MIPI I3C®

The Next-Generation Utility and Control Bus Supporting Time-Critical Applications

Michele Scarlatella, MIPI Alliance

Embedded World Conference

10 April 2024

 © 2024 MIPI Alliance, Inc.

2003 THE CELL PHONE MARKET





IN 2003 MIPI ALLIANCE WAS FORMED TO STANDARDIZE CAMERA AND DISPLAY INTERFACES

Board Members

Google intel Qualcomm BOSCH SAMSUNG SYNOPSYS® Texas Instruments

Contributor Members





 © 2024 MIPI Alliance, Inc.

MIPI Specializes in High-Performance, Low-Power, Low-EMI Connectivity

Today, MIPI specifications are widely leveraged beyond mobile:



50+ Current Specifications

MOST RECOGNIZED SPECIFICATIONS

- CSI-2 (Camera)
- DSI-2 (Display)
- SoundWire (Audio)
- RFFE (RF front end)
- I3C (Bus interface)
- A-PHY (Long-reach physical layer - up to 15m)
- C-PHY & D-PHY (Short-reach physical layer - up to 4m)

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

Over the last 20 years, MIPI specifications have been proven in billions of devices

MIPI Alliance Member Ecosystem



 © 2024 MIPI Alliance, Inc.

The MIPI I3C Interface

Smarter interface, more capable for sensing and control



Mobile
Consumer
Wearables



IoT
Industrial IoT



Automotive
Transportation

Fast-growing sensor markets

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

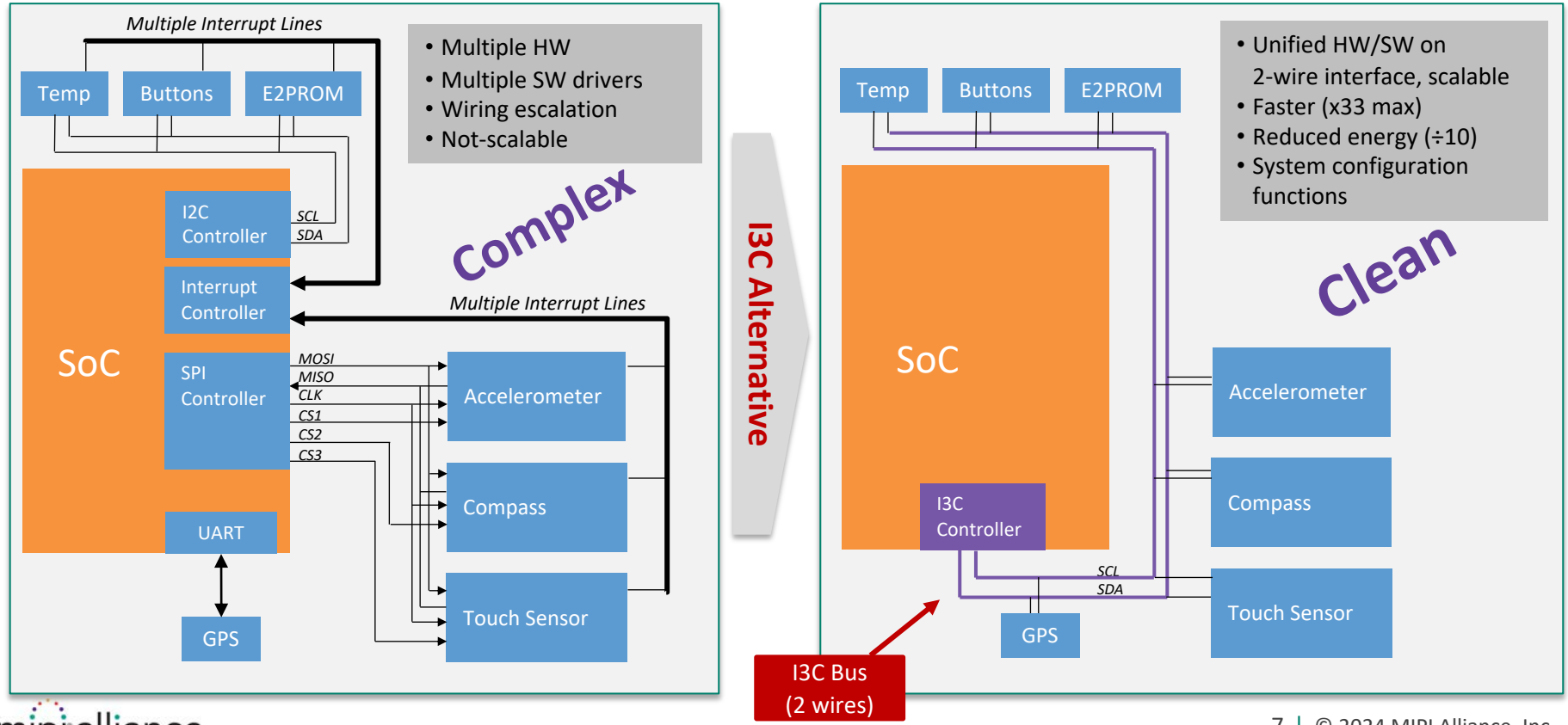
Camera / Display
HPC
Automotive

I3C a scalable, cost-effective interface

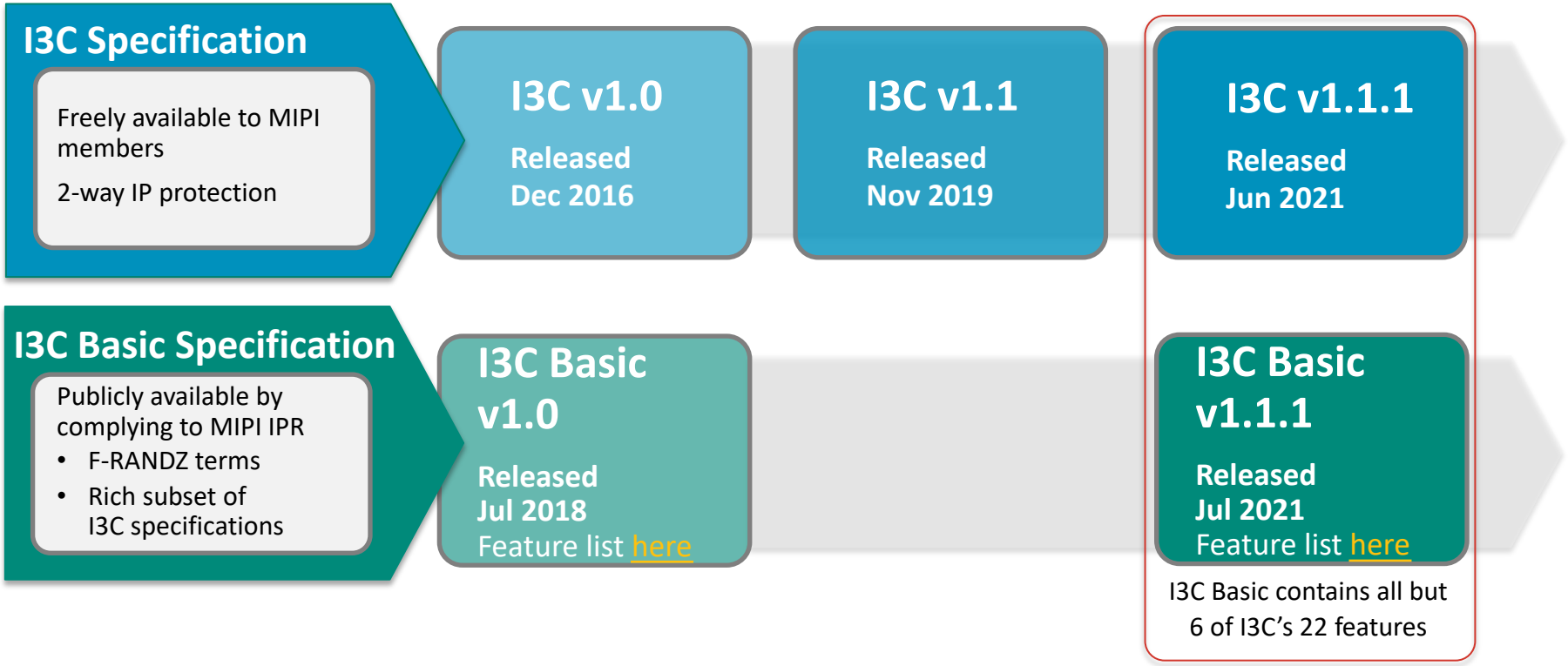
- Targeting mobile & IoT devices, automotive, server manageability
- Simplifies connecting and managing multiple sensors in a device
- Reduced pin count and signal paths
- CMOS I/O compatible
- Support for low-power, high-speed communication (vs. I2C/SPI/UART)
- Low complexity target
- Compatibility with I2C devices

MIPI I3C – Two-Wire-Based Intelligent Interface

Better and unifying alternative, replacing I2C, SPI, SMBus, UART



MIPI I3C Specifications



MIPI I3C Software Specifications

All publicly available!

MIPI I3C HCI

A host controller interface for efficiently accessing MIPI I3C devices and capabilities

- **v1.0** Released Apr 2018
- **v1.1** Released May 2021
- **v1.2** Released Apr 2023

MIPI I3C TCRI

Defines standard transfer command and transfer response structures

- **v1.0** Released Sep 2022

MIPI DisCo for I3C

Simplifies software integration for devices that use MIPI I3C

- **v1.0** Released Jun 2019
- **v1.1** Released Feb 2023

 © 2024 MIPI Alliance, Inc.

MIPI I3C Technical Brief

© 2024 MIPI Alliance, Inc.

MIPI I3C Features

- Multi-controller
- Dynamic address
- Simple device discovery
  - Pre-assigned 8-bit Device Characteristic Register (DCR) Value
- Messaging:
  - Unicast
  - Broadcast
  - Group-broadcast
- Bus operating parameters
  - Max packet length
  - Timing delay (→ max speed)
- Error detection and recovery
- Target reset action
- Special bus functions

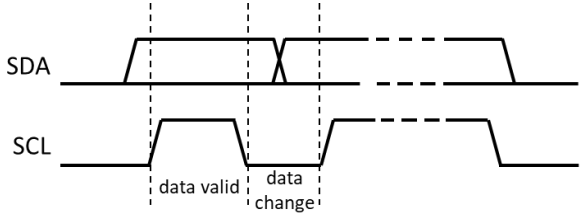
Device Characteristic Register Family

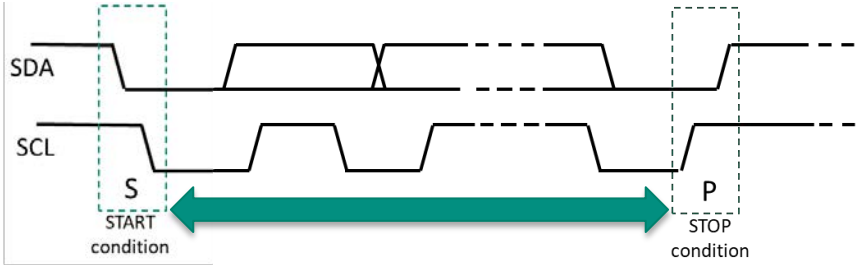
Family	Example Devices
HMI	Touch, gesture, fingerprint, haptic, …
Environmental	Pressure, temperature, UV, …
Industrial	RGB color match, shock, mass flow, …
Medical	EKG, ECG, glucose meter, oximeter, …
Navigation	Gyro, magnetometer, humidity, IR, ..
Security	UICC (ETSI), embedded SE (mobile, IoT), …
Communication	RF device, IR device, RF ranging, ..
Special MIPI devices	Bridge, hub, bus monitor, secondary controller, PMIC

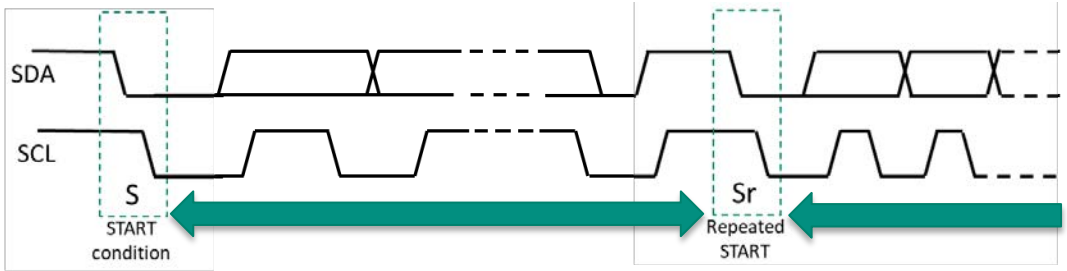
Complete DCR list at: www.mipi.org/mipi_i3c_device_characteristics_register

MIPI I3C Electrical Bus Drive

- Heavily borrowed from I2C
- Similar (not equal) to I2C transactions:
  - SCL: driven by the controller
    - Always(*) in push-pull mode
  - SDA: driven by controller or target
    - Dynamically switch among:
    - Open-drain / push-pull / Hi-Z / Hi-Keeper
- Multi-controller
- Dynamic address
- Simple Device discovery
- Operating voltage: 1.2V-3.3V
- Bus speed:
  - SCL max frequency: 12.5MHz (low EMI)
  - Standard Data Rate (SDR): 11.1Mbps
  - Hight Data Rates, up to 100Mbps
- Compatible with I2C(+)



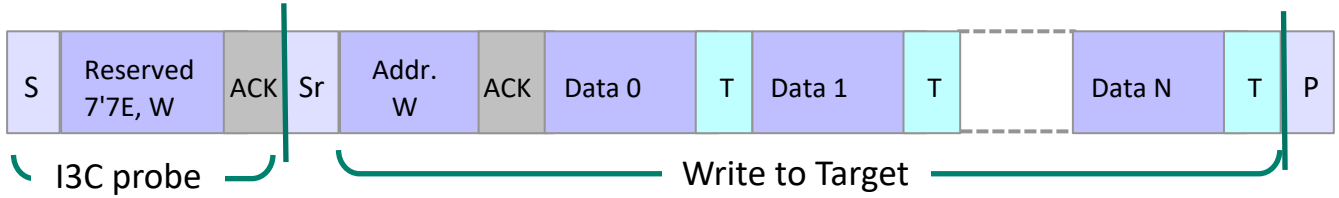


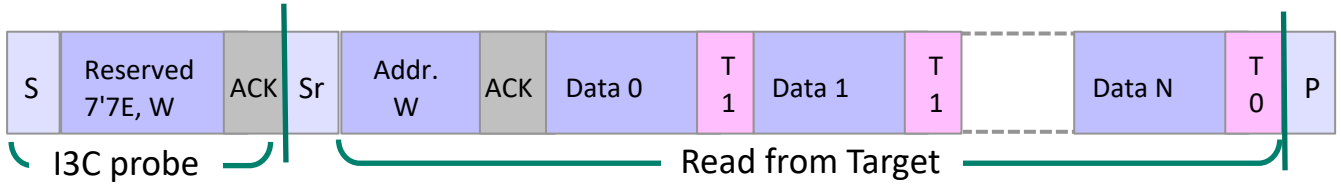


(*) SCL can be in Hi-Keeper mode in some short transitions, e.g., during controller handover
(+) Optimal compatibility achieved if I2C devices have 50ns spike filter enabled

 © 2024 MIPI Alliance, Inc.

Private Bus Transaction





S START

Sr START repeated

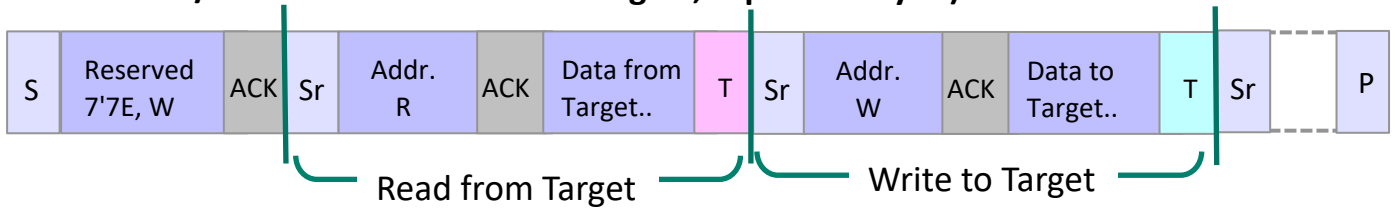
P STOP

T-bit use

T More data bit in READ transactions

T Parity bit in WRITE transactions

Private Read/Write with one or more Targets, separated by Sr)



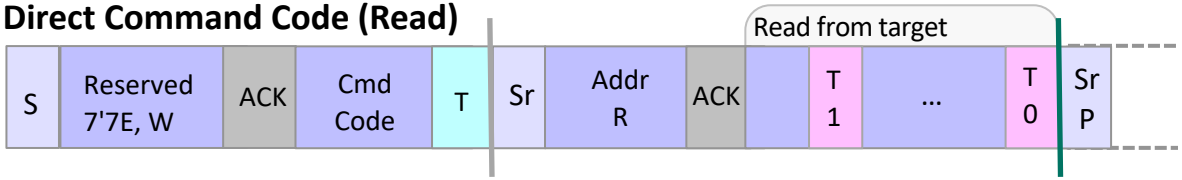
Common Command Codes (CCC)

- CCC is a standardized command set to facilitate bus management functions
- 57 CCCs defined + vendor extensions
  - Required (16)
  - Optional
  - Conditional
- Broadcast CCCs (controller to bus)
  - For faster setting of parameters (write only)
  - Global or group address
- Direct CCCs (controller to/from target)
  - **Read:** retrieval of target parameters
  - **Write:** setting of target parameters

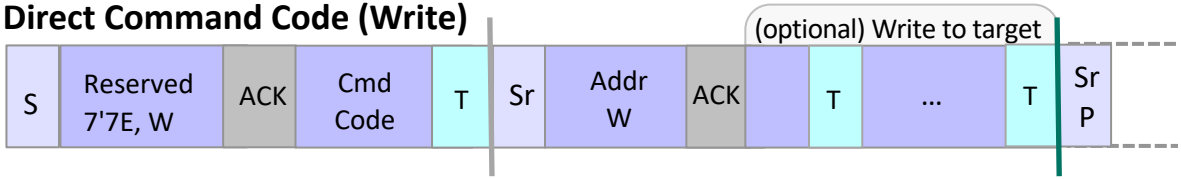
Management Function	Command Code
Bus events	ENEC/DISEC, RSTACT, GET/SETXTIME, ENTASx (x=0..3)
Dynamic address	ENTDAA, RSTDAA, SETDASA, SETNEWDA, SETGRPA, RASTGRA, DEFTGTS, DEFGRPA, GETCAPS, SETAASA RSTACT, GETPID
Data transfer	GET/SETMWL, GET/SETMRL, ENDXFER
Enter HDR mode	ENTHDRx (x=0..5)
Multi-lane	MLANE
Enquire about target capability	GETBCR, GETDCR, GETSTATUS, GETMXDS, GETCAPS
Routing, bridging	SETROUT, SETBRGTGT
Device-to-device transfer	D2DXFER
Set bus context for higher-level protocols	SETBUSCON

Common Command Codes

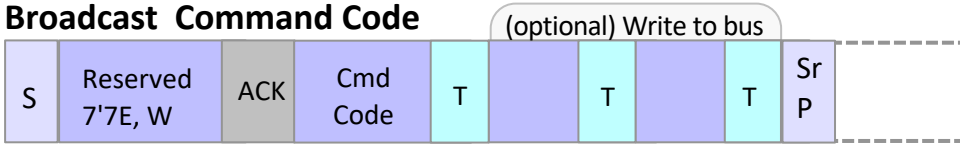
Direct Command Code (Read)



Direct Command Code (Write)



Broadcast Command Code



Dynamic Address in MIPI I3C

- **Primary Controller executes a Dynamic Address Assignment (DAA) at bus initialization**
- **DAA allows to tie address-to-function of a device, simplifying system management & SW drivers**
- **Targets must have a 7-bit address to function**
  - Pre-assigned in I2C, optionally in I3C, too
- **Available addresses: 103**
  - Extra 9 addresses, conditional to presence of I2C
- **Addresses not available:**
  - Reserved addresses for I3C management
  - Exclude reserved I2C address for compatibility
  - If I2C target present, remove I2C static address from list
- **Address type:**
  - Physical for single device
  - Virtual (multiple end-points inside a physical device)
  - Group
- **A set of CCCs is available for address management**

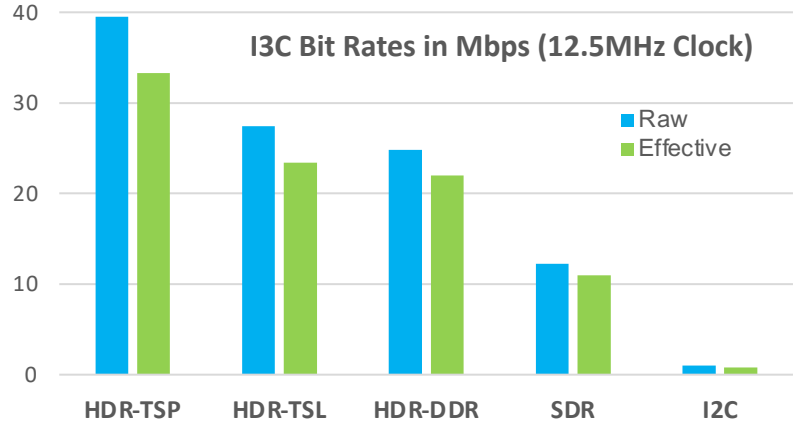
CCCs Dedicated to Address Management		
CCC	Function	Type
ENTDAA	Executes DDA assignment for the entire bus	Broadcast
RSTDAA	Resets all assigned addresses	Broadcast
SETDASA	Assigns a DA to static address	Direct
SETNEWDA	Changes an assigned DA to a new DA of a target	Direct
SETGRPA RSTGRPA	Assigns/resets group address assignment of a target	Direct
DEFTGTS DEFGRPA	Addresses management across controllers	Broadcast
GETCAPS	Get target capability	Direct

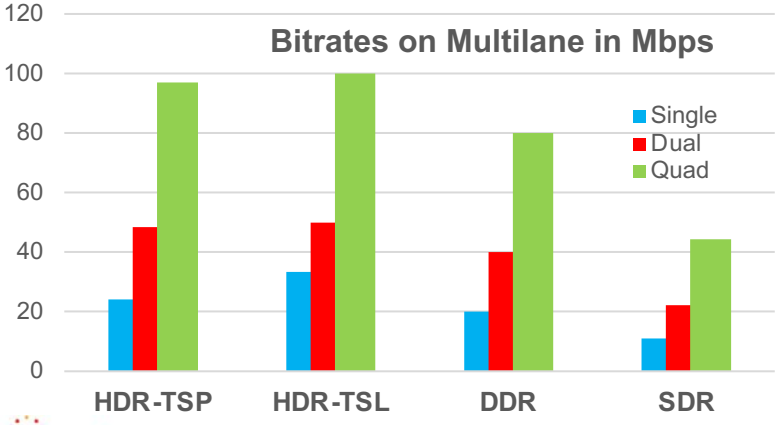
 © 2024 MIPI Alliance, Inc.

High Data Rates in MIPI I3C

- **High Data Rate (HDR) modes are an optional feature**
  - More bit per same clock frequency, more efficient
  - No change in Max Frequency and no EMI increase
  - Block-oriented transport, with CRC check: suitable for safety requirements
- **HDR is entered with the CCC: ENTHDRn n=0-7**
- **Four modes are defined:**
  - Mode 0 : HDR-DDR Double Data Rate
  - Mode 1 : HDR-TSP Ternary Symbol for Pure (I3C) bus
  - Mode 2 : HDR-TSL Ternary Symbol Legacy (I2C) inclusive bus
  - Mode 3 : HDR-BT Bulk Transfer
- **Additionally, multiple SDA data lane modes are possible for HDR-DDR and HDR-BT modes**
  - SDA x2 , SDA x4
- **Target can selectively support HDR modes**
  - Unsupported modes are ignored

Bus Bit Rates





Effective Bitrates (Mbps)			
Line mode	Single SDA	Dual SDA	Quad SDA
SDR	11.1	22.2	44.4
HDR-DDR	20	40	80
HDR-BT	24.24	48.48	96.96
HDR-TSL	23.52	n/a	n/a
HDR-TSP	33.33	50	100

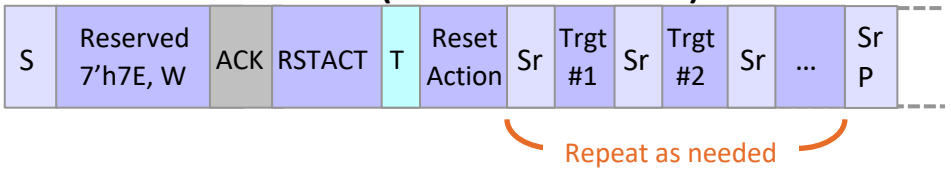
- **I3C performance vs. I2C**
  - **SDR mode:** 11 times faster
  - **HDR-DDR mode:** 22 times faster
  - **HDR-TSP:** 33 times faster

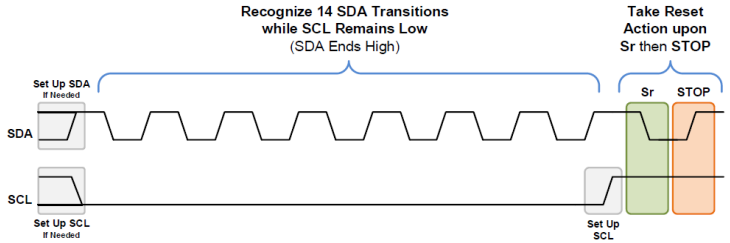
Bus Reliability Features

Targets Reset mechanism

- Executed with a unique HW pattern
- Target(s) can be individually set (using a CCC command)
- Programmable targets reset action:
  - Wake-up device, Reset peripheral, Reset whole Target, no-action

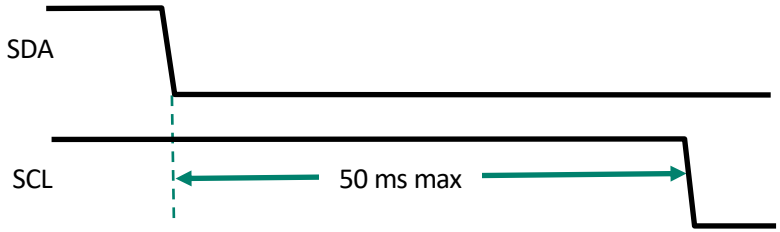
Reset Action Command (Broadcast or Direct)





Dead Bus Recovery

- Watchdog device monitors bus
- Check active controller status by pulling SDA low (IBI initiation)
- If it malfunctions:
  - Start the Bus Recovery Procedure as spec'ed



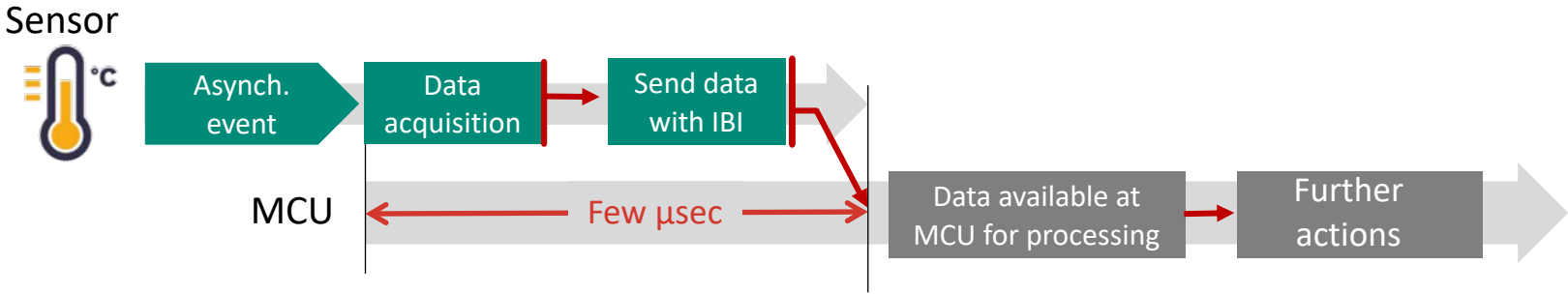
 © 2024 MIPI Alliance, Inc.

Data Acquisition with Improved Timing Precision

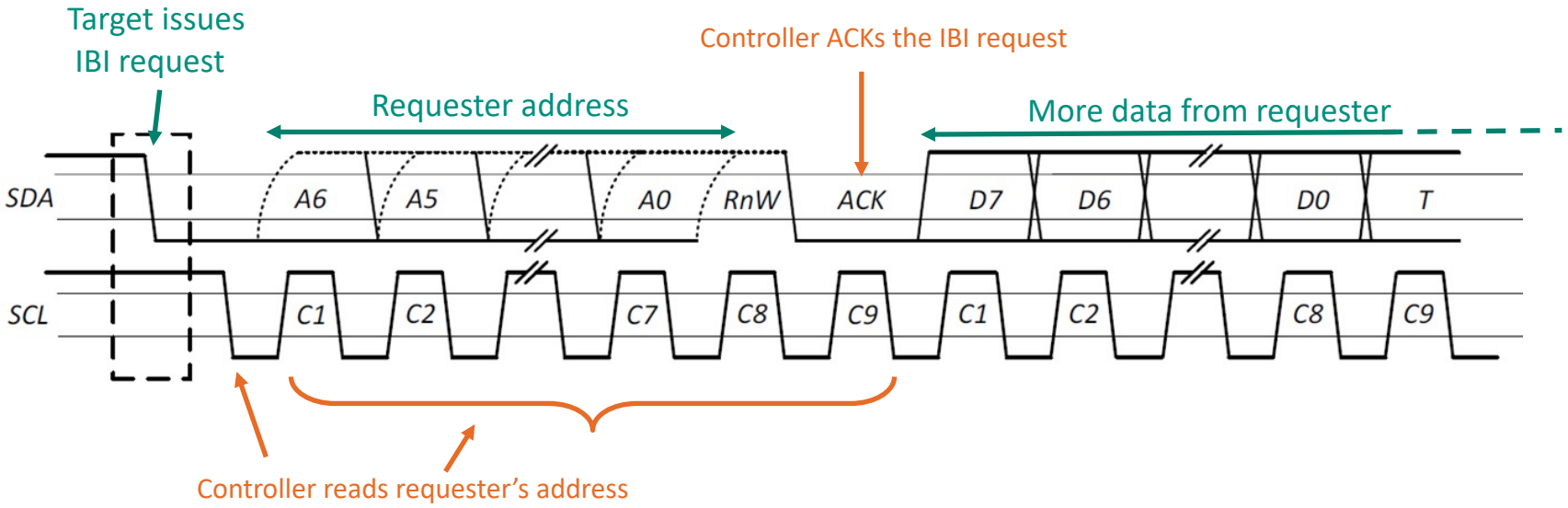
Data Acquisition with In-Band Interrupt (IBI)

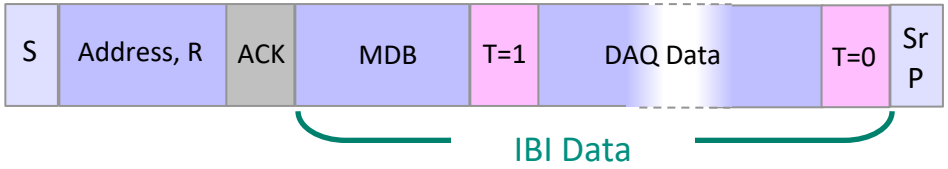
IBI allows fast and efficient asynch data acquisition, and data processing:

- Data produced by a target is transferred to upstream controller for further processing
- IBI intended as an efficient mechanism for targets to grab a controllers' attention
- Avoids extra dedicated wires, or inefficient polling mechanism
- Interrupt priority based on address
- Complete MDB interrupt table at:
  mipi.org/mipi_i3c_mandatory_data_byte_values_public



In-Band Interrupt (IBI) Sequence





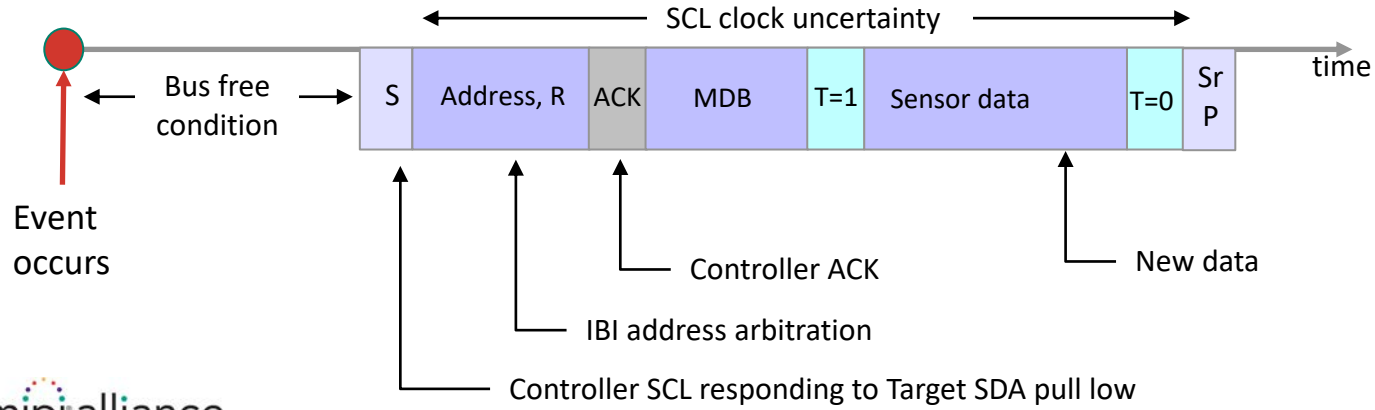
Data from Target IBI:

- MDB Mandatory Data Byte: (3+5 bits) predefined groups, extension
- (optional) Additional data terminated by T=0

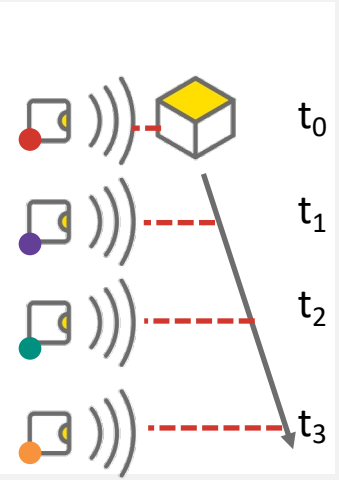
Improving Timing Precision of Events

- Some applications need precise time information of when an event occurred (example: distance measurement)
- Target reports the event with the In-Band-Interrupt (IBI)
- **I3C Timing Control** removes uncertainties of timing information due to bus activity, and adds a time-stamp of the event

Major sources of uncertainty:

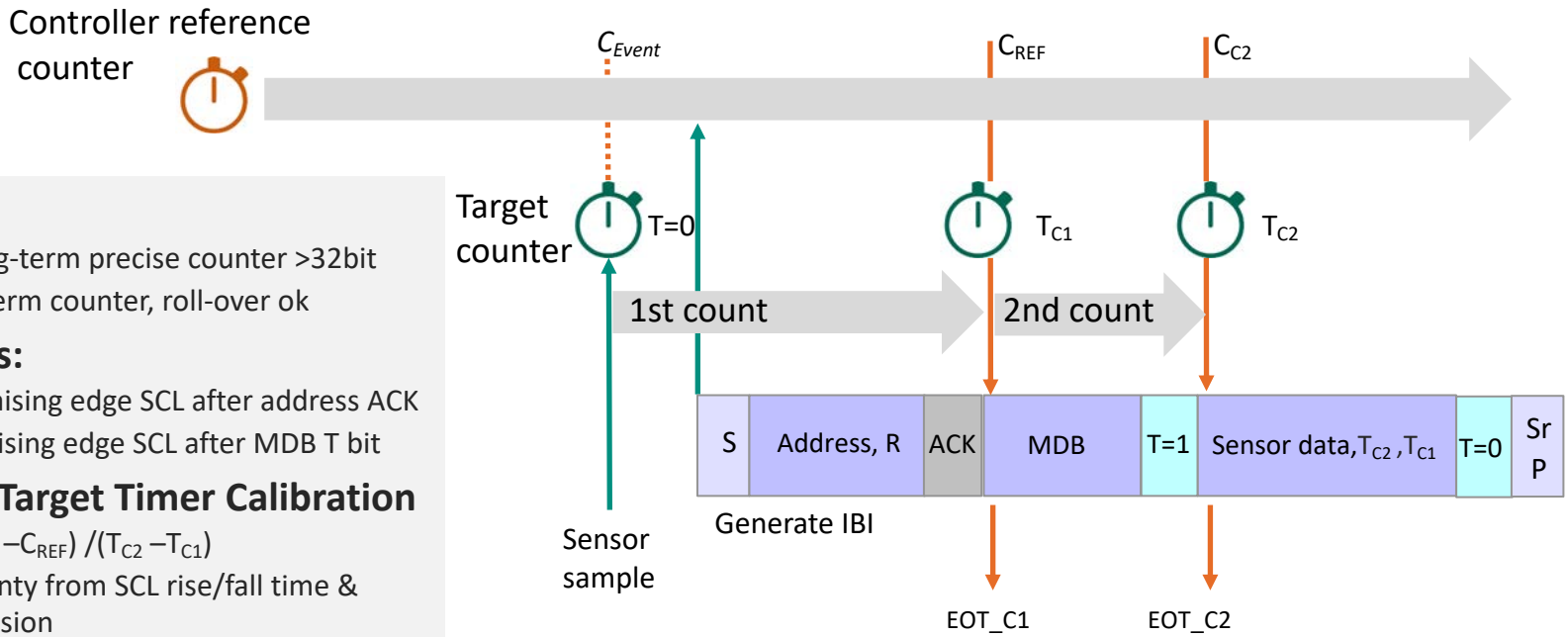


Tracking a moving object



Asynchronous Timing Control Mode0

- **Required:**
  - Controller long-term precise counter >32bit
  - Target short-term counter, roll-over ok
- **Time Markers:**
  - EOT_C1 : 1st raising edge SCL after address ACK
  - EOT_C2: 1st raising edge SCL after MDB T bit
- **Controller & Target Timer Calibration**
  - Using cal=(C_{C2} –C_{REF}) /(T_{C2} –T_{C1})
  - Main uncertainty from SCL rise/fall time & counters precision
  - Faster counters can improve precision
- **Additional Modes (1-3) for higher precision**



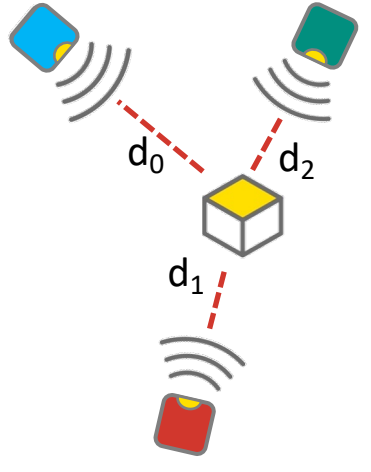
$C_{Event} = C_{REF} \times (C_{C2} - C_{REF}) / (T_{C2} - T_{C1})$

Precision ~100ns

Timing Control for Synchronous Systems

- **Synchronous system can be used when:**
  - Multiple sensors (targets) provide periodically sampled data
  - Targets collect data at synchronized times
- **Main advantages**
  - Robust against target oscillator inaccuracies and drift
  - Data can be read sequentially without affecting timing precision
  - Allows data collected from different sensors to be fused
- **Sync Tick** by the controller as the main synchronization mechanism

Object localization



 © 2024 MIPI Alliance, Inc.

Sequence for Synchronization of Events

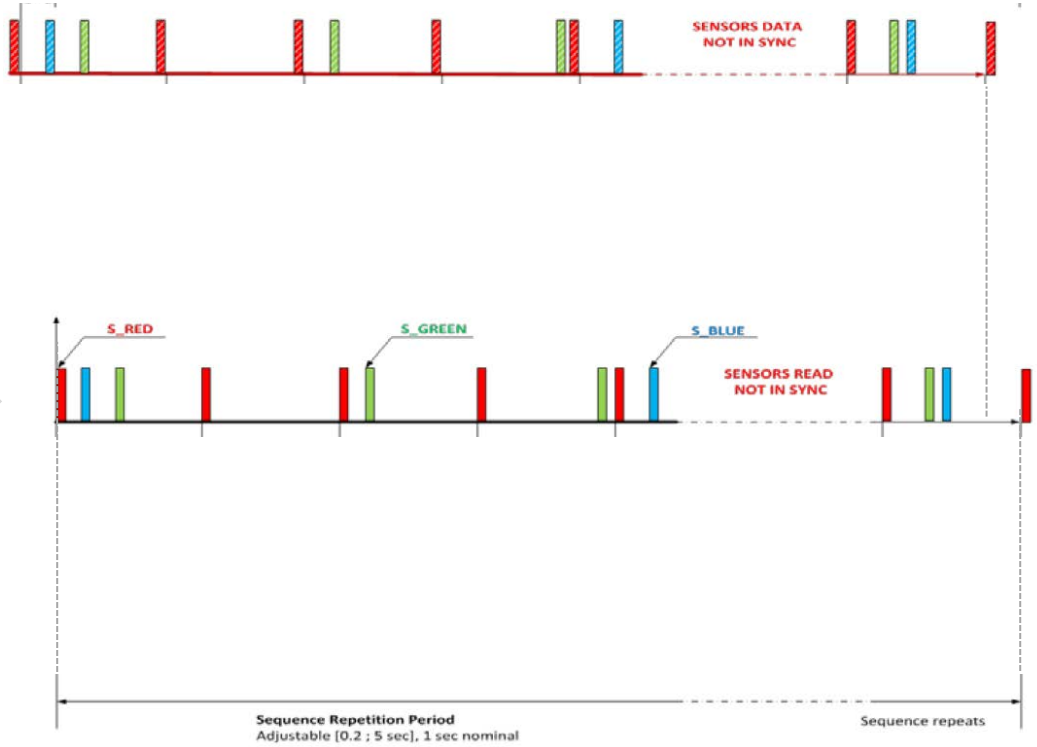
For S_RED, S_GREEN, S_BLUE sensors

Sampling

Sensors sampling not synchronized

Readouts

Sensors reads not synchronized



 © 2024 MIPI Alliance, Inc.

Sequence for Synchronization of Events

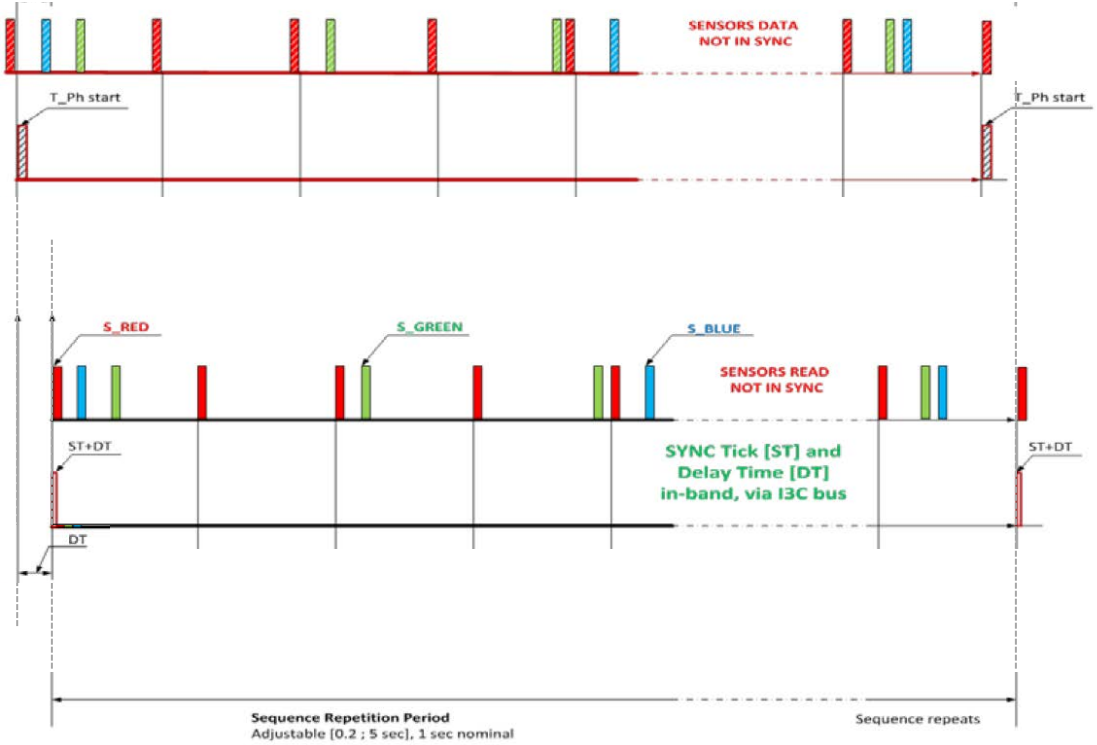
For S_RED, S_GREEN, S_BLUE sensors

Sampling

- Sensors sampling not synchronized
- Time & Phase (vs. Sync Tick) provided with SETXTIME

Readouts

- Sensors reads not synchronized
- Sync Tick & Delay Time provided with SETXTIME



Sequence for Synchronization of Events

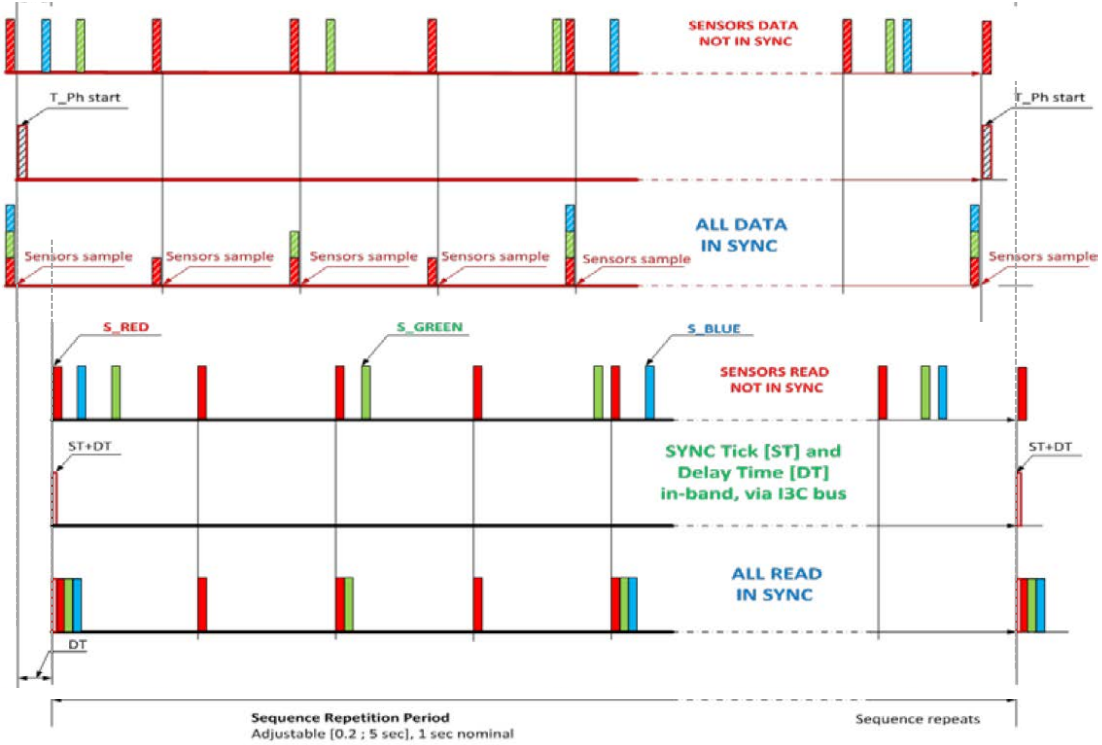
For S_RED, S_GREEN, S_BLUE sensors

Sampling

- Sensors sampling not synchronized
- Time & Phase (vs. Sync Tick) provided with SETXTIME
- Sensors sample in Sync according to the ODT

Readouts

- Sensors reads not synchronized
- Sync Tick & Delay Time provided with SETXTIME
- Sensors read in in Sync according to the ODT



 © 2024 MIPI Alliance, Inc.

Timing Control (TC) Command Codes

Synchronous Timing Control uses two CCCs:

- **GETXTIME – Get Exchange Timing Information**
  - Query target for Exchange Timing support: Which TC is supported, TC rate, target frequency accuracy
- **SETXTIME – Get Exchange Timing Information, with sub-code**
  - **ST**: Indicates to target which bus event should be used at **Sync Tick**.
    - Typically, SDA falling edge of START event with SETXTIME/ST
  - **DT**: Delay time between event sampling and readouts
  - **TPH**: Repetition period of synchronization event
  - **ODR**: Output data rate, how often a sensor should do a readout (every Sync Tick, etc.)

 © 2024 MIPI Alliance, Inc.

Summary

- **MIPI I3C has many innovative features:**
  - Command Codes, Dynamic addressing, device classifications
- **Suited for demanding applications requiring full control of a system:**
  - Reliability: Error Recovery, dead-bus recovery, and Target reset
  - Precision: Asynchronous and Synchronous Timing control

***I3C** is not just a faster, lower-power bus, with **advanced system management and precision timing features, it enables new, state-of-the-art control applications** that are not possible using legacy interfaces.*

 © 2024 MIPI Alliance, Inc.

Get Involved / Resources



Download the Public MIPI I3C Basic specification

www.mipi.org/mipi-i3c-basic-download



Check out the MIPI I3C webpage

www.mipi.org/specifications/i3c-sensor-specification



Join the MIPI I3C Working Group

Open to MIPI Contributor Members
Meets Wednesdays @ 08:00 PST



Contact the MIPI I3C Working Group

i3c@mipi.org (members)
admin@mipi.org (non-members)

Questions

Thanks

Contact: michele.scarlatella@team.mipi.org

 © 2024 MIPI Alliance, Inc.