2-Wire Sideband Translator

Component Specification for I3C/SMBUS Forwards/Backwards Compatibility

February 2020
Revision 0.2

Intellectual Property Disclaimer

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE.

A COPYRIGHT LICENSE IS HEREBY GRANTED TO REPRODUCE AND DISTRIBUTE THIS SPECIFICATION FOR INTERNAL USE ONLY. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY OTHER INTELLECTUAL PROPERTY RIGHTS IS GRANTED OR INTENDED HEREBY.

INTEL CORPORATION AND THE AUTHORS OF THIS SPECIFICATION DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF PROPRIETARY RIGHTS, RELATING TO IMPLEMENTATION OF INFORMATION IN THIS DOCUMENT AND THE SPECIFICATION. INTEL CORPORATION AND THE AUTHORS OF THIS SPECIFICATION ALSO DO NOT WARRANT OR REPRESENT THAT SUCH IMPLEMENTATION(S) WILL NOT INFRINGE SUCH RIGHTS. ALL SUGGESTIONS OR FEEDBACK RELATED TO THIS SPECIFICATION BECOME THE PROPERTY OF INTEL CORPORATION UPON SUBMISSION.

INTEL CORPORATION MAY MAKE CHANGES TO SPECIFICATIONS, PRODUCT DESCRIPTIONS, AND PLANS AT ANY TIME, WITHOUT NOTICE.

Intel Corporation and its subsidiaries (collectively, "Intel") would like to receive input, comments, suggestions and other feedback (collectively, "Feedback") on this specification. To be considered for incorporation into the specification, Feedback must be submitted by e-mail to: 2wireTranslatorSpec@intel.com. To the extent that You provide Intel with Feedback, You grant to Intel a worldwide, non-exclusive, perpetual, irrevocable, royalty-free, fully paid, transferable license, with the right to sublicense, under Your Intellectual Property Rights, to make, use, sell, offer for sale, import, disclose, reproduce, make derivative works, distribute, or otherwise exploit Your Feedback without any accounting. As used in this paragraph, "Intellectual Property Rights" means, all worldwide copyrights, patents, trade secrets, and any other intellectual or industrial property rights, but excluding any trademarks or similar rights. By submitting Feedback, you represent that you are authorized to submit Feedback on behalf on your employer, if any, and that the Feedback is not confidential.

Notice: Implementations developed using the information provided in this specification may infringe the patent rights of various parties including the parties involved in the development of this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights (including without limitation rights under any party's patents) are granted herein.

All product names are trademarks, registered trademarks, or service marks of their respective owners

Table of Contents

1 Introduction 5

1.1 PCIe© Sideband Translator Overview 5

1.2 Reference Documents 6

1.3 Terminology 6

1.4 Use Cases 7

1.4.1 Case 1: Out of band Manageability (NVMe, etc.) 7

1.4.2 Case 2: Authentication and Securities exchange 8

1.4.3 Case 3: Retimer Configuration 8

1.4.4 Case 4: Link Subdivision (Bifurcation) 8

1.4.5 Case 5: Sideband Consolidation 8

1.5 General Architecture 8

1.5.1 Host Side Behavior 9

1.5.2 Device Side Behavior 10

1.5.3 Voltages 10

1.5.4 Signal Drivers 10

1.5.5 Bus Speed 10

1.5.6 Clocking 10

1.5.7 Bus Mastering 11

1.5.8 Reset 11

1.5.9 Initialization 11

1.5.10 Address Space 11

2 Electrical 11

2.1 Voltage differences 11

2.2 High level description 11

2.2.1 SMBUS Electrical 12

2.2.2 I3C Electrical 12

3 Operation 12

3.1 I3C Passthough 12

3.1.1 Interconnect State Machine 13

3.1.2 Port State Machine 15

4 Configuration and Status Registers 17

5 Mechanical 19

Table of Figures

Figure 1 System Configurations 6
Figure 5 NVMe Manageability Use Case 7
Figure 2 Operational Block Diagram 12
Figure 3 Interconnect State Machine 13
Figure 4 Port State Machine 15

1 Introduction

NOTE TO REVIEWERS: THIS DOCUMENT IS A DRAFT PROPOSAL INTENDED TO SUPPORT INDUSTRY DISCUSSION REGARDING HOW WE MIGRATE FROM OUR EXISTING IO FORWARD. IT IS IMPORTANT TO NOTE THAT ALL CONTENT IS SUBJECT TO CHANGE. FEEDBACK IS WELCOME AT (2wireTranslatorSpec@intel.com).

Due to the nature of the high speed IO industry, we expect the current protocol configuration will evolve. We will continue to support translation past the current SMBUS/I3C configuration.

1.1 PCIe© Sideband Translator Overview

Sideband IO is used in high speed communication systems for out of band (OOB) applications such as bus management, device discovery, configuration, debug, telemetry, etc. Sideband operation allows critical communication outside the domain of the high speed IO owner and works even if the high speed path is in low power state or not functional.

High speed IO continues to evolve to faster speeds while shrinking silicon geometries require lower votages. The next generation devices need considerably more bandwidth than the 2-wire interface can currently provide and using lower voltages to avoid level shifters.

Backwards compatibility is essential to ease adoption into many existing hardware platforms. This specification aims to define a 2-wire sideband translater to accomplish these goals. Initially it would be a separate bridge device and over time it could be integrated into devices on both sides of the 2-wire interface or the legacy use case may also be deprecated.

PCIe© has used the SMBUS protocol since 1995 for sideband commmunication. Current PCIe© devices typically implement 3.3V SMBUS at speeds of 100kHz and sometimes 400kHz. I3C offers a 100x improvement on speed as well as lower operating voltages with backwards compatibility.

The 2-wire protocol translator operates in different configurations as shown in **Figure 1** below. The “New” configuration in the figure below operates at I3C speeds and voltages while all else operate at SMBus speeds and voltages.

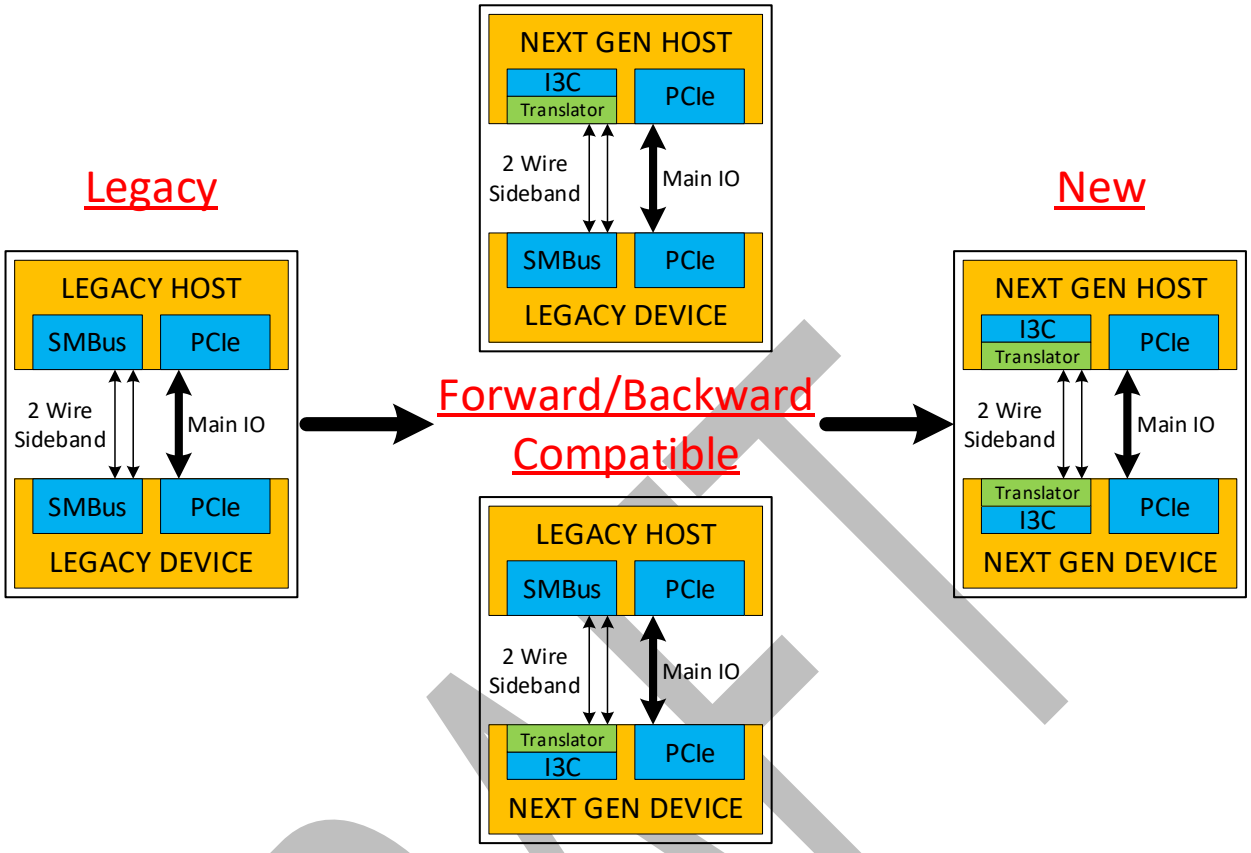

Figure 1 System Configurations

1.2 Reference Documents

Document	Location
PCI Express Base Specification, Revision 5.0	https://members.pcisig.com/wg/PCI-SIG/document/13005
System Management Bus (SMBus) Specification, v3.1	http://smbus.org/specs/smbus31.pdf
MIPI I3C Specification v1.0	https://members.mipi.org/wg/All-Members/document/71249
MIPI I3C Basic Specification v1.0	http://resources.mipi.org/mipi-i3c-basic-v1-download

1.3 Terminology

Selected terms and acronyms used in this specification are defined below.

I^2C	Inter Integrated Circuit
I3C	Improved Inter Integrated Circuit
TWI	Two-Wire Interface
SMBUS	System Management Bus

The terms "*shall*" and "*must*" are used to describe mandatory requirements. The terms "*should*" and "*may*" are used to identify optional requirements.

1.4 Use Cases

The below sections outline common use cases that the translator component may enable on the PCIe© interface.

1.4.1 Case 1: Out of band Manageability (NVMe, etc.)

A common use case for the translator component on the PCIe© interface is for out-of-band manageability. **Figure 2Error! Reference source not found.** shows an example of a NVMe SSD connected by a PCIe© Connector to a Host Processor and Management Controller. For Compatibility the SSD contains a translator to talk to I3C or SMBus equipped hosts. Likewise, the Management Controller also contains a translator to talk to I3C or SMBus equipped peripherals. If both sides support I3C then I3C speeds and lower voltages are achieved. This interface is used to discover the type of device inserted, configure it, detect current state, and retrieve telemetry data. Faster speeds have a big impact on firmware update and telemetry download times.

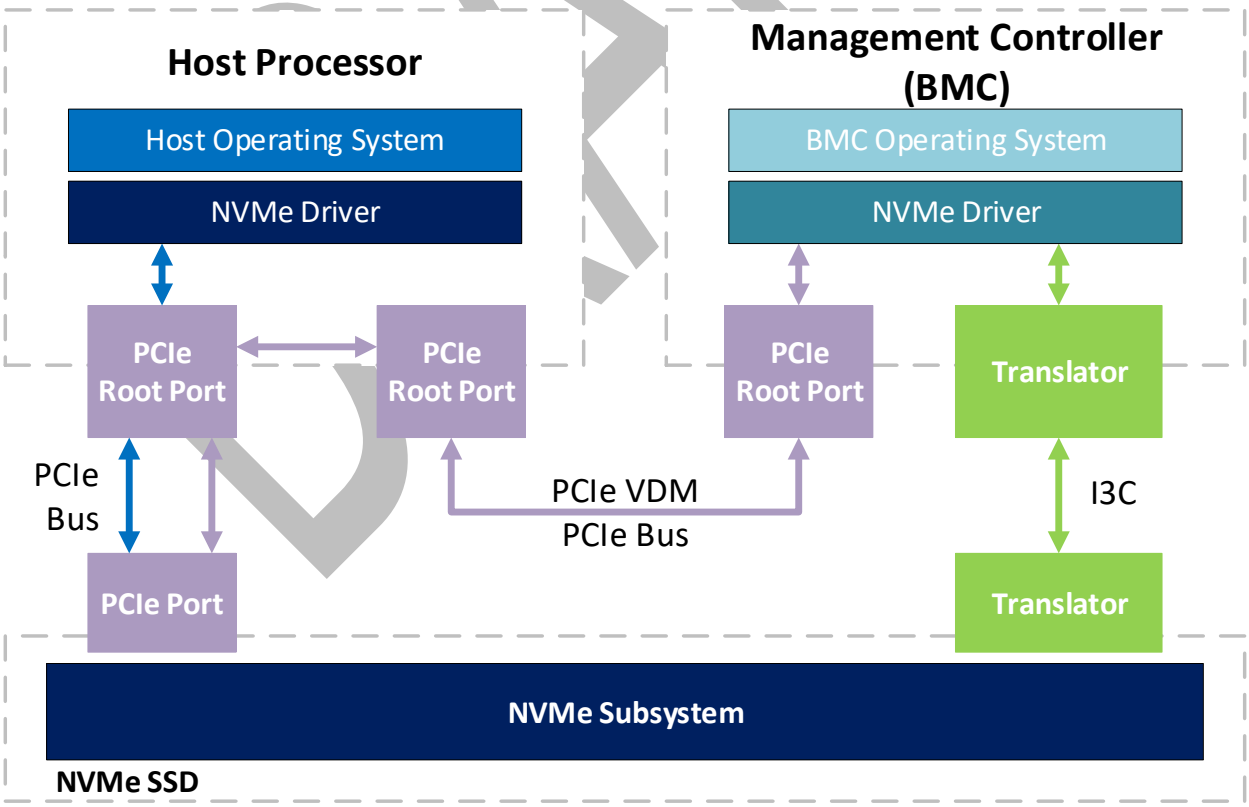

Figure 2 NVMe Manageability Use Case

1.4.2 Case 2: Authentication and Security exchange

Security and Authentication continue to be important topics for interconnects. As computer systems continue to develop, the number of components that require security and authentication increases. System security from local and remote attackers must be present in modern systems. In addition, authentication of components is a highly sought after feature that prevents malicious devices from replacing the original component. The translator component will allow communication of various securities and authentication messages to be sent between devices. The translator component works to interpret between the differing protocols. Faster bus speeds can reduce power on time.

1.4.3 Case 3: Retimer Configuration

PCIe© Systems commonly use retimers to deliver signal quality over extended distances. The configuration of these retimers is the responsibility of the PCIe© system. To keep the configuration traffic from consuming the PCIe© main bus' bandwidth, the system can use the sideband bus to configure retimers. The translator component ensures that proper retimer configuration is completed in new or legacy configurations. Faster bus speeds can reduce power on time and enable quicker low power state transitions for power savings.

1.4.4 Case 4: Link Subdivision (Bifurcation)

Link Subdivision, also commonly known as Bifurcation, is a common IO feature that allows a system to divide a single communication channel into multiple smaller channels that operate independently. In order for link subdivision to work properly, components on either side of the communicaiton channel can use the sideband bus to configure the channels to their agreed upon number and size. By using the sideband bus, enumeration time and main band bandwidth can be saved. An EEPROM in the translator enables auto discovery of backplane, riser cards, carrier devices and fanout cables.

1.4.5 Case 5: Sideband Consolidation

High speed interconnects commonly operate with one or more sideband bus'. Sideband bus' use many different protocols and can operate with one or many wires. Each wire takes up valuable pin count on a component. The translator component allows more dynamic sideband protocols to be used without breaking backwards compatibility. These dynamic, scalable sidebands allow for consolidation of sideband bus', saving pin count for high speed interconnects.

1.5 General Architecture

The 2 wire translator is designed to work at either the host or device end of the PCIe© connection. By using the same silicon on either side of the connection the scale ramps faster to reduce the initial cost until the volume is high enough to justify specialized devices. A pin called

Mode would be tied low for use in the host and tied high for use in the Device. Their unique and common features are outlined in the sections below:

This specification is for a translator to overcome the 2 limitations of I3C. The first is voltage conversion, there are currently no protocol aware level shifters for I3C and that would be a requirement to support legacy devices at higher voltages. The second is clock stretching, I3C does not tolerate SMBus devices that stretch the clock for flow control.

In addition, the translator would also include features to dynamically mux multiple channels where some could be detected as legacy SMBus and others could be I3C passthrough. The channels running in SMBus mode would cache and convert I3C traffic to the slower SMBus speeds for background communications without slowing down the rest of the I3C network.

1.5.1 Host Side Behavior

The host typically has a management controller that connects to multiple devices through a mux. The management controller would connect to the translator using I3C as a (leader/initiator). The translator have an I3C (follower/target) port listening to the management controller. The (follower/target) port would auto detect the voltage from the host to know if the host was using the lower voltage I3C or a higher voltage SMBus. In either case, the (follower/target) port would expose multiple registers that the management controller would use to configure the translator.

The other side of the translator would have multiple ports that are each connected to PCIe© devices in any of the PCIe© form factors. These devices may be any combination of legacy SMBus or I3C. The translator shall support hot plug and surprise removal. After reset or power on these ports would assume that the attached devices were I3C capable using low voltage. The host would detect if a device was plugged in and then communicate through the translator to discover if the attached device was I3C capable. If not, it was fall back to the higher voltage SMBus mode for that port and enable the internal pullup resistors.

In SMBus mode the translator would accept a command from the host into a buffer. Then the host would forward the command to the attached device. (follower/target) Responses or (leader/initiator) Mode transmissions from the attached device would be buffered in the translator. Upon a stop condition the translator would set a status register and assert the interrupt signal. If the host was connected by I3C instead of SMBus then the translator would also issue an In-Band Interrupt (IBI) so that the interrupt signal did not need to be connected.

A 2k Serial EEPROM in the translator would be accessible through the (follower/target) port facing the host. It would be read by the host after reset to determine what type of motherboard, riser card, backplane, or cable contained the translator. It would also indicate characteristics such as max speed and cable lengths. The format of the data in the Serial EEPROM would follow the IPMI MultiRecord extensions developed by NVMe-MI 1.1 specification for the Element type.

1.5.2 Device Side Behavior

The device would use the same silicon but be configured slightly differantly after reset if the Mode pin was tied low. The host port would still be connected towards the host or external port of the PCIe© form factor. The other ports would be connected to one or more devices inside the PCIe© device which may also be a combination of SMBus and I3C devices. The internal Serial EEPROM would also be readable over the host connection to determine what type of device was plugged in using the same data format as in the host usage.

The most significant difference from when the translator is used on the host side is that the register access and Serial EEPROM programming is initially only permitted from the first downstream port. If the high voltage on the Mode pin is at SMBus levels then the downstream port is configured for SMBus. If the high voltage is at the lower I3C levels then the downstream port is left in the I3C mode. The device thus can configure the translator or program its Serial EEPROM without traffic getting echoed through to the host.

1.5.3 Voltages

The range of operating voltages can vary greatly from one IO technology to the next. To be able to interoperate, voltage differences must be resolved by this component. Additionally, as silicon processes advance the need for lower voltage systems becomes essential. Legacy IO technologies can reach operating voltages up to 5V. Newer IO technologies are now targeting lower voltages closer to 1V.

1.5.4 Signal Drivers

Differing Tx drivers can cause complications on a mixed bus. SMBus uses an open collector transmitter with voltage converters that simply clamp voltage low and can waste power. Newer technologies like I3C use Push-Pull mode with only specific periods of time time in the message where it tolerates an open collector (follower/target) driving the data line low. This component must be protocol aware to know when collisions are safe.

1.5.5 Bus Speed

IO Bus speed is another characteristic that can vary greatly between technologies. As this component addresses 2-wire interfaces, it will interact with low speed IO that operate within the kB/s and MB/s range. Even with the lower speed buses, even small differences in bus speed can cause communication issues if not dealt with properly.

1.5.6 Clocking

Clock synchronization must always be considered when interoperating with two different technologies, but there are other aspects in clocking that must be considered as well. As an example, SMBUS and I^2C use a technique called clock stretching that other IO technologies may

not use. In order to guaruntee interoperability between technologies, clock stretching and any other similar techniques must be handled correctly.

1.5.7 Bus Mastering

On a Multi-drop bus there is the role of a bus (leader/initiator) that is capable of controlling the bus. Whether there is only one (leader/initiator) or multiple (leader/initiator)s is technology specific, but differences must be resolved in order to ensure compatibility.

1.5.8 Reset

Reset can be accomplished either inband or out of band. In addition, technologies can have different patterns that are designated for the reset signaling. The translator component must be protocol aware to be able to detect any resets and be able to convey them reliably to a different IO technology.

1.5.9 Initialization

Initializtion is the important process that devices must go through to establish the initial configuration of the devices. Procedures such as address assignment, device role and operation mode are typcailly established in this process. The translator component must be able to operate under the differing initialization process and be albe to translate between technologies to ensure that the initail configuration across devices is correct and interoperable.

1.5.10 Address Space

This device contains its own address space for configuration and operation settings and also must interact with other devices' address spaces and be able to understand it completely. Communication and interaction with other address spaces will ensure thorough interoperability.

2 Electrical

There are fundamental differences in the way 2-wire IO Technologies operate electrically. As such, it is important to understand these differences when implementing the translator component. The differences in electrical specification are described in the sections below.

2.1 Voltage differences

In order to handle the differing electrical specifications between technologies, the translator will use level shifters and possibly other components to translate accordingly.

2.2 High level description

The translator component will operate as a point-to-point topology despite whether the IO technology is primarily multi-drop or point-to-point itself. Since it will commonly be used over

a connector, the point-to-point functionality is useful for simplifing the component and it's operation.

2.2.1 SMBUS Electrical

SMBUS's two physical wires are are used as clock and bidirectional data. Supply voltage for SMBUS is up to the designer as long as the chosen voltage falls within the specified range (See **Error! Reference source not found.**). Since different supply voltages can be on the bus, all device signals will be pulled up to the highest voltage on the bus. Therefore, each SMBUS device must operate at voltages up to the maximum allowed voltage (5.5V).

2.2.2 I3C Electrical

I3C's two physical wires are are used as clock and bidirectional data. The I3C bus operates in two different modes: Push-Pull Mode and Open Drain Mode. The target voltages for I3C are 1.2V, 1.8V and 3.3V; the interface is not designed to support 5V, but could support if sufficient changes are made to electrical parameters such as Driver strength, data speed, etc. Capacitance loading for the interface at top speeds is much lower than Legacy I^2C (reduced to 50pF). Higher capacitance is possible with reduced parameters like speed and features (i.e. no Legacy I^2C on the bus). There is also optional support in I3C Basic for 1.0V operation at 100pF loading.

3 Operation

The following sections describe the operation of the translator component. **Figure 3** shows an example block diagram of BMC MUX connected to a Peripheral interface using the translator component and a PCIe© interconnect.

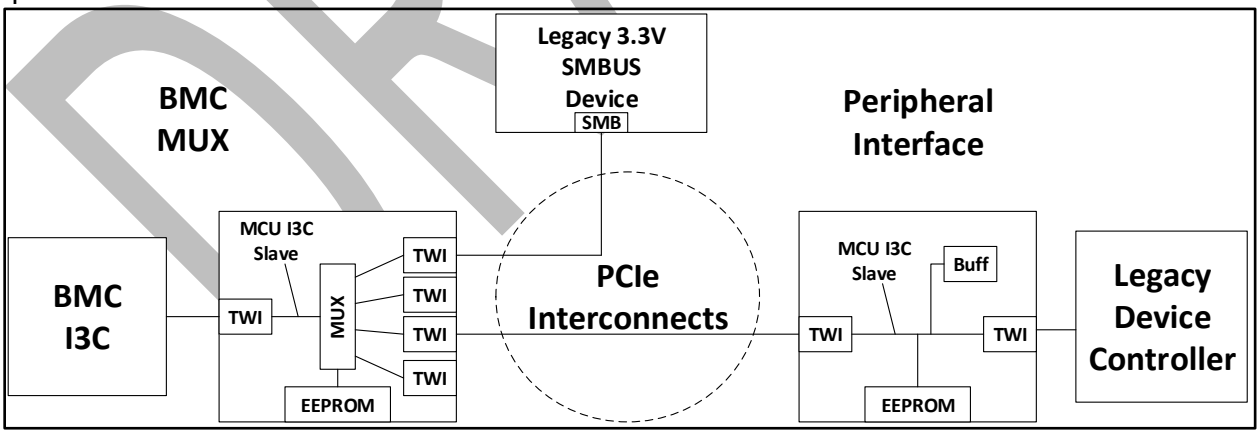

Figure 3 Operational Block Diagram

3.1 I3C Passthough

The translator component operates using two separate state machines. One state machine is the Interconnect state machine which is responsible for sniffing the data and is seen in **Error!**

Reference source not found.. The second state machine is the Port state machine which sets each ports operating state as far as (leader/initiator)/(follower/target) and Tx/Rx as shown in **Figure 5Error! Reference source not found.**. Details of the state machines and their individual states is described in the following sections.

3.1.1 Interconnect State Machine

The interconnect state machine is designed to analyze bus traffic and formulate the raw data stream into bytes and addresses. The state machine can be seen below in **Error! Reference source not found.**.

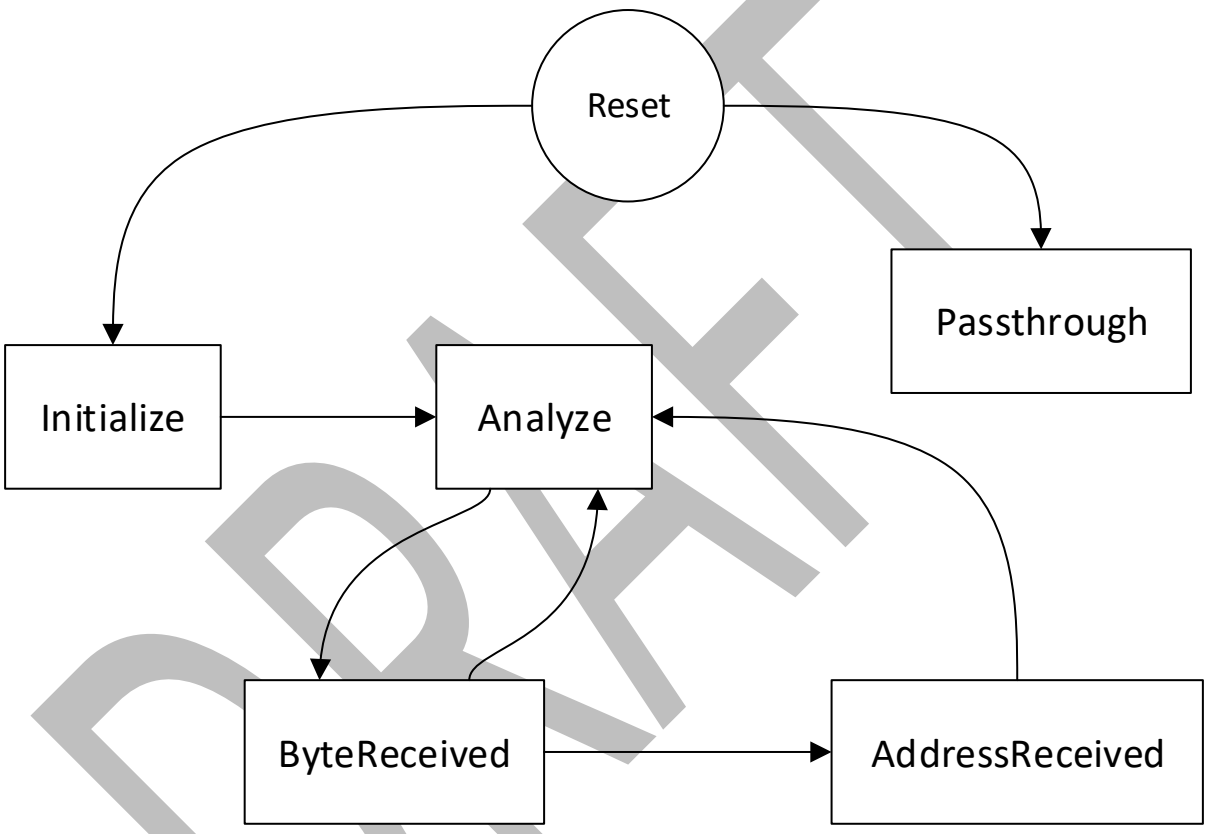

Figure 4 Interconnect State Machine

3.1.1.1 Reset State

3.1.1.2 Initialize State

The Intitialize state is entered on Power up, Reset or Stop condition to initialize registers to their default values. This state is a transitory state that automatically exits into the Analyze State when complete.

3.1.1.3 Analyze State

The Analyze state is entered from the Initialize state automatically or from the ByteReceived state if address reception is complete. While in this state the component watches traffic

between ports. It detects the read/write direction for the data patch and sets a flag during the Tbit. After a complete byte is received, the state machine exits this state and proceeds to the ByteReceived state.

3.1.1.4 ByteReceived State

The ByteReceived state is entered from the Analyze state after a full byte is received. While in this state the component updates the registers for the byte received. Once this state is complete it will enter into the AddressReceived state if address reception is still in progress. Otherwise, it returns to the Analyze state.

3.1.1.5 AddressReceived State

The AddressReceived state is entered from the ByteReceived state. This state detects the end of address reception then automatically returns back to the Analyze state.

3.1.1.6 Passthrough State

The Passthrough state is entered from Power Up, Reset or Stop condition to indicate that Data and Clock are the logical AND of the individual ports. This state is used when the protocols are the same on upstream and downstream side of the component and data is meant to be sent straight through the component without any translation.

3.1.2 Port State Machine

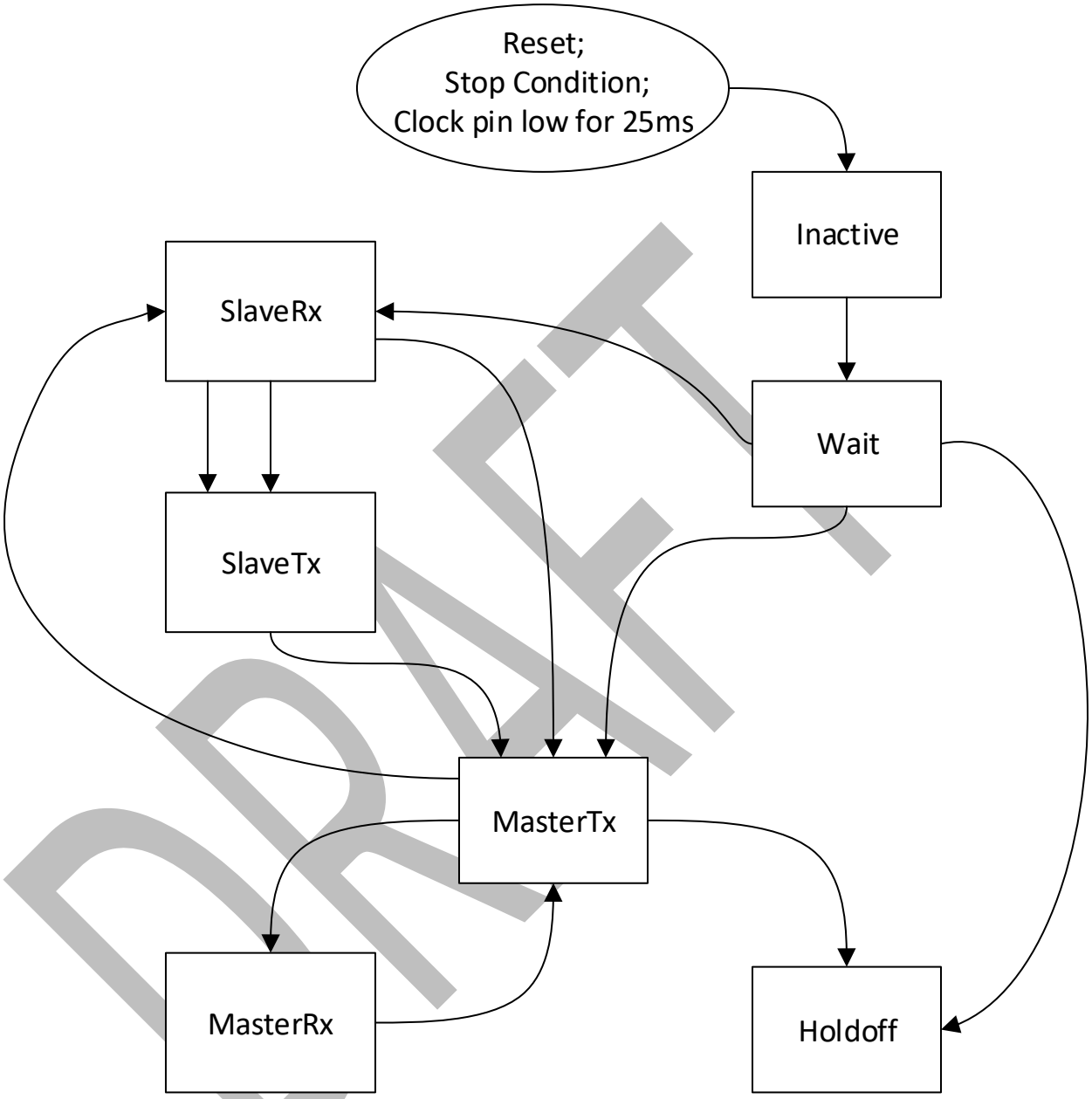

Figure 5 Port State Machine

3.1.2.1 Inactive State

The Inactive State is entered upon Reset, Stop Condition or Clock pin low for 25ms. This state ensures a clean Stop occures for prior transactions and puts the IO pins into input at low Vih. If both Clock and Data pins go high then this state is exited and the state machine proceeds to Wait State.

3.1.2.2 Wait State

The Wait state is entered from the Inactive State when Clock and Data are high. This state occurs between transactions on the bus. During this state the Vih is sampled for autodetection of signal levels on Ports that enable auto detect. A start condition causes the state machine to transition to the (follower/target)Rx state. If a different port enters (follower/target)Rx state then this port's state machine will transition to (follower/target)Tx state. If a Port enters the Wait state during a transaction, then it will automatically transition to the Holdoff state.

3.1.2.3 (follower/target)Rx State

The (follower/target)Rx state is entered from the Wait state when a Start Condition is received or it is entered from the (follower/target)Tx state when a port loses arbitration. In this state the port is passing external clock and data to the internal colck and data with the data direction flipped during the Tbit. This typically means that the (leader/initiator) device is connected to the port and sending data to another device through the translator. If the transaction becomes a (leader/initiator) read, then the state machine transitions to the (follower/target)Tx state. If a device is attached to another port and wins (leader/initiator), then this port transitions to the (leader/initiator)Tx state to put back pressure against the original (leader/initiator).

3.1.2.4 (follower/target)Tx State

The (follower/target)Tx state is entered from the (follower/target)Rx state when a transaction becomes a (leader/initiator) read. In this state the port is receiving clock and driving external data, but the data direction is flipped during the Tbit. Typically in this state the (leader/initiator) device is connected to this port and reading data from another device through the translator. The Port transitions to the (follower/target)Rx state when Data output high loses to an external low, indicating a device close to the (leader/initiator) won arbitration. The Port will transition to the (leader/initiator)Tx State if another port takes (follower/target) role.

3.1.2.5 (leader/initiator)Tx State

The (leader/initiator)Tx state is entered from the (follower/target)Tx, (follower/target)Rx, (leader/initiator)Rx or the Wait states. In this state the port is driving the internal clock and data out through this port but the data direction is flipped during the Tbit. Typically in this state the (leader/initiator) device is connected to a different port and writing to a (follower/target) device connected on this port. If Clock and Data output high loses to an external low then this port transitions to the (follower/target)Rx state because a competing device externally attaced to this port won arbitration. If a transaction becomes a read then this port transitions the the (leader/initiator)Rx state. If there is no ACK to the address then this port transitions to the Holdoff state.

3.1.2.6 (leader/initiator)Rx State

The (leader/initiator)Rx state is entered from the (leader/initiator)Tx state when a transaction is a read. In this state the port is driving the intenal Clock out and receiving the external Data but the data direction is flipped during the Tbit. Typically in this state the

(leader/initiator) device is connected to a different port and reading from a (follower/target) device connected to this port. If external Data high does not match internal Data the an internal collision has occurred and this port transitions to the (leader/initiator)Tx state to force an external collision.

3.1.2.7 Holdoff State

The Holdoff state is entered from the (leader/initiator)Tx state if an address is not ACKed. In this state the port is forcing Data and Clock low to prevent any external device attached to this port from attempting to start a transaction. This state is held until the transaction is completed with a maximum time of 25ms.

4 Configuration and Status Registers

This section will describe how any registers are used in cooperation with the 2-wire translator component.

Reg Name	Read/ Write	Reg Address	Reset Value	Description
Connections	R/W	00h	00h	Each bit represents one of the 8 downstream ports, any combination of bits may be set connected to the corresponding port to the current I3C (leader/initiator) Port. Some hubs may have less than 8 ports, the hub's I3C (follower/target) is always connected.
Interrupt Status	R/W	01h	00h	Each bit represents a pending interrupt from one of up to 8 downstream ports
Interrupt Mask	R/W	02h	00h	Each bit represents one of up to 8 downstream ports, if bit is set then interrupts from that port are passed through to the current I3C (leader/initiator) else they are blocked.
Temperature	R/W	03h	80h	Optional temperature of mux and potentially max temperature of writes from downstream devices. A write from the current (leader/initiator) resets
Health	R/W	04h	00h	Health status of mux and ORed with writes from downstream devices. A write from the current (leader/initiator) can clear status Bit 7 – Device health issue – no longer working Bit 6 – Device health issue – reduced functionality Bit 5 – Device health issue – warning, no use impact Bit 4 – Thermals out of max range Bit 3 – Thermals out of safe range Bit 2 – Power delivery issues Bit 1 – PCIe© link issues Bit 0 – I3C/SMBus link issues

Hardware	R	05h	Impl Spec	Bit 7 – Interrupts supported if 1 Bit 6:3 – Reserved Bits 2:0 – The number of downstream ports
Current Port	R	06h	0Fh	Bits 7:5 – Reserved Bit 4 – External selection input signal Bits 3:0 – The port number of the ClockSource
Permissions	R/W	07h	00h	Bits 7:6 – Reserved Bits 5:4 – 0 if any port can write the critical control registers, 1 if only the normal (leader/initiator) can, 2 if only the alternate (leader/initiator), 3 if only normal or Alternate (leader/initiator) can Bits 3:2 – 0 if any port can write the rest of these registers, 1 if only the normal (leader/initiator) can, 2 if only the alternate (leader/initiator), 3 if only normal or Alternate (leader/initiator) can Bits 1:0 – 0 if any port can read these registers and write to Temperature and Health, 1 if only the normal (leader/initiator) can, 2 if only the alternate (leader/initiator), 3 if only normal or Alternate (leader/initiator) can
Alternate (leader/initiator)	R/W	08h	Impl Spec	Bit 7 – Alternate (leader/initiator) hardware represent if set (read only) Bit 6 – Tolerates clock stretching if ClockSource when set to 1 Bit 5:4 – Voltage (Auto Detect=0, 1.0v=1, Vcc=2, rsvd=3) Bits 3:0 – The port number for the alternate (leader/initiator), ports 0-7 are the downstream ports, 8 is the normal (leader/initiator), 9 is the extra (leader/initiator) if hardware present, 10 is the extra (leader/initiator) with internal pull up if present (defaults to 1.0V if auto detect voltage)
Default (leader/initiator)	R/W	09h	80h	Bit 7 – Connected to alternate (leader/initiator) if set, this setting connects default & alternate (leader/initiator)s to hear each other in addition to Connections ports Bit 6 – Tolerates clock stretching if ClockSource when set to 1 Bit 5:4 – Voltage (Auto Detect=0, 1.0v=1, Vcc=2, rsvd=3) Bit 3 – Use internal pull up if set to 1, defaults to 1.0v if auto detect voltage Bits 2:0 – Reserved
Port 0	R/W	10h		Bit 7 – Reserved Bit 6 – Tolerates clock stretching if ClockSource when set to 1 Bit 5:4 – Voltage (Auto Detect=0, 1.0v=1, Vcc=2, rsvd=3) Bit 3 – Use internal pull up if set to 1, defaults to 1.0V if auto detect voltage Bits 2:0 – Reserved (may indicate max speed)
…		…		

Port 7	R/W	17h		Bit 7 – Reserved Bit 6 – Tolerates clock stretching if ClockSource when set to 1 Bit 5:4 – Voltage (Auto Detect=0, 1.0v=1, Vcc=2, rsvd=3) Bit 3 – Use internal pull up if set to 1, defaults to 1.0V if auto detect voltage Bits 2:0 – Reserved (may indicate max speed)

5 Mechanical

We are considering a package footprint

It is possible to integrate the translator device into an existing component such as an EEPROM, feedback welcome.