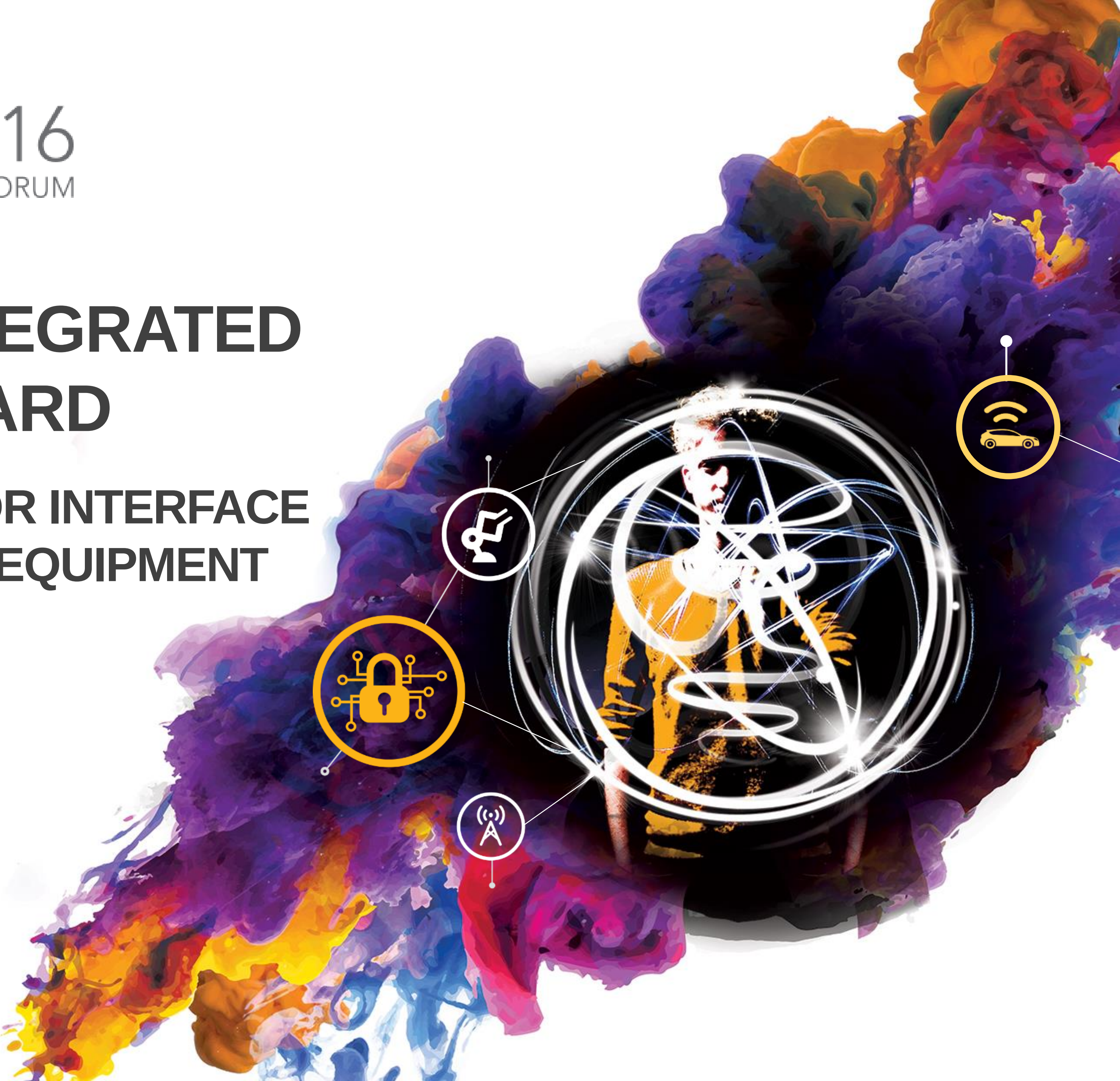
NXP | FTF 2016 TECHNOLOGY FORUM

IMPROVED INTER INTEGRATED CIRCUIT (I3C) STANDARD

NEW SERIAL BUS FOR SENSOR INTERFACE IN MOBILE AND ELECTRONIC EQUIPMENT

EMMANUEL T. NANA
PRODUCT APPLICATIONS ENGINEER
SESSION #: FTF-INS-N1913
WEDNESDAY, MAY 18, 2016 (5:45PM)



PUBLIC USE

ABSTRACT

Philips Semiconductor (now NXP Semiconductors) invented the Inter-Integrated Circuit (I^2C) as a multi-master, multi-slave serial bus in the early nineteen-eighties. The bus was intended to be a simple two-wire interface consisting of **S**erial **CL**ock (SCL) and **S**erial **DA**ta (SDA) lines. However, with the introduction of new peripheral devices, additional pins are most often needed for reset, interrupt, slave address, etc. To further simplify the bus back to a true two-wire interface, while maintaining the these common features and increasing the bus speed, MIPI Alliance is leading the industry in the definition and standardization of a new standard called the Improved Inter Integrated Circuit (I3C). This class will discuss the new features and benefits of the I3C bus and present distinct contrasts between the I^2C and I3C specifications.

AGENGA

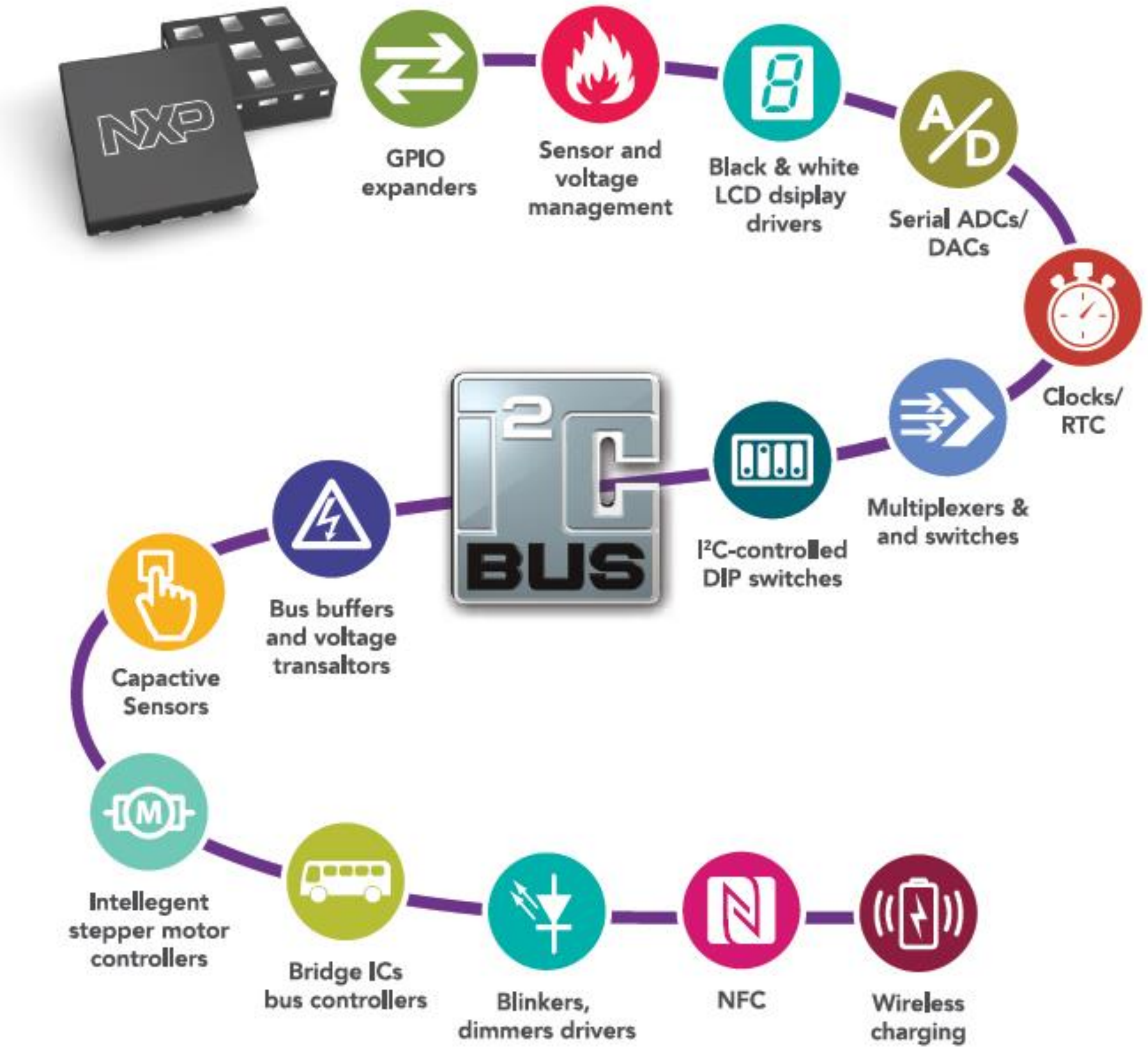
- Introduction to I^2C & Shortcomings
- Features of the New I3C Specification
- Status of Specification Development
- Summary

INTRODUCTION TO I^{2}C & SHORTCOMINGS

I^2C-Bus History and Quick Facts

- Originally released in 1982 as a simple and robust way to communicate with various ICs in television
- I^2C (Inter-Integrated Circuit), pronounced I-squared-C or I-two-C
- Two wires, bidirectional, half-duplex
- Standard clock speeds from 100kHz to 5MHz
  - Surprising throughput due to minimal overhead
- Major bus used for peripheral communication and control
- Additional pins required for interrupt (/INT), reset (/RST)
  - For a typical system using an App Processor with five I^2C interfaces, the number of I2C related pins on the SoC = 5*2 (SCL, SDA) + 10 (INT) = 20 pins!!!
- Patented and licensed by Philips. No fees required since 2006
  - Specification maintained by NXP – UM10204

The Universal Bus



GPIO expanders
Sensor and voltage management
Black & white LCD dsiplay drivers
Serial ADCs/ DACs
Clocks/ RTC
Multiplexers & and switches
I²C-controlled DIP switches
I²C BUS
Bus buffers and voltage transaltors
Capactive Sensors
Intellegent stepper motor controllers
Bridge ICs bus controllers
Blinkers, dimmers drivers
NFC
Wireless charging

System Migration

- **Early (2007) sensor usage,** merely 3-4, i.e. touch screen, proximity, and accelerometer.
- **Today,** smart-phones have 10-20 sensors and growing with applications like gaming, mobile fitness/sports, health and wellness, enhanced gesture recognition, motion compensation for uses like camera/video recording, and indoor navigation (location estimation based on last known GPS acquisition), etc.
- **In future,** more sensors and higher speed requirements… User Interface enhancements like augmented reality is increasing the frequency and data rate requirement of various sensors.

Mechanical/Motion

- Compass/Magnetometer
- Gyro
- Accelerometer
- Proximity
- Touch screen
- Grip
- Time of Flight (gestures)
- Audio/Ultrasonic (events)

Biometrics/Health

- Fingerprint
- Glucometer
- Heart rate
- Olfactory (e.g. breathalyzer)
- EKG
- GSR (galvanic skin response)

Environmental Sensing

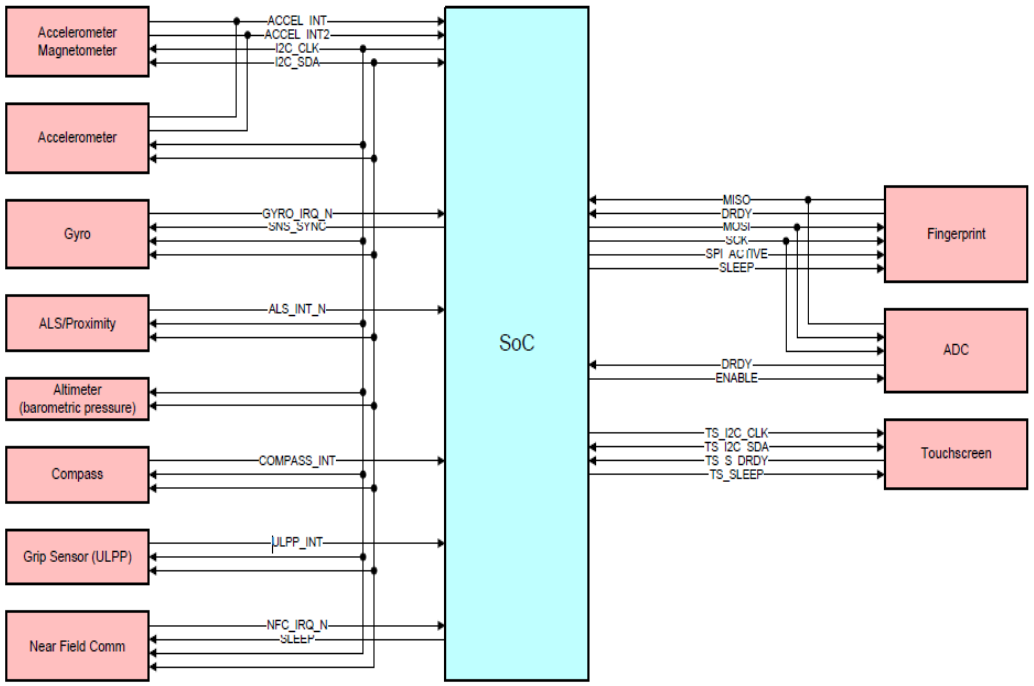
- Ambient light
- Barometric pressure/altimeter
- Temperature
- Carbon monoxide/pollutants
- Humidity

Other

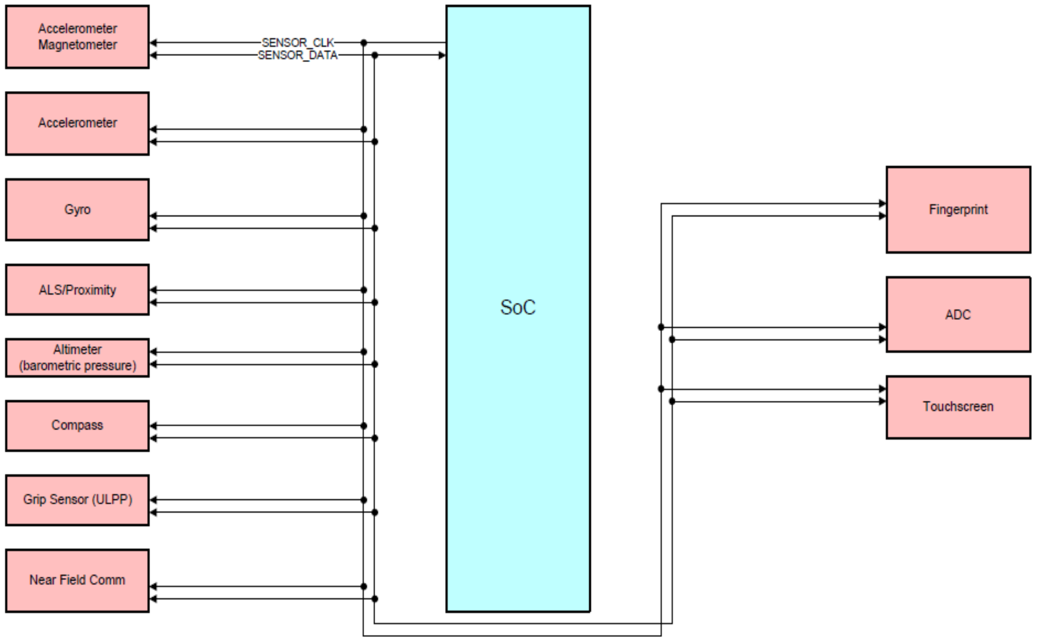
- NFC (Near Field Communication)
- Haptic feedback
- IR (smart TV remote)
- UV/RGB

Sensor Interface Block Diagram

Current Scenario



Desired Scenario



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

Sensors and Other Devices with & without Interrupts

No Interrupt Required

Ultrasonic sensor

Carbon Monoxide & Dioxide sensor

Humidity sensor

Barometric Pressure Sensor

Olfactory (Breath Analyzer) sensor

Galvanic Skin Response (GSR) sensor

Haptic Feedback

Remote Control IR port

Switches and RF Front End Control

Memory Control

LED Backlight Control

LED Flash Control

Camera Module Control

Time of Flight Sensor Control

Interrupt Required

Compass Sensor

Gyro Sensor

Accelerometer

Proximity Sensor

Touch Screen Sensor

Grip Sensor

Ambient light Sensor

UV/RGB Sensor

PMIC

Temperature Sensor

Fingerprint sensor

Heart Rate Sensor

EKG Sensor

Near Field Communication

Feature Wish List for New Interface

- Lower number of interface pins; Support in-band interrupt functionality without added pins
- Simplify with minimal die impact (i.e. <3x the die size of regular I^{2}C)
- Similar or lower power consumption as Regular I^{2}C; No continuous clock
- Support similar or higher data rate as HS I^{2}C
- Backwards compatible with the existing I^{2}C devices
- Support in-band power modes (Hibernate, Shutdown, Reset, Wakeup)
- Prefer regular I/O’s for this new interface for easier configurability
- Support in-bound error checking or CRC
- Support slave-slave communication
- Support free-running clock if required
- Support hot-plug capability

FEATURES OF THE NEW I3C SPECIFICATION

Next Generation from I^{2}C

- MIPI I3C is a follow on to I^{2}C
  - Has major improvements in use and power
  - Provides an alternative to SPI for mid-speed

- Two-wire multi-drop bus capable of 12.5MHz clock with up to 11 devices
  - Uses standard pads (versus I^{2}C special pads) with 4mA drive
  - Slave addresses are dynamically assigned – no static address
    - But, slaves **may** have a static address at start
  - Slaves may use inbound clock as the peripheral clock
    - So devices may have slow/inaccurate clocks internally
  - Slave normally ends Read, but Master may terminate
    - Unlike I^{2}C and SPI with the problems of Master having to “know” length

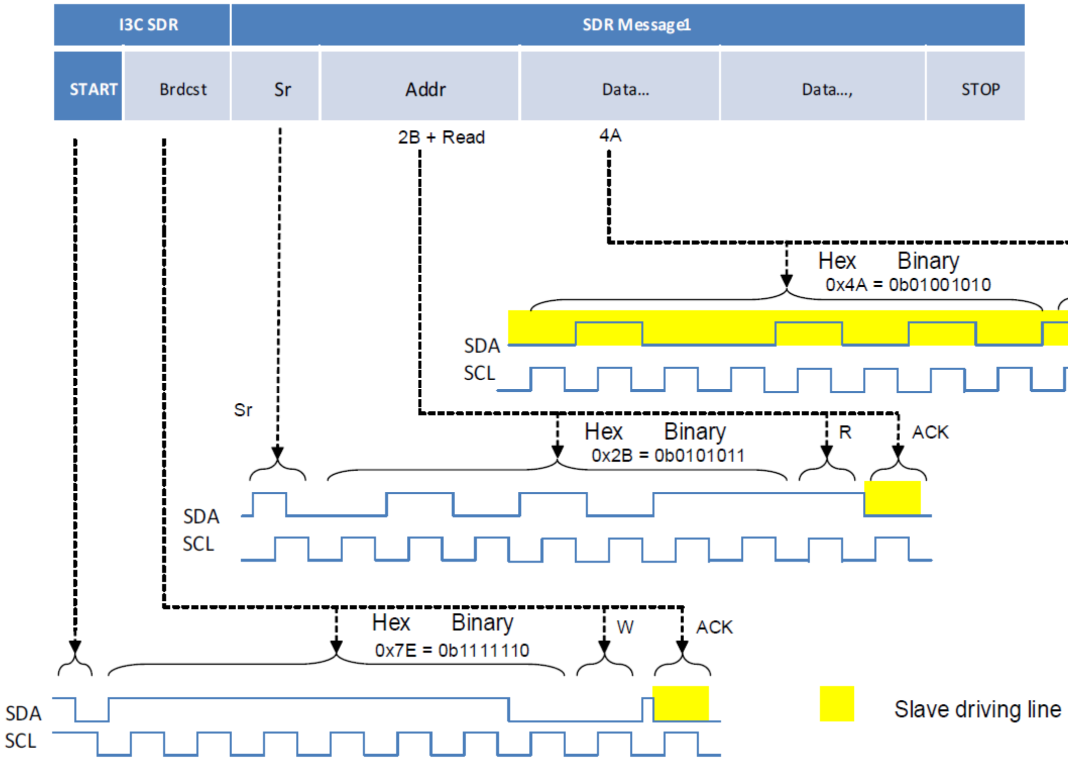
- Slaves as small as <2k gates, Masters as small as 2.5k gates
  - Allowing for fully state machine driven as well as using processor

Improved Inter Integrated Circuit (I3C) Features

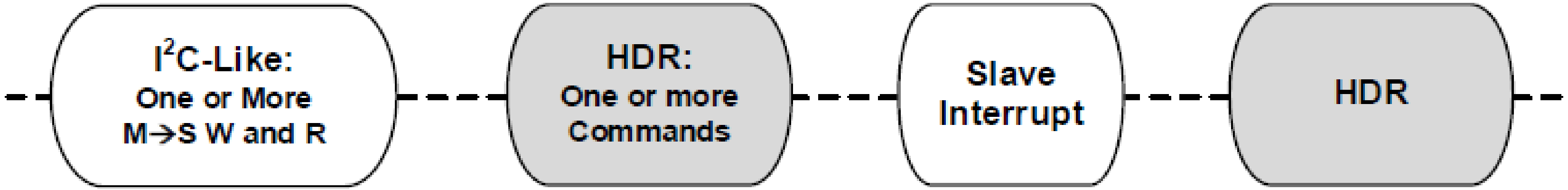
1. Backward compatible to regular I^2C
   - I3C in-band-interrupt master supports a bus with a mix-and-match of Regular I^2C slaves and I3C slaves
2. Single Master system with multiple slaves
3. No Sideband Signals - Lower number of wires
   - In-band interrupt, reset, Wakeup, Shutdown, Sleep and power management capabilities
4. Interrupt prioritization mechanism via allocation of interrupt vector at reset
5. Up to 30Mbps with multiple speed classes support
6. Support slave-slave communication with single clock source of Master
7. Support Free running clock (if required by any slave on bus such as display)
8. Support Hot-plug detection for the pluggable accessories
9. Reduced Power consumption than regular I^2C
   - Option to remove passive resistors on SCL and SDA
10. Minor firmware or hardware changes compared to regular I^2C Slave
11. Support a capability register to indicate the slave capability
12. Option to limit data payload to 32 bytes to contain interrupt response delay

Communication Using I3C Coding SDR

- Supports several communication formats, all sharing a two-wire interface
  - SDA is a bi-directional data pin
  - SCL can be either a clock pin or a data pin while in certain HDR modes
- Supports the mixing of various message types
  - I²C messages to legacy I²C slaves
  - I²C-like SDR messages, with clock speeds up to 12.5MHz
  - Broadcast and direct common command code (CCC) messages that allow the master to communicate to all or one of the slaves
  - HDR mode messages, which achieve higher data rates
  - Slave-initiated requests to the Master
  - Peer-to-peer messages between I3C devices



Slight dip on SDA indicates weak pull- up. Master may choose to terminate read and take over.



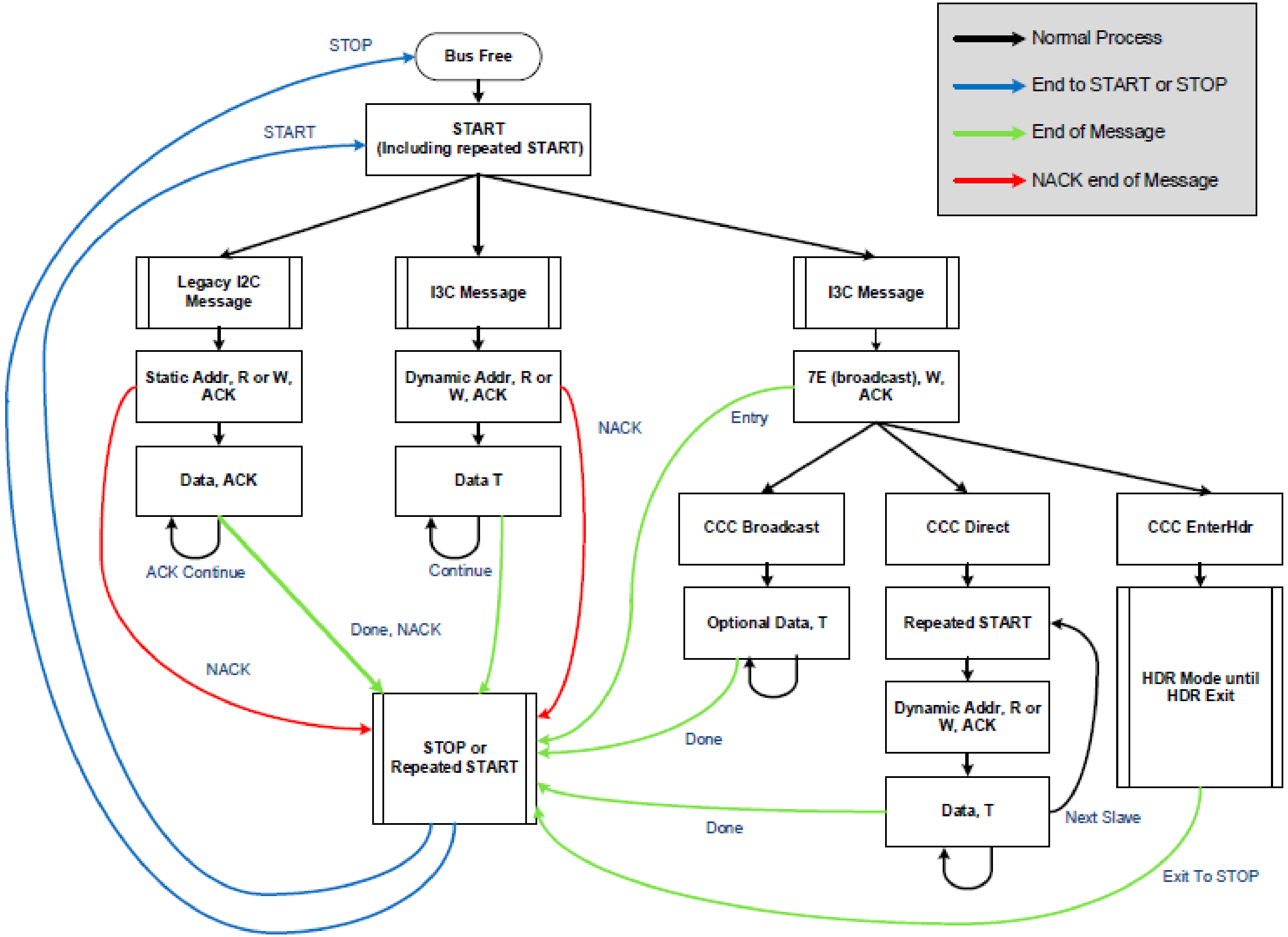
I²C-Like: One or More M→S W and R

HDR: One or more Commands

Slave Interrupt

HDR

Communication Using I3C Coding SDR



I3C Capability Registers

- System designer must know which devices are legacy and which implement the new specs
- Out of reset master reads the Capability Address to check what capabilities are supported.
- The register in each device (slave) indicates which version of the interface specification and which optional features the device supports:
- e.g. 0xF0 Capability register
  - Bits[3:0] – Version number
  - Bit [4] – Support Low Power Mode w/ push-pull output
  - Bit [5] - Support Slave to Slave Communication (CMD Mode)
  - Bit [6] – Indicate that this device is hot-pluggable
  - Bit [7] – Requires Free running clock
  - Bit [8] – In-band interrupt supported (An I2C device that doesn't need interrupt at all doesn't need to set this)
  - Bit [11:9] - Reserved
  - Bit [12] – In-band reset supported (CMD mode)
  - Bit[14:13] – Power Management modes supported (CMD mode)
  - …
- When master does a write the 1st byte after the address shall be interpreted as CMD by the slave supporting this feature
- Master shall not drive Command byte for slaves not supporting this feature
- Master will drive specific commands for in-band reset, entry/exit from power management states using these commands.
- Example Commands: NOP(normal), Force Reset, Enter/Exit Idle, Enter/Exit Sleep

I3C Terminology

- **Master:** I^{2}C Host Controller
- **Slave:** Regular I^{2}C devices on the bus
- **Master-Slave:** Master-Slave is terminology used for Slave that can act as a master for other sub-slaves. This slave maintains the address database of sub-slaves from which it can request the data. These are reserved registers written by the Host controller after the reset.

In-Band Interrupts (Allow Slaves to Notify Master)

- Can be both equivalent to a separate GPIO, but can also be directly data bearing
- Prioritized so that if multiple Slaves wish to interrupt at the same time, the order is resolved
  - Dynamic addresses used for this, so priority controlled by Master
- Interrupts can be started even when Master is not active on the bus, and yet no free running clock needed
- Time-stamping option to allow resolution of initial event vs. when interrupt gets through

Built-in Commands

- In separate space to not collide with normal Master → Slave message
- Controls bus behavior, modes and states, low power states, enquiries, etc.
- Has additional room for new built-in commands to be used by other groups

Mastering, Late Start / Insertion, Mixed Buses

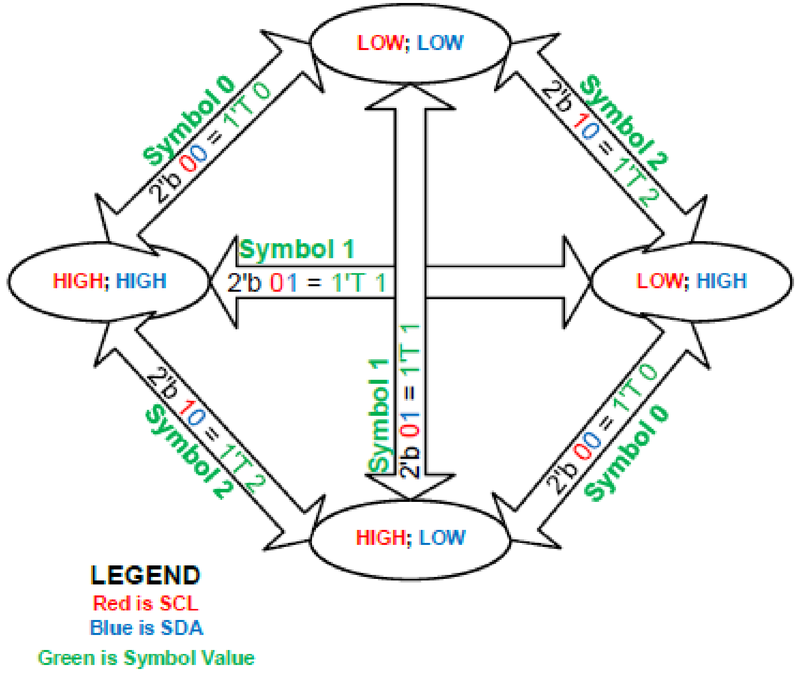
- Organized forms of multi-master:
  - Slave can request Master to allow it to message another Slave, yet not needing to generate its own clock – called Peer-to-Peer
  - Secondary Masters which can use clean handoffs between each Master

- Hot-join onto bus allows devices to come on-line later than initial bus bring up
  - May be due to late wake up (power-up) or physical insertion
  - Provides clean method for notification

- Mixed I^{2}C and I3C capable
  - I3C has support for certain legacy I^{2}C devices on the bus
    - While maintaining high speeds
  - I3C Slave devices capable of operating on I^{2}C buses as I^{2}C slaves
  - Also support for bridging (to I^{2}C, SPI, UART, etc.)

High Data Rate (HDR) Modes for Higher Throughput

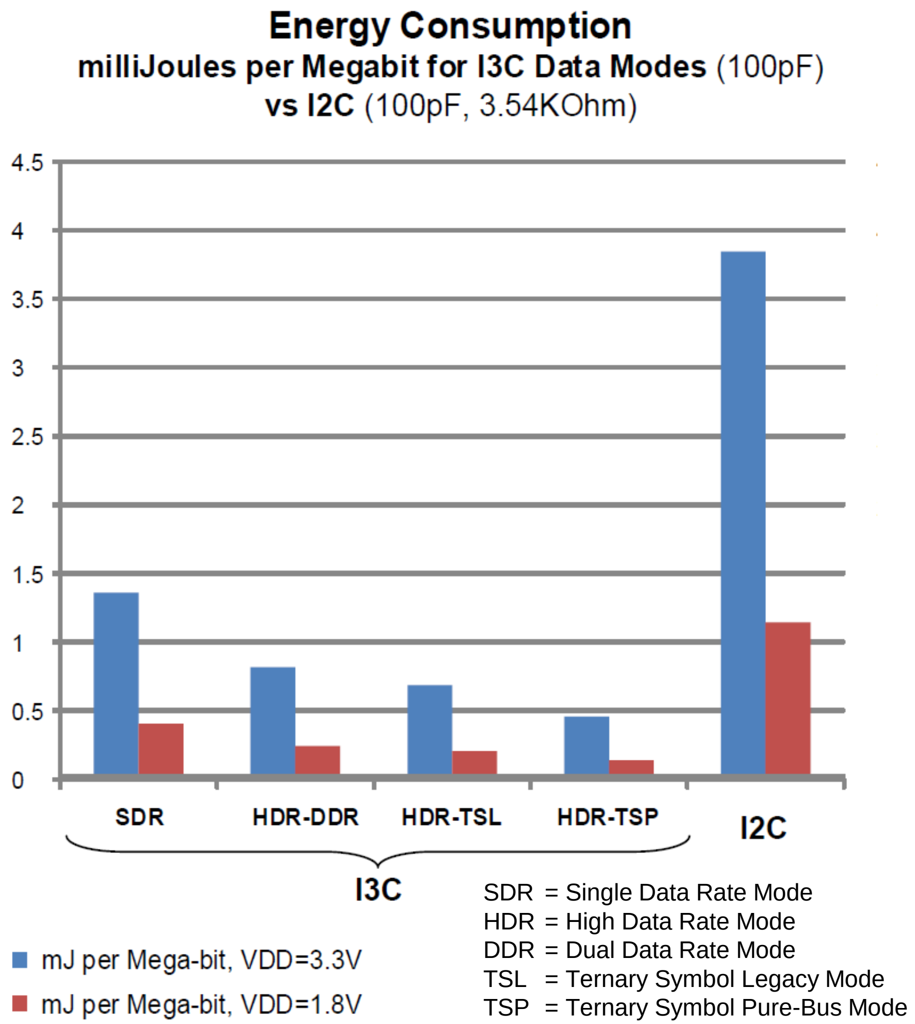
- High data rate modes is optionally available
  - Only Master and specific Slaves can support
    - Other Slaves know how to ignore
  - Has an HDR-DDR form
    - About 2x the data rate of SDR (so about 20Mbps)
    - Also includes CRC
  - Has an HDR-TSP (ternary symbols)
    - Which are up to 3x the data rate (so about 30Mbps)
    - But, this requires a fast internal clock in Slave
    - May have patent issues when used outside of mobile - QC



SDR = Single Data Rate Mode
HDR = High Data Rate Mode
DDR = Dual Data Rate Mode
TSL = Ternary Symbol Legacy Mode
TSP = Ternary Symbol Pure-Bus Mode

Lower Power Consumption

- I3C supports different communication modes
- As data throughput increases, the energy consumption reduces
- Significantly lower power consumption with I3C devices than with legacy I^2C devices



SDR = Single Data Rate Mode
HDR = High Data Rate Mode
DDR = Dual Data Rate Mode
TSL = Ternary Symbol Legacy Mode
TSP = Ternary Symbol Pure-Bus Mode

STATUS OF SPECIFICATION DEVELOPMENT

I3C Specification Development Timeline

- Latest specification document: Version 0.7 (March 4, 2016)
- Target version for initial release: Version 1.0 (Due ~June-2016)

- More Information:
  - I3C Sensor Specification
  - MIPI Alliance Sensor Working Group: http://mipi.org/working-groups/sensor

- NXP will release I3C Slave RTL under BSD License
  - Available to MIPI members under NDA
  - Available with Version 0.9 specification

SUMMARY

Summary

- I3C simplifies system design to a true two-wire interface
- Backward compatible to existing I^2C devices (Supports legacy I^2C messaging)
- Reduces device die size
- Lower power consumption
- Supports in-bound error checking and CRC
- Support peer-to-peer slave communication
- Support hot-plug capability
- Dynamic addressing while supporting Static Addressing for legacy I^2C devices
- Supports I^2C-like SDR messaging and optional HDR messaging (up to 30Mbps)
- Supports multi-master and multi-drop capabilities

SECURE CONNECTIONS
FOR A SMARTER WORLD

ATTRIBUTION STATEMENT

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, CoolFlux, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE Classic, MIFARE DESFire, MIFARE Plus, MIFARE FleX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TrenchMOS, UCODE, Freescale, the Freescale logo, AltiVec, C 5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and μVision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. © 2015–2016 NXP B.V.