Recap from last class

- Dynamic power management policies
  - Power state machine
  - Calculation of break-even time (T_{BE})
    - Depends on the power consumption during transition
    - $T_{BE} = T_{TR} + T_{TR}(P_{TR} - P_{On})/(P_{On} - P_{Off})$
  - Calculation of power saving
    - $E_S(T_{idle}) = (T_{idle} - T_{TR})(P_{On} - P_{OFF}) + T_{TR}(P_{On} – P_{TR})$
- Predictive techniques
  - Estimating the duration of idle periods
  - Safety: If an observed event happens, what's the probability of $T_{idle} > T_{BE}$?
  - Efficiency: If $T_{idle} > T_{BE}$, what's the probability of successfully predicting it in advance (e.g., o happens)?

ECE 1175
Embedded System Design

Power Management - III

Wei Gao

Metrics of Prediction Quality

- **Safety**: conditional probability Prob(p|o)
  - If an observed event happens, what's the probability of $T_{idle} > T_{BE}$?
  - Ideally, safety = 1.
- **Efficiency**: Prob(o|p)
  - If $T_{idle} > T_{BE}$, what's the probability of successfully predicting it in advance (e.g., o happens)?
- Overprediction
  - State transition too much
  - High performance penalty → poor safety
- Underprediction
  - State transition not enough
  - Wastes energy → poor efficiency

Selecting Best Timeout T_{TO}

- Interpretation as safety and efficiency metrics
- Safety: $Pr(T_{idle}>T_{TO}+T_{BE} \mid T_{idle} > T_{TO})$
  - Increasing T_{TO} improves safety
- Efficiency: $Pr(T_{idle} > T_{TO} \mid T_{idle}>T_{TO}+T_{BE})$
  - Reducing T_{TO} improves efficiency
- Karlin’s result: $T_{TO} = T_{BE}$
  - Assuming that T_{idle} follows exponential distribution

Outline

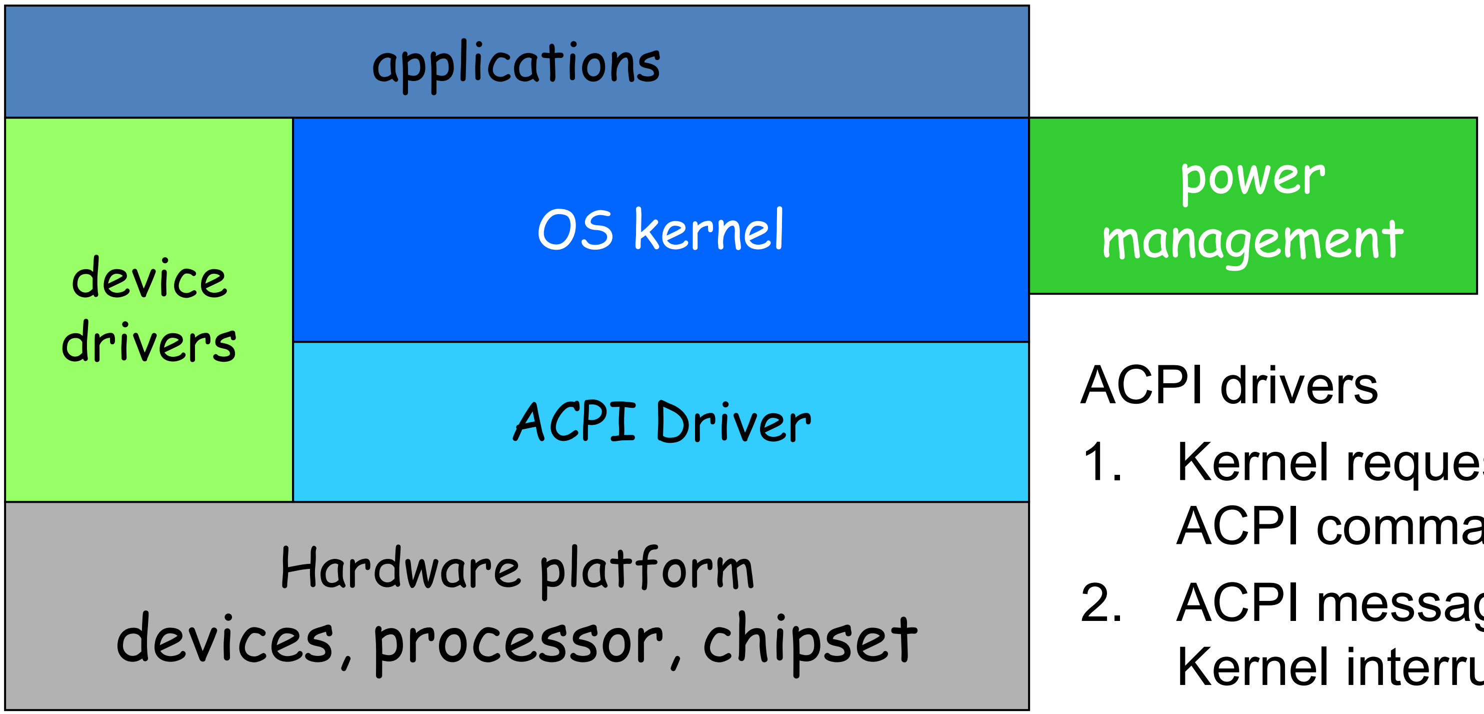
- Hardware support
- Power management policy
- Power manager
- Holistic approach

Power Manager

- Power manager is usually implemented in software (OS) for flexibility
- Hardware and software co-design
  - Software implements policy
  - Hardware implements power change mechanisms
- Need standard interfaces to deal with hardware diversity
  - Different vendors
  - Different devices: processor, sensor, controller …

Advanced Configuration and Power Interface (ACPI)

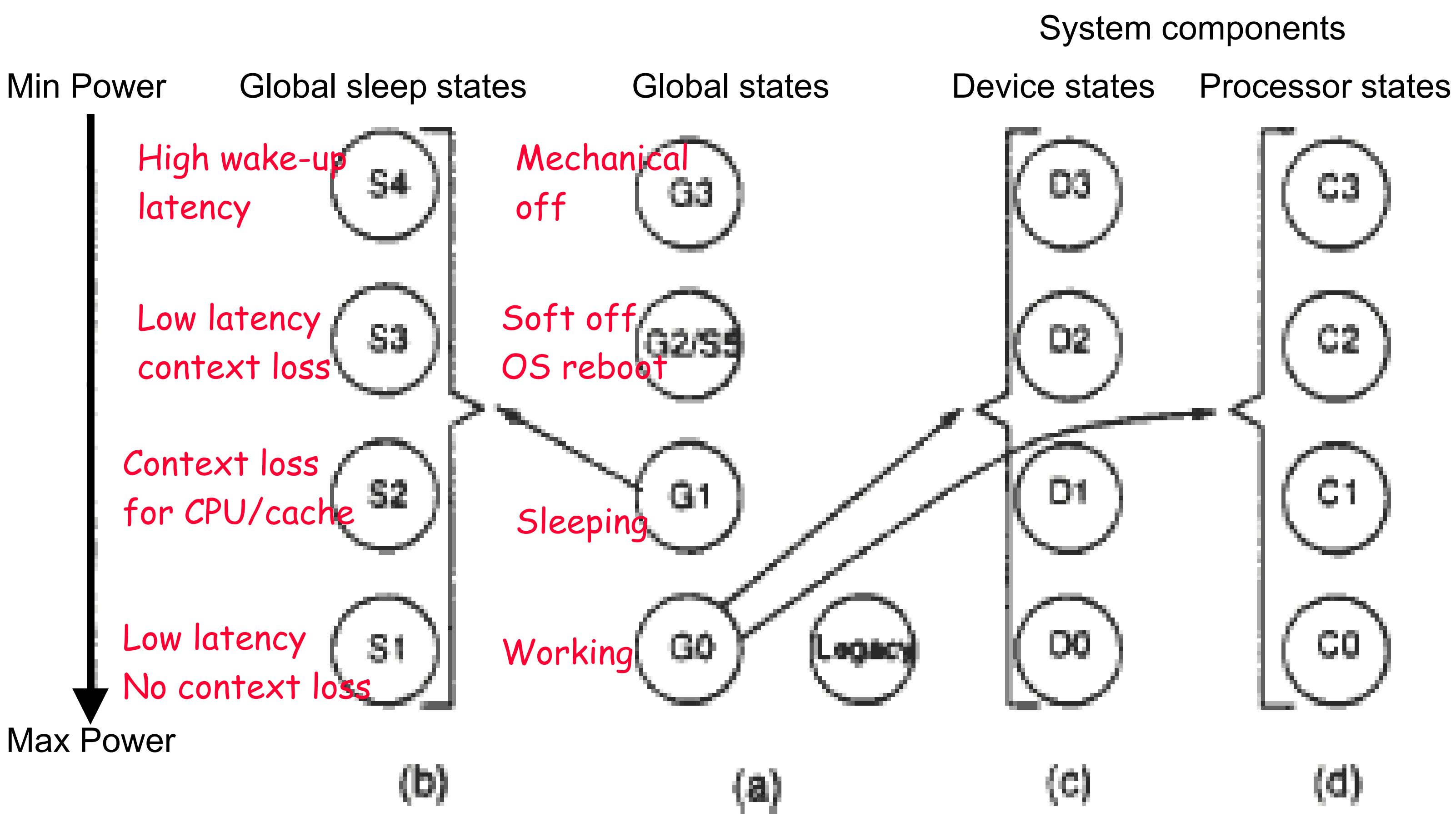
- Open standard for power management services.
  - Proposed by Intel, Microsoft and Toshiba



ACPI drivers

1. Kernel requests → ACPI commands
2. ACPI messages → Kernel interrupts

ACPI System States



Used as contracts between hardware and OS vendors

ACPI Global Power States

- G3: mechanical off – no power consumption
- G2: soft off – restore requires full OS reboot
- G1: sleeping state
  - S1: low wake-up latency with no loss of context
  - S2: low latency with loss of CPU/cache state
  - S3: low latency with loss of all state except memory
  - S4: lowest-power state with all devices off
- G0: working state

Outline

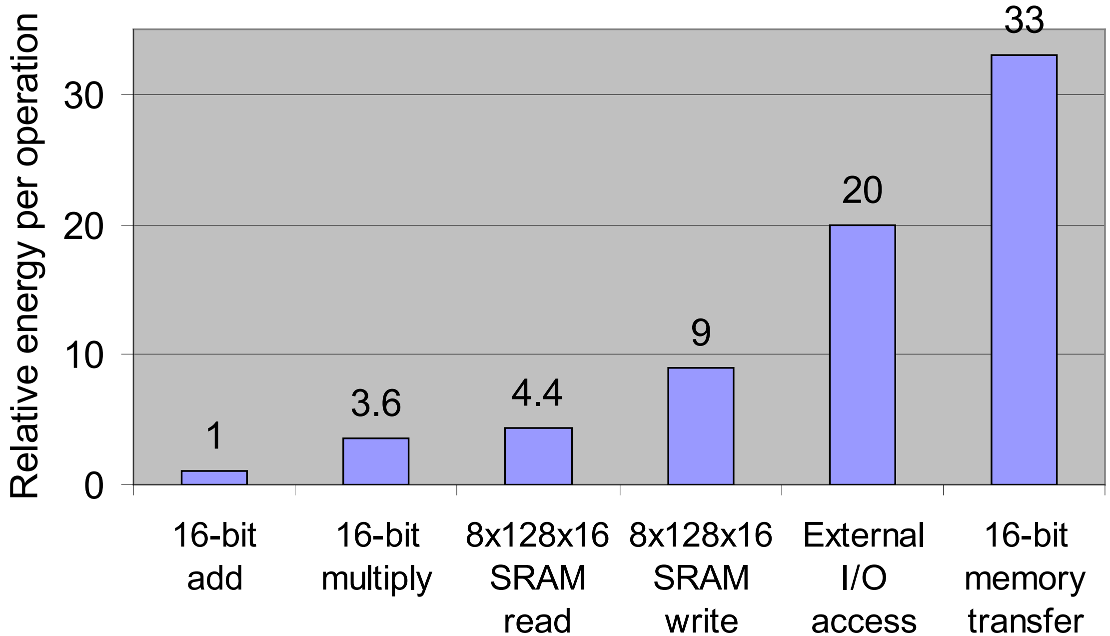
- Hardware support
- Power management policy
- Power manager
- Holistic approach

Holistic View of Power Consumption

- Instruction execution (CPU)
- Cache (instruction, data)
- Main memory
- Other: non-volatile memory, display, network interface, I/O devices

Sources of Energy Consumption

- Relative energy per operation (Catthoor et al):



- Memory transfer is the most expensive operation
  - Biggest energy optimization comes from properly organizing memory
- Energy consumption: memory > caches > registers

Memory Power Optimization

- Memory controllers put idle memory modules into low-power mode
- Minimize the number of active memory modules
  - Data placement
    - Allocate new memory pages only to active memory modules
  - Memory address mapping
    - Map consecutive memory address to a single controller
  - Memory compression
    - Compress data in multiple memory modules to one module

Cache Behavior

- Cache behavior is important
- Energy consumption has a sweet spot as cache size changes:
  - cache too small: program thrashes, burning energy on external memory accesses;
  - cache too large: cache itself burns too much power.

Optimization for Power

- Reduce memory footprint
  - Reduce code size
- Find correct cache size
  - Analyze cache behavior (size of working set)
- Minimize memory and cache access
  - Use registers efficiently → less cache access
  - Identify and eliminate cache conflicts → less memory access
- Better performance → More idle time → More power saving by power management

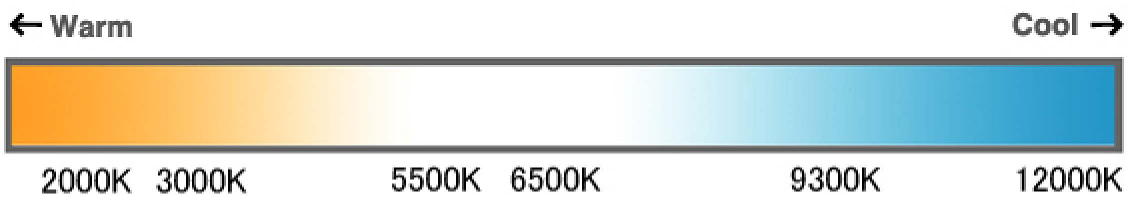
Saving the Power of I/O Devices

- I/O devices are becoming the majority of power consumption sources

- LCD display
- External storage
- Sensor readings
- Wireless connections

LCD Display

- LCD’s power consumption is related to:
  - Brightness
  - Color temperature
  - Refreshing rate
- Color temperature



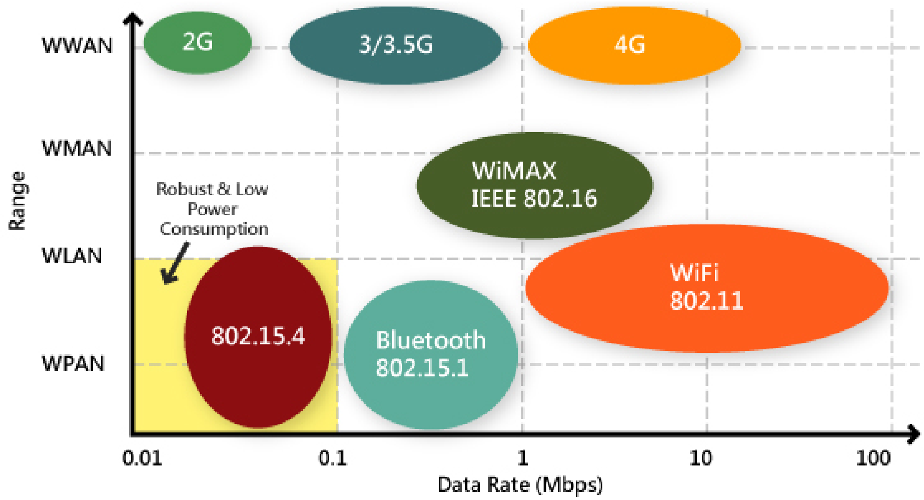
- Refreshing rate
  - 60Hz vs. 90Hz

External Storage

- Mechanical > digital
- Writes >> Reads
- Startup costs matter
  - Try to do defragmentation as much as possible

Wireless Connections

- Proportional to range & data rates



- Long-tail phenomenon in cellular networks
- Startup cost matters

Summary

- Dynamic power management policy
  - Workload-independent metrics for energy saving calculation
  - Predictive techniques
    - Fixed timeout policy
    - Predictive shutdown and wakeup
- Power manager
  - Advanced Configuration and Power Interface (ACPI)
- Holistic approach
  - Memory system
  - Cache behavior
  - Optimizing operations on I/O devices