ECE 571 – Advanced Microprocessor-Based Design Lecture 16

Vince Weaver

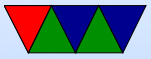
http://www.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

29 March 2016

Announcements

- Project topics
- HW#7 was sent out, simple, by Thurs
- ECE435 comments



Go Over Midterms

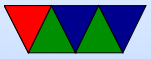
Average grade is an 83%.

1. Performance/Benchmarking

(a) Benchmark question:

Companies often have build farms. Why? Slow to build (try android, or Linux-distro, or mozilla) Also, some will build and test any time you make a git commit

Was just looking for a compiler benchmark (could be anything, but SPEC CPU does include a gcc



benchmark).

Integer over FP partial credit.

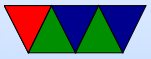
Integer benchmarks can have more unpredictable behavior, but if that's what you're using the machine for it's what you should test against.

Not gprof.

(b) perf-record/perf-annotate: memory is slow! (key theme from this class)

skid is likely the reason

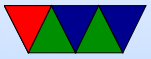
2. Power



(a) Table
(b) Energy
(c) Performance – just plain time, not ED or ED2
(d) Energy Delay
(e) Why Energy Delay? Takes performance into account

3. Branch Prediction

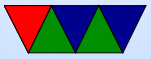
(a) 1 miss/9 hits
(b) 1 miss/9 hits – so why use dynamic? It potentially does better in the non-loop case (loops are easy)
(c) Why worse prediction on embedded? Likely have



smaller/less tuned branch predictors compiler? in theory, but not sure compilers ever optimize for branch prediction rates. maybe on arm/cond exec?

4. Cache

(a) 4/4/24
(b) Cache example, most did well. The problem was on misses. All were cold. Cold if you've never been in cache before, doesn't matter if you kick something out. Even with more ways it doesn't help a cold miss.
(c) Prefetch. Can hurt



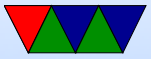
5. Virtual Memory

(a) Advantages: Memory Protection, memory can appear larger, reduce fragmentation

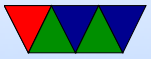
(b) two pages can have same virt addr as long as physical is different TLB has nothing to do with it. TLB is just a page table cache, nothing more.

(c) How can you make a test? Don't leave blank!
Walk over big chunks of memory. Force context switch/flush TLB.

(d) Why is it wrong? TLB counter wrong? You wrote the code wrong? several TLB levels page size may be

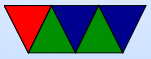


larger than 4k



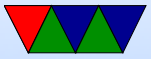
CPU Power and Energy

Some things based on questions in class last time.



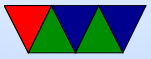
DVFS and other CPU Power/Energy Saving Methods

- A lot of related work
- Will focus on actual implementations rather than academic papers this time

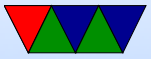


DVFS

- Voltage planes – on CMP might share voltage planes so have to scale multiple processors at a time
- DC to DC converter, programmable.
- Phase-Locked Loops. Orders of ms to change. Multiplier of some crystal frequency.
- Senger et al ISCAS 2006 lists some alternatives. Two phase locked loops? High frequency loop and have programmable divider?

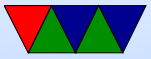


- Often takes time, on order of milliseconds, to switch frequency. Switching voltage can be done with less hassle.



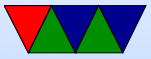
Adaptive Body Biasing

- Related to but not always considered part of DVFS
- Control voltage applied to body
- Change the threshold voltage
- Reduces leakage but slows performance



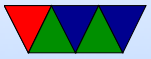
Cache Power and Energy

Large area, low-hanging fruit



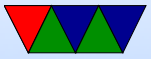
Decay Caches

- Kaxiras, Ho, Martinosi (ISCA 2001)
- Turn off cache lines not being used to reduce leakage
- DRAM cache with no refresh
- Decayed values can be re-fetched from memory. Tradeoff.



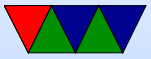
Drowsy Caches

- Flautner, Kim, Martin, Blaauw, Mudge. ISCA 2002.
- Move cold cache lines into "drowsy" mode.
  Lower power enough to hold state, not enough to lose contents. Reduce leakage. Better than decay as not lose data.



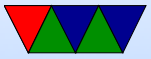
Adaptive Caches

- Albonesi (Micro 1999). Manually turn off ways in cache with an instruction.
- Size the caches



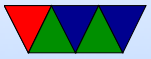
Cache Compression

- Dynamic zero compression for cache energy reduction (L Villa, M Zhang, K Asanović. Micro 2001).
- Cache Compression ("sign compression" – top bits) Energy savings 20% (simulated) (Kim, Austin, Mudge WMPI 2002)



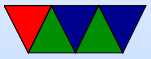
Banking and Filtering

- Filter cache, banking (only have half of cache active) (Mudge 2001)
- Slowing Down Cache Hits, Banked Data Cache. (Huang, Renau, Yoo, and Torrellas. Micro 2000.)
- Vertical Banking, Horizontal Banking (Su and Despain, ISLPED 1995).



Code Scheduling

- Can Schedule code for lower power.
- Better cache rates lower power. performance/power can go hand in hand. (Kandemir, Vijaykrishnan, Irwin)



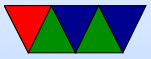
Branch Predictors

- Parikh, Skadron, Zhang, Barcella, Stan

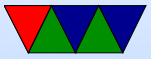
- 4 concerns:

  1. Accuracy. Not affect power, but performance
  2. Configuration (may affect power)
  3. Number of lookups
  4. Number of updates

- Tradeoff power vs time.

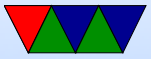


- brpred can be size of small cache, 10% of power
- Can use banking to mitigate



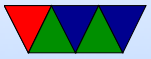
Branch Predictors

- can watch icache, not activate predictor if nobranches
- Pipeline gating, keep track of each predicted branch confidence. If confidence hits certain threshold, stop speculating. Show this may or may not be good.
- Integer code, large predictors good
- FP, tight loops, predictors not as important.



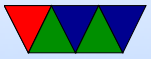
Branch Predictor Evaluation

- (Strasser, 1999). Simulation, small branch predictor can help energy.
- (Co, Weikle, Skadron) Formula for break even point. Leakage matters, what brpred hides is stall cycles.
- SEPAS: A Highly Accurate Energy-Efficient Branch Predictor (Baniasadi, Moshovos. ISLPED 2004). Once a branch prediction reaches steady state (unlikely to change) stop accessing/updating predictor, saving



energy.

- Low Power/Area Branch Prediction Using Complementary Branch Predictors (Sendag, Yi, Chuang, Lija. IPDPS 2008)
  Complementary Branch Predictor to handle the tough cases.

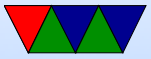


Prefetching

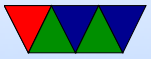
- Prefetching does not get looked at as closely.
  Various studies show it can be a win energy wise, but it is a close thing.

- (Guo, Chheda, Koren, Krishna, Moritz. PACS'04)
  HW Prefetch increase power 30%; have compiler help augment with hints, filters.

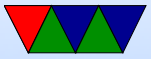
- (Tang, Liu, Gu, Liu, Gaudiot. Computer Architecture Letters, 2011).



Mixed results.



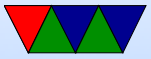
TLB Energy



TLB Optimization – Assume in Same Page

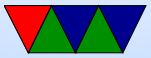
- Optimizing instruction TLB energy using software and hardware techniques (Kadayif, Sivasubramaniam, Kandemir, Kandiraju, Chen. TODAES 2005).
  Don't access TLB if not necessary. Compare to last access (assume stay in same page) Circuit improvements

- (Kadayif,Sivasubramaniam, Kandemir, Kandiraju, Chen. Micro 2002)
  Generating Physical Addresses Directly for Saving Instruction TLB Energy Cache page value.



TLB Optimization – Use Virtual Caches

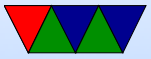
- (Ekman and Stenström, ISLPED 2002) Use virt address cache. Less TLB energy, more snoop energy. TLB keeps track of shared pages.



TLB Optimization – Reconfiguring

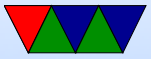
- (Basu, Hill, Swift. ISCA 2012) Reducing Memory Reference Energy with Opportunistic Virtual Caching
  Have the OS select if memory region physical or virtual cached.

- (Delaluz, Kandemir, Sivasubramaniam, Irwin, Vijaykrishnan. ICCD 2013) Reducing dTLB Energy Through Dynamic Resizing.
  Size TLB as needed, shutting off banks. Easier if fully-associative.



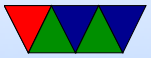
TLB Optimization – Memory Placement

- (Jeyapaul, Marathe, Shrivastava, VLSI'09) Try to keep as much in one page as possible via compiler.
- Energy Efficient D-TLB and Data Cache using Semantic-Aware Multilateral Partitioning (Lee, Ballapuram. ISLPED'03) Split memory regions by region (text/data/heap). Better TLB performance, better energy.



Bus Protocols

- Bus Protocols
- Cache-Coherence Protocols



Busses

- Grey Code, only one bit change when incrementing. Lower energy on busses? (Su and Despain, ISLPED 1995).