

STORAGE DEVELOPER CONFERENCE



*BY Developers FOR Developers*

# Overcoming SMBus Limitations with I3C

Presented by:

Janusz Jurski (Intel), Myron Loewen (Solidigm),  
Anthony Constantine (Intel), Juan Orozco (Intel)

Co-authors:

Bryan Kelly (Microsoft), Zbigniew Lukwinski (Intel)

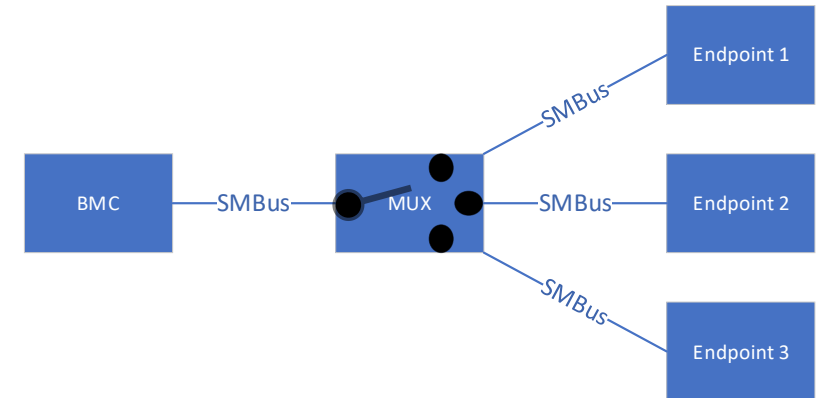
# Agenda

- SMBus Limitations
- I3C-based Solution
- Experimentation
- Summary & Call to Action

# SMBus Limitations

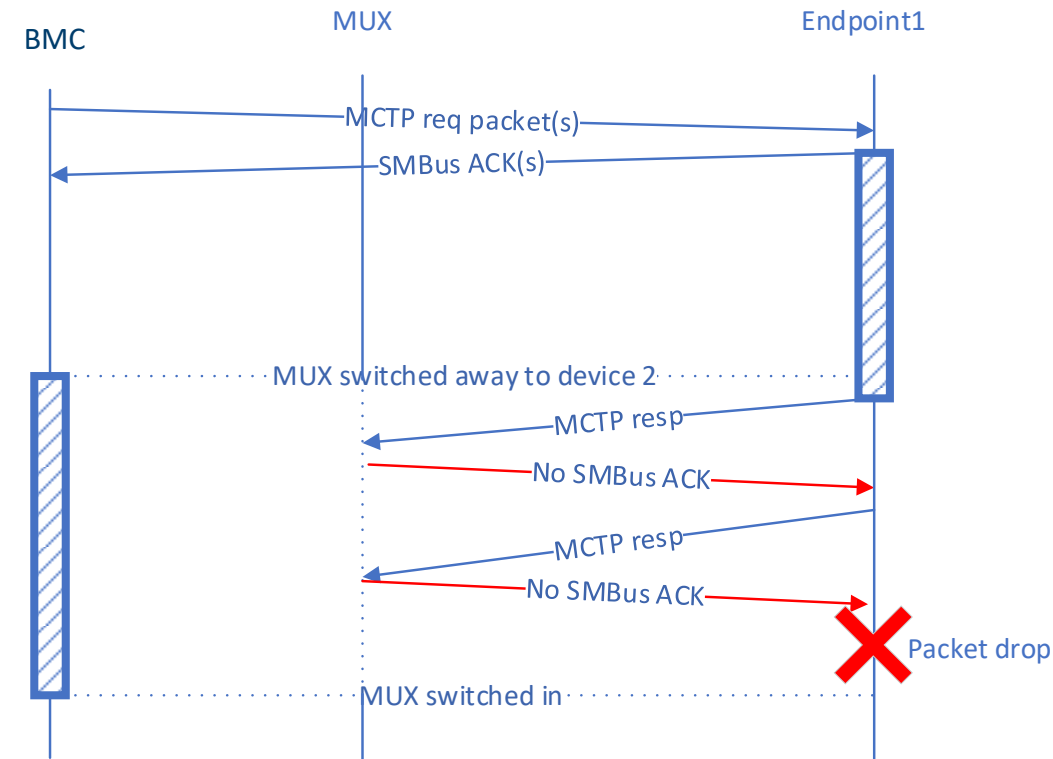
# SMBus/I2C Sideband Management Overview

- [SMBus](#)/I2C sideband interface used by all PCIe/CXL form factors, incl. storage
- No common I2C/SMBus addressing architecture
  - ARP expected by CEM Spec but often not implemented (ARP optional in SMBus spec)
  - system vendors maintain address databases to avoid collisions
  - vendor-dependent proprietary solutions used instead, typically involving I2C/SMBus MUX
- Security expectations drive MUX-based architectures
  - prevents peer-to-peer communications
- Legacy use cases
  - simple communication driven by only BMC with quick responses by endpoints (FRU read, temp sensor, etc.)
  - low bandwidth sufficient (typically 100kHz)

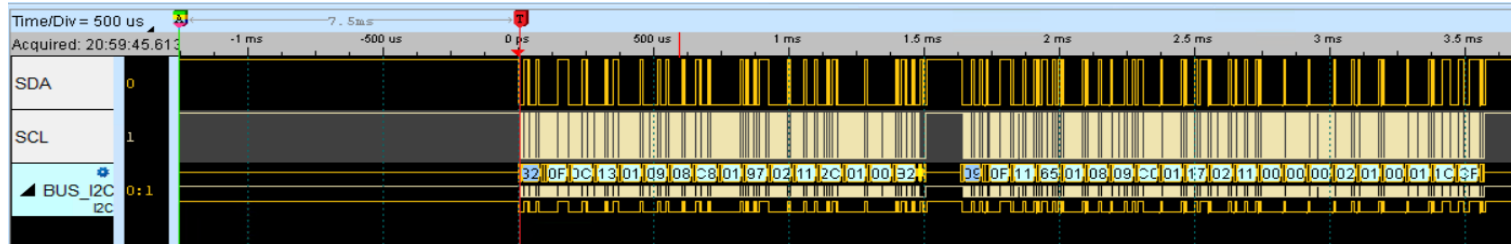


# MCTP over SMBus w/MUX & Advanced Use Cases Don't Work

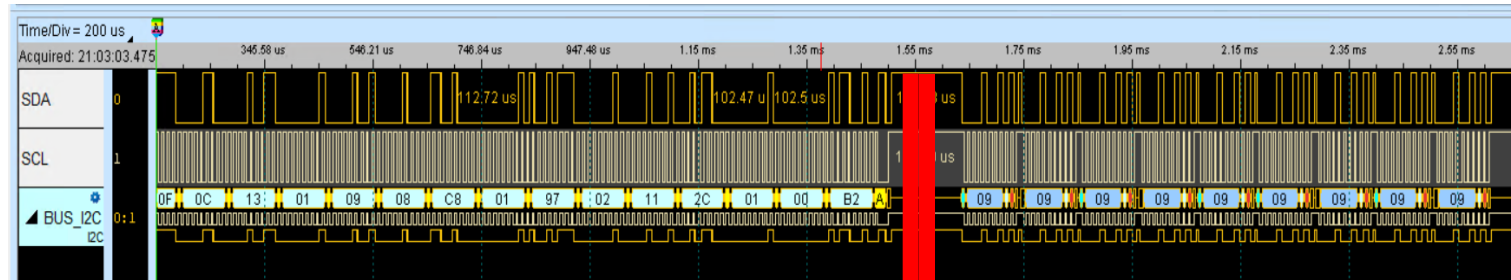
- [MCTP over SMBus](#) expects continuous SMBus
  - expectation not valid in systems with MUX
  - SMBus arbitration does not work over MUX-based I2C/SMBus
- Endpoints try to transmit when MUX switched away
  - following the specifications, due to no ACK, they retry and drop packets after a couple of milliseconds
- MCTP over SMBus with MUX generally unusable for:
  - long-running tasks (SPDM, etc.)
  - asynchronous communication (alerts, events, notifications, etc.)
  - streaming from endpoint (telemetry, etc.)
  - large MCTP messages fragmented into many MCTP packets (frequent packet losses in these scenarios)
- MUX switching during ongoing transmission
  - truncated transactions interpreted/consumed with unpredictable consequences
  - many SMBus devices hang due to glitches



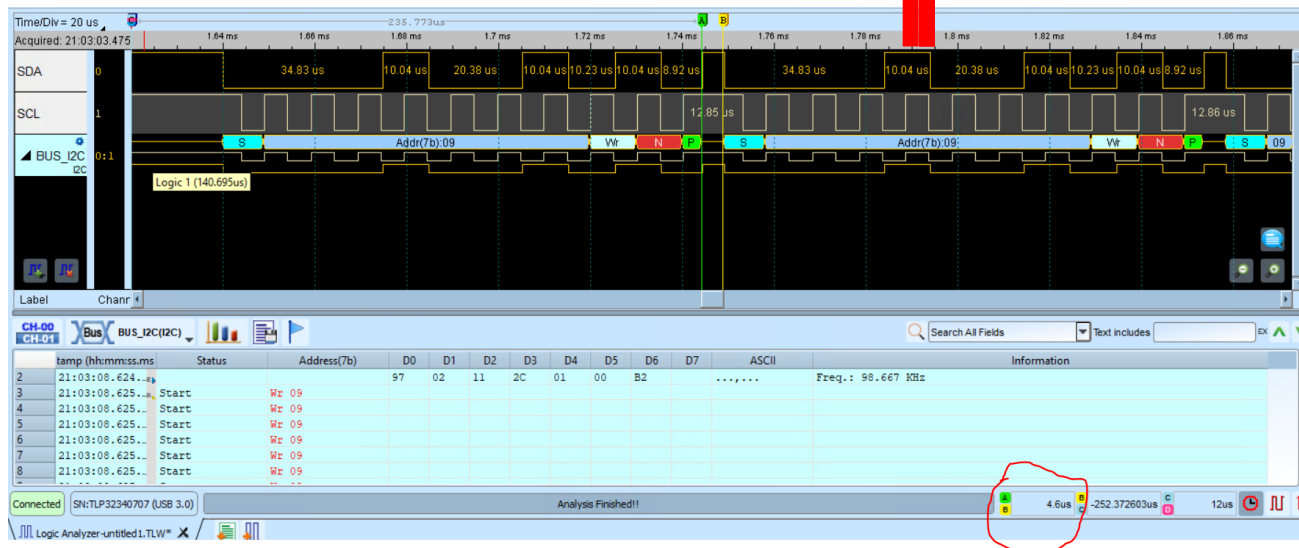
# Packet Losses with Typical SMBus MUX Configuration



1. Sample successful request-response sequence when SMBus MUX not switched away



2. Failure when SMBus MUX switched away to another device



2a. Retry every 4.6us (no ACK when SMBus MUX switched away)

# Workarounds Today

- Long running tasks or large MCTP messages:
  - BMC waiting idle for endpoint to process the request and respond
    - 100s of milliseconds wasted with every transaction
  - proprietary or higher-protocol control commands pause/resume device responses
    - NVMe-MI standardizes this approach to some extent (with NVMe-MI-specific assumptions)
  - retries (usually don't work)
    - no way for the device to be aware of the MUX being switched away
    - retries repeat same sequence with same result
  - every vendor is different
- No workaround for truly asynchronous communication

# I3C-based Solution



# Industry Landscape with I3C

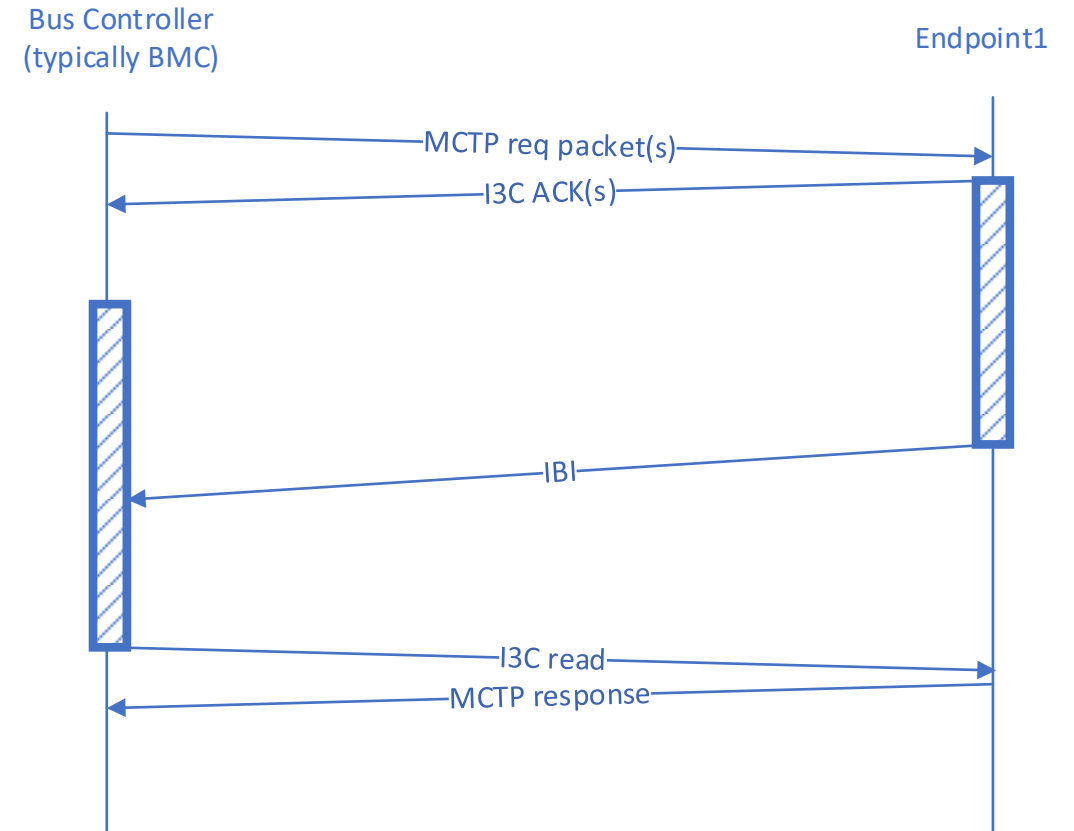
- [MIPI I3C Basic](#) – natural upgrade to address SMBus/I2C limitations
  - Upgrade details defined in EDSFF specification ([SNIA SFF-TA-1009 revision 3.1, published January 6<sup>th</sup>, 2023](#))
  - Expecting other form factors to follow EDSFF solution
    - SNIA is donating EDSFF I3C solution to other standards organizations to keep specs aligned
    - PCIe [Architectural Out-of-Band Management](#) under review
- [MCTP I3C Binding](#) defined by DMTF
- Common I3C HUB specification (Intel RDC #766079) with standard pinout and registers (HW and SW drop-in compatible)
  - Renesas part numbers: RG3MxxB12B0
  - NXP part numbers: P3H2x4x
- Off-the shelf components already available from multiple vendors

# I3C Addresses SMBus Limitations

- Clear addressing architecture – all devices support dynamic address
- Comprehends MUX-based and MUX-less topologies:
  - BMC is initiator driving clock (redundancy supported)
  - Only BMC performs packet writes and reads
    - Endpoints do not initiate transactions
    - BMC optionally enables IBIs from endpoints (useful in MUX-less topology or with HUB)
- Improves security:
  - peer-to-peer communications must go thru I3C Controller (typically BMC)
- Other improvements:
  - supports in-band interrupts (IBIs)
  - supports in-band reset/recovery
  - 12.5Mbps in SDR mode, 25Mbps in DDR mode

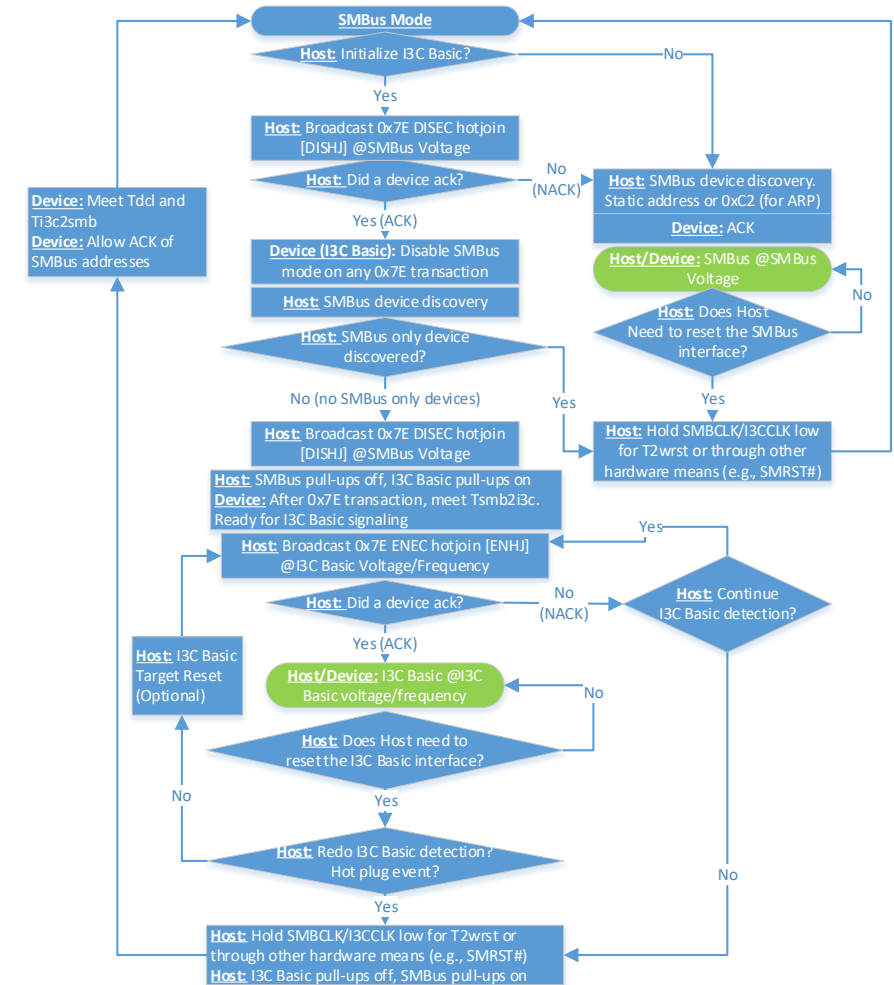
# MCTP I3C Binding ([DSP0233](#))

- **Single I3C Controller**
  - only BMC initiates read/write transactions
  - works well even with traditional MUXes
- **Optional IBIs**
  - improve efficiency but no data loss if dropped
- **I3C CCCs to standardize behaviors, e.g.:**
  - discover capabilities (e.g., protocol)
  - MTU negotiation
  - IBI enable/disable
- **Binding improves robustness over pure MIPI I3C Basic**
  - added error detection and recovery mechanisms



# I3C Discovery Flow as per [EDSFF Specification](#)

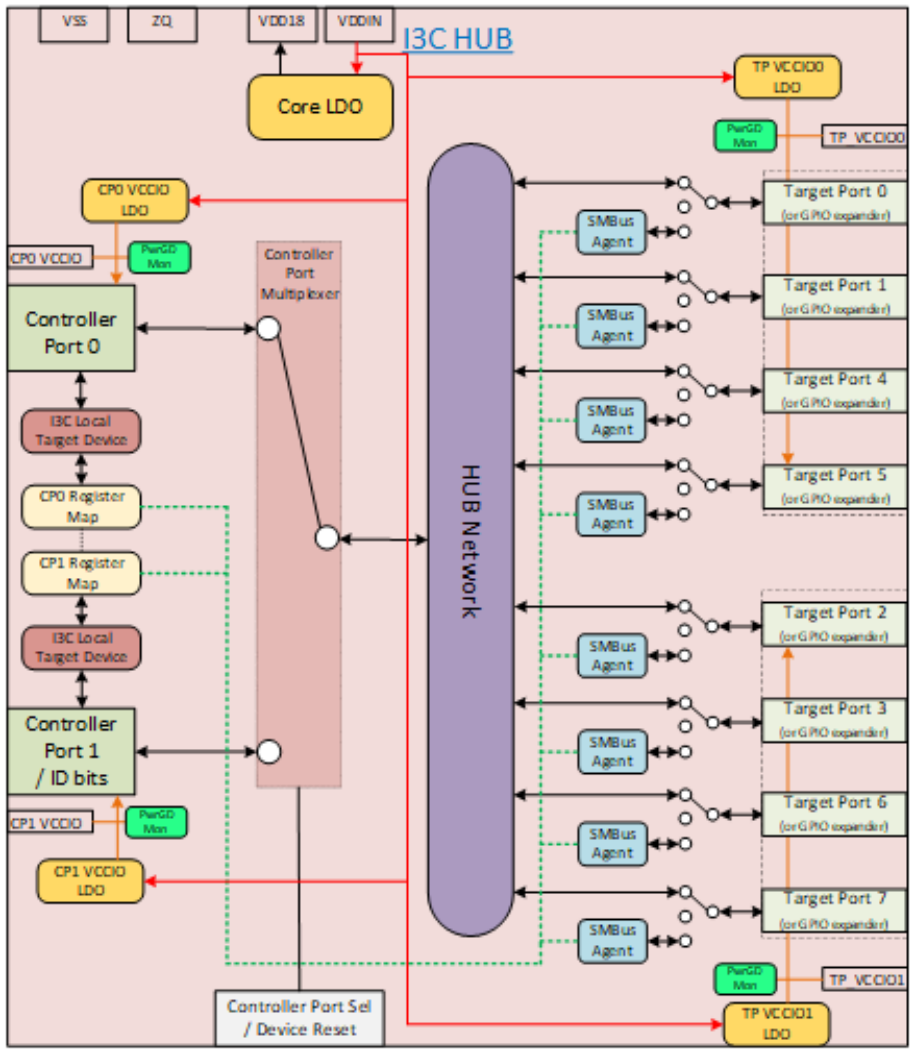
- Discover if any I3C-capable devices are attached
  - Using I3C reserved 0x7E address
- Discover if any SMBus-only devices are attached
  - using ARP or static address scan
- Stay in SMBus mode if at least one SMBus-only device present on targeted EDSFF port
  - I3C-capable devices still must be backwards compatible
  - [MCTP over SMBus binding](#) in use
- Switch to I3C if all devices on targeted EDSFF port support I3C
  - transition to lower voltage
  - [MCTP over I3C binding](#) in use
- Each downstream port can operate at either I3C or SMBus independently



Taken from SFF-TA-1009 Rev 3.1

# I3C HUB Solves SMBus Issues & Enables Transition to I3C

- No arbitration or address issues for legacy SMBus
  - downstream port SMBus agent allows asynchronous and bi-directional communication with multiple SMBus endpoints
    - no endpoint devices changes needed to avoid SMBus limitations
  - SMBus agents enable concurrent and independent communication on each port
  - supports “SMBus busy signaling” for flow control to prevent packet losses with protocols such as MCTP
  - no bus switching during transmission as in existing SMBus/I2C Muxes
- Supports mix of SMBus and I3C devices
  - downstream ports independently operate in I3C transparent mode or SMBus agent mode
  - port in I3C mode allows for electrical isolation while maintaining same logical I3C network (protocol transparent)
  - supports voltage translation
- Reduces number of SMBus/I3C ports needed on BMCs
  - up to 8 downstream ports
  - two upstream ports (allowing upstream device redundancy)

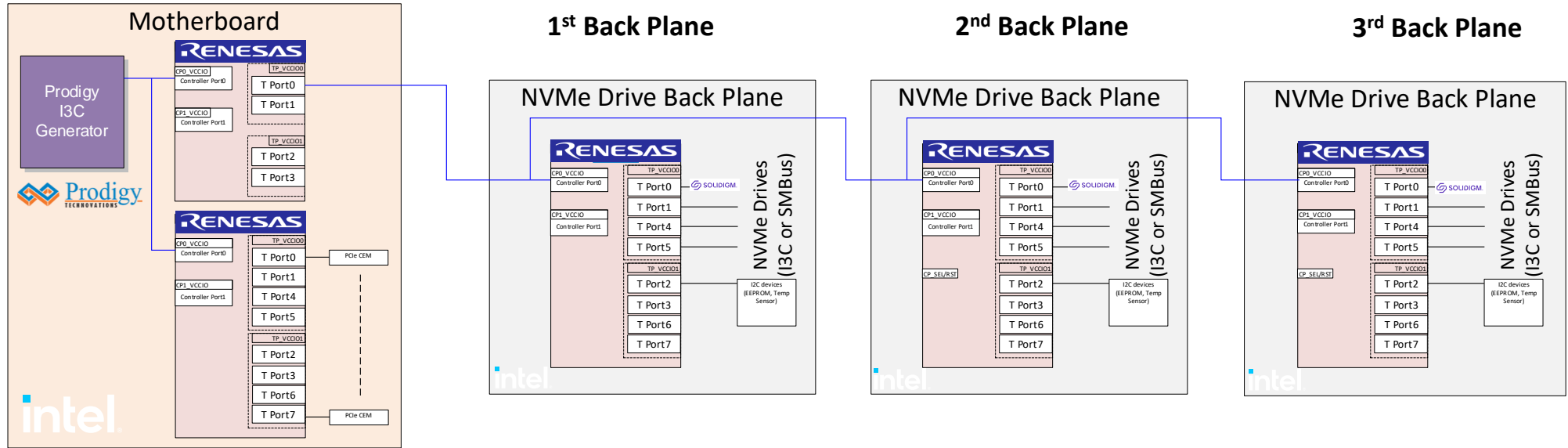


# Experimentation

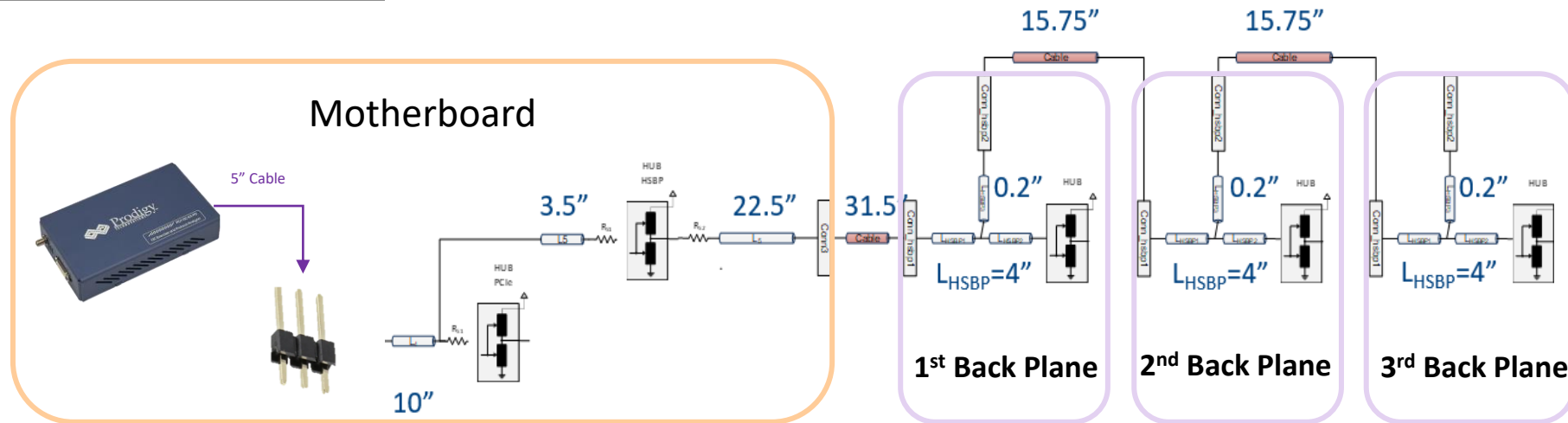
*Intel<sup>®</sup>, Solidigm<sup>®</sup>, Aspeed<sup>®</sup>, Renesas<sup>®</sup>, Introspect<sup>®</sup>, Microchip<sup>®</sup>, Microsoft<sup>®</sup>, Total Phase<sup>®</sup>, and Aardvark<sup>™</sup> registered and unregistered trademarks, service marks, and logos are the property of their owners. Unauthorized use is strictly prohibited.*

# I3C Electricals in typical Storage back plane 2-wire (I3C/SMBus) Channel Topologies

Device Connectivity



Physical Topology



# I3C Electricals in typical Storage back plane 2-wire (I3C/SMBus) Channel Topologies

I3C IO Voltage: 1.8V

I3C Open Drain Frequency: 100KHz

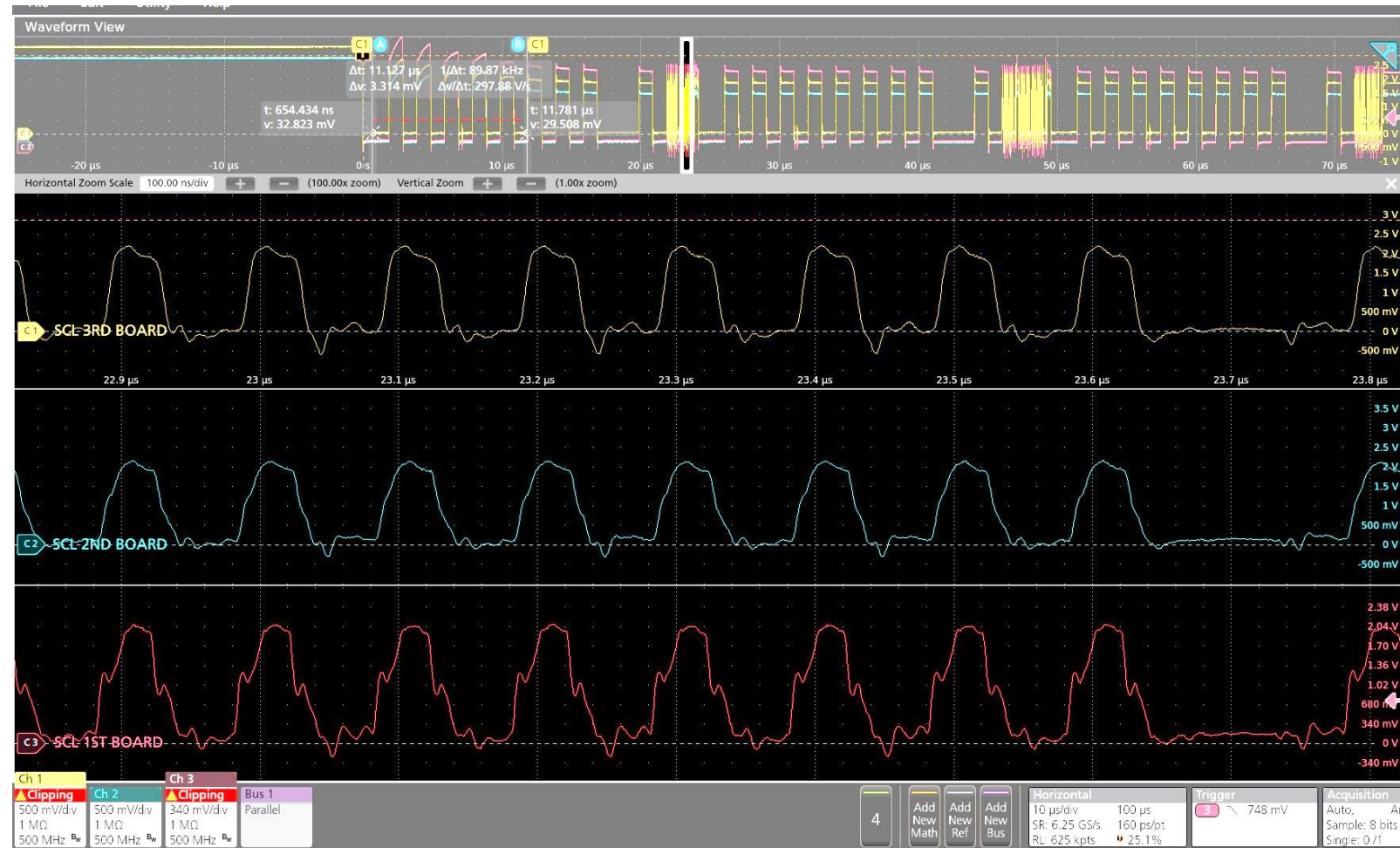
I3C Push-pull Frequency: 10MHz

Operating Conditions: Typical silicon, temp, voltage

I3C SCL on 3<sup>rd</sup> Back Plane

I3C SCL on 2<sup>nd</sup> Back Plane

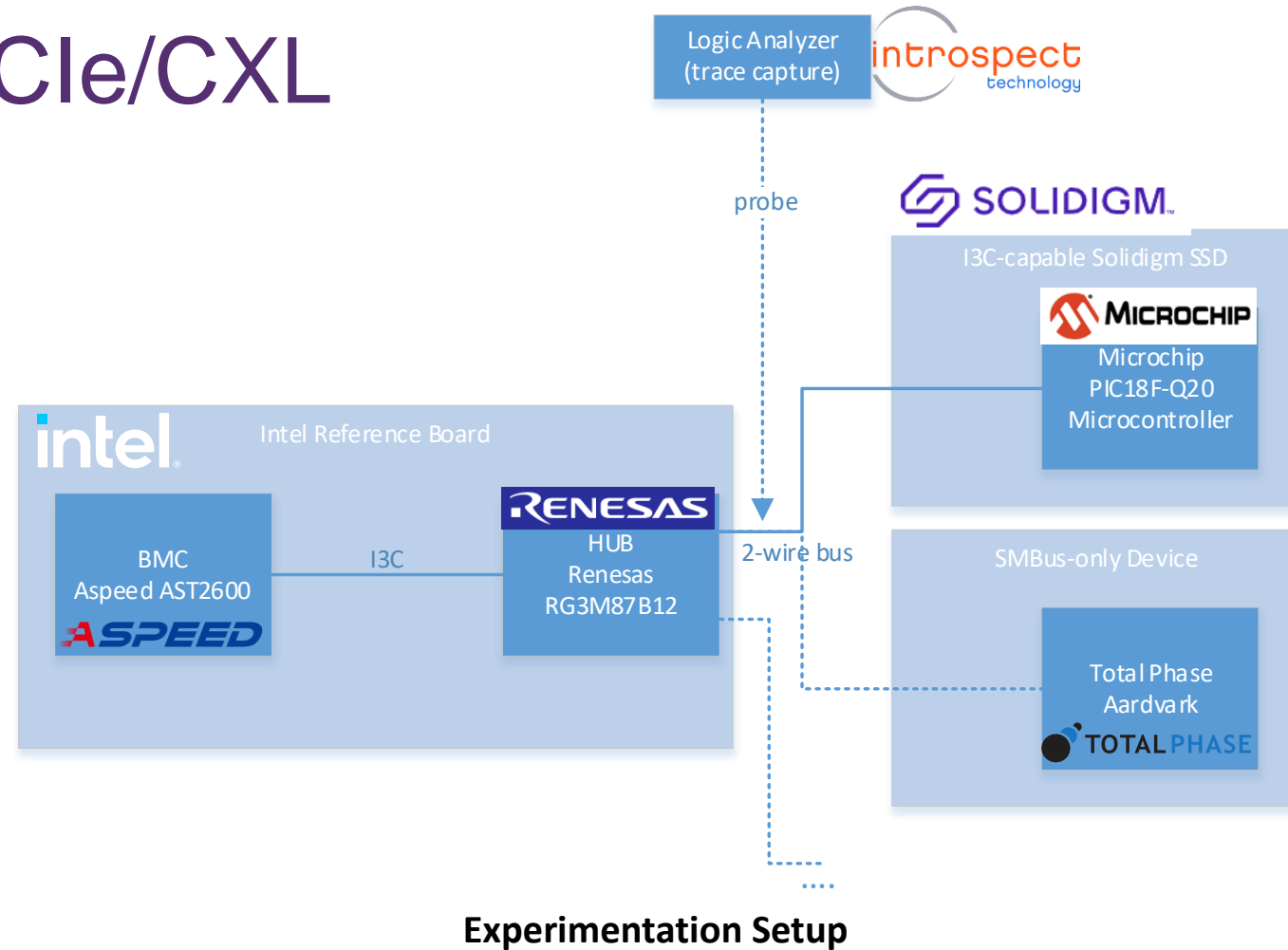
I3C SCL on 1<sup>st</sup> Back Plane



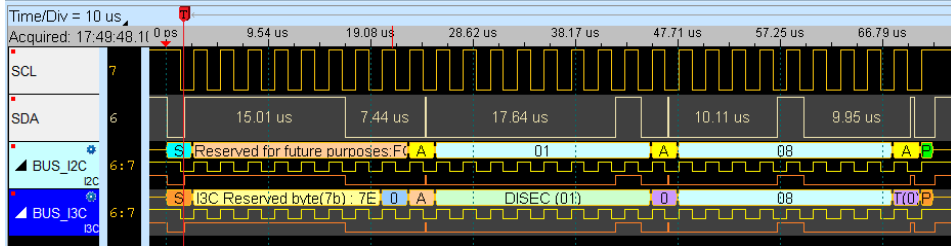
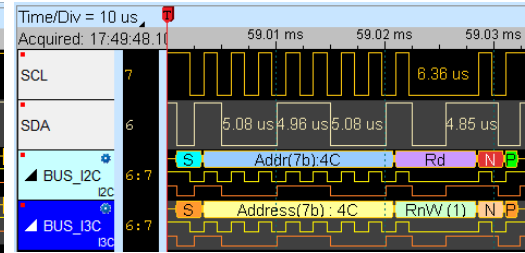
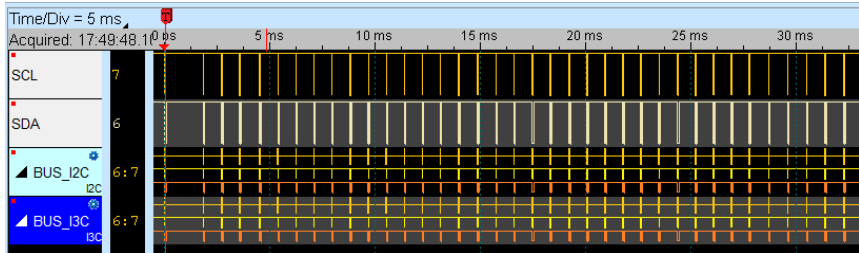
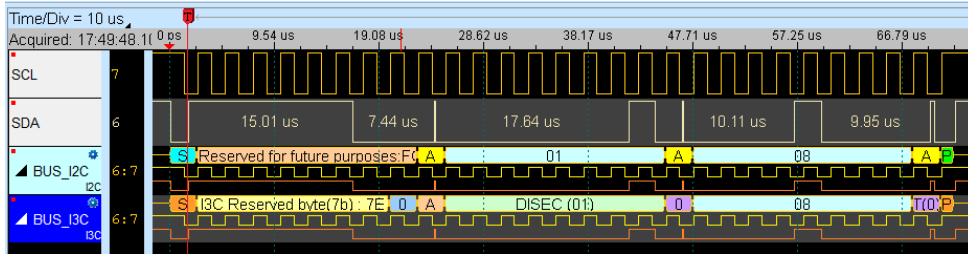


# System Architecture for PCIe/CXL

- BMC with I3C Controller (Aspeed® [AST2600](#))
  - [OpenBMC FW](#) with [MCTP over I3C binding](#) support
- HUB ([Renesas® RG3M87B12](#))
  - Enables SMBus to I3C transition as per EDSFF [SFF-TA-1009](#) specification
  - Fixes [MCTP over SMBus](#) multi-initiator related challenges with multiplexers
  - Common HUB specification (drop-in compatible devices from multiple sources)
- I3C-capable device ([Microchip® PIC18F16Q20](#))
  - Supports [MCTP over I3C binding](#) with sample commands
- Legacy SMBus device (emulated with [Total Phase® Aardvark™](#))
  - Unaware of I3C
  - Unable to work efficiently with multiplexed SMBus (separate experimentation)
- [Introspect SV4E](#) logic analyzer



# Experimentation – Flow with only I3C-capable Devices



1. DISEC in SMBus mode  
(I3C-capable device ACKs)

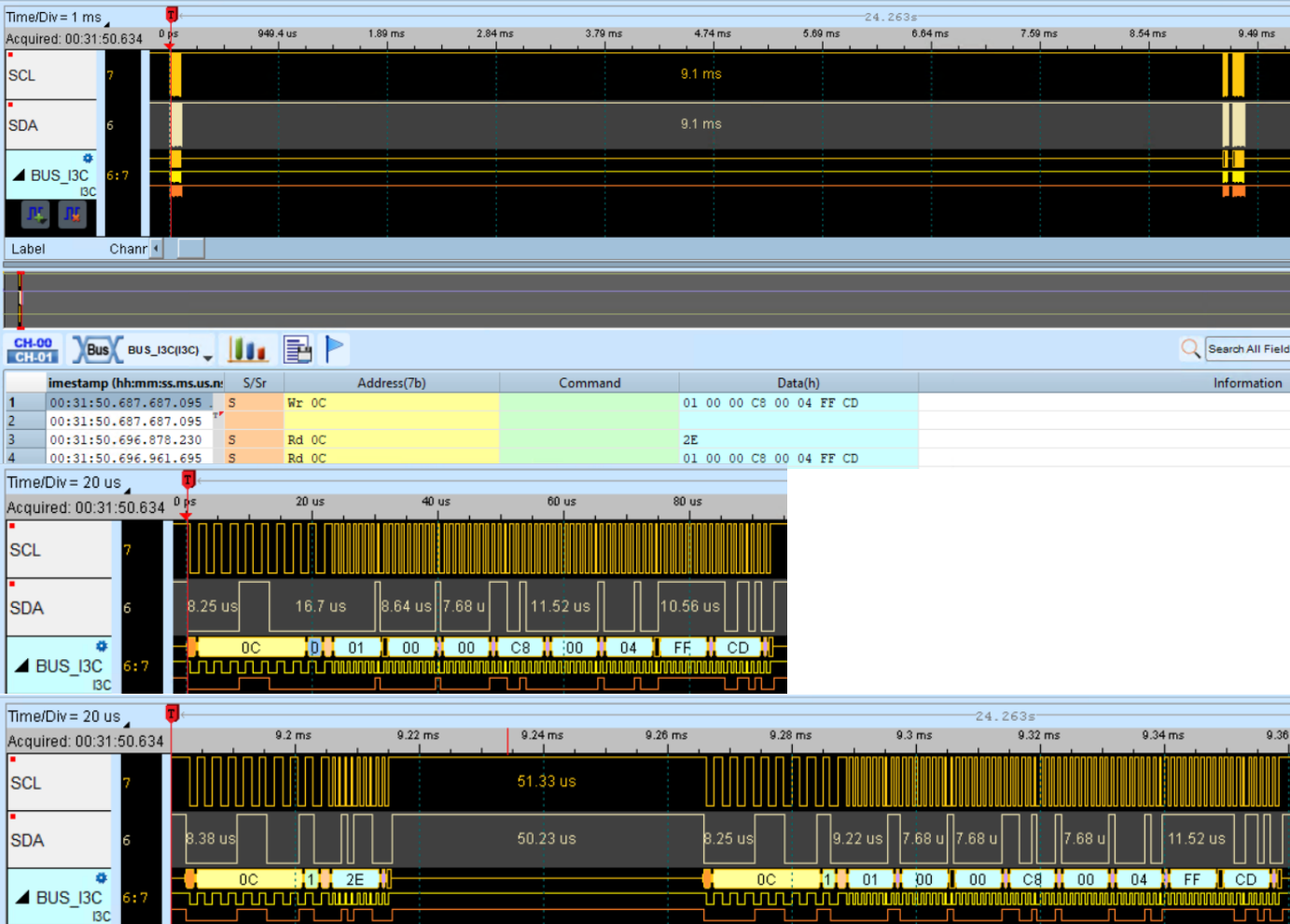
2. SMBus discovery  
2a. No SMBus device ACKed

3. DISEC in SMBus mode  
(triggers switch to I3C mode)

4. Voltage transition 3.3V to 1.8V  
(I3C transactions follow – next slides)

# Experimentation – MCTP request and response in I3C mode with IBI

I3C-capable device responds in I3C mode



Request + IBI + Response

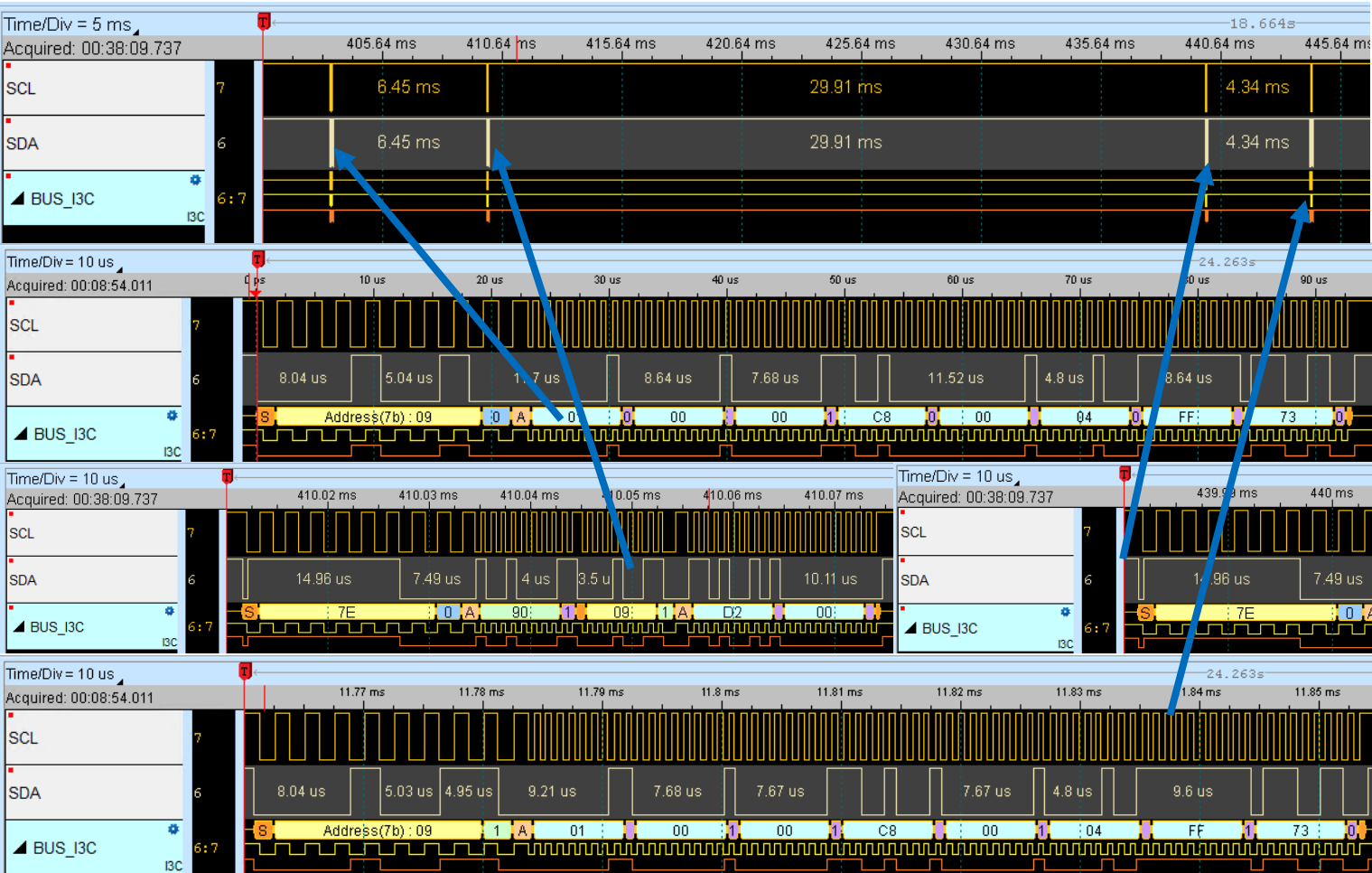
Request

IBI + Response



# Experimentation – MCTP request and response in I3C mode with polling

I3C-capable device responds in I3C mode



Request + 2 x GETSTATUS + Response

Request

GETSTATUS

Response

# Summary & Call to Action

# Summary & Call to Action

- Experimentation confirmed off-the-shelf devices enable backward-compatible transition from SMBus to I3C and address key SMBus limitations
  - MCTP over I3C works well regardless of system architecture
  - MCTP over SMBus works well when MUX is replaced by I3C HUB – no endpoint device changes needed
- Adopt consistent (and backward-compatible) solution in other industry specifications
  - PCI-SIG: [PCI-SIG Protocol WG – Sideband Ad-hoc WG](#), [U.2/SFF](#), [M.2](#)
  - OCP: [Datacenter NVMe® SSD Specification](#), [Datacenter Secure Control Module \(DC-SCM\) 2.0](#), [OCP NIC 3.0](#), [OCP Firmware Recovery](#), [DC-MXIO/DC-MHS](#), [OCP OAI/UBB/OAM](#), ...
- Make your product plans!
  - Intel's reference platform HW ready now, OpenBMC FW aligned with PCIe 6.0
  - Solidigm ready to co-validate with additional partners in 2024



Please take a moment to rate this session.

Your feedback is important to us.