

# **Working Draft American National Standard**

# **Project T10/BSR INCITS 490**

Revision 7  
20 September 2013

## **Information technology - PCI Express<sup>®</sup> Queuing Interface (PQI)**

This is an internal working document of T10, a Technical Committee of Accredited Standards Committee INCITS (InterNational Committee for Information Technology Standards). As such this is not a completed standard and has not been approved. The contents may be modified by the T10 Technical Committee. The contents are actively being modified by T10. This document is made available for review and comment only.

Permission is granted to members of INCITS, its technical committees, and their associated task groups to reproduce this document for the purposes of INCITS standardization activities without further permission, provided this notice is included. All other rights are reserved. Any duplication of this document for commercial or for-profit use is strictly prohibited.

T10 Technical Editor:

le Wei Njoo  
PMC-Sierra  
1380 Bordeaux Drive  
Sunnyvale, CA 94089  
USA

Telephone: (408) 239-8273  
Facsimile: (408) 492-9192  
Email: leWei.Njoo@pmcs.com

**Reference number  
ISO/IEC 14776-171:20xx  
ANSI INCITS 490-20xx**

## Points of Contact

### InterNational Committee for Information Technology Standards (INCITS) T10 Technical Committee

#### T10 Chair

John B. Lohmeyer  
LSI Logic  
4420 Arrows West Drive  
Colorado Springs, CO 80907-3444  
USA

Telephone: (719) 533-7560  
Email: lohmeier@t10.org

T10 Web Site: <http://www.t10.org>

T10 E-mail reflector:

Server: majordomo@t10.org  
To subscribe send e-mail with "subscribe" in message body  
To unsubscribe send e-mail with "unsubscribe" in message body

#### T10 Vice-Chair

William Martin  
Samsung Semiconductor, Inc  
7213 Marblethorpe Drive  
Roseville, CA 95747  
USA

Telephone: (916) 765-6875  
Email: bill.martin@ssi.samsung.com

#### INCITS Secretariat

1101 K Street, NW  
Suite 610  
Washington, DC 20005  
USA

Telephone: 202-737-8888  
Web site: <http://www.incits.org>  
Email: incits@itic.org

#### Information Technology Industry Council

Web site: <http://www.itic.org>

#### Document Distribution

INCITS Online Store  
managed by Techstreet  
1327 Jones Drive  
Ann Arbor, MI 48105  
USA

Web site: <http://www.techstreet.com/incitsgate.tmpl>  
Telephone: (734) 302-7801 or (800) 699-9277

Global Engineering Documents, an IHS Company  
15 Inverness Way East  
Englewood, CO 80112-5704  
USA

Web site: <http://global.ihs.com>  
Telephone: (303) 397-7956 or (303) 792-2181 or (800) 854-7179

# PCI Express<sup>®</sup> Queuing Interface (PQI)

## Draft

Secretariat  
**Information Technology Industry Council**

Approved mm.dd.yy

American National Standards Institute, Inc.

## ABSTRACT

This standard defines a circular queue interface for transferring information between a PQI host and a PQI device over the PCI Express architecture, and defines a scatter gather list (SGL) format that is used to describe data buffers.

## Draft

American  
National  
Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer. Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that effort be made towards their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give interpretation on any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

**CAUTION NOTICE:** This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

**CAUTION:** The developers of this standard have requested that holders of patents that may be required for the implementation of the standard, disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard. As of the date of publication of this standard, following calls for the identification of patents that may be required for the implementation of the standard, no such claims have been made. No further patent search is conducted by the developer or the publisher in respect to any standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

Published by

**American National Standards Institute**  
**11 W. 42nd Street, New York, New York 10036**

Copyright © 2013 by Information Technology Industry Council (ITI). All rights reserved.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without prior written permission of ITI, 1101 K Street, NW, Suite 610, Washington, DC 20005.

Printed in the United States of America

## Contents

	Page
Contents.....	v
Tables .....	xi
Figures .....	xiii
FOREWORD (This foreword is not part of this standard) .....	xiv
INTRODUCTION .....	xix
 1 Scope .....	 1
 2 Normative references .....	 2
2.1 Normative references overview .....	2
2.2 References under development .....	2
2.3 Other references .....	2
 3 Definitions, symbols, abbreviations, and conventions .....	 3
3.1 Definitions .....	3
3.2 Symbols and abbreviations .....	8
3.3 Keywords .....	8
3.4 Editorial conventions .....	9
3.5 Numeric and character conventions .....	10
3.5.1 Numeric conventions .....	10
3.5.2 Units of measure .....	11
3.5.3 Byte encoded character strings conventions .....	11
3.6 State machine conventions .....	12
3.6.1 State machine conventions overview .....	12
3.6.2 Transitions .....	13
3.6.3 Messages, requests, indications, confirmations, responses, and event notifications .....	13
3.6.4 State machine counters, timers, and variables .....	13
3.6.5 State machine arguments .....	13
3.7 Bit and byte ordering .....	14
3.8 Notation for procedure calls .....	16
3.9 Notation for UML figures .....	16
3.9.1 Introduction .....	16
3.9.2 Class notation .....	18
3.9.3 Class association relationships notation .....	19
3.9.4 Class aggregation relationships notation .....	20
3.9.5 Class generalization relationships notation .....	21
3.9.6 Class dependency relationships notation .....	22
3.9.7 Object notation .....	22
 4 General concepts .....	 23
4.1 ASCII data field requirements .....	23
 5 Model .....	 24
5.1 General overview .....	24
5.2 PQI classes .....	26
5.2.1 PQI domain class .....	26
5.2.2 PQI host classes .....	26
5.2.2.1 PQI host classes overview .....	26
5.2.2.2 PQI Host class .....	28
5.2.2.3 PQI Host Management Application Client class .....	28
5.2.2.4 PQI Host Administrator OQ Information class .....	28
5.2.2.5 PQI Host Administrator IQ Information class .....	28
5.2.2.6 PQI Host Operational Application Client Interface class .....	28
5.2.2.7 PQI Host Operational OQ Information class .....	29
5.2.2.8 PQI Host Operational IQ Information class .....	29

5.2.3 PQI device classes .....	30
5.2.3.1 PQI device classes overview .....	30
5.2.3.2 PQI Device class .....	33
5.2.3.3 PD State Machine class .....	33
5.2.3.4 PQI Device Management Device Server class .....	33
5.2.3.5 PQI Device Operational Device Server Interface class .....	33
5.2.3.6 Administrator OQ CI class .....	33
5.2.3.7 Administrator IQ PI class .....	33
5.2.3.8 Operational OQ CI class .....	33
5.2.3.9 Operational IQ PI class .....	34
5.2.3.10 PQI Device Administrator OQ Information class .....	34
5.2.3.11 PQI Device Operational OQ Information class .....	34
5.2.3.12 PQI Device Administrator IQ Information class .....	34
5.2.3.13 PQI Device Operational IQ Information class .....	35
5.2.4 PQI service delivery subsystem classes .....	35
5.2.4.1 PQI service delivery subsystem classes overview .....	35
5.2.4.2 PQI Service Delivery Subsystem class .....	36
5.2.4.3 PCI Express Fabric class .....	36
5.2.4.4 PQI Queue Structure class .....	36
5.2.5 Circular queue classes .....	37
5.2.5.1 Circular queue classes overview .....	37
5.2.5.2 Circular Queue class .....	38
5.2.5.2.1 Circular Queue class overview .....	38
5.2.5.2.2 Element Array Address attribute .....	38
5.2.5.2.3 PI Address attribute .....	38
5.2.5.2.4 CI Address attribute .....	38
5.2.5.3 Element Array class .....	38
5.2.5.3.1 Element Array class overview .....	38
5.2.5.3.2 Number Of Elements attribute .....	38
5.2.5.3.3 Element Length attribute .....	39
5.2.5.4 Element class .....	39
5.2.5.5 PI class .....	39
5.2.5.5.1 PI class overview .....	39
5.2.5.5.2 Index attribute .....	39
5.2.5.6 CI class .....	39
5.2.5.6.1 CI class overview .....	39
5.2.5.6.2 Index attribute .....	39
5.2.5.7 IQ class .....	39
5.2.5.7.1 IQ functional overview .....	39
5.2.5.7.2 Example of IQ object locations that are not separated .....	41
5.2.5.7.3 Example of IQ object locations that are separated .....	42
5.2.5.8 Administrator IQ class .....	43
5.2.5.8.1 Administrator IQ class overview .....	43
5.2.5.8.2 Element Array Address attribute .....	43
5.2.5.8.3 Element Length attribute .....	43
5.2.5.8.4 Number Of Elements attribute .....	43
5.2.5.8.5 PI Address attribute .....	43
5.2.5.8.6 CI Address attribute .....	43
5.2.5.9 Operational IQ class .....	43
5.2.5.9.1 Operational IQ class overview .....	43
5.2.5.9.2 IQ ID attribute .....	43
5.2.5.9.3 Element Array Address attribute .....	44
5.2.5.9.4 Element Length attribute .....	44
5.2.5.9.5 Number Of Elements attribute .....	44
5.2.5.9.6 PI Address attribute .....	44
5.2.5.9.7 CI Address attribute .....	44
5.2.5.9.8 Spanning Allowed attribute .....	44

5.2.5.9.9 Queue Protocol attribute .....	45
5.2.5.10 OQ class .....	45
5.2.5.10.1 OQ functional overview .....	45
5.2.5.10.2 Example of OQ object locations that are not separated .....	46
5.2.5.10.3 Example of OQ object locations that are separated .....	47
5.2.5.10.4 Interrupt Message Number attribute .....	48
5.2.5.11 Administrator OQ class .....	48
5.2.5.11.1 Administrator OQ class overview .....	48
5.2.5.11.2 Interrupt Message Number attribute .....	48
5.2.5.11.3 Element Array Address attribute .....	48
5.2.5.11.4 Element Length attribute .....	48
5.2.5.11.5 Number Of Elements attribute .....	48
5.2.5.11.6 PI Address attribute .....	48
5.2.5.11.7 CI Address attribute .....	48
5.2.5.12 Operational OQ class .....	49
5.2.5.12.1 Operational OQ class overview .....	49
5.2.5.12.2 OQ ID attribute .....	49
5.2.5.12.3 Element Array Address attribute .....	49
5.2.5.12.4 Element Length attribute .....	49
5.2.5.12.5 Number Of Elements attribute .....	49
5.2.5.12.6 PI Address attribute .....	49
5.2.5.12.7 CI Address attribute .....	50
5.2.5.12.8 Minimum Coalescing Time attribute .....	50
5.2.5.12.9 Maximum Coalescing Time attribute .....	50
5.2.5.12.10 Coalescing Count attribute .....	50
5.2.5.12.11 Wait For Rearm attribute .....	50
5.2.5.12.12 Interrupt Message Number attribute .....	51
5.2.5.12.13 Spanning Allowed attribute .....	51
5.2.5.12.14 Queue Protocol attribute .....	51
5.3 Queuing model .....	52
5.3.1 Queuing model overview .....	52
5.3.2 Circular queue model .....	52
5.3.2.1 Circular queue functional overview .....	52
5.3.2.2 Relationship of IU to elements .....	54
5.3.2.3 Queue producer .....	54
5.3.2.4 Queue consumer .....	55
5.3.2.5 Enqueue operation .....	55
5.3.2.5.1 Enqueue To IQ operation overview .....	55
5.3.2.5.2 Enqueue To OQ operation overview .....	56
5.3.2.5.3 PQI host enqueueing IUs to an IQ .....	56
5.3.2.5.4 PQI device enqueueing IUs to an OQ .....	57
5.3.2.6 Dequeue operation .....	59
5.3.2.6.1 Dequeue From IQ operation overview .....	59
5.3.2.6.2 Dequeue From OQ operation overview .....	59
5.3.2.6.3 PQI host dequeuing IUs from an OQ .....	59
5.3.2.6.4 PQI device dequeuing IUs from an IQ .....	61
5.3.2.7 IU Available notification .....	62
5.3.2.7.1 IU Available On IQ operation overview .....	62
5.3.2.7.2 IU Available On OQ operation overview .....	62
5.3.2.7.3 IU Available On OQ notification .....	62
5.3.2.7.4 IU Available On IQ notification .....	62
5.3.3 Creating circular queues .....	63
5.3.3.1 Creating circular queues overview .....	63
5.3.3.2 Creating the administrator queue pair .....	63
5.3.3.3 Creating operational queues .....	66
5.3.4 Deleting circular queues .....	67
5.3.4.1 Deleting circular queues overview .....	67

5.3.4.2 Deleting the administrator queue pair .....	67
5.3.4.3 Deleting operational queues .....	68
5.3.5 IQ priority .....	69
5.4 OQ service notification methods .....	69
5.4.1 OQ service notification methods overview .....	69
5.4.2 Sending OQ service notifications in MSI-X mode .....	69
5.4.2.1 Sending OQ service notifications in MSI-X mode overview .....	69
5.4.2.2 Sending OQ service notifications for an administrator OQ .....	69
5.4.2.3 Sending OQ service notifications for an operational OQ .....	69
5.4.3 Sending OQ service notifications in legacy INTx mode .....	72
5.4.4 Sending OQ service notifications in polled mode .....	73
5.4.5 Servicing OQ service notifications in MSI-X mode .....	73
5.4.6 Servicing OQ service notifications in legacy INTx mode .....	73
5.4.7 Servicing OQ service notifications in polled mode .....	73
5.5 PD (PQI device) state machine .....	74
5.5.1 PD (PQI device) state machine overview .....	74
5.5.2 PD0:Power_On_And_Reset state .....	74
5.5.2.1 PD0:Power_On_And_Reset state description .....	74
5.5.2.2 Transition PD0:Power_On_And_Reset to PD1:PQI_Status_Available .....	75
5.5.3 PD1:PQI_Status_Available state .....	75
5.5.3.1 PD1:PQI_Status_Available state description .....	75
5.5.3.2 Transition PD1:PQI_Status_Available to PD0:Power_On_And_Reset .....	75
5.5.3.3 Transition PD1:PQI_Status_Available to PD2:All_Registers_Ready .....	75
5.5.3.4 Transition PD1:PQI_Status_Available to PD4>Error .....	75
5.5.4 PD2:All_Registers_Ready state .....	75
5.5.4.1 PD2:All_Registers_Ready state description .....	75
5.5.4.2 Transition PD2:All_Registers_Ready to PD0:Power_On_And_Reset .....	76
5.5.4.3 Transition PD2:All_Registers_Ready to PD1:PQI_Status_Available .....	76
5.5.4.4 Transition PD2:All_Registers_Ready to PD3:Administrator_Queue_Pair_Ready .....	76
5.5.4.5 Transition PD2:All_Registers_Ready to PD4>Error .....	76
5.5.5 PD3:Administrator_Queue_Pair_Ready state .....	76
5.5.5.1 PD3:Administrator_Queue_Pair_Ready state description .....	76
5.5.5.2 Transition PD3:Administrator_Queue_Pair_Ready to PD0:Power_On_And_Reset .....	76
5.5.5.3 Transition PD3:Administrator_Queue_Pair_Ready to PD1:PQI_Status_Available .....	76
5.5.5.4 Transition PD3:Administrator_Queue_Pair_Ready to PD2:All_Registers_Ready .....	77
5.5.5.5 Transition PD3:Administrator_Queue_Pair_Ready to PD4>Error .....	77
5.5.6 PD4>Error state .....	77
5.5.6.1 PD4>Error state description .....	77
5.5.6.2 Transition PD4>Error to PD0:Power_On_And_Reset .....	77
5.5.6.3 Transition PD4>Error to PD1:PQI_Status_Available .....	77
5.6 Register based error information .....	78
5.7 PQI reset .....	80
5.7.1 PQI reset overview .....	80
5.7.2 PQI soft reset .....	80
5.7.3 PQI firm reset .....	81
5.7.4 PQI hard reset .....	81
6 PCI Express requirements and PQI device registers .....	82
6.1 PCI Express requirements .....	82
6.2 PQI device memory space .....	83
6.2.1 PQI device memory space overview .....	83
6.2.2 PQI device standard registers .....	84
6.2.3 PQI device assigned registers .....	85
6.2.4 PQI Device Signature register .....	85
6.2.5 Administrator Queue Configuration Function register .....	86
6.2.6 PQI Device Capability register .....	87
6.2.7 Legacy INTx Interrupt Status register .....	88



6.2.8 Legacy INTx Interrupt Mask Set register .....	89
6.2.9 Legacy INTx Interrupt Mask Clear register .....	89
6.2.10 PQI Device Status register .....	90
6.2.11 Administrator IQ PI Offset register .....	91
6.2.12 Administrator OQ CI Offset register .....	91
6.2.13 Administrator IQ Element Array Address register .....	92
6.2.14 Administrator OQ Element Array Address register .....	92
6.2.15 Administrator IQ CI Address register .....	92
6.2.16 Administrator OQ PI Address register .....	93
6.2.17 Administrator Queue Parameter register .....	93
6.2.18 PQI Device Error register .....	94
6.2.19 PQI Device Error Details register .....	95
6.2.20 PQI Device Reset register .....	96
6.2.21 PQI Device Power Action register .....	97
7 Queuing layer .....	101
7.1 IQ CI, IQ PI, OQ CI, and OQ PI structures .....	101
7.1.1 IQ CI dword .....	101
7.1.2 IQ PI register .....	101
7.1.3 OQ CI register .....	101
7.1.4 OQ PI dword .....	102
8 SGL (scatter gather list) .....	103
8.1 SGL overview .....	103
8.2 Standard SGL segment and last standard SGL segment .....	103
8.3 SGL descriptors .....	104
8.3.1 SGL descriptors overview .....	104
8.3.2 Data Block descriptor .....	104
8.3.3 Bit Bucket descriptor .....	105
8.3.4 Standard SGL Segment descriptor .....	106
8.3.5 Last Standard SGL Segment descriptor .....	107
9 Common properties for all IU layers .....	108
9.1 Overview of common properties for all IU layers .....	108
9.2 IUs and elements .....	108
9.2.1 IUs and elements overview .....	108
9.2.2 IU contained within a single element .....	108
9.2.3 IU that spans multiple elements .....	108
9.3 Common IU header for all IU layers .....	109
10 Administrator IUs and administrator functions .....	111
10.1 IU definition .....	111
10.1.1 IU definition overview .....	111
10.1.2 Administrator IU header .....	112
10.1.3 NULL IU .....	114
10.1.4 GENERAL ADMIN REQUEST IU .....	114
10.1.5 GENERAL ADMIN RESPONSE IU .....	117
10.1.5.1 GENERAL ADMIN RESPONSE IU overview .....	117
10.1.5.2 Additional status descriptor for STATUS field set to DATA-IN BUFFER UNDERFLOW .....	120
10.1.5.3 Additional status descriptor for STATUS field set to INVALID FIELD IN REQUEST IU .....	120
10.1.5.4 Additional status descriptor for STATUS field set to INVALID FIELD IN DATA-OUT BUFFER .....	121
10.2 Administrator functions .....	122
10.2.1 Administrator functions overview .....	122
10.2.2 REPORT PQI DEVICE CAPABILITY function .....	123
10.2.2.1 REPORT PQI DEVICE CAPABILITY request .....	123
10.2.2.2 REPORT PQI DEVICE CAPABILITY response .....	124
10.2.2.3 REPORT PQI DEVICE CAPABILITY parameter data .....	125

10.2.3 REPORT MANUFACTURER INFORMATION function .....	129
10.2.3.1 REPORT MANUFACTURER INFORMATION request .....	129
10.2.3.2 REPORT MANUFACTURER INFORMATION response .....	130
10.2.3.3 REPORT MANUFACTURER INFORMATION parameter data .....	131
10.2.4 CREATE OPERATIONAL IQ function .....	132
10.2.4.1 CREATE OPERATIONAL IQ request .....	132
10.2.4.2 CREATE OPERATIONAL IQ response .....	135
10.2.5 CREATE OPERATIONAL OQ function .....	137
10.2.5.1 CREATE OPERATIONAL OQ request .....	137
10.2.5.2 CREATE OPERATIONAL OQ response .....	141
10.2.6 DELETE OPERATIONAL IQ function .....	142
10.2.6.1 DELETE OPERATIONAL IQ request .....	142
10.2.6.2 DELETE OPERATIONAL IQ response .....	143
10.2.7 DELETE OPERATIONAL OQ function .....	144
10.2.7.1 DELETE OPERATIONAL OQ request .....	144
10.2.7.2 DELETE OPERATIONAL OQ response .....	145
10.2.8 CHANGE OPERATIONAL IQ PROPERTIES function .....	146
10.2.8.1 CHANGE OPERATIONAL IQ PROPERTIES request .....	146
10.2.8.2 CHANGE OPERATIONAL IQ PROPERTIES response .....	147
10.2.9 CHANGE OPERATIONAL OQ PROPERTIES function .....	148
10.2.9.1 CHANGE OPERATIONAL OQ PROPERTIES request .....	148
10.2.9.2 CHANGE OPERATIONAL OQ PROPERTIES response .....	150
10.2.10 REPORT OPERATIONAL IQ LIST function .....	151
10.2.10.1 REPORT OPERATIONAL IQ LIST request .....	151
10.2.10.2 REPORT OPERATIONAL IQ LIST response .....	152
10.2.10.3 REPORT OPERATIONAL IQ LIST parameter data .....	153
10.2.11 REPORT OPERATIONAL OQ LIST function .....	155
10.2.11.1 REPORT OPERATIONAL OQ LIST request .....	155
10.2.11.2 REPORT OPERATIONAL OQ LIST response .....	156
10.2.11.3 REPORT OPERATIONAL OQ LIST parameter data .....	157
Annex A (normative) Alternative SGL segment .....	161
Annex B (informative) Operating system suggestions .....	163
B.1 Power actions .....	163
Annex C (informative) SGL examples.....	164
C.1 Data transfer from a source data stream to a destination data buffer .....	164
C.2 Data transfer from a source data buffer to a destination data stream .....	166
C.3 Memory to memory data transfer .....	167
Annex D (informative) Bibliography .....	168

## Tables

	Page
Table 1 — Numbering conventions .....	11
Table 2 — Comparison of decimal prefixes and binary prefixes .....	11
Table 3 — Example of a 32-bit big-endian field .....	14
Table 4 — Bit assignments in a 32-bit big-endian field .....	15
Table 5 — Example of a 32-bit little-endian field .....	15
Table 6 — Bit numbers for a 32-bit little-endian field .....	15
Table 7 — Class diagram constraints and notes notation .....	17
Table 8 — Class diagram multiplicity notation .....	17
Table 9 — Class diagram notation for classes .....	18
Table 10 — Class diagram notation for associations .....	19
Table 11 — Class diagram notation for aggregations .....	20
Table 12 — Class diagram notation for generalizations .....	21
Table 13 — Class diagram notation for dependencies .....	22
Table 14 — Notation for objects .....	22
Table 15 — PQI device registers to be written during administrator queue pair creation .....	65
Table 16 — OQ service notification methods .....	69
Table 17 — Interrupt generation conditions and events .....	70
Table 18 — Register based error information structure .....	78
Table 19 — PQI device standard registers from offset 000h to offset 0FFh .....	84
Table 20 — PQI Device Signature register .....	85
Table 21 — Administrator Queue Configuration Function register .....	86
Table 22 — FUNCTION AND STATUS CODE field for memory reads .....	86
Table 23 — FUNCTION AND STATUS CODE field for memory writes .....	87
Table 24 — PQI Device Capability register .....	88
Table 25 — Legacy INTx Interrupt Status register .....	88
Table 26 — Legacy INTx Interrupt Mask Set register .....	89
Table 27 — Legacy INTx Interrupt Mask Clear register .....	89
Table 28 — PQI Device Status register .....	90
Table 29 — PQI DEVICE STATE field .....	90
Table 30 — Administrator IQ PI Offset register .....	91
Table 31 — Administrator OQ CI Offset register .....	91
Table 32 — Administrator IQ Element Array Address register .....	92
Table 33 — Administrator OQ Element Array Address register .....	92
Table 34 — Administrator IQ CI Address register .....	93
Table 35 — Administrator OQ PI Address register .....	93
Table 36 — Administrator Queue Parameter register .....	94
Table 37 — PQI Device Error register .....	95
Table 38 — PQI Device Error Details register .....	95
Table 39 — PQI Device Reset register .....	96
Table 40 — RESET ACTION field for memory writes .....	96
Table 41 — RESET ACTION field for memory reads .....	96
Table 42 — RESET TYPE field .....	97
Table 43 — PQI Device Power Action register .....	98
Table 44 — POWER ACTION field for memory writes .....	98
Table 45 — POWER ACTION field for memory reads .....	98
Table 46 — SYSTEM POWER ACTION field .....	99
Table 47 — DEVICE POWER ACTION field .....	100
Table 48 — IQ CI dword .....	101
Table 49 — IQ PI register .....	101
Table 50 — OQ CI register .....	102
Table 51 — OQ PI dword .....	102
Table 52 — Standard SGL segment .....	103
Table 53 — SGL descriptor format .....	104
Table 54 — SGL DESCRIPTOR TYPE field .....	104
Table 55 — Data Block descriptor .....	105

Table 56 — Bit Bucket descriptor .....	106
Table 57 — Standard SGL Segment descriptor .....	106
Table 58 — Last Standard SGL Segment descriptor .....	107
Table 59 — Common IU header for all IU layers .....	109
Table 60 — PQI IUs (IU TYPE field) .....	111
Table 61 — Administrator IU header .....	112
Table 62 — Administrator IU header error handling .....	113
Table 63 — NULL IU .....	114
Table 64 — GENERAL ADMIN REQUEST IU .....	115
Table 65 — SGL descriptor type support requirements for administrator request IUs .....	116
Table 66 — STATUS field values for PCI Express errors that occur while accessing the Data-In Buffer or the Data-Out Buffer .....	117
Table 67 — GENERAL ADMIN RESPONSE IU .....	118
Table 68 — STATUS field and additional status descriptor .....	119
Table 69 — Additional status descriptor if the STATUS field is set to DATA-IN BUFFER UNDERFLOW .....	120
Table 70 — Additional status descriptor if the STATUS field is set to INVALID FIELD IN REQUEST IU .....	120
Table 71 — Additional status descriptor if the STATUS field is set to INVALID FIELD IN DATA-OUT BUFFER .....	121
Table 72 — Administrator functions (FUNCTION CODE field) .....	122
Table 73 — REPORT PQI DEVICE CAPABILITY request .....	123
Table 74 — REPORT PQI DEVICE CAPABILITY response .....	124
Table 75 — REPORT PQI DEVICE CAPABILITY parameter data (i.e., Data-In Buffer contents) .....	125
Table 76 — IU layer specific descriptor .....	128
Table 77 — REPORT MANUFACTURER INFORMATION request .....	129
Table 78 — REPORT MANUFACTURER INFORMATION response .....	130
Table 79 — REPORT MANUFACTURER INFORMATION parameter data (i.e., Data-In Buffer contents) .....	131
Table 80 — CREATE OPERATIONAL IQ request .....	132
Table 81 — OPERATIONAL QUEUE PROTOCOL field .....	134
Table 82 — CREATE OPERATIONAL IQ response .....	135
Table 83 — CREATE OPERATIONAL OQ request .....	137
Table 84 — CREATE OPERATIONAL OQ response .....	141
Table 85 — DELETE OPERATIONAL IQ request .....	142
Table 86 — DELETE OPERATIONAL IQ response .....	143
Table 87 — DELETE OPERATIONAL OQ request .....	144
Table 88 — DELETE OPERATIONAL OQ response .....	145
Table 89 — CHANGE OPERATIONAL IQ PROPERTIES request .....	146
Table 90 — CHANGE OPERATIONAL IQ PROPERTIES response .....	147
Table 91 — CHANGE OPERATIONAL OQ PROPERTIES request .....	148
Table 92 — CHANGE OPERATIONAL OQ PROPERTIES response .....	150
Table 93 — REPORT OPERATIONAL IQ LIST request .....	151
Table 94 — REPORT OPERATIONAL IQ LIST response .....	152
Table 95 — REPORT OPERATIONAL IQ LIST parameter data (i.e., Data-In Buffer contents) .....	153
Table 96 — Operational IQ property descriptor .....	153
Table 97 — REPORT OPERATIONAL OQ LIST request .....	155
Table 98 — REPORT OPERATIONAL OQ LIST response .....	156
Table 99 — REPORT OPERATIONAL OQ LIST parameter data (i.e., Data-In Buffer contents) .....	157
Table 100 — Operational OQ property descriptor .....	158
Table A.1 — Last Alternative SGL Segment descriptor .....	161
Table A.2 — Alternative SGL segment .....	162
Table A.3 — Alternative Data Block descriptor .....	162
Table B.1 — Windows PowerAction member to SYSTEM POWER ACTION field .....	163
Table B.2 — Windows DevicePowerState member to DEVICE POWER ACTION field .....	163

## Figures

	Page
Figure 0 — Layered view of the clauses in this standard .....	xx
Figure 1 — SCSI document relationships .....	1
Figure 2 — State machine conventions .....	12
Figure 3 — Example class association relationships .....	19
Figure 4 — Example class aggregation relationships .....	20
Figure 5 — Example class generalization relationships .....	21
Figure 6 — Example class dependency relationships .....	22
Figure 7 — PQI device, PQI host, PQI service delivery subsystem, IQs, and OQs .....	25
Figure 8 — PQI Domain class diagram .....	26
Figure 9 — PQI Host class diagram .....	27
Figure 10 — PQI Device class and PCI Express classes .....	30
Figure 11 — PQI Device class model .....	32
Figure 12 — PQI Service Delivery Subsystem class diagram .....	36
Figure 13 — Circular Queue classes .....	37
Figure 14 — Example of IQ object locations that are not separated .....	41
Figure 15 — Example where IQ object locations are separated .....	42
Figure 16 — Example of OQ object locations that are not separated .....	46
Figure 17 — Example of OQ object locations that are separated .....	47
Figure 18 — Circular queue .....	52
Figure 19 — Example of a full circular queue and an empty circular queue .....	53
Figure 20 — Example location of IQ CI and PQI host IQ PI local copy .....	57
Figure 21 — Example location of OQ CI and PQI device OQ PI local copy .....	58
Figure 22 — Example location of OQ PI and PQI host OQ CI local copy .....	60
Figure 23 — Example location of IQ PI and PQI device IQ CI local copy .....	61
Figure 24 — Interrupt coalescing example with the WAIT FOR REARM bit set to zero .....	71
Figure 25 — Interrupt coalescing example with the WAIT FOR REARM bit set to one .....	72
Figure 26 — Legacy INTx sources and masks .....	72
Figure 27 — PD (PQI device) state machine .....	74
Figure 28 — PQI device memory space .....	83
Figure 29 — Example of an IU that fits within a single element .....	108
Figure 30 — Example of an IU spanning across multiple elements .....	109
Figure C.1 —SGL example of a data transfer from a source data stream to a destination data buffer .....	165
Figure C.2 —SGL example of a data transfer from a source data buffer to a destination data stream .....	166
Figure C.3 —SGL example of a memory to memory data transfer .....	167

## FOREWORD (This foreword is not part of this standard)

This standard defines a queuing interface for transferring information units over PCI Express® architecture. This standard may be used in conjunction with the SCSI over PCI Express (SOP) standard.

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the INCITS Secretariat, International Committee for Information Technology Standards, Information Technology Industry Council, Suite 610, 1101 K Street, NW, Washington, DC 20005.

This standard was processed and approved for submittal to ANSI by the International Committee for Information Technology Standards (INCITS). Committee approval of the standard does not necessarily imply that all committee members voted for approval. At the time it approved this standard, INCITS had the following members:

INCITS Technical Committee T10 on SCSI Storage Interfaces, which developed and reviewed this standard, had the following members:

John B. Lohmeyer, Chair  
William Martin, Vice-Chair  
Ralph O. Weber, Secretary

<i>Organization Represented</i>	<i>Name of Representative</i>
Amphenol Interconnect .....	Gregory McSorley Paul Coddington (Alt) Zhineng Fan (Alt) Adrian Green (Alt) Simon Hui (Alt) Alex Persaud (Alt) Chansy Phommachanh (Alt) Michael Wingard (Alt) CN Wong (Alt)
Bellwether Electronic Corporation.....	Chris Schmolze Larry Hsu (Alt)
Brocade.....	David Peterson Scott Kipp (Alt)
Dell, Inc. ....	Kevin Marks Mark Bokhan (Alt) Janet Long (Alt) Bill Lynn (Alt) Ash McCarty (Alt) Daniel Oelke (Alt) Daniel Suman (Alt) Han Wang (Alt)
EMC Corporation .....	Gary Robinson David Black (Alt) Mickey Felton (Alt) Jason Pritchard (Alt) Marlon Ramroopsingh (Alt)
Emulex .....	Hossein Hashemi
FCI Electronics.....	Donald Harper Paul Rubens (Alt) Dave Sideck (Alt)
Foxconn Electronics.....	Fred Fons Gary Hsieh (Alt) Glenn Moore (Alt) Mike Shu (Alt)

Fujitsu America Inc.....	Kun Katsumata Jim DeCaires (Alt) Osamu Kimura (Alt) Mark Malcolm (Alt) Gene Owens (Alt) Sandy Wilson (Alt)
Fusion-io .....	Walt Hubis Ashish Batwara (Alt) Bob Wood (Alt)
Futurewei Technologies Inc .....	Serge Manning Xiaoyu Ge (Alt) Xiaoyan He (Alt) Alan Yoder (Alt) HengLiang Zhang (Alt)
Hewlett-Packard Company .....	Curtis Ballard Wayne Bellamy (Alt) Rob Elliott (Alt) Steven Fairchild (Alt) Joe Foster (Alt) Barry Olawsky (Alt) Jeff Wolford (Alt)
HGST .....	Dan Colegrove Frank Chu (Alt)
IBM Corporation .....	Kevin Butt Mark Andresen (Alt) Mike Osborne (Alt) Ted Vojnovich (Alt)
Integrated Device Technology (IDT) .....	Peter Onufryk Kwok Kong (Alt)
Intel Corporation .....	Pak-Lung Seto Richard Mellitz (Alt) Mark Seidel (Alt)
JDS Uniphase Corporation .....	David Freeman George Bullis (Alt) Dominic Coupal (Alt) Gerard Geary (Alt) Paul Gentieu (Alt) Geoff Hibbert (Alt) Yamini Shastry (Alt)
KnowledgeTek Inc.....	Dennis Moore Hugh Curley (Alt)
LSI Corporation.....	John Lohmeyer Patrick Bashford (Alt) Brad Besmer (Alt) Srikiran Dravida (Alt) Steve Johnson (Alt) Bernhard Laschinsky (Alt) Harvey Newman (Alt) George Penokie (Alt) Sumit Puri (Alt) Robert Sheffield (Alt) Richard Solomon (Alt) Ross Stenfort (Alt) Bill Voorhees (Alt)
Marvell Semiconductor Inc.....	David Geddes Paul Wassenberg (Alt)

Micron Technology Inc .....	John Geldman Jerry Barkley (Alt) Neal Galbo (Alt) Michael George (Alt) Alan Haffner (Alt) Randy Matthews (Alt) Niels Reimers (Alt) Victor Tsai (Alt) Bob Warren (Alt)
Microsoft Corporation.....	Calvin Chen Paul Luber (Alt) Bryan Matthew (Alt) Steve Olsson (Alt)
Molex Inc.....	Jay Neer Michael Rost (Alt) Darian Schulz (Alt)
NetApp .....	Frederick Knight Chris Fore (Alt) Jaimon George (Alt) Subhash Sankuratripati (Alt) Ali Yavari (Alt)
NVidia Corporation.....	Mark Overby Andrew Currid (Alt) Prajakta Gudadhe (Alt)
Oracle .....	Dennis Appleyard Jon Allen (Alt) Martin Faltese (Alt) Seth Goldberg (Alt) Dale LaFollette (Alt) Martin Petersen (Alt) Wan Shan (Alt) Phi Tran (Alt) Lee Wan-Hui (Alt)
PMC-Sierra .....	Tim Symons David Allen (Alt) Paul Borsetti (Alt) Graeme Boyd (Alt) Guillaume Fortin (Alt) Mathieu Gagnon (Alt) Steve Gerson (Alt) Bill Lye (Alt) Ie-Wei Njoo (Alt) Keith Shaw (Alt) Gregory Tabor (Alt) Neil Wanamaker (Alt) Rod Zavari (Alt)
Quantum Corporation .....	Paul Suhler Paul Stone (Alt) Rod Wideman (Alt)



Samsung Semiconductor Inc (SSI).....	William Martin Judy Brock (Alt) Edward Chang (Alt) HeeChang Cho (Alt) KeunSoo Jo (Alt) ByeungJun Lee (Alt) Jake Lee (Alt) Sung Lee (Alt) Bhavith M.P. (Alt) Truong Nguyen (Alt) Marc Noblitt (Alt) Dmitry Obukhov (Alt) Changhyun Park (Alt) Santosh Singh (Alt)
SanDisk IL Ltd.....	Avraham Shimor Dave Landsman (Alt) Yoni Shternhell (Alt)
Seagate Technology .....	Gerald Houlder Alvin Cox (Alt) Jim Hatfield (Alt) Judy Westby (Alt)
STEC Inc.....	David Brewer Jason Gao (Alt) Nadesan Narenthiran (Alt)
TE Connectivity.....	Dan Gorenc Mike Davis (Alt) Michael Fogg (Alt) Tom Grzysiewicz (Alt) John Hackman (Alt) Doron Lapidot (Alt) Joel Meyers (Alt) Andy Nowak (Alt) Eric Powell (Alt) Yasuo Sasaki (Alt) Robert Wertz (Alt)
Toshiba America Electronic Components Inc.....	Mike Fitzpatrick Don Jeanette (Alt) Yuji Katori (Alt) Tom McGoldrick (Alt) Scott Wright (Alt)
VMware Inc .....	Murali Rajagopal Lawrence Lamers (Alt) Neil H. MacLean (Alt) Leah Schoeb (Alt) Sandeep Uttamchandani (Alt) Christopher Reed (Alt)
Western Digital Corporation .....	Curtis Stevens Mark Evans (Alt) Stephen Finch (Alt) Robert Griswold (Alt) Michael Koffman (Alt) Dan Reno (Alt) Michael Rogers (Alt) Andy Tomlin (Alt) Ralph Weber (Alt)

Xyratex..... Alan Westbury  
Michael K. Connolly (Alt)  
Neil Edmunds (Alt)

## INTRODUCTION

The PCI Express<sup>®</sup> Queuing Interface (PQI) standard is divided into the following clauses:

Clause 1 (Scope) describes the relationship of this standard to the SCSI standards and PCI Express specifications.

Clause 2 (Normative references) provides references to other standards and specifications.

Clause 3 (Definitions, symbols, abbreviations, and conventions) defines terms and conventions used throughout this standard.

Clause 4 (General concepts) defines data field requirements used throughout this standard.

Clause 5 (Model) describes PQI classes, the queuing model, OQ service notification methods, the PD (PQI device) state machine, register based error information and PQI reset.

Clause 6 (PCI Express requirements and PQI device registers) describes PCI Express requirements and PQI device registers.

Clause 7 (Queuing layer) describes the IQ CI, IQ PI, OQ CI, and OQ PI structures.

Clause 8 (SGL (scatter gather list)) describes SGLs, SGL segments, and SGL descriptors.

Clause 9 (Common properties for all IU layers) describes common properties for all IU layers.

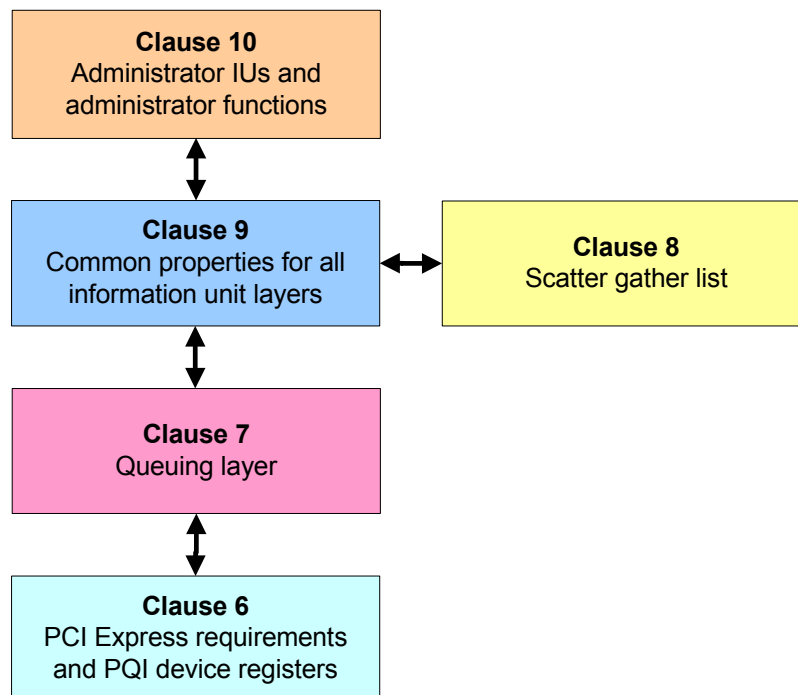
Clause 10 (Administrator IUs and administrator functions) describes the administrator IUs and administrator functions.

Annex A (Alternative SGL segment) describes the alternative SGL segment.

Annex B (Operating system suggestions) describes operating system suggestions for the power actions.

Annex C (SGL examples) describes SGL examples.

Figure 0 shows a layered view of the clauses of this standard.



**Figure 0 — Layered view of the clauses in this standard**

American National Standard  
for Information Technology -

PCI Express<sup>®</sup> Queuing Interface (PQI)

1 Scope

The SCSI family of standards provides for different transport protocols that define the methods for exchanging information between SCSI devices. This standard defines the transport methods for exchanging information between SCSI devices using a PCI Express interconnect. This standard defines a queuing layer, used by SOP. Other SCSI transport protocol standards define the methods for exchanging information between SCSI devices using other interconnects. Figure 1 shows the relationship of this standard to the other standards and related projects in the SCSI family of standards.

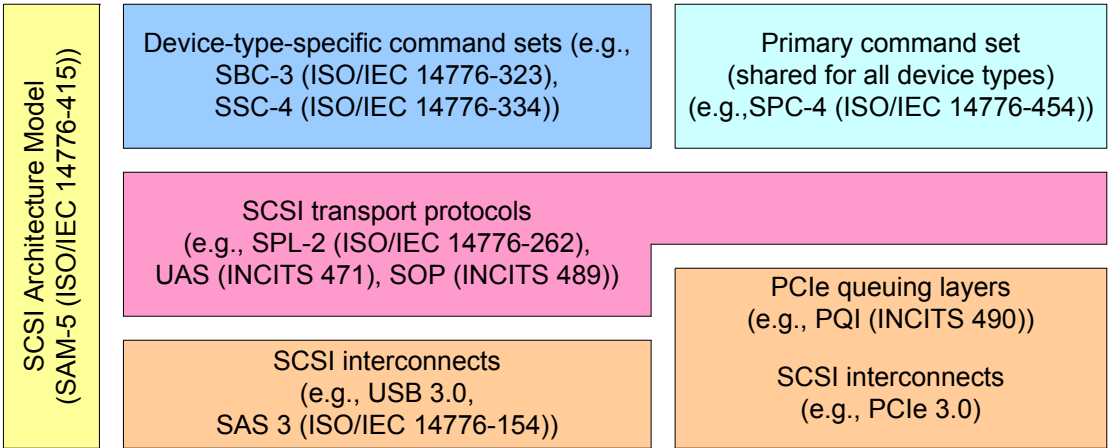


Figure 1 — SCSI document relationships

PCI Express is used as the interconnect for this standard. This standard defines a generic queuing model to allow transporting IUs.

## 2 Normative references

### 2.1 Normative references overview

Referenced standards and specifications contain provisions that, by reference in the text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

Copies of the following documents may be obtained from ANSI:

- a) approved ANSI standards;
- b) approved and draft international and regional standards (ISO, IEC, CEN/ELEC, ITU-T); and
- c) approved and draft foreign standards (including BSI, JIS, and DIN).

For further information, contact ANSI Customer Service Department at 212-642-4980 (phone), 212-302-1286 (fax) or via the World Wide Web at <http://www.ansi.org>.

Additional availability contact information is provided below as needed.

### 2.2 References under development

At the time of publication, the following referenced standards were still under development. For information on the current status of a document, or regarding availability, contact the relevant standards body or other organization as indicated.

ISO/IEC 14776-415, *SCSI Architecture Model - 5 (SAM-5)* (T10/BSR INCITS 515)

ISO/IEC 14776-454, *SCSI Primary Commands - 4 (SPC-4)* (T10/BSR INCITS 513)

ISO/IEC 14776-271, *SCSI Over PCI Express® (SOP)* (T10/BSR INCITS 489)

NOTE 1 - For more information on the current status of these documents, contact the INCITS Secretariat at 202-737-8888 (phone), 202-638-4922 (fax) or via Email at [incits@itic.org](mailto:incits@itic.org). To obtain copies of these documents, contact Global Engineering at 15 Inverness Way, East Englewood, CO 80112-5704 at 303-792-2181 (phone), 800-854-7179 (phone), or 303-792-2192 (fax) or see <http://www.incits.org>.

### 2.3 Other references

For information on the current status of a document, or regarding availability, contact the indicated organization.

*PCI Local Bus Specification Revision 3.0 (PCI)*, 3 February 2004

*PCI Express® Base Specification Revision 3.0 (PCIe)*, 10 November 2010

*PCI Code and ID Assignment Specification (PCI-ID) Revision 1.3*, 4 September 2012

*PCI Bus Power Management Interface Specification (PCI-PM) revision 1.2*, 3 March 2004

*PCI Express Card Electromechanical Specification (PCIe-CEM) revision 3.0, version 0.9*, 23 May 2011

NOTE 2 - For information on the current status of PCI documents, contact the PCI-SIG (see <http://www.pcisig.com>).

*Advanced Configuration and Power Interface Specification (ACPI). Revision 5.0*, 6 December 2011

NOTE 3 - For information on the current status of ACPI documents, contact the ACPI Promoters (see <http://www.acpi.info>).

## 3 Definitions, symbols, abbreviations, and conventions

### 3.1 Definitions

#### 3.1.1 administrator function (see 10.2)

unit of work to be performed by a PQI device management device server

#### 3.1.2 administrator queue pair

administrator IQ (see 5.2.5.8) and administrator OQ (see 5.2.5.11)

#### 3.1.3 aggregation (see 3.9.4)

when used in class diagrams, form of association that defines a whole-part relationship between the whole (i.e., aggregate) and its parts

#### 3.1.4 alternative SGL segment (see annex A)

data structure in a contiguous region of memory space describing all or part of a data buffer

#### 3.1.5 association

when used in class diagrams, relationship between two or more classes that specifies connections among their objects (i.e., relationship that specifies that objects of one class are connected to objects of another class)

#### 3.1.6 attribute

when used in class diagrams, named property of a class that describes a range of values that the class or its objects may hold; when used in object diagrams, named property of an instance of a class

#### 3.1.7 big-endian (see 3.7)

format for binary data in which the most significant byte is first

#### 3.1.8 byte

sequence of eight contiguous bits considered as a unit

#### 3.1.9 circular queue (see 5.3.2)

data structure with an element array, consumer index, and producer index, that is used as if the first element follows the last element

#### 3.1.10 configuration space (see PCI and PCIe)

byte addressable physical address space defined by PCI used to access registers of PCI functions (see PCI), addressed using ID-based routing (i.e., bus number, device number, if any, and function number), and accessible with configuration transactions

#### 3.1.11 configuration transaction (see PCIe)

Request (see PCIe) and Completion (see PCIe) used to access configuration space

#### 3.1.12 consume

read the content of the occupied element in a circular queue indicated by the circular queue's CI working copy and then increment the CI working copy

#### 3.1.13 consumer

entity that consumes one or more elements from a circular queue

#### 3.1.14 consumer index (CI) (see 5.2.5.6)

index of a circular queue element array that points to the next element to be read by the consumer

**3.1.15 constraint**

when used in class diagrams and object diagrams, mechanism for specifying semantics or conditions that are maintained as true between entities (e.g., a required condition between associations)

**3.1.16 class**

description of a set of objects that share the same attributes, operations, relationships, and semantics; classes may have attributes and may support operations

**3.1.17 data block**

contiguous region of memory space

**3.1.18 data buffer**

one or more data blocks

**3.1.19 Data Buffer**

Data-In Buffer or Data-Out Buffer

**3.1.20 Data-In Buffer**

buffer specified by a PQI host management application client to receive data from a PQI device management device server

**3.1.21 Data-Out Buffer**

buffer specified by a PQI host management application client to supply data that is transferred to a PQI [device](#) management device server

**3.1.22 dequeue**

read an IU from a circular queue by consuming all elements within a circular queue that contain portions of that single IU

**3.1.23 destination data stream**

data that is being written to the PQI device

**3.1.24 dword**

sequence of four contiguous bytes considered as a unit

**3.1.25 element**

set of contiguous bytes in memory space that are part of a circular queue

**3.1.26 element array** (see 5.2.5.3)

set of contiguous elements

**3.1.27 element array address**

address of the start of the first element in the element array

**3.1.28 enqueue**

write an IU to a circular queue by producing sufficient elements to contain the content of the IU

**3.1.29 error code**

combination of the ERROR CODE field (see 5.6) and the ERROR CODE QUALIFIER field (see 5.6)

**3.1.30 field**

group of one or more contiguous bits that are part of a larger structure such as an IU

**3.1.31 generalization**

when used in class diagrams, relationship among classes where one class (i.e., superclass) shares the attributes and/or operations of one or more classes (i.e., subclasses)



**3.1.32 inbound IU**

IU contained in an IQ

**3.1.33 inbound queue (IQ)** (see 5.2.5.7)

circular queue that is used to transfer IUs from a PQI host to a PQI device

**3.1.34 information unit (IU)** (see 9.2)

delimited and sequenced set of information in a format appropriate for transport by the service delivery subsystem

**3.1.35 interrupt coalescing** (see 5.4.2)

technique for reducing the number of interrupts

**3.1.36 IQ ID** (see 5.1)

identifier assigned to an IQ

**3.1.37 least significant bit (LSB)**

in a binary code, bit or bit position with the smallest numerical weighting in a group of bits that, when taken as a whole, represent a numerical value (e.g., in the number 0001b, the bit that is set to one)

**3.1.38 legacy INTx** (see PCI and PCIe)

interrupt mechanism defined by PCI and supported by PCI Express using virtual INTx wires

**3.1.39 little-endian** (see 3.7)

format for binary data in which the least significant byte is first

**3.1.40 memory read transaction** (see PCIe)

memory transaction used to read from memory space

**3.1.41 memory space** (see PCI and PCIe)

64-bit byte addressable physical address space defined by PCI, addressed using address based routing (i.e., a 64-bit address), and accessible with memory transactions

**3.1.42 memory transaction** (see PCIe)

Request (see PCIe) and one or more Completions (see PCIe) used to access memory space

**3.1.43 memory write transaction** (see PCIe)

memory transaction used to write to memory space

**3.1.44 most significant bit (MSB)**

in a binary code, bit or bit position with the largest numerical weighting in a group of bits that, when taken as a whole, represent a numerical value (e.g., in the number 1000b, the bit that is set to one)

**3.1.45 MSI-X Table** (see PCI and PCIe)

structure containing MSI-X vector control, message address, and message data registers

**3.1.46 MSI-X PBA** (see PCI)

structure containing MSI-X pending bits

**3.1.47 multiplicity**

when used in class diagrams, indication of the range of allowable instances that a class or an attribute may have

**3.1.48 non-data administrator function**

administrator function that does not transfer data

**3.1.49 object**

entity with a well defined boundary and identity that encapsulates state and behavior; all objects are instances of classes (i.e., a concrete manifestation of a class is an object)

**3.1.50 occupied element**

element that contains valid data

**3.1.51 operation**

when used in class diagrams, service that may be requested from any object of the class in order to effect behavior

**3.1.52 OQ ID** (see 5.1)

identifier assigned to an OQ

**3.1.53 outbound IU**

IU contained in an OQ

**3.1.54 outbound queue (OQ)** (see 5.2.5.10)

circular queue that is used to transfer IUs from a PQI device to a PQI host

**3.1.55 PCI Express device** (see PCIe)

collection of one or more PCI functions identified by a common bus number (see PCIe) and common device number (see PCIe), if any

**3.1.56 PCI Express hierarchy domain** (see PCIe)

the part of an I/O system (e.g., PCI Express switches and PCI Express links) that allows a PQI host and a PQI device to communicate

**3.1.57 PCI Express host** (see PCIe)

device containing a Root Complex Component (see PCIe) that connects to a host CPU (see PCI)

**3.1.58 PCI Express reset** (see PCIe)

PCI Express cold reset (see PCIe), PCI Express function-level reset (see PCIe), PCI Express hard reset (see PCIe), PCI Express hot reset (see PCIe), PCI Express soft reset (see PCIe), or PCI Express warm reset (see PCIe)

**3.1.59 PD function**

unit of work to be performed by a PQI device that is specified in the Administrator Queue Configuration Function register (see 6.2.5)

**3.1.60 PQI reset** (see 5.7)

PQI soft reset (see 5.7.2), PQI firm reset (see 5.7.3), or PQI hard reset (see 5.7.4)

**3.1.61 PQI device** (see 5.2.3)

kind of PCI function (see PCI) that is compliant with this standard

**3.1.62 PQI device memory space**

memory space (see 3.1.41) in a PQI device whose address is specified in the first memory BAR in the PQI device's configuration space

**3.1.63 PQI domain** (see 5.2.1)

one or more PQI hosts, one or more PQI devices, and the PQI service delivery subsystem through which they communicate

**3.1.64 PQI host** (see 5.2.2)

kind of PCI Express host or PCI Express device that communicates with a PQI device

**3.1.65 PQI host management application client** (see 5.2.2.3)

object in a PQI host that requests management tasks to be performed by PQI device management device servers

**3.1.66 PQI device management device server** (see 5.2.3.4)

object in a PQI device that performs management tasks requested by PQI host management application clients

**3.1.67 produce**

write content to the vacant element in the circular queue indicated by the circular queue's PI working copy and then increment the PI working copy

**3.1.68 producer**

entity that produces one or more elements to a circular queue

**3.1.69 producer index (PI)** (see 5.2.5.5)

index of a circular queue's element array where the new entries are written by the producer

**3.1.70 PCI Express Queuing Interface (PQI)**

queuing layer defined by this standard

**3.1.71 PCI memory BAR** (see PCI)

BAR (see PCI) that describes a region of memory space

**3.1.72 read administrator function**

administrator function for which a PQI device management device server transfers data to a PQI host management application client using a Data-In Buffer

**3.1.73 role**

when used in class diagrams and object diagrams, label at the end of an association or aggregation that defines a relationship to the class on the other side of the association or aggregation

**3.1.74 scatter gather list (SGL)** (see clause 8)

data structure used to describe a data buffer

**3.1.75 SGL segment**

standard SGL segment or alternative SGL segment

**3.1.76 source data stream**

data that is being read from the PQI device

**3.1.77 standard SGL segment** (see 8.2)

data structure in a contiguous region of memory space describing all or part of a data buffer and the next SGL segment, if any

**3.1.78 vacant element**

element that does not contain valid data

**3.1.79 virtual INTx wire** (see PCIe)

virtual wire controlled by PCI Express Assert\_INTx and Deassert\_INTx Messages for emulation of legacy INTx interrupts

**3.1.80 write administrator function**

administrator function for which a PQI device management device server transfers data from a PQI host management application client using a Data-Out Buffer

## 3.2 Symbols and abbreviations

Units and abbreviations used in this standard:

Abbreviation	Meaning
+	add
-	subtract
< or LT	less than
= or EQ	equal
> or GT	greater than
≥ or GE	greater than or equal to
ACPI	Advanced Configuration and Power Interface (see 2.3)
ID	identifier
ARI	Alternative Routing-ID Interpretation (see PCIe)
B	byte
BAR	base address register (see PCI)
CI	consumer index (see 3.1.14)
INTx	interrupt A, B, C, or D (see PCI)
IQ	inbound queue (see 3.1.33)
IU	information unit (see 3.1.34)
LUN	logical unit number
LSB	least significant bit (see 3.1.37)
ms	millisecond (i.e., 10 <sup>-3</sup> seconds)
MSB	most significant bit (see 3.1.44)
MSI-X	Message Signaled Interrupt Extended (see PCI)
ns	nanosecond (i.e., 10 <sup>-9</sup> seconds)
OQ	outbound queue (see 3.1.54)
PBA	Pending Bit Array (see 3.1.46 and PCI)
PCI	Peripheral Component Interconnect (see 2.3)
PCIe	PCI Express® Base Specification (see 2.3)
PD	PQI device state machine (see 5.5)
PI	producer index (see 3.1.69)
POST	power on self test
PQI	PCI Express Queuing Interface (i.e., this standard)
SAM-5	SCSI Architecture Model-5 (see 2.2)
SGL	scatter gather list (see 3.1.74)
SOP	SCSI over PCI Express®
SPC-4	SCSI Primary Commands-4 (see 2.2)

## 3.3 Keywords

### 3.3.1 invalid

keyword used to describe an illegal or unsupported bit, byte, word, field or code value; receipt of an invalid bit, byte, word, field or code value shall be reported as an error

**3.3.2 mandatory**

keyword indicating an item that is required to be implemented as defined in this standard

**3.3.3 may**

keyword that indicates flexibility of choice with no implied preference (equivalent to “may or may not”)

**3.3.4 may not**

keywords that indicate flexibility of choice with no implied preference (equivalent to “may or may not”)

**3.3.5 obsolete**

keyword indicating that an item was defined in prior standards but has been removed from this standard

**3.3.6 optional**

keyword that describes features that are not required to be implemented by this standard; however, if any optional feature defined in this standard is implemented, then it shall be implemented as defined in this standard

**3.3.7 reserved**

keyword referring to bits, bytes, words, fields and code values that are set aside for future standardization; a reserved bit, byte, word or field shall be set to zero, or in accordance with a future extension to this standard; recipients are not required to check reserved bits, bytes, words or fields for zero values; receipt of reserved code values in defined fields shall be reported as an error

**3.3.8 RsvdC**

keyword referring to bits, bytes, or fields that are set aside for future standardization that shall be set to zero or in accordance with a future extension to this standard, and shall be checked for zero values by the recipient

**3.3.9 RsvdZ**

keyword referring to bit, byte, or field in a PQI device for future standardization that should be set to zero by the PQI host during memory writes, shall be ignored by the PQI device during memory writes, and shall be set to zero by the PQI device during memory reads

**3.3.10 restricted**

keyword referring to bits, bytes, words, and fields that are set aside for other identified standardization purposes; a restricted bit, byte, word, or field shall be treated as a reserved bit, byte, word or field in the context where the restricted designation appears

**3.3.11 shall**

keyword indicating a mandatory requirement; designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard

**3.3.12 should**

keyword indicating flexibility of choice with a strongly preferred alternative (equivalent to “is strongly recommended”)

**3.3.13 vendor specific**

something (e.g., a bit, field, or code value) that is not defined by this standard and may be used differently in various implementations

**3.4 Editorial conventions**

Certain words and terms used in this standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in the glossary or in the text where they first appear.

Upper case is used when referring to the name of a numeric value defined in this specification or a formal attribute possessed by an entity. When necessary for clarity, names of objects, procedure calls, arguments or

discrete states are capitalized or set in bold type. Names of fields are identified using small capital letters (e.g., NACA bit).

Names of procedure calls are identified by a name in bold type (e.g., **Execute Command**). Names of arguments are denoted by capitalizing each word in the name (e.g., Sense Data is the name of an argument in the **Execute Command** procedure call). For more information on procedure calls see 3.8.

Quantities having a defined numeric value are identified by large capital letters (e.g., CHECK CONDITION). Quantities having a discrete but unspecified value are identified using small capital letters. (e.g., TASK COMPLETE, indicates a quantity returned by the **Execute Command** procedure call). Such quantities are associated with an event or indication whose observable behavior or value is specific to a given implementation standard.

Lists sequenced by lowercase or uppercase letters show no ordering relationship between the listed items.

EXAMPLE 1 - The following list shows no relationship between the named items:

- a) red (i.e., one of the following colors):
  - A) crimson; or
  - B) amber;
- b) blue; or
- c) green.

Lists sequenced by numbers show an ordering relationship between the listed items.

EXAMPLE 2 -The following list shows an ordered relationship between the named items:

- 1) top;
- 2) middle; and
- 3) bottom.

If a conflict arises between text, tables, or figures, the order of precedence to resolve the conflicts is text; then tables; and finally figures. Not all tables or figures are fully described in the text. Tables show data format and values.

Notes and examples do not constitute any requirements for implementors and notes are numbered consecutively throughout this standard.

## 3.5 Numeric and character conventions

### 3.5.1 Numeric conventions

A binary number is represented in this standard by any sequence of digits comprised of only the Arabic numerals 0 and 1 immediately followed by a lower-case b (e.g., 0101b). Underscores or spaces may be included in binary number representations to increase readability or delineate field boundaries (e.g., 0 0101 1010b or 0\_0101\_1010b).

A hexadecimal number is represented in this standard by any sequence of digits comprised of only the Arabic numerals 0 through 9 and/or the upper-case English letters A through F immediately followed by a lower-case h (e.g., FA23h). Underscores or spaces may be included in hexadecimal number representations to increase readability or delineate field boundaries (e.g., B FD8C FA23h or B\_FD8C\_FA23h).

A decimal number is represented in this standard by any sequence of digits comprised of only the Arabic numerals 0 through 9 not immediately followed by a lower-case b or lower-case h (e.g., 25).

A range of numeric values is represented in this standard in the form “a to z”, where a is the first value included in the range, all values between a and z are included in the range, and z is the last value included in the range (e.g., the representation “0h to 3h” includes the values 0h, 1h, 2h, and 3h).

This standard uses the following conventions for representing decimal numbers:

- a) the decimal separator (i.e., separating the integer and fractional portions of the number) is a period;
- b) the thousands separator (i.e., separating groups of three digits in a portion of the number) is a space;

- c) the thousands separator is used in both the integer portion and the fraction portion of a number; and
- d) the decimal representation for a year is 1999 not 1 999.

Table 1 shows some examples of decimal numbers using various conventions.

**Table 1 — Numbering conventions**

French	English	This standard
0,6	0.6	0.6
3,141 592 65	3.14159265	3.141 592 65
1 000	1,000	1 000
1 323 462,95	1,323,462.95	1 323 462.95

### 3.5.2 Units of measure

This standard represents values using both decimal units of measure and binary units of measure. Values are represented by the following formats:

- a) for values based on decimal units of measure:
  - 1) numerical value (e.g., 100);
  - 2) space;
  - 3) prefix symbol and unit:
    - 1) decimal prefix symbol (e.g., M) (see table 2); and
    - 2) unit abbreviation (e.g., B);
- and
- b) for values based on binary units of measure:
  - 1) numerical value (e.g., 1 024);
  - 2) space;
  - 3) prefix symbol and unit:
    - 1) binary prefix symbol (e.g., Gi) (see table 2); and
    - 2) unit abbreviation (e.g., b).

Table 2 compares the prefix, symbols, and power of the binary and decimal units.

**Table 2 — Comparison of decimal prefixes and binary prefixes**

Decimal			Binary		
Prefix name	Prefix symbol	Power (base-10)	Prefix name	Prefix symbol	Power (base-2)
kilo	k	$10^3$	kibi	Ki	$2^{10}$
mega	M	$10^6$	mebi	Mi	$2^{20}$
giga	G	$10^9$	gibi	Gi	$2^{30}$
tera	T	$10^{12}$	tebi	Ti	$2^{40}$
peta	P	$10^{15}$	pebi	Pi	$2^{50}$
exa	E	$10^{18}$	exbi	Ei	$2^{60}$
zetta	Z	$10^{21}$	zebi	Zi	$2^{70}$
yotta	Y	$10^{24}$	yobi	Yi	$2^{80}$

### 3.5.3 Byte encoded character strings conventions

When this standard requires one or more bytes to contain specific encoded characters, the specific characters are enclosed in single quotation marks. The single quotation marks identify the start and end of the characters

that are required to be encoded but are not themselves to be encoded. The characters that are to be encoded are shown in the case that is to be encoded.

An ASCII space character (i.e., 20h) may be represented in a string by the character '¬' (e.g., 'SCSI¬device').

The encoded characters and the single quotation marks that enclose them are preceded by text that specifies the character encoding methodology and the number of characters required to be encoded.

EXAMPLE - Using the notation described in this subclause, stating that eleven ASCII characters 'SCSI device' are encoded to the following sequence of byte values: 53h 43h 53h 49h 20h 64h 65h 76h 69h 63h 65h.

## 3.6 State machine conventions

### 3.6.1 State machine conventions overview

Figure 2 shows how state machines are described.

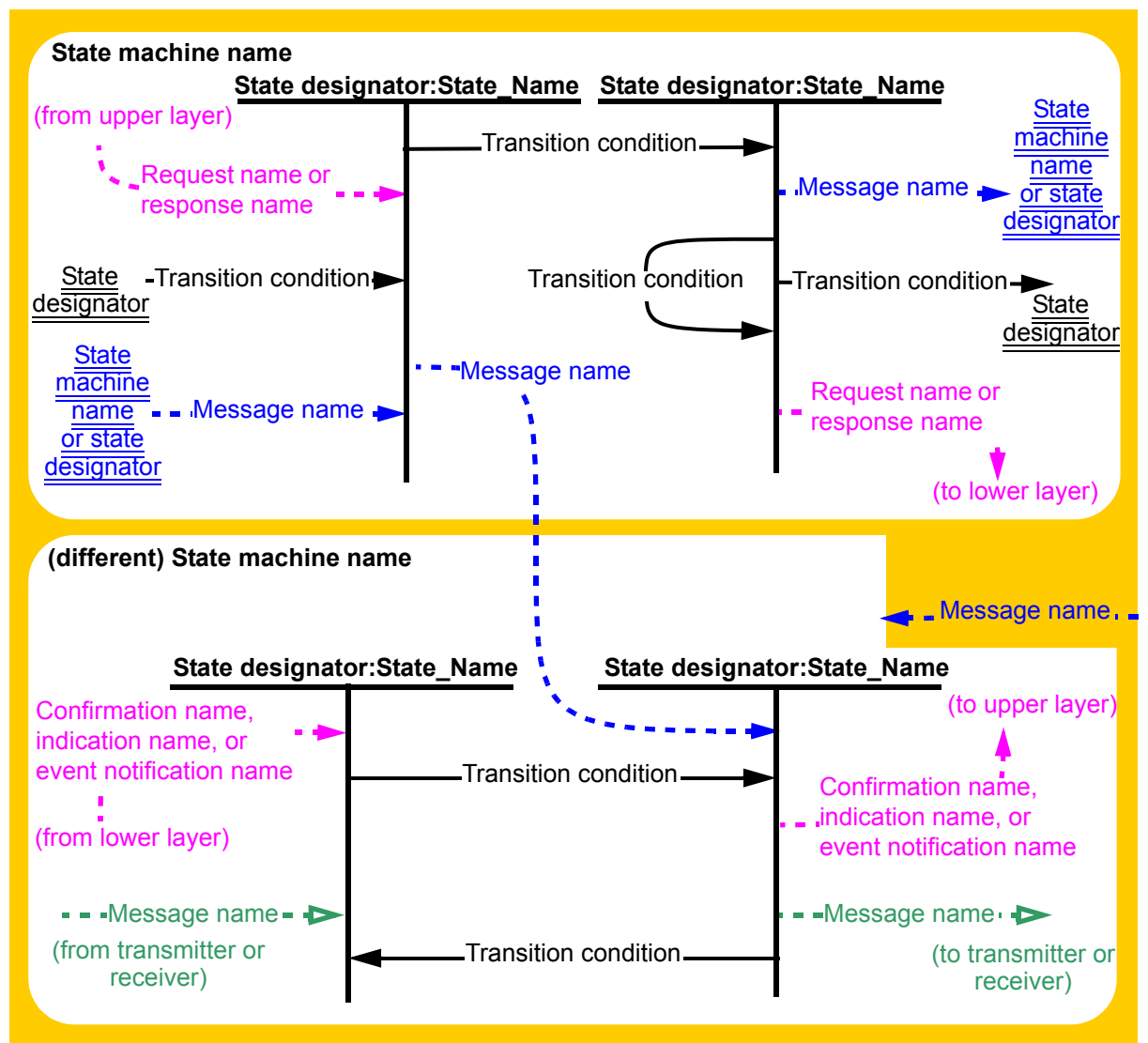


Figure 2 — State machine conventions



When multiple state machines are present in a figure, they are enclosed in boxes with rounded corners.

Each state is identified by a state designator and a state name. The state designator (e.g., SL1) is unique among all state machines in this standard. The state name (e.g., Idle) is a brief description of the primary action taken during the state, and the same state name may be used by other state machines. Actions taken while in each state are described in the state description text.

### 3.6.2 Transitions

Transitions between states are shown with solid lines, with an arrow pointing to the destination state. A transition may be labeled with a transition condition label, a brief description of the event or condition that causes the transition to occur.

If the state transition leaves the page, the transition label goes to or from a state designator label with double underlines rather than to or from a state.

The conditions and actions are described fully in the transition description text. In case of a conflict between a figure and the text, the text shall take precedence.

Upon entry into a state, all actions to be processed in that state are processed. If a state is re-entered from itself, all actions to be processed in the state are processed again. A state may be entered and exited in zero time if the conditions for exiting the state are valid upon entry into the state. Transitions between states are instantaneous.

### 3.6.3 Messages, requests, indications, confirmations, responses, and event notifications

Messages passed between state machines are shown with dashed lines labeled with a message name. When messages are passed between state machines within the same layer of the protocol, they are identified by either:

- a) a dashed line to or from a state machine name label with double underlines and/or state name label with double underlines, if the destination is in a different figure from the source;
- b) a dashed line to or from a state in another state machine in the same figure; or
- c) a dashed line from a state machine name label with double underlines to a "(to all states)" label, if the destination is every state in the state machine.

The meaning of each message is described in the state description text.

Requests, indications, confirmations, responses, and event notifications are shown with curved dashed lines originating from or going to the top or bottom of the figure. Each request, indication, confirmation, response, and event notification is labeled. The meaning of each request, indication, confirmation, response, and event notification is described in the state description text.

Messages with unfilled arrowheads are passed to or from the state machine's transmitter or receiver, not shown in the state machine figures, and are directly related to data being transmitted on or received from the physical link.

### 3.6.4 State machine counters, timers, and variables

State machines may contain counters, timers, and variables that affect the operation of the state machine. The scope of state machine counters, timers, and variables is the state machine itself. They are created and deleted with the state machines with which they are associated. State machine transitions specify the initialization and modification of state machine timers, counters, and variables. Transitions out of a state may be conditioned upon specific criteria regarding the current value of a state machine counter, timer, or variable. State machine timers may continue to run while a state machine is in a given state, and a timer may cause a state transition upon reaching a defined threshold value (e.g., zero for a timer that counts down).

### 3.6.5 State machine arguments

State machines may contain one or more arguments received in a message or confirmation as state machine arguments. The following properties apply to state machine arguments:

- a) the state machine that sends an argument owns that argument's value;
- b) the state machine that receives an argument shall not modify that argument's value;

- c) the state machine that sends an argument may resend that argument with a different value;
- d) the scope of a state machine argument is the state machine itself; and
- e) state machine argument usage is described in the state descriptions and the transition descriptions.

### 3.7 Bit and byte ordering

In this standard, data structures may be defined by a table. A table defines a complete ordering of elements (i.e., bits, bytes, fields, and dwords) within the structure. The ordering of elements within a table does not in itself constrain the order of storage or transmission of the data structure, but in combination with other normative text in this standard, may constrain the order of storage or transmission of the structure.

Tables defining data structures are shown with one row per byte and one column per bit. The lowest byte offset is at the top and the highest byte offset is at the bottom. The least significant bit (LSB) of each byte is numbered 0 and is shown on the right, and the most significant bit (MSB) of each byte is numbered 7 and shown on the left.

In a field in a table consisting of more than one bit that contains a single value (e.g., a number), the least significant bit (LSB) is shown on the right and the most significant bit (MSB) is shown on the left (e.g., in a byte, bit 7 is the MSB and is shown on the left, bit 0 is the LSB and is shown on the right). The MSB and LSB are not labeled if the field consists of eight or fewer bits and is contained within one row. The MSB and LSB are labeled if the field consists of more than eight bits, crosses a row, and has no internal structure defined.

In a big-endian field, the byte containing the MSB is at the lowest byte offset and the byte containing the LSB is at the highest byte offset. The bits in big-endian fields are not shaded.

In a little-endian field, the byte containing the MSB is at the highest byte offset and the byte containing the LSB is at the lowest byte offset. The bits in little-endian fields are shaded.

In a field in a table consisting of more than one byte that contains multiple fields each with their own values (e.g., a descriptor), there is no MSB and LSB of the field itself and thus there are no MSB and LSB labels for that field. The MSB and LSB of each subfield may be shown in another table.

In a field containing a text string (e.g., ASCII or UTF-8), only the MSB of the first character and the LSB of the last character are labeled.

Multiple byte fields are represented with three rows, with the non-sequentially increasing byte numbers separated by a row labeled '...'. .

Table 3 shows how this standard depicts a 32-bit big-endian field.

**Table 3 — Example of a 32-bit big-endian field**

Byte\Bit	7	6	5	4	3	2	1	0
...	Other fields, if any							
n	(MSB)							
...	Field name							
n + 3								
...	Other fields, if any							

Table 4 shows the bit numbers for the field shown in table 3.

**Table 4 — Bit assignments in a 32-bit big-endian field**

Byte\Bit	7	6	5	4	3	2	1	0
...	Other fields, if any							
<b>n</b>	(MSB) Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
<b>n + 1</b>	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
<b>n + 2</b>	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
<b>n + 3</b>	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
...	Other fields, if any							

EXAMPLE 1 - If the field in table 3 and table 4 contains a value of 00010203h, then:

- a) byte n contains 00h;
- b) byte n+1 contains 01h;
- c) byte n+2 contains 02h; and
- d) byte n+3 contains 03h.

Table 5 shows how this standard depicts a 32-bit little-endian field.

**Table 5 — Example of a 32-bit little-endian field**

Byte\Bit	7	6	5	4	3	2	1	0
...	Other fields, if any							
<b>n</b>	Field name (LSB)							
...								
<b>n + 3</b>								
...	Other fields, if any							

Table 6 shows the bit numbers for the field shown in table 5.

**Table 6 — Bit numbers for a 32-bit little-endian field**

Byte\Bit	7	6	5	4	3	2	1	0
...	Other fields, if any							
<b>n</b>	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	(LSB) Bit 0
<b>n + 1</b>	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
<b>n + 2</b>	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
<b>n + 3</b>	Bit 31 (MSB)	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
...	Other fields, if any							

EXAMPLE 2 - If the field in table 5 and table 6 contains a value of 00010203h, then:

- a) byte n contains 03h;
- b) byte n+1 contains 02h;
- c) byte n+2 contains 01h; and
- d) byte n+3 contains 00h.

### 3.8 Notation for procedure calls

In this standard, the model for functional interfaces between entities is a procedure call. Such interfaces are specified using the following notation:

**[Result =] Procedure Name (IN ( [input-1] [,input-2] ...), OUT ( [output-1] [,output-2] ... ))**

Where:

Result	A single value representing the outcome of the procedure or function.
Procedure Name	A descriptive name for the function to be performed.
IN (Input-1, Input-2, ...)	A comma-separated list of names identifying caller-supplied input data objects.
OUT (Output-1, Output-2, ...)	A comma-separated list of names identifying output data objects to be returned by the procedure.
[...]	Brackets enclose optional or conditional parameters and arguments.

This notation allows arguments to be specified as inputs and outputs. An interface between entities may require only inputs. If a procedure call has no output arguments, the word OUT, preceding comma, and associated pair of balanced parentheses are omitted.

The following is an example of a procedure call specification:

**Found = Search** (IN (Pattern, Item List), OUT ([Item Found]))

Input arguments:

**Pattern:** Argument containing the search pattern.

**Item List:** **Item<NN>** contains the items to be searched for a match.

Output arguments:

**Item Found:** Item located by the search procedure call. This argument is only returned if the search succeeds.

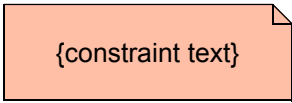
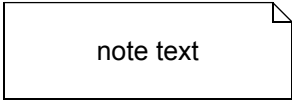
### 3.9 Notation for UML figures

#### 3.9.1 Introduction

This standard contains class diagram figures that use notation that is based on UML.

Some class diagrams contain constraints or notes that use the notion shown in table 7.

**Table 7 — Class diagram constraints and notes notation**

Notation	Description
	The presence of the curly brackets defines constraint that is a normative requirement. An example of a constraint is shown in figure 4.
	The absence of curly brackets defines a note that is informative. An example of a note is shown in figure 5.

The notation used to denote multiplicity in class diagrams is shown in table 8.

**Table 8 — Class diagram multiplicity notation**

Notation	Description
not specified	The number of instances of an attribute is not specified.
1	One instance of the class or attribute exists.
0..*	Zero or more instances of the class or attribute exist.
1..*	One or more instances of the class or attribute exist.
0..1	Zero or one instance of the class or attribute exists.
n..m	n to m instances of the class or attribute exist (e.g., 2..8).
x,n..m	Multiple disjoint instances of the class or attribute exist (e.g., 2, 8..15).

Class diagrams show:

- a) two or more classes (see 3.9.2); and
- b) one or more of the following relationships between them:
  - A) association (see 3.9.3);
  - B) aggregation (see 3.9.4);
  - C) generalization (see 3.9.5); and
  - D) dependency (see 3.9.6).

### 3.9.2 Class notation

The notation used for classes is shown in table 9.

**Table 9 — Class diagram notation for classes**

Notation			Description
<b>Class Name</b>	<b>Class Name</b>	<b>Class Name</b>	A class with no attributes or operations
	<b>Class Name</b>	<b>Class Name</b>	A class with attributes and no operations
	Attribute01[1] Attribute02[1]	Attribute01[1] Attribute02[1]	
		<b>Class Name</b>	A class with operations and no attributes
		Operation01() Operation02()	
		<b>Class Name</b>	A class with attributes and operations
		Attribute01[1] Attribute02[1] Operation01() Operation02()	
		<b>Class Name</b>	A class with attributes that have a specified multiplicity (see table 8) and operations
		Attribute01[1..*] Attribute02[1] Operation01() Operation02()	

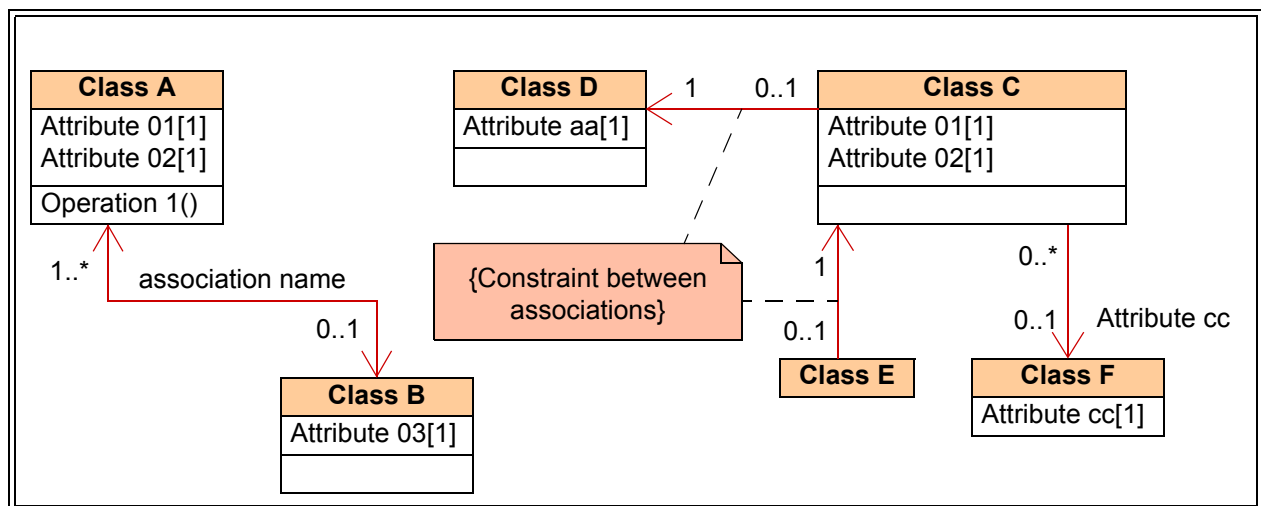
### 3.9.3 Class association relationships notation

The notation used to denote association (i.e., “knows about”) relationships between classes is shown in table 10. Unless the two classes in an association relationship also have an aggregation relationship (see 3.9.4), association relationships have multiplicity notation (see table 8) at each end of the relationship line.

**Table 10 — Class diagram notation for associations**

Notation	Description
	Class A knows about Class B (i.e., read as "Class A association name Class B") and Class B knows about Class A (i.e., read as "Class B association name Class A")
	Class B knows about Class A (i.e., read as "Class B knows about Class A") but Class A does not know about Class B
	Class A knows about Class B (i.e., read as "Class A uses the role name attribute of Class B") but Class B does not know about Class A
Note: The use of role names and association names are optional.	

Several example association relationships between classes are shown in figure 3.

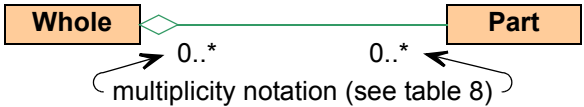



**Figure 3 — Example class association relationships**

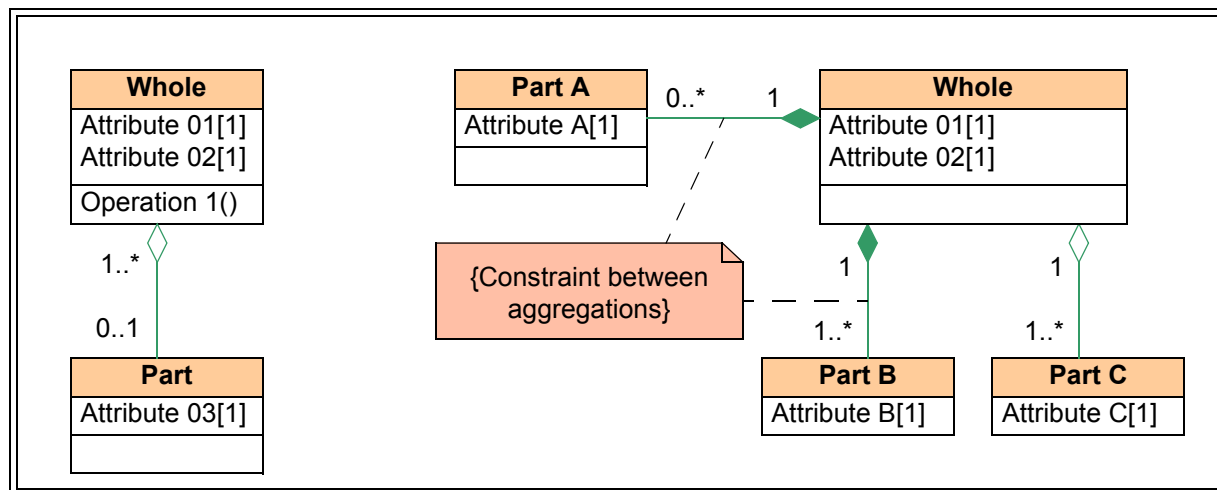
### 3.9.4 Class aggregation relationships notation

The aggregation relationship is a specific type of association. The notation used to denote aggregation (i.e., “is a part of” or “contains”) relationships between classes is shown in table 11. Aggregation relationships always include multiplicity notation (see table 8) at each end of the relationship line.

**Table 11 — Class diagram notation for aggregations**

Notation	Description
	The Part class is part of the Whole class and may continue to exist even if the Whole class is removed (i.e., read as “the Whole contains the Part.”)
	The Part class is part of the Whole class, shall only belong to one Whole class, and shall not continue to exist if the Whole class is removed (i.e., read as “the Whole contains the Part.”)

Several example aggregation relationships between classes are shown in figure 4.




**Figure 4 — Example class aggregation relationships**



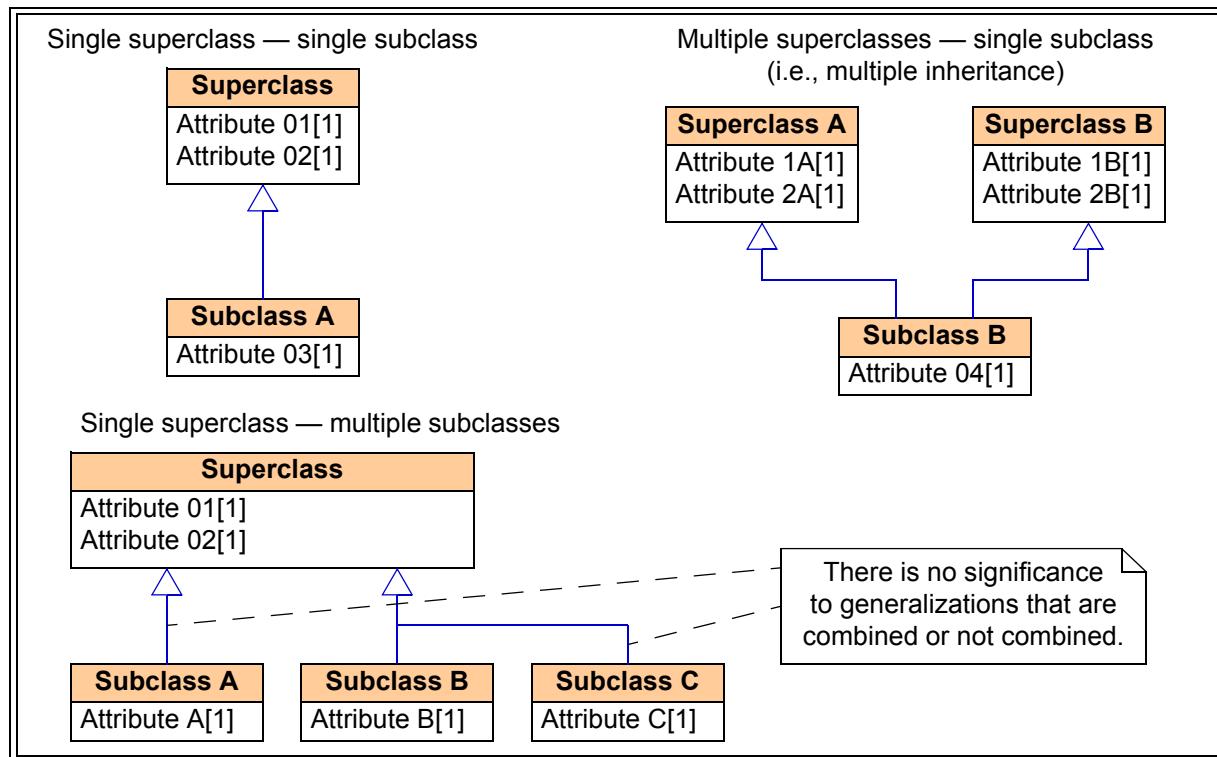
### 3.9.5 Class generalization relationships notation

The notation used to denote generalization (i.e., “is a kind of”) relationships between classes is shown in table 12.

**Table 12 — Class diagram notation for generalizations**

Notation	Description
	Subclass is a kind of superclass. A subclass shares all the attributes and operations of the superclass (i.e., the subclass inherits from the superclass).

Several example generalization relationships between classes are shown in figure 5.




**Figure 5 — Example class generalization relationships**

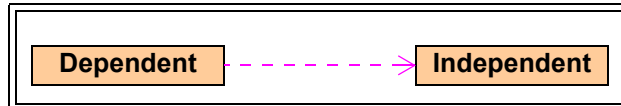
### 3.9.6 Class dependency relationships notation

The notation used to denote dependency (i.e., “depends on”) relationships between classes is shown in table 13.

**Table 13 — Class diagram notation for dependencies**

Notation	Description
	Class A depends on class B. A change in class B may cause a change in class A.

An example dependency relationship between classes is shown in figure 6.



**Figure 6 — Example class dependency relationships**

### 3.9.7 Object notation

The notation used for objects is shown in table 14.

**Table 14 — Notation for objects**

Notation	Description
	Notation for a named object with no attributes
<div>label : Class Name</div> <div>Attribute01 = x</div> <div>Attribute02 = y</div>	Notation for a named object with attributes
<div>: Class Name</div>	Notation for an anonymous object with no attributes
<div>: Class Name</div> <div>Attribute01 = x</div> <div>Attribute02 = y</div>	Notation for an anonymous object with attributes

## **4 General concepts**

### **4.1 ASCII data field requirements**

ASCII data fields shall contain only ASCII printable characters (i.e., code values 20h to 7Eh) and may be terminated with one or more ASCII null (00h) characters.

ASCII data fields described as being left-aligned shall have any unused bytes at the end of the field (i.e., highest offset) and the unused bytes shall be filled with ASCII space characters (20h).

ASCII data fields described as being right-aligned shall have any unused bytes at the start of the field (i.e., lowest offset) and the unused bytes shall be filled with ASCII space characters (20h).

## 5 Model

### 5.1 General overview

This standard defines:

- a) a method to manage (e.g., create, delete, and configure) circular queues;
- b) an interface for transferring information between a PQI host and a PQI device over a PQI service delivery subsystem in a PQI domain, using circular queues; and
- c) an SGL (see clause 8) format that is used to describe data buffers.

Certain types of information (e.g., command/functions information and response information) are packaged into IUs and transferred using circular queues. Other types of information (e.g., data) are transferred using memory transactions.

Information is transferred between a PQI host and a PQI device using one or more circular queues. A producer encapsulates the information into an IU and enqueues that IU to a circular queue. A consumer dequeues the IU from the circular queue and extracts the information from the IU.

Each circular queue is unidirectional:

- a) an IQ is used to transfer IUs from the PQI host to the PQI device; and
- b) an OQ is used to transfer IUs from the PQI device to the PQI host.

A PQI host provides:

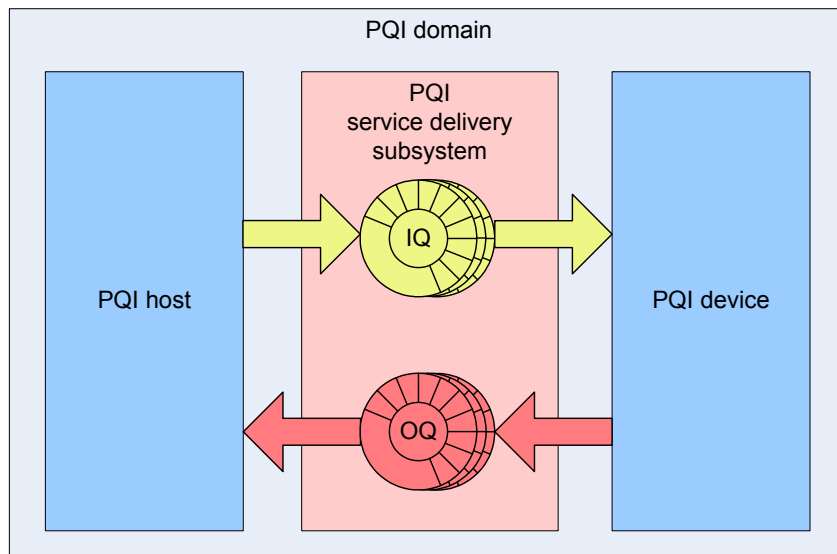
- a) the PQI host management application client to perform management of the circular queues; and
- b) an interface to support the PQI host operational application client.

A PQI device provides:

- a) the PQI device management device server to perform PQI management request functions requested by the PQI host management application client; and
- b) an interface to support the PQI operational device server.

Figure 7 shows:

- a) a PQI host;
- b) a PQI device; and
- c) a PQI service delivery subsystem that contains a PQI domain with:
  - A) IQs; and
  - B) OQs.



**Figure 7 — PQI device, PQI host, PQI service delivery subsystem, IQs, and OQs**

This standard defines the following types of circular queues:

- a) administrator queues; and
- b) operational queues.

Administrator queues are used by the PQI host management application client and the PQI device management device server to transfer administrator IUs that are used to manage the PQI domain. Only one administrator IQ and only one administrator OQ (i.e., an administrator queue pair) shall be present within a PQI domain.

Operational queues are used by a PQI host operational application client interface and a PQI operational device server as specified by other IU layers (e.g., SOP) to transfer IU layer specific IUs. Multiple operational IQs and multiple operational OQs may be present within a PQI domain. The number of operational IQs may be greater than, less than, or equal to the number of operational OQs.

The queuing model in this standard applies to both types of circular queues, although the methods of creating queues of the two types differ.

Administrator IUs:

- a) are used for managing the PQI device and are defined in this standard (see clause 10); and
- b) are transferred between the PQI host management application client and the PQI device management device server using an administrator IQ or an administrator OQ.

The administrator IQ and administrator OQ are created and configured using PQI device registers (see 6.2).

Operational IUs:

- a) are defined in other standards (e.g., SOP); and
- b) are transferred using the PQI host operational application client interface and the PQI device operational device server interface through an operational IQ or an operational OQ.

Operational IQs and OQs are created and configured by the PQI host management application client using administrator IUs (see 10.2).

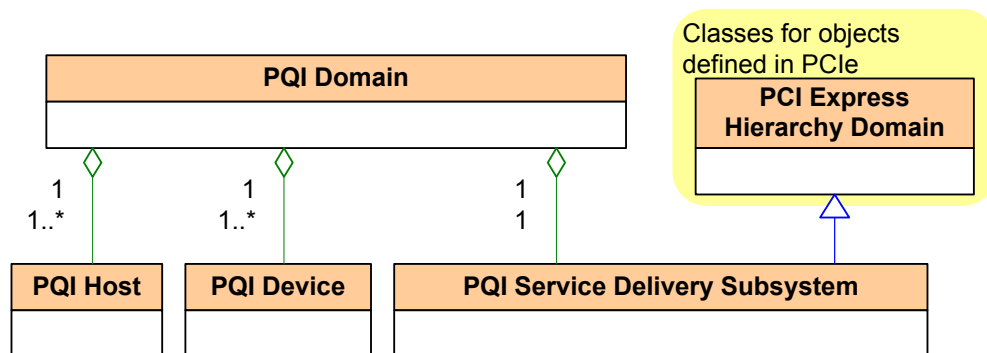
An IQ ID value is specified when an operational IQ is created and is used to identify that specific operational IQ. An OQ ID value is specified when an operational OQ is created and is used to identify that specific operational OQ. The values of the IQ IDs assigned to operational IQs are independent from the values of the OQ IDs assigned to operational OQs.

## 5.2 PQI classes

### 5.2.1 PQI domain class

Figure 8 describes the classes related to the PQI domain, showing the relationships between the following classes:

- a) PQI Domain;
- b) PCI Express Hierarchy Domain;
- c) PQI Service Delivery Subsystem (see 5.2.4);
- d) PQI Device (see 5.2.3.2); and
- e) PQI Host (see 5.2.2.2).



**Figure 8 — PQI Domain class diagram**

Each instance of a PQI Domain class (see figure 8) shall contain the following:

- a) one or more instances of the PQI Host class (see 5.2.2);
- b) one or more instances of the PQI Device class (see 5.2.3); and
- c) one instance of the PQI Service Delivery Subsystem class (see 5.2.4).

The PQI domain represents an instance of the PQI Domain class.

### 5.2.2 PQI host classes

#### 5.2.2.1 PQI host classes overview

Figure 9 describes the classes related to the PQI host, showing the relationship between the following classes:

- a) PCI Express Host (see PCIe);
- b) PCI Express Function (see PCIe);
- c) PQI Host (see 5.2.2.2);
- d) PQI Host Management Application Client (see 5.2.2.3);
- e) PQI Host Administrator OQ Information (see 5.2.2.4);
- f) PQI Host Administrator IQ Information (see 5.2.2.5);
- g) PQI Host Operational Application Client Interface (see 5.2.2.6);
- h) PQI Host Operational OQ Information (see 5.2.2.7); and
- i) PQI Host Operational IQ Information (see 5.2.2.8).

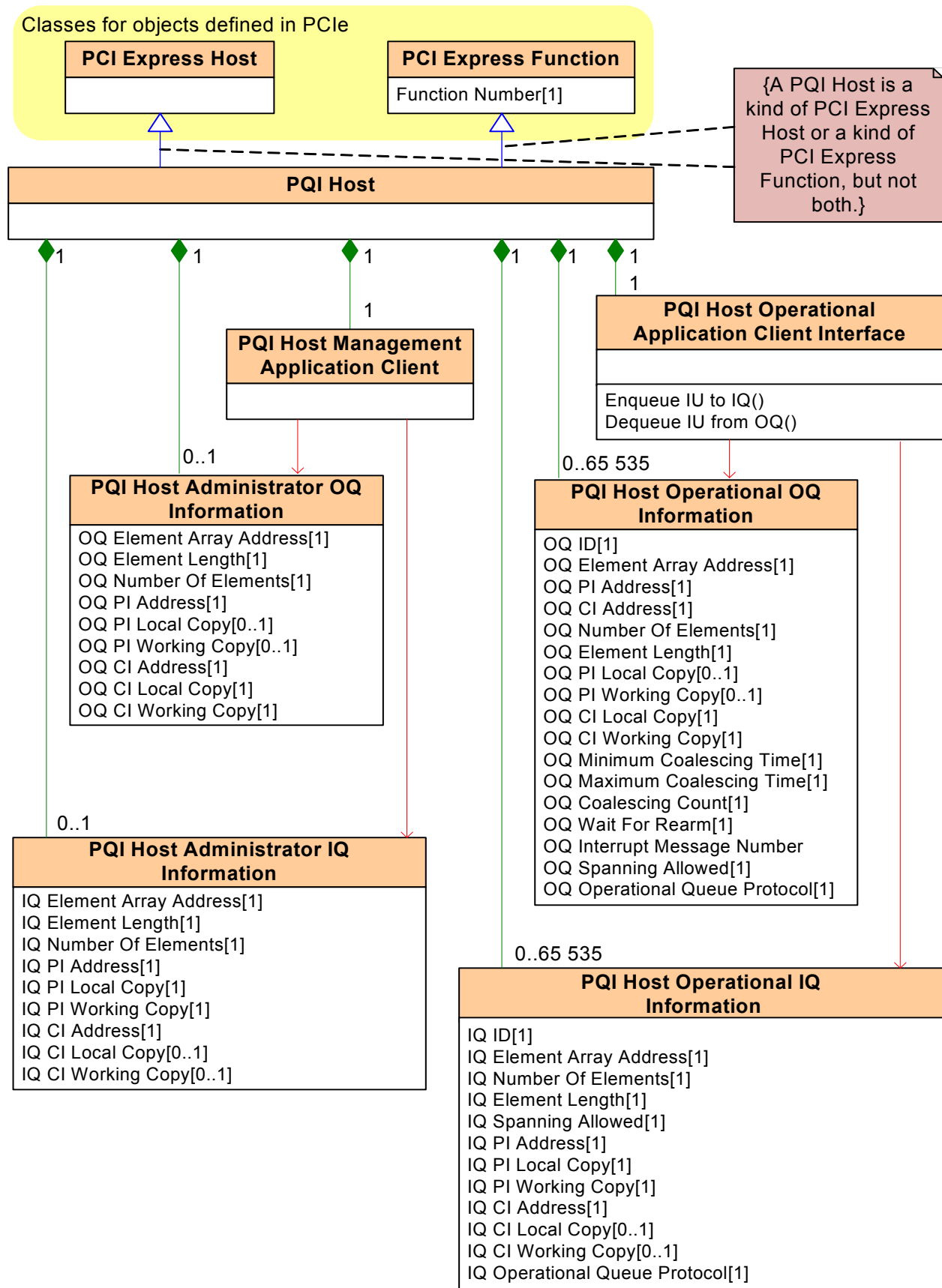


Figure 9 — PQI Host class diagram

### 5.2.2.2 PQI Host class

The PQI Host class (see figure 9) is a kind of either PCI Express Host class or a kind of PCI Express Function class but not both, and contains the:

- a) PQI Host Management Application Client class (see 5.2.2.3);
- b) PQI Host Administrator OQ Information class (see 5.2.2.4);
- c) PQI Host Administrator IQ Information class (see 5.2.2.5);
- d) PQI Host Operational Application Client Interface class (see 5.2.2.6);
- e) PQI Host Operational OQ Information class (see 5.2.2.7); and
- f) PQI Host Operational IQ Information class (see 5.2.2.8).

The PQI host represents an instance of the PQI Host class.

### 5.2.2.3 PQI Host Management Application Client class

The PQI Host Management Application Client class (see figure 9) requests administrator operations to be performed by a PQI device management device server.

### 5.2.2.4 PQI Host Administrator OQ Information class

The PQI host administrator OQ information represents an instance of the PQI Host Administrator OQ Information class (see figure 9).

Each instance of the PQI Host Administrator OQ Information class (see figure 9) contains the following attributes that describe an instance of the Administrator OQ class (see 5.2.5.11 and figure 13):

- a) the OQ Element Array Address attribute, which contains the OQ element array address (see 5.2.5.11.3);
- b) the OQ Number Of Elements attribute, which contains the number of elements in the OQ (see 5.2.5.11.5);
- c) the OQ Element Length attribute, which contains the length in bytes of the elements of the OQ (see 5.2.5.11.4);
- d) the OQ PI Address attribute, which contains the OQ PI address (see 5.2.5.11.6);
- e) the OQ PI Local Copy attribute, if any, which contains the OQ PI local copy (see 5.3.2.3);
- f) the OQ PI Working Copy attribute, if any, which contains the OQ PI working copy (see 5.3.2.3);
- g) the OQ CI Local Copy attribute, which contains the OQ CI local copy (see 5.3.2.4); and
- h) the OQ CI Working Copy attribute, which contains the OQ CI working copy (see 5.3.2.4).

### 5.2.2.5 PQI Host Administrator IQ Information class

The PQI host administrator IQ information represents an instance of the PQI Host Administrator IQ Information class (see figure 9).

Each instance of the PQI Host Administrator IQ Information class (see figure 9) contains the following attributes that describe an instance of the Administrator IQ class (see 5.2.5.8 and figure 13):

- a) the IQ Element Array Address attribute, which contains the IQ element array address (see 5.2.5.8.2);
- b) the IQ Number Of Elements attribute, which contains the number of elements in the IQ (see 5.2.5.8.4);
- c) the IQ Element Length attribute, which contains the length in bytes of the elements of the IQ (see 5.2.5.8.3);
- d) the IQ PI Address attribute, which contains the IQ PI address (see 5.2.5.8.5);
- e) the IQ PI Local Copy attribute, which contains the IQ PI local copy (see 5.3.2.3);
- f) the IQ PI Working Copy attribute, which contains the IQ PI working copy (see 5.3.2.3);
- g) the IQ CI Address attribute, which contains the IQ CI address (see 5.2.5.8.6);
- h) the IQ CI Local Copy attribute, if any, which contains the IQ CI local copy (see 5.3.2.4); and
- i) the IQ CI Working Copy attribute, if any, which contains the IQ CI working copy (see 5.3.2.4).



### 5.2.2.6 PQI Host Operational Application Client Interface class

The PQI Host Operational Application Client Interface class (see figure 9) provides services to the host queuing layer interface defined in the IU layer standard (e.g., SOP). The PQI Host Operational Application Client Interface provides:

- a) enqueueing of IUs compliant with clause 9 to operational IQs;
- b) dequeueing of IUs compliant with clause 9 from operational OQs; and
- c) generation of notifications of IUs on OQs available to be dequeued.

The PQI Host Operational Application Client Interface class processes Enqueue To IQ() requests (see 5.3.2.5) and Dequeue From OQ() requests (see 5.3.2.6) from the host queuing layer interface defined in the IU layer standard (e.g., SOP).

The PQI Host Operational Application Client Interface class provides IU Available On OQ operations to the host queuing layer interface defined in the IU layer standard (e.g., SOP).

### 5.2.2.7 PQI Host Operational OQ Information class

The PQI host operational OQ information represents an instance of the PQI Host Operational OQ Information class (see figure 9).

Each instance of the PQI Host Operational OQ Information class (see figure 9) contains the following attributes that describe an instance of the Operational OQ class (see 5.2.5.12 and figure 13):

- a) the OQ ID attribute, which contains the OQ ID (see 5.2.5.12.2);
- b) the OQ Element Array Address attribute, which contains the OQ element array address (see 5.2.5.12.3);
- c) the OQ PI Address attribute, which contains the OQ PI address (see 5.2.5.12.6);
- d) the OQ CI Address attribute, which contains the OQ CI address (see 5.2.5.12.7);
- e) the OQ Number Of Elements attribute, which contains the number of elements in the OQ (see 5.2.5.12.5);
- f) the OQ Element Length attribute which, contains the length in bytes of the elements of the OQ (see 5.2.5.12.4);
- g) the OQ PI Local Copy attribute, if any, which contains the OQ PI local copy (see 5.3.2.3);
- h) the OQ PI Working Copy attribute, if any, which contains the OQ PI working copy (see 5.3.2.3);
- i) the OQ CI Local Copy attribute, which contains the OQ CI local copy (see 5.3.2.4);
- j) the OQ CI Working Copy attribute, which contains the OQ CI working copy (see 5.3.2.4);
- k) the OQ Minimum Coalescing Time attribute, which contains the minimum coalescing time (see 5.2.5.12.8);
- l) the OQ Maximum Coalescing Time attribute, which contains the maximum coalescing time (see 5.2.5.12.9);
- m) the OQ Coalescing Count attribute, which contains the coalescing count (see 5.2.5.12.10);
- n) the OQ Wait For Rearm attribute, which specifies whether the device shall wait for a rearm before generating another interrupt (see 5.2.5.12.11);
- o) the OQ Interrupt Message Number attribute, which contains the interrupt message number (see 5.2.5.12.12);
- p) the OQ Spanning Allowed attribute, which indicates whether spanning is allowed for this OQ (see 5.2.5.12.13); and
- q) the OQ Operational Queue Protocol attribute, which contains the operational queue protocol (see 10.2.5).

### 5.2.2.8 PQI Host Operational IQ Information class

The PQI host operational IQ information represents an instance of the PQI Host Operational IQ Information class (see figure 9).

Each instance of the PQI Host Operational IQ Information class (see figure 9) contains the following attributes that describe an instance of the Operational IQ class (see 5.2.5.9 and figure 13):

- a) the IQ ID attribute, which contains the IQ ID (see 5.2.5.9.2);
- b) the IQ Element Array Address attribute, which contains the IQ element array address (see 5.2.5.9.3);
- c) the IQ Number Of Elements attribute, which contains the number of elements in the IQ (see 5.2.5.9.5);

- d) the IQ Element Length attribute, which contains the length in bytes of the elements of the IQ (see 5.2.5.9.4);
- e) the IQ Spanning Allowed attribute, which indicates whether spanning is allowed for this IQ (see 5.2.5.9.8);
- f) the IQ PI Address attribute, which contains the IQ PI address (see 5.2.5.9.6);
- g) the IQ PI Local Copy attribute, which contains the IQ PI local copy (see 5.3.2.3);
- h) the IQ PI Working Copy attribute, which contains the IQ PI working copy (see 5.3.2.3);
- i) the IQ CI Address attribute, which contains the IQ CI address (see 5.2.5.9.7);
- j) the IQ CI Local Copy attribute, if any, which contains the IQ CI local copy (see 5.3.2.4);
- k) the IQ CI Working Copy attribute, if any, which contains the IQ CI working copy (see 5.3.2.4); and
- l) the IQ Operational Queue Protocol attribute, which contains the operational queue protocol (see 10.2.5).

### 5.2.3 PQI device classes

#### 5.2.3.1 PQI device classes overview

Figure 10 describes the PCI Express classes related to the PQI device.

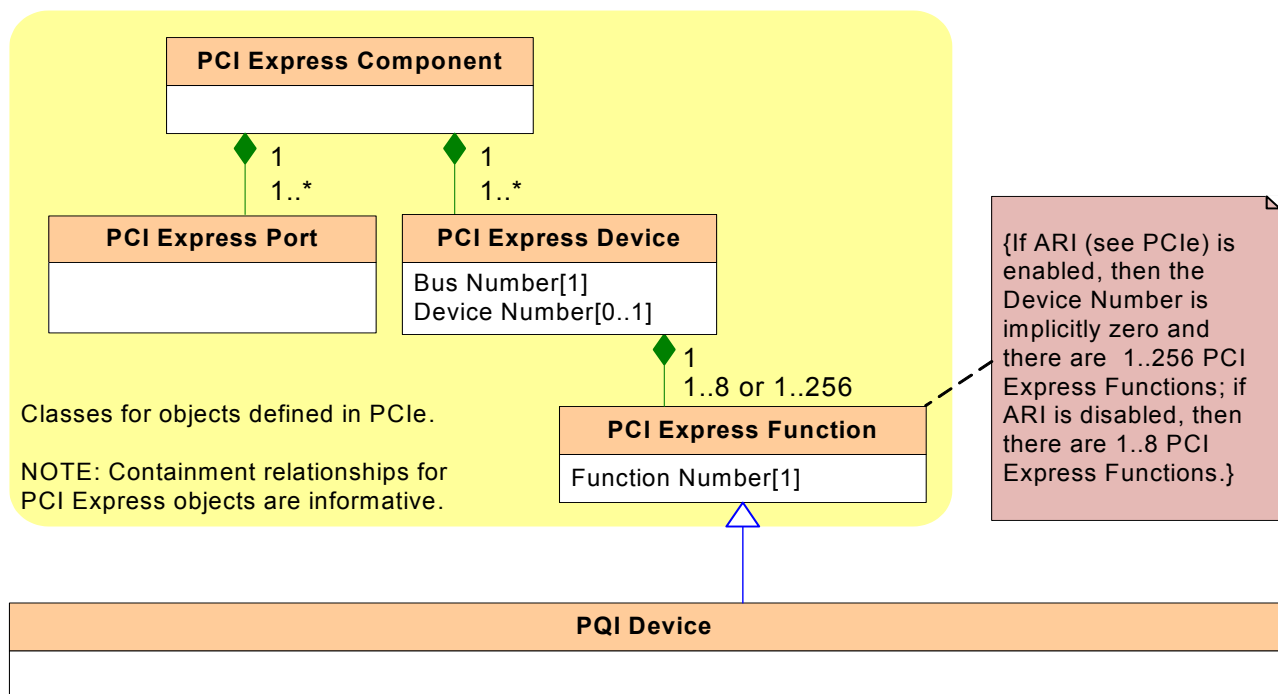


Figure 10 — PQI Device class and PCI Express classes

Figure 11 describes the classes related to the PQI device, showing the relationship between the following classes:

- a) PQI Device (see 5.2.3.2);
- b) OQ (see 5.2.5.10);
- c) CI (see 5.2.5.6);
- d) IQ (see 5.2.5.7);
- e) PI (see 5.2.5.5);
- f) PD State Machine (see 5.2.3.3);
- g) PQI Device Management Device Server (see 5.2.3.4);
- h) PQI Device Operational Device Server Interface (see 5.2.3.5);
- i) Administrator OQ CI (see 5.2.3.6);
- j) Administrator IQ PI (see 5.2.3.7);
- k) Operational OQ CI (see 5.2.3.8);
- l) Operational IQ PI (see 5.2.3.9);
- m) PQI Device Administrator OQ Information (see 5.2.3.10);
- n) PQI Device Operational OQ Information (see 5.2.3.11);
- o) PQI Device Administrator IQ Information (see 5.2.3.12); and
- p) PQI Device Operational IQ Information (see 5.2.3.13).

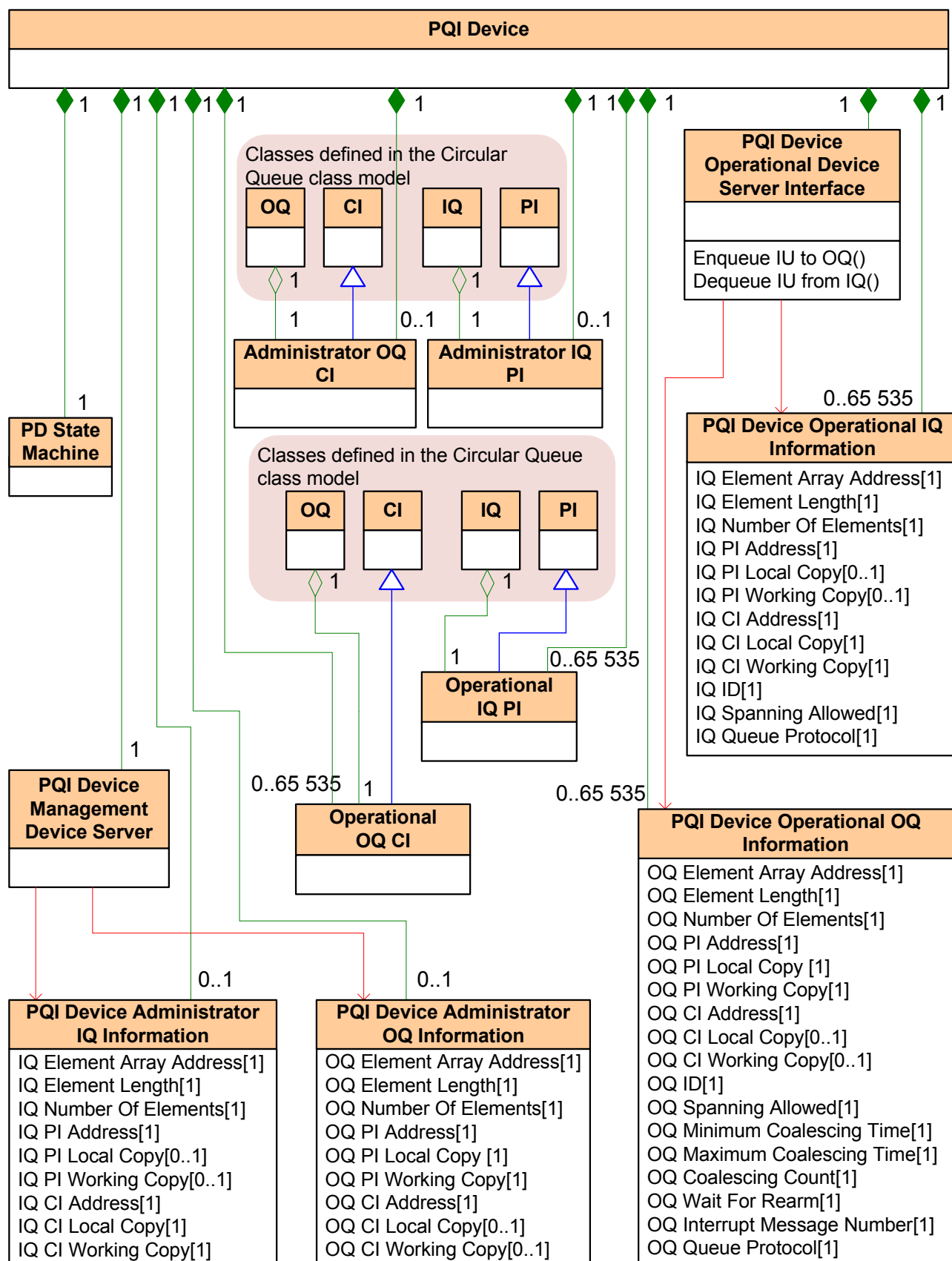


Figure 11 — PQI Device class model

### 5.2.3.2 PQI Device class

The PQI Device class (see figure 11) contains the following classes:

- a) PD State Machine (see 5.2.3.3);
- b) PQI Device Management Device Server (see 5.2.3.4);
- c) PQI Device Operational Device Server Interface (see 5.2.3.5);
- d) Administrator OQ CI (see 5.2.3.6);
- e) Administrator IQ PI (see 5.2.3.7);
- f) Operational OQ CI (see 5.2.3.8);
- g) Operational IQ PI (see 5.2.3.9);
- h) PQI Device Administrator OQ Information (see 5.2.3.10);
- i) PQI Device Operational OQ Information (see 5.2.3.11);
- j) PQI Device Administrator IQ Information (see 5.2.3.12); and
- k) PQI Device Operational IQ Information (see 5.2.3.13).

The PQI device represents an instance of the PQI Device class (see figure 11).

### 5.2.3.3 PD State Machine class

The PD state machine (see 5.5) represents an instance of the PD State Machine class (see figure 11).

### 5.2.3.4 PQI Device Management Device Server class

The PQI device management device server represents an instance of the PQI Device Management Device Server class (see figure 11).

A PQI device management device server performs administrator functions requested by the PQI host management application client.

### 5.2.3.5 PQI Device Operational Device Server Interface class

The PQI device operational device server interface represents an instance of the PQI Device Operational Device Server class (see figure 11).

A PQI device operational device server provides services to the device queuing layer interface defined in the IU layer standard (e.g., SOP) that include:

- a) enqueueing of IUs compliant with clause 9 to operational OQs;
- b) dequeuing of IUs compliant with clause 9 from operational IQs; and
- c) generating notifications of IUs on OQs available to be dequeued.

The PQI Device Operational Device Server Interface class processes Enqueue To OQ() requests (see 5.3.2.5.2) and Dequeue From IQ() (see 5.3.2.6.1) requests from the host queuing layer interface defined in the IU layer standard (e.g., SOP).

The PQI Device Operational Device Server Interface class generates IU Available On IQ operations to the device queuing layer interface defined in the IU layer standard (e.g., SOP).

### 5.2.3.6 Administrator OQ CI class

The Administrator OQ CI class (see figure 11) is a kind of CI class (see 5.2.5.6) and is part of the OQ class (see 5.2.5.10).

### 5.2.3.7 Administrator IQ PI class

The IQ PI class (see figure 11) is a kind of PI class (see 5.2.5.5) and is part of the IQ class (see 5.2.5.7).

### 5.2.3.8 Operational OQ CI class

The Operational OQ CI class (see figure 11) is a kind of CI class (see 5.2.5.6) and is part of the OQ class (see 5.2.5.10).

### 5.2.3.9 Operational IQ PI class

The Operational IQ PI class (see figure 11) is a kind of PI class (see 5.2.5.5) and is part of the IQ class (see 5.2.5.7).

### 5.2.3.10 PQI Device Administrator OQ Information class

The PQI device administrator OQ information represents an instance of the PQI Device Administrator OQ Information class (see figure 11).

Each instance of the PQI Device Administrator OQ Information class (see figure 11) contains the following attributes that describe an instance of the Administrator OQ class (see 5.2.5.11 and figure 13):

- a) the OQ Element Array Address attribute, which contains the OQ element array address (see 5.2.5.11.3);
- b) the OQ Element Length attribute, which contains the length in bytes of the elements of the OQ (see 5.2.5.11.4);
- c) the OQ Number Of Elements attribute, which contains the number of elements in the OQ (see 5.2.5.11.5);
- d) the OQ PI Address attribute, which contains the OQ PI address (see 5.2.5.11.6);
- e) the OQ PI Local Copy attribute, which contains the OQ PI local copy (see 5.3.2.3);
- f) the OQ PI Working Copy attribute, which contains the OQ PI working copy (see 5.3.2.3);
- g) the OQ CI Address attribute which, contains the OQ CI address in PQI device memory space (see 5.2.5.11.7);
- h) the OQ CI Local Copy attribute, if any, which contains the OQ CI local copy (see 5.3.2.4); and
- i) the OQ CI Working Copy attribute, if any, which contains the OQ CI working copy (see 5.3.2.4).

### 5.2.3.11 PQI Device Operational OQ Information class

The PQI device operational OQ information represents an instance of the PQI Device Operational OQ Information class (see figure 11).

Each instance of the PQI Device Operational OQ Information class (see figure 9) contains the following attributes that describe an instance of the Operational OQ class (see 5.2.5.12 and figure 13):

- a) the OQ Element Array Address attribute, which contains the OQ element array address (see 5.2.5.12.3);
- b) the OQ Element Length attribute which contains the length in bytes of the elements of the IQ (see 5.2.5.12.4);
- c) the OQ Number Of Elements attribute, which contains the number of elements in the IQ (see 5.2.5.12.5);
- d) the OQ PI Address attribute, which contains the OQ PI address (see 5.2.5.12.6);
- e) the OQ PI Local Copy attribute, which contains the OQ PI local copy (see 5.3.2.3);
- f) the OQ PI Working Copy attribute, which contains the OQ PI working copy (see 5.3.2.3);
- g) the OQ CI Address attribute, which contains the OQ CI address in PQI device memory space (see 5.2.5.12.7);
- h) the OQ CI Local Copy attribute, if any, which contains the OQ CI local copy (see 5.3.2.4);
- i) the OQ CI Working Copy attribute, if any, which contains the OQ CI working copy (see 5.3.2.4);
- j) the OQ ID attribute, which contains the OQ ID (see 5.2.5.12.2);
- k) the OQ Spanning Allowed attribute, which indicates whether spanning is allowed for this OQ and is equal to the value of the OUTBOUND SPANNING bit in the IU layer specific descriptor of the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3); and
- l) the OQ Queue Protocol attribute, which indicates the operational queue protocol (see table 81) that is supported by this operational OQ.

### 5.2.3.12 PQI Device Administrator IQ Information class

The PQI device administrator IQ information represents an instance of the PQI Device Administrator IQ Information class (see figure 11).

Each instance of the PQI Device Administrator IQ Information class (see figure 11) contains the following attributes that describe an instance of the Administrator IQ class (see 5.2.5.8 and figure 13):

- a) the IQ Element Array Address attribute, which contains the IQ element array address (see 5.2.5.8.2);
- b) the IQ Element Length attribute, which contains the length in bytes of the IQ element (see 5.2.5.8.3);

- c) the IQ Number Of Elements attribute, which contains the number of elements in the IQ (see 5.2.5.8.4);
- d) the IQ PI Address attribute, which contains the IQ PI address in PQI device memory space (see 5.2.5.8.5);
- e) the IQ PI Local Copy attribute, if any, which contains the IQ PI local copy (see 5.3.2.3);
- f) the IQ PI Working Copy attribute, if any, which contains the IQ PI working copy (see 5.3.2.3);
- g) the IQ CI Address attribute, which contains the IQ CI address in PQI device memory space (see 5.2.5.8.6);
- h) the IQ CI Local Copy attribute, which contains the IQ CI local copy (see 5.3.2.4); and
- i) the IQ CI Working Copy attribute, which contains the IQ CI working copy (see 5.3.2.4).

### 5.2.3.13 PQI Device Operational IQ Information class

The PQI device operational IQ information represents an instance of the PQI Device Operational IQ Information class (see figure 11).

Each instance of the PQI Device Operational IQ Information class (see figure 11) contains the following attributes that describe an instance of the Operational IQ class (see 5.2.5.9 and figure 13):

- a) the IQ Element Array Address attribute, which contains the IQ element array address (see 5.2.5.9.3);
- b) the IQ Element Length attribute, which contains the length in bytes of the elements of the IQ (see 5.2.5.9.4);
- c) the IQ Number Of Elements attribute, which contains the number of elements in the IQ (see 5.2.5.9.5);
- d) the IQ PI Address attribute, which contains the IQ PI address in PQI device memory space (see 5.2.5.9.6);
- e) the IQ PI Local Copy attribute, if any, which contains the IQ PI local copy (see 5.3.2.3);
- f) the IQ PI Working Copy attribute, if any, which contains the IQ PI working copy (see 5.3.2.3);
- g) the IQ CI Address attribute, which contains the IQ CI address (see 5.2.5.9.7);
- h) the IQ CI Local Copy attribute, which contains the IQ CI local copy (see 5.3.2.4);
- i) the IQ CI Working Copy attribute, which contains the IQ CI working copy (see 5.3.2.4);
- j) the IQ ID attribute, which contains the IQ ID (see 5.2.5.9.2);
- k) the IQ Spanning Allowed attribute, which indicates whether spanning is allowed for this IQ and is equal to the value of the INBOUND SPANNING bit in the IU layer specific descriptor of the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3); and
- l) the IQ Queue Protocol attribute, which indicates the operational queue protocol (see table 81) that is supported by this operational IQ.

## 5.2.4 PQI service delivery subsystem classes

### 5.2.4.1 PQI service delivery subsystem classes overview

Figure 12 describes the classes related to PQI service delivery subsystem, showing the relationship between the following classes:

- a) PQI Service Delivery Subsystem (see 5.2.4.2);
- b) PCI Express Fabric (see 5.2.4.3); and
- c) PQI Queue Structure (see 5.2.4.4).

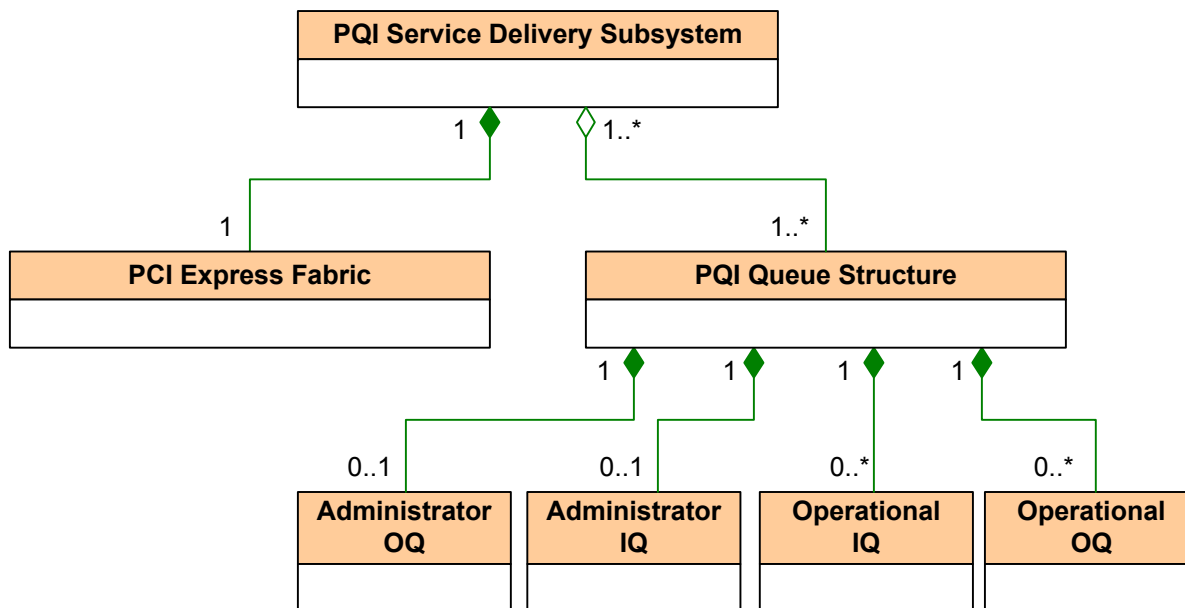


Figure 12 — PQI Service Delivery Subsystem class diagram

#### 5.2.4.2 PQI Service Delivery Subsystem class

Each instance of a PQI Service Delivery Subsystem class shall contain the following objects:

- a) one PCI Express fabric; and
- b) one or more PQI queue structures.

#### 5.2.4.3 PCI Express Fabric class

The PCI Express Fabric class connects the PQI Host and the PQI device in the PQI domain, providing a mechanism through which application clients communicate with device servers and task managers.

NOTE 4 - The PCI Express Fabric class includes backplane traces, cables, and PCIeExpress switches (see PCIe).

#### 5.2.4.4 PQI Queue Structure class

The PQI Queue Structure class (see figure 12) contains the following classes:

- a) Administrator IQ (see 5.2.5.8);
- b) Administrator OQ (see 5.2.5.11);
- c) Operational IQ (see 5.2.5.9); and
- d) Operational OQ (see 5.2.5.12).

Each instance of a PQI Queue Structure class (see figure 12) shall contain the following objects:

- a) zero or one Administrator IQ;
- b) zero or one Administrator OQ;
- c) zero or more Operational IQs; and
- d) zero or more Operational OQs.



## 5.2.5 Circular queue classes

### 5.2.5.1 Circular queue classes overview

Figure 13 describes the classes related to the circular queue, showing the relationship between the following classes:

- a) Circular Queue (see 5.2.5.2);
- b) Element Array (see 5.2.5.3);
- c) Element (see 5.2.5.4);
- d) PI (see 5.2.5.5);
- e) CI (see 5.2.5.6);
- f) IQ (see 5.2.5.7);
- g) Administrator IQ (see 5.2.5.8);
- h) Operational IQ (see 5.2.5.9);
- i) OQ (see 5.2.5.10);
- j) Administrator OQ (see 5.2.5.11); and
- k) Operational OQ (see 5.2.5.12).

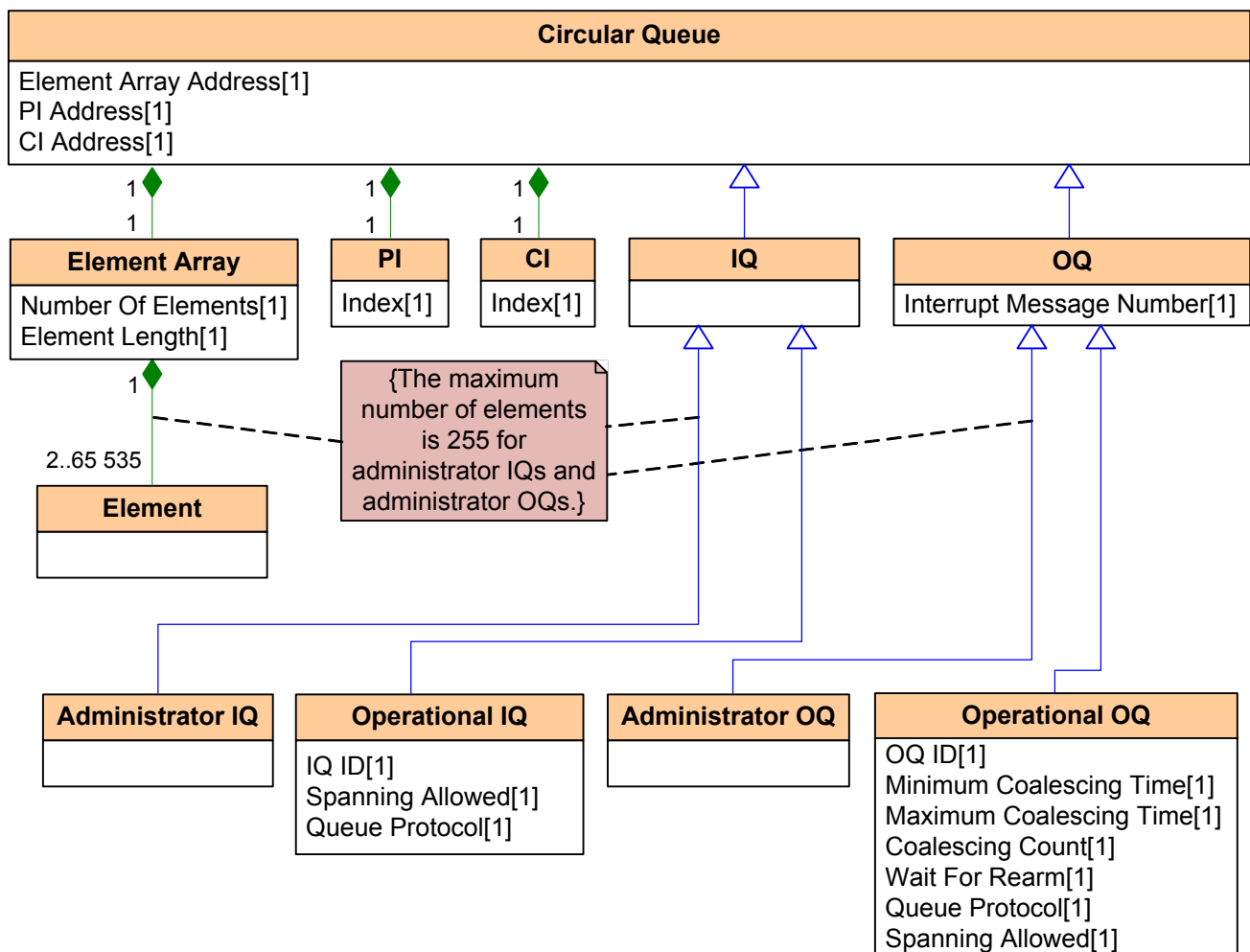


Figure 13 — Circular Queue classes

### **5.2.5.2 Circular Queue class**

#### **5.2.5.2.1 Circular Queue class overview**

The Circular Queue class (see figure 13) contains the following classes:

- a) Element Array (see 5.2.5.3);
- b) PI (see 5.2.5.5); and
- c) CI (see 5.2.5.6).

The Circular Queue class (see figure 13) may be substituted with:

- a) the Operational IQ class (see 5.2.5.9); or
- b) the Administrator IQ class (see 5.2.5.8).

For the Circular Queue class object rules see 5.2.5.4 and 5.3.2.1.

#### **5.2.5.2.2 Element Array Address attribute**

The Element Array Address attribute contains the memory space address of the first byte of the element array. For constraints on the Element Array Address attribute see:

- a) 5.2.5.8 for the Administrator IQ class constraints;
- b) 5.2.5.9 for the Operational IQ class constraints;
- c) 5.2.5.11 for the Administrator OQ class constraints; and
- d) 5.2.5.12 for the Operational OQ class constraints.

#### **5.2.5.2.3 PI Address attribute**

The PI Address attribute contains the memory space address of the PI.

For constraints on the PI Address attribute see:

- a) 5.2.5.8 for the Administrator IQ class constraints;
- b) 5.2.5.9 for the Operational IQ class constraints;
- c) 5.2.5.11 for the Administrator OQ class constraints; and
- d) 5.2.5.12 for the Operational OQ class constraints.

#### **5.2.5.2.4 CI Address attribute**

The CI Address attribute contains the memory space address of the CI.

For constraints on the CI Address attribute see:

- a) 5.2.5.8 for the Administrator IQ class constraints;
- b) 5.2.5.9 for the Operational IQ class constraints;
- c) 5.2.5.11 for the Administrator OQ class constraints; and
- d) 5.2.5.12 for the Operational OQ class constraints.

### **5.2.5.3 Element Array class**

#### **5.2.5.3.1 Element Array class overview**

The Element Array class contains the following class:

- a) Element (see 5.2.5.4).

#### **5.2.5.3.2 Number Of Elements attribute**

The Number Of Elements attribute contains the number of elements in the element array.

For constraints on the Number Of Elements attribute see:

- a) 5.2.5.8 for the Administrator IQ class constraints;
- b) 5.2.5.9 for the Operational IQ class constraints;
- c) 5.2.5.11 for the Administrator OQ class constraints; and
- d) 5.2.5.12 for the Operational OQ class constraints.

#### **5.2.5.3.3 Element Length attribute**

The Element Length attribute contains the length in bytes of an element in the element array (i.e., the element length). The element length shall be a multiple of 16 bytes.

For constraints on the Element Length attribute see:

- a) 5.2.5.8 for the Administrator IQ class constraints;
- b) 5.2.5.9 for the Operational IQ class constraints;
- c) 5.2.5.11 for the Administrator OQ class constraints; and
- d) 5.2.5.12 for the Operational OQ class constraints.

#### **5.2.5.4 Element class**

The Element class is one element in an element array.

Each instance of an Element class shall:

- a) contain from 16 to 1 048 560 bytes; and
- b) be an integer multiple of 16.

#### **5.2.5.5 PI class**

##### **5.2.5.5.1 PI class overview**

The PI class is an index of a circular queue's element array where the new entries are written by the producer (see 5.3.2.1).

##### **5.2.5.5.2 Index attribute**

The Index attribute specifies the contents of the PI.

#### **5.2.5.6 CI class**

##### **5.2.5.6.1 CI class overview**

The CI class is an index of a circular queue's element array where the new entries are read by the consumer (see 5.3.2.1).

##### **5.2.5.6.2 Index attribute**

The Index attribute specifies the contents of the CI.

#### **5.2.5.7 IQ class**

##### **5.2.5.7.1 IQ functional overview**

An IQ is a circular queue used to transfer IUs from a PQI host to a PQI device.

For an IQ:

- a) the IQ element array starts in memory space at the IQ element array address;
- b) the IQ CI is contained in the IQ CI dword (see 7.1.1) which is located at the IQ CI address; and
- c) the IQ PI is contained in the IQ PI register (see 7.1.2) which is located at the IQ PI address and shall be in PQI device memory space.

The PQI host maintains an IQ PI local copy.

The PQI device maintains an IQ CI local copy.

As specified by PCIe, the PQI device does not use PCI memory transactions to access:

- a) the IQ PI;
- b) an IQ element array that is located in PQI device memory space;
- c) an IQ CI that is located in PQI device memory space; and
- d) the PQI device memory space.

A PQI host shall not use the Relaxed Ordering ordering type (see PCIe) in memory transactions accessing the IQ element array, IQ CI, or IQ PI.

A PQI device shall not use the Relaxed Ordering ordering type in memory transactions accessing the IQ element array or IQ CI.

A PQI device shall use the ID-Based Ordering ordering type (see PCIe) in memory transactions accessing the IQ element array and IQ CI unless the IQ element array and IQ CI are not both located in PQI host memory space. The mechanism for determining the location of the IQ element array and IQ CI is outside the scope of this standard.

### 5.2.5.7.2 Example of IQ object locations that are not separated

Figure 14 shows an IQ example where:

- a) the IQ element array is in the PQI host memory space;
- b) the IQ CI is in the PQI host memory space; and
- c) the IQ PI is in the PQI device memory space.

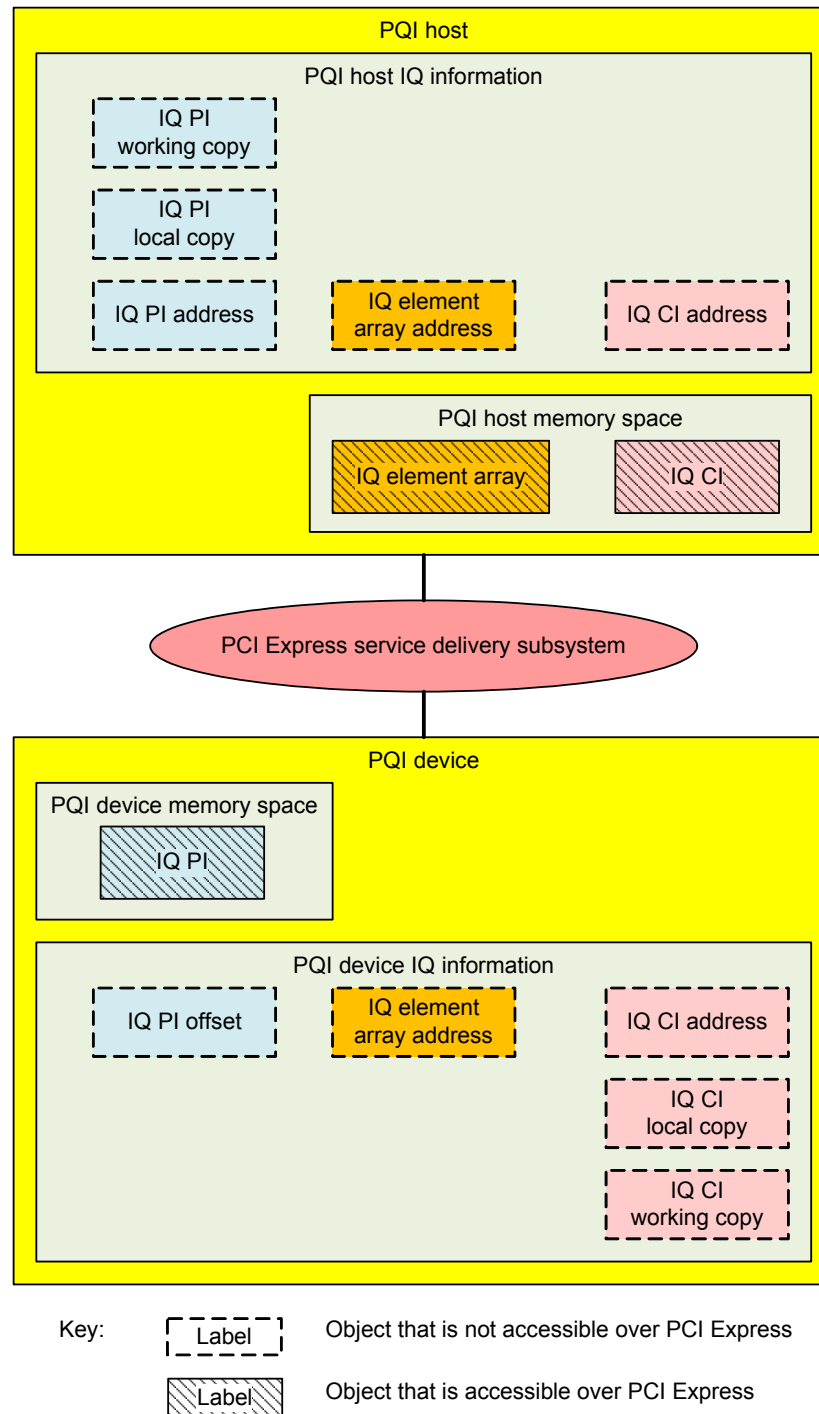


Figure 14 — Example of IQ object locations that are not separated

### 5.2.5.7.3 Example of IQ object locations that are separated

Figure 15 shows an IQ example where:

- the IQ element array is in the memory space of one PCI function;
- the IQ CI is in the memory space of another PCI function;
- the IQ PI is in the PQI device memory space; and
- the PQI host is separated from all of them.

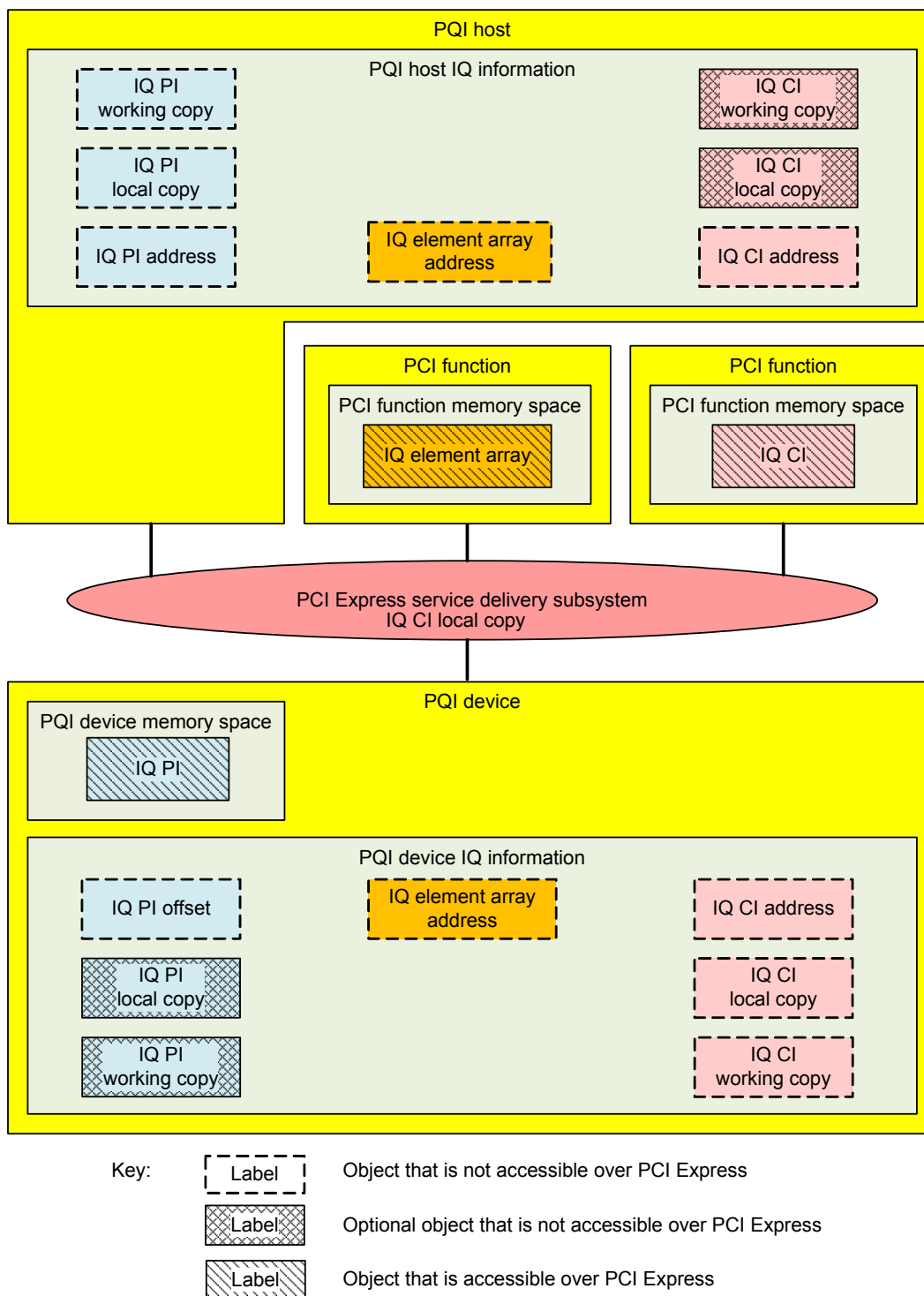


Figure 15 — Example where IQ object locations are separated

### 5.2.5.8 Administrator IQ class

#### 5.2.5.8.1 Administrator IQ class overview

The Administrator IQ class is a kind of IQ class (see 5.2.5.7).

#### 5.2.5.8.2 Element Array Address attribute

For an Administrator IQ class:

- a) the address alignment of the element array is 64 bytes; and
- b) the Element Array Address attribute is specified and indicated in the ADMINISTRATOR IQ ELEMENT ARRAY ADDRESS field in the Administrator IQ Element Array Address register (see 6.2.13).

#### 5.2.5.8.3 Element Length attribute

For an Administrator IQ class, the Element Length attribute is indicated in the ADMINISTRATOR IQ ELEMENT LENGTH field in the PQI Device Capability register (see 6.2.6).

#### 5.2.5.8.4 Number Of Elements attribute

For an Administrator IQ class the Number Of Elements attribute:

- a) is indicated in the NUMBER OF ADMINISTRATOR IQ ELEMENTS field in the Administrator Queue Parameter register (see 6.2.17);
- b) minimum value is two and maximum value is indicated in the MAXIMUM ADMINISTRATOR IQ ELEMENTS field in the PQI Device Capability register (see 6.2.6); and
- c) maximum value is less than or equal to 255.

#### 5.2.5.8.5 PI Address attribute

For an Administrator IQ class, the PI Address attribute is specified and indicated in the ADMINISTRATOR IQ PI OFFSET field in the Administrator IQ PI Offset register (see 6.2.11).

#### 5.2.5.8.6 CI Address attribute

For an Administrator IQ class:

- a) the address alignment of the IQ CI is 64 bytes; and
- b) the CI Address attribute is specified and indicated in the ADMINISTRATOR IQ CI ADDRESS field in the Administrator IQ CI Address register (see 6.2.15).

### 5.2.5.9 Operational IQ class

#### 5.2.5.9.1 Operational IQ class overview

The Operational IQ class is a kind of IQ class (see 5.2.5.7).

#### 5.2.5.9.2 IQ ID attribute

The IQ ID attribute contains the IQ ID (see 5.1).

The IQ ID attribute is:

- a) specified in the IQ ID field in the CREATE OPERATIONAL IQ request (see 10.2.4.1); and
- b) indicated in the IQ ID field in the operational IQ property descriptor in the REPORT OPERATIONAL IQ LIST parameter data (see 10.2.10.3).

The minimum value is one and the maximum value is 65 535.

#### 5.2.5.9.3 Element Array Address attribute

For an Operational IQ class:

- a) the minimum address alignment of the element array is 64 bytes;
- b) the Element Array Address attribute is specified in the IQ ELEMENT ARRAY ADDRESS field in the CREATE OPERATIONAL IQ request (see 10.2.4.1); and
- c) the Element Array Address attribute is indicated in the IQ ELEMENT ARRAY ADDRESS field in the operational IQ property descriptor in the REPORT OPERATIONAL IQ LIST parameter data (see 10.2.10.3).

#### 5.2.5.9.4 Element Length attribute

For an Operational IQ class, the Element Length attribute:

- a) is specified in the ELEMENT LENGTH field in the CREATE OPERATIONAL IQ request (see 10.2.4.1);
- b) is indicated in the ELEMENT LENGTH field in the operational IQ property descriptor in the REPORT OPERATIONAL IQ LIST parameter data (see 10.2.10.3); and
- c) minimum value is 16 bytes and maximum value is 1 048 560 (i.e.,  $16 \times 65\,535$ ) bytes.

#### 5.2.5.9.5 Number Of Elements attribute

For an Operational IQ class, the Number Of Elements attribute:

- a) is specified in the NUMBER OF ELEMENTS field in the CREATE OPERATIONAL IQ request (see 10.2.4.1);
- b) is indicated in the NUMBER OF ELEMENTS field in the operational IQ property descriptor in the REPORT OPERATIONAL IQ LIST parameter data (see 10.2.10.3);
- c) minimum value is two and maximum value is indicated in the MAXIMUM OPERATIONAL IQ ELEMENTS field in the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3); and
- d) the maximum value is less than or equal to 65 535.

#### 5.2.5.9.6 PI Address attribute

For an Operational IQ class, the PI Address attribute is:

- a) indicated by the PQI device;
- b) indicated in the IQ PI OFFSET field in the CREATE OPERATIONAL IQ response (see 10.2.4.2); and
- c) indicated in the IQ PI OFFSET field in the operational IQ property descriptor in the REPORT OPERATIONAL IQ LIST parameter data (see table 95).

#### 5.2.5.9.7 CI Address attribute

For an Operational IQ class:

- a) the minimum address alignment of the CI is four bytes;
- b) the CI Address attribute is specified in the IQ CI ADDRESS field in the CREATE OPERATIONAL IQ request (see 10.2.4.1); and
- c) the CI Address attribute is indicated in the IQ CI ADDRESS field in the operational IQ property descriptor in the REPORT OPERATIONAL IQ LIST parameter data (see 10.2.10.3).

#### 5.2.5.9.8 Spanning Allowed attribute

The Spanning Allowed attribute:

- a) indicates whether spanning is allowed for this operational IQ; and
- b) is reported in the INBOUND SPANNING bit in the IU layer specific descriptor of the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3).



#### 5.2.5.9.9 Queue Protocol attribute

The Queue Protocol attribute:

- a) indicates the operational queue protocol (see table 81) that is supported by this operational IQ; and
- b) is specified in the OPERATIONAL QUEUE PROTOCOL field in the CREATE OPERATIONAL IQ request (see 10.2.4.1).

#### 5.2.5.10 OQ class

##### 5.2.5.10.1 OQ functional overview

An OQ is a circular queue used to transfer IUs from a PQI device to a PQI host.

For an OQ:

- a) the OQ element array starts in memory space at the OQ element array address;
- b) the OQ PI is contained in the OQ PI dword (see 7.1.4) which is located at the OQ PI address; and
- c) the OQ CI is contained in the OQ CI register (see 7.1.3) which is located at the OQ CI address and is in PQI device memory space.

The PQI host maintains an OQ CI local copy.

The PQI device maintains an OQ PI local copy.

The PQI device:

- a) does not use memory transactions to access the OQ CI;
- b) does not use memory transactions to access an OQ element array or an OQ PI that is located in PQI device memory space; and
- c) does not use memory transactions to access the PQI device memory space.

A PQI host should not use the Relaxed Ordering ordering type (see PCIe) in memory transactions accessing the OQ element array, OQ CI, or OQ PI.

A PQI device shall not use the Relaxed Ordering ordering type in memory transactions accessing the OQ element array or OQ PI.

A PQI device shall use the ID-Based Ordering ordering type (see PCIe) in memory transactions accessing the OQ element array and OQ PI unless the OQ element array and OQ PI are not both located in PQI host memory space. The mechanism for determining the location of the OQ element array and OQ PI is outside the scope of this standard.

### 5.2.5.10.2 Example of OQ object locations that are not separated

Figure 16 shows an OQ example where:

- a) the OQ element array is in the PQI host memory space;
- b) the OQ PI is in the PQI host memory space; and
- c) the OQ CI is in the PQI device memory space.

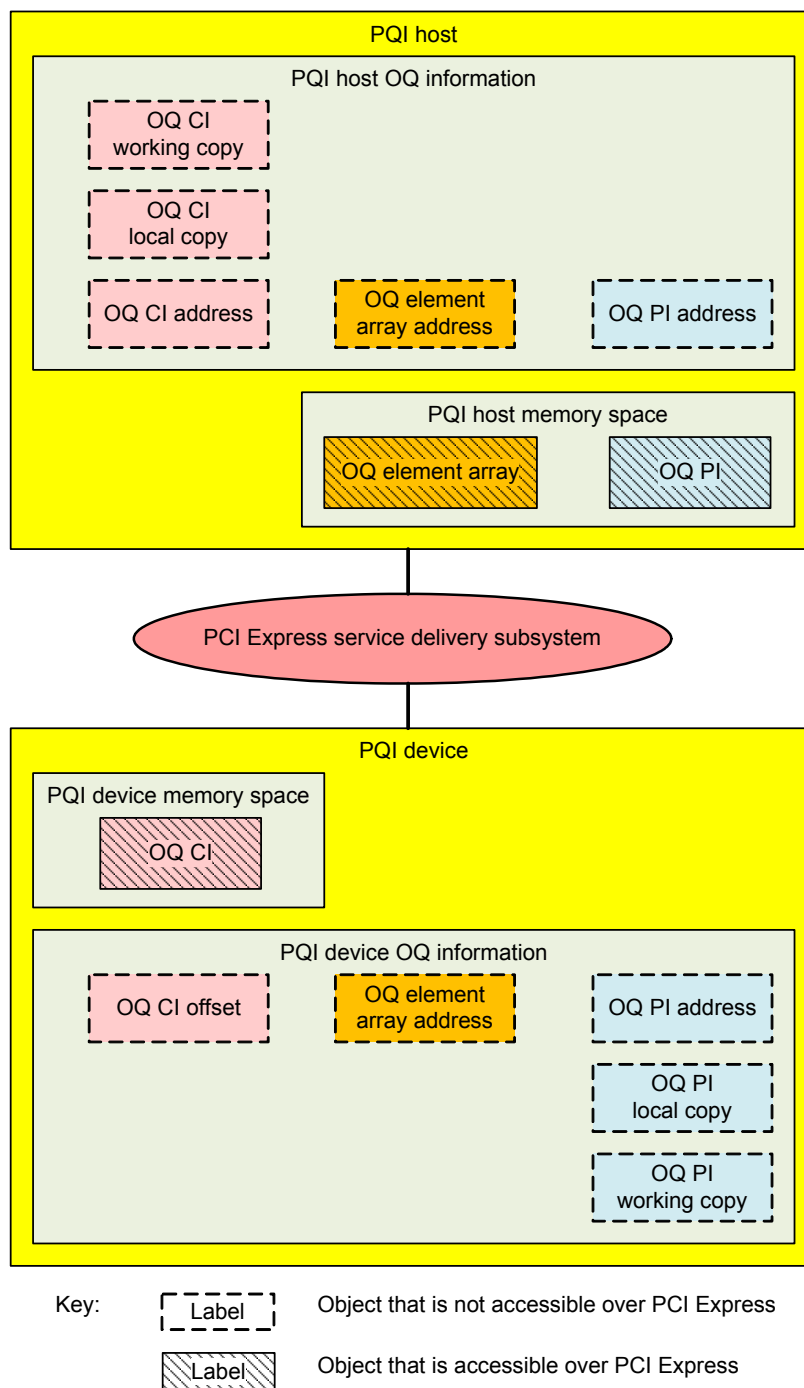


Figure 16 — Example of OQ object locations that are not separated

### 5.2.5.10.3 Example of OQ object locations that are separated

Figure 17 shows an OQ example where:

- the OQ element array is in the memory space of one PCI function;
- the OQ PI is in the memory space of another PCI function;
- the OQ CI is in the PQI device memory space; and
- the PQI host is separated from all of them.

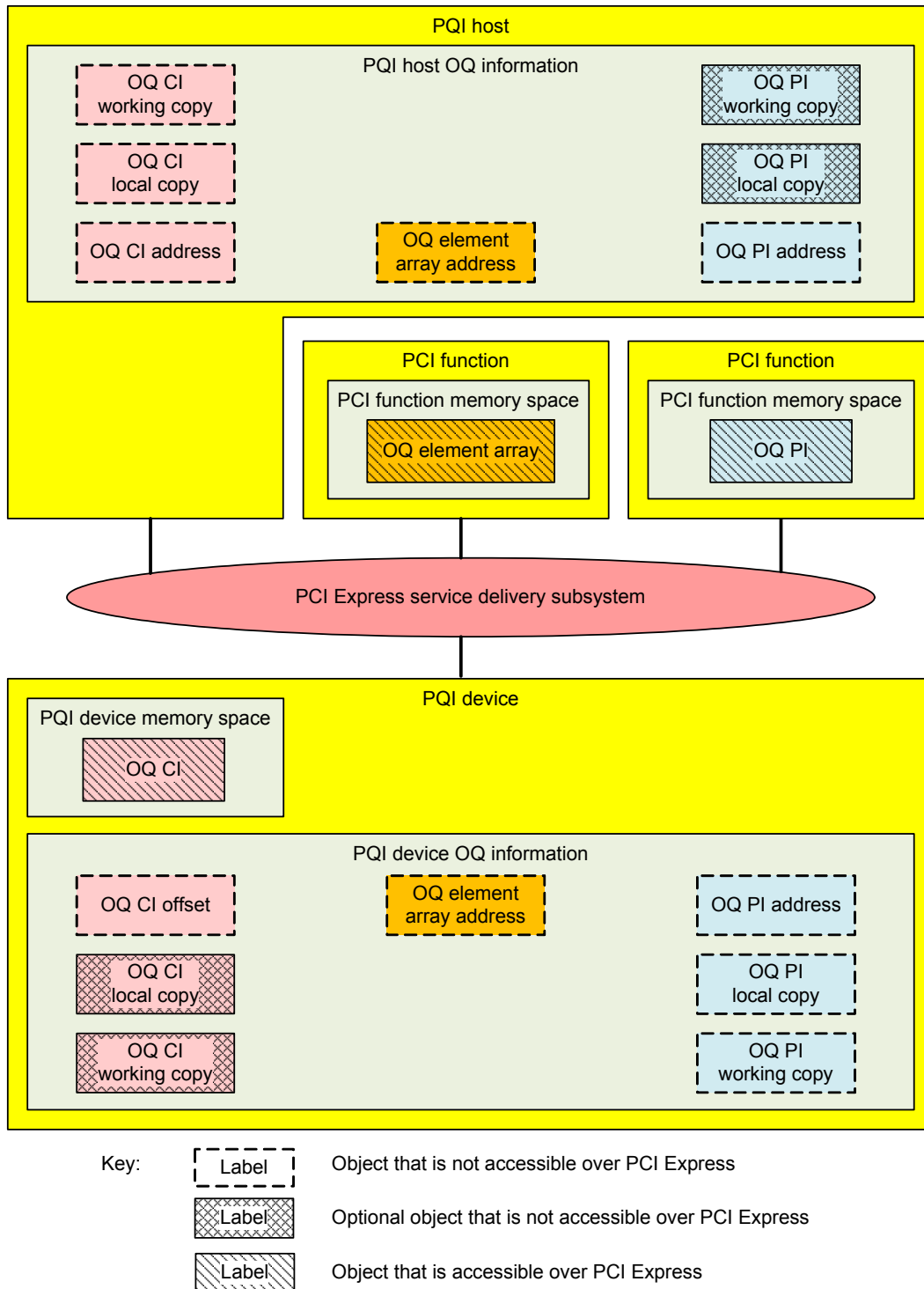


Figure 17 — Example of OQ object locations that are separated

#### 5.2.5.10.4 Interrupt Message Number attribute

The Interrupt Message Number attribute contains the interrupt message number (see PCIe).

The minimum value is zero and the maximum value is 2 047.

#### 5.2.5.11 Administrator OQ class

##### 5.2.5.11.1 Administrator OQ class overview

The Administrator OQ class is a kind of OQ class (see 5.2.5.10).

##### 5.2.5.11.2 Interrupt Message Number attribute

The Interrupt Message Number attribute is inherited from the OQ class (see 5.2.5.10).

The Interrupt Message Number attribute is:

- a) specified in the INTERRUPT MESSAGE NUMBER field in the Administrator Queue Parameter register (see 6.2.17) when the administrator queue pair are created; and
- b) indicated in the INTERRUPT MESSAGE NUMBER field in the Administrator Queue Parameter register while the PQI device is in the PD3:Administrator\_Queue\_Pair\_Ready state (see 5.5.5).

##### 5.2.5.11.3 Element Array Address attribute

For an administrator OQ:

- a) the address alignment of the element array is 64 bytes; and
- b) the Element Array Address attribute is specified and indicated in the ADMINISTRATOR OQ ELEMENT ARRAY ADDRESS field in the Administrator OQ Element Array Address register (see 6.2.14).

##### 5.2.5.11.4 Element Length attribute

For an Administrator OQ class, the Element Length attribute is indicated in the ADMINISTRATOR OQ ELEMENT LENGTH field in the PQI Device Capability register (see 6.2.6).

##### 5.2.5.11.5 Number Of Elements attribute

For an Administrator OQ class, the Number Of Elements attribute:

- a) is indicated in the NUMBER OF ADMINISTRATOR OQ ELEMENTS field in the Administrator Queue Parameter register (see 6.2.17);
- b) minimum value is two and maximum value is indicated in the MAXIMUM ADMINISTRATOR OQ ELEMENTS field in the PQI Device Capability register (see 6.2.6); and
- c) maximum value is less than or equal to 255.

##### 5.2.5.11.6 PI Address attribute

For an Administrator OQ class:

- a) the address alignment of the OQ PI is 64 bytes; and
- b) this attribute is specified and indicated in the ADMINISTRATOR OQ PI OFFSET field in the Administrator OQ PI Address register (see 6.2.16).

##### 5.2.5.11.7 CI Address attribute

For an Administrator OQ class, the CI Address attribute is indicated in the ADMINISTRATOR OQ CI ADDRESS field in the Administrator OQ CI Offset register (see 6.2.12).

### 5.2.5.12 Operational OQ class

#### 5.2.5.12.1 Operational OQ class overview

The Operational OQ class is a kind of OQ class (see 5.2.5.10).

#### 5.2.5.12.2 OQ ID attribute

The OQ ID attribute contains the OQ ID (see 5.1).

The OQ ID attribute is:

- a) specified in the OQ ID field in the CREATE OPERATIONAL OQ request (see 10.2.5.1); and
- b) indicated in the OQ ID field in the operational OQ property descriptor in the REPORT OPERATIONAL OQ LIST parameter data (see 10.2.11.3).

The minimum value is one and the maximum value is 65 535.

#### 5.2.5.12.3 Element Array Address attribute

For an Operational OQ class:

- a) the minimum address alignment of the element array is 64 bytes;
- b) the Element Array Address attribute is specified in the OQ ELEMENT ARRAY ADDRESS field in the CREATE OPERATIONAL OQ request (see 10.2.5.1); and
- c) the Element Array Address attribute is indicated in the OQ ELEMENT ARRAY ADDRESS field in the operational OQ property descriptor in the REPORT OPERATIONAL OQ LIST parameter data (see 10.2.11.3).

#### 5.2.5.12.4 Element Length attribute

For an Operational OQ class, the Element Length attribute is:

- a) specified in the ELEMENT LENGTH field in the CREATE OPERATIONAL OQ request (see 10.2.5.1);
- b) indicated in the ELEMENT LENGTH field in the operational OQ property descriptor in the REPORT OPERATIONAL OQ LIST parameter data (see 10.2.11.3); and
- c) minimum value is 16 bytes and maximum value is 1 048 560 (i.e.,  $16 \times 65\,535$ ) bytes.

#### 5.2.5.12.5 Number Of Elements attribute

For an Operational OQ class, the Number Of Elements attribute:

- a) is specified in the NUMBER OF ELEMENTS field in the CREATE OPERATIONAL OQ request (see 10.2.5.1);
- b) is indicated in the NUMBER OF ELEMENTS field in the operational OQ property descriptor in the REPORT OPERATIONAL OQ LIST parameter data (see 10.2.11.3);
- c) minimum value is two and maximum value is indicated in the MAXIMUM OPERATIONAL OQ ELEMENTS field in the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3); and
- d) maximum value is less than or equal to 65 535.

#### 5.2.5.12.6 PI Address attribute

For an Operational OQ class:

- a) the minimum address alignment of the PI is 4 bytes;
- b) this attribute is specified in the OQ PI ADDRESS field in the CREATE OPERATIONAL OQ request (see 10.2.5.1); and
- c) the PI Address attribute is indicated in the OQ PI ADDRESS field in the operational OQ property descriptor in the REPORT OPERATIONAL OQ LIST parameter data (see 10.2.11.3).

#### 5.2.5.12.7 CI Address attribute

For an Operational OQ class, the CI Address attribute is:

- a) indicated by the PQI device;
- b) indicated in the OQ CI OFFSET field in the CREATE OPERATIONAL OQ response (see 10.2.5.2); and
- c) indicated in the OQ CI OFFSET field in the operational OQ property descriptor in the REPORT OPERATIONAL OQ LIST parameter data (see 10.2.11.3).

#### 5.2.5.12.8 Minimum Coalescing Time attribute

The Minimum Coalescing Time attribute contains the minimum coalescing time in 100 ns intervals.

The Minimum Coalescing Time attribute:

- a) is specified in the MINIMUM COALESCING TIME field in the CREATE OPERATIONAL OQ request (see 10.2.5.1);
- b) is indicated in the MINIMUM COALESCING TIME field in the operational OQ property descriptor in the REPORT OPERATIONAL OQ LIST parameter data (see 10.2.11.3); and
- c) may be modified as specified in the MINIMUM COALESCING TIME field in the CHANGE OPERATIONAL OQ PROPERTIES request (see 10.2.9.1).

The minimum value is zero and the maximum value is 4 294 967 295 (i.e., FFFF\_FFFFh).

#### 5.2.5.12.9 Maximum Coalescing Time attribute

The Maximum Coalescing Time attribute contains the maximum coalescing time in 100 ns intervals.

The Maximum Coalescing Time attribute:

- a) is specified in the MAXIMUM COALESCING TIME field in the CREATE OPERATIONAL OQ request (see 10.2.5.1);
- b) is indicated in the MAXIMUM COALESCING TIME field in the operational OQ property descriptor in the REPORT OPERATIONAL OQ LIST parameter data (see 10.2.11.3); and
- c) may be modified as specified in the MAXIMUM COALESCING TIME field in the CHANGE OPERATIONAL OQ PROPERTIES request (see 10.2.9.1).

The minimum value is zero and the maximum value is 4 294 967 295 (i.e., FFFF\_FFFFh).

#### 5.2.5.12.10 Coalescing Count attribute

The Coalescing Count attribute contains the coalescing count.

The Coalescing Count attribute:

- a) is specified in the COALESCING COUNT field in the CREATE OPERATIONAL OQ request (see 10.2.5.1);
- b) is indicated in the COALESCING COUNT field in the operational OQ property descriptor in the REPORT OPERATIONAL OQ LIST parameter data (see 10.2.11.3); and
- c) may be modified as specified in the COALESCING COUNT field in the CHANGE OPERATIONAL OQ PROPERTIES request (see 10.2.9.1).

The minimum value is zero and the maximum value is 65 535.

#### 5.2.5.12.11 Wait For Rearm attribute

The Wait for Rearm attribute specifies the wait for rearm setting (see 5.4.2.3).

The Wait for Rearm attribute:

- a) is specified in the WAIT FOR REARM bit in the CREATE OPERATIONAL OQ request (see 10.2.5.1);
- b) is indicated in the WAIT FOR REARM bit in the operational OQ property descriptor in the REPORT OPERATIONAL OQ LIST parameter data (see 10.2.11.3); and

- c) may be modified as specified in the WAIT FOR REARM bit in the CHANGE OPERATIONAL OQ PROPERTIES request (see 10.2.9.1).

The minimum value is zero and the maximum value is one.

#### **5.2.5.12.12 Interrupt Message Number attribute**

The Interrupt Message Number attribute is inherited from the OQ class.

The Interrupt Message Number attribute is:

- a) specified in the INTERRUPT MESSAGE NUMBER field in the CREATE OPERATIONAL OQ request (see 10.2.5.1); and
- b) indicated in the INTERRUPT MESSAGE NUMBER field in the operational OQ property descriptor in the REPORT OPERATIONAL OQ LIST parameter data (see 10.2.11.3).

#### **5.2.5.12.13 Spanning Allowed attribute**

The Spanning Allowed attribute indicates whether spanning is allowed for this operational OQ and is reported in the OUTBOUND SPANNING bit in the IU layer specific descriptor of the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3).

#### **5.2.5.12.14 Queue Protocol attribute**

The Queue Protocol attribute:

- a) indicates the operational queue protocol (see table 81) that is supported by this operational OQ; and
- b) is specified in the OPERATIONAL QUEUE PROTOCOL field in the CREATE OPERATIONAL OQ request (see 10.2.5.1).

## 5.3 Queuing model

### 5.3.1 Queuing model overview

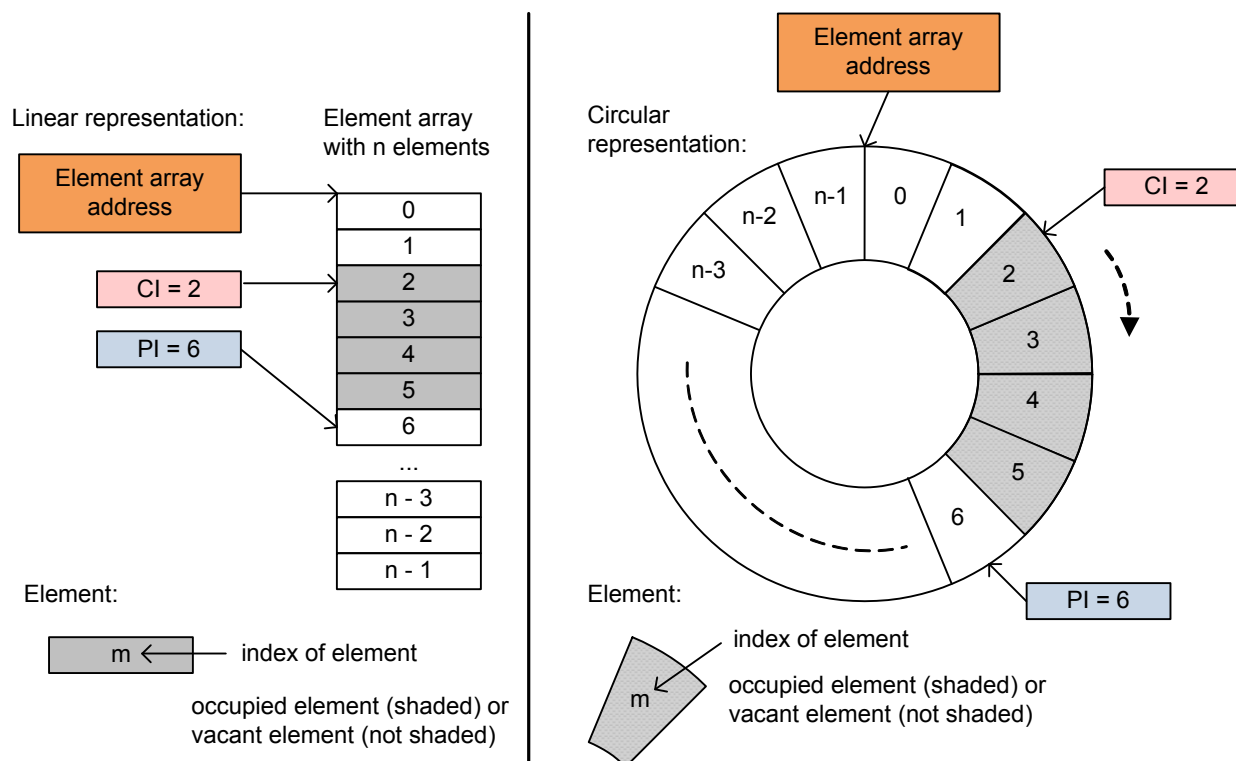
The queuing model defines:

- circular queue configuration and operations (see 5.3.2.1);
- circular queue PIs (see 5.3.2.1);
- circular queue CIs (see 5.3.2.1);
- circular queue elements within an element array (see 5.3.2.1); and
- IQ priority (see 5.3.5).

### 5.3.2 Circular queue model

#### 5.3.2.1 Circular queue functional overview

Figure 18 shows the circular queue model.



**Figure 18 — Circular queue**

A circular queue is created on request of a PQI host and the creation is completed on confirmation from the PQI device.

The element array contains a number of contiguous elements of fixed size. All elements within an element array are of the same size. The size and the number of elements in an element array are configured when the circular queue is created. For an element array of size  $n$ , the elements are indexed from zero to  $(n-1)$ .

The element array address is the address of the start of the first element in the element array. The address of an element is determined by multiplying the index value of the element by the element length and adding that value to the element array address.



An element within a circular queue is an occupied element if:

- a) the CI does not equal the PI; and
- b) the index of the element:
  - A) is equal to the index contained in the CI;
  - B) is greater than the CI and less than the PI and the CI is less than the PI;
  - C) is greater than the CI and the CI is greater than the PI; or
  - D) is less than the PI and the CI is greater than the PI.

All other elements within the circular queue are vacant elements.

A producer writes to vacant elements in an element array and a consumer reads from occupied elements in an element array.

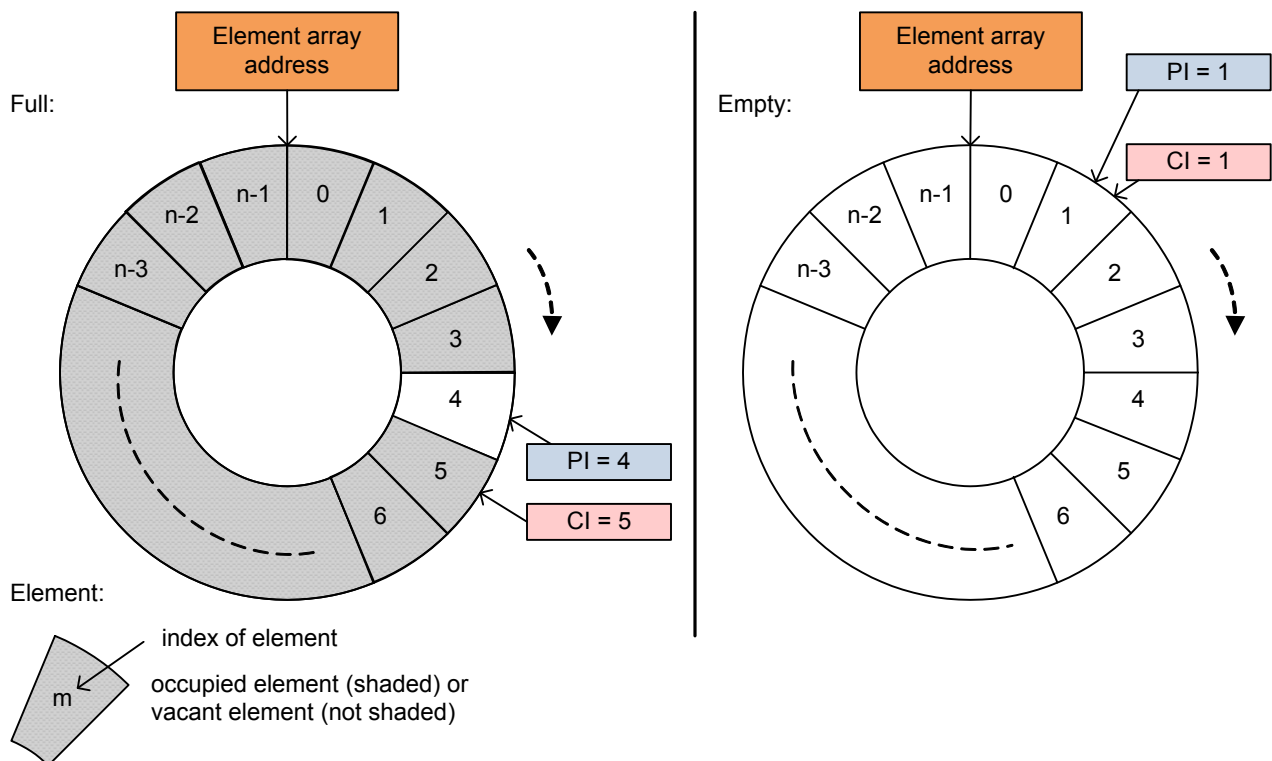
The PI contains the index of the next vacant element to be written by the producer. The initial value of the PI is zero.

If the CI does not equal the PI, then the CI contains the index of the next occupied element to be read by the consumer. The initial value of the CI is zero.

The circular queue is empty while the CI and PI are equal (i.e., all elements are vacant).

The circular queue is full when the PI is one behind the CI. There is always at least one vacant element in the element array. The maximum number of occupied elements in an element array of size  $n$  is  $n-1$ .

Figure 19 shows an example of a full circular queue with  $PI=4$  and  $CI=5$ , and an example of an empty circular queue with  $PI=1$  and  $CI=1$ .



**Figure 19 — Example of a full circular queue and an empty circular queue**

### 5.3.2.2 Relationship of IU to elements

A circular queue contains a number of elements.

A circular queue is used to transfer IUs. An IU may or may not fit within a single element of an element array.

For operational queues, the PQI host and PQI device may provide the capability to span an IU across multiple elements to support the transfer of IUs that exceed the size of an element. This capability is called spanning.

Spanning is not defined for administrator queues.

The support of spanning for operational queues is reported in the IU layer specific descriptors in the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3).

The maximum inbound IU length and the maximum outbound IU length for operational queues is reported in the IU layer specific descriptors in the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3).

If an IU to be produced exceeds the size of an element of the circular queue to which it is to be produced and the circular queue supports spanning, then the IU may be produced by dividing the IU into element sized pieces and placing these pieces into sequential elements of the queue.

If an IU is spanned, then each portion except possibly the last portion of the spanned IU shall be equal to the size of an element of the circular queue to which the IU is to be produced. If an IU or the last portion of a spanned IU is smaller than the size of an element, then the IU content is placed into the element at the first byte of the element and continues contiguously until the end of the IU is reached. The content of any portion of an element that does not contain a portion of the content of an IU is undefined and shall be ignored by the consumer of the IU.

If a consumer detects an IU whose length is greater than the specified maximum IU length, then the consumer shall:

- a) stop consuming from that operational queue; and
- b) if detected by the PQI device:
  - 1) set the OP IQ ERROR bit to one in the PQI Device Status register (see 6.2.10); and
  - 2) set the IQ ERROR bit to one in the Operational IQ property descriptor (see 10.2.10.3).

If the PQI device receives an administrator IU with the IU LENGTH field set to a value greater than the administrator OQ element length minus four, then:

- 1) the PQI device shall set the error code to INVALID IU LENGTH IN ADMINISTRATOR REQUEST IU in the PQI Device Error register (see 6.2.18); and
- 2) the PD state machine transitions as described in 5.5.5.5.

This standard does not define specific reasons for a PQI device to stop producing to an OQ as a result of errors.

A producer shall not request to produce an IU that is larger than the number of elements in the circular queue minus one, multiplied by the size of an element of the queue. The handling of this error condition is vendor specific.

### 5.3.2.3 Queue producer

The producer writes one or more IUs to vacant elements starting at the element indicated by the PI and continuing sequentially up to and including the element indicated by CI minus two, wrapping at the end of the element array if necessary. After writing one or more IUs to the element array, the producer increments PI by the number of occupied elements produced, modulo the number of elements in the array.

While writing to vacant elements in a circular queue, the producer maintains two copies of the PI:

- a) a PI local copy (i.e., a mirror of the value last written to the PI); and
- b) a PI working copy (i.e., a variable containing the index of the next vacant element in the element array).

While writing one or more IUs to vacant elements in a circular queue, the producer compares the PI working copy to the CI. The producer, after producing each IU:

- a) may update the PI and the PI local copy to the value of the PI working copy; and

- b) should update the PI and the PI local copy to the value of the PI working copy if the CI is approaching or equal to the PI local copy (i.e., the consumer considers the circular queue to be almost empty or empty).

NOTE 5 - Since the consumer is consuming IUs from the circular queue at the same time as the producer is producing IUs to the circular queue, regular PI updates keep the consumer from detecting that the circular queue is empty and allow the consumer to continue consuming from the circular queue. The algorithm for determining the threshold is vendor specific.

NOTE 6 - There is a benefit in reducing the number of memory write transactions used to update the PI by producing multiple IUs and updating the IQ PI with a single memory write transaction.

A producer should use the same Traffic Class (see PCIe) for memory write transactions to the PI as it uses for memory write transactions to the element array.

#### 5.3.2.4 Queue consumer

To read IUs, the consumer reads occupied elements starting from the element indicated by the CI and continues sequentially wrapping at the end of the element array. After reading from one or more IUs, the consumer increments CI by the number of occupied elements consumed, wrapping if the value equals the number of elements.

While reading from elements in a circular queue, the consumer maintains two copies of the CI:

- a) a CI local copy (i.e., a mirror of the value last written to the CI); and
- b) a CI working copy (i.e., a variable containing the index of the next element in the element array to be read).

While reading one or more IUs in the circular queue, the consumer compares the CI working copy to the PI. The consumer, after consuming each IU:

- a) may update the CI and the CI local copy to the value of the CI working copy;
- b) should update CI and the CI local copy to the value of the CI working copy if the PI is approaching the CI local copy (i.e., the producer considers the circular queue to be almost full or to be full); and
- c) shall update CI and the CI local copy to the value of the CI working copy if the CI working copy is equal to the PI (i.e., the consumer considers the circular queue to be empty).

To prevent deadlocks where the producer and consumer are both waiting on each other to update indexes, the consumer shall not refuse to update the CI because the consumer is waiting for the producer to update the PI.

NOTE 7 - Since the producer is producing IUs to the circular queue at the same time as the consumer is consuming IUs from the circular queue, regular CI updates keep the producer from detecting that the circular queue is full and allows the producer to continue producing to the circular queue. The algorithm for determining the threshold is vendor specific.

NOTE 8 - There is a benefit in reducing the number of memory write transactions by reading multiple IUs and updating the IQ CI with a single memory write transaction.

A consumer should use the same Traffic Class (see PCIe) for memory write transactions to the CI as it uses for memory read transactions from the element array.

The consumer should not perform memory read transactions of a vacant element.

#### 5.3.2.5 Enqueue operation

##### 5.3.2.5.1 Enqueue To IQ operation overview

The Enqueue To IQ operation is modeled by the following procedure call:

**Enqueue To IQ (IN (IQ ID, Number of IUs, IUs))**

The Enqueue To IQ operation is supported by the PQI operational application client Interface in the PQI host to enqueue IUs to the specified IQ.

Input arguments:

**IQ ID:** The IQ to which the IU is to be produced.

**Number of IUs:** The number of IUs to be produced.

**IUs:** The IUs to be produced.

The Enqueue To IQ operation produces one or more IUs to the element array of the IQ and increments the associated PI, wrapping at the end of the element array. The Enqueue To IQ operation may enqueue an IU if there are sufficient vacant elements for that IU without requiring sufficient vacant elements for all IUs specified in the Enqueue To IQ request (see 5.3.2.5.3).

#### 5.3.2.5.2 Enqueue To OQ operation overview

The Enqueue To OQ operation is modeled by the following procedure call:

**Enqueue To OQ (IN (OQ ID, Number of IUs, IUs))**

The Enqueue To OQ operation is supported by the PQI device operational device server interface in the PQI device to enqueue IUs to the specified OQ.

Input arguments:

**OQ ID:** The OQ to which the IU is to be produced.

**Number of IUs:** The number of IUs to be produced.

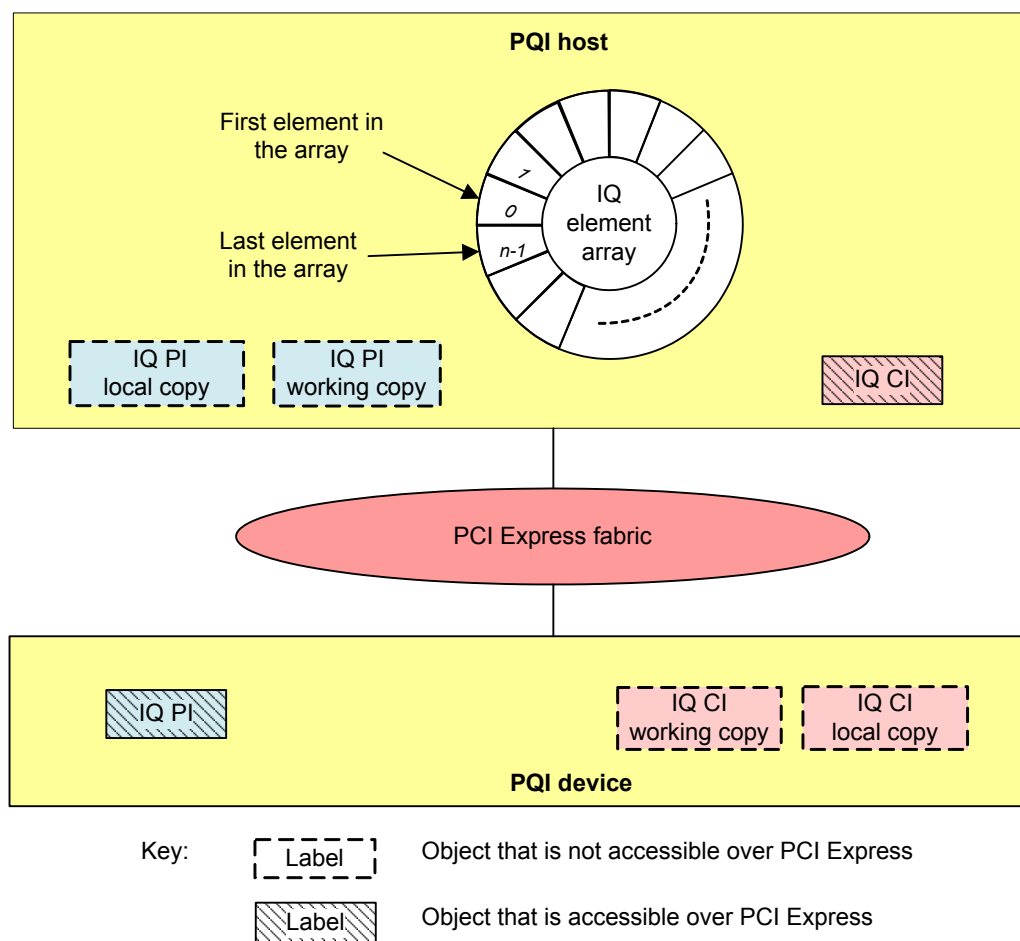
**IUs:** The IUs to be produced.

The Enqueue To OQ operation produces one or more IUs to the element array of the OQ and increments the associated PI, wrapping at the end of the element array. The Enqueue To OQ operation may enqueue an IU if there are sufficient vacant elements for that IU without requiring sufficient vacant elements for all IUs specified in the Enqueue To OQ request (see 5.3.2.5.4).

#### 5.3.2.5.3 PQI host enqueueing IUs to an IQ

The PQI host determines the number of vacant IQ elements available to contain new IUs by comparing the IQ CI and the IQ PI local copy. The number of elements available to contain new IUs is equal to the number of vacant elements in the IQ minus one.

Figure 20 shows an example location of the IQ CI and the IQ PI local copy used by the PQI host to determine the number of vacant IQ elements.



**Figure 20 — Example location of IQ CI and PQI host IQ PI local copy**

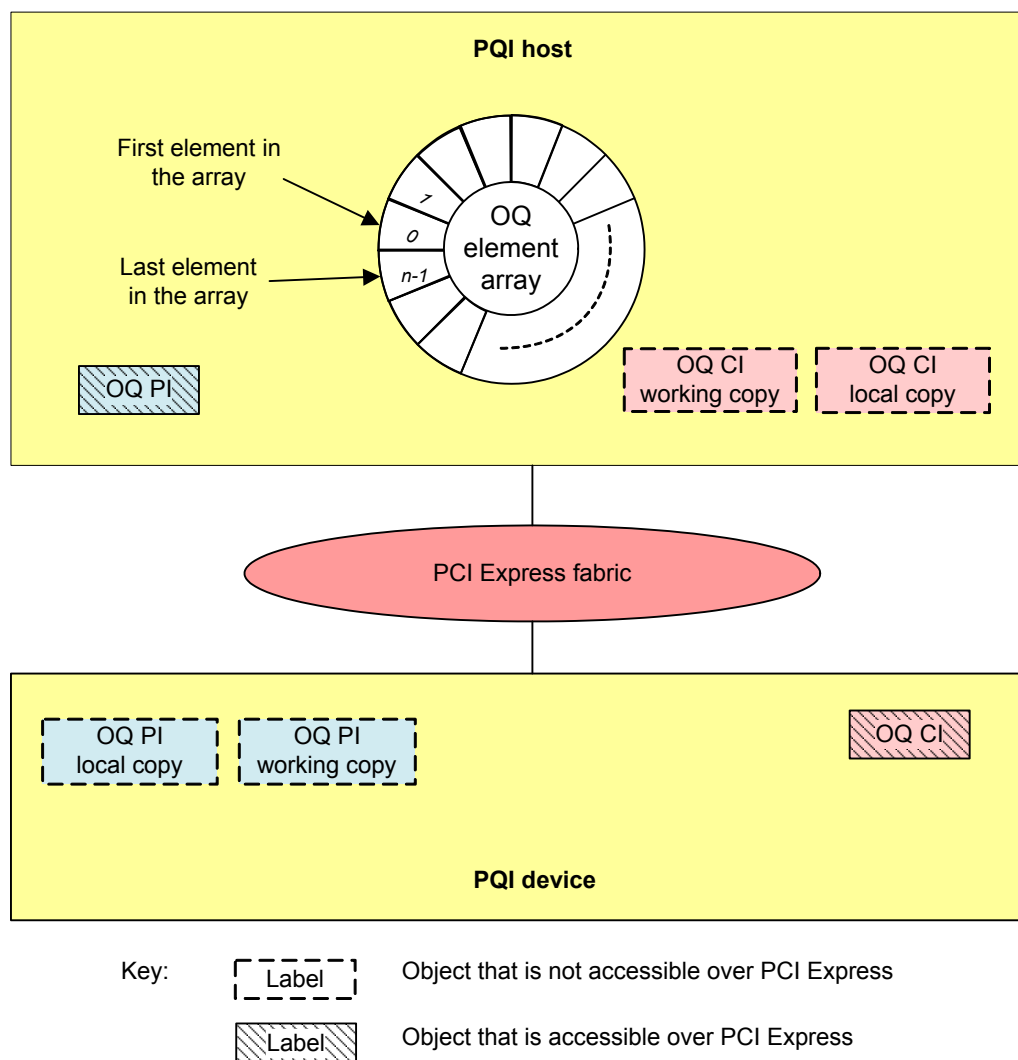
The PQI host may produce one or more IUs to the IQ if the total number of elements required by the IUs to be produced is less than the number of vacant IQ elements. The PQI host should use the following steps to produce IUs to the IQ:

- 1) for each IU to be produced:
  - 1) write the IU into the IQ element array beginning at the address indicated by the IQ PI working copy using wrapping as described in 5.3.2.1; and
  - 2) update the IQ PI working copy;
- 2) update the IQ PI local copy with the IQ PI working copy; and
- 3) update the IQ PI with the IQ PI working copy.

#### 5.3.2.5.4 PQI device enqueueing IUs to an OQ

The PQI device determines the number of vacant OQ elements available to contain new IUs by comparing the OQ CI and the OQ PI local copy. The number of elements available to contain new IUs is equal to the number of vacant elements in the IQ minus one.

Figure 21 shows an example location of the OQ CI and the OQ PI local copy used by the PQI device to determine the number of vacant OQ elements.



**Figure 21 — Example location of OQ CI and PQI device OQ PI local copy**

The PQI device may produce one or more IUs to the OQ. If the total number of elements required by the IUs to be produced is less than or equal to the number of vacant OQ elements minus one, then the PQI device shall use the following steps to produce IUs to the OQ:

- 1) for each IU to be produced:
  - 1) write the IU into the OQ element array beginning at the address indicated by the OQ PI working copy using wrapping as described in 5.3.2.1; and
  - 2) update the OQ PI working copy;
- 2) update the OQ PI local copy with the OQ PI working copy; and
- 3) update the OQ PI with the OQ PI working copy.

If the IU layer requests that the PQI device send an administrator IU with the IU LENGTH field value larger than the administrator OQ element length, then:

- a) the PQI device shall set the error code to OQ SPANNING CONFLICT; and
- b) the PD state machine transitions as described in 5.5.5.5.

### 5.3.2.6 Dequeue operation

#### 5.3.2.6.1 Dequeue From IQ operation overview

The Dequeue From IQ operation is modeled by the following procedure call:

**Dequeue From IQ (IN (IQ ID, Number of IUs), OUT (IUs))**

Input arguments:

**IQ ID:** The IQ from which the IU is to be consumed.

**Number of IUs:** The number of IUs to be consumed.

Output arguments:

**IUs:** The IUs consumed.

The Dequeue From IQ operation returns one or more IUs. The Dequeue From IQ operation waits for the circular queue to have the specified number of IUs to be consumed (see 5.3.2.6.4).

#### 5.3.2.6.2 Dequeue From OQ operation overview

The Dequeue From OQ operation is modeled by the following procedure call:

**Dequeue From OQ (IN (OQ ID, Number of IUs), OUT (IUs))**

Input arguments:

**OQ ID:** The OQ from which the IU is to be consumed.

**Number of IUs:** The number of IUs to be consumed.

Output arguments:

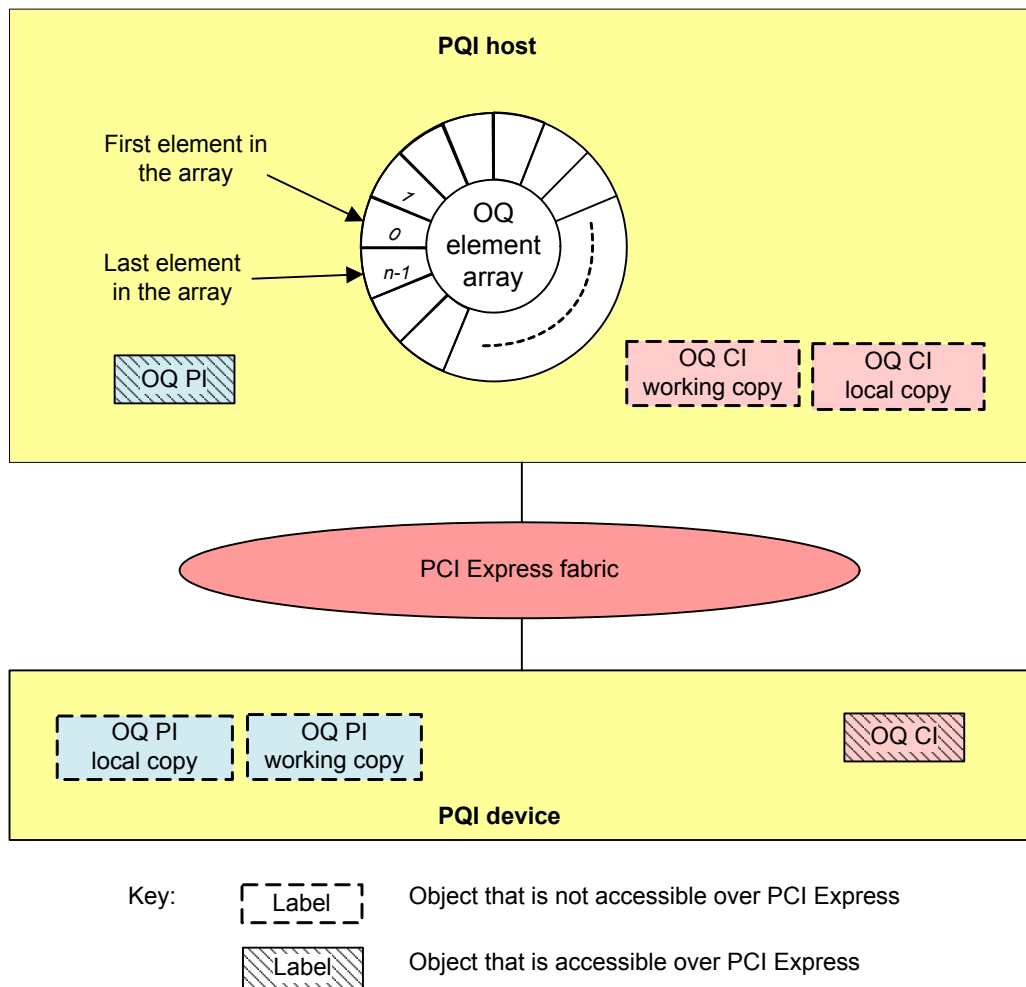
**IUs:** The IUs consumed.

The Dequeue From OQ operation returns one or more IUs. The Dequeue From OQ operation waits for the circular queue to have the specified number of IUs to be consumed (see 5.3.2.6.3).

#### 5.3.2.6.3 PQI host dequeuing IUs from an OQ

The PQI host determines the number of occupied OQ elements by comparing the OQ PI and the OQ CI local copy.

Figure 22 shows an example location of the OQ PI and the OQ CI local copy used by the PQI host to determine the number of occupied OQ elements.



**Figure 22 — Example location of OQ PI and PQI host OQ CI local copy**

The PQI host may consume one or more IUs from the OQ using the following steps:

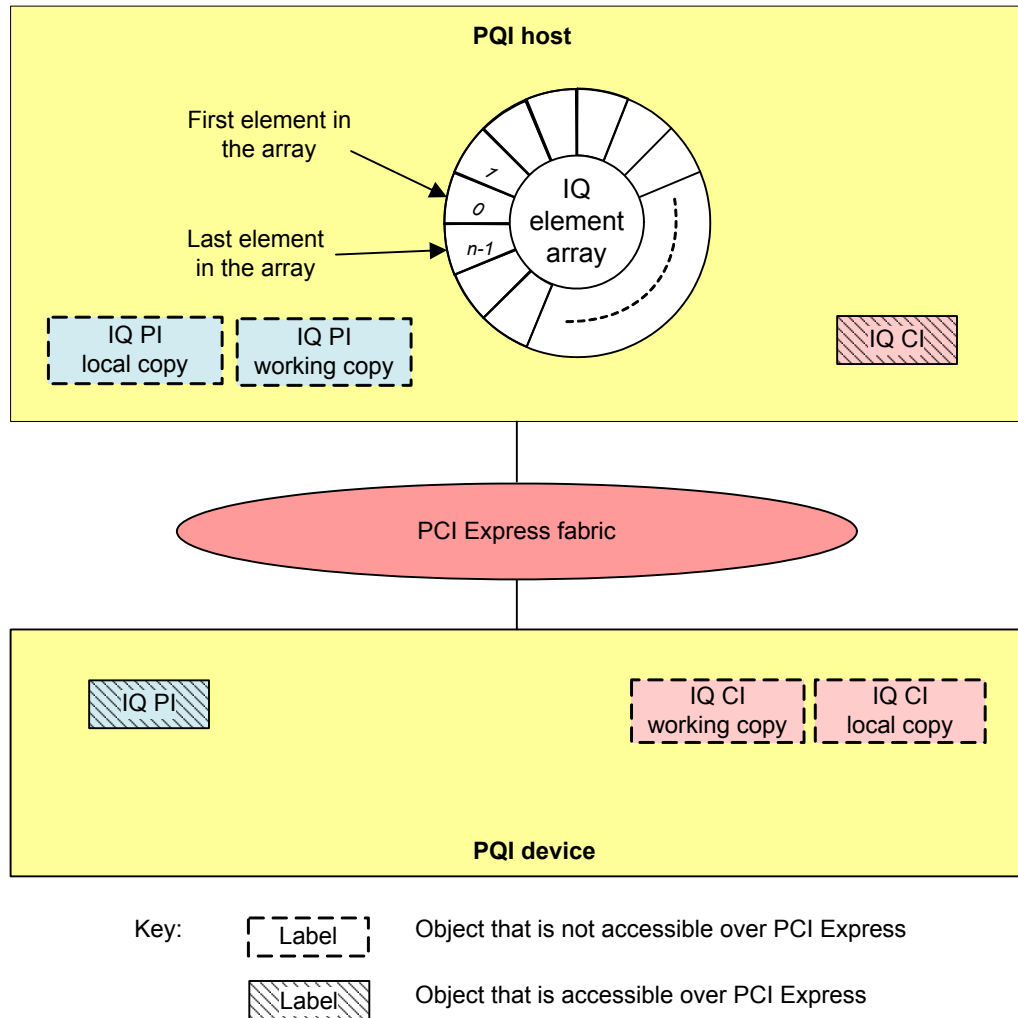
- 1) for each IU to be consumed:
  - 1) read the IU from the OQ element array beginning at the address indicated by the OQ CI working copy, using wrapping as described in 5.3.2.1; and
  - 2) update the OQ CI working copy;
- 2) update the OQ CI local copy with the OQ CI working copy; and
- 3) update the OQ CI with the OQ CI working copy.



#### 5.3.2.6.4 PQI device dequeuing IUs from an IQ

The PQI device determines the number of occupied elements of an IQ by comparing the IQ PI and the IQ CI local copy.

Figure 23 shows an example location of the IQ PI and the PQI device IQ CI local copy used by the PQI device to determine the number of available IQ elements.



**Figure 23 — Example location of IQ PI and PQI device IQ CI local copy**

The PQI device shall use the following steps to consume IUs from the IQ:

- 1) for each IU to be consumed:
  - 1) read the IU from the IQ element array beginning at the address indicated by the IQ CI working copy, using wrapping as described in 5.3.2.1; and
  - 2) update the IQ CI working copy;
- 2) update the IQ CI local copy with the IQ CI working copy; and
- 3) update the IQ CI with the IQ CI working copy.

### 5.3.2.7 IU Available notification

#### 5.3.2.7.1 IU Available On IQ operation overview

The IU Available On IQ operation is modeled by the following procedure call:

##### **IU Available On IQ (OUT (IQ ID))**

The IU Available On IQ operation call is invoked by the PQI device operational device server interface in the PQI device.

Output arguments:

**IQ ID:** The operational IQ<sub>x</sub> which contains one or more IUs.

Once an IU Available On IQ operation is invoked for an operational IQ, another IU Available On IQ operation should not be invoked until a Dequeue IU From IQ operation specifying that queue is invoked.

#### 5.3.2.7.2 IU Available On OQ operation overview

The IU Available On OQ operation is modeled by the following procedure call:

##### **IU Available On OQ (OUT (OQ ID))**

The IU Available On OQ operation call is invoked by the PQI host operational application client interface in the PQI host.

Output arguments:

**OQ ID:** The operational OQ<sub>x</sub> which contains one or more IUs.

Once an IU Available On OQ operation is invoked for an operational OQ, another IU Available On OQ operation should not be invoked until a Dequeue IU From OQ operation specifying that queue is invoked.

#### 5.3.2.7.3 IU Available On OQ notification

After a PQI device produces an IU to an operational OQ, the PQI device sends an OQ service notification (see 5.4). After a PQI host receives an OQ service notification, the PQI host should examine the PI and CI of the operational OQ to determine if there are one or more occupied elements available in the operational OQ.

After the PQI host operational application client interface completes the processing of a Dequeue From OQ request, the PQI host operational application client interface should examine the current values of the PI and CI of the operational OQ to determine if the operational OQ contains any occupied elements.

If an operational OQ contains occupied elements, then that operational OQ contains at least one IU.

If an operational OQ contains occupied elements and an IU Available On OQ operation for which a corresponding Dequeue From OQ request has not been received is not pending, then an IU Available On OQ operation should be invoked.

A PQI host operational application client interface may limit the number of parallel IU Available On OQ operations that are outstanding.

A PQI host operational application client interface may assign vendor-specific priority levels to operational OQs and invoke IU Available On OQ operations based upon such priority levels.

#### 5.3.2.7.4 IU Available On IQ notification

After a PQI host produces an IU to an operational IQ, the PQI host updates the PI for the operational IQ. The detection of an update to the PI of an operational IQ serves as notification to the PQI device that an IU may be available on the operational IQ. The PQI host should examine the PI and CI of the associated operational IQ to determine if there are one or more occupied elements available in the operational IQ.

After the PQI operational device server completes the processing of a Dequeue From IQ request, the PQI device operational device server interface should examine the current values of the PI and CI of the operational IQ to determine if the operational IQ contains any occupied elements.

If a operational IQ contains occupied elements, then that operational IQ contains at least one IU.

If an operational IQ contains occupied elements and an IU Available On IQ operation for which a corresponding Dequeue From IQ request has not been received is not pending, then an IU Available On IQ operation should be invoked.

A PQI device operational device server interface may limit the number of parallel IU Available On IQ operations that are outstanding.

A PQI device operational device server interface may assign vendor-specific priority levels to IQs (see 5.3.5) and invoke IU Available On IQ operations based upon such priority levels.

### **5.3.3 Creating circular queues**

#### **5.3.3.1 Creating circular queues overview**

The PQI management application client initiates the creation of the administrator queue pair by writing to the Administrator Queue Configuration Function register (see 5.3.3.2).

The PQI management application client initiates the creation of an operational IQ by:

- a) allocating resources for the element array to be used for the operational IQ;
- b) allocating resources for the IQ CI;
- c) constructing a CREATE OPERATIONAL IQ request; and
- d) enqueueing the IU to the administrator IQ.

The PQI management device server processes the CREATE OPERATIONAL IQ request by:

- a) allocating resources for the IQ PI;
- b) constructing a CREATE OPERATIONAL IQ response; and
- c) enqueueing the response to the administrator OQ.

The PQI management application client initiates the creation of an operational OQ by:

- a) allocating resources for the element array to be used for the operational OQ;
- b) allocating resources for the OQ PI;
- c) constructing a CREATE OPERATIONAL OQ request; and
- d) enqueueing the IU to the administrator IQ.

The PQI management device server processes the CREATE OPERATIONAL OQ request by:

- a) allocating resources for the OQ CI;
- b) constructing a CREATE OPERATIONAL OQ response; and
- c) enqueueing the response to the administrator OQ.

After the PQI management application client dequeues the CREATE OPERATIONAL IQ response or CREATE OPERATIONAL OQ response and the STATUS field of the response is set to GOOD (see 10.1.5.1), then the queue has been successfully created.

During the creation process, both the PQI management application client and the PQI management device server save attribute values needed for queue operation.

See 5.3.3.3 for additional information on the creation of operation queues.

#### **5.3.3.2 Creating the administrator queue pair**

A PQI device shall support one administrator IQ and one administrator OQ (i.e., an administrator queue pair).

If the FUNCTION AND STATUS CODE field is set to 00h (i.e., IDLE) in the Administrator Queue Configuration Function register (see 6.2.5) and the PQI DEVICE STATE field is set to 2h (i.e., PD2:All\_Registers\_Ready) in the PQI Device Status register (see 6.2.10), then the PQI host may create an administrator queue pair using the following steps:

- 1) read the PQI Device Capability register to determine the element length and the maximum number of elements in both the administrator IQ and administrator OQ;
- 2) allocate the memory spaces using a mechanism outside the scope of this standard for the following:
  - A) administrator IQ element array at an address with the required alignment (see 6.2.13);
  - B) administrator OQ element array at an address with the required alignment (see 6.2.14);
  - C) administrator IQ CI at an address with the required alignment (see 6.2.15); and
  - D) administrator OQ PI at an address with the required alignment (see 6.2.16);
- 3) set:
  - A) the administrator IQ CI to zero;
  - B) the administrator OQ PI to zero;
  - C) the administrator IQ PI local copy to zero; and
  - D) the administrator OQ CI local copy to zero;
- 4) set the PQI device registers as described in table 15;
- 5) set the value of the FUNCTION AND STATUS CODE field to 01h (i.e., CREATE ADMINISTRATOR QUEUE PAIR) in the Administrator Queue Configuration Function register;
- 6) read the Administrator Queue Configuration Function register until:
  - A) the FUNCTION AND STATUS CODE field is set to 00h (i.e., IDLE); or
  - B) 100 ms has elapsed after step 5);
- 7) if the FUNCTION AND STATUS CODE field is not set to 00h (i.e., IDLE) and 100 ms has elapsed, then read the Administrator Queue Configuration Function register;
- 8) if the FUNCTION AND STATUS CODE field is set to 00h, then:
  - A) read the administrator IQ PI offset from the Administrator IQ PI Offset register and save the value in PQI host local memory; and
  - B) read the administrator OQ CI offset from the Administrator OQ CI Offset register and save the value in PQI host local memory;
- 9) if the FUNCTION AND STATUS CODE field is not set to 00h and PD state machine is in the PD4:Error state (see 5.5.6), then read the PQI Device Status register, the PQI Device Error register, and the PQI Device Error Details register, and report an error in a vendor specific manner; and
- 10) if the FUNCTION AND STATUS CODE field is not set to 00h and the PD state machine is not in the PD4:Error state, then report an error in a vendor specific manner.

**Table 15 — PQI device registers to be written during administrator queue pair creation**

Register	Field	Value
Administrator IQ Element Array Address (see 6.2.13)	ADMINISTRATOR IQ ELEMENT ARRAY ADDRESS	Administrator IQ element array address
Administrator OQ Element Array Address (see 6.2.14)	ADMINISTRATOR OQ ELEMENT ARRAY ADDRESS	Administrator OQ element array address
Administrator IQ CI Address (see 6.2.15)	ADMINISTRATOR IQ CI ADDRESS	Administrator IQ CI address
Administrator OQ PI Address (see 6.2.16)	ADMINISTRATOR OQ PI ADDRESS	Administrator OQ PI address
Administrator Queue Parameter (see 6.2.17)	NUMBER OF ADMINISTRATOR IQ ELEMENTS	Number of elements in the administrator IQ
	NUMBER OF ADMINISTRATOR OQ ELEMENTS	Number of elements in the administrator OQ
	INTERRUPT MESSAGE NUMBER	MSI-X Table entry used to generate the interrupt message for OQ PI updates to the administrator OQ in MSI-X mode.

The PQI device reports an error for the CREATE ADMINISTRATOR QUEUE PAIR PD function if that PD function is invoked while:

- a) the FUNCTION AND STATUS CODE field is not set to 00h (i.e., IDLE) in the Administrator Queue Configuration Function register (see 6.2.5); or
- b) the PQI DEVICE STATE field is set to 3h (i.e., PD2:All\_Registers\_Ready) (see 5.5.5.4).

If the PQI device is requested to create the administrator queue pair, the PQI device shall:

- 1) allocate PQI device memory space for the administrator IQ PI;
- 2) set the ADMINISTRATOR IQ PI OFFSET field to the administrator IQ PI offset in the Administrator IQ PI Offset register (see 6.2.11);
- 3) allocate PQI device memory space for the administrator OQ CI;
- 4) set the ADMINISTRATOR OQ CI OFFSET field to the administrator OQ CI offset in the Administrator OQ CI Offset register (see 6.2.12);
- 5) initialize:
  - A) the administrator IQ PI to zero;
  - B) the administrator OQ CI to zero;
  - C) the administrator IQ CI local copy to zero; and
  - D) the administrator OQ PI local copy to zero.

and
- 6) set the FUNCTION AND STATUS CODE field to 00h (i.e., IDLE) in the Administrator Queue Configuration Function register (see 5.5.4.4).

If the PQI device fails to complete any of the steps to create the administrator queue pair, then the PQI device shall exit the sequence and:

- a) the PQI device shall not change the FUNCTION AND STATUS CODE field from the current value of 01h (i.e., CREATING ADMINISTRATOR QUEUE PAIR) in the Administrator Queue Configuration Function register;
- b) the PQI device shall set the error code to ERROR CREATING ADMINISTRATOR QUEUE PAIR in the PQI Device Error register (see 6.2.18 and 5.5.4.5); and
- c) the PD state machine transitions as described in 5.5.4.5.

### 5.3.3.3 Creating operational queues

Operational queues are created by the CREATE OPERATIONAL IQ function (see 10.2.4) and the CREATE OPERATIONAL OQ function (see 10.2.5) using the administrator queue pair.

Prior to enqueueing a CREATE OPERATIONAL IQ request (see 10.2.4.1) to the administrator IQ, the PQI host should:

- 1) use the REPORT PQI DEVICE CAPABILITY administrator function (see 10.2.2) to determine the supported element length and number of elements;
- 2) allocate PCI memory spaces using a mechanism outside the scope of this standard for the following:
  - A) the operational IQ element array, with a 64 byte address alignment; and
  - B) the operational IQ CI, with a 4 byte address alignment;
 and
- 3) initialize the values in:
  - A) the operational IQ CI to zero; and
  - B) the operational IQ PI local copy to zero.

If the PQI device processes a CREATE OPERATIONAL IQ request, then the PQI device shall:

- 1) allocate PQI device memory space for the administrator IQ PI; and
- 2) initialize:
  - A) the operational IQ PI to zero; and
  - B) the operational IQ CI local copy to zero.

If the PQI device successfully completes creating the operational IQ, then the PQI device shall enqueue a CREATE OPERATIONAL IQ response (see 10.2.4.2) with the STATUS field set to GOOD (see 10.1.5.1).

If the PQI device fails to complete creating the operational IQ, then the PQI device shall enqueue a CREATE OPERATIONAL IQ response with the STATUS field and the additional status descriptor set to a value that indicates the error (see 10.1.5.1).

Prior to enqueueing a CREATE OPERATIONAL OQ request (see 10.2.5.1) to the administrator IQ, the PQI host should:

- 1) use the REPORT PQI DEVICE CAPABILITY administrator function to determine the supported element length and number of elements;
- 2) allocate PCI memory spaces using a mechanism outside the scope of this standard for the following:
  - A) the operational OQ element array, with a 64 byte address alignment; and
  - B) the operational OQ PI, with a 4 byte address alignment;
 and
- 3) initialize the values in:
  - A) the operational OQ PI to zero; and
  - B) the operational OQ CI local copy to zero.

If the PQI device processes a CREATE OPERATIONAL OQ request, then the PQI device shall:

- 1) allocate PQI device memory space for the administrator OQ CI; and
- 2) initialize:
  - A) the operational OQ CI to zero; and
  - B) the operational OQ PI local copy to zero.

If the PQI device successfully completes creating the operational OQ, then the PQI device shall enqueue a CREATE OPERATIONAL OQ response (see 10.2.5.2) with the STATUS field set to GOOD (see 10.1.5.1).

If the PQI device fails to complete creating the operational OQ, then the PQI device shall enqueue a CREATE OPERATIONAL OQ response with the STATUS field and the additional status descriptor set to a value that indicates the error (see 10.1.5.1).

### 5.3.4 Deleting circular queues

#### 5.3.4.1 Deleting circular queues overview

The management application client requests that the administrator queue pair and the operational queues be deleted in the following order:

- 1) delete all of the operational IQs and operational OQs using administrator functions (see 10.2); and
- 2) delete the administrator queue pair using the DELETE ADMINISTRATOR QUEUE PAIR PD function (see 6.2.5).

#### 5.3.4.2 Deleting the administrator queue pair

If the FUNCTION AND STATUS CODE field is set to 00h (i.e., IDLE) in the Administrator Queue Configuration Function register (see 6.2.5) and the PQI DEVICE STATE field is set to 3h (i.e., PD3:Administrator\_Queue\_Pair\_Ready) in the PQI Device Status register (see 6.2.10), then the PQI host may delete the administrator queue pair using the following steps:

- 1) set the value of the FUNCTION AND STATUS CODE field to 02h (i.e., DELETE ADMINISTRATOR QUEUE PAIR) in the Administrator Queue Configuration Function register;
- 2) read the Administrator Queue Configuration Function register until:
  - A) the FUNCTION AND STATUS CODE field is set to 00h (i.e., IDLE); or
  - B) 100 ms has elapsed;
- 3) if the FUNCTION AND STATUS CODE field is not set to 00h (i.e., IDLE), then read the Administrator Queue Configuration Function register;
- 4) if the FUNCTION AND STATUS CODE field is not set to 00h (i.e., IDLE), then:
  - A) read the PQI Device Status register;
  - B) read the PQI Device Error register (see 6.2.18); and
  - C) report an error in a vendor specific manner;
 and
- 5) deallocate the PCI memory spaces using a mechanism outside the scope of this standard for the following:
  - A) administrator IQ element array;
  - B) administrator OQ element array;
  - C) administrator IQ CI; and
  - D) administrator OQ PI.

If the PQI device is requested to delete the administrator queue pair, then the PQI device shall:

- 1) deallocate PQI device memory space for the administrator IQ PI;
- 2) set the ADMINISTRATOR IQ PI OFFSET field to zero in the Administrator IQ PI Offset register (see 6.2.11);
- 3) deallocate PQI device memory space for the administrator OQ CI; and
- 4) set the ADMINISTRATOR OQ CI OFFSET field to zero in the Administrator OQ CI Offset register (see 6.2.12).

If the PQI device successfully completes deleting the administrator queue pair, then:

- 1) the PQI device shall set the FUNCTION AND STATUS CODE field to 00h (i.e., IDLE) in the Administrator Queue Configuration Function register; and
- 2) the PD state machine transitions as described in 5.5.5.4.

If the PQI device fails to complete the deletion of the administrator queue pair, then:

- a) the PQI device shall not change the FUNCTION AND STATUS CODE field from the current value of 02h (i.e., DELETING ADMINISTRATOR QUEUE PAIR) in the Administrator Queue Configuration Function register;
- b) the PQI device shall set the error code to ERROR DELETING ADMINISTRATOR QUEUE PAIR in the PQI Device Error register (see 6.2.18); and
- c) the PD state machine transitions as described in 5.5.5.5.

A PQI device shall delete the administrator queue pair during a PQI reset or a PCI Express reset.

After a PQI reset or a PCI Express reset, the PQI host may deallocate the PCI memory spaces using a mechanism outside the scope of this standard for the following:

- a) administrator IQ element array;
- b) administrator OQ element array;
- c) administrator IQ CI; and
- d) administrator OQ PI.

#### 5.3.4.3 Deleting operational queues

Operational queues are deleted by the DELETE OPERATIONAL IQ function (see 10.2.6) and the DELETE OPERATIONAL OQ function (see 10.2.7) using the administrator queue pair.

The PQI host should wait until the IQ is empty before issuing the DELETE OPERATIONAL IQ function. The result of deleting an operational IQ if it is not empty is not defined by this standard.

The PQI host management application client should make sure that the corresponding operational OQ is not required for any pending operation before issuing the DELETE OPERATIONAL OQ function. The result of deleting an operational OQ if the operational OQ is still being used is not defined by this standard.

If the PQI device is requested to delete an operational IQ, then the PQI device shall deallocate PQI device memory space for the operational IQ PI.

If the PQI device successfully completes deleting the operational IQ, then the PQI device shall enqueue a DELETE OPERATIONAL IQ response (see 10.2.6.2) with the STATUS field set to GOOD (see 10.1.5.1).

If the PQI device fails to complete deleting the operational IQ, then the PQI device shall enqueue a DELETE OPERATIONAL IQ response with the STATUS field and the additional status descriptor set to a value that indicates the error (see 10.1.5.1).

After dequeuing a DELETE OPERATIONAL IQ response from the administrator OQ, the PQI host should deallocate the PCI memory spaces using a mechanism outside the scope of this standard for the following:

- a) operational IQ element array; and
- b) operational IQ CI.

If the PQI device is requested to delete an operational OQ, then the PQI device shall deallocate PQI device memory space for the operational OQ CI.

If the PQI device successfully completes deleting the operational OQ, then the PQI device shall enqueue a DELETE OPERATIONAL OQ response (see 10.2.7.2) with the STATUS field set to GOOD (see 10.1.5.1).

If the PQI device fails to complete deleting the operational OQ, then the PQI device shall enqueue a DELETE OPERATIONAL OQ response with the STATUS field and the additional status descriptor set to a value that indicates the error (see 10.1.5.1).

After dequeuing a DELETE OPERATIONAL OQ response from the administrator OQ, the PQI host should deallocate the PCI memory spaces using a mechanism outside the scope of this standard for the following:

- a) operational OQ element array; and
- b) operational OQ PI.

A PQI device shall delete all operational IQs and all operational OQs during a PQI reset or a PCI Express reset.



After a PQI reset or a PCI Express reset, the PQI host may deallocate the PCI memory spaces using a mechanism outside the scope of this standard for the following:

- a) operational IQ element arrays;
- b) operational OQ element arrays;
- c) operational IQ CIs; and
- d) operational OQ PIs.

### 5.3.5 IQ priority

The PQI device should consume all IUs from the administrator IQ (i.e., the administrator IQ is empty) before consuming IUs from any operational IQ.

## 5.4 OQ service notification methods

### 5.4.1 OQ service notification methods overview

Table 16 lists the OQ service notification methods that are used by the PQI host to determine when an entry is available in an OQ.

If MSI-X (see PCI) is enabled, then MSI-X mode is used. If MSI-X is not enabled then legacy INTx (see PCI) may be used. If the PQI host masks off the generation of MSI-X interrupts or legacy INTx interrupts by the PQI device and uses other methods (e.g., reading OQ PI) to determine when an event has occurred, then polled mode is being used.

**Table 16 — OQ service notification methods**

Mode	PQI device	PQI host
MSI-X	5.4.2	5.4.5
Legacy INTx	5.4.3	5.4.6
Polled	5.4.4	5.4.7

### 5.4.2 Sending OQ service notifications in MSI-X mode

#### 5.4.2.1 Sending OQ service notifications in MSI-X mode overview

A memory write transaction used to send an MSI-X interrupt should use the same Traffic Class (see PCIe) that is used to write the IU to the OQ and to update the OQ PI for that OQ and shall not be sent until any update to the OQ PI that caused the generation of the MSI-X interrupt has been sent.

#### 5.4.2.2 Sending OQ service notifications for an administrator OQ

In MSI-X mode, the PQI device shall send the PQI device interrupt each time after the administrator OQ PI is written with a new value. If the administrator OQ is mapped to the same MSI-X interrupt vector (see PCI) as one or more operational OQs in a PQI device, then interrupt coalescing for the MSI-X interrupt associated with that MSI-X interrupt vector is vendor specific.

#### 5.4.2.3 Sending OQ service notifications for an operational OQ

If only one operational OQ is mapped to a MSI-X interrupt vector in a PQI device, then the PQI device sends an MSI-X interrupt (see PCI and PCIe) every time an interrupt event (see table 17) occurs. If more than one operational OQ is mapped to a MSI-X interrupt vector in a PQI device, then interrupt coalescing for the MSI-X interrupt associated with that MSI-X interrupt vector is vendor specific.

In MSI-X mode, a PQI device interrupt event may occur when an operational OQ contains one or more occupied elements.

A PQI device implements a coalescing timer for each operational OQ. If the coalescing timer is reset, then a coalescing timer is set to a value of zero. If the coalescing timer is started, then the coalescing timer increases as time elapses. If the coalescing timer is stopped, then the coalescing timer retains its current value.

For each operational OQ, a PQI device maintains the following attributes related to the generation of MSI-X interrupts:

- a) Minimum Coalescing Time attribute (see 5.2.5.12.8);
- b) Maximum Coalescing Time attribute (see 5.2.5.12.9);
- c) Coalescing Count attribute (see 5.2.5.12.10);
- d) Wait For Rearm attribute (see 5.2.5.12.11); and
- e) Interrupt Message Number attribute (see 5.2.5.12.12).

The Minimum Coalescing Time attribute, Maximum Coalescing Time attribute, Coalescing Count attribute, Wait For Rearm attribute, and Interrupt Message Number attribute are specified by the CREATE OPERATIONAL OQ function.

The Minimum Coalescing Time attribute, Maximum Coalescing Time attribute, Coalescing Count attribute, and Wait For Rearm attribute may be modified by the CHANGE OPERATIONAL OQ PROPERTIES function.

For an operational OQ:

$$\text{occupied element count} = (n + \text{OQ PI} - \text{OQ CI}) \bmod n$$

where:

- occupied element count    the number of occupied elements in the OQ;
- n                                is the number of elements in the OQ;
- OQ PI                        is the index of the OQ element where the producer writes next; and
- OQ CI                        is the index of the OQ element where the consumer reads next.

For an operational OQ an interrupt event occurs if:

- a) the condition described in table 17 is true; and
- b) the associated event described in table 17 occurs.

If an interrupt event occurs, then the PQI device should send the MSI-X interrupt designated for that operational OQ. If more than one event in table 17 occurs at the same time for an operational OQ, then the PQI device shall only generate one MSI-X interrupt.

**Table 17 — Interrupt generation conditions and events**

Condition	Event
The occupied element count is greater than or equal to the Coalescing Count attribute and the Minimum Coalescing Time attribute is set to non-zero.	The coalescing timer equals the Minimum Coalescing Time attribute.
The occupied element count is greater than or equal to one and the Maximum Coalescing Time attribute is set to non-zero.	The coalescing timer equals the Maximum Coalescing Time attribute.
The coalescing timer is greater than or equal to the Minimum Coalescing Time attribute.	PQI device writes to the operational OQ PI causing the occupied element count to be greater than or equal to the Coalescing Count attribute.
The coalescing timer is greater than or equal to the Maximum Coalescing Time attribute.	PQI device writes to the operational OQ PI causing the occupied element count to be non-zero.

The delay between an event from table 17 and the MSI-X memory write transaction on the PCI Express interface is vendor specific.

After sending an interrupt for an operational OQ:

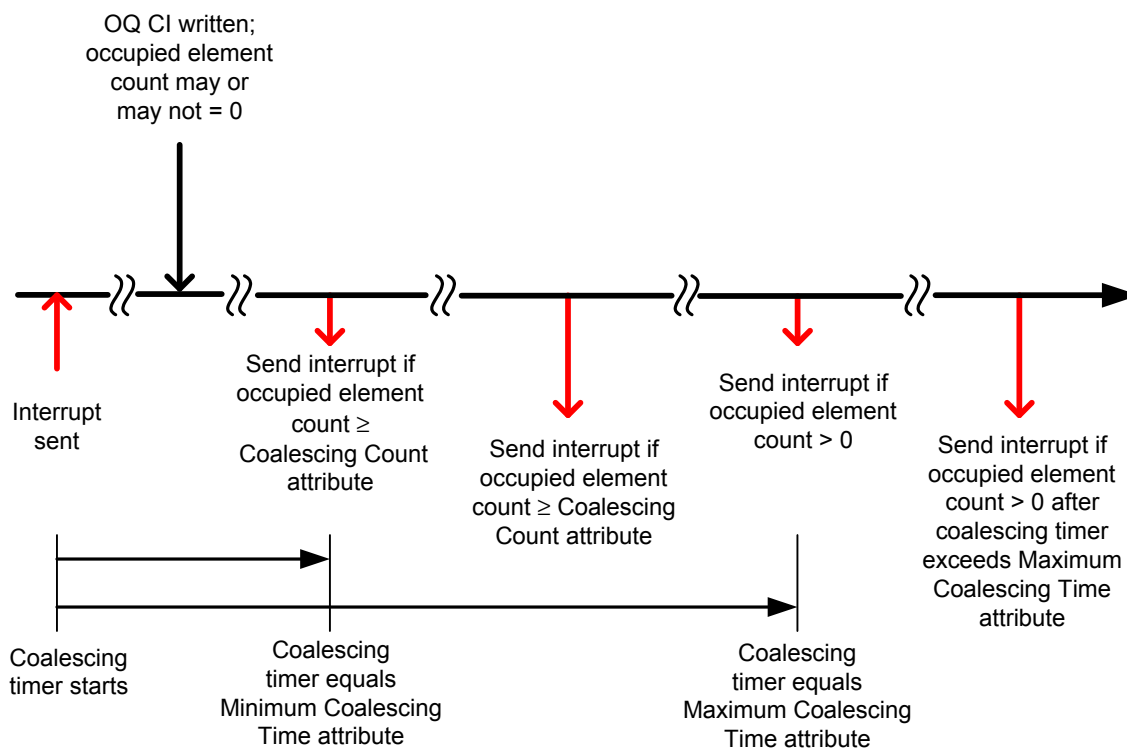
- a) if the Wait For Rearm attribute is set to zero, then the PQI device shall reset and start the coalescing timer; and
- b) if the Wait For Rearm attribute is set to one, then the PQI device shall reset and stop the coalescing timer.

If:

- a) a memory write transaction to the operational OQ CI dword is received with the REARM INTERRUPT bit set to one (see 7.1.3); and
- b) the Wait For Rearm attribute is set to one,

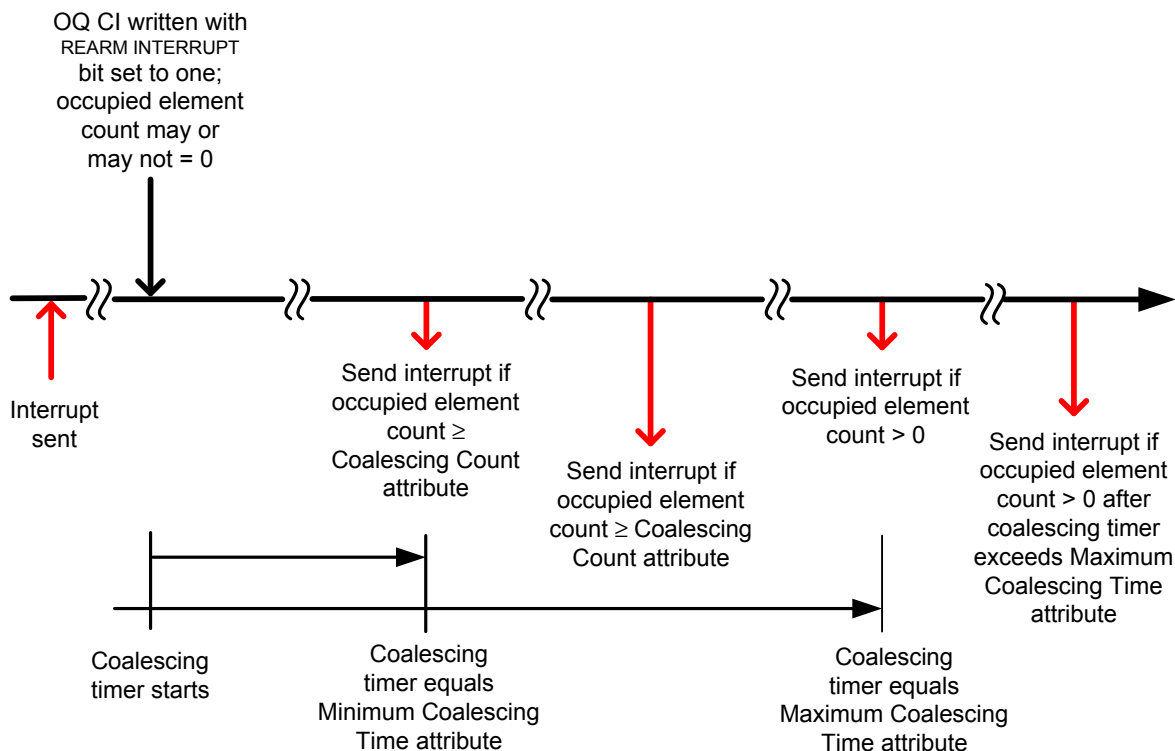
then the PQI device shall reset and start the coalescing timer.

Figure 24 shows an example of interrupt coalescing with the WAIT FOR REARM bit set to zero.



**Figure 24 — Interrupt coalescing example with the WAIT FOR REARM bit set to zero**

Figure 25 shows an example of interrupt coalescing with the WAIT FOR REARM bit set to one.



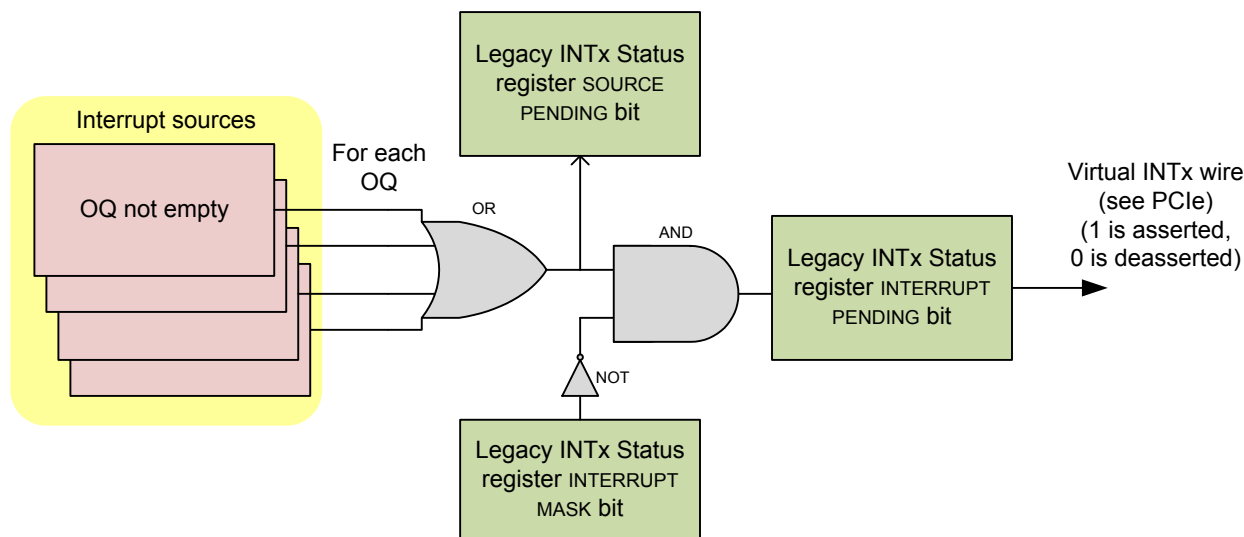
**Figure 25 — Interrupt coalescing example with the WAIT FOR REARM bit set to one**

#### 5.4.3 Sending OQ service notifications in legacy INTx mode

In legacy INTx mode, the PQI device sends Assert\_INTx and Deassert\_INTx Messages (see PCIe) to maintain a virtual wire that reflects whether interrupt sources are asserted or deasserted (i.e., level triggered).

In legacy INTx mode, any OQ containing one or more occupied elements is an interrupt source.

Figure 26 shows the legacy INTx interrupt sources and masks.



**Figure 26 — Legacy INTx sources and masks**

The PQI device shall assert the virtual INTx wire while:

- a) any of its interrupt sources are asserted; and
- b) the Legacy INTx Interrupt Mask register INTERRUPT MASK bit is set to zero.

The PQI device shall deassert the virtual INTx wire while:

- a) all of its interrupt sources are not asserted; or
- b) the Legacy INTx Interrupt Mask register INTERRUPT MASK bit is set to one.

An OQ interrupt source is cleared when the OQ CI equals the OQ PI (i.e., OQ is empty).

#### **5.4.4 Sending OQ service notifications in polled mode**

In polled mode, the PQI device shall not send interrupts.

#### **5.4.5 Servicing OQ service notifications in MSI-X mode**

In MSI-X mode, if the WAIT FOR REARM bit (see 5.4.2 and 10.2.5.1) is set to one, then after reading one or more elements from the OQ, the PQI host should set the REARM INTERRUPT bit (see 7.1.3 and 10.2.5.1) to one before exiting the interrupt service routine (e.g., after consuming all occupied elements in the OQ).

If the PQI host receives an interrupt assigned to more than one OQ, then the PQI host should consume all IUs from each OQ assigned to that interrupt.

#### **5.4.6 Servicing OQ service notifications in legacy INTx mode**

In legacy INTx mode, if the PQI host receives an interrupt, then the PQI host should:

- 1) if the legacy INTx interrupt is shared with other PCI functions (see PCI), then read the Legacy INTx Interrupt Status register (see 6.2.7); and

NOTE 9 - As specified by PCIe, the read from the Legacy INTx Status register ensures that all memory writes from the PQI device have completed.

- 2) if the Legacy INTx Interrupt Status register INTERRUPT PENDING bit is set to one (i.e., the PQI device is a source of the interrupt), then determine the PQI device's interrupt source(s) by checking the OQ PI of each OQ to determine if the OQ is not empty.

#### **5.4.7 Servicing OQ service notifications in polled mode**

In polled mode, the PQI host should poll the OQ PI of each OQ to determine if the OQ is not empty.

## 5.5 PD (PQI device) state machine

### 5.5.1 PD (PQI device) state machine overview

The PD state machine describes the PQI device states and transitions for PCI configuration and device operation.

This state machine consists of the following states:

- a) PD0:Power\_On\_And\_Reset (see 5.5.2) (initial state);
- b) PD1:PQI\_Status\_Available (see 5.5.3);
- c) PD2:All\_Registers\_Ready (see 5.5.4);
- d) PD3:Administrator\_Queue\_Pair\_Ready (see 5.5.5); and
- e) PD4:Error (see 5.5.6).

The PD state machine shall start in the PD0:Power\_On\_And\_Reset state when the PCI Function enters the D0 uninitialized state (see PCI-PM), (e.g., upon power on or following a PCI Express reset (see PCIe)).

The PQI device state is indicated by the PQI DEVICE STATE field in the PQI Device Status register (see 6.2.10). The PQI device shall set the PQI DEVICE STATE field (see table 29) to the appropriate value upon entry into each state. Unless otherwise specified, PCI-PM state transitions (see PCI-PM) do not cause PD state transitions.

Figure 27 describes the PD state machine.

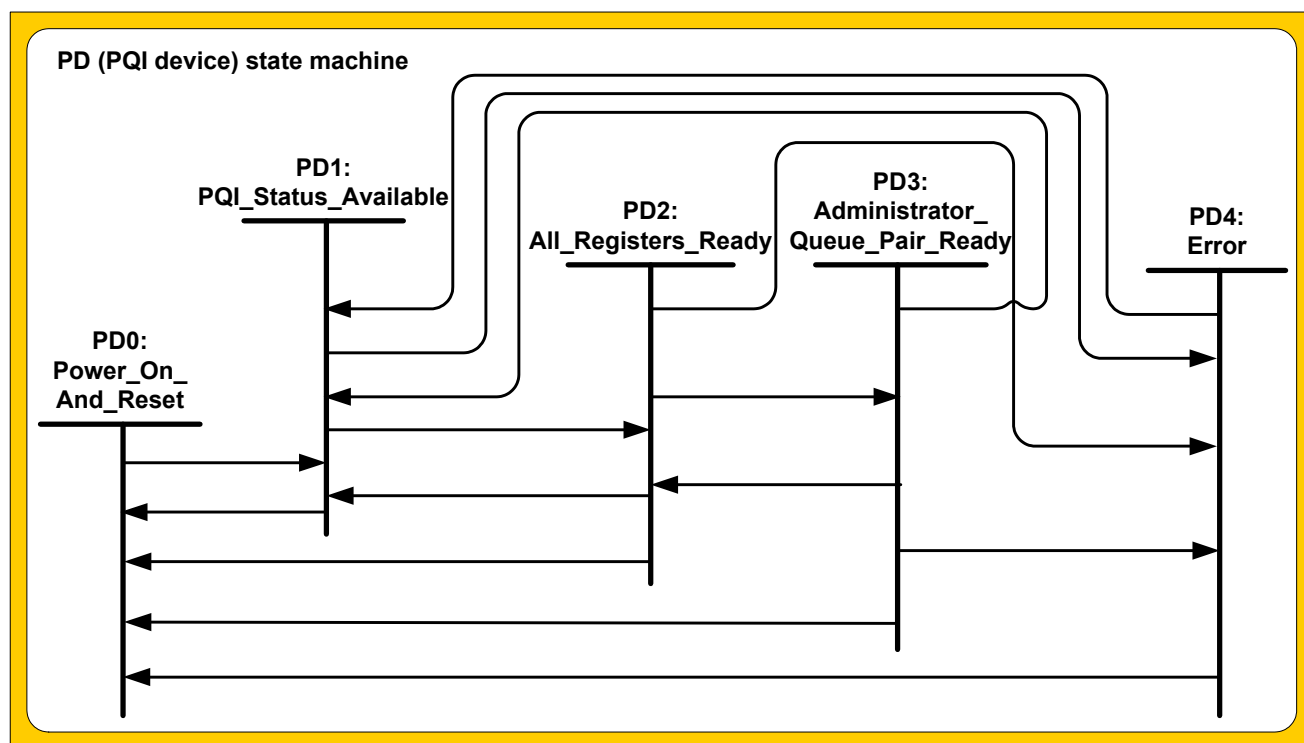


Figure 27 — PD (PQI device) state machine

### 5.5.2 PD0:Power\_On\_And\_Reset state

#### 5.5.2.1 PD0:Power\_On\_And\_Reset state description

A PQI device enters this initial state when the PCI Function enters the D0 uninitialized state (see PCI-PM) (e.g., upon power on or following a PCI Express reset (see PCIe)).

While in this state, the PQI device shall perform internal initialization (e.g., POST) and shall set all register values to their default values (see 6.2).

#### 5.5.2.2 Transition PD0:Power\_On\_And\_Reset to PD1:PQI\_Status\_Available

This transition shall occur after the PQI device:

- a) reaches the D0 active power management state (see PCI-PM) with the Memory Space Enable bit set to one in the Command register (see PCI).

#### 5.5.3 PD1:PQI\_Status\_Available state

##### 5.5.3.1 PD1:PQI\_Status\_Available state description

While in this state the PQI device is initialized by system software (see PCI).

Upon entry to this state PQI device initialization begins. While in this state:

- a) if an error occurs during PQI device initialization, then:
  - A) the PQI device shall set the error code to ERROR DETECTED DURING INITIALIZATION in the PQI Device Error register (see 6.2.18); and
  - B) the PD state machine transitions as described in 5.5.3.4;
- b) all queues are deleted and their corresponding objects are set to their uninitialized state;
- c) register attributes (see PCIe) are as defined in table 19; and
- d) the PQI device is completing PQI reset.

##### 5.5.3.2 Transition PD1:PQI\_Status\_Available to PD0:Power\_On\_And\_Reset

This transition shall occur if:

- a) the PQI device detects a PCI Express reset.

##### 5.5.3.3 Transition PD1:PQI\_Status\_Available to PD2:All\_Registers\_Ready

This transition shall occur if the HOLD IN PD1 bit is set to zero in the PQI Device Reset register (see 6.2.20) and after:

- a) PQI device internal initialization (e.g., POST) is complete; and
- b) the PQI device is ready to process PD functions (see 6.2.5).

##### 5.5.3.4 Transition PD1:PQI\_Status\_Available to PD4:Error

This transition shall occur if:

- a) an error is detected during PQI device initialization;
- b) an internal error (e.g., device error) is detected; and
- c) an error is detected when completing PQI reset.

#### 5.5.4 PD2:All\_Registers\_Ready state

##### 5.5.4.1 PD2:All\_Registers\_Ready state description

While in this state:

- a) the PQI device memory space is configured;
- b) the PQI device standard registers are available;
- c) the PQI host may create the administrator queue pair (see 5.3.3.2);
- d) the PQI device shall process the Administrator Queue Configuration Function register function (e.g., CREATE ADMINISTRATOR QUEUE PAIR) (see 5.3.3.2); and
- e) if the PQI device detects an error described in table 23, then the PQI shall set the error code as described in table 23.

Register attributes (see PCIe) are as defined in table 19.

**5.5.4.2 Transition PD2:All\_Registers\_Ready to PD0:Power\_On\_And\_Reset**

This transition shall occur if:

- a) the PQI device detects a PCI Express reset.

**5.5.4.3 Transition PD2:All\_Registers\_Ready to PD1:PQI\_Status\_Available**

This transition shall occur if:

- a) the PQI device detects a PQI reset (see 5.7).

**5.5.4.4 Transition PD2:All\_Registers\_Ready to PD3:Administrator\_Queue\_Pair\_Ready**

This transition shall occur if:

- a) the FUNCTION AND STATUS CODE field is set to 00h (i.e., IDLE) in the Administrator Queue Configuration Function register; and
- b) the administrator queue pair has been successfully created (see 5.3.3.2).

**5.5.4.5 Transition PD2:All\_Registers\_Ready to PD4:Error**

This transition shall occur if the PQI device detects:

- a) a reserved code value used in the FUNCTION AND STATUS CODE field in the Administrator Queue Configuration Function register (see table 23);
- b) an invalid value in the Administrator Queue Parameter register (see 6.2.17);
- c) an administrator queue pair create error (see table 23);
- d) an administrator queue pair delete error (see table 23);
- e) an internal error (e.g., device error); or
- f) the FUNCTION AND STATUS CODE field is set to 01h or 02h and the PQI device is already processing a PD function (see table 23).

**5.5.5 PD3:Administrator\_Queue\_Pair\_Ready state****5.5.5.1 PD3:Administrator\_Queue\_Pair\_Ready state description**

While in this state:

- a) the PQI device is initialized;
- b) the PQI device is ready to process any PCI transaction (see PCI-PM);
- c) the PQI device shall process the Administrator Queue Configuration Function register (see 6.2.5) function (e.g., DELETE ADMINISTRATOR QUEUE PAIR);
- d) the PQI device shall process administrator IUs using the administrator queue pair;
- e) if the operational IQ and the operational OQ have been created (see 5.3.3.3), then the PQI device shall process operational IUs using the operational IQ and the operational OQ;
- f) register attributes (see PCIe) are as defined in table 19; and
- g) if the PQI device detects an error described in table 23, then the PQI shall set the error code as described in if the PQI device detects an error described in table 23, then the PQI shall set the error code as described in table 23.

**5.5.5.2 Transition PD3:Administrator\_Queue\_Pair\_Ready to PD0:Power\_On\_And\_Reset**

This transition shall occur if:

- a) the PQI device detects a PCI Express reset.

**5.5.5.3 Transition PD3:Administrator\_Queue\_Pair\_Ready to PD1:PQI\_Status\_Available**

This transition shall occur if:

- a) the PQI device detects a PQI reset.



**5.5.5.4 Transition PD3:Administrator\_Queue\_Pair\_Ready to PD2:All\_Registers\_Ready**

This transition shall occur after:

- a) the PQI device successfully completes deleting the administrator queue pair using the DELETE ADMINISTRATOR QUEUE PAIR PD function (see 6.2.5); and
- b) the PQI device sets the FUNCTION AND STATUS CODE field to 00h (i.e., IDLE) in the Administrator Queue Configuration Function register.

**5.5.5.5 Transition PD3:Administrator\_Queue\_Pair\_Ready to PD4:Error**

This transition shall occur if the PQI device detects:

- a) a reserved code value used in the FUNCTION AND STATUS CODE field in the Administrator Queue Configuration Function register (see 6.2.5);
- b) an administrator queue creation (see 5.3.3.2) is initiated by PQI host;
- c) an administrator queue pair delete error (see 5.3.4.2);
- d) an internal error (e.g., device error);
- e) an invalid IU TYPE field (see 9.3) in an inbound IU;
- f) an invalid IU LENGTH field (see 9.3) in an inbound IU;
- g) an inbound IU in an operational IQ that specifies an invalid OQ ID (see 5.2.5.12.2);
- h) the FUNCTION AND STATUS CODE field is set to 01h or 02h and the PQI device is already processing a PD function (see table 23); or
- i) all operational queues have not been deleted and the PQI device receives the delete administrator queue pair request (see table 23).

**5.5.6 PD4:Error state****5.5.6.1 PD4:Error state description**

Upon entry to this state:

- a) the PQI Device Error register indicates the error (see 6.2.18);
- b) all outstanding IU operations are aborted; and
- c) the administrator queue pair, if any, and the operational queues, if any, are in an unknown state.

If an internal error occurs, then the PQI device shall set the error code to INTERNAL ERROR in the PQI Device Error register (see 5.6).

**5.5.6.2 Transition PD4:Error to PD0:Power\_On\_And\_Reset**

This transition shall occur if:

- a) the PQI device detects a PCI Express reset.

**5.5.6.3 Transition PD4:Error to PD1:PQI\_Status\_Available**

This transition shall occur if:

- a) the PQI device detects a PQI reset.

## 5.6 Register based error information

Register based error information indicates error information to the PQI host that is not able to be reported using an administrator outbound IU (e.g., the administrator OQ is not available or the administrator OQ is corrupted).

The register based error information is indicated by the following fields in the PQI Device Error register (see 6.2.18):

- a) the ERROR CODE field;
- b) the ERROR CODE QUALIFIER field;
- c) the BYTE POINTER field; and
- d) the BIT POINTER field.

If more than one error is detected, then the PQI device may report any one of the errors.

The ERROR DETAILS REGISTER VALID bit in the PQI Device Error register indicates if the error details information is available in the PQI Device Error Details register (see 6.2.19).

Table 18 defines the register based error information.

**Table 18 — Register based error information structure** (part 1 of 2)

Error code			BYTE POINTER field and BIT POINTER field	ERROR DETAILS REGISTER VALID bit	Reference
ERROR CODE field	ERROR CODE QUALIFIER field	Name			
00h	00h	NO ERROR	Invalid	0 or 1 <sup>a</sup>	6.2.18
	01h to FFh	Reserved			
01h	00h	ERROR DETECTED DURING INITIALIZATION	Invalid	0 or 1 <sup>a</sup>	5.5.3.1
	01h to FFh	Reserved			
02h	00h	Reserved			
	01h	INVALID PD FUNCTION	Invalid	0 or 1 <sup>a</sup>	6.2.5
	02h	INVALID PARAMETER FOR PD FUNCTION	Valid	0 or 1 <sup>a</sup>	6.2.17
	03h to FFh	Reserved			
03h	00h	ERROR CREATING ADMINISTRATOR QUEUE PAIR	Invalid	0 or 1 <sup>a</sup>	
	01h	ERROR DELETING ADMINISTRATOR QUEUE PAIR	Invalid	0 or 1 <sup>a</sup>	5.3.4.2
	02h to FFh	Reserved			
<sup>a</sup> If set to one, then the content of the PQI Device Error Details register is vendor specific.					

Table 18 — Register based error information structure (part 2 of 2)

Error code			BYTE POINTER <b>field and</b> BIT POINTER <b>field</b>	ERROR DETAILS REGISTER VALID <b>bit</b>	Reference
ERROR CODE <b>field</b>	ERROR CODE QUALIFIER <b>field</b>	Name			
04h	00h	Reserved			
	01h	INVALID IU TYPE IN GENERAL ADMIN REQUEST IU	Invalid	0 or 1 <sup>a</sup>	10.1.2
	02h	INVALID IU LENGTH IN GENERAL ADMIN REQUEST IU	Invalid	0 or 1 <sup>a</sup>	10.1.2
	03h to FFh	Reserved			
05h	00h	INTERNAL ERROR	Invalid	0 or 1 <sup>a</sup>	5.5.6.1
	01h	OQ SPANNING CONFLICT	Invalid	0 or 1 <sup>a</sup>	5.3.2.5.4
	02h to FFh	Reserved			
06h	00h	Reserved			
	01h	ERROR COMPLETING PQI SOFT RESET	Invalid	0 or 1 <sup>a</sup>	5.7.2
	02h	ERROR COMPLETING PQI FIRM RESET	Invalid	0 or 1 <sup>a</sup>	5.7.3
	03h	ERROR COMPLETING PQI HARD RESET	Invalid	0 or 1 <sup>a</sup>	5.7.4
	04h to FFh	Reserved			
07h to 7Fh	00h to FFh	Reserved			
80h to FFh	00h to FFh	Vendor specific			
<sup>a</sup> If set to one, then the content of the PQI Device Error Details register is vendor specific.					

## 5.7 PQI reset

### 5.7.1 PQI reset overview

PQI soft reset (see 5.7.2), PQI firm reset (see 5.7.3), and PQI hard reset (see 5.7.4) allow the PQI host to reset the PQI device without accessing the configuration space (see 6.1) of the PQI device.

The configuration space (see 6.1) of all PQI devices in the PCI Express device shall not be affected by any PQI reset. By preserving the configuration space, the PQI host may be able to resume communication with the PQI device without the need for the system software (see PCI) to re-scan the PCI bus to determine what PCI devices are present following the completion of the PQI reset.

The PQI host may specify that the PQI device remain in the PD1:PQI\_Status\_Available state (see 5.5.3) following a PQI reset if the HOLD IN PD1 bit was set to one in the PQI Device Reset register (see 6.2.20).

The PQI host should initiate a PQI reset using the following steps:

- 1) write to the PQI Device Reset register (see 6.2.20) with the RESET ACTION field set to 001b (i.e., RESET), with a specific reset type in the RESET TYPE field, and the specific option in the HOLD IN PD1 bit;
- 2) wait for at least 100 ms;
- 3) repeatedly read the PQI Device Reset register (see 6.2.20) until:
  - A) the RESET ACTION field is set to 010b (i.e., RESET COMPLETED); or
  - B) the timeout value indicated by the MAXIMUM TIMEOUT FOR PQI DEVICE RESET field in the PQI Device Capability register (see 6.2.6) has expired after step 2);
- 4) if the RESET ACTION field is not set to 010b (i.e., RESET COMPLETED), then read the PQI Device Status register (see 6.2.10) and report the error in a vendor specific manner; and
- 5) if the RESET ACTION field is set to 010b (i.e., RESET COMPLETED), then the PQI reset is completed without error.

If the host set the HOLD IN PD1 bit to one as part of invoking the PQI reset, then to request that the PQI device exit the PD1:PQI\_Status\_Available state (see 5.5.3.3), the PQI host should initiate the following steps:

- 1) write to the PQI Device Reset register (see 6.2.20) with the RESET ACTION field set to 001b (i.e., RESET), with the RESET TYPE field set to 000b (i.e., NO RESET) and the HOLD IN PD1 bit set to zero;
- 2) wait for at least 100 ms;
- 3) repeatedly read the PQI Device Reset register (see 6.2.20) until:
  - A) the RESET ACTION field is set to 010b (i.e., RESET COMPLETED); or
  - B) the value indicated by the MAXIMUM TIMEOUT FOR PQI DEVICE RESET field in the PQI Device Capability register (see 6.2.6) has expired; read the PQI Device Status register (see 6.2.10), the PQI Device Error register (see 6.2.10), and the PQI Device Error Details register (see 6.2.19), and report the error in a vendor specific manner; and
- 4) if the RESET ACTION field is set to 010b (i.e., RESET COMPLETED), then the PQI reset is completed without error.

### 5.7.2 PQI soft reset

A PQI soft reset:

- a) resets the PQI device standard registers for this PQI device (see 6.2.2);
- b) deletes the IU layer (see clause 9 and the IU layer standard) content from only this PQI device;
- c) deletes the administrator queue pair, if any (see 5.3.4.2), for this PQI device;
- d) deletes operational IQs, if any, and operational OQs, if any, as described in 5.3.4.3, for this PQI device;
- e) aborts all administrator functions; and
- f) causes the PD state machine to transition to a known state (see 5.5).

If the PQI device fails to complete PQI soft reset, then:

- a) the PQI device shall set the error code to ERROR COMPLETING PQI SOFT RESET in the PQI Device Error register (see 6.2.18); and
- b) the PD state machine transitions as described in 5.5.

### 5.7.3 PQI firm reset

A PQI firm reset:

- a) resets the PQI device registers for this PQI device (see 6.2);
- b) deletes the IU layer (see clause 9 and the IU layer standard) content from all PQI devices in the PCI Express device;
- c) deletes the administrator queue pair, if any (see 5.3.4.2), for this PQI device;
- d) deletes operational IQs, if any, and operational OQs, if any, as described in 5.3.4.3, for this PQI device;
- e) aborts all administrator functions; and
- f) causes the PD state machine to transition to a known state (see 5.5).

If the PQI device fails to complete PQI firm reset, then:

- a) the PQI device shall set the error code to ERROR COMPLETING PQI FIRM RESET in the PQI Device Error register (see 6.2.18); and
- b) the PD state machine transitions as described in 5.5.

### 5.7.4 PQI hard reset

A PQI hard reset:

- a) resets the PQI device registers (see 6.2) for all PQI devices in the PCI Express device;
- b) deletes the IU layer (see clause 9 and the IU layer standard) content from all PQI devices in the PCI Express device;
- c) deletes the administrator queue pair, if any (see 5.3.4.2), for all PQI devices in the PCI Express device;
- d) deletes operational IQs, if any, and operational OQs, if any, as described in 5.3.4.3, for all PQI devices in the PCI Express device;
- e) aborts all administrator functions; and
- f) causes the PD state machine to transition to a known state (see 5.5).

If the PQI device fails to complete PQI hard reset, then:

- a) the PQI device shall set the error code to ERROR COMPLETING PQI HARD RESET in the PQI Device Error register (see 6.2.18); and
- b) the PD state machine transitions as described in 5.5.

## 6 PCI Express requirements and PQI device registers

### 6.1 PCI Express requirements

Clause 6 describes information associated with PCI Express configuration to support this standard.

A PQI device is a kind of PCI function (see PCI).

A PQI device shall:

- a) be a PCI Express Endpoint (see PCIe) or a Root Complex Integrated Endpoint (see PCIe);
- b) support the Advanced Error Reporting Extended Capability (see PCIe);
- c) not require the use of Virtual Channels (see PCIe) or Traffic Classes (see PCIe) beyond the default VC0/TC0 (see PCIe);
- d) support MSI-X (see PCIe); and
- e) use a PCI class code assigned for an IU layer over PQI (see PCI-ID).

A PQI device that has device-specific registers mapped to its first memory BAR that:

- a) have read side effects; or
- b) do not tolerate write merging (see PCIe),

shall set the Prefetchable bit (see PCIe) to zero in its first memory BAR.

A PQI device that has device-specific registers mapped to its first memory BAR that:

- a) do not have read side effects; and
- b) tolerate write merging (see PCIe),

shall set the Prefetchable bit (see PCIe) to one in its first memory BAR. A PQI device shall set its first memory BAR to at least 512 bytes in size allowing for the PQI device registers plus at one administrator queue pair plus up to 126 operational queues (e.g. 63 IQs and 63 OQs). A PQI device should set its first memory BAR to at least 4 KiB in size to match typical CPU page sizes.

A PQI device may support Advanced Error Reporting Multiple Header Recording (see PCIe).

If a PQI device is a function in a PCI Multi-Function device that contains more than eight PCI functions (see PCI), then the PCI Multi-Function device shall support the Alternative Routing-ID Interpretation (ARI) (see PCIe) (i.e., be an ARI Device, support the ARI Capability, and not support Phantom Functions).

A PQI device shall support the Power Budgeting Extended Capability (see PCIe).

A PQI device shall support 32-bit memory writes and 64-bit memory writes to PQI device registers, IQ PI registers, and OQ CI registers. 64-bit PQI device registers may be accessed by 32-bit memory writes in any order (e.g., offset 000h followed by offset 004h, or offset 004h followed by offset 000h). Handling of 8-bit memory writes and the 16-bit memory writes is outside the scope of this standard.

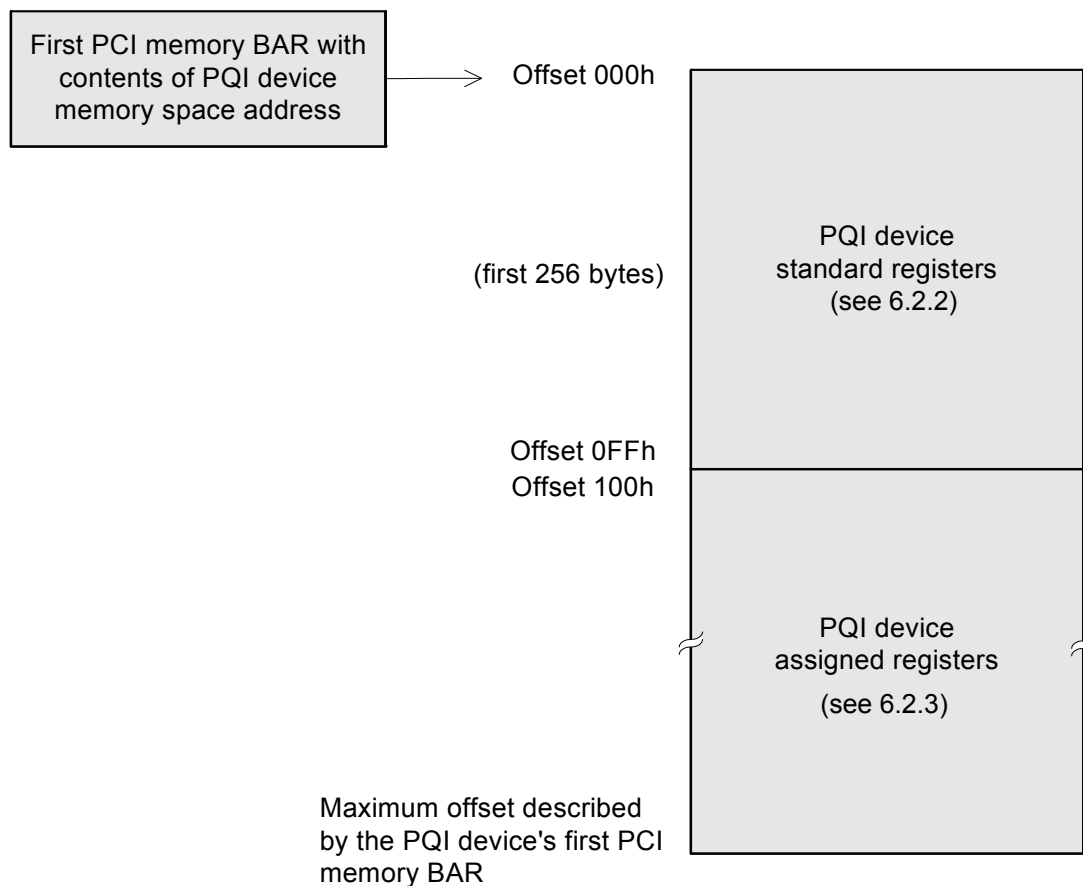
A PQI device shall support zero-length reads (see PCIe), 8-bit memory reads, 16-bit memory reads, 32-bit memory reads, and 64-bit memory reads from PQI device registers, IQ PI registers, and OQ CI registers.

## 6.2 PQI device memory space

### 6.2.1 PQI device memory space overview

PQI device memory space is the memory space described by the PQI device's first PCI memory BAR.

Figure 28 shows the PQI device memory space.



**Figure 28 — PQI device memory space**

The first 256 bytes of the PQI device memory space contain PQI device standard registers as defined in 6.2.2.

The bytes after the first 256 bytes of PQI device memory space are defined in 6.2.3.

### 6.2.2 PQI device standard registers

Table 19 defines the PQI device registers from PQI device memory space offsets 000h through 0FFh (i.e., the first 256 bytes).

**Table 19 — PQI device standard registers from offset 000h to offset 0FFh (part 1 of 2)**

Offset	Size (bytes)	Register name	Register attribute (see PCIe) <sup>a</sup> based on PD state machine state				Reference
			PD1	PD2	PD3	PD4	
000h	8	PQI Device Signature	RO	RO	RO	RO	6.2.4
008h	8	Administrator Queue Configuration Function	RO	RW	RW	RO	6.2.5
010h	8	PQI Device Capability	RO	RO	RO	RO	6.2.6
018h	4	Legacy INTx Interrupt Status	RO	RO	RO	RO	6.2.7
01Ch	4	Legacy INTx Interrupt Mask Set	RO	RW	RW	RO	6.2.8
020h	4	Legacy INTx Interrupt Mask Clear	RO	RW	RW	RO	6.2.9
024h	28	RsvdZ					
040h	4	PQI Device Status	RO	RO	RO	RO	6.2.10
044h	4	RsvdZ					
048h	8	Administrator IQ PI Offset	RO	RO	RO	RO	6.2.11
050h	8	Administrator OQ CI Offset	RO	RO	RO	RO	6.2.12
058h	8	Administrator IQ Element Array Address	RO	RW	RO	RO	6.2.13
060h	8	Administrator OQ Element Array Address	RO	RW	RO	RO	6.2.14
068h	8	Administrator IQ CI Address	RO	RW	RO	RO	6.2.15
070h	8	Administrator OQ PI Address	RO	RW	RO	RO	6.2.16
078h	4	Administrator Queue Parameter	RO	RW	RO	RO	6.2.17
07Ch	4	RsvdZ					
080h	4	PQI Device Error	RO	RO	RO	RO	6.2.18
084h	4	RsvdZ					
Key: RW = Memory reads and memory writes are supported (see PCIe). RO = Memory read are supported, and memory writes are ignored (see PCIe).							
<sup>a</sup> Reading any register defined in this table shall not cause any side effects (e.g., change any register value).							



**Table 19 — PQI device standard registers from offset 000h to offset 0FFh (part 2 of 2)**

Offset	Size (bytes)	Register name	Register attribute (see PCIe) <sup>a</sup> based on PD state machine state				Reference
			PD1	PD2	PD3	PD4	
088h	8	PQI Device Error Details	RO	RO	RO	RO	6.2.19
090h	4	PQI Device Reset	RW	RW	RW	RW	6.2.20
094h	4	Power Action	RO	RW	RW	RO	6.2.21
098h	104	RsvdZ					
Key: RW = Memory reads and memory writes are supported (see PCIe). RO = Memory read are supported, and memory writes are ignored (see PCIe).							
<sup>a</sup> Reading any register defined in this table shall not cause any side effects (e.g., change any register value).							

Unless otherwise specified, writes of reserved values to defined fields within defined registers shall be ignored.

### 6.2.3 PQI device assigned registers

The PQI device memory space offset range from 100h to the maximum offset described by the first PCI memory BAR (see figure 28):

- a) shall contain IQ PI registers (see 7.1.2), if any;
- b) shall contain OQ CI registers (see 7.1.3), if any;
- c) may contain MSI-X Table entries;
- d) may contain MSI-X PBA entries; and
- e) may contain vendor specific registers and vendor specific memory space.

### 6.2.4 PQI Device Signature register

The PQI Device Signature register contains a fixed value.

The PQI host may read this register to determine that the PQI device is present.

Register attributes (see PCIe) are as defined in table 19.

Table 20 defines the PQI Device Signature register.

**Table 20 — PQI Device Signature register**

Byte\Bit	7	6	5	4	3	2	1	0
0	SIGNATURE ('PQI DREG')							
...								
7								

The SIGNATURE field contains 8 bytes of ASCII data (see 4.1) set as shown in table 20 (i.e., 5051\_4920\_4452\_4547h).

### 6.2.5 Administrator Queue Configuration Function register

The Administrator Queue Configuration Function register is used to perform a PD function and to indicate the status of a PD function.

The PQI device shall set the fields defined in the Administrator Queue Configuration Function register to zero after power on, PCI Express reset, or PQI reset (see 5.7).

Register attributes (see PCIe) are as defined in table 19.

Table 21 defines the Administrator Queue Configuration Function register.

**Table 21 — Administrator Queue Configuration Function register**

Byte\Bit	7	6	5	4	3	2	1	0
0	FUNCTION AND STATUS CODE							
1	RsvdZ							
...								
7								

The FUNCTION AND STATUS CODE field is used to specify the PD function to perform and to indicate the status of a PD function being performed.

Table 22 defines the FUNCTION AND STATUS CODE field for memory reads.

**Table 22 — FUNCTION AND STATUS CODE field for memory reads**

Code	Name	Description
00h	IDLE	The PQI device is not processing a PD function.
01h	CREATING ADMINISTRATOR QUEUE PAIR	The PQI device is in the process of creating the administrator queue pair (see 5.3.3.2).
02h	DELETING ADMINISTRATOR QUEUE PAIR	The PQI device is in the process of deleting the administrator queue pair (see 5.3.4.2).
All others	Reserved	

Table 23 defines the FUNCTION AND STATUS CODE field for memory writes.

**Table 23 — FUNCTION AND STATUS CODE field for memory writes**

Code	Name	Description
00h	NOP	Shall be ignored by the PQI device.
01h	CREATE ADMINISTRATOR QUEUE PAIR	The PQI device shall create the administrator queue pair (see 5.3.3.2).
02h	DELETE ADMINISTRATOR QUEUE PAIR	The PQI device shall delete the administrator queue pair (see 5.3.4.2).
All others <sup>a</sup>	Reserved	
<sup>a</sup> If the PQI host writes a reserved code value to the FUNCTION AND STATUS CODE field, then the PQI device shall set the error code to INVALID PD FUNCTION in the PQI Device Error register (see 6.2.18). See 5.5.4.5 and 5.5.5.5 in the PD state machine. <sup>b</sup> If the PQI device is processing a PD function, then the PQI device shall set the error code to INVALID PD FUNCTION in the PQI Device Error register. See 5.5.4.5 and 5.5.5.5 in the PD state machine. <sup>c</sup> If the administrator queue pair has not been created and the DELETE ADMINISTRATOR QUEUE PAIR PD function is requested, then PQI device shall set the error code to ERROR DELETING ADMINISTRATOR QUEUE PAIR in the PQI Device Error register. See 5.5.4.5 in the PD state machine. <sup>d</sup> If the administrator queue pair has already been created and the CREATE ADMINISTRATOR QUEUE PAIR PD function is requested, then PQI device shall set the error code to ERROR CREATING ADMINISTRATOR QUEUE PAIR in the PQI Device Error register. See 5.5.5.5 in the PD state machine. <sup>e</sup> If all operational queues have not been deleted and the DELETE ADMINISTRATOR QUEUE PAIR PD function is requested, then PQI device shall set the error code to ERROR DELETING ADMINISTRATOR QUEUE PAIR in the PQI Device Error register. See 5.5.5.5 in the PD state machine.		

### 6.2.6 PQI Device Capability register

The PQI Device Capability register indicates the capabilities of the PQI device.

Register attributes (see PCIe) are as defined in table 19.

Table 24 defines the PQI Device Capability register.

**Table 24 — PQI Device Capability register**

Byte\Bit	7	6	5	4	3	2	1	0
0	MAXIMUM ADMINISTRATOR IQ ELEMENTS							
1	MAXIMUM ADMINISTRATOR OQ ELEMENTS							
2	ADMINISTRATOR IQ ELEMENT LENGTH							
3	ADMINISTRATOR OQ ELEMENT LENGTH							
4	MAXIMUM TIMEOUT FOR PQI DEVICE RESET							(LSB)
5								(MSB)
6	RsvdZ							
7								

The MAXIMUM ADMINISTRATOR IQ ELEMENTS field indicates the maximum number of administrator IQ elements (see 5.2.5.3.2) supported by the PQI device. The PQI device shall support at least two administrator IQ elements.

The MAXIMUM ADMINISTRATOR OQ ELEMENTS field indicates the maximum number of administrator OQ elements (see 5.2.5.3.2) supported by the PQI device. The PQI device shall support at least two administrator OQ elements.

The ADMINISTRATOR IQ ELEMENT LENGTH field indicates the administrator IQ element length (see 5.2.5.3.3) in 16-byte increments (e.g., 01h means 16 bytes and FFh means 4 080 bytes). The length shall be greater than or equal to the size of the largest administrator inbound IU supported by the PQI device.

The ADMINISTRATOR OQ ELEMENT LENGTH field indicates the administrator OQ element length (see 5.2.5.3.3) in 16-byte increments (e.g., 01h means 16 bytes and FFh means 4 080 bytes). The length shall be greater than or equal to the size of the largest administrator outbound IU supported by the PQI device.

The MAXIMUM TIMEOUT FOR PQI DEVICE RESET field indicates the timeout value in units of 100 ms for the PQI device to complete a PQI reset (see 5.7).

### 6.2.7 Legacy INTx Interrupt Status register

Register attributes (see PCIe) are as defined in table 19.

Table 25 defines the Legacy INTx Interrupt Status register.

**Table 25 — Legacy INTx Interrupt Status register**

Byte\Bit	7	6	5	4	3	2	1	0
0	RsvdZ					SOURCE PENDING	INTERRUPT MASK	INTERRUPT PENDING
1	RsvdZ							
...								
3								

A SOURCE PENDING bit set to one indicates that one or more interrupt sources are asserted. A SOURCE PENDING bit set to zero indicates that no interrupt source is asserted.

An INTERRUPT MASK bit set to one indicates that the virtual INTx wire mask is enabled. An INTERRUPT MASK bit set to zero indicates that the virtual INTx wire mask is disabled.

An INTERRUPT PENDING bit set to one indicates that the virtual INTx wire is asserted by the PQI device. An INTERRUPT PENDING bit set to zero indicates that the virtual INTx wire is deasserted by the PQI device.

The SOURCE PENDING bit and the INTERRUPT PENDING bit are set to zero after power on, PCI Express reset, or PQI reset (see 5.7) as a result of all OQs being deleted. The INTERRUPT MASK bit shall be set to zero after power on, PCI Express reset, or PQI reset (see 5.7).

### 6.2.8 Legacy INTx Interrupt Mask Set register

The PQI device shall set the fields defined in the Legacy INTx Interrupt Mask Set register to zero at the completion of a power on, PCI Express reset, or PQI reset (see 5.7).

Register attributes (see PCIe) are as defined in table 19.

Table 26 defines the Legacy INTx Interrupt Mask Set register.

**Table 26 — Legacy INTx Interrupt Mask Set register**

Byte\Bit	7	6	5	4	3	2	1	0
0	RsvdZ							INTERRUPT MASK SET
1	RsvdZ							
...								
3								

For memory writes:

- an INTERRUPT MASK SET bit set to one specifies that the legacy INTx interrupt shall be masked (i.e., the virtual INTx wire (see PCIe) shall not be asserted by the PQI device); and
- an INTERRUPT MASK SET bit set to zero shall be ignored.

For memory reads, the INTERRUPT MASK SET bit is the same as the INTERRUPT MASK bit in Legacy INTx Interrupt Status register (see 6.2.7).

### 6.2.9 Legacy INTx Interrupt Mask Clear register

The PQI device shall set the fields defined in the Legacy INTx Interrupt Mask Clear register to zero at the completion of a power on, PCI Express reset, or PQI reset (see 5.7).

Register attributes (see PCIe) are as defined in table 19.

Table 27 defines the Legacy INTx Interrupt Mask Clear register.

**Table 27 — Legacy INTx Interrupt Mask Clear register**

Byte\Bit	7	6	5	4	3	2	1	0
0	RsvdZ							INTERRUPT MASK CLEAR
1	RsvdZ							
...								
3								

For memory writes:

- a) an INTERRUPT MASK CLEAR bit set to one specifies that the legacy INTx interrupt shall be unmasked (i.e., the virtual INTx wire may be asserted by the PQI device); and
- b) an INTERRUPT MASK CLEAR bit set to zero shall be ignored.

For memory reads, the INTERRUPT MASK CLEAR bit is the same as the INTERRUPT MASK bit in Legacy INTx Interrupt Status register (see 6.2.7).

### 6.2.10 PQI Device Status register

The PQI Device Status register indicates the status of the PQI device.

Register attributes (see PCIe) are as defined in table 19.

Table 28 defines the PQI Device Status register.

**Table 28 — PQI Device Status register**

Byte\Bit	7	6	5	4	3	2	1	0
0	RsvdZ				PQI DEVICE STATE			
1	RsvdZ						OP IQ ERROR	OP OQ ERROR
2	RsvdZ							
3								

Table 29 defines the PQI DEVICE STATE field.

**Table 29 — PQI DEVICE STATE field**

Code	PD state machine state <sup>a</sup>
0h	PD0:Power_On_And_Reset
1h	PD1:PQI_Status_Available
2h	PD2:All_Registers_Ready
3h	PD3:Administrator_Queue_Pair_Ready
4h	PD4:Error
All others	Reserved
<sup>a</sup> Refer to section 5.5.1 for the PD state machine.	

The PQI device sets the PQI DEVICE STATE field to 0h (i.e., PD0:Power\_On\_And\_Reset) after power on and PCI Express reset.

The PQI device sets the PQI DEVICE STATE field to 1h (i.e., PD1:PQI\_Status\_Available) after PQI reset completion (see 5.7).

An OP IQ ERROR bit set to one indicates that the PQI device has stopped consuming from one or more operational IQs due to an error. An OP IQ ERROR bit set to zero indicates that the PQI device has not stopped consuming from one or more operational IQs due to an error. If no operational IQ is in error, then the PQI device shall set the OP IQ ERROR bit to zero.

An OP OQ ERROR bit set to one indicates that the PQI device has stopped producing to one or more operational OQs due to an error. An OP OQ ERROR bit set to zero indicates that the PQI device has not stopped producing to one or more operational OQs due to an error. If no operational OQ is in error, then the PQI device shall set the OP OQ ERROR bit to zero.

The OP IQ ERROR bit and OP OQ ERROR bit shall be set to zero after power on, PCI Express reset, or PQI reset (see 5.7).

### 6.2.11 Administrator IQ PI Offset register

The Administrator IQ PI Offset register is updated during administrator queue pair creation (see 5.3.3.2) and during administrator queue pair deletion (see 5.3.4.2).

The PQI device shall set the fields defined in the Administrator IQ PI Offset register to zero at the completion of a power on, PCI Express reset, or PQI reset (see 5.7).

Register attributes (see PCIe) are as defined in table 19.

Table 30 defines the Administrator IQ PI Offset register.

**Table 30 — Administrator IQ PI Offset register**

Byte\Bit	7	6	5	4	3	2	1	0
0	(LSB)							
...								
7	(MSB)							

The ADMINISTRATOR IQ PI OFFSET field indicates the 64-bit offset in PQI device memory space (see 6.2.1) of the administrator IQ PI (i.e., the administrator IQ PI address is the address contained in the first PCI memory BAR plus the administrator IQ PI offset). The ADMINISTRATOR IQ PI OFFSET field shall be a multiple of four (i.e., byte 0 bit 0 set to zero and byte 0 bit 1 set to zero).

### 6.2.12 Administrator OQ CI Offset register

The Administrator OQ CI Offset register is updated during administrator queue pair creation (see 5.3.3.2) and during administrator queue pair deletion (see 5.3.4.2).

The PQI device shall set the fields defined in the Administrator OQ CI Offset register to zero at the completion of a power on, PCI Express reset, or PQI reset (see 5.7).

Register attributes (see PCIe) are as defined in table 19.

Table 31 defines the Administrator OQ CI Offset register.

**Table 31 — Administrator OQ CI Offset register**

Byte\Bit	7	6	5	4	3	2	1	0
0	ADMINISTRATOR OQ CI OFFSET (LSB)							
...								
7								

The ADMINISTRATOR OQ CI OFFSET field indicates the 64-bit offset in PQI device memory space (see 6.2.1) of the administrator OQ CI (i.e., the administrator OQ CI address is the address contained in the first PCI memory BAR plus the administrator OQ CI offset). The ADMINISTRATOR OQ CI OFFSET field shall be a multiple of four (i.e., byte 0 bits 1 to 0 set to 00b).

### 6.2.13 Administrator IQ Element Array Address register

The Administrator IQ Element Array Address register is updated during administrator queue pair creation (see 5.3.3.2).

The PQI device shall set the fields defined in the Administrator IQ Element Array Address register to zero at the completion of a power on, PCI Express reset, or PQI reset (see 5.7).

Register attributes (see PCIe) are as defined in table 19.

Table 32 defines the Administrator IQ Element Array Address register.

**Table 32 — Administrator IQ Element Array Address register**

Byte\Bit	7	6	5	4	3	2	1	0						
0	(LSB)		RsvdZ											
...	ADMINISTRATOR IQ ELEMENT ARRAY ADDRESS													
7									(MSB)					

The ADMINISTRATOR IQ ELEMENT ARRAY ADDRESS field specifies and indicates the upper 58 bits of the 64-bit administrator IQ element array address. The least significant six bits of the 64-bit administrator IQ element array address, which are not specified by the ADMINISTRATOR IQ ELEMENT ARRAY ADDRESS field, are zero.

### 6.2.14 Administrator OQ Element Array Address register

The Administrator OQ Element Array Address register is updated during administrator queue pair creation (see 5.3.3.2).

The PQI device shall set the fields defined in the Administrator OQ Element Array Address register to zero at the completion of a power on, PCI Express reset, or PQI reset (see 5.7).

Register attributes (see PCIe) are as defined in table 19.

Table 33 defines the Administrator OQ Element Array Address register.

**Table 33 — Administrator OQ Element Array Address register**

Byte\Bit	7	6	5	4	3	2	1	0						
0	(LSB)		RsvdZ											
...	ADMINISTRATOR OQ ELEMENT ARRAY ADDRESS													
7									(MSB)					

The ADMINISTRATOR OQ ELEMENT ARRAY ADDRESS field specifies and indicates the upper 58 bits of the 64-bit administrator OQ element array address (see 6.2.1). The least significant six bits of the 64-bit administrator OQ element array address, which are not specified by the ADMINISTRATOR OQ ELEMENT ARRAY ADDRESS field, are zero.

### 6.2.15 Administrator IQ CI Address register

The Administrator IQ CI Address register is updated during administrator queue pair creation (see 5.3.3.2).

The PQI device shall set the fields defined in the Administrator IQ CI Address register to zero at the completion of a power on, PCI Express reset, or PQI reset (see 5.7).

Register attributes (see PCIe) are as defined in table 19.



Table 34 defines the Administrator IQ CI Address register.

**Table 34 — Administrator IQ CI Address register**

Byte\Bit	7	6	5	4	3	2	1	0
0	ADMINISTRATOR IQ CI ADDRESS						(LSB)	
...							RsvdZ	
7								

The ADMINISTRATOR IQ CI ADDRESS field specifies and indicates the upper 62 bits of the 64-bit administrator IQ CI address. The least significant two bits of the 64-bit administrator IQ CI address, which are not specified by the ADMINISTRATOR IQ CI ADDRESS field, are zero.

#### 6.2.16 Administrator OQ PI Address register

The Administrator OQ PI Address register is updated during administrator queue pair creation (see 5.3.3.2) and during administrator queue pair deletion (see 5.3.4.2).

The PQI device shall set the fields defined in the Administrator OQ PI Address register to zero at the completion of a power on, PCI Express reset, or PQI reset (see 5.7).

Register attributes (see PCIe) are as defined in table 19.

Table 35 defines the Administrator OQ PI Address register.

**Table 35 — Administrator OQ PI Address register**

Byte\Bit	7	6	5	4	3	2	1	0
0	ADMINISTRATOR OQ PI ADDRESS						(LSB)	
...							RsvdZ	
7								

The ADMINISTRATOR OQ PI ADDRESS field specifies and indicates the upper 62 bits of the 64-bit administrator OQ PI address. The least significant two bits of the 64-bit administrator OQ CI address, which are not specified by the ADMINISTRATOR OQ PI ADDRESS field, are zero.

#### 6.2.17 Administrator Queue Parameter register

The Administrator Queue Parameter register is updated during administrator queue pair creation (see 5.3.3.2).

The PQI device shall set the fields defined in the Administrator Queue Parameter register to zero at the completion of a power on, PCI Express reset, or PQI reset (see 5.7).

Register attributes (see PCIe) are as defined in table 19.

Table 36 defines the Administrator Queue Parameter register.

**Table 36 — Administrator Queue Parameter register**

Byte\Bit	7	6	5	4	3	2	1	0
0	NUMBER OF ADMINISTRATOR IQ ELEMENTS							
1	NUMBER OF ADMINISTRATOR OQ ELEMENTS							
2	INTERRUPT MESSAGE NUMBER (LSB)							
3	RsvdZ				(MSB)			

The NUMBER OF ADMINISTRATOR IQ ELEMENTS field specifies and indicates the number of elements in the administrator IQ. This field shall be set to:

- a) a value less than or equal to the value contained in the MAXIMUM ADMINISTRATOR IQ ELEMENTS field in the PQI Device Capability register (see 6.2.6); and
- b) at least 02h.

If the NUMBER OF ADMINISTRATOR IQ ELEMENTS field is set to an invalid value, then the PQI device sets the error code to INVALID PARAMETER FOR PD FUNCTION in the PQI Device Error register (see 6.2.18 and 5.5.5.5).

The NUMBER OF ADMINISTRATOR OQ ELEMENTS field specifies and indicates the number of elements in the administrator OQ. This field shall be set to:

- a) a value less than or equal to the value contained in the MAXIMUM ADMINISTRATOR OQ ELEMENTS field in the PQI Device Capability register (see 6.2.6); and
- b) at least 02h.

If the NUMBER OF ADMINISTRATOR OQ ELEMENTS field is set to an invalid value, then the PQI device sets the error code to INVALID PARAMETER FOR PD FUNCTION in the PQI Device Error register (see 6.2.18 and 5.5.5.5).

The INTERRUPT MESSAGE NUMBER field specifies and indicates the MSI-X Table entry used to generate the interrupt message for updates to the administrator OQ PI in MSI-X mode.

If the INTERRUPT MESSAGE NUMBER field is larger than the MSI-X Table, then the PQI device sets the error code to INVALID PARAMETER FOR PD FUNCTION in the PQI Device Error register (see 6.2.18 and 5.5.5.5).

### 6.2.18 PQI Device Error register

The PQI Device Error register indicates the cause or the source of an error (see 5.6) in the PQI device. The PQI Device Error register value is valid while the PD state machine is in the PD4:Error state (see 5.5.6).

Register attributes (see PCIe) are as defined in table 19.

Table 37 defines the PQI Device Error register.

**Table 37 — PQI Device Error register**

Byte\Bit	7	6	5	4	3	2	1	0
0	ERROR CODE							
1	ERROR CODE QUALIFIER							
2	BYTE POINTER							
3	ERROR DETAILS REGISTER VALID	BIT POINTER			RsvdZ			

The ERROR CODE field indicates generic information describing a reported condition (see table 18).

The ERROR CODE QUALIFIER field indicates further information (see table 18) related to the condition reported in the ERROR CODE field.

If defined as valid for the error code (see table 18), then the BYTE POINTER field indicates the byte offset in the memory space of the PQI device registers described by the first PCI memory BAR that contains the field with the invalid value.

An ERROR DETAILS REGISTER VALID bit set to one indicates that there is valid error specific data (see table 18) in the PQI Device Error Details register. An ERROR DETAILS REGISTER VALID bit set to zero indicates that there is no valid error specific data in the PQI Device Error Details register (see 6.2.19).

If defined as valid for the error code (see table 18), the BIT POINTER field indicates the offset in the PQI device register byte of the first bit (i.e., the lowest bit number) containing the field with the invalid value.

The PQI device shall set the ERROR CODE field and the ERROR CODE QUALIFIER field to:

- a) NO ERROR after power on;
- b) NO ERROR after PCI Express reset; or
- c) the error code that indicates the status of the PQI device (see table 18).

#### 6.2.19 PQI Device Error Details register

The PQI Device Error Details register is used to describe error details (see 5.6) in combination with the PQI Device Error register (see 6.2.18). If the ERROR DETAILS REGISTER VALID bit is set to one in the PQI Device Error register, then the PQI Device Error Details register contains valid error details. If the ERROR DETAILS REGISTER VALID bit is set to zero in the PQI Device Error register, then the PQI Device Error Details register does not contain error details.

Register attributes (see PCIe) are as defined in table 19.

Table 38 defines the PQI Device Error Details register.

**Table 38 — PQI Device Error Details register**

Byte\Bit	7	6	5	4	3	2	1	0
0	Error details							
...								
7								

The PQI Device Error Details register value along with the PQI Device Error register value shall be set to the values that indicate the error details of the PQI device after power on or after PCI Express reset.

The PQI Device Error Details register value shall be set to the values that indicate the error details of the PQI device after PQI reset (see 5.7).

### 6.2.20 PQI Device Reset register

The PQI Device Reset register is used to initiate a PQI reset (see 5.7) and to provide PQI reset status.

Register attributes (see PCIe) are as defined in table 19.

Table 39 defines the PQI Device Reset register.

**Table 39 — PQI Device Reset register**

Byte\Bit	7	6	5	4	3	2	1	0
0	RESET ACTION			RsvdZ		RESET TYPE		
1	RsvdZ							HOLD IN PD1
2	RsvdZ							
3								

Table 40 defines the RESET ACTION field for memory writes.

**Table 40 — RESET ACTION field for memory writes**

Code	Name	Description
000b	NO ACTION	No action.
001b	RESET	Start processing the PQI reset action specified in the RESET TYPE field.
All others	Reserved	

Table 41 defines the RESET ACTION field for memory reads.

**Table 41 — RESET ACTION field for memory reads**

Code	Name	Description
000b	NO ACTION	No reset action has been processed.
001b	PROCESSING RESET	The PQI device is processing the reset action with the reset type specified in the RESET TYPE field.
010b	RESET COMPLETED	The PQI device has completed the reset action with the reset type specified in the RESET TYPE field.
All others	Reserved	

The RESET ACTION field shall be set to 000b (i.e., NO ACTION) after power on and after PCI Express reset. After a PQI reset, the RESET ACTION field shall be set to:

- a) 001b (i.e., PROCESSING RESET) if the PQI reset does not successfully complete; or
- b) 010b (i.e., RESET COMPLETED) if the PQI reset successfully completes.

Table 42 defines the RESET TYPE field.

**Table 42 — RESET TYPE field**

Code	Name	Description
000b	NO RESET	No reset action <sup>a</sup>
001b	SOFT RESET	PQI soft reset (see 5.7.2)
010b	FIRM RESET	PQI firm reset (see 5.7.3)
011b	HARD RESET	PQI hard reset (see 5.7.4)
All others	Reserved	
<sup>a</sup> The NO RESET reset type allows the PD state machine (see 5.5) to transition from the PD1:PQI_Status_Ready state to the PD2:All_Registers_Ready state after a previous PQI reset was performed with the HOLD IN PD1 bit set to one (see 5.7.1).		

The RESET TYPE field shall be set to 000b (i.e., NO RESET) after power on and after PCI Express reset. After a PQI reset, the RESET TYPE field shall be set to:

- a) 001b (i.e., SOFT RESET) if the PQI host previously requested a PQI soft reset;
- b) 010b (i.e., FIRM RESET) if the PQI host previously requested a PQI firm reset; or
- c) 011b (i.e., HARD RESET) if the PQI host previously requested a PQI hard reset.

A HOLD IN PD1 bit set to one specifies that the PD state machine remains in the PD1:PQI\_Status\_Available state (see 5.5.3) following a PQI reset. A HOLD IN PD1 bit set to zero specifies that the PD state machine is not held in the PD1:PQI\_Status\_Available state following a PQI reset.

The PQI device shall set the HOLD IN PD1 bit to zero after power on and after PCI Express reset.

The PQI device shall set the HOLD IN PD1 bit to:

- a) zero if the PQI host previously requested a PQI reset with the HOLD IN PD1 bit to zero; or
- b) one if the PQI host previously requested a PQI reset with the HOLD IN PD1 bit to one.

#### 6.2.21 PQI Device Power Action register

The PQI Device Power Action register is used to notify the PQI device about upcoming operating system-directed power management transitions and device power state transitions.

The PQI device shall set the fields defined in PQI Device Power Action register to zero at the completion of a power on, PCI Express reset, or PQI reset (see 5.7).

Register attributes (see PCIe) are as defined in table 19.

Table 43 defines the PQI Device Power Action register.

**Table 43 — PQI Device Power Action register**

Byte\Bit	7	6	5	4	3	2	1	0
0	POWER ACTION		SYSTEM POWER ACTION					
1	RsvdZ			DEVICE POWER ACTION				
2	RsvdZ							
3								

Table 44 defines the POWER ACTION field for memory writes.

**Table 44 — POWER ACTION field for memory writes**

Code	Description
00b	No action
01b	Process the power action specified in the SYSTEM POWER ACTION field and the DEVICE POWER ACTION field.  If the PQI device receives a write specifying a different power action while processing a previous power action, then the results are vendor specific.
All others	Reserved

Table 45 defines the POWER ACTION field for memory reads.

**Table 45 — POWER ACTION field for memory reads**

Code	Description
00b	No power action has been processed.
01b	The PQI device is processing the power action indicated in the SYSTEM POWER ACTION field and the DEVICE POWER ACTION field.
10b	The PQI device has completed processing the power action indicated in the SYSTEM POWER ACTION field and the DEVICE POWER ACTION field.
11b	Reserved

Table 46 defines the SYSTEM POWER ACTION field.

**Table 46 — SYSTEM POWER ACTION field**

Code	Description
General notifications (00h to 0Fh)	
00h	No system power action notification (e.g., system remains in ACPI G0 (S0) (see ACPI))
01h	Operating system is going to shutdown and reboot (e.g., system remains in ACPI G0 (S0))
02h	Operating system is going to either: a) shutdown and reboot (e.g., system remains in ACPI G0 (S0)); or b) shutdown and enter ACPI G2 (see ACPI) (i.e., soft off)
03h to 0Fh	Reserved
Notifications that the system is going to sleep (10h to 1Fh)	
10h	System is going to enter ACPI G1 (S1, S2, or S3) (see ACPI) (i.e., sleeping)
11h	System is going to enter ACPI G1 (S1) (i.e., sleeping)
12h	System is going to enter ACPI G1 (S2) (i.e., sleeping)
13h	System is going to enter ACPI G1 (S3) (i.e., sleeping)
14h	System is going to enter ACPI G1 (S4) (i.e., hibernating)
15h	System is going to enter ACPI G2 (S5) (i.e., soft off)
16h to 1Fh	Reserved
Notifications that the system is back from sleep (20h to 2Fh)	
20h	System is back from ACPI G1 (S1, S2, or S3) (i.e., sleeping)
21h	System is back from ACPI G1 (S1) (i.e., sleeping)
22h	System is back from ACPI G1 (S2) (i.e., sleeping)
23h	System is back from ACPI G1 (S3) (i.e., sleeping)
24h	System is back from ACPI G1 (S4) (i.e., hibernating)
25h to 2Fh	Reserved
Other (30h to 3Fh)	
30h to 3Fh	Reserved

Table 47 defines the DEVICE POWER ACTION field.

**Table 47 — DEVICE POWER ACTION field**

Code	Description
General notifications (00h to 0Fh)	
00h	No device power action notification
01h to 0Fh	Reserved
Notifications that the device is going to be requested to change device state (10h to 1Fh)	
10h	System is going to request that the device enter D0 (see ACPI and PCI-PM) (i.e., on)
11h	System is going to request that the device enter D1 (see ACPI and PCI-PM)
12h	System is going to request that the device enter D2 (see ACPI and PCI-PM)
13h	System is going to request that the device enter D3 <sub>hot</sub> or D3 (i.e., off) (see ACPI and PCI-PM)
14h to 1Fh	Reserved

The PowerState field in the PCI Power Management Capabilities Power Management Control/Status register (i.e., PMCSR) (see PCI-PM) is used to set the PQI device to a new device power state (e.g., D0, D1, D2, or D3<sub>hot</sub>) after the device power action notification.

This standard does not define the specific actions that a PQI device should take to prepare for the specified power actions (e.g., flush caches, terminate background tasks, spin down rotating media on a notification that the system is going to sleep, or resume background tasks and spin up rotating media if the system is back from sleep).

The PQI device shall continue to process administrator queues and operational queues after being notified of an upcoming power action.



## 7 Queuing layer

### 7.1 IQ CI, IQ PI, OQ CI, and OQ PI structures

#### 7.1.1 IQ CI dword

An IQ CI dword is a dword in memory space that contains an IQ CI (see 5.2.5.7).

Table 48 defines the structure of the IQ CI dword.

**Table 48 — IQ CI dword**

Byte\Bit	7	6	5	4	3	2	1	0
0	IQ CI							(LSB)
1								(MSB)
2	Reserved or RsvdZ							
3								

The IQ CI field contains the IQ CI. It is not an error to set the IQ CI field to the same value that it already contains.

Bits that are defined as reserved or RsvdZ are processed as:

- a) RsvdZ if the IQ CI dword is implemented as a register (e.g., in PQI device memory space); or
- b) reserved if the IQ CI dword is not implemented as a register (e.g., in PQI host memory space).

#### 7.1.2 IQ PI register

An IQ PI register is a register in PQI device memory space that contains an IQ PI (see 5.2.5.7).

Table 49 defines the structure of the IQ PI register.

**Table 49 — IQ PI register**

Byte\Bit	7	6	5	4	3	2	1	0
0	IQ PI							(LSB)
1								(MSB)
2	RsvdZ							
3								

The IQ PI field contains the IQ PI. It is not an error to set the IQ PI field to the same value that it already contains.

#### 7.1.3 OQ CI register

An OQ CI register is a register in PQI device memory space that contains an OQ CI (see 5.2.5.10).

Table 50 defines the structure of the OQ CI register.

**Table 50 — OQ CI register**

Byte\Bit	7	6	5	4	3	2	1	0
0	OQ CI (LSB)							
1								
2	RsvdZ							
3	REARM INTERRUPT	RsvdZ						

The OQ CI field contains the OQ CI. It is not an error to set the OQ CI field to the same value that it already contains (e.g., to set the REARM INTERRUPT bit to one).

When written, a REARM INTERRUPT bit set to:

- a) one specifies that the PQI device shall rearm the interrupt associated with the OQ (see 5.4.2.3); and
- b) zero shall be ignored.

When read, the PQI device shall return the REARM INTERRUPT bit set to zero.

#### 7.1.4 OQ PI dword

An OQ PI dword is a dword in memory space that contains an OQ PI (see 5.2.5.10).

Table 51 defines the structure of the OQ PI dword.

**Table 51 — OQ PI dword**

Byte\Bit	7	6	5	4	3	2	1	0
0	OQ PI (LSB)							
1								
2	Reserved or RsvdZ							
3								

The OQ PI field contains the OQ PI. It is not an error to set the OQ PI field to the same value that it already contains.

Bits that are defined as reserved or RsvdZ are processed as:

- a) RsvdZ if the OQ PI dword is implemented as a register (e.g., in PQI device memory space); or
- b) reserved if the OQ PI dword is not implemented as a register (e.g., in PQI host memory space).

## 8 SGL (scatter gather list)

### 8.1 SGL overview

An SGL is a data structure used to describe a data buffer. A data buffer is either a source data buffer or a destination data buffer. An SGL contains one or more SGL segments, ending with a last SGL segment.

SGL errors are processed using a method defined in the IU layer standard (e.g., SOP).

### 8.2 Standard SGL segment and last standard SGL segment

A standard SGL segment:

- a) is a data structure in a contiguous region of memory space describing all, part, or none of a data buffer and the next SGL segment, if any; and
- b) consists of an array of one or more SGL descriptors (see 8.3).

If the array has more than one SGL descriptor, then each SGL descriptor before to the last SGL descriptor shall not be:

- a) a Standard SGL Segment descriptor (see 8.3.4);
- b) a Last Standard SGL Segment descriptor (see 8.3.5); or
- c) a Last Alternative SGL Segment descriptor (see Annex A).

A last standard SGL segment is a standard SGL segment that does not contain:

- a) a Standard SGL Segment descriptor (see 8.3.4);
- b) a Last Standard SGL Segment descriptor (see 8.3.5); or
- c) a Last Alternative SGL Segment descriptor (see Annex A).

An SGL shall be processed as having an error (see 10.1.4 and the IU layer standard (e.g., SOP)) if it contains:

- a) a standard SGL segment that contains more than one of the following:
  - A) a Standard SGL Segment descriptor;
  - B) a Last Standard SGL Segment descriptor; or
  - C) a Last Alternative SGL Segment descriptor;

or

- b) a last standard SGL segment that contains one of the following:
  - A) Standard SGL Segment descriptor;
  - B) Last Standard SGL Segment descriptor; or
  - C) Last Alternative SGL Segment descriptor.

Table 52 defines the standard SGL segment.

**Table 52 — Standard SGL segment**

Byte\Bit	7	6	5	4	3	2	1	0
0	SGL descriptor [first] (see table 53)							
...								
15								
...	...							
n - 15	SGL descriptor [last] (see table 53)							
...								
n								

The standard SGL segment contains one or more SGL descriptors (see 8.3).

## 8.3 SGL descriptors

### 8.3.1 SGL descriptors overview

Table 53 defines the SGL descriptor format.

**Table 53 — SGL descriptor format**

Byte\Bit	7	6	5	4	3	2	1	0
0	Descriptor type specific							
...								
14								
15	SGL DESCRIPTOR TYPE				Descriptor type specific			

The SGL DESCRIPTOR TYPE field (see table 54) specifies the SGL descriptor type. If the SGL DESCRIPTOR TYPE field is set to a reserved or unsupported value, then the SGL descriptor shall be processed as having an error (see 10.1.4 and the IU layer standard (e.g., SOP)).

**Table 54 — SGL DESCRIPTOR TYPE field**

Code	Descriptor	Reference
0h	Data Block descriptor	8.3.2
1h	Bit Bucket descriptor	8.3.3
2h	Standard SGL Segment descriptor	8.3.4
3h	Last Standard SGL Segment descriptor	8.3.5
4h	Last Alternative SGL Segment descriptor	Annex A
Fh	Vendor specific	
All others	Reserved	

A SGL descriptor set to all zeros is a Data Block descriptor (see 8.3.2) with the ADDRESS field set to 00000000\_00000000h and the LENGTH field set to 00000000h and may be used in the SGL to specify no data buffer.

### 8.3.2 Data Block descriptor

The Data Block descriptor describes a data block.

Table 55 defines the Data Block descriptor.

**Table 55 — Data Block descriptor**

Byte\Bit	7	6	5	4	3	2	1	0
0	ADDRESS							(LSB)
...								
7								(MSB)
8	LENGTH							(LSB)
...								
11								(MSB)
12	Reserved							
...								
14								
15	SGL DESCRIPTOR TYPE (0h)				ZERO			

The ADDRESS field specifies the starting 64-bit memory address of the data block.

The LENGTH field specifies the length in bytes of the data block. A LENGTH field set to 00000000h specifies that no data be transferred. A Data Block descriptor specifying that no data be transferred shall not be processed as having an error.

If the value in the ADDRESS field plus the value in the LENGTH field is greater than 1\_00000000\_00000000h, then the Data Block descriptor shall be processed as having an error (see 10.1.4 and the IU layer standard (e.g., SOP)).

The ZERO field shall contain 0h. A Data Block descriptor containing a ZERO field set to a value other than 0h shall be processed as having an error (see 10.1.4 and the IU layer standard (e.g., SOP)).

The SGL DESCRIPTOR TYPE field is defined in 8.3.1 and shall be set as shown in table 55 for the Data Block descriptor.

### 8.3.3 Bit Bucket descriptor

The Bit Bucket descriptor is used to discard (i.e., skip over) parts of the source data stream if the SGL describes a destination data buffer.

Table 56 defines the Bit Bucket descriptor.

**Table 56 — Bit Bucket descriptor**

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
...								
7								
8	LENGTH (LSB)							
...								
11								
12	Reserved							
...								
14								
15	SGL DESCRIPTOR TYPE (1h)				ZERO			

If the SGL describes a destination data buffer, then the LENGTH field specifies the number of bytes of the source data stream to be discarded (i.e., the number of bytes to not be transferred to the destination data buffer). A LENGTH field set to 00000000h specifies that none of the source data stream shall be discarded and shall not be processed as having an error.

If the SGL describes a source data buffer, then the LENGTH field shall be ignored.

The ZERO field shall contain 0h. A Bit Bucket descriptor containing a ZERO field set to a value other than 0h shall be processed as having an error (see 10.1.4 and the IU layer standard (e.g., SOP)).

The SGL DESCRIPTOR TYPE field is defined in 8.3.1 and shall be set as shown in table 56 for the Bit Bucket descriptor.

### 8.3.4 Standard SGL Segment descriptor

The Standard SGL Segment descriptor describes the next SGL segment, which is a standard SGL segment (see 8.2) and may or may not be the last standard SGL segment (see 8.2).

Table 57 defines the Standard SGL Segment descriptor.

**Table 57 — Standard SGL Segment descriptor**

Byte\Bit	7	6	5	4	3	2	1	0
0	(LSB)				Reserved			
...	ADDRESS							
7								
8	(LSB)							
...	LENGTH							
11								
12	Reserved							
...								
14								
15	SGL DESCRIPTOR TYPE (2h)				ZERO			

The ADDRESS field specifies the upper 60 bits of the 64-bit memory space address of the next SGL segment, which is a standard SGL segment (see 8.2). The least significant four bits of the 64-bit memory space address of the next SGL segment, which are not specified by the ADDRESS field, are zero.

The LENGTH field specifies the length in bytes of the next SGL segment. A LENGTH field set to zero or a value that is not a multiple of 16 shall be processed as having an error (see 10.1.4 and the IU layer standard (e.g., SOP)).

If the value in the ADDRESS field plus the value in the LENGTH field is greater than 1\_00000000\_00000000h, then the Standard SGL Segment descriptor shall be processed as having an error (see 10.1.4 and the IU layer standard (e.g., SOP)).

The ZERO field shall contain 0h. A Standard SGL Segment descriptor containing a ZERO field set to a value other than 0h shall be processed as having an error (see 10.1.4 and the IU layer standard (e.g., SOP)).

The SGL DESCRIPTOR TYPE field is defined in 8.3.1 and shall be set as shown in table 57 for the Standard SGL Segment descriptor.

### 8.3.5 Last Standard SGL Segment descriptor

The Last Standard SGL Segment descriptor describes the next SGL segment, which is a last standard SGL segment (see 8.2).

Table 58 defines the Last Standard SGL Segment descriptor.

**Table 58 — Last Standard SGL Segment descriptor**

Byte\Bit	7	6	5	4	3	2	1	0	
0	(LSB)				Reserved				
...	ADDRESS								
7									(MSB)
8									(LSB)
...	LENGTH								
11									(MSB)
12									Reserved
...									
14									
15	SGL DESCRIPTOR TYPE (3h)				ZERO				

The ADDRESS field specifies the upper 60 bits of the 64-bit memory space address of the next SGL segment, which is the last SGL segment and is a standard SGL segment (see 8.2). The least significant four bits of the 64-bit memory space address of the next SGL segment, which are not specified by the ADDRESS field, are zero.

The LENGTH field specifies the length in bytes of the SGL segment. A LENGTH field set to zero or a value that is not a multiple of 16 shall be processed as having an error (see 10.1.4 and the IU layer standard (e.g., SOP)).

If the value in the ADDRESS field plus the value in the LENGTH field is greater than 1\_00000000\_00000000h, then the Last Standard SGL Segment descriptor shall be processed as having an error (see 10.1.4 and the IU layer standard (e.g., SOP)).

The ZERO field shall contain 0h. A Last Standard SGL Segment descriptor containing a ZERO field set to a value other than 0h shall be processed as having an error (see 10.1.4 and the IU layer standard (e.g., SOP)).

The SGL DESCRIPTOR TYPE field is defined in 8.3.1 and shall be set as shown in table 58 for the Last Standard SGL Segment descriptor.

## 9 Common properties for all IU layers

### 9.1 Overview of common properties for all IU layers

The PQI circular queue provides the following common IU layer properties:

- a) support for IUs that span across multiple elements; and
- b) IU header that is consistent across different IU layers.

### 9.2 IUs and elements

#### 9.2.1 IUs and elements overview

An IU contains a header (see table 59) followed by information defined by the IU type.

The size of an IU is defined by the IU length as follows:

- a) if the IU is less than or equal to the size of an element, then the IU is contained within a single element (see 9.2.2);
- b) if an IU is larger than the size of an element and IUs that span multiple elements in the circular queue is supported, then the IU spans across multiple consecutive elements (see 9.2.3); or
- c) if an IU is larger than the size of an element and IUs that span multiple elements in the circular queue is not supported, then the producer shall not place the IU in the circular queue.

#### 9.2.2 IU contained within a single element

Figure 29 shows an example of an IU that fits within a single element.

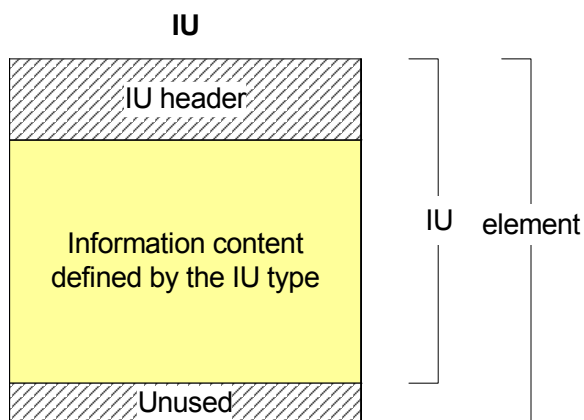


Figure 29 — Example of an IU that fits within a single element

#### 9.2.3 IU that spans multiple elements

When two or more elements are spanned to form a larger IU, only the first element includes the IU header. PQI host should not enqueue an IU greater than  $n-1$  elements, where  $n$  is the size of the queue in elements.



Figure 30 shows an example of an IU spanning across multiple elements.

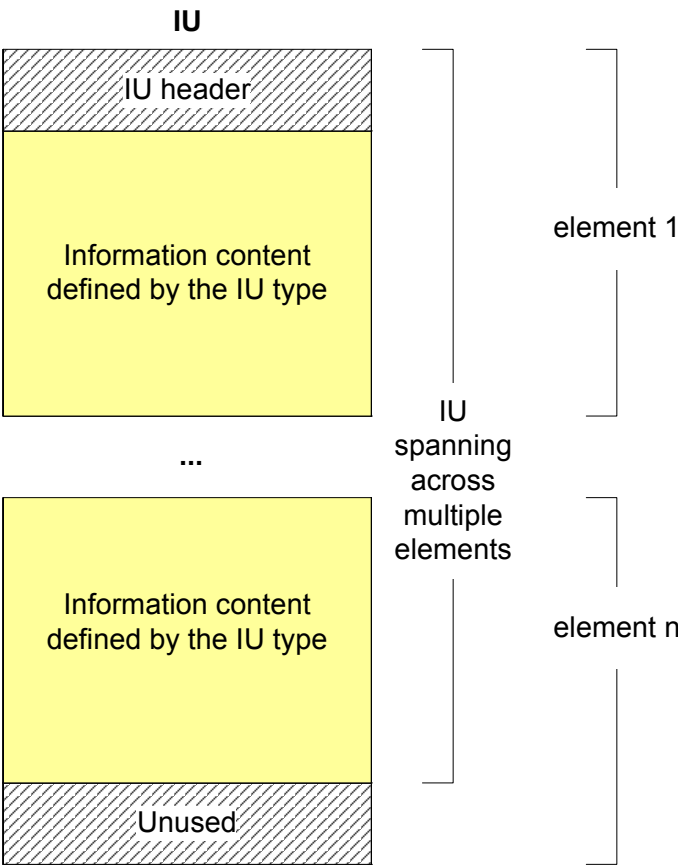


Figure 30 — Example of an IU spanning across multiple elements

The administrator queue pair does not support IU spanning multiple elements. The support of IU spanning multiple elements for the operational queue is indicated by the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3).

9.3 Common IU header for all IU layers

Table 59 defines the common IU header for all IU layers.

Byte labels for the IU header are shaded in tables defining IUs.

Table 59 — Common IU header for all IU layers

Byte\Bit	7	6	5	4	3	2	1	0
0	Restricted for IU layer							
1	PQI COMPATIBLE FEATURES (0h)				Restricted for IU layer			
2	(LSB)							
3	(MSB)							

The restricted for IU layer bits are defined by the IU layer (e.g., for PQI see 10.2.1).

The PQI COMPATIBLE FEATURES field shall be set to the value shown in table 59. The recipient of the IU shall ignore the PQI COMPATIBLE FEATURES field.

The IU LENGTH field contains the number of bytes that follow in the IU (i.e., the IU LENGTH field does not include the number of bytes in the IU header).

## 10 Administrator IUs and administrator functions

### 10.1 IU definition

#### 10.1.1 IU definition overview

Table 60 defines the PQI IUs.

**Table 60 — PQI IUs (IU TYPE field)**

Code	IU	Minimum length (bytes)	Queue type	Reference
00h	NULL IU	4	IO	10.1.3
Request IUs (01h to 7Fh)				
01h to 5Fh	Reserved			
Administrator request IUs (60h to 6Fh)				
60h	GENERAL ADMIN REQUEST IU	64	I	10.1.4
61h to 6Fh	Reserved			
Vendor-specific administrator request IUs (70h to 7Fh)				
70h to 7Fh	Vendor specific			
Response IUs (80h to FFh)				
80h to DFh	Reserved			
Administrator response IUs (E0h to EFh)				
E0h	GENERAL ADMIN RESPONSE IU	64	O	10.1.5
E1h to EFh	Reserved			
Vendor-specific administrator response IUs (F0h to FFh)				
F0h to FFh	Vendor specific			
<b>Key:</b> I = inbound queue O = outbound queue IO = inbound queue and outbound queue				

### 10.1.2 Administrator IU header

Table 61 defines the administrator IU header, which is compatible with the common IU header defined in 9.3.

**Table 61 — Administrator IU header**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	(LSB)							
3	(MSB)							

The IU TYPE field contains the PQI IU type (see table 60).

The PQI COMPATIBLE FEATURES field contains bits set aside for new functionality, if any, defined by future versions of this standard.

The PQI COMPATIBLE FEATURES field shall be:

- a) set to the value shown in table 61 for the administrator IU header; and
- b) ignored by the recipient.

The IU LENGTH field contains the number of bytes that follow in the IU.

Table 62 defines how a PQI device and a PQI host handle errors in the Administrator IU header.

**Table 62 — Administrator IU header error handling**

<b>Administrator IU header problem</b>	<b>PQI device handling</b>	<b>PQI host handling</b>
The IU TYPE field is set to a reserved or unsupported value.	<p>The PQI device shall:</p> <ul style="list-style-type: none"> <li>a) stop consuming from the administrator IQ; and</li> <li>b) set the error code to INVALID IU TYPE IN GENERAL ADMIN REQUEST IU in the PQI Device Error register (see 6.2.18).</li> </ul> <p>The PD state machine transitions as described in 5.5.5.5.</p>	<p>The PQI host should:</p> <ul style="list-style-type: none"> <li>a) stop consuming from the administrator OQ; and</li> <li>b) delete the administrator queue pair.</li> </ul>
<p>The IU LENGTH field is set to a supported value and either:</p> <ul style="list-style-type: none"> <li>a) the IU LENGTH field bits 1 to 0 are not set to 00b (i.e., the IU length is not a multiple of four);</li> <li>b) the IU LENGTH field is greater than the value indicated in the: <ul style="list-style-type: none"> <li>A) ADMINISTRATOR IQ ELEMENT LENGTH field in the PQI Device Capability register (see 6.2.6) for request IUs; or</li> <li>B) ADMINISTRATOR OQ ELEMENT LENGTH field in the PQI Device Capability register (see 6.2.6) for response IUs;</li> </ul> </li> </ul> <p>or</p> <ul style="list-style-type: none"> <li>c) the IU LENGTH field is less than the minimum length in bytes specified in table 60 minus four.</li> </ul>	<p>The PQI device shall:</p> <ul style="list-style-type: none"> <li>a) stop consuming from the administrator IQ; and</li> <li>b) set the error code to INVALID IU LENGTH IN GENERAL ADMIN REQUEST IU in the PQI Device Error register (see 6.2.18).</li> </ul> <p>The PD state machine transitions as described in 5.5.5.5.</p>	<p>The PQI host should:</p> <ul style="list-style-type: none"> <li>a) stop consuming from the administrator OQ; and</li> <li>b) delete the administrator queue pair.</li> </ul>

### 10.1.3 NULL IU

The NULL IU specifies no operation.

Table 63 defines the NULL IU.

**Table 63 — NULL IU**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (00h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	(LSB)							
3	(MSB)							

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are part of the common IU header for all IU layers (see 9.3) and shall be set as shown in table 63 for the NULL IU.

### 10.1.4 GENERAL ADMIN REQUEST IU

A GENERAL ADMIN REQUEST IU contains an administrator function request and is sent by a PQI host to a PQI device to deliver that request from a PQI host management application client to a PQI device management device server (see 10.2).

If:

- a) a RsvdC bit or byte is set to a value other than zero in administrator function request in a GENERAL ADMIN REQUEST IU; or
- b) a defined field is set to a reserved value or unsupported value in administrator function request in a GENERAL ADMIN REQUEST IU after the IU header,

then the PQI device management device server shall return a GENERAL ADMIN RESPONSE IU with the STATUS field set to INVALID FIELD IN REQUEST IU (see 10.1.5.1).

Table 64 defines the GENERAL ADMIN REQUEST IU.

**Table 64 — GENERAL ADMIN REQUEST IU**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (60h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6								
7	WORK AREA							
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE							
11	Function specific fields							
...								
43								
44	DATA-IN BUFFER SIZE (if any), DATA-OUT BUFFER SIZE (if any), or function specific fields (LSB)							
...								
47								
48	SGL descriptor (if any) (see table 53) or function specific fields							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are part of the common IU header for all IU layers (see 9.3) and shall be set as shown in table 64 for the GENERAL ADMIN REQUEST IU.

The WORK AREA field may be set to any value by the PQI host and shall be ignored by the PQI device.

The REQUEST IDENTIFIER field specifies a request identifier that the PQI device shall use in the response IUs for this request IU. The request identifier allows the PQI host to determine the context for each response IU. The PQI device management device server shall check the request identifiers for overlaps with other administrator functions being processed. If the PQI device management device server detects an overlap, then the PQI device management device server shall:

- a) abort the old administrator function with that request identifier;
- b) terminate the new administrator function with that request identifier; and
- c) return a single GENERAL ADMIN RESPONSE IU with:
  - A) the FUNCTION CODE field set to the function code specified in the new GENERAL ADMIN REQUEST IU; and
  - B) the STATUS field set to OVERLAPPED REQUEST IDENTIFIER ATTEMPTED (see 10.1.5.1).

The FUNCTION CODE field specifies which administrator function is being requested and is defined in 10.2.1.

The function specific fields are defined by the specific administrator function being processed.

The DATA-IN BUFFER SIZE field, if any, specifies the size in bytes of the Data-In Buffer. The PQI device shall terminate transfers to the Data-In Buffer when the number of bytes specified by the DATA-IN BUFFER SIZE field have been transferred or when all available data has been transferred, whichever is less. A DATA-IN BUFFER SIZE field set to zero specifies that no data shall be transferred and is not an error unless otherwise specified for the administrator function. If the information being transferred to the Data-In Buffer includes fields containing the number of bytes to be transferred in some or all of the data, then the contents of these fields shall not be altered to reflect the truncation, if any, that results from an insufficient Data-In Buffer size.

The DATA-OUT BUFFER SIZE field, if any, specifies the size in bytes of the Data-Out Buffer. A DATA-OUT BUFFER SIZE field set to zero specifies that no data shall be transferred and is not an error unless otherwise specified for the administrator function.

The SGL descriptor, if any, contains the first standard SGL segment (see 8.3) of an SGL describing:

- a) for a read administrator function, the Data-In Buffer as a destination data buffer; or
- b) for a write administrator function, the Data-Out Buffer as a source data buffer.

Table 65 defines the SGL descriptor type support requirements for administrator request IUs.

**Table 65 — SGL descriptor type support requirements for administrator request IUs**

Code <sup>a</sup>	SGL descriptor	Support	Reference
0h	Data Block descriptor	M	8.3.2
1h	Bit Bucket descriptor	M	8.3.3
2h	Standard SGL Segment descriptor	M	8.3.4
3h	Last Standard SGL Segment descriptor	M	8.3.5
4h	Last Alternative SGL Segment descriptor	O	Annex A
Fh	Vendor specific		
Support key: M = Support is mandatory. O = Support is optional.			
<sup>a</sup> The REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3) reports the SGL descriptor types that are supported by a PQI device. The PQI host management application client should only use the mandatory SGL descriptor types until it retrieves the REPORT PQI DEVICE CAPABILITY parameter data (e.g., to send the REPORT PQI DEVICE CAPABILITY function itself).			

If the PQI device management device server performs reserved field checking on the SGL descriptor contained in the IU and a reserved bit or byte in the SGL descriptor is set to a value other than zero, then the PQI device management device server shall return a GENERAL ADMIN RESPONSE with the STATUS field set to INVALID FIELD IN REQUEST IU (see 10.1.5.1 and 10.1.5.3).

If the PQI device management device server performs reserved field checking on an SGL descriptor in the SGL that is not contained in the IU and a reserved bit or byte in that SGL descriptor is set to a value other than zero, then the PQI device management device server shall return a GENERAL ADMIN RESPONSE IU with the STATUS field set to DATA BUFFER ERROR (see 10.1.5.1).

If the PQI device management device server processes an SGL that has an error (see 8.2) or an SGL descriptor that has an error (see 8.3 and Annex A), then the PQI device management device server shall return a GENERAL ADMIN RESPONSE IU with the STATUS field set to DATA BUFFER ERROR.

For a read administrator function, if the PQI device management device server transfers fewer bytes than specified in the DATA-IN BUFFER SIZE field, then the PQI device management device server shall return a



GENERAL ADMIN RESPONSE IU with the STATUS field set to DATA-IN BUFFER UNDERFLOW (see 10.1.5.1 and 10.1.5.2). For a write administrator function, no indication is provided that the PQI device management device server transferred fewer bytes than specified in the DATA-OUT BUFFER SIZE field.

If the PQI device management device server attempts to transfer beyond the length of the Data-In Buffer or Data-Out Buffer described by the SGL, then the PQI device management device server shall return a GENERAL ADMIN RESPONSE IU with the STATUS field set to DATA BUFFER OVERFLOW (see 10.1.5.1).

While accessing the Data-In Buffer or Data-Out Buffer, if the PQI device management device server encounters:

- 1) a PCI Express error described in table 66, then the PQI device management device server shall return a GENERAL ADMIN RESPONSE IU with the STATUS field set to the corresponding value described in table 66;
- 2) a PCI Express error not described in table 66, then the PQI device management device server shall return a GENERAL ADMIN RESPONSE IU with the STATUS field set to PCIE FABRIC ERROR; and
- 3) an error that the management server is not able to determine is a PCI Express error or not, then the PQI device management device server shall return a GENERAL ADMIN RESPONSE IU with the STATUS field set to DATA BUFFER ERROR.

Table 66 defines the STATUS field values for PCI Express errors while accessing the Data-In Buffer or Data-Out Buffer.

**Table 66 — STATUS field values for PCI Express errors that occur while accessing the Data-In Buffer or the Data-Out Buffer**

PCI Express error	STATUS field
PCI Express completion timeout	PCIE COMPLETION TIMEOUT
PCI Express completer abort	PCIE COMPLETER ABORT
PCI Express poisoned TLP received	PCIE POISONED TLP RECEIVED
PCI Express ECRC check failed	PCIE ECRC CHECK FAILED
PCI Express unsupported request	PCIE UNSUPPORTED REQUEST
PCI Express ACS violation	PCIE ACS VIOLATION
PCI Express TLP prefix blocked	PCIE TLP PREFIX BLOCKED

## 10.1.5 GENERAL ADMIN RESPONSE IU

### 10.1.5.1 GENERAL ADMIN RESPONSE IU overview

A GENERAL ADMIN RESPONSE IU is sent by a PQI device to a PQI host to deliver a response for an administrator request IU (see 10.1.1) from a PQI device management device server to a PQI host management application client.

Table 67 defines the GENERAL ADMIN RESPONSE IU.

**Table 67 — GENERAL ADMIN RESPONSE IU**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (E0h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE							
11	STATUS							
12	Additional status descriptor							
...								
15								
16	Function specific fields							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are part of the common IU header for all IU layers (see 9.3) and shall be set as shown in table 67 for the GENERAL ADMIN RESPONSE IU.

The WORK AREA field may be set to any value by the PQI device and should be ignored by the PQI host.

The REQUEST IDENTIFIER field indicates the request identifier of the administrator function for which this response is being returned.

The FUNCTION CODE field indicates the administrator request for which this response is being returned and is defined in 10.2.1.

Table 68 defines the STATUS field and additional status descriptor.

**Table 68 — STATUS field and additional status descriptor** (part 1 of 2)

STATUS field		Additional status descriptor
Code	Name	
Results indicating administrator function success (00h to 3Fh)		
00h	GOOD	Reserved
01h	DATA-IN BUFFER UNDERFLOW	See 10.1.5.2
02h to 3Fh	Reserved	
Results indicating administrator function failure (40h to FFh)		
Errors accessing the Data Buffer (40h to 7Fh)		
Miscellaneous errors accessing the Data Buffer (40h to 5Fh)		
40h	DATA BUFFER ERROR	Reserved
41h	DATA BUFFER OVERFLOW	Reserved
42h to 5Fh	Reserved	
PCI Express related errors accessing the Data Buffer (60h to 6Fh)		
60h	PCIE FABRIC ERROR	Reserved
61h	PCIE COMPLETION TIMEOUT	Reserved
62h	PCIE COMPLETER ABORT	Reserved
63h	PCIE POISONED TLP RECEIVED	Reserved
64h	PCIE ECRC CHECK FAILED	Reserved
65h	PCIE UNSUPPORTED REQUEST	Reserved
66h	PCIE ACS VIOLATION	Reserved
67h	PCIE TLP PREFIX BLOCKED	Reserved
68h to 6Fh	Reserved	
Other errors accessing the Data Buffer (70h to 7Fh)		
70h to 7Fh	Reserved	

**Table 68 — STATUS field and additional status descriptor (part 2 of 2)**

STATUS field		Additional status descriptor
Code	Name	
Other errors (80h to EFh)		
80h	GENERIC ERROR	Reserved
81h	OVERLAPPED REQUEST IDENTIFIER ATTEMPTED	Reserved
82h	INVALID FIELD IN REQUEST IU	See 10.1.5.3
83h	INVALID FIELD IN DATA-OUT BUFFER	See 10.1.5.4
84h to EFh	Reserved	
Vendor specific (F0h to FFh)		
F0h to FFh	Vendor-specific	

The function specific fields are defined by each administrator function (see 10.2).

#### 10.1.5.2 Additional status descriptor for STATUS field set to DATA-IN BUFFER UNDERFLOW

Table 69 defines the additional status descriptor if the STATUS field is set to DATA-IN BUFFER UNDERFLOW (see 10.1.5.1).

**Table 69 — Additional status descriptor if the STATUS field is set to DATA-IN BUFFER UNDERFLOW**

Byte\Bit	7	6	5	4	3	2	1	0	
0	DATA TRANSFERRED								(LSB)
...									
3									(MSB)

The DATA TRANSFERRED field indicates the number of contiguous bytes starting with offset zero in the Data-In Buffer that the PQI device management device server transferred.

#### 10.1.5.3 Additional status descriptor for STATUS field set to INVALID FIELD IN REQUEST IU

Table 70 defines the additional status descriptor if the STATUS field is set to INVALID FIELD IN REQUEST IU (see 10.1.5.1).

**Table 70 — Additional status descriptor if the STATUS field is set to INVALID FIELD IN REQUEST IU**

Byte\Bit	7	6	5	4	3	2	1	0
0	(LSB)							
1	(MSB)							
2	Reserved							
3	Reserved	BIT POINTER			Reserved			

The BYTE POINTER field indicates the offset in the GENERAL ADMIN REQUEST IU of the first byte (i.e., the lowest byte number) containing the field with the invalid value.

The BIT POINTER field indicates the offset in the byte in the GENERAL ADMIN REQUEST IU pointed to by the BYTE POINTER field of the first bit (i.e., the lowest bit number) containing the field with the invalid value.

#### 10.1.5.4 Additional status descriptor for STATUS field set to INVALID FIELD IN DATA-OUT BUFFER

Table 71 defines the additional status descriptor if the STATUS field is set to INVALID FIELD IN DATA-OUT BUFFER (see 10.1.5.1).

**Table 71 — Additional status descriptor if the STATUS field is set to INVALID FIELD IN DATA-OUT BUFFER**

Byte\Bit	7	6	5	4	3	2	1	0
0	<div style="text-align: right;">(LSB)</div> <div style="text-align: center;">BYTE POINTER</div> <div style="text-align: left;">(MSB)</div>							
...								
2								
3	Reserved	BIT POINTER			Reserved			

The BYTE POINTER field indicates the offset in the Data-Out Buffer of the first byte (i.e., the lowest byte number) containing the field with the invalid value. If the byte that was in error is at an offset greater than FF\_FFFFh, then the BYTE POINTER field shall be set to FF\_FFFFh and the BIT POINTER field shall be set to 111b.

The BIT POINTER field indicates the offset in the byte of the Data-Out Buffer pointed to by the BYTE POINTER field of the first bit (i.e., the lowest bit number) containing the field with the invalid value.

## 10.2 Administrator functions

### 10.2.1 Administrator functions overview

Table 72 defines the administrator functions.

**Table 72 — Administrator functions (FUNCTION CODE field)**

Code	Function	Description	Type	Support	Reference
General functions (00h to 0Fh)					
00h	REPORT PQI DEVICE CAPABILITY	Report PQI device capabilities	R	M	10.2.2
01h	REPORT MANUFACTURER INFORMATION	Report PQI device manufacturing information	R	M	10.2.3
02h to 0Fh	Reserved				
Operational queue functions (10h to 1Fh)					
10h	CREATE OPERATIONAL IQ	Create an operational IQ	N	M	10.2.4
11h	CREATE OPERATIONAL OQ	Create an operational OQ	N	M	10.2.5
12h	DELETE OPERATIONAL IQ	Delete an operational IQ	N	M	10.2.6
13h	DELETE OPERATIONAL OQ	Delete an operational OQ	N	M	10.2.7
14h	CHANGE OPERATIONAL IQ PROPERTIES	Change properties of an operational IQ	N	O	10.2.8
15h	CHANGE OPERATIONAL OQ PROPERTIES	Change properties of an operational OQ	N	O	10.2.9
16h	REPORT OPERATIONAL IQ LIST	Report list of configured operational IQs	R	M	10.2.10
17h	REPORT OPERATIONAL OQ LIST	Report list of configured operational OQs	R	M	10.2.11
18h to 1Fh	Reserved				
Other (20h to DFh)					
20h to DFh	Reserved				
Vendor specific (E0h to FFh)					
E0h to FFh	Vendor specific				
<b>Key:</b> M = Function implementation by the PQI device is mandatory O = Function implementation by the PQI device is optional R = Read administrator function W = Write administrator function N = Non-data administrator function					

## 10.2.2 REPORT PQI DEVICE CAPABILITY function

### 10.2.2.1 REPORT PQI DEVICE CAPABILITY request

The REPORT PQI DEVICE CAPABILITY function requests that the PQI device management device server return information about PQI device capabilities in the Data-In Buffer as defined in 10.2.2.3.

Table 73 defines the REPORT PQI DEVICE CAPABILITY request.

**Table 73 — REPORT PQI DEVICE CAPABILITY request**

Byte\Bit	7	6	5	4	3	2	1	0	
0	IU TYPE (60h)								
1	PQI COMPATIBLE FEATURES (0h)				Reserved				
2	IU LENGTH (003Ch) (LSB)								
3									(MSB)
4	Reserved								
5									
6	WORK AREA								
7									
8	REQUEST IDENTIFIER (LSB)								
9									(MSB)
10	FUNCTION CODE (00h)								
11	RsvdC								
...									
43									
44	DATA-IN BUFFER SIZE (LSB)								
...									
47	(MSB)								
48	SGL descriptor (see table 53)								
...									
63									

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.4 and shall be set to as shown in table 73 for the REPORT PQI DEVICE CAPABILITY request.

The WORK AREA field may be set to any value by the PQI host and shall be ignored by the PQI device.

The REQUEST IDENTIFIER field is defined in 10.1.4.

The FUNCTION CODE field is defined in 10.2.1 and is set to the value shown in table 73 for the REPORT PQI DEVICE CAPABILITY request.

The DATA-IN BUFFER SIZE field is defined in 10.1.4.

The SGL descriptor is defined in 10.1.4.

### 10.2.2.2 REPORT PQI DEVICE CAPABILITY response

Table 74 defines the REPORT PQI DEVICE CAPABILITY response.

**Table 74 — REPORT PQI DEVICE CAPABILITY response**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (E0h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE (00h)							
11	STATUS							
12	Additional status descriptor							
...								
15								
16	Reserved							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.5 and shall be set as shown in table 74 for the REPORT PQI DEVICE CAPABILITY response.

The WORK AREA field may be set to any value by the PQI device and should be ignored by the PQI host.

The REQUEST IDENTIFIER field is defined in 10.1.5.1.

The FUNCTION CODE field is defined in 10.2.1 and is set to the value shown in table 74 for the REPORT PQI DEVICE CAPABILITY response.

The STATUS field and the additional status descriptor are defined in 10.1.5.



## 10.2.2.3 REPORT PQI DEVICE CAPABILITY parameter data

The format of the parameter data returned in the Data-In Buffer is shown in table 75.

**Table 75 — REPORT PQI DEVICE CAPABILITY parameter data (i.e., Data-In Buffer contents)** (part 1 of 2)

Byte\Bit	7	6	5	4	3	2	1	0
0	PARAMETER DATA LENGTH (023Eh)							(LSB)
1								(MSB)
2	Reserved							
...								
15								
Operational IQ capabilities								
16	MAXIMUM OPERATIONAL IQS							(LSB)
17								(MSB)
18	MAXIMUM OPERATIONAL IQ ELEMENTS							(LSB)
19								(MSB)
20	Reserved							
...								
23								
24								MAXIMUM OPERATIONAL IQ ELEMENT LENGTH
25	(MSB)							
26	MINIMUM OPERATIONAL IQ ELEMENT LENGTH							(LSB)
27								(MSB)
Operational OQ capabilities								
28	Reserved							CIC
29	Reserved							
30	MAXIMUM OPERATIONAL OQS							(LSB)
31								(MSB)
32	MAXIMUM OPERATIONAL OQ ELEMENTS							(LSB)
33								(MSB)
34	INTERRUPT COALESCING TIME GRANULARITY							(LSB)
35								(MSB)

**Table 75 — REPORT PQI DEVICE CAPABILITY parameter data (i.e., Data-In Buffer contents) (part 2 of 2)**

Byte\Bit	7	6	5	4	3	2	1	0
36	MAXIMUM OPERATIONAL OQ ELEMENT LENGTH							(LSB)
37								(MSB)
38	MINIMUM OPERATIONAL OQ ELEMENT LENGTH							(LSB)
39								(MSB)
Other parameters								
40	Reserved							
...								
43								
44	OPERATIONAL QUEUE PROTOCOL SUPPORT BITMASK							
...								
47								
48	ADMINISTRATOR SGL DESCRIPTOR TYPE SUPPORT BITMASK							
49								
50								
...	Reserved							
63								
IU layer specific descriptor list								
64	IU layer specific descriptor [first] (see table 76)							
...								
79	...							
...								
560	IU layer specific descriptor [last] (see table 76)							
...								
575								

The PARAMETER DATA LENGTH field indicates the number of bytes that follow and shall be set as shown in table 75 for the REPORT PQI DEVICE CAPABILITY parameter data.

The MAXIMUM OPERATIONAL IQS field indicates the maximum number of operational IQs supported by the PQI device.

The MAXIMUM OPERATIONAL IQ ELEMENTS field indicates the maximum number of elements in each of the operational IQs supported by the PQI device. The MAXIMUM OPERATIONAL IQ ELEMENTS field shall be set to a value greater than 0001h.

The MAXIMUM OPERATIONAL IQ ELEMENT LENGTH field indicates the maximum length of each operational IQ element in 16-byte increments (e.g., 0001h means 16 bytes and 00FFh means 4 080 bytes). The MAXIMUM OPERATIONAL IQ ELEMENT LENGTH field shall be greater than or equal to the MINIMUM OPERATIONAL OQ ELEMENT LENGTH field.

The MINIMUM OPERATIONAL IQ ELEMENT LENGTH field indicates the minimum length of each operational IQ element in 16-byte increments (e.g., 0001h means 16 bytes and 00FFh means 4 080 bytes) supported by the PQI device. The MINIMUM OPERATIONAL IQ ELEMENT LENGTH field shall not be set to 0000h.

A common interrupt coalescing (CIC) bit set to one indicates that the PQI device manages the values of the Coalescing Count attribute (see 5.2.5.12.10), the Minimum Coalescing Time attribute (see 5.2.5.12.8), the Maximum Coalescing Time attribute (see 5.2.5.12.9), and the Wait For Rearm attribute (see 5.2.5.12.11), such that:

- a) the Coalescing Count attribute is the same value for all operational OQs;
- b) the Minimum Coalescing Time attribute is the same value for all operational OQs;
- c) the Maximum Coalescing Time attribute is the same value for all operational OQs; and
- d) the Wait For Rearm attribute is the same value for all operational OQs.

A CIC bit set to zero indicates:

- a) the Coalescing Count attribute may be a different value for each operational OQ;
- b) the Minimum Coalescing Time attribute may be a different value for each operational OQ;
- c) the Maximum Coalescing Time attribute may be a different value for each operational OQ; and
- d) the Wait For Rearm attribute may be a different value for each operational OQ.

The MAXIMUM OPERATIONAL OQS field indicates the maximum number of operational OQs supported by the PQI device.

The MAXIMUM OPERATIONAL OQ ELEMENTS field indicates the maximum number of elements in each of the operational OQs supported by the PQI device. The MAXIMUM OPERATIONAL OQ ELEMENTS field shall be set to a value greater than 0001h.

An INTERRUPT COALESCING TIME GRANULARITY field indicates the granularity supported for the coalescing timers (see 10.2.5.1) in 100 ns intervals. An INTERRUPT COALESCING TIME GRANULARITY field shall not be set to 0000h.

The MAXIMUM OPERATIONAL OQ ELEMENT LENGTH field indicates the maximum length of each operational OQ element in 16-byte increments (e.g., 0001h means 16 bytes and 00FFh means 4 080 bytes) supported by the PQI device. The MAXIMUM OPERATIONAL OQ ELEMENT LENGTH field shall be greater than or equal to the MINIMUM OPERATIONAL OQ ELEMENT LENGTH field.

The MINIMUM OPERATIONAL OQ ELEMENT LENGTH field indicates the minimum length of each operational OQ element in 16-byte increments (e.g., 0001h means 16 bytes and 00FFh means 4 080 bytes) supported by the PQI device. The MINIMUM OPERATIONAL OQ ELEMENT LENGTH field shall not be set to 0000h.

The OPERATIONAL QUEUE PROTOCOL SUPPORT BITMASK field indicates the operational queue protocols (see table 81) that are supported by the PQI device. A bit set to one indicates that the corresponding operational queue protocol is supported. A bit set to zero indicates that the corresponding operational queue protocol is not supported. The first bit (i.e., byte 44 bit 0) corresponds to an operational queue protocol of 00h (i.e., SOP); the last bit (i.e., byte 47 bit 7) corresponds to an operational queue protocol of 1Fh (i.e., vendor specific).

The ADMINISTRATOR SGL DESCRIPTOR TYPE SUPPORT BITMASK field indicates the SGL descriptor types (see table 54) that are supported by the PQI device for administrator request IUs. A bit set to one indicates that the corresponding SGL descriptor type is supported. A bit set to zero indicates that the corresponding SGL descriptor type is not supported. The first bit (i.e., byte 48 bit 0) corresponds to an SGL descriptor type of 0h (i.e., the Data Block descriptor); the last bit (i.e., byte 49 bit 7) corresponds to an SGL descriptor type of Fh. Bits 3 to 0 represent the mandatory SGL descriptor types for administrator IUs (see table 65) and shall be set to 1111b.

The SGL descriptor type support bitmask for other IU layers (e.g., SOP) is not defined in this standard.

The IU layer specific descriptor (see table 76) specifies the additional features supported by the operational queue protocol. The first IU layer specific descriptor corresponds to an operational queue protocol of 00h (i.e., SOP) and the last (i.e., the 32nd) IU layer specific descriptor corresponds to an operational queue protocol of 1Fh (i.e., vendor specific).

Table 76 — IU layer specific descriptor

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							INBOUND SPANNING
1	Reserved							
...								
5								
6	MAXIMUM INBOUND IU LENGTH							(LSB)
7								(MSB)
8	Reserved							OUTBOUND SPANNING
9	Reserved							
...								
13								
14	MAXIMUM OUTBOUND IU LENGTH							(LSB)
15								(MSB)

An INBOUND SPANNING bit set to one indicates that the IU layer (e.g., SOP) supports spanning an inbound IU across multiple elements. An INBOUND SPANNING bit set to zero indicates that the IU layer (e.g., SOP) does not support spanning an inbound IU across multiple elements.

The MAXIMUM INBOUND IU LENGTH field indicates the maximum number of bytes in an inbound IU supported by the IU layer.

An OUTBOUND SPANNING bit set to one indicates that the IU layer (e.g., SOP) supports spanning an outbound IU across multiple elements. An OUTBOUND SPANNING bit set to zero indicates that the IU layer (e.g., SOP) does not support spanning an outbound IU across multiple elements.

The MAXIMUM OUTBOUND IU LENGTH field indicates the maximum number of bytes in an outbound IU supported by the IU layer.

For each of the IU layer specific descriptors, if the PQI device does not support the operational queue protocol, the PQI device shall set:

- the INBOUND SPANNING bit to zero;
- the MAXIMUM INBOUND IU LENGTH field to zero;
- the OUTBOUND SPANNING bit to zero; and
- the MAXIMUM OUTBOUND IU LENGTH field to zero.

### 10.2.3 REPORT MANUFACTURER INFORMATION function

#### 10.2.3.1 REPORT MANUFACTURER INFORMATION request

The REPORT MANUFACTURER INFORMATION function requests that the PQI device management device server return information about the PQI device manufacturer in the Data-In Buffer as defined in 10.2.3.3.

Table 77 defines the REPORT MANUFACTURER INFORMATION request.

**Table 77 — REPORT MANUFACTURER INFORMATION request**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (60h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE (01h)							
11	RsvdC							
...								
43	DATA-IN BUFFER SIZE (LSB)							
44								
...	DATA-IN BUFFER SIZE (MSB)							
47								
48	SGL descriptor (see table 53)							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.4 and shall be set as shown in table 77 for the REPORT MANUFACTURER INFORMATION request.

The WORK AREA field may be set to any value by the PQI host and shall be ignored by the PQI device.

The REQUEST IDENTIFIER field is defined in 10.1.4.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 77 for the REPORT MANUFACTURER INFORMATION request.

The DATA-IN BUFFER SIZE field is defined in 10.1.4.

The SGL descriptor is defined in 10.1.4.

### 10.2.3.2 REPORT MANUFACTURER INFORMATION response

Table 78 defines the REPORT MANUFACTURER INFORMATION response.

**Table 78 — REPORT MANUFACTURER INFORMATION response**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (E0h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE (01h)							
11	STATUS							
12	Additional status descriptor							
...								
15								
16	Reserved							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.5 and shall be set as shown in table 78 for the REPORT MANUFACTURER INFORMATION response.

The WORK AREA field may be set to any value by the PQI device and should be ignored by the PQI host.

The REQUEST IDENTIFIER field is defined in 10.1.5.1.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 78 for the REPORT MANUFACTURER INFORMATION response.

The STATUS field and the additional status descriptor are defined in 10.1.5.

## 10.2.3.3 REPORT MANUFACTURER INFORMATION parameter data

Table 79 defines the REPORT MANUFACTURER INFORMATION parameter data.

**Table 79 — REPORT MANUFACTURER INFORMATION parameter data (i.e., Data-In Buffer contents)**

Byte\Bit	7	6	5	4	3	2	1	0
0	PARAMETER DATA LENGTH (007Eh)							(LSB)
1								(MSB)
2	Reserved							
3								
4	PCI VENDOR ID							(LSB)
5								(MSB)
6	PCI DEVICE ID							(LSB)
7								(MSB)
8	PCI REVISION ID							
9								(LSB)
...	PCI CLASS CODE							
11								(MSB)
12	PCI SUBSYSTEM VENDOR ID							(LSB)
13								(MSB)
14	PCI SUBSYSTEM ID							(LSB)
15								(MSB)
16	PRODUCT SERIAL NUMBER							
...								(LSB)
47	T10 VENDOR IDENTIFICATION							
48								(MSB)
...	PRODUCT IDENTIFICATION							(LSB)
55								
56	PRODUCT REVISION LEVEL							(LSB)
...								
71	Reserved							
72								(MSB)
...								
87								(LSB)
88								
...								
127								

The PARAMETER DATA LENGTH field indicates the number of bytes that follow and shall be set as shown in table 79.

The PCI VENDOR ID field indicates the identification of the manufacturer of the PCI device and shall be identical to the Vendor ID field in configuration space (see PCI).

The PCI DEVICE ID field indicates the identification of the PCI device allocated by the PCI device manufacturer and shall be identical to the Device ID field in configuration space (see PCI).

The PCI REVISION ID field indicates the revision of the PCI device allocated by the PCI device manufacturer and shall be identical to the Revision ID field in configuration space (see PCI).

The PCI CLASS CODE field indicates the identification of the generic function and register level programming interface of the PCI device defined by PCI-ID and shall be identical to the Class Code field in configuration space (see PCI).

The PCI SUBSYSTEM VENDOR ID field indicates the manufacturer of the add-in card or subsystem containing the PCI device and shall be identical to the Subsystem Vendor ID field in configuration space (see PCI).

The PCI SUBSYSTEM ID field indicates the identification of the add-in card or subsystem identification allocated by the subsystem manufacturer used to identify the add-in card or subsystem where the PCI device resides and shall be identical to the Subsystem ID field in configuration space (see PCI).

The PRODUCT SERIAL NUMBER field indicates ASCII data (see 4.1) that is a manufacturer defined serial number. If the product serial number is not available, then the PRODUCT SERIAL NUMBER field shall contain ASCII spaces (20h).

The T10 VENDOR IDENTIFICATION field indicates eight bytes of left-aligned ASCII data (see 4.1) identifying the manufacturer of the PQI device. The T10 vendor identification shall be one assigned by INCITS. A list of assigned T10 vendor identifications is defined in SPC-4 and on the T10 web site (<http://www.t10.org>).

The PRODUCT IDENTIFICATION field indicates sixteen bytes of left-aligned ASCII product identification data (see 4.1) defined by the manufacturer.

The PRODUCT REVISION LEVEL field indicates sixteen bytes of left-aligned ASCII product revision level data (see 4.1) defined by the manufacturer.

## 10.2.4 CREATE OPERATIONAL IQ function

### 10.2.4.1 CREATE OPERATIONAL IQ request

The CREATE OPERATIONAL IQ function requests that the PQI device management device server create a new operational IQ.

Table 80 defines the CREATE OPERATIONAL IQ request.

**Table 80 — CREATE OPERATIONAL IQ request (part 1 of 2)**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (60h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
2	Reserved							
3								
6	WORK AREA							
7								



Table 80 — CREATE OPERATIONAL IQ request (part 2 of 2)

Byte\Bit	7	6	5	4	3	2	1	0						
8	REQUEST IDENTIFIER (LSB)													
9									(MSB)					
10	FUNCTION CODE (10h)													
11	RsvdC													
12	IQ ID (LSB)													
13									(MSB)					
14	RsvdC													
15														
16	(LSB)		RsvdC											
...														
23	IQ ELEMENT ARRAY ADDRESS (MSB)													
24														
...	IQ CI ADDRESS (LSB)						RsvdC							
31									(MSB)					
32	NUMBER OF ELEMENTS (LSB)													
33									(MSB)					
34	ELEMENT LENGTH (LSB)													
35									(MSB)					
36	Reserved			OPERATIONAL QUEUE PROTOCOL										
37	RsvdC													
...														
59														
60									Vendor specific					
...														
63														

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.4 and shall be set as shown in table 80 for the CREATE OPERATIONAL IQ request.

The WORK AREA field may be set to any value by the PQI host and shall be ignored by the PQI device.

The REQUEST IDENTIFIER field is defined in 10.1.4.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 80 for the CREATE OPERATIONAL IQ request.

The IQ ID field specifies the IQ ID (see 5.2.5.9.2) to be assigned to the operational IQ. If:

- the IQ ID field is set to 0000h;
- the IQ ID field is set to a value that is already assigned to an operational IQ; or

- c) the IQ ID field is set to a value greater than the MAXIMUM OPERATIONAL IQS field in the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3),

then the PQI device management device server shall return a CREATE OPERATIONAL IQ response with the STATUS field set to INVALID FIELD IN REQUEST IU (see 10.1.5.1 and 10.1.5.3).

The IQ ELEMENT ARRAY ADDRESS field specifies the upper 58 bits of the 64-bit operational IQ element array address (see 5.3.2.1). The least significant six bits of the 64-bit operational IQ element array address, which are not specified by the IQ ELEMENT ARRAY ADDRESS field, are zero.

The IQ CI ADDRESS field specifies the upper 62 bits of the 64-bit operational IQ CI address (see 5.2.5.9.7). The least significant two bits of the 64-bit operational IQ CI address, which are not specified by the IQ CI ADDRESS field, are zero.

The NUMBER OF ELEMENTS field specifies the number of elements in the operational IQ element array (see 5.2.5.9.5). If:

- a) the NUMBER OF ELEMENTS field is set to a value less than 0002h; or
- b) the NUMBER OF ELEMENTS field is set to a value greater than the MAXIMUM OPERATIONAL IQ ELEMENTS field in the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3),

then the PQI device management device server shall return a CREATE OPERATIONAL IQ response with the STATUS field set to INVALID FIELD IN REQUEST IU (see 10.1.5.1 and 10.1.5.3).

The ELEMENT LENGTH field specifies the element length in 16-byte increments (e.g., a value of 0001h in the ELEMENT LENGTH field specifies an element length of 16 bytes). If the ELEMENT LENGTH field is set to a value:

- a) less than the MINIMUM OPERATIONAL IQ ELEMENT LENGTH field in the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3); or
- b) greater than the MAXIMUM OPERATIONAL IQ ELEMENT LENGTH field in the REPORT PQI DEVICE CAPABILITY parameter data,

then the PQI device management device server shall return a CREATE OPERATIONAL IQ response with the STATUS field set to INVALID FIELD IN REQUEST IU (see 10.1.5.1 and 10.1.5.3).

The OPERATIONAL QUEUE PROTOCOL (see table 81) specifies the operational queue protocol (i.e., IU layer) used by the operational IQ (see 5.3.1).

**Table 81 — OPERATIONAL QUEUE PROTOCOL field**

Code	Operational queue protocol (i.e., IU layer)	Reference
00h	SOP	SOP
01h to 0Fh	Reserved	
10h to 1Fh	Vendor specific	

## 10.2.4.2 CREATE OPERATIONAL IQ response

Table 82 defines the CREATE OPERATIONAL IQ response.

Table 82 — CREATE OPERATIONAL IQ response

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (E0h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE (10h)							
11	STATUS							
12	Additional status descriptor							
...								
15								
16	IQ PI OFFSET (LSB)							
...								
23	(MSB)							
24	Reserved							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.5 and shall be set as shown in table 82 for the CREATE OPERATIONAL IQ response.

The WORK AREA field may be set to any value by the PQI device and should be ignored by the PQI host.

The REQUEST IDENTIFIER field is defined in 10.1.5.1.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 82 for the CREATE OPERATIONAL IQ response.

The STATUS field and the additional status descriptor are defined in 10.1.5.

If the STATUS field is set to GOOD (see 10.1.5.1), then:

- a) the IQ PI OFFSET field indicates the offset in PQI device memory space of the operational IQ PI (i.e., the IQ PI address is the memory address contained in the first PCI memory BAR plus the IQ PI OFFSET field); and
- b) the IQ PI OFFSET field shall be a multiple of four (i.e, byte 0 bit 0 set to zero and byte 0 bit 1 set to zero).

If the STATUS field is not set to GOOD, then the IQ PI OFFSET field is invalid.

## 10.2.5 CREATE OPERATIONAL OQ function

### 10.2.5.1 CREATE OPERATIONAL OQ request

The CREATE OPERATIONAL OQ function requests that the PQI device management device server create a new operational OQ.

Table 83 defines the CREATE OPERATIONAL OQ request.

**Table 83 — CREATE OPERATIONAL OQ request** (part 1 of 2)

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (60h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE (11h)							
11	RsvdC							
12	OQ ID (LSB)							
13								
14	RsvdC							
15								
16	(LSB)		RsvdC					
...								
23	OQ ELEMENT ARRAY ADDRESS (MSB)							
24	OQ PI ADDRESS (LSB)						RsvdC	
...								
31	(MSB)							
32	NUMBER OF ELEMENTS (LSB)							
33								

Table 83 — CREATE OPERATIONAL OQ request (part 2 of 2)

Byte\Bit	7	6	5	4	3	2	1	0
34	(LSB)							
35	ELEMENT LENGTH							
36	Reserved			OPERATIONAL QUEUE PROTOCOL				
37	RsvdC							
...								
39								
40	(LSB)							
41	WAIT FOR REARM	RsvdC				(MSB)		
42	(LSB)							
43	COALESCING COUNT							
44	(LSB)							
...	MINIMUM COALESCING TIME							
47								
48	(LSB)							
...	MAXIMUM COALESCING TIME							
51								
52	RsvdC							
...								
59								
60	Vendor specific							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.4 and shall be set as shown in table 83 for the CREATE OPERATIONAL OQ request.

The WORK AREA field may be set to any value by the PQI host and shall be ignored by the PQI device.

The REQUEST IDENTIFIER field is defined in 10.1.4.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 83 for the CREATE OPERATIONAL OQ request.

The OQ ID field specifies the OQ ID to be assigned to the operational OQ. If:

- the OQ ID field is set to 0000h;
- the OQ ID field is set to a value that is already assigned to an operational OQ; or
- the OQ ID field is set to a value greater than the MAXIMUM OPERATIONAL OQS field in the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3),

then the PQI device management device server shall return a CREATE OPERATIONAL OQ response with the STATUS field set to INVALID FIELD IN REQUEST IU (see 10.1.5.3).

The OQ ELEMENT ARRAY ADDRESS field specifies the upper 58 bits of the 64-bit operational OQ element array address (see 5.3.2.1). The least significant six bits of the 64-bit operational OQ element array address, which are not specified by the OQ ELEMENT ARRAY ADDRESS field, are zero.

The OQ PI ADDRESS field specifies the upper 62 bits of the 64-bit operational OQ PI address (see 5.3.3). The least significant two bits of the 64-bit operational OQ CI address, which are not specified by the OQ PI ADDRESS field, are zero.

The NUMBER OF ELEMENTS field specifies the number of elements in the operational OQ element array (see 5.3.3). If:

- a) the NUMBER OF ELEMENTS field is set to a value less than 0002h; or
- b) the NUMBER OF ELEMENTS field is set to a value greater than the MAXIMUM OPERATIONAL OQ ELEMENTS field in the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3),

then the PQI device management device server shall return a CREATE OPERATIONAL OQ response with the STATUS field set to INVALID FIELD IN REQUEST IU (see 10.1.5.3).

The ELEMENT LENGTH field specifies the element length in 16-byte increments (e.g., a value of 0001h in the ELEMENT LENGTH field specifies an element length of 16 bytes). If the ELEMENT LENGTH field is set to a value:

- a) less than the MINIMUM OPERATIONAL OQ ELEMENT LENGTH field in the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3); or
- b) greater than the MAXIMUM OPERATIONAL OQ ELEMENT LENGTH field in the REPORT PQI DEVICE CAPABILITY parameter data,

then the PQI device management device server shall return a CREATE OPERATIONAL OQ response with the STATUS field set to INVALID FIELD IN REQUEST IU (see 10.1.5.3).

The OPERATIONAL QUEUE PROTOCOL field (see table 81) specifies the operational queue protocol used by the operational OQ (see 5.3.1).

The INTERRUPT MESSAGE NUMBER field specifies the MSI-X Table entry used to generate the interrupt message for operational OQ PI updates in MSI-X mode (see 5.4.2). This field is not used in legacy INTx mode (see 5.4.3) or polled mode (see 5.4.4). If the INTERRUPT MESSAGE NUMBER field is set to value greater than the Table Size field in the Message Control register in the MSI-X Capability Structure (see PCI), then the PQI device management device server shall return a CREATE OPERATIONAL OQ response with the STATUS field set to INVALID FIELD IN REQUEST IU (see 10.1.5.3).

NOTE 10 - The Table Size field is 11 bits, therefore the maximum interrupt message number that the PQI device is capable of supporting is less than or equal to 2 047.

The WAIT FOR REARM bit, the COALESCING COUNT field, the MINIMUM COALESCING TIME field, and the MAXIMUM COALESCING TIME field are used for controlling interrupt coalescing in MSI-X mode (see 5.4.2).

If:

- a) the CIC bit is set to one in the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3);
- b) one or more operational OQs already exist; and
- c) one or more of the following conditions is true:
  - A) the COALESCING COUNT field is set to a value that is not already in use for the Coalescing Count attribute by the existing operational OQs;
  - B) the MINIMUM COALESCING TIME field is set to a value that is not already in use for the Minimum Coalescing Time attribute by the existing operational OQs;
  - C) the MAXIMUM COALESCING TIME field is set to a value that is not already in use for the Maximum Coalescing Time attribute by the existing operational OQs; or
  - D) the WAIT FOR REARM bit is set to a value that is not equal to the value of the Wait For Rearm attribute for the existing operational OQs,

then the PQI device management device server shall return a CREATE OPERATIONAL OQ response with the STATUS field set to INVALID FIELD IN REQUEST IU (see 10.1.5.3).

A WAIT FOR REARM bit set to one specifies that the interrupt coalescing timer is reset and started upon receipt of the REARM INTERRUPT bit set to one (see 5.4.2). A WAIT FOR REARM bit set to zero specifies that the interrupt coalescing timer is reset and started upon the sending of a prior interrupt. The WAIT FOR REARM bit may be changed using the CHANGE OPERATIONAL OQ PROPERTIES function (see 10.2.9).

The COALESCING COUNT field specifies a number of occupied operational OQ element array entries used for interrupt coalescing (see 5.4.2 and 5.2.5.12.10). The COALESCING COUNT field may be changed using the CHANGE OPERATIONAL OQ PROPERTIES function (see 10.2.9).

The MINIMUM COALESCING TIME field specifies a minimum coalescing time in 100 ns intervals (see 5.4.2 and 5.2.5.12.8). If the MINIMUM COALESCING TIME field is greater than the MAXIMUM COALESCING TIME field, then the minimum coalescing time shall be set to zero. If the MINIMUM COALESCING TIME field is set to a value other than zero that is not a multiple of the INTERRUPT COALESCING TIME GRANULARITY field in the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3), then the minimum coalescing time in the PQI device shall be rounded up to the next multiple of the INTERRUPT COALESCING TIME GRANULARITY field (e.g., if the MINIMUM COALESCING TIME field is set to 21 and the INTERRUPT COALESCING TIME GRANULARITY field is set to 10, then the minimum coalescing time shall be rounded up to 30). The MINIMUM COALESCING TIME field may be changed using the CHANGE OPERATIONAL OQ PROPERTIES function (see 10.2.9).

The MAXIMUM COALESCING TIME field specifies a maximum coalescing time in 100 ns intervals (see 5.4.2 and 5.2.5.12.9). If the MAXIMUM COALESCING TIME field is set to a value other than zero that is not a multiple of the INTERRUPT COALESCING TIME GRANULARITY field in the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3), then the maximum coalescing time in the PQI device shall be rounded up to the next multiple of the INTERRUPT COALESCING TIME GRANULARITY field (e.g., if the MAXIMUM COALESCING TIME field is set to 53 and the INTERRUPT COALESCING TIME GRANULARITY field is set to 10, then the maximum coalescing time shall be rounded up to 60). The MAXIMUM COALESCING TIME field may be changed using the CHANGE OPERATIONAL OQ PROPERTIES function (see 10.2.9).



## 10.2.5.2 CREATE OPERATIONAL OQ response

Table 84 defines the CREATE OPERATIONAL OQ response.

Table 84 — CREATE OPERATIONAL OQ response

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (E0h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE (11h)							
11	STATUS							
12	Additional status descriptor							
...								
15								
16	OQ CI OFFSET (LSB)							
...								
23	(MSB)							
24	Reserved							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.5 and shall be set as shown in table 84 for the CREATE OPERATIONAL OQ response.

The WORK AREA field may be set to any value by the PQI device and should be ignored by the PQI host.

The REQUEST IDENTIFIER field is defined in 10.1.5.1.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 84 for the CREATE OPERATIONAL OQ response.

The STATUS field and the additional status descriptor are defined in 10.1.5.

If the STATUS field is set to GOOD (see 10.1.5.1), then:

- a) the OQ CI OFFSET field indicates the offset in PQI device memory space of the operational OQ CI (i.e., the operational OQ CI address is the memory address contained by the first PCI memory BAR plus the OQ CI OFFSET field); and
- b) byte 0 bits 1 to 0 of the OQ CI OFFSET field shall be set to 00b.

If the STATUS field is not set to GOOD, then the OQ CI OFFSET field is invalid.

## 10.2.6 DELETE OPERATIONAL IQ function

### 10.2.6.1 DELETE OPERATIONAL IQ request

The DELETE OPERATIONAL IQ function requests that the PQI device management device server delete the specified operational IQ.

See 5.3.4.3 for information on when it is safe to delete an operational IQ.

Table 85 defines the DELETE OPERATIONAL IQ request.

**Table 85 — DELETE OPERATIONAL IQ request**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (60h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE (12h)							
11	RsvdC							
12	IQ ID (LSB)							
13								
14	RsvdC							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.4 and shall be set as shown in table 85 for the DELETE OPERATIONAL IQ requests.

The WORK AREA field may be set to any value by the PQI host and shall be ignored by the PQI device.

The REQUEST IDENTIFIER field is defined in 10.1.4.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 85 for the DELETE OPERATIONAL IQ requests.

The IQ ID field specifies the IQ ID (see 5.2.5.9.2) of the operational IQ to be deleted. If the IQ ID field is set to an IQ ID that does not exist, then the PQI device management device server shall return a DELETE OPERATIONAL IQ response with the STATUS field set to INVALID FIELD IN REQUEST IU (see 10.1.5.3).

#### 10.2.6.2 DELETE OPERATIONAL IQ response

Table 86 defines the DELETE OPERATIONAL IQ response.

**Table 86 — DELETE OPERATIONAL IQ response**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (E0h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	(LSB) IU LENGTH (003Ch)							
3								
4	Reserved							
5								
6								
7	WORK AREA							
8	(LSB) REQUEST IDENTIFIER							
9								
10	FUNCTION CODE (12h)							
11	STATUS							
12	Additional status descriptor							
...								
15								
16	Reserved							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.5 and shall be set as shown in table 86 for the DELETE OPERATIONAL IQ response.

The WORK AREA field may be set to any value by the PQI device and should be ignored by the PQI host.

The REQUEST IDENTIFIER field is defined in 10.1.5.1.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 86 for the DELETE OPERATIONAL IQ response.

The STATUS field and the additional status descriptor are defined in 10.1.5.

## 10.2.7 DELETE OPERATIONAL OQ function

### 10.2.7.1 DELETE OPERATIONAL OQ request

The DELETE OPERATIONAL OQ function requests that the PQI device management device server delete the specified operational OQ.

See 5.3.4.3 for information on when it is safe to delete an operational OQ.

Table 87 defines the DELETE OPERATIONAL OQ request.

**Table 87 — DELETE OPERATIONAL OQ request**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (60h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch)							(LSB)
3								(MSB)
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER							(LSB)
9								(MSB)
10	FUNCTION CODE (13h)							
11	RsvdC							
12	OQ ID							(LSB)
13								(MSB)
14	RsvdC							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.4 and shall be set as shown in table 87 for the DELETE OPERATIONAL OQ request.

The WORK AREA field may be set to any value by the PQI host and shall be ignored by the PQI device.

The REQUEST IDENTIFIER field is defined in 10.1.4.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 87 for the DELETE OPERATIONAL OQ request.

The OQ ID field specifies the OQ ID (see 5.2.5.12.2) of the operational OQ to be deleted. If the OQ ID field is set to an OQ ID that does not exist, then the PQI device management device server shall return a DELETE OPERATIONAL OQ response with the STATUS field set to INVALID FIELD IN REQUEST IU (see 10.1.5.3).

### 10.2.7.2 DELETE OPERATIONAL OQ response

Table 88 defines the DELETE OPERATIONAL OQ response.

**Table 88 — DELETE OPERATIONAL OQ response**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (E0h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE (13h)							
11	STATUS							
12	Additional status descriptor							
...								
15								
16	Reserved							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.5 and shall be set as shown in table 88 for the DELETE OPERATIONAL OQ response.

The WORK AREA field may be set to any value by the PQI device and should be ignored by the PQI host.

The REQUEST IDENTIFIER field is defined in 10.1.5.1.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 88 for the DELETE OPERATIONAL OQ response.

The STATUS field and the additional status descriptor are defined in 10.1.5.

**10.2.8 CHANGE OPERATIONAL IQ PROPERTIES function****10.2.8.1 CHANGE OPERATIONAL IQ PROPERTIES request**

The CHANGE OPERATIONAL IQ PROPERTIES function requests that the PQI device management device server change the properties of the specified operational IQ.

Table 89 defines the CHANGE OPERATIONAL IQ PROPERTIES request.

**Table 89 — CHANGE OPERATIONAL IQ PROPERTIES request**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (60h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE (14h)							
11	RsvdC							
12	IQ ID (LSB)							
13								
14	RsvdC							
...								
59	Vendor specific							
60								
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.4 and shall be set as shown in table 89 for the CHANGE OPERATIONAL IQ PROPERTIES request.

The WORK AREA field may be set to any value by the PQI host and shall be ignored by the PQI device.

The REQUEST IDENTIFIER field is defined in 10.1.4.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 89 for the CHANGE OPERATIONAL IQ PROPERTIES request.

The IQ ID field specifies the IQ ID (see 5.2.5.9.2) of the operational IQ to be configured. If the IQ ID field is set to an IQ ID that does not exist, then the PQI device management device server shall return a CHANGE OPERATIONAL IQ PROPERTIES response with the STATUS field set to INVALID FIELD IN REQUEST IU (see 10.1.5.3).

### 10.2.8.2 CHANGE OPERATIONAL IQ PROPERTIES response

Table 90 defines the CHANGE OPERATIONAL IQ PROPERTIES response.

**Table 90 — CHANGE OPERATIONAL IQ PROPERTIES response**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (E0h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE (14h)							
11	STATUS							
12	Additional status descriptor							
...								
15								
16	Reserved							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.5 and shall be set as shown in table 90 for the CHANGE OPERATIONAL IQ PROPERTIES response.

The WORK AREA field may be set to any value by the PQI device and should be ignored by the PQI host.

The REQUEST IDENTIFIER field is defined in 10.1.5.1.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 90 for the CHANGE OPERATIONAL IQ PROPERTIES response.

The STATUS field and the additional status descriptor are defined in 10.1.5.

**10.2.9 CHANGE OPERATIONAL OQ PROPERTIES function****10.2.9.1 CHANGE OPERATIONAL OQ PROPERTIES request**

The CHANGE OPERATIONAL OQ PROPERTIES function requests that the PQI device management device server change the properties of the specified operational OQ.

Table 91 defines the CHANGE OPERATIONAL OQ PROPERTIES request.

**Table 91 — CHANGE OPERATIONAL OQ PROPERTIES request** (part 1 of 2)

Byte\Bit	7	6	5	4	3	2	1	0	
0	IU TYPE (60h)								
1	PQI COMPATIBLE FEATURES (0h)				Reserved				
2	IU LENGTH (003Ch) (LSB)								
3									(MSB)
4	Reserved								
5									
6	WORK AREA								
7									
8	REQUEST IDENTIFIER (LSB)								
9									(MSB)
10	FUNCTION CODE (15h)								
11	RsvdC								
12	OQ ID (LSB)								
13									(MSB)
14	RsvdC								
...									
40									
41	WAIT FOR REARM	RsvdC							
42	COALESCING COUNT (LSB)								
43									(MSB)
44	MINIMUM COALESCING TIME (LSB)								
...									
47	(MSB)								



**Table 91 — CHANGE OPERATIONAL OQ PROPERTIES request** (part 2 of 2)

Byte\Bit	7	6	5	4	3	2	1	0
48	MAXIMUM COALESCING TIME (LSB)							
...								
51								
52	RsvdC (MSB)							
...								
59								
60	Vendor specific							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.4 and shall be set as shown in table 91 for the CHANGE OPERATIONAL OQ PROPERTIES request.

The WORK AREA field may be set to any value by the PQI host and shall be ignored by the PQI device.

The REQUEST IDENTIFIER field is defined in 10.1.4.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 91 for the CHANGE OPERATIONAL OQ PROPERTIES function.

If the CIC bit is set to zero in the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3), then:

- a) the OQ ID field specifies the OQ ID (see 5.2.5.12.2) of the operational OQ to be configured; and
- b) if the OQ ID field is set to an OQ ID that does not exist, then the PQI device management device server shall return a CHANGE OPERATIONAL OQ PROPERTIES response with the STATUS field set to INVALID FIELD IN REQUEST IU (see 10.1.5.3).

If the CIC bit is set to one in the REPORT PQI DEVICE CAPABILITY parameter data (see 10.2.2.3), then:

- a) if no operational OQs exist, then the PQI device management device server shall return a CHANGE OPERATIONAL OQ PROPERTIES response with the STATUS field set to INVALID FIELD IN REQUEST IU (see 10.1.5.3); and
- b) if one or more operational OQs exist, then:
  - A) the OQ ID field shall be ignored;
  - B) the Coalescing Count attribute for all operational OQs shall be set to the value of the COALESCING COUNT field;
  - C) the Minimum Coalescing Time attribute for all operational OQs shall be set to the value of the MINIMUM COALESCING TIME field;
  - D) the Maximum Coalescing Time attribute for all operational OQs shall be set to the value of the MAXIMUM COALESCING TIME field; and
  - E) the Wait For Rearm attribute for all operational OQs shall be set to value of the WAIT FOR REARM bit.

The WAIT FOR REARM bit, the COALESCING COUNT field, the MINIMUM COALESCING TIME field, and the MAXIMUM COALESCING TIME field are defined in the CREATE OPERATIONAL OQ request (see 10.2.5.1).

**10.2.9.2 CHANGE OPERATIONAL OQ PROPERTIES response**

Table 92 defines the CHANGE OPERATIONAL OQ PROPERTIES response.

**Table 92 — CHANGE OPERATIONAL OQ PROPERTIES response**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (E0h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE (15h)							
11	STATUS							
12	Additional status descriptor							
...								
15								
16	Reserved							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.5 and shall be set as shown in table 92 for the CHANGE OPERATIONAL OQ PROPERTIES response.

The WORK AREA field may be set to any value by the PQI device and should be ignored by the PQI host.

The REQUEST IDENTIFIER field is defined in 10.1.5.1.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 92 for the CHANGE OPERATIONAL OQ PROPERTIES response.

The STATUS field and the additional status descriptor are defined in 10.1.5.

## 10.2.10 REPORT OPERATIONAL IQ LIST function

### 10.2.10.1 REPORT OPERATIONAL IQ LIST request

The REPORT IQ LIST function requests that the PQI device management device server return a list of all existing operational IQs and their properties in the Data-In Buffer as defined in 10.2.10.3.

Table 93 defines the REPORT OPERATIONAL IQ LIST request.

**Table 93 — REPORT OPERATIONAL IQ LIST request**

Byte\Bit	7	6	5	4	3	2	1	0	
0	IU TYPE (60h)								
1	PQI COMPATIBLE FEATURES (0h)				Reserved				
2	IU LENGTH (003Ch) (LSB)								
3									(MSB)
4	Reserved								
5									
6	WORK AREA								
7									
8	REQUEST IDENTIFIER (LSB)								
9									(MSB)
10	FUNCTION CODE (16h)								
11	RsvdC								
...									
43									
44	DATA-IN BUFFER SIZE (LSB)								
...									
47	(MSB)								
48	SGL descriptor (see table 53)								
...									
63									

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.4 and shall be set as shown in table 93 for the REPORT OPERATIONAL IQ LIST request.

The WORK AREA field may be set to any value by the PQI host and shall be ignored by the PQI device.

The REQUEST IDENTIFIER field is defined in 10.1.4.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 93 for the REPORT OPERATIONAL IQ LIST request.

The DATA-IN BUFFER SIZE field is defined in 10.1.4.

The SGL descriptor is defined in 10.1.4.

### 10.2.10.2 REPORT OPERATIONAL IQ LIST response

Table 94 defines the REPORT OPERATIONAL IQ LIST response.

**Table 94 — REPORT OPERATIONAL IQ LIST response**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (E0h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE (16h)							
11	STATUS							
12	Additional status descriptor							
...								
15								
16	Reserved							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.5 and shall be set as shown in table 94 for the REPORT OPERATIONAL IQ LIST response.

The WORK AREA field may be set to any value by the PQI device and should be ignored by the PQI host.

The REQUEST IDENTIFIER field is defined in 10.1.5.1.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 94 for the REPORT OPERATIONAL IQ LIST response.

The STATUS field and the additional status descriptor are defined in 10.1.5.

## 10.2.10.3 REPORT OPERATIONAL IQ LIST parameter data

Table 95 defines the REPORT OPERATIONAL IQ LIST parameter data.

**Table 95 — REPORT OPERATIONAL IQ LIST parameter data (i.e., Data-In Buffer contents)**

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
...								
5								
6	(LSB)							
7	NUMBER OF OPERATIONAL IQ PROPERTY DESCRIPTORS							
	(MSB)							
Operational IQ property descriptor list								
8	Operational IQ property descriptor [first] (see table 96)							
...								
135								
...	...							
m-127	Operational IQ property descriptor [last] (see table 96)							
...								
m								

The NUMBER OF OPERATIONAL IQ PROPERTY DESCRIPTORS field indicates the number of operational IQ property descriptors in the operational IQ property descriptor list.

Table 96 defines the operational IQ property descriptor.

**Table 96 — Operational IQ property descriptor (part 1 of 2)**

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
...								
11								
12	(LSB)							
13	(MSB)	IQ ID						
14	Reserved							IQ ERROR
15	Reserved							
16	(LSB)							
...	IQ ELEMENT ARRAY ADDRESS							
23	(MSB)							

Table 96 — Operational IQ property descriptor (part 2 of 2)

Byte\Bit	7	6	5	4	3	2	1	0	
24	IQ CI ADDRESS (LSB)								
...									
31									(MSB)
32	NUMBER OF ELEMENTS (LSB)								
33									(MSB)
34	ELEMENT LENGTH (LSB)								
35									(MSB)
36	Reserved			OPERATIONAL QUEUE PROTOCOL					
37	Reserved								
...									
59									
60	Vendor specific								
...									
63									
Bytes corresponding to the CREATE OPERATIONAL IQ response									
64	IQ PI OFFSET (LSB)								
...									
71									(MSB)
72	Reserved								
...									
127									

The IQ ID field indicates the IQ ID (see 5.2.5.9.2) of the operational IQ being described.

An IQ ERROR bit set to one indicates that the PQI device has stopped consuming from the operational IQ due to an error. An IQ ERROR bit set to zero indicates that the PQI device has not stopped consuming from the operational IQ due to an error.

The IQ ELEMENT ARRAY ADDRESS field indicates the operational IQ element array address.

The IQ CI ADDRESS field indicates the operational IQ CI address.

The NUMBER OF ELEMENTS field indicates the number of elements in the operational IQ element array.

The ELEMENT LENGTH field indicates the element length in 16-byte increments (e.g., a value of 0001h in the ELEMENT LENGTH field specifies an element length of 16 bytes).

The OPERATIONAL QUEUE PROTOCOL field (see table 81) indicates the operational queue protocol used by the operational IQ (see 5.3.1).

The IQ PI OFFSET field indicates the offset in PQI device memory space of the operational IQ PI (i.e., the operational IQ PI address is the memory address contained in the first PCI memory BAR plus the IQ PI OFFSET field).

## 10.2.11 REPORT OPERATIONAL OQ LIST function

### 10.2.11.1 REPORT OPERATIONAL OQ LIST request

The REPORT OPERATIONAL OQ LIST function requests that the PQI device management device server return a list of all existing operational OQs and their properties in the Data-In Buffer as defined in 10.2.11.3.

Table 97 defines the REPORT OPERATIONAL OQ LIST request.

**Table 97 — REPORT OPERATIONAL OQ LIST request**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (60h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE (17h)							
11	RsvdC							
...								
43								
44	DATA-IN BUFFER SIZE (LSB)							
...								
47	(MSB)							
48	SGL descriptor (see table 53)							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.4 and shall be set as shown in table 97 for the REPORT OPERATIONAL OQ LIST request.

The WORK AREA field may be set to any value by the PQI host and shall be ignored by the PQI device.

The REQUEST IDENTIFIER field is defined in 10.1.4.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 97 for the REPORT OPERATIONAL OQ LIST request.

The DATA-IN BUFFER SIZE field is defined in 10.1.4.

The SGL descriptor is defined in 10.1.4.

### 10.2.11.2 REPORT OPERATIONAL OQ LIST response

Table 98 defines the REPORT OPERATIONAL OQ LIST response.

**Table 98 — REPORT OPERATIONAL OQ LIST response**

Byte\Bit	7	6	5	4	3	2	1	0
0	IU TYPE (E0h)							
1	PQI COMPATIBLE FEATURES (0h)				Reserved			
2	IU LENGTH (003Ch) (LSB)							
3								
4	Reserved							
5								
6	WORK AREA							
7								
8	REQUEST IDENTIFIER (LSB)							
9								
10	FUNCTION CODE (17h)							
11	STATUS							
12	Additional status descriptor							
...								
15								
16	Reserved							
...								
63								

The IU TYPE field, the PQI COMPATIBLE FEATURES field, and the IU LENGTH field are defined in 10.1.5 and shall be set as shown in table 98 for the REPORT OPERATIONAL OQ LIST response.

The WORK AREA field may be set to any value by the PQI device and should be ignored by the PQI host.

The REQUEST IDENTIFIER field is defined in 10.1.5.1.

The FUNCTION CODE field is defined in 10.2.1 and shall be set as shown in table 98 for the REPORT OPERATIONAL OQ LIST response.

The STATUS field and the additional status descriptor are defined in 10.1.5.



## 10.2.11.3 REPORT OPERATIONAL OQ LIST parameter data

Table 99 defines the REPORT OPERATIONAL OQ LIST parameter data.

**Table 99 — REPORT OPERATIONAL OQ LIST parameter data (i.e., Data-In Buffer contents)**

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
...								
5								
6	NUMBER OF OPERATIONAL OQ PROPERTY DESCRIPTORS (LSB)							
7								
Operational OQ property descriptor list								
8	Operational OQ property descriptor [first] (see table 100)							
...								
135								
...	...							
m-127	Operational OQ property descriptor [last] (see table 100)							
...								
m								

The NUMBER OF OPERATIONAL OQ PROPERTY DESCRIPTORS field indicates the number of operational OQ property descriptors in the operational OQ property descriptor list.

Table 100 defines the operational OQ property descriptor.

**Table 100 — Operational OQ property descriptor** (part 1 of 2)

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
...								
11								
12	(LSB)							
13	(MSB)							
14	Reserved							OQ ERROR
15	Reserved							
16	(LSB)							
...	OQ ELEMENT ARRAY ADDRESS							
23	(MSB)							
24	(LSB)							
...	OQ PI ADDRESS							
31	(MSB)							
32	(LSB)							
33	(MSB)							
34	(LSB)							
35	(MSB)							
36	Reserved			OPERATIONAL QUEUE PROTOCOL				
37	Reserved							
...								
39								
40	(LSB)							
41	WAIT FOR REARM	Reserved				(MSB)		
42	(LSB)							
43	(MSB)							
44	(LSB)							
...	MINIMUM COALESCING TIME							
47	(MSB)							

Table 100 — Operational OQ property descriptor (part 2 of 2)

Byte/Bit	7	6	5	4	3	2	1	0
48	MAXIMUM COALESCING TIME (LSB)							
...								
51								
52	Reserved							
...								
59								
60	Vendor specific							
...								
63								
Bytes corresponding to the CREATE OPERATIONAL OQ response								
64	OQ CI OFFSET (LSB)							
...								
71								
72	Reserved							
...								
127								

The OQ ID field indicates the OQ ID (see 5.2.5.12.2) of the operational OQ being described.

An OQ ERROR bit set to one indicates that the PQI device has stopped producing to the operational OQ due to an error. An OQ ERROR bit set to zero indicates that the PQI device is producing to the operational OQ.

The OQ ELEMENT ARRAY ADDRESS field indicates the operational OQ element array address.

The OQ PI ADDRESS field indicates the operational QQ PI address.

The NUMBER OF ELEMENTS field indicates the number of elements in the operational OQ element array.

The ELEMENT LENGTH field indicates the length in 16-byte increments of elements in the operational OQ (e.g., a value of 01h in the ELEMENT LENGTH field specifies an element length of 16 bytes).

The OPERATIONAL QUEUE PROTOCOL field (see table 81) indicates the operational queue protocol used by the operational OQ (see 5.3.1).

The INTERRUPT MESSAGE NUMBER field indicates the MSI-X Table entry used to generate the interrupt message for updates to the operational OQ PI in MSI-X mode (see PCI).

The WAIT FOR REARM bit, the COALESCING COUNT field, the MINIMUM COALESCING TIME field, and the MAXIMUM COALESCING TIME field control interrupt coalescing parameters in MSI-X mode (see 5.4.2.3).

The WAIT FOR REARM bit is defined in 10.2.5.1.

The COALESCING COUNT field is defined in 10.2.5.1.

The MINIMUM COALESCING TIME field is defined in 10.2.5.1.

The MAXIMUM COALESCING TIME field is defined in 10.2.5.1.

The OQ CI OFFSET field indicates the offset in PQI device memory space of the operational OQ CI (i.e., the operational OQ CI address is the memory address contained in the first PCI memory BAR plus the OQ CI OFFSET field).

## Annex A

(normative)

### Alternative SGL segment

An alternative SGL segment (see table A.2):

- a) is a kind of SGL segment (see 8.2) pointed to by a Last Alternative SGL Segment descriptor (see table A.1); and
- b) contains Alternative Data Block descriptors (see table A.3).

The length and format of the Alternative Data Block descriptor differs from that of the SGL descriptors defined in 8.3 (i.e., 20 bytes rather than 16 bytes).

An alternative SGL segment is a last SGL segment.

Table A.1 defines the Last Alternative SGL Segment descriptor.

**Table A.1 — Last Alternative SGL Segment descriptor**

Byte\Bit	7	6	5	4	3	2	1	0
0	(LSB)						Reserved	
...	ADDRESS							
7								
8	(LSB)							
...	NUMBER OF DESCRIPTORS							
11								
12	Reserved							
...								
14								
15	SGL DESCRIPTOR TYPE (4h)				Reserved			

The ADDRESS field specifies the upper 62 bits of the 64-bit memory space address of the next SGL segment, which is the last SGL segment (see 8.2) and is an alternative SGL segment (see table A.2). The least significant two bits of the 64-bit memory space address of the next SGL segment, which are not specified by the ADDRESS field, are zero.

The NUMBER OF DESCRIPTORS field specifies the number of Alternative Data Block descriptors in the alternative SGL segment. If the NUMBER OF DESCRIPTORS field is set to zero, then the Alternative Data Block descriptor shall be processed as having an error (see 10.1.4 and the IU layer standard (e.g., SOP)).

If the value in the ADDRESS field plus the value in the NUMBER OF DESCRIPTORS field times 20 (i.e., the length in bytes of each Alternative Data Block descriptor) is greater than 1\_00000000\_00000000h, then the Alternative Data Block descriptor shall be processed as having an error (see 10.1.4 and the IU layer standard (e.g., SOP)).

The SGL DESCRIPTOR TYPE field is defined in 8.3.1 and shall be set as shown in table A.1 for the Last Alternative SGL Segment descriptor.

Table A.2 defines the alternative SGL segment.

**Table A.2 — Alternative SGL segment**

Byte\Bit	7	6	5	4	3	2	1	0
0	Alternative Data Block descriptor [first] (see table A.3)							
...								
19								
...	...							
n - 19	Alternative Data Block descriptor [last] (see table A.3)							
...								
n								

The alternative SGL segment shall contain at least one Alternative Data Block descriptor.

Table A.3 defines the Alternative Data Block descriptor.

**Table A.3 — Alternative Data Block descriptor**

Byte\Bit	7	6	5	4	3	2	1	0
0	ADDRESS (LSB)							
...								
7								
8	LENGTH (LSB)							
...								
11								
12	Vendor specific							
...								
19								

The ADDRESS field specifies the starting 64-bit memory address of a data block.

The LENGTH field specifies the length in bytes of the data block. A LENGTH field set to 00000000h specifies that no data is transferred. An Alternative Data Block descriptor specifying that no data is transferred shall not be processed as having an error.

If the value in the ADDRESS field plus the value in the LENGTH field is greater than 1\_00000000\_00000000h, then the Alternative Data Block descriptor shall be processed as having an error (see 10.1.4 and the IU layer standard (e.g., SOP)).

## Annex B

(informative)

### Operating system suggestions

#### B.1 Power actions

If a Windows® operating system Storport miniport driver's **HwStorBuildIo** function is invoked with an *Srb* parameter pointing to a SCSI\_POWER\_REQUEST\_BLOCK (i.e., the Function member is set to SRB\_FUNCTION\_POWER), then the miniport driver should set the fields in the PQI Device Power Action register (see 6.2.21) as follows:

- a) set the SYSTEM POWER ACTION field as described in table B.1; and

**Table B.1 — Windows PowerAction member to SYSTEM POWER ACTION field**

PowerAction member	SYSTEM POWER ACTION field
0 (i.e., StorPowerActionNone)	00h (i.e., no action)
1 (i.e., StorPowerActionReserved)	00h (i.e., no action)
2 (i.e., StorPowerActionSleep)	10h (i.e., S1, S2, or S3)
3 (i.e., StorPowerActionHibernate)	14h (i.e., S4)
4 (i.e., StorPowerActionShutdown)	02h (i.e., S0 or S5)
5 (i.e., StorPowerActionShutdownReset)	01h (i.e., S0)
6 (i.e., StorPowerActionShutdownOff)	15h (i.e., S5)
7 (i.e., StorPowerActionWarmEject)	00h (i.e., no action)

- b) set the DEVICE POWER ACTION field as described in table B.2.

**Table B.2 — Windows DevicePowerState member to DEVICE POWER ACTION field**

DevicePowerState member	DEVICE POWER ACTION field
0 (i.e., StorPowerDeviceUnspecified)	00h (i.e., no action)
1 (i.e., StorPowerDeviceD0)	10h (i.e., D0)
2 (i.e., StorPowerDeviceD1)	11h (i.e., D1)
3 (i.e., StorPowerDeviceD2)	12h (i.e., D2)
4 (i.e., StorPowerDeviceD3)	13h (i.e., D3)

NOTE 11 - Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

NOTE 12 - For information on Windows, see the Microsoft Developer Network web site (see <http://msdn.microsoft.com>).

## **Annex C**

(informative)

### **SGL examples**

#### **C.1 Data transfer from a source data stream to a destination data buffer**

Figure C.1 shows an example of a data transfer from a source data stream to a destination data buffer in memory space described by an SGL. The total length of the source data stream is 13 KiB and the total length of the destination data buffer is 11 KiB. The destination data buffer is described by an SGL that contains three standard SGL segments. The three standard SGL segments contain a total of three Data Block descriptors with lengths of 3 KiB, 4 KiB, and 4 KiB respectively. Standard SGL segment 1 of the Destination SGL contains a Bit Bucket descriptor (see 8.3.3) with a length of 2 KiB that specifies discarding (i.e., not transferring) 2 KiB of data of the source data stream. SGL segment 1 also contains a Last Standard SGL Segment descriptor (see 8.3.5) specifying that the standard SGL segment pointed to by the descriptor is the last standard SGL segment.



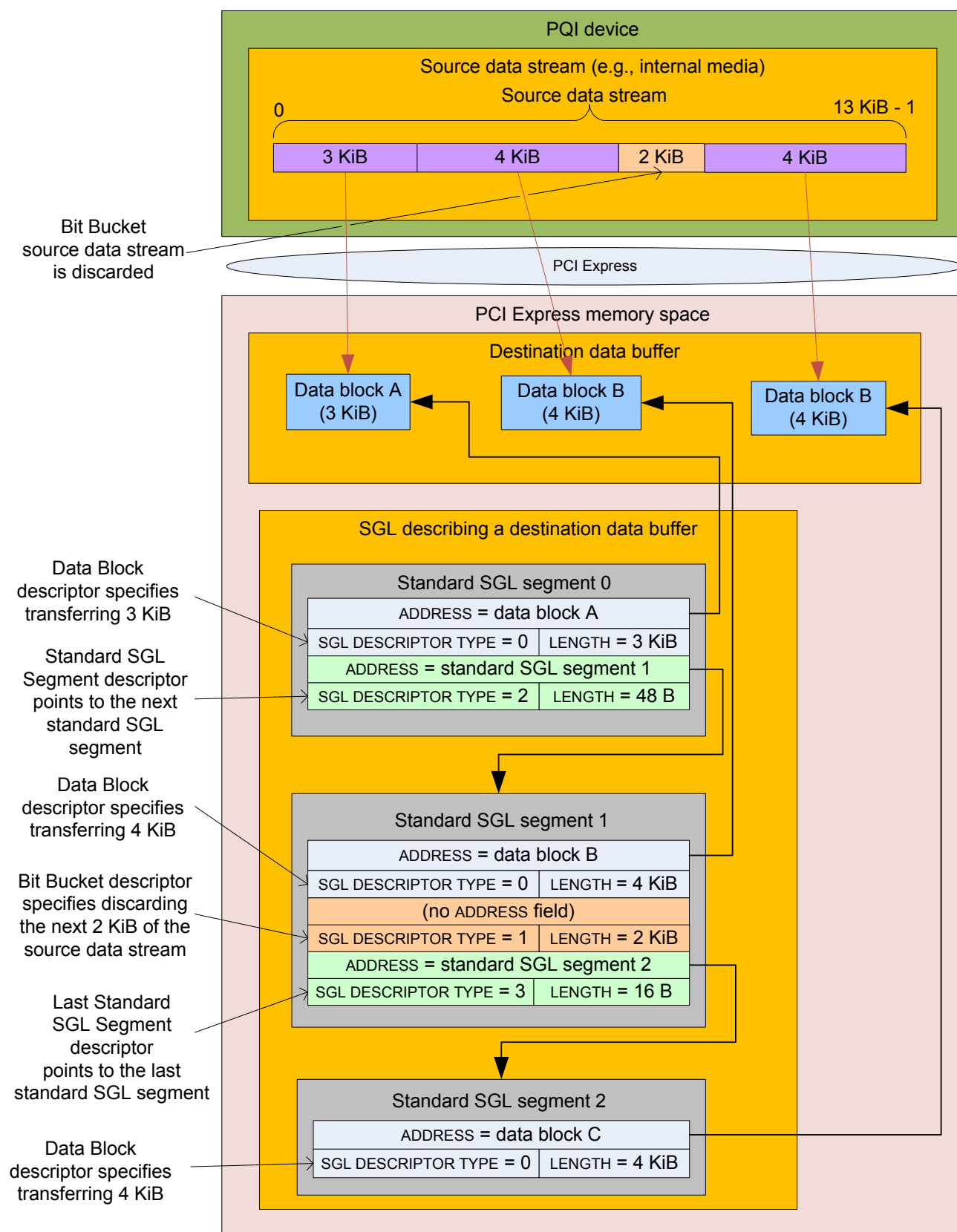


Figure C.1 — SGL example of a data transfer from a source data stream to a destination data buffer

## C.2 Data transfer from a source data buffer to a destination data stream

Figure C.2 shows an example of a data transfer from a source data buffer in memory space described by an SGL to a destination data stream. The total length of the destination data stream is 11 KiB and the total length of the source data buffer is 11 KiB. The source data buffer is described by an SGL that contains one standard SGL segment that contain three Data Block descriptors with lengths of 3 KiB, 4 KiB, and 4 KiB respectively. The source SGL also contains a Bit Bucket descriptor with a length of 2 KiB that is ignored because the SGL describes a source data buffer.

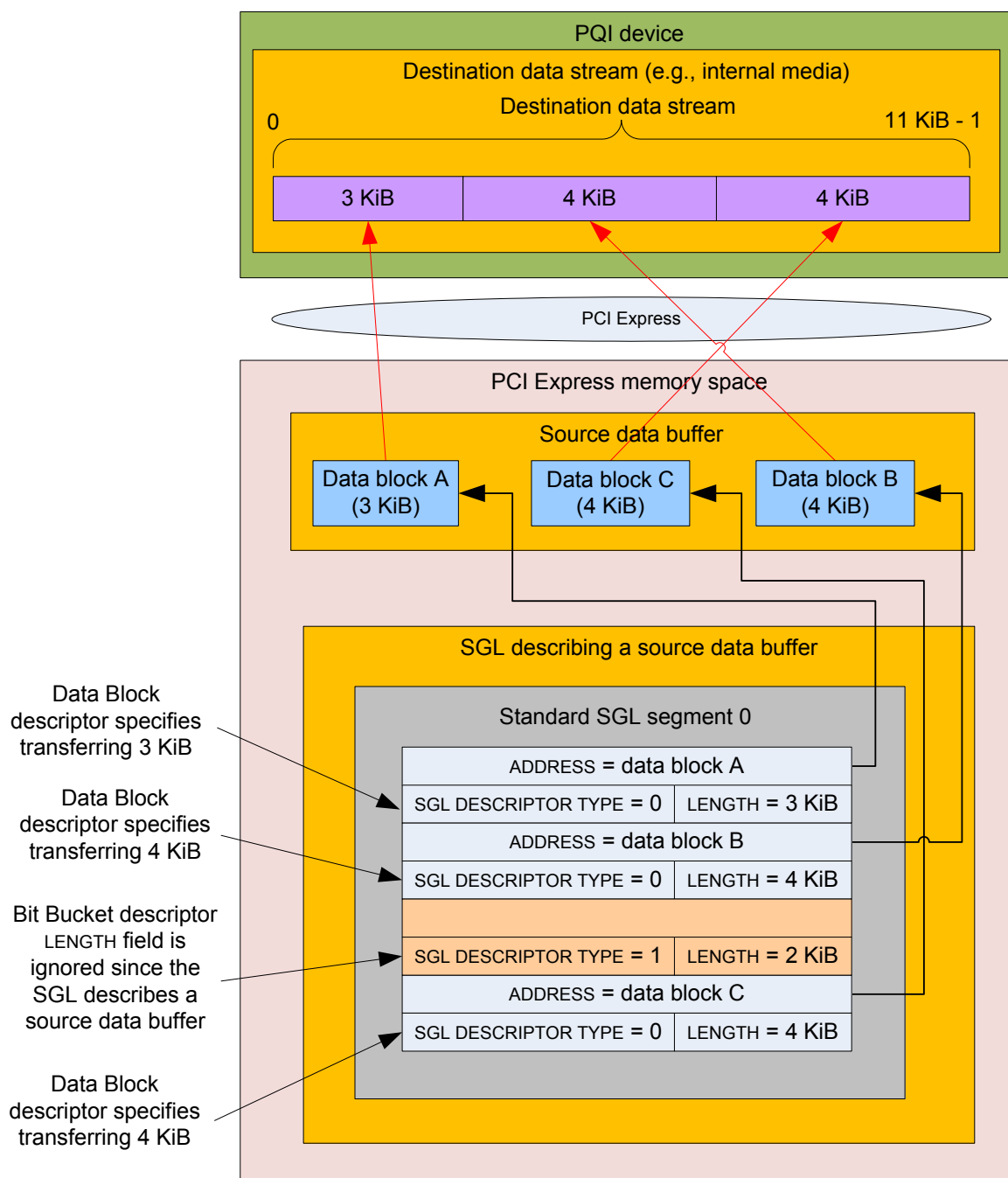


Figure C.2 — SGL example of a data transfer from a source data buffer to a destination data stream

### C.3 Memory to memory data transfer

Figure C.3 shows an example of a 12 KiB memory to memory data transfer from a source data buffer described by an SGL to a destination data buffer described by an SGL. The total length of the destination data buffer is 12 KiB and the total transferable length of the source data buffer is 12 KiB. The source data buffer is described by an SGL that contains one standard SGL segment that contains two Data Block descriptors (see 8.3.2) with lengths of 6 KiB and 6 KiB respectively. The destination data buffer is described by an SGL made up of one standard SGL segment that contains three SGL Data Buffer descriptors with lengths of 4 KiB, 4 KiB, and 4 KiB respectively.

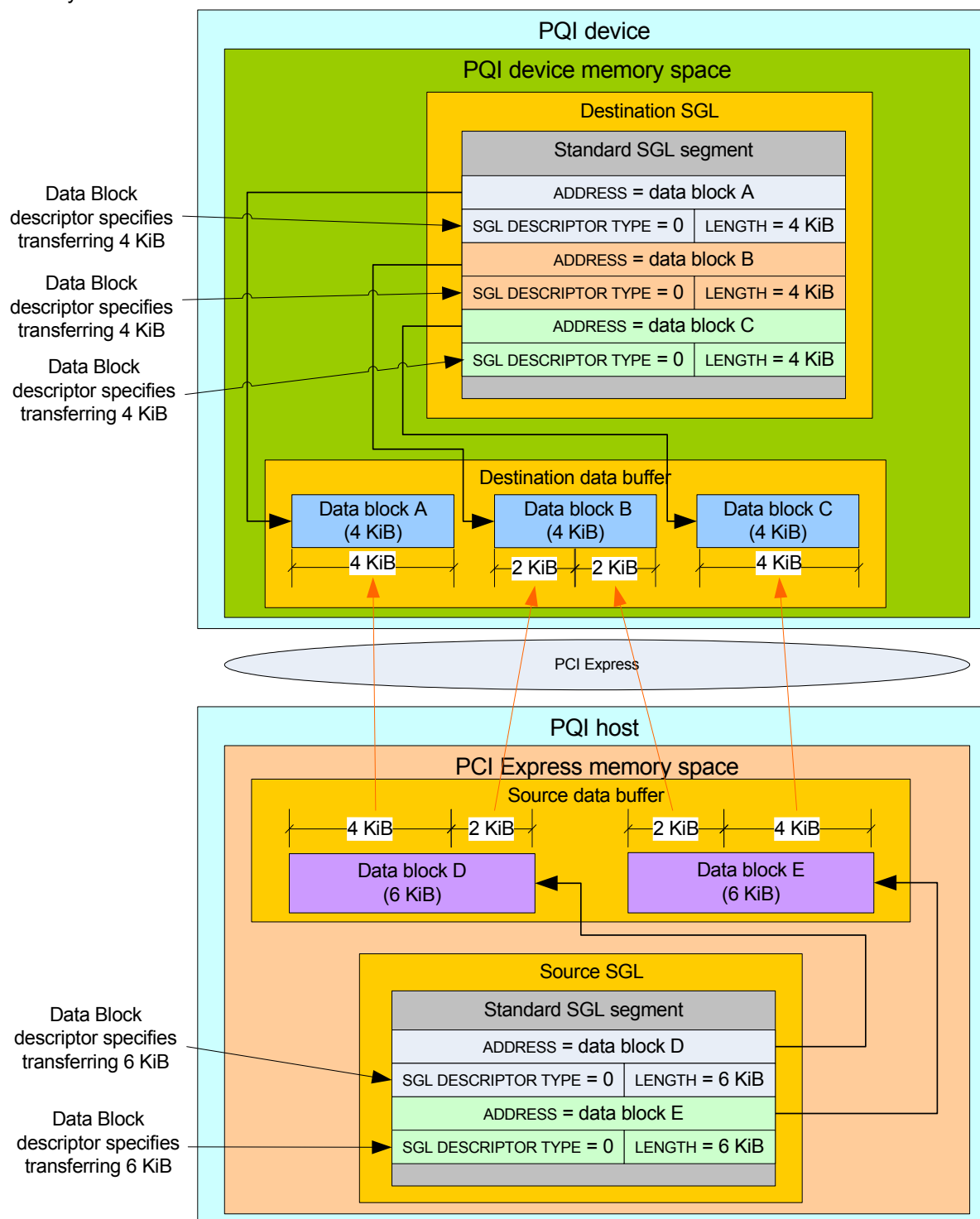


Figure C.3 — SGL example of a memory to memory data transfer

## **Annex D**

(informative)

### **Bibliography**

ISO/IEC 14776-323, *SCSI Block Commands - 3 (SBC-3)* (T10/BSR INCITS 514)

ISO/IEC 14776-334, *SCSI Stream Commands - 4 (SSC-4)* (T10/BSR INCITS 516)

ISO/IEC 14776-262, *SAS Protocol Layer - 2 (SPL-2)* (T10/BSR INCITS 505)

ISO/IEC 14776-154, *Serial Attached SCSI - 3 (SAS-3)* (T10/BSR INCITS 519)