CM0256 Pervasive Computing
Lecture 9 – Power Management

Tom Goodale
t.r.goodale@cs.cardiff.ac.uk

Lecture Outline

In this lecture we:

- Examine power management issues for mobile devices
  - What is power management ?
  - What power management strategies are there ?
  - Device issues.

Power Management

- Power Management is a way for the computer or other device to save power by turning off certain features of the computer such as the monitor, hard disk drives and other computer peripherals.
- Can be very simple
  - off and on of whole device
- or more complex
  - dynamic switching of power states of components

Power and Energy

- Energy is the total amount of work a system performs over a period of time, while power is the rate at which the system performs that work.

$$P = W/T$$

$$E = P*T$$

- In the context of computers, work involves activities associated with running programs (e.g., addition, subtraction, memory operations), power is the rate at which the computer consumes electrical energy (or dissipates it in the form of heat) while performing these activities, and energy is the total electrical energy the computer consumes (or dissipates as heat) over time.

Power and Energy

- This distinction between power and energy is important because techniques that reduce power do not necessarily reduce energy.
- For example, the power consumed by a computer can be reduced by halving the clock frequency, but if the computer then takes twice as long to run the same programs, the total energy consumed will be similar.
- Whether one should reduce power or energy depends on the context. In mobile applications, reducing energy is often more important because it increases the battery lifetime.

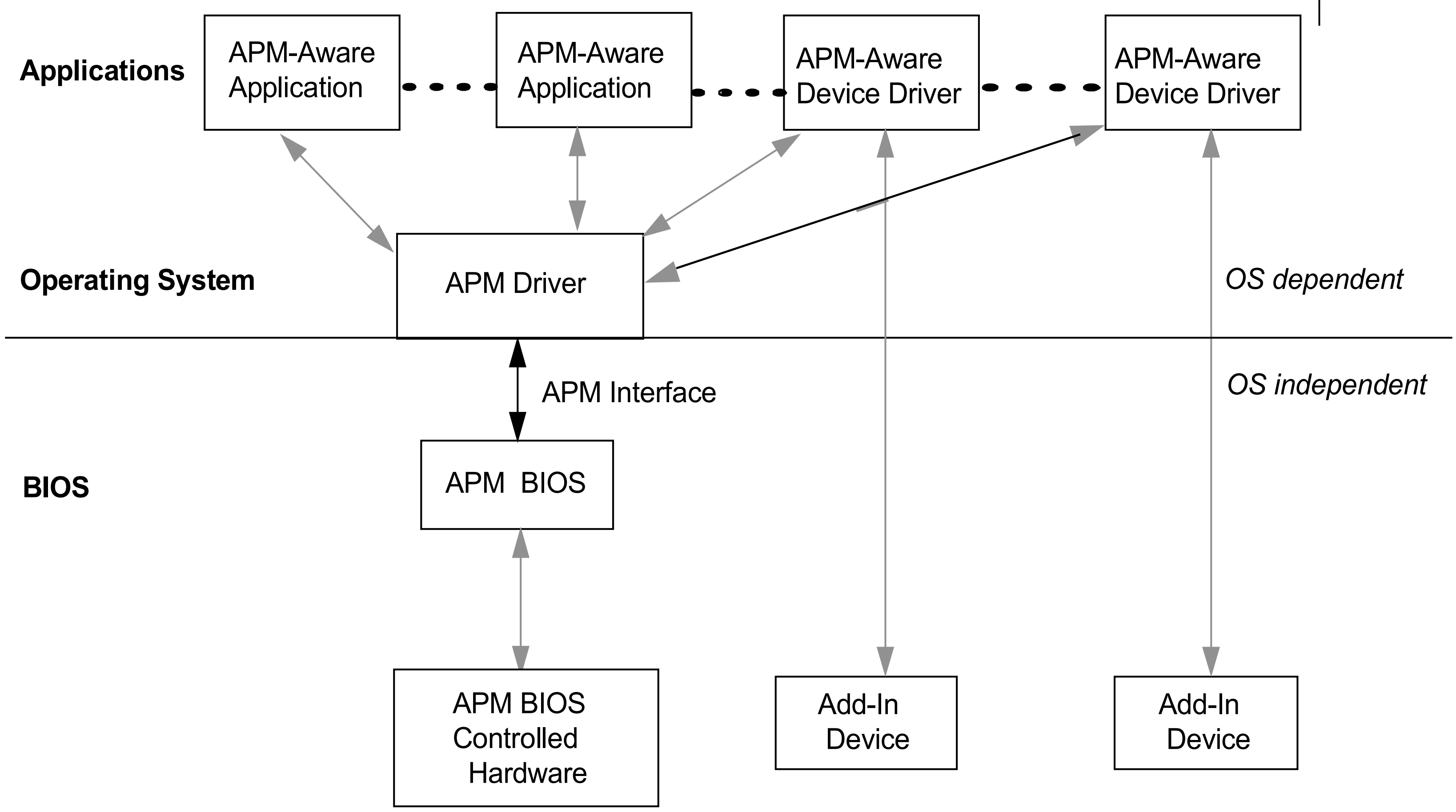
Advanced Power Management (APM)

- API developed by Intel and Microsoft which allows a BIOS to perform power management, such as reducing the CPU speed, turning off the hard disk or turning off power to the display after a preset period of inactivity in order to conserve electrical power, especially for laptop computers. Monitors supporting such APIs are usually referred to as “green monitors”, meaning environmentally friendly.

APM

- The APM BIOS is supplied by the (system) vendor and it is specific to the hardware platform. An APM driver in the OS mediates access to the APM Software Interface, which allows management of power levels.

APM


Applications
APM-Aware Application
APM-Aware Application
APM-Aware Device Driver
APM-Aware Device Driver
Operating System
APM Driver
OS dependent
OS independent
APM Interface
BIOS
APM BIOS
APM BIOS Controlled Hardware
Add-In Device
Add-In Device

APM System States

- An APM system has five general system power states:
  - Full On, APM Enabled, APM Standby, APM Suspend, and Off.
- The main difference between the states is the latency needed for the system to power up to a working state. Power consumption and performance are greatest in the Full On State and decrease with each state. APM States are defined in general terms and allow for implementation variations.

APM System States

- The Full On State is the default mode when the system is not doing power management.
- In the APM Enabled State the system is doing work but some unused devices may not be powered.
- The system enters the APM Standby State after a short period of inactivity. Recovery from the APM Standby State to the APM Enabled state appears instantaneous.
- The system enters the APM Suspend State after a relatively long period of inactivity. It takes a relatively long time to recover from the APM Suspend State to the APM Enabled State.

APM System States

System States	Characteristics
Full On	System is working. System is not power managed. All devices are on.
APM Enabled	System is working. System is power managed. The CPU clock is slowed or stopped as needed. Devices are power managed as needed.
APM Standby	System may not be working. System is in a low power state with some power savings. Most devices are in a low power mode. The CPU clock is slowed or stopped. Operational parameters are retained. System returns quickly to the APM Enabled State. The Resume Timer event must return the system to the APM Enabled State. User activity may be required to return the system to the APM Enabled State. The operating system is notified after the system transitions to the APM Enabled State. Prior operation resumes after returning to the APM Enabled State. Interrupts must still be processed normally. This may require waking up the CPU temporarily if it was stopped. The CPU may be stopped again when the APM Driver calls the CPU Idle function.
APM Suspend	System is not working. System is in a low power state with maximum power savings. Most power managed devices are not powered. The CPU clock is stopped. The CPU core is in its minimum power state. Operational parameters are saved to be restored later when resuming. System takes a relatively long time to return to the APM Enabled State. Wakeup events defined by the OEM return the system to the APM Enabled State. The Resume Timer event must be one of the wakeup events. The operating system is notified after the system transitions to the APM Enabled State. Prior operation resumes after returning to the APM Enabled State. System may go into hibernation state to save operational parameters and allow transition into and out of the off state. Hibernation is a special implementation of Suspend.
Off	System is not working. The power supply is off. Operational parameters are not saved.

APM Device States

Power Managed Device Mode	Characteristics
Device On	Device is fully powered and able to perform work. All device features are available.
Device Power Managed	Device is working, but some features may not be operational, or may be functioning at reduced performance levels. Power is maintained. Operational parameters are retained. Devices may have variable numbers of device power managed submodes.
Device Low Power	Device is not working. Power is maintained. Operational parameters are retained. Devices may have variable numbers of low power submodes.
Device Off	Device is not working. Device is powered off. Operational parameters are not retained.

APM CPU Core Control

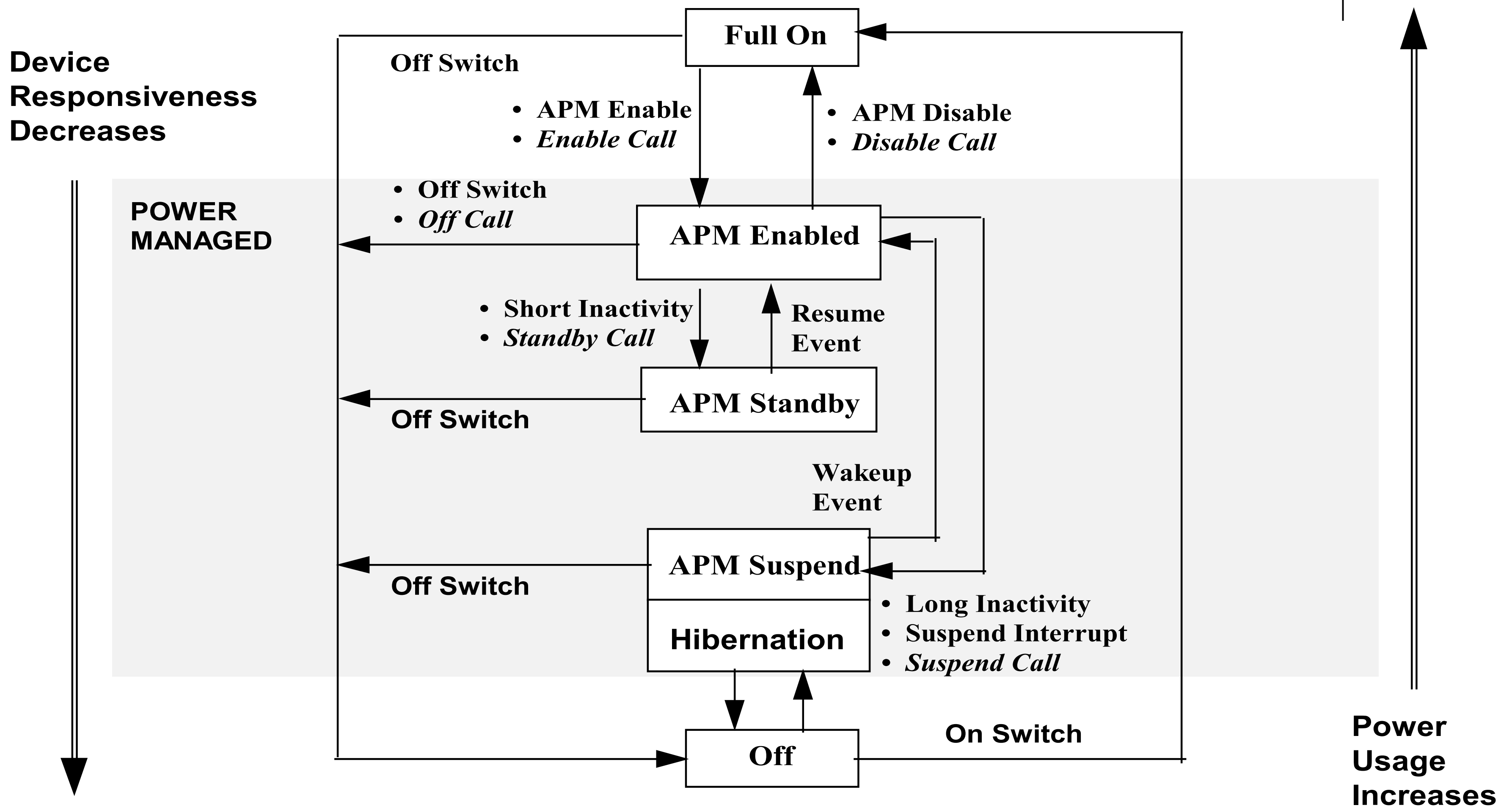
- The CPU core is managed differently than other devices. It is typically the last device to power off and the first device to power on.
- The CPU core includes the CPU clock, cache, system bus, and system timers. The CPU core is required to perform system power state changes.
  - In the APM Enabled State the CPU clock is controlled locally by being turned off and on as needed but the CPU core is never turned off.
  - In the APM Standby State the CPU clock is stopped but the CPU core is never turned off.
  - In the APM Suspend State the CPU clock is stopped and the CPU core is in its minimum power state.

APM CPU Core Control

- The CPU core is controlled through the APM BIOS. An APM Driver may notify the APM BIOS about CPU usage but the APM BIOS determines the action to take. The APM function calls that notify the APM BIOS are CPU Idle, and CPU Busy.

CPU Core Modes	Characteristics
Full On	Full speed operation. High power consumption. High performance level.
Slow Clock	Reduced speed operation. Reduced power consumption. Reduced performance level. Interrupts restore to Full On Mode. Restoration to Full On appears instantaneous.
Stop	Only a hardware interrupt starts the clock. Restoration to Full On appears instantaneous.

APM Power Transitions



Shortcomings of APM

- There are four major problems in APM.
  - Firstly, power management is done by the (vendor-specific) BIOS, and the OS does not have any knowledge of it. One example of this, is when the user sets idle-time values for a hard drive in the APM BIOS, that when exceeded, it (BIOS) would spin down the hard drive, without the consent of the OS.
  - Secondly, the APM logic is embedded in the BIOS, and it operates outside the scope of the OS. This means users can only fix problems in their APM BIOS by flashing a new one into the ROM; which is a very dangerous procedure with the potential to leave the system in an unrecoverable state if it fails.

APM Shortcomings

- Thirdly, APM is a vendor-specific technology, which means that there is a lot of duplication of effort, and bugs found in one vendor's BIOS, may not be solved in others.
- Last but not the least, the APM BIOS did not have enough room to implement a sophisticated power policy, or one that can adapt very well to the purpose of the machine.

Advanced Configuration and Power Interface (ACPI)

- an open industry standard first released in December 1996 developed by HP, Intel, Microsoft, Phoenix and Toshiba that defines common interfaces for hardware recognition, motherboard and device configuration and power management.

ACPI Principle Goals: 1

- Enable all computer systems to implement motherboard configuration and power management functions, using appropriate cost/function tradeoffs.
  - Computer systems include (but are not limited to) desktop, mobile, workstation, and server machines.
  - Machine implementers have the freedom to implement a wide range of solutions, from the very simple to the very aggressive, while still maintaining full OS support.
  - Wide implementation of power management will make it practical and compelling for applications to support and exploit it. It will make new uses of PCs practical and existing uses of PCs more economical.

ACPI Principle Goals: 2

- Enhance power management functionality and robustness.
  - Power management policies too complicated to implement in a ROM BIOS can be implemented and supported in the OS, allowing inexpensive power managed hardware to support very elaborate power management policies.
  - Gathering power management information from users, applications, and the hardware together into the OS will enable better power management decisions and execution.
  - Unification of power management algorithms in the OS will reduce conflicts between the firmware and OS and will enhance reliability.

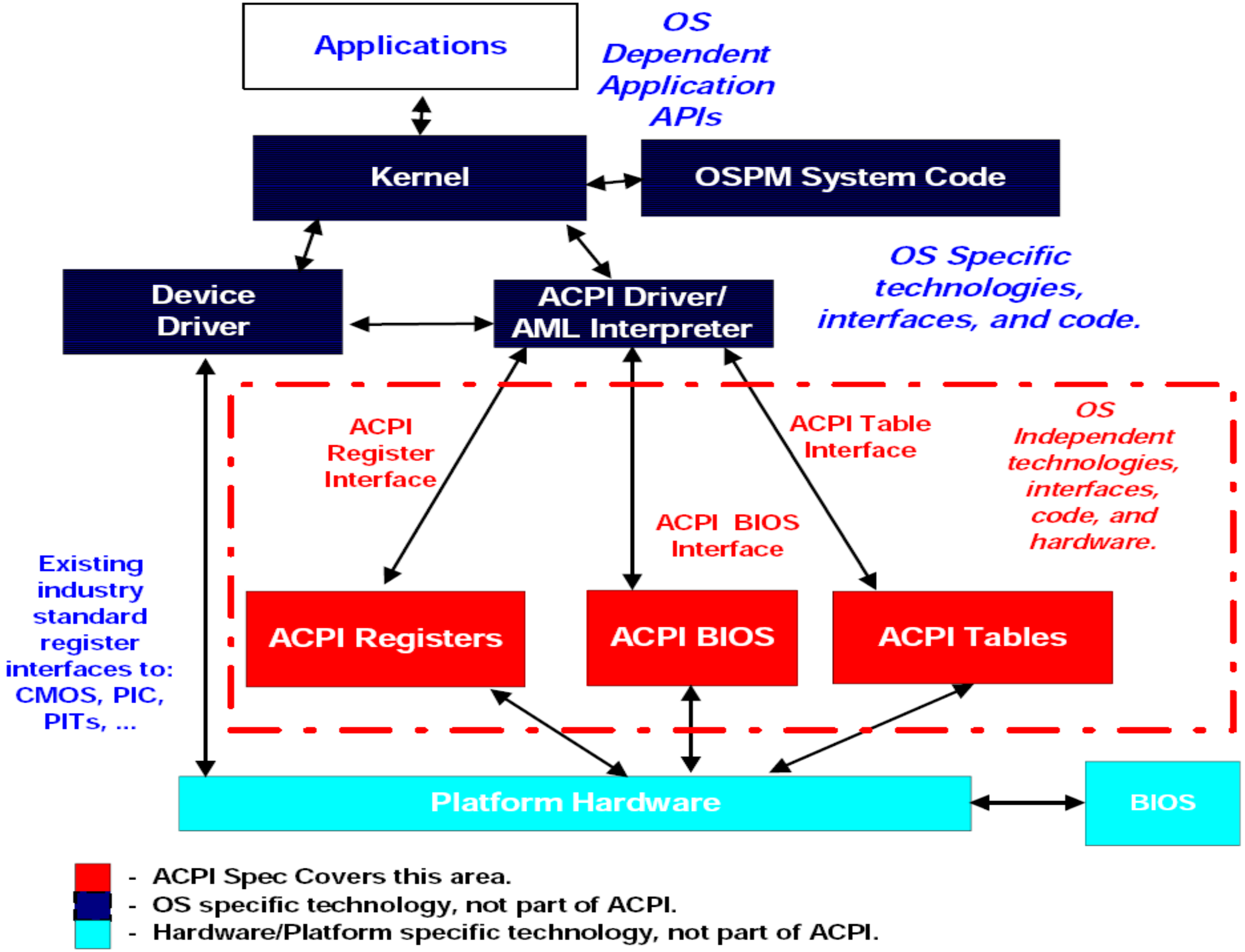
ACPI Principle Goals: 3

- Facilitate and accelerate industry-wide implementation of power management.
  - OSPM and ACPI will reduce the amount of redundant investment in power management throughout the industry, as this investment and function will be gathered into the OS. This will allow industry participants to focus their efforts and investments on innovation rather than simple parity.
  - The OS can evolve independently of the hardware, allowing all ACPI-compatible machines to gain the benefits of OS improvements and innovations.

ACPI Principle Goals: 4

- Create a robust interface for configuring motherboard devices.
  - Enable new advanced designs not possible with existing interfaces.

ACPI Global System



ACPI Global System State Definitions

- Global system states (Gx states) apply to the entire system and are visible to the user.
- Defined by six principal criteria:
  1. Does application software run?
  2. What is the latency from external events to application response?
  3. What is the power consumption?
  4. Is an OS reboot required to return to a working state?
  5. Is it safe to disassemble the computer?
  6. Can the state be entered and exited electronically?

ACPI G3 Mechanical Off

- A computer state that is entered and left by a mechanical means (for example, turning off the system's power through the movement of a large red switch). Various government agencies and countries require this operating mode.
- It is implied by the entry of this off state through a mechanical means that no electrical current is running through the circuitry and that it can be worked on without damaging the hardware or endangering service personnel.
- The OS must be restarted to return to the Working state.
- No hardware context is retained. Except for the real-time clock, power consumption is zero.

ACPI G2/S5 Soft Off

- A computer state where the computer consumes a minimal amount of power. No user mode or system mode code is run.
- This state requires a large latency in order to return to the Working state.
- The system's context will not be preserved by the hardware.
- The system must be restarted to return to the Working state.
- It is not safe to disassemble the machine in this state.

ACPI G1 Sleeping

- A computer state where the computer consumes a small amount of power, user mode threads are not being executed, and the system "appears" to be off (from an end user's perspective, the display is off, and so on).
- Latency for returning to the Working state varies on the wake environment selected prior to entry of this state (for example, whether the system should answer phone calls).
- Work can be resumed without rebooting the OS because large elements of system context are saved by the hardware and the rest by system software.
- It is not safe to disassemble the machine in this state.

ACPI G0 Working

- A computer state where the system dispatches user mode (application) threads and they execute. In this state, peripheral devices (peripherals) are having their power state changed dynamically.
- The user can select, through some UI, various performance/power characteristics of the system to have the software optimize for performance or battery life.
- The system responds to external events in real time.
- It is not safe to disassemble the machine in this state.

ACPI S4 Non-Volatile Sleep

- Special global system state that allows system context to be saved and restored (relatively slowly) when power is lost to the motherboard.
- If the system has been commanded to enter S4, the OS will write all system context to a file on non-volatile storage media and leave appropriate context markers. The machine will then enter the S4 state.
- When the system leaves the Soft Off or Mechanical Off state, transitioning to Working (G0) and restarting the OS, a restore from a NVS file can occur.
  - will only happen if a valid non-volatile sleep data set is found, certain aspects of the configuration of the machine have not changed, and the user has not manually aborted the restore.

ACPI S4 Non-Volatile Sleep

- The net effect for the user is what looks like a resume from a Sleeping (G1) state (albeit slower). Aspects of the machine configuration that must not change include, but are not limited to, disk layout and memory size.
- Notice that for the machine to transition directly from the Soft Off or Sleeping states to S4, the system context must be written to non-volatile storage by the hardware; entering the Working state first so that the OS or BIOS can save the system context takes too long from the user's point of view. The transition from Mechanical Off to S4 is likely to be done when the user is not there to see it.
- Because the S4 state relies only on non-volatile storage, a machine can save its system context for an arbitrary period of time (on the order of many years).

Summary of ACPI Global Power States

Global System State	Software Runs	Latency	Power Consumption	OS restart required	Safe to disassemble computer	Exit state electronically
G0 Working	Yes	0	Large	No	No	Yes
G1 Sleeping	No	>0, varies with sleep state	Smaller	No	No	Yes
G2/S5 Soft Off	No	Long	Very near 0	Yes	No	Yes
G3 Mechanical Off	No	Long	RTC Battery	Yes	Yes	No

ACPI Device Power States

- Device states apply to any device on any bus. They are generally defined in terms of four principal criteria:
  - Power consumption. How much power the device uses.
  - Device context. How much of the context of the device is retained by the hardware. The OS is responsible for restoring any lost device context (this may be done by resetting the device).
  - Device driver. What the device driver must do to restore the device to full on.
  - Restore time. How long it takes to restore the device to full on.

ACPI Device Power States

- Many devices do not have all four power states defined. Devices may be capable of several different low-power modes, but if there is no user perceptible difference between the modes, only the lowest power mode will be used.
- D3 Off
  - Power has been fully removed from the device. The device context is lost when this state is entered, so the OS software will reinitialize the device when powering it back on. Since device context and power are lost, devices in this state do not decode their address lines. Devices in this state have the longest restore times. All classes of devices define this state.

ACPI Device Power States

- D2
  - The meaning of the D2 Device State is defined by each device class. Many device classes may not define D2. In general, D2 is expected to save more power and preserve less device context than D1 or D0. Buses in D2 may cause the device to lose some context (for example, by reducing power on the bus, thus forcing the device to turn off some of its functions).
- D1
  - The meaning of the D1 Device State is defined by each device class. Many device classes may not define D1. In general, D1 is expected to save less power and preserve more device context than D2.

ACPI Device Power States

- D0 Fully-On
  - This state is assumed to be the highest level of power consumption. The device is completely active and responsive, and is expected to remember all relevant context continuously.

Device State	Power Consumption	Device Context Retained	Driver Restoration
D0 - Fully On	As needed for operation	All	None
D1	D0>D1>D2>D3	>D2	<D2
D2	D0>D1>D2>D3	<D1	>D1
D3 - Off	0	None	Full initialisation and load

ACPI CPU Power States

- Cx states are processor power consumption and thermal management states within the global working state, G0. They have specific entry and exit semantics:
- C0 Processor Power State
  - While the processor is in this state, it executes instructions.
- C1 Processor Power State
  - Lowest latency power state. The hardware latency in this state must be low enough that the operating software does not consider the latency aspect of the state when deciding whether to use it. Aside from putting the processor in a non-executing power state, this state has no other software-visible effects.

ACPI CPU Power States

- C2 Processor Power State
  - Improved power savings over the C1 state. The worst-case hardware latency for this state is provided via the ACPI system firmware and the operating software can use this information to determine when the C1 state should be used instead of the C2 state. Aside from putting the processor in a non-executing power state, this state has no other software-visible effects.

ACPI CPU Power States

- C3 Processor Power State
  - Improved power savings over the C1 and C2 states. The worst-case hardware latency for this state is provided via the ACPI system firmware and the operating software can use this information to determine when the C2 state should be used instead of the C3 state. While in the C3 state, the processor’s caches maintain state but ignore any snoops. The operating software is responsible for ensuring that the caches maintain coherency.

ACPI Sleeping State Definitions

- Sleeping states (Sx states) are types of sleeping states within the global sleeping state, G1.
- S1 Sleeping State
  - A low wake latency sleeping state. No system context is lost (CPU or chip set) and hardware maintains all system context.
- S2 Sleeping State
  - A low wake latency sleeping state. Similar to the S1 sleeping state except that the CPU and system cache context is lost (the OS is responsible for maintaining the caches and CPU context). Control starts from the processor’s reset vector after the wake event.

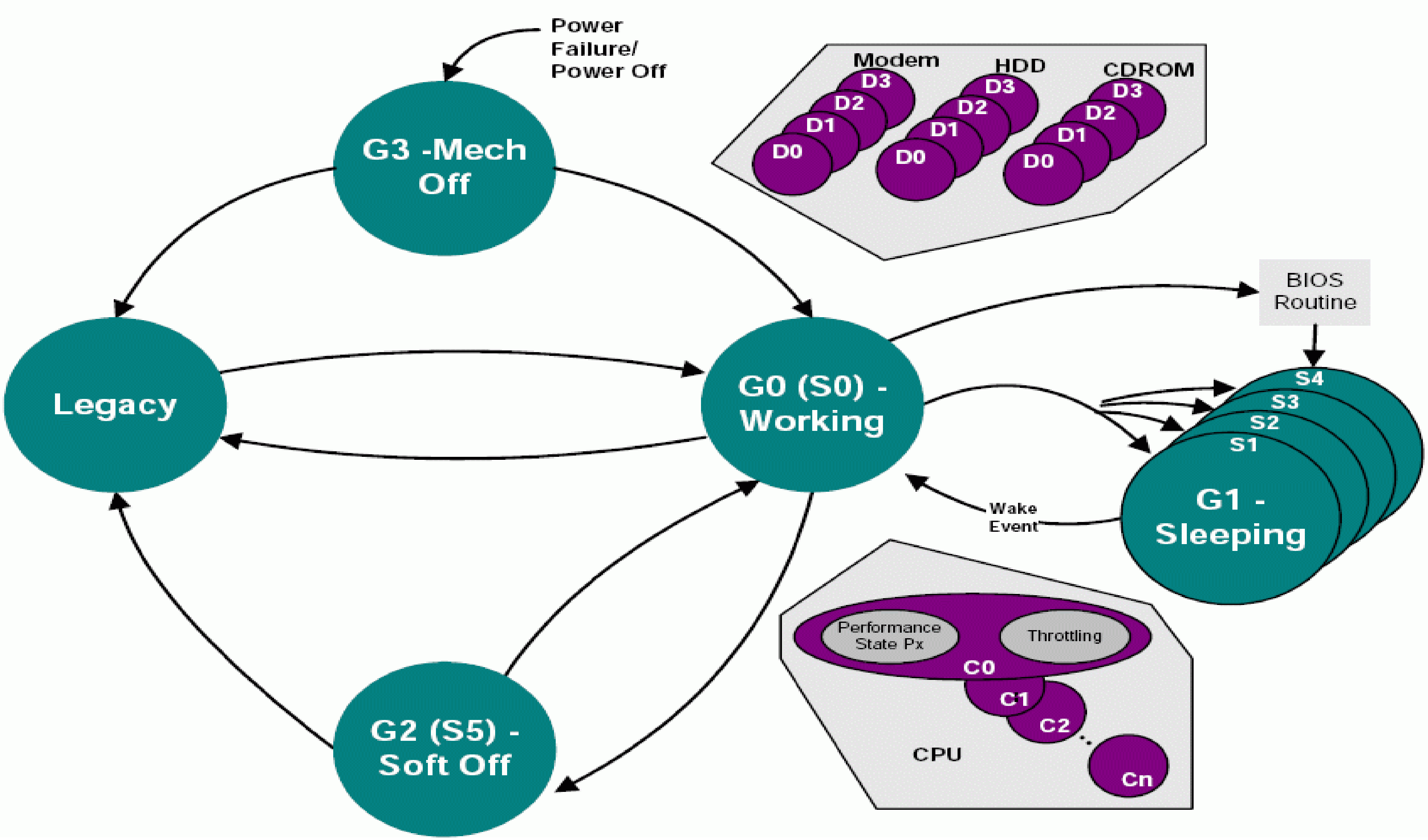
ACPI Sleeping State Definitions

- S3 Sleeping State
  - A low wake latency sleeping state where all system context is lost except system memory. CPU, cache, and chip set context are lost in this state. Hardware maintains memory context and restores some CPU and L2 configuration context. Control starts from the processor’s reset vector after the wake event.
- S4 Sleeping State
  - The S4 sleeping state is the lowest power, longest wake latency sleeping state supported by ACPI. In order to reduce power to a minimum, it is assumed that the hardware platform has powered off all devices. Platform context is maintained.

ACPI Sleeping State Definitions

- S5 Soft Off State
  - The S5 state is similar to the S4 state except that the OS does not save any context. The system is in the “soft” off state and requires a complete boot when it wakes. Software uses a different state value to distinguish between the S5 state and the S4 state to allow for initial boot operations within the BIOS to distinguish whether or not the boot is going to wake from a saved memory image.

ACPI States and Transitions

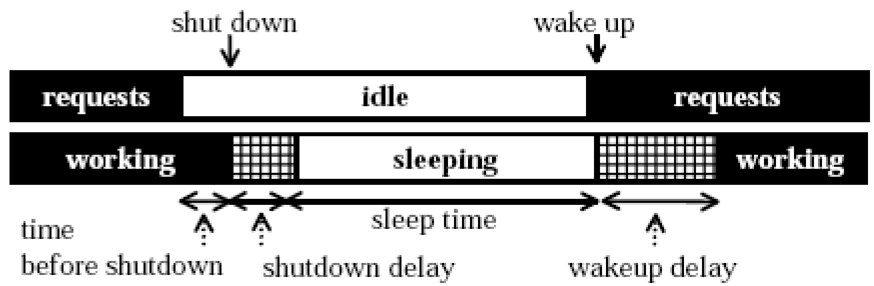


Dynamic Power Management (DPM)

- An approach to reduce power consumption without significantly degrading performance.
- DPM shuts down devices when they are not being used and wakes them up when necessary.
- When a device is not used, it is called idle; otherwise, it is called busy.
- DPM algorithms observe request patterns and predict the length of idle periods. Idle periods can be defined in different ways;
  - e.g. as “time with no requests waiting for service”.

DPM

The device is in a working state when it can serve requests with higher power consumption



- The device is sleeping when it consumes less power and cannot serve requests.
- Shutting down and waking up a device usually cause performance degradation and require extra energy. Therefore, DPM algorithms shut down a device only when an idle period is long enough to justify performance degradation and state-transition energy.

DPM

- DPM encompasses a set of techniques that achieve energy efficient computation by selectively turning off (or reducing the performance of) system components when they are idle (or partially unexploited).
- DPM is used in various forms in most portable (and some stationary) electronic designs; yet its application is sometimes primitive because its full potential is still unexplored and because the complexity of interfacing heterogeneous components has limited designers to simple solutions.

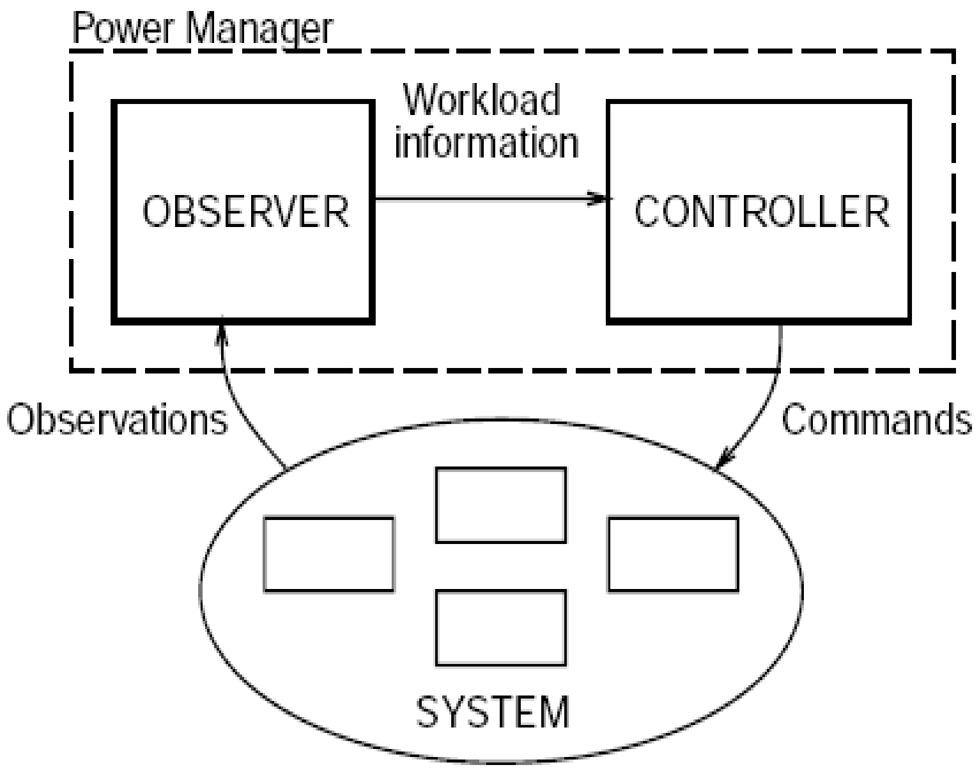
DPM Premises

- The fundamental premise for the applicability of DPM is that systems (and their components) experience nonuniform workloads during operation time. Such an assumption is valid for most systems, both when considered in isolation and when internetworked.
- A second assumption of DPM is that it is possible to predict, with a certain degree of confidence, the fluctuations of workload. Workload observation and prediction should not consume significant energy.

DPM

- Dynamic power managers can have different embodiments, according to the level (e.g., component, system, network) where DPM is applied and to the physical realization style (e.g., timer, hard-wired controller, software routine). Typically, a power manager (PM) implements a control procedure based on some observations and/or assumptions about the workload
- The control procedure is often called a policy.
  - An example of a simple policy, ubiquitously used for laptops and palmtops, is the timeout policy, which shuts down a component after a fixed inactivity time, under the assumption that it is highly likely that a component will remain idle if it has been idle for the timeout time.

Abstract View of System-Level Power Manager



DPM Techniques

- Putting a power managed component (PMC) into an inactive state causes a period of inactivity whose duration **Tn** is the sum of the actual time spent in the target state and the time spent to enter and exit it. Define the break-even time (denoted **TBE**) as the minimum inactivity time required to compensate the cost of shutting down a component.
- The break-even time TBE is inferred directly from the power-state machine of a PMC.

DPM Techniques

- If Tn < TBE, either there is not enough time to enter and exit the inactive state or the power saved when in the inactive state does not amortize the additional power consumption typically required to turn on the component.
- DPM aims at exploiting idleness to transition a component to an inactive low-power state. If no performance loss is tolerated, the length of the workload’s idle periods is an upper bound for the inactivity time of the resource.

DPM Techniques

- On the other hand, if some performance loss is tolerated, inactivity times may be longer than idle periods.
- An analysis of the break-even times of the system's PMCs in conjunction with the workload statistics can measure the usefulness of applying DPM.
- On the other hand, when designing components for power-managed systems, workload statistics can yield useful information on the transition times and power levels required for the component to be useful.
- When designing a power-managed system, the central problem is to determine the policy that the PM will implement.

Predictive Techniques

- In most real-world systems there is little knowledge of future input events, and DPM decisions have to be taken based on uncertain predictions. The rationale in all predictive techniques is that of exploiting the correlation between the past history of the workload and its near future, in order to make reliable predictions about future events.
- we are interested in predicting idle periods long enough to go to sleep; that is, events for which $T_{idle} > T_{BE}$.

Predictive Techniques

- The most common predictive PM policy is the fixed timeout that uses the elapsed idle time as the observed event used to predict the total duration of the current idle period $T\_idle > TTO + TBE$.
- The policy can be summarized as follows: when an idle period begins, a timer is started with duration TTO. If after TTO the system is still idle, then the PM forces the transition to the off state. The system remains off until it receives a request from the environment that signals the end of the idle period. The critical design decision is obviously the choice of the timeout value TTO.

Adaptive Techniques

- Since the optimality of DPM strategies depends on the workload statistics, static predictive techniques are all ineffective (i.e., suboptimal) when the workload is either unknown a priori, or nonstationary. So some form of adaptation is required. For timeouts the only parameter to be adjusted is timer duration, for history-based predictors even the type of observed events could in principle be adapted to the workload.

Adaptive Techniques

- Several adaptive predictive techniques were proposed to deal with nonstationary workloads.
  - In the work by Krishnan et al. a set of timeout values is maintained, and each timeout is associated with an index indicating how successful it will be. The policy chooses, at each idle time, the timeout that will perform best among the set of available ones.

Adaptive Techniques

- Another policy, presented by Helmbold et al., also keeps a list of candidate timeouts, and assigns a weight to each based on how well it will perform relative to an optimum offline strategy for past requests. The actual timeout is obtained as a weighted average of all candidates with their weight.
- Another approach, introduced by Douglis et al. is to keep only one timeout value and to increase it when it is causing too many shutdowns. The timeout is decreased when more shutdowns can be tolerated.

CPU

- Dynamic voltage scaling
  - power consumption scales with the cube of the voltage
- Frequency changing
  - power consumption scales linearly with frequency
- Halt instruction
  - Stop processor when idle

Hard disks

- Hard disks may consume up to 20% of total energy in a computer.
  - Recent studies find that hard disks will remain major power consumers in the near future.
- Hard disks have large performance and power overhead because of mechanical inertia.
  - Spinning down or up disk plates takes several seconds, equivalent to hundreds of millions of instructions in modern computers.
- Hard disks are not always needed when computers are running if the physical memory contains all the information needed;
  - for example, caching may avoid unnecessary spin-ups.

Hard disks

- Most of the power loss stems from the rotating platter during disk activity and idle periods. To save energy, operating systems pause disk rotation after a fixed period of inactivity and restart it during the next disk access. Decision of when to spin down an idle disk involves tradeoffs between power and performance.
- High idle time thresholds lead to more energy loss but better performance since the disk need not restart to process requests.
- Low thresholds allow a disk to spin down sooner when idle but increase the probability of an immediate restart to process a spontaneous request. These immediate restarts (bumps) waste energy and cause delays.

Display

- Brightness
  - Most LCDs are backlit. Can reduce brightness when on battery power.
  - Normally, an operating system dims the display after a sufficiently long interval with no user response. This heuristic may not coincide with a user's intentions.
- Blanking
  - For even longer idle time, can bank or switch off display.

Network Interfaces

- Network interfaces such as wireless cards present a unique challenge for PM techniques because the naive solution of shutting them off would disconnect hosts from other devices with which they may have to communicate. Thus power management strategies also include protocols for synchronizing the communication of these devices with other devices in the network.

Other Components

- Fan
  - Needed to keep temperature of other components within tolerances
- Bluetooth
- USB
- Firewire
- Cardbus/PCMCIA

Lecture Summary

In this lecture we have:

- Looked at the old APM system
- Looked at the new ACPI system
- Examined DPM strategies.
- Examined device specific issues.

End of Lecture