Microsoft
Plug and Play Specification

Plug and Play Design Specification for IEEE 1394

Version 1.0b — October 17, 1997

This document is provided to support requirements defined in *PC 97 Hardware Design Guide* (Microsoft Press, 1996; ISBN 1-57231-381-1), which refers to this document to define the Plug and Play requirements for PC 97. This document also supplements requirements defined in *PC 98 System Design Guide* (co-authored by Microsoft Corporation and Intel Corporation; Microsoft Press, 1997; ISBN 1-57231-621-7).

Contents

This specification is available at http://www.microsoft.com/hwdev/ specs/.

Revision Record		
Version	Date Published	Revision Comments
0.1	August 1996	Original Release.
1.0	March 1997	Final draft version 1.0
1.0a	June 1997	Update to PC Hardware Design Guide 1998, Version 0.9, Draft
1.0b	October 1997	Update in support of current industry specifications and additional Config ROM information.

Introduction

This specification is a design reference to aid consistent implementation and ensure interoperability of devices compliant with IEEE 1394–1995. The intent of this specification is to simplify device development through clarification of the requirements for devices and systems under the Microsoft® Windows NT® version 5.0 and Windows® 98 operating systems.

Design references span a range of topics including IEEE 1394 advances, device Configuration ROM, speed-independent and power-independent connectors and cabling, and Plug and Play details related to IEEE 1394 power specifications. This specification also outlines requirements for interoperability testing and provides guidelines for application of cable power.

All devices must comply with this specification to earn the “Designed for Microsoft Windows” logo. For information about the dates for logo compliance for specific IEEE 1394 requirements, please see the PC 97 and PC 98 FAQs, available at http://www.microsoft.com/hwdev/pc97.htm and http://www.microsoft.com/hwdev/pc98.htm, respectively.

IEEE 1394 Advances for 1998 Deployment

This section defines Plug and Play requirements related to the evolution of industry standards to achieve higher levels of system integration, performance, and ease of use.

1. Host controllers support 400 Mb/s PHY
To effectively support concurrent multi-streaming audio/video applications, 400-Mb/s bus bandwidth is required. At 400 Mb/s, many of the applications targeted for IEEE 1394 can coexist, enhancing the overall Plug and Play capabilities of the system. Most important, 400 Mb/s allows an application to use a lower percentage of available bus bandwidth to provide head room for system expansion. Speed traps (a slow device separating two faster devices) are also much less an issue at 400 Mb/s. Recommended design targets for devices with less than 400-Mb/s bandwidth are to limit peak bus utilization to less than 30 percent.

2. Host controllers are compliant with 1394 OpenHCI specification
The *1394 Open Host Controller Interface* (OpenHCI) specification defines standard hardware and software for PC connections to an IEEE 1394 bus. OpenHCI defines standard register addresses and functions, data structures, and direct memory access (DMA) models. The benefits of this standard are standard software support, improved performance, security, and error handling.

Availability of compliant host controllers is expected in mid-1998. Logo compliance dates are defined in relation to expected availability of host controllers.

3. PHY and Link use short arbitration reset as defined in IEEE 1394A
The advances in the IEEE 1394A specification enhance system performance and integration of components. Fast arbitrated reset reduces bus off-time following a bus reset condition to minimize interruption of isochronous audio-video (A/V) streams.

The IEEE 1394A performance enhancements should be adopted as they become available. Compliance with IEEE1394A will be required approximately 12 months after the specification is frozen for implementation.

4. Bus-management–capable devices support the abdicate bit as specified in IEEE 1394A
All bus-manager nodes shall implement the abdicate bit in the STATE_SET register as specified in IEEE P1394A Draft 1.0 or later.

Plug and Play for IEEE 1394 Devices

This section summarizes the Plug and Play requirements for IEEE 1394 devices, including PC host controllers and PC peripherals.

5. Topology faults do not cause the bus to fail
Standard IEEE 1394 protocols have been defined to eliminate topology faults. However, to ensure correct implementation, the following items describe test criteria for industry compatibility workshops. In each case, connection or removal of a device must not stall the bus, but the faulting device might not function. The test criteria include the following:

- Safe removal. All devices that provide a front-panel power switch must signal the operating system in response to a local shut-down request (that is, hot unplugging) in order to allow safe removal. Safe removal requires that the end user monitor the PC bus manager's response to the request before removing the device.
- Detection and warning. A PC acting as bus manager must detect topology faults and provide a warning message to the user when situations occur when the bus is extended too far.
- Warnings for more than 63 devices on a local IEEE 1394 bus. If the 63-device limit is exceeded, the 64th and later devices will be assigned a physical ID of 63. The 64th device must be detected by the PC bus manager and must provide a warning message to the user.

Removable Media

This section defines the Plug and Play requirements for removable media.

6. Removable media devices support media status notification
Removable media devices must use an electronic switch to notify the PC in the event of media change requests. This is necessary in order for device applications to have the capability to lock, unlock, and eject media.

Device Configuration ROM

This section defines the Plug and Play requirements related to device configuration ROM.

7. Device provides a configuration ROM for unique device identification
For Plug and Play the device configuration ROM must provide configuration information as specified in the following standards, as appropriate:

- IEEE 1394-1995 standard
- ISO/IEC 13213:1994
- IEEE 1394A

The configuration ROM is required for unique detection of the device and is used by a PC to enumerate the bus and to load the correct device driver. Table 1 provides an example ROM that combines all the elements outlined in the requirements listed in this section.

Table 1. Example Configuration ROM (located at FFFF F000 0400)

Block	Offset	Description			
First Quadlet	400h	info_length 04h	CRC_length **h	ROM_CRC_value (calculated) **h	**h
Bus_Info Block	404h	'1394' in ASCII 31h	33h	39h	34h
	408h	m c i b p reserved	cyc_clk_acc	max_rec	reserved
	40Ch	node_vendor_id **h	**h	**h	chip_id-hi **h
	410h	chip_id-low **h	**h	**h	**h
Root Directory	414h	directory_length 00h	03h	directory_crc **h	**h
	418h	module_vendor_ID key 03h	**h	module_vendor_id **h	**h
	41Ch	node_ capabilities key 0Ch	**h	node_capabilities **h	**h
	420h	unit_directory key D1h	00h	unit_directory offset 00h	01h
Unit Directory	424h	unit_directory_length 00h	03h	unit_directory_crc **h	**h
	428h	unit_spec_id key 12h	**h	unit_spec_id **h	**h
	42Ch	unit_sw_version key 13h	**h	unit_sw_version **h	**h
	430h	unit_dep_dir key D4h	00h	unit_dep_dir offset 00h	01h
Unit Dependent Directory	434h	unit_dep_dir length 00h	04h	unit_dep_dir_crc **h	**h
	438h	vendor_offset key 81h	00h	vendor key offset 00h	04h
	43Ch	vendor_offset key 81h	00h	vendor key offset 00h	0Bh
	440h	model_offset key 82h	00h	model key offset 00h	10h
	444h	model_offset key 82h	00h	model key offset 00h	15h

Note: "**h" indicates information that is filled in by the vendor.

Table 1, *continued*

Block	Offset	Description			
Vendor Leaf (Unicode)	448h	vendor leaf length		vendor_leaf_crc	
		00h	07h	**h	**h
	44Ch	vendor spec_id			
		80h	00h	00h	00h
	450h	vendor language_id			
		00h	00h	04h	09h
	454h	vendor text			
		4Dh	00h	69h	00h
	458h	vendor text			
		63h	00h	72h	00h
	45Ch	vendor text			
		6Fh	00h	73h	00h
	460h	vendor text			
		6Fh	00h	66h	00h
	464h	vendor text			
		74h	00h	00h	00h
Vendor Leaf (ascii)	468h	vendor leaf length		vendor_leaf_crc	
		00h	05h	**h	**h
	46Ch	vendor spec_id			
		00h	00h	00h	00h
	470h	vendor language_id			
		00h	00h	00h	00h
	474h	vendor text			
		4Dh	69h	63h	72h
	478h	vendor text			
		6Fh	73h	6Fh	66h
	47Ch	vendor text			
		74h	00h	00h	00h
Model Leaf (Unicode)	480h	model leaf length		model_leaf_crc	
		00h	05h	**h	**h
	484h	model spec_id			
		80h	00h	00h	00h
	488h	model language_id			
		00h	00h	04h	09h
	48Ch	model text			
		31h	00h	33h	00h
	490h	model text			
		39h	00h	34h	00h
	494h	model text			
		00h	00h	00h	00h
Model Leaf (ascii)	498h	model leaf length		model_leaf_crc	
		00h	04h	**h	**h
	49Ch	model spec_id			
		00h	00h	00h	00h
	4A0h	model language_id			
		00h	00h	00h	00h
	4A4h	model text			
		31h	33h	39h	34h
	4A8h	model text			
		00h	00h	00h	00h

Note: "**h" indicates information that is filled in by the vendor.

8. Device configuration ROM implements the general ROM format
The general configuration ROM format is specified in the IEEE 1394-1995 and ISO/IEC 13213:1994 standards. The general ROM format is an extensible tree structure that enables a managed environment by providing node-specific and unit-specific information as required for Plug and Play, power management, and isochronous data transfers. The general format also provides for definition of multifunction device units. The bus information block and root directory of the general ROM format are required as specified in Table 1.

9. Bus information block implemented at a base address offset of 0404h
The format of the bus information block is defined by the IEEE 1394-1995 standard. Following is a brief description of the required fields.

The first quadlet of the bus information block at offset 404h is the configuration ROM signature field used to identify an IEEE 1394 configuration ROM. This quadlet must contain the ASCII string "**1394**".

The second quadlet of the bus information block at offset 408h contains several bits that indicate node capabilities. These bits are defined as shown in the following list, together with their required values.

Bit or field	Table 1 symbol	Value and description
Irmc bit	m	Indicates the node supports isochronous resource manager capabilities.
Cmc bit	c	Must be 1 if the node supports cycle master capabilities; otherwise, this value must be 0.
isc bit	i	Must be 1 if the node supports isochronous operations; otherwise, this value must be 0.
Bmc bit	b	Indicates the node supports bus manager capabilities.
Pmc bit	p	Indicates the node is power-manager capable. The *pmc* bit is not defined by the IEEE 1394-1995 standard and is an extension created by this specification.
Cyc_clk_acc field	—	Specifies the accuracy of the node's cycle master clock in parts per million. If the *cmc* bit is 1, the field's value must be between 0 and 100. If the *cmc* bit is 0, this field must be all ones.
Max_rec field	—	Defines the maximum payload size of a block-write transaction addressed to the node. The range of the maximum payload size is from 4 to 2048 bytes. A *max_rec* value of 0 indicates that the maximum payload size is not specified. Otherwise, within the range of defined payload sizes, the maximum size is equal to $2^{max\_rec + 1}$. The *max_rec* field does not place any limits on the maximum payload size in asynchronous data packets—either requests or responses—that the node might transmit.

10. Configuration ROM provides globally unique device ID
The third and fourth quadlets of the bus information block of the configuration ROM must provide a globally unique device ID, which appears in Table 1 beginning at offset 40Ch. This unique 64-bit node ID is the only way to recognize the presence of a given device, because the physical device addresses can change following a bus reset. The unique ID is required for device detection and PC device driver loading.

If multiple units are supported by a bus node, then the unique 64-bit ID must not be referential to any one unit directory to allow for unique identification of a unit in a multifunction device.

The globally unique device ID in the bus information block must be invariant when read with quadlet read requests. That is, it must not be alterable in any way by software.

11. Root directory is located at a fixed address after the bus information block
The root directory must be located at a fixed address following the bus information block. For example, the root directory shown in Table 1 is fixed at offset 414h. All other directories and leaves are addressed by entries in their parent directories starting with the root directory.

12. Configuration ROM includes a unit directory for each independent device function
A unit directory is required for independent function and control of each device unit. A valid pointer to a unit directory must be provided in the root directory, as shown in Table 1, in compliance with the general ROM format specified in IEEE 1394-1995 and the directory format specified in ISO/IEC 13213:1994.

13. Each unit directory provides a valid Unit_Spec_Id and Unit_Sw_Version
Within a unit directory, the Unit_Spec_Id identifies the specification authority, and the Unit_Sw_Version identifies the particular document describing the unit. When it is added to the beginning of the Unit_Spec_Id, the Unit_Sw_Version uniquely identifies the unit's software interface.

14. Each unit directory provides a pointer to a unit-dependent directory
The unit-dependent leaf directory must provide additional information about the device unit's vendor and model in associated leaf directories. The format of the information contained in the vendor and model leaves is specific to the Unit_Spec_Id and the Unit_Sw_Version.

The valid pointer to a unit-dependent directory must be in accordance with the generic directory format specified in ISO/IEC 13213:1994. The unit-dependent directory must provide valid pointers to vendor and model leaves.

15. Vendor and model leaves support textual descriptor leaf format
Textual descriptors are required for each unit directory entries in the configuration ROM in order to display this information to the user. Textual descriptors are recommended for all other configuration ROM entries. Each textual descriptor points to a leaf that contains a single character string.

As an alternative, pointers to one or more textual descriptor leaves can be provided to support multiple languages. An example implementation is provided in Table 1, where the first textual descriptor leaf is implemented in Unicode and the following textual descriptor leaf in ASCII. Leaf format and textual descriptor leaves are specified in ISO/IEC 13213:1994.

Textual descriptor leaves must include the following:

- The spec_type field must be "0" to correspond to a 24-bit specifier_id for a standards body, or "1" to correspond to a 24-bit specifier_id for a defining vendor company_id.

  To operate using Windows NT-style language_IDs, the value for spec_type and specifier_id must be 0x80000000.
- The language_id field must be derived from the Windows NT locale number. Windows NT locale ID numbers are defined in Microsoft Developer Network (MSDN).
- Text string_info must be in ASCII for any spec_id in the range 0–7fffffff or in Unicode for any spec_id with the most significant bit set (for example, 0x80000000.

Plug and Play for Cabling and Connectors

This section defines the Plug and Play requirements for IEEE 1394 cabling and connectors.

16. Device provides three connector ports
Recommended: All devices should provide three 6-pin connector ports for optimum cabling options, subject to cable-power distribution constraints. Fewer than three ports promotes long daisy chains

with greater potential for speed traps (a slow device separating two faster devices). Therefore, three-port IEEE 1394 device nodes are recommended, with exceptions noted in the "Device uses standard 6-pin IEEE 1394 connector" requirement later in this section.

For internal-only devices, a minimum of two ports enables daisy-chaining of devices. However, a limit of 15 hops (end-to-end distance) limits total devices to 16, sufficient for most internal configurations.

Devices that consume cable power should be limited to a single connector to encourage short source-to-sink power delivery while eliminating the build up of voltage drop associated with a long daisy chain of power consumers.

17. Device uses standard 6-pin IEEE 1394 connector
A single connector eliminates unnecessary choices for the end user. For every *n* supported connector, there are 2^{n-1} cable choices. Two connector styles yield three end-user cable choices. Consistent use of the standard 6-pin IEEE 1394 connector eliminates an undesirable break in the power bus for power-dependent device applications.

Other benefits include volume pricing and consistent electrical performance. Therefore, all external pluggable IEEE 1394 devices must use the standard 6-pin IEEE 1394 connector. The exception is an option to use the 4-pin connector IEEE 1394A for miniature single-port (leaf-node) devices.

18. Standard 400-Mb/s–rated IEEE 1394 cable provided with devices
For Plug and Play, it is important to use one standard-performance cable for all device configurations to eliminate cable choices for the end user, because a mix of cable types and ratings creates an unfriendly user experience. This is especially important given the range of devices possible on an IEEE 1394 bus. Therefore, all cables must be have a minimum 400-Mb/s rating and, if bundled, must be shipped with a standard cable.

19. Devices power their PHY at all times
All devices must perform the bus repeater function when powered down, as specified in the IEEE 1394-1995 specification. Therefore, a device power switch must allow for local power to the PHY when switched off. Alternately, a device can implement a standard protocol to request cable power (if available) from the power manager. However, if a device provides local power to its PHY when it is switched off, then the device does not need to attempt to power its PHY from the cable when its AC connection is removed.

An exception to these requirements is necessary for battery-powered or mobile devices. An exception to these requirements is also necessary for PC add-on cards and system-board host connection devices that are subject to the power characteristics of the PCI bus.

A device that does not provide power to its PHY or consume power from the cable for its PHY will terminate the bus at the point of connection and must, therefore, terminate the pass-through of power.

20. Self-powered devices propagate the power bus through each connector
Self-powered devices must comply with *1394 Trade Association Specification for Power Management*.

21. Devices report power source and cable power consumption in Self_id packet
Self-powered devices must report zero power consumed in the power class field of the Self_id packet. Alternately, if a device consumes cable power only to keep its PHY alive, it must report this consumption in the Self_id packet. This allows the power manager to reserve power for this occasion.

22. Devices implement link power control
All cable-powered devices must implement the Link_on packet and Link_off bit in the State_Clear register. These controls allow a power-management–capable bus manager to control the node's power state. Access to the device configuration ROM must be possible following a Link_on. A device cannot increment its power consumption by more than 3 W following a Link_on. Cable-powered devices must rely on the power manager to enable their link.

This is only a recommendation for self-powered devices, which can power up with Link_on.

23. Device requiring power increments in excess of Link_on implements unit-power CSRs
All cable-powered and self-powered devices that require power increments in excess of Link_on power must implement standard unit-power CSRs as specified in *1394 Trade Association Specification for Power Management*. This is necessary to allow seamless integration of centralized power-management capabilities when a device is connected to a mini-system.

24. IEEE 1394–enabled PC sources cable power
An AC-powered PC must source cable power to the bus. Cable power in turn enhances Plug and Play with a single connection for low-cost cable-powered devices. Battery-powered mobile and notebook devices are exempt from this requirement. Minimum power wattage is defined in the following requirement.

25. Cable power source supplies a minimum of 20 volts at 15 watts
Recommended: To minimize the cost of a power source, actual power output can be reduced below the 40 V at 1.5 amps as specified in IEEE 1394-1995. Also, a cable power source should supply enough power for at least one cable-powered device (15 W) while also addressing voltage drop in the cable. Therefore, a minimum cable power source of 20 V direct current (DC) at a current limit of 1 amp is recommended. However, at the expense of higher component ratings, a 30 V cable power source will reduce power loss in the cable.

For example, a minimum 20 W output will ensure delivery of only 15 W to a load some distance away from the source device because of a cable voltage drop of 5 V—that is, 1 amp × .66 ohm × 7 to 8 cable hops separating source node from sink node at a rated cable hop resistance of 0.66 ohms. The voltage at the load will drop to 15 V with the source current limited at 1 amp. Therefore, a practical design target for a cable power source is a minimum of 20 V with a current limit of 1 amp.

A device such as a notebook that wants to source less than 20 V can do so if it reports in its Self_id packet that it does not source power, but does report in its configuration ROM the exact power it provides.

References for IEEE 1394

This section lists some of the publications, services, and tools available to help build hardware compliant with this specification.

IEC 1883 Digital Interface for Consumer Electronic Audio/Video Equipment
ISO/IEC 13213:1994
http://www.iec.ch

IEEE 1394 standards, publications, and organizations:
http://www.ieee.org (published standards can be ordered here)
ftp://ftp.symbios.com/pub/standards/io/1394/ (drafts of unpublished standards)
ASK*IEEE
Fax: (212) 310-4091
E-mail: askieee@ieee.org

Microsoft web site for hardware development:
http://www.microsoft.com/hwdev/

Open Host Controller Interface Specification
ftp://www.austin.ibm.com/pub/chrptech/1394ohci/

1394 Trade Association (for industry standard publications and events)
http://www.1394ta.org
E-mail reflector: 1394-sig@apple.com

1394 Trade Association Power Specification Part 1: Cable Power Distribution
http://www.1394ta.org

1394 Trade Association Specification for Power Management
http://www.p1394pm.org

SBP-2 and *SBP-2 Device Attachment*
ftp://ftp.symbios.com/pub/standards/io/x3t10 (drafts of unpublished standards)