CS 152 Computer Architecture and Engineering

Lecture 20: Power and Energy

John Wawrzynek
EECS, University of California at Berkeley
http://inst.eecs.berkeley.edu/~cs152

Thanks to John Lazzaro for the slides.
Announcements

- PS 5 now due in section (extra day)
- Quiz 5 covering up to but not including power/energy
- Lab 5 out this weekend. Due Dec 2 (day after final class).
- Remember – No Final exam!
Today: Power and Energy

Metrics: Power and energy (intro)

Metrics: Power and energy (technique)
Universal: Power and energy units are comparable across all of applied physics.

Power and Energy

So, we use automobiles to introduce terminology.
The Watt: Unit of power. A rate of energy (J/s). A gas pump hose delivers 6 MW.


The Watt: Unit of power. A rate of energy (J/s). A gas pump hose delivers 6 MW.

1 J = 1 W s. 1 W = 1 J/s.

120 KW: The power delivered by a Tesla Supercharger. Tesla Model S has a 306 MJ battery (good for 265 miles).
Sad fact: Computers turn electrical energy into heat. Computation is a byproduct.

Energy and Performance

Air or water carries heat away, otherwise chip melts.
The Joule: Unit of energy. Can also be expressed as Watt-Seconds. Burning 1 Watt for 100 seconds uses 100 Watt-Seconds of energy.

This is how electric tea pots work ...

1 Joule heats 1 gram of water 0.24 degree C

1 Joule of Heat Energy per Second

The Watt: Unit of power. The amount of energy burned in the resistor in 1 second.

20 W rating: Maximum power the package is able to transfer to the air. Exceed rating and resistor burns.
Cooling an iPod nano ...

Like resistor on last slide, iPod relies on passive transfer of heat from case to the air.

Why? Users don’t want fans in their pocket ...

To stay “cool to the touch” via passive cooling, power budget of 5 W.

If iPod nano used 5W all the time, its battery would last 15 minutes ...
Powering an iPod nano (2005 edition)

1.2 W-hour battery: Can supply 1.2 watts of power for 1 hour.

\[
1.2 \text{ W-hr} / 5 \text{ W} \approx 15 \text{ minutes.}
\]

More W-hours require bigger battery and thus bigger “form factor” -- it wouldn’t be “nano” anymore :-).

Real specs for iPod nano:

- 14 hours for music,
- 4 hours for slide shows.

- 85 mW for music.
- 300 mW for slides.
Finding the (2005) iPod nano CPU ...

A close relative ...

Two 80 MHz CPUs. One CPU used for audio, one for slides.

Low-power ARM roughly 1mW per MHz ...

... variable clock, sleep modes.

85 mW system power realistic
The CPU is only part of power budget!

“Amrdahl’s Law for Power”

If our CPU took no power at all to run, that would only double battery life!

2004-era notebook running a full workload.

Data courtesy Mahesri et al., U of Illinois, 2004
What's happened since 2005?

2010 nano
0.74 ounces
(50% of 2005 Nano)

"Up to" 24 hours audio playback.
70% improvement from 2005 nano.

0.39 W Hr
(33% of 2005 Nano)
Processors and Energy
Moore’s Law

The curve shows transistor count doubling every two years.

Transistor count

2.6 Billion

1 Million

2 Thousand

Main driver: device scaling ...

If we scale the gate length by a factor $\kappa$, how should we scale other aspects of transistor to get the “best” results?
Dennard Scaling

Things we do: scale dimensions, doping, Vdd.

What we get: $\kappa^2$ as many transistors at the same power density!

Whose gates switch $\kappa$ times faster!

Power density scaling ended in 2003 (Pentium 4: 3.2GHz, 82W, 55M FETs). Why? We could no longer scale Vdd.
Why? We can no longer fully scale $V_{dd}$ ... because MOS transistor leakage current is no longer a negligible part of the power budget.

---

**Figure 1.11 Growth in clock rate of microprocessors in Figure 1.1.** Between 1978 and 1986, the clock rate improved less than 15% per year while performance improved by 25% per year. During the "renaissance period" of 52% performance improvement per year between 1986 and 2003, clock rates shot up almost 40% per year. Since then, the clock rate has been nearly flat, growing at less than 1% per year, while single processor performance improved at less than 22% per year.
Switching Energy: Fundamental Physics

Every logic transition dissipates energy.

How can we (1) Reduce # of clock transitions. But we have work to do ...

  limit (2) Reduce Vdd. But lowering Vdd limits the clock speed ...

switching (3) Fewer circuits. But more transistors can do more work.

to energy? (4) Reduce C per node. One reason why we scale processes.
Scaling switching energy per gate ...

IC process scaling ("Moore's Law")

Due to reducing V and C (length and width of Cs decrease, but plate distance gets smaller).

Recent slope more shallow because V is being scaled less aggressively.

Second Factor: Leakage Currents

Even when a logic gate isn’t switching, it burns power.

\[ 0V = V_{IN} \]

\[ V_{OUT} \]

\[ I_{Gate} \]

\[ I_{Sub} \]

\[ C_L \]

**Igate:** Ideal capacitors have zero DC current. But modern transistor gates are a few atoms thick, and are not ideal.

**Isub:** Even when this nFet is off, it passes an \( I_{off} \) leakage current.

We can engineer any \( I_{off} \) we like, but a lower \( I_{off} \) also results in a lower \( I_{on} \), and thus a lower maximum clock speed.

Intel’s 2006 processor designs, leakage vs switching power

A lot of work was done to get a ratio this good ... 50/50 is common.

Bill Holt, Intel, Hot Chips 17
We can increase $I_{on}$ by raising $V_{dd}$ and/or lowering $V_t$.

$I_{ds} = 1.2 \text{ mA} = I_{on}$

$0.7 = V_{dd}$

$0.25 = V_t$

$I_{off} = 0$ ???

$0.0012$
Plot on a “Log” Scale to See “Off” Current

We can decrease $I_{\text{off}}$ by raising $V_t$ - but that lowers $I_{\text{on}}$. 

$V_{dd} = 0.7$  
$1.2 \text{ mA} = I_{\text{on}}$  
$0.25 \approx V_t$  
$I_{\text{off}} \approx 10 \text{ nA}$
Device engineers trade speed and power

We can reduce $CV^2(P_{\text{active}})$ by lowering $V_{dd}$.

We can increase speed by raising $V_{dd}$ and lowering $V_t$.

We can reduce leakage ($P_{\text{standby}}$) by raising $V_t$.

The design window rapidly diminishes as technology is scaled down.

From: Silicon Device Scaling to the Sub–10–nm Regime
Meikei Ieong,1* Bruce Doris,2 Jakub Kedzierski,1 Ken Rim,1 Min Yang1

CS 152: L19: Power and Energy
Customize processes for product types ...

Transistor channel is a raised fin. Gate controls channel from sides and top. Channel depth is fin width. 12-15nm for L=22nm.
Clock rates have flattened out, but ...

Figure 1.11 Growth in clock rate of microprocessors in Figure 1.1. Between 1978 and 1986, the clock rate improved less than 15% per year while performance improved by 25% per year. During the “renaissance period” of 52% performance improvement per year between 1986 and 2003, clock rates shot up almost 40% per year. Since then, the clock rate has been nearly flat, growing at less than 1% per year, while single processor performance improved at less than 22% per year.
Performance: Put more transistors to work

Figure 1.1 Growth in processor performance since the late 1970s. This chart plots performance relative to the VAX 11/780 as measured by the SPEC benchmarks (see Section 1.8). Prior to the mid-1980s, processor performance growth was largely technology driven and averaged about 25% per year. The increase in growth to about 52% since then is attributable to more advanced architectural and organizational ideas. By 2003, this growth led to a difference in performance of about a factor of 25 versus if we had continued at the 25% rate. Performance for floating-point-oriented calculations has increased even faster. Since 2003, the limits of power and available instruction-level parallelism have slowed uniprocessor performance, to no more than 22% per year, or about 5 times slower than had we continued at 52% per year. (The fastest SPEC performance since 2007 has had automatic parallelization turned on with increasing number of cores per chip each year, so uniprocessor speed is harder to gauge. These results are limited to single-socket systems to reduce the impact of automatic parallelization.) Figure 1.11 on page 24 shows the improvement in clock rates for these same three eras. Since SPEC has changed over the years, performance of newer machines is estimated by a scaling factor that relates the performance for two different versions of SPEC (e.g., SPEC89, SPEC92, SPEC95, SPEC2000, and SPEC2006).
Moore’s Law

We still scale to get more transistors per unit area … but we use design techniques to reduce power.
Takeaway Abstractions

From Part I ...
Small circuits can go very fast in standard CMOS ...

This oscillator runs at 210 GHz in a 32 nm SOI CMOS logic process, and consumes 42 mW ...

But if we used these techniques for a CPU, our 150 W air-cooled power limit would limit our design to using about ten thousand transistors ...

... but we are used to using 100s of millions!
Dennard scaling stopped working in 2004. Why? MOSFET off currents became non-negligible. We limit clock speed to prevent chip from melting.

Figure 1.11 Growth in clock rate of microprocessors in Figure 1.1. Between 1978 and 1986, the clock rate improved less than 15% per year while performance improved by 25% per year. During the “renaissance period” of 52% performance improvement per year between 1986 and 2003, clock rates shot up almost 40% per year. Since then, the clock rate has been nearly flat, growing at less than 1% per year, while single processor performance improved at less than 22% per year.
Dynamic Power: 4 ways to reduce it ...

- Every logic transition dissipates energy.

How can we limit switching energy?

1. Reduce # of clock transitions. But we have work to do ...
2. Reduce Vdd. But lowering Vdd limits the clock speed ...
3. Fewer circuits. But more transistors can do more work.
4. Reduce C per node. One reason why we scale processes.

Strong result: Independent of technology.
Static power: We trade off speed for power.

Even when a logic gate isn’t switching, it burns power.

\[ 0V = V_{IN} \]

**I\_\text{Gate}:** Ideal capacitors have zero DC current. But modern transistor gates are a few atoms thick, and are not ideal.

**I\_\text{Sub}:** Even when this nFet is off, it passes an \( I_{\text{off}} \) leakage current.

We can engineer any \( I_{\text{off}} \) we like, but a lower \( I_{\text{off}} \) also results in a lower \( I_{\text{on}} \), and thus a lower maximum clock speed.

Intel’s 2006 processor designs, leakage vs switching power

A lot of work was done to get a ratio this good ... 50/50 is common.

Bill Holt, Intel, Hot Chips 17
Factor of 60 in leakage vs. 3.5 in speed

Chart shows 9 different NAND gates for an IC process, each with a different speed vs. static power tradeoff.

(40, 45, 50) are transistor channel lengths (in nm)
Delay and Power versus VDD

Dynamic Power (and leakage) can be traded by delay
Announcements

° Lab 5 due Friday

° Quiz 5 grades out Thursday

° Thursday – Final class meeting
  • Course wrap up / future trends
  • Time reserved for online course survey

° Friday – final discussion section
  • wrap up and announce the results of the competition
Five low-power design techniques

- Parallelism and pipelining
- Power-down idle transistors
- Slow down non-critical paths
- Clock gating
- Thermal management
Trading Hardware for Power

via Parallelism and Pipelining ...
Gate delay roughly linear with \( V_{dd} \)

And so, we can transform this:

Block processes stereo audio. 1/2 of clocks for “left”, 1/2 for “right”.

Into this:

Top block processes “left”, bottom “right”.

CV\(^2\) power only

This magic trick brought to you by Cory Hall ...
Multiple Cores for Low Power

Trade hardware for power, on a large scale ...
Cell:
The PS3 chip

2006
Cell (PS3 Chip): 1 CPU + 8 “SPUs”

- **L2 Cache**: 512 KB
- **8 Synergistic Processing Units (SPUs)**
- **PowerPC**
One Synergistic Processing Unit (SPU)

SPU issues 2 inst/cycle (in order) to 7 execution units
256 KB Local Store, 128 128-bit Registers
SPU fills Local Store using DMA to DRAM and network
A “Schmoo” plot for a Cell SPU ...

The lower Vdd, the less dynamic energy consumption.

\[ E_{0\rightarrow 1} = \frac{1}{2} CV_{dd}^2 \]

\[ E_{1\rightarrow 0} = \frac{1}{2} CV_{dd}^2 \]

The lower Vdd, the longer the maximum clock period, the slower the clock frequency.
Clock speed alone doesn’t help E/op...

But, lowering clock frequency while keeping voltage constant spreads the same amount of work over a longer time, so chip stays cooler...

\[ E_{0\rightarrow1} = \frac{1}{2} CV_{dd}^2 \quad E_{1\rightarrow0} = \frac{1}{2} CV_{dd}^2 \]
Scaling $V$ and $f$ does lower energy/op

1 W to get 2.2 GHz performance. 26 C die temp.

7 W to reliably get 4.4 GHz performance. 47 C die temp.

If a program that needs a 4.4 Ghz CPU can be recoded to use two 2.2 Ghz CPUs ... big win.
How iPod nano 2005 puts its 2 cores to use ...

Two 80 MHz CPUs. Was used in several nano generations, with one CPU doing audio decoding, the other doing photos, etc.
2013 Macbook Air

Voltage range: 0.655V to 1.041V ... 2.5x in $CV^2$ energy

<table>
<thead>
<tr>
<th>CPU Type</th>
<th>Idle</th>
<th>Cinebench 11.5 (1 thread)</th>
<th>Cinebench 11.5 (4 threads)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Core i5-4250U</td>
<td>0.665V</td>
<td>0.852V - 0.904V (2.3GHz - 2.6GHz*)</td>
<td>0.842V (2.3GHz)</td>
</tr>
<tr>
<td></td>
<td>(800MHz)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intel Core i7-4650U</td>
<td>0.655V</td>
<td>0.949V - 1.041V (2.9GHz - 3.3GHz*)</td>
<td>0.786V - 0.949V (2.8GHz - 2.9GHz*)</td>
</tr>
<tr>
<td></td>
<td>(800MHz)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Haswell CPU/GPU
Powering down idle circuits
Add “sleep” transistors to logic ...

Example: Floating point unit logic.

When running fixed-point instructions, put logic “to sleep”.

+++ When “asleep”, leakage power is dramatically reduced.

--- Presence of sleep transistors slows down the clock rate when the logic block is in use.
Intel example: Sleeping cache blocks

A tiny current supplied in “sleep” maintains SRAM state.

Intel Medfield

- LPDDR2
- eMMC
- SD/MMC
- Primary Camera: 8MP, 15fps, 1080p
- Secondary Camera: 1.3MP, 1080p

Central Processing Unit (CPU) with 512KB L2 cache

- Security Engine
- Power Manager
- Low Power Audio

- 2D/3D Graphics
- Video Encode/Decode (1080p30)
- Image Signal Processor
- Display Controller (3 pipes)

- Power Delivery IC: VRs Audio CODEC USB2 OTG
- Rails I2S ULPI
- MIPI-HSI
- IMC 6260 HSPA+ Modem
- TI WiFi & BT
- CSR GPS
- HDMI Display
- Internal Display
- MIPI-DSI
- SDIO
- UART
Intel Medfield Switches 45 power “islands.”

Fine-grained control of leakage power, to track user activity.

“Race to idle” strategy -- finish tasks quickly, to get to power down.
Playing a game ...
Watching a video ...

CPU is now off!
Looking at phone screen, not doing anything ...
Phone in your pocket, waiting for a call ...

Standby State – just waiting for wakes
Slow down “slack paths”
Fact: Most logic on a chip is “too fast”

The critical path

Most wires have hundreds of picoseconds to spare.

Use several supply voltages on a chip ...

Why use multi-Vdd? We can reduce dynamic power by using low-power Vdd for logic off the critical path.

What if we can’t do a multi-Vdd design? In a multi-Vt process, we can reduce leakage power on the slow logic by using high-Vth transistors.

Logical partition into 0.8V and 1.0V nets done manually to meet 350 MHz spec (90nm).

Level-shifter insertion and placement done automatically.

Dynamic power in 0.8V section cut 50% below baseline.

Leakage power in 1.0V section cut 70% below baseline.

From a chapter from new book on ASIC design by Chinnery and Keutzer (UCB).
Gating clocks to save power
On a CPU, where does the power go?

Half of the power goes to latches (Flip-Flops).

Most of the time, the latches don’t change state.

So gated clocks are a big win.
Done with CAD tools in a disciplined way.

From: Bose, Martonosi, Brooks: Sigmetrics-2001 Tutorial
Synopsis Design Compiler can do this ...

“Up to 70% power savings at the block level, for applicable circuits”
Synopsis Data Sheet
Power Compiler also can do this ...

10-20% push-button power savings, using techniques like this one.
Thermal Management
Keep chip cool to minimize leakage power

A recipe for thermal runaway

![Graph showing I_{CCINTQ} vs. Junction Temperature with increase relative to 25°C](image)

Figure 3: $I_{CCINTQ}$ vs. Junction Temperature with Increase Relative to 25°C