ECE 471 – Embedded Systems
Lecture 31

Vince Weaver

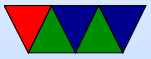
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

30 November 2018

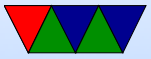
Announcements

- HW#10 was due
- Project update was due
- HW#11 will be posted



HW#9

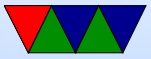
- C code review
- Do note, it's an LED display not LCD
- Error checking. 0 points if segfaults. Also if prints a wrong value to display.
- Leaking file descriptors. Close (or keep open) instead of just re-opening
- 1-wire, printing based on ASCII not floating point. Could you even test this? Made it a pain to grade.
- How do you convert from float to decimal?



- Lots of people miss 0 due to gt/lt
- 45.9 print as 45.8?

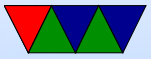
  Floating point math is a pain! What do you get if you do `int result=10*(45.9-45);`? 9.000000? Print more digits 8.999999999999857891452847979963 for fp to int conversion just drops the floating point part, doesn't round

- Following a spec?
  - Corner cases
  - Spec says degree symbol, not F or C
  - Single-digit temps (unclear spec) Leading zeroes. Spec



says 02.0 not 2.0 or 2.00

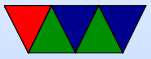
- is Zero negative?
- Rounding
- Do you need a . after a three digit temp?
- Left/right justified for single digit
- Reporting error! Must be sure display not printing invalid info! (door on walk-in oven. If it goes from 70F to 1000F (off scale) between readings, don't want it to stay at 70F, you want ERR or HOT or some way to notify something is wrong) More realistically, probe wire broke, should it just report last reading? Or



maybe go blank?

- What to do if temperature is -99.4 degrees?

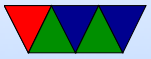
- Check inputs! Recent problem with europe Mars probe crashing! Was invalid input causing it to think it was below the ground.
- Error checking Most handled i2c error OK, but not 1-wire error.
- Buffer overruns
  sprintf into a too-small buffer, over-writing key variables
- List an *example* of poorly written embedded code.
- Why write good code?



Cut-and-pasting, good practice, among other reasons.

- Why is touch useful? force make to rebuild
- 2038 problem
  Time in Linux is seconds since 1-1-1970. Not a problem 64-bit machines, but overflows in 2038 for 32-bit. Can avoid with a 64-bit system or else a specially patched Linux system
  * discuss y2k problem ** worst problem year 19100 on websites

- ctime – last status (metadata) change (originally create



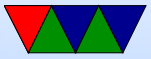
time) things like permissions change, ownership change, rename

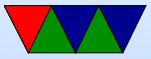
mtime – last modified

atime – last access

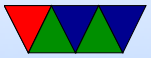
- In stat syscall. `stat` command. Why atime bad? noatime, relatime
- utime() used by touch. Cannot change ctime, set to current time
- why not believe timestamp? maybe could look at ctime. also set clock back if own machine.

  HW assignment at Cornell



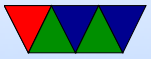


Go over the Pi Cluster Paper



Measuring Power and Energy

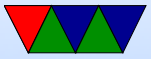
- Sense resistor or Hall Effect sensor gives you the current
- Sense resistor is small resistor. Measure voltage drop. Current V=IR Ohm's Law, so V/R=I
- Voltage drops are often small (why?) so you made need to amplify with instrumentation amplifier
- Then you need to measure with A/D converter
- $P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



Definitions

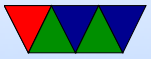
People often say Power when they mean Energy

- Dynamic Power – only consumed while computing
- Static Power – consumed all the time.
  Sets the lower limit of optimization

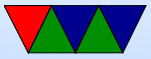


Units

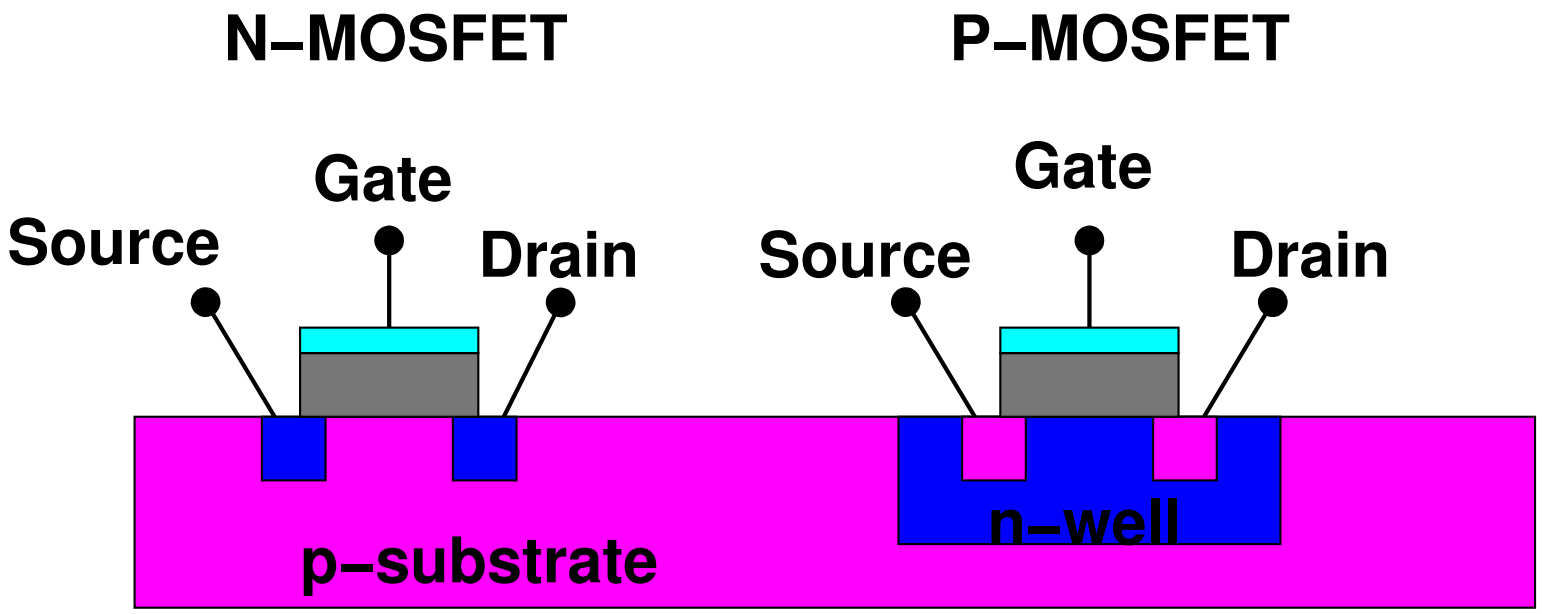
- Energy – Joules, kWH (3.6MJ), Therm (105.5MJ), 1 Ton TNT (4.2GJ), eV (1.6×10^{-19} J), BTU (1055 J), horsepower-hour (2.68 MJ), calorie (4.184 J)
- Power – Energy/Time – Watts (1 J/s), Horsepower (746W), Ton of Refrigeration (12,000 Btu/h)
- Volt-Amps (for A/C) – same units as Watts, but not same thing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

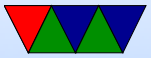


CPU Power and Energy



CMOS Transistors



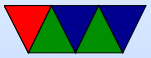


CMOS Dynamic Power

- $P = C\Delta V V_{dd} \alpha f$
  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 $P \approx \frac{1}{2} C V_{dd}^2 f$

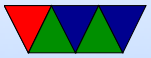
- Some pass-through loss (V momentarily shorted)



CMOS Dynamic Power Reduction

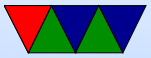
How can you reduce Dynamic Power?

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



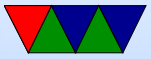
CMOS Static Power

- 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: $P_{static} = I_{leakage} V_{dd}$



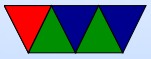
Leakage Mitigation

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



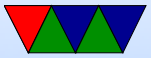
Total Energy

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



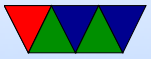
Delay

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



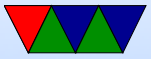
Thermal Issues

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



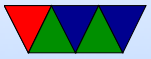
Metrics to Optimize

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



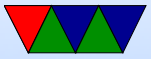
Power Optimization

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



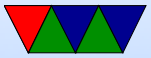
Energy Optimization

- Lowering energy can affect time too, as parts can run slower at lower voltages



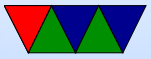
Energy Delay – Watt/t*t

- 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



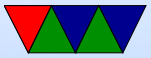
- Technology Scaling – reduces V and power.
- Transition Reduction – better logic design, have fewer transitions
  Get rid of clocks? Asynchronous? Clock-gating?
- Example with inverse ED (higher better):

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

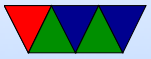


Energy Delay Squared– E*t*t

- 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



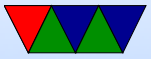
product, squared seems to be good enough



Power and Energy Concerns

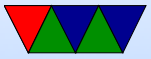
Table 1: ATLAS 300x300 DGEMM (Matrix Multiply)

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



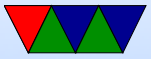
Questions

- 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?

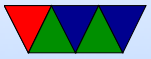


Chromebook per minute: $(1.39s \times 8.1W) + (58.61s \times 5.4W) = 327.75J$

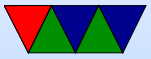
Chromebook per hour: 327.75J * 60 = 19.7kJ

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

Gumstix per hour: 164.1J * 60 = 9.8kJ

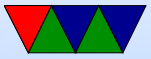


Easy ways to reduce Power Usage

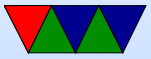


DVFS

- 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)