Power and Energy Efficiency in Embedded Systems

Embedded Systems Design Course
EDAN15

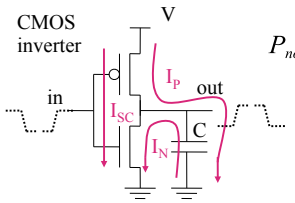
Lecture Overview

- Motivation
- Power and Energy in CMOS
- Design Choices Overview
- Hardware Design for Low-Power
- System Design
- Software Design for Low-Energy

Motivation

- package cost: 3x cheaper for low-power [SIA99]
- longer battery life, cheaper power sources:
  - smart phones, tablets, laptops
  - DS2 probe battery ('99) ~ 2h talk/Ericsson628 ('96)
  - Mars Pathfinder 13W (1996) vs. Intel Core i7 130W (2011)
  - Pace-makers: heart muscle powered?
  - Artificial retinae: powered by incident light?
- environment

Power Consumption in CMOS

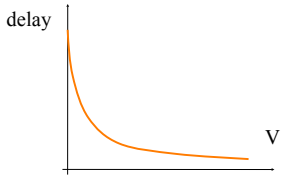


$$P_{node} = P_{st} + P_{dy} \approx \frac{1}{2}\alpha C f V^2$$

- α = signal switching (0-1,1-0)
- C = node capacitance
- f = clock frequency
- V = supply voltage

- P_{st} (static power) = leakage = 30-50% of total in sub-micron!
- P_{dy} (dynamic power) = short circuit + C charge/discharge

Power and Circuit Delay



$$delay_{node} \approx {}^{1}\!/_{V}$$

- The signal propagates slower at lower voltage:
  - Power vs. Speed trade-off

Degrees of Freedom in Power Reduction (I)

- Reduce the Supply Voltage (V):
  - attractive method (V/2 means P/4)
  - also means slowing down the clock
    - unless: CMOS threshold voltage is lowered -> leakage Power increases!
- Slower Clock Frequency (f):
  - slower design -> increase parallelism (Area)
  - can combine with V reduction

Degrees of Freedom in Power Reduction (II)

- Lower the Capacitance (C):
  - reduce interconnection length
    - stacked chips, lower feature size
  - reduce device size -> Intel Ivy Bridge 3D trans.: 22nm
    - lower Vth to maintain speed!
  - Choose better components (+: CLA, RCA, WT)
- Reduce Switching (α):
  - static: data representation & input dependent
  - dynamic: unbalanced paths (->glitches) (a-synchronous design?)

Energy Consumption

- More information than power:

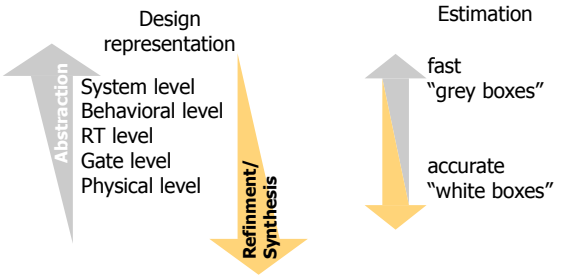
$$E = \int_0^{Time} P \times dt = P_{avg} \times Time$$

- Computation time is important!
- Reduce energy consumption:
  - Faster computations (-Time) < > lower power (-P)
  - Choose suitable resources (+-P,+-Time)

Power vs. Energy

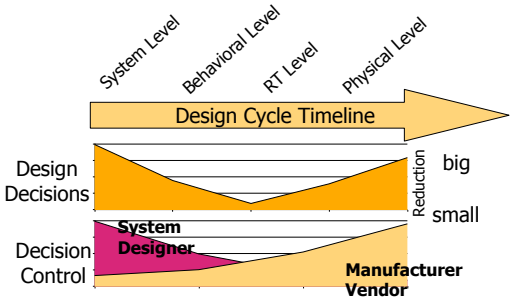
- Power
  - Cooling, packaging
  - Power supply design, wiring
- Energy
  - Battery life, operation time
- Low-power does not necessarily mean low-energy!

System Synthesis Overview

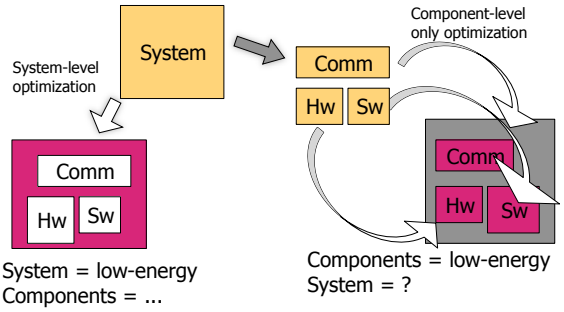


- Optimize: performance, cost, power, energy,...

Design Choices: When to focus on Power?



Design Choices: When to focus on Energy?



System = low-energy
Components = ...

Components = low-energy
System = ?

Design Choices:
Where does the energy go?

Hardware (ASIC) vs. Software (μP+Code)

- fine grain control
- not flexible on-field
- long design time
- coarse grain control
- flexible on-field
- shorter design time

Computation vs. Communication

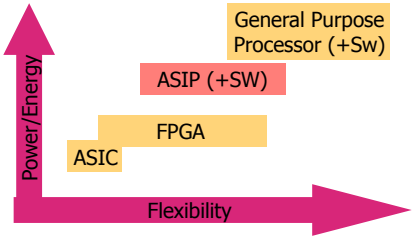
- fast if parallel (lower energy)
- no "best" resource
- a must in most systems

Where Do Embedded Systems Designers Have a Word to Say?

- Likely to choose:
  - Used CotS (boards, chips, cores, IPs)
  - System Architecture
  - Software
- Unlikely to:
  - Design/finely tune cores, ASICs, technology libraries
  - Use custom designed chips (low volume)

Design Choices:
What components to use?

OBS:
Software(function) ~ 10x Power of Hardware(same function)

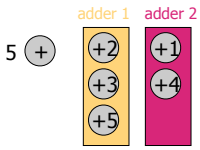


Low Power Hardware Techniques (I)

- Technology level
  - Tr sizing/feature size (C, V)
  - Low swing (V)
  - Logic styles: domino, adiabatic (C, α, f)
- Gate level
  - Dual-voltage design: high only on critical path
  - Reduce interconnect (C)

Low Power Hardware Techniques (II)

- Register-Transfer (RT) level:
  - module selection (C, f)
  - Binding, data representation, FSM states coding (α)
  - Clock gating, distribution (α, f)
  - Asynchronous vs. Synchronous Design



$Sw = Sw_{0,2} + Sw_{2,3} + Sw_{3,5} + Sw_{0,1} + Sw_{1,4}$

adder 2, adder 1 = CLA, RCA,...?

Data = 2's complement, gray, SM, redundant...?

Low Power Hardware Techniques (III)

- Behavioral level:
  - Pipelining, parallelism (larger area but early finish = lower f, V)
  - Selective shutdown, algorithm selection (avoid useless computation = lower α, C, f)
  - Compiler-like optimizations
    - Common sub-expression elimination
    - Loop un-rolling
    - ...

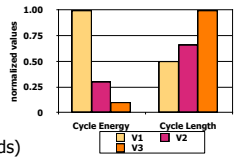
Low Power Hardware Techniques (IV)

- Processor specific issues
  - instruction set/instruction encoding
    - Communication vs. decoder power tradeoff (ThumbArm: reduced length 16bit vs. Normal 32bit)
  - register file size/cache performance
  - I/O encoding
    - Address bus: gray, NXT redundant, etc.
  - power modes: i.e. Intel XScale defines 4 modes

i80200 Modes	Active	Idle	Drowsy	Sleep
PLL/ArchState	On/Kept	On/Kept	Off/Kept	Off/Lost
To "Active" delay	-	~1clk	1000clk	2000clk
Icc (733Mhz, 1.5V)	720mA	190mA	?	~0mA

Low Power Hardware Techniques (V)

[Processor specific issues cont'd]
- Dynamic clock frequency
- Dynamic supply voltage



- Examples:
  - Intel SpeedStep™ (2 clk speeds)
    - Mobile PIII: 7/11W@ 500/600MHz, 1.35/1.1V
  - Transmeta LongRun™
    - Crusoe: 1-6W@300-600MHz(+33), 1.2-1.6V(+25mV)
  - Intel XScale ™ (allows for 5-6 clk speeds)
  - AMD PowerNow! ™ (allows for 32 Vcc)
    - AMD-K6-IIIE+500ANZ: 6-11W@ 200-500MHz, 1.7-1.4V

Low Power/Energy Hw Trends

- Globally Asynchronous Locally Synchronous GALS
  - voltage islands, independently powered, synchronous
  - asynchronous communications (FIFOs)
- Multi/Many cores
  - more parallelism instead of faster clock
  - Ex: Tilera TilePro64, Intel SCC, Nethra Ambric
- Networks on Chip
  - scalable communication structure
  - allows for GALS