SNIA™
Advancing storage & information technology

SFF-TA-1005

Specification for

Universal Backplane Management (UBM)

Rev 1.3.2 September 20, 2021

SECRETARIAT: SFF TA TWG

ABSTRACT: This specification defines the Universal Backplane Management structure.

This specification provides a common reference for systems manufacturers, system integrators, and suppliers.

POINTS OF CONTACT:

Josh Sinykin/Jason Stuhlsatz
Broadcom Limited
4385 River Green Parkway
Duluth, GA 30096

Ph: 678-728-1406
Email: josh.sinykin@broadcom.com / jason.stuhlsatz@broadcom.com

Chairman: SFF TA TWG
Email: SFF-Chair@snia.org

Foreword

The development work on this specification was done by the SNIA SFF TWG, an industry group. Since its formation as the SFF Committee in August 1990, the membership has included a mix of companies which are leaders across the industry.

For those who wish to participate in the activities of the SFF TWG, the signup for membership can be found at https://www.snia.org/sff/join.

Revision History

Rev 1.0 May 4, 2018
- Initial release

Rev 1.1 November 16, 2018
- Update to 2Wire_RESET# signal definition related to 2Wire Mux topology (Table 4-2 and Section 6.2.11)
- Update to PCIe Reset field definition (Section 4.16)
- Update to 2Wire Max Byte Count definition to include 128 and 256 bytes (Section 5.3.1.2.2)
- Update to Operational State field definition (Section 4.20 and 6.2.1)
- Added Clarifying statements to PMDT Read and Write Transactions (Section 6.2.6.1)
- Added Note to Number of Bytes in a Sector Index (Section 6.2.6.2)
- Updated references to sections (incorrect from previous specification)

Rev 1.2 April 25, 2019
- Update DFC Status and Control to include an individual change count for each DFC Status and Control Descriptor.
- Fixed error in document. UBM Host uses Read Checksum instead of LCS to check for valid Read response.
- Fixed error in Change Count Command field description.

Rev 1.3 January 15, 2020
- Updated Section 3 definition of HFC
- Updated Section 4.10 definition of HFC Identity to match Section 3 and Section 4.12
- Updated Section 4.12 with new Figures and expanded language
- Updated Section 4.16 with DFC PERST# Management Override support and field usages
- Updated 5.3.1.2.3 – UBM FRU Invalid field description
- Updated Data Byte 4 definition of UBM Port Route Descriptor – clarification of bit rates for SAS and PCIe and SATA
- Updated 6.2.9 – Backplane Type field description
- Updated 6.2.11 Capabilities Command with DFC PERST# Management Override
- Updated 6.2.12 Features Command with DFC PERST# Management Override
- Updated 6.2.15 – Device Off handling description
- Updates Section B.5 – Backplane Number and Backplane Type field usages
- Added Appendix C

Rev 1.3.1 September 17, 2021
- Updated Signal Definition
- Updated UBM Overview Data Byte 9 Description
- Updated UBM Port Route Descriptor Supported Types (See 6.3.2.2.3)
- Updated Capabilities (See 7.2.11)
- Updated Feature Command (See 7.2.12)
- Updated Change Count Command (See 7.2.13)
- Updated Get Non-Volatile Storage Geometry Subcommand definition for the first Sector Index instance. (See 7.2.6.2)

Rev 1.3.2 September 20, 2021
- Fixed broken reference links due to new file formatting
- Various editorial and formatting changes

CONTENTS

FIGURES

TABLES

1. Scope

This specification defines Universal Backplane Management (UBM) which provides a common backplane management framework for a host to determine SAS/SATA/PCIe backplane capabilities, Drive Facing Connector (DFC) Status and Control information, and to read the port routing of the Drive Facing Connectors to Host Facing Connectors (HFC) of the backplane.

The Universal Backplane Management framework provides:

- Backplane capabilities including:
  - PCIe Reference Clock expectations (RefClk or SRIS/SRNS)
  - PCIe Reset expectations
  - PwrDIS signal support
  - Dual Port support
- High speed lane port routing assignments to Host Facing Connectors
- Number of Drive Facing Connectors supported by the backplane
- Status and Control over Drive Facing Connector I/O and LED States
- Host to backplane cable installation order independence
- Backplane programmable code update

This specification does not mandate backplane LED pattern definitions.

2. References

2.1 Industry Documents

The following documents are relevant to this specification:

– Gen-Z	Scalable Connector Specification 1.0
– Gen-Z	SFF 8639 2.5-Inch Compact Specification
– INCITS 534/T10	Serial Attached SCSI - 4 (SAS-4)
– INCITS 518/T10	SCSI Enclosure Services – 4 (SES-4)
– PCI-SIG	PCI Express SFF-8639 Module Specification
– PCI-SIG	PCI Express Base Specification, Revision 3.0
– Serial ATA	International Organization Serial ATA Specification
– IPMI	Platform Management FRU Information Storage Definition – Rev 1.3
– SFF-8448	SAS Sideband Signal Assignments
– SFF-8485	Serial GPIO Bus
– SFF-8489	Serial GPIO IBPI (International Blinking Pattern Interpretation)
– SFF-8630	Serial Attachment 4X 12 Gb/s Unshielded Connector
– SFF-8639	Multifunction 6X Unshielded Connector
– SFF-8680	Serial Attachment 2X 12 Gb/s Unshielded Connector
– SFF-9402	Multi-Protocol Internal Cables for SAS and/or PCIe
– SFF-9639	Multifunction 6X Unshielded Connector Pinouts
– SFF-TA-1001	Universal x4 Link Definition for SFF-8639

2.2 Sources

The complete list of SFF documents which have been published, are currently being worked on, or that have been expired by the SFF Committee can be found at https://www.snia.org/sff/specifications. Suggestions for improvement of this specification will be welcome, they should be submitted to https://www.snia.org/feedback.

Other standards may be obtained from the organizations listed below:

Standard	Organization	Website
ANSI standards	International Committee for Information Technology Standards (INCITS)	https://www.incits.org
Gen-Z Consortium	Gen-Z Consortium	https://genzconsortium.org/specifications/
IPMI	Intel	https://www.intel.la/content/www/xl/es/servers/ipmi/ipmi-technical-resources.html
PCIe	PCI-SIG	https://pcisig.com

2.3 Conventions

The ISO convention of numbering is used i.e., the thousands and higher multiples are separated by a space and a period is used as the decimal point. This is equivalent to the English/American convention of a comma and a period.

American	French	ISO
0.6	0,6	0.6
1,000	1 000	1 000
1,323,462.9	1 323 462,9	1 323 462.9

3. Keywords, Acronyms, and Definitions

For the purposes of this document, the following keywords, acronyms, and definitions apply.

3.1 Keywords

Optional: This term describes features which are not required by the SFF Specification. However, if any feature defined by the SFF Specification is implemented, it shall be done in the same way as defined by the Specification. Describing a feature as optional in the text is done to assist the reader. If there is a conflict between text and tables on a feature described as optional, the table shall be accepted as being correct.

Reserved: Where this term is used for defining the signal on a connector contact. Its actual function is set aside for future standardization. It is not available for vendor specific use. Where this term is used for bits, bytes, fields and code values; the bits, bytes, fields and code values are set aside for future standardization. The default value shall be zero. The originator is required to define a Reserved field or bit as zero, but the receiver should not check Reserved fields or bits for zero.

3.2 Acronyms and Abbreviations

CPRSNT#: Cable Present signal, an active-low signal provided by an Endpoint to indicate that it is both present and its power is within tolerance
DFC: Drive Facing Connector, describes the connector assembly on the backplane that connects to the drive
DFC 2Wire: 2Wire interface connected to the Drive Facing Connector
FRU: Field Replaceable Unit
HFC: Host Facing Connector, describes the connector assembly on the backplane that connects to the Host
IPMI: Intelligent Platform Management Interface
NVRAM: Non-volatile Random Access Memory, used to store the UBM FRU data structures
PCIe: PCI Express
PERST#: A PCI signal which provides a reset to a PCIe device.
SMBRST#: A Serial Management Bus (SMBus) Reset signal which provides a reset to the DFC SMBus logic.
SRIS: Separate Reference Clock with Independent Spread Spectrum Clocking
SRNS: Separate Reference Clock with No Spreading Spectrum Clocking
UBM: Universal Backplane Management represents this specification.

3.3 Definitions

2Wire Master: Industry standard two wire protocol responsible for initiating communication

2Wire Slave: Industry standard two wire protocol responsible for accepting communication when addressed by a 2Wire Master

Backplane: A board containing Host Facing Connectors and Device Facing Connectors.

Converged: A port that supports PCIe protocol and SAS/SATA protocol via the same port (e.g., SFF-TA-1001)

Chassis: Physical enclosure which contains the system and backplanes

Device: A hard drive or solid state drive that plugs into the Drive Facing Connector on the backplane

Host: A Storage Controller Adapter, PCIe Switch, and/or Root Complex port which is responsible for 2Wire Master communication with the UBM FRU and UBM Controller on the backplane

Management Resource: An exposed interface to provide management services (i.e., Redfish Chassis Resource, or SCSI Enclosure Services Device)

Port: Groupings of the high speed transmit and receive differential signals.

Quad PCIe: Complies with PCI-SIG PCI Express SFF-8639 Module Specification

RefClk: PCIe Reference Clock

Segregated: A port that supports PCIe protocol and does not support SAS/SATA protocol via the same port.

Tri-mode: Host that may provide connectivity for SAS, SATA, and PCIe devices.

UBM FRU: A 256 byte non-volatile memory storage device which contains an IPMI FRU formatted record content. Content provides port routing map describing the Drive Facing Connector ports to Host Facing Connectors.

UBM Controller: Microcontroller, CPLD or ASIC which provides 2Wire Slave interfaces that provide the UBM command interface.

UBM Controller Image: A vendor specific programmable dataset that implements the UBM Controller functionality (e.g., Microcontroller Firmware or CPLD compiled dataset)

4. General Description

This specification provides the 2Wire management transaction format and content to define the UBM backplane capabilities and port routing of the backplane.
The UBM Controller presents a 2Wire Slave interface that provides backplane capabilities and DFC Status and Control Descriptors. The UBM FRU connected to the same 2Wire Slave interface implements a NVRAM formatted as an IPMI FRU record. The UBM IPMI Multi-records provide the UBM Port Route Information Descriptors which is used by the Host to create an access map consisting of the Drive Facing Connector, port link width, Host Facing Connector, and Host Facing Connector Starting Lane within the connector. The UBM IPMI MultiRecords also provides the 2Wire Slave address for one or more UBM Controllers.

The port routing information provides the Host the ability to support x4, x2, and x1 high speed ports routed between the DFC and the HFC.

The UBM Controller provides backplane implementation features and options that are important for the initialization process of the devices.
These features and options include:

- PCIe Reference Clock expectations (RefClk or SRIS/SRNS)
- Device Power Control via PwrDIS (Power Disable)
- PCIe Reset Control
- Detection of the installed device type via PRSNT#, IFDET#, and IFDET2# signals
- Single or Dual Port supported
- Backplane UBM Controller Image Update

The UBM Backplane in Figure 4-1 depicts the Host Facing Connector relationships to the UBM FRU, UBM Controller(s) and Drive Facing Connectors. The Host Facing Connector provides sideband I/O signals which route to the UBM Controller(s) and the UBM FRU. High speed I/O routes from the Host Facing Connector to the Drive Facing Connector(s). The UBM Controllers manage both the sidebands from the Host Facing Connector and the Drive Facing Connectors I/O signals. The UBM FRU provides non-volatile memory storage that contains the routing information for the UBM Controller(s) and the Drive Facing Connector high speed I/O ports.

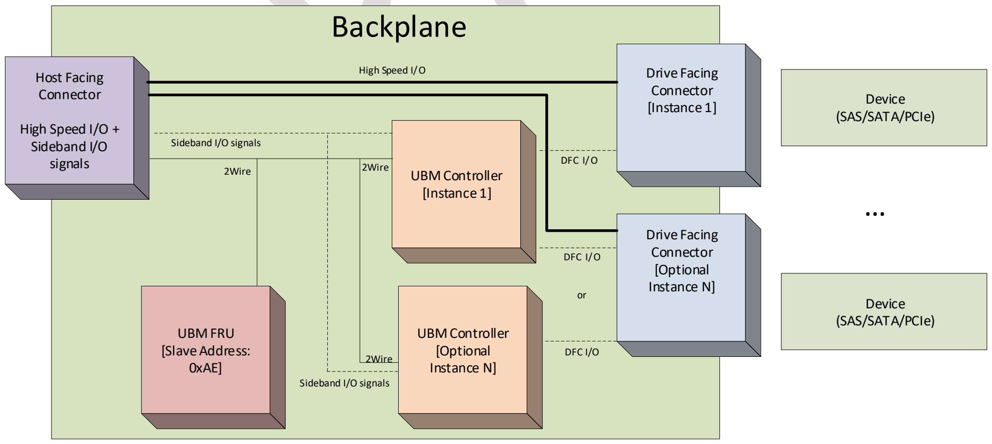

Backplane
Host Facing Connector
High Speed I/O + Sideband I/O signals
High Speed I/O
Sideband I/O signals
2Wire
2Wire
UBM Controller [Instance 1]
Drive Facing Connector [Instance 1]
DFC I/O
Device (SAS/SATA/PCIe)
...
Drive Facing Connector [Optional Instance N]
DFC I/O
or
DFC I/O
Device (SAS/SATA/PCIe)
UBM FRU [Slave Address: 0xAE]
2Wire
Sideband I/O signals
UBM Controller [Optional Instance N]

Figure 4-1 UBM Backplane Overview

The backplane may implement more than one Drive Facing Connector per Host facing connector which paves the way for x1, x2, x4 and future drive port link width route mapping. The backplane may also implement more than one Host Facing Connector to support additional Drive Facing Connector routings or Host attachments. The UBM Controller implementation shall provide a unique Host Facing Connector Identity field within the same Backplane indicating the same Backplane Number field. Multiple Host Facing Connectors shall not interconnect their 2Wire interfaces with other Host Facing Connectors 2Wire interfaces. These requirements enable cable installation order resolution by the Host.

The UBM System Deployment view in Figure 4-2 shows many connection options between a Host (e.g., Adapters, Root Complexes, PCIe Switches, SAS Expanders) and Backplanes. Multiple backplanes may exist inside the chassis. The High speed cable and sideband I/O signals are used for communicating with the backplane. The UBM FRU shall be 2Wire addressed at a fixed 8-bit address of 0xAE. The UBM FRU provides the UBM Controller 2Wire addresses necessary for the Host to communicate with the UBM Controllers on the backplane.

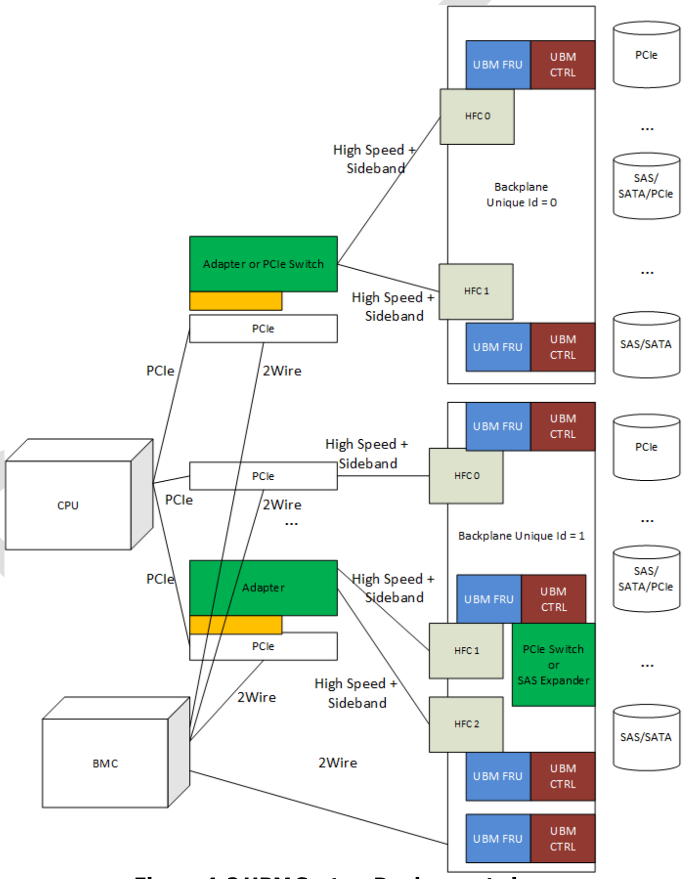


Figure 4-2 UBM System Deployment view

5. Concepts

5.1 Host Facing Connector Requirements

Each HFC routes its set of high speed lanes to zero or more DFCs on the backplane. The UBM backplane shall route high speed lanes from a DFC in consecutive order to the HFC.

Note: The HFC may supply any number of lanes, but typically are seen in the form of 4 or 8 lanes.

At least one HFC per backplane shall have a 2Wire serial bus connection with the proper BP_TYPE signal usage as defined in SFF-8448 in order to properly detect and operate in 2Wire mode as opposed to the electrically different SGPIO interface (SFF-8485).

The HFC shall use the cable sideband I/O signals as defined in Table 5-1.

Table 5-1 Host Facing Connector Sideband Signal Requirements

SIDE BAND I/O SIGNAL	BACKPLANE I/O TYPE	DESCRIPTION	MANDATORY / OPTIONAL	BACKPLANE INITIALIZATION STATE
SDA	Bidirectional (open-drain)	2Wire Data	Mandatory	HIGH
SCL	Bidirectional (open-drain)	2Wire Clock	Mandatory	HIGH
GROUND	Ground	Connection to the ground plane	Mandatory	GROUND
2WIRE_RESET#	Input (open-drain)	Reset for 2Wire interface of the UBM FRU and UBM Controller. Driven LOW by the Host to indicate a Reset of the 2Wire Slave in the UBM FRU and UBM Controller. Floated HIGH for normal 2Wire Slave operation.	Optional	HIGH
CPRSNT# / CHANGE_DETECT#	Output (open-drain)	The CHANGE_DETECT# signal provides an interrupt mechanism for the backplane to inform the Host of a change in the backplane. Driven LOW by the UBM Controller instance to indicate a change has been detected. When multiple UBM Controllers are associated to a Host Facing Connector, the CHANGE_DETECT# is driven LOW and held Low by the UBM Controller that detected the change. The Host shall clear the CHANGE_DETECT# by writing the obtained Change Count field to each UBM Controller until the CHANGE_DETECT# returns HIGH. Once the CHANGE_DETECT# signal returns HIGH, the Host may complete its UBM Controller change detection search. Floated HIGH by the UBM Controller when the change has been cleared by the Host. See Section 5.8 and Section 5.9 for additional information about the CPRSNT#/CHANGE_DETECT# signal. Note: This signal is also known as CONTROLLER_TYPE in the SFF-8448 specification, and known as CPRSNT# (Cable Present) in the SFF-9402 specification.	Mandatory	LOW
BP_TYPE	Output (resistive pull-up)	Backplane Type signal requirement is defined in the SFF-8448 specification. A UBM Controller shall pull this signal HIGH to indicate a 2Wire backplane interface.	Mandatory	HIGH
REFCLK+-	Input	RefClk, optional for SRIS/SRNS backplanes	Optional	HIGH
PERST#	Input (open-drain)	PCIe Reset	Mandatory	HIGH

5.2 HFC 2WIRE_RESET# signal

The HFC 2WIRE_RESET# signal is an optional open-drain input to the UBM Controller. The 2WIRE_RESET# Operation field (See 7.2.11) indicates the 2WIRE_RESET# operation support. If multiple UBM Controllers are implemented, each UBM Controller shall indicate the same value in the 2WIRE_RESET# Operation field. The HFC 2WIRE_RESET# signal, if implemented on the backplane, may be used to reset:

a. the UBM Controller 2Wire Slave interface and 2Wire Mux if present;

or

b. the UBM FRU, the UBM Controller(s) and 2Wire Mux depending upon the length of time that the 2WIRE_RESET# signal is asserted as defined by Table 5-2.

Table 5-2 Host And UBM Controller 2WIRE_RESET# Timing

HOST ASSERTION MIN	HOST ASSERTION MAX	UBM CONTROLLER ASSERTION DETECTION MIN	UBM CONTROLLER ASSERTION DETECTION MAX	RESET DESCRIPTION
N/A	N/A	0us	900us	Assertions Ignored
1000 us	5000 us	900 us	5100 us	UBM Controller 2Wire Slave Interface and 2Wire Mux
6000 us		5900 us		UBM FRU and UBM Controller(s) and 2Wire Mux

5.3 HFC PERST# signal

The HFC PERST# signal when asserted by the Host shall assert the corresponding port specific DFC PERST# signals that are routed from the DFC to the HFC.

5.4 UBM FRU Sizing Considerations

The UBM FRU is a 256 byte NVRAM. The UBM FRU data is organized in an IPMI FRU format. The remaining memory for Vendor Specific usage is affected by the number of DFC ports described by the UBM FRU.

Table 5-3 provides memory size requirements for some example configurations:

Table 5-3 UBM FRU Memory Size Considerations

NUMBER OF DFC PORTS	IPMI COMMON HEADER (BYTES)	UBM FRU OVERVIEW AREA (BYTES)	UBM PORT ROUTE INFORMATION AREA	TOTAL SIZE CONSUMED (BYTES)	REMAINING SIZE FOR VENDOR SPECIFIC USE (BYTES)
1	8	16	12 Bytes Padded to 16 Bytes	40	216
2	8	16	19 Bytes Padded to 24 Bytes	48	208
4	8	16	33 Bytes Padded to 40 Bytes	64	192
8	8	16	61 Bytes Padded to 64 Bytes	88	168
16	8	16	117 Bytes Padded to 120 Bytes	144	112
24	8	16	173 Bytes Padded to 176 Bytes	200	56
32	8	16	229 Bytes Padded to 232 Bytes	256	0

5.5 2Wire Device Topology

The 2Wire Device Arrangement field (See Section 6.3.1.2.2) indicates the backplane 2Wire topology. This topology may have all 2Wire devices in parallel, or it may use a 2Wire Mux for 2Wire slaves associated on a per DFC basis, including access to a Drive Facing Connector I/O 2Wire interface (e.g., NVMe-MI). If a 2Wire Mux topology is used, the UBM FRU shall not be placed behind the 2Wire Mux. The Mux 2Wire Slave Address is formed by having the significant 2Wire slave address as 1110b followed by 3 bits indicated in the Mux 2Wire Slave Address field for addressing flexibility (i.e., 2Wire Mux Slave Address format is 1,1,1,0,A2,A1,A0,R/W#).

If 2Wire Device Arrangement field is set 0h (i.e., No Mux), then:

a. a 2Wire Mux is not present (See Figure 4-1);
b. the Mux 2Wire Slave Address field is not used;
c. the UBM Controller 2Wire Slave Address field (See 6.3.2.2.1) indicates the 2Wire Slave Address of the UBM Controller.
d. the DFC 2Wire interface is optional.

If 2Wire Device Arrangement field is set 1h (i.e., DFC 2Wire located behind the Mux), then:

a. a 2Wire Mux is present and in parallel with the UBM FRU and UBM Controller(s) (See Figure 5-1);
b. the Mux 2Wire Slave Address field is valid;
c. the UBM Controller 2Wire Slave Address field (See 6.3.2.2.1) indicates the 2Wire Slave Address of the UBM Controller(s);
d. the Mux Channel to communicate to the DFC 2Wire interface is equal to the DFC Status and Control Descriptor Index field (See 6.3.2.2.7).

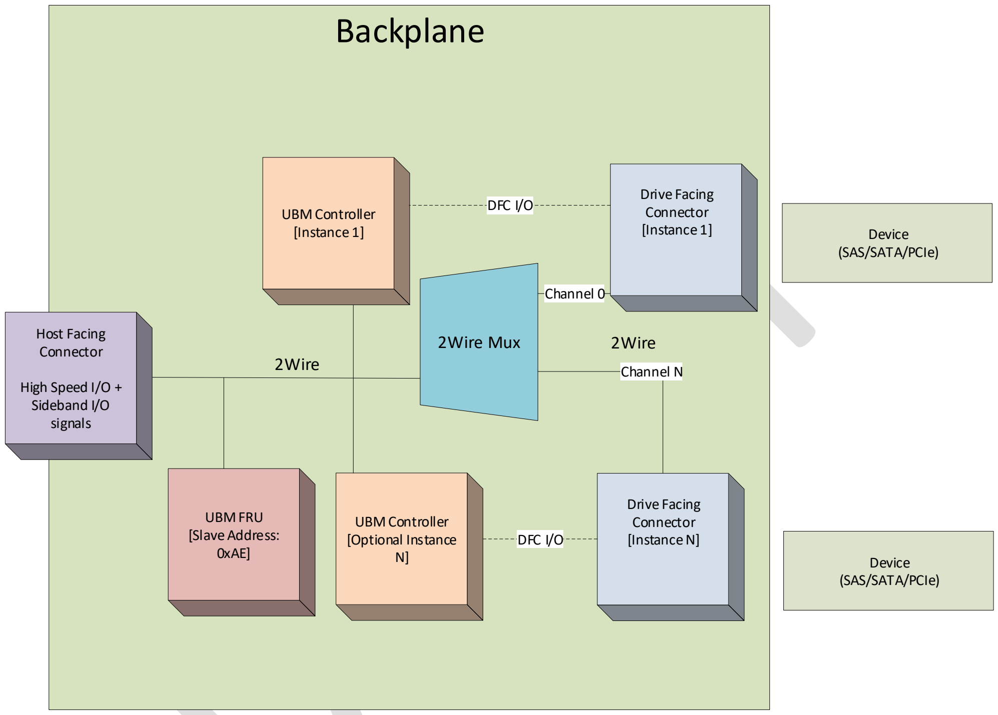

Backplane
UBM Controller [Instance 1]
DFC I/O
Drive Facing Connector [Instance 1]
Device (SAS/SATA/PCIe)
Channel 0
2Wire Mux
2Wire
Host Facing Connector
High Speed I/O + Sideband I/O signals
2Wire
Channel N
UBM FRU [Slave Address: 0xAE]
UBM Controller [Optional Instance N]
DFC I/O
Drive Facing Connector [Instance N]
Device (SAS/SATA/PCIe)

Figure 5-1 2Wire Device Arrangement with DFC 2Wire behind Mux

If 2Wire Device Arrangement field is set 3h (i.e., UBM Controllers and DFC 2Wire interface are located behind the Mux), then:

a. the 2Wire Mux is present and in parallel with the UBM FRU (See Figure 5-2);
b. the Mux 2Wire Slave Address field is valid;
c. the UBM Controller(s) and DFC 2Wire interface are in parallel behind the 2Wire Mux;
d. the UBM Controller 2Wire Slave Address (See 6.3.2.2.1) indicates the 2Wire Slave Address of the UBM Controller(s);
e. the Mux Channel to communicate to the DFC 2Wire interface is equal to the DFC Status and Control Descriptor Index field (See 6.3.2.2.7).

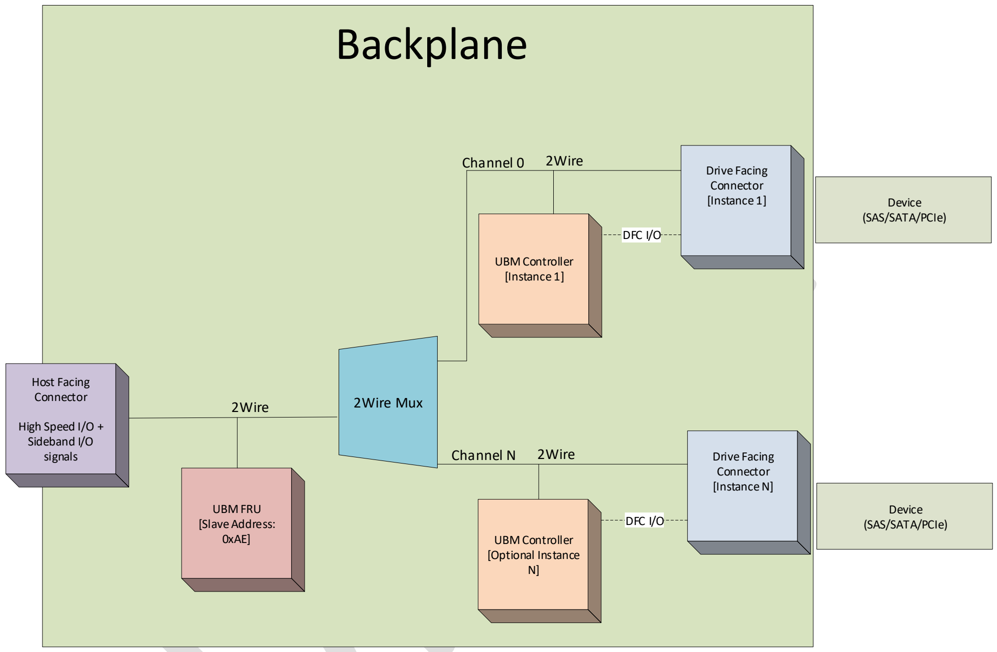

Backplane
Channel 0
2Wire
Drive Facing Connector [Instance 1]
Device (SAS/SATA/PCIe)
DFC I/O
UBM Controller [Instance 1]
Host Facing Connector
High Speed I/O + Sideband I/O signals
2Wire
2Wire Mux
Channel N
2Wire
Drive Facing Connector [Instance N]
Device (SAS/SATA/PCIe)
UBM FRU [Slave Address: 0xAE]
DFC I/O
UBM Controller [Optional Instance N]

Figure 5-2 2Wire Device Arrangement with UBM Controllers and DFC 2Wire behind Mux

5.6 UBM Controller Initialization Process

1. Initialize Output of DFC I/O signals
2. DFC PERSTA# and DFC PERSTB# signals are pulled LOW (i.e., PCIe Reset is asserted)
3. RefClk is disabled
4. PwrDIS signal is pulled LOW (i.e., Power is enabled)
5. Initialize Output and Bidirectional sideband I/O signals for HFC as defined in Table 5-1.
6. Set the UBM Controller Operational State to INITIALIZING (See Table 7-7).
7. Initialize UBM FRU if needed and set the UBM FRU Invalid field to 0 (i.e., Valid).
8. Setup and Enable UBM Controller 2Wire slave interface
9. Set the UBM Controller Operational State to READY for all 2Wire Slave interfaces.
10. Begin monitoring the DFC Inputs for changes (i.e., Presence or Loss of a Drive)

5.7 Host UBM Backplane Discovery Process

The Host uses the following process to discover UBM backplanes:

1. At System Power on, the Backplane UBM FRU and UBM Controller initialize and stabilize content. The CPRSNT# / CHANGE_DETECT# sideband I/O signal is driven LOW by the UBM Controller.
2. The HFC PERST# signal shall be driven LOW, until the Host RefClk, if any, has stabilized.
3. The Host samples BP_TYPE signal to determine if the backplane is representing SGPIO (LOW) or 2Wire communication (HIGH). If the BP_TYPE signal indicates 2Wire, then the Host shall proceed for detection of a UBM Backplane.
4. Host reads the UBM FRU (See Section 6.1).
5. The Host can proceed to the next step if the UBM FRU Invalid field is set to 0 (i.e., Valid).
6. The Host accesses the UBM FRU to obtain the IPMI FRU formatted content which describes the Backplane.
    a. The Host reads the UBM Overview Area content to determine the Number of Backplane DFC's, the Number of UBM Port Route Information Descriptors, and the Number of DFC Status and Control Descriptors.
    b. The Host reads the UBM Port Routing Information Descriptors to determine the DFC port mapping to HFC and the 2Wire address for one or more UBM Controllers.
7. The Host resolves which DFC Status and Control Descriptors are mapped to the Host Facing Connector.
8. The Host accesses the UBM FRU to determine UBM Controller Max Time Limit (See Section 6.3.1.2.3).
9. The Host attempts communication with the UBM Controllers specified in the UBM Port Routing Information Descriptors.
10. If the UBM Controller is unresponsive or indicating an Operational State other than READY, the Host re-attempts communication until the Max Time limit has been reached.
11. Upon successful UBM Controller communication and READY Operational State, the Host accesses the UBM Controller(s) to obtain:
    a. The backplane capabilities including:
        i. PCIe Reset expectations
        ii. RefClk expectations
    b. The Change Count field
    c. The Host Facing Connector Identity field to resolve the DFC Status and Control Descriptor Indexes to be accessed
    d. The DFC Status and Control Descriptors to obtain the type of the installed device.
12. The Host configures the high-speed port protocol and link and resets the device as defined in Section 5.16.
13. The Host writes the Change Count value read from the UBM Controller into the Change Count field. The UBM Controller acknowledges the Change Count write of the correct value by allowing the CHANGE_DETECT# signal to float HIGH.

5.8 CPRSNT# / CHANGE_DETECT# signal

UBM provides an optional interrupt mechanism via the CPRSNT# signal. SFF-9402 indicates this signal is mapped to CPRSNT# (Cable Present), which provides indication that a Quad PCIe drive has been installed and the host shall enable its RefClk, if any, for this device. To account for legacy applications, the UBM Controller indicates in the UBM FRU the definition of the CPRSNT# / CHANGE_DETECT# signal. If the CPRSNT# Legacy Mode is indicated, then CPRSNT# / CHANGE_DETECT# signal functions in the legacy Cable Present method until at such time a host writes a 0 (i.e., CHANGE_DETECT# interrupt operation) to the CPRSNT# Legacy Mode field. The UBM Controller shall indicate the change of the CPRSNT# Legacy Mode field via the Change Count field and the assertion of the CHANGE_DETECT# signal (i.e., LOW) until the Host handles the CHANGE_DETECT# signal (See Section 5.9). If multiple UBM Controllers are implemented, the Host shall configure the CPRSNT# Legacy Mode field in each UBM Controller identically.

5.9 CHANGE_DETECT# signal interrupt handling

If the CPRSNT# Legacy Mode feature (See Table 7-46) is set to 0 (i.e., CHANGE_DETECT# interrupt operation) and the CHANGE_DETECT# Interrupt Operation (See Table 7-43) is set 1 (i.e. CHANGE_DETECT# interrupt operation is supported by the UBM Controller), then the CHANGE_DETECT# signal indicates that the UBM Controller has detected a change that the host needs to be aware of. The Host can control the Change Count field incrementing (i.e., situations in which CHANGE_DETECT# signal is driven LOW) via masks found in the Features of the UBM Controller (See Section 7.2.12). The following process defines the expected host behavior when the CHANGE_DETECT# signal is asserted (i.e., LOW).

1. If the UBM Controller Operational State (See Section 5.20 and Section 7.2.1) does not indicate READY, then the host shall avoid further UBM Controller commands until the next assertion (i.e., LOW) of the CHANGE_DETECT# signal.
2. If the UBM Controller Operational State is READY, the Host:
    a. Reads the UBM Controller Change Count field,
    b. Writes each Descriptor Index associated with the Host Facing Connector,
    c. After writing the DFC Status and Control Descriptor Index, the DFC Status and Control Descriptor shall be read for each Descriptor Index.
3. The Host clears the CHANGE_DETECT# when all Descriptor Indexes associated to the Host Facing Connector have been examined by writing the current Change Count field back to the Change Count read at the beginning of this process.
4. If the UBM Last Command Status field is 05h (i.e., CHANGE COUNT DOES NOT MATCH), return to Step 1.
5. If the UBM Last Command Status field is 01h (i.e., SUCCESS), the Host advances to Step 6.
6. Steps 1 to 5 are repeated for each UBM Controller until the CHANGE_DETECT# signal becomes HIGH (i.e., all UBM Controllers have had their change counts serviced).

5.10 Host Facing Connector Identity

The Host Facing Connector Identity field indicates a unique value for each HFC within the same Backplane indicating the same Backplane Number field. The Host Facing Connector Identity field is used in Chassis Slot Mapping (See Section 5.10) and enables cable installation order independence.

5.11 Host Facing Connector Starting Lane

The Host Facing Connector Starting Lane field (See Section 6.3.2.2.6) indicates the high speed Tx/Rx differential signal lane assignment in the Host Facing Connector that is associated with a DFC port lane 0. The DFC lane mapping to HFC lane routing shall be in order to maintain the Host PCIe port ordering rules consistently through the cable, backplane and the DFC. Table 5-4 is an example of 2 DFC's routing to a single HFC. The HFC Starting Lane is associated to the DFC port Lane 0 that is routed through the backplane. DFC port lane 1 is adjacent to DFC port lane 0 in the HFC. The port route associated to DFC Status and Control Descriptor Index 0 is described in UBM Port Route Information Descriptor 0, while the port route associated to DFC Status and Control Descriptor Index 1 is described in UBM Port Route Information Descriptor 1. The HFC Starting Connector Lane for UBM Port Route Information Descriptor 0 is 0. The HFC Starting Connector Lane for UBM Port Route Information Descriptor 1 is 2.

Table 5-4 HFC Starting Lane Example of 2x2 DFC to 1 HFC

UBM PORT ROUTE INFORMATION DESCRIPTOR INDEX	HOST FACING CONNECTOR IDENTITY	HOST FACING CONNECTOR LANE	DFC STATUS AND CONTROL DESCRIPTOR INDEX	DFC PORT
0	0	0	0	Lane 0
0	0	1	0	Lane 1
1	0	2	1	Lane 0
1	0	3	1	Lane 1

5.12 Chassis Slot Mapping

The Host is responsible for mapping the UBM FRU and the associated UBM Controllers. As the Host processes the UBM Port Route Information Descriptors, the Host performs a chassis slot mapping process for the drive facing connectors in the backplanes in the chassis.

The Host creates an access map using this data set.

Table 5-5 Access Map to Find Actual Slot Location

MAPPING ELEMENT	ELEMENT LOCATION
Host 2Wire Port Number	Host
UBM Controller 2Wire Address	UBM Port Route Information Descriptor
Host Facing Connector Identity	UBM Controller & UBM Port Route Information Descriptor
Backplane Number and Backplane Type	UBM Controller
DFC Status and Control Descriptor Index	UBM Port Route Information Descriptor
Starting Slot	UBM Controller
Slot Offset	UBM Port Route Information Descriptor
Derived Actual Slot Location	

The Host 2Wire Port represents the internal mapping to the Hosts resources. It is necessary to use the correct 2Wire Port as the 2Wire Master for the 2Wire interface responsible for communicating with the UBM Controller.

The UBM Controller provides the Backplane Number and Backplane Type fields. The Backplane Number field shall be unique among all backplanes in the chassis. Multiple backplanes in the chassis shall be managed together using the same Backplane Type field value (e.g., To create a Redfish chassis resource or virtual SES resource).

The 2Wire interface from the Host communicates with the UBM Controller, upon which the Host Facing Connector Identity field of the backplane is determined. The Host Facing Connector Identity field shall be unique per HFC on the Backplane containing the same Backplane Number field. After determining the Host Facing Connector Identity the Host examines the UBM Port Route Information Area accessed during discovery to create a DFC Status and Control Descriptor Index list that corresponds to DFC Indexes routed through the Host Facing Connector.

The UBM Controller also provides the Starting Slot field which is added to the Slot Offset field from the UBM Port Route Information Area to resolve the Derived Actual Slot Location in the chassis. There shall be no duplicate Derived Actual Slot Location values within the same Backplane containing the same Backplane Number field. The Derived Actual Slot Location shall be unique among all Slots sharing the same Backplane Type field value (i.e., No duplicate Derived Actual Slots can be found among all backplanes in the same Backplane Type field value).

Figure 5-3 is an example of a single management resource instance of 16 uniquely Derived Actual Slots implemented across two backplanes.

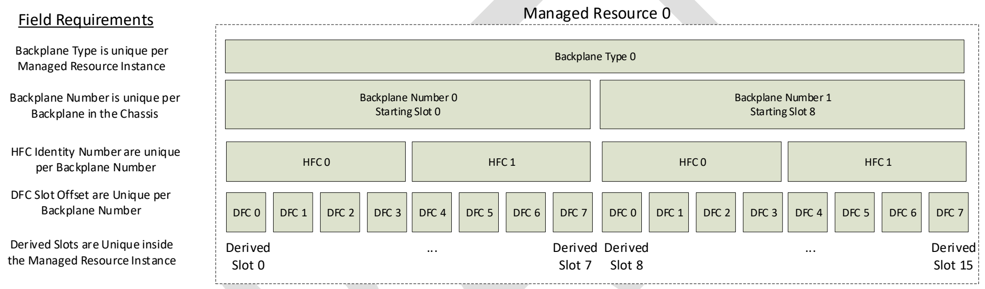

Field Requirements
Managed Resource 0
Backplane Type is unique per Managed Resource Instance
Backplane Type 0
Backplane Number is unique per Backplane in the Chassis
Backplane Number 0 Starting Slot 0
Backplane Number 1 Starting Slot 8
HFC Identity Number are unique per Backplane Number
HFC 0
HFC 1
HFC 0
HFC 1
DFC Slot Offset are Unique per Backplane Number
DFC 0
DFC 1
DFC 2
DFC 3
DFC 4
DFC 5
DFC 6
DFC 7
DFC 0
DFC 1
DFC 2
DFC 3
DFC 4
DFC 5
DFC 6
DFC 7
Derived Slots are Unique inside the Managed Resource Instance
Derived Slot 0
...
Derived Slot 7
Derived Slot 8
...
Derived Slot 15

Figure 5-3 Example of Multiple Backplanes Managed by One Managed Resource

Figure 5-4 is an example of two management resource instances of each with 8 uniquely Derived Actual Slots implemented across two backplanes. Due to uniquely specified Backplane Type fields the derived slot locations can be reused in Management Resource 1 instance when compared to Management Resource 0.

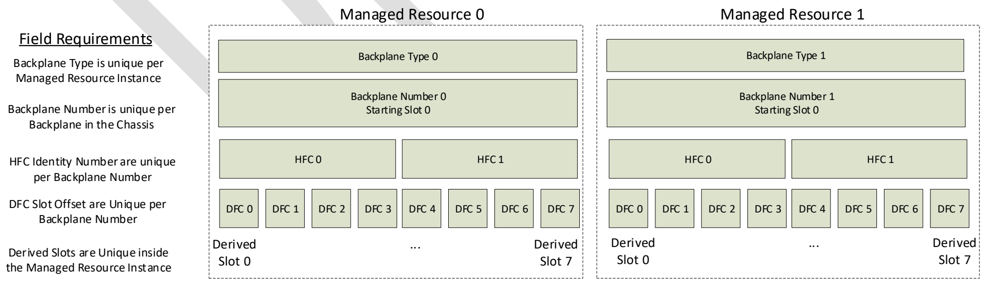

Field Requirements
Managed Resource 0
Managed Resource 1
Backplane Type is unique per Managed Resource Instance
Backplane Type 0
Backplane Type 1
Backplane Number is unique per Backplane in the Chassis
Backplane Number 0 Starting Slot 0
Backplane Number 1 Starting Slot 0
HFC Identity Number are unique per Backplane Number
HFC 0
HFC 1
HFC 0
HFC 1
DFC Slot Offset are Unique per Backplane Number
DFC 0
DFC 1
DFC 2
DFC 3
DFC 4
DFC 5
DFC 6
DFC 7
DFC 0
DFC 1
DFC 2
DFC 3
DFC 4
DFC 5
DFC 6
DFC 7
Derived Slots are Unique inside the Managed Resource Instance
Derived Slot 0
...
Derived Slot 7
Derived Slot 0
...
Derived Slot 7

Figure 5-4 Example of Multiple Backplanes Managed by Two Separate Managed Resources

5.13 LED State

The DFC Status and Control Descriptor contains the SES Array Device Slot element bytes, that defines LED States (IDENT, PRDFAIL, OK, etc..).

5.14 LED Pattern Behavior

LED pattern behavior is not defined by UBM, but may use the IBPI (SFF-8489) specification or OEM defined LED pattern behavior specification.

5.15 Drive Activity Behavior

Drive activity LED generation is out of the scope for UBM (SFF-TA-1005).

5.16 PCIe Clock Routing and PCIe Reset Control Management

The PCIe Reset Control bit (See 7.2.11) is used to control the port specific DFC PERST# signal (i.e., PERSTA# or PERSTB#) assertion and deassertion I/O timing. If RefClk is routed through the backplane, then PCIe Reset Control shall ensure the RefClk is forwarded to the Drive Facing Connector before port specific DFC PERST# signal is deasserted per the PCIe timing specification.

The Clock Routing bit (See 7.2.11) indicates if RefClk is routed from the HFC to the devices. (i.e., Support for PCIe devices that do not support SRIS/SRNS).

If the Clock Routing bit is set to 0 (i.e., Clock routing is not present), and PCIe Reset Control bit is set to 0 (i.e., PCIe Reset Control is not supported) then no Host interaction is required for the UBM Controller (e.g., a SAS/SATA Only Backplane).

If the Clock Routing bit is set to 0 (i.e., No clock routing) and the PCIe Reset Control bit is set to 1 (i.e., PCIe Reset Control is supported), then:

a. If the DFC PERST# Management Override Supported field (See 7.2.11) is set to 0h (i.e., Override is not supported), then No Host interaction is required for the UBM Controller to manage the port specific DFC PERST# signal (Note: PCIe Reset field of 0h does not guarantee that the UBM controller has released the DFC PERST# and the device is fully functional);
b. If the DFC PERST# Management Override Supported field is set to 1h (i.e., Override is supported) and the DFC PERST# Management Override field (See 7.2.12) is set to either 0h or 2h (i.e., No Override or DFC PERST# automatically released upon install), then No Host interaction is required for the UBM Controller to manage the port specific DFC PERST# signal (Note: PCIe Reset field of 0h does not guarantee that the UBM controller has released the DFC PERST# and the device is fully functional);
c. If the DFC PERST# Management Override Supported field is set to 0h (i.e., Override is not supported) and no device is present, then the UBM Controller shall assert the port specific DFC PERST# signal and set the PCIe Reset field to 0h (i.e., No Operation);
d. If the DFC PERST# Management Override Supported field is set to 1h (i.e., Override is supported) and the DFC PERST# Management Override field is set to either 0h or 2h (i.e., No Override or DFC PERST# automatically released upon install), then the UBM Controller shall assert the port specific DFC PERST# signal and set the PCIe Reset field to 0h (i.e., No Operation);
e. If the DFC PERST# Management Override Supported field is set to 1h (i.e., Override is supported), the DFC PERST# Management Override field is set to 1h (i.e., DFC PERST# is Managed), and a device is present, then the UBM Controller shall keep the port specific DFC PERST# signal asserted and set the PCIe Reset field to 2h (i.e., LOW);
f. If the DFC PERST# Management Override Supported field is set to 1h (i.e., Override is supported), the DFC PERST# Management Override field is set to 1h (i.e., DFC PERST# is Managed), and no device is present, then the UBM Controller shall assert the port specific DFC PERST# signal and set the PCIe Reset field to 2h (i.e., DFC PERST# signal is held LOW);
g. If the PCIe Reset field is set to 2h (See 7.2.15), then the UBM Controller shall assert the port specific DFC PERST# signal (i.e., LOW);
h. If the PCIe Reset field is set to 1h (i.e., Initiate PCIe Reset Sequencing), then the UBM Controller shall deassert the port specific DFC PERST# signal per PCIe timing specification and set the PCIe Reset field to 0h (i.e. No Operation).
i. If the PCIe Reset field when read indicates 0h and a device is present, then the port specific DFC PERST# signal is deasserted (i.e., HIGH);

If the Clock Routing bit is set to 1 (i.e., Clock routing is present) and the PCIe Reset Control bit is set to 1 (i.e., PCIe Reset Control is supported), then:

a. If the DFC PERST# Management Override Supported field (See 7.2.11) is set to 0h (i.e., Override is not supported) and no device is present, then the UBM Controller shall assert the port specific DFC PERST# signal and set the PCIe Reset field to 2h (i.e., LOW);
b. If the DFC PERST# Management Override Supported field is set to 1h (i.e., Override is supported), the DFC PERST# Management Override field (See 7.2.12) is set to either 0h or 1h (i.e., No Override or DFC PERST# is Managed upon Install), and no device is present, then the UBM Controller shall assert the port specific DFC PERST# signal and set the PCIe Reset field to 2h (i.e., LOW);
c. If the DFC PERST# Management Override Supported field is set to 0h (i.e., Override is not supported) and a device is present, then the UBM Controller shall keep the port specific DFC PERST# signal asserted and set the PCIe Reset field to 2h (i.e., LOW);
d. If the DFC PERST# Management Override Supported field is set to 1h (i.e., Override is supported), the DFC PERST# Management Override field is set to either 0h or 1h (i.e., No Override or DFC PERST# is Managed upon Install), and a device is present, then the UBM Controller shall keep the port specific DFC PERST# signal asserted and set the PCIe Reset field to 2h (i.e., LOW);
e. If the DFC PERST# Management Override Supported field is set to 1h (i.e., Override is supported) and the DFC PERST# Management Override field is set to 2h (i.e., DFC PERST# automatically released upon install), then No Host interaction is required for the UBM Controller to manage port specific DFC PERST# signal (Note: PCIe Reset field of 0h does not guarantee that the UBM controller has released the DFC PERST# and the device is fully functional);
f. If the DFC PERST# Management Override Supported field is set to 1h (i.e., Override is supported), the DFC PERST# Management Override field is set to 2h (i.e., DFC PERST# automatically released upon install), and no device is present, then the UBM Controller shall assert the port specific DFC PERST# signal and set the PCIe Reset field to 0h (i.e., No Operation);
g. If the PCIe Reset field is set to 2h (See 7.2.15), then the UBM Controller shall assert the port specific DFC PERST# signal (i.e., LOW);
h. If the PCIe Reset field is set to 1h (i.e., Initiate PCIe Reset Sequencing) then the UBM Controller shall deassert the port specific DFC PERST# signal per PCIe timing specification and set the PCIe Reset field to 0h (i.e., No Operation).
i. If the PCIe Reset field when read indicates 0h and a device is present, then the port specific DFC PERST# signal is deasserted (i.e., HIGH);

If the Clock Routing bit is set to 1 (i.e., Clock routing is present) and the PCIe Reset Control bit is set to 0 (i.e., PCIe Reset Control not supported) then:

a. Management of the port specific DFC PERST# signal is vendor specific; and
b. Host RefClk stability is vendor specific.

If the DFC PERST# Management Override Supported field is set to 1h (i.e., Override is supported) and the Host transitions the DFC PERST# Management Override field from 0h to 2h (i.e., DFC PERST# automatically released upon install), then the UBM Controller shall deassert all DFC PERST# signals per PCIe timing specification for each PCI Reset field set to 2h and set the PCIe Reset field to 0h (i.e., No Operation).

Note: Transitioning a backplane to automatically release RefClk dependent DFC's PERST# signals requires the Host and Backplane to ensure the RefClk is stable before DFC PERST# deassertion. Support of automatic management of DFC PERST# for RefClk systems reduces the Host management overhead required for Hotplug events. A backplane, on power on, should not default with the DFC PERST# Management Override field set to 2h (i.e., Automatic release upon install) due to the timing requirements associated with ensuring the RefClk is valid and stable. After completing UBM Discovery, the Host may set the DFC PERST# Management Override field to 2h to allow the UBM Controller to automatically manage DFC PERST# signal deassertions when drives are inserted.

Table 5-6 summarizes the PCIe Clock Routing and PCIe Reset Control management options, as well as the behavior of the backplane in relationship to the HFC PERST# signal and the PCIe Reset field defined in this specification.

Table 5-6 PCIe Clock Routing And PCIe Reset Control Management (No DFC PERST# Management Override)

USE CASE	PCIE CLOCK ROUTING	PCIE RESET CONTROL	HFC PERST# SIGNAL	PCI RESET FIELD	DESCRIPTION
1	0	0	X	Xh	Device reset is managed by a method external to the UBM Controller. (e.g., SAS/SATA only backplane)
2	0	1	0 (i.e., LOW)	Xh	The UBM Controller asserts (i.e., LOW) all port specific DFC PERST# signals corresponding to the HFC.
2a			1 (i.e., HIGH)	0h	The UBM Controller performs port specific DFC PERST# signal deassertion after system power up and detection of a device installed.
2b				1h	The UBM Controller performs the requested port specific DFC PERST# signal deassertion per PCIe specification timings and then sets the PCIe Reset field to 0h when completed (See Section 7.2.15).
2c				2h	The UBM Controller asserts the port specific DFC PERST# signal (i.e., LOW).
3	1	1	0 (i.e., LOW)	Xh	The UBM Controller asserts all port specific DFC PERST# signals corresponding to the HFC.
3a			1 (i.e., HIGH)	0h	Port specific DFC PERST# signal and RefClk are controlled by the UBM Controller. After power up, port specific DFC PERST# signal is not released until the Host initiates the PCIe Reset sequence (See Section 7.2.15).
3b				1h	The UBM Controller performs the requested port specific DFC PERST# signal deassertion per PCIe specification timings and then sets the PCIe Reset field to 0h when completed (See Section 7.2.15).
3c				2h	The UBM Controller asserts the port specific DFC PERST# signal (i.e., LOW).
4	1	0	0 (i.e., LOW)	Xh	All port specific DFC PERST# signals are asserted (i.e., LOW) that correspond to the HFC.
4a			1 (i.e., HIGH)	Xh	Vendor specific

Notes:
Use Case 1:
Backplane supports SAS and SATA devices only.
PCIe devices are not supported by the backplane.
Use Case 2, Case 2a, Case 2b, Case 2c:
Backplane supports SAS, SATA, and PCIe SRIS/SRNS devices.
Use Case 3, Case 3a, Case 3b, Case 3c:
Backplane supports SAS, SATA and PCIe devices.

Table 5-7 summarizes the PCIe Clock Routing and PCIe Reset Control management options, as well as the behavior of the backplane in relationship to the HFC PERST# signal and the PCIe Reset field defined in this specification when the DFC PERST# Management Override is set to 1h (i.e., Managed upon install) and DFC PERST# Management Override Supported is 1h (i.e., Override supported).

Table 5-7 PCIe Clock Routing And PCIe Reset Control Management (DFC PERST# Management Override Set to 1h and override supported)

USE CASE	PCIE CLOCK ROUTING	PCIE RESET CONTROL	HFC PERST# SIGNAL	PCI RESET FIELD	DESCRIPTION
1	0	0	X	Xh	Device reset is managed by a method external to the UBM Controller. (e.g., SAS/SATA only backplane)
2	0	1	0 (i.e., LOW)	Xh	The UBM Controller asserts (i.e., LOW) all port specific DFC PERST# signals corresponding to the HFC.
2a			1 (i.e., HIGH)	0h	Port specific DFC PERST# signal is controlled by the UBM Controller. After power up, port specific DFC PERST# signal is not released until the Host initiates the PCIe Reset sequence (See Section 7.2.15).
2b				1h	The UBM Controller performs the requested port specific DFC PERST# signal deassertion per PCIe specification timings and then sets the PCIe Reset field to 0h when completed (See Section 7.2.15).
2c				2h	The UBM Controller asserts the port specific DFC PERST# signal (i.e., LOW).
3	1	1	0 (i.e., LOW)	Xh	The UBM Controller asserts all port specific DFC PERST# signals corresponding to the HFC.
3a			1 (i.e., HIGH)	0h	Port specific DFC PERST# signal and RefClk are controlled by the UBM Controller. After power up, port specific DFC PERST# signal is not released until the Host initiates the PCIe Reset sequence (See Section 7.2.15).
3b				1h	The UBM Controller performs the requested port specific DFC PERST# signal deassertion per PCIe specification timings and then sets the PCIe Reset field to 0h when completed (See Section 7.2.15).
3c				2h	The UBM Controller asserts the port specific DFC PERST# signal (i.e., LOW).
4	1	0	0 (i.e., LOW)	Xh	All port specific DFC PERST# signals are asserted (i.e., LOW) that correspond to the HFC.
4a			1 (i.e., HIGH)	Xh	Vendor specific

Notes:
Use Case 1:
Backplane supports SAS and SATA devices only.
PCIe devices are not supported by the backplane.
Use Case 2, Case 2a, Case 2b, Case 2c:
Backplane supports SAS, SATA, and PCIe SRIS/SRNS devices.
Use Case 3, Case 3a, Case 3b, Case 3c:
Backplane supports SAS, SATA and PCIe devices.

Table 5-8 summarizes the PCIe Clock Routing and PCIe Reset Control management options, as well as the behavior of the backplane in relationship to the HFC PERST# signal and the PCIe Reset field defined in this specification when the DFC PERST# Management Override is set to 2h (i.e., DFC PERST# automatically released upon install) and DFC PERST# Management Override Supported is 1h (i.e., Override supported).

Table 5-8 PCIe Clock Routing and PCIe Reset Control Management (DFC PERST# Management set to 2h and override supported)

USE CASE	PCIE CLOCK ROUTIN G	PCIE RESET CONTRO L	HFC PERST# SIGNAL	PCI RESET FIELD	DESCRIPTION
1	0	0	X	Xh	Device reset is managed by a method external to the UBM Controller. (e.g., SAS/SATA only backplane)
2	0	1	0 (i.e., LOW)	Xh	The UBM Controller asserts (i.e., LOW) all port specific DFC PERST# signals corresponding to the HFC.
2a			1 (i.e., HIGH)	0h	The UBM Controller performs port specific DFC PERST# signal deassertion after detection of a device installed.
2b				1h	The UBM Controller performs the requested port specific DFC PERST# signal deassertion per PCIe specification timings and then sets the PCIe Reset field to 0h when completed (See Section 7.2.15).
2c				2h	The UBM Controller asserts the port specific DFC PERST# signal (i.e., LOW).
3	1	1	0 (i.e., LOW)	Xh	The UBM Controller asserts all port specific DFC PERST# signals corresponding to the HFC.
3a			1 (i.e., HIGH)	0h	The UBM Controller performs port specific DFC PERST# signal deassertion after detection of a device installed.
3b				1h	The UBM Controller performs the requested port specific DFC PERST# signal deassertion per PCIe specification timings and then sets the PCIe Reset field to 0h when completed (See Section 7.2.15).
3c				2h	The UBM Controller asserts the port specific DFC PERST# signal (i.e., LOW).
4	1	0	0 (i.e., LOW)	Xh	All port specific DFC PERST# signals are asserted (i.e., LOW) that correspond to the HFC.
4a			1 (i.e., HIGH)	Xh	Vendor specific
Notes: Use Case 1: Backplane supports SAS and SATA devices only. PCIe devices are not supported by the backplane. Use Case 2, Case 2a, Case 2b, Case 2c: Backplane supports SAS, SATA, and PCIe SRIS/SRNS devices. Use Case 3, Case 3a, Case 3b, Case 3c: Backplane supports SAS, SATA and PCIe devices.					

5.17 DFC Status and Control Descriptor

The DFC Status and Control Descriptor (See Section 7.2.15) provide the following capabilities:

- Indicates if a device is installed
- Indicates the protocol of an installed device
- If the device in the drive facing connector is supported
- Control of LED State and PwrDIS signal via the SES Array Device Slot Element (See SES-4 Specification)
- Requesting PCIe Reset sequence for the device

5.18 Bifurcation Port

Bifurcation allows the dividing of the host facing connector into equal size port widths. The UBM Port Route Information Descriptors provide the static map between the DFCs to HFCs. The Host uses this map to determine the link width and lane assignments within the HFC. The Bifurcate Port field in the DFC Status and Control Descriptors allows the UBM Controller to instruct the Host that the DFC to HFC link width route shall be divided by 2. This is useful in scenarios where the cable attached is no longer a direct port mapping but instead the cable design has routed half of the link width assignments from the HFC to the Host. Cable Detection of these scenarios is Vendor Specific.

5.19 UBM Port Route Information Descriptors

UBM Port Route Information Descriptors (See Section 0) indicate various information about the Drive Facing Connectors on a port basis. Drive Facing Connectors may support one or more ports. It also provides the mapping of the Drive Facing Connectors to Host Facing Connectors. This mapping includes details relating to domain and the Drive Facing Connector port type routing (Converged or Segregated). A converged port allows multiple protocols to communicate over the same high-speed port. A segregated port represents the PCIe port segment in the SFF-8639 connector.

Table 5-9 describes port usages for backplanes with SFF-8639 connectors.

Table 5-9 SFF-8639 Connector Port Usages

PORT USAGE	UBM PORT ROUTE INFORMATION DESCRIPTOR INSTANCE	LANES DESCRIBED BY UBM PORT ROUTE INFORMATION DESCRIPTOR	DOMAIN	CONVERGED / SEGREGATED
SAS only (Dual Port)	0	SAS 0 (SAS 2)	Primary	Converged
	1	SAS 1 (SAS 3) (Optional)	Secondary	Converged
PCIe only	0	PCIe 0/1/2/3	Primary	Segregated
Dual Port PCIe only	0	PCIe 0/1	Primary	Segregated
	1	PCIe 2/3	Secondary	Segregated
Segregated	0	PCIe 0/1/2/3	Primary	Segregated
	1	SAS 0	Primary	Converged
	2	SAS 1 (Optional)	Secondary	Converged
Segregated x2 Dual Port	0	PCIe 0/1	Primary	Segregated
	1	PCIe 2/3	Secondary	Segregated
	2	SAS 0	Primary	Converged
	3	SAS 1 (optional)	Secondary	Converged
Segregated x1 Dual Port (Not Practical)	0	PCIe 0	Primary	Segregated
	1	PCIe 2	Secondary	Segregated
	2	SAS 0 (SAS 2)	Primary	Converged
	3	SAS 1 (SAS 3) (optional)	Secondary	Converged
Note: SAS 2 and SAS 3 Ports are routed when the Dual Port bit is set in the Capabilities (See Section 7.2.11)				

Table 5-10 describes port usages for backplanes with SFF-TA-1001 connectors.

Table 5-10 SFF-TA-1001 Connector Port Usages

PORT USAGE	UBM PORT ROUTE INFORMATION DESCRIPTOR INSTANCE	LANES DESCRIBED BY UBM PORT ROUTE INFORMATION DESCRIPTOR	DOMAIN	CONVERGED / SEGREGATED
SAS/SATA and PCIe	0	x4 (any protocol)	Primary	Converged
SAS/SATA and PCIe Dual Port	0	x2 (any protocol)	Primary	Converged
	1	x2 (any protocol)	Secondary	Converged

5.20 UBM Controller Operational State

The UBM Controller indicates an Operational State field (See Section 7.2.1) to the Host. The UBM Controller must provide a valid response to the Operational State command. If the Operational State is INITIALIZING, BUSY, or REDUCED FUNCTIONALITY, responses to other commands or information in the UBM Controller may not be valid. The Host polls at low frequency or utilizes the CHANGE_DETECT# signal as an interrupt (if enabled) to wait for the UBM Controller Operational State to become READY. REDUCED FUNCTIONALITY Operational State shall be indicated whenever the UBM Controller has entered into Programmable Update Mode (See Section 7.2.4).

While the UBM Controller Operational State indicates REDUCED FUNCTIONALITY, the UBM Controller shall continue to manage Drive Facing Connector I/O in the case of device removal. Upon device insertion, the DFC I/O handling shall be delayed until the UBM Controller Operational State is READY.

Note: Device Removal I/O handling entails returning the port specific DFC PERST# signal to asserted (i.e., LOW) and RefClk to disabled when a drive is not present in the DFC. Device Insertion will keep these I/O signals in these states until such time that UBM Controller exits REDUCED FUNCTIONALITY Operational State.

Note: I/O signal state should be passed between Programmable Update Mode and the UBM Controller Operational State of READY, if the backplane is intended to support programmable updates while the devices are online.

Note: The UBM FRU contains the amount of time the Host waits for the UBM Controller to initialize upon request to exit the REDUCED FUNCTIONALITY Operational State (See Section 6.3.1.2.3). The host uses this information to determine if the update has been successful.

5.21 UBM Controller Image Update

The UBM Controller may support a UBM Controller Image Update. This image is vendor specific data (e.g., microcontroller firmware) programmed into non-volatile storage. The UBM Controller supports the UBM Controller Image Update process if the Programmable Update Modes field indicates a 01h (i.e., Programming Update supported while Devices remain online) or a 02h (i.e., Programming Update supported while Devices are offline). The UBM Controller Image Update process utilizes Subcommands that are defined by the Programmable Mode Data Transfer Command (See 7.2.6).

The UBM Controller Image Update process is:

a. The Host issues the Enter Programmable Update Mode Command with the defined Unlock Sequence fields and sets the Transfer to Programmable Update Mode field to 1h (i.e., Enter Programmable Update Mode).
b. The Host issues the Get Non-Volatile Storage Geometry Subcommand (See 7.2.6.2) to determine the storage sector quantity and size of the storage sectors.
c. The Host issues one or more Erase Subcommands (See 7.2.6.3) to erase the non-volatile storage.
d. The Host issues Erase Status Subcommands (See 7.2.6.4) to verify the status of the Erase Subcommands.
e. The Host issues one or more Program Subcommands (See 7.2.6.5) as necessary to program the Non-Volatile Storage.
f. The Host issues Program Status Subcommands (See 7.2.6.6) to verify the status of the Program Subcommands.
g. The Host may issue one or more Verify Subcommands (See 7.2.6.7) to check for successful programming.
h. The Host issues Verify Status Subcommands (See 7.2.6.8) to verify the status of the Verify Subcommands.
i. The Host may verify the entire non-volatile UBM Controller Image by issuing the Verify Image Subcommand (See 7.2.6.9) followed by the Verify Image Status Subcommand (See 7.2.6.10).
j. The Host issues the Set Active Image Subcommand (See 7.2.6.11).
k. The Host issues the Active Image Status Subcommand to verify the image has been updated successfully for activation (See 7.2.6.12).
l. The Host issues the Exit Programmable Update Mode Command (See 7.2.7).

6. UBM FRU

The UBM FRU on the backplane is responsible for reporting static backplane information.

The UBM FRU is a 256 byte read-only NVRAM with IPMI FRU formatted content. The IPMI FRU format consists of an IPMI common header, which provides the starting offset to the Multi-Record area which stores the UBM Overview Area content and the UBM Port Route Information Area content. The UBM specification does not preclude Board Area or Product Area records in the UBM FRU, but the NVRAM size must be considered. Figure 6-1 shows the overall format of the UBM FRU.

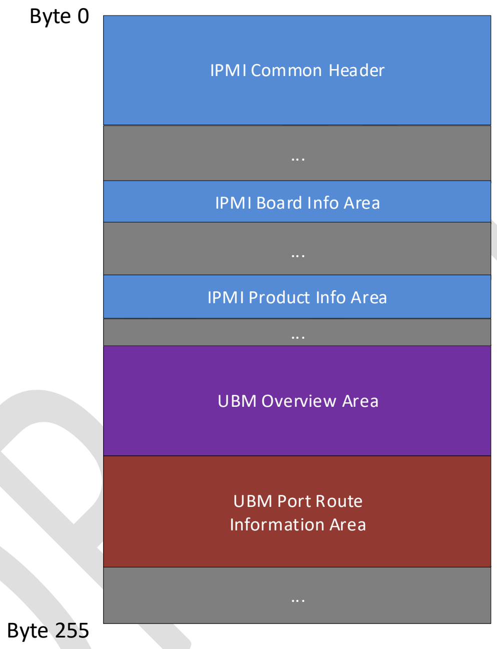


Figure 6-1 UBM FRU Format

6.1 UBM FRU 2Wire Protocol

The UBM FRU uses a single byte 2Wire addressing. The UBM FRU is addressed using an 8-bit 2Wire address of 0xAE. Single or multi-byte transactions shall be supported by this device. The UBM FRU formatted content is protected by checksums in the content structure. The host should validate the checksums and retry as appropriate to ensure the data transferred is valid. Table 6-1 provides information to be able to read the subsequent 2Wire Transaction figures.

Note: If a single UBM Controller is implemented, then each UBM Port Route Information Descriptor will contain the same 2Wire Slave address. If Multiple UBM Controllers are implemented, then each UBM Controller shall have a unique 2Wire Slave address.

Table 6-1 UBM FRU 2Wire Transaction Legend

LEGEND	DESCRIPTION
	Driven by 2Wire Master
▨	Driven by 2Wire Slave
W	Driven LOW by 2Wire Master to indicate Write Phase
R	Driven HIGH by 2Wire Master to indicate Read Phase

A UBM FRU Read transaction consists of the 2Wire Master writing the Slave Address and the Address Byte, and then the 2Wire Master continues the transaction by reading one or more data bytes from the 2Wire Slave.

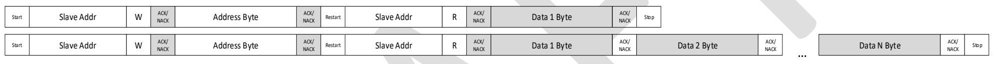


Figure 6-2 UBM FRU 2Wire Read Transaction

A UBM FRU Write transaction consists of the 2Wire Master writing the Slave Address, the Address Byte, and the one or more data bytes to the 2Wire Slave.

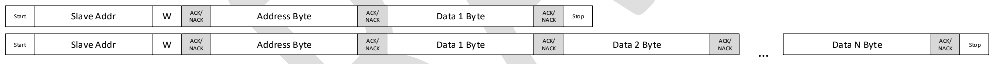


Figure 6-3 UBM FRU 2Wire Write Transaction

6.2 IPMI Defined Data

The IPMI Common Header, Board Info Area and Product Info Area are defined in the IPMI Platform Management FRU Information Storage Definition specification.

6.3 MultiRecords

The MultiRecord Area shall start on an 8 byte boundary of the address map as defined by the Common Header MultiRecord Area Starting Offset field. The UBM Overview Area and UBM Port Route Information Area reside in the MultiRecord Area of the UBM FRU.

6.3.1 UBM Overview Area

The UBM Overview Area is defined in Table 6-2.

Table 6-2 UBM Overview Area

RECORD LABEL	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Header	+0	Record Type ID = 0xA0							
Header	+1	End of List = 0	Reserved			Record Format = 2h			
Header	+2	Record Length = 0x0B							
Header	+3	Record Checksum = 0xXX							
Header	+4	Header Checksum = 0xYY							
Data 0	+5	UBM Specification Version							
Data 1	+6	UBM Controller 2Wire Max Byte Count			2Wire Mux Address			2Wire Device Arrangement	
Data 2	+7	UBM Controller Max Time Limit							UBM FRU Invalid
Data 3	+8	Default UBM Controller Features							
Data 4	+9								
Data 5	+10	Number of DFC Status and Control Descriptors							
Data 6	+11	Number of UBM Port Route Information Descriptors							
Data 7	+12	Number of Backplane DFC							
Data 8	+13	Maximum Power per DFC							
Data 9	+14	~~Reserved~~2Wire Mux Description							
		Valid	Mux Type	Reserved		Enable Bit Location		Mux Channel Count	
Data 10	+15	Reserved							

6.3.1.1 Header

The UBM Overview Area Header is described in the IPMI MultiRecord Header format. The Record Type ID field shall be set to 0xA0.

6.3.1.2 Data

The Data segment of the UBM Overview Area provides backplane information.

6.3.1.2.1 Data Byte 0 Definition

Table 6-3 UBM Overview Area: Data Byte 0 Definition

Bits	R/W	UBM Overview Area Data Byte 0 Definition
7:0	R	UBM Specification Version - defined in Table 7-12.

6.3.1.2.2 Data Byte 1 Definition

Table 6-4 UBM Overview Area: Data Byte 1 Definition

Bits	R/W	UBM Overview Area Data Byte 1 Definition
7:5	R	UBM Controller 2Wire Max Byte Count– indicates the maximum number of bytes that can exist between the 2Wire Slave Address and Stop/Restart of a 2Wire transaction. 0h = No Limit 1h = 16 bytes 2h = 32 bytes 3h = 64 bytes 4h = 128 bytes 5h = 256 bytes 6h-7h = Reserved
4:2	R	Mux 2Wire Slave Address – indicates bits [3:1] of the 8-bit Address for a 2Wire Mux. See Section 5.5 for 2Wire Device Topology concept.
1:0	R	2Wire Device Arrangement – indicates the 2Wire device arrangement (See Section 5.5) 0h = No Mux routed on HFC 2Wire interface 1h = DFC 2Wire interface behind Mux 2h = Reserved 3h = UBM Controller(s) and DFC 2Wire interface located behind Mux

6.3.1.2.3 Data Byte 2 Definition

Table 6-5 UBM Overview Area: Data Byte 2 Definition

Bits	R/W	UBM Overview Area Data Byte 2 Definition
7:1	R	UBM Controller Max Time Limit – provides the maximum amount of time in seconds that the Host shall wait for the UBM Controller to initialize and the UBM Controller Operational State field indicates READY.
0	R	UBM FRU Invalid – Indicates the validity of the UBM FRU data. Once the UBM FRU data is valid, it shall not become invalid until a subsequent power cycle. The Host shall wait a maximum 10 seconds for the UBM FRU Invalid to become Valid (i.e. Set to 0h). 0h = Valid 1h = Invalid

6.3.1.2.4 Data Byte 3 and Data Byte 4 Definition

The Default Features field indicates the default values for the UBM Controller Feature field as defined in Section 7.2.12.

6.3.1.2.5 Data Byte 5 Definition

Table 6-6 UBM Overview Area: Data Byte 5 Definition

Bits	R/W	UBM Overview Area Data Byte 5 Definition
7:0	R	Number of DFC Status and Control Descriptors – indicates the number of DFC Status and Control Descriptors supported by the UBM Controllers (See Section 7.2.15).

6.3.1.2.6 Data Byte 6 Definition

Table 6-7 UBM Overview Area: Data Byte 6 Definition

Bits	R/W	UBM Overview Area Data Byte 6 Definition
7:0	R	Number of UBM Port Route Information Descriptors – indicates the number of Port Route Information Descriptors (See Section 0).

6.3.1.2.7 Data Byte 7 Definition

Table 6-8 UBM Overview Area: Data Byte 7 Definition

Bits	R/W	UBM Overview Area Data Byte 7 Definition
7:0	R	Number of Backplane DFC – indicates the number of Drive Facing Connectors on the backplane.

6.3.1.2.8 Data Byte 8 Definition

Table 6-9 UBM Overview Area: Data Byte 8 Definition

Bits	R/W	UBM Overview Area Data Byte 8 Definition
7:0	R	Maximum Power per DFC – indicates the maximum supported power in watts for each DFC. A zero value in this field indicates there is no power limit.

6.3.1.2.9 Data Byte 9 Definition

Table 6-10 UBM Overview Area: Data Byte 9 Definition

Bits	R/W	UBM Overview Area Data Byte 9 Definition
7	R	2Wire Mux Description Valid 0h = 2Wire Mux Description Byte is not Valid 1h = 2Wire Mux Description Byte is Valid
6	R	2Wire Mux Enable Channel Selection Method (i.e., Mux Type) 0h = Channels are selected using bit location (E.g., PCA9543,PCA9546, PCA9548) 1h = Channels are selected using enable bit and channel byte (E.g., PCA9540, PCA9542, PCA9544, PCA9547)
5:4	R	Reserved
3:2	R	2Wire Mux Enable Bit Location 0h = Mux Enable is not applicable 1h = Reserved 2h = Mux Enable located at Bit 2 of Channel Select Byte (E.g., PCA9540, PCA9542, PCA9544) 3h = Mux Enable located at Bit 3 of Channel Select Byte (E.g., PCA9547)
1:0	R	2Wire Mux Channel Count 0h = No Mux implemented 1h = 2 Channel Mux implemented 2h = 4 Channel Mux implemented 3h = 8 Channel Mux implemented

~~6.3.1.2.9~~6.3.1.2.10 ~~and~~ Data Byte 10 Definition

Data Byte ~~9 and~~ 10 ~~are~~ is Reserved.

6.3.2 UBM Port Route Information Area

The UBM Port Route Information Area is defined in Table 6-11. The Data section of this record provides an array of UBM Port Route Information Descriptors. The number of Port Route Information Descriptors is indicated in the Number of Port Route Information Descriptor field.

6.3.2.1 Header

The UBM Port Route Information Area Header is described in the IPMI MultiRecord Header format. The Record Type ID field shall be set to 0xA1. The other fields in the UBM Port Route Information Area Header shall be set as defined in Table 6-11.

Table 6-11 UBM Port Route Information Area

RECORD LABEL	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Header	+0	Record Type ID = 0xA1							
Header	+1	End of List = 1	Reserved			Record Format = 2h			
Header	+2	Record Length = 0xNN							
Header	+3	Record Checksum = 0xXX							
Header	+4	Header Checksum = 0xYY							
Data	+5 To +11	UBM Port Route Information Descriptor 0							
…	…	…							
Data	Computed	UBM Port Route Information Descriptor N-1							

6.3.2.2 Data

The Data segment of the UBM Port Route Information Area consists of an array of Port Route Information Descriptors as defined in Table 6-12.

Table 6-12 UBM Port Route Information Descriptor

OFFSET \ BYTE	7	6	5	4	3	2	1	0
+0	UBM Controller 2Wire Slave Address							UBM Controller Type
+1	DFC Status and Control Descriptor Index							
+2	DFC Empty	Reserved	Quad PCIe Support	SAS/ SATA Support	Gen-Z Support	Reserved	SFF-TA-1001 PCIe Support	Reserved
+3	Domain	Port Type	Reserved		Link Width			
+4	Max SAS Link Rate Supported			Max PCIe Link Rate Supported			Max SATA Link Rate Supported	
+5	Host Facing Connector Identity				Host Facing Connector Starting Lane			
+6	Slot Offset							

6.3.2.2.1 Data Byte 0 Definition

Table 6-13 Port Route Information: Data Byte 0 Definition

Bits	R/W	UBM Port Route Information Descriptor Byte 0 Definition
7:1	R	UBM Controller 2Wire Slave Address – the upper 7-bits of the 8-bit 2Wire Slave Address.
0	R	UBM Controller Type 0h = UBM Controller is defined by this specification 1h = UBM Controller is Vendor Specific

6.3.2.2.2 Data Byte 1 Definition

Table 6-14 Port Route Information: Data Byte 1 Definition

Bits	R/W	UBM Port Route Information Descriptor Byte 1 Definition
7:0	R	DFC Status and Control Descriptor Index – indicates the index to be used to address this DFC Status and Control Descriptor via the specified UBM Controller 2Wire Slave Address. A DFC Status and Control Descriptor Index value of FFh indicates there is no valid Drive Facing Connector routing to the Host Facing Connector (e.g., the HFC high speed signals routed to a PCIe Switch or SAS Expander). If 2Wire Device Arrangement implements a 2Wire Mux, this field also represents the Mux Channel. (See Section 5.5)

6.3.2.2.3 Data Byte 2 Definition

Table 6-15 Port Route Information: Data Byte 2 Definition

Bits	R/W	UBM Port Route Information Descriptor Byte 2 Definition
7:0	R	Drive Types Supported – indicates which drive types are supported in the Drive Facing Connector. The value returned is a bitmask of device types that are supported by the DFC. The Host uses this field with the Drive Type Installed field (See Section 7.2.15) to determine if the device installed is supported.

IFDET2#	IFDET#	PRSNT#	Support Bit Position	Device Type
0	0	0	0	~~Reserved~~Other (e.g. SFF-TA-1002)
0	0	1	1	SFF-TA-1001 PCIe
0	1	0	2	Reserved
0	1	1	3	Gen-Z
1	0	0	4	SAS/SATA
1	0	1	5	Quad PCIe
1	1	0	6	Reserved
1	1	1	7	DFC Empty

Examples:
If the backplane implements a SFF-TA-1001 drive facing connector, bits 1, 4, and 7 are set to 1.

If the backplane implements a Quad PCIe drive facing connector, bits 5 and/or 4 and 7 are set to 1.

If the backplane implements a SFF-TA-1002 drive facing connector (e.g., EDSFF), the PRSNT0# pin on the connector is associated to IFDET2#, IFDET#, and PRSNT# signals.

6.3.2.2.4 Data Byte 3 Definition

Table 6-16 Port Route Information: Data Byte 3 Definition

Bits	R/W	UBM Port Route Information Descriptor Byte 3 Definition
7	R	Domain – indicates if this UBM Port Route Information Descriptor is describing the primary or secondary port of a DFC. 0 = Primary Port 1 = Secondary Port
6	R	Port Type – indicates the connector port type which is routed from the DFC to the HFC. 0 = Converged (i.e., supports PCIe protocol and SAS/SATA protocol) 1 = Segregated (i.e., supports PCIe protocol via the Quad PCIe port lanes) Note: The Host uses this field, the Drive Types Supported (See Section 6.3.2.2.3) and Max SAS, SATA and/or PCIe Link Rate (See 6.3.2.2.5) and the Drive Type Installed field (See Section 7.2.15) to determine the actual device protocol supported and installed.
5:4	R	Reserved
3:0	R	Link Width – indicates the number of lanes in the port. 0h = 1 lane 1h = 2 lanes 2h = 4 lanes 3h = 8 lanes 4h = 16 lanes 5h-Fh = Reserved

6.3.2.2.5 Data Byte 4 Definition

Table 6-17 Port Route Information: Data Byte 4 Definition

Bits	R/W	Port Route Information Descriptor Byte 4 Definition
7:5	R	Max SAS Link Rate 0h = Not Supported 1h = SAS-1 (3 Gb/s) 2h = SAS-2 (6 Gb/s) 3h = SAS-3 (12 Gb/s) 4h = SAS-4 (22.5 Gb/s) 5h = SAS-5 (TBD) 6h = SAS-6 (TBD) 7h = No Limit
4:2	R	Max PCIe Link Rate 0h = Not Supported 1h = PCIe-1 (2.5 GT/s) 2h = PCIe-2 (5 GT/s) 3h = PCIe-3 (8 GT/s) 4h = PCIe-4 (16 GT/s) 5h = PCIe-5 (32 GT/s) 6h = PCIe-6 (TBD) 7h = No Limit
1:0	R	Max SATA Link Rate 0h = Not Supported 1h = 3 Gb/s 2h = 6 Gb/s 3h = No Limit

6.3.2.2.6 Data Byte 5 Definition

Table 6-18 Port Route Information: Data Byte 5 Definition

Bits	R/W	UBM Port Route Information Descriptor Byte 5 Definition
7:4	R	Host Facing Connector Identity – indicates the Host Facing Connector Identity (See Section 7.2.8).
3:0	R	Host Facing Connector Starting Lane – indicates the Host Facing Connector Starting Lane (See Section 5.11).

6.3.2.2.7 Data Byte 6 Definition

Table 6-19 Port Route Information: Data Byte 6 Definition

Bits	R/W	UBM Port Route Information Descriptor Byte 6 Definition
7:0	R	Slot Offset – indicates the backplane slot offset for the Drive Facing Connector. See Section 5.10

7. UBM Controller

The UBM Controller manages the Host Facing Connector sideband I/O signaling, the Drive Facing Connector I/O signaling and the LED states for the DFC. The UBM Controller also provides information for the Host to determine the Drive Facing Connectors associated to the Host Facing Connector. The sections that follow provide the 2Wire transaction protocol and commands to access the UBM Controller. One or more UBM Controllers may be associated to a single UBM FRU.

7.1 2Wire Protocol

The UBM Controller is accessed via a 2Wire transaction checksum protected protocol.

Each of the checksums are computed by summing an initial checksum seed value of 0xA5 and all of the specified bytes as unsigned 8-bit binary numbers and discarding any overflow bits. The two's complement of this summation is used as the checksum value.

Table 7-1 provides information to be able to read the subsequent 2Wire Transaction figures.

Table 7-1 UBM Controller 2wire Transaction Legend

LEGEND	DESCRIPTION
	Driven by 2Wire Master, Checksum checked by 2Wire Slave
▨	Driven by 2Wire Slave, Checksum checked by 2Wire Master
W	Driven LOW by 2Wire Master to indicate Write Phase
R	Driven HIGH by 2Wire Master to indicate Read Phase

A UBM Controller Write transaction consists of the 2Wire Master writing the Slave Address, the Command Byte, one or more Data Bytes, and a Write Checksum to the 2Wire Slave. The Write Checksum includes all bytes transferred prior to the Write Checksum, including the byte containing the Slave Address and Command Byte. The Host shall use the Last Command Status command to determine if the UBM Controller successfully received the previous command.

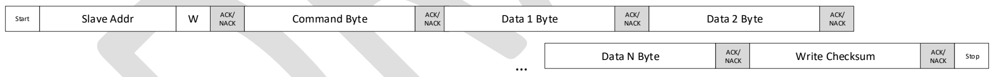


Figure 7-1 UBM Controller Write Transaction

A UBM Controller Read transaction consists of the 2Wire Master writing the Slave Address, the Command Byte and the Command Checksum, then the 2Wire Master continues the transaction by reading one or more data bytes and the Read Checksum from the 2Wire Slave. The Command Checksum includes all bytes transferred prior to the Command Checksum, including the byte containing the Slave Address. The Read Checksum includes all of the transferred Data Byte values. The Host shall use the Read Checksum to determine if the UBM Controller successfully received the previous command.

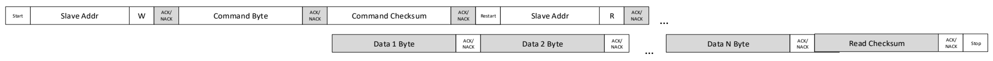


Figure 7-2 UBM Controller Read Transaction

Table 7-2 thru

Table 7-5 describes the expected 2Wire usages of the UBM Controller:

Table 7-2 UBM Controller Successful Read Transaction Sequence

Successful Read Transaction	
HOST	UBM CONTROLLER
Issues write phase of the Command request (Slave Addr, Command, Command Checksum).	
	Validates Command Checksum. Prepares read data for the Command request, including the Read Checksum.
Issues read phase of Command request (Slave Addr, Data Bytes, Read Checksum, Stop)	
	Returns Read Data and Read Checksum
Validates Read Checksum.	

Table 7-3 UBM Controller Successful Write Transaction Sequence

Successful Write Transaction	
HOST	UBM CONTROLLER
Issues the Write transaction (Slave Addr, Command, Data Bytes, and Write Checksum).	
	Validates Write Checksum, processes the Command and Data Bytes, then the UBM Controller sets Last Command Status to SUCCESS.
Issue write phase with the Last Command Status Command request (Slave Addr, Command, Command Checksum).	
	Validates Command Checksum. Prepares read data for the Last Command Status request including Read Checksum.
Issues read phase of the Last Command Status Command request (Slave Addr, Data Byte, Read Checksum)	
	Returns Last Command Status and Read Checksum.
Validates Read Checksum.	
Validate Last Command Status is successful.	

Table 7-4 UBM Controller Invalid Write Transaction Sequence

Invalid Write Command or Invalid Write Checksum detected by UBM Controller	
HOST	UBM CONTROLLER
Host issues write phase of Command request (Slave Addr, Command, Data Bytes, Write Checksum).	
	Validates Write Checksum, settings Last Command Status as defined in Section 7.2.2.
Issue write phase with the Last Command Status Command request.	
	Validates Command Checksum. Prepares read data for the Last Command Status request including Read Checksum.
Issues read phase of Command request (Slave Addr, Data Byte, Read Checksum)	
	Returns Last Command Status and Read Checksum.
Validates Read Checksum.	
Validate Last Command Status as defined in Section 7.2.2	

Table 7-5 UBM Controller Invalid Read Transaction Sequence

Invalid Read Checksum detected by UBM Controller	
HOST	UBM CONTROLLER
Host issues write phase of Command request (Slave Addr, Command, Command Checksum).	
	Detects Invalid Command or Invalid Command Checksum. Prepares all read data bytes to FFh for the Command request, including the Read Checksum. The Host detects the invalid Read Checksum for the next transaction.
Issues read phase of Command request (Slave Addr, Data, Read Checksum, Stop)	
	Transmission error occurs while returning the Read Data and Read Checksum.
Detects Invalid Read Checksum.	
Host reattempts the write phase of the command transaction.	

7.2 UBM Controller Commands

Table 7-6 shows a summary of the UBM Controller Command Set.

Table 7-6 UBM Controller Command Set

COMMAND CODE	READ/WRITE	COMMAND NAME	NUMBER OF DATA BYTES	DESCRIPTION	MANDATORY / OPTIONAL	REFERENCE
00h	Read Only	Operational State	1	Returns the operating state of the UBM Controller	Mandatory	7.2.1
01h	Read Only	Last Command Status	1	Returns the last command execution status of the UBM Controller	Mandatory	7.2.2
02h	Read Only	Silicon Identity and Version	14	Returns UBM Controller identification data	Mandatory	0
03h	Read Only	Programming Update Mode Capabilities	1	Returns the Programming Update Mode capabilities of the UBM Controller	Mandatory	0
04h – 1Fh	Reserved for future Generic Commands					
20h	Read/Write	Enter Programmable Update Mode	5	Indicates a sequence to unlock and Transfer to Programmable Update Mode.	Optional	7.2.5
21h	Read/Write	Programmable Mode Data Transfer	N	Indicates method to exchange multiple bytes of command, status and data	Optional	7.2.6
22h	Read/Write	Exit Programmable Update Mode	4	Indicates to transfer out of Programmable Update Mode.	Optional	7.2.7
23h – 2Fh	Reserved for future Programmable Commands					
30h	Read Only	Host Facing Connector Info	1	Returns the Host Facing Connector information	Mandatory	7.2.8
31h	Read Only	Backplane Info	1	Returns the backplane number and type that is unique in the chassis.	Mandatory	0
32h	Read Only	Starting Slot	1	Returns the Starting Slot which is applied to the Slot Offset found in the UBM Port Route Information of the UBM FRU. See Section 5.11	Mandatory	7.2.10
33h	Read Only	Capabilities	2	Returns the backplane capabilities.	Mandatory	7.2.11
34h	Read/Write	Features	2	Indicates the UBM Controller features.	Mandatory	7.2.12
35h	Read/Write	Change Count	2	Counter used to manage UBM Controller interrupts.	Mandatory	0
36h	Read/Write	DFC Status and Control Descriptor Index	1	Controls the DFC Status and Control Descriptor to access.	Mandatory	7.2.14
37h – 3Fh	Reserved for future Backplane Management Commands					
40h	Read/Write	DFC Status and Control Descriptor	8	Indicates the DFC Status and Control Descriptor data for the current DFC Status and Control Descriptor Index.	Mandatory	7.2.15
41h – 4Fh	Reserved for future Descriptors					
50h – 9Fh	Reserved					
A0h – AFh	Vendor Specific					
B0h – FFh	Reserved					

7.2.1 Operational State Command

The Operational State Command returns a valid and current Operational State of the UBM Controller as defined in Table 7-7. An Operational State other than READY indicates to the Host that responses to other commands or data in the UBM Controller cannot be trusted.

Table 7-7 Operational State Command

READ/ WRITE	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Read	+0	Operational State							

Table 7-8 Operational State Command Descriptions

OPERATIONAL STATE	VALUE	DESCRIPTION
INVALID	00h	Reserved
INITIALIZING	01h	State during the UBM Controller Initialization before configuration has completed
BUSY	02h	State indicates the Data in UBM Controller is inconsistent.
READY	03h	State indicates UBM Controller has been configured and data provided is consistent
REDUCED FUNCTIONALITY	04h	State used to indicate UBM is currently operating in Reduced Functionality
Reserved	05h – 0Fh	Reserved
Vendor Specific	10h – 1Fh	Vendor Specific
Reserved	20h – FFh	Reserved

7.2.2 Last Command Status Command

The Last Command Status Command returns the status of the previous Write Transaction request status (i.e., Last Command Status field) as defined in Table 7-9.

Table 7-9 Last Command Status Command

READ/ WRITE	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Read	+0	Last Command Status							

Table 7-10 Last Command Status Descriptions

LAST COMMAND STATUS NAME	LAST COMMAND STATUS VALUE	DESCRIPTION
FAILED	00h	UBM Controller last command request has failed.
SUCCESS	01h	Last Command received was processed correctly
INVALID CHECKSUM	02h	Invalid Checksum detected
TOO MANY BYTES WRITTEN	03h	Write transaction byte count larger than the Command
NO ACCESS ALLOWED	04h	Host facing connector is not allowed to perform command request
CHANGE COUNT DOES NOT MATCH	05h	The Change Count command did not specify the current Change Count value.
BUSY	06h	UBM Controller is busy processing the last command request. UBM Controller Busy timeout is 30 seconds.
COMMAND NOT IMPLEMENTED	07h	UBM Controller does not support the command request.
INVALID DESCRIPTOR INDEX	08h	UBM Controller has detected an invalid descriptor index.
Reserved	09h – 0Fh	Reserved
Vendor Specific	10h – 1Fh	Vendor Specific
Reserved	20h – FFh	Reserved

7.2.3 Silicon Identity and Version Command

The Silicon Identity and Version Command returns UBM Controller information as defined in Table 7-11.

Table 7-11 Silicon Identity and Version Command

READ/ WRITE	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Read Only	+0	UBM Specification Major Version				UBM Specification Minor Version			
Read Only	+1	PCI Vendor ID [LSB]							
Read Only	+2	PCI Vendor ID [MSB]							
Read Only	+3	Reserved							
Read Only	+4	UBM Controller Device Code [LSB]							
Read Only	+5	UBM Controller Device Code							
Read Only	+6								
Read Only	+7	UBM Controller Device Code [MSB]							
Read Only	+8	Reserved							
Read Only	+9	Reserved							
Read Only	+10	UBM Controller Image Version Minor							
Read Only	+11	UBM Controller Image Version Major							
Read Only	+12	Vendor Specific							
Read Only	+13	Vendor Specific							

The UBM Specification version is provided in a single byte. The Major and Minor specification version is expressed via the examples in Table 7-12.

Table 7-12 UBM Specification Version (Examples)

UBM SPECIFICATION VERSION	UBM SPECIFICATION MAJOR VERSION	UBM SPECIFICATION MINOR VERSION	VALUE
0.1	0	1	01h
0.6	0	6	06h
0.7	0	7	07h
1.0	1	0	10h
1.N	1	N	1Nh
2.N	2	N	2Nh
Major(Y).Minor(X)	Y	X	YXh

The PCI Vendor ID provides the Vendor ID assigned by PCI-SIG.

The UBM Controller Device Code provides the silicon identity device code of the UBM Controller and is unique per PCI Vendor ID.

The UBM Controller Image Version Major and Minor fields provide the UBM Controller Image version information.

7.2.4 Programmable Update Mode Capabilities Command

The Programming Update Mode Capabilities Command returns the UBM Controller Programming Update Mode Capabilities as defined in Table 7-13.

Table 7-13 Programming Update Mode Capabilities Command

READ/ WRITE	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Read Only	+0	Reserved						Programmable Update Modes	

Table 7-14 Programming Update Mode Capabilities: Data Byte 0 Definition

BITS	READ/ WRITE	BYTE 0 DEFINITION
7:2	R	Reserved
1:0	R	Programmable Update Modes 0h = Programming Update is not supported 1h = Programming Update supported while Devices remain online. 2h = Programming Update supported while Devices are offline. 3h = Programming Update support is Vendor Specific.

7.2.5 Enter Programmable Update Mode Command (Optional)

The Enter Programmable Update Mode Command unlocks the UBM Controller for a UBM Controller Image Update. This command requires a specific unlock sequence to ensure Programmable Update Mode is not entered unless specifically requested. The Enter Programmable Update Mode Command is defined in Table 7-15.

Table 7-15 Enter Programing Update Mode Command

R/W	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Read	+0	2Wire Slave Address for Programmable Update Mode							
Read/ Write	+1	Unlock Sequence 0 (55h)							
Read/ Write	+2	Unlock Sequence 1 (42h)							
Read/ Write	+3	Unlock Sequence 2 (4Dh)							
Write	+4	Reserved							Transfer to Programmable Update Mode

The 2Wire Slave Address for Programmable Update Mode field specifies the 2Wire Slave Address used to update the UBM Controller Image.

To request the transition to Programmable Update Mode, the Unlock Sequence fields are set to the values defined in Table 7-15, and the Transfer to Programmable Update Mode field is set to 1h. If the Unlock Sequence field values or the Transfer to Programmable Update Mode field is set to 0h, then the UBM Controller fails the command request (i.e., Last Command Status field indicates a 00h (i.e., FAILED)) and does not transfer to Programmable Update Mode.

The UBM Controller shall transfer to Programmable Update Mode immediately after completing the UBM Controller Write transaction for the Enter Programming Update Mode Command.

While in Programmable Update Mode, the Operational State shall reflect REDUCED FUNCTIONALITY. Only upon successful exit from Programmable Update Mode, the Operational State shall leave the REDUCED FUNCTIONALITY Operational State.

See Section 5.20 for further details about REDUCED FUNCTIONALITY Operational State.

7.2.6 Programmable Mode Data Transfer Command (Optional)

The Programmable Mode Data Transfer (PMDT) Command defines Subcommands that are used to update a UBM Controller Image. This command is only successfully processed when the UBM Controller is in Programmable Update Mode (See 7.2.4). This command uses 2Wire variable length transactions defined in Section 7.2.6.1.

A PMDT Write command is a UBM Controller PMDT Write Transaction that consists of the write transfer of data bytes in PMDT Write Format.

A PMDT Read command is a UBM Controller PMDT Read Transaction that consists of the write transfer of data bytes in PMDT Write Format followed by the read transfer of data bytes in PMDT Read Format.

The PMDT Write Format is defined in Table 7-16.

Table 7-16 PMDT Write Format

R/W	7	6	5	4	3	2	1	0
Write	Programmable Mode Subcommand							
Write	Number of Data Bytes (N)							
Write	Data 1 Byte							
Write	…							
Write	Data N Byte							

The Programmable Mode Subcommands field is defined in Table 7-17.

Table 7-17 Programmable Mode Subcommands

PROGRAMMABLE MODE SUB**COMMANDS**	VALUE	PMDT COMMAND	DESCRIPTION	REFERENCE
INVALID COMMAND	00h	Reserved		
GET NON VOLATILE STORAGE GEOMETRY	01h	Read	Returns the nonvolatile structure and size of the programmable segments.	7.2.6.2
ERASE	02h	Write	Erases a segment of the nonvolatile location to prepare for programming.	7.2.6.3
ERASE STATUS	03h	Read	Returns the status of an erase request.	7.2.6.4
PROGRAM	04h	Write	Writes a segment of data into the nonvolatile location.	0
PROGRAM STATUS	05h	Read	Returns the status of a program request.	0
VERIFY	06h	Write	Sets the Sector and Sector Index for a Verify Status request.	7.2.6.7
VERIFY STATUS	07h	Read	Returns the checksum for the nonvolatile segment.	7.2.6.8
VERIFY IMAGE	08h	Write	Sets the Image Number for an Image Number Status request.	0
VERIFY IMAGE STATUS	09h	Read	Returns information indicating UBM Controller Image is valid.	7.2.6.10
SET ACTIVE IMAGE	0Ah	Write	If multiple UBM Controller Images are supported, this command is used to set the next image to use.	0
ACTIVE IMAGE STATUS	0Bh	Read	Returns the status of a set active image request.	7.2.6.12
Reserved	0Ch – 0Fh	Reserved		
Vendor Specific	20h - FFh	Vendor Specific		

The Number of Data Bytes field specifies the number of data bytes that follow in the PMDT Write Format.

The PMDT Read Format is defined in Table 7-18.

Table 7-18 PMDT Read Format

R/W	7	6	5	4	3	2	1	0
Read	Programmable Mode Status							
Read	Number of Data Bytes (N)							
Read	Data 1 Byte							
Read	...							
Read	Data N Byte							

The Programmable Mode Status field is defined in Table 7-19.

Table 7-19 Programmable Mode Status

PROGRAMMABLE MODE STATUS	VALUE	DESCRIPTION
INVALID STATUS	00h	Reserved
SUCCESS	01h	Last Command was successful and contains returned data following this Status code.
IMAGE VERIFY FAILED	02h	UBM Controller Image did not verify properly.
UNSUPPORTED DEVICE	03h	UBM Controller Image is not supported by the UBM Controller device.
NON-VOLATILE LOCATION INVALID	04h	Non-Volatile Location requested is invalid.
UNKNOWN ERROR	05h	Unknown programming error has occurred.
BUSY	06h	Last Command is still busy executing. Host should retry command.
Reserved	07h – 0Fh	Reserved

The Number of Data Bytes field indicates the number of data bytes that follow in the PMDT Read Format.

7.2.6.1 2 Wire Variable Length Transactions

The Programmable Mode Data Transfer Command uses PMDT Write Transactions and a PMDT Read Transactions.

A UBM Controller PMDT Write Transaction consists of the 2Wire Master writing the Slave Address, the Command Byte (i.e., value of 21h), the Subcommand Byte, Number of Data Bytes, one or more data bytes defined by the PMDT Write Format, and a Write Checksum to the 2Wire Slave. The Write Checksum includes all bytes transferred prior to the Write Checksum, including the Slave Address, Command Byte, and all of the transferred data byte values. The Host shall use the Last Command Status command to determine if the UBM Controller successfully received the previous command. Figure 7-3 depicts the UBM Controller PMDT Write Transaction.

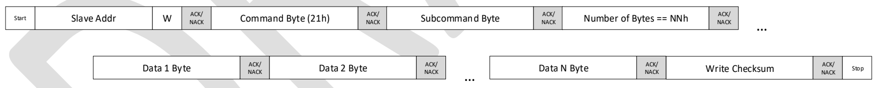


Figure 7-3 UBM Controller PMDT Write Transaction

A UBM Controller PMDT Read Transaction consists of the 2Wire Master writing the Slave Address, the Command Byte (i.e., value of 21h), one or more data bytes defined by the PMDT Write Format, and the Command Checksum, then the 2Wire Master continues the transaction by reading one or more data bytes defined by the PMDT Read Format, and the Read Checksum from the 2Wire Slave. The Command Checksum includes all bytes transferred prior to the Command Checksum, including the byte containing the Slave Address. The Read Checksum includes all of the transferred data byte values. The UBM Controller shall pad bytes after the Read Checksum with FFh for the remainder of the UBM Controller 2Wire Max Byte Count (See 6.3.1.2.2) for the read transaction. The UBM Host shall not include padded bytes in Read Checksum calculation. In the event a UBM Host terminates a read transaction before the Read Checksum has been received, the UBM Controller shall gracefully handle the subsequent transaction as a new transaction. Figure 7-4 depicts the UBM Controller PMDT Read Transaction.

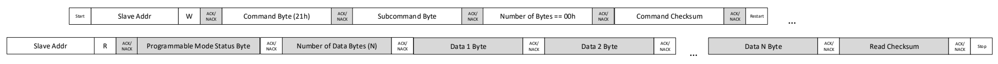


Figure 7-4 UBM Controller PMDT Read Transaction

7.2.6.2 Get Non-Volatile Storage Geometry Subcommand

The Get Non-Volatile Storage Geometry Subcommand indicates the storage sector quantity and size of the storage sectors. Erasing, Programming and Verifying use the Sector Number and Sector Index to describe where the erase, program, and verify operations are performed in the Non-Volatile storage map. Figure 7-5 defines the layout relationship between Sectors and Sector Indexes of a non-volatile storage device. Each Sector is comprised of multiple indexes into the sector.

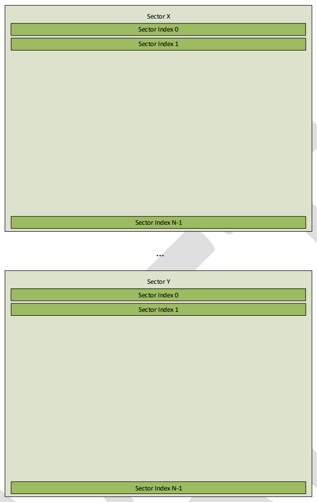


Figure 7-5 Non-Volatile Storage Geometry Diagram

Table 7-20 defines the PMDT Write Format for the Get Non-Volatile Storage Geometry Subcommand.

Table 7-20 PMDT Write Format for the Get Non-Volatile Storage Geometry Subcommand

R/W	7	6	5	4	3	2	1	0
Write	Programmable Mode Subcommand = 01h (Get Non Volatile Storage Geometry)							
Write	Number of Data Bytes (N = 0)							

The Programmable Mode Subcommand field is defined in Table 7-17 and shall be set to 01h (i.e., GET NON VOLATILE STORAGE GEOMETRY).

The Number of Data Bytes field specifies the number of data bytes that follow in the PMDT Write Format.

Table 7-21 defines the PMDT Read Format for the Get Non-Volatile Storage Geometry Subcommand.

Table 7-21 PMDT Read Format for the Get Non-Volatile Storage Geometry Subcommand

R/W	7	6	5	4	3	2	1	0
Read	Programmable Mode Status = 01h [Success]							
Read	Number of Data Bytes							
Read	Number of Sectors (Y)							
Read	Sector Size							
Read	First Sector Index (Sector ~~0~~X)							
Read	Last Sector Index (Sector ~~0~~X)							
Read	…							
Read	First Sector Index (Sector Y-1)							
Read	Last Sector Index (Sector Y-1)							

The Programmable Mode Status field is defined in Table 7-19.

The Number of Data Bytes field indicates the number of data bytes that follow in the PMDT Read Format.

The Number of Sectors field indicates the number of First and Last Sector Index field pairs in the PMDT Read Format.

The Sector Size field is represented as a power of two. The Sector Size is calculated as 2 ^ Sector Size bytes.

The First Sector Index field indicates the lowest Sector Index value for the Sector.

The Last Sector Index field indicates the largest Sector Index value for the Sector.

Note: The Number of Bytes in a Sector Index is calculated by (2 ^ Sector Size) / Number of Sector Indexes. The Number of Sector Indexes is the count of Indexes from First Sector Index and Last Sector Index.

Note: The Sector X depicted in the diagram can be Sector Index 0, or the first Sector Index defined by the programmable image where the first physical sector resides for programming of the non-volatile memory.

7.2.6.3 Erase Subcommand

The Erase Subcommand erases the non-volatile storage at the location specified by Sector Number and Sector Index.

Table 7-22 indicates the PMDT Write Format for the Erase Subcommand.

Table 7-22 PMDT Write Format for the Erase Subcommand

R/W	7	6	5	4	3	2	1	0
Write	Programmable Mode Subcommand = 02h (Erase)							
Write	Number of Data Bytes (02h)							
Write	Sector Number (0 to Y-1)							
Write	Sector Index (First to Last)							

The Programmable Mode Subcommand field is defined in Table 7-17 and shall be set to 02h (i.e., ERASE).

The Number of Data Bytes field specifies the number of data bytes that follow in the PMDT Write Format.

The Sector Number field specifies the Sector in the Non-Volatile Storage.

The Sector Index field specifies the Sector Index in the Sector of the Non-Volatile Storage.

The Sector Number and Sector Index fields shall be in the range indicated by the Get Non-Volatile Storage Geometry Subcommand. If the Sector Number or Sector Index fields are not in the range, the Programmable Mode Status in the Erase Status Subcommand (See 7.2.6.4) shall indicate a value of 04h (i.e., NON-VOLATILE LOCATION INVALID).

7.2.6.4 Erase Status Subcommand

The Erase Status Subcommand indicates the status of the last Erase Subcommand (See 7.2.6.3) issued to the UBM Controller. The PMDT Write Format for the Erase Status Subcommand is defined in Table 7-23.

Table 7-23 PMDT Write Format for the Erase Status Subcommand

R/W	7	6	5	4	3	2	1	0
Write	Programmable Mode Subcommand = 03h (Erase Status)							
Write	Number of Data Bytes (00h)							

The Programmable Mode Subcommand is defined in Table 7-17 and shall be set to 03h (i.e., ERASE STATUS).

The Number of Data Bytes field specifies the number of data bytes that follow in the PMDT Write Format.

The PMDT Read Format for the Erase Status Subcommand is defined in Table 7-24.

Table 7-24 PMDT Read Format for the Erase Status Subcommand

R/W	7	6	5	4	3	2	1	0
Read	Programmable Mode Status = XXh							
Read	Number of Data Bytes (02h)							
Read	Sector Number							
Read	Sector Index							

The Programmable Mode Status is defined in Table 7-19.

The Number of Data Bytes field indicates the number of data bytes that follow in the PMDT Read Format.

The Sector Number field indicates the Sector in the Non-Volatile Storage.

The Sector Index field indicates the Sector Index in the Sector of the Non-Volatile Storage.

7.2.6.5 Program Subcommand

The Program Subcommand programs the Non-Volatile Storage at the location specified by the Sector Number and Sector Index. If more than one stream of bytes is necessary to complete the sector index programming, the application sequence number shall be incremented for each subsequent stream of data bytes. The PMDT Write Format for the Program Subcommand is defined in Table 7-25.

Table 7-25 PMDT Write Format for the Program Subcommand

R/W	7	6	5	4	3	2	1	0
Write	Programmable Mode Subcommand = 04h (Program)							
Write	Number of Data Bytes (N)							
Write	Sector Number (~~0~~ X to Y-1)							
Write	Sector Index							
Write	Application Sequence Number							
Write	First Data Byte							
Write	…							
Write	Last Data Byte							

The Programmable Mode Subcommand field is defined in Table 7-17 and shall be set to 04h (i.e., PROGRAM).

The Number of Data Bytes field specifies the number of data bytes that follow in the PMDT Write Format.

The Sector Number field specifies the Sector in the Non-Volatile Storage.

The Sector Index field specifies the Sector Index in the Sector of the Non-Volatile Storage.
The Sector Number and Sector Index fields shall be in the range indicated by the Get Non-Volatile Storage Geometry Subcommand. If the Sector Number or Sector Index fields are not in the range, the Programmable Mode Status in the Program Status Subcommand (See 7.2.6.6) shall indicate a value of 04h (i.e., NON-VOLATILE LOCATION INVALID).

The Application Sequence Number field specifies the sequence number of the Program Subcommand. The Host uses this field as a reference to check the status of the Program Subcommand by issuing the Program Status Subcommand (See 7.2.6.6).

The Data Bytes are the data to be programmed at the specified Sector Index within the Sector Number location of the Non-Volatile Storage.

7.2.6.6 Program Status Subcommand

The Program Status Subcommand provides programming status specific to the last application sequence issued with a Program Subcommand (See 7.2.6.5). The PMDT Write Format for the Program Status Subcommand is defined in Table 7-26.

Table 7-26 PMDT Write Format for the Program Status Subcommand

R/W	7	6	5	4	3	2	1	0
Write	Programmable Mode Subcommand = 05h (Program Status)							
Write	Number of Data Bytes (00h)							

The Programmable Mode Subcommand field is defined in Table 7-17 and shall be set to 05h (i.e., PROGRAM STATUS).

The Number of Data Bytes field specifies the number of data bytes that follow in the PMDT Write Format.

The PMDT Read Format for the Program Status Subcommand is defined in Table 7-27.

Table 7-27 PMDT Read Format for the Program Status Subcommand

R/W	7	6	5	4	3	2	1	0
Read	Programmable Mode Status = XXh							
Read	Number of Data Bytes (01h)							
Read	Application Sequence Number							

The Programmable Mode Status field is defined in Table 7-19.

The Number of Data Bytes field indicates the number of data bytes that follow in the PMDT Read Format.

The Application Sequence Number field indicates the sequence number of the Program Subcommand.

7.2.6.7 Verify Subcommand

The Verify Subcommand verifies the Non-Volatile Storage at the location specified by the Sector Number and the Sector Index. The PMDT Write Format for the Verify Subcommand is defined in Table 7-28.

Table 7-28 PMDT Write Format for the Verify Subcommand

R/W	7	6	5	4	3	2	1	0
Write	Programmable Mode Subcommand = 06h (Verify)							
Write	Number of Data Bytes (02h)							
Write	Sector Number (~~0~~ X to Y-1)							
Write	Sector Index							

The Programmable Mode Subcommand field is defined in Table 7-17 and shall be set to 06h (i.e., VERIFY).

The Number of Data Bytes field specifies the number of data bytes that follow in the PMDT Write Format.

The Sector Number field specifies the Sector in the Non-Volatile Storage.

The Sector Index field specifies the Sector Index in the Sector of the Non-Volatile Storage.

The Sector Number and Sector Index fields shall be in the range indicated by the Get Non-Volatile Storage Geometry Subcommand. If the Sector Number or Sector Index fields are not in the range, the Programmable Mode Status in the Verify Status Subcommand (See 7.2.6.8) shall indicate a value of 04h (i.e., NON-VOLATILE LOCATION INVALID).

7.2.6.8 Verify Status Subcommand

The Verify Status Subcommand indicates the status of the last Verify Subcommand (See 7.2.6.7). The PMDT Write Format for the Verify Status Subcommand is defined in Table 7-29.

Table 7-29 PMDT Write Format for the Verify Status Subcommand

R/W	7	6	5	4	3	2	1	0
Write	Programmable Mode Subcommand = 07h (Verify Status)							
Write	Number of Data Bytes (00h)							

The Programmable Mode Subcommand field is defined in Table 7-17 and shall be set to 07h (i.e., VERIFY STATUS).

The Number of Data Bytes field specifies the number of data bytes that follow in the PMDT Write Format.

The PMDT Read Format for the Verify Status Subcommand is defined in Table 7-30.

Table 7-30 PMDT Read Format for the Verify Status Subcommand

R/W	7	6	5	4	3	2	1	0
Read	Programmable Mode Status = XXh							
Read	Number of Data Bytes (03h)							
Read	Sector Number							
Read	Sector Index							
Read	Sector Index Checksum							

The Programmable Mode Status field is defined in Table 7-19.

The Number of Data Bytes field indicates the number of data bytes that follow in the PMDT Read Format.

The Sector Number field indicates the Sector in the Non-Volatile Storage.

The Sector Index field indicates the Sector Index in the Sector of the Non-Volatile Storage.

The Sector Index Checksum field indicates the two’s complement of the summation of bytes located at the Sector Index.

7.2.6.9 Verify Image Subcommand

The Verify Image Subcommand verifies the specified Image Number. The PMDT Write Format for the Verify Image Subcommand is defined in Table 7-31.

Table 7-31 PMDT Write Format for the Verify Image Subcommand

R/W	7	6	5	4	3	2	1	0
Write	Programmable Mode Subcommand = 08h (Verify Image)							
Write	Number of Data Bytes (01h)							
Write	Image Number							

The Programmable Mode Subcommand field is defined in Table 7-17 and shall be set to 08h (i.e., VERIFY IMAGE).

The Number of Data Bytes field specifies the number of data bytes that follow in the PMDT Write Format.

The Image Number field specifies the Image Number associated to the vendor specific data set in the Non-Volatile Storage to be verified.

7.2.6.10 Verify Image Status Subcommand

The Verify Image Status Subcommand indicates the status of the last Verify Image Subcommand (0). The PMDT Write Format for the Verify Image Status Subcommand is defined in Table 7-32.

Table 7-32 PMDT Write Format for the Verify Image Status Subcommand

R/W	7	6	5	4	3	2	1	0
Write	Programmable Mode Subcommand = 09h (Verify Image Status)							
Write	Number of Data Bytes (00h)							

The Programmable Mode Subcommand field is defined in Table 7-17 and shall be set to 09h (i.e., VERIFY IMAGE STATUS).

The Number of Data Bytes field specifies the number of data bytes that follow in the PMDT Write Format.

The PMDT Read Format for the Verify Image Status Subcommand is defined in Table 7-33.

Table 7-33 PMDT Read Format for the Verify Image Status Subcommand

R/W	7	6	5	4	3	2	1	0
Read	Programmable Mode Status = XXh							
Read	Number of Data Bytes (01h)							
Read	Image Number							

The Programmable Mode Status field is defined in Table 7-19.

The Number of Data Bytes field indicates the number of data bytes that follow in the PMDT Read Format.

The Image Number field indicates the Image Number associated to the vendor specific data set in the Non-Volatile Storage to be verified.

7.2.6.11 Set Active Image Subcommand

The Set Active Image Subcommand is used to specify the UBM Controller Image that should be activated upon exiting from Programmable Update Mode. If the UBM Controller Image is not valid, the UBM Controller Image will not be activated. The PMDT Write Format for the Set Active Image Subcommand is defined in Table 7-34.

Table 7-34 PMDT Write Format for the Set Active Image Subcommand

R/W	7	6	5	4	3	2	1	0
Write	Programmable Mode Subcommand = 0Ah (Set Active Image)							
Write	Number of Data Bytes (01h)							
Write	Image Number							

The Programmable Mode Subcommand field is defined in Table 7-17 and shall be set to 0Ah (i.e., SET ACTIVE IMAGE).

The Number of Data Bytes field specifies the number of data bytes that follow in the PMDT Write Format.

The Image Number field specifies the Image Number associated to the vendor specific data set in the Non-Volatile Storage to be verified.

7.2.6.12 Active Image Status Subcommand

The Active Image Status Subcommand indicates the status of the last Set Active Image Subcommand (See 7.2.6.11). The PMDT Write Format for the Active Image Status Subcommand is defined in Table 7-35.

Table 7-35 PMDT Write Format for the Active Image Status Subcommand

R/W	7	6	5	4	3	2	1	0
Write	Programmable Mode Subcommand = 0Bh (Active Image Status)							
Write	Number of Data Bytes (00h)							

The Programmable Mode Subcommand field is defined in Table 7-17 and shall be set to 0Bh (i.e., ACTIVE IMAGE STATUS).

The Number of Data Bytes field specifies the number of data bytes that follow in the PMDT Write Format.

The PMDT Read Format for the Active Image Status Subcommand is defined in Table 7-36.

Table 7-36 PMDT Read Format for the Active Image Status Subcommand

R/W	7	6	5	4	3	2	1	0
Read	Programmable Mode Status = XXh							
Read	Number of Data Bytes (01h)							
Read	Image Number							

The Programmable Mode Status field is defined in Table 7-19.

The Image Number field indicates the Image Number associated to the vendor specific data set in the Non-Volatile Storage to be verified.

7.2.7 Exit Programmable Update Mode Command (Optional)

The Exit Programmable Update Mode Command requests the UBM Controller to exit the Programmable Update Mode, reset, and execute the Active Image Number. The Exit Programmable Update Mode Command is defined in Table 7-37.

Table 7-37 Exit Programmable Update Mode Command

R/W	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Read/ Write	+0	Lock Sequence 0 (55h)							
Read/ Write	+1	Lock Sequence 1 (42h)							
Read/ Write	+2	Lock Sequence 2 (4Dh)							
Read/ Write	+3	Reserved							Transfer to Operational Mode

To request the transition to Operational Mode (i.e., the READY Operational State), the Lock Sequence fields are set to the values defined in Table 7-37, and the Transfer to Operational Mode field is set to 1h. If the Lock Sequence field values or the Transfer to Operational Mode field is set to 0h, then the UBM Controller fails the command request (i.e., Last Command Status field indicates a 00h or FAILED value) and does not transfer to Operational Mode.

7.2.8 Host Facing Connector Info Command

The Host Facing Connector Info Command returns the Host Facing Connector information as defined in Table 7-38.

Table 7-38 Host Facing Connector Info Command

READ/ WRITE	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Read Only	+0	Port Type	Reserved			Host Facing Connector Identity			

Table 7-39 Host Facing Connector Info: Data Byte 0 Definition

BITS	READ/ WRITE	BYTE 0 DEFINITION
7	R	Port Type – indicates the Host Facing Connector port type which is routed to the drive facing connector ports in the backplane. 0 = Converged (i.e., supports PCIe protocol and SAS/SATA protocol) 1 = Segregated (i.e., supports PCIe protocol via the Quad PCIe port lanes)
6:4	R	Reserved
3:0	R	Host Facing Connector Identity – indicates the Host Facing Connector Identity (See Section 5.10).

7.2.9 Backplane Info Command

The Backplane Info Command returns the backplane information as defined in Table 7-40.

Table 7-40 Backplane Info Command

READ/ WRITE	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Read Only	+0	Backplane Type			Reserved	Backplane Number			

The Backplane Number field shall be unique in the chassis from another instance of a backplane. The method to determine the Backplane Number field is out of the scope of the UBM specification. Before the UBM Controller reaches the Operational State of READY, the Backplane Number field shall be unique in the chassis.

Table 7-41 Backplane Info: Data Byte 0 Definition

BITS	READ/ WRITE	BYTE 0 DEFINITION
7:5	R	Backplane Type – indicates a type value of the backplane. Multiple backplanes in the chassis shall be managed together using the same Backplane Type field value (See Section 5.12)
4	R	Reserved
3:0	R	Backplane Number – indicates a unique backplane number in the chassis.

7.2.10 Starting Slot Command

The Starting Slot Command indicates the Starting Slot value as defined in Table 7-42. See Section 5.12 for more information on the Host to Slot mapping process.

Table 7-42 Starting Slot Command

READ/ WRITE	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Read Only	+0	Starting Slot							

7.2.11 Capabilities Command

The Capabilities Command returns the UBM Controller capabilities as defined in Table 7-43.

Table 7-43 Capabilities Command

READ/ WRITE	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Read Only	+0	DFC Change Count	CHANGE_ DETECT# Interrupt Operation	2WIRE_RESET# Operation		Dual Port	PCIe Reset Control	Slot Power Control	Clock Routing
Read Only	+1	Reserved			DFC SMBus Reset Control Supported	DFC PERST# Management Override Supported	IFDET2# Reported	IFDET# Reported	PRSNT# Reported

Table 7-44 Capabilities Command: Data Byte 0 Definition

BITS	READ/ WRITE	BYTE 0 DEFINITION
7	R	DFC Change Count – indicates if a change count is maintained per an individual DFC Status and Control Command Descriptor. 0 = DFC Change Count field is not supported 1 = DFC Change Count field is supported
6	R	CHANGE_DETECT# Interrupt Operation – indicates if the CHANGE_DETECT# signal interrupt operation is supported. 0 = CHANGE_DETECT# interrupt operation is not supported 1 = CHANGE_DETECT# interrupt operation is supported
5:4	R	2WIRE_RESET# Operation – indicates the 2WIRE_RESET# signal support. 0h = 2WIRE_RESET# is not supported 1h = 2WIRE_RESET# 2Wire Slave Reset and 2Wire Mux is supported. 2h = 2WIRE_RESET# UBM FRU and UBM Controller is supported. 3h = 2WIRE_RESET# 2Wire Slave Reset and UBM FRU and UBM Controller and 2Wire Mux are supported.
3	R	Dual Port – indicates if Dual Port DFC connectors are routed. 0 = Single Port only 1 = Dual Port Supported (e.g., Quad PCIe/SFF-TA-1001 DualPortEn# signal is LOW)
2	R	PCIe Reset Control – indicates if PCIe Reset Control is supported. 0 = PCIe Reset Control is not supported 1 = PCIe Reset Control is supported See Section 5.16
1	R	Slot Power Control – indicates if the Drive Facing Connectors support Power Disable (i.e., PwrDIS signal). 0 = Drive Facing Connectors do not support Power Disable 1 = Drive Facing Connectors support Power Disable
0	R	Clock Routing – indicates availability of high speed differential clock routing (i.e., RefClk) from the Host Facing Connector to the Drive Facing Connector. 0 = No clock routing (e.g., SAS, SATA, or PCIe SRIS/SRNS) 1 = Clock routing is present See Section 5.16

Table 7-45 Capabilities Command: Data Byte 1 Definition

BITS	READ/ WRITE	BYTE 1 DEFINITION
7:~~3~~5	R	Reserved
4	R	DFC SMBus Reset Control Supported – indicates if the UBM Controller supports control over the DFC SMBRST# signals (e.g. See SFF-TA-1009) for all DFC's managed by the HFC. 0 = No Support for DFC SMB Reset Control 1 = Support for DFC SMB Reset Control
3	R	DFC PERST# Management Override Supported – indicates if the UBM Controller supports the DFC PERST# Management Override field in the Features Command. 0 = No Support for DFC PERST# Management Override 1 = Support for DFC PERST# Management Override
2	R	IFDET2# Reported – indicates if the IFDET2# signal is reported. 0 = IFDET2# signal is not reported 1 = IFDET2# signal is reported
1	R	IFDET# Reported – indicates if the IFDET# signal is reported. 0 = IFDET# signal is not reported 1 = IFDET# signal is reported Note: The minimum requirement to report if a Drive Type is Installed, or a Drive Type is Not Installed, then IFDET# shall be reported.
0	R	PRSNT# Reported – indicates if the PRSNT# is reported. 0 = PRSNT# signal is not reported 1 = PRSNT# signal is reported Note: The minimum requirement to determine if a SAS/SATA or PCIe Drive Type Installed is for PRSNT# signal to be reported. The minimum requirement to detect SAS/SATA or PCIe Drive Type Installed or DFC Empty both IFDET# and PRSNT# signals shall be reported.

7.2.12 Features Command

The Features Command is used indicate and specify the UBM Controller features as defined in Table 7-46.

Table 7-46 Features Command

READ/ WRITE	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Write/ Read	+0	DFC PERST# Management Override		Operational State Change Count Mask	Drive Type Installed Change Count Mask	PCIe Reset Change Count Mask	CPRSNT# Legacy Mode	Write Checksum Checking	Read Checksum Creation
Write/ Read	+1	Reserved							DFC SMBus Reset Control

Table 7-47 Features Command: Data Byte 0 Definition

BITS	READ/ WRITE	BYTE 0 DEFINITION
7:6	R/W	DFC PERST# Management Override – indicates the DFC PERST# behavior when a Drive has been installed. 0 = No Override (e.g., RefClk Host Managed, SRIS/SRNS Automatically released from Section 5.16) 1 = DFC PERST# Managed upon install 2 = DFC PERST# Automatically released upon install 3 = Reserved
5	R/W	Operational State Change Count Mask – indicates if a change to Operational State field causes the Change Count field to increment. 0 = Operational State transitions do not cause the Change Count field to increment 1 = Operational State transitions cause the Change Count field to increment
4	R/W	Drive Type Installed Change Count Mask – indicates if a change to Drive Type Installed field causes the Change Count field to increment. 0 = Drive Type Installed field changes do not cause the Change Count field to increment 1 = Drive Type Installed field changes cause the Change Count field to increment
3	R/W	PCIe Reset Change Count Mask – indicates if a change to PCIe Reset field causes the Change Count field to increment. 0 = PCIe Reset field changes do not cause the Change Count field to increment 1 = PCIe Reset field changes cause the Change Count field to increment
2	R/W	CPRSNT# Legacy Mode – indicates the behavior of the CPRSNT#/CHANGE_DETECT# signal. 0 = CHANGE_DETECT# interrupt operation 1 = CPRSNT# legacy operation Note: UBM FRU provides the initial default state of this operation, while the UBM Controller provides the current setting of this feature (See Section 5.8).
1	R/W	Write Checksum Checking – indicates if the UBM Controller performs Checksum verification on the write phase of a 2Wire transaction. 0 = No Checksum Checking 1 = Checksum checking is enabled
0	R/W	Read Checksum Creation – indicates if the UBM Controller generates a valid Read Checksum for the read phase of a 2Wire transaction. 0 = No Checksum Creation is performed 1 = Checksum Creation is enabled

Table 7-48 Features Command: Data Byte 1 Definition

BITS	READ/ WRITE	BYTE 1 DEFINITION
7:~~0~~1	R	Reserved
0	R/W	DFC SMBus Reset Control – controls the DFC SMBRST# signal for all DFC's associated under the HFC. 0 = NOP (e.g. No DFC SMBus Reset sequence outstanding) 1 = Initiate DFC SMBus Reset sequence Note: UBM Host requests the initiation of the DFC SMBus Reset sequence. The UBM Controller transitions the DFC SMB Reset Control field to 0 when the DFC SMBus Reset sequence completes. Timing of the DFC SMBus Reset sequence is platform specific and is out of scope of this specification.

7.2.13 Change Count Command

The Change Count Command is used to access the UBM Controller Change Count field as defined in Table 7-49.

Table 7-49 Change Count Command

READ/ WRITE	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Read / Write	+0	Change Count							
Read	+1	~~Reserved~~UBM Controller Reset Change Source	Reserved	Op State Change Source	Drive Type Installed Change Source	PCIe Reset Change Source	Reserved	Reserved	CPRSNT# Legacy Mode Change Source

The Change Count field when read contains a wrapping counter that increments each time there is a change:

a. in the Drive Type Installed field and the Drive Type Installed Change Count Mask bit (See Section 7.2.12) is set to 1 (i.e., Drive Type Installed field change causes an increment in Change Count field),
b. in the PCIe Reset field in a DFC Status and Control Descriptor and the PCIe Reset Change Count Mask bit (See Section 7.2.12) is set to 1 (i.e., PCIe Reset field change causes an increment in Change Count field),
c. in the UBM Controller Operational State and the Operational State Change Count Mask bit (See Section 7.2.12) is set to 1 (i.e., Operational State field change causes an increment in the Change Count field),
d. in the CPRSNT# Legacy Mode field (See Section 7.2.12).
e. in any DFC PERST# signal from LOW to HIGH (i.e., Deassertion) when the DFC PERST# Management Override field is set to 2h (i.e., DFC PERST# Automatically released upon install) (See Section 7.2.12).
f. in the UBM Controller Reset Change Source field changes from LOW to HIGH.

~~a.~~

The Change Count field wraps back to zero after reaching FFh. The incrementing of this register may affect the CHANGE_DETECT# signal (See Section 5.8). If this register is written with a value that is different than the current value, then the write is ignored and the Last Command Status is set to 05h (i.e., CHANGE COUNT DOES NOT MATCH).

The UBM Controller Reset Change Source field provides an indicator to a UBM Host that the UBM Controller has been reset and may require the UBM Host to perform re-enumeration of the Operational State, DFC S&C Descriptors, Change Count, and Feature values.

The UBM Controller Reset Change Source field is set to 1 in the following conditions:

a. upon initial power on of the backplane,
b. upon activation of a new programmable firmware image,
c. upon completion of a UBM Controller Reset from a 2WIRE_RESET sequence (See Section 5.2),
d. upon a vendor specific purpose.

The UBM Controller Reset Change Source field, ~~T~~the Op State Change Source field, the Drive Type Installed Change Source field, the PCIe Reset Change Source and the CPRSTN# Legacy Mode Change Source field (i.e., Change Source fields) indicate reasons for Change Count field incrementing. The respective Change Source field is set to a value of 1 when the Change Count field is incremented. All Change Source fields are set to a value of 0 when the Change Count field is written with the current Change Count field value.

> Note: The Change Count field is valid when the UBM Controller Operational State is READY. REDUCED FUNCTIONALITY Operational State shall not assert the CHANGE_DETECT# or increment the Change Count field. Upon exit from REDUCED FUNCTIONALITY Operational State the CHANGE_DETECT# signal will assert, and the Change Count field may be incremented or reset.

7.2.14 DFC Status and Control Descriptor Index Command

The DFC Status and Control Descriptor Index Command is used to access the DFC Status and Control Descriptor Index as defined in Table 7-50.

Table 7-50 DFC Status and Control Descriptor Index Command

R/W	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Read / Write	+0	DFC Status and Control Descriptor Index							

The DFC Status and Control Descriptor Index field specifies the descriptor being accessed by the DFC Status and Control Descriptor Command (See Section 7.2.15). If the specified value is not valid, then this command shall fail with an INVALID DESCRIPTOR INDEX status.

7.2.15 DFC Status and Control Descriptor Command

The DFC Status and Control Descriptor Command indicates the status of the drive installed in the Drive Facing Connector along with controlling various aspects of the Drive Facing Connector. The specific descriptor being accessed is specified by the DFC Status and Control Descriptor Index command (See 7.2.14). Table 7-51 defines the DFC Status and Control Descriptor Command.

Table 7-51 DFC Status and Control Descriptor Command

READ/ WRITE	OFFSET \ BYTE	7	6	5	4	3	2	1	0
Read/ Write	+0	PCIe Reset		Bifurcate Port	Reserved		Drive Type Installed		
Read/ Write	+1	SES Array Device Slot Element							
Read/ Write	+2								
Read/ Write	+3								
Read/ Write	+4								
Read	+5	DFC Change Count							
Read/ Write	+6	Vendor Specific							
Read/ Write	+7	Vendor Specific							

Table 7-52 DFC Status and Control Descriptor: Data Byte 0 Definition

BITS	READ/ WRITE	BYTE 0 DEFINITION
7:6	R/W	PCIe Reset – specifies the port specific DFC PERST# signal behavior. 0 = NOP (i.e., No Operation) 1 = Initiate PCIe Reset Sequence 2 = PERST# signal is held asserted (i.e., LOW) 3 = Reserved See Section 5.16 for PCIe Reset Control Management behavior See Section 7.2.13 for Change Count field and CHANGE_DETECT# signal behavior
5	R	Bifurcate Port – indicates if the DFC port link width shall be bifurcated (See Section 5.18). 0 = No Bifurcation applied 1 = Bifurcation applied (Divide Port Width by 2)
4:3	R	Reserved
2:0	R	Drive Type Installed – indicates the type of device in the DFC. (See ~~SFF-TA-1001~~ Section 6.3.2.2.3) Bit 2 = IFDET2# Bit 1 = IFDET# Bit 0 = PRSNT# One or more of these bits may not be reported as indicated in the data returned by the Capabilities Command (See Section 7.2.11). Note: The UBM Host uses Section 6.3.2.2.3 and the reported Capabilities (See Section 7.2.11) for IFDET2#, IFDET# and PRSNT# signals to determine if a Drive is installed and if it is supported by the backplane.

The SES Array Device Slot Element field is defined by the SES-4 specification for an Array Device Slot Element and follows the accessing rules of SES (e.g., if the SELECT bit is set to 0, then the rest of the bits in the field are ignored).

If the DEVICE OFF bit in the SES Array Device Slot Element bit indicates a 1 (i.e, PwrDIS signal is HIGH or Device is turned off), then DFC PERST# signal shall be asserted (i.e., LOW).

If the DEVICE OFF bit in the SES Array Device Slot Element bit transitions from 1 to 0 (e.g., The Host is requesting the DFC transition from power off to power on) and the Drive Type Installed field is not set to 0x7 (i.e., DFC Empty), then UBM Controller shall perform the sequence/step/processes as described in Section 5.16 for a newly installed drive/device.

If the DFC Change Count Capability bit (See 7.2.11) indicates no support (i.e., 0), then the DFC Change Count shall be set to a value of 00h.

If the DFC Change Count Capability bit (See 7.2.11) indicates support (i.e., 1), then the DFC Change Count shall be initialized to a value of 01h.

The DFC Change Count field, when supported (See 7.2.11) and read, contains a wrapping counter that increments each time there is a specific DFC change:

a. in the Drive Type Installed field and the Drive Type Installed Change Count Mask bit (See Section 7.2.12) is set to 1 (i.e., Drive Type Installed field change causes an increment in DFC Change Count field),
b. in the PCIe Reset field in a DFC Status and Control Descriptor and the PCIe Reset Change Count Mask bit (See Section 7.2.12) is set to 1 (i.e., PCIe Reset field change causes an increment in DFC Change Count field).

The DFC Change Count field wraps back to 01h after reaching FFh. The DFC Change Count field is read only. Attempts to write the DFC Change Count field by the UBM Host shall be ignored by the UBM Controller.

Note: The DFC Change Count field provides the UBM Host a mechanism to determine if changes have occurred on a specific DFC. This can be used to reduce the number of UBM Controller commands exchanged when a change is detected on the backplane.

Appendix A. (Informative) Host Facing Connector Sideband Signal Assignments

The Host Facing Connector sideband I/O signal assignments are defined in Table 7-53.

Table 7-53 SFF-9402 Sideband Signal Assignments

Sideband	SFF-9402	SFF-TA-1005 (UBM)
SB/VSP 0	2WIRE_SCL	2WIRE_SCL
SB/VSP 1	2WIRE_SDA	2WIRE_SDA
SB/VSP 2	Ground	Ground
SB/VSP 3	Ground	Ground
SB/VSP 4	RESET	2WIRE_RESET#
SB/VSP 5	ADD (Address) SFF-8654 – PERST#	PERST#
SB/VSP 6	CTLR_TYPE / DRV_IN_PLACE#	CPRSNT#/ CHANGE_DETECT#
SB/VSP 7	Backplane Type(1)	Backplane Type(1)
SB/VSP +	RefClk+	RefClk+
SB/VSP -	RefClk-	RefClk-

The HFC PERST# signal (See Section 5.3) indicates the two behaviors of the Host:

a. If HFC PERST# signal is LOW (i.e., Asserted), the Host has not enabled the RefClk and is holding the HFC PERST# signal LOW until the RefClk has stabilized.
b. If the HFC PERST# signal is HIGH (i.e., Deasserted), then the Host has enabled RefClk and has released the HFC PERST# signal.

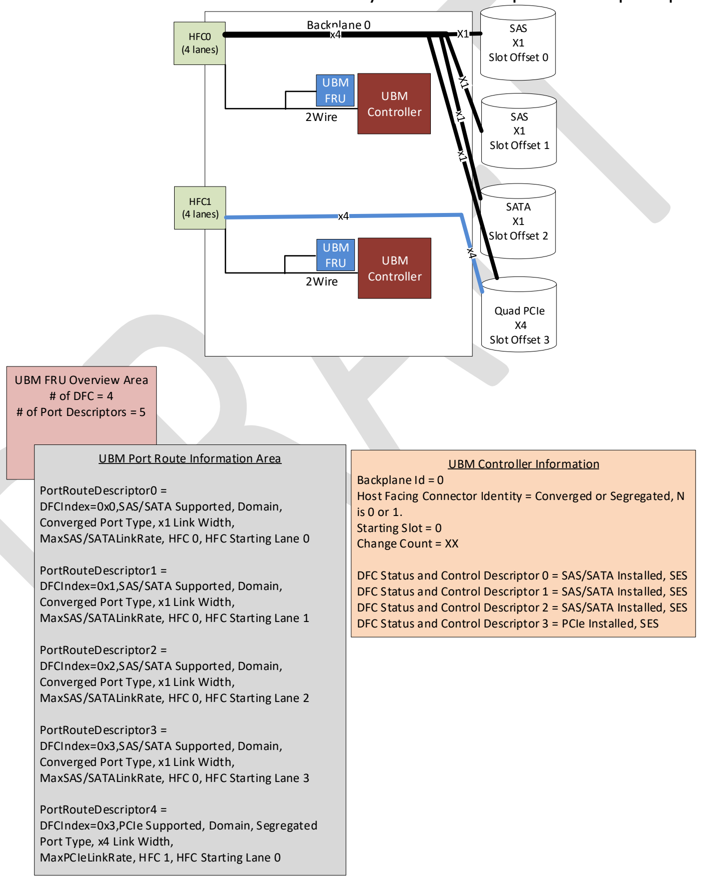
If the Host is supplying RefClk to a HFC and there are no devices installed in the associated DFCs, then the Host should disable the RefClk.

Appendix B. (Informative) Backplane Examples

The examples provided in this section provide a subset of system deployments and does not constitute all system deployments to this standard. The example figures provide a graphical diagram and examples of field settings in the UBM FRU and UBM Controller. If the field depends on the deployment then the field name is used in place of the field value.

B.1. Backplane Routing

This standard provides the ability to describe multiple DFC port routings to support various device attachments. In Figure B-1 a backplane is described that supports SAS and SATA devices via HFC0 and supports one Quad PCIe device in Slot Offset 3 via HFC1. In this example HFC0 indicates it is the converged Port Type via the Host Facing Connector Info Command (See 7.2.8), while HFC1 indicates it is the segregated Port Type. The system designer may choose to implement the UBM FRU identically between HFC0 and HFC1 as is depicted in Figure B-1 or specifically such that the UBM FRU from HFC0 and HFC1 only contains data specific to its port specific DFC routings.



UBM FRU Overview Area
# of DFC = 4
# of Port Descriptors = 5

UBM Port Route Information Area

PortRouteDescriptor0 =
DFCIndex=0x0,SAS/SATA Supported, Domain, Converged Port Type, x1 Link Width, MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 0

PortRouteDescriptor1 =
DFCIndex=0x1,SAS/SATA Supported, Domain, Converged Port Type, x1 Link Width, MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 1

PortRouteDescriptor2 =
DFCIndex=0x2,SAS/SATA Supported, Domain, Converged Port Type, x1 Link Width, MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 2

PortRouteDescriptor3 =
DFCIndex=0x3,SAS/SATA Supported, Domain, Converged Port Type, x1 Link Width, MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 3

PortRouteDescriptor4 =
DFCIndex=0x3,PCIe Supported, Domain, Segregated Port Type, x4 Link Width, MaxPCIeLinkRate, HFC 1, HFC Starting Lane 0

UBM Controller Information

Backplane Id = 0
Host Facing Connector Identity = Converged or Segregated, N is 0 or 1.
Starting Slot = 0
Change Count = XX

DFC Status and Control Descriptor 0 = SAS/SATA Installed, SES
DFC Status and Control Descriptor 1 = SAS/SATA Installed, SES
DFC Status and Control Descriptor 2 = SAS/SATA Installed, SES
DFC Status and Control Descriptor 3 = PCIe Installed, SES

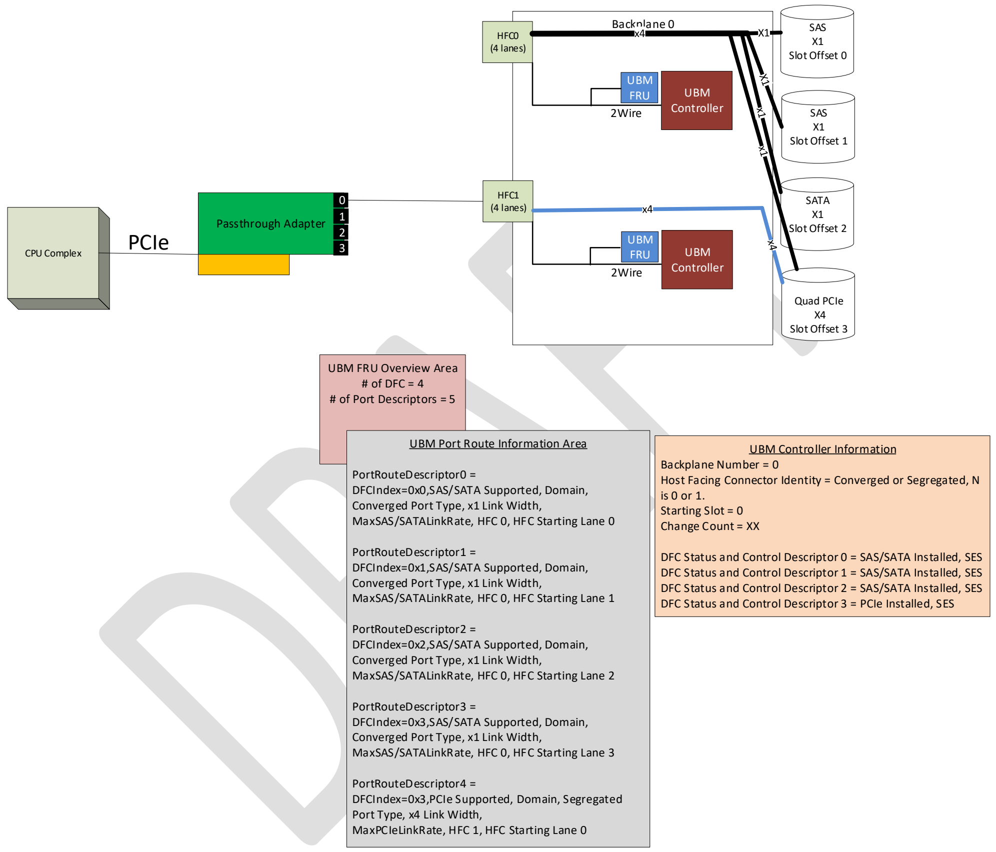
Figure B-1 Multiple DFC Routing Backplane Example

B.2. Adapters cabled to the Backplane

An Adapter can take multiple forms in a system such as:

a. CPU Complex (e.g., Connector on system board or PCIe Passthrough Adapter)
b. PCIe Switch Adapter
c. HBA (e.g. SAS/SATA and/or PCIe capable)

An example of Adapters cabled to the backplane can be found in Figure B-2, Figure B-3 and Figure B-4.



UBM FRU Overview Area
# of DFC = 4
# of Port Descriptors = 5

UBM Port Route Information Area

PortRouteDescriptor0 =
DFCIndex=0x0,SAS/SATA Supported, Domain, Converged Port Type, x1 Link Width, MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 0

PortRouteDescriptor1 =
DFCIndex=0x1,SAS/SATA Supported, Domain, Converged Port Type, x1 Link Width, MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 1

PortRouteDescriptor2 =
DFCIndex=0x2,SAS/SATA Supported, Domain, Converged Port Type, x1 Link Width, MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 2

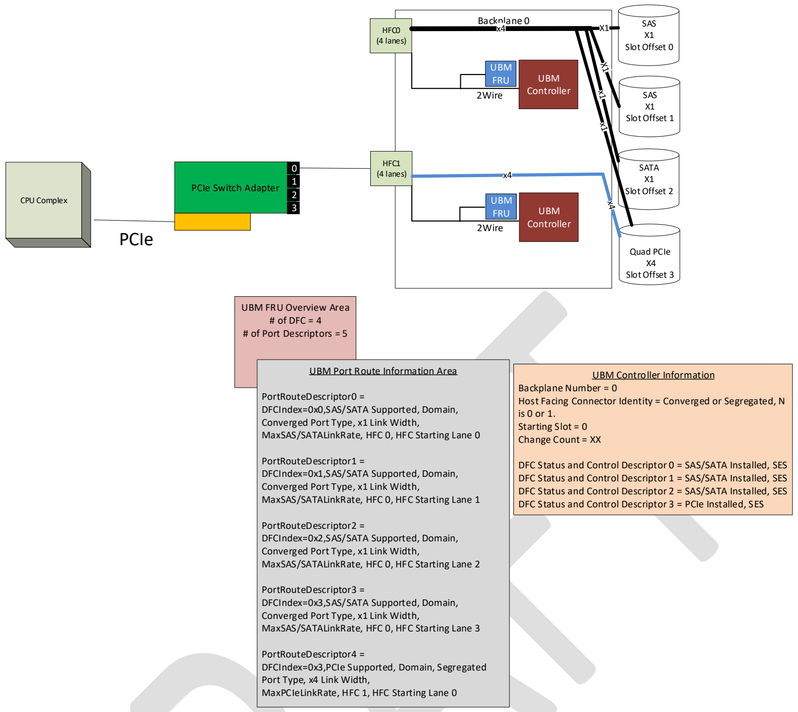
PortRouteDescriptor3 =
DFCIndex=0x3,SAS/SATA Supported, Domain, Converged Port Type, x1 Link Width, MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 3

PortRouteDescriptor4 =
DFCIndex=0x3,PCIe Supported, Domain, Segregated Port Type, x4 Link Width, MaxPCIeLinkRate, HFC 1, HFC Starting Lane 0

UBM Controller Information
Backplane Number = 0
Host Facing Connector Identity = Converged or Segregated, N is 0 or 1.
Starting Slot = 0
Change Count = XX

DFC Status and Control Descriptor 0 = SAS/SATA Installed, SES
DFC Status and Control Descriptor 1 = SAS/SATA Installed, SES
DFC Status and Control Descriptor 2 = SAS/SATA Installed, SES
DFC Status and Control Descriptor 3 = PCIe Installed, SES

Figure B-2 PCIe Passthrough Adapter Cabled to the Backplane Example



UBM FRU Overview Area
# of DFC = 4
# of Port Descriptors = 5

UBM Port Route Information Area

PortRouteDescriptor0 =
DFCIndex=0x0,SAS/SATA Supported, Domain,
Converged Port Type, x1 Link Width,
MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 0

PortRouteDescriptor1 =
DFCIndex=0x1,SAS/SATA Supported, Domain,
Converged Port Type, x1 Link Width,
MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 1

PortRouteDescriptor2 =
DFCIndex=0x2,SAS/SATA Supported, Domain,
Converged Port Type, x1 Link Width,
MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 2

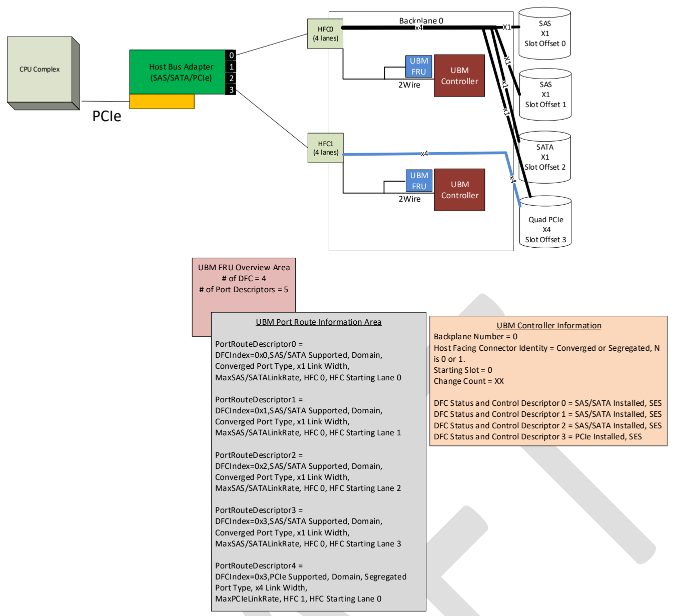
PortRouteDescriptor3 =
DFCIndex=0x3,SAS/SATA Supported, Domain,
Converged Port Type, x1 Link Width,
MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 3

PortRouteDescriptor4 =
DFCIndex=0x3,PCIe Supported, Domain, Segregated
Port Type, x4 Link Width,
MaxPCIeLinkRate, HFC 1, HFC Starting Lane 0

UBM Controller Information
Backplane Number = 0
Host Facing Connector Identity = Converged or Segregated, N is 0 or 1.
Starting Slot = 0
Change Count = XX

DFC Status and Control Descriptor 0 = SAS/SATA Installed, SES
DFC Status and Control Descriptor 1 = SAS/SATA Installed, SES
DFC Status and Control Descriptor 2 = SAS/SATA Installed, SES
DFC Status and Control Descriptor 3 = PCIe Installed, SES

Figure B-3 PCIe Switch Adapter Cabled to the Backplane Example



UBM FRU Overview Area
# of DFC = 4
# of Port Descriptors = 5

UBM Port Route Information Area

PortRouteDescriptor0 =
DFCIndex=0x0,SAS/SATA Supported, Domain,
Converged Port Type, x1 Link Width,
MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 0

PortRouteDescriptor1 =
DFCIndex=0x1,SAS/SATA Supported, Domain,
Converged Port Type, x1 Link Width,
MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 1

PortRouteDescriptor2 =
DFCIndex=0x2,SAS/SATA Supported, Domain,
Converged Port Type, x1 Link Width,
MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 2

PortRouteDescriptor3 =
DFCIndex=0x3,SAS/SATA Supported, Domain,
Converged Port Type, x1 Link Width,
MaxSAS/SATALinkRate, HFC 0, HFC Starting Lane 3

PortRouteDescriptor4 =
DFCIndex=0x3,PCIe Supported, Domain, Segregated
Port Type, x4 Link Width,
MaxPCIeLinkRate, HFC 1, HFC Starting Lane 0

UBM Controller Information
Backplane Number = 0
Host Facing Connector Identity = Converged or Segregated, N is 0 or 1.
Starting Slot = 0
Change Count = XX

DFC Status and Control Descriptor 0 = SAS/SATA Installed, SES
DFC Status and Control Descriptor 1 = SAS/SATA Installed, SES
DFC Status and Control Descriptor 2 = SAS/SATA Installed, SES
DFC Status and Control Descriptor 3 = PCIe Installed, SES

Figure B-4 Host Bus Adapter Cabled to the Backplane Example

B.3. PCIe Switch on the Backplane

In the case of a PCIe Switch implemented on the backplane the UBM FRU and UBM Controller indicate the PCIe Switch HFC link width and port route information. The PCIe Switch based backplane UBM FRU and UBM Controller does not directly provide DFC routings to the HFC, nor does it provide the DFC SES Array Device Slot management. The SES management is provided by the PCIe Switch. The Host can use the link width and port routing to the HFC connectors to configure the PCIe root complex port link widths. An example of the PCIe Switch on the backplane is depicted in Figure B-5.

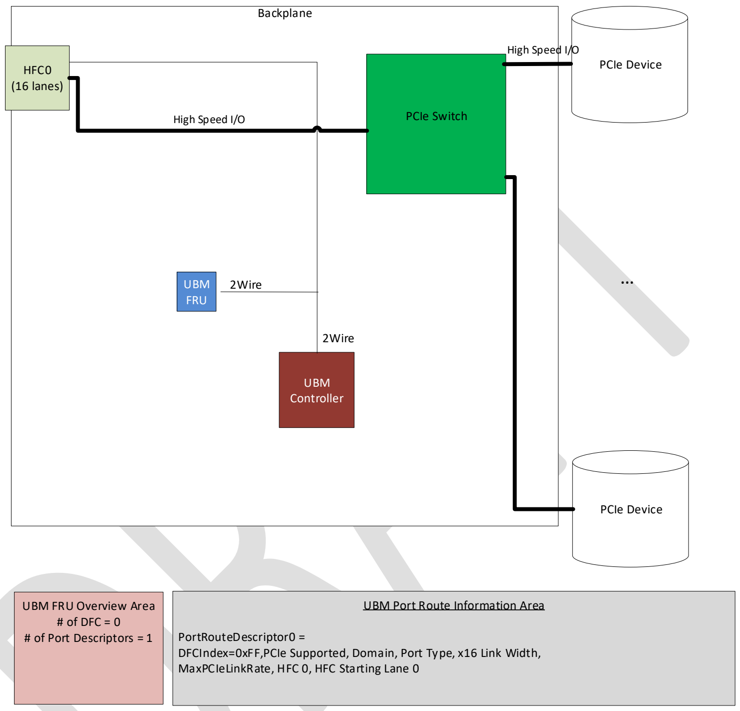
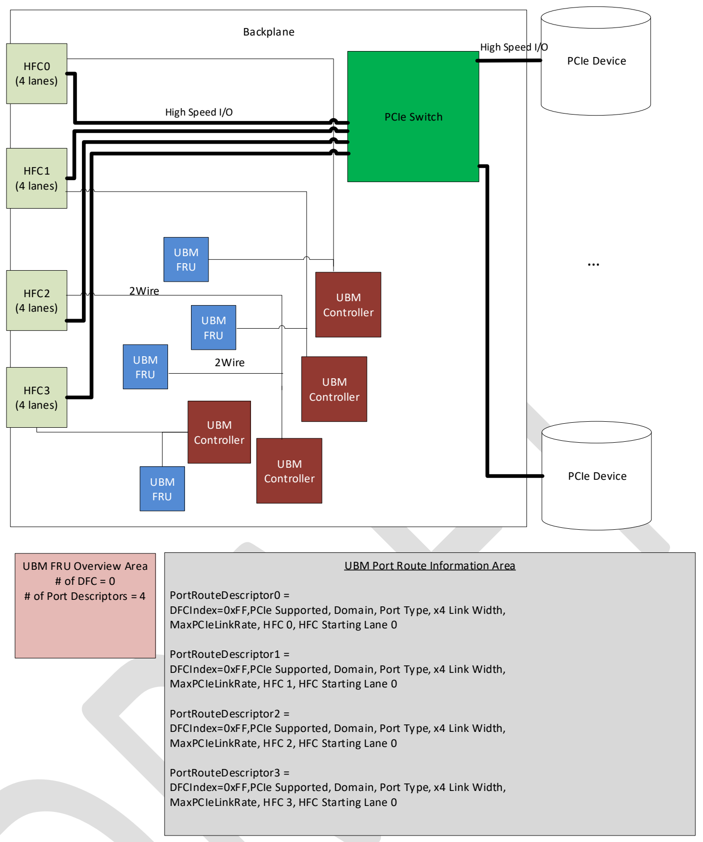


Figure B-5 PCIe Switch on the Backplane Example

In Figure B-6 multiple HFC connectors are implemented from the PCIe Switch. Each HFC provides its corresponding UBM FRU and UBM Controller.



UBM FRU Overview Area
# of DFC = 0
# of Port Descriptors = 4

UBM Port Route Information Area

PortRouteDescriptor0 =
DFCIndex=0xFF,PCIe Supported, Domain, Port Type, x4 Link Width, MaxPCIeLinkRate, HFC 0, HFC Starting Lane 0

PortRouteDescriptor1 =
DFCIndex=0xFF,PCIe Supported, Domain, Port Type, x4 Link Width, MaxPCIeLinkRate, HFC 1, HFC Starting Lane 0

PortRouteDescriptor2 =
DFCIndex=0xFF,PCIe Supported, Domain, Port Type, x4 Link Width, MaxPCIeLinkRate, HFC 2, HFC Starting Lane 0

PortRouteDescriptor3 =
DFCIndex=0xFF,PCIe Supported, Domain, Port Type, x4 Link Width, MaxPCIeLinkRate, HFC 3, HFC Starting Lane 0

Figure B-6 PCIe Switch on the Backplane with Multiple Connectors

B.4. SAS Expander on the Backplane

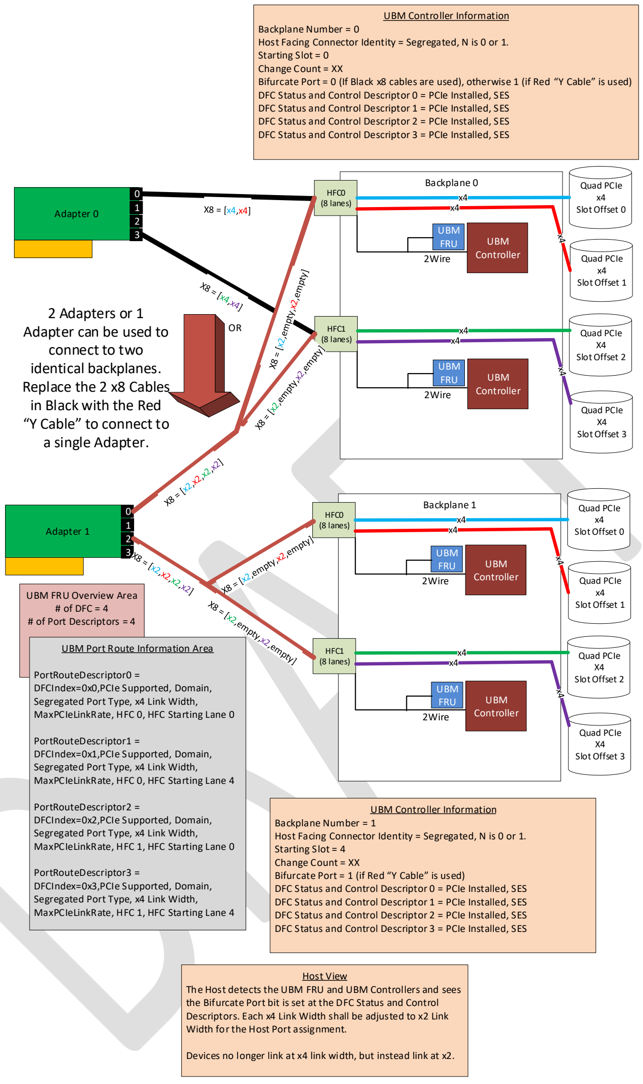
This standard allows for SAS expander backplanes to be described in the same approach as the PCIe Switch on the backplane.

Note: SAS Hosts use Identify Frame and SAS Addresses to detect and configure wide ports. It is not necessary to expressly define them before the port is allowed to link up. The implementation of this standard for a SAS Expander on the backplane may not be necessary.

B.5. Multiple Backplanes in the Chassis

In Figure B-7 two identical backplanes are implemented in the chassis and two adapters are depicted. Adapter 0 is connected to Backplane 0 with two x8 wide cables which provides four devices of x4 link width port routing to the Host. If Adapter 1 is selected to connect to the two backplanes, then a "Y Cable" should be used. The UBM Controllers must indicate the presence of the "Y Cable" by setting the Bifurcate Port bit to 1 in the DFC Status and Control Descriptor (See 7.2.15). The "Y Cable" described in this example replaces the DFC x4 link width port routing with DFC x2 link width routing at the Host Adapter connector. In order to communicate properly with the UBM Controllers and FRU, the "Y Cable" also must route two 2Wire interfaces from the Host (i.e., the HFC0 and HFC1 2Wire interfaces are accessible by the Host via the "Y Cable").

Commented [HA1]: Check is cross reference; link was broken


Adapter 0
0
1
2
3
X8 = [x4,x4]
X8 = [x4,x4]
OR
2 Adapters or 1 Adapter can be used to connect to two identical backplanes. Replace the 2 x8 Cables in Black with the Red "Y Cable" to connect to a single Adapter.
X8 = [x2,empty,x2,empty]
X8 = [x2,empty,x2,empty]
X8 = [x2,x2,x2,x2]
Adapter 1
X8 = [x2,x2,x2,x2]
X8 = [x2,empty,x2,empty]
X8 = [x2,empty,x2,empty]
Backplane 0
HFC0 (8 lanes)
HFC1 (8 lanes)
x4
UBM FRU
UBM Controller
2Wire
Quad PCIe x4 Slot Offset 0
Quad PCIe x4 Slot Offset 1
Quad PCIe X4 Slot Offset 2
Quad PCIe X4 Slot Offset 3
Backplane 1
HFC0 (8 lanes)
HFC1 (8 lanes)
Quad PCIe x4 Slot Offset 0
Quad PCIe x4 Slot Offset 1
Quad PCIe X4 Slot Offset 2
Quad PCIe X4 Slot Offset 3

UBM Controller Information

Backplane Number = 0
Host Facing Connector Identity = Segregated, N is 0 or 1.
Starting Slot = 0
Change Count = XX
Bifurcate Port = 0 (If Black x8 cables are used), otherwise 1 (if Red "Y Cable" is used)
DFC Status and Control Descriptor 0 = PCIe Installed, SES
DFC Status and Control Descriptor 1 = PCIe Installed, SES
DFC Status and Control Descriptor 2 = PCIe Installed, SES
DFC Status and Control Descriptor 3 = PCIe Installed, SES

UBM FRU Overview Area
# of DFC = 4
# of Port Descriptors = 4

UBM Port Route Information Area

PortRouteDescriptor0 = DFCIndex=0x0,PCIe Supported, Domain, Segregated Port Type, x4 Link Width, MaxPCIeLinkRate, HFC 0, HFC Starting Lane 0

PortRouteDescriptor1 = DFCIndex=0x1,PCIe Supported, Domain, Segregated Port Type, x4 Link Width, MaxPCIeLinkRate, HFC 0, HFC Starting Lane 4

PortRouteDescriptor2 = DFCIndex=0x2,PCIe Supported, Domain, Segregated Port Type, x4 Link Width, MaxPCIeLinkRate, HFC 1, HFC Starting Lane 0

PortRouteDescriptor3 = DFCIndex=0x3,PCIe Supported, Domain, Segregated Port Type, x4 Link Width, MaxPCIeLinkRate, HFC 1, HFC Starting Lane 4

UBM Controller Information

Backplane Number = 1
Host Facing Connector Identity = Segregated, N is 0 or 1.
Starting Slot = 4
Change Count = XX
Bifurcate Port = 1 (if Red "Y Cable" is used)
DFC Status and Control Descriptor 0 = PCIe Installed, SES
DFC Status and Control Descriptor 1 = PCIe Installed, SES
DFC Status and Control Descriptor 2 = PCIe Installed, SES
DFC Status and Control Descriptor 3 = PCIe Installed, SES

Host View

The Host detects the UBM FRU and UBM Controllers and sees the Bifurcate Port bit is set at the DFC Status and Control Descriptors. Each x4 Link Width shall be adjusted to x2 Link Width for the Host Port assignment.

Devices no longer link at x4 link width, but instead link at x2.

Y-Cabled System View

Host Adapter	Host 2Wire I/F	Backplane Number	HFC Identity	HFC Starting Lane	Derived Starting Lane	DFC Status and Control Index	Starting Slot	Slot Offset	Derived Slot Location
1	Channel 0	0	0	0	0	0	0	0	0
1	Channel 0	0	0	4	2	1	0	1	1
1	Channel 1	0	1	0	0	2	0	2	2
1	Channel 1	0	1	4	2	3	0	3	3
1	Channel 2	1	0	0	0	0	4	0	4
1	Channel 2	1	0	4	2	1	4	1	5
1	Channel 3	1	1	0	0	2	4	2	6
1	Channel 3	1	1	4	2	3	4	3	7

Figure B-7 Two Identical Backplanes Example

This standard provides for unique or relative Slot assignments via the Slot Offset field, and Starting Slot field returned from the UBM Controller. If multiple backplanes are deployed in the chassis, then the Backplane Number field (See 7.2.9) must be unique among all backplanes in the chassis. If the Starting Slot fields are the same amongst multiple backplanes then the system designer should assign unique Slot Offset assignments, otherwise the Slot Offset field is determined per Section 5.12. If the system designed intends to have duplicate Derived Actual slot locations, the duplicating backplane should indicate a different Backplane Type field from the other backplanes.

Appendix C. (Informative) Host Considerations

The Host should consider the following to ensure the implementation will interoperate with a large variety of UBM Backplanes:

1. The 2Wire Communication should occur at the slowest supported 2Wire device rate for the 2Wire devices on the bus.

   An example of this would be: if the 2Wire topology includes a 2Wire Mux @ 100kHz, UBM FRU @ 100kHz, UBM Controller @ 400kHz, then the Host should utilize 100 kHz for communication with all devices on the 2Wire channel.

2. The UBM FRU represents a 256 byte EEPROM. There is potential for the UBM FRU to be emulated in a programmable device. It is recommended to process the UBM FRU in multiple 2Wire transactions to account for various programmable device 2Wire service rates.

   An example of this would be to perform 8 transactions of 32 bytes to read the entire UBM FRU.

3. The Host should support the 2Wire Clock Stretching feature. Support of this feature allows for a large selection of 2Wire Slave components including microcontrollers, CPLDs and ASICs.