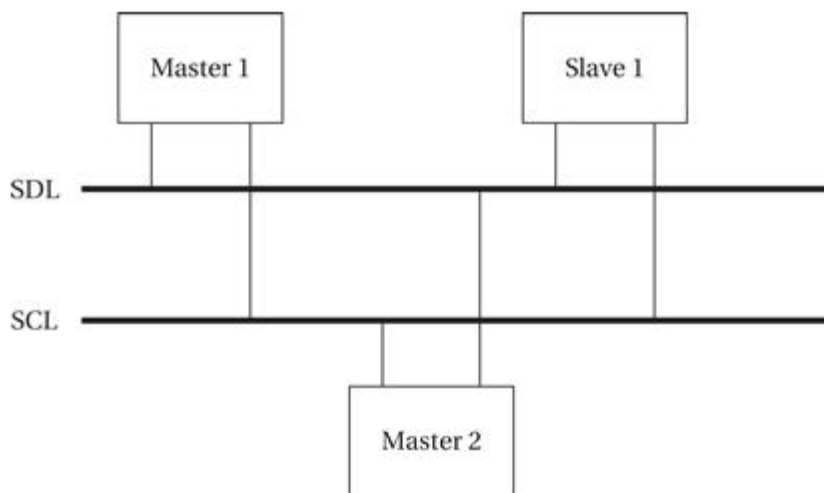


I2C :

The **I2C bus** is a well-known bus commonly used to link microcontrollers into systems. It has even been used for the command interface in an MPEG-2 video chip; while a separate bus was used for high-speed video data, setup information was transmitted to the on-chip controller through an I2C bus interface.

I2C is designed to be low cost, easy to implement, and of moderate speed (up to 100 kilobits per second for the standard bus and up to 400 kbits/sec for the extended bus). As a result, it uses only two lines: the **serial data line (SDL)** for data and the **serial clock line (SCL)**, which indicates when valid data are on the data line. Figure shows the structure of a typical I2C bus system. Every node in the network is connected to both SCL and SDL. Some nodes may be able to act as bus masters and the bus may have more than one master. Other nodes may act as slaves that only respond to requests from masters.



The I2C bus is designed as a multimaster bus—any one of several different devices may act as the master at various times. As a result, there is no global master to generate the clock signal on SCL. Instead, a master drives both SCL and SDL when it is sending data. When the bus is idle, both SCL and SDL remain high. When two devices try to drive either SCL or SDL to different values, the open collector/open drain circuitry prevents errors, but each master device must listen to the bus while transmitting to be sure that it is not interfering with another message—if the device receives a different value than it is trying to transmit, then it knows that it is interfering with another message.

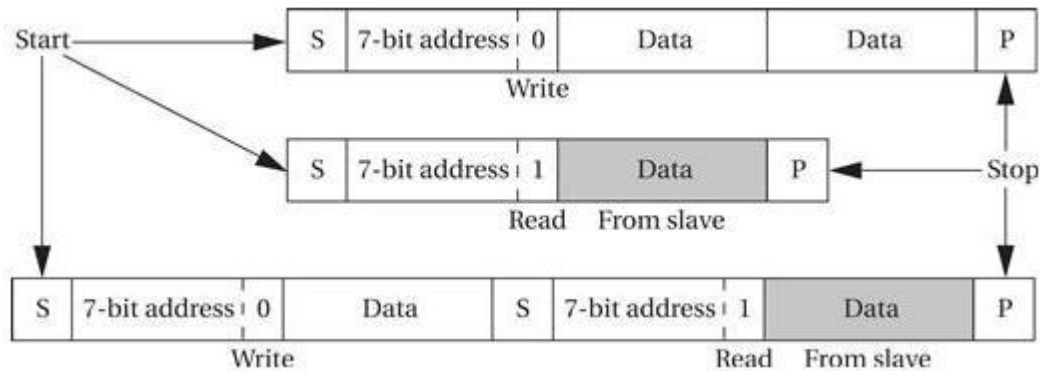
Every I2C device has an address. The addresses of the devices are determined by the system designer, usually as part of the program for the I2C driver. The addresses must of course be chosen so that no two devices in the system have the same address. A device address is 7 bits in the standard I2C definition (the extended I2C allows 10-bit addresses). The address 0000000 is used to signal a general call or bus broadcast, which can be used to signal all devices simultaneously. The address 11110XX is reserved for the extended 10-bit addressing scheme; there are several other reserved addresses as well.

A bus transaction is comprised of a series of one-byte transmissions and an address followed by one or more data bytes. I2C encourages a data-push programming style. When a master wants to write a slave, it transmits the slave's address followed by the data. Because a slave cannot initiate a transfer, the master must send a read request with the slave's address and let the slave transmit the data. Therefore, an address transmission includes the 7-bit address and 1

bit for data direction: 0 for writing from the master to the slave and 1 for reading from the slave to the master.

A bus transaction is initiated by a start signal and completed with an end signal:

- A start is signaled by leaving the SCL high and sending a 1 to 0 transition on SDL.
- A stop is signaled by setting the SCL high and sending a 0 to 1 transition on SDL.



The bus uses this feature to arbitrate on each message. When sending, devices listen to the bus as well. If a device is trying to send a logic 1 but hears a logic 0, it immediately stops transmitting and gives the other sender priority. (The devices should be designed so that they can stop transmitting in time to allow a valid bit to be sent.) In many cases, arbitration will be completed during the address portion of a transmission, but arbitration may continue into the data portion. If two devices are trying to send identical data to the same address, then of course they never interfere and both succeed in sending their message.

The I2C interface on a microcontroller can be implemented with varying percentages of the functionality in software and hardware. As illustrated in Figure, a typical system

has a 1-bit hardware interface with routines for byte-level functions. The I2C device takes care of generating the clock and data. The application code calls routines to send an address, send a data byte, and so on, which then generates the SCL and SDL, acknowledges, and so forth. One of the microcontroller's timers is typically used to control the length of bits on the bus. Interrupts may be used to recognize bits. However, when used in master mode, polled I/O may be acceptable if no other pending tasks can be performed, because masters initiate their own transfers.

