# $I^2C$

# Introduction to I$^2$C

- I$^2$C is a control bus providing communications link between integrated circuits in a system.
- Developed by Philips in the early 1980s.
- Examples of I$^2$C-compatible devices found in embedded systems include EEPROMs, thermal sensors and real-time clock.

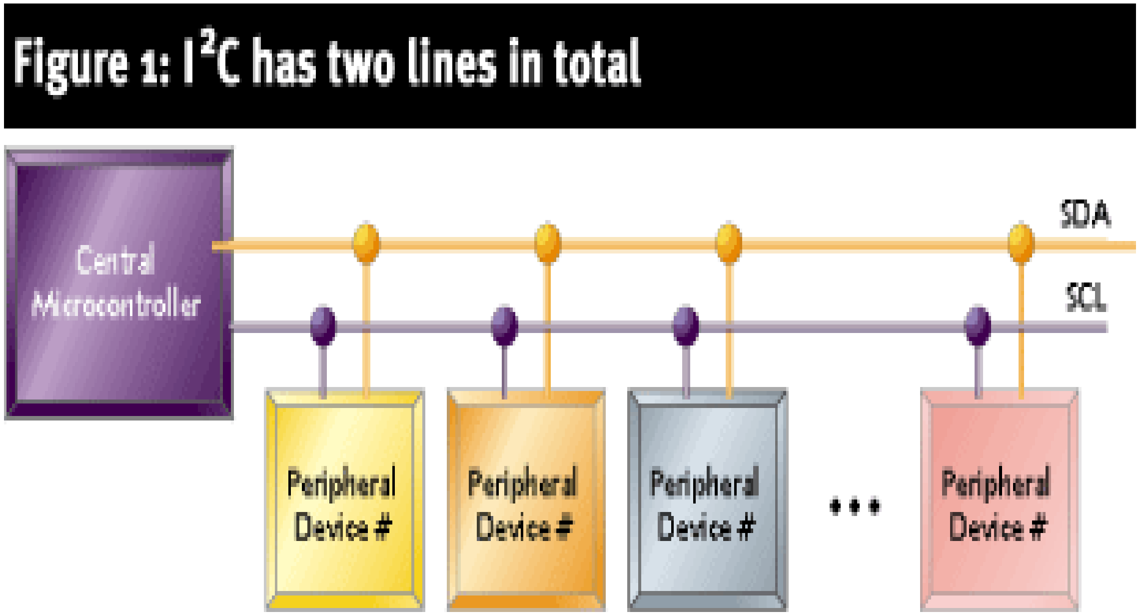
# Why I2C has endured?

- Bus has kept pace with performance and today offers 3 levels of data transfer rate
  - 100kbps in standard mode
  - 400kbps in Fast mode
  - 3.4 Mbps in high-speed mode

# Why I2C has endured?

- Reliable performance using software-controlled collision detection and arbitration.
- Ease of use. 2 lines connect all ICs in a system.
- Software controlled addressing scheme eliminating need for address-decoding hardware.

# What does I2C consist of?




I2C is a 2 wire serial bus as shown above. The 2 signals are

SDA→ Serial Data

SCL → Serial Clock

Together these signals make it possible to support serial transmission.

# I2C BUS

- The device that initiates the transaction on the I2C bus is termed the master. The master normally controls the clock signal.
- A device being addressed by the master is called the slave.
- There needs to be at least one master( a microcontroller or a DSP) on the bus, but there can be more than one master. All the masters on a bus have equal priority.

- Every device on the I2C bus has a unique 7 bit (or 10 bit) I2C address.
- Typically the 4 most significant bits are fixed and assigned to specific categories of devices. Ex: 1010 is assigned to serial EEPROM.
- The lower 3 bits are programmable allowing 8 devices of one kind to be present on a single I2C bus.

# Communication Process

**Start Condition:** The master generates a start condition to signal the beginning of communication. This is done by pulling SDA low while SCL is high.

**Address Frame:** The master sends the 7-bit address of the target slave device followed by a read/write bit.

**Data Transfer:** The master or slave can send data in 8-bit chunks. Each byte of data is followed by an acknowledgment (ACK) bit. If the receiver acknowledges, the next byte is sent; if not, the communication may end.

**Stop Condition:** The master generates a stop condition to signal the end of communication. This is done by pulling SDA high while SCL is high.

# I2C Data Transfer Example:

**Master Initiates Communication:** The master sends a start condition.

**Slave Address:** The master sends the address of the slave device (for example, 0x68).

**Read/Write:** The master sends a read/write bit.

**Data Transfer:** The master and slave exchange data byte by byte, with each byte being followed by an acknowledgment bit (ACK).

**Stop Condition:** After the communication is complete, the master sends a stop condition to release the bus.