PCI-SIG ENGINEERING CHANGE NOTIFICATION

TITLE:	Class Code & Capability ID Extraction
DATE:	April 28, 2010
AFFECTED DOCUMENT:	*PCI Local Bus Specification, Revision 3.0*
SPONSOR:	Joe Cowan; Hewlett-Packard Company

Part I

1. Summary of the Functional Changes

This ECN extracts the Class Code definitions from Appendix D and the Capability ID definitions from Appendix H, for consolidation into a new standalone document that's easier to maintain. The new document will also consolidate Extended Capability definitions from the PCIe Base spec and various other PCIe specs.

This new document is called the *PCI Code and ID Assignment Specification*, and it is separate from this ECN.

2. Benefits as a Result of the Changes

The new *PCI Code and ID Assignment Specification* will be more easily updated as new Class Codes, Capability IDs, and Extended Capability IDs are assigned over time. There is less chance of lost or duplicate assignments.

3. Assessment of the Impact

New specifications, ECNs, or Class Code requests that result in new Class Code or Capablity ID assignments should trigger updates to the new *PCI Code and ID Assignment Specification*.

4. Analysis of the Hardware Implications

None.

5. Analysis of the Software Implications

None.

6. Analysis of the C&I Test Implications

None.

Part II

Detailed Description of the changes

Change Section 6.2.1 as follows:

6.2.1. Device Identification

…

Class Code — The Class Code register is read-only and is used to identify the generic function of the device and, in some cases, a specific register-level programming interface. The register is broken into three byte-size fields. The upper byte (at offset 0Bh) is a base class code which broadly classifies the type of function the device performs. The middle byte (at offset 0Ah) is a sub-class code which identifies more specifically the function of the device. The lower byte (at offset 09h) identifies a specific register-level programming interface (if any) so that device independent software can interact with the device. Encodings for base class, sub-class, and programming interface are provided in ~~Appendix D~~ the *PCI Code and ID Assignment Specification*. All unspecified encodings are reserved.

Change Section 6.7 as follows:

6.7. Capabilities List

…

Each defined capability must have a SIG assigned ID code. These codes are assigned and handled much like the Class Codes. Refer to ~~Appendix H~~ the *PCI Code and ID Assignment Specification* for a list of currently defined Capabilities. Each Capability must define the detailed register map for that capability. These registers must immediately follow the pointer to the next capability.

Change Appendix D as follows:

D. Class Codes

The Class Code encodings defined in earlier versions of this specification have been moved to the *PCI Code and ID Assignment Specification*, the latest version of which can be found on the PCI-SIG website. Companies wishing to define a new encoding should contact the PCI-SIG. All unspecified values are reserved for PCI-SIG assignment.

~~This appendix describes the current Class Code encodings. This list may be enhanced at any time. The PCI-SIG web pages contain the latest version. Companies wishing to define a new encoding should contact the PCI-SIG. All unspecified values are reserved for PCI-SIG assignment.~~

~~**Base Class**~~	~~**Meaning**~~
~~00h~~	~~Device was built before Class Code definitions were finalized~~
~~01h~~	~~Mass storage controller~~
~~02h~~	~~Network controller~~
~~03h~~	~~Display controller~~
~~04h~~	~~Multimedia device~~
~~05h~~	~~Memory controller~~
~~06h~~	~~Bridge device~~
~~07h~~	~~Simple communication controllers~~
~~08h~~	~~Base system peripherals~~
~~09h~~	~~Input devices~~
~~0Ah~~	~~Docking stations~~
~~0Bh~~	~~Processors~~
~~0Ch~~	~~Serial bus controllers~~
~~0Dh~~	~~Wireless controller~~
~~0Eh~~	~~Intelligent I/O controllers~~
~~0Fh~~	~~Satellite communication controllers~~
~~10h~~	~~Encryption/Decryption controllers~~
~~11h~~	~~Data acquisition and signal processing controllers~~
~~12h - FEh~~	~~Reserved~~
~~FFh~~	~~Device does not fit in any defined classes~~

D.1. Base Class 00h

This base class is defined to provide backward compatibility for devices that were built before the Class Code field was defined. No new devices should use this value and existing devices should switch to a more appropriate value if possible.

For class codes with this base class value, there are two defined values for the remaining fields as shown in the table below. All other values are reserved.

Base Class	Sub-Class	Interface	Meaning
00h	00h	00h	All currently implemented devices except VGA-compatible devices
	01h	00h	VGA-compatible device

D.2. Base Class 01h

This base class is defined for all types of mass storage controllers. Several sub-class values are defined. The IDE controller class is the only one that has a specific register-level programming interface defined.

Base Class	Sub-Class	Interface	Meaning
01h	00h	00h	SCSI bus controller
	01h	xxh	IDE controller (see Note 1)
	02h	00h	Floppy disk controller
	03h	00h	IPI bus controller
	04h	00h	RAID controller
	05h	20h	ATA controller with ADMA interface – single stepping (see Note 2)
		30h	ATA controller with ADMA interface – continuous operation (see Note 2)
	06h	00h	Serial ATA controller - vendor specific interface
		01h	Serial ATA controller - AHCI 1.0 interface
	07h	00h	Serial Attached SCSI (SAS) controller
	80h	00h	Other mass storage controller

Notes:

1. Register interface conforms to the PCI Compatibility and PCI Native Mode Bus interface defined in *ANSI INCITS.370:2003: ATA Host Adapters Standard* (see http://www.incits.org and http://www.t13.org).
2. Register interface conforms to the ADMA interface defined in *ANSI INCITS.370:2003: ATA Host Adapters Standard* (see http://www.incits.org and http://www.t13.org).

D.3. Base Class 02h

This base class is defined for all types of network controllers. Several sub-class values are defined. There are no register-level programming interfaces defined.

Base Class	Sub-Class	Interface	Meaning
02h	00h	00h	Ethernet controller
	01h	00h	Token Ring controller
	02h	00h	FDDI controller
	03h	00h	ATM controller
	04h	00h	ISDN controller
	05h	00h	WorldFip controller
	06h	xxh (see below)	PICMG 2.14 Multi Computing
	80h	00h	Other network controller

For information on the use of this field see the PICMG 2.14 Multi Computing Specification (http://www.picmg.com).

D.4. Base Class 03h

This base class is defined for all types of display controllers. For VGA devices (Sub-Class 00h), the programming interface byte is divided into a bit field that identifies additional video controller compatibilities. A device can support multiple interfaces by using the bit map to indicate which interfaces are supported. For the XGA devices (Sub-Class 01h), only the standard XGA interface is defined. Sub-Class 02h is for controllers that have hardware support for 3D operations and are not VGA compatible.

Base Class	Sub-Class	Interface	Meaning
03h	00h	0000 0000b	VGA-compatible controller. Memory addresses 0A 0000h through 0B FFFFh. I/O addresses 3B0h to 3BBh and 3C0h to 3DFh and all aliases of these addresses.
		0000 0001b	8514-compatible controller -- 2E8h and its aliases, 2EAh-2EFh
	01h	00h	XGA controller
	02h	00h	3D controller
	80h	00h	Other display controller

D.5. Base Class 04h

This base class is defined for all types of multimedia devices. Several sub-class values are defined. There are no register-level programming interfaces defined.

Base Class	Sub-Class	Interface	Meaning
04h	00h	00h	Video device
	01h	00h	Audio device
	02h	00h	Computer telephony device
	80h	00h	Other multimedia device

D.6. Base Class 05h

This base class is defined for all types of memory controllers. Several sub-class values are defined. There are no register-level programming interfaces defined.

Base Class	Sub-Class	Interface	Meaning
05h	00h	00h	RAM
	01h	00h	Flash
	80h	00h	Other memory controller

D.7. Base Class 06h

This base class is defined for all types of bridge devices. A PCI bridge is any PCI device that maps PCI resources (memory or I/O) from one side of the device to the other. Several sub-class values are defined.

Base Class	Sub-Class	Interface	Meaning
06h	00h	00h	Host bridge
	01h	00h	ISA bridge
	02h	00h	EISA bridge
	03h	00h	MCA bridge
	04h	00h	PCI-to-PCI bridge
		01h	Subtractive Decode PCI-to-PCI bridge. This interface code identifies the PCI-to-PCI bridge as a device that supports subtractive decoding in addition to all the currently defined functions of a PCI-to-PCI bridge.
	05h	00h	PCMCIA bridge
	06h	00h	NuBus bridge
	07h	00h	CardBus bridge
	08h	xxh	RACEway bridge (see below)
	09h	40h	Semi-transparent PCI-to-PCI bridge with the primary PCI bus side facing the system host processor
		80h	Semi-transparent PCI-to-PCI bridge with the secondary PCI bus side facing the system host processor
	0Ah	00h	InfiniBand to PCI host bridge
	80h	00h	Other bridge device

RACEway is an ANSI standard (ANSI/VITA 5-1994) switching fabric. For the Programming Interface bits, [7:1] are reserved, read-only, and return zeros. Bit 0 defines the operation mode and is read-only:

0 - Transparent mode

1 - End-point mode

D.8. Base Class 07h

This base class is defined for all types of simple communications controllers. Several sub-class values are defined, some of these having specific well-known register-level programming interfaces.

Base Class	Sub-Class	Interface	Meaning
07h	00h	00h	Generic XT-compatible serial controller
		01h	16450-compatible serial controller
		02h	16550-compatible serial controller
		03h	16650-compatible serial controller
		04h	16750-compatible serial controller
		05h	16850-compatible serial controller
		06h	16950-compatible serial controller
	01h	00h	Parallel port
		01h	Bi-directional parallel port
		02h	ECP 1.X compliant parallel port
		03h	IEEE1284 controller
		FEh	IEEE1284 target device (not a controller)
	02h	00h	Multiport serial controller
	03h	00h	Generic modem
		01h	Hayes compatible modem, 16450-compatible interface (see below)
		02h	Hayes compatible modem, 16550-compatible interface (see below)
		03h	Hayes compatible modem, 16650-compatible interface (see below)
		04h	Hayes compatible modem, 16750-compatible interface (see below)
	04h	00h	GPIB (IEEE 488.1/2) controller
	05h	00h	Smart Card
	80h	00h	Other communications device

For Hayes-compatible modems, the first base address register (at offset 10h) maps the appropriate compatible (i.e., 16450, 16550, etc.) register set for the serial controller at the beginning of the mapped space. Note that these registers can be either memory or I/O mapped depending what kind of BAR is used.

D.9. Base Class 08h

This base class is defined for all types of generic system peripherals. Several sub-class values are defined, most of these having a specific well-known register-level programming interface.

Base Class	Sub-Class	Interface	Meaning
08h	00h	00h	Generic 8259 PIC
		01h	ISA PIC
		02h	EISA PIC
		10h	I/O APIC interrupt controller (see below)
		20h	I/O(x) APIC interrupt controller
	01h	00h	Generic 8237 DMA controller
		01h	ISA DMA controller
		02h	EISA DMA controller
	02h	00h	Generic 8254 system timer
		01h	ISA system timer
		02h	EISA system timers (two timers)
	03h	00h	Generic RTC controller
		01h	ISA RTC controller
	04h	00h	Generic PCI Hot-Plug controller
	05h	00h	SD Host controller
	80h	00h	Other system peripheral

For I/O APIC Interrupt Controller, the Base Address Register at offset 10h is used to request a minimum of 32 bytes of non-prefetchable memory. Two registers within that space are located at Base+00h (I/O Select Register) and Base+10h (I/O Window Register). For a full description of the use of these registers, refer to the data sheet for the Intel 8237EB in the 82420/82430 PCIset EISA Bridge Databook #290483-003.

D.10. Base Class 09h

This base class is defined for all types of input devices. Several sub-class values are defined. A register-level programming interface is defined for gameport controllers.

Base Class	Sub-Class	Interface	Meaning
09h	00h	00h	Keyboard controller
	01h	00h	Digitizer (pen)
	02h	00h	Mouse controller
	03h	00h	Scanner controller
	04h	00h	Gameport controller (generic)
		10h	Gameport controller (see below)
	80h	00h	Other input controller

A gameport controller with a Programming Interface == 10h indicates that any Base Address registers in this function that request/assign I/O address space, the registers in that I/O space conform to the standard 'legacy' game ports. The byte at offset 00h in an I/O region behaves as a legacy gameport interface where reads to the byte return joystick/gamepad information, and writes to the byte start the RC timer. The byte at offset 01h is an alias of the byte at offset 00h. All other bytes in an I/O region are unspecified and can be used in vendor unique ways.

D.11. Base Class 0Ah

This base class is defined for all types of docking stations. No specific register-level programming interfaces are defined.

Base Class	Sub-Class	Interface	Meaning
0Ah	00h	00h	Generic docking station
	80h	00h	Other type of docking station

D.12. Base Class 0Bh

This base class is defined for all types of processors. Several sub-class values are defined corresponding to different processor types or instruction sets. There are no specific register-level programming interfaces defined.

Base Class	Sub-Class	Interface	Meaning
0Bh	00h	00h	386
	01h	00h	486
	02h	00h	Pentium
	10h	00h	Alpha
	20h	00h	PowerPC
	30h	00h	MIPS
	40h	00h	Co-processor

D.13. Base Class 0Ch

This base class is defined for all types of serial bus controllers. Several sub-class values are defined. There are specific register-level programming interfaces defined for Universal Serial Bus controllers and IEEE 1394 controllers.

Base Class	Sub-Class	Interface	Meaning
0Ch	00	00h	IEEE 1394 (FireWire)
		10h	IEEE 1394 following the 1394 OpenHCI specification
	01h	00h	ACCESS.bus
	02h	00h	SSA
	03h	00h	Universal Serial Bus (USB) following the Universal Host Controller Specification
		10h	Universal Serial Bus (USB) following the Open Host Controller Specification
		20h	USB2 host controller following the Intel Enhanced Host Controller Interface
		80h	Universal Serial Bus with no specific programming interface
		FEh	USB device (not host controller)
	04h	00h	Fibre Channel
	05h	00h	SMBus (System Management Bus)
	06h	00h	InfiniBand
	07h (see Note 1 below)	00h	IPMI SMIC Interface
		01h	IPMI Kybd Controller Style Interface
		02h	IPMI Block Transfer Interface
	08h (see Note 2 below	00h	SERCOS Interface Standard (IEC 61491)
	09h	00h	CANbus

Notes:

1. The register interface definitions for the Intelligent Platform Management Interface (Sub-Class 07h) are in the IPMI specification.
2. There is no register level definition for the SERCOS Interface standard. For more information see IEC 61491.

D.14. Base Class 0Dh

This base class is defined for all types of wireless controllers. Several sub-class values are defined. There are no specific register-level programming interfaces defined.

Base Class	Sub-Class	Interface	Meaning
0Dh	00	00h	iRDA compatible controller
	01h	00h	Consumer IR controller
	10h	00h	RF controller
	11h	00h	Bluetooth
	12h	00h	Broadband
	20h	00h	Ethernet (802.11a – 5 GHz)
	21h	00h	Ethernet (802.11b – 2.4 GHz)
	80h	00h	Other type of wireless controller

D.15. Base Class 0Eh

This base class is defined for intelligent I/O controllers. The primary characteristic of this base class is that the I/O function provided follows some sort of generic definition for an I/O controller.

Base Class	Sub-Class	Interface	Meaning
0Eh	00	xxh	Intelligent I/O (I2O) Architecture Specification 1.0
		00h	Message FIFO at offset 040h

The specification for Intelligent I/O Architecture I/O can be downloaded from: ftp.intel.com/pub/IAL/i2o/.

D.16. Base Class 0Fh

This base class is defined for satellite communication controllers. Controllers of this type are used to communicate with satellites.

Base Class	Sub-Class	Interface	Meaning
0Fh	01h	00h	TV
	02h	00h	Audio
	03h	00h	Voice
	04h	00h	Data

D.17. Base Class 10h

This base class is defined for all types of encryption and decryption controllers. Several sub-class values are defined. There are no register-level interfaces defined.

Base Class	Sub-Class	Interface	Meaning
10h	00h	00h	Network and computing en/decryption
	10h	00h	Entertainment en/decryption
	80h	00h	Other en/decryption

D.18. Base Class 11h

This base class is defined for all types of data acquisition and signal processing controllers. Several sub-class values are defined. There are no register-level interfaces defined.

Base Class	Sub-Class	Interface	Meaning
11h	00h	00h	DPIO modules
	01h	00h	Performance counters
	10h	00h	Communications synchronization plus time and frequency test/measurement
	20h	00h	Management card
	80h	00h	Other data acquisition/signal processing controllers

Change Appendix H as follows:

H. Capability IDs

The Capability ID assignments defined in earlier versions of this specification have been moved to the *PCI Code and ID Assignment Specification*, the latest version of which can be found on the PCI-SIG website. Each defined capability must have a PCI SIG-assigned ID code. These codes are assigned and handled much like the Class Codes.

~~This appendix describes the current Capability IDs. Each defined capability must have a PCI SIG-assigned ID code. These codes are assigned and handled much like the Class Codes.~~

Section 6.7 of this specification provides a full description of the ~~Extended~~ Capabilities mechanism for PCI devices.

~~Table H-1: Capability IDs~~

~~ID~~	~~Capability~~
~~00h~~	~~Reserved~~
~~01h~~	~~PCI Power Management Interface – This capability structure provides a standard interface to control power management features in a PCI device. It is fully documented in the *PCI Power Management Interface Specification*. This document is available from the PCI SIG as described in Chapter 1 of this specification.~~
~~02h~~	~~AGP – This capability structure identifies a controller that is capable of using Accelerated Graphics Port features. Full documentation can be found in the *Accelerated Graphics Port Interface Specification*. This is available at http://www.agpforum.org.~~
~~03h~~	~~VPD – This capability structure identifies a device that supports Vital Product Data. Full documentation of this feature can be found in Section 6.4 and Appendix I of this specification.~~
~~04h~~	~~Slot Identification – This capability structure identifies a bridge that provides external expansion capabilities. Full documentation of this feature can be found in the *PCI to PCI Bridge Architecture Specification*. This document is available from the PCI SIG as described in Chapter 1 of this specification.~~
~~05h~~	~~Message Signaled Interrupts – This capability structure identifies a PCI function that can do message signaled interrupt delivery as defined in Section 6.8 of this specification.~~
~~06h~~	~~CompactPCI Hot Swap – This capability structure provides a standard interface to control and sense status within a device that supports Hot Swap insertion and extraction in a CompactPCI system. This capability is documented in the *CompactPCI Hot Swap Specification PICMG 2.1, R1.0* available at http://www.picmg.org.~~
~~07h~~	~~PCI-X – Refer to the *PCI-X Addendum to the PCI Local Bus Specification* for details.~~
~~08h~~	~~HyperTransport – This capability structure provides control and status for devices that implement HyperTransport Technology links. For details, refer to the HyperTransport I/O Link Specification available at http://www.hypertransport.org.~~

ID	Capability
09h	Vendor Specific – This ID allows device vendors to use the capability mechanism for vendor specific information. The layout of the information is vendor specific, except that the byte immediately following the "Next" pointer in the capability structure is defined to be a length field. This length field provides the number of bytes in the capability structure (including the ID and Next pointer bytes). An example vendor specific usage is a device that is configured in the final manufacturing steps as either a 32-bit or 64-bit PCI agent and the Vendor Specific capability structure tells the device driver which features the device supports.
0Ah	Debug port
0Bh	CompactPCI central resource control – Definition of this capability can be found in the *PICMG 2.13 Specification* (http://www.picmg.com).
0Ch	PCI Hot-Plug – This ID indicates that the associated device conforms to the Standard Hot-Plug Controller model.
0Dh	PCI Bridge Subsystem Vendor ID
0Eh	AGP 8x
0Fh	Secure Device
10h	PCI Express
11h	MSI-X – This ID identifies an optional extension to the basic MSI functionality.
12h-0FFh	Reserved