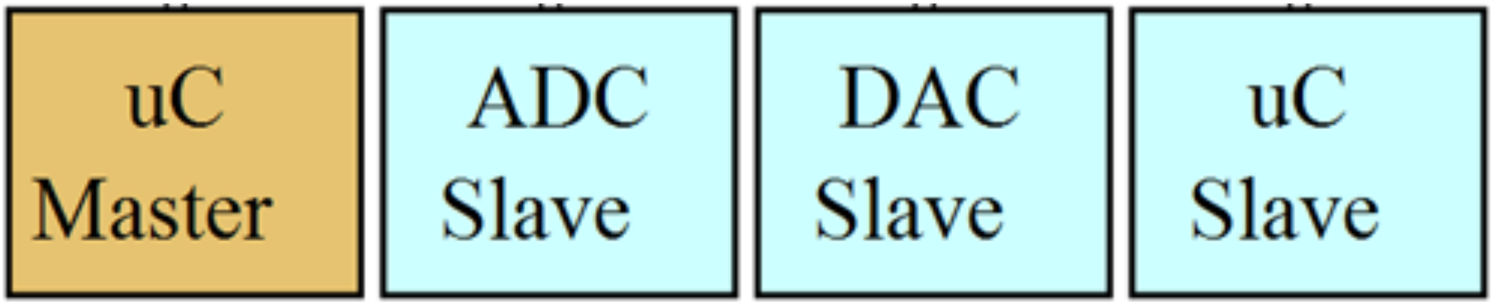
Serial communication with the I^2C bus

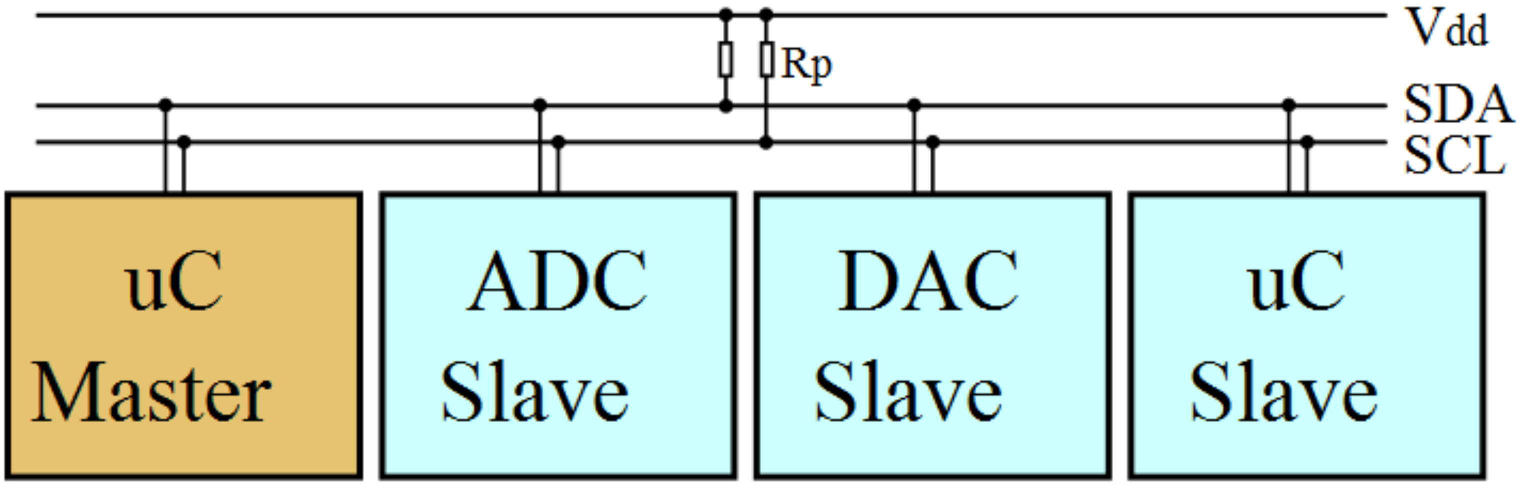
(the *Inter-Integrated Circuit* bus)

ECE 2534

Motivation: a microcontroller needs to communicate with several peripherals (maybe scattered around a circuit board)



I^{2}C interface



- SDA – serial data
- SCL – serial clock

I^{2}C interface

- **I^{2}C is a standard for 2-wire, half-duplex, synchronous, serial communication**
- Pronounced “Eye Squared See” or “Eye Two See”
- Developed by Phillips in the 1980s,
  and it became a de-facto standard in the 1990s
- “Duplex”: communication can go in 2 directions
- “Half-duplex”: communication can go in 2 directions, but only in 1 direction at any given time
- Normally used at the board level
- Often called a 2-wire serial “bus”

Example I2C interconnections between a microcontroller and an accelerometer chip

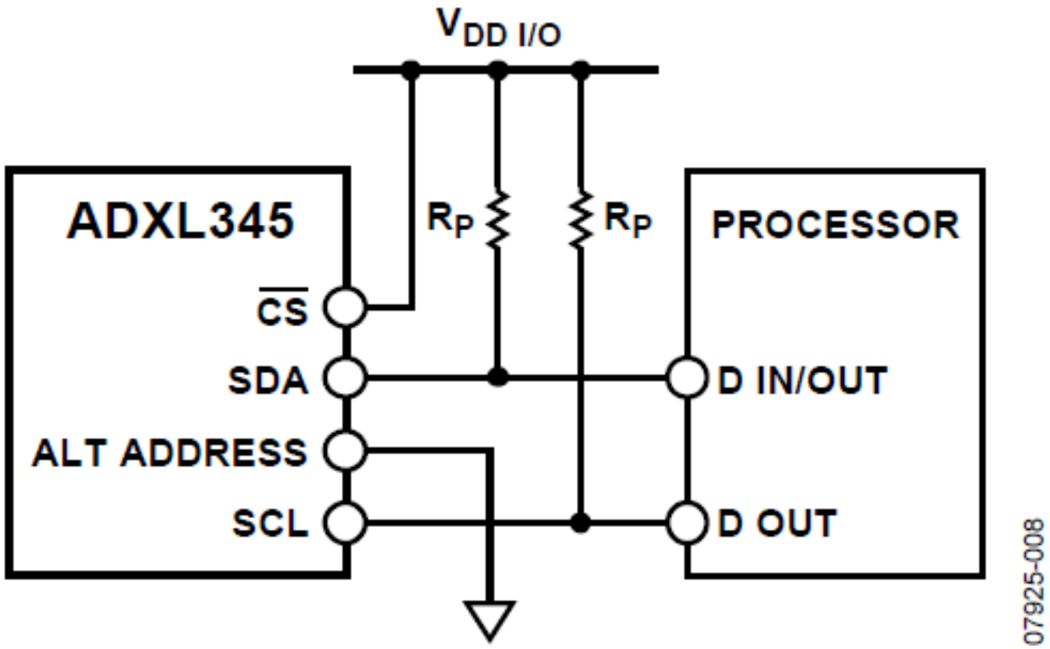


Figure 40. I^{2}C Connection Diagram (Address 0x53)

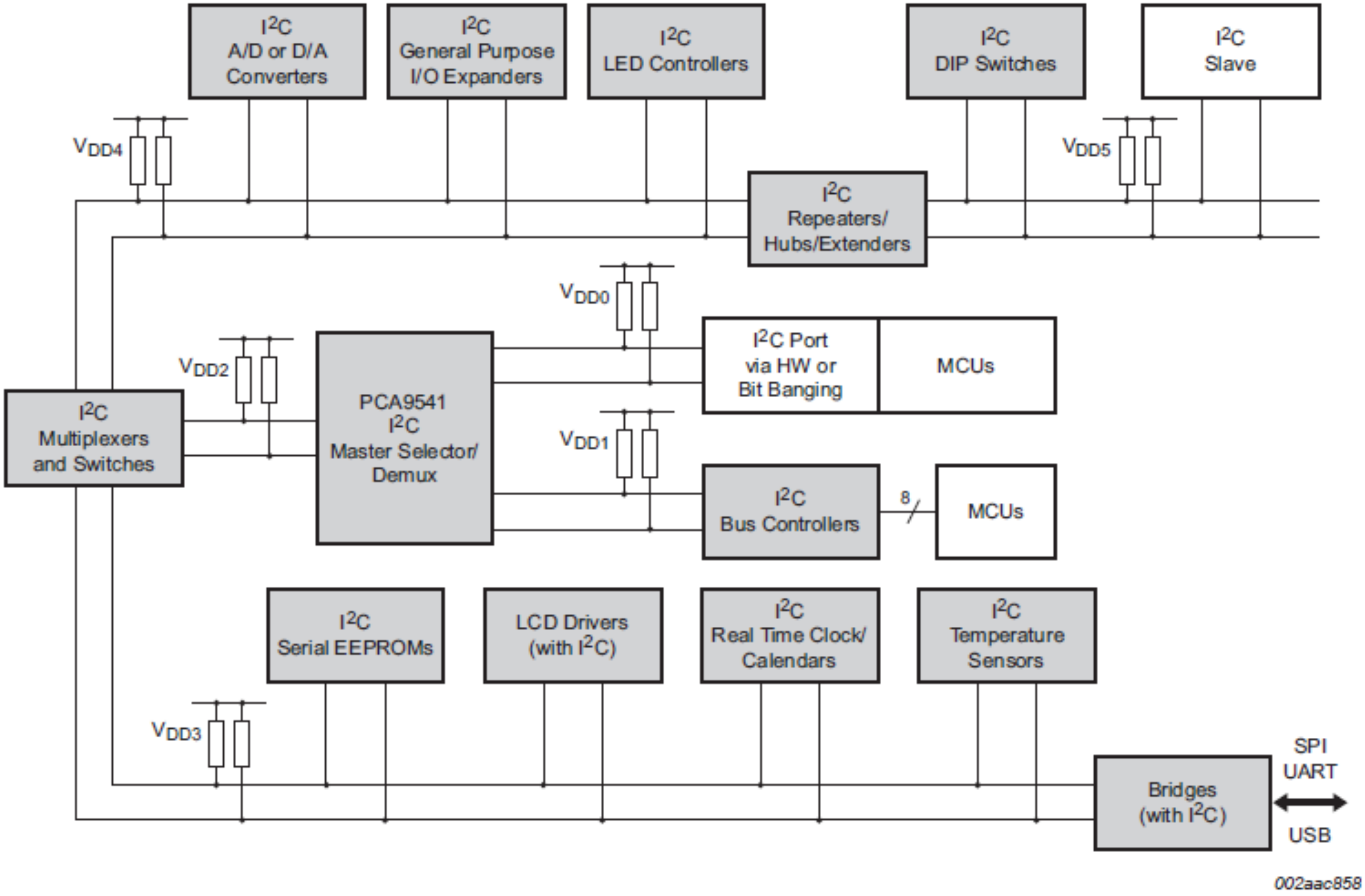
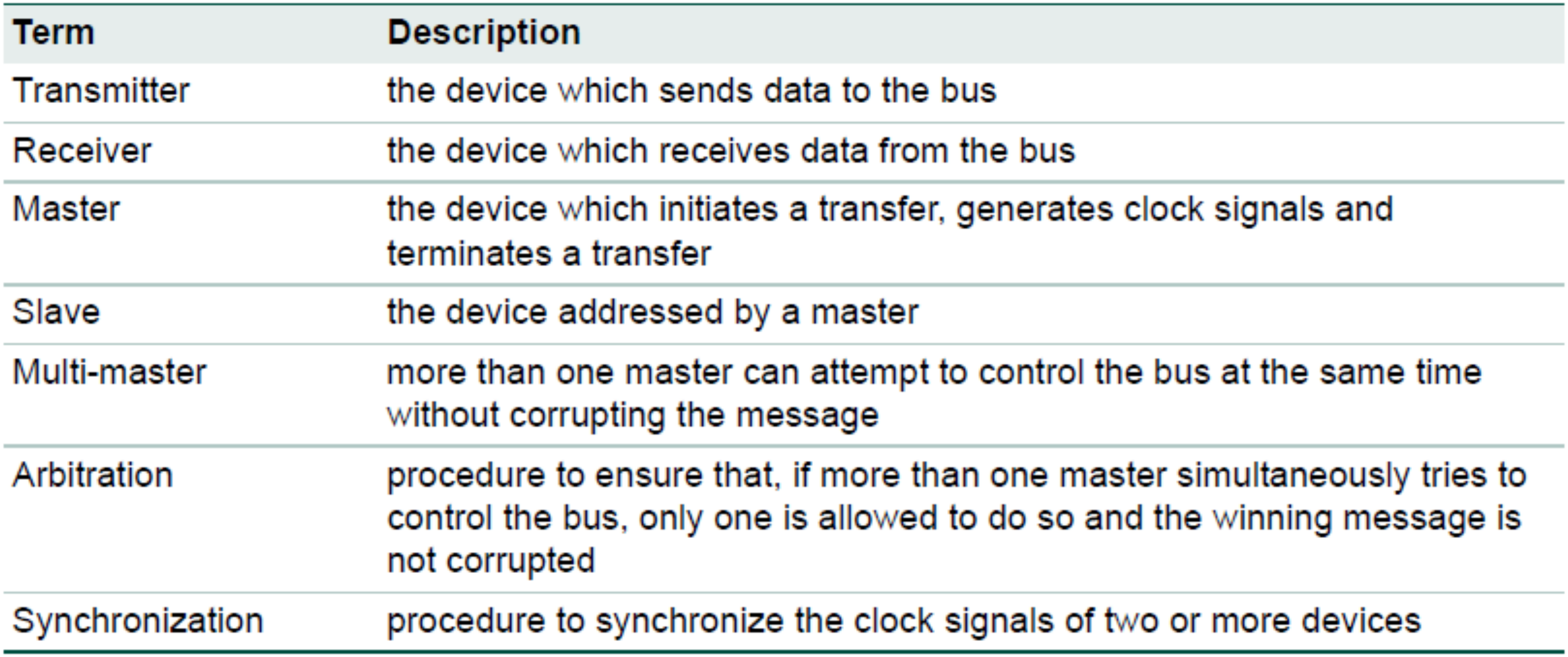


Fig 1. Example of I²C-bus applications

Some I^{2}C terminology



Term	Description
Transmitter	the device which sends data to the bus
Receiver	the device which receives data from the bus
Master	the device which initiates a transfer, generates clock signals and terminates a transfer
Slave	the device addressed by a master
Multi-master	more than one master can attempt to control the bus at the same time without corrupting the message
Arbitration	procedure to ensure that, if more than one master simultaneously tries to control the bus, only one is allowed to do so and the winning message is not corrupted
Synchronization	procedure to synchronize the clock signals of two or more devices

I^{2}C data rates

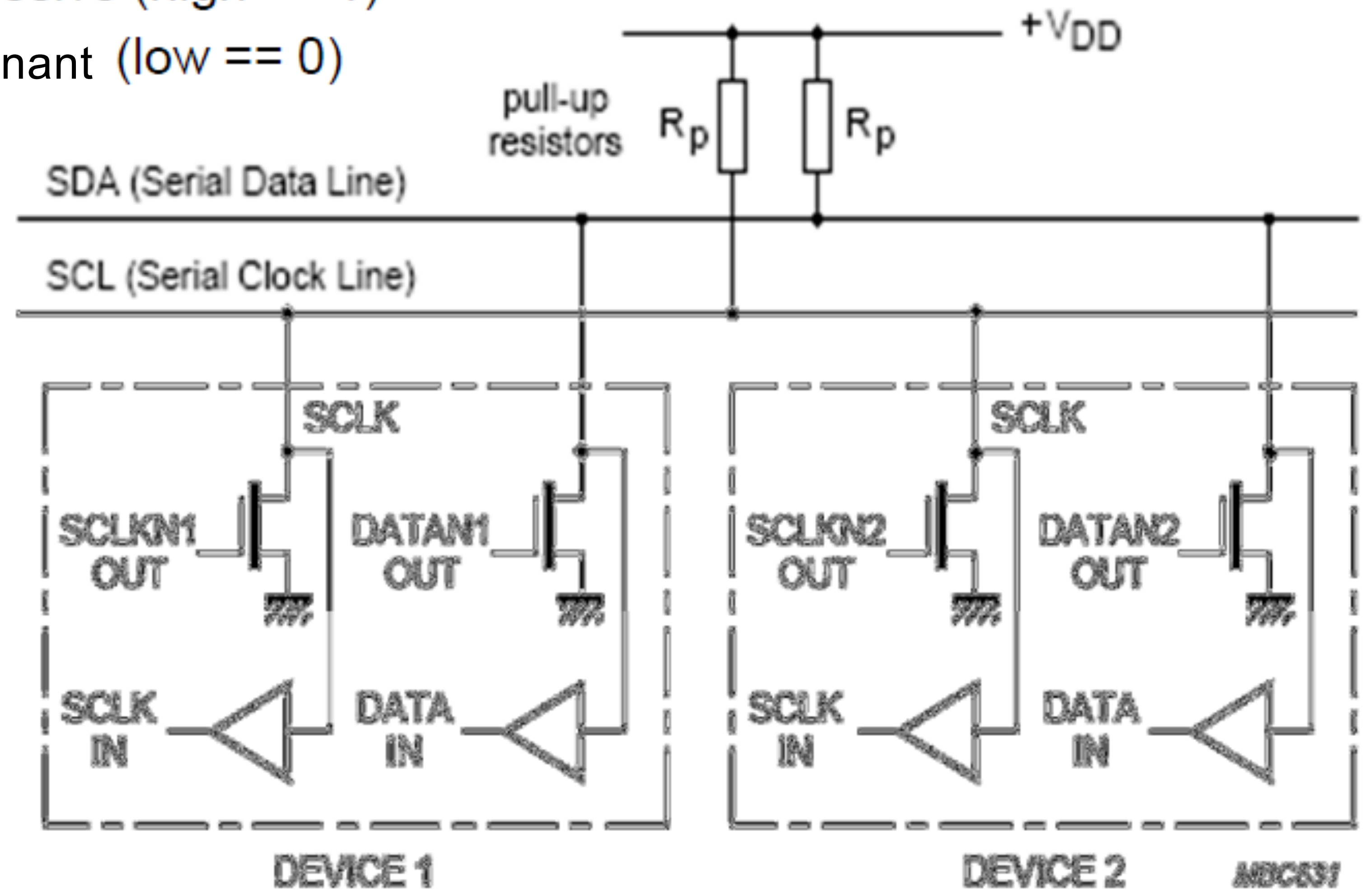
Standard	100 kbit/s
Fast	400 kbit/s
Fast+	1 Mbit/s
High speed	3.4 Mbit/s

I^{2}C Physical Layer

Bit States

Recessive (high == 1)

Dominant (low == 0)



I^2C protocol

- In the I2C interface protocol, each device has an address.
- When a master wishes to initiate a data transfer, it first transmits the I2C address of the device that it wants to “talk” to.
- All devices “listen” to see if this is their address. Within this address, bit 0 (the LSB) specifies whether the master wishes to read from or write to the slave device.
- The master and slave are always in opposite modes of operation (transmitter/receiver) during a data transfer. That is, they can be thought of as operating in either of the following two relations:
  - Master-transmitter and slave-receiver
  - Slave-transmitter and master-receiver
- In both cases, the master originates the clock signal (SCL).

I^{2}C protocol, continued

- Data transfer may be initiated only when the bus is not busy
- During data transfer, the data line must remain stable whenever the SCLx clock line is high.
- Changes in the data line while the SCLx clock line is high will be interpreted as a Start or Stop condition.

I^{2}C framing

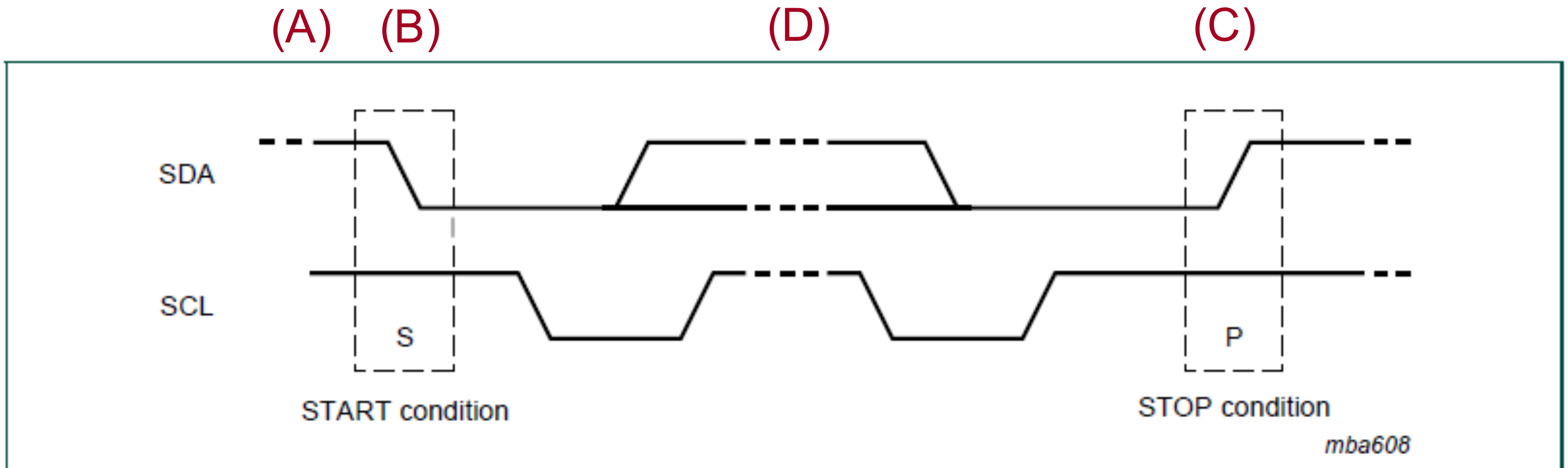


Fig 5. START and STOP conditions

(A) IDLE: SCL, SDA high

(B) START: SDA falls while SCL is high

(C) STOP: SDA rises while SCL is high

(D) SDA constant while SCL is high;
SDA changes only when SCL is low

I^2C bit transfer

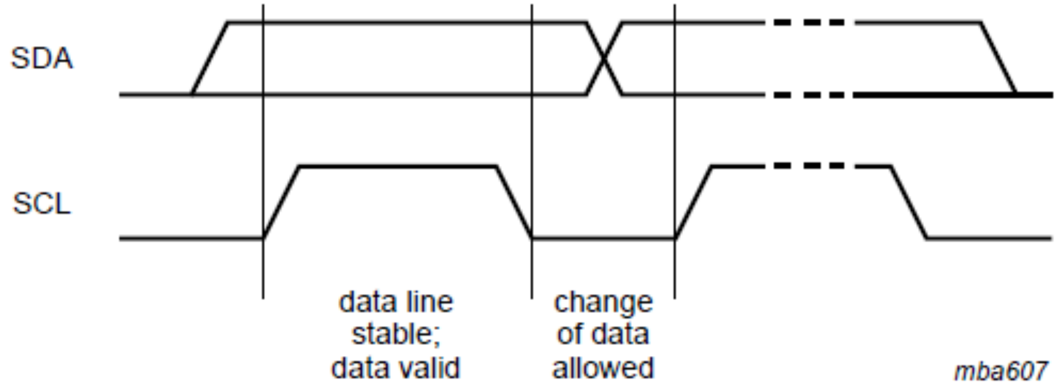


Fig 4. Bit transfer on the I^2C-bus

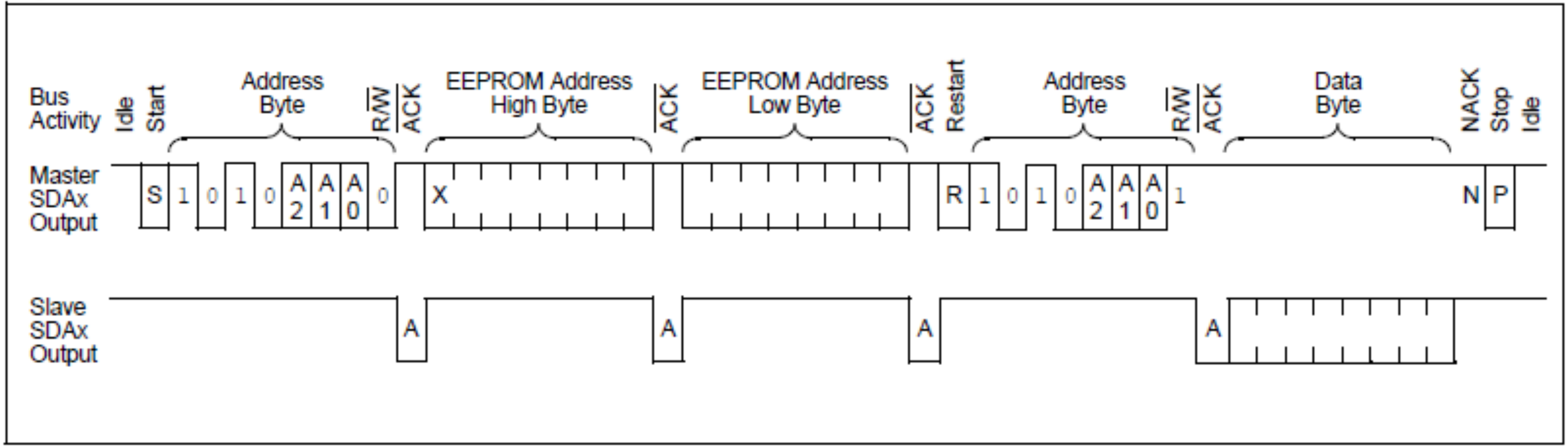
Source: **UM10204 - I2C-bus specification and user manual**

I^{2}C protocol, continued

- Data is transferred in byte-sized chunks, MSB first
- No restriction on the number of bytes in a transfer
- Between bytes, a slave can hold SCL low to force the master to wait
- “Clock stretching” – the master can hold the clock low between bits to slow things down
- For each byte, there’s a 9th clock pulse during which the receiver can indicate success (ACK = “acknowledge”) by pulling SDA low
  - Sender must relinquish SDA
  - If SDA remains high, NACK (= “Not Acknowledge”)
  - Master receiver can use NACK to signal slave transmitter that master wants no more data

I^{2}C data transfers – example interaction

Figure 24-4: A Typical I^{2}C™ Message: Read of Serial EEPROM (Random Address Mode)



Source: **61116E.pdf – PIC32 Reference Manual**

I^{2}C data transfer sequence

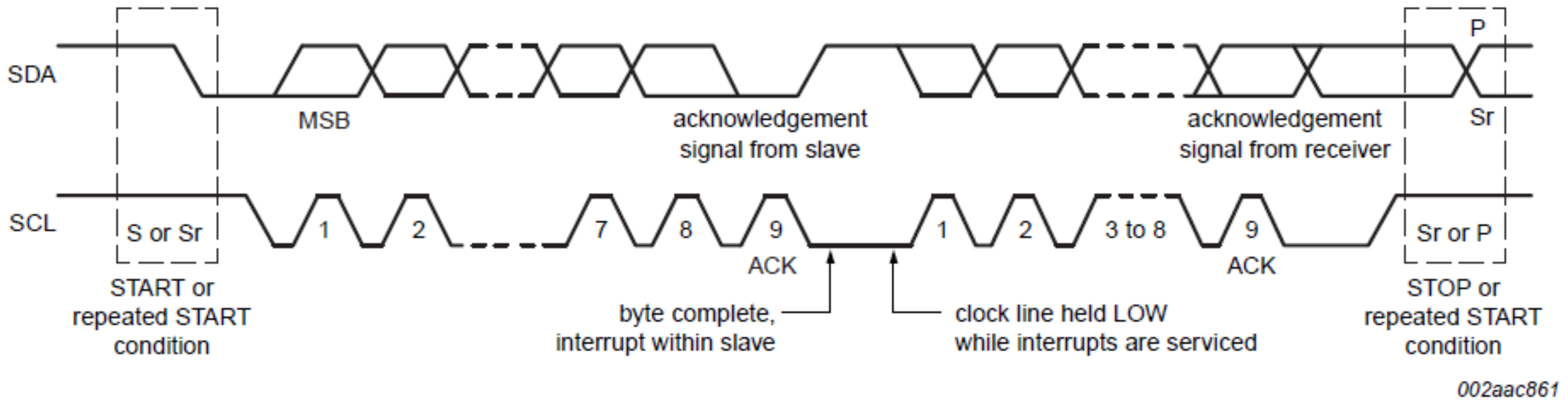

SDA
SCL
P
Sr
MSB
acknowledgement signal from slave
acknowledgement signal from receiver
S or Sr
1
2
7
8
9
ACK
1
2
3 to 8
9
ACK
Sr or P
START or repeated START condition
byte complete, interrupt within slave
clock line held LOW while interrupts are serviced
STOP or repeated START condition
002aac861

Fig 6. Data transfer on the I^{2}C-bus

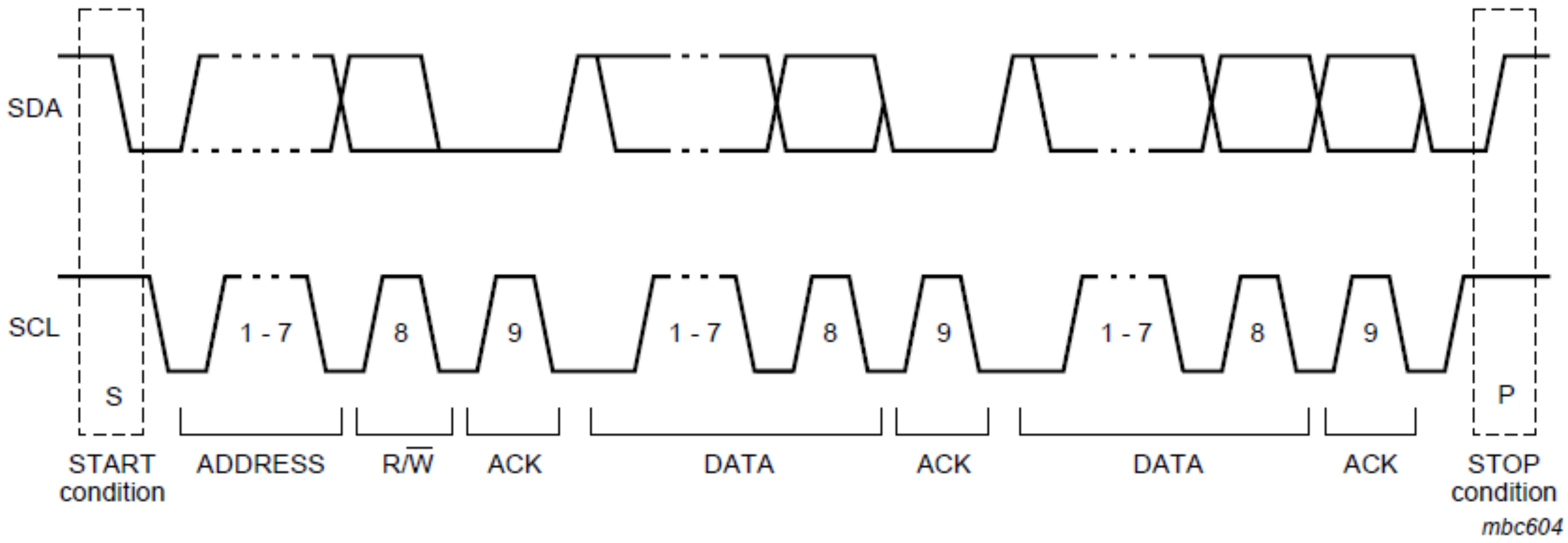


Fig 9. A complete data transfer

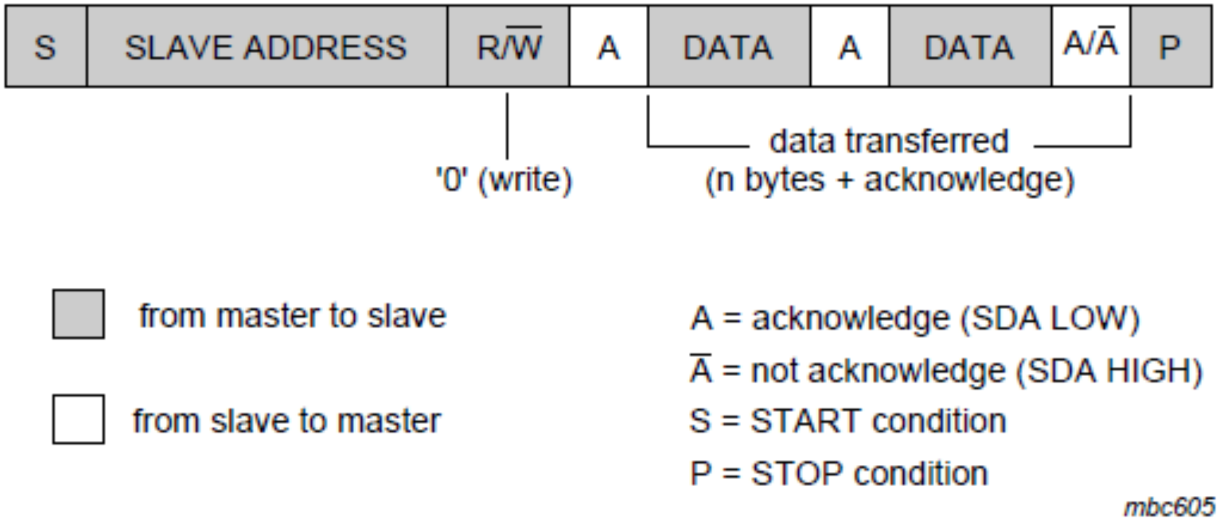


Fig 11. A master-transmitter addressing a slave receiver with a 7-bit address (the transfer direction is not changed)

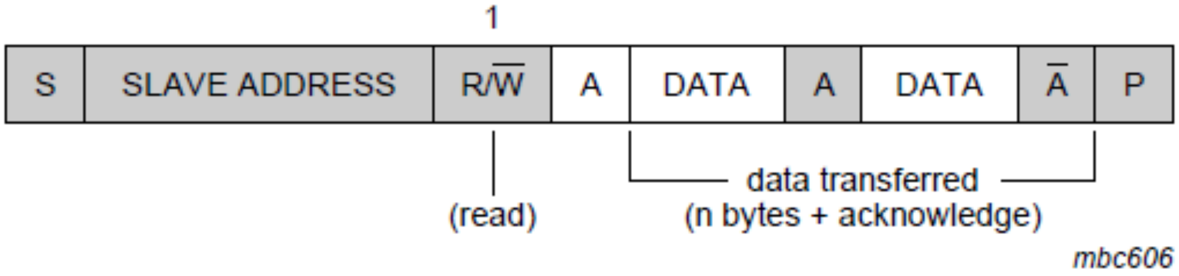
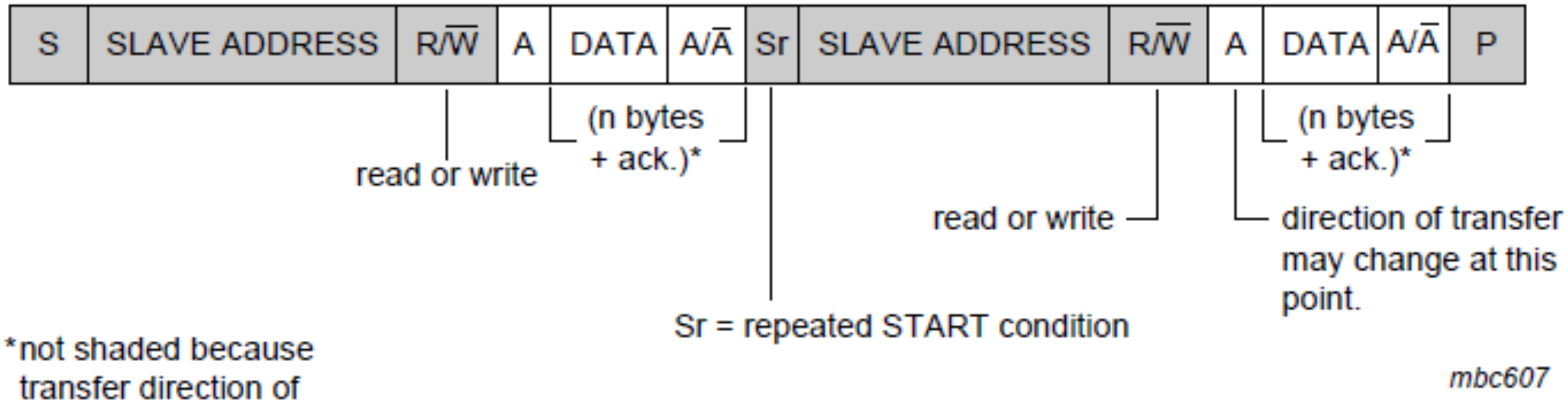


Fig 12. A master reads a slave immediately after the first byte



S | SLAVE ADDRESS | $R/\overline{W}$ | A | DATA | $A/\overline{A}$ | Sr | SLAVE ADDRESS | $R/\overline{W}$ | A | DATA | $A/\overline{A}$ | P

read or write

(n bytes + ack.)*

Sr = repeated START condition

read or write

(n bytes + ack.)*

direction of transfer may change at this point.

mbc607

*not shaded because transfer direction of data and acknowledge bits depends on $R/\overline{W}$ bits.

Fig 13. Combined format

Summary

- The **I^{2}C bus** is a popular way to connect several devices at the board level
  - half-duplex
  - synchronous
  - SDA, SCL
- A popular alternative:
  the **SPI (Serial Peripheral Interface)** bus