ECE 571 – Advanced Microprocessor-Based Design Lecture 15

Vince Weaver

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

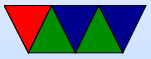
vincent.weaver@maine.edu

19 March 2013

Prelim Comments

- Question 2a: How is cortex A9 different than x86?
  I was looking for kernel vs user events distinction. I also accepted no L2 cache events. Part credit for saying different number of counters.

- Question 3b: Results were Cold Cold Cold Hit *Cold Conflict*
  Second-to-last is cold because never appeared in cache before.
  Last is conflict because it was kicked out (not capacity

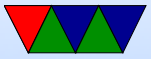


as cache is not full).

- Question 4a: Physically tagged TLB performance
  With virtual memory, a physically tagged cache has to do a virtual to physical conversion *every* cache access, so the TLB is the critical step.

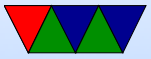
- Question 5a: There are 11 total branches, as the instruction is `ble` not `blt`

- Question 5b: Surprise that 2-bit branch predictor no better than static?
  2-bit predictors are only better than static when loops go



unexpected ways or change their behavior periodically. Until they "warm up" can have poor behavior (if initialized to all 0s). How to avoid that?

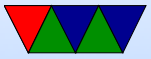
- Question 5c: When does global history help?
  Nested loops was a common answer; this is possible but would need more explanation.
  Function calls is a more typical example. Any time where you can get to a branch by multiple paths, having global history taken into account lets you separate the behavior of the different paths.



HW3 Comments

- Behavior is hard to see on these plots on ARM. ATLAS Matrix Multiply would have different results, as optimized to avoid cache problems. Only naive or random implementations that blast memory show cache effects clearly.

- Note how huge decrease in branch misses only makes 5% overall cycles difference



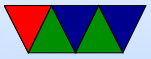
Project Discussion

Conduct power/energy/performance research that's applicable to embedded systems.

See website for more details.

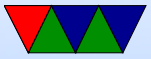
- Choose a topic by March 26th.
  You can work with a partner, but more will be expected.

- Two status updates; one to make sure the hardware/benchmarks are available, one to make sure you are on track. (Preliminary April 2nd and April 16th)



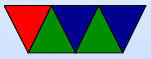
- Write a short paper (at least 6 pages) due by May 2nd
  - Introduction
  - Related work. Do a literature search to see if it has been done before. It is OK if it has been, replicating previous results is useful.
  - Experimental Setup
  - Results

- Give a 20-minute presentation the last week of class describing the project and findings.

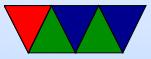


Possible Topics

- Power/Energy overhead from prefetching (can turn off prefetching on some systems, like core2 and cortex a9)
- Power/Energy overhead from branch prediction (system that can turn off branch prediction?)
- Power/Energy overhead from caches (find a system you can disable the cache?)
- Power/Energy vs performance on various ARM processors

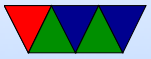


- Estimating power/energy using performance counters
- Power/Energy implications from frequency scaling (operating system? Linux/OSX/Windows)
- Validating the RAPL measurements on real hardware.
- More to come, open to suggestions

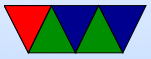


Equipment Available for Use

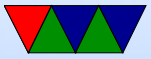
- ARM Cortex A9 Pandaboard with performance counters
- ARM Cortex A8 Beagleboard with power measurement and performance counters
- ARM Cortex A15 Chromebook with performance counters
- ARM 1176 Raspberry Pi (perf counters not working yet)
- Watts Up Pro logging power meter



- Ivy Bridge laptop with RAPL power counters
- Measuring power/energy with simulators
- Various other machines (Pentium II, Pentium D, SPARC64, AVR)
- Feel free to use any other systems you have available



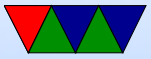
Power and Energy



Definitions and Units

People often say Power when they mean Energy

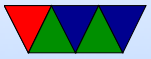
- Dynamic Power – only consumed while computing
- Static Power – consumed all the time.
  Sets the lower limit of optimization
- Energy – Joules, kWH, mAh (batteries)
- Power – Energy/Time – Watts, Volt-Amps (for A/C)



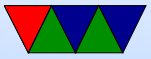
Power and Energy in a Computer System

Mahersi and Vardhan, PACS'04.

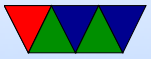
- Measured V and Current. P=IIR. V=IR P=IV, subtractive
- Thinkpad Laptop, Pentium M, 256M, 14" display
- Hard Drive 0.5-2W (Flash/SSD less)
- LCD 1W (slightly more black than white)



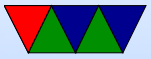
- Backlight Inverter (this is before LED) 1-4W depending on brightness
- Total System Power 14-30W
- CPU 2-15W (with scaling)
- GPU 1-5W
- Memory 0.45 - 1.5W
- Power Supply Loss - 0.65W
- Wireless 0.1 - 3W (wifi on cellphones)



- CDROM 3-5W
- (USB 2.0 – 5V, can draw 5 units of 100mA each, 2.5W)



CPU Power and Energy

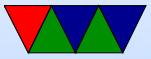


CMOS Dynamic Power

- $P = C\Delta V V_{dd} \alpha F$
  Charging and discharging capacitors big factor ($C\Delta V V_{dd}$) from V_{dd} to ground
  α is activity factor, transitions per clock cycle
  F is frequency

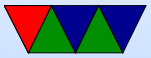
- How can you reduce Dynamic Power?

- Also some loss due to pass-through (V momentarily connected to ground)



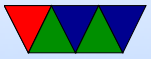
CMOS Dynamic Power Reduction

- Reduce C – scaling
- Reduce Vdd – eventually hit transistor limit
- Reduce α (design level)
- Reduce F – makes processor slower



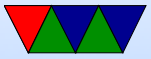
Metrics to Optimize

- Power
- Energy
- MIPS/W, FLOPS/W (don't handle quadratic V well)
- $Energy * Delay$
- $Energy * Delay^2$



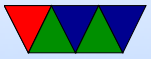
Power Optimization

- Does not take into account time. Lowering power does no good if it increases runtime.



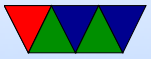
Energy Optimization

- Lowering energy can affect time too, as parts can run slower at lower voltages

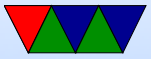


Energy Delay – Watt/t*t

- Horowitz, Indermaur, Gonzalez (Low Power Electronics, 1994)
- Need to account for delay, so that lowering Energy does not made delay (time) worse
- Voltage Scaling – in general scaling low makes transistors slower
- Transistor Sizing – reduces Capacitance, also makes transistors slower

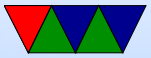


- Technology Scaling – reduces V and power.
- Transition Reduction – better logic design, have fewer transitions
  Get rid of clocks? Asynchronous? Clock-gating?
- Example with inverse ED2 (higher better):
  Alpha 21064, SPEC=155, Power=30W, SPEC*SPEC/W=8
  PPC603, SPEC=80, Power=3W, SPEC*SPEC/W=2100



Energy Delay Squared– E*t*t

- Martin, Nyström, Pénzes – Power Aware Computing, 2002
- Independent of Voltage in CMOS
- Et can be misleading
  Ea=2Eb, ta=tB/2
  Reduce voltage by half, Ea=Ea/4, ta=2ta, Ea=Eb/2, ta=tb
- Can have arbitrary large number of delay terms in Energy



product, squared seems to be good enough