device
BAY

Device Bay Interface Specification

Compaq
Intel
Microsoft
Revision 0.90
November 6, 1998

INTELLECTUAL PROPERTY DISCLAIMER

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE.
NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED OR INTENDED HEREBY.
COMPAQ, INTEL, AND MICROSOFT, DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF PROPRIETARY RIGHTS, RELATING TO IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. COMPAQ, INTEL, AND MICROSOFT, DO NOT WARRANT OR REPRESENT THAT SUCH IMPLEMENTATION(S) WILL NOT INFRINGE SUCH RIGHTS.
THE DEVICE BAY INTERFACE SPECIFICATION HAS REACHED A LEVEL OF MATURITY SUITABLE FOR DEVICE DEVELOPMENT. THE AUTHORS OF THIS SPECIFICATION DO NOT BELIEVE THAT IT IS REASONABLE TO EXPECT THAT ALL PROBLEMS CAN BE DISCOVERED BEFORE IMPLEMENTATIONS ARE ATTEMPTED. IMPLEMENTORS ARE ENCOURAGED TO USE THE DEVICE BAY REFLECTORS, (device_bay@europa.com OR devicebay@acmenet.com), TO ASK QUESTIONS ABOUT PORTIONS OF THE SPECIFICATION THAT ARE NOT PERFECTLY CLEAR; TO POINT OUT INCONSISTENCIES; AND TO IDENTIFY AND PROPOSE FIXES TO ERRORS.

Contents

1 Introduction

Device Bay is an industry specification defining a mechanism for easily adding and upgrading PC peripheral devices without opening the chassis. The Device Bay specification applies to all classes of computers, explicitly including desktop, mobile, home, and server machines.

This specification defines characteristics of both peripheral devices and system bays. The Device Bay specification details:

- Mechanical form factors (DB32, DB20, and DB13)
- Power and thermal dissipation requirements
- Bus interfaces
- The connector set
- Device Bay Controller (DBC) logic
- Operating system support for Device Bay

Definition of these characteristics enables compatibility and interoperability between any Device Bay device and any Device Bay-enabled bay. (Notice that devices with a smaller Device Bay form factor require an adapter to be inserted into a larger Device Bay bay). Device Bay supports a wide variety of devices, explicitly focused at, but not limited to:

- Mass storage
- Communications and connectivity
- Security

Device Bay does not support processors or memory. Device Bay uses the following next-generation, industry-standard, extensible bus interfaces, which provide a broad range of bandwidths and scaleable performance, in order to support the requirements of PC peripherals for the next five to ten years:

- 1394, high-bandwidth serial bus, scaleable from 100 Mbps to 3200 Mbps data rates
- Universal Serial Bus (USB), a medium-bandwidth serial bus that operates at 1.5 Mbps or 12 Mbps

Device Bay is complementary to and coexists with USB and 1394 external connectors.

Device Bay bays can be built into any system that supports all specified bus interfaces, form factors, and thermal, mechanical, and electrical characteristics, and that provides the specified operating system behavior. Device Bay is applicable to all PC types and adds value in the following ways:

- Provides a simple path for user upgrade, expansion, and replacement, allowing PC peripheral devices to be added as easily as inserting a floppy disk into a drive
- Enables new form factors by using next-generation I/O interfaces, reducing the requirement to support legacy I/O such as Integrated Device Electronics (IDE) and Industry Standard Architecture (ISA)
- Provides an architecture that enables device security features

1.1 Desktop and Mobile Platforms

The overriding goal of this specification is to create a common interface for all platforms while addressing any unique aspects of desktop or mobile systems. The Device Bay specification will:

- Provide a means for interoperability between the form factors defined by the Device Bay specification.
- Create a common design specification wherever possible, especially in the areas of connector, electrical interface, and operating system support.

To guarantee interoperability between the Device Bay form factors, the following characteristics are shared:

- The connector pin-out and electrical definition are the same for all form factors (for more information, see section 4 of this specification).
- The connector form factor and presentation placement are the same for all form factors (for more information, see section 4 of this specification).

- The following mechanical design criteria are the same for all form factors (for more information, see section 5 of this specification):

- Guidance/alignment system for blind-mating tolerance control
- Multiple security recesses, allowing alternatives for implementing the required software-controlled device security interlock
- Mechanical protection
- Electrostatic discharge (ESD) protection
- Devices dimensioned for either sheet-metal or plastic enclosure
- Space in the bay reserved for non-compliant power connector area

The Device Bay specification is intended to minimize constraints on platform designers and device designers, thus maintaining design flexibility.

1.1.1 Unique Characteristics per Form Factor

To address the unique needs of the various Device Bay form factors, the following differences between DB32, DB20, and DB13 are defined in this specification:

- To accommodate a wide variety of device and system needs, each form factor’s size is different.
- The power and thermal thresholds are significantly higher for the DB32 device than for DB20 or DB13.
- The set of power and thermal alternatives is greater for the DB32 device than for DB20 or DB13.

For more information about power specifications, see section 3 of this specification. For more information about thermal specifications, see section 5.

1.2 Possible Uses of Device Bay

Possible uses of Device Bay include:

- Allowing original equipment manufacturers (OEMs), retailers, and end users to easily add the right peripherals to support specific application needs. For example, a CD-R drive could be added to provide a large storage media for digital imaging or audio authoring, a DVD-ROM drive could be added to enable DVD-Video playback, or a smart card reader could be added for secure online banking or shopping.

- Swapping a hard disk drive, and thus a set of data and applications, between a desktop system used in the office and a mobile system used on the road. Similarly, in the corporate environment, a hard disk drive could be removed from a failed system and inserted into a working system, minimizing employee downtime and thus lowering total cost of ownership.

1.3 Technical Overview of Device Bay

Figure 1-1 shows the relationship between a Device Bay device and a Device Bay bay on the system board. Important features include:

- The device interface is USB, 1394, or a combination of the two. A bridge is needed to interface with legacy devices. The legacy bridge, which enables existing devices to work with Device Bay, is on the device. Figure 1-1 shows an IDE-to-1394 bridge that enables legacy IDE storage devices to work with Device Bay.
- The host side controls identification power. Notice that the power field-effect-transistor (FET) or switch component shown in Figure 1-1 is on the system side of the Device Bay connector. This enables the instrumentation of the bay to control device power.
- The device controls its usage of the available operating power.

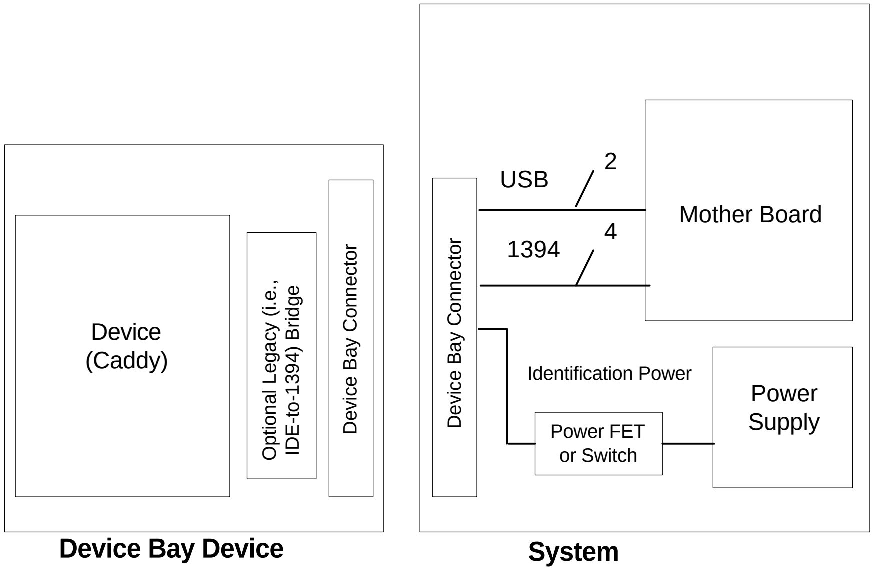

Figure 1-1. Relationship Between Device Bay Device and Bay

Figure 1-2 shows the device interface to Device Bay without the legacy bridge shown in Figure 1-1. Figure 1-2 could show a “native” 1394 bus device interface if the line labeled “Hot Plug Buses” was a 1394 bus. Notice that the Device Bay connector on the device side does not require support for both serial data transfer buses. If the device is exclusively a 1394 device, for example, USB support need not be provided.

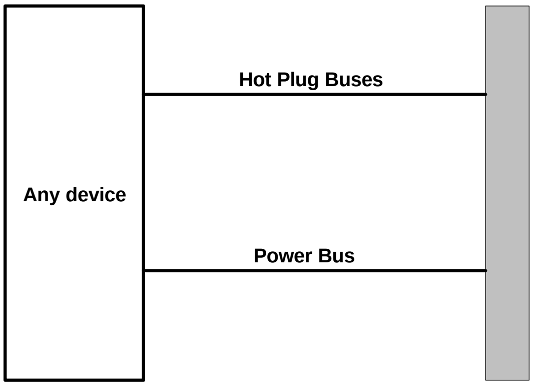

Figure 1-2. Device Interface to Device Bay

1.4 Definition of Terms

This section defines various terms used in this specification.

1394

For the purposes of this document, 1394 refers to IEEE 1394-1995 and 1394b. IEEE 1394-1995 is a hot-pluggable, high-speed serial bus that operates at speeds of 100 Mbps, 200 Mbps, and 400 Mbps. 1394b is expected to operate at speeds of 800 Mbps, 1600 Mbps, and 3200 Mbps. Device Bay requires the host system to support a minimum speed of 400 Mbps. 1394 can handle asynchronous and isochronous transfers. Typical devices include hard-disk drives, CD-ROMs, DVD-ROMs, tape drives, and consumer-electronics devices such as camcorders, VCRs, DVD players, and set-top boxes.

1394 PHY

1394 drivers and receivers.

ACPI Specification

Advanced Configuration and Power Interface Specification, Revision 1.0 or later. A specification that defines a new interface to the system board that enables the operating system to implement operating system–directed power management and system configuration. Following the ACPI specification allows system manufacturers to build systems consistent with the OnNow design initiative for instantly available PCs.

Bay

The receiving slot, dock, or cavity in the larger system that provides connectivity for the device. In this specification, all uses of the term *bay* refer to a Device Bay-compliant bay.

CRC

Cyclic redundancy check.

CSR

Control and status register.

DB13

A Device Bay device form factor that encompasses the global Device Bay device characteristics and also meets the unique power, connector, mechanical, and thermal requirements for the DB13 form factor (as defined in sections 3, 4, and 5 of this specification). One of the most obvious differentiating characteristics of a DB13 device is that it is approximately 13 mm high.

DB20

A Device Bay device form factor that encompasses the global Device Bay device characteristics and also meets the unique power, connector, mechanical, and thermal requirements for the DB20 form factor (as defined in sections 3, 4, and 5 of this specification). One of the most obvious differentiating characteristics of a DB20 device is that it is approximately 20 mm high.

DB32

A Device Bay device form factor the global Device Bay device characteristics and also meets the unique power, connector, mechanical, and thermal requirements for the DB32 form factor (as defined in sections 3, 4, and 5 of this specification). One of the most obvious differentiating characteristics of a DB32 device is that it is approximately 32 mm high.

DBC

Device Bay Controller. The DBC resides on the host side of the bay and contributes to the following processes: device insertion, device removal request, device enumeration, staged power consumption, and mapping USB and 1394 ports to bays. One of the primary benefits of the DBC is its role in discouraging the user from removing a device without first making a removal request. Removal requests enable the operating system to do whatever is necessary to ensure the integrity of user data, applications, and the operating system. A DBC can be implemented in a variety of ways: as part of a serial bus hub function, as a stand-alone device, or as an integral part of a system-board chip set. A DBC can interface to the system either as a USB device or as an ACPI object. Using ACPI, the DBC can reside on any bus in the system that can be described through ACPI. Examples are PCI, I^2C/SMBus, and embedded controllers.

DBPC

Device Bay Power Capabilities. 32-bit data entries that indicate the power that can be provided to the bays by each voltage present in a Device Bay subsystem, thus improving the power manageability of Device Bay.

Device

A modular component that is intended to be swapped, upgraded, or replaced on, between, or in a larger system. For Device Bay, this can be a device for mass storage, communications and connectivity, security, or a variety of other uses. For Device Bay, this does not mean a system processor (CPU) or memory. A Device Bay-compliant device meets the power, connector, mechanical, and thermal requirements in the Device Bay specification.

Device Bay

The overall concept of providing devices and bays under a standardized and interchangeable environment. Also, an industry specification that defines a mechanism for both peripheral devices and system bays that allows adding and upgrading PC peripheral devices without opening the chassis.

Device Bay connector set

The Device Bay connector set consists of a plug connector that resides on a removable device and a receptacle connector that resides in a bay. The Device Bay connector set has the following four pin groups: 1394 (interfaces to a 1394 PHY), USB (interfaces to a USB hub), power (interfaces to the power supply), and miscellaneous bay management signals (interfaces to the DBC). The Device Bay connector set specification ensures blind mating, live insertion and removal, and high insertion/removal cycle durability (a minimum of 2,500 cycles).

Device Bay subsystem

An implementation of the Device Bay concept. In a Device Bay subsystem, all bays are controlled by one DBC.

EMI/ESD

Electromagnetic interference and electrostatic discharge.

FET

Field-effect transistor. The current that flows through a FET is proportional to the voltage applied to its gate. FETs allow low power signals to switch high-power busses.

Gbps

Gigabits per second.

Mbps

Megabits per second.

PCA

Printed Circuit Assembly

USB

Universal Serial Bus. A bi-directional, isochronous, dynamically attachable serial interface for adding peripheral devices such as game controllers, serial and parallel ports, and input devices on a single bus. USB operates at 1.5 Mbps or 12 Mbps.

1.5 Related Documents

Specifications that must be used in conjunction with the Device Bay specification include:

- *P1394a Draft Standard for a High Performance Serial Bus (Supplement),* which is available at ftp://ftp.symbios.com/pub/standards/io/1394a/drafts
- *1394-1995 IEEE Standard for a High Performance Serial Bus,* which can be ordered from the IEEE Standards Catalog at http://www.ieee.org*/catalog/it.html*

- *1394 Power Management Spec, version 1.04, 6/10/97*, which is available at http://www.p1394pm.org, and currently exists as three separate documents, with the following revision numbers:

  Suspend/Resume Proposal Power Specification

  1394 Trade Association Power Specification, Part 1: Cable Power Distribution Revision 0.91x

  Power Management Draft Specification Revision 1.05
- *1394 Specification for Power State Management.* The latest version of this specification can be downloaded from ftp://ftp.zayante.com/ftp/pub/p1394b/pmonger.
- *Universal Serial Bus Specification, Revision 1.0,* which is available at http://www.usb.org.
- *Universal Serial Bus Device Bay Controller Specification, Revision 0.9, which is available at* http://www.usb.org/dwgdocs
- *Advanced Configuration and Power Interface Specification, Revision 1.0*, which is available at http://www.teleport.com/~acpi.
- *OnNow Power Management and USB*, which is available at http://www.microsoft.com/hwdev/pcfuture/usbdpm.htm
- *ASME Y14.5M-1994, Dimensioning and Tolerancing, Engineering Drawings and Related Documentation Practices, which is available from* the American Society of Mechanical Engineers (ASME), New York, NY, 1994.

The Plug and Play specification for legacy hardware platforms is the *Plug and Play BIOS Specification, Version 1.0a*, available at http://www.microsoft.com/hwdev/specs.
A description of the OnNow power management initiative is available at http://www.microsoft.com/hwdev, where the link “OnNow/ACPI: System-wide solution to power management control and device configuration” leads to the following technical papers:

- *OnNow: The Evolution of the PC Platform*
- *OnNow Power Management Architecture for Applications*
- *OnNow: Device Power Management*
- *OnNow Power Management and the Win32® Driver Model*

1.6 Organization of the Device Bay Specification, Revision 0.90

- Section 1. Introduction. An overview of the Device Bay interface and the interface specification.
- Section 2. Device Classes. Describes the device classes supported by the Device Bay interface and gives usability scenarios for each device class.
- Section 3. Buses. Describes the buses included in the Device Bay interface connector. The data transfer buses (USB and 1394) are the same for all form factors, but there are differences in the power bus for DB32 versus DB20 and DB13. These differences are specified in section 3.
- Section 4. Device Bay Connector Set Requirements. Describes the design goals and features of the Device Bay connector set, specifies the electrical and mechanical requirements for the connector set, and describes recommended implementation features of the connector set.
- Section 5. Mechanical Requirements. Specifies the Device Bay mechanical feature set, including form factors and thermal characteristics.
- Section 6. Device Bay Controller. Specifies the DBC registers and the DBC interfaces to ACPI and USB.
- Section 7. Software. Describes a Device Bay-capable operating system, presented in terms of the support required by an operating system from the mechanical, electrical, and DBC components of a Device Bay.
-
- Annex A. Legacy Drive Support. Describes how near-term legacy drive support is provided in the bays for the DB32, DB20, and DB13 devices.
- Annex B. Bay Status Indicator. Details a recommended method for implementing two-color light-emitting diode (LED) bay status indicators, as well as related DBC control signal requirements.
- Annex C. Modeling Device Bay Topologies with ACPI Objects. Provides several example ACPI name spaces that include a Device object for an APCI-based DBC, as well as sample ACPI Source Language (ASL) control methods based on one of these name spaces.
- Annex D. DBC Simple Link Controller. Describes the link controller needed to go into the DBC when the DBC is connected to a USB bus. Lists the required CSRs and Configuration ROM needed for the 1394 section of the DBC. Also discusses minimum requirements for the 1394 link controller as well as the extra functionality required for some “non-minimal” link controller implementations.
- Please see the Revision History at the back of the Specification, which tracks all changes from Revision 0.81 through 0.90.

2 Device Classes

Device Bay targets, but is not limited to, the following device categories:

- Mass storage
- Communications and connectivity
- Security

These device categories represent the intersection of end-user upgrade needs and the technical ability to implement the upgrades within reasonable host system cost adders. Additional device categories may be supported by the Device Bay infrastructure, although they are not explicitly named in this specification. Device Bay does *not* support the following device categories:

- CPU
- System memory

2.1 Benefits of Device Bay Features

Device Bay features benefit end users and the PC industry at large in a number of ways. These benefits are discussed in this section of the specification.

2.1.1 Benefits for Desktop and Mobile End Users

Device Bay bays are accessible. Users can easily upgrade, replace, and add devices without opening the PC case.

Devices inserted into Device Bay bays are automatically configured. The end user can immediately start using a newly inserted device. This is accomplished by a Plug and Play-capable operating system running on a platform that includes a Device Bay.

Devices can be inserted into, removed from, and swapped between Device Bay bays while the PC is powered up. This eliminates the time-consuming power-down and power-up cycles used by many of today’s systems. It also contributes to the end user being able to immediately use the new device.

Device Bay peripherals work properly when plugged in to any Device Bay bay. This enables the user not only to swap devices in any combination on one PC, but also to easily swap devices between different Device Bay-equipped PCs. Interoperability between platforms that provide bays is assured. Additionally, devices with a smaller form factor can be inserted into a larger Device Bay bay using an adapter.

A broad diversity of peripherals can be Device Bay devices. For example, consumer-electronics devices can be Device Bay devices, thereby expanding the usefulness of a PC.

PCs can be quickly and easily configured/reconfigured to a user’s specific application needs at any time. For example, a consumer may insert a DVD-ROM drive in the Device Bay bay in order to play a DVD-Video title. The same system can be reconfigured for consumer imaging by inserting a CD-R/E drive or DVD-RAM. Later, the user can replace the CD-R drive with a smart card reader in order to securely purchase an item over the Internet.

Future high-performance peripherals will work in Device Bays. As time goes by, earlier Device Bay-enabled host systems will accept future Device Bay peripheral devices. This ensured compatibility will enhance consumer confidence that Device Bay-enabled hardware investments will not become obsolete.

Security features can be accommodated by Device Bay. Device Bay has a software-controlled physical interlock that can enable security features.

Surprise removal of devices is guarded against. A Device Bay-enabled operating system uses a software-controlled interlock to make sure all applicable open files and applications are closed before the user is able to remove a device (for example, a hard disk drive) from a bay. This protects the current state of the user's data at all times.

Compound devices are supported. An individual device in a bay can concurrently use both USB and 1394 buses.

Device Bay resources, such as bandwidth and power, can be allocated. The Device Bay architecture supports the inclusion of a software agent that can manage the system resources as they are allocated to devices.

Device Bay provides deterministic hardware platforms. A Device Bay system maps devices to specific bays. This provides a deterministic model for the system to manage the insertion, removal, and operation of devices.

2.1.2 Benefits for OEMs

Device Bay simplifies the manufacturing process and system configuration. This allows OEMs to design and deliver customized systems while optimizing the manufacturing and distribution processes. Just-in-time, configure-to-order manufacturing processes can be put in place because the modular nature of Device Bay allows for cost-effective delivery of tailored PC configurations.

Device Bay simplifies product design and enables rapid adoption of new technologies into existing product lines, without altering either the PC system design or manufacturing processes. Using the standardized Device Bay form factors and device interfaces can reduce design lead time because the Device Bay specification addresses connectivity and interoperability design issues. The addition of new Device Bay-enabled peripherals can occur much later in the manufacturing pipeline than has previously been possible.

Device Bay reduces obsolescence issues for OEMs. Earlier peripheral devices can be used on evolving platforms and evolving peripherals can be used on platforms that have not changed.

Device Bay lowers support costs. A principal goal of the Device Bay specification is to reduce support calls and related costs due to improper installation of new peripherals. Device Bay design ensures that many current configuration conflicts will be eliminated.

2.1.3 Benefits for IHVs

Device Bay enables development of new product segments and enables faster integration of devices by OEMs into their platforms. Implementing new designs based on Device Bay will also mean more rapid adoption of new technologies once an installed base of Device Bay-enabled systems is present.

Device Bay provides for standardized design of device interfaces, connectors, and form factors. The enhanced interoperability that Device Bay ensures means that independent hardware vendors (IHVs) have a clear indication of what to build and can realize great economies of scale for connectors, casings, and other components.

2.1.4 Benefits for Operating System Vendors

Device Bay reduces the number of different software interfaces that have to be provided to support a wide variety of devices. Over time, various devices can be supported through just the two Device Bay data transfer buses, USB and 1394.

Surprise removal of devices is guarded against. Software controls the insertion and removal requests of Device Bay devices.

Device Bay provides deterministic hardware platforms. A Device Bay system maps devices to specific bays. This provides a deterministic model for the system to manage the insertion, removal, and operation of devices.

2.2 Device Categories

This section details some of the possible device categories supported by Device Bay.

2.2.1 Storage Devices

The need for removable, large-capacity storage devices is supported by Device Bay, where large capacity means from hundreds of megabytes to multiple Gigabytes.

Device Bay supports both removable and fixed-media storage devices. Examples of supported storage devices are:

- Hard disk drives
- Tape backups
- CD-ROMs, DVD-ROMs, tape drives, and consumer-electronics devices such as camcorders, VCRs, DVD players, and set-top boxes.

Device Bay support for these devices enables PC users to add extra storage, to backup files, to transport large files, and to have a portable, personalized environment. Rich multimedia content drives the need for larger, faster storage devices. This is reflected by the fact that storage is near the top of the list for consumer upgrades. Rich multimedia content has resulted in larger file sizes, rendering the traditional mobile storage media (the 1.44-MB floppy disk) nearly obsolete. New means of transporting large files, such as the 120MB+ HD floppy disk, are desired.

Device Bay-capable operating systems and BIOS enable the user to boot a PC from a Device Bay mass storage device.

Device Bay support for mass storage also enables users to place their personal data and environment on an easily transportable device, thereby enhancing mobility and system flexibility. In the corporate world, this would enable a user to exchange a mobile hard disk drive between a desktop and a mobile system for travel. The same device could be carried to another site and used in a virtual office/shared PC environment (“communal or community workstation”). Both of these scenarios lead to increased productivity and decreased employee difficulty during travel. Additionally, failed hardware can be serviced by removing the device to an operable PC, and then repairing the broken system offline. This approach minimizes employee downtime and therefore the total cost of ownership.

Device Bay support for mass storage also enables a home PC scenario where different members of the family have their own devices, which can be easily inserted and removed and even transported to a friend’s PC.

2.2.2 Communications and Connectivity Devices

Communications and connectivity devices enable the PC to interface with the outside world as well as to distribute data within the home or business. POTS modems, Integrated Service Digital Network (ISDN) adapters, network cards, cable interfaces, and wireless infrared (IR) and radio frequency (RF) devices all fall into this category. This device class is supported by Device Bay, enabling the PC to be more easily configured as the center of communications to, from, and within the home and business.

The presence of richer media types, such as graphics, video, and audio, on the Internet and Intranets is a driving factor for higher capacity interfaces into the PC. End-user desire for better and faster communications devices is underlined by the fact that modem cards were one of the most popular end-user PC upgrades in 1995. The ability to easily increase the capacity of the communications pipe into the PC in order to keep pace with rapid technological advances will improve the end-user experience and reduce fear of hardware obsolescence.

The PC must be able to communicate with other PCs and devices within the home or business; this enriches the end-user experience. Different applications and usage models drive different connectivity requirements. Due to the number of possible connections and usage models, supporting all user needs with a standard set of connectors is difficult and costly. Connections to printers, scanners, and other traditional PC peripherals may be supported.

As it becomes more cost-effective, this networking capability may move to multi-PC homes. The advent of digital media formats such as MPEG-2 video, Dolby AC-3 multichannel audio, and digital video (camcorders and VCRs) presents the opportunity to bring the computational power of the PC platform to bear on the consumer-electronics world, using digital connections such as 1394. Additionally, as the PC continues growing in the family-room space, the resulting shift in usage model will drive the demand for wireless I/O devices, both IR and RF. Device Bay provides a simple method of allowing OEMs and/or end users to add ports and connectors based on the specific application needs of a PC.

2.2.3 Data Security Devices

Device Bay supports security devices as a means to provide user authentication, useful both in the corporate world and for consumer applications such as home shopping. Simply adding a Device Bay smart card reader to a corporate PC or mobile system provides a high level of data security, even meeting requirements for PCs purchased by the U.S. government. For the consumer, inserting a smart card reader enables secure online credit-card shopping. Such lack of security has been cited by consumers as the primary reason why they have not made online purchases. Even higher levels of security can be achieved with encryption module security devices.

2.3 Non-Compliant Power Connector Area

Device Bay allows for devices that use the Device Bay bay, but are not compliant with Device Bay power requirements. An area is reserved for non-compliant power connectors. Devices that do not interface across 1394 or USB buses or that are not compatible with Device Bay power requirements are not addressed by the Device Bay specification. Support for these devices (such as batteries) can be provided by OEM-specific solutions using the defined connector areas. Any such OEM-specific solution is required to not interfere with standard Device Bay usage.

3 Buses

This section provides information about the following bus types supported by the bay connector (that is, the connector on the host side):

- USB – A medium bandwidth data bus. (1.5 Mbps to 12 Mbps)
- 1394 – A high bandwidth data bus (100 – 400+ Mbps)
- Power

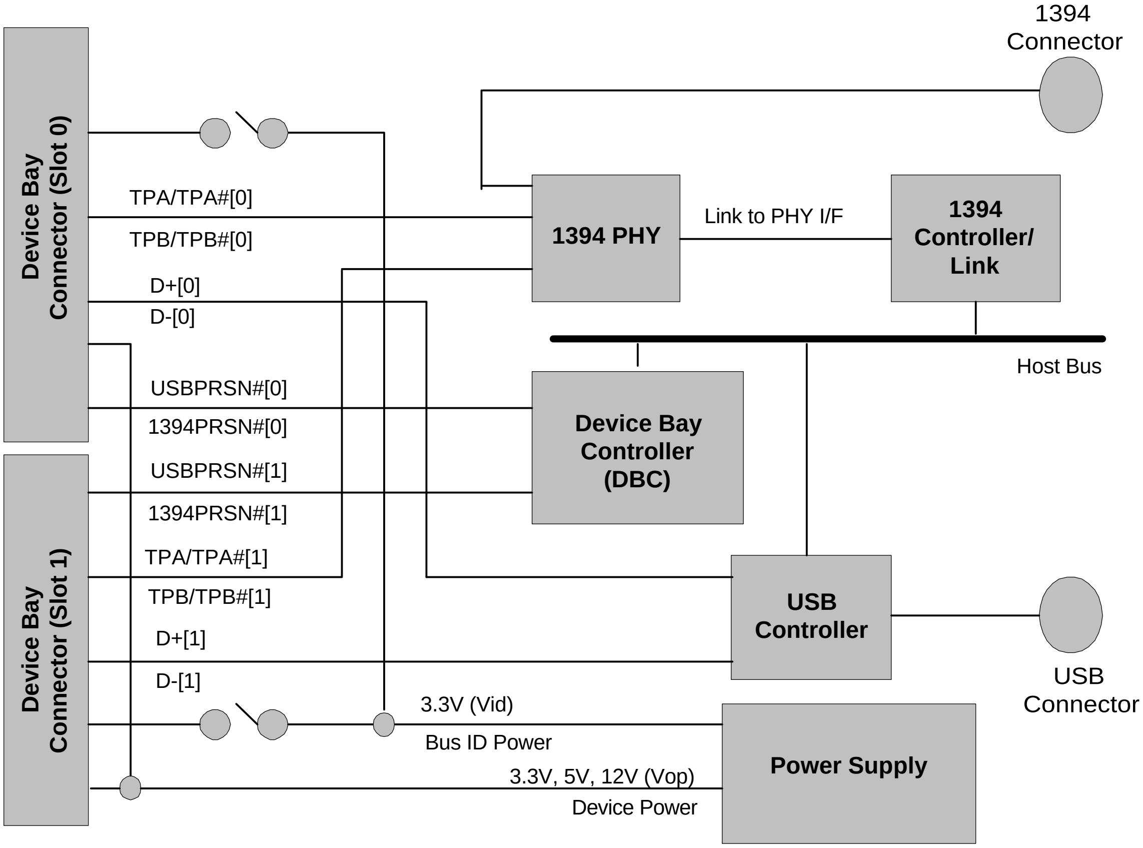

Figure 3-1. Example DB32 Bay Connector Wiring

3.1 Data Buses

This section describes general requirements of and support for the USB and 1394 data buses within a Device Bay system.

The electrical interface definition is driven by the needs of the device classes that will be supported. A device is not required to implement all the interfaces on the Device Bay connector. A Device Bay device, at minimum, must support a power bus and one data transfer bus (either USB or 1394). The data transfer buses can be used in any combination in a Device Bay device. Compound devices, which use both USB and 1394 buses, are possible.

3.1.1 USB

USB provides a low-cost interface to medium-speed serial devices. Typical devices are line interfaces, such as ISDN and POTS, or audio codecs. USB also allows the user to add a simple USB hub to augment the USB interface built into their system.

Each Device Bay device that has a USB component must support a configuration space containing a unique Serial Number (iSerialNumber) as described in Section 9.6.1 (Device Descriptor) of the USB specification, as well as the device's power requirements to be described in *the USB Power Management Specification*.

Both documents are referenced in section 1.5 of this specification.

The USB signals in the Device Bay connector, and each signal's pin count, are shown in Table 3-1.

Table 3-1. USB Signal and Pin Count

Signal	Number of pins
USB D+	1
USB D-	1

3.1.1.1 USB Identification and Enumeration

USB identification and enumeration is considered complete upon the Devices reception and decode of the end-configuration command.

3.1.2 1394

1394 provides a high-performance interface for high-bandwidth devices. Initial support in Device Bay is for the IEEE 1394-1995 (100 to 400 Mbps) specification data rates and, to the extent possible, higher 1394 data rates (800 to 3200 Mbps) as they become available (such as 1394b).

Each Device Bay device that has a 1394 component must support 1394 CSR and ROM space, and the ROM space must contain the device's power requirements.

For reference information on 1394, see section 1.5 of this specification.

The 1394 signals in the Device Bay connector, and each signal's pin count, are shown in Table 3-2. Each bay must support a 100, 200 and 400 Mbps connection.

Table 3-2. 1394 Signal and Pin Count

Signal	Number of pins
1394 TPA 1394 TPA#	2
1394 TPB 1394 TPB#	2

3.1.2.1 1394 Identification and Enumeration

1394 identification and enumeration is considered complete after the login command is successfully completed (including any password comparison if so enabled).

3.2 Power Distribution Across the Device Bay Connector

This section describes direct current (DC) power distribution across the Device Bay connector.

Devices draw power through the Device Bay connector. They cannot draw any power through their native bus, as 1394 and USB cable power are not provided.

3.2.1 Bus Connector Requirements

For host system and device connector requirements see section 4 – Device Bay Connector Set Requirements.

3.2.2 Device Bay Power Distribution Examples

Form factor power distribution is shown in Figures 3-2 and 3-3. Distribution of power from a bay to a device, across the Device Bay connector, is shown. For definitions of the labels used in Figure 3-2 (such as V_{id} and V_{12}), see section 3.3 of this specification. For more information about the Device Bay connector, see section 4.

It is recommended that the bay provide similar slew rate control on the individual V_{id} lines to minimize the impact on currently enabled Device Bay devices or the system that contains the device.

3.2.2.1 Example DB32 Form Factor Device Bay Power Distribution

Figure 3-2 shows an example implementation of power distribution for a DB32 form factor.

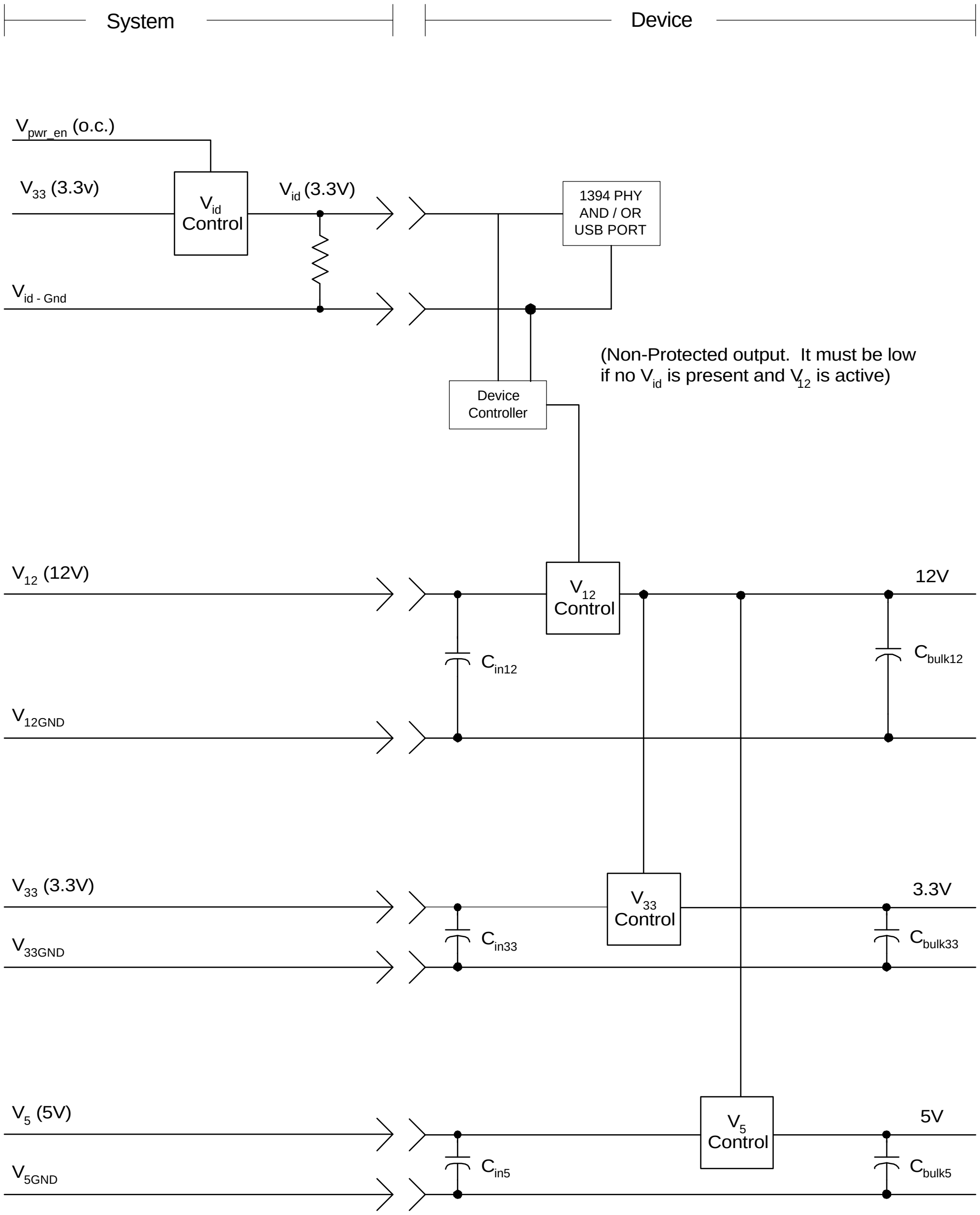

Figure 3-2. Example DB32 Device Bay Power Distribution Diagram

3.2.2.2 Example DB20 and DB13 Form Factor Device Bay Power Distribution

DB20 and DB13 form factor power distribution is defined in Figure 3-3, which shows the distribution of power from a bay, across a Device Bay connector, and to a device. For definitions of the labels used in Figure 3-3 (such as V_{id}), see section 3.3 of this specification. For more information about the Device Bay connector, see section 4.

A major difference between DB20 and DB13 power distribution and DB32 power distribution is the lack of 12-volt support in the DB20 and DB13 form factors.

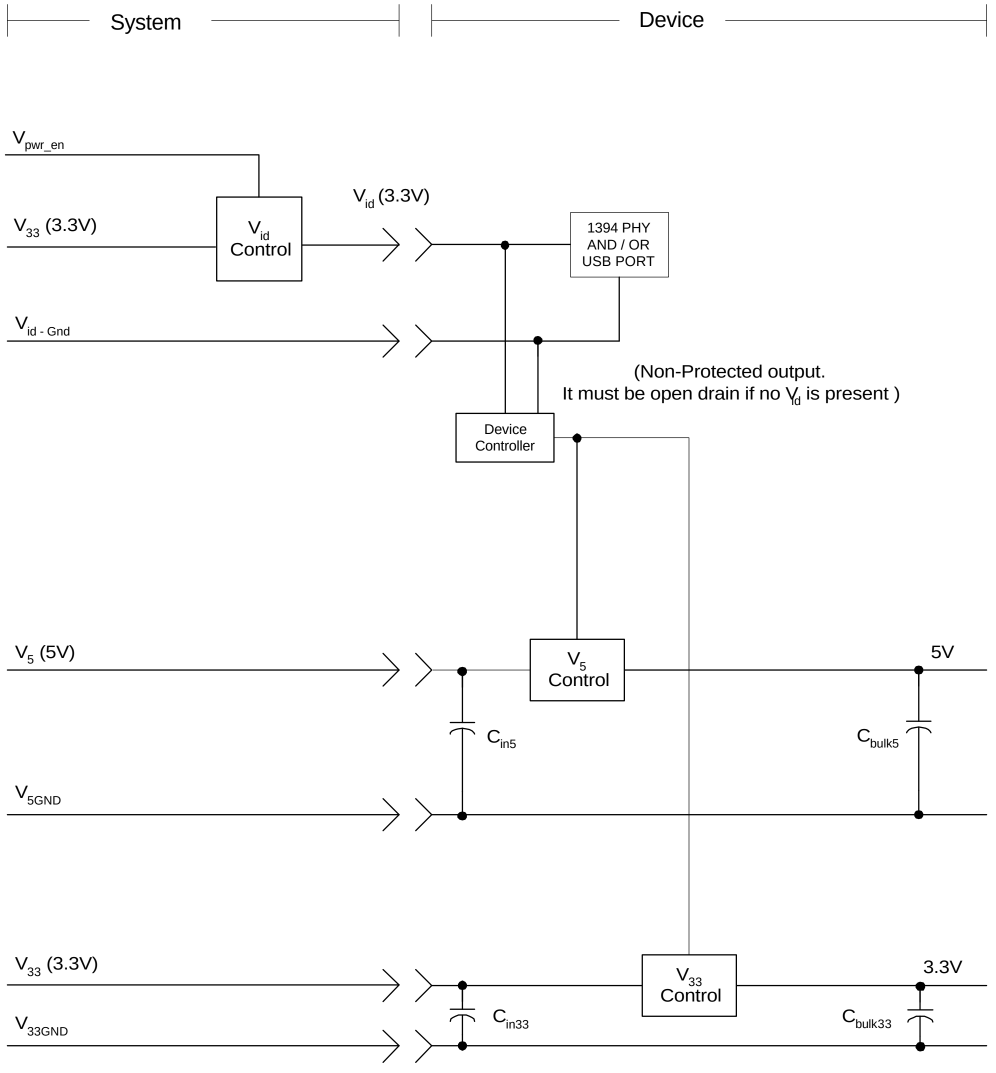

Figure 3-3. Example DB20 and DB13 Device Bay Power Distribution Diagram

3.3 Power Signal Definitions

V_{id}	Identification voltage. 3.3V switched by the bay. See section 3.6.1 for V_{id} Requirements.
V_{12}	12-Volt Operational Voltage. See section 3.6.2 for V_{op} requirements.
V_5	5-Volt Operational Voltage. . See section 3.6.2 for V_{op} requirements.

V_{33}	3.3-Volt Operational Voltage. See section 3.6.2 for V_{op} requirements.
V_{op}	For DB32, this is a combination of V_{33}, V_5, and V_{12}. For DB20 and DB13, this is a combination of V_{33} and V_5. to be used to operate the device after identification and enumeration. identification.

3.4 Power Level Definitions

Trans	Transient Power. A power level that is supported only for a duration of no more than 100 µs. This value is measured by calculating the worst-case instantaneous Peak Power draw.
Peak	Peak Power. A power level that is supported only for a duration of no more than 30 seconds. This value is measured by calculating the worst-case RMS power consumption over a 30-second period.
E_{Cont}	Electrical Continuous Power. A power level that is supported only for a duration of no more than 300 seconds. This value should correspond to the continuous power supply capability of the power supply, connectors, and wiring. This value is measured by calculating the worst-case RMS power consumption over a 300-second period.
T_{Cont}	Thermal Continuous Power. A power level that is supported for any duration greater than 300 seconds. This value is measured by calculating the worst-case RMS power consumption over a 20-minute period.
T_{Aggr}	Total Aggregate power. The sum total of all power that the device is drawing from the $V_{id,}$ $V_{12,}$ $V_{5,}$ and V_{33} rails.

For more information about T_{Aggr}, see section 3.8.4.3 of this specification.

3.5 Power Attribute Definitions

The power attributes are voltage regulation, ripple, and di/dt. These are measured at the following locations:

- Bay attributes cover from the system through the bay side of the Device Bay connector (Receptacle), including the V_{id} FET.
- Device attributes include the device side of the Device Bay connector (Plug) and the device V_{op} FETs.

Di/dt is the rate of change of current that the device can draw. Di/dt for V_{id} is the minimum rate of current change the system will supply. Di/dt for V_{op} is the maximum rate of current change the device can present to the system. Di/dt for V_{id} is a system requirement only; it is provided to the device only as a reference.

3.6 Power Signal Requirements

This section contains all the rules and requirements for using the V_{id} and V_{op} power rails in a Device Bay subsystem.

3.6.1 Requirements for the V_{id} Power Signal

The system side of the Device Bay connector provides the V_{id} power signal to enable the system to identify and enumerate the device at a low power level. V_{id} and its associated ground in the signal segment and all of the grounds in the power segment of the Device Bay connector are the only required power connections on the device. For more information see chapter 4 of this specification.

3.6.1.1 1394 Interface Device V_{id} requirements

The V_{id} requirements for a device that uses the 1394 data interface are as follows:

1) The device must power the 1394 Physical (PHY) interface off of V_{id}.
2) The device must power the 1394 Link interface off of V_{id}.

3) The device must power enough of the 1394 ROM and CSR registers to allow the system to:
    a) Read all of the ROM space, including the Device Bay required registers (See section 3.1.2.1).
    b) Proceed from power up through the process of setting up the CSR regsisters and processing a Login Management ORB, including password comparison if so enabled.
4) The device must not draw any other power from the Device Bay sub-system until all of the above requirements are met.

3.6.1.2 USB Interface Device V_{id} requirements

The V_{id} requirements for a device that uses the USB data interface are as follows:

1) The device must power the USB Physical interface off of $V_{id.}$
2) The device must power enough of the USB support electronics to all the system to:
    a) Read all of the USB Descriptor space, including the Device Bay required descriptor space (see section 3.1.1.1).
    b) Proceed from power up to the processing of the “end configuration command.”

3) The device must not draw any other power from the Device Bay subsystem until all of the above requirements are met.

3.6.1.3 Compound Device V_{id} requirements

A compound device is a device that uses both the 1394 and USB data interfaces for data communication.

A compound device must use V_{id} to power the components detailed under both the 1394 and USB interface V_{id} requirements.

3.6.1.4 General V_{id} requirements

Table 3-3 presents the V_{id} power signal design rules and requirements, and identifies the Device Bay components that enforce each rule and requirement. These requirements apply to all Device Bay devices regardless of the data interface(s) they use.

Table 3-3. General V_{id} Power Requirements

Rule or requirement	Locus of enforcement	Comment
A device must never draw must never draw more the maximum current on V_{id} as described in section 3.8.4 (identification/enumeration and full operation)	Device design	
A device may V_{id} during full operational mode provided it does not violate any of the V_{id} maximum power requirements.	Device design	
If no voltage is present on V_{id}, then the device must not draw any power from any operating voltage rail, nor may it interact in any way with 1394 or USB busses.	Device design	The device must provide a high impedance connection to the 1394 and/or the USB bus as to not cause the device to be detected on either bus.
V_{id} must be switched on a per-bay basis.	Bay power rail design (see example drawings)	

Rule or requirement	**Locus of enforcement**	**Comment**
A device must never draw must never draw more the maximum current on V_{id} as described in section 3.8.4 (identification/enumeration and full operation)	Device design	
A device may V_{id} during full operational mode provided it does not violate any of the V_{id} maximum power requirements.	Device design	
V_{id} must be disabled when no device is present.	DBC	DBC circuitry does this; DBC driver is not involved.
V_{id} can be disabled to minimize the power drawn by a non-enabled device.	DBC driver	
V_{id} must be disabled to disable a device.	DBC driver	
If neither device presence pin is active, then V_{id} must not be enabled, even if the software attempts to enable power for that bay.	DBC circuitry	
The assertion of a device presence pin must not cause V_{id} to be enabled without operating system interaction.	DBC circuitry and firmware interaction with DBC driver	
The bay must always provide a ground connection to all bay-side ground connector pins.	Bay power design	
The DBC must disable V_{id} if the software-controlled interlock mechanism is disengaged.	DBC circuitry	
In the event that it becomes necessary to override the software-controlled interlock (for example, software malfunction), V_{id} must be disabled. Additional methods can be employed as long as they meet all the requirements.	The user and/or the bay system design.	It is recommended that this be accomplished by removing the main power from the system (that is, pull the power cord or remove the battery).
When V_{id} is disabled, the bay must provide a resistive connection of no more than 10KΩ to ground.	Bay V_{id} power rail design	
When V_{id} is enabled, the bay may provide a resistive connection of no more than 10KΩ to ground. This facilitates the bleeding off of the V_{id} voltage, after it has been disabled, to minimize the amount of time between when the device is turned off and when it is really off.	Bay V_{id} power rail design	
The device can use an external power source (battery, wall connection, network, and so on) to sense the level of V_{id}.	Device design	

3.6.2 Requirements for the V_{op} Power Signals

Table 3-4 presents the V_{op} power signal design rules and requirements and identifies the Device Bay components that enforce each rule and requirement. These rules are applicable to all Device Bay devices, regardless of the data interface used.

Table 3-4. General V_{op} Power Requirements

Rule or requirement	Locus of enforcement	Comment
Once the device is identified and enumerated, it may draw up to the maximum allowable power for V_{12}, but it must not exceed the total aggregate power requirement.	Device design	See Aggregate power requirements also.
Once the device is identified and enumerated, it may draw up to the maximum allowable power for V_5, but it must not exceed the total aggregate power requirement.	Device design	See Aggregate power requirements also.
Once the device is identified and enumerated, it may draw up to the maximum allowable power for V_{33}, but it must not exceed the total aggregate power requirement.	Device design	See Aggregate power requirements also.
A device must not require V_{op} to retain the hardware context required by the OnNow initiative.	Device design	
The bay may disable any and all of the V_{op} rails in any order after the device is in a non-operational state or if V_{id} has been disabled for the bay.	Bay design and device validation	V_{op} power sequencing requirement.
The bay may provide additional power protection/detection (over-current etc) provided it meets all the other bay power requirements (current, voltage, sequence, etc)		Applicable for high availability situations (Servers, industrial etc).
V_{op} is not required to be enabled or disabled during device insertion or removal.	DBC circuitry	V_{op} power sequencing requirement.
The bay must supply all required V_{op} power rails prior to V_{id} being enabled	Bay design	V_{op} power sequencing requirement.
The device must meet the power requirements for a device not currently enumerated, regardless of the availability or sequence of the powering of the V_{op} rails.	Device design and/or DBC	Non-enumerated power requirements.
Power requirements for a device not currently enumerated are: until a device is enumerated and enabled, the device must draw no more than 5µA and must not expect any more than 0 µA from any operational power rails, except from V_{id} (for a maximum of 1.5 watts).	Device Design	Non-enumerated power requirement.

3.7 Minimum Device Bay Sub-System Power Distribution Requirements

Each Device Bay-enabled subsystem must support at least one “worst-case” Device Bay device while using any supported power source, for example, battery powered and alternating current (AC) powered modes.

For the purposes of this section, a worst-case device is one that barely meets one or more of the three mandatory maximum allowable wattage requirements: Peak, E_{Cont}, and T_{Cont}. For the maximum allowable device power wattages for all three Device Bay form factors, see Table 3-5.

3.7.1 Minimum Battery-Powered Device Bay subsystem requirements

If a Device Bay device is battery-powered, then it must meet this worst-case requirement while operating from an adequately charged primary battery pack and without optional batteries. If insufficient power is available to support the device, the system must gracefully reject it.

3.7.2 Minimum Device Bay Sub-System power example

Take for example a DB32 device that displays power characteristics of *exactly* 45 watts for Peak, and just under 30 watts for E_{Cont} (28.33 watts), and that never sustains this value (otherwise in violation of the 25-watt T_{Cont} maximum allowed) for longer than 300 seconds. Such a device, while compliant, is hardly overachieving and must be designated a worst-case Device Bay device. The aggregate power profile for such a device is graphically represented in Figure 3-5.

3.8 Device Power Distribution Requirements

This section describes:

- Device power distribution requirements
- Adherence to form factor thermal specifications
- Cautionary device power considerations
- Measuring of maximum allowable device capacitance on the power rails
- Power requirements for devices not yet enumerated
- Maximum allowable device power

3.8.1 Power and Thermal Relationships

The Device Bay power specification for each form factor cannot be specified independently of the form factor thermal specification. For more information about thermal requirements, see section 5 of this specification.

3.8.2 Cautionary Device Power Notes

The following are cautions for powering Device Bay devices:

- Be careful of exceeding Di/dt, since this may cause the voltage rail to go out of tolerance, not only for the device in question, but also for devices sharing the same power subsystem.
- The device cannot short V_{id} and V_{33} together (after the device FET) because the device will not be able to guarantee that it does not violate the maximum current draw on V_{id}.

3.8.3 Maximum Allowable Device Power Input Capacitance

The maximum allowable device capacitance on the power rails is measured by shorting all the pins for that voltage rail together on the mating side of the device connector and measuring the input capacitance while the device is in a powered-off state.

Table 3-5. Maximum Allowable Device Power Input Capacitance

Power rail	DB32	DB20	DB13
V_{id}	NA	NA	NA
V_{33}	2 µF	1 µF	1 µF
V_5	2 µF	1 µF	1 µF
V_{12}	2 µF	1 µF	1 µF

3.8.4 Maximum Allowable Device Power

Important: A device must be below all maximum power requirements, which are shown in the maximum allowable device power tables, at all times. This includes, but is not limited to, Trans, Peak, E_{Cont}, and T_{Cont} power levels for both individual power rails and T_{Aggr} power.

The limitations expressed in the maximum allowable power tables govern how often a device can go into the next-higher power level. A device may draw peak power as often as needed, as long as it also meets the E_{Cont} and T_{Cont} power requirements.

The absolute maximum power level is defined as the transient power level.

A device must meet the Di/dt requirements for the rising edge.

A device should meet the Di/dt requirements for the falling edge.

3.8.4.1 Maximum Allowable DB32 Device Power

Table 3-8. Voltage and Current Supplies Available to Each Device

Ident	Voltage	Reg-ulation	Ripple	di/dt	Transient < 100 uS		Peak 100uS - 30 Sec		E Cont 30 - 300 sec		T Cont. > 300 Sec	
	(Volts)	(+-%)	(+-mV)	(A/uS)	(A)	(W)	(A)	(W)	(A)	(W)	(A)	(W)
V_{id}	3.3	5	50	1A/20uS	0.91	N/A	0.45	1.5	0.45	1.50	0.45	1.5
V_{12}	12	10	150	1A/20uS	5	N/A	3.75	45	2.5	30.00	2	24
V_{5}	5	+5/-3	50	1A/20uS	4	N/A	3	15	2	10.00	2	10
V_{33}	3.3	+5/-3	50	1A/20uS	7	N/A	5.25	17.33	3.5	11.55	3.5	11.55
T Aggr Power (Watts)								45		30		25

3.8.4.2 Maximum Allowable DB20 and DB13 Form Factor Device Power

Table 3-9. Voltage and Current Supplies Available to Each Device

Ident	Voltage	Reg-ulation	Ripple	di/dt	Transient < 100 uS		Peak 100uS - 30 Sec		E Cont 30 - 300 sec		T Cont. > 300 Sec	
	(Volts)	(+-%)	(+-mV)	(A/uS)	(A)	(W)	(A)	(W)	(A)	(W)	(A)	(W)
V_{id}	3.3	5	50	1A/20uS	0.91	N/A	0.45	1.50	0.45	1.50	0.45	1.50
V_{5}	5	+5/-3	50	1A/25uS	2.00	N/A	1.60	8.00	0.80	4.00	0.80	4.00
V_{33}	3.3	+5/-3	50	1A/20uS	3.03	N/A	2.42	8.00	1.21	4.00	1.21	4.00
T Aggr Power (Watts)								8.00		4.00		4.00

3.8.4.3 Continuous power combination examples.

The system power supply is not required to supply more than the T_{Aggr} power. For example, for the DB32 form factor, a compliant device could require the system to provide 25 watts for a thermal continuous duration in the combinations shown in Tables 3-6 and 3-7.

Table 3-6. First Example of Power Combinations Totaling 25 Watts for TCont Duration

Voltage Rail	Voltage (Volts)	Current (Amps)	Power (Watts)
V_{id}	3.3	0.454	1.498
V_{33}	3.3	3.5	11.6
V_5	5	2	10
V_{12}	12	0.1585	1.902
Total Power			25

Table 3-7. Second Example of Power Combinations Totaling 25 Watts for TCont Duration

Voltage Rail	Voltage (Volts)	Current (Amps)	Power (Watts)
V_{id}	3.3	0.303	1
V_{33}	3.3	0	0.0
V_5	5	0	0
V_{12}	12	2	24
Total Power			25

3.8.4.4 Aggregate Power Examples

The example non-compliant DB32 device shown in Figure 3-4 violates the aggregate power rules in the following two ways:

- Exceeds the E_{Cont} maximum of 30-watts RMS with a value of 31.5 watts, calculated as follows:

```
((45 * 30 ) + ( 30 * 270 ))/300 = 31.5
```

- Exceeds the T_{Cont} maximum of 25-watts RMS with a value of 26.625 watts, calculated as follows:

```
((45 * 30 ) + ( 30 * 270 ) + (25 * 900) )/1200 = 26.625
```

and Figure 3-4 describes the time durations and associated power levels of a non-compliant DB32 device.

Table 3-8. Non-Compliant DB32 Power Summary Table

Time	Power	Agg Pwr
0	45	
30	45	1350
30	30	0
300	30	8100
300	25	0
1200	25	22500
1200	45	
1230	45	1350
1230	30	0
1500	30	8100
1500	25	0
2400	25	22500
Ave T_{aggr}		26.625

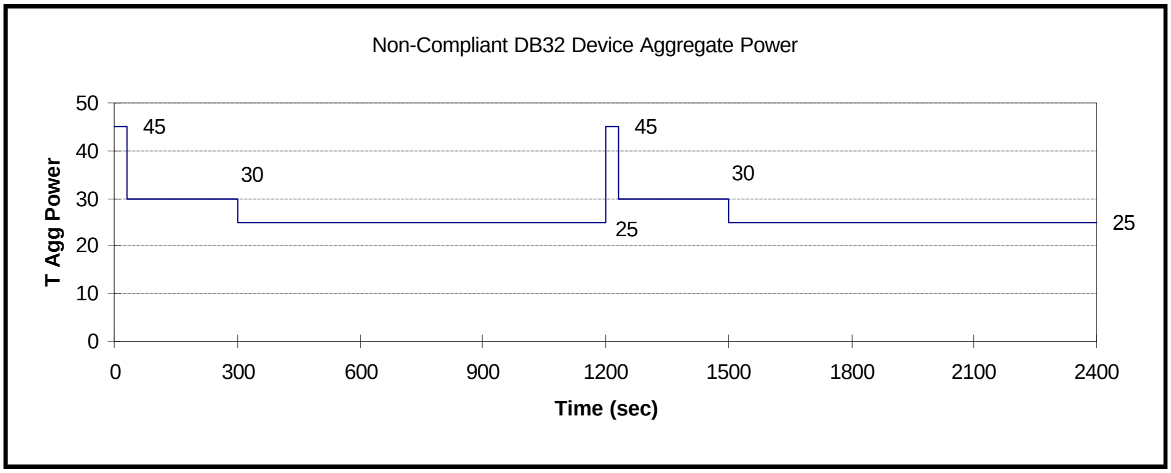

Figure 3-4. Non-Compliant DB32 Power Summary Figure

Table 3-9 and Figure 3-5 shows an example power profile of a power compliant DB32 device.

Table 3-9. Compliant DB32 Power Summary Table Example #1

Time	Power	Agg Pwr
0	45.00	
30	45.00	1350.00
30	28.33	0.00
300	28.33	7650.00
300	23.33	0.00
1200	23.33	21000.00
1200	45.00	
1230	45.00	1350.00
1230	28.33	0.00
1500	28.33	7650.00
1500	23.33	0.00
2400	23.33	21000.00
Ave T_{aggr}		25.00

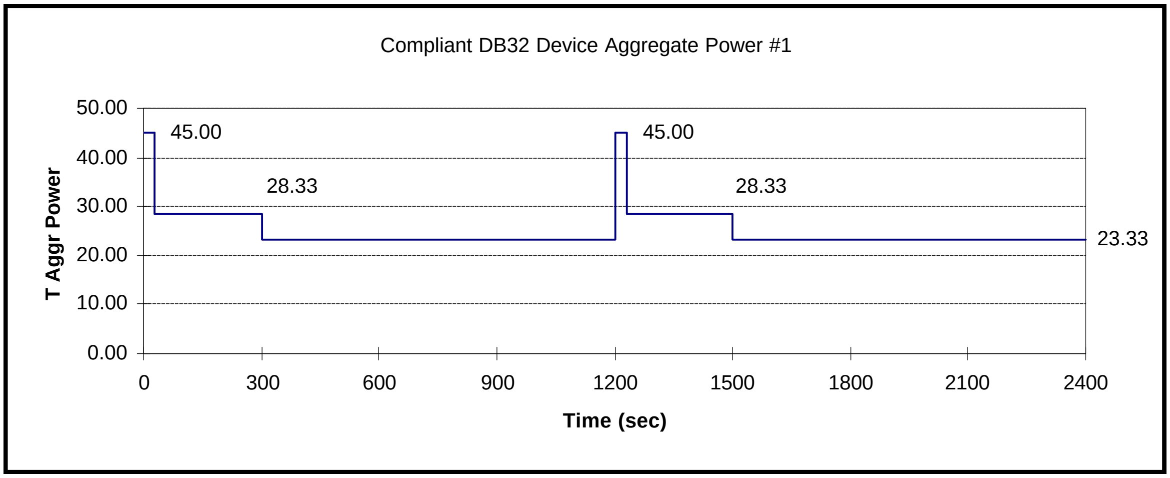

Figure 3-5. Compliant DB32 Device Aggregate Power Example #1

Table 3-10 and Figure 3-6 shows a second example power profile of a power compliant DB32 device.

Table 3-10. Power summary table for a compliant DB32 device Example #2

Time	Power	Agg Pwr
0	45.00	
30	45.00	1350.00
30	28.33	0.00
300	28.33	7650.00
300	45.00	0.00
330	45.00	1350.00
330	28.33	0.00
600	28.33	7650.00
600	45.00	0.00
630	45.00	1350.00
630	28.33	0.00
900	28.33	7650.00
900	10.00	0.00
1200	10.00	3000.00
1200	45.00	
1230	45.00	1350.00
1230	28.33	0.00
1500	28.33	7650.00
1500	45.00	0.00
1530	45.00	1350.00
1530	28.33	0.00
1800	28.33	7650.00
1800	45.00	0.00
1830	45.00	1350.00
1830	28.33	0.00
2100	28.33	7650.00
2100	10.00	0.00
2400	10.00	3000.00
Ave T_{aggr}		25.00

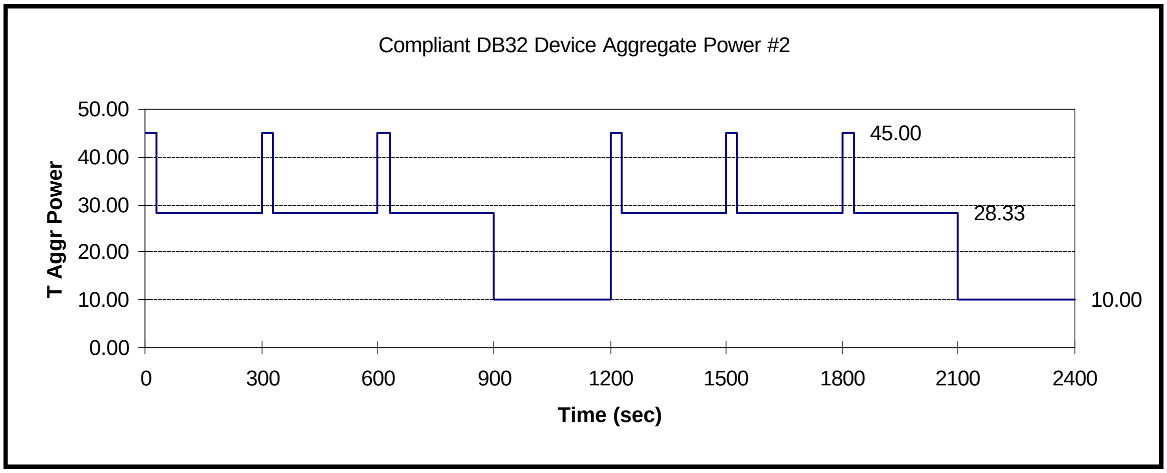

Figure 3-6. Compliant DB32 Device Aggregate Power Example #2

Table 3-11 and Figure 3-7 shows a third example power profile of a power compliant DB32 device.

Table 3-11. Power summary table for a compliant Device Example #3

Time	Power	Agg Pwr
0	25.00	
30	25.00	750.00
30	25.00	0.00
300	25.00	6750.00
300	25.00	0.00
1200	25.00	22500.00
1200	25.00	
1230	25.00	750.00
1230	25.00	0.00
1500	25.00	6750.00
1500	25.00	0.00
2400	25.00	22500.00
Ave T_{aggr}		25.00

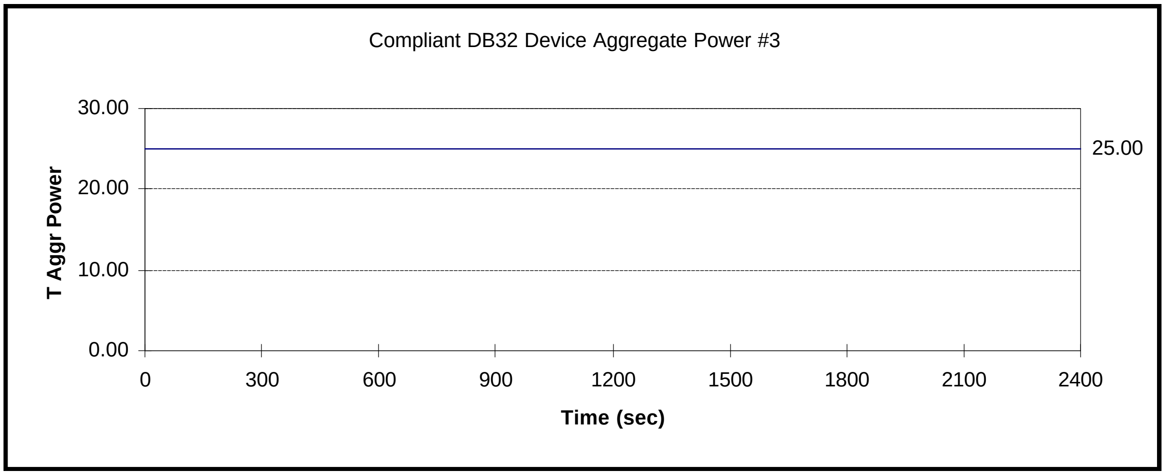

Figure 3-7. Compliant DB32 Device Aggregate Power Example #3

3.9 Power Management

This section describes the power management requirements for Device Bay subsystems, USB devices inserted in a Device Bay subsystem, and 1394 devices inserted in a Device Bay subsystem. The intent of this section is to:

- Fully describe the power state transition model for Device Bay subsystems and devices inserted in a bay, with examples (see sections 3.9.3.2 and 3.9.3.3).
- Fully describe the self-reporting requirements for Device Bay subsystems and devices that enable an operation system to implement the power state transition model. Power reporting requirements are specified for subsystems with both ACPI-based DBCs and subsystems with USB-based DBCs.

Note: For Device Bay subsystem power management, the subsystem is represented by the Device Bay Controller (DBC) that controls the device insertion and removal events in the bay(s) that are part of the subsystem.

This section contains many references to other specifications and other chapters of this specification to fully describe the power management model and self-reporting requirements.

3.9.1.1 Overview of Device Bay Power Reporting Requirements

This section describes the power reporting requirements for:

- Device Bay subsystems (that is, the Device Bay Controller)
- USB devices inserted into a bay
- 1394 devices inserted into a bay

Figure 3-8 shows a schematic of a two-bay Device Bay subsystem with an ACPI-based DBC, and a USB device inserted in one bay and a 1394 device inserted in the other bay.

All of the power reporting information required of each of the components of the subsystem (the DBC, the USB device, and the 1394 device) resides in the same location: in the ACPI BIOS on the system board of the host that contains the Device Bay subsystem.

Figure 3-8. Two-bay Device Bay subsystem with an ACPI-based DBC

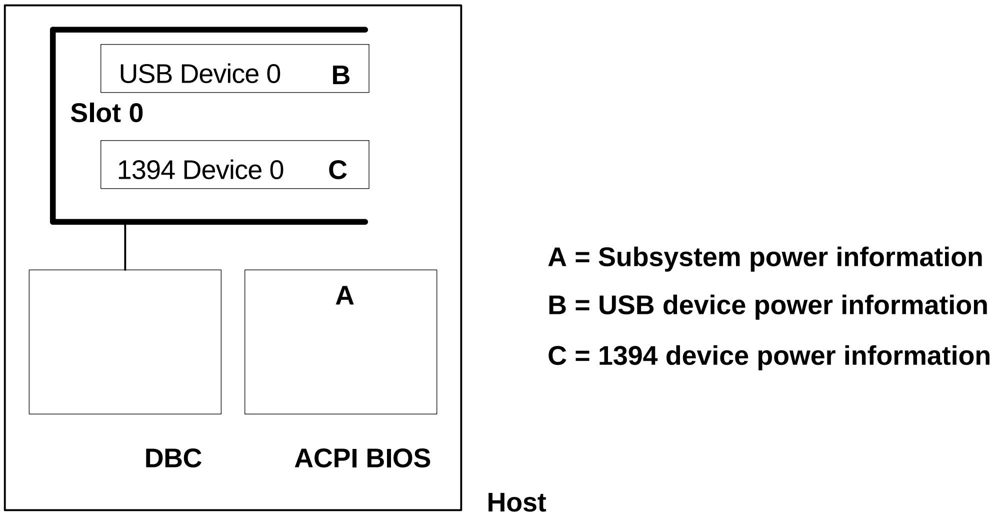
USB Device 0
B
Slot 0
1394 Device 0
C
A
DBC
ACPI BIOS
Host
A = Subsystem power information
B = USB device power information
C = 1394 device power information

Figure 3-9 shows a schematic of a two-bay Device Bay subsystem with a USB-based DBC and a USB device inserted in one bay and a 1394 device inserted in the other bay.

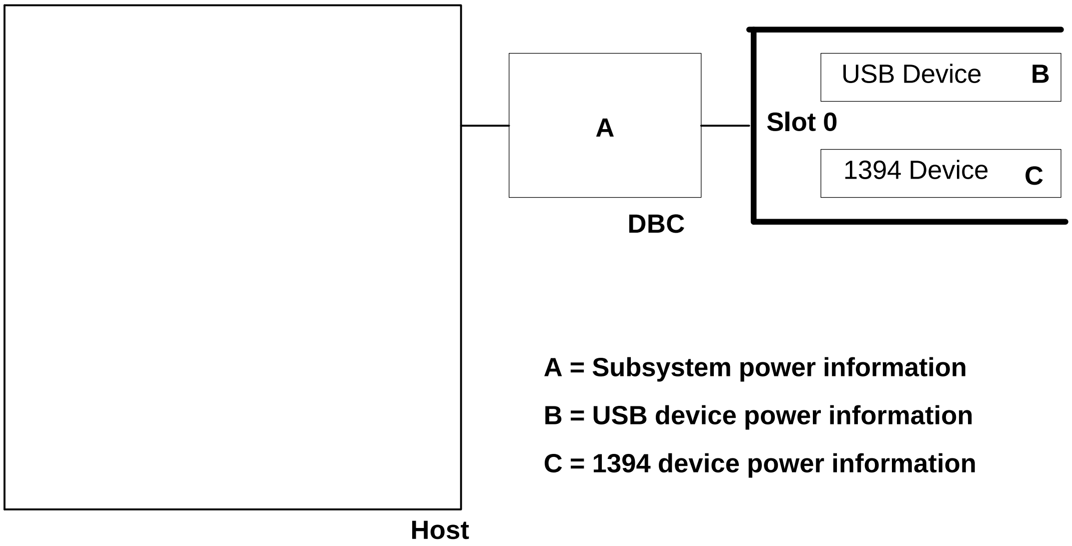
USB Device
B
A
Slot 0
1394 Device
C
DBC
A = Subsystem power information
B = USB device power information
C = 1394 device power information
Host

Figure 3-9. Two-bay Device Bay subsystem with a USB-based DBC

Table 3-12 summarizes the information contained in Figure 3-8 and Figure 3-9, and adds information about the format of the power reporting information at each location.

Table 3-12. Summary of Device Bay Power Reporting Information

Type of DBC	**Format and location of power reporting information**	**Comment**
ACPI-based DBC	A - Bay subsystem power reporting information is in the APCI BIOS; format is ACPI objects based on the Device Bay Power Capabilities (DBPC) register model.	For a description of the DBPC register model, see Annex C.
	B - USB device power reporting information is in the device descriptors, as specified in the USB Core specification and the USB Common Class Interface Power Management Feature specification.	The DBC inspects this information on an as-needed basis as it is passed between the USB native bus drivers.
	C - 1394 device power reporting information is in the device Configuration ROM, as specified in the 1394 Specification for Power State Management (see Section 1.5 for a detailed	The DBC inspects this information on an as-needed basis as it is passed between the 1394 native bus drivers.

Type of DBC	**Format and location of power reporting information**	**Comment**
	reference)	
USB-based DBC	A - Bay subsystem power reporting information is in the DBC silicon; format is a USB descriptor.	For more information about the DBC device descriptor, see the USB DBC Class specification, (See section 1.5 for a detailed reference)
	B - USB device power reporting information is in the device firmware; format is a USB descriptor.	For more information about the USB Device Bay device-specific power descriptor, see section 1.5 for a detailed reference.
	C - 1394 device power reporting information is in the device Configuration ROM.	For more information about the 1394 Device Bay device-specific power descriptor, see section 1.5 for a detailed reference.

3.9.2 Device Bay Sub-System Power Capabilities Register Model

For a detailed description of the Device Bay sub-system power capabilities (DBPC) model, please see Section 6.6.10 in the Device Bay Controller chapter.

3.9.3 Coordinating Bay and Device Power State Transitions

The OS must coordinate the power state of a bay and the power state of an inserted device. The power state of the device is controlled by its native bus power management and the power state of the device always dictates the power state of the bay. The OS must ensure the power state of the bay is consistent with the power state of any 1394 or USB device inserted into the bay as defined by the OnNow initiative.

3.9.3.1 OnNow and Wake-up Requirements for V_{id} and V_{op}

Bays are devices that can transition between the states D0, D1, D2, and D3. The OS controls V_{id} during bay power state transitions between D0, D1, D2, and D3. In general,

- The operating system directly enables or disables V_{id} using the Bay x V_{id} Power Control bit (BCERx/PWR_CTL) provided through the DBC interface.
- The use of V_{op} is controlled by the native bus power management components.
- V_{id} is enabled in all states except D3.

A bay:

- Must support a D2 wake-up device with a bus level of D0 and a node level (if applicable) of N0.
- Can support D2 wake up from a D1 or D2 bus state, or a N1 or N2 node state. This is not required, and is dependent on future 1394 bus and PHY characteristics.

A device:

- Can support D3 wake up but must not require any system power (V_{id} or V_{op}) to do so.
- Must support D2 (this may be the same condition as the un-enumerated state).
- Can support D2 wake up (if appropriate for this device class) to achieve low total system power.

3.9.3.2 Example V_{id} and V_{op} States for 1394 Bus, Node, and Device States

In Table 3-13, the terms “node” and “unit” are defined as follows (for more information, see the 1394 power management draft specification, which is referenced in section 1.5 of this specification):

- A 1394 node is the 1394 Link and the 1394 PHY
- A 1394 unit is the rest of the 1394 device's electronics

The device must report its power requirements for all D*x* states, including D3, using the 1394 *Power Management Specification*, under the "Node_Power_Directory" and "Unit_Power_Directory" (see the reference in section 1.5 of this specification). These entries include information on:

- Voltage
- Which D*x* state, if it can wake-up from this power state
- How much power is needed

These entries are repeated to describe all voltages and all power levels.

Table 3-13. Example V_{id} and V_{op} States for 1394 Bus, Node, and Device States

Bus ACPI State	**Node ACPI State**	**Unit ACPI State**	**V_{id} state**	**V_{op} state**	**Hard Disk Storage Example**	**Network Interface Example**
B0	N0	D0	Full Power (<= 1.5W)	Full power (Wattage depends on Form Factor)	Drive is fully operational 1394 Interface full operational	NI is fully operational
B0	N0	D1	Full Power (<= 1.5W)	Low Power draw	Platters spun down, operational logic still powered and running. 1394 Interface full operational	NI is draws lower V_{op} power by putting ram buffers and microcontroller in sleep/ low-power state.
B0	N0	D2	Full Power (<= 1.5W)	Device may draw up to 5 uA / V_{op} System may supply 0 power	Platters spun down, operational logic power down 1394 Interface full operational	Device can support magic packet detection but must do so in < 1.5W combined with 1394 I/F. 1394 Interface full operational
B0	N1 or N2	D2	Low Power (May keep PHY alive to send wake-up)	Device may draw up to 5 uA / V_{op} System may supply 0 power	This state could be used to wake-up to go to D1 for a media eject. Platters spun lasses of computers, explicitly inc luding desktop, mobile, and server machines. This	ecification defines characteristicke up. Node level is dependent on where the wake up event goes (PHY or link) N1 - Link Standby, PHY is On N2 - Link OFF, PHY in suspend
B1	N2	D2	Lower than low Power. Both Device and	Device may draw up to 5 uA / V_{op} System may supply 0 power		Device can support magic packet wake up. Device depends on a Bus wake-up propagated up and down the 1394 tree

Bus ACPI State	**Node ACPI State**	**Unit ACPI State**	**V_{id} state**	**V_{op} state**	**Hard Disk Storage Example**	**Network Interface Example**
			the Link are off and the PHY is in suspend state.			as the location of the ROOT node is unknown.
B1	N2	D3	System must switch off V_{id} as the device is not ever required to not draw any power off of V_{id}.	Device may draw up to 5 uA / V_{op} System may supply 0 power	Device is off	Device is off, no wake up is required.
B3	N3	D3	System must switch off V_{id} as the device is not ever required to not draw any power off of V_{id}.	Device may draw up to 5 uA / V_{op} System may supply 0 power	Device is off, Node is off, Bus is off.	Device is off, Node is off, Bus is off.

3.9.3.3 Example V_{id} and V_{op} States for USB Bus, Interface, and Device Power States

A USB device must report its power requirements and power management capabilities using the descriptors specified in the v1.0 USB Specification, or later, and the v1.0 USB Common Class Interface Power Management Feature Specification, or later. Both these specifications are available at www.usb.org/developers/.

This section contains examples for USB Storage and Network devices. These examples are based on the *Device Class Power Management Reference Specification for the Network Device Class* and the *Device Class Power Management Reference Specification for the Storage Device Class*. Both these specifications are available from www.microsoft.com/hwdev/onnow.htm/.

Note that USB storage devices are relatively slow speed storage devices, such as floppy disk drives. The floppy disk drive part of the *Device Class Power Management Reference Specification for the Storage Device Class* specification is used in the example below, the Hard Disk, CD-ROM, and IDE/ATAPI removable storage device power state definitions are not used.

The device power state definitions for floppy disk drives and network devices are reproduced here for the convenience of the reader. These definitions may not be the most current definition. If there is a conflict between the definitions in the tables below and the *Device Class Power Management Reference Specification for the Network Device Class* or the *Device Class Power Management Reference Specification for the Storage Device Class* then the definitions in those specifications are the ones to use.

Table 3-14. Floppy Disk Device Power State Definitions

Power State	**Definition**
D0	Drive controller (interface and control electronics) fully functional.

	Drive motor (spindle) turning.
D1	Not defined for floppy drives. Use D3 instead.
D2	Not defined for floppy drives. Use D3 instead.
D3	Drive controller not functional and context lost. Drive motor stopped.

Table 3-15. Network Device Power State Definitions

Power State	**Definition**
D0	Device is on and running and is delivering full functionality and performance to the user. Device is fully compliant with the requirements of the attached network.
D1	Not used for USB network devices.
D2	No bus transmission allowed. No bus reception allowed. No interrupts can occur. Device context may be lost.
D3	Power may be removed from the device. Device context assumed to be lost. No bus transmission allowed. No bus reception allowed. No interrupts can occur.

In the examples below, the USB bus state B2 corresponds to the Suspend state specified in the *v1.0 USB Specification*, or later.

The first example, shown in Table 3-16, is for USB devices that do not implement the USB Common Class Interface Power Management feature.

Table 3-16. Example V_{id} and V_{op} States for USB Bus and Device States

Bus ACPI State	Interface ACPI State	Device State	V_{id} state	V_{op} state	Floppy Disk Storage Example	Network Interface Example
B0	n/a	D0	Full Power (<= 1.5W)	Full power (Wattage depends on Form Factor)	Drive controller fully operational and drive motor turning	NIC is fully operational
B0	n/a	D1	Full Power (<= 1.5W)	Low Power draw	n/a	n/a
B0	n/a	D2	Full Power (<= 1.5W)	Device may draw up to 5 uA / V_{op} System may supply 0 power	n/a	No bus transmission allowed. No bus reception allowed. No interrupts can occur. NIC can generate a wakeup event.
B0	n/a	D3	Low Power	Device may draw up to 5 uA / V_{op} System may supply 0 power	Drive controller not functional and context lost. Drive motor stopped.	No bus transmission allowed. No bus reception allowed. No interrupts can occur. NIC can generate a wakeup event.
B2	n/a	D2	Lower than low Power. Bus is in USB Suspend state. Maximum device current draw is 500uA.	Device may draw up to 5 uA / V_{op} System may supply 0 power	n/a	No bus transmission allowed. No bus reception allowed. No interrupts can occur. NIC can generate a wakeup event.
B2	n/a	D3	Lower than low Power. Bus is in USB Suspend state. Maximum device current draw is 500uA.	Device may draw up to 5 uA / V_{op} System may supply 0 power	Drive controller not functional and context lost. Drive motor stopped.	No bus transmission allowed. No bus reception allowed. No interrupts can occur. NIC can generate a wakeup event.
B3	n/a	D3	System must switch off V_{id} as the device is not ever required to not draw any	Device may draw up to 5 uA / V_{op} System may supply 0 power	Drive controller not functional and context lost. Drive motor stopped.	No bus transmission allowed. No bus reception allowed. No interrupts can occur.

Bus ACPI State	Inter face ACPI State	Device State	V_{id} state	V_{op} state	Floppy Disk Storage Example	Network Interface Example
			power off of V_{id}.			

The second example, shown in Table 3-17, is for USB interfaces that are part of a USB composite device, and that do implement the USB Common Class Interface Power Management feature. The model of a composite USB device that implements this Common Class feature is one of separately power-managed components: there is the device core logic, plus the logic that implements each of the one or more USB class interfaces.

- An example of a composite USB device that contains a storage interface is a USB video camera and Flash memory combination that enables the user to press a button on the camera to capture a still frame image in the Flash memory. For power management purposes, this example device is made up of the device core logic, the USB Imaging interface, and the USB Storage interface.
- An example of a composite USB device that contains a Network interface (or, in USB terms, a Communications interface) is a USB IP (Internet Protocol) telephone. This example device is made up of the device core logic, a Communications interface, a HID interface (for the keypad), an Audio In interface (for the speaker), and an Audio Out interface (for the microphone).

Table 3-17. Example V_{id} and V_{op} States for USB Bus and Device States

Bus ACPI State	Core Logic ACPI State	Class Inter-face State	V_{id} state	V_{op} state	Storage Class Interface Example	Communications Class Interface Example
B0	D0	D0	Full Power (<= 1.5W)	Full power (Wattage depends on Form Factor)	Storage interface fully operational. Video camera device core logic fully operational.	Communications interface fully operational. IP telephone core logic fully operational.
B0	D0	D1	Full Power (<= 1.5W)	Low Power draw	n/a	n/a
B0	D0	D2	Full Power (<= 1.5W)	Device may draw up to 5 uA / V_{op} System may supply 0 power	n/a	No bus transmission allowed. No bus reception allowed. No interrupts can occur. Communications class interface can generate a wakeup event.
B0	D0	D3	Low Power	Device may draw up to 5 uA / V_{op} System may supply 0 power	Storage interface in lowest power state. Video camera device core logic fully operational.	No bus transmission allowed. No bus reception allowed. No interrupts can occur. Communications class interface can generate a wakeup event.
B0	D1	D1	Full Power (<= 1.5W)	Low Power draw	n/a	n/a
B0	D1	D2	Full Power (<= 1.5W)	Device may draw up to 5 uA / V_{op} System may supply 0 power	n/a	No bus transmission allowed. No bus reception allowed. No interrupts can occur.

Bus ACPI State	**Core Logic ACPI State**	**Class Inter-face State**	**V_{id} state**	**V_{op} state**	**Storage Class Interface Example**	**Communications Class Interface Example**
						Communications class interface can generate a wakeup event.
B0	D1	D3	Low Power	Device may draw up to 5 uA / V_{op} System may supply 0 power	Storage interface in lowest power state. Video camera device core logic in low power state.	No bus transmission allowed. No bus reception allowed. No interrupts can occur. Communications class interface can generate a wakeup event.
B0	D2	D2	Full Power (<= 1.5W)	Device may draw up to 5 uA / V_{op} System may supply 0 power	n/a	No bus transmission allowed. No bus reception allowed. No interrupts can occur. Communications class interface can generate a wakeup event.
B0	D3	D3	Low Power	Device may draw up to 5 uA / V_{op} System may supply 0 power	Storage interface is not operational. Video camera device core logic not operational.	No bus transmission allowed. No bus reception allowed. No interrupts can occur. Communications class interface can generate a wakeup event.
B2	D2	D2	Lower than low Power. Bus is in USB Suspend state. Maximum device current draw is 500uA.	Device may draw up to 5 uA / V_{op} System may supply 0 power	n/a	No bus transmission allowed. No bus reception allowed. No interrupts can occur. Communications class interface can generate a wakeup event, but otherwise not operational.
B2	D2	D3	Lower than low Power. Bus is in USB Suspend state. Maximum device current draw is 500uA.	Device may draw up to 5 uA / V_{op} System may supply 0 power	Storage interface is not operational. Video camera device core logic is in low power state.	No bus transmission allowed. No bus reception allowed. No interrupts can occur. Communications class interface can generate a wakeup event, but otherwise not operational.
B3	D3	D3	System must switch off V_{id} as the device is not ever required to not draw any	Device may draw up to 5 uA / V_{op} System may supply 0 power	Storage interface is not operational. Video camera device core logic not operational.	No bus transmission allowed. No bus reception allowed. No interrupts can occur.

Bus ACPI State	**Core Logic ACPI State**	**Class Inter-face State**	**V_{id} state**	**V_{op} state**	**Storage Class Interface Example**	**Communications Class Interface Example**
			power off of V_{id}.			

3.10 Power Budgeting

The Device Bay Power Capabilities (DBPC) registers provide the OS with the information needed to provide power management and budgeting for a Device Bay subsystem. For information, see section 6.6.10 in the Device Bay Controller chapter and section 3.8.4.

4 Device Bay Connector Set Requirements

This section defines the connector set requirements for the Device Bay bays and the Device Bay devices.

4.1 Goals

The goals for the Device Bay connector set are:

- Industry Open and royalty free
- Ensured interoperability of all Device Bay devices in all Device Bay systems
- Connector set defined in such a way that it may be used for removable Device Bay devices or fixed internal devices, that is, cable-friendly
- Cost effective (including cable assemblies)
- One connector set requirement specification for all Device Bay form factors

4.2 Scope

Each Device Bay connector set supports one 1394 port, one USB port, the power requirements for the 1394 and/or USB device(s), and bay management signals for the Device Bay implementation. The Device Bay connector set does not support additional ports or system-dependent management signals such as the "removal request."

The requirements defined in this section apply to the mating interface only between the connector plug (which is in the device), and receptacle (which is in the bay) so that different connector types can be implemented to best suit the application in hand.

This section does not define any standard connector, i.e., it does not define:

- The mounting features of the Device Bay connector set
- The requirements for cabling or cable termination
- The methods on how the bay PCB connects to the other components of the system (such as the power supply).

Since these are not specification requirements, a separate document named the *Device Bay Design Guide* describes several connector and cable design examples with detailed information on termination, mounting, and implementation tips.

The following sections of this specification discuss details that are related to the connector set requirements:

- Section 3 defines the Device Bay power requirements and bus signal general descriptions.
- Section 5 defines the form-factor requirements for Device Bay devices, including the connector plug location in a device, and the hardware for removal of a device from a bay.
- Section 6 defines the requirements for Device Bay system-dependent management signals (control and status signals), such as removal request, bay status, and the software-controlled interlock.

4.3 General Description

The connector set consists of a plug connector that is located in a removable device, and a receptacle connector that is located in a bay. Figure 4-1 shows an example implementation of a plug connector and a receptacle connector.

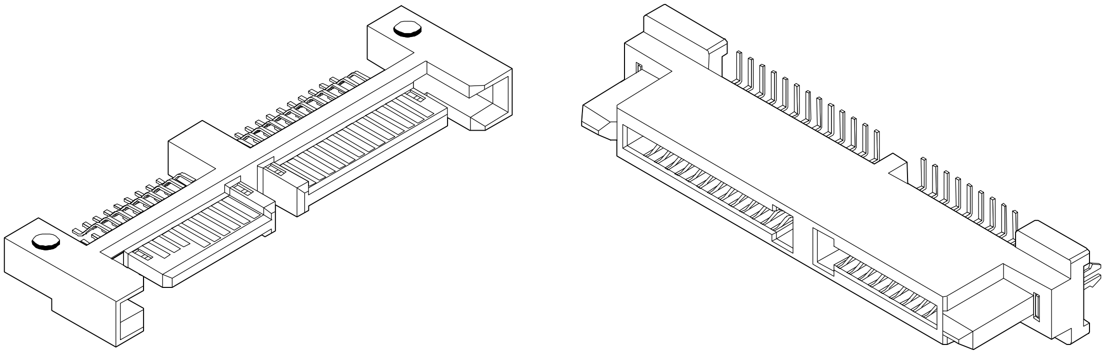

Figure 4-1. Device Bay Plug and Receptacle Connector Pair Examples

Figure 4-2 shows an example of a plug connector in a device.

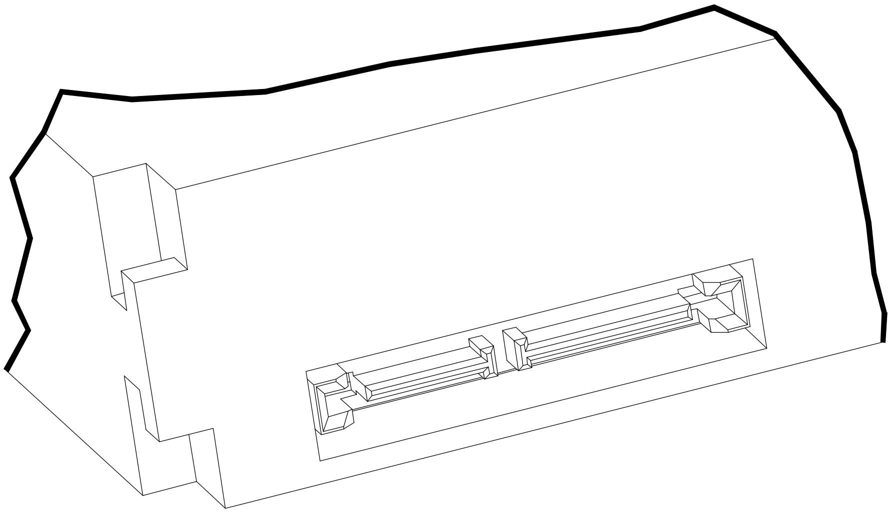

Figure 4-2. Example Device Bay Plug Connector in a Device

Figure 4-3 shows an example of a receptacle mounted in a bay on a Device Bay backplane. In a Device Bay system, there may be one or more bays, each having a receptacle terminated on a cable or mounted on a backplane PCB. See section 5 of this specification for the details on the locations of the plug in a device to ensure blind-mating.

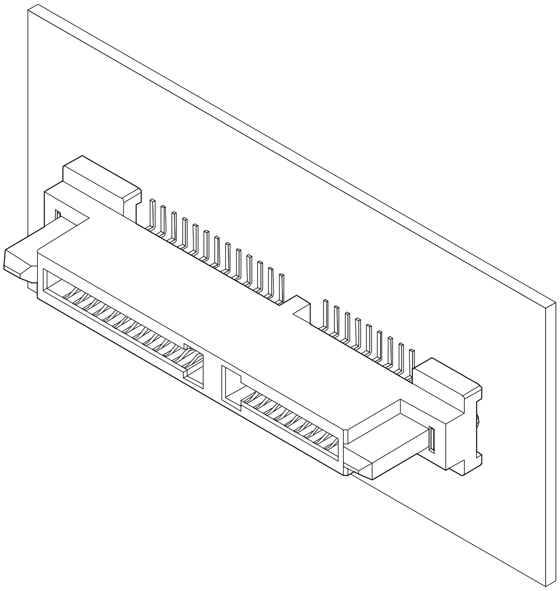

Figure 4-3. Example Device Bay Receptacle Connector in a Bay

A Device Bay device may be a 1394 device, a USB device, or a compound device (having both 1394 and USB interfaces). The Device Bay connector set has the following pin groups and associated grounds:

- 1394 – interfaces to a 1394 PHY
- USB – interfaces to a USB hub
- Miscellaneous bay management signals – interfaces to a DBC
- Power – from the power supply

The important features of the Device Bay connector are:

- Blind-mating.
- Hot insertion/removal by means of staggered mating contact pin lengths.
- High insertion/removal cycle durability (minimum of 2,500 cycles).
- Support for the following signals:
- 1394[1]
- USB
- Bay management
- Supports the following power supply levels:
- 3.3 volt
- 5.0 volt
- 12.0 volt
- Both the plug and the receptacle can be mounted on either PCB or cable (implementation-dependent termination).

---

[1] The connector must support IEEE 1394-1995 data rates and must be capable of supporting 1394b data rates up to 3.2Gbps (S3200).

4.4 Electrical Descriptions

This section specifies the Device Bay connector set's signal descriptions, pin configurations, pin bussing options, and grounding requirements.

4.4.1 Pin Descriptions

A Device Bay connector consists of the reserved pin, ground pins, and pins for the following groups:

- 1394
- USB
- Bay management
- Power

4.4.1.1 Reserved Pin

The Reserved pin must be a no-connect pin.

4.4.1.2 1394 Pins

TPA, TPA#

Differential signal pair A from a 1394 PHY, as defined in the 1394 specifications. For a reference, see section 1.5 of this Specification.

TPB, TPB#

Differential signal pair B from a 1394 PHY, as defined in the 1394 specifications. For a reference, see section 1.5 of this Specification.

Note:

- Only the data signals (and not the power) specified in the IEEE 1394 specification are used in the Device Bay connector set.
- Unlike the 1394 external cable assembly[2] defined in the 1394 specification, crossing of the 1394 TPA and TPB differential signals is accomplished at the connector pair by assigning different pin names on the device plug and the bay receptacle. For more information, see section 4.4.2 of this specification.

4.4.1.3 USB Pins

D+, D-

Differential signal pair as defined in the *Universal Serial Bus Specification*. For a reference, see section 1.5 of this specification.

Note: Only the data signals (and not the power) specified for the USB specification are used in the Device Bay connector set.

---

[2] In the 1394 specification, crossing of the 1394 TPA and TPB differential signal is accomplished within the external cable assembly.

4.4.1.4 Bay Management Pins

1394PRSN#

The presence signal for a 1394 device. This static pin, when asserted low, means a 1394 device is present. When not asserted low, it means no 1394 device is present.

If a device uses the 1394 interface, this signal must be connected to the device ground with a maximum resistance of 10 Ω; otherwise, it must be left open.

The bay must provide a pull-up resistor of no less than 10 KΩ to the power supply rail of the DBC. For more information, see section 3.3.3.1 of this specification.

USBPRSN#

The presence signal for a USB device. This static pin, when asserted low means a USB device is present. When not asserted low, it means no USB device is present.

If a device uses the USB interface, this signal must be connected to the device ground with a maximum resistance of 10 Ω; otherwise, it must be left open.

The bay must provide a pull-up resistor of no less than 10 KΩ to the power supply rail of the DBC. For more information, see section 3.3.3.1 of this specification.

DEV_ACT#

This is an active low, open-drain/open-collector signal driven by the device to indicate device activities. This signal is optional for both the device and the bay. A system may use this signal to indicate the device activity, for example, by a flashing display.

When supported by the device, a resistance to the device ground of no more than 10 Ω must be provided to indicate device activities. When unused, this signal must be left unconnected in the device.

When supported by the bay, the bay must provide a pull-up resistor of no less than 10 KΩ to the power supply rail of the DBC. When unused, this signal must be left unconnected in the bay.

4.4.1.5 Power Supply Pins

This section describes the power supply pin counts and termination requirements. For more information on the voltage and current requirements of the power supply pins, see section 3 of this specification. For the requirements on the ground pins, see section 4.4.4.

V_{id}

There is one pin for 3.3 V identification voltage for the system to query the device for its interface type and its power requirements. This pin must be used, hence terminated, in all devices.

V_{33}

There are six pins (B1 through B6) for 3.3 V operating voltage. If a device uses 3.3 V then it must terminate at least three adjacent pins – B1 through B3, or B3 through B6. If a device does not use 3.3 V then it is optional to terminate any of B1 through B6.

V_5

There are three pins (B16 through B18) for 5.0 V operating voltage. If a device uses 5.0 V then it must terminate all three pins – B16 through B18. If a device does not use 5.0 V then it is optional to terminate any of B16 through B18.

V_{12}

There are three pins (B7 through B9) for 12.0 V operating voltage. If a device uses 12.0 V then it must terminate all three pins – B7 through B9. If a device does not use 12.0 V then it is optional to terminate any of B7 through B9.

All bays must support (i.e., terminate) V_{id}, V_{33} and V_5. DB32 bays must also support V_{12}.

4.4.2 Pin Configuration

The Device Bay connector has 44 pins, arranged in two compartments or segments as follow:

- The signal segment has 26 pins arranged in two rows with 13 pins in each row.
- The power segment has 18 pins arranged in two rows with 9 pins in each row.

Figures 4-4 and 4-5 illustrate the pin configuration on the plug and the receptacle, respectively.

A14	Gnd (1st mate)	Gnd (1st mate)	A1
A15	Gnd (1st mate)	TPA	A2
A16	Gnd (1st mate)	TPA#	A3
A17	Gnd (1st mate)	Gnd (1st mate)	A4
A18	Gnd (1st mate)	TPB	A5
A19	Gnd (1st mate)	TPB#	A6
A20	Gnd (1st mate)	Gnd (1st mate)	A7
A21	Gnd (1st mate)	1394PRSN#	A8
A22	Gnd (1st mate)	DEV_ACT#	A9
A23	Gnd (1st mate)	USBPRSN#	A10
A24	Gnd (1st mate)	D+	A11
A25	Gnd (1st mate)	D-	A12
A26	Reserved	V_{id} (1st mate)	A13
Individual key for the Signal Segment Gap Individual key for the Power Segment			
B10	Gnd (1st mate)	V_{33}	B1
B11	Gnd (1st mate)	V_{33}	B2
B12	Gnd (1st mate)	V_{33}	B3
B13	Gnd (1st mate)	V_{33}	B4
B14	Gnd (1st mate)	V_{33}	B5
B15	Gnd (1st mate)	V_{33}	B6
B16	V_5	V_{12}	B7
B17	V_5	V_{12}	B8
B18	V_5	V_{12}	B9

Figure 4-4. Device Bay Connector Plug Pin-out
(Note: All pins that are not marked for the mating sequence are last-mates.)

There are two levels of mating of the contact pins. The mating sequence must be as follows:

- All the ground pins and the V_{id} are 1st mates
- The reserved pin, all the supply pins, and the signal pins are last-mates

It should be noted that only the device plug connector, which is to mate directly with the receptacle connector in the bay, requires the mating sequence. Any other connector plug or receptacle in a system or a device is not required to have the mating sequence. For more information about the mating contact pin lengths, see section 4.5.1.5 of this specification.

"Crossing[3]" of the 1394 TPA and TPB signals is accomplished at the connector pair by assigning different pin names on the device plug and the bay receptacle. Therefore, the 1394 differential signal pins A2, A3, A5, and A6 are defined differently between the plug and the receptacle.

For some implementations, the 1394 signals may be routed to the bay receptacle directly from the PHY on a PCB. For some other implementations, the 1394 signals may be routed to the bay receptacle after going through a few stages of connector pairs and cable assemblies. In this case, it is important to note that all the connectors within a system enclosure that use the Device Bay connector pin configuration will follow the "Device Bay Receptacle Pin Configuration" (Figure 4-5) regardless of the connector type being a plug or a receptacle. Similarly, there may be cable assemblies within a device, in which case all the connectors within a device that use the Device Bay connector pin configuration will follow the "Device Bay Plug Pin Configuration" (Figure 4-4) regardless of the connector type being a plug or a receptacle. This is to ensure proper crossing of the 1394 TPA and TPB signals.

A1	Gnd	Gnd	A14
A2	TPB	Gnd	A15
A3	TPB#	Gnd	A16
A4	Gnd	Gnd	A17
A5	TPA	Gnd	A18
A6	TPA#	Gnd	A19
A7	Gnd	Gnd	A20
A8	1394PRSN#	Gnd	A21
A9	DEV_ACT#	Gnd	A22
A10	USBPRSN#	Gnd	A23
A11	D+	Gnd	A24
A12	D-	Gnd	A25
A13	V_{id}	Reserved	A26
Individual key for the Signal Segment Gap Individual key for the Power Segment			
B1	V_{33}	Gnd	B10
B2	V_{33}	Gnd	B11
B3	V_{33}	Gnd	B12
B4	V_{33}	Gnd	B13
B5	V_{33}	Gnd	B14
B6	V_{33}	Gnd	B15
B7	V_{12}	V_5	B16
B8	V_{12}	V_5	B17
B9	V_{12}	V_5	B18

Figure 4-5. Device Bay Connector Receptacle Pin-out
(Note: All contacts have the same length.)

---

[3] For the 1394 differential signals, the TPA/TPA# of the device's PHY should be interfaced to the TPB/TPB# of the bay's PHY, respectively. Similarly, the TPB/TPB# of the device's PHY should be interfaced to the TPA/TPA# of the bay's PHY, respectively. See the *1394-1995 IEEE Standard for a High Performance Serial Bus* specification for details. For the 1394 external (also known as "walkup") connectors/cable, TPA/TPA# of one end goes to TPB/TPB# of the other end, respectively, of the cable.

For reliability[4] and for consistent insertion/removal forces, all device plugs must have contact pins in the mating area that meet the mechanical and electrical requirements in this specification. All bay receptacles contact pins with defined functional use must meet the electrical and mechanical requirements in this specification. Position A26 noted as "Reserved" may be selectively loaded at connector manufacturers discretion on the bay receptacles only until such time A26 pin function becomes defined in use by this specification.

Beyond the mating interface requirements defined, system designers have the option to terminate certain pins depending on the application. For example:
- Not terminating ground pins A14 through A25 of the signal segment on the cable. See section 4.4.4.2 for more details.

Similarly device designers have the option to terminate certain pins on the PCB. For example:
- A 1394 mobile device may leave the USBPRSN# pin, the USB differential signal pins, and the 12V pins unconnected on its PCB.
- If a device can operate at 3.3V and the power required is not more than 1.5W then it can be powered only by the V_{id}. For this type of device, all the pins in the power segment may be left unconnected provided that the other requirements defined in this specification are met. For example, the connector plug must have enough mechanical retention on the PCB to meet the requirements defined in section 5 of this specification.

4.4.3 Pin Busing Options

The connector plug and receptacle have pins that can be bused together for certain application requirements; for example, solder termination, thermal characteristics, and so on. Pin bussing may be done in the mounting or termination area of the plug and the receptacle. Pin busing may also be done in the mating area of the plug, but must not be done in the mating area of the receptacle in order to keep the insertion/removal force consistent.

The pin busing options are as follows:
- In the signal and the power segments, two or more ground pins may be bused together.
- In the power segment:
  - Two or more V_{33} pins may be bused together.
  - Two or more V_5 pins may be bused together.
  - Two or more V_{12} pins may be bused together.

It should be noted that the power supply pins must comply with the termination requirements defined in section 4.4.1.5, regardless of the power pins being bused or not.

4.4.4 Grounding Options

This section describes the pin-grounding options for the Device Bay connector set. The grounding options are different for the device side and the bay side.

4.4.4.1 Device-Side

In a device, all the grounding pins in the signal segment, all the grounding pins in the power segment, and the ESD grounding pads on the device case must be connected to the same ground—the device ground.

---

[4] The connector durability (i.e., insertion/removal cycle) depends on its contact plating durability. Device plug contacts as specified in this specification must be present to assure damage will not be introduced to bay receptacle contact pins plating.

4.4.4.2 Bay-Side

The bay is required to provide all the power supplies specified in section 3 of this specification. The receptacle in the bay may be mounted on a PCB or terminated on a cable(s).

In a PCB implementation, the following grounds are required:

- All the grounding pins of the power segment must be connected to the power supply ground.
- All the grounding pins of the signal segment must be connected to the signal ground of the other Device Bay logic; for example, DBC.
- The grounding pins in the signal segment and the power segment may or may not be connected together in the bay.

For cable-terminated applications, the following ground pin connections are required:

- For the signal segment, the grounding pins A14 through A25 may or may not be terminated on the cable. These ground pins are intended for impedance control only, not for the signal return current. Since all the connectors are required to have the mating contacts present, and the plug in the device is required to have a common ground for all its grounding pins, the mated connector pair in the signal segment will always have proper grounding.
- For shielded cables, the cable shield may be terminated to one or more pins of A14 through A25.
- If a device uses the USB and/or 1394 interface, the grounding pins A1, A4, and A7 must be connected to the system ground for signal and V_{id} returns.
- All the grounding pins (B10 through B15) in the power segment must be terminated and connected to the power supply ground.

4.5 Mechanical Descriptions

This section specifies the connector mating features and labeling. The mechanical requirements for usage of the connector set are defined in section 5 of this specification.

For Device Bay applications, the connector pair provides tapered lead-in features to provide blind-mating and misalignment correction of about ±2.00 mm.

The following features are also provided on the connector set for cabled applications:

- The two segments of the plug and the receptacle are individually polarized to avoid 180-degree disorientation of a cable-terminated receptacle on each segment.
- The connector plug has passive detents to mechanically retain a cable-terminated receptacle.

The termination contacts are implementation-dependent, for example, through-hole, surface-mount, right-angled, and vertical. A separate document, the *Device Bay Design Guide*, illustrates several connector and cable assembly implementation examples.

4.5.1 Connector Mating Features

The mating portion of the plug and the receptacle must be 48.0 mm wide and a 5.0 mm high. The depth of the plug and the receptacle can vary depending on the application, that is, the mounting and the termination choices. See Figures 4-6 through 4-11 for mating area details. The interpretation of the drawings must be in accordance with “ASME Y14.5M-1994, Dimensioning and Tolerancing.”

4.5.1.1 Mating Mechanical Retention

The signal segment and the power segment of a plug must have passive detents, as shown in Figure 4-6(b). In non-Device Bay applications, raised plastic guides, or bumps, inside a receptacle may be used to engage the detents and ensure that the connector is retained. This allows a Device Bay device such as a hard disk drive to be used as an internal drive. In this environment, a power cable assembly may be used between a power supply and a device, and a signal cable assembly may be used between a system board and a device. Notice that the durability of the Device Bay connector set is defined to be in the thousand’s of cycles, and the durability of the detents/bumps mechanism is in the ten’s of cycles. For this reason, a Device Bay receptacle is not required to have the raised plastic bumps.

4.5.1.2 Plug Mating Dimensions

Figure 4-6(a) shows the dimensions (P1 through P33), and tolerances (T1 through T5) of the Device Bay plug mating portion. Figure 4-6(b) illustrates the structural drawing. Figure 4-6(a) and (b) layouts are oriented to be viewed on side-by-side pages.

P1	5.40	P19	47.50	T1	0.05
P2	1.27	P20	1.00	T2	0.08
P3	10.16	P21	13.02	T3	0.10
P4	15.24	P22	18.10	T4	0.15
P5	0.80	P23	0.65	T5	0.20
P6	6.65	P24	1.60 [Note(a)]		
P7	36.40	P25	0.50 [Note(a)]		
P8	4.43	P26	0. 35		
P9	0.65	P27	3.90		
P10	1.40	P28	0.38		
P11	4.50	P29	2.44		
P12	2.30	P30	1.20		
P13	2.80	P31	1.50		
P14	3.00	P32	6.01		
P15	45.5	P33	0.93		
P16	48.00				
P17	5.00				
P18	1.55				

Note:

(a) The plastic must not rise above the level of the contact metal.

Figure 4-6(a). Device Bay Plug Mating Portion Dimension and Tolerance Chart (in mm)

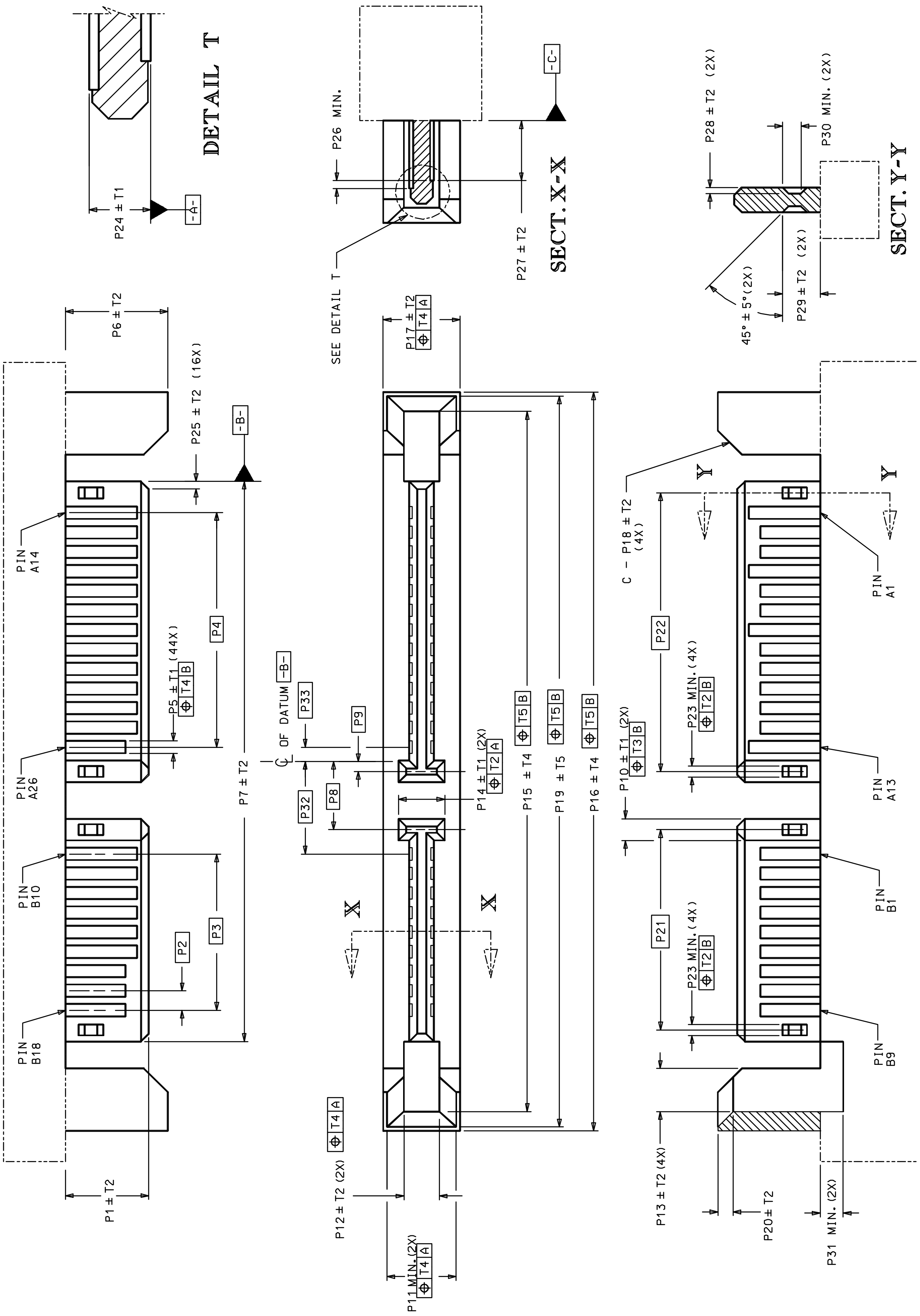

Figure 4-6(b). Device Bay Plug Mating Portion

4.5.1.3 Receptacle Mating Dimensions

Figure 4-7(a) shows the dimensions (R1 through R27), and tolerances (T1 through T5) of the Device Bay receptacle mating portion. Figure 4-7(b) illustrates the structural drawing. Figure 4-7(a) and (b) layouts are oriented to be viewed on side-by-side pages.

R1	36.60	R15	0.30	T1	0.05
R2	4.43	R16	6.50	T2	0.08
R3	0.65	R17	see text	T3	0.10
R4	10.16	R18	1.85	T4	0.15
R5	15.24	R19	1.90	T5	0.20
R6	1.41	R20	5.60		
R7	1.27	R21	0.25		
R8	3.30	R22	5.00		
R9	0.40	R23	1.74		
R10	1.70	R24	1.00		
R11	39.00	R25	1.00		
R12	44.80	R26	6.01		
R13	48.00	R27	0.93		
R14	42.70				

Figure 4-7(a). Device Bay Receptacle Mating Portion Dimension and Tolerance Chart (in mm)

In Figure 4-7(b), R17, R13, and R22 are shown to describe the "shoulder" region's height, length and thickness, respectively, of a receptacle. When a bay receptacle is fully mated to a device plug the blind-mate features of the two connectors bottom-out at the "platform" immediately above the R17. Because the plug connector is recessed within the device, during a fully-mated condition, the shoulder region (R13, R22 and part or all of R17) of a receptacle will be inside a device enclosure. Therefore, a receptacle's shoulder region must conform to the dimensions R13 and R22 as specified in Figure 4-7(a). R17 is implementation dependent, but it must meet the following condition:

- As specified in section 5, DB13 and DB20 devices require the plug to be recessed 0.50±0.25mm, but DB32 requires the plug to be recessed 2.00±0.50mm, inside the device enclosure. It should be noted that the plug skew tolerances are part of the recessed tolerances. See section 5 for the skew tolerance. To ensure a fully-mated condition, R17 must be 0.75mm minimum for DB13 and DB20 applications, and 2.50mm minimum for DB32 applications.

The following variations are possible for R17 implementations:

- A receptacle's R17 for DB13 and DB20 applications may be smaller than for DB32 applications.
- R17 for a cable-terminated receptacle may be different from R17 for a PCB-mount receptacle.
- R17 for a PCB-mount receptacle with mounting holes may be larger than PCB-mount receptacle without mounting holes (e.g., with fork-locks), to have enough clearance between the back surface of the device and the top of the screw-heads.

It should be noted that R17 for an implementation will have direct implication on the total contact pin length. Therefore other specification parameters, such as the LLCR (Table 4-4 in Section 4.6.3) must also be met for a receptacle implementation.

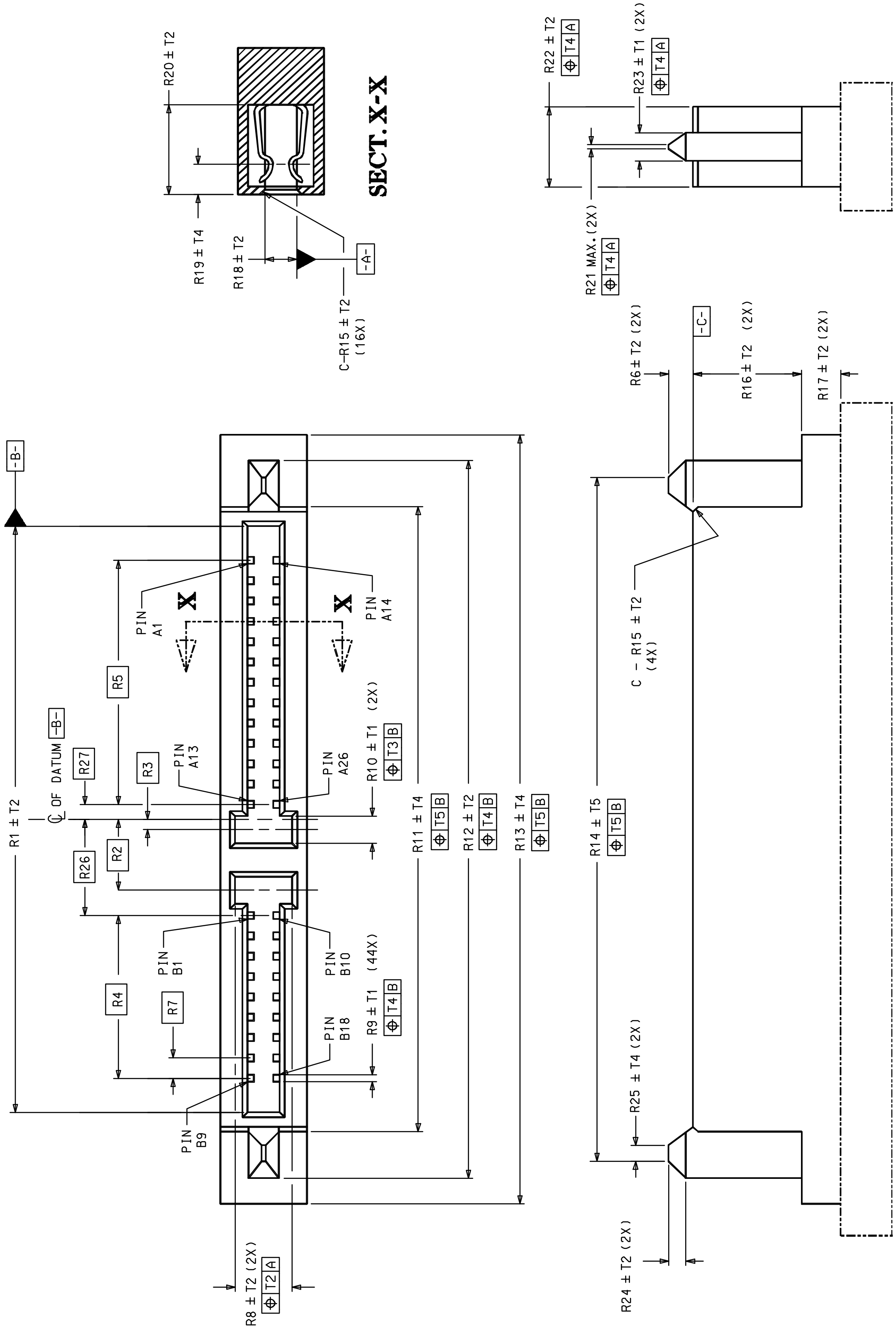

Figure 4-7(b). Device Bay Receptacle Mating Portion

4.5.1.4 Blind-Mating

In addition to the chamfered plug tongue and receptacle cavity, the receptacle has chamfered wings and the plug has chamfered grooves on the side ends to blind-mate the connector pair. The maximum blind-mate misalignment tolerance is ±2.00 mm for the horizontal axis, and ±1.975 mm for the vertical axis, as illustrated in Figures 4-8 and 4-9, respectively. Any skew angle of the plug, with respect to the receptacle, will reduce the blind-mate tolerance. See section 5 for the skew tolerances.

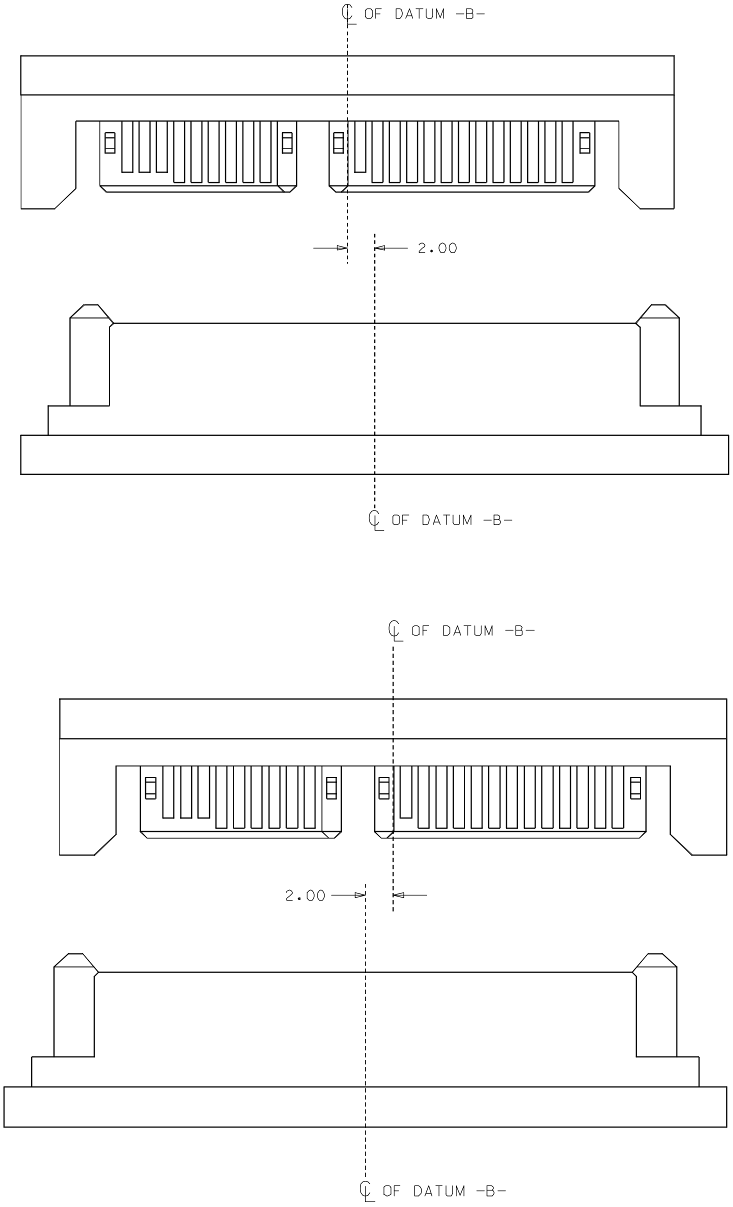

Figure 4-8. Device Bay Connector Pair Blind-Mate Horizontal Misalignment ±2.00 mm Tolerance.

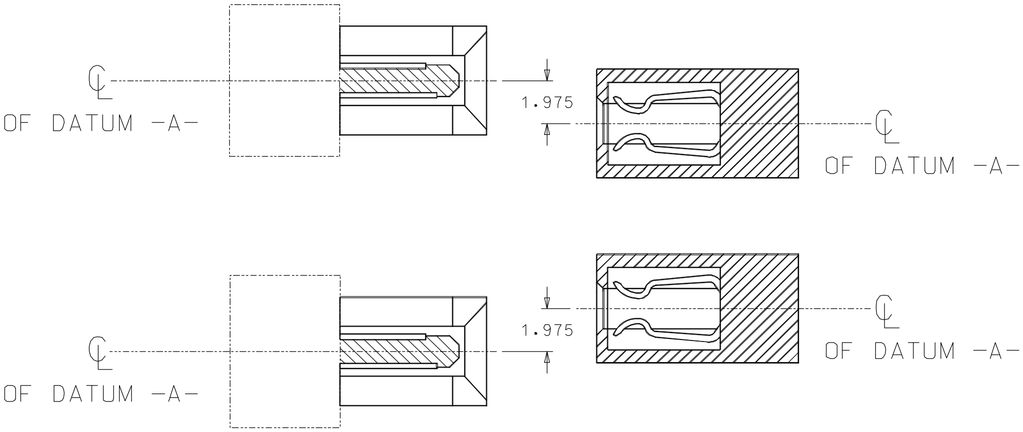

Figure 4-9. Device Bay Connector Pair Blind-Mate Vertical Misalignment ±1.975 mm Tolerance.

4.5.1.5 Mating Contact Pins Dimensions

All the mating contact pins have 1.27 mm [0.050"] pitch. All mating contacts must be present in the plug and the receptacle connectors. This is for reliability reasons, and for maintaining consistent insertion and removal forces.

The contact pin width and length requirements in the mating area for the plug and the receptacle are shown in Figures 4-6 and 4-7, respectively. Notice that for a receptacle, all the contacts in the mating area have the same length, and for a plug, the contacts in the mating area have varying lengths depending on the mating sequence specified in Figure 4-4.

4.5.1.6 Fully-Mated Conditions and Minimum Wipe

The plug and the receptacle bottom-out at the blind-mate features when they are fully mated, as shown in Figure 4-10. The following conditions are true:

- If there is a gap between the top surface of the plug's base (datum -C- for the plug shown in Figure 4-10) and the receptacle's top surface (datum -C- for the receptacle shown in Figure 4-10) then it is not more than 0.31 mm.
- The tip of the plug's tongue does not touch the bottom of the receptacle's cavity, as shown in Figure 4-11.
- The *minimum wipe* (that is, the worst-case distance from the tip of a plug contact pin to the tangential mating point of a receptacle contact pin as shown in Figure 4-11) is 1.46 mm.

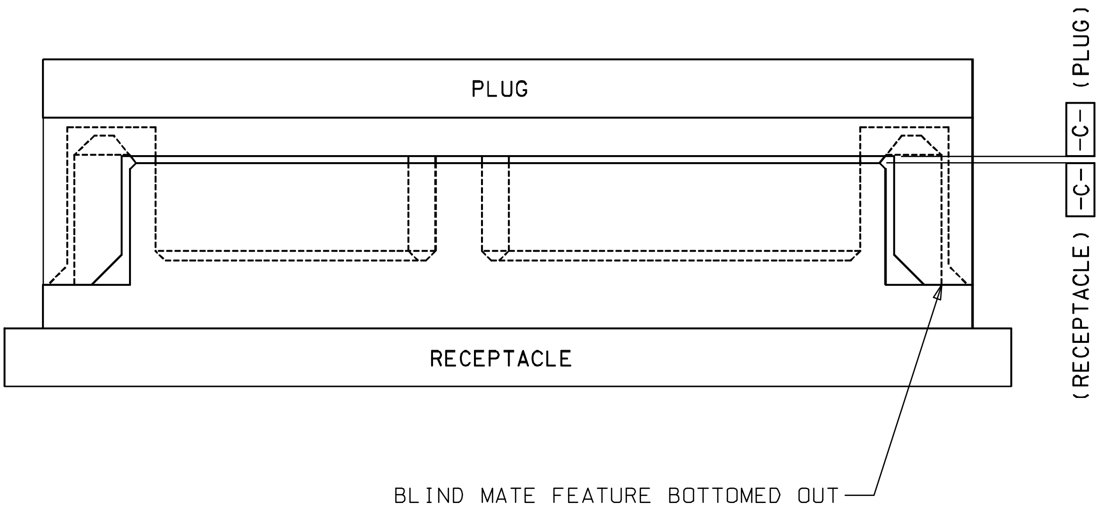

Figure 4-10. Connector Pair Fully-Mated (Worst-Case)

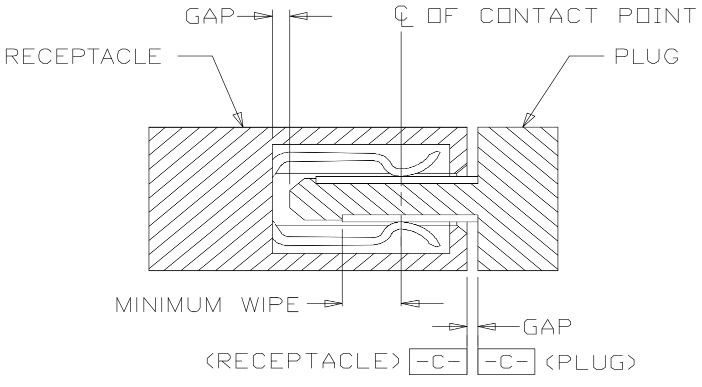

Figure 4-11. Connector Pair Fully Mated (Worst-Case) Cross-Section View

4.5.1.7 Contact Material and Plating

Table 4-1 shows the minimum requirements for Device Bay connector set’s contact material and plating.

Table 4-1. Minimum Contact Material and Plating Requirements

Parameter	Minimum Requirements
Material	Copper-based alloy.
Mating Side Plating	Exposed underplate or base material is not allowed in the mating area. 1.27 µm (50 µin) minimum Ni with either 0.76 µm (30 µin) minimum Au or 0.76 µm (30 µin) minimum 80/20 Pd/Ni with 0.076 µm (3 µin) minimum Au overplate.
Solder Side Plating	Exposed base material is allowed in small areas where the contact is excised from its carrier strip or bandolier. The plating may be either Sn/Pb plating or Pd/Ni with Au flash plating. Preferred: 1.27 µm (50 µin) minimum Ni with 5.08 µm (200 µin) minimum Sn/Pb. Alternate: 1.27 µm (50 µin) minimum Ni with 0.76 µm (30 µin) minimum 80/20 Pd/Ni with 0.076 µm (3 µin) minimum Au overplate.

4.5.2 Connector Set Labeling

This section defines the pin number, speed grade, and logo labeling requirements for the Device Bay connector set.

4.5.2.1 Pin Number Labeling

The plug and the receptacle housings must at least have the pin numbers labeled as shown in Figures 4-12 and 4-13, respectively.

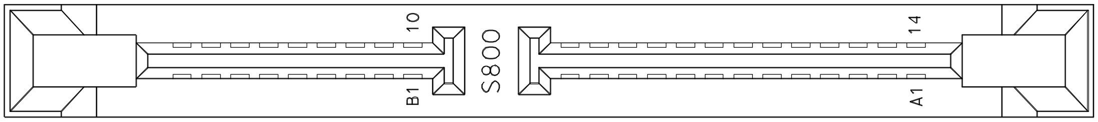

Figure 4-12. Example Plug Labeling.

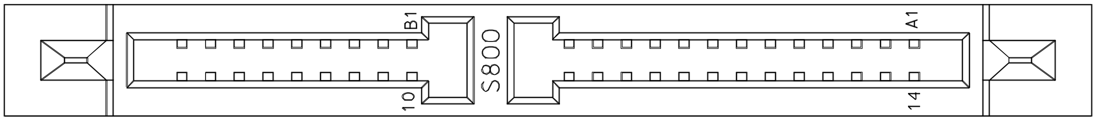

Figure 4-13. Example Receptacle Labeling.

4.5.2.2 Speed Grade Labeling

A Device Bay connector may label the maximum 1394 data rate that it can support for manufacturing identification. Speed grade labeling is optional. If implemented then the speed grade label must be one of the following speeds:

- S400 (for 400 Mbps)
- S800 (for 800 Mbps)
- S1600 (for 1.6 Gbps)
- S3200 (for 3.2 Gbps)

S800 is used as an example in Figure 4-12 and 4-13 to illustrate the location of the speed grade label on the plug and the receptacle, respectively.

4.5.2.3 Logo Labeling

Labeling on a Device Bay connector with the connector manufacturing logo or Device Bay logo is optional.

4.6 Connector Performance Parameters and Test Methods

This section specifies Device Bay connector performance parameters, test procedures, and the test result requirements, in order to ensure component integrity during typical application, handling, or assembly processes.

Before beginning the test procedures specified in this section, the connectors must have been stored for at least 24 hours in the non-mated state under the following environmental conditions. Unless otherwise specified by the Electronics Industries Association (EIA) procedure or test details, all measurements must be performed under the following conditions:

- Mated
- Temperature: 15° to 35° C
- Relative Humidity: 20% to 80%
- Atmospheric Pressure: 650 mm to 800 mm of Hg

If the EIA test is specified without a letter suffix in the test procedures, then the latest approved version of that test must be used.

4.6.1 Signal Electrical Parameters

This section defines a set of electrical parameters for a mated connector pair, not including PCB traces or cable wires. To clarify further, a connector designed for cable-termination should be tested without a cable to comply with the requirements defined in this section. See section 3 and Device Bay Design Guide for system level electrical parameters.

Table 4-2. Signal Electrical Parameters, Test Procedures, and Requirements Through a Mated-Pair

Parameter	**Procedure**	**Requirements** [Note(a)]
Differential Crosstalk Isolation [Note(b)]	EIA 364-90	>50 dB (<500 MHz) >40 dB (<2 GHz)
Differential Insertion Loss [Note(c)]	See Note(d) and (e).	<0.2 dB (250 MHz) <0.4 dB (500 MHz) <0.8 dB (1 GHz) <2.0 dB (2 GHz)
Propagation Delay		<200 ps
Single-Ended Signal Characteristic Impedance	EIA 364-67	50 Ω±10%
Differential Signal Impedance	EIA 364-67	100 Ω±10%

Note:

a) The signal toggle (or transition) rates of 250MHz, 500MHz, 1GHz and 2GHz are associated with 1394b's 8b/10b data rates of 400Mbps, 800Mbps, 1.6Gbps and 3.2Gbps, respectively. For example, 1394b's 3.2Gbps data rate is equivalent to 4Gbaud/sec symbol rate due to 8b/10b encoding. The maximum transition frequency for a signal toggling at 4Gbaud/sec is 2GHz.

b) The differential crosstalk isolation is the isolation measured between the two differential signal pairs through a mated connector pair.

c) The differential insertion loss (or attenuation) is the insertion loss measured on a differential signal pair through a mated connector pair.

d) Use a network analyzer, power splitters/combiners, and test cables/connectors that are rated for at least the frequency of 10% higher than the targeted maximum frequency. For example, use power splitters rated at least for 2.2GHz to measure 2.0GHz.

e) To minimize the measurement errors, use semi-rigid coax cables or micro-probing stations to measure the insertion losses. If connectors are to be mounted on test PCB's and there are traces and SMA connectors for the data path, then the network analyzer used should be "through" calibrated using the calibration traces very similar to the entire data path traces.

4.6.2 Examination of Connectors

Table 4-3. Connector Examination Procedures and Requirements

Parameter	Procedure	Requirement
Visual and Dimensional Inspections	EIA 364-18 Visual, dimensional and functional per applicable quality inspection plan.	Meets product drawing requirements.

4.6.3 Housing and Contact Electrical Parameters

Each connector assembly must meet the following requirements:

Table 4-4. Housing and Contact Electrical Parameters, Test Procedures, and Requirements

Parameter	Procedure	Requirement
Insulation Resistance	EIA 364-21 After 500 VDC for 1 minute, measure the insulation resistance between the adjacent contacts of mated and unmated connector assemblies.	1000 MΩ minimum
Dielectric Withstanding Voltage	EIA 364-20 Method B Test between adjacent contacts of mated and unmated connector assemblies.	The dielectric must withstand 500 VAC for 1 minute at sea level.
Low Level Contact Resistance (LLCR)	EIA 364-23 Subject mated contacts assembled in housing to 20 mV maximum open circuit at 100 mA maximum	30 mΩ maximum
Low Level Contact Inductance	EIA 364-69	5 nH maximum Unmated per contact
Contact Capacitance	EIA 364-30	2 pF maximum Unmated per contact
Current Rating (Signal Segment)	EIA 364-70 Method B.	1.0 A minimum at 250 VAC. The temperature rise above ambient shall not exceed 30°C at any point in the connector when contact positions are powered. The ambient condition is still air at 25°C.
Current Rating (Power Segment)	As specified in section 4.6.9.	As specified in section 4.6.9.

4.6.4 Housing and Contact Mechanical Parameters

Each connector pair must meet the following requirements:

Table 4-5. Housing and Contact Mechanical Parameters, Test Procedures, and Requirements

Parameter	**Procedure**	**Requirement**
Insertion force	EIA 364-13 Measure force necessary to mate the connector assemblies at max. rate of 12.5 mm (0.492") per minute.	38.8N [8.72 lbs.] maximum(a)
Removal force	EIA 364-13 Measure force necessary to unmate the connector assemblies at maximum rate of 12.5 mm (0.492") per minute.	6.47N [1.45 lbs.] minimum(a)
Durability	EIA 364-09 2,500 insertion and removal cycles at maximum rate of 200 cycles per hour.	See note (b).
Physical shock	EIA 364-27 Condition H Subject mated connectors to 30 g's half-sine shock pulses of 11 msec duration. Three shocks in each direction applied along three mutually perpendicular planes for a total of 18 shocks.	No discontinuities of 1 μs or longer duration.
Vibration (random)	EIA 364-28 Condition V Test letter A Subject mated connectors to 5.35 g's RMS. 15 minutes in each of three mutually perpendicular planes.	No discontinuities of 1 μs longer duration.

(a) The insertion and removal forces specified in this table are for the Device Bay connectors. Note that Device Bay receptacle connectors do not have the retention bumps, though the plugs have the retention detents. Non-Device Bay applications of the connector may have different insertion and removal forces. For reference purposes, the insertion and removal forces per contact pin are 0.88N [0.20 lb.] max. and 0.15N [0.03 lb.] min., respectively.

(b) Must meet EIA 364-18 Visual Examination requirements, show no physical damage, and must meet requirements of additional tests as specified in the test sequence in section 4.6.8.

4.6.5 Environmental Parameters

Table 4-6. Environmental Parameters, Test Procedures, and Requirements

Parameter	Procedure	Requirement
Humidity	EIA 364-31 Method II Test Condition A. Subject mated connectors to 96 hours at 40°C with 90% to 95% RH.	See note(a).
Temperature life	EIA 364-17 Test Condition III Method A. Subject mated connectors to temperature life at +85°C for 250 hours.	See note(a).
Thermal shock	EIA 364-32 Test Condition I. Subject mated connectors to 10 cycles between –55°C and +85°C.	See note(a).

(a) Must meet EIA 364-18 Visual Examination requirements, show no physical damage, and must meet requirements of additional tests as specified in the test sequence in section 4.6.8.

4.6.6. Additional Requirements

Table 4-7. Additional Requirements

Parameter	Procedure	Requirement
Flammability	UL 94V-0	Material certification or certificate of compliance required with each lot to satisfy the Underwriters Laboratories follow-up service requirements.

Table 4-8. Additional Requirements for Surface-Mount Connectors Only

Parameter	Procedure	Requirement
Housing temperature	Temperature steps of 225° C for 1 minute, and 235 °C for 15 seconds.	Without preconditioning, material must withstand the procedure without blistering, discoloring, or degrading original material form or finish.
Warpage	Measure the warpage (bow and twist) before and after the exposure to 225° C for one minute, and 235° C for 15 seconds.	Maximum warpage of 0.1 mm (0.004") over the length of connector before and after the exposure.
Solderability	EIA 364-52	Contact tails shall pass 95% coverage after 1 hour steam age.

Table 4-9. Product Drawing Requirements for Surface-Mount Connectors Only

Solder tails	Entire pin and metal tab retention feature is to be plated. Exposed base metal at the end of solder tail (due to fabrication process) is acceptable. If solder tail fabrication requires a shearing process, a shear direction of bottom to the top of solder tail is preferred to allow for plating wipe at the end of tail.
Solder tail co-planarity	Co-planarity of 0.1 mm (0.004")

4.6.7 Sample Selection

Samples must be prepared in accordance with applicable manufacturers' instructions and must be selected at random from current production. Unless otherwise specified, a test group must consist of a minimum of five connector pairs. From these connector pairs, a minimum of 20 contact pairs per mated connector must be selected and identified. This will provide 100 data points per test group for a good statistical representation of the test result.

4.6.8 Test Sequence

Table 4-10 shows the Device Bay connector test sequences for five groups of tests.

Table 4-10. Device Bay Connector Test Sequences

Test Group ➔	**A**	**B**	**C**	**D**	**E**
Test or Examination ➔					
Examination of the connector(s)	1, 5	1, 9	1, 5	1, 10	1, 3
Low-Level Contact Resistance (LLCR)	2, 4	3, 7	2, 4		
Insulation resistance				4, 8	
Dielectric withstanding voltage				5, 9	
Low-Level Contact Inductance				3	
Contact capacitance				2	
Current rating					2
Insertion force		2			
Removal force		8			
Durability	3	4(a)			
Physical shock		6			
Vibration		5			
Humidity				7	
Temperature life			3		
Thermal shock				6	

(a) 500 insertion and removal cycles at the maximum rate of 200 cycles per hour.

For example, to perform the temperature life test (Test Group C), one would perform the following steps:

1. Examination of the connector(s)
2. LLCR
3. Temperature life
4. LLCR
5. Examination of the connector(s)

4.6.9 Measuring Current Carrying Capacity on Connector Power Pins

This section describes a procedure for a current rating test on the contacts of a Device Bay connector power segment. This current rating test is to be applied to three contacts in parallel in the power segment as shown in Figures 4-14. The procedure is to increase current on one or more contacts, while measuring the temperature rise on the contacts, using a thermocouple as shown in Figure 4-15.

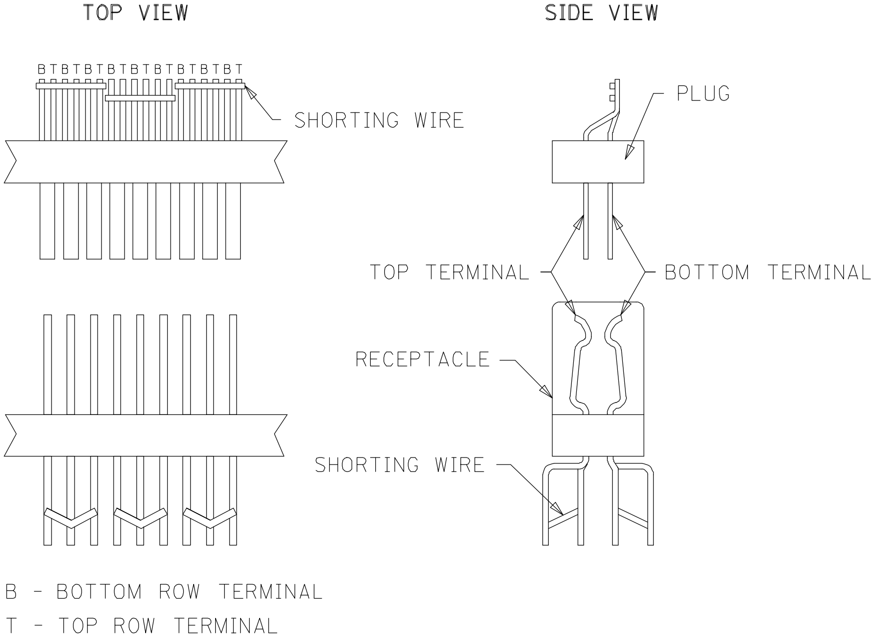

Figure 4-14. Current Capacity Test Setup

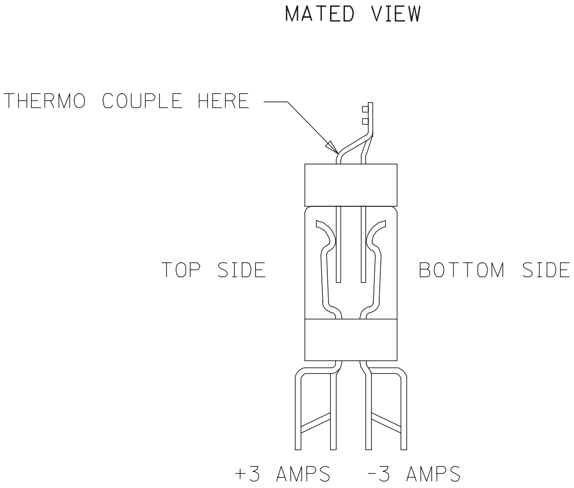

Figure 4-15. Mated View of Current Capacity Test Setup

The current rating of the pins is the maximum applied current that does not raise the temperature of the contact more than 30°C. The ambient condition is still air at 25°C. Note that the current is applied at different levels over time, as shown in Figure 4-16.

The connector must support the 3-pin combined current on the sets of pins as shown in Table 4-12, at the levels and times specified above in Figure 4-16 and Table 4-11.

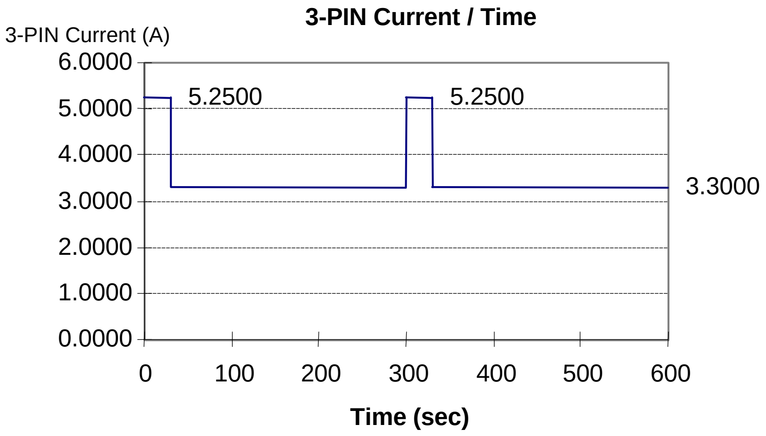

Figure 4-16. Time Pattern of Current Application to Contacts

Table 4-11. Table of 3-Pin Current Carrying Contact Capabilities

Time	3-PIN Current	AMP-Seconds
0	5.2500	
30	5.2500	157.5000
30	3.3000	0.0000
300	3.3000	891.0000
300	5.2500	0.0000
330	5.2500	157.5000
330	3.3000	0.0000
600	3.3000	891.0000
RMS Current		3.4950

Table 4-12. Pin Designations for Voltages

Voltage (V)	Current (high / dwell) (A)	Outgoing Pins	Return Pins
3.3	5.25 / 3.30	B1, B2, B3	B10, B11, B12
3.3	5.25 / 3.30	B4, B5, B6	B13, B14, B15
5.0	5.25 / 3.30	B16, B17, B18	B13, B14, B15
12	5.25 / 3.30	B7, B8, B9	B13, B14, B15

5 Mechanical Requirements

This section defines the mechanical requirements for the form factors.

5.1 Device Bay Mechanical Features—Overview

This section introduces the form factors and gives examples of their use.

5.1.1 Dimensions and Features of the Different Form Factors

This section defines three (3) device form factors. They are as follows:

- DB32 - 32.00 X 146.00 X 178.00mm [1.260” X 5.748” X 7.008”]
- DB20 - 20.00 X 130.00 X 141.50mm [.787” X 5.118” X 5.571”]
- DB13 - 13.00 X 130.00 X 141.50mm [.512” X 5.118” X 5.571”]

All Device Bay bays have the following features:

- Device Bay connector receptacle
- Eject scheme
- Retention mechanism
- EMI/ESD ground contacts
- Software-controlled interlock

All Device Bay devices have the following features.

- Device Bay connector plug
- Ejector landing area
- Retention features
- EMI/ESD ground pad

Figure 5-1 shows an example of these features using an DB32 device; Figure 5-2 shows an example of these features for a DB20 device. The universally located connector plug area and ejector landing area on the device are called out in both drawings.

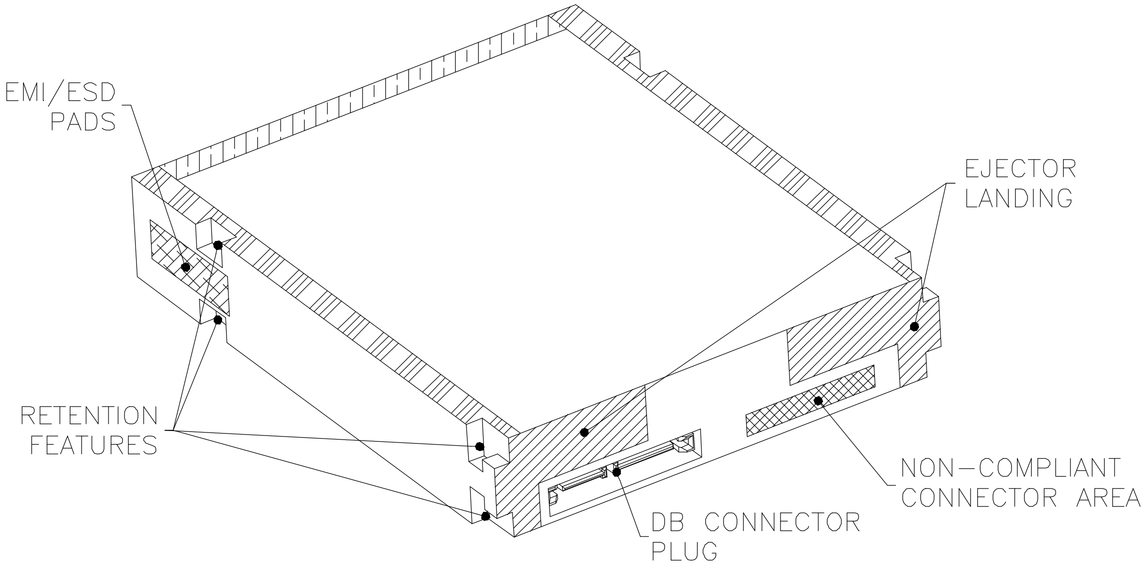

Figure 5-1. Example of Device Bay device features using an DB32 device

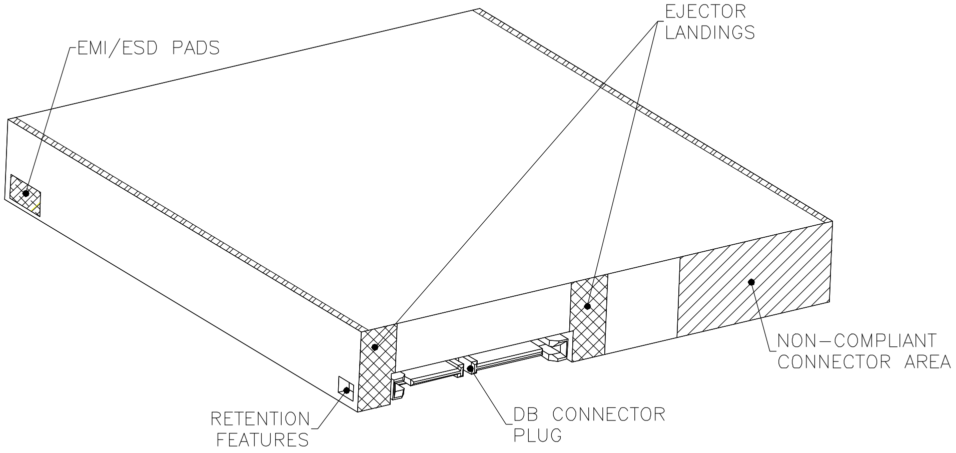

Figure 5-2. Example of Device Bay device features using an DB20 device.

5.2 Device Bay Mechanical Features

This section contains the detailed requirements for the form factors.

5.2.1 Feature Requirements

This section specifies the following feature requirements:

- Device Bay connector location
- Retention features
- Ejector landing area
- EMI/ESD grounding pad
- Grip and clamp zones
- Non-compliant power connector area
- Reserved areas
- Maximum device weight.

5.2.1.1 Device Bay Connector Location

All devices must have a Device Bay connector plug (for more information, see section 4).

The connector plug must be recessed from the back surface of the device.

- For DB32 devices see Figure 5-8, section view B-B (or Figure 5-11 for the enlarged view).
- For DB20 devices, see Figure 5-16, section view B-B (or Figure 5-19 for the enlarged view).
- For DB13 devices, see Figure 5-23, section view B-B (or Figure 5-26 for the enlarged view).

5.2.1.2 Maximum Device Weight

A device, for each of the form factors, must not exceed the maximum weight shown in the following table.

Table 5-1. Maximum device weights for different form factors.

Form Factor	Weight (kg [lbs.])
DB32	1.40 [3.08]
DB20	0.50 [1.10]
DB13	0.35 [0.77]

5.2.1.3 Retention Features

All devices must provide all of the retention features shown to enable the system OEMs to design mechanisms to retain the device in the bay after insertion. These areas may be exposed to significant shock and vibration events; therefore these features need to comprehend the possible loads on these surfaces.

A device's retention features must *individually* be able to support a retention force exerted by the system.

For DB32, the retention features must *individually* support a minimum of 35 lbs. ***or*** 40 g's, *whichever is more* (calculated using the device's weight), and the minimum contact area described in the table below.

For DB13 and DB20, the retention features must *individually* support a minimum of 20 lbs. and the minimum contact area described in the table below.

Table 5-2. Minimum Contact Area

Form Factor	**Min. Contact Area (mm^2 [in^2])**
DB32	12.00 [.019]
DB20	3.00 [.0047]
DB13	3.00 [.0047]

The possible load surfaces for the retention features are shown in the following figures for each form factor:

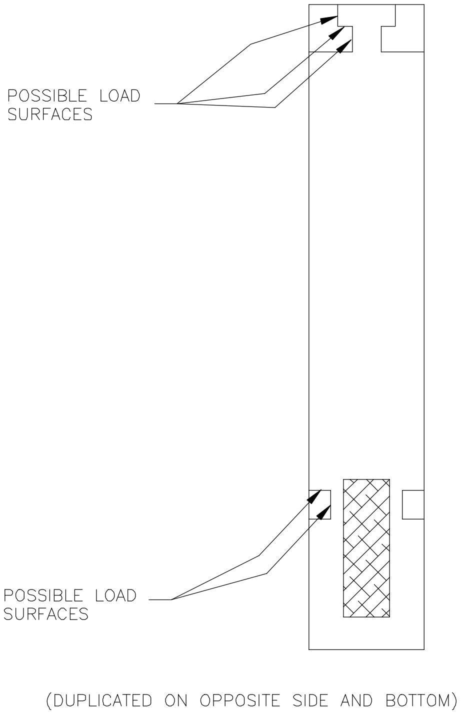

Figure 5-3. Expected load surfaces for DB32 device

.

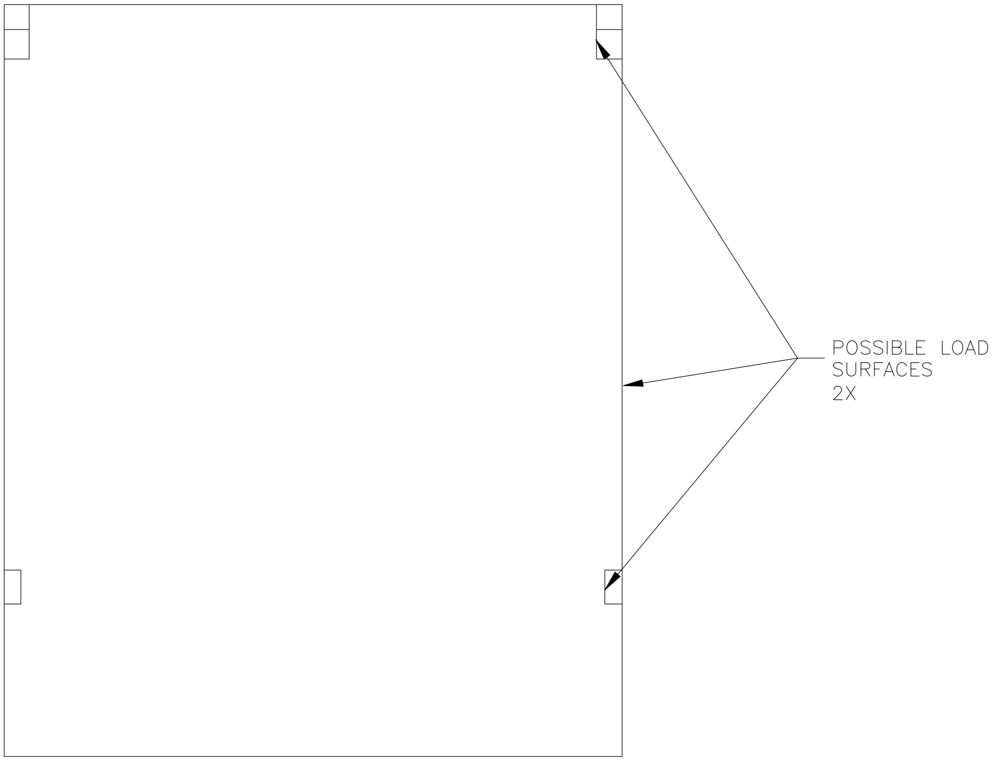

Figure 5-4. Expected load surfaces for DB32 device

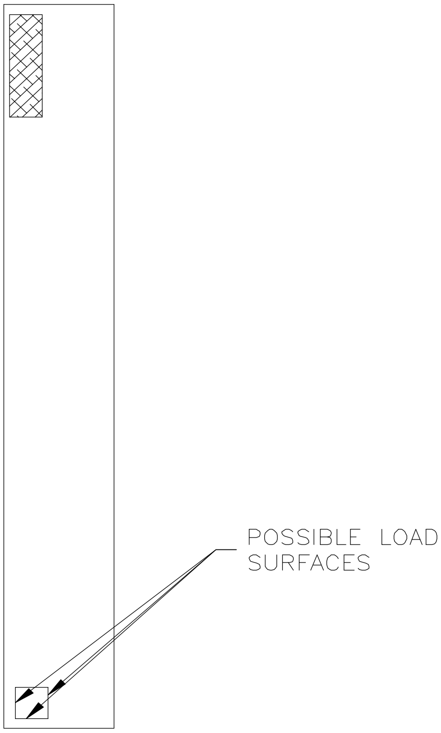

Figure 5-5. Expected load surfaces for DB20/DB13 devices

5.2.1.4 Ejector Landing Area

All devices must provide the defined areas for the bay to use for ejecting the device. This area must be capable of taking the ejection force (point load) as specified in the section 5.3.2.1, Table 5-4.

5.2.1.5 EMI/ESD Ground Pad

All devices must provide the specified EMI/ESD grounding pads unless the body of the device is metal and satisfies the conditions set forth in Section 5.2.1.5.1.

Note: These surfaces are required. No holes or gaps in surfaces are allowed.

5.2.1.5.1 Ground Pad Material Compatibility

The material used as a ground pad must provide the same characteristics as described in section 5.3.2.5.1.

Note: This pad could see significant contact force from the bay for contact resistance and possibly device stabilization.

5.2.1.5.2 ESD/EMI Ground Requirements

The connection of the ESD/EMI pads by the device to the Device Bay Connector is optional, but is allowed as required by the device. If there is a connection, it must be connected to ground pins A1, A4, and A7; the resistance between the device ground pad and the connector ground will depend on device EMI requirements.

5.2.1.6 Grip and Clamp Zones

In order for the device to be protected and removed from the bay, the device must provide the specified grip and clamp zones. DB32 has both clamp and grip zones, whereas DB13 and DB20 have only clamp zones.

- The grip zone provides the necessary structure for the user to grasp the device while removing it from the bay, typically after ejection of the device.
- The clamp zone enables the system OEMs to provide features in the bay to hold the device in place after insertion. This can be done with spring fingers, springs, and so on. The clamp zone helps protect the device during vibration and shock events, since the device is not rigidly mounted in the system.
- The keepout area is a no-load area. This prevents potential damage to the spindles and motors of such devices as a HDD or an optical drive. No forces or pressures can be applied on this area.

The minimum contact area for the clamp zone is shown in Table 5-2.

It is desirable for all form factors to keep pressure loads and forces off of the center of the devices, see Figure 5-6. The keepout area on a DB32 device is bounded by clamp and grip zones. The DB13 and DB20 keepout areas are only bounded by the clamp zones, see Figure 5-7.

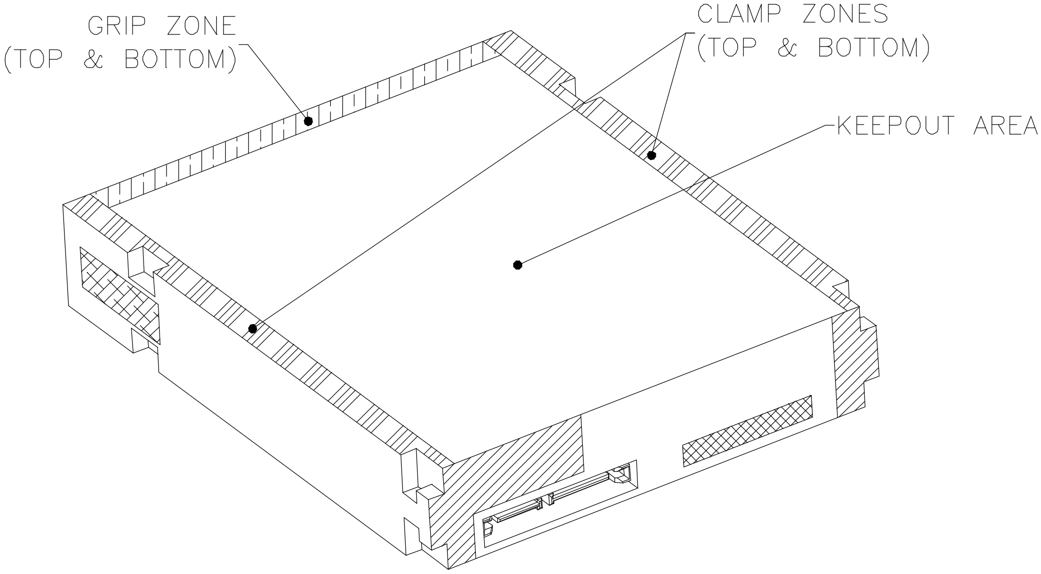

Figure 5-6. DB32 clamp zones, grip zone, and keepout area

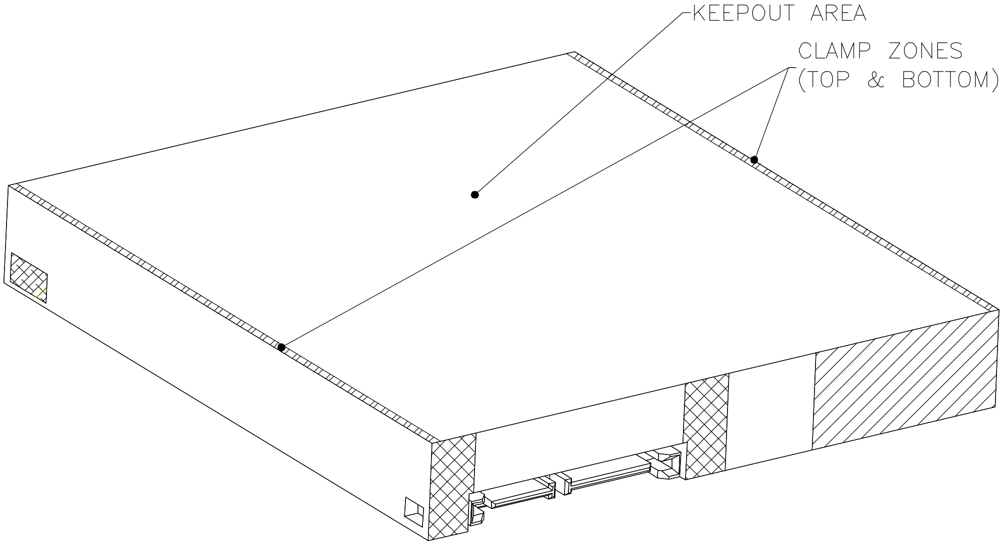

Figure 5-7. DB20 and DB13 clamp zones and keepout area

5.2.1.7 Non-compliant Power Connector Area

A non-compliant power connector area is specified to provide for the needs as defined in section 2.3 of this specification. A device that uses this area is, by definition, incompatible with the Device Bay specification. The non-compliant power connector area requirements are:

- The device and its non-compliant power connector must avoid damaging the Device Bay connector in the bay and the device (if present).
- The connector, both the plug and the receptacle, must be shrouded to prevent exposed pins to the outside environment, so that any non-grounded or current-carrying conductive surfaces cannot extend outside the non-conductive shroud.

When a device uses a non-compliant power connector and does not use a Device Bay connector, a recess in the non-compliant device must be made so that there is no damage or interference with the Device Bay connector receptacle in the bay.

5.2.1.8 Reserved Area

All areas on the back of the Device not defined by the above areas are reserved for the future use of the Device Bay specification. Devices must not use any area outside of the defined areas.

5.2.1.9 Maximum Reaction Force

Motion of components inside the device causes reaction forces applied by the device to the bay. When the device is installed in the bay, the maximum reaction force and the reaction torque applied to the bay must be less than Table 5-3.

Reaction forces due to imbalance of rotating components are included in these limits. Bay designers should account for the periodic nature of imbalance forces.

Table 5-3. Reaction Force and Torque

Form Factor	Reaction Force (kg [lbs.])	Reaction Torque (Nm [in-lbs])
DB32	0.10 [0.220]	0.5 [4.40]
DB20	0.02 [0.044]	0.05 [0.44]
DB13	0.02 [0.044]	0.05 [0.44]

5.2.1.10 Acoustic Test Specification

All acoustic measurements of a device outside the system must be made in compliance with ISO 7779. Operating conditions for the device during measurements are given in ECMA 074, Dec. 1997 edition, which is available at www.ecma.ch.

5.2.1.11 Exposed PCAs and components

Devices may have exposed components as long as the bezel is rated 94V-1; when they are installed in the bay, they are considered to be within a fire enclosure.

5.2.1.12 Adapters

In order to allow for smaller form factor devices to be installed into larger form factor bays, adapters will be needed. These adapters need to provide the corresponding device characteristics to the system and the proper bay characteristics to the smaller device (that is, an adapter must be able to support the smaller device as if it is in its respective bay).

5.2.1.13 Air Vents

Air vents may be placed on devices as required provided that one (1) dimension of each opening is 1.50mm [.059 in] maximum.

5.2.2 Mechanical Dimensions

This section contains a set of drawings that specify the dimensions of devices of different form factors, along with the dimensions of the required device features.

5.2.2.1 DB32 Device Dimensions

Figures 5-8 through 5-15 show the details for all required features on a DB32 device. Figure 5-8 shows the top, rear, side, and front views of the DB32 device. Figures 5-9 through 5-15 show close-up views of each angle so that it is easier to see the dimensions and placement of the features. For all figures:

- All tolerances not explicitly shown in the drawing are +/- 0.50mm [.020”].
- All draft is to be within dimensional tolerances.
- All round dimensions are described in section 5.2.2.5.1.
- A cutout is required around the volume of the connector for relief to allow the connector to blind-mate, as shown in Figures 5-10 and 5-11.

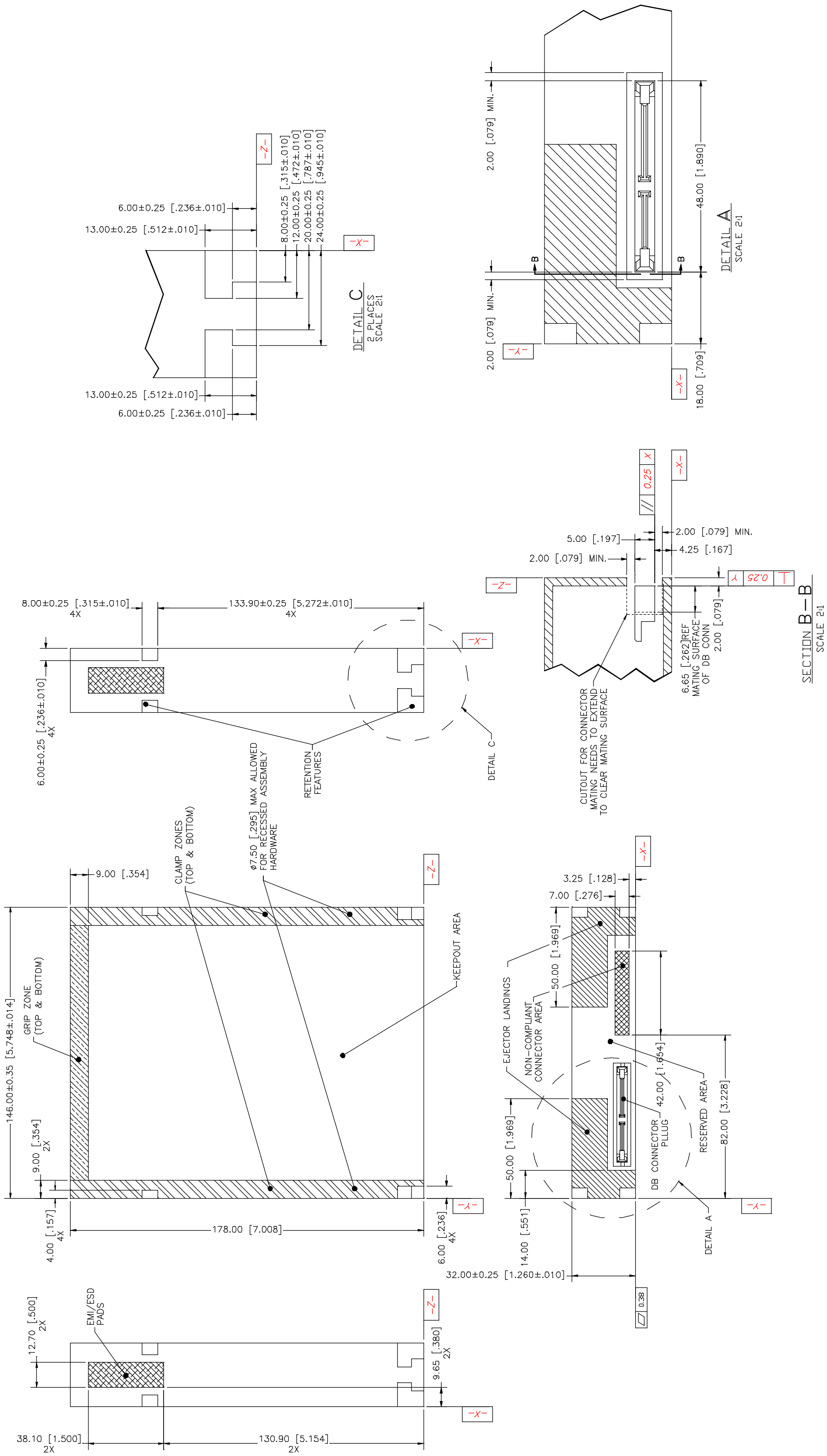
DETAIL C
2 PLACES
SCALE 2:1
6.00±0.25 [.236±.010]
13.00±0.25 [.512±.010]
8.00±0.25 [.315±.010]
12.00±0.25 [.472±.010]
20.00±0.25 [.787±.010]
24.00±0.25 [.945±.010]
-Z-
-X-
-Y-
DETAIL A
SCALE 2:1
2.00 [.079] MIN.
48.00 [1.890]
18.00 [.709]
B
SECTION B-B
SCALE 2:1
5.00 [.197]
4.25 [.167]
0.25 Y
0.25 X
6.65 [.262]REF MATING SURFACE OF DB CONN
2.00 [.079]
CUTOUT FOR CONNECTOR MATING NEEDS TO EXTEND TO CLEAR MATING SURFACE
8.00±0.25 [.315±.010] 4X
133.90±0.25 [5.272±.010] 4X
6.00±0.25 [.236±.010] 4X
RETENTION FEATURES
DETAIL C
CLAMP ZONES (TOP & BOTTOM)
ø7.50 [.295] MAX ALLOWED FOR RECESSED ASSEMBLY HARDWARE
KEEPOUT AREA
9.00 [.354]
GRIP ZONE (TOP & BOTTOM)
146.00±0.35 [5.748±.014]
9.00 [.354] 2X
4.00 [.157] 4X
178.00 [7.008]
6.00 [.236] 4X
3.25 [.128]
7.00 [.276]
EJECTOR LANDINGS
NON-COMPLIANT CONNECTOR AREA
50.00 [1.969]
42.00 [1.654]
DB CONNECTOR PLUG
RESERVED AREA
82.00 [3.228]
14.00 [.551]
32.00±0.25 [1.260±.010]
0.38
DETAIL A
EMI/ESD PADS
12.70 [.500] 2X
9.65 [.380] 2X
38.10 [1.500] 2X
130.90 [5.154] 2X

Figure 5-8. DB32 device overview
(For details, see Figures 5-9—5-15)

Figure 5-9 shows a close-up rear view of the DB32 device.

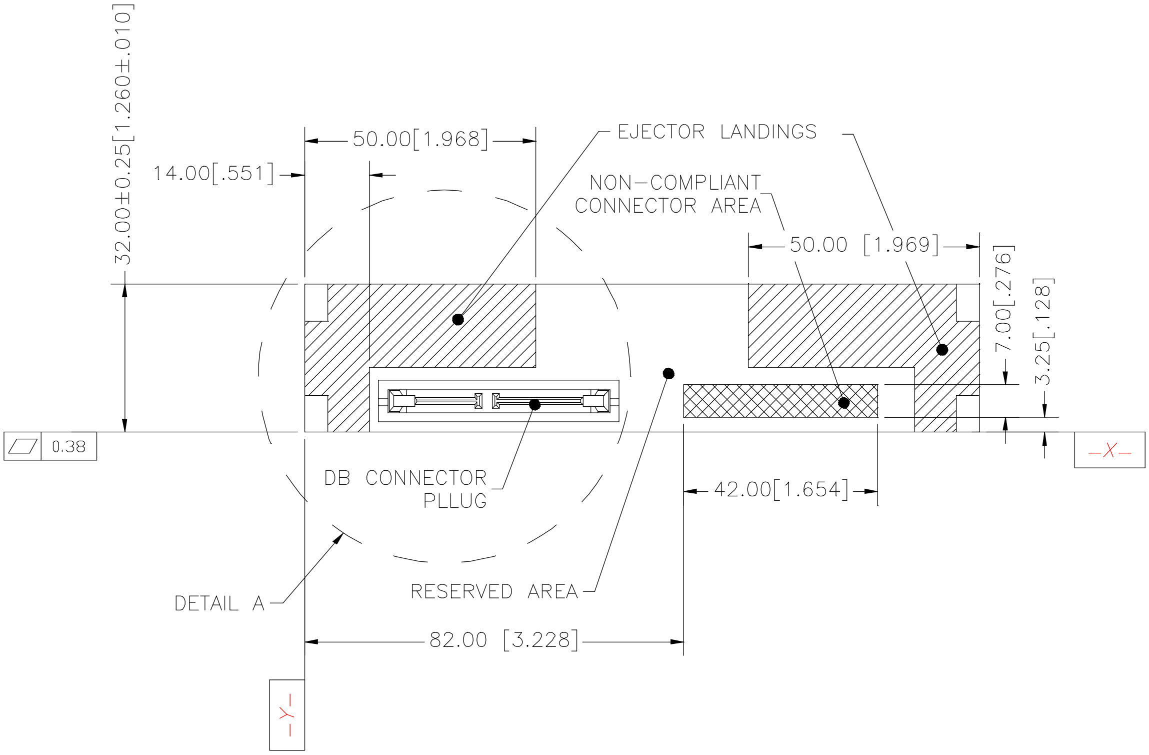

Figure 5-9. Rear view of DB32 device

Figure 5-10 shows an enlarged view of DETAIL A shown in Figure 5-9.

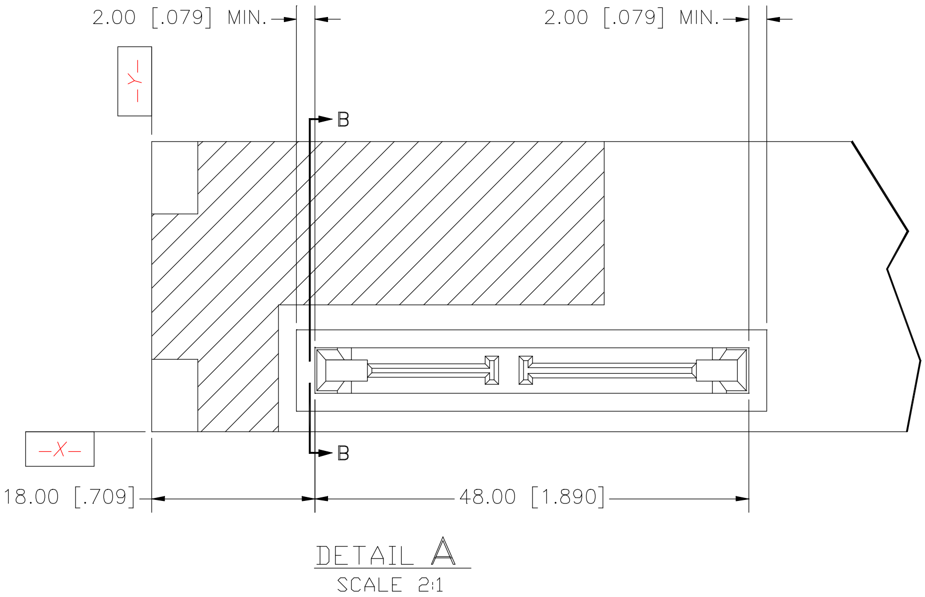

Figure 5-10. Rear view - DETAIL A of DB32 Device

Figure 5-11 shows an enlarged view of SECTION B-B shown in Figure 5-8.

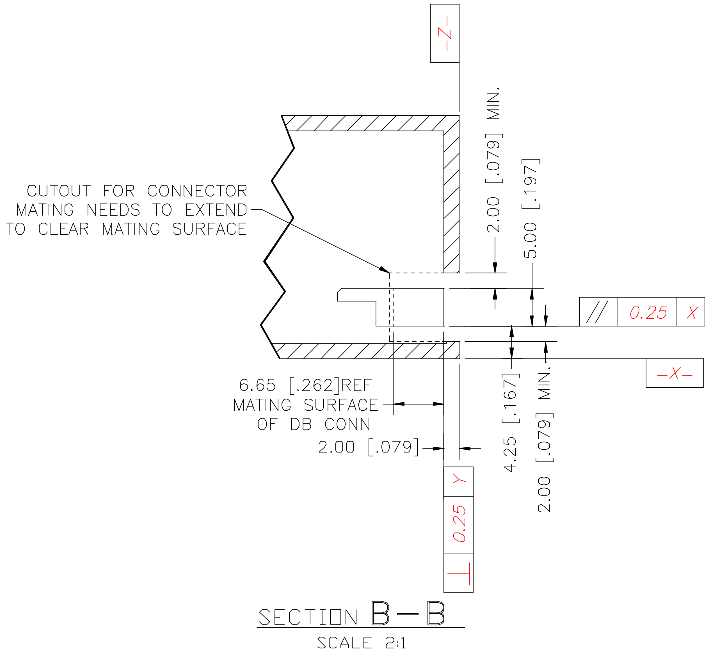

Figure 5-11. Section B - B

Figure 5-12 shows a close-up top view of the DB32 device.

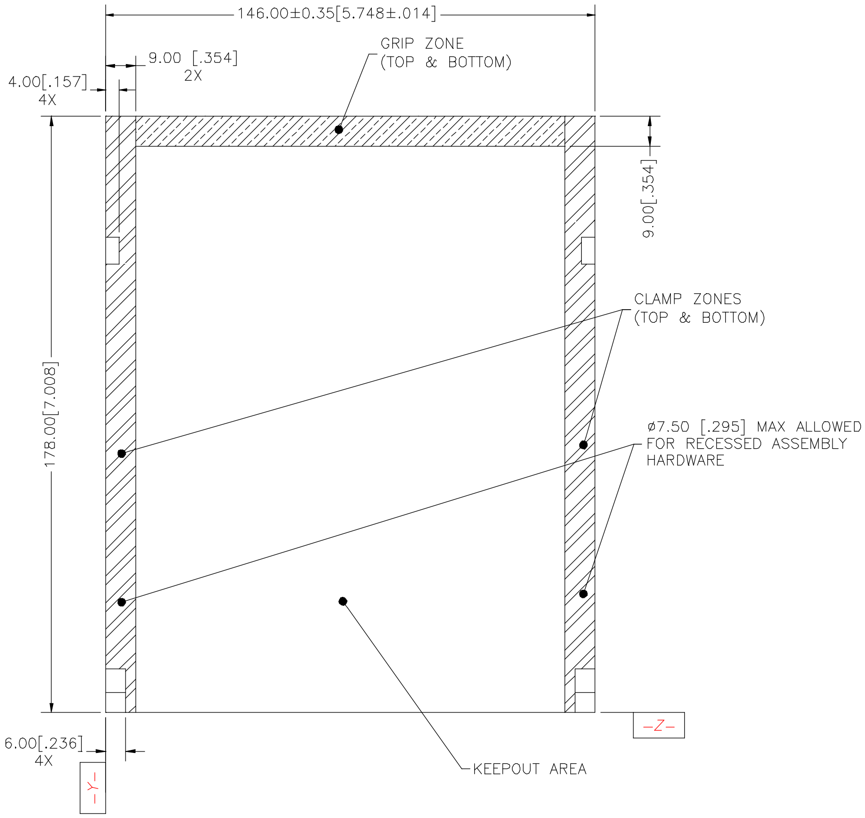

Figure 5-12. Top view of DB32 device

Figure 5-13 shows a close-up view of the right side of the DB32 device.

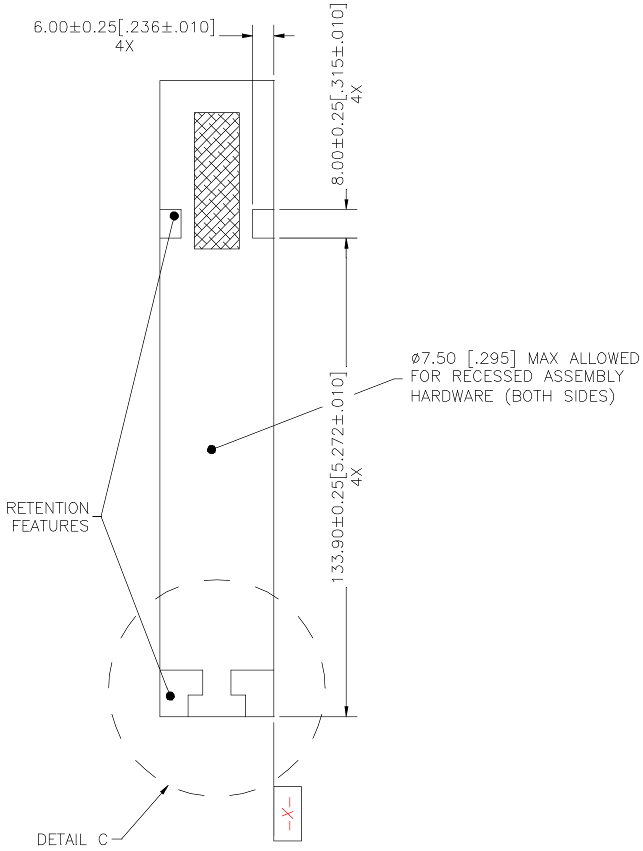

Figure 5-13. Right side view of DB32 device

Figure 5-14 shows DETAIL C from the right-side view of a DB32 device (which is shown in Figure 5-13).

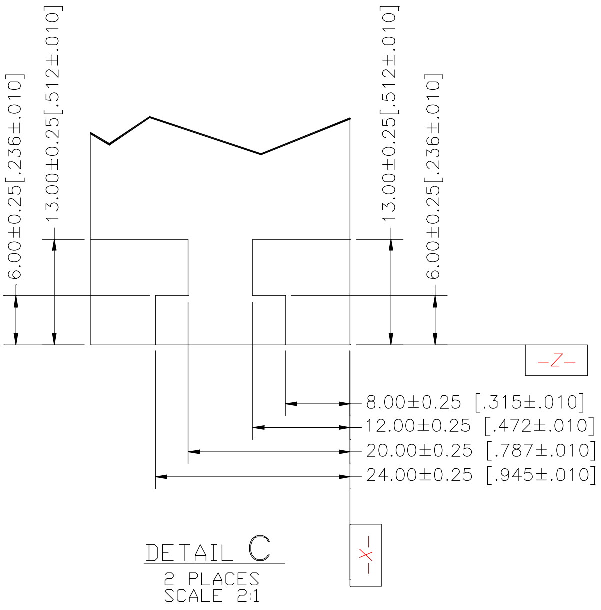

Figure 5-14. Right side –Detail C of DB32 device

Figure 5-15 shows a close-up view of the left side of the DB32 device.

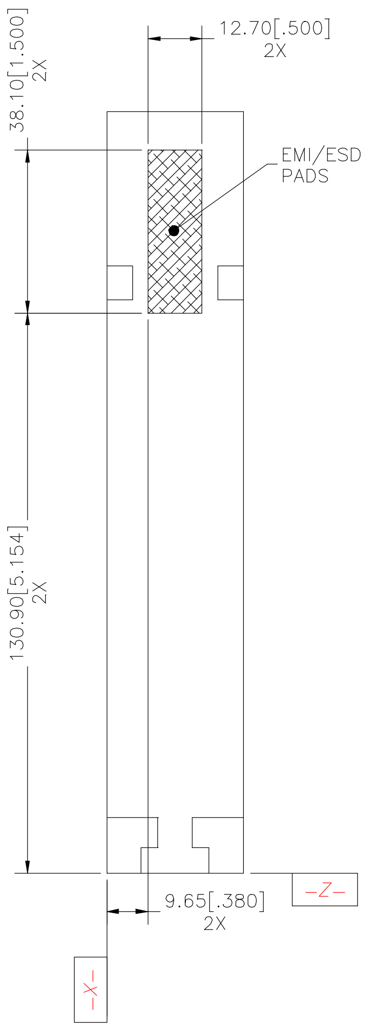

Figure 5-15. Left side view of DB32 device

5.2.2.2 DB20 Device Dimensions

Figures 5-16 through 5-22 show the overall dimensions of the DB20 device. Figure 5-16 shows the top, rear, side, and front views of the DB20 device. Figures 5-17 through 5-22 show close-up views of each angle so it is easier to see the dimensions and placement of the features.

- All tolerances not explicitly shown in the drawing are +/- 0.50mm [.020”].
- All draft is to be within dimensional tolerances.
- All round dimensions are described in section 5.2.2.5.2.
- A cutout is required around the volume of the connector for relief to allow the connector to blind-mate, as shown in Figures 5-18 and 5-19.

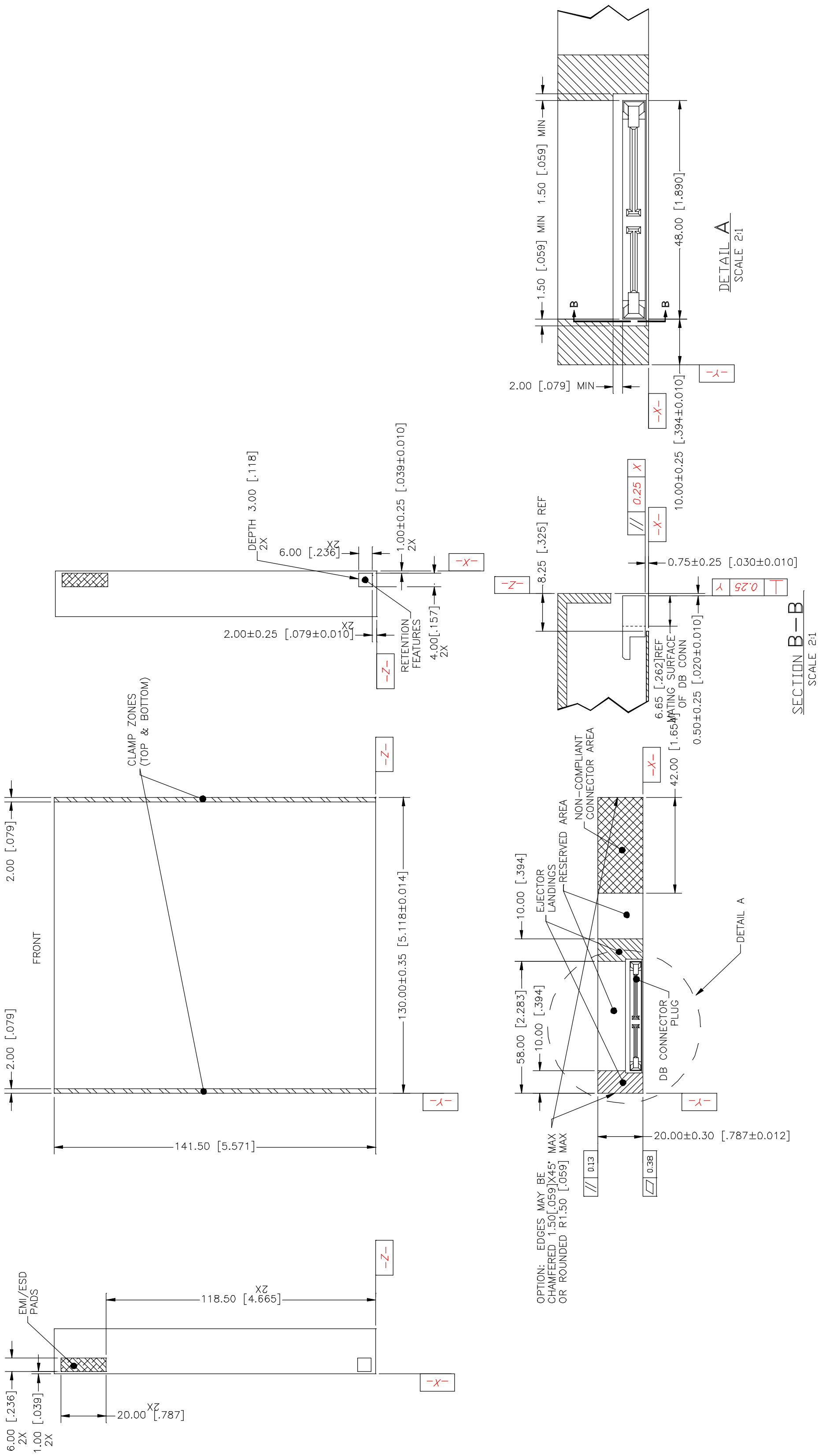

Figure 5-16. DB20 device

Figure 5-17 shows close-up rear view of the DB20 device.

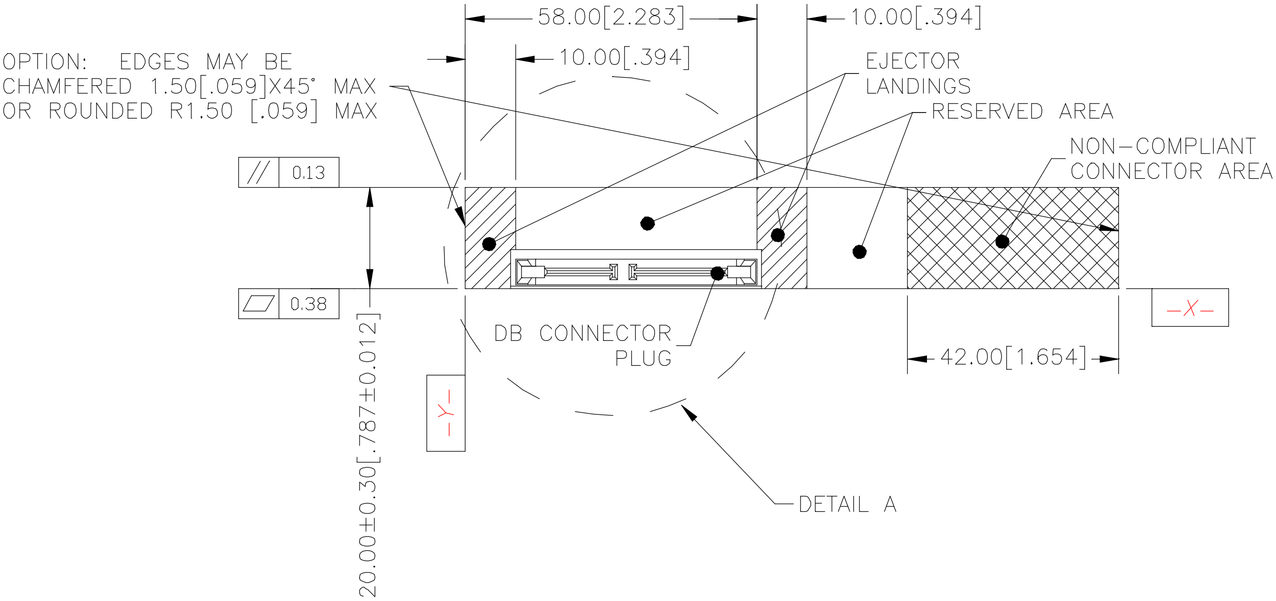

Figure 5-17. Rear view of DB20 device

Figure 5-18 shows a close-up of Detail A shown in Figure 5-16 for the DB20 device.

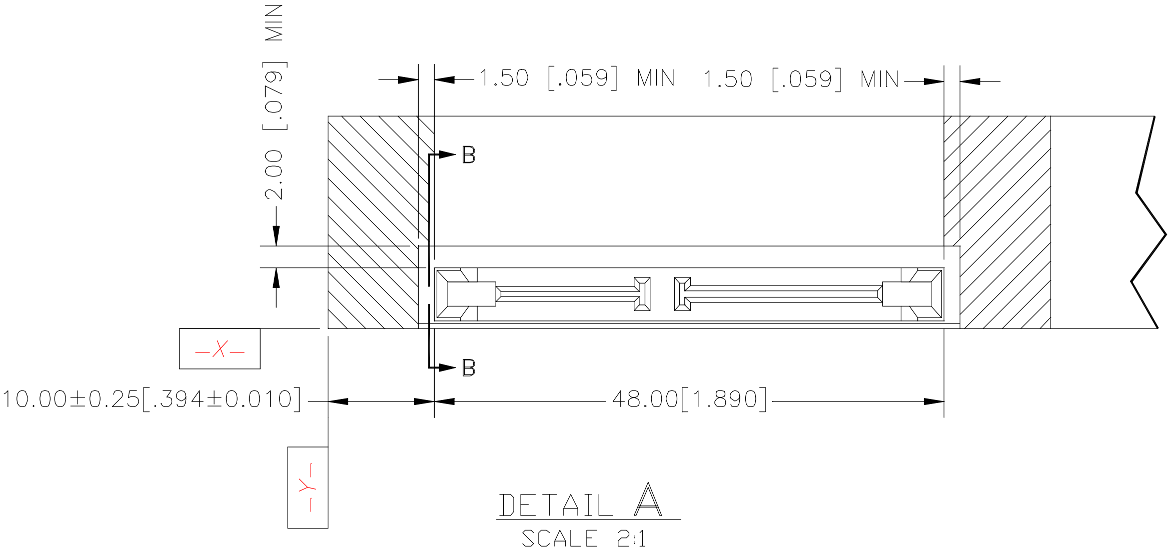

Figure 5-18. DETAIL A for DB20 device

Figure 5-19 shows close-up view of SECTION B-B (from Figure 5-16) for a DB20 device.

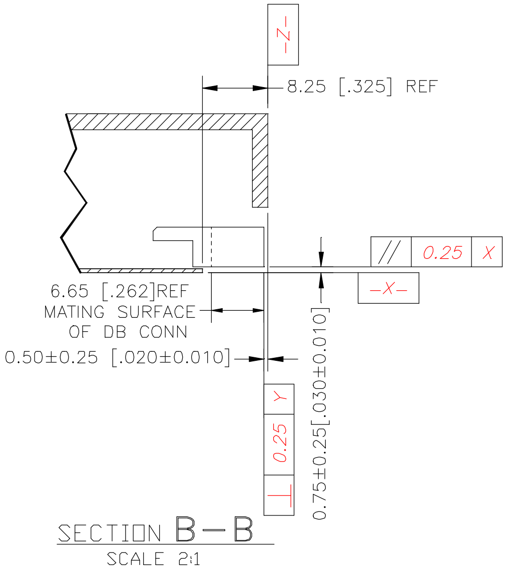

Figure 5-19. SECTION B-B for DB20 device

Figure 5-20 shows a close-up right side view of the DB20 device.

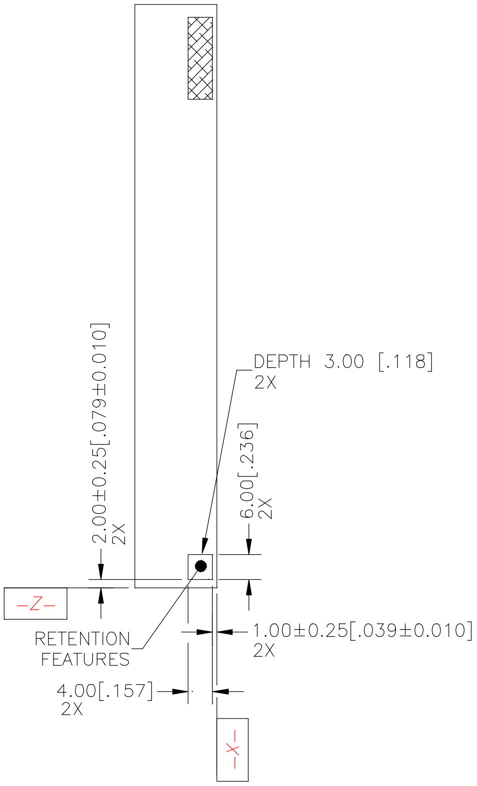

Figure 5-20. Right views of DB20 device

Figure 5-21 shows a close-up view of the left side of a DB20 device.

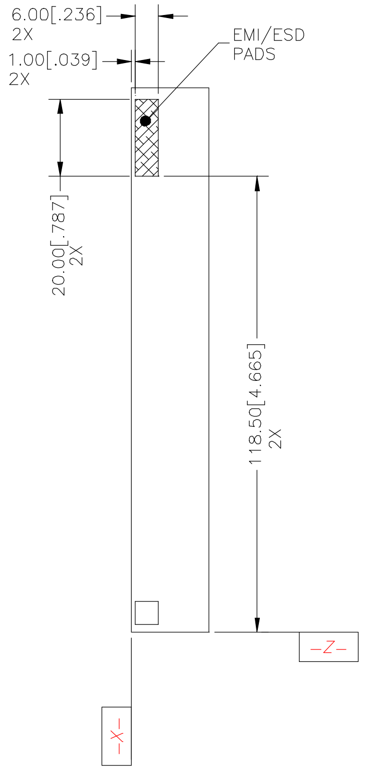

Figure 5-21. Left view of DB20 device

Figure 5-22 shows a close-up top view of the DB20 device.

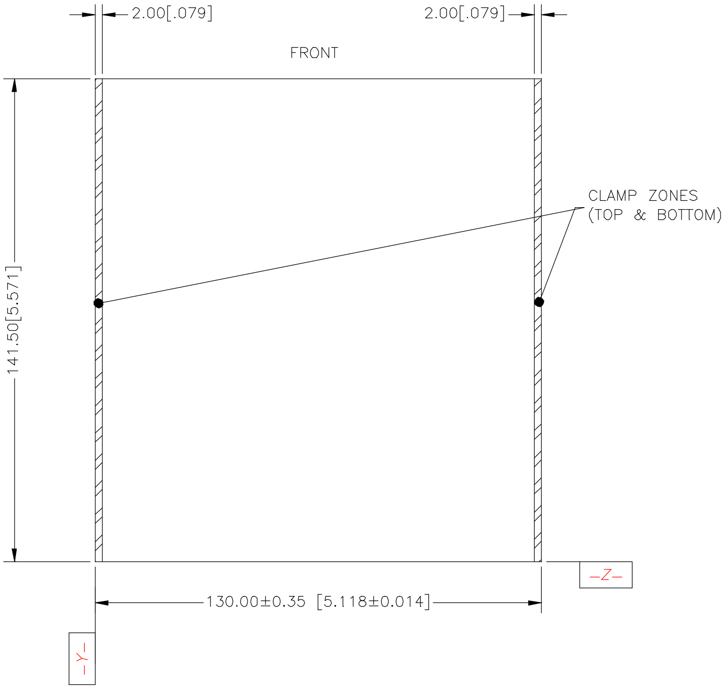

Figure 5-22. Top view of DB20 device

5.2.2.3 DB13 Device Dimensions

Figures 5-23 through 5-30 show the overall dimensions of the DB13 device. Figure 5-23 shows the top, rear, side, and front views of the DB13 device. Figures 5-24 through 5-30 show close-up views of each angle so it is easier to see the dimensions and placement of the features.

- All tolerances not explicitly shown in the drawing are +/- 0.50mm [.020”].
- All draft is to be within dimensional tolerances.
- All round dimensions are described in section 5.2.2.5.2.
- A cutout is required around the volume of the connector for relief to allow the connector to blind-mate, as shown in Figures 5-25 and 5-26.
- A hole and slot are provided on the top surface to provide an attachment method for an adapter so that the DB13 can be used in a DB20 bay, as shown in Figure 5-30.

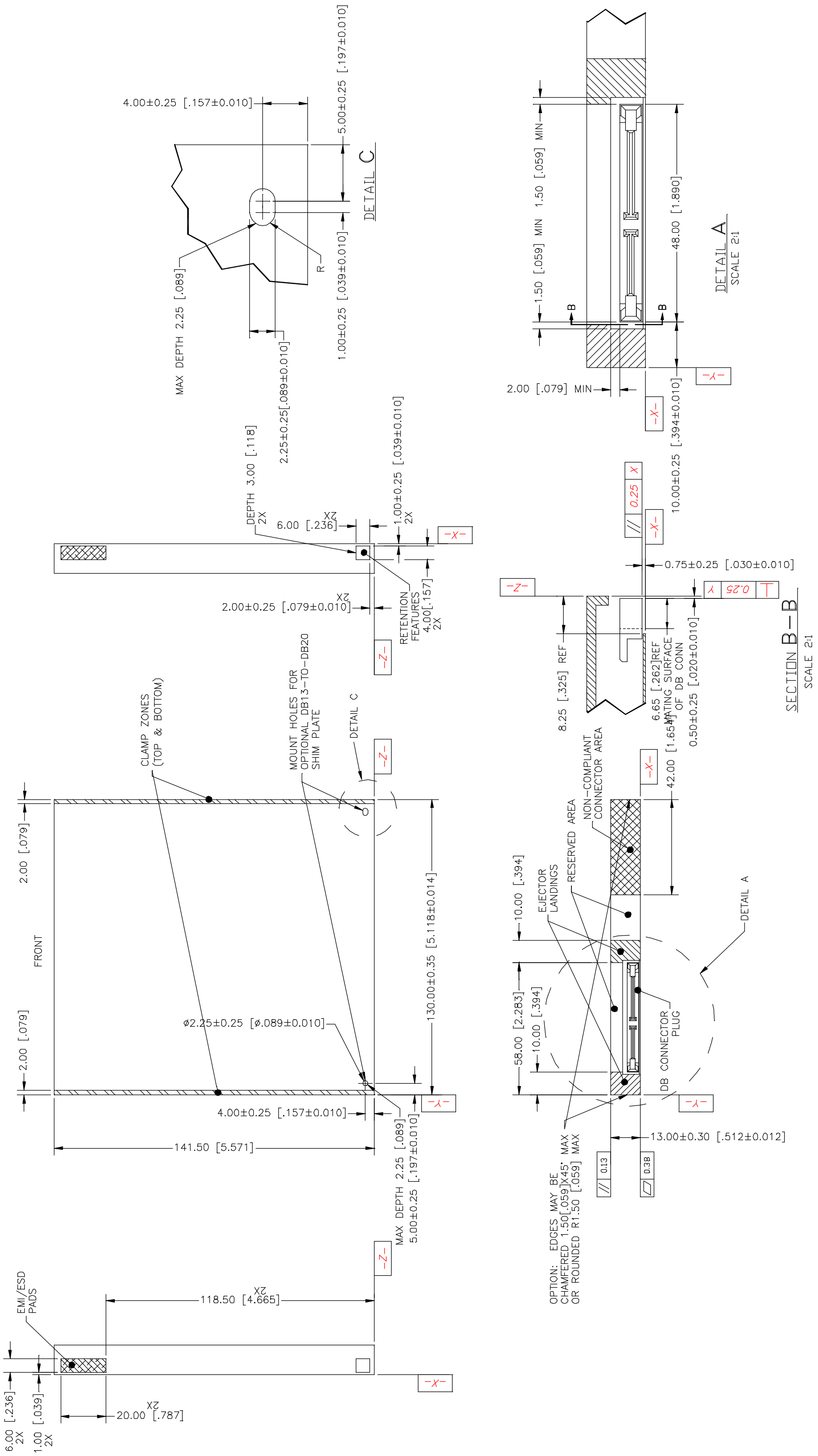

Figure 5-23. DB13 device

Figure 5-24 shows a close-up rear view of the DB13 device.

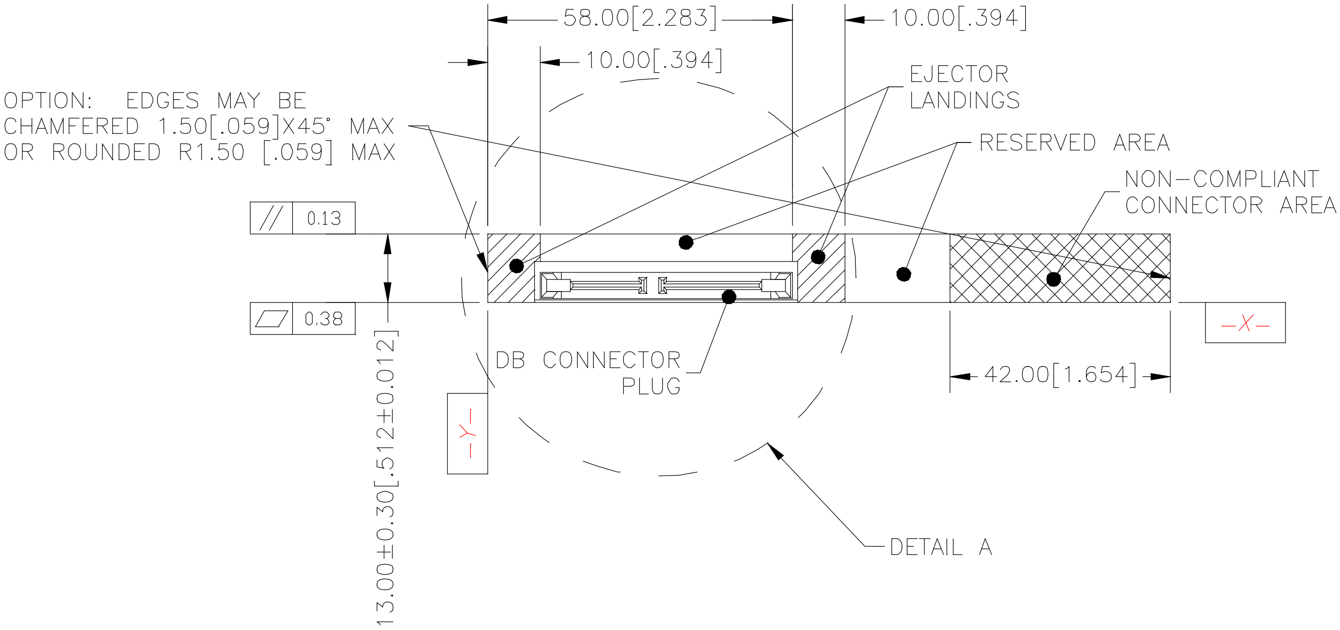

Figure 5-24. Rear view of DB13 device

Figure 5-25 shows a close-up of Detail A, shown in Figure 5-24, for the DB13 device.

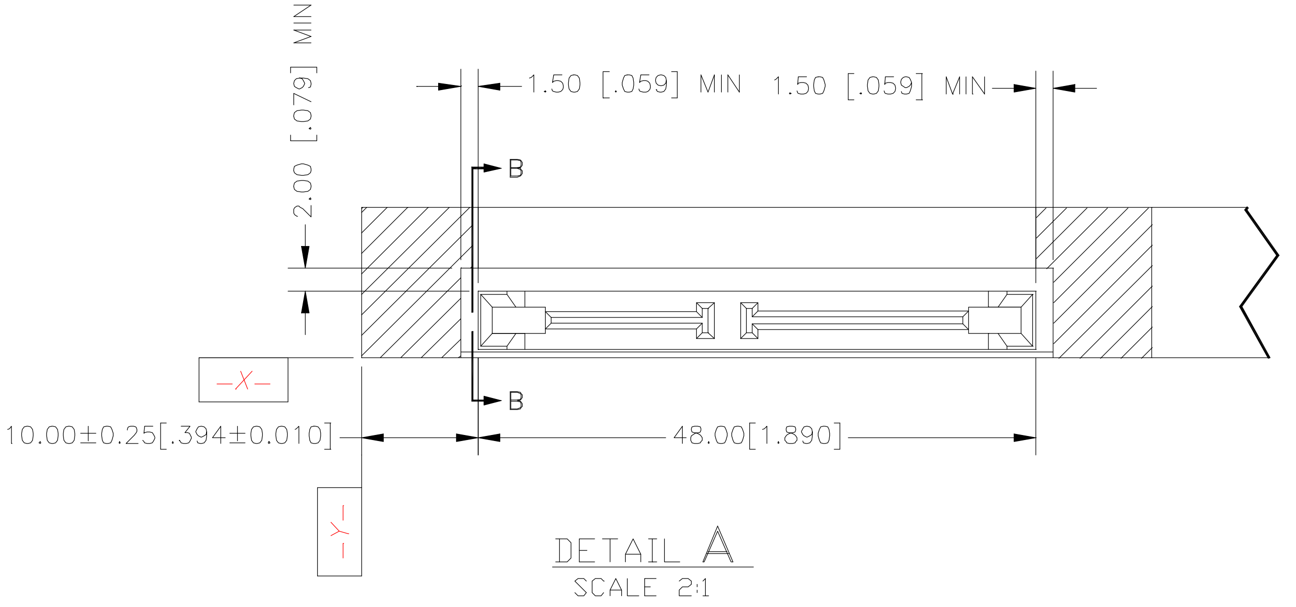

Figure 5-25. DETAIL A for DB13 device

Figure 5-26 shows a close-up view of SECTION B - B shown in Figure 5-23.

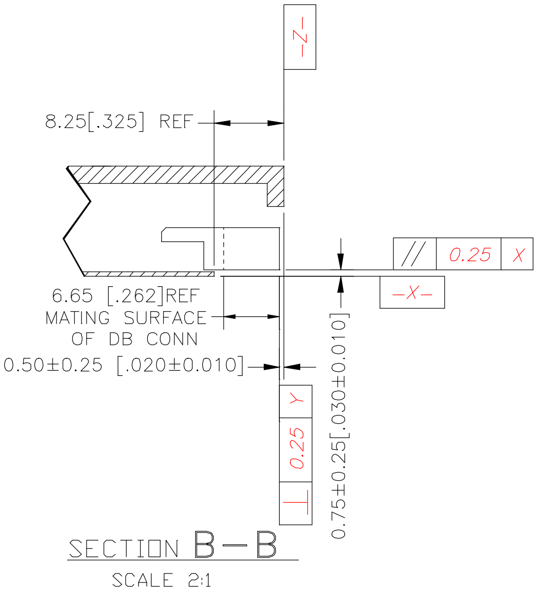

Figure 5-26. SECTION B - B for DB13 device

Figure 5-27 shows a close-up of the left side of the DB13 device.

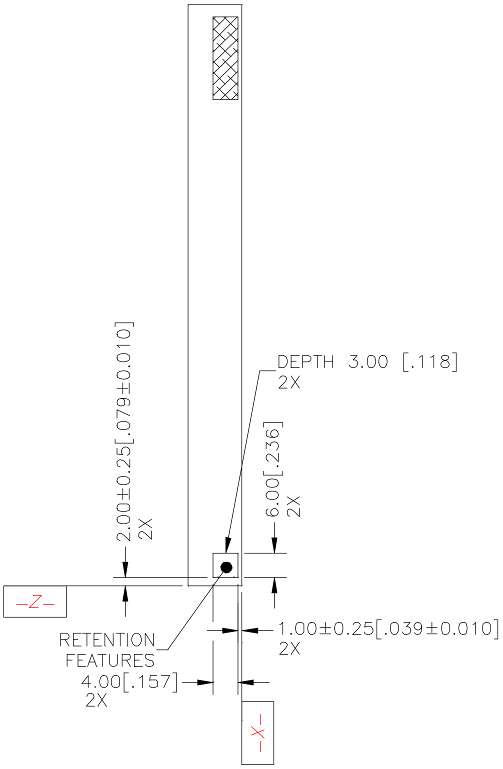

Figure 5-27. Right side view of DB13 device

Figure 5-28 shows a close-up of the left side of the DB13 device.

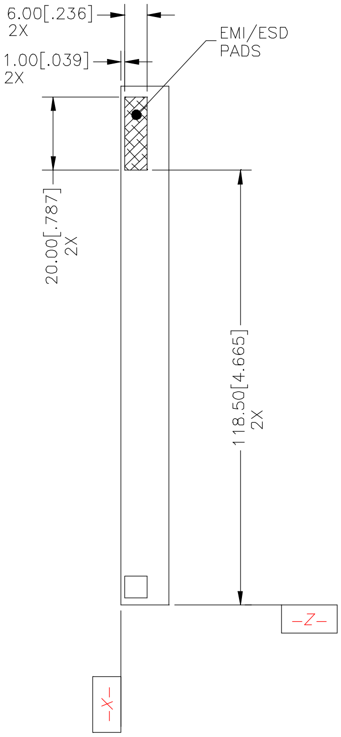

Figure 5-28. Left side view of DB13 device

Figure 5-29 shows a close-up top view of the DB13 device.

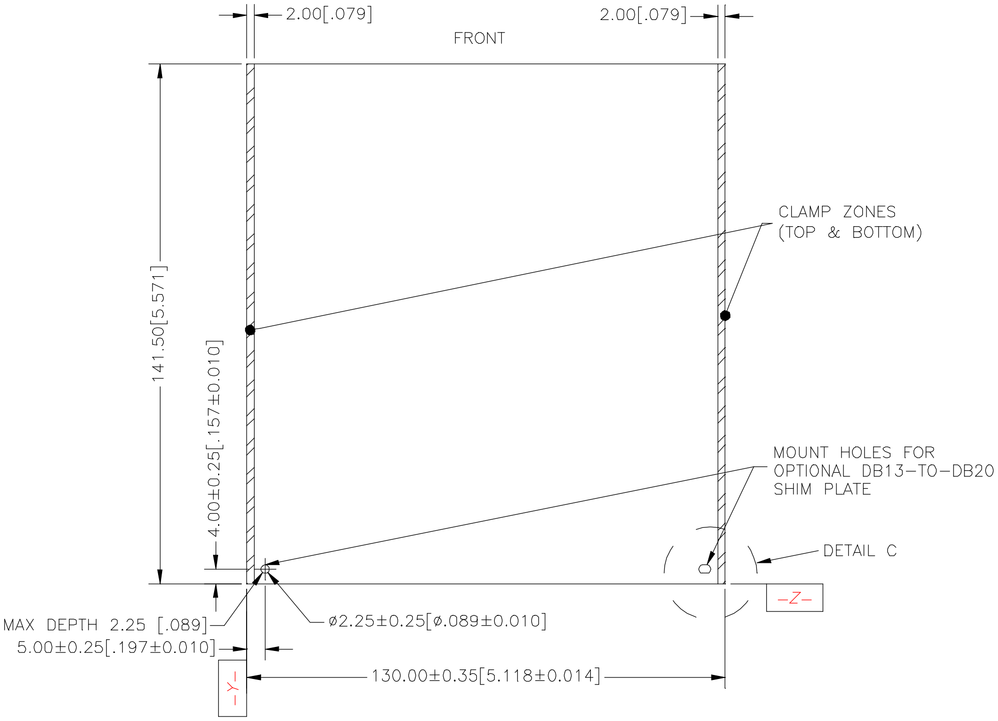

Figure 5-29. Top view of DB13 device

Figure 5-30 shows a close-up view of DETAIL C from Figure 5-29.

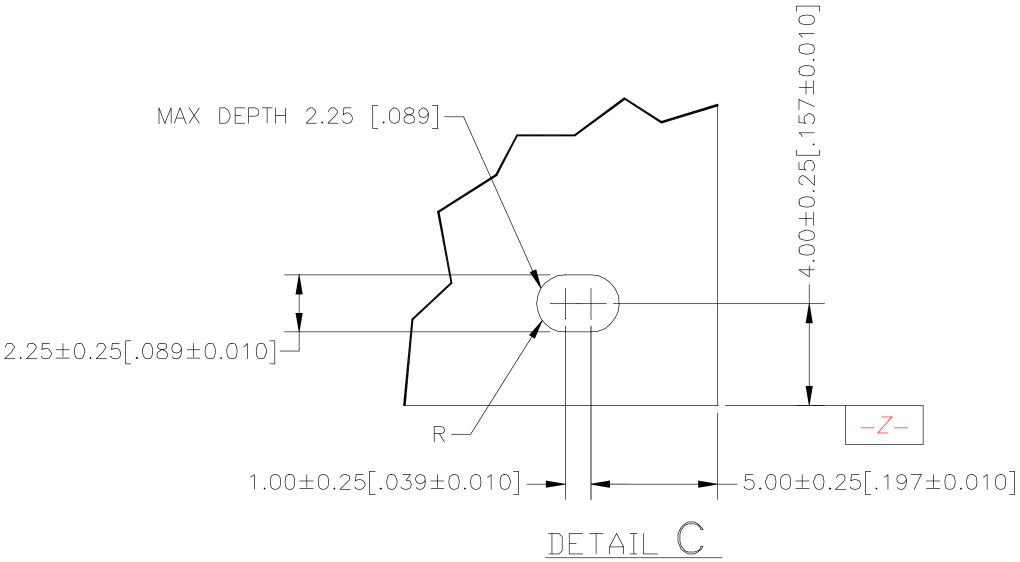

Figure 5-30. Detail C for DB13 Device

5.2.2.4 Rounds

The round dimensions called out in the following figures depend on the edge's use for each form factor.

5.2.2.4.1 DB32

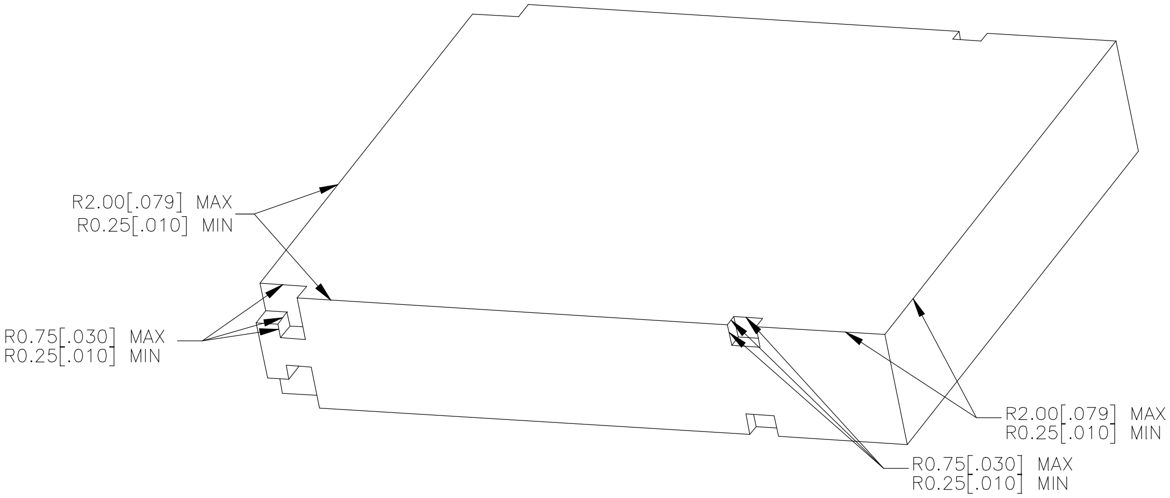

Figure 5-31. Outside Edge Round dimensions for DB32

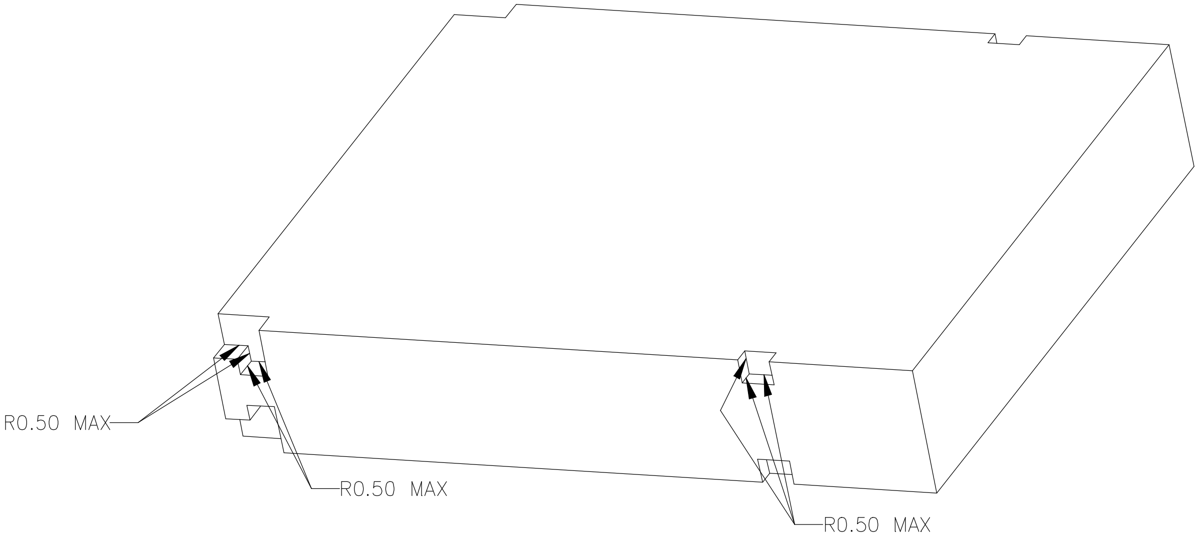

Figure 5-32. Inside Edge Round Dimensions for DB32

5.2.2.4.2 DB20 and DB13

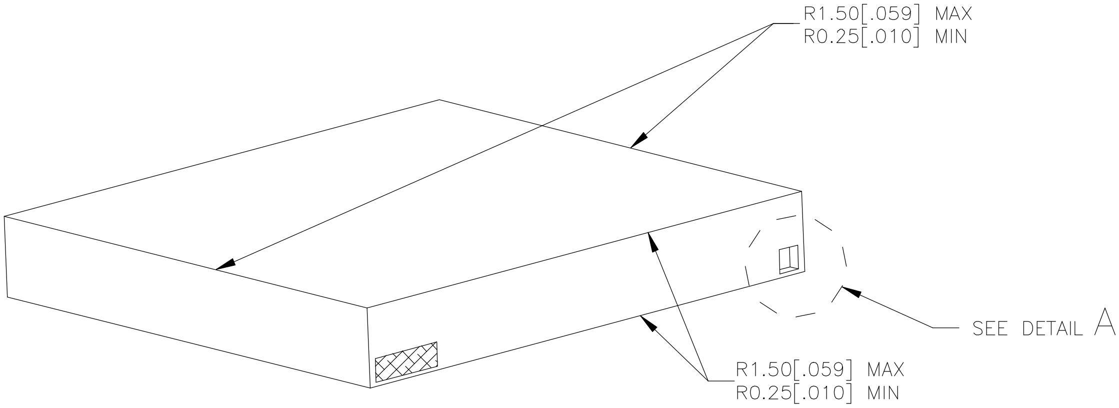

Figure 5-33. Edge Round Dimensions for DB20 and DB13

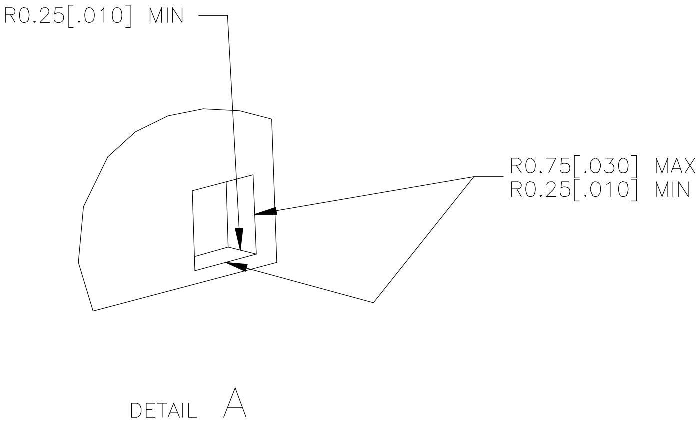

Figure 5-34. Retention Feature Round Dimensions for DB20 and DB13

5.2.2.5 Required Surfaces

The following surfaces are required to ensure interoperability between different manufacturers and designs of devices and Device Bay systems. This is also to allow system designers to know what surfaces are known to be available for usage in bay designs for such purposes as alignment or retention.

5.2.2.5.1 DB32 Required Surfaces

For DB32,

- Grip and Clamp Zones
- Ejector Landing Areas
- Possible Load Surfaces
- EMI/ESD Pads

5.2.2.5.2 DB20 and DB13 Required Surfaces

All surface callouts for DB20 and DB13 devices are required.

5.2.2.6 Other Design Considerations

All screws and other hardware used in the assembly of a device must be flush with or recessed into the volume so there is no potential interference with the bay.

5.2.3 Bezels

This section specifies the front bezels for DB32, DB20, and DB13 devices. Note that the form factor length encompasses the bezel.

5.2.3.1 DB32 Bezel

A DB32 bezel has edges that match the width 146.00mm [5.748"] and height 32.00mm [1.260"] of the device and provide adequate air impedance as described in Section 5.5.

5.2.3.2 DB20 and DB13 Bezels

DB20 and DB13 bezels must have edges that match the width (130.00mm) and the height (20.00 and 13.00mm, respectively). Also the device must have corner to corner lengths that match the length of the device (141.5mm).

The bezel is allowed to protrude past the 141.5mm length to enable notebook vendors to match the industrial design of their systems as long as that dimension does not exceed the maximum protrusion allowed in Section 5.2.3.3.

If a vendor has an ID-specific bezel on their device, this bezel must not interfere with the ejection and retention of devices in the bay as described in their respective sections.

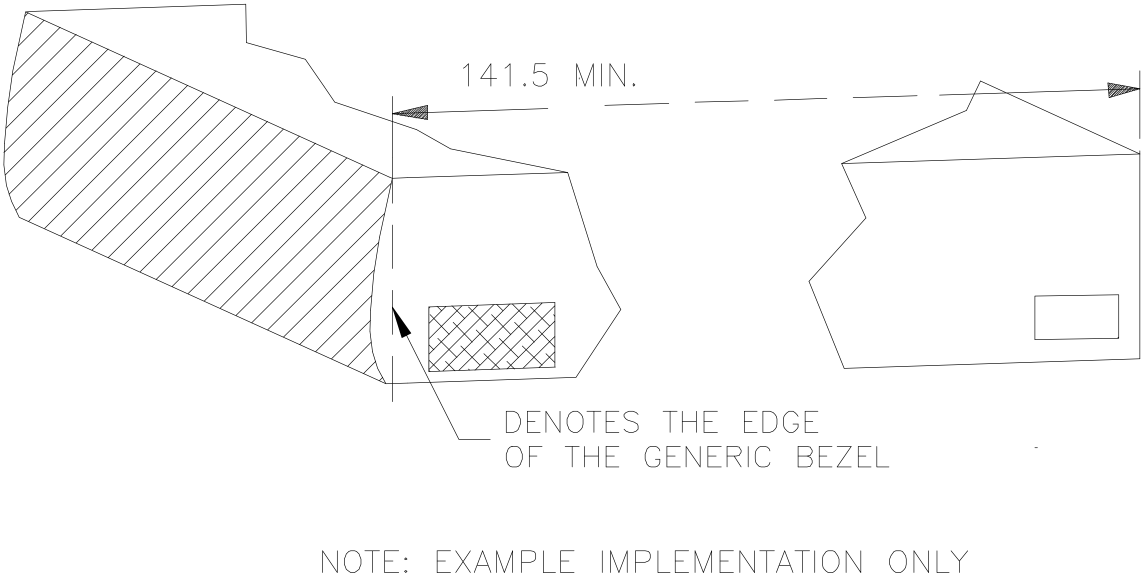

Figure 5-34. ID-specific bezel for DB20 device

5.2.3.3 Maximum Front Protrusion

Some devices, such as floppy and optical drives, require features such as external buttons that extend outside the form factor. At any point in the feature's operation, the maximum overall length is as shown in Figure 5-35 and 5-36. Also, the shape of the bezel may extend past the length of the device, not to exceed the maximum length described in the following illustration.

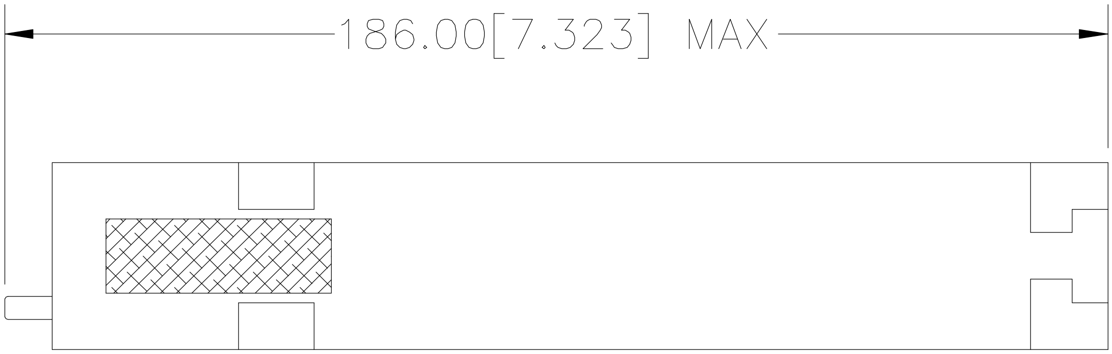

Figure 5-35. Maximum extension beyond the DB32 form factor

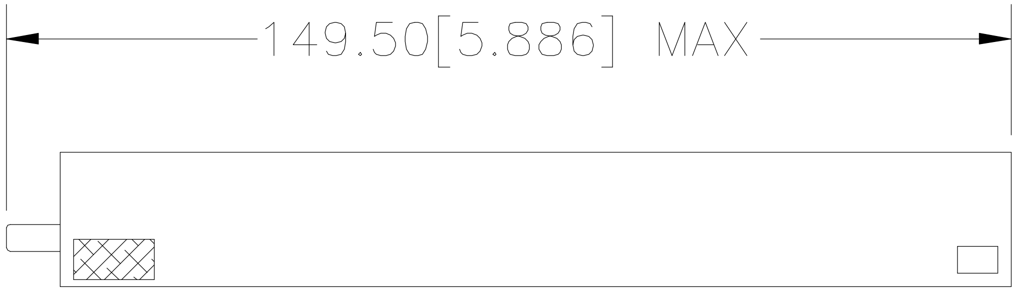

Figure 5-36. Maximum extension beyond the DB20/DB13 form factor

5.2.3.4 Device Bay Logo

The use of the approved Device Bay logotype is restricted to compliant devices. The Device Bay logotype must be of a certain size and located on the top surface of the device within the area shown in the following figures. (Logo dimensions and locations to be shown when decided upon.)

5.3 Mechanical Bay Design Considerations

The Device Bay bay components that are discussed in this section are the *minimum* considerations for designing a bay. The following bay design considerations are discussed in this section:

- Eject scheme, retention features, and software interlocks
- EMI/ESD
- Connector presentation
- Device support feature

5.3.1 Mechanical Insertion and Removal Sequence

Note: To see the sequence of tasks that accomplish device insertion and removal, see the section 7 of this specification; the task sequences in that section include not only the Device Bay mechanical components, but the Device Bay Controller (DBC) and operating system (OS) components as well.

5.3.2 Eject Scheme, Retention Mechanisms, and Software Interlocks

The bay is responsible for providing the following:

- A means for presenting the device for removal from the bay
- A means for retaining the device during normal operation and shipping of the unit
- A software-controlled interlock to discourage surprise removal of the device

All bays must have an eject scheme, retention mechanism, and a software-controlled interlock. These features can be separate mechanisms or incorporated to varying degrees as the same mechanism.

5.3.2.1 Eject Scheme

The bay is required to provide the ability to present the device for removal from the bay. The design used by the bay will be driven by target-market requirements. Some possible provisions include:

- A very low-cost system may provide a recess or cutaway in the system bezel so that the user may grasp the device and remove the device out of the bay without an eject mechanism.
- An eject linkage, similar to the one shown in Figure 5-37, could provide a purely mechanical system.

- An electromechanical eject mechanism could be provided, similar to the portable expansion docking bases seen on the market.

The eject scheme must provide the following features:

- Must be solely a feature of the bay (the device cannot aid in its ejection).
- After the eject scheme has pushed the device off of the connector, the device must protrude beyond the system bezel, as specified in Figure 5-38.
- Sufficient force to unseat the connector, not to exceed the values listed in Table 5-4, for a minimum contact area as specified in Table 5-2.

Table 5-4. Ejection forces for different form factors

Form Factor	Minimum (kg [lbs.])	Maximum (kg [lbs.])
DB32	3.17 [7.0]	6.35 [14.0]
DB20	1.36 [3.0]	2.72 [6.0]
DB13	1.36 [3.0]	2.72 [6.0]

Figure 5-37 shows a device with an example eject mechanism.

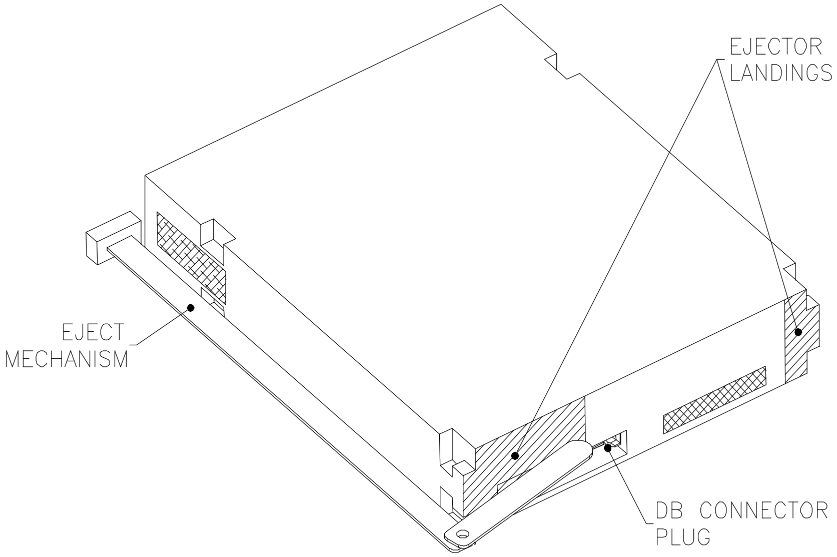

Figure 5-37. Device with an example eject mechanism

Figure 5-38 shows the protrusion beyond the front bezel after an ejection event.

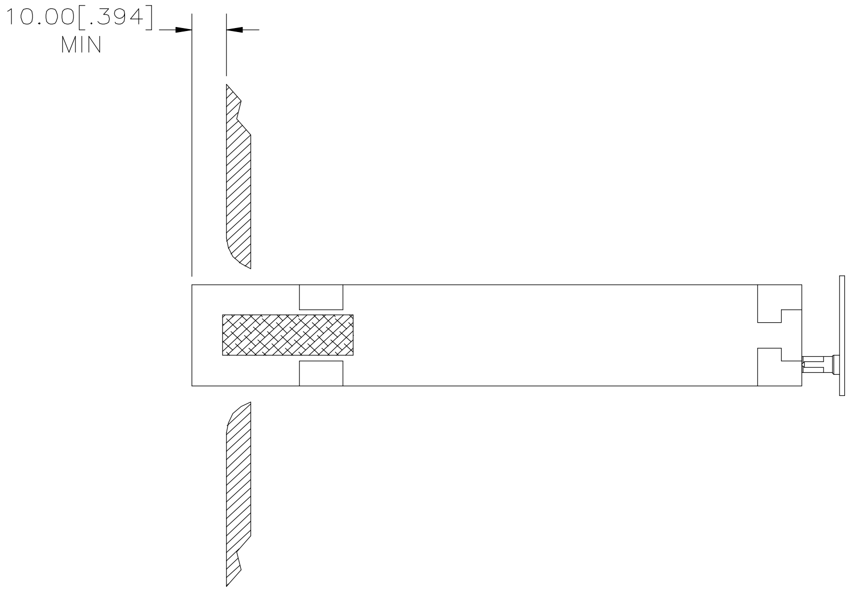

Figure 5-38. Protrusion of device beyond front bezel after an eject event

The eject scheme must not be allowed to strike the device, causing a shock event in the device if the bay has engaged the software-controlled interlock.

The eject scheme cannot override the software-controlled interlock (for more information about the software-controlled interlock, see section 5.3.2.4).

5.3.2.2 Retention Mechanism

Since Device Bay is replacing components which are rigidly mounted in current systems, features need to be built into the bay to retain the device in the bay. The retention mechanism feature

- Must retain the device in the bay during normal operation.
- Must maintain the electrical contact between the device and the bay.
- Makes use of one or more of the retention features on the device itself.

When the device is inserted into the bay, the blind-mate features on the plug and receptacle contact first and serve as the stop in the system. Since the connector is the stop in the system, the retention feature needs to comprehend the tolerance on the connector placement in the device (for more information about connector placement on the device, see section 5.2.2). The retention feature must ensure electrical contact is maintained during operation of the device. In order for this to occur, the retention mechanism must prevent the device from exceeding the wipe of the 2nd stage of the Device Bay connector (for more information, see section 4.5.1.6). This includes incorporating the tolerances on the size and location of the retention features and the location of the connector.

5.3.2.3 Impact Forces

The bay's retention mechanism and eject mechanism must not generate impact forces (for durations of less 10 milliseconds) above the specified maximums in Table 5-5, and the contact area specified in Table 5-2.

Table 5-5. Maximum impact forces

Form Factor	Impact forces (kg [lbs.])
DB32	90 [200]
DB20	45 [100]

Form Factor	Impact forces (kg [lbs.])
DB13	45 [100]

5.3.2.4 Software-controlled Interlock Mechanism

It may not always be clear to the user when it is appropriate for a device to be removed. Surprise removal may be devastating to an operating system and result in data loss. Historically, surprise removals are avoided by forcing the user to turn off the power and open the case. Device Bay does not require this.

To guard against surprise removal, every bay must have a software-controlled hardware interlock mechanism. This interlock is controlled by electromechanical methods so that the operating system can permit the removal of devices, using a software command.

A bay’s software-controlled interlock must be solely a feature of the bay. This software-controlled interlock must inhibit the ejection scheme.

The primary interface to the interlock is a release button or software icon. The interface must first trigger a removal request and receive authorization before releasing the interlock.

The system must also provide a means to override the interlock in order to remove the device in case of power outage, power supply failure, or other events that could prevent the software control from functioning properly.

A system designer can use either the retention features provided on the device, the ejection scheme, or the retention mechanism to provide the interlock requirement. An example of an eject mechanism with such a feature is shown in Figure 5-39.

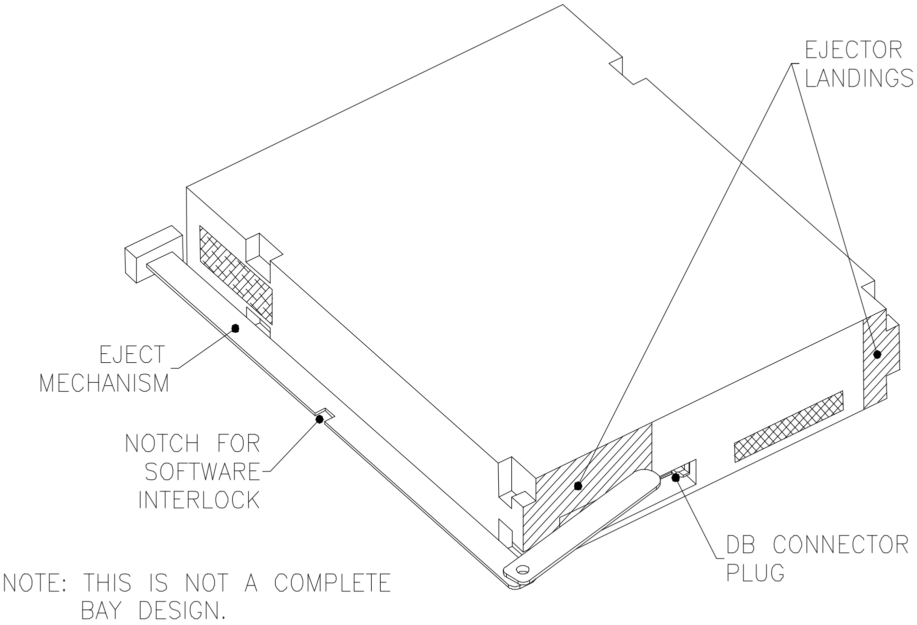
EJECTOR LANDINGS
EJECT MECHANISM
NOTCH FOR SOFTWARE INTERLOCK
DB CONNECTOR PLUG
NOTE: THIS IS NOT A COMPLETE BAY DESIGN.

Figure 5-39. Example of DB32 Device Bay interlock mechanism

5.3.2.5 EMI/ESD

The system must maintain contact with the side grounding pads after full insertion of the Device Bay connectors. Also, the system must make contact with at least one grounding pad before the Device Bay connectors begin mating.

The following criteria should be followed for designing and testing the system's EMI/ESD scheme.

5.3.2.5.1 Material Compatibility

Material selected for the mating surfaces of the ground clips must be compatible to prevent galvanic corrosion. The base alloy of the mating surfaces on the device and the system grounding clips must be within 0.3V, using non-magnetic stainless steel as the reference material.

5.3.2.5.2 Ground Contact Resistance

The initial contact resistance between the device and the chassis ground must not exceed 0.250 Ω and must not degrade during the life of the Device Bay connector (for more information, see section 4.6.3) by more than 0.020 Ω from its initial resistance measurement.

5.3.2.5.3 Ground Contact Measurement Procedure

The resistance of the Device Bay ground contact must be measured in accordance with EIA 364-B, test procedure 23 with the following test conditions:

- open circuit voltage equals 25 mV
- test current equals 100 mA

Refer to Figure 5-40 for lead connection.

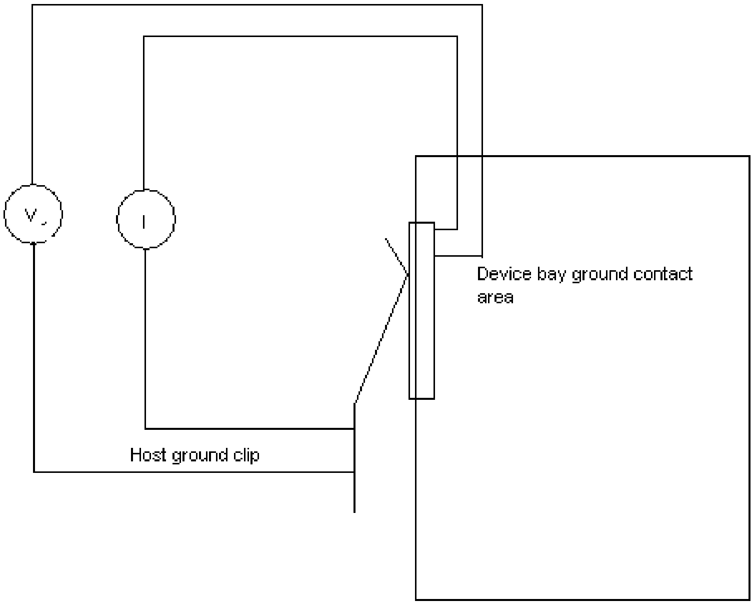

Figure 5-40. Lead connection diagram

5.3.2.6 Connector Placement and Presentation

5.3.2.6.1 Connector Presentation

A means of accurately mounting the Device Bay connector must be provided to ensure appropriate presentation of the receptacle to the device.

A backplane PCB may span multiple bays and carry the power bus rails.

A sheet-metal bracket could be used for a single bay system. Figure 5-39 shows two devices being aligned with a backplane PCB. Figure 5-41 shows an example backplane layout located on the bay.

When designing the placement of the receptacle in the bay, the overall tolerance stackup needs to include the tolerance limits of the connector wipe (see section 4.5.1.6). This needs to include such considerations as the co-planarity and mounting of the PCB.

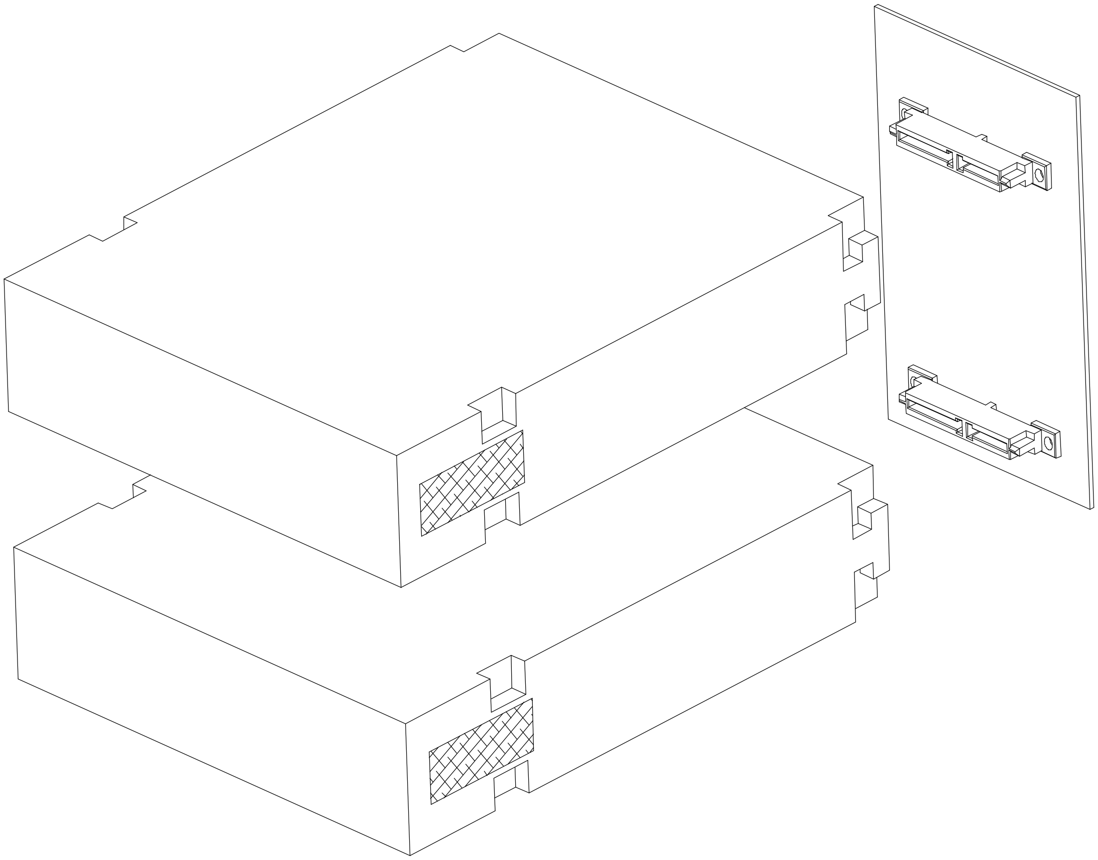

Figure 5-41. Example of a two-bay system

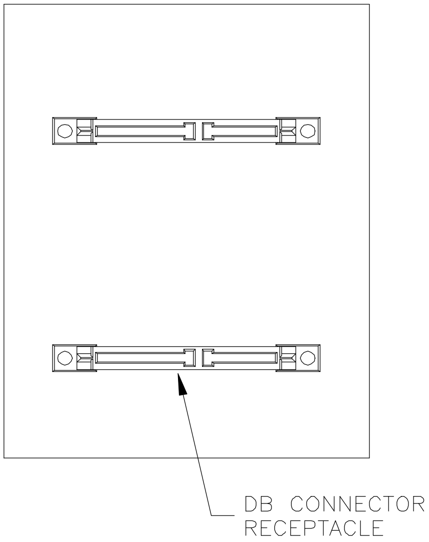

Figure 5-42. Example of a backplane layout

5.3.2.7 Device Support Features

The bay must physically support the device. An example of this feature is shown in Figure 5-43. The feature provides rough alignment for the connector.

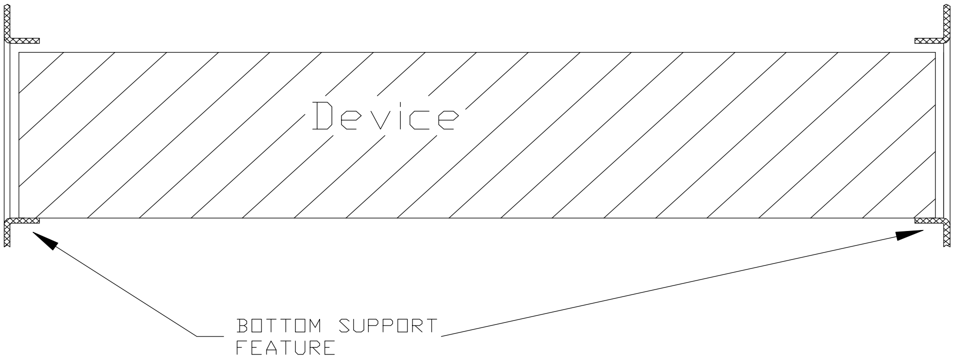

Figure 5-43. Example of a bottom support feature

Since these devices may be hard disk drives and other shock sensitive components that are no longer rigidly mounted in the system, provisions in the bay must address the needs of such devices. A method of dampening the front of the device during vibration or a shock event is needed. An example of this is shown in Figure 5-44.

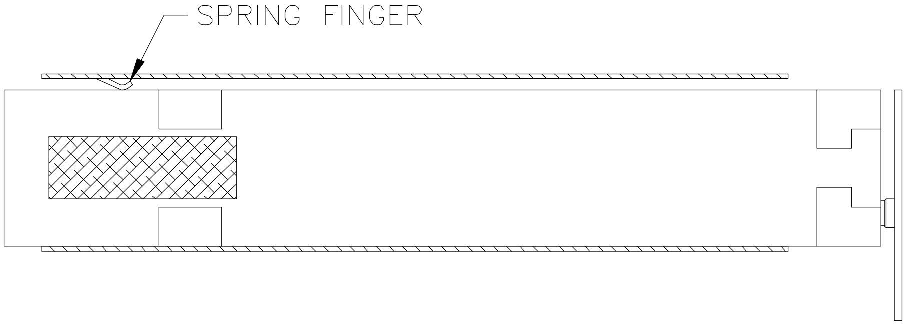

Figure 5-44. Example of shock and vibration dampening feature for DB32 device

5.4 Shock and Vibration

No minimum threshold requirements are defined for shock and vibration in this specification. This allows for various usage models to exist. The industry is expected to respond to normal market demands. Device Bay promoters will work with device suppliers to establish standard test methods for the industry.

5.5 Thermal Design Considerations

This section specifies the thermal design requirements for Device Bay devices and bays.

5.5.1 Thermal Classification

All Device Bay device and bay form factors are grouped into one of two thermal classifications: passively cooled and actively cooled. Table 5-6 below shows the thermal classification for each form factor. Passively cooled Device Bay devices transfer heat to their surroundings primarily by thermal conduction through a thin layer of air into the bay structure. Some free convection and radiation heat transfer may also take place. Passively cooled Device Bay devices offer the advantages of relatively low power consumption and minimal system cooling capacity requirements.

Actively cooled Device Bay devices transfer heat to their surroundings primarily by forced convection to the bay cooling air passages. Some conduction and radiation heat transfer may also take place. Actively cooled Device Bay devices are intended for systems that can accommodate higher power consumption and provide more cooling capacity.

Table 5-6. Cooling Method by Device Form Factor

Device Form Factor	Cooling Method	Power Dissipation (T_{cont})
DB13	Passively Cooled	4 watts
DB20	Passively Cooled	4 watts
DB32	Actively Cooled	25 watts

5.5.2 Passively Cooled Device Thermal Specification

All passively cooled devices must meet the following thermal design requirements at the worst-case thermal operating environment as defined in section 5.5.2.1.

5.5.2.1 Thermal Power Dissipation

The maximum thermal power dissipation from a passively cooled device must not exceed the power specified in Table 5-6 under steady state conditions, see section 3.

The power dissipation for a device must not exceed the Thermal Continuous power reported to the OS by the device. This includes power obtained from any and all sources (Device Bay connector, external connector, and so on).

5.5.2.2 Device Case Surface Temperature

For a passively cooled device the *average* device case surface temperature for any side must not exceed 57°C.

Additionally, the device case surface temperature must not exceed 60°C at any location (at so-called hot spots) under any operating condition.

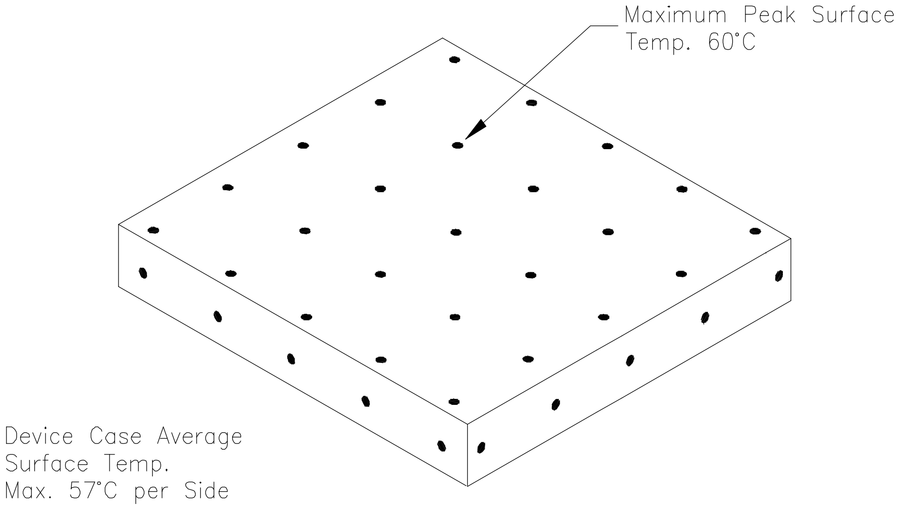

Figure 5-45. DB20 or DB13 device case surface temperature measuring

5.5.2.3 Passive Thermal Verification

5.5.2.3.1 Device Thermal Verification

All passively cooled devices must meet the thermal design requirements (as specified above) for this kind of device at an ambient temperature of 48°C (still air, sea level, and 25% relative humidity).

Note: This is for a stand-alone device, not for a device inserted in the system.

5.5.2.3.2 Bay Thermal Verification

The bay of a passively cooled device must maintain a maximum average surface temperature of 55°C on the device.

A thermal-load device for DB13 and DB20 that uniformly dissipates the maximum thermal power (see Table 5-6) can be constructed by centering a resistive load on an aluminum plate covered by a minimum of 1.5 mm of polycarbonate on all sides and by conforming to the required mechanical dimensions.

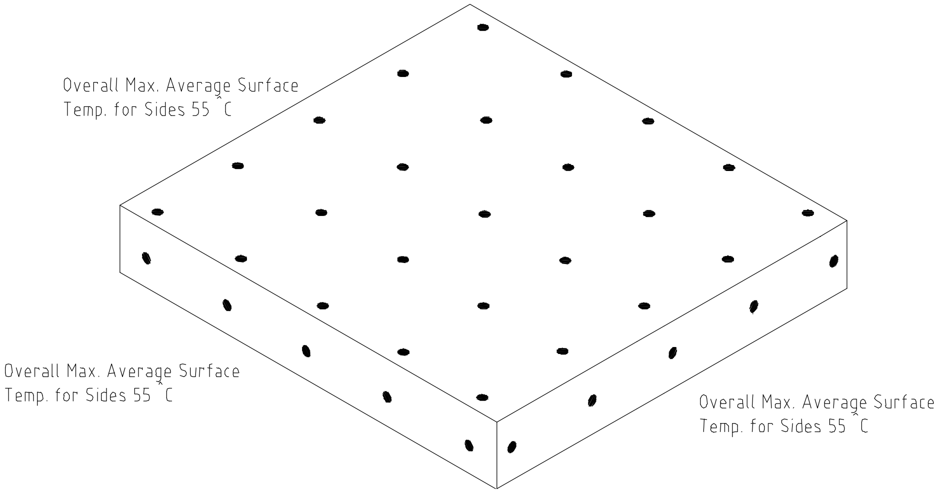
Overall Max. Average Surface
Temp. for Sides 55 °C
Overall Max. Average Surface
Temp. for Sides 55 °C
Overall Max. Average Surface
Temp. for Sides 55 °C

Figure 5-46. DB20 or DB13 device case surface temperature points

5.5.3 Actively Cooled Device Thermal Specification

All actively cooled devices must meet the following thermal design requirements at the worst-case thermal operating environment as defined in section 5.5.3.1.

5.5.3.1 Device Thermal Power Dissipation

The maximum power dissipation for a device must not exceed the power specified in Table 5-6 under steady-state conditions as outlined in section 3.

The power dissipation for a device must not exceed the Thermal Continuous power reported to the operating system by the device. This includes all power obtained from any and all sources (Device Bay connector, external connector, and so on).

5.5.3.2 Device Case Surface Temperature

The device case surface temperature must not under any operating conditions exceed 60°C at any point. The temperature gradient on the exterior of the device case must not exceed 10°C, with local gradients not

5.5.3.3 Active Thermal Verification

5.5.3.3.1 Device Thermal Verification

5.5.3.3.1.1 Device Airflow

The device must be designed to meet the case surface temperature requirement while receiving cooling airflow from the system chassis. The volume of airflow supplied is a function of the power dissipation of the device. The maximum airflow volume for a device at an ambient 35°C and 25% relative humidity is calculated as follows:

- $V_d = 0.153*P_d$ (at 10,000 feet)
- $V_d = 0.103*P_d$ (at sea level)

V_d is the airflow volume supplied to the device in cubic feet per minute (CFM), and P_d is the device power dissipation in watts.

5.5.3.3.1.2 Device Pressure Drop

The device, when installed in the thermal test bay as shown in Figure 5-47, must have a minimum airflow impedance as defined in Figure 5-48. This requirement ensures that, in multi-device systems, one device will not short-circuit the airflow of an adjacent device.

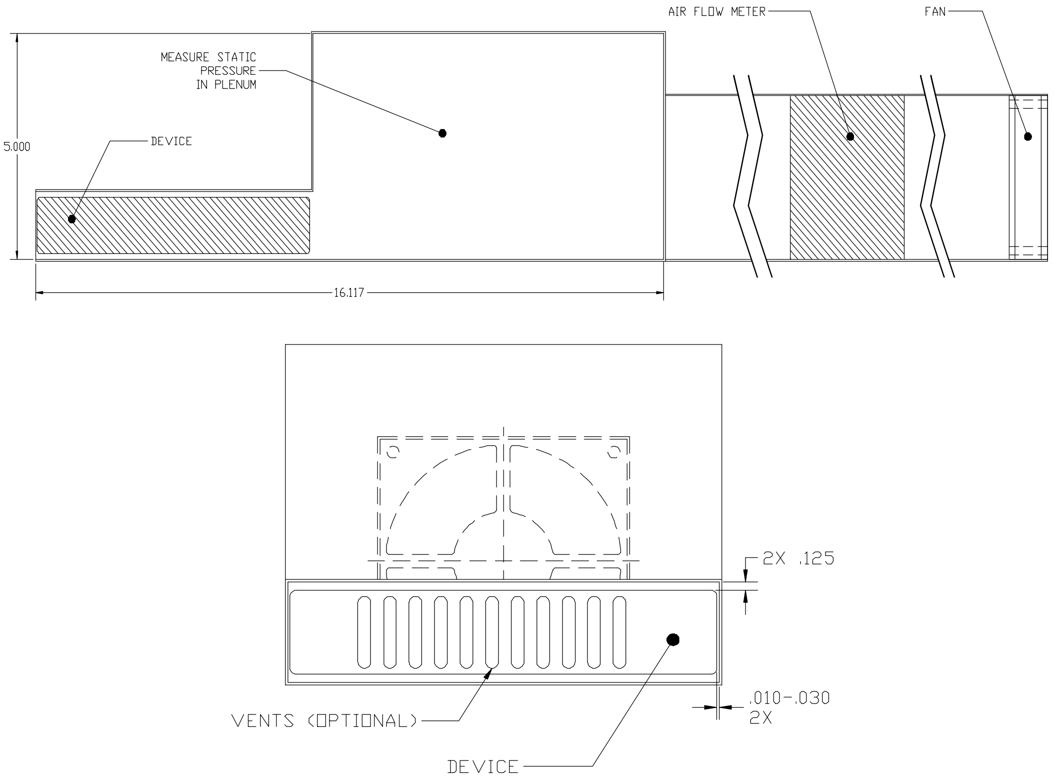

Figure 5-47. Side and front views of thermal test bay

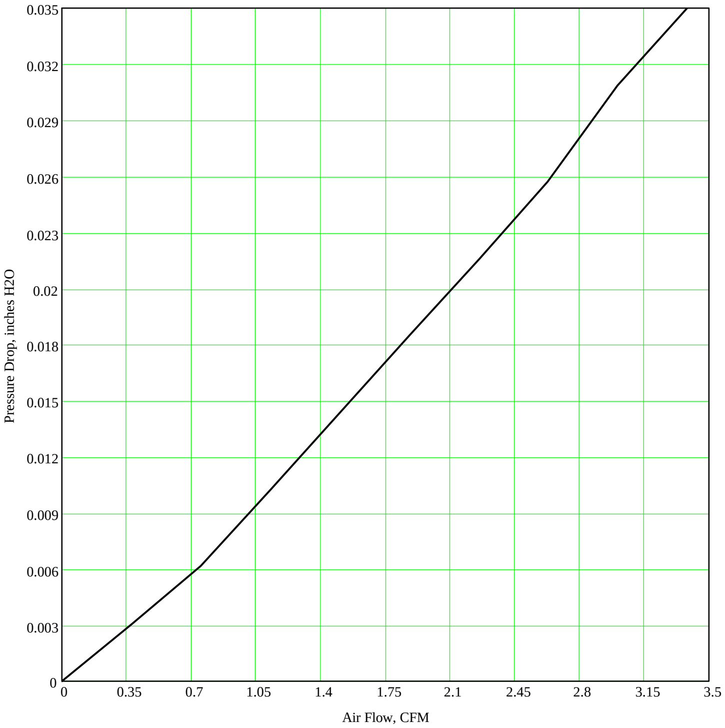

Figure 5-48. Minimum device airflow impedance

5.5.3.3.2 Actively Cooled Bay Thermal Specification

All bays must meet the following thermal design requirements at the worst-case thermal operating environment, as defined by the system manufacturer.

5.5.3.3.2.1 Bay Air Flow

All bays must be cooled by forced convection airflow. The airflow direction can be front to back, back to front, or side to side.

5.5.3.3.2.2 Bay Cooling Capacity

All systems incorporating actively cooled bays must provide cooling capacity sufficient to ensure that the case surface temperature of a device does not exceed 60°C under all operating conditions. Device Bay systems must provide a minimum cooling capacity that is rated at the maximum device power dissipation per bay.

5.5.3.3.2.3 Bay Air-Pressure Drop

The cooling design of a Device Bay system chassis must provide sufficient air-pressure drop to obtain the required airflow volume over and through the device. Particular attention should be paid to the effects of chassis design on the bay entrance and exit pressure losses. Additionally, a means must be provided to restrict airflow to a bay when a device is removed to prevent a short-circuit in the airflow of the overall system and/or adjacent devices in multi-bay; for example, a filler panel or retractable door.

5.5.3.3.2.4 Bay Cooling Verification

Bay cooling performance must be demonstrated by use of a thermal-load device operating at the thermal power rating of the Bay under test. Maximum temperature and temperature gradient data will be obtained using an array of thermocouples attached to the outer surface of the thermal-load device.

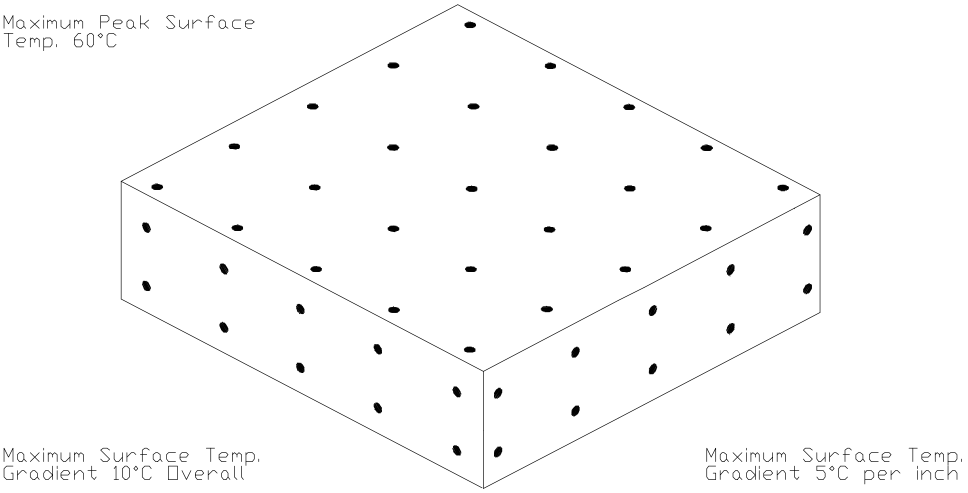

Figure 5-49. Actively cooled device case surface temperature points

6 Device Bay Controller

Although USB and 1394 are hot plug buses with support for insertion notification, it is desirable to have a separate mechanism for device insertion and removal notification. The benefits of this include:

- Discouraging the user from removing a device without first making a removal request. Removal requests enable the operating system to do whatever is necessary to ensure the integrity of user data, applications, and the operating system.
- Cleaner enumeration. This is important for association of devices with physical ports.
- Staged power consumption. This is important for mobile hosts in low-battery situations.

To achieve these benefits, a Device Bay Controller (DBC) is required on the host side of the bay. A DBC manages or provides the following features:

- Insertion events
- Removal events
- 1394 PHY port mappings
- USB port mappings

A DBC can be implemented in a variety of ways: as part of a serial bus hub function, as a stand-alone device, or as an integral part of a system-board chip set. The DBC can interface to the system either as a USB device or as an ACPI object. Using ACPI, the DBC can reside on any bus in the system that can be described through ACPI. Examples are PCI, I²C/SMBus, and embedded controllers.

The two basic Device Bay topologies are illustrated in this section of the specification. Both are described with reference to the manner in which the DBC interfaces to the operating system.

6.1 Example USB-based DBC Implementation

In Figure 6-1, the DBC is implemented as a USB device and is connected to a USB hub. The DBC must have a link/PHY interface and must be connected to its own 1394 PHY. The DBC can, but is not required to, be integrated with the hub.

The number of bays shown is for illustration purposes only. The number of bays is limited to the smaller of one less than the number of 1394 PHY ports, or the number of downstream USB hub ports, and also is limited by the capacity of the DBC. Although not shown it is perfectly acceptable to have walk-up ports connected to the same USB hub or 1394 PHY that is connected to the bays.

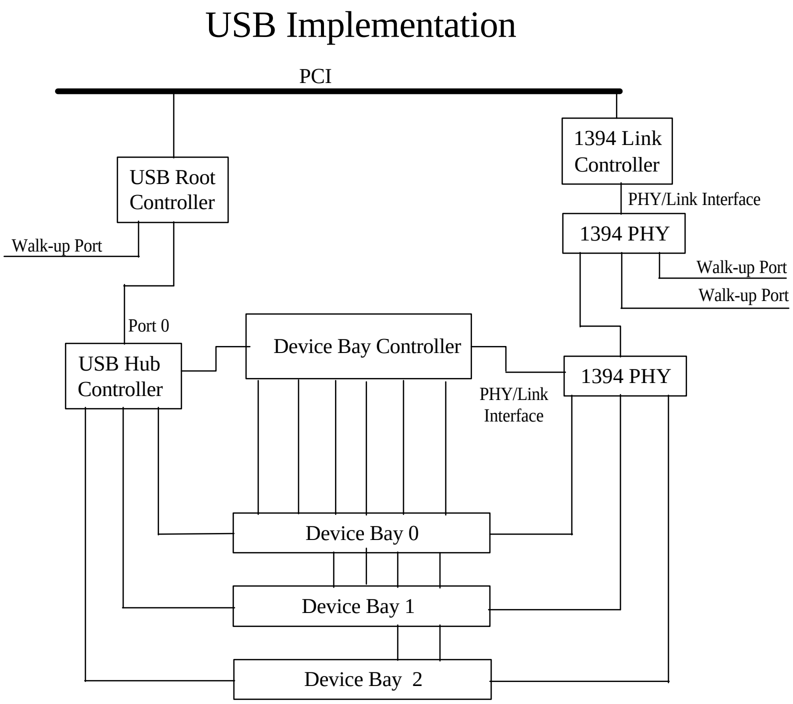

Figure 6-1. Example USB-based DBC Implementation

6.2 Example ACPI-based Implementation

The ACPI connection in Figure 6-2 indicates that ACPI Name Space and control methods are used to describe how the DBC is implemented. This implementation has no physical connection between the DBC and the on-board PHY. The number of bays shown is for illustration purposes only. The number of bays is limited to the smaller of: either the number of 1394 PHY ports, or the number of downstream USB hub ports. It is also limited by the capacity of the DBC.

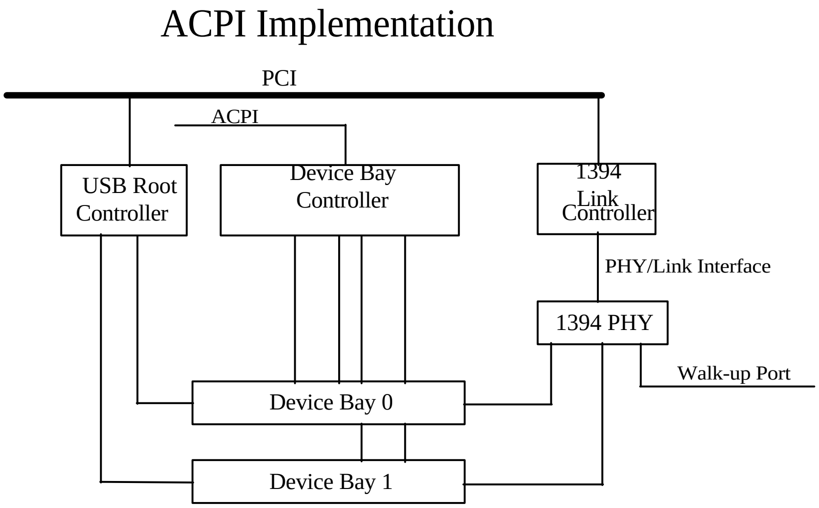

Figure 6-2. Example ACPI-based DBC Implementation

6.3 DBCs and Device Bay Subsystems

There can be more than one Device Bay subsystem in a system, and each DBC may need its own associated driver. Each subsystem must abide by the following guidelines:

- There must be one and only one DBC per subsystem.
- If the DBC is implemented as a USB device, then there must be one and only one USB hub and one and only one 1394 PHY in that subsystem.
- If the DBC is implemented as an ACPI object, then there can be more than one USB root controller or hub feeding the bays. There can also be more than one 1394 link controller feeding the bays.

6.4 DBC Specification Overview

This section provides an overview of the detailed DBC specifications that follow.

6.4.1 Diagram Legends

This section uses simplified logic diagrams to represent how certain aspects of the hardware are implemented. The symbols in Figure 6-3 are used in the logic diagrams to represent programming bits.

⊗ = Programming (enable, control, or status) bit
☒ = Sticky status bit

Figure 6-3. Logic Diagram Symbols

The round symbol represents a programming bit. As an enable or control bit, software writing this bit high or low will result in the bit being read as high or low (unless otherwise noted). As a status bit, it directly represents the value of the signal.

The square symbol represents a sticky status bit, a bit that is set by a hardware signal’s active level. A sticky status bit is cleared only by the software writing a “1” to its position.

6.4.2 Insertion Events

The bay connector defines two presence pins. Each pin indicates which of the two interfaces (USB or 1394) is to be used by an inserted device.

On the device, the appropriate pin—USBPRSN*x*# or 1394PRSN*x*#—must be connected to a ground. On compound devices (that is, a device that uses both USB and 1394), both presence pins must be connected to a ground.

When a device is inserted into a bay, the presence pin inputs to the DBC are grounded. Logic in the DBC must detect a status change on either presence pin (that is, a high-to-low transition) and, after debouncing (a debounce period of 64ms is recommended), begin a device debounce timer. The purpose of this timer is to allow the device to settle on the bay connector and to become firmly engaged by the device retention mechanism. After the device debounce time period has expired, the DBC will set a sticky status bit (DEVSTSCHG). This bit can be cleared only by the software writing a “1” to its position.

Each device status change event bit must have a corresponding enable bit (DEVSTSCHG_EN) for the purpose of enabling or disabling an interrupt event within the DBC. The logic structure is illustrated in Figure 6-4.

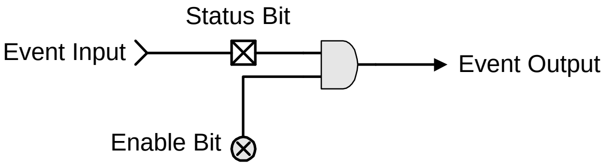

Figure 6-4. Logic Structure of Enable and Status Bits

6.4.3 Removal Events

Requested removal events establish a safer, more controlled sequence to ensure the integrity of user data, applications, and the operating system. Although there must be a software-controlled interlock mechanism on each bay, there is still the possibility that the user will forcibly remove the device from the system without first requesting permission. Such events can result in catastrophic effects to the system, such as loss of user data, loss of a swap file, and so on.

6.4.3.1 Removal Request Events

Each bay can include a hardware removal request button that the user can employ to request permission of the host system software in order to remove a device from a bay. The associated bay status indicator (for example, an LED on the bay) must signal the user when it is safe to remove the device and when it is not.

If a hardware removal request button is implemented, then the DBC is required to support the REMREQ_STS and REMREQ_EN bits and the REMREQ signal. For more information, see sections 6.6.4 and 6.6.5.

A Device Bay-compliant system or stand-alone bay must provide a way for the user to issue a removal request, either by hardware or by software.

When the button is pressed and a device is present, logic in the DBC must set a sticky status bit (REMREQ_STS). The DBC should contain debounce logic so that only a single event is generated from each button press. A debounce period of 64ms is recommended. As in the logic structure for device status change events, there must be a corresponding removal request enable bit (REMREQ_EN).

6.4.4 Port Numbers

The DBC driver requires information that enables the host operating system to track the dynamic configuration of the 1394 bus and the USB bus that are in Device Bay. In the case of an ACPI-based implementation of a DBC, this information is provided through the ACPI NameSpace. For more information, see Section 7.7.2.1. For the USB implementation of a DBC the information is provided by that DBC. For more information, see the related documents in section 1.5.

6.4.5 Power Control

For more information, see sections 6.5 and 6.6.

6.5 DBC-Specific Signals (per bay)

The signals that interface between the DBC and each bay are described in Table 6-1. The signals that interface between each device and the DBC are described in section 4.4.1.4. A method to connect the signals to the DBC is not specified in this document and is left to the system designer.

In the following table, the polarity of the DBC signals is not defined in this specification and is left up to the DBC implementer.

Table 6-1. DBC-specific Signals

Signal name	Type	Required?	Description
LOCK_EN	Output	Yes	Lock EnableThis signal controls the software-controlled interlock mechanism. It is a function of the LOCK_CTL bit in the BCER*x*. The relationship to LOCK_CTL is dependent upon the actual physical mechanism used. For example, this signal may track the state of LOCK_CTL. Alternatively it may be briefly asserted (i.e. pulsed) when LOCK_CTL transitions from “1” to “0.”
PWR_EN	Output	Yes	Power Enable • When asserted by the DBC, this signal enables the device (that is, the FET) that controls the V_{id}. • When negated, the V_{id} power is disabled.

Signal name	**Type**	**Required?**	**Description**
			This signal is controlled by a corresponding bit (PWR_CTL) in the BCER*x*.
REMREQ	Input	No (Recommended)	Removal Request • When asserted by the push button on the bay, this signal indicates that the user wishes to remove the device.
SEC_LOCK	Input	No	Security Lock Status • When asserted, the physical security lock is engaged. • When negated, the physical security lock is disengaged.

6.6 DBC Data Structures

This section specifies the DBC data structures. The data structures are implemented as a register set in an ACPI-based DBC, and as descriptors in an USB-based DBC. For a specification of the USB descriptor implementation, see the *USB Device Bay Controller Class Specification*. This section specifies the ACPI-based DBC register set.

- DBC ID registers
- DBC capabilities register
- Bay status registers (one per bay, controlled by the DBC)
- Bay control and enable registers (one per bay, controlled by the DBC)

6.6.1 Register Summary

Table 6-2 summarizes the DBC registers for an ACPI-based implementation. The associated descriptor definitions for a USB-based DBC are provided in the appropriate USB device class specification.

An ACPI-based DBC must implement a minimum of $2n + 5$ registers (where n is the number of bays supported). Although each of the bay-specific registers is defined as 32-bits wide, from the DBC driver point of view, in most cases it is *not* required to implement all 32 bits in the DBC hardware. This is due to the use of ACPI as a hardware abstraction layer.

Table 6-2. ACPI-based DBC Registers

Register[5]	**Index**	**Width**[6]	**Description**
Vendor ID	0x00	16 bits	DBC Vendor ID
Revision ID	0x04	8 bits	DBC Revision ID
Subsystem Vendor ID	0x08	16 bits	DBC Subsystem Vendor ID
Subsystem ID	0x0A	16 bits	DBC Subsystem ID
DBCCR	0x0C	32 bits	Device Bay Controller Capabilities Register
BSTR0	0x10	32-bits	Bay 0 Status Register
BCER0	0x14	32-bits	Bay 0 Control and Enable Register
•	•	•	•
•	•	•	•
•	•	•	•

---

[5] It is recommended that these registers be addressed in the order listed and with the index listed; doing so may minimize the ACPI firmware required to support them.

[6] All registers are in little endian form; that is, bit 0 is the least significant bit.

Register[5]	**Index**	**Width**[6]	**Description**
BSTR(n-1)[7]	8(n–1) + 0x10	32-bits	Bay (*n*–1) Status Register
BCER(n-1)[5]	8(n–1) + 0x14	32-bits	Bay (*n*–1) Control and Enable Register

6.6.2 ID Registers

There are four registers that provide ID information about the DBC, listed in 6.6.2.1 – 6.6.2.4.

6.6.2.1 Vendor ID

- Attribute: Read-only
- Size: 16 bits

The contents of this register identify the manufacturer of the device. The value used should be the same as the one allocated by the PCI SIG. A value of 0xFFFF is invalid.

6.6.2.2 Revision ID

- Attribute: Read-only
- Size: 8 bits

The contents of this register specify a vendor-chosen revision number.

6.6.2.3 Subsystem Vendor ID

- Attribute: Read-only
- Size: 16 bits

The contents of this register identify the subsystem manufacturer of the device, if applicable. The value used should follow the same format of the Vendor ID. For example, a DBC manufactured Company A (Vendor ID) is installed in a Device Bay subsystem that is developed by Company Z (Subsystem Vendor ID).

6.6.2.4 Subsystem ID

- Attribute: Read-only
- Size: 16 bits

The contents of this register specify a subsystem vendor-defined ID. For example, if the DBC is implemented using a microcontroller, this register might contain the firmware revision.

6.6.3 Capabilities Register

There is one register that provides the capabilities of the DBC to its driver.

---

[7] If implemented; *n* is the total number of bays implemented in the Device Bay subsystem.

6.6.3.1 Device Bay Controller Capabilities Register (DBCCR)

- Size: 32 bits
- Default Value: Implementation Dependent

Table 6-3. DBCCR Bits

Bit	Name	Access	Description
31–12	Reserved	R/O	Reserved for future use. Read as 0.
11-8	DEB_TIME[3:0]	R/O [4]	This field defines the amount of time that the DBC waits in the Device Debounce state before setting the DEVSTSCHG bit and transitioning to the Device Inserted state. The value in this field represents the debounce time in 0.5 second increments. A value of 0000b represents a debounce time of 0.5 seconds; a value of 1111b represents a debounce time of 8.0 seconds.
7–5	Reserved	R/O	Reserved for future use. Read as 0.
4	SECLOCK	R/O[8]	Security Lock Support • If set (1), then at least one bay in this Device Bay subsystem controlled by this DBC contains a physical security lock (such as a key lock) that is not necessarily controllable by software. In addition, the current state of the security mechanism (that is, engaged or disengaged) is available in the Bay *x* Status Register (BSTR*x*). • If cleared (0), then there are no physical security locks in this Device Bay subsystem.
3–0	BAYCNT[3:0]	R/O [1]	Represents the number of bays implemented with this DBC. It does *not* indicate how many bays are populated. The value in this field is a binary encoding of the number of bays (for example, 0100b indicates that four bays are implemented). **Note:** The DBC hardware may actually support a greater number of bays than indicated by the value in this field. The system designer may have chosen to implement fewer bays.

---

[8] Although some bits in this register are marked as read-only, it is acceptable to implement them with a “write-once” feature. The system (for example, the Boot Loader) must initialize these bits prior to loading the operating system. Alternatively, for example, these bits could be initialized using “strapping/configuration” pins on the DBC.

6.6.4 Bay Status Register

There is one status register for each bay.

- Size: 32 bits
- Default Value: Implementation Dependent

Table 6-4. Bay *x* Status Register Bits

Bit	Name		Access	Description
31–11	Reserved		R/O	Reserved for future use. Read as 0.
10–8	BAY_FF[2:0]		R/O[9]	This field indicates the form factor of the bay as follows: **BAY_FF[2:0] Form Factor** 000 DB32 001 DB20 010 DB13 011 Reserved for future use 100 Reserved for future use 101 Reserved for future use 110 Reserved for future use 111 Reserved for future use The value in this field does *not* represent the actual form factor of the device. For example, a DB13 device adapted to insert into a DB32 bay would still be considered a DB32 form factor as reported here.
7	SL_STS		R/O	Indicates the state of the optional bay-mounted physical security lock. • When read as a “1,” the physical security lock is engaged for this bay. • When read as a “0,” the physical security lock is disengaged for this bay. The value of this bit represents the state of the SEC_LOCK*x* pin. If the physical security lock is not implemented, then a value of “0” must be returned for read accesses.
6–4	BAY_ST[2:0]		R/O	This field represents the actual state of the bay. The states are decoded as follows: **BAY_ST[2:0] Meaning** 000 Bay Empty 001 Device Inserted 010 Device Enabled 011 Removal Requested 100 Device Removal Allowed 101 Device Debounce 110 Reserved 111 Reserved The value read from this register does *not*

---

[9] Although some bits in this register are marked as read-only, it is acceptable to implement them with a “write-once” feature. The system (for example, the Boot Loader) must initialize these bits prior to loading the operating system. Alternatively, for example, these bits could be initialized using “strapping/configuration” pins on the DBC.

Bit	**Name**		**Access**	**Description**
				necessarily represent the last state written to the Bay *x* Control and Enable Register. The DBC's bay state machine has the ability to override values previously written to the BCER*x* when an appropriate hardware event occurs (such as when the user removes the device) or when an inappropriate value is written to the bay state change request field of BCER*x*. Only hardware events can cause transitions from any state to the Bay Empty state. See Figure 6-9.
3	REMREQ_STS		R/WC	Indicates that the removal request button for this bay has been pressed. • This is a sticky bit that is set to "1" when the button is pressed *and* a device is present in the bay (see bits 1 and 0). This bit will not be set, and therefore no event is generated when the button is released. • This bit, along with the corresponding enable bit in the Bay *x* Control and Enable Register, can cause an interrupt within the DBC. • After power-on reset, this bit can only be cleared by writing a "1" to it. If a device is present upon power-on reset the state of this bit is undefined. In other words the DBC may set it or clear it. If the removal request push button is not supported, then a value of "0" must be returned for read accesses; write accesses must have no effect. An example of the logic structure for one bit is shown in Figure 6-8.
2	DEVSTSCHG		R/WC	Indicates that device status has changed in this bay. • This is a sticky bit that is set to "1" when the state of either of the presence pins for this bay transitions. The setting of this bit must be qualified with the expiration of the Device Debounce Timer when a device is inserted (i.e. presence pin(s) transition from negated to asserted). • This bit, along with the corresponding enable bit in the Bay *x* Control and Enable Register, can cause an interrupt within the DBC. • This bit can only be cleared by writing a "1" to it. An example of the logic structure for one bit is shown in Figure 6-7.
1	1394PRSN_STS		R/O	Indicates 1394 device presence for this bay. If a 1394 device is present, then the appropriate bit will be "1"; if no 1394 device is present, then the appropriate bit will be "0." The value of this bit represents the logical inverse of the state of the

Bit	Name		Access	Description
				1394 presence (1394PRSN*x*#) pin.
0	USBPRSN_STS		R/O	Indicates USB device presence for this bay. If a USB device is present, then the appropriate bit will be “1”; if no USB device is present, then the appropriate bit will be “0.” The value of this bit represents the logical inverse of the state of the USB presence (USBPRSN*x*#) pin.

An example of the device status change logic is illustrated in Figure 6-5.

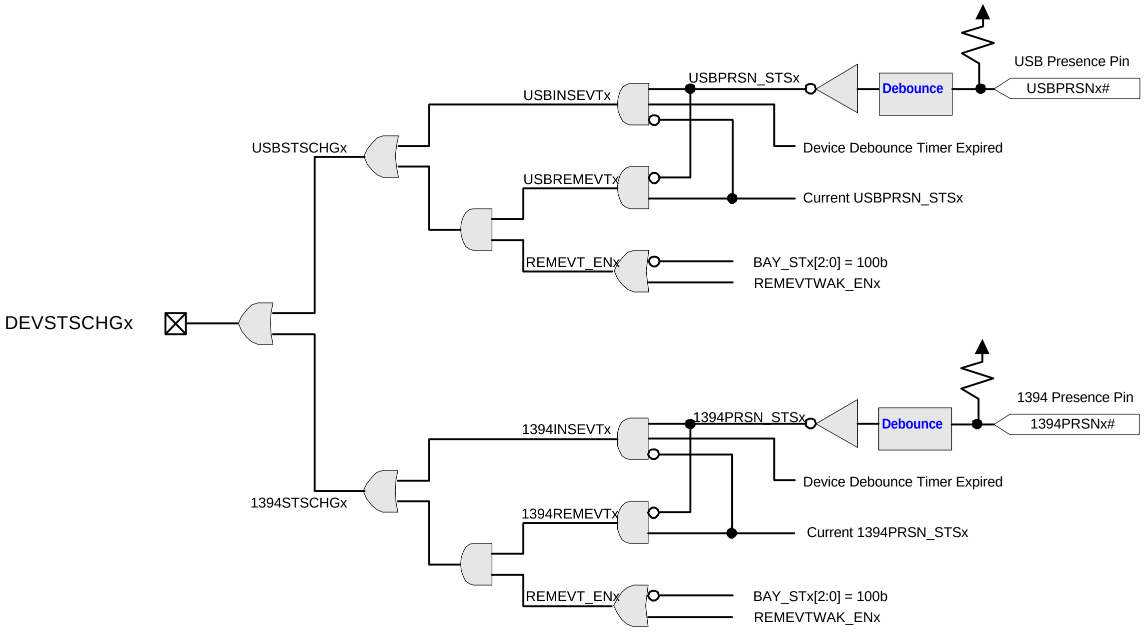

Figure 6-5. Device Status Change Logic

An example of the device removal request logic is illustrated in Figure 6-6.

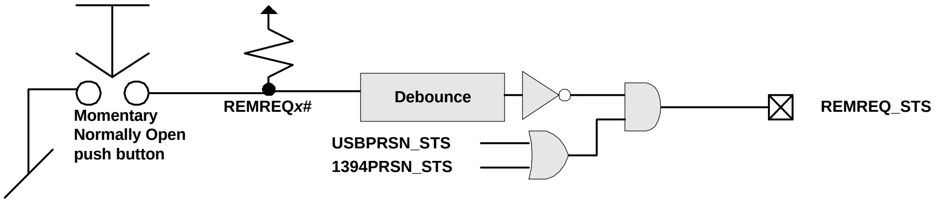

Figure 6-6. Device Removal Request Logic

6.6.5 Bay Control and Enable Register

There is one control and enable register for each bay.

- Size: 32 bits
- Default Value: 0x0000_0000

Table 6-5. Bay *x* Control and Enable Register Bits

Bit	Name		Access	Description
31–8	Reserved		R/O	Reserved for future use. Read as 0.
7	LOCK_CTL		R/W	Engages/disengages the software controlled interlock mechanism for this bay. • When set to "1" the lock mechanism is engaged (that is, the device is physically locked into the bay). • When cleared to "0" the software controlled interlock mechanism is disengaged (that is, the device can be physically removed).
6–4	BAY_STREQ[2:0]		R/W	This field represents the state of this bay as *requested* by the operating system, but it does *not* necessarily represent the actual state of the bay. The states are decoded as follows: **BAY_STREQ[2:0]** **Meaning** 000 No change to bay state requested 001 Request to change bay state to Device Inserted 010 Request to change bay state to Device Enabled 011 Request to change bay state to Removal Requested 100 Request to change bay state to Device Removal Allowed 101 Reserved for Device Debounce; no effect to bay state 110 Reserved; no effect to bay state 111 Reserved; no effect to bay state If 000b is written, then no change of state is requested *and* the previous non-zero value of this field is retained. This allows software to modify other bits in this register without inadvertently causing a bay state change. Any legal, non-zero value written to this field indicates a requested change of bay state. A bay state transition must only occur at the time of the write to this field with a device present; there is no queueing of state transition requests. If no device is present at the time a write of a non-zero value occurs, then no state transition will ever take effect for that write event; however, this field must be updated with that non-zero value. The DBC must reset this field to 000b when the device is removed from the bay.

Bit	Name		Access	Description
3	REMREQ_EN		R/W	Enables/disables an internal DBC interrupt due to a hardware removal request. Each bit is ANDed with the corresponding bit in the Bay *x* Status Register to create an interrupt. • When set to "1," an interrupt is enabled. • When cleared to "0," no interrupts will be generated for the corresponding removal request. The logic structure for one bit is shown in Figure 6-8.
2	DEVSTSCHG_EN		R/W	Enables/disables an internal DBC interrupt due to a device status change event. This bit is ANDed with the corresponding bit in the Bay *x* Status Register to create an interrupt. • When set to "1," a device status change interrupt is enabled. • When cleared to "0," no interrupts will be generated for the corresponding device status change event. The logic structure for one bit is shown in Figure 6-7.
1	REMEVTWAK_EN		R/W	Enables/disables an internal DBC interrupt due to a device removal event. This bit gates the device removal event in the DEVSTSCHG logic (BSTR*x*, bit 2). The intent of this bit is to conditionally allow device removal as a wake-up event. When the bay is not in state 100b a device removal event will always cause DEVSTSCHG to be set. • When set to "1," a device removal event will be cause DEVSTSCHG to be set when the bay is in state 100b. A corresponding interrupt will be generated provided DEVSTSCHG_EN (see bit 2) is set. • When cleared to "0," device removal events when the bay is in state 100b will not cause DEVSTSCHG to be set. The logic structure for one bit is shown in Figure 6-8.
0	PWR_CTL		R/W	Enables/disables the V_{id} rail for this bay. • When set to "1," V_{id} power is enabled. • When cleared to "0," V_{id} power is disabled. The DBC will not allow this bit to be set if there is no device present in the bay (see BSTR*x*, bits 1 and 0) or if the software controlled interlock is disengaged (see BCER*x*, bit 7). Furthermore, this bit must be cleared automatically by the DBC whenever a device is removed or whenever the software controlled interlock is

Bit	Name		Access	Description
				disengaged.

The device status change event logic is illustrated in Figure 6-7.

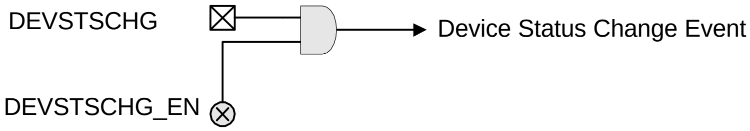

Figure 6-7. Device Status Change Event Logic

The device removal request event logic is illustrated in Figure 6-8.

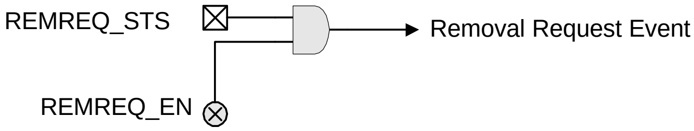

Figure 6-8. Device Removal Request Event Logic

6.6.6 Bay State Machine

This section defines the bay state transitions.

6.6.6.1 Bay State Transitions

The state of each bay can change due to hardware events (for example, device insertion) or software actions (for example, enabling the device). The following state diagrams and tables describe the bay states and transitions. Although the diagrams and tables are divided into “hardware” and “software” for the sake of clarity, in reality there is only one state machine per bay.

The behavior of the bay state machine due to hardware events is depicted in Figure 6-9. In Figure 6-9:

- PRSN=0 indicates that no device is present in the bay (neither USBPRSN_STS nor 1394PRSN_STS in BSTRx are set).The state transitions to the Bay Empty state are independent of the state of DEVSTSCHG_EN.
- PRSN=1 indicates that a device is present in the bay (either USBPRSN_STS and/or 1394PRSN_STS are set). The transition from the Bay Empty state to the Device Debounce state and the transition from the Device Debounce state to the Device Inserted state can only be made if DEVSTSCHG_EN in the BCER*x* is set.
- It is assumed that a hardware removal request button is present. If no button is present, then the transitions in and out of the “Removal Requested” state would never be made by hardware events. REMREQ=0 indicates that either REMREQ_STS in the BSTR*x* is not set or REMREQ_EN in the BCER*x* is not set, while REMREQ=1 indicates that REMREQ_STS is set and REMREQ_EN is set.
- The two arcs labeled “PRSN = 0 *” are specified as hardware fail-safe transitions. In the event that the user forcibly overrides the software controlled interlock mechanism, the DBC will return the bay state machine to the correct state. These transitions are independent of the state of DEVSTSCHG_EN.

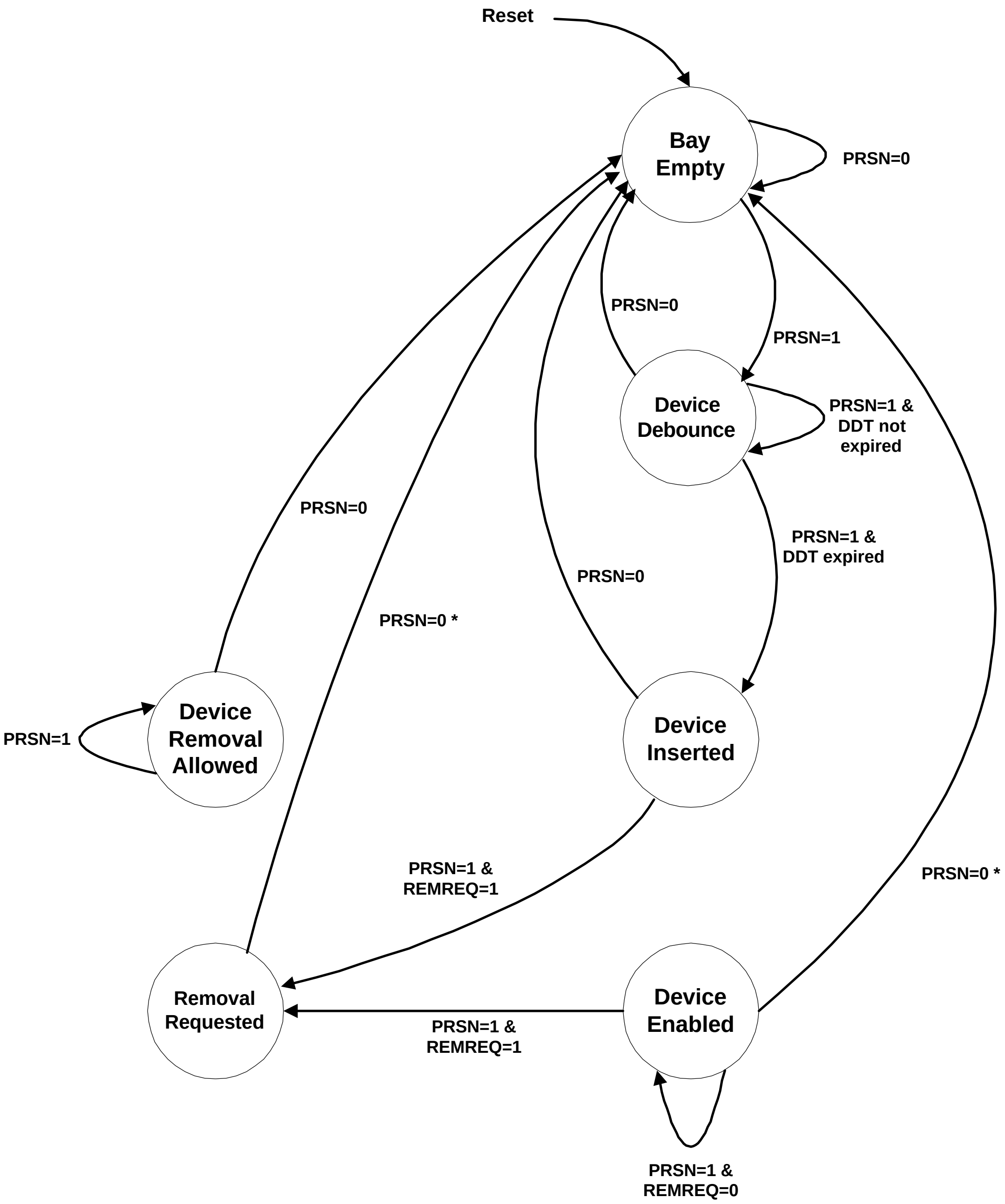

Figure 6-9. Bay State Diagram

These transitions are described in more detail in Table 6-6.

Table 6-6 Hardware Event Bay State Transition Table

Current State	Next State	Hardware Event	Conditions	Notes
- - -	Bay Empty (no devicepresent)	Power-on reset		Establish initial state.
- - -	Bay Empty (device present)	Power-on reset		A device occupying a bay at power-on time will not necessarily cause DEVSTSCHG to be set. Furthermore, since DEVSTSCHG_EN defaults

				to “0” no bay state transition will result. Either the system BIOS or the OS, upon loading, will automatically enumerate all occupied bays and MUST clear DEVSTSCHG if it was set.
Bay Empty	Device Debounce	Device inserted into the bay.	Either or both presence bits in the BSTRx transition from 0 → 1 AND DEVSTSCHG_EN in the BCERx is set.	Wait for device to settle on the bay connector and become firmly engaged by the device retention mechanism.
Device Debounce	Bay Empty	Device removed from the bay.	Presence bit(s) have transitioned from 1 → 0 independent of the state of DEVSTSCHG_EN.	Unexpected user behavior. The device was removed prior to being enumerated by the OS.
Device Debounce	Device Inserted	Device debounce time period expired.	Either or both presence bits in the BSTRx asserted (1) AND DEVSTSCHG_EN in the BCERx is set.	The DBC must set DEVSTSCHG as part of this state transition.
Device Inserted	Bay Empty	Device removed from the bay.	Presence bit(s) have transitioned from 1 → 0 independent of the state of DEVSTSCHG_EN.	Unexpected user behavior. The device was removed prior to being properly enabled by the OS. In this case the software controlled interlock mechanism may have been overridden.
Device Inserted	Removal Requested	User pressed the hardware removal request button, if present.	REMREQ_STS is set and REMREQ_EN is set.	User requested device removal before the device was enabled by the OS.
Device Enabled	Bay Empty	Device was removed from the bay.	Presence bit(s) have transitioned from 1 → 0 independent of the state of DEVSTSCHG_EN.	Unexpected user behavior. The device was removed from the bay without going through the proper removal request sequence. In this case the software controlled interlock must have been overridden.
Device Enabled	Removal Requested	User pressed the hardware removal request button, if present.	REMREQ_STS is set and REMREQ_EN is set.	User requested device removal through the hardware removal request button.
Removal Requested	Bay Empty	Device removed from the bay.	Presence bit(s) have transitioned from 1 → 0 independent of the state of DEVSTSCHG_EN.	Unexpected user behavior. The device was removed prior to completion of the proper removal request sequence. In this case the software controlled interlock mechanism may have been overridden.
Device Removal Allowed	Bay Empty	Device was removed from the bay.	Presence bit(s) have transitioned from 1 → 0 independent of the state of DEVSTSCHG_EN.	Completion of normal device removal sequence.

The behavior of the bay state machine due to software actions is depicted in Figure 6-10. The OS, using the BAY_STREQ field in the BCER*x*, can transition the state machine from one state to any other state with the following restrictions:

- A device must be present in the bay. The DBC hardware must ignore any software state transition requests when the bay is empty.
- Transitions from any state to the Bay Empty state must be done by DBC hardware only
- The transition from the Bay Empty state to the Device Debounce state can only be done by DBC hardware and is not shown in the state diagram below

For the sake of clarity in the state diagram, transitions that are a result of "unexpected OS behavior" are not shown. However, they are described in Table 6-7. All state transitions in Figure 6-10 are denoted as a write to the BAY_STREQ field of the BCER*x* (bits [6:4]). The write to the BCER*x* causes the state transition, as opposed to a static value in the BCER*x*. For example, the notation 010b → BAY_STREQ indicates that the state transition occurred when 010b was written to bits [6:4] of the BCER*x*, at the time the bay was occupied..

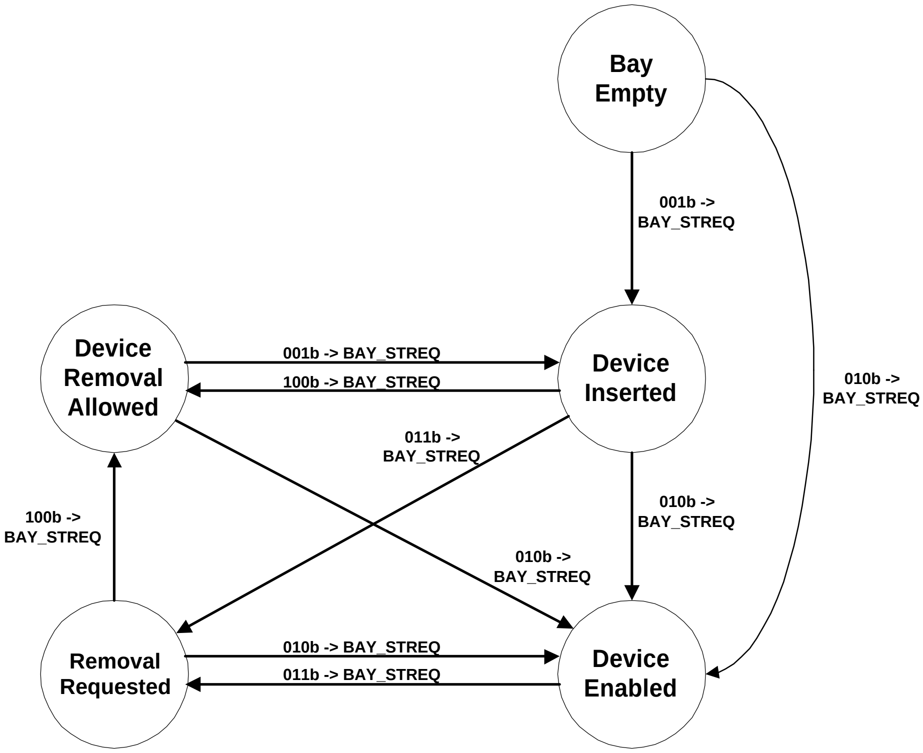

Figure 6-10. Software Action Bay State Diagram

These state transitions are described in more detail in Table 6-7. In this table the term “Bay Empty” actually means that although a device is present – a device MUST be present in order to cause any software initiated state transitions – the DBC hardware could not transition out of the Bay Empty state.

Table 6-7. Software Action Bay State Transition Table

Current State	Next State	Software Action	Notes
Bay Empty (device is present)	Device Inserted	001b → BAY_STREQ	This transition request could be the result of a device currently in the bay with DEVSTSCHG_EN cleared (as such, the DBC hardware could not transition the bay state).
Bay Empty (device is present)	Device Enabled	010b → BAY_STREQ	This transition request is a result of a device that has been properly enumerated and enabled on its native bus(es). In all likelihood the DEVSTSCHG_EN bit was cleared so DBC hardware could not first transition the bay state from Bay Empty to Device Debounce to Device Inserted.
Bay Empty (device is present)	Removal Requested	011b → BAY_STREQ	Unexpected OS behavior.
Bay Empty (device is present)	Device Removal Allowed	100b → BAY_STREQ	Unexpected OS behavior.
Device Inserted	Device Enabled	010b → BAY_STREQ	Device properly enumerated and enabled on its native bus(es).
Device Inserted	Removal Requested	011b → BAY_STREQ	Unexpected user behavior. User request device removal through the UI before the device was enabled.
Device Inserted	Device Removal Allowed	100b → BAY_STREQ	OS could not enable the device. This could be a result of the OS’s failure to properly enumerate the device, lack of sufficient operational power, etc.
Device Enabled	Device Inserted	001b → BAY_STREQ	Unexpected OS behavior.
Device Enabled	Removal Requested	011b → BAY_STREQ	User request device removal through the UI.
Device Enabled	Device Removal Allowed	100b → BAY_STREQ	Unexpected OS behavior.
Removal Requested	Device Inserted	001b → BAY_STREQ	Unexpected OS behavior.
Removal Requested	Device Enabled	010b → BAY_STREQ	OS decided that device removal was not allowed. This could be the result of an active application disallowing the removal. Alternatively, the user could have cancelled the removal request through the UI.
Removal Requested	Device Removal Allowed	100b → BAY_STREQ	OS has completed the device removal sequence.
Device Removal Allowed	Device Inserted	001b → BAY_STREQ	User has requested to “re-use” the device through the UI. This state transition eliminates the need for the user to remove the device and then immediately re-insert it. This could be especially useful in the presence of an engaged physical security lock. This bay state transition should be used in order to provide user feedback (via the bay status indicator) while the OS is performing the steps necessary to re-enable the device.
Device Removal Allowed	Device Enabled	010b → BAY_STREQ	Unexpected OS behavior. The OS should first transition the bay to the Device Inserted state while performing the steps

			necessary to re-enable the device.
Device Removal Allowed	Removal Requested	011b → BAY_STREQ	Unexpected OS behavior.

6.6.7 System Wake-up Events

The user must be able to choose to wake a sleeping system by simply inserting a device into a bay, removing a device from a bay, or by pressing the removal request button, if present. As such, the DBC must treat insertion, removal, and removal request events as system wake-up events, if enabled. This has the following implications:

- The DBC must be powered from a standby or auxiliary power rail.
- For an ACPI-based implementation, the DBC's ACPI interrupt output must be defined as a wake-up source.
- For a USB-based implementation, the DBC must generate a resume event.

The user may be allowed to enable or disable these events as sources for system wake up. Support for this behavior exists in the event structures defined by the BSTR*x* and BCER*x*.

6.6.8 Software Controlled Interlock

It may not always be clear to the user when a device can be removed without causing a system hang-up or damage to the device or the system. Surprise removal can be devastating to an operating system and can result in data loss. In a Device Bay system, surprise removals are more likely because it might not be clear to the user that a device is in use, and it will be much easier for the user to grab a device and pull it out.

To guard against surprise removals, every bay must have a software-controlled interlock mechanism. The software controlled interlock is controlled by electromechanical methods such that the operating system can permit the removal of devices using a software command.

The primary interface to the software controlled interlock is a release button or software icon that must first trigger a removal request to the system and receive authorization before releasing the software controlled interlock.

The software controlled interlock is not intended to provide physical security.

6.6.9 Physical Security

In some operating environments (such as commercial PCs), it may be necessary to provide physical security against theft of a Device Bay device. After a user has requested device removal and the operating system allows it, the software controlled interlock mechanism is disengaged. However, at that point, it may still be necessary to retain the device in the bay. There are provisions in this specification to allow OEM-specific implementations of a physical security mechanism (refer to the DBCCR and BSTR*x*). The OEM may choose to implement a separate physical security mechanism that is not necessarily controllable by the operating system (such as a keylock). Alternatively, the OEM may choose to utilize the same mechanism that serves as the software controlled interlock; in order to prevent potential damage to this mechanism an OS component is required to check for an engaged physical security lock before disengaging the software controlled interlock.. The definition and implementation of the physical security mechanism is beyond the scope of this document and is left to the OEM.

6.6.10 Power Capabilities Registers

To improve the power manageability of Device Bay, the power that can be provided to the bays by each voltage present in a Device Bay subsystem is reported by a set of 32-bit Device Bay Power Capabilities (DBPC) entries.

- For a USB implementation of the DBC, the location of the DBPC entries is specified by the *USB Device Bay Controller Class Specification.* (A reference to the USB DBC specification, which describes the mechanism used by a USB-based DBC to read and write the DBC power registers, will be included in this document when a draft version is available on the web.)
- For an ACPI implementation of the DBC, a set of DBPC entries are data objects in the ACPI Name Space.

There are two types of DBPC entry formats:

- Voltage/ Power Capabilities
- Total Power/Thermal Dissipation Capabilities

6.6.10.1 Device Bay Subsystem Power Environment Overview

This section presents an overview of the Device Bay subsystem power environment, which the DBPC entries describe. For more detailed information about the power environment and the Device Bay power bus, see section 3.

There are three voltages present in Device Bay: +3.3V, +5V, and +12V.

The +3.3-volt V_{id} is the enumeration voltage. The enumeration voltage is used at low power to enumerate a device in a bay. Device Bay allots 1.5 watts per bay for device enumeration. A device must never draw more than 1.5 watts from V_{id}.

6.6.10.2 DBPC Entries Overview

For the Device Bay subsystem it controls, the DBC specifies its power capabilities by reporting:

- The maximum continuous power (E_{cont}) that each voltage in the subsystem can deliver.
- The maximum peak power (E_{peak}) a voltage can deliver.
- The total amount of power (T_{aggr}) that can be delivered.
- The thermal power (T_{cont}) that can be dissipated.

All DBC implementations must have a minimum of four DBPC entries. In general, a different DBPC entry reports each of the following:

- +3.3 volts maximum continuous power (using the Voltage/Power entry format)
- +12 volts maximum continuous power (using the Voltage/Power entry format)
- +5 volts maximum continuous power (using the Voltage/Power entry format)
- Total amount of power (using the Total Power/Thermal Dissipation entry format)

Additionally, each voltage can specify:

- How much peak power can be provided by each voltage, if different from the maximum continuous power that can be delivered (using the Voltage/Power entry format).
- How much thermal power can be handled, if different from the amount of total power that can be delivered (using the Total Power/Thermal entry format).

6.6.10.3 DBPC Voltage/Power Entry Format and Usage Rules

The DBPC Voltage/ Power entry format consists of 32 bits, as shown in Table 6-8.

Table 6-8 DBPC Voltage/Power Enrty Format

Voltage	Power type	Continuous/Peak Power (E_{cont}/E_{peak})
(8)	(2)	(22)

- *Voltage* is 8-bits long, with a granularity of 1 decivolt per bit. Table 6-9 shows the encoding for three Device Bay voltages.

Table 6-9 Voltage Entries

Voltage	Code
+3.3 V	00100001 (33_{10})
+5.0 V	00110010 (50_{10})
+12.0 V	01111000 (120_{10})

- *Continuous/Peak Power* is a 22-bit entry with a granularity of 1 milliwatt per bit.
- The meanings of the *Power Type* bits are described in Table 6-10.

Table 6-10 Power Type Bit Field Definitions

Current Type bits	Description
00	The total amount of continuous power that the voltage encoded in the *Voltage* field of the entry can provide to the Device Bay subsystem controlled by the DBC.
01	The total peak power available from the voltage encoded in the *Voltage* field. The peak power value does not indicate an increment above the continuous power, but the absolute amount of power available for a surge condition.
10	Reserved
11	Reserved

Rules for using the DBPC Voltage/ Power entry format include:

- Every set of DBPC entries must include a Voltage/ Power entry that reports the maximum continuous power (E_{cont}) each voltage in the Device Bay subsystem can deliver. For example, if a Device Bay subsystem provides +3.3 volts, +5 volts, and +12 volts, three Voltage/ Power formatted entries must be used.
- Optionally, for each voltage, a Voltage/ Power formatted entry can be used to report the peak power (E_{peak}) for that voltage. Notice that if only one Voltage/ Power type entry exists for a given voltage, the maximum continuous and peak power are assumed to be equal.

  For example, a Device Bay subsystem provides +3.3 volts, +5 volts, and +12 volts, and if there is no difference between E_{cont} and E_{peak} for any of these voltages, then only three Voltage/ Power formatted entries must be used. However, if E_{cont} and E_{peak} are different for all three voltages, then six Voltage/ Power type entries must be reported in the DBC.

6.6.10.4 DBPC Total Power/Thermal Dissipation Entry Format and Usage Rules

The format of a Total Power/Thermal Dissipation entry is shown in Table 6-11. The general entry format is very similar to the Voltage/Current entry type format.

Table 6-11 DBPC Total Power/Thermal Dissipation Entry Format

11111111	Power type	Watts
(8)	(2)	(22)

The value of the first 8 bits must always be 0xFF.

- The meanings of the *Power Type* bits are described in Table 6-12.

Table 6-12 Power Type Bit Field Definitions

Power Type bits	Description
00	The total amount, in watts (T_{cont}), that the Device Bay subsystem can dissipate (how much heat the Device Bay subsystem can remove).
01	The total power, in watts (T_{aggr}), that can be delivered to the Device Bay subsystem controlled by the DBC.
10	Reserved
11	Reserved

- *Watts* is a 22-bit field with a granularity of 1 milliwatt per bit.

Rules for using the DBPC Total Power/Thermal entry format include:

- A DBC must report a Total Power/Thermal Dissipation entry with the Electrical/Thermal field set to 0x01 and a T_{aggr} electrical value.
- A DBC can report a Total Power/Thermal Dissipation entry with the Electrical/Thermal field set to 0x00 and a T_{cont} thermal dissipation value. If no such entry is reported, it is assumed that the thermal dissipation value (measured in watts) is equal to the required electrical value.

6.6.10.5 Host Software Uses of the DBPC Entries

Host software can manage power to a Device Bay subsystem by using the data reported in the DBPC entries. The power manager entity in the host:

- Must ensure that no voltage is allowed to deliver more than the maximum continuous power reported in the *Power Type* field of its Voltage/ Power entry.
- Must ensure that the sum of the wattage delivered by the three voltages present in the Device Bay subsystem does not exceed the value reported in the *Watts* field of the Total Power/Thermal Dissipation entry.
- Must insure that the sum of the peak powers for a given voltage for Device Bay devices is not larger than the peak power capability of the same voltage reported by the DBC.

Also, the DBC must ensure that the combined thermal power of all devices in the Device Bay subsystem is less than the maximum thermal power that the Device Bay subsystem can dissipate. It is likely that the sum of the maximum power that can be delivered by each voltage will be higher than the *Watts* field of the Total Power/Thermal Dissipation entry.

The software must maintain the relationships shown in Table 6-13.

Table 6-13 Power Relationships

Relationship	Description
$E_{cont\_dev1} + E_{cont\_dev2} + \ldots + E_{cont\_devn} < E_{cont\_DBC}$	The sum of the power of all Device Bay devices for a given voltage is less than the reported maximum continuous power for that voltage.
$E_{peak\_dev1} + E_{peak\_dev2} + \ldots + E_{peak\_devn} < E_{peak\_DBC}$	The sum of the peak powers for each voltage of the Device Bay devices is less than the peak power values reported by the DBC. In most cases, the relationship will simplify to one device compared against the peak power reported in the DBC, shown as follows: $E_{peak\_dev} < E_{peak\_DBC}$
$T_{aggr\_dev1} + T_{aggr\_dev2} + \ldots + T_{aggr\_devn} < T_{aggr\_DBC}$	The sum of the power drawn from each voltage in a Device Bay subsystem cannot be more than the maximum power allowed in the

	subsystem.
6.6.10.5.1.1.1 $T_{cont\_dev1} + T_{cont\_dev2} + \ldots + T_{cont\_devn} < T_{cont\_DBC}$	The thermal energy generated by all the Device Bay devices must be less than the thermal energy the Device Bay subsystem can handle.

Note: For a description of the ACPI DBPC register model, please refer to Annex C of this specification. For the USB DBC DBPC descriptor model, please go to http://www.usb.org and refer to the *USB DBC Specification.*

7 Software

This section describes a Device Bay-capable operating system (OS). This description is presented in terms of the support required by an OS from the mechanical, electrical, and Device Bay Controller (DBC) components of a Device Bay. The OS is involved:

- When the user inserts a device in a bay.
- When the user requests removal of a device from a bay.
- When a bay makes a transition from one power management state to another.
- When the user boots the PC from a mass storage device in a bay.

The relationship between the OS, and USB-based and ACPI-based DBCs is described in this section.

7.1 Overview

This section provides a general answer to the question, “Does any functionality have to be added to an OS

7.1.1 Device Insertion

To manage a device insertion event, an OS is required that carries out the tasks listed in the fourth column of the table found in section 7.9.1.

7.1.2 Device Enumeration

Each device in a bay is enumerated by its native bus enumerator (that is, the 1394 and USB bus device drivers).

7.1.3 Device Control

Device control is handled through the native bus device drivers (that is, the 1394 and USB bus device drivers).

7.1.4 Coordinating Activities Between a Bay and Inserted Devices

The activities that must be coordinated are:

- Under normal operating conditions, a device must not be removed from the bay unless the OS authorizes it. An OEM-provided device removal override mechanism can be used to recover from catastrophic system failure.
- The device operating power (V_{op}) must not be utilized by any device until the OS authorizes it.
- The power that enables the OS to enumerate a device (V_{id}) must be enabled and disabled as devices are inserted and removed from a bay.
- The power state of a bay must be consistent with the power state of any 1394 or USB device inserted in that bay, as defined by the OnNow initiative. Note that devices transition between power states using mechanisms defined on their native buses.

For more information about the tasks a Device Bay-capable OS carries out in coordinating these activities, see the task sequence tables at the end of the Software section. Information about these coordination activities can be found throughout the task sequence tables.

7.1.5 Device Removal Request

To manage a device removal request event, OS components carry out the tasks listed in the column headed “OS” in the table found in section 7.9.1.

7.1.6 Mapping Devices to Bays

The DBC driver requires information that enables the host operating system to track the dynamic configuration of the 1394 bus and the USB bus that are in Device Bay. In the case of an ACPI-based implementation of a DBC, this information is provided through the ACPI NameSpace. For more information, see Annex C. For the USB implementation of a DBC, the information is provided by that DBC. For more information, see section 7.7.2.1.

7.2 The OS and Bay Electrical Features

The Bay electrical system must provide separate V_{id} and V_{op} power rails. The OS must enable V_{id} in order to enumerate a device. For more information about V_{id} and V_{op}, see Section 3.

For more information about how an OS manages and budgets V_{id} and V_{op} power, see section 7.5.

7.3 The OS and the Mechanical Features of the Bay

The Device Bay-capable OS architecture counts on certain Device Bay mechanical features. The Device Bay mechanical system

- Must provide a software-controlled device interlock mechanism that discourages surprise removal of a device from a bay. This enables the OS to authorize removal of a device, by responding to a user removal request.
- Can provide a device removal request button on a bay that starts the sequence of events leading to OS authorization to remove the device. The OS must display a user interface (UI) that enables the user to initiate a device removal request. The OS counts on the DBC registers to report whether or not the bays controlled by a DBC have a removal request button.
- Can provide a physical security lock. The OS counts on the DBC registers to report whether any of the bays in the subsystem controlled by the DBC have a physical security lock, and to report on the status of the lock (engaged or not).

For more information about the software-controlled interlock mechanism, the eject mechanism, the optional request removal button, and the optional security lock mechanism, see section 5.

7.4 The OS and the Device Bay Controller

The Device Bay Controller (DBC) provides a register set that the OS uses to carry out its bay management functions of device insertion management, device removal management, coordinating bay and device states, mapping devices to bays, and bay power management.

A high-level block diagram of the DBC is shown below. The OS accesses the Device Bay mechanical and electrical features exclusively through the DBC register set. The DBC registers also provide the OS with information about the bays controlled by the DBC, such as the device presence pins. For the complete detailed specification of the DBC register set, see section 6.

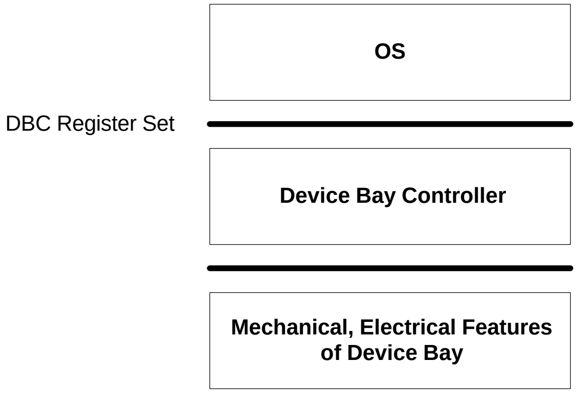

Figure 7-1. OS/DBC Interface

Note that a DBC can be implemented in a variety of ways. For more information about the different ways a DBC can be implemented, see the introduction to section 6.

7.4.1 OS Uses of the DBC Registers

The way an OS uses each of the DBC registers is summarized in the following table. For example sequences of DBC register usage, in the context of bay mechanical and electrical features, see the topics "Device Insertion Task Sequence Tables" and "Device Removal Request Task Sequence Tables" in sections 7.9.1 and 7.9.2, respectively.

Staus bit processing must only occur if the corresponding enable bit is set.

Note: The DBC ID registers (Vendor ID, Revision ID, Subsystem Vendor ID, and Subsystem ID) are not used in the following table. An OS can use these registers in any way that is useful. For example, an OS could use Vendor ID to choose which DBC driver to load.

Table 7-1 OS Uses of the DBC Registers

Register/Field	Description	OS Usage
DBCCR/SECLOCK	DBC Security Lock Support	Read Only. • OS can read this bit to determine whether or not any bay in the Device Bay subsystem incorporates a hardware security lock, which is an optional feature of Device Bay. Note that a hardware security lock may be different than the required software-controlled interlock. If the subsystem incorporates a security lock, the OS may need to display some additional UI during the device insertion and removal events to prompt the user to do things with the security lock.
DBCCR/BAYCNT	DBC Bay Count	Read Only. • OS can read this value to determine the maximum number of BPMRx, BSTRx, and BCERx registers are in the DBC.

Register/Field	**Description**	**OS Usage**
CGUIDR	DBC GUID Register	Read Only. • OS reads the GUID of the 1394 Link associated with the DBC.
BPMRx/1394PHYPN	Bay x 1394 PHY Port Number	Read Only. • OS reads the port number of the 1394 PHY connected to bay x.
BPMRx/USBPN	Bay x USB Port Number	Read Only. • OS reads port number of the USB port connected to bay x.
BSTRx/BAY_FF	Bay x Form Factor	Read Only. • Indicates the form factor of bay x (DB32, DB20, or DB13).
BSTRx/SL_STS	Bay x Security Lock Status	Read Only. • If a security lock is present (see DBCCR/SECLOCK definition), then OS can read this bit to determine whether the security lock is locked or unlocked. The OS can use this information in the additional UI it may need to display during the device insertion and removal events, to prompt the user to do things with the security lock. Note that this is a read-only status bit only; the OS has no control of a security lock through the DBC registers.
BSTRx/BAY_ST	Bay x LED Status	Read Only. • OS can read these bits to determine the current state of the bay. For more information about the bay states and state transitions, see section 6.6.6.
BSTRx/REMREQ_STS	Bay x LED Removal Request Status	Read/Write-to-Clear. • OS reads this bit to determine cause of interrupt on bay x; if this bit is set, the user caused the interrupt by pressing the removal request button on bay x. • OS also writes a “1” to this bit to clear it.
BSTRx/DEVSTSCHG	Bay x Device Status Change	Required and Read/Write-to-Clear. • OS reads this bit to determine cause of interrupt on bay x; if this bit is set, interrupt was caused by a device being inserted into or removed from bay x. • OS also writes a “1” to this bit to clear it. OS can read the 1394PRSN_STS and USBPRSN_STS bit values to differentiate between an insertion event and a removal event.
BSTRx/1394PRSN_STS	Bay x 1394 Presence	Read Only.

Register/Field	Description	OS Usage
	Status	• OS reads this bit to determine if a 1394 device is in bay x.
BSTRx/USBPRSN_STS	Bay x USB Presence Status	Read Only. • OS reads this bit to determine if a USB device is in bay x.
BCERx/LOCK_CTL	Bay x Interlock Control	Read/Write. • OS writes a “1” to the bit to physically lock the device into bay x, using the software-controlled interlock provided by the bay mechanical subsystem. • OS writes a “0” to free the software-controlled interlock so the device in bay x can be removed. • OS can read this bit to determine the state of the software-controlled interlock.
BCERx/BAY_STREQ	Bay x Bay State Request	Required and Read/Write. • OS writes 001b to request bay state change to Device Inserted. • OS writes 010b to request bay state change to Device Enabled; usually done after the OS has successfully enabled a device in bay x. • OS writes 011b to request bay state change to Removal Requested. • OS writes 100b to request bay state change to Device Removal Allowed.. • OS writes 000b to make no change to the bay state (perform no action, leave it unchanged) OS can read these bits to determine the last bay state change request it sent to the DBC. OS must read BSTRx/BAY_ST to determine the actual current state of the bay. For more information about the bay states and state transitions, see section 6.6.7.
BCERx/REMREQ_EN	Bay x Removal Request Interrupt Enable/Disable	Read/Write. • OS writes a “1” to this bit to enable a removal request interrupt from bay x. • OS writes a “0” to this bit to disable a removal request interrupt from bay x. • OS can read this bit to determine whether the removal request interrupt for bay x is currently enabled or disabled.
BCERx/DEVSTSCHG_EN	Bay x Device Status Change Enable/Disable	Read/Write. • OS writes a “1” to this bit to enable a device status change interrupt from bay x. • OS writes a “0” to this bit to disable a device status change interrupt from bay x.

Register/Field	Description	OS Usage
		• OS can read this bit to determine whether the device status change interrupt for bay x is currently enabled or disabled.
REMEVTWAK_EN	Bay xDBC Internal Interrupt Enable/Disable	Read/Write • OS writes a “1” to this bit to enable interrupts. • OS writes a “0” to this bit disable interrupts.
BCERx/PWR_CTL	Bay x V_{id} Power Control	Read/Write. • OS writes a “1” to this bit to turn on the V_{id} power rail in bay x. • OS writes a “0” to this bit to turn off the V_{id} power rail in bay x. • OS can read this bit at any time to determine whether the V_{id} power rail for bay x is currently ON or OFF.

7.5 The OS and Bay Power

From the point of view of system power management, a bay is another device that needs to be power managed.

7.5.1 Device Power Management State Model

Bays are devices that can transition between the states D0, D1, D2, and D3. The OS controls V_{id} during bay power state transitions between D0, D1, D2, and D3. In general, the OS directly enables or disables V_{id} using the Bay x V_{id} Power Control bit (BCERx/PWR_CTL) provided through the DBC interface. The use of V_{op} is controlled by the native bus power management components. V_{id} is enabled in all states except D3.

For more information about power distribution between a bay and a device and the V_{op} and V_{id} power rails, see section 3.

7.5.2 The OS and Bay Power Management Coordination Issue

The OS must coordinate the power state of a bay and the power state of an inserted device. The power state of the device is controlled by its native bus power management and the power state of the device always dictates the power state of the bay. The OS must ensure the power state of the bay is consistent with the power state of any 1394 or USB device inserted into the bay as defined by the OnNow initiative.

7.5.3 Power Budgeting

The Device Bay Power Capabilities (DBPC) registers provide the OS with the information needed to provide power budgeting for a Device Bay subsystem.

The DBPC registers report, for the subsystem controlled by the DBC:

- The maximum power that each voltage can deliver.
- The maximum peak power each voltage can deliver.
- The total amount of power that can be delivered.
- The thermal power that can be dissipated.

For more information about the Device Bay Power Capabilities registers, see section 6.

7.6 The OS and USB-Based DBC Implementations

This section

- Identifies all the electrical, mechanical, and DBC features required by the OS in a USB-based DBC implementation.
- Shows a wiring diagram for an example USB-based DBC implementation (see Figure 7-2). This is an example only; there are many possible wiring diagrams.
- Shows how an OS might model the example hardware wiring diagram shown in Figure 7-2 with a set of loaded device drivers (see Figure 7-3). Note that there are many other ways to model the example hardware wiring diagram with loaded device drivers.

7.6.1 Example USB-Based DBC Hardware Implementation

An example USB-based DBC implementation is shown in Figure 7-2. Note that in this example, there are three downstream ports on the USB Hub Controller. Port number 1 is shown wired to the DBC, port 2 is wired to Device Bay 0, and port 3 is wired to Device Bay 1. These three downstream ports could be wired in any other possible combination. For example, port 1 could be wired to Device Bay 1, port 2 could be wired to the DBC, and port 3 could be wired to Device Bay 0. There is no reason the DBC must be wired to the lowest-numbered downstream port on the hub (when the DBC is enumerated, V_{id} is disabled in all the bays controlled by that DBC).

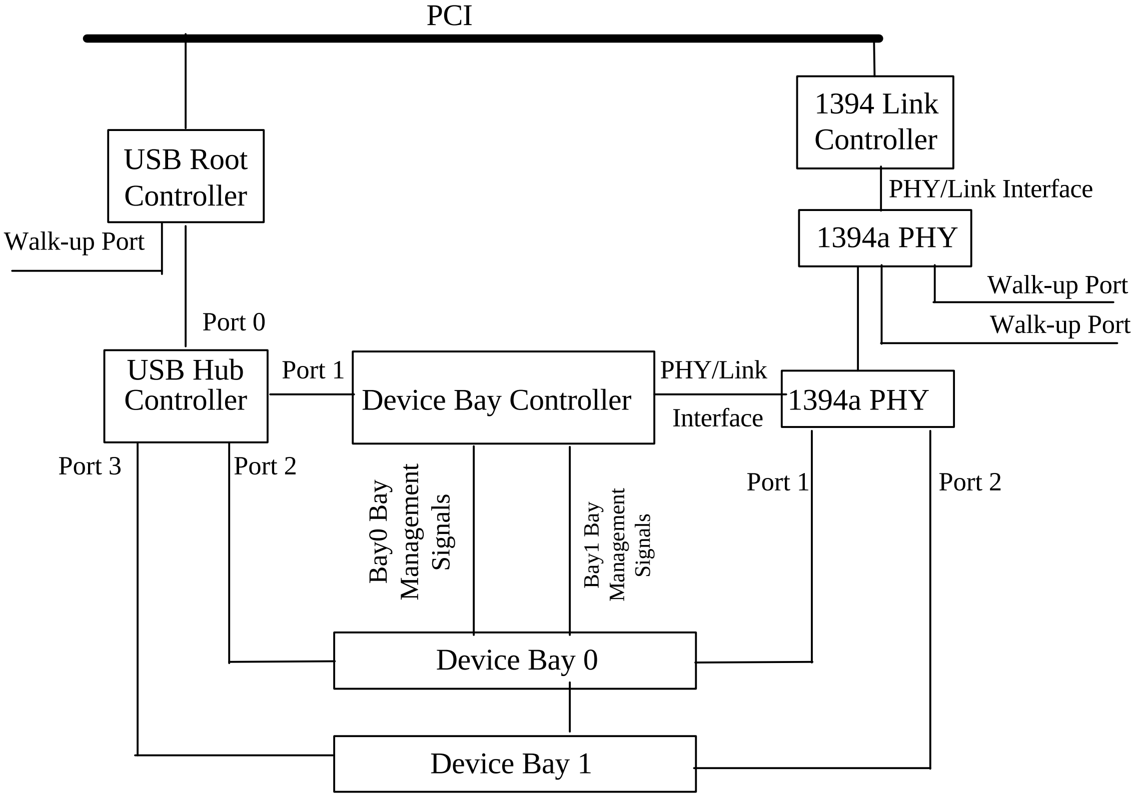

Figure 7-2. Example USB Implementation of Device Bay

7.6.1.1 OS Issue: Finding the 1394 Link Controller

The OS must be able to determine the 1394 link controller that is a parent of the DBC. A Plug-and-Play OS maintains a parent-child internal map of the devices on a platform and must know what device is the parent of the DBC.

Looking at the example hardware implementation shown in Figure 7-2, the 1394 link controller that is the parent of the DBC can be identified. However, an OS can model the hardware with loaded device drivers as shown in Figure 7-3. There is no association between the 1394 Host and the DBC.

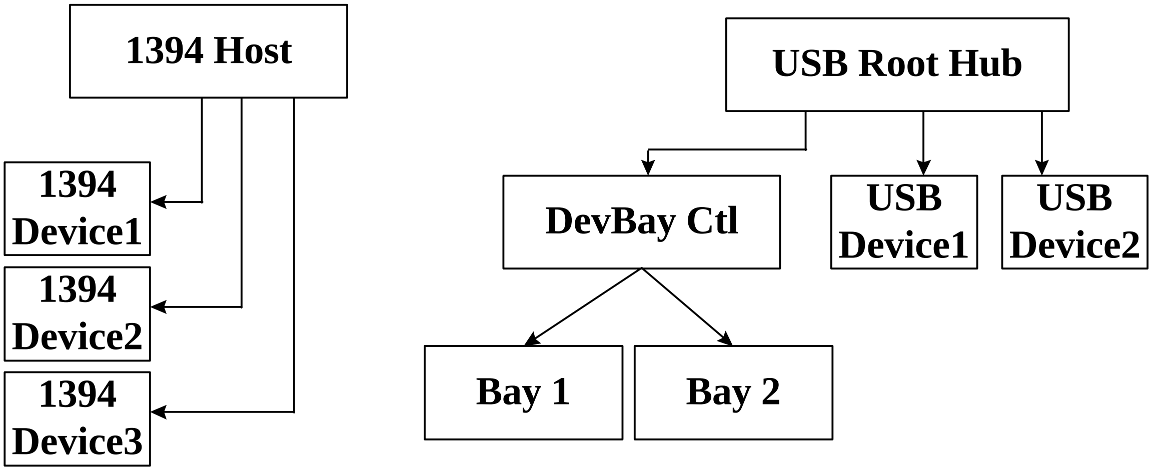

Figure 7-3. USB Implementation Device Driver Structure

A USB-based DBC provides a 64-bit CGUIDR register from which the OS can always read the GUID of the 1394 Link associated with the DBC. For more information about the CGUIDR register, see section 6. The 1394 bus enumerator finds a DBC while enumerating the 1394 bus. The *Unit_Spec_Id* and *Unit_SW_Version* entries in the 1394 ROM space of the DBC provide values the OS can use to identify this 1394 device as a DBC and the anchor point in the Device Bay topology.

The GUID of the DBC must be used to anchor the DBC, rather than a node ID, because node IDs can change on the 1394 bus with each bus reset. At each bus reset, as the 1394 bus is re-enumerated, the GUIDs present during the enumeration must be compared against the GUID of the DBC that was present before the bus reset. Once the matching GUID is found, the DBC is found, and the new node ID of the DBC is known. If no matching GUID is found, then the DBC and the associated Device Bay tree are gone from the bus.

As a double-check, when the USB bus is enumerated and the DBC is found on the USB bus, the OS can read the 1394 GUID of the DBC link from the USB bus. The OS can compare the GUID from the USB side with the GUID found during the 1394 enumeration. If the two GUIDs do not match, an enumeration error has occurred.

7.6.1.2 OS Issue: Finding the USB Hub the DBC is Wired To

In order to find the hub the DBC is wired to, the OS must count on the following architecture for a USB-based DBC: the DBC and all ports controlled by the DBC must be connected to the same USB hub.

7.6.1.3 OS Issue: Mapping Devices to Bays

How does the OS know which device is inserted into which bay? This is a problem for both USB and 1394 devices. Looking at the example USB hardware implementation in Figure 7-2, port 2 of the USB hub is wired to bay 0 and port 3 is wired to bay 1. However, looking at the way an OS can model the hardware (shown in Figure 7-3), the OS cannot derive this wiring information from its model. Similarly, in Figure 7-2, the port 1 of the 1394a PHY is wired to bay 0 and port 2 is wired to bay 1. However, this information is missing in the OS model shown in Figure 7-3.

For USB-based DBCs, the DBC register set provides BPMR*x*/USBPN, which always contains the number of the USB port connected to bay *x*, and BPMR*x*/1394PHYPN, which always contains the port number of the 1394 PHY connected to bay *x*.

7.6.1.4 USB Device Bay Controller Class Specification

The USB Device Bay Controller Class Specification specifies the information reported to the OS from the USB side of the USB-based DBC.

For a reference to the *USB Device Bay Controller Class Specification* see section 1.5.

7.7 OS Requirements for ACPI-Based DBCs

This section

- Identifies all the mechanical and DBC features required by the OS in an ACPI-based DBC implementation.
- Shows a wiring diagram for an example ACPI-based DBC implementation (see Figure 7-4). This is an example only; there are many possible wiring diagrams.
- Shows how an OS might model the example hardware wiring diagram shown in Figure 7-4 with a set of loaded device drivers (see Figure 7-5). Note that there are many other ways to model the example hardware wiring diagram with loaded device drivers.

7.7.1.1 ACPI Name Space Support for Multiple USB Root Hubs and 1394 HCIs

To implement a single Device bay subsystem utilizing more than one USB root hub and/or more than one 1394 Host Controller, an ACPI-based DBC must be used. An ACPI-compatible OS will search ACPI name space with explicitly-named paths to each Device Bay bay object, so there are no limits to number of USB hubs or 1394 Link Controllers.

7.7.2 Example ACPI-Based DBC Hardware Implementation

An example ACPI-based DBC implementation is shown in the following block diagram.

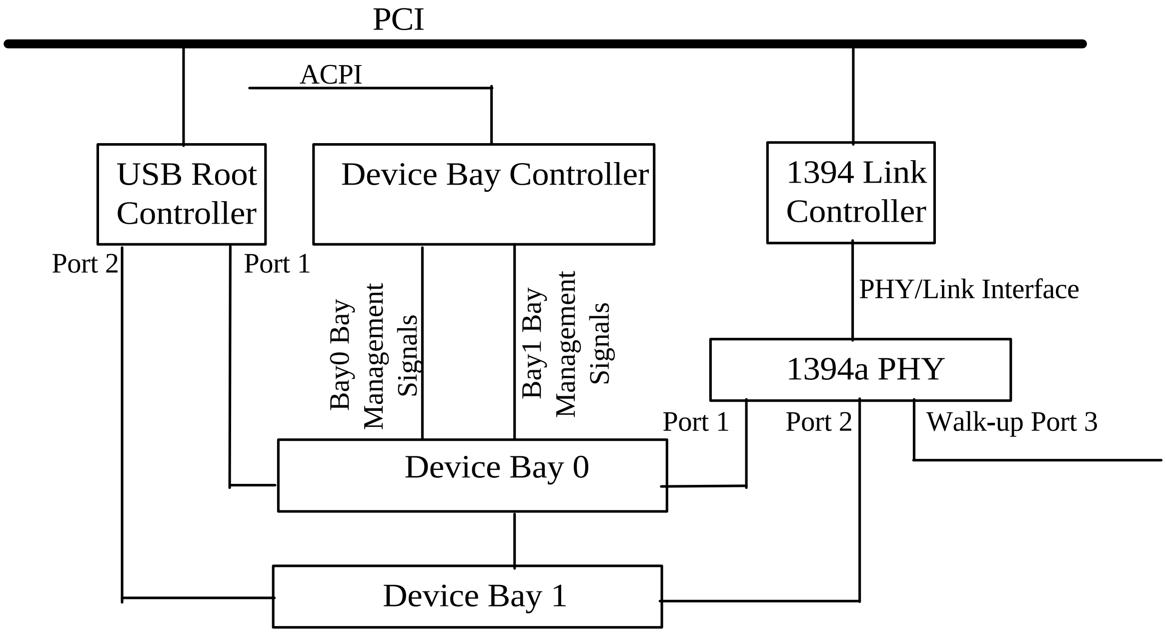

Figure 7-4. Example ACPI Implementation Wiring Diagram

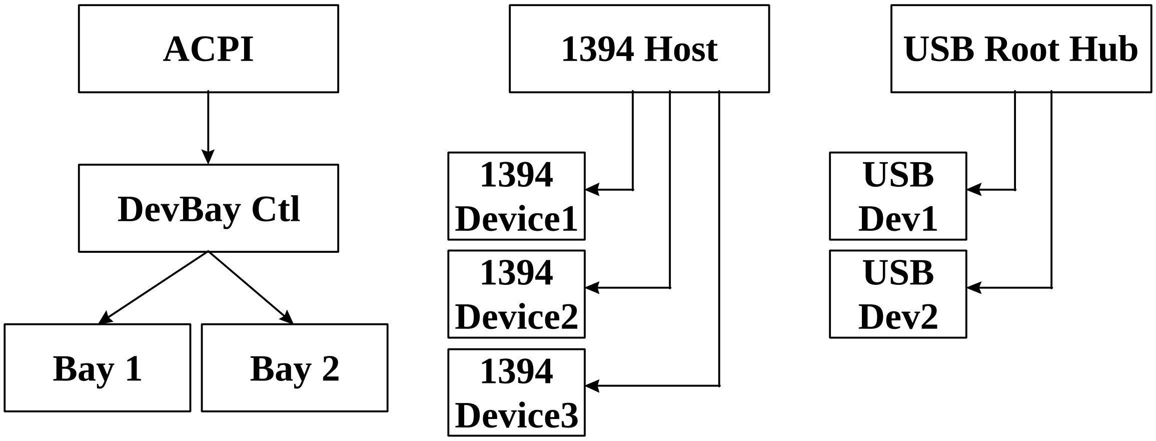

Figure 7-5. ACPI Implementation Device Driver Structure

7.7.2.1 OS Issue: Mapping Devices to Bays

How does the OS know which device is inserted into which bay? In Figure 7-5, for example, 1394 Device1 and USB Dev1 may be in the same device container in one bay, but 1394 Device1 is enumerated by the 1394 bus enumerator and USB Dev1 is enumerated by the USB bus enumerator. However, the gross structure of the diagram above shows three independent driver stacks with no relationship to each other, and that can cause problems for an OS that is trying to coordinate activities between device drivers.

For an ACPI implementation of the Device Bay Controller, the parent-child relationships can be modeled in the ACPI name space. Values in the ACPI object _ADR can be used to link device objects in the hierarchical ACPI name space with the information in the non-hierarchical driver information structure shown in Figure 7-5.

For more information about the use of the _ADR object in an ACPI name space that includes a DBC, see Annex C.

7.8 Advanced Topics

7.8.1 Booting from a Device in a Bay

Nothing in the Device Bay specification prohibits booting from a Device Bay device. For more information, see the 1394 specifications, the USB specification, BIOS vendor, and/or your OS vendor. Note: For explicit references to the 1394 and USB specifications, see section 1.5.

7.8.2 The OS and Compound Devices

The other parts of this software section have assumed that a device in a bay is either a USB device or a 1394 device. However, the device in a bay can be a compound device, that is, a device that contains both a USB device and a 1394 device. For example, a compound device might contain a 1394-based LAN device and a USB-based modem When the user requests removal of a device from a bay and the OS discovers that a compound device is in the bay and that one of the devices is active, what does the OS do?

- Through the DBC interface, the OS can determine that a compound device is in the bay by examining BSTRx/1394PRSN_STS and BSTRx/USBPRSN_STS.
- The reasonable thing for the OS to do is to warn the user, by displaying a UI, that an active device is about to be removed and give the user a chance to OK the removal.
- When a compound device is inserted into a bay and either the 1394 device or the USB device is “broken” (cannot be enabled by the OS), what does the OS do? The reasonable thing for the OS to

do is to warn the user, by displaying a UI, and keep the device in the bay, with all “non-broken” devices enabled.

7.9 Task Sequence Working Tables

This section presents a set of task sequence tables that walk through the sequence of tasks carried out by the user interaction with

- The physical bay interface (the bay, the bay indicator, and the optional removal request button)
- The bay mechanical features
- The DBC that controls the bay
- The OS
- And the user interaction with the UI displayed by the OS.

The purpose of these task sequence tables is to outline the steps that occur to accomplish device insertion, device removal, system power up, and system power down. The tables cover the cases in which the process goes as expected. Branches for error conditions are not shown and these tables are not to be used as implementation guidelines.

7.9.1 Device Insertion Task Sequence Tables

This section contains the device insertion task sequence table for an ACPI-based DBC. For the device insertion task sequence table for a USB-based DBC, please refer to the *Universal Serial Bus Device Bay Controller Specification*, which is available at http://www.usb.org.

7.9.1.1 ACPI-Based DBC Insertion Task Sequence Table

This section parses the tasks carried out by the Device Bay mechanical features, an ACPI-based DBC, and the OS to accomplish device insertion.

Coordinating the Bay	Mechanical Features	DBC	OS	Coordinating the UI
1-Device is inserted into the bay *x*.				
	2-A shelf provides rough alignment.			
	3-Device Bay connector fine-tunes the alignment.			
	4-Device seats into the connector.			
	5-Connector presence pin(s) asserted.			
		6- Sets 1394PRSN_STS or USBPRSN_STS bit in BSTRx (if compound device, sets both bits).		
		7-Sets DEVSTSCHG bit in BSTRx.		
		8- Sets BAY_ST bits in BSTRx to 001b.		
	9a- If present, the hardware bay status indicator is set to indicate Device Inserted.		9b- UI bay status indicator is set to indicate Device Inserted.	
		10-Generates an SCI		
			11-Receives the SCI.	
			12-Executes the appropriate event handler control method (for more information, see section 5.6 of the *ACPI Specification, Revision 1.0.*)	
			13-Event handler determines the bay location of the SCI; calls the control method that reads the DEVSTSCHG bit in each BSTRx register until it finds one that is set.	
			14-Event handler determines the cause of the SCI;	

Coordinating the Bay	Mechanical Features	DBC	OS	Coordinating the UI
			calls the control methods that read the 1394PRSN_STS bits and USBPRSN_STS bits in BSTRx and determines one or both are set, so device must have been inserted.	
			15-Event handler calls control method to engage the software-controlled interlock and physically lock the device into place (by writing a “1” to the LOCK_CTL bit in BCERx).	
		16-Engages the software-controlled interlock.		
	17-Software-controlled interlocks are engaged.			
			18-Event handler calls control method to clear the “sticky” device status change bit by writing a “1” to the DEVSTSCHG bit in BSTRx.	
			19-Control method determines that 1.5W is available to turn on the V_{id} power rail in bay x.	
			20-Control method enables V_{id} power to the device by writing a “1” to the PWR_CTL bit in BCERx.	
		21-Enables V_{id} power rails to bay x.		
	22-V_{id} power flows to the device in bay x			
			23-Device appears on native bus; native bus enumerates the device.	

Coordinating the Bay	Mechanical Features	DBC	OS	Coordinating the UI
			24-Identifies device.	
			25-Determines that adequate power exists and informs the device, over its native bus, to enable V_{op}.	
			26-Loads the device driver for the identified device.	
			27-Enables the device on its native bus.	
			28-OS invokes control method to request bay state change to Device Enabled (writes 010b to BAY_STREQ in BCERx).	
		29- Sets BAY_ST bits in BSTRx to 010b.		
	30a-If present, the hardware bay status indicator is set to indicate Device Enabled.		30b-If active, the UI bay status indicator is set to indicate Device Enabled.	
31-User sees the device is ready.				

7.9.2 Device Removal Request Task Sequence Tables

This section contains the following device insertion task sequence tables:

- A removal request button-initiated, ACPI-based DBC task sequence.
- A UI-initiated ACPI-based, DBC task sequence.

For the USB-based DBC device removal task sequence tables (both for removal request button initiated removal, as well UI-initiated removal), please refer to the *Universal Serial Bus Device Bay Controller Specification, which is available at* http://www.usb.org.

This section parses the tasks carried out by the Device Bay mechanical features, an ACPI-based DBC, and the OS to accomplish device removal after the user presses the removal request button at bay x.

Note that the removal request button is an option on a Device Bay. If there is no removal request button on a bay, the user must initiate a device removal request from UI displayed by the OS (for more information about this, see section 7.9.2.4).

Coordinating the Bay	Mechanical Features	DBC	OS	Coordinating the UI
1-User presses removal request button				
	2-Removal Request signal is asserted.			
		3-Bay state (BSTRx/BAY_ST) changes to 011b.		
	4(a)-If present, the hardware bay state indicator is set to indicate Device Removal Request.		4(b)-If hardware bay state indicator is not present, the UI bay state indicator is set to indicate Device Removal Request.	
		5-Sets REMREQ_STS bit in BSTRx.		
		6-Generates an SCI		
			7-Receives the SCI.	
			8-Executes the appropriate event handler control method.	
			9-Event handler determines the bay location of the SCI; calls the control method that reads the REMREQ_STS bit in BSTRx and it is set.	
			10-Event handler calls the control method that clears the REMREQ_STS bit in BSTRx.	
			11-If “appropriate,” place device in a logical “off” state using native bus driver (for example, notify apps, etc.).	
	12-Device stops using V_{op}.			
			13-If possible, unloads any appropriate	

Coordinating the Bay	Mechanical Features	DBC	OS	Coordinating the UI
			device drivers.	
			14-Executes control method that requests disabling V_{id} (writes a “0” to PWR_CTL bit in BCERx).	
		15-Disables the V_{id} power rail in bay x.		
	16-V_{id} is removed from the device.			
			17-Waits the time interval specified by the device.	
			18-Executes control method that unlocks the software-controlled interlock (writes “0” into the LOCK_CTL bit of BCERx).	
		19-Disables the software-controlled interlock.		
	20-Software-controlled interlocks are disabled.			
			21-Executes control method that determines if any bays have a security lock (reads SECLOCK bit in DBCCR). If bit not set, skips ahead to step 29.	
			22-Executes control method that determines if security lock is engaged (reads SL_STS bit in BSTRx). If bit not set, skip ahead to step 29.	
			23-Prompts user that before device can be removed, user must disengage	

Coordinating the Bay	Mechanical Features	DBC	OS	Coordinating the UI
			security lock.	
				24-User responds to prompt.
			25-Executes control method that determines if security lock is disengaged (reads SL_STS bit in BSTRx). If bit is set, go back to step 23.	
			26-Executes control method that requests bay state indicator change to Device Removal Allowed (writes 100b into the BAY_STREQ bits of BCERx).	
		27-Sets the BAY_ST bits in BSTRx to 100b.		
	28(a)-If present, the hardware bay state indicator is set to indicate Device Removal Allowed.		28(b)-If active, the UI bay state indicator is set to indicate Device Removal Allowed.	
29-User realizes it is safe to remove the device.				
	30-If present, eject mechanism ejects device for user			
31-User removes device.				
	32-Connector presence pin(s) deasserted.			
		33- Clears 1394PRSN_STS or USBPRSN_STS bit in BSTRx (if compound device, clears both bits).		
		34-Sets DEVSTSCHG bit in BSTRx.		
		35-Sets the BAY_ST bits in		

Coordinating the Bay	Mechanical Features	DBC	OS	Coordinating the UI
		the BSTRx to 000b. Sets the BAY_STREQ bits in the BCERx to 000b.		
		36-Generates an SCI.		
			37-Receives the SCI.	
			38-Executes the appropriate event handler control method.	
			39-Event handler determines bay location of the interrupt by reading the DEVSTSCHG bit in each BSTRx register until it finds one that is set.	
			40-Event handler determines the cause of the interrupt by determining both the 1394PRSN_STS and USBPRSN_STS bits are cleared, so device must have been removed.	
			41-Event handler calls control method to clear the “sticky” DEVSTSCHNG bit.	
	42(a)-If present, the hardware bay state indicator is set to indicate Bay Empty.		42(b)-.If active, the UI bay state indicator is set to indicate Device Removal Allowed.	
				43-User gets closure on the request.

7.9.2.1 UI-Initiated Removal Request, ACPI-Based DBC

This section parses the tasks carried out by the Device Bay mechanical features, an ACPI-based DBC, and the OS to accomplish device removal after the user initiates a device removal request from the UI displayed by the OS.

Coordinating the Bay	Mechanical Features	DBC	OS	Coordinating the UI
				1-User initiates a device removal request.
			2-Uses information returned by the UI to determine which bay the user wants to remove the device from (bay x), then executes control method that sets the bay to the Removal Requested state (writes 011b to BAY_STREQ in BCERx).	
		3-Sets BAY_ST bits in BSTRx to 011b.		
	4(a)-If present, the hardware bay state indicator is set to indicate Device Removal Request. (Or, should the hardware bay state indicator remain inactive, since this is a UI-based removal?)		4(b)-If hardware bay state indicator is not present, the UI bay state indicator is set to indicate Device Removal Request. (Or, should this say, “Regardless of whether hardware bay state indicator is present…”).	
5-User realizes system has begun process of removing device from bay x.				
			6-If “appropriate,” place device in a logical “off” state using native bus driver (for example, notify apps, etc.).	
	7-Device stops using V_{op}.			
			8-Unloads any appropriate device drivers.	
			9-Executes control method that disables V_{id} power in bay	

Coordinating the Bay	Mechanical Features	DBC	OS	Coordinating the UI
			x.	
		10-Disables V_{id} power rail in bay x.		
	11-V_{id} is removed from device.			
			12-Waits the time interval specified by the device.	
			13-Executes control method that requests disabling the software-controlled interlock.	
		14-Disables the software-controlled interlock.		
	15-Software-controlled interlocks are disabled.			
			16-Determines if any bay has a security lock; executes control method that reads SECLOCK bit in DBCCR. If bit not set, skips ahead to step 21.	
			17-Determines if security lock is engaged; executes control method that reads SL_STS bit in BSTRx. If bit not set, skip ahead to step 21.	
			18-Prompts user that before device can be removed, user must disengage security lock.	
				19-User responds to prompt.
			20-Determines if security lock is disengaged; executes control method that reads SL_STS bit	

Coordinating the Bay	Mechanical Features	DBC	OS	Coordinating the UI
			in BSTRx. If bit is set, go back to step 18.	
			21-Executes control method that changes the bay indicator state to Device Removal Allowed.	
		22-Sets the BAY_ST bits in BSTRx to 100b to indicate Device Removal Allowed.		
	23(a)-If present, the hardware bay state indicator is set to indicate Device Removal Allowed.		23(b)- If active, the UI bay state indicator is set to indicate Device Removal Allowed.	
24-User realizes device can be removed.				
				25-User realizes it is safe to remove the device.
	26-If present, eject mechanism ejects device for user			
27-User removes device.				
	28-Connector presence pin(s) deasserted.			
		29-Clears 1394PRSN_STS or USBPRSN_STS bit in BSTRx (if compound device, clears both bits).		
		30-Sets DEVSTSCHG bit in BSTRx.		
		31-Sets the BAY_ST bits in the BSTRx to 000b. Sets the BAY_STREQ bits in the BCERx to 000b.		
		32-Generates an SCI.		
			33-Receives the SCI	

Coordinating the Bay	Mechanical Features	DBC	OS	Coordinating the UI
			34-Executes the appropriate event handler control method.	
			35-Event handler determines bay location of the interrupt by executing control method that reads the DEVSTSCHG bit in each BSTRx register until it finds one that is set.	
			36-Event handler determines the cause of the interrupt by executing control method that determines both the 1394PRSN_STS and USBPRSN_STS bits are cleared, so device must have been removed.	
			37-Event handler calls control method that clears the “sticky” DEVSTSCHNG bit.	
	38(a)-If present, the hardware bay state indicator is set to indicate Bay Empty.		38(b)-If active, the UI bay state indicator is set to indicate Bay Empty.	
				39-User gets closure on the request.

7.9.3 System Power On Task Sequence Tables

As the system powers up, the DBC resets and the PWR_EN and LOCK_EN control signals are negated. Therefore, V_{id} will not be applied to any present devices and the software-controlled interlocks will not be engaged. (Device retention mechanisms or optional security mechanisms may be engaged.) After the OS has loaded, the OS will read the DBC status registers, (BSTRx), to determine if devices are present, and proceed through the appropriate previously defined device insertion sequence.

A platform provider may chose to support booting from a Device Bay device. In this case, the system BIOS must have the ability to find and communicate with the DBC, identify possible boot devices via their native bus, (1394 or USB), and load the OS from a Device Bay device. A possible sequence is described in the following section.

7.9.4 System Power On Task Sequence Tables

This section parses the tasks carried out by the Device Bay mechanical features, the DBC, the USB root hub, the system BIOS, and the OS when the system is powered up.

Coordinating the System and Bay	**Mechanical Features**	**DBC/USB Root Hub Controller**	**OS / BIOS**	**Coordinating the UI**
1-User initiates a system power-up.				
		2-LOCK_EN and PWR_EN for all bays are negated. Interrupt enables are negated.		
		3-Sets 1394PRSN_STS or USBPRSN bit in BSTRx (if compound device, sets both bits) for each bay with a device present		
		[NOTE: No interrupt is generated from devices already present during the power-up sequence.]		
			4-BIOS determines the number of bays in the system and the bay types from the DBC capabilities register, (DBCCR).	
			5-BIOS determines the total Device Bay sub-system power and thermal capabilities from the power capabilities register, (DBPC).	
			6-BIOS determines the maximum number of devices that can be supported. [Note: Possible power budgeting software agents are not likely active yet.]	
			7-BIOS determines if the system OS is supported via a Device Bay device. If not, once the OS is loaded, it will control the Device Bay sub-system. If yes, continue.	
			8-BIOS reads the 1394PRSN_STS and USBPRSN_STS bits of the bays; if one (or both) are set in any bays, a device(s) are	

Coordinating the System and Bay	**Mechanical Features**	**DBC/USB Root Hub Controller**	**OS / BIOS**	**Coordinating the UI**
			present.	
			9-BIOS sends an OUT TRANSACTION to the DBC via the USB root hub controller to physically lock the first present device into place. (This transaction writes a “1” to the LOCK_CTL bit in BCERx.)	
		10-Enables the software-controlled interlock.		
	11-Software-controlled interlocks are enabled.			
			12-BIOS sends an OUT TRANSACTION to the DBC via the USB root hub controller to enable V_{id} power to the device. (This transaction writes a “1” to the PWR_CTL bit in BCERx.)	
		13-Enables V_{id} power rails to bay x.		
	14-V_{id} power flows to the device in bay x			
			15-Device appears on native bus. BIOS identifies the device to determine if it can load the OS from this device type. If yes, skip to #23.	
			16-Sends an OUT TRANSACTION to the DBC via the USB root hub controller to disable V_{id} power to bay x.	
		17-Disables V_{id} power rail to bay x.		
	18-V_{id} is removed the device.			
			19-Sends an OUT TRANSACTION to the	

Coordinating the System and Bay	Mechanical Features	DBC/USB Root Hub Controller	OS / BIOS	Coordinating the UI
			DBC via the USB root hub controller to disable the software-controlled interlock.	
		20-Disables the software-controlled interlock.		
	21-The Software-controlled interlock is disabled.			
			22-Return to #9 and continue looking for a possible boot device.	
			23-BIOS sends an OUT TRANSACTION to the DBC via the USB root hub controller to request bay state change to Device Enabled (this transaction writes 010b to BAY_STREQ in BCERx).	
			24-BIOS indicates via the native bus that the device can enable V_{op}.	
			25-BIOS determine if the OS is present on the device. If yes, skip to #37.	
			26-BIOS indicates via the native bus that the device must disable V_{op}.	
			27-Sends an OUT TRANSACTION to the DBC via the USB root hub controller to turn off V_{id} power in bay x.	
		28-Disables V_{id} power rail in bay x.		
	29-V_{id} is removed from device.			
			30-Waits the time interval specified by the	

Coordinating the System and Bay	Mechanical Features	DBC/USB Root Hub Controller	OS / BIOS	Coordinating the UI
			device.	
			31-Sends an OUT TRANSACTION to the DBC via the USB root hub controller to disable the software-controlled interlock.	
		32-Disables the software-controlled interlock.		
	33-Software-controlled interlock is disabled.			
			34-Sends an OUT TRANSACTION to the DBC via the USB root hub controller to return the bay indicator state to Device is inserted but not ready.	
		35- Sets the BAY_ST bits in BSTRx to 001b.		
			36-Return to #9 and find the next device.	
			37-OS is booted and assumes control of the Device Bay sub-system	

7.9.5 System Power Down Task Sequence Tables

This section parses the tasks carried out by the Device Bay mechanical features, the DBC, the USB root hub, and the OS to power-down the system after the user initiates a shutdown request from the UI displayed by the OS.

Coordinating the Bay	Mechanical Features	DBC/USB Root Hub	OS	Coordinating the UI
				1-User initiates a system shutdown request.
			2-If "appropriate," place device in a logical "off" state using native bus driver (for example, notify apps, etc.).	
	3- Device stops using V_{op}.			
			4-Sends an OUT TRANSACTION to the DBC via the USB root hub controller to disable V_{id} power to all bays.	
		5-Disables V_{id} power rail to all bays.		
	6-V_{id} is removed from devices.			
			7-Waits the longest time interval specified by any device.	
			8-Sends an OUT TRANSACTION to the DBC via the USB root hub controller to disable the software-controlled interlock.	
		9-Disables the software-controlled interlock.		
	10-Software-controlled interlocks are disabled.			
			11-DBC capabilities registers is read to verify the presence of any security interlocks	
			12-If present, the state of device security	

Coordinating the Bay	Mechanical Features	DBC/USB Root Hub	OS	Coordinating the UI
			locks are determined from the DBC Bay Status Register.	
				13-If necessary, a UI indicating the bays that are locked is displayed.
			14-Sends an OUT TRANSACTION to the DBC via the USB root hub controller to change the bay state for each bay to Device Removal Allowed, (provided a security lock is not present and engaged).	
		15- Sets the BAY_ST bits in BSTRx to 100b to indicate Device Removal Allowed for each bay, as appropriate.		
16-User realizes devices can be removed.				
			17-Shutdown process completes	
				18-System power-down message is displayed.
19-User may remove power and unlocked devices.				

7.9.6 Suspend / Resume Interaction

Device insertion and removal requests may generate a system wake-up event.

When the system is in a sleeping state (ACPI states S1-S3), locked Device Bay devices must not be removed:

- All Device Bay software-controlled interlocks must retain their state set by the operating system. For example, locked devices remain locked, and unlocked devices remain unlocked. The OEM interlock override only functions in these systems states if the OEM interlock override is implemented as a wake-up event that cannot be disabled.
- The Device Bay Controller is required to latch any insertion or removal events, maintaining DEVSTSCHG, USBPRSN_STS, and 1394PRSN_STS bits while the system is asleep.
- Upon waking, the operating system will not automatically re-enumerate the bays; the DBC driver must signal the operating system if any insertions or removals have occurred based on the above bits.

When the system is in the S4 or S5 state (in other words, the system will go through POST when the system wakes up again), devices may be removed or inserted.

- Device Bay software-controlled interlocks must be disengaged by the operating system prior to entering this state.
- The DBC is not required to latch any insertion or removal events.

Upon waking (or booting), the OS must automatically re-enumerate bays.

Annex A - Legacy Drive Support

This annex describes how near-term legacy drive support is provided in the bays.

A.1 The DB32 Device

Near-term legacy drive support is provided with an adapter that has a circuit board and bridge circuitry. Such an adapter permits early applications of the following devices:

- A 3.5-inch hard disk drive for a disk drive device (for example, see Figure A-1).
- A portable CD-ROM to address CD-ROM applications.
- A high-capacity removable media device.

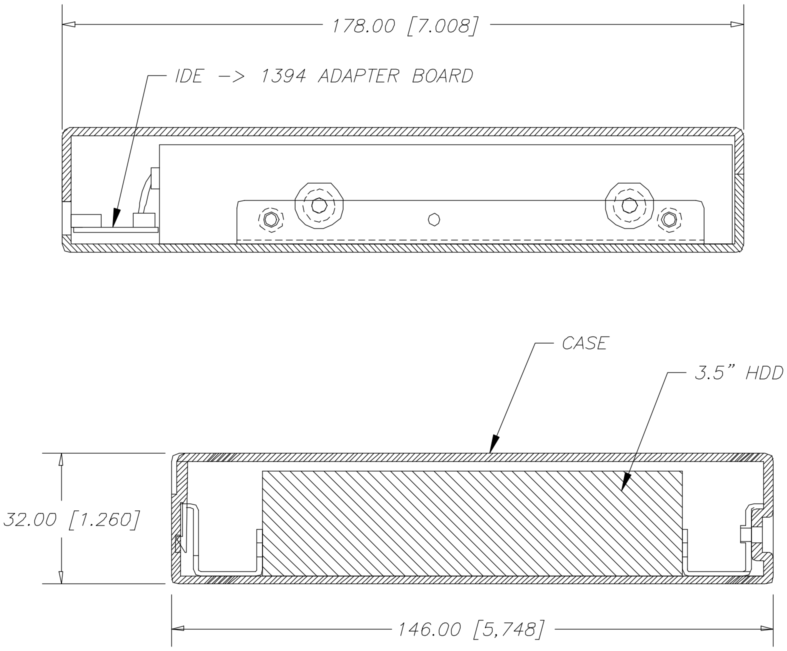

Figure A-1. Example DB32 Legacy Drive Adapter

A.2 The DB20 and DB13 Devices

Due to form factor volume constraints, it is anticipated that all drives and components fitting into the DB20 and DB13 form factors will migrate to either 1394 or USB buses without using the legacy migration path.

Annex B - Bay Status Indicator

This Annex describes a recommended way to implement a bay status indicator with an LED.
A color indicator can provide user feedback during various bay-related events. Such an indicator could be implemented in software as a user interface or in hardware on or in close proximity to the bay. In either case, the association between multiple bays and multiple indicators must be clear to the user.

The bay states defined in section 6 are mapped to indicator states as shown in Table B-1.

Table B-1. Bay Status Indicator Mapping

Bay state	**Indicator state**	**Indicator color**
Empty	Dark/Off	N/A
Device inserted	Flashing[1]	Green
Device enabled	On	Green
Removal requested	Flashing[1]	Yellow
Device removal allowed	Dark/Off	N/A

[1] The flash rate must be 0.5 Hz ± 20 percent at a 50 percent duty cycle.

Two indicator colors make it easier for the user to recognize the difference between insertion and removal request events. The choice of yellow and green indicators (versus red and green) has a two-fold reasoning. First, red/green color-blind persons are able to differentiate between yellow and green. Second, red indicators are considered a sign of danger in many European countries.

There is no provision for a “critical error” indication. Such an indication is expected to be conveyed to the user through an operating system–provided or application-provided user interface.

A bay-mounted status indicator could be implemented either by using a two-color (yellow/green) LED or two discrete LEDs. Two-color LEDs are either two-pin devices (elements connected anti-parallel) or three-pin devices (separate anodes and a common cathode or separate cathodes and a common anode). Two DBC control signals, LED_OUTG*x* and LED_OUTY*x*, are needed for each bay utilizing an indicator. Active-high LED control signals with bi-polar drive capability (equal sink and source current) offer the system designer the option of using either a two-pin or three-pin common cathode two-color LED or two discrete LEDs. An example of this for two-color LEDs is shown in Figure B-1. Notice that current limiting resistors may be required, but are not shown.

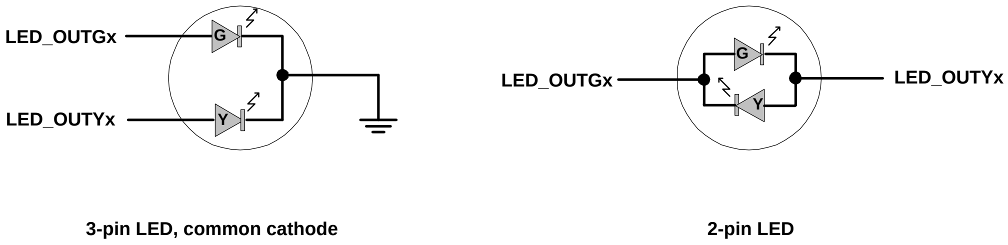

Figure B-1. DBC Control Signals and Two-Color LEDs

Annex C Device Bay and ACPI

C.1 ACPI Control Methods Needed for the Device Bay Controller

For ACPI-based DBCs, a limited number of ACPI control methods are needed to communicate between the ACPI driver and the Device Bay controller. The following are the ACPI control methods needed by BIOS developers in order to know which functions must be supported and what their parameters are. The use of these control methods is outlined in the Device Insertion and Device Removal task sequence tables in Section 7 of this specification.

- **Read/Write DBC register**

Name: **DBCC**
Arg0 – DWORD 0=Write, 1=Read
Arg1 – DWORD Zero based offset of register
Arg2 – DWORD Data to write if Arg0 equals 0

Returns: DWORD value, data read from the requested DBC register if Arg0 is 1.

- **Get GUID for 1394 Host Controller**

Name: **GUID**
Arg0 – DWORD 0 or 1 – 0 = return first 4 bytes of guid 1 = return second 4 bytes

Give an 8 byte GUID

```
| byte 0 | byte 1 | byte | byte 3 | byte 4 | byte 5 | byte 6 | byte 7 |
|----------------------------------|--------------------------------------|
         return if Arg0 is 0            return if Arg0 is 1
```

Returns: 4bytes of 1394 Host controller Guid

- **Get 1394 Bay/Port Mapping**

Name: **BPM3**
Arg0 – DWORD Index of Bay to get map for 1,2,3…..

Returns: DWORD value, 1394 port associated with the Bay Number in Arg0 (BPM3)

- **Get USB Bay/Port Mapping**

Name: **BPMU**
Arg0 – DWORD Index of Bay to get map for 1,2,3…..

Returns: TBD

- **Disable/Enable BIOS Device Bay Control**

Name: **BCTR**
Arg0 – DWORD 1 = enable bios control of bays 0 = disable BIOS control of bays

C.2 Example use of DBCC ACPI Control Method in ASL Code

The following is an implementation of the Read/Write DBC Register Control Method in ACPI Source Language.

```
//*********************************************************************
// Declare byte used in the DBCC control method
//*********************************************************************
       Name (TMP0, 0xff)
//********************************************************************
//  DBCC Method
//  Worker method that reads/writes DBC registers
//
//  Arg0 -  0 -> Write operation requested;
//          1 -> Read operation requested
//  Arg1 -  Zero-based offset of the desired DBC register (see
//          section 6.6.1 of the Device Bay specification)
//  Arg2 -  Data to write on a Write request
//
//  On a read request the value in the desired DBC register is returned
//*********************************************************************
       Method (DBCC,3) {             // Read or Write the DBC reg.
           Store(0x00, TMP0)         // Clear
           If (Lequal(Arg0, Zero)) { // If this is a write request
               Store(Arg2, DAT0)     // Write data byte
           }
           If (Lequal(Arg0, One) ) { // If this is a read request
           Store(DAT0, TMP0)         // Read byte from bus
       }
// Return value read from desired DBC register
       Return (TMP0)                 // Return data read
} // end of DBCC method
       .
       .
       .
```

C.3 BIOS Requirements for Handling Interrupts Generated by the Device Bay Controller

Note: It is important that the BIOS clear the interrupt condition on the Device Bay Controller after signaling a Notify event. That is, the BIOS must write a “1” to the DEVSTCHG or the REMREQ_STS bits in the BAY STATUS REGISTER after signalling a Notify event. The BIOS should not depend on the OS driver to clear the interrupt condition.

C.4 ACPI-Based DBC Power Reporting Requirements

An ACPI-based DBC must report its power requirements and capabilities in a set of ACPI objects in the ACPI namespace of the ACPI BIOS.

This section provides an example of one way to convert the DBPC register model into a set of ACPI objects in an example ACPI name space.

The following ACPI name space is populated with the Device objects that model a possible implementation of a two-bay subsystem on a host.

```
\SB
  .
  .
  .
   EC
       SMB
           DBC
               BAY0
               BAY1
```

The following ACPI name space is populated with Device objects and the other objects that enable an ACPI-compatible operating system to manage bay power. Of particular interest to this section of the

Device Bay specification is the DBPC object, which is placed under the DBC object in the ACPI name space.

```
\SB
  .
  .
  .
   EC
       SMB
           DBC
               _ADR                    //SMBus address
               DBPC                    //Device Bay Power Capabilities object
               BAY0
                   _PRx                //Method to control Vid power rail
                   MAP                 //\_SB.PCI0.UHC0.HUB0.HUB1.SLT0;\_SB.PCI0.1394.SLT0
               BAY1
                   _PRx
                   MAP                 //\_SB.PCI0.UHC0.HUB0.HUB1.SLT1;\_SB.PCI0.1394.SLT1
               BAY2
                   _PRx
                   MAP                 //\SB.PCI0,UHC1.HUB0.SLT2;\_SB.PCI0.1394.SLT2
```

The DBPC object can be implemented as a Package. For example, suppose the DBPC register model entry values for a DB20/DB13 form factor bay subsystem were as shown in Table C-1.

Table C-1. DBPC Register Model Entry Values Example (DB20/DB13 Form Factor)

Voltage (8 bits)	**Power Type (2 bits)**	**Power (22 bits, in DeciWatts)**
V_{id}	Continuous	15.00
V_{33}	Continuous	40.00
V_5	Continuous	40.00
V_{id}	Peak	15.00
V_{33}	Peak	80.00
V_5	Peak	80.00

The following ASL construct encodes these values in an object named DBPC.

```
Name (DBPC,            // Device Bay Power Capabilities
    Package ( ) {
        Package ( ) {
            0x21,
            0x00000f
        }   // end of Vid continuous entry, 15 deciwatts
        Package ( ) {
            0x21,
            0x000028
        }   // end of V33 continuous entry, 40 deciwatts
        Package ( ) {
            0x32,
            0x000028
        }   // end of V5 continuous entry, 40 deciwatts
        Package ( ) {
            0x21,
            0x40000f
        }   // end of Vid peak entry, 15 deciwatts
        Package ( ) {
            0x21,
            0x400050
        }   // end of V33 peak entry, 80 deciwatts
        Package ( ) {
            0x32,
            0x400050
        }   // end of v5 peak entry, 80 deciwatts
    //
    // Thermal Taggr value is same as electrical Taggr value
    //
    } // end package
} // end DBPC object
```

Annex D - DBC Simple Link Controller

This Annex calls out a simple link controller to go in the DBC when the DBC is connected to a USB bus. Section 6.8 of this specification lists the required CSRs and Configuration ROM needed for the 1394 section of the DBC.

This section details the minimal link controller behavior required for the DBC. Section D.8 lists additional features that may be added to a DBC link controller implementation if the design is not able to be minimal.

D.1 Transaction Capable

The minimal DBC link controller must be a transaction-capable 1394 node. The link must participate in asynchronous transactions and does not need to recognize isochronous transfers. Section 8.3.1.2 of the IEEE 1394-1995 specification lists the requirements for a transaction-capable node. The minimal link controller must not implement the SPLIT_TIMEOUT register that is called out in section 8.3.1.2 of the IEEE 1394-1995 specification. The minimal link controller must be able to respond to read requests using concatenated subactions and must respond to write requests using a unified response; no split transactions are necessary.

In addition to the registers listed in the IEEE 1394-1995 specification, the link controller also must have a Configuration ROM in the general ROM format and is described in section 6.8.2 of this specification.

D.2 1394 Packets

The PHY attached to the DBC link controller must pass 1394 packets to the minimal link controller in the DBC. The link controller must recognize only two types of packets from the 1394 bus and must generate only one type of packet to the 1394 bus.

D.2.1 Packet Reception

The DBC link controller must ignore a packet that contains a CRC error in the header_CRC. The header_CRC is the only CRC that the DBC link controller needs to check, since the minimal link controller will never respond to a packet with a data payload. The max_rec field of the DBC is set to 1h, meaning the maximum packet size received is 1 quadlet (4 bytes).

The minimal DBC link controller must properly recognize and process packets with the transaction codes listed in Table D-1.

Table D-1. DBC Link Controller Transaction Codes

Packet	Transaction code
Write request for data quadlet	0h
Read request for data quadlet	4h

The max_rec field of the DBC link controller is set to “1,” which means that the link will only receive packets that have a payload of 1 quadlet. Therefore, no block transactions are recognized by the minmal DBC link controller.

In addition to the two packet types, the DBC link controller must also properly handle broadcast packets. If the 10-bit bus_ID is between 000 and 3FEh, and the physical_ID is set to 3Fh, then this is a broadcast to the bus encoded in the bus_ID. If the bus_ID of the DBC matches the bus_ID of the packet, and if the tcode indicates that the packet is a write request for the data quadlet, then the DBC will accept the packet. No acknowledge (Ack) is returned in response to this type of broadcast.

If the 10-bit bus_ID is set to 3FFh and the physical_ID is set to 3Fh, then this is a broadcast to the local bus. If the tcode indicates that the packet is a write request for the data quadlet, then the DBC will accept the packet. No acknowledge is returned in response to this type of broadcast.

A packet addressed to the DBC link controller containing a tcode other than 0 or 4 will be acknowledged with an ack_type_error response. Ack_type_error indicates that a field in the request packet header was set to an unsupported or incorrect value, or that an invalid transaction was attempted

D.2.2 Packet Generation

A minimal implementation of a DBC link controller must be set to generate one type of packet and transaction code, which is shown in Table D-2.

Table D-2. DBC Link Controller Packet Generation Types

Packet	Transaction code
Read response for data quadlet	6h

The read response packet is generated as a concatenated subaction to a read request.

D.3 Retry Code

The minimal DBC link controller does not support retries; the BUSY_TIMEOUT register is not implemented. The rt field in a read response for data quadlet packet must always be retry_X.

D.4 Retries

Retries are not supported in a minimal DBC link controller; the BUSY_TIMEOUT register is not implemented. The DBC will not retry a response packet.

D.5 Response Codes

The DBC link controller must respond to requests using the response codes shown in Table D-3.

Table D-3. DBC Link Controller Response Codes

Response code	Response name	Comments
0h	resp_complete	The node has completed the command — no errors.
6h	resp_type_error	A field in the request packet header was set to an unsupported or incorrect value, or an invalid transaction was tried (such as a write to a read-only address).

D.6 Acknowledge Codes

The DBC link controller must respond to the packets it recognizes with an acknowledge packet. The ack_codes used by the DBC link controller are listed in Table D-4.

Table D-4. DBC Link Controller Acknowledgment Codes

Acknowledgment code	Acknowledgment name	Comments
1h	ack_complete	The node has successfully accepted the packet. This is sent when the DBC receives a non-broadcast write request for data quadlet (tcode 0) to a valid, writable DBC address.
2h	ack_pending	This is sent in response to a read request for data quadlet if the read targets a valid DBC address. To be followed by a concatenated response packet containing the data quadlet.
Eh	ack_type_error	A field in the request packet header was set to an unsupported or incorrect value, or an invalid transaction was tried. This is sent in response to all requests other than requests with tcodes 0 and 4.

D.7 Physical Interface

The first-generation DBC link controller must interface to a 400 Mbps 1394 PHY. Future generations of DBC link controllers will need to interface to next-generation 1394 PHYs at future 1394 bus speeds (for example, 1394a PHYs, and speeds of 800 Mbps or higher).

D.8 Additional Features

Previous sections detailed the minimum behavior required of the 1394 link controller section of a DBC. However, not all link controller implementations are able to meet the established minimum behavior. If a particular link controller implementation must add extra complexity, then it is the responsibility of the link controller implementers to account for the extra functionality required by their design, assuring that their design works properly on a 1394 bus.

The following are examples of additional features required in a non-minimal link controller. Many more possibilities exist. However, listing all possible cases where a DBC link controller would add extra complexity to its design is beyond the scope of this document.

- *Split transactions:* If a DBC link-controller implementation can be a requestor on the 1394 bus and uses split transactions, then the SPLIT_TIMEOUT register and extra logic to support split transactions must be included the link controller design.
- *Retries:* If the link controller implementation must issue retries, then the BUSY_TIMEOUT register must be implemented along with the logic necessary to handle the retries.

Revision History

Changes from 0.81 to 0.82

Chapter 5

- The DB13 and DB20 front retention features were made larger to accommodate a sturdier retention mechanism.
- The size of the DB13 and DB20 EMI/ESD pads was modified.

Chapter 6

- Update and clarify the hardware and software state transition tables, (Tables 6-6 and 6-7), and figures, (Figure 6-9 and 6-10).

Changes from 0.82 to 0.83

Chapter 4

- Blindmate features have been modified to achieve worst-case tolerance of +/-1.975mm in the vertical and +/-2.00mm in the horizontal. The blindmate feature, the "wings" or the "ribs," on the receptacle have been raised 1.00mm, and the base of the plug has been indented on the sides. To avoid stubbing and interference, several parameters have been redimensioned, including thinning the “T’s” on the plug, centering most dimensions with respect to their corresponding datums, tightening tolerances, and changing a few dimension values.
- Drawings in other chapters that also feature the connector (for example, several in Chapter 5) may not reflect the "look" of the new connector. In such cases, these drawings will be updated by Revision 0.85 of the Device Bay Specification.

Annex A

- Reflects the changes made to the connector in Chapter 4.

Changes from 0.83 to 0.85

Chapter 3

- Re-established all figure links, auto figure/table numbering
- Figure 3-1: Added voltages 3.3V and 5V to diagram
- 3.3.1.1: Removed description of Rcharge etc
- Figure 3-2 Replaced FETs with generic control blocks.
- Figure 3-3 Replaced FETs with generic control blocks.
- 3.3.1.2: Now require 10K Ohm (or less) resistor to ground on V_{id}; one too many Vids; were changed to Vops
- 3.3.1.6: Bullet 3 – Changed device to V_{id}
- 3.3.1.8: Added OnNow in the heading
- Device is required to support D2Chapter 4 and Annex A
- Minor text changes
- Figures 4-8, 4-9, 4-10, 4-12, 4-13: Corrected for the modified blindmate features and center-line-of-

datum descriptions

- Figure A-3: 2.29=>2.28mm for the center mounting hole from the first contact of the power segment
- Figure A-5: corrected a mirrored font for 0.85mm pad width
- Figure A-6: Datum -B- added
- Figure 4-6(a) and Fig.4-6(b): Note that these figures are intentionally placed on even and odd pages, respectively, so that when the specification is opened to these figures, they will be visible on opposing pages. This applies to Figure 4-7(a) and Figure 4-7(b) as well.
- Figure 4-2: Note this figure still does not show cavity for the blindmate feature. Not a technical concern.

Chapter 5

- Added the following lines to the beginning of each form factor dimensional section:
- All draft must be within dimensional tolerance.

- 5.2.2.4.: All round dimensions are described
- 5.2.2.4: Round dimensional criteria based on edge type.
- Dimensional tolerance changed for overall width for all form factors
- Changed width tolerance from +/-0.50 [.019] to +/-0.35 [0.014]
- EMI material specified.
- Added flatness callout to bottom surface of all form factors
- Moved EMI pads 2mm towards rear of device on DB32
- Added skew tolerance on connector placement for both devices and bays
- Clarified Device EMI/ESD opening statement.

Chapter 6

- 6.4.1: Wording change
- 6.4.2: Debounce period recommendation added.
- 6.4.3.1: Clarification of realm of system reponisbility for user-based issuance of removal request; also, debounce period recommendation.
- Table 6-1: DBC-Specific signals: Lock Enable and Power Enable updated.
- Table 6-4: Bay x Status Register Bits: Bit 3 set/clear status clarification.
- Table 6-5: Bay x Control and Enable Register Bits: Bit 3 removal of specific removal request push button.
- Table 6-6: Hardware Event Bay State Transition Table: Refer to Table for specific updates.
- Table 6-7: Clarification of meaning of “Bay Empty.”
- 6.6.9: OS requirement to check for engaged physical secutiry lock.

Changes from 0.85 to 0.90

Chapter 3

Major reorganization to increase clarification and make it easier to identify the requirements.

Changes include:

- Changed voltage tolerance on V5 and V3.3 for DB32, DB20 and DB13 from +-5% to +5-3% to make allowance for the devices required voltage control.
- Detailed power sequence requirements for 1394 and USB devices.
- Clarified Identification and enumeration requirement uses of V_{id}.
- Added entire section for Device Bay Power reporting requirements

- Added pointers to the 1394 and USB power reporting specifications.
- Added a ACPI Device Bay Power reporting description to Annex C
- Added an Example V_{id} and V_{op} power state transition for a USB device.
- Added an Example V_{id} and V_{op} power state transition for a compound device.
- Removed section 3.1.2.2: 1394 Electrical Device Bay requirements.

Chapter 4

- Changed the reference to “1394 specifications” instead of just “IEEE1394-1995” in 4.4.1.2., to also cover the 1394b’s 8b/10b signaling. [A reference to the 1394b specification (draft) is to be added in Section 1.5.]
- Pin configuration clarification in 4.4.2.
  - It should be noted that only the device plug connector, that is to mate directly with the receptacle connector in the bay, requires the mating sequence. Any other connector plug or receptacle in a system or a device is not required to have the mating sequence.
  - For some implementations the 1394 signals may be routed to the bay receptacle directly from the PHY on a PCB. For some other implementations, the 1394 signals may be routed to the bay receptacle after going through a few stages of connector pairs and cable assemblies. In this case, it is important to note that all the connectors within a system enclosure that use the Device Bay connector pin configuration will follow the “Device Bay Receptacle Pin Configuration” (Figure 4-5), regardless of the connector type being a plug or a receptacle. Similarly, there may be cable assemblies within a device, in which case all the connectors within a device that use the Device Bay connector pin configuration will follow the “Device Bay Plug Pin Configuration” (Figure 4-4), regardless of the connector type being a plug or a receptacle. This is to ensure the proper crossing of the 1394 TPA and TPB signals.
- In 4.5, the Annex-A reference for the reference design connectors was replaced with the Design Guide reference for the connector and cable assembly implementation examples.
- Removal of the R17 dimension and an addition of R17 explanation in 4.5.1.3.

  In Figure 4-7(b), R17, R13 and R22 are shown to describe the “shoulder” region’s height, length and thickness, respectively, of a receptacle. When a bay receptacle is fully mated to a device plug the blind-mate features of the two connectors bottom-out at the “platform” immediately above the R17. Because the plug connector is recessed within the device, during a fully-mated condition, the shoulder region (R13, R22 and part or all of R17) of a receptacle will be inside a device enclosure. Therefore, a receptacle’s shoulder region must conform to the dimensions R13 and R22 as specified in Figure 4-7(a). R17 is implementation dependent, but it must meet the following condition:

  - As specified in section 5, DB13 and DB20 devices require the plug to be recessed 0.50±0.25mm, but DB32 requires the plug to be recessed 2.00±0.50mm, inside the device enclosure. To ensure a fully-mated condition, R17 must be 0.75mm minimum for DB13 and DB20 applications, and 2.50mm minimum for DB32 applications.

  The following variations are possible for R17 implementations:

  - A receptacle’s R17 for DB13 and DB20 applications may be smaller than for DB32 applications.
  - R17 for a cable-terminated receptacle may be different from R17 for a PCB-mount receptacle.
  - R17 for a PCB-mount receptacle with mounting holes may be larger than PCB-mount receptacle without mounting holes (e.g., with fork-locks), to have enough clearance between the back surface of the device and the top of the screw-heads.
  - It should be noted that R17 for an implementation will have direct implication on the total contact pin length. Therefore other specification parameters, such as the LLCR (Table 4-4 in Section 4.6.3) must also be met for a receptacle implementation.

- Added a skew tolerance reference (section 5) for blind-mating in 4.5.1.4.

- Removed references to 1394 and USB specifications in 4.6.1, and clarify the signal electrical requirements in this section are for the mated connector pair only. [Note: The signal electrical requirements for the host and the device are to be added in Section 3, Buses.]

- In 4.2,
  - replaced the reference of Annex A to the *Device Bay Design Guide*,
  - rewrote the scope of this section.
- In 4.3,
  - changed the footnote on the DB connector support of 1394b data rates to be up to S3200 instead of S1600 [Comment: Tests on DB connector samples showed that this can be achieved],
  - removed "It is highly desirable that the connector also supports the 1394b data rate of 3.2Gbps

- In 4.4.1.2,
  - changed the reference to "1394 specifications" instead of just "IEEE1394-1995", to also cover the 1394b's 8b/10b signaling. [A reference to the 1394b specification (draft) is to be added in Section 1.5],
  - clarified that TPA and TPB pairs are from the 1394 PHY, and the pair-crossing is unlike in the 1394 specification,
  - added a note in the footnote on TPA and TPB pair crossing in the walkup cable.
- In 4.4.1.5, added more description for power supply pins.
- In 4.4.2,
  - clarified pin configuration,
  - added a footnote to clarify "reliability" in the contacts presence discussion,
  - added the option to load A26 contact for the bay receptacles.
- In 4.4.4.1, changed the grounding requirements for the device-side to require connecting all the ground pins on the device plug connector to the device ground.
- In 4.5, the Annex-A reference for the reference design connectors was replaced with the Design Guide reference for the connector and cable assembly implementation examples.
- In 4.5.1.2, Figure 4-6(b), changed the P5's positional tolerance from T3 to T4.
- In 4.5.1.3, removed R17 dimension and an added R17 explanation.
- In 4.5.1.4, added a skew tolerance reference (section 5) for blind-mating.
- In 4.6.1,
  - changed the crosstalk isolation for <500MHz cases.
  - changed the insertion losses for <2GHz cases.
  - removed references to 1394 and USB specifications,
  - clarified the signal electrical requirements in this section are for the mated connector pair only.
  - clarified the signal frequency values correlation with 1394b data rates.
- In 4.6.3 Table 4-4,
  - separated the current rating for Signal Segment and Power Segment,
  - referenced EIA standard only for the Signal Segment current rating procedure,
  - referenced section 4.6.9 for the Power Segment current rating procedure.
- In 4.6.8, reformatted Table 4-10 for a cleaner look.
- Added Section 4.6.9 (The basic information of this section used to be the Annex F of V0.85).
- General minor wording corrections (e.g., "blind mate" to "blind-mate") and minor format changes.

Chapter 5

- Revised all FF drawings
  * DB32 - Redim'ed Detail C, cutout around DB connector
  * Added datum callouts to all FF
  * Changed connector cutout dimensioning
  * Added additional notes as needed

- Ejector Landings were increased on DB32
- Skew tolerance for connector location added to form factor drawings

- Inside/Outside Round dimensions changed on DB32 (Section 5.2.2.5)

- Grip Zone is on Top & Bottom surfaces

- Bezel section for DB13/20 to be simplified

- EMI/ESD note added that surface is to 100% required. No holes or screws allowed.

- Air vent dimensions added in Device Section

- Maximum dimension for holes in Clamp Zones added as note in DB32 drawing

- Required surfaces section added for all 3 form factors

Section 5 - Known errata/areas for further definition

- Figure 5-6 is missing the right hand side updated ejector landing area
- Figure 5-37 is missing the right hand side updated ejector landing area
- Figure 5-39 is missing the right hand side updated ejector landing area
- Section 5.2.3.4 - Logo Definition. Logo definition is missing – Prior to defining it, it will be necessary to determine:
    - a) Height vs Width ratio
    - b) Smallest font size allowed
    - c) Font type to be used
    - d) Define the color of green that we want used.