mipi alliance

IF IT'S NOT MIPI, IT'S NOT MOBILE

A Deep Dive into MIPI Debug for I3C℠

 |

Webinar Agenda

- Overview of MIPI Debug for I3C
- Network Adaptors
- Debug Common Command Codes (CCCs)
- Session Management and Debug Resets
- I3C In-Band Interrupts for Debug Events
- Debug Ecosystem
- Conclusions

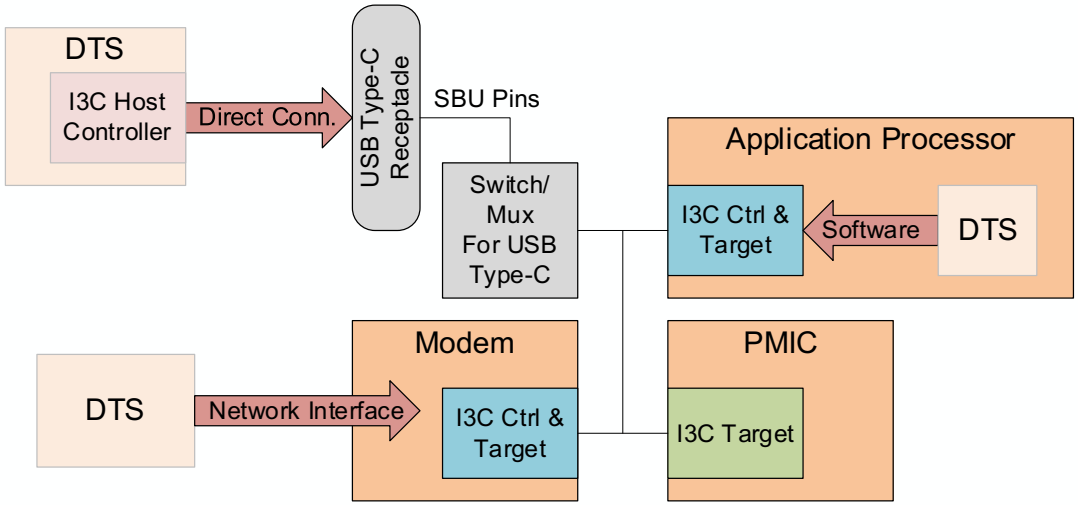
A Deep Dive into MIPI Debug for I3C℠

Matthew Schnoor
Intel Corporation
MIPI Debug Working Group

About MIPI Debug for I3C

Enables new debug interface standard, where existing interface solutions (e.g., JTAG/cJTAG, I^2C, UART) are falling short

- Enables debug of multiple components on the board
- Debug and test system (DTS) can attach to external pins, yet still use the application processor and modem as DTS
- Must work if application processor is powered down (e.g., low power state)

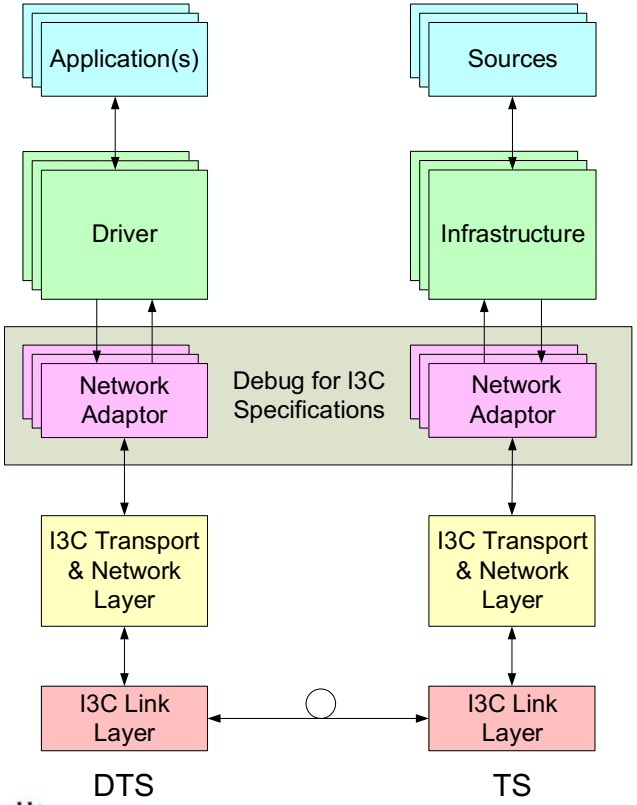


MIPI Debug for I3C Key Features

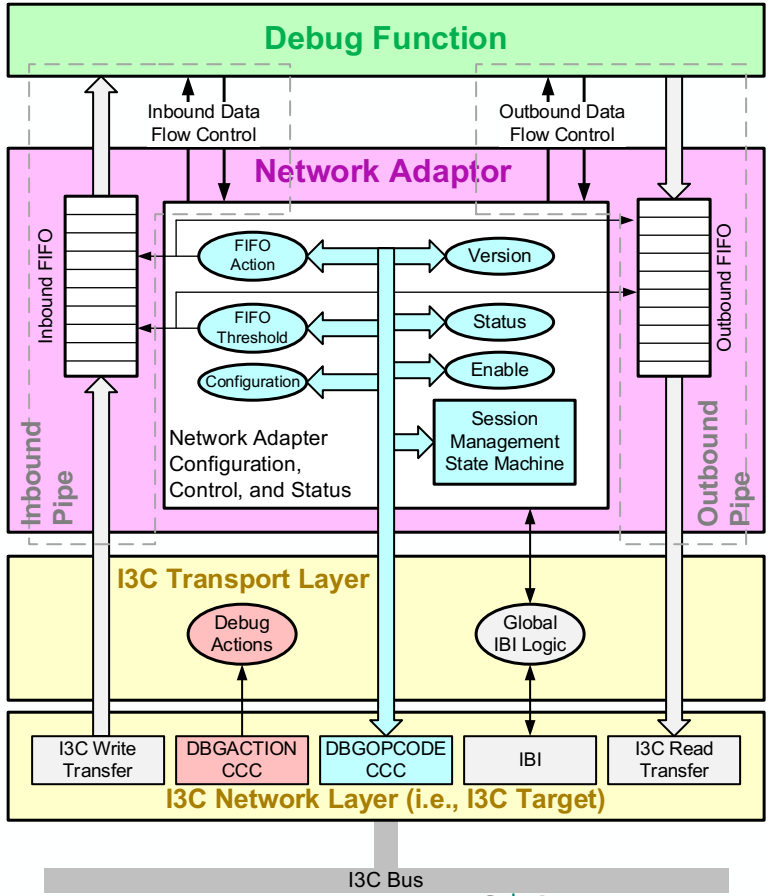
- Uses core capabilities of MIPI I3C® or MIPI I3C Basic℠
- Allows dedicated or shared bus topologies
- Handles debug communication over defined byte-oriented streaming interface ports that can support different protocols
- Allows the target system (TS) to expose multiple debug interfaces/ports (referred to as **Network Adaptors**) from a single physical connection
- Allows the debug and test system (DTS) to send broadcast or directed action requests (e.g., halt, reset)
- Allow the TS to send event indications via IBIs (e.g., triggers, requests)

Conceptual System Diagram

MIPI Debug WG Functional Layering Diagram



Debug for I3C Functional Diagram



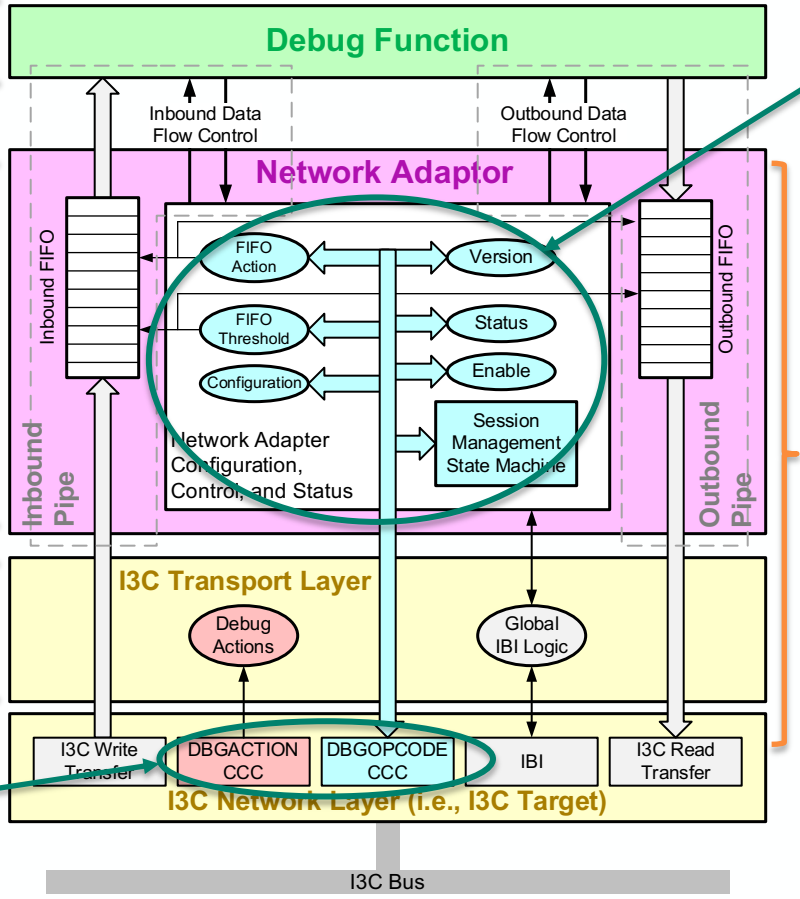
Conceptual Target System Diagram

e.g., Internal TAP network, Debug Registers, Trace Infrastructure

Specific to the debug function(s) on the given SoC. TS may have several Network Adaptors

Debug-specific Transport Layer Logic (i.e., beyond base I3C)

Debug "Extensions"



Common Logic per Network Adaptor "Extensions"

Defined by the Debug for I3C specification

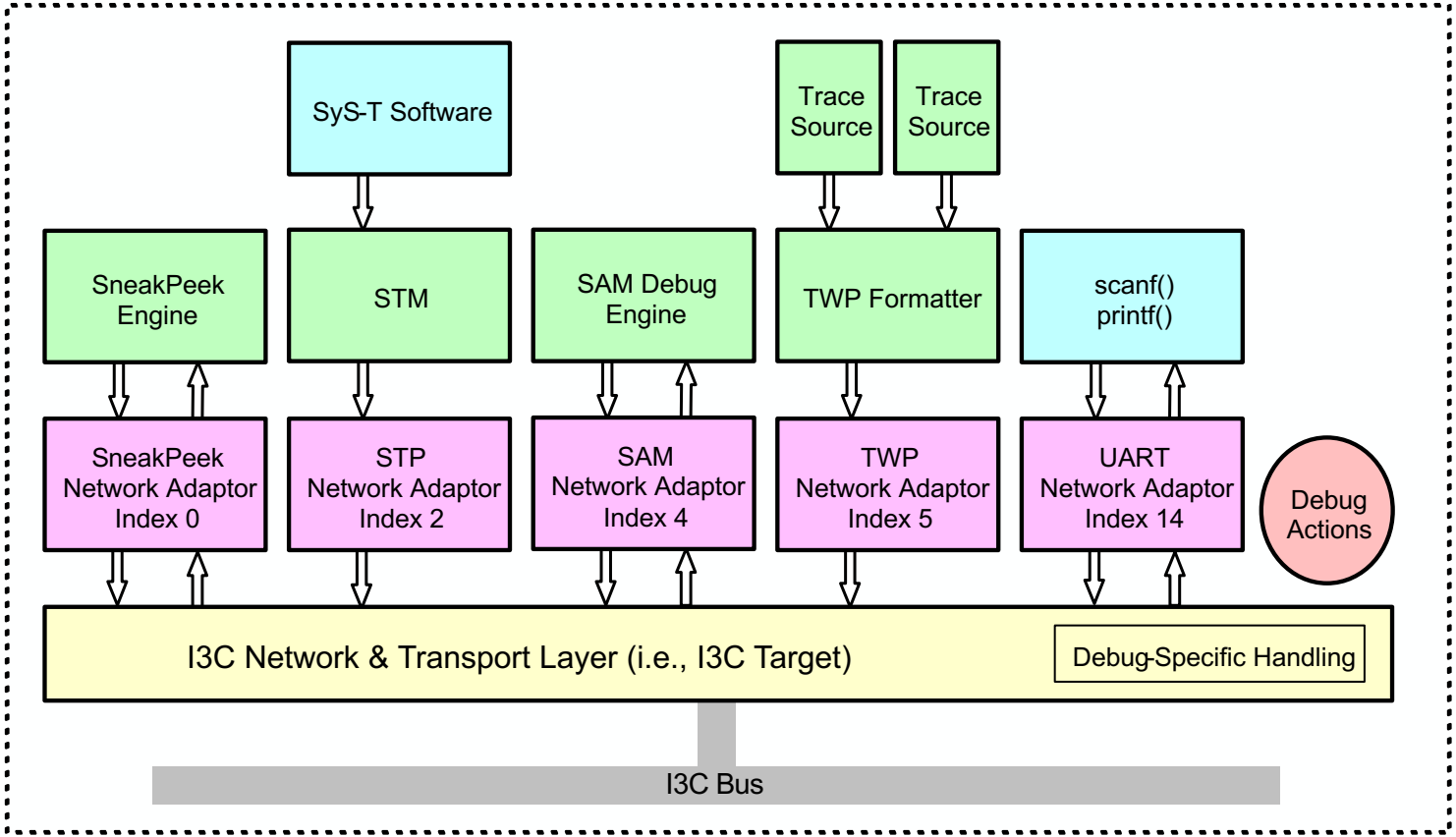
Network Adaptor

- **A mechanism used for communicating with debug functions within the TS**
- **Contains either an Inbound data pipe, an Outbound data pipe, or both**
  - Accessed via I3C Private Write and Private Read transfers
- **Mapped to a single instance of a debug function with a particular protocol**
  - Maximum 16 instances
  - Each Network Adaptor does not have to use a unique protocol

Supported Network Adaptor Protocols

Protocol	Direction	Details
SneakPeek Protocol (SPP)	Inbound & Outbound	Used for debug communication to a SneakPeek Command Engine. Bi-directional transfer of blocks of bytes (SPTBs) formatted to TinySPP
Trace Wrapper Protocol (TWP)	Outbound	Used for trace data output from trace infrastructure. Uni-directional transfer of a stream of bytes formatted to TWP
System Trace Protocol (STP)	Outbound	Used for trace data output from a System Trace Macrocell. Uni-directional transfer of a stream of bytes formatted to STP
Simplified Address-Mapped Protocol (SAM)	Inbound & Outbound	Used for communication with simple address-mapped access to TS resources
UART	Inbound & Outbound	Used for communication with character-oriented software agents such as `scanf()`/`printf()`, or a GDB monitor. Bi-directional transfer of bytes
Implementation-Defined	Inbound & Outbound	Implementation-defined function not described in this specification

Example Implementation of Network Adaptors



Network Adaptor Details: SPP

Protocol	Direction	Details
SneakPeek Protocol (SPP)	Inbound & Outbound	Used for debug communication to a SneakPeek Command Engine. Bi-directional transfer of blocks of bytes (SPTBs) formatted to TinySPP

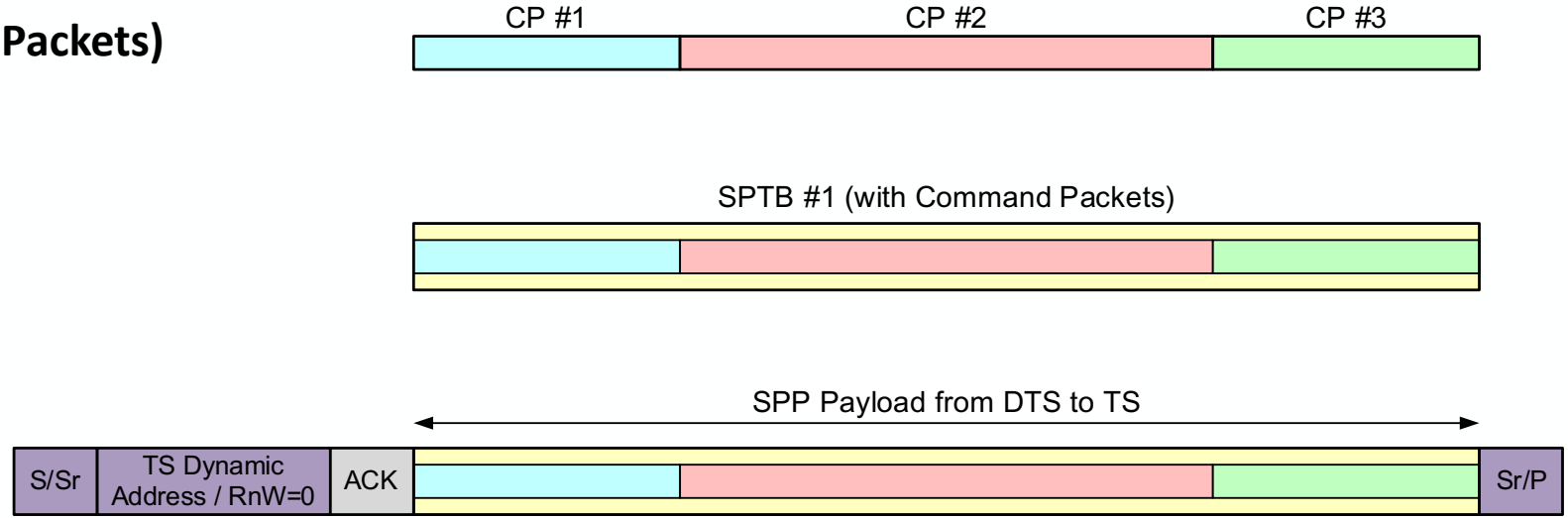
SPP provides a richer address-mapped mechanism for read+write transactions, to replace dedicated interfaces (such as JTAG).

Each **SneakPeek Transfer Block** (SPTB) is a single I3C Private Write or Private Read transfer.

Reference: https://www.mipi.org/specifications/spp

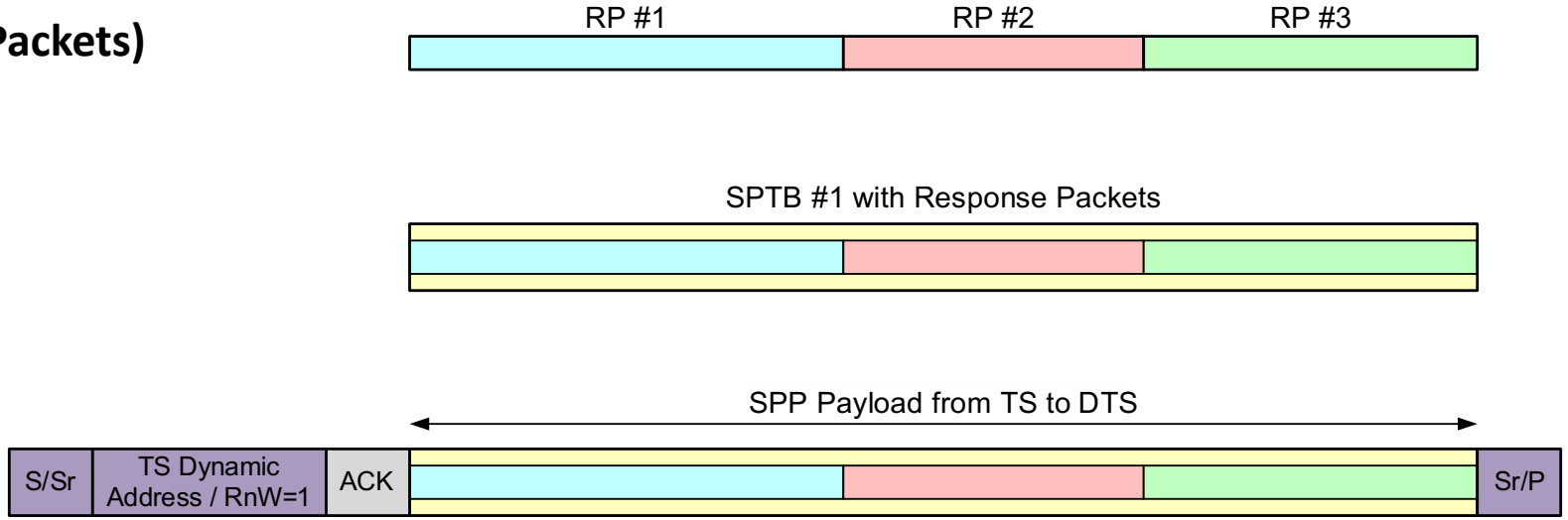
Network Adaptor Details: SPP

SPP write: SPTB (Command Packets) DTS → TS



Network Adaptor Details: SPP

SPP read: SPTB (Response Packets)
DTS ← TS



Network Adaptor Details: TWP

Protocol	Direction	Details
Trace Wrapper Protocol (TWP)	Outbound	Used for trace data output from trace infrastructure. Uni-directional transfer of a stream of bytes formatted to TWP

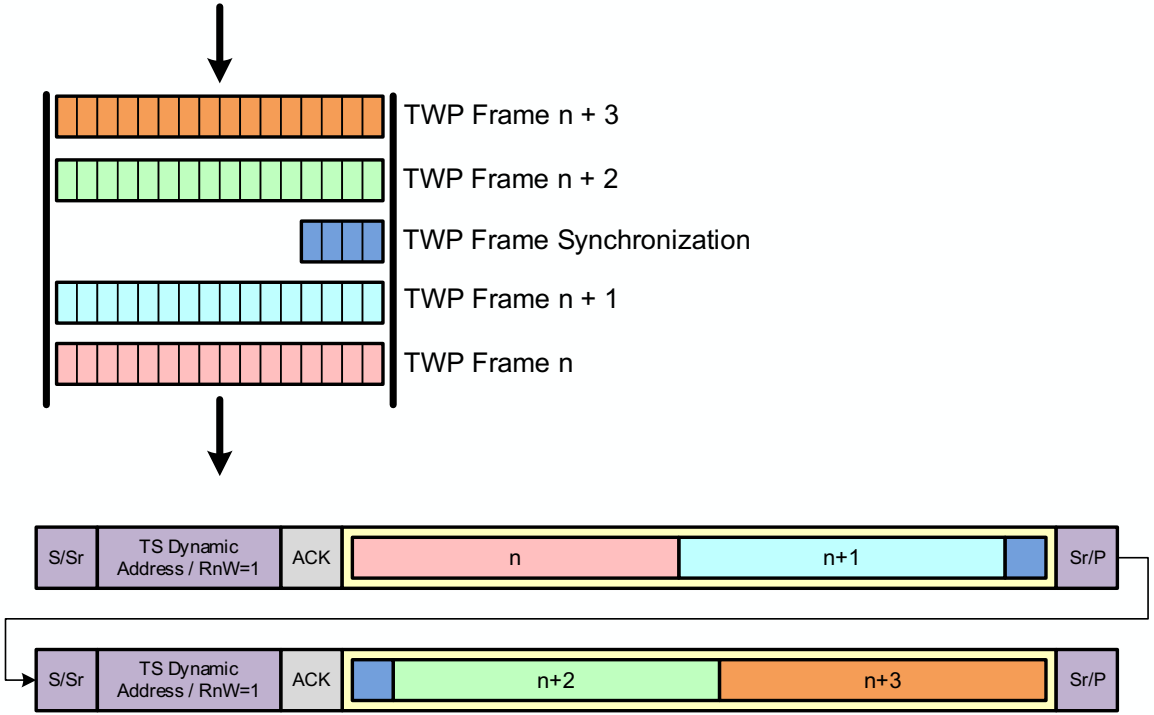
TWP enables trace data output from multiple trace sources.

TWP Frames can be spread across one or more I3C Private Read transfers.

Reference: https://www.mipi.org/specifications/twp

Network Adaptor Details: TWP

TWP read Frame(s):
DTS ← TS



Network Adaptor Details: STP

Protocol	Direction	Details
System Trace Protocol (STP)	Outbound	Used for trace data output from a System Trace Macrocell. Uni-directional transfer of a stream of bytes formatted to STP

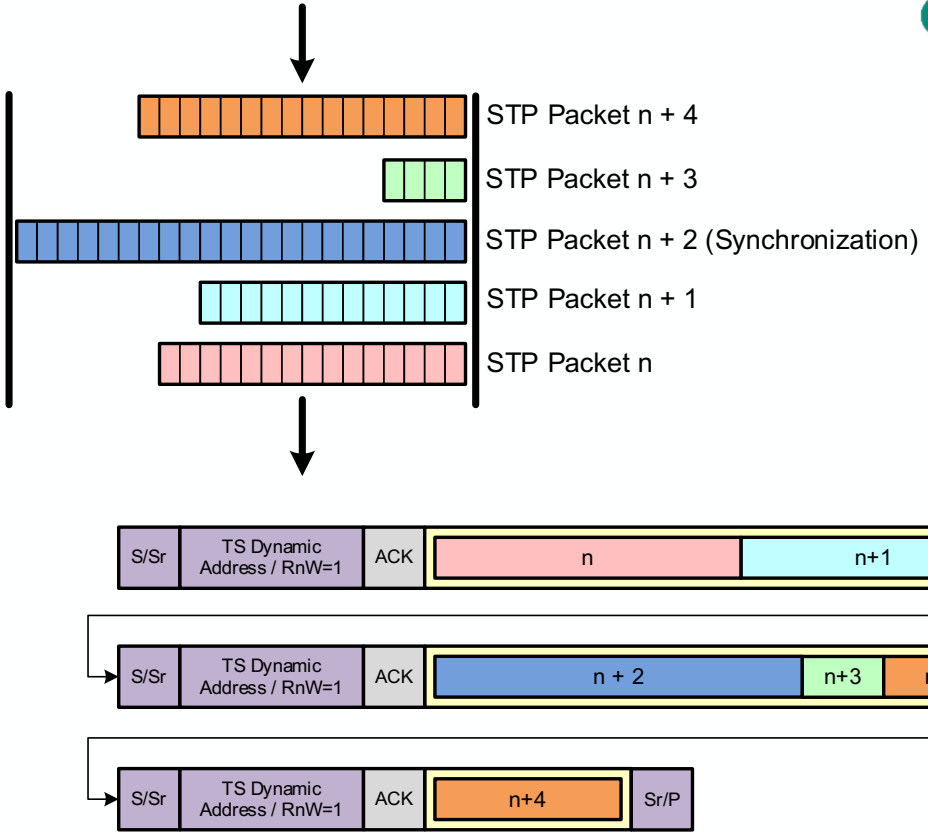
STP enables trace data output from application-specific trace protocols

STP Packets can be spread across one or more I3C Private Read transfers

Reference: https://www.mipi.org/specifications/stp

Network Adaptor Details: STP

STP read Packet(s):
DTS ← TS



Network Adaptor Details: SAM

Protocol	Direction	Details
Simplified Address-Mapped Protocol (SAM)	Inbound & Outbound	Used for communication with simple address-mapped access to TS resources

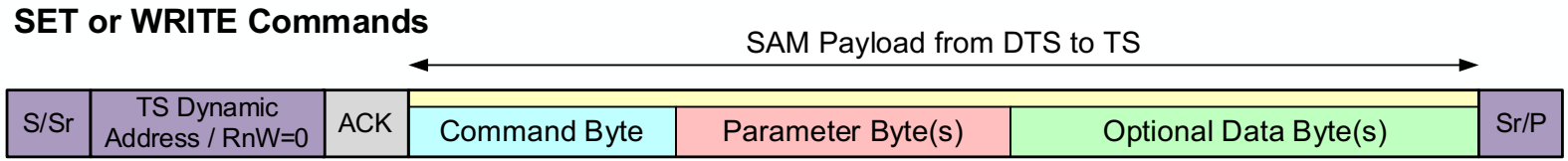
SAM provides a simple address-mapped mechanism for read+write transactions, intended for register or memory access.

SAM commands can be GET, SET, READ or WRITE.

- SET + WRITE commands use one I3C Private Write.
- GET + READ commands use "**Write**-then-**Read**" semantics: one I3C Private Write followed by one I3C Private Read.

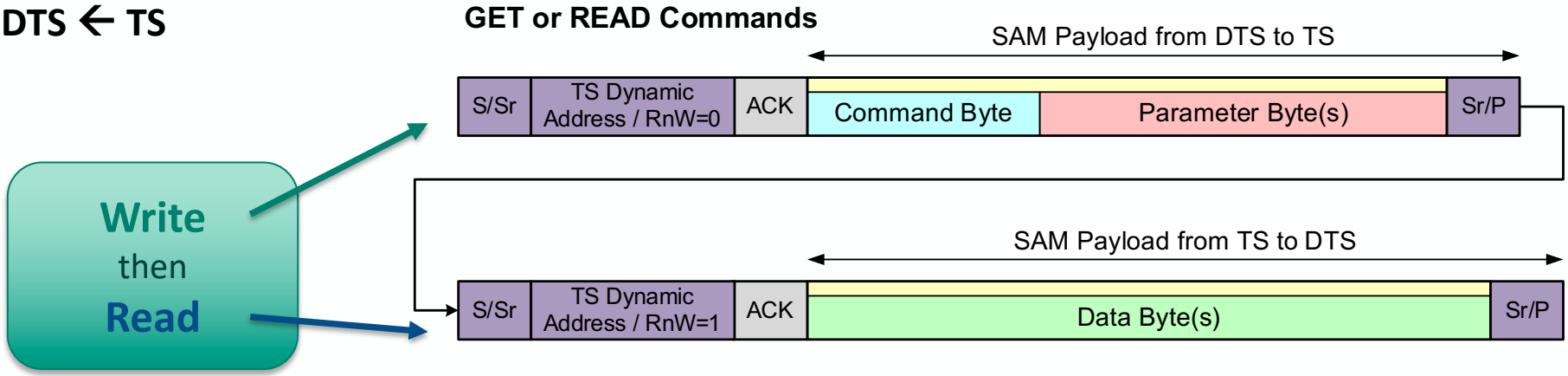
Network Adaptor Details: SAM

SAM SET or WRITE command example:
DTS → TS



Network Adaptor Details: SAM

SAM GET or READ command example: DTS ← TS



Debug Common Command Codes (CCC)

Defined CCCs specifically for use by the debug logic:

- **DBGOPCODE** – Direct CCC (0xD7)
  - Request a particular operation of a given Network Adaptor that is part of the TS.
  - Can be Write only, or **Write**-then-**Read**, or both (per opcode)
- **DBGACTION** – Direct (0xD8) or Broadcast (0x58)
  - Initiate one or more particular debug actions on a single TS (Direct) or all TS instances (Broadcast) on the I3C bus

Debug Common Command Codes (CCC)

- **DBGOPCODE** – Direct CCC (0xD7) opcodes:

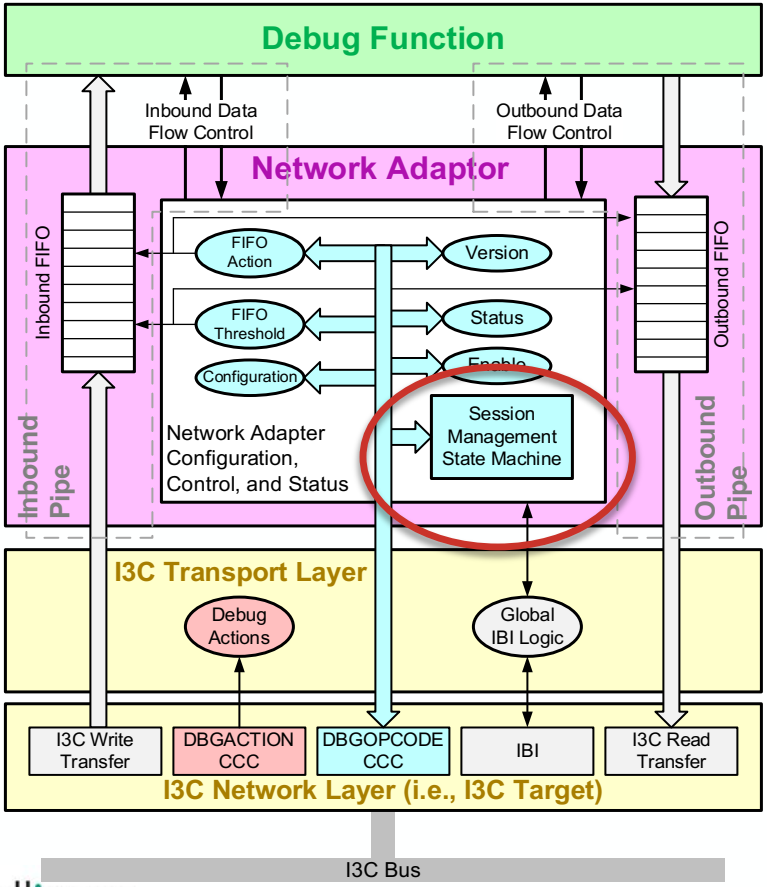
Opcode	Format	Purpose
CAPABILITIES	Read only (**Write**-then-**Read**)	Read Debug for I3C version and capabilities
CFG	Write / **Read**	Configure polling vs. interrupts
START_NA	Write / **Read**	Network Adaptor start session (and status)
STOP_NA	Write / **Read**	Network Adaptor stop session (and status)
FIFO_THRESHOLD	Write / **Read**	Network Adaptor FIFO threshold settings
FIFO_ACTION	Write / **Read**	Network Adaptor FIFO other actions
SELECT	**Write** only	Select Network Adaptor
REPORT_ERROR	**Write** only	Inform Network Adaptor of received error

Debug Common Command Codes (CCC)

- **DBGACTION** – Direct (0xD8) or Broadcast (0x58)
  - Build a set of actions, in continuous CCC framing

Value	Name	Purpose
0x00	DBGRST	Debug Reset
0x01	STARTSET	Start a set of actions
0x02	EXECSET	Execute the set of actions
0x03 – 0x7F	Reserved	-
0x80 – 0xFF	IMPDEF	Implementation defined

Network Adaptor Session Management



Concept of Session Management

- Each Network Adaptor is managed by its internal state machine
- DTS can start/stop transactions by using DBGOPCODE CCC

- *Specific meanings of TS state transitions will depend on the Network Adaptor type.*

Debug Resets

- **How to reset Debug logic in TS?**
  - DBGACTION CCC with ‘DBGRST’ value 0x00

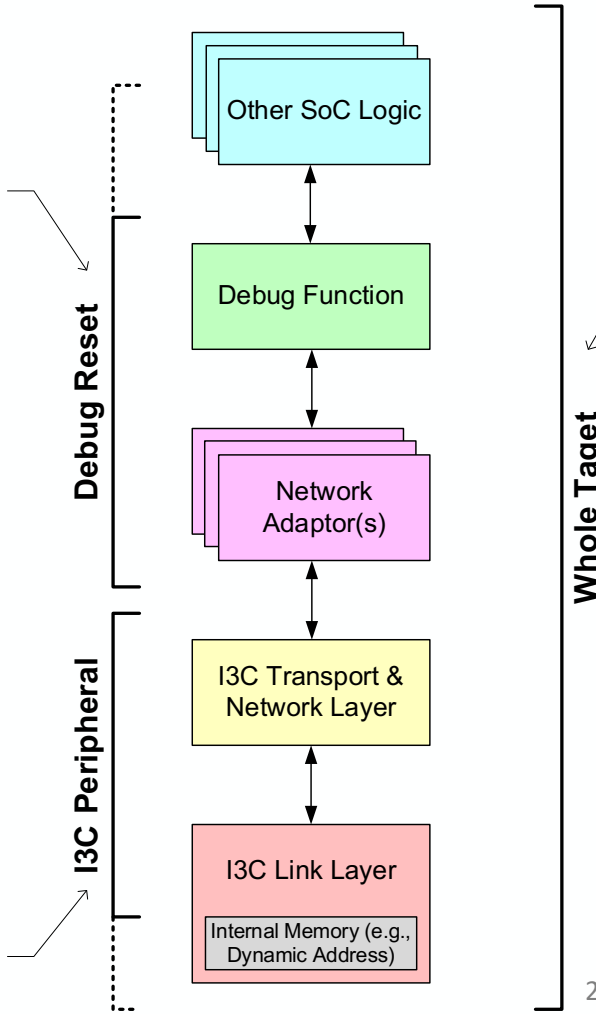
  or

  - RSTACT CCC (w/ Defining Byte 0x03) followed by **I3C Target Reset Pattern**

Debug Resets and Layers

Mechanism:

- DBGACTION CCC DBGRST Action (I3C v1.0 or later)
- Target Reset with RSTACT Defining Byte of 0x03 (I3C v1.1 or later)



Mechanism:

- Target Reset of Whole Chip (I3C v1.1 or later)

Mechanism:

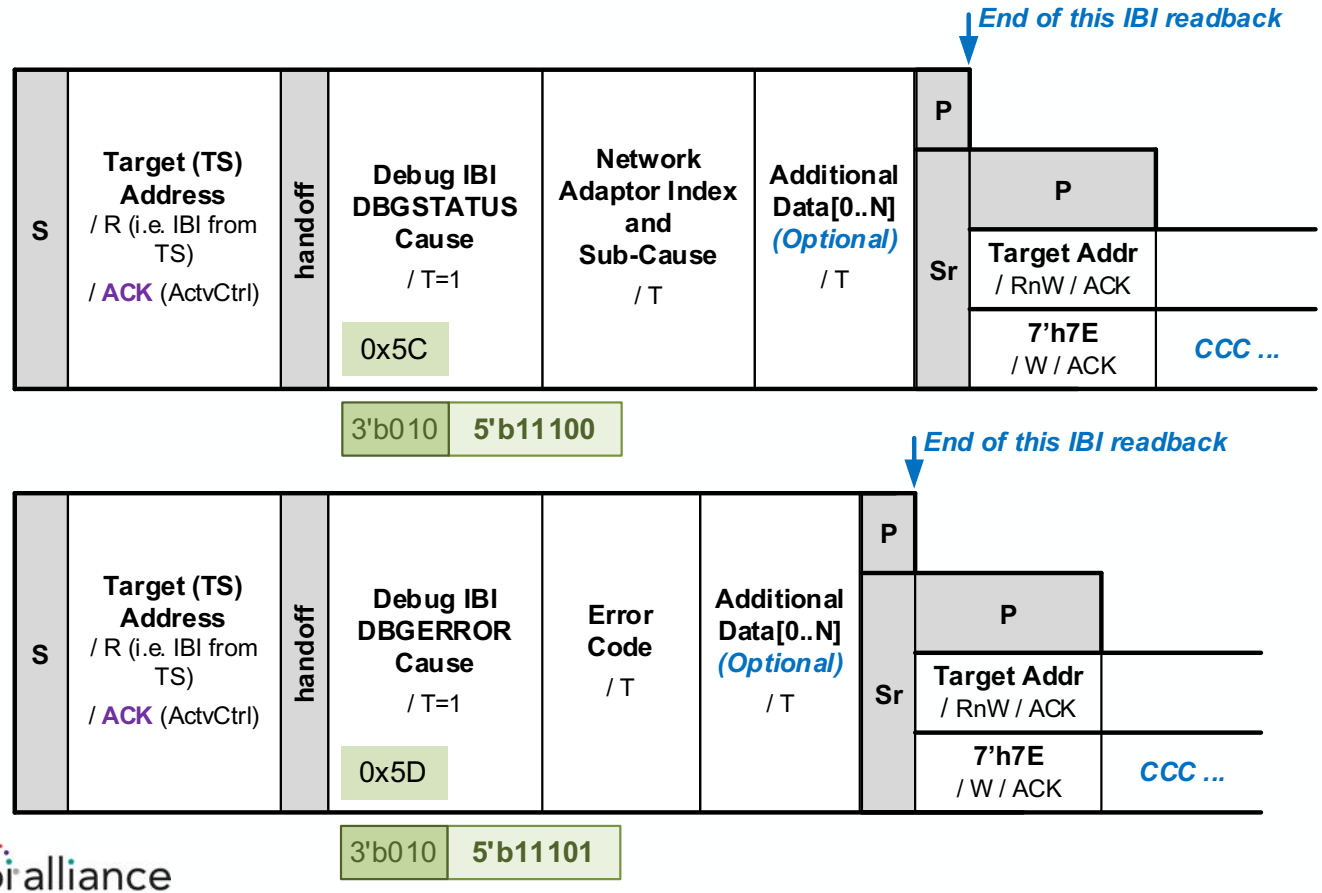
- Target Reset of I3C Peripheral (I3C v1.1 or later)

Debug In-Band Interrupts

Defined three mandatory data byte (MDB) identifiers specifically for debug events:

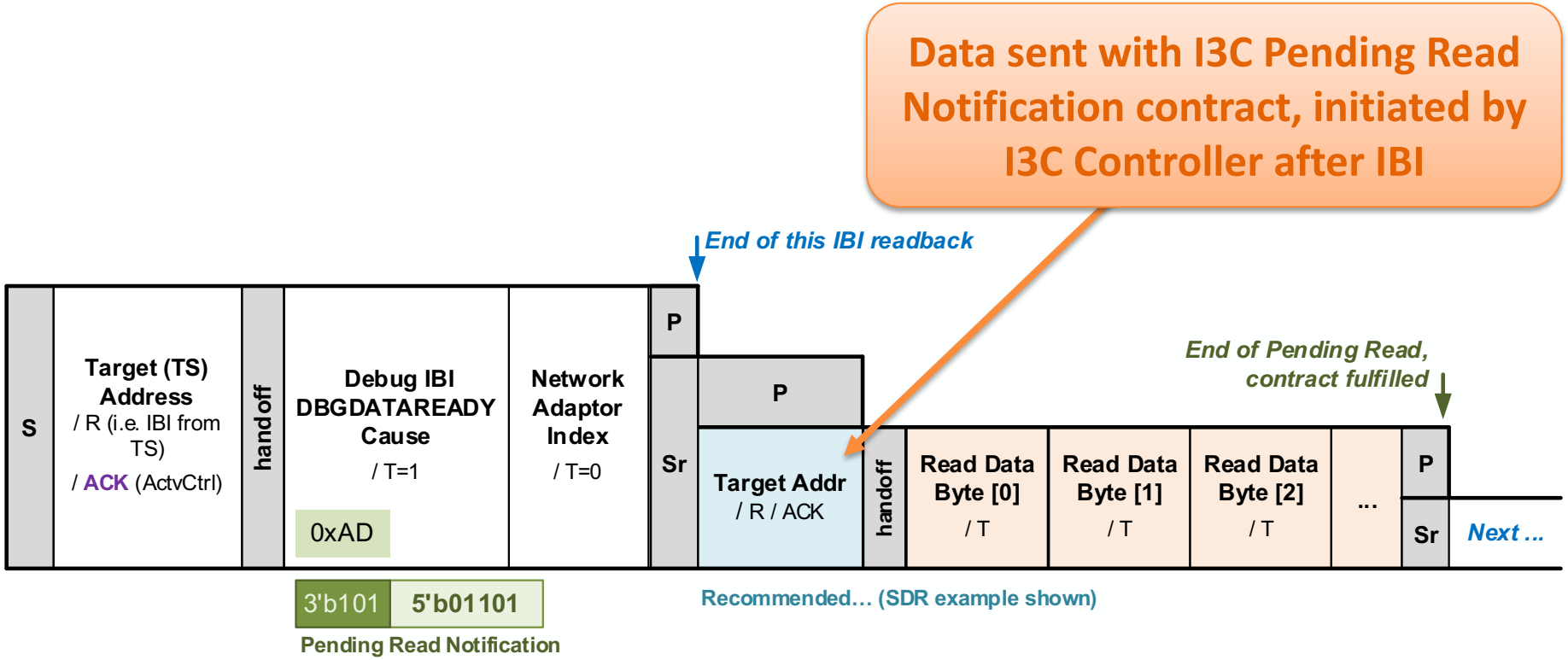
- **DBGSTATUS** – (MDB = 0x5C) Used to indicate a change in status
  - e.g., FIFO threshold met, session stopped, processor halted
- **DBGERROR** – (MDB = 0x5D) Used to indicate an error condition
  - e.g., I3C transport layer error or an error in the Network Adaptor
- **DBGDATAREADY** – (MDB = 0xAD) Used to indicate data is ready in a given outbound FIFO
  - A specific Pending Read Notification for debug data

Debug In-Band Interrupts (example)



 |

Debug In-Band Interrupts (example)



Debug Ecosystem Components

- I3C Target interface extensions
  - Implementing support for Debug CCCs, IBIs and connection to debug functions via Network Adaptors
- I3C Controller interface to DTS Host
  - MIPI I3C HCI[SM] available for a standard Host Controller
  - USB-IF Device Class standard is in development…
- Debug connectors that include I3C serial pins
  - Not yet defined, stay tuned…

Conclusions

- MIPI I3C® provides a **scalable, multi-mastering debug interface** that can **connect power-managed components** on a platform.
  - It meets newer requirements/use cases that legacy interfaces (e.g., JTAG/cJTAG, I^2C, UART) do not.
- The MIPI Debug for I3C specification **extends the I3C bus interface** for debug:
  - Builds on the core I3C capabilities, including two wires, hot-join, IBI, multi-drop, broadcast messaging and multi-mastering
  - Adds debug/test-specific CCCs and standardizes the data exchange mechanisms

For more information…

- MIPI Alliance website: http://mipi.org
- MIPI Debug WG page: https://www.mipi.org/specifications/debug
- MIPI Architecture Overview for Debug whitepaper: https://www.mipi.org/sites/default/files/mipi_Architecture-Overview-for-Debug_v1-2.pdf
- MIPI I3C page: https://mipi.org/specifications/i3c-sensor-specification
- MIPI Debug for I3C page: https://mipi.org/specifications/debug-i3c

Questions and Answers

 |

Thank you!

Backup Slides

Network Adaptors: Key Differences

Capability	SneakPeak (TinySPP)	SAM	UART
Separate debug functions per Adaptor	Up to 8	Up to 16	1
Function discovery and identifiers	Yes	Yes	No
Function handling	Multiple methods	Address/Data only	N/A
Addressable space	64-bit	32-bit	N/A
Commands provided	Rd, Wr, Wr+Rd, Loops, Triggers	Rd, Wr	N/A
Largest transfer size (bytes)	2^7 - 1	2^16 - 1	N/A
Address pointers (per function)	Yes	Optional	N/A
Many more advanced capabilities	Yes	No	No

Debug Ecosystem

- DTS connections via USB
  - Developing new USB-IF Device Class for I3C Controller
  - Work is in progress…