Outline

• About MIPI Alliance
• Introduction to MIPI I3C<sup>SM</sup>
• Usages beyond sensing
  – MIPI Camera Control Interface (CCI<sup>SM</sup>)
  – MIPI Touch over I3C<sup>SM</sup>
  – MIPI Debug for I3C<sup>SM</sup>
• MIPI I3C<sup>SM</sup> feature descriptions
• Implementation guidelines
  – Legacy Device Support
  – HDR Modes
  – Timing Control
  – Varied Topologies
• Summarized good design practices
About MIPI Alliance

Peter Lefkin
MIPI Alliance Managing Director
About MIPI Alliance

MIPI is a global, collaborative organization founded in 2003 that comprises 300+ member companies spanning the mobile and mobile-influenced ecosystems.

MIPI’s mission:
To provide the hardware and software interface specifications device vendors need to create state-of-the-art, innovative mobile-connected devices while accelerating time-to-market and reducing costs.
Who Our Members Are

- Device OEMs
- Handset Manufacturers
- IP and VIP Providers
- Test Labs
- Test Equipment Companies
- Semiconductor Companies
- Software Providers
- Application Processor Developers
- Consumer Electronics – Camera, Tablet, PC/ Laptop, Peripherals
What Does MIPI Alliance Do?

- **Define and promote specifications** focusing on the mobile interface but applicable to IoT, Auto, wearables, etc.
- **Complement existing standards bodies** through collaboration
- **Provide members with access to licenses** as needed to implement and market specified technologies
- **Promote member companies’ brands** through promotion, public relations, tradeshows, events and speaking opportunities
Mobile & Mobile-Influenced Markets

MIPI’s focus has always been on mobile. In fact, every smartphone on the market today has at least one MIPI specification.

With the development of new mobile-influenced markets, you can now find MIPI specifications in a variety of products:
A System of Mobile Interfaces

To date, MIPI has developed more than 45 specifications. Our leading specifications:

- CSI, CCS
- RFFE
- DSI
- I3C
- SoundWire
- UniPro

Physical layers:

- C-PHY
- D-PHY
- M-PHY

This diagram is represents a potential system – MIPI Alliance does not define or recommend a particular architecture.
Introduction to MIPI I3C℠

Ken Foust, Intel Corp.
MIPI Sensor WG Chair
MIPI I3C<sup>SM</sup> for Ubiquitous Low Speed Interfacing

- Anywhere sensors are used, MIPI I3C<sup>SM</sup> belongs
- Aimed toward historical I<sup>2</sup>C, SPI and UART applications in...
What is MIPI I3C℠?

- Innovative new 2-Wire Sensor interface
- Key features address historical pain points
  - In-band Interrupt, Dynamic Addressing, Multi-Master, Standardized Commands, Time Control, Hot-Join, Error Detection and Recovery
  - Plus...

![Diagram showing MIPI I3C℠ components and features]
Sensor Proliferation

Sensors Trends for Handsets

- Accelerometer
- Magnetometer
- Gyro
- Ambient Light
- Proximity
- Pressure
- Touch
- Haptics
- Fingerprint
- Health
- Environmental
- UV & RGB
- Humidity
- Microphone

- Motion characterization
- Contextual awareness
- Personal health & fitness
- Smart TV remote controller
- Gestures
- Sensor-aided navigation
- Biometrics & fingerprint wakeup/unlock
- Heart rate & blood sugar monitors
- CO / pollutant detector
- Multi-media
- video/camera/audio
- Touch detection
- Haptic feedback
- Building floor detection

Source: MIPI_Sensor_Interface_Bof_telco_14sep2012_v0.4.pdf
MIPI I3C℠ Vision

Too Many I/Os!
Fragmented Interfaces!
Usages Beyond Sensing

- **MIPI CCI**\textsuperscript{SM} (Camera Control Interface) over I3C\textsuperscript{SM} offers faster, lower latency and more efficient camera control with future abilities to support grouped data write and AON imaging.

- Current touch screen controller interfaces are fragmented.
- For many touch screens, MIPI Touch over I3C\textsuperscript{SM} presents a converged interface option for processed and raw touch data, leveraging IBI and HDR modes.

- MIPI Debug for I3C\textsuperscript{SM} offers a more complete closed chassis, scalable and power aware platform debug capability with minimum boundary pin count.
MIPI I3C℠ Features

- **I3C SDR – The Base Interface**
  - Up to 12.5 MHz $I^2C$-like clocking with defined Open-Drain / Push-Pull
  - Supports multiple classes of Devices
    - I3C Main Master
      - SDR-only Main Master
    - I3C Secondary Master
      - SDR-Only Secondary Master
    - I3C Slave
      - SDR-Only Slave
    - $I^2C$ slave
MIPI I3C℠ Features

• **SDR Dynamic Address Assignment**
  – Standardized procedure for dynamic assignment of 7-bit Addresses to all I3C Devices
    • I3C Slaves have two standardized characteristics registers and an internal 48-bit Provisional ID to aide the procedure
    – Legacy I²C Devices still use their static I²C Address

• **SDR In-band Interrupt**
  – Slave device can issue START Request when in “Bus Available” state
  – Master provides Interface Clock for Slave to drive it’s Master-assigned address onto the bus
  – Lowest assigned address wins arbitration in Open-Drain configuration
  – A data payload (i.e. Mandatory Data Byte) can immediately accompany the In-band Interrupt
**MIPI I3C<sup>SM</sup> Feature**

- **Hot-Join**
  - *Definition*: Slaves that join the Bus after it is already started, whether because they were not powered previously or because they were physically inserted into the Bus; the Hot-Join mechanism allows the Slave to notify the Master that it is ready to get a Dynamic Address.
  - Allows Slaves to join the I3C Bus after it is already configured
  - Ideal for I3C Devices:
    - Mounted on the same board, but de-powered until needed
    - Mounted on a module/board that is physically inserted after the I3C Bus has already been configured
  - *Note*: Such Devices shall not violate specified electrical limits and shall not disrupt the I3C lines during physical insertion
MIPI I3C™ Features

• **Error Detection and Recovery Methodology**
  – For Master and Slave generated errors (9 Error Types identified, Parity, CRC5)

• **Common Command Codes**
  – Standardized command mode with extensible set of MIPI-defined codes that can be Broadcasted and/or Directed, Read and/or Write

<table>
<thead>
<tr>
<th>S or Sr</th>
<th>0x7E / W / ACK</th>
<th>Command Code / T</th>
<th>Data (Optional) (Broadcast CCC only) / T</th>
<th>Sr or P</th>
</tr>
</thead>
</table>

  – Standardized Command Codes
  • Event Enable/Disable
  • Activity States
  • Payload Mgmt
  • I3C Feature Mgmt (Dynamic Address Assignment, Mastership, HDR Modes, Timing Control)
  • Test Modes
  • Extensible Space (MIPI and Vendor)

* Example of Broadcast CCC Frame
Guidelines - Legacy I²C Device Support

- Fm and Fm+ Speeds Supported
- 50ns Spike Filter ($t_{sp}$) Needed for 12.5MHz I3C℠ Clocking
- Clock Stretching is Not Allowed – I3C SCL is Push/Pull
- 20mA Open Drain Drivers ($I_{OL}$) are Not Used
- I²C Extended Addresses (10 bit) are Not Used

$t_{sp}$: pulse width of spikes that must be suppressed by the input filter

*UM10204: I2C-bus specification and user manual Rev. 6*
MIPI I3C™ Features

- **I3C High Data Rate (HDR) Modes**
  - Optionally supported beyond the base SDR mode: 12.5MHz, SDA/SCL
    - HDR-DDR: Double Data Rate
    - HDR-TSL/TSP: Ternary Symbol
  - Offer bit rates over 33Mbps at a fraction of the per bit power of I²C Fast Mode
  - Simple Slave-side digital implementations
  - Coexistent with legacy I²C Devices
  - Leverage rising and falling edges
  - Individually entered using broadcasted MIPI-defined Common Command Codes
  - Universally exited and restarted via MIPI-defined toggling patterns
  - Allows non-HDR I3C Devices to “ignore” HDR transmissions

<table>
<thead>
<tr>
<th></th>
<th>I3C</th>
<th>Msg1</th>
<th>Msg2</th>
<th>I3C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>START</td>
<td>Brdct CCC</td>
<td>EnterHDRx</td>
<td>HDR Cmd</td>
</tr>
<tr>
<td>SDA</td>
<td>Setup (SDA/SCL) (if needed)</td>
<td></td>
<td>End</td>
<td></td>
</tr>
<tr>
<td>SCL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Diagram:
- **HDR Exit**: Stop after stop.
- **HDR Restart**: Restart pattern after stop.
- **Possible Restart**: Restart if needed.
- **Next Edge**: Confirms next edge.

© 2018 MIPI Alliance, Inc.
**MIPI I3C℠ Features**

- **HDR-DDR: Double Data Rate**
  - Uses SCL as a clock, however Data and Commands change SDA on both SCL edges. By contrast, SDR Mode changes SDA only when SCL is Low.
  - HDR-DDR moves data by Words, which generally contains 2 preamble bits, 2 payload bytes and 2 parity bits. 4 Word Types defined: Command, Data, CRC, and Reserved.

- **Simple protocol:**
  - From Master to Slave
  - From Slave to Master
  - Transition Bit
  - (Parity Bit for CCC)

---

**Diagram Notes:**

- **S or Sr:** I3C Reserved byte (0x7E) (R/W=0)
- **ACK:** Enter HDR-DDR CCC (ENTHDR0)
- **T:** HDR-DDR Command
- **CRC:** HDR DDR Data (1 or more words)
- **P:** HDR DDR CRC
- **Preamble Bits:** Define the subsequent Word Types
- **Command, Data, or CRC:** Based on Preamble (2-bit MSB)
- **Parity Bits:** P1: Odd Parity bit, P0: Even Parity bit

---

**Parity Calculation:**

- **P1:** Odd Parity bit
- **P0:** Even Parity bit

**Word Types:**

- **S:** START Condition
- **Sr:** RESTART Condition
- **P:** STOP Condition
- **T:** Transition Bit Alternative to ACK/NACK
MIPI I3C℠ Features

- HDR-TSL/TSP: Ternary Symbol Coding
  - Ternary symbol coding for pure (TSP) and I²C legacy-inclusive (TSL) systems
  - Given a two-wire interface with ‘simultaneous’ transitions and no traditional clock, there are 3 possible symbols available – 0, 1, 2
  - At least one line must transition each period
  - Ideally, there are 3 possible “next” transition
  - Transition indices are used to efficiently encode Binary into Ternary
  - Simple protocol:

<table>
<thead>
<tr>
<th>I3C SDR</th>
<th>Msg1</th>
<th>Msg2</th>
<th>I3C</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>Brdcst CCC</td>
<td>EnterHDRx</td>
<td>HDR Cmd</td>
</tr>
</tbody>
</table>

- I²C and I3C SDR

- One symbol diagram
Guidelines - HDR Modes

- Enter HDR Commands Supported

<table>
<thead>
<tr>
<th>I3C</th>
<th>Msg1</th>
<th>Msg2</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>Brdct CCC</td>
<td>EnterHDRx</td>
</tr>
</tbody>
</table>

- HDR Exit Pattern detected by all I3C Devices

- Non-HDR Devices shall ignore I3C HDR bus traffic until the HDR Exit Pattern is detected
MIPI I3CSM Features

- Extensible Timing Control Framework – Synchronous and Asynchronous Solutions Provided
  - Synchronous Systems and Events
    - Slave sample synchronization to allow Master read within common time window
  - Asynchronous Modes – Interpolating unexpected time-of-occurrence
    - Basic: Mobile device targeted accuracy limited by each Device’s internal timer stability
    - Advanced: Higher time stamping accuracy by restricting Device timer runtime and introducing Master-generated periodic sync events.
    - High-Precision Low-Power: Highest precision through additional Master complexity by timestamping I3C SCL frequency changes.
    - High-Precision Trigger able: Supports precise time triggering of multiple sensors with precise time measurements
Guidelines - Timing Control

- Sensor/Device Clock Accuracy

- Support exchange of timing information via I3C XTIME CCCs
  - Clock/Oscillator Frequency and Error (variation)
Guidelines - Varied Topologies

• Impacts on signal transition/transit times (maximum bus frequency)
  – SDA/SCL drive strength: “weaker” for lower power and interference vs “stronger” for faster over larger topologies/loads
  – Trace length and material: short vs long and pcb vs cable
  – SCL/SDA pad capacitance
  – Clock to Data Turnaround Time ($t_{SCO}$)

• Legacy I²C Devices impact maximum bus frequency (MHz)
  – Must run I3C at speeds/pulses beyond Spike Filter or slow Bus to that of slowest I²C Device

• Impacts on signal integrity/reliability
  – Device Location: close and far Devices can cause interference from reflections
Summarized Good Design Practices

• Thoroughly understand capability of coexistent Legacy I²C Devices
  – 50ns Spike Filter
  – Disabled Clock Stretch

• If leveraging Timing Control, choose an approach that best matches application requirements with I3C Device capabilities
  – Internal oscillator/clock frequency and error

• Understand bus topology and performance tradeoffs Mixed (I3C and Legacy I²C Devices) vs Pure Bus (I3C Devices Only)
  – Trace length and material
  – SDA/SCL pad capacitance
  – Clock to Data Turnaround Time ($t_{SCO}$)
  – Device location
Any Questions?