# I3C

Presentation 1

# Index

- Introduction
- Basic Terminology
- Why we need I3C
- Features of I3C
- Advantages/Unique features of I3C
- Fundamental data
- Single Data Rate mode
- Bus Information in detail
- I3C SDR packets read and writes

# INTRODUCTION

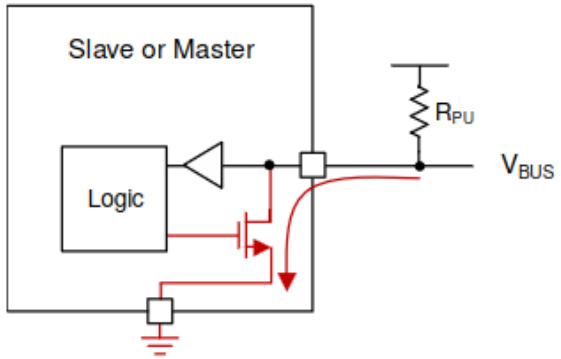
# Introduction

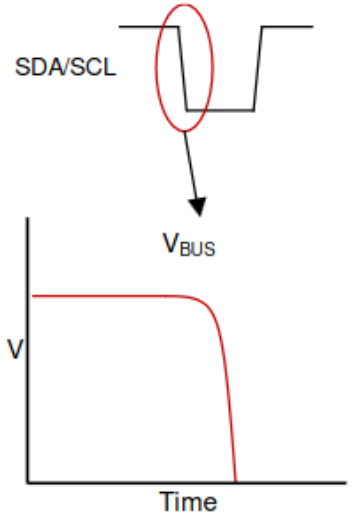
- I3C is developed by MIPI alliance (Mobile Industry Processor Interface).
- There are two versions of I3C
  - I3C Full specification
  - I3C Basic specification
- I3C basic, is
  - Feature reduced
  - Lower complexity
  - Royality free version of I3C
- It is important to know that first I3C full spec was released then I3C basic spec was released for general public.

# BASIC TERMINOLOGY

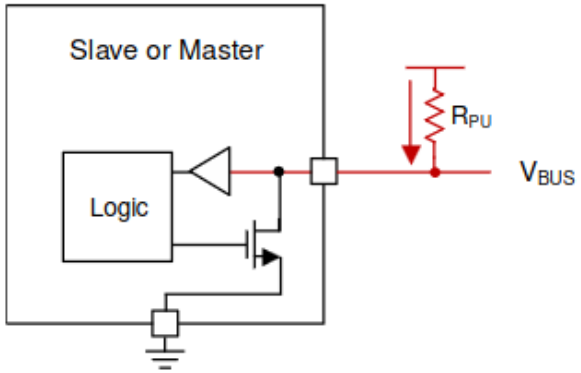
# Open Drain

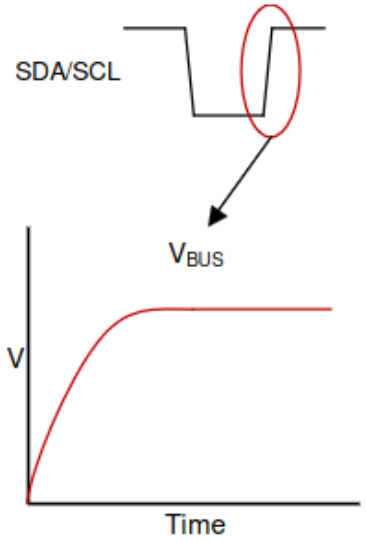
## Pulling Bus low (Start)

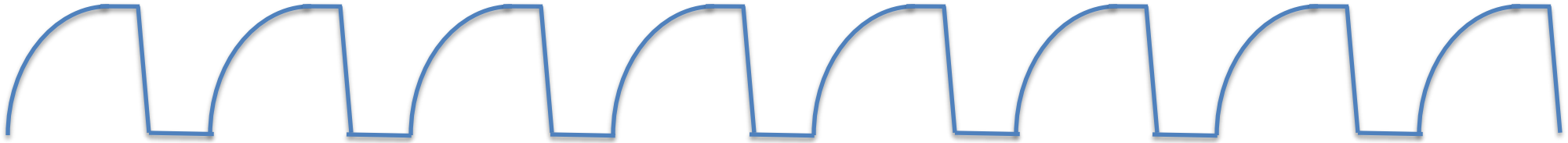



## Releasing the Bus (Stop)

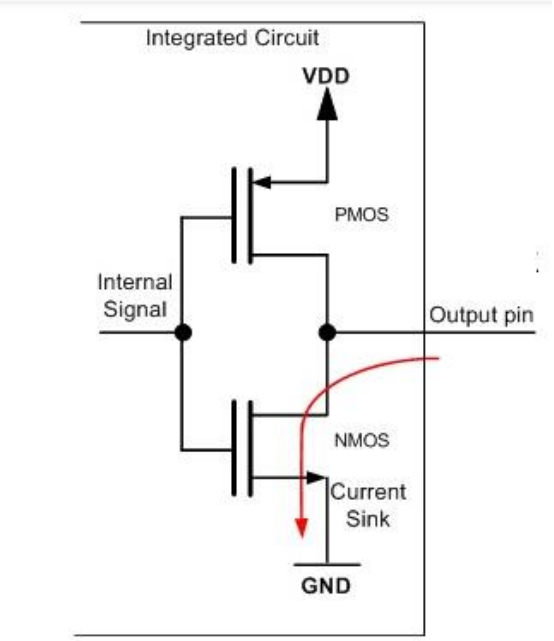


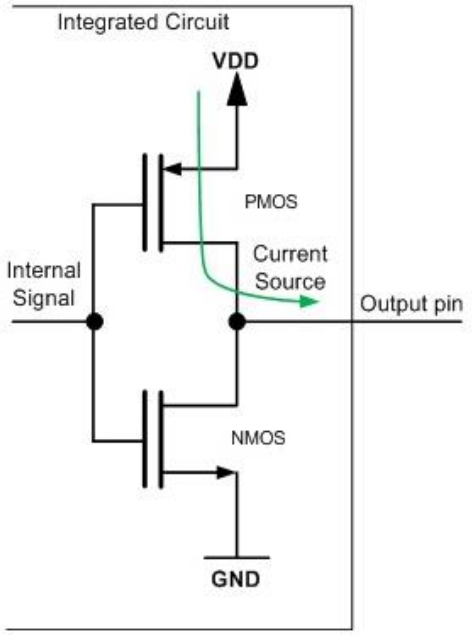



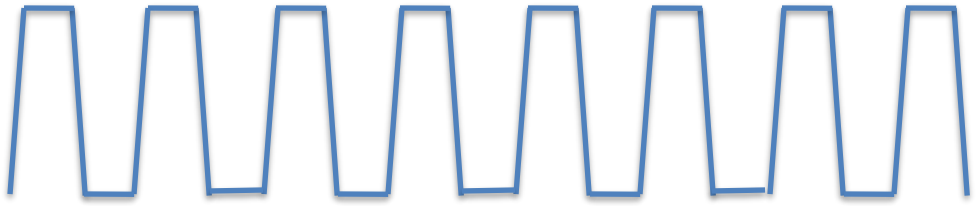
# Push Pull

**Pulling bus low (Start) (Pull)**




**Releasing**



# Push Pull vs Open drain

## Open Drain

- Slower rise time  i.e. slower slope.
- More power consumption.
- Used only as output configuration.
- **Can be used for bidirectional single line interface**

## Push Pull

- Faster rise time i.e. faster slope.
- Less power consumption.
- Can be used as both input and output by  Hi-z.
- **Cannot be used for bidirectional single line interface.**

I3C
is
Plain and Powerful

# WHY WE NEED I3C

# Why we need I3C ?

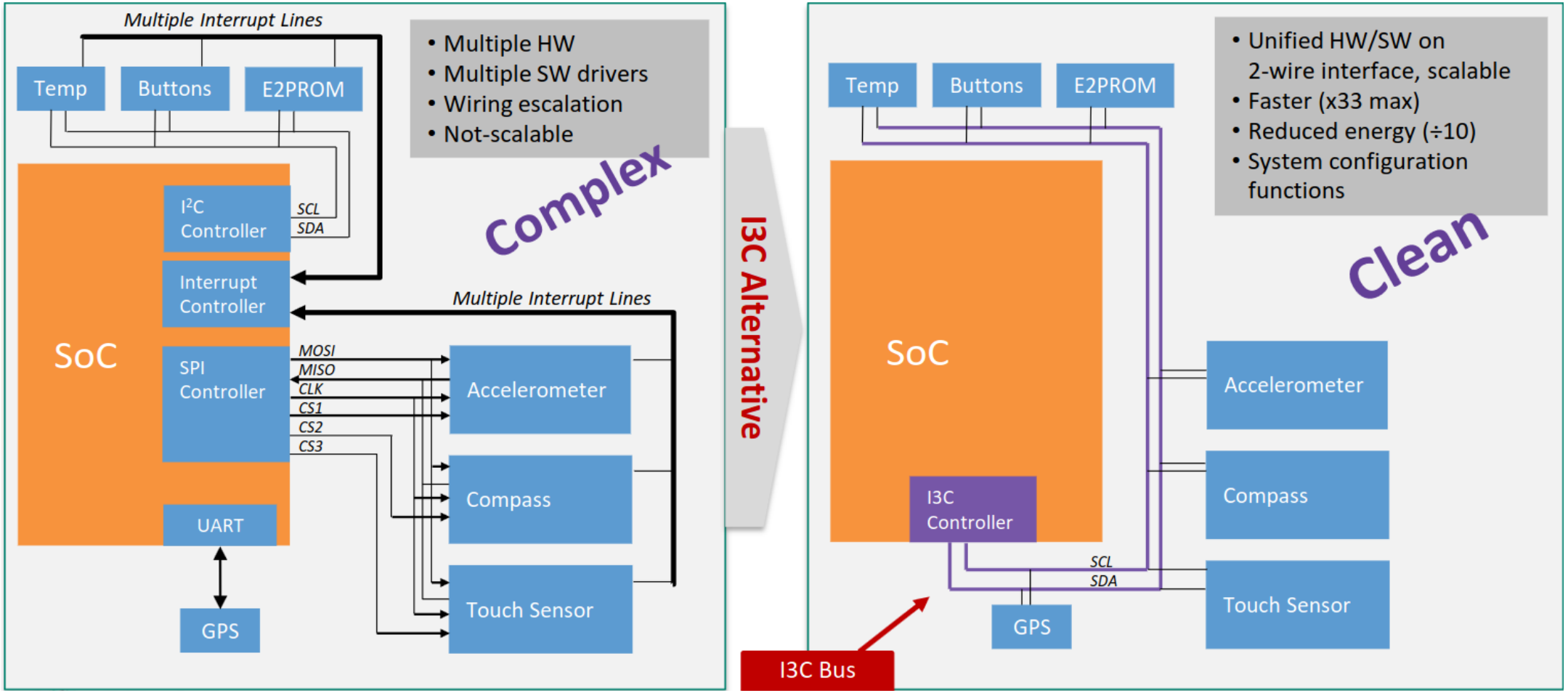
Disadvantages of I2C

- Low speed
- Only open drain mechanism
- Only static addressing which causes issues
- No inbuilt interrupt mechanism
- No dynamic detection of devices

Disadvantages of SPI

- More wires needed
- More power consumption
- No flow control and ACK mechanism
- No dynamic addressing
- No inbuilt interrupt mechanism

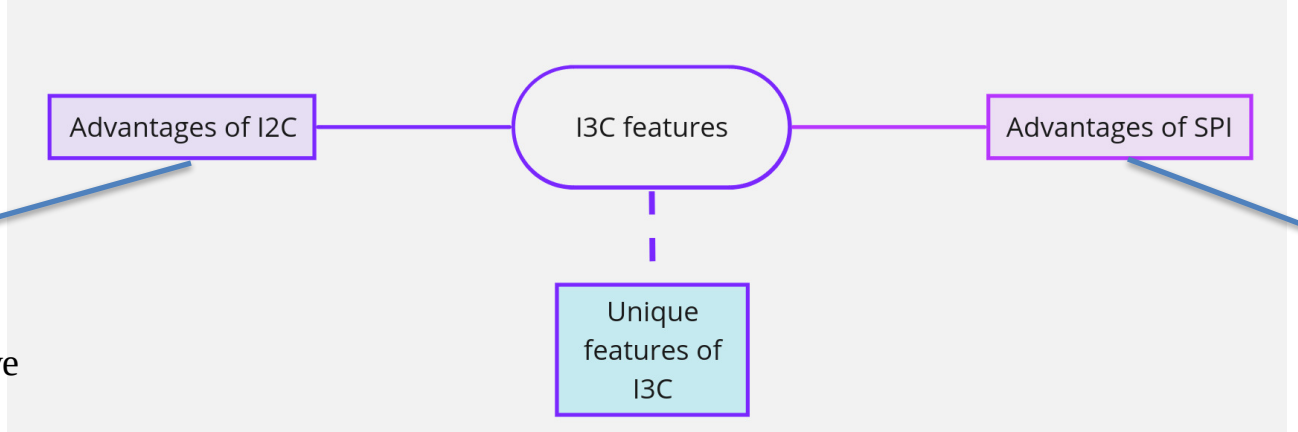
# Need a better and unifying alternative, replacing I2C, SPI, SMBus and UART



# FEATURES OF I3C

# Features of I3C

- I3C mainly focuses upon
  - Use of little energy as possible in transporting data and control
  - Reduce the number of physical pins used by interface




1. Only two pin count
2. Multi-master and Multi-slave
3. Simple packet Structure
4. Half-duplex

1. Higher speed and throughput
2. Push Pull driver mech.

# ADVANCED/UNIQUE FEATURES OF I3C

# Advanced/Unique features of I3C

- Two-wire serial interface up to **12.5 MHz** using Push-Pull.
- **Dynamic Addressing** while supporting Static Addressing for Legacy I2C Devices.
- **Multi-Drop** a multitude of physically and virtually attached Devices.
- Supports **Multi- Lane** data Transfer.
- Broadcast and Direct **Common Command Codes**.
- Different **Data Modes**.
- **Multi-Controller** capability.
- **In-Band Interrupt (IBI)** support.
- **Hot-Join** support **(HJ)**.
- Asynchronous Time Stamping (for Mode 0 only).
- Grouped Addressing.
- Target Reset without additional wires.
- Supports Legacy I2C devices and existence.

# FUNDAMENTAL DATA

# Fundamental data

- Two wires
  - SDA → Serial Data which is bi-directional data pin, either controller or target.
  - SCL → Serial Clock which is either a clock pin or Bi-directional data pin in certain HDR modes, always by controller.
  - SCL maximum frequency is 12.5 MHz .
  - Standard Data Rate (SDR) : 11.1 Mbps (Mega bits per second) .
  - Maximum speed is up to 33.3Mbps.
  - Operating voltage: 1.2 V – 3.3 V.
- Two types of devices
  - Controller.
  - Target.

# I3C Target (Slave)

- I3C target responds to the commands of controller
- Every device must have the target capability on an I3C bus
- These are two types : Normal , SDR-only
- Additionally it can support the following
  - Request In-Band Interrupts
  - Generate Hot-Join events
  - Request to become Active Controller, if the I3C Target Device also has I3C Controller Device capability. (i.e., an I3C Secondary Controller Device)
  - Support any combination of I3C's defined HDR Modes.

# I3C Controller (Master)

- An I3C controller is a device that is capable of controlling the I3C Bus.
- A given I3C Bus always has one Controller (active) and one or more Targets.
- Controller has both controller and target capabilities.
- Controllers have different device roles
  - Primary Controller -> Normal and SDR-only
  - Secondary Controller -> Normal and SDR-only
- Active controller concept -> only one controller shall be active at a time.

# Responsibilities of I3C controller

- Should generate I3C commands, I2C messages if required.
- Generation of bus clock in SDR, HDR-DDR and HDR-BT mode.
- Management of power structures.
- Dynamic Address assignment.
- Response for I3C target requests along with address arbitration
  1) In-band interrupts 2) Hot join events 3) Controller Role request
- Support I3C SDR mode at least.
- Support I2C Legacy target devices.

# Controller Roles

- Active Controller
  - The I3C Device that presently has control (i.e., the Controller Role) of the I3C Bus.
- Primary Controller
  - The Controller-capable I3C Device that initializes the I3C Bus and performs configuration of all Target Devices. It acts as the authority for the Bus in its initial state, and becomes the first Active Controller once the Bus is configured.
- Secondary Controller
  - Initially acts as a Target.
  - It can accept the Controller Role from any Active Controller or it may request .
  - Once a Secondary Controller accepts the Controller Role it becomes the new Active Controller and have all privileges of active controller i.e. it can pass the controller role also.

# SINGLE DATA RATE (SDR) MODE

# Single Data Rate (SDR) Mode

- Default mode.
- Used to enter and configure everything in I3C.
- Supports legacy I2C target devices.
  - I2C traffic from an I3C Controller to an I2C Target will be properly ignored by all I3C Targets, because the I3C protocol is designed to allow I2C traffic.
  - I3C traffic from an I3C Controller to an I3C Target will not be seen by most Legacy I2C Target Devices, because the I2C Spike Filter is opaque to I3C's higher clock speed.

# BUS CONFIGURATION

# Bus Configuration

- It deals about types of devices that are connected.
- Defines the device roles and responsibilities
  - It means what the device is expected to do in their given role and how the configuration impacts performance.
    1. I3C Target can have a Static Address but also support Dynamic Addressing.
    2. A Device shall not have a 50 ns Spike Filter enabled when used in an I3C Bus operating at full clock speed.
    3. Only Full mode / Full Mode + speed is allowed on I3C for I2C devices
- How to obtain device characteristics.

# Bus Configuration

## Virtual Target

- A physical Device that represents multiple I3C Targets that shares peripherals.

## I3C Bridge device

- Device that allows conversion from the native I3C bus protocol to another protocol (i.e. SPI, UART etc.)

## I3C Routing device

- Allows conversations between two or more different I3C Buses.
- The Routing Device buffers/queues the transactions, causing the Device to be non-transparent whereas Bridge device doesn't .

**Note** : there can be multiple Virtual Targets inside a single Device or a Bridge/Hub that manages the Bus connection.

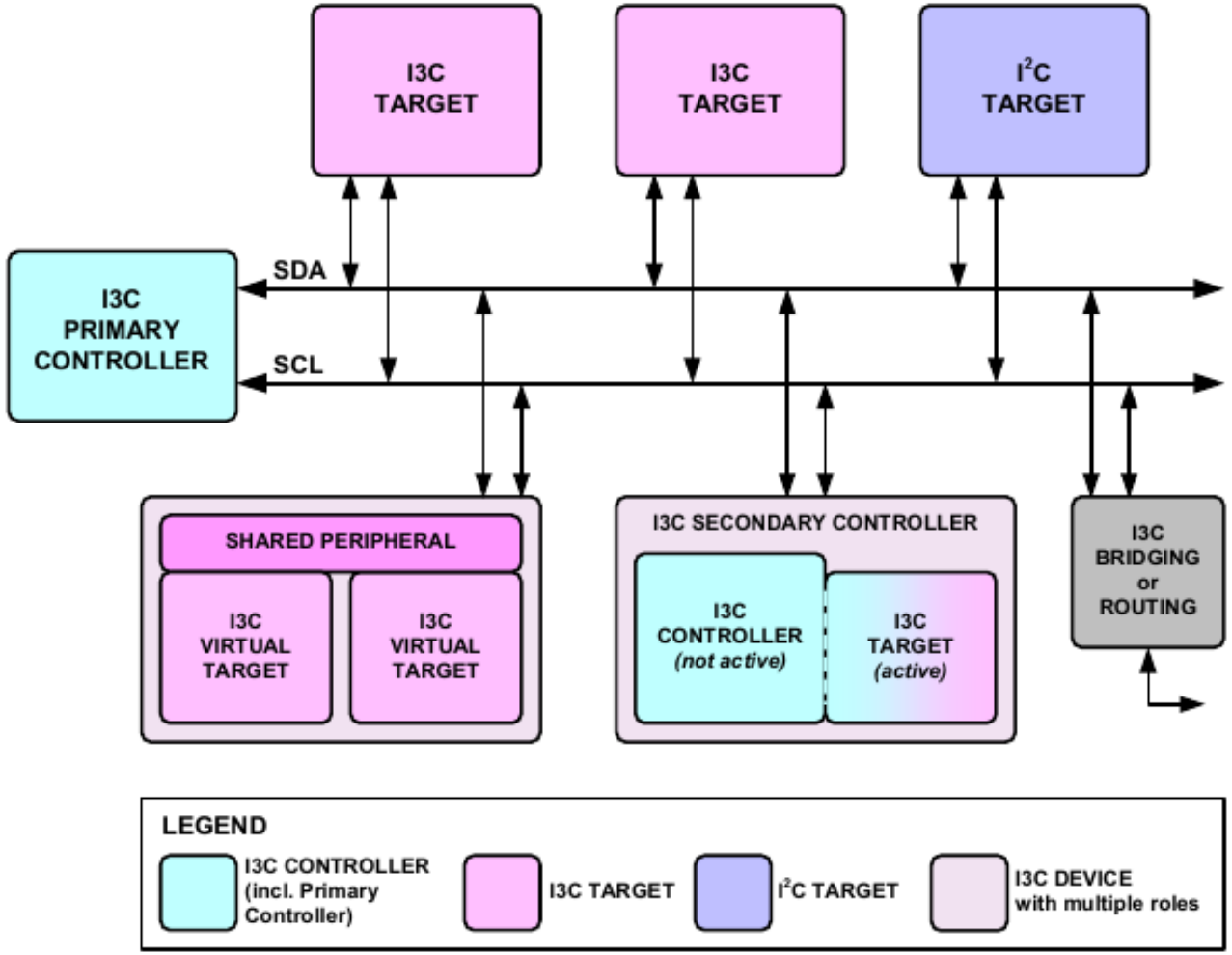


**Table 2 I3C Devices Roles vs Responsibilities**

| Responsibilities / Features | Comments | Roles | | | | | |
|---|---|---|---|---|---|---|---|
| | | Primary Controller | Secondary Controller | SDR-Only Primary Controller | SDR-Only Secondary Controller | Target | SDR-Only Target |
| **Manages SDA Arbitration** | For Address Arbitration, In-Band Interrupt, Hot-Join, Dynamic Address, as appropriate | Y | Y | Y | Y | N | N |
| **Dynamic Address Assignment** | Controller assigns Dynamic Address | Y | Optional[3] | Y | Optional[3] | N | N |
| **Hot-Join Dynamic Address Assignment** | Controller capable of Dynamic Address assignment after Hot-Join | Y | Optional[3] | Y | Optional[3] | N | N |
| **Self Dynamic Address Assignment** | Only Primary Controller can self-assign a Dynamic Address | Y | N | Y | N | N | N |
| **Static $I^2C$ Address**[1] | – | N/A | Optional | N/A | Optional | Optional | Optional |
| **Memory for Targets' Addresses and Characteristics** | Retaining registers | Y | Y | Y | Y | N | N |
| **HDR Target capable** | Supports being accessed in at least one HDR Mode | Y | Y | N | N | Y | N |
| **HDR Controller capable** | Supports Bus control in at least one HDR Mode | Y | Y | N | N | N/A | N/A |
| **HDR Exit Pattern Generation capable**[2] | Able to generate the HDR Exit Pattern on the Bus for error recovery | Y | Y | Y | Y | N | N |
| **HDR Tolerant** | Recognizes HDR Exit Pattern | Y | Y | Y | Y | Y | Y |

***Note:***

*1) A Static Address may be used to more quickly assign a Dynamic Address. See **Section 5.1.4**.*

*2) All Targets require an HDR Exit Pattern Detector, even Targets that are not HDR capable.*

*3) A Secondary Controller may assign a Dynamic Address using the ENTDAA CCC while it is the Active Controller (i.e., when it holds the Controller Role, per **Section 5.1.7.3.3**). Some Secondary Controllers might have limited capabilities or reduced functionality while acting as the Active Controller of the I3C Bus. See **Section 5.1.7.3.4** and **Section 5.1.7.3.6**. However, only the Primary Controller may assign Dynamic Addresses using the SETDASA and/or SETAASA CCCs (i.e., during Bus Initialization), per **Section 5.1.4.2**.*

**Table 3 I²C Features Allowed in I3C Targets**

| I²C Feature When Used on an I3C Bus | Required on I3C | Desirable on I3C | Not Used on I3C | Not Allowed on I3C | Note |
|---|---|---|---|---|---|
| Fm/Fm+ Speed | X | – | – | – | 2, 3, 4 |
| HS Speed | – | – | X | – | 3 |
| UFm Speed | – | – | X | – | 3 |
| Static I²C Address | – | X | – | – | – |
| 50 ns Spike Filter | – | – | X | X (Shall disable) | 3 |
| Clock Stretch | – | – | – | X | – |
| 20 mA Open Drain Driver | – | – | X | – | 1, 3 |
| Matches I²C AC Timing | – | – | X | – | 2, 3 |
| I²C Extended Address (10 bit) | – | – | X | – | 3 |
| I3C Reserved Addresses | – | – | – | X | – |

***Note:***

1) *See **Table 82** and **Table 85***
2) *I3C drive and timing requirements are different from I²C.*
3) *If an I3C Target has I²C features intended for use on an I²C Bus, then they will not be used on an I3C Bus. As stated in **Section 5.1.1.1**, once the Target sees a 7'h7E, it will disable I²C features that are not used by I3C.*
4) *For a Mixed Bus, timing requirements when communicating with I²C-only Devices may depend on the maximum SCL clock frequency supported by such I²C-only Devices. See **Table 84** and **Table 85**.*

**Table 4 Legacy I²C-Only Target Categories and Characteristics**

| Index Specific | I²C-Only Devices Index 0 | I²C-Only Devices Index 1 | I²C-Only Devices Index 2 |
|---|---|---|---|
| 50 ns IO Spike Filter[1] | Y | N | N |
| Max SCL clock frequency ($f_{SCL}$) tolerant[2] | N/A | Y | N |

***Note:***

1) *Allows tolerance of HDR Modes and SDR at SCL High periods of $t_{DIG\_H\_MIXED}$ or less*
2) *Allows compliance up to maximum SDR SCL clock frequency ($f_{SCL}$)*

# Bus Configuration

- I3C Characteristics Registers describe and define an I3C compatible Device's capabilities and functions on the I3C Bus.
- Devices without I3C Characteristics Registers shall not be connected to a common I3C Bus.
- Three Characteristics Register types
  - Bus Characteristics Register (BCR)
  - Device Characteristics Register (DCR)
  - Legacy Virtual Register (LVR)

# Bus Characteristics Reg.

- Read Only register
- It describes the I3C compliant Device's role and capabilities for use in DAA and CCC
- Mandatory for every connected I3C device on I3C bus

**Table 5 Bus Characteristics Register (BCR)**

| Bit | Name | Description | Notes |
|---|---|---|---|
| BCR[7] | Device Role[1] | 2'b00: I3C Target<br>2'b01: I3C Controller capable<br>2'b10: Reserved for future definition by MIPI Alliance I3C WG<br>2'b11: Reserved for future definition by MIPI Alliance I3C WG | 1 |
| BCR[6] | Device Role[0] | | |
| BCR[5] | Advanced Capabilities | 1: Supports optional advanced capabilities. Use GETCAPS CCC (***Section 5.1.9.3.19***) to determine which ones.<br>0: Does not support optional advanced capabilities | 2 |
| BCR[4] | Virtual Target Support | 0: Is not a Virtual Target and does not expose other downstream Device(s)<br>1: Is a Virtual Target, or exposes other downstream Device(s) | 3 |
| BCR[3] | Offline Capable | 0: Device will always respond to I3C Bus commands<br>1: Device will not always respond to I3C Bus commands | 4 |
| BCR[2] | IBI Payload | 0: No data bytes follow the accepted IBI<br>1: One data byte (MDB) shall follow the accepted IBI, and additional data bytes may follow; see also the Set/Get Maximum Read Length CCC (***Section 5.1.9.3.6***). Data byte continuation is indicated by T-Bit per ***Section 5.1.2.3.4***.<br>See also ***Section 5.1.8*** on use of IBI Payloads for Timing Control. | – |
| BCR[1] | IBI Request Capable | 0: Not Capable<br>1: Capable | – |
| BCR[0] | Max Data Speed Limitation | 0: No Limitation<br>1: Limitation | 5 |

*Note:*

*1) The BCR Device Role bits for any I3C Device capable of acting as I3C Controller (either Primary Controller or Secondary Controller) are 2'b01. The Primary Controller identifies itself to the Targets with the DEFTGTS CCC (**Section 5.1.9.3.7**); its Static Address is the value 7'h7E.*

*2) In I3C v1.0, bit BCR[5] only indicated HDR support, and that the Controller should use the GETHDRCAPS CCC to determine what HDR Modes were supported. In I3C v1.1 and above, bit BCR[5] now indicates that the Controller should use the same CCC (now renamed GETCAPS) to determine what advanced features the I3C Device supports, including (but no longer limited to) HDR. Note that the I3C v1.0 method is fully interoperable with the I3C v1.1 and above method.*

*3). In I3C v1.0, bit BCR[4] only indicated Bridge Devices required to comply with the I3C Specification. In I3C v1.1 and above, bit BCR[4] now indicates that the Device is a Virtual Target or exposes other downstream Devices, and that the Controller should use the GETCAPS CCC with Defining Byte VTCAPS (**Section 5.1.9.3.19**) to determine what features and capabilities the I3C Device supports, including (but no longer limited to) Bridge Devices.*

*4) Offline Capable Devices retain the Dynamic Address, and are specified in **Section 2.2**.*

*5) Controller shall use the GETMXDS CCC to interrogate the Target for specific limitation.*

# Device Characteristics Reg.

- Read Only register
- It describes the I3C compliant Device type for use in DAA and CCC.
- E.g. Accelerometer, Gyroscope etc.
- Mandatory for every connected I3C device on I3C bus

**Table 6 I3C Device Characteristics Register (DCR)**

| Bit | Name | Description |
|---|---|---|
| DCR[7] | Device ID[7] | 255 available codes for describing the type of sensor, or Device.<br><br>Examples: Accelerometer, gyroscope, composite Devices.<br><br>Default value is 8'b0: Generic Device |
| DCR[6] | Device ID[6] | |
| DCR[5] | Device ID[5] | |
| DCR[4] | Device ID[4] | |
| DCR[3] | Device ID[3] | |
| DCR[2] | Device ID[2] | |
| DCR[1] | Device ID[1] | |
| DCR[0] | Device ID[0] | |

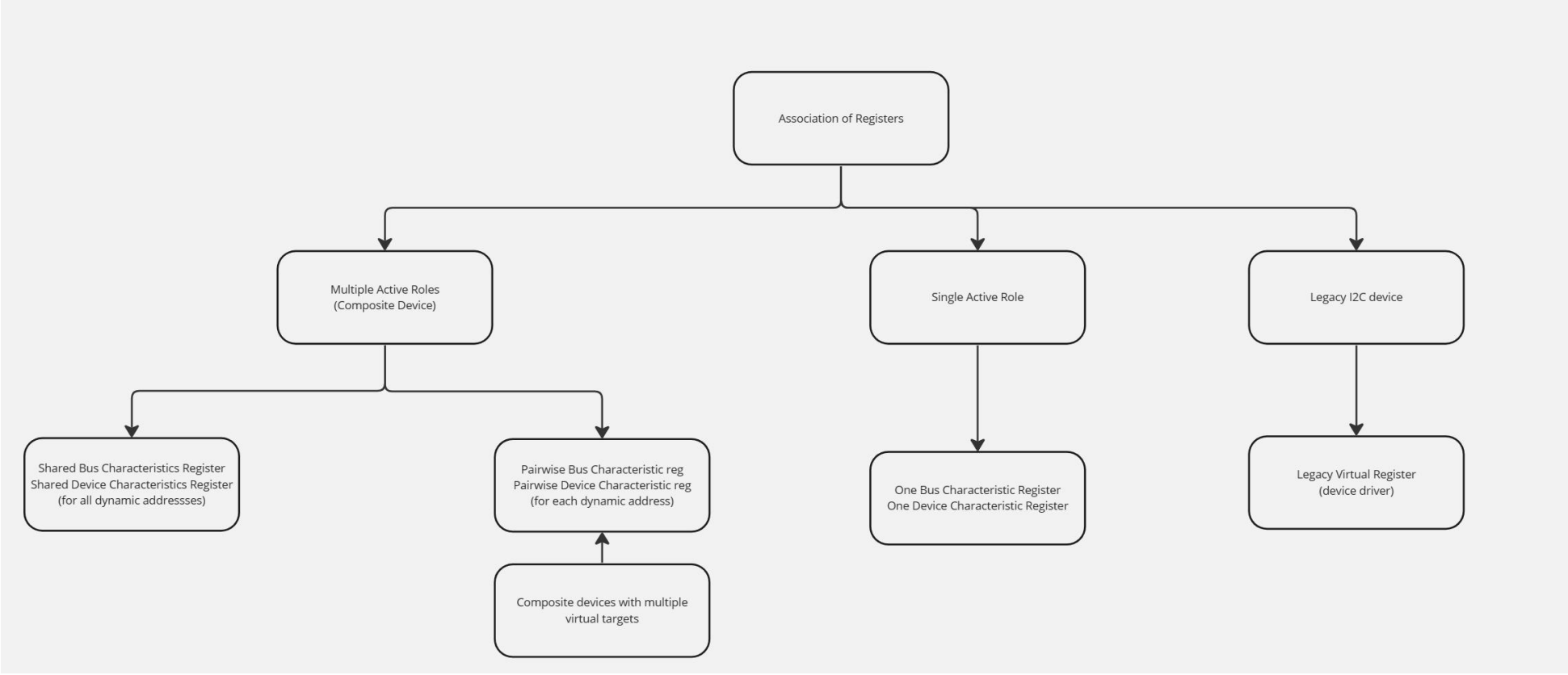
# Legacy Virtual Reg.

- Read Only register
- Only mandatory register for legacy I2C devices.
- Describes about key features like allowed modes and maximum SCL clock freq.
- Written in device driver and known to primary controller prior to bus configuration.

**Table 7 Legacy I²C Virtual Register (LVR)**

| Bits | Name | Values |
|---|---|---|
| **LVR[7:5]** | **Legacy I²C-Only [2:0]** per ***Table 4*** | **3'b000:** Index 0: Spike Filter, Max SCL clock freq tolerant N/A<br>**3'b001:** Index 1: No Spike Filter, Max SCL clock freq tolerant<br>**3'b010:** Index 2: No Spike Filter, Not Max SCL clock freq tolerant |
| | | **3'b011 – 3'b111:** Index 3 – 7: Reserved |
| **LVR[4]** | **I²C Mode Indicator** | **0:** I²C Fm+<br>**1:** I²C Fm |
| **LVR[3:0]** | MIPI Alliance I3C WG Reserved | **0:** Reserved |
| | | **1 – 15:** 15 available codes for describing the Device capabilities and function on the sensors' system. |

# Association of Characteristics registers



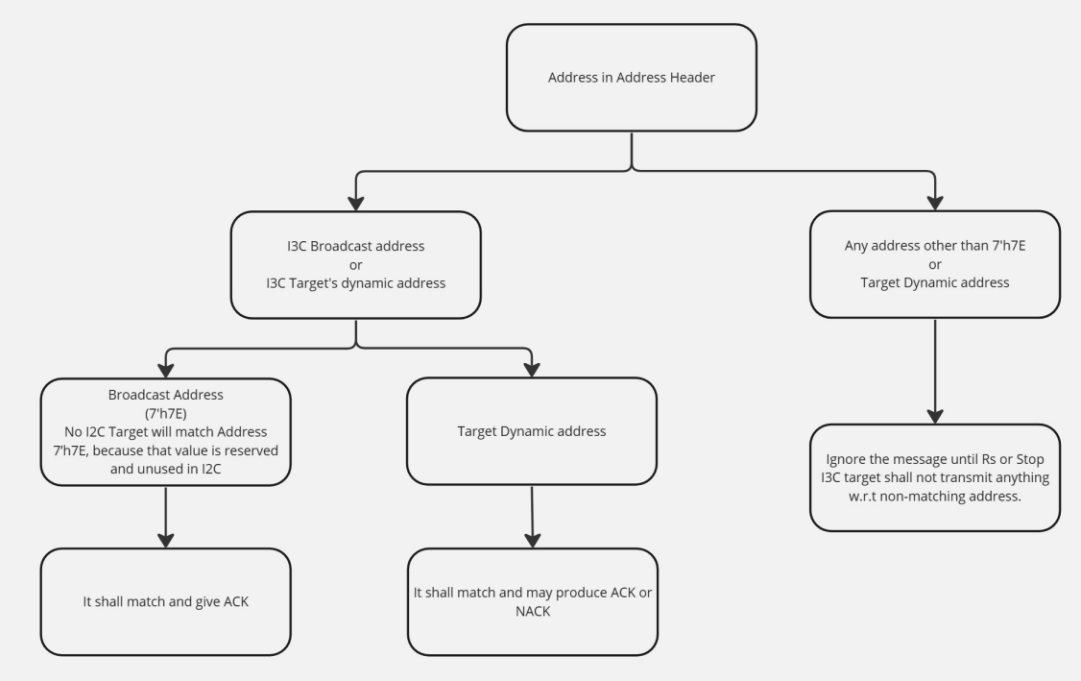
# BUS COMMUNICATION

# Bus Communication

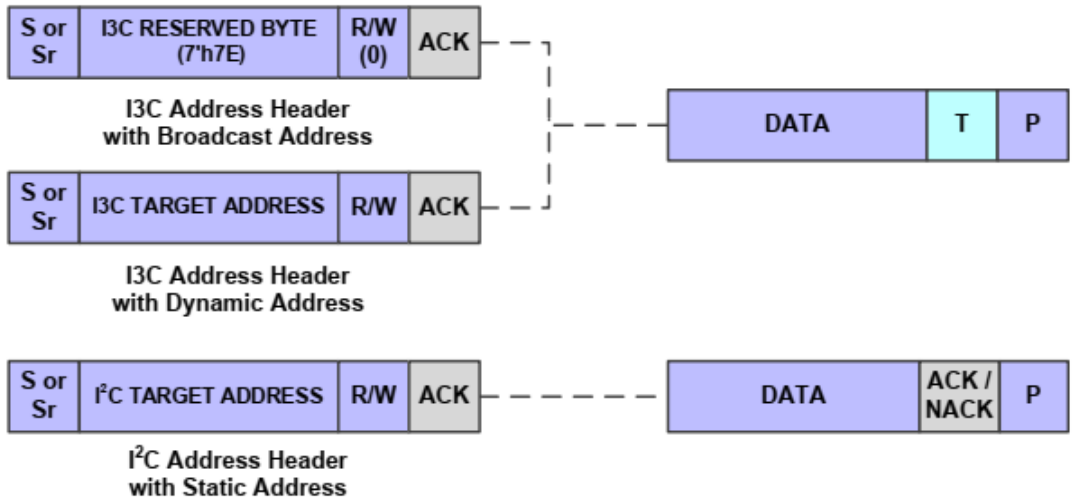
- Elaborates about I3C address header in SDR protocol
- Role of I3C target
- I3C address header
- I3C SDR Data words
- Use of Clock speed to prevent legacy I2C devices from seeing I3C traffic
- Controller Clock stalling

# I3C address header in SDR protocol

## Identification of I3C SDR message




## I2C vs I3C address header & 9th bit difference



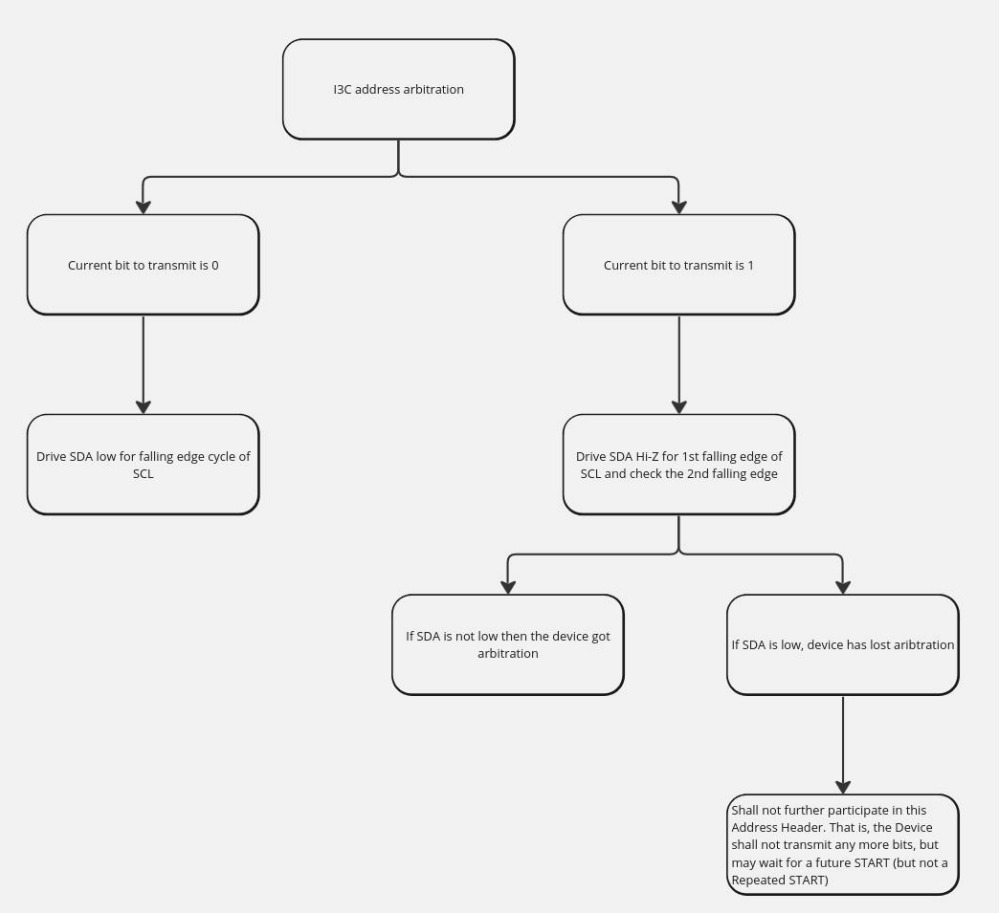
# Role of an I3C target

The role defines the behavior of the target with respect to the

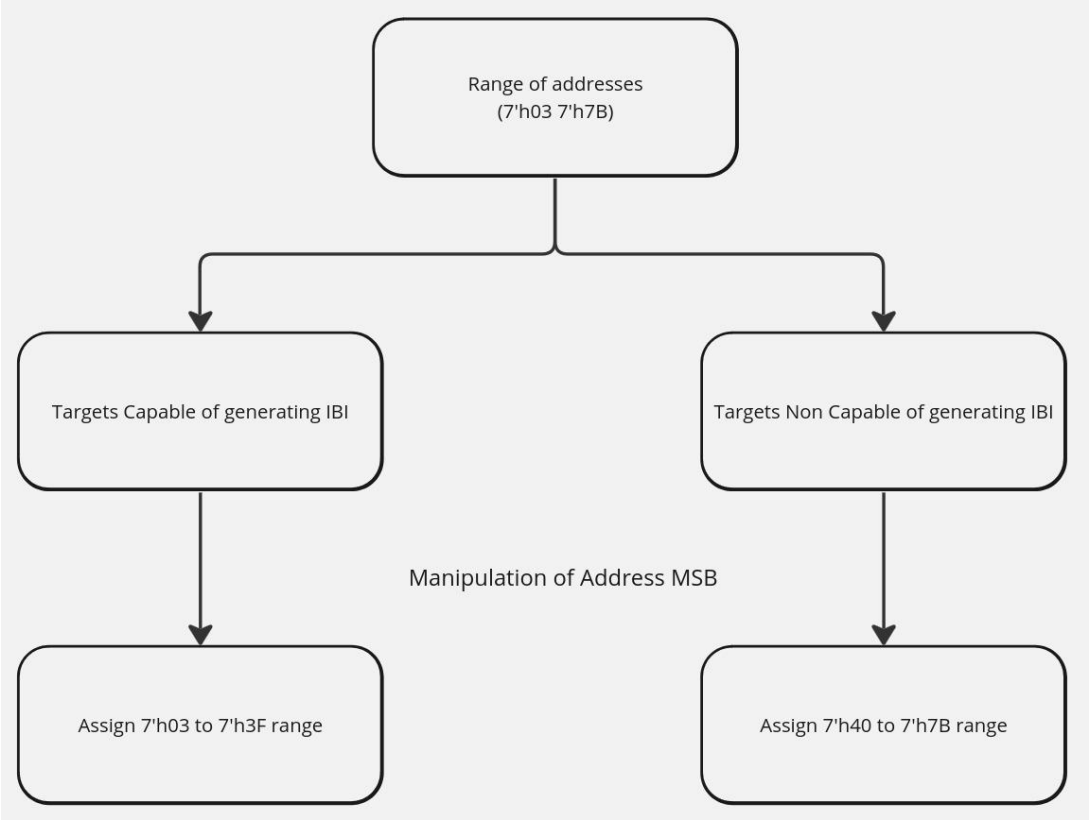
- Type of the CCC received
  - Broadcasted CCC
  - Directed CCC
- Address Header matching
- Type of address that the target is having
  - Role of I3C target acting as an I2C target with static address
- Type of the device i.e. Composite devices and Virtual targets
- Support of Group addressing

# I3C address header

- I3C address header follows Start or Repeated Start
- The format is the same as I 2C: 7 bits of Address, 1 bit of RnW, and 1 bit of ACK/NACK.
- Address arbitration
  - Address Header following a START (but not a Repeated START) is subject to Arbitration.
  - Which means both the Controller and one or more Targets may attempt to drive an Address onto the Bus, using SDA.
  - Such Address Headers are defined as Arbitrable Address Headers.
  - Arbitration model follows open-drain approach.



- Address optimization
  - During arbitration controller has no way to detect whether I3C target is sending address.
  - Split the addresses into two halves based on IBI capability.
  - Now optimize based upon address MSB bit (6th bit)



- Controller transmitting 1 on SDA, if same in next rising edge of SCL then push pull remaining header.
- Controller/Target transmitting 0 on SDA, if same in next rising edge of SCL then open drain remaining header.
- Adjusting clock width depending upon I2C communication visibility.

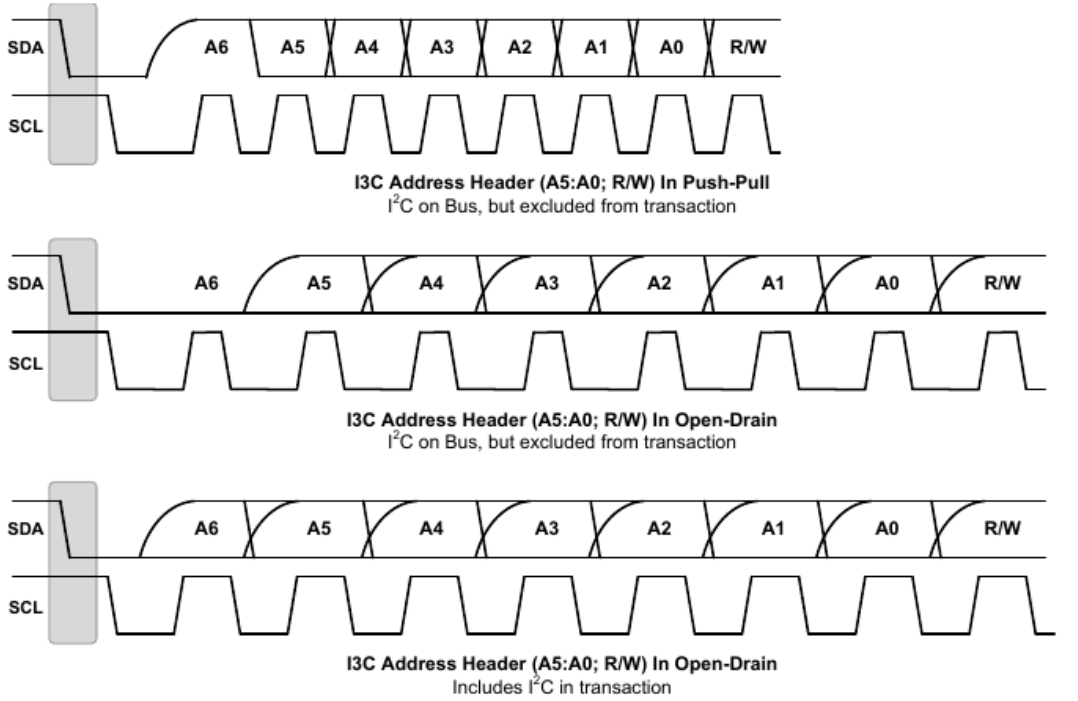



**Figure 13 Address Arbitration During Header**

- Consequence of Controller Starting a Frame with I3C Target Address.
  - Arbitration mechanism.
- Address Header Following a Repeated START is Push-Pull.
  - I3C Controller following a Repeated START shall not be arbitrated.
- I3C Target Address Restrictions
  - Reserved address, Addresses leading to single bit error and if I2C devices are present then those unusable addresses cannot be used.

**Table 8 I3C Target Address Restrictions**

| Target Dynamic Address | | Restriction | Description |
|---|---|---|---|
| Binary | Hex | | |
| 000 0000 | 7'h00 | Shall not use | I3C Reserved |
| 000 0001 | 7'h01 | Shall not use | I3C Reserved: For use with SETDASA CCC in special Point-to-Point Communication. See ***Section 5.1.9.3.10***. |
| 000 0010 | 7'h02 | Shall not use | I3C Reserved: Hot-Join Address |
| 000 0011 | 7'h03 | Optional | Marked 'Reserved' by I²C |
| 000 0100 | 7'h04 | Conditional | Available for use only if no Legacy I²C Devices supporting I²C "High-Speed Mode" are present on the Bus |
| 000 0101 | 7'h05 | | |
| 000 0011 | 7'h06 | | |
| 000 0011 | 7'h07 | | |
| 000 1000<br>011 1101 | 7'h08 – 7'h3D | Available for use | 54 Addresses |
| 011 1110 | 7'h3E | Shall not use | I3C Reserved: Broadcast Address single bit error detect |
| 011 1111<br>101 1101 | 7'h3F – 7'h5D | Available for use | 31 Addresses |
| 101 1110 | 7'h5E | Shall not use | I3C Reserved: Broadcast Address single bit error detect |
| 101 1111<br>110 1101 | 7'h5F – 7'h6D | Available for use | 15 Addresses |
| 110 1110 | 7'h6E | Shall not use | I3C Reserved: Broadcast Address single bit error detect |
| 110 1111<br>111 0101 | 7'h6F – 7'h75 | Available for use | 7 Addresses |
| 111 0110 | 7'h76 | Shall not use | I3C Reserved: Broadcast Address single bit error detect |
| 111 0111 | 7'h77 | Available for use | 1 Address |
| 111 1000 | 7'h78 | Conditional | Available for use only if no Legacy I²C Devices are present on the Bus that both a) support I²C "Extended Address Mode", and b) either have an Extended Address, or would be impacted by the Extended Address mechanism |
| 111 1001 | 7'h79 | | |
| 111 1010 | 7'h7A | Shall not use | I3C Reserved: Broadcast Address single bit error detect |
| 111 1011 | 7'h7B | Conditional | Available for use only if no Legacy I²C Devices are present on the Bus that both a) support I²C "Extended Address Mode", and b) either have an Extended Address, or would be impacted by the Extended Address mechanism |
| 111 1100 | 7'h7C | Shall not use | I3C Reserved: Broadcast Address single bit error detect (Also not available for use if any Legacy I²C Devices supporting I²C "Device ID Mode" are present on the Bus.) |
| 111 1101 | 7'h7D | Conditional | Available for use only if no Legacy I²C Devices supporting I²C "Device ID Mode" are on the Bus |
| 111 1110 | 7'h7E | Shall not use | I3C Reserved: Broadcast Address single bit error detect (See Error Type TE0, ***Section 5.1.10.1.1***.) |
| 111 1111 | 7'h7F | Shall not use | I3C Reserved: Broadcast Address single bit error detect |

# I3C SDR data words

- In I3C SDR, the Data Words match I2C only in the sense that they are both 9 bits long.
- I3C SDR Data Words differ from I2C in three ways
  1. Handoff from Address ACK to SDR Controller Write Data.
  2. Ninth Bit of SDR Controller Written Data as Parity (odd parity).
     - XOR( Data[7:0], 1 ) at Tx and XOR (9 bits) shall be equal.
  3. Ninth Bit of SDR Target Returned (Read) Data as End-of-Data.
     - Target can control the amount of data it returns and also allows the Controller to abort the message.

# Types of I3C bus configurations

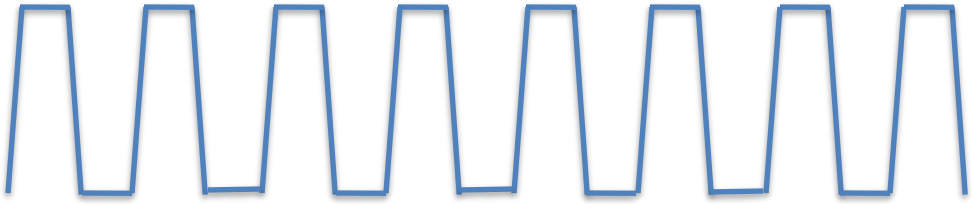
Three possible I3C bus configurations

1. Pure Bus
   - Only I3C devices are present.
2. Mixed Fast Bus
   - I3C, legacy I2C devices are present and I2C devices with true 50ns spike filter.
3. Mixed Slow/Limited Bus
   - I3C, legacy I2C devices are present and I2C devices without 50ns spike filter.

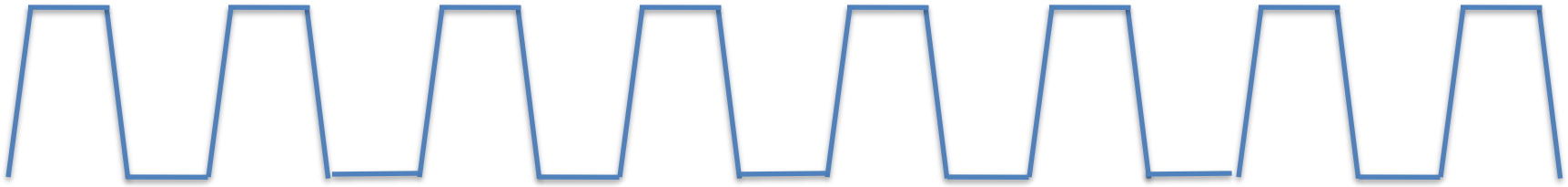
# Clock Speed to Prevent Legacy I2C Devices from Seeing I3C Traffic

- Useful in Mixed Fast Bus
- Use cases are
  - Accommodate legacy I2C Devices with Spike Filters that need a longer Low period in order to function properly.
  - Devices running at lower frequency but I2C legacy devices should not see that.
  - Changing the duty cycle will resolve the above two cases.

40 ns and
50% duty cycle
12.5 MHz

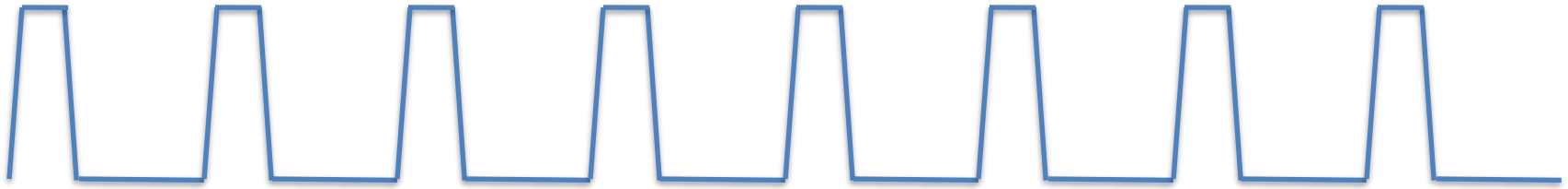


62.5 ns and
50% duty cycle
8 MHz



Should not simply lower frequency

40ns SDA high
85ns SDA low
32% duty cycle
8 MHz



Instead Use Duty cycle of clock for effective frequency

# Controller Clock Stalling

- Only controller is allowed for stalling clock.
- Stalling may be necessary because of two reasons
  - For fine grained data timing control i.e. the absolute or relative timing of a Message to a specific Target, or to all Targets
  - Synchronization of data
    - This may be due to parts of the Controller's system waking up in response to data, to changing state, or to otherwise needing time during a transaction.
- Four phases/conditions of stalling for various scenarios and maximum stall time for each scenario.

# BUS CONDITIONS

# Bus Conditions

- Bus Inactive Conditions
  - After a STOP, three defined timing conditions are used to create permission for a Controller or a Target to take defined actions leading to a START.
- Activity states
  - It's a mechanism for the Controller to inform Targets about expected upcoming levels of activity on the I3C Bus, in order to help the Targets better manage their internal states.

# Bus Inactive Conditions

1. Bus Free Condition
   - Defined as a period occurring after a STOP and before a START, and with the following duration

     Pure Bus : $t_{CAS}$

     Mixed Bus : $t_{BUF}$
2. Bus Available Condition
   - Bus free condition for at least $t_{AVAL}$
   - Minimum time for IBI and CRR
3. Bus Idle Condition
   - Bus free condition for at least $t_{IDLE}$
   - Minimum time for Hot-join events

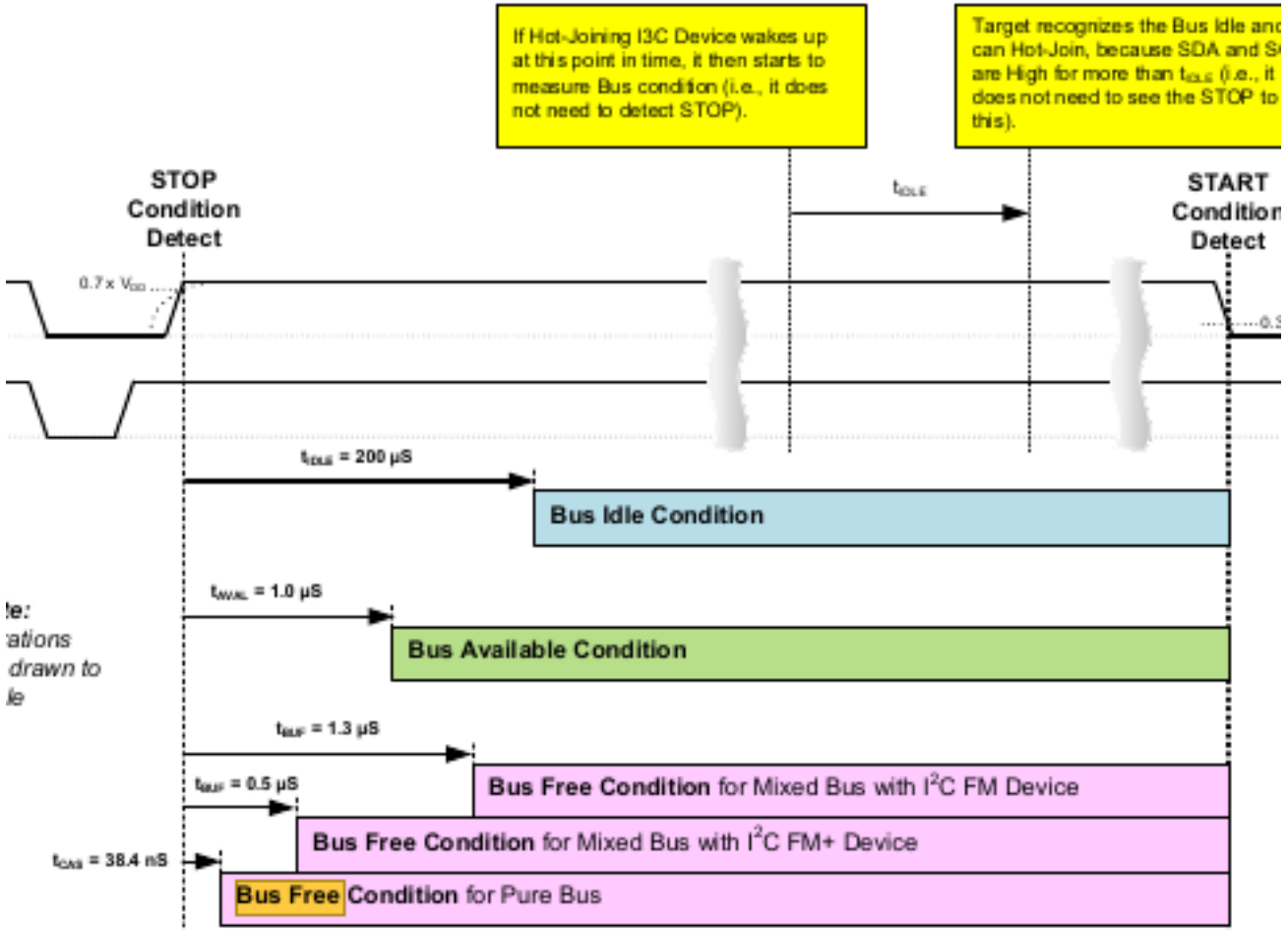



**Figure 24 Bus Condition Timing**

# Activity States

- Indicates response time of the controller to target once the target pulled SDA low.
- Hints the time for target to be ready after the activity interval.
- Used to adjust power savings, FIFO trigger levels, clock rates, NACKing if tx/rx before the interval etc.
- It is used typically used for latency information

**Table 11 Activity States**

| Activity State | Activity Interval | CCC |
|---|---|---|
| 0 | 1 µs | ENTAS0 |
| 1 | 100 µs | ENTAS1 |
| 2 | 2 ms | ENTAS2 |
| 3 | 50 ms | ENTAS3 |

# BUS INITIALIZATION & DAA

# Bus Initialization and Dynamic Address Assignment Mode

- Always done by Primary Controller
- Dynamic address is a unique device address that is assigned or allocated during initialization of the Bus. Usually occurs after power up.
- Dynamic Address shall be used in all subsequent transactions on the I3C Bus, until and unless the Controller changes the Dynamic Address (RSTDAA,SETNEWDAA).
- Once primary controller initializes, any active controller that supports Hot-join can do DAA .

- Sub-topics under this section are :
  - Device Requirements for DAA
  - Bus Initialization sequence with DAA
  - Provisioned ID Collision Detection and Correction
  - Group Address Assignment Procedure

# Device requirements for DAA

- Target that support broadcast CCC ENTDAA shall have 48 bit provisioned ID for DAA (not static address).
- Provisioned ID : Typically given by Vendor. Consists 3 parts
  - Bits[47:33]: MIPI Manufacturer ID
  - Bit[32]: Provisioned ID Type Selector
  - Remaining bits inform about Part ID, Instance ID, additional meaning like DCR values etc.
- Either 48-bit Provisioned ID or Static address is mandatory for each I3C device connected on I3C bus for initialization.

# Important CCC's

- ENTDAA : Enter Dynamic Address Assignment
  - This Broadcast CCC indicates to all I3C Devices that the Controller requires them to enter the DAA procedure
- SETDASA : Set Dynamic Address from Static Address
  - This CCC allows the Controller to assign a Dynamic Address to one Target using the Target's Static Address
- SETAASA : Set All Addresses to Static Address
  - This CCC allows the Controller to request that all connected Targets that have I2C Static Addresses use their I2C Static Address as their Dynamic Address
- SETNEWDA : Set New Dynamic Address
- SETAASA < SETDASA < ENTDAA (time to allocate Dynamic Address)

- RSTDAA : Reset DAA
  - Used to clear controller assigned DA/GA.
- SETGRPA : Set Group Address
  - Assign GA to one or more I3C targets to form a group.
- GETCAPS : Get Capabilities
  - Get the capabilities of I3C targets on I3C bus.
- DEFTGTS : Define List of Targets
  - Informs any Secondary Controllers about what Addresses, including any Group Addresses, have been assigned to the I3C Targets on the I3C Bus.
- DEFGRPA : Define List of Group Addresses
  - Informs any Secondary Controllers that support the Group Address feature about what I3C Targets on the I3C Bus are included in each Group Address.
- RSTGRPA : Reset Group Address
  - Removes one or more I3C Targets from a Group by resetting their assigned Group Address, disbands a single Group, or removes all Groups and Group Addresses.

## A.1 I3C CCC Transfers

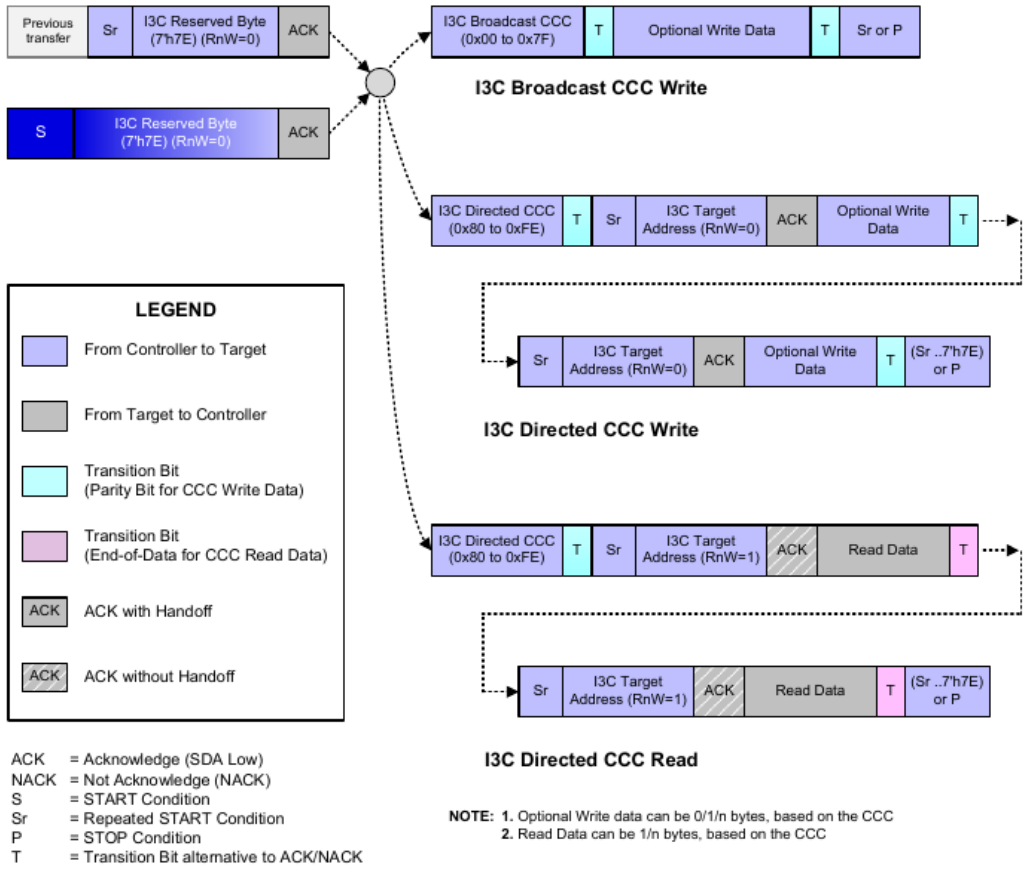



**Figure 157 I3C CCC Transfers**

# Bus Initialization Sequence with DAA

- Before DAA the primary controller shall have following data in its NV memory or through host.
  - The number of I3C compliant Devices that need to receive a Dynamic Address
  - The data for any I3C Devices resident on the I3C Bus that already have I2C Static Addresses.
  - The data for any Legacy I2C Devices resident on the I3C Bus.


**Figure 170 Dynamic Address Assignment FSM**

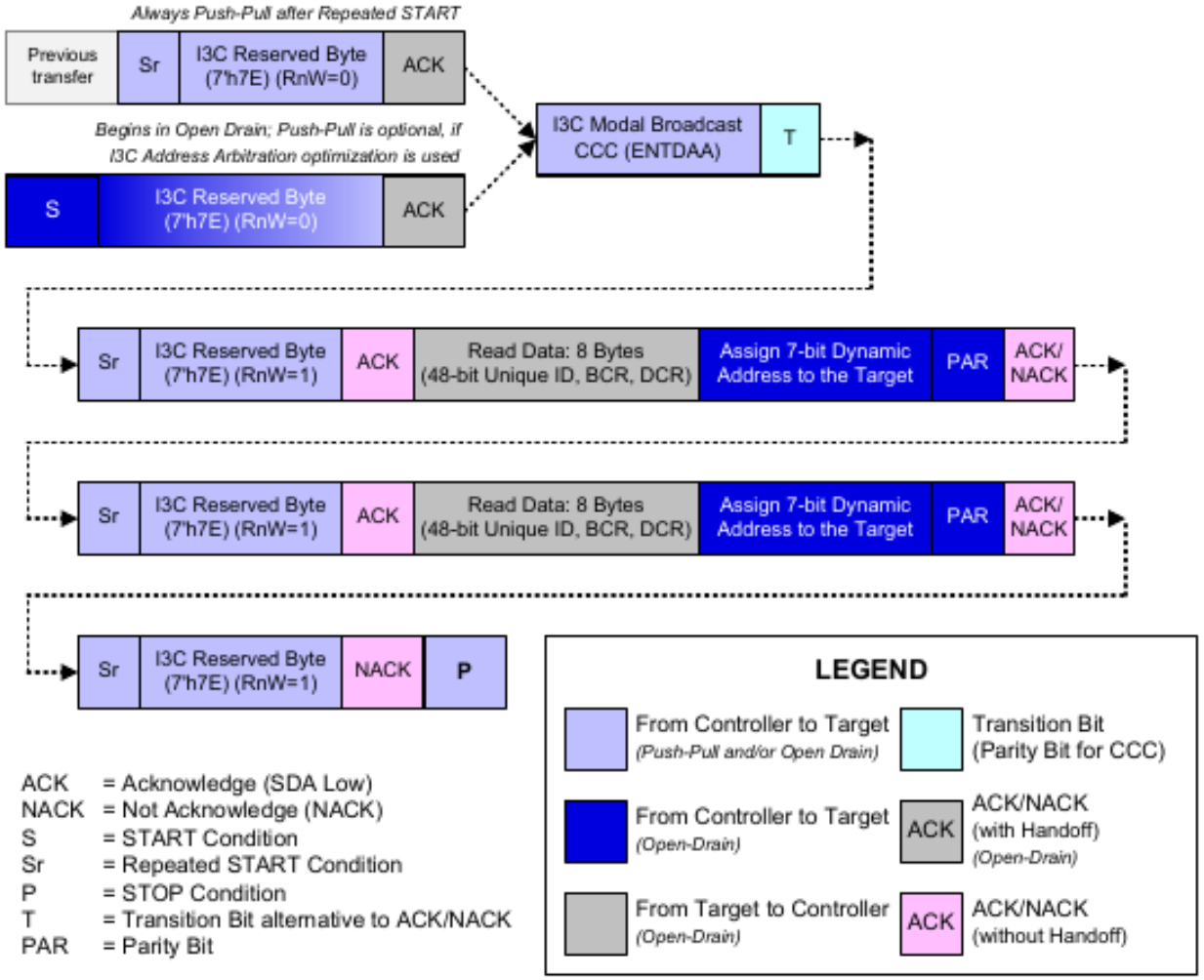



**Figure 25 Dynamic Address Assignment Transaction**

# Provisioned ID Collision Detection and Correction

- Collision happens when Provisioned ID and Characteristics Registers are identical.
- Probability is very less but still chance exists.
- If Number of Dynamic Addresses < Number of Dynamic devices, means that multiple devices have received same DA, thus collision occurred.
- Typically done by Primary Controller.
- Collision is corrected by RSTDAA and re-allocating DA.
- If collision cleared it's okay, else check until the number of devices is equal to DA's or until threshold limit is reached.

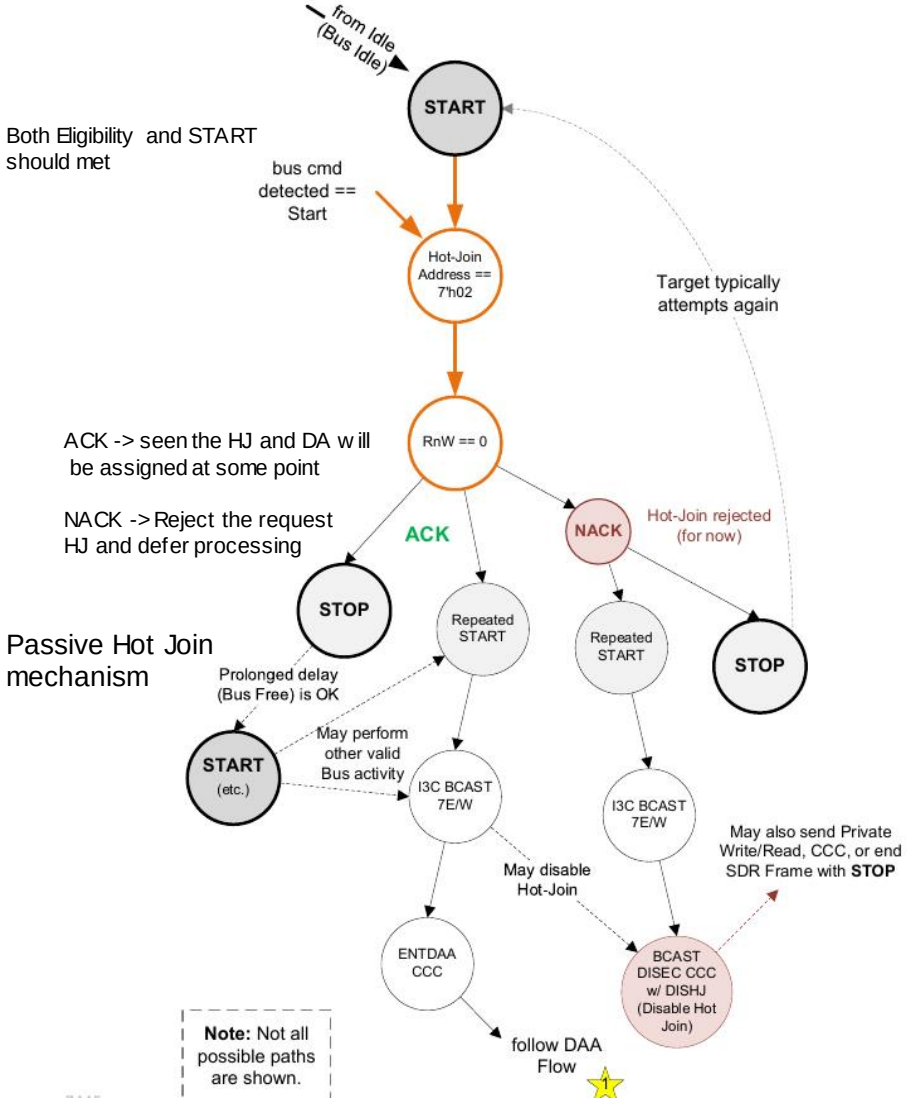
# Group Address Assignment Procedure

- Optional feature for I3C targets.
- Allows Controller to send a given I3C Message to all Target Devices in the Group at once.
- GA and DA exist simultaneously.
- Only Writes are possible but Reads are not.
- Target can be a member of any number of groups.
- No IBI shall be generated.
- Multilane is supported.

# Hot-Join Mechanism

- I3C Devices mounted on the same board, but de-powered until needed.
- I3C Devices mounted on a module/board that is physically inserted after the I3C Bus has already been configured.
- Two types of Hot-Join : Based on conditions of eligibility
  - Standard Hot-Join : Shall wait for bus idle condition for eligibility
  - Passive Hot-Join :
    - Wait at least Bus-free condition between Stop and Start -> I3C transaction in SDR mode for eligibility
    - Wait at least Bus-Available condition between Stop and Start -> IBI and CRR request for eligibility
- Shall not attempt with legacy I2C target or on a legacy I2C bus.

Passive Hot Join mechanism



# IN-BAND INTERRUPT

# In-Band Interrupt

- Priority level
- I3C Target Interrupt request
  - Mandatory Data Byte
  - Pending Read Notification
- I3C Secondary Controller Request to be Active Controller
- I3C Active Controller Initiating a transaction
  - It ensures that I3C devices can initiate IBI or CRR.
  - Initiate frames with START (not Sr) followed by broadcast address 7'h7E.

# Priority Level

- Controls the order in which In-Band Interrupt requests and Controller Role Requests from I3C Targets are processed.
- Lower the Dynamic Address higher the Priority i.e. address bits with value 0 prevail over bits with value 1.
- Thus during DAA, the Active Controller should assign lower Addresses to Targets for higher Priority In-Band Interrupt requests.

# Mandatory Data Byte (MDB)

- The Mandatory Data Byte of the IBI payload is the data that follows the Dynamic Address when a Device sends an IBI request.
- It is called MDB because the Target Device takes over the line when the Controller ACKs the IBI request and the Target continues to send data.
- The Mandatory Data Byte provides the Controller additional information about the event that has happened

# I3C Target Interrupt Request

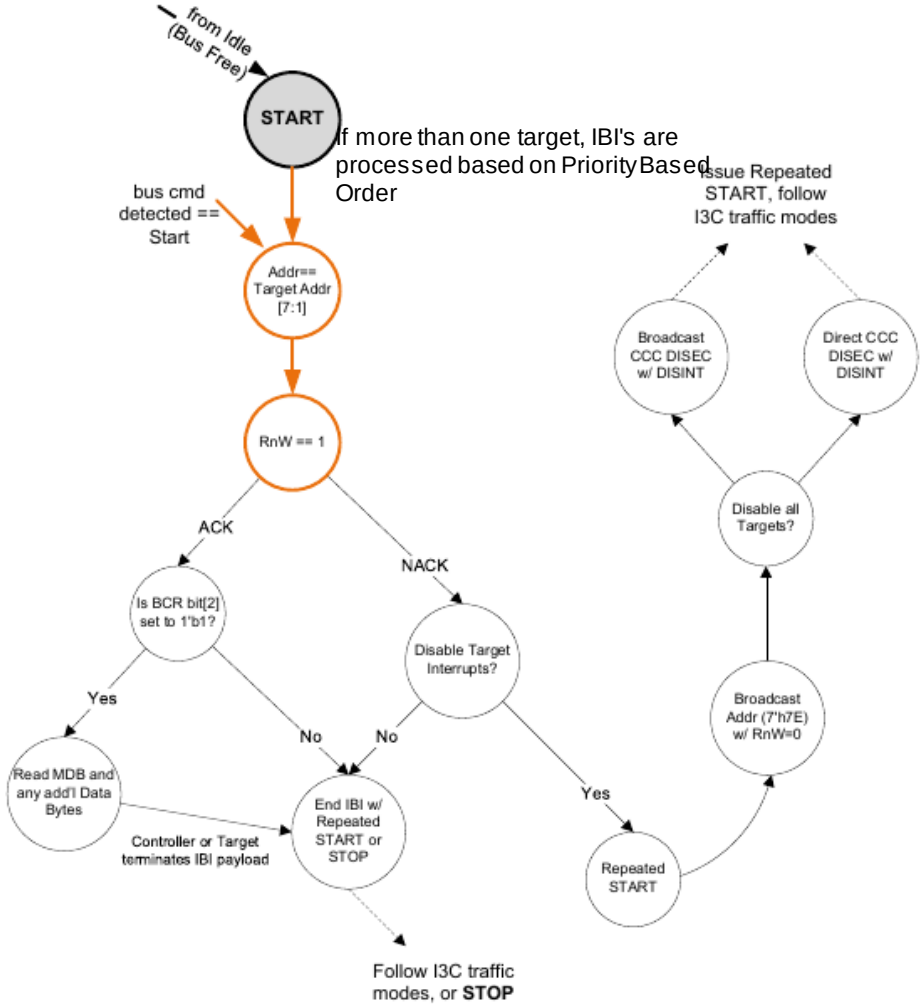
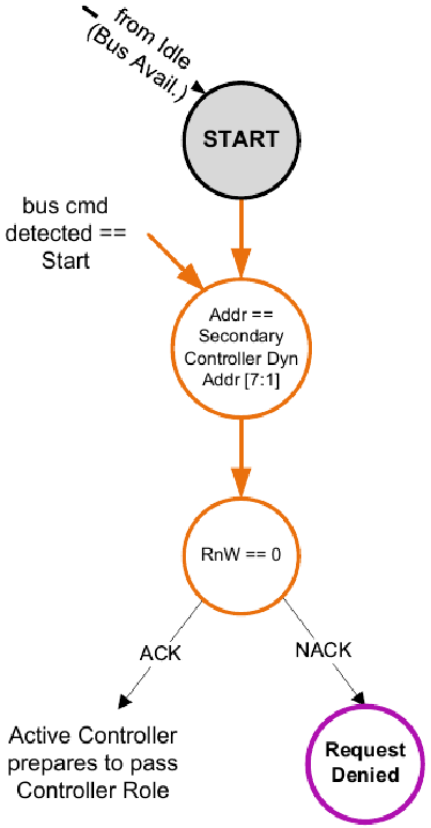



**Figure 169 In-Band Interrupt (Target Interrupt) Request FSM**

# Pending Read Notification

- Through IBI target may inform the controller that there is data to be read for specific purpose. Sent through MDB 3'b101.
- Only one active Pending Read Notification at any time.
- Controller may ACK or NACK.
  - If ACK's then controller must accept MDB and target will treat PRN as active.
  - If NACK's then target shall retry later.
- There may be multiple PRN' s from multiple targets.
- Controller can private read the target at any time. If time exceeds for a particular target it may re-transmit the IBI as a remainder.

# I3C Secondary Controller Requests to be Active Controller



# I3C BUS WITH MULTIPLE CONTROLLERS

# I3C Bus with Multiple Controllers

- Although only one I3C Controller-capable Device can act as Active Controller of the I3C Bus at any given time, the role of Active Controller may be passed between Controller-capable Devices.
- Handoff is a procedure through which
  - Multiple Controller-capable Devices can cooperatively pass the Controller Role back and forth.
  - One Controller-capable Device can direct the flow of passing the Controller Role among other Controller-capable Devices depending upon application and capabilities of I3C SC.
- Steps for passing Controller Role from Active Controller to Secondary Controller are
  - Prepare the Bus and the indicated Secondary Controller for Handoff.
  - Pass the Controller Role to the indicated Secondary Controller using the Controller to Controller Handoff Procedure.
  - Monitor the Bus to ensure that the indicated Secondary Controller asserts its Controller Role.

# Preparing for Handoff

- Preparing the I3C Target devices (SC's) for an optimal transition of the Controller Role.
- It is mandatory whether the handoff is
  - due to a Secondary Controller requesting the Controller Role via a Controller Role Request

    (or)
  - due to the Active Controller having simply chosen to pass the Controller Role to a particular Secondary Controller.

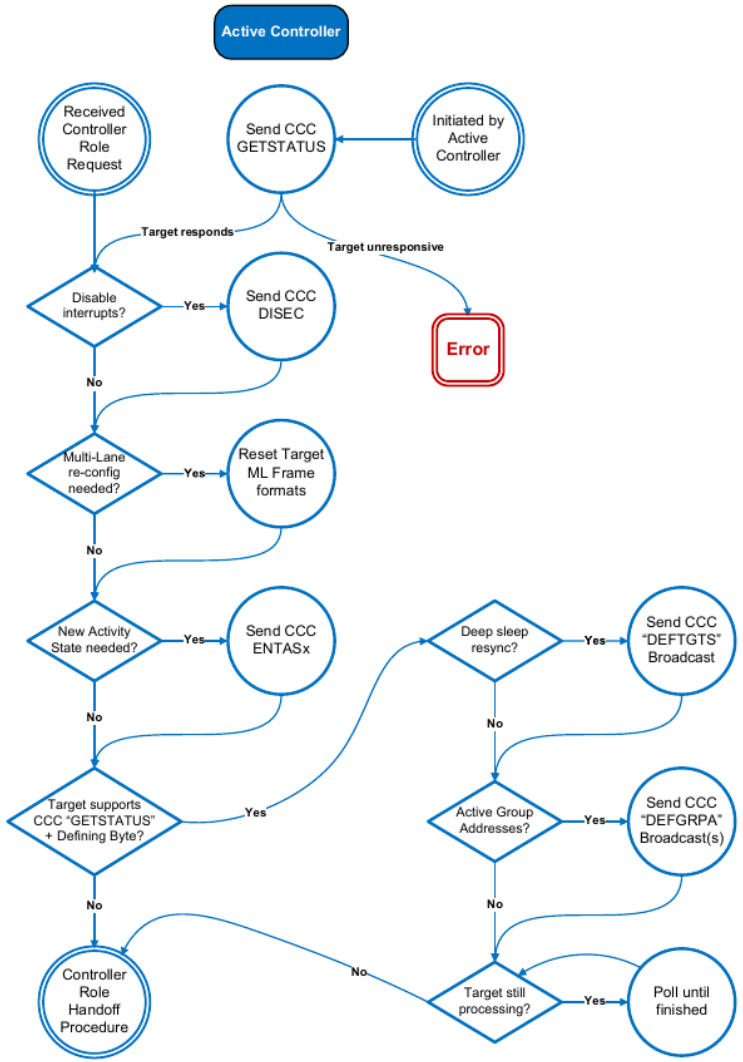



**Figure 27 Pre-Handoff Steps Flow**

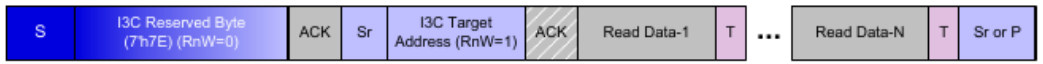
# Controller to Controller Handoff Procedure

- GETACCCR : Get Accept Controller Role
  - Used to verify whether SC is ready for Handoff.
- After Handoff preparation, Active controller shall issue GETACCCR followed by stop if Handoff is successful.
- After STOP followed by Bus available condition SC assumes the role of Active Controller and takes control of I3C bus.
- If the new active controller shall not pull SCL low before 100us after SDA is pulled low by former controller, then former active controller has to regain control by pulling SCL low.

## A.2 I3C Private Write and Read Transfers

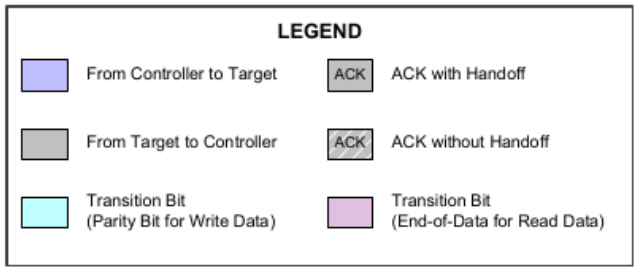

**I3C Private Write Transfer Initiated with START Condition**


**I3C Private Write Transfer Initiated with Repeated START Condition**




**I3C Private Read Transfer Initiated with START Condition**


**I3C Private Read Transfer Initiated with Repeated START Condition**




ACK = Acknowledge (SDA Low)
NACK = Not Acknowledge (NACK)
S = START Condition
Sr = Repeated START Condition
P = STOP Condition
T = Transition Bit alternative to ACK/NACK

**Figure 158 I3C Private Write and Read Transfers with 7'h7E Address**

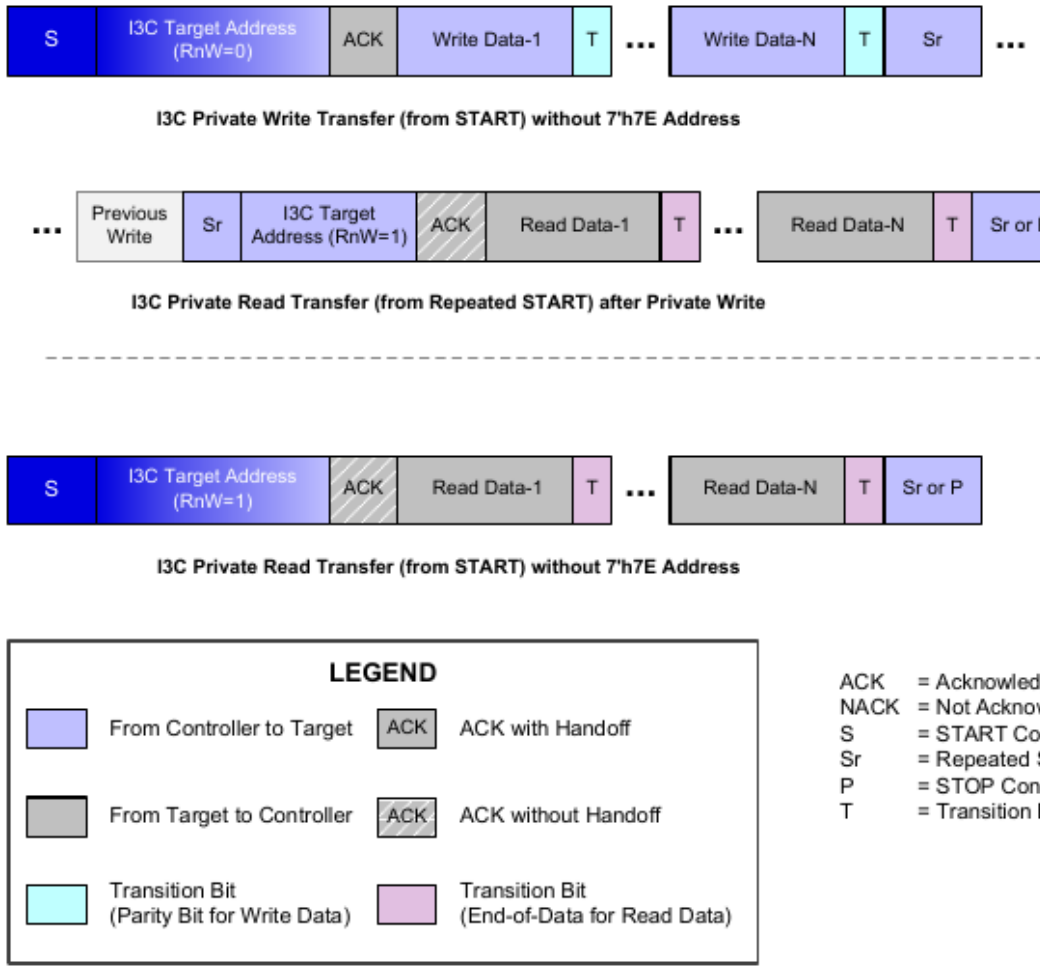


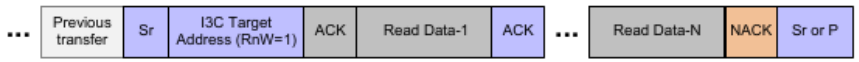

**Figure 159 I3C Private Write and Read Transfers without 7'h7E Address**

## A.3 Legacy I²C Write and Read Transfers on the I3C Bus

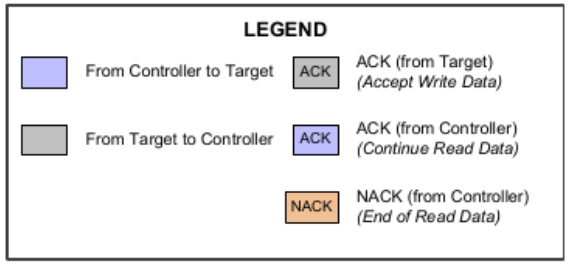

**Legacy I²C Write Transfer Initiated with START Condition**


**Legacy I²C Write Transfer Initiated with Repeated START Condition**


**Legacy I²C Read Transfer Initiated with START Condition**




**Legacy I²C Read Transfer Initiated with Repeated START Condition**




ACK = Acknowledge (SDA Low)
NACK = Not Acknowledge (NACK)
S = START Condition
Sr = Repeated START Condition
P = STOP Condition

**Figure 160 Legacy I²C Write and Read Transfers in I3C Bus with 7'h7E Address**

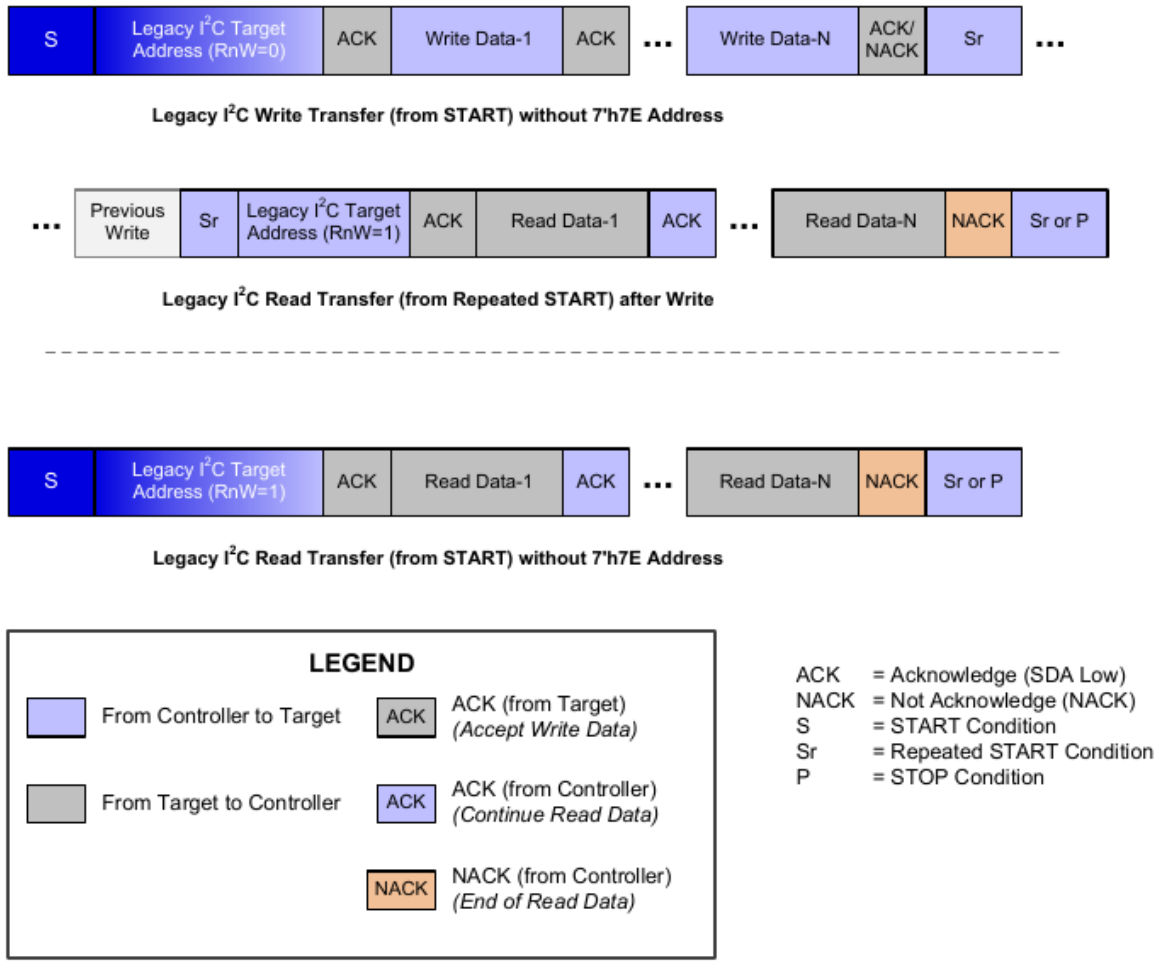
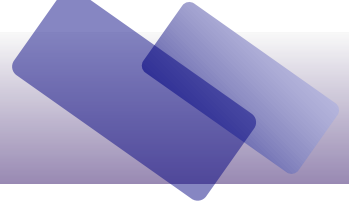



**Figure 161 Legacy I²C Write and Read Transfers in I3C Bus without 7'h7E Address**



THANK YOU