Modular-Peripheral Sideband Tunneling Interface (M-PESTI) Base Specification

Part of the

DC-MHS (Data Center Modular Hardware System) Revision 1.0 family

Version 1.2 Release Candidate 2

March 14, 2024

M-PESTI Revision 1.x Primary Editors/Contributors:

Dell, Inc: Jeff Kennedy, Tim Lambert, Shawn Dube, Charlie Ziegler

Intel Corporation: Javier Lasa, Eduardo Estrada, Clifford Dubay, Aurelio Rodriguez Echevarria, Brian Aspnes

Microsoft Corporation: Priya Raghu, Priscilla Lam

Advanced Micro Devices, Inc: Greg Sellman

Table of Contents

M-PESTI Revision 1.x Primary Editors/Contributors: 2
1. License 5
1.1. Open Web Foundation (OWF) CLA 5
2. Version History 6
3. PESTI 8
3.1 Introduction 8
3.2 Physical Interface 9
3.2.1 Electrical Circuit 9
3.2.2 Electrical Component Selection Factors 10
3.2.3 Voltage Backfeed Detection 10
3.2.4 DC Specifications 11
3.3 Target Detection 12
3.3.1 Target Detection Rules 12
3.4 Protocol Commands 13
3.4.1 Discovery Payload Request (DPR) 13
3.4.2 Virtual Wire Exchange (VWE) 13
3.4.3 Fanout MUX Control 13
3.5 Initiator Abort 13
3.5.1 During Initiator Transmission 14
3.5.2 During Target Transmission 14
3.6 Broadcast 15
4. Initiator Control and Status Registers 16
5. AC Specifications 19
6. Discovery 20
6.1 Payload Requirements 20
6.2 DPR Command and Target Response 21
6.3 Discovery Rules 22
7. Optional Source to Destination (S2D) Detection 23
8 Active Phase 24

8.1 Active Phase Rules 25
8.2 Packet Error Checking 25
8.3 CEM Riser/Interposer DEVICE_CLASS = 0000h 26
8.3.1 Virtual Wires 26
8.3.2 Riser Discovery Payload Requirements 27
9. Target Reset and Fault Handling 27
10. PESTI Fan Out 28
11. Security Considerations 31

1. License

1.1. Open Web Foundation (OWF) CLA

Contributions to this Specification are made under the terms and conditions set forth in Open Web Foundation Modified Contributor License Agreement ("OWF CLA 1.0") ("Contribution License") by:

- Advanced Micro Devices, Inc
- Dell, Inc.
- Google LLC
- Hewlett Packard Enterprise Company
- Intel Corporation
- Meta Platforms, Inc.
- Microsoft Corporation

Usage of this Specification is governed by the terms and conditions set forth in **Open Web Foundation Modified Final Specification Agreement ("OWFa 1.0") ("Specification License").**

You can review the applicable OWFa1.0 Specification License(s) referenced above by the contributors to this Specification on the OCP website at http://www.opencompute.org/participate/legal-documents/. For actual executed copies of either agreement, please contact OCP directly.

Notes:

1. The above license does not apply to the Appendix or Appendices. The information in the Appendix or Appendices is for reference only and non-normative in nature.

NOTWITHSTANDING THE FOREGOING LICENSES, THIS SPECIFICATION IS PROVIDED BY OCP "AS IS" AND OCP EXPRESSLY DISCLAIMS ANY WARRANTIES (EXPRESS, IMPLIED, OR OTHERWISE), INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, OR TITLE, RELATED TO THE SPECIFICATION. NOTICE IS HEREBY GIVEN, THAT OTHER RIGHTS NOT GRANTED AS SET FORTH ABOVE, INCLUDING WITHOUT LIMITATION, RIGHTS OF THIRD PARTIES WHO DID NOT EXECUTE THE ABOVE LICENSES, MAY BE IMPLICATED BY THE IMPLEMENTATION OF OR COMPLIANCE WITH THIS SPECIFICATION. OCP IS NOT RESPONSIBLE FOR IDENTIFYING RIGHTS FOR WHICH A LICENSE MAY BE REQUIRED IN ORDER TO IMPLEMENT THIS SPECIFICATION. THE ENTIRE RISK AS TO IMPLEMENTING OR OTHERWISE USING THE SPECIFICATION IS ASSUMED BY YOU. IN NO EVENT WILL OCP BE LIABLE TO YOU FOR ANY MONETARY DAMAGES WITH RESPECT TO ANY CLAIMS RELATED TO, OR ARISING OUT OF YOUR USE OF THIS SPECIFICATION, INCLUDING BUT NOT LIMITED TO ANY LIABILITY FOR LOST PROFITS OR ANY CONSEQUENTIAL, INCIDENTAL, INDIRECT, SPECIAL OR PUNITIVE DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS SPECIFICATION, WHETHER BASED ON BREACH OF CONTRACT, TORT

(INCLUDING NEGLIGENCE), OR OTHERWISE, AND EVEN IF OCP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

2. Version History

Date	Version #	Description
4/22/22	v0.84	Initial public release
6/22/22	v0.85	-Relaxed tAPTAR timing to align to M-CRPS requirements. -Revised Source to Destination detection algorithm to remove 'misses.' -Added Fanout support chapter
8/11/22	v0.90	Added Fanout dead lock handling guidance General cleanup Removed Future Work section Added contributor name
9/19/22	V1.0 Release Candidate	Added security section Clarified width field refers to connector width and not AIC Clarified fanout mode status and control
9/23/22	V1.0 Release Candidate 2	Added new intro section Removed Supplemental B. It was M-XIO supported configs unrelated to M-PESTI base spec Numerous minor rewordings based on feedback for clarifications.
6/23/23	V1.1 RC5	BAUD accuracy tolerance changed to +/-2% Deleted the following sections: Acknowledgements, Scope, Benefits, Application Examples, Typical OCP Sections Supplemental Material Examples (Candidate for white paper) Introduction was revised Re-ordered sections to describe required features before introducing optional features Revised Source to Destination discovery method Updated Figure 2 to more clearly represent voltage domains. Pull-up resistor recommendation/example changed from 127K to 100K ohms (Figure 2,10) Deleted Figure 3 (Figure 4 is a superset) Deleted table 1 (redundant) Deleted Target Implementation Options table (redundant) Deleted Payload Option table (redundant) Simplified & Moved Discovery Status transition table (now table 3) Deleted "Express" Discovery Example (redundant) Changed "AFU" to "RSVD" to align to PCIe Spec language Deleted Figure 13-16 (not substantive) Candidates for a white paper

		Formatted Figure 17 to match others Discovery rule error correction related to RX timeout Many clarifications and simplifications (general cleanup) to align with Cabled PCIe specification. Corrected Endpoint Descriptor byte length of payload. Added page numbers (footer) Added PESTI fanout protocol details (New MSC command, tBPAUSE AC Timing parameter, and fanout rules) Simplified payload requirements. Format revision 01h and later governed by this specification Modified tAPTAR from 20 us to 200 us in AC timing table Added MUX control timeout to AC timing table Revised system recommendation to voltage feedback description. Added MUX switch control response value definition. MUX fanout snooper rules have been added Clarified CRC-8 exclusion statements during active phase Modified fanout snooper rule to allow virtual wire exchange when no downstream channels are enabled. Clarified SPDM = Security Protocol & Data Model Fixed Section 11 (Security Considerations) header Re-ordered paragraphs in section 9 (Target Reset and Fault Handling) Replaced “shall” with “must” and “may” to “is permitted to”
3/14/24	V1.2 RC2	Added simple presence pull down resistor range. Repaced static timing value with tDBREAK name in section 3.1 Added Discovery Bypass and Active Phase PEC Enable Initiator control registers. Modified Discovery rules with Discovery Bypass considered. Added optional Packet Error Checksum target support during Virtual Wire Exchange Modified discovery payload format and bitfields to match PAYLOAD_FORMAT_VERSION = 02h defined in the PCI-SIG 6.2 specification Removed reference to Example payload in section 8 Clarified discovery status behavior during the active phase in section 8.1

Table 1 : Version History

3. PESTI

3.1 Introduction

This Modular Peripheral Sideband Tunnelling Interface (M-PESTI) specification includes base requirements for protocol and electrical compatibility between components of a DC-MHS platform. The PESTI protocol overloads a common ***PRSNT#*** signal with half-duplex, bidirectional Universal Asynchronous Receive Transmit (UART) communication. Many of the companion DC-MHS specifications require implementation of M-PESTI on I/O and power interfaces at the Host Processor Module (HPM). However, it is not a requirement for any peripheral to implement the PESTI target capability. A peripheral that asserts a static low (present) on any PESTI signal is supported and compliant to this specification.

Half-duplex UART communication requires roles to be defined for each communication partner. PESTI communication is based on a command and response method where the peripheral/target is permitted to only transmit data upon request by the initiator. The initiator is typically a Programmable Logic Device (PLD) integrated into a HPM and/or Power Distribution Board (PDB.) The target is a peripheral that is attached to the system in any number of forms as described in other DC-MHS specifications such as:

- Storage subsystem
- Control Panel
- Riser/Interposer
- Fan Controller Board

3.2 Physical Interface

PESTI utilizes UART framing that has been in broad use across information data systems for decades. RS-232 is one example of a communication standard that utilizes UARTs to exchange data with one another. PESTI does not require a RS-232 transceiver because it operates directly at 3.3V logic levels and is a single, half-duplex signal.

- +3.3 Volt LVCMOS (open drain)
- 250,000 BAUD +/- 2%
- 8-O-1 (8 data bits, Odd Parity, 1 stop bit)
- Half-duplex

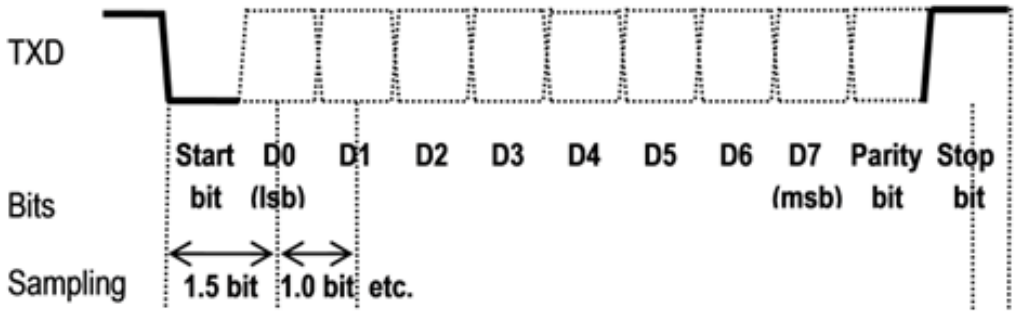

Figure 1 : UART Data Framing

3.2.1 Electrical Circuit

The circuit shown below is such that an unpowered (V_3P3_Target = 0 Volts) peripheral device will hold the PESTI wire low indicating simple presence. Once the peripheral has power and can respond to a command, the target General Purpose Input/Output (GPIO) sourcing the gate of the P-FET will be driven low. Once V_3P3_Target is engaged by turning the P-channel FET on, a rising edge of the PESTI signal will be observed at the initiator. For this mechanism to operate as intended, the default state of the target GPIO must be an input (high impedance).

When used as simple presence, the initiator side circuit must remain the same as shown in Figure 2. The adapter or component side circuit can be simplified to a pulldown resistor with a maximum value of 2.27 Kohm +5% tolerance and minimum nominal value of 200 ohm -5% tolerance.

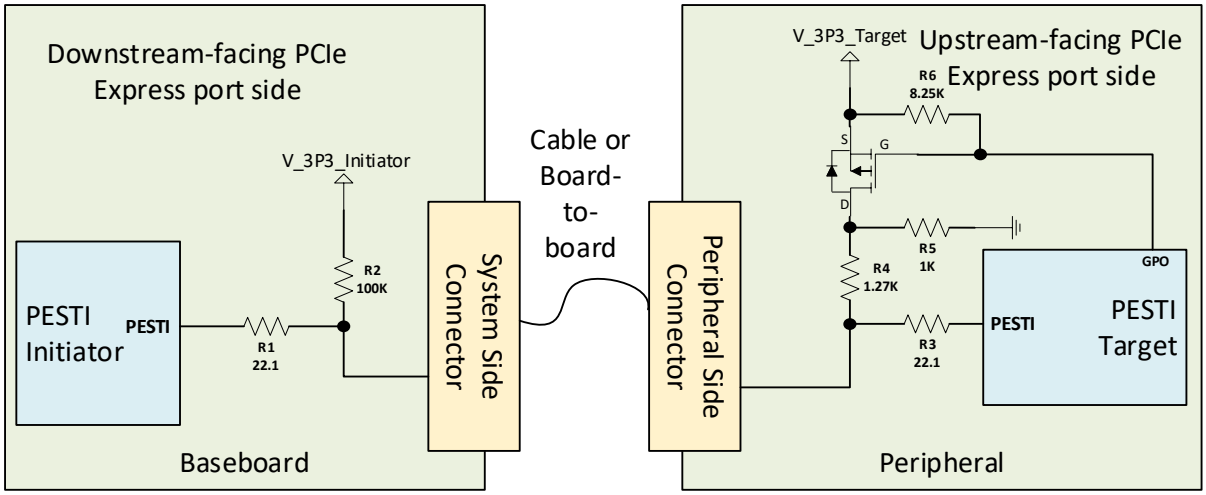

Figure 2 : PESTI Circuit Diagram

For non-hot-plug applications, PESTI is permitted to double as the presence signal, thereby incurring no added pin tax. Doing so for hot-plug applications is not advised due to potential ambiguity or delay in determining a physical presence change versus an unresponsive target. PESTI is permitted to be utilized to indirectly tunnel physical presence status of hot-plug capable downstream targets.

3.2.2 Electrical Component Selection Factors

Referring to Figure 2 above, R1& R3 are recommended series termination resistor values. The values can be adjusted to the driver and transmission line characteristics.

R2 must be selected considering:

- Select the maximum value to minimize the current sourced into an unpowered target.
- Logic high=1 must be guaranteed to meet the baseboard initiator logic Vin High Minimum when cable/PESTI target is not attached/present.

R4/R5 must be selected considering:

- R4 &R5 (pull-down) and R2 (pull-up) must guarantee a logic low=0 at the baseboard initiator logic for voltage back feed detection.
- R4 value provides the rise time to a logic high at the MCU/FPGA for an open-drain interface.
- R5 provides a path to GND for the current sourced by R2 on the source board to drain or bleed voltage accumulation at an unpowered target.

3.2.3 Voltage Backfeed Detection

It is permissible for System Management Firmware (SMFW) to prevent system power on if an unpowered peripheral is detected because of a cabling error. An unpowered target device is indistinguishable from a target that implements simple presence. It is recommended that system peripherals support M-XIO, PESTI_CBL_PRES_N source to destination discovery via I2C even when PESTI virtual wires are not required. See Optional Source to Destination Discovery section for implementation details.

3.2.4 DC Specifications

Symbol	Parameter	Min	Max	Units	Comments
VDD	Bus Voltage	3.135	3.465	V	3.3 +/-5%
VIH	HIGH level input voltage	2.0		V	3.3V LVCMOS
VIL	LOW level input voltage		0.8	V	3.3V LVCMOS

Table 2 : DC Specifications

3.3 Target Detection

Presence of a PESTI target is characterized by a rising edge of PESTI. The default input state of PESTI at an initiator is HIGH=1 indicating that there is no device present. If the PESTI signal is observed as static low for more than tDBREAK, that target device is indicating simple presence. A rising edge of the PESTI signal indicates to the initiator that a target is ready to receive and process commands.

A UART BREAK event is defined as the wire being driven/held low for a time greater than the entire frame length. Example: At 250 KBAUD, the PESTI frame length is nominally 4 us * 11-bit positions which is 44 us. PLD logic is permitted to detect a BREAK condition by starting a timer at the falling edge of the signal. A valid BREAK assertion does not require a falling edge to be detected. In some cases, target UART receiver logic might not implement a BREAK_DETECT status register. Depending upon the UART implementation in a Microcontroller Unit (MCU) or other device, a BREAK event might present itself in status registers as an abnormal frame with a data value = 00h that has both a Parity Error (PE) and a Framing Error (FE : Stop bit = 0.)

A valid PESTI frame is characterized by a Start bit of LOW=0, Stop bit of HIGH=1, and an odd number of ones (including Parity.)

3.3.1 Target Detection Rules

1. An initiator must not attempt communication while PESTI is held low (simple presence) by the target.
2. A PESTI target must not release BREAK until it is ready to respond to the payload request command.
3. Minimum low (BREAK) assertion width required to guarantee detection at the initiator is tDBREAK.
    a. If minimum assertion width is met, BREAK must always be detected by the initiator.
4. A device is permitted to request a re-start of the discovery process by asserting and releasing BREAK at any time prior to the transition to the active phase (see **section 3.4.2: Virtual Wire Exchange**)

The PESTI circuit shown in Figure 2 guarantees that BREAK will be asserted from the time that system input power is applied to the time that the target is powered and ready to respond to a command. In a DC-MHS platform, the target is not powered until 12V_PRIMARY (defined in M-PIC specification) is enabled.

3.4 Protocol Commands

3.4.1 Discovery Payload Request (DPR)

PESTI compliant targets must support the following command.
Discovery Payload Request: Data Byte Value = 00h

Following PESTI target presence detection, the initiator will transmit the payload request command to the target. The target response is to transmit the contents of its discovery payload for capture by the initiator. See ***section 12*** for payload requirements. The command is permitted to be sent autonomously by a PLD or under the direction of SMFW. See ***section 4*** for initiator interface registers accessible to SMFW.

3.4.2 Virtual Wire Exchange (VWE)

PESTI compliant targets must support the following command.

- Virtual Wire Exchange: Data Byte Value = 01h

Once a target's attributes (number, type and order of virtual wires) have been identified by parsing the payload, an exchange of virtual wires between initiator and target is permitted to commence. This phase of the protocol is defined as the Active Phase. Virtual wires are optional, and a target is not required to support any virtual wire bytes in or out following the VWE command.

If a PESTI target does not support virtual wires, it must respond with an acknowledge data byte (value = 00h) after receiving the VWE command.

The acknowledge byte can be utilized as a health monitor of the target. Alternatively, SMFW is permitted to disable the active phase by clearing APEN=0 (see **section 4**) to disable communication to a target after the discovery phase.

3.4.3 Fanout MUX Control

The Mux Switch Command (MSC) command is intended for PESTI fanout control devices only.
MUX/SW Control: Data Byte Value = 02h, See ***PESTI Fanout*** section of this document for additional details.

3.5 Initiator Abort

A PESTI transaction sequence is defined as a series of data bytes exchanged between an initiator and a target. A sequence begins with the initiator transmitting one or more data bytes to the target. The sequence ends when the initiator receives the expected number of data bytes

from the target. During the active phase, a target is required to recognize that a sequence is being terminated prior to its completion. The conditions for requiring termination of a sequence are architecture and implementation specific.

3.5.1 During Initiator Transmission

In place of a valid data frame, the initiator is permitted to assert a BREAK condition to cancel a sequence that is in progress. The initiator must not interrupt a data frame after the Start bit has been sent. If an event occurs that requires an abort after a Start bit has been transmitted, the initiator must not assert BREAK until ***tTIDLE*** (See PESTI AC Specification table) after the previous Stop bit has been transmitted. When observed by a target, it must ignore previous data bytes in that sequence and prepare for the start of the next sequence.

3.5.2 During Target Transmission

The initiator abort assertion is permitted to occur at any point within a target transmitted frame. It is also permitted to occur during the active phase turn around period between initiator transmit and target transmit. In this case, it would be asserted following the Stop bit of the final byte transmitted by the initiator. The assertion width (tABREAK: See PESTI AC Specification table) is guaranteed to be detected by a target that samples the PESTI logic state prior to transmitting any Start bit in the sequence. When an abort is detected, a target must not attempt to transmit any additional bytes and prepare itself for a new sequence start from the initiator.

3.6 Broadcast

To minimize initiator PLD logic, a single UART instance is permitted to be shared (multiplexed) among several targets. Multiplexing requires that targets be serviced in a round-robin rotation. Round-robin servicing has a limitation that responses cannot be captured from multiple targets simultaneously. However, an initiator is permitted to implement logic to transmit a command to several targets in parallel.

A broadcast sequence is characterized by the first byte of the sequence beginning with a data byte value = FFh. The second byte of the sequence is the command. A target must not respond to the broadcast command with data or an acknowledge back to the initiator.

The Power Brake (e.g., PCI Express PWRBRK#) sideband signal is permitted to be used to rapidly reduce power consumption of one or more endpoint devices. PESTI initiators that implement a shared UART must support broadcast of a Power Brake event. PESTI targets that implement virtual wire tunneling of Power Brake must support the following broadcast sequence

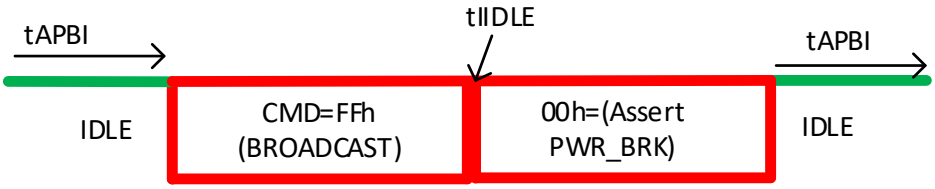

Figure 3 : Broadcast Message Format

Round-robin servicing naturally staggers PWR_BRK de-assertion during a virtual wire exchange which is a common system preference to avoid excessive inrush current from high power loads. Additional delay between PWR_BRK de-assertion to multiple targets is implementation specific and controlled via SMFW and PLD logic that feeds into the appropriate PESTI channel.

An initiator is permitted to issue broadcast commands back-to-back without waiting for a response.

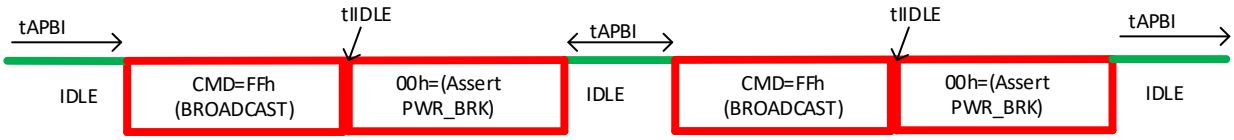

Figure 4 : Broadcast with Optional Repeat

4. Initiator Control and Status Registers

The following initiator registers enable SMFW control and status indicators for each PESTI wire connected.

DISCOVERY_STATUS[1:0] (RO) [AKA: DSTAT]

00b: No BREAK detected; PESTI wire is static HIGH=1 (***Default***)
01b: PESTI wire is static LOW=0 (simple presence)
10b: Discovery payload has been received with a good checksum.
11b: BREAK release detected, but payload has not been successfully received.

Initiator implementation requires the following behavior.

- Once a BREAK release is detected, the allowable DSTAT values must exclude 00b until the next fundamental PLD reset or cycle of system input power.
- If APEN (see below) is SET=1b and DSTAT=10b, then the target device is in the active phase and a BREAK condition on the PESTI wire must not transition DSTAT to a value of 01b. If APEN = 0b, then a BREAK asserted by the target will result in DSTAT=01b. See ***section 9*** (Target Reset and Fault Handling.)

DSTAT Current State	DSTAT Future State	State Input
XXb = Any State	00b = Absent	Initiator Power on Reset (POR)
Absent	01b = PRSNT#	Simple PRSNT#, an unpowered peripheral or PESTI BREAK asserted following POR de-assertion
PRSNT#	11b = PESTI Presence	BREAK release by the target
PRSNT#	11b = PESTI Presence	Simple presence or PESTI device removal prior to target BREAK release while the initiator logic is active
PESTI Presence	10b = Payload Good	Payload received without any errors
Payload Good	11b = Present / M-PESTI	BREAK assertion and release by target if not in active phase. Payload Good is protected/locked during the active phase against target resets. This transition also occurs if DPEN is toggled from 0 to 1 while the device is still in the discovery phase (APEN=0.)
Present/PESTI	01b = Present/Simple	Target asserts all 1WIREs low that are not used for communication at the transition to the active phase

Table 3 : Discovery Status State Transitions

DISCOVERY_PAYLOAD_ENABLE (R/W) [AKA: DPEN]

0b: (***Default***) Initiator will not send payload request command.

1b: Initiator will send payload request command, with retries, until DSTAT=10b

Initiator logic implementation requires that a transition of DPEN from LOW=0 to HIGH=1 by SMFW must clear any "Discovery Complete" flag indicated by DSTAT=10b. As a result, DSTAT would transition to 11b and the DPR command would be transmitted to the target.

ACTIVE_PHASE_ENABLE (R/W) [AKA: APEN]

0b: Initiator will not enter virtual wire exchange phase for that PESTI instance.

1b: (***Default***) Initiator will enter into the active phase if DSTAT[1:0] = 10b

ACTIVE_PHASE_ERROR (R/W1C) [AKA: APERR]

0b: (***Default***) Response RX error (i.e., parity, framing) or timeout has NOT occurred.

1b: Response RX timeout or data error has occurred since last cleared (sticky.)

SMFW must write a '1' to this bit to clear the error once it has been acknowledged.

ACTIVE_PHASE_STAT (R/W) [AKA: ASTAT]

0b : (***Default***) Target device is in the discovery phase.

1b : The Initiator and Target communication is in the active phase

DISCOVERY_BYPASS (R/W) [AKA: DBYP]

0b: (***Default***) Discovery payload must be received to enable transition to the active phase: DSTAT[1:0] = 10b

1b: Transition to the active phase occurs without successful discovery payload capture by the initiator: DSTAT[1:0] = 11b

In both cases above, APEN must be set=1 to transition the target device to the active phase.

ACTIVE_PHASE_PEC_ENABLE (R/W) [AKA: APPECEN]

0b : (***Default***) Packet Error Check (PEC) byte is not enabled during the active phase

1b : PEC is enabled during the active phase

Completion of discovery is characterized by both of the following being true.

1. Discovery payload has been captured by the Initiator without any errors.
2. APEN = 1

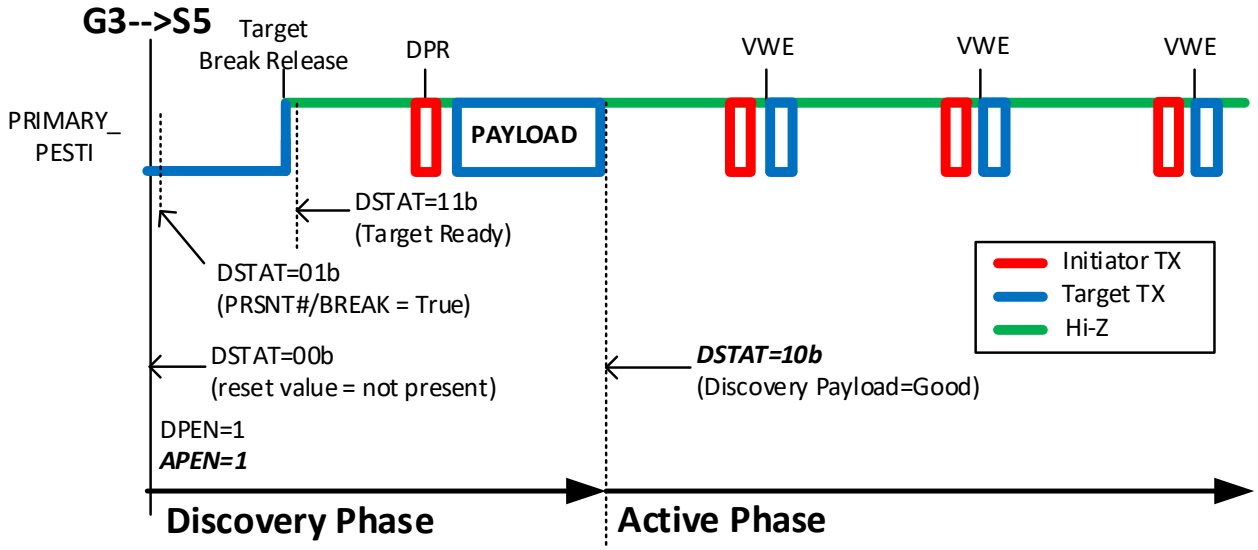

Figure 5 : PESTI Protocol Phases

5. AC Specifications

Symbol	Parameter	Min	Max	Units	Comments
tBAUD	BAUD rate	245000	255000	Hz	250000 +/- 2%
tFRAME	Start + 8b Data + Parity + Stop	42.7	45.3	us	1/tBAUD * 11 bits
tF	Fall Time	-	120	ns	Same as 1MHz SMBus ($V_{IH,MIN}$ + 0.15 V) to ($V_{IL,MAX}$ - 0.15 V)
tR	Rise Time	-	120	ns	Same as 1MHz SMBus ($V_{IL,MAX}$ - 0.15 V) to ($V_{IH,MIN}$ + 0.15 V)
tSPIKE	Noise Spike suppression time	0	50	ns	Same as 400kHz SMBus. Noise suppression is recommended, but not required
tABREAK	Initiator Abort BREAK assertion time	50	55	us	Synchronous to a normal START of frame when the initiator is transmitting. It is asynchronous to START when the target is transmitting.
tDBREAK	Target Discovery BREAK assertion time	50	-	us	PESTI target BREAK can be persistent.
tBPAUSE	Broadcast Pause	20	-	us	Pause to allow fanout snooper completion of channel enables
tIIDLE	Initiator End of STOP to START	0	-	ns	STOP is typically sampled at the midpoint of the bit, so there is approximately 2 us of time to the next START.
tTIDLE	Target End of STOP to START	-	1000	ns	This is required for the target to sample for initiator abort prior to START of target TX.
tMARK	End of BREAK to START time	3.88	4.12	us	MARK time is 1 BAUD period between end of initiator abort BREAK and START of new command
tDPTAR	Discovery Phase Turnaround Time	100	-	ns	MAX not specified. It is bound by payload size and tDPRTO.
tAPTAR	Active Phase Turnaround Time	0.1	200	us	Between Target RX and Target TX of a response. MAX reduces time to sample initiator BREAK/Abort signal just before START. Target MCU should not have trouble meeting minimum time required to allow the initiator to prepare for RX following TX.
tDPRTO	Discovery payload receive timeout	-	250	ms	Allows for 150 ms tDPTAR + 2048 byte payload size.
tAPRTO	Active Phase receive timeout	-	500	us	Includes margin beyond $tAPTAR_{MAX}$ + 8*$tFRAME_{MAX}$ + 7*$tTIDLE_{MAX}$ = 389 us
tAPBI	Active phase bus IDLE time	10	-	us	Between initiator RX from target and initiator TX to same target

tMSC_WDT	Mux Switch Control Watchdog Timeout	100	-	ms	Time for a snooper to detect a stuck bus following a downstream channel select operation

Table 4 : AC Specifications

6. Discovery

A PESTI target is required to support the DPR command when simple presence is not implemented. Payload format requirements enable the initiator implementation to be common and target agnostic. The discovery payload is typically consumed by the initiator logic (e.g., SIZE, CHECKSUM) and SMFW. The payload contents can be scaled to meet vendor requirements.

6.1 Payload Requirements

Byte/Bit	7	6	5	4	3	2	1	0
00h	**PAYLOAD_FORMAT_VERSION [7:0]**							
01h	**DISCOVERY_PAYLOAD_SIZE [7:0]**							
02h	**VENDOR_ID [15:8]**							
03h	**VENDOR_ID [7:0]**							
04h	**DEVICE_CLASS [7:0]**							
05h	**DEVICE_ID [15:8]**							
06h	**DEVICE_ID [7:0]**							
07h	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	**AP_PEC_SUPPORT**	**VENDOR_ID_FORMAT**
08h to N	Optional Vendor Specific Region (Varies) + **Padding (0 – 7 Bytes)							
Nh	**Checksum							

Table 5 : PESTI Payload

**Required regions

PAYLOAD_FORMAT_VERSION[7:0] : Indicates the format version of the payload. This field must be 02h to differentiate it from the previous 01h definition and match the PCI-SIG format version.

DISCOVERY_PAYLOAD_SIZE[7:0] : Indicates the total static payload size. The value of this bit field represents SIZE/8.

VENDOR_ID[15:0] : Utilize the PCI-SIG assigned Vendor ID, when available. If the vendor does not have a PCI-SIG Vendor ID (which applies to PESTI included subsystems such as power converter board or a cable), then set VENDOR_ID_FORMAT = 1.

DEVICE_CLASS[7:0] : Indicates the device class of the PESTI target peripheral Note that a given form factor is permitted to define specific module classes.

DEVICE_ID[15:0] : Vendor specific with intent to represent unique type of device. Not intended for unit specific information such as a serial number.

VENDOR_ID_FORMAT :

0 = PCI-SIG assigned Vendor IDof the company delivering the adapter or component with the PESTI target.

1 = Non PCI-SIG assigned Vendor ID. e.g., Open Compute Project assigned

AP_PEC_SUPPORT : 1 means that the target supports byte packet error check (PEC) during the active phase on command write data and read data returned to the PESTI initiator.

Vendor Specific Region : Implementation specific but is recommended to contain its own sub-payload version.

Padding of N bytes with a value of 00h is required to make the payload size an integer multiple of eight. If the vendor specific region is not used, the number of padding bytes = 00h in the payload is seven.

CHECKSUM[7:0] : CRC-8 checksum with polynomial = 0x07, Seed = 0x00. Both the input and output of the CRC are not inverted.

6.2 DPR Command and Target Response

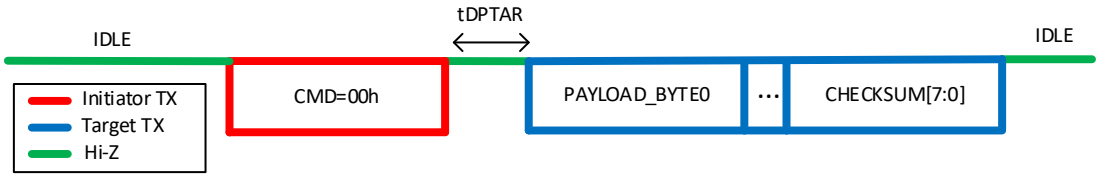

Figure 6 : Discovery Command and Response Message Format

Discovery Payload Turnaround Time (tDPTAR):
Maximum time allowed between initiator completing transmission of the DPR command and the response beginning to be received at the initiator.

6.3 Discovery Rules

Following the BREAK deassertion by a PESTI target, the initiator must transmit the discovery request command to that target if DPEN=1.

- A target that receives the payload discovery command must perform the following sequence.
    1. Tri-state all PESTI target connections that are not utilized for communication.
    2. Sample the input logic state of the GPIOs used for stimulus and response.
    3. Transmit the discovery payload to include any stimulus input state sampled in step 2.
- Turnaround time from target command received to start of response transmission has a minimum requirement of 100 ns.
- A target must complete the discovery payload response within the payload RX timeout of 250 ms.
- If a target BREAK condition is observed by the initiator with APEN=0, DSTAT[1:0] must be transitioned by logic to 11b (Discovery not complete) until a new payload is received successfully.
- The initiator must not transmit the VWE command unless APEN=1 and one of the following:
    1. DSTAT[1:0]=10b indicating the discovery payload has been received successfully
    2. A PESTI target is present and discovery bypass is enabled. DSTAT[1:0]=11b AND DBYP=1
- An initiator must continuously attempt to retrieve a discovery payload from a PESTI target that is present unless DPEN = 0.
    1. Example: Round-robin servicing by a single initiator UART to multiple targets: If the discovery payload is not successfully received after an Initial attempt plus two retries per target, the next target in the round-robin rotation will be serviced. When servicing returns to the target, the discovery request command is sent again as a set (initial + two retries) until successful.

If the target is unresponsive, the minimum time between successive transmissions of the DPR command from initiator to target is 250 ms (tDPRTO). Retries described in the example above are applicable to round-robin implementations only and not a requirement of this specification.

7. Optional Source to Destination (S2D) Detection

An I2C GPIO expander implementation (See Figure 7 below) is recommended to enable SMFW to determine power and data cable couplings between source and destination.

Process Steps:

1. BMC parses the PESTI discovery payload to determine the number of power and data fabric connections to the peripheral.
2. BMC utilizes the I2C segment along the discovered PESTI path to communicate with an I2C GPIO expander.
   - Set known destination CBL_PRES_N GPO to Output = 1.
3. BMC reads the status of CBL_PRES_N GPI at HPM to determine data cable source associated with the peripheral destination that was stimulated.
4. BMC writes to I2C GPIO expander to change the CBL_PRES_N signal state.
   - Set GPO to Output = 0.
5. Continue stimulating each coupled connection, one at a time, to determine the HPM sources.

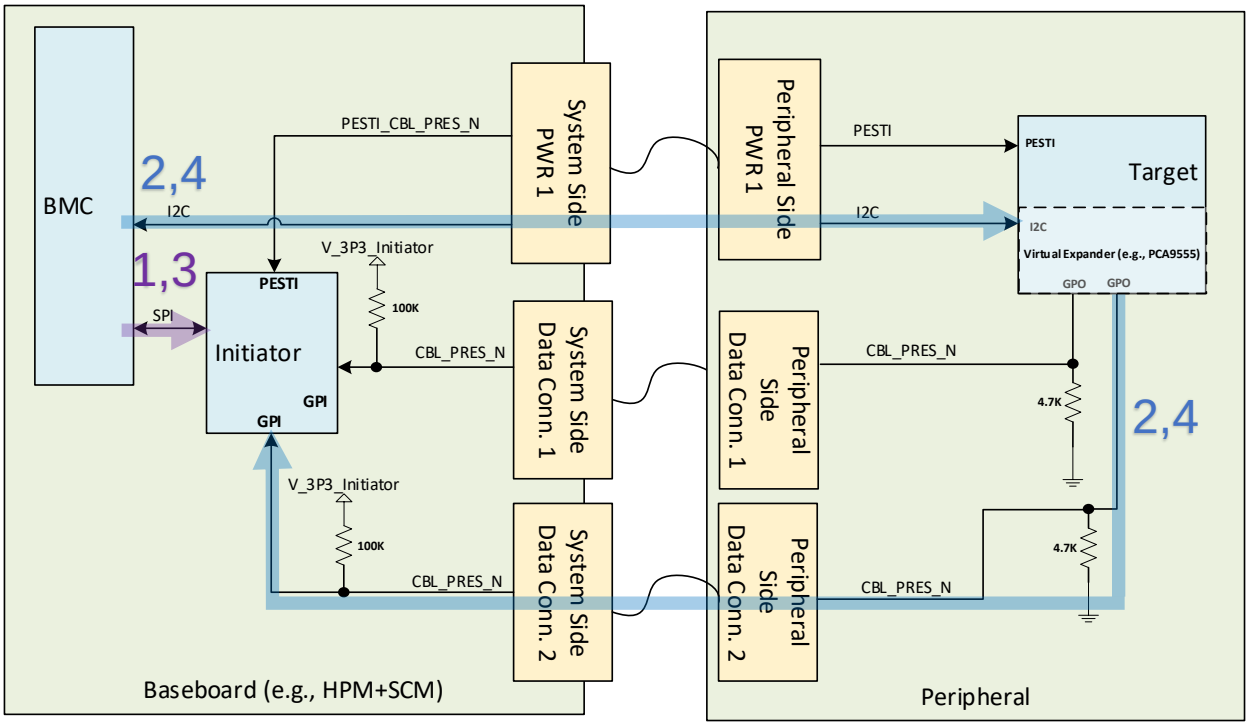

Figure 7 : Source to Destination Diagram

Figure 7 shows a logical view of the stimulus and response paths and includes numeric labels for those paths that align to the process steps above. In this example, the PESTI path is along the power connection sideband (e.g., M-PIC Power Sideband 1) while the data connections (M-XIO PESTI_CBL_PRES_N) signals behave as CBL_PRES_N only and do not require PESTI functionality/logic at the baseboard implementation. If the system design restricts PESTI usage

to the managed power connection sidebands, then all data connector couplings can be implemented as GPI only significantly reducing HPM FPGA logic resource requirements.

8 Active Phase

Once a PESTI is transitioned to the active phase, the initiator autonomously exchanges hardware (HW) controlled virtual wires with each target device. The HW controlled virtual wire inputs and outputs are target DEVICE_CLASS[15:0] specific. The number of bytes in and out and their respective usages are fixed per device class.

If the number of virtual wire bytes out is zero, the initiator will skip transmitting Virtual Wire Output (VWOUT_0) and wait to receive any Virtual Wire Input (VWIN_N) bytes. If the number of virtual wire bytes in is zero, then an acknowledged target response of 00h is expected at the initiator. SMFW is permitted to clear APEN=0 so that the VWE command is not transmitted, and the bus will remain idle until the next power on reset or re-entry to the discovery phase.

Virtual Wire Exchange Example (1 Byte Out/In)

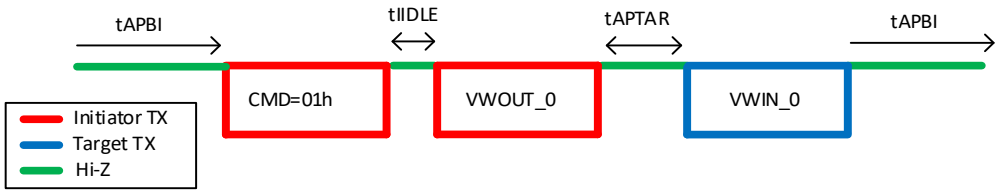

Figure 8: Single Byte Virtual Wire Exchange

Virtual Wire Exchange Example (N Byte Out/ M Byte In)

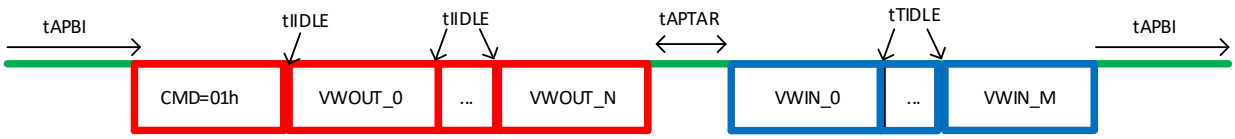

Figure 9: Multi-byte Virtual Wire Exchange

8.1 Active Phase Rules

- Minimum PESTI idle period between initiator RX of the previous target response & TX of the next target command is tAPBI
- While ASTAT=1, DSTAT must remain locked to 11b (BREAK detected) or 10b (Discovery Complete)
- A target assertion of BREAK while ASTAT=1 must result in APERR=1.
- A target must wait a minimum of tAPTAR before transmitting a VWE response.
- A target must complete a response to the initiator within the RX timeout of tAPRTO.
- The maximum period between commands sent from an initiator to that same target is not bound by this specification. The active phase of any device is under the control of SMFW and can be disabled.
- If target supports virtual wire OUTs only, the target must respond with 00h back to the initiator as confirmation of receiving the OUTs.
- SMFW is permitted to clear APEN to disable the active phase for a device.
- Asymmetrical number of out and in bytes is permitted.

The method to route virtual wires to/from internal system logic or other entities and the PESTI is outside the scope of this document. The usage of the virtual wires (internal commands/policies or connections to local physical signals) by the target is also outside the scope of this document.

8.2 Packet Error Checking

Packet Error Checking (PEC) is optional in the initiator and target. PEC is recommended in the active phase when information integrity is critical to the application (e.g., an errant control may cause more ill effects than a bit flip on an analog read that gets averaged out over time). A target's PEC support is described in a discovery payload bit. PEC utilizes CRC-8. Enabling PEC adds to active phase latency.

When supported by the target and set by the initiator, the one-byte command (CMD) bit 7 indicates to the target that the PEC byte is being delivered following with the command byte.

Example: CMD=81h, VWOUT_0, PEC, Turnaround, VWIN_0, PEC

If the target indicates no PEC support in the discovery payload, then the initiator must not set bit 7 in the CMD nor deliver an extra PEC byte following the command. This is because following the turnaround, the target may drive contention on against the incoming PEC byte.

If the CMD to target has bit 7 set indicating PEC:

1. The initiator must supply the PEC byte.

2. If the PEC check at the target fails, then the target must drop the command and, if applicable, receive payload. A method to indicate from target to initiator that a PEC failure occurred is out of scope of this specification.

3. The target must append a PEC byte to the end of the active phase payload response (just as it does for the discovery payload).

4. If the initiator calculates a bad PEC on the target response message, the initiator must drop the response. Further handling, such as retries, is implementation specific.

For single byte commands such as CMD=00h for get discovery payload, PEC is optional, since the 1-byte message includes parity.

Example Supported Commands with and without PEC
01h = Put 1 Virtual Wire Byte, Get 1 Virtual Wire Byte
81h = Put 1 Virtual Wire Byte with PEC, Get 1 Virtual Wire Byte with PEC

8.3 CEM Riser/Interposer DEVICE_CLASS = 0000h

A common building block for system builders is a riser/interposer that electrically and mechanically adapts the DC-MHS (M-XIO + M-PIC) standardized interfaces to one or more PCIe CEM add-in card slots. The following virtual wire mappings are required at an initiator for this device class. A target is permitted to support all, a subset or none of the IN and OUT bytes for the advertised device class.

8.3.1 Virtual Wires

VWIRE_OUT_0 (Initiator to Target)

7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	S0_RUN	PWR_BRK

PWR_BRK : Active high (1=Assert, 0=De-assert) virtual wire

S0_RUN : Active high (1=True, 0=False) virtual wire indicating system power state is ACPI S0_RUN.

RSVD : Reserved for Future Use

VWIRE_IN_0 (Target to Initiator)

7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	WAKE

WAKE : Active high (1=Assert, 0=De-assert) virtual wire that indicates the target device is requesting entry to the ACPI S0_RUN state.

RSVD : Reserved for Future Use

8.3.2 Riser Discovery Payload Requirements

For a target that implements virtual wire mapping that matches DEVICE_CLASS = 00h, it must report the following in the header region of the payload to enable initiator controlled VWE.

DEVICE_CLASS[15:0] = 0000h

A target of DEVICE_CLASS = 0000h that supports zero virtual wires must respond with 00h (acknowledge) to the VWE command. In that case, the target must ignore VWIRE_OUT_0.

9. Target Reset and Fault Handling

If the target resets during the discovery phase prior to the entry to the active phase (e.g., APEN=0), the DSTAT value would revert to 11b (Payload not received.) The initiator would autonomously send the payload request command if DPEN = 1. The target device would not be transitioned to the active phase until DSTAT=10b. If discovery had not previously completed or DPEN=0, the device would not be discovered and transition to the active phase.

If the target resets during the active phase, it must be observed as temporary unresponsiveness at the initiator. This would be reflected in the ACTIVE_PHASE_ERROR (APERR) register described above if any RX timeout, PEC or parity error occurred. Following target reset, a BREAK assertion and release would be observed by the initiator. During the active phase (DSTAT[1:0]=10b & APEN=1), DSTAT must be locked. Locking the discovery status and payload data is required so that FW can safely consume the contents at any warm or cold reset. Since discovery has already occurred, the initiator would resume sending the virtual wire exchange command and the target would not observe a discovery request command. Once a device has transitioned to the active phase, the only method to unlock the discovery payload and status is by SMFW clearing, then setting DPEN.

10. PESTI Fan Out

Applications exist where the ability to fan out a PESTI bus to multiple targets exist. One such example is a motherboard or Host Processor Module (HPM) connection to a Power Distribution Board (PDB) with N number of target subsystems such as backplanes or risers. Since the number N is not a priori known by the motherboard, and pre-plumbing for a maximum quantity requires additional interconnects, PESTI fan out support helps scale the ability to support PESTI type features.

This specification version includes the scope of fanout from between 2 to 8 downstream buses. Although out of scope, nesting of multiple tiers of fan out within a single hierarchy is possible with additional command codes and circuitry. In all fanout cases, it is left to the designer to understand the latency effects of fanout width (and depth). Two methods are shown where the MUX method is for typical fanout needs and the Switch method is targeted for applications that require the ability to broadcast commands simultaneously with all targets.

Two fanout methods of operation include the MUX and Switch methods (see circuits below).

- **MUX method**: Typical 1-to-many fanout. Broadcast commands are ineffective.
- **Switch method**: For applications requiring broadcast commands to all attached targets.

Two modes of operation for a fanout controller include Target Mode & Snoop Mode

- **Target Mode (Default)**:
    - Controller acts as a target supporting discovery and active phases.
    - Only the fanout controller is permitted to be attached to the initiator bus.
        - MUX Method: No channels are selected.
        - Switch Method: Fanout controller on Ch0 is always enabled; all others default disabled.
    - Is permitted to support >=1 status command for the initiator reading information from the fanout controlling PESTI target such as:
        - The current MUX/switch settings
        - If an issue was observed such as a closed switch bus hang watchdog timeout
        - Live status of downstream subsegments (when the MUX is not focused, or switch closed on a particular subsegment)
- **Snoop Mode**: Fanout controllers must:
    - Enter snoop mode any time any other target(s) are attached to a bus.
    - Process broadcast commands if application relevant.
    - Listen/Process only special fanout control commands (MUX Select or Switch channel enable), thereby ignoring discovery and target mode active phase commands intended for other targets.

      Example MUX Select commands are permitted to include:
        - Go to Target Mode and de-select all subsegments.

- Select specific subsegment(s) as connected, which are permitted to be comprised of 1, all or select groups.

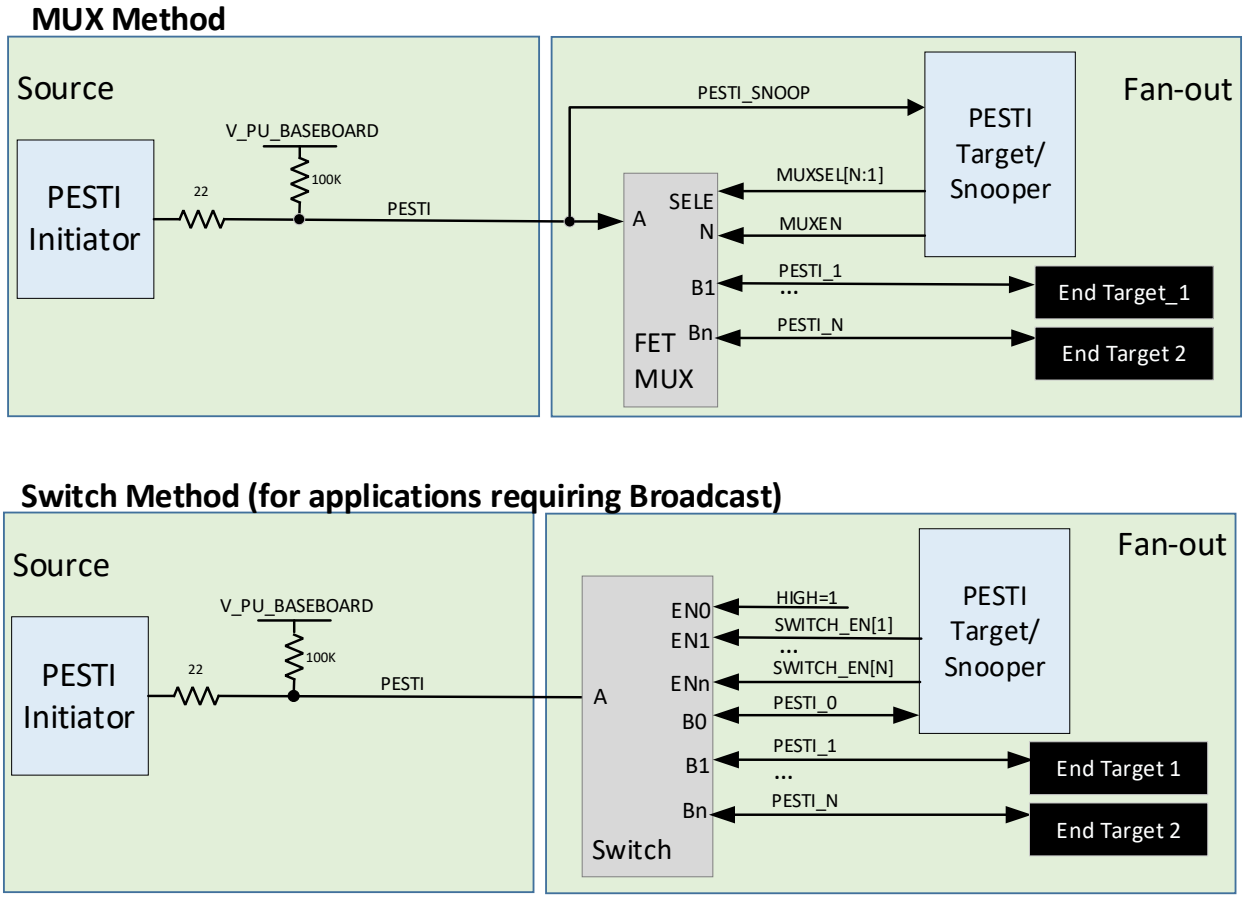

Figure 10 : Example PESTI Fan-out Methods

Figure 11 : Example PESTI Fan-out Methods

Buffer(s) are permitted if the Target/Snooper and the FET MUX or Switch have significant layout stubs. The designer is permitted to choose a switch with active buffers versus a FET switch to avoid reflection effects.

Since any downstream interface can have a strap (such as a strong presence pulldown resistor) or be in a stuck state driving high or low, it is imperative to not create a deadlock condition where the Fanout controller cannot observe new commands to change the MUX or switch settings. Therefore, it is required to have the fanout controller implement a watchdog timer of at least 100ms following selection of 1 or more channels to observe any signaling activity. If there is no such bus activity (rising or falling edges) generated by the initiator, the PESTI snooper must open the selected channel(s) after the timeout.

An alternative option is for the fanout controller to always be able to observe and report upstream the state of any downstream subchannel so that the PESTI initiator can choose to not close a suspect channel.

MUX Switch Control (MSC) Command Structure

MSC_CTRL_VAL (Initiator to PESTI Snooper Target) Data Byte Value = 02h

7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	DS_ALL_CH_SEL	DS_CH_SEL[2:0]			ENABLE

ENABLE : Active high (1=Enable, 0=Disable)
When the mux or switch is disabled, only the snooper target can receive communication from the initiator. All downstream MUX or switch channels must be disabled.

DS_CH_SEL[2:0] : Hex representation of downstream channel select (0…7 decimal). This bitfield is only valid if ENABLE = 1b and DS_ALL_CH_SEL = 0b.

DS_ALL_CH_SEL : When set to 1b, then all downstream channels are connected. Used for broadcast commands only. This bitfield is only valid if ENABLE = 1b. When this bitfield is set to 1b then DS_CH_SEL bitfield must be set to 000b.

RSVD : Reserved for Future Use.

MSC_STAT_VAL (PESTI Snooper Target to Initiator)

7	6	5	4	3	2	1	0
RSVD							NOT_ACK

NOT_ACK : Not Acknowledge
0b : Success
1b : Error

RSVD : Reserved for Future Use.

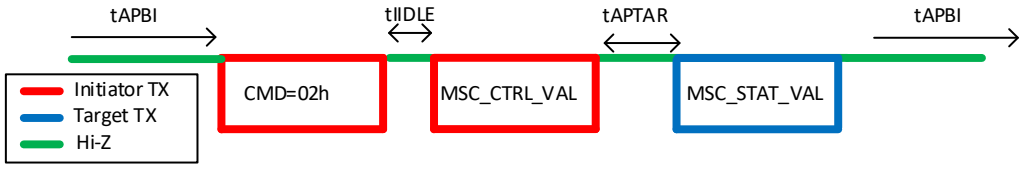

Figure 12 : Mux Switch Control Format

Fanout Rules

1. PESTI snoopers must not be nested. Only a single fanout level is supported.
2. A PESTI snooper fanout target must default to all downstream PESTI channels disabled during the discovery phase.
3. A PESTI snooper must not respond to a discovery payload request command = 00h when any downstream channel is selected.
4. A PESTI snooper must not respond to any virtual wire exchange using command 01h when any downstream channel is selected.
5. Except for when broadcast commands are supported and transmitted, only a single downstream channel of a Switch can be selected at any given time.
6. Non-snooper targets must ignore argument(s) following the MUX_SW_CTRL command = 02h.
7. On any PESTI that a fanout switch is present, the initiator must begin any broadcast with the following sequence.
    a. Broadcast start value = FFh
    b. Wait for tBPAUSE (Broadcast Pause)
    c. Resume with typical broadcast sequence of FFh (repeated) followed by the broadcasted command value(s).
8. A fanout switch snooper must enable all downstream channels at any start of a broadcast indicated by 0xFF as the first, or repeated, data byte of a sequence.
9. Once the broadcast sequence is complete, the initiator must utilize the MSC command to select a single downstream channel of any switch or MUX prior to resuming typical communication.
10. PESTI snooper must unselect any downstream channel after 100 ms of bus stuck low following a command from the initiator to select that channel.

11. Security Considerations

Although PESTI is a basic, low-level data communication channel the following threats are identified with possible mitigations, although the specific mitigations are implementation specific.

Threats

1) Physical implant / signal re-routing: Assurance, where applicable, that the PESTI target is on the same HW (or same target) as other management interfaces. Security Protocol and Data Model (SPDM) types of security capabilities are the likely current art in which to address this threat.
2) Physical implant, man-in-the-middle snooping or alteration of payloads or virtual wires in flight. Potentially mitigated with encrypted payloads most likely using SPDM defined methods.