Runtime Power Monitoring in High-End Processors:

Methodology and Empirical Data

Canturk Isci & Margaret Martonosi
Princeton University

MICRO-36

Motivation

- Power is important!
- Measurement/Modeling techniques help guide design of power-aware and temperature-aware systems
- Real-system measurements:
  - Help observe long time periods
  - Help guide on-the-fly adaptive management
- Our work: live, run-time, per-component power/thermal measures

Simulation vs. Multimeters...

Simulation:
+ Arbitrary detail
+ Common base
- Slow
- Possibly inaccurate

Multimeter:
+ Fast
+ Fairly Accurate
+/- Existing systems
- No on-chip detail

Counter-Based Power Estimation:
+ Fast (Real-time)
+ Offers estimated view of on-chip detail
But:
1) Are the "right" counters available?
2) How accurate is it?

Questions We Answer

- To what extent can CPU performance counters act as proxies to estimate CPU power?
- Can counter-based power estimations offer useful accuracy compared to multimeter measurements?
- What are some interesting uses of this measurement approach?

Power Simulation: Overview

- Idealized view: For all components in a processor chip...

Power of component I =

MaxPower[I] * ArchScaling[I] * AccessRate[I]

Die area and Capacitance estimate

Cycle-level simulator gathers event counts, activity factors and adjusts scaling...

Counter-Based Power Estimation: An Overview of Our Approach

- Idealized view: For all components in a processor chip...

Power of component I =

MaxPower[I] * ArchScaling[I] * AccessRate[I]

Die area + Stressmarks

From microarch. properties

CPU Performance Counters!

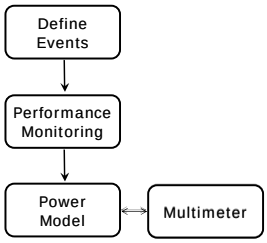
- More realistic view: Handle non-linear scaling...

... + NonGatedPower[I]

Empirical Multimeter measurement

Counter-Based Power Estimation: General Implementation

- Develop per-component access heuristics and scaling factors
- Build performance monitoring SW to sample hardware event counts
- Use stressmarks to calibrate scaling factors
- Use multimeter measurements to validate counter-based power estimates



Specific Example: Intel Pentium 4

- Define components: L1 cache, BPU, Regs, … whose powers we'll model:
  - from annotated layout
- Determine combination of P4 events that represent component accesses best
- Gather counter info with minimal power overhead and program interruption using built Linux loadable kernel module
- Convert counter based access rates into component powers
- Verify total power against measured processor power

Complete Example: Retirement Logic

- Retirement Logic defined from annotated die layout
- Access rate approximation based on performance counters: $\frac{UopsRetired}{\Delta Cycles}$
- Retirement Power:
  - Can retire at most 3 uops/cycle

$$Power(Ret) = \left[\frac{AccessRate(Ret)}{3}\right] \bullet MaxPower(Ret) + ClkPower(Ret)$$

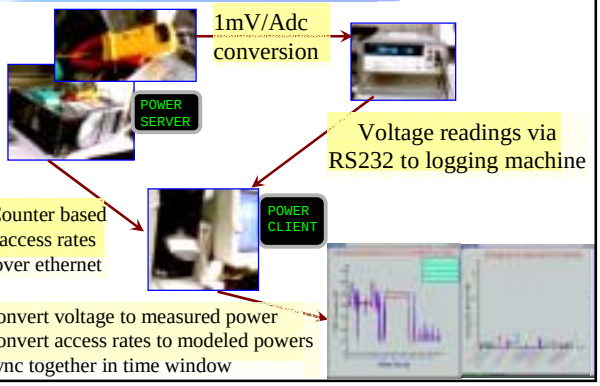
- Initial MaxPower: Area-based estimate
  - MaxPower = Area% x Max Processor Power = 4.7W
- Estimates after stressmark tuning:
  - MaxPower = 1.5W; ClkPower = 2W
- Final hardcoded power equation for retirement logic:

$$\text{Power(Ret)} = \frac{\text{AccessRate(Ret)}}{(3)} \bullet \lceil 0.5 \cdot (3) \rceil + 2.0$$

Questions We Answer

- To what extent can CPU performance counters act as proxies to estimate CPU power?
- Can counter-based power estimations offer useful accuracy compared to multimeter measurements?
  - Validation Setup
  - Stressmark measurements
  - SPEC measurements
  - Component-wise validation
- What are some interesting uses of this measurement approach?

Validation Measurement Setup



Questions We Answer

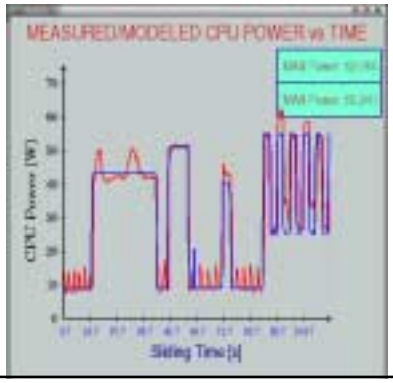
- To what extent can CPU performance counters act as proxies to estimate CPU power?
- Can counter-based power estimations offer useful accuracy compared to multimeter measurements?
  - Validation Setup
  - Stressmark measurements
  - SPEC measurements
  - Component-wise validation
- What are some interesting uses of this measurement approach?

Counter-based Power Estimation: Validation Step 1

—Measured
— Modeled

Using die area proportions for max power

Counter-based Power Estimation: Validation Step 2



—Measured
— Modeled

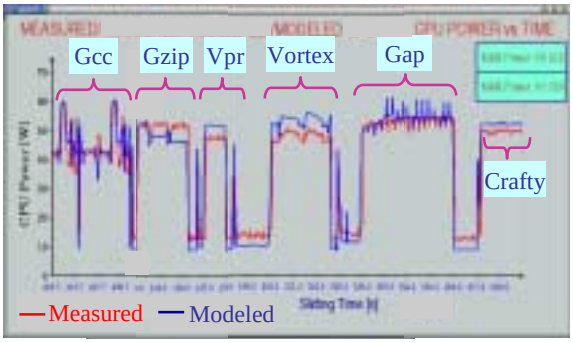
Adjusting max power coeff-s on stressmarks

Questions We Answer

- To what extent can CPU performance counters act as proxies to estimate CPU power?
- Can counter-based power estimations offer useful accuracy compared to multimeter measurements?
  - Validation Setup
  - Stressmark measurements
  - SPEC measurements
  - Component-wise validation
- What are some interesting uses of this measurement approach?

15

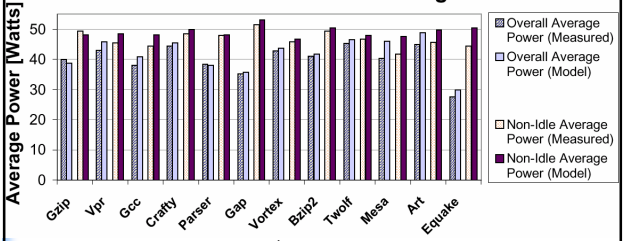
Validation for Accuracy: SPEC Results



16

Average SPEC Total Powers

Measured vs. Counter Based Average Power



- 1st set: Overall, 2nd set: Non-idle power
- Average difference between measurement and estimation: 3W
- Worst case: Equake (5.8W)

17

Questions We Answer

- To what extent can CPU performance counters act as proxies to estimate CPU power?
- Can counter-based power estimations offer useful accuracy compared to multimeter measurements?
  - Validation Setup
  - Stressmark measurements
  - SPEC measurements
  - Component-wise validation
- What are some interesting uses of this measurement approach?

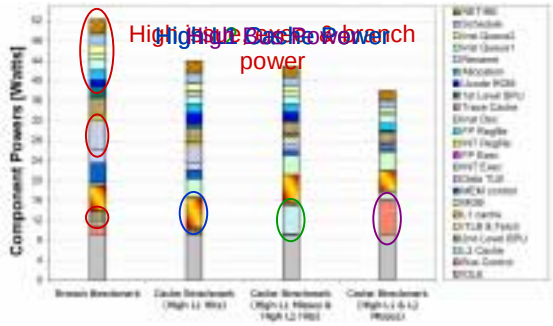
18

Per-Component Validation

- What we would like to do
  - Compare against gold-standard simulator, or…
  - Compare against detailed published per-component measured data
- What we can do
  - Validate using per-component stressmarks
  - Sanity check on trends in real-benchmarks

19

Validation for Fidelity: Benchmark Power Breakdowns



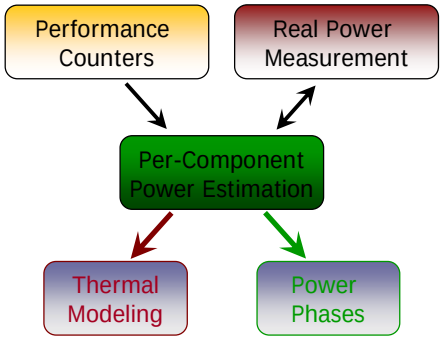
20

Questions We Answer

- To what extent can CPU performance counters act as proxies to estimate CPU power?
- Can counter-based power estimations offer useful accuracy compared to multimeter measurements?
- What are some interesting uses of this measurement approach?

21

Counter-Based Power Estimation: Uses and Big Picture



22

Conclusions

- Contributions:
  - Performance counter based runtime power model and runtime verification with synchronous real power measurement for arbitrarily long timescales!
  - Physical component based power estimates for processor, which can be used in power phase analyses and thermal modeling
- Outcome:
  - We can perform reasonably accurate runtime power estimates without inducing any significant overhead to power profile

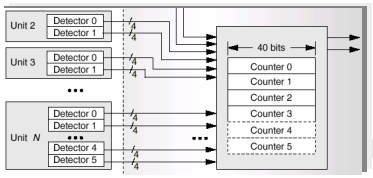
23

EOP

24

- IF TIME PERMITS:

Live CPU Performance Monitoring with Hardware Counters

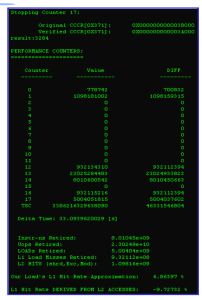


[Sprunt, 2002]

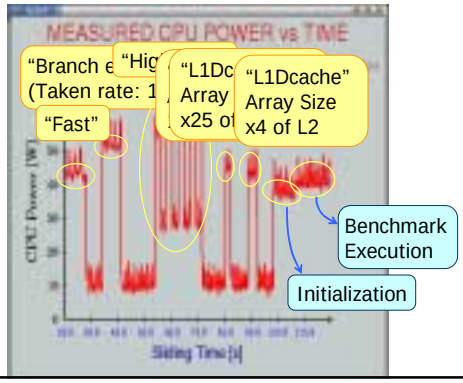
- Most CPUs have hardware performance counters
- Intel P4 Performance Monitoring HW:
  - 18 Event Counters to count hundreds of possible events
  - How to Count? 18 Counter Config Control Registers
  - What to Count? 45 Event Selection Control Registers
  - Plus additional control registers...

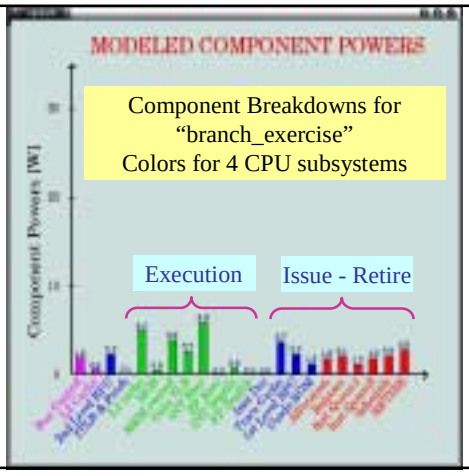
Our Event Counter: Performance Reader

- Performance Reader implemented as Linux Loadable Kernel Module
  - Implements 6 syscalls:
    - *select_events()*
    - *reset_event_counter()*
    - *start_event_counter()*
    - *stop_event_counter()*
    - *get_event_counts()*
    - *set_replay_MSRs()*
- User Level Interface:
  - Defines the events → Starts counters
  - Stops counters → Reads counters & TSC

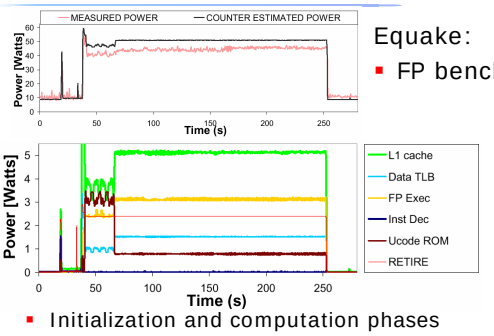


Multimeter Measurements: Stressmarks





SPEC2000 Results: Example



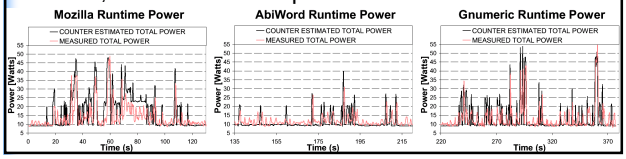
Equake:

- FP benchmark

- Initialization and computation phases
- FP intensive mesh computation phase
- Initialization with high complex IA32 instructions

Desktop Applications

- We aim to track low power utilizations as well.
- Desktop applications are usually low power with intermittent power bursts
- 3 applications, with common operations such as open/close application, web, streaming media, text editing, save to disk, statistical computations.



Related Work

- **Implementing counter readers:**
  - PCL [Berrendorf 1998], Intel VTune, Brink & Abyss [Sprunt 2002]
- **Using counters for Power:**
  - CASTLE [Joseph 2001], power profilers
  - event driven OS/cruise control [Bellosa 2000,2002]
- **Real Power Measurement:**
  - Compiler Optimizations [Seng 2003]
  - Cycle-accurate measurement with switch caps [Chang 2002]
- **Power Management and Modeling Support:**
  - Instruction level energy [Tiwari 1994]
  - PowerScope: Procedure level energy [Flinn 1999]
  - Event counter driven energy coprocessor [Haid 2003]
  - Virtual Energy Counters for Mem. [Kadayif 2001]
  - ECOsystem: OS energy accounting [Ellis 2002]

32

Our Work in Comparison

- Power estimation for a complex, aggressively clock-gated processor
- Component power estimates with physical binding to die layout
  - Laying the groundwork for thermal modeling
- Portable implementation with current probe and power server LKM
- Power oriented phase analysis with acquired power vectors

33