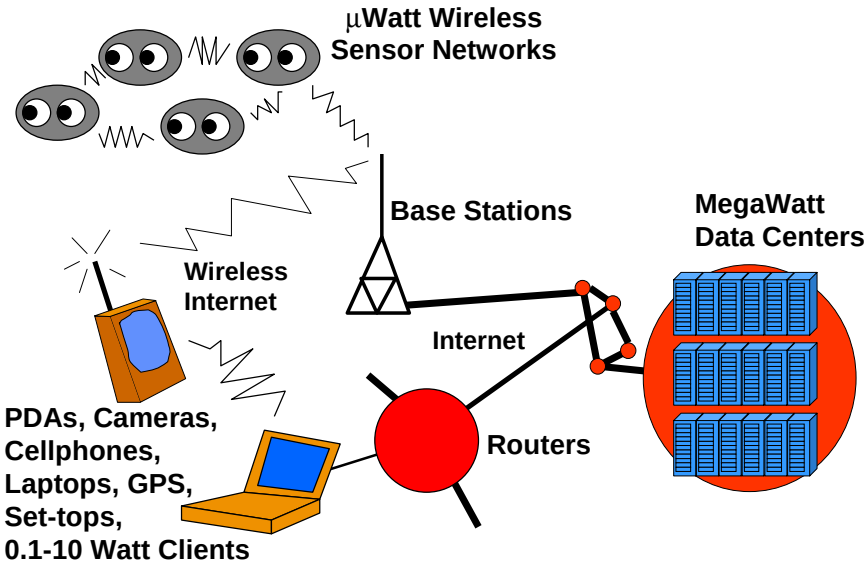
Low-Power Design

Krste Asanovic

krste@lcs.mit.edu

http://www.cag.lcs.mit.edu/6.893-f2000/

Computers Defined by Watts not MIPS



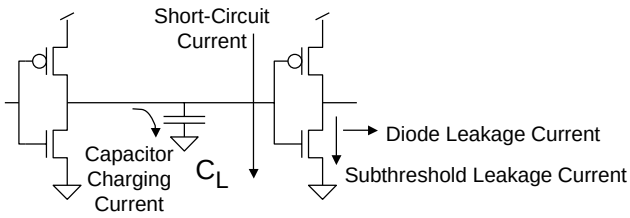
Definitions

- Energy measured in Joules
- Power is rate of energy consumption measured in Watts (Joules/second)
- Instantaneous power is Vdd * Idd

Power Impacts on System Design

- Energy consumed per task determines battery life
  - Second order effect is that higher current draws decrease effective battery energy capacity
- Current draw causes IR drops in power supply voltage
  - Requires more power/ground pins to reduce resistance R
  - Requires thick&wide on-chip metal wires or dedicated metal layers
- Switching current (dI/dT) causes inductive power supply voltage bounce $\propto$ LdI/dT
  - Requires more pins/shorter pins to reduce inductance L
  - Requires on-chip/on-package decoupling capacitance to help bypass pins during switching transients
- Power dissipated as heat, higher temps reduce speed and reliability
  - Requires more expensive packaging and cooling systems

Power Dissipation in CMOS



Primary Components:

- **Capacitor Charging (85-90% of active power)**
  - **Energy is ½ CV^2 per transition**
- **Short-Circuit Current (10-15% of active power)**
  - **When both p and n transistors turn on during signal transition**
- **Subthreshold Leakage (dominates when inactive)**
  - **Transistors don't turn off completely**
- **Diode Leakage (negligible)**
  - **Parasitic source and drain diodes leak to substrate**

Reducing Power

- **Switching power ∝ activity*½ CV^2*frequency**
  - **(Ignoring short-circuit and leakage currents)**
- **Reduce activity**
  - **Clock and function gating**
  - **Reduce spurious logic glitches**
- **Reduce switched capacitance C**
  - **Different logic styles (logic, pass transistor, dynamic)**
  - **Careful transistor sizing**
  - **Tighter layout**
  - **Segmented structures**
- **Reduce supply voltage V**
  - **Quadratic savings in energy per transition – BIG effect**
  - **But circuit delay is reduced**
- **Reduce frequency**
  - **Doesn't save energy just reduces rate at which it is consumed**
  - **Some saving in battery life from reduction in current draw**

System Levels for Energy Management

Application	Export computation to server
Algorithm	Variable resolution processing
Source Code	Improved code structure
Compiler	Energy-conscious compiler
Run-Time/O.S.	Just-in-time scheduling
Instruction Set	Energy-exposed architectures
Microarchitecture	Clock gating
Circuit Design	Low voltage-swing circuits
Fabrication Technology	SOI, Low-k dielectrics

Can usually combine savings at different levels

Voltage Scaling for Reduced Energy

- **Reducing supply voltage by 0.5 improves energy per transition by 0.25**
- **Performance is reduced – need to use slower clock**
- **Can regain performance through parallel architecture**
- **Alternatively, can trade surplus performance for lower energy by reducing supply voltage until "just enough" performance**

Parallel Architectures for Reduced Energy at Constant Throughput

- 8-bit adder/comparator
  - 40MHz at 5V, area = 530 $k\mu^2$
  - Base power Pref
- Two parallel interleaved adder/compare units
  - 20MHz at 2.9V, area = 1,800 $k\mu^2$ (3.4x)
  - Power = 0.36 Pref
- One pipelined adder/compare unit
  - 40MHz at 2.9V, area = 690 $k\mu^2$ (1.3x)
  - Power = 0.39 Pref
- Pipelined and parallel
  - 20MHz at 2.0V, area = 1,961 $k\mu^2$ (3.7x)
  - Power = 0.2 Pref
- Chandrakasan et. al. "Low-Power CMOS Digital Design", IEEE JSSC 27(4), April 1992

System Operating Modes

- Fixed throughput
  - e.g., MP3 player
  - want to minimize energy at fixed throughput (equivalent to minimizing power)
- Maximum throughput
  - e.g., spreadsheet update
  - want to run "*as fast as possible"??*
- How do we trade performance and energy/operation?
  - energy-delay product gives equal weighting
  - ED^2 gives greater weight to delay term

How do architectural ideas impact energy-efficiency?

- Instruction encoding
- Pipeline depth
- CISC versus RISC
- Register file size
- In-order versus out-of-order Superscalar
- VLIW
- Vector
- Cache hierarchy
- Branch prediction
- Multiprocessors
- Reconfigurable