MIPI VGI\textsuperscript{SM} for Sideband GPIO and Messaging Consolidation on Mobile System
Agenda

• The Problem Statement
• Virtual GPIO Interface (MIPI VGI\textsuperscript{SM} ) : Concept
• MIPI VGI\textsuperscript{SM} Architecture
• Application Scenarios
• Summary
• Q&A
Mobile Connectivity Expansion Trends

**Cellular**
- 2G/3G/4G ➔ LTE-Advanced ➔ 5G

**WiFi**
- 802.11a/b/g/n/ac ➔ ax
- 802.11ad/WiGig

**Video**
- VGA/SD/HD ➔ 4K ➔ 8K

**Docking**
- Charging/audio/video ➔ Productivity, Games and External Storage

**Mobile Influenced**
- Drones, IoT, Automotive, …
- CAT-1 to CAT-3 Low-Power LTE Modem Support
The Problem of Sideband Proliferation

- **Modem**
  - GPIOs: x9

- **Wireless LAN Bluetooth**
  - GPIOs: x5

- **Gigabit Wireless LAN (60-GHz)**
  - GPIOs: x5

**Applications Processor (SoC)**
- GPIOs: x6
- GPIOs: x8

**Companion / Bridge-Chip**
- Ethernet

Optional Connector
The Problem of Sideband Proliferation

Typical Sideband Utilization

<table>
<thead>
<tr>
<th>Domain</th>
<th>Number of Sideband I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera/Imaging</td>
<td>6 to 12</td>
</tr>
<tr>
<td>Audio CODEC</td>
<td>4 to 7</td>
</tr>
<tr>
<td>Cellular Modem</td>
<td>3 to 10</td>
</tr>
<tr>
<td>Wireless LAN Modem</td>
<td>3 to 10</td>
</tr>
<tr>
<td>Bridge Chip</td>
<td>3 to 8</td>
</tr>
<tr>
<td>Sensor Hub</td>
<td>4 to 18</td>
</tr>
</tbody>
</table>

Typical Sideband GPIOs: 23 to 65
MIPI VGI<sup>SM</sup>: Solution to Sideband Proliferation
MIPI VGI<sup>SM</sup>: The Concept

- **MIPI VGI** consolidates *N*-sideband GPIOs and sub-100 MHz serial messaging over 2 or 3 wire interface in a Point-to-Point configuration.

- **2-wire MIPI VGI**: Asynchronous, Full-Duplex (4-Mbps max.)

- **3-wire MIPI VGI**: Synchronous, Full-Duplex

- **MIPI VGI** Rev-1 (3-wire) Max Speed: 76.8 MHz

- **Consolidates Low Speed Messaging Interface and Sideband GPIOs** (*N*-pins to 2/3-pins reduction)
Limitation of Conventional Techniques

- HLOS processing latency varies widely
- Deep-sleep to active-state typical latency: Typically $\rightarrow$ 30 to 100 ms
- Timing uncertainty not suitable for the key IPC side-band signaling
MIPI VGI<sup>SM</sup> Architectural Block-Diagram
MIPI VGI\textsuperscript{SM} Physical Interface: 2-wire or 3-wire

1. Asynchronous MIPI VGI
   - Initial and Power State Transition mode communication over 2-wire, 4-Mbps max.

2. Synchronous MIPI VGI
   - Common clock (Up to 76.8 MHz in VGI Rev-1)
   - Sleep clock based operation supported in Low Power Modes
MIPI VGI\textsuperscript{SM} Techniques At-a-Glance

VGI-Techniques

E.g. Power-ON/Default mode Communication RO-PWM

3-Wire I/F

2-Wire I/F

Device #1

Device #2

Synchronous Mode

Device #1

Device #2

Device #1

Device #2

Device #1

Device #2

Asynchronous Modes

* Ring Oscillator based PWM, ** Phase-Modulated PWM
## MIPI VGI<sup>SM</sup> Roadmap

<table>
<thead>
<tr>
<th>#</th>
<th>VGI Features</th>
<th>VGI v1.0</th>
<th>VGI Next</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2-wire and 3-wire I/F support</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>Default PWM encoding</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>UART Encoding</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>PM-PWM Encoding (Phase-Modulated PWM)</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>5</td>
<td>2-wire mode max throughput</td>
<td>4 Mbps</td>
<td>8 Mbps (PM-PWM)</td>
</tr>
<tr>
<td>6</td>
<td>3-wire mode max throughput</td>
<td>76.8 Mbps</td>
<td>153.6 Mbps</td>
</tr>
<tr>
<td>7</td>
<td>1.2V, 1.8V Operation support</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>8</td>
<td>1-wire mode support</td>
<td>-</td>
<td>✓</td>
</tr>
</tbody>
</table>
MIPI VGI℠ Init Sequence

- Host VGI module gets initialized with the preset number of GPIOs.
- Host’s Tx o/p level is set to LOW
- Host’s Rx is ready for input level read
  - Input = LOW => Slave not ready
  - Input = HIGH => Slave ready

Is Slave VGI ready?

Yes

- Host sends enumeration-initiation packet
  - Slave responds

Further Communication as needed
From this point onwards
Synchronous 3-Wire MIPI VGI<sup>SM</sup>

- **Shift register includes buffering option to allow GPIO changes during ongoing transmission. Buffer depth is predefined.**
- **GPIO state Transmission starts when current state does not match with past state.**
- **Tx on Neg-Edge**
- **Rx on Pos-Edge**
- **GPIO state is updated at the end of the full frame reception. Frame reception is tracked using the common clock ticks.**
Asynchronous 2-wire MIPI VGI\textsuperscript{SM}: UART Mode

Illustration\#1: 8-bit frame

Illustration\#2: 12-bit frame

Illustration\#2: 16-bit frame
Asynchronous 2-wire MIPI VGI\textsuperscript{SM}: UART Mode

H/W Flow Control over Tx/Rx eliminates RTS/CTS physical pins
Asynchronous MIPI VGI\textsuperscript{SM}: Phase-Modulated PWM

Symbol to signal mapping for a joint (PWM+Phase) modulation scheme

Example Representation of an arbitrary data-sequence “10011101”

Link throughput and power :: A comparative look

Highlights:
- All Digital Solution
- 2x Throughput
- Time-domain data compression
- Link power Reduction by 50%
## MIPI VGI℠ Protocol

<table>
<thead>
<tr>
<th>Start-bit 0</th>
<th>Fn_Bit-0</th>
<th>Fn_Bit-1</th>
<th>GPIO/Msg Bit-0</th>
<th>GPIO/Msg Bit-1</th>
<th>GPIO/Msg Bit-2</th>
<th>...</th>
<th>GPIO/Msg Bit-(n-1)</th>
<th>GPIO/Msg Bit-n</th>
<th>Stop-bit 1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Type_Bit (vGPIO/Msg)</th>
<th>Stream-Ln Bit-0</th>
<th>Stream-Ln Bit-1</th>
<th>...</th>
<th>Stream-Ln Bit-(n-1)</th>
<th>Stream-Ln Bit-n</th>
</tr>
</thead>
</table>

### Function_Bits

<table>
<thead>
<tr>
<th>Function_Bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>Following bits are vGPIO states</td>
</tr>
<tr>
<td>0 1</td>
<td>Following bits are message bits.</td>
</tr>
<tr>
<td>1 0</td>
<td>Following bit-stream represent the vGPIO stream length to be set on the receiver side.</td>
</tr>
<tr>
<td>1 1</td>
<td>Following bit-stream represent the new vGPIO stream length acknowledgement w.r.t the previously received stream-length programming command.</td>
</tr>
</tbody>
</table>

**NOTE:** The mechanism has a fixed overhead of two-bits over the base-line vGPIO implementation.
MIPI VGI\textsuperscript{SM} FSM Integration with MIPI I3C\textsuperscript{SM}

- VGi FSM could be integrated with a serial interface of choice, such as MIPI I3C\textsuperscript{SM}
- I3C\textsuperscript{SM} supports MIPI VGI integration through dedicated Common Command codes (CCC) support in I3C\textsuperscript{SM} v1.0
- Helps reduce Hardware event pins at system level
**MIPI VGI\textsuperscript{SM} FSM Integration with MIPI I3C\textsuperscript{SM}**

- HW Event sideband signals are eliminated
- VGI-FSM (Finite State Machine) performs I3C\textsuperscript{(SM)} message encoding/decoding for HW events and thus frees up the associated CPU on the host-SoC for these tasks.
- Impact is reduced Latency and Power consumption.
Comparing MIPI VGI$^{SM}$

- **SPI**
  - Master-Slave approach
  - Custom implementations, no common methods
- **MIPI I3C$^{(SM)}$**
  - Multi-Master Multi-Slave, Open-Drain approach
  - In-band interrupts
- **MIPI RFFE$^{(SM)}$**
  - Master-Multi Multi-Slave approach
- **UART**
  - Custom implementations, requires reference clocks
- **MIPI VGI$^{(SM)}$**
  - Symmetric control approach (No Master No Slave)
  - Initialization from either side
Comparing MIPI VGI\textsuperscript{SM} - Clocking

- **UART**
  - Requires Reference Clock with Agreed rates

- **SPI, MIPI I3C\textsuperscript{(SM)}, MIPI RFFE\textsuperscript{(SM)}**
  - Clock is forwarded from Master to Slave

- **MIPI VGI\textsuperscript{(SM)}**
  - Using RO-PWM PHY option, the clocking is forwarded with data
  - Only Transmitter requires clock to create telegrams
  - Receiver captures telegrams without internal clock
    - Useful for devices which power down
    - Useful for very simple write-only devices (LED bank)
Phased MIPI VGI\textsuperscript{SM} Adoption – Leveraging Smaller FPGAs

- **Full VGI Adoption**
  - Device A (e.g., Host)
  - Device B (e.g., Peripheral)
  - VGI Sideband (SB) / GPIOs + Messaging
  - Native VGI Interface

- **Partial VGI Adoption**
  - Device A
  - Device B
  - VGI
  - Small FPGA
  - SB / GPIOs
  - FPGA VGI Bridging: Case-1

- **Partial VGI Adoption**
  - Device A
  - Device B
  - VGI
  - Small FPGA
  - SB / GPIOs
  - FPGA VGI Bridging: Case-2

- **No VGI Adoption**
  - Device A
  - Device B
  - SB: Sideband Signals
  - Across connectors, cables, hinges or pogo-pins etc.
  - FPGA VGI Bridging: Case-3
Summary

- Sideband GPIOs add to SoC and PCB level cost and complexity
- MIPI VGI consolidates sideband GPIOs and Low-Speed serial messaging interface in P2P configuration to reduce I/O pins
- Both 2 and 3-wire interface options are available
- Common PWM start-up mode ensures interoperability
- The VGI FSM can be combined with any other interface bus of choice, e.g. I3C\textsuperscript{(SM)} VGI
- The MIPI VGI Specification is to be released in 2018
mipi®
DEVCON
THANK YOU

2017
MIPI ALLIANCE
DEVELOPERS
CONFERENCE

HSINCHU CITY, TAIWAN
MIPI.ORG/DEVCON