



With



***Now includes NVMe 1.3 &
NVMe Management Interface***



NVMe 1.3 Rev.11 ■ 32.E.19

KnowledgeTek, Inc.

8690 Wolff Court, Suite 110, Westminster, CO 80031 • (303) 465-1800 • e-mail:seminars@knowledgetek.com • www.knowledgetek.com

Copyright Notice

Copyright ©2017 KnowledgeTek, Inc.

All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from KnowledgeTek, Inc.



Disclaimer

This material is not a specification for any interface or system.

KnowledgeTek, Inc. and the instructor make no warranty and assume no liability arising from the application or use of any product, hardware, software, system, circuit, or anything else described herein.

KnowledgeTek, Inc. and the instructor assume no responsibility for errors appearing in this document.

KnowledgeTek, Inc. and the instructor assume no responsibility for any claims that the concepts or details discussed in the seminar, or disclosed in the course materials, are proprietary to any person or company.

Course participants are urged to clear any designs proposed for products with their patent and copyright council.

Company, brand, product, trade, and other names used in this document are trademarks of their respective holders.



Contact KnowledgeTek

KnowledgeTek, Inc.

Technical Training
8690 Wolff Court, Suite 110
Westminster, Colorado 80031
Voice: (303) 465-1800
Fax: (303) 317-8972
E-mail: seminars@knowledgetek.com
Website: www.knowledgetek.com



KnowledgeTek Training

KnowledgeTek specializes in technical training on computer peripherals and related technologies.

INTERFACES

Serial Attached SCSI (SAS 3) Seminar – 3 Days
Serial ATA (SATA 3.3) Seminar – 2 Days
Serial ATA (SATA 3.3) &
 Serial Attached SCSI (SAS 3) Seminar – 4 Days
PCI Express (PCIe) – 2 Day or 3 Day Seminars
NVM Express Seminar – 1 Day
NEW! NVMe over Fabrics – 1 Day
PCIe & NVMe Combination Seminar – 4 Days
SCSI Commands Seminar – 1 Day
ATA Command Set (ACS-4) Seminar – 1 Day

STORAGE TECHNOLOGIES

Solid-State Drive Technology (SSDT) Seminar – 2 Days
NEW! Fundamentals of 3D Non-Volatile Memory – ½ Day
Hybrid-Disk Drive Technology (HDDT) Seminar – 3 Days

½ DAY OVERVIEW SEMINARS

KnowledgeTek also offers ½ day overviews of our courses for those non-technical people who need to get up-to-speed on these subjects.

NVMe
PCIe
SSDT
SATA
SAS
SCSI Commands
ATA Commands



About the Instructor

Hugh Curley began working on mainframe computers in 1967 and expanded to personal computers in 1981. His background includes hands-on technical and managerial experience in field service, system-level test in manufacturing, and system-level test in engineering. In 1975 Hugh began teaching computers to engineers and discovered that he not only had good skills for the classroom process, but that he enjoyed teaching working engineers. Hugh has accumulated extensive experience in developing and presenting highly technical courses to engineering specialists from different disciplines. He applies that experience and skill to every course he presents.

Hugh is an expert in data storage interfaces. He currently teaches SATA, SAS, USB 3, NVMe, PCIe, SCSI commands and ATA commands (ACS-4). He has also taught SCSI Nuts and Bolts, IDE Nuts and Bolts, ATAPI, IEEE-1394, iSCSI, CE-ATA, and Disk Drive Technology. As an instructor for KnowledgeTek, Hugh has successfully presented hundreds of interface courses.



Mr. Curley can be reached at hugh@hughcurley.com

Table of Contents

- Sect. 1 – Overview
- Sect. 2.1 – PCIe Basics
- Sect. 2.2 – PCIe Details for NVMe
- Sect. 3 – NVMe Registers
- Sect. 4 – Commands Formats
- Sect. 5 – Admin Commands
- Sect. 6 – NVMe Commands
- Sect. 7 – Out-of-Band Management Interface
- Sect. 8 – NVMe 1.3
- Sect. 9 – Translating SCSI Commands
- Sect. 10 – Future Directions



Notes



Section 1

Overview



NVM Express

Section 1: Overview

Covered in this Course

NVMe™ Overview

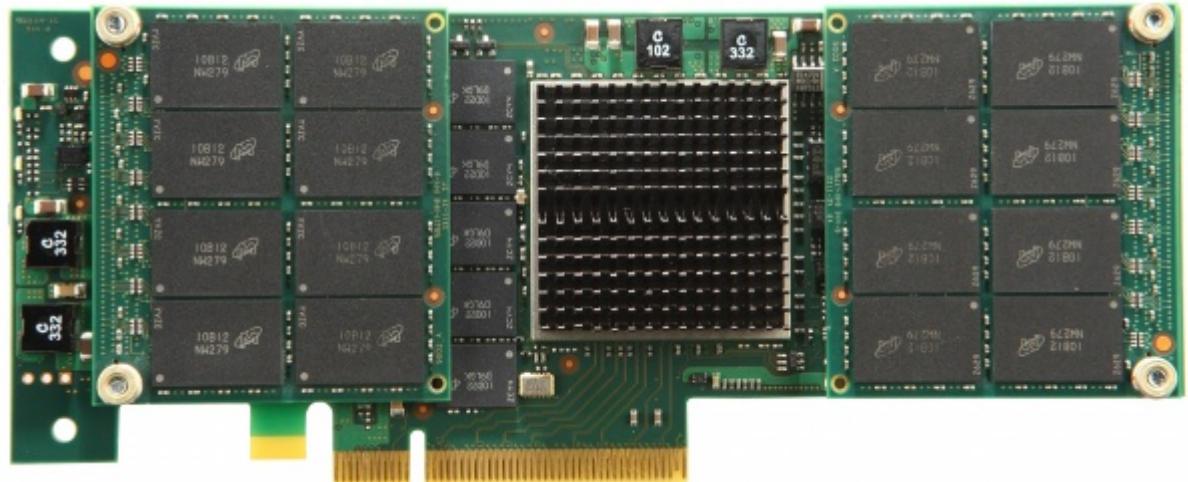
NVMe Queues

NVMe PCIe

NVMe Commands

NVMe Management Interface – NVMe-MI™

NVMe – Non-Volatile
Memory Express



NVM Express

Section 1: Overview

Covered in this Section

What is NVMe

Why NVMe?

Overview of how NVMe works

Queues

Doorbell

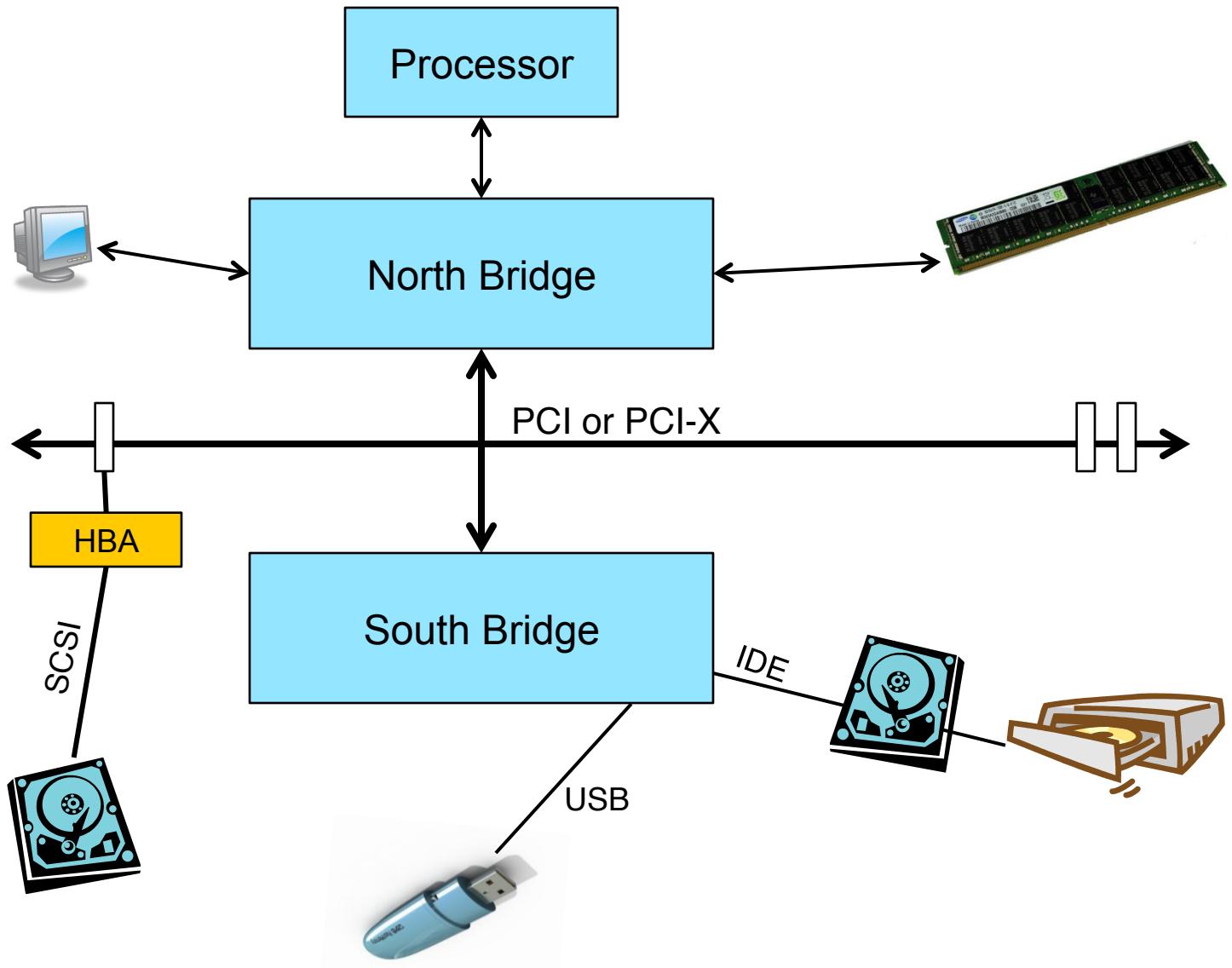
RDMA

Power States

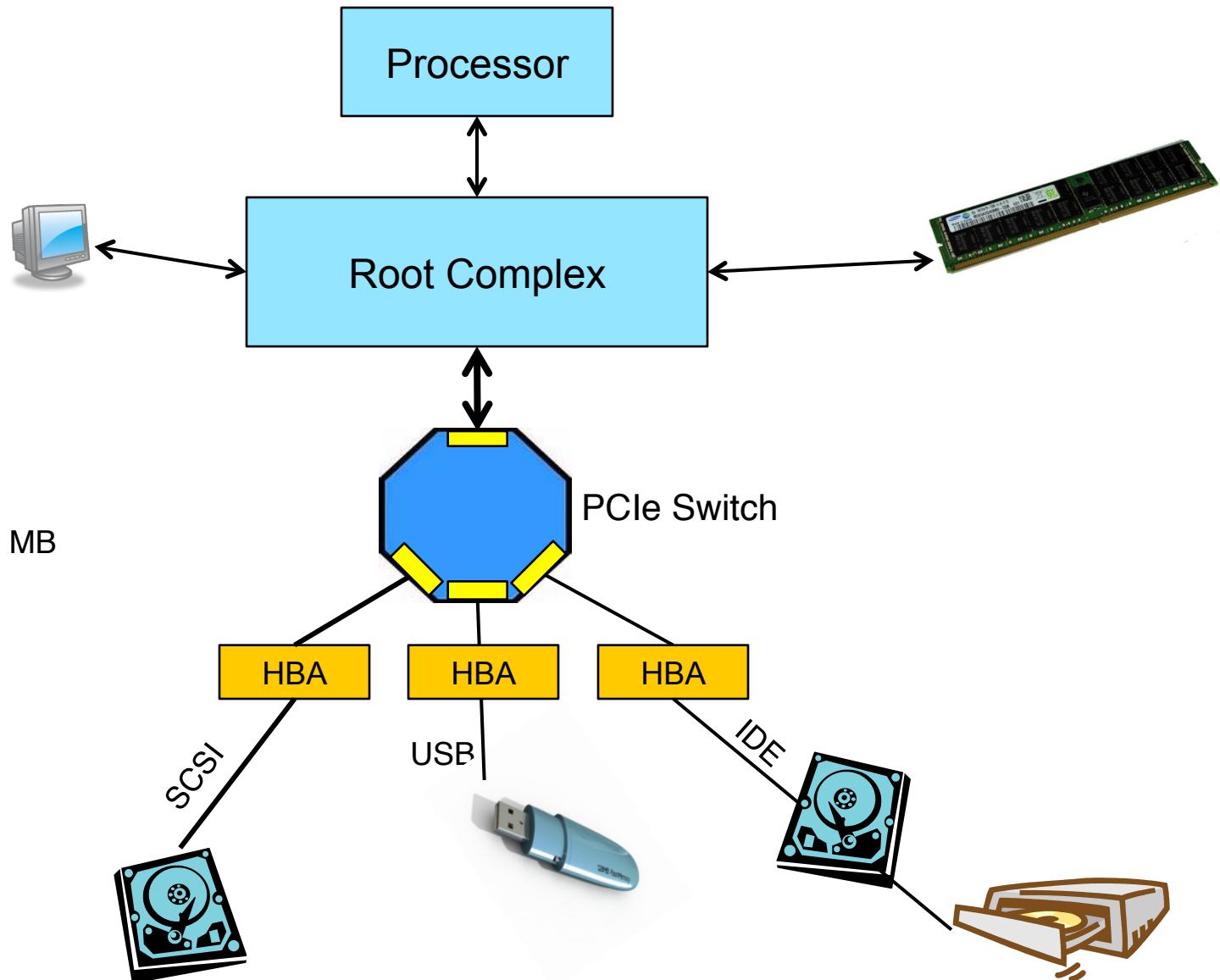
Resets



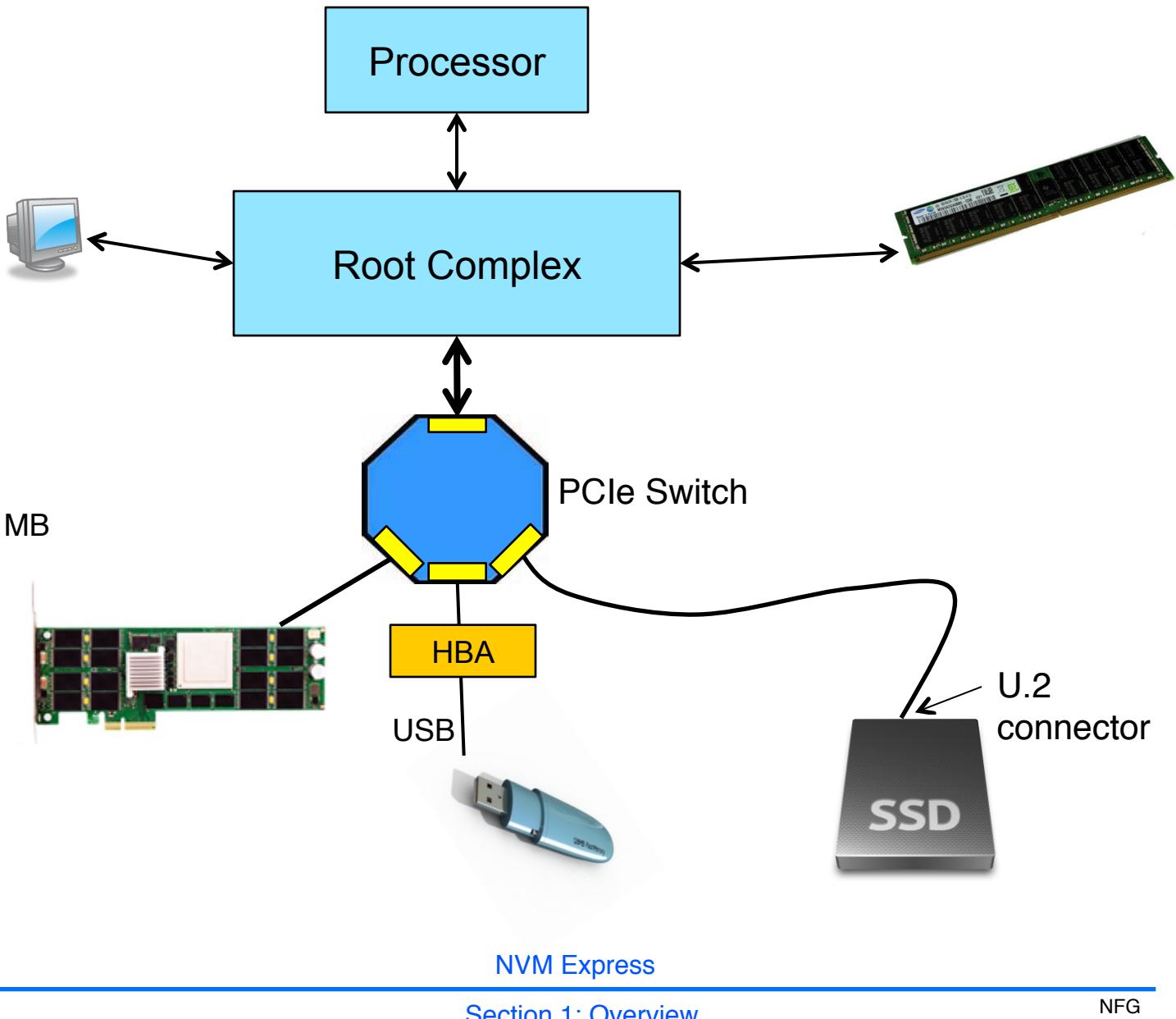
PCI Computer Block Diagram (mid-90's)



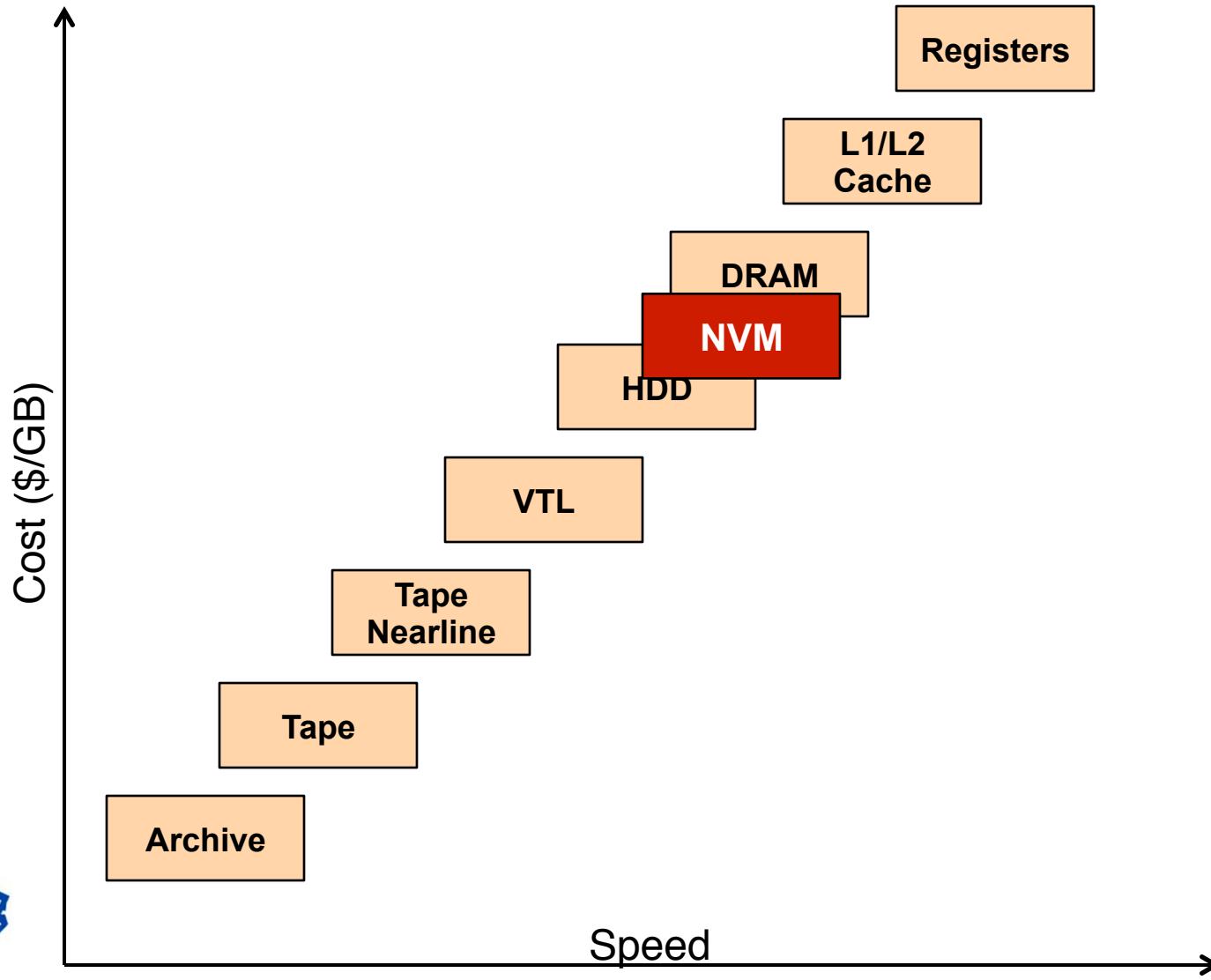
PCI Computer Block Diagram (Current)



PCI Computer Block Diagram (Soon)



Storage Hierarchy



Bottlenecks of HDD

SAS To Computer

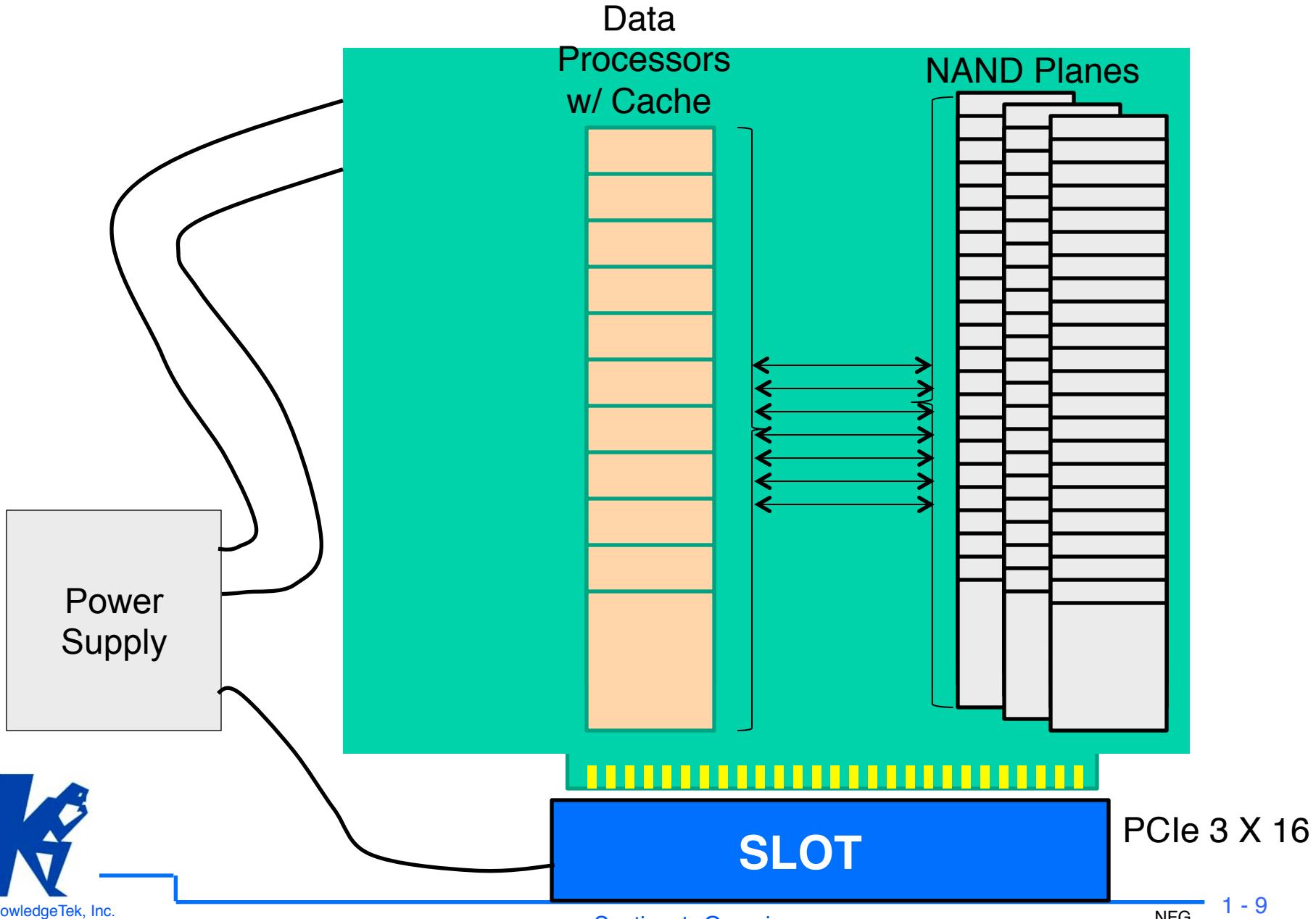
Processor
&
Cache

Read Channel

Power
Supply



Possibilities of PCIe Drives



Problems with Existing I/O Protocols

Signals must be converted from host language or PCI language to I/O language

SCSI and ATA command sets and protocols were designed for slow devices with long seek times

SSDs are more like memory than I/O:

- No seek times
- Memory-mapped access
- Massively parallel access

SSDs can be mounted on PCIe cards without need for external I/O cables



What is NVMe?

NVMe is:

- A short name for Non-Volatile Memory Express
- Designed from the beginning to work with SSDs on PCIe
- Initially designed for Enterprise, Data Center and Client systems;
- Later expanded to HPC, laptops and tablets
- Optimized commands and completion path
- Scalable based on requirements
- Expandable based on future needs

NVMe is based on:

- Paired Submission and Completion Queues
- Data transferred by PCIe transactions

Website: nvmexpress.org

HPC – High
Performance
Computing



What does NVMe Include?

NVMe includes:

- Driver Interface
- Queue Interface
- Command Set
- Command Processing procedure



Features/Benefits of NVMe (Part 1 of 2)

SSD (Storage)

- Zero seek time
- Parallel Read Channels
- Low latency
- Higher MTBF

*MTBF – Mean time
between failure*

PCIe (Transport)

- Up to 75 Watts available at the slot
- Faster (Gen 3 X 16 = 16 GB/s each direction)
- No HBA

*HBA – Host Bus
Adapter*



Features/Benefits of NVMe (Part 2 of 2)

NVMe (Communication/Protocol)

Streamline command set

Expandable

 number and size of queues

 Function required memory can be on the Function

Single command set and protocol for desktop, client,
 workstation, server, enterprise, HPC

End-to-end data protection

No intermediate transport conversion

Multiple Namespaces per Function for flexible data/device management

HPC – High Performance Computing



NVMe Attributes

- Maximum of 1 MMIO register write is necessary to issue commands
- Support for up to 64K I/O queues with each queue supporting 64K commands
- Priority associated with each I/O queue
- All information to complete a read is in 64B command
- Efficient and streamlined command set
- Support for MSI/MSI-X and interrupt aggregation
- Support for Multiple namespaces
- Efficient support for I/O virtualization architectures
- Enterprise support for end-to-end data protection (DIF/DIX)
- Enterprise: support for multi-path I/O including reservations

DIF = Data
Integrity Field

DIX = Data
Integrity
Extensions

MMIO = Memory
Mapped Input/
Output

MSI = Message
Sighaled Interrupt

NVMe Subsystem

An NVM subsystem includes:

one or more Controllers,

one or more Namespaces,

one or more PCI Express ports,

a non-volatile memory storage medium*, and

an interface between the controller(s) and non-volatile memory storage medium.

* Non-volatile memory storage medium could be an SSD



A Few Definitions

NVMe

Originally, Non-Volatile Memory over PCI Express
Now, Non-Volatile Memory Express

Controller

A PCIe Function that implements NVMe

Namespace

A group of contiguous Logical Blocks



A Few More Definitions

Link

The collection of 2 ports and their interconnecting lanes.

A link is a dual-simplex communications path between 2 components

Port

1. Logically, an interface between a component and a PCIe Link
2. Physically, a group of transmitters and receivers located on the same chip that define a Link

Physical Layer

The layer that directly interacts with the communication medium between two components

Phy

Not officially defined by PCIe or NVMe specifications

Component

A physical device (a single package)

Competing Standards?

SCSI Express

The name of a marketing project within the STA.
Not a standards specification

SATA Express

Defines a connector that accepts a SATA HDD, SATA SSD or PCIe SSD



Almost blank page for spacing





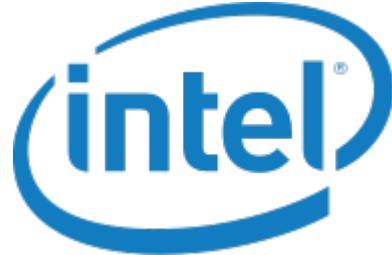
NetApp



SEAGATE



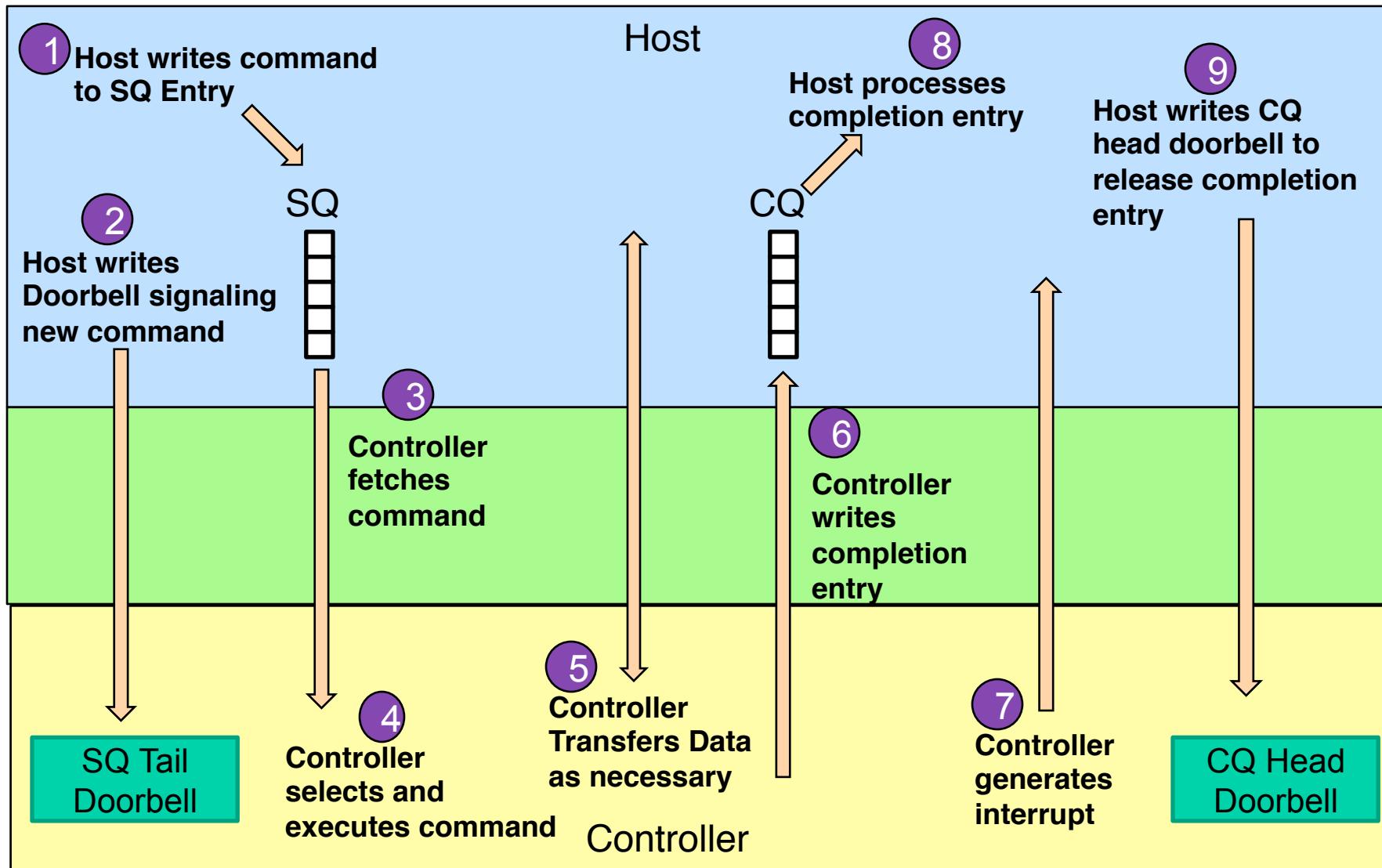
Promoter Group



ORACLE®



NVMe Command Processing (Picture – Informative)



NVMe Command Processing (Words – Informative)

1. Host creates a command and places it in the appropriate SQ in host memory
2. Host updates the SQ Tail Doorbell with the new tail entry pointer
3. Controller fetches the command(s)
4. Controller performs command arbitration, selects and executes a command
5. Controller transfers data as necessary
6. Controller writes a completion queue entry to associated CQ in host memory
7. Controller generates interrupt to the host (pin-based, MSI or MSI-X)
8. Host processes the completion queue entry, including action on errors
9. Host writes CQ Head Doorbell register to indicate that the entry has been processed

NVMe Command Processing (Trace – Actual)

Teledyne LeCroy PETracer(TM) - PCI Express Protocol Analyzer - [C:\Users\Public\Documents\LeCroy\PETracer\Z3_drive_emulation_boot_and_play_video.pex]																		
File Setup Record Generate Report Search View Tools Window Help                                 																		
NVM	R→	2.5	x8	RequesterID	SQyTDBL	IO SQT QID = 3	Time Delta	Time Stamp										
116				000:00:0		0x0002	220.432 us	0013 . 641 768 552 s										
NVM	R←	2.5	x8	RequesterID	CompleterID	IO Cmd	OPC	FUSE	CID	NSID	MPTR Hi	MPTR Low	PRP1 Hi	PRP1 Low	PRP2 Hi	PRP2 Low		
117				001:00:0		Read	b00	0x0001	0x00000001	0x000000001	0x000000000	0x000000000	0x000000002	0x1C940000	0x00000002	0x1C941000		
					SLBA	LR	FUA	PRINFO	NLB	DSM	Incompressible	SR	AL	AF	EILBRT	ELBAT	Time Delta	Time Stamp
					0x00000000:00000080	0	0	0x0	0x000F		0	0	None	None	0x000000000	0x00000	1.772 ms	0013 . 641 988 984 s
NVM	R←	2.5	x8	RequesterID	CMD PRP	Addr Hi	Addr Lo	Data Len	Data		Time Delta	Time Stamp						
118				001:00:0		0x00000002	0x1C940000	0x00001000	1024 quadlets		48.888 us	0013 . 643 760 568 s						
NVM	R←	2.5	x8	RequesterID	CMD PRP	Addr Hi	Addr Lo	Data Len	Data		Time Delta	Time Stamp						
119				001:00:0		0x00000002	0x1C941000	0x00001000	1024 quadlets		785.904 us	0013 . 643 809 456 s						
NVM	R←	2.5	x8	RequesterID	Command Completion	SQHD	SQID	CID	P	ST	SC	SCT	M	DNR	Time Delta	Time Stamp		
120				001:00:0	0x00000000	0x0002	0x0003	0x0001	1	0x0000	0x000	0	0	0	24.352 us	0013 . 644 595 360 s		
Link Tra	R←	2.5	x8	TLP	Mem	MWr(32)	Length	RequesterID	Tag	Address	1st BE	Last BE	Data	VC ID	Explicit ACK	Metrics	# Packets	
1515				800		010:00000	1	001:00:0	0	FEE0F00C	1111	0000	1 dword	0	Packet #9109		2	
					Time Delta	Time Stamp												
					4.768 us	0013 . 644 619 712 s												
NVM	R→	2.5	x8	RequesterID	CQyHDBL	IO CQH QID = 3	Time Delta	Time Stamp										
121				000:00:0		0x0002	10.068 ms	0013 . 644 624 480 s										

NVM I/O Commands

Read – Transfers indicated LBs from controller to host

Write – Transfers indicated LBs from host to controller

Write Uncorrectable – Marks LB as invalid

Write Zeros – Set a range of LBs to zero

Compare – Controller compares data from indicated LBs to buffer

Flush – Host instructs controller to move data from volatile to non-volatile storage

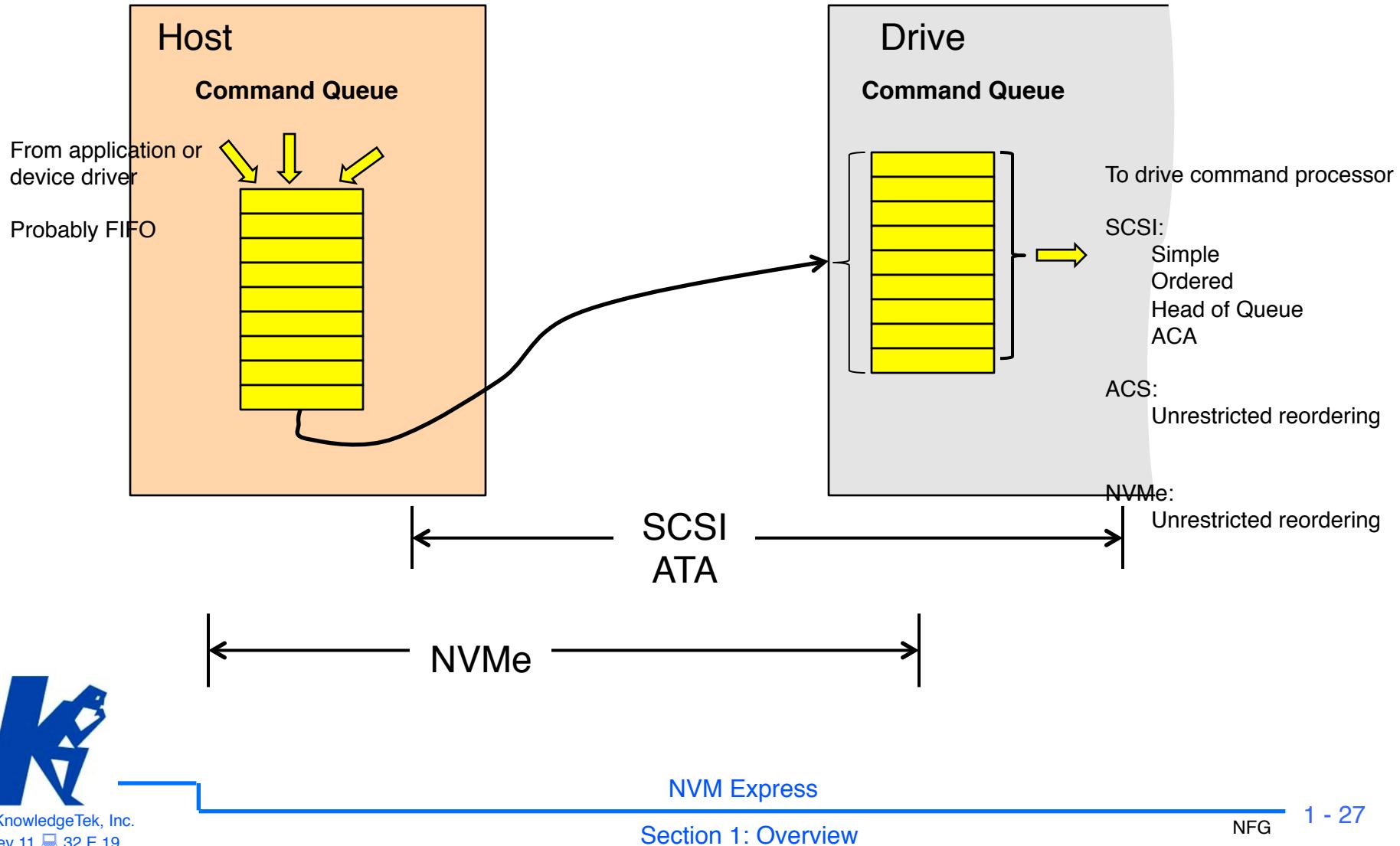
Dataset Management – Advisory command; used by host to tell controller

about data so controller can optimize performance and reliability.

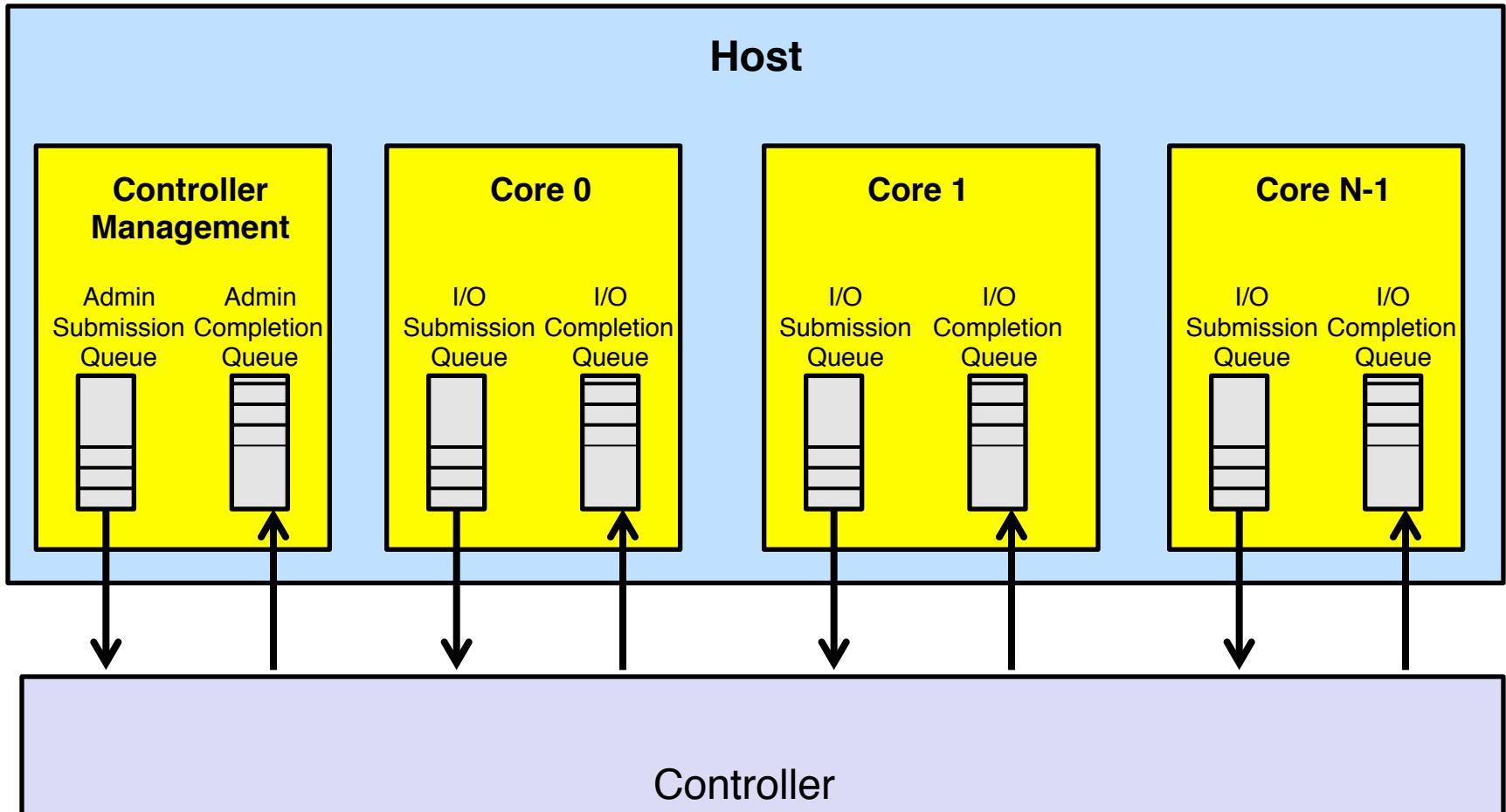
Reservations – Used by host to lock out other hosts (initialization, error recovery, rebuild, special operations)

Queues

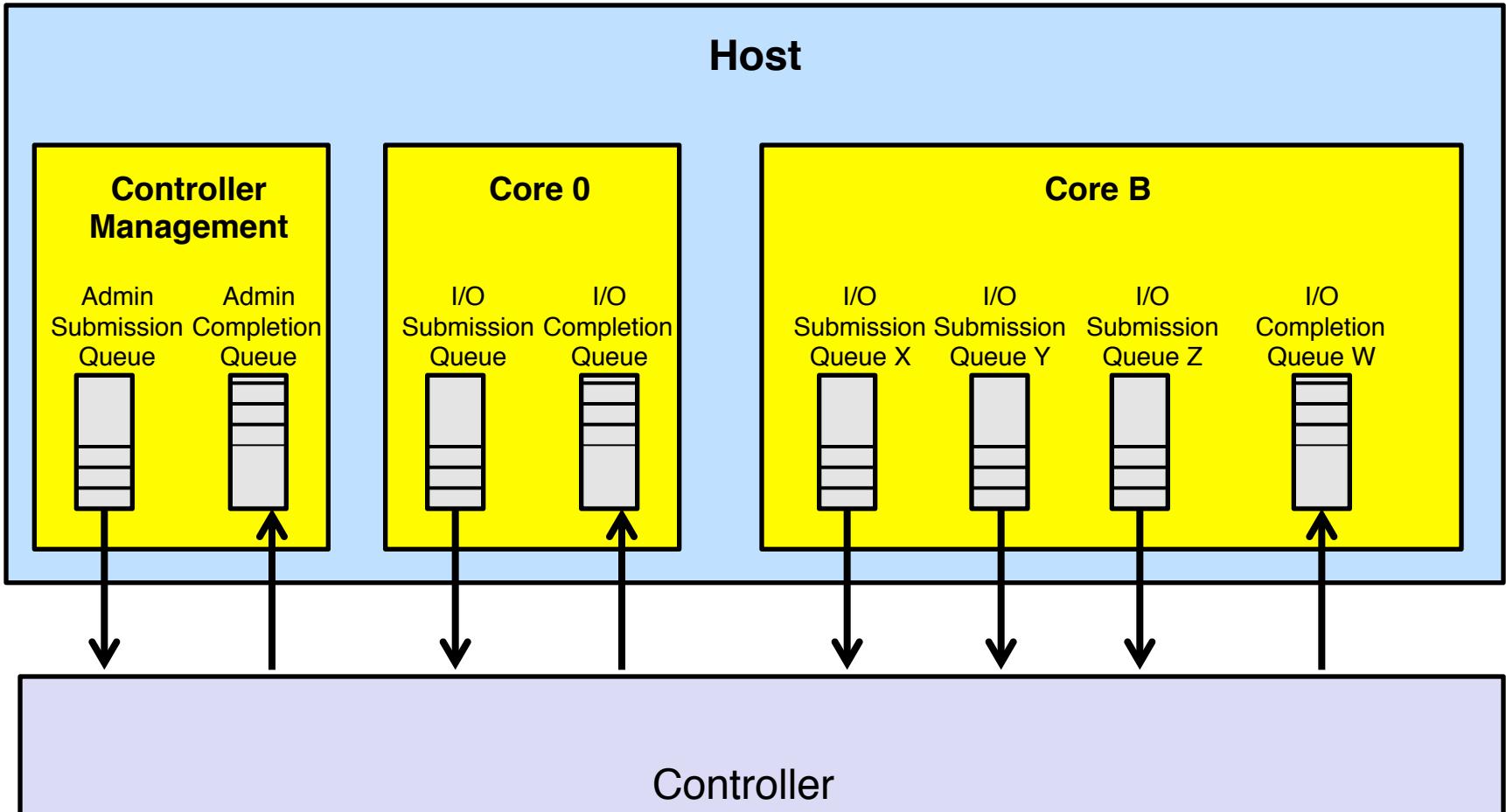
Storage Queues



Queue Pair 1:1 Example



Queue Pair N:1 Example



Each Submission queue is “Paired” with a completion queue
Each Completion queue can be “Paired” with multiple SQ
NVM Express

Queue Notes

Command Ordering

Except for fused operations

there are no ordering restrictions for processing of the commands
within or across Submission Queues.

Priority

Submission Queues have a priority; higher priority commands should
be submitted to higher priority queues.

Defined priority schemes are round robin and weighted round robin

Status Ordering

There are no ordering restrictions for presenting command status



Queues

Definition: a circular buffer with a fixed slot size used to communicate commands from the host to the controller, or command completions from the controller to the host.

Types:

Admin: a submission and a completion queue with identifier 0, used to manage the controller.

I/O: submission and completions queues with identifiers > 0, used to communicate commands and status

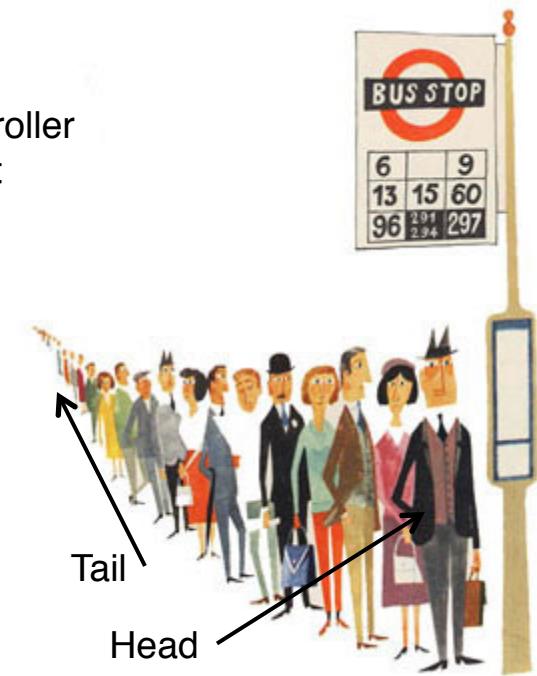
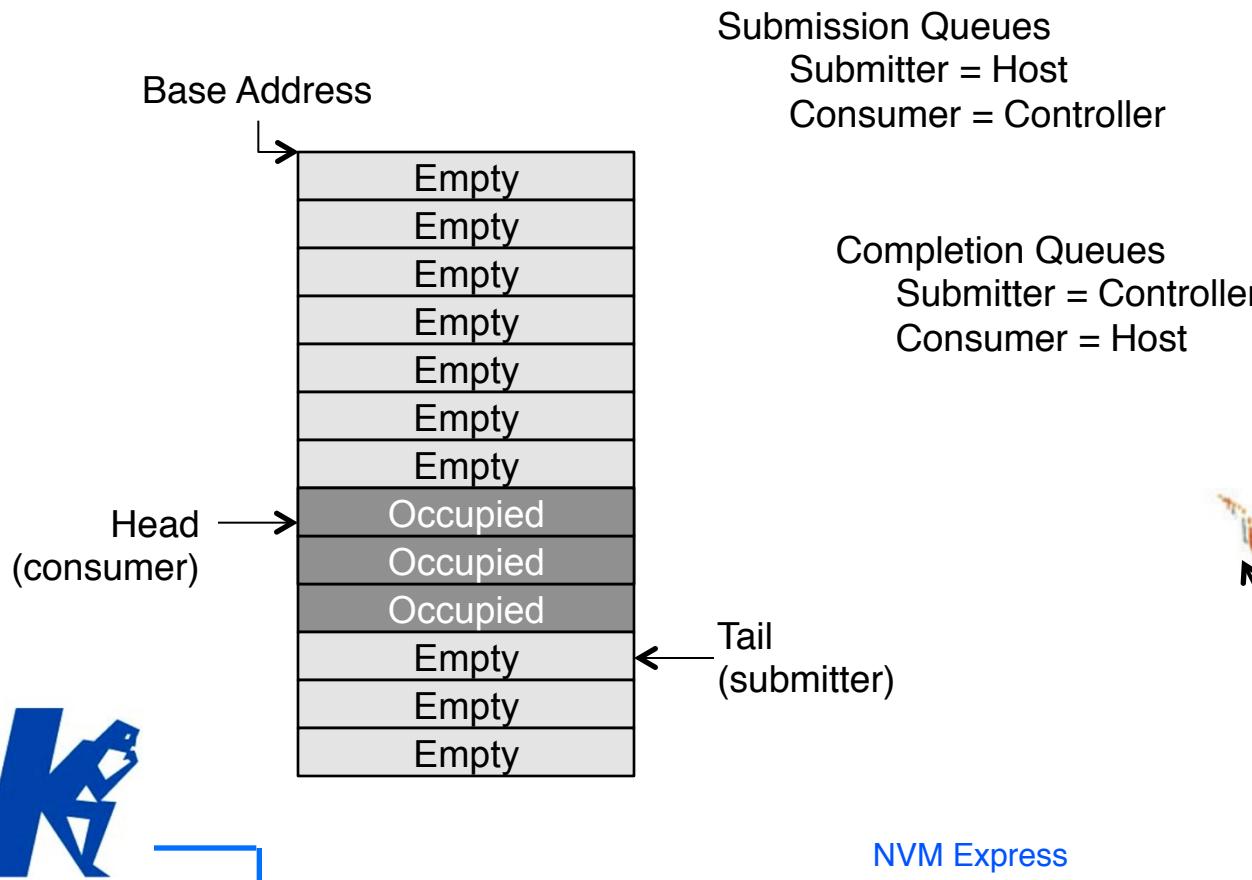
Submission: Used to send commands from the host to the controller.

Completion: Used by the controller to respond to commands upon completion.

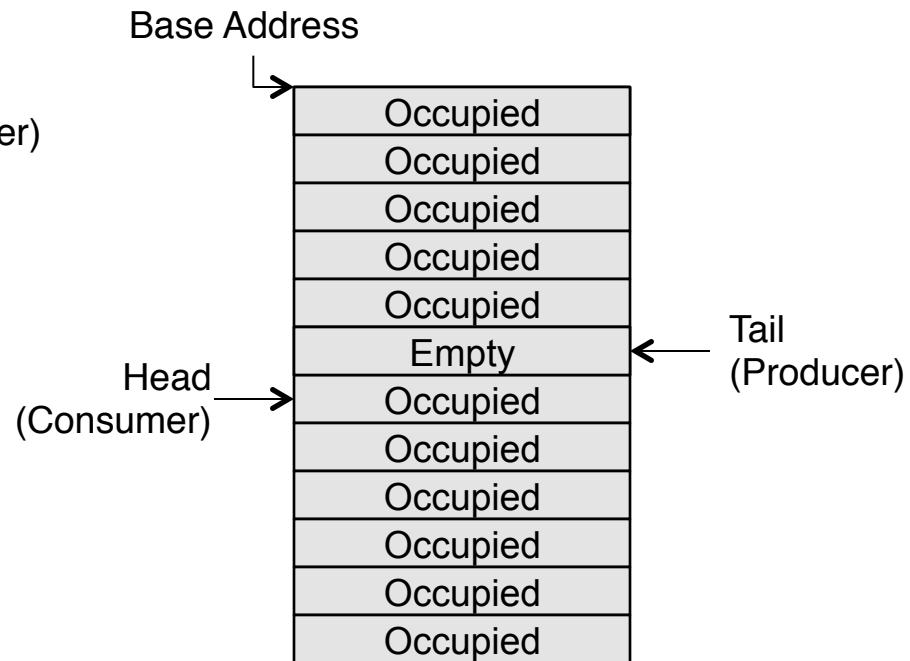
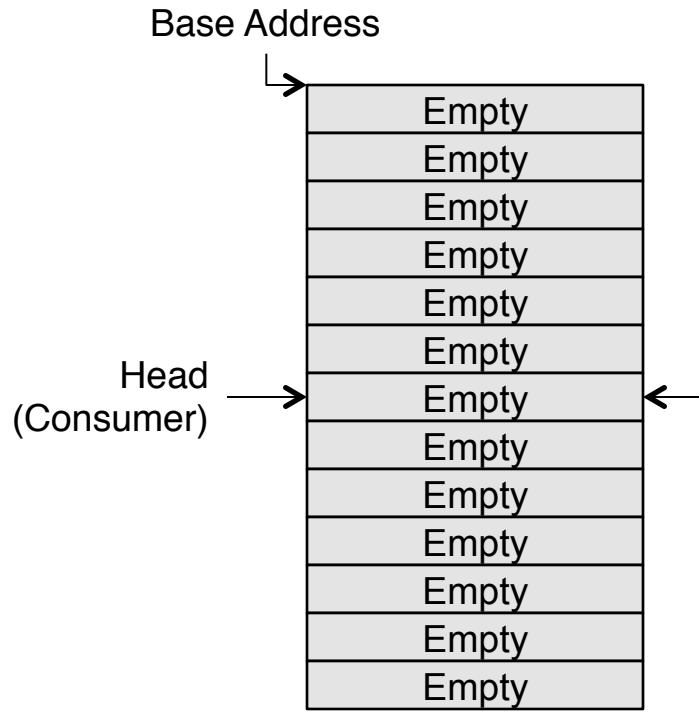
Queue Pointers

Submitter uses tail pointer to identify the next open queue entry space

Consumer uses head pointer to identify next entry to be pulled off the queue



Queue Empty and Queue Full

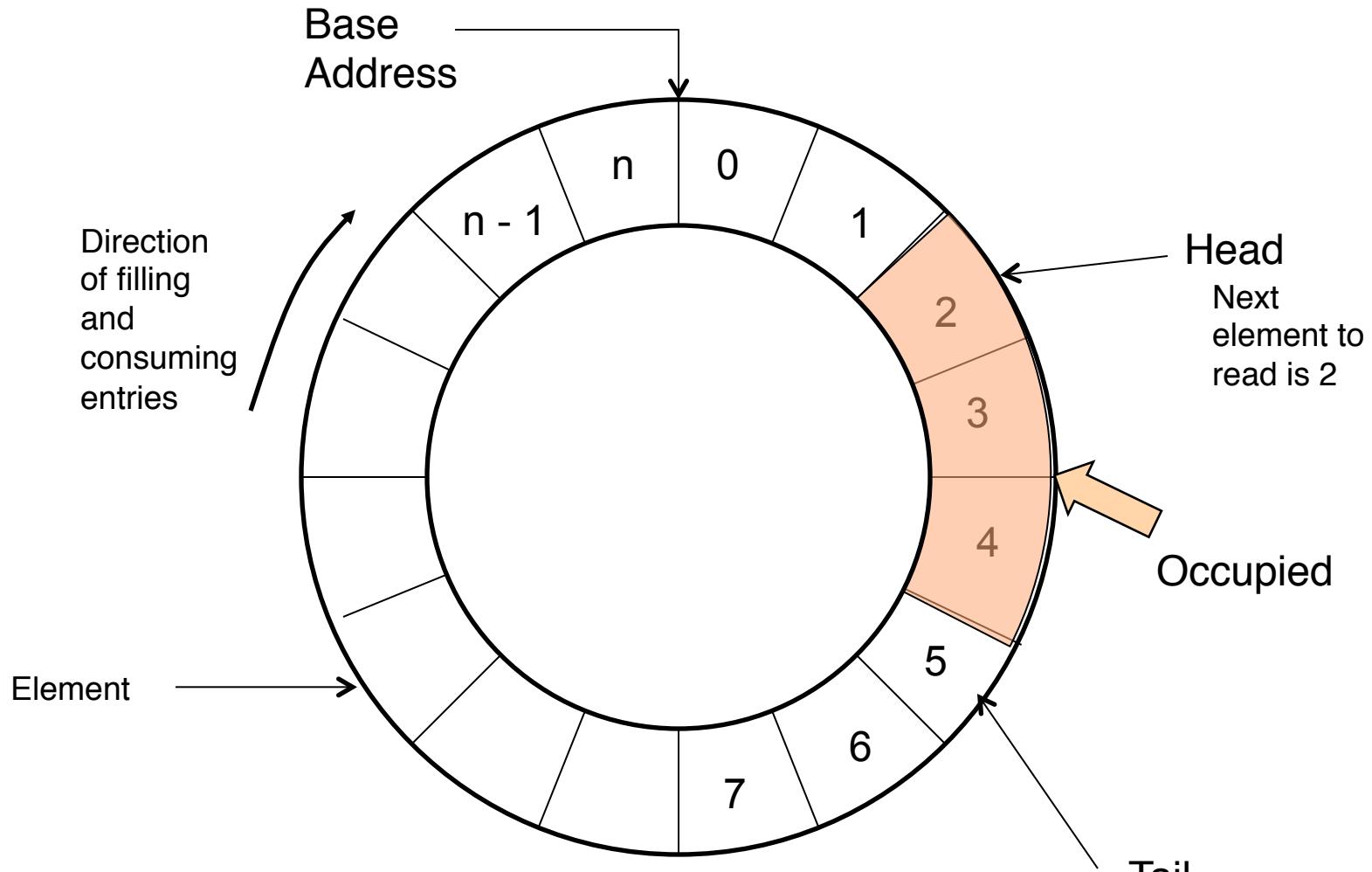


Queue Empty: Head pointer = tail pointer



Queue Full: Head pointer = tail pointer + 1

Queues in Operation



It is allowed (encouraged) to add more than one entry at a time.
It is allowed (encouraged) to read more than one entry at a time.

NVM Express

Passing NVMe Queue Pointers

Queue	Device	Pointer	Need to Know	Information Maintained or Passed
Submission	Host	Head	Is Q full?	Passed in Completion queue entry
		Tail	Add entry	Information maintained by host
	Controller	Head	Fetch Command entry	Information maintained by controller
		Tail	There more entries	Passed in the SQ Tail Doorbell
Completion	Host	Head	Fetch Status entry	Information maintained by host
		Tail	There are more entries	Information not needed by host. Host uses P bit to discover last entry
	Controller	Head	Is Q full?	Passed in CQ doorbell
		Tail	Add entry	Information maintained by controller

Queue Number and Size

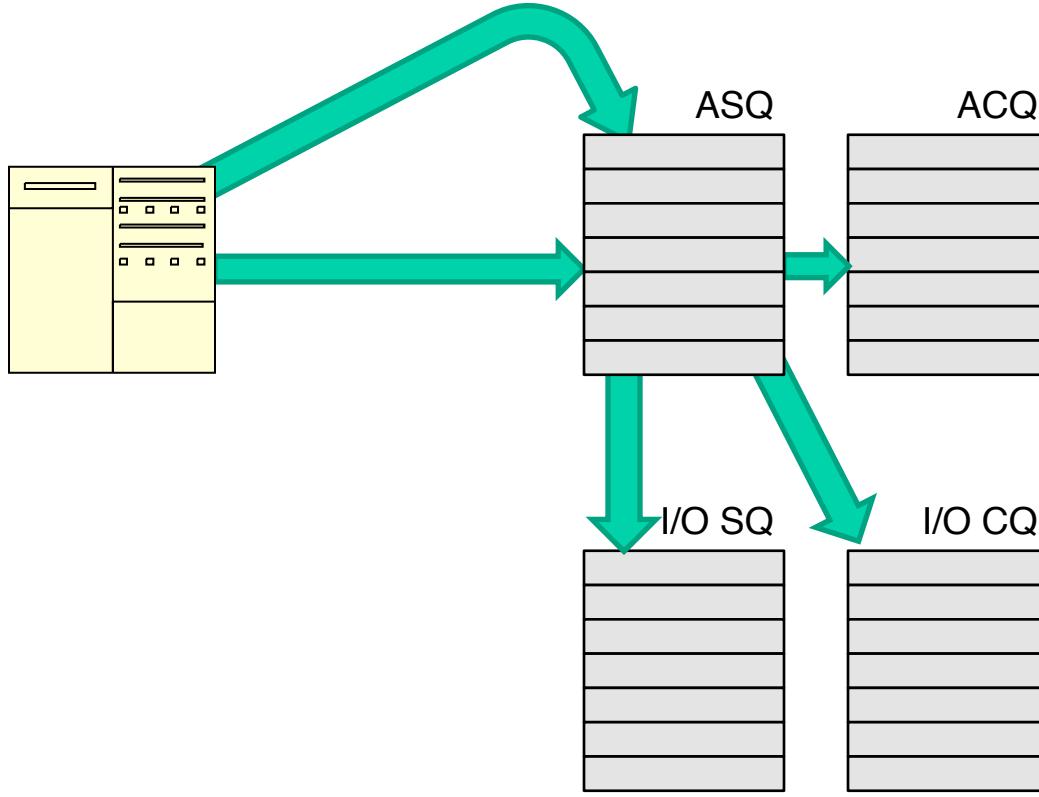
		Admin		I/O	
		Submission	Completion	Submission	Completion
Architecture Limit	Number of Queues	1	1	16 bits for Q ID = 64K	
	Queue Size	N/A	N/A	16 bits = 64K	
Defined Number of Entries	Maximum	4K	4K	64K	64K
	Minimum	2 entries			
Implementation Limit		4K	4K	Reported in CAP.MQES register field	
Actual Implementation		Configured in Admin Queue Attributes Register		Configured in Dword 10 of Create I/O SQ/CQ Admin Commands	
Entry Size		64 bytes	16 bytes	64 bytes	16 bytes

Queue Size = number of entries

Entry Size = number of bytes per entry

Defined Number of Entries – From the NVM Express specification

I/O Queue Creation Sequence



Step 1

Host creates Admin Queues by writing to registers

Step 2

Host creates I/O CQ by issuing Admin Command

Step 3

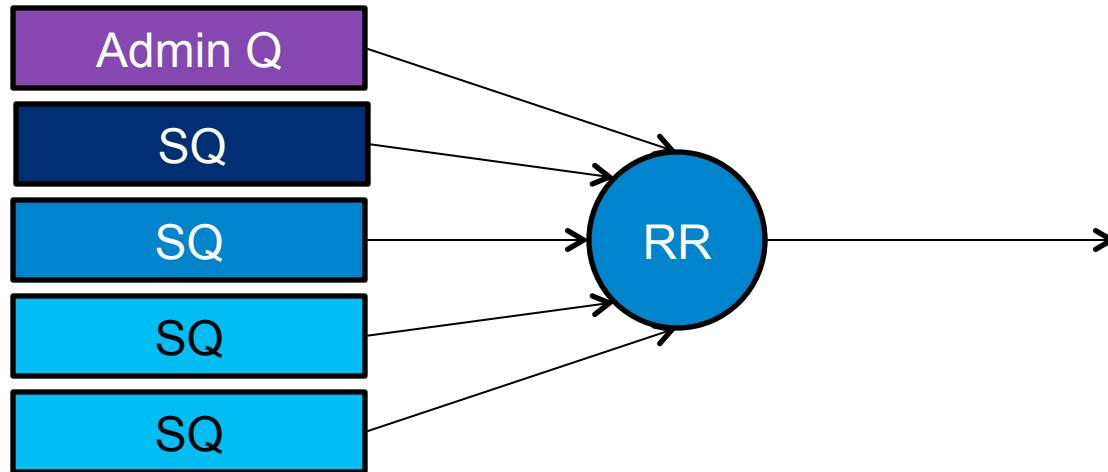
Host creates I/O SQ by issuing Admin Command and pairing it to an existing I/O CQ

I/O Queue Deletion Sequence

Step 1: Host issues Delete I/O SQ. Queue should be empty before deletion.

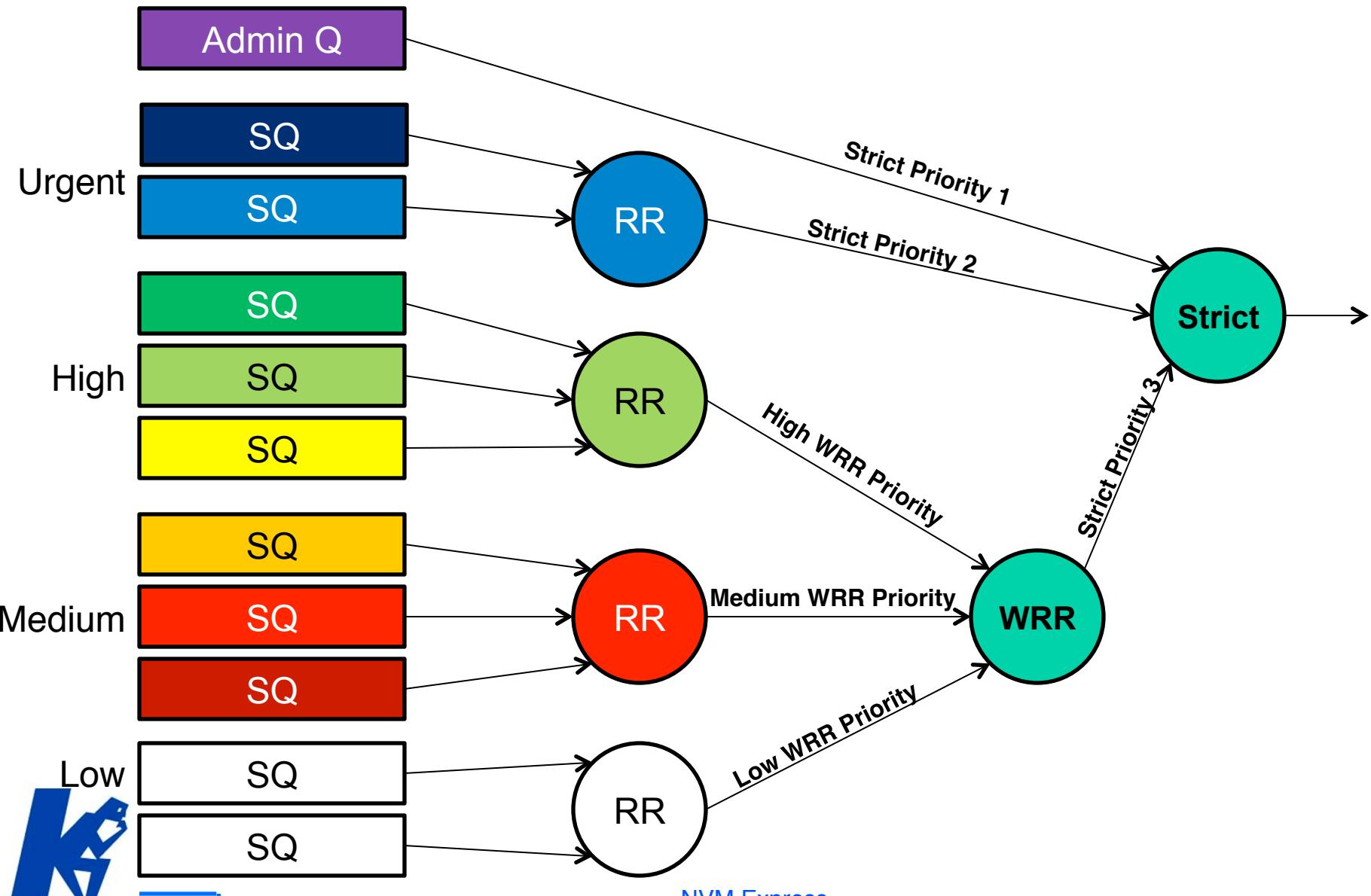
Step 2: Host issues Delete I/O CQ. All paired SQ must be deleted first

Round Robin Command Arbitration



All queues, including Admin, have same priority

Weighted Round Robin Command Arbitration



Doorbells

Doorbell

Definition: a 32-bit register written by the host to notify the Function that a queue has been updated.

Operation: Command Submission - The host will write the new commands into the appropriate submission queue in host memory space, then write the SQ Tail value in the controller to notify it of the new commands available.

Operation: Command Completion - The controller will write the command completion information into the appropriate completion queue in host memory and notify the host via interrupt. The host will update the CQ Head pointer after it has processed all of the completions.

SQ Tail doorbell register – one per submission queue; Indicates the new value of the Submission Queue Tail entry pointer.

CQ Head doorbell register – one per completion queue; Indicates the new value of the Completion Queue Head entry pointer.

Doorbell – pictures

Submission Queue

Empty
Empty
Occupied
Occupied
Occupied
Empty

Next command to be fetched by controller,
pointer maintained by controller

Tail Pointer to end of commands,
pointer communicated to controller by doorbell

Completion Queue

Empty
Occupied
Occupied
Occupied
Empty
Empty
Empty

Head Pointer. Host has processed all
completions to here.

Pointer communicated to controller by doorbell

Next completion status to be stored by controller,
pointer maintained by controller



Doorbell Stride

Used for S/W implementations of NVMe controller

Aligns Doorbell addresses for S/W access
e.g. one doorbell per cacheline

Bytes between doorbells is $(2^{(2 + \text{CAP.DSTRD})})$

For hardware implementation, DSTRD will usually equal 0h (one doorbell every 4 bytes; no bytes between doorbells)

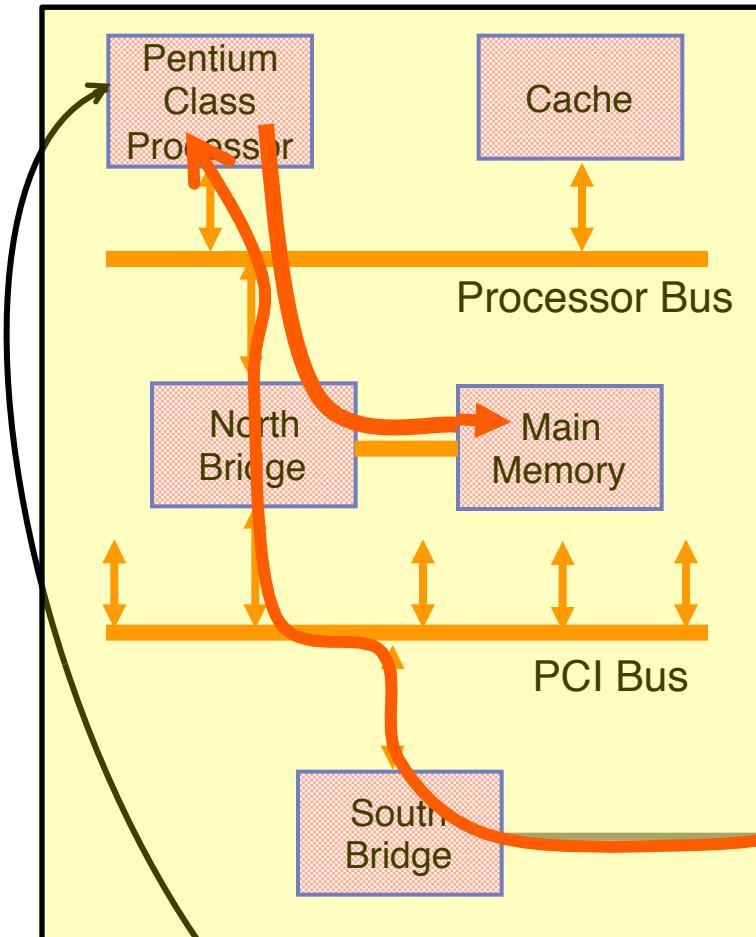


RDMA



PIO = Programmed I/O Read

Local Device



Local host issues command

remote address

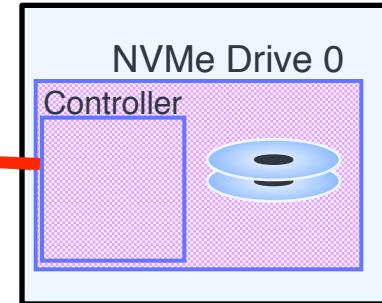
what to do (read)

how much

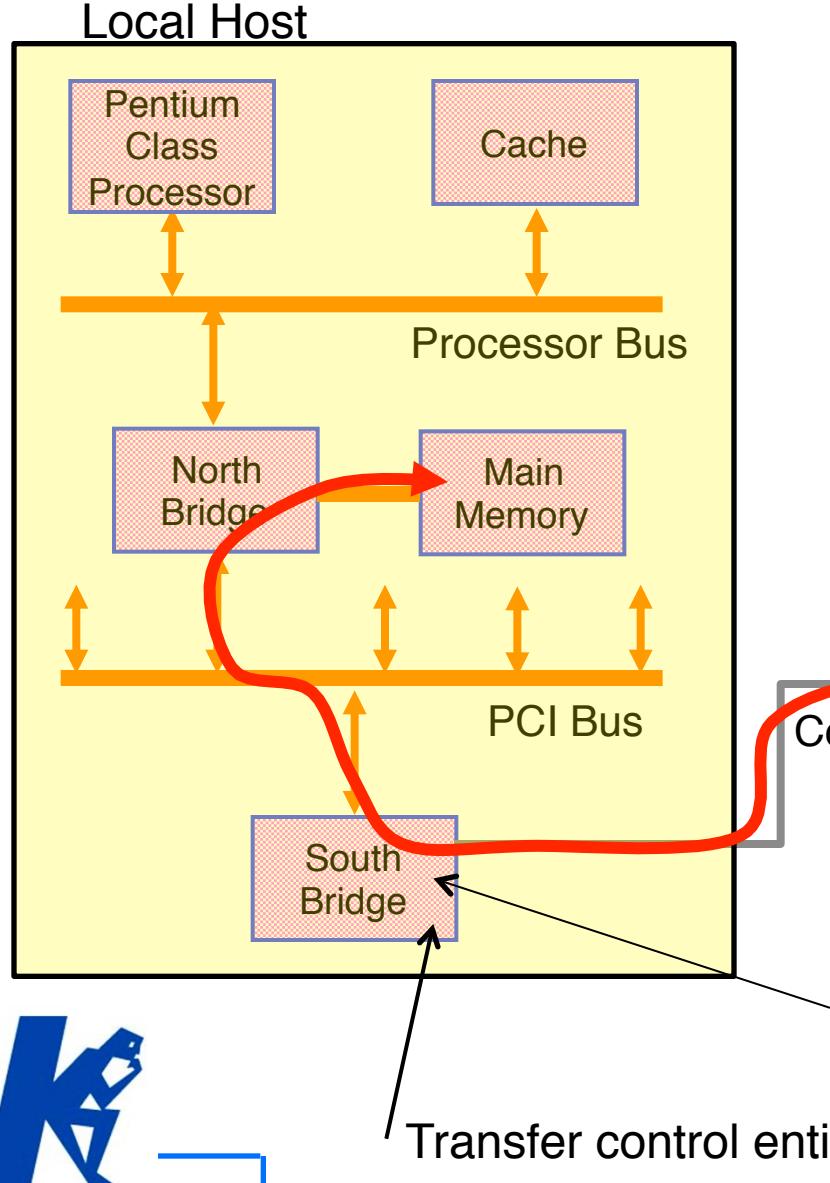
Local host fetches data

Local host stores data

Remote Device



DMA = Direct Memory Access Read



Local host sets up local DMA controller:
memory address

Local host issues command

remote address

what to do (read)

how much

Local DMA controller fetches data
Local DMA controller stores data

Remote Device

NVMe Drive 0

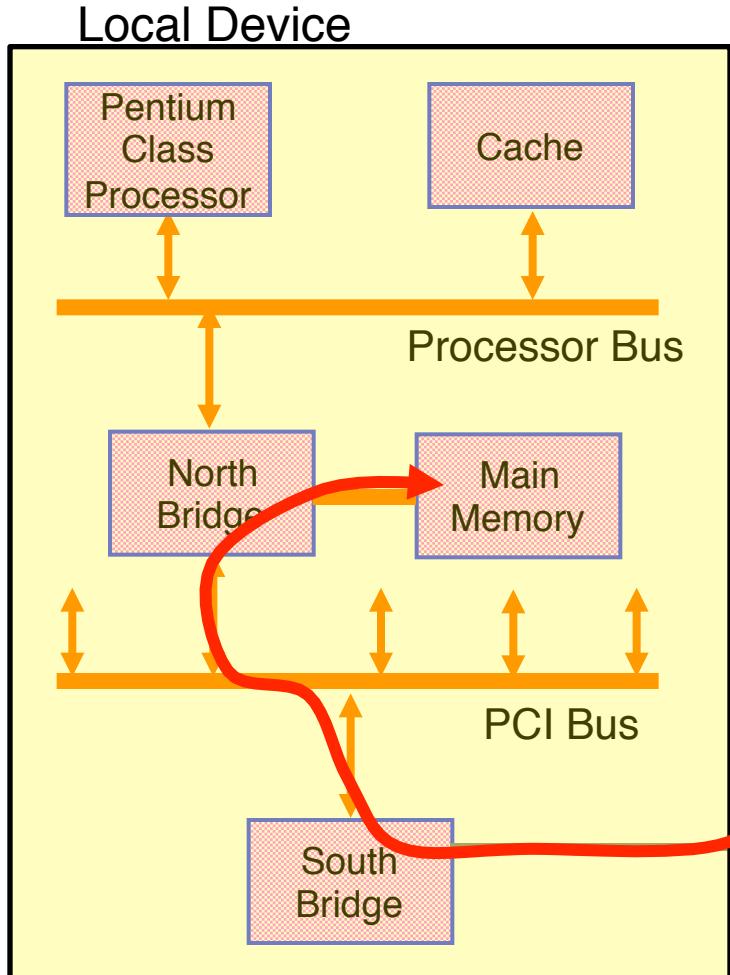
Controller



Communication Media

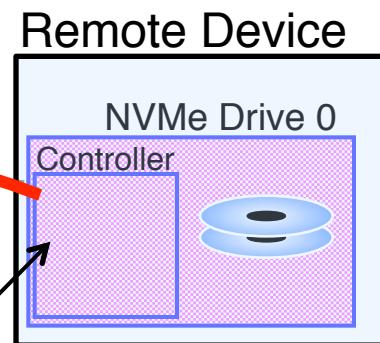
Local DMA controller is in host
Probably South Bridge or chip set

RDMA = Remote Direct Memory Access Read



Local host issues command:
local memory address
remote address
what to do (read)
how much

Remote controller sends data to local device memory using write transactions



Power States

Power States

Definition: Up to 32 definitions per controller specifying:
the maximum power consumed,
Entry and Exit latency, and the
performance relative to other power states

Operation:

Read controller power states with Identify

Transition immediately to Power State x with Set Features (02h)

Set time to enter each power state with Set Features (0Ch)

Details:

Power State details are covered in Admin Commands Section
Identify Controller Data Structure
Set/Get Features

Resets

Three types of Resets defined in NVMe specification

NVM Subsystem Reset

Controller Level Reset

Queue Level Reset



NVM Subsystem Reset

Initiated when:

- Power is applied to the NVM Subsystem,
- “NVMe” is written to the NSSR.NSSRC register field, or
- A vendor specific event occurs.

What it does:

- Entire subsystem is reset,
- Controller Level Reset on all controllers in subsystem
- Transition to the Detect LTSSM state for all PCIe ports



Controller Level Reset

Initiated when:

- NVM Subsystem Reset,
- Conventional Reset (PCIe Hot, Warm or Cold reset),
- PCIe Data Link Down status,
- Function Level Reset (PCI reset), or
- Controller Reset (CC.EN transition from 1 to 0)

What it does:

- The controller stops processing Admin and I/O commands,
- All I/O SQ and CQ are deleted,
- Controller goes to idle state, CSTS.RDY is cleared to 0,
- All controller registers go to Reset value except AQA, ASQ and ACQ.



Queue Level Reset

Initiated when:

Delete and re-create an I/O SQ and/or CQ

What it does:

Host should ensure queues are empty before deleting them,
Allows host to change attributes of queue.



Covered in this Section

What is NVMe

Overview of how NVMe works

Queues

Doorbell

RDMA

Power States

Resets



Notes



Notes



Section 2

PCIe for NVMe

Overview



Covered in this Section

PCI/PCIe Concepts

Topology Discovery and Enumeration

PCI Transactions

PCIe Link Layer

PCIe Physical Layer

Flow Control

ACK/NAK protocol

PCI Versions

PCI

Parallel bus



PCIe

Bit serial per lane
Point to point with switches
Byte parallel
Up to 32 lanes per link
Dual simplex
Differential signaling



PCI-X

PCI - eXtended

Parallel bus

Primarily for high end workstations or servers

Up to 533 * 64bit transfers

ECC on header and data

Completely backward compatible



PCIe Speeds

Aggregate GB per second

	GT/s	Link Width						
		X1	X2	X4	X8	X12	X16	X32
Gen 1	2.5	0.5	1	2	4	6	8	16
Gen 2	5.0	1	2	4	8	12	16	32
Gen 3	8.0	2	4	8	16	24	32	64
Gen 4	16.0	4	8	16	32	48	64	128

Formula:

$$\frac{\text{GT/s} * \text{width} * 2 \text{ directions}}{\text{bits per byte/character}} = \text{GB per second}$$

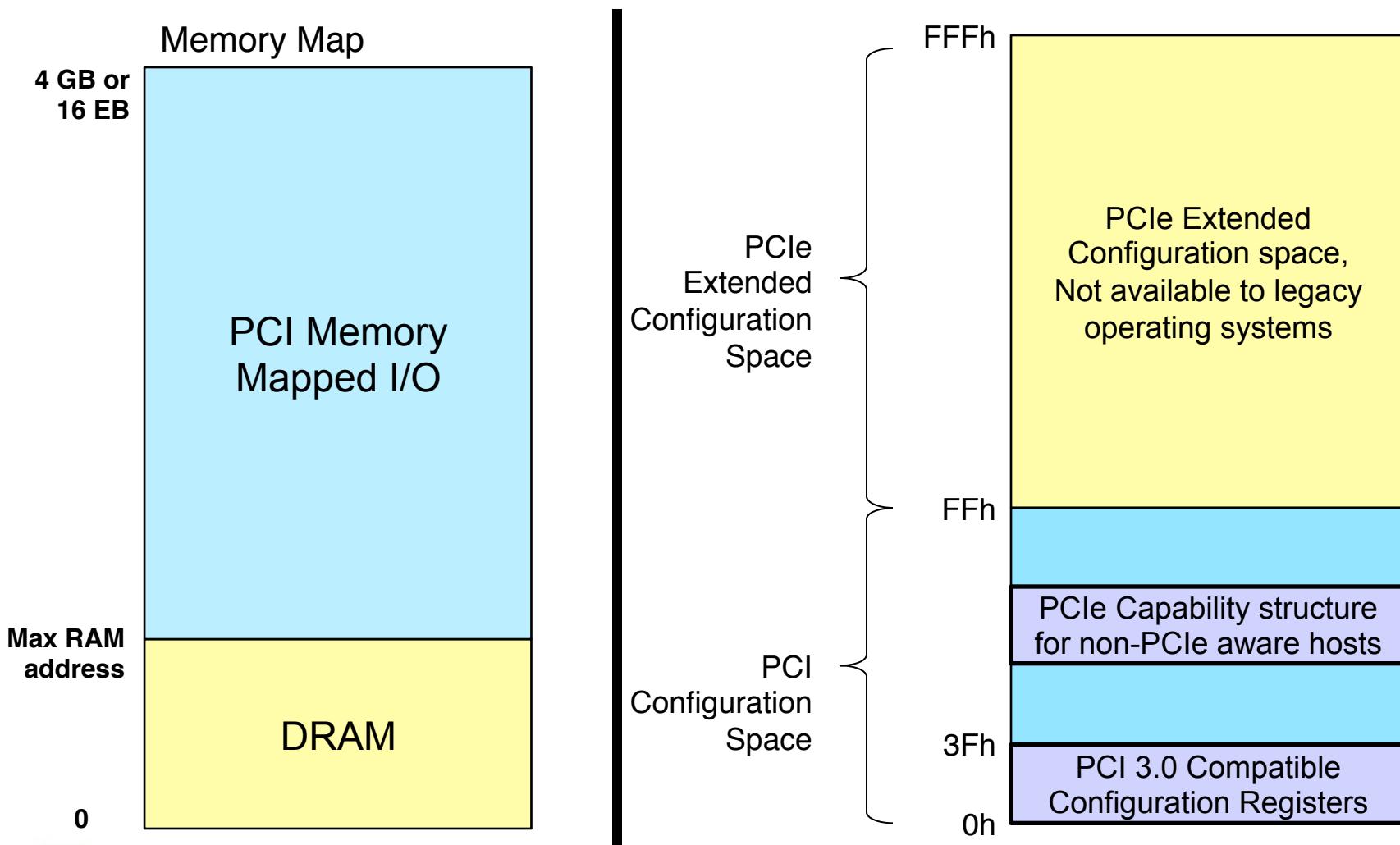
This chart includes the 8b/10b encoding of Gen 1 and Gen 2

It does not take into account the 128b/130b encoding of Gen 3 & 4 so their speeds are 1.5625% overstated.

This chart shows signaling speed, not overhead

PCIe Configuration

Memory Space and Configuration Mapping



Caution: PCIe Specification shows lowest address at bottom of picture

PCI Configuration Header for NVMe

Identification

Type 0 Configuration Space Header (Endpoint)	Byte
Device ID	0
Status	4
Class Code	8
BIST	C
Header Type = 0h	
Master Latency Timer	
Cache Line Size	
BAR0 – MLBAR – NVMe Registers	10
BAR1 – MUBAR – NVMe Registers	14
BAR2 – I/O based accesses, if supported	18
BAR3 - Reserved	1C
BAR4 – Vendor Specific	20
BAR5 – Vendor Specific	24
Cardbus CIS Pointer	28
Subsystem ID	2C
Subsystem Vendor ID	
Expansion ROM Base Address	30
Reserved	
Capabilities Pointer	
Reserved	38
Max Latency = 00h	3C
Min Grant = 00h	
Interrupt Pin	
Interrupt Line	

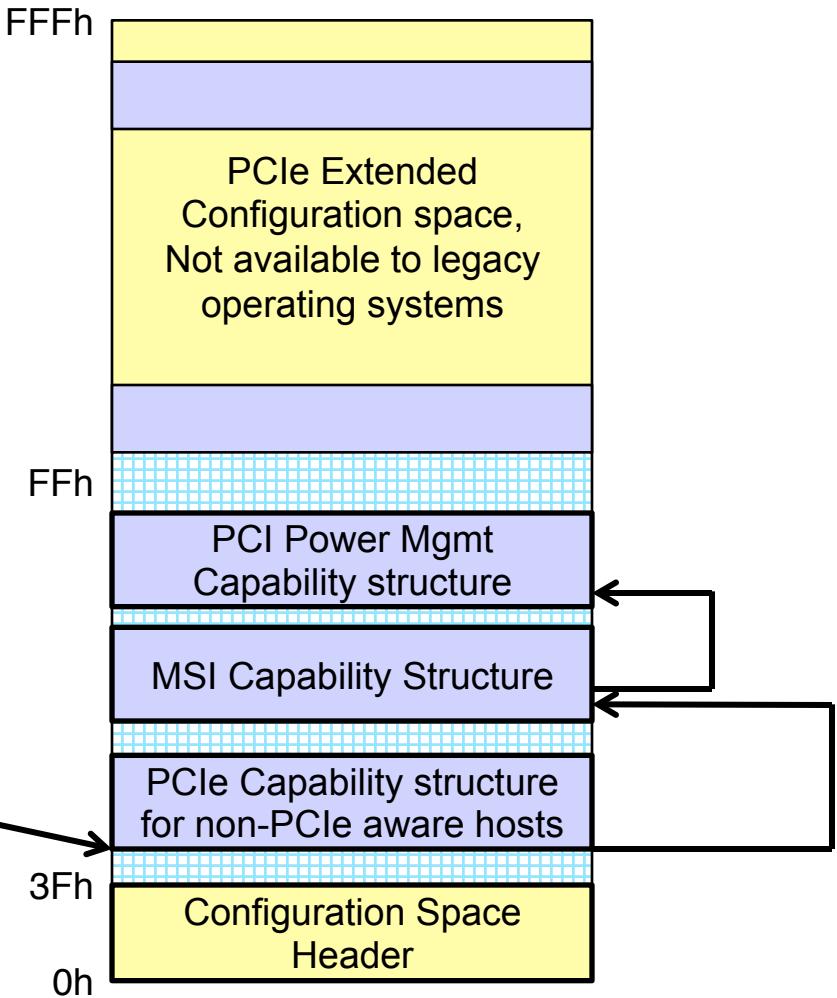
Capabilities



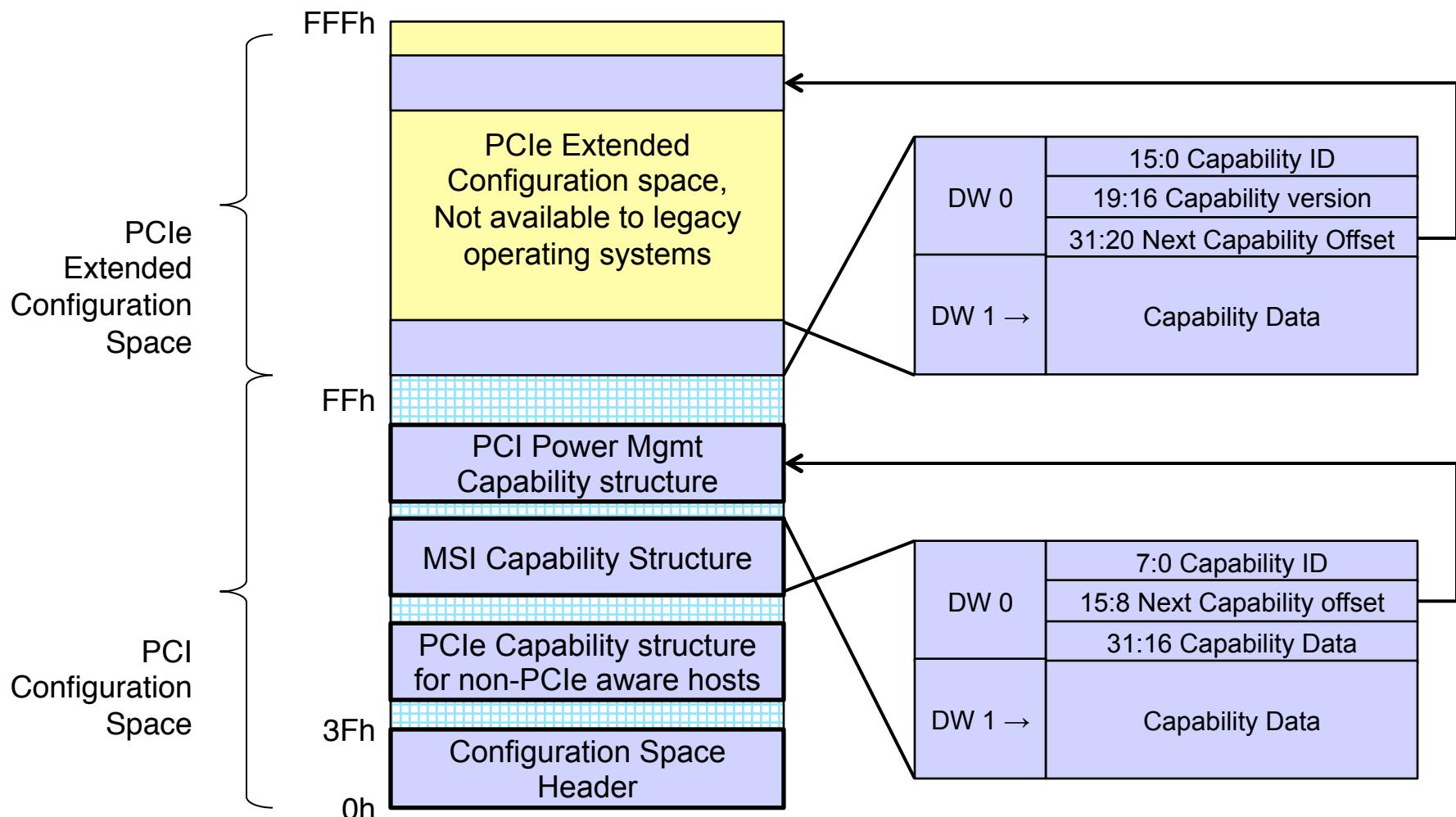
Caution: PCIe Specification shows lowest address at top of picture

Capability Link List

Device ID	Vendor ID		
Status	Command		
Class Code		Revision ID	
BIST	Header Type = 0h	Master Latency Timer	Cache Line Size
BAR0			
BAR1			
BAR2			
BAR3			
BAR4			
BAR5			
Cardbus CIS Pointer			
Subsystem ID	Subsystem Vendor ID		
Expansion ROM Base Address			
Reserved		Capabilities Pointer	
Reserved			
Max Latency = 00h	Min Grant = 00h	Interrupt Pin	Interrupt Line



PCIe Configuration Space



Caution: PCIe Specification shows lowest address at bottom of picture on left and top of picture on right

Some PCI Capability Registers

Capability ID	PCIe Name	NVMe Name	Name
01h	Power Mgmt Capability	PMCAP Section 2.2	PCI Power Management Capability
05h	MSI Capability LB3.0 Section 6.8.1	MSICAP Section 2.3	MSI Capability
11h	MSI-X Capability LB 3.0 Section 6.8.2	MSIXCAP Section 2.4	MSI-X Capability
10h	PCI Express PCI Express Base 11 Section 7.89	PXCAP Section 2.5	PCI Express Capability

PCIe Capability Structure

	31	23	15	7	0			
					00h			
Device	PCIe Capabilities Register		Next Cap pointer	PCIe Cap ID = 10	04h			
	Device Capabilities							
Link	Device Status		Device Control					
	Link Capabilities							
Slot	Link Status		Link Control					
	Slot Capabilities							
Root	Slot Status		Slot Control					
	Root Capabilities							
Device 2	Root Control							
	Root Status							
Link 2	Device Capabilities 2							
	Device Status 2		Device Control 2					
Slot 2	Link Capabilities 2							
	Link Status 2		Link Control 2					
	Slot Capabilities 2							
	Slot Status 2		Slot Control 2					



Caution: PCIe Specification shows lowest address at top of picture

PCIe Extended Configuration Capability – Part 1

Capability ID	Name
0001h	Advanced Error Reporting
0002h	Virtual Channel Capability w/o multi-function virtual channel
0003h	Device Serial Number Capability
0004h	Power Budgeting Capability
0005h	Root Complex Link Declaration Capability
0006h	Root Complex Internal Link Control
0007h	Root Complex Event Collector Capability
0008h	Multi-Function Virtual Channel Capability
0009h	Virtual Channel Capability w multi-function virtual channel
000Ah	RCRB (Root Complex Register Block) Header Capability
000Bh	Vendor Specific Capability
000Ch	Correlation Access Capability
000Dh	Access Control Services Extended Capability(ACS)
000Eh	ARI (Alternative Routing-ID Interpretation Capability
000Fh	Address Translation Services (ATS)
0010h	SR-IOV
0011h	MR-IOV
0012h	Multicast Capability
0013h	ATS Page Request Interface (PRI)
0015h	Resizable BAR (Base Address Register) Capability

PCIe Extended Configuration Capability – Part 2

Capability ID	Name
0016h	Dynamic Power Allocation Capability
0017h	TPH (TLP Processing Hints) Requester Capability
0018h	Latency Tolerance Reporting Capability
0019h	Secondary PCIe Extended Capability
001Bh	PASID
001Ch	Lightweight Notification
001Dh	Downstream Port Containment
001Eh	L1 PM Substates
001Fh	Precision Time Management
0020h	M-PCIe Extended Capability
0021h	Function Readiness Status
0022h	Readiness Time Reporting

Addressing

TLP Addressing Modes

Memory Address

Used with Memory and I/O Requests

ID

Used with Configuration Requests, ID Routed Messages and Completions
Call Bus-Device-Function in PCI and PCI-X

Implicit

Used with Message Requests only
Routing type implies destination

A Few Details on BDF

PCI and PCI-X devices are addressed as Bus – Device - Function

Bus addresses are assigned by host during initialization

8 bits – 256 possible buses

Device addresses are assigned by host during initialization

5 bits – legacy

Always 00h in PCIe

0 bits – with ARI

Function addresses are assigned by manufacturer

3 bits – legacy, up to 8 Functions per device

8 bits – with ARI, up to 256 Functions

ARI – Alternative Routing-ID Interpretation

Applicable to Requester IDs, Completer IDs and Routing IDs

ARI – Alternative Routing-ID Interpretation

4 DW Header with ID Routing – without ARI

FMT	Type			
Bus Number	Device ID.	Function No.		

4 DW Header with ID Routing – with ARI

FMT	Type			
Bus Number	Function Number			

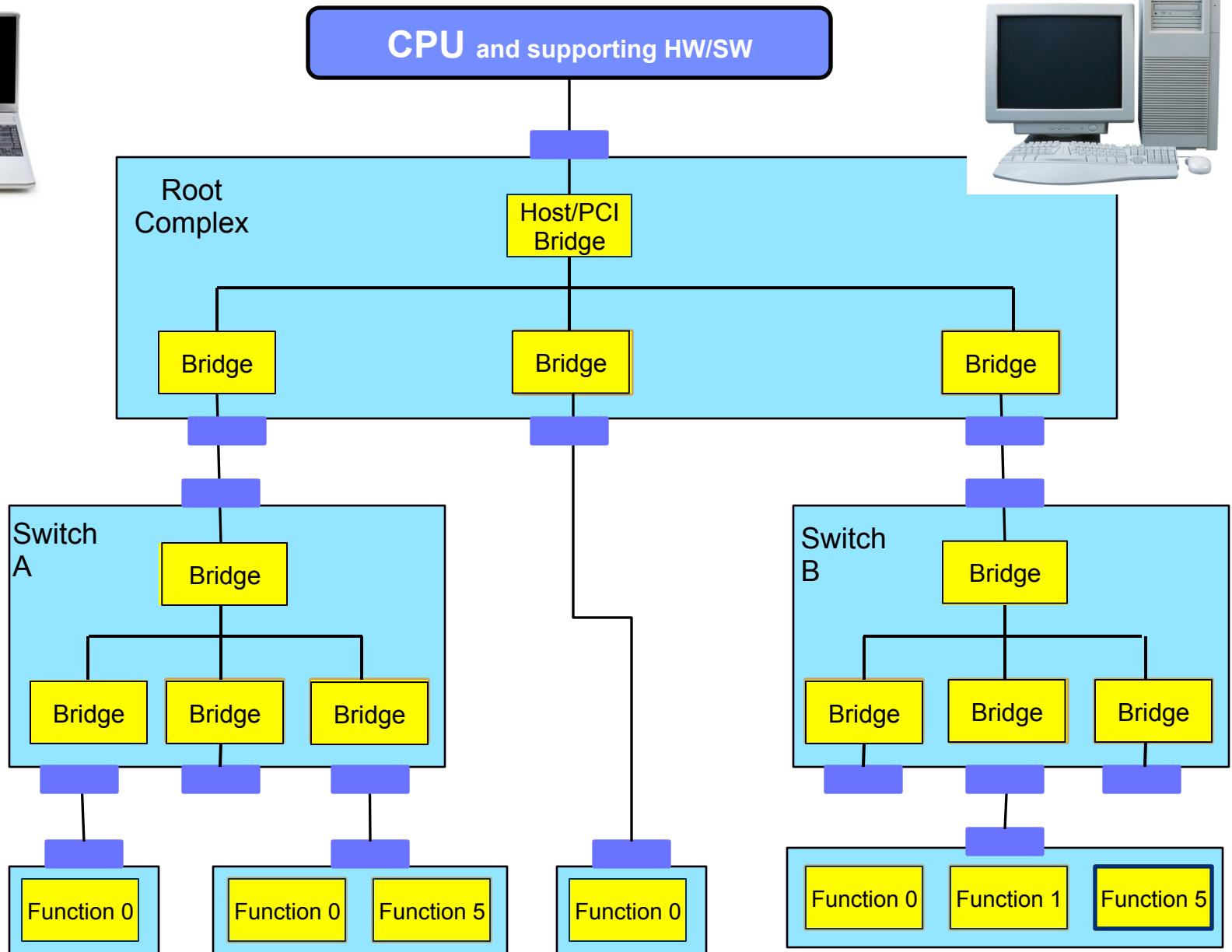
Addressing Mode – Address

	Byte + 0		Byte + 1		Byte + 2		Byte + 3	
0	FMT 0_x_1	Type	R	TC	R	^A ^T ^T ^R	R	T H D P
1	Fields dependent on FMT and Type							
2	64 bit address							
3	32 bit address							

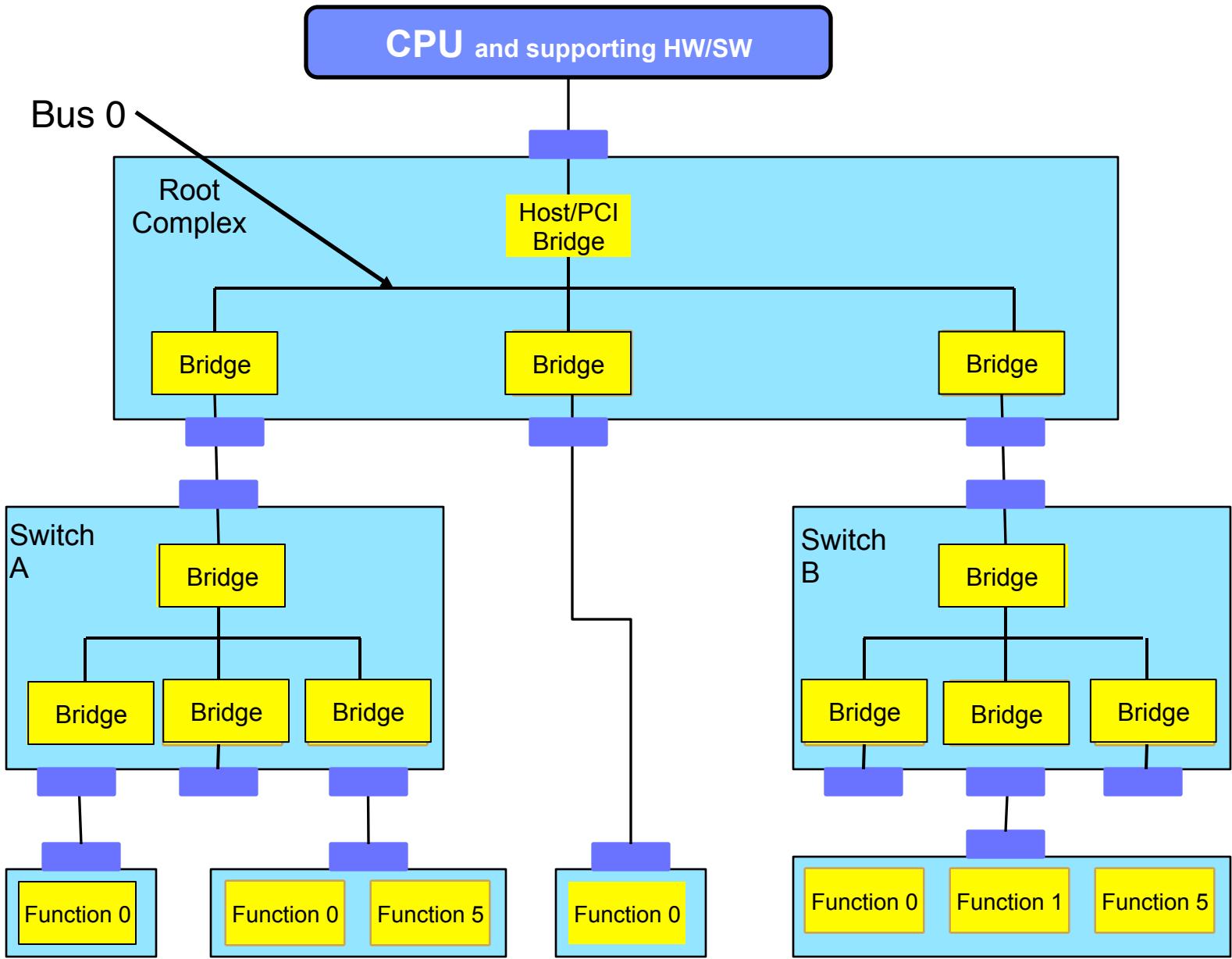
	Byte + 0		Byte + 1		Byte + 2		Byte + 3	
0	FMT 0_x_0	Type	R	TC	R	^A ^T ^T ^R	R	T H D P
1	Fields dependent on FMT and Type							
2	32 bit address							

Topology Discovery And Enumeration

PCIe Example Topology



Topology after Power On/Reset



Discovery Process – Too Much Detail

Set Bus variables Primary ID = Secondary ID = Subordinate ID = 0

Next Device: Host does Configuration Read to Primary ID, Device 0, Function 0

If valid Vendor ID, read header type

If 01h (bridge) goto Bridge

If 00h (end device) goto Next Bus

If Multi-Function bit is on, check Functions 1-7 or ARI Capability

Bridge: set variables:

Primary ID to Primary ID

Increment Secondary ID and Subordinate ID

Set Secondary ID in this bridge

Set Subordinate ID in all bridges between this and Primary ID = 0

Increment Primary ID

Goto Next Device

Next Bus: Set Primary ID = Primary ID + 1

Goto Next Device

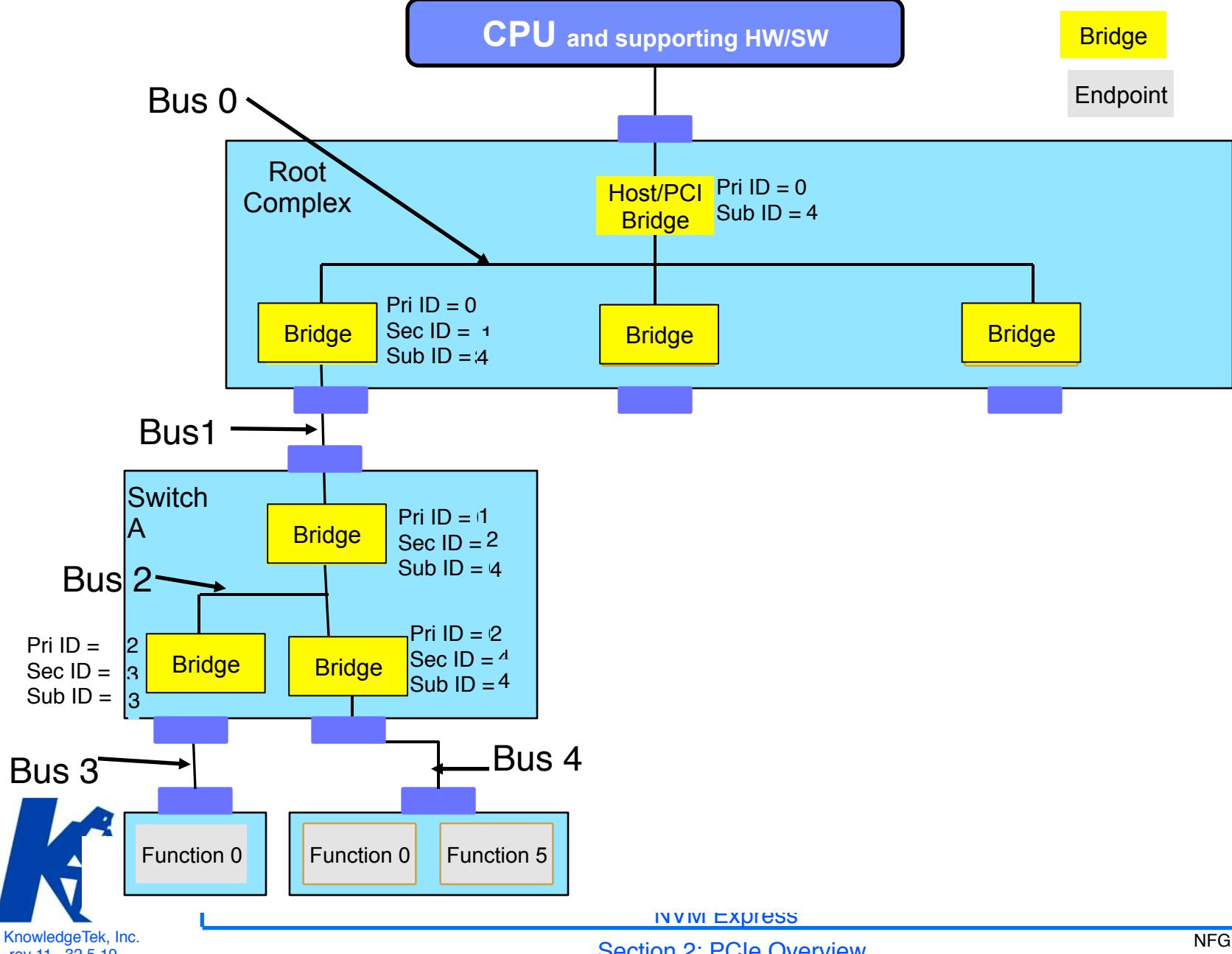
Go deep, then wide

Primary ID = Host side bus

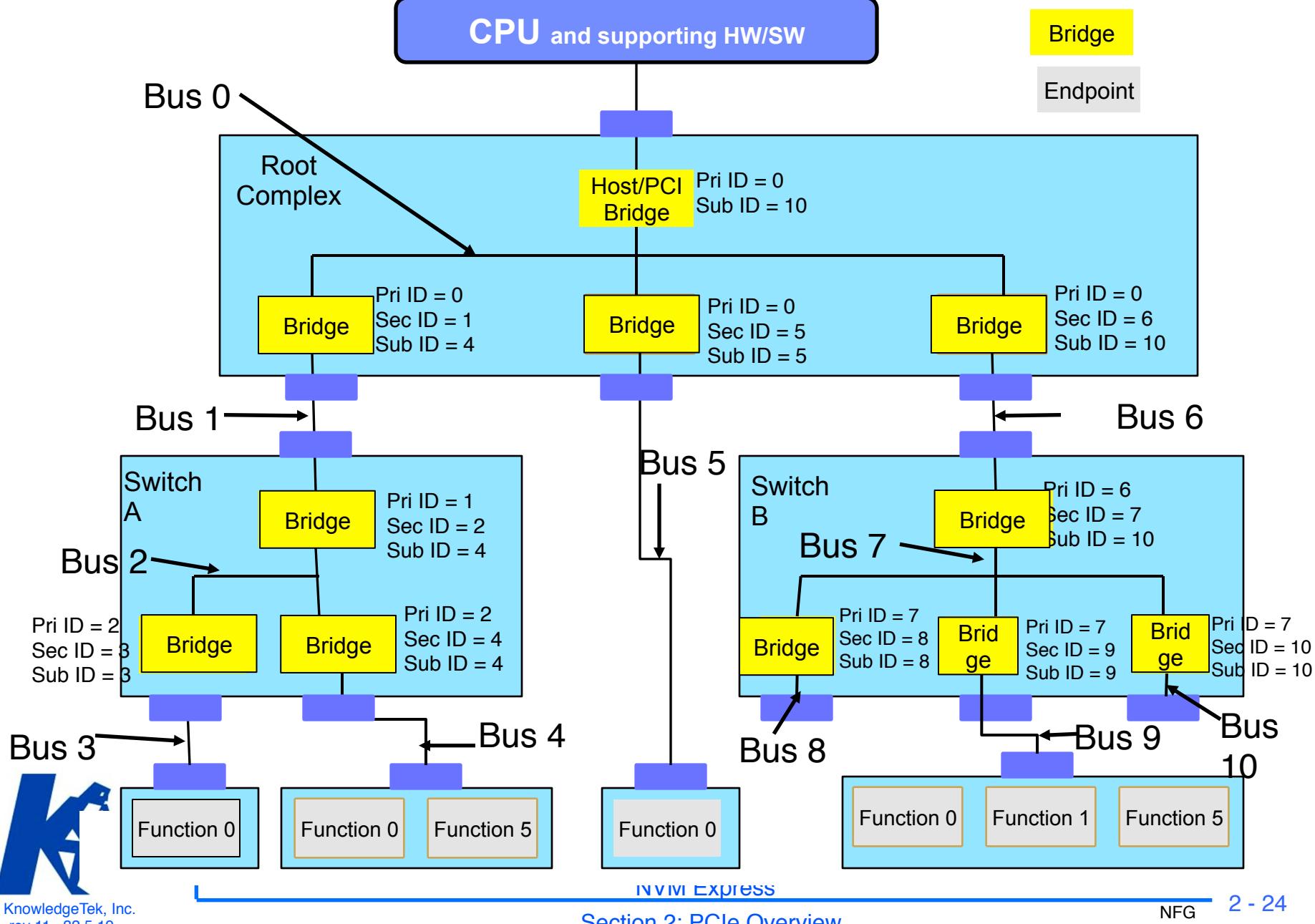
Secondary ID = Device side bus, first bus reached from this bridge

Sub ID = last bus reached from this bridge

Topology after Discovery and Enumeration



Topology after Discovery and Enumeration



Configuration Space Header

Byte (h) Type 1 Configuration Space Header (Bridge)

0	Device ID	Vendor ID			
4	Status	Command			
8	Class Code				
C	BIST	Header Type = 1h			
10	Master Latency Timer				
14	Cache Line Size				
18	BAR0				
20	BAR1				
24	Secondary Latency Timer	Subordinate Bus No	Secondary Bus No.		
28	Primary Bus No.				
32	Secondary status	I/O Limit	I/O Base		
36	Memory Limit	Memory Base			
40	Prefetchable Memory limit	Prefetchable Memory base			
44	Prefetchable Base Upper 32 bits				
48	Prefetchable Limit Upper 32 bits				
52	I/O Limit Upper 16 bits	I/O Base Upper 16 bits			
56	Reserved		Capabilities Pointer		
60	Expansion ROM Base Address				
64	Bridge Control	Interrupt Pin	Interrupt Line		

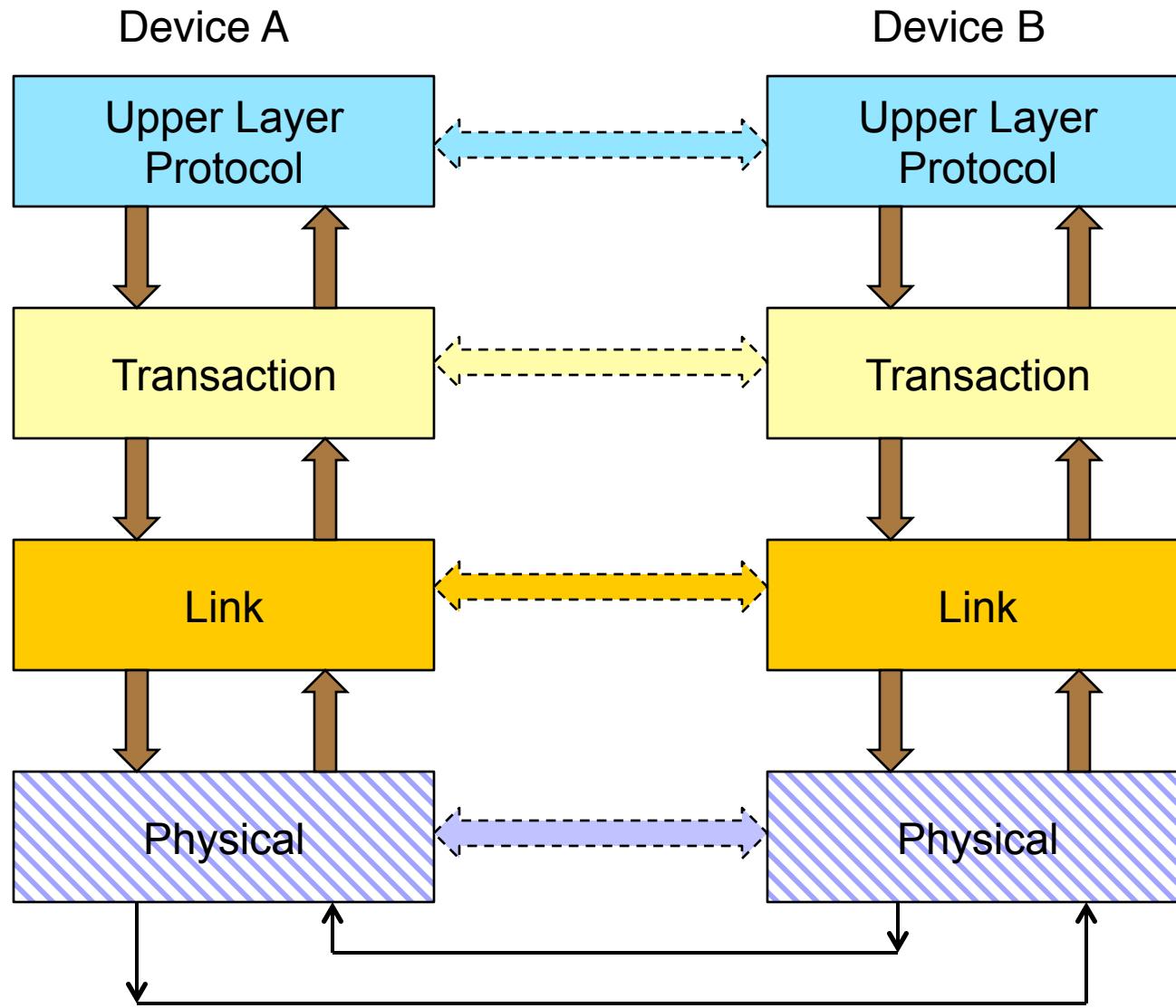
The diagram illustrates the Type 1 Configuration Space Header (Bridge) structure. It shows the fields defined in the table above, along with several annotations:

- ID Routing:** An arrow points from the "Header Type = 1h" field in byte 14 to the "Secondary Latency Timer" field in byte 24.
- Non-Prefetchable Memory:** An arrow points from the "I/O Limit" field in byte 32 to the "Memory Base" field in byte 36.
- Prefetchable Memory:** An arrow points from the "Prefetchable Memory limit" field in byte 40 to the "Prefetchable Memory base" field in byte 44.
- I/O Addressing:** A large bracket on the right side groups the "I/O Base" and "I/O Limit" fields from bytes 32 and 36, and the "I/O Base" and "I/O Limit" fields from bytes 44 and 52, under the heading "I/O Addressing".

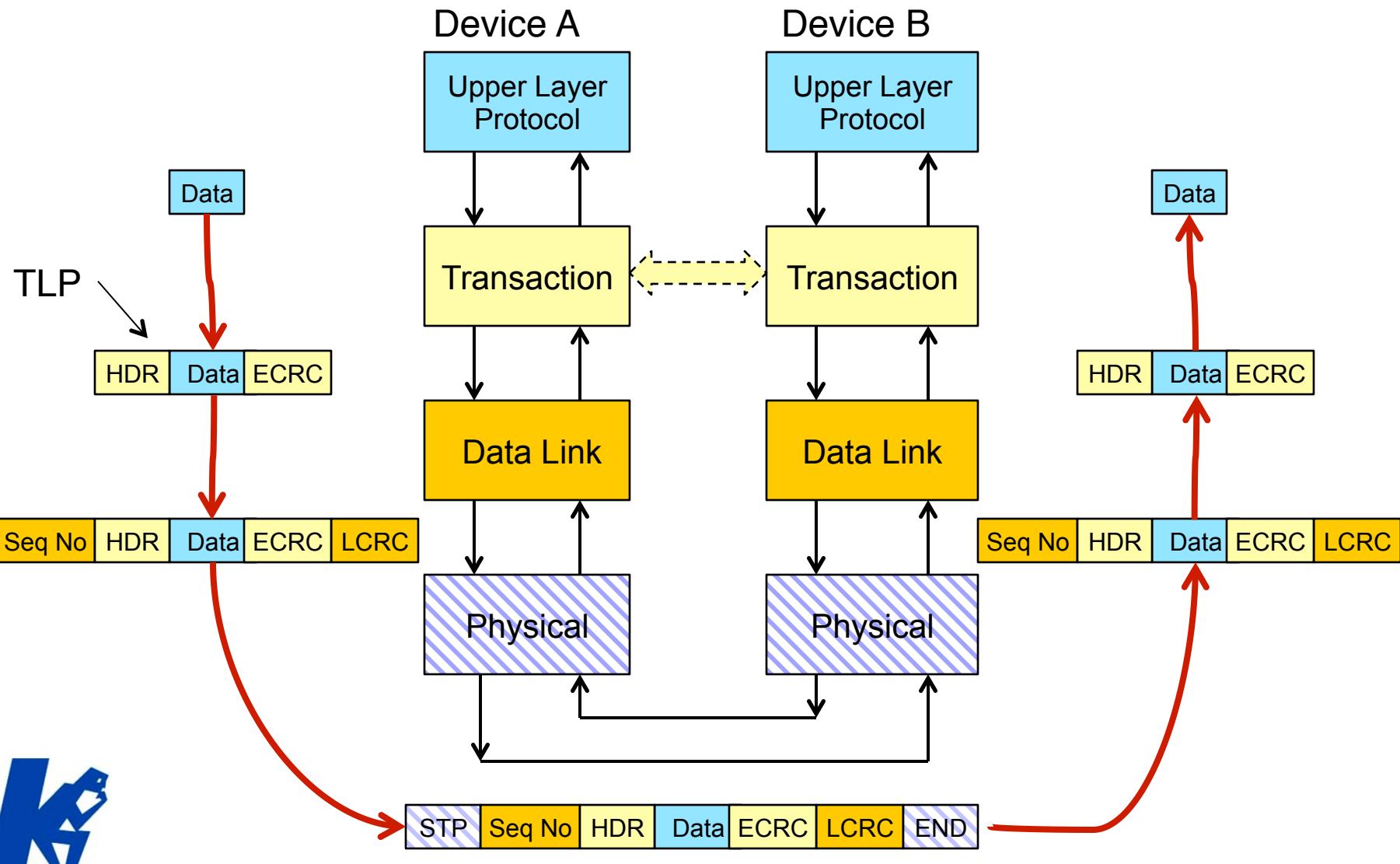
PCI Packets



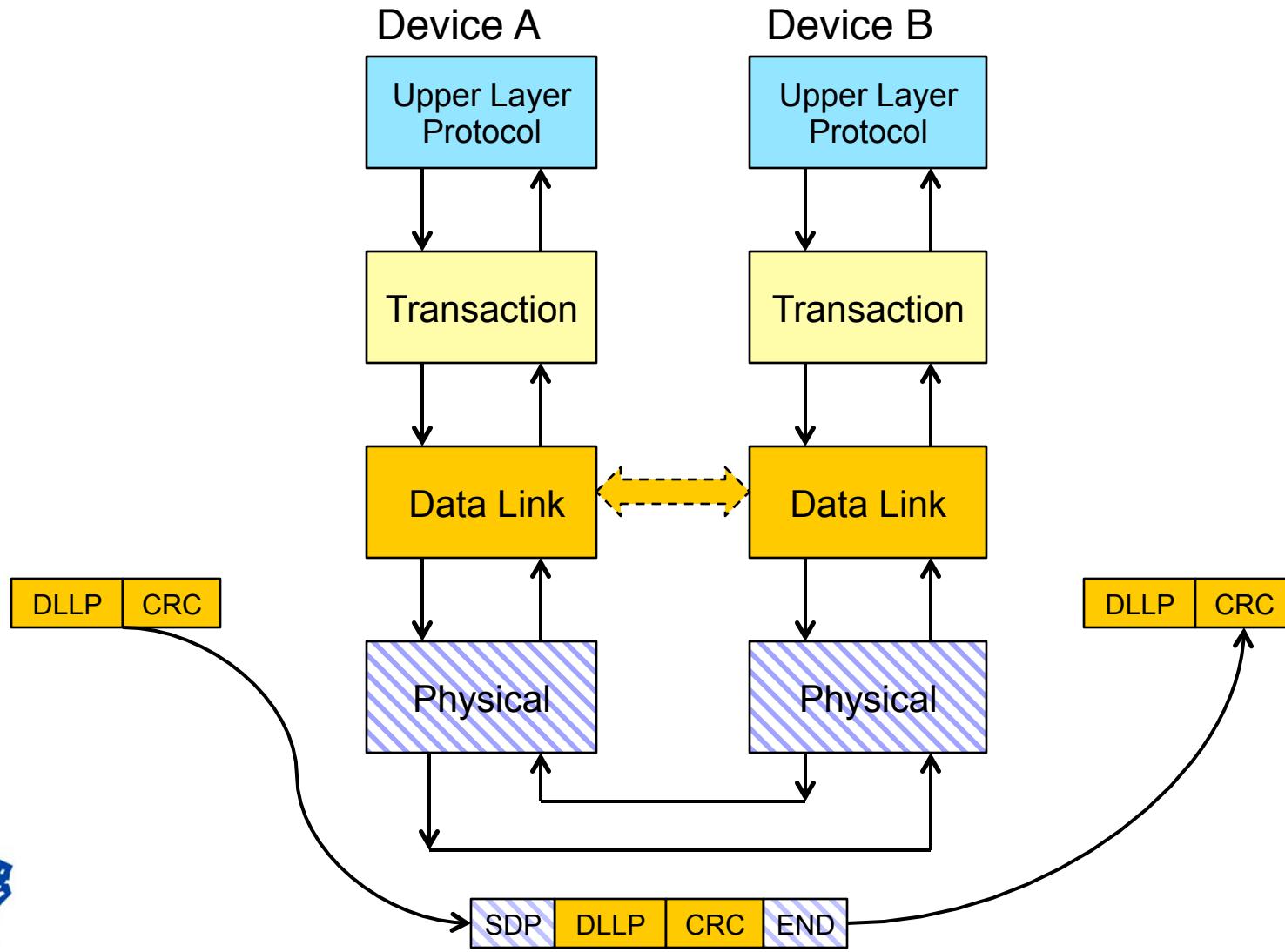
Layers



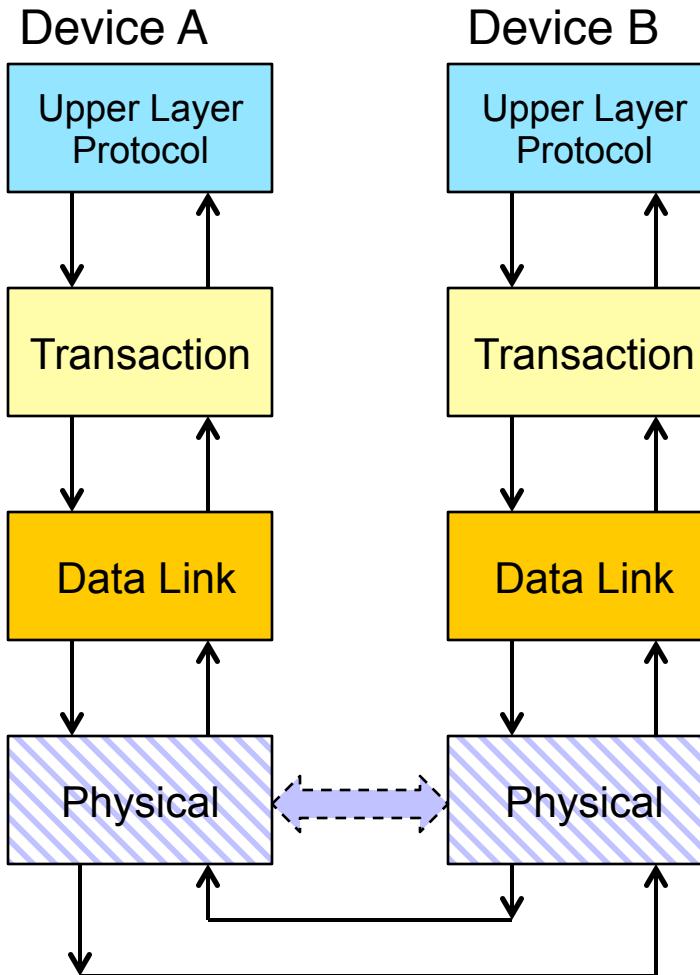
Transaction Layer Packets



Data Link Layer Packets



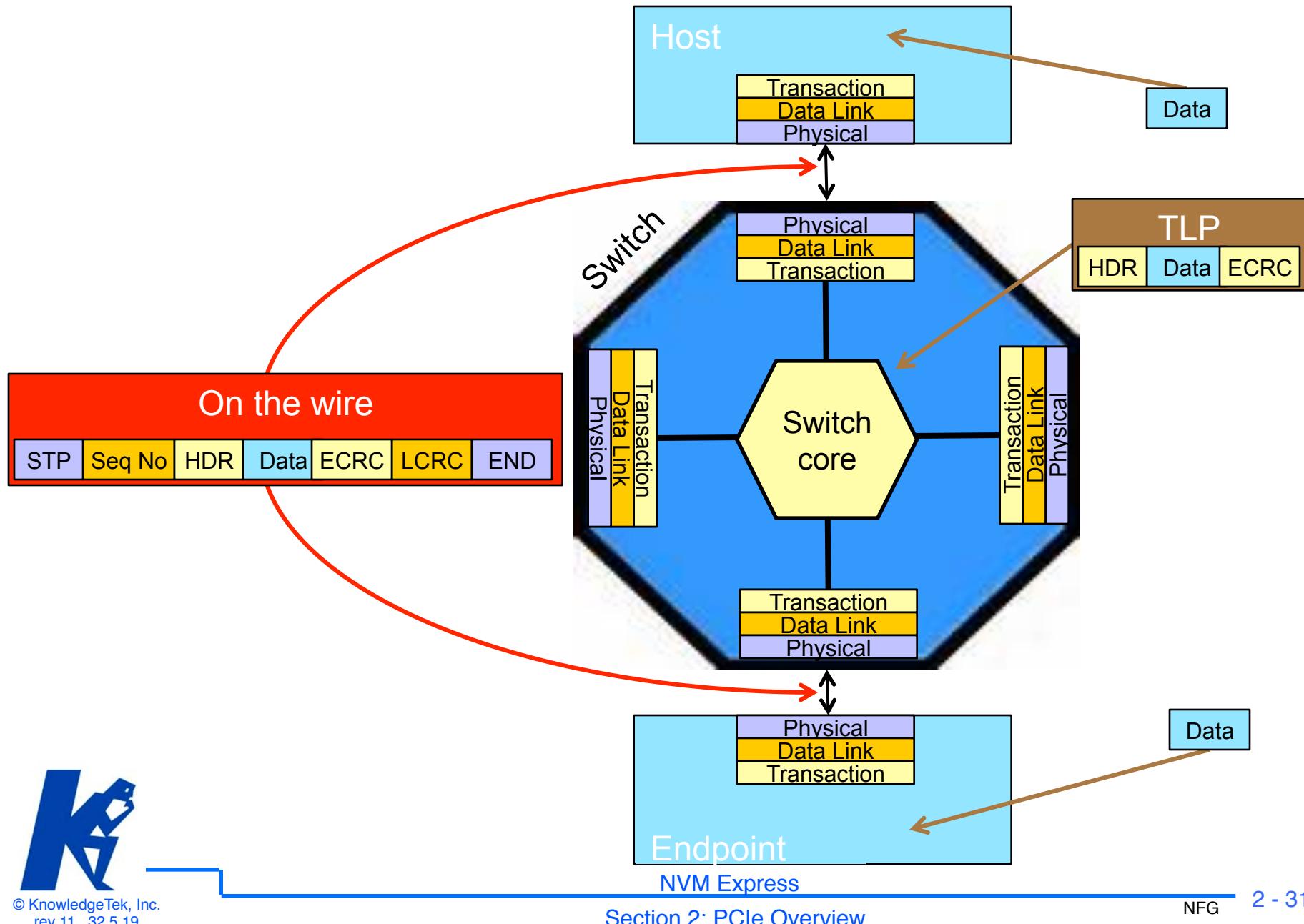
Ordered Sets



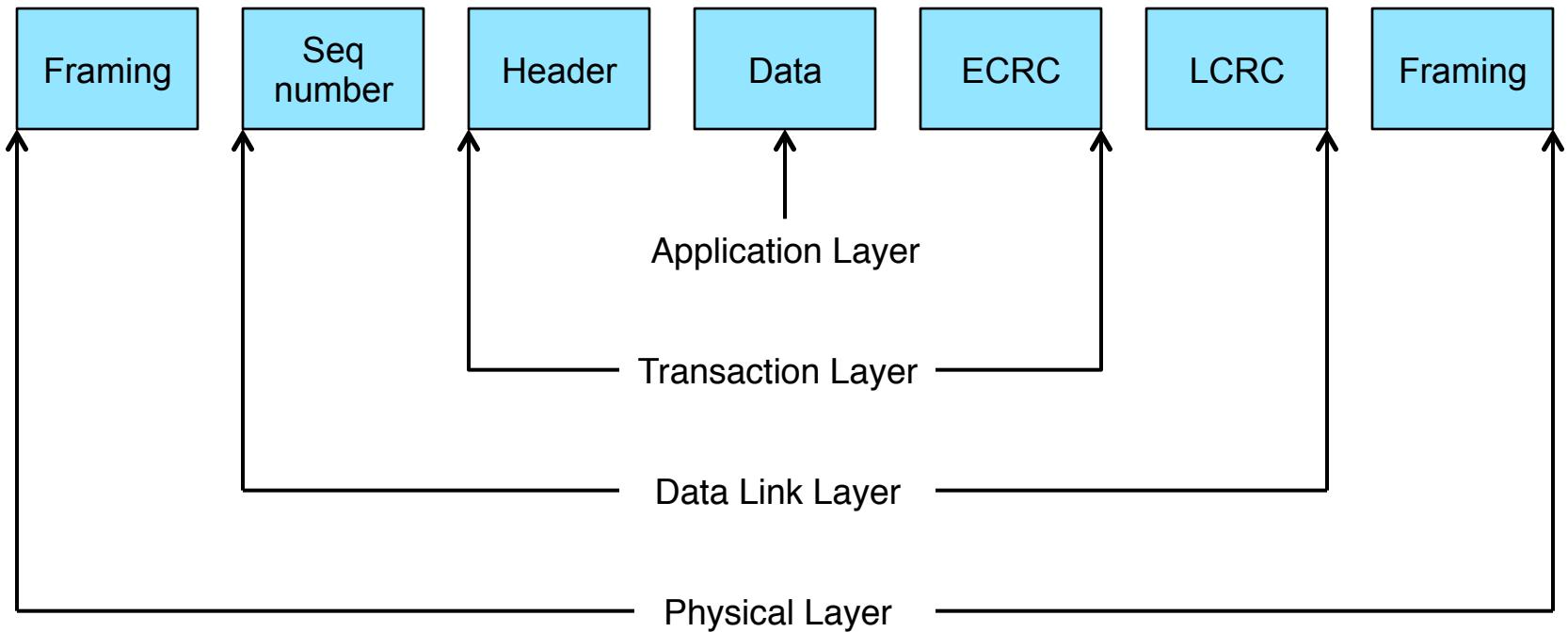
NVM Express

Section 2: PCIe Overview

Layers



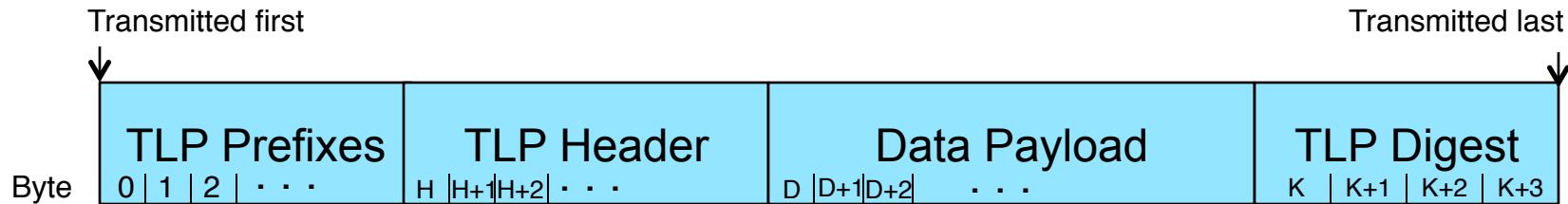
Packet Development



ECRC – End to End protection

LCRC – Link layer protection

Transaction Layer Packet Format Overview



TLP Prefixes

Optional

TLP Header

Type of packet

Routing information

Data Payload

When applicable

From application layer

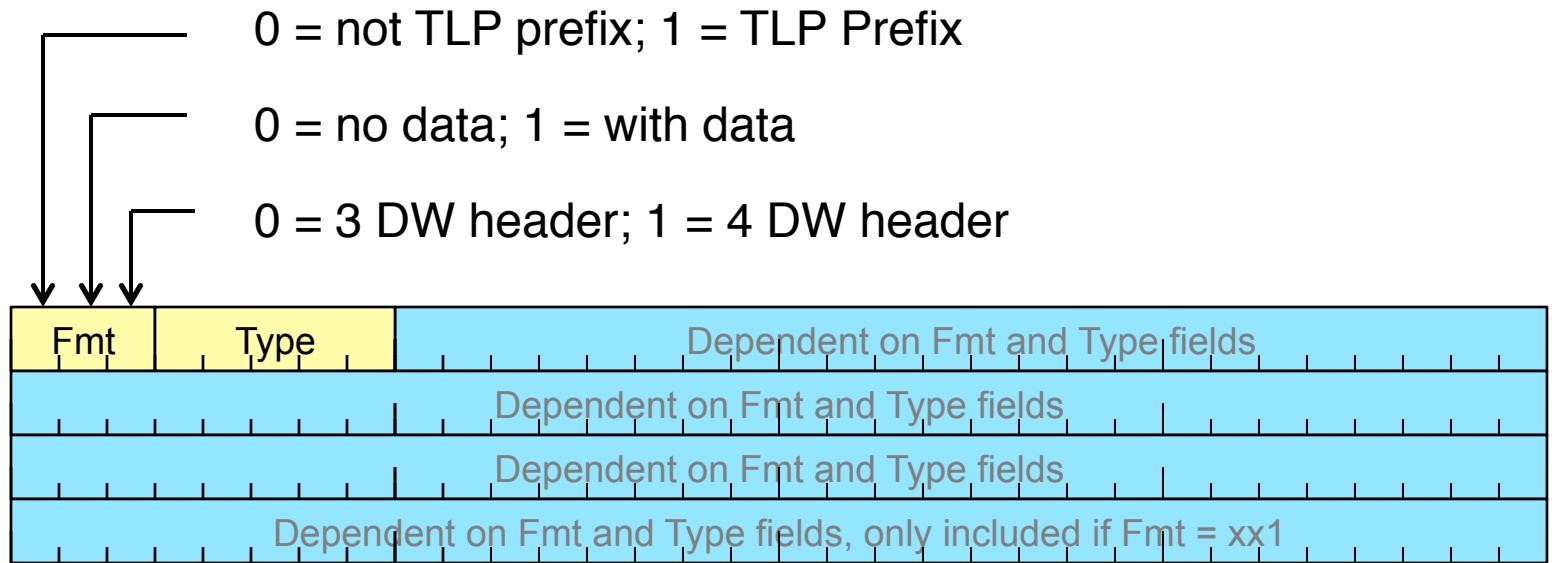
TLP Digest

32 bit CRC

Optional



Packet Header Fields



Type field, see next page

Packet Header Fields

FMT (b)	TYPE (h)	Description
00x	00	Memory Read Request
00x	01	Memory Read Request Locked
01x	00	Memory Write Request
000	02	I/O Read Request
010	02	I/O Write Request
000	04	Configuration Ready Type 0
010	04	Configuration Write Type 0
000	05	Configuration Read Type 1
010	05	Configuration Write Type 1
000	1B	Deprecated TLP Type
010	1B	Deprecated TLP Type
001	1 0r2r1r0	Message Request; r2r1r0 specify message routing mechanism
011	1 0r2r1r0	Message Request with data payload
000	0A	Completion without Data
010	0A	Completion with Data
000	0B	Completion for Locked Memory Read without data
010	0B	Completion for Locked Memory Read
01x	0C	Fetch and Add AtomicOp Request
01x	0D	Unconditional Swap AtomicOp Request
01x	0E	Compare and Swap AtomicOP Reqeust
100	0 c3l2l1l0	Local TLP Prefix; l3l2l1l0 specify the Local TLP Prefix Type
100	1 e3e2e1e0	End-End TLP Prefix; e3e2e1e0 specify TLP prefix type

Note: Type field is only 5 bits.

NVM Express

PCI Transactions



PCI Transactions

Memory Read

Memory Write

Configuration read – initialization and configuration

Configuration write – initialization and configuration

Message without Data

Message with Data

I/O read (legacy only)

I/O write (legacy only)



PCI Writes and Reads

Posted (no response)

Memory Writes

Message Requests

Non-posted (response expected)

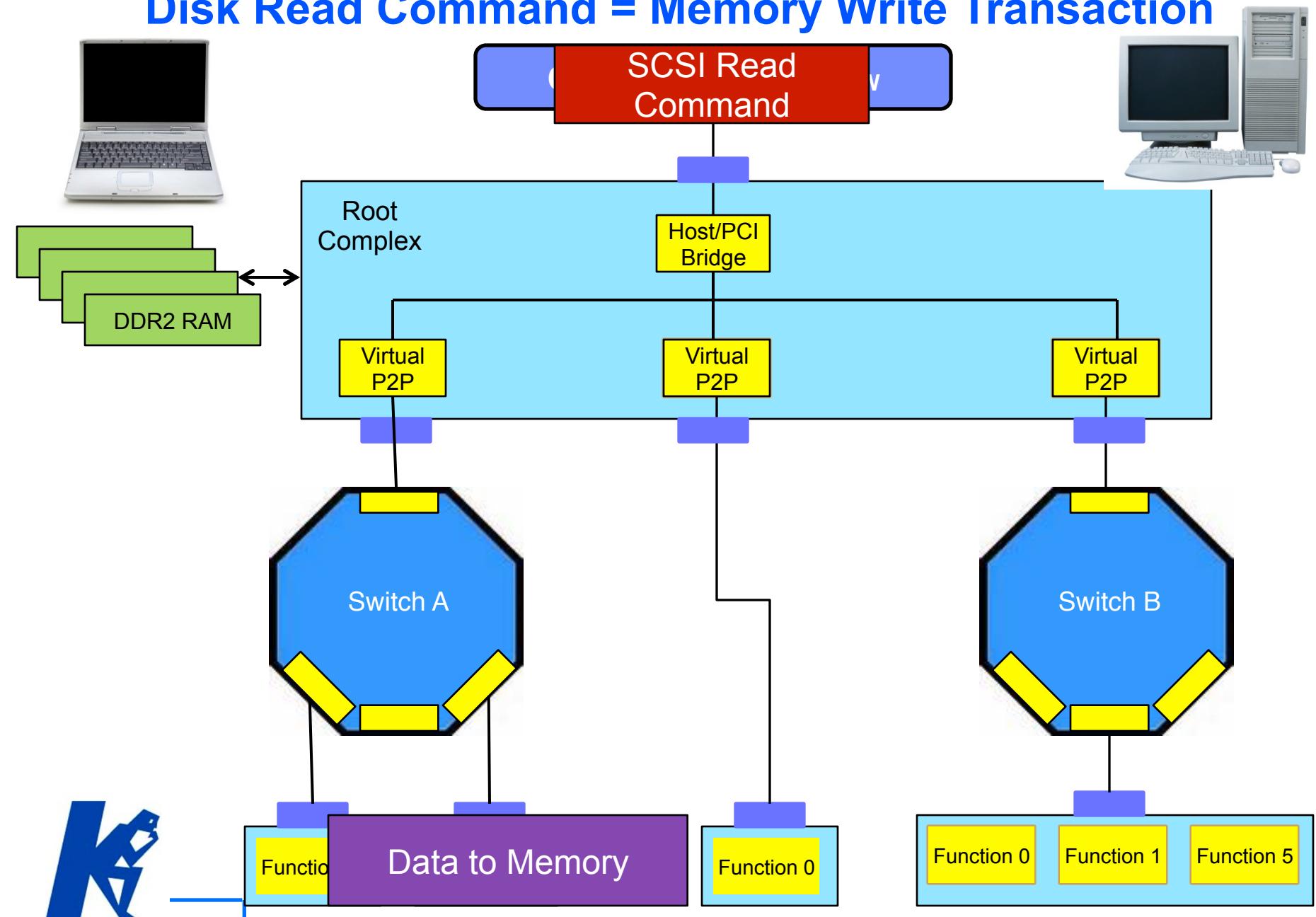
All reads – read data is returned

I/O writes

Configuration writes

Command vs. Transaction

Disk Read Command = Memory Write Transaction



Command vs. Transaction Terminology

Generally

Commands are issued by higher level components

e.g. Read data, write data, Inquiry

Transactions are issued by lower level components

e.g. Posted write, non-posted read, configuration write

Usually issuing commands will be implemented by issuing transactions

PCIe usage

PCIe uses the term “transactions” throughout the specifications

Except:

PCI command register in Configuration Header Space, and

Writing to the Slot Control register in Hot-plug capable DS ports

MSI

MSI-X



Message Signaled Interrupts

A Device Function requests service by writing system-specified Message Data to the system-specified address using Memory Write transaction

2 Systems – MSI and MSI-X

Device may implement both, but only one can be enabled

Each Function may have one MSI capability

Both are disabled following a reset, must be enabled by software or use INT#



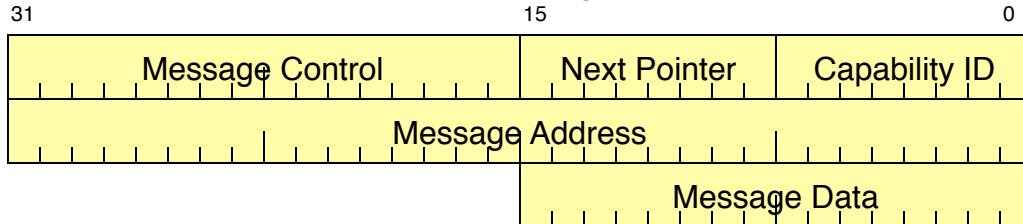
Check out: http://en.wikipedia.org/wiki/Message_Signaled_Interrupts

Interrupt Capability

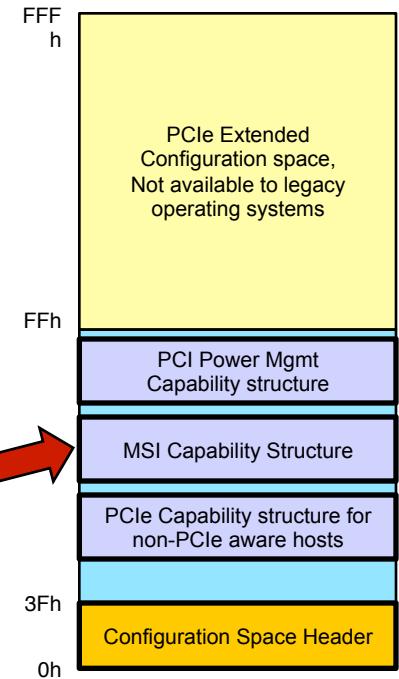
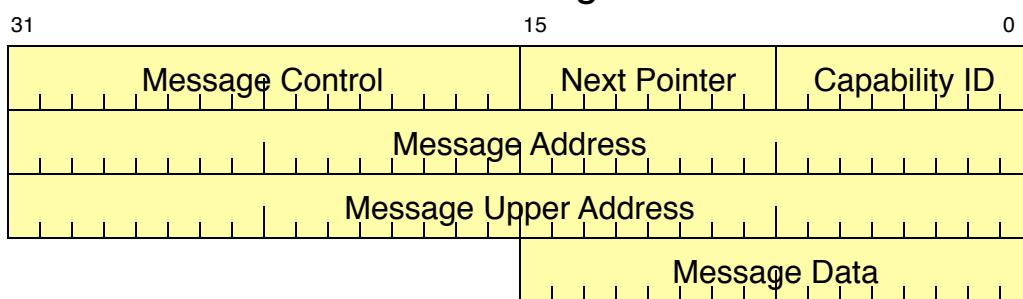
INT	Out-of-Band	Carry over from ISA 4 shared interrupts (INT A/B/C/D) Signaled by messages in PCIe Support encouraged, use discouraged
MSI	In-Band	Added in PCI 2.2 Up to 32 non-shared interrupts Device uses Message Data to identify the sender and IRQ number
MSI-X	In-Band	Added in PCI 3.0 Up to 2048 non-shared interrupts Device uses vector table to identify the sender and IRQ number

MSI Capability Structures

32-bit Message Address

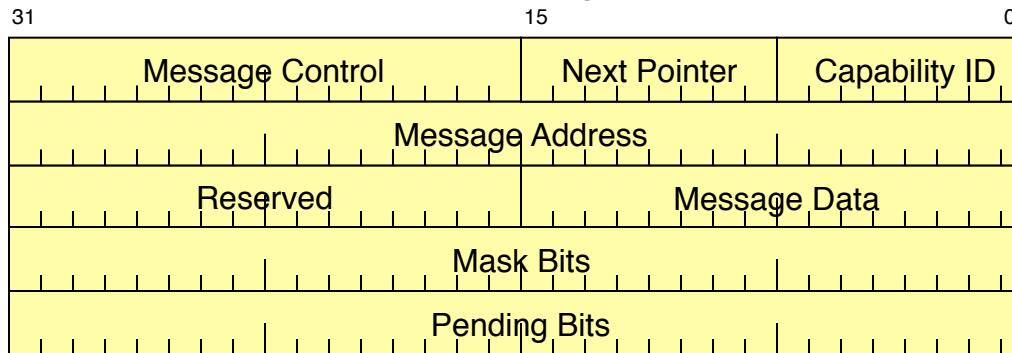


64-bit Message Address

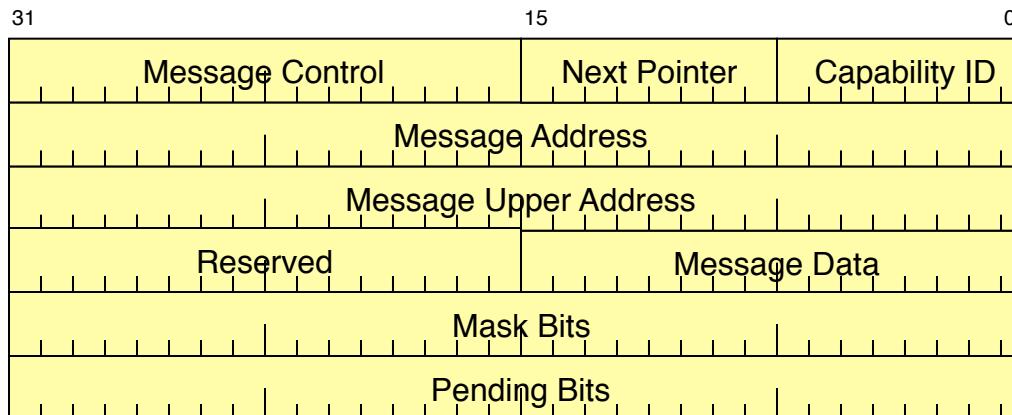


MSI Capability Structures w/Pre-Vector Masking

32-bit Message Address



64-bit Message Address



MSI Capability Structures Field Descriptions

Capability ID – 05h means MSI capable

Next Pointer – Pointer to the next item in the capabilities list (Null for final item)

Message Control –

Bits 15:09	Reserved
8	Pre-Vector masking capable
7	64-bit address capable
6:4	Multiple Message Enable
3:1	Multiple Message Capable
0	MSI Enable

Message Address –

63/31:02	System assigned Address to write interrupt messages
01:00	Reserved

Message Data – System specified identifier for the Device/Function

Mask Bits – For each mask bit set, the Function is prohibited from sending the associated message

Pending Bits – For each Pending Bit set, the Function has a pending associated message

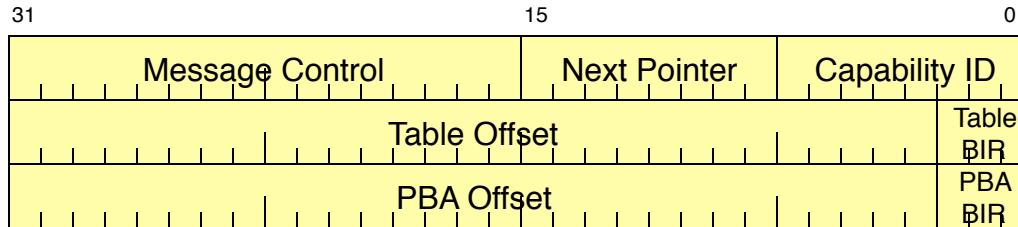


System assigned ID for
this Device/Function

1 – 5 bits to identify
Interrupt number (31:1)

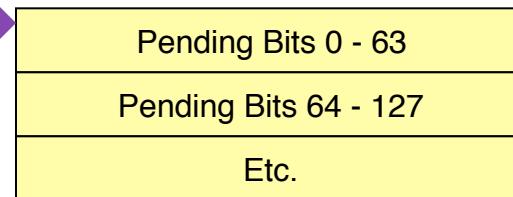
MSI-X Structures

MSI-X Capability Structure

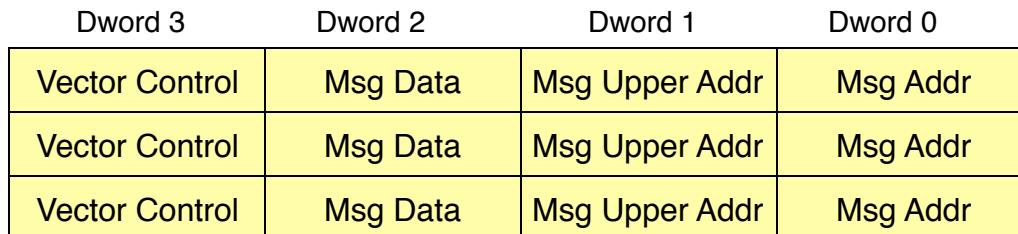


See
next
page

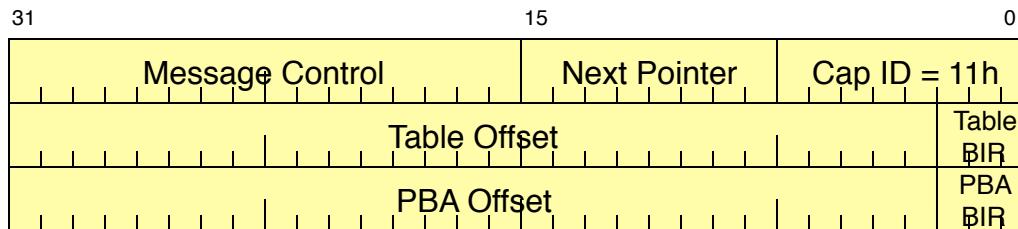
MSI-X Pending Bit Structure



MSI-X Table Structure

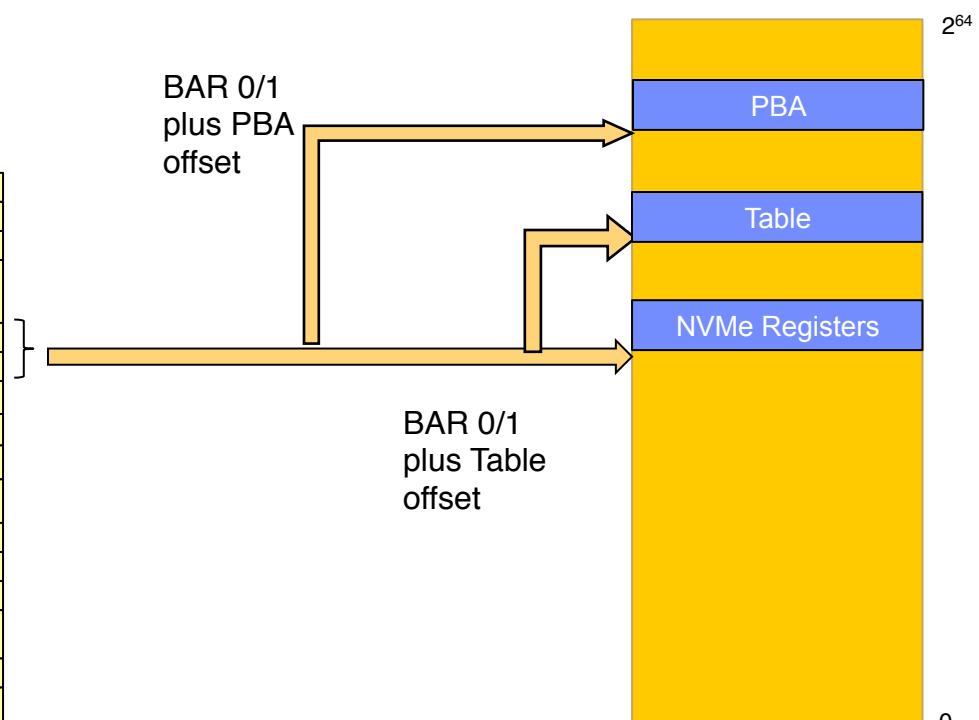


MSI-X Capability Structure



Byte

0	Device ID	Vendor ID
4	Status	Command
8	Class Code	Revision ID
C	BIST	Header Type = 0h
	Header Type = 0h	Master Latency Timer
	Cache Line Size	
10	BAR0 – MLBAR – NVMe Registers	
14	BAR1 – MUBAR – NVMe Registers	
18	BAR2 – I/O based accesses, if supported	
1C	BAR3 - Reserved	
20	BAR4 – Vendor Specific	
24	BAR5 – Vendor Specific	
28	Cardbus CIS Pointer	
2C	Subsystem ID	Subsystem Vendor ID
30	Expansion ROM Base Address	
34	Reserved	Capabilities Pointer
38	Reserved	
3C	Max Latency = 00h	Min Grant = 00h
	Interrupt Pin	Interrupt Line



MSI-X Capability Structures Field Descriptions

Capability ID – 11h means MSI-X capable

Next Pointer – Pointer to the next item in the capabilities list (Null for final item)

Message Control –

Bits 15	MSI-X enable
14	Function Mask
13:11	Reserved
10:00	Table Size

Table/PBA BIR – Which Function BAR is used to map MSI-X Table into memory space

0	10h
1	14h
2	18h
3	1Ch
4	20h
5	24h
6:7	Reserved

Message Address –

63/31:02	Message Address
01:00	Reserved

Message Data – System specified message data

Mask Bits – For each mask bit set, the Function is prohibited from sending the associated message

Pending Bits – For each Pending Bit set, the Function has a pending associated message

Vector Control –

31:01	Reserved or Steering Table
00	Mask Bit

NVM Express

Link Layer

Covered in this Section

DLLP's

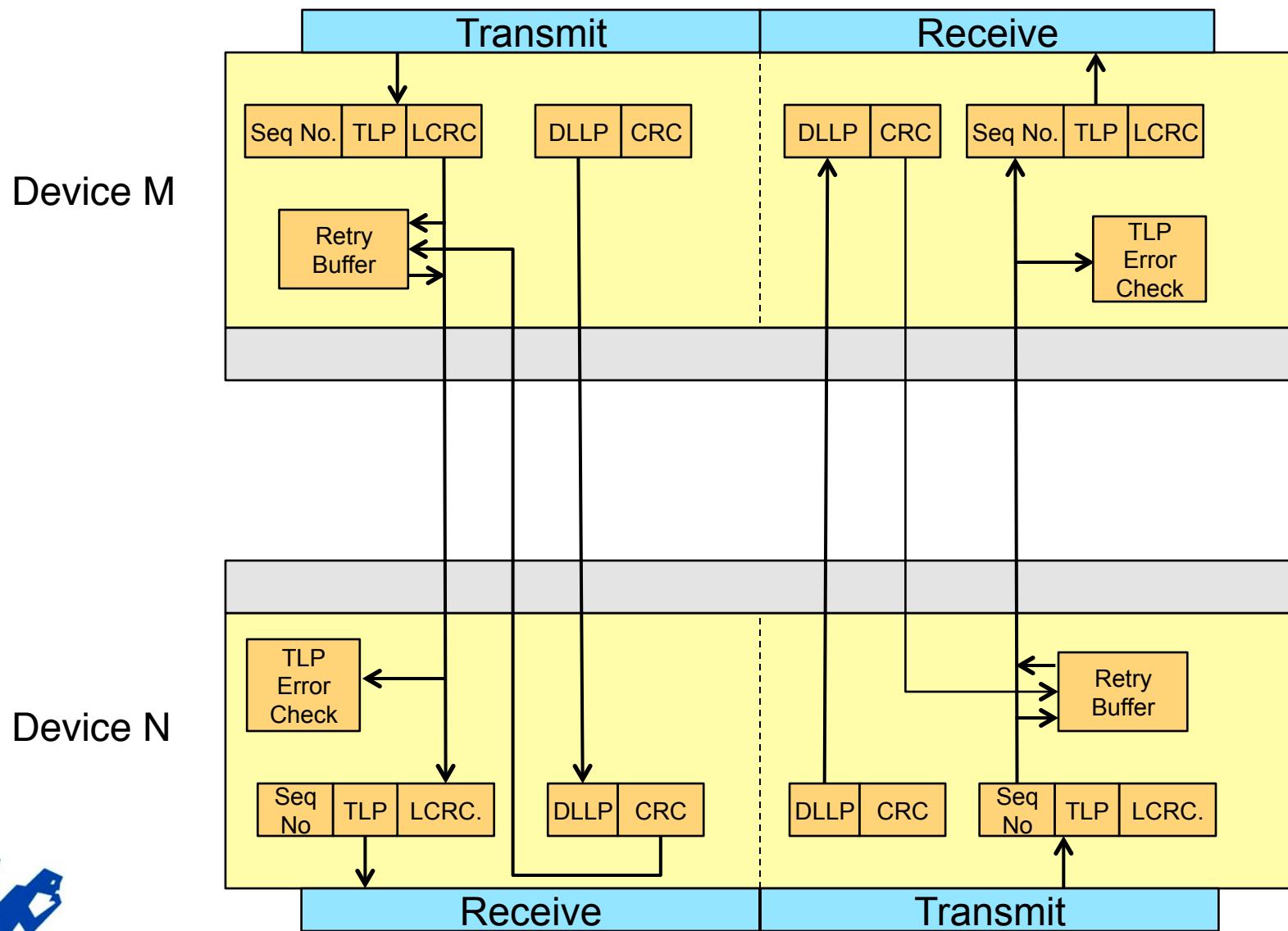
LCRC

Sequence numbers are explained in a later section

ACK/NAK protocol is in a later section



Data Link Layer Block Diagram



DLLP

DLLP = Data Link Layer Packet

From one link layer to its peer

Never go to the transaction layer

DLLP Type	Type Dependent bytes	CRC=16
-----------	----------------------	--------

DLLP Types

Type Code	DLLP Type
00h	ACK
10h	NAK
20h	PM Enter L1
21h	PM Enter L23
23h	PM Active State Request L1
24h	PM Request ACK
30h	Vendor Specific
4 *h	Init FC1-P
5*h	Init FC1-NP
6*h	Init FC1-Cpl
C*h	Init FC2-P
D*h	Init FC2-NP
E*h	Init FC2-Cpl
8*h	Update FC-P
9*h	Update FC-NP
A*h	Update FC-Cpl

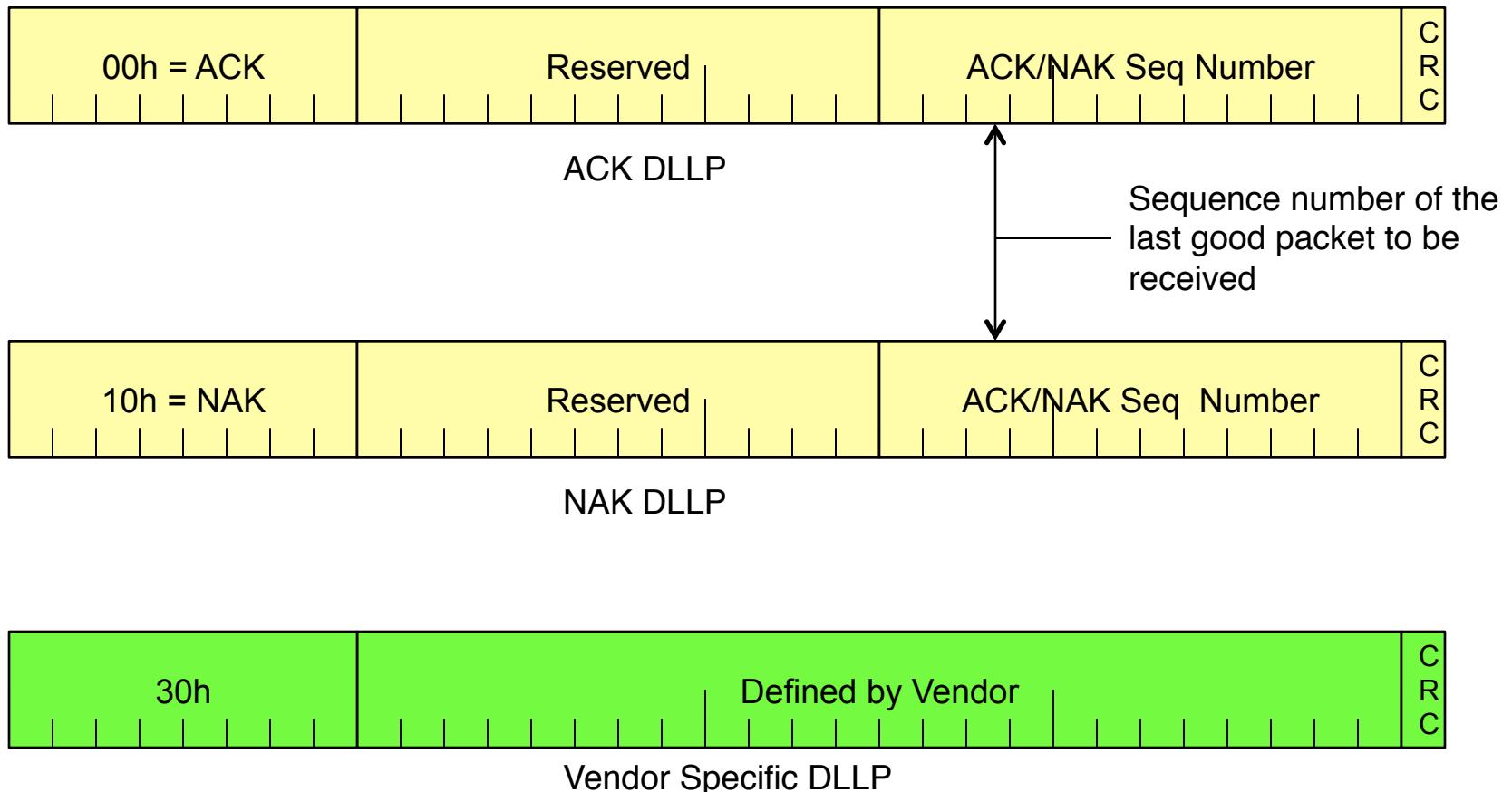
P = Posted
NP = Non-Posted
Cpl = Completion
FC = Flow Control

* Specifies Virtual Channel

* = 0 V₂ V₁ V₀



DLLP Details (Continued)



Teledyne LeCroy PETracer(TM) - PCI Express Protocol Analyzer - [C:\Users\Public\Documents\LeCroy\PETracer\Sample Files\cfg_pci_express.pex]

File Setup Record Generate Report Search View Tools Window Help

Packet 5 R← 2.5 DLLP ACK AckNak_Seq_Num CRC 16 Idle Time Stamp
 5 x1 2.5 33 23 0xD53A 0.000 ns 0000 . 000 000 368 s

Packet 6 R← 2.5 TLP Cpl CplID Length RequesterID Tag CompleterID Status BCM Byte Cnt
 6 x1 33 010:01010 1 001:02:3 25 004:05:6 SC 0 4

Lwr Addr Reserved Capability Pointer ECRC LCRC Time Delta Time Stamp
 0x00 0x000000 0x Register 'Capability Pointer' (Read-only) Offset 0x34
 Points to PCI Express Capability Structure 1 112.000 ns 0000 . 000 000 400 s

Packet 7 R→ 2.5 DLLP ACK AckNak_Seq_Num CRC 16 Idle Time Stamp
 7 x1 33 33 0x104D 0.000 ns 0000 . 000 000 512 s

Packet 8 R→ 2.5 TLP Cfg CfgRd1 Length RequesterID Tag DeviceID Register 1st BE
 8 x1 27 000:00101 1 001:02:3 29 004:05:6 0x044 1111

ECRC LCRC Time Delta Time Stamp

Link Tracker - Packet # 5

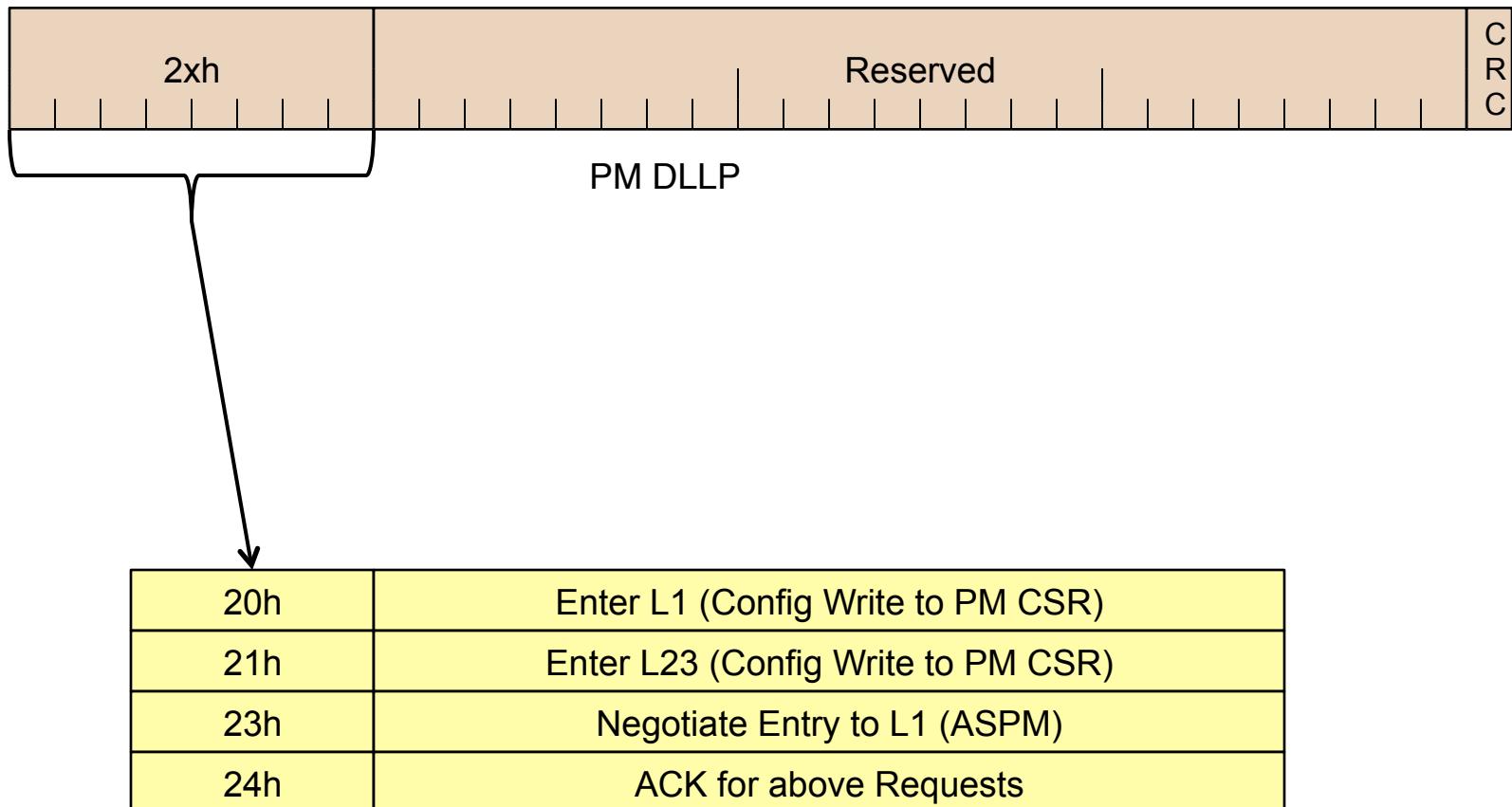
Time	Packet #	Upstream	Downstream
0.000 000 364			FD
0.000 000 368	5 (Upstream)	5C 00 00 00 17 D5 3A FD	
0.000 000 372			
0.000 000 376			
0.000 000 380			
0.000 000 384			
0.000 000 388			
0.000 000 392			
0.000 000 396			
0.000 000 400	6 (Upstream)	FE 00 00 21 4A 00 00 80 01 04 2E 00 04 01 13 19 00 44 00 00	
0.000 000 404			
0.000 000 408			
0.000 000 412			
0.000 000 416			
0.000 000 420			
0.000 000 424			
0.000 000 428			
0.000 000 432			
0.000 000 436			
0.000 000 440			
0.000 000 444			
0.000 000 448			
0.000 000 452			
0.000 000 456			
0.000 000 460			
0.000 000 464			
0.000 000 468			

5C = SDP
 017 = Seq number
 D53A = CRC
 FD = END
 FB = STP

Ready

Search: Fwd FG 2 - 57

DLLP Details (Continued)



DLLP Details

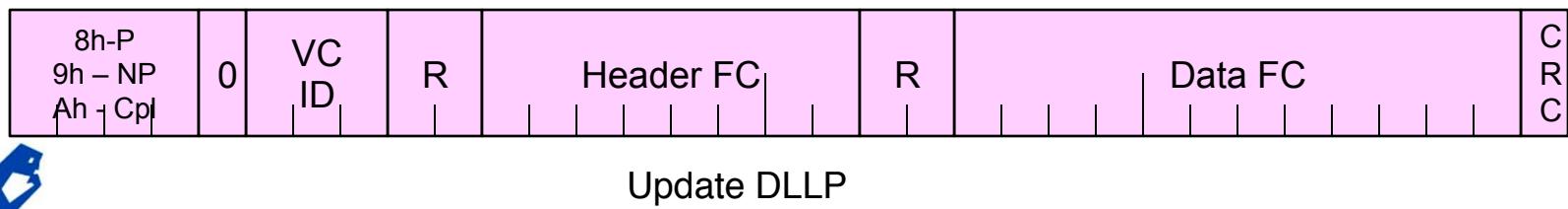
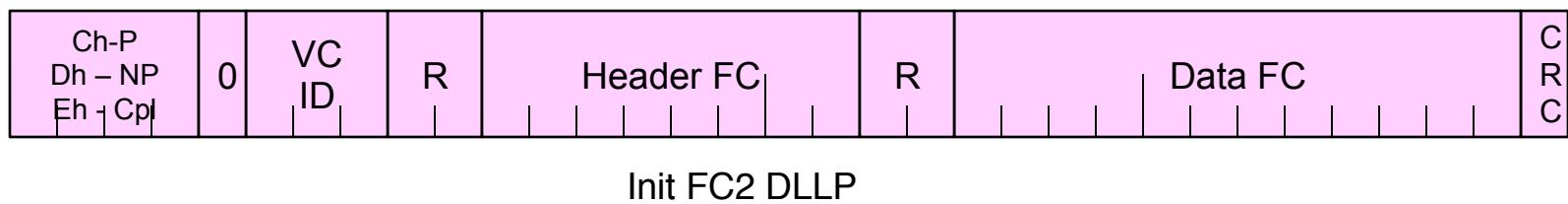
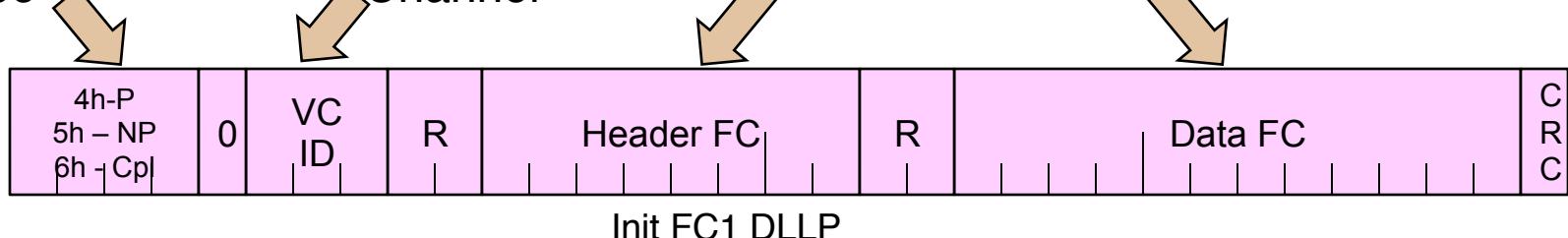
Advertised Credits for:

Which Transaction Type

Which Virtual Channel

Header

Data



Covered in later section
on flow control

NVM Express

Section 2: PCIe Overview

File Setup Record Generate Report Search View Tools Window Help

TRG RTZ

Pkt Link Split NUM ATA SCSI

Trace View

Packet 2206	R← 2.5:x4	DLLP	InitFC2-NP	VC ID:0	HdrFC:8	DataFC:2047	CRC 16:0x250F	Time Delta:0.000 ns	TS:0000 . 129 541 400 s
Packet 2207	R← 2.5:x4	DLLP	InitFC2-P	VC ID:0	HdrFC:4	DataFC:16	CRC 16:0x81C6	Time Delta:8.000 ns	TS:0000 . 129 541 400 s
Packet 2208	R← 2.5:x4	DLLP	InitFC2-Cpl	VC ID:0	HdrFC:0	DataFC:0	CRC 16:0xA2ED	Time Delta:0.000 ns	TS:0000 . 129 541 408 s
Packet 2209	R→ 2.5:x4	DLLP	InitFC2-NP	VC ID:0	HdrFC:12	DataFC:0	CRC 16:0x92E6	Time Delta:8.000 ns	TS:0000 . 129 541 408 s
Packet 2210	R← 2.5:x4	DLLP	InitFC2-P	VC ID:0	HdrFC:127	DataFC:2047	CRC 16:0xF246	Time Delta:0.000 ns	TS:0000 . 129 541 416 s
Packet 2211	R→ 2.5:x4	DLLP	InitFC2-Cpl	VC ID:0	HdrFC:0	DataFC:0	CRC 16:0xA2ED	Time Delta:8.000 ns	TS:0000 . 129 541 416 s
Packet 2212	R← 2.5:x4	DLLP	InitFC2-NP	VC ID:0	HdrFC:8	DataFC:2047	CRC 16:0x250F	Time Delta:0.000 ns	TS:0000 . 129 541 424 s
Packet 2213	R→ 2.5:x4	DLLP	InitFC2-P	VC ID:0	HdrFC:4	DataFC:16	CRC 16:0x81C6	Time Delta:8.000 ns	TS:0000 . 129 541 424 s
Packet 2214	R← 2.5:x4	DLLP	InitFC2-Cpl	VC ID:0	HdrFC:0	DataFC:0	CRC 16:0xA2ED	Time Delta:0.000 ns	TS:0000 . 129 541 432 s
Packet 2215	R→ 2.5:x4	DLLP	InitFC2-NP	VC ID:0	HdrFC:12	DataFC:0	CRC 16:0x92E6	Time Delta:8.000 ns	TS:0000 . 129 541 432 s
Packet 2216	R← 2.5:x4	DLLP	InitFC2-P	VC ID:0	HdrFC:127	DataFC:2047	CRC 16:0xF246	Time Delta:0.000 ns	TS:0000 . 129 541 440 s
Packet 2217	R→ 2.5:x4	DLLP	InitFC2-Cpl	VC ID:0	HdrFC:0	DataFC:0	CRC 16:0xA2ED	Time Delta:8.000 ns	TS:0000 . 129 541 440 s
Packet 2218	R← 2.5:x4	DLLP	InitFC2-NP	VC ID:0	HdrFC:8	DataFC:2047	CRC 16:0x250F	Time Delta:0.000 ns	TS:0000 . 129 541 448 s
Packet 2219	R→ 2.5:x4	DLLP	UpdateFC-P	VC ID:0	HdrFC:4	DataFC:16	CRC 16:0x3CF9	Time Delta:8.000 ns	TS:0000 . 129 541 448 s
Packet 2220	R← 2.5:x4	DLLP	InitFC2-Cpl	VC ID:0	HdrFC:0				
Packet 2221	R→ 2.5:x4	DLLP	UpdateFC-NP	VC ID:0	HdrFC:1				
Packet 2222	R← 2.5:x4	DLLP	InitFC2-P	VC ID:0	HdrFC:1				
Packet 2223	R← 2.5:x4	DLLP	InitFC2-NP	VC ID:0	HdrFC:8				
Packet 2224	R← 2.5:x4	DLLP	InitFC2-Cpl	VC ID:0	HdrFC:0				
Packet 2225	R← 2.5:x4	DLLP	InitFC2-P	VC ID:0	HdrFC:1				
Packet 2226	R← 2.5:x4	DLLP	InitFC2-NP	VC ID:0	HdrFC:8				
Packet 2227	R← 2.5:x4	DLLP	InitFC2-Cpl	VC ID:0	HdrFC:0				
Packet 2228	R← 2.5:x4	DLLP	UpdateFC-P	VC ID:0	HdrFC:1				
Packet 2229	R← 2.5:x4	DLLP	UpdateFC-NP	VC ID:0	HdrFC:8				
Packet 2230	R→ 2.5:x4	SKIP	COM:K28.5	SKIP Symbols:K28.0 K					
Packet 2231	R← 2.5:x4	SKIP	COM:K28.5	SKIP Symbols:K28.0 K					
Packet 2232	R→ 2.5:x4	SKIP	COM:K28.5	SKIP Symbols:K28.0 K					
Packet 2233	R← 2.5:x4	SKIP	COM:K28.5	SKIP Symbols:K28.0 K					
Packet 2234	R→ 2.5:x4	DLLP	UpdateFC-P	VC ID:0	HdrFC:4				
Packet 2235	R→ 2.5:x4	DLLP	UpdateFC-NP	VC ID:0	HdrFC:1				
Packet 2236	R→ 2.5:x4	SKIP	COM:K28.5	SKIP Symbols:K28.0 K					
Packet 2237	R← 2.5:x4	SKIP	COM:K28.5	SKIP Symbols:K28.0 K					

Raw Symbols Display For Packet # 2218

Physical Lanes

Row #	0	1	2	3
0	0x5C	0xD0	0x02	0x07
1	0xFF	0x25	0x0F	0xFD

Data Appearance

- Byte
- Scrambled Byte
- 10 bit code
- Symbol (RD)
- LFSR
- Packet Fields

Packet

<- Prev Next ->

Running Disparity Error

Prev Next Done

Check for Understanding

- 1) What are DLLP packets?
- 2) When does the sender add LCRC?
- 3) How many bits in a Sequence number?

Physical Layer



Covered in this Section

Physical Layer Logical
Transmit Logic

Byte/lane striping

8b/10b

Encoding and scrambling

K-codes

Ordered Sets

128b/130b

Byte/lane striping

Framing tokens

Ordered sets

Encoding and scrambling

Receive specific logic

Clock recovery

Elastic buffer

Differential signaling

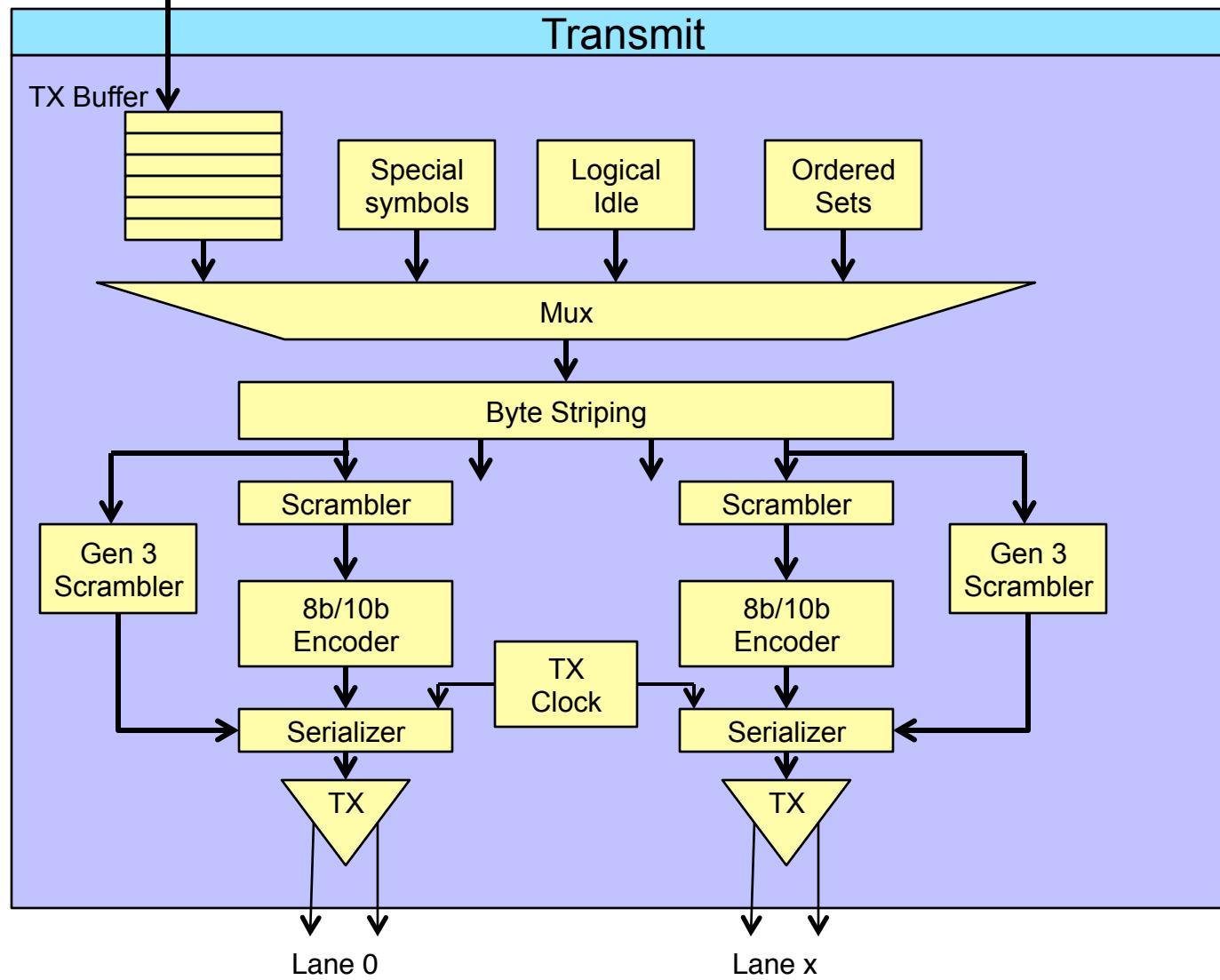
Initialization & training in later section



Physical Layer Transmit Logic

Phy Layer Block Diagram - Transmit

TLP & DLLP
From Link Layer

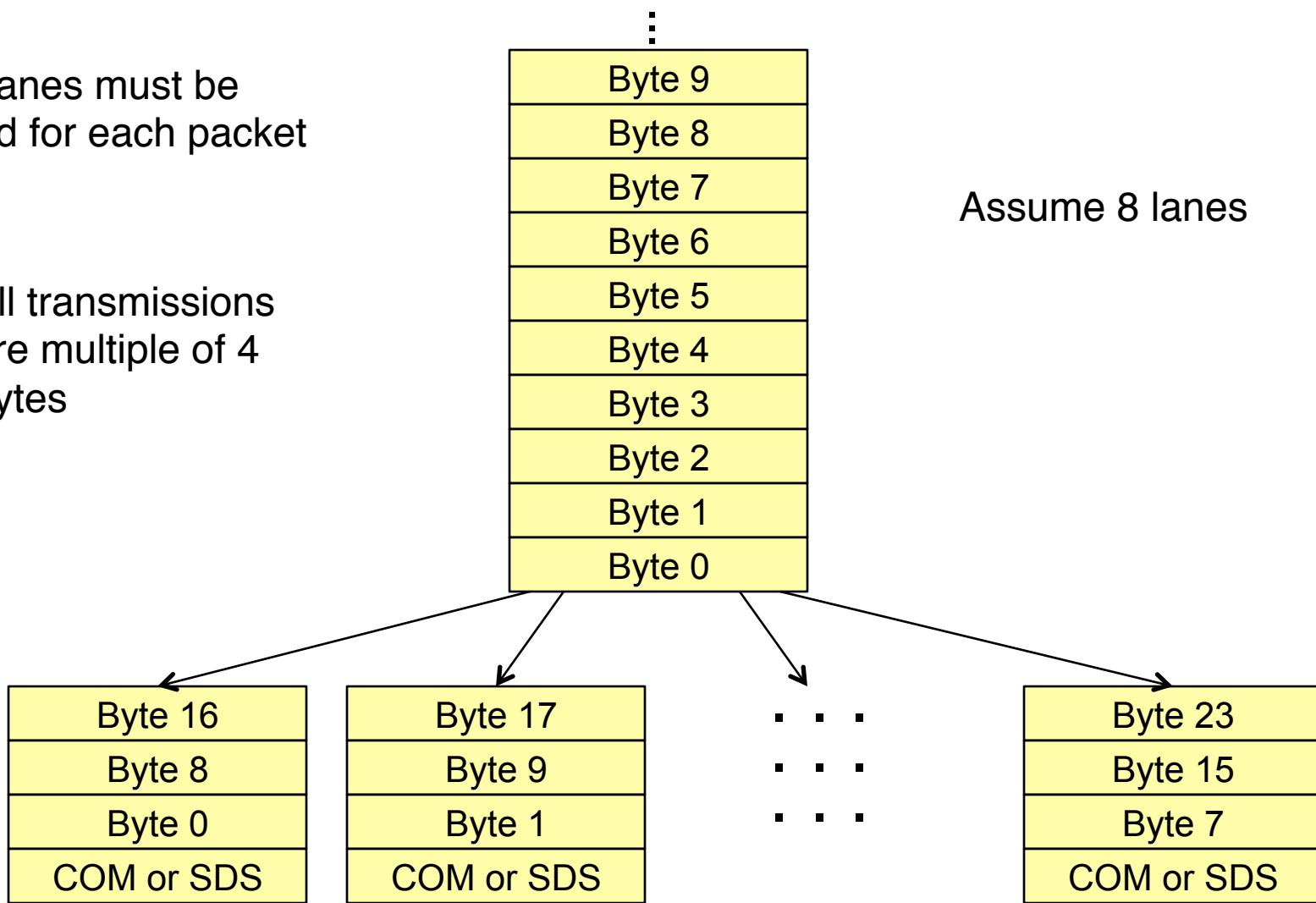


Byte Striping

All lanes must be used for each packet

All transmissions are multiple of 4 bytes

Assume 8 lanes



PCIe Generation 1 and 2

2.5 Gb/s and 5 Gt/s

8b/10b Encoding



8b/10b encoding

Why 8B/10B Encoding?

The Problem

Any pattern of 8 bit data is legal. Some patterns create problems for serial interfaces.

Long strings of all ones or all zeros have no transitions

Receive clock syncs on transition

Long strings of all ones or all zeros cause DC offset

AC coupling

Receiver AGC (if any) adjusts to average of 1's & 0's

The Solution

Encode each 8 bit byte into a 10 bit character. Use only those 10 bit patterns that do not have long strings of all ones or all zeros.

There are 1024 possible 10 bit characters. All 256 possible 8 bit bytes have at least one valid 10 bit translation. These are called **D-characters**.

Twelve 10 bit characters are not mapped to any 8 bit value. Serial transports use these for control functions. These are called **K-characters**.

8B/10B Rules

The only valid 10 bit characters are those which have:

4 ones and 6 zeros (e.g. 1001000101)

5 ones and 5 zeros (e.g. 1100011001)

6 ones and 4 zeros (e.g. 1001110101)

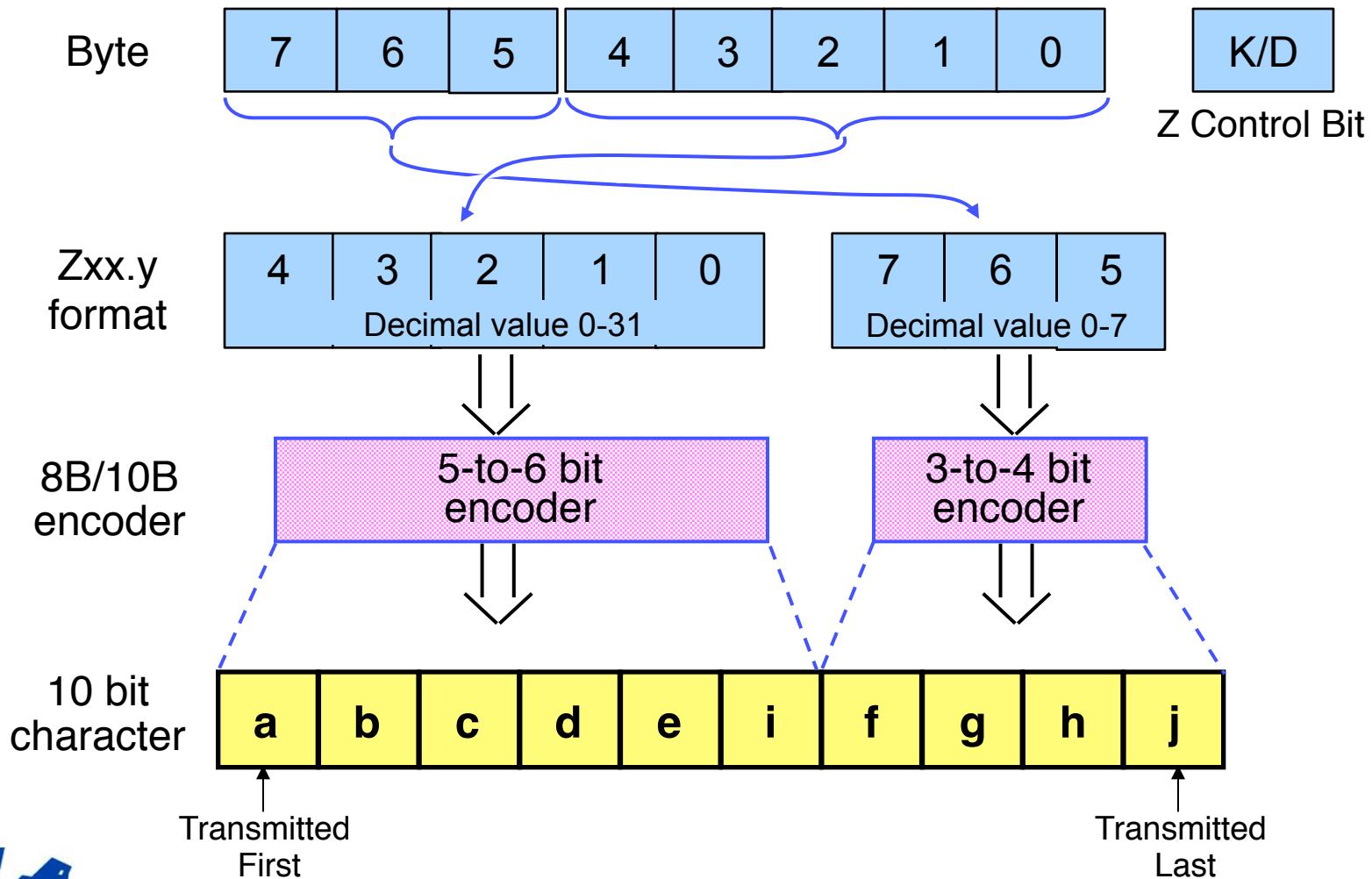
No more than 4 consecutive ones or zeros* within a 10 bit character

No more than 5 consecutive ones or zeros across two 10 bit characters, and no case of 2 zeros and 5 ones or visa versa*.

Full information on 8B/10B encoding can be found in IBM patent 4,486,739. Summary details of 8b/10b encoding are in T11/01-581v1.pdf or the book *Serial Attached SCSI: Beyond the Basics*.

* There is one exception to this rule. The K28.5 character contains a unique pattern with 5 consecutive ones or zeros, 1100000 or 0011111.

The Encoding Steps



8b/10b Transmission Code

Neutral Two
Encodes

Data Byte Name	Bits hgf edcba	Current RD - abcdei fghj	Current RD + abcdei fghj
D0.0	000 00000	100111 0100	011000 1011
D1.0	000 00001	011101 0100	100010 1011
D2.0	000 00010	101101 0100	010010 1011
D3.0	000 00011	110001 1011	110001 0100
D4.0	000 00100	110101 0100	001010 1011
D5.0	000 00101	101001 1011	101001 0100
D6.0	000 00110	011001 1011	011001 0100
D7.0	000 00111	111000 1011	000111 0100
D8.0	000 01000	111001 0100	000110 1011
D9.0	000 01001	100101 1011	100101 0100
D10.0	000 01010	010101 1011	010101 0100
D11.0	000 01011	110100 1011	110100 0100
D12.0	000 01100	001101 1011	001101 0100
D13.0	000 01101	101100 1011	101100 0100
D14.0	000 01110	011100 1011	011100 0100
D15.0	000 01111	010111 0100	101000 1011
D16.0	000 10000	011011 0100	100100 1011
D17.0	000 10001	100011 1011	100011 0100
D18.0	000 10010	010011 1011	010011 0100
D19.0	000 10011	110010 1011	110010 0100
D20.0	000 10100	001011 1011	001011 0100
D21.0	000 10101	101010 1011	101010 0100
D22.0	000 10110	011010 1011	011010 0100
D23.0	000 10111	111010 0100	000101 1011
D24.0	000 11000	110011 0100	001100 1011
D25.0	000 11001	100110 1011	100110 0100
D26.0	000 11010	010110 1011	010110 0100
D27.0	000 11011	110110 0100	001001 1011
D28.0	000 11100	001110 1011	001110 0100
D29.0	000 11101	101110 0100	010001 1011
D30.0	000 11110	011110 0100	100001 1011
D31.0	000 11111	101011 0100	010100 1011
D0.1	001 00000	100111 1001	011000 1001
D1.1	001 00001	011101 1001	100010 1001
D2.1	001 00010	101101 1001	010010 1001
D3.1	001 00011	110001 1001	110001 1001
D4.1	001 00100	110101 1001	001010 1001
D5.1	001 00101	101001 1001	101001 1001
D6.1	001 00110	011001 1001	011001 1001
D7.1	001 00111	111000 1001	000111 1001
D8.1	001 01000	111001 1001	000110 1001
D9.1	001 01001	100101 1001	100101 1001
D10.1	001 01010	010101 1001	010101 1001

Data Byte Name	Bits hgf edcba	Current RD - abcdei fghj	Current RD + abcdei fghj
D11.1	001 01011	110100 1001	110100 1001
D12.1	001 01100	001101 1001	001101 1001
D13.1	001 01101	101100 1001	101100 1001
D14.1	001 01110	011100 1001	011100 1001
D15.1	001 01111	010111 1001	101000 1001
D16.1	001 10000	011011 1001	100100 1001
D17.1	001 10001	100011 1001	100011 1001
D18.1	001 10010	010011 1001	010011 1001
D19.1	001 10011	110010 1001	110010 1001
D20.1	001 10100	001011 1001	001011 1001
D21.1	001 10101	101010 1001	101010 1001
D22.1	001 10110	011010 1001	011010 1001
D23.1	001 10111	111010 1001	000101 1001
D24.1	001 11000	110011 1001	001100 1001
D25.1	001 11001	100110 1001	100110 1001
D26.1	001 11010	010110 1001	010110 1001
D27.1	001 11011	110110 1001	001001 1001
D28.1	001 11100	001110 1001	001110 1001
D29.1	001 11101	101110 1001	010001 1001
D30.1	001 11110	011110 1001	100001 1001
D31.1	001 11111	101011 1001	010100 1001
D0.2	010 00000	100111 0101	011000 0101
D1.2	010 00001	011101 0101	100010 0101
D2.2	010 00010	101101 0101	010010 0101
D3.2	010 00011	110001 0101	110001 0101
D4.2	010 00100	110101 0101	001010 0101
D5.2	010 00101	101001 0101	101001 0101
D6.2	010 00110	011001 0101	011001 0101
D7.2	010 00111	111000 0101	000111 0101
D8.2	010 01000	111001 0101	000110 0101
D9.2	010 01001	100101 0101	100101 0101
D10.2	010 01010	010101 0101	010101 0101
D11.2	010 01011	110100 0101	110100 0101
D12.2	010 01100	001101 0101	001101 0101
D13.2	010 01101	101100 0101	101100 0101
D14.2	010 01110	011100 0101	011100 0101
D15.2	010 01111	010111 0101	101000 0101
D16.2	010 10000	011011 0101	100100 0101
D17.2	010 10001	100011 0101	100011 0101
D18.2	010 10010	010011 0101	010011 0101
D19.2	010 10011	110010 0101	110010 0101
D20.2	010 10100	001011 0101	001011 0101
D21.2	010 10101	101010 0101	101010 0101

Non-Neutral Two
Encodes

Neutral One
Encode

K Codes

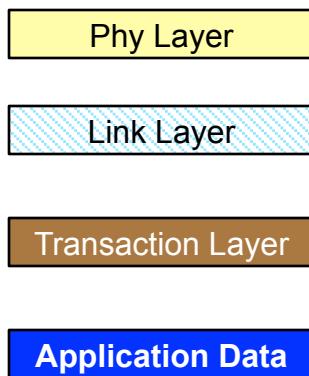
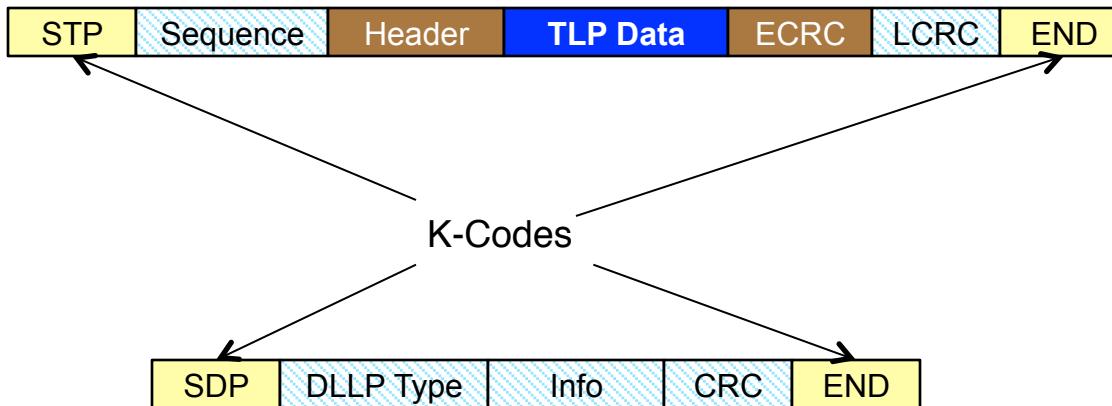
What Are K Codes?

K Codes are:

- Non-data symbols that have defined meaning
- Used for link control

Symbol	Encoding	Name
COM	K28.5	Comma
STP	K27.7	Start TLP
SDP	K28.2	Start DLLP
END	K29.7	End
ENB	K30.7	End Bad
PAD	K23.7	PAD
SKP	K28.0	Skip
FTS	K28.1	Fast Training Sequence
IDL	K28.3	Idle
	K28.4	Reserved
	K28.6	Reserved
EIE	K28.7	Electrical Idle Exit

TLP and DLLP with Link and Phy Additions



Ordered Sets

Gen 1 and 2 Ordered Sets

Name	1 st Char	Other Characters			Description
Skip	COM	SKP	SKP	SKP	Used for lane de-skew and elastic buffer management
EIOS ¹	COM	IDL	IDL	IDL	Enter Electrical Idle
FTS	COM	FTS	FTS	FTS	Fast Training Sequence
EIEOS	COM	K28.7 (* 14)	TS1 ID	Electrical Idle Exit Only above 2.5 Gt/s	
TS1	COM	Defined Later		Training Sequence 1 16 Char set beginning with COM	
TS2	COM	Defined Later		Training Sequence 2 16 Char set beginning with COM	

¹ Sent once for 2.5 GT/s

Sent twice for 5 GT/s

Byte Striping

All transmissions
are multiple of 4
bytes

Lane 0	Lane 1	Lane 2	Lane 3	Lane 4	Lane 5	Lane 6	Lane 7
STP	Sequence	Sequence	TLP	—	—	—	→
—	—	—	—	—	—	—	→
—	—	→	LCRC	LCRC	LCRC	LCRC	END
COM	COM	COM	COM	COM	COM	COM	COM
SKP	SKP	SKP	SKP	SKP	SKP	SKP	SKP
SKP	SKP	SKP	SKP	SKP	SKP	SKP	SKP
SKP	SKP	SKP	SKP	SKP	SKP	SKP	SKP
SDP	DLLP	—	—	—	—	→	END
STP	Sequence	Sequence	—	—	—	—	→
—	—	—	—	—	—	→	LCRC
LCRC	LCRC	LCRC	END	PAD	PAD	PAD	PAD
IDL	IDL	IDL	IDL	IDL	IDL	IDL	IDL
IDL	IDL	IDL	IDL	IDL	IDL	IDL	IDL
IDL	IDL	IDL	IDL	IDL	IDL	IDL	IDL

Scrambling

Scrambling

Scrambling is done to reduce Electromagnetic Interference (EMI) generated by sending the same bit pattern repeatedly.

The LFSRs for all lanes in a link must be synchronized

Scrambling is done before encoding; descrambling is done after decoding

LFSR is initialized on every COM

LFSR is advanced 8 shifts per symbol except SKP

All symbols are scrambled except:

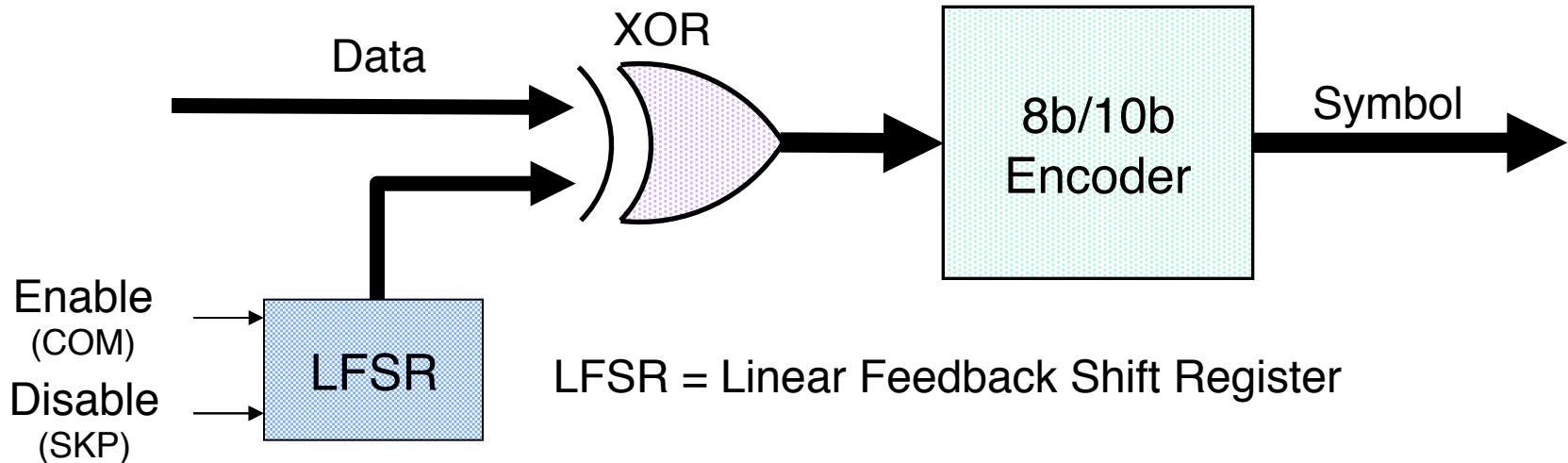
TS1 and TS2 in Gen 1 and 2

DC Balance characters in TS1 and TS2 in Gen 3

Compliance patterns

K codes in Gen 1 and 2

8b/10b Data Scrambling



LFSR polynomial: $G(x) = X^{16} + X^5 + X^4 + X^3 + 1$
Seed = FFFFh

PCIe Generations 3 and 4

8 Gt/s and 16 Gt/s

128b/130b encoding



Improvements in Gen 3 and Gen 4

Encoding yields 25% gain over Gen 1/2

Requires more robust receivers

Requires improved scrambler

Allows elimination of K codes

More control of signal de-emphasis

Information Types in Gen 3 and Gen 4

Data Stream

Start with SDS Ordered Set;
Includes TLPs, DLLPs, and Framing Tokens
Made up of Data Blocks
 16 bytes preceded by sync header of 10b
Ends with EDS Framing Token

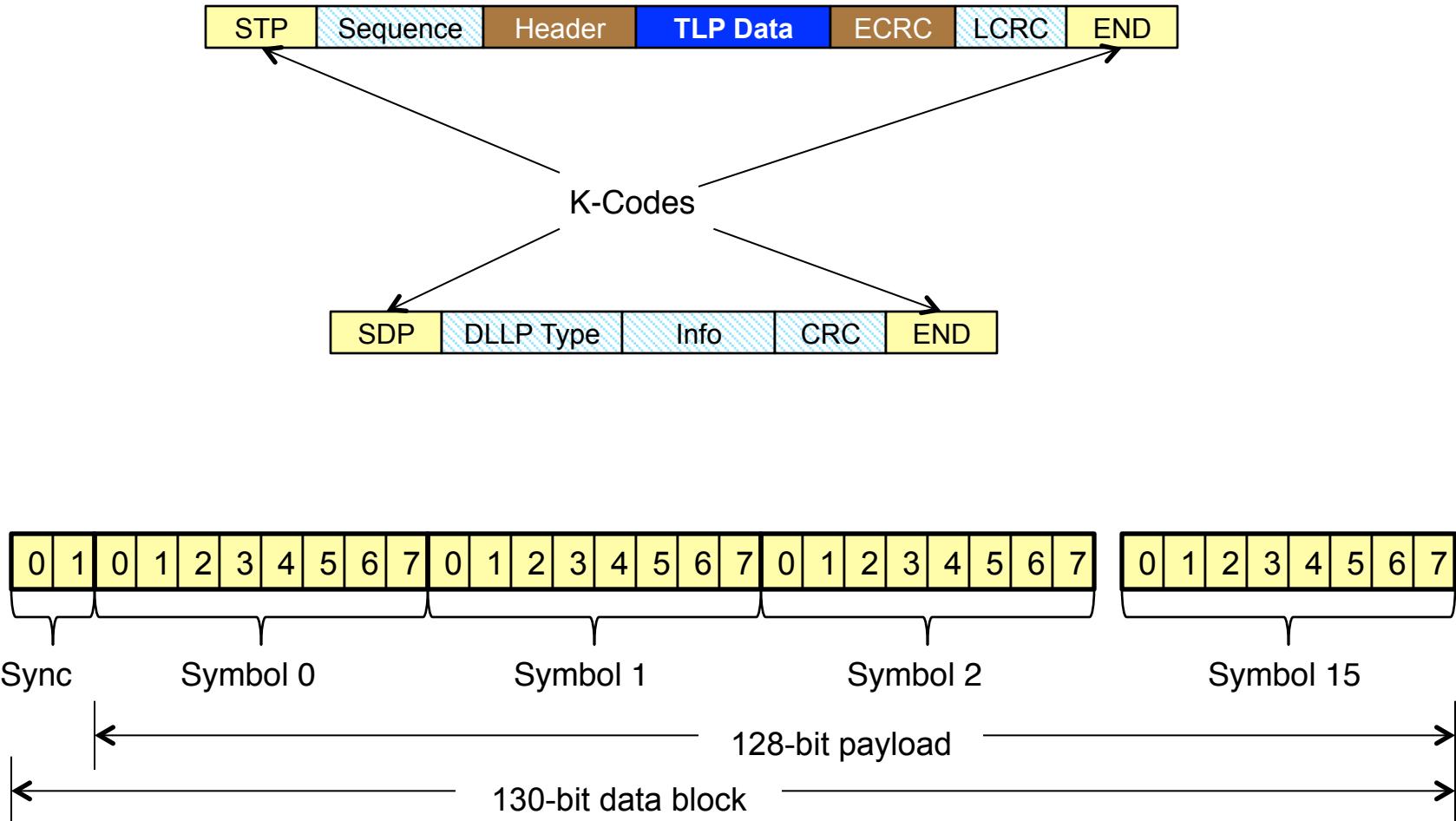
Ordered Sets

Originates from Link or Physical layer
16 bytes preceded by sync header of 01b

Framing Token

Originates from Physical layer
Variable length
Part of Data Steam

Gen 1 and 2 vs. Gen 3 and Gen 4



Data Stream and Ordered Sets - Text

Data Streams begin with SDS Ordered Set
and end with EDS Token

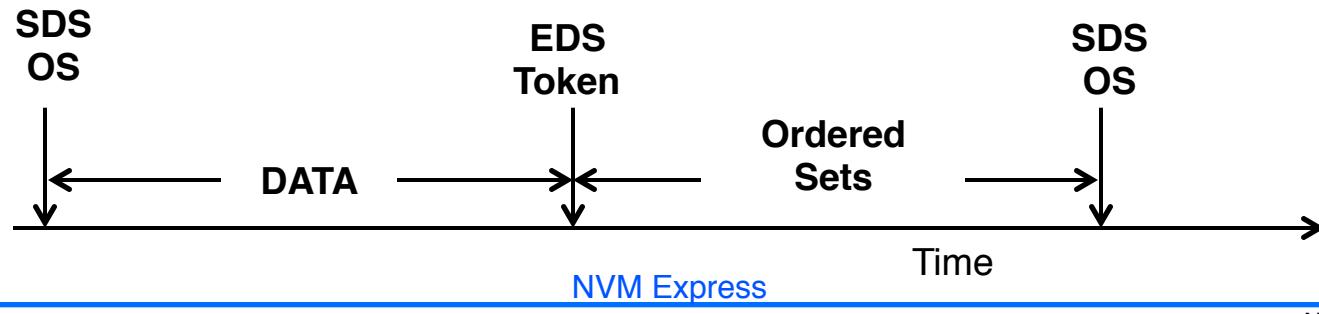
Data Streams include:

- TLP
- DLLP
- Tokens
- Sync Headers

Ordered Sets begin with EDS Token
and end with SDS Ordered Set

Ordered Sets include:

- Physical Layer Control



Data Stream Lane Usage

	Lane 0	Lane 1	Lane 2	Lane 3	Lane 4	Lane 5	Lane 6	Lane 7
	SDS OS	SDS OS	SDS OS	SDS OS	SDS OS	SDS OS	SDS OS	SDS OS
Sync Header	01	01	01	01	01	01	01	01
Symbol 0	<	STP	><			TLP Header DW 0		>
Symbol 1	<			TLP Header DW 1 and 2				>
Symbol 2	<	TLP Header DW 3		><		TLP Payload 1 DW		>
Symbol 3	<	LCRC		><	SDP	><	DLLP Payload	>
Symbol 4	<	DLLP Payload	<>	CRC	>	IDL	IDL	IDL
Symbol 5	IDL	IDL	IDL	IDL	IDL	IDL	IDL	IDL
Symbol 6	<	STP	><			TLP Header DW 0		>
Symbol 7	<			TLP Header DW 1 and 2				>
Symbol 8	<	TLP Header DW 3		><		TLP Payload DW 0		>

Symbol 15	<	TLP Payload DW 13	><		TLP Payload DW 14		>	
Sync Header	01	01	01	01	01	01	01	01
Symbol 0	<	TLP Payload DW 15	><		TLP Payload DW 16		>	
Symbol 1	<	LCRC	>	IDL	IDL	IDL	IDL	IDL

Payload Byte	X	X+1	X+2	X+3	X+4	X+5	X+6	X+7
--------------	---	-----	-----	-----	-----	-----	-----	-----

NVM Express

Scrambling

Scrambling

Scrambling is done to reduce Electromagnetic Interference (EMI) generated by sending the same bit pattern repeatedly.

Scrambling is done on a per lane basis,
but the LFSRs for all lanes in a link must be synchronized

Notes that will be meaningful when we cover initialization in the next Section:

LFSR is initialized on the last symbol of EIEOS

Header bits are not scrambled

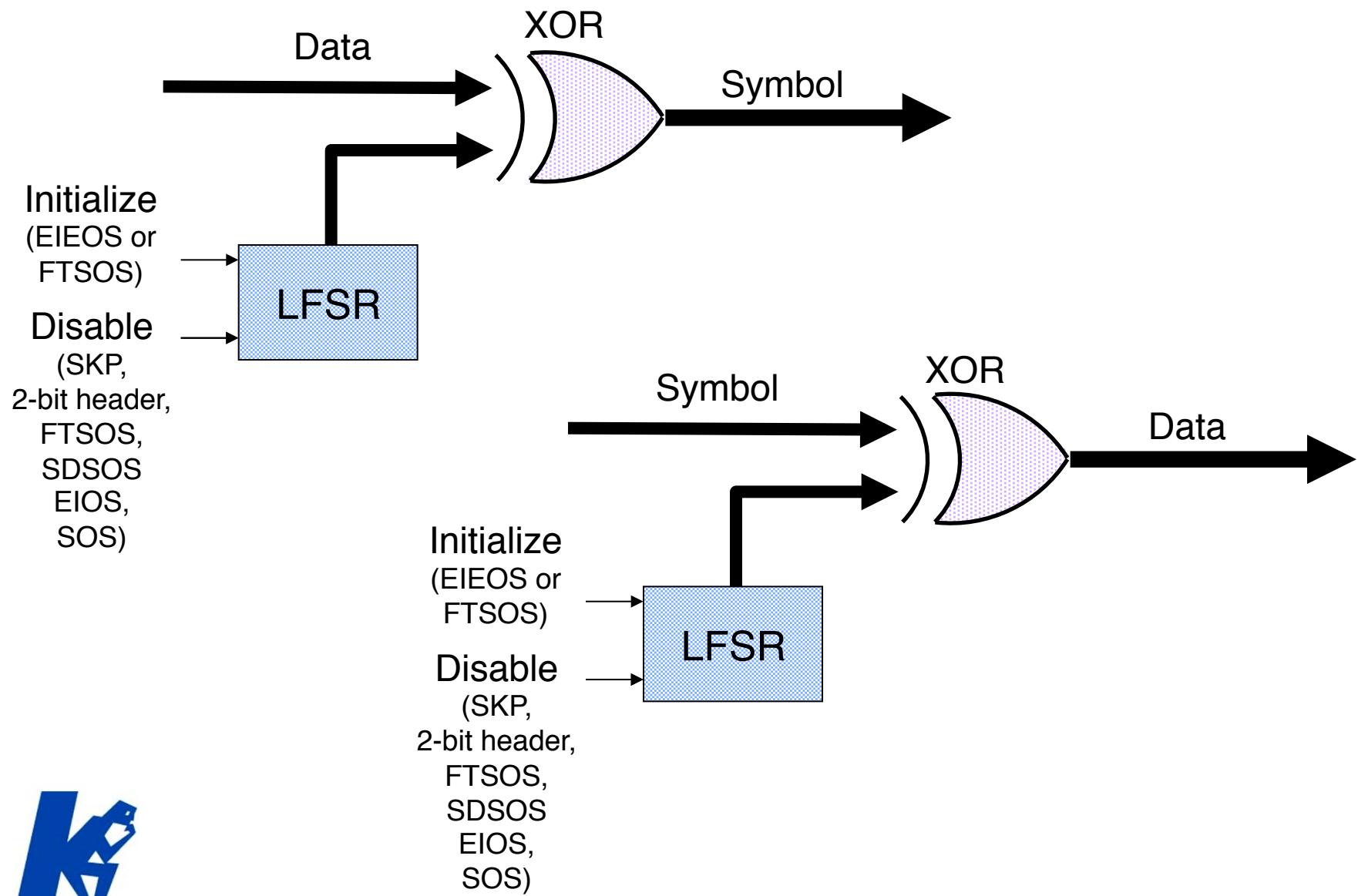
All 16 symbols of EIOS, EIEOS, FTS, SDS and SKP Ordered Sets bypass scrambling

TS1 and TS2 are scrambled except Symbol 0 and

Symbols 14 and 15 if not needed for DC balance

All data symbols are scrambled

Data Scrambling/De-Scrambling



8b/10b vs. 128b/130b Scrambling

8b/10b encoding guaranteed:

- DC balance at the end of each 10-bit symbol

- Adequate bit transitions for clock recovery

128b/130b encoding solutions:

- New scrambling algorithm

- TS1 and TS2 Ordered Sets can adjust DC balance

- Receivers must tolerate up to 511 bits of DC imbalance

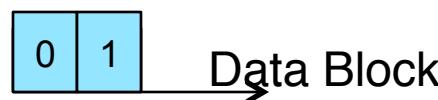
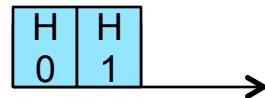


Sync Symbols

In the documentation

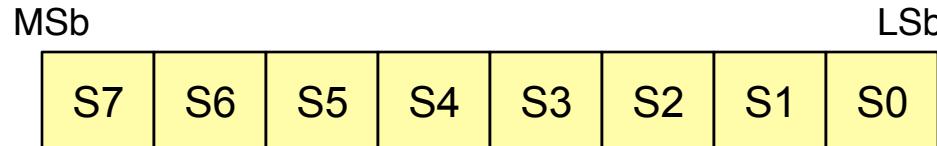
H1	H0	Description
0b	0b	Reserved
0b	1b	Ordered Set Block
1b	0b	Data Block
1b	1b	Reserved

On the wire

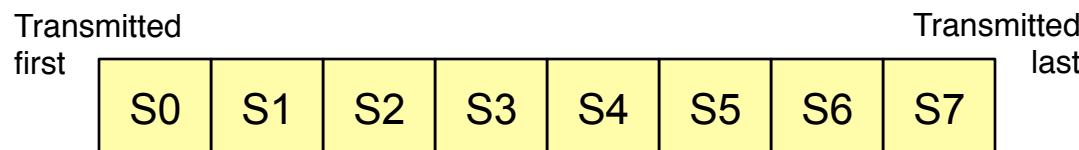


Information Symbols

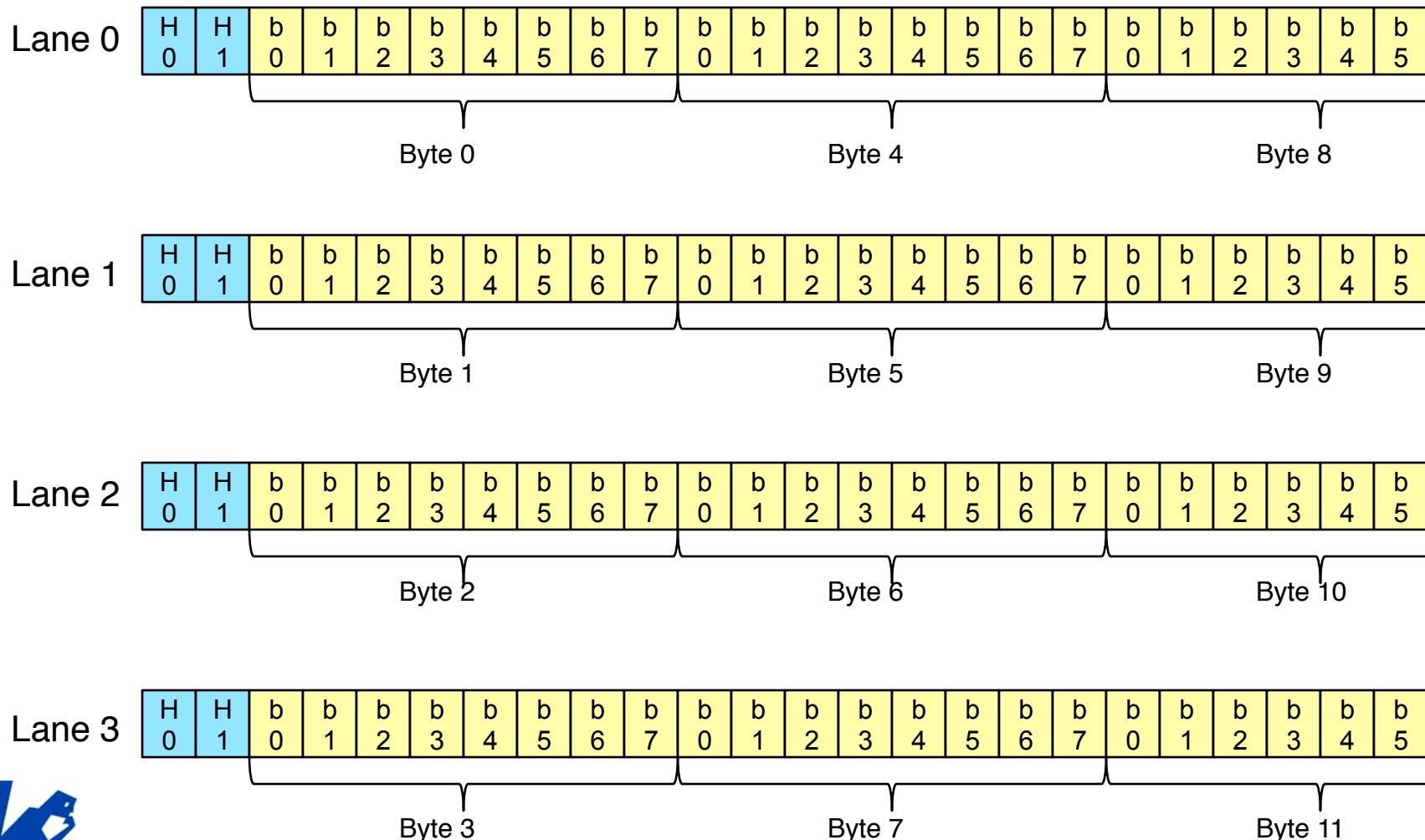
In the documentation



On the wire



Symbol Placement on Multi-Lane Configurations



Tokens



Framing Tokens

Symbols	Format	Length	Description
STP	Fh + Length in Dwords[10:0]	4 Bytes	Start Transaction Packet Indicates that the TLP information follows, includes TLP Length
SDP	0F35h	2 Bytes	Start of DLLP indicates the DLLP follows
EDS	F801 0900h	4 Bytes	End Data Stream Begin Ordered Sets
EDB	0303 0303h	4 Bytes	End Bad Nullify packet Confirms that the previous TLP was nullified
IDL	00h	1 Bytes	Logical Idle No packets to send

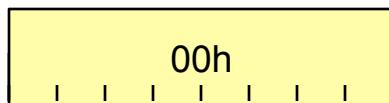
 Note: There is no “good end” token. STP token has length field
Assume it was good if not signaled as bad

NVM Express

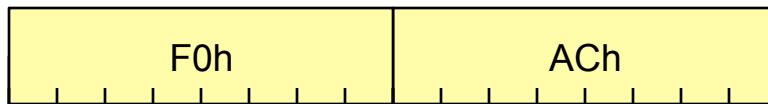
Note: this shows format on the wire

Framing Tokens - Contents

Example:



Logical Idle Framing Token



SDP Framing Token

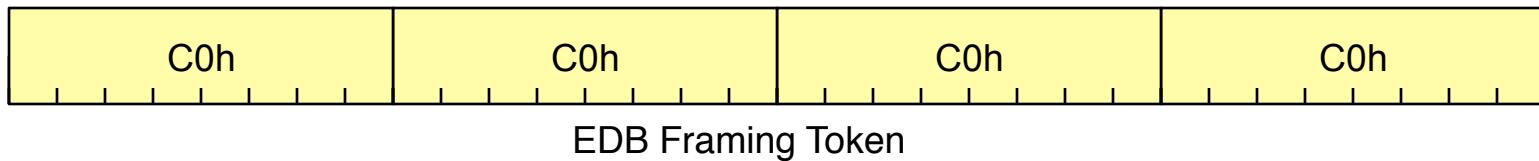
Bits	7	6	5	4	3	2	1	0	Documentation
	1	1	1	1	0	0	0	0	1 0 1 0 1 1 0 0
Value	1	1	1	1	0	0	0	0	1 0 1 0 1 1 0 0

Bits	0	1	2	3	4	5	6	7	On the wire
	0	0	0	0	1	1	1	1	0 1 2 3 4 5 6 7
Value	0	0	0	0	1	1	1	1	0 0 1 1 0 1 0 1

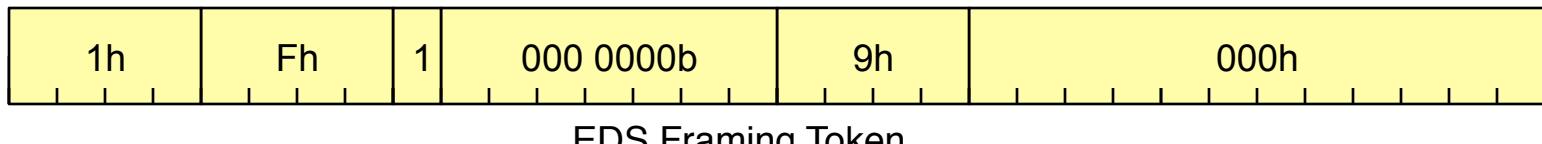


STP Framing Token

FP = Frame Parity (even); FCRC is 4 bit CRC for TLP Length



EDB Framing Token



EDS Framing Token

NVM Express

The figure shows a network protocol analyzer interface with the following details:

- Protocol View:** Shows two main sections: "S-T-P" and "S".
- Details Pane:** Displays the structure of a TLP frame with sequence number 0x576. The fields include:
 - Header:** Fmt (2 bytes), Type (0A CplD).
 - Requester ID:** 0000.
 - Completion Status:** 0 (Successful Completion).
 - Byte Count:** 040.
 - Completer ID:** 0000.
 - Command:** OPC (05 Create I/O Completion Queue).
 - Requester ID:** 0100.
 - Tag:** 00.
 - Lower Address:** 40.
- Command:** OPC (05 Create I/O Completion Queue) with fields:
 - Fused Operation: 0 (Normal operation).
 - Reserved: 00.
 - Command Identifier: 0005.
 - NSID: 00000000.
 - Reserved: (multiple dots).
- Packet View:** Shows the raw hex, decimal, and ASCII representation of the captured frames. Red arrows point to specific bytes in the hex dump corresponding to the fields in the Details pane.
- Bottom Navigation:** Includes tabs for Protocol View, Spreadsheet View, Transaction View, Lane View, and a search bar.

Framing Requirements

TLP

Begin with STP

Entire contents must follow, even if nullified

If nullified, EDB must immediately follow last Dword

EDB is not included in TLP length

DLLP

Begin with SDP

Entire contents must follow

SOS (Skip Ordered Set – defined later in this chapter)

End previous data stream with EDS

Send SOS as Ordered Set

Send another data block immediately after SOS, this resumes data steam

Multiple SOS cannot be back-to-back as required in Gen 1 and Gen 2

Data blocks between SOS can be TLP, DLLP or IDL

Multi-Lane Links

After Logical Idle token, 1st symbol of next TLP or DLLP must be in Lane 0

Else, TLP and DLLP can start in lane number divisible by 4

Ordered Sets

Gen 3/4 Ordered Sets

Only used in Physical Layer, not passed up

Appear on all lanes of a link at the same time

Consist of 16 bytes,
except SKP which may have 8, 12, 16, 20, or 24

Sync is 10b (little endian) on the wire

Gen 3/4 Ordered Sets

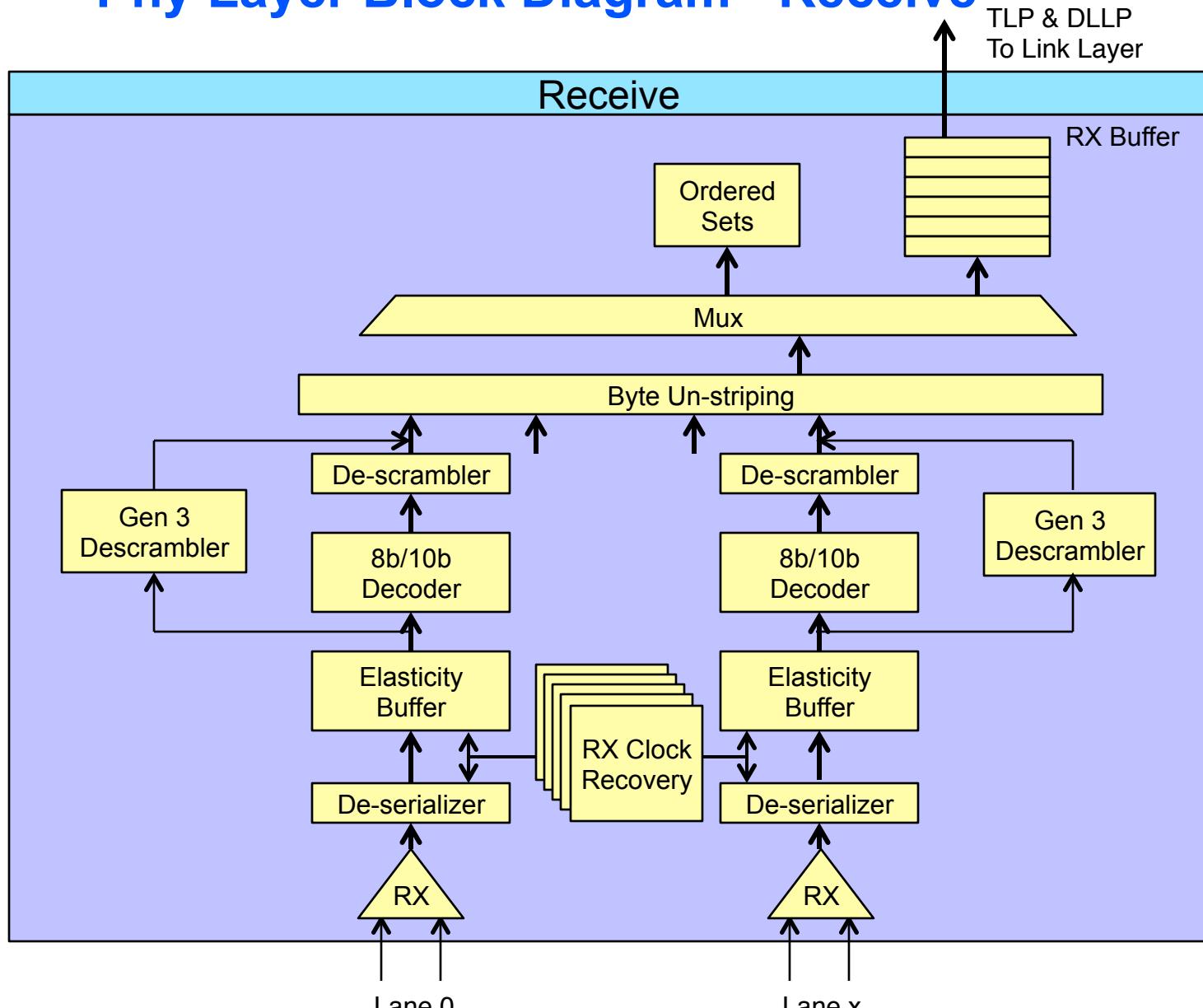
Ordered Set	Identifier	Description
EOS	66h	Electrical Idle Ordered Set Used to enter Electrical Idle
EIEOS	00h	Electrical Idle Exit Ordered Set Used to: Exit electrical Idle, and achieve block alignment
FTS	55h	Fast Training Ordered Set Used to transition from L0s to L0 state
SDS	E1h	Start of Data Stream Used to indicate new data stream
SOS	AAh	SKP Ordered Set Used for clock compensation
TS1	1Eh	Training Sequence 1 Used for: Phy training and communicating phy capabilities
TS2	2Dh	Training Sequence 2 Used for: Phy training and Communicating phy capabilities



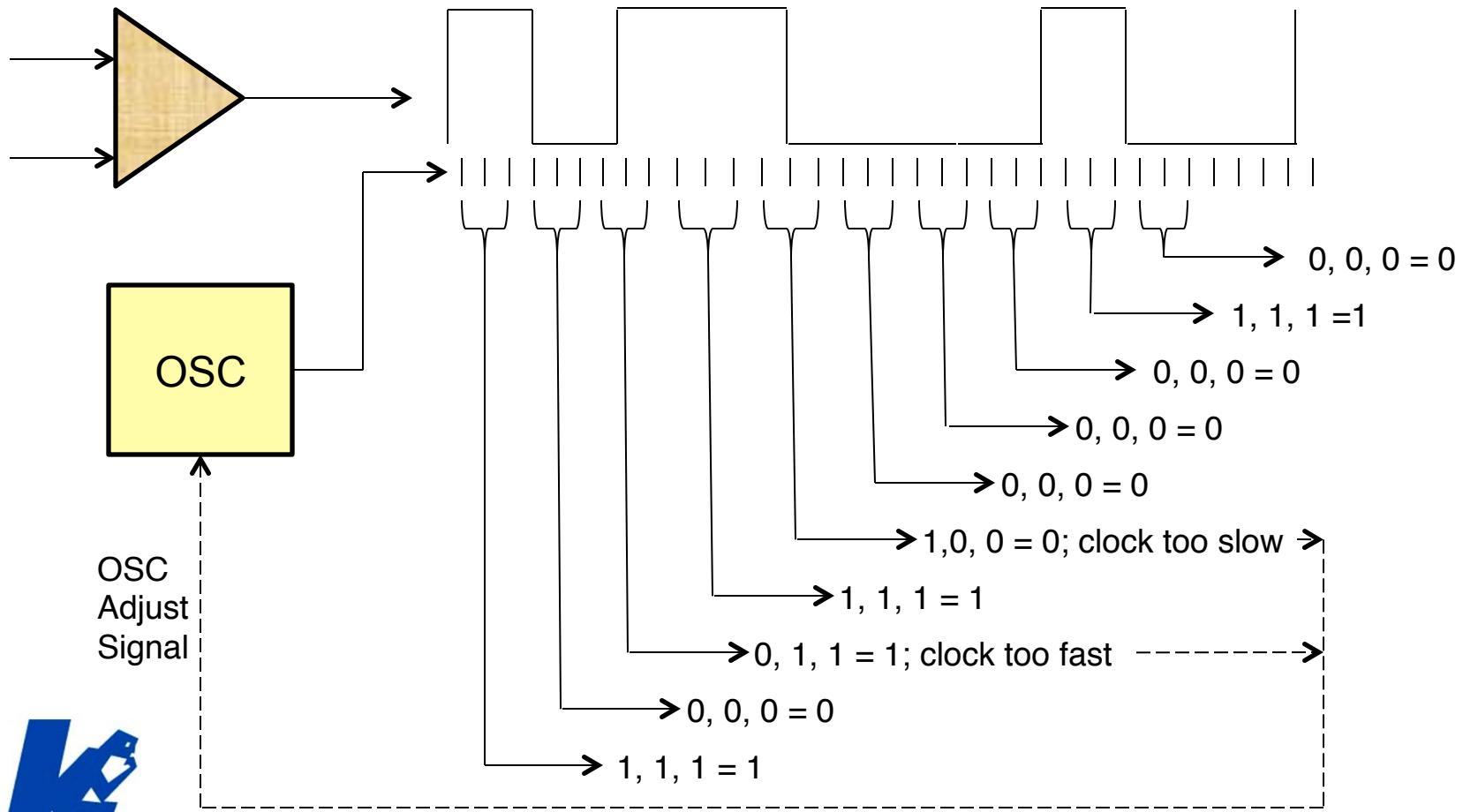
Physical Layer Receive Logic



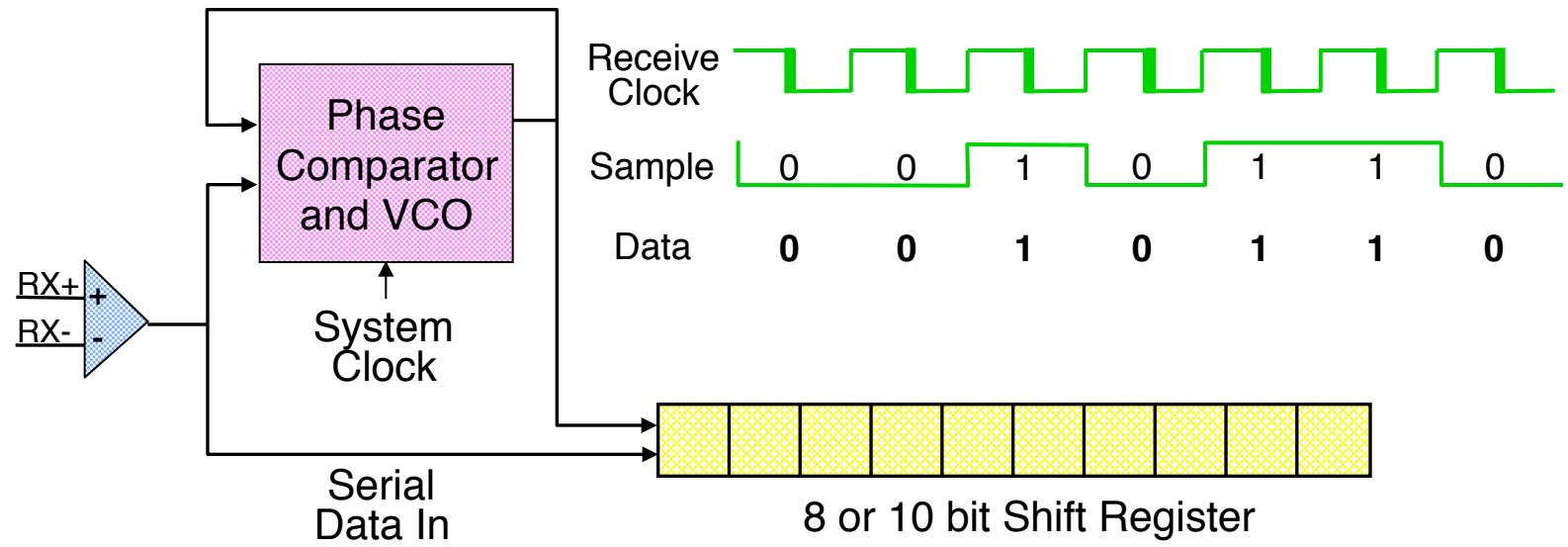
Phy Layer Block Diagram - Receive



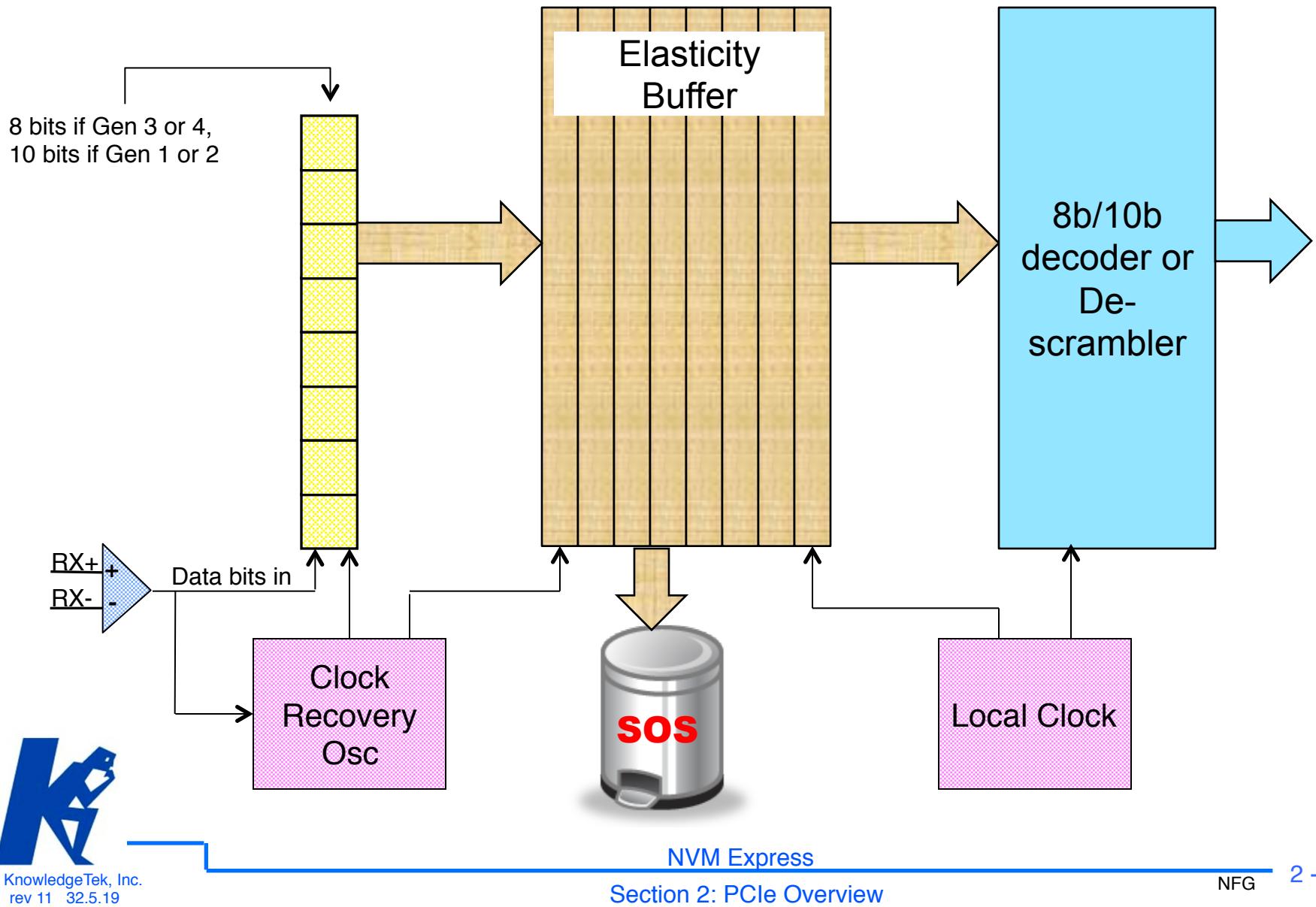
Clock Recovery – 3x Oversampling



Clock Recovery – Tracking



Elasticity Buffer



TX - RX Clock

Serial output of each lane is clocked using the TX Clock

Common Reference Clock architectures introduce no difference between TX and RX

Separate Reference Clock, no SSC (SRNS)

Accurate to +/- 300 ppm

TX and RX can be off by 600 ppm

Clock can gain or lose 1 clock period every 1666 clocks

Separate Reference Clock with Independent SSC (SRIS)

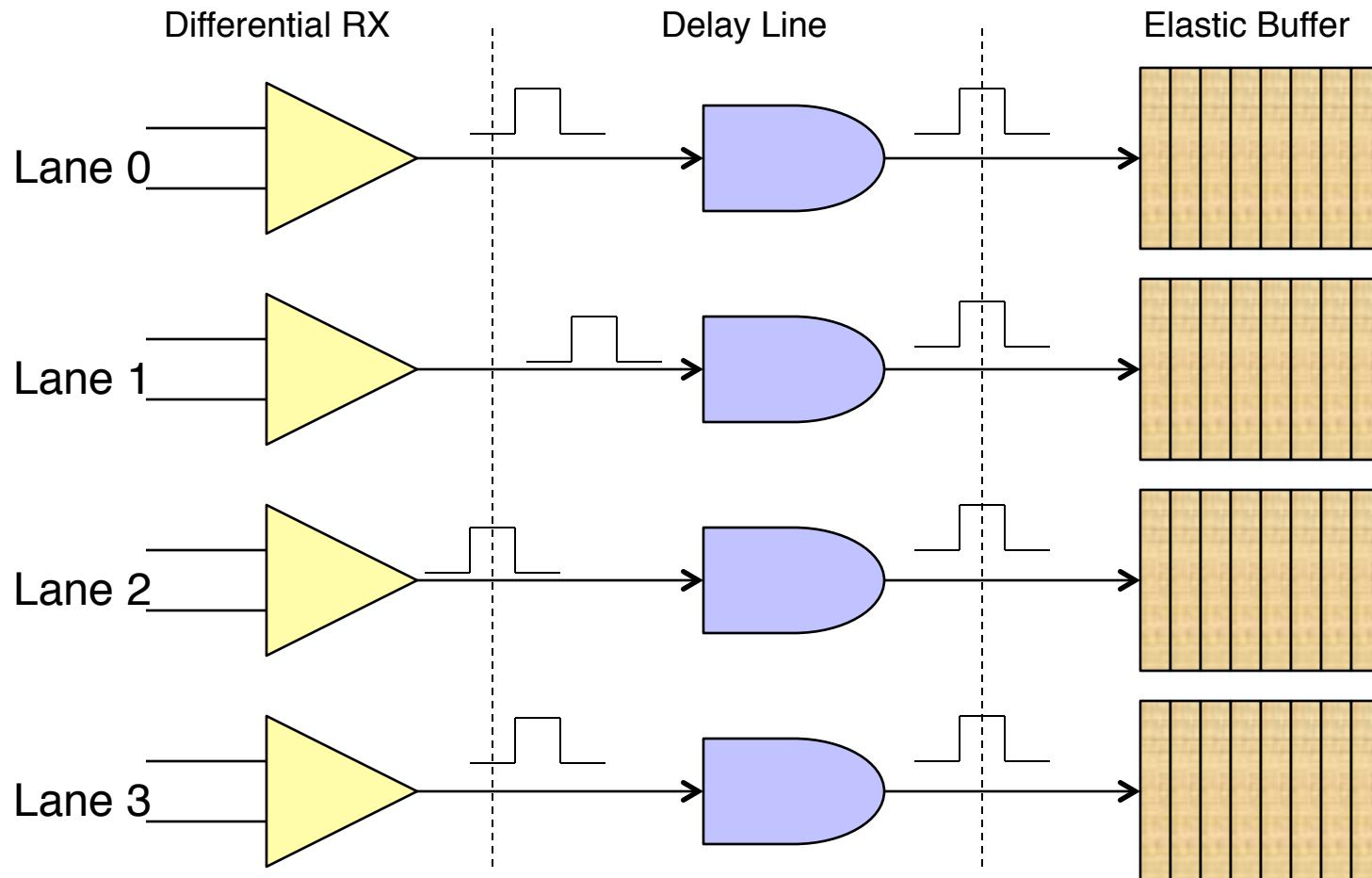
Added in PCIe 3.1

Clocks can be different by 5600 ppm

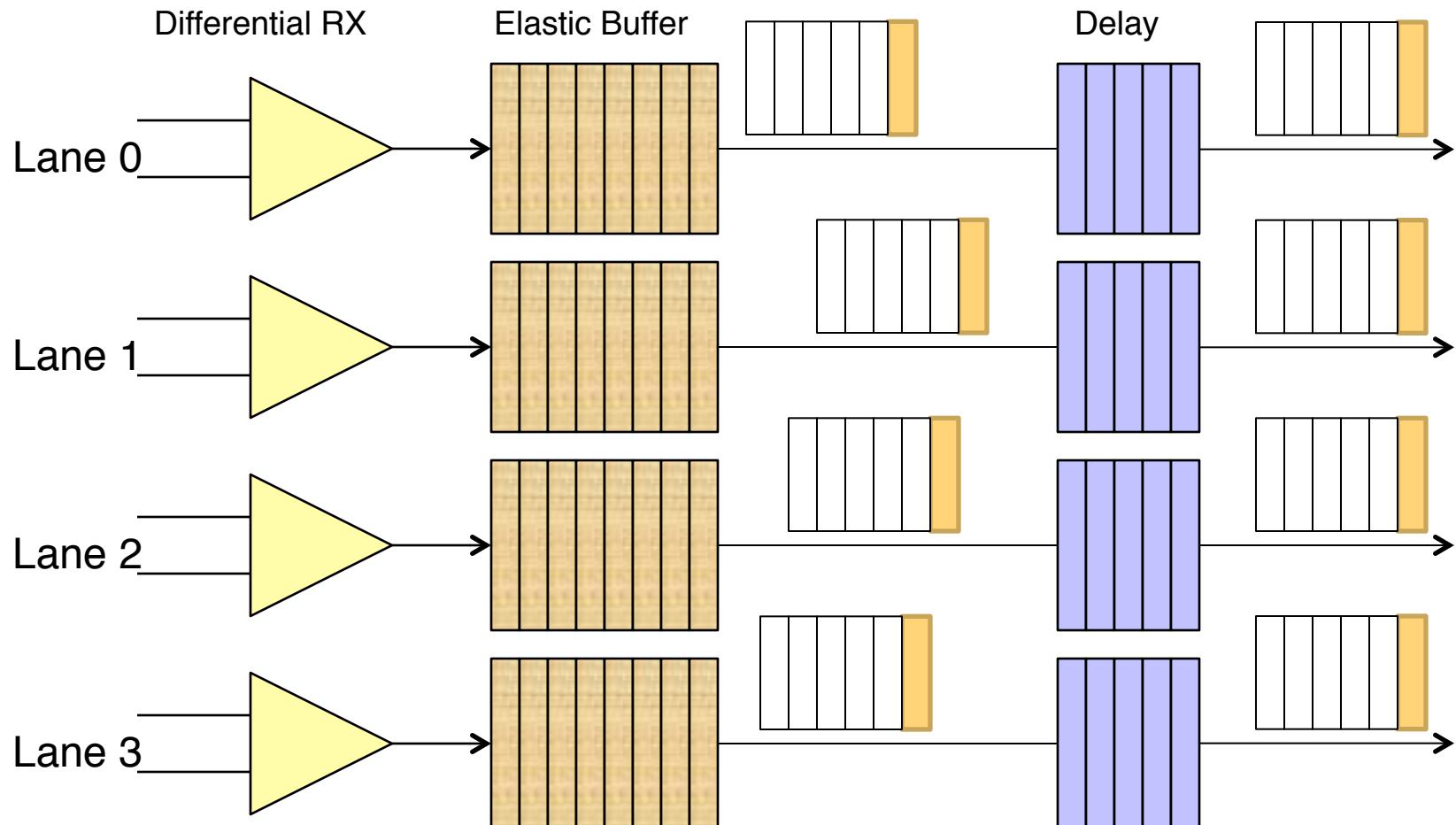
Clocks can gain or lose 1 clock period every 178 clocks

May need more buffer slots in elasticity buffer

Lane De-Skew – Analogue Style



Lane De-Skew – Digital Style



Flow Control



Covered in this Section

Flow control in general

Flow control specifics

Flow control initialization

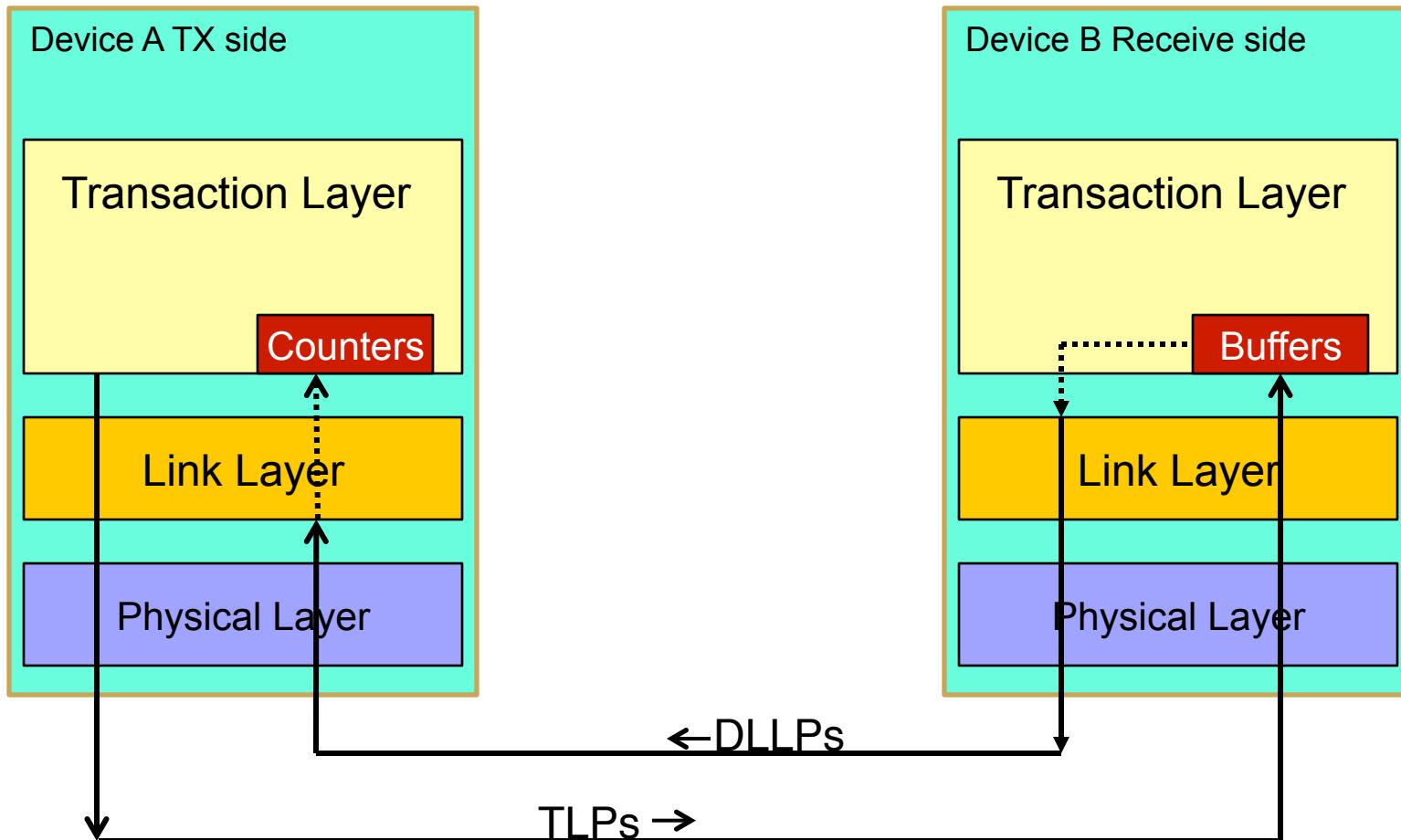
What is Flow Control?

A method of ensuring the receiver has enough space to receive the transmit frames

Why use Flow Control?

- A. To ensure frames are not lost in transit
- B. To prevent buffer overflow
- C. To eliminate time wasting retries

Flow Control Overview



How does Credit Based Flow Control Work?

Receiver tells transmitter how many buffer units are available

Transmitter keeps track of how many units have been sent

Before sending any more units, transmitter compares buffers available minus units sent. If there is room in the receive buffer for the additional transmit units, then send them

Flow Control PCIe Specific

How much is One Buffer Unit in Flow Control?

Data	4 DWords
Header	Maximum-size header + TLP Digest + End to End Prefixes

How Many Buffers does PCIe Track?

6 per virtual channel

 Posted Header

 Posted Data

 Non-posted Header

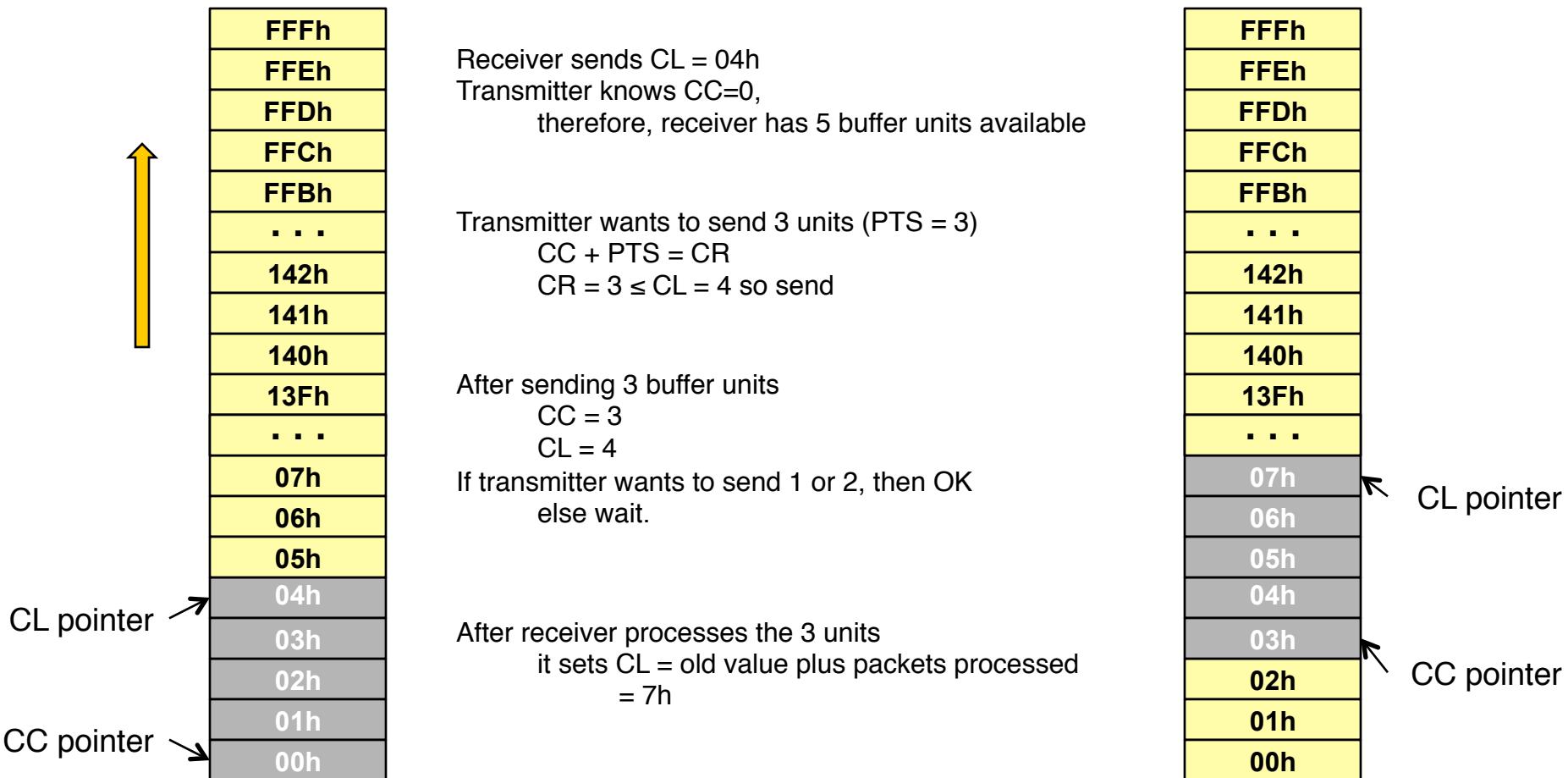
 Non-posted Data

 Completion Header

 Completion Data

Draw Me a Picture of Initialization!

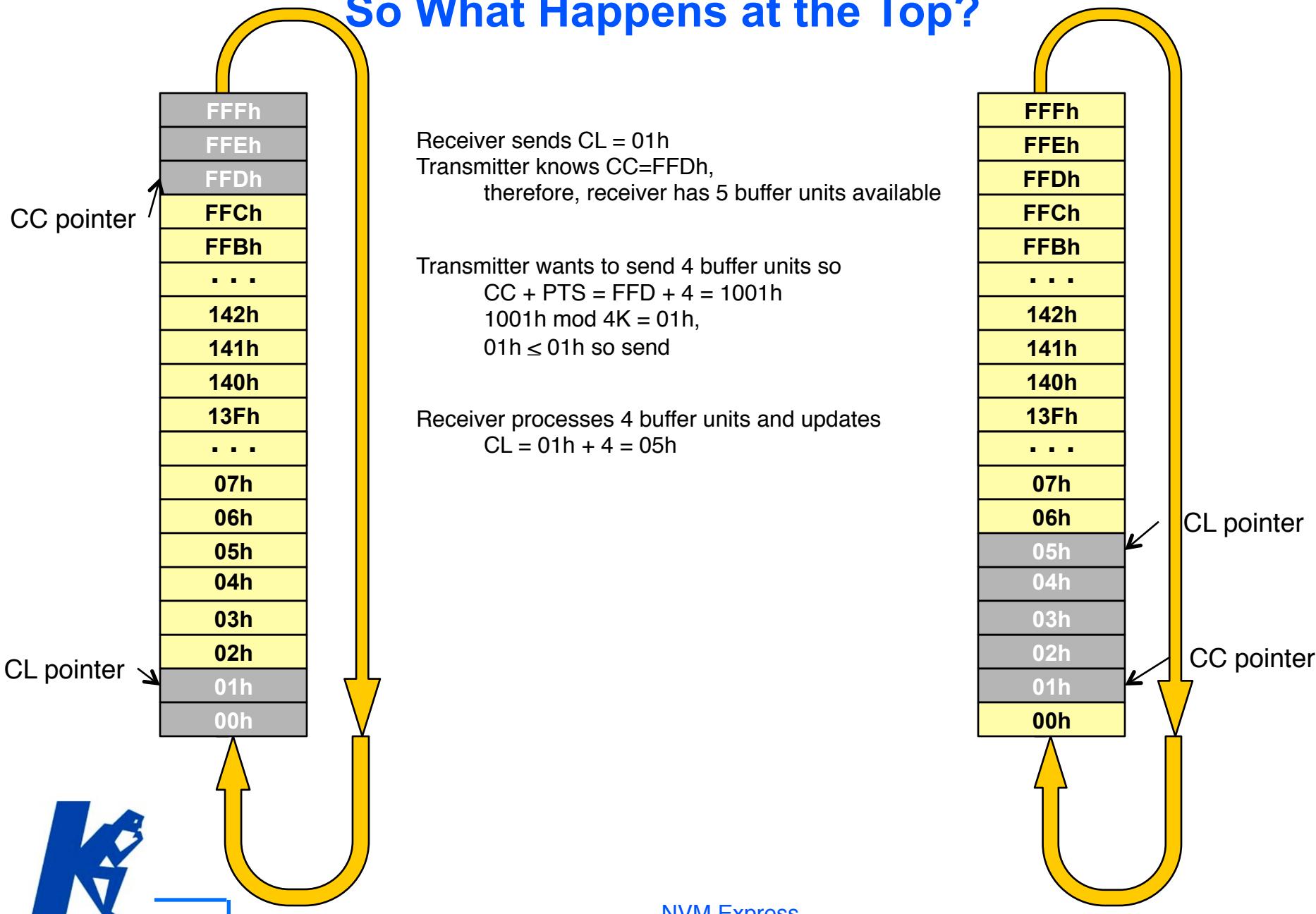
Buffer Addressing Space



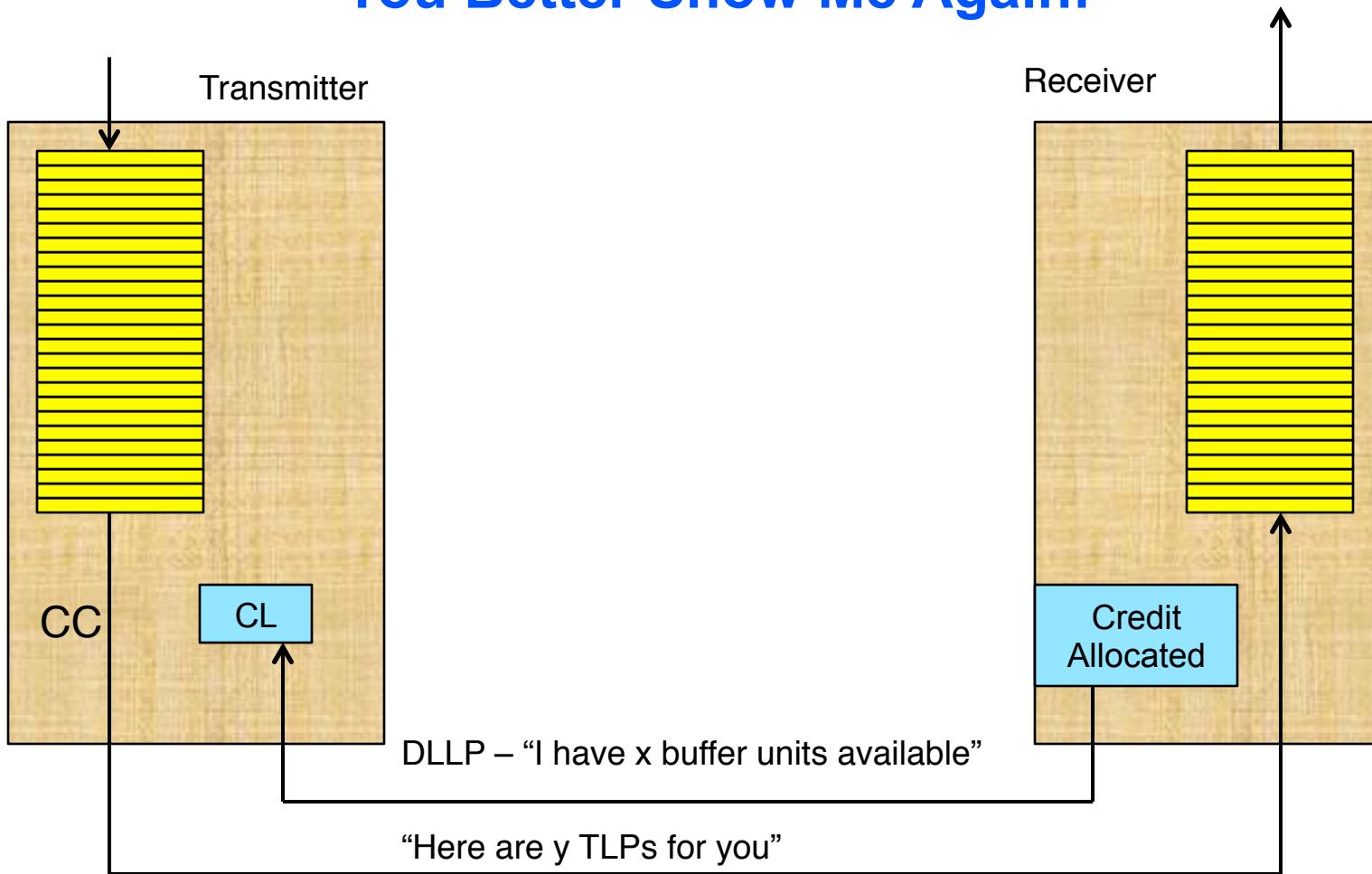
CL pointer value – passed from receiver as a buffer slot
 CC pointer value – Credits Consumed

Available Buffer

So What Happens at the Top?



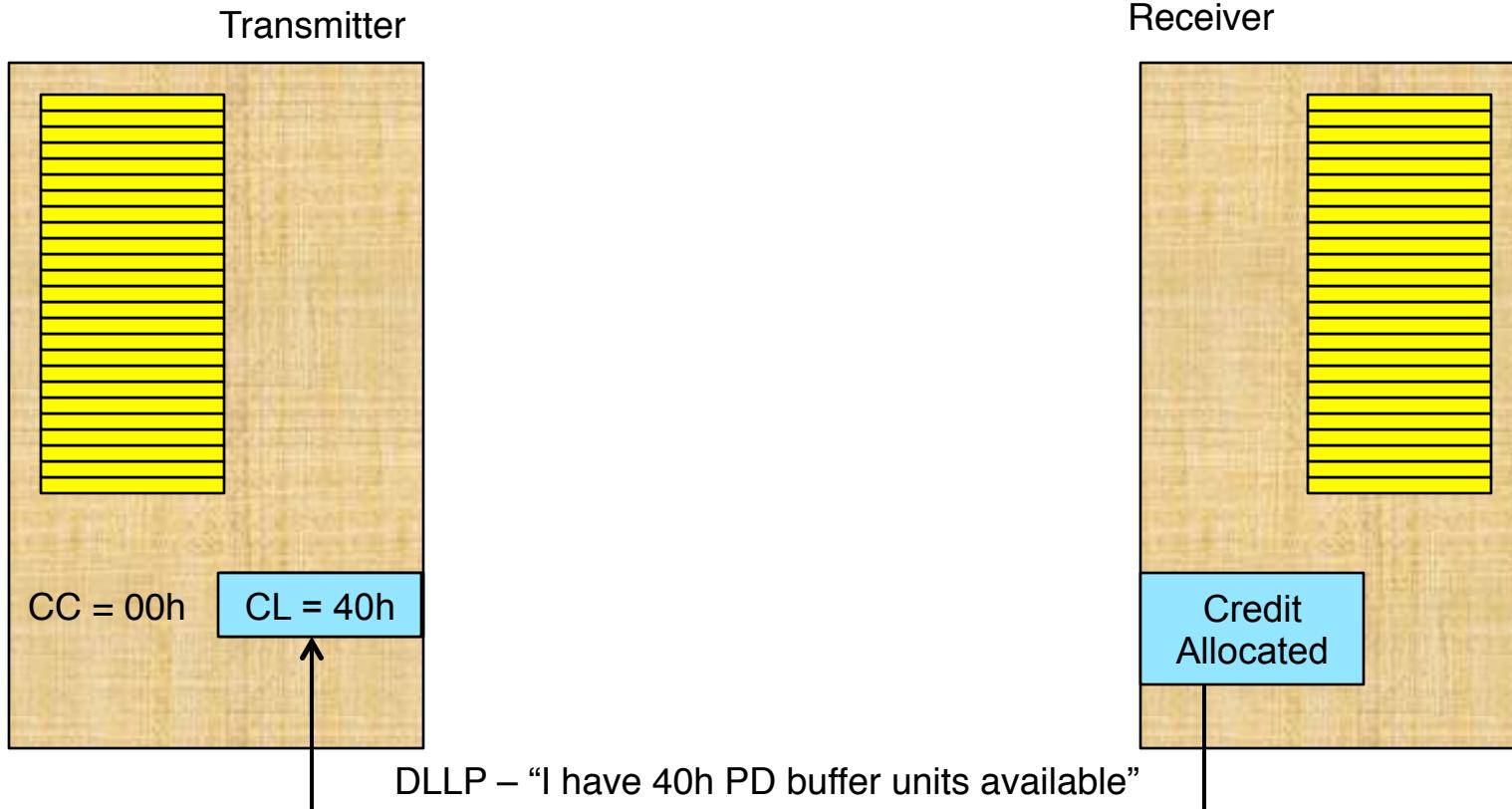
You Better Show Me Again!



CC – Credits Consumed, number of flow control units transmitter has sent to receiver

CL – Credit Limit, number of units receiver has allocated for RX buffers

Initialization



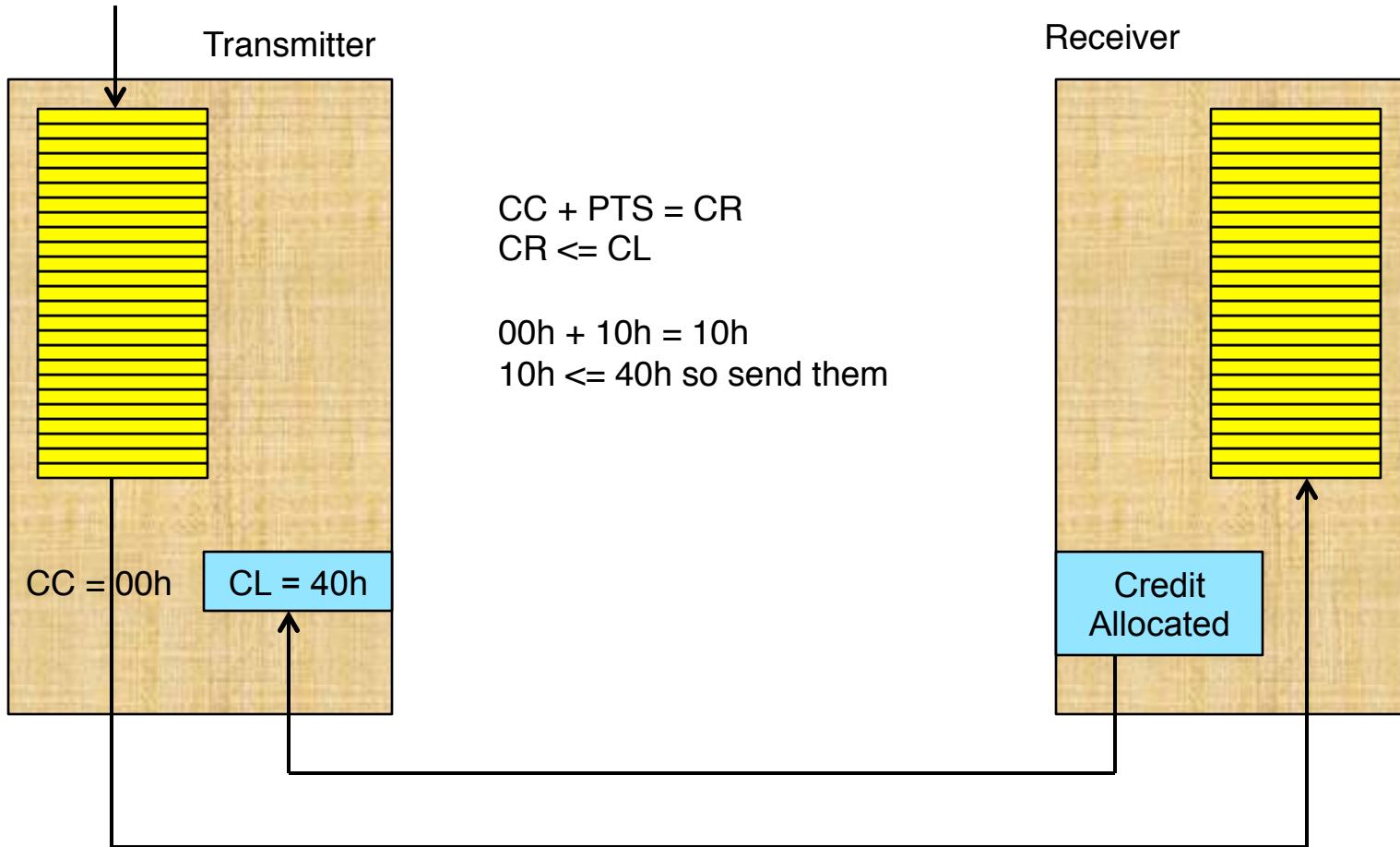
CC – Credits Consumed, number of units transmitter has filled in receiver

CL – Credit Limit, number of units receiver has allocated for RX buffers

PD – Posted Data

Case 1 – Send

Please send these 10h FC units



CC – Credits Consumed, number of units transmitter has sent to receiver

CL – Credit Limit, number of units RX has allocated for RX buffers

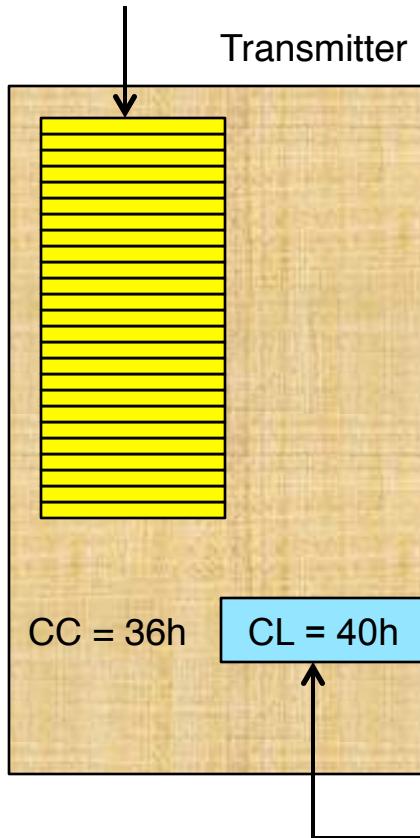
PTS – FC Units to Send

CR – Credit Required

FC – Flow Control

Case 2 – Don't sent

Please send these 10h FC units



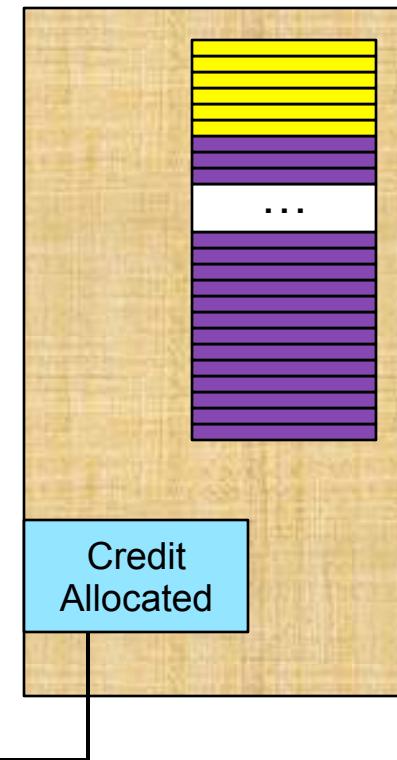
$$CC + PTS = CR$$

$$CR \leq CL$$

$$36h + 10h = 46h$$

46h > 40h not enough space

Receiver



— Buffer slot available

— Buffer slot not available

CC – Credits Consumed, number of units transmitter has sent to receiver

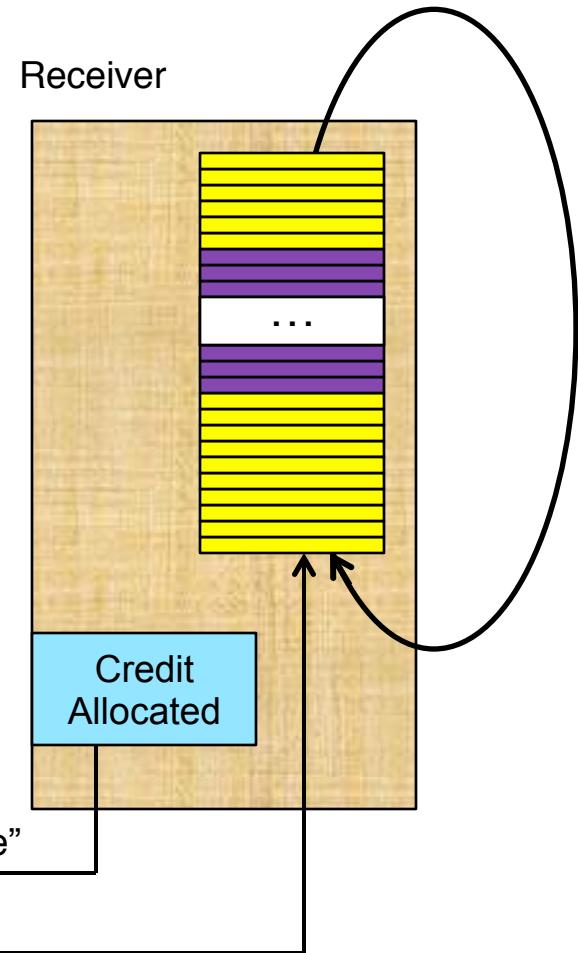
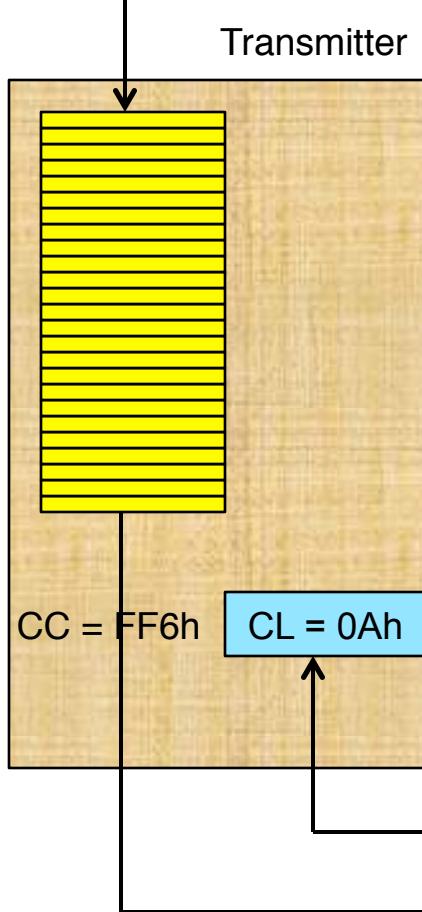
CL – Credit Limit, number of units RX has allocated for RX buffers

PTS – FC Units to Send

CR – Credit Required

Case 3 – Wrap

Please send these 10h FC units



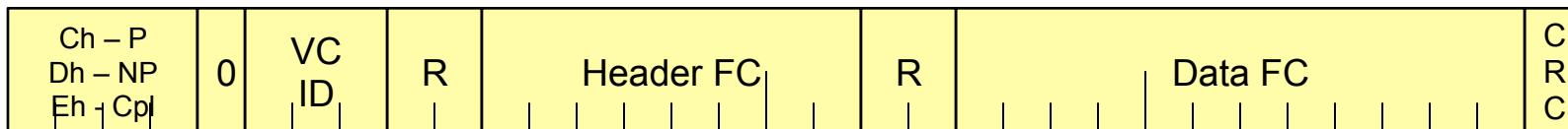
Formula in specification is:
 $(CL - CR)mod2^{[\text{field size}]} \leq 2^{[\text{field size}]} / 2$

Flow Control DLLP Details

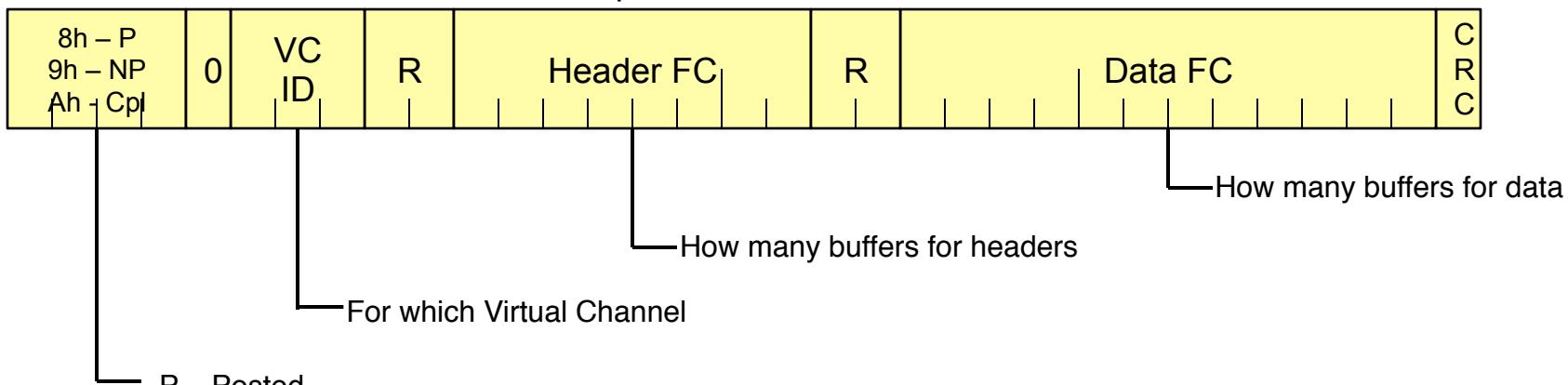
Init FC1 DLLP



Init FC2 DLLP



Update DLLP

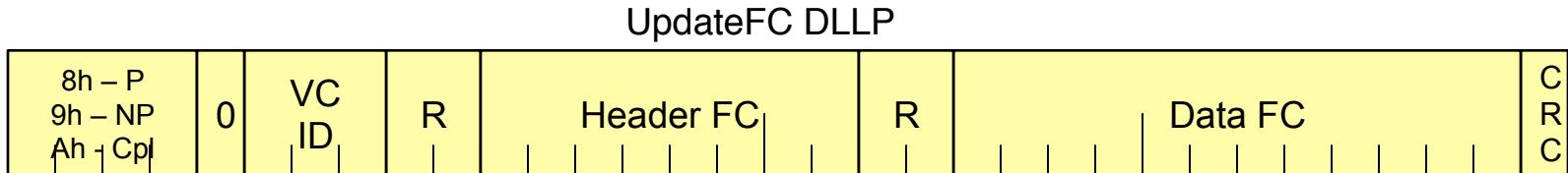


P – Posted

NP – Non-Posted

Cpl – Completion

Flow Control DLLP Details



Scheduling UpdateFC DLLP

If credit is infinite, no updateFC DLLP is sent

NPH, NPD, PH, CPLH

When credits = 0 or

When NPD credit < 2 and receiver supports Atomic OP routing

One or more units are made available

PD, CPLD

Number of Available credits < Max payload size

Or more frequently as desired

If the Link is in L0 or L0s state

30 us (-0% to +50%)

120 us (-0% to +50%) if Extended Sync bit in Control Link register is set

What is the Minimum Advertised Flow Control?

Credit Type	Minimum Advertisement
Posted Header	1 Unit
Posted Data	Largest possible setting of Max Payload size divided by Flow Control Unit size. (for example: 1024 bytes/16 = 40h)
Non-posted Header	1 Unit
Non-Posted Data	Receiver that supports Atomic OP: 2 Units All other receivers: 1 Unit
Completion Header	Root Complex (supporting peer-to-peer traffic) and switch: 1 Unit Root Complex (not supporting peer-to-peer traffic) and endpoints: Infinite (initial value = 00h)
Completion Data	Root Complex (supporting peer-to-peer traffic) and switch: Largest possible setting of Max Payload size divided by Flow Control Unit size. Root Complex (not supporting peer-to-peer traffic) and endpoints: Infinite (initial value = 00h)



Check for Understanding

1. Why do we implement flow control?
2. For each flow control unit granted by the receiver, how much information may the transmitter send?
3. What is the minimum number of flow control units allowed?
What is the maximum number of flow control units allowed?
4. What is the difference between Init1 FC and Init2 FC?
5. When does the flow control initialize?

Check for Understanding

1. What are the PCI/PCI-X and PCIe Transactions?
2. What are the three methods of addressing?
3. Where is the destination address of a packet placed?
4. How does the software locate the memory space for a device?
5. How does the software discover a device's capability?
6. Explain MSI operation.
7. Explain MSI-X operation.

ACK/NAK

Covered in this Section

ACK – NAK Protocol

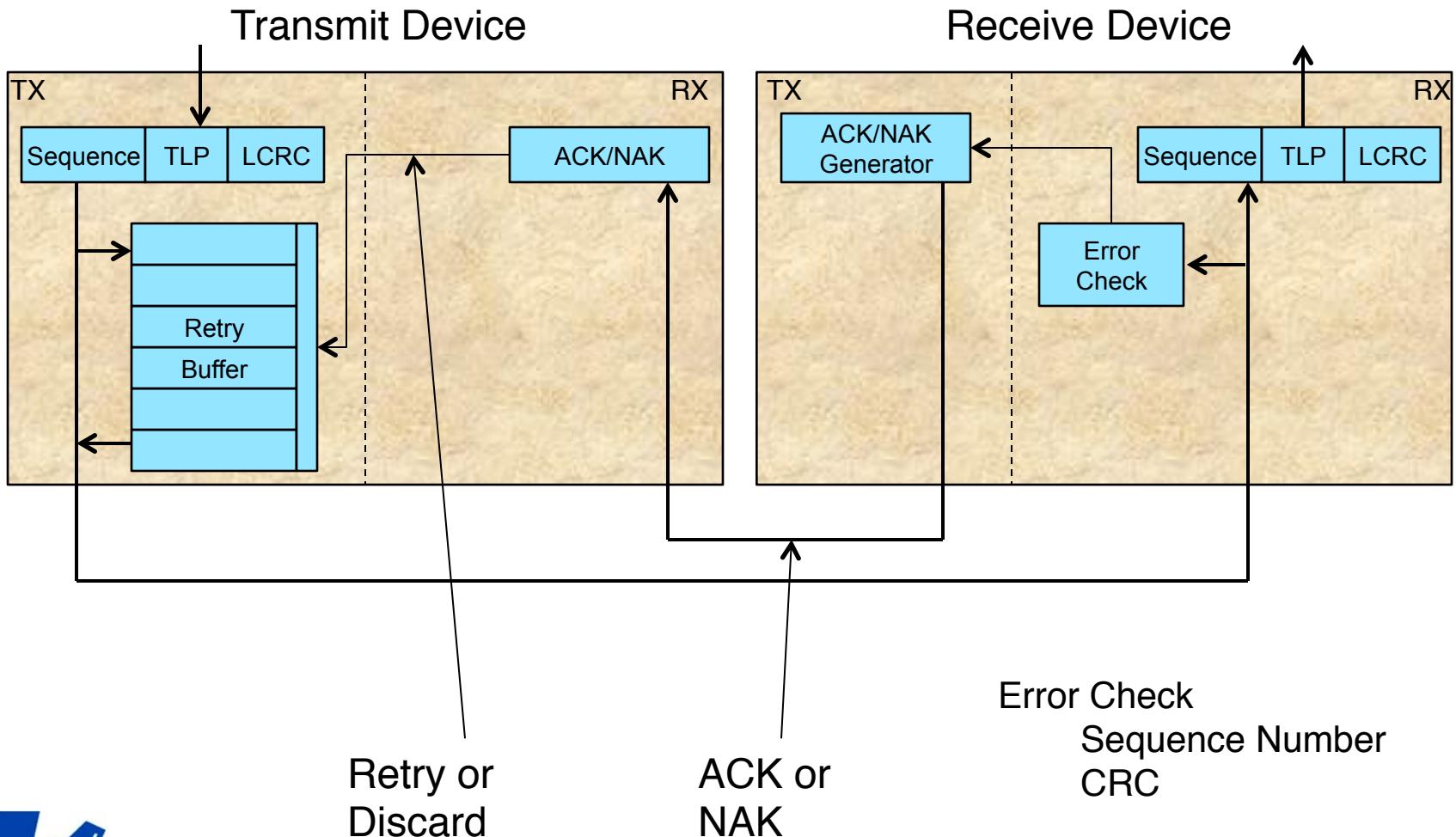
Retry Buffer

Lost ACK, Lost NAK

ACK/NAK Timeout

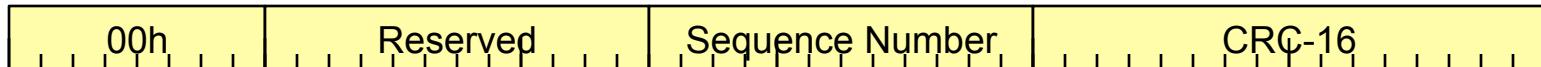


Overview of ACK/NAK Operation

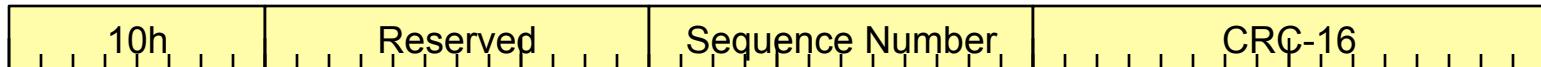


ACK and NAK DLLP Format

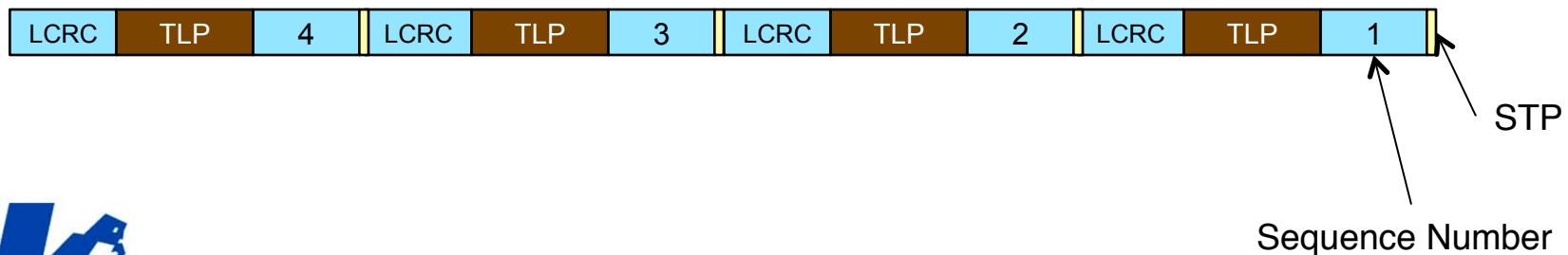
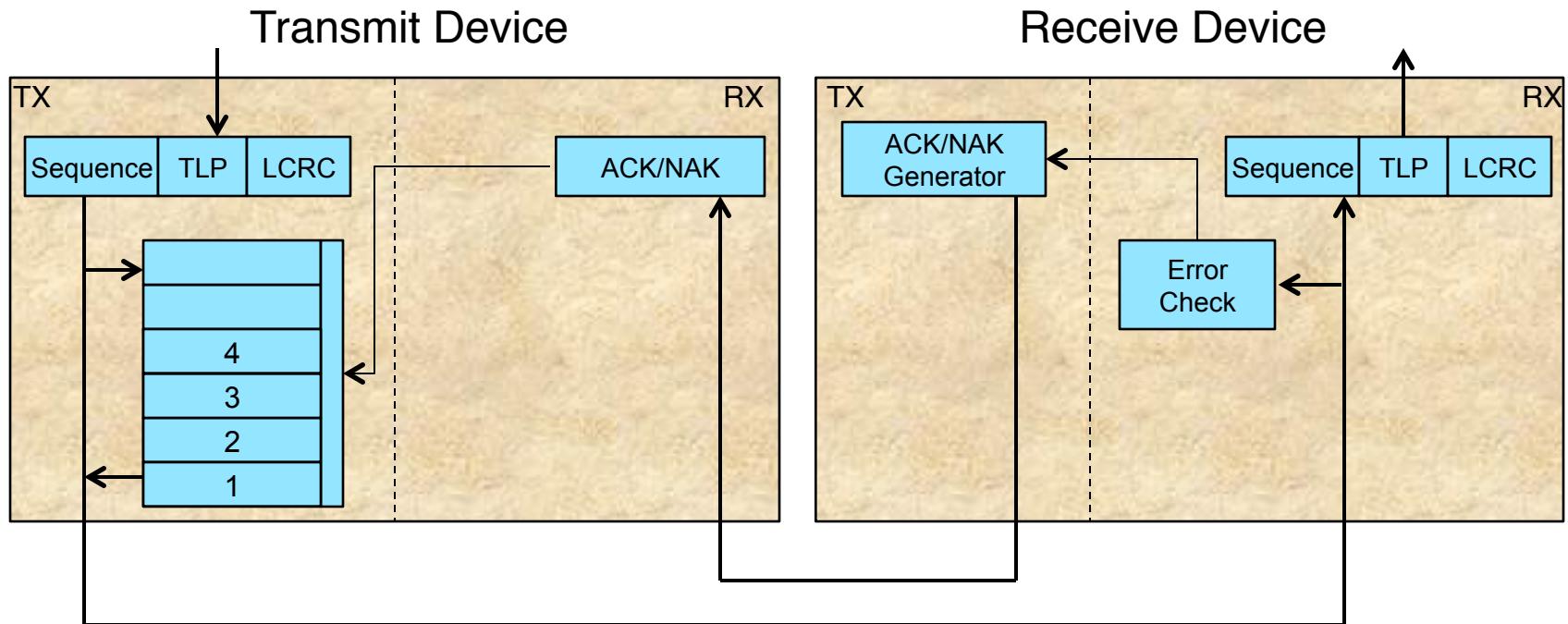
ACK Format



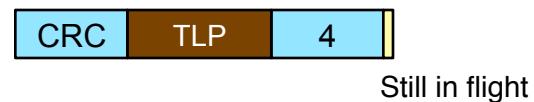
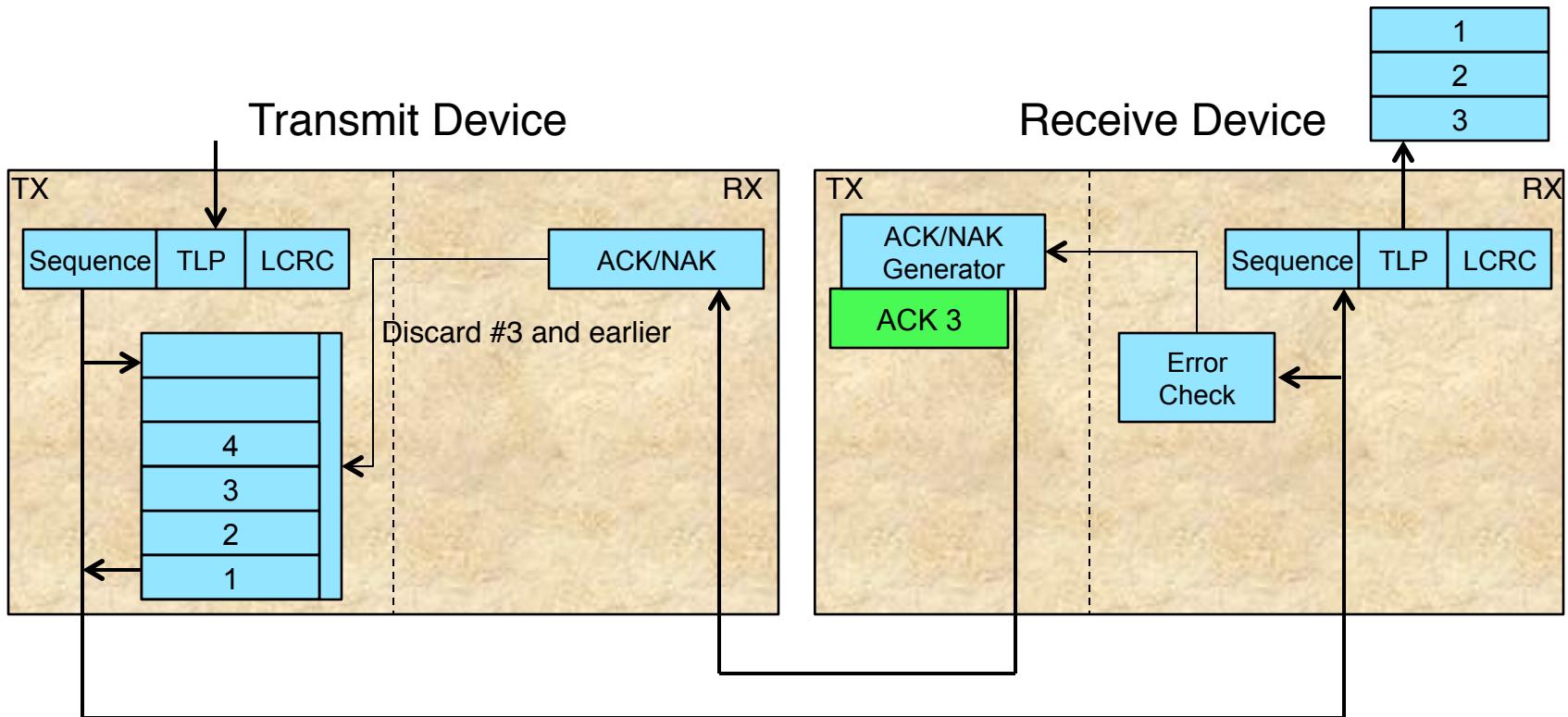
NAK Format



Transmit Packets Through Number 4

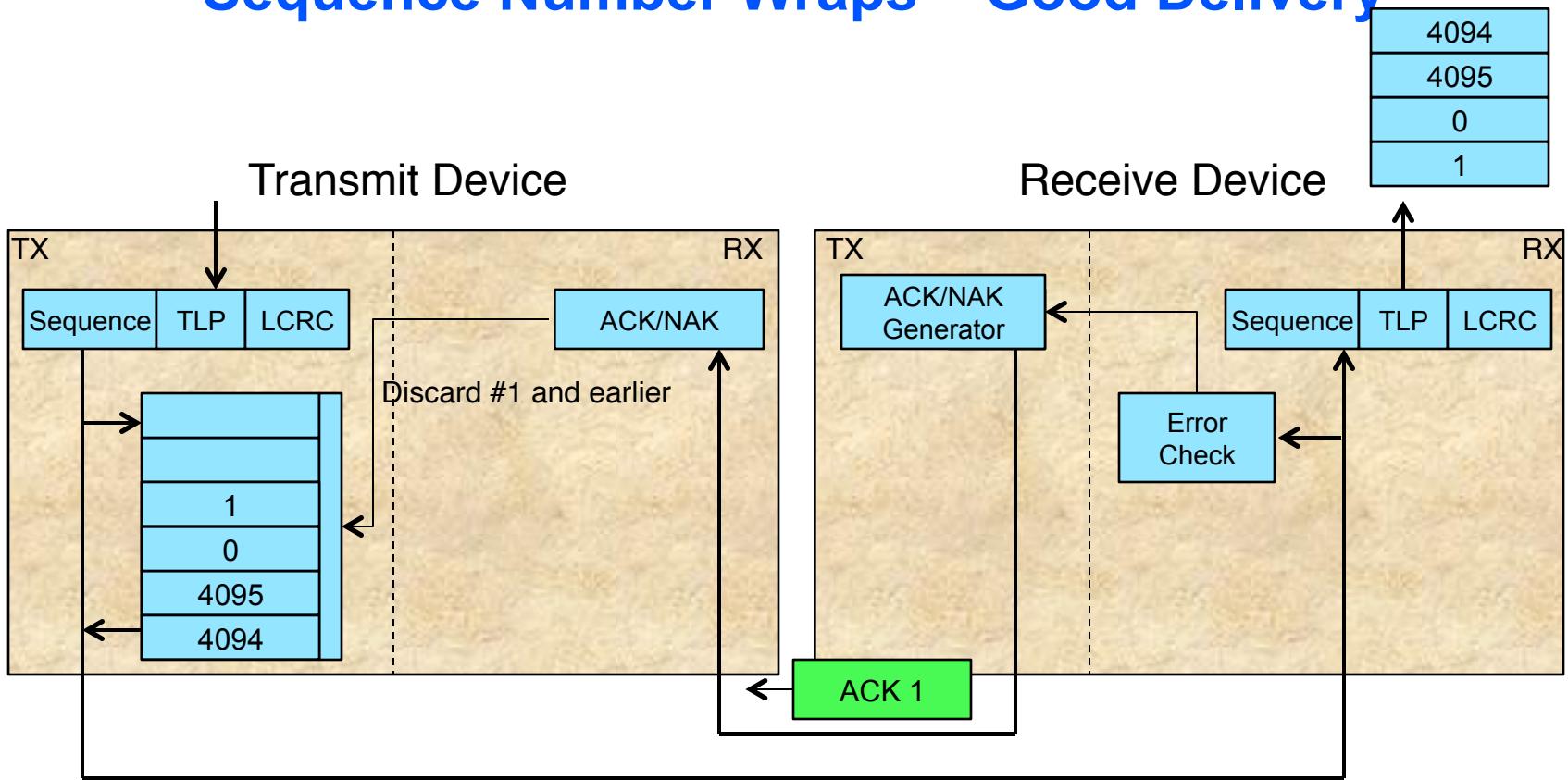


Acknowledge Through Number 3



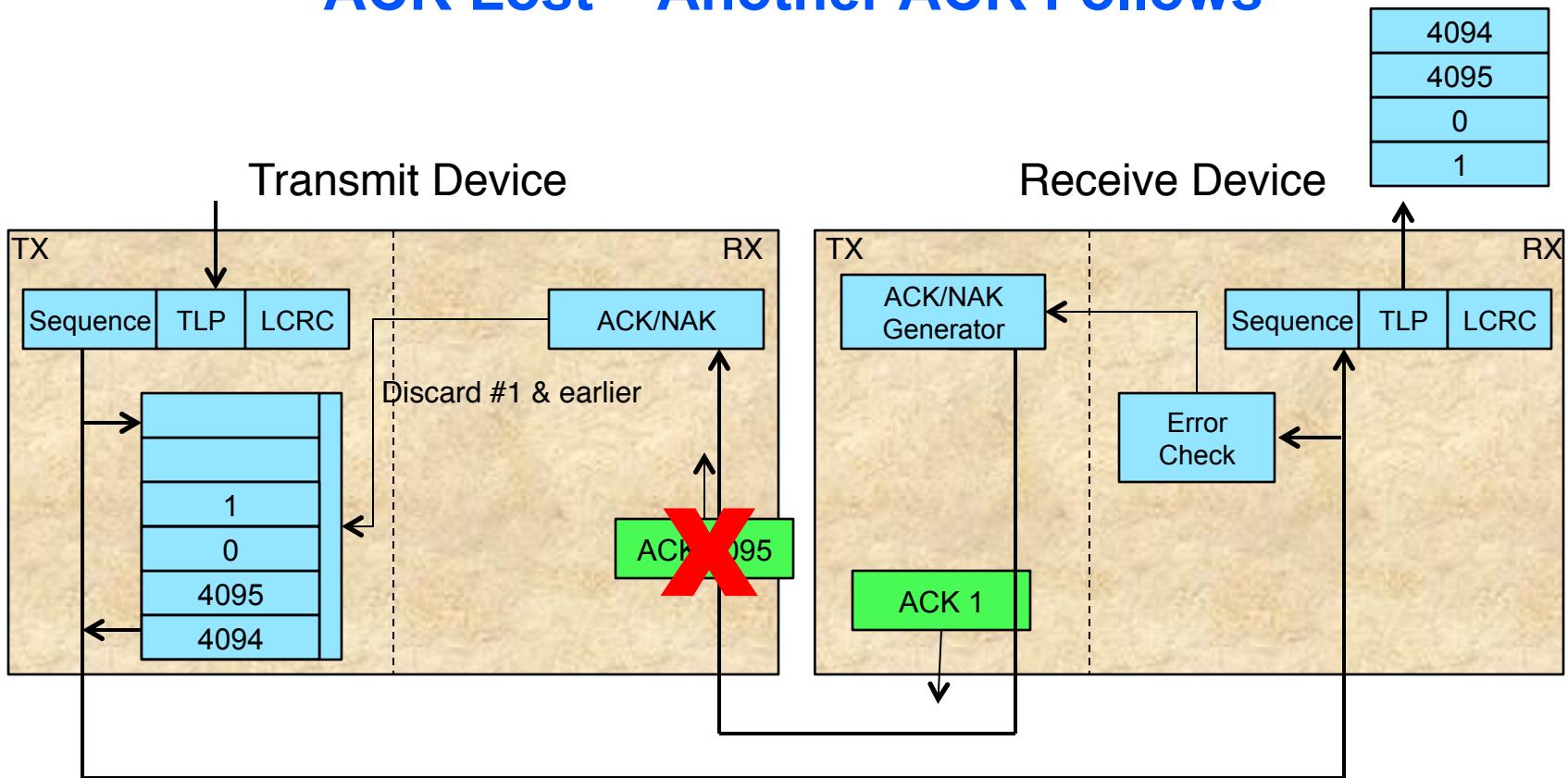
Packets must be delivered in order
ACK shows Sequence Number of last good packet

Sequence Number Wraps – Good Delivery

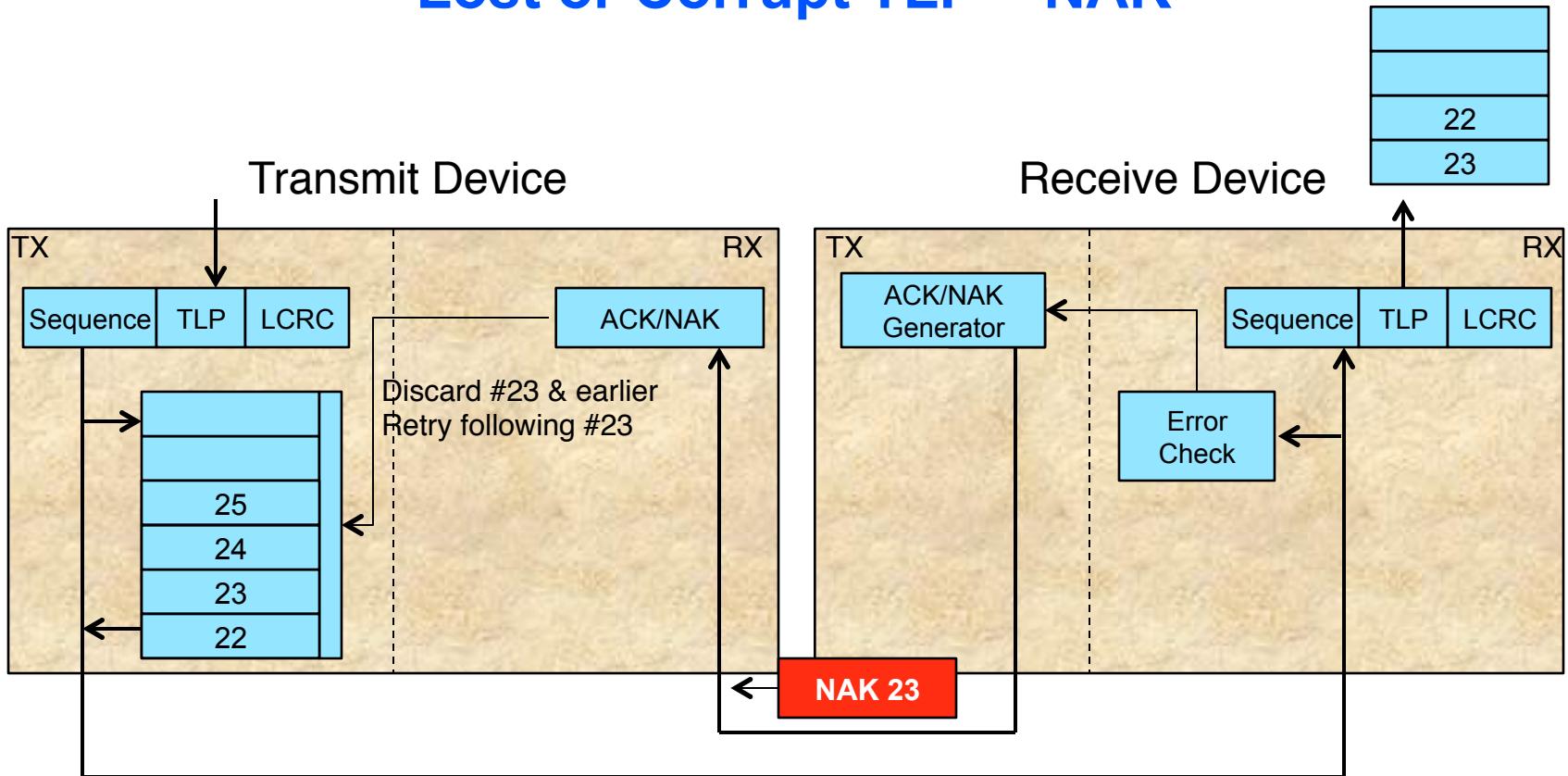


Sequence number uses a 12-bit counter
Set to 000h in DL_Inactive state
Wraps to 000h on overflow

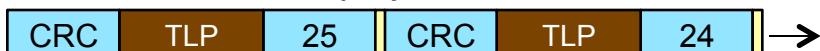
ACK Lost – Another ACK Follows



Lost or Corrupt TLP – NAK



Replay TLPs

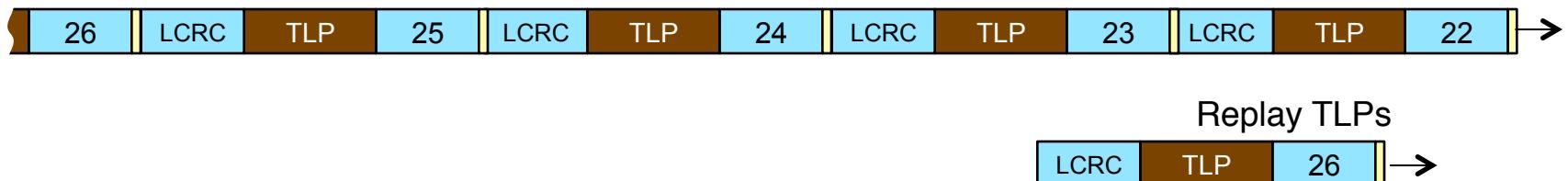
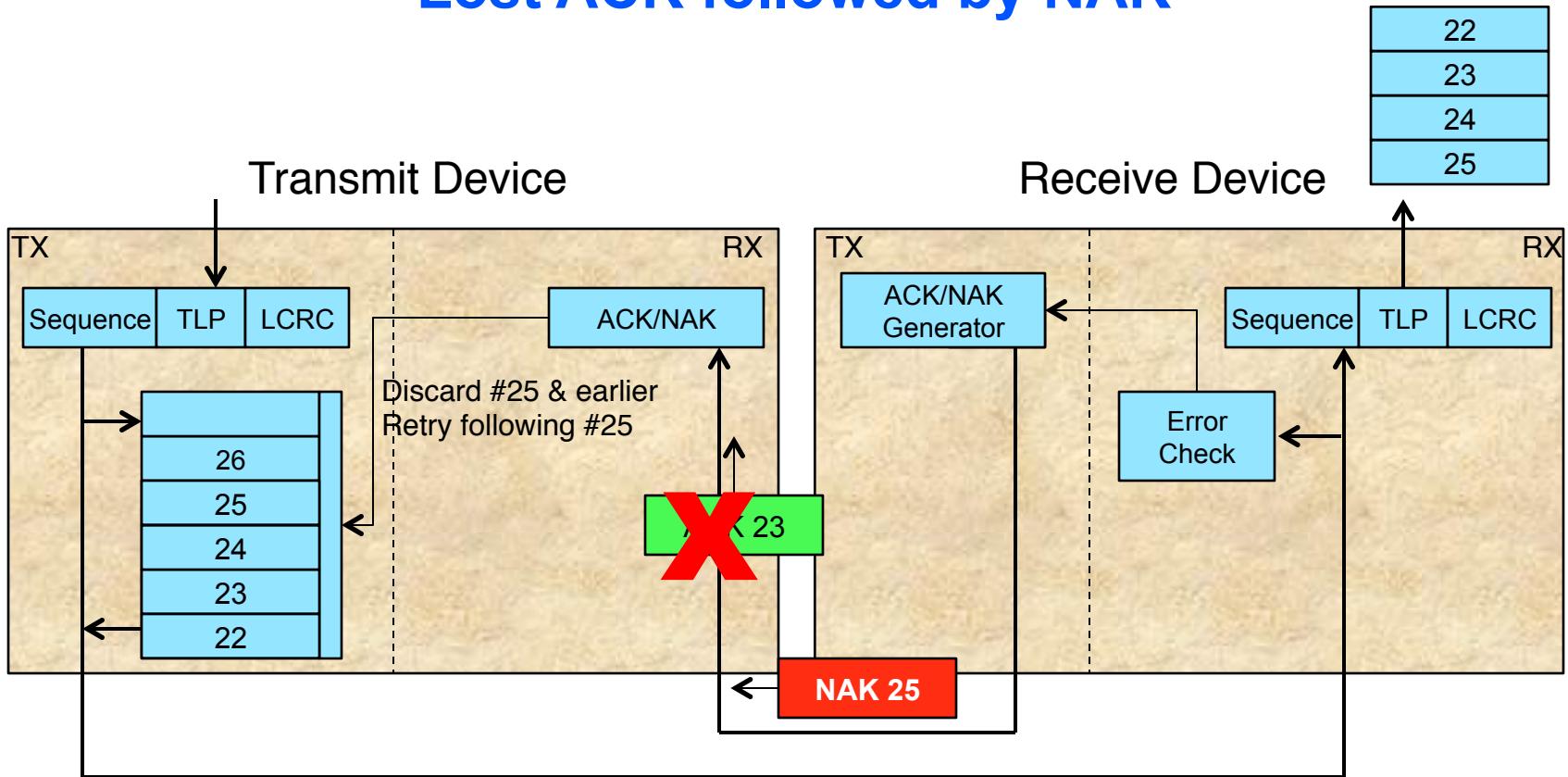


NAK signals the last good packet

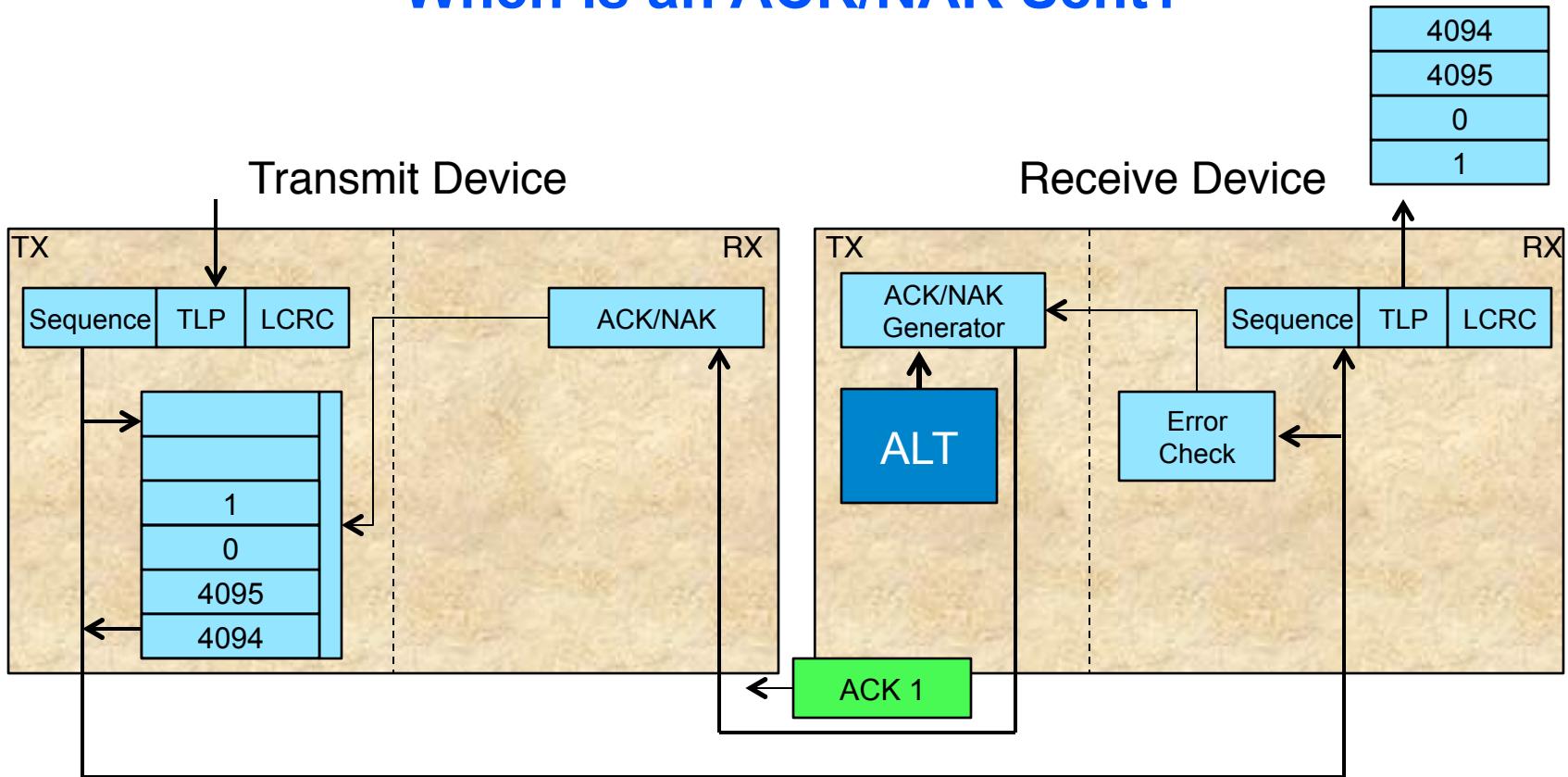
Following NAK, something failed

Only 1 NAK will be sent until retry is attempted

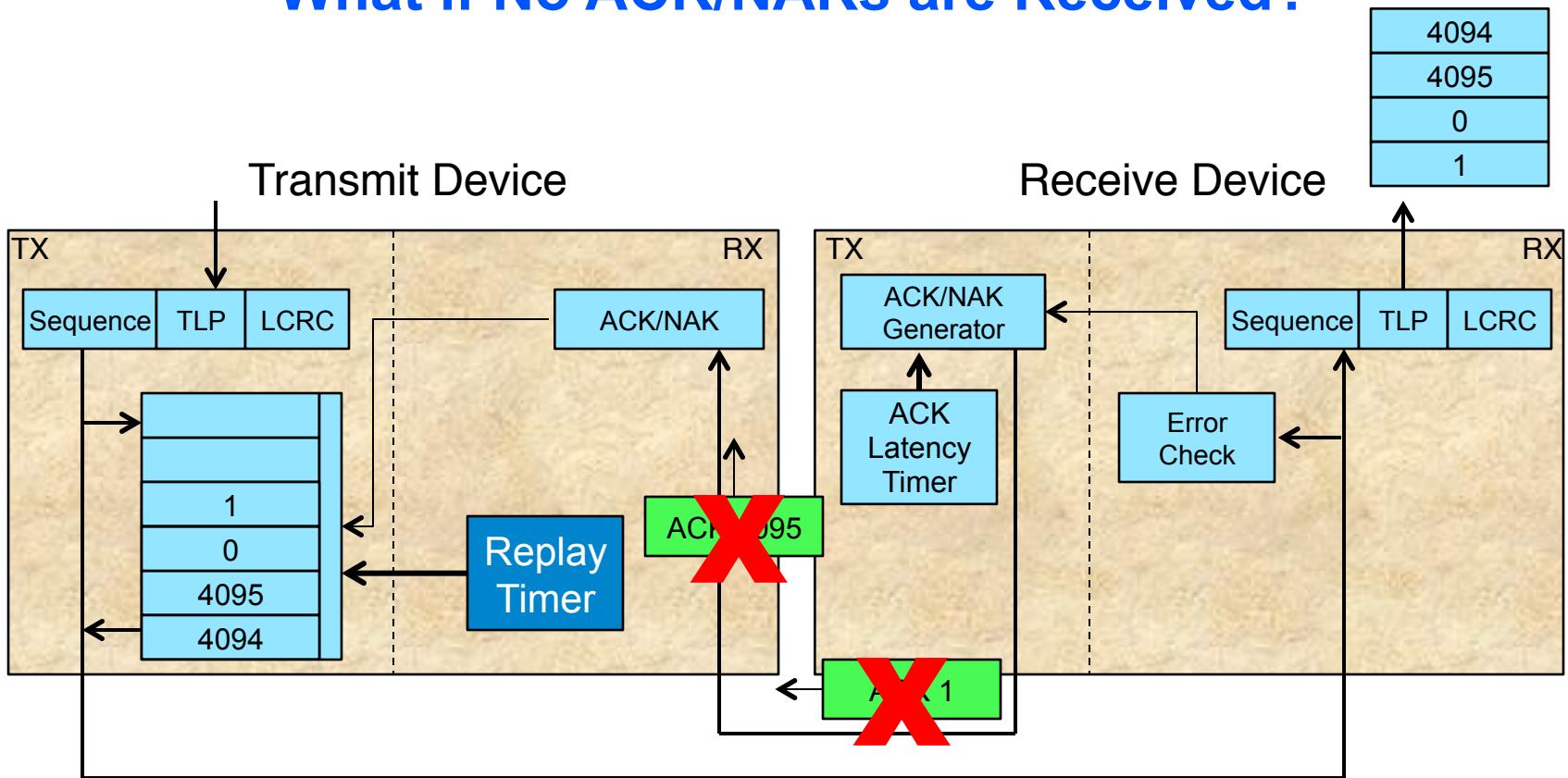
Lost ACK followed by NAK



When is an ACK/NAK Sent?



What if No ACK/NAKs are Received?



Receiver deletes duplicates

Specification defines duplicate as:
 $(\text{Next_Rcv_Seq} - \text{TLP Seq No.}) \bmod 4096 \leq 2048$

Replay Timer Rules

Purpose

Recover from lost ACKs or NAK

Reset timer means set it back to beginning

Method

Start timer on last symbol of TLP sent

Restart timer on each ACK DLLP received

if there are unacknowledged TLPs outstanding

On NAK, reset and hold timer

Reset timer if there are no unacknowledged TLPs outstanding

Replay complete buffer when timer expires

Restart timer means set it back to beginning and start it.

Timing

Suggest duration of timer in Specification

Based on Max Payload Size and Link width

$$\left[\frac{(\text{Max Payload Size} + \text{TLP Overhead}) * \text{ACK Factor}}{\text{Link Width}} + \text{Internal Delay} \right] * 3 + \text{L0s exit latency}$$

Replay Timer Rules

If Replay number rolls over from 11b to 00b:

- Transmitter signals the Physical Layer to retrain
- Data Link Layer state is not reset,
- Contents of the Retry Buffer remain

File	Setup	Record	Generate	Report	Search	View	Tools	Window	Help
Packet 0	R→ 2.5 x16	TLP 2829	Cpl	CplID 010:01010	Length 16	RequesterID 001:00:0	Tag 1	CompleterID 000:04:0	Status SC
									BCM 0 Byte Cnt 64 Lwr Addr 0x40
									Data 16 dwords LCRC 0x509C8967 Time Delta 164.000 ns
Packet 1	R← 2.5 x16	DLLP	ACK	AckNak_Seq_Num 2829	CRC 16 0x8AB7	Idle 640.000 ns	Time Stamp 0000 . 014 396 696 s		
Packet 2	R← 2.5 x16	TLP 2454	Mem	MRd(32) 000:00000	TC 1 TH 0 TD 0 EP 001 Attributes AT 00 Length 16	RequesterID 001:00:0	Tag 0	Address 0DC5F780	1st BE 1111 Last BE 1111 LCRC 0xFB62783B Idle 0.000 ns
Packet 3	R← 2.5 x16	TLP 2455	Mem	MRd(32) 000:00000	Length 16	RequesterID 001:00:0	Tag 1	Address 0DC5F7C0	1st BE 1111 Last BE 1111 LCRC 0x4B296E5B Time Delta 132.000 ns Time Stamp 0000 . 014 397 344 s
Packet 4	R→ 2.5 x16	DLLP	Update EC-NP	VC ID 0 HdrFC 37 DataFC 1	CRC 16 0x6E26	Idle 8.000 ns	Time Stamp 0000 . 014 397 476 s		
Packet 5	R← 2.5 x16	DLLP	ACK	AckNak_Seq_Num 2455	CRC 16 0xB8C8	Idle 116.000 ns	Time Stamp 0000 . 014 397 488 s		
Packet 6	R→ 2.5 x16	TLP 2830	Cpl	CplID 010:01010	Length 16	RequesterID 001:00:0	Tag 0	CompleterID 000:04:0	Status SC
									BCM 0 Byte Cnt 64 Lwr Addr 0x00
									Data 16 dwords LCRC 0xCEB9096A Idle 0.000 ns
Packet 7	R→ 2.5 x16	TLP 2831	Cpl	CplID 010:01010	Length 16	RequesterID 001:00:0	Tag 1	CompleterID 000:04:0	Status SC
									BCM 0 Byte Cnt 64 Lwr Addr 0x40
									Data 16 dwords LCRC 0x5BE9940D Idle 72.000 ns
Packet 8	R→ 2.5 x16	SKIP	COM	SKIP Symbols K28.5 K28.6 K28.7 K28.8	Time Delta 72.000 ns	Time Stamp 0000 . 014 397 728 s			
Packet 9	R← 2.5 x16	DLLP	ACK	AckNak_Seq_Num 2831	CRC 16 0xC880	Idle 116.000 ns	Time Stamp 0000 . 014 397 800 s		
Packet 10	R← 2.5 x16	TLP 2456	Mem	MRd(32) 000:00000	Length 16	RequesterID 001:00:0	Tag 0	Address 0DC5F800	1st BE 1111 Last BE 1111 LCRC 0xA4F1BEAB Idle 0.000 ns Time Stamp 0000 . 014 397 920 s

Ready

Search: Fwd



PCI Express Transaction Details																
Packet	R→	2.5	TLP	Mem	MWr(64)	TC	TH	TD	EP	Attributes	AT	Length	RequesterID	Tag	Address	
0	R→	2.5	TLP	Mem	MWr(64)	011:00000	0	0	1	0	000	00	1023	000:01:2	3	781F33AB:12340000
	1st BE	Last BE	Data		ECRC	LCRC	Idle	Time Stamp								
	1111	1000	1023 dwords		0x0D632C96	0x36214D17	0.000 ns	0000 . 000 000 000 s								
Packet	R→	2.5	TLP	Mem	MRd(32)	Length	RequesterID	Tag			Address	1st BE	Last BE	ECRC		
1	R→	2.5	TLP	Mem	MRd(32)	Length	RequesterID	Tag			Address	1st BE	Last BE	ECRC		
	1111	1111	1111	1111	000:00000	1023	000:01:2	4			00010000	1111	1111	0xA83F0CE		
	LCRC	Time Delta	Time Stamp													
	0xA3AD0991	96.000 ns	0000 . 000 016 480 s													
Packet	R←	2.5	DLL	NAK	AckNak_Seq_Num	CRC 16	Time Delta	Time Stamp								
2	R←	2.5	DLL	NAK	AckNak_Seq_Num	CRC 16	Time Delta	Time Stamp								
	x1	x1			1	0xF91E	32.000 ns	0000 . 000 016 576 s								
Packet	R→	2.5	TLP	Mem	MRd(32)	Length	RequesterID	Tag			Address	1st BE	Last BE	ECRC		
3	R→	2.5	TLP	Mem	MRd(32)	Length	RequesterID	Tag			Address	1st BE	Last BE	ECRC		
	x1	x1	2	Mem	000:00000	1023	000:01:2	4			00010000	1111	1111	0xA83F0CE		
	LCRC	Time Delta	Time Stamp													
	0xA3AD0991	96.000 ns	0000 . 000 016 608 s													
Packet	R←	2.5	DLLP	ACK	AckNak_Seq_Num	CRC 16	Idle	Time Stamp								
4	R←	2.5	DLLP	ACK	AckNak_Seq_Num	CRC 16	Idle	Time Stamp								
	x1	x1			2	0xF155	0.000 ns	0000 . 000 016 704 s								
Packet	R←	2.5	TLP	Cpl	CplID	Length	RequesterID	Tag			CompleterID	Status	BCM	Byte Cnt	Lwr Addr	
5	R←	2.5	TLP	Cpl	CplID	Length	RequesterID	Tag			CompleterID	Status	BCM	Byte Cnt	Lwr Addr	
	x1	x1	100	Cpl	010:01010	1012	000:01:2	4			001:01:0	SC	0	4092	0x00	
	Data	ECRC	LCRC													
	1012 dwords	0x9EB43326	0x68BE71A6													
	0.000 ns	0000 . 000 016 736 s														
Packet	R←	2.5	TLP	Cpl	CplID	Length	RequesterID	Tag			CompleterID	Status	BCM	Byte Cnt	Lwr Addr	
6	R←	2.5	TLP	Cpl	CplID	Length	RequesterID	Tag			CompleterID	Status	BCM	Byte Cnt	Lwr Addr	
	x1	x1	101	Cpl	010:01010	11	000:01:2	4			001:01:0	SC	0	44	0x00	
	Data	ECRC	LCRC													
	11 dwords	0xE7F30A0	0x5B06C2B5													
	272.000 ns	0000 . 000 033 024 s														
Packet	R→	2.5	DLLP	ACK	AckNak_Seq_Num	CRC 16	Idle	Time Stamp								
7	R→	2.5	DLLP	ACK	AckNak_Seq_Num	CRC 16	Idle	Time Stamp								
	x1	x1			101	0x904B	0.000 ns	0000 . 000 033 296 s								

Switch Buffering

Description: Switch Buffering

Switches may be:

Store and Forward

Receive entire packet and verify before passing it to out port

Cut-through

Pass the packet on before the entire packet is received

If packet is determined to be bad

invert CRC

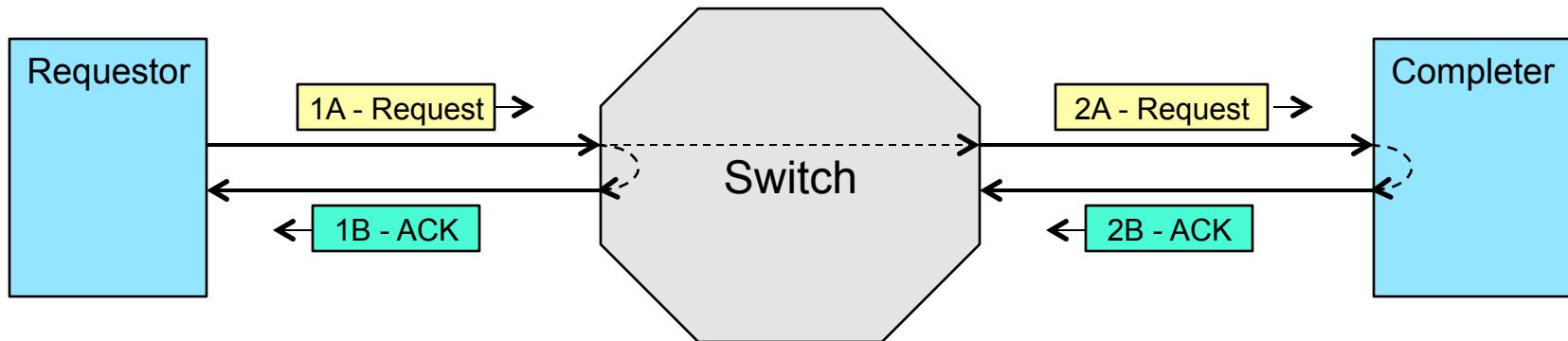
add END BAD framing

recipient will ignore

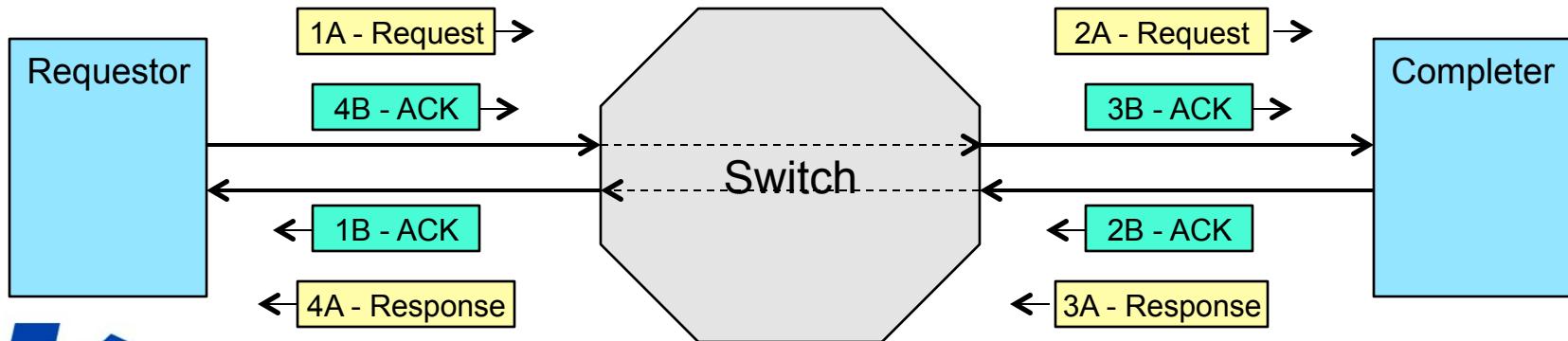
Operation is shown later

Switch – Store and Forward

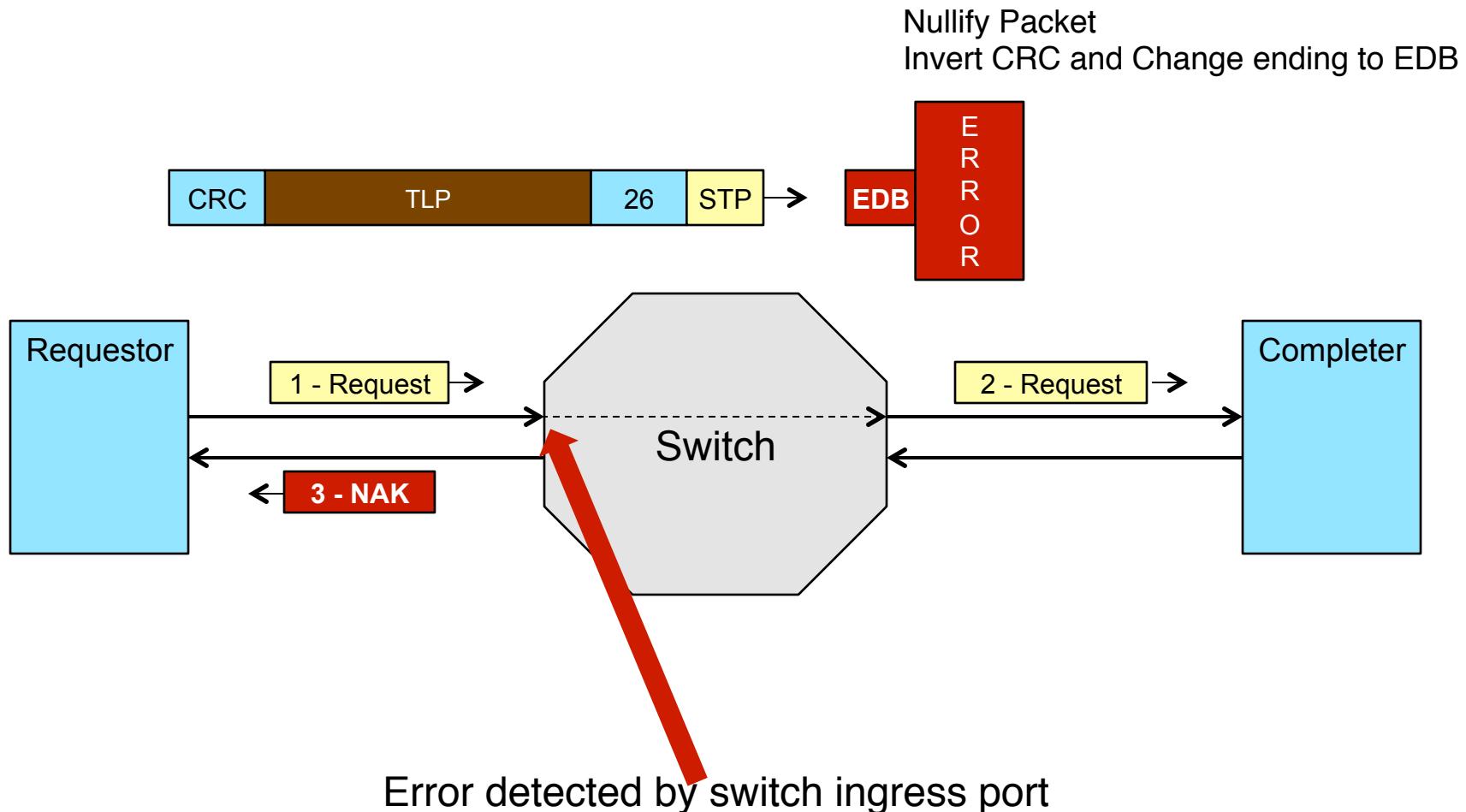
Posted Transaction



Non-Posted Transaction



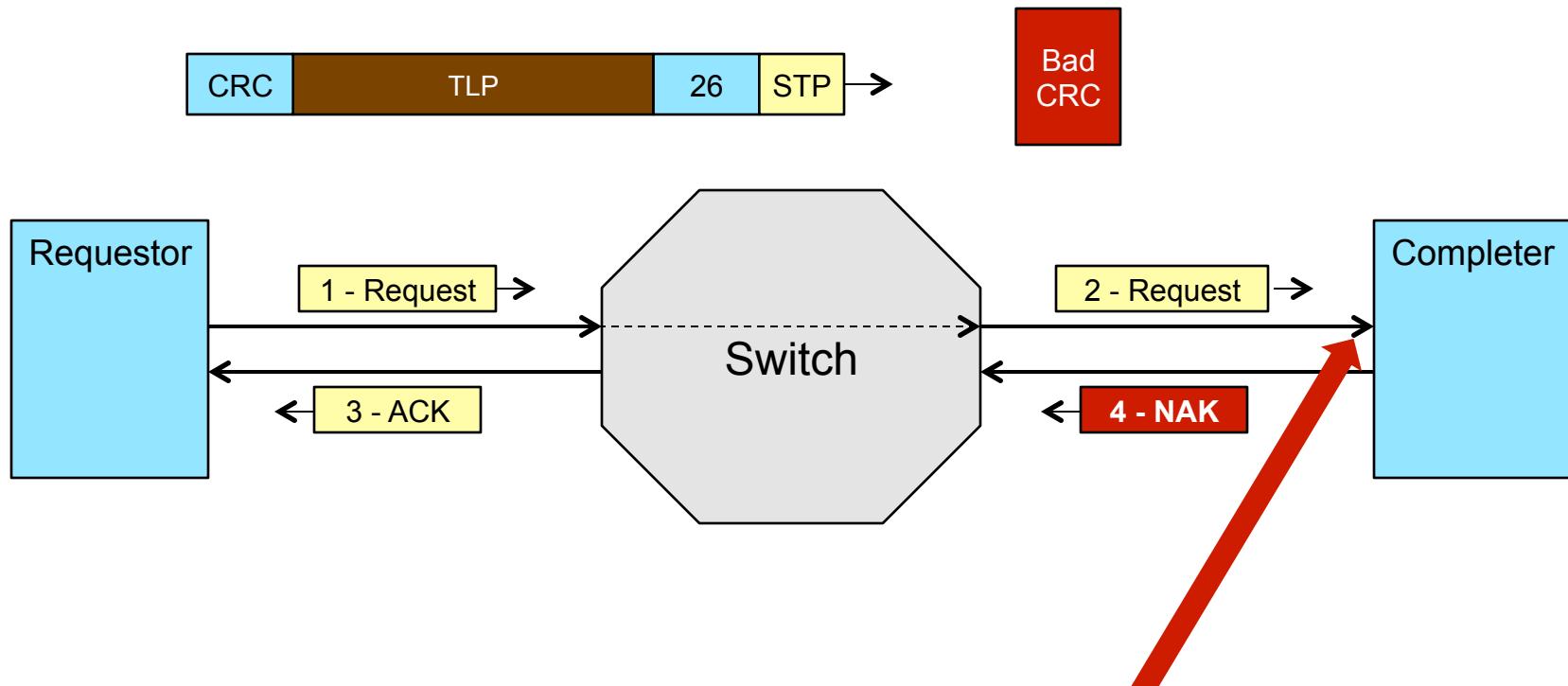
Switch – Cut-Through



Requestor gets NAK and replays buffer
Completer got nullified packet and ignored it
so is ready to receive packet on next try.

Switch – Cut-Through

Nullify Packet
Invert CRC and Change ending to EDB



Error detected after switch
Switch stored good packet
Switch to Completer link replays the retry buffer

Check for Understanding

1. What does the sequence number in a NAK indicate?
2. When is an ACK sent?
3. When is a NAK sent?
4. How many times is a packet retried?

Covered in this Section

PCI/PCIe Concepts

Topology Discovery and Enumeration

PCI Transactions

PCIe Link Layer

PCIe Physical Layer

Flow Control

ACK/NAK protocol

Notes



Section 2

PCIe Overview



Covered in this Section

PCI/PCIe Concepts

Topology Discovery and Enumeration

PCI Transactions



PCI vs. SCSI

SCSI

- Initiator transfers commands
- Target transfers data and status

PCI

- Initiator stores commands
- Target transfers commands, data, and status



PCI Versions

PCI

Parallel bus



PCIe

Bit serial per lane
Point to point with switches
Byte parallel
Up to 32 lanes per link
Dual simplex
Differential signaling



PCI-X

PCI - eXtended

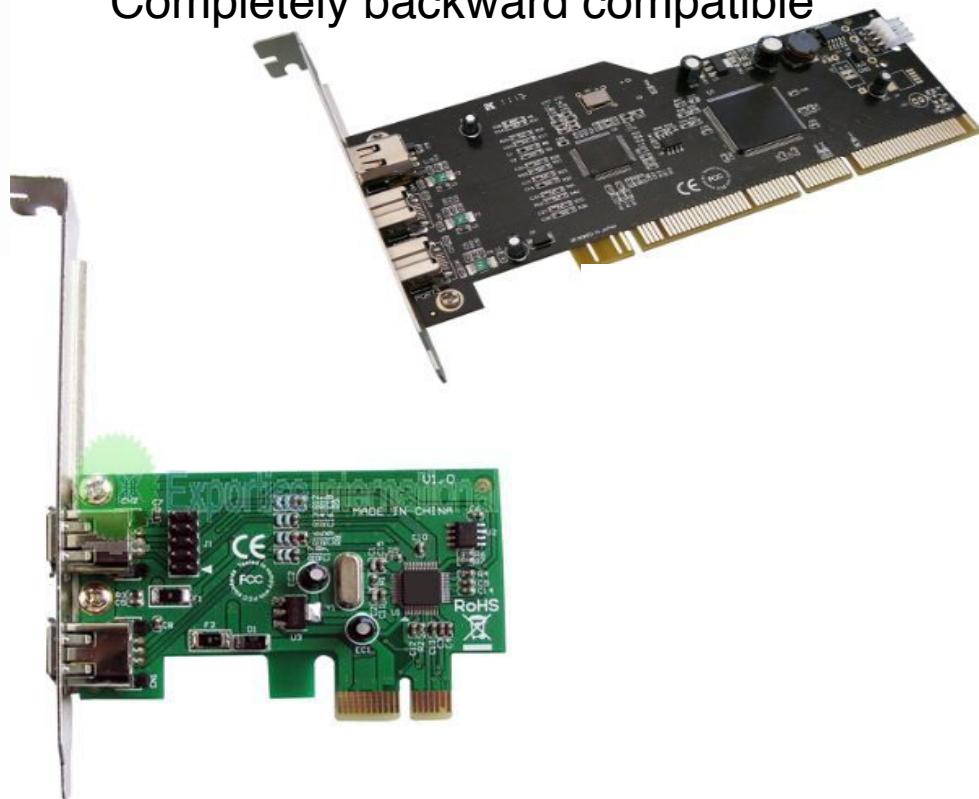
Parallel bus

Primarily for high end workstations or servers

Up to 533 * 64bit transfers

ECC on header and data

Completely backward compatible



PCIe Speeds

Aggregate GB per second

GT/s	Link Width							
		X1	X2	X4	X8	X12	X16	X32
Gen 1	2.5	0.5	1	2	4	6	8	16
Gen 2	5.0	1	2	4	8	12	16	32
Gen 3	8.0	2	4	8	16	24	32	64
Gen 4	16.0	4	8	16	32	48	64	128

Formula:

$$\frac{\text{GT/s} * \text{width} * 2 \text{ directions}}{\text{bits per byte/character}} = \text{GB per second}$$

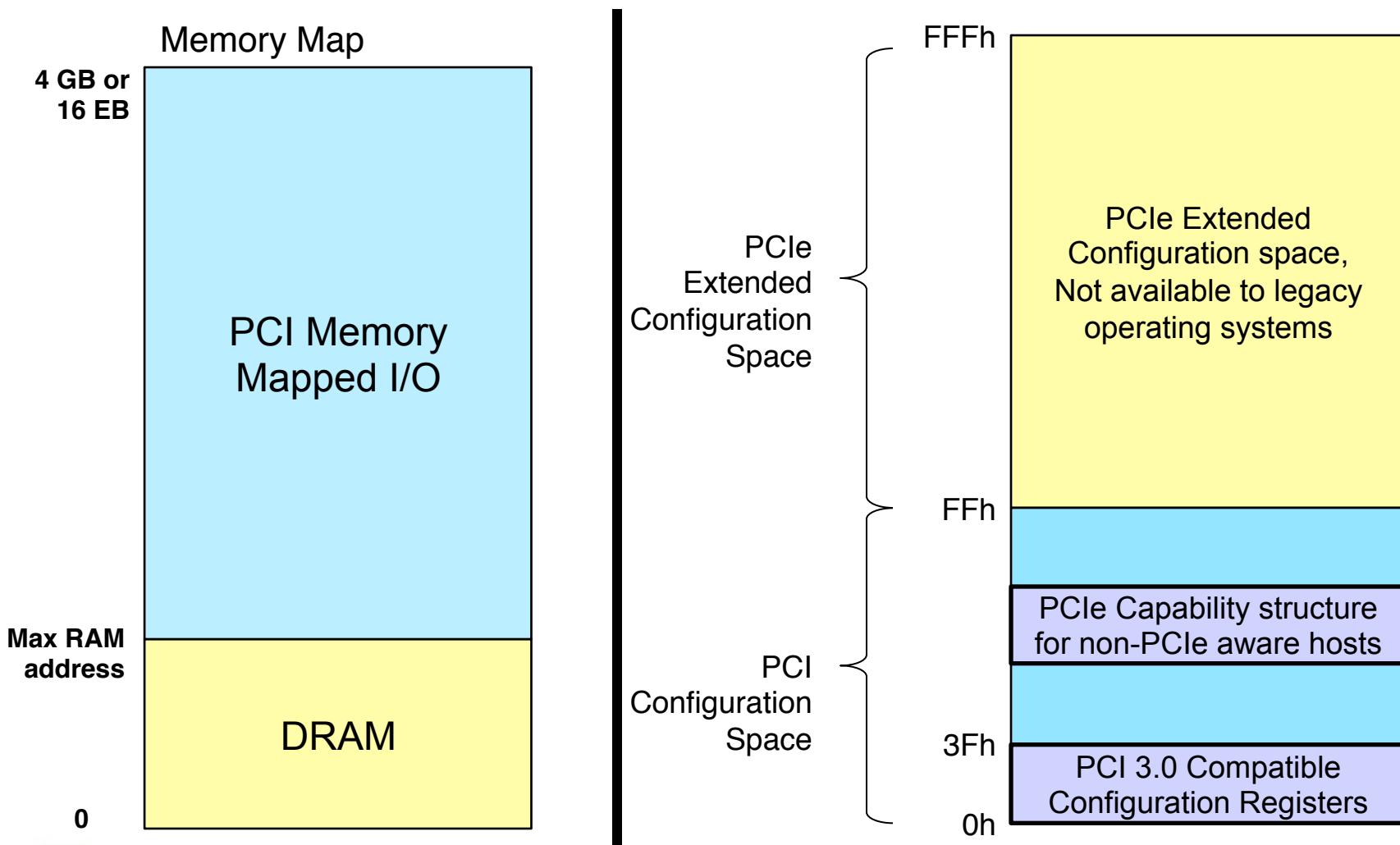
This chart includes the 8b/10b encoding of Gen 1 and Gen 2

It does not take into account the 128b/130b encoding of Gen 3 & 4 so their speeds are 1.5625% overstated.

This chart shows signaling speed, not overhead

PCIe Configuration

Memory Space and Configuration Mapping



Caution: PCIe Specification shows lowest address at bottom of picture

PCI Configuration Header for NVMe

Identification

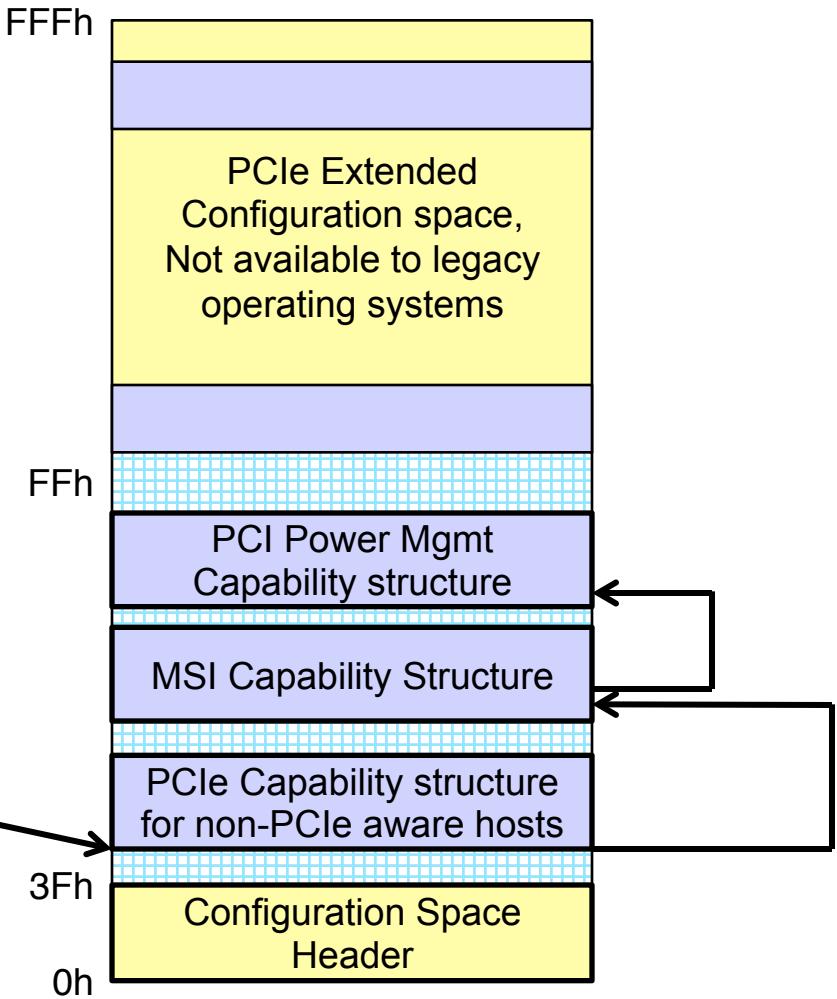
Type 0 Configuration Space Header (Endpoint)	Byte
Device ID	0
Status	4
Class Code	8
BIST	C
Header Type = 0h	
Master Latency Timer	
Cache Line Size	
BAR0 – MLBAR – NVMe Registers	10
BAR1 – MUBAR – NVMe Registers	14
BAR2 – I/O based accesses, if supported	18
BAR3 - Reserved	1C
BAR4 – Vendor Specific	20
BAR5 – Vendor Specific	24
Cardbus CIS Pointer	28
Subsystem ID	2C
Subsystem Vendor ID	
Expansion ROM Base Address	30
Reserved	
Capabilities Pointer	
Reserved	38
Max Latency = 00h	3C
Min Grant = 00h	
Interrupt Pin	
Interrupt Line	



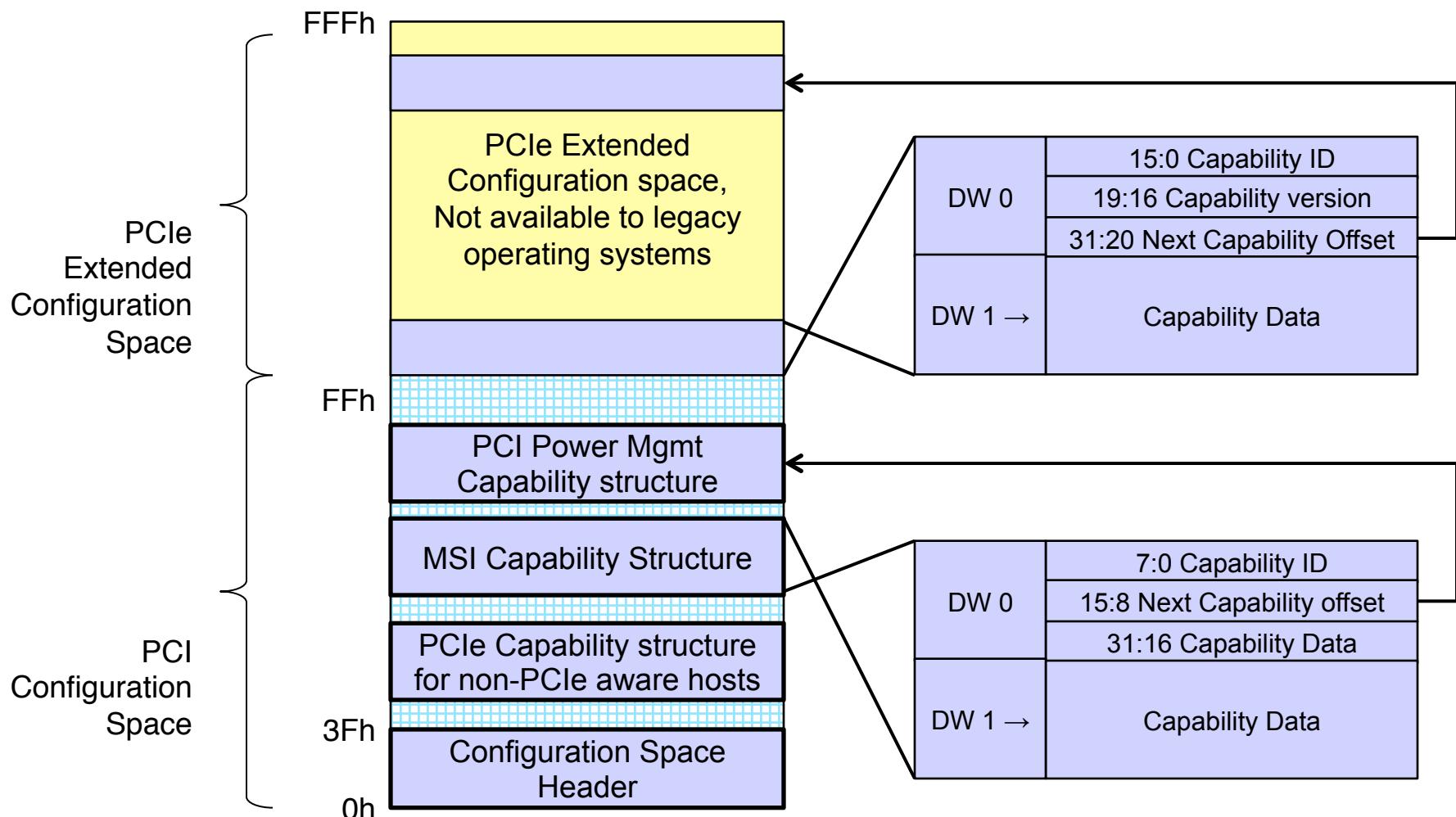
Caution: PCIe Specification shows lowest address at top of picture

Capability Link List

Device ID	Vendor ID		
Status	Command		
Class Code		Revision ID	
BIST	Header Type = 0h	Master Latency Timer	Cache Line Size
BAR0			
BAR1			
BAR2			
BAR3			
BAR4			
BAR5			
Cardbus CIS Pointer			
Subsystem ID	Subsystem Vendor ID		
Expansion ROM Base Address			
Reserved		Capabilities Pointer	
Reserved			
Max Latency = 00h	Min Grant = 00h	Interrupt Pin	Interrupt Line



PCIe Configuration Space



Caution: PCIe Specification shows lowest address at bottom of picture on left and top of picture on right

Some PCI Capability Registers

Capability ID	PCIe Name	NVMe Name	Name
01h	Power Mgmt Capability	PMCAP Section 2.2	PCI Power Management Capability
05h	MSI Capability LB3.0 Section 6.8.1	MSICAP Section 2.3	MSI Capability
11h	MSI-X Capability LB 3.0 Section 6.8.2	MSIXCAP Section 2.4	MSI-X Capability
10h	PCI Express PCI Express Base 11 Section 7.89	PXCAP Section 2.5	PCI Express Capability

PCIe Capability Structure

	31	23	15	7	0			
					00h			
Device	PCIe Capabilities Register		Next Cap pointer	PCIe Cap ID = 10	04h			
	Device Capabilities							
Link	Device Status		Device Control					
	Link Capabilities							
Slot	Link Status		Link Control					
	Slot Capabilities							
Root	Slot Status		Slot Control					
	Root Capabilities							
Device 2	Root Control							
	Root Status							
Link 2	Device Capabilities 2							
	Device Status 2		Device Control 2					
Slot 2	Link Capabilities 2							
	Link Status 2		Link Control 2					
	Slot Capabilities 2							
	Slot Status 2		Slot Control 2					



Caution: PCIe Specification shows lowest address at top of picture

PCIe Extended Configuration Capability – Part 1

Capability ID	Name
0001h	Advanced Error Reporting
0002h	Virtual Channel Capability w/o multi-function virtual channel
0003h	Device Serial Number Capability
0004h	Power Budgeting Capability
0005h	Root Complex Link Declaration Capability
0006h	Root Complex Internal Link Control
0007h	Root Complex Event Collector Capability
0008h	Multi-Function Virtual Channel Capability
0009h	Virtual Channel Capability w multi-function virtual channel
000Ah	RCRB (Root Complex Register Block) Header Capability
000Bh	Vendor Specific Capability
000Ch	Correlation Access Capability
000Dh	Access Control Services Extended Capability(ACS)
000Eh	ARI (Alternative Routing-ID Interpretation Capability
000Fh	Address Translation Services (ATS)
0010h	SR-IOV
0011h	MR-IOV
0012h	Multicast Capability
0013h	ATS Page Request Interface (PRI)
0015h	Resizable BAR (Base Address Register) Capability

PCIe Extended Configuration Capability – Part 2

Capability ID	Name
0016h	Dynamic Power Allocation Capability
0017h	TPH (TLP Processing Hints) Requester Capability
0018h	Latency Tolerance Reporting Capability
0019h	Secondary PCIe Extended Capability
001Bh	PASID
001Ch	Lightweight Notification
001Dh	Downstream Port Containment
001Eh	L1 PM Substates
001Fh	Precision Time Management
0020h	M-PCIe Extended Capability
0021h	Function Readiness Status
0022h	Readiness Time Reporting

Addressing

TLP Addressing Modes

Memory Address

Used with Memory and I/O Requests

ID

Used with Configuration Requests, ID Routed Messages and Completions
Call Bus-Device-Function in PCI and PCI-X

Implicit

Used with Message Requests only
Routing type implies destination

A Few Details on BDF

PCI and PCI-X devices are addressed as Bus – Device - Function

Bus addresses are assigned by host during initialization
8 bits – 256 possible buses

Device addresses are assigned by host during initialization
5 bits – legacy
Always 00h in PCIe
0 bits – with ARI

Function addresses are assigned by manufacturer
3 bits – legacy, up to 8 Functions per device
8 bits – with ARI, up to 256 Functions

ARI – Alternative Routing-ID Interpretation
Applicable to Requester IDs, Completer IDs and Routing IDs

ARI – Alternative Routing-ID Interpretation

4 DW Header with ID Routing – without ARI

FMT	Type			
Bus Number	Device ID.	Function No.		

4 DW Header with ID Routing – with ARI

FMT	Type			
Bus Number	Function Number			

Addressing Mode – Address

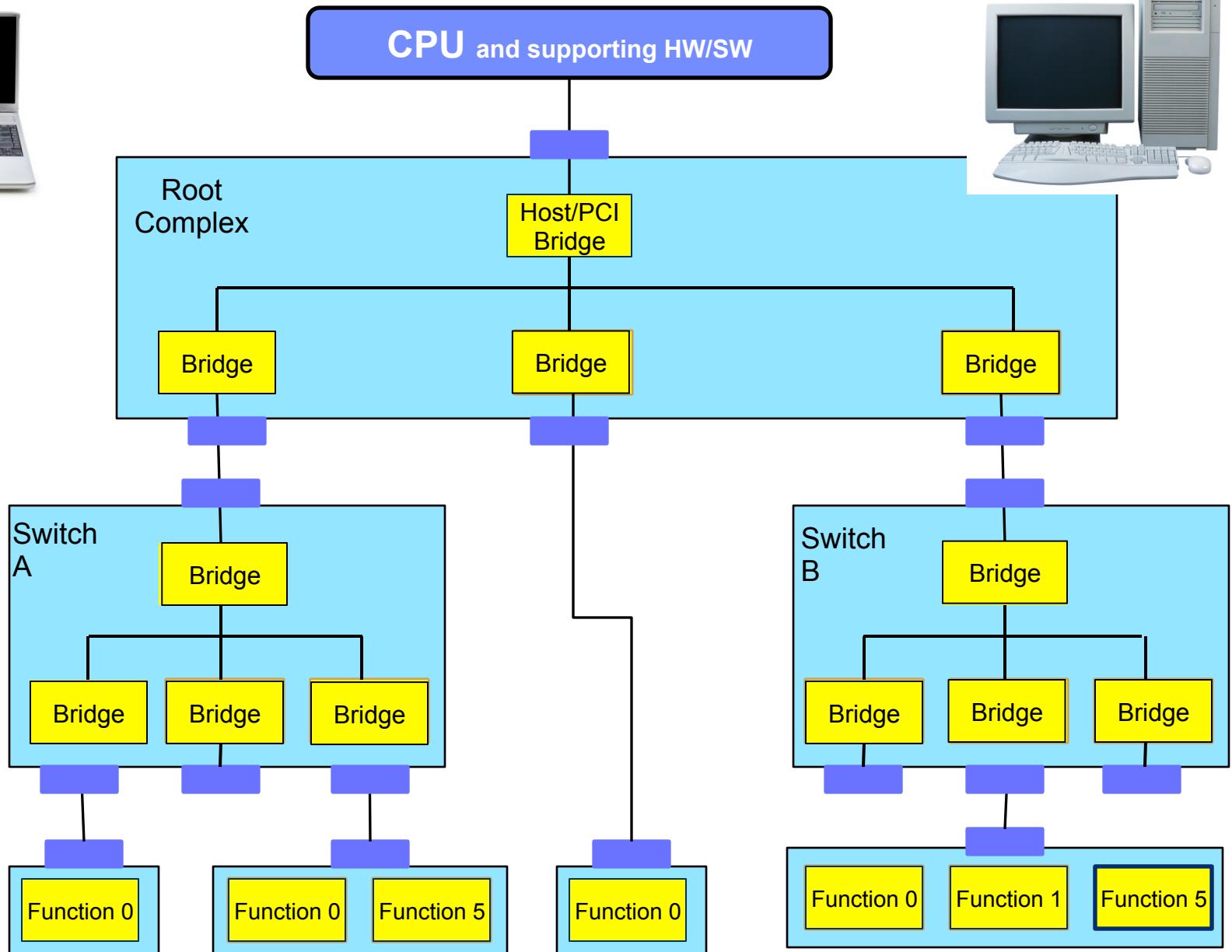
	Byte + 0		Byte + 1		Byte + 2		Byte + 3	
0	FMT 0_x_1	Type	R	TC	R	^A ^T ^T ^R	R	T H D P
1	Fields dependent on FMT and Type							
2	64 bit address							
3	32 bit address							

	Byte + 0		Byte + 1		Byte + 2		Byte + 3	
0	FMT 0_x_0	Type	R	TC	R	^A ^T ^T ^R	R	T H D P
1	Fields dependent on FMT and Type							
2	32 bit address							

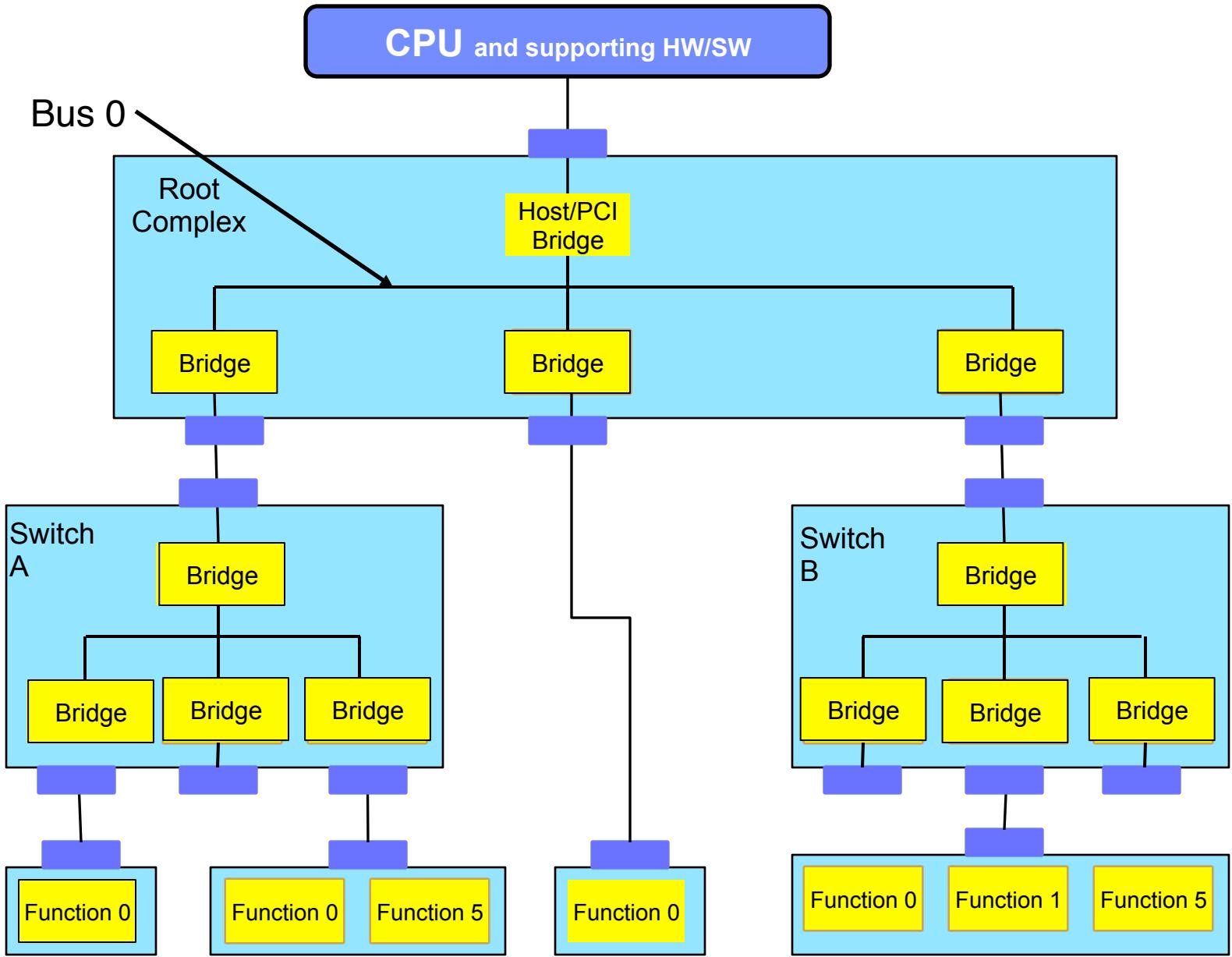
Topology Discovery And Enumeration



PCIe Example Topology



Topology after Power On/Reset



Discovery Process

Set Bus variables Primary ID = Secondary ID = Subordinate ID = 0

Next Device: Host does Configuration Read to Primary ID, Device 0, Function 0

If valid Vendor ID, read header type

If 01h (bridge) goto Bridge

If 00h (end device) goto Next Bus

If Multi-Function bit is on, check Functions 1-7 or ARI Capability

Bridge: set variables:

Primary ID to Primary ID

Increment Secondary ID and Subordinate ID

Set Secondary ID in this bridge

Set Subordinate ID in all bridges between this and Primary ID = 0

Increment Primary ID

Goto Next Device

Next Bus: Set Primary ID = Primary ID + 1

Goto Next Device

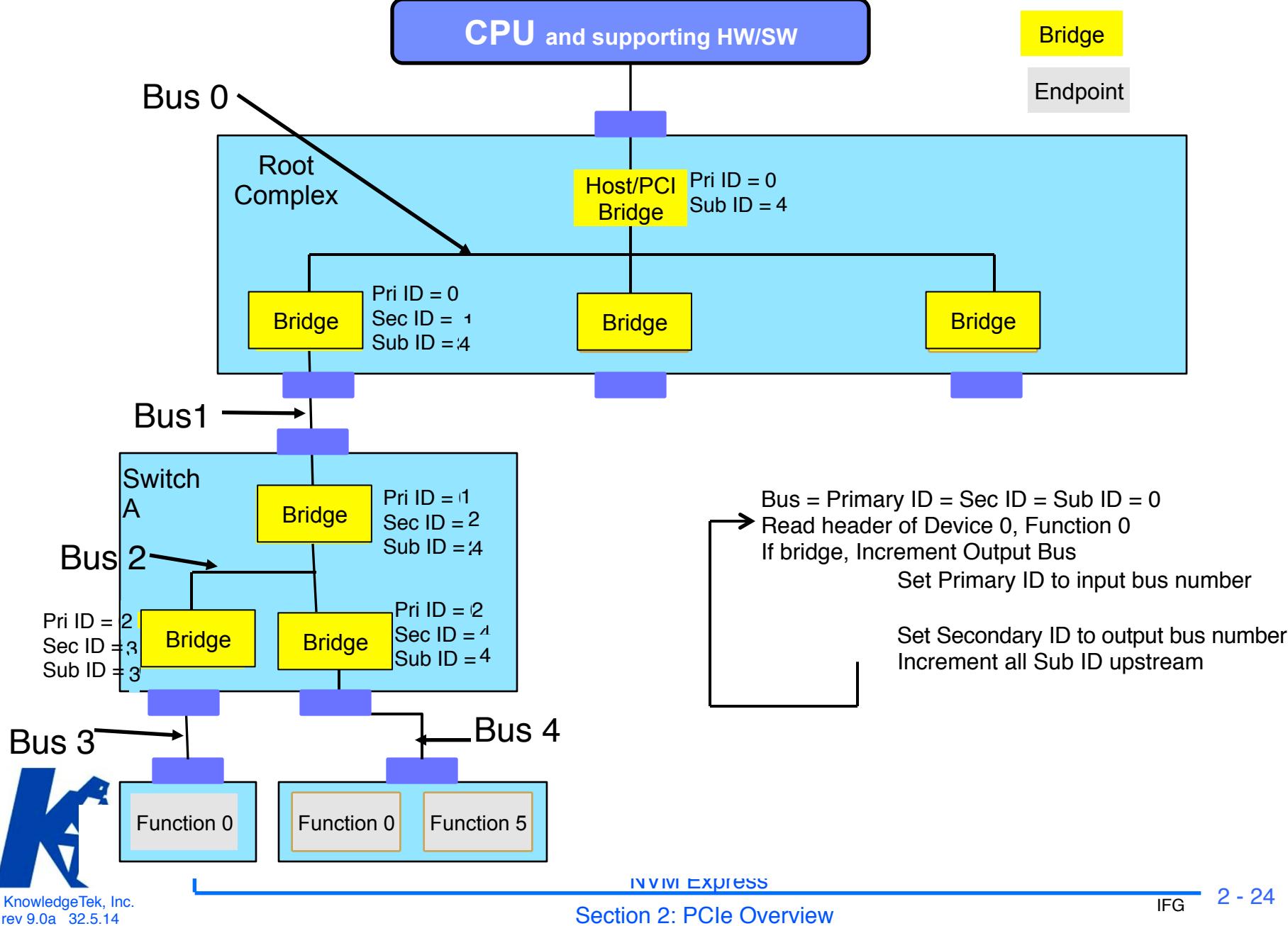
Go deep, then wide

Primary ID = Host side bus

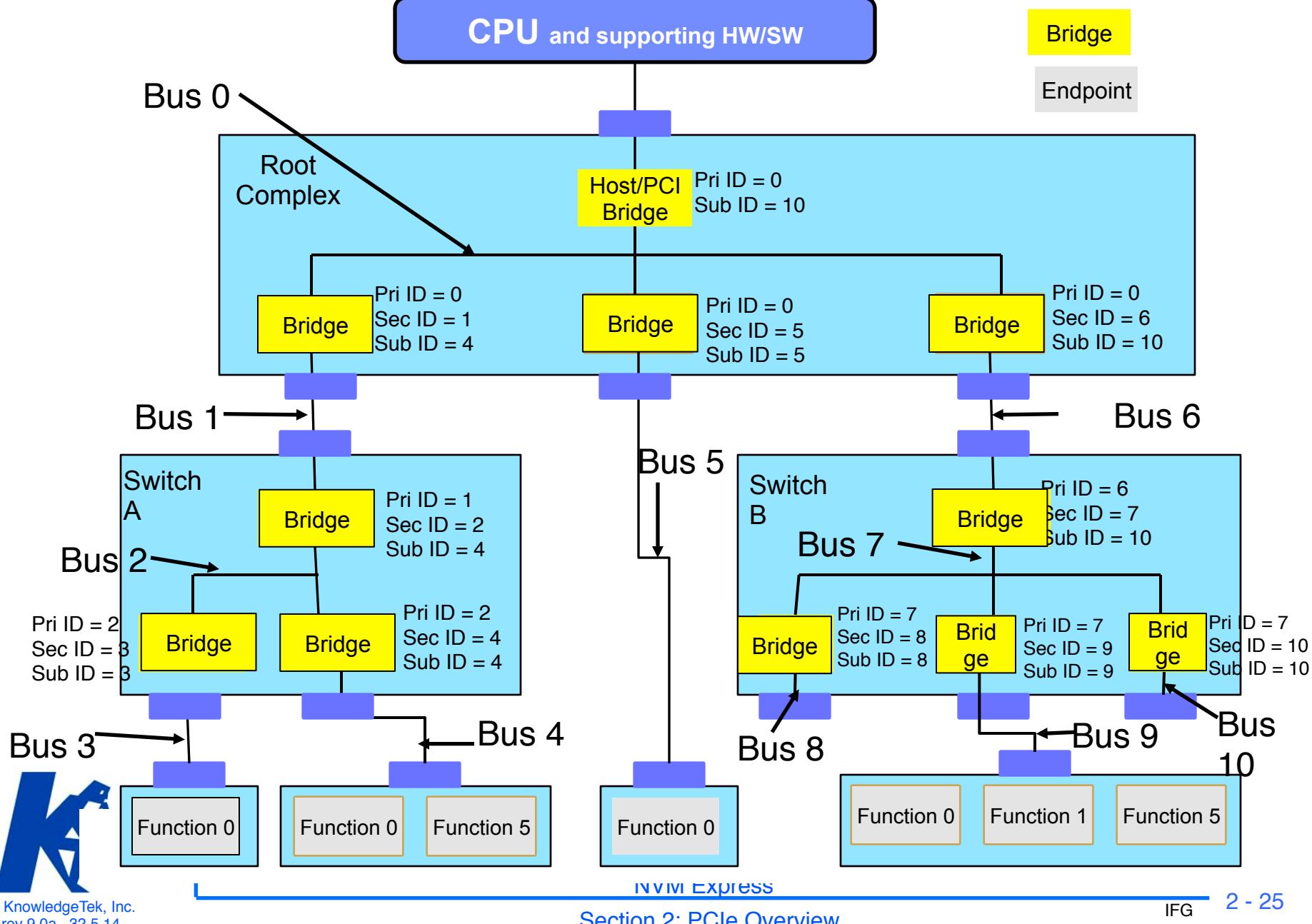
Secondary ID = Device side bus, first bus reached from this bridge

Sub ID = last bus reached from this bridge

Topology after Discovery and Enumeration



Topology after Discovery and Enumeration



Configuration Space Header

Byte (h) Type 1 Configuration Space Header (Bridge)

0	Device ID	Vendor ID			
4	Status	Command			
8	Class Code				
C	BIST	Header Type = 1h			
10	Master Latency Timer				
14	Cache Line Size				
18	BAR0				
20	BAR1				
24	Secondary Latency Timer	Subordinate Bus No	Secondary Bus No.		
28	Primary Bus No.				
32	Secondary status	I/O Limit	I/O Base		
36	Memory Limit	Memory Base			
40	Prefetchable Memory limit	Prefetchable Memory base			
44	Prefetchable Base Upper 32 bits				
48	Prefetchable Limit Upper 32 bits				
52	I/O Limit Upper 16 bits	I/O Base Upper 16 bits			
56	Reserved		Capabilities Pointer		
60	Expansion ROM Base Address				
64	Bridge Control	Interrupt Pin	Interrupt Line		

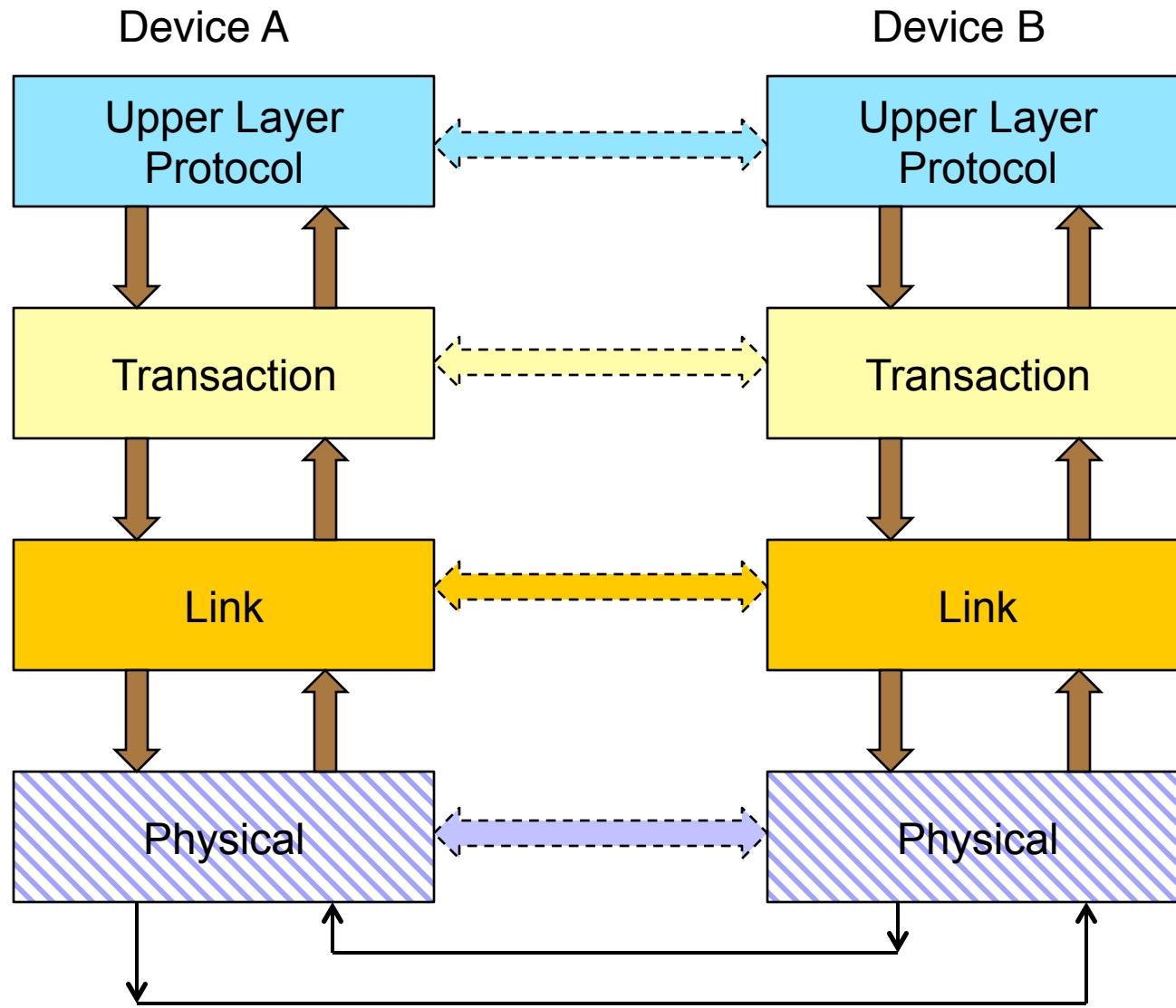
The diagram illustrates the Type 1 Configuration Space Header (Bridge) structure with several annotations:

- ID Routing:** An arrow points to the "Header Type = 1h" field in the Class Code section.
- Non-Prefetchable Memory:** An arrow points to the "I/O Limit" and "I/O Base" fields in the Secondary status section.
- Prefetchable Memory:** An arrow points to the "Memory Base" and "Prefetchable Memory base" fields in the Memory Limit section.
- I/O Addressing:** A large bracket on the right side groups the "I/O Limit" and "I/O Base" fields, the "Memory Base" and "Prefetchable Memory base" fields, and the "I/O Limit Upper 16 bits" and "I/O Base Upper 16 bits" fields, indicating they are part of the I/O Addressing space.

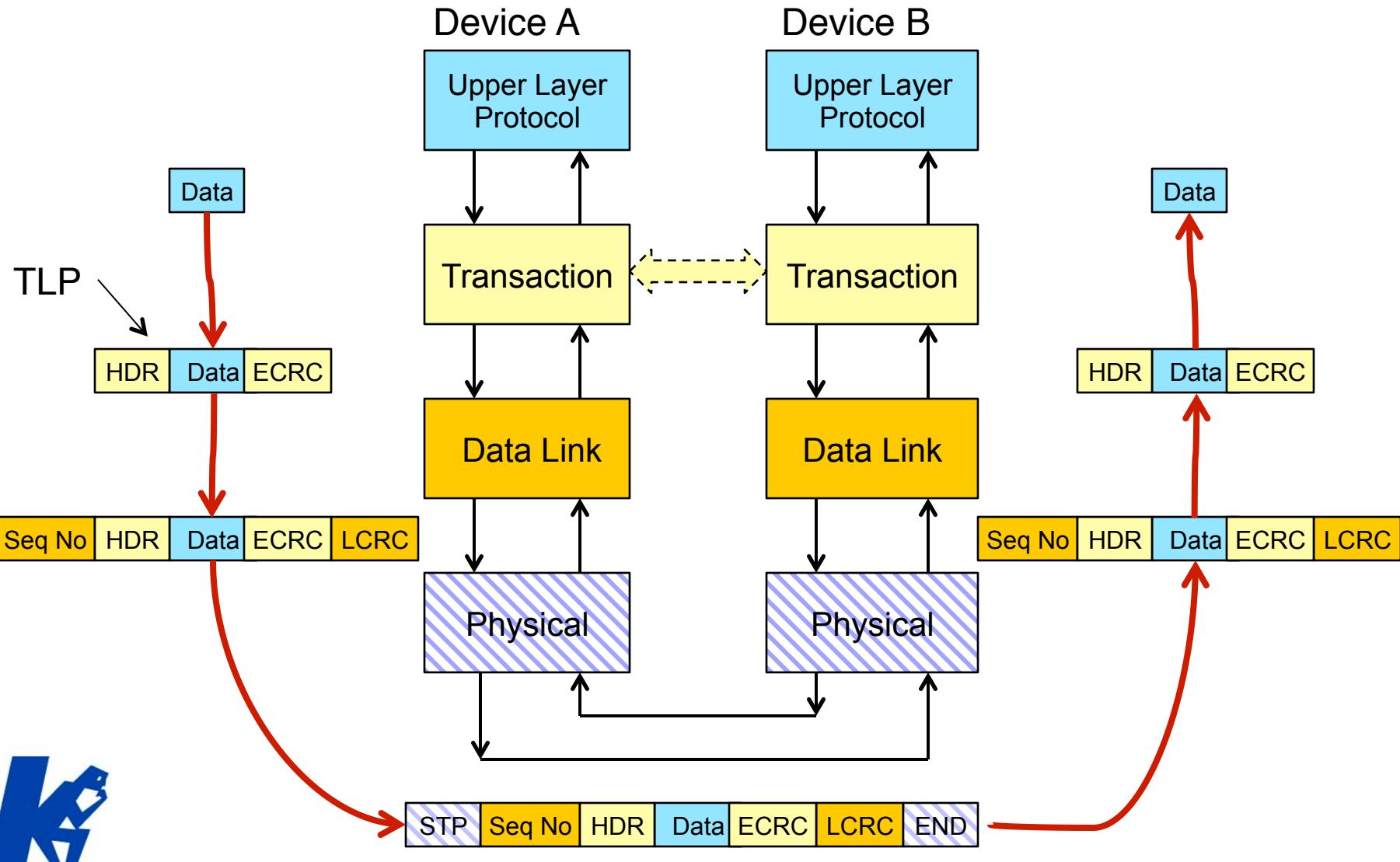
PCI Packets



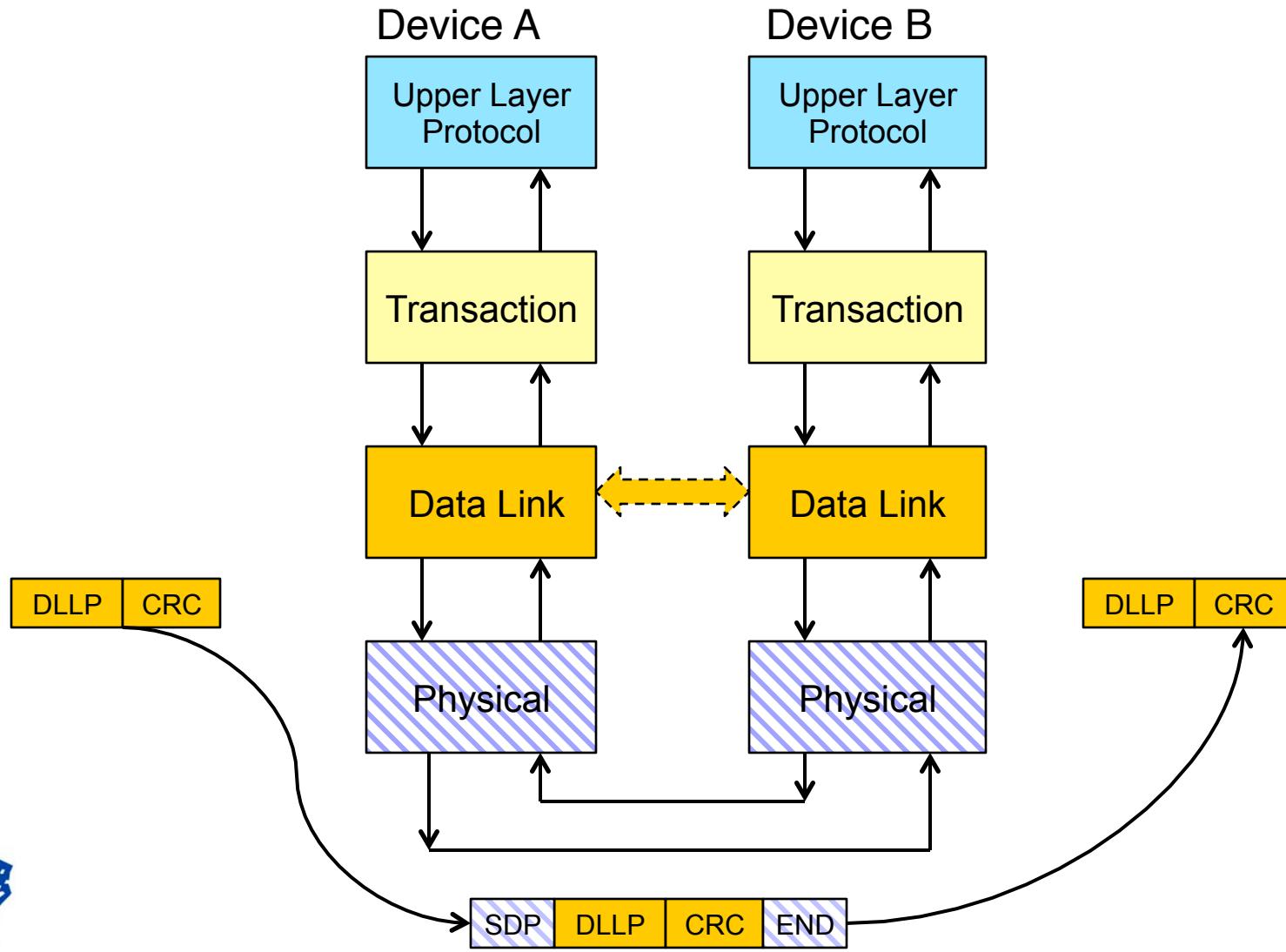
Layers



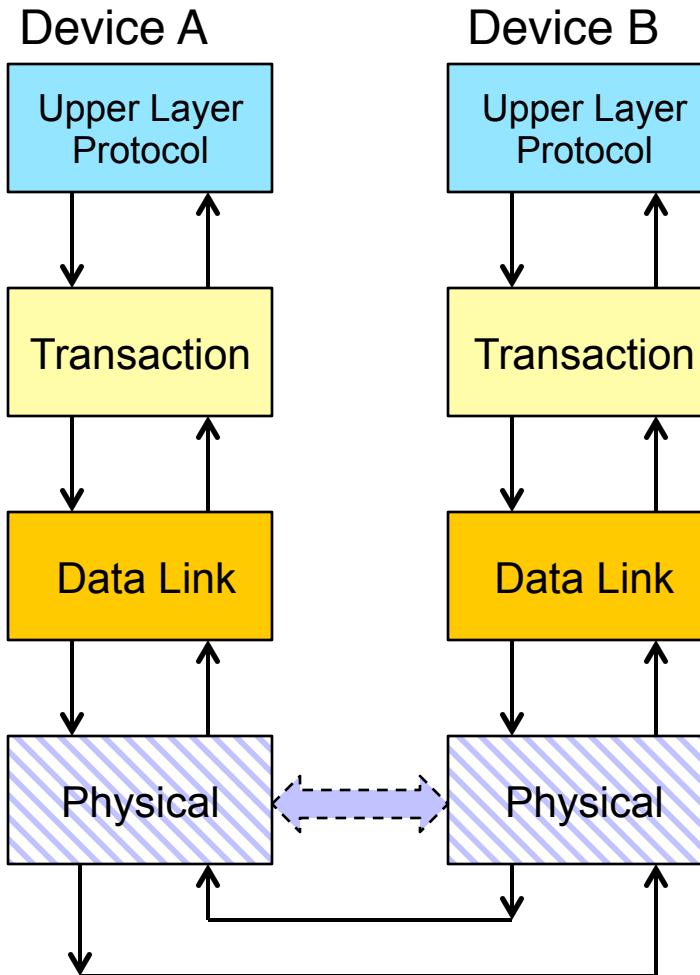
Transaction Layer Packets



Data Link Layer Packets



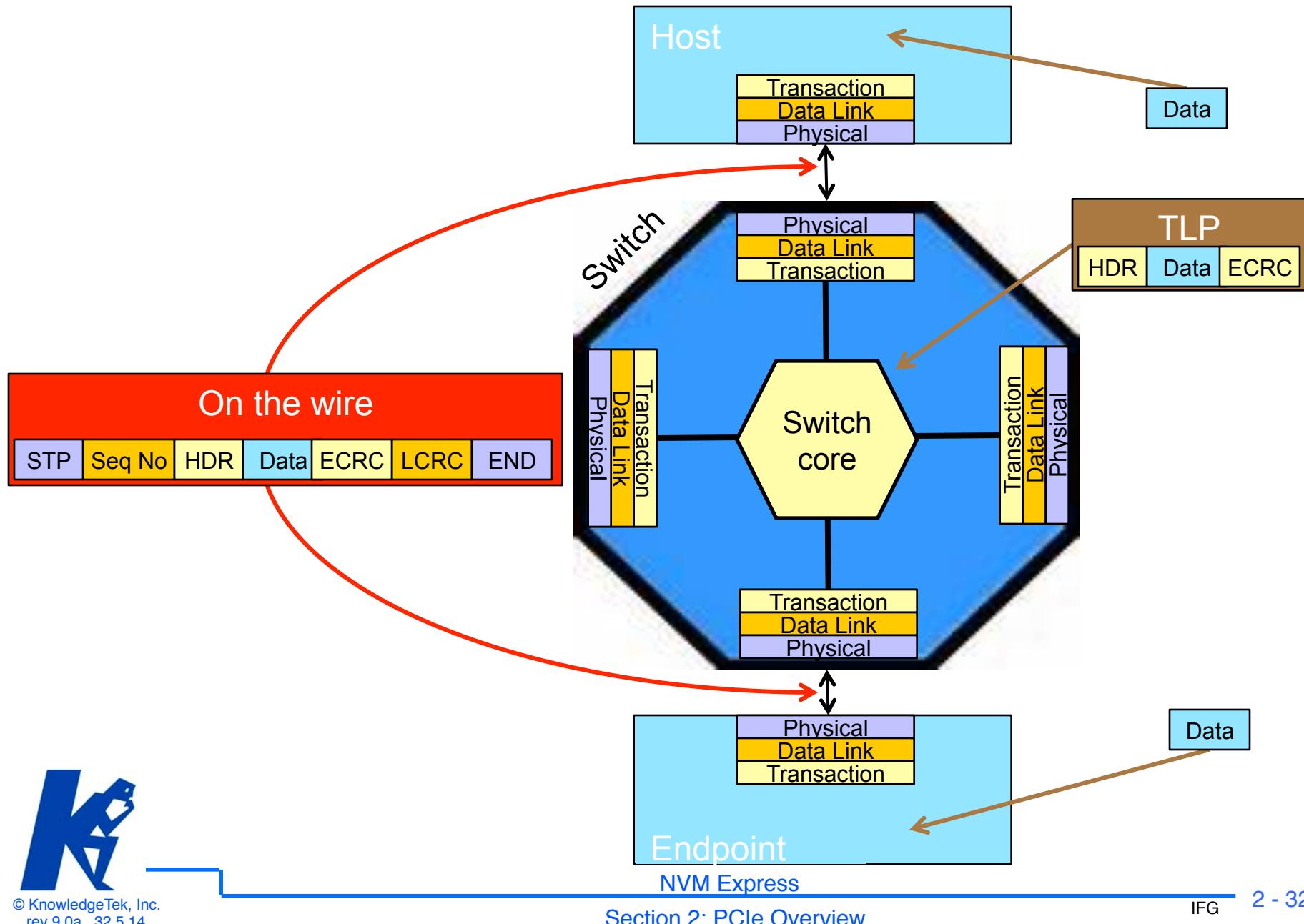
Ordered Sets



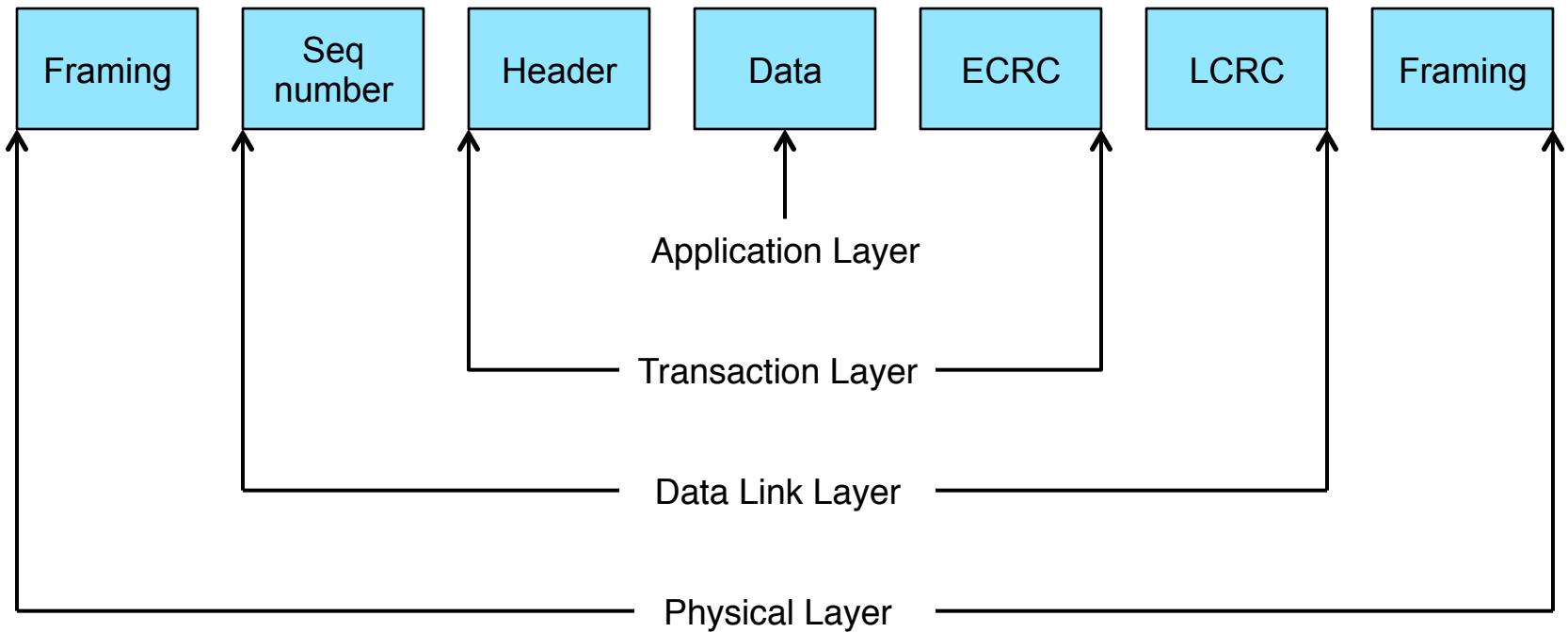
NVM Express

Section 2: PCIe Overview

Layers



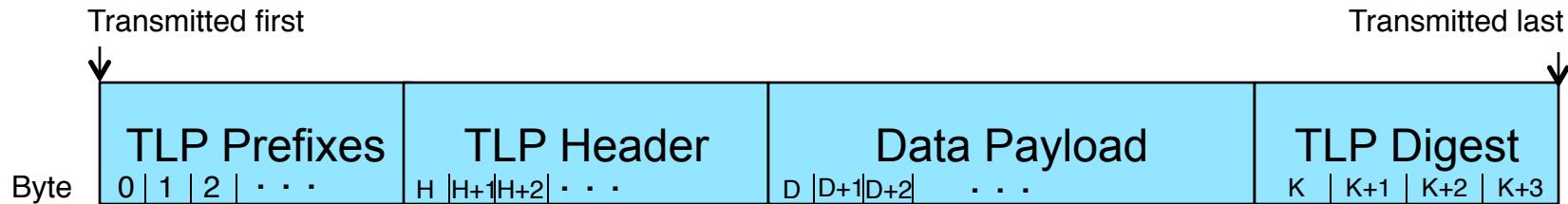
Packet Development



ECRC – End to End protection

LCRC – Link layer protection

Transaction Layer Packet Format Overview



TLP Prefixes

Optional

TLP Header

Type of packet

Routing information

Data Payload

When applicable

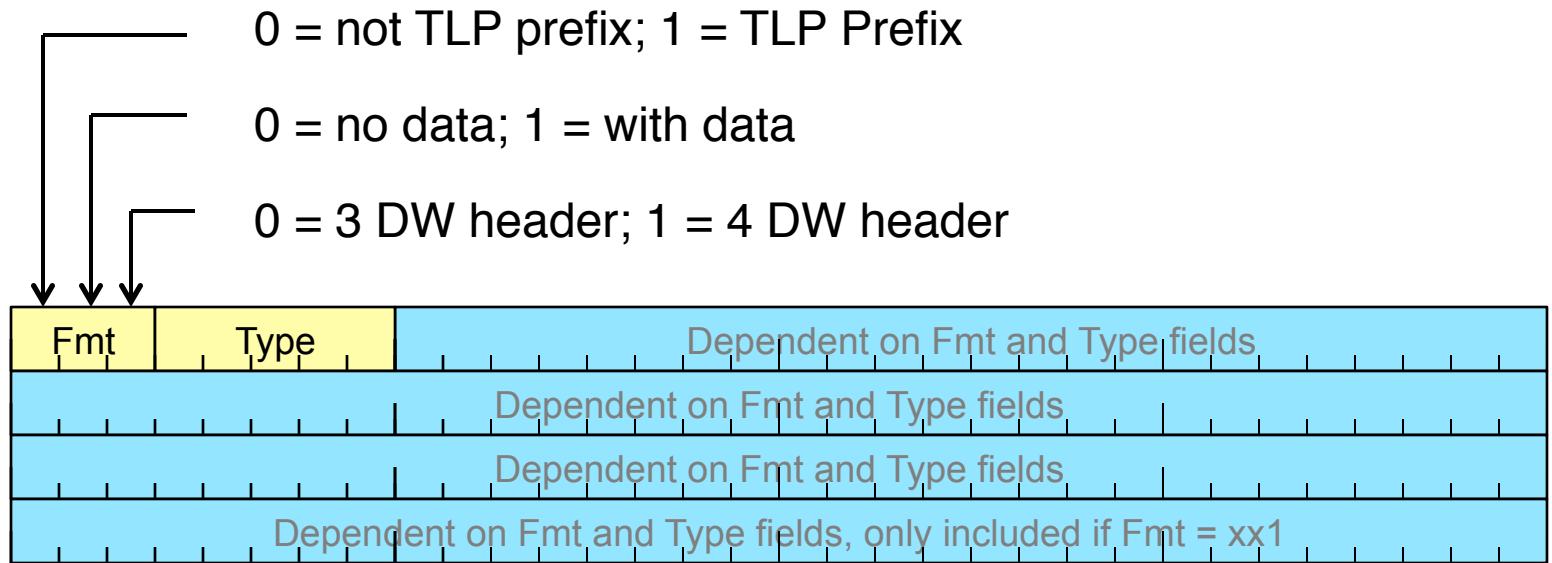
From application layer

TLP Digest

32 bit CRC

Optional

Packet Header Fields



Type field, see next page

Packet Header Fields

FMT (b)	TYPE (h)	Description
00x	00	Memory Read Request
00x	01	Memory Read Request Locked
01x	00	Memory Write Request
000	02	I/O Read Request
010	02	I/O Write Request
000	04	Configuration Ready Type 0
010	04	Configuration Write Type 0
000	05	Configuration Read Type 1
010	05	Configuration Write Type 1
000	1B	Deprecated TLP Type
010	1B	Deprecated TLP Type
001	1 0r2r1r0	Message Request; r2r1r0 specify message routing mechanism
011	1 0r2r1r0	Message Request with data payload
000	0A	Completion without Data
010	0A	Completion with Data
000	0B	Completion for Locked Memory Read without data
010	0B	Completion for Locked Memory Read
01x	0C	Fetch and Add AtomicOp Request
01x	0D	Unconditional Swap AtomicOp Request
01x	0E	Compare and Swap AtomicOP Reqeust
100	0 c3l2l1l0	Local TLP Prefix; l3l2l1l0 specify the Local TLP Prefix Type
100	1 e3e2e1e0	End-End TLP Prefix; e3e2e1e0 specify TLP prefix type

Note: Type field is only 5 bits.

NVM Express

PCI Transactions

PCI Transactions

Memory Read

Memory Write

Configuration read – initialization and configuration

Configuration write – initialization and configuration

Message without Data

Message with Data

I/O read (legacy only)

I/O write (legacy only)

PCI Writes and Reads

Posted (no response)

Memory Writes

Message Requests

Non-posted (response expected)

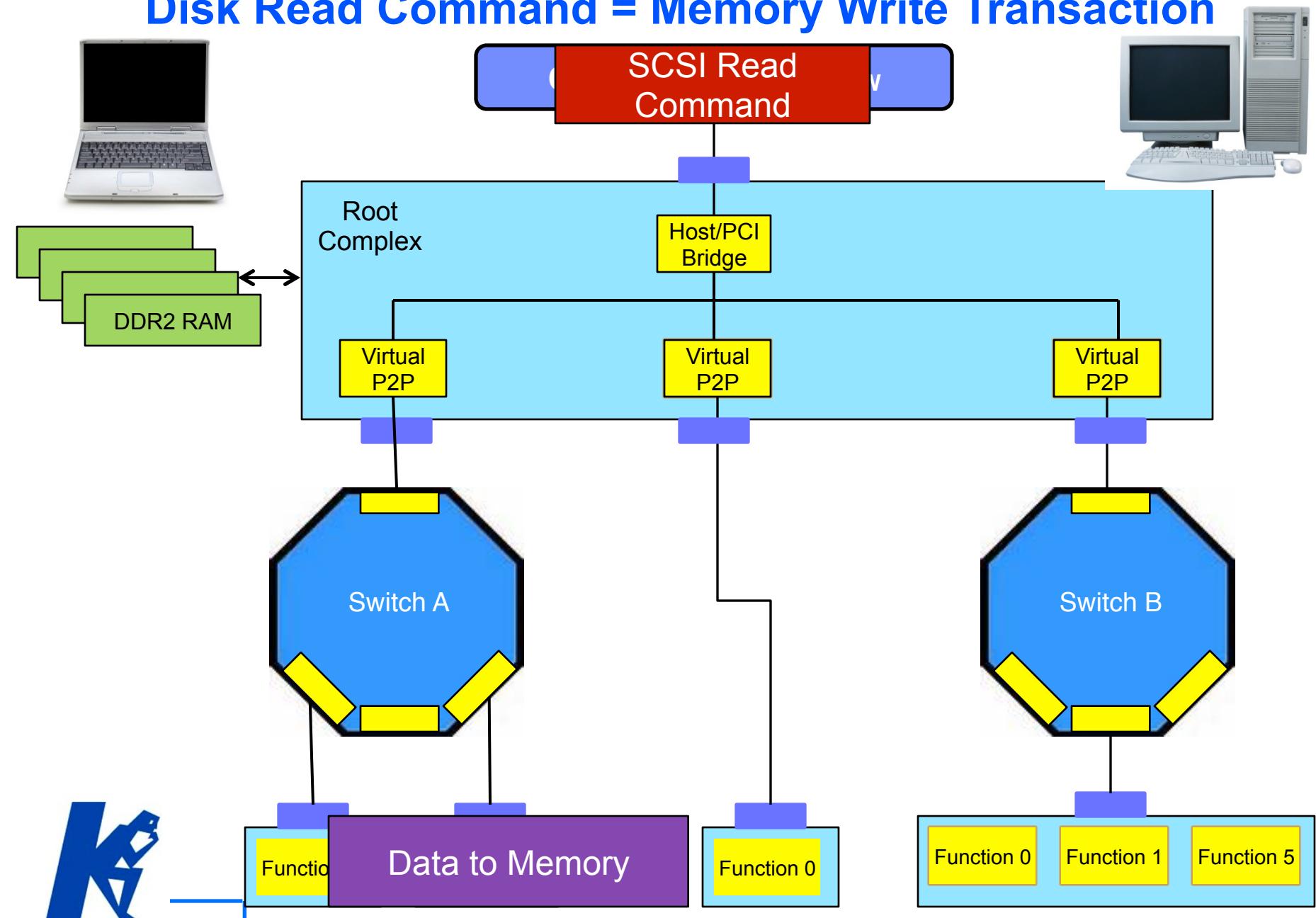
All reads – read data is returned

I/O writes

Configuration writes

Command vs. Transaction

Disk Read Command = Memory Write Transaction



Command vs. Transaction Terminology

Generally

Commands are issued by higher level components

e.g. Read data, write data, Inquiry

Transactions are issued by lower level components

e.g. Posted write, non-posted read, configuration write

Usually issuing commands will be implemented by issuing transactions

PCIe usage

PCIe uses the term “transactions” throughout the specifications

Except:

PCI command register in Configuration Header Space, and

Writing to the Slot Control register in Hot-plug capable DS ports

MSI

MSI-X



Message Signaled Interrupts

A Device Function requests service by writing system-specified Message Data to the system-specified address using Memory Write transaction

2 Systems – MSI and MSI-X

Device may implement both, but only one can be enabled

Each Function may have one MSI capability

Both are disabled following a reset, must be enabled by software or use INT#



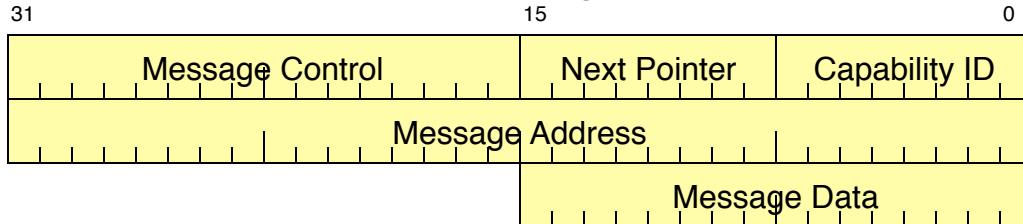
Check out: http://en.wikipedia.org/wiki/Message_Signaled_Interrupts

Interrupt Capability

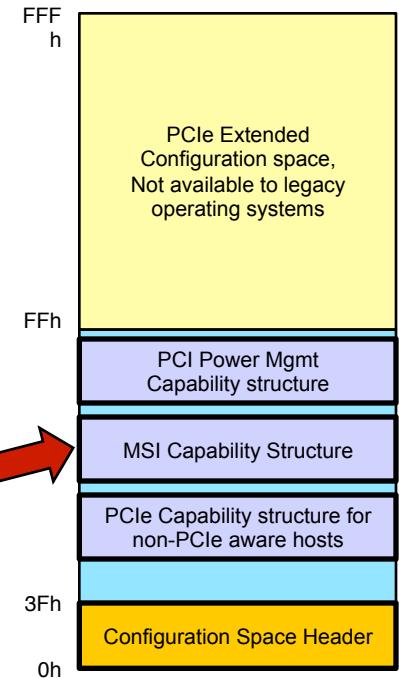
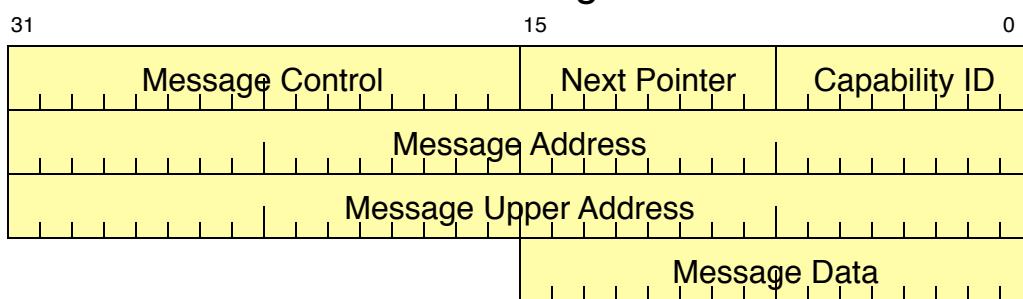
INT	Out-of-Band	Carry over from ISA 4 shared interrupts (INT A/B/C/D) Signaled by messages in PCIe Support encouraged, use discouraged
MSI	In-Band	Added in PCI 2.2 Up to 32 non-shared interrupts Device uses Message Data to identify the sender and IRQ number
MSI-X	In-Band	Added in PCI 3.0 Up to 2048 non-shared interrupts Device uses vector table to identify the sender and IRQ number

MSI Capability Structures

32-bit Message Address

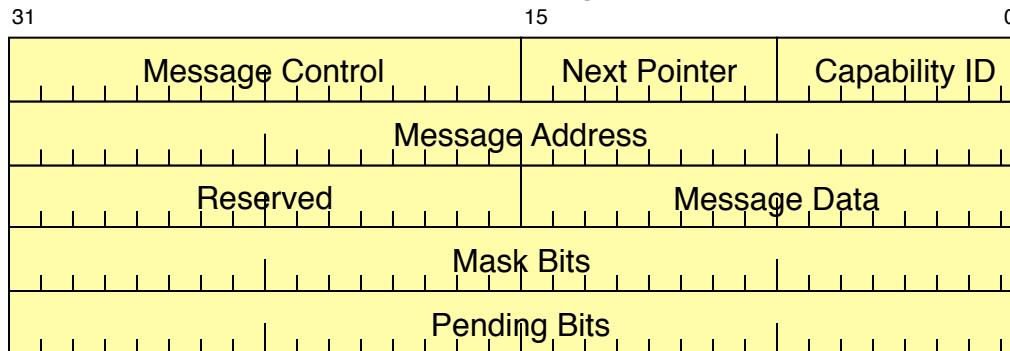


64-bit Message Address

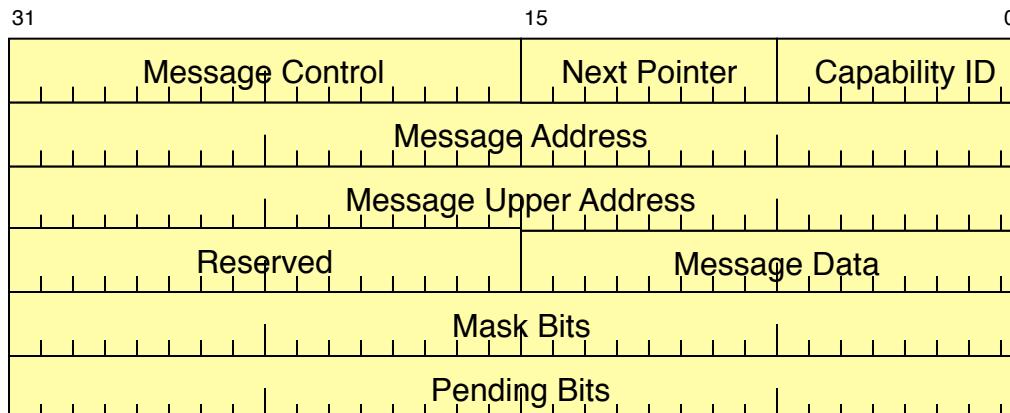


MSI Capability Structures w/Pre-Vector Masking

32-bit Message Address



64-bit Message Address



MSI Capability Structures Field Descriptions

Capability ID – 05h means MSI capable

Next Pointer – Pointer to the next item in the capabilities list (Null for final item)

Message Control –

Bits 15:09	Reserved
8	Pre-Vector masking capable
7	64-bit address capable
6:4	Multiple Message Enable
3:1	Multiple Message Capable
0	MSI Enable

Message Address –

63/31:02	System assigned Address to write interrupt messages
01:00	Reserved

Message Data – System specified identifier for the Device/Function

Mask Bits – For each mask bit set, the Function is prohibited from sending the associated message

Pending Bits – For each Pending Bit set, the Function has a pending associated message

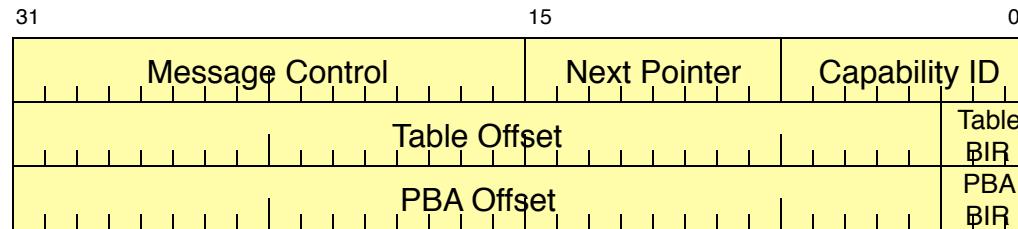


System assigned ID for
this Device/Function

1 – 5 bits to identify
Interrupt number (31:1)

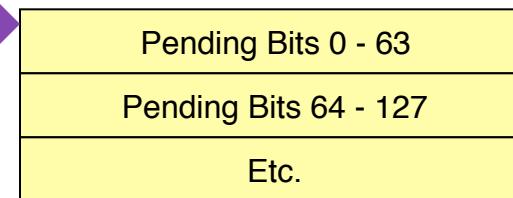
MSI-X Structures

MSI-X Capability Structure

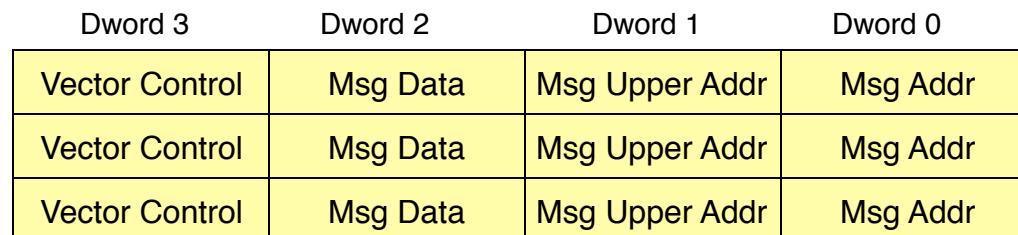


See
next
page

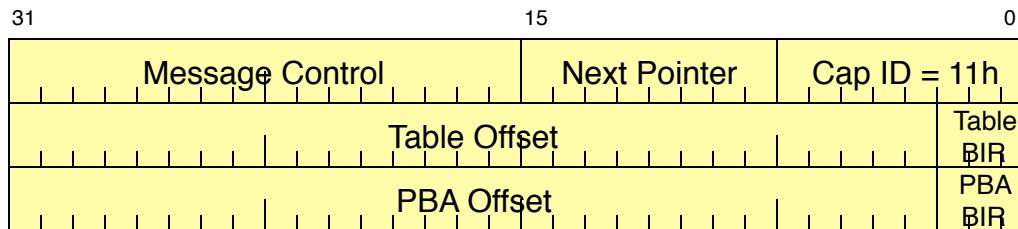
MSI-X Pending Bit Structure



MSI-X Table Structure

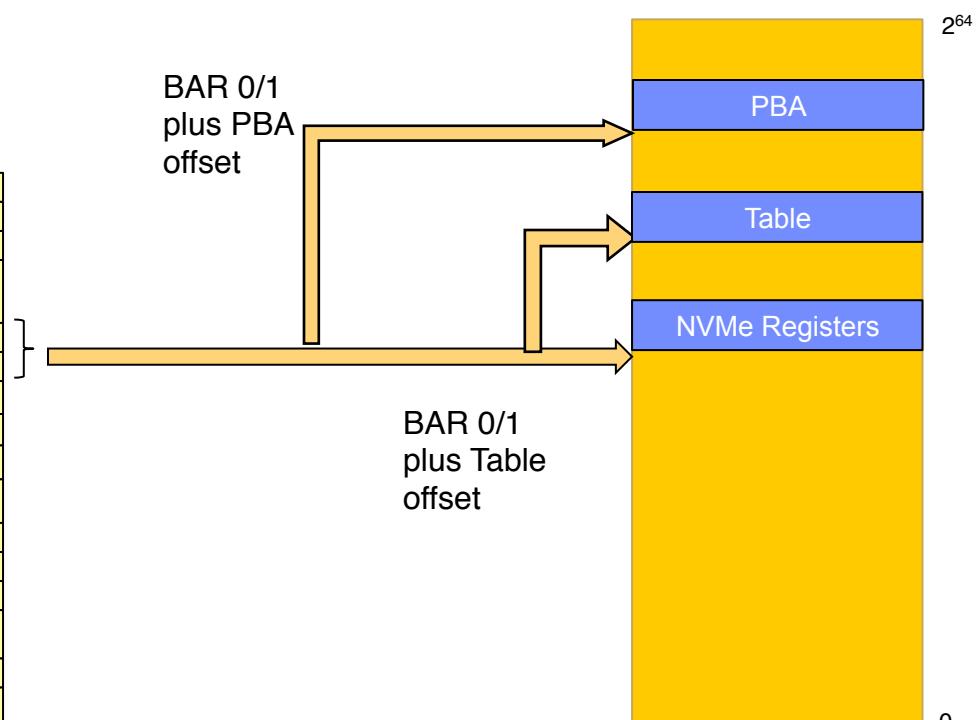


MSI-X Capability Structure



Byte

0	Device ID	Vendor ID
4	Status	Command
8	Class Code	Revision ID
C	BIST	Header Type = 0h
	Master Latency Timer	Cache Line Size
10	BAR0 – MLBAR – NVMe Registers	
14	BAR1 – MUBAR – NVMe Registers	
18	BAR2 – I/O based accesses, if supported	
1C	BAR3 - Reserved	
20	BAR4 – Vendor Specific	
24	BAR5 – Vendor Specific	
28	Cardbus CIS Pointer	
2C	Subsystem ID	Subsystem Vendor ID
30	Expansion ROM Base Address	
34	Reserved	Capabilities Pointer
38	Reserved	
3C	Max Latency = 00h	Min Grant = 00h
	Interrupt Pin	Interrupt Line



MSI-X Capability Structures Field Descriptions

Capability ID – 11h means MSI-X capable

Next Pointer – Pointer to the next item in the capabilities list (Null for final item)

Message Control –

Bits 15	MSI-X enable
14	Function Mask
13:11	Reserved
10:00	Table Size

Table/PBA BIR – Which Function BAR is used to map MSI-X Table into memory space

0	10h
1	14h
2	18h
3	1Ch
4	20h
5	24h
6:7	Reserved

Message Address –

63/31:02	Message Address
01:00	Reserved

Message Data – System specified message data

Mask Bits – For each mask bit set, the Function is prohibited from sending the associated message

Pending Bits – For each Pending Bit set, the Function has a pending associated message

Vector Control –

31:01	Reserved or Steering Table
00	Mask Bit

NVM Express

Check for Understanding

1. What are the PCI/PCI-X and PCIe Transactions?
2. What are the three methods of addressing?
3. Where is the destination address of a packet placed?
4. How does the software locate the memory space for a device?
5. How does the software discover a device's capability?
6. Explain MSI operation.
7. Explain MSI-X operation.

Covered in this Section

PCI/PCIe Concepts

Topology Discovery and Enumeration

PCI Transactions

Notes



Section 3

NVMe Registers



Covered this Section

NVMe Controller Registers

Creating the Admin Queues



Registers

The following registers are addressed from

MLBAR (Lower 32 bits) and
MUBAR (upper 32 bits)

Each controller will have
a set of the registers
listed on the next page.



Type 0 Configuration Space Header (Endpoint)

Device ID	Vendor ID	0
Status	Command	4
Class Code	Revision ID	8
BIST	Header Type = 0h	Master Latency Timer
Cache Line Size		
BAR0 – MLBAR – NVMe Registers		
BAR1 – MUBAR – NVMe Registers		
BAR2 – I/O based accesses, if supported		
BAR3 - Reserved		
BAR4 – Vendor Specific		
BAR5 – Vendor Specific		
Cardbus CIS Pointer		
Subsystem ID	Subsystem Vendor ID	2C
Expansion ROM Base Address		
Reserved		Capabilities Pointer
Reserved		
Max Latency = 00h	Min Grant = 00h	Interrupt Pin
		Interrupt Line

Controller Registers

Start (h)	Length In bytes (h)	Symbol	Name
0	8	CAP	Controller Capabilities
8	4	VS	Version = 0001 0xxxh
C	4	INTMS	Interrupt Mask Set
10	4	INTMC	Interrupt Mask Clear
14	4	CC	Controller Configuration
18	4	R	Reserved
1C	4	CSTS	Controller Status
20	4	NSSR	NVM Subsystem Reset (optional)
24	4	AQA	Admin Queue Attributes
28	8	ASQ	Admin Submission Queue Base Address
30	8	ACQ	Admin Completion Queue Base Address
38	4	CMBLOC	Controller Memory Buffer Location
3C	4	CMBSZ	Controller Memory Buffer Size
40	4	BPINFO	Boot Partition Information
44	4	BPRSEL	Boot Partition Read Select
48	8	BPMBL	Boot Partition Memory Buffer Location
50	EC0	Reserved	Reserved
F00	100	R	Command Set Specific
1000	4	SQ0TDBL	Submission Queue 0 Tail Doorbell (Admin)

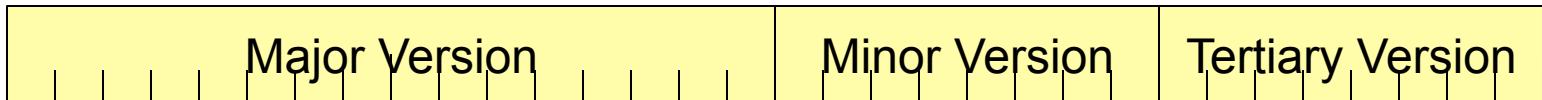
Controller Registers

Start (h)	Length In bytes (h)	Symbol	Name
1000	4	SQ0TDBL	Submission Queue 0 Tail Doorbell (Admin)
1000 + (1 * (4 << CAP.DSTRD))	4	CQ0HDBL	Completion Queue 0 Head Doorbell (Admin)
1000 + (2 * (4 << CAP.DSTRD))	4	SQ1TDBL	Submission Queue 1 Tail Doorbell
1000+(3 * (4 << CAP.DSTRD))	4	CQ1HDBL	Completion Queue 1 Head Doorbell
1000 + (2y * (4 << CAP.DSTRD))	4	SQyTDBL	Submission Queue y Tail Doorbell
1000+((2y+1) * (4 << CAP.DSTRD))	4	CQyHDBL	Completion Queue y Head Doorbell

Controller Capabilities Register

Bits	Type	Reset	Symbol	Description
63:56	RO	0h		Reserved
55:52	RO	Impl Spec	MPSMAX	Memory Page Size Maximum ($2^{(12+ MPSMAX)}$)
51:48	RO	Impl Spec	MPSMIN	Memory Page Size Minimum ($2^{(12+ MPSMIN)}$)
47:46	RO	0h		Reserved
45	RO	0h	BPS	Boot Partition Support
44:37	RO	Impl Spec	CSS	Command Set(s) Supported (bit 37 = NVM)
36	RO	Impl Spec	NSSRS	NVM Subsystem Reset Supported (1 = supported)
35:32	RO	Impl Spec	DSTRD	Doorbell Stride ($2^{(2 + DSTRD)}$)
31:24	RO	Impl Spec	TO	TimeOut: Worst case time in 500 ms units for controller to become ready
23:19	RO	0h		Reserved
18:17	RO	Impl Spec	AMS	Arbitration Mechanism Supported bit 17 – Weighted Round Robin w/Urgent bit 18 – Vendor Specific
16	RO	Impl Spec	CQR	Contiguous Queues Required
15:00	RO	Impl Spec	MQES	Maximum Queue Entries Supported

Version Register



Specification	Date	Major	Minor	Tertiary
NVMHCI 1.0 Gold	4/14/2008		Field not defined	
NVMe 1.1 Gold	10/11/2012	1	1*	0
NVMe 1.1b	7/2/2014	1	1	Reserved
NVMe 1.2 Gold	11/3/2014	1	2	Reserved
NVMe 1.2.1 Gold	6/5/2016	1	2	1
NVMe 1.3 Gold	6/5/2016	1	3	0

* The text indicates this should be a 1.
The table does shows NVMe 1.0 but not 1.1

Controller Configuration Register

Bits	Type	Reset	Symbol	Description
31:24	RO	0h		Reserved
23:20	RW	0h	IOCQES	I/O Completion Queue Entry size (in bytes, power of 2)
19:16	RW	0h	IOSQES	I/O Submission Queue Entry size (in bytes, power of 2)
15:14	RW	0h	SHN	Shutdown Notification 00b – No notification 01b – Normal shutdown notification 10b – Abrupt shutdown notification
13:11	RW	0h	AMS	Arbitration Mechanism Selected 000b – Round Robin 001b – Weighted Round Robin w/ Urgent 111b – Vendor Specific
10:07	RW	0h	MPS	Memory Page Size ($2^{(12+ MPS)}$) (MPS)
06:04	RW	0h	CSS	I/O Command Set Selected (NVM = 000b) (CSS)
03:01	RO	0h		Reserved
00	RW	0h	EN	Enable

Controller Status Register

Bits	Type	Reset	Symbol	Description
31:06	RO	0h		Reserved
05	RO	0h	PP	Processing Paused
04	RW	H/W Init	NSSRO	NVM Subsystem Reset Occurred
03:02	RO	0h	SHST	<p>Shutdown Status</p> <p>00b Normal operation (no shutdown requested) 01b Shutdown processing occurring 10b Shutdown processing complete 11b Reserved</p> <p>Before issuing commands after Shutdown complete, must first issue a reset</p>
01	RO	H/W Init	CFS	Controller Fatal Status
00	RO	0h	RDY	Ready

Interrupt Mask Set and Clear

Use: to set and clear interrupt masks when using:

pin-based interrupts (single vector)

single message MSI

multiple message MSI

(Access when configured for MSI-X is undefined)

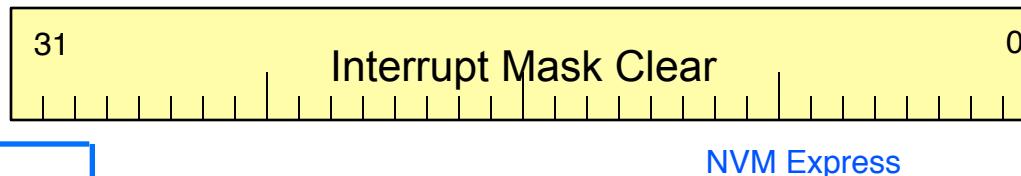
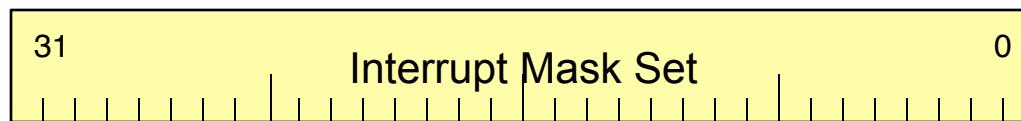
Registers are bit significant

Writing a “1” sets or clears the indicated interrupt

Writing a “0” has no effect

Reading either register returns the current interrupt mask within the controller

These perform a similar function as the MSI Mask registers of PCIe.
Suggestion: use either method of masking interrupts, but only one



Controller NVMe Reset Register

Bits	Type	Reset	Description
31:00	RW	0h	<p>A write of value 4E564D65h (“NVMe”) causes an NVMe subsystem reset.</p> <p>A write of any other value has no effect.</p> <p>Register returns 0h when read.</p>



Controller Memory Buffer

Added in NVMe 1.2

Memory on the controller that can function for various purposes

Defined by:

CMBLOC – Controller memory location register

BIR + offset in CMBSZ increments

CMBSZ – Controller memory buffer size

Possible Uses:

Submission Queue (Admin or I/O)

Completion Queue (Admin or I/O)

PRP and SGL

Small quantities of data

Metadata

Host writes SQ entries and/or PRP/SGL entries directly to controller's internal memory (Posted Write).
Controller does not have to use a read to fetch.

Benefit!!!

Also see Host
Memory Buffer
under Admin
Commands:
Set Features



Controller Memory Buffer Location Register

Bits	Type	Reset	Symbol	Description
31:12	RO	Impl Spec	OFST	Offset in multiples of Size Unit from indicated BAR
11:03	RO	0h	R	Reserved
2:0	RO	Impl Spec	BIR	Base Indicator Register

BIR – BAR Index
Register
BAR – Base
Address Register



Controller Memory Buffer Size Register

Bits	Type	Reset	Symbol	Description
31:12	RO	Impl Spec	SZ	Indicates size in multiples of Size Unit
11:08	RO	Impl Spec	SZU	Size Unit 0h 4KB 1h 64KB 2h 1MB 3h 16MB 4h 256MB 5h 4GB 6h 64GB
7:5	RO	0	R	Reserved
4	RO	Impl Spec	WDS	Write Data Support (Data and Metadata)
3	RO	Impl Spec	RDS	Read Data Support (Data and Metadata)
2	RO	Impl Spec	LISTS	PRP and SGL Support
1	RO	Impl Spec	CQS	Completion Queue Support (must be contiguous)
0	RO	Impl Spec	SQS	Submission Queue Support (must be contiguous)

Boot Partition Information

Bits	Type	Reset	Symbol	Description
31	RO	Impl Spec	ABPID	Active Boot Partition ID
30:16	RO	0h		Reserved
25:24	RO	0	BRS	Boot Read Status
				00b – No Boot Partition Read operation requested
				01b – Boot Partition read in progress
				10b – Boot Partition read competed successfully
				11b – Error completion Boot Partition Read
23:15	RO	0h		Reserved
14:00	RO	Impl Spec	BPSZ	Boot Partition Size (in multiples of 128KB)

Boot Partition Read Select

Bits	Type	Reset	Symbol	Description
31	RW	0h	BPID	Boot Partition ID
30	RO	0h		Reserved
29:10	RW	0h	BPROF	Boot Partition Read Offset (multiples of 4KB increments)
9:0	RW	0h	BPRSZ	Boot Partition Read Size (multiple of 4KB units)

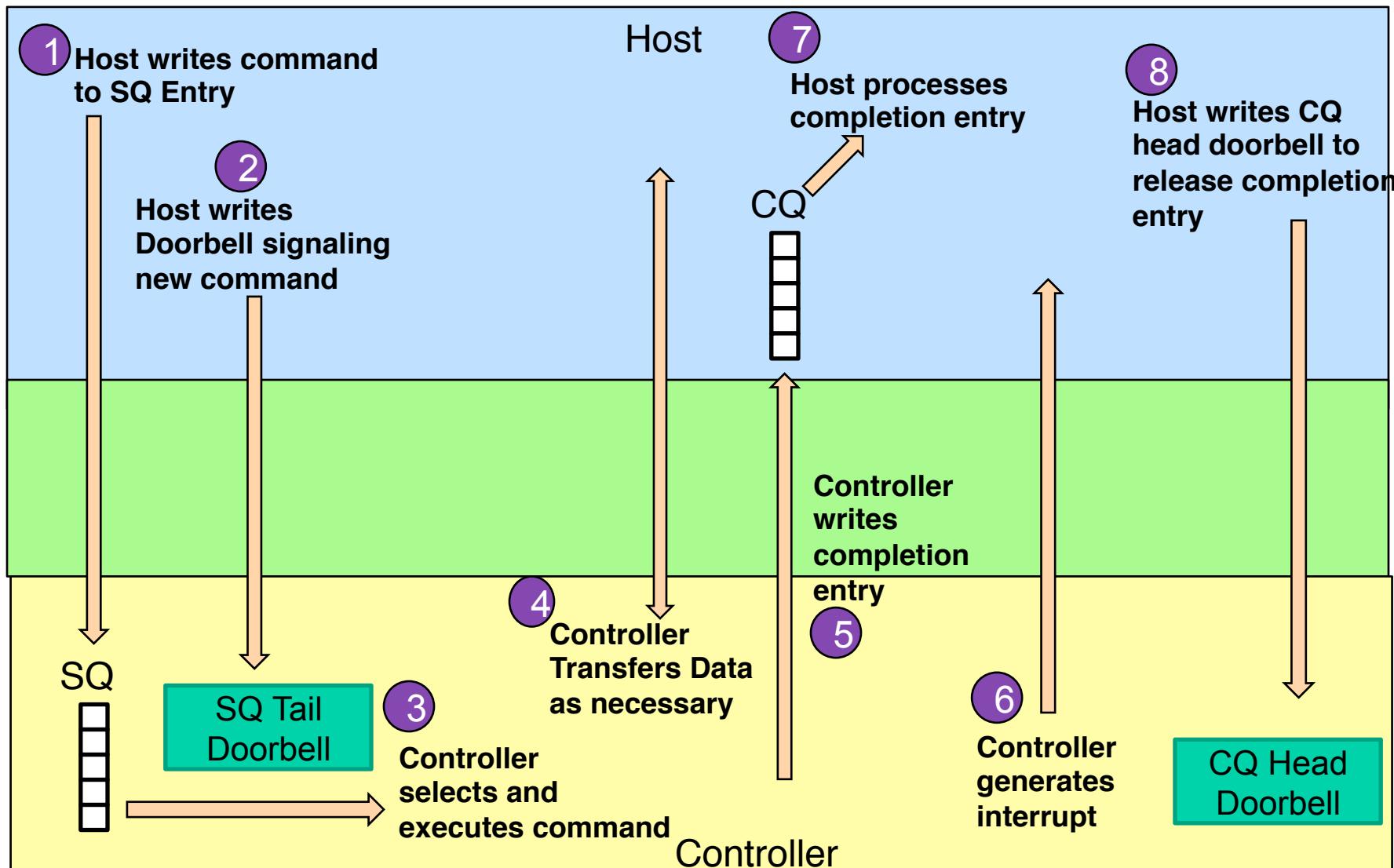
Boot Partition Memory Buffer Location

Bits	Type	Reset	Symbol	Description
63:12	RW	0h	BMBBA	Boot Partition Memory Buffer Base Address
11:0	RO	0h		Reserved

Boot Partition Information

Bits	Type	Reset	Symbol	Description
31	RO	Impl Spec	ABPID	Active Boot Partition ID
30:16	RO	0h		Reserved
25:24	RO	0	BRS	Boot Read Status
				00b – No Boot Partition Read operation requested
				01b – Boot Partition read in progress
				10b – Boot Partition read competed successfully
				11b – Error completion Boot Partition Read
23:15	RO	0h		Reserved
14:00	RO	Impl Spec	BPSZ	Boot Partition Size (in multiples of 128KB)

NVMe Command Processing (Picture – Informative)



Creating the ADMIN Queues



Steps in Creating Admin Queues

1. PCIe discovers device, reads Config Header and writes BARs
2. NVMe S/W reads PCIe Config Header offset 09h (Class code)
3. NVMe S/W reads BARs to find NVMe registers
4. NVMe S/W writes to offset 14h to disable Controller
5. NVMe S/W writes to offset 24h (Admin Queue Attributes)
6. NVMe S/W writes to offset 28h (Submission Queue base address)
7. NVMe S/W writes to offset 30h (Completion Queue base address)
8. NVMe S/W enables the controller by setting CC.EN = 1b
9. NVMe S/W polls CSTS.RDY = 1b to indicate controller is ready

Creating Admin Queues

Teledyne LeCroy PETracer(TM) - PCI Express Protocol Analyzer - [C:\Users\Public\Documents\L...\Z3_drive_emulation_boot_and_play_video.pex]												
File Setup Record Generate Report Search View Tools Window Help												
File Setup Record Generate Report Search View Tools Window Help												
NVM 2	R→ x8	2.5	RequesterID 000:00:0	CC 0	EN NVM command set	CSS 0	MPS b00	AMS b00	SHN 6	IOSQES 6	IOCQES 40.000 ns	Time Stamp 0013 . 128 987 976 s
NVM 3	R→ x8	2.5	RequesterID 000:00:0	CompleterID 001:00:0	CAP 65535	MQES 1	CQR 1	AMS 0	TO 313.324 us	Time Delta 0013 . 128 988 016 s	Time Stamp	
NVM 4	R→ x8	2.5	RequesterID 000:00:0	AQA 127	ASQS 127	ACQS 1.468 us	Time Delta 0013 . 129 301 340 s	Time Stamp				
NVM 5	R→ x8	2.5	RequesterID 000:00:0	ASQ 0x00000002	ASQB AddressHi 0x1F0E0000	ASQB AddressLow 2.872 us	Time Delta 0013 . 129 302 808 s	Time Stamp				
NVM 6	R→ x8	2.5	RequesterID 000:00:0	ACQ 0x00000002	ACQB AddressHi 0x1F0E2000	ACQB AddressLow 2.872 us	Time Delta 0013 . 129 305 680 s	Time Stamp				
NVM 7	R→ x8	2.5	RequesterID 000:00:0	CC 1	EN NVM command set	CSS 0	MPS b00	AMS b00	SHN 0	IOSQES 0	IOCQES 15.232 us	Time Stamp 0013 . 129 308 552 s
NVM 8	R→ x8	2.5	RequesterID 000:00:0	CompleterID 001:00:0	CSTS 1	RDY 0	CFS b00	SHST 11.573 ms	Time Delta 0013 . 129 323 784 s	Time Stamp		
NVM 9	R→ x8	2.5	RequesterID 000:00:0	SQyTDBL 0x0001	Admin SQT QID = 0 0x0001	Time Delta 175.784 us	Time Stamp 0013 . 140 896 512 s	Time Stamp				



Admin Queue Attributes

Bits	Type	Reset	Symbol	Description
31:28	RO	0h		Reserved
27:16	RW	0h	ACQS	Admin Completion Queue Size
15:12	RO	0h		Reserved
11:00	RW	0h	ASQS	Admin Submission Queue Size

Queue Size = Number of Entries
Range 2 <-> 4096

NVMe Class Codes

Class	Description
00h	Devices built pre PCI 2.0
01h	Mass storage controller
02h	Network controller
03h	Display controller
04h	Multimedia Device
05h	Memory Controller
06h	Bridge Device
07h	Simple Comm Controllers
08h	Base System Peripherals
09h	Input Devices
0Ah	Docking Stations
0Bh	Processors
0Ch	Serial Bus Controllers
0Dh	Wireless Controller
0Eh	Intelligent Controller
0Fh	Satellite Comm Controller
10h	Encryption Controller
11h	Signal Processing Controller
12h	Processing Accelerator
13 - FEh	Reserved
FFh	Misc.

Sub-Class	Description
00h	SCSI Controller
01h	IDE Interface
02h	Floppy Disk controller
03h	IPI Controller
04h	RAID Controller
05h	ATA Controller
06h	SATA Controller
07h	SAS Controller
08h	Solid State Controller
80h	Other

Programming
Interface

Code	Description
00h	No PI Defined
01h	NVMHCI 1.0 (obs)
02h	Enterprise NVMe 1.X



Check
PCI Code and ID Assignment
for latest assignments

Check for Understanding

1. What are the NVMe Registers?
2. Where are the NVMe Registers located?
3. How are the Admin Queues created?



Covered this Section

NVMe Controller Registers

Creating the Admin Queues



Notes



Section 4

Commands Formats



Covered in this Section

General Command format

Command Double Word 0

MetaData

Scatter/Gather Lists

SGL

PRP

Status

Interrupts



Common Command Fields

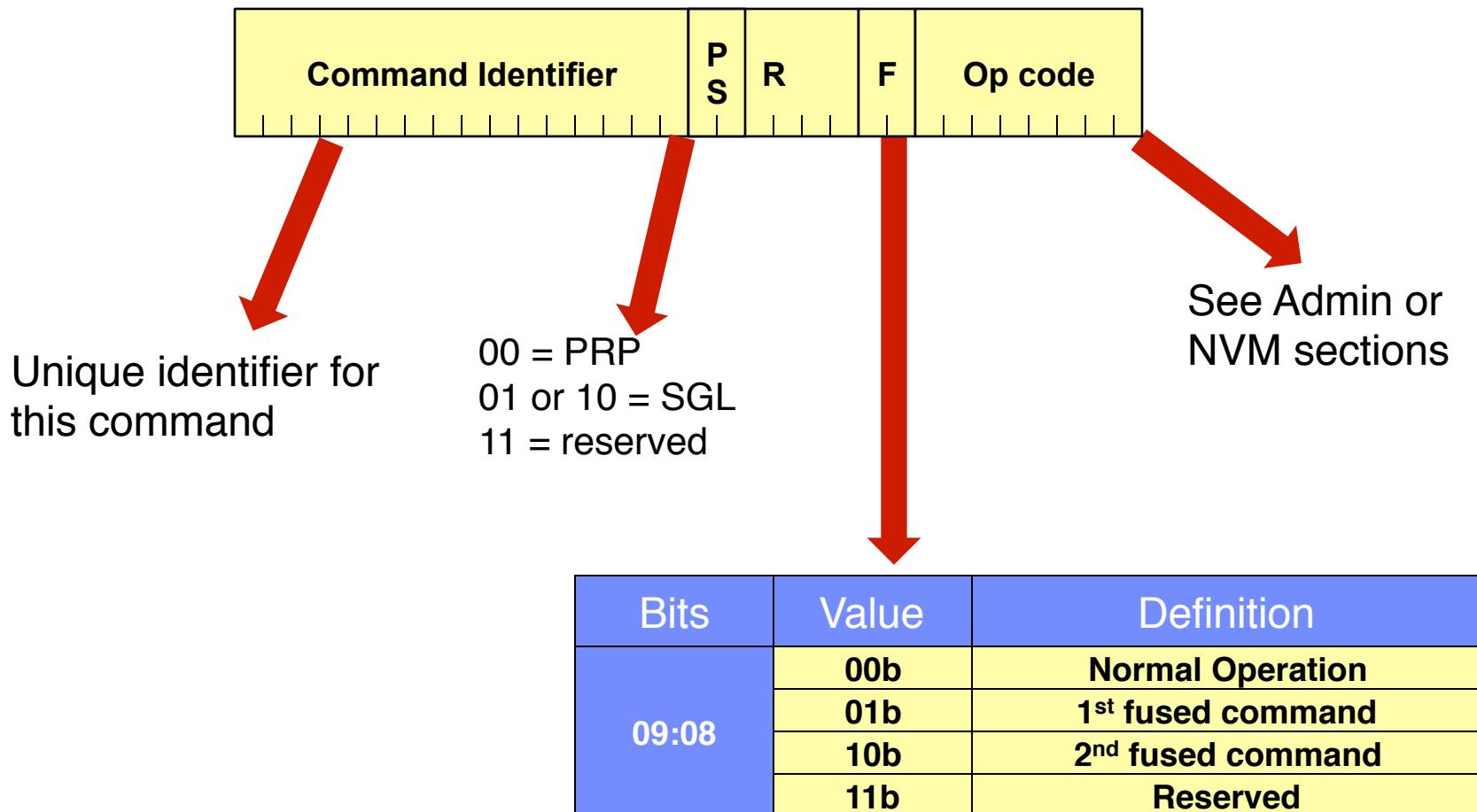


ADMIN and NVMe Command Format

Dword	Bytes	31	Name	0
0	03:00		Command ID (CID)	P S
1	07:04		Namespace Identifier (NSID)	Res
2	11:08		Reserved	
3	15:12			
4	19:16		Metadata Pointer (MPTR) – Address of physical buffer for metadata	
5	23:20			
6	27:24		PRP Entry 1 (PRP1)	
7	31:28			or SGL 1
8	35:32		PRP Entry 2 (PRP2)	
9	39:36			
10	43:40			
11	47:44			
12	51:48		Command specific fields	
13	55:52			
14	59:56			
15	63:60			



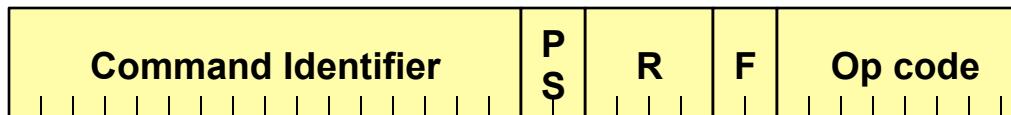
Command Dword 0



Fused commands are created by joining two commands to be processed one immediately after the other.

Command Dword 0 – Fused Operations

Optional



Bits	Value	Definition
09:08	00b	Normal Operation
	01b	1 st fused command
	10b	2 nd fused command
	11b	Reserved

Rules

- Fused commands execute in the order indicated, as though no other operations have been executed between them
- Operation ends when an error is encountered in either command
 - If 1st command fails, 2nd command is aborted
 - If 2nd command fails, completion of 1st command is sequence specific
- LBA range, if used, shall be the same for both commands
- Commands shall be sequential in same SQ
- Commands shall have only one doorbell update
- Abort is issued for each command
- CQ entry is posted for each command



Namespaces



Namespaces

Dword	Bytes	31	Name	0
0	03:00		Command ID (CID)	P S
1	07:04		Namespace Identifier (NSID)	Res
2	11:08		Reserved	F
3	15:12			OP Code
4	19:16		Metadata Pointer (MPTR) – Address of physical buffer for metadata	
5	23:20			
6	27:24		PRP Entry 1 (PRP1)	
7	31:28			or SGL 1
8	35:32		PRP Entry 2 (PRP2)	
9	39:36			
10	43:40			
11	47:44			
12	51:48		Command specific fields	
13	55:52			
14	59:56			
15	63:60			



Namespace

NVMe Definition: A collection of logical blocks that range from 0 to the capacity of the namespace -1. The controller supports access to any valid namespace from any I/O Submission Queue.

Wikipedia Definition: A container for a set of identifiers (names), and allows the disambiguation of homonym identifiers residing in different namespaces.

Purpose:

- A flexible means to categorize your data by application
- A means to allow different data characteristics for on a single endpoint
- Allows separate data storage for each user
- Provide a different NS for each generation of a database or program
- Provide a different NS for each SR-IOV System Image

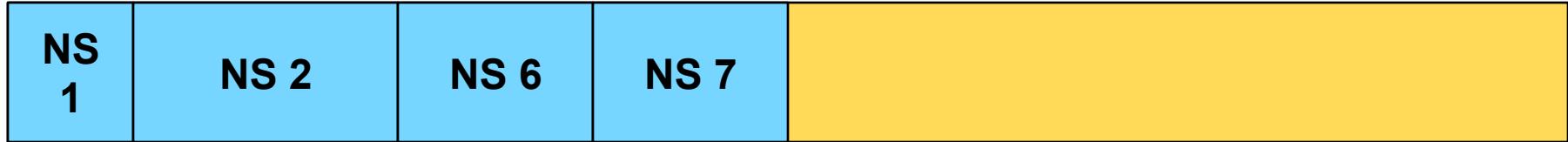


NameSpace (ID) – Conditions

LBA
0

Physical Device

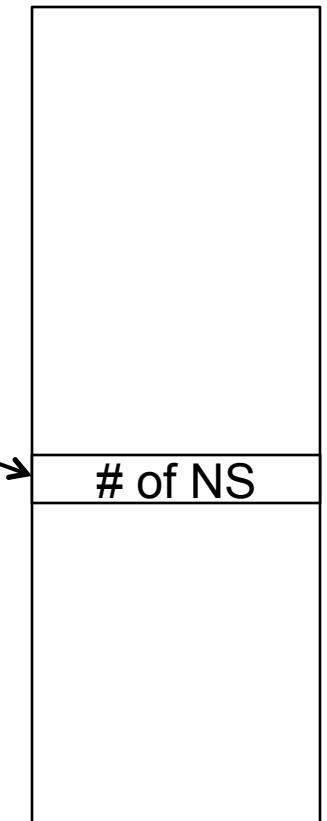
LBA
Max



Valid Namespace (ID) – Any namespace ID between 1 and # of NS (inclusive)

Identify Controller

0



Valid NSID Type	The associated NS
Active	Is attached to this controller
Inactive	Is not attached to this controller
Allocated	Exists in the NVM subsystem
Unallocated	Does not exist in the NVM subsystem

NVM Express

NameSpaces (from NVMe Specification)

Management of namespaces is defined in NVMe 1.2 specification

Private Namespace Definition

A namespace that can be accessible by only one controller

Shared Namespace Definition

A namespace that is accessible by two or more controllers in an NVM Subsystem



Namespace Characteristics

The Namespace Management Command sets the following characteristics:

Namespace Size (total size in Logical Blocks)

Namespace Capacity (Maximum number of LBs that may be allocated)

Formatted LBA Size (indicates one of the 16 supported LBA Formats)

Relative Performance (Best, Better, Good, Degraded)

LBA Data Size in bytes

Metadata Size in bytes

End-to-End Data Protection Type

NS Multipath and Sharing capabilities

Within a Namespace, LBA Range Types may be defined:

Starting LBA

Number of Logical Blocks

128 bit GUID

Type

File System

RAID

Cache

Page/Swap file

GUID –
Globally Unique
IDentifier



Namespace

An NVMe controller is associated with a single PCI Function

Controller capabilities are indicated in CAP register

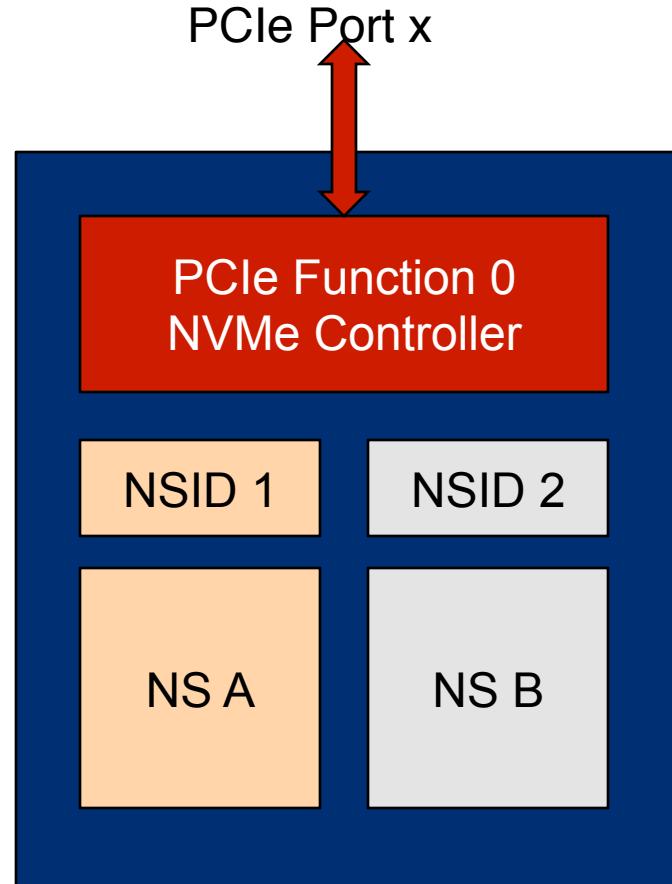
Identify Controller Data Structure indicates capabilities and settings that apply to the entire controller

Identify Namespace Data Structure indicates capabilities and settings that are specific to a particular namespace

Each NameSpace has a EUI64/128 bit Globally Unique Identifier in the NS. Used to determine if more than one controller has access to a namespace.



Controller with two Private Namespaces



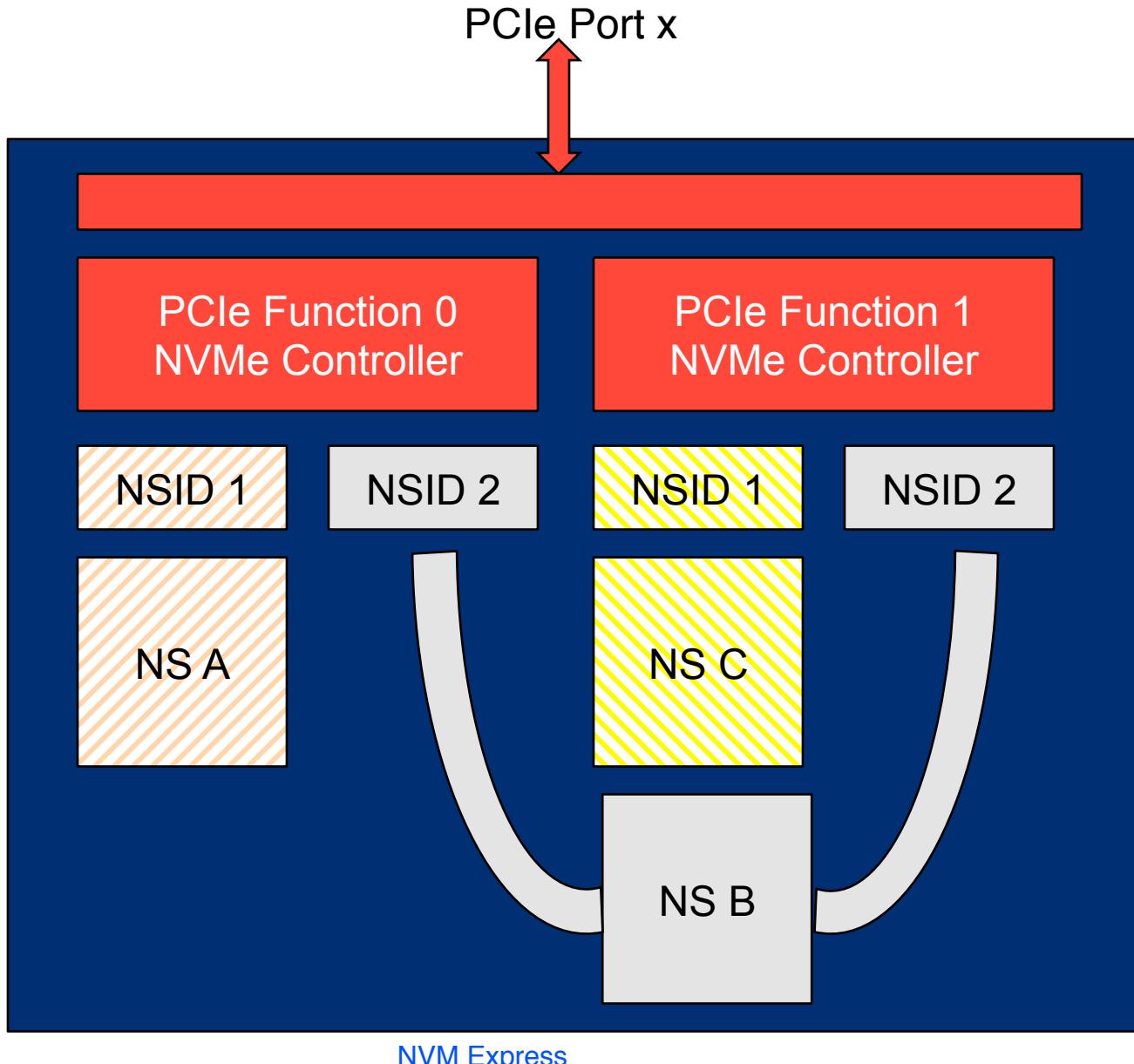
Namespaces are not shared

Multi-path is not used

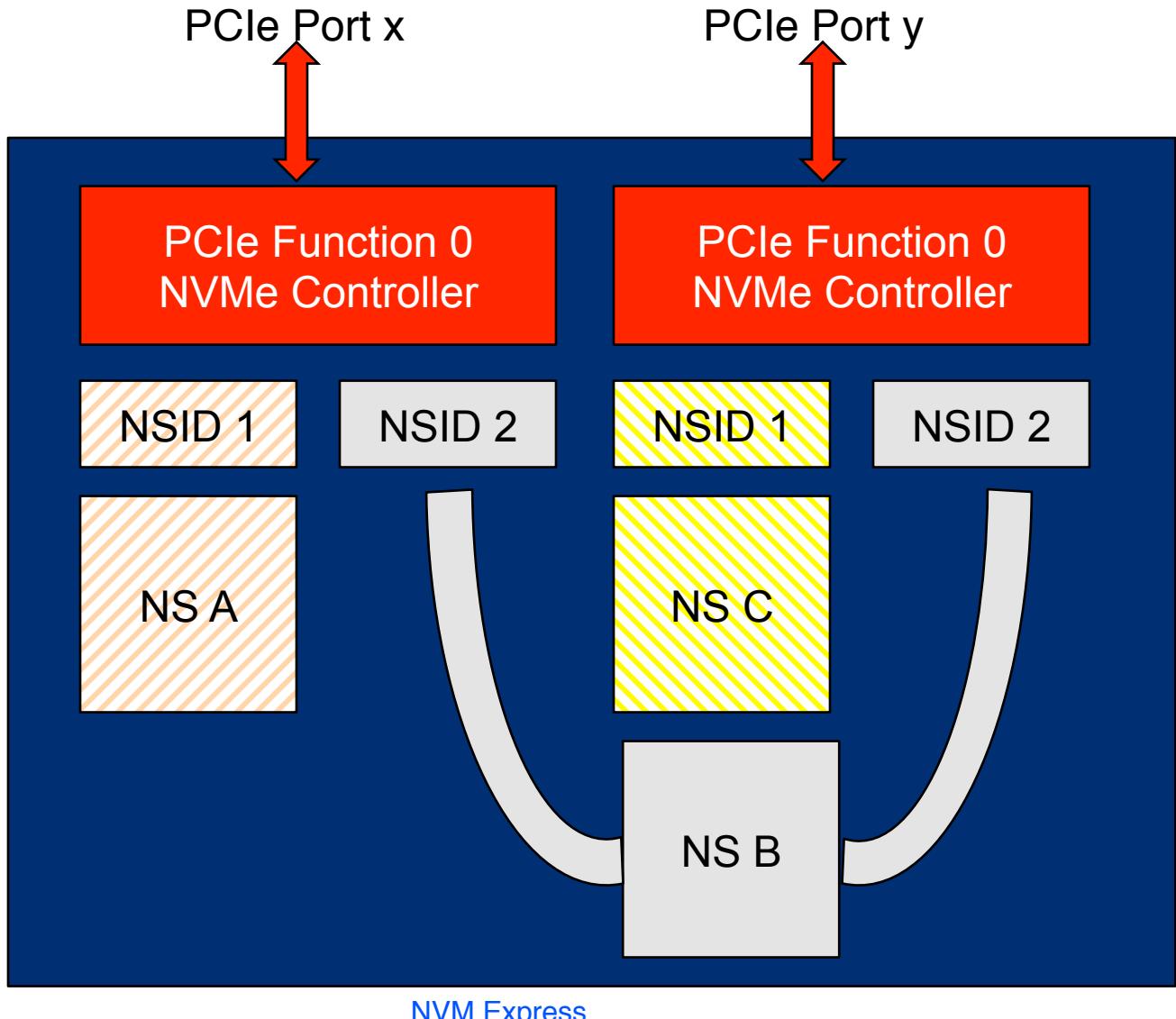
Two Controllers with Private and Shared Namespaces

Namespaces A and C
are private and not
shared.

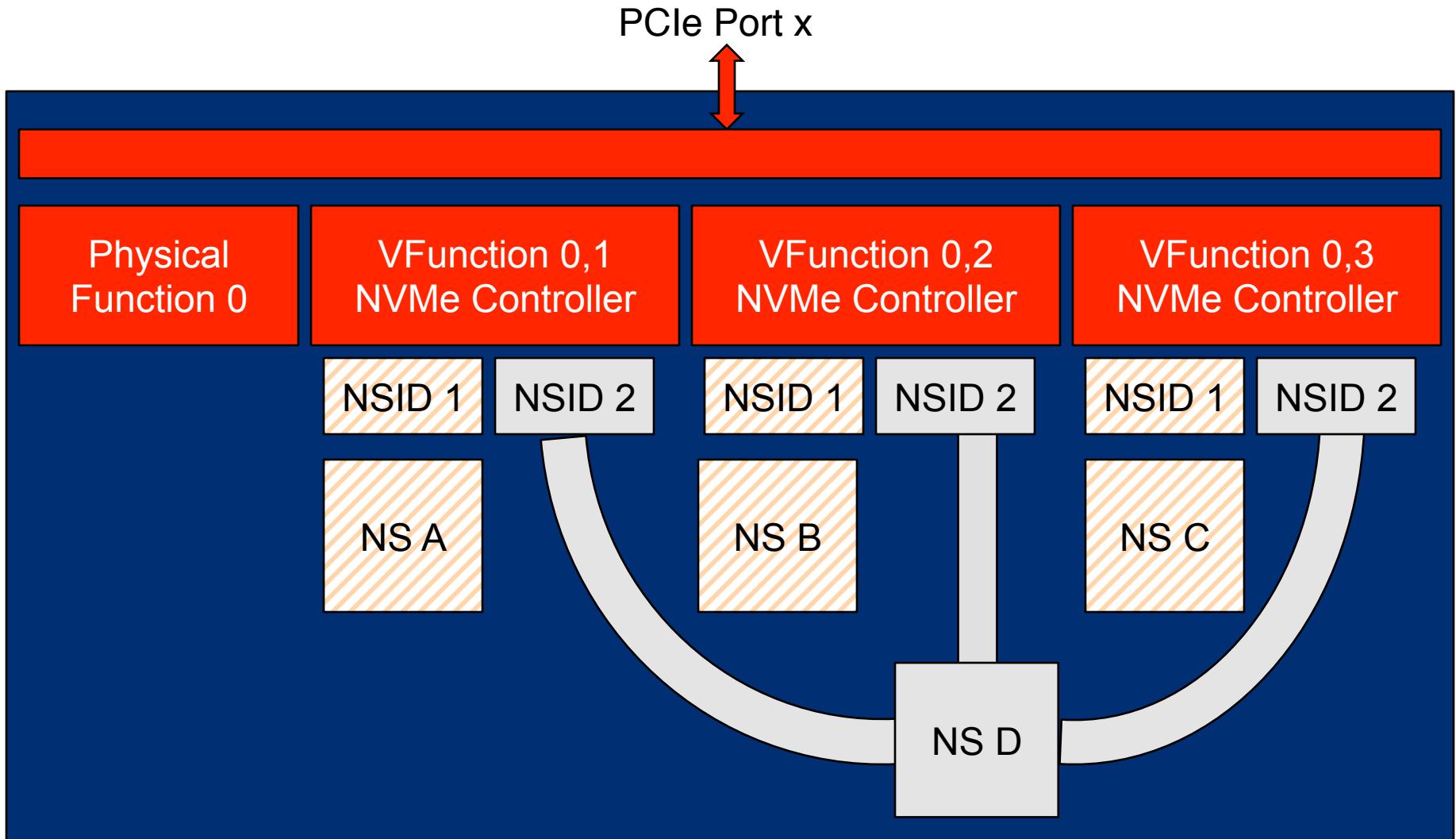
Namespace B is
shared and both
controllers must
reference it with the
same NSID.



Two Controllers with Two Ports



PCIe Device Supporting SR-IOV



MetaData

Dword	Bytes	31	Name	0
0	03:00		Command ID (CID)	P S
1	07:04		Namespace Identifier (NSID)	F
2	11:08		Reserved	
3	15:12			
4	19:16		Metadata Pointer (MPTR) – Address of physical buffer for metadata	
5	23:20			
6	27:24		PRP Entry 1 (PRP1)	
7	31:28			or SGL 1
8	35:32		PRP Entry 2 (PRP2)	
9	39:36			
10	43:40			
11	47:44			
12	51:48		Command specific fields	
13	55:52			
14	59:56			
15	63:60			



MetaData

What is MetaData?

Contextual information about a particular LBA of data.

Examples:

Creation or expiration date

Protection Information

Program or version of program that created the data

Information for search and fast lookup

LB – Logical Block
LBA – Logical Block Address

General Characteristics

May be stored in an extended LB with the data or

May be stored separate from associated LB.

Uses Metadata pointer for location of metadata region

For writes, metadata shall be written atomically with LB

End-to-End protection may be first 8 bytes or last 8 bytes of metadata



SGL And PRP



Data Buffer Pointers

Command Dwords 6-9

Dword	Bytes	31	Name	P S	Res	F	0
0	03:00		Command ID (CID)				OP Code
1	07:04		Namespace Identifier (NSID)				
2	11:08		Reserved				
3	15:12						
4	19:16		Metadata Pointer (MPTR) – Address of physical buffer for metadata				
5	23:20						
6	27:24		PRP Entry 1 (PRP1)				
7	31:28				or	SGL 1	
8	35:32		PRP Entry 2 (PRP2)				
9	39:36						
10	43:40		Command specific fields				
11	47:44						
12	51:48						
13	55:52						
14	59:56						
15	63:60						



Text Stuff

Data Buffer Pointer is a data structure that defines a data buffer

Problem:

Data is fragmented in memory

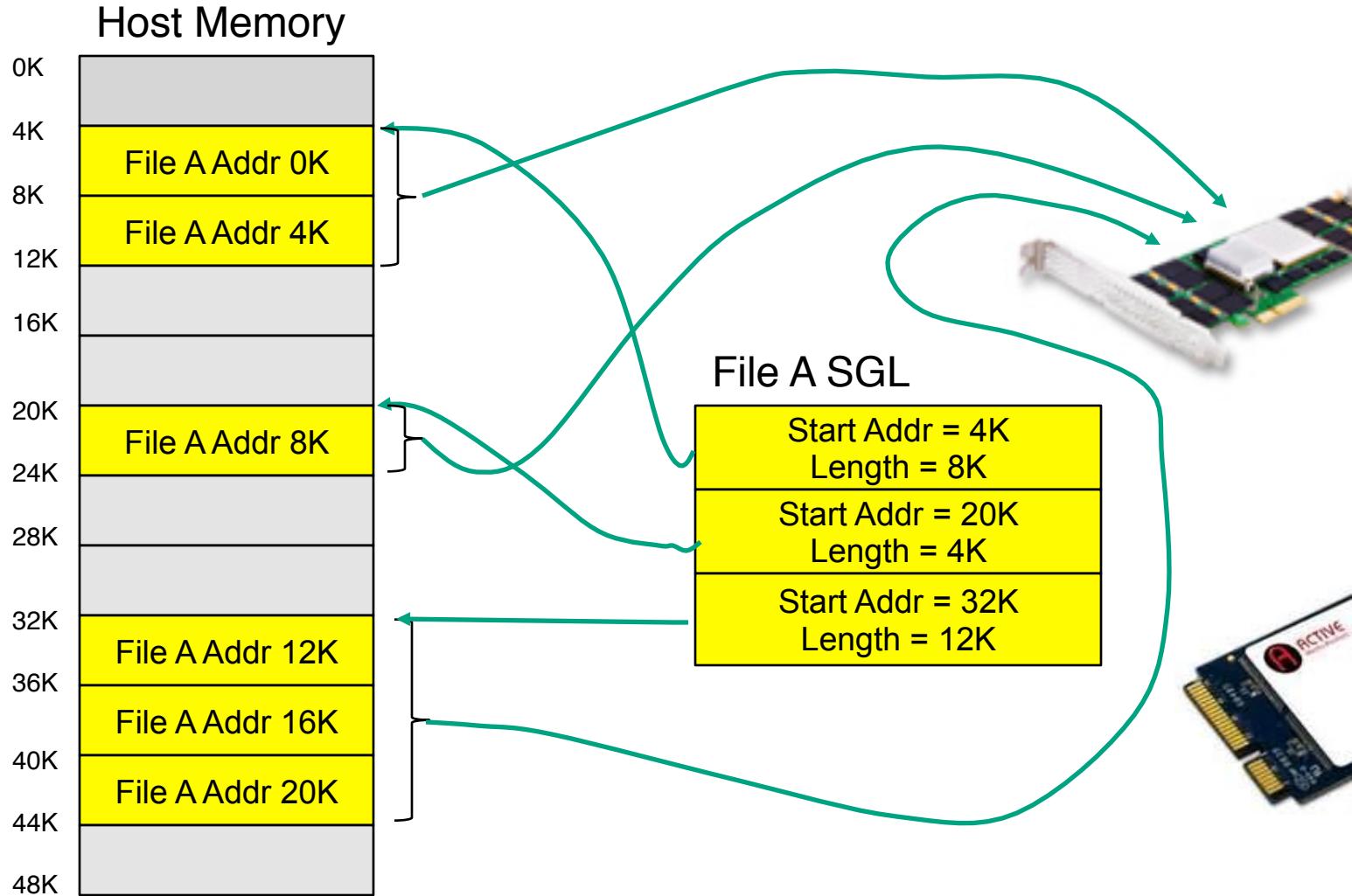
Data may be fragmented in storage

Solution:

Create a data structure consisting of pointers showing where the data buffers are



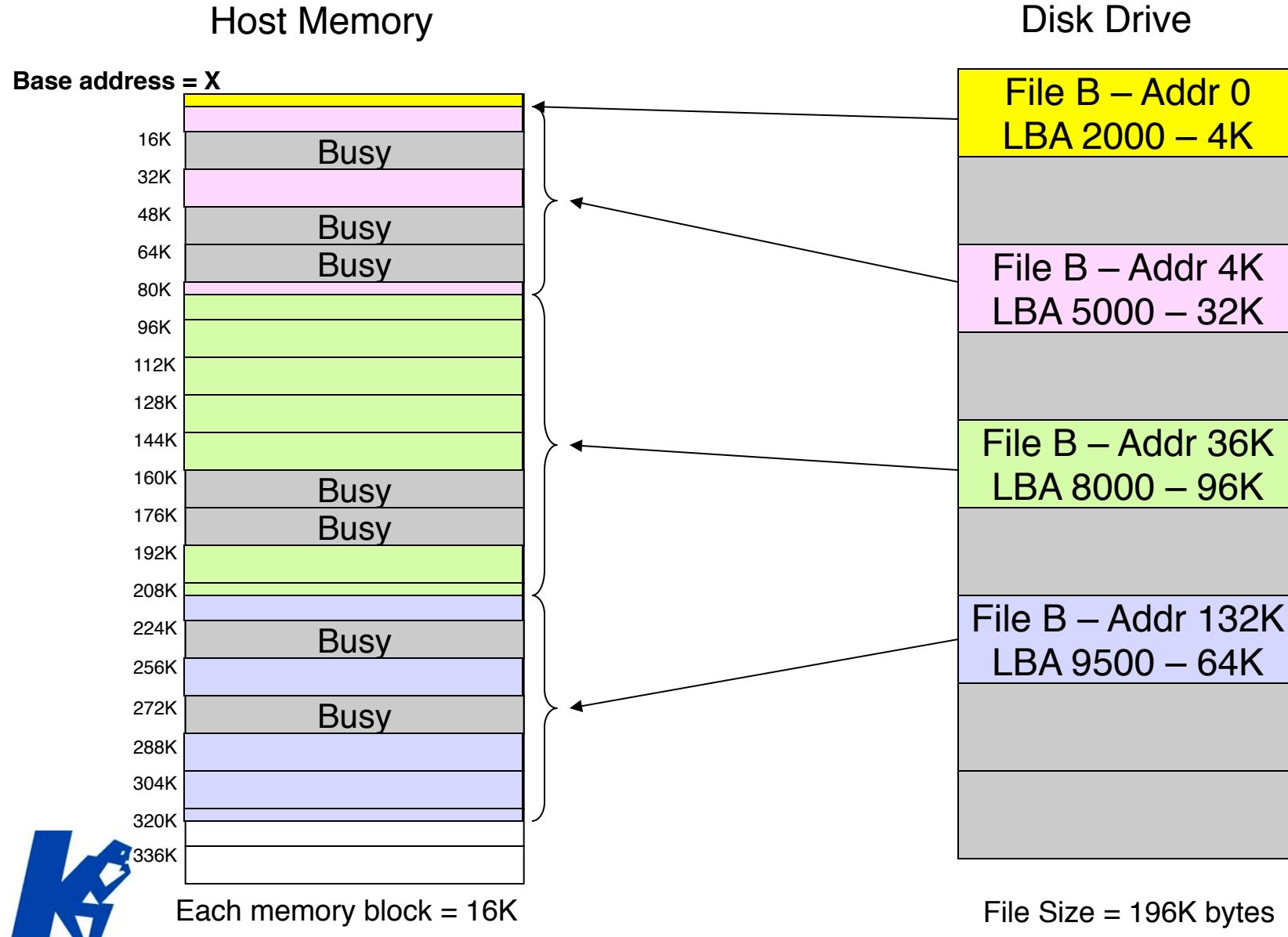
Generic SGL



Also called a Physical Region Descriptor Table (PRDT)

NVM Express

System Memory map to Device LBA map – Example



Mapping – Commands – Example

Please reference previous page

Read LBA 2000, 8 sectors

DBA = X, DBC = 4K

File is fragmented
on disk and
in memory

Read LBA 5000, 64 sectors

DBA = X + 4K, DBC = 12K
DBA = X + 32K, DBC = 16K
DBA = X + 80K, DBC = 4K

One SGL

Another SGL

Read LBA 8000, 192 sectors

DBA = X + 84K, DBC = 76K
DBA = X + 192K, DBC = 20K

One more SGL

Read LBA 9500, 128 sectors

DBA = X + 212K, DBC = 12K
DBA = X + 256K, DBC = 16K
DBA = X + 288K, DBC = 36K

Yet another SGL

DBA –
Data Buffer
Address

DBC –
Data Byte
Count

Assume 512 bytes
per logical sector

NVM Express

File B Scatter-Gather List in Memory

Host Memory

Base address = X



Each memory block = 16K

File B SGL

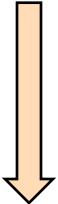
Start Addr = X + 0K	Length = 16K
Start Addr = X + 32K	Length = 16K
Start Addr = X + 80K	Length = 80K
Start Addr = X + 192K	Length = 32K
Start Addr = X + 156K	Length = 16K
Start Addr = X + 288K	Length = 36K

NVM Express

NVMe Defined SGL Terms

Scatter Gather List

A data structure in memory address space used to describe a data buffer.

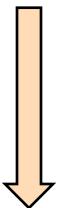


SGL Segment

Data structure of contiguous memory that contains:

16-byte data buffer descriptors, and/or
pointer to the next segment, if any

May define all of, part of, or none of the data buffer



SGL Descriptor

Data Block Descriptor – contains memory address and length for next data

Bit bucket descriptor – contains length of data to ignore

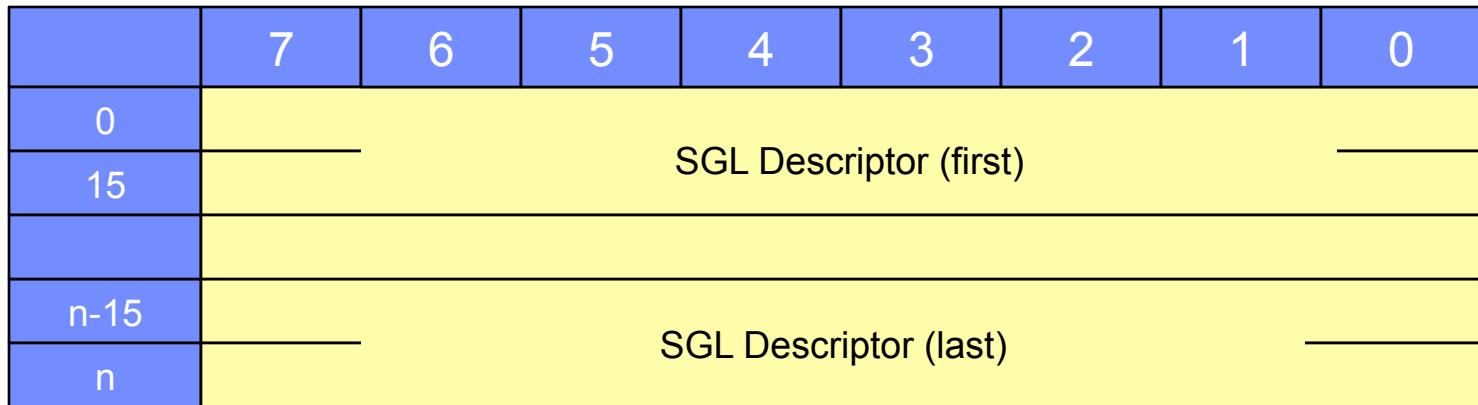
Segment descriptor – contains address of next segment

Last segment descriptor – contains address of last segment in this chain

Keyed Data Block Descriptor – contains a 32-bit key associated with the data block



SGL Segment



SGL Descriptors

	7	6	5	4	3	2	1	0
0								
14								
15								



Type Code	Descriptor	Desc. Size	Purpose
0h	Data Block	16 bytes	Define one contiguous area in memory for data-in or data-out
1h	Bit Bucket	16 bytes	Segment on source to ignore
2h	Standard Segment	16 bytes	Pointer to next segment
3h	Last Standard Segment	16 bytes	Pointer to last standard segment
4h	Key Data Block	16 bytes	Data block with 32-bit key
Fh			Vendor Specific



Data Block Descriptor

	7	6	5	4	3	2	1	0
0								
7								
8								
11								
12								
14								
15	SGL descriptor type (0h)							Sub Type

Points to a data block in memory

Sub Type for type 0

0h – Address field is a 64-bit memory byte address

1h – Not valid in PCIe implementations

For NVMe over Fabrics, Data is <Starting Address> from Command

A – Fh – NVMe Transport Specific



Bit Bucket Descriptor

	7	6	5	4	3	2	1	0
0								
7								Reserved
8								
11							Length in bytes	
12								
14							Reserved	
15	SGL descriptor type (1h)						SGL Sub Type	

Specifies a block of source data to skip over

Sub Type for type 1
A – Fh – NVMe Transport Specific

SGL Segment Descriptor

	7	6	5	4	3	2	1	0
0								Reserved
7					Starting Address			
8								
11					Length in bytes			
12								
14					Reserved			
15				SGL descriptor type (2h)				SGL Sub Type

Points to next segment

Sub Type for type 2

0h – Address field is a 64-bit memory byte address

1h – Not valid in PCIe implementations

For NVMe over Fabrics,

<Starting Address> is distance in bytes from end of command to SGL in capsule

A – Fh – NVMe Transport Specific

NVM Express

SGL Last Segment Descriptor

	7	6	5	4	3	2	1	0
0								Reserved
7			Starting Address					
8								
11				Length in bytes				
12								
14					Reserved			
15			SGL descriptor type (3h)			SGL Sub Type		

Points to last segment

Last Segment Descriptor can contain:

Data Descriptors or
Bit Bucket Descriptors

Sub Type for type 3

0h – Address field is a 64-bit memory byte address

1h – Not valid in PCIe implementations

For NVMe over Fabrics, <Starting Address> is
distance in bytes from end of command
to SGL in capsule

A – Fh – NVMe Transport Specific

Last Segment cannot contain a:

SGL Segment Descriptor or
SGL Last Segment Descriptor

Keyed SGL Data Block Descriptor

	7	6	5	4	3	2	1	0
0								
7								Starting Address
8								
10								Length in bytes
11								
14								Key
15				SGL descriptor type (4h)			SGL Sub Type	

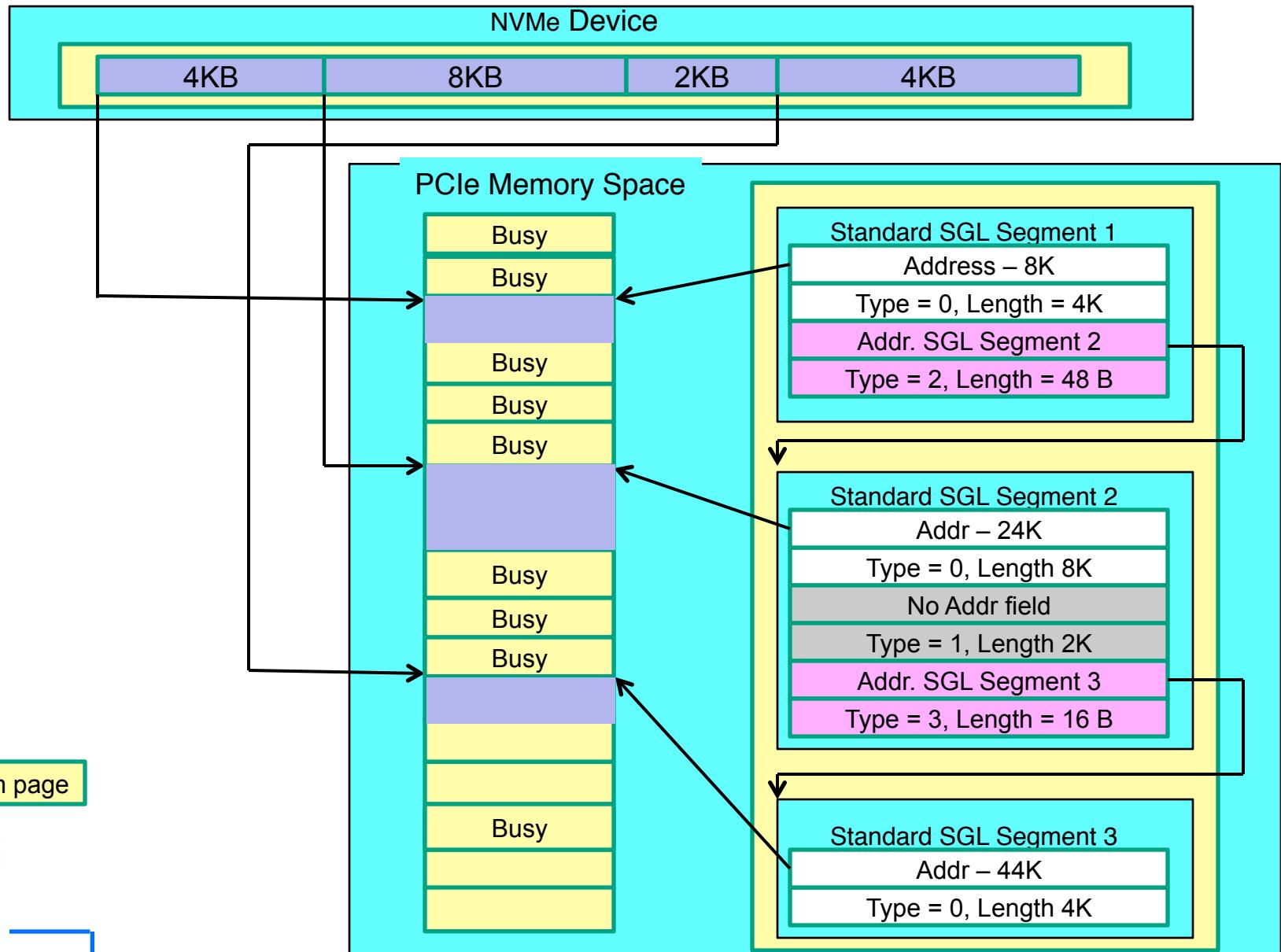
Similar to Data Block Descriptor
but specifies a 32 bit key associated with the data block

Sub Type for type 4

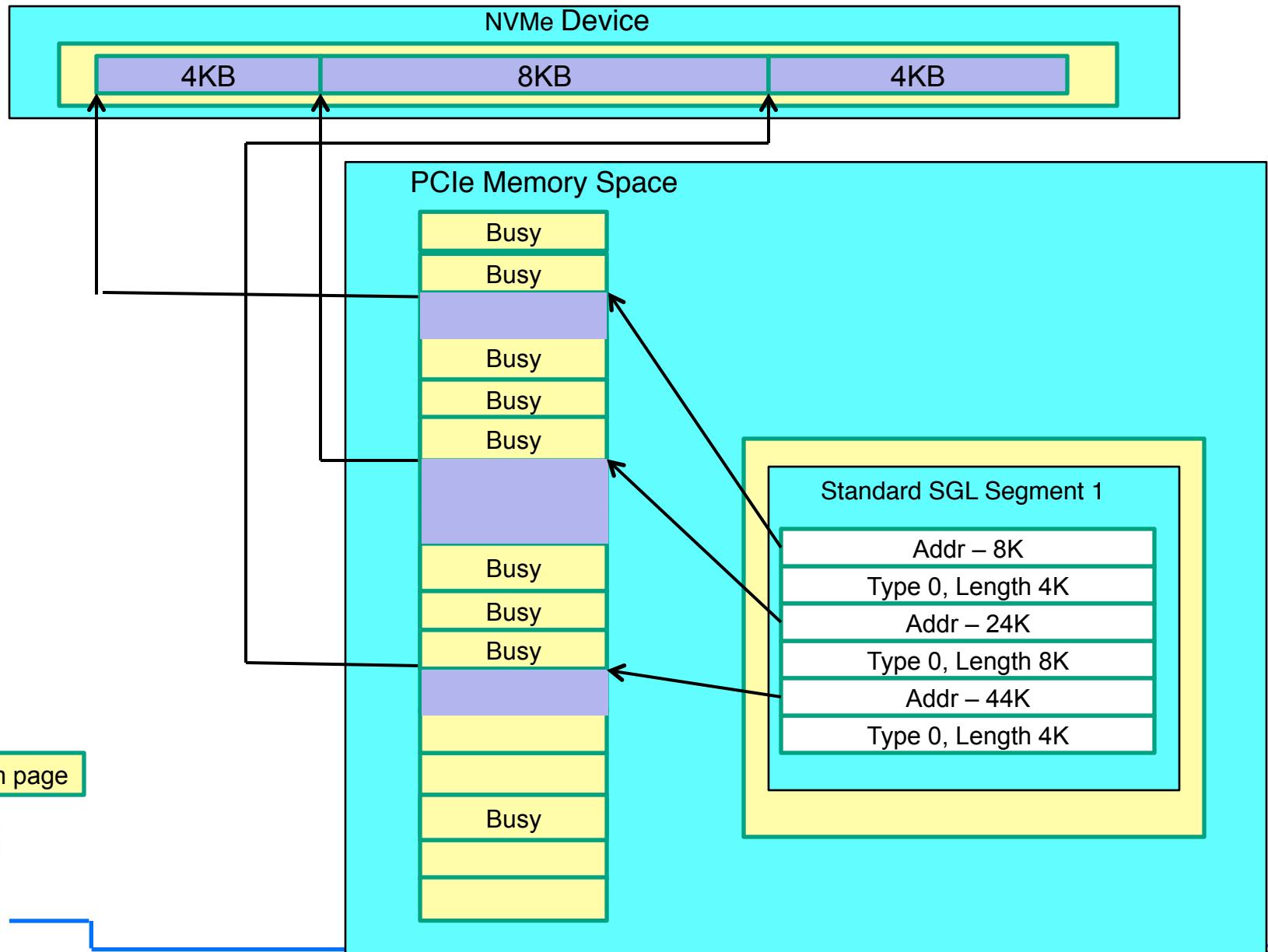
0h – Address field is a 64-bit memory byte address

A – Fh – NVMe Transport Specific

SGL Structure for Transfer “TO” Data Buffer



SGL Structure for Transfer “From” Data Buffer



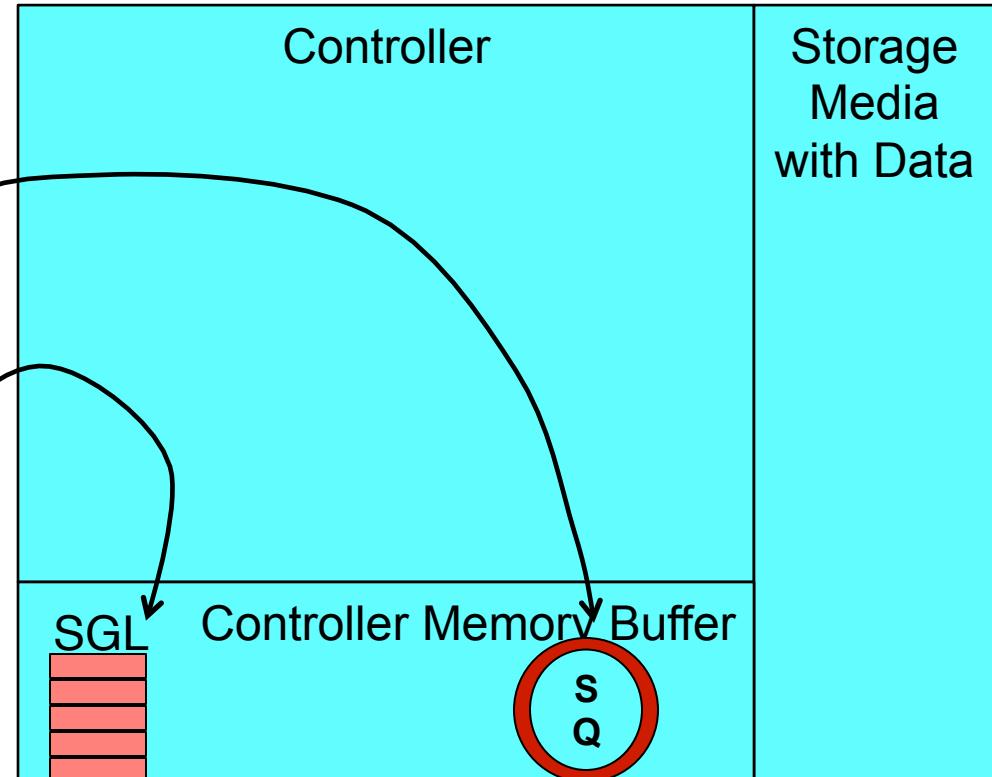
NVMe 1.2 Enhancement

SGL and SQ in CMB

Host Memory

X	Occupied
X+4k	Occupied
X+8k	
X+12k	Occupied
X+16k	Occupied
X+20k	Occupied
X+24k	
X+28k	
X+32k	Occupied
X+36k	Occupied
X+40k	Occupied
X+44k	
X+48k	
X+52k	
X+56k	

4K page



Host creates SGL and writes to CMB

Host creates command and writes command to SQ

Both SQ and SGL are in CMB

Host Memory is address X

Controller Memory Buffer is address Y

NVM Express

Physical Region Page

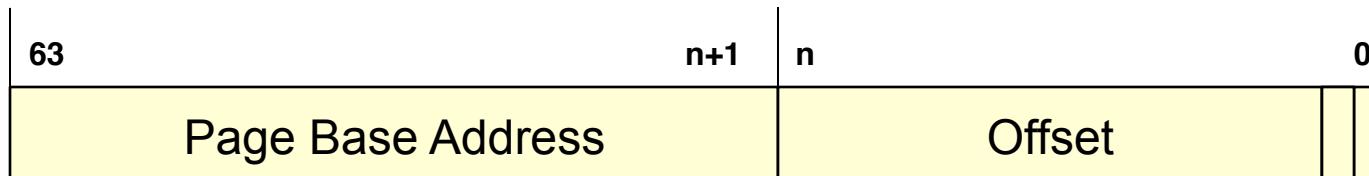
What is a Physical Region Page (PRP) Pointer?

64 bit data structure that describes a physical memory page

Similar to Scatter/Gather Lists except:

All PRP entries define a single memory page as defined in CC.MPS

All PRP entries are 8 bytes long, SGL descriptors are 16 bytes long



Page Base Address – lowest address for this page

Offset – number of bytes into this page. 0h on all except 1st entry

n – indicates page size, $2^{(n+1)}$ (n = 11 for 4096 byte pages)

If a PRP entry is the last one on a page, and
it does not complete the data transfer, then
that entry is a pointer to the next PRP list

NVM Express

PRP List

A set of PRP entries in a single page of contiguous memory.

Lists PRP entries that could not be described in the command

First PRP entry is in command

Second PRP entry in command may be:

- Last PRP entry for this command or

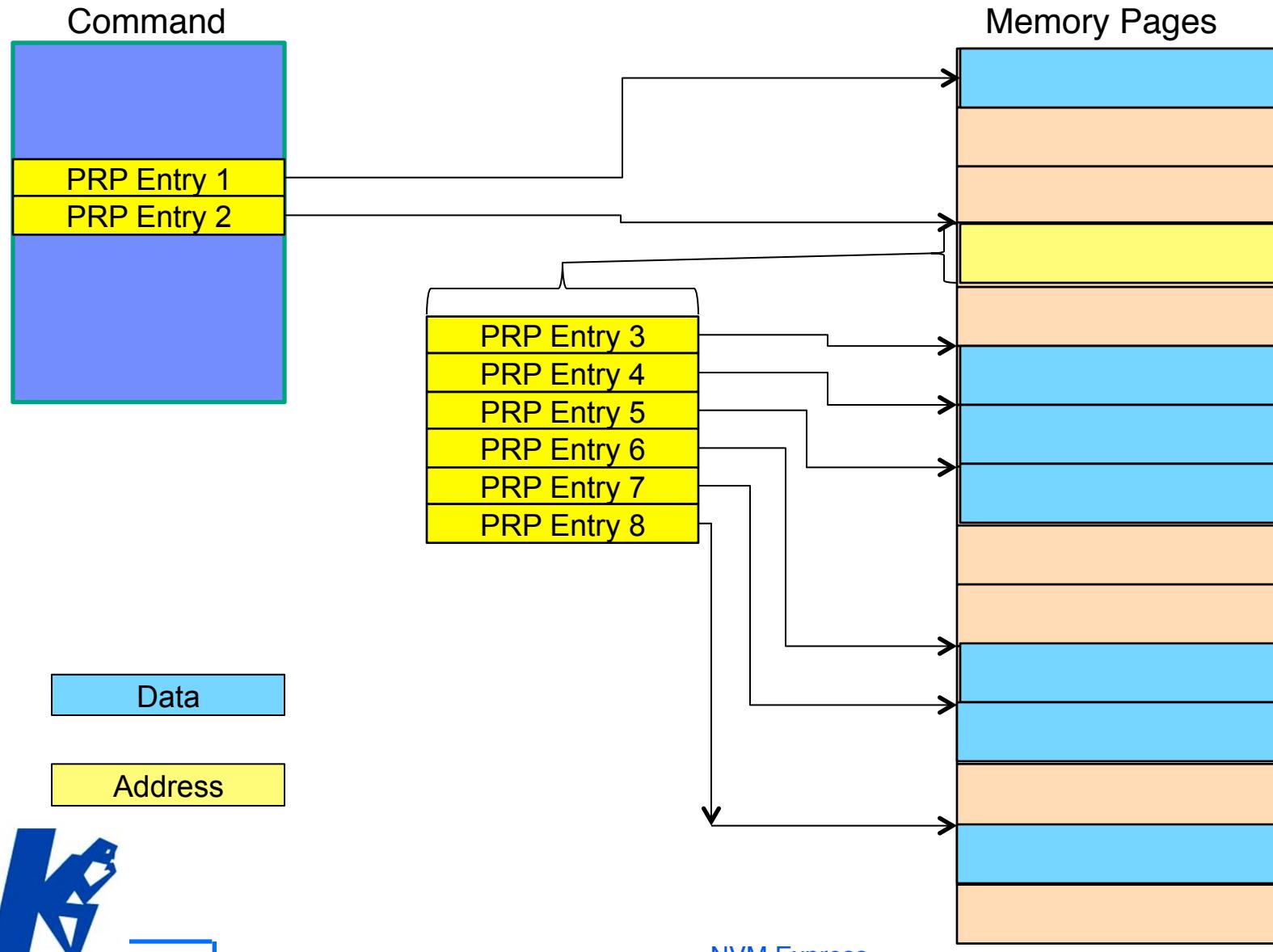
- Pointer to PRP List

PRP Lists may be chained

End of PRP list is implied by the command parameters and memory page size



PRP Example



PRP/SGL in Command

DWord	Bytes	Bits	Description	
6	27:24	63:00	PRP Entry 1 (PRP1)	LSB
7	31:28			MSB
8	35:32	127:64	PRP Entry 2 (PRP2)	LSB
9	39:36			MSB

Or

DWord	Bytes	Bits	Description	
6	27:24	127:00	SGL Entry 1	LSB
7	31:28			
8	35:32			
9	39:36			MSB

MetaData Pointer

PSDT	Description of Bytes 23:16
00	Contiguous physical buffer, Dword aligned
01	Contiguous physical buffer, Byte aligned
10	SGL segment w/1 descriptor, Qword aligned
11	Reserved

Bytes	31	Name	0
03:00	Command ID (CID)	P S	Res F OP Code
07:04	Namespace Identifier (NSID)		
11:08			
15:12		Reserved	
19:16	Metadata Pointer (MPTR) – Address of physical buffer for metadata		
23:20			
27:24	PRP Entry 1 (PRP1)		
31:28			
35:32	PRP Entry 2 (PRP2)		
39:36			
43:40			
47:44			
51:48			
55:52			
59:56			
63:60	Command specific fields		

Data Pointer

PSDT	Description of Bytes 39:24
00	Points to PRP Entry/Entries
01	Points to SGL segment
10	Points to SGL segment
11	Reserved



Status



Command Completion - Generic

Dword	Name
0	Command Specific
1	Reserved
2	SQ Identifier
3	Status Field P Command Identifier

SQ Identifier (SQID): Indicates the submission queue that the associated command used.

Command Identifier (CID): Indicates the identifier of the command being completed.

SQ Head Pointer (SQHD): Indicates the current head pointer for the submission queue that the associated command used.

Phase Tag (P): Identifies whether a Completion Queue entry is new.

All queue slots set to 0 before any use.

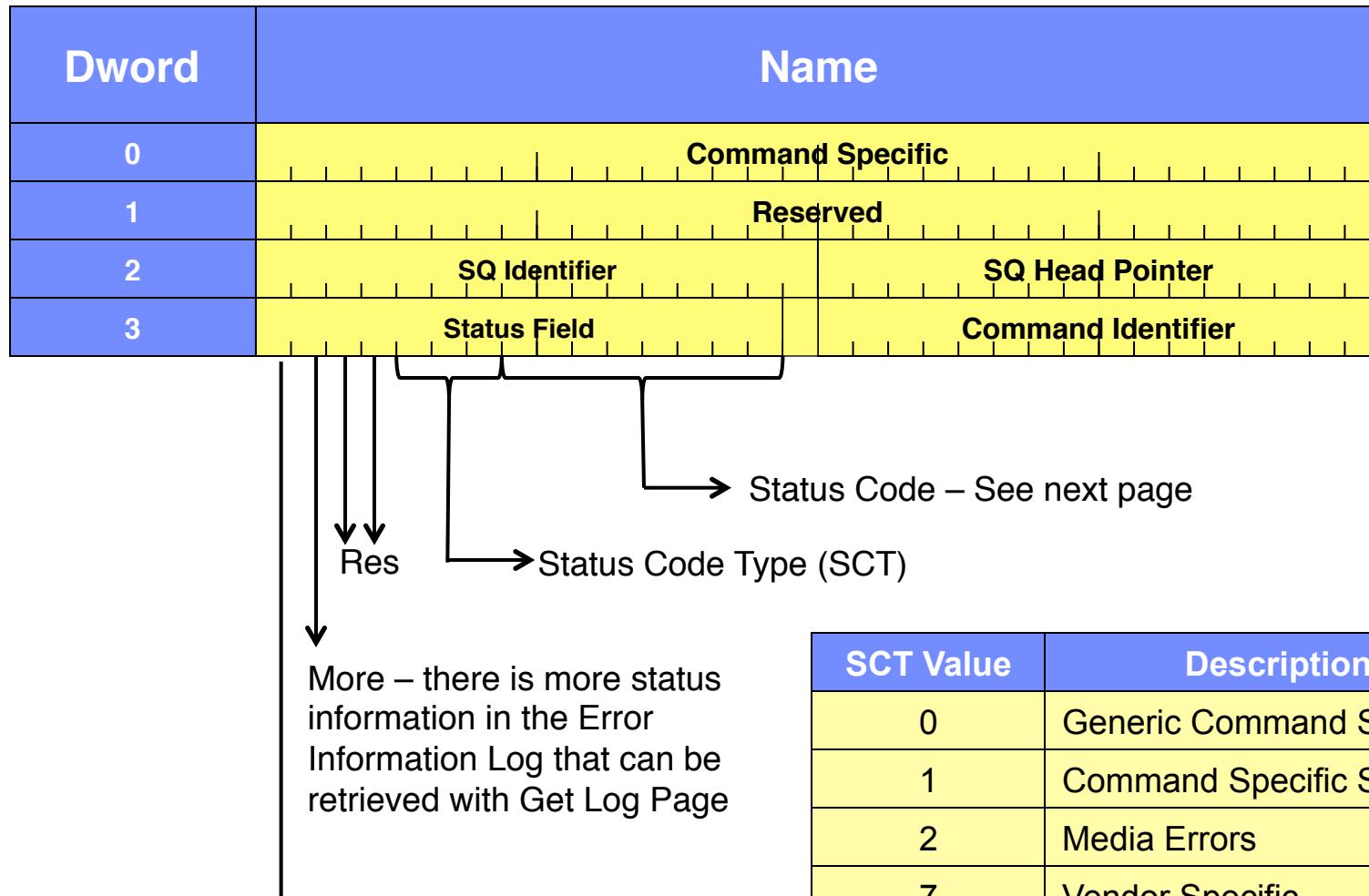
When a slot is written the bit is flipped.

First time through all slots, P bit is set to 1

Second time through all slots, P bit is set to 0



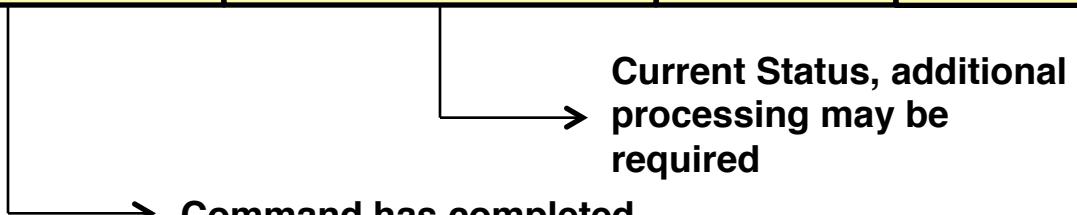
Command Completion – Status Field



SCT Value	Description
0	Generic Command Status
1	Command Specific Status
2	Media Errors
7	Vendor Specific

Command Completion – Status Codes

	Generic Command Status Values SCT=0h	Command Specific Status Values SCT = 1h	Media Specific Status Values SCT = 2h	Vendor Specific Status Values SCT = 7h
Applicable to Admin Command Set or across multiple command sets	00h	Successful Completion	Completion Queue Invalid	Reserved Vendor Defined
	01h	Invalid OpCode	Invalid Queue ID	
	02h	Invalid field in cmd	Invalid Queue size	
	03h	Command ID conflict	Abort cmd limit exceeded	
	04h	Data Transfer Error	Reserved	
	05h	Cmd Aborted – Power Loss Notification	Asynch Event Req limit exceeded	
	06h	Internal device error	Invalid Firmware slot	
	07h	Command Abort Request	Invalid Firmware Image	
	08h	Cmd aborted – SQ deletion	Invalid interrupt vector	
	09h	Cmd Aborted – Failed Fused Cmd	Invalid log page	



Interrupts



Methods of Signaling Interrupts

Pin-Based

INT A/B/C/D

MSI

Single MSI

MSI with a single vector enabled

Multiple Vector MSI

MSI, may use up to 32 interrupt vectors

Multiple-Message MSI

MSI, allows completions to be aggregated on a per vector basis.

MSI-X

Covered in the PCIe course



Interrupt Aggregation

Also called Interrupt Coalescing

NVMe host passes Interrupt Aggregation parameters to controller.
Controller algorithm is vendor specific

Values passed to controller are advisory only

Aggregation Threshold

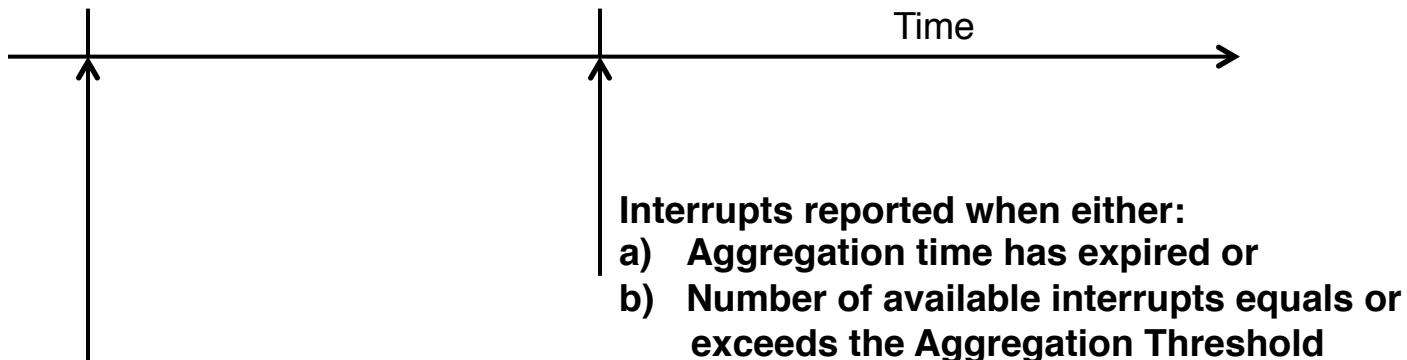
Host desired number of Completion Queue entries per interrupt vector before controller produces an interrupt

Aggregation Time

Host desired maximum delay that a controller may wait before an interrupt is signaled to the host



Interrupt Aggregation - Diagram



Covered in this Section

General Command format

Command Double Word 0

MetaData

Scatter/Gather Lists

SGL

PRP

Status

Interrupts



Notes



Section 5

Admin Commands

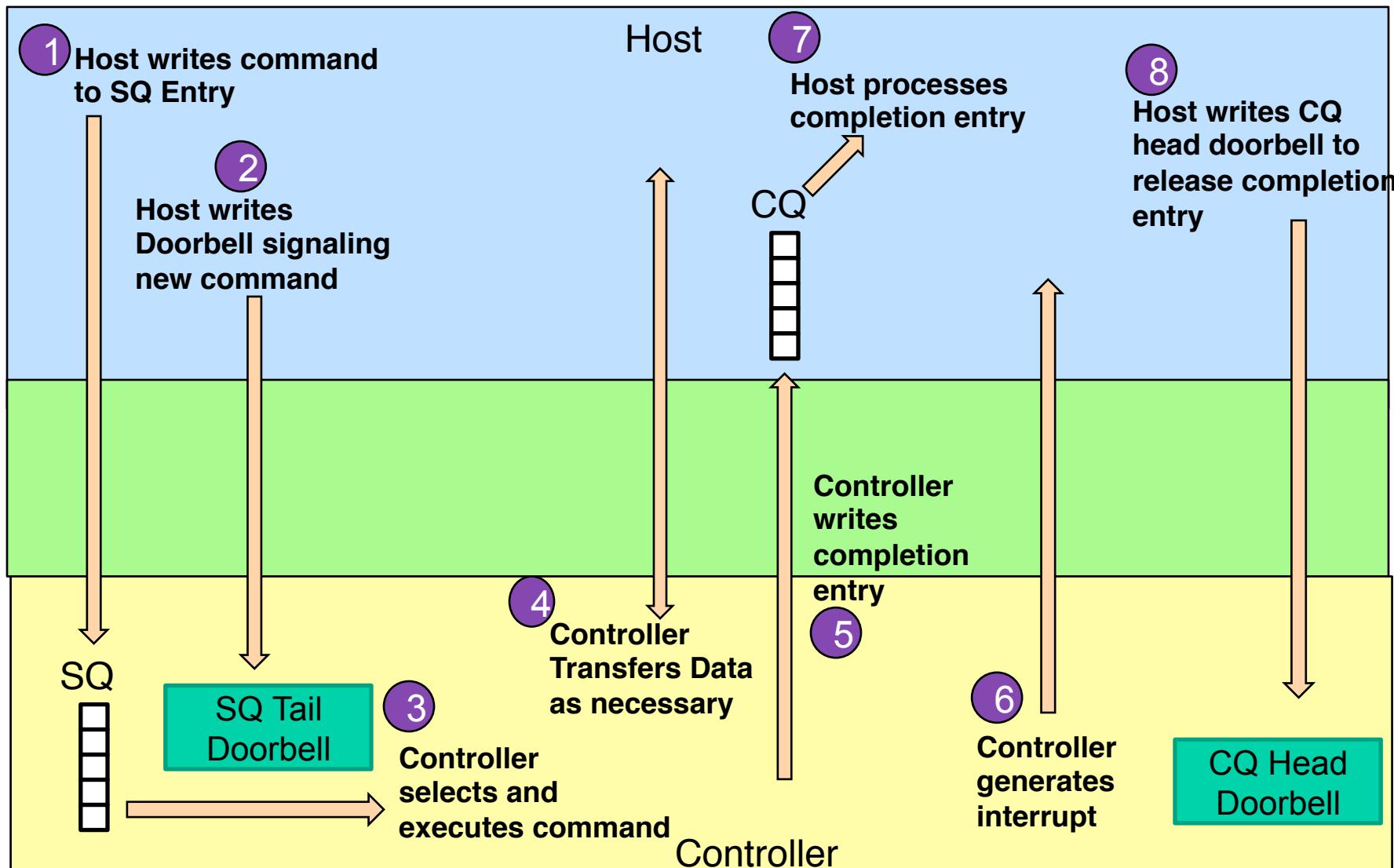


Covered in this Section

NVMe Admin Command Set



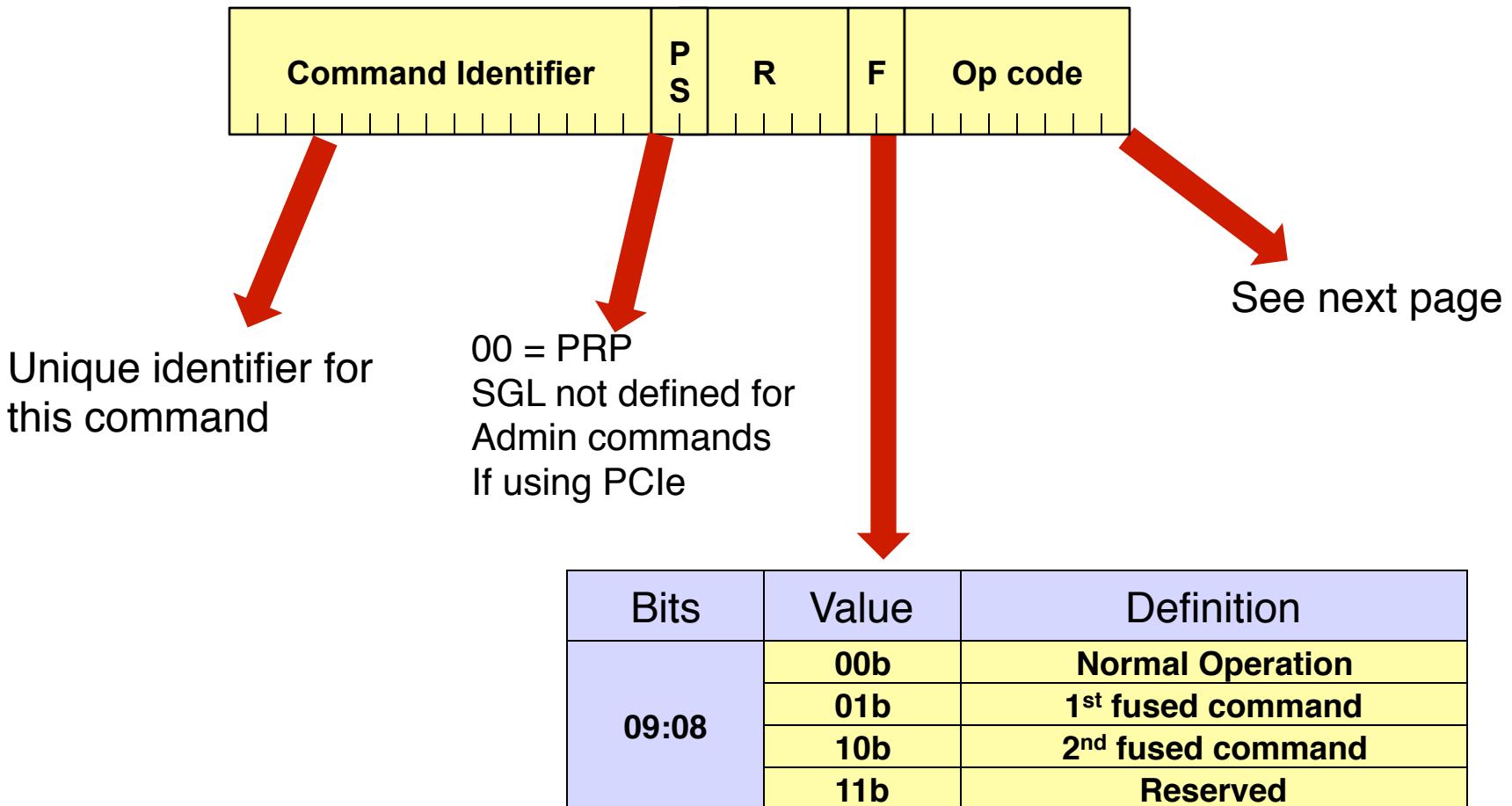
NVMe Command Processing (Picture – Informative)



ADMIN Command Format

Dword	Bytes	Name			
0	03:00	Command ID (CID)	P S	Res	F OP Code
1	07:04	Namespace Identifier (NSID)			
2	11:08	Reserved			
3	15:12				
4	19:16	Metadata Pointer (MPTR) – Address of physical buffer of metadata			
5	23:20				
6	27:24	PRP Entry 1 (PRP1)			
7	31:28				
8	35:32	PRP Entry 2 (PRP2)			
9	39:36				
10	43:40				
11	47:44				
12	51:48				
13	55:52	Command specific fields			
14	59:56				
15	63:60				

Command Dword 0 (Review)



A complex command is created by “fusing” together two simpler commands.

Op Code for Admin Commands

Op Code	Command
00h	Delete I/O Submission Queue
01h	Create I/O Submission Queue
02h	Get Log Page
04h	Delete I/O Completion Queue
05h	Create I/O Completion Queue
06h	Identify
08h	Abort
09h	Set Features
0Ah	Get Features
0Ch	Asynchronous Event Request
0Dh	Namespace Management

Op Code	Command
10h	Firmware Commit/Firmware Activate
11h	Firmware Image Download
14h	Device Self-Test
15h	Namespace attachment
18h	Keep Alive
19h	Directive Send
1Ah	Directive Receive
1Ch	Virtualization Management
1Dh	NVMe-MI Send
1Eh	NVMe-MI Receive
7Ch	Doorbell Buffer Config
7Fh	Fabrics Commands
84h	Sanitize
C0 – FFh	Vendor specific



Admin Commands are issued
to the Admin Submission Queue.

Op Code for Admin Commands

NVM Command Set Specific

OpCode	NVMe over PCIe
80h	Format NVM
81h	Security Send
82h	Security Receive

I/O Queue Management



Create I/O Completion Queue

Dword	Bytes	Name	
0	03:00	Common Fields	Op Code 05h
1-5	23:04	Common Fields	
6-7	31:24	PRP-1	
8-9	39:32	Reserved	
10	43:40	Queue Size	Queue Identifier
11	47:44	Interrupt Vector	Reserved I C
12	51:48	Reserved	
13	55:52	Reserved	
14	59:56	Reserved	
15	63:60	Reserved	

PRP-1 – Starting address of Queue or pointer to PRP List

Queue Size – Number of entries in the queue, 0 based

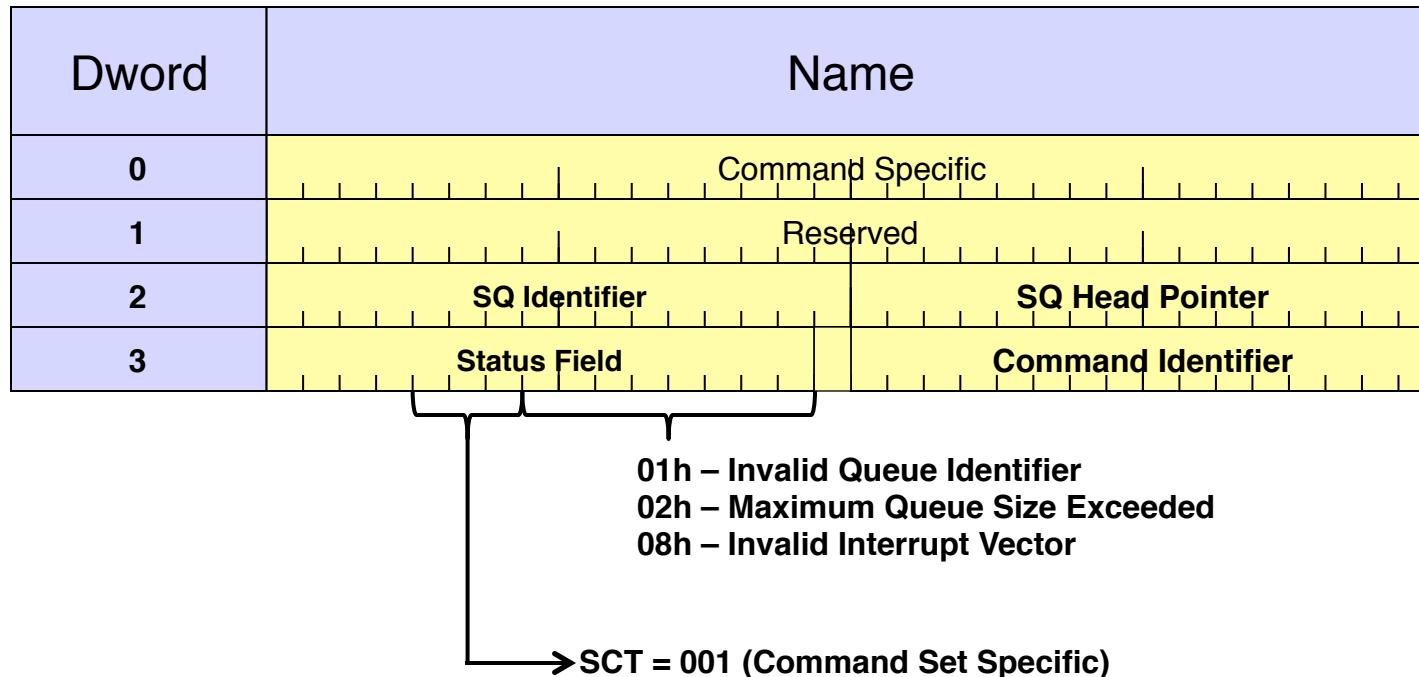
Interrupt Vector – MSI-X or multiple message MSI vector

I (IEN) – Interrupts enabled

C (PC) – Physically Contiguous – Completion queue is Physically Contiguous and there is one PRP Entry

NVM Express

Create I/O Completion Queue Completion



Successful completion is indicated by SCT = 000b and Status = 00h

Create I/O Submission Queue

Dword	Bytes	Name		
0	03:00	Common Fields	Op Code 01h	
1-5	23:04	Common Fields		
6-7	31:24	PRP-1		
8-9	39:32	Reserved		
10	43:40	Queue Size	Queue Identifier	
11	47:44	Completion Queue Identifier	Reserved	P C
12	51:48	Reserved		
13	55:52	Reserved		
14	59:56	Reserved		
15	63:60	Reserved		

Queue Size – Number of entries in the queue, 0 based

P (QPRIO) – Queue Priority

00b – Urgent

01b – High

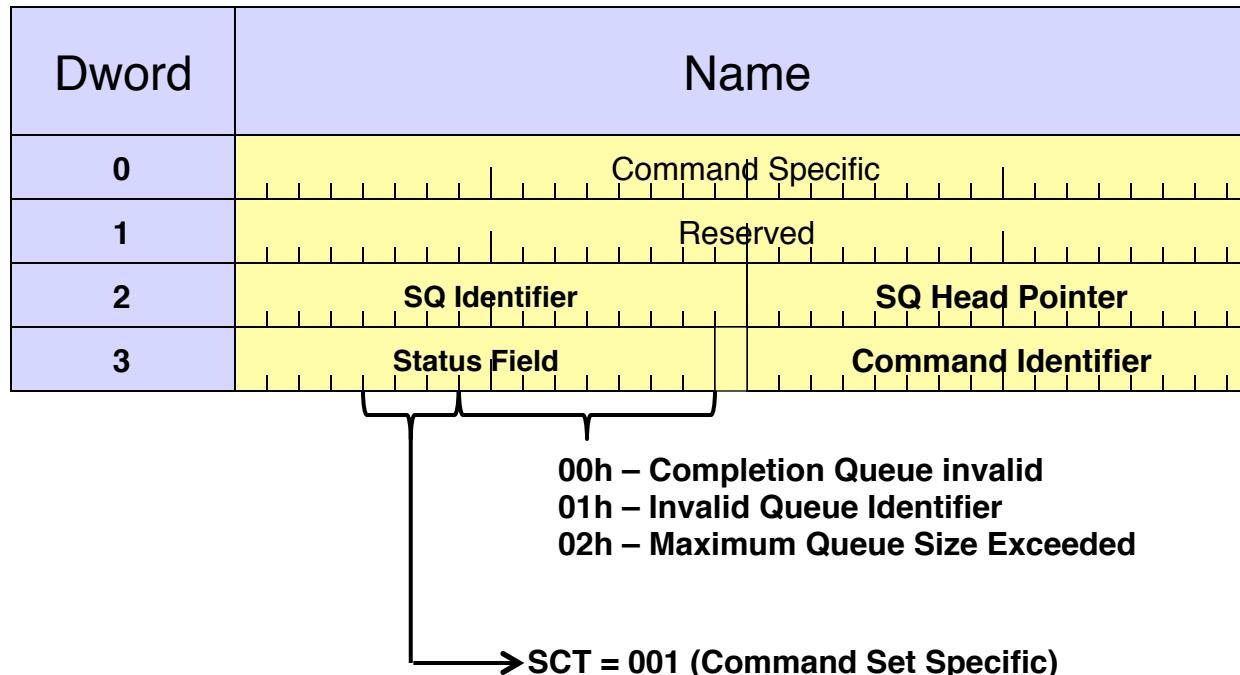
10b – Medium

11b – Low



C (PC) – Physically Contiguous – Submission queue is PC and there is one PRP Entry
NVM Express

Create I/O Submission Queue Completion

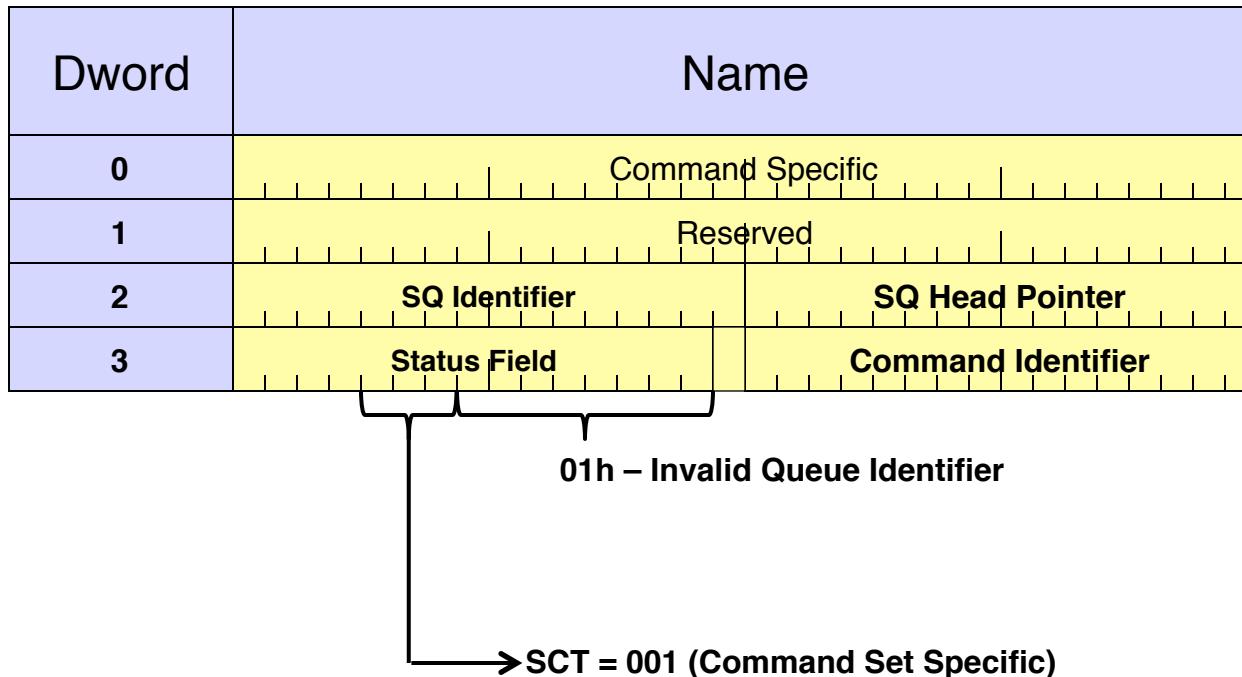


Successful completion is indicated by SCT = 000b and Status = 00h

Delete I/O Submission Queue

Dword	Bytes	Name	
0	03:00	Common Fields	Op Code 00h
1-9	39:04	Common Fields	
10	43:40	Reserved	Queue Identifier
11	47:44	Reserved	
12	51:48	Reserved	
13	55:52	Reserved	
14	59:56	Reserved	
15	63:60	Reserved	

Delete I/O Submission Queue Completion

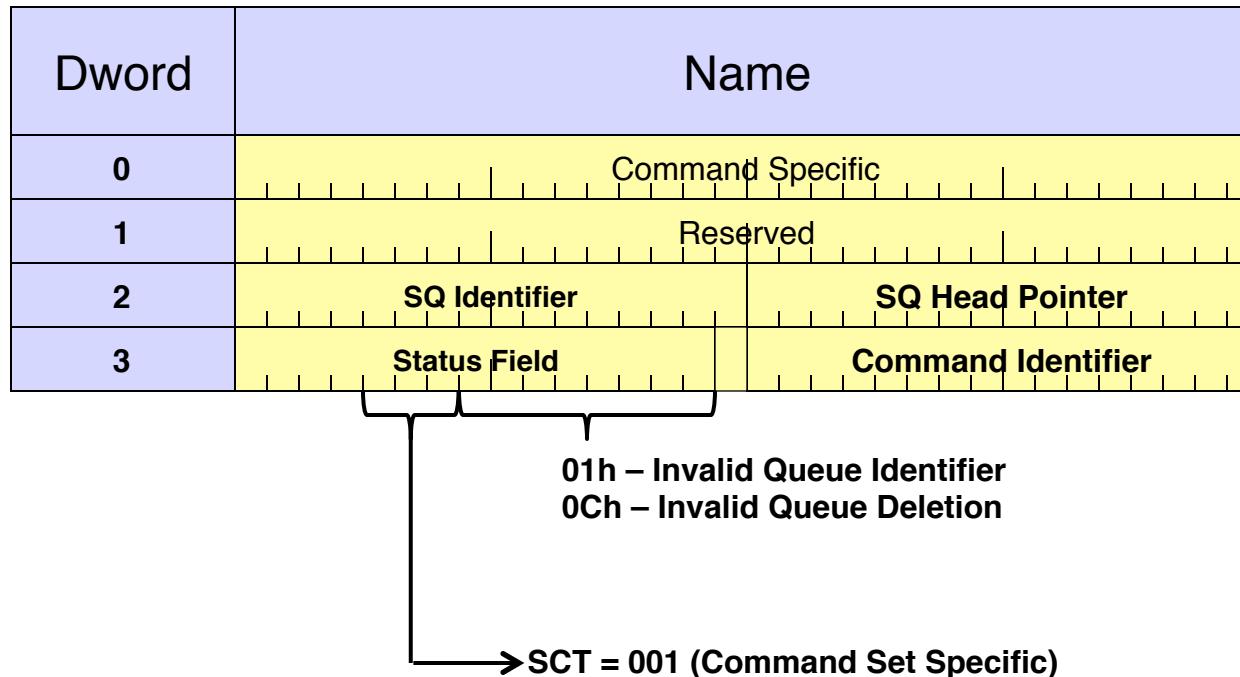


Successful completion is indicated by SCT = 000b and Status = 00h

Delete I/O Completion Queue

Dword	Bytes	Name	
0	03:00	Common Fields	Op Code 04h
1-9	39:04	Common Fields	
10	43:40	Reserved	Queue Identifier
11	47:44	Reserved	
12	51:48	Reserved	
13	55:52	Reserved	
14	59:56	Reserved	
15	63:60	Reserved	

Delete I/O Completion Queue Completion



Successful completion is indicated by SCT = 000b and Status = 00h

NVMe

Identify Command



Identify Command Format

Dword	Bytes	Name			
0	03:00	Command ID (CID)	Res	Fuse	OP Code = 06h
1	07:04	Namespace Identifier (NSID)			
2	11:08	Reserved			
3	15:12	Reserved			
4	19:16	Metadata Pointer (MPTR) – Reserved			
5	23:20	PRP Entry 1 (PRP1) – 1 st Physical Region Page			
6	27:24	Buffer where input data is returned			
7	31:28	PRP Entry 2 (PRP2) – 2 nd Physical Region Page			
8	35:32	Buffer where input data is returned			
9	39:36	Controller ID		Reserved	CNS
10	43:40	Command specific fields			
11	47:44				
15	63:60				

Returns 4096 bytes of capabilities and status
 for the controller or namespace or a list of namespaces
 to the location(s) in PRP Entry 1 and PRP Entry 2

Controller or Namespace Structure (CNS)

Value	Description
00h	Return Identify data structure for Namespace in dword 1 <i>if attached to this controller</i>
01h	Return Identify data structure for controller
02h	Return a list of <i>active</i> namespace ID greater than namespace in dword 1 Up to 1024 namespaces Zero filled
03h	Return Namespace Identifier type (EUI64, NGUID, UUID)
10h	List of up to 1024 Namespace ID > NSID in CDW1.NSID
11h	Data structure for Namespace identified in CDW1.NSID
12h	List of up to 2047 controller IDs identified in CWD10.CNTID <i>and attached to CDW1.NSID</i>
13h	List of up to 2047 controller ID containing the controller in CDW10.CNTID but may or may not be attached to namespaces.
14h	Primary Controller capabilities structure
15h	List of up to 127 Secondary Controllers associated with Primary Controller that issued this command



Example Identify and Initialization

Step	CNS	Action
1	01h	Return information about the controller, including number of Name Spaces
2	02h	Return list of active Name Spaces
3	00h	Return configuration of NS identified in dWord 1, if attached to this controller
4	00h	Return configuration of next NS identified in dWord 1, if attached to this controller
.	.	.
.	.	.
.	.	.

Identify Namespace Data Structure (part 1)

Bytes	Description
7:00	Namespace Size in Logical Blocks
15:8	Namespace Capacity for thin provisioning
23:16	Namespace Utilization (LB currently allocated)
24	Namespace Features Bit 0 – Supports thin provisioning Bit 1 – Host should use NAWUN, NAWUPF, and NACWU for namespace Bit 2 – Controller supports Deallocated Logical Block error for this NS
25	Number of LBA Formats
26	Formatted LBA Size (data + metadata) Bits 3:0 – Indicates which supported format this is Bit 4 – 0 = Metadata is transfer as a separate buffer of data 1 = Metadata is transferred at the end of the data LBA
27	Metadata Capabilities Bit 0 – Namespace supports transferring metadata as part of an extended LBA Bit 1 – Namespace support transferring metadata as part of a separate buffer
28	End to end data protection capabilities Bit 0 – Namespace supports Protection Type 1 Bit 1 – Namespace supports Protection Type 2 Bit 2 – Namespace supports Protection Type 3 Bit 3 – Namespace supports PI transferred as 1 st 8 bytes of metadata Bit 4 – Namespace supports PI transferred as last 8 bytes of metadata



Identify Namespace Data Structure (part 2)

Bytes	Description
29	End to End Data Protection Type
30	Namespace I/O and Namespace sharing Bit 0 – 0 = Namespace can only be accessed by controller that returned this NS data structure 1 = Namespace can be accessed by 2 or more controllers
31	Reservation Capabilities Bit 0 – Persist Through Power Loss Bits 1 – 6 indicate which reservation type NS supports Bit 1 – Write Exclusive Bit 2 – Exclusive Access Bit 3 – Write Exclusive – Registrants Only Bit 4 – Exclusive Access – Registrants Only Bit 5 – Write Exclusive – All Registrants Bit 6 – Exclusive Access – All Registrants
32	Format Progress Indicator
33	Reserved



Reservations will be defined in NVMe Command Section

NVM Express

Section 5: Admin Commands

NFG

5 - 22

Identify Namespace Data Structure (part 3)

Bytes	Description
35:34	Namespace Atomic Write Unit Normal
37:36	Namespace Atomic Write Unit Power Fail
39:38	Namespace Atomic Compare and Write Unit
41:40	Namespace Atomic Boundary Size Normal
43:42	Namespace Atomic Boundary Offset
45:44	Namespace Atomic Boundary Size Power Fail
47:46	Reserved
63:48	NVM Capacity allocated to this namespace in bytes
103:64	Reserved
119:104	Namespace Globally Unique Identifier (NGUID128)
127:120	IEEE Extended Unique Identifier (EUI64)
z: y	LBA Format x supported
383:192	Reserved
4095:384	Vendor Specific

Formula:
For $x = 0 \rightarrow 15$
 $y = 128 + x * 4$
 $z = y + 4$
Where $x = \text{LBA format}$

LBA Format Data Structure

Bits	Description
15:00	Metadata Size (bytes)
23:16	Logical Block Data Size (2^{**n})
25:24	Relative Performance 00b – Best performance 01b – Better performance 10b – Good performance 11b – Degraded performance
31:26	Reserved

Identify Controller Data Structure

Bytes	Description
255:00	Controller Capabilities and Feature
511:256	Admin Command Set Attributes and Optional Controller Capabilities
2047:512	NVM Command Set Attributes
3071:2048	Power State Descriptors
4095:3072	Vendor Specific

Power State Transitions

Types of Power State Transition:

Static

- Host set it and forget it

- Host uses Set Features to indicate new Power State

Dynamic

- Host adjusts to varying workload requirements

- See next page for description

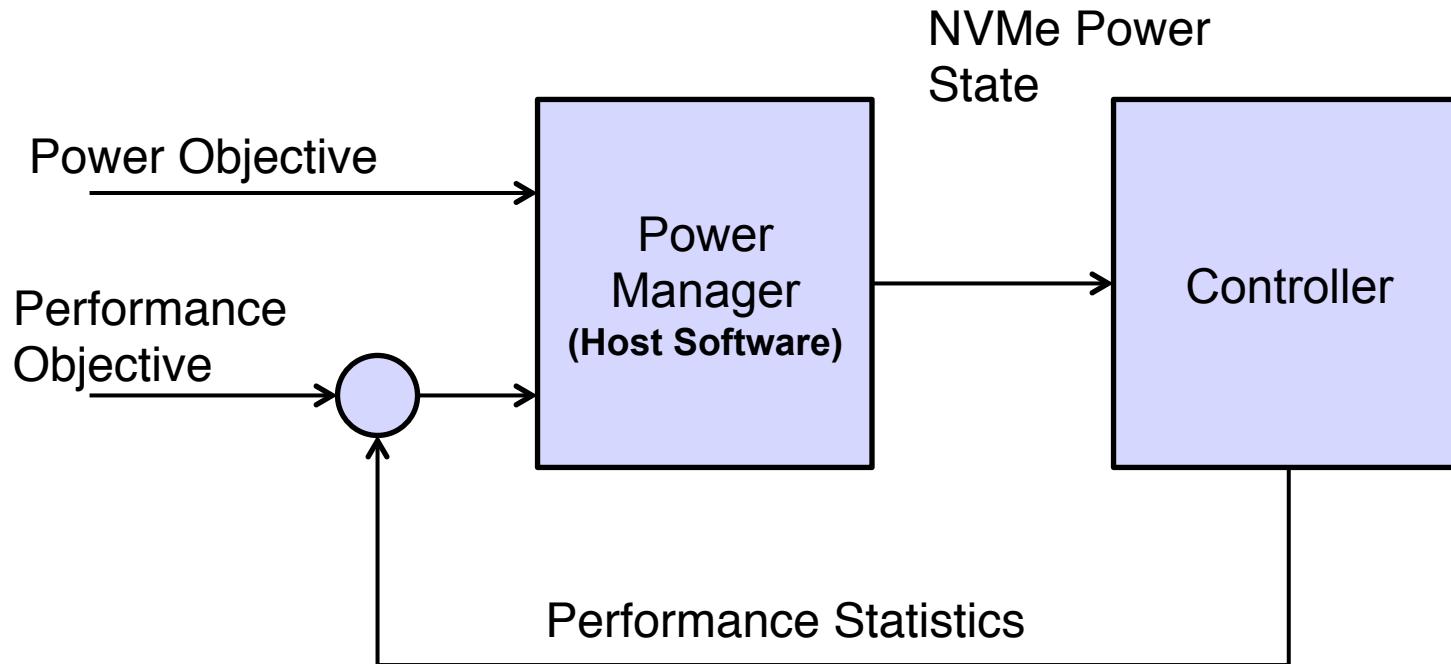
Autonomous

- Device/Controller transitions as necessary

- Host set idle time and new state for transition



Dynamic Power State Transitions



Power State Information

Power states define power consumption, entry and exit time, and relative performance of a device. Entered by manufacturer

Power states are numbered from 0 to max supported by device.

Max defined by specification is 32.

Power state 0 uses greatest amount of power;
each successively higher number uses less power



Power State Descriptor Table - Example

State	Max Power (W)	Entry Latency (μs)	Exit Latency (μs)	Relative Read Throughput	Relative Read Latency	Relative Write Throughput	Relative Write Latency
0	25	5	5	0	0	0	0
1	18	5	7	0	0	1	0
2	18	5	8	1	0	0	0
3	10	20	15	2	0	2	0
4	2	20	30	1	1	3	0
5	90mw	20	50				
6	5mw	20	5000		Non-Operational		



Power State Descriptor Data Structure

Bits	Description
15:00	Maximum Power
23:16	Reserved
24	Max Power Scale 0 – 0.01 Watts 1 – 0.0001 Watts
25	Non-operational State 0 – Controller processes cmds in this state 1 – Controller does not process cmds in this state
31:26	Reserved
63:32	Maximum Entry Latency in μsec
95:64	Maximum Exit Latency in μsec
100:96	Relative Read Throughput
103:101	Reserved
108:104	Relative Read Latency
111:109	Reserved
116:112	Relative Write Throughput
119:117	Reserved
124:120	Relative Write Latency

Relative to other power states.
Lower value means better performance.

Power State Descriptor Data Structure (concluded)

Bits	Description
127:125	Reserved
143:128	Idle Power (typical over 30 seconds)
149:144	Reserved
151:150	Idle Power Scale 00b – Not reported 01b – 0.0001W 10b – 0.01W
159:152	Reserved
175:160	Active Power (largest average over 10 seconds)
178:176	Active Power Workload
181:179	Reserved
183:182	Active Power Scale 00b – Not reported 01b – 0.0001W 10b – 0.01W
255:184	Reserved

Identify Command Trace

Teledyne LeCroy PETracer(TM) - PCI Express Protocol Analyzer - [C:\Users\Public\Documents\LeCroy\PETracer\Sample Files\NVMe_Z3DriveEmulation.pex]

File Setup Record Generate Report Search View Tools Window Help

Trace View

NVM 6	H	QID 0x0000	SQyTDBL	Admin SQT QID = 0 0x0002	Time Delta 213.996 us	Time Stamp 0038 . 868 079 906 s									
NVM 7	H	QID 0x0000	CID 0x0001	Address 00000002:2E062040	ASQ Identify	OPC PRP	PSDT Normal operation	FUSE	CID 0x0001	NSID 0x00000001	MPTR MPTR Low 0x0000000000	MPTR Hi			
					PRP1 0x2E071000	PRP1 Low 0x00000002	PRP2 0x00000000	PRP2 Low 0x00000000	PRP2 Hi Namespace	CNS	Time Delta 1.122 ms	Time Stamp 0038 . 868 293 902 s			
NVM 8	D	QID 0x0000	CID 0x0001	Identify NS	NSIZE 0x000000000000000080000	NCAP 0x00000000000080000	NUSE 0x00000000000000000000000000000000	NFEAT 0x00	NLBAF 0x00	FLBAS 0x00	MC 0x00				
		DPC 0x00	DPS Disabled	NMIC 00000000	RESCAP 00000000	EUI64 00:00:00:00:00:00:00:00	LBA0 0x0000	MS 0x09	LBADS Best	RP VS 1024 dwords	Data 782.616 us				
		Time Stamp 0038 . 869 415 998 s													
NVM 9	D	QID 0x0000	CID 0x0001	Address 00000002:2E064010	ACQ	Command Specific 0x00000000	SQHD 0x0002	SQID 0x0000	CID 0x0001	P 1	ST 0x00	SC 0x0	SCT 0	M 0	DNR 0
		Time Delta 23.064 us Time Stamp 0038 . 870 198 614 s													
Link Tra 77	R← x1	2.5 927	TLP Mem	MWr(32) 010:00000	Length 1	RequesterID 006:00:0	Tag 0	Address FEE3F00C	1st BE 1111	Last BE 0000	Data 1 dword	VC ID 0			
		Explicit ACK Packet #155	Metrics	# Packets 2	Time Delta 7.828 us	Time Stamp 0038 . 870 221 678 s									
NVM 10	H	QID 0x0000	CQyHDBL	Admin SQT QID = 0 0x0002	Time Delta 13.449 ms	Time Stamp 0038 . 870 229 506 s									
QuickTiming markers not set															
Ready												Errors detected!	Search: Fwd		

Namespace List Format

Bytes	Description
3:0	Identifier 0
7:4	Identifier 1
...	...
(N*4+3):(N*4)	Identifier N

Following the last namespace in
the list, all entries must be 0h.



Controller List Format

Bytes	Description
1:0	Number of Identifiers
3:2	Identifier 0
5:4	Identifier 1
...	...
(N*2+3):(N*2+2)	Identifier N

Namespace Management Command

Added in Rev 1.2

Allows user to create or delete Namespaces

Host passes Data Structure similar to Identify Namespace
(see next page)

Create: NSID field is ignored, Controller uses any inactive NSID

Delete: NSID field specifies namespace to delete

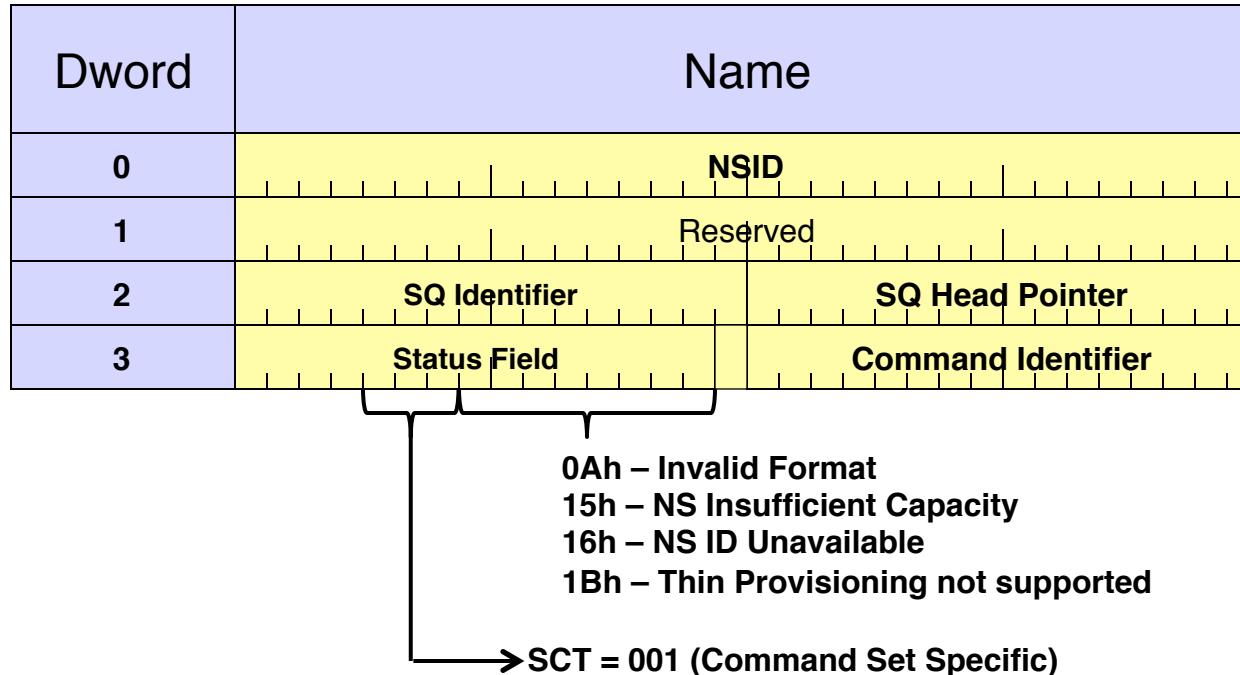
Bits	Description
31:04	Reserved
03:00	0h – Create 1h – Delete

Data Structure for Namespace Management Command

Bytes	Host Specified	Description
7:0	Yes	Namespace Size
15:8	Yes	Namespace Capacity
25:16		Reserved
26	Yes	LB Format from bytes 128-191 of Identify Namespace
28:27		Reserved
29	Yes	End to End Data Protection Type and Settings
30	Yes	NS Multi-path I/O and NS Sharing Capabilities
383:31		Reserved
1023:384		Reserved
4095:1024		Vendor Specific

For definition of fields, see Identify Namespace data structure

Namespace Management Command Completion



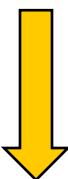
Successful completion is indicated by SCT = 000b and Status = 00h

Namespace Attachment Command

Allows attachment and detachment of controllers to namespaces

PRP Entry points to the 4096 byte list of controllers

Command Dword 10 specifies to attach or detach



Bits	Description
31:04	Reserved
03:00	0h – Controller Attach 1h – Controller Detach

Features



Feature Identifiers (FID <80h)

FID	Persistent	Memory Buffer	M/O	Description
01h	No	No	M	Arbitration
02h	No	No	M	Power Management
03h	Yes	Yes	O	LBA Range Type
04h	No	No	M	Temperature Threshold
05h	No	No	M	Error Recovery
06h	No	No	O	Volatile Write Cache
07h	No	No	M	Number of Queues
08h	No	No	M	Interrupt Coalescing (PCIe only)
09h	No	No	M	Interrupt Vector Configuration (PCIe only)
0Ah	No	No	M	Write Atomicity
0Bh	No	No	M	Asynchronous Event Configuration
0Ch	No	Yes	O	Autonomous Power State Transition
0Dh	No	Yes/No*	O	Host Buffer Memory
0Eh	No	No	O	Timestamp
0Fh	No	No	O	Keep Alive Timer
10h	Yes	No	O	Host Controlled Thermal Management
11h	No	No	O	Non-Operational Power State Configuration

* Yes for Get Features, No for Set Features

NVM Express

Feature Identifiers (FID >7Fh)

FID	Persistent	Memory Buffer	M/O	Description
80h	Yes	No	O	Software Progress Marker
81h	No	Yes	O	Host Identifier
82h	No	No	O	Reservation Notification Mask
83h	Yes	No	O	Reservation Persistence



* Yes for Get Features, No for Set Features

NVM Express

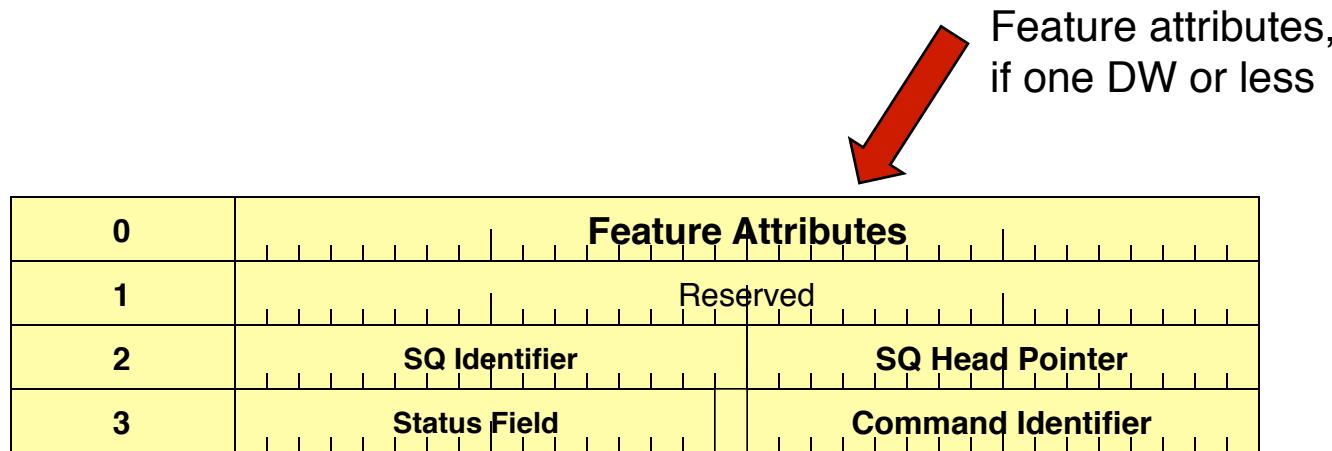
Get Features Command

Dword	Bytes	Name		
0	03:00	Common Fields		Op Code 0Ah
1-5	23:04	Common Fields		
6	27:24	PRP Entry 1		
7	31:28	Data buffer for returned information		
8	35:32	PRP Entry 2		
9	39:36	2 nd data buffer for returned information		
10	43:40	Reserved	SEL	Feature ID
11-15	63:44	Reserved		

SEL – Select

- 000b – Current
- 001b – Default
- 010b – Saved
- 011b – Supported Capabilities

Get Features Command Completion



Successful completion is indicated by SCT = 000b and Status = 00h

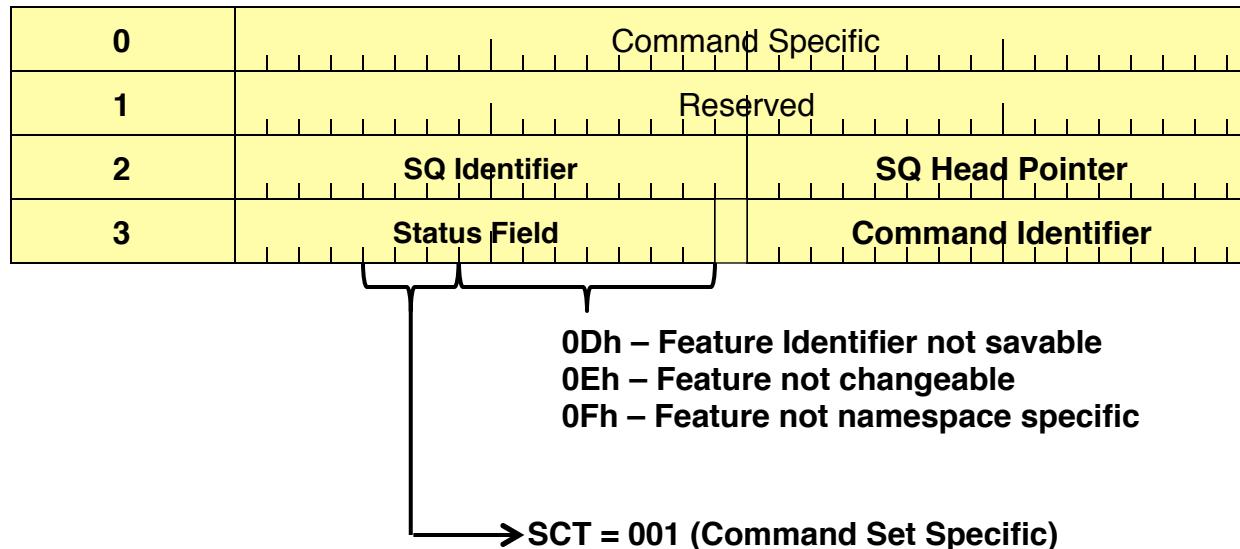
Set Features Command

Dword	Bytes	Name	
0	03:00	Common Fields	Op Code 09h
1-5	23:04	Common Fields	
6	27:24	PRP Entry 1 Data buffer if feature info is in a data structure	
7	31:28		
8	35:32	PRP Entry 2	
9	39:36	2 nd data buffer if feature info is in a data structure	
10	43:40	S Reserved	Feature ID
11	47:44	Feature Attributes	
12	51:48	Reserved	
13	55:52	Reserved	
14	59:56	Reserved	
15	63:60	Reserved	

Feature attributes, if one DW or less

S – Save attribute through power states and resets

Set Features Command Completion



Successful completion is indicated by SCT = 000b and Status = 00h

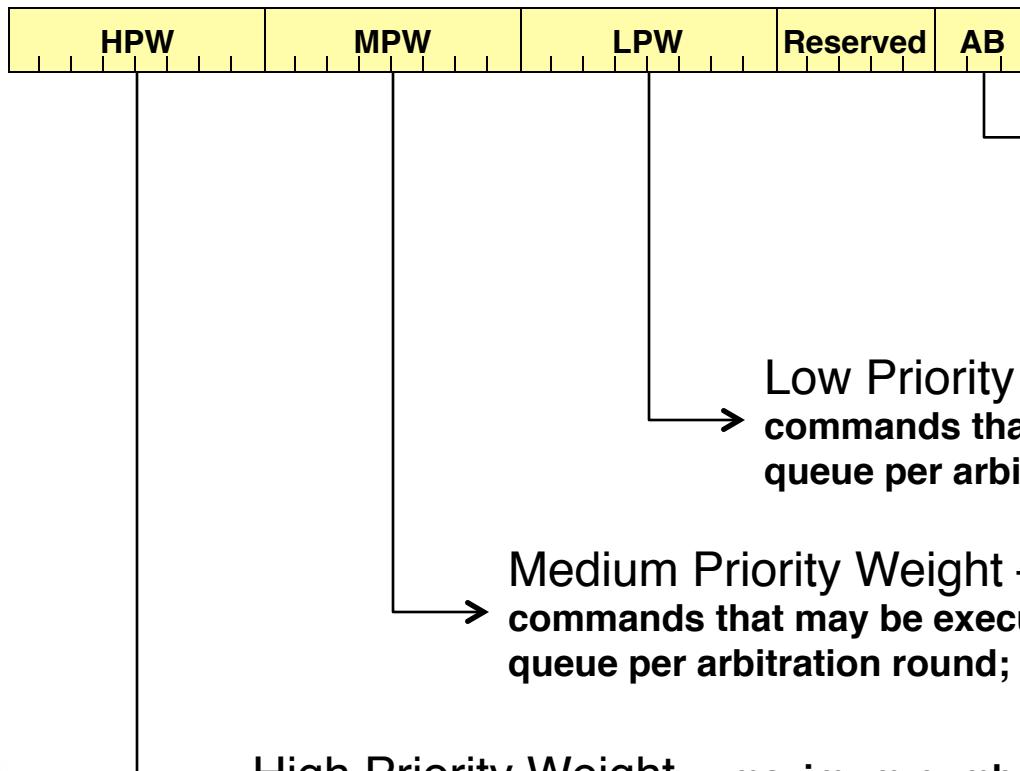
Features Defined – Dword

Arbitration (FID = 01h)

Set Feature – Command Dword 11

Get Feature – Values returned in Dword 0

Reference back two pages
Reference back three pages



Arbitration Burst – maximum number of commands that controller may launch at one time from a submission queue.
 2^{AB} ; $2^0 = 1$; $2^7 = \text{no limit}$

Low Priority Weight – maximum number of commands that may be executed from low priority queue per arbitration round; 0 based value

Medium Priority Weight – maximum number of commands that may be executed from medium priority queue per arbitration round; 0 based value

High Priority Weight – maximum number of commands that may be executed from high priority queue per arbitration round; 0 based value

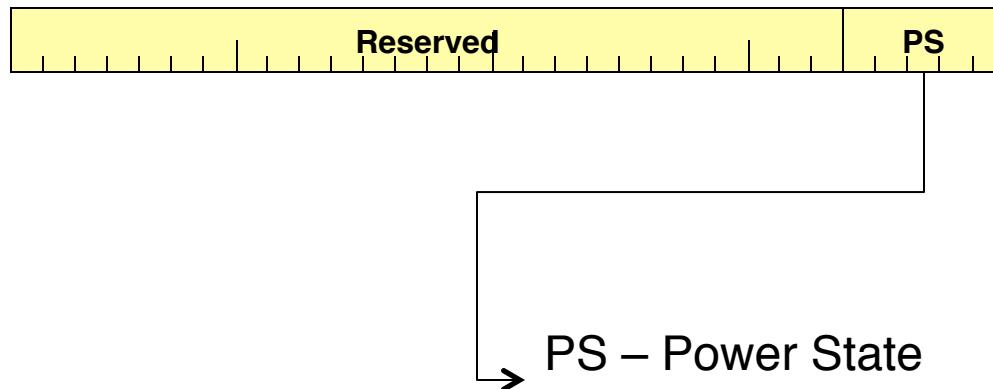


Features Defined – Dword

Power Management (FID = 02h)

Set Feature – Command Dword 11

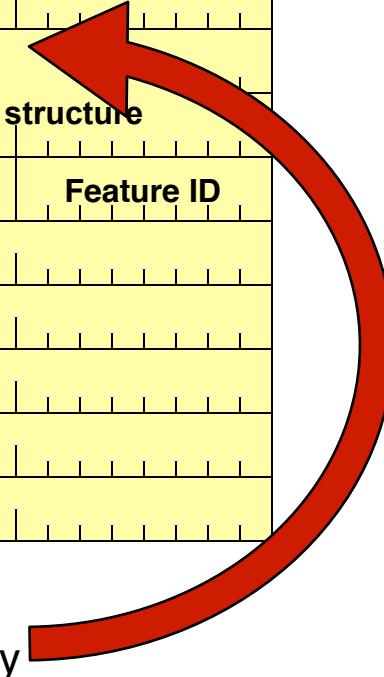
Get Feature – Values returned in Dword 0



Get/Set Features Command

Dword	Bytes	Name		
0	03:00	Common Fields		Op Code 0A/09h
1-5	23:04	Common Fields		
6	27:24	PRP Entry 1		
7	31:28	Data buffer if feature info is in a data structure		
8	35:32	PRP Entry 2		
9	39:36	2 nd data buffer if feature info is in a data structure		
10	43:40	X	Reserved	Feature ID
11	47:44		XXX	
12	51:48		Reserved	
13	55:52		Reserved	
14	59:56		Reserved	
15	63:60		Reserved	

Buffer Pointer(s), if necessary

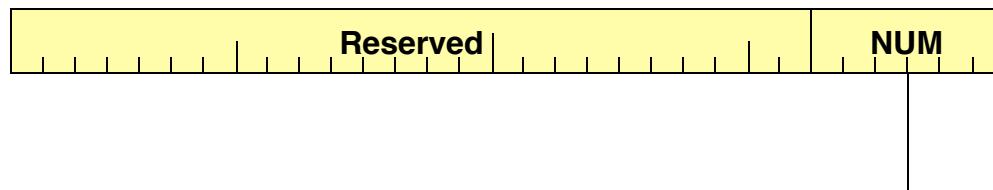


Features Defined – Using Memory Buffer

LBA Range Type (FID = 03h)

Set Feature – Command Dword 11

Get Feature – Values returned in Dword 0



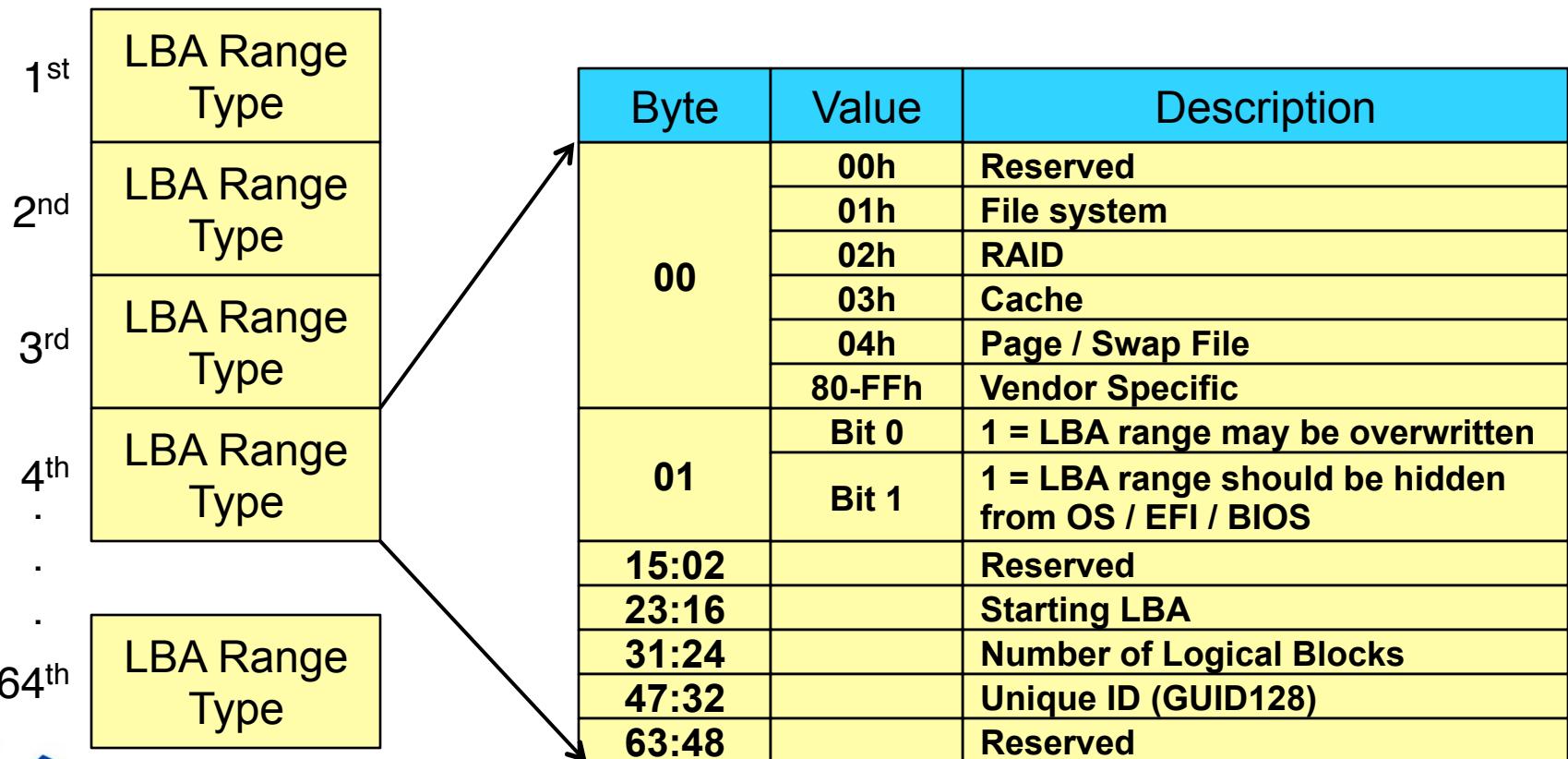
→ **NUM – Number of LBA Ranges**
0 based value

Features Defined

LBA Range Type (FID = 03h)

Set Feature – Data Structure

Get Feature – Data Structure



Data structure is 4096 bytes and shall be physically contiguous



Unique IDs in NVMe

Namespace (EUI64)

First assigned in NVMe 1.1

Bytes 127:120 of Identify Namespace

24-bit or 36-bit Company ID assigned by IEEE

Namespace (NGUID128)

First assigned in NVMe 1.2

Bytes 119:104 of Identify Namespace

Network Qualified Name (NQN)

Up to 223 bytes long; added in NVMe 1.2.1

UTF-8 characters

Used to uniquely describe a host on NVM subsystem

Used for identification and authentication

LBA Range Type (GUID128) (SET/GET Feature)

Specifies Type of LBA range

See NVMe.org for well-known types

First assigned in NVMe 1.0

Bytes 47:32 of LBA Range Structure



EFI (Extensible Firmware Interface) (GUID128)

Defined in NVMe 1.0,

not mentioned in NVMe 1.1, 1.2, 1.2.1, or NVME over Fabrics

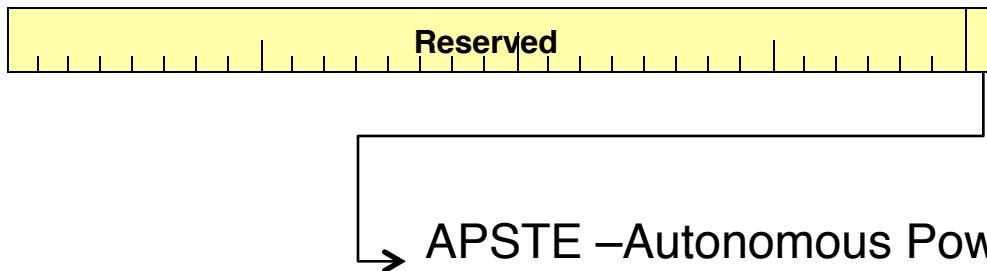
51116915-C34B-4D8E-86DB-6A70F2E60DAA

Features Defined – Using Memory Buffer

Autonomous Power State Transition (FID = 0Ch)

Set Feature – Command Dword 11

Get Feature – Values returned in Dword 0



Features Defined

Autonomous Power State Transition
(FID = 0Ch)



After the device has been idle in this state for the time in bits 31:08, it will transition to the state specified in bits 07:03

Bits	Description
02:00	Reserved
07:03	Idle Transition Power State
31:08	Idle Time Prior to Transition in ms. 0 = disable transition
63:32	Reserved



Unused power state structures must be cleared to 0.

Power states must begin with power state 0 in bytes 7:0 then increase sequentially.

Power State 0 is mandatory, 1 – 31 are optional

Host Memory Buffer - Notes

Host provides a range of host memory addresses for exclusive access by controller

Purpose is vendor specific

Once enabled, host should not change memory

Host may provide limited or no host buffer memory; controller is required to operate properly regardless

Host S/W should request that the controller release memory before a shutdown event or reset

HMB is not persistent across resets

Also see
Controller
Memory Buffer in
Registers Section



Features Defined

Host Memory Buffer (FID = 0Dh)

Dword	Name	
0	Common Fields	Op Code 09h
1-9	Common Fields	
10	S Reserved	Feature ID
11	Reserved	M E
12	Host Memory Buffer Size in CC.MPS	
13	Host Memory Buffer Descriptor List Address	
14		
15	Host Memory Descriptor List Entry Count	

 M – Memory Return to a previously allocated memory
E – Enable host memory feature

Host Memory Buffer - Operation

Controller provides desired size in Identify Controller Data structure

Preferred size – bytes 275:272

Minimum size – bytes 279:276

Host issues Set Features to create Host Memory Buffer

Enable = 1b

Size

Location of descriptor list. Each entry contains:

Memory Address

Size

Host issues Set Features to disable Host Memory Buffer

Enable = 0b

Also see
Controller
Memory Buffer in
Registers Section



Features Defined Review

Using the PCIe/NVMe Reference Manual, or the NVMe Specification, please answer the following questions:

1. What does a value in the temperature threshold feature mean?
2. In what increment is the time limit for limited error recovery?
3. What is Feature ID 6?
4. What does “Time” and “THR” mean for Interrupt Coalescing Feature?
5. What feature is used to tell the controller to autonomously change power states?
6. What feature registers a host identifier with a controller?

Miscellaneous Other Commands



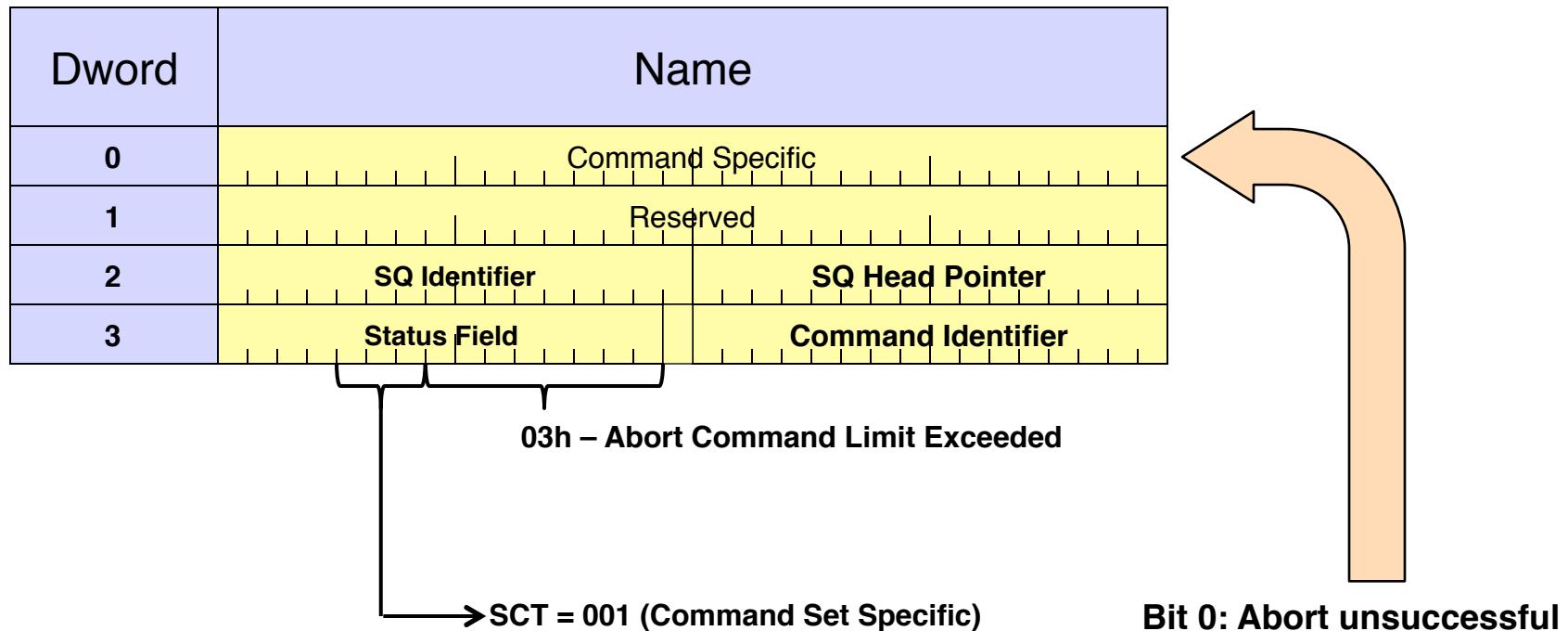
Abort Command

Dword	Bytes	Name	
0	03:00	Common Fields	Op Code 08h
1-9	39:04	Common Fields	
10	43:40	Command ID of Cmd to be aborted	SQID
11	47:44	Reserved	
12	51:48	Reserved	
13	55:52	Reserved	
14	59:56	Reserved	
15	63:60	Reserved	

Function:

Abort the single command identified by Dword 10
Controller is to use best effort

Abort Command Completion – Status Field



Asynchronous Event Request

Dword	Bytes	Name	
0	03:00	Common Fields	Op Code 0Ch
1-9	39:04	Common Fields	
10-15	63:40	Reserved	

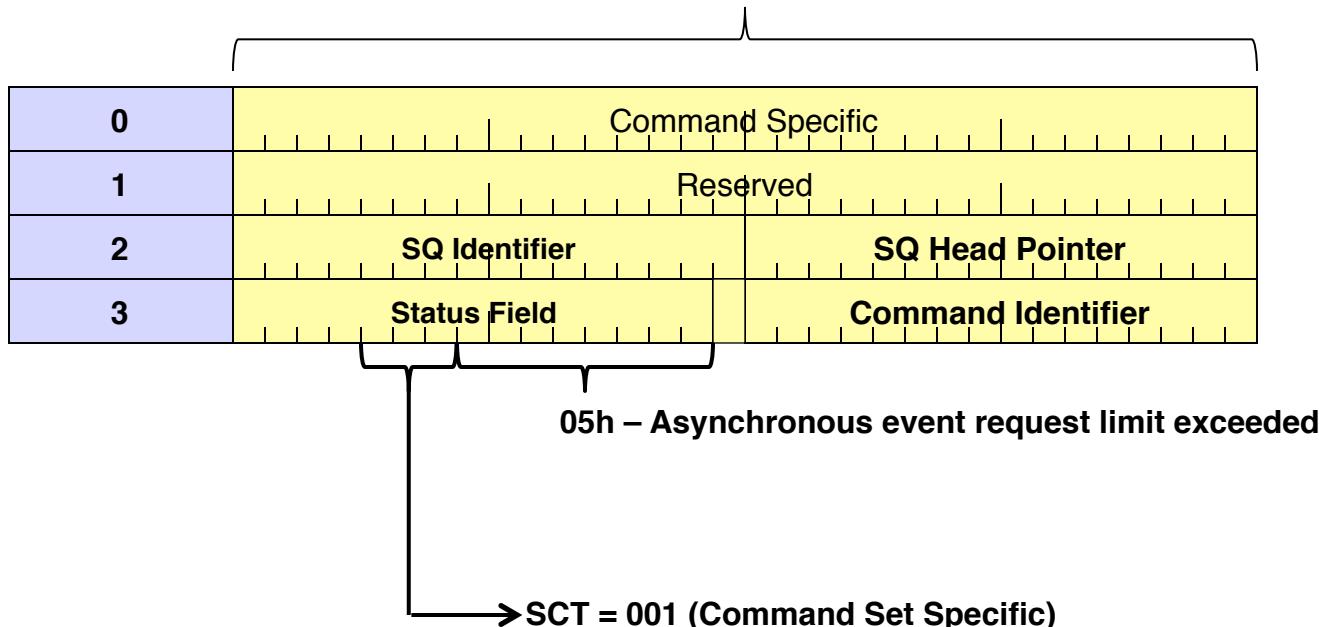
Function: Provides a method for the controller to report asynchronous events to the host.

More than one Asynchronous Event Requests may be outstanding at a time to prevent latency.

**Host issues one or more AER commands;
controller completes command when it has an event to report
(command has no timeout)**

Asynch Event Request Completion

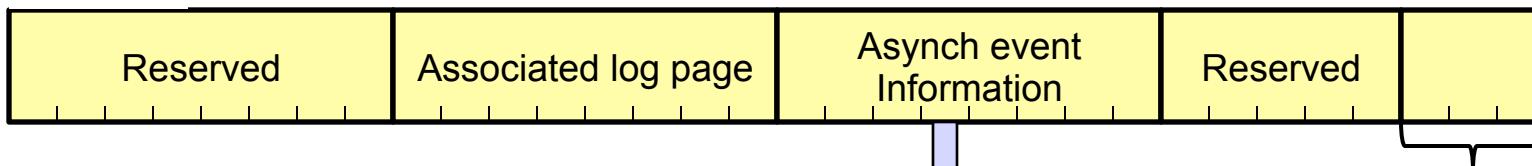
Dword 0 – see next page



Successful completion is indicated by SCT = 000b and Status = 00h

Asynch Event Request Completion – Dword 0

Dword 0



Asynch event type

- 0h – Error status
- 1h – SMART /Health status
- 6h – I/O Command Set specific status
- 7h – Vendor specific

Asynch event information			
	Error status	SMART / Health status	I/O Cmd Set Status
00h	Invalid Submission Queue	Device reliability	Reservation Log page Available
01h	Invalid Doorbell Write value	Temperature above threshold	Reserved
02h	Diagnostic failure	Spare below threshold	
03h	Persistent Internal device error		
04h	Transient internal device error		
05h	Firmware Image Load Error	Reserved	

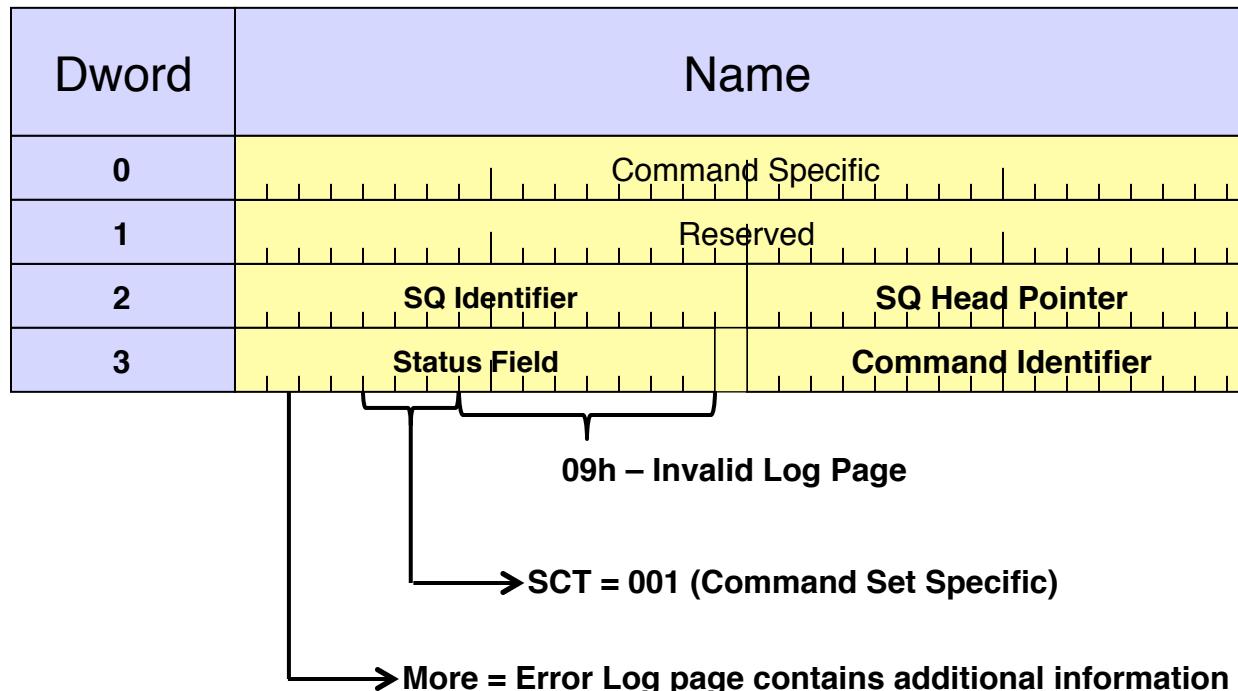


Get Log Page Command

Dword	Bytes	Name			
0	03:00		Common Fields		Op Code 02h
1-5	23:04		Common Fields		
6-7	31:24		PRP Entry 1 – 1 st buffer for data		
8-9	39:32		PRP Entry 2 (if required) – 2 nd buffer for data		
10	43:40	Reserved	Number of DWords	Reserved	Log Page ID
11 - 15	63:44		Reserved		

Log Page ID	Description
00h	Reserved
01h	Error Information
02h	SMART / Health Information
03h	Firmware Slot Information
04h	Changed Namespace List
05h	Command Effects Log
80h - BFh	I/O Command Set Specific
C0h - FFh	Vendor Specific

Get Log Page Command Completion – Status Field



Successful completion is indicated by SCT = 000b and Status = 00h

Log Page 1 – Error Information

Byte	Name	
00	Error Count, 1's based	
07	Sticky, unique identifier for this error	
08	SQ ID	
09		
10	Command ID	
11		
12	Status Field	P
13		
14	Parameter Error Location	
15		
16	Failing LBA	
23		
24	Namespace ID	
27		
28	Vendor Specific Information	
29	Reserved	
31		
32	Command Specific Information	
39		
40	Reserved	
63		

Log Page 2 – SMART/Health Information

Critical
Warning



Byte	Name						
00	Reserved	BU	RO	DR	Temp	AS	
2:1	Temperature of device in Kelvin						
3	Available Spares remaining in percentage						
4	Available Spare Threshold in percentage						
5	Percentage of estimated device life used						
31:6	Reserved						
47:32	Number of 512 byte data units read in thousands, not metadata						
63:48	Number of 512 byte data units written in thousands, not metadata						
79:64	Number of Read commands completed						
95:80	Number of Write commands completed						
111:96	Minutes controller busy with I/O commands -						
127:112	Number of power cycles						
143:128	Power-on hours, not in low power state						
159:144	Number of Unsafe Shutdowns						
175:160	Number of controller detected unrecovered data integrity errors						

Log Page 2 – SMART/Health Information (concluded)

Byte	Name
191:176	Number of Error Information Log Entries
195:192	Warning Composite Temperature Time
199:196	Critical Composite Temperature Time
201:200	Temperature Sensor 1
203:202	Temperature Sensor 2
205:204	Temperature Sensor 3
207:206	Temperature Sensor 4
209:208	Temperature Sensor 5
211:210	Temperature Sensor 6
213:212	Temperature Sensor 7
215:214	Temperature Sensor 8
511:216	Reserved

Log Page 3 – Firmware Slot Information

Byte	Name		
00	R	Next load slot	R
01	Reserved		
07			
08	F/W Revision for Slot 1		
15			
16	F/W Revision for Slot 2		
23			
24	F/W Revision for Slot 3		
31			
32	F/W Revision for Slot 4		
39			
40	F/W Revision for Slot 5		
47			
48	F/W Revision for Slot 6		
55			
56	F/W Revision for Slot 7		
63			
64	Reserved		
511			

Log Page 4 – Changed Namespace List

Log page is a list of up to 1024 namespaces that have changed NS information since the last time this log page was read.

Log Page 5 – Commands Supported and Effects

Log page is a list of up to 1024 namespaces that have changed NS information since the last time this log page was read.

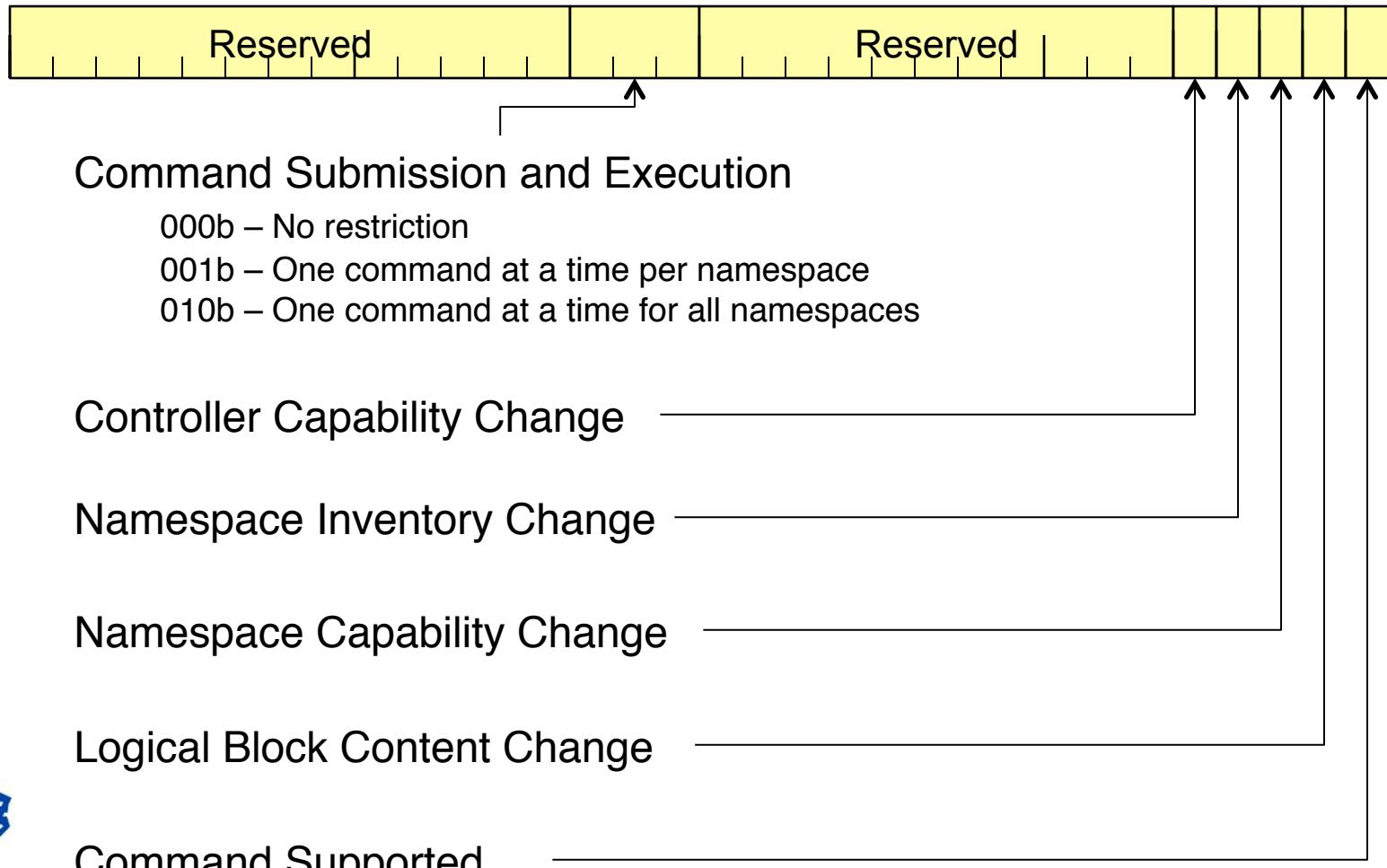
Byte	Name
03:00	Admin Command Supported with opcode 0
07:04	Admin Command Supported with opcode 1
...	...
1019:1016	Admin Command Supported with opcode 254
1023:1020	Admin Command Supported with opcode 255
1027:1024	I/O Command Supported with opcode 0
1031:1028	I/O Command Supported with opcode 1
...	...
2043:2040	I/O Command Supported with opcode 254
2047:2044	I/O Command Supported with opcode 255
4095:2048	Reserved

Structure on next page

NVM Express

Section 5: Admin Commands

Log Page 5 – Command Effect Data Structure



Log Page 6 – Device Self-Test

Byte	Name														
00	Reserved Current test in operation 0h – No Device Self-test 1h – Short test 2h – Extended test Eh – Vendor specific test														
01	Res	Percentage Complete													
3:2	Reserved														
31:4	Newest Self-Test Result Data Structure														
59:32	2 nd Newest Self-Test Result Data Structure														
535:508	19 th Newest Self-Test Result Data Structure														
563:536	20 th Newest Self-Test Result Data Structure														

Device Self-test Data Structure

Byte	Name					
00	Test type for this Status 1h – Short test 2h – Extended test Eh – Vendor Specific	Result of test that created this entry				
01	1 st failing segment number					
2	Reserved	SC valid	SCT valid	FLBA valid	NSID valid	
3	Reserved					
11:4	Power On Hours					
15:12	NS ID					
23:16	Failing LBA					
24	Status Code Type					
25	Status Code					
27:26	Vendor Specific					

Result of test

0h – No error
 1h – Aborted by Test cmd
 2h – Aborted by CLR
 3h – Aborted by removal of NS
 4h – Aborted by Format NVM cmd
 5h – Fatal or Unknown error

6h – Unknown segment failed
 7h – 1 or more segments failed, first failed segment identified
 8h – Aborted for unknown reason
 Fh – This entry not used



Log Page 7 – Host-Initiated Telemetry Log

Byte	Name
00	Log Page ID = 07h
4:1	Reserved
7:5	IEEE OUI
9:8	Data Area 1 Last Block
11:10	Data Area 2 Last Block
13:12	Data Area 3 Last block
381:14	Reserved
382	Telemetry Controller-Initiated Data Available
383	Telemetry Controller-Initiated Data Generation Number
511:384	Reason Identifier
1023:512	Data Block 1
1535:1024	Data Block 2
(N*512) + 511: (N*512)	Data Block N



Contents of Data Block are Manufacturer Unique

Log Page 8 – Controller-Initiated Telemetry Log

Byte	Name
00	Log Page ID = 08h
4:1	Reserved
7:5	IEEE OUI
9:8	Data Area 1 Last Block
11:10	Data Area 2 Last Block
13:12	Data Area 3 Last block
381:14	Reserved
382	Telemetry Controller-Initiated Data Available
383	Telemetry Controller-Initiated Data Generation Number
511:384	Reason Identifier
1023:512	Data Block 1
1535:1024	Data Block 2
(N*512) + 511: (N*512)	Data Block N



Contents of Data Block are Manufacturer Unique

Log Page 80 – Reservation Notification Log

Byte	Name
00	
07	Log Page count, 1's based
08	Reservation Notification Log Page Type 00h – Empty Log Page 01h – Registration Preempted 02h – Reservation Released 03h – Reservation Preempted FF:04h – Reserved
09	Number of Available (unread) Log Pages
10	Reserved
11	
12	
15	Namespace ID
16	Reserved
63	

Log Page 81 – Sanitize Status

Byte	Name	
1:0	Sanitize Progress in fraction over 65,536	
	Sanitize Status	
3:2	Reserved	
	Completed passes for Overwrite	000b – Subsystem has never been sanitized 001b – Most recent sanitize op completed successfully 010b – Sanitize op currently in progress 011b – Most recent sanitize op failed
7:4	Sanitize CDW10 Information	
11:8	Estimated Time for Overwrite in seconds	
15:12	Estimated Time for Block Erase in seconds	
19:16	Estimated Time for Crypto Erase in seconds	
511:20	Reserved	

Firmware Download Command

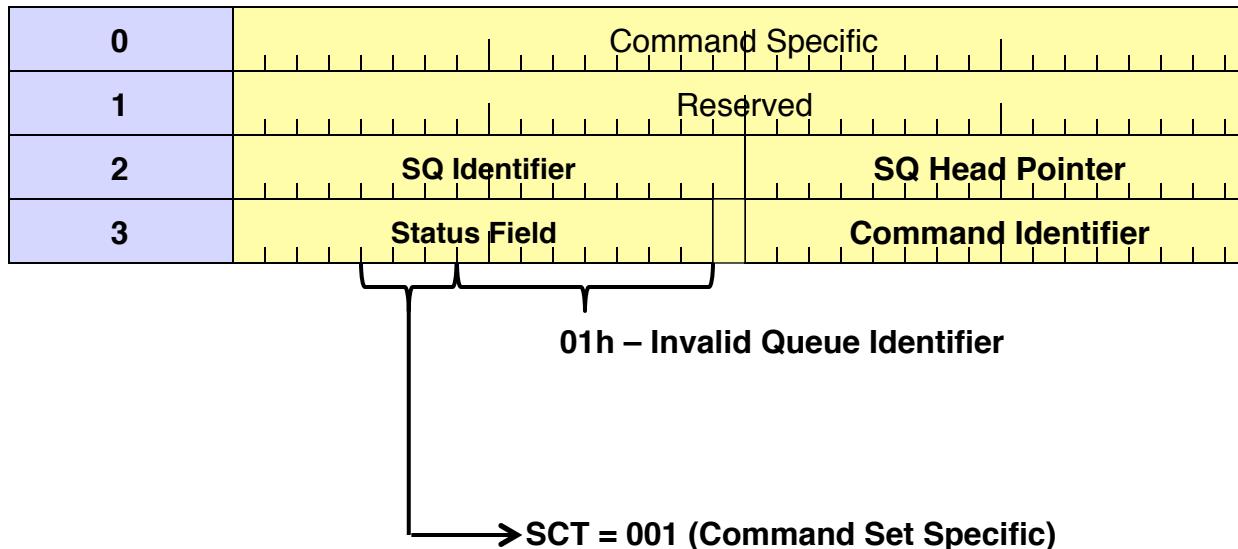
Dword	Bytes	Name
0	03:00	Common Fields Op Code 11h
1-5	23:04	Common Fields
6-7	31:24	PRP Entry 1 – 1 st buffer of data to download
8-9	39:32	PRP Entry 2 (if required) – 2 nd buffer data buffer or pointer
10	43:40	Number of DWords
11	47:44	Offset if downloading in multiple pieces
12	51:48	Reserved
13	55:52	Reserved
14	59:56	Reserved
15	63:60	Reserved



PRP Entry 1 – Specifies 1st location of data for download
PRP Entry 2 – Specifies 2nd location of data for download or pointer to PRP list

NVM Express

Firmware Download Completion



Successful completion is indicated by **SCT = 000b** and **Status = 00h**

Firmware Commit Command

Dword	Bytes	Name			
0	03:00		Common Fields		Op Code 10h
1-9	39:04		Common Fields		
10	43:40	B	Reserved		CA FS
11-15	63:44		Reserved		

CA – Activate Action

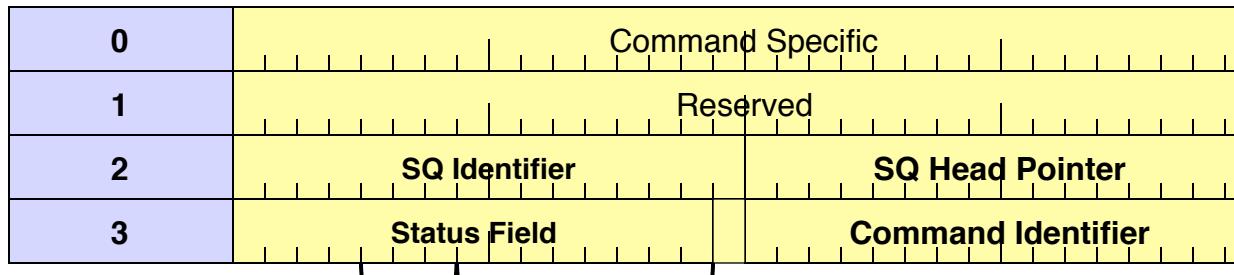
- 000b – Downloaded image replaces image indicated by FS; new image not activated
- 001b – Downloaded image replaces image indicated by FS;
new image activated at next reset
- 010b – Image indicated by FS is activated at next reset
- 011b – Image indicated by FS is requested to be activated immediately without reset
- 110b – D/L image replaces the Boot Partition specified by Boot Partition ID
- 111b – Mark Boot Partition active, update BPINFO.ABPID

FS – Firmware slot

0b – controller choose the firmware slot (1-7) for the operation

B – Boot Partition ID

Firmware Commit Completion



06h – Invalid Firmware slot
07h – Invalid Firmware image
0Bh – F/W application requires Conventional Reset
10h – F/W application requires NVM Subsystem Reset

→ SCT = 001 (Command Set Specific)



Successful completion is indicated by SCT = 000b and Status = 00h

Keep Alive Command

Dword	Bytes	Name	
0	03:00	Common Fields	Op Code 18h
1-9	39:04	Common Fields	
10-15	63:40	Reserved	

Keep Alive timer is a watchdog timer

Controller indicates the granularity of this timer in the Identify Controller

Controller has a keep alive timeout. If that time expires without a Keep Alive command, the controller signals a Keep Alive Timeout to the host.

Records an Error Information Log Entry

Status code Keep Alive Timeout Expired

Controller Fatal Status = 1b



NVM Command Set Specific Commands



NVM Command Set Specific Commands

Op Code	Command	Namespace Used
80h	Format NVM	Yes
81h	Security Send	Yes
82h	Security Receive	Yes

These are Admin Commands and are issued to the Admin Submission Queue

Format NVM Command

Dword	Bytes	Name	
0	03:00	Common Fields	Op Code 80h
1-9	39:04	Common Fields	
10	43:40	Format NVM Control	
11	47:44	Reserved	
12	51:48	Reserved	
13	55:52	Reserved	
14	59:56	Reserved	
15	63:60	Reserved	

Perform low level format on the NVM media to

- change LB data size,
- change metadata size,
- change or set protection information, or
- perform Secure Erase.

Format NVM Command – Dword 10

PI – Protection Information

000b – PI is not enabled

001b – PI Type 1 enabled

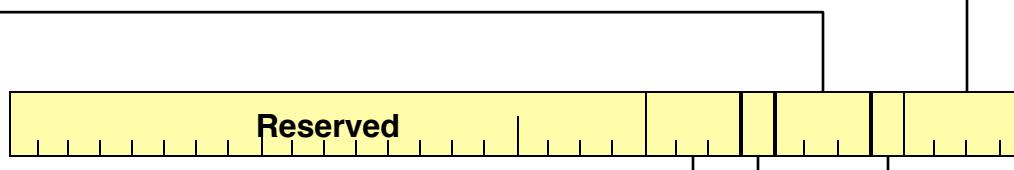
010b – PI Type 2 enabled

011b – PI Type 3 enabled

LBAF – LBA Format

Specifies which LBA format to apply.

LBA formats are defined in Identify data.



MS – Metadata settings

1 = metadata is part of an extended data LBA.

0 = metadata is transferred as part of a separate buffer

PIL – Protection Information Location

1 = PI is first 8 bytes of metadata

0 = PI is last 8 bytes of metadata

SES – Secure Erase Settings

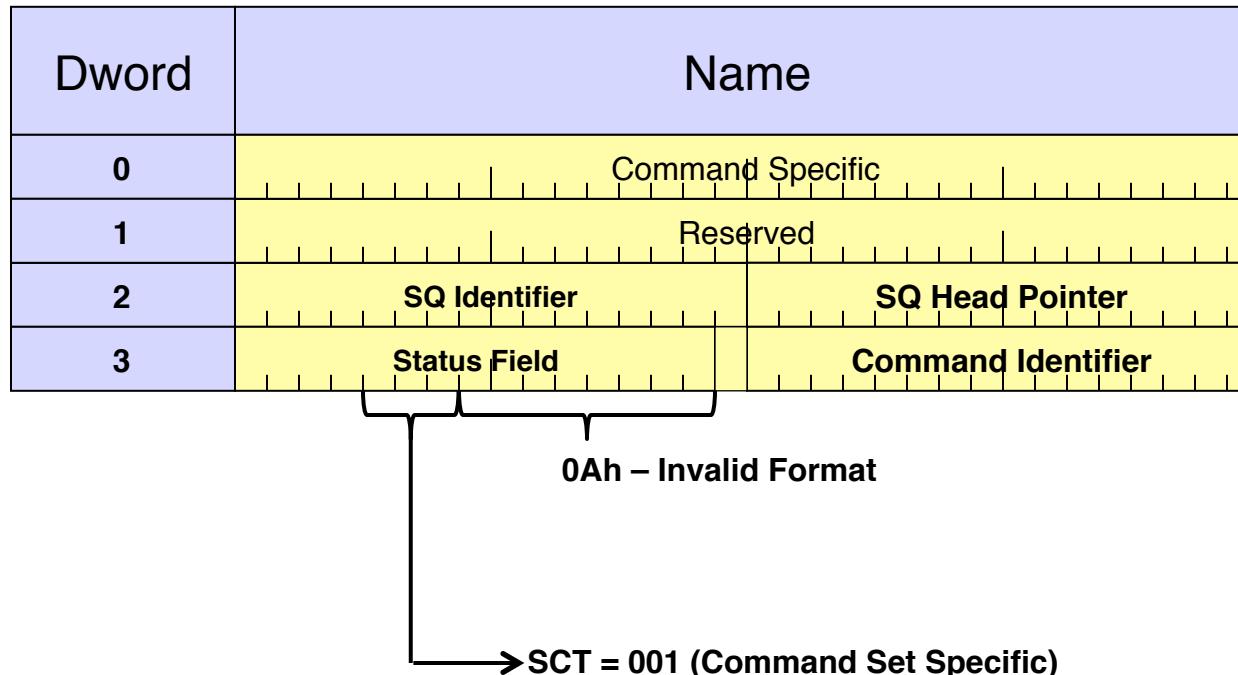
000b = No secure erase op requested

001b = User Data erase

010b = Crypto Erase



Format NVM Completion – Status Field



Successful completion is indicated by SCT = 000b and Status = 00h

Security Commands

For Details on Security Send and Security Receive commands, please refer to www.t10.org, document SFSC (Security Features for SCSI Commands)



Covered in this Section

NVMe Admin Commands



Notes



Notes



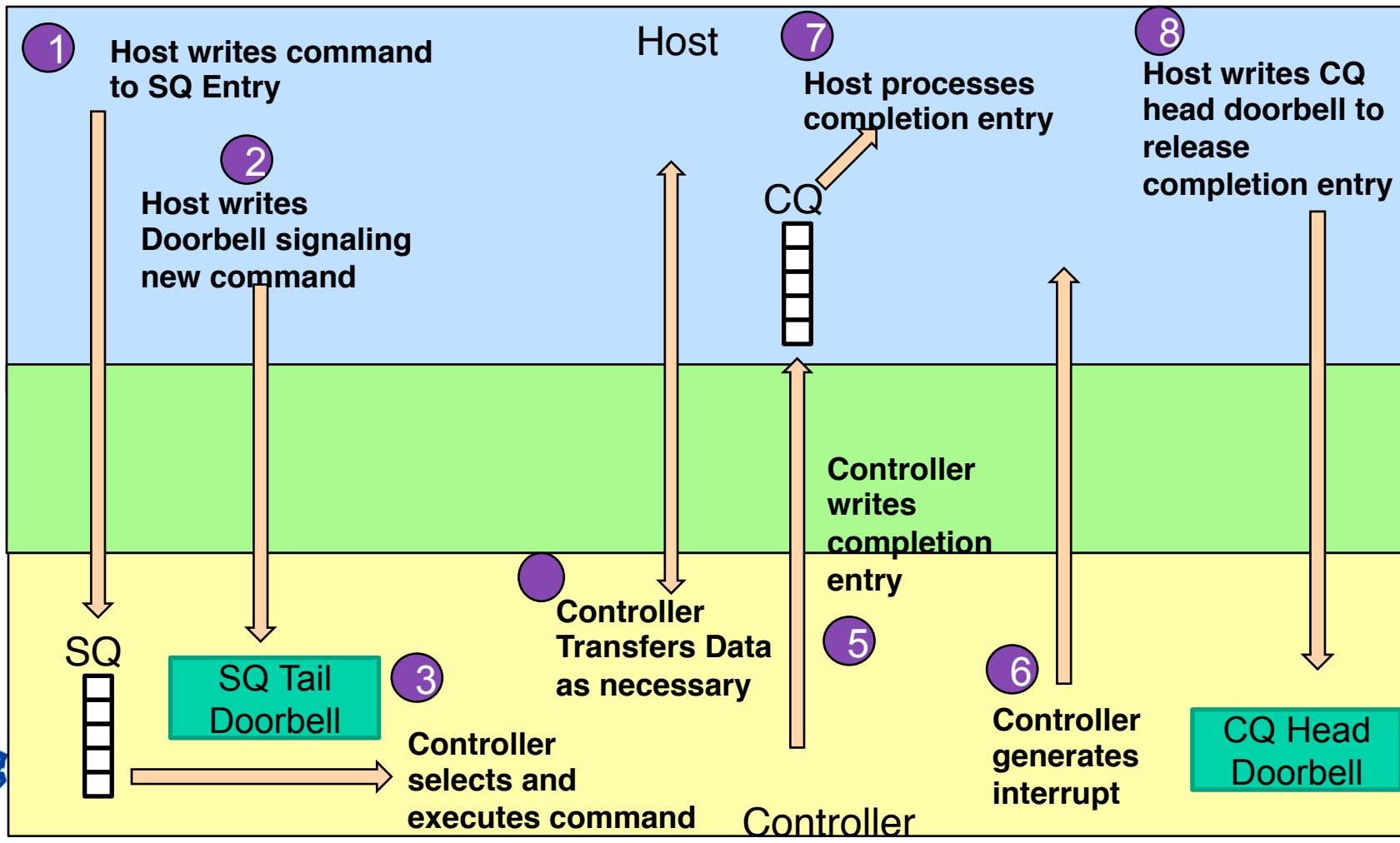
Section 6

NVMe Commands



Covered in this Section

NVMe Commands



NVMe Commands

M – Mandatory
O – Optional
M/O – Mandatory if Reservations are supported

Op Code	M/O	Command
00h	M	Flush
01h	M	Write
02h	M	Read
04h	O	Write Uncorrectable
05h	O	Compare
08h	O	Write Zeros
09h	O	Dataset Management
0Dh	M/O	Reservation Register
0Eh	M/O	Reservation Report
11h	M/O	Reservation Acquire
15h	M/O	Reservation Release
80-FFh	O	Vendor Specific

Command Re-ordering Rules

Controller has no restriction on reordering commands
whether in different SQs or the same SQ
except fused commands

Fused commands must be executed in the indicated sequence and
as though no other commands intervened



Command Atomic Rules

Problem:

Write command fails leaving some blocks with new data and some with old.

Solution:

Atomic Write Unit is guaranteed to either write completely or not at all.

Atomic Write Unit:

Size of write operation guaranteed to be written atomically to media

Value may be different for normal operations and power fail operations

Originally defined for entire controller only

NVMe 1.2 added per Namespace limits

If a write command is submitted

with size less than AWUN,

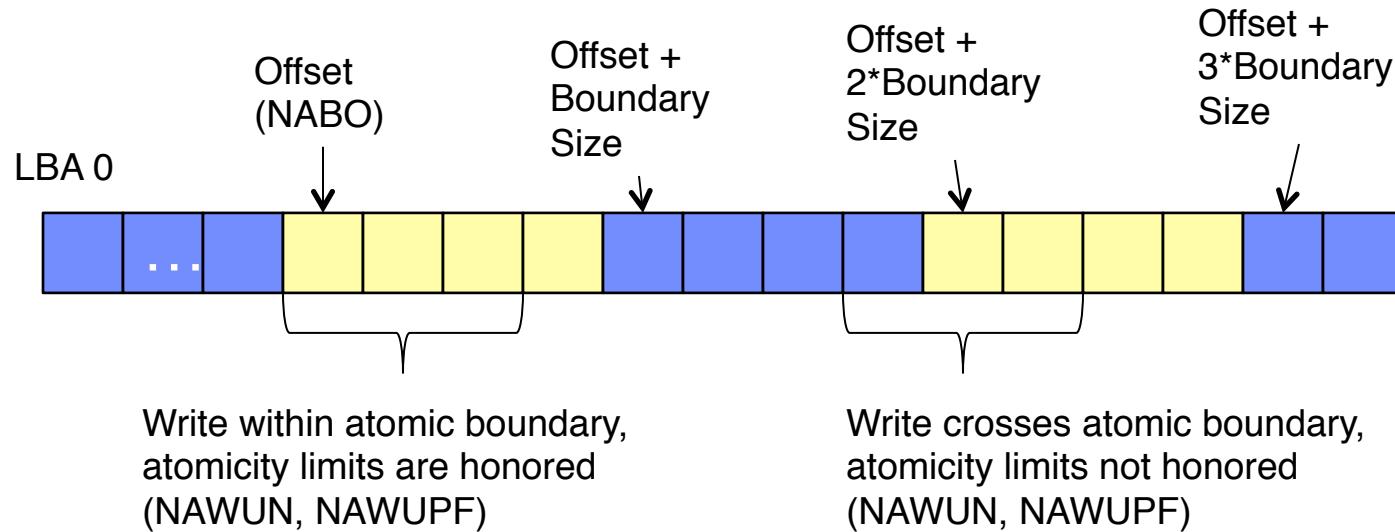
and it does not cross atomic boundary,

then the host is guaranteed that the write command is atomic
with respect to other read and write commands.

AWUN –
Atomic Write
Unit Normal



Atomic Write Boundaries Example



Atomic Write Units

	Parameter Name	Value
Controller Atomic Parameters	Atomic Write Unit Normal (AWUN)	
	Atomic Write Unit Power Fail (AWUPF)	\leq AWUN
	Atomic Compare and Write (ACWU)	
Namespace Atomic Parameters	Namespace Atomic Write Unit Normal (NAWUN)	\geq AWUN
	Namespace Atomic Write Unit Power Fail (NAWUPF)	\geq AWUPF \leq NAWUN
	Namespace Atomic Compare and Write (NACWU)	\geq ACWU
Namespace Atomic Boundary Parameters	Namespace Atomic Boundary Size Normal (NABSN)	\geq NAWUN
	Namespace Atomic Boundary Offset (NABO)	\leq NABSN \leq NABSPF
	Namespace Atomic Boundary Size Power Fail (NABSPF)	\geq NAWUPF



Atomic Write Unit – Number of Logical Blocks
guaranteed to be written atomically

NVM Express

Section 6: NVMe Commands

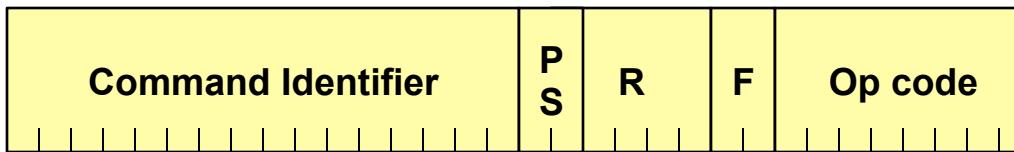
NFG

6 - 7

NVMe Command Format

Dword	Bytes	Name				
0	03:00	Command ID (CID)	P S	Res	F	OP Code
1	07:04	Namespace Identifier (NSID)				
2	11:08	Reserved				
3	15:12					
4	19:16	Metadata Pointer (MPTR) – Address of physical buffer for metadata				
5	23:20					
6	27:24	PRP Entry 1 (PRP1)				
7	31:28	Or SGL Entry 1				
8	35:32	PRP Entry 2 (PRP2)				
9	39:36					
10	43:40					
11	47:44					
12	51:48					
13	55:52	Command specific fields				
14	59:56					
15	63:60					

Command Dword 0



PSDT –
00b = PRP
01b or 10b = SGL
for Data Transfer

Fuse

Bits	Value	Definition
09:08	00b	Normal Operation
	01b	1 st fused command
	10b	2 nd fused command
	11b	Reserved

Command Identifier –
When combined with Submission Queue Identifier, specifies a unique identifier for the command

A complex command is created by “fusing” together two simpler commands.

Command Completion – Generic (Review)

Dword	Name	
0		Command Specific
1		Reserved
2	SQ Identifier	SQ Head Pointer
3	Status Field	Command Identifier

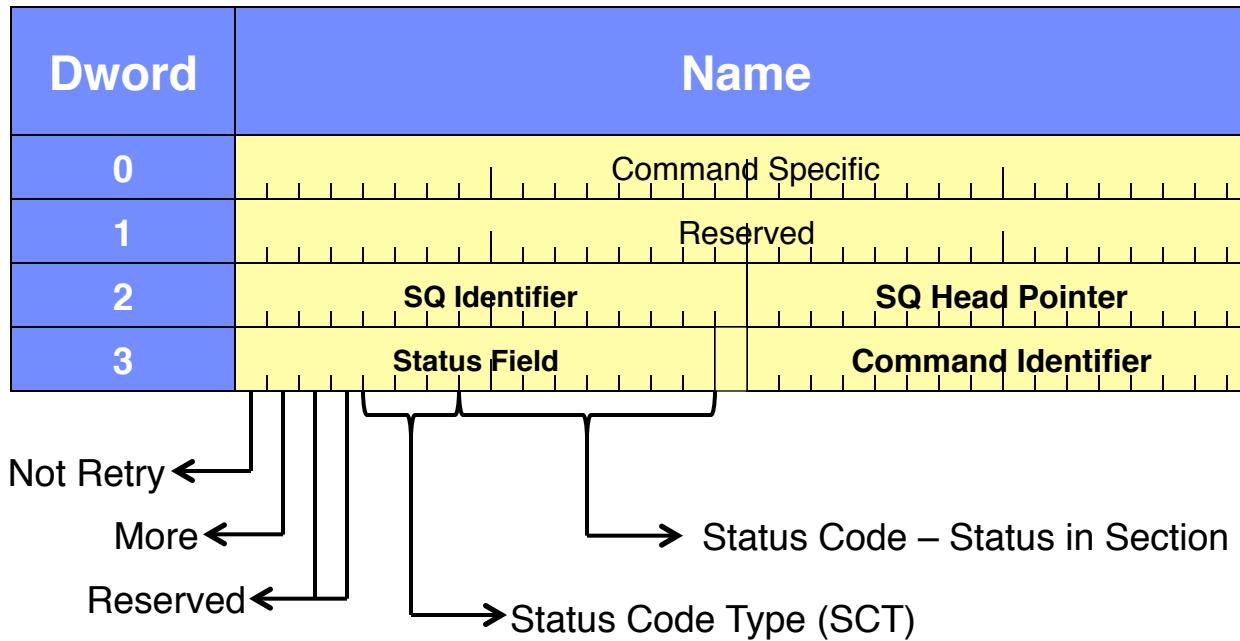
SQ Identifier (SQID): Indicates the Submission Queue for the completed command.

Command Identifier (CID): Indicates the identifier of the command being completed.

SQ Head Pointer (SQHD): Indicates the current head pointer for the Submission Queue that held the associated command.

Phase Tag (P): Identifies whether a Completion Queue entry is new.

Command Completion – Status Field



SCT Value	Description
0	Generic Command Status
1	Command Specific Status
2	Media Errors
7	Vendor Specific

Successful completion is SCT = 0h and Status Code = 00h
NVM Express

Flush Command

Dword	Bytes	Name	
0	03:00	Common Fields	Op Code 00h
1	39:04	Namespace ID	
2-9	39:04	Common Fields (n/a)	
10	43:40	Reserved	
11	47:44	Reserved	
12	51:48	Reserved	
13	55:52	Reserved	
14	59:56	Reserved	
15	63:60	Reserved	

Move any data in volatile storage to non-volatile storage

Command applied to a specified Namespace

Read Command

Read Command

Dword	Bytes	Name		
0	03:00		Common Fields	Op Code 02h
1-3	23:04		Common Fields	
4	23:04		Metadata Pointer	
5	23:04			
6	27:24	PRP 1		
7	31:28		or	SGL
8	35:32	PRP2		
9	39:36			
10	43:40	Starting LBA bits 31:00		
11	47:44	Starting LBA bits 63:32		
12	51:48	Note A	Reserved	Number of LB
13	55:52		Reserved	Note B
14	59:56		Expected Initial LB Reference Tag	
15	63:60	Expected Logical Block Application Tag Mask		Expected Logical Block Application Tag



Read Command Fields

Dword	Bit	Description	
12	31	Time Limited Error Retry	
	30	Force Unit Access (FUA)	
	29:26	Protection Information Field	
13	7	Incompressible	
	6	Sequential Request	
	5:4	Access Latency	
	3:0	Access Frequency	

Note A

Note B

Bit	Meaning if Set to 1
29	Strip Protection Information
28	Enable checking of Guard field
27	Enable checking of Application Tag field
26	Enable checking of Reference Tag field

Value	Definition
00b	None, no latency information provided
01b	Idle, longer latency acceptable
10b	Normal, typical latency
11b	Low, smallest possible latency

Value	Definition
0000b	No frequency information provided
0001b	Typical number of r/w for this LBA range
0010b	Infrequent r/w for this LBA range
0011b	Infrequent w, frequent read
0100b	Frequent w, Infrequent read
0101b	Frequent r/w for this LBA range
0110b	One time read
0111b	Speculative read (prefetch op)
1000b	LBA range to be overwritten soon



NVM Read Command

Teledyne LeCroy PETracer(TM) - PCI Express Protocol Analyzer - [C:\Users\Public\Documents\LeCroy\PETracer\Z3_drive_emulation_boot_and_play_video.pex]																	
File Setup Record Generate Report Search View Tools Window Help																	
Pkt Link Split NVM PQI AHCI																	
NVM	R→	2.5	x8	RequesterID	SQyTDBL	IO SQT QID = 3	Time Delta	Time Stamp									
116				000:00:0		0x0002	220.432 us	0013 . 641 768 552 s									
NVM	R←	2.5	x8	RequesterID	CompleterID	IO Cmd	OPC	FUSE	CID	NSID	MPTR Hi	MPTR Low	PRP1 Hi	PRP1 Low			
117				001:00:0	000:00:0	Read	b00	0x0001	0x00000001	0x00000000	0x00000000	0x00000000	0x00000002	0x1C940000			
						SLBA	LR	FUA	PRINFO	NLB	DSM	Incompressible	SR	AL	AF		
						0x000000002	0x1C941000	0x00000000:00000080	0	0	0x000F	0	0	None	None		
						Time Delta	Time Stamp										
						1.772 ms	0013 . 641 988 984 s										
NVM	R←	2.5	x8	RequesterID	CMD PRP	Addr Hi	Addr Lo	Data Len	Data			Time Delta	Time Stamp				
118				001:00:0		0x00000002	0x1C940000	0x00001000	1024	quadlets		48.888 us	0013 . 643 760 568 s				
NVM	R←	2.5	x8	RequesterID	CMD PRP	Addr Hi	Addr Lo	Data Len	Data			Time Delta	Time Stamp				
119				001:00:0		0x00000002	0x1C941000	0x00001000	1024	quadlets		785.904 us	0013 . 643 809 456 s				
NVM	R←	2.5	x8	RequesterID	Command Completion	SQHD	SQID	CID	P	ST	SC	SCT	M	DNR	Time Delta	Time Stamp	
120				001:00:0	0x00000000	0x0002	0x0003	0x0001	1	0x0000	0x00	0	0	0	24.352 us	0013 . 644 595 360 s	
Link Tra	R←	2.5	TLP	Mem	MWr(32)	Length	RequesterID	Tag	Address			1st BE	Last BE	Data	VC ID	Explicit ACK	Metrics
1515				800	010:00000	1	001:00:0	0	FEE0F00C	1111	0000	1	dword	0	Packet #9109		
					# Packets	Time Delta	Time Stamp										
					2	4.768 us	0013 . 644 619 712 s										
NVM	R→	2.5	x8	RequesterID	CQyHDBL	IO CQH QID = 3	Time Delta	Time Stamp									
121				000:00:0		0x0002	10.068 ms	0013 . 644 624 480 s									



NVM RequesterID 2.5x8 IO SQyTDBL IO SQT QID = 3 Time Delta Time Stamp																		
NVM 116	R→ x8	2.5	RequesterID 000:00:0	SQyTDBL	IO SQT QID = 3 0x0002	Time Delta 220.432 us	Time Stamp 0013.641 768 552 s											
NVM 117	R← x8	2.5	RequesterID 001:00:0	CompleterID 000:00:0	IO Cmd Read	OPC b00	FUSE 0x0001	CID 0x00000001	NSID 0x00000000	MPTR Hi 0x00000000	MPTR Low 0x00000000	PRP1 Hi 0x00000002	PRP1 Low 0x00000002	PRP2 Hi 0x1C940000	PRP2 Low 0x00000002	SLBA 0x00000000:00000080	LR 0	FUA 0
PRINFO NLB DSM Incompressible SR AL AF EILBRT ELBATM ELBATM Time Stamp																		
Split Tra 655	R← x8	2.5	Mem 001:00000	MRd(64) 001:00000	RequesterID 001:00:0	CompleterID 000:00:0	Tag 82	TC 0	VC ID 0	Address 00000002:1EC64040	Status SC	Data 16 dwords	Metrics 2	# LinkTras 0013.641 988 984 s				
Link Tra R← 2.5 TLP Mem MRd(64) Length RequesterID Tag Address 1st BE Last BE VC ID Explicit ACK Metrics # Packets Time Stamp																		
Link Tra 1480	R← x8	2.5	TLP 766	Mem 001:00000	MRd(64) Length 16	RequesterID 001:00:0	Tag 82	Address 00000002:1EC64040	1st BE 1111	Last BE 1111	VC ID 0	Explicit ACK Packet #9039	Metrics 2	# Packets 0013.641 988 984 s				
Packet R→ 2.5 TLP Mem MRd(64) Length RequesterID Tag Address 1st BE Last BE LCRC Time Delta Time Stamp																		
Packet 9038	R→ x8	2.5	TLP 766	Mem 001:00000	MRd(64) Length 16	RequesterID 001:00:0	Tag 82	Address 00000002:1EC64040	1st BE 1111	Last BE 1111	LCRC 0x9E0561E8	Time Delta 116.000 ns	Time Stamp 0013.641 989 100 s					
Link Tra R→ 2.5 TLP Cpl CplID Length RequesterID Tag CompleterID Status BCM Byte Cnt Lwr Addr Data VC ID ExplicitACK Metrics # Packets Time Stamp																		
Link Tra 1481	R→ x8	2.5	TLP 714	Cpl 010:01010	CplID 16	RequesterID 001:00:0	Tag 82	CompleterID 000:00:0	Status SC	BCM 0	Byte Cnt 64	Lwr Addr 0x40	Data 16 dwords	VC ID 0	ExplicitACK Packet #9041	Metrics 2	# Packets 0013.641 989 216 s	
Time Stamp																		
Packet 9040	R→ x8	2.5	TLP 714	Cpl 010:01010	CplID 16	RequesterID 001:00:0	Tag 82	CompleterID 000:00:0	Status SC	BCM 0	Byte Cnt 64	Lwr Addr 0x40	Data 16 dwords	VC ID 0	ExplicitACK Packet #9041	Metrics 2	# Packets 0013.641 989 216 s	
Time Stamp																		
Packet 9041	R→ x8	2.5	DLLP 714	ACK AckNak Seq Num	CRC 16 0x846F	Time Delta 1.770 ms	Time Stamp 0013.641 990 456 s											
NVM 118	R← x8	2.5	RequesterID 001:00:0	CMD PRP	Addr Hi 0x00000002	Addr Lo 0x1C940000	Data Len 0x00001000	Data	Time Delta 48.888 us	Time Stamp 0013.643 760 568 s								
NVM 119	R← x8	2.5	RequesterID 001:00:0	CMD PRP	Addr Hi 0x00000002	Addr Lo 0x1C941000	Data Len 0x00001000	Data	Time Delta 785.904 us	Time Stamp 0013.643 809 456 s								
NVM 120	R← x8	2.5	RequesterID 001:00:0	Command Completion 0x00000000	SQHD 0x0002	SQID 0x0003	CID 0x0001	P ST 1	SCT 0x0000	M DNR 0 0	Time Delta 24.352 us	Time Stamp 0013.644 595 360 s						
Link Tra 1515	R← x8	2.5	TLP 800	Mem 010:00000	MWr(32) 1	Length 001:00:0	RequesterID 001:00:0	Tag 0	Address FEE0F00C	1st BE 1111	Last BE 0000	Data AB490000	VC ID 0	Explicit ACK Packet #9109	Metrics 2	# Packets 4.768 us	Time Delta 0013.644 619 712 s	
NVM 121	R→ x8	2.5	RequesterID 000:00:0	CQyHDBL	IO CQH QID = 3 0x0002	Time Stamp 0013.644 624 480 s												



Write Command

Write Command

Dword	Bytes	Name		
0	03:00	Common Fields	Op Code 01h	
1-3	23:04	Common Fields		
4	23:04		Metadata Pointer	
5	23:04			
6	35:32	Data buffer for write information		
9	39:36	2 Dwords each for 2 PRPs or 4 Dwords for SGL		
10	43:40	Starting LBA bits 31:00		
11	47:44	Starting LBA bits 63:32		
12	51:48	Note A	Reserved	Number of LB
13	55:52	Reserved		Note B
14	59:56	Initial LB Reference Tag		
15	63:60	Logical Block Application Mask	Logical Block Application Tag	

Write Command Fields

Dword	Bit	Description		Bit	Meaning if Set to 1
12	31	Time Limited Error Retry		29	Insert Protection Information
	30	Force Unit Access (FUA)		28	Enable checking of Guard field
	29:26	Protection Information Field		27	Enable checking of Application Tag field
13	7	Incompressible		26	Enable checking of Reference Tag field
	6	Sequential Request			
	5:4	Access Latency			
	3:0	Access Frequency			
Value	Definition		Value	Definition	
00b	None, no latency information provided		0000b	No frequency information provided	
			0001b	Typical number of r/w for this LBA range	
			0010b	Infrequent r/w for this LBA range	
			0011b	Infrequent write, frequent read	
			0100b	Frequent write, Infrequent read	
			0101b	Frequent r/w for this LBA range	
10b	Idle, longer latency acceptable		0110b	One time read	
11b	Normal, typical latency				



Write Command with Directive



Write Uncorrectable Command

Dword	Bytes	Name
0	03:00	Common Fields Op Code 04h
1-5	23:04	Common Fields
6	27:24	PRP or SGL
7	31:28	No Data Transferred
8	35:32	PRP or SGL
9	39:36	No Data Transferred
10	43:40	Starting LBA bits 31:00
11	47:44	Starting LBA bits 63:32
12	51:48	Reserved Number of LB
13	55:52	Reserved
14	59:56	Reserved
15	63:60	Reserved



**Used to mark a logical block(s) as invalid
A Read to these blocks will return Unrecoverable Read Error Status
To clear, perform a write to the logical block(s)**

NVM Express

Compare Command

Dword	Bytes	Name		
0	03:00	Common Fields		Op Code 05h
1-3	23:04	Common Fields		
4	23:04		Metadata Pointer	
5	23:04			
6	27:24	PRP or SGL		
7	31:28	Data buffer for compare information		
8	35:32	PRP or SGL		
9	39:36	Data buffer for compare information		
10	43:40	Starting LBA bits 31:00		
11	47:44	Starting LBA bits 63:32		
12	51:48	Note A	Reserved	Number of LB
13	55:52		Reserved	
14	59:56	Expected Initial LB Reference Tag		
15	63:60	Expected Logical Block Application Tag Mask	Expected Logical Block Application Tag	

Completion

If storage media and buffer are the same, including metadata if any, send good status; otherwise send Compare Failure status.

Use case

Followed by write command.
Used to update statistics
Used to synchronize processors

Write Zeros Command

Dword	Bytes	Name		
0	03:00	Common Fields		Op Code 08h
1-9	39:04	Common Fields		
10	43:40	Starting LBA bits 31:00		
11	47:44	Starting LBA bits 63:32		
12	51:48	Note A	Reserved	Number of LB
13	55:52		Reserved	
14	59:56		Initial LB Reference Tag	
15	63:60	Logical Block Application Tag Mask		Logical Block Application Tag

Data Set Management



Data Set Management Command

Data Set Management Command

Allows host to provide attributes for data usage to controller

- Read or Write

- Frequency of access

- Access size (number of LBA)

- Desired latency

Information is advisory only



Data Set Management Command

Dword	Bytes	Name	
0	03:00	Common Fields	Op Code 09h
1-5	23:04	Common Fields	
6	27:24	PRP or SGL	
7	31:28	Data buffer for send information	
8	35:32	PRP or SGL	
9	39:36	Data buffer for send information	
10	43:40	Reserved	# of Ranges
11	47:44	Reserved	
12	51:48	Reserved	
13	55:52	Reserved	
14	59:56	Reserved	
15	63:60	Reserved	

Bit 0	Optimize for Read Access
Bit 1	Optimize for Write Access
Bit 2	1 = NVM subsystem may deallocate

DSM Command – Data – Range Definition

Range	Bytes	Field
0	03:00	Context Attributes
	07:04	Length in Logical Blocks
	15:08	Starting LBA
1	19:16	Context Attributes
	23:20	Length in Logical Blocks
	31:24	Starting LBA
2	35:32	Context Attributes
	39:36	Length in Logical Blocks
	47:40	Starting LBA

255	4083:4080	Context Attributes
	4087:4084	Length in Logical Blocks
	4095:4088	Starting LBA

DSM Command – Context Attributes

Bits	Attribute	Description
31:24	Command Access Size	Number of LB expected to be transferred in a single Read or Write command
23:11	Reserved	
10	WP: Write Prepare	1 = listed range is to be written in near future
09	SW: Sequential Write Range	1 = DS should be optimized for sequential write
08	SR: Sequential Read Range	1 = DS should be optimized for sequential read
07:06	Reserved	
05:04	AL: Access Latency	00b – No latency information provided
		01b – Longer latency acceptable
		10b – Typical latency
		11b – Smallest possible latency
03:00	AF: Access Frequency	0h – No frequency information provided
		1h – Typical number of R&W expected
		2h – Infrequent R&W to this LBA range
		3h – Infrequent Writes and frequent Reads
		4h – Frequent writes and infrequent reads
		5h – Frequent R&W to this LBA range

Reservations

Reservations Notes

NVMe Reservations provide capabilities that may be utilized by two or more hosts to coordinate access to a shared namespace.

A reservation on a namespace restricts hosts access to that namespace. If a host submits a command to a namespace in the presence of a reservation and lacks sufficient rights, then the command is aborted by the controller with a status of Reservation Conflict .

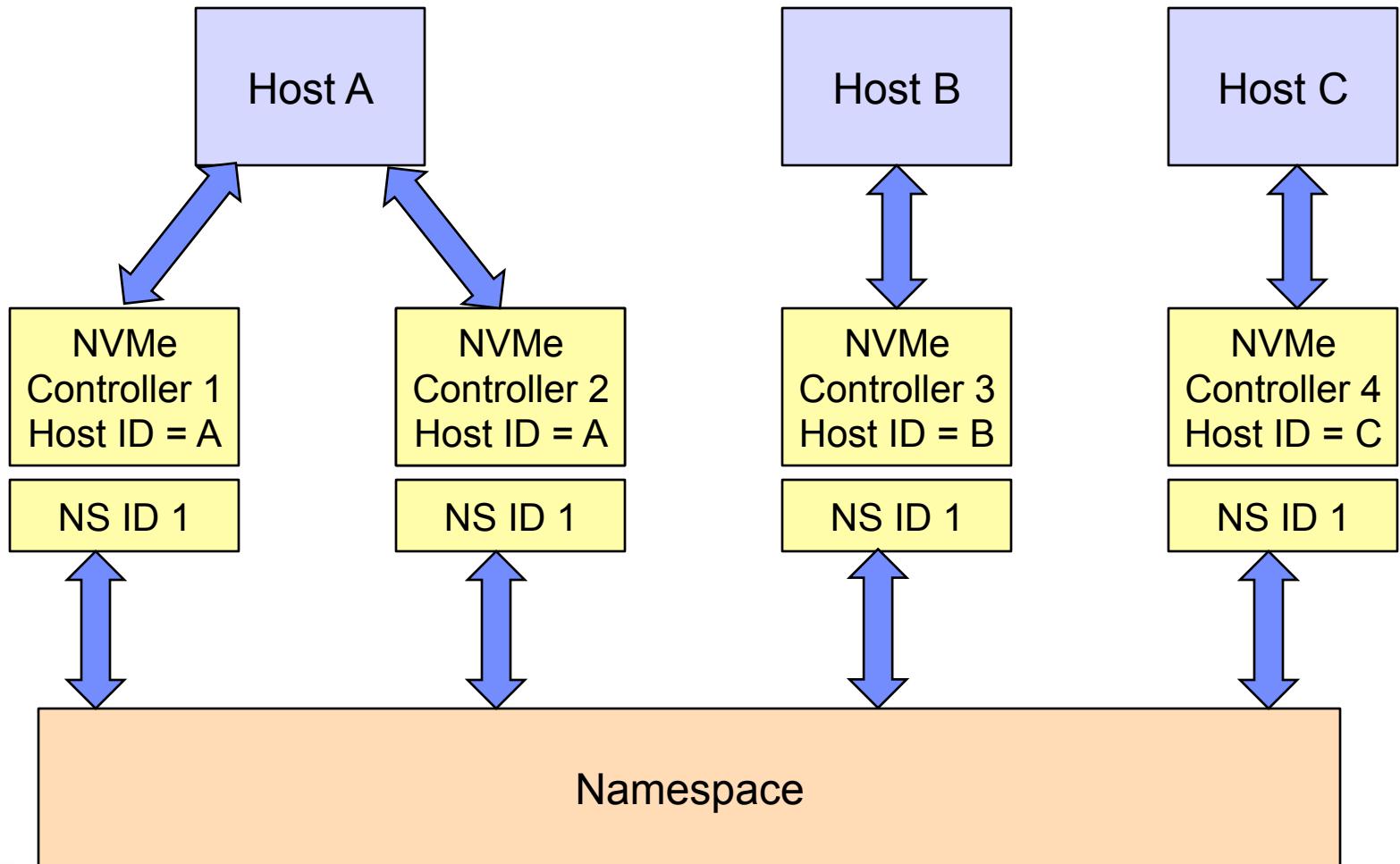
Support for Reservations is Optional

Reservations and registrations persist across all Controller Level Resets and all NVMe Subsystem Resets except power loss. Reservations may persist across power loss if Persist Through Power Loss State is set.

Controllers in a NVMe Subsystem must have the same support for reservations.

Namespaces are not required to have the same support for reservations

Reservations Example



Reservations are between Host ID and Namespace ID

Reservation Keys

Before requesting a reservation, host must register a key with the namespace.

Value of key and method of creating it is outside scope of specification

Key is to identify the host, authenticate the registrant and to preempt registrant

Multiple hosts may register with the same key value.



Reservation Support

Namespace indicates support in the Reservation Capabilities (RESCAP) field in Identify Namespace data structure

Controllers indicate support in the Optional NVM Command Support (ONCS) field (bit 5) in the Identify Controller data structure

Reservations Commands

Reservation Register

Used to register, unregister or replace a reservation key

Reservation Acquire

Used to reserve a namespace, preempt a reservation, and abort a reservation

Reservation Release

Used to release or clear a reservation

Reservation Report

Returns status that describes registration and reservation status of a namespace

Covered in this Section

NVMe Commands

Notes



Notes



Section 7

NVMe Management



Covered in this Section

NVMe Management Interface

Overview

NVMe-MI Detail

MCTP

SMB operations

NVMe Basic Management



NVMe-MI Management Interface



NVMe-MI Overview



NVMe-MI

Provides out-of-band method of:

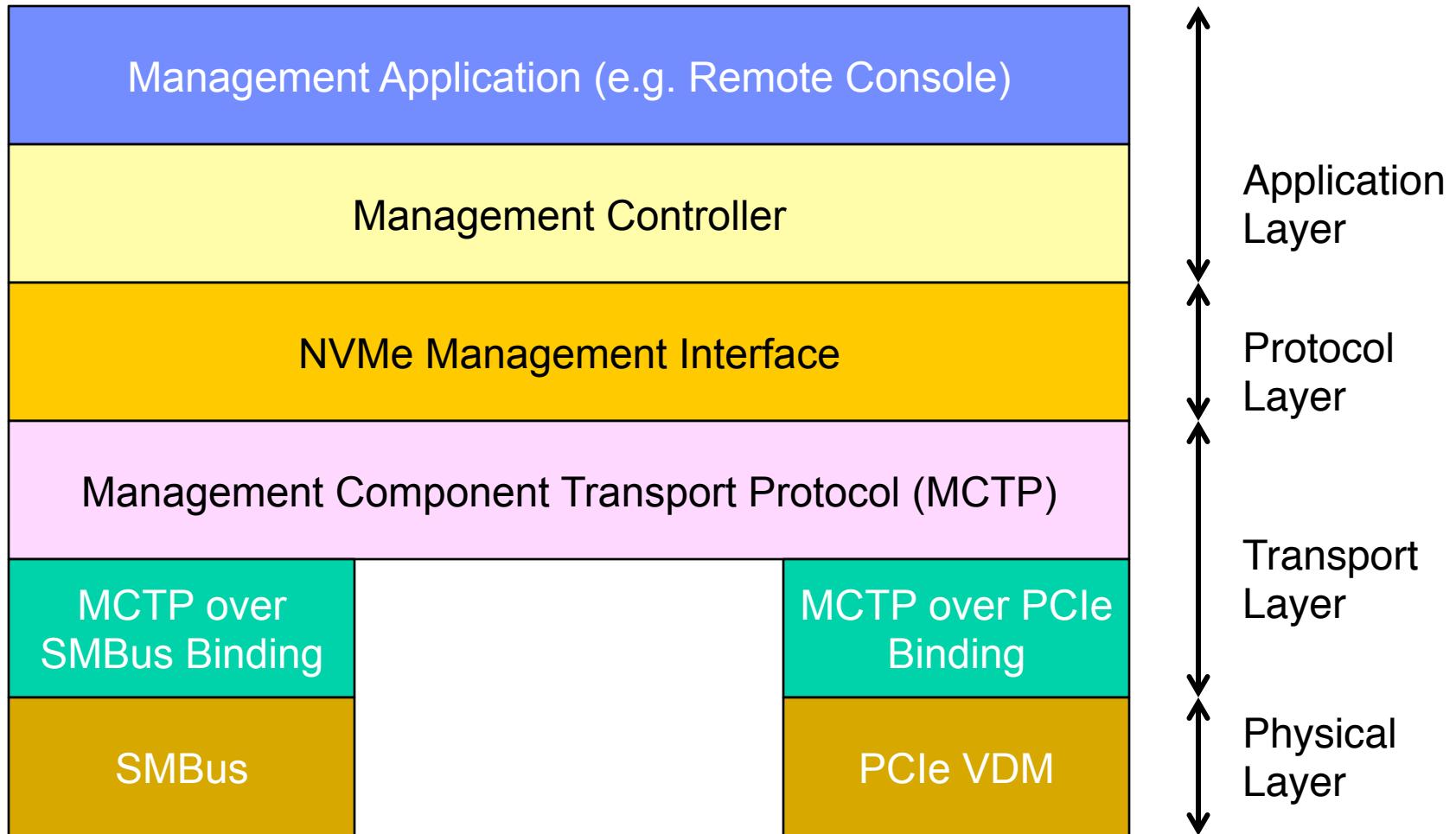
- Discovering devices and learn their capabilities
- Store data about the host environment
- Monitor health and temperature

Features of NVMe-MI:

- Multiple Command Slots to prevent long latency commands from blocking others
- Processor and OS agnostic
- Preserves data at rest security
- Provides a standard format for VPD



NVMe MI Protocol Layering



Defining Documents

Available from www.dmft.org.

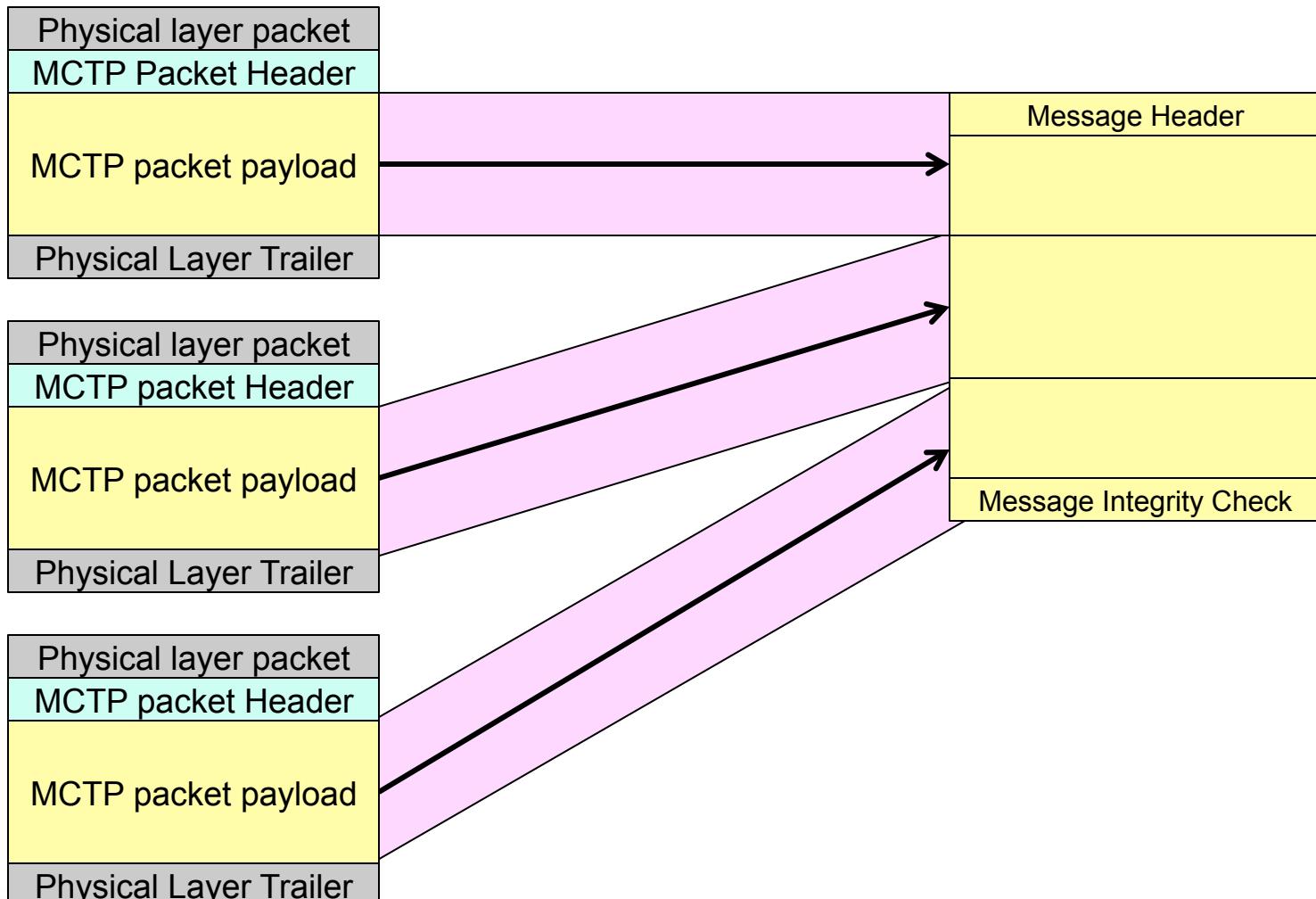
- DSP0235 – NVMe Management over MTP Binding Specification
- DSP0236 – MCTP Base Specification
- DSP0237 – MCTP SMBus Transport Binding
- DSP0238 – MCTP PICe VDM Transport Binding
- DSP0239 – MCTP IDs and Codes

Available from www.nvmeexpress.org.

- NVMe 1.2.1 – Base NVMe specification
- NVMe Management Interface 1.0 Gold – Management Interface



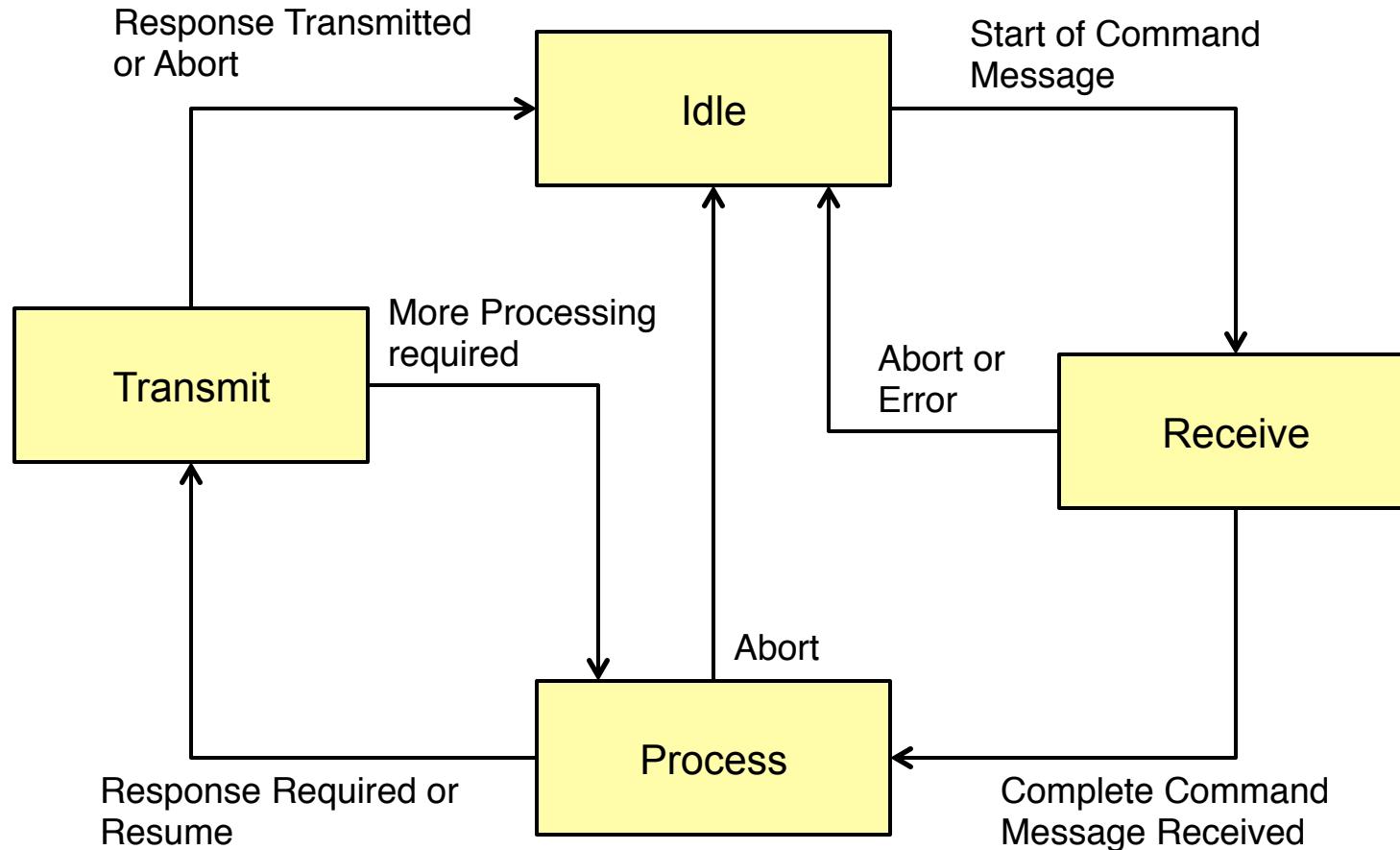
NVMe-MI using Multiple Packets for a Message



NVMe-MI Detail



Command Slot Diagram



Command Slot Notes

Similar to Logical Units

NVMe-MI supports exactly two command slots

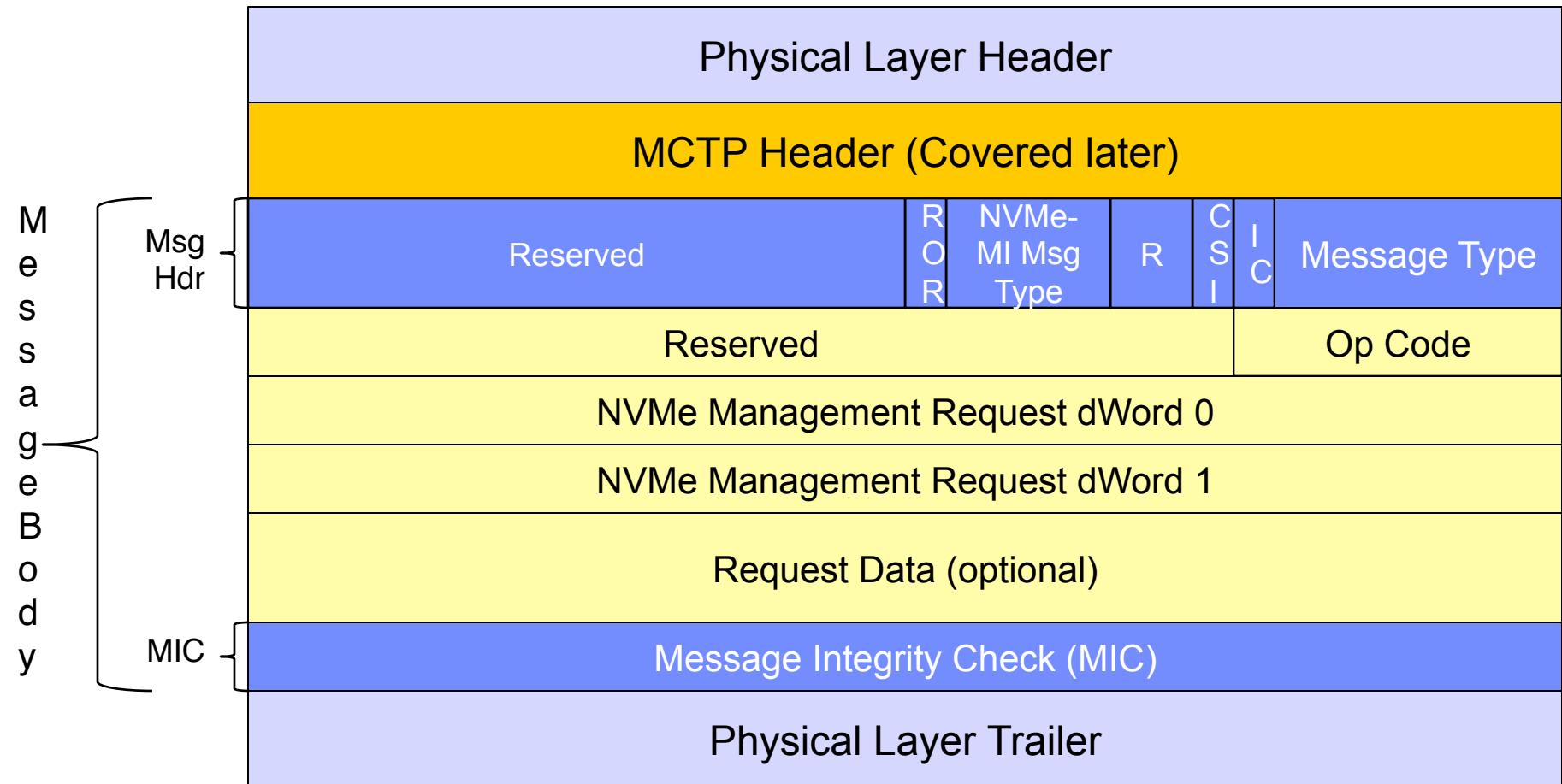
Command Slots are designed to handle one command at a time

If a 2nd command is received while the 1st one is in receive state,
then the 1st CMD is discarded and the 2nd CMD is processed

If a 2nd command is received while the 1st one is in process or transmit state,
then the 2nd CMD is discarded without response



NVMe-MI Command



NVMe-MI Request

Message Type – set to 4h for all NVMe-MI messages

IC – Integrity Check, shall be set to 1b in NVMe-MI

CSI – Command Slot Identifier

NVMe-MI Message Type

0h – Control Primitive

1h – NVMe-MI Command

2h – NVMe Admin Command

4h – PCIe Command

ROR – Request or Response, 1b = Response

NVMe Management Request dWord 0/1 – Command Specific

Data – Maximum for message is 4224 bytes (4K + 128)

MIC – 32-bit CRC computed over the message body contents

Control Primitive

Reserved	R O R	NVMe- MI Msg Type	R	C S I	I C	Message Type
Control Primitive Specific Parameter	Tag			Control Primitive Op Code		
Message Integrity Check (MIC)						

Control Primitive Op Code	Command	Parameter
00h	Pause	Response – Pause status of cmd slots 0/1
01h	Resume	N/A
02h	Abort	Response – Cmd completed, not started, partially completed
03h	Get State	Request – 1 = Clear error state flags Response – see Reference Manual or NVMe-MI Specification
04h	Replay	Request – bits 7:0 – Response Replay offset (which packet) Response – bit 0 – 1 = Requested Response retransmitted
F0-FFh	Vendor Specific	Vendor Specific



Tag – Returned in Response to match Request and Response

NVMe-MI Commands

Op Code	Command
00h	Read NVMe-MI Data Structure
01h	NVM Subsystem Health Status Poll
02h	Controller Health Status Poll
03h	Configuration Set
04h	Configuration Get
05h	VPD Read
06h	VPD Write
07h	Reset
C0 – FFh	Vendor Specific

NVMe Admin Commands for NVMe-MI

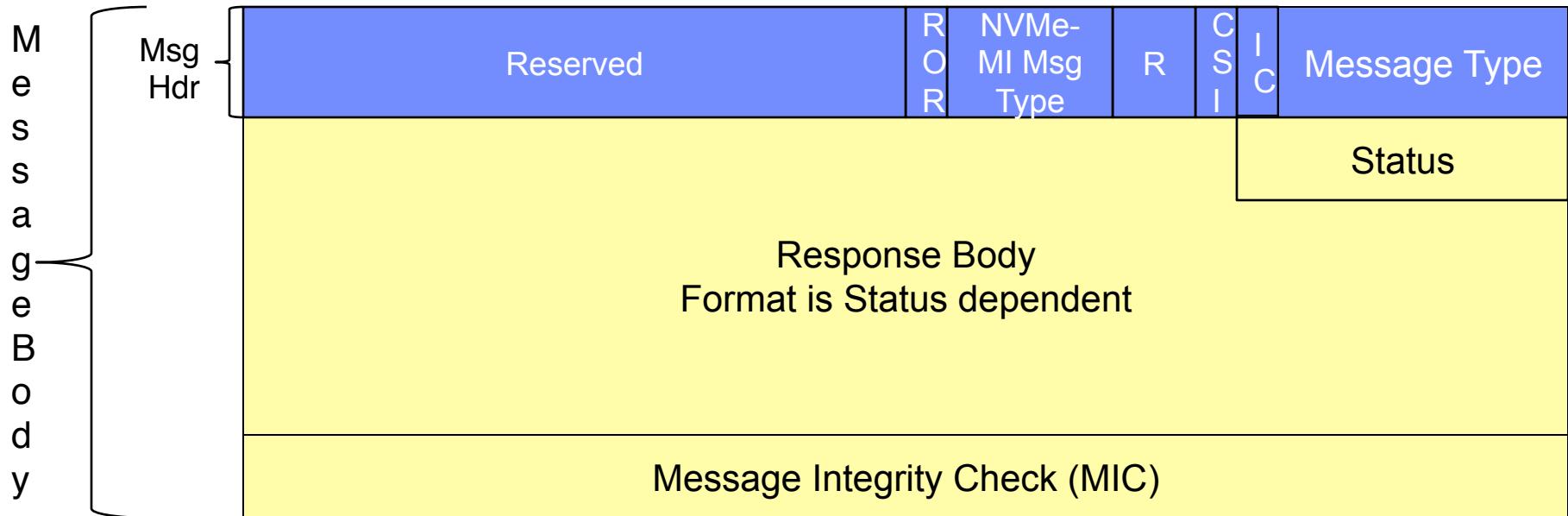
Op Code	Command	M/O
10h	Firmware Activate/Commit	O
11h	Firmware Image Download	O
80h	Format NVM	O
0Ah	Get Features	M
02h	Get Log Page	M
06h	Identify	M
0Dh	Namespace Management	O
15h	Namespace Attachment	O
81h	Security Send	O
82h	Security Receive	O
09h	Set Features	O
C0-FFh	Vendor Specific	O

PCIe Commands for NVMe-MI

Fmt/Type	Command	M/O
04 or 05h	PCIe Configuration Read	M
44 or 45h	PCIe Configuration Write	M
00 or 20h	PCIe Memory Read	M
40 or 60h	PCIe Memory Write	M
02h	PCIe I/O Read	M
42h	PCIe I/O Write	M

Not covered in this course; listed for completeness

Response Message Format



Message Type – set to 4h for all NVMe-MI messages

IC – Integrity Check, shall be set to 1b in NVMe-MI

CSI – Command Slot Identifier

NVMe-MI Message Type

- 0h – Control Primitive

- 1h – NVMe-MI Command

- 2h – NVMe Admin Command

- 4h – PCIe Command

ROR – Request or Response, 1b = Response

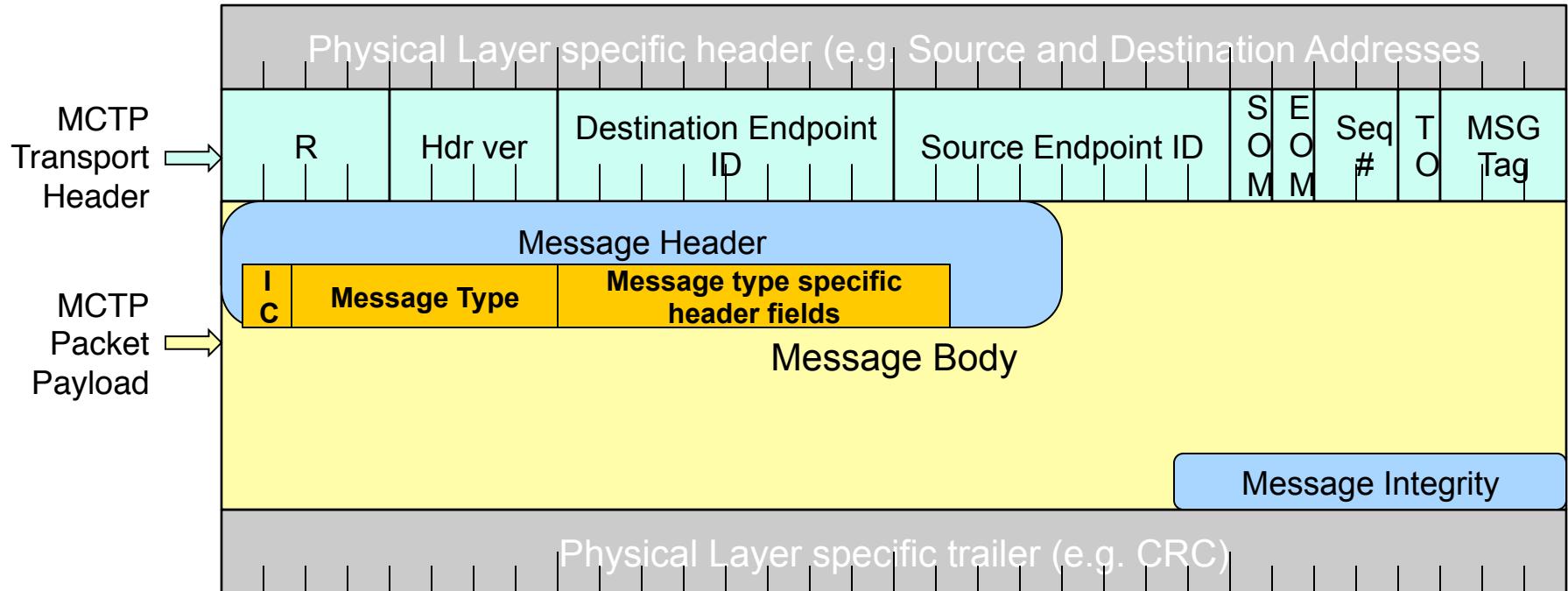
NVMe-MI Status Codes

Value	Meaning
00h	Success
01h	More Processing Required
02h	Internal Error
03h	Invalid Command Opcode
04h	Invalid Parameter
05h	Invalid Command Size
06h	Invalid Command Input Data Size
07h	Access Denied (vendor specific protection)
20h	VPD Updates Exceeded
21h	PCIe Inaccessible
E0 – FFh	Vendor Specific

NVMe-MI MCTP



MCTP Packet Fields



MCTP Header Codes

SOM – Start of Message, 1st packet of this message

EOM – End of Message, Last packet of this message

Packet Sequence Number – Sequential number of this packet in this message

Increments modulo 4

TO – Tag Owner, identifies if this message tag was originated by the endpoint that is the source of the message or by the destination. 1b indicates that te source of the message originated the message tag.

Msg Tag – with Source Endp0int ID and TO identifies a unique message at the MCTP transport level.

Message type specific header fields – 0 or more bytes as specified by the message type

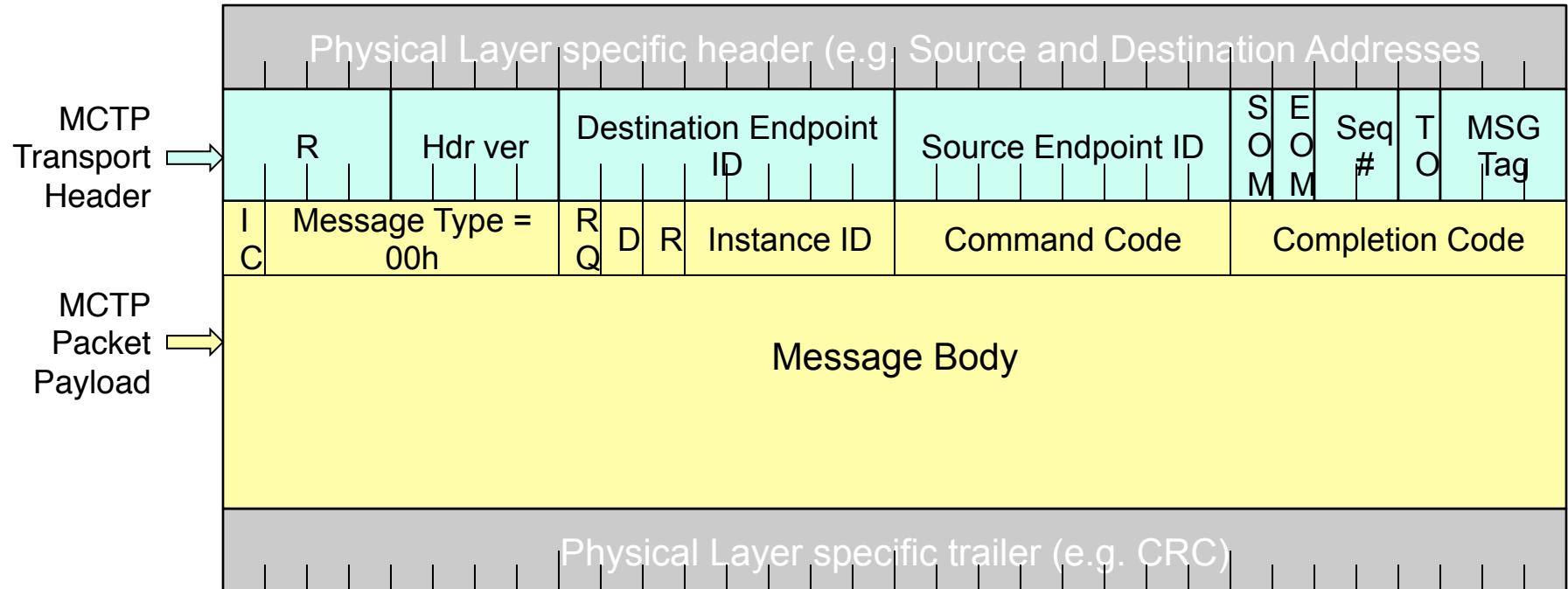
Normally only in the first packet of multiple packet messages

Message body – 0 or more bytes as specified by the message type

Message Integrity – length is message type specific

Normally only in the last packet of multiple packet messages

MCTP Control Message Format



IC – Integrity Check, 1b

RQ – 1 = Request

D – Datagram

Instance ID – Differentiate between new and retried messages

Match responses to requests

Command Code – used on both requests and responses

Completion Code – present only on responses



MCTP Control Commands

Command Code	Command
00h	Reserved
01h	Set Endpoint ID
02h	Get Endpoint ID
03h	Get Endpoint UUID
04h	Get MCTP Version Support
05h	Get Message Type Support
06h	Get Vendor Defined Message (VDM) Support
07h	Resolve Endpoint ID
08h	Allocate Endpoint ID
09h	Routing Information Update
0Ah	Get Routing Table Entries
0Bh	Prepare for Endpoint Discovery
0Ch	Endpoint Discovery
0Dh	Discovery Notify
0Eh	Get Network ID
0Fh	Query Hop
10h	Resolve UUID
F0-FFh	Transport Specific



MCTP Control Completion Codes

Completion Code	Meaning
00h	Success
01h	Error (generic)
02h	Error Invalid Data
03h	Error Invalid Length
04h	Error – Not Ready
05h	Error – Unsupported Command
06h	Get Vendor Defined Message (VDM) Support
80-FFh	Command Specific

Completion Code	Meaning	Command
80h	Message type number not supported	Get MCTP Version Support
80h	Insufficient Space to add requested entries	Routing Information Update

SMB

System Management Bus



SMB Notes

Purpose of SMB

Polling status and health without loading the data path (out of band)

General Notes of SMB

SMB is a two wire bus: Clock and Data

Clock is controlled by the Master, but the Slave can extend it

Bus is pulled by resistor or current source to the highest V_{DD}

V_{DD} can be between 1.8 and 5 Volts $\pm 10\%$

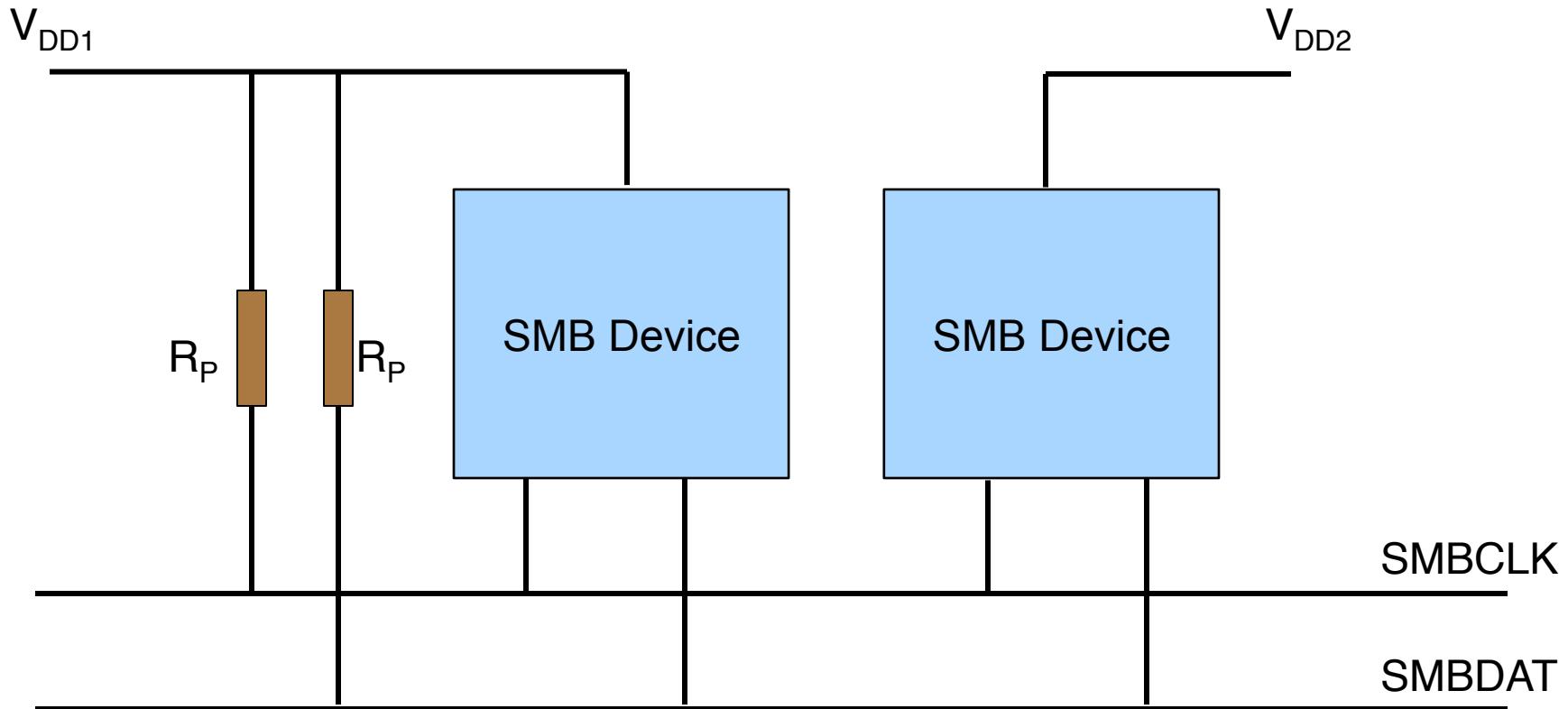
Multiple devices can drive the bus to 0V making it a Logical And

Each byte is ACKed or NACKed by the recipient

Clock rates defined are 100Kb/s, 400Kb/s, and 1Mb/s



SMB Topology Visual



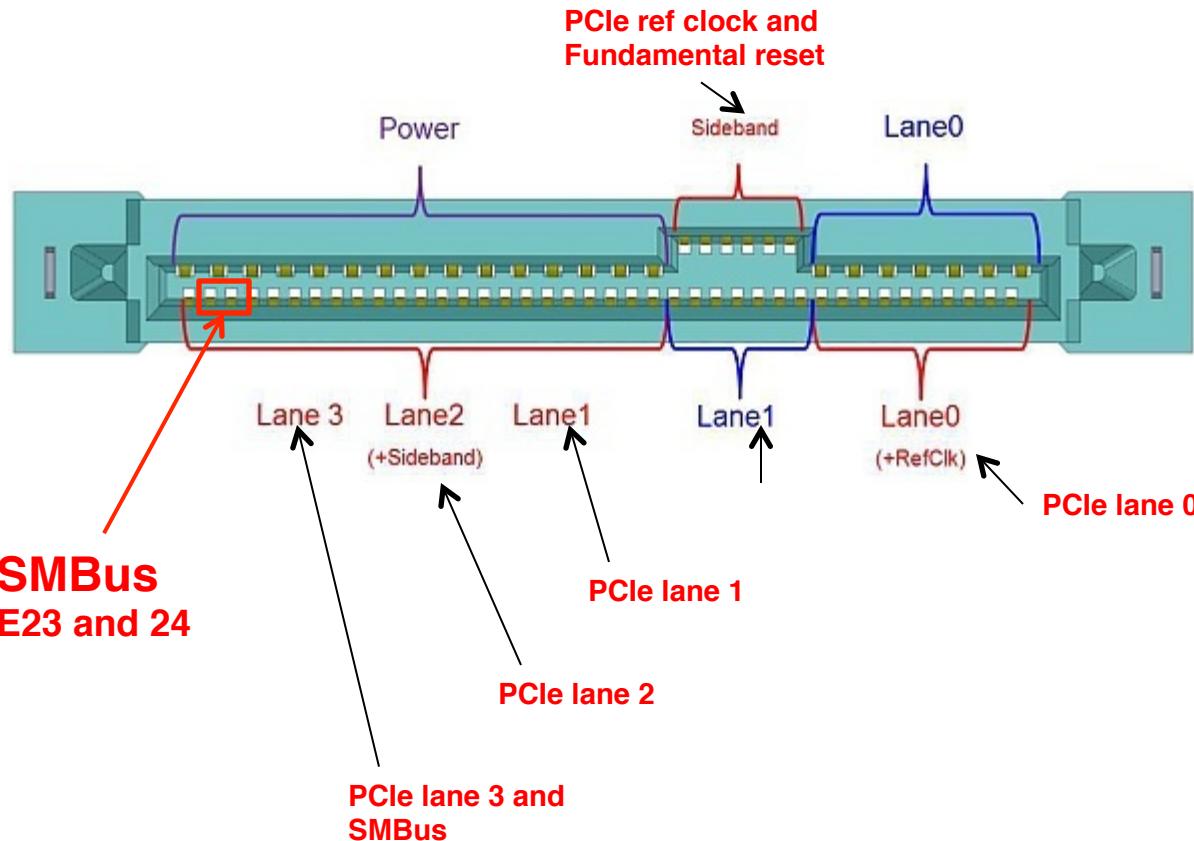
R_P may be a resistor or current source

SMB Physical Connections – Card Edge

Pin		Side B		Side A
1	P	+12 V	S	Prsnt1#
2	P	+12 V	P	+12 V
3	P	+12 V	P	+12 V
4	G	Ground	G	Ground
5	D	SMCKL	I	TCK
6	D	SMDAT	I	TDI
7	G	Ground	O	TDO
8	P	+ 3.3 V	I	TMS
9	I	TRST#	P	+ 3.3 V
10	P	+ 3.3 V Aux	P	+ 3.3 V
11	O	Wake#	I	Perst#
Key Notch				
12	R	Reserved	G	Ground
13	G	Ground	I	RefClk+
14	I	HSoP(0)	I	RefClk-
15	I	HSON(0)	G	Ground
16	G	Ground	O	HSIP(0)
17	S	Prsnt2#	O	HSIN(0)
18	G	Ground	G	Ground
PCI X1 boards end at pin 18				

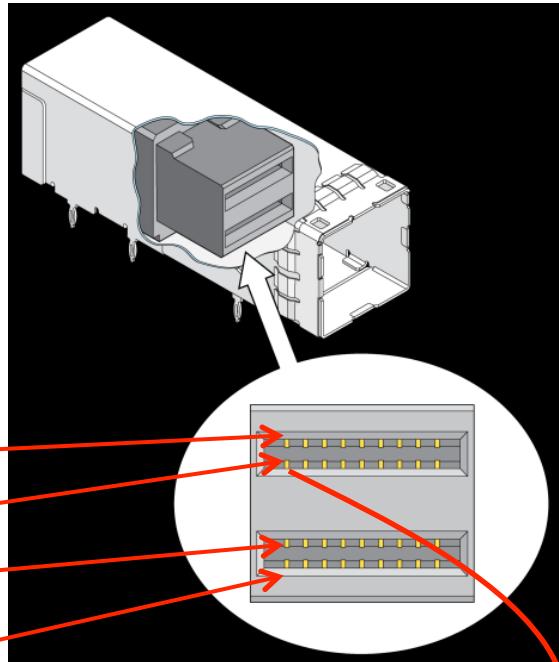
**SMBus
Clock and
Data**

SMB Physical Connections – SFF-8639



SMB Physical Connections – SFF-8644

SMB name changed to CMI
(Cable Management Interface)

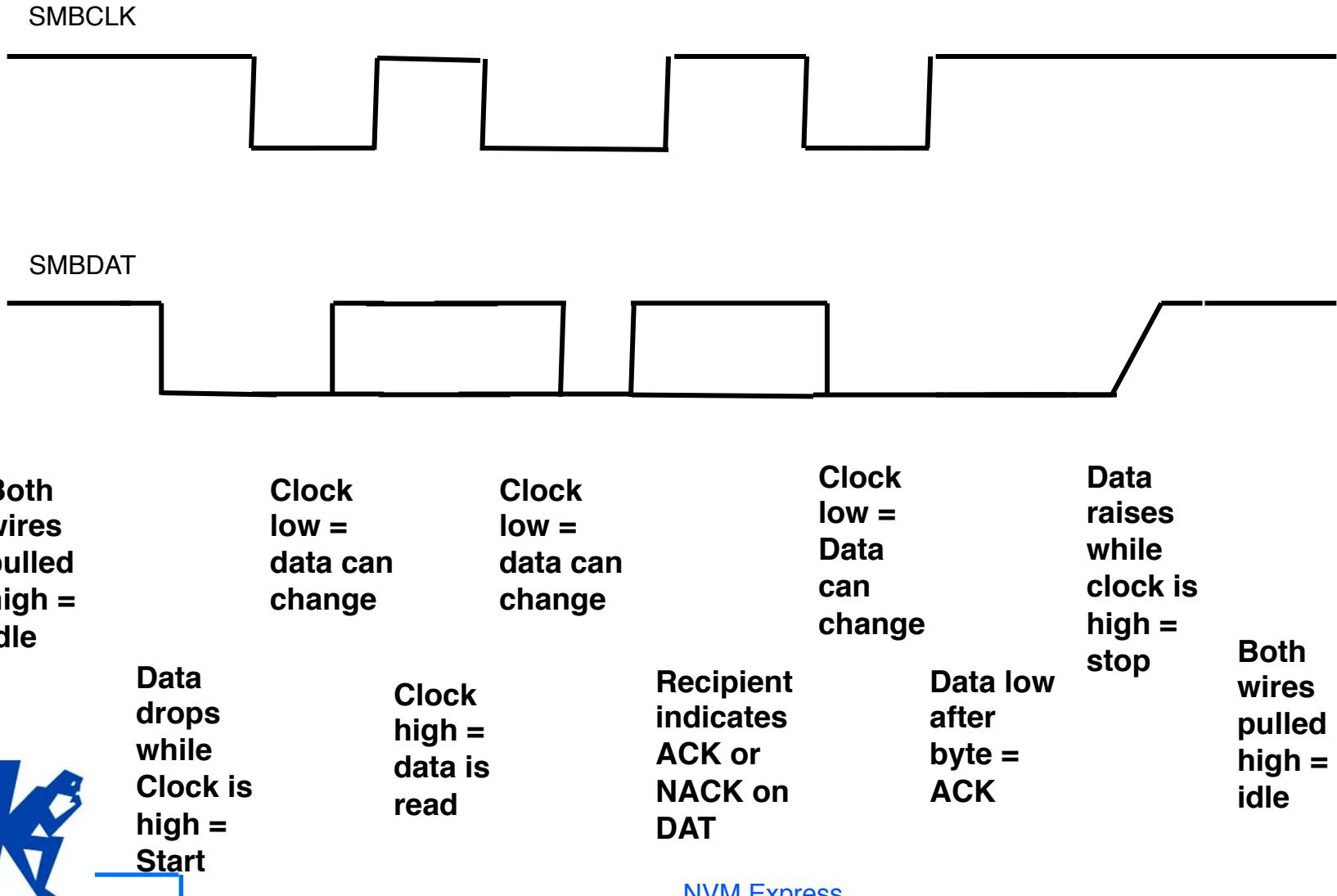


Row\Col	9	8	7	6	5	4	3	2	1
D	GND	PETn2	PETp2	GND	PETn1	PETp1	GND	MGTPWRR	Pwr
C	GND	PETn3	PETp3	GND	PETn0	PETp0	GND	CMIDAT	CMICLK
B	GND	PERn2	PERp2	GND	PERn1	PERp1	GND	CBLPRSNT#	Pwr
A	GND	PERn3	PERP3	GND	PERn0	PERp0	GND	CINT#	CAaddr

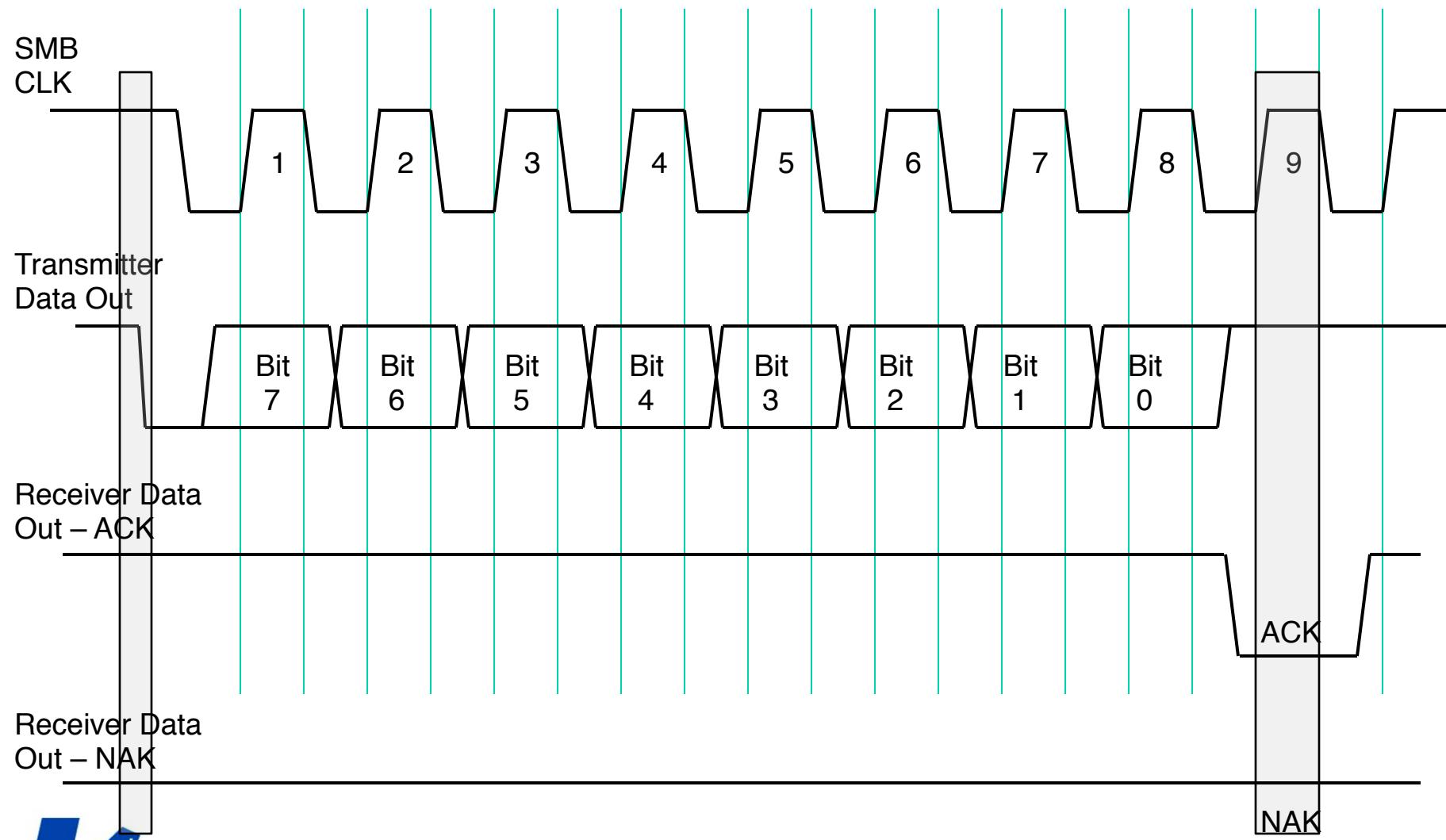


SMB Signaling

Start – One bit - ACK - Stop



SMB Signaling – ACK/NAK Signaling



Device Types

Master Devices

- Issues commands, generates clocks, terminates the transfer.
- May be a transmitter or receiver
- There may be more than 1 master on a bus

Slave Devices

- Responds to its address; receives commands
- May be a transmitter or receiver
- May be designed to be come Master (e.g. when using Host Notify)

Host Devices

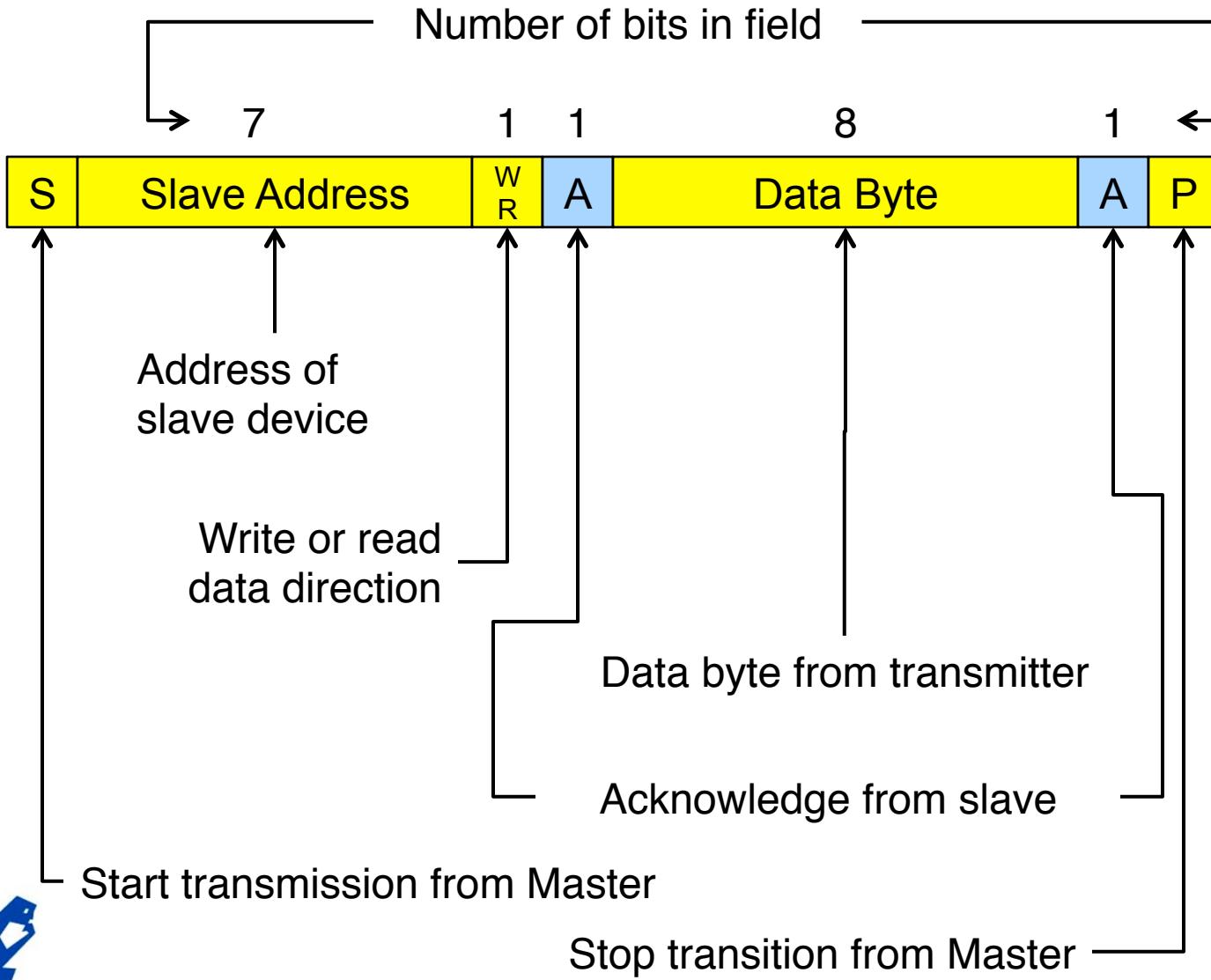
- Specialized Master, provides main interface to System's CPU or mgmt
- Must be able to function as Master and Slave
- Must support Host Notify
- There can be 0 or 1 hosts on a bus



Command Protocols



Generic Transaction Diagram



Yellow fields (white in Specification) indicate from Master.
Blue fields (shaded in Specification) indicate from Slave.



Send Byte Command Protocol



Data Byte may be a command.

Example:

Bits 7:1 define a feature

Bit 0 indicates to turn it on or off



Block Read Command Protocol



Sr = Repeated Start

Optional
Not included in Block Count

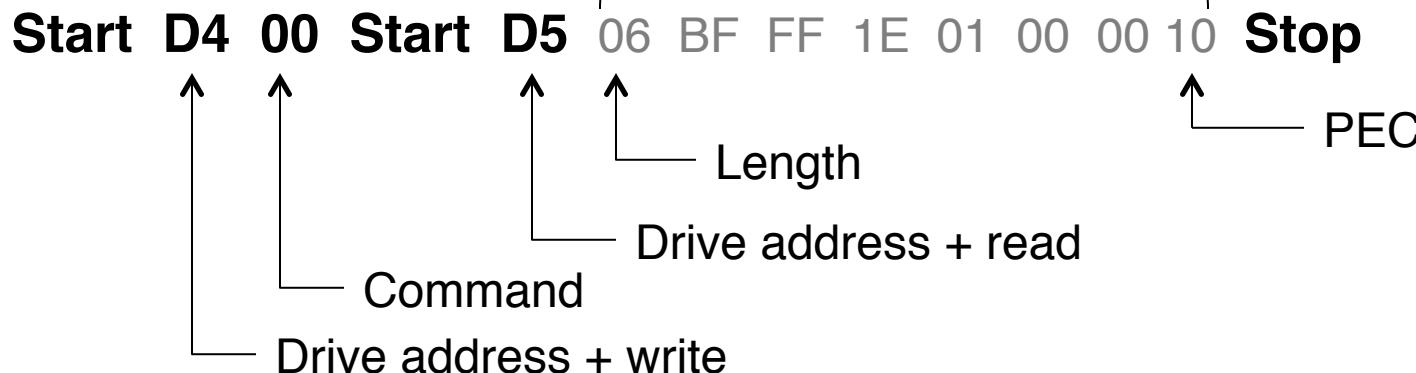
NACK from Master
signifies end of read
transfer

NVMe Basic Management



Block Read Command Examples

Read Drive Status



Read Static Data

Start D4 08 Start D5 16 12 34 "A" "Z" "1" "2" "3" "4" "5" "6" 20 20 20 20
20 20 20 20 20 20 20 DA **Stop**

Send Byte to reset Arbitration bit

Start D4 FF Stop



NVMe Defined Commands

CMD Code	Offset	Description
00	00	Length of Status (always 6) (1's based)
	01	Bit 7 – SMBus Arbitration
		Bit 6 – Powered UP
		Bit 5 – Drive Functional
		Bit 4 – Reset Not Required to resume ops
		Bit 3 – PCIe Link Active
		Bits 2-0 – Reserved, Set to 1b
	02	Smart Warnings (Byte 0 of NVMe SMART Info Log)
	03	Composite Temperature 00 – 7E – Temp in Celsius (0-126C) 7F – Temp 127C or higher 80 – No temp data or data is > 5 seconds old 81 – Temp sensor failure 82-C3 – Reserved C4 – Temp 60C or lower C5-FF – Temp in Celsius (-1 to -59C)
		04 Percentage Drive Life Used (Percentage Used from SMART Log)
	06:05	Reserved: set to 0000h
	07	PEC



NVMe Defined Commands

CMD Code	Offset	Description
08	08	Length of Return Data (always 22) (1's based)
	10:09	Vendor ID: Assigned by PCI SIG
	30:11	Serial Number: From NVMe Identify Controller
	31	PEC
32+	255:32	Vendor Specific

Length field is the number of bytes between the length field and PEC, non-inclusive

Covered in this Section

NVMe Management Interface

Overview

NVMe-MI Detail

MCTP

SMB operations

NVMe Basic Management



Notes



Section 8

NVMe 1.3



Covered in this Section

New Admin Commands

- Directive Send and Receive
- Device Self-Test
- NVMe-MI Send and Receive
- Doorbell Buffer Configuration
- Virtualization Management
- Sanitize

New Features

- Time Stamp
- Host Controller Thermal Management
- Telemetry
- Non-Operational Power State Configurations

New Registers

- Boot Partition Information
- Boot Partition Read Select
- Boot Partition Buffer Location



New Admin Commands



Op Code for Admin Commands

Op Code	Command
00h	Delete I/O Submission Queue
01h	Create I/O Submission Queue
02h	Get Log Page
04h	Delete I/O Completion Queue
05h	Create I/O Completion Queue
06h	Identify
08h	Abort
09h	Set Features
0Ah	Get Features
0Ch	Asynchronous Event Request
0Dh	Namespace Management

Op Code	Command
10h	Firmware Commit/Firmware Activate
11h	Firmware Image Download
14h	Device Self-Test
15h	Namespace attachment
18h	Keep Alive
19h	Directive Send
1Ah	Directive Receive
1Ch	Virtualization Management
1Dh	NVMe-MI Send
1Eh	NVMe-MI Receive
7Ch	Doorbell Buffer Config
7Fh	Fabrics Commands
84h	Sanitize
C0 – FFh	Vendor specific

ADMIN and NVMe Command Format

Dword	Bytes	31	Name	0
0	03:00		Command ID (CID)	P S
1	07:04		Namespace Identifier (NSID)	Res
2	11:08		Reserved	F
3	15:12			OP Code
4	19:16		Metadata Pointer (MPTR) – Address of physical buffer for metadata	
5	23:20			
6	27:24		PRP Entry 1 (PRP1)	
7	31:28			or SGL 1
8	35:32		PRP Entry 2 (PRP2)	
9	39:36			
10	43:40			
11	47:44			
12	51:48		Command specific fields	
13	55:52			
14	59:56			
15	63:60			



Directive Send and Receive

Dword	Bytes	31	Name	0	
0	03:00	Command ID (CID)	P S	Res	
1-5	23:04	Common Fields			
6	27:24	PRP Entry 1 (PRP1)			
7	31:28				
8	35:32	PRP Entry 2 (PRP2)			
9	39:36				
10	43:40	Number of DWords			
11	47:44	Directive Specific	Directive Type	Directive Op	
12	51:48	Res	Directive Type	Res	
13-15	63:52	Reserved			

E - Enable

Directive Type	Definition
00h	Identify
01h	Streams

Directive Notes

Used to expand the command set for more complex commands or features

Directives may also appear in I/O commands

Directives are defined in Section 9 of NVMe 1.3

Support for Directives is indicated in Optional Admin Command Support field in Identify Controller data structure.

If the controller supports directives, the Identify Directive is mandatory

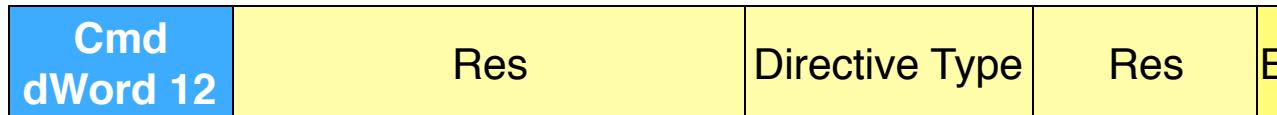
On NVMe Subsystem reset, all Directives (except Identify) are disabled

Directives other than Identify may be enabled or disabled by the host

Identify Directive

Used to identify which Directives controller supports and to enable or disable used of supported directives

Directive Command	Directive Type	Directive Op	Directive Op Name
Directive Receive	00h	01h	Return parameters
Directive Send	00h	01h	Enable Directive



Byte 63
Bit 7

Byte 32
Bit 0

Byte 0
bit 0

Directives Enabled

Directives Supported

Bytes 4095 – 63 are reserved

Streams Directive

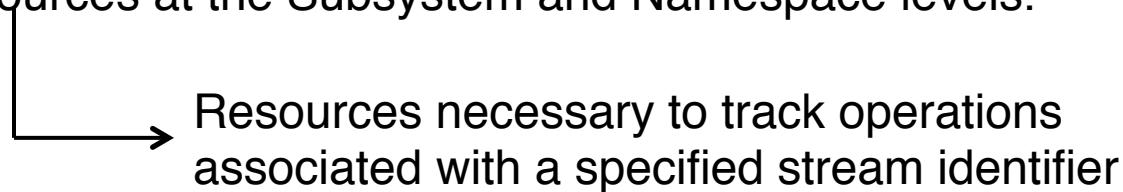
Enables the host to indicate by stream identifier to the controller that the specified logical blocks in a write command are part of one group of associated data.

Directive Command	Directive Type	Directive Op	Directive Op Name
Directive Receive	01h	01h	Return parameters
		02h	Get Status
		03h	Allocate Resources
Directive Send	01h	01h	Release Identifier
		02h	Release Resources

Streams Directive Notes

Return parameters identifies:

Stream Write Size (SWS) (optimal write size),
Stream Granularity Size (SGS) (media that is prepared),
Stream resources at the Subsystem and Namespace levels.



Resources necessary to track operations
associated with a specified stream identifier

Steam Identifiers are assigned by the host and are:
from 1h to FFFFh inclusive.



Streams Directive Return Parameters

Byte	Description
NVM Subsystem Specific Fields	
1:0	Max Streams Limit
3:2	NVM Subsystem Streams Available
5:4	NVM Subsystem Streams Open
15:6	Reserved
Namespace Specific Fields	
19:16	Stream Write Size (SWS) (# of Logical Blocks)
21:20	Stream Granularity Size (SGS) (# of SWS)
23:22	Namespace Streams Allocated
25:24	Namespace Streams Open
31:26	Reserved

Streams Directive Get Status

Byte	Description
1:0	Open Stream Count
3:2	Stream Identifier 1
5:4	Stream Identifier 2
131071:131070	Stream Identifier 65535



Contains the value of the open stream, starting with the lowest numerical value

Host uses Get Status to determine which streams are open

If NSID was FFFFFFFF, then return open streams for the subsystem not associated with a Namespace



Streams Directive Allocate Resources

DWord	Description	
Directive Receive Command		
CMD Dword 12	Reserved	NS Streams Requested
Directive Receive Completion		
Comp Dword 0	Reserved	NS Streams Allocated

Streams Directive Release Identifier

DWord	Description		
Directive Receive Command			
CMD Dword 11	Steam Identifier	DTYPE	Op Code

Release only the specified Stream

Streams Directive Release Resources

Release all streams resources allocated for the namespace attached to all controllers associated with the same Host Identifier that processed this command.

Device Self-Test

Causes the controller to start or abort a self-test operation of the device and namespace specified

When Device Self-test is in progress,
Some commands may process concurrently,
Other commands may be suspended.

Short test should complete in two minutes or less

Aborts on Controller Level Reset, Format NVM, Device Self-test abort

Extended test should complete in time indicated in Identify Controller

Aborts on Format NVM, Device Self-test abort

Persists on Controller Level Reset

Resumes after completion of Reset or restoration of power



Device Self-test Command Fields

Namespace field:

0h – Test only the controller, no namespaces

1 – FFFFFFFEh – Specifies the namespace to test

FFFFFFFFFFh – Test shall include all active namespaces

DWord	Description
CMD Dword 10	Reserved



STC	Description
1h	Start a short device self-test operation
2h	Start an extended device self-test operation
Eh	Vendor Specific
Fh	Abort device self-test operation

Device Self-test Completion



Command Specific Status:

1Dh – Device Self-test in Progress

NVM Express

NVMe-MI Send and Receive

Provides a means to send and receive Out-of-Band management functions through in-band commands and status

Refer to Out-of-Band management section of this course or the NVMe-MI specification for format and servicing of NVMe-MI Request and Response messages

NVMe-MI Send Command Format

Dword	Bytes	31	Name	0
0	03:00	Command ID (CID)	P S	Res F 1Dh
1-5	23:04	Common Fields		
6	27:24	PRP 1		
7	31:28	Points to data buffer with NVMe-MI Request Message		
8	35:32	PRP 2		
9	39:36			
10	43:40	Reserved		NVMe-MI Specific
11	47:44	Reserved		
12-15	63:48	Reserved		

Dword	Name
0	Number of Dwords returned
1	Reserved
2	SQ Identifier
3	SQ Head Pointer
	Status Field P Command Identifier
	NVM Express

NVMe-MI Receive Command Format

Dword	Bytes	31	Name	0
0	03:00	Command ID (CID)	P S	Res F 1Eh
1-5	23:04	Common Fields		
6	27:24	PRP 1		
7	31:28	Points to data buffer with NVMe-MI Request Message		
8	35:32	PRP 2		
9	39:36			
10	43:40	Reserved	NVMe-MI Specific	
11	47:44	Number of Dwords in data structure (0's based)		
12-15	63:48	Reserved		

Doorbell Buffer Config

Intended for Emulated Controllers

Provides for memory buffers that mirror the physical controllers doorbell registers

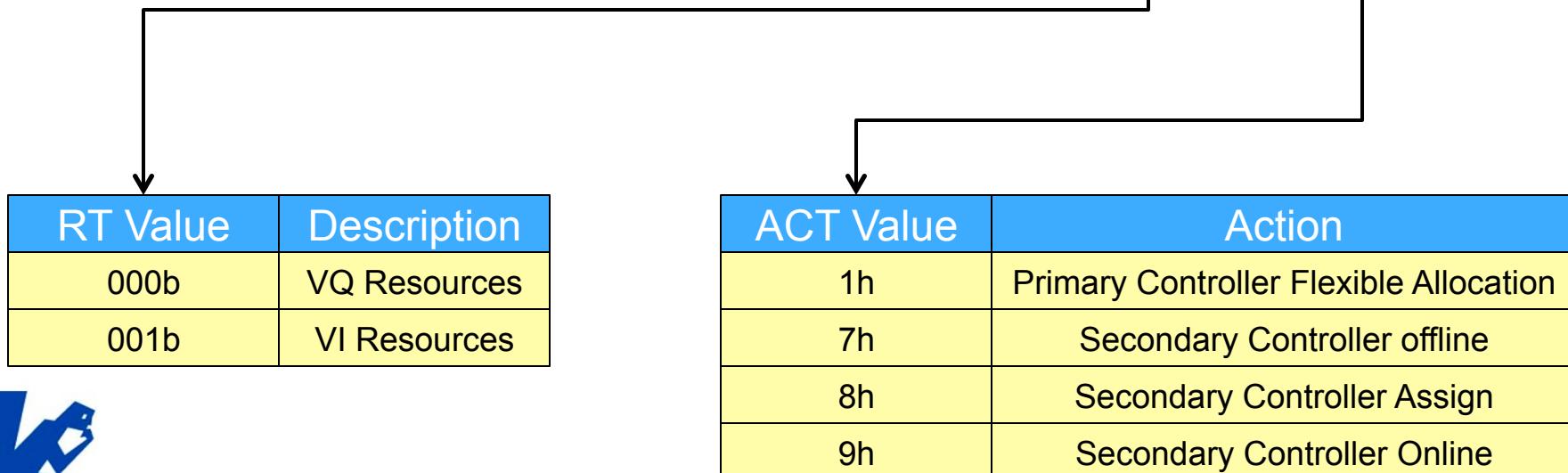
Dword	Bytes	31	Name	0
0	03:00	Command ID (CID)	P S	Res F 7Ch
1-5	23:04	Common Fields		
6	27:24	PRP 1 Points to Shadow Doorbell buffer		
7	31:28	PRP 2 Points to EventIdx buffer		
8	35:32			
9	39:36			
10-15	63:40	Reserved		

After configured, virtual host will write to shadow doorbell registers

If the update causes the shadow doorbell register to a value greater than the value of the EventIdx, then the controllers doorbell register shall also be updated to match the value of that entry in the Shadow doorbell.

Virtualization Management

Dword	Bytes	31	Name	0	
0	03:00	Command ID (CID)	P S	Res F 1Ch	
1-5	23:04	Common Fields			
6-9	39:24	PRP			
10	43:40	Controller ID	Res	RT Res ACT	
11	47:44	Reserved	Number of Controller Resources to allocate or assign		
12-15	63:48	Reserved			



Virtualization Management definitions

Primary Controller – A controller that supports Virtualization Management.
(e.g. a PCIe Physical Function that supports Virtualization Enhancements)

Secondary Controller - A controller that depends on a primary controller for management of some controller resources (e.g. an SR-IOV virtual function)

VQ Resources – A resource that manages one SQ and one CQ
A primary controller must have at least 2 queues (1 admin, 1 I/O)
If a secondary controller has no VQ Resources it remains offline

VI Resources – A resource that manages one interrupt vector
VI Resource identifier is its interrupt vector number
MSI-X is the only supported VI Resource (NVMe 1.3)

Flexible Resources – Controller resources that may be assigned to the primary controller or one of its secondary controllers

Private Resources – permanently assigned to a primary controller or secondary controller. Not supported by the Virtualization Mgmt Cmd



Sanitize

Makes all user data in the NVM subsystem in-accessible

Includes:

- non-volatile media,
- volatile media,
- cache,
- Controller Memory Buffer,
- unallocated or deallocated areas of the media,
- log page data that could aid in retrieving user data.

Includes any interface (NVMe, NVMe-MI, maintenance)

Does not include:

- Boot partitions,
- Replay Protected Memory Block,
- media or caches that do not contain user data

Methods defined to make data inaccessible:

- Block erase
- Overwrite (with or without invert)
- Crypto Erase



Sanitize

Once started Sanitize:

Continues until complete ever across resets and power cycles

I/O commands fail with Sanitize in Progress status

Certain Admin commands can execute concurrent with Sanitize

Abort

Asynchronous Event Request

Create/Delete I/O queues

Set/Get Features

Get Log Page

Identify

Keep Alive

Op Code 7Fh

Vendor Specific commands that do not affect or retrieve user data

Sanitize

All Sanitize commands execute in the background

Command complete status does not indicate completion of the Sanitize
(called Immediate in SCSI)

End of Sanitize operation is indicated by Sanitize Log Page and
Sanitize Operation Completed Asynchronous Event (if outstanding).



New Features

Timestamp Notes

Controller indicates support for Timestamp in Identify Controller

Host writes timestamp value in the controller with Set Features command

Host reads timestamp value with Get Features

Use of Timestamp is beyond the scope of the Specification

Controller may stop counting during non-operational power states

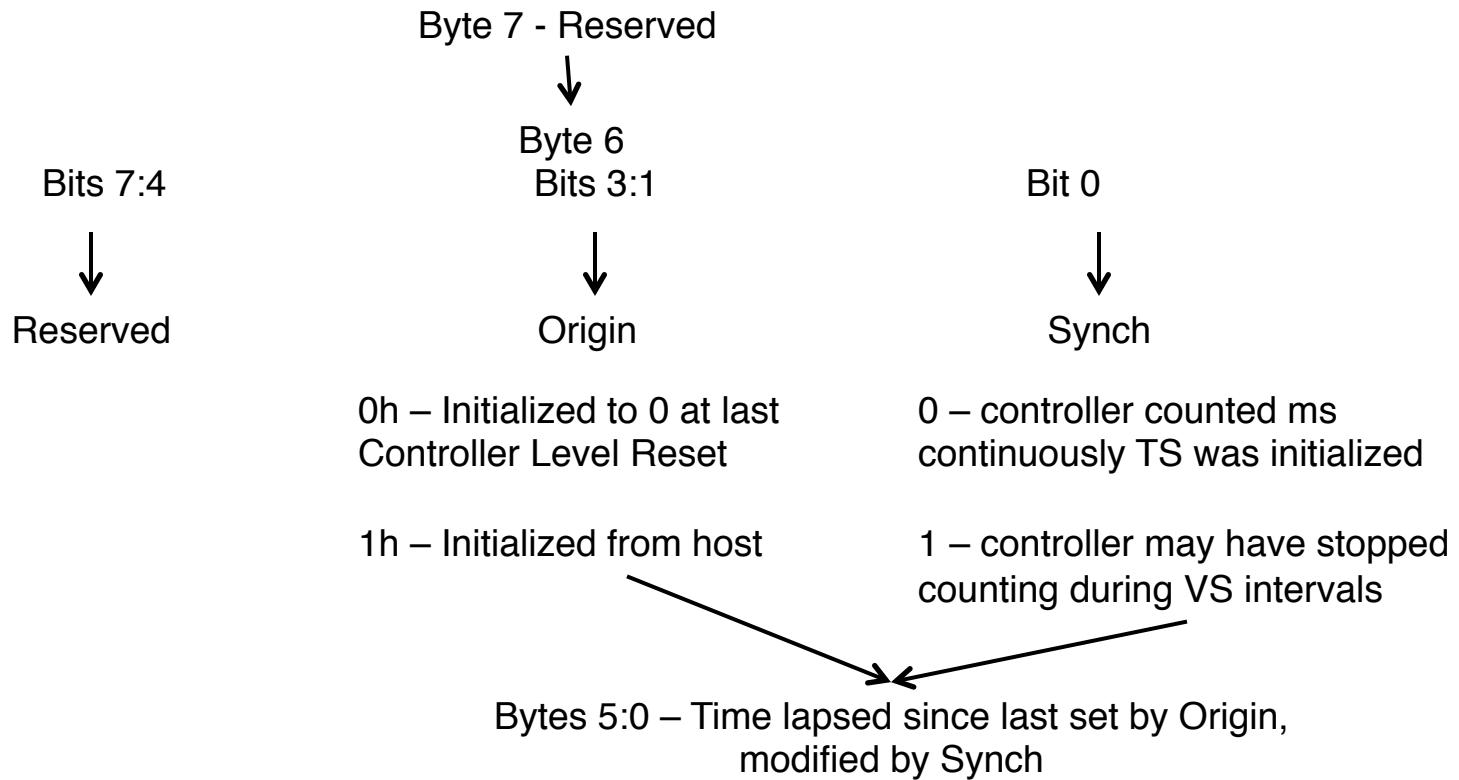


Timestamp Data Structures

Set Features

Bytes	Description
5:0	Number of ms since midnight 1/1/1970 UTC
7:6	Reserved

Get Features



Host Controller Thermal Management

Allows the host to set temperatures (in Kelvin) to instruct the controller to perform light throttling or heavy throttling to reduce the composite temperature

Feature Identifier 10h

Temperatures are set with Set Features Command, Dword 11
Setting can be read with Get Features, Dword 0

Temperatures must be between Minimum Thermal Management Temperature and Maximum Thermal Management Temperature in Identify Controller

Set Features Dword 11

Get Features Dword 0

Bits	Description
31:16	Thermal Management Temperature 1 (light throttling)
15:0	Thermal Management Temperature 2 (heavy throttling)

Telemetry

Enables manufacturers to collect logs to improve functionality and reliability

Data is returned in the Host-Initiated log page or
Controller-Initiated log page

Controller support is indicated in Log Page Attributed in Identify Controller

Non-Operational Power State Configuration

Feature Identifier 11h

Set Features provides information to controller in Dword 11

Get Features get the information from the controller in Dword 0

If Bit 0 = 1b, then the controller may exceed the limits of any non-operational power states, up to the limits of the last operational power state, to perform background operations.

Bits 31:01 are reserved



New Registers



Boot Partition Operation

General

Controller indicates support for Boot Partition Operation in CAP.BPS

Areas are the same size, but may contain different levels of boot code

Write

Boot code is loaded with F/W Download and F/W Commit commands

Read

Host reads BP Size and BP Read Size registers

Host creates memory buffer and reads BP into that buffer

Boot code may be read with CC.EN = 0

Protection

Boot code can be protected with Replay Protected Memory Block

See CAP, BPINFO, BPRSEL, and BPMBL
registers in NVMe 1.3 Specification

ADMIN and NVMe Command Format

Dword	Bytes	31	Name	0
0	03:00		Command ID (CID)	P S
1	07:04		Namespace Identifier (NSID)	Res
2	11:08		Reserved	F
3	15:12			OP Code
4	19:16		Metadata Pointer (MPTR) – Address of physical buffer for metadata	
5	23:20			
6	27:24		PRP Entry 1 (PRP1)	
7	31:28			or SGL 1
8	35:32		PRP Entry 2 (PRP2)	
9	39:36			
10	43:40			
11	47:44			
12	51:48		Command specific fields	
13	55:52			
14	59:56			
15	63:60			



Covered in this Section

New Admin Commands

- Directive Send and Receive
- Device Self-Test
- NVMe-MI Send and Receive
- Doorbell Buffer Configuration
- Virtualization Management
- Sanitize

New Features

- Time Stamp
- Host Controller Thermal Management
- Telemetry
- Non-Operational Power State Configurations

New Registers

- Boot Partition Information
- Boot Partition Read Select
- Boot Partition Buffer Location



Notes



Section 9

Translating

SCSI Commands

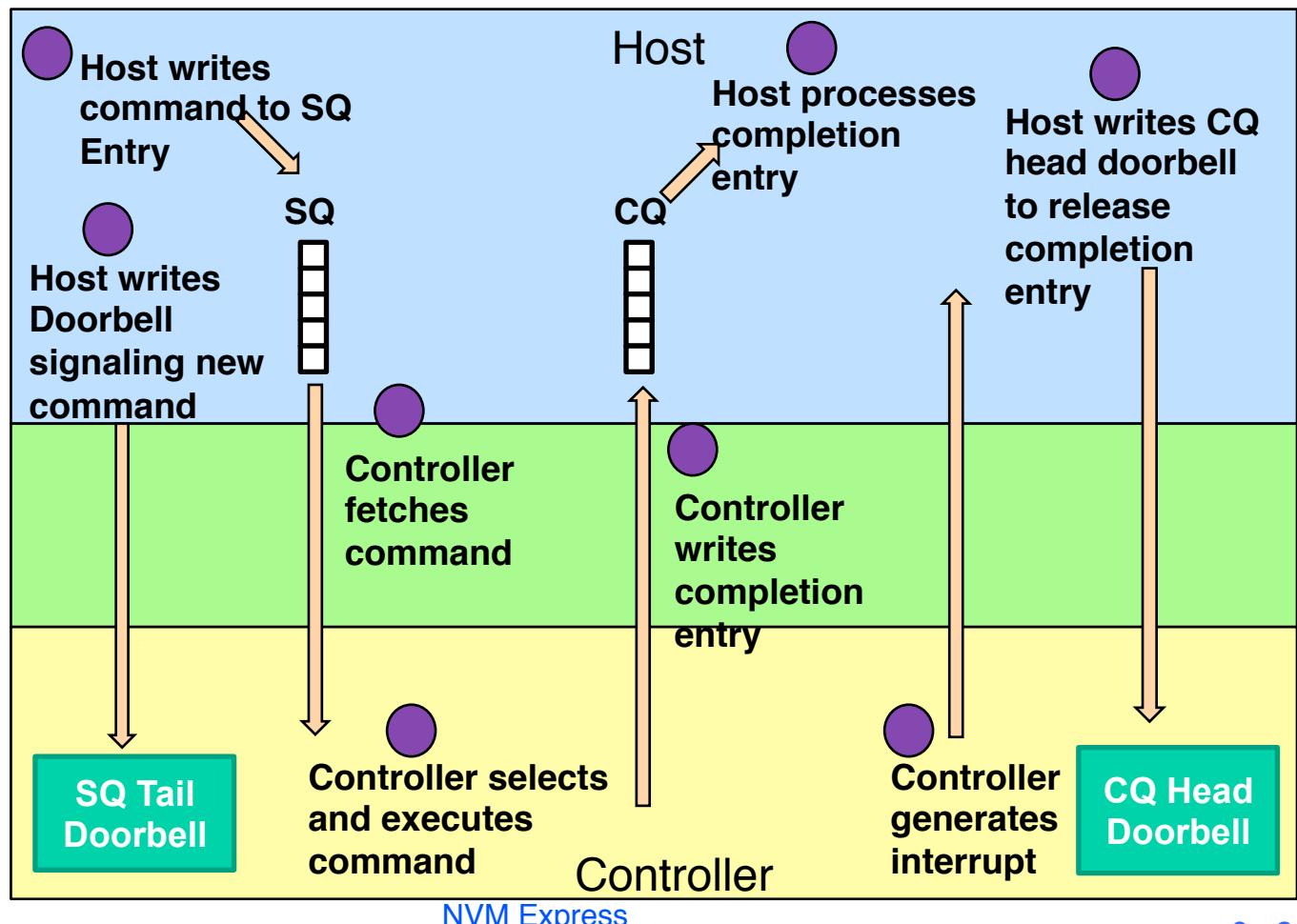


Covered in this Section

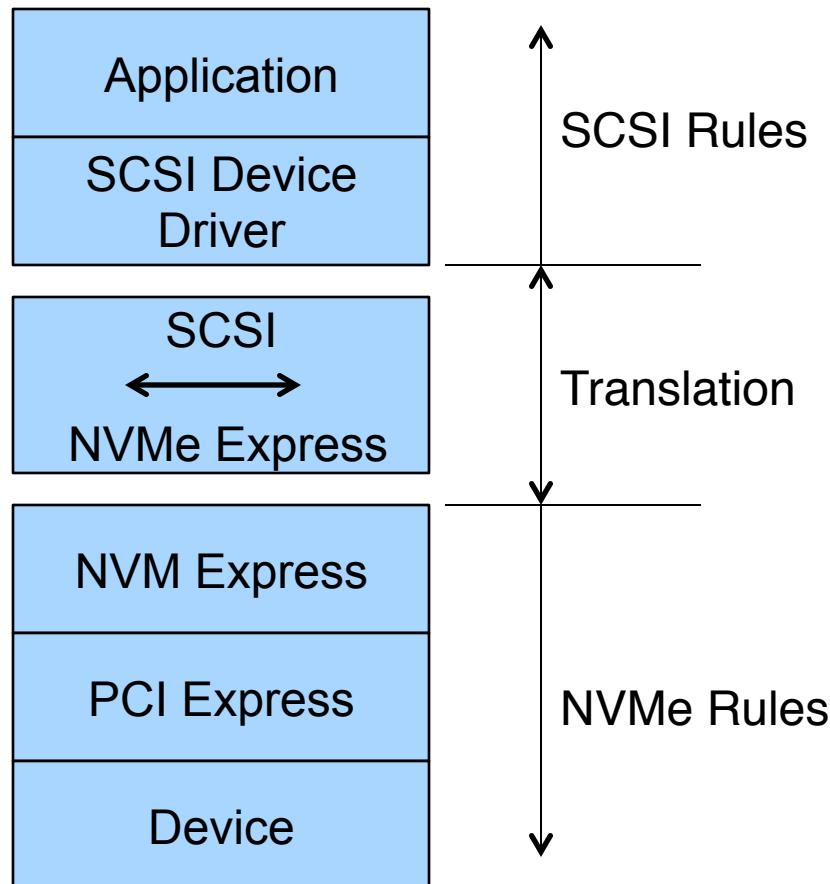
Translating SCSI commands

Translating Status to SCSI

Translating SCSI Task Management Functions



Layers



Format References

Information on SCSI Commands is in:

NVMe – SCSI Translation Reference

Information on SCSI Data is in:

applicable SCSI Command Standard

Information on SCSI Status is in:

SCSI SPC-4 Standard

Information on SCSI Task Management Functions is in:

NVMe – SCSI Translation Reference



Mapping SCSI Commands (SPC)

SCSI Command	NVMe Command	Notes
Inquiry	Identify	Inquiry and VPD supported
Log Sense	Get Features, Get Log Page	Not translated
Mode Select (6, 10)		Up to Device Driver
Mode Sense (6, 10)	Identify, Get Features	
Report LUNs	Identify	
Request Sense	Uses autosense	Responds to both fixed and descriptor formats but with almost no information
Security Protocol In	Security Receive	
Security Protocol Out	Security Send	
Start Stop Unit	Set Features, Get Features, Power state transitioning	
Test Unit Ready		Up to Device Driver
Write Buffer	F/W Image Download, F/W Image Commit	

Mapping SCSI Commands (SBC)

SCSI Command	NVMe Command	Notes
Compare and Write	Compare, Write	Fused operation
Format Unit	Format NVM	
Read (6, 10, 12, 16)	Read	
Read Capacity (10, 16)	Identify	
Synchronize Cache (10, 16)	Flush	
Unmap	Dataset Management	
Verify (10, 12, 16)	Compare	
Write (6, 10, 12, 16)	Write	
Write Long (10, 16)	Write Uncorrectable	

Common Field Mapping

SCSI Command Field	NVMe Support
Allocation Length	Application responsible to ensure adequate buffers are available
Bytchk	Required translation to FUA field of NVMe command
Control	Support unspecified
DPO	Support unspecified
Group Number	Support unspecified
FUA	Requires translation to FUA field on NVMe command
FUA_NV	Support unspecified
LBA	Requires translation to Starting LBA field on NVMe command
Parameter List Length	Application responsible to ensure adequate Buffers are available
Product ID	1 st 16 bytes of Model Number within Identify Controller Data Structure
Product Revision Level	1 st 4 bytes of F/W revision within Identify Controller Data Structure
Immed	0b is supported; 1b returns Illegal Request sense key
T10 Vendor ID	“NVMe” + four spaces
Transfer Length	Requires translation to Number of LB field of NVMe command

Common Field Mapping (Concluded)

SCSI Command Field	NVMe Support		
WRProtect RDProtect VRProtect	WRProtect	PRACT	PRCHK
	000b	X	000b
	001b, 101b	0	111b
	010b	0	011b
	011b	0	000b
	100b	0	100b

PRACT – Protection Action

- 1 – Strip PI on Read, Insert PI on Write
- 0 – Pass PI

PRCK – Protection Check

- Bit 0 – 1 enables PI checking on LBA Reference Tag
- Bit 1 – 1 enables PI checking on Application Tag
- Bit 2 – 1 enables PI checking on Guard field

PRACT and PRCHK are only used if namespace is formatted for End-to-End Protection

Read Command



Read Command

Dword	Bytes	Name		
0	03:00	Common Fields	Op Code 02h	
1-5	23:04	Common Fields		
6	27:24			
7	31:28	PRP or SGL		
8	35:32	Data buffer for returned information		
9	39:36			
10	43:40	Starting LBA		
11	47:44			
12	51:48	Note A	Reserved	Number of LB
13	55:52		Reserved	Note B
14	59:56	Expected Initial LB Reference Tag		
15	63:60	Expected Logical Block Application Tag Mask		

Read Command Fields

Dword	Bit	Description
12	31	Limited Retry
	30	Force Unit Access (FUA)
	29:26	Protection Information Field
13	7	Incompressible
	6	Sequential Request
	5:4	Access Latency
	3:0	Access Frequency

RDProtect	PRACT	PRCHK
000b	1	111b
001b, 101b	0	111b
010b	0	011b
011b	0	000b
100b	0	100b

Value	Definition
00b	None, no latency information provided
01b	Idle, longer latency acceptable
10b	Normal, typical latency
11b	Low, smallest possible latency

Value	Definition
0000b	No frequency information provided
0001b	Typical number of r/w for this LBA range
0010b	Infrequent r/w for this LBA range
0011b	Infrequent write, frequent read
0100b	Frequent write, infrequent read
0101b	Frequent r/w for this LBA range
0110b	One time read
0111b	Speculative read (prefetch op)
1000b	LBA range to be overwritten soon



Write Command



Write Command

Dword	Bytes	Name		
0	03:00	Common Fields	Op Code 01h	
1-5	23:04	Common Fields		
6	27:24			
7	31:28	PRP or SGL		
8	35:32	Data buffer for write information		
9	39:36			
10	43:40	Starting LBA		
11	47:44			
12	51:48	Note A	Reserved	Number of Logical Blocks
13	55:52		Reserved	Note B
14	59:56	Expected Initial LB Reference Tag		
15	63:60	Expected Logical Block Application Tag Mask		

Write Command Fields

Dword	Bit	Description
12	31	Limited Retry
	30	Force Unit Access (FUA)
	29:26	Protection Information Field
13	7	Incompressible
	6	Sequential Request
	5:4	Access Latency
	3:0	Access Frequency

WRProtect	PRACT	PRCHK
000b	1	000b
001b, 101b	0	111b
010b	0	011b
011b	0	000b
100b	0	100b

Value	Definition
00b	None, no latency information provided
01b	Idle, longer latency acceptable
10b	Normal, typical latency
11b	Low, smallest possible latency

Value	Definition
0000b	No frequency information provided
0001b	Typical number of r/w for this LBA range
0010b	Infrequent r/w for this LBA range
0011b	Infrequent write, frequent read
0100b	Frequent write, infrequent read
0101b	Frequent r/w for this LBA range
0110b	One time write



SCSI

Inquiry Command



Translating SCSI Inquiry Command

Dword	Bytes	Name				
0	03:00	Command ID (CID)	Res	Fuse	OP Code = 06h	
1	07:04	Namespace Identifier (NSID)				
2	11:08		Reserved			
3	15:12					
4	19:16	Metadata Pointer (MPTR) – Address of physical buffer of metadata				
5	23:20					
6	27:24					
7	31:28		PRP			
8	35:32		Data buffer for inquiry information			
9	39:36					
10	43:40		Reserved	CNS		
11	47:44					
12	51:48					
13	55:52		Command specific fields			
14	59:56					
15	63:60					



Controller or Namespace Structure (CNS)

Required if
Namespace
Management is
supported

Value	Description
00h	Return Identify data structure for Namespace in dword 1 if attached to this controller
01h	Return Identify data structure for controller
02h	Return a list of active namespace ID greater than namespace in dword 1 Up to 1024 namespaces Zero filled
03-0Fh	Reserved
10h	List of up to 1024 Namespace ID > NSID in CDW1.NSID
11h	Data structure for Namespace identified in CDW1.NSID
12h	List of up to 2047 controller IDs identified in CWD10.CNTID and attached to CDW1.NSID
13h	List of up to 2047 controller ID containing the controller in CDW10.CNTID but may or may not be attached to namespaces.
14-FFh	Reserved

Standard Inquiry Return Data Locations

Byte	7	6	5	4	3	2	1	0
0	Peripheral Qualifier = 000b			Peripheral Device Type = 00h				
1	RMB = 0	Reserved = 00h						
2	Version = 06h (SPC-4)							
3	R = 0	R = 0	NormACA = 0	HiSup = 1	Response Data Format = 0010b (SPC-4)			
4	Additional Length = 1Fh							
5	SCCS = 0	ACC = 0	TPGS = 00b		3PC = 0	Reserved = 00b		Protect
6	Obs = 0	EncServ = 0	VS	MultiP = 0	Obs = 0	R = 0	R = 0	Addr16 = 0
7	Obs = 0	R = 0	WBus16 = 0	Sync = 0	Obs = 0	R = 0	CmdQue = 1	VS
8	T10 Vendor ID = "NVMe "							
15								
16	Product Identification =							
31	1 st 16 bytes of Model Field within Identify Controller Data Structure							
32	Product Revision Level =							
35	1 st 4 bytes of F/W Revision within Identify Controller Data Structure							



Vital Product Data Pages (VPD)

Page Code	Page Name	Page Length	Location
00h	Supported Pages	Actual length	List of pages
80h	Unit Serial Number	20	Identify data bytes EUI64 in ASCII
83h	Device ID	Actual length	See Next 4 Pages
86h	Extended Inquiry	3C	
B0h	Block Limits	3C	See NVMe – SCSI Translation Standard
B1h	Block Device Characteristics	3Ch	Medium Rotation Rate = 0001h (non-rotating) Nominal Form Factor = 0h (form factor not reported)
B2h	Provisioning	04h	See ahead
All Others	Controller terminates command with Check Condition, Illegal Request Sense Key		



VPD Page 83 – Page Format

Byte	7	6	5	4	3	2	1	0
0	Peripheral Qualifier = 000b				Peripheral Device Type = 00h			
1				Page Code = 83h				
2					Page Length			
3								
4				First Descriptor				
n				Last Descriptor				

VPD Page 83 – IEEE Registered Extended Descriptor

Byte	7	6	5	4	3	2	1	0					
0	Protocol Identifier = 0h					Code Set = 1h (Binary)							
1	PIV = 0		R = 0	Assoc = 00b (LU)			Designator Type = 3h (NAA)						
2	Reserved = 00h												
3	Designator Length (n-3)												
Descriptor	0	NAA = 6h (IEEE Registered Extended)											
	1	IEEE Company ID											
	2												
	3												
	4	Vendor Specific ID											
	7												
	8	Vendor Specific ID Extension											
	15												

Vendor Specific ID and Vendor Specific ID Extension
are formed by concatenating 36 bits of 0's to EUI64.

EUI64 identifies each namespace and can be found in
Identify Namespace Data Structure - bytes 127:120

NVM Express

VPD Page 83 – Locally Assigned Descriptor

Byte	7	6	5	4	3	2	1	0								
0	Protocol Identifier = 0h						Code Set = 1h (Binary)									
1	PIV = 0	R = 0	Assoc = 00b (LU)			Designator Type = 3h (NAA)										
2	Reserved = 00h															
3	Designator Length =08															
Descriptor																
Designator																
0	NAA = 3h (Locally Assigned Designator)															
1																
2																
3																
4																
7																
Locally assigned ID																

EUI64 identifies each namespace and can be found in
Identify Namespace Data Structure - bytes 127:120

VPD Page 83 – T10 ID Descriptor

Byte	7	6	5	4	3	2	1	0			
0	Protocol Identifier = 0h					Code Set = 2h (ASCII)					
1	PIV = 0	R = 0	Assoc = 00b (LU)			Designator Type = 8h (SCSI name)					
2	Reserved = 00h										
3	Designator Length										
Descriptor											
Designator											
0	T10 Vendor ID = “NVMe”										
7											
8	Vendor Specific ID										
n											

T10 Vendor ID is “NVMe” followed by 4 spaces

Vendor Specific ID is concatenation of:

- 1st - first 16 bytes of Model Number field within the Identify Controller Data Structure
- 2nd - IEEE EUI64 of Identify Namespace Data Structure



VPD Page B1 – Block Device Characteristics

Byte	7	6	5	4	3	2	1	0
0	Peripheral Qualifier = 000b				Peripheral Device Type = 00h			
1				Page Code = B1h				
2					Page Length = 003Ch			
3								
4				Medium Rotation Rate = 0001h (non-rotating)				
5								
6				Reserved				
7		Reserved			Nominal Form Factor = 0h (not reported)			
8				Reserved				
63								

VPD Page B2 – Logical Block Provisioning

Byte	7	6	5	4	3	2	1	0
0	Peripheral Qualifier = 000b				Peripheral Device Type = 00h			
1				Page Code = B2h				
2					Page Length = 0004h			
3								
4				Threshold Exponent = 00h				
5	LBPU	LBPWE	LBPWS10	Reserved	LBPRZ	ANC-SUP	DP = 0	
6				Reserved		Provisioning Type		
7				Reserved				
8					Provisioning Group Descriptor (none)			
n								

VPD Page B2 – Logical Block Provisioning

LBPU = 1b if Dataset Management command (Deallocate) is supported

LBPWS = 0b to indicate Write Same (16) to unmap is not supported

LBPWS10 = 0b to indicate Write Same (10) to unmap is not supported

LBPRZ = 1b if Dataset Management command (Deallocate) is supported

Anc_sup = 1b to indicate that setting anchor bit in unmap is supported
0b to indicate that setting the anchor bit in unmap is not supported

DP = 0b to indicate that no Provisioning Group Descriptors follow

Provisioning Type

0 = Full

1 = Resource

2 = Thin

SCSI

Request Sense

Command



Request Sense Return Data – Fixed Format

Byte	7	6	5	4	3	2	1	0
0	Valid = 0							Response Code = 70h
1								Obsolete
2	FM = 0	EOM = 0	ILI = 0	SDAT OVFL				Sense Key = 0h
3								Information bytes = 00 00 00 00h
6								
7								Additional Sense Length
8								
11								Command-Specific Info = Info depending on command
12								Additional Sense Code = 00h if device is in power state 00h, otherwise Low Power Condition On
13								Additional Sense Code Qualifier = 00h
14								Field Replaceable Unit = 00h
15	SKSV = 0							
16								
17								Sense Key Specific Information = 00 00 00h
18								
n								Additional Sense Bytes



Request Sense Return Data – Descriptor Format

Byte	7	6	5	4	3	2	1	0
0								Response Code = 72h
1				Reserved = 0h				Sense Key = 0h
2				Additional Sense Code = 00h if device is in power state 00h, otherwise Low Power Condition On				
3				Additional Sense Code Qualifier = 00h				
4	SDAT OVFL				Reserved = 00h			
5								
6					Reserved = 00 00h			
7				Additional Sense Length = 00h				

SCSI

Autosense



Status Mapping

SCSI			NVMe
Status Code	Sense Key	ASC	Status Code
Good	No Sense	N/A	Success Completion
Check Condition	Medium Error	Not Ready, Cause not reportable	NS Not Ready
			Data Transfer Error
			Capacity Exceeded
		Peripheral Device Write Fault	Write Fault
		Unrecoverable Read Error	Unrecoverable Read Error
		Logical Block Guard check failed	E-to-E Guard check error
		Logical Block application tag check failed	E-to-E Application Tag Check failed
		LB Ref Tag Check failed	E-to-E Reference tag check error
		Internal Target Failure	Internal Device Error

Status Mapping

SCSI			NVMe
Status Code	Sense Key	ASC	Status Code
Check Condition	Illegal Request	Access Denied – Invalid LU ID	Invalid NS or Format
		LBA out of Range	LBA out of Range
		Invalid Cmd Op Code	Invalid Cmd Op Code
		Invalid field in CDB	Invalid Field in Cmd
			Completion Q Invalid
			Abort Command Limit Exceeded
		Format Command Failed	Invalid Format
		Invalid field in CDB	Conflicting Attributes
		Access Denied – Invalid LU Identifier	Access Denied
	Miscompare	Miscompare during verify Op	Compare Failure

Status Mapping – Autosense

SCSI			NVMe
Status Code	Sense Key	ASC	Status Code
Task Aborted	Aborted Command	Warning – Power Loss expected	Cmds aborted due to Power Loss Notification
			Cmd Abort Requested
			Cmd Aborted due to Failed fused cmd
			Cmd Aborted due to Missing fused cmd
			Cmd Aborted due to SQ Deletion

Translating SCSI Task Management Functions



Mapping SCSI Task Management Functions

SCSI Task Mgmt Function	NVMe Command	Notes
Abort Task	Abort Command	
Abort Task Set	Abort Command	Issued for each command
Clear Task Set	Abort Command	Issued for each command
Clear ACA		Unspecified
IT Nexus Reset		Return: a) Function Succeeded if outstanding commands in submission queue or b) Function Complete
Logical Unit Reset	Supported by writing 0 to Enable field in Controller Config Reg.	Creates Controller Reset
Query Task	N/A	May be supported
Query Task Set	N/A	Unspecified
Query Asynch event	N/A	Unspecified Should be handled by Admin Cmd – Asynch Event Request



Covered in this Section

Mapping SCSI commands

Mapping SCSI Status

Mapping SCSI Task Management Functions



Notes



Notes



Section 10

Future Directions



NVMe Future

NVMe 1.4

I/O Determinism

Enhancement of SES over NVMe

Persistent Memory

Polling instead of Interrupt???



NVMe Management 1.1

Enclosure Management

Native PCIe enclosure management

SES-based enclosure management



NVMe Over Fabrics

The specification exists
needs to catch on



Covered in this Section

NVMe 1.4

NVMe Management 1.1

NVMe Over Fabrics



Thank You

KnowledgeTek
Seminars@KnowledgeTek.com

Hugh Curley
Hugh@HughCurley.com



Notes



Notes

