Universal Serial Bus I3C® Device Class Specification

Version 1.0
January 07, 2022

I3C® is a registered service mark of MIPI Alliance, Inc.

Revision History

Version	Comments	Issue Date
1.0	Initial Release	January, 2022

Please send comments via electronic mail to:
techadmin@usb.org

CONTENTS

FIGURES

TABLES

Specification Work Group Contributors

Analogix Semiconductor, Inc.	Greg Stewart		
Apple Inc.	James Orr		
Intel Corporation	Matthew A. Schnoor	Aruni P. Nelson (Editor)	Abdul R. Ismail
	John S. Howard	Rolf H. Kühnis	Brad Saunders
	Stephanie S. Wallick	Ashwin Umapathy	Enrico D. Carrieri
	Bradley H. Smith	Tomasz Pielaszkiewicz	Robert E. Gough
	Rajesh Bhaskar	Felipe Balbi	Christophe G. Zeitouny
	Xiang A. Wang	Patrik Eder	Vui Yong Liew
	Abhishek Pandey	Sathyanarayanan Ramesh Babu	Mayank L. Malde
	Ashutosh Singh	Noah Bacon	Cristian Culman
	Keith A. Jones	Eric A. Upson	Shrinidhi Kulkarni Venkatesh
Microsoft Corporation	Toby Nixon	Dan Iatco	
NXP Semiconductors	Paul Kimelman		
Synopsys, Inc.	Suresh Venkatachalam		
USB-IF	Paul E. Berg		

1 Introduction

This specification describes the USB I3C Device Class which exposes the functionality of an I3C Controller and the connected Target devices on the I3C Bus over the USB interface to a Host system.

1.1 Scope

This specification describes the following:

- An interface to expose and configure the I3C Function within a USB Device to allow interaction between Host software and the I3C Device, to drive transactions on I3C Bus to/from Target devices. This I3C Function can either be the sole USB Function within a USB Device, or a USB Function within a multi-function or "composite" USB Device.
- An operational model consisting of USB Control, Bulk, and Interrupt transfers to communicate with the USB I3C Device.
- Data Structures, Command and Response Descriptors used in the above Operational model.

Any mechanism to update the firmware of such a USB Device and the hardware implementation (including the interface between the I3C Function and the I3C Device) is beyond the scope of this specification.

Readers are expected to be familiar with the specifications listed in the Related Documents section.

If there are conflicts between this specification and the specifications listed in the Related Documents section, the specifications in the Related Documents sections shall take precedence on all issues of conflict.

1.2 Purpose

This specification allows the development of a Host software stack to control and communicate with the USB I3C Device.

1.3 Related Documents

- [MIPII3C] – MIPI® I3C® Specification as a generic reference, either for version 1.0 or version 1.1.1 as chosen by the implementer.
    - MIPI Alliance Specification for I3C (Improved Inter Integrated Circuit), version 1.0, *MIPI Alliance, Inc.,* 23 December 2016 (MIPI Board Adopted 31 December 2016).
    - MIPI Alliance Specification for I3C (Improved Inter Integrated Circuit), version 1.1.1, *MIPI Alliance, Inc.,* 11 June 2021 (MIPI Board Adopted 8 June 2021).
    - The latest version of the MIPI I3C Specification is available to MIPI Alliance member companies, at https://www.mipi.org/specifications/i3c-sensor-specification.
- [MIPII3CBASIC] – MIPI® I3C Basic℠ Specification as a generic reference, either for version 1.0 or version 1.1.1 as chosen by the implementer.
    - MIPI Alliance Specification for I3C Basic (Improved Inter Integrated Circuit), version 1.0, *MIPI Alliance, Inc.,* 19 July 2018 (MIPI Board Adopted 8 October 2018).
    - MIPI Alliance Specification for I3C Basic (Improved Inter Integrated Circuit), version 1.1.1, *MIPI Alliance, Inc.,* 9 June 2021 (MIPI Board Adopted 23 July 2021).
    - The latest version of the MIPI I3C Basic Specification is available at https://resources.mipi.org/mipi-i3c-basic-download.
    - The MIPI I3C Basic Specification is a subset of the MIPI I3C Specification. Throughout this Device Class Specification, most references to either of the MIPI I3C specifications will only use [MIPII3C] for brevity. Specific referred section numbers of [MIPII3C] may also be substituted with the same section numbers of [MIPII3CBASIC], unless a specific feature of an I3C Controller or I3C Target requires certain definitions that are not included in MIPI I3C Basic. Implementers of a USB I3C Device may choose to use either MIPI I3C or MIPI I3C Basic.

- [MIPII3CHCI] – MIPI® I3C Host Controller Interface (I3C HCI℠) Specification (referred to in this document as the I3C HCI Specification), either for version 1.0 or version 1.1 as chosen by the implementer.
    - MIPI Alliance Specification for I3C Host Controller Interface (I3C HCI), version 1.0, *MIPI Alliance, Inc.,* 29 September 2017 (MIPI Board Adopted 4 April 2018).
    - MIPI Alliance Specification for I3C Host Controller Interface (I3C HCI), version 1.1, *MIPI Alliance, Inc.,* 20 May 2021 (MIPI Board Adopted 20 May 2021).
    - The latest version of the MIPI I3C HCI Specification is available at https://mipi.org/specifications/i3c-hci).
- [MIPIDISCOI3C] – MIPI® Discovery and Configuration (DisCo℠) Specification for I3C, version 1.0.
    - MIPI Alliance Discovery and Configuration (DisCo) Specification for I3C, version 1.0, *MIPI Alliance, Inc.,* 23 January 2019 (MIPI Board Adopted 18 June 2019).
    - The latest version of the MIPI DisCo Specification for I3C is available at https://mipi.org/specifications/mipi-disco-i3c.
- [USB2.0] – Universal Serial Bus Specification, Revision 2.0, (including errata and ECNs through June 27, 2017) (referred to in this document as the USB 2.0 Specification) (available at http://www.usb.org/developers/docs).
- [USB3.2] – Universal Serial Bus 3.2 Specification, Revision 1.0, (including errata and ECNs through July 16, 2020) (referred to in this document as the USB 3.2 Specification) (available at http://www.usb.org/developers/docs).

Note: MIPI® and I3C® are registered service marks of MIPI Alliance, Inc. I3C Basic℠, I3C HCI℠ and DisCo℠ are service marks of MIPI Alliance, Inc.

1.4 Terms and Abbreviations

This section defines terms and abbreviations used throughout this document. For additional terms and abbreviations that pertain to the Universal Serial Bus, see Chapter 2, "Terms and Abbreviations," in [USB2.0] and [USB3.2].

Table 1-1: Terms and Abbreviations

Term	Description
ACK'd	Acknowledged
Active I3C Controller	I3C Device that currently has the control of the I3C Bus, that is the I3C Device is in the I3C Controller Role. This I3C Device could be either the Primary I3C Controller or a Secondary I3C Controller. Refer Section 2 of [MIPII3C] for additional information.
CCC	Common Command Code. Common Command Codes are standard MIPI I3C Commands. Refer Section 5.1.9 of [MIPII3C] for details. Some of these CCCs are Required and some are Conditional or Optional for I3C Target devices.
DisCo	Discovery and Configuration is MIPI-defined set of properties for I3C Host Controllers and Target devices on the I3C Bus.
ENTDAA	Enter Dynamic Address Assignment. This is a Broadcast CCC to indicate to all I3C Target devices that the I3C Controller requires them to enter the Dynamic Address Assignment procedure (refer Section 5.1.4 in [MIPII3C]).
HDR	High Data Rate mode defined by [MIPII3C].
HDR-BT	HDR Bulk Transfer mode defined by [MIPII3C].
HDR-DDR	HDR Double Data Rate mode defined by [MIPII3C].

Term	Description
HDR-TS	HDR-Ternary Symbol mode which could be Legacy or Pure Bus mode defined by [MIPII3C]. *Note: The I3C Controller determines whether Legacy or Pure Bus mode should be used for a transfer.*
Hot-Join	I3C Targets that join the I3C Bus after the I3C Bus is configured. The Hot-Join mechanism allows the I3C Target device to notify the I3C Controller that it is ready to receive a Dynamic Address. Refer Section 5.1.5 of [MIPII3C].
I2C Target	The legacy I2C devices connected on the I3C Bus. Refer Section 2 of [MIPII3C] for additional information.
I3C Controller	I3C Controller is a device on the I3C Bus that controls the I3C Bus. *Note: Terms "Primary I3C Controller" and "I3C Controller" are interchangeably used throughout this document.* Refer Section 2 and Section 4.2 of [MIPII3C] for additional details.
I3C Target	The I3C devices connected on the I3C Bus. These devices may be Secondary I3C Controller capable. Refer Section 2 of [MIPII3C] for additional information.
IBI (for I3C)	In-Band Interrupt, a type of Target device Interrupt request that an I3C Target device can initiate on the I3C Bus.
NACK'd	Not Acknowledged
Primary I3C Controller	Primary I3C Controller is the device on the I3C Bus responsible for I3C Bus initialization and configuration. Primary I3C Controller may relinquish control of the I3C Bus to another Controller role capable Device (i.e., a Secondary I3C Controller), thereby becoming a Secondary I3C Controller. When I3C Device role is I3C Controller it shall be the Primary I3C Controller. Refer Section 2 of [MIPII3C] for additional information.
PID	Provisional ID
Repeated Start	Two or more instances of a Start in a row without an intervening Stop. A Repeated Start is used in circumstances where the I3C Controller wishes to retain the control of the I3C Bus to continue communicating on the I3C Bus.
RSTACT	Reset Action. This Broadcast or Directed CCC is used to configure the Target Reset Action.
RSTDAA	Reset Dynamic Address Assignment. This Broadcast CCC indicates to all I3C Target devices that the I3C Controller requires them to clear/reset their I3C Controller assigned Dynamic Address.
SDR	Single Data Rate defined by [MIPII3C].
Secondary I3C Controller	Secondary I3C Controller is a device on the I3C Bus which functions as a Target device but can also request control of the I3C Bus as an I3C Controller. Refer Section 2 of [MIPII3C] for additional information.
SETDASA	Set Dynamic Address from Static Address. This CCC is used by the I3C Controller to set the dynamic address of one I3C Target device its static addresses if static address is available. Refer Section 5.1.9.3.10 of [MIPII3C] for additional information.
SETAASA	Set All Addresses to Static Address. This CCC is used by the I3C Controller to set the dynamic addresses of all I3C Target devices to their static addresses if static addresses are available. Refer Section 5.1.9.3.23 of [MIPII3C] for additional information.
Start	Start is the I3C Bus condition of a High to Low transition on the SDA line while the SCL line remains High.
Stop	Stop is the I3C Bus condition of a Low to High transition on the SDA line while the SCL line remains High.
UDR	User Defined Rate. These are user defined transfer rates for I2C and I3C.

1.5 Conventions and Notations

1.5.1 Precedence

If there is a conflict between text, figures, and tables, the precedence shall be tables, figures, and then text.

1.5.2 Keywords

The following keywords differentiate between the levels of requirements and options.

1.5.2.1 Informative

Informative is a keyword that describes information with this specification that intends to discuss and clarify requirements and features as opposed to mandating them.

1.5.2.2 May

May is a keyword that indicates a choice with no implied preference.

1.5.2.3 N/A

N/A is a keyword that indicates that a field or value is not applicable and has no defined value and shall not be checked or used by the recipient.

1.5.2.4 Normative

Normative is a keyword that describes features that are mandated by this specification.

1.5.2.5 Optional

Optional is a keyword that describes features not mandated by this specification. However, if an optional feature is implemented, the feature shall be implemented as defined by this specification (optional normative).

1.5.2.6 Reserved

Reserved is a keyword indicating reserved bits, bytes, words, fields, and code values that are set-aside for future standardization. Their use and interpretation may be specified by future extensions to this specification and, unless otherwise stated, shall not be utilized, or adapted by vendor implementation. For registers or fields in a fixed-length data structure, a reserved bit, byte, word, or field shall be set to zero by the sender and shall be ignored by the receiver. For fields in a variable-length data structure, reserved fields at the end of such a data structure shall not be sent by the sender, and the receiver shall ignore any additional data past the end of the defined fields.

1.5.2.7 Shall

Shall is a keyword indicating a mandatory (normative) requirement. Designers are mandated to implement all such requirements to ensure interoperability with other compliant Devices.

1.5.2.8 Should

Should is a keyword indicating flexibility of choice with a preferred alternative. Equivalent to the phrase “it is recommended that”.

1.5.3 Numbering

Numbers that are immediately followed by a lowercase “b” (e.g., 01b) are binary values. Numbers that are immediately followed by an uppercase “B” are byte values. Numbers that are immediately followed by a lowercase “h” (e.g., “3Ah”) are hexadecimal values. Numbers not immediately followed by either a “b”, “B”, or “h” are decimal values. Binary or hexadecimal values may be “zero-padded” (i.e., preceded by zero bits), to illustrate the size of a bitfield or other data entity.

Additionally, some numeric constants which are directly referenced from various MIPI specifications may instead use Verilog-style literals and shall specify the exact bit width of a bitfield as well as the numeric base of the following numbers. For example, “1’b1” or “7’h7E” may represent binary or hexadecimal literals (respectively) of a fixed width.

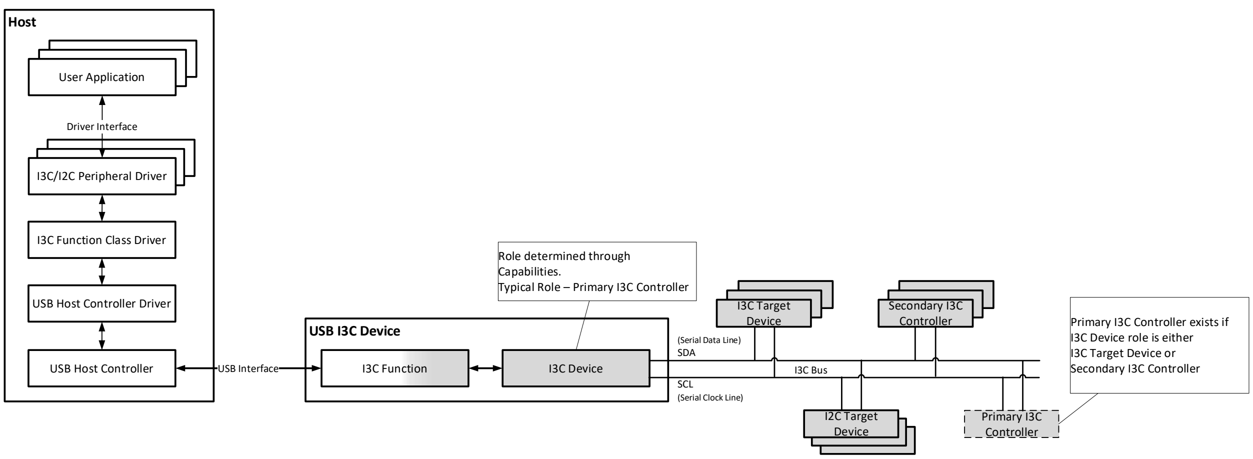
1.5.4 Byte Ordering

All multiple byte fields in this specification are moved over the USB bus in little-endian order, i.e., the least significant byte (LSB) first and the most significant byte (MSB) last unless otherwise specified.

1.5.5 Use of terms “Host”, “I3C Function” and “I3C Device”

Figure 1-1 illustrates the topology and logical composition of the Host, the USB Device with I3C Function, the I3C Device, and other Target Devices on the I3C Bus.

Figure 1-1: Host, USB Device, I3C Device and Target Devices



*Note: * An I3C Device can take the role of an I3C Controller (Primary I3C Controller), I3C Target device, or I3C Target device capable of I3C Secondary Controller role.*

The Host’s logical composition as shown in Figure 1-1 includes the following:

- USB Host Controller.
- Host software (USB Driver Stack, I3C Function Class Driver, any I3C or I2C Peripheral Driver, Driver Interface, and User Application).

The USB Device’s logical composition as shown in Figure 1-1 includes the following:

- I3C Function (Provides interface between the Host and the I3C Device).
- I3C Device (May act as a Primary I3C Controller, an I3C Target device, or an I3C Target device capable of Secondary I3C Controller Role).
- Interface to an I3C Bus, which may have other I3C Target devices, I2C Target devices, one or more Secondary I3C Controllers, and a Primary I3C Controller as indicated in Figure 1-1.

2 Management Overview

This specification defines a USB Device that exposes an I3C Function. Such a USB Device includes an I3C Device which may take the role of Primary I3C Controller that configures and controls the I3C Bus connected to its I3C Device. Refer Figure 1-1.

While the detection and enumeration of such USB Device occurs through the standard USB control transfers, the initialization and configuration of the I3C Function occurs through USB I3C class specific requests.

The I3C Function uses USB's Control, Bulk, and Interrupt endpoints to send requests and receive responses from the Target devices.

Control transfers are used to get the I3C capabilities of the I3C Device. The I3C Function supports requests to read the I3C capability, by returning a data structure that indicates the role of its I3C Device and the subsequent initialization and transactions on the I3C Bus that it will perform. Refer Section 4 for details.

If the I3C Device supports either the I3C Controller role or the I3C Secondary Controller role, any IBIs received on the I3C Bus are sent to the Host through Interrupt notifications by the I3C Function.

Bulk request and response transfers can either be a strictly ordered list of I3C Commands and the corresponding I3C responses, or they can contain a single I3C Command and the corresponding I3C response. Bulk response transfers may also contain data corresponding to a specific IBI. Refer Section 4.6 for details.

The I3C Function acts as an intermediary between the Host and the other I3C Devices on an I3C Bus. The I3C Function communicates with the Host using the USB protocol to exercise the MIPI-defined I3C Commands, Responses, Read and Write transfers.

The Host sends requests and responses to the I3C Function in USB format, and the I3C Function translates these requests and submits the I3C Commands to its I3C Device. Based on the characteristics and capabilities of the I3C Bus and its Target devices, the I3C Device may decide if a command or transfer can be executed. Similarly, the I3C Function receives responses from its I3C Device and translates them to USB format so that these can be transferred to the Host.

3 USB Descriptors and Requests

This section defines the data structures, descriptors and requests used to communicate between a Host and an I3C Function.

3.1 Descriptors

The USB I3C Device shall support the following USB descriptors as per USB3.2 and USB2.0 specifications:

- **Device:** Fields *bDeviceSubClass and bDeviceProtocol* of the Device descriptor shall be set to 00h, since it is recommended that *bDeviceClass* be set to 00h. Refer defined class codes at USB.org.

- **Configuration:** The Configuration descriptor shall contain at least one Interface descriptor.

- **Interface:** An Interface descriptor shall exist for each such I3C Function exposed by the Device. Each Interface Descriptor shall have fields *bNumEndpoints* set to 0x3, *bInterfaceClass* set to USBI3CDEVICE_CLASS (See Appendix), *bInterfaceSubClass* set to USBI3CDEVICE_SUBCLASS (See Appendix) and *bInterfaceProtocol* set to USBI3CDEVICE_PROTOCOL (See Appendix).
- **Endpoint:** The USB Device with an I3C Function shall support Control, Interrupt-IN, Bulk-IN and Bulk-OUT Endpoints.

  - The Interrupt-IN endpoint is used by I3C Function to send an event or status information to the Host. Such notifications may require an action from the Host. Field *wMaxPacketSize* of Interrupt-IN endpoint descriptor shall be limited to 64 bytes.

  - The Bulk-IN endpoint is used by I3C Function to send data and status to the Host. When operating in full speed mode the *wMaxPacketSize* of the Bulk-IN endpoint descriptor shall be set to 64 bytes.

  - The Bulk-OUT endpoint is used by the Host to send commands and data to the I3C Function. When operating in full speed mode the *wMaxPacketSize* of the Bulk-OUT endpoint descriptor shall be set to 64 bytes.

For additional information about these and other standard descriptors, see Chapter 9, "USB Device Framework," of [USB2.0] or [USB3.2].

If the USB Device supports USB 3.2 Enhanced SuperSpeed, these endpoints may be implemented with support for bursting. To enable bursting, the Device must also implement a SuperSpeed Endpoint Companion Descriptor for each endpoint, with a suitable value for field '*bMaxBurst*'. (See [USB3.2] Section 9.6.7, Table 9-27)

3.1.1 Descriptors for SuperSpeed and Enhanced SuperSpeed

If a USB Device that supports I3C Functions also supports USB 3.2 at SuperSpeed or any of its variants or enhancements, then it must also implement additional descriptors as part of full compliance with [USB3.2]. These additional descriptors shall include the following, as applicable:

- USB 2.0 Extension.
- SuperSpeed USB Device Capability.
- SuperSpeedPlus USB Device Capability.
- SuperSpeed Endpoint Companion.

3.2 Standard Requests

The USB I3C Device Class shall support at the minimum the following standard requests described in Chapter 9, "USB Device Framework," of [USB2.0] or [USB3.2].

- Get Configuration
- Get Descriptor
- Get Status
- Set Configuration
- Set Feature
- Set Interface

3.3 Class-Specific Requests

A USB I3C Device Class supports class-specific requests. Table 3-1 lists the supported class-specific requests.

Table 3-1: Class-Specific Requests

bmRequestType	bRequest	wValue	wIndex		wLength	Data
0.01.00001B	CLEAR_FEATURE	Selector	Zero/ Target Address	Interface	Zero	None
0.01.00001B	SET_FEATURE	Selector	Zero/ Target Address	Interface	Zero	None
1.01.00001B	GET_I3C_CAPABILITY	Zero	Reserved	Interface	Varies	I3C Capability
0.01.00001B	INITIALIZE_I3C_BUS	Address Assignment Mode	Reserved	Interface	Varies	Target Device Table
1.01.00001B	GET_TARGET_DEVICE_TABLE	Zero	Reserved	Interface	Varies	Target Device Table
0.01.00001B	SET_TARGET_DEVICE_CONFIG	Zero	Reserved	Interface	Varies	Target Device Configuration
0.01.00001B	CHANGE_DYNAMIC_ADDRESS	Zero	Reserved	Interface	Varies	Address Change Data
1.01.00001B	GET_ADDRESS_CHANGE_RESULT	Zero	Reserved	Interface	Varies	Address Change Result
1.01.00001B	GET_BUFFER_AVAILABLE	Zero	Reserved	Interface	Four	Buffer Available
0.01.00001B	CANCEL_OR_RESUME_BULK_REQUEST	Action	Reserved	Interface	Zero	None

Table 3-2: USB I3C Device Class Specific Request Codes

Class Specific Request	Value
Reserved	00h

Class Specific Request	Value
CLEAR_FEATURE	01h
Reserved	02h
SET_FEATURE	03h
GET_I3C_CAPABILITY	04h
INITIALIZE_I3C_BUS	05h
GET_TARGET_DEVICE_TABLE	06h
SET_TARGET_DEVICE_CONFIG	07h
CHANGE_DYNAMIC_ADDRESS	08h
GET_ADDRESS_CHANGE_RESULT	09h
GET_BUFFER_AVAILABLE	0Ah
CANCEL_OR_RESUME_BULK_REQUEST	0Bh

3.3.1 Events and their effect on Device parameters

This section lists (refer Table 3-3) the various entities associated with USB Device with I3C Function and the effect on those entities when the USB Device receives a control transfer command or when it observes a USB reset.

Table 3-3: I3C related Entities and Events

Entity	Events											
	Warm Reset	**Hot Reset**	**Dis-connect**	**Set Interface**	**Clear Feature (I3C Bus)**	**Clear Feature (remote wakes)**	**Configure I3C Controller**	**Initialize I3C Bus**	**Set Feature (Reset I3C Bus)**	**Set Feature (Reset I3C Target device)**	**Set Feature (remote wakes)**	**Change Dynamic Address**
I3C Function	Def	Def	Def	Def								
I3C Device (I3C Controller role)	Def	Def	Def	Def			Mod	Mod				
Function Remote Wake	Def	Def	Def	Def		Mod					Mod	
Target Device Table	Def	Def	Def	Def	Def			Mod				Mod
I3C Bus (I3C Device in I3C Controller role)	Def	Def	Def	Dis	Dis			Active	Mod			
I3C Target devices (I3C Device in I3C Controller role)	Def	Def	Def					Mod		Mod		Mod

Note:

Active (Act): *I3C Bus, I3C Target device initialization complete, and I3C Bus transactions are valid*
Default (Def): *I3C Bus is active, any dynamic address assignments and modified Target device configurations are lost.*
Disabled (Dis): *I3C Bus is inactive. No I3C bus transactions are possible.*
Modified (Mod)*:* *Changed from default state.*

3.3.2 I3C Function Management

This section describes the class-specific requests used by the Host to configure and initialize the I3C Function, obtain the capabilities of the internal I3C Device, and interact with the Target devices on the I3C Bus.

The I3C Function shall interface with its internal I3C Device (i.e., within the USB Device) to coordinate Requests from the Host and Responses to the Host. The implementation details of the interface between the I3C Function and its internal I3C Device are beyond the scope of this specification.

3.3.2.1 Cancel or Resume Bulk Request

The CANCEL_OR_RESUME_BULK_REQUEST request described in Table 3-4 is used for two purposes:

- Cancel execution of all dependent commands after a stalled command is encountered; cancelled commands include commands in subsequent Bulk request transfers with *Dependent On Previous* field set to 1b (refer Table 3-30).

  Note: Cancellation shall affect all subsequent and consecutive Bulk request transfers with Dependent On Previous field set to 1b, which includes all Bulk request transfers that have been received by the Bulk-OUT endpoint and those currently in transit to Bulk-OUT endpoint, as long as they form a continuous sequence from the Bulk request transfer that had the stalled command. Cancellation shall end at the first Bulk request transfer with Dependent on Previous field set to 0b.

- Resume execution of commands from the stalled command in Bulk request transfer.

Table 3-4: CANCEL_OR_RESUME_BULK_REQUEST Request Fields

bmRequestType	bRequest	wValue	wIndex		wLength	Data
0.01.00001B	CANCEL_OR_RESUME_BULK_REQUEST	Action	Reserved	Interface	Zero	None

Default state: Request is invalid, and Device shall respond with a Request Error.

Address state: Request is invalid, and Device shall respond with a Request Error.

Configured state: Request is valid.

It is a Request Error if *wValue, wIndex* or *wLength* are other than as specified above.

Action	wValue	Description
CANCEL_BULK_REQUEST	0h	This value indicates the I3C Function shall: - Cancel all dependent commands in this Bulk request and all subsequent Bulk request transfers with *Dependent On Previous* field set to 1b (refer Table 3-30). - Clear the Bulk request with the stalled command, for all subsequent and consecutive Bulk request transfers with *Dependent On Previous* field set to 1b (refer Table 3-30) from the Bulk-OUT endpoint. The I3C Controller shall not execute any dependent commands that are cancelled. The I3C Function shall generate Bulk response structures for such dependent commands affected by cancellation and mark them as “not attempted” (i.e., *Attempted* field set to 0b; refer Table 3-33). For any other Bulk requests without dependency (i.e., starting at the first Bulk request with *Dependent On Previous* field set to 0b; refer Table 3-30) the transfer commands may still be executed by the I3C Controller.

RESUME_BULK_REQUEST	1h	The I3C Controller shall resume operations, by retrying the stalled command, and if it is successful the I3C Controller shall continue to execute subsequent dependent commands. After resuming, note that if the retried command fails with the same error, then the I3C Function shall stall execution again, and shall generate another stall notification (refer Section 3.4.1.6).

3.3.2.2 Change Dynamic Address

The CHANGE_DYNAMIC_ADDRESS request described in Table 3-5 is used to change the previously assigned Dynamic Address of one or more I3C Target devices. This request is applicable when the I3C Device is the Active I3C Controller.

Note: This request instructs the I3C Device to use the SETNEWDA CCC (refer [MIPII3C] Section 5.1.9.3.11) with one or more iterations, based on the contents of the data structure sent in the Data stage of this request.

Table 3-5: CHANGE_DYNAMIC_ADDRESS Request Fields

bmRequestType	bRequest	wValue	wIndex		wLength	Data
0.01.00001B	CHANGE_DYNAMIC_ADDRESS	Zero	Reserved	Interface	Varies	Address Change Data

Default state: Request is invalid, and Device shall respond with a Request Error.

Address state: Request is invalid, and Device shall respond with a Request Error.

Configured state: Request is valid.

It is a Request Error if *wValue, wIndex* or *wLength* are other than as specified above.

Refer Section 3.4.2 for data sent by the Host in the Data stage of this request. Refer Section 4.2 for details on changing the dynamic address of I3C Target devices.

3.3.2.3 Clear I3C Feature

The CLEAR_FEATURE request described in Table 3-6 is used to disable features defined by the value of Selector (refer Table 3-7). This request is applicable when the I3C Device is the Active I3C Controller.

Table 3-6: CLEAR_FEATURE Request Fields

bmRequestType	bRequest	wValue	wIndex		wLength	Data
0.01.00001B	CLEAR_FEATURE	Selector	Zero/ Target Address	Interface	Zero	None

Default state: Request is invalid, and Device shall respond with a Request Error.

Address state: Request is invalid, and Device shall respond with a Request Error.

Configured state: Request is valid.

It is a Request Error if *wValue, wIndex* or *wLength* are other than as specified above.

Table 3-7 lists the I3C clear feature selector definitions.

Table 3-7: I3C Clear Feature Selector Values

Selector	wValue	Description
Reserved	0h	Reserved
I3C_BUS	1h	Disables I3C Bus.
I3C_CONTROLLER_ROLE_HANDOFF	2h	Disables handoff of I3C Controller role to Secondary I3C Controller
REGULAR_IBI	3h	Disables regular In-Band Interrupts on the I3C Bus
HOT_JOIN	4h	Disables Hot-Join feature on the I3C Bus.
Reserved	5h	Reserved
REGULAR_IBI_WAKE	6h	Disables remote wake from a regular IBI from a Target device on the I3C Bus.
HOT_JOIN_WAKE	7h	Disables remote wake from a Hot-Join request on the I3C Bus.
I3C_CONTROLLER_ROLE_REQUEST_WAKE	8h	Disables remote wake from an I3C Controller role request on the I3C Bus.
HDR_MODE_EXIT_RECOVERY	9h	Forces all I3C Target devices to exit HDR mode.

3.3.2.3.1 Disable I3C Bus

The CLEAR_FEATURE request described in Table 3-6 is used to disable the I3C Bus. To disable the I3C Bus, the Host shall set the Selector value to I3C_BUS (refer Table 3-7), the least significant byte of *wIndex* to the index of the Interface and the most significant byte of *wIndex* to Zero.

Upon receiving this request, the I3C Function performs the following operations:

- Issue RSTDAA broadcast command on the I3C Bus to clear/reset all I3C Controller assigned Dynamic Addresses;
- Clear the stored Target Device Table; and
- Disable I3C Bus.

3.3.2.3.2 Disable I3C Controller role handoff

The CLEAR_FEATURE request described in Table 3-6 is used to disable handoff of the I3C Controller role to any other I3C Controller-capable devices on the I3C Bus. To disable I3C Controller role handoff, the Host shall set the Selector value to I3C_CONTROLLER_ROLE_HANDOFF (refer Table 3-7), the least significant byte of *wIndex* to the index of the Interface and the most significant byte of *wIndex* to Zero.

3.3.2.3.3 Disable all regular In-Band Interrupts on I3C Bus

The CLEAR_FEATURE request described in Table 3-6 is used to disable all regular In-Band Interrupts from I3C Target devices on the I3C Bus. To disable regular In-Band Interrupts, the Host shall set the Selector value to REGULAR_IBI (refer Table 3-7), the least significant byte of *wIndex* to the index of the Interface and the most significant byte of *wIndex* to Zero.

3.3.2.3.4 Disable Hot-Join

The CLEAR_FEATURE request described in Table 3-6 is used to disable the Hot-Join feature on the I3C Bus. To disable Hot-Join of any I3C Target device on the I3C Bus, the Host shall set the Selector value to HOT_JOIN (refer Table 3-7), the least significant byte of *wIndex* to the index of the Interface and the most significant byte of *wIndex* to Zero.

3.3.2.3.5 Disable USB remote wake from regular In-Band Interrupts

The CLEAR_FEATURE request described in Table 3-6 is used to disable USB remote wake from regular IBI on the I3C Bus. To disable USB remote wake from regular IBI, the Host shall set the Selector value to REGULAR_IBI_WAKE (refer Table 3-7), the least significant byte of *wIndex* to the index of the Interface and the most significant byte of *wIndex* to Zero.

3.3.2.3.6 Disable USB remote wake from Hot-Join

The CLEAR_FEATURE request described in Table 3-6 is used to disable USB remote wake from regular Hot-Join on the I3C Bus. To disable USB remote wake from Hot-Join, the Host shall set the Selector value to HOT_JOIN_WAKE (refer Table 3-7), the least significant byte of *wIndex* to the index of the Interface and the most significant byte of *wIndex* to Zero.

3.3.2.3.7 Disable USB remote wake from I3C Controller role request

The CLEAR_FEATURE request described in Table 3-6 is used to disable USB remote wake from an I3C Controller role request on the I3C Bus. To disable USB remote wake from an I3C Controller role request, the Host shall set the Selector value to I3C_CONTROLLER_ROLE_REQUEST_WAKE (refer Table 3-7), the least significant byte of *wIndex* to the index of the Interface and the most significant byte of *wIndex* to Zero.

3.3.2.3.8 Exit HDR Mode for recovery

The CLEAR_FEATURE request described in Table 3-6 is used to force all I3C Target devices to exit HDR Mode. The Host shall set the Selector value to HDR_MODE_EXIT_RECOVERY (refer Table 3-7), the least significant byte of *wIndex* to the index of the Interface and set the most significant byte of *wIndex* with 7 bits (bits 0 to 6) of Target Address set to 7Eh and bit 7 set to zero. This request is applicable when the I3C Device is the Active I3C Controller.

Note: Exiting HDR Mode is an important step to recovering I3C Target devices that might have detected certain errors on the I3C Bus (refer [MIPII3C] Section 5.1.10.1 and [MIPII3C] Section 5.2.1.1). The HDR Exit Pattern causes I3C Target devices to recover from such an error condition. The Host should not use this request while there are outstanding transfer commands sent via Bulk requests; instead, the Host should wait until it receives responses to such transfer commands and the I3C Bus is quiesced. The Host should not use this request to send the HDR Exit Pattern after executing any HDR transfer commands sent in a Bulk request, as the I3C Device shall handle this automatically.

3.3.2.4 Get Address Change Result

The Host uses GET_ADDRESS_CHANGE_RESULT request described in Table 3-8 after the completion of the CHANGE_DYNAMIC_ADDRESS (refer Section 3.3.2.2) request to get the result of Address Change for the affected Target devices. This request is applicable when the I3C Device is the Active I3C Controller.

Table 3-8: GET_ADDRESS_CHANGE RESULT Request Fields

bmRequestType	bRequest	wValue	wIndex		wLength	Data
1.01.00001B	GET_ADDRESS_CHANGE_RESULT	Zero	Reserved	Interface	Varies	Address Change Result

Default state: Request is invalid, and Device shall respond with a Request Error.

Address state: Request is invalid, and Device shall respond with a Request Error.

Configured state: Request is valid.

It is a Request Error if *wValue, wIndex* or *wLength* are other than as specified above.
Refer Section 3.4.3 for data sent by the I3C Function in the Data stage of this request.

3.3.2.5 Get Buffer Available

The Host uses the GET_BUFFER_AVAILABLE request described in Table 3-9 to get the size of buffer available before sending a Bulk request transfer. The I3C Function shall account for buffer space that is expected to be used by all commands in any currently enqueued Bulk requests, and the corresponding Bulk responses that would be generated as a result of executing the commands in such Bulk requests (refer Table 3-26).

Table 3-9: GET_BUFFER_AVAILABLE Request Fields

bmRequestType	bRequest	wValue	wIndex		wLength	Data
1.01.00001B	GET_BUFFER_AVAILABLE	Zero	Reserved	Interface	Four	Buffer Available

Default state: Request is invalid, and Device shall respond with a Request Error.

Address state: Request is invalid, and Device shall respond with a Request Error.

Configured state: Request is valid.

It is a Request Error if *wValue, wIndex* or *wLength* are other than as specified above.
Refer Section 3.4.4 for data sent by the I3C Function in the Data stage of this request.

3.3.2.6 Get I3C Capability

The GET_I3C_CAPABILITY request described in Table 3-10 is used to read the capability of the I3C Device in the USB Device and Target devices on the I3C Bus. Refer Section 3.4.5 for I3C Capability data structure.

Table 3-10: GET_I3C_CAPABILITY Request Fields

bmRequestType	bRequest	wValue	wIndex		wLength	Data
1.01.00001B	GET_I3C_CAPABILITY	Zero	Reserved	Interface	Varies	I3C Capability

Default state: Request is invalid, and Device shall respond with a Request Error.

Address state: Request is invalid, and Device shall respond with a Request Error.

Configured state: Request is valid.

It is a Request Error if *wValue, wIndex* or *wLength* are other than as specified above.

The Host can determine the size and I3C Capability data by reading the *I3C_CAPABILITY_HEADER* field at the beginning of the I3C Capability data structure (refer Table 3-27).

If field *wLength* is set to a value larger than the size of the I3C Capability data, the I3C Function shall send the complete I3C Capability data (refer Table 3-27) and no additional data.

3.3.2.7 Get Target Device Table

The GET_TARGET_DEVICE_TABLE request described in Table 3-11 is used to get the Target Device Table from the I3C Function. Refer Section 3.4.6 for Target Device Table data structure. This request shall only be sent after the I3C Function initialization operation is complete. This request is applicable when the I3C Device is the Active I3C Controller.

Table 3-11: GET_TARGET_DEVICE_TABLE Request Fields

bmRequestType	bRequest	wValue	wIndex		wLength	Data
1.01.00001B	GET_TARGET_DEVICE_TABLE	Zero	Reserved	Interface	Varies	Target Device Table

Default state: Request is invalid, and Device shall respond with a Request Error.

Address state: Request is invalid, and Device shall respond with a Request Error.

Configured state: Request is valid.

It is a Request Error if *wValue, wIndex* or *wLength* are other than as specified above.

The Host can determine the size of the Target Device Table by reading the *Target Device Table Size* field at the beginning of the Target Device Table data structure (refer Table 3-28).

If field *wLength* is set to a value larger than the size of Target Device Table, the I3C Function shall send the complete Target Device Table (refer Table 3-28) and no additional data.

3.3.2.8 Initialize I3C Bus

The INITIALIZE_I3C_BUS request described in Table 3-12 is used to initialize the I3C Bus, after the Host determines the role of the I3C Device as the I3C Controller (Primary I3C Controller).

Table 3-12: INITIALIZE_I3C_BUS Request Fields

bmRequestType	bRequest	wValue	wIndex		wLength	Data
0.01.00001B	INITIALIZE_I3C_BUS	Address Assignment Mode	Reserved	Interface	Varies	Target Device Table

Default state: Request is invalid, and Device shall respond with a Request Error.

Address state: Request is invalid, and Device shall respond with a Request Error.

Configured state: Request is valid.

It is a Request Error if *wValue, wIndex* or *wLength* are other than as specified above.

Table 3-13: Address Assignment Mode Values

Address Assignment Modes	wValue	Description
I3C_CONTROLLER_DECIDED_ADDRESS_ASSIGNMENT	0h	I3C Controller determines how to assign dynamic addresses to the I3C Target devices on the I3C Bus. I3C Controller may choose to perform SETDASA and/or SETAASA before ENTDAA, based on the information about Target devices. Refer [MIPII3C] Section 5.1.4.
ENTER_DYNAMIC_ADDRESS_ASSIGNMENT	1h	Enter dynamic address assignment mode defined by MIPI. Refer [MIPII3C] Section 5.1.4.

SET_STATIC_ADDRESS_AS_DYNAMIC_ADDRESS	2h	Set known static address of an I3C Target device as its dynamic address. Refer [MIPII3C] Section 5.1.4. ENTDAA is not performed.

The Host shall not send Target Device Table in the Data stage of this request, if the I3C Controller has the knowledge of Target devices on the I3C Bus. Refer Section 4.1.1, Figure 4-2.

The Host shall send Target Device Table in the Data stage of this request, if the I3C Controller has no knowledge of Target devices on the I3C Bus. Refer Section 4.1.1, Figure 4-3.

3.3.2.9 Set I3C Feature

The SET_FEATURE request described in Table 3-14 is used to enable features defined by the value of Selector (refer Table 3-15). This request is applicable when the I3C Device is the Active I3C Controller.

Table 3-14: SET_FEATURE Request Fields

bmRequestType	bRequest	wValue	wIndex		wLength	Data
0.01.00001B	SET_FEATURE	Selector	Zero/Target Address	Interface	Zero	None

Default state: Request is invalid, and Device shall respond with a Request Error.

Address state: Request is invalid, and Device shall respond with a Request Error.

Configured state: Request is valid.

It is a Request Error if *wValue, wIndex* or *wLength* are other than as specified above.

Table 3-15 lists the I3C feature selector definitions.

Table 3-15: I3C Set Feature Selector Values

Selector	wValue	Description
Reserved	0h	Reserved.
Reserved	1h	Reserved.
I3C_CONTROLLER_ROLE_HANDOFF	2h	Enables handoff of I3C Controller role to Secondary I3C Controller
REGULAR_IBI	3h	Enables regular In-Band Interrupts on the I3C Bus
HOT_JOIN	4h	Enables Hot-Join feature on the Bus. *Note: MSB of wIndex is set to Zero*
Reserved	5h	Reserved.
REGULAR_IBI_WAKE	6h	Enables remote wake from a regular IBI from a Target device on the I3C Bus.
HOT_JOIN_WAKE	7h	Enables remote wake from a Hot-Join request on the I3C Bus.
I3C_CONTROLLER_ROLE_REQUEST_WAKE	8h	Enables remote wake from an I3C Controller role request on the I3C Bus.

Reserved	9h	Reserved.

3.3.2.9.1 Enable I3C Controller role handoff

The SET_FEATURE request described in Table 3-14 is used to enable handoff of the I3C Controller role to another I3C Controller-capable Device (i.e., an I3C Secondary Controller) on the I3C Bus. To initiate the procedure for I3C Controller role handoff, the Host shall set the Selector value to I3C_CONTROLLER_ROLE_HANDOFF (refer Table 3-15), the least significant byte of *wIndex* to the index of the Interface and the most significant byte of *wIndex* to Zero.

3.3.2.9.2 Enable all regular In-Band Interrupts on I3C Bus

The SET_FEATURE request described in Table 3-14 is used to enable all regular In-Band Interrupts from I3C Target devices on the I3C Bus. To enable regular In-Band Interrupts, the Host shall set the Selector value to REGULAR_IBI (refer Table 3-15), the least significant byte of *wIndex* to the index of the Interface and the most significant byte of *wIndex* to Zero.

3.3.2.9.3 Enable Hot-Join

The SET_FEATURE request described in Table 3-14 is used to enable the Hot-Join feature on the I3C Bus. To enable Hot-Join Requests from any I3C Target device on the I3C Bus, the Host shall set the Selector value to HOT_JOIN (refer Table 3-15), the least significant byte of *wIndex* to the index of the Interface and the most significant byte of *wIndex* to Zero.

3.3.2.9.4 Enable USB remote wake from regular In-Band Interrupts

The SET_FEATURE request described in Table 3-14 is used to enable USB remote wake from regular IBI on the I3C Bus. To enable USB remote wake from regular IBI, the Host shall set the Selector value to REGULAR_IBI_WAKE (refer Table 3-15), the least significant byte of *wIndex* to the index of the Interface and the most significant byte of *wIndex* to Zero. This request is applicable when the I3C Device is the Active I3C Controller.

3.3.2.9.5 Enable USB remote wake from Hot-Join

The SET_FEATURE request described in Table 3-14 is used to enable USB remote wake from regular Hot-Join on the I3C Bus. To enable USB remote wake from Hot-Join, the Host shall set the Selector value to HOT_JOIN_WAKE (refer Table 3-15), the least significant byte of *wIndex* to the index of the Interface and the most significant byte of *wIndex* to Zero. This request is applicable when the I3C Device is the Active I3C Controller.

3.3.2.9.6 Enable USB remote wake from I3C Controller role request

The SET_FEATURE request described in Table 3-14 is used to enable USB remote wake from an I3C Controller role request on the I3C Bus. To enable USB remote wake from an I3C Controller role request, the Host shall set the Selector value to I3C_CONTROLLER_ROLE_REQUEST_WAKE (refer Table 3-15), the least significant byte of *wIndex* to the index of the Interface and the most significant byte of *wIndex* to Zero. This request is applicable when the I3C Device is the Active I3C Controller.

3.3.2.10 Set Target Device Config

The SET_TARGET_DEVICE_CONFIG request described in Table 3-16 is used to set the configurable parameters of one or more Target devices. For configurable parameters, refer to the *Config Change* field in Table 3-29. This request is applicable when the I3C Device is the Active I3C Controller.

Table 3-16: SET_TARGET_DEVICE-CONFIG Request Fields

bmRequestType	bRequest	wValue	wIndex	wLength	Data

0.01.00001B	SET_TARGET_DEVICE_CONFIG	Zero	Reserved	Interface	Varies	Target Device Configuration

Default state: Request is invalid, and Device shall respond with a Request Error.

Address state: Request is invalid, and Device shall respond with a Request Error.

Configured state: Request is valid.

It is a Request Error if *wValue, wIndex* or *wLength* are other than as specified above.
Refer Section 3.4.7 for data sent by the Host in the Data stage of this request.

After receiving this request, the I3C Function shall update the configurable parameters of the Target device in the stored Target Device Table (refer Table 3-28).

3.4 Data Structures

The following data structures describe the layout and interpretation of the binary data messages used for various request flows, involving the Control, Interrupt and Bulk endpoints of the USB Device.

3.4.1 Notification Data Structures

This section defines the notifications that an I3C Function may send to the Host. An I3C Function shall not send a notification unless it is in the Configured state.

Table 3-17 lists the notification types supported in this version of specification.

Table 3-17: Notification Types

Notification	Value
Reserved	00h
I3C_BUS_INITIALIZATION_STATUS	01h
ADDRESS_CHANGE_STATUS	02h
I3C_BUS_ERROR	03h
I3C_IBI	04h
ACTIVE_I3C_CONTROLLER_EVENT	05h
STALL_ON_NACK	06h

3.4.1.1 I3C Bus Initialization Notification Format

This notification is sent by the I3C Function to indicate if I3C Bus is successfully initialized, as a result of receiving the INITIALIZE_I3C_BUS request from the Host (refer Section 3.3.2.8).

Table 3-18: I3C Controller and I3C Bus Initialization Notification

Offset	Field	Size	Value	Description
0	***bNotificationType***	1	Constant	Set to I3C_BUS_INITIALIZATION_STATUS (refer Table 3-17).
1	***wNotificationCode***	2	Number	Value / Description 0h Successful I3C Bus initialization 1h Failure to enable I3C Bus

				2h Failure with device discovery and dynamic address assignment 3h Failure with Target Device Table generation and/or update Other Values Reserved.
3	***bReserved***	1	Zero	Reserved. Shall be set to 0.

3.4.1.2 Address Change Notification Format

This notification is sent by the I3C Function to indicate that the Dynamic Addresses of one or more I3C Target devices were successfully changed, as a result of receiving and processing the CHANGE_DYNAMIC_ADDRESS request from the Host (refer Section 3.3.2.2). This notification is also sent by the I3C Function to indicate if a Hot-Joined I3C Target device was successfully assigned an address.

Table 3-19: Address Change Notification

Offset	Field	Size	Value	Description
0	***bNotificationType***	1	Constant	Set to ADDRESS_CHANGE_STATUS (refer Table 3-17)
1	***wNotificationCode***	2	Number	<u>Value</u> <u>Description</u> 0h All addresses changed successfully. 1h Failure with address change. 2h Successfully assigned address to Hot-Joined I3C Target device. 3h Failed to assign address to Hot-Joined I3C Target device. Other Values Reserved.
3	***bReserved***	1	Zero	Reserved. Shall be set to 0.

3.4.1.3 I3C Bus Error Notification Format

This notification is sent by the I3C Function to indicate any errors on the I3C Bus.

Table 3-20: I3C Bus Error Notification

Offset	Field	Size	Value	Description
0	***bNotificationType***	1	Constant	Set to I3C_BUS_ERROR (refer Table 3-17)
1	***wNotificationCode***	2	Number	<u>Value</u> <u>Description</u> 0h Reserved. 1h CCC format incorrect 2h Data transmitted incorrect. (Optional in MIPI) 3h Broadcast Address NACK'd 4h New I3C Controller fails to drive the I3C Bus Other Values Reserved.

Offset	Field	Size	Value	Description
3	***bReserved***	1	Zero	Reserved. Shall be set to 0.

3.4.1.4 I3C IBI Notification Format

This notification is sent by the I3C Function to report any IBIs (In-Band Interrupts) on the I3C Bus.

Table 3-21: I3C IBI Notification

Offset	Field	Size	Value	Description
0	***bNotificationType***	1	Constant	Set to I3C_IBI (refer Table 3-17)
1	***wNotificationCode***	2	Number	Value — Description 0h — Reserved. 1h — Regular IBI without payload ACK'd by the I3C Controller. 2h — Regular IBI with payload ACK'd by the I3C Controller. 3h — IBI with Auto-Command/Private Read initiated by the I3C Controller. 4h — Regular IBI NACK'd by the I3C Controller. 5h — Hot-Join IBI ACK'd by the I3C Controller. 6h — Hot-Join IBI NACK'd by the I3C Controller. 7h — Secondary Controller role request ACK'd by the I3C Controller. 8h — Secondary Controller role request NACK'd by the I3C Controller. Other Values — Reserved.
3	***bReserved***	1	Zero	Reserved. Shall be set to 0.

3.4.1.5 Active I3C Controller Notification Format

This notification is sent by the I3C Function to indicate events from the Active I3C Controller on I3C Bus to either this I3C Target device, or this I3C Target device capable of Secondary I3C Controller role. This notification is applicable when I3C Device is not in the I3C Controller role (i.e., Primary I3C Controller).

Table 3-22: Active I3C Controller Notification

Offset	Field	Size	Value	Description
0	***bNotificationType***	1	Constant	Set to ACTIVE_I3C_CONTROLLER_EVENT (refer Table 3-17)
1	***wNotificationCode***	2	Number	Value — Description 0h — Reserved. 1h — Reserved. 2h — Received a CCC. 3h — Received a Read request. 4h — Received a Write request.

Offset	Field	Size	Value	Description
				Other Values Reserved.
3	***bReserved***	1	Zero	Reserved. Shall be set to 0.

3.4.1.6 Stall on NACK Notification Format

This notification is sent by the I3C Function to indicate that I3C Controller stalled the execution of commands in Bulk request transfer upon receiving a NACK from an I3C Target device.

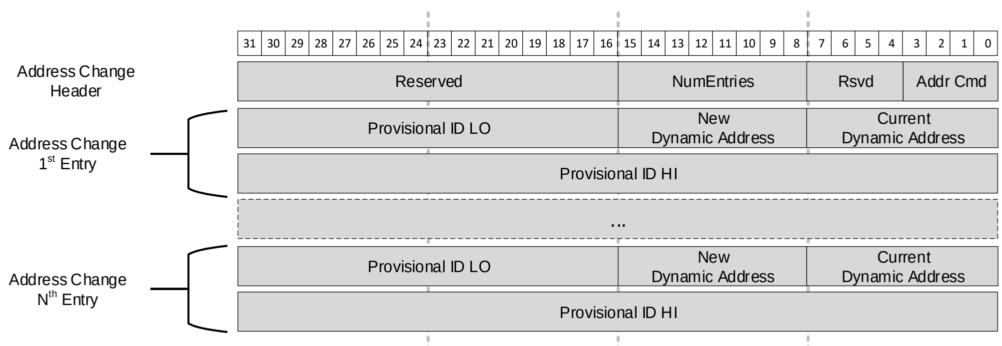
Table 3-23: Active I3C Controller Notification

Offset	Field	Size	Value	Description
0	***bNotificationType***	1	Constant	Set to STALL_ON_NACK (refer Table 3-17)
1	***wNotificationCode***	2	Number	Set to the *Request ID* value of the *COMMAND_BLOCK_HEADER* (refer Table 3-30) in the Bulk request transfer that I3C Controller stalled on.
3	***bReserved***	1	Zero	Reserved. Shall be set to 0.

3.4.2 Address Change

This structure is sent to the I3C Function in the CHANGE_DYNAMIC_ADDRESS request from the Host (refer Section 3.3.2.2). Figure 3-1 illustrates the format of this data structure.

Figure 3-1: Address Change Data Structure



The fields of Address Change data structure are defined in Table 3-24.

Table 3-24: Address Change Data Structure Fields

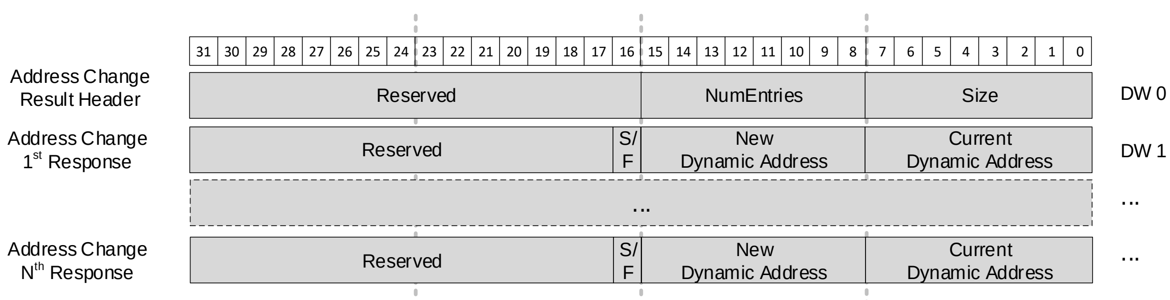
DW	Field	Bits	Description
0	ADDRESS_CHANGE_HEADER	3:0	**Address Change Command Type (Addr Cmd)** This field contains the Address Command Type. The field value definitions are as listed below: 1h – CHGDA (Change Dynamic Address) Other Values - Reserved

DW	Field	Bits	Description
		7:4	**Reserved** This field shall be set to all zeros.
		15:8	**Number of Entries (NumEntries)** This field is set to the number of Target devices for which the Host intends to change the Dynamic Address. *Note: Entry per Target device comprises of 64 bits with 8 bits of Current Dynamic address, 8 bits of New Dynamic Address, followed by 16 bits of Provisional ID LO and 32 bits of Provisional ID HI.*
		31:16	**Reserved** This field shall be set to all zeros.
1	Address Change 1st Entry	7:0	**Current Dynamic Address** This field contains 7 bits of current dynamic address (bits 6:0) of the 1st I3C Target device and 1 bit (bit 7) set to 0.
		15:8	**New Dynamic Address** This field contains 7 bits of new dynamic address (bits 14:8) of the 1st I3C Target device and 1 bit (bit 15) set to 0.
		31:16	**Provisional ID LO** This field contains the 15:0 bits of PID, if the I3C Target device has a valid PID.
2		31:0	**Provisional ID HI** This field contains the 47:16 bits of PID, if the I3C Target device has a valid PID.
…	…	…	Additional Address Change entries.

3.4.3 Address Change Result

The Address Change Result data structure is returned by the I3C Function, in response to receiving the GET_ADDRESS_CHANGE_RESULT request from the Host (refer Section 3.3.2.4). Figure 3-2 illustrates the format of this data structure.

Figure 3-2: Address Change Result Data Structure



The fields of Address Change Result data structure are defined in Table 3-25.

Table 3-25: Address Change Result Data Structure Fields

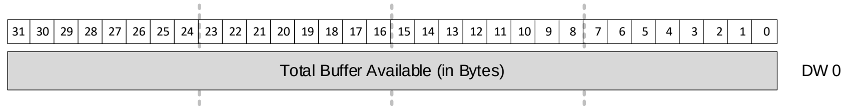
DW	Field	Bits	Description
0	ADDRESS_CHANGE_RESULT_HEADER	7:0	**Size** This field contains the total size (in bytes) of the Get Dynamic Address Response data structure including the ADDRESS_CHANGE_RESULT_HEADER size.
		15:8	**Number of Entries (NumEntries)** This field is set to the number of Target devices for which the I3C Function attempted to change the Dynamic Address. *Note: Entry per Target device comprises of 32 bits with 8 bits of Current Dynamic address, followed by 8 bits of New Dynamic Address and 16 reserved bits.*
		31:16	**Reserved** This field shall be set to all zeros.
1	Address Change 1st Response	7:0	**Current Dynamic Address** This field contains 7 bits of current dynamic address (bits 6:0) of the 1st I3C Target device and 1 bit (bit 7) set to 0.
		15:8	**New Dynamic Address** This field contains 7 bits of new dynamic address (bits 14:8) of the 1st I3C Target device and 1 bit (bit 15) set to 0.
		16	**Success/Failure (S/F)** This field indicates if this specific Dynamic Address was successfully changed. This bit shall be set to 0b to indicate Success and set to 1b to indicate Failure.
		31:17	**Reserved** This field shall be set to all zeros.
…	…	…	Additional Address Change Responses

3.4.4 Buffer Available

The Buffer Available data structure is returned by I3C Function, in response to receiving the GET_BUFFER_AVAILABLE request from the Host (refer Section 3.3.2.5). Figure 3-3 illustrates the format of this data structure.

Note: The total buffer available shall account for buffer space needed for the current Bulk request and its corresponding Bulk response, as well as all previously enqueued Bulk requests and Bulk responses: this includes any Bulk requests that currently contain Write data for I3C Targets (in requests) and those that will cause I3C Targets to provide Read data (in Bulk responses). The I3C Function shall return an available buffer size that is safe for the Host to use, assuming that all currently enqueued Bulk request transfers are executed successfully. The I3C Function should also account for buffer space that might be temporarily used by I3C In-Band Interrupts (IBIs) with data payloads, as well as any Pending Read data that could accompany such IBIs.

Figure 3-3: Buffer Available Data Structure



The fields of the Buffer Available data structure are defined in Table 3-26.

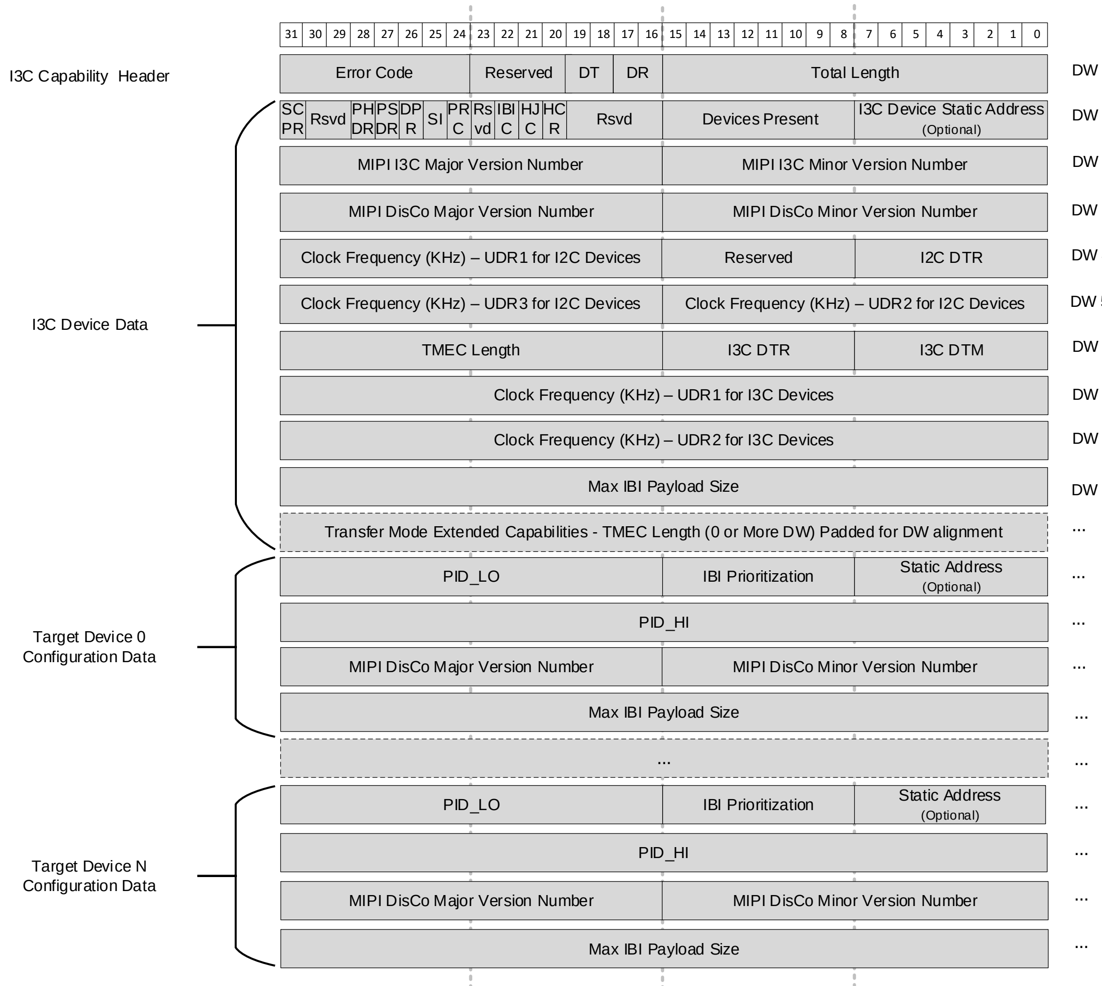
Table 3-26: Buffer Available Data Structure Fields

DW	Bits	Field	Description
0	31:0	TOTAL_BUFFER_AVAILABLE	**Total Buffer Available** This field contains the size of total buffer available (in bytes) for both reads and writes with I3C Function.

3.4.5 I3C Capability

The I3C Capability data structure is returned by the I3C Function, in response to receiving the GET_I3C_CAPABILITY request from the Host (refer Section 3.3.2.6). Figure 3-4 illustrates the format of this data structure.

Figure 3-4: I3C Capability Data Structure



The fields of the I3C Capability data structure are defined in Table 3-27.

Table 3-27: I3C Capability Data Structure Fields

DW	Bits	Field	Description
0	15:0	I3C_CAPABILITY_HEADER	**Total Length** This field contains the total size (in bytes) of the I3C Capability data including the size of the I3C_CAPABILITY_HEADER.
	17:16		**Device Role (DR)** This field indicates the role of the internal I3C Device. Value — Description 1h — I3C Controller role. I3C Device is the Primary I3C Controller role. 2h — I3C Target device 3h — I3C Target device capable of Secondary I3C Controller role. Other Values — Reserved

DW	Bits	Field	Description
	19:18		**Data Type (DT)** This field indicates the type of data in the I3C Capability data structure. Value — Description 0h — Reserved 1h — Static data. I3C Device is aware of the Target devices on the I3C Bus. 2h — No Static data. I3C Device is not aware of the Target devices on the I3C Bus. 3h — Dynamic data. I3C Device is aware of the Target devices on the I3C Bus through previously received information from the Host.
	23:20		**Reserved** This field shall be set to all zeros.
	31:24		**Error Code (ER)** This field indicates if the I3C Device contains I3C Capability data structure. Value — Description 00h — I3C Device contains the I3C Capability data structure. FFh — I3C Device does not contain the I3C Capability data structure. Other Values — Reserved
1	7:0	I3C_DEVICE_CAPABILITY	**I3C Device Static Address** This field is optional. This field contains 7 bits (bits 6:0) of the I3C Device's Static Address and 1 Reserved bit (bit 7) set to zero.
	15:8		**Devices Present** This field indicates the type of I2C Target devices present on the I3C Bus which are controlled by this I3C Controller. The field value definitions are as listed below: Value — Description 0h — No I2C Target devices on the I3C Bus 1h — I2C Target devices with 50ns spike filter present on the I3C Bus. 2h — I2C Target devices without 50ns spike filter present on the I3C Bus 3h — Mix of I2C Target devices with and without 50ns spike filter present on the I3C Bus Other Values — Reserved
	19:16		**Reserved** This field shall be set to all zeros.

DW	Bits	Field	Description
	20		**Handoff Controller Role (HCR)** This field indicates if I3C Device is capable of handing off the I3C Controller role to Secondary I3C Controller on the I3C Bus. This field is set to 0b if I3C Device is not capable of handing off the I3C Controller role. This field is set to 1b if I3C is capable of handing off the I3C Controller role.
	21		**Hot-Join Capability (HJC)** This field indicates if I3C Device is capable of handling Hot-Joins. This field is set to 0b if I3C Device is not capable of handling Hot-Join. This field is set to 1b if I3C Device is capable of handling Hot-Join.
	22		**In-Band Interrupt Capability (IBIC)** This field indicates if I3C Device is capable of handling IBIs. This field is set to 0b if I3C Device is not capable of handling IBI. This field is set to 1b if I3C Device is capable of handling IBI.
	23		**Reserved** This field shall be set to zero.
	24		**Pending Read Capability (PRC)** This field indicates if this I3C Device supports pending read for an IBI. This field is to 0b if I3C Device is not capable of IBI pending read. This field is set to 1b if I3C Device is capable of IBI pending read. Refer Section 5.1.6.2 and 5.1.9.3.19 of [MIPII3C] for details on IBI and Pending Read.
	25		**Self-Initiated (SI)** This field is valid if *Pending Read Capability* field is set to 1b. This field indicates if this I3C Device self-initiates pending read or if pending read is initiated by the Host. This field is to 0b if I3C Device self-initiates the pending read. This field is set to 1b if Host initiates pending read.

DW	Bits	Field	Description
	26		**Delayed Pending Read (DPR)** This field is valid if *Pending Read Capability* field is set to 1b. This field indicates if this I3C Device performs immediate pending read by aborting any outstanding I3C transfers or if the I3C Device performs delayed pending read after completing any outstanding I3C transfers. This field is set to 0b if I3C Device performs immediate pending read. This field is set to 1b if I3C Device performs delayed pending read.
	27		**Pending Read SDR (PSDR)** This field is valid if *Pending Read Capability* field is set to 1b. This field indicates if this I3C Device supports pending read in SDR mode. This field is set to 0b if I3C Device does not support pending read in SDR mode. This field is set to 1b if I3C Device supports pending read in SDR mode.
	28		**Pending Read HDR (PHDR)** This field is valid if *Pending Read Capability* field is set to 1b. This field indicates if this I3C Device supports pending read in HDR mode. This field is set to 0b if I3C Device does not support pending read in HDR mode. This field is set to 1b if I3C Device supports pending read in HDR mode.
	30:29		**Reserved** This field shall be set to all zeros.
	31		**Single Command Pending Read (SCPR)** This field is valid if *Pending Read Capability* field is set to 1b. This field indicates if this I3C Device supports pending read which consists of either a single read command or multiple commands. This field is set to 0b if I3C Device supports single pending read command. This field is set to 1b if I3C Device supports multiple commands to accomplish pending read.
2	15:0		**MIPI I3C Minor Version Number** This field contains the Minor Version Number of [MIPII3C] the I3C Controller complies with.
	31:16		**MIPI I3C Major Version Number** This field contains the Major Version Number [MIPII3C] the I3C Controller complies with.

DW	Bits	Field	Description
3	15:0		**MIPI I3C DisCo Minor Version Number** This field contains the Minor Version Number of [MIPIDISCOI3C] the I3C Controller complies with.
	31:16		**MIPI I3C DisCo Major Version Number** This field contains the Major Version Number [MIPIDISCOI3C] the I3C Controller complies with.
4	7:0		**I2C Data Transfer Rates (I2C DTR)** This bitmap field indicates all the I2C data transfer rates Supported by the I3C Device. Bits — Description 0 — Set to 1b if data rate of 100 KHz is supported, else set to 0b. 1 — Set to 1b if data rate of 400 KHz is supported, else set to 0b. 2 — Set to 1b if data rate of 1 MHz is supported, else set to 0b. 3 — Set to 1b if User defined I2C data rate UDR1 is supported, else set to 0b. 4 — Set to 1b if User defined I2C data rate UDR2 is supported, else set to 0b. 5 — Set to 1b if User defined I2C data rate UDR3 is supported, else set to 0b. 7:6 — Reserved, set to all zeros.
	15:8		**Reserved** This field shall be set to all zeros.
	31:16		**Clock Frequency I2C UDR1** This field contains the clock frequency in KHz when using I2C UDR1 transfer rate.
5	15:0		**Clock Frequency I2C UDR2** This field contains the clock frequency in KHz when using I2C UDR2 transfer rate.
	31:16		**Clock Frequency I2C UDR3** This field contains the clock frequency in KHz when using I2C UDR3 transfer rate.

DW	Bits	Field	Description
6	7:0		**<u>I3C Data Transfer Modes (I3C DTM)</u>** This bitmap field indicates all the I3C data transfer modes supported by the I3C Device. <u>Bits</u> <u>Description</u> 0 Set to 1b if single lane I3C SDR mode is supported, else set to 0b. 1 Set to 1b if single lane I3C HDR-DDR mode is supported, else set to 0b. 2 Set to 1b if single lane I3C HDR-TS mode is supported, else set to 0b. 3 Set to 1b if single lane I3C HDR-BT mode is supported, else set to 0b. 7:4 Reserved, set to all zeros.
	15:8		**<u>I3C Data Transfer Rates (I3C DTR)</u>** This bitmap field indicates all the I3C data transfer rates supported by the I3C Device. <u>Bits</u> <u>Description</u> 8 Set to 1b if sustainable data rate of 2 MHz is supported, else set to 0b. 9 Set to 1b if sustainable data rate of 4 MHz is supported, else set to 0b. 10 Set to 1b if sustainable data rate of 6 MHz is supported, else set to 0b. 11 Set to 1b if sustainable data rate of 8 MHz is supported, else set to 0b. 12 Set to 1b if data rate of max up to 12.5 MHz is supported, else set to 0b. 13 Set to 1b if User Defined I3C data rate 1 is supported, else set to 0b. 14 Set to 1b if User Defined I3C data rate 2 is supported, else set to 0b. 15 Reserved, shall be set to 0b.
	31:16		**<u>Transfer Mode Extended Capability Length (TMEC Length)</u>** This field indicates the length of Transfer Mode Extended Capabilities in Bytes.
7	31:0		**<u>Clock Frequency I3C UDR1</u>** This field contains the clock frequency in KHz when using I3C UDR1 transfer rate.
8	31:0		**<u>Clock Frequency I3C UDR2</u>** This field contains the clock frequency in KHz when using I3C UDR2 transfer rate.

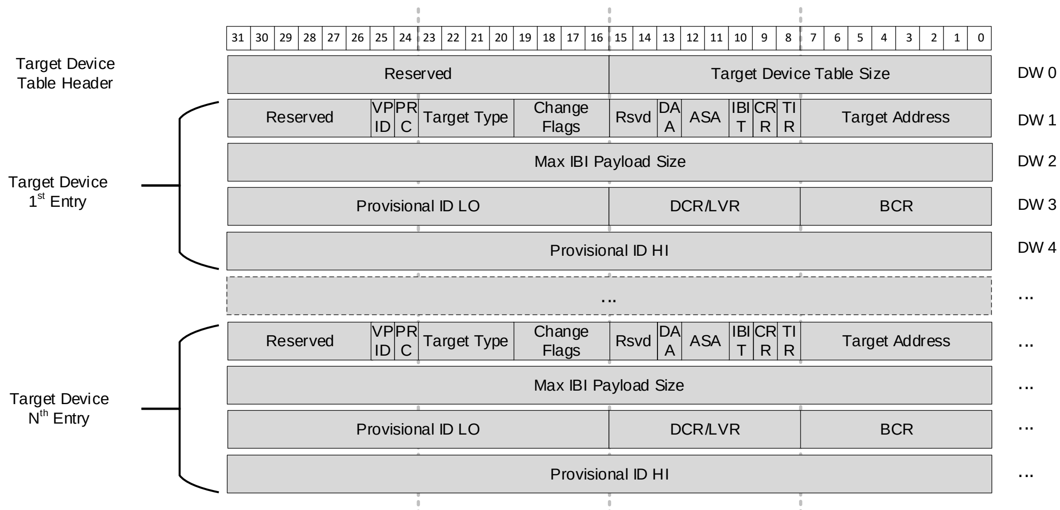
DW	Bits	Field	Description
9	31:0		**Max IBI Payload Size** For *Device Role* field set to 1h this field indicates the maximum IBI payload size this I3C Device can read for an IBI it acknowledges. For *Device Role* field set to 2h this field indicates the maximum IBI payload size this I3C Device is allowed to send as with an IBI to the I3C Controller. For *Device Role* field set to 3h this field indicates the maximum IBI payload size this I3C Device can read or send for an IBI based on its role. A value of 0h indicates unlimited IBI payload size. Refer Section 5.1.6.2 and 5.1.9.3.19 of [MIPII3C] for details on IBI and Pending Read.
...	Variable		**Transfer Mode Extended Capabilities** This field is reserved for extended capabilities associated with specific I3C Data Transfer Mode, defined by [MIPII3C]. This field shall be DWORD aligned. *Note: Details pertaining to this field are expected to be included in future revisions of this specification.*
...	7:0	Target Device 0 Capability	**Static Address** This field is optional for I3C Target devices. This field contains 7 bits (bits 6:0) of the I3C Static Address and 1 Reserved bit (bit 7) set to zero.
	15:8		**IBI Prioritization** This field indicates the priority of the IBI generated by this Target device compared to other Target devices on the I3C Bus. This field can contain value between 0x00 and 0xFF, lower value implies higher priority.
	31:16		**PID_LO** This field contains Target device's Provisional ID Low. Bits [15:0] of the I3C Target device's PID.
...	31:0		**PID_HI** This field contains Target device's Provisional ID High. Bits [47:16] of the I3C Target device's PID.
...	15:0		**MIPI I3C DisCo Minor Version Number** This field contains the Minor Version Number of [MIPIDISCOI3C] the I3C Controller complies with.
	31:16		**MIPI I3C DisCo Major Version Number** This field contains the Major Version Number [MIPIDISCOI3C] the I3C Controller complies with.

DW	Bits	Field	Description
	31:0		**Max IBI Pending Read Size** This field indicates the maximum size of data this Target Device is allowed to send as pending read for an IBI to the I3C Controller. A value of 0h indicates unlimited maximum IBI pending read size. This field is valid if I3C Target device supports Pending Read for IBI (refer Section 5.1.6.2 and 5.1.9.3.19 of [MIPII3C]).
...	...	...	Additional Target Device Configurations.

3.4.6 Target Device Table

The Target Device Table data structure is returned by the I3C Function, in response to receiving the GET_TARGET_DEVICE_TABLE request from the Host (refer Section 3.3.2.7). The Target Device Table is also sent to the I3C Function in the INITIALIZE_I3C_BUS request (refer Section 3.3.2.8). Figure 3-5 illustrates the format of this data structure.

Figure 3-5: Target Device Table Data Structure



The fields of Target Device Table data structure are defined in Table 3-28.

Table 3-28: Target Device Table Data Structure Fields

DW	Field	Bits	Description
0	TARGET_DEVICE_ TABLE_HEADER	15:0	**Target Device Table Size** This field contains the total size (in bytes) of the Target Device Table data structure including the TARGET_DEVICE_TABLE_HEADER size.
		31:16	**Reserved** This field shall be set to all zeros.

DW	Field	Bits	Description
1	Target Device 1st Entry	7:0	**<u>Target Address</u>** This field contains 7 bits of Target device Address (bits 6:0) and 1 bit (bit 7) set to 0. Target Address for I3C Target Device is changeable. Target device Address shall be static and fixed if the *Target Type* field is set to 1h (Target device is I2C device). Target device Address may be changed if the *Target Type* field is set to 0h (Target device is I3C device).
		8	**<u>Target Interrupt Request (TIR)</u>** This field is configurable. This field controls whether the Active I3C Controller will accept or reject interrupts from this Target device. If this bit is set to 0b, the Active I3C Controller shall ACCEPT interrupts from this Target device. If this bit is set to 1b, Active I3C Controller shall REJECT interrupts from this Target device.
		9	**<u>Controller Role Request (CRR)</u>** This field is configurable. This field controls whether the Active I3C Controller accepts or rejects the I3C Controller role request. If this bit is set to 0b, Active I3C Controller shall ACCEPT the I3C Controller role requests from Secondary I3C Controllers. If this bit is set to 1b, Active I3C Controller shall REJECT the I3C Controller role requests from Secondary I3C Controllers.
		10	**<u>IBI Timestamp (IBIT)</u>** This field is configurable. This field enables or disables timestamping of IBIs from the Target device. If this bit is set to 0b, Active I3C Controller shall not timestamp IBIs from this Target device. If this bit is set to 1b, Active I3C Controller shall timestamp IBIs from this Target device.
		12:11	**<u>Assignment from Static Address (ASA)</u>** This field is configurable when the Host sends the Target Device Table to the I3C Function during I3C Bus initialization. Refer Section 4.1.1, Figure 4-3. The field value definitions are as listed below: <u>Value</u> <u>Description</u> 0h I3C Target does not have a Static Address 1h I3C Target supports SETDASA directed CCC 2h I3C Target supports SETAASA broadcast CCC 3h I3C Target supports both SETDASA and SETAASA CCCs

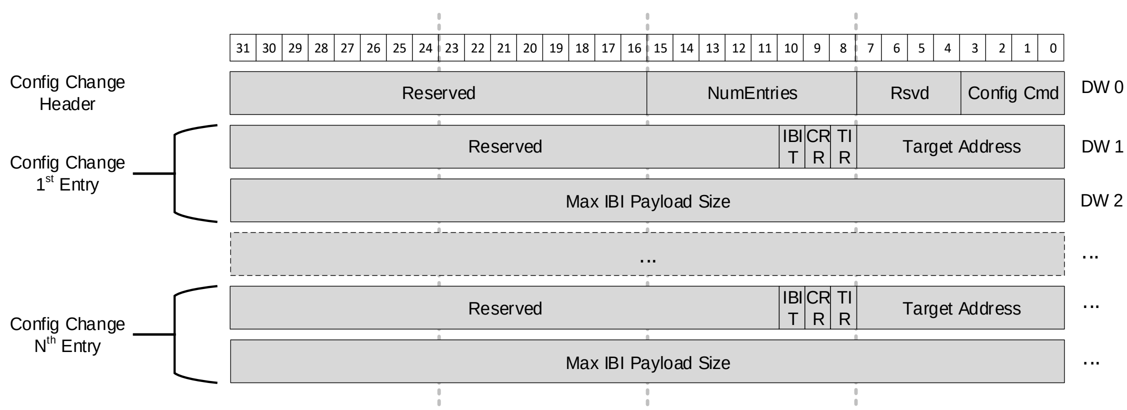
DW	Field	Bits	Description
		13	**<u>Dynamic Address Assignment with ENTDAA (DAA)</u>** This field is configurable when the Host sends the Target Device Table to the I3C Function during I3C Bus initialization. Refer Section 4.1.1, Figure 4-3. If this bit is set to 0b, the Active I3C Controller shall not use the ENTDAA CCC to configure this I3C Target device. If this bit is set to 1b, the Active I3C Controller shall use the ENTDAA CCC to configure this I3C Target device.
		15:14	**Reserved** This field shall be set to all zeros.
		19:16	**<u>Change Flags</u>** This field is set by I3C Function to indicate if there is a change in any R/W fields of this Table Entry. The field value definitions are as listed below: <u>Value</u> — <u>Description</u> 0h — No Change 1h — Change in *Target Address* field (Dynamic Addresses can be changed) 2h — Change in *Target Interrupt Request* field 3h — Change in *Controller Role Request* field. 4h — <u>Change in *IBI Timestamp* field</u> Other Values — Reserved At I3C Bus initialization, the I3C Function shall set this field to 0h. <u>*Note: This field is cleared by I3C Function only after the I3C Function sends the entire Table contents to the Host.*</u>
		23:20	**<u>Target Type</u>** If the Target device is an I3C device, this field shall be set to 0h. If the Target device is an I2C device, this field shall be set to 1h.
		24	**<u>Pending Read Capability</u>** This field indicates if the I3C Target device supports IBI pending read capability. If this bit is set to 0b, the I3C Device does not support IBI pending read. If this bit is set to 1b, the I3C Device supports IBI pending read.
		25	**<u>Valid PID (VPID)</u>** This field indicates if the I3C Target device has a valid 48-bit PID. If this bit is set to 0b, *Provisional ID Low* and *Provisional ID High* fields shall not be populated. If this bit is set to 1b, *Provisional ID Low* and *Provisional ID High* fields shall be populated.
		31:26	**Reserved** This field shall be set to all zeros.

DW	Field	Bits	Description
2		31:0	**Max IBI Payload Size** This field is configurable to a max value of up to 4GB. This field indicates the maximum IBI payload size that this I3C Device is allowed to send for an IBI to the I3C Controller. A value of 0h indicates unlimited maximum IBI payload size. Refer Section 5.1.6.2 and 5.1.9.3.19 of [MIPII3C].
3		7:0	**Bus Characteristic Register (BCR)** This field contains the Target device's I3C Bus Characteristics Register. This field is applicable for I3C Target device.
		15:8	**Device Characteristic Register (DCR)** This field contains the I3C Target device's I3C Device Characteristics Register. If the Target device is an I2C Target device, this field contains the I2C Target device's Legacy Virtual Register (LVR).
		31:16	**Provisional ID Low (Provisional ID_LO)** This field is populated if Valid PID bit is 0b. This field contains bits 15:0 of the I3C Target device's Provisional ID Low.
4		31:0	**Provisional ID High (Provisional ID_HI)** This field is populated if Valid PID bit is 0b. This field contains bits 47:16 of the I3C Target device's Provisional ID High.
…	…	…	Additional Target device entries (4DWs each).

3.4.7 Target Device Configuration

This structure is sent to the I3C Function in the SET_TARGET_DEVICE_CONFIG request (refer Section 3.3.2.10). Figure 3-6 illustrates the format of this data structure.

Figure 3-6: Target Device Configuration Data Structure



The fields of Target Device Configuration data structure are defined in Table 3-29.

Table 3-29: Target Device Configuration Change Data Structure Fields

DW	Field	Bits	Description
0	CONFIG_CHANGE_HEADER	3:0	**Config Change Command Type** This field contains the Config Change Command Type. The field value definitions are as listed below: 1h – CHGCONFIG (Change Config) 2h – CLEARCONFIG Other Values - Reserved
		7:4	**Reserved** This field shall be set to all zeros.
		15:8	**Number of Entries (NumEntries)** This field is set to the number of Target devices for which the Host intends to change the configurable parameters (refer *Target Device* field in Table 3-28).
		31:16	**Reserved** This field shall be set to all zeros.
1	Config Change 1st Entry	7:0	**Target Address** This field contains 7 bits of Target device Address (bits 6:0) and 1 bit (bit 7) set to 0.
		8	**Target Interrupt Request (TIR)** Refer Table 3-28.
		9	**Controller Role Request (CRR)** Refer Table 3-28.
		10	**IBI Timestamp (IBIT)** Refer Table 3-28.
		31:11	**Reserved** This field shall be set to all zeros.
2		31:0	**Max IBI Payload Size** This field is configurable to a max value of up to 4GB. This field indicates the maximum IBI payload size that this I3C Device is allowed to send for an IBI to the I3C Controller. A value of 0h indicates unlimited maximum IBI payload size. Refer Section 5.1.6.2 and 5.1.9.3.19 of [MIPII3C].
...	...	...	Additional Config Change entries

3.4.8 Bulk Transfers

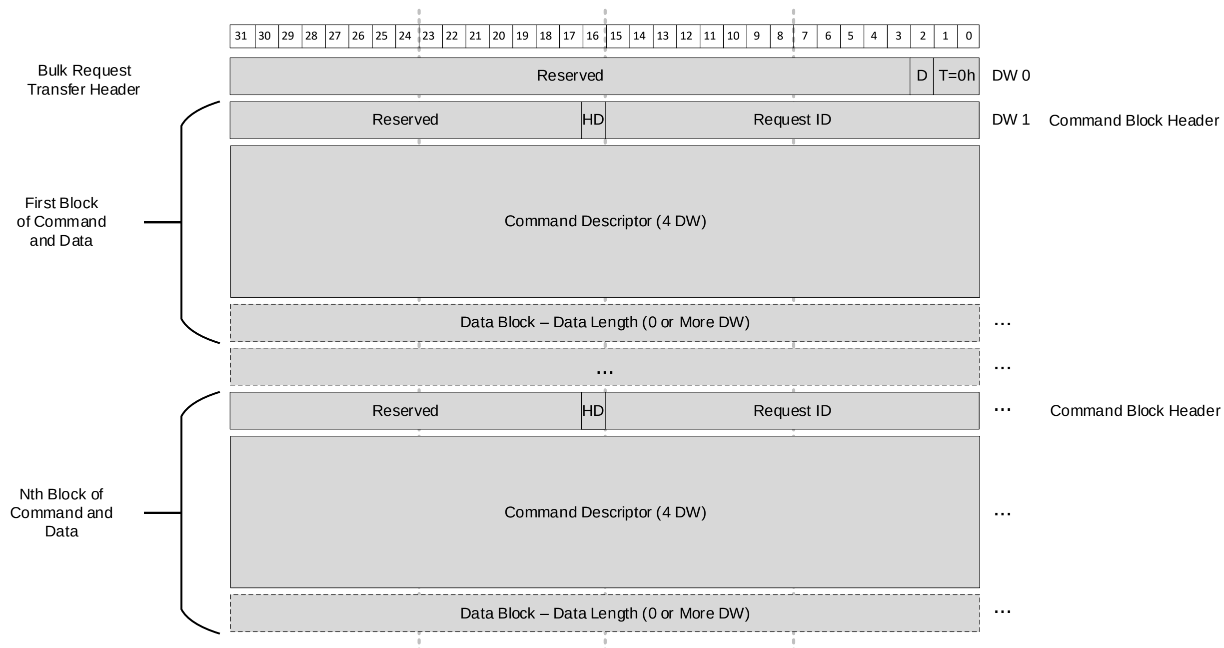
This section describes the Bulk request and response transfer data structures.

3.4.8.1 Bulk Request

This structure is sent as a Bulk request to the I3C Function. This data structure can comprise of one or more commands along with associated data for the command.

Figure 3-7 illustrates the format of this data structure. The Data Block in the structure shall be 32-bit aligned, and the Host shall pad the high-order bits of the Data Block with 0's if the Data Block is not 32-bit aligned.

Figure 3-7: Bulk Request Data Structure



The fields of Bulk Request data structure are defined in Table 3-30.

Table 3-30: Bulk Request Data Structure Fields

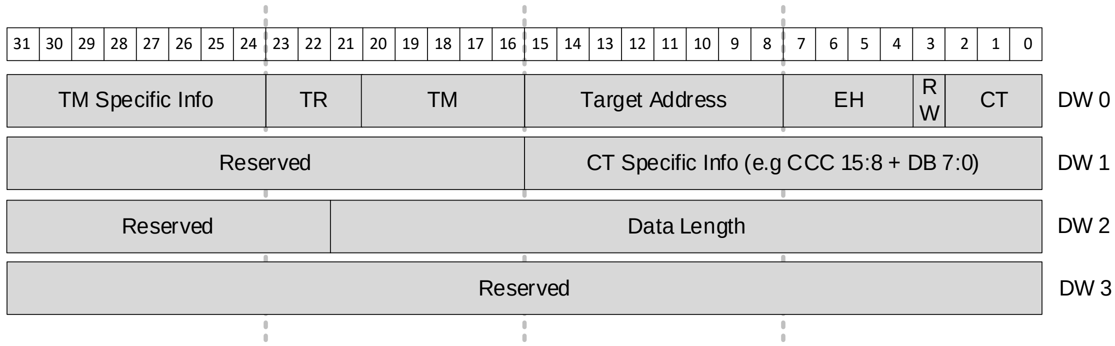
DW	Field	Bits	Description
0	BULK_REQUEST_TRANSFER_HEADER	1:0	**Tag (T)** This field indicates if the Bulk request transfer is a regular bulk request or a vendor specific request. The field value definitions are as listed below: 0h – Regular Bulk request transfer 1h – Reserved 2h – Vendor specific Bulk request transfer Other Values – Reserved

DW	Field	Bits	Description
		2	**Dependent On Previous (D)** This field indicates if this Bulk request transfer is dependent on the previous Bulk request. If the previous Bulk request stalls on a NACK, the execution or cancellation of Bulk request with this field set to 1b relies on the CANCEL_OR_RESUME_BULK_REQUEST (refer Section 3.3.2.1) sent by the Host. The field value definitions are as listed below: 0b – Bulk request is not dependent on previous Bulk request transfer 1b – Bulk request is dependent on previous Bulk request transfer.
		31:3	**Reserved** This field shall be set to all zeros.
1	COMMAND_BLOCK_HEADER	15:0	**Request ID** This field contains the request number for the command and data block. The Request IDs in a Bulk request transfer shall be monotonically increasing. The Host shall ensure no outstanding requests have the same Request ID.
		16	**Has Data (HD)** This field is set to 0b if the command and data block does not have any data appended after the Command Descriptor. This field is set to 1b if there is a data block appended after the Command Descriptor.
		31:17	**Reserved** This field shall be set to all zeros.
2	COMMAND_DESCRIPTOR	127:0	**Command Descriptor** Refer Section 3.4.8.1.1
6	DATA_BLOCK	Varies	**Data Block** This field is valid if *Has Data* field in *COMMAND_BLOCK_HEADER* is 1b. This field contains the data associated with the CCC or the Write command in the Command Descriptor.
...	...	...	Additional Command and Data blocks.

3.4.8.1.1 Command Descriptor

This structure is used to define an I3C Command including its parameters. Figure 3-8 illustrates the format of this data structure.

Figure 3-8: Command Descriptor Data Structure



The fields of Command Descriptor data structure are defined in Table 3-31.

Table 3-31: Command Descriptor Data Structure Fields

DW	Field	Bits	Description
0	COMMAND_TYPE	2:0	**Command Type (CT)** This field contains the Command Type which defines the format of other fields in the Command Descriptor. The field value definitions are as listed below: 0h – Regular Command 1h – CCC without Defining Byte 2h – CCC with Defining Byte 3h – Target Reset Pattern Other Values – Reserved for future use *Note: The Target Reset Pattern (CT = 3h) is not a CCC. When Command Type is set to 3h, the Host shall build the sequence of dependent commands as described in Section 5.1.11.1 of [MIPII3C] and all other fields in the Command Descriptor shall be set to zero.* *If a Bulk request transfer contains a Command Descriptor structure for a Target Reset Pattern (CT = 3h), then it is recommended that such Bulk requests only contain Command Descriptors that are Target Resets Patterns (CT = 3h) and/or Command Descriptors for the RSTACT CCC with Defining Byte (CT = 2h), as per the defined flows in the [MIPII3C].* *A Bulk request transfer with the Target Reset Pattern (CT = 3h) should not contain other CCCs (i.e., not the RSTACT CCC) or any Regular Commands, as these should be handled separately. Additionally, Error Handling should always be used (i.e., EH = 0h) for any RSTACT CCCs that precede a Target Reset Pattern (CT = 3h) in the same Bulk request transfer."*
	READ_OR_WRITE	3	**Read Or Write (RW)** This field indicates the direction of command. This field is set to 1b for Read. This field is set to 0b for Write.

DW	Field	Bits	Description
	ERROR_HANDLING	7:4	**Error Handling (EH)** This field indicates the error condition in which the I3C Controller shall terminate the execution of subsequent I3C Commands in the list of dependent commands in the Bulk request transfer. This field indicates the direction of command. 0h – Terminate on any error (where NACK is an error) 1h – Terminate on any error except NACK (where NACK is not an error) 2h – Don't Terminate on error including NACK 3h – Terminate on Short Read 4h – Terminate on any error, but stall execution on NACK Other Values – Reserved *Note:* - *Short Read is only applicable for Read commands.* - *The specific behavior of an I3C Controller upon receiving NACK before stalling the execution of a command is implementation specific, and beyond the scope of this specification. The I3C Controller may perform retries upon receiving NACK from a Target device before it stalls execution and notifies the Host.*
	TARGET_ADDRESS	15:8	**Target Address** This field contains 7 bits (bits 14:8) of the Target device address and 1 bit (bit 15) set to 0.
	TRANSFER_MODE	20:16	**Transfer Mode (TM)** This field indicates the transfer mode for the I3C or I2C commands as defined by [MIPII3C]. *Note: For I3C HDR-TS Mode, the I3C Controller determines whether Legacy or Pure Bus mode should be used for a transfer, based on whether any I2C Target devices are present on the I3C bus.* Value – Description 0h – I3C SDR Mode 1h – I3C HDR-DDR Mode 2h – I3C HDR-TS Mode 3h – I3C HDR-BT Mode 7h - 4h – Reserved for future HDR Modes 8h – I2C Mode Other Values – Reserved for future use.

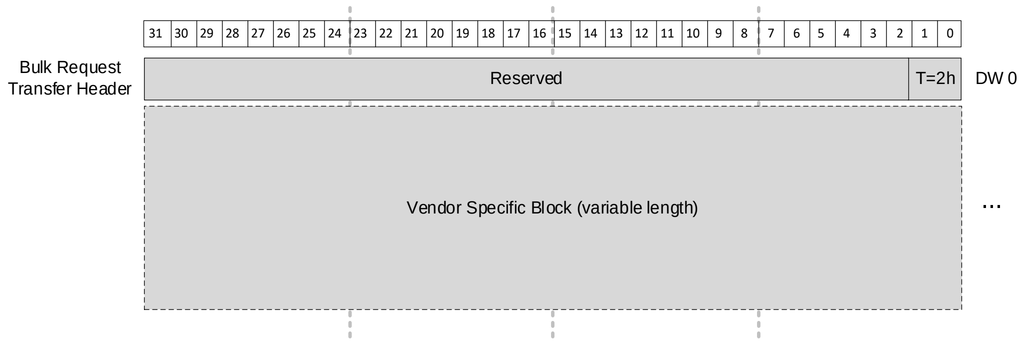
DW	Field	Bits	Description
	TRANSFER_RATE	23:21	**Transfer Rate (TR)** This field indicates the transfer rate for the selected transfer mode in *TRANSFER_MODE* field. The I3C transfer rates are applicable when *TRANSFER_MODE* field is set to one of the I3C modes. I3C Transfer Rate Value — Description 0h — 2 MHz 1h — 4 MHz 2h — 6 MHz 3h — 8 MHz 4h — 12.5 MHz 5h — User defined I3C data rate 1 6h — User defined I3C data rate 2 7h — Reserved for future use. The I2C transfer rates are applicable when *TRANSFER_MODE* field is set to 8h (I2C Mode). I2C Transfer Rate Value — Description 0h — 100 KHz 1h — 400 KHz 2h — 1 MHz 3h — User defined I2C data rate 1 4h — User defined I2C data rate 2 5h — User defined I2C data rate 3 7h - 6h — Reserved for future use.
	TM_SPECIFIC_INFO	31:24	**Transfer Mode Specific Information (TM Specific Info)** This field is reserved for Transfer Mode specific information as defined by [MIPII3C]. *Note: Details pertaining to this field are expected to be included in future revisions of this specification.*
1	CT_SPECIFIC_INFO	7:0	**Defining Byte Present** This field contains the Defining Byte for the CCC if the COMMAND_TYPE field is set to 2h.
		15:8	**Common Command Code (CCC)** This field contains the value for CCC if the COMMAND_TYPE field is either set to 1h or 2h. Refer Section 5.1.9 of [MIPII3C] for values of CCC. *Note: Field CT_SPECIFIC_INFO shall be used for HDR specific Command field, as specified by [MIPII3C].*
	Reserved	31:16	**Reserved** This field shall be set to all zeros.
2	DATA_LENGTH	21:0	**Data Length** This field indicates the number of bytes associated with a CCC, or number of bytes to be read or written as part of this Command.

DW	Field	Bits	Description
	Reserved	31:22	**Reserved** This field shall be set to all zeros.
3	Reserved	31:0	**Reserved** This field shall be set to all zeros.

3.4.8.1.2 Vendor Specific Bulk Request Transfer

Figure 3-9 illustrates the format of a vendor specific Bulk Request data structure. The Vendor Specific Block in the structure shall be 32-bit aligned, and the Host shall pad the high-order bits of the Vendor Specific Block with 0's if it is not 32-bit aligned.

Figure 3-9: Vendor Specific Bulk Request Data Structure



The fields of a vendor specific Bulk Request data structure are defined in Table 3-32.

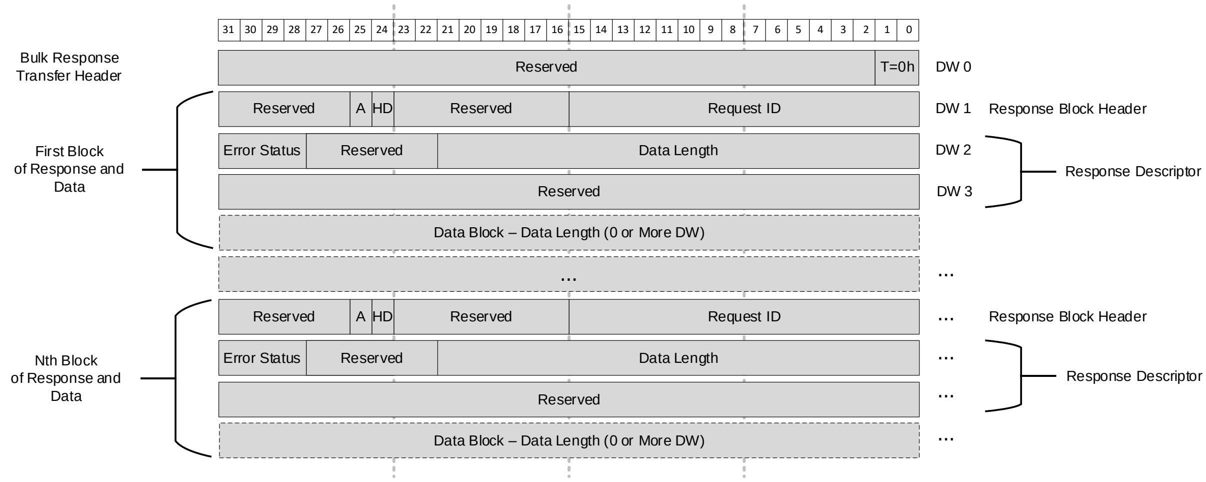
Table 3-32: Vendor Specific Bulk Request Data Structure Fields

DW	Field	Bits	Description
0	BULK_REQUEST_TRANSFER_HEADER	1:0	**Tag (T)** This field shall be set to 2h for a vendor specific Bulk request transfer.
		31:2	**Reserved** This field shall be set to all zeros.
...	Vendor Specific Block	...	**Vendor Specific Block** This field contains vendor defined content.

3.4.8.2 Bulk Response

This structure is sent as a Bulk response to the Host's Bulk request with one or more commands. This data structure can comprise of one or more responses along with associated data. Figure 3-10 illustrates the format of this data structure. The Data Block in the structure shall be 32-bit aligned, and the I3C Function shall pad the high-order bits of Data Block with 0's if Data Block is not 32-bit aligned.

Figure 3-10: Bulk Response Data Structure



The fields of Bulk response data structure are defined in Table 3-33.

Table 3-33: Bulk Response Data Structure Fields

DW	Field	Bits	Description
0	BULK_RESPONSE_TRANSFER_HEADER	1:0	**Tag (T)** This field indicates if the Bulk response transfer is a response to a regular Bulk request transfer, vendor specific response to a vendor specific Bulk request transfer or a Bulk response associated with an interrupt. The field value definitions are as listed below: 0h – Regular Bulk response transfer 1h – Interrupt Bulk response transfer 2h – Vendor specific Bulk response transfer Other Values – Reserved.
		31:2	**Reserved** This field shall be set to all zeros.
1	RESPONSE_BLOCK_HEADER	15:0	**Request ID** This field contains the corresponding Request ID generated by Host for a Command and Data block in the list of commands in the Bulk request transfer.
		23:16	**Reserved** This field shall be set to all zeros.
		24	**Has Data (HD)** This field is set to 0b if the response block does not have any data appended after the Response Descriptor. This field is set to 1b if there is data block appended after the Response Descriptor.

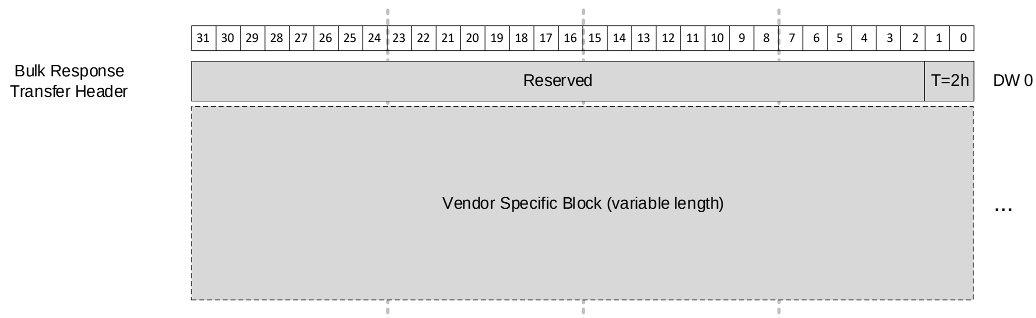
DW	Field	Bits	Description
		25	**Attempted (A)** This field indicates if the Command in the list of commands in Bulk request transfer was attempted. This field is set to 0b if the command was not attempted. This field is set to 1b if the command was attempted.
		31:26	**Reserved** This field shall be set to all zeros.
2	RESPONSE_DESCRIPTOR	21:0	**Data Length** This field indicates the number of bytes to be transferred as part of this response.
		27:22	**Reserved** This field shall be set to all zeros.
		31:28	**Error Status** This field indicates the status for the processed command. A value of 0h indicates a successful command. The field value definitions are as listed below: **Value** — **Description** 0h — Success 1h — CRC error 2h — Parity error 3h — Frame error 4h — Address Header or Broadcast Address error 5h — NACK received from a Target device. For example, Target address not acknowledged or Dynamic Address Assignment not acknowledged *Note: Based on the implementation, the I3C Controller may attempt retries, or stall execution and sent a notification (refer Section 3.4.1.6) to the Host. The exact behavior of the I3C Controller upon receiving NACK before stalling the execution of a command is implementation specific, and beyond the scope of this specification.* 6h — Reserved 7h — Short read error 8h — I3C Controller error 9h — Write data or I3C Bus transfer error Ah — Bad Command or Command not supported Bh — Command aborted with CRC error Ch to Fh — Implementation specific Transfer type error

DW	Field	Bits	Description
		63:32	**Reserved** This field shall be set to all zeros.
4	DATA_BLOCK	Varies	**Data Block** This field is valid if field *HAS_DATA* in the *RESPONSE_BLOCK_HEADER* is 1b. This field contains the data associated with the Response.
	…	…	Additional Response and Data blocks.

3.4.8.2.1 Vendor Specific Bulk Response Transfer

Figure 3-11 illustrates the format of a vendor specific Bulk Response data structure. The Vendor Specific Block in the structure shall be 32-bit aligned, and the I3C Function shall pad the high-order bits of Vendor Specific Block with 0's if it is not 32-bit aligned.

Figure 3-11: Vendor Specific Bulk Response Data Structure



The fields of a vendor specific Bulk Response data structure are defined in Table 3-34.

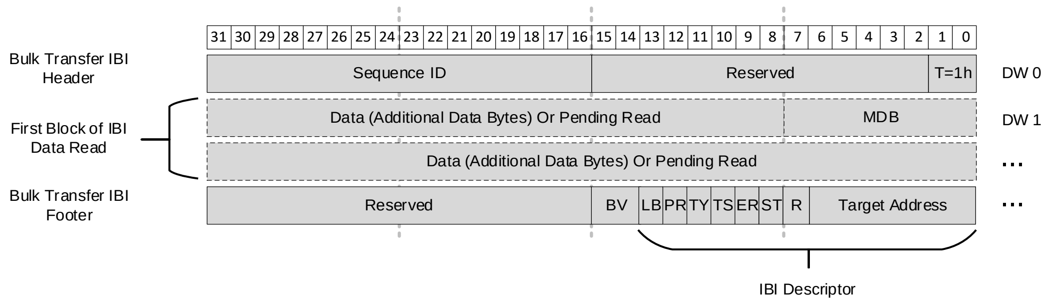
Table 3-34: Vendor Specific Bulk Response Data Structure Fields

DW	Field	Bits	Description
0	BULK_RESPONSE_TRANSFER_HEADER	1:0	**Tag (T)** This field shall be set to 2h to indicates the Bulk response transfer is a vendor specific Bulk response transfer.
		31:2	**Reserved** This field shall be set to all zeros.
…	VENDOR_SPECIFIC_BLOCK	…	**Vendor Specific Block** This field contains vendor defined content.

3.4.8.3 In-Band Interrupt Bulk Response

This structure contains data associated with an In-Band Interrupt on the I3C Bus. This data structure is sent to the Host, and comprises of single IBI along with the IBI data read by the I3C Controller. Figure 3-12 illustrates the format of this data structure. The Data block in the structure shall be 32-bit aligned, and the I3C Function shall pad the high-order bits of Data block with 0's if Data block is not 32-bit aligned.

Figure 3-12: In-Band Interrupt Bulk Response Data Structure



The fields of In-Band Interrupt Bulk Response data structure are defined in Table 3-35.

Table 3-35: In-Band Interrupt Bulk Response Data Structure Fields

DW	Field	Bits	Description
0	IBI_BULK_RESPONSE_HEADER	1:0	**Tag (T)** This field shall be set to 1h for Interrupt Bulk response transfer.
		15:2	**Reserved** This field shall be set to all zeros.
		31:16	**Sequence ID** This field shall be monotonically increasing number at 0h and incrementing by 1 for subsequent Bulk response transfers if the IBI data is split across multiple Bulk response transfers.
1, etc.	IBI_DATA	Varies	**IBI Data** This field contains the IBI data read and streamed by the I3C Controller for Bulk response transfer. If *Pending Read* (PR) field in the *IBI_BULK_RESPONSE_FOOTER* is 0b, then this field is used for the IBI data payload. The first such field (i.e., DWORD 1 in this data block) shall contain the mandatory data byte (MDB) along with any additional data bytes. Subsequent fields (if they exist, i.e., DWORDs 2, etc.) shall contain additional data bytes in the payload, to be populated per the length of the IBI data payload. If *Pending Read* (PR) field in the *IBI_BULK_RESPONSE_FOOTER* is 1b, then this field is used for data read through pending read (i.e., Auto-Command) by the I3C Controller. Data read through pending read occurs after the IBI data payload ends or is terminated, and the I3C Controller shall initiate a subsequent I3C read transfer (refer Section 4.6.4).
...	IBI_BULK_RESPONSE_FOOTER	6:0	**Target Address** This field contains 7 bits of Target address for the 1st command.
		7	**Read or Write (RW)** This field is set to 0b for Write. This field is set to 1b for Read.

DW	Field	Bits	Description
		8	**IBI Status (ST)** This field indicates if the I3C Controller acknowledged the IBI. This field is set to 0b if IBI was acknowledged. This field is set to 1b if IBI was not acknowledged.
		9	**Error (ER)** This field indicates if this interrupt was caused due to an error in execution of a command. This field is set to 0b if no error was encountered during a command execution. This field is set to 1b if error was encountered during a command execution.
		10	**IBI Timestamp (TS)** This field indicates if an IBI is timestamped. This field is set to 0b if IBI is not timestamped. This field is set to 1b if IBI is timestamped.
		11	**IBI Type (TY)** This field indicates if this IBI is a regular IBI or from a scheduled command or a Secondary I3C Controller. This field is set to 0b if IBI is a regular IBI. This field is set to 1b if IBI is from a scheduled command or from a Secondary I3C Controller.
		12	**Pending Read (PR)** This field indicates if this IBI data is read through pending read request from the I3C Controller. This field is set to 0b if IBI data is not read as pending read request. This field is set to 1b if IBI data is read as pending read request.
		13	**Last Byte (LB)** This field indicates if IBI data block has the last IBI data byte read by the I3C Controller. This field is set to 0b if the IBI data block does not contain the last IBI data byte. This field is set to 1b if the IBI data block contains the last IBI data byte.
		15:14	**Bytes Valid (BV)** This field indicates number of valid bytes in the last DWORD before this footer. The field value definitions are as listed below: 00b – All four bytes are valid, no padding 01b – First byte is valid, remaining bytes are zero padded 10b – First two bytes are valid, remaining bytes are zero padded 11b – First three bytes are valid, remaining bytes are zero padded

DW	Field	Bits	Description
		31:16	**Reserved** This field shall be set to all zeros.

4 Operational Model

This section describes the typical operational details of the USB Device with I3C Function, and ties together the descriptors, class-specific requests and data structures used for various operational flows. It provides guidance for the implementer of the Device and lists requirements and expectations of the Host's software stack.

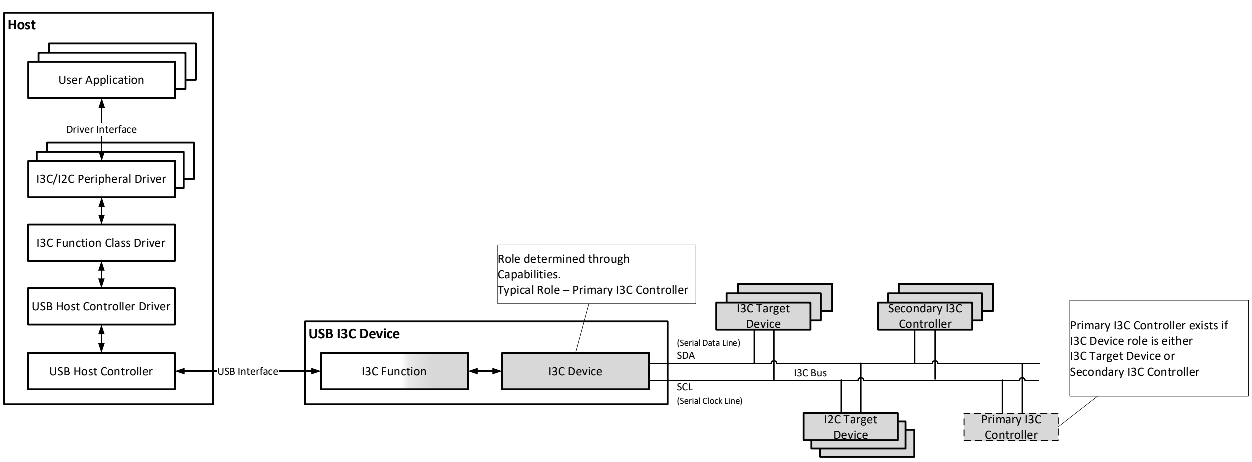
4.1 Initialization and Configuration

The detection of a USB Device with I3C Function connected to a USB Host occurs through the USB port hardware. Enumeration of a USB Device with I3C Function occurs through standard USB requests. The I3C Function becomes operational after successful completion of Set Configuration request.

Figure 4-1 below illustrates a USB Device with an I3C Function. The I3C Device inside such a USB Device shall support one of the following three roles:

1. I3C Controller role (Primary I3C Controller).
2. I3C Target device role.
3. I3C Target device capable of Secondary Controller role.

Figure 4-1: USB I3C Device Topology



*Note: * An I3C Device can take the role of an I3C Controller (Primary I3C Controller), an I3C Target device, or an I3C Target device capable of I3C Secondary Controller role.*

After the USB Device is detected and I3C Function successfully enumerated, the Host sends the SET_FEATURE request with *bInterfaceNumber* and *bAlternateSetting* fields set to zero, to initialize the I3C Function. Until the I3C Function is initialized, any messages, interrupts or transfers from a previously configured I3C Controller shall be dropped. As part of I3C Function initialization, the I3C Bus shall be disabled to ensure the I3C Bus was not left in an unknown state from previous operations or failures. This ensures that the I3C Device will respond to this request if it is controlling the I3C Bus.

Note: An I3C Device in the I3C Target device role shall not respond to the request for disabling I3C Bus.

As part of I3C Function initialization, this I3C Function shall configure the I3C Controller (i.e., Primary I3C Controller). The Host then sends the GET_I3C_CAPABILITY request to determine the role of the I3C Device and the data type in the I3C Capability structure (refer Table 3-27). When an I3C Device does not contain I3C Capability data, the I3C Function shall return the *I3C_CAPABILITY_HEADER* of I3C Capability data structure with field *Total Length* set to 4 bytes, fields *Device Role* and *Data Type* set to 0h, and the *Error Code* field set to FFh. In this case, the Host shall assume that this I3C Device has the I3C Controller role and has no knowledge of Target devices on I3C Bus (refer Section 4.1.1.1), and that this I3C Device supports basic MIPI I3C functionality as described in version 1.1.1 or newer (see [MIPII3C] or [MIPII3CBASIC]).

Initialization and configuration for each role is described below. The I3C Device shall check for I3C Bus conditions needed to perform transactions on the I3C Bus, as specified in Section 5.1.3 of [MIPII3C].

4.1.1 I3C Controller Role (Primary I3C Controller)

The I3C Controller performs the initial configuration of the I3C Bus and all the Target devices on the I3C Bus, including dynamic address assignment of the I3C Target devices.

The I3C Controller shall have the static information about itself. However, the static information related to I3C Target devices on the I3C Bus may or may not be available with the I3C Controller.

It is recommended that any existing static information pertaining to the Target devices on the I3C Bus including their roles and capabilities (refer I3C Capability in Section 3.4.5) be stored in the I3C Controller, otherwise this static information shall be sent from the Host to the I3C Controller.

Refer I3C Capability data structure in Section 3.4.5 for details on the static information.

The I3C Function Driver in the Host system ensures that the I3C Function/Interface in the USB Device is initialized and available for use by Host application/s. The Host application or other Host software logic then determines when to enable the I3C Bus, and performs subsequent operations with USB Device (i.e., through the I3C Function Driver). Typical initialization and configuration flows are described below.

1. When the *Device Role* field in I3C Capability structure is set to 0x1 and the *Data Type* field is set to 0x1, this indicates that the I3C Controller has the knowledge of Target devices on the I3C Bus (refer Figure 4-2):

    a. The Host sends INITIALIZE_I3C_BUS request with Address Assignment Mode set to 0h (refer Section 3.3.2.8).

    b. The I3C Controller performs I3C Target device discovery based on the Address Assignment Mode and assigns dynamic addresses to the I3C Target devices (refer [MIPII3C] Section 5.1.4 for additional details).

    c. The I3C Function generates and stores the Target Device Table (refer Table 3-28).

    Alternately, when the *Device Role* field in I3C Capability structure is set to 0x1 and *Data Type* field is set to 0x2, this indicates that the I3C Controller has no knowledge of Target devices on the I3C Bus (refer Figure 4-3):

    a. The Host sends INITIALIZE_I3C_BUS request with Target Device Table and Address Assignment Mode (refer Section 3.3.2.8).

    b. The I3C Controller performs I3C Target device discovery based on the Target Device Table and Address Assignment Mode received from the Host and assigns dynamic addresses to the I3C Target devices (refer Section 4.1.1.1 and [MIPII3C] Section 5.1.4 for additional details).

c. The I3C Function updates and stores the Target Device Table (refer Table 3-28).

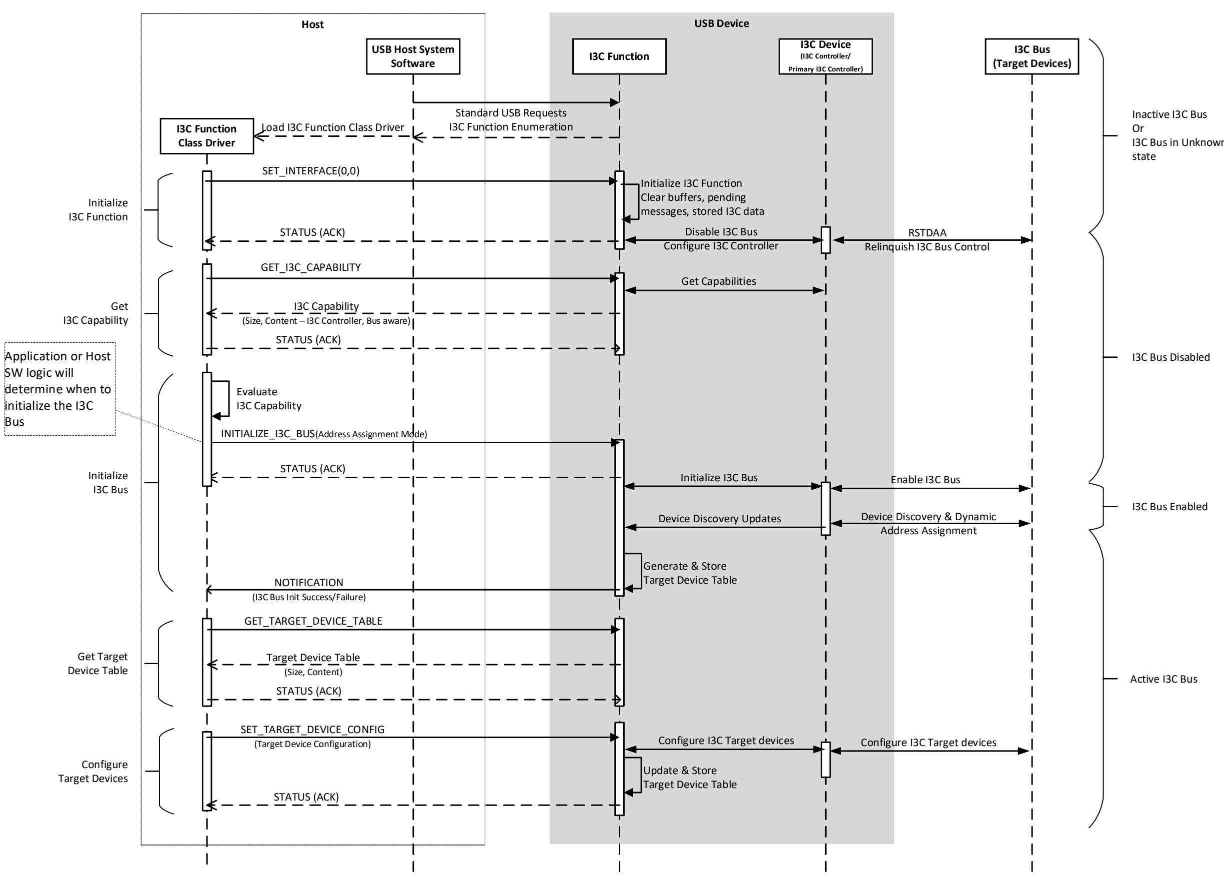
Note: The *I3C Controller should handle the scheduling of operations and events/interrupts generated on the I3C Bus, such as Hot-Join during device discovery and address assignment.*

2. The I3C Function sends a notification on Interrupt-IN endpoint to Host indicating success or failure of I3C Bus initialization (refer Section 3.4.1.1).

3. On receiving a Notification indicating failed initialization of the I3C Bus, the Host may retry the INITIALIZE_I3C_BUS request.

4. On receiving a Notification indicating successful initialization of the I3C Bus, the Host shall subsequently issue GET_TARGET_DEVICE_TABLE request (refer Section 3.3.2.7) to get the updated Target Device Table (refer Table 3-28). The Host may optionally send the SET_TARGET_DEVICE_CONFIG request (refer Section 3.3.2.10) to configure the Target devices (refer Table 3-29 for configurable parameters).

Figure 4-2: Initialization and Configuration Flow - I3C Controller aware of Target devices on I3C Bus



Note:

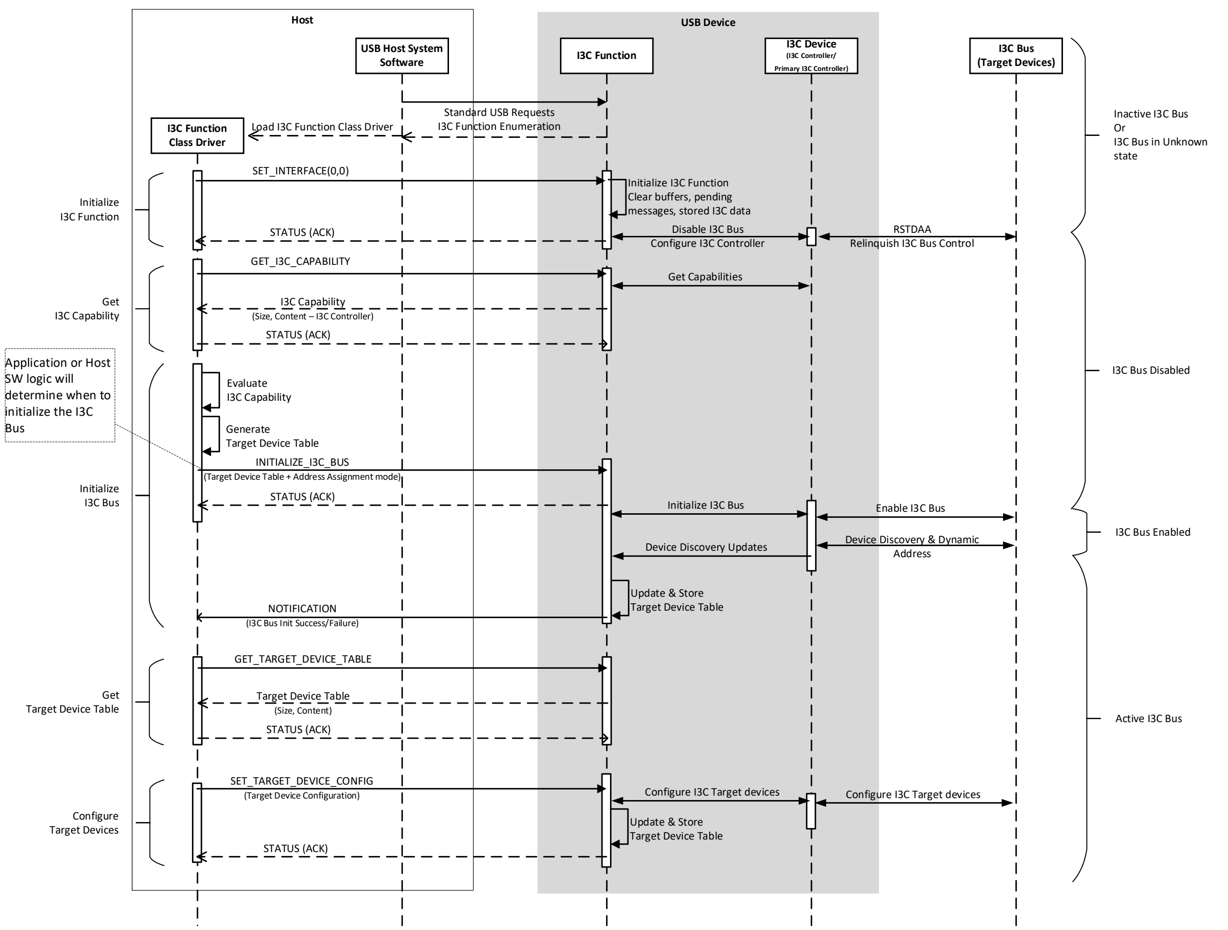
Inactive I3C Bus: *I3C Controller does not control the I3C Bus and the Target devices on the I3C Bus are unknown and not functional.*

I3C Bus Disabled: *I3C Controller does not control the I3C Bus. I3C Controller shall not respond to any actions initiated by Target devices on the I3C Bus.*

I3C Bus Enabled: *I3C Controller controls the I3C Bus. I3C Controller shall respond to actions initiated by Target devices on the I3C Bus. This is the I3C Bus state before the initial Target device discovery and address assignment. I3C Bus can be in Bus Free, Bus Available or Bus Idle condition (refer Section 5.1.3.2 of [MIPII3C]) in this state.*

Active I3C Bus: *I3C Controller controls the I3C Bus, and the Target devices on the I3C Bus are functional. I3C Bus is active and Target devices are addressable. I3C Bus can be in Bus Free, Bus Available or Bus Idle condition (refer Section 5.1.3.2 of [MIPII3C]) in this state.*

Figure 4-3: Initialization and Configuration Flow - I3C Controller unaware of Target devices on I3C Bus



4.1.1.1 I3C Target Device Discovery and Address Assignment (Host sends Target Device Table)

Addresses for I2C Target devices are static and cannot be changed using I3C CCCs or other standard I3C Commands. When Host sends the Target Device Table to the I3C Function during initialization, the *Target Address* field (refer Table 3-28) shall contain this static address. During initialization, the

Active I3C Controller shall reserve these addresses and not use them for Dynamic Address Assignment (i.e., for any other I3C Target devices that might be present).

Field *Target Address* in the Target Device Table (refer Table 3-28) holds the dynamic address for I3C Target devices. The dynamic address may subsequently be changed, either by the Host sending the CHANGE_DYNAMIC_ADDRESS request (refer Section 3.3.2.2) or by sending a CCC (refer Section 5.1.9.3.11 of [MIPII3C]).

When the I3C Controller has no knowledge of Target devices on the I3C Bus, the Host sends the Target Device Table during I3C Bus initialization. I3C Target devices can each be assigned a dynamic address by one of the methods listed below:

- **Using the ENTDAA CCC:** This method includes dynamic address assignment along with discovery. If the *Dynamic Address Assignment with ENTDAA* field in Target Device Table (refer Table 3-28) is set to 1b, the initial dynamic address (refer to the *Target Address* field Table 3-28) shall be assigned to the I3C Target with matching *Bus Characteristic Register*, *Device Characteristic Register* and Provisional ID (*Provisional ID Low* and *Provisional ID High*) fields in Target Device Table (refer Table 3-28).
- **Using the SETDASA directed CCC:** If *Assignment from Static Address* field in Target Device Table (refer Table 3-28) is set to 1h, the initial dynamic address of the I3C Target device shall be assigned based on the known static address of the I3C Target, as indicated by *Target Address* field (refer Table 3-28).
- **Using the SETAASA broadcast CCC:** This method sets all dynamic addresses from static addresses. If *Assignment from Static Address* field) field in Target Device Table (refer Table 3-28) is set to 2h, the initial dynamic address of each I3C Target device shall be the same as the known static address of the I3C Target device indicated by *Target Address* field (refer Table 3-28).

Note: All I3C Target devices present on the I3C Bus that are operational and support SETAASA CCC will be configured simultaneously when this CCC is sent. However, with this CCC there is no method for the I3C Active Controller to determine whether dynamic address assignment using SETAASA was successful for any particular I3C Target device.

If the Host does not set any of Assignment from Static Address and Dynamic Address Assignment with ENTDAA fields for a I3C Target device, that I3C Target device shall not be configured during I3C Bus initialization.

4.1.2 I3C Target Device Role

When the I3C Device has the I3C Target device role, the Host can request that I3C Target device will use the Hot-Join Request to join an already initialized I3C Bus. Figure 4-4 illustrates the flow for an I3C Target device.

1. When the *Device Role* field in I3C Capability structure (refer Table 3-27) is set to 0x2 and the field *Data Type* field is set to 0x2, the Host sends a Bulk request transfer with a command to Hot-Join an existing initialized I3C Bus (refer [MIPII3C] Section 5.1.5).

2. The I3C Target device notifies the I3C Function of Hot-Join success or failure.

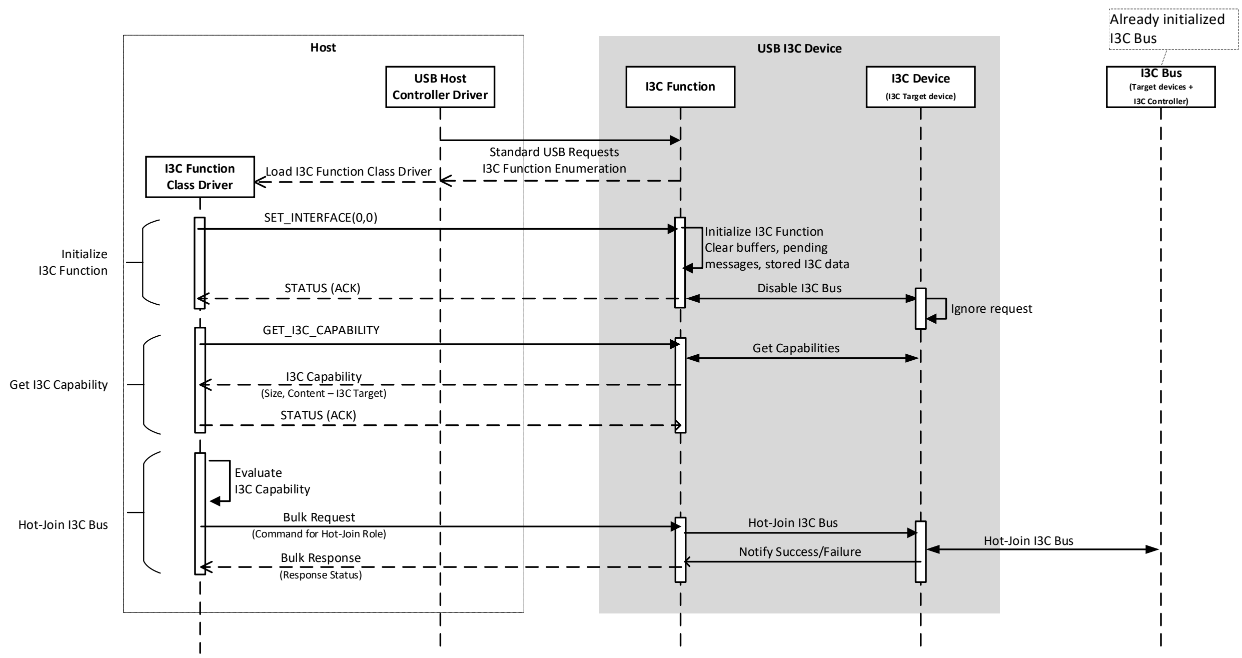
3. The I3C Function generates a Bulk response transfer which contains the *RESPONSE_DESCRIPTOR* with *Error Status* generated by I3C Target device (refer Section 3.4.8.2).

4. On receiving a Notification indicating failed Hot-Join of the I3C Device, the Host may retry Bulk request for Hot-Join.

5. On receiving a Notification indicating successful Hot-Join of the I3C Device, the Host may send Bulk requests (refer Section 3.4.8.1) for transferring data or commands on the I3C Bus.

Note: Additional details for I3C Target Device Role are expected to be added in future revisions.

Figure 4-4: I3C Device as I3C Target Device



4.1.3 I3C Target Device capable of Secondary I3C Controller Role

When the I3C Device is an I3C Target device capable of Secondary I3C Controller role, the Host can request the I3C Target device would use the Hot-Join Request (similar to an I3C Target role). Figure 4-5 illustrates the flow for an I3C Target device capable of Secondary I3C Controller role.

1. When the *Device Role* field in I3C Capability structure (refer Table 3-27) is set to 0x3 and the field *Data Type* field is set to 0x2, the Host sends a Bulk request transfer with a command to Hot-Join an existing initialized I3C Bus (refer [MIPII3C] Section 5.1.5).

2. The I3C Target device notifies the I3C Function of Hot-Join success or failure.

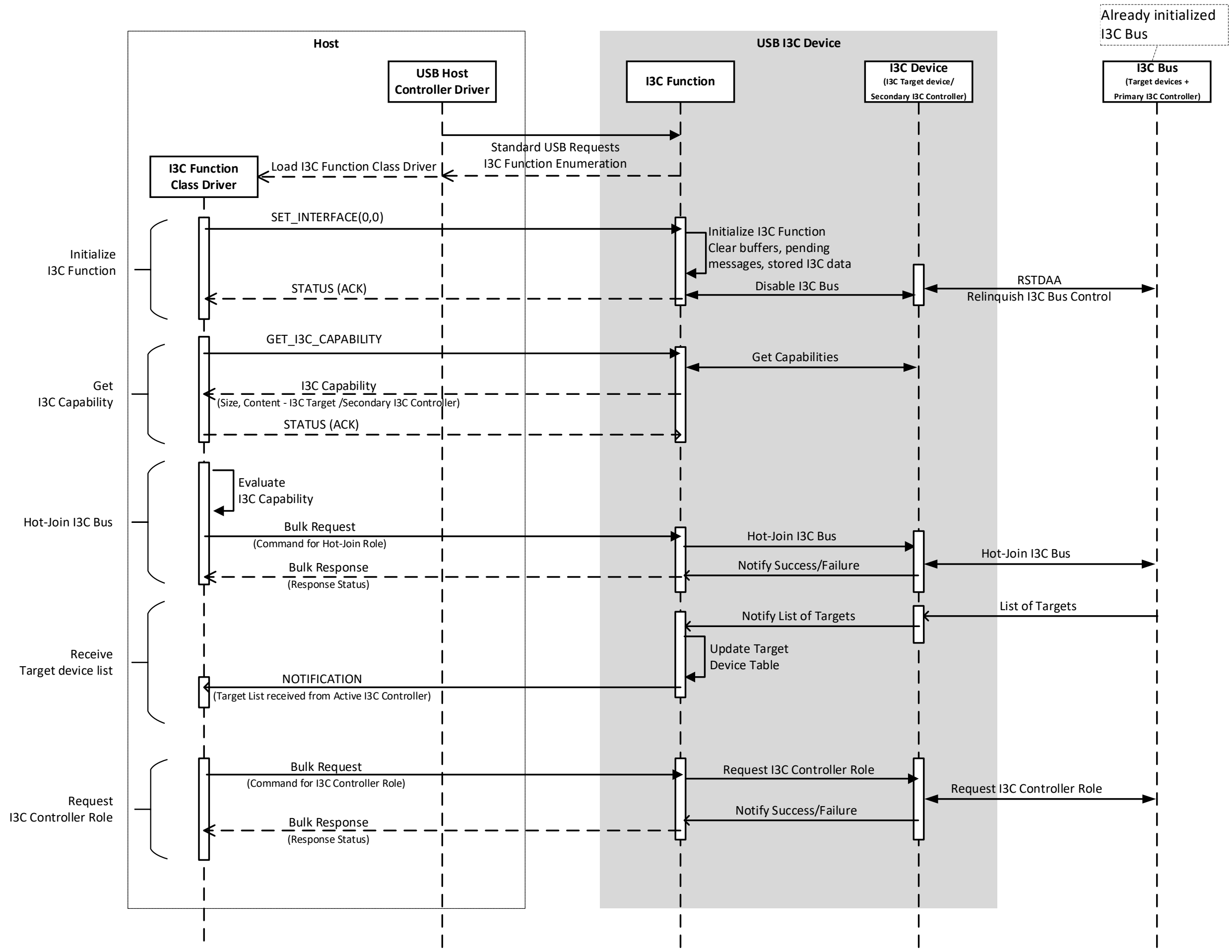
3. The I3C Function generates a Bulk response transfer which contains the *RESPONSE_DESCRIPTOR* with *Error Status* generated by I3C Target device (refer Section 3.4.8.2).

4. On receiving a Notification indicating failed Hot-Join of the internal I3C Device, the Host may retry Bulk request for Hot-Join.

5. When the I3C Target device successfully Hot-Joins an I3C Bus, it may receive a list of Target devices on I3C Bus from the Active I3C controller on the I3C Bus.

6. On receiving a Notification indicating successful Hot-Join, the Host may subsequently issue a Bulk request transfer (refer Section 3.4.8.1) with a command to request the I3C Controller role (i.e., to become the new Active Controller).

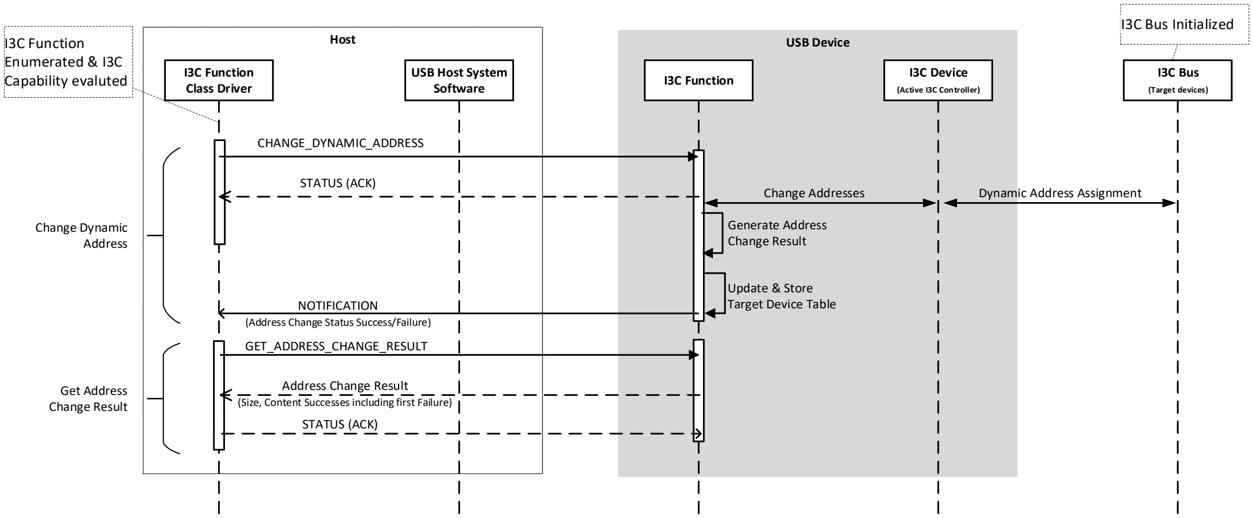
Figure 4-5: I3C Device as I3C Target device capable of Secondary I3C Controller role



4.2 Change I3C Target Device Dynamic Address

The Host may change the previously assigned Dynamic Address of one or more I3C Target devices, provided that the I3C Target device supports this operation. Figure 4-6 illustrates the sequence of operations involved in changing the dynamic address and retrieving the results.

Figure 4-6: Dynamic Address Change Sequence



After successful completion of the CHANGE_DYNAMIC_ADDRESS transfer, the I3C Function shall perform the following operations:

1. Change the Dynamic Address of the I3C Target device to the value specified in *New Dynamic Address* field (refer Table 3-24).
    a. The I3C Function shall use the SETNEWDA CCC for each specified I3C Target.
2. Generate a data structure with the list of *Current Dynamic Address*, *New Dynamic Address* and *Success/Failure* status for each Address Change entry, including the first encountered failure (if any) (refer Table 3-25).
3. Update the stored Target Device Table to include the changed Dynamic Addresses of Target devices.
4. Send a notification (refer Section 3.4.1.2) on the Interrupt-IN endpoint to indicate “success” to the Host, if all Address Change entries (refer, Table 3-24) were successfully changed; or send a “failure” notification (refer Section 3.4.1.2) on encountering the first error.
    a. On encountering any error, the I3C Function shall not attempt to change any subsequent entries in the Address Change data structure (refer Table 3-24) after the entry that caused the error. Reasons for an error include (but are not limited to) an I3C Target failing to acknowledge (ACK) the SETNEWDA CCC for the attempted change of Dynamic Address; a Current Dynamic Address value that is not known to the I3C Function; a New Dynamic Address value that is in conflict with a current Dynamic Address of any other Target on the I3C Bus; or any other I3C Bus error that occurs during the Address Change operation.
5. Upon receiving notification (refer Section 3.4.1.2) the Host uses the GET_ADDRESS_CHANGE_RESULT request described in Section 3.3.2.4 to get the result of Address Change for the affected Target devices.
    a. In case of failure, the Host may retry the request to change the addresses.
    b. In case of success, the Host may retrieve the updated Target Device Table using GET_TARGET_DEVICE_TABLE (refer Section 3.3.2.7).

4.3 Regular IBI Handling

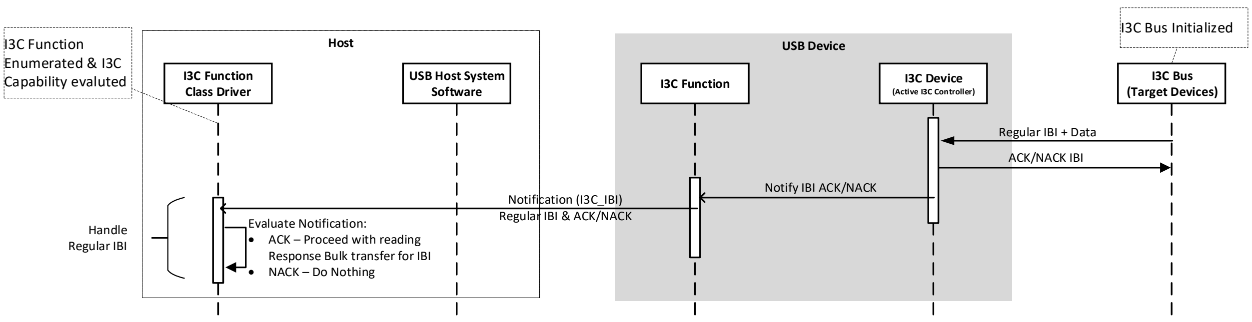
This section describes how IBIs generated on the I3C Bus are handled by the Active I3C Controller (i.e., either Primary or Secondary I3C Controller). IBIs received on the I3C Bus are sent to the Host through Interrupt notification by the I3C Function (refer Section 3.4.1.4).

When the regular IBI is acknowledged by the I3C Controller, the Host reads the Bulk response transfer to get the IBI Descriptor and associated data (refer Section 3.4.8.3).

When the regular IBI is not acknowledged by the I3C Controller, the Host does not need totake any further action.

Figure 4-7 illustrates the Regular IBI handling.

Figure 4-7: Regular IBI handling by Active I3C Controller



4.4 Target Device Hot-Join

This section describes the Hot-Join flow when I3C Device is in the I3C Controller role or I3C Target device role.

4.4.1 I3C Controller Role

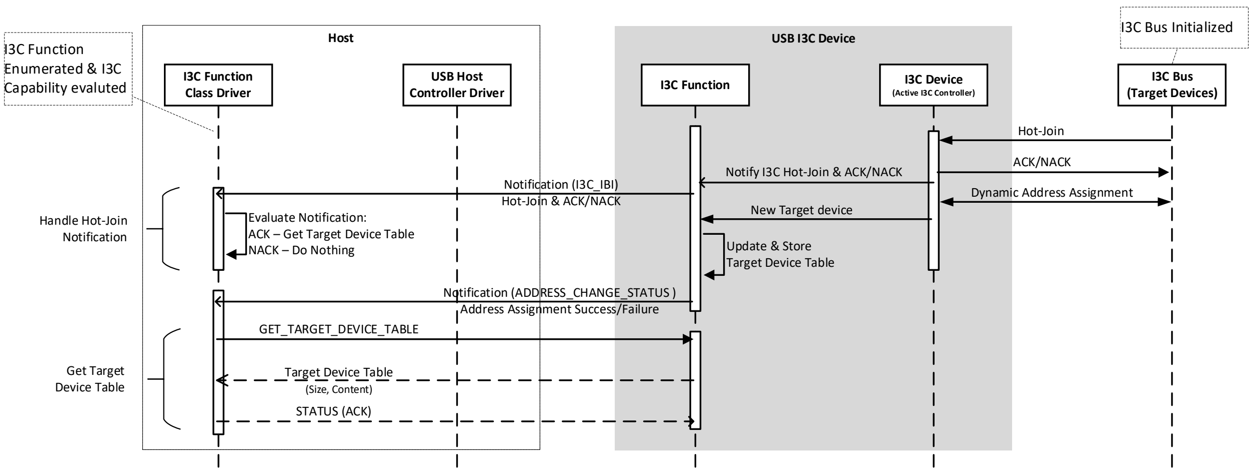
I3C Device as Active I3C Controller shall handle the Hot-Join Request, when a I3C Target device Hot-Joins an initialized I3C Bus (refer Section 5.1.5 of [MIPII3C]).

When the I3C Controller acknowledges the Hot-Join Request, it shall perform the initial Dynamic Address Assignment for all the Hot-Joined Target devices. The I3C Function shall notify the Host when one or more I3C Target devices Hot-Join the I3C Bus (refer Section 3.4.1.4). The I3C Function shall update the Target Device Table with any new entries for the Hot-Joined Target devices. The I3C Function shall notify the Host after it has completed the dynamic address assignment for the Hot-Joined Target devices and updated its Target Device Table. After receiving this notification, the Host may issue a GET_TARGET_DEVICE_TABLE request (refer Section 3.3.2.7) to get the updated Target Device Table with details about the new Target devices.

When the I3C Controller does not acknowledge the Hot-Join request, the I3C Function shall notify the Host that one or more I3C Target devices attempted to Hot-Join the I3C Bus. The Host does not need to take any further action if the Hot-Join request is not acknowledged by the I3C Controller.

Figure 4-8 illustrates how the I3C Controller handles the Hot-Join.

Figure 4-8: I3C Controller handling Hot-Join on I3C Bus



4.4.2 I3C Target Device Role

The I3C Target device Hot-Joins an already initialized I3C Bus (refer Section 5.1.5 of [MIPII3C]) when Host sends the command to Hot-Join the Bus through the Bulk request transfer (refer Section 3.4.8.1).

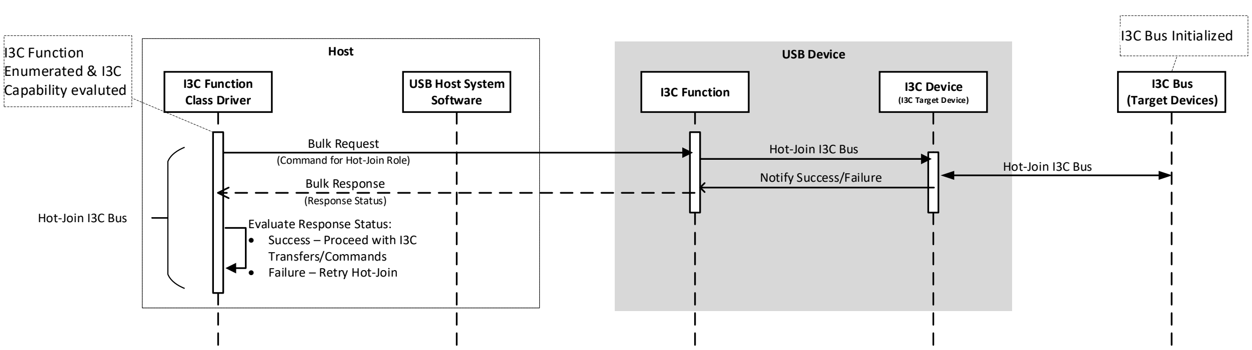
The I3C Function shall generate a Bulk response transfer which contains the *RESPONSE_DESCRIPTOR* with *Error Status* generated by I3C Target device, to indicate either success or failure of Hot-Join (refer Section 3.4.8.2).

On receiving a Bulk response transfer with failed Hot-Join on I3C Bus, the Host may retry Hot-Join through another Bulk request transfer.

On receiving a Bulk response transfer with successful Hot-Join on I3C Bus, the Host may subsequently issue Bulk request transfers to send data from the I3C Target device to the I3C Bus.

Figure 4-9 illustrates how an I3C Target device Hot-Joins an already initialized I3C Bus.

Figure 4-9: I3C Target Device Hot-Joins an initialized I3C Bus



4.5 I3C Controller role handoff

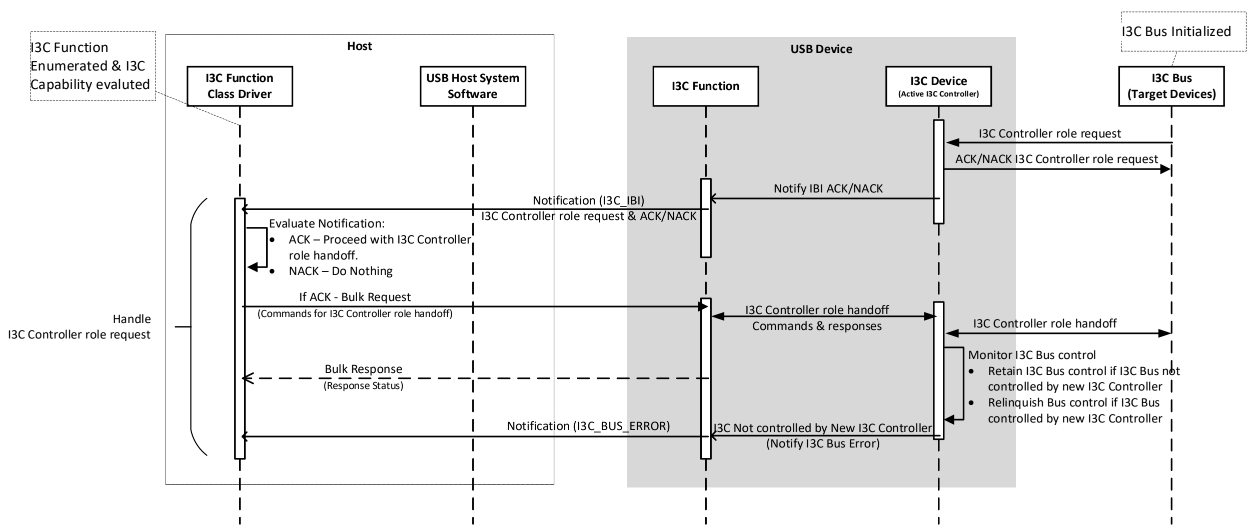
This section describes how the I3C Controller role request on the I3C Bus is handled by the Active I3C Controller (Primary or I3C Secondary Controller). The I3C Controller role request is received as an IBI on the I3C Bus, and the I3C Function sends an Interrupt notification to the Host (refer Section 3.4.1.4).

The Host evaluates the notification from the I3C Function. For an acknowledged I3C Controller role request, the Host will initiate the I3C Controller role handoff to the new I3C Controller-capable device (i.e., Secondary Controller) through Bulk request transfer. The I3C Function will then generate a Bulk response transfer which contains the *RESPONSE_DESCRIPTOR* with *Error Status* (refer Section 3.4.8.2) generated by the I3C Controller.

The I3C Controller monitors the I3C Bus and ensures the new I3C Controller drives the I3C Bus. If the new I3C Controller does not drive the I3C Bus, then the I3C Controller notifies the I3C Function. As a result, the I3C Function sends notification on the Interrupt-IN endpoint to Host indicating I3C Bus error (refer Section 3.4.1.3) and continues to control the I3C Bus (refer Section 5.1.7 of [MIPII3C]).

Figure 4-10 illustrates the I3C Controller role handoff flow.

Figure 4-10: I3C Controller role handoff by an Active I3C Controller



4.6 Bulk Requests and Responses

The Host may send a Bulk request transfer (refer Section 3.4.8.1) consisting of one or more commands along with the associated data. To ensure that I3C Function has sufficient buffer to handle a sequence of commands, the Host may evaluate the Buffer Available structure (refer Section 3.3.2.5) returned by I3C Function, to determine if it should send a Bulk request transfer of certain size. Implementation details of memory resource management within the USB Device is beyond the scope of this specification.

The amount of data sent in the *Data Block* of Bulk request (refer Figure 3-7 and Table 3-30) transfer shall match the *Data Length* specified in the *COMMAND_DESCRIPTOR* (refer Table 3-31). Similarly, the amount of data received in the *Data Block* of Bulk response transfer shall match the *Data Length* specified in the *RESPONSE_DESCRIPTOR* (refer Figure 3-10 and Table 3-33).

All Bulk request and response transfers shall be terminated with a short packet.

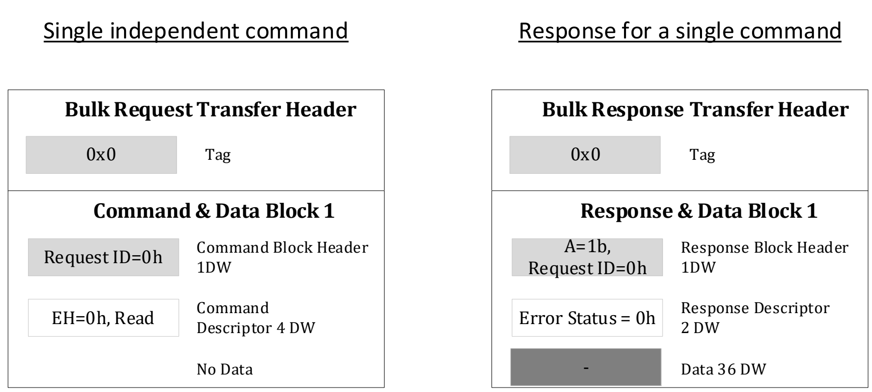
Sections 4.6.1 and Section 4.6.2 describe the Bulk Transfers for a single independent command and a list of dependent commands.

4.6.1 Bulk Transfer as a single independent command

- The Host shall generate the Bulk request transfer data structure, as indicated in Section 3.4.8.1, with a single command and data block.
- The I3C Controller shall generate the Response Descriptor with *Error Status* (refer Table 3-33) for the command executed.
- Upon encountering a failure with the command, the I3C Device shall:
  - Generate the Response Descriptor with *Error Status* (refer Table 3-33).
  - Indicate the failure to the I3C Function; and
  - Relinquish the I3C Bus control.
- The I3C Function shall send the Bulk response transfer (Section 3.4.8.2) containing a response block indicating Success/Failure for the command that was executed in the Bulk request.
- The I3C Controller shall relinquish the I3C Bus control after successfully completing the execution of the command in the Bulk request transfer.

Figure 4-11 illustrates an example for a Bulk request and response transfer for a single I3C Command and Data block. In this example, the Host sets the *Error Handling (EH)* field in Command Descriptor (refer Table 3-31) to 0h indicating that the I3C Controller shall relinquish the bus control after the execution of command.

Figure 4-11: Example - Bulk Request and Response Transfer for a single independent I3C Command with successful completion of the I3C Command



4.6.2 Bulk Transfer as a list of dependent commands

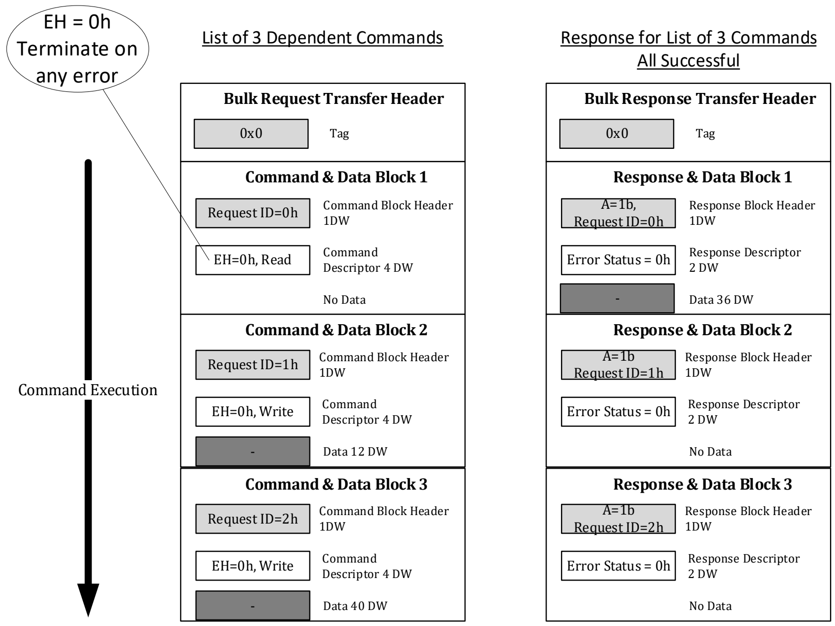
- The Host shall generate the Bulk request transfer data structure, as indicated in Section 3.4.8.1.
- The I3C Controller shall execute each of these commands in strict order, from first to last command.
- The I3C Controller shall conditionally not execute any subsequent commands in the Bulk request transfer if the previous command in the list fails. The I3C Controller shall determine whether to terminate the execution of subsequent commands on a

failure, based on the *Error Handling* field in the Command Descriptor (refer Table 3-31) sent by the Host.

- Upon encountering failure with a command, the I3C Device shall
  - Generate the Response Descriptor with *Error Status* (refer Table 3-33);
  - Indicate the failure for the specific Request ID to the I3C Function; and
  - Relinquish the I3C Bus control.
- The I3C Function shall send the Bulk response transfer (refer Section 3.4.8.2) containing response blocks for all corresponding command blocks. These response blocks shall have the *RESPONSE_BLOCK_HEADER* (refer Table 3-33).
  - For any commands that were attempted by the I3C Device, the *RESPONSE_BLOCK_HEADER* shall have the *Attempted* bit set to 1b (refer Table 3-33). The I3C Device shall generate the *RESPONSE_DESCRIPTOR* with *Error Status* (refer Table 3-33) indicating success or failure for each of the commands that are attempted by I3C Device.
  - For any commands that are not attempted by the I3C Device, the *RESPONSE_BLOCK_HEADER* shall have the *Attempted* bit set to 0b (refer Table 3-33). The I3C Device shall not generate the *RESPONSE_DESCRIPTOR* (refer Table 3-33) for the commands that are not attempted by the I3C Device.
  - Refer to Figure 4-12 and Figure 4-13 for examples of Bulk request and response transfers.
- The I3C Device shall relinquish the I3C Bus control after successfully completing the execution of all attempted commands in the Bulk request transfer.

Figure 4-12 illustrates an example for Bulk request and response transfers with three dependent commands in a list. In this example, the Host has set the *Error Handling (EH)* field in each Command Descriptor (refer Table 3-31) to 0h, indicating that the I3C Controller must terminate execution of subsequent commands on any transfer error. The example shows a Bulk response transfer where all commands in the Bulk request transfer were attempted and were successful.

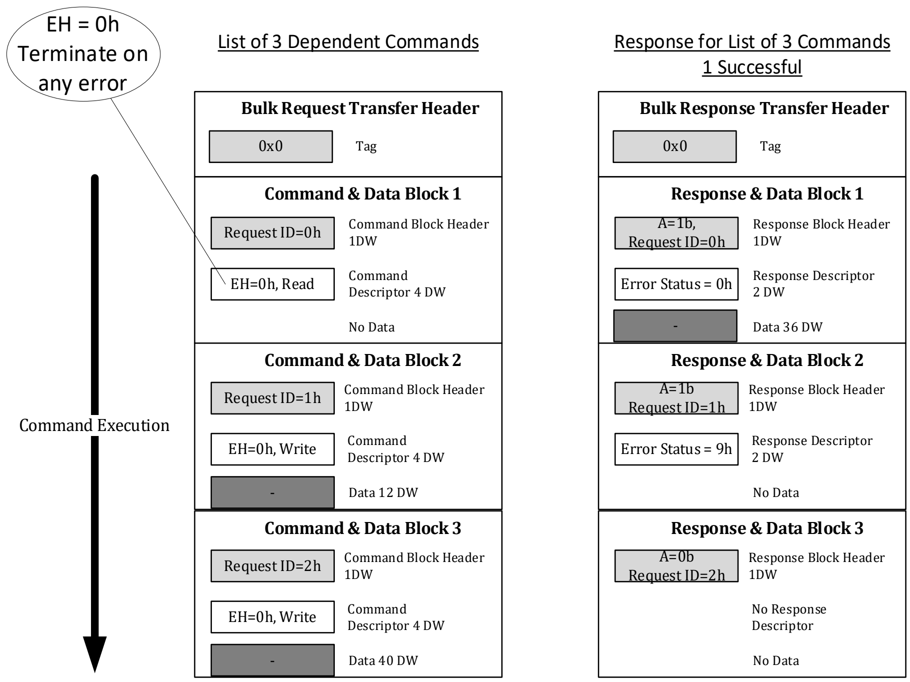
Figure 4-12: Example - Bulk Request and Response Transfer for List of I3C Commands with all successful I3C Commands



Note: The illustration does not indicate all the fields for data structures (refer Table 3-30, Table 3-31 and Table 3-33) for all the data fields.

Figure 4-13 illustrates an example for Bulk request and response transfers with three dependent commands in a list where only the first two commands are executed. The first command is successful, but the second command failed, leading to termination of execution before attempting the third command.

Figure 4-13: Example – Bulk Request and Response Transfer for List of I3C Commands with 1 successful I3C Command



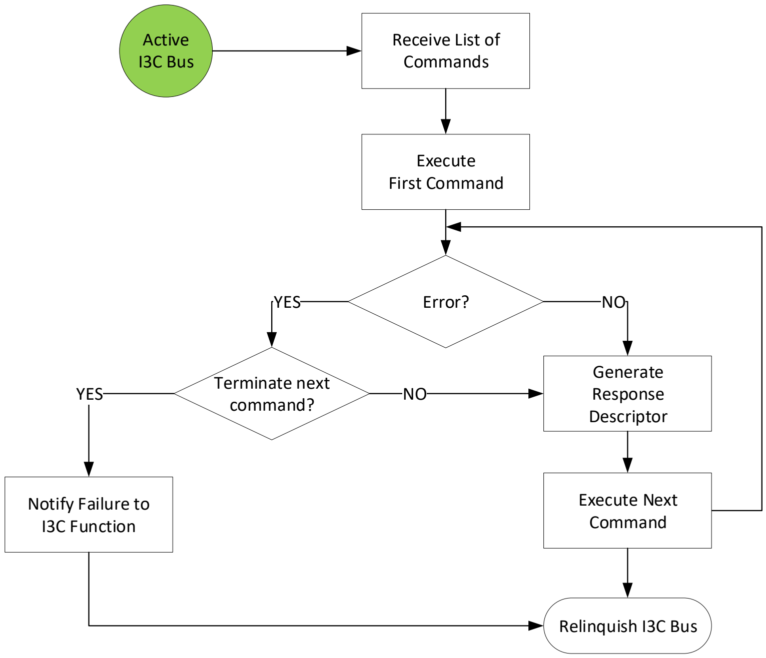
Note: The illustration does not indicate all the fields for data structures (refer Table 3-30, Table 3-31 and Table 3-33) for all the data fields.

The I3C Device executes these commands in a sequence and generates the *RESPONSE_DESCRIPTOR* with *Error Status* (refer Table 3-33) for every command it attempts to execute. The I3C Function generates fields *BULK_RESPONSE_TRANSFER_HEADER* and *RESPONSE_BLOCK_HEADER* with the *Request ID* (refer Table 3-33) for each response and data block, with the corresponding *Request ID* of the command and data block in the Bulk request transfer (refer Table 3-30).

Figure 4-14 illustrates the behavior of the I3C Device executing a list of dependent commands. The I3C Device executes the list of commands in a strict order from first to the last. As each command is executed, the I3C Device evaluates any errors encountered based on the *Error Handling* field in the Command Descriptor (refer Table 3-31) to determine if the subsequent command in the list should be executed. If the error encountered during the execution of a command leads to termination of execution of subsequent command, the I3C Controller notifies the I3C Function.

Note: Any self-initiated error recovery mechanism within I3C Device is implementation specific and beyond the scope of this specification.

Figure 4-14: I3C Device executing a list of dependent I3C Commands

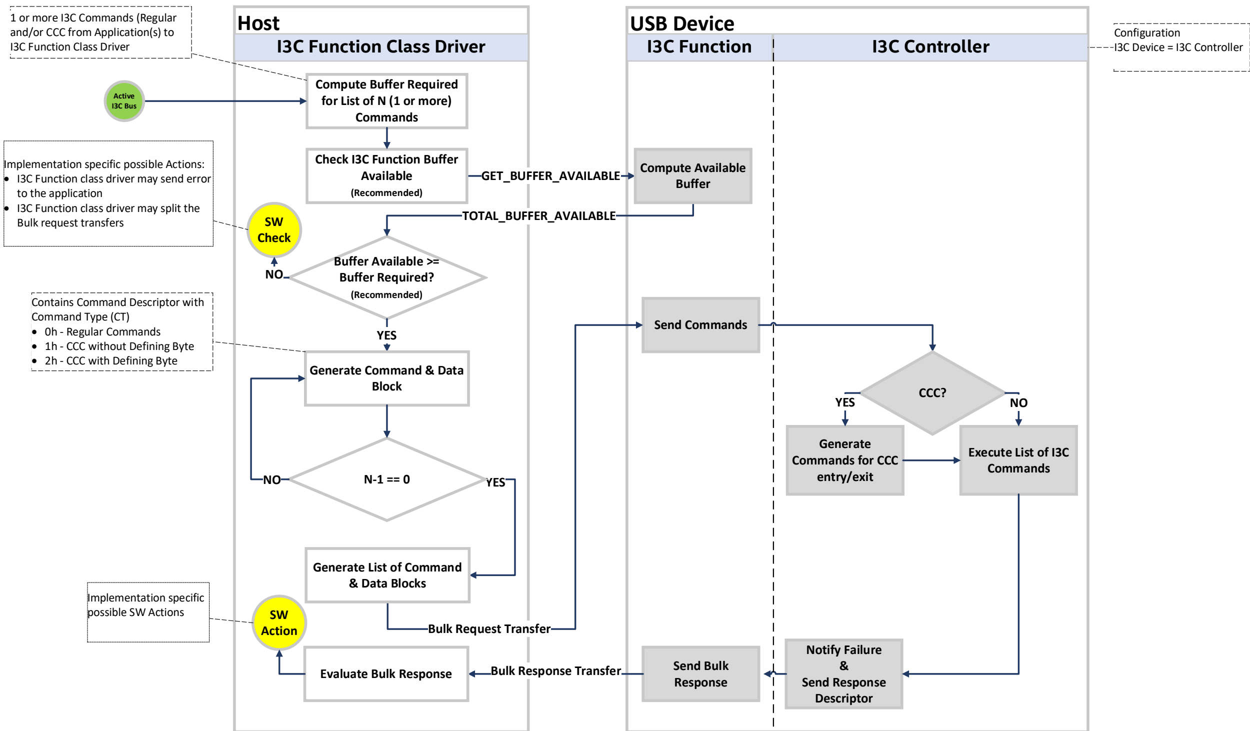


4.6.3 Regular I3C Commands and CCCs in Bulk Transfers

A Bulk request transfer may contain command(s) which are regular I3C Commands, CCCs or a mix of both (refer Section 5 of [MIPII3C]). Responsibility of handling the CCC entry and exit commands for different I3C transfer modes (refer Section 5.1.9 and 5.2 of [MIPII3C]) shall reside with the I3C Controller.

Figure 4-15 is an example illustrating a high-level flow for executing a list of commands containing CCCs.

Figure 4-15: Example - List of Commands with CCC



Note: CCC is I3C defined Common Command Code (refer[MIPII3C] Section 5.1.9 and 5.2).

4.6.4 IBI Bulk Response Transfer Flow

This section describes a high-level flow of IBI handling and IBI Bulk response transfer.

After I3C Bus is active, and after the Host retrieves the updated Target Device Table (refer Table 3-28), the Host may configure the I3C Target devices, by setting the *Max IBI Payload Size* (refer Table 3-29) through the SET_TARGET_DEVICE_CONFIG request (refer Section 3.3.2.10). The value for field *Max IBI Payload Size* may be set based on the content protocol used for the I3C Controller and I3C Target devices (refer Figure 4-16). Implementers of I3C Controller and I3C Target devices may also limit the value of *Max IBI Payload Size.*

Figure 4-16: Initialization Flow with IBI related I3C capabilities

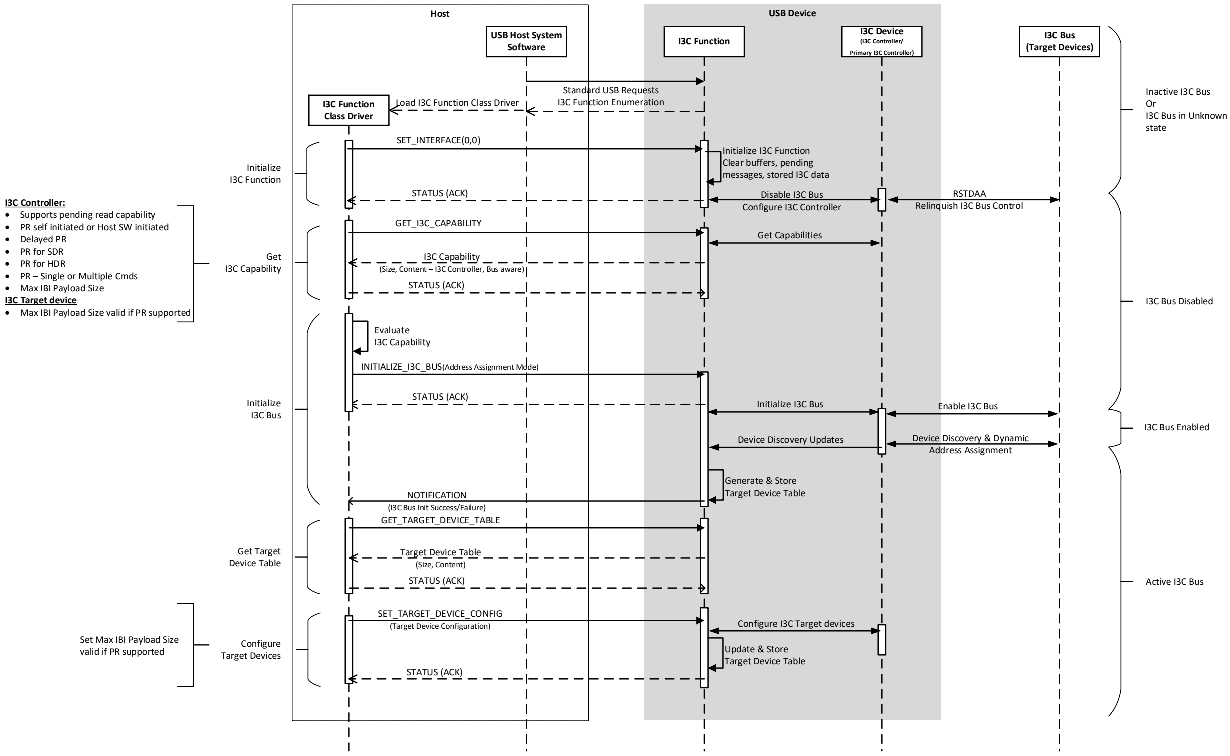


Figure 4-17 illustrates an example with a regular IBI generated by an I3C Target device. Note that this is a typical flow: not all possible flows and execution paths are described in this section. The flow is described below:

1. The I3C controller receives an IBI Request, and determines if it should acknowledge, not acknowledge, or disable the IBI without acknowledging the IBI.
2. The I3C Controller sends a notification to the I3C Function for the IBIs it does not acknowledge. In this case, the I3C Controller aborts the unacknowledged IBIs.
3. For IBIs that the I3C Controller chooses to acknowledge, the I3C Controller checks the value of BCR[2] of the I3C Target device (refer Section 5.1.1.2.1 of [MIPII3C]) to determine if the IBI has a data payload.
4. The I3C Controller reads the mandatory data byte and any additional data bytes.
5. The I3C Controller notifies I3C function of the acknowledged IBI (refer Table 3-21).
6. The I3C Controller checks the mandatory data byte to determine if the I3C Target expects a pending read for any remaining data after the IBI data payload (refer Section 5.1.6.2 of [MIPII3C]).
7. If there is no pending read, then the I3C Controller stops further reads. It sends the data to the I3C Function along with the *IBI_BULK_RESPONSE_FOOTER* (refer Table 3-35) with *Last Byte* field set to 1b, which indicates the end of IBI data payload.
8. If there is a pending read, then the I3C Controller executes the command or commands needed to read the remaining data.
    i. After this, the I3C Controller stops further reads, sends the IBI data to the I3C Function along with *IBI_BULK_RESPONSE_FOOTER* (refer Table 3-35) and *Last Byte* field set to 1b, indicating end of IBI data payload, if any of the following conditions are met:
        a. An error was encountered while reading the IBI data.
        b. The I3C Controller has read the maximum IBI Payload size worth of data from the Target device.

   c. The I3C Target device indicated end of IBI data (i.e., the I3C Controller received the last byte of IBI data).
   ii. If the none of the above conditions are met, the I3C Controller sends the data it has read along with the fields of an *IBI_BULK_RESPONSE_FOOTER* (refer Table 3-35) and *Last Byte* field set to 0b. The I3C Function then continues to read IBI data.
9. The I3C Function does the following:
   i. Generates an *IBI_BULK_RESPONSE_HEADER* (refer Table 3-35) in the Bulk-IN endpoint;
   ii. Continues to populate the Bulk-IN endpoint with the data received from the I3C Controller, provided that each data chunk and the *IBI_BULK_RESPONSE_FOOTER* can be accommodated in the endpoint;
   iii. Pads the data at the end to align it with 4 bytes; and
   iv. Appends the *IBI_BULK_RESPONSE_FOOTER* received from the I3C Controller and updates the *Bytes Valid* field in *IBI_BULK_RESPONSE_FOOTER* (refer Table 3-35).
10. If the IBI payload data is split across multiple Bulk response transfers, the I3C Function increments the *Sequence ID* field (refer Table 3-35) monotonically and repeats step 9 until all the IBI payload data is transferred.
11. When Host receives the IBI notification and determines if the IBI was acknowledged by the I3C Controller, it processes any previously pending Bulk responses before reading the IBI Bulk response. The Host continues to read the IBI bulk response until it receives the final Bulk response transfer with *Last Byte* field (refer Table 3-35) set to 1b.

For details on how an I3C Controller handles IBI (refer Section 5.1.5 and 5.1.6 of [MIPII3C]).

Figure 4-17: IBI Bulk Response Flow

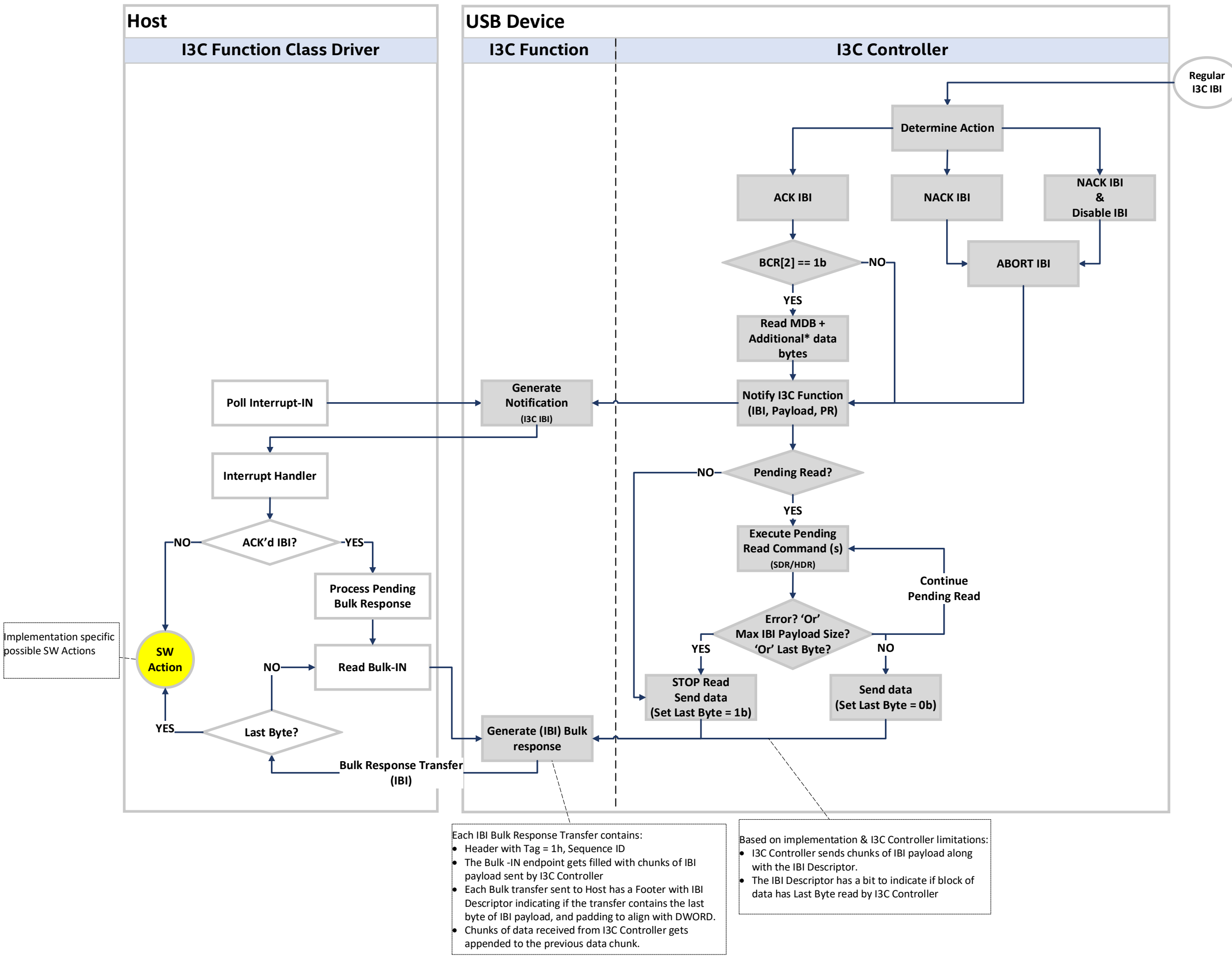
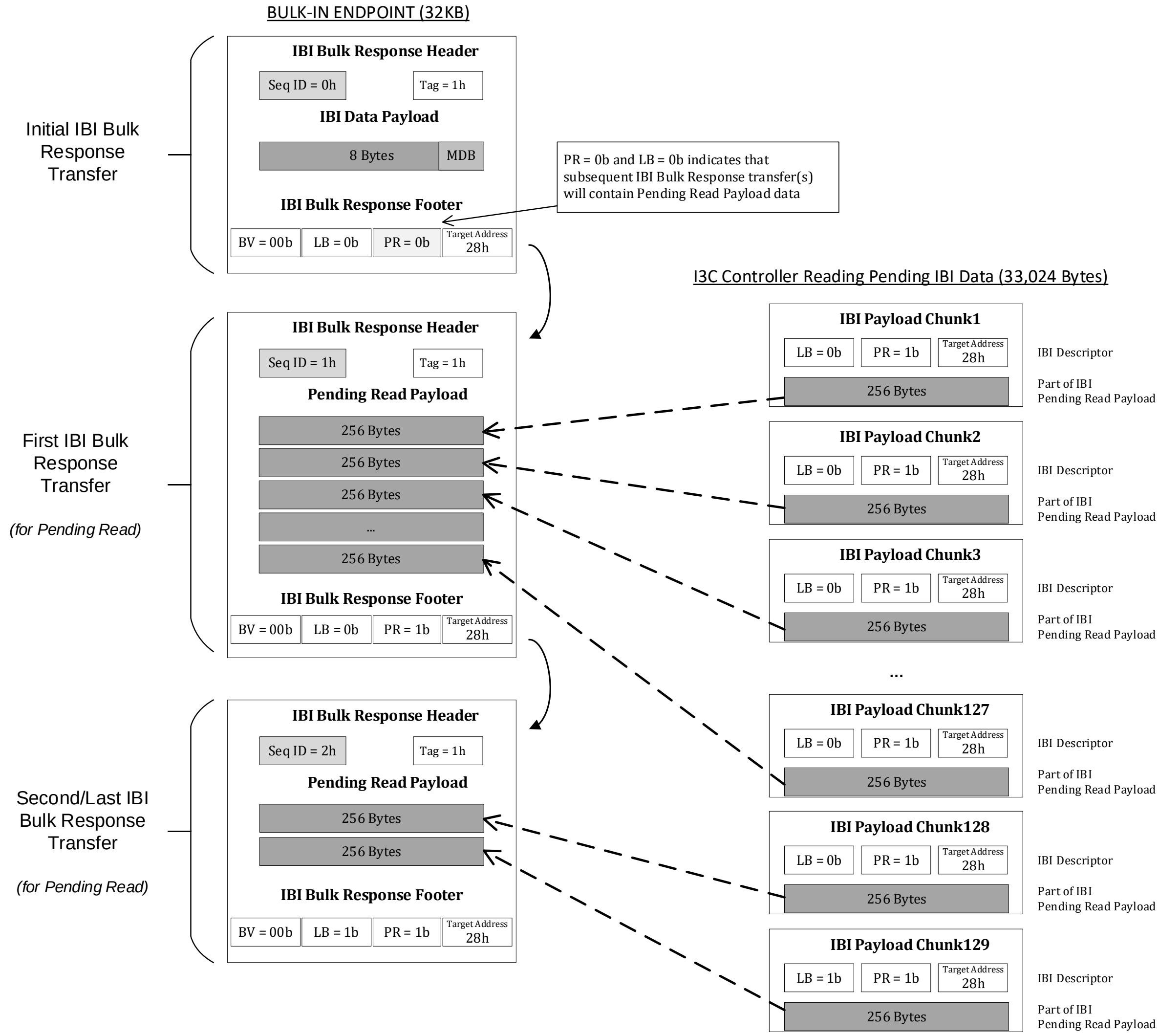


Figure 4-18 illustrates an example of an IBI Bulk response transfer where the I3C Controller is capable of sending up to 256 Bytes of data at a time to the I3C Function. In this example, the Bulk-IN endpoint size is 32KB, and IBI pending read data size is 33,024 Bytes. This example shows that the *IBI Data* (Table 3-35) is split into three Bulk response transfers with Sequence ID (SeqID) 0h, Sequence ID (SeqID) 1h and Sequence ID (SeqID) 2h (refer *Sequence ID* field in Table 3-35).

Figure 4-18: Example IBI Data and IBI Bulk Response Transfers



4.7 Queueing and Prioritization

Prioritization of interrupts from I3C Bus and queuing of messages from I3C Bus is implementation specific, and beyond the scope of this specification.

4.8 Power Management and Remote Wakeup

A USB I3C Device shall follow the requirements for Power Management and Suspended device state as described in Section 9.2.5 and Section 9.1.1.6 of [USB2.0] and [USB3.2] respectively.

The remote wake feature of the I3C Function can be enabled or disabled through the standard SET_FEATURE request as described in Section 9.4.9 of [USB2.0] and [USB3.2].

The I3C Controller may receive an I3C Bus interrupt which requires the Host's attention, such as a regular IBI from a Target device, a Hot-Join request, or an I3C Controller role request. If these interrupts occur, then the USB I3C Device shall be capable of triggering Remote Wakeup from Suspended device state, provided that the remote wake feature of the I3C Function is enabled. The remote wake capability from these events can be enabled through the SET_FEATURE request (refer

Section 3.3.2.9). The remote wake capability from these events can be disabled through the CLEAR_FEATURE request (refer Section 3.3.2.3).

A Values of Constants

A.1 USB I3C Device Class Code

A - 1: USB I3C Device Class Code

USB I3C Device Class Code	Value
USBI3CDEVICE _CLASS	3Ch

A.2 USB I3C Device Subclass Codes

A - 2: USB I3C Device Subclass Code

USB I3C Device SubClass Code	Value
USBI3CDEVICE _SUBCLASS	00h

A.3 USB I3C Device Protocol Codes

A - 3: USB I3C Device Protocol Code

USB I3C Device Protocol Code	Value
USBI3CDEVICE_PROTOCOL	00h