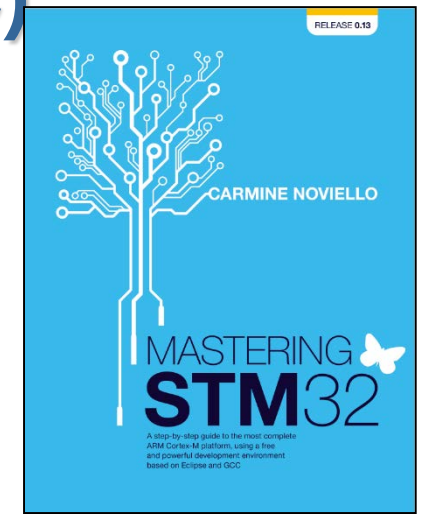
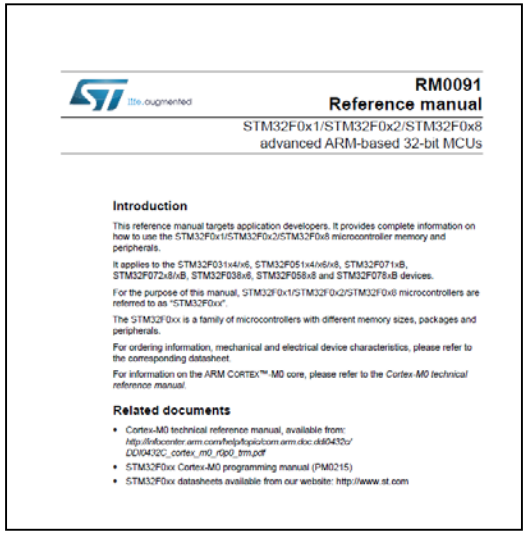
Embedded Communication

Inter-integrated circuit (I^2C)



Chapter 14 I^2C
"Mastering STM32" by Carmine Noviello
https://leanpub.com/mastering-stm32

Chapter 25 Inter-integrated circuit (I^2C)
RM0091: STM32F0x1 Reference manual


RM0091
Reference manual
STM32F0x1/STM32F0x2/STM32F0x8
advanced ARM-based 32-bit MCUs

Introduction

This reference manual targets application developers. It provides complete information on how to use the STM32F0x1/STM32F0x2/STM32F0x8 microcontroller memory and peripherals.

It applies to the STM32F031x4/x6, STM32F051x4/x6/x8, STM32F071xB, STM32F072x8/xB, STM32F038x8, STM32F058x8 and STM32F078xB devices.

For the purpose of this manual, STM32F0x1/STM32F0x2/STM32F0x8 microcontrollers are referred to as "STM32F0xx".

The STM32F0xx is a family of microcontrollers with different memory sizes, packages and peripherals.

For ordering information, mechanical and electrical device characteristics, please refer to the corresponding datasheet.

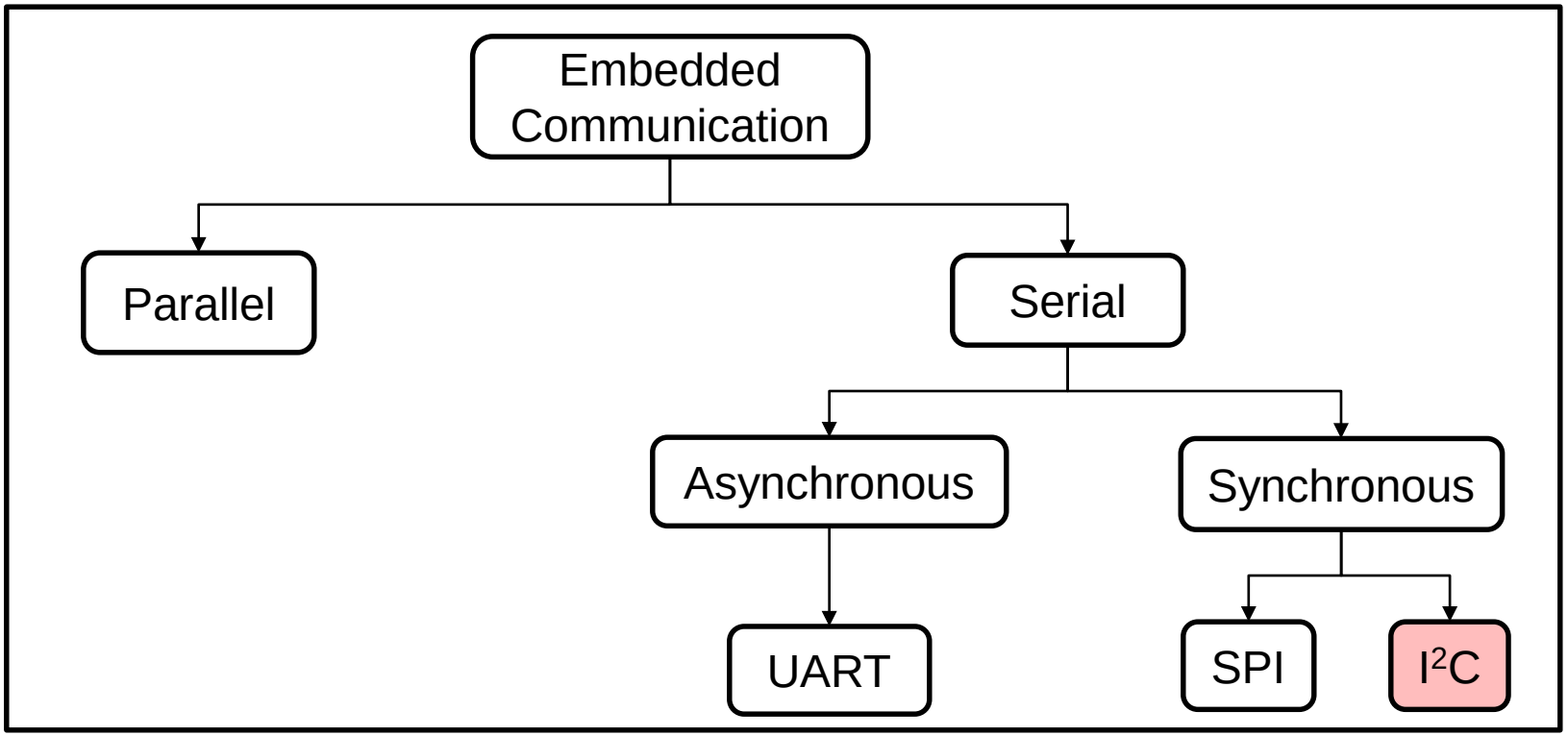
For information on the ARM CORTEX™-M0 core, please refer to the *Cortex-M0 technical reference manual*.

Related documents

- Cortex-M0 technical reference manual, available from: *http://infocenter.arm.com/help/topic/com.arm.doc.ddi0432c/DDI0432C_cortex_m0_r0p0_trm.pdf*
- STM32F0xx Cortex-M0 programming manual (PM0215)
- STM32F0xx datasheets available from our website: http://www.st.com

I²C and the big picture
Context

- I²C is a serial synchronous type of communication
  - Two devices share a common clock signal to synchronise the exchange of data



Overview of embedded communication

Inter-integrated Circuit (I^2C)

Introduction and basic concepts

Inter-integrated Circuit (I^2C)
Introduction and Basic Concepts

- History
  - I^2C was developed by Phillips Semiconductor in 1982 to connect low speed devices on the same PCB
  - I^2C is also called a “two wire interface”
  - Synchronous serial data transfer

Inter-integrated Circuit (I^2C)
Introduction and Basic Concepts

- History
  - I^2C was developed by Phillips Semiconductor in 1982 to connect low speed devices on the same PCB
  - I^2C is also called a “two wire interface”
  - Synchronous serial data transfer
- Why I^2C over SPI or UART/RS232?
  - Uses fewer connections: 2 wires instead of 4 for SPI
  - The message structure includes an “acknowledge” command, for verification purposes:
    - The master to verify that a slave exists
    - When the master writes to the slave, the acknowledge command is used to verify that the data was received by the slave

Inter-integrated Circuit (I^2C)
Introduction and Basic Concepts

- History
  - I^2C was developed by Phillips Semiconductor in 1982 to connect low speed devices on the same PCB
  - I^2C is also called a “two wire interface”
  - Synchronous serial data transfer
- Why I^2C over SPI or UART/RS232?
- Where is I^2C used?
  - Analogue to Digital Converters (ADC)
  - Pressure sensors
  - Real time clock
  - Memory devices (EEPROM)

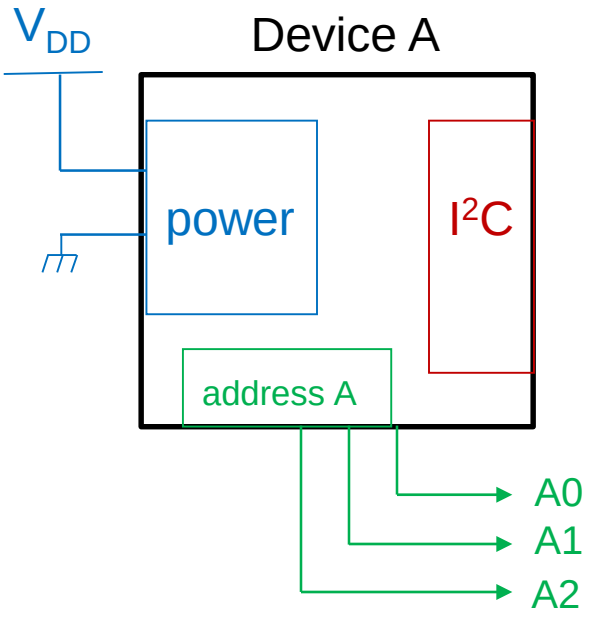
Inter-integrated Circuit (I^2C)
Introduction and Basic Concepts

- History
  - I^2C was developed by Phillips Semiconductor in 1982 to connect low speed devices on the same PCB
  - I^2C is also called a “two wire interface”
  - Synchronous serial data transfer
- Why I^2C over SPI or UART/RS232?
- Where is I^2C used?
- Data transfer speeds
  - Low speed : 10 kbits/s
  - Standard speed : 100 kbits/s
  - Fast speed : 400 kbits/s

Inter-integrated Circuit (I^2C)
Connections

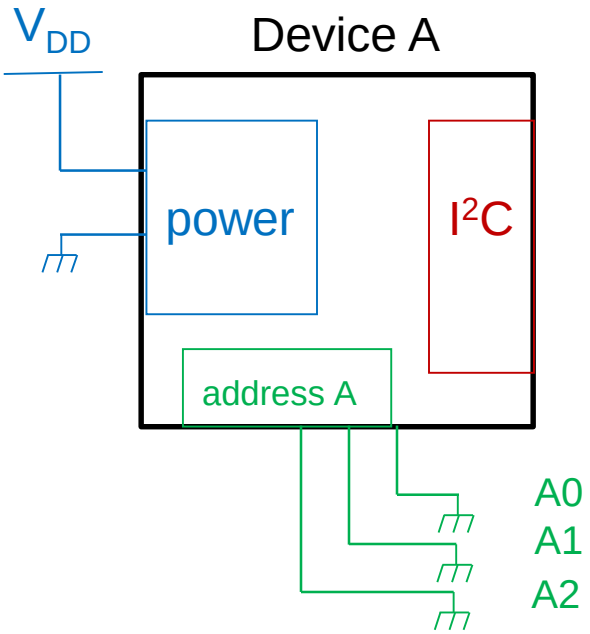
Inter-integrated Circuit (I^2C)
Connection: single device

- Device address
  - An I^2C device has a 7-bit address: A6, A5, A4, A3, A2, A1, A0
  - Typically, some bits are set by hardware. Example: A2, A1, A0
  - Typically, some bits are fixed. Example: A6, A5, A4, A3 = ‘0011’



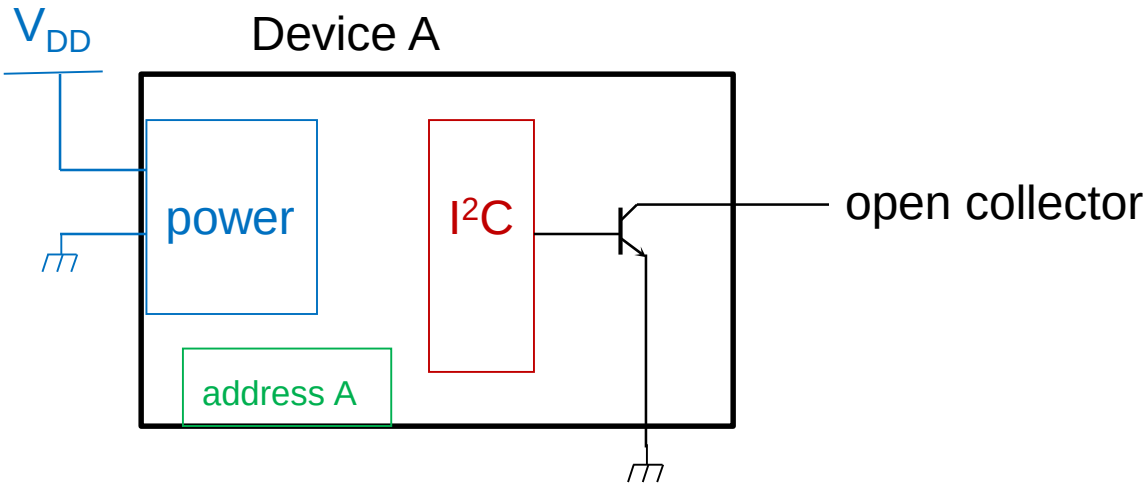
Inter-integrated Circuit (I^2C)
Connection: single device

- Device address
  - An I^2C device has a 7-bit address: A6, A5, A4, A3, A2, A1, A0
  - Typically, some bits are set by hardware. Example: A2, A1, A0
  - Typically, some bits are fixed. Example: A6, A5, A4, A3 = ‘0011’
  - Setting A2, A1, A0 = ‘000’



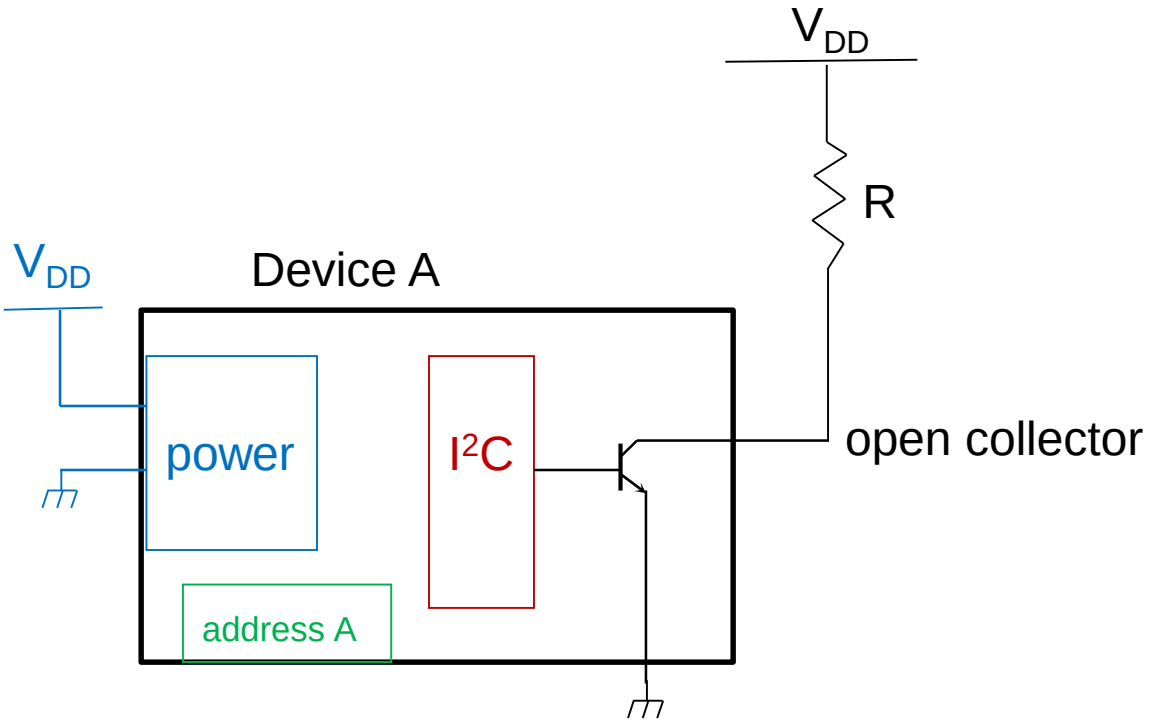
Inter-integrated Circuit (I^2C)
Connection: single device

- Device address
- Output
  - I^2C have open collectors or open drains on their outputs



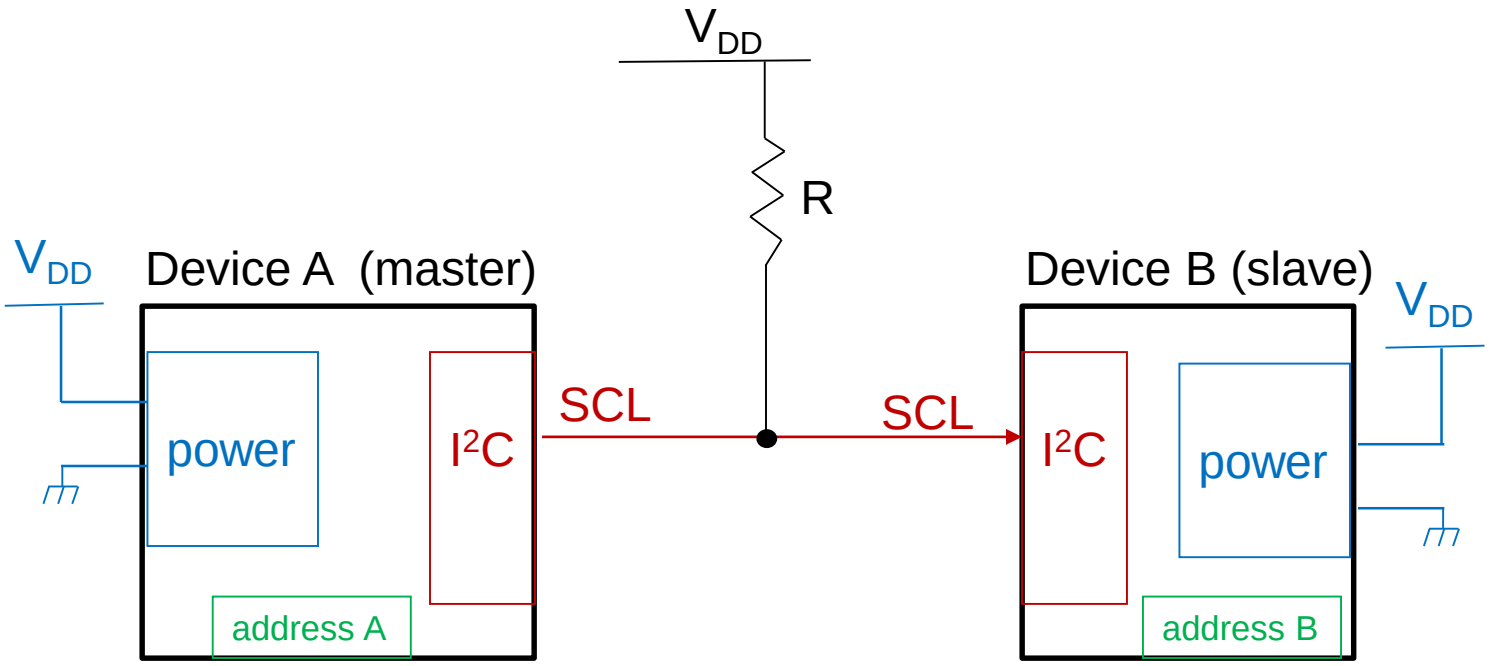
Inter-integrated Circuit (I^2C)
Connection: single device

- Device address
- Output
  - I^2C have open collectors or open drains on their outputs
  - External pull-up resistors are needed



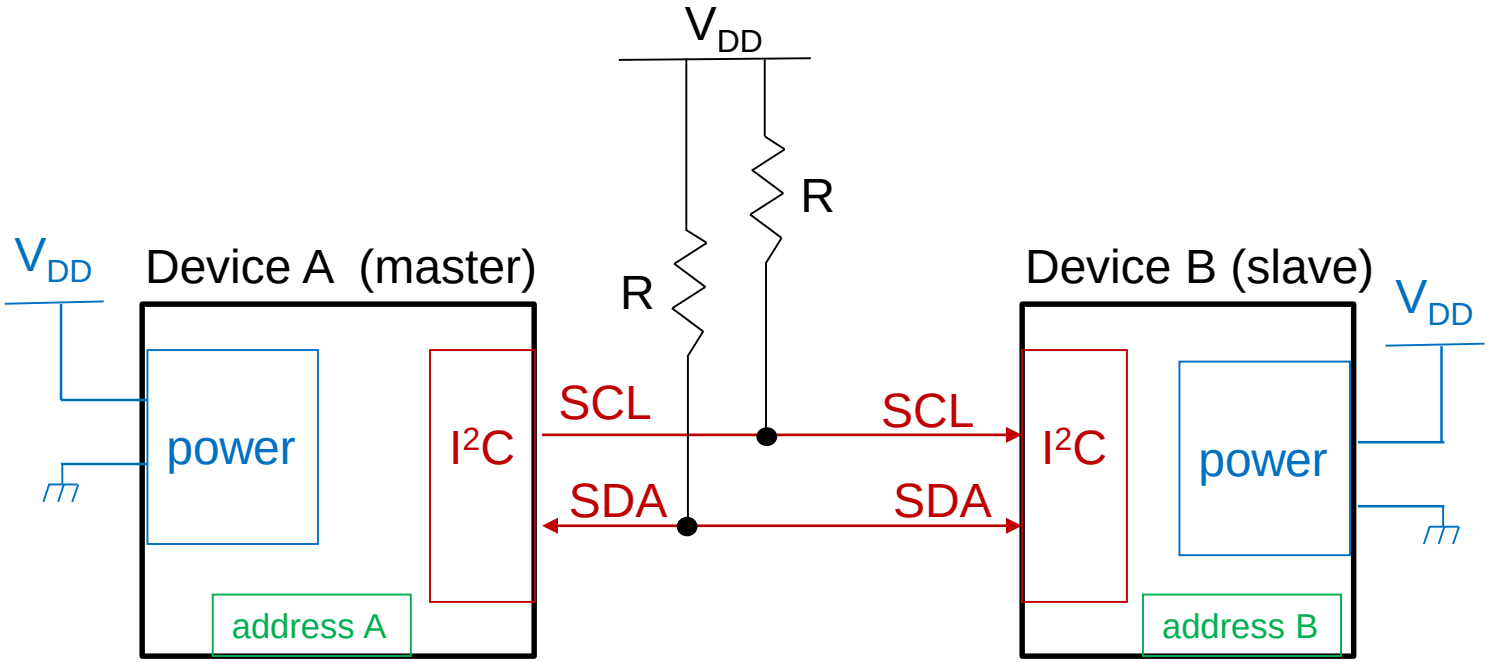
Inter-integrated Circuit (I^2C)
Connection: one to one

- Master and slave
  - The device that generates the clock signal (SCK) is termed “master”



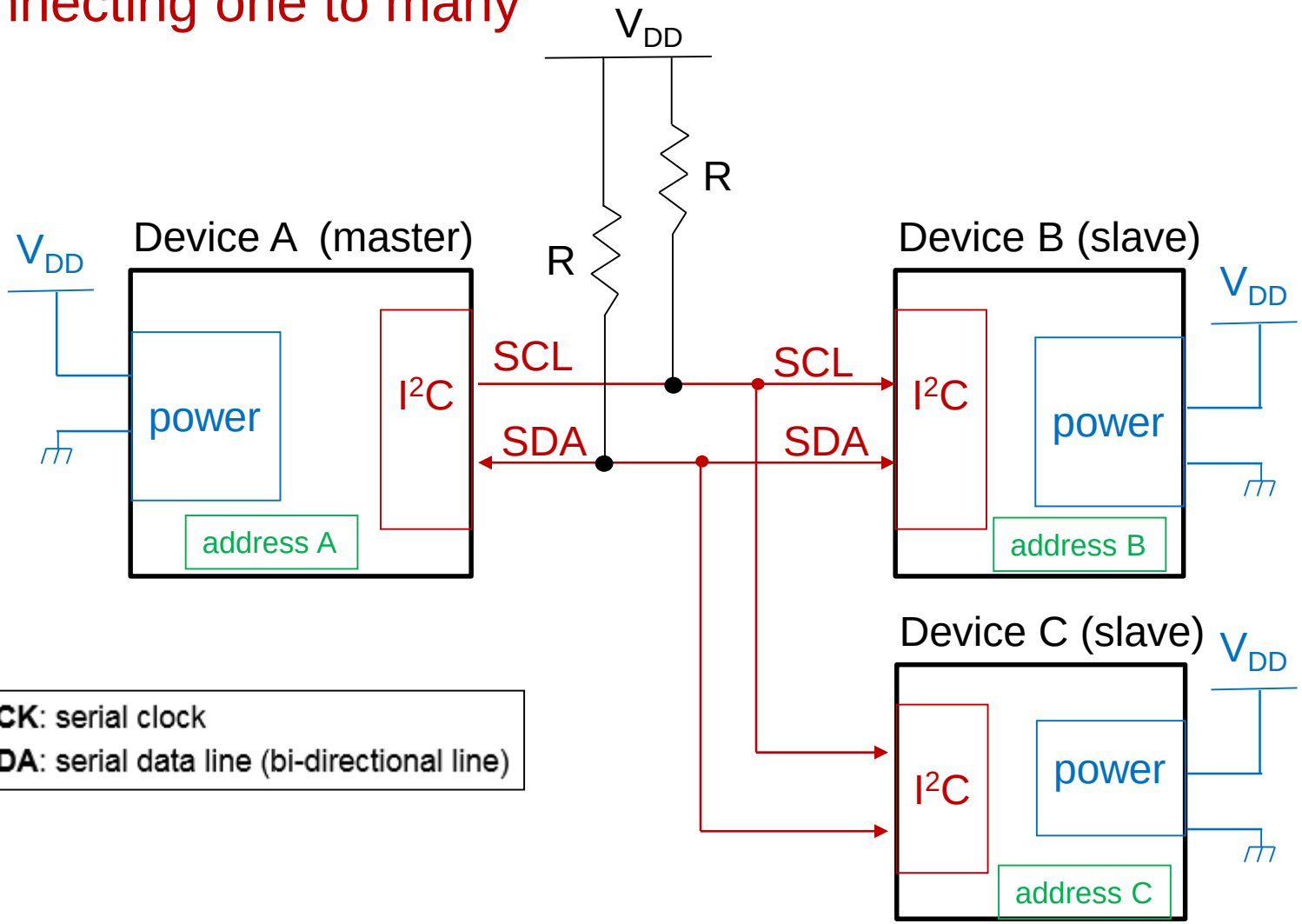
Inter-integrated Circuit (I^2C)
Connection: one to one

- Master and slave
  - The device that generates the clock signal (SCK) is termed “master”
- Connecting one to one
  - **SCL**: serial clock
  - **SDA**: serial data line (bi-directional line)



Inter-integrated Circuit (I^2C)
Connection: one to many

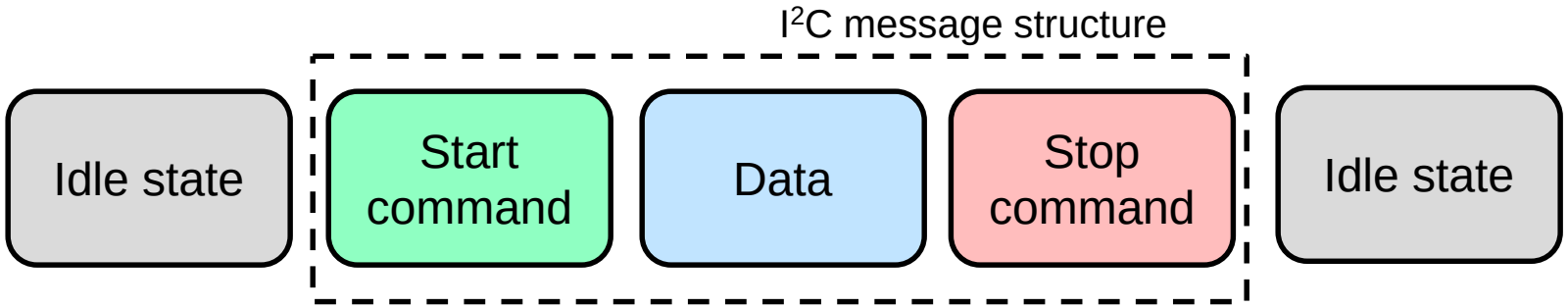
- Connecting one to many



Inter-integrated Circuit (I^2C)
Message Structure

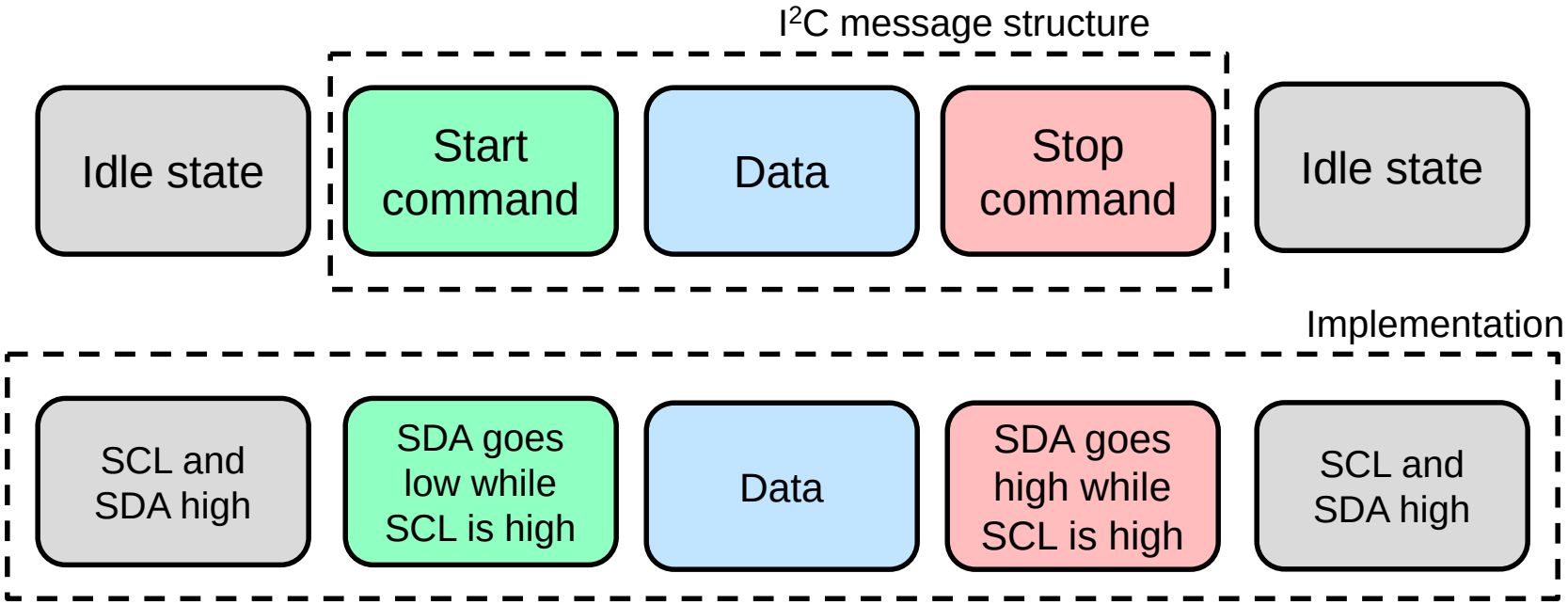
Inter-integrated Circuit (I^2C)

Message Structure



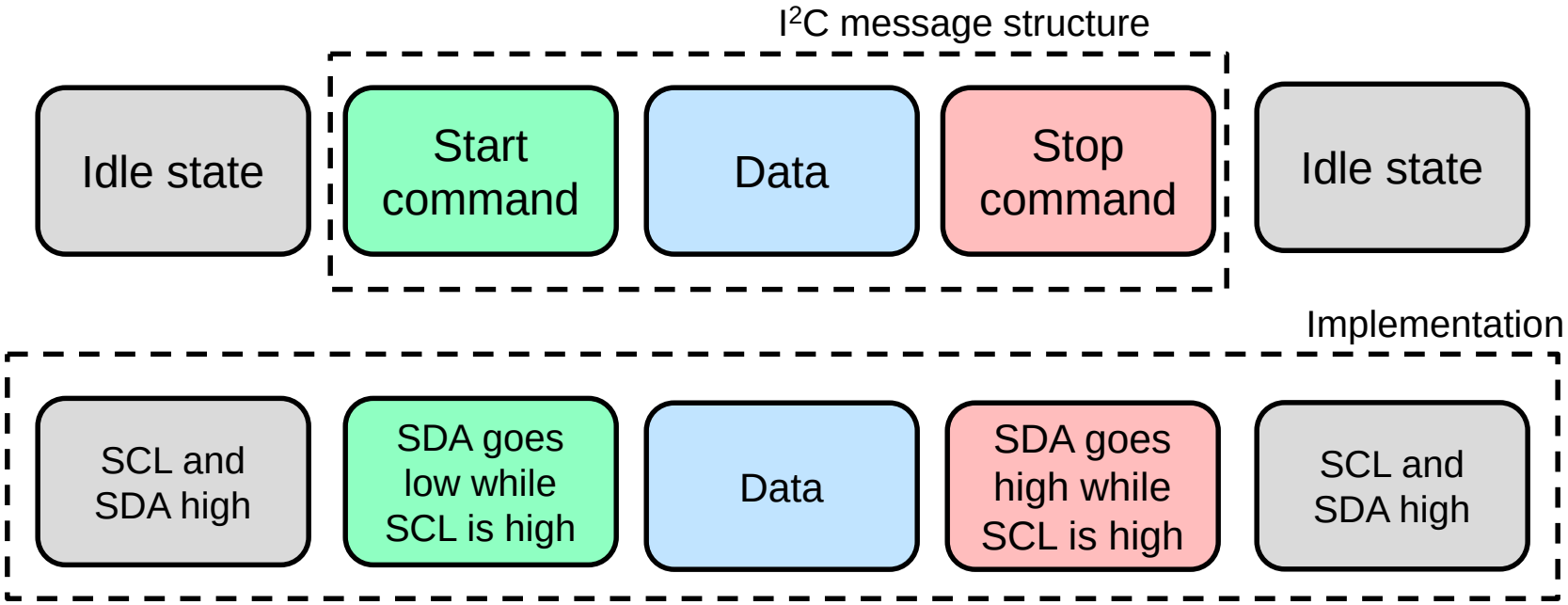
Inter-integrated Circuit (I^2C)

Message Structure



Inter-integrated Circuit (I^2C)

Message Structure



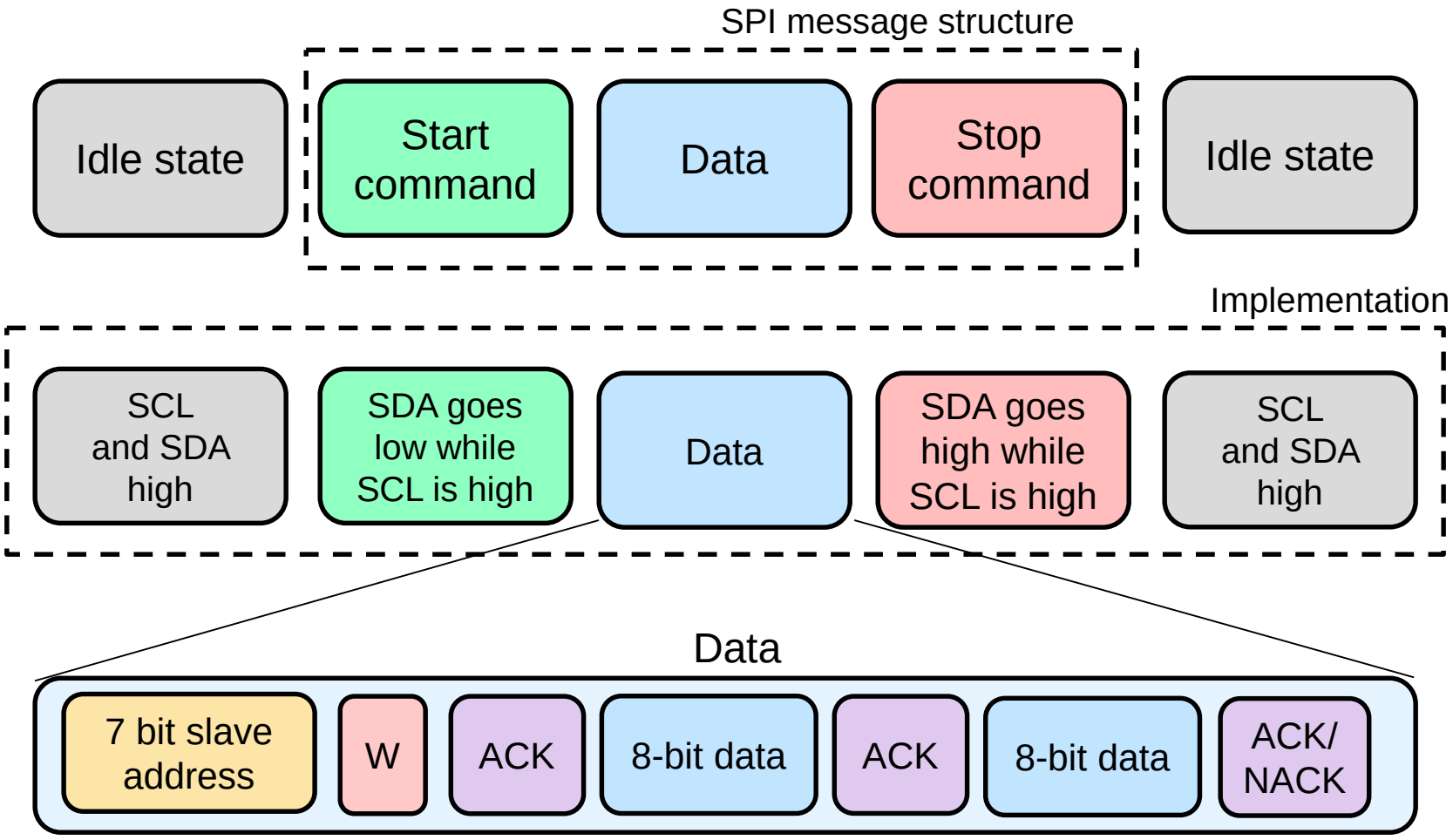
- **Idle state** : both clock signal (SCL) and SDA are high
- **Start command** : SDA transitions from high to low while SCL is high
- **Data** : more details given in later slides
- **Stop command** : SDA transitions from low to high while SCL is high
- **Idle state** : both clock signal (SCL) and SDA are high

Inter-integrated Circuit (I^2C)

Message Structure: master transfers data to slave

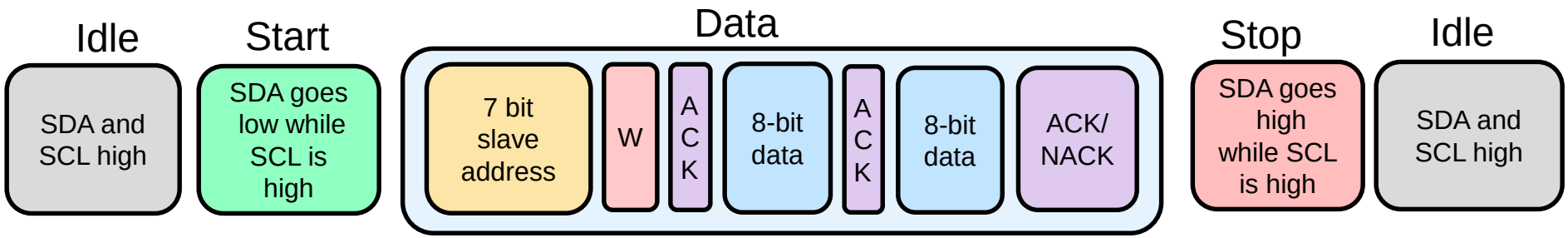
Inter-integrated Circuit (I^2C)

Message Structure: master transfers data to slave

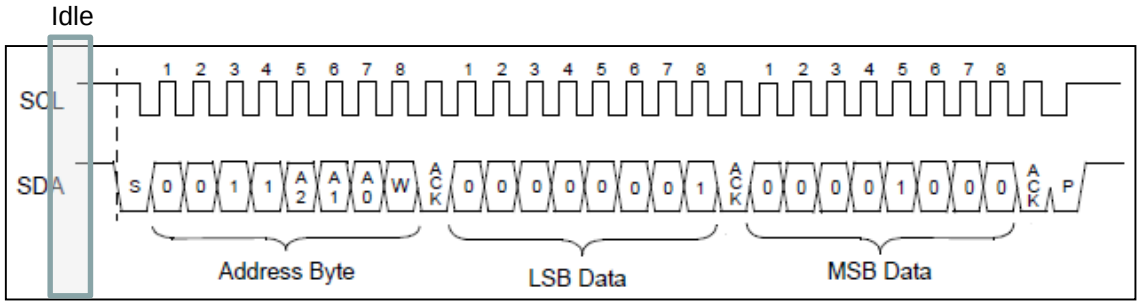


Inter-integrated Circuit (I^2C)

Message Structure: master transfers data to slave

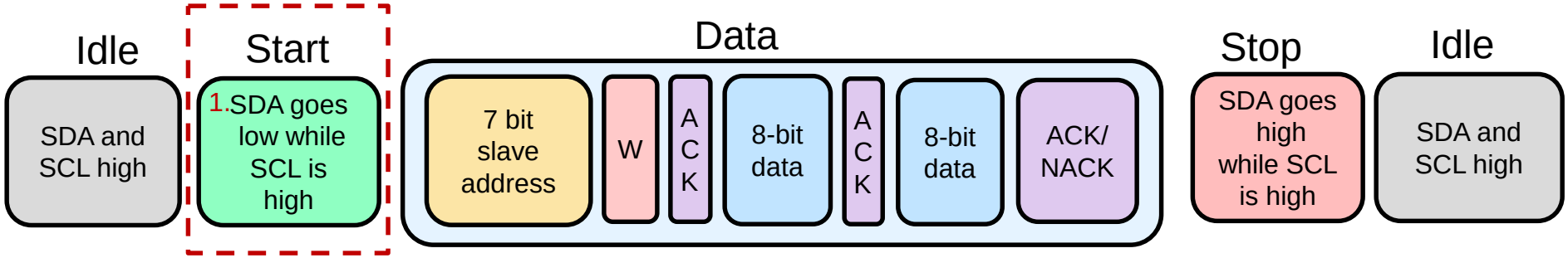


- Data transfer from master to slave

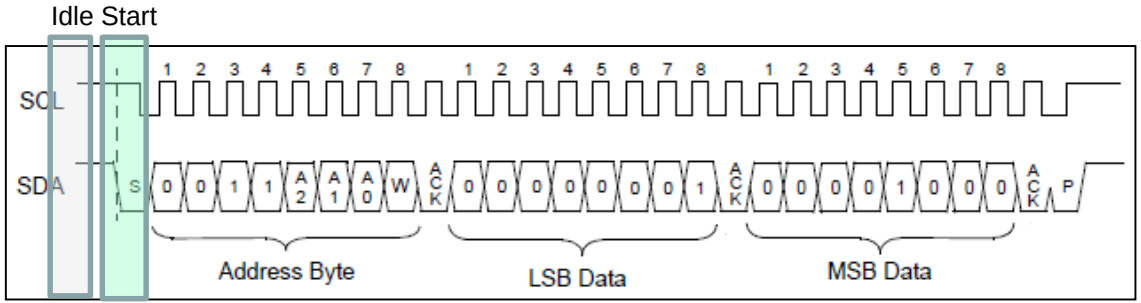


Inter-integrated Circuit (I^2C)

Message Structure: master transfers data to slave

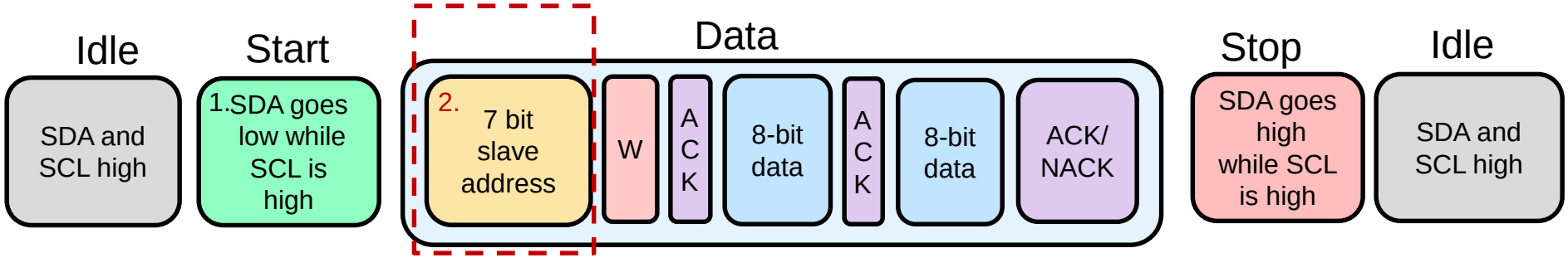


- Data transfer from master to slave
  1. Master transmits a **start command**: transition SDA from high to low

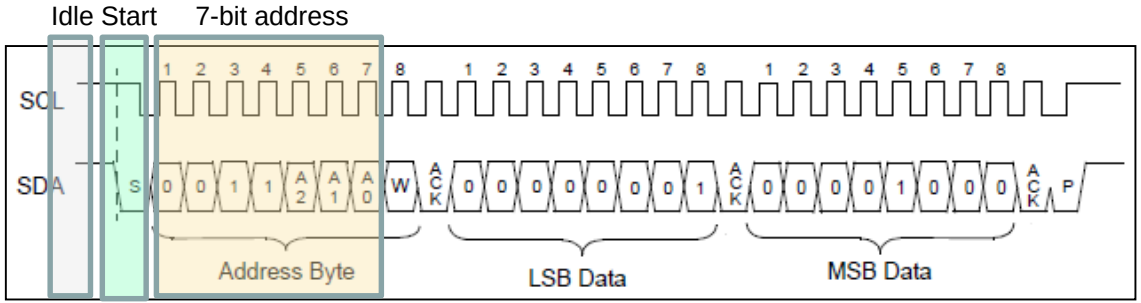


Inter-integrated Circuit (I^2C)

Message Structure: master transfers data to slave

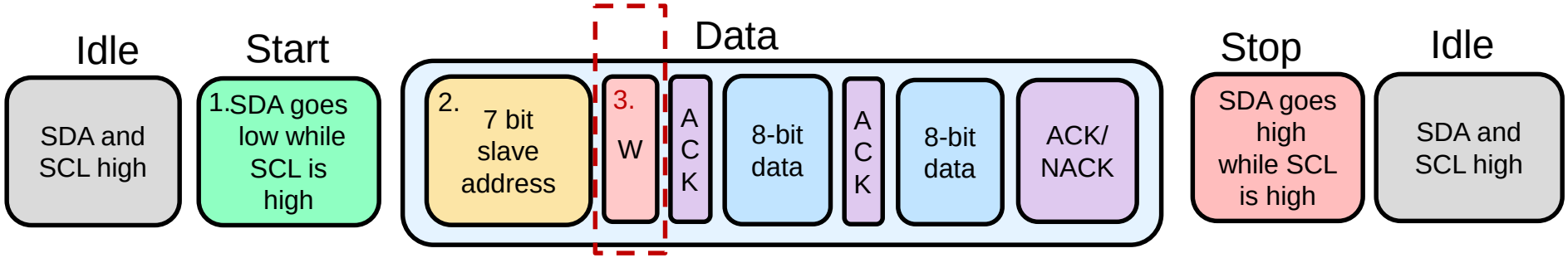


- Data transfer from master to slave
  1. Master transmits a **start command**: transition SDA from high to low
  2. Master transmits the **7-bit slave address**

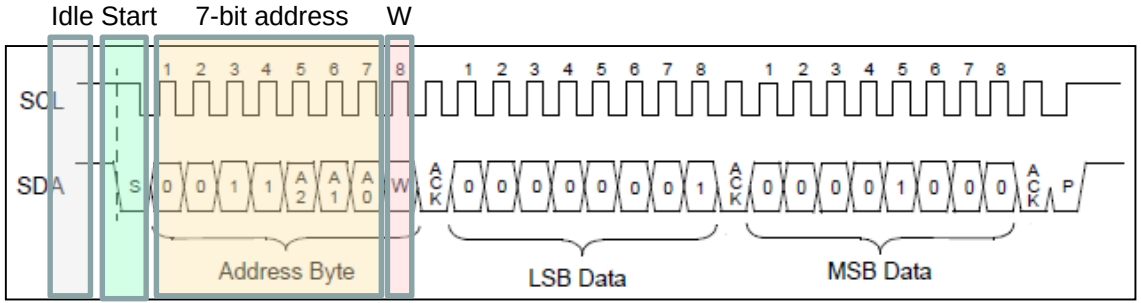


Inter-integrated Circuit (I^2C)

Message Structure: master transfers data to slave

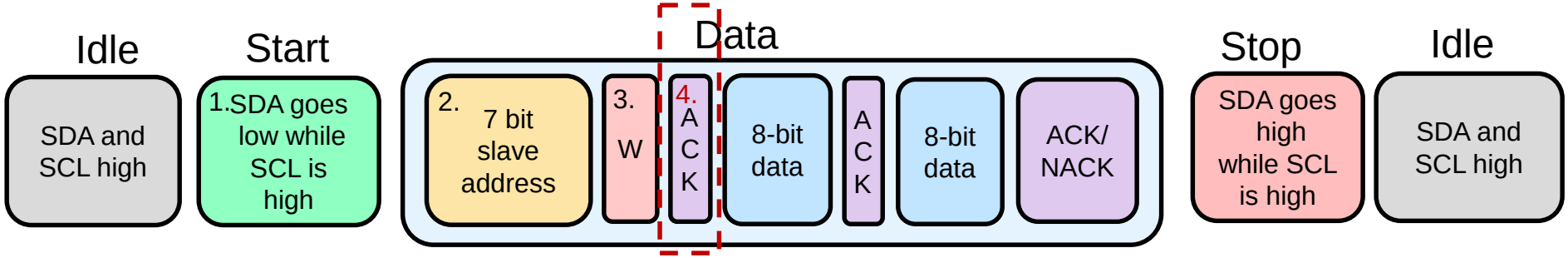


- Data transfer from master to slave
  1. Master transmits a **start command**: transition SDA from high to low
  2. Master transmits the **7-bit slave address**
  3. Master transmits a **write command** (logic 0)

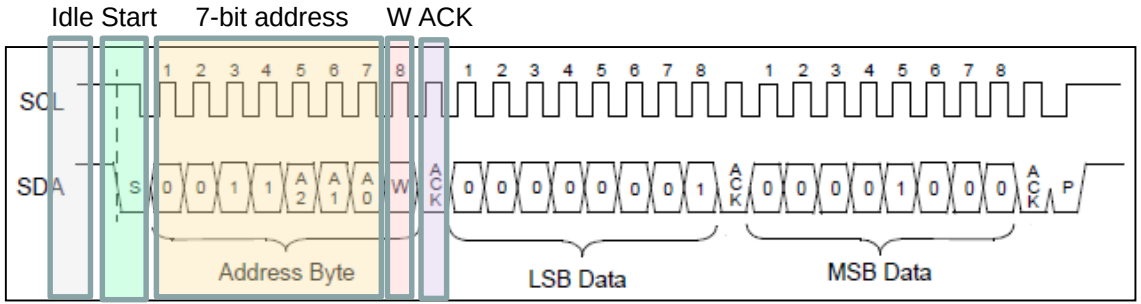


Inter-integrated Circuit (I^2C)

Message Structure: master transfers data to slave

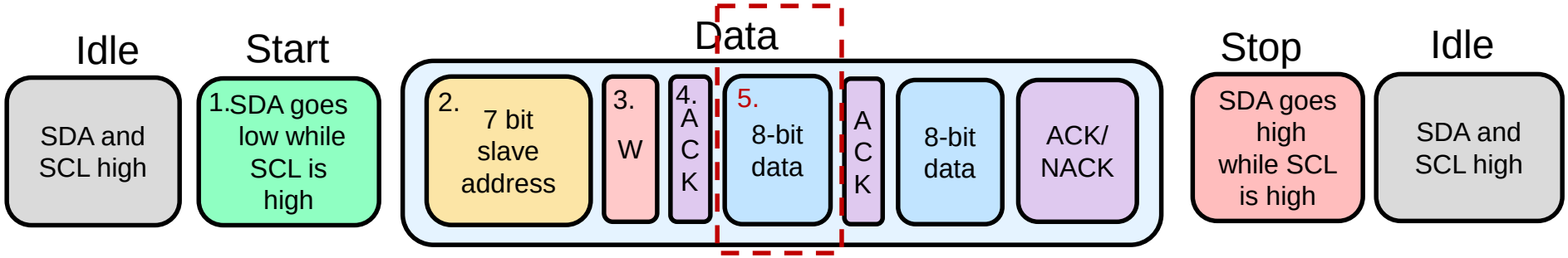


- Data transfer from master to slave
  1. Master transmits a **start command**: transition SDA from high to low
  2. Master transmits the **7-bit slave address**
  3. Master transmits a **write command** (logic 0)
  4. Slave sends an **'ACK' command** (logic 0) to let master know that it is present

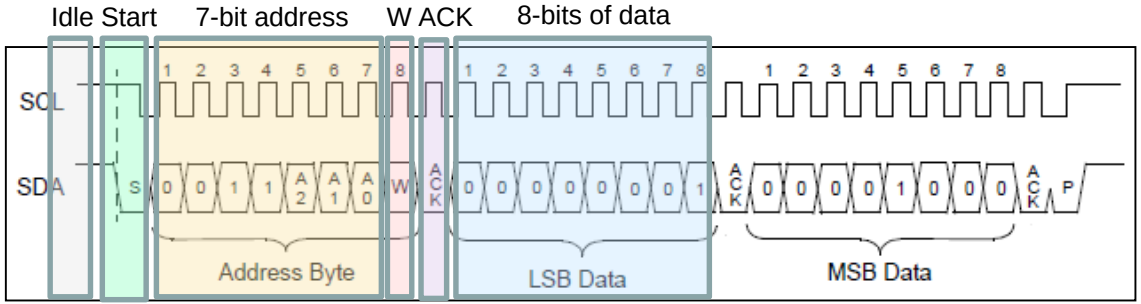


Inter-integrated Circuit (I^2C)

Message Structure: master transfers data to slave

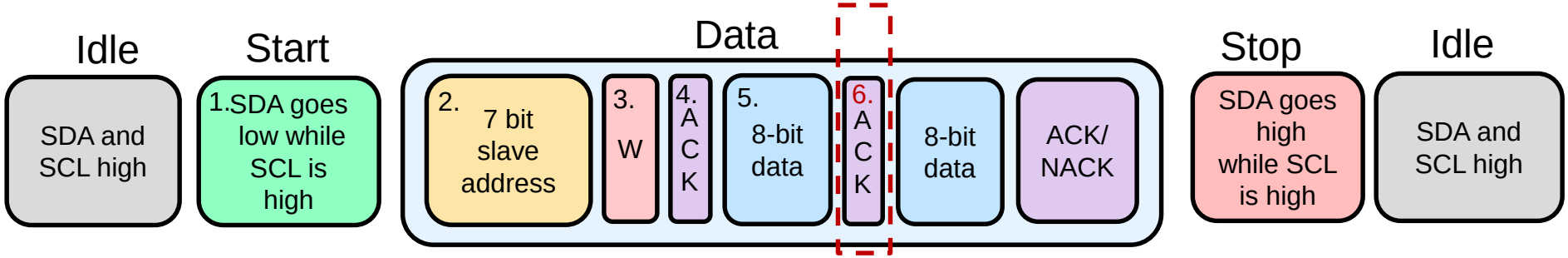


- Data transfer from master to slave
  1. Master transmits a **start command**: transition SDA from high to low
  2. Master transmits the **7-bit slave address**
  3. Master transmits a **write command** (logic 0)
  4. Slave sends an **'ACK' command** (logic 0) to let master know that it is present
  5. Master transmits **8-bits of data**

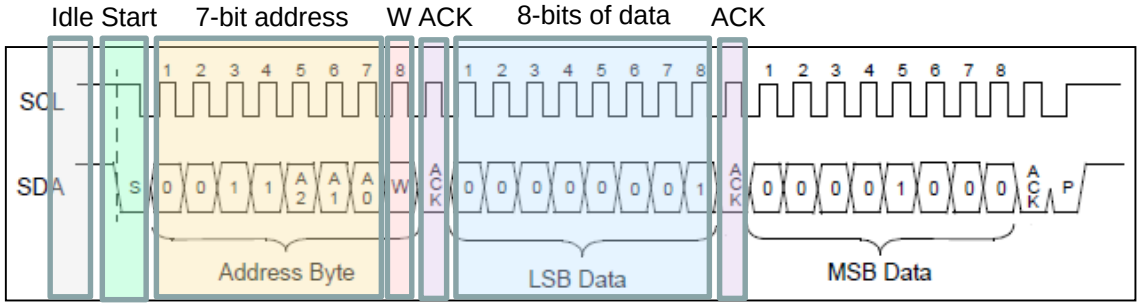


Inter-integrated Circuit (I^2C)

Message Structure: master transfers data to slave

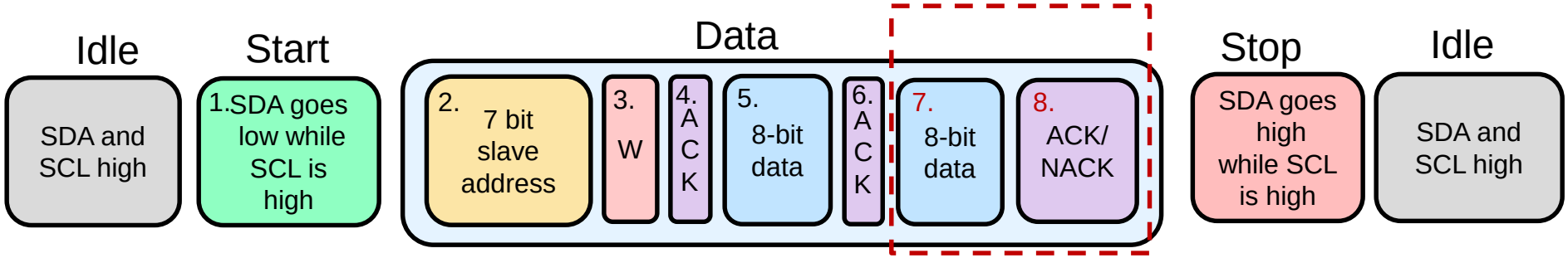


- Data transfer from master to slave
  1. Master transmits a **start command**: transition SDA from high to low
  2. Master transmits the **7-bit slave address**
  3. Master transmits a **write command** (logic 0)
  4. Slave sends an **'ACK' command** (logic 0) to let master know that it is present
  5. Master transmits **8-bits of data**
  6. Slave sends an **'ACK' command** (logic 0) to acknowledge that 8-bits of data was received

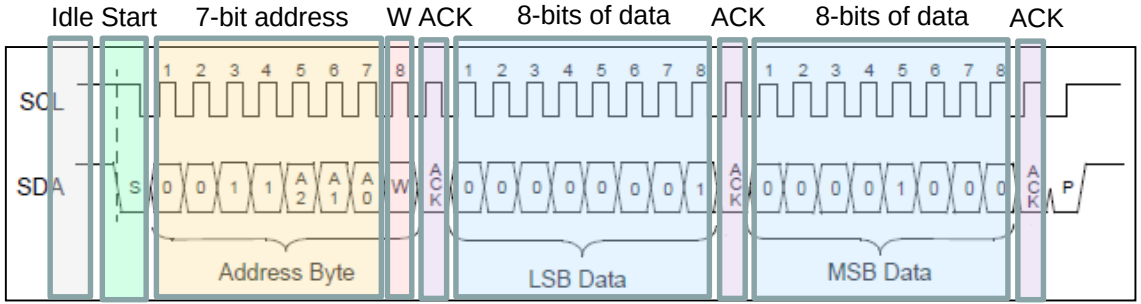


Inter-integrated Circuit (I^2C)

Message Structure: master transfers data to slave

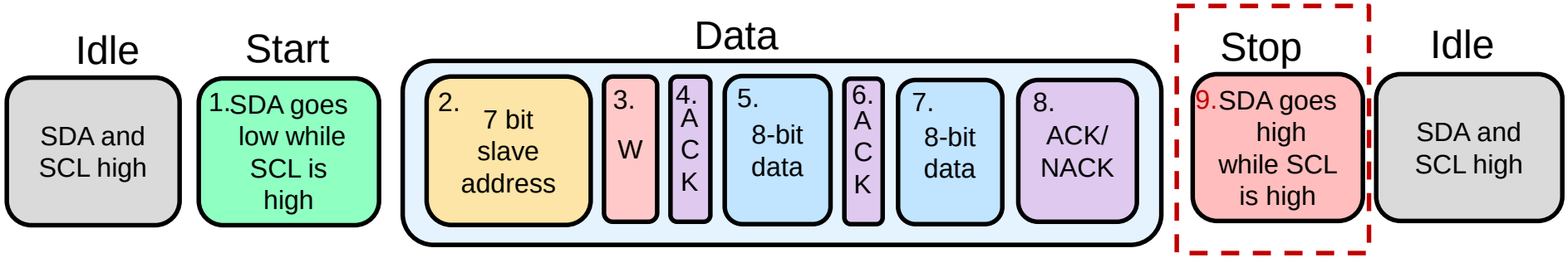


- Data transfer from master to slave
  1. Master transmits a **start command**: transition SDA from high to low
  2. Master transmits the **7-bit slave address**
  3. Master transmits a **write command** (logic 0)
  4. Slave sends an **'ACK' command** (logic 0) to let master know that it is present
  5. Master transmits **8-bits of data**
  6. Slave sends an **'ACK' command** (logic 0) to acknowledge that 8-bits of data was received
  7. Master transmits **8-bits of data**
  8. Slave sends **'ACK' command** (logic 0) to notify master to stop sending data

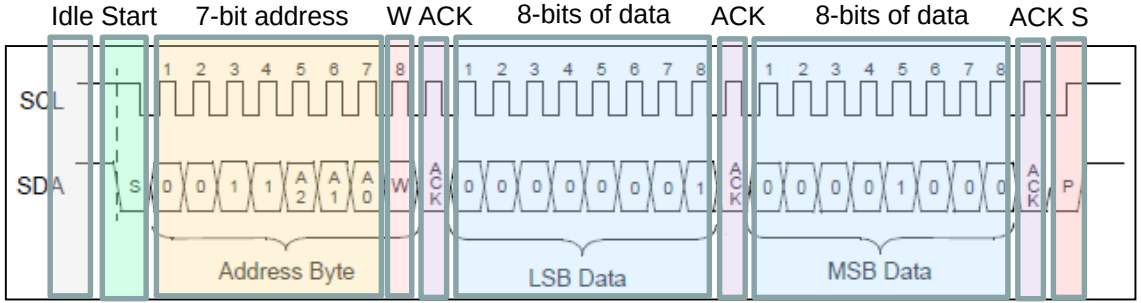


Inter-integrated Circuit (I^2C)

Message Structure: master transfers data to slave

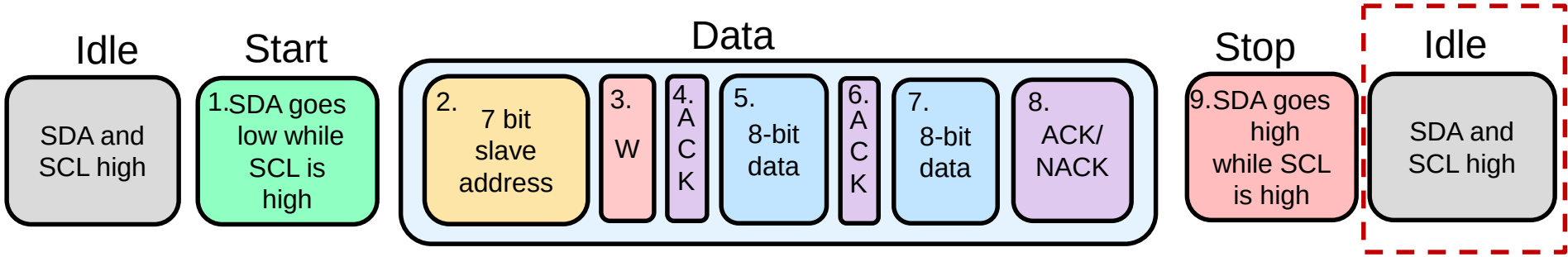


- Data transfer from master to slave
  1. Master transmits a **start command**: transition SDA from high to low
  2. Master transmits the **7-bit slave address**
  3. Master transmits a **write command** (logic 0)
  4. Slave sends an **'ACK' command** (logic 0) to let master know that it is present
  5. Master transmits **8-bits of data**
  6. Slave sends an **'ACK' command** (logic 0) to acknowledge that 8-bits of data was received
  7. Master transmits **8-bits of data**
  8. Slave sends **'ACK' command** (logic 0) to notify master to stop sending data
  9. Master transmits **stop command**: transition SDA form low to high

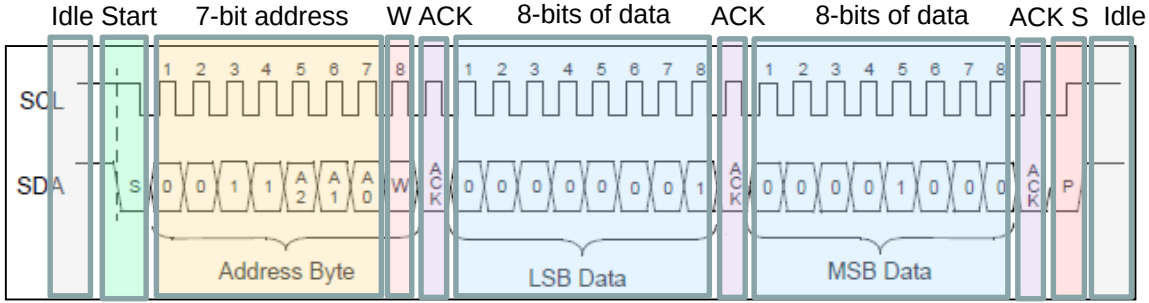


Inter-integrated Circuit (I^2C)

Message Structure: master transfers data to slave

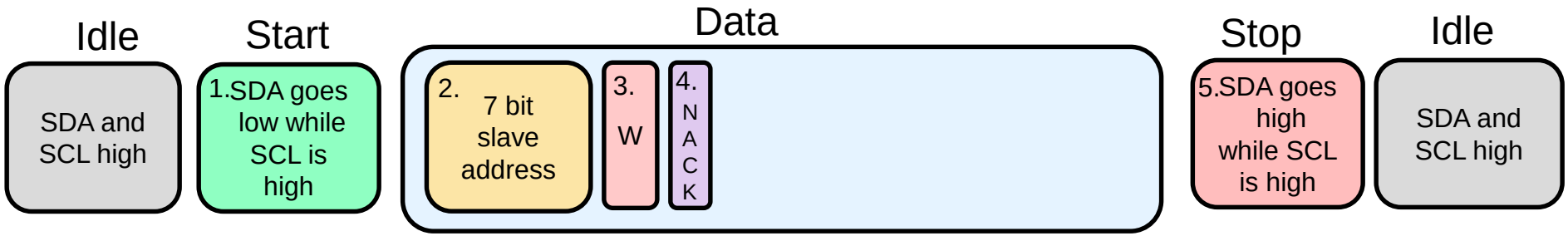


- Data transfer from master to slave
  1. Master transmits a **start command**: transition SDA from high to low
  2. Master transmits the **7-bit slave address**
  3. Master transmits a **write command** (logic 0)
  4. Slave sends an **'ACK' command** (logic 0) to let master know that it is present
  5. Master transmits **8-bits of data**
  6. Slave sends an **'ACK' command** (logic 0) to acknowledge that 8-bits of data was received
  7. Master transmits **8-bits of data**
  8. Slave sends **'ACK' command** (logic 0) to notify master to stop sending data
  9. Master transmits **stop command**: transition SDA form low to high

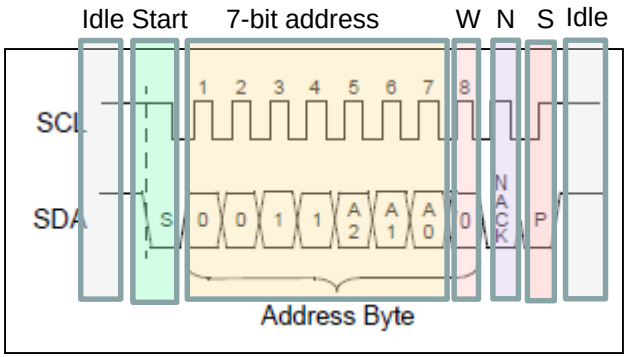


Inter-integrated Circuit (I^2C)

Message Structure: master transfers data to slave



- Data transfer from master to slave: when a slave is not present
  - A **'NACK' command** (logic 1) is received by the master: means that a slave with this address is not present
  - Thereafter, the master sends a **stop command**

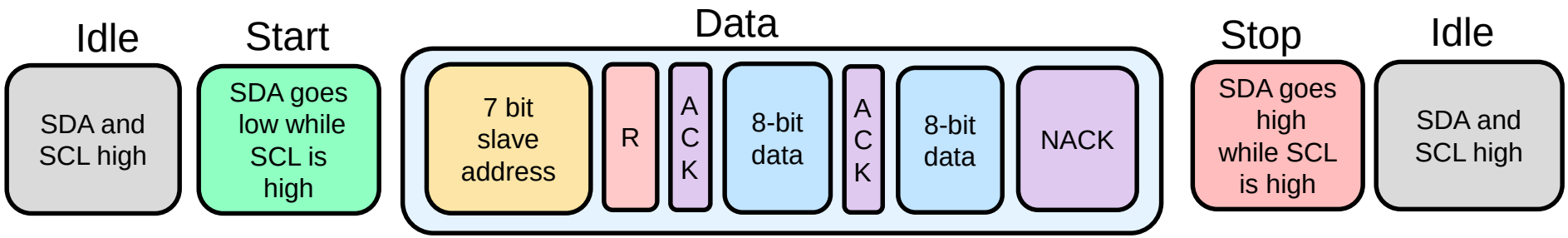


Inter-integrated Circuit (I^2C)

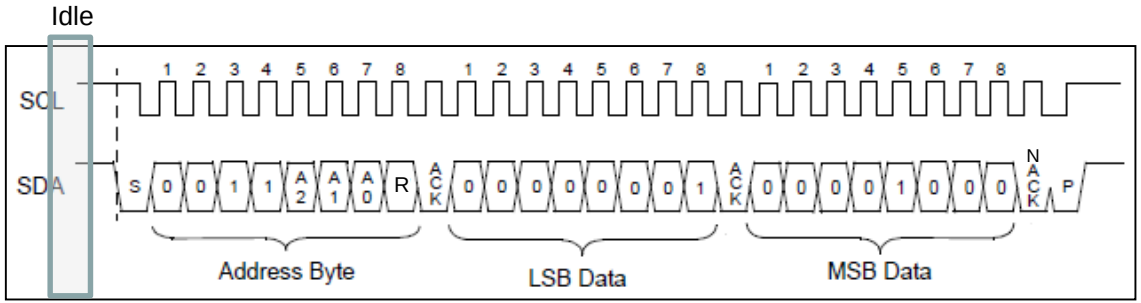
Message Structure: slave transfers data to master

Inter-integrated Circuit (I^2C)

Message Structure: slave transfers data to master

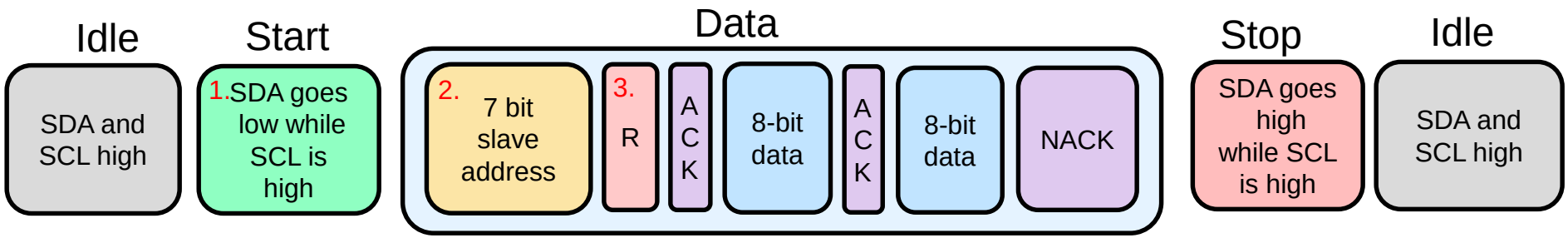


- Data transfer from slave to master

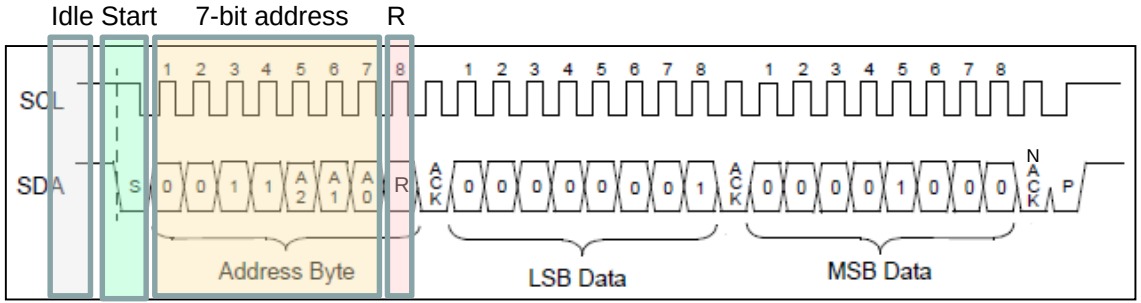


Inter-integrated Circuit (I^2C)

Message Structure: slave transfers data to master

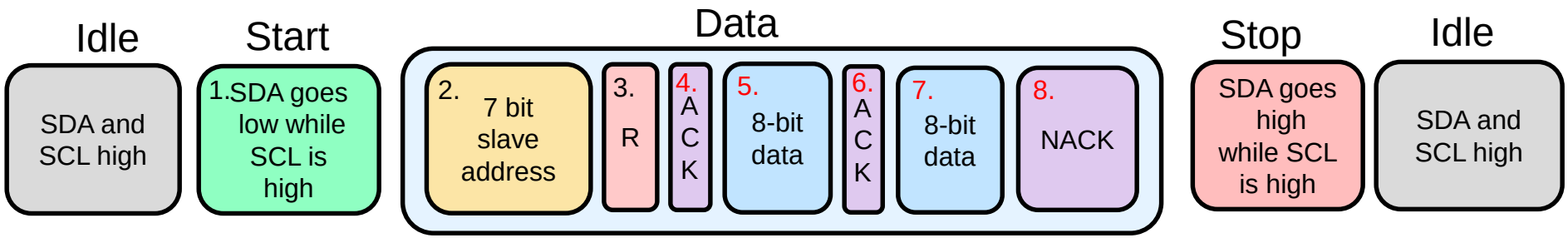


- Data transfer from slave to master
  1. Master transmits a **start command**: transition SDA from high to low
  2. Master transmits the **7-bit slave address**
  3. Master transmits a **read command** (logic 1)

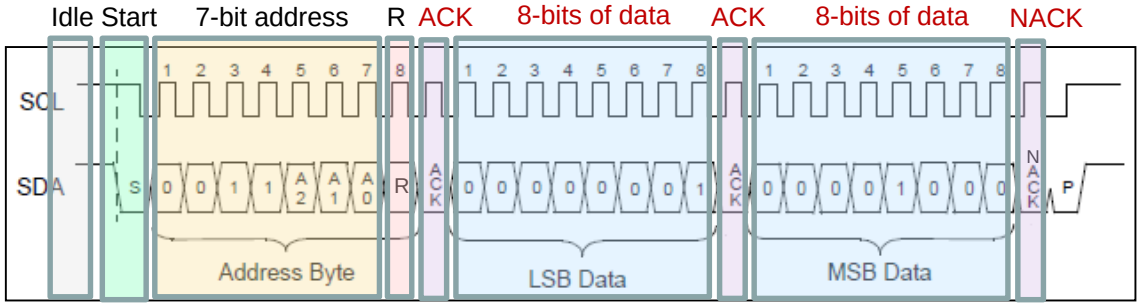


Inter-integrated Circuit (I^2C)

Message Structure: slave transfers data to master

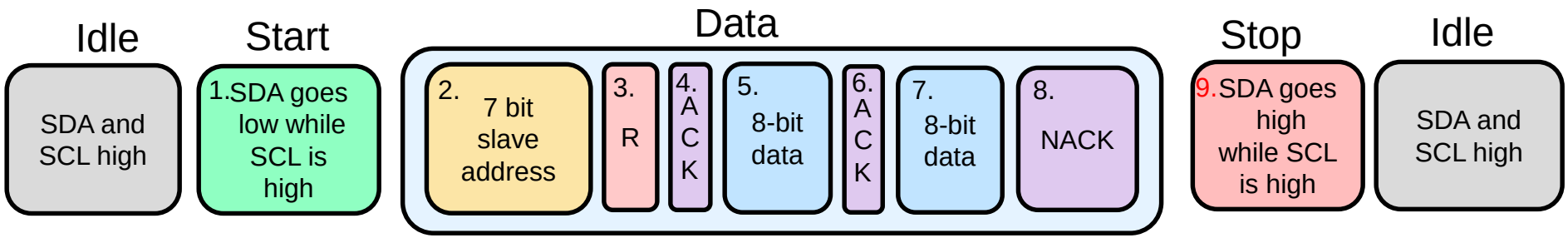


- Data transfer from slave to master
  1. Master transmits a **start command**: transition SDA from high to low
  2. Master transmits the **7-bit slave address**
  3. Master transmits a **read command** (logic 1)
  4. Slave sends an **'ACK' command** (logic 0) to let master know that it is present
  5. Slave transmits **8-bits of data**
  6. Master sends an **'ACK' command** (logic 0) to acknowledge that 8-bits of data was received
  7. Slave transmits **8-bits of data**
  8. Master sends **'NACK' command** (logic 1) to notify slave to stop sending data

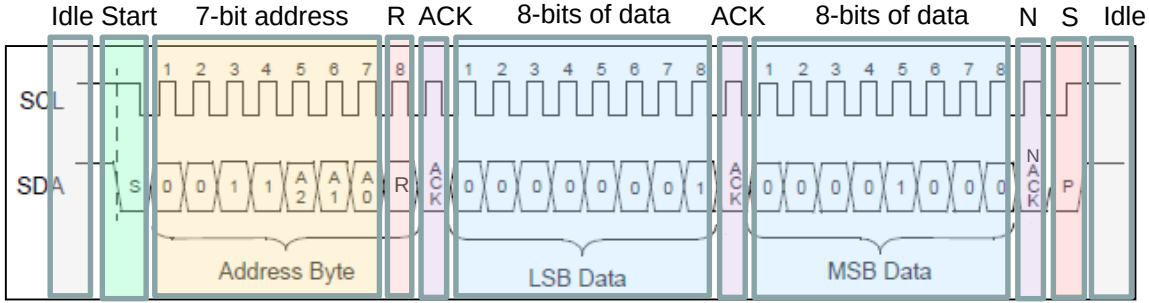


Inter-integrated Circuit (I^2C)

Message Structure: slave transfers data to master



- Data transfer from slave to master
  1. Master transmits a **start command**: transition SDA from high to low
  2. Master transmits the **7-bit slave address**
  3. Master transmits a **read command** (logic 1)
  4. Slave sends an **'ACK' command** (logic 0) to let master know that it is present
  5. Slave transmits **8-bits of data**
  6. Master sends an **'ACK' command** to acknowledge that 8-bits of data was received
  7. Slave transmits **8-bits of data**
  8. Master sends **'NACK' command** (logic 1) to notify slave to stop sending data
  9. Master sends **stop command**: transition SDA from low to high



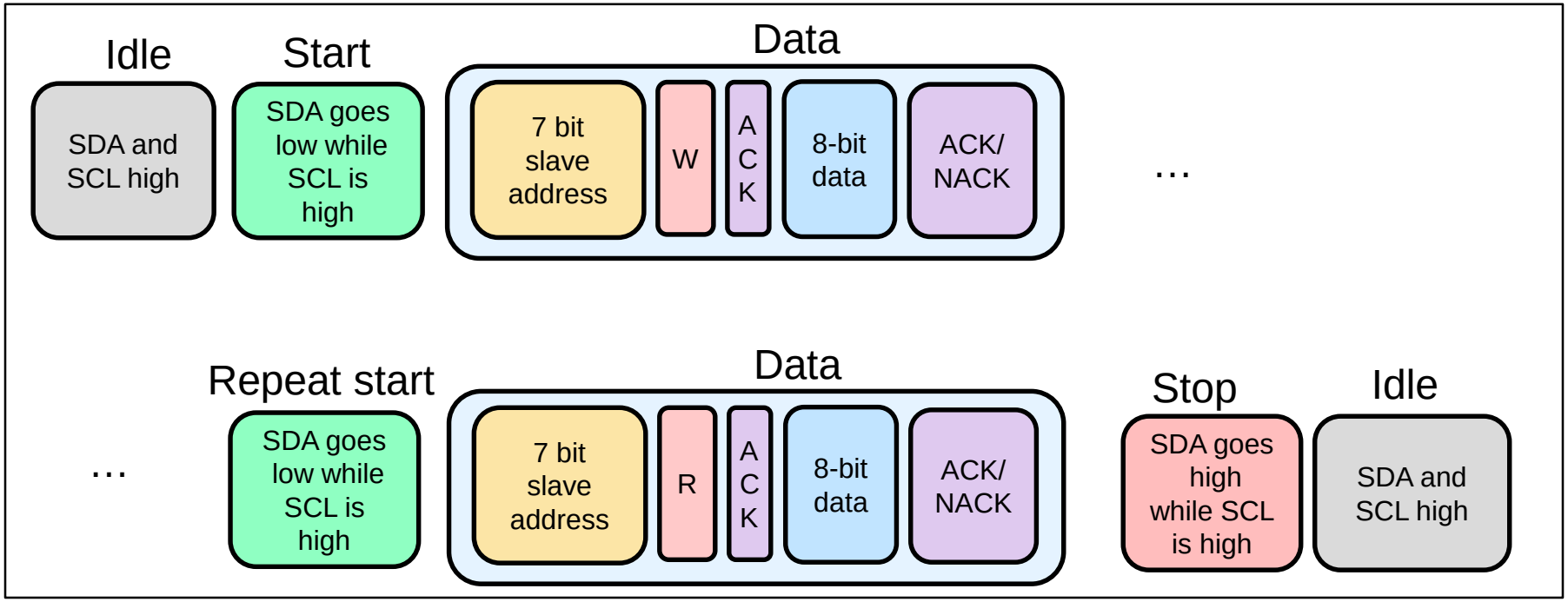
Inter-integrated Circuit (I^2C)

Message Structure: master writes and reads in the same transfer

Inter-integrated Circuit (I^2C)

Message Structure: master w/r in the same transfer

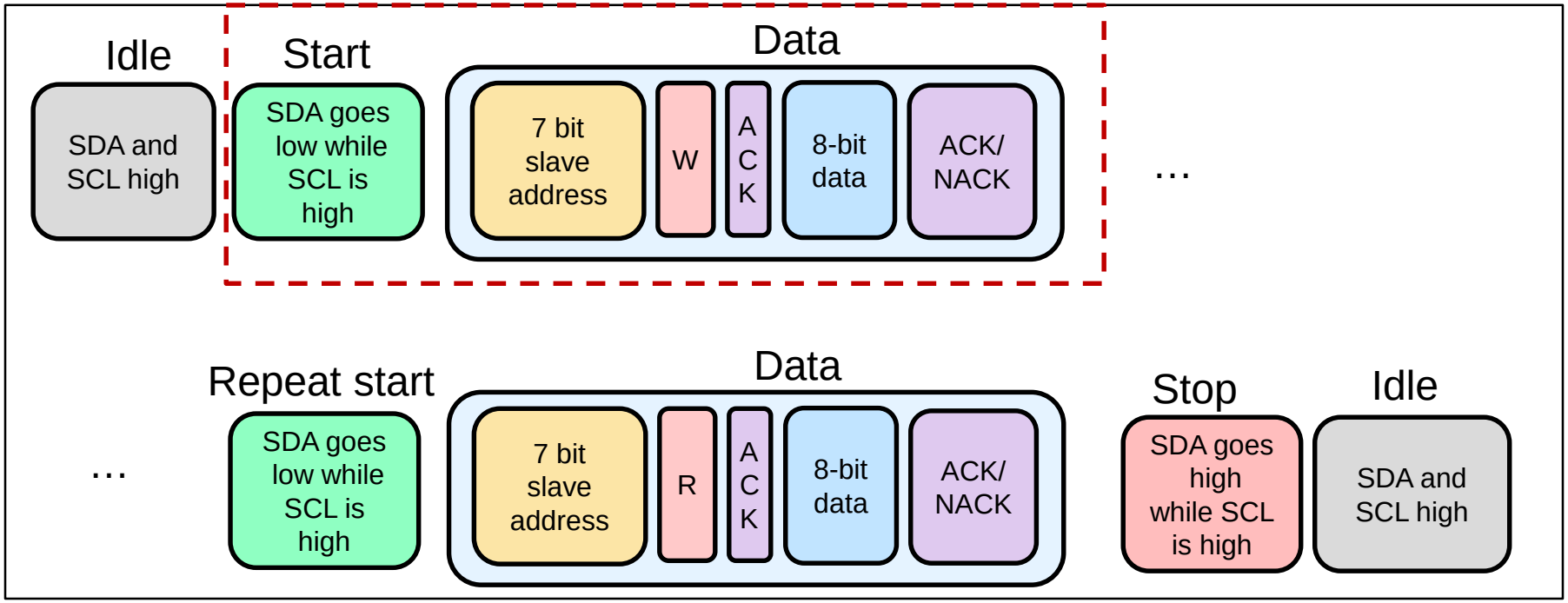
- Master writes to slave and reads from slave in one transfer:



Inter-integrated Circuit (I^2C)

Message Structure: master w/r in the same transfer

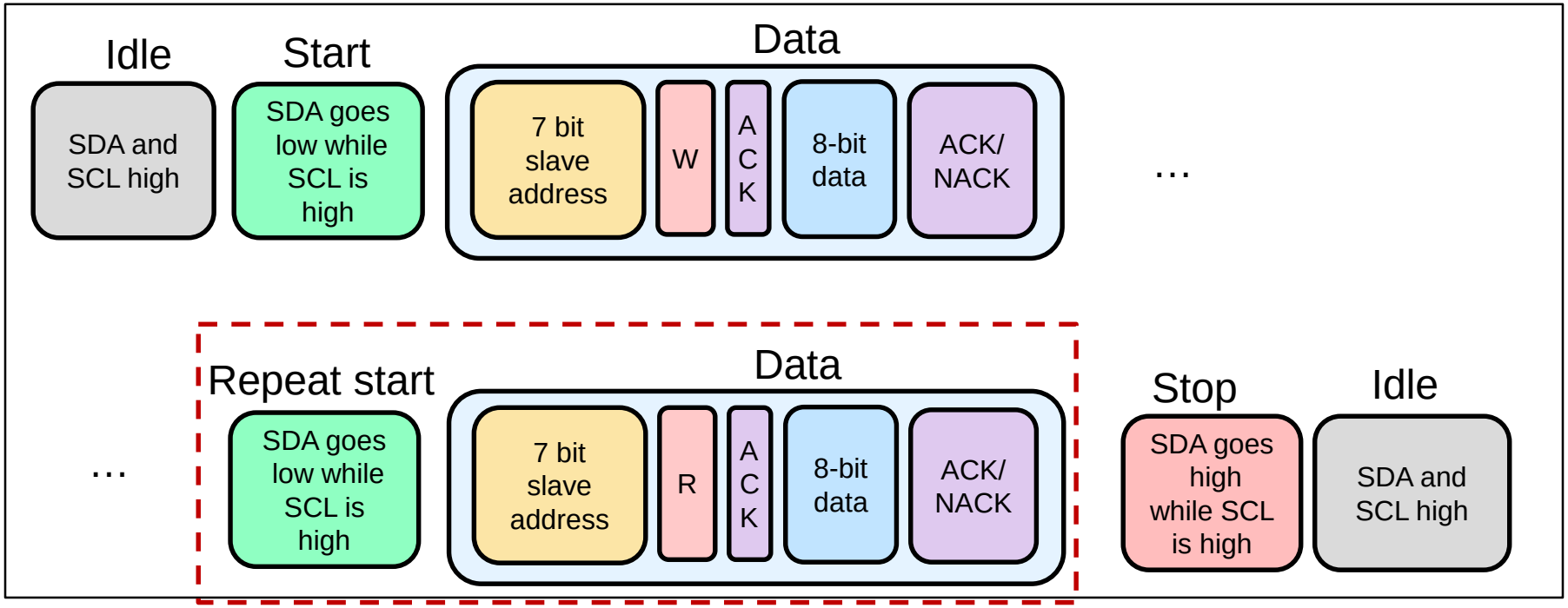
- Master writes to slave and reads from slave in one transfer
  - Write portion
  - Read portion



Inter-integrated Circuit (I^2C)

Message Structure: master w/r in the same transfer

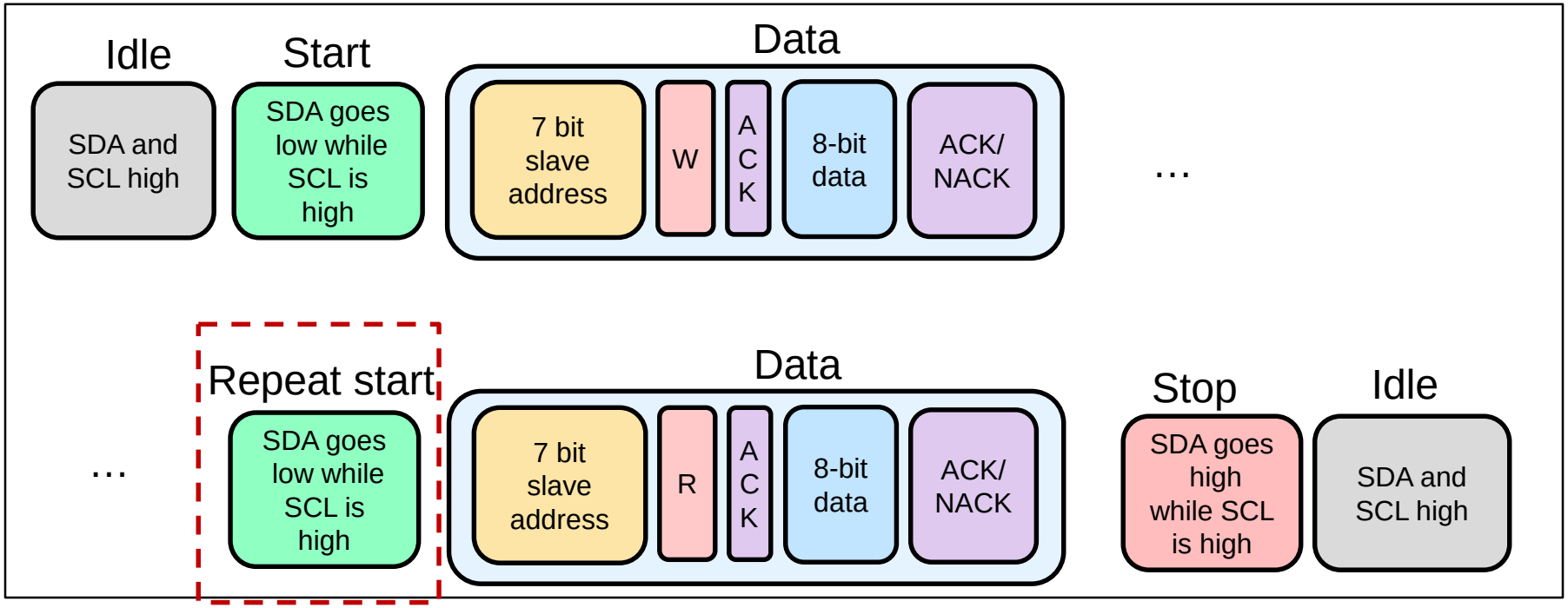
- Master writes to slave and reads from slave in one transfer
  - Write portion
  - Read portion



Inter-integrated Circuit (I^2C)

Message Structure: master w/r in the same transfer

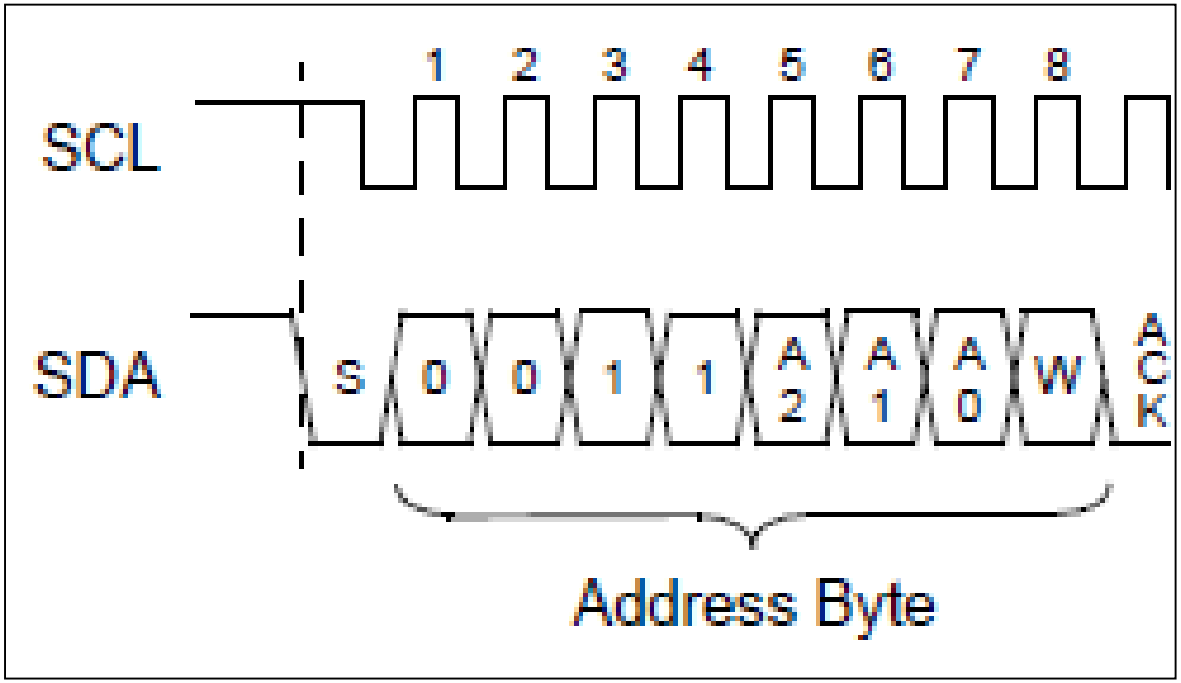
- Master writes to slave and reads from slave in one transfer
  - A repeat start command is sent when the master wants to change the direction of the data transfer



Inter-integrated Circuit (I^2C)
Timing

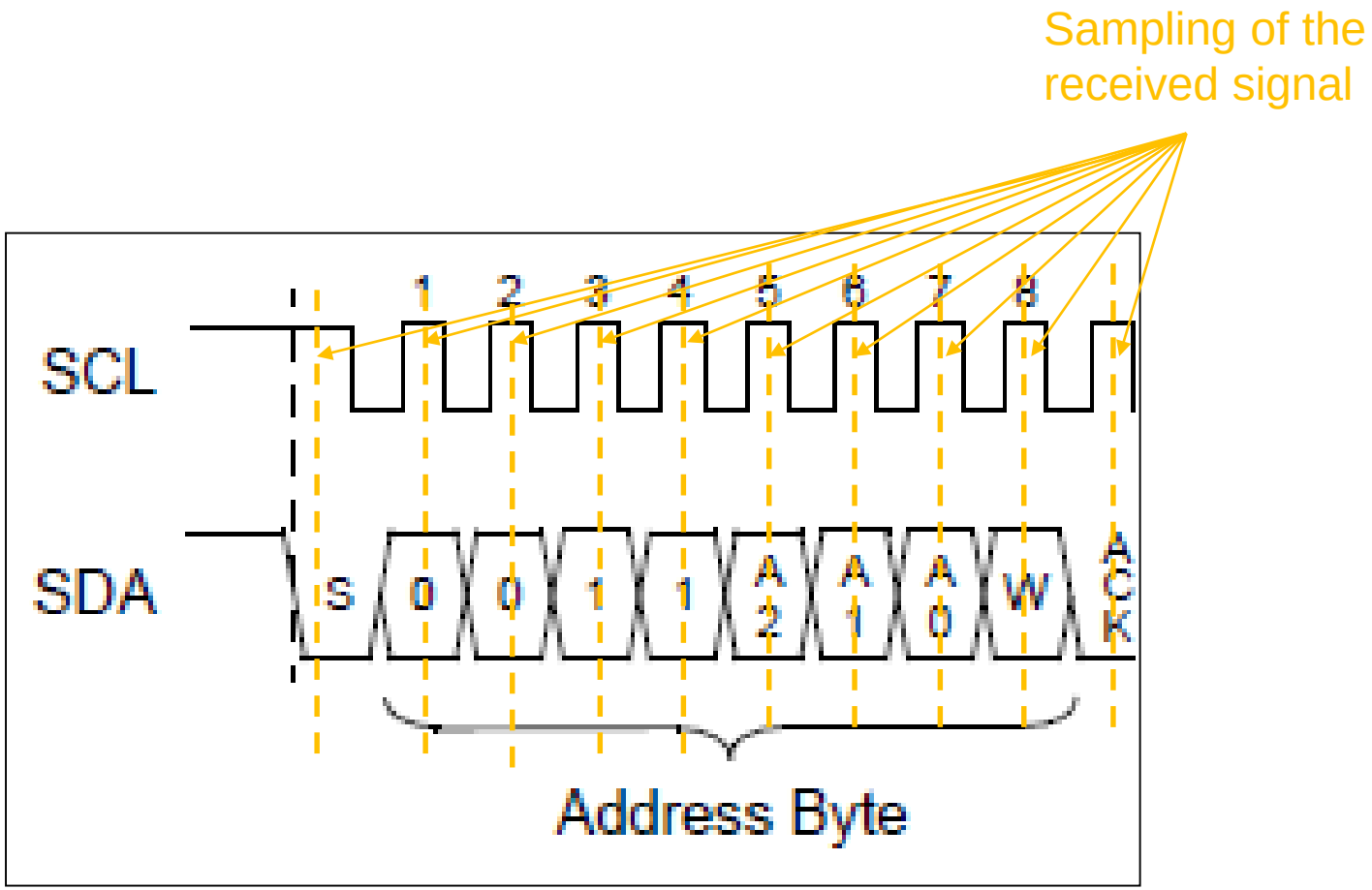
Inter-integrated Circuit (I^2C)
Timing

- When is the received signal sampled?



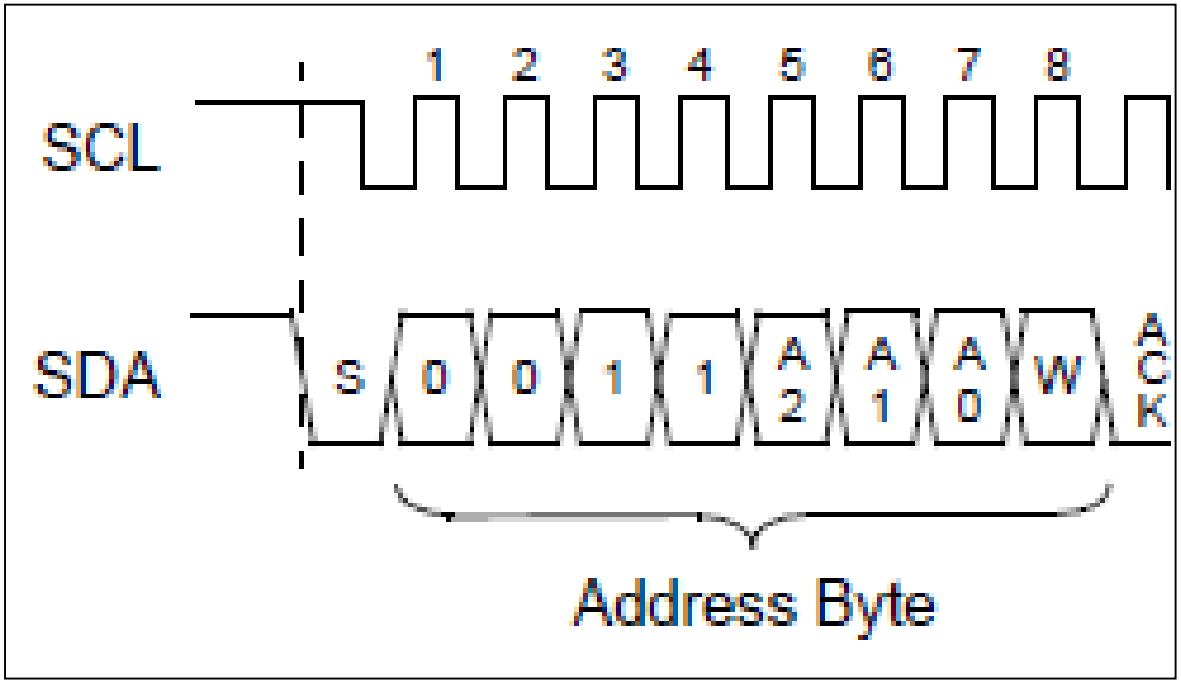
Inter-integrated Circuit (I^2C)
Timing

- When is the received signal sampled?
  - Occurs during the middle of the interval when SCL is high



Inter-integrated Circuit (I^2C)
Timing

- When is the received signal sampled?
  - Occurs during the middle of the interval when SCL is high
- When does the transmit signal change state?



Inter-integrated Circuit (I^2C)
Timing

- When is the received signal sampled?
  - Occurs during the middle of the interval when SCL is high
- When does the transmit signal change state?
  - Occurs during the middle of the interval when SCL is low