ECE 571 – Advanced Microprocessor-Based Design Lecture 6

Vince Weaver

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

vincent.weaver@maine.edu

8 February 2018

Announcements

- HW#1 grades out
- HW#3 will be posted
- Note, the equake benchmark takes a while to run (a few minutes). Don't give up on it.

Paper Discussion

Producing Wrong Data Without Doing Anything Obviously Wrong! by Mytkowicz, Diwan, Hauswirth and Sweeney, ASPLOS'09.

Power and Energy

Definitions and Units

People often say Power when they mean Energy

- Energy – Joules, kWH (3.6MJ), Therm (105.5MJ), 1 Ton TNT (4.2GJ), eV (1.6×10^{-19} J), BTU (1055 J), horsepower-hour (2.68 MJ), calorie (4.184 J)
- Power – Energy/Time – Watts (1 J/s), Horsepower (746W), Ton of Refrigeration (12,000 Btu/h)
- Volt-Amps (for A/C) – same units as Watts, but not same thing
- Charge – mAh (batteries) – need V to convert to Energy

Power and Energy in a Computer System

Power Consumption Breakdown on a Modern Laptop, A. Mahersi and V. Vardhan, PACS'04.

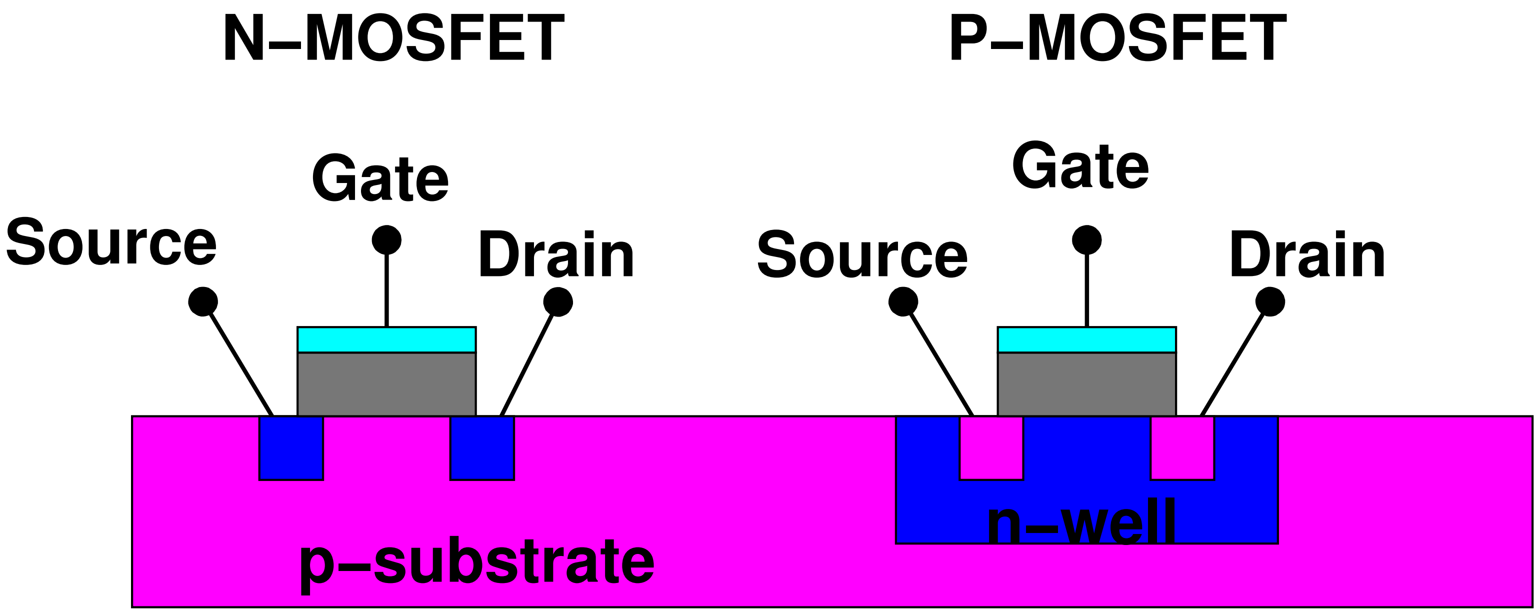
- Old, but hard to find thorough breakdowns like this
- Thinkpad Laptop, 1.3GHz Pentium M, 256M, 14" disp
- Oscilloscope, voltage probe and clamp-on current probe
- Measured V and Current. P=IIR. V=IR P=IV, subtractive for things w/o wires
- Total System Power 14-30W
- Old: no LED backlight, no SDD, etc.

Modern results are from CUGR/REU student research.

	Laptop (2004)	Modern	Server?
Hard Drive	0.5-2W	5W	
LCD	1W		
Backlight	1-4W		
CPU	2-15W	60+W	
GPU	1-5W	50+W	
Memory	0.5-1.5W	1-5W	
Power Supply	0.65W		
Wireless	0.1 - 3W		
CD-ROM	3-5W		
USB	(max 2.5W)		
USB keyboard		0.04W	
USB mouse		0.03W	
USB flash		0.5W	
USB wifi		0.5W	

CPU Power and Energy

CMOS Transistors



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
- α often approximated as $\frac{1}{2}$, $\Delta V V_{dd}$ as V_{dd}^2 leading to $P \approx \frac{1}{2} C V_{dd}^2 f$
- Some pass-through loss (V momentarily shorted)

CMOS Dynamic Power Reduction

How can you reduce Dynamic Power?

- Reduce C – scaling
- Reduce V_{dd} – eventually hit transistor limit
- Reduce α (design level)
- Reduce f – makes processor slower

CMOS Static Power

- Leakage Current – bigger issue as scaling smaller. Forecast at one point to be 20-50% of all chip power before mitigations were taken.
- Various kinds of leakage (Substrate, Gate, etc)
- Linear with Voltage: $P_{static} = I_{leakage}V_{dd}$

Leakage Mitigation

- SOI – Silicon on Insulator (AMD, IBM but not Intel)
- High-k dielectric – instead of SO2 use some other material for gate oxide (Hafnium)
- Transistor sizing – make only the critical transistors fast; non-critical ones can be made slower and less leakage prone
- Body-biasing
- Sleep transistors

Notes on Process Technology

- 65nm – 2006
  p4 to core2, IBM Cell
  1.0v, High-K dielectric, gate thickness a few atoms
  193/248nm light (UV)

- 45nm – 2008
  core2 to nehalem
  large lenses, double patterning, high-k

- 32nm – 2010

sandybridge to westmere
immersion lithography

- 22nm – 2012 ivybridge, haswell
  oxide only 0.5nm (two silicon atoms)
  fin-fets

- 14nm and smaller – ??
  Extreme UV (13.5nm light, hard-vacuum required)?
  Electron beam?

Notes on Process Technology

- TI-OMAP cell phone processor (more or less discontinued by TI, big layoffs in 2012)
  Beagle Board and Gumstix OMAP35?? – 65nm
- OMAP4460 (Pandaboard) 45nm
- Cortex A15 28nm
- Rasp-pi BCM2835 – 45nm?

Total Energy

- $E_{tot} = [P_{dyanmic} + P_{static}]t$
- $E_{tot} = [(C_{tot}V_{dd}^2\alpha f) + (N_{tot}I_{leakage}V_{dd})]t$

Delay

- $T_d = \frac{C_L V_{dd}}{\mu C_{ox}(\frac{W}{L})(V_{dd} - V_t)}$
- Simplifies to $f_{MAX} \sim \frac{(V_{dd} - V_t)^2}{V_{dd}}$
- If you lower f, you can lower V_{dd}

Thermal Issues

- Temperature and Heat Dissipation are closely related to Power
- If thermal issues, need heatsinks, fans, cooling

Metrics to Optimize

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

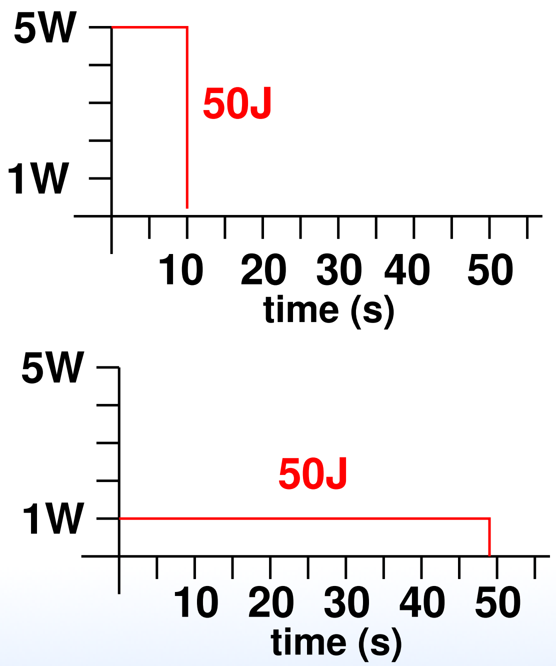
Power Optimization

- 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

Which is better?



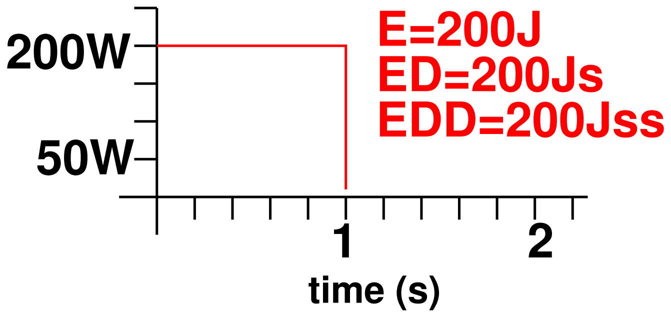
Energy Delay – Watt/t*t

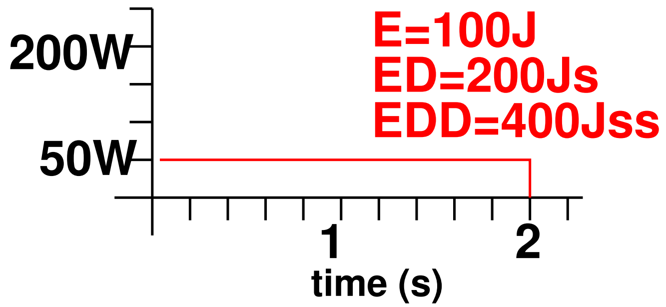
- Horowitz, Indermaur, Gonzalez (Low Power Electronics, 1994)
- Need to account for delay, so that lowering Energy does not made delay (time) worse
- Voltage Scaling – in general scaling low makes transistors slower
- Transistor Sizing – reduces Capacitance, also makes transistors slower

- Technology Scaling – reduces V and power.
- Transition Reduction – better logic design, have fewer transitions
  Get rid of clocks? Asynchronous? Clock-gating?

ED Optimization

Which is better?





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

Energy Delay / Energy Delay Squared

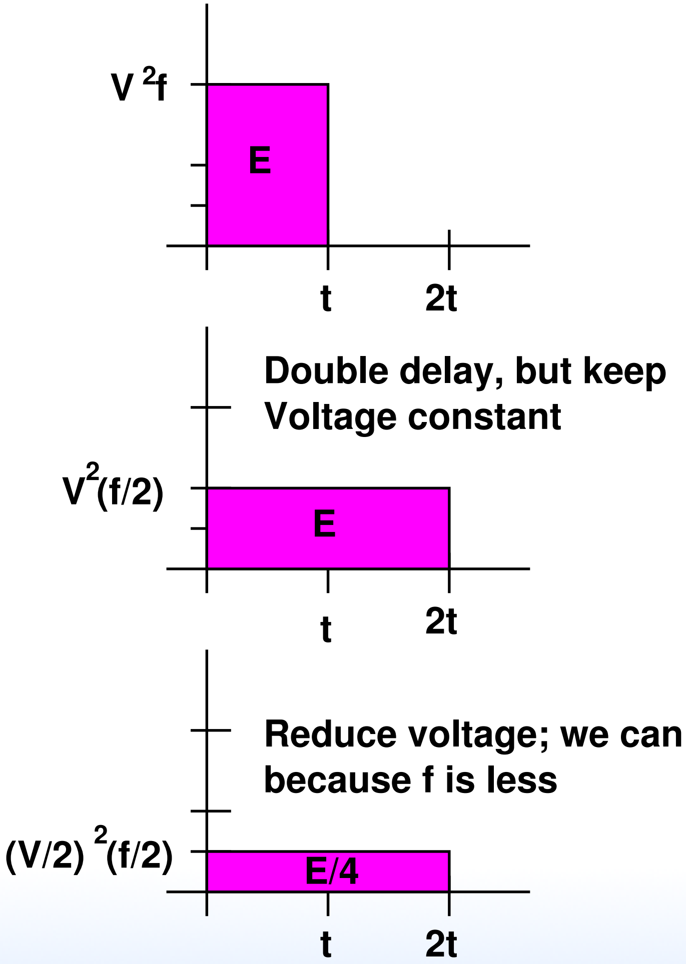
Lower is better.

Energy	Delay	ED	ED^2
5J	2s	$10Js$	$20Js^2$
5J	3s	$15Js$	$45Js^2$

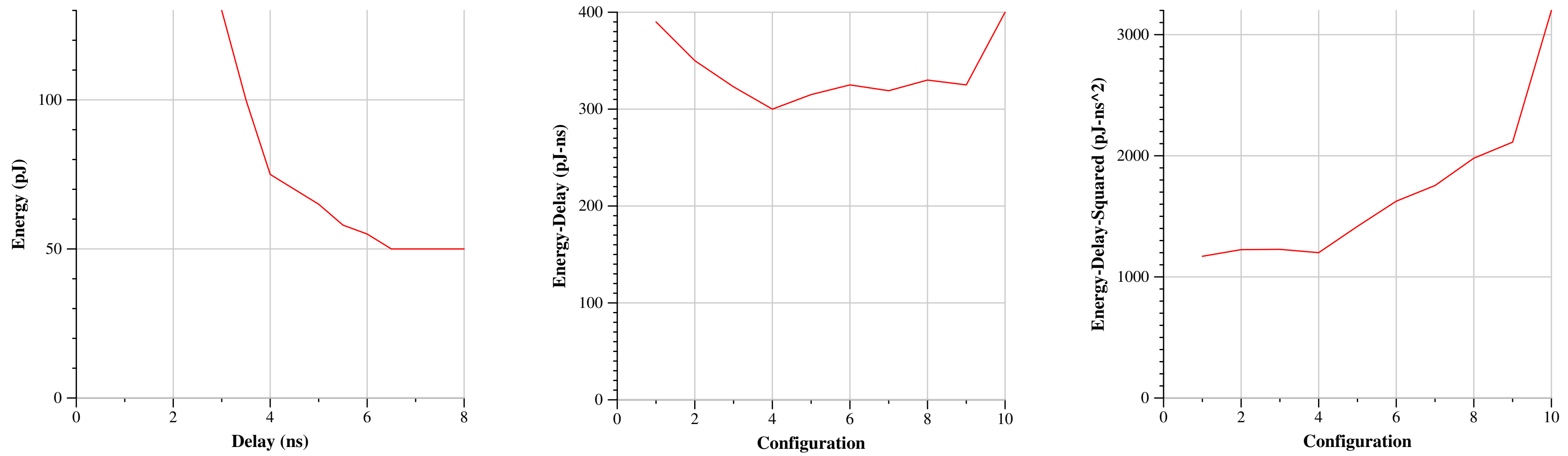
Same ED, Different ED^2

Energy	Delay	ED	ED^2
5J	2s	$10Js$	$20Js^2$
2J	5s	$10Js$	$50Js^2$

Energy Example



Energy-Delay Product Redux



Roughly based on data from "Energy-Delay Tradeoffs in CMOS Multipliers" by Brown et al.

Raw Data

Delay	Energy	ED	ED^2
3	130	390	**1170**
3.5	100	350	1225
3.8	85	323	1227
4	75	**300**	1200
4.5	70	315	1418
5	65	325	1625
5.5	58	319	1755
6	55	330	1980
6.5	**50**	390	2535
8	50	400	3200

Other Metrics

- $Energy - Delay^n$ – choose appropriate factor
- $Energy - Delay - Area^2$ – takes into account cost (die area) [McPAT]
- Power-Delay – units of Energy – used to measure switching
- Energy Delay Diagram – [SWEEP]
- Energy-Delay-FIT (reliability?)

Measuring Power and Energy

Why?

- New, massive, HPC machines use impressive amounts of power
- When you have 100k+ cores, saving a few Joules per core quickly adds up
- To improve power/energy draw, you need some way of measuring it

Energy/Power Measurement is Already Possible

Three common ways of doing this:

- Hand-instrumenting a system by tapping all power inputs to CPU, memory, disk, etc., and using a data logger
- Using a pass-through power meter that you plug your server into. Often these will log over USB
- Estimating power/energy with a software model based on system behavior

Measuring Power and Energy

- Sense resistor or Hall Effect sensor gives you the current
- Sense resistor is small resistor. Measure voltage drop. Current V=IR Ohm's Law, so V/R=I
- Voltage drops are often small (why?) so you made need to amplify with instrumentation amplifier
- Then you need to measure with A/D converter
- $P = IV$ and you know the voltage
- How to get Energy from Power?

Hall Effect Current Sensors

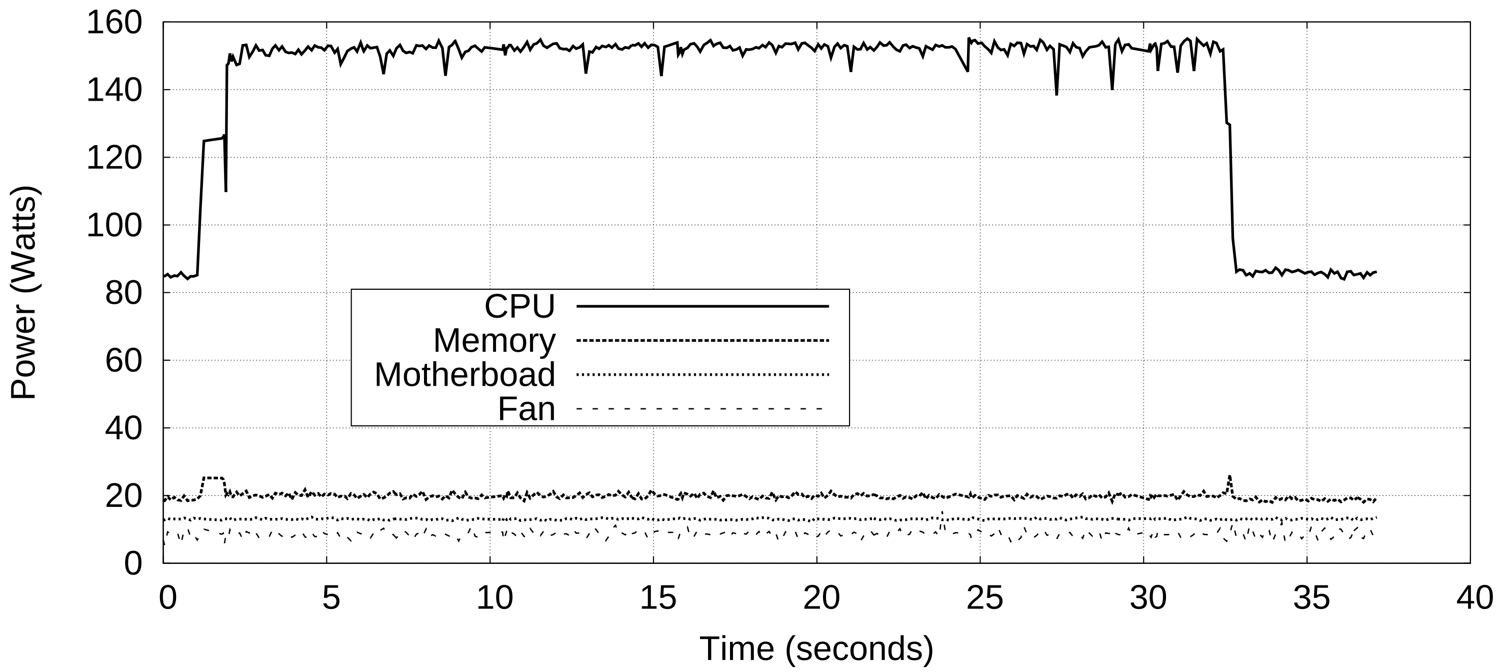
- Output voltage varies based on magnetic field.
- Current in wire causes magnetic field
- Voltage output is linear proportional to current
- Ideally little to no resistance (unlike sense resistor)
- Can measure higher current. 5, 20, 30A
- Need that? 100W CPU at 3.3V is roughly 30A

Other Issues

- Matching up internal and external measurements?
- Serial port? ntp? signal?
- Hard for small time intervals.

Existing Related Work

Plasma/dposv results with Virginia Tech's PowerPack



Powerpack

- Measure at Wall socket: WattsUp, ACPI-enabled power adapter, Data Acquisition System
- Measure all power pins to components (intercept ATX power connector?)
- CPU Power – CPU powered by four 12VDC pins.
- Disk power – measure 12 and 5VDC pins on disk power connecter

- Memory Power – DIMMs powered by four 5VDC pins
- Motherboard Power – 3.3V pins. Claim NIC contribution is minimal, checked by varying workload
- System fans

PowerMon 2

- PowerMon 2 is a custom board from RENCI
- Plugs in-line with ATX power supply.
- Reports results over USB
- 8 channels, 1kHz sample rate
- We have hardware; currently not working

Shortcomings of current methods

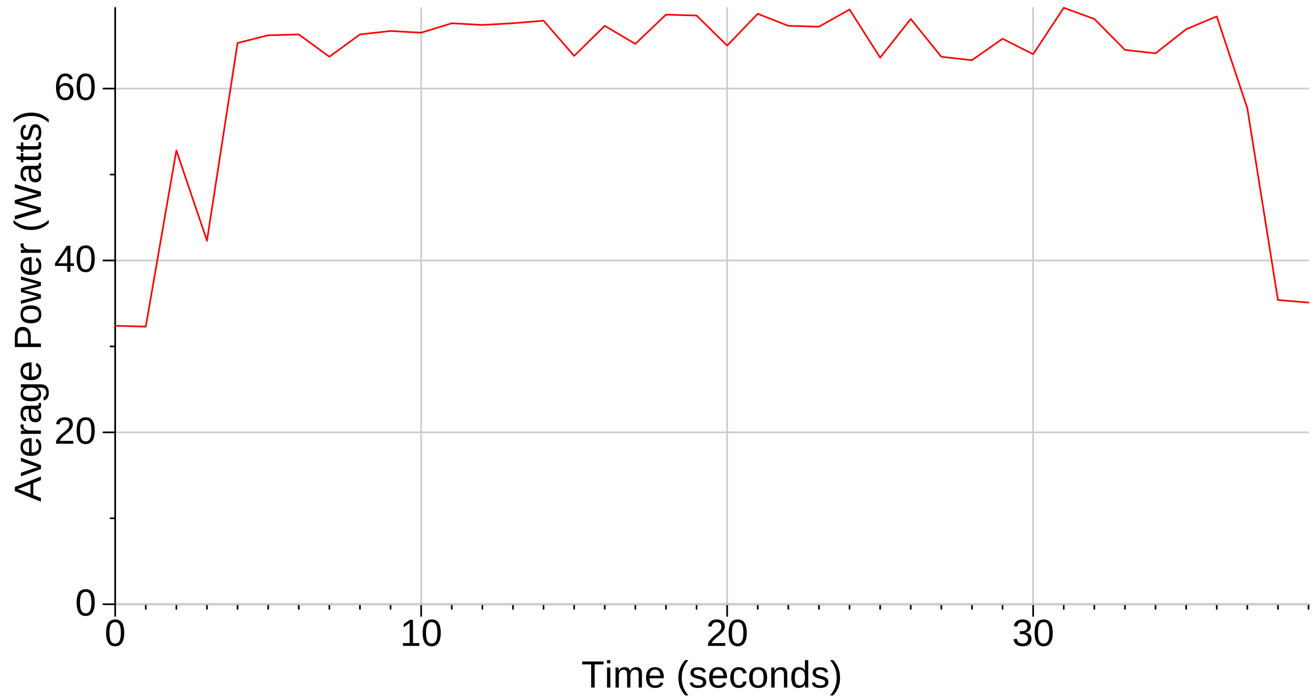
- Each measurement platform has a different interface
- Typically data can only be recorded off-line, to a separate logging machine, and analysis is done after the fact
- Correlating energy/power with other performance metrics can be difficult
- How often can you measure (a lot happens on a CPU at 2GHz)

Watt's Up Pro Meter

Watt's Up Pro Features

- Can measure 18 different values with 1 second resolution (Watts, Volts, Amps, Watt-hours, etc.)
- Values read over USB
- Joules can be derived from power and time
- Can only measure system-wide

Watt's Up Pro Graph



PLASMA Cholesky Factorization N=10,000 threads=2

Measured on Core2 Laptop

Estimating Power

- Popular thing to do. One example: *Real Time Power Estimation and Thread Scheduling via Performance Counters* by Singh, Bhadauria and McKee.
- Have some sort of hardware measurement setup.
- Then measure lots of easy-to-measure things. Performance counters. Temperature. etc.
- Create a model (machine learning?) that can estimate
- Apparently using as few as 4 counters can give pretty good results

RAPL

- **R**unning **A**verage **P**ower **L**imit
- Part of an infrastructure to allow setting custom per-package hardware enforced power limits
- User Accessible Energy/Power readings are a bonus feature of the interface

How RAPL Works

- RAPL is *not* an analog power meter (usually, Haswell-EP exception)
- RAPL uses a software power model, running on a helper controller on the main chip package
- Energy is estimated using various hardware performance counters, temperature, leakage models and I/O models
- The model is used for CPU throttling and turbo-boost, but the values are also exposed to users via a model-specific register (MSR)

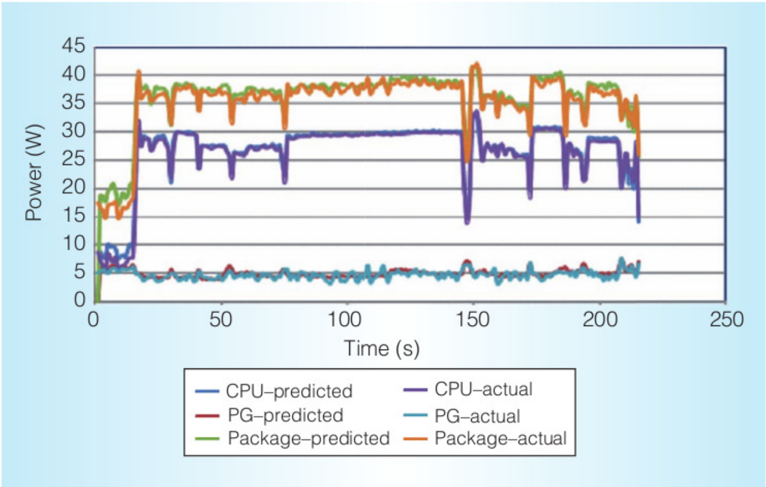
Available RAPL Readings

- `PACKAGE_ENERGY`: total energy used by entire package
- `PP0_ENERGY`: energy used by "power plane 0" which includes all cores and caches
- `PP1_ENERGY`: on original Sandybridge this includes the on-chip Intel GPU
- `DRAM_ENERGY`: on Sandybridge EP this measures DRAM energy usage. It is unclear whether this is just the interface or if it includes all power used by all the DIMMs too

- SoC energy (skylake and newer?)

RAPL Measurement Accuracy

- Intel Documentation indicates Energy readings are updated roughly every millisecond (1kHz)
- Rotem at al. show results match actual hardware



Rotem et al. (IEEE Micro, Mar/Apr 2012)

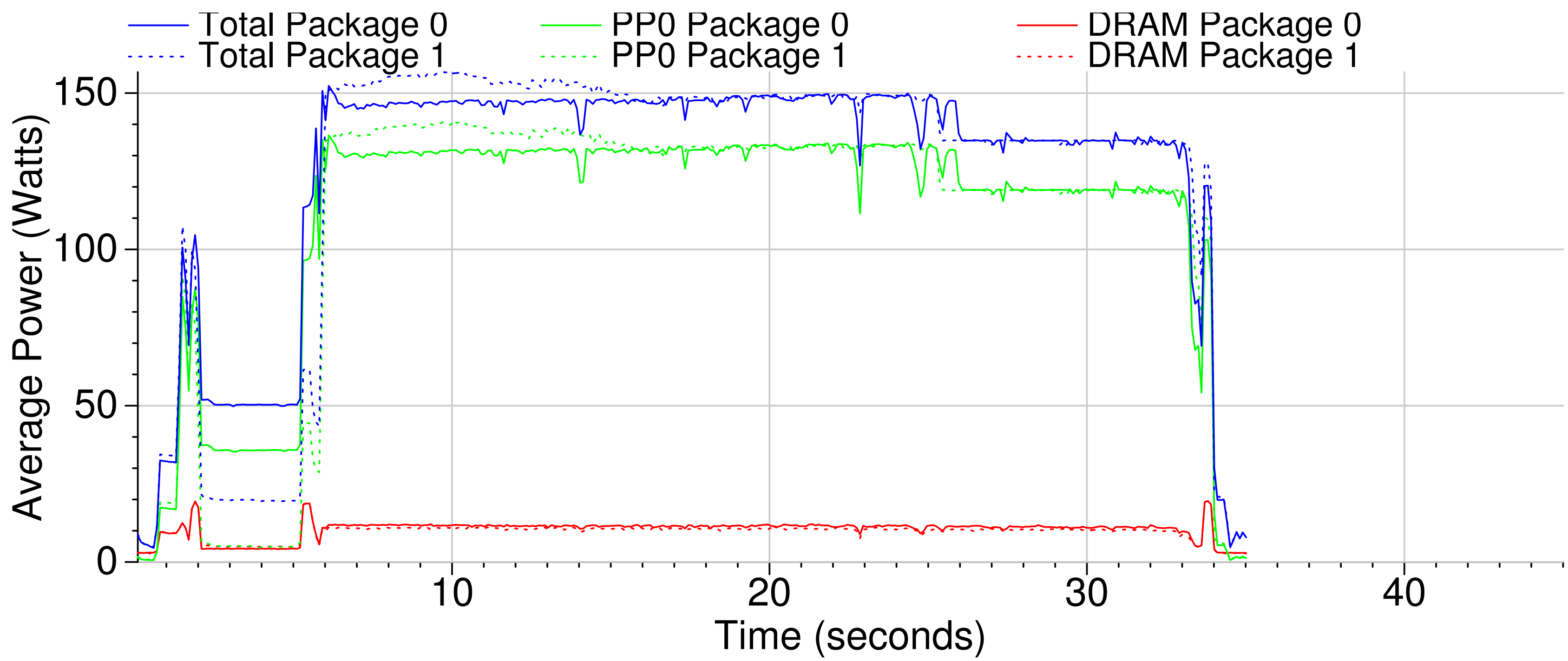
RAPL Accuracy, Continued

- The hardware also reports minimum measurement quanta. This can vary among processor releases. On our Sandybridge EP machine all Energy measurements are in multiples of 15.2nJ
- Power and Energy can vary between identical packages on a system, even when running identical workloads. It is unclear whether this is due to process variation during manufacturing or else a calibration issue.

RAPL Validation

- The Dresden Paper
- My MEMSYS paper (include some plots?)

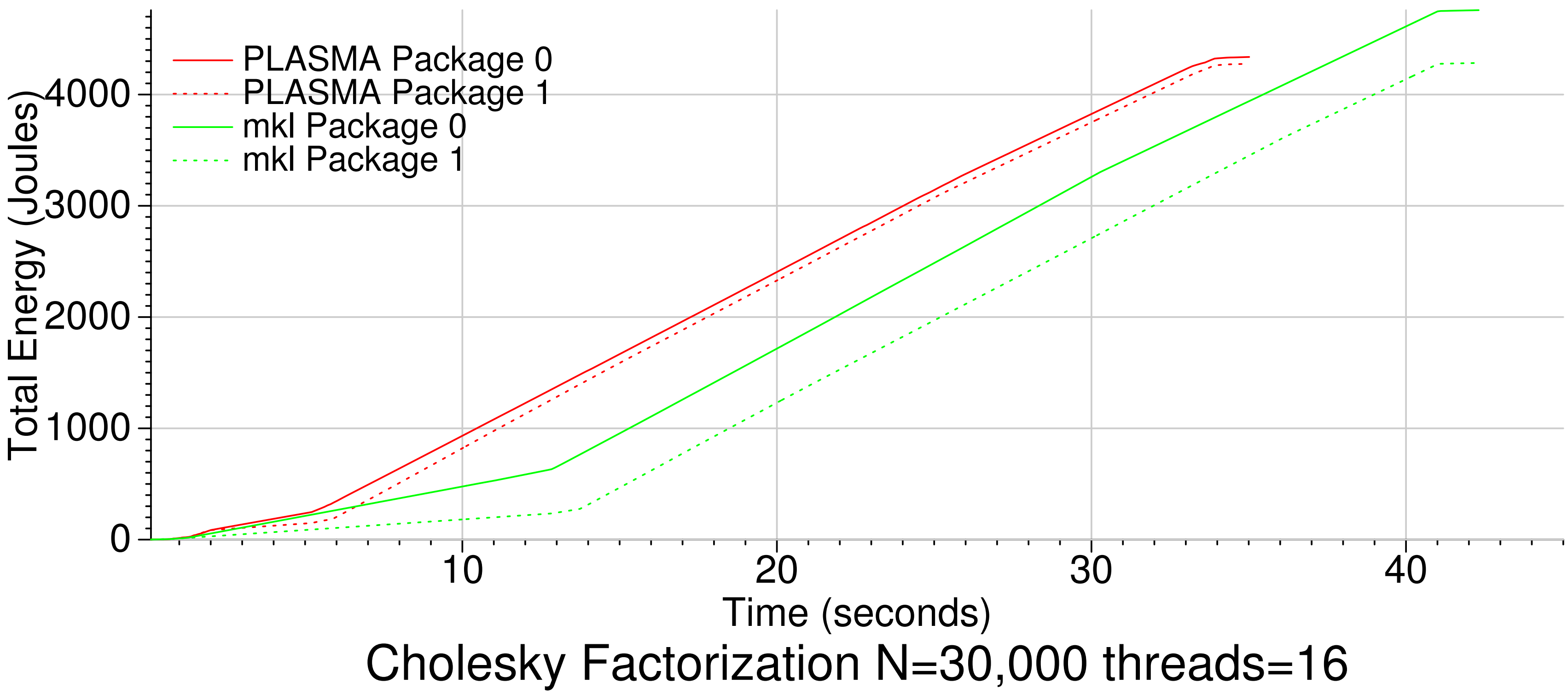
RAPL Power Plot



PLASMA Cholesky Factorization N=30,000 threads=16

Measured on SandyBridge EP

RAPL Energy Plot

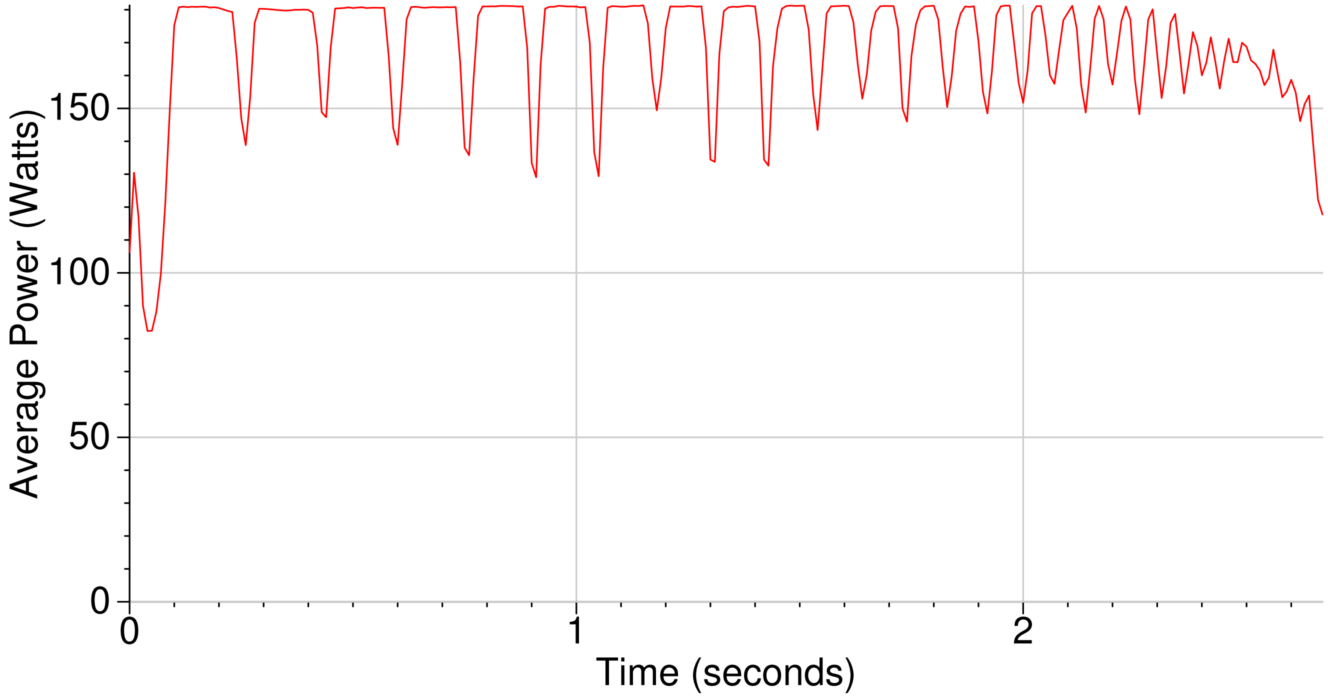


Measured on SandyBridge EP

NVML

- Recent NVIDIA GPUs support reading power via the NVIDIA Management Library (**NVML**)
- On Fermi C2075 GPUs it has milliwatt resolution within $\pm$5W and is updated at roughly 60Hz
- The power reported is that for the entire board, including GPU and memory

NVML Power Graph



MAGMA LU 10,000, Nvidia Fermi C2075

AMD Application Power Management

- Recent AMD Family 15h processors also can report "Current Power In Watts" via the Processor Power in the TDP MSR
- Support for this can be provided similar to RAPL
- Have had bad luck getting accurate readings. Have found various chip errata on fam15h and fam16h hardware

Other ways to measure Power

- IPMI – many server machines have built in (low frequency) measurement of power supply values.
- Thermal? IR camera? Can see how much individual parts of chip use.
  Overheat? Use IR transparent liquid to cool it?

Using RAPL

- On Linux, at least 3 ways to get these values
- Read msr directly, either with instruction or /dev/msr. Need root as you can do bad things with msrs. "safemsr"
- perf_event
- hwmon/powercap (/sys/class/powercap/)

Listing Events

```
$ perf list
...
  power/energy-cores/                                              [Kernel P
  power/energy-gpu/                                                [Kernel P
  power/energy-pkg/                                                [Kernel P
  power/energy-ram/                                                [Kernel P
...
```

Measuring

```
$ perf stat -a -e power/energy-cores/,power/energy-ram/,instru

 Performance counter stats for 'system wide':

              63.79 Joules power/energy-cores/
               2.34 Joules power/energy-ram/
        21038123875        instructions              #    1.06
        19782762541        cycles

        3.407427702 seconds time elapsed
```

Measuring

- The key is -a which enables system-wide mode (needs root too if not configured as such)
- Why do you need system-wide?
- What does that do to the other metrics?

Power and Energy Concerns

Table 1: OpenBLAS HPL N=10000 (Matrix Multiply)

Machine	Processor	Cores	Freq	Idle Power	Load Power	Time Time	Total Energy
Raspberry Pi 2	Cortex-A7	4	900MHz	1.8W	3.4W	454s	1543J
Dragonboard	Cortex-A53	4	1.2GHz	2.4W	4.7W	241s	1133J
Raspberry Pi 3	Cortex-A53	4	1.2GHz	1.8W	4.3W	178s	765J
Jetson-TX1	Cortex-A57	4	1.9GHz	2.1W	13.4W	47s	629J
Macbook Air	Broadwell	2	1.6GHz	10.0W	29.1W	14s	407J

1. Which machine has the lowest under-load power draw?
   Pi 2

2. Which machine consumes the least amount of energy?
   Broadwell Macbook Air

3. Which machine computes the result fastest?
   Broadwell Macbook Air

4. Consider a use case with an embedded board taking a picture once every 60 seconds and then performing a matrix-multiply similar to the one in the benchmark (perhaps for image-recognition purposes). Could all of the boards listed meet this deadline?
   No, only the Jetson and Macbook Air can meet the

deadline

5. Assume a workload where a device takes a picture once a minute then does a large matrix multiply (as seen in Table 1). The device is idle when not multiplying, but under full load when it is.

(a) Over a mine, what is the total energy usage of the Jetson TX-1?

Each Minute = (13s Idle * 2.1W) + (47s Load *13.4W) = 657J

(b) Over a minute, what is the total energy usage of the Macbook Air?

Each Minute $= (46s * 10W) + (14*29.1) = 867J$

Pandaboard Power Stats

- Wattsuppro: 2.7W idle, seen up to 5W when busy
- `http://ssvb.github.com/2012/04/10/cpuburn-arm-cortex-a9.html`
- With Neon and CPU burn:

Idle system	550 mA	2.75W
cpuburn-neon	1130 mA	5.65W
cpuburn-1.4a (burnCortexA9.s)	1180 mA	5.90W
ssvb-cpuburn-a9.S	1640 mA	8.2W

Easy ways to reduce Power Usage

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.

When can we scale CPU down?

- System idle
- System memory or I/O bound
- Poor multi-threaded code (spinning in spin locks)
- Thermal emergency
- User preference (want fans to run less)