# ECE 571 – Advanced Microprocessor-Based Design Lecture 18

Vince Weaver

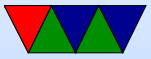
http://www.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

28 March 2013

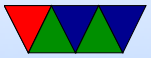
# Project/HW Reminder

- Homework will be posted soon.

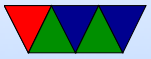


# Estimating Power – Why?

- Have no power measurement tools available
- Don't want to void warranty by splicing wires into your power supply
- Have a large cluster / supercomputer and getting meters for all is cost prohibitive
- Planning a new processor design and want to estimate power in advance

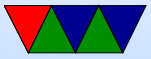


- On CMP hard to differentiate per-core (rather than per-package) as voltage planes shared
- Estimate the power consumption of different functional units
- Hard to measure energy, spiky, DMM tend to smooth/average (Chang, Kim, Lee 2000)
- Power-aware scheduling



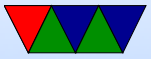
# Simulators

- At the transistor or verilog level – slow
- How can you speed this up?

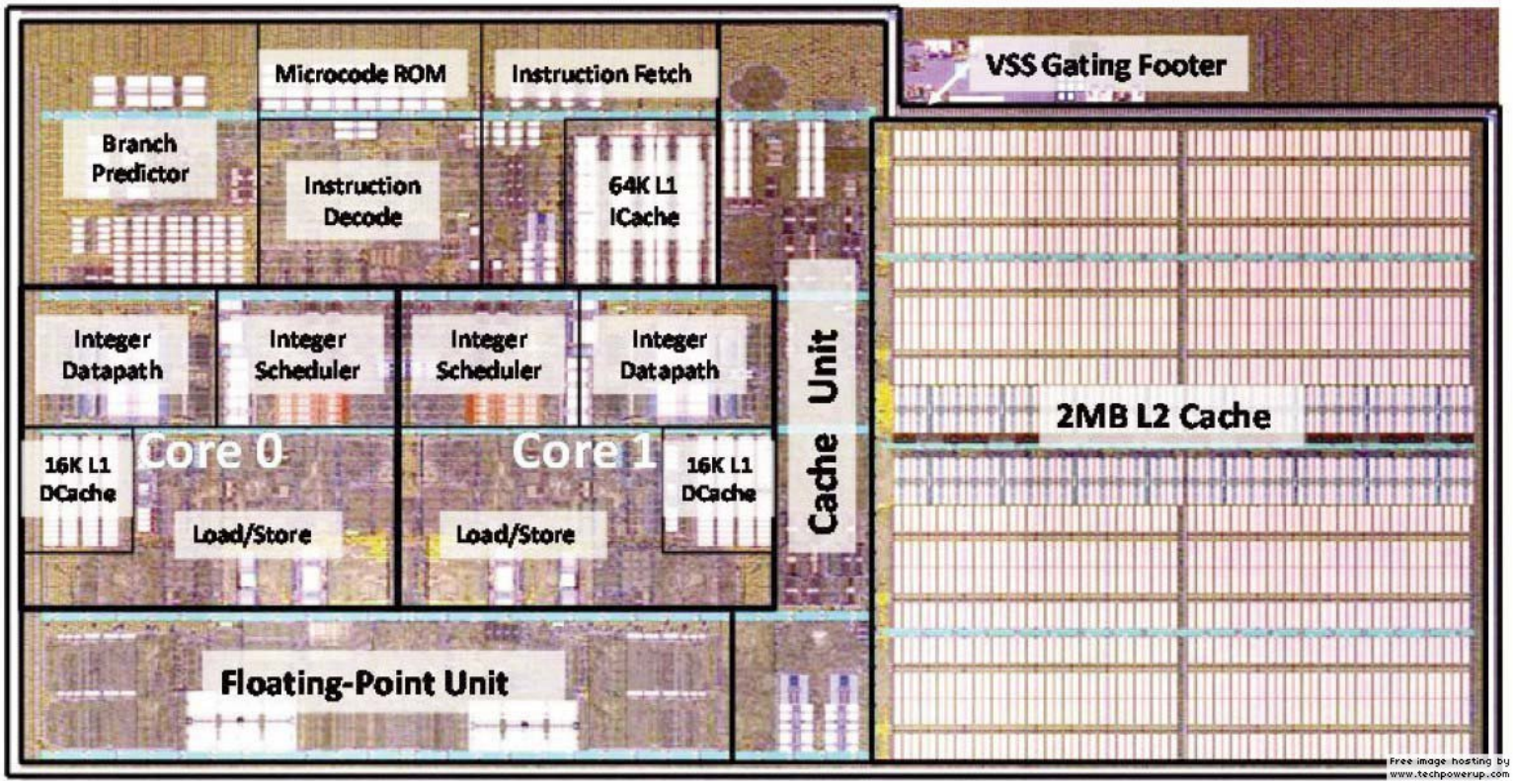


# Simulators

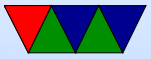
- How would you write an energy simulator?



# AMD Bulldozer Die Shot Again

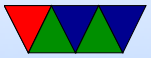



yes, people do make estimates based on these



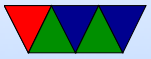
# Methodology

- Use CMOS dynamic power equation
- Estimate capacitance
- You know Vdd and frequency
- Calculate/Estimate activity factor

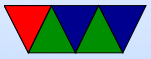


# Wattch

- (Brooks, Tiwari, Martonosi. ISCA 2000)
- 1000X faster than layout based tools with 10% accuracy
- Parameterized power modules of the units.
- Arrays (caches), CAMs (fully assoc), Combinational Logic, Clocking
- Estimate Capacitance for all, use CMOS dynamic mem model

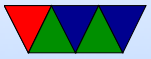


- Simple scalar provides stats for usage of units.
- 30% overhead over just performance simulation
- Powermill on 64-bit adder with 100 vectors, same time wattch 280M insn.
- Dynamic Only



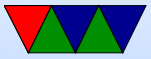
# Sim-PowerCMP

- (Flores, Aragón, Acacio. SEC 2007)
- Merges various models together
- RSIM (execution), Wattch (dynamic), Hotleakage (Leakage), Orion (Interconnect)



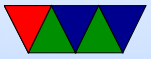
# Why re-use old models?

- In theory, validated
- Starting from scratch is hard!
- Especially if you're a grad student



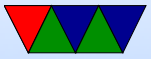
# Hotleakage

- (Zhang, Parikh, Sankaranarayanan, Skadron, Stan. 2003)
- Leakage exponentially related to temperatures



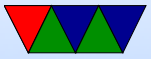
# Hotspot

- (Huang, Ghosh, Sankaranarayanan, Skadron, and Stan. 2005).
- Estimate thermal constraints on CPU
- Heat flow problems. Springs?
- Various papers from version 1.0 to 5.0



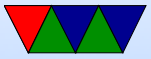
# How can we speed up estimating power?

- "cycle-accurate" simulators are slow
- Similar (though less detailed, but more real-world) data available through performance counters
- Can also use pure software methods based on instruction mix (Tiwari et al 1996). Something like Valgrind. perf-counter probably be better



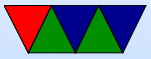
# Using Performance Counters

- Counters readily available, no need to buy a power meter for each node
- Real-time estimates, not after-the-fact
- limitation: only 2 - 8 counters at a time typically (why a lot of papers use AMD (4) or Pentium 4 (18)

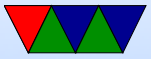


# Singh, Bhadauria, McKee. DasCMP 2008

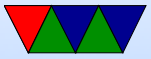
- On AMD Phenom 5.8% error on NAS, 3.9% SPEC-OMP, 7.2% SPEC2006
- FP Units – RETIRED_MMX_AND_FP_INSTRUCTIONS:AI
- Memory – L2_CACHE_MISS:ALL
- L3 has high miss rate. Non-inclusive so essentially large victim cache for L2. So L2 miss good proxy for memory.
- Stalls – DISPATCH_STALLS



- Proxy for out-of-order
- Instructions Retired — RETIRED_UOPS
- Watts Up Power Meter
- Try 13 events
- Only 4 counters available, highest correlation
- Use custom ubench to make model
- FP Correlates negative, longer latency

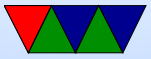


- Look at temperature. Related, but no per-core temp available
- perfmon2
- In software maintain power envelope

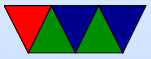


# Power Prediction for Intel Xscale Processors Using Performance Monitoring Unit Events

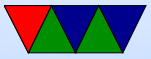
- (Contreras, Martonosi. ISLPED 2005)
- First-order linear model
- Within 4
- Instructions Executed; IPC highly correlates with power
- Data Dependencies; highly correlates



- Instruction Cache Miss
- TLB Miss
- $Power_{CPU} = \alpha_1(IFetch_{miss}) + \alpha_2(DataDep) + \alpha_3(DataTLB_{miss}) + \alpha_4(InstTLB_{miss}) + \alpha_5(InstExec_{miss}) - \kappa_{CPU}$
- Kcpu is baseline CPU power
- Want to track SDRAM, not always able to get enough relevant events.

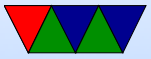


- Measure with oscilloscope
- Custom performance counter implementation
- Only can produce results after the fact, as cannot measure all 5 events at once.
- SPEC CPU2000, Java CDC, JAVA CLDC (C and Java represent embedded systems) MiBench
- Note, no floating point. Why?
  PXA255 ARMv5TE – No floating point



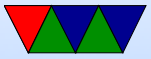
# In-chip power breakdowns

- Runtime Power Monitoring in High-End Processors: Methodology and Empirical Data
- (Isci and Martinosi. Micro 2003)
- Measure all 18 P4 counters plus power, provide breakdown.
- Custom counter interface.
- Clamp Ammeter. All 17 data lines, benchmarks to



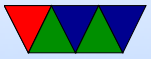
determine which relevant.

- SPEC Benchmarks
- 22 Processor Components
- Hard to validate



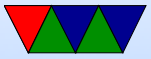
# Non-academic

- Intel ALPS and ASIM
- (Powell, Biswas, Emer, Mukerjee, Sheikh, Yardi. HPCA 2009).



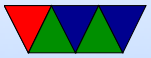
# Full-System estimation

- Complete System Power Estimation: A Trickle-Down Approach Based on Performance Events
- (Bircher, John).
- Cache and DMA events correspond to offcore power
- Get within 9
- Zedlewski disk power estimation



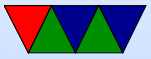
# RAPL

- Does this all in hardware
- Uses performance info, and has a leakage model based on temperature and voltage
- Used to enable Turbo-boost and to keep with in power envelope, but also exposed to the user.



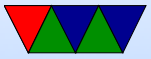
# Performance Counters for Heat

- (Chung and Skadron, 2006)
- On-chip temperature sensors are available
- May only be one, and might not be at a useful location
- Use perf counters to estimate, much like with power
- Hysteresis. Need to simulate a long time (seconds to minutes) because it takes a while to heat/cool a big heatsink.



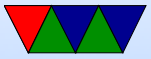
# Estimating for Projects

- Calculate base load (when idle)
- Pick some events you might think will work. Be sure to say why you chose.
- Make a few microbenchmarks. Measure their performance and power. (An integer benchmark might just be a simple loop, memory might be reading large array, etc.)
- Assign weights and estimate power



- Then try applying the values you found (your model) to some unrelated program and see how well it matches.

- Linear? Piece-wise linear? Fancier regressions?



# Issues with Raspberry Pi

- Potential narrow range of power. Only 5W?