PC Power Management Technology

March 20, 2002
By: Winn Rosch

Introduction and Background

Every ecologist knows that it is more economical to save energy than to produce it. The same rule holds true for notebook computer power or the power for any battery-operated device. Computer makers know they can gain far more battery life by cutting consumption than by increasing capacity.

Cuts and increases simply play by different rules. Increasing battery capacity is a matter of chemistry, and no breakthroughs loom on the horizon for radically increasing battery capacity. The hardest work faced by battery makers is squeezing a few more milliamp hours from cells through refinements in centuries-old technologies.

Trimming power use, however, is a software and silicon problem--areas in which breakthroughs occur regularly. If Intel's engineers had tried to create a Pentium 4 microprocessor with twenty-year-old technology, the result would put any space-heater to shame.

To gain a competitive advantage, each of today's major microprocessor manufacturers has developed its own power-saving strategy for its products. AMD calls its power-saving features PowerNow. Intel builds SpeedStep technology into all of its new mobile microprocessors. And Transmeta calls its technology LongRun.

Chip-makers make mighty claims for their power-saving technologies, and each claims its technique is best. Yeah, sure. We wanted to know the truth--does this black magic really work? Or is it all marketing mumbo-jumbo?

For some story background, in Fall 2001, we gathered sample systems using each of the three power management technologies: a Compaq Presario 1215, one of the first available systems using AMD's Mobile Athlon with PowerNow! technology; an IBM ThinkPad T21 using Intel's SpeedStep and a Mobile Pentium III; and a NEC Versa Ultralite, equipped with a Transmeta Crusoe 5800 chip using LongRun technology.

Our initial week-long project last Fall stretched into months as we were going back and forth with vendors and our own battery life benchmark developers at eTesting Labs on various battery testing issues. We'll admit we back-burned this story for a while, due to other priorities, but we did finally obtain trustworthy results recently. Even though these systems are a bit older than currently shipping notebooks, they are still quite representative of products deployed in quantity in the field today. Note that these particular notebooks are targeted to very different markets, thus our goal was not to compare battery life between the systems, but rather to see the effects of power management settings and technologies used within each system.

In this initial segment of our three part series on PC power management technology, we'll introduce you PC power management at a high level, provide some testing results of the different power management technologies, and review typical power consumption percentages of various notebook subsystems.

Power Management Overview

The goal in a power management system is to cut waste, minimizing the power needs of a circuit by making sure you only use what you need. For example, not all circuits in a component or computer need to be active all of the time. You don't need your modem when you're not on-line. Similarly, a microprocessor doesn't need to run its floating-point unit when it's not crunching numbers. Switching off unused circuits saves power.

Also, you don't need to use the full speed potential of a microprocessor when the program you're running is in a loop waiting for you to press the next key or, worse yet, doing nothing at all while you're staring off into space. As long as the computer or circuit responds to your next move in a reasonable time -- say a few milliseconds -- you'll be happy. Trimming clock speed dynamically while you're otherwise occupied thus becomes a viable power-saving strategy.

All computer manufacturers take advantage of these basic design principles to conserve power in their chip designs. No matter what a company calls its technology, it likely takes advantage of these same techniques in power-saving: voltage, speed, and selective shut-down. The chief difference between the cleverly named technologies is in how the power-saving features are controlled.

We found that power management can gain you as much battery life as moving up battery technologies from Nickel-Metal Hydride to Lithium Ion-- or possibly from Lithium Ion to something beyond today's chemists' dreams. More down-to-earth, power management gives you the same benefit as stuffing a pound or two more batteries into your notebook computer without any of the penalty.

On the other hand, as sophisticated and competitive as microprocessor power management strategies are, they are only bit players in determining how much extra life you can get from your system. Every milliwatt in your PC is managed in a modern notebook computer, and the power-savings come from every subsystem--microprocessor, memory, disk, screen, and support circuits. Moreover, the differences between systems in power saving are not as important as what you do. If you always demand full power, power management buys you nothing. Take a breather and give power management a chance to, well, manage your system's power, and each system will deliver substantial power savings. It's the combination of what you do and how your computer's power management works that determines how long your batteries will keep your system running.

Despite the ballyhoo chipmakers have accorded their power management systems in the last few years or so, neither the technology nor the underlying concepts are new. Through the years, engineers have developed a number of strategies for conserving power in computers, many of which have become enshrined as industry standards. They start by choosing the lowest-power technology consistent with the design goals of the equipment--trading off, for example, the last iota of power savings to make the gear affordable. Then they look to managing the power used by their designs to squeeze the most value from every what. The result has been several industry power management standards.

The industry made its initial stabs at unification with the Advanced Power Management (APM) specification released in the early 90s. Today, however, the power-saving standard bearer-and the reason you get great power management in most notebook PCs-is the Advanced Configuration and Power Interface (ACPI), which we'll cover in depth in Part II.

Note, however, there are incentives in saving and managing power beyond wringing longer runtimes from computer batteries. Lower power consumption also means circuits can be packed more densely, which helps in the design of higher-speed and more affordable chips.

While Part I focuses on a few interesting test results, later segments will dive into the ways circuit technology helps conserve power, industry standards for power management (APM and ACPI), and more details about SpeedStep, PowerNow, LongRun, and other techniques used to conserve system power.

Real-World Power Savings

Were power management to work perfectly, it would be invisible. Your notebook computer would run longer on a single battery charge without your being aware of any of the fancy footwork of proprietary microprocessor technologies, or even ACPI. When you're not looking--and not working--the power management features of your PC would swing into action and switch off every circuit of your system that is not in use. Look back, and it would pump performance back up to full speed.

Although the goal is laudable, finding whether the technology really works is an evaluator's nightmare. When it is working best, you should not be able to detect it working at all. When you can detect it, one or more of its strategies have failed.

If you always demand full power from your microprocessor, microprocessor power management would be meaningless. For example, set your computer deriving the exact value of pi or some other endless problem (for example, calculating your income tax), demand for your system's microprocessor never slacks, and the elaborate power reduction systems built into modern microprocessors never have a chance to take effect. Let your system sit idle, and it will soon switch off its screen, its disk, and itself. Sitting there as inert as some Congressmen, your computer would conserve the most battery power possible, but (also like some Congressmen) it would accomplish zero work.

These switch-it-off techniques do yield important power savings, but they are most valuable for the forgetful--those who accidentally leave their systems on and walk away (or drift away to dreamland). When these folks return to reality, they can pop their PCs back on and the batteries will still have some battery power left.

Power-saving technologies work best in the intermediate zone, when you need some processing power but not every last gigaflop. According to AMD, the best example of an application that falls squarely in this middle ground is viewing a DVD on a notebook computer screen. Although decoding the MPEG-2 video files on a DVD does tax your microprocessor (often with assistance from graphics cards), the processing power demand is moderate and consistent--which is what the proprietary microprocessor power-saving technologies handle best. The difference between a full-speed microprocessor and a power-managed microprocessor in playing back a DVD is either seeing the ending of a thriller, or getting left hanging on a cliff.

Test Results

Our latest Business Winstone 2001-based BatteryMark test attempts to reflect typical usage your laptop computer. Instead of keeping the system constantly occupied, the test gives the system some breathing room--all those pauses when you would be collecting your thoughts, pondering results, and preparing your next move. The test mixes together moments of pedal-to-the-floor processing such as during intensive CPU calculations, with think times during tasks such as word processing. The table below shows our BatteryMark results, run with power management disabled and enabled, and with display brightness set at minimum (Min), maximum (Max), and in-between (Med).

To repeat what we stated earlier-- these notebooks are targeted at different markets and have very different battery runtime characteristics, and our goal was not to compare battery life between the systems, but rather to see the effects of power management settings and technologies used within each system.

	PM Disabled Runtime (minutes)	**PM Enabled Runtime (minutes)**	**Improvement (PM Enabled / PM Disabled)**
Compaq Presario 1215 (AMD PowerNow!)			
Min	79	162	2.050633
Med	76	154	2.026316
Max	71	150	2.112676
IBM T21 (Intel SpeedStep)			
Min	108	200	1.851852
Med	94	189	*2.010638
Max	90	160	1.777778
NEC Versa Ultralite (Transmeta LongRun)			
Min	162	292	1.802469
Med	135	256	1.896296
Max	54	200	*3.703704

** We received repeated values in this range, as anomalous as they appear*

Surprise! Power management proved its mettle. In fact, we reaped power savings in excess of the chip-makers' claims for their power management technology. Compared to full-demand operation, power management alone was able to about double the runtime of our test systems.

Another surprise was the small difference between the highly touted power-management technologies.

The oldest, least severe microprocessor power management system (Intel's SpeedStep as implemented the T21) scored nearly identically to the more recent and supposedly more sophisticated Transmeta's LongRun. Actually, in our best analysis, SpeedStep came out about 5% better than LongRun. When we dimmed the screen as much as we could, the IBM SpeedStep-equipped ThinkPad gained about 85% from power management with a reduced processor load. The NEC Versa with LongRun gained 80%. The difference is well within the margin of error of our tests. AMD's PowerNow! did a bit better, more than doubling the runtime of the Compaq Presario, boosting runtime by 105%.

While we'd like to pronounce PowerNow! as the winner in percentage runtime gained, neither our results nor conclusions can be so straightforward. The microprocessor is not the only part of a notebook computer that is power-managed. And simply by altering the brightness of the screens downward, we extended runtime by as much as 46%. The effect of screen brightness adjustment varies with each system--mostly because each manufacturer puts a different range of brightnesses under your control.

Our conclusion, based on our limited testing, is that chip-makers' proprietary microprocessor power management technologies *should not* be your chief reason for choosing a particular computer. You can expect any of the three technologies to yield about the same results in typical PC use. The biggest difference we measured between the power management systems (about 15%) was swamped by other factors. And while the NEC had far and away the best battery life, that's attributable to an overall more frugal design, not any extraordinary power management abilities attributable to LongRun technology. If you refer to Nick's IDF Report, you'll see he summarized the developer session on notebook power management design at the recent Spring 2002 Intel Developer Forum, and how there are numerous design and configuration issues required to properly power-manage a notebook.

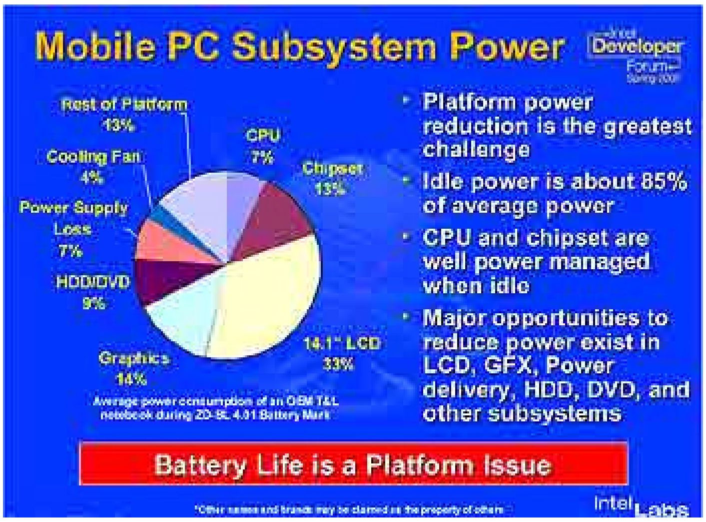
Our tests should give you new respect for what stringent power management can do for you. Although power management may be a small part of your life and your concerns about your PC, having any of the three flavors offered by the major chipmakers could be important if you watch a classic Hitchcock thriller and want to see the last minute in which the entire plot gets sorted out.

Notebook Subsystem Power Demands

The power demands of a notebook computer depend on its design purpose. Some desktop-replacement systems (especially new Mobile P4-class notebooks) might have hefty power demands, as high as 50 watts for certain intensive workloads, a level approaching that of a true desktop system. On the other hand, a small, power-conserving sub-notebook computer may consume on the order of 10-15 watts during normal operation.

The following pie chart shows the percentage of power utilization by various subsystems within a typical SpeedStep-enabled Mobile Pentium III system that dissipates around 12W of power on average, in total. Of course, your percentages may vary depending on your usage patterns, notebook design, and power management settings. The chart was included as part of Intel Labs Mobile Architecture Principal Engineer Guy Therien's recent IDF presentation titled "Extending Mobile PC Platform Battery Life". Guy said Intel Labs engineers used our older BatteryMark 4.01 test, but noted our newer Business Winstone BatteryMark yielded approximately the same percentages.

We recently received newer data from Intel Labs stating the Mobile P4 changes some of the data when running the same version of BatteryMark, as expected, since the Mobile P4 is a higher power component. The CPU goes from 7% to 10%, and the chipset goes from 13% to 15%.



Mobile PC Subsystem Power

Average power consumption of an OEM T&L notebook during ZD-BL 4.01 Battery Mark

- Platform power reduction is the greatest challenge
- Idle power is about 85% of average power
- CPU and chipset are well power managed when idle
- Major opportunities to reduce power exist in LCD, GFX, Power delivery, HDD, DVD, and other subsystems

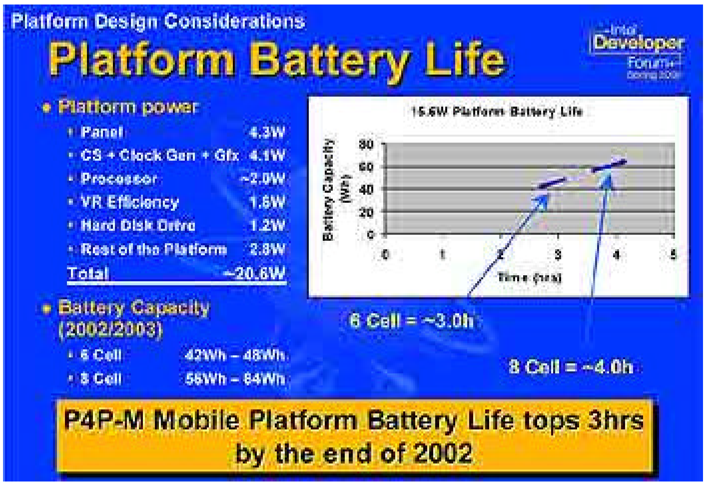
Battery Life is a Platform Issue

*Other names and brands may be claimed as the property of others

click on image for full view

The following chart was extracted from another presentation at IDF called "Mobile Intel Pentium 4 Processor-M Platform Design Considerations" by Manny Pitta, Technical Marketing Manager of Intel's Mobile Platforms Group, and it shows typical power consumption of various subsystem components in a P4 notebook in watts, plus estimated battery life with different battery configurations.

You should understand the difference between the oft-quoted TDP (Thermal Design Power) for a CPU and its normal power consumption. As the chart below indicates, the P4 processor consumes around 2 watts on average (which is related to mainstream business workloads per Intel), but TDP, which is often stated at typical (typ) or maximum (max), is much higher. TDP (typ) relates to worse case power dissipation running standard applications at nominal voltages under normal operating conditions, and TDP (max) is running worst-case instruction sequences that stress many parts of the processor at nominal voltages under normal operating conditions. The ambient temperature used to derive TDP is also a factor, and different vendors use different ambient temperatures. The P4 specs on p.85 of the new P4-M datasheet at http://developer.intel.com/design/mobile/datashts/mnwd_ds.pdf states TDP (typ) is approx. 30 watts. And they did not publish TDP (max), but the document states you should call your Intel Field Representative for further information…



Platform Design Considerations

Platform Battery Life

- Platform power

Panel	4.3W
CS + Clock Gen + Gfx	4.1W
Processor	~2.0W
VR Efficiency	1.6W
Hard Disk Drive	1.2W
Rest of the Platform	2.8W
Total	**~20.6W**

- Battery Capacity (2002/2003)

6 Cell	42Wh – 48Wh
8 Cell	56Wh – 64Wh

6 Cell = ~3.0h

8 Cell = ~4.0h

P4P-M Mobile Platform Battery Life tops 3hrs by the end of 2002

click on image for full view

As a preface to subsequent segments of our story, you can see from the above chart that it makes sense for engineers to approach power savings in each of the three major areas-microprocessor, display, and support circuits. They extend battery life with two basic strategies: designing circuits and components to use less power, and managing the power used by the devices. Long ago, each computer manufacturer used its own expertise to create proprietary power-saving technologies. Over the years, industry groups have developed standards to facilitate the development of software and hardware components that can be used in the widest possible variety of computers, with a common set of controls. In Part II we'll discuss circuit design strategies and power management standards APM and ACPI, and in Part III we'll dive much deeper into SpeedStep, LongRun, and PowerNow!, and other CPU power management techniques.

PC Power Management Technology

March 28, 2002
By: Winn Rosch

Designing Power Efficient Circuits

In Part I of this series we introduced you PC power management at a high level, provided some testing results of the different power management technologies, and reviewed typical power consumption percentages of various notebook subsystems. Here in Part II, we'll cover circuit design strategies and the APM and ACPI power management standards.

Let's begin by discussing the process of designing power management features into systems. The first bit of power management starts when the idea of a portable computer is just a glint in an engineer's eye. When engineers sit down at the drawing board, they start by choosing circuits that require the least amount of power to operate. By making the best design choices, the engineers can create more frugal computers (or any kind of electronic device), managing power use by requiring less of it from the very beginning.

The power required by an electronic circuit depends on a wide variety of factors, among them the basic semiconductor technology, the size of the circuits, the voltage at which the circuits operate, and the speed at which they run. Engineers have tinkered with all of these factors to reduce the power needs of their circuit creations.

Even though two modern microprocessor chips may look similar externally--little ceramic or plastic packages with gold legs on their undersides as seen in Pin-Grid Array (PGA) packages, or little solder balls on their underside as seen in Ball-grid Arrays (BGA) packaging--inside they may be as different as apples and orange-flavored castor oil. Current chips are often manufactured from different materials using different fabrication technologies and different arrangements of their internal components.

The different materials used for fabricating semiconductors affect power consumption. Exotic compounds such as gallium arsenide (GaAs) promise higher speeds and lower power than the silicon-based circuits that dominate the industry currently, but no mainstream chips use GaAs for a variety of technical and economic reasons. The recent move to copper interconnects (the fine wires that link circuit elements) from aluminum boosts speed and lower power losses.

Even the way that the transistors are connected on the sliver of silicon affects power consumption. At one time, N-channel Metal Oxide Semiconductors (NMOS) were the choice for circuit chips. NMOS circuits were fast, easy to design, and relatively inexpensive to make because each switch or logic gate required only one transistor. But NMOS has an inherent problem. To make a gate stay on, the single transistor must be switched constantly on, which means it continually draws power.

The more efficient design alternative is Complementary Metal Oxide Semiconductors (CMOS), which use two transistors per gate, arranged so that when one switches on, the other goes off (hence, they are *logical complements* of one another). Circuits are arranged with the two complementary transistors in series-- that is, so power must flow through both. Because one transistor blocks the flow through the other, current cannot pass through the pair. The only time much current flows is when the transistors change state. In effect, two transistors take the place of one to remember each bit (or make a logic decision) with the payoff that current (and thus power) consumption is drastically reduced. CMOS is more complex (requiring twice the transistors, for example), more expensive, inherently slower, and trickier to design. Click here to see an older yet still effective Java-based demo of CMOS circuit operation.

At first all microprocessors used NMOS designs. Engineers adopted CMOS for the first low-power microprocessors for portable computers in the early 1980's. Then, as the power used by chips and the heat they generated became a limit to their speed and processing ability, CMOS became the mainstream microprocessor technology. By 1987, nearly all new microprocessors used some variation of the CMOS technology, as do nearly all current microprocessor designs.

Circuit Size, Current, Voltage, and Speed

Circuit Size and Current

Size is an issue, and don't let anyone tell you anything different. Smaller circuits inherently need, use, and waste less power. The bigger the circuit, the more electrons there are to move to make it work, and the more power it takes to move them. It's the electronic equivalent of economy cars and sport-utility vehicles. The smaller, lighter cars require less horsepower to move and gulp less gasoline. Trim electronic circuits require less current, and that cuts their power needs.

Smaller logic gates require less current to change state. That saves power right off. In addition, waste in the form of heat is directly proportional to current, so trimming the current also trims the waste. Smaller circuits are naturally more energy frugal and efficient.

According to one of the fundamental principles of circuit design, Ohm's Law, the relationship between current and power is exponential. Power increases to the square of the current. As a result, cutting current usage in half reduces power consumption to one-quarter. Add in the reduction in heat generation, and size reduction becomes the most important factor in reducing the current needs of a microelectronic circuit.

Voltage

The voltage of an electronic circuit is the force that moves a charge or current around. It is also another factor that has a major impact on the power a computer or other electronic device consumes. As it does with current, Ohm's Law dictates that the power consumed by an electronic device increases exponentially as the voltage increases. Doubling the voltage quadruples the power consumption; trimming the voltage by half cuts consumption to one-quarter. Consequently voltage has been another area on which engineers have concentrated their power-saving efforts.

When the world, integrated circuits, and portable computers were young, most logic circuits operated at five volts--a compromise level that assured reliable operation without the need for dangerous and power-hungry higher voltages. Because the favored design for computer-like circuits at the time used a technology called Transistor-Transistor Logic, five volts was often called TTL Voltage. Not all circuits were happy with a mere five volts, however. For example, early CMOS designs wanted a full twelve volts to operate, and this voltage was sometimes called CMOS Voltage.

In the late 1980's, however, engineers tried to tame the growing power demands of high-speed circuits and the consequent heat the circuits generated by lowering the voltage standard from TTL's 5 volts down to 3.3 volts. The reduction was enough to cut power consumption by half. Because the most power-dense circuits would benefit most from the reduction, chips with the highest number of transistors--microprocessors and memory--were the first to be adapted to the lower voltage. Soon, however, engineers developed low-voltage equivalents of all circuit components.

Microprocessors, in particular, remain one of the prime candidates for voltage reduction because they are one of the prime consumers of power in computers. The latest trend in microprocessor design is to make their voltage requirement dynamic. That is, engineers have designed chips that operate at different voltages depending on how they are used. Low voltage saves power but reduces the speed potential of a circuit. All of the major proprietary microprocessor power-saving technologies--SpeedStep, PowerNow, and LongRun--include an element of dynamic voltage reduction.

Mobile microprocessors like the 750MHz Mobile Pentium III Processor-M can operate at an ultra-low .95 volts in battery optimized mode (350MHz operation). New technologies promise that microprocessors may one day commonly operate with less than a volt even in high-performance modes.

Speed

The faster a circuit operates, the more current it needs. It simply has to do more work moving more electrons around. Double the speed requires double the work. Each transistor in a logic circuit, for example, may require ten thousand electrons to change its state. Make it change twice in the same period, and you need to move double the electrons--which means double the current. As speeds go up, the demand for current naturally follows.

In addition, voltage and speed go hand-in-hand. Faster circuits often require higher voltage to operate. The higher voltage helps the electrons in the electronics move in and out of circuit elements faster. To move double the current through a circuit requires more force, more voltage.

Speed thus has a double-hit on power demand. Increasing the speed at which a circuit operates raises both its demand for current and voltage. For circuit designers, the corollary is more important. Reducing speed gives a double-hit in power savings--one hit from the speed reduction, then a second from a possible voltage reduction. The slower the circuit, the less power it will need.

Unfortunately humans have finite lives, so the prospects of really great power savings by extremely slow circuits aren't bright. Modern software and human impatience demand that circuits run ever faster. As a result, designers have resorted to trickery to shave computer speed and power usage. The major proprietary microprocessor power-saving technologies all cut microprocessor speed at the times that they think you won't notice it--or won't mind giving up some performance to gain battery life.

The Development of Standards

The history of portable computing has been mostly a matter of managing power with increasing precision and frugality.

The first portable computers did little to minimize their power use. After all, engineers had their hands full fitting all the circuitry of a desktop computer inside the paltry space available in laptop systems. The first efforts at power management were about the same as you'd institute in a time of rolling blackouts, switching off the lights when you don't need them. Manufacturers added timers to displays and disk drives to make them idle or switch off when the computer did not call for their use.

Microprocessors, the most power hungry of PC circuits, were among the first devices to gain built-in power management. An early development that appeared with Intel's 386SL microprocessor, System Management Mode (SMM) endowed processors with the ability to slow down and shut-off unnecessary circuits when they were idle. Similarly makers of hard disk drives added sleep modes to spin down their platters and reduce power needs. Most PCs also incorporated timers to darken their screens to further conserve power.

These primitive power control features were independent, but as small computers became popular and chipmakers developed dedicated support circuits, control of them became centralized. To provide a more consistent interface between power management facilities and software control, Intel and Microsoft jointly developed the Advanced Power Management (APM) system in 1992. Most of the APM functions were implemented through the system BIOS with loose links to the operating system.

In 1996, a new interface specification called ACPI (Advanced Configuration and Power Interface) emerged to allow operating systems to directly control and modify motherboard device settings, including power management functionality. Over the past few years, most computers have supported ACPI at the hardware and BIOS levels, but only until Windows 98SE and Windows 2000 did full operating system support for ACPI actually occur.

APM Overview

Every modern computer, whether a desktop or notebook model, has a wealth of power-saving features and functions, from multi-speed microprocessors and fans to stoppable disks and dimmable screens. If every computer maker went its own way in the design and control of these features, you the user would be as frustrated as a carpenter who forgot the key to his tool chest—that is, there would be so much that you would be able to do, if only you had access to your tools. Programmers could not hope to craft their products to match a nearly endless variety of controls and commands—maybe your operating system could track your keystrokes and keep the screen from darkening while you work, but maybe not. You likely would have little real control over the power features of your computer. That's not to say it might not have built-in power features, but you would be locked into whatever the manufacturer chose. Heaven help you if the manufacturer tried to make things look good on the spec sheet and slowed everything to a near stop every time you unplugged your notebook PC.

Helping you avoid such a straitjacket is the basic concept behind the Advanced Power Management (APM) standard, and the current ACPI standard, which is built upon its foundation. APM technology is at the core of ACPI, and its specifications are incorporated into ACPI.

In essence, APM takes all the good ideas that computer and component makers have created and provides a common, industry-wide control system for them. That way system designers and software writers would have a common standard that would allow them to make their products work with the widest variety of equipment. All the power savings features of a computer would have a single common access method, and the channel through which they are addressed would be the system Basic Input/Output System or BIOS. After all, every computer had a BIOS, and all BIOSs already had to be compatible to assure you that your programs would work on whatever computer you chose.

BIOS control means two different things, both of which you'll find in any computer that supports APM, which means just about every computer sold today. Through the setup program that's part of your BIOS, you can control APM features—for example, setting the time your computer waits before it assumes you've died or otherwise lost interest in your work before shutting off the display screen. In addition, APM creates a BIOS interface that links your programs—and your operating system in particular—to the power control functions built into your PC.

As you might expect for a beast that links software to hardware, the original developers of APM were the leading software and computer hardware companies of the time, Microsoft and Intel. They jointly published the Advanced Power Management BIOS Interface Specification in January, 1992. In February, 1996, it was updated to the current version, 1.2. The APM version 1.2 specification is available at the Microsoft website.

The most significant changes in the update gave the operating system greater control over power management functions. For example, the revisions allowed the operating system to force the BIOS to wait for the operating system to prepare programs and drivers for the switch before switching to a lower power mode. In addition, the operating system can override a hardware request for shifting to suspend mode, such as when the hardware detects no user interaction but the operating system senses a program actively calculating.

Although nominally a BIOS interface, the APM specification describes a layered control system that controls PC components and related devices to reduce power consumption. APM sets up the BIOS interface and the software interface (the Application Program Interface or API) to access those functions. For the system to work, APM requires a compatible BIOS and hardware devices that recognize APM control. In addition, APM allows manufacturers to add automatic power management functions that are not controlled by your PC's software. For example, a hard disk drive may automatically power down after a given period without accesses and this would not require specific commands from your PC.

APM States

The basic problem faced by the designers of APM was what they should control. For example, they could specify an APM command to control the voltage supplied to the microprocessor, but different microprocessors require different voltages, defeating the purpose of a universal command system. Instead of controlling such intimate details of each computer, APM created a set of power-saving levels called states that apply to the overall operation of the computer. The APM specification describes each state and assigns control functions to it, but the manufacturer of the computer is left to decide exactly how to achieve power reduction in a given state, what features to switch off, what to slow down.

In general, a computer implementing APM power-saving is allowed to operate in any of five states categorized by power-saving and recovery time—that is, how long it takes the system to shift from one state to another. For example, one part of a low-power state may tell the system's hard disk to stop spinning its platters. Moving from that power-saving state to the computer's normal function will entail a substantial delay as the disk spins back up to speed.

The APM specification gives each of its five operating states as well as a sixth, non-controlled state, a specific state name, as follows:

- **Full On state** is full-power, non-managed operation. The system operates at full speed and full power without any management at all. The APM software is not in control, and no power savings can be achieved. A system without APM or with its APM features disabled operates in full on state.
- **APM Enabled state** makes the APM control system active, but does not activate any power savings. All devices run in their normal, full power consumption modes. The system is up and ready to do business.
- **APM Standby state** reduces power consumption by switching off devices or reducing their operating power. The system microprocessor may even stop. In this state, the computer system usually cannot process data, but its memory is kept alive and the status of all devices is preserved. When your activity or some other event requires system attention, the PC can rapidly shift from

standby to enabled state.

- **APM Suspend state** shifts the computer system to its maximum power savings mode—most devices in a system that follow the APM standard are switched off, and the microprocessor switches to its lowest power state with its clock turned off. Your PC becomes a vegetable, unable to do anything until it shifts to a higher state.
- **Hibernation** is a special implementation of suspend state that allows the system to be switched entirely off and still be restored to the point at which it entered suspend state. When entering suspend state, the system saves all of its operating parameters. In entering hibernation, the system copies memory and other status data to non-volatile storage such as hard disk, allowing you to switch off memory power. A system event can shift back to enabled state from suspend or hibernation, but changing modes from suspend to enabled takes substantially longer than from standby to enabled.
- **Off state** is exactly what the name implies. Power to the system is entirely off. The computer is more a mineral than vegetable. The only event that restores the system is turning it back on. If you enter off state directly—say by switching your PC off—no status information or memory gets saved. The system must run through the entire boot up process and starts with a clean slate.

APM Structure

Although APM works at the BIOS level, the specification embraces much more than a few bytes of BIOS code. The standard adds a complete layered control system that gives you, your operating system, and your computer hardware a mechanism to shift states manually or automatically.

The bottom layer of the system is the APM BIOS itself, which provides a common command system for controlling the power-saving hardware features of your computer. The APM specification requires that the BIOS in your PC have at least a real mode interface that uses interrupt 15(Hex) to control the power-saving features of the PC. Beyond that basic interface, the APM BIOS may also use 16- or 32-bit protected mode using entry points, that is, specific addresses for code functions. Because the APM BIOS is meant to manage the power of a specific computer's motherboard, its code is specific to that computer and motherboard.

The APM specification is not all-embracing. It allows engineers both to work around the APM BIOS and even avoid it entirely. Under the APM specification, the APM BIOS can operate independently of other APM layers to effect some degree of power saving in the system by itself. Your PC's operating system can switch off this internal BIOS APM control to manage system power itself, still using the APM BIOS interface functions to control hardware features.

APM Operation

The APM BIOS functions both as a self-contained unit and as a channel into the power features of your computer. The first of these functions relies on simple timers. For the second, APM provides a programmatic interface similar to traditional BIOS functionality.

The timers want to activate APM's power-saving features. They anxiously count down the seconds until they can cut off (or reduce) power to part of your PC. As soon as the countdown period completes, the APM code automatically swings into action and trims the power flow.

Every time you or your software interacts with the APM BIOS, however, the countdown timers' functions are thwarted. With your every interaction—essentially your every keystroke—the timers reset and start the count again. This countdown and reset function is APM's way of determining whether you've lost interest in what you're doing, taking a break, or lying on the floor in cardiac arrest. In any case, you probably won't miss having the full performance and feature array of your computer at your instant command, so APM can instruct the system to cut back on power or device speed without your missing anything.

APM has three basic sensors to control its automatic functions. It monitors the keyboard for your interaction; it checks the disk drives for their use; and it monitors the data stream to graphics memory, judging whether the display gets changed. Typically each sensor controls its own timer and power-saving resource. Nothing happening at the keyboard or on the screen can trigger switching off the screen's backlight (or the monitor of your green PC) and the hibernation or shutdown of the entire computer. Sensors can also control individual functions, such as powering down an inactive hard disk.

In addition, the APM system has a built-in fail-safe that takes effect if your system crashes or the operating system otherwise loses control. The APM driver must interact with the BIOS at least once per second. If it does not, after the next second, the BIOS assumes the operating system has malfunctioned and takes self-contained control. An operating system driver can regain control by sending the appropriate commands (interrupts) to the BIOS.

Certain system events termed wake-up calls tell the APM system to shift modes. Interrupts generated by events like a press of the resume button, or the modem detecting an incoming telephone ring, or an alarm set on the real time clock can command the APM BIOS to shift the system from suspend to enabled state.

APM Interface

To link the APM BIOS to your operating system, you must have APM Driver software that communicates with the APM BIOS. (When you install Windows on an APM computer, the Window setup procedure automatically handles APM driver installation.) The APM BIOS is system-specific and provides the link between the motherboard circuitry and the APM driver, so each motherboard does not require its own APM driver software. The APM driver provides a set of function calls to the operating system. The operating system requests an APM feature by sending a request to the driver, which translates the request into a function call to the system BIOS, which sends the proper software interrupt to the APM BIOS code. The APM driver is more than a mere translator. It interacts with both the BIOS and operating system. For example, the BIOS may generate its own request to power down the system, notifying the driver, and the driver then checks with the operating system to determine whether it should permit the power down. The APM driver also periodically polls the APM BIOS to see if any power management state changes are desired.

The APM specification allows two ways of accessing power control functions through the BIOS, through either the real mode of x86 processors or through protected mode. This bifurcation accommodates the needs of primitive computers that run nothing but real mode environments (like older DOS operating systems) as well as providing access during the boot-up process. A protected mode interconnection to the APM BIOS must initially be set up through real mode. In real mode, APM control uses extensions to the standard INT 15 software interrupt BIOS function call.

To access APM functions in real mode, the programmer sets the AH register prior to the INT 15 call to the value 53(Hex), identifying that the requested function is for APM. The value in the AL register defines the specific APM function to be carried out (see chart below). Other registers indicate which devices in the system (which essentially means the microprocessor or everything else) to affect and parameters of the command.

For example, to shift the system from On to APM Enable state, your operating system can issue software interrupt 15(Hex) with AH set at 53(Hex) and AL set at 08(Hex). The BX register identifies the devices to be affected and CX tells the BIOS whether to enable (set at 0001) or disable (set at 0000) power management. The following table lists the 17 functions defined for the APM BIOS that are set via the AH value using INT 15H.

APM real mode interrupt functions

AL value	Function
00(Hex)	APM installation check
01(Hex)	APM real mode interface connect
02(Hex)	APM protected mode connect 16 bit
03(Hex)	APM protected mode connect 32 bit
04(Hex)	APM interface disconnect
05(Hex)	CPU idle
06(Hex)	CPU busy
07(Hex)	Set power state
08(Hex)	Enable/disable power management
09(Hex)	Restore power-on defaults
0A(Hex)	Get power status

0B(Hex)	Get power managed event
0C(Hex)	Get power state
0D(Hex)	Enable/disable device management
0E(Hex)	APM driver version
0F(Hex)	Engage/disengage power management
10(Hex)	Get capabilities
11(Hex)	Get/set/disable resume timer
12(Hex)	Enable/disable resume on ring indicator
13(Hex)	Enable/disable timer based requests
80(Hex)	OEM APM function

By loading the BX register with an appropriate value, the driver or operating system can command an individual device, class of devices, or the entire APM system. Device classes include mass storage, the display system, serial ports, parallel ports, network adapters, and PC Card sockets.

When a protected mode interface is initialized with real mode function calls 02 (for 16-bit protected mode) or 03 (for 32-bit protected mode). The BIOS responds with the base address of the protected-mode APM BIOS entry points. The APM driver uses these entry points rather than software interrupts to issue APM function calls.

All implementations of APM must include a real-mode interface. The APM 1.2 specification makes a protected mode interface with support for both 16-bit and 32-bit modes mandatory.

Determining Device State using APM

To determine the state of devices in the system, the APM design requires that the BIOS be polled at the rate of once per second. The APM driver monitors the status of power managed events using the 0B(Hex) Get PM Event function, and the BIOS responds by sending an event code back to the driver in its BX register. Of course, several events might occur in the second between polls. To accommodate multiple events, the driver repeatedly polls the BIOS. The BIOS reports each event in sequence. The driver ceases its polling when the BIOS runs out of events to report. The table below lists APM power management events.

APM power management events

BX value	Event
0001(Hex)	System standby request
0002(Hex)	System suspend request
0003(Hex)	Normal resume system
0004(Hex)	Critical resume system
0005(Hex)	Battery low
0006(Hex)	Power status change
0007(Hex)	Update time
0008(Hex)	Critical system suspend
0009(Hex)	User system standby request
000A(Hex)	User system suspend request
000B(Hex)	System standby resume
000C(Hex)	Capabilities change
000D to 00FF(Hex)	Reserved system events
0100 to 01FF(Hex)	Reserved device events
0200 to 02FF(Hex)	OEM-Defined APM events

0300 to FFFF(Hex)	Reserved

The driver can take appropriate action on its own or relay the information it obtains to the operating system, which then makes its own judgment about what to do.

ACPI Overview

Power saving strategies far outstrip the limited capacity of the computer BIOS to hold control software. To eke the last milliwatt of efficiency from a system, each power-using device needs to be individually managed, a monumental task for a handful of code. At least that's the admitted rationale for the creation of the Advanced Configuration and Power Interface. At another level, ACPI seems to be Microsoft's way of saying, "Eeew! Don't touch that BIOS thing. Use the operating system for power management."

This new standard, developed jointly by Intel, Microsoft, and Toshiba (later joined by BIOS-maker Phoenix Technologies), does exactly that, shifting power management responsibilities from the BIOS (primarily APM) to the operating system. It both builds on the foundation of APM and supercedes it with the goal of putting the operating system into complete control of the PC power system.

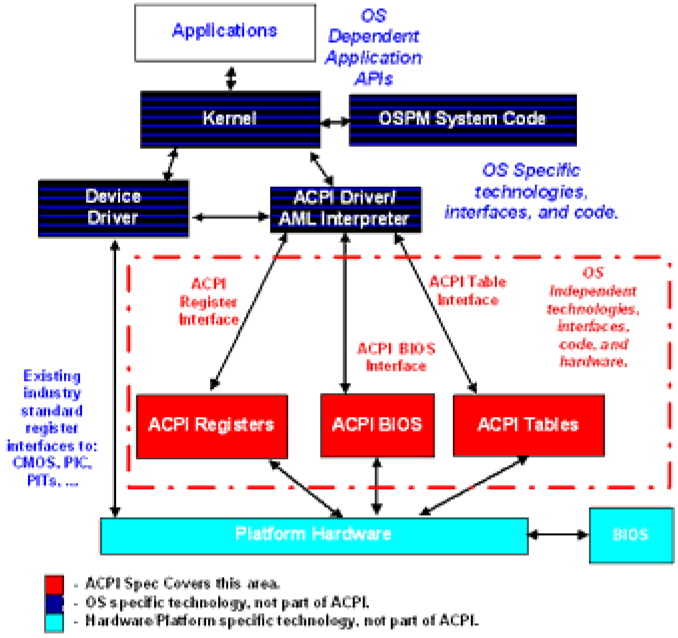
ACPI became an official standard with the release of Version 1.0 in December, 1996. The current version is Revision 2.0, first published on July 27, 2000 as a 450-page tome which includes all the details of the architecture and controlling its features.

As its name implies, ACPI is more than power management, however. Its configuration functions are the successors to the Plug-and-Play standard, endowing computers with almost automatic setup (but without the often-disparaged Plug-and-Play name). The pairing of power management and setup is natural. To effectively control power usage of PC components, the software needs to know what each component is, its needs, and its resource and interface requirements—exactly the same information the Plug-and-Play system gathered and tracked.

As a power management system, the ACPI specification can accommodate the needs of any operating system, integrating all the necessary power management features required in a PC from the application software down to the hardware level. It enables the operating system to automatically turn on and off and adjust the power consumption of nearly any peripheral, from hard disk drives to displays to printers. It can reach beyond the PC to other devices that may be connected into a single system some time in the future—televisions, stereos, VCRs, telephones, and even other appliances. Using the SmartBattery specification, under ACPI the operating system takes command of battery charging and monitoring. It also monitors the thermal operation of the system, reducing speed or shutting down a PC that should overheat.

The ACPI standard itself defines the interface for controlling device power and a means of identifying hardware features. The basic interface uses a set of five hardware registers that are controlled via a higher-level application programming interface (API) through the operating system. Its descriptive elements identify not only power management, but also device features through a set of nested tables. Its configuration function supplements Plug-and-Play technology, extending its existing structure with an architecture-independent implementation and replaces the Plug-and-Play BIOS with a new ACPI BIOS.

To drill home the aspect of the OS having more control over power management functions through ACPI, read this excerpt written by ExtremeTech's Nick Stam in the 3/7/2000 issue of PC Magazine-- "one of the heavily promoted features of ACPI is the ability to design very elaborate power management policies controlled by the operating system, that aren't possible at the BIOS level. The OS has much greater knowledge of system and application state than the BIOS, and thus should be able to control power usage much more effectively. But at the application level, there is little ACPI API support for power management control beyond requesting the screen stay lit (useful for presentations or streaming video). The notion you may have heard that applications are "ACPI-aware" and can actively provide the operating system with detailed device usage requirements via ACPI calls is a fallacy per Microsoft and Intel ACPI experts. They claimed the best way for an application to assist in power conservation is to not do dumb things, like continually polling a disk or network port for data."



click on image for full view

ACPI Global and Sleep States

Rather than interact with individual features of your hardware, ACPI operates on states, much as does APM. The ACPI specification defines states as levels of power usage and features availability, but leaves to the manufacturer of particular components or entire computers to determine what form the power savings takes. ACPI does not care that power-savings may reduce microprocessor speed by 100 megahertz and cut its supply voltage by a volt. ACPI sees such a measure only as shifting from an active state to a reduced-power state with a specific designation.

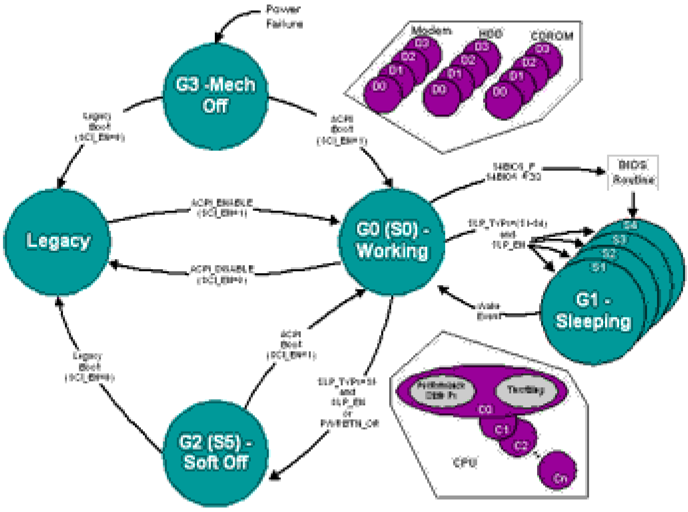
ACPI implements two main types of states called "global" and "device" states. The global states affect overall power consumption and response of your computer, and device states are concerned with individual devices such as hard disk drives and display panel backlights. The ACPI global states most closely correspond to the APM modes, and the device-level states are entirely new to ACPI.

The global states of ACPI divide power usage and system response into four levels, designated G0 through G3. Level G1 is further subdivided into sleep states, with its own hierarchy of response and power savings. The global states include the following:

- **G0** is working state in which the PC is operating normally. Programs execute actively. Even in G0 state, however, some devices that are inactive may automatically power down, but they will quickly resume normal operation when they are called upon.
- **G1** is sleeping state during which no computation visibly appears to be going on. When in the G1 state, any of a number of system events can cause the PC to return to working state. In the ACPI definition, G1 has many sub-states that are defined by the devices being managed. The standard allows great flexibility in this state; including how (and how quickly) normal operation is resumed. Within G1 are several special sleeping states that tradeoff resume speed for reduction in power consumption (see below).
- **G2** is soft-off state. In this mode, your PC acts like you've shut it off and requires rebooting to restart it. But it doesn't remove all power from the system. A slight bit of power continues to be supplied to the motherboard and expansion boards enabling them to monitor external events. For example, a network board will still listen to network traffic for packets targeted at it. A modem or fax board may lie in wait of a telephone call. Or you may set a wake up event, such as enabling a tape backup at midnight. When any of these designated external events occurs, the PC automatically switches itself back on to deal with it. This is the normal powered-down condition of a new PC. But in this state, your PC still needs to be plugged in, and it's really not safe to open the case and install expansion boards or memory.
- **G3** is complete power off, equivalent to unplugging the PC. It is the only state in which you can safely open your computer's case and replace components.

Within the G1 state, ACPI defines five levels of sleep from mere daydreaming to deep slumber. The defining differences concern the time required for the system to wake up and return to G0 state from its sleep. Any given computer need not implement all of these states. The complete list of sleep states includes the following:

- **S1** sleep state is more of a closing of eyes than even a catnap. All the registers and cache in the microprocessor, and all the bits in memory are preserved, so the system can pop back into action fairly quickly. The drives and display are powered off, but CPU, memory, and fans are active. This is a standby state.
- **S2** sleep state is also a standby state and meant to let the system switch quickly back on, but it does not preserve the contents of the cache or the microprocessor's registers. The microprocessor wakes up and resets itself, and retrieves its register values and cache contents from storage. The operating system is in charge of putting these things away so that they can be quickly recovered.
- **S3** sleep state lets everything slip away except for the contents of system memory. It's a "suspend to RAM" state that permits the "Instantly Available PC". The microprocessor resets itself when it wakes up and must restore all other system parameters from storage.
- **S4** sleep state loses everything including the contents of memory. Before the system can return to normal operation, the microprocessor has to recover its settings, the cache, and everything in memory from storage (usually the system hard disk). This state corresponds to what many PC makers call "hibernate" mode.
- **S5** sleep state is the same thing as G2 soft off. The system must reboot to move from S5 back to G0 normal operation.



click on image for full view

ACPI Device States

Each device that comes under ACPI control holds the potential of being individually adjusted. This intimate level of control permits the ACPI operating system to switch off unnecessary components, for example, powering down the modem when you're not on-line. As it does for its global controls, ACPI defines four states for each device. In addition, ACPI reserves the microprocessor as a special case with its own device-level control states.

The meaning of the ACPI device modes varies with the device type. The states differ in four chief characteristics—the amount of power that the state saves over normal operation; how long is required to restore the device from the state to normal operation; how much of the operating context of the device is saved in entering the state; and what must be done to return the device back to normal operation. All four of the normal device states are designated with names beginning with the letter "D" (for Device) as follows:

- **D0** designates the fully on state at which the device operates at top speed, is fully responsive, and consumes the most power.
- **D1** saves power over the D0 state. How it achieve that goal depends on the type of device. In general, the device can quickly shift back to D0 state without needing to reset or losing data—for

example, dimming the screen backlight or shutting down the control circuitry of a hard disk while the platters continue to spin.

- **D2** further saves power over the D1 state, and is again device-specific. In general, the device becomes less responsive. It may need to reset itself or go through its power-on sequence to return to the D0 state—for example, stopping the spin of a hard disk so that it needs to spin up and recalibrate itself before use.
- **D3** corresponds to the power-off state. Electrical power is removed from the device, and the device does not function. It must go through its power-on sequence to begin operations again. Upon entering D3, none of its operating context gets saved. This achieves the greatest power savings but requires the longest restoration time.

Under ACPI, the microprocessor is a special device that has its own four operating states, the designations for which are each prefixed with the letter "C." These include:

- **C0** state designates the processor executing normally. This includes full speed or any reduced speed if the microprocessor can switch back to full speed without requiring any change in itself or the rest of the computer. In other words, the microprocessor simply switches speed without affecting its registers or memory and imposing no delay in execution.
- **C1** state puts the microprocessor in its halt state under command of the ACPI driver without affecting other aspect of its operation.
- **C2** state shifts the microprocessor to low power state and maintains the integrity of the memory caches. In a fully implemented ACPI system, the microprocessor will shift to this state if a bus master takes control of the system.
- **C3** state pushes the microprocessor down to low power state and does not maintain cache memory.

All devices do not need to implement all states defined by the ACPI specification. For example, a device may support only the D0, D1, and D3 modes. In addition, a device may operate in different performance levels within a single state if it can change between levels invisibly—that is, without affecting the execution of your programs.

ACPI 2.0 defines performance states within the D0 and C0 states in which the speed of operation of a device may be reduced without switching it off. Currently, the most important application for these performance states is microprocessor speed and voltage control. The various levels under the SpeedStep, PowerNow, and LongRun technologies are performance states under the ACPI definition. In any reduced speed mode, the microprocessor remains in C0 state but operates in a lower performance state.

ACPI 2.0 defines up to 17 performance states for any device, with P0 the highest performance state and P16 the lowest. No all devices will have a full 17 states. In fact, most devices other than microprocessors may have only a single performance state corresponding to its C0 or D0 state. Note that the performance states are different from the power control effected through SpeedStep, PowerNow, or LongRun. The ACPI states are designed for software control while the integral microprocessor power-saving technologies are automatic—the microprocessor shifts between them without the control of the operating system. The entire range of 32 PowerNow settings appears as a single ACPI state, for example.

ACPI User-Level Power Management Controls

Under ACPI the world changes, and things are not what they seem. Because of the variety of states, "On" may not mean that your system is fully on and operating at top performance, and "Off" may not be anything like a familiar off switch. Perhaps the most radical difference is that even the old-fashioned power button on the front of your computer doesn't necessarily turn your computer off. In systems equipped to handle ACPI, the "Power" button doesn't control power but sends a signal to the ACPI system that you want your system to switch off. ACPI is in ultimate control of the power and, even when you push the button, won't necessarily turn your computer completely off.

Pressing the front panel "Power" button puts an ACPI-compliant PC in a mode called soft off (state G2), or more likely, the button-press will initiate a process at the end of which your computer will be in the G2 state. For example, pressing the button can warn Windows you want to shutdown, and Windows will follow through with its elaborate shutdown hierarchy.

ACPI envisions that some manufacturers will also put a "Sleep" switch on the front panel in addition to the "Power" button. Pressing it will put the PC in a sleep mode that uses somewhat more power than soft off but allows the system to resume operation more quickly. Up to now, only a small but growing number of manufacturers have put such switches on their desktop systems, although similar hardware functions are available on many notebook computers.

Because the both the "Power" buttons and the "Sleep" switch only send signals to the operating system, they need not be mounted anywhere near the actual power supply of your computer, affording engineers a new degree of design freedom. A growing trend among them is to put "Sleep" and even "Power" buttons on the keyboard of desktop PCs, as seen in Macs.

The other major change in user controls afforded by ACPI moves the setup of your system's power management out of the BIOS and boot-up process and into the operating system itself. The Power Options adjustments in Control Panel of recent versions of Windows supercede the BIOS functions available through system setup. Make an adjustment through Windows, and Windows reached down into the BIOS and alters the settings there. But most systems still retain and expose BIOS power management controls to the user, which is necessary when running non-ACPI compliant operating systems. In fact, virtually all systems continue to support APM functionality.

Power Options Properties divides control into a number of tabs. The Power Schemes tabs most directly corresponds to the control afforded by BIOS setup functions. It allows you to adjust the various inactivity timers that activate power-down functions. The Advanced tab provides password protection when resuming operation from standby and allows you to define the function of the power switch. The Hibernate tab allows you to force your computer to save its operating state, including the entire contents of memory, to disk before shutting down so that you can resume your work exactly where you left off. The optional UPS tab lets you monitor and control an external (but properly connected) uninterruptible power system.

ACPI Configuration

To handle its configuration function, ACPI must manage a tremendous amount of data not only about the power needs and management capabilities of the system but also describing the features available for all of the devices connected to the system. ACPI stores this information in a hierarchy of tables.

The overall master table is called the Root System Description Table (RSDT). It has no fixed place in memory. Rather, upon booting up, the BIOS locates a pointer to the table during the memory scan that's part of the boot-up process. A Root System Descriptor Pointer (RSDP) is located in low memory space of a system. It provides the physical address of the RSDT. The RSDT itself is identified in memory because it starts with the signature "RSDT." Following the signature is an array of pointers that tell the operating system the location of other description tables that provide it with the information it needs about the standards defined on the current system and individual devices.

Engineers originally designed the RSDT for 32-bit addresses, which is inadequate for the latest microprocessors. With ACPI version 2.0, the specification adds 64-bit addressing capabilities through an Extended System Description Table, located via the RSDP also, but with the signature of XSDT. To maintain backward compatibility, the RSDT is maintained in new systems.

One of these tables located by the RSDT or XSDT is called the Fixed ACPI Description Table. In it the operating system finds the base address of the registers used for controlling the power management system. In addition, the Fixed ACPI Description Table also points to the Differentiated System Description Table which provides variable information about the design of the base system. Some of the entries in this table are Differentiated Definition Blocks, which can contain data about a device or even a program that sets up other structures and define new attributes. ACPI defines its own languages for programming these functions, compatible with the C programming language.

That wraps up Part II, and in the next segment we'll explore Intel, AMD, and Transmeta power management techniques in more detail.