mipi alliance

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

What's New in MIPI I3C Basic and the I3C Ecosystem

15 September 2021

Tim McKee
Intel Corporation,
MIPI I3C Working Group Chair

Matthew Schnoor
Intel Corporation,
MIPI I3C HCI Subgroup Lead

Michele Scarlatella
I3C Technical Consultant
MIPI Alliance

Webinar Agenda

1. **What's new with I3C/I3C Basic**
   - **Tim McKee** - MIPI I3C Working Group Chair
2. **What's new with the MIPI I3C Host Controller Interface (HCI)**
   - **Matthew Schnoor** - MIPI I3C HCI Subgroup Lead
3. **How I3C Enables Power Efficient Designs**
   - **Michele Scarlatella** - I3C Technical Consultant
4. **Questions and Answers**

mipi alliance

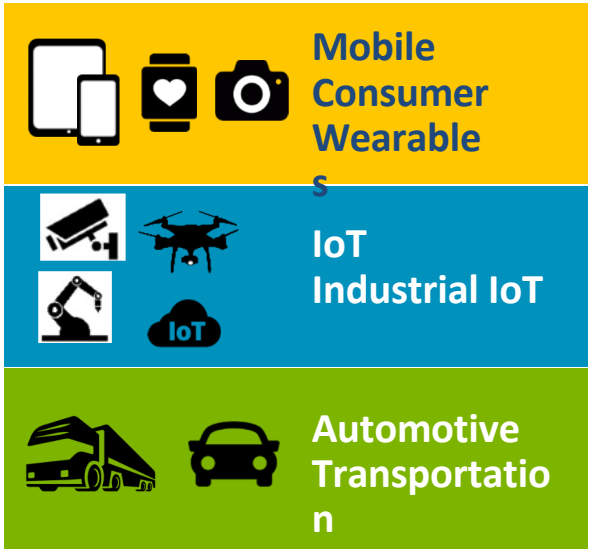
IF IT'S NOT MIPI, IT'S NOT MOBILE

What's new with I3C/I3C Basic

Tim McKee
Intel Corporation,
MIPI I3C Working Group Chair

MIPI I3C Interface - Introduction

Fast Growing Sensor Markets



Smarter, more capable

Avg >20 sensors/device (projected)

- Accelerometer
- Gyroscope
- Magnetometer
- Ambient light
- Pressure
- Humidity
- Temperature
- .. others

I3C - a scalable, cost-effective interface

- Targeting mobile & IoT devices, automotive, server manageability
- Simplify connecting and managing multiple sensors in a device
- Reduced pin count and signal paths
- Support for low-power, high-speed communication (vs. I2C/SPI/UART)
- Multi-host arbitration
- Low complexity (<2Kgates / target)
- Compatibility with I2C devices

Evolution of MIPI I3C Interface Specification

I3C v1.0

- First version specification
- Released **Dec 2016**

I3C v1.1

- New I3C specification
- New features added
- Released **Nov 2019**

I3C v1.1.1

- Updated with errata
- No new major features
- Released **Jun 2021**

I3C Basic v1.0

- License free
- Subset of features
- Released **Jul 2018**

I3C Basic v1.1.1

- More closely aligned feature set
- Released **Jul 2021**

MIPI I3C Key Features

Fast Efficient Communication Channel

- Multidrop SDA/SCL 2-wire interface
  - 12.5 MHz max Clock rate
  - 1.2V-3.3V Voltage supported
- Dynamic switch between pull-up/push-pull/Hi-Z
- Line Coding modes for higher throughput:
  - SDR, HDR-DDR/TSP/TSL/BT
  - Multilane x2, x4
- Low EMI
- Physical layer CMOS I/O compatible

System Management

- Primary & Secondary Controllers
- Dynamic Address Assignment including Group Addressing
- Device Code Assignment (by MIPI)
- Descriptors:
  - **DCR** Device Configuration Register
  - **BCR** Bus Configuration Register

Advanced Functions

- Multi-operation via repeated START
- Unicast, Broadcast, Multicast messages
- In-Band Interrupt with qualified information
- Device reset
- Hot-Join
- Error detection (parity, CRC)

Backward Compatibility

- Mixed-bus operation: I3C and I2C devices
  - Static address space reserved for I2C legacy devices
- Fast operations invisible to I2C thanks to 50ns spike filter

From v1.1

New Specifications for 2021

- **I3C Version 1.1.1**
  - Clarifications (Small errors and typos)
  - Terminology changes (Controller/Target)
  - Incorporation of previous modifications (Target Reset)
- **I3C Basic Version 1.1.1**
  - Synchronized with I3C v1.1.1
  - Clarifications and Terminology changes
  - New features (HDR Modes, CCCs, Timing Control)
- **I3C Host Controller Interface (HCI) Version 1.1**
  - Presentation coming up……..

Comparison of Features

Feature	I3C v1.0	I3C Basic v1.0	I3C v1.1.1	I3C Basic v1.1.1
12.5 MHz SDR (Controller, Target and Legacy I²C Target Compatibility)	✓	✓	✓	✓
Target can operate as I²C device on I²C bus and on I3C bus using HDR modes	✓	✓	✓	✓
Target Reset	✗	✗	✓	✓
Specified 1.2V-3.3V Operation for 50pf C load	✓	✓	✓	✓
In-Band Interrupt (w/MDB)	✓	✓	✓	✓
Dynamic Address Assignment	✓	✓	✓	✓
Error Detection and Recovery	✓	✓	✓	✓
Secondary Controller	✓	✓	✓	✓
Hot-Join Mechanism	✓	✓	✓	✓
Common Command Codes (Required/Optional)	✓	✓ ✗	✓	✓ ✗
Specified 1.0V Operation for 100pf C load	✗	✓	✗	✓
Set Static Address as Dynamic Address CCC (SETAASA)	✗	✓	✓	✓
Synchronous Timing Control	✓	✗	✓	✗
Asynchronous Timing Control (Mode 0)	✓	✗	✓	✓
Asynchronous Timing Control (Mode 1-3)	✓	✗	✓	✗
HDR-DDR	✓	✗	✓	✓
HDR-TSL/TSP	✓	✗	✓	✗
HDR-BT (Multi-Lane Bulk Transport)	✗	✗	✓	✓
Grouped Addressing	✗	✗	✓	✓
Device to Device(s) Tunneling	✗	✗	✓	✗
Multi-Lane for Speed (Dual/Quad for SDR and HDR-DDR)	✗	✗	✓	✗
Monitoring Device Early Termination	✗	✗	✓	✗

Where is I3C Being Leveraged?

- **JEDEC**
  - Serial Presence Detect (SPD) Hub for DDR Memory
- **DMTF**
  - Management Component Transport Protocol (MCTP) and I3C protocols for bus management
- **Others in development**
  - PCIe I3C sideband
  - I3C over IEEE 1722
  - Protocol Adaptation Layer (PAL) for I3C over A-PHY
  - ETSI for Smart Secure Platform
  - More?

MIPI I3C Ecosystem / Support

- Conformance Test Suite (CTS)
- Host Controller Interface
- I3C Subsystem Linux Kernel Module
  https://www.kernel.org/doc/html/latest/driver-api/i3c/index.html
- Debug for I3C: Enables debugging over I3C
- DisCo for I3C: Simplifies software integration
- FAQs
- Plugfest Interoperability Testing
- Application Notes

Get Involved / Sources of Further Information

- **I3C Working Group**
  - Open to MIPI contributor members (Meets: Wednesdays 8am PST)
- **Contact the Working Group**
  - Let us know your questions/comments
  - Email: i3c@mipi.org (members)
  - Email: admin@mipi.org (non-members)
- **I3C Supporting Documents**
  - FAQs
  - App Notes
  - Errata
- **Website:** https://www.mipi.org/specifications/i3c-sensor-specification

mipi alliance

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

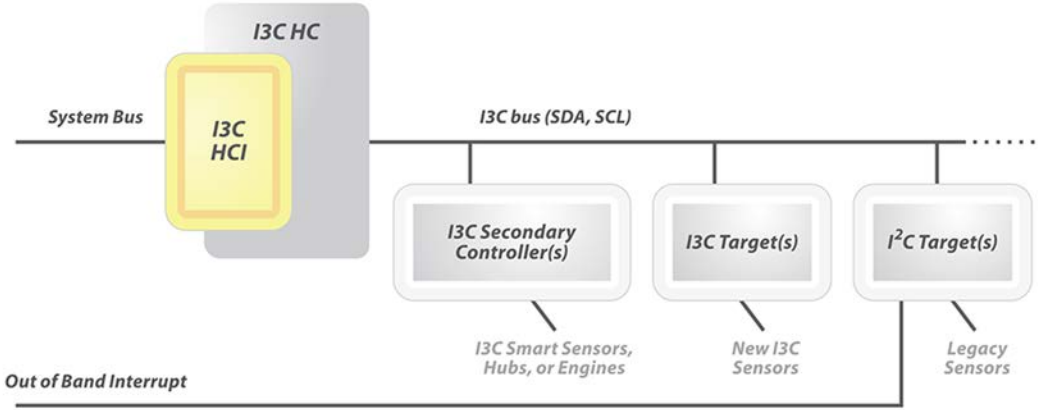
What's new with the MIPI I3C Host Controller Interface (HCI)

Matthew Schnoor
Intel Corporation,
MIPI I3C HCI Subgroup Lead

 |

I3C Host Controller Interface (HCI) – What and Why?

- Intended to standardize the interface that platform software uses to access I3C Target Devices and their capabilities.
- Enables a rich ecosystem for development; usable for various use cases, including IoT, mobile, datacenter and other emerging applications.

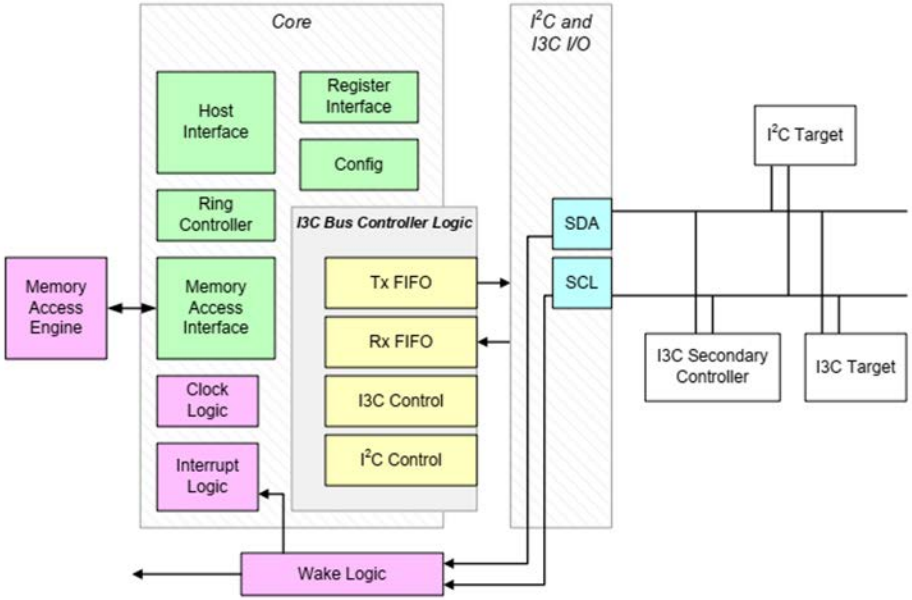


Example of I3C Bus with connection to Host via System Bus

I3C HCI Specification provides a standardized command/response interface

Benefits of standardizing the interface

- Ecosystem can use a common software driver to detect, configure and present I3C Targets to their application
  - Common register definitions for Controller configuration
  - Common data structures for transfer commands and I3C Target responses
  - Two operating modes allow implementer flexibility for different needs
- Application developers can focus their attention on higher-level flows
- Implementers can define Extended Capabilities as needed (per use case)



Example of I3C HCI Implementation

I3C HCI Specification defines the common register definitions, data structures and operating modes for the implementer.

History of I3C HCI

- **Version 1.0 – Adopted 4 April 2018**
  - Supports up to 32 I^2C and I3C Target Devices per I3C Bus
  - Supports features defined in I3C v1.0 and (subsequent) I3C Basic v1.0
  - Supports I3C Transfers in SDR Mode, HDR-DDR and HDR-Ternary Modes
  - Supports Broadcast and Direct CCCs (with managed framing)
  - Supports Time Stamping (Asynchronous)
  - Supports Dynamic Address Assignment with ENTDAA and SETDASA CCCs
  - Operating modes:
    - PIO Mode (lightweight HW, Host-intensive for each TX/RX transfer)
    - DMA Mode (adds memory controller, offloads TX/RX queue operations from Host)

History of I3C HCI

- **Version 1.1 – Adopted 20 May 2021**
  - Supports many of the features defined in I3C v1.1/v1.1.1 and (subsequent) I3C Basic v1.1.1
  - Adds support for Grouped Addressing
  - Adds Dynamic Address Assignment method for SETAASA (as Broadcast CCC)
  - Adds support for CCCs with Defining Bytes
  - Adds support for Target Reset Pattern flows (with optional preceding RSTACT CCCs)
  - Adds more robust error handling methods and Bus reset/recovery flows
  - Adds scatter-gather support (optional) for DMA Mode
  - Many clarifications, bug fixes and other issues addressed (from implementer feedback)
  - Expanded and reorganized “Theory of Operation” section

I3C HCI enables the I3C ecosystem

- **Linux I3C HCI Driver has been released to the open-source community**
  - Driver development funded by MIPI Alliance
  - Available in mainstream Linux kernel tree (v5.x)
  - Supports HCI v1.1 (adopted) and 2.0 (early draft)

- **Useful as a key component of integrations and other I3C subsystems**
  - HCI-compliant I3C Controller IPs are available on the market today

- **MIPI Software WG wants to hear about your use integrations cases, feature requests or any improvement ideas…**

Next Steps for I3C HCI

- **Version 1.2 – optional normative feature adds, development is in progress**
  - Answering requests from other MIPI WGs, MIPI Contributors and industry liaisons, including:
  - Support for I3C “Dead Bus Recovery” methods (optional)
  - Better support for CSI-2 v4.0 use cases based on I3C
  - Optional support for Standby mode in limited Secondary Controller Role
  - Optional support for scheduled commands (i.e., periodic polling of I3C Targets)
  - Support for HDR Flow Control capabilities, etc.

- WG aims to **preserve SW compatibility** and wisely invest effort in HCI v1.x, to add support for new I3C optional capabilities that address specific use cases.

Next Steps for I3C HCI

- **Version 2.0 – generational advancement; feature development currently paused**
  - Looking to add performance and capability upgrades requested by MIPI Contributors, including:
  - Support for additional HDR Modes (e.g., HDR-BT)
  - Support for Multi-Lane transfers
  - Additional support for I3C Secondary Controller role
  - Additional run-time controls for Queue/Ring status
  - More extensibility and configurability

- Larger feature additions will require **substantial** changes to HCI definitions, with impact to the existing ecosystem… we need MIPI Contributors to **review and comment**.

mipi alliance

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

How I3C Enables Power Efficient Designs

Michele Scarlatella
I3C Technical Consultant,
MIPI Alliance

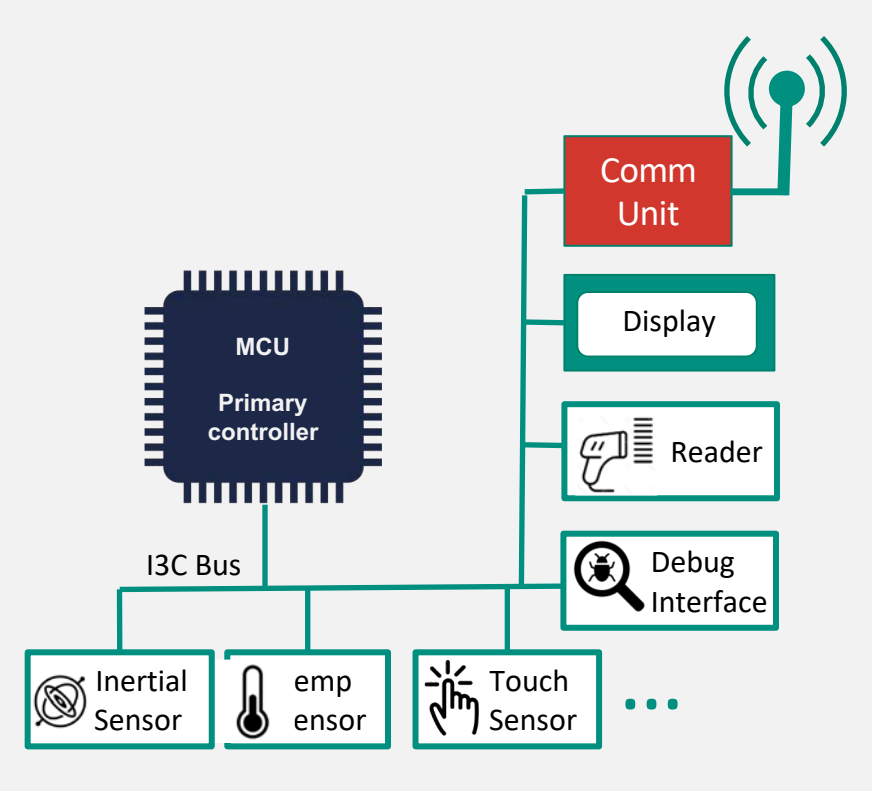
 |

How I3C helps Power Efficiency for IoT Devices

- **Many classes of IoT devices work under tight power budget (Wearables, in-file data collection units, etc.)**
- **Low energy consumption and Power efficiency are key features**
  - Low-voltage operation
  - Running on battery
  - Non-rechargeable, targeting several years operation
- **I3C can help solve above challenges**
  - Excellent bus electrical characteristics
  - Handling of synchronous and asynchronous events
  - Selective power management of sub-components
  - System segmentation

I3C Features & specs are great help to System Designers to architect Power Efficient IoT Devices for their next project

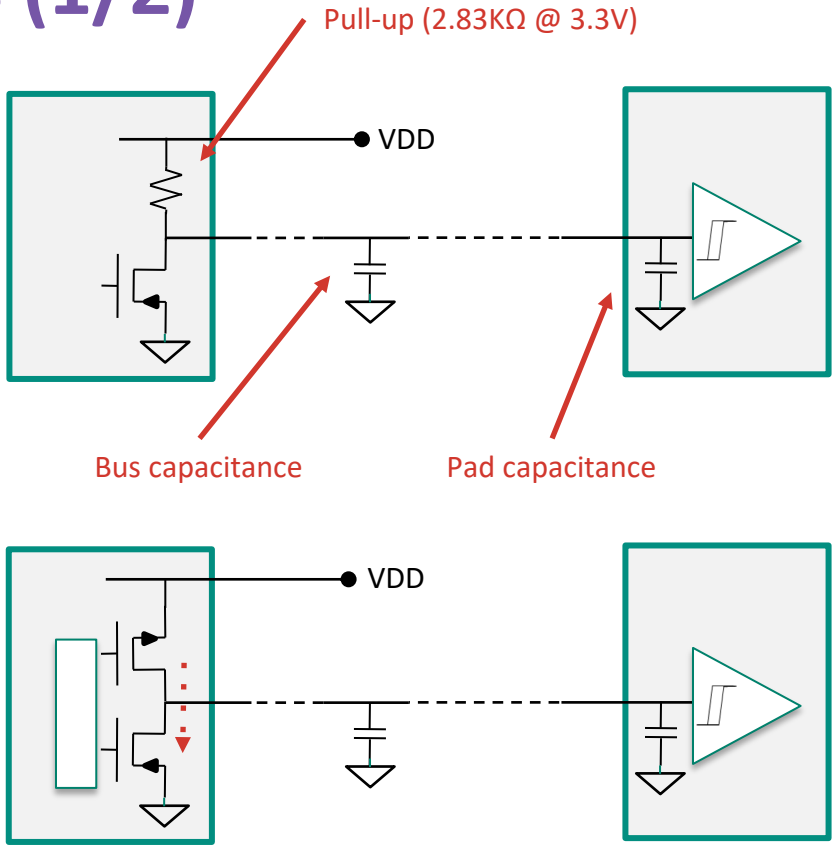
Parameters Affecting Energy Efficiency



- Low voltage operation
- Bus electrical characteristics
- Voltage compatibility
- Efficiency of bus transactions
  - Scheduled and asynchronous data acquisition and transfer
- Selective power management of sub-units
- Managing «idle time»

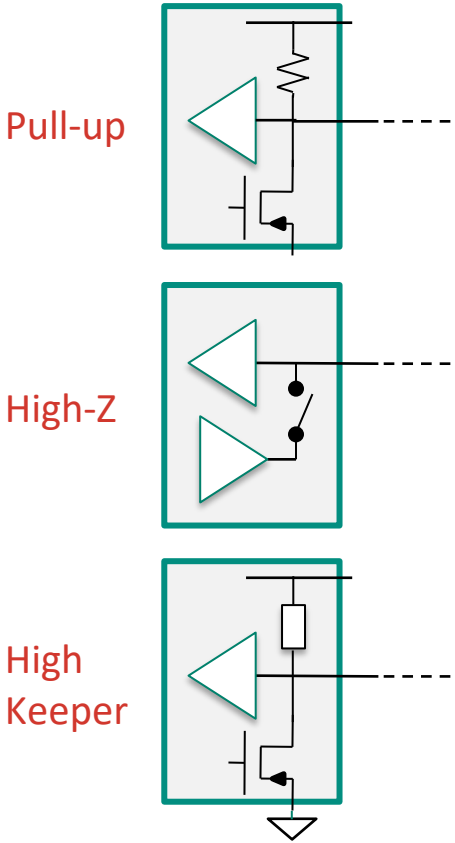
I3C Bus Electrical Features (1/2)

- **Electrical**
  - Low Operating voltage: 3.3V - 1.2V (…soon below)
  - Bus capacitance <10pF / device (total 50-100pF)
  - Pull-up: 1.1-2.8 kΩ
- **Factors affecting Energy consumption**
  - Switching shoot-through current
    - optimize IP selection
  - Bus capacitance
    - keep short bus length,
    - reduced Capacitance on input pads
  - Pull-down current
    - minimize pull-down time
    - use push-pull whenever possible

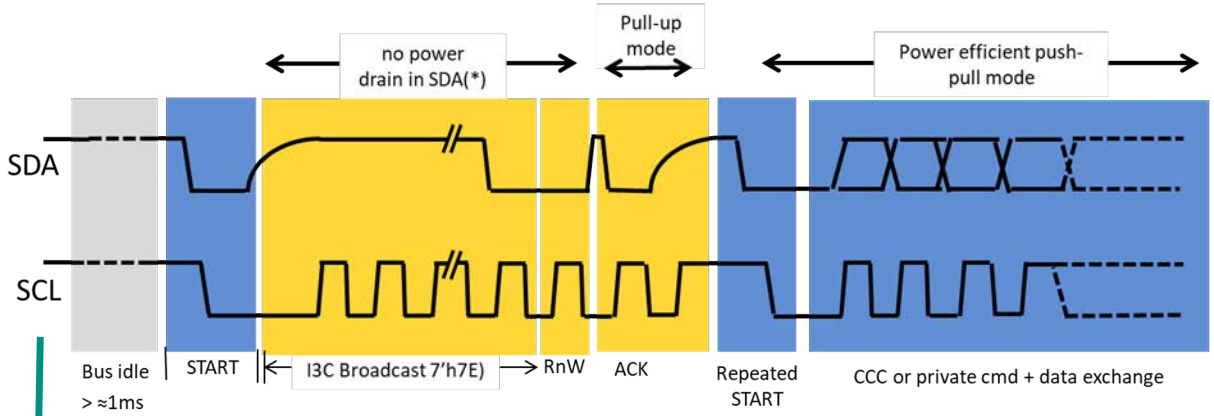


I3C Bus Electrical Features (2/2)

- **When not active, I3C bus lines can be in three states:**
  - Pull-up (Open Drain)
  - High-Z
  - High-Keeper (light pull-up, higher dynamic R)
- **SCL** is (almost) always Push-Pull
  - No clock stretching allowed
- **SDA** is switched dynamically by Active Controller between Pull-up (Open Drain), Push-Pull, High-Z, High-Keeper



Typical Bus Transaction



SCL is usually in power efficient Push/Pull mode, driven by Controller(**)

Highlights:

- Line interfaces structured to avoid inefficient pull-up mode
- High Data Rate (HDR) modes boast excellent mJ/bit performance
- Faster bus transactions imply reduced CPU cycles

(*) If assigned Dynamic Addresses values are < 0x3F (54 avail.)

(**) There are a few instances where SCL is in Open drain mode, mostly related to I2C compatibility

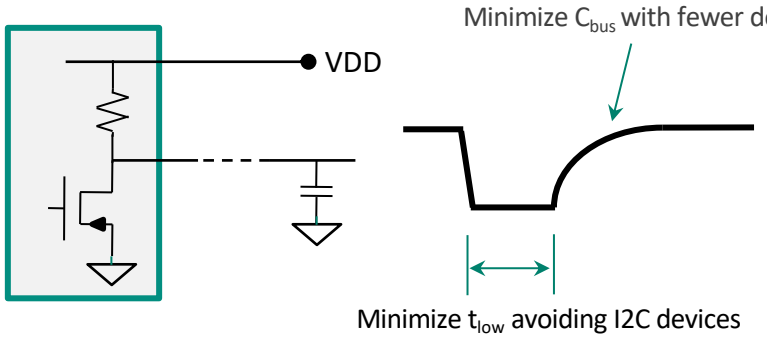
I3C Energy Consumption Estimate

1/8th lower than I2C

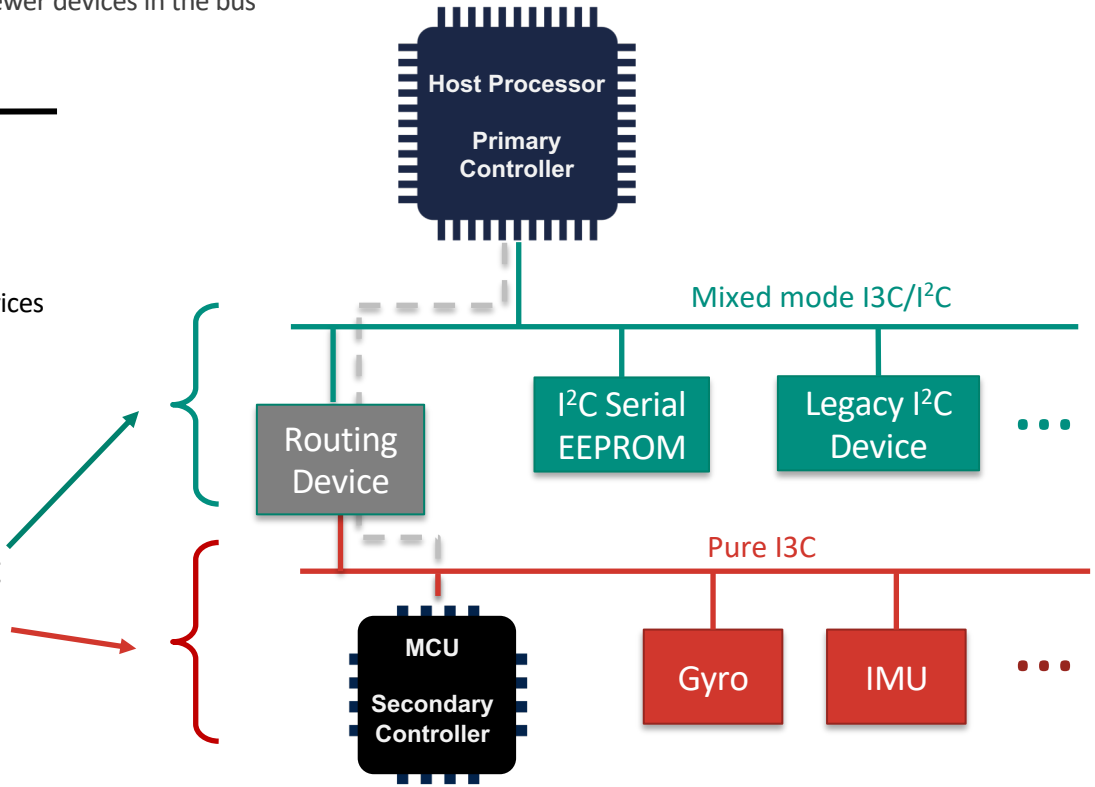
- High Data Rate (HDR):
  - Switching waveforms preserve good power performance for all modes
- Faster transaction execution
- Reduced CPU cycles

SDR = Single Data Rate
DDR = Double Data Rate
TSL = Ternary Symbol Legacy inclusive bus
TSP = Ternary Symbol Pure bus

Optimize by Bus Segmentation



- Isolate I3C high activity devices
- Consider Segmentation using Routing Devices
  - **Low activity bus** & mixed mode I3C/I²C
  - **High activity bus** (pure I3C)
- Better processing efficiency



Specs ref. v1.1.1 (5.1.9.3.17/20)

Improving Idle-time with Bus Conditions

Informing Targets of bus idle condition facilitates low power management

- Specific CCCs allocated: ENTAS[0..3]
- ENTASx CCC informs Target(s) about low activity intervals
  - 2 ms & 50 ms idle time can rack up significant power savings

Activity state CCC	Bus Idle time
ENTAS0	1 µs
ENTAS1	100 µs
ENTAS2	2 ms
ENTAS3	50 ms

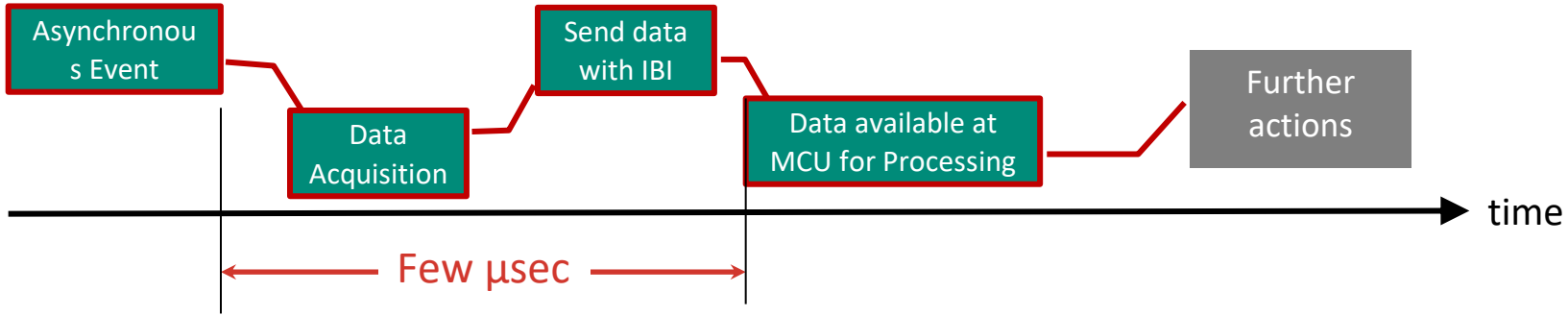
Remarks:

- ENTASx CCC is only a “suggestion” to Target
- This does not replace application specific or custom power savings agreements (i.e., “private contracts”)

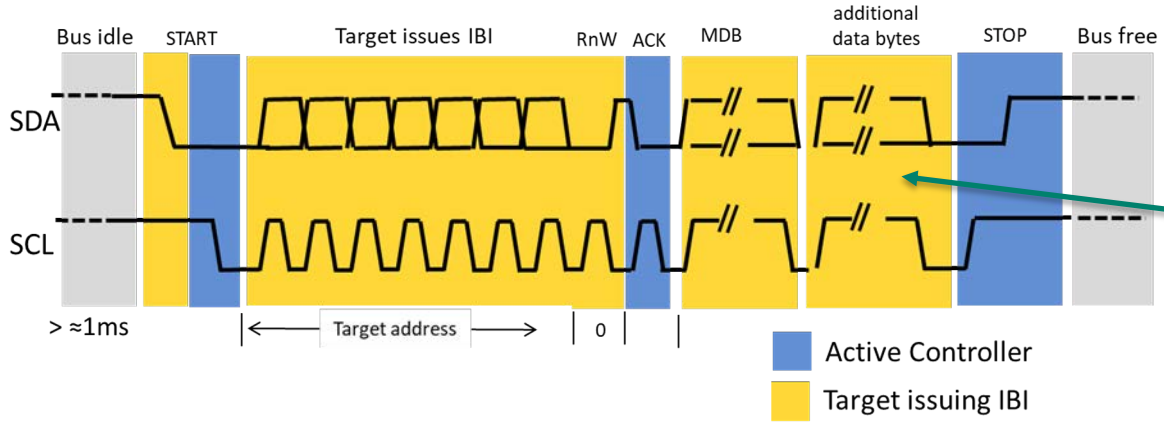
Efficient Data Acquisition with In-Band-Interrupts (IBI)

IBI allows fast and efficient asynchronous data acquisition, and event processing

- Data produced by a target is promptly transferred to upstream controller for further processing
- IBI are intended as an efficient mechanism for Targets to grab Controller attention
- Avoids extra dedicated wires, or inefficient polling mechanism



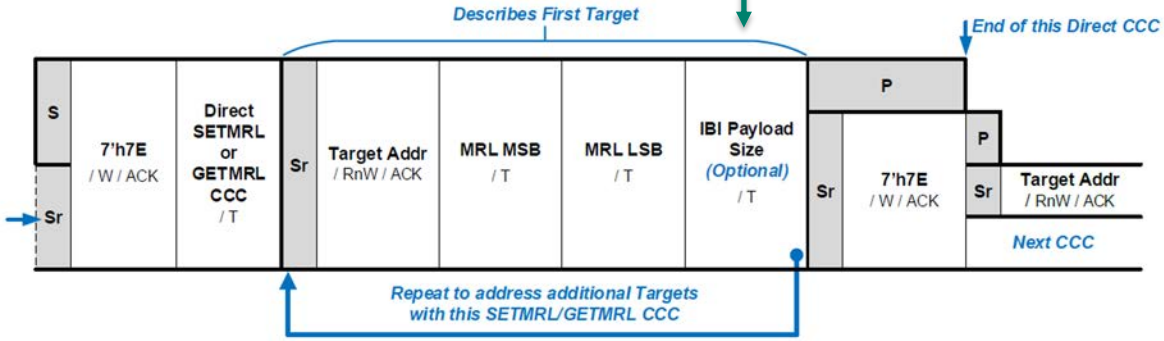
Efficient Data Acquisition with IBI



How is it done:
Additional data is inserted as part of the IBI request

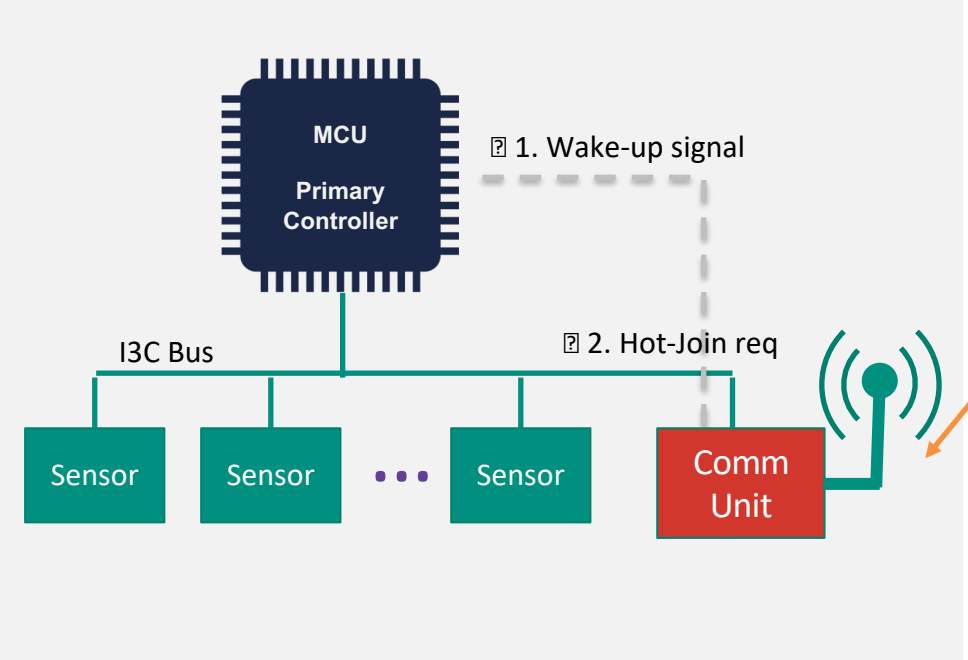
Additional data size definition by SETMRL CCC

- **Data transfer by a READ or using Mandatory Data Byte (MDB)**
  - Active if BCR[2]=1
- **Structure of MDB:**
  - Interrupt Group Identifier (3-bits)
  - Specific Interrupt Identifier (4 bits)
  - Vendor defined meaning possible



Specs ref. v1.1.1 (5.1.6, 5.1.9.3.6)

Power Management with Hot-Join (HJ)



HJ Primary uses

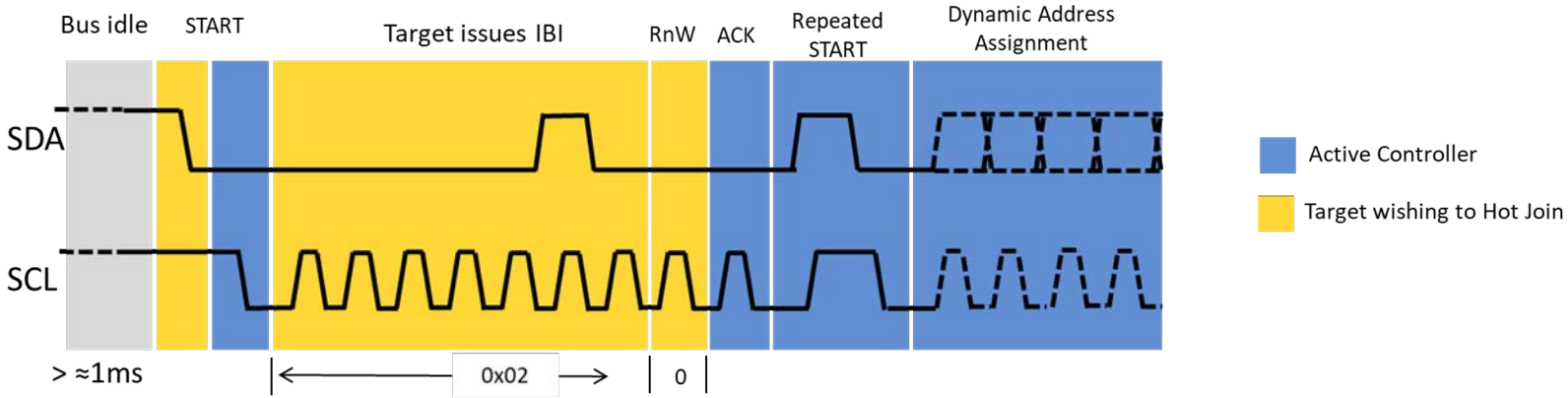
- Attach device after bus is configured
- Provision exist for late power-up of a target

Improved power management

- Selective powering of sub-units
- Wake-up only when needed
- Wake-up signal can be
  - Out-of-band (HW wire)
  - In-band with Target Reset Action (RSTACT) target reacting to a predefined pattern

(*) IBI=In Band Interrupt

Power management with Hot-Join - How



- HJ has similar pattern to IBI with predefined high-priority address 0x02
- Dynamic Address Assignment need to be executed
- During off-state target must insure no power is inadvertently drained through SDA/SCL wires

Summary and Key Takeaways

- I3C electrical characteristics are well suited for IoT low-power devices with wide voltage classes & low power consumption
- I3C is not just a ‘bit-pipe’ but modern bus with many high-level functions that facilitate the design of power efficient IoT devices
  - Segmentation into sub-systems
  - Improved event management
  - Efficient data acquisition
- Hot Join feature allow ‘power segmented’ designs, by keeping units active only when needed, with interoperable and standard procedures
- As more and more I3C MCU’s and sensors are expected to be in the market, I3C is a key candidate for next generation designs

MIPI I3C Additional Information #1

- **MIPI I3C Basic Specification v1.1.1**
  - https://resources.mipi.org/mipi-i3c-basic-download
- **MIPI I3C Specification v1.1.1**
  - https://www.mipi.org/specifications/i3c-sensor-specification
- **MIPI I3C v1.1 – A Conversation with Ken Foust**
  - https://resources.mipi.org/blog/mipi-i3c-v1.1-a-conversation-with-ken-foust
- **Whitepaper: Introduction to the MIPI I3C Standardized Sensor Interface**
  - https://resources.mipi.org/hubfs/MIPI_Alliance_I3C_Whitepaper.pdf
- **MIPI DevCon 2021 – I3C related sessions:**
  - MIPI I3C Signal Integrity Challenges on DDR5-based Server Platform Solutions
    - https://www.mipi.org/devcon/2021/agenda/mipi-i3c-signal-integrity-challenges-ddr5-based-server-platform-solutions
  - MIPI I3C Application and Validation Models for IoT Sensor Nodes
    - https://www.mipi.org/devcon/2021/agenda/mipi-i3c-application-and-validation-models-iot-sensor-nodes
  - MIPI I3C Under the Spotlight: A Fireside Chat with the I3C Experts
    - https://www.mipi.org/devcon/2021/agenda/mipi-i3c-under-spotlight-fireside-chat-i3c-experts

MIPI I3C Additional Information #2

- **I3C Basic in JEDEC DDR5: A Sum Greater Than Its Parts**
  - https://resources.mipi.org/blog/mipi-i3c-basic-in-jedec-ddr5-a-sum-greater-than-its-parts
- **System Integrators Application Note for MIPI I3C v1.0 and I3C Basic v1.0 (2018)**
  - https://www.mipi.org/sites/default/files/mipi_I3C-and-I3C-Basic_app-note-system-integrator_v1-0p.pdf
- **I3C and I3C Basic Frequently Asked Questions (2021)**
  - https://resources.mipi.org/hubfs/FAQs/mipi-I3C-v1-1-1-and-I3C-Basic-v1-1-1-FAQ-v1-Public.pdf
- **I3C Application Note: Hot-Join (2021)**
  - https://resources.mipi.org/hubfs/white-papers/mipi_I3C_app-note_Hot-Join_v1-0p.pdf
- **I3C Application Note: Virtual Devices and Virtual Targets (2021)**
  - https://resources.mipi.org/hubfs/white-papers/mipi_I3C_app-note_Virtual-Devices_v1-0p.pdf
- **I3C Basic Target Reset (now in v1.1.1)**
  - https://resources.mipi.org/mipi-i3c-basic-download
- **I3C Host Controller Interface**
  - https://www.mipi.org/specifications/i3c-hci

Questions and Answers

 |

Thanks for attending!

