Flanders (presumably invoking the Tiburtine Sibyl Hoax)

- “100% frequency increase”
- “35%-90% improvement over P4”
- Sprangle and Carmean sing of a 52-stage pipeline, clocked “in the far infrared” (as they'd say, grinning, back in 2002), and a future where hip kids in Austin and Willamette talk about MOEs (Moles of Ops Executed (as in 6.02*10^23)).
- This limit, and the paper's title, center on the lemma that branch misprediction delay (BMD) bounds the overall performance curve. See Figure 12 (this paper's Money Figure for sure), where performance graphs of various L2 cache sizes all see a critical point at a 52-cycle BMD.
- Analysis simulates the the P4 (which Sprangle and Carmean'd helped design). Here's some P4 basics:
  - The first processor built around NetBurst architecture (the PIII had used P6).
  - All P4's have used NetBurst, which was designed to scale to 10GHz.
  - Willamette (180nm), Northwood (130nm, more cache), 20 stages
  - Prescott (90nm), Cedar Mill (65nm), 31 stages. 130W TDP on Presler P-D.
  - “Son of [Sprangle 2002].” Echoes of the Go-Go Nineties. Extreme Editions!

But, meanwhile in Santa Clara, not all is well...

- Let's run pipelines on 90nm at 10GHz and make that paper, son!
- Switching (dynamic) power: C*V^2*ν.
  - Doubling frequency doubles power (you can't win).
  - Doubling state transitions doubles power (you can't break even).
  - Cutting voltage hits seemingly fundamental limits. (you can't leave the game).
    - Yes, there's dynamic management of all this now, but sometimes you need run, boy (please read Observations on Power Management)
  - Power consumption is: bad. Oh, and power consumption leads to heat.
  - Heat is also: bad (the Frogurt is also cursed). Removing it requires...power!
- Amplifies pipeline overhead as portion of cycle. Must reduce:
  - Usable time (combinatorial logic improvements, Vitamin V, stage-splitting)
  - Jitter (process/materials improvements, time-borrowing, 50-75ps)
  - Latch delay (3x 25ps F04's in 180nm, direct dominoes, pulse clocks, 0-50ps)
  - Skew (smaller (less powerful) processors, janky local clocks)

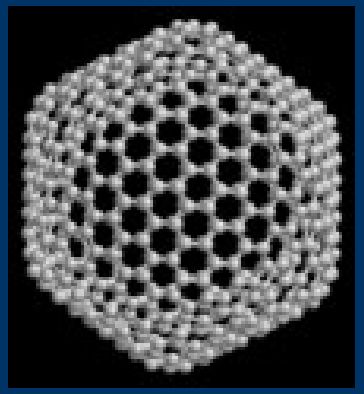
Amplifies the effects of some architectural delays: branch misprediction, long-latency instructions, memory accesses. Here again there are tradeoffs.

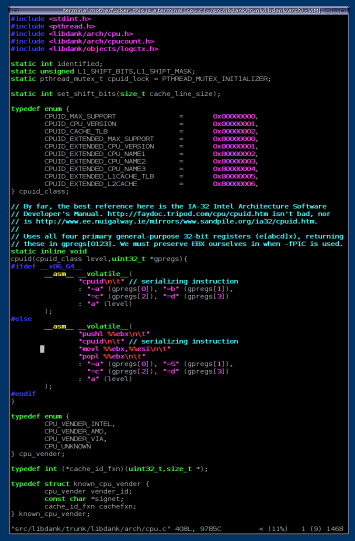
“Why this is hell, nor am I out of it./ Think’st thou that I, who saw the face of God,/And tasted the eternal joys of heaven,/ Am not tormented with ten thousand hells/ In being deprived of everlasting bliss.” - **Mephistopheles**, *Dr. Faustus (Marlowe, 1604)*

Left: ***Satan****,* from Gustave Doré's illustrations for *Paradise Lost (1887)*

The Assumptions

- Usable worktime/cycle can be decreased towards zero, using ever more complex systems of time-borrowing, approaching a Ptolemaic epicyclical scheme (Where the Cray-1 was a loveseat, 10GHz P4's would have required geodesic domes, fullerine-based forms, hypercubes or leasing some Japanese neutrino detectors).
- The “Skeleton” simulator's implementation is error-free, and faithfully implements the processor and not a preconception of the processor. Output is used to vet assumptions and hypotheses both.
  - Rant here – do these things implement CPUID? Why is it thought that changing one parameter in a simulator is acceptable, when a well-planned design embraces all factors?
- The LIT's (Long Instruction Traces) are meaningful, and simulating them generates all major result patterns.
  - NOT “covers all app types” -- insert rant!
- “Making cache larger” will preclude most accesses of main store (a prejudice of much of the SPEC benchmarks).
- No mention of increased memory bandwidths, Amdahl's Balanced System Law etc (1MIPS per 1MBps). Static architecture (p27).
- Compiler doesn't change.
  - Rant rant rant! This is moronic! Who doesn't use -march and -mtune arguments to gcc or /Q to icc (see CPUID above for runtime library-based methods in conjunction with this kind of thing). That's like compiling without -O; it just ain't done, boyo!





The Claims and Conclusions

- Lower bound on cycle time == Upper bound on frequency == Upper bound on pipeline depth: minimum cycle time results from constant pipeline overhead for a given circuit technology (p27)
  - 2GHz P4: 500ps cycle, 90ps overhead yields 11.1GHz
- Branch misprediction detection will continue to occur late in the pipeline, even as stage count increases (p27). (Why?)
  - Same P4: 20-stage pipeline yields BMD of 8200 useful ps
- ALU cycles are strongest of all (but won't further pipelining require slicing of the ALU stages? Unaddressed!)
- Unique relation of BMD and performance: simple, strong critical point (first derivative at zero) system and thus global optimum. See Figures 5, 7, 8 and 12 (and my fly gimp/poppler skills) below.
- 52stage optimum ~= 100% freq gain. BIG MONEY! BIG PRIZES!

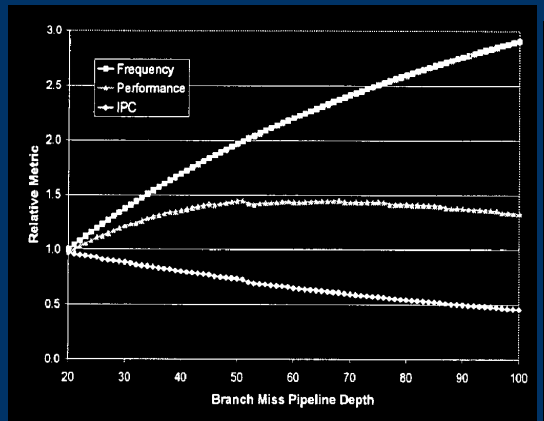


Figure 5: Frequency/IPC/Performance vs. branch misprediction pipeline depth.

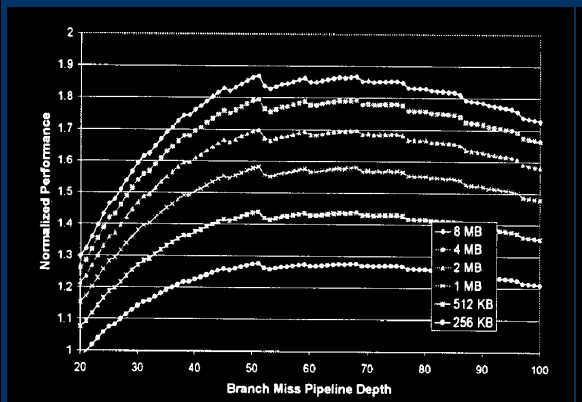


Figure 12: Performance vs. pipeline depth for different L2 cache sizes.

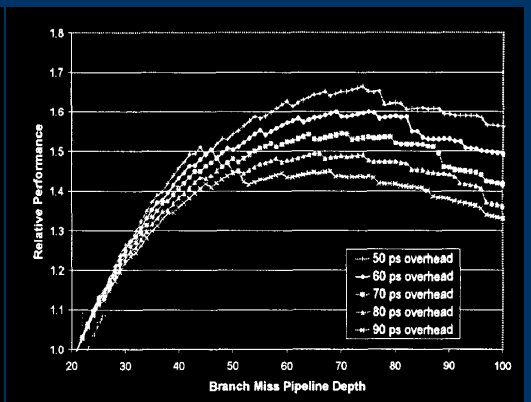


Figure 8: Performance vs. pipeline depth vs. pipeline overhead

30 Californians Agree

- P&H 4th Edition: Pipeline delays due to branching are a major cause of processor underutilization
- Note the epic contribution to SPECJBB especially!
- Some branches can't be predicted any better than a guess (see the adversarial model of cryptographic evaluation).
- Is there a SPEC benchmark for "if(bit from /dev/random % 2)"?
  - A suitably large predictor **could** predict a linear congruential PRNG! =]
- What about algorithmic complexity attacks at the architectural level?
- What about information leakage at the architectural level?
  - Go read Bernstein's paper there – it's phat like ham bone, yet tight like gnat booty.

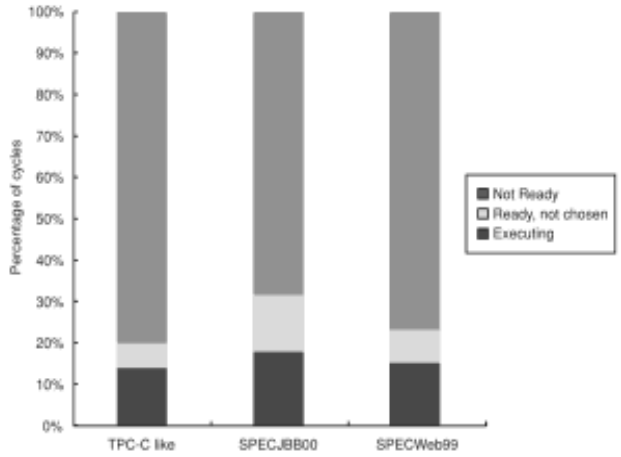


Figure 4.30 Breakdown of the status on an average thread. Executing indicates the thread issues an instruction in that cycle. Ready but not chosen means it could issue, but another thread has been chosen, and *not ready* indicates that the thread is awaiting the completion of an event (a pipeline delay or cache miss, for example).

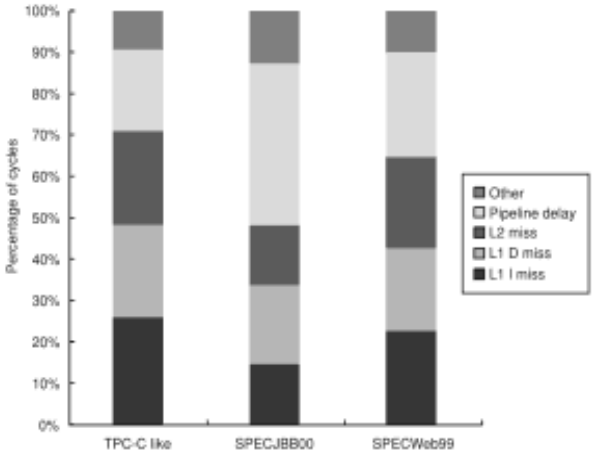


Figure 4.31 The breakdown of causes for a thread being not ready. The contribution to the "other" category varies. In TPC-C, store buffer full is the largest contributor; in SPEC-JBB, atomic instructions are the largest contributor; and in SPECWeb99, both factors contribute.

Pages 28-29: Totally phoned in

- After assaulting their Vigilant Readers with that 10GHz figure and all these 100%'s and doublings and the first of what'll be several ugly graphs all conveying the same single data point, the Mssrs. figure they'll need time to recover. The two specious and supernumerary pages following are the “load delay slot”; there were jokes about “the R in R3000 standing for Reader.”
  - Uncomfortable silence. Does anyone know about latter-day -mips1 flags and LDS?
  - It is nice to see a shout-out to the 21264.regarding RAT pipelining, but...
  - No mention of 21264/EV8's “prophet-critic “hybrid predictor??!? The single best-named thing in computer architecture (“barrel shifter” is right up there), and we can't mention it in the Conclusion in a paper about branch prediction's importance? Maybe if it was called “gprophet-critic”, or “Seattle-Tacoma Hyperpredictor”, or “fused looker-upper-predictor”
- What on earth does it mean to “pipeline a wire” (section 9.2)?
- Oh great, they have a citation! Here we go, footnote [9]...
- “[9] Personal communication with David Sager, Pentium Processor Architecture Group, Intel.”
- ....
- We can make stuff up, too: “Well let me have Rosie pull Old Bull Sager out of the Rolodex; we'll get together with some beer and some mescal and have the boys beat hell outta one another, like back in the lush-rolling days. Ahoy-hoy, Sager, you rakish Pentium Architect! Proof by reference to inaccessible, unpublished water-cooler talk eh old boy? Good show, good show.”

Extremely large caches have poor track records

- Sprangle and Carmean assume caches will grow in effectiveness alongside processors – that the hit rate won't drop.
- Figures 10 and 11 measure only misses per instructions, not performance. Larger caches often have more latency.
- The P4 Extreme Edition, with a heapin' helping of L3 cache, performed better on some benchmarks with the L3 turned off due to added latencies.
- Only SRAM is clocked to the processor (SDRAM runs on its own clock); main memory's access time in cycles will increase at a rate exactly offsetting the rate of increase in frequency (for cache misses).
- If main store is **not** bounding performance, and thus the frequency increase can improve performance at all, memory accesses per unit time can at best hold even and will likely increase – more memory bandwidth, and still more underutilization (due to unrealized potentialities of greater work).

More Hz means more hardware everywhere

- Need a larger reorder buffer, lest we stall on retire
- Need more write buffers, lest we stall on write
- Need bigger caches and support, lest we stall on core
- Need exponentially(?) more interconnect, lest we stall on updates.
- Bigger buses need more pins, with heavy real estate requirements.
- All of it running at saganv(“billions and billions”) Hz
- Don't forget about that x86 compatibility code! I want to run BCD-intensive code compiled for a 20Hz 386 and maybe, maybe I want to do it in real mode! What the hell happens when one JMP's to FFFF:0000 these days? Is Int 19h being serviced with traditional verve? Millions of transistors are working to make the answer: undefined, but undefined the same way every time.
- 500 million years of catagenesis for this?