Recap from last class

- Basic compilation optimization
  - Expression simplification
  - Dead code elimination
  - Function inlining
  - Loop optimizations
  - Register allocation
- Optimization for embedded systems
  - Optimizing for execution time
  - Optimizing for energy/power
    - Measurement, sources of energy consumption, cache
  - Optimizing for program size
    - Reduce data size and code size

ECE 1175
Embedded System Design

Power Management - I

Wei Gao

The Power Problem

- Microprocessors improve performance at the cost of power!
  - Performance/watt remains low.
- Solutions
  - Microprocessors offer features (hardware support) for controlling power consumption.
  - Software performs power management.

Outline

- Hardware support
- Power management policy
- Power manager
- Holistic approach

Types of Power Consumption

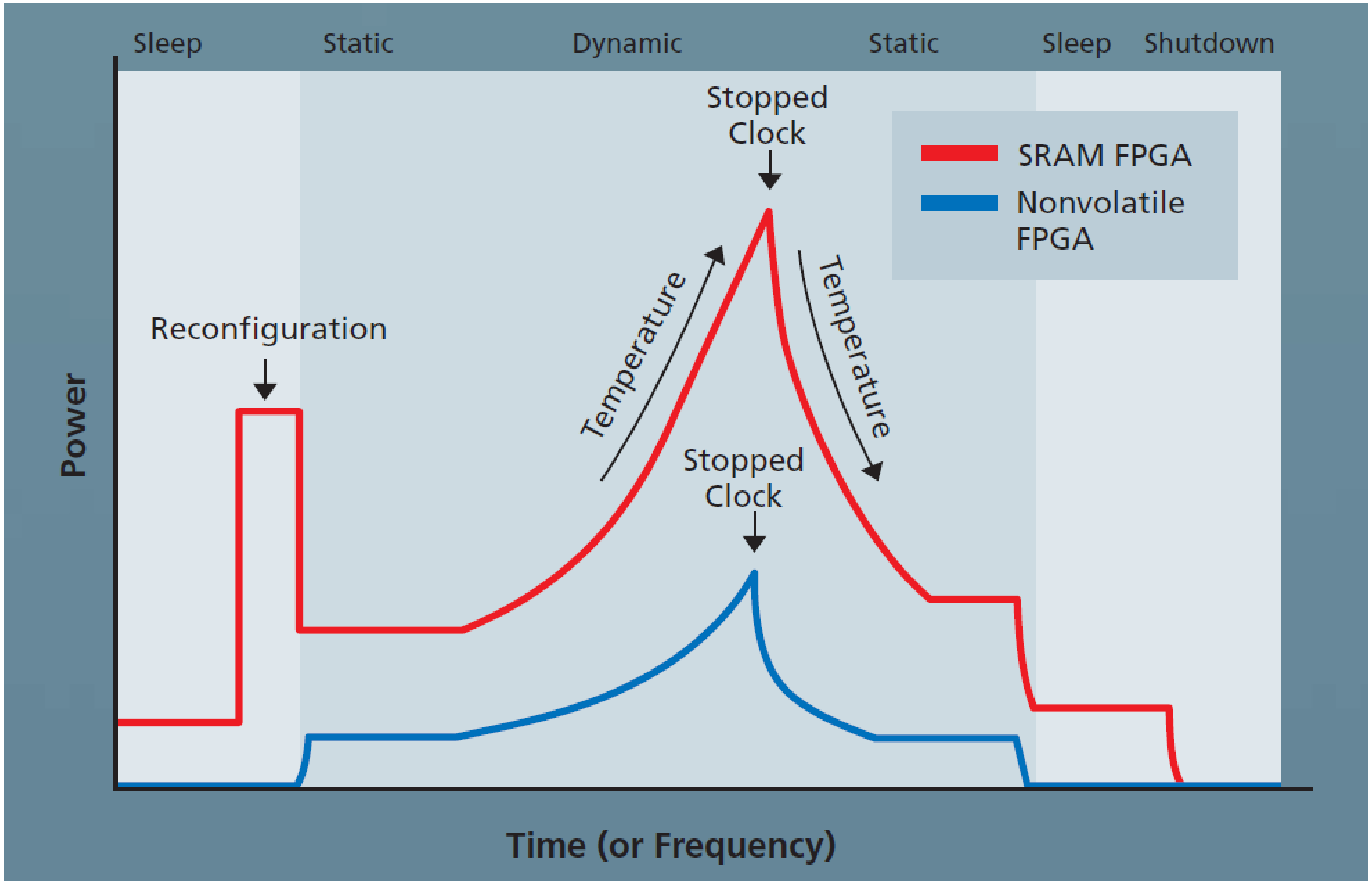
- Startup
  - SRAM devices have startup transients but flash and antifuse do not
- Static
  - Very thin oxide layer leak current when transistors are not switching
  - Increases exponentially with temperature
  - Major contributor in low-speed antifuse designs (e.g. >50% in a RTAX2000 running at 10MHZ)
  - Minor contributor in typical flash-based designs (e.g. 0.2% in example IGLOO design)

Types of Power Consumption

- Dynamic
  - Design-Specific
  - When both transistors in a logic gate are temporarily turned on during switching, get very small current surge. With millions of transistors...
  - P = f * C * V^2 (Each component power = frequency * capacitive load * voltage swing squared)
  - Increases with frequency, logic utilization, toggle rates
  - Decreases with lower voltage levels

Types of Power Consumption

- System-Dependent



CMOS Power Consumption

- CMOS technology
  - CMOS: Complementary Metal-Oxide Semiconductor
  - A common technology for constructing integrated circuits
- Voltage drops
  - Power consumption of a CMOS is proportional to the square of the power supply voltage (V^2).
- Toggling
  - CMOS uses higher power when having more activity
- Leakage
  - Even when CMOS is inactive, some charge leaks out of the circuit’s nodes

General Power-Saving Features

To deal with toggling

- Run at lower clock frequency (slower)
- Reduce activity by disabling function units when not in use.

To deal with leakage

- Eliminate leakage current by disconnecting parts from power supply when not in use.

To deal with voltage drops

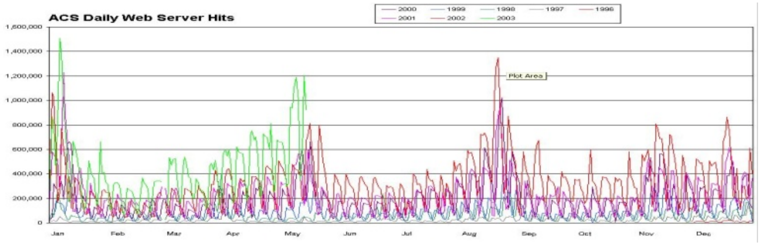
- Reduce power supply voltage to the lowest level that provides required performance
  - Pentium MMX: 2.8v
  - i7: ~1.5v
  - Cortex A8: 1.2v

Clock Gating

- Applicable to clocked digital components
  - Processors, controllers, memories
- Stop the clock → stop signal propagation in circuits
- Pros
  - Simple
  - Very short transition time if only clock distribution is stopped while clock generation is not stopped
- Cons
  - Clock itself still consumes energy
  - Cannot prevent power leaking – exist as long as power supply is connected

Dynamic Voltage Scaling

- Why voltage scaling?
  - Power ∝ V^2
    - Reduce power supply voltage → save energy
  - Lower clock frequency allows lower voltage
    - Tradeoff between performance and battery lifetime
    - Overclocking? Raise voltage for better stability
- Why dynamic?
  - Power consumption is not a constant.
    - Depending on current workloads.
    - Peak computing rate is usually much higher than average.
  - Have to be dynamic to respond to power consumption variations.



Dynamic Voltage Scaling

- Changing voltage takes time
  - Need to stabilize power supply and clock
- Continuous and discrete voltage are both possible
  - Many microprocessors have discrete power modes
    - Pentium: discrete voltage levels at 0.1v

Outline

- Hardware support
- Power management policy
- Power manager
- Holistic approach

Power Management Styles

- Static Power Management
  - Does not depend on activity.
  - Example: user-activated power-down mode.
    - Laptop monitor changes to low-power state when unplugged
- Dynamic Power Management (DPM)
  - Automatic action based on activity.
  - Example: automatically disabling function units.
    - Monitor automatically turns off after no activity for a certain period of time

Dynamic Power Management

- Goals: Energy conservation AND good performance
  - Need tradeoff between conflicting goals!
- Fundamental premises
  - Systems have varying workloads during operation
    - Running workloads: higher power and better performance
    - Idling: lower power and worse performance
  - It is possible to predict the fluctuations of workload with some degree of accuracy
    - Daytime vs. nighttime
- Performed by a Power Manager
  - React to or predict workload variations

Problem Formulations of Dynamic Power Management

- Minimize power under performance constraints
  - Real-time constraint

OR

- Optimize performance under power constraints
  - Battery lifetime constraint
  - Best server performance within the capacity of the power supply and cooling facilities

Cost of Dynamic Power Management

- Power management is not free
- Going into/out of an inactive mode costs:
  - time;
  - energy.
- Must determine if going into mode is worthwhile.
- Can model CPU power states with Power State Machine (PSM)

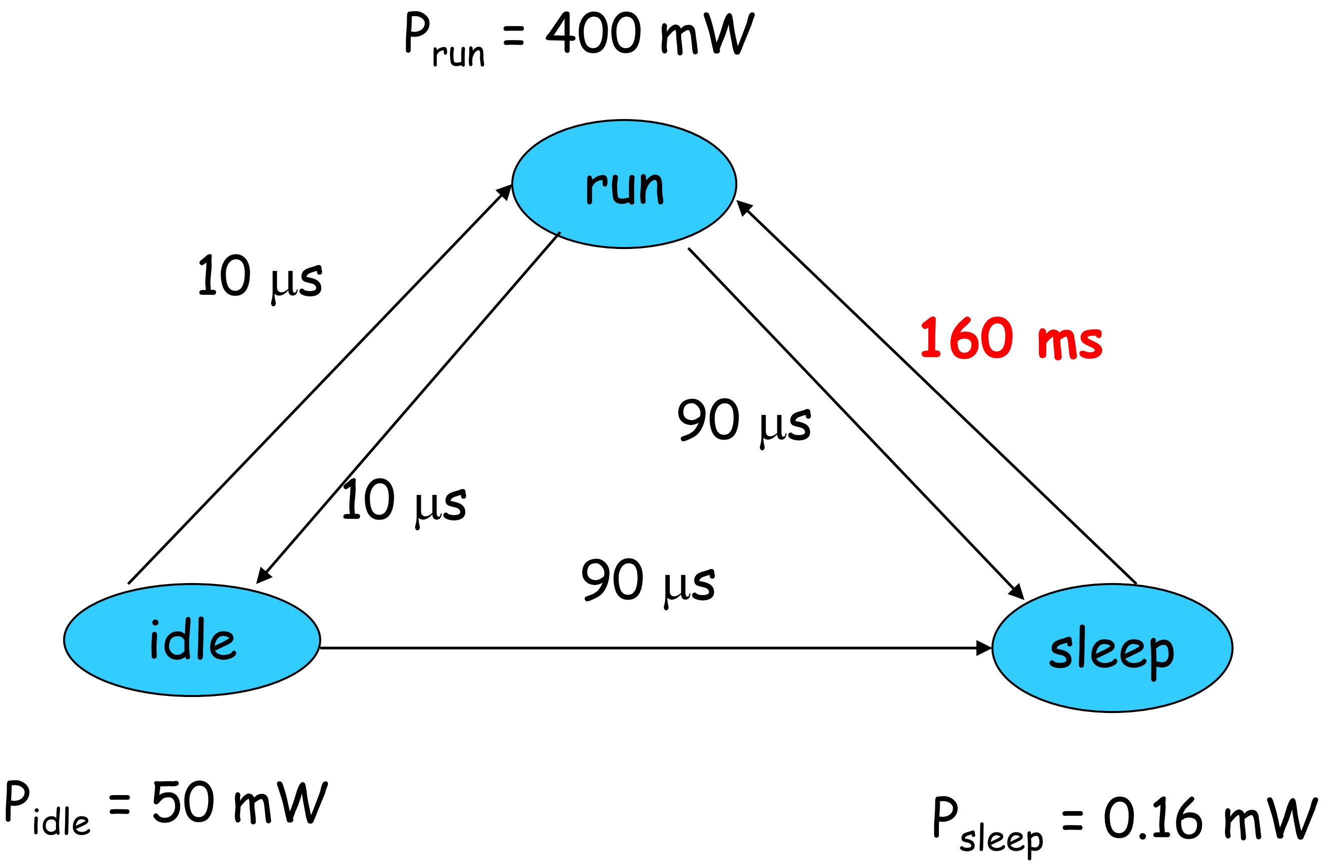
PSM Example: SA-1100

- SA-1100 is a StrongARM processor from Intel
  - Designed to provide sophisticated power management capabilities controlled by the on-chip power manager
- Three power modes:
  - Run: normal operation.
  - Idle: stops CPU clock, with I/O logic still powered.
  - Sleep: shuts off most of chip activity

SA-1100 SLEEP

- RUN → SLEEP
  - (30 μs) Flush to memorize CPU states (registers)
  - (30 μs) Reset processor state and wakeup event
  - (30 μs) Shut down clock
- SLEEP → RUN
  - (**10 ms**) Ramp up power supply
  - (**150 ms**) Stabilize clock
  - (negligible) CPU boot
- Overhead of sleep to run is much larger

SA-1100 Power State Machine



Reading

- Sections I, II, III.A, III.B, IV of

  L. Benini, A. Bogliolo and G. De Micheli, "A Survey of Design Techniques for System-Level Dynamic Power Management," IEEE Transactions on VLSI, pp. 299-316, June 2000.