MIPI I3C TECHNOLOGY
AN INTRODUCTION TO MIPI I3C

EMMANUEL T. NANA
TECHNICAL MARKETING MANAGER

AMF-DES-T2686 | AUGUST 30, 2017

SECURE CONNECTIONS
FOR A SMARTER WORLD

AGENDA

1. Introduction
2. Basic MIPI I3C^{SM} signaling and protocol
3. Bus signals and address arbitration
4. High Data Rate (HDR) modes
5. Error detection and recovery
6. Device Identifier – Provisional-ID
7. Common Command Codes (CCC)
8. NXP’s free MIPI I3C^{SM} slave RTL
9. Summary
10. Public information

01.

Introduction

MIPI I3C = Next generation from I^{2}C

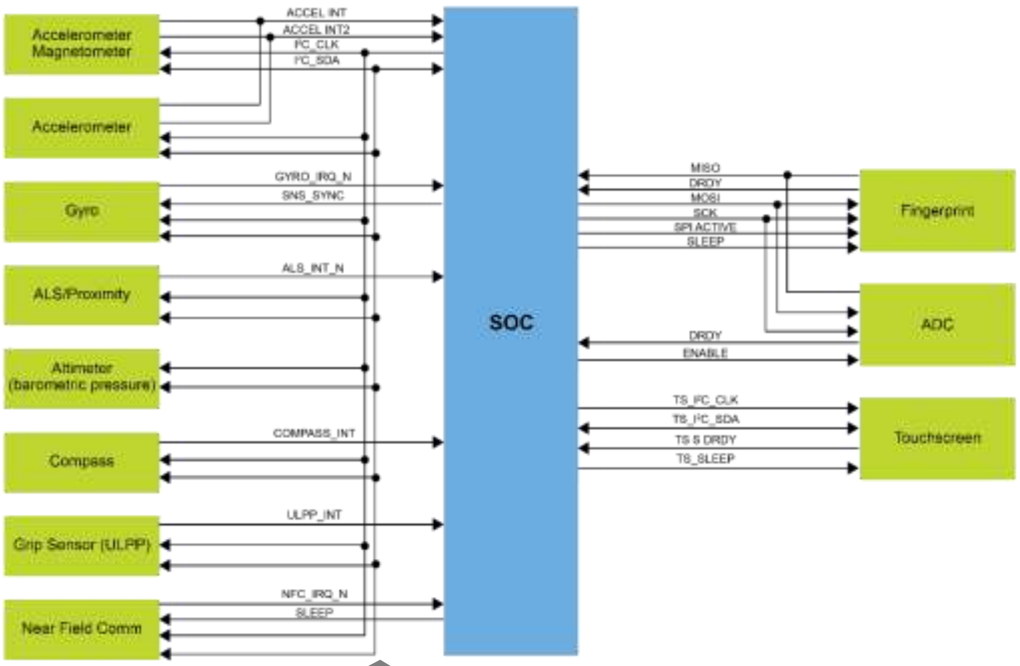
MIPI I3C
no logo
yet

- MIPI I3C is a follow on to I^{2}C
  - Has major improvements in use and power and performance
  - Optional alternative to SPI for mid-speed (equivalent to 30 Mbps)
- Background
  - NXP (Philips legacy) is I^{2}C leader and spec owner
  - I2C is used predominantly as control and communication interface with a focus in sensors (>90% according to 2013 MIPI Alliance survey)
  - MIPI Alliance Sensor Interface Workgroup initiated an upgrade of requirements in 2013
- Rationale for upgrade
  - In-band interrupt to reduce # of GPIO wires on SoC, as # of sensors increase on the mobile devices
  - I^{2}C speed has become limiting, as amount of data increases on the bus
  - Upgrade Constraints
    - Maintain backward compatibility, to enable a smooth transition from I2C to MIPI I3C and focus on *simple* implementation (recall I2C wide adoption is due to its seeming simplicity)
- MIPI I3C Spec Contributors
  - Primary Spec authoring: NXP (Paul Kimelman), Qualcomm, Intel, other contributors: Invensense, TI, STM, Synopsys, Cadence, Mentor, Sony, Knowles, Lattice

Sensor Interface Block Diagram

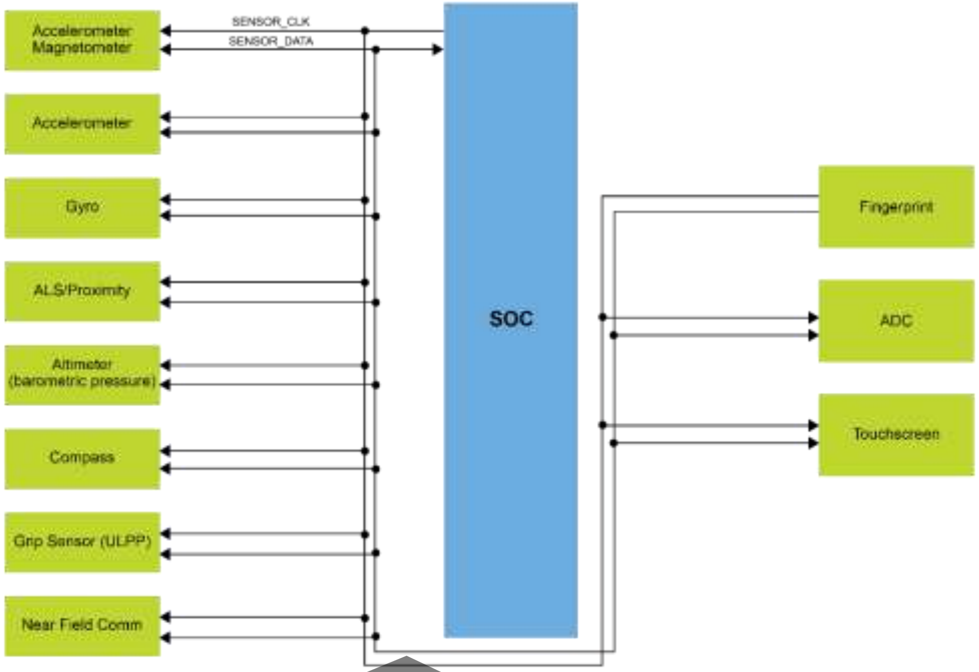
- In addition to higher data rate of the main interface,
  side-band channels such as dedicated interrupts, enable, and sleep signals might be needed
- Increased number of GPIOs is adding **system cost** in the form of added SoC package pins and PCB layer count

Current Scenario



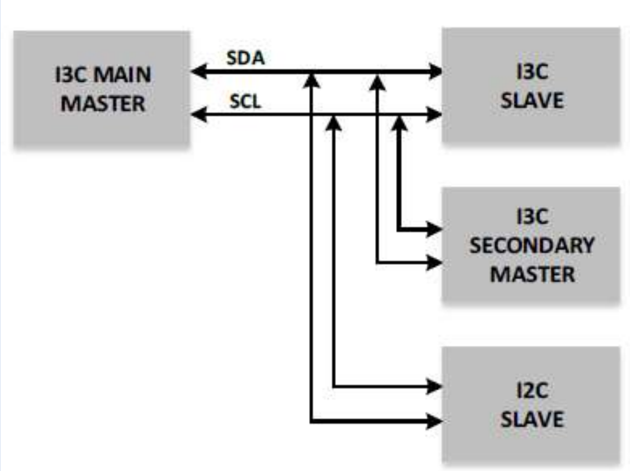
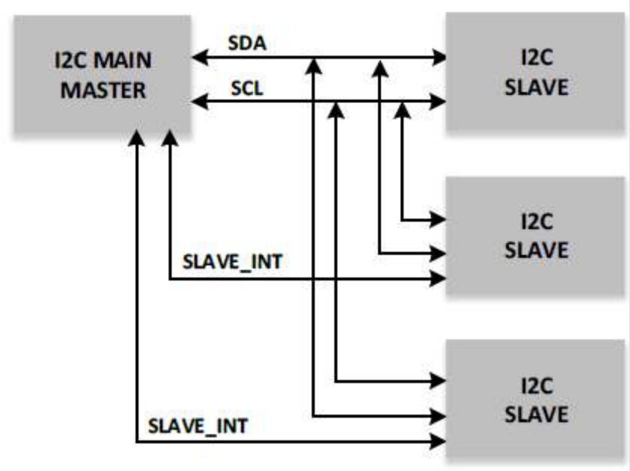
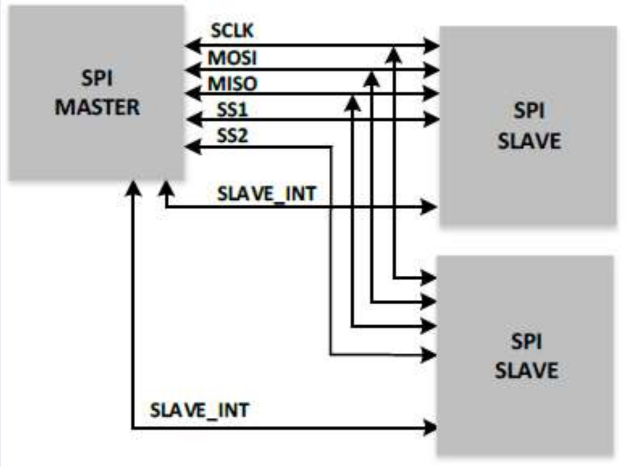
I^{2}C and SPI devices with side band channels EN, INT, etc

Desired Scenario



MIPI I3C with in-band interrupt, Common Command Codes for device control

Sensor Interface Block Diagram for MIPI I3C vs. I^2C & SPI

Parameter	MIPI I3C	I^2C	SPI
Overview	 I3C MAIN MASTER SDA SCL I3C SLAVE I3C SECONDARY MASTER I2C SLAVE	 I2C MAIN MASTER SDA SCL I2C SLAVE SLAVE_INT I2C SLAVE SLAVE_INT I2C SLAVE	 SPI MASTER SCLK MOSI MISO SS1 SS2 SLAVE_INT SPI SLAVE SPI SLAVE SLAVE_INT
# of lines	2-wire	2-wire (plus separate wires for each required interrupt signal)	4-wire (plus separate wires for each required interrupt signal)
Effective Data Bitrate	33.3 Mbps max at 12.5 MHz (Typ.:10.6 Mbps at 12 MHz SDR)	3 Mbps max at 3.4 MHz (Hs) 0.8 Mbps max at 1 MHz (Fm+) 0.35 Mbps max at 400 KHz (Fm)	Approx. 60 Mbps max at 60 MHz for conventional implementations (Typically: 10 Mbps at 10 MHz

From MIPI I3C White paper: http://resources.mipi.org/MIPI I3C-sensor-whitepaper-from-mipi-alliance

MIPI I3C versus I^{2}C at-a-glance

	I^{2}C	MIPI I3C
Clock Speed & Data Rate	Fast mode: 400kb/s Fast Mode+: 1Mb/s High speed: 3.4Mb/s Actual Data: computed 8/9th – 1 byte	SDR: up to 12.5Mbps raw rate (Actual Data Rate: 8/9th – per 1 byte) HDR-DDR: Actual Data Rate 20Mbps – 1 word HDR-TSP: Actual Data Rate to ~30Mbps – 1 word

# wires	2 – multi-drop (OpenDrain IF) 	
	SCL: clock – from Master(s), Slaves stretch
	SDA: data – bidirectional (OpenDrain)
		2 – multi-drop (SCL is push-pull, SDA OpenDrain and push-pull)
		SCL = clock (except for HDR-TSP) - from current Master only
		SDA = data – bidirectional (OpenDrain and push-pull)

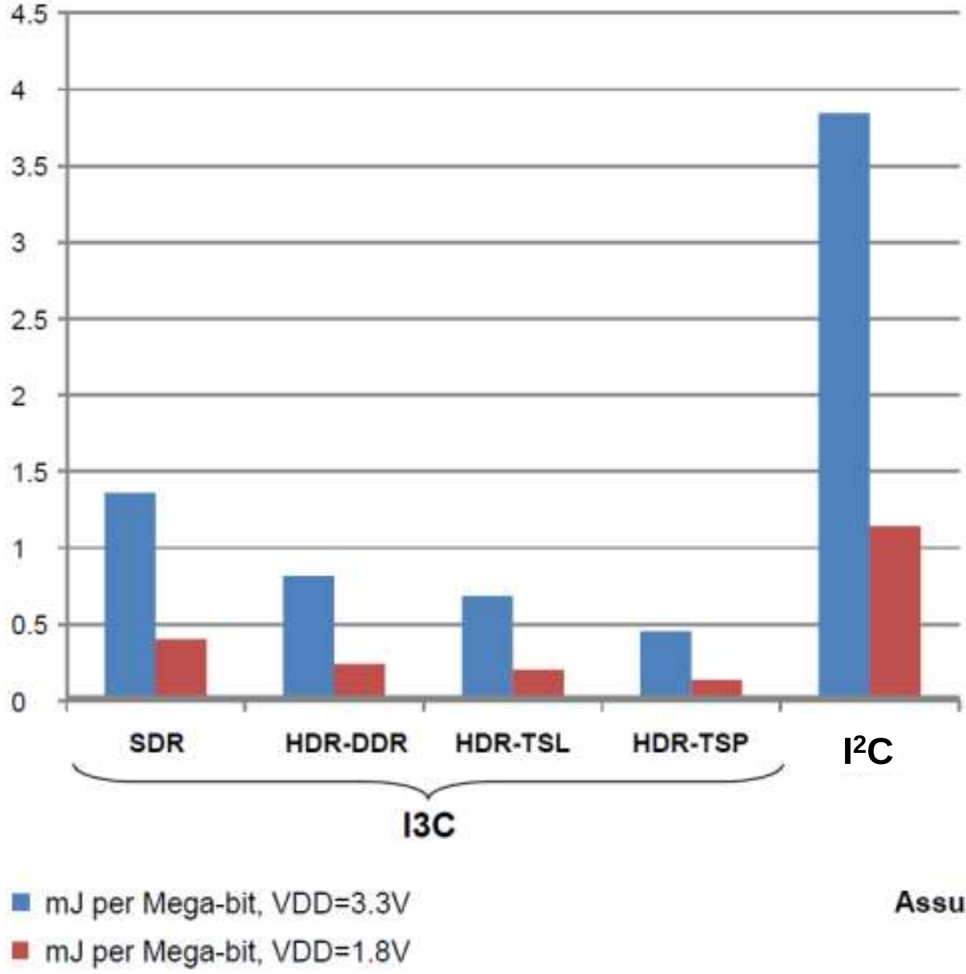
Power	High due to open-drain SCL , SDA	Lower due to SCL being push-pull only and SDA working in push-pull most of the time
Slave Read termination	Master has to end Read (so has to know length in advance)	Slave ends Read, but Master may terminate early
In-Band Interrupts	None – use a separate wire/pin per slave	Integrated, prioritized, and may include a byte (or more) of context
Hot-Plug	None – proprietary systems only	Built-in. Same mechanism a in-band-interrupt

MIPI I3C versus I²C at-a-glance

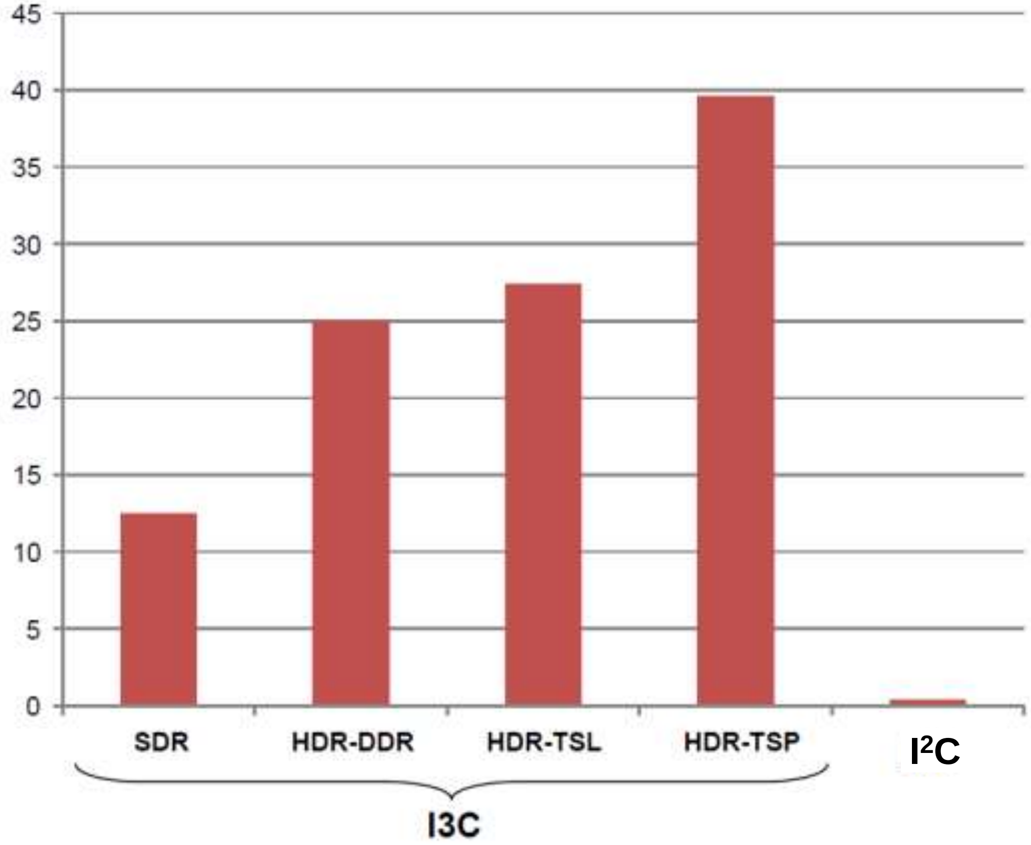
	I²C	MIPI I3C
Error detection	No protocol inherent error detection	Master and slave side error detection features
Time stamping	Has to be done by master once separate INT signal is triggered	Is an essential part of the MIPI I3C spec – no dedicated INT signal required.
Built-in Commands	None. Proprietary messages only	Built-in for control, capabilities discovery, bus management, etc. Expandable: e.g. Time Control, IO Expander use
Master / Slave	Master-Slave, Multi-master optional	Master-Slave; Master handoff (old Master->Slave)
IO pads	I²C special pads (e.g. 50ns spike filter)	Standard pads 4 mA drive, no spike filter
Slave address	Static	Dynamically assigned during initialization. Slaves may have static address at start
Clocking	Slaves normally use inbound clock	Slaves use inbound clock (allows slow/no internal clock)
Complexity	Low for Slaves. Higher for masters, especially around multi-master	Slaves as small as 2 K gates Masters as small as 2.5 K gates State machine or processor implementations

Advantages in energy and data rate

Energy Consumption
milliJoules per Megabit for I3C Data Modes (100pF)
vs I^2C (100pF, 3.54KOhm)



Raw Bitrate
Mbps for I3C Data Modes (@12.5MHz)
vs I^2C (@400KHz)



Assumptions: 1) All symbols in each mode have equal probability for use.
2) Energy consumption is the energy delivered by pull-up devices to the bus (which includes drivers and resistors).

02.

Basic MIPI I3C signaling and protocol

So, what does an MIPI I3C message look like?

MIPI I3C SDR looks almost the same as I^{2}C:

- E.g. Write data

	1 bit	8 bits	1 bit	8 bits	1 bit	…	1 bit
MIPI I3C	S or Sr	Addr+W	ACK/ NACK	1 Byte data	T bit = Parity	More data	Sr or P
I2C					ACK/ NACK		

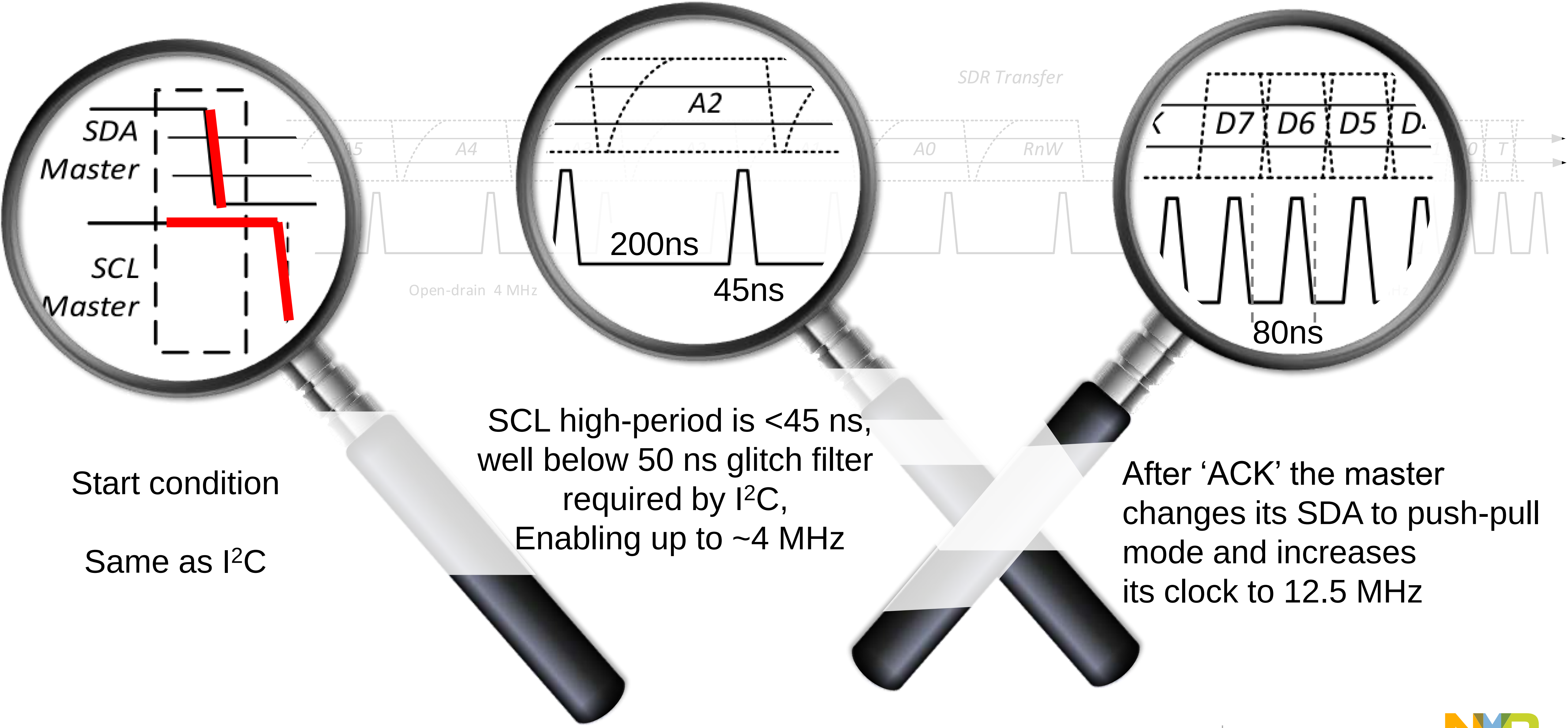
- E.g. Read data (typical approach):

	1 bit	8 bits	1 bit	8 bits	1 bit	1 bit	8 bit	1 bit	8 bit	1 bit	…	1 bit
MIPI I3C	S or Sr	Addr+ W	ACK / NACK	1 Byte data	T bit = Parity	Sr	Addr + R	ACK / NACK	1 Byte from Slave	T bit = ‘1 then Z’ to continue ‘0’ Slave ends transmission	More data	Sr or P Master ends read
I2C					ACK/ NACK					ACK/ NACK **Slave can’t abort read**		Master ends read

03.

Bus signals and address arbitration

MIPI I3C Bus signal in SDR mode after dynamic address assignment



Start condition

Same as I²C

SCL high-period is <45 ns, well below 50 ns glitch filter required by I²C, Enabling up to ~4 MHz

After ‘ACK’ the master changes its SDA to push-pull mode and increases its clock to 12.5 MHz

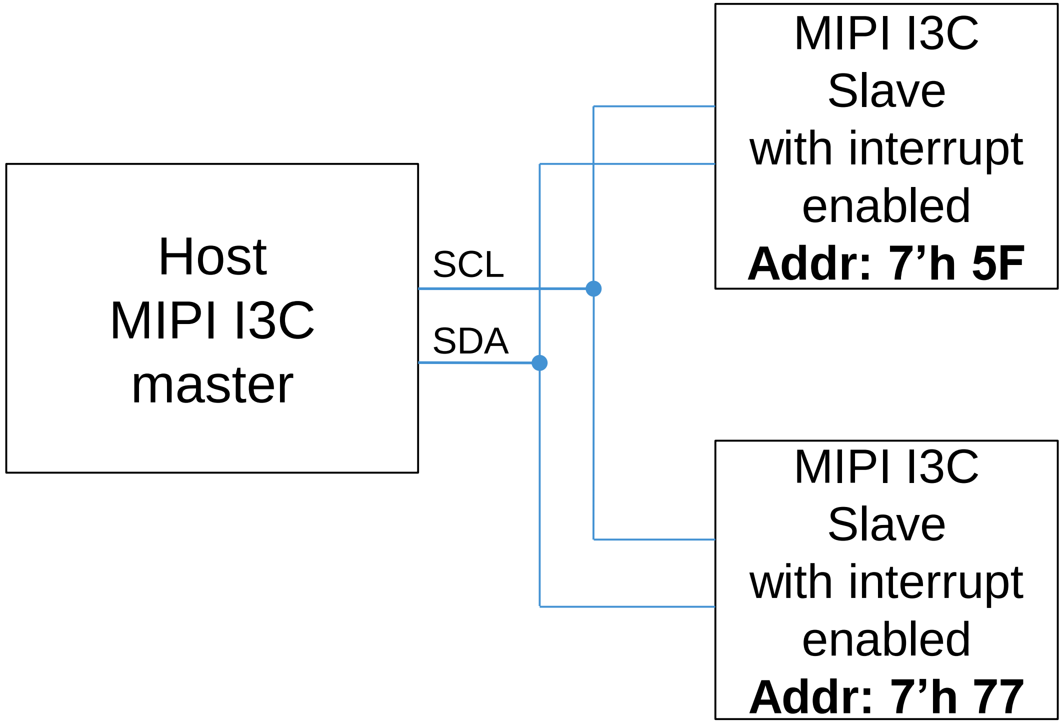
Why is address arbitration important in MIPI I3C

Address arbitration is used for multiple function in the MIPI I3C specification:

- In-Band Interrupt
  - slaves can trigger the master by pulling SDA low during a quiet period and the master will starting its SCL (start condition)
- Hot-Join
- Bus initialization if not all slave addresses are known

Address Arbitration

- System setup
  - MIPI I3C only system
  - 2 slaves with In-Band Interrupt enabled
  - BOTH slaves trigger an interrupt at the same time



Host MIPI I3C master

SCL

SDA

MIPI I3C Slave with interrupt enabled **Addr: 7’h 5F**

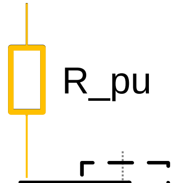
MIPI I3C Slave with interrupt enabled **Addr: 7’h 77**

A6	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**
1	**0**	**1**	**1**	**1**	**1**	**1**

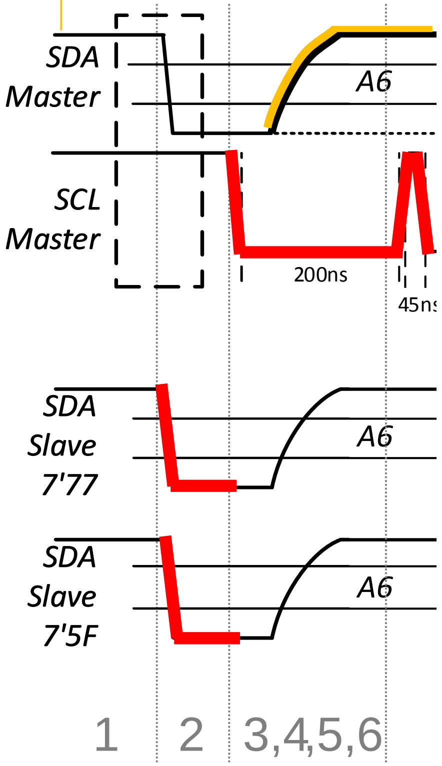
A6	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**
1	**1**	**1**	**0**	**1**	**1**	**1**

Address Arbitration

- Example: Interrupt triggered by slave in a system with 2 IBI capable slaves



A6	A5	A4	A3	A2	A1	A0
1	-	-	-	-	-	-



1. Master is idle with SCL stopped and SDA being pulled high by resistor
2. BOTH Slaves trigger an interrupt by pulling SDA low
3. Master starts SCL, pulling it low
4. Slave releases SDA
5. SDA is pulled high by R_pu
6. SCL pulse to latch address bit A6

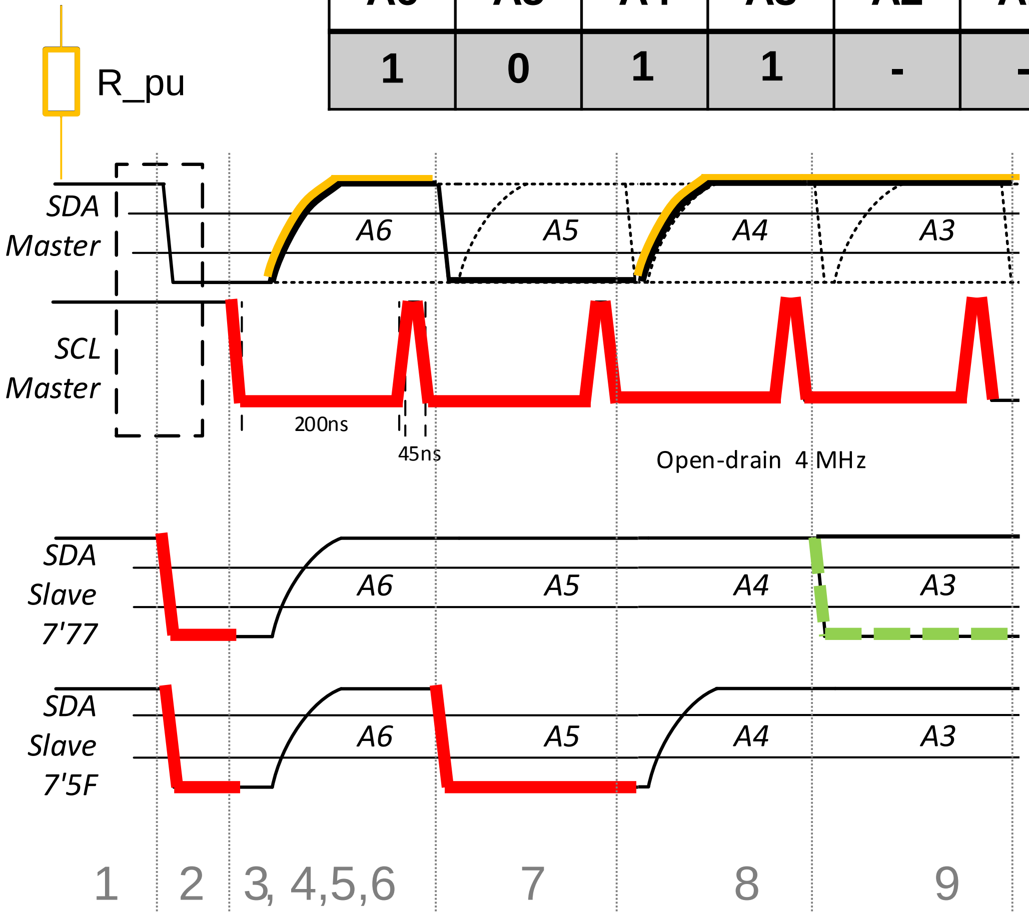
Actively driving device

signal on bus

Address Arbitration

- Example: Interrupt triggered by slave in a system with 2 IBI capable slaves

A6	A5	A4	A3	A2	A1	A0
1	0	1	1	-	-	-



1. Master is idle with SCL stopped and SDA being pulled high by resistor
2. BOTH Slaves trigger an interrupt by pulling SDA low
3. Master starts SCL, pulling it low
4. Slave releases SDA
5. SDA is pulled high by R_pu
6. SCL pulse to latch address bit A6
7. Slave 7'5F pulls A5 low. Latched with next SCL pulse – slave 7'77 keeps listening
8. Slave 7'5F releases SDA so A4 is '1'
9. Slave **7'77** does NOT communicate since A5 deviates from it's address so it 'lost' the arbitration and abstains from communication until the Start condition

🟥 Actively driving device

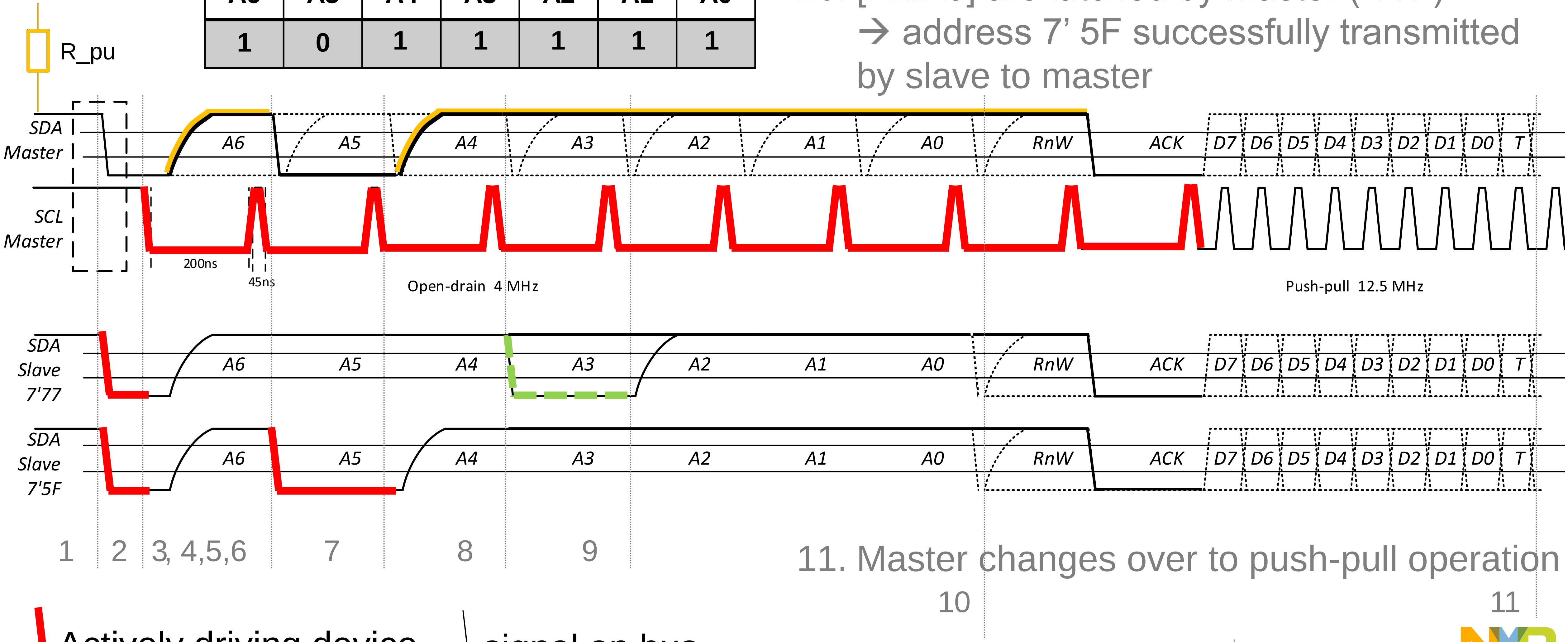
| signal on bus

Address Arbitration

- Example: Interrupt triggered by slave in a system with 2 IBI capable slaves

A6	A5	A4	A3	A2	A1	A0
1	0	1	1	1	1	1

10. [A2:A0] are latched by master (‘111’)
→ address 7’ 5F successfully transmitted by slave to master



11. Master changes over to push-pull operation

Actively driving device

signal on bus

In-band interrupts

In-Band interrupt allows Slaves to notify the master

- Can be used as an equivalent function compared to a separate GPIO
  - additionally, the IBI data frame can also be directly data bearing
  - and an IBI is prioritized. The lowest dynamic address slave will gain highest priority during arbitration
- Interrupts can be started even when Master is not active on the bus
  - No free running clock required (lower power)
- Time-stamping option to allow resolution of time of initial event
  - 2 ways to do: both relate to when actual IBI gets through to Master

Open Drain			Hand Off	Push-Pull	Drive High or Low, and then High-Z	Optional (push-pull)	Push-Pull
S	Slave_addr_as_IBI/R	**Master_ACK**	SCL High	Slave_byte (‘mandatory byte’)	T	More bytes	**Sr**

Note: How to increase the speed of the arbitration process

When a master has determined the slave’s dynamic address triggering the in-band interrupt, it can switch from open drain mode to push-pull mode for the next clock cycle as no other slave shall temper with SDA.

Therefore, dynamic addresses differentiated through their MSBs rather than their LSBs can be used. E.g. have all slave addresses start with ‘0’, so broadcasting 7h7E can detect any arbitration early on.

A hot-join / hot-plug device will announce itself by issuing 7’b 0000_010.

04.

High Data Rate (HDR) modes

HDR modes for higher throughput

- High data rate (HDR) modes are optionally available
  - No faster clock, but more bits for same frequency
  - Optional to support for Master and Slave
    - Incapable slaves know how to ignore, so others may use safely
- May have an HDR-DDR format
  - About 2x the data rate of SDR (so about 20 Mbps net using 12.5 MHz SCL)
  - Also includes CRC
  - Uses same SCL clocking, so small adder to Slave logic
- May use HDR-TSP (Ternary Symbols)
  - Results in up to 3x the data rate (so about 30 Mbps at 12.5 MHz) by using symbols on SCL,SDA rather than separate clock and data

05.

Error detection and recovery

Error Detection and Recovery Methods

The MIPI I3C bus specification details error detection and recovery methods for an SDR slave, the SDR master and HDR mode(s).

The error detection and recovery methods specified are provided in order to avoid fatal conditions when errors occur.

A set of 6 mandated methods and 1 optional method are specified for MIPI I3C Slave Devices, and a separate set of required methods is specified for MIPI I3C Master Devices.

Side note:
Clock stretching by slaves is NOT permitted
(→ SCL is driven via push-pull by the master)

Device error types

Slave side errors

Error Type	Description
S0	Broadcast Address/W (= 7’h7E /W) or Dynamic Address/RW
S1	CCC Code
S2	Write Data
S3	Assigned Address during Dynamic Address Arbitration
S4	7’h7E /R after Sr during Dynamic Address Arbitration
S5	Transaction after detecting CCC
S6 (optional)	Monitoring Error

Master side errors

Error Type	Description
M0	Transaction after sending CCC
M1 (optional)	Monitoring Error
M2	No response to Broadcast Address (7’h7E)

06.

Device Identifier – Provisional-ID

Device Identifier - MIPI I3C slave addresses

Device Identifier

In order to support the Dynamic Address Assignment procedure, each MIPI I3C Device to be connected to an MIPI I3C Bus shall be uniquely identifiable in **one** of two ways, before starting the procedure.

1. *The Device may have a Static Address, in which case the Master may use that Static Address*
   For example, an Address similar to what I2C specifies
2. The Device shall **in all cases** have a 48-bit Provisional ID.
   The Master shall rely on this 48-bit Provisional ID, unless the Device has a Static Address used by the master.

The 48-bit Provisional ID is composed of three parts:

Bits [47:33]	Bit [32]	Bits [31:00]		
		[31:16] 16-bits	[15:12] 4-bits	[11:0] 12-bit
MIPI Manufacturer ID (Note: MSB is discarded)	Provisional ID Type Selector 1’b1: Random 1’b0: Fixed	Part ID: The meaning of this 16-bit field is left to the Device vendor to define	Instance ID: Value to identify the individual example: straps, fuses, non-volatile memory, or another appropriate method	This is left for definition with additional meaning. For example: deeper Device Characteristics, which could optionally include Device Characteristic Register values
	If Bit [32] = 1’b1: Random Value: **Bits [31:0]: 32-bit value randomly generated by the Device.**			

07.

Common Command Codes (CCC)

Command space (CCC – Common Command Codes)

- Built-in Commands (>40) in separate “space” to avoid collision with normal Master → Slave messages
  - Controls bus behavior, modes and states, low power state, enquiries, etc.
  - Has additional room for new built-in commands to be used by other groups
  - Some are required, some are optional
  - Some may be direct communication with a single slave or are broadcasts to all slaves (same commands might be available in either category)

09.

NXP’s free MIPI I3C slave RTL

NXP’s free MIPI I3C slave

- NXP offers a free license for companies that are and are not members of MIPI: http://www.nxp.com/webapp/software-center/library.jsp#/home/query/MIPI%20MIPI I3C%20Slave%20IP%20for%20MIPI/~filter~/popularity/0
  - Non-members must agree to a confidentiality clause from MIPI (a requirement of MIPI Alliance: http://mipi.org). This IP is provided with no warranty, as must be agreed to in the click-wrap license.

Full commercial license

- Support and a full commercial license is available from Silvaco at http://www.silvaco.com/products/IP/MIPI_I3C.html

10.

Summary

Summary

- MIPI I3C simplifies system design to a true two-wire interface
- Backward compatible to existing I^{2}C devices (Supports legacy I^{2}C messaging)
- Very small RTL footprint requiring 2 pins only
- Lower power consumption than I^{2}C
- Supports in-bound error checking and CRC
- Supports peer-to-peer slave communication
- Supports hot-plug capability
- Dynamic addressing while supporting Static Addressing for legacy I^{2}C devices
- Supports I^{2}C-like SDR messaging and optional HDR messaging (up to 30Mbps)
- Supports multi-master and multi-drop capabilities

11.

Public information

Publicly available information:

- White paper: http://resources.mipi.org/MIPI I3C-sensor-whitepaper-from-mipi-alliance

- Press release: https://mipi.org/content/mipi-alliance-releases-MIPI I3C-sensor-interface-specification

- Commercially available IP (e.g.): http://www.silvaco.com/news/pressreleases/2016_12_01_01.html

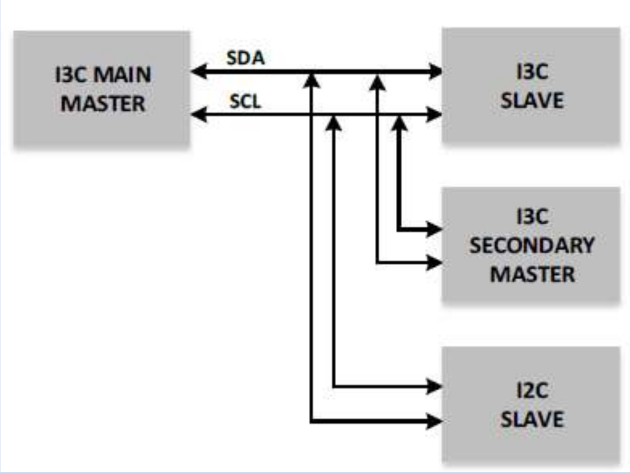
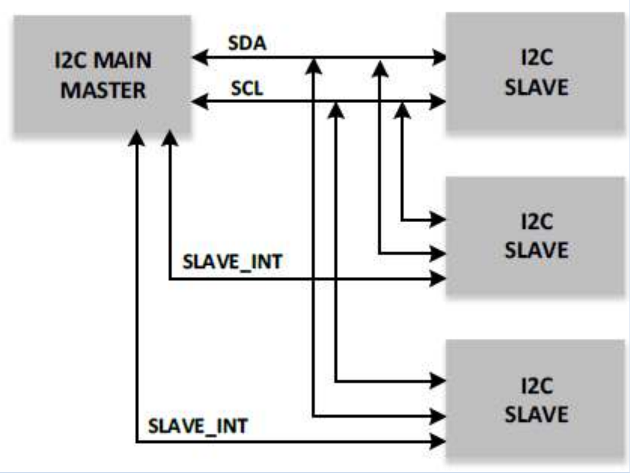
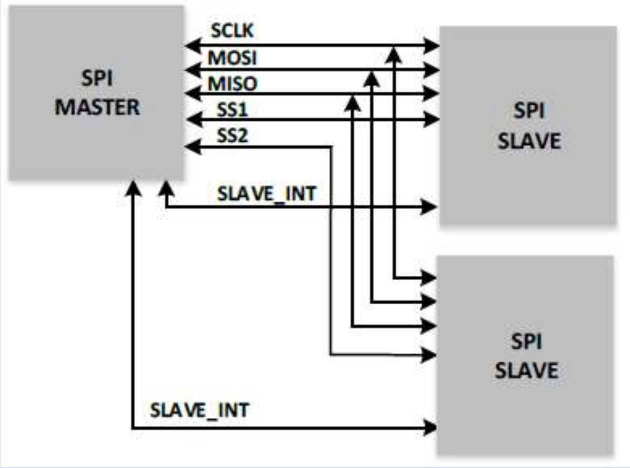
- Free license MIPI I3C (incl. I2C) slave IP from NXP: https://www.nxp.com/webapp/Download?colCode=MIPI I3C-NXP-FREE-LICENSE-SLAVE&amp;appType=license&amp;location=null&fsrch=1&sr=1&pageNum=1&Parent_nodeId=&Parent_pageType=&Parent_nodeId=&Parent_pageType

  or go to: www.nxp.com and search for ‘mipi i3c slave verilog’

12.

QUESTIONS?

From MIPI I3C white paper 1/3

Parameter	MIPI I3C	I2C	SPI
Overview	 I3C MAIN MASTER SDA SCL I3C SLAVE I3C SECONDARY MASTER I2C SLAVE	 I2C MAIN MASTER SDA SCL I2C SLAVE I2C SLAVE I2C SLAVE SLAVE_INT SLAVE_INT	 SPI MASTER SCLK MOSI MISO SS1 SS2 SLAVE_INT SPI SLAVE SPI SLAVE SLAVE_INT
# of lines	2-wire	2-wire (plus separate wires for each required interrupt signal)	4-wire (plus separate wires for eachrequired interrupt signal
Effective Data Bitrate	33.3 Mbps max at 12.5 MHz (Typ.:10.6 Mbps at 12MHz SDR)	3 Mbps max at 3.4 MHz (Hs) 0.8 Mbps max at 1 MHz (Fm+) 0.35 Mbps max at 400 KHz (Fm)	Approx. 60 Mbps max at 60 MHz for conventional implementations (Typically: 10 Mbps at 10 MHz

From MIPI I3C White paper: http://resources.mipi.org/MIPI I3C-sensor-whitepaper-from-mipi-alliance

From MIPI I3C white paper 2/3

Parameter	MIPI I3C	I^{2}C	SPI
Advantages	• Only two signal lines • Legacy I^{2}C devices co-exist on the same bus (with some limitations) • Flexible data transmission rates • Dynamic addressing and supports static addressing for legacy I^{2}C devices • I^{2}C-like data rate messaging (SDR) • Optional high data rate messaging modes (HDR) • Multi-drop capability and dynamic addressing avoids collisions • Multi-master capability • In-band Interrupt support • Hot-join support • A clear master ownership and handover mechanism is defined • In-band integrated commands (CCC) Support	• Only two signal lines • Flexible data transmission rates • Each device on the bus is independently addressable • Devices have a simple master/slave relationship • Simple implementation • Widely adopted in sensor applications and beyond • Supports multi-master and multidrop capability features	• Full duplex communication • Push-pull drivers • Good signal integrity and high speed below 20MHz (higher speed are challenging) • Higher throughput than I^{2}C and SMBus • Not limited to 8-bit words • Arbitrary choice of message size, content and purpose • Simple hardware interfacing • Lower power than I^{2}C • No arbitration or associated failure modes • Slaves use the master's clock • Slaves do not need a unique address • Not limited by a standard to any maximum clock speed (can vary between SPI devices)

From MIPI I3C White paper: http://resources.mipi.org/MIPI I3C-sensor-whitepaper-from-mipi-alliance

From MIPI I3C white paper 3/3

Parameter	MIPI I3C	I^{2}C	SPI
Disadvantages	• Only 7-bits are available for device addressing • Slower than SPI (i.e. 20Mbps) • New standard, adoption needs to be proven • Limited number of devices* on a bus to around a dozen devices	• Only 7-bits (or 10-bits) are available for static device addressing • Limited communication speed rates and many devices do not support the higher speeds • Slaves can hang the bus; will require system restart • Slower devices can delay the operation of faster speed devices • Uses more power than SPI • Limited number of devices on a bus to around a dozen devices • No clear master ownership and handover mechanism • Requires separate support signals for Interrupts	• Need more pins than I^{2}C/MIPI I3C • Need dedicated pin per slave for slave select (SS) • No in-band addressing • No slave hardware flow control • No hardware slave acknowledgment • Supports only one master device • No error-checking protocol is defined • No formal standard, validating conformance is not possible • SPI does not support hot swapping • Requires separate support signals for interrupts

From MIPI I3C White paper: http://resources.mipi.org/MIPI I3C-sensor-whitepaper-from-mipi-alliance

* This means physical devices. Total bus capacitance <= 50 pF.

SECURE CONNECTIONS
FOR A SMARTER WORLD