mipi alliance

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

A Beginner's Guide to MIPI Debug Solutions

Enrico Carrieri

Principal Engineer, Intel Corporation

& Chair of MIPI Debug Working Group

 |

Webinar Agenda

Introduction to MIPI Alliance

- **Peter Lefkin,** *MIPI Alliance Managing Director*

MIPI Debug Working Group Focus and Goals

Approach to Specifications

Debug System Framework and Solution Stack

Overview of Available Specifications

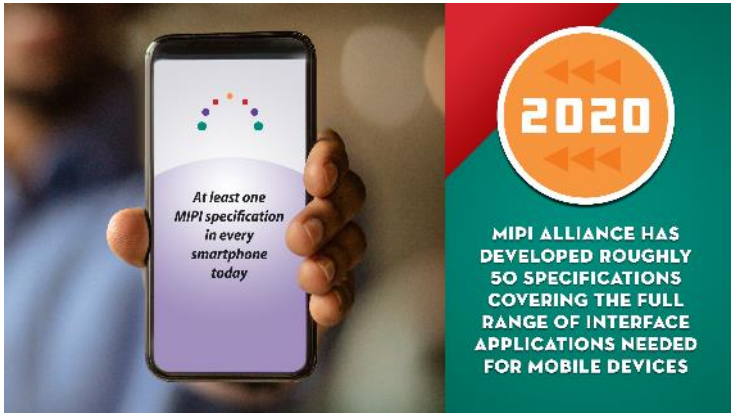
Summary and Conclusions

Introduction to MIPI Alliance

Peter Lefkin

Managing Director, MIPI Alliance

About MIPI Alliance



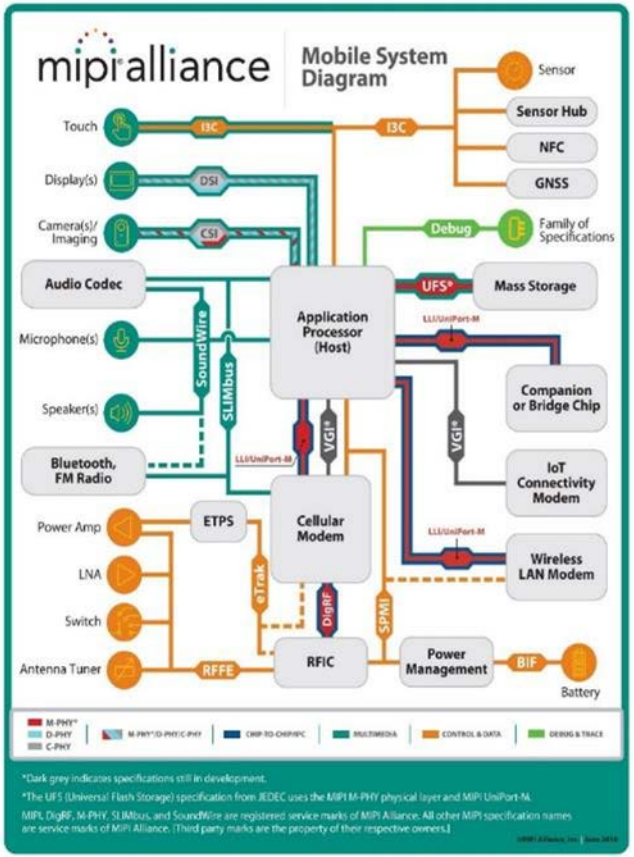
TODAY'S MIPI MEMBER ECOSYSTEM

Board and Contributor Members

Contributor Members

Board Members

MIPI Specifications Leveraged Beyond Mobile



50

Number of current specifications

Fundamentally, usage rights are granted to members royalty free for implementation of MIPI specifications from all MIPI members

mipi alliance

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

A Beginner's Guide to MIPI Debug Solutions

Enrico Carrieri

Principal Engineer, Intel Corporation

& Chair of MIPI Debug Working Group

MIPI Debug Working Group - Focus

- **The focus of the group is to unify/define:**
  - Protocols that support debug/trace:
    - With a particular focus on highly integrated, fielded systems
    - Software layers to support and/or implement these protocols
  - Configuration/control mechanisms required directly by debug/trace protocols
  - Reuse of functional interfaces and protocols for debug/trace
  - Mating connections and pin assignments
  - Electrical characteristics

MIPI Debug Working Group - Goals

- **Enable the best system debug support to:**
  - Reduce development and implementation/usage cost
  - Provide interoperability by defining standards
  - Provide hardware and software interfaces interacting with / supporting system debug
  - Leverage functional interfaces to increase debug visibility in fielded systems
- **Supply benefits for manufacturers as well as for users by:**
  - Identifying the gaps in current non-MIPI and MIPI standards
  - Developing recommendations and proposing solutions
- **Cooperate with other MIPI working groups and other industry bodies when it is in support of our focus and goals**

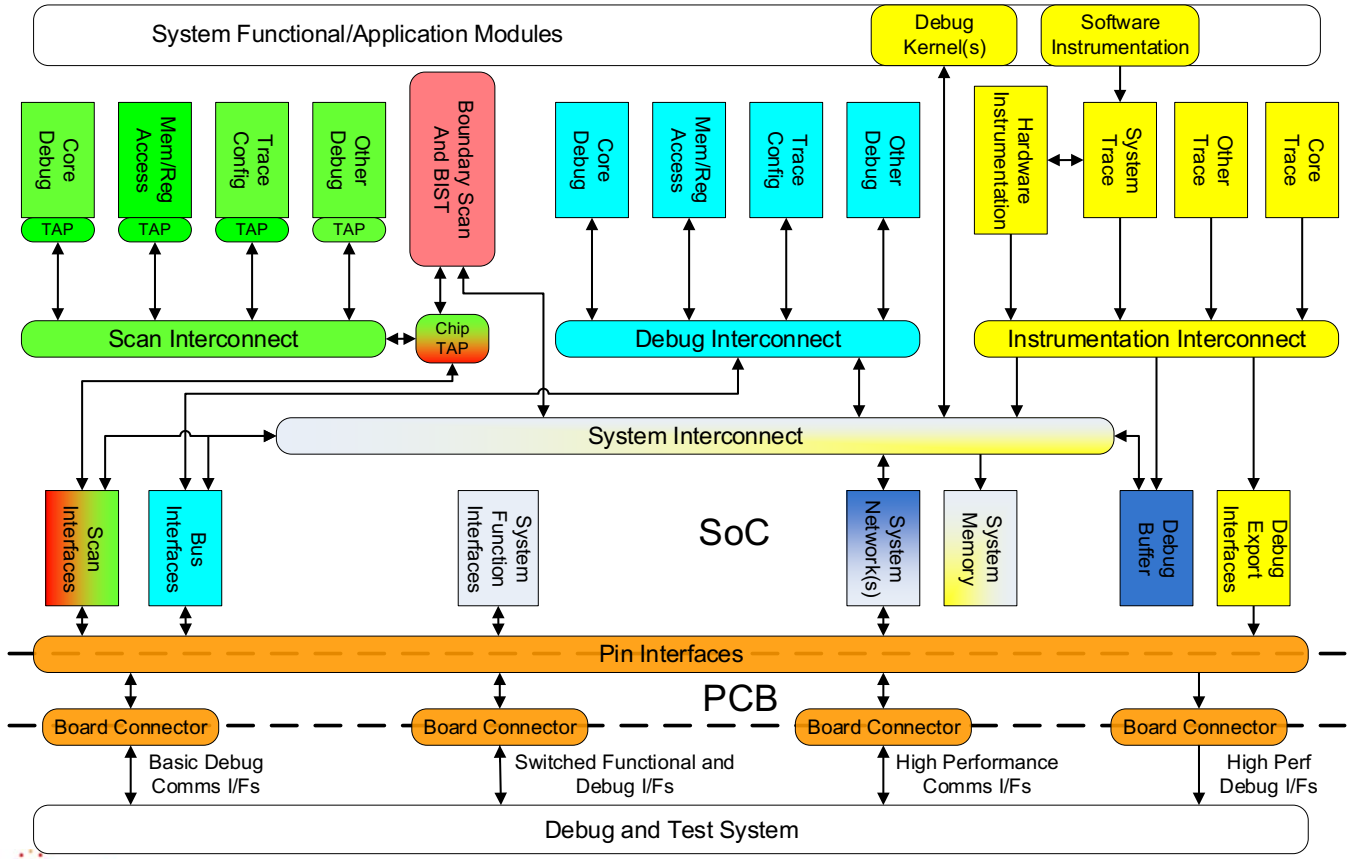
“Layered” Approach to Specifications

- **The set of Debug specifications provide a cohesive solution across the platform subsystems and across the solution stack**
- **Implementers are not required to implement all the specifications to get value from a given specification:**
  - Implement what you need…add more later
- **Specifications are focused on a particular “layer” or part of the platform solution:**
  - Allows flexibility and reuse of pieces of the solution
  - One solution at a given layer can be interchanged with another

Debug System Framework

- **MIPI Debug breaks down modern SoC debug architectures into the following major subsystems:**
  - Access and Control
  - Instrumentation and Visibility
  - Physical Interfaces
  - Network Interfaces

MIPI Debug Generic System Framework

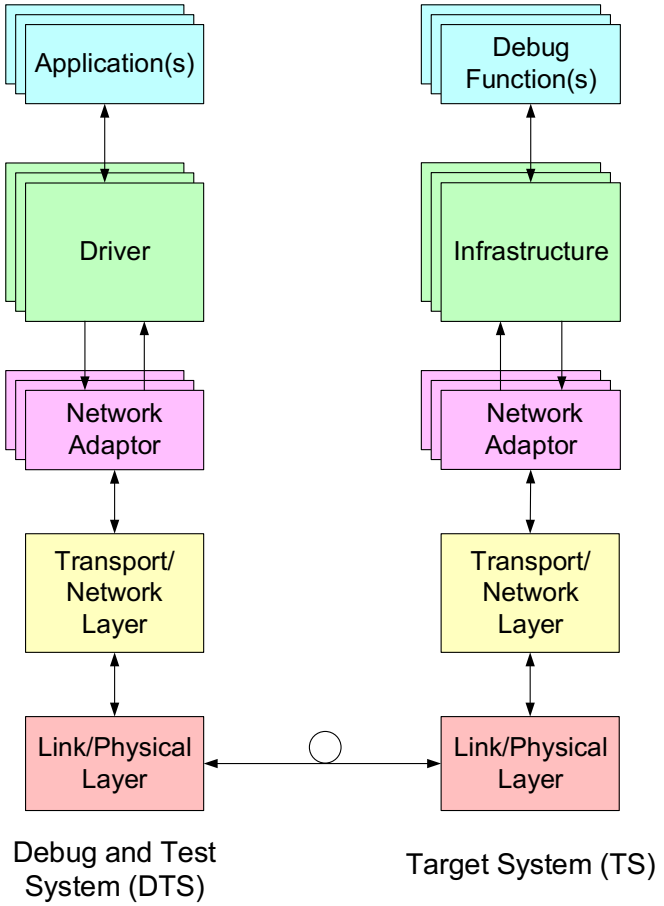


Legend

- System Test Subsystem (STS)
- Memory Mapped Debug Access and Control Subsystem (M-DACS)
- Scan Based Debug Access and Control Subsystem (S-DACS)
- Debug Instrumentation and Visibility Subsystem (DIVS)
- Debug Network Interfaces
- Debug Physical Interfaces

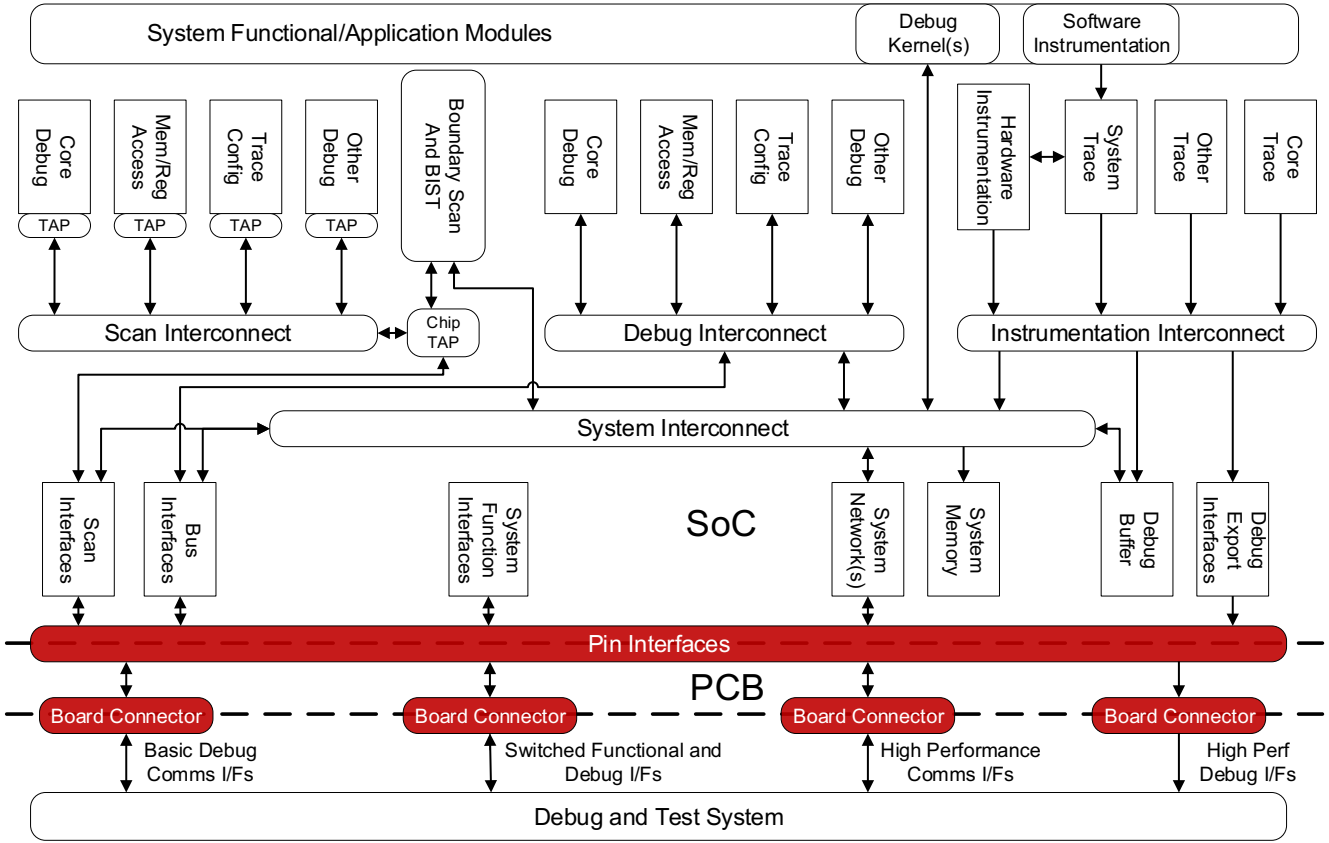
Debug System Stack

- **Debug specifications are focused on a particular layer or set of layers**
- **Allows a given specification to be flexible with other specifications focused on different layers to improve reuse and lower cost:**
  - e.g., specifications targeting the Network Adaptor can be used across different transports/networks



Debug Physical Interfaces (DPIs)

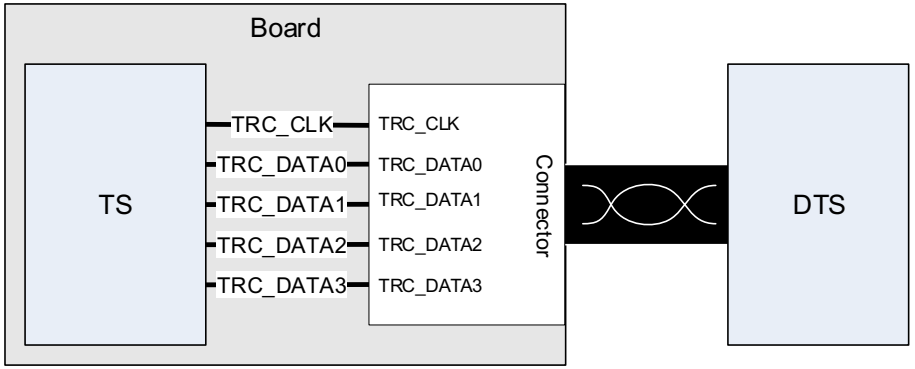
The physical interfaces that support debug at the SoC boundary and on the PCB



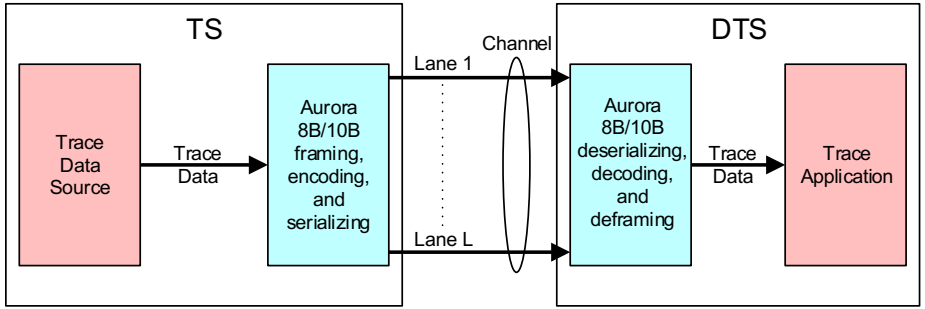
 |

DPI Specifications

- **Parallel Trace Interface (MIPI PTI℠)**
  - A low-level, parallel interface to export trace data in a bare-metal environment

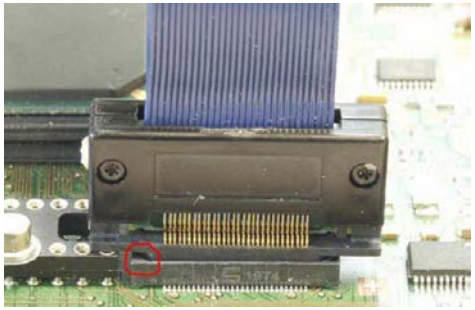


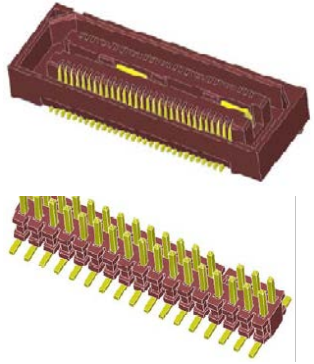
- **High-Speed Trace Interface (MIPI HTI℠)**
  - Serial implementation of PTI—using PHYs such as PCI Express®, DisplayPort™, HDMI®, or USB



DPI Specifications (cont.)

- **Debug Connector Recommendations**
  - Set of connectors and mappings to address debug use scenarios

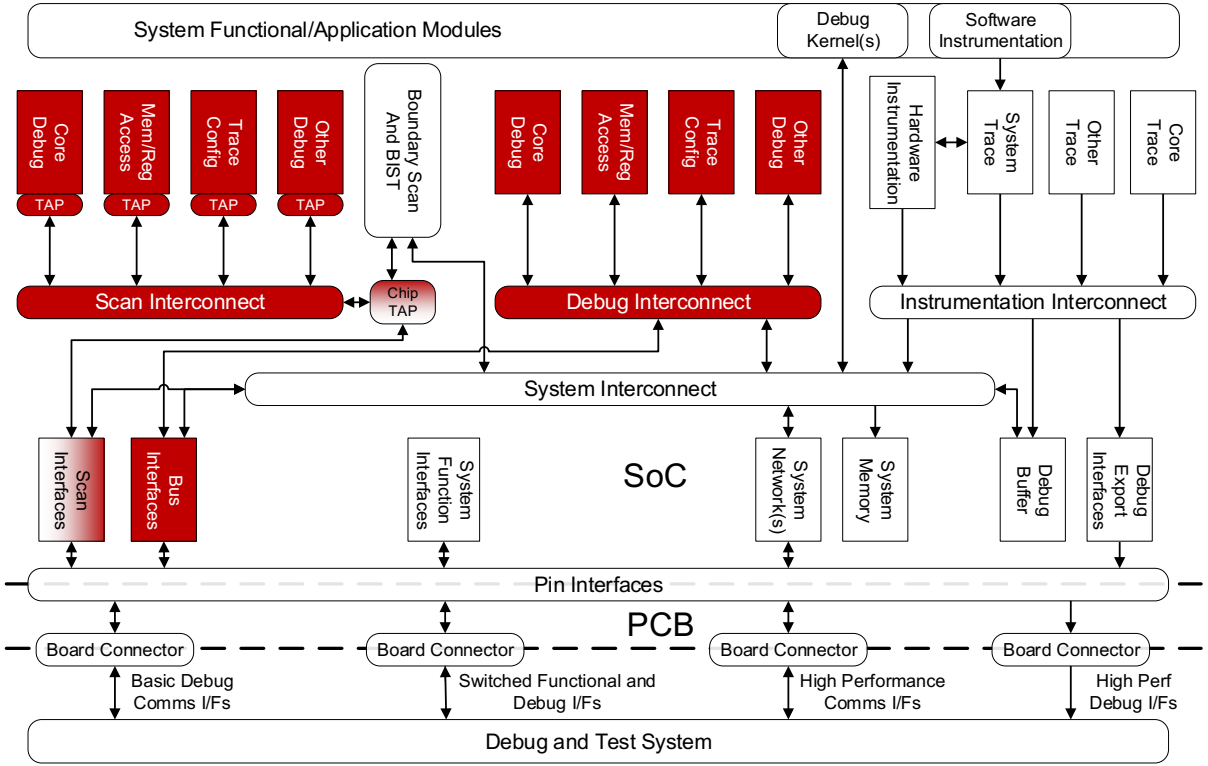




- **Narrow Interface for Debug and Test (MIPI NIDnT℠)**
  - Provides the means to use non-debug interfaces for debug purposes

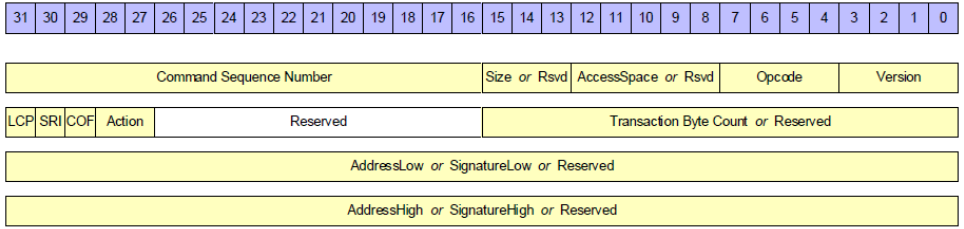
Debug Access and Control Subsystem (DACS)

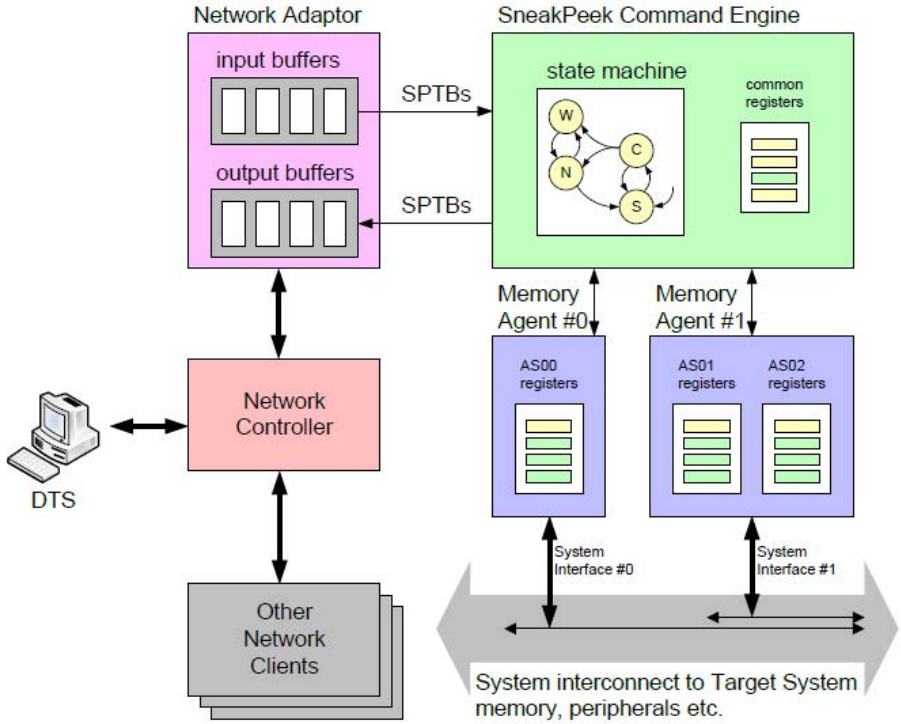
- Provides a path for the debug and test system to obtain direct access to application-visible system resources (e.g., registers and memory)
- Provides bidirectional communication for configuration and control of debug-specific modules in the target system



DACS Specifications

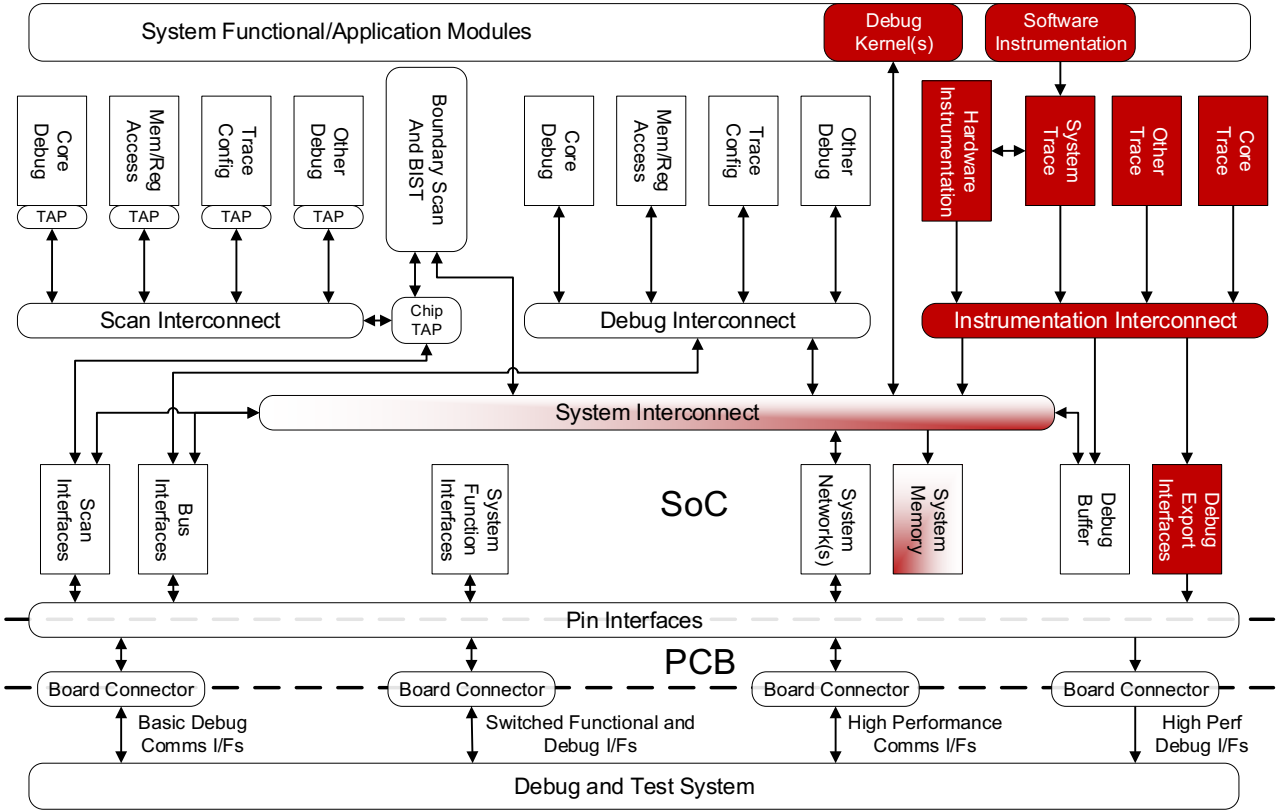
- **IEEE 1149.7**
  - While not a MIPI specification, this was started in MIPI to define a minimal pin interface (Advanced Protocol: TCKC & TMSC)
- **MIPI SneakPeek Protocol Specification (MIPI SPP℠)**
  - Abstracts the system designer from dedicated debug communication interfaces and replaces them with address-mapped read and write transactions to communicate with target system





Debug Instrumentation and Visibility Subsystem (DIVS)

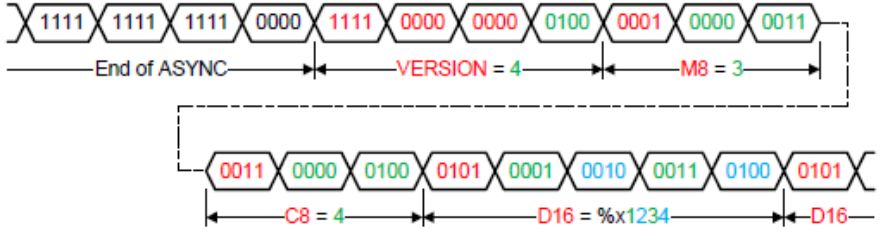
- **Provides communication and storage of data generated by debug instrumentation modules (e.g., processor and system trace) in the target system**



DIVS Specifications

System Trace Protocol (MIPI STP℠)

- Generic base protocol that can be shared by multiple, application-specific trace protocols



Trace Wrapper Protocol (MIPI TWP℠)

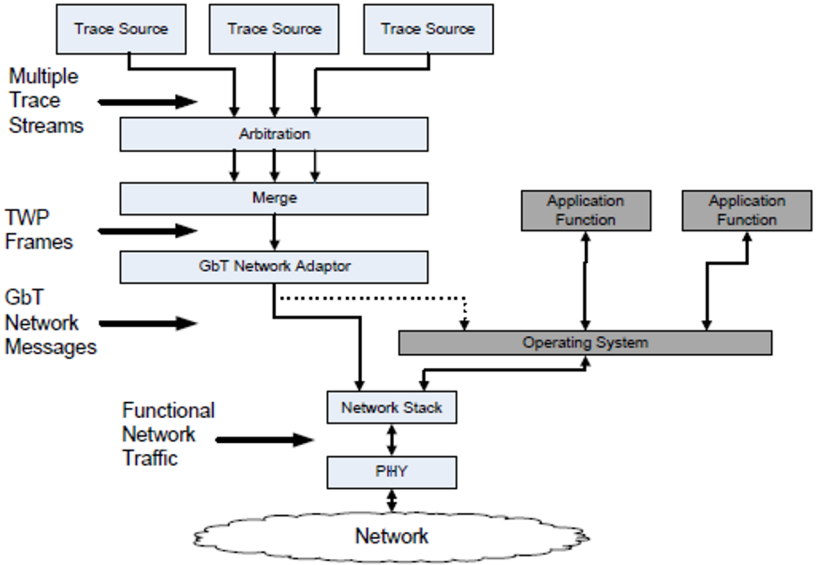
- Combines multiple source trace streams into a single trace stream by using system unique IDs

	31–24	23–17	16	15–8	7–1	0
Bytes 3 to 0	Data 0xA7	Data 0x53	0	Data 0xAA	ID 0x03	1
Bytes 7 to 4	Data 0x52	Data 0x2A	0	Data 0xA8	ID 0x15	1
Bytes 11 to 8	Data 0xCA	ID 0x03	1	Data 0x54	Data 0x29	0
Bytes 15 to 12	0 0 0 1 1 1 0 0	Data 0x64	0	Data 0xC7	Data 0x63	0

DIVS Specifications (cont.)

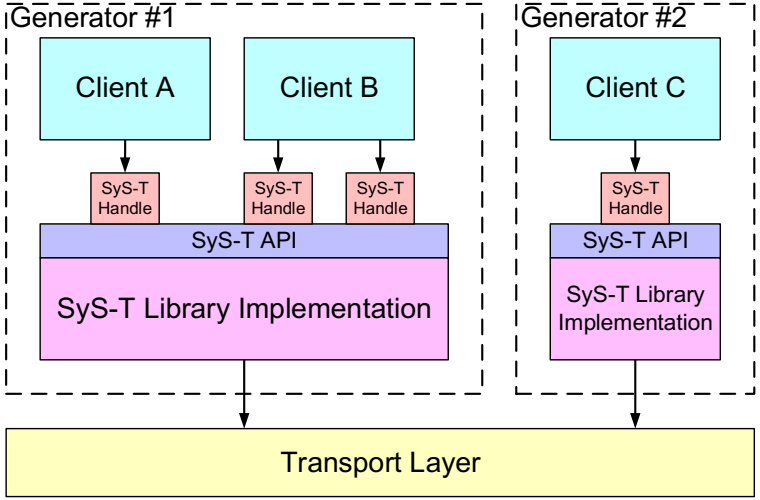
Gigabit Trace (MIPI GbT℠)

- Packages trace data as a stream of network messages suitable for transport over a shared network and/or interconnect



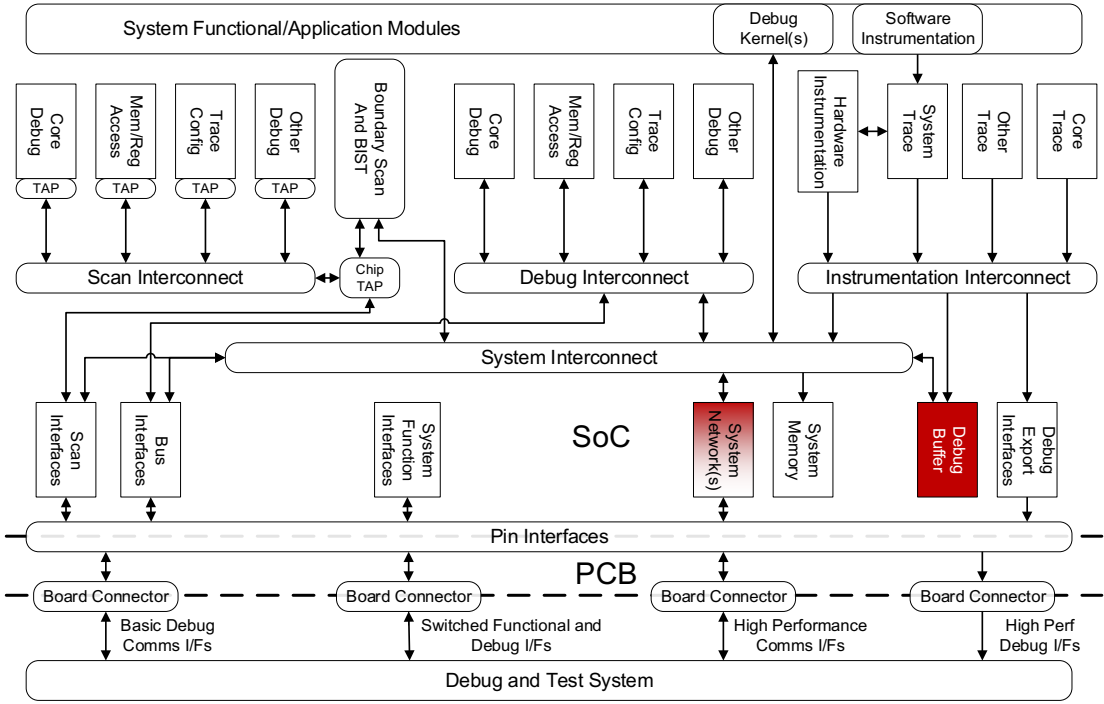
System Software Trace (MIPI SyS-T℠)

- An OS-independent software tracing protocol used primarily on a target system running embedded software



Debug Network Interfaces (DNIs)

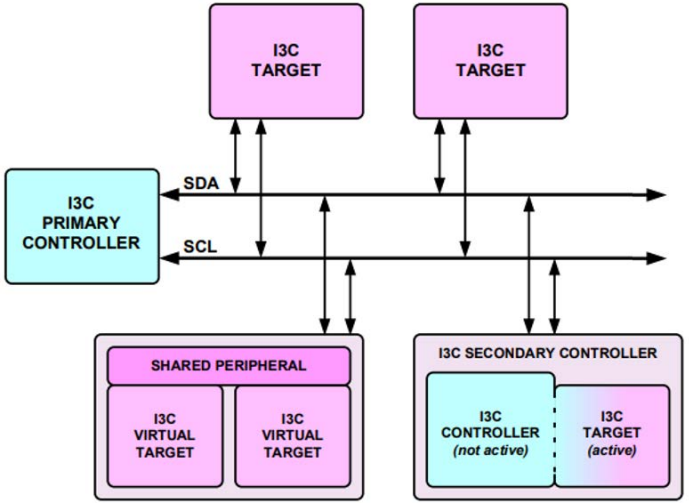
- **Interfaces that allow debug and trace data to be transmitted to and from the debug and test system on functional (non-debug) networks**
  - Usually with dedicated intelligent resources (sometimes called the Debug Butler)
- **DIVS protocols (e.g., SneakPeek and Gigabit Trace) over network can co-exist with other network traffic (as normal application layer functions)**



DNI Specifications

- **Gigabit Debug (MIPI GbD℠)**
  - For USB
  - For Internet Protocol Sockets (IPS)

- **MIPI Debug for I3C℠**
  - Bare-metal, minimal-pin interface for transporting debug controls and data between a debug and test system and a target system
  - **Upcoming webinar: A Deep Dive into MIPI Debug for I3C**
  - 17 February 2021, 8:00 AM – 9:00 AM PST
  - Learn more about the webinar »



Summary and Conclusions

- **MIPI Debug Working Group has a set of specifications ranging across the different "layers" of the stack:**
  - Done to reduce effort, maximize reuse, and allow individual pieces to be implemented (not all or nothing)
- **The set of specifications provides scalable solutions:**
  - Simple ASICs to complex SoCs
  - Single to multi function
  - Top to bottom of the solution stack
- **The specifications work together and can be used in part with other pieces (i.e., non-MIPI)**

For More Information…

- **MIPI Alliance Website:**
  http://mipi.org
- **MIPI Debug Public Page:**
  https://www.mipi.org/specifications/debug
- **MIPI Architecture Overview for Debug:**
  https://www.mipi.org/sites/default/files/mipi_Architecture-Overview-for-Debug_v1-2.pdf

Questions and Answers

