Software API for PCIe Multicast Configuration

Application Note AN-710

Notes

By Alex Chang

Introduction

The IDT PCIe Gen2 16-port 64-lane switch (PES64H16G2) supports multicast within its active switch partitions (up to 16) with IDs 0 to 15. This feature is defined by the PCI-SIG Multicast ECN. The PES64H16G2 supports up to 64 multicast groups. When multicast is enabled, a multicast TLP is forwarded to all the functions within same multicast group. When a given port of the PES64H16G2 receives a multicast TLP, it can be configured to either block and discard the multicast TLP or forward the multicast TLP and transmit it onto the link associated with the switch port corresponding to the PCI-to-PCI bridge.

This application note describes the basics of PCIe TLP multicast and the software API that can be used to configure multicast-related registers to meet the requests from customers' applications.

Multicast

Associated with each PES64H16G2 switch port is a set of multicast registers defined by Multicast ECN. These registers can be used to:

- Enable/disable multicast function
- Set up multicast BAR
- Set up multicast group address range
- Configure multicast TLP blocking and receiving
- Set up multicast Overlay BAR

IDT multicast software API is provided to set up multicast related registers. The multicast BAR registers need to be programmed properly before enabling multicast.

Multicast TLPs

Multicast allows a single TLP sent from a source to be distributed to multiple recipients, resulting in a very high performance gain. Only Posted Request are supported in multicast:

- Posted Memory Write TLP
- Posted Address Routed Message TLP.

Multicast Address Window

Multicast TLPs need to be routed via an address. Therefore, associated with each enabled multicast group is a pre-defined equal sized multicast address window. The size can be decided via the software API. To enable TLP multicasting, all functions within a PCIe switch supporting Multicast Capability, including bridges and endpoints, must configure the following identically:

- Multicast Base Address
- Size of multicast address window
- Number of multicast group
- Enabling of multicast.

If the address in a Non-Posted Memory Request hits in a multicast window, no multicast activity occurs and the TLP is processed as a normal unicast packet.

Notes

Multicast Routing

When a TLP is received and determined to be handled as a multicast TLP, normal address routing rules do not apply. There is a dedicated bit associated with each multicast group in the Receive, Block All, and Block Untranslated registers. When a function is configured to receive a multicast TLP meant for multicast group# 2 and 5, the bit 2 and 5 need to be set in its multicast Receive register.

When a function receives a multicast TLP, it extracts the multicast group number from the address in the TLP, then checks the multicast Block All and Block Untranslated bits corresponding to the extracted multicast group number. If the Block bit corresponding to the extracted multicast group number is set, or the Block Untranslated bit corresponding to the extracted multicast group number is set and the TLP contains an Untranslated Address, the TLP is silently dropped. Otherwise, the TLP is forwarded to its downstream functions.

Multicast via Overlay

When an endpoint does not contain a Multicast Capability structure, the multicast overlay mechanism can be used to enable multicast and unicast TLP reception for the endpoint. This usage requires the following configurations:

- Connect the endpoint to one of downstream ports of PES64H16G2
- Allocate a single BAR of the endpoint
- Set up Overlay BAR of the downstream port so that the multicast address range, or a portion of it, is remapped onto the memory space range accepted by the endpoint's BAR
- Enable overlay mapping by setting overlay size no less than 6.

At the upstream port of the PES64H16G2, this mechanism can be used to overly a portion of the multicast address range onto a memory space range associated with host memory. This usage requires the following configurations:

- Connect a upstream ports of PES64H16G2 to the host
- Allocate a memory space range in host memory
- Set up Overlay BAR of the upstream port so that the multicast address range, or a portion of it, is remapped onto the memory space range in host memory
- Enable overlay mapping by setting overlay Size no less than 6.

For more details of PCIe Multicast, please refer to Multicast ECN published by PCI-SIG. For more details regarding IDT's multicast software APIs or other PCIe switch-related issues, please contact IDT at ssdhelp@idt.com.

Software API

A Linux-based software API package is implemented to make it easier to configure multicast-related registers within the PES64H16G2. The API package is developed in C language and can be easily ported to other operating systems. It supports Fedora Core 6, 7, 8, and 9 releases with both 32 and 64-bit kernels. The software API package is divided into three different layers as shown in Figure 1.

The top layer is the User Application that runs in the user space or the user device driver that runs in the kernel space. This layer is provided by the customers per their specific applications. The middle layer is the core Multicast APIs that is provided by IDT. It provides APIs to the user level program for multicast configuration. The bottom layer is I/O Access APIs that is system and OS dependent. This layer accesses PCIe configuration space registers via the LibPCI library to finalize multicast-related configuration requests from the User Application layer.

Notes

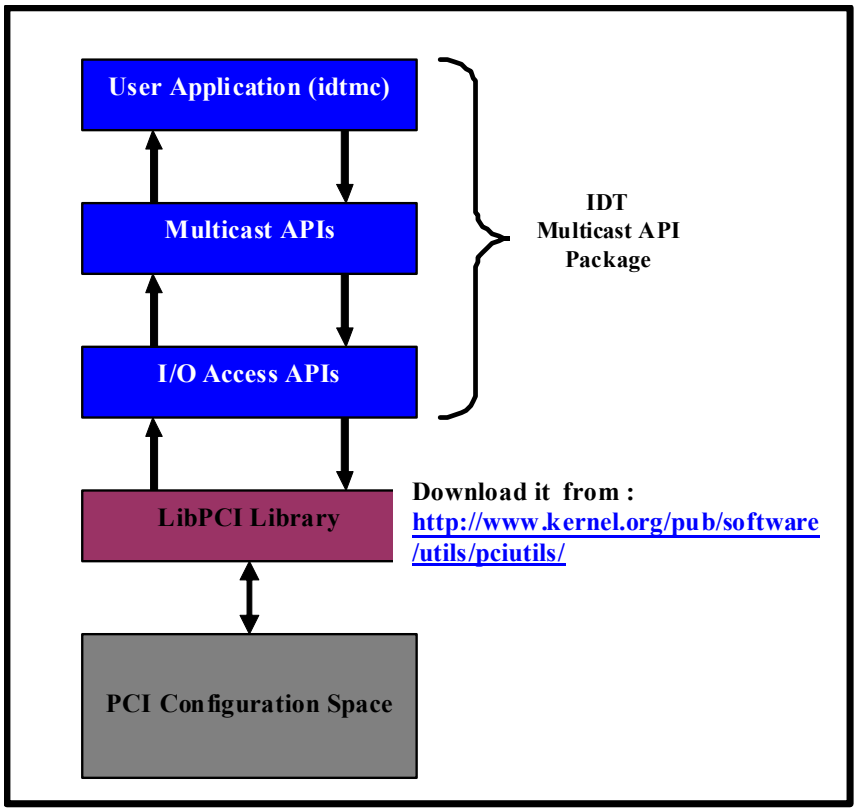

Figure 1 Multicast API Architecture

User Application

IDT provides a sample application called idtmc in the Multicast API package to demonstrate how to configure multicast. The usage of idtmc is detailed in Table 1.

The syntax of idtmc is: **idtmc bus# device# function# [-pneimbrauow] [value]**

Please note that the bus#, device#, function# are read as hexadecimal numbers and a valid value needs to be specified and immediately following each specified option.

Options	Description
none	List usages of idtmc.
bus# dev# fun#	List the current multicast status of the port associated with the specified bus#, dev# and fun#.
-p	Specify the target switch partition[0..15], read as decimal numbers. List current multicast status of the partition if no other options specified. If specified, the above bus#, dev# and fun# are associated with the upstream port of the partition.
-n	Specify number of multicast group[0..1f].
-e	Enable(!0)/Disable(0) multicast of the specified switch partition.
-i	Specify value of Index Position of a multicast address window. Valid values are [12..63], read as decimal numbers.
-m	Specify multicast BAR address, valid values are [0..ffffffffffffffff].

Table 1 Use of idtsp (Page 1 of 2)

Notes

-b	Specify bit offset of multicast Receive, Block All or Block Untranslated registers. Valid values are [0..63], read as decimal numbers.
-r	Clear(0)/Set(!0) bit value of multicast Receive register.
-a	Clear(0)/Set(!0) bit value of multicast Block All register.
-u	Clear(0)/Set(!0) bit value of multicast Block Untranslated register.
-o	Specify multicast overlay size. To enable overlay process, valid values are [6..63] and read as decimal numbers. Otherwise, it is disabled.
-w	Specify multicast Overlay BAR address. Valid values are [0..ffffffffffffffff].
Examples	
idtmc 4 0 0 –p 0 -e 1 -i 12 -n 10 -m f8000000	
Enable multicast for partition#0 with 16 groups, Index Position as 18 and BAR address at 0xF8000000.	
idtmc 6 0 0 –b 0 -r 0	
Clear bit 0 of multicast Receive register of the port associated specified bit, device, and function numbers.	
idtmc 5 0 0 –p 0 -b 2 -a 1	
Set bit 2 of multicast Block All register of all ports associated with Partition# 0.	
idtmc 4 0 0 –o 10 -w feee0000	
Enable multicast overlay processing by specifying Overlay BAR address at 0xfeee0000 and overlay size as 16.	

Table 1 Use of idtsp (Page 2 of 2)

Multicast APIs

Multicast APIs are provided to satisfy requests from the User Application layer. Multicast and I/O Access API share the following return codes, detailed in Table 2, when return type is an unsigned integer:

Value	Code	Meaning
-1	RC_NO_DEVICE	No device found associated with the BDF.
0	RC_SUCCESS	Function completes successfully.
1	RC_INV_PARA	Invalid parameter(s) found.
2	RC_NOT_SUPPORT	Request can not be fulfilled.
3	RC_REG_FAILURE	Fails on accessing register(s).
0xFF	RC_UNKNOWN_ERR	Unknown error.

Table 2 API Return Codes

Notes

The major APIs are listed below:

unsigned idtmcFormBDF(unsigned bus, unsigned dev, unsigned fun)

Description	Prepare an unsigned value containing Bus, Device and Function numbers.
Input(s)	
bus	Bus number of a PCI-to-PCI bridge.
dev	Device number of a PCI-to-PCI bridge.
fun	Function number of a PCI-to-PCI bridge.
Output(s)	
none	
Return	
unsigned	An unsigned value composed of Bus# (bit[15..8]), Device#(bit[7..3]) and Function#(bit[2..0]).

unsigned idtmcFindPortNum(unsigned bdf, unsigned* port_num)

Description	Determine the port number associated with specified Bus, Device, and Function numbers.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtmcFormBDF may be used to generate the value.
port_num	Pointer to an unsigned value.
Output(s)	
*port_num	Port number.
Return	
unsigned	Refer to Return Codes in Table 2.

Notes

unsigned idtmcFindPorts(unsigned bdf, unsigned part_num, unsigned* bdfs)

Description	Find out Bus, Device, Function numbers of all the ports associated with the specified partition. This function gets called when desired changes or configurations applied to all the ports of a given partition.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
part_num	Partition number [0..15].
bdfs	Pointer to a buffer provided by user application. This buffer needs to be at least 64 bytes in length.
Output(s)	
*bdfs	BDF number(s) of port(s) associated with the specified partition.
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcEnable(unsigned bdf, unsigned enable)

Description	Enable/Disable multicast functionality of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
enable	None-zero means enable, zero means disable.
Output(s)	
none	
Return	
unsigned	Refer to Return Codes in Table 2.

Notes

unsigned idtmcIsEnabled(unsigned bdf, unsigned* enable)

Description	Query if multicast functionality of the port associated with the specified BDF is enable or not.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
enable	Pointer to an unsigned value.
Output(s)	
*enable	One means enabled and zero means disabled.
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcSetBarAddr(unsigned bdf, unsigned bar_low, unsigned bar_high)

Description	Set up the BAR address for multicast functionality of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
bar_low	Lower 32 bits of the address.
bar_high	Higher 32 bits of the address.
Output(s)	
none	
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcGetBarAddr(unsigned bdf, unsigned* bar_low, unsigned* bar_high)

Description	Retrieve the BAR address for multicast functionality of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
bar_low	Pointer to an unsigned value.
bar_high	Pointer to an unsigned value.
Output(s)	
*bar_low	Lower 32-bit value of the address.

Notes

*bar_high	Higher 32 bit value of the address.
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcSetGroupNum(unsigned bdf, unsigned group_num)

Description	Set up the number of multicast groups of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
group_num	Number of multicast groups.
Output(s)	
none	
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcGetGroupNum(unsigned bdf, unsigned* group_num)

Description	Retrieve the number of multicast groups of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
group_num	Pointer to an unsigned value.
Output(s)	
*group_num	Number of multicast groups.
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcSetIndexPos(unsigned bdf, unsigned index_pos)

Description	Set up the Index Position of multicast of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
index_pos	Index Position of multicast.

Notes

Output(s)	
none	
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcGetIndexPos(unsigned bdf, unsigned* index_pos)

Description	Retrieve the Index Position of multicast of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
index_pos	Pointer to an unsigned value.
Output(s)	
*index_pos	Index Position of multicast.
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcSetOverlaySize(unsigned bdf, unsigned size)

Description	Set up the size of multicast overlay window of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
size	Size of multicast overlay window.
Output(s)	
none	
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcGetOverlaySize(unsigned bdf, unsigned* size)

Description	Retrieve the size of multicast overlay window of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.

Notes

size	Pointer to an unsigned value.
Output(s)	
*size	Size of multicast Overlay window.
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcSetOverlayAddr(unsigned bdf, unsigned ov_low, unsigned ov_high)

Description	Set up the address of multicast overlay window of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
ov_low	Lower 32 bits of the address.
ov_high	Higher 32 bits of the address.
Output(s)	
none	
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcGetOverlayAddr(unsigned bdf, unsigned* ov_low, unsigned* ov_high)

Description	Retrieve the address of multicast overlay window of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
ov_low	Pointer to an unsigned value.
ov_high	Pointer to an unsigned value.
Output(s)	
*ov_low	Lower 32-bit value of the address.
*ov_high	Higher 32 bit value of the address.
Return	
unsigned	Refer to Return Codes in Table 2.

Notes

unsigned idtmcSetReceiveRegs(unsigned bdf, unsigned reg_val, unsigned HiOrLow)

Description	Set up high/low values of multicast Receive registers of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
reg_val	32-bit Value.
HiOrLow	None-zero mean higher 32-bit value, zero means lower 32-bit value.
Output(s)	
none	
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcGetReceiveRegs(unsigned bdf, unsigned* reg_val, unsigned HiOrLow)

Description	Retrieve the high/low values of multicast Receive registers of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
reg_val	Pointer to an unsigned value.
HiOrLow	None-zero means higher 32-bit value and zero means lower 32-bit value.
Output(s)	
*reg_val	32-bit value.
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcSetBlkAllRegs(unsigned bdf, unsigned reg_val, unsigned HiOrLow)

Description	Set up high/low values of multicast Block All registers of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
reg_val	32-bit value.
HiOrLow	None-zero mean higher 32-bit value, zero means lower 32-bit value.
Output(s)	

Notes

none	
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcGetBlkAllRegs(unsigned bdf, unsigned* reg_val, unsigned HiOrLow)

Description	Retrieve the high/low values of multicast Block All registers of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
reg_val	Pointer to an unsigned value.
HiOrLow	None-zero means higher 32-bit value, zero means lower 32-bit value.
Output(s)	
*reg_val	32-bit value.
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcSetBlkUntRegs(unsigned bdf, unsigned reg_val, unsigned HiOrLow)

Description	Set up high/low values of multicast Block Untranslated registers of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
reg_val	32-bit value.
HiOrLow	None-zero mean higher 32-bit value, zero means lower 32-bit value.
Output(s)	
none	
Return	
unsigned	Refer to Return Codes in Table 2.

Notes

unsigned idtmcGetBlkUntRegs(unsigned bdf, unsigned* reg_val, unsigned HiOrLow)

Description	Retrieve the high/low values of multicast Block Untranslated registers of the port associated with the specified BDF.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
reg_val	Pointer to an unsigned value.
HiOrLow	None-zero means higher 32-bit value, zero means lower 32-bit value.
Output(s)	
*reg_val	32-bit value.
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcSetReceiveBits(unsigned bdf, unsigned group_index, unsigned set)

Description	Set/Clear a bit value of multicast Receive registers of the port associated with the specified BDF. The bit is associated with one of the multicast groups.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
group_index	Multicast group number [0..63].
set	Non-zero means set and zero mean clear.
Output(s)	
none	
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcGetReceiveBits(unsigned bdf, unsigned group_index, unsigned* bit)

Description	Retrieve a bit value of multicast Receive registers of the port associated with the specified BDF. The bit is associated with one of the multicast groups.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
group_index	Multicast group number [0..63].
bit	Pointer to an unsigned value.
Output(s)	

Notes

*bit	The retrieved bit value.
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcSetBlkAllBits(unsigned bdf, unsigned group_index, unsigned set)

Description	Set/Clear a bit value of multicast Block All registers of the port associated with the specified BDF. The bit is associated with one of the multicast groups.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
group_index	Multicast group number [0..63].
set	Non-zero means set and zero mean clear.
Output(s)	
none	
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcGetBlkAllBits(unsigned bdf, unsigned group_index, unsigned* bit)

Description	Retrieve a bit value of multicast Block All registers of the port associated with the specified BDF. The bit is associated with one of the multicast groups.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
group_index	Multicast group number [0..63].
bit	Pointer to an unsigned value.
Output(s)	
*bit	The retrieved bit value.
Return	
unsigned	Refer to Return Codes in Table 2.

Notes

unsigned idtmcSetBlkUntBits(unsigned bdf, unsigned group_index, unsigned set)

Description	Set/Clear a bit value of multicast Untranslated registers of the port associated with the specified BDF. The bit is associated with one of the multicast groups.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
group_index	Multicast group number [0..63].
set	Non-zero means set and zero mean clear.
Output(s)	
none	
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned idtmcGetBlkUntBits(unsigned bdf, unsigned group_index, unsigned* bit)

Description	Retrieve a bit value of multicast Block Untranslated registers of the port associated with the specified BDF. The bit is associated with one of the multicast groups.
Input(s)	
bdf	The Bus, Device, and Function numbers of any enabled PCI-to-PCI bridge in the PES64H16G2. API idtspFormBDF may be used to generate the value.
group_index	Multicast group number [0..63].
bit	Pointer to an unsigned value.
Output(s)	
*bit	The retrieved bit value.
Return	
unsigned	Refer to Return Codes in Table 2.

Notes

I/O Access APIs

I/O Access APIs are designed to access configuration space via LibPCI. They all share the same return codes in Table 2.

unsigned ioaPCIReadConfig(unsigned bdf, unsigned offset, unsigned* value)

Description	Retrieve the current value at specified offset in configuration space associated with the bus, device and function numbers.
Input(s)	
bdf	The Bus, Device, and Function numbers associated with a PCI-to-PCI bridge. API idtsp-FormBDF may be used to generate the value.
offset	Where to read from.
value	Pointer to an unsigned value.
Output(s)	
*value	Current read back value.
Return	
unsigned	Refer to Return Codes in Table 2.

unsigned ioaPCIWriteConfig(unsigned bdf, unsigned offset, unsigned value)

Description	Write the given value at the specified offset in the configuration space associated with the bus, device, and function numbers.
Input(s)	
bdf	The Bus, Device, and Function numbers associated with a PCI-to-PCI bridge. API idtsp-FormBDF may be used to generate the value.
offset	Where to write to.
value	Value to write.
Output(s)	
none	
Return	
unsigned	Refer to Return Codes in Table 2.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.