ECE 471 – Embedded Systems Lecture 31

Vince Weaver

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

vincent.weaver@maine.edu

25 November 2024

Announcements

- HW#9 was due
- Project updates due, will respond to them
- HW#10 will be posted (it will be short)
- Return your parts when you are done with them
- Have a great Thanksgiving break!

Definitions

People often say Power when they mean Energy

- Power is instantaneous

- Energy is total Power used over time

Units

- 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 AC) – same units as Watts, but not quite the same thing
- Charge – mAh (batteries) – need voltage to convert to Energy

Aside on Batteries and Charging

- Small embedded systems might run off of a battery
- Often rechargeable: In old days lead-acid, these days LiOn (Lithium Ion) or Li-Po (Lithium Polymer)
- Battery management might involve own embedded system
  - if discharge too far can damage
  - of over-charge or puncture can explode or catch on fire
  - This is why limits on battery devices in airplanes and shipments

- Battery Charging
  - Constant current fast-charge
  - Cell balancing (if multiple cells)
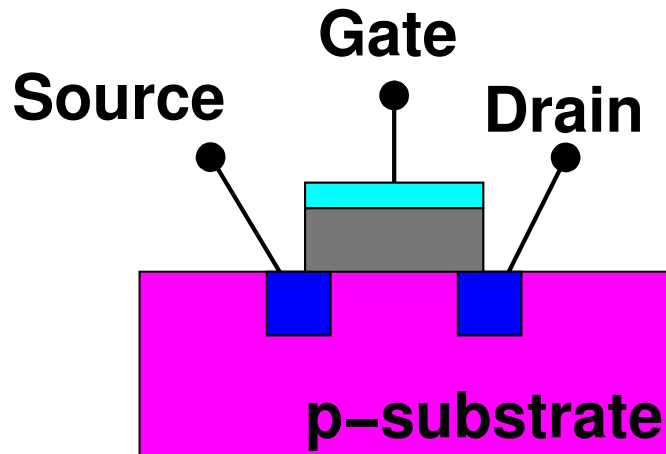 - Constant voltage trickle-charging when near full
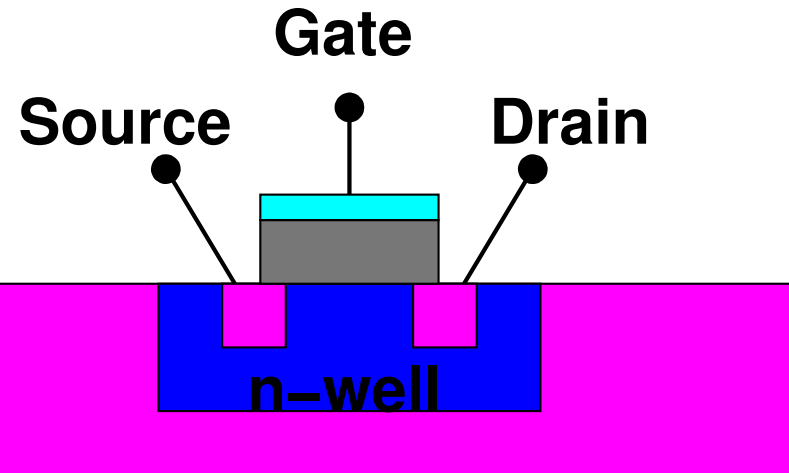
CPU Power and Energy
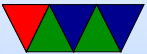
CMOS Transistors

N–MOSFET

P–MOSFET

Gate

Source

Drain

Gate

Source

Drain

p–substrate

n–well

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 $P \approx \frac{1}{2}CV_{dd}^2 f$

- 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 (threshold)

- 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 critical transistors fast; non-critical can be made slower and less leakage prone

- Body-biasing

- Sleep transistors

Total Energy

- $E_{tot} = [P_{dyanmic} + P_{static}]t$

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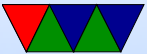
Delay

- $T_d = \frac{C_L V_{dd}}{\mu C_{ox}(\frac{W}{L})(V_{dd} - V_t)}$

- 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

- Important for embedded

14

Embedded Concerns

- Gaming desktop, 750W power supply?
- Laptop, maybe 60W power supply?
- Raspberry Pi4, 15W (older ones were fine with 5W)
- STM32L476G Discovery (old ECE271) – 300uA (1.5mW or so?) when everything configured
- STM32L1 claim 170nA in low-power sleep with SRAM retention 1.2uA when running?
- Compare with Intel 150W CPU at $1.5V = 100A$?

Power and Energy Concerns

Table 1: ATLAS 300x300 DGEMM (Matrix Multiply)

Machine	Processor	Cores	Frequency	Idle	Load	Time	Total Energy
Raspberry Pi	ARM 1176	1	700MHz	3.0W	3.3W	23.5s	77.6J
Gumstix Overo	Cortex-A8	1	600Mhz	2.6W	2.9W	27.0s	78.3J
Beagleboard	Cortex-A8	1	800MHz	3.6W	4.5W	19.9s	89.5J
Pandaboard	Cortex-A9	2	900MHz	3.2W	4.2W	1.52s	6.38J
Chromebook	Cortex-A15	2	1.7GHz	5.4W	8.1W	1.39s	11.3J

Questions

- Which machine consumes the least amount of energy? (Pandaboard)

- Which machine computes the result fastest? (Chromebook)

- Chromebook is a laptop so also includes display and wi-fi

- Consider a use case with an embedded board taking a picture once every 20 seconds and then performing a

300x300 matrix multiply transform on it. Could all of the boards listed meet this deadline?

No, the Raspberry Pi and Gumstix Overo both take longer than 20s and the Beagleboard is dangerously close.

- Assume a workload where a device takes a picture once a minute then does a 300x300 matrix multiply (as seen in Table 1). The device is idle when not multiplying, but under full load when it is. Over an hour, what is the energy usage of the Chromebook? What is the energy usage of the Gumstix?

Chromebook per minute: $(1.39s \times 8.1W) + (58.61s \times 5.4W) = 327.75J$

Chromebook per hour: 327.75J * 60 = 19.7kJ

Gumstix per minute: $(27s \times 2.9W) + (33s \times 2.6W) = 164.1J$

Gumstix per hour: 164.1J * 60 = 9.8kJ

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)

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?

RAPL – Running Average Power Limit

- Estimated power provided by the CPU
- Small embedded system calculates it based on model
- Provided for turbo-boost and power capping

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
- Example:
  - App1 uses 5W for 2s, how much Energy? (10J)
  - App2 uses 0.1W for 90s, how much Energy? (9J)
  - 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?
- Example with inverse ED (higher better):

Alpha 21064	SPEC=155	Power=30W	SPEC*SPEC/W=800
PPC603	SPEC=80	Power=3W	SPEC*SPEC/W=2100

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