MIIPI Discovery and Configuration (DisCo℠) Creation Tool
Agenda

• DisCo Background
• DisCo Creation Tool
• Demo
What is DisCo (Discovery and Configuration)?

- ACPI based specifications that describe hardware
  - Defines properties that are device-specific: MIPI I3C HCI\textsuperscript{SM}, MIPI Soundwire Master, etc.
- MIPI currently defines 4 DisCo specifications (software.mipi.org) - more on the way
- Information is encoded in ACPI Objects like \_DSD (Device-Specific Data) properties and includes:
  - Platform-specific information
  - Component-specific information
- DisCo properties are published by platform firmware into system memory, and consumed by software drivers in the operating system
DisCo Properties

- Properties are often hierarchical, many layers of sub properties
- Properties are written in Advanced Configuration and Power Interface Source Language (ASL)

<table>
<thead>
<tr>
<th>Property String</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>“mipi-sdw-sw-interface-revision”</td>
<td>Integer</td>
<td>This is a 32-bit value where the upper word contains the major version number of this specification, and the lower word contains the minor version number.</td>
</tr>
<tr>
<td>“mipi-sdw-max-clock-frequency”</td>
<td>Integer</td>
<td>This value provides the maximum Bus clock in Hz for this master. This is the maximum usable Bus clock frequency for this platform.</td>
</tr>
<tr>
<td>“mipi-sdw-clock-frequencies-supported”</td>
<td>Package</td>
<td>A package containing one integer entry for each clock frequency supported. Frequencies are represented in Hz.</td>
</tr>
<tr>
<td>“mipi-sdw-supported-clock-gears”</td>
<td>Package</td>
<td>A package containing one integer entry for each supported gear, e.g. {1, 2, 3, 4, 16}. Some Masters may only support a single gear, or powers of two.</td>
</tr>
</tbody>
</table>
Challenges with DisCo Properties

• Properties are written “by hand” (with a text editor)
  – DisCo Property sets are documented in multiple specs
  – Vendor-specific property definitions also exist
  – Development process is prone to error

• Property values must be provided by a component vendor, or discerned by the firmware developer from a datasheet
  – Hand-coding complex hierarchies is easy to get wrong
  – Current compilers do not check _DSD syntax or data types
DisCo Creation Tool

- Allows component vendors to select the properties that apply to their components, select/enter the value(s) that apply to those properties, and ensures that the input values and format conform to DisCo specifications
- Allows vendors to extend with vendor-specific properties as needed
- Provides open-source access to device property templates via repository
- Makes DisCo properties easy to manage and use
- Reduces amount of effort required to develop platform firmware images
- Reduces effort needed for platform firmware/software debug

Microsoft Corporation & Intel Corporation
Demo

Microsoft Corporation & Intel Corporation
Typical Workflow

- **MIPI Alliance**
  - Define New Specification
  - Build DisCo Tool Template for New Specification
  - Maintain DisCo Tool

- **Component Vendors**
  - Use MIPI provided Device Templates
  - Create Device Configuration Template for their Device

- **System Integrators**
  - Obtain Device Templates from Repository
  - Generate Final ASL

- **Private Template Repository**

Microsoft Corporation & Intel Corporation

© 2019 MIPI Alliance, Inc.
Current status

- Initial code contributed by Microsoft
- New features are constantly getting added
- Tentative Public Availability: June 2020
ADDITIONAL RESOURCES

• DisCo Specifications: software.mipi.org

Microsoft Corporation & Intel Corporation
THANK YOU