

ECE 571 – Advanced Microprocessor-Based Design Lecture 6

Vince Weaver

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

vincent.weaver@maine.edu

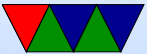
7 February 2017

Announcements

- HW#1 and HW#2 grades out
- HW#3 was posted
- Note, the equake benchmark takes a while to run (a few minutes). Don't give up on it.



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.

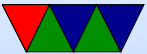


Modern results are from CUGR/REU student research.

	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	

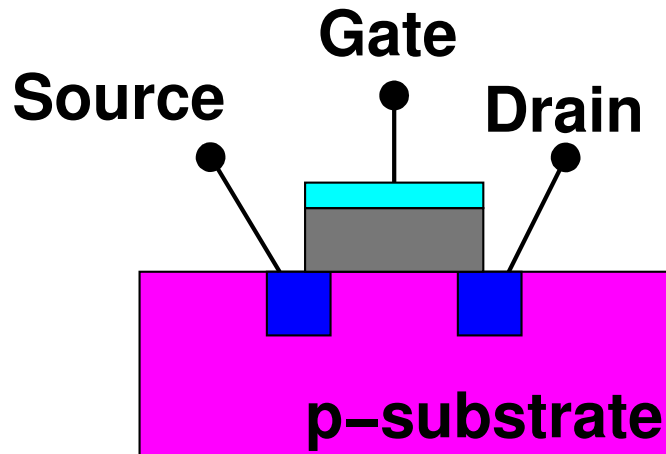


CPU Power and Energy

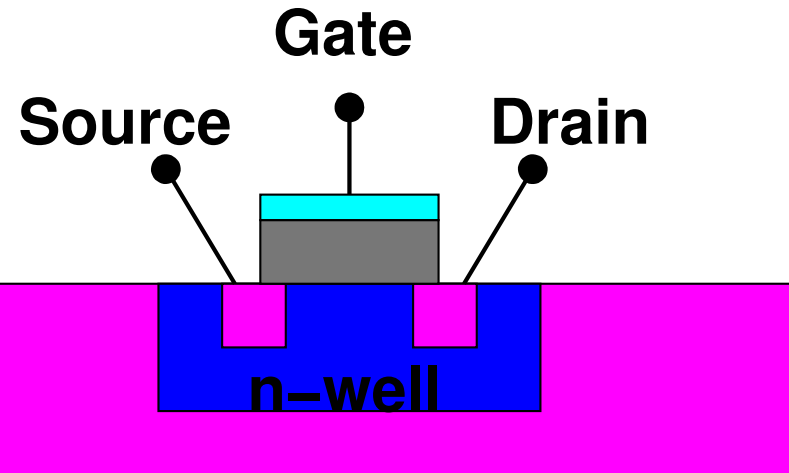


CMOS Transistors

N-MOSFET



P-MOSFET



CMOS Dynamic Power

- $P = C\Delta VV_{dd}\alpha f$

Charging and discharging capacitors big factor
($C\Delta VV_{dd}$) from V_{dd} to ground

α is activity factor, transitions per clock cycle

F is frequency

- α often approximated as $\frac{1}{2}$, ΔVV_{dd} as V_{dd}^2 leading to
 $P \approx \frac{1}{2}CV_{dd}^2f$

- 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



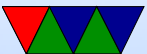
CMOS Static Power

- 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 SiO_2 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?



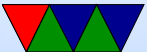
Total Energy

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



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

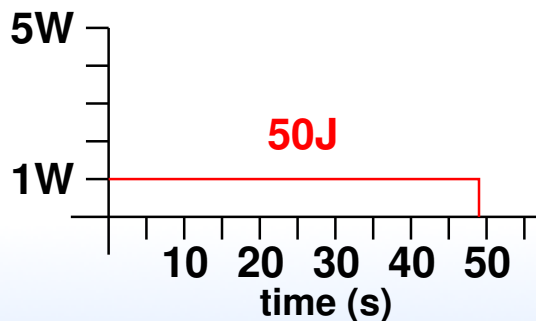
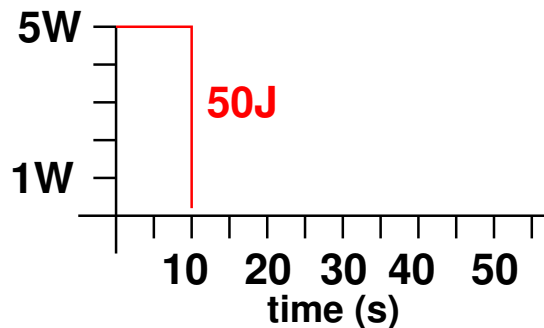
- Does not take into account time. Lowering power does no good if it increases runtime.



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

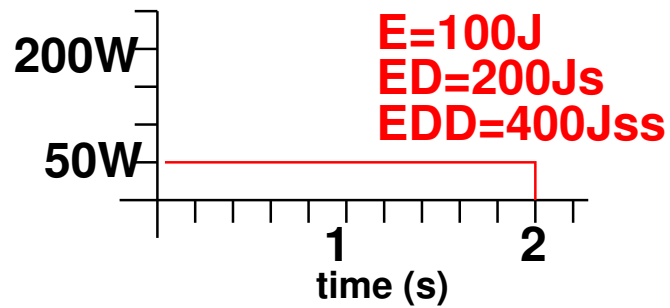
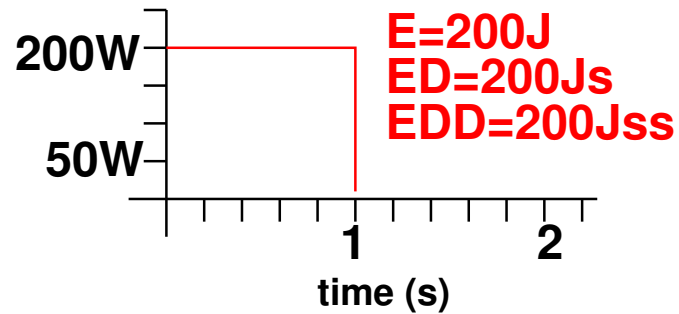


- Technology Scaling – reduces V and power.
- Transition Reduction – better logic design, have fewer transitions
Get rid of clocks? Asynchronous? Clock-gating?



ED Optimization

Which is better?



Energy Delay Squared– $E*t*t$

- Martin, Nyström, Péntzes – Power Aware Computing, 2002
- Independent of Voltage in CMOS
- $E*t$ can be misleading
 $E_a=2E_b$, $t_a=t_b/2$
Reduce voltage by half, $E_a=E_a/4$, $t_a=2t_a$, $E_a=E_b/2$,
 $t_a=t_b$
- 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.

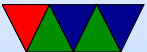
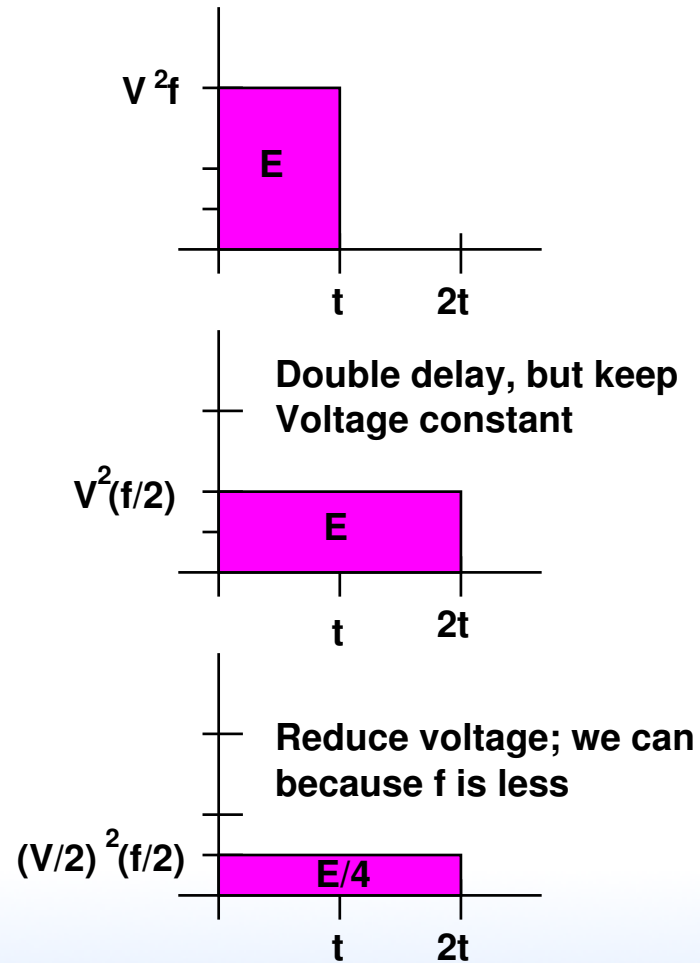
Energy	Delay	ED	ED^2
5J	2s	$10Js$	$20Js^2$
5J	3s	$15Js$	$45Js^2$

Same ED , Different ED^2

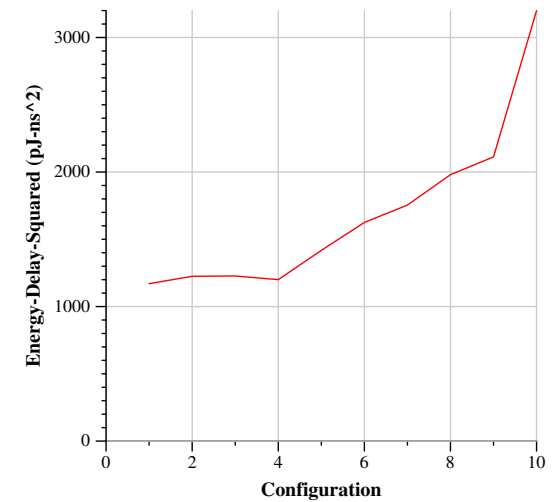
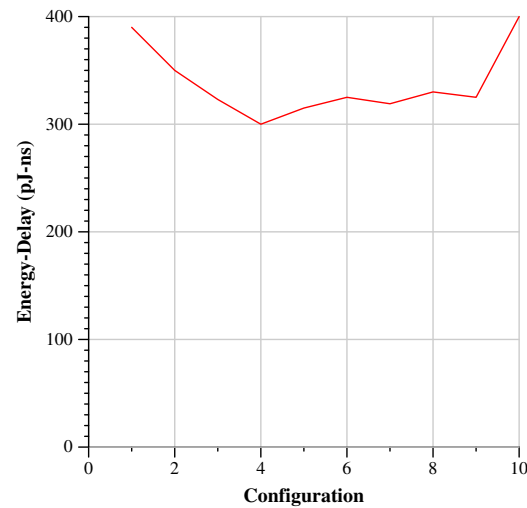
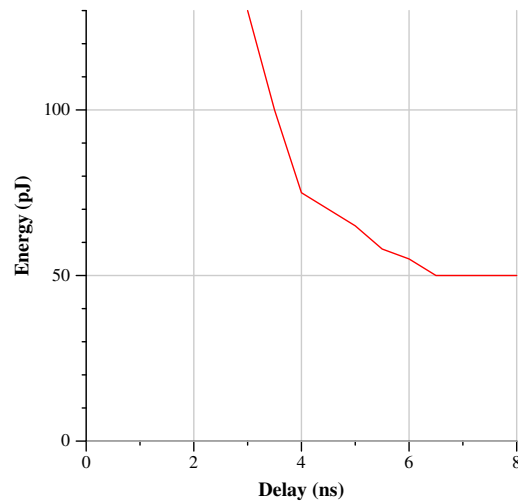
Energy	Delay	ED	ED^2
5J	2s	$10Js$	$20Js^2$
2J	5s	$10Js$	$50Js^2$



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?)

