

I2C bus
Inter Integrated circuit bus
by Philips Semiconductor

History

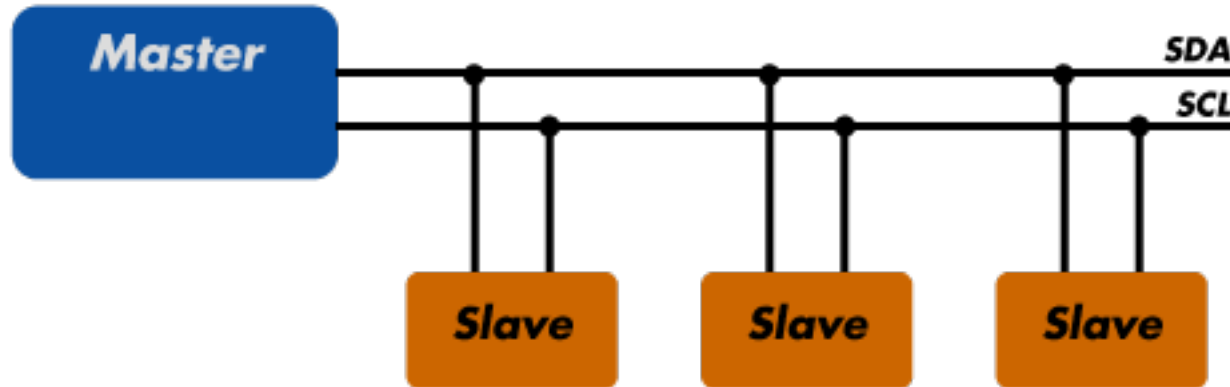
- When connecting multiple devices to a microcontroller, the address and data lines of each device were conventionally connected individually. This would take up precious pins on the microcontroller, result in a lot of traces on the PCB, and require more components to connect everything together. This made these systems expensive to produce and susceptible to interference and noise.
- To solve this problem, Philips developed Inter-IC bus, or I2C, in the 1980s. I2C is a low-bandwidth, short distance protocol for on board communications. All devices are connected through two wires: serial data (SDA) and serial clock (SCL).

Features

I2C has many important features worth mentioning

- It supports multiple data speeds:
 - standard (100 kbps)
 - fast (400 kbps)
 - high speed (3.4 Mbps) communications.
- Built in collision detection
- 7 bit or 10-bit unique Addressing
- Multi-master support (arbitration)
- Data broadcast (general call).
- two-wired bus
- data transfers: serial, 8-bit oriented, bi-directional
- master/slave relationships with multi-master option (arbitration)
- master can operate as transmitter or receiver
- device count limit: max. capacitance 400 pF

Sample I2C implementation



- Regardless of how many slave units are attached to the I2C bus, there are only two signals connected to all of them
- All devices must have a unique address to identify it on the bus.
- Slave devices have a predefined address, but the lower bits of the address can be assigned to allow for multiples of the same devices on the bus.

Theory of operation

I2C has a master/slave protocol. The master initiates the communication. The sequence of events are

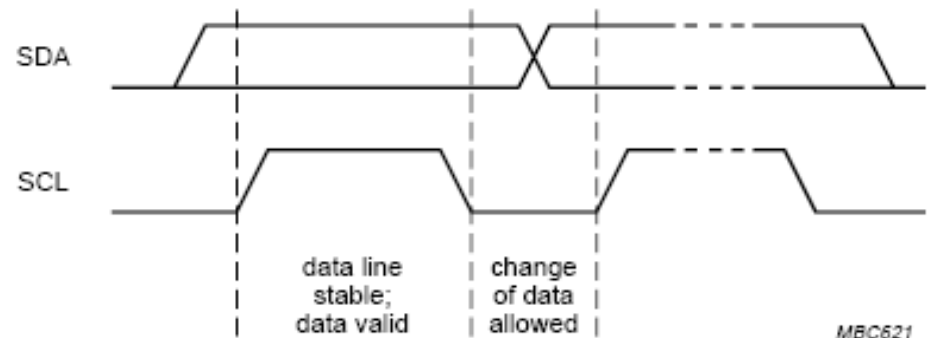
2. The Master device issues a start condition. This condition informs all the slave devices to listen on the serial data line for instructions.
3. The Master device sends the address of the target slave device and a read/write flag.
4. The Slave device with the matching address responds with an acknowledgement signal .
5. Communication proceeds between the Master and the Slave on the data bus. Both the master and slave can receive or transmit data depending on whether the communication is a read or write. The transmitter sends 8-bits of data to the receiver which replies with a 1-bit acknowledgment
6. When the communication is complete, the master issues a stop condition indicating that everything is done



Standard Mode

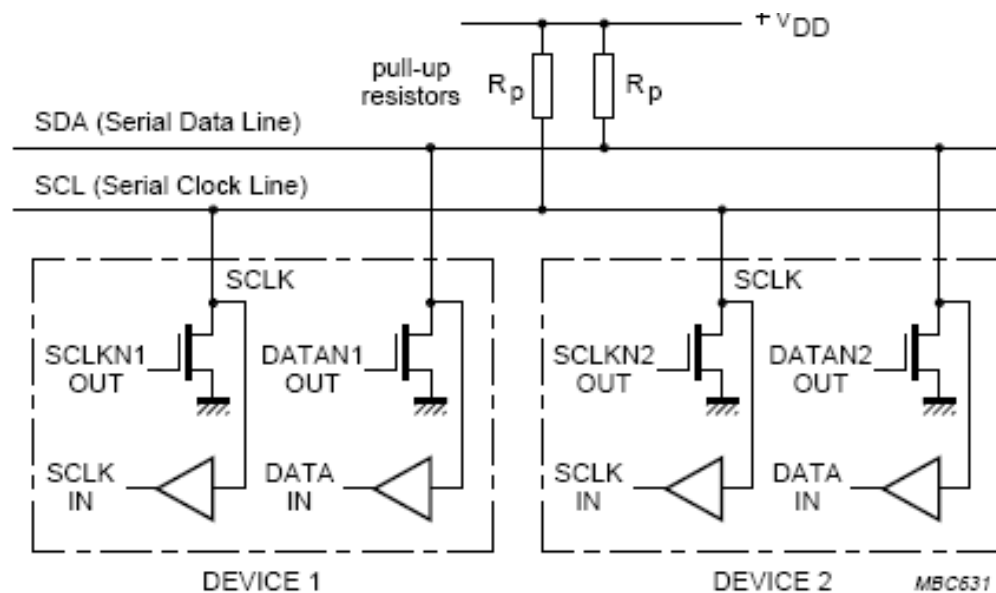
Wires And Signal

- Two-wired bus
 - serial data line (SDA)
 - serial clock line (SCL)
- Voltage levels
 - HIGH 1
 - LOW 0
 - not fixed, depends on associated level of voltage
- Bit transfer (level triggered)
 - SCL = 1 SDA = valid data
 - one clock pulse per data bit
 - stable data during high clocks
 - data change during clocks low



Wired-AND connection

- bus is free SDA and SCL are high
 - by pull-up resistors
- device output is ANDed with signal on bus



Frame

start condition (S)

– SDA 10 transition when SCL = 1

• stop condition (P)

– SDA 01 transition when SCL = 1

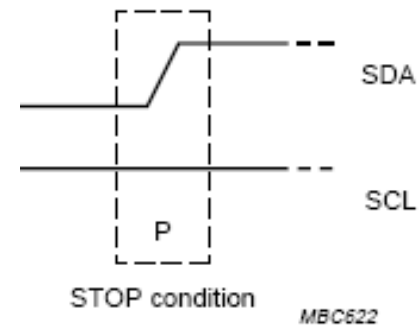
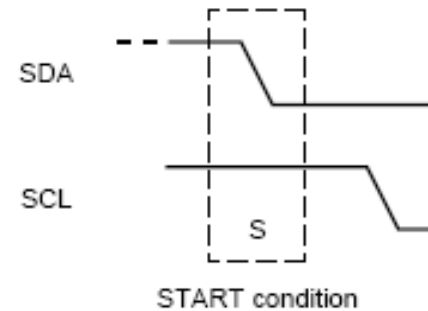
• repeated start (Sr)

– start is generated instead of stop

• bus state

– busy ... after S and before next P

– free ... after P and before next S



Masters and Slaves

- Master device
 - controls the SCL
 - starts and stops data transfer
 - controls addressing of other devices
- Slave device
 - device addressed by master
- Transmitter/Receiver
 - master or slave
 - master-transmitter sends data to slave-receiver
 - master-receiver requires data from slave-transmitter

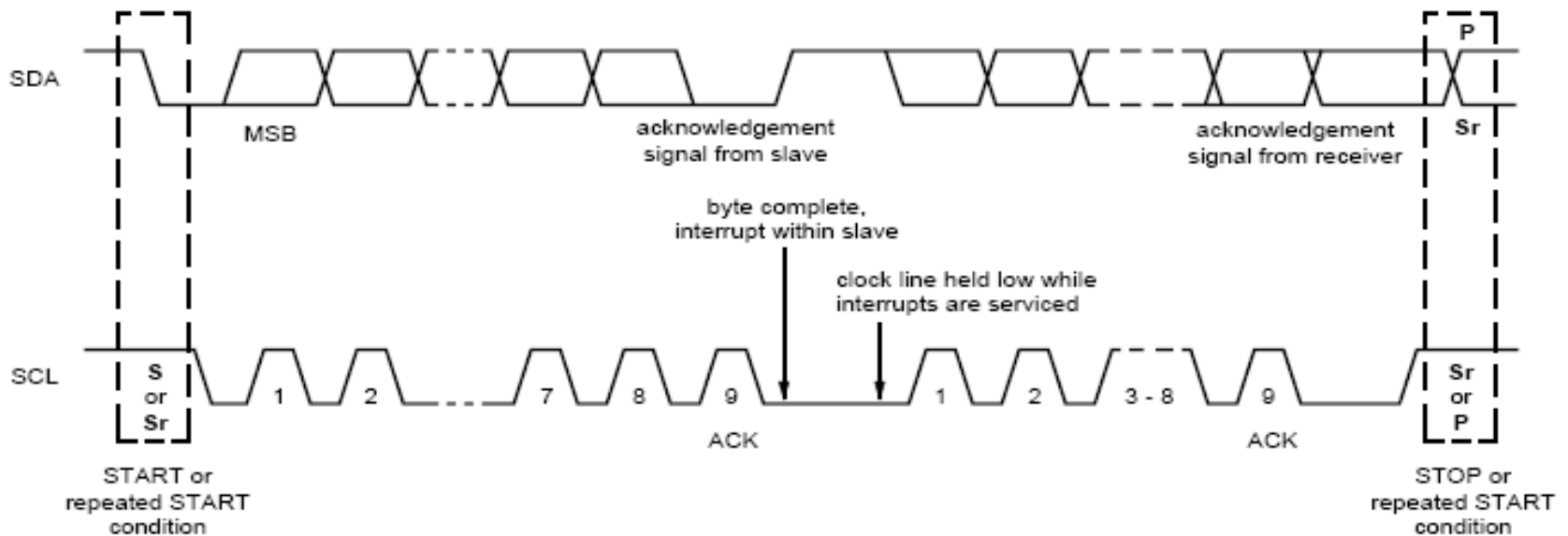
Data Transfer

data bits are transferred after start condition

- transmission is byte oriented
- byte = 8 bits + one acknowledge bit
- most significant bit (MSB) first
- slave address is also datum
 - first byte transferred
 - during the first byte transfer:
master is transmitter
addressed slave is receiver
 - next bytes: depends on the last bit in address byte

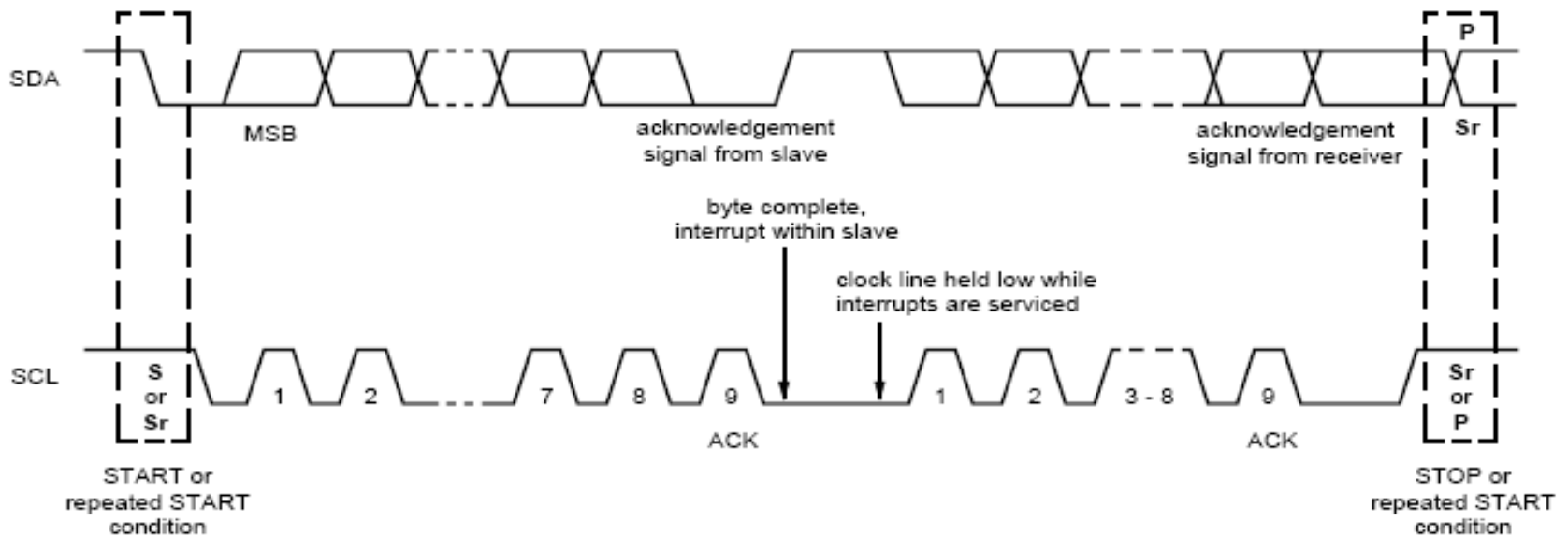
Data Transfer - SCL

- master sets $SCL = 0$ and generates pulse for each data bit
- 8 pulses for data bits are followed by one pulse for ack. bit
- after ack.
 - master tries to generate next byte's first pulse
 - slave can hold SCL low master switches to wait state



Data Transfer - SDA

- data bits are generated by transmitter as SCL pulses
- 9-th pulse:
 - transmitter releases SDA
 - receiver must hold SDA low in order to ack. received data
 - slave must release SDA after ack. bit (allows master to end frame)



Multiple Masters

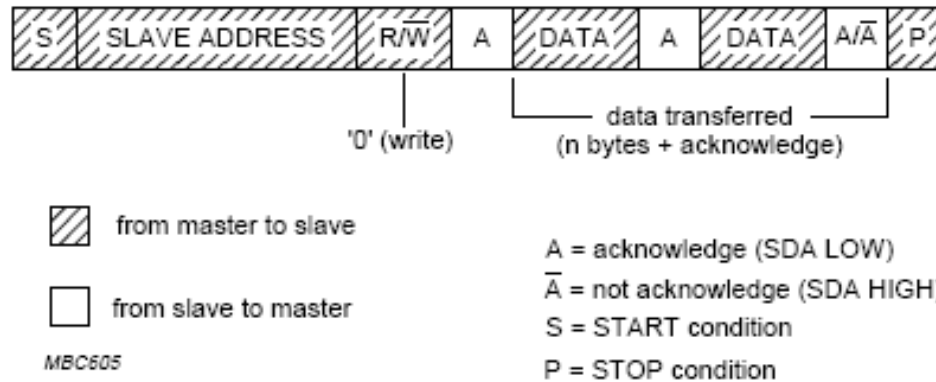
- More bus controllers can be connected
- Several masters can start frame at once
- Synchronization needed on SCL
- Arbitration needed on SDA
- Using wired-AND connection to SCL/SDA

Addressing by 7 bits

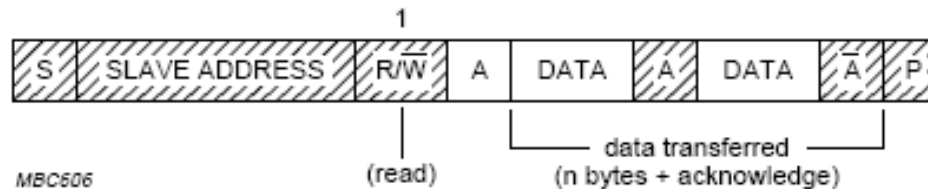
- the first byte transmitted by master:
 - 7 bits: address
 - 1 bit: direction (R/W)
 - 0 ... master writes data (W), becomes transmitter
 - 1 ... master reads data (R), becomes receiver
- data transfer terminated by stop condition
- master may generate repeated start and address another device
- each device listens to address
 - address matches its own device switches state according to R/W bit
- address = fixed part + programmable part
 - fixed part assigned by I2C committee

Frame Formats

master-transmitter



master-receiver (since second byte)



Benefits and Drawbacks

- Only two wires are required
- I2C is well suited for boards with many devices connected on the bus.
- Reduce the cost and complexity of the circuit as additional devices are added to the system.
- Due to the presence of only two wires, there is additional complexity in handling the overhead of addressing and acknowledgments .
- This can be inefficient in simple configurations and a direct-link interface such as SPI might be preferred.