

C H A P T E R 2

# OnNow and WDM for PC 97

This chapter introduces the PC 97 design issues related to the OnNow design initiative and Win32 Driver Model (WDM).

**Contents**

---

OnNow Design Initiative and PC Evolution. . . . .	16
ACPI for PC 97 . . . . .	17
Win32 Device Driver Architecture . . . . .	18
References for OnNow and WDM . . . . .	20

## OnNow Design Initiative and PC Evolution

For 1997 and beyond, platform advancements will focus in these areas:

- Ease of use. The combination of hardware, software, and operating system interfaces still make it difficult for a novice to walk up and effectively use a PC for the first time.
- Serviceability. This relates to the availability and simplicity of the PC for performing tasks that the customer may encounter in everyday life.
- Extensibility and accessibility. PC systems must be capable of easily incorporating new technologies, rather than becoming obsolete as new technologies are introduced. New capabilities must be as easy to implement in a PC system for the user as plugging in a power cord.

Although the industry has come far with Plug and Play and other recent advancements to make the PC platform easier to use, customers—and potential customers—want the PC to be even more convenient to use. To enable the PC to become a much more integral part of daily life both in the office and the home, the PC must be instantly available to answer the phone, display new e-mail, or browse the Internet, but appear to be off when not in use.

The goal of the OnNow initiative is to advance the PC platform to a high level of usability and robustness that will allow the PC to deliver these new capabilities. To meet the goals of this initiative, it is essential that all system components work together to ensure the PC operates as the user would expect.

The following describes how the OnNow PC platform is expected to work:

- The PC is ready for use immediately when the user presses the power button.
- The PC is perceived to be off when not in use but is still capable of responding to wakeup events. In this context, appearing off means that the PC seems to be off according to what the user hears and sees, but also that the user is confident that the PC is not consuming excess energy and that data and programs are reliably saved. Wakeup events might be triggered by a device receiving some input such as a phone ringing or a timed alarm set by software.
- Software adjusts its behavior when the PC's power state changes. The operating system and applications work together intelligently to operate the PC in accordance with the user's current needs and expectations and to deliver effective power management. For example, applications will not inadvertently keep the PC busy when it is unnecessary and instead will proactively participate in shutting down the PC to conserve energy and reduce noise.
- All devices connected to the PC or added by the user participate in the device power management scheme. Any new device can have its power state changed as system usage dictates.

To achieve this set of capabilities requires a comprehensive, system-wide approach that affects all aspects of the PC. New requirements for the operating system, applications, device drivers, and hardware must be incorporated into the PC system to deliver transparent power management and improved integration of components. The necessary changes are the following:

- **Enhanced core operating system functionality for power management.** In the OnNow architecture, the operating system assumes the central role of coordinating power management activities at all levels and takes responsibility for defining the power state transitions for the system. It is planned for both Windows 95 and Windows NT to incorporate these changes in 1997.
- **A unified device driver model for power management and Plug and Play.** This involves the creation of a device driver model that supports per-device power management and Plug and Play for both Windows and Windows NT. The Win32 Driver Model is designed to fill this need.
- **A new system interface for power management and Plug and Play.** A design is required for implementing operating system-directed power management and enabling per-device power management of devices on the system board. The design defined by the Advanced Configuration and Power Interface meets this need and also provides future extensibility and improved system integration.
- **Device and bus hardware power management advances.** In the same way that Plug and Play unified configuration mechanisms throughout the PC, OnNow provides a framework for defining power management interfaces and state definitions. For example, new power management specifications have been developed for PCI, USB, and IEEE 1394 buses.
- **An application architecture that integrates applications into power management.** Building on the existing Plug and Play mechanisms for applications, it is necessary to further improve the flow of control and information through the application interface so that the user's environment is much more smoothly integrated with the underlying PC capabilities.

## ACPI for PC 97

A key component of the OnNow design initiative is a new system board design based on the Advanced Configuration and Power Interface (ACPI) specification. ACPI defines a flexible and abstract hardware interface that enables a wide variety of PC systems to implement power and thermal management functions while still meeting the cost and feature requirements of the target market.

ACPI also provides device configuration and generic system event mechanisms for Plug and Play. ACPI therefore replaces the Plug and Play BIOS and unifies the power management interface with the Plug and Play interface. It also provides support for configurations and resource types that could not be supported before.

ACPI leverages the Plug and Play BIOS data structures while providing an implementation that is independent of the processor architecture and compatible with both Windows 95 and Windows NT.

For OnNow, ACPI enables the operating system to direct power management throughout the PC system, improving integration, increasing effectiveness, and simplifying implementation. System designers have the freedom to implement a wide range of solutions, from the very simple to the very aggressive, while still maintaining full operating system support under both Windows 95 and Windows NT.

Further, power management policies that might be too complicated to implement in hardware or a ROM BIOS can be implemented and supported in the operating system, allowing inexpensive power-managed hardware to support elaborate power management policies.

Finally, ACPI enables investment in power management as a mainstream feature of both PC system hardware and operating systems, while allowing the ongoing independent evolution of each.

This pervasive implementation of power management will enable applications to routinely implement support for these capabilities, introduce new OnNow features, and improve the effectiveness of power management in the PC. Because of the emphasis on convenience and the OnNow goals, and because of the need to enable power management in both Windows 95 and Windows NT, the PC 97 system designs are based on this important new standard.

## Win32 Device Driver Architecture

Currently, support for new buses and new classes of PC peripherals poses new challenges for driver development. New buses increase the number of devices and requirements for device drivers. Increasing integration of multifunction devices further complicates driver development. At the same time, low-latency interactive applications demand tighter integration of software and hardware. For 1997, each of these factors are accounted for in the evolution of the unified Win32 Driver Model for Windows 95 and Windows NT. WDM allows a single driver source (x86 binary) to support a new bus or a new device on both Windows 95 and Windows NT.

Key goals for WDM are to simplify driver development by providing the flexibility to support new and innovative hardware while also reducing the number and complexity of drivers that must be developed. WDM must also provide a common framework for Plug and Play and device power management. WDM is a key component to enable innovative device support that is simple and convenient to use.

To achieve these goals, WDM is based on a common set of services provided by the Windows NT I/O subsystem. WDM advances consist of a set of kernel extensions to support Plug and Play, device power management, and low-latency streaming I/O. In addition to common platform services and extensions, WDM implements a modular, layered class minidriver structure. Class drivers implement functional interfaces required to support a generic bus, protocol, or device class. The generic nature of class drivers offers the opportunity for standardization of logical device command sets, protocols, and bus interfaces necessary for code reusability. WDM support for standard class interfaces reduces the number and complexity of device drivers required for both Windows 95 and Windows NT.

Minidrivers allow extension of generic class drivers to support a specific device protocol or physical programming interface. For example, a minidriver can be implemented to extend an IEEE 1394 bus class driver to support a specific host controller programming interface. Minidrivers are easy to develop because they simply extend generic class driver interface functions. Although minidrivers are simple by design, the benefits of reusable minidriver modules might also be realized in support of standard device programming interfaces. An example of this is the USB host controller interfaces (OpenHCI or UHCI).

The flexibility and uniform interfaces of the modular WDM architecture allows the operating system to dynamically configure various driver modules to support a specific device. A typical driver stack might consist of generic device, protocol, and bus class drivers linked by protocol-specific and bus-specific minidrivers. For example, the operating system can configure a driver stack to support a camera whose commands are defined by the Imaging class and delivered by the Function Control Protocol (FCP) class to the IEEE 1394 bus class. This flexibility also makes it easy to support a multifunction device by simply implementing a minidriver to interface the multifunction hardware to several device class interfaces. The dynamic construction of WDM driver stacks is key to Plug and Play device support.

WDM services enable a common low-latency model for Windows NT and Windows 95. WDM offers multiple levels of execution priority including kernel mode and non-kernel mode threads, IRQ levels, and Deferred Procedure Calls (DPC). All WDM class and minidrivers execute as privileged threads (not interrupted by CPU scheduler) in kernel mode (ring 0). Thirty-two IRQ levels are available to prioritize hardware interrupt service. For each interrupt, a DPC can be queued to execute on completion of the IRQ service routine with interrupts enabled. DPCs enhance system response to interrupts by effectively reducing the time during which interrupts are disabled. For x86-based PC systems that use multiple processors, support under Windows NT for interrupts is based on the MultiProcessor Specification Version 1.4 from Intel.

For applications requiring life-like multimedia, WDM provides low latency interfaces for processing I/O streams in kernel mode. WDM streaming interfaces will be exposed through standard WDM class interfaces. With WDM, a media stream can be processed completely in kernel mode by one or more software filters and device drivers. WDM streaming enables direct hardware access for streaming I/O acceleration, eliminates latency associated with transitions between non-kernel mode and kernel mode, and also eliminates the need for intermediate I/O buffering.

To leverage advances offered by WDM, you are encouraged to build Plug and Play-compatible, power-managed input, audio, imaging, and storage peripherals for USB and IEEE 1394.

WDM drivers can coexist with existing Windows NT drivers on Windows NT and with existing Windows drivers on Windows 95. Existing Windows NT and Windows 95 drivers will continue to be supported, but without the benefits of WDM advances. The extensible WDM class drivers supplied by Microsoft provide the best alternative for new device support. Before beginning development of a new WDM class driver, hardware developers should consult Microsoft for information about support of the particular device class. Whenever possible, class drivers should be written once and extended for specific hardware interfaces by using WDM minidrivers.

## References for OnNow and WDM

This section lists some of the publications, services, and tools available to help build hardware that works with Windows operating systems.

Advanced Configuration and Power Interface (ACPI) specification

<http://www.teleport.com/~acpi/>

Power management specifications for device and bus classes

<http://www.microsoft.com/hwdev/onnow.htm>

WDM white papers and information

<http://www.microsoft.com/hwdev/pcfuture/wdm.htm>

Plug and Play specifications

<http://www.microsoft.com/hwdev/pnpspecs.htm>

Microsoft Device Driver Kits (DDKs) for Windows operating systems

Microsoft Developer Network (MSDN) Professional membership