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 retiniae: powered by incident light?
- environment

Power Consumption in CMOS

\[ P_{\text{node}} = P_n + P_d = \frac{1}{2} \alpha CV^2 \]

- \(P_n\) (static power) = leakage = 30-50% of total in sub-micron!
- \(P_d\) (dynamic power) = short circuit + C charge/discharge

Power and Circuit Delay

- 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 (\( \tau \)):
  - static: data representation & input dependent
  - dynamic: unbalanced paths (\( \rightarrow \) glitches)
    - (a-synchronous design?)

Energy Consumption

- More information than power:
  \[ E = \int_0^{\text{Time}} P \times dt = P_{\text{avg}} \times \text{Time} \]
- Computation time is important!
- Reduce energy consumption:
  - Faster computations (-Time) \( \leq \) 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?
### Design Choices: Where does the energy go?

#### Hardware (ASIC) vs. Software (μP+Code)
- **Fine grain control**
- **Coarse grain control**
- **Not flexible on-field**
- **Flexible on-field**
- **Long design time**
- **Shorter design time**

#### Computation vs. Communication
- **Fast if parallel (lower energy)**
- **No “best” resource**

### Where Do Embedded Systems Designers Have a Word to Say?

- **Likely to choose:**
  - Used CoS (boards, chips, cores, IPs)
  - System Architecture
  - Software
- **Unlikely to:**
  - Design/fine 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)

- **General Purpose Processor (+Sw)**
- **ASIP (+SW)**
- **FPGA**
- **ASIC**

### Low Power Hardware Techniques (I)

- **Technology level**
  - Tr sizing/feature size \(C, V\)
  - Low swing \(V\)
  - Logic styles: domino, adiabatic \(C, \alpha, f\)
- **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 \(\alpha\)
  - Clock gating, distribution \(\alpha, f\)
  - Asynchronous vs. Synchronous Design
- **Adder 1 + Adder 2**
- \(S_2 = S_{1,2} + S_{2,5} + S_{1,3} + S_{2,4}\)
- 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 \(\alpha, C, f\))
  - 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

<table>
<thead>
<tr>
<th>i80200 Modes</th>
<th>Active</th>
<th>Idle</th>
<th>Drowsy</th>
<th>Sleep</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLL/ArchState</td>
<td>On/Kept</td>
<td>On/Kept</td>
<td>Off/Kept</td>
<td>Off/Lost</td>
</tr>
<tr>
<td>To &quot;Active&quot; delay</td>
<td>~100k</td>
<td>10000k</td>
<td>20000k</td>
<td>~0ns</td>
</tr>
<tr>
<td>Vcc (733MHz, 1.5V)</td>
<td>720mA</td>
<td>190mA</td>
<td>?</td>
<td>&lt;0mA</td>
</tr>
</tbody>
</table>

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