Embedded Systems

Ch 6B.
Serial Interface

Part B

Byung Kook Kim
Dept of EECS
Korea Advanced Institute of Science and Technology

Overview

- 1. Introduction to Serial Ports
- 2. RS-232C
- 3. Serial Hardware
- 4. UARTs in Xscale
- 5. Serial Peripheral Interface (SPI)
- 6. Inter Integrated Circuit (I^2C)
- 7. RS-422
- 8. Universal Serial Bus (USB)

5. Serial Peripheral Interface (SPI)

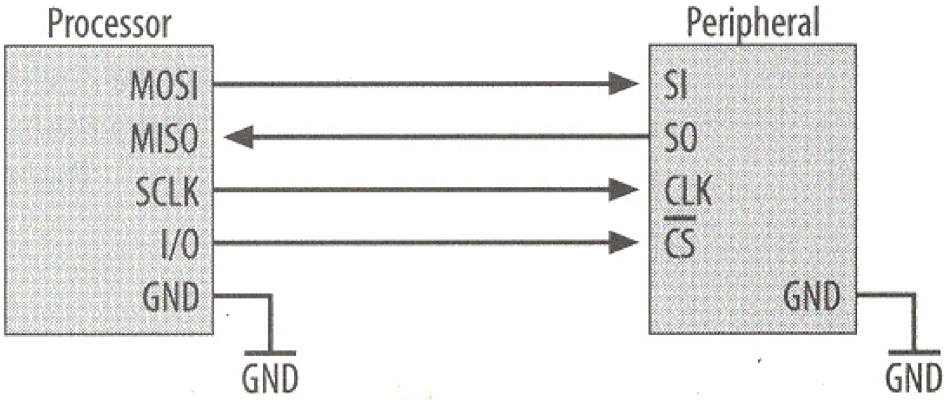
- **Serial Peripheral Interface**
  - Developed by Motorola
  - To provide a low-cost and simple interface between microcontrollers and peripheral chips
  - Four-wire interface
  - Can be used to interface to memory, A/D converters, D/A converters, real-time clock calendars, LCD drivers, sensors, audio chips, and even other processors
  - The range of components that support SPI is large and growing all the time.

SPI (II)

- SPI Interface
  - Synchronous protocol in which all transmissions are referenced to a common clock, generated by the master processor
  - The receiving peripheral (slave) uses the clock to synchronize its acquisition of the serial bit stream.
  - Many chips may be connected to the same SPI interface of a master.
    - A master selects a slave to receive by asserting the slave's chip select input.

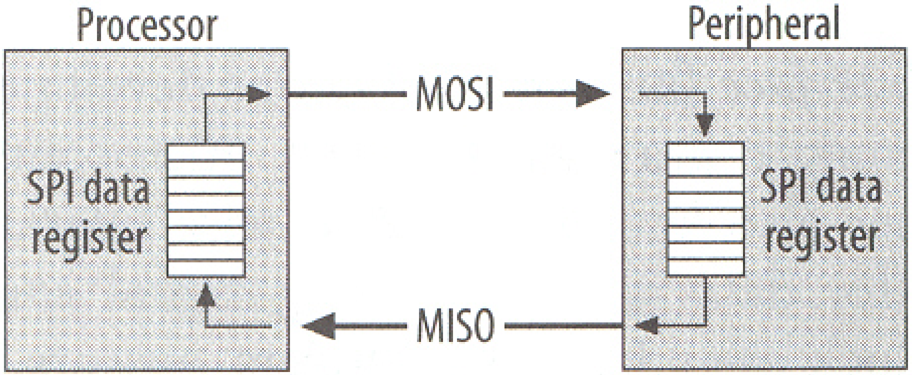
SPI (III)

- SPI signals
  - MOSI (Master Out Slave In)
    - Generated by the master
    - Labeled as SI (Serial In) or SDI (Serial Data In)
  - MISO (Master In Slave Out)
    - Generated by the slave
    - Labeled as SO (Serial Output) or SDO (Serial Data Out)
  - SCLK (Serial CLocK)
  - CSb (Chip Select)
    - Normally generated using a spare I/O pin of the master
  - SSb (Slave Select)
- Basic SPI interface ->



SPI (IV)

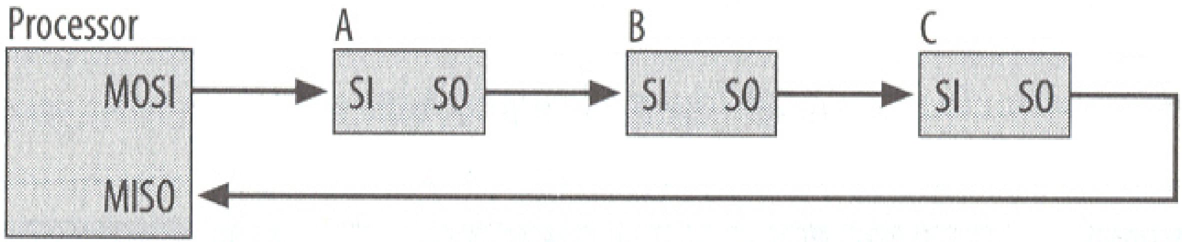
- SPI transmission
  - Both masters and slaves contain a serial shift register.
  - The master starts a transfer of a byte by writing it to its SPI shift register.
  - As the register transmits the byte to the slave on the MOSI line, the slave transfers the contents of its shift register back to the master on the MISO signal line ->



  - The contents of the two shift registers are exchanged.
    - Both a write and a read operation are performed with the slave simultaneously.
  - SPI can therefore be a very efficient protocol.

SPI (V)

- Remarks on SPI
  - Write only
    - The master ignores the byte it received.
  - Read only
    - The master transfers a dummy byte.
  - Some peripherals can handle multiple byte transfers, with a continuous stream of data shifted from the master.
    - CS for the SPI slave must remain low for the entire duration of transmission.
    - Ex: Memory chip interface
      - Write command, four address bytes (starting address), and data bytes to be stored.
  - Daisy chaining may be supported ->



SPI (VI)

- SPI mode of operation (I)

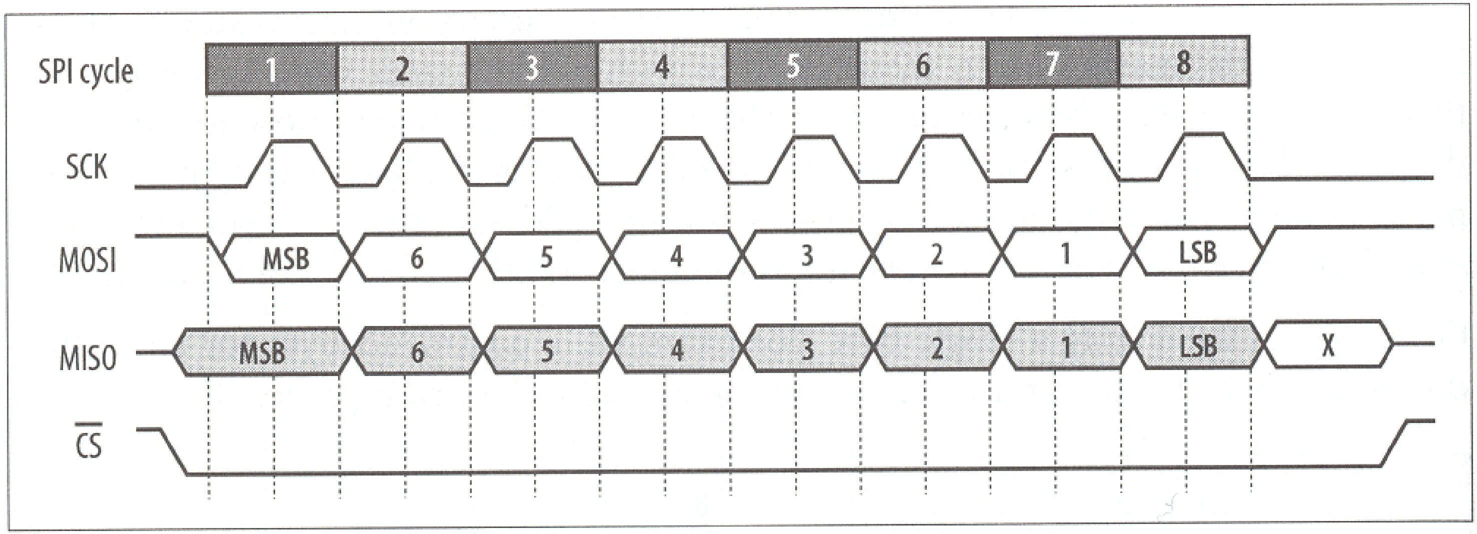


Figure 9-4. SPI timing with clock polarity low and clock phase zero

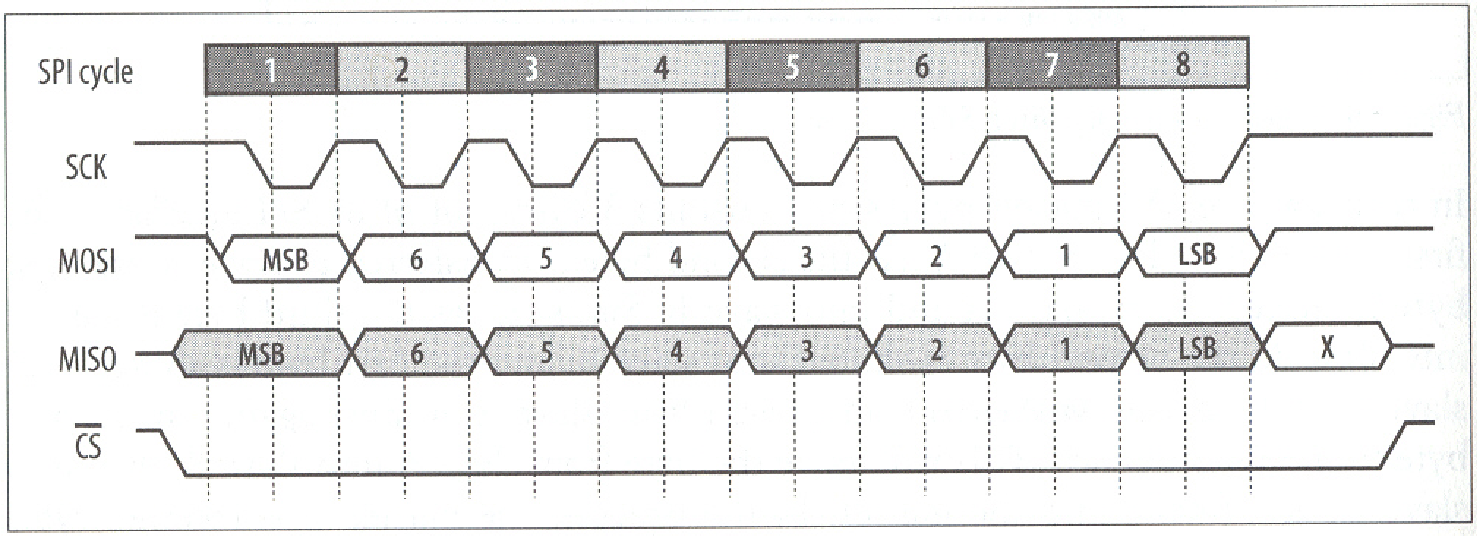


Figure 9-5. SPI timing with clock polarity high and clock phase zero

SPI (VII)

- SPI mode of operation (II)

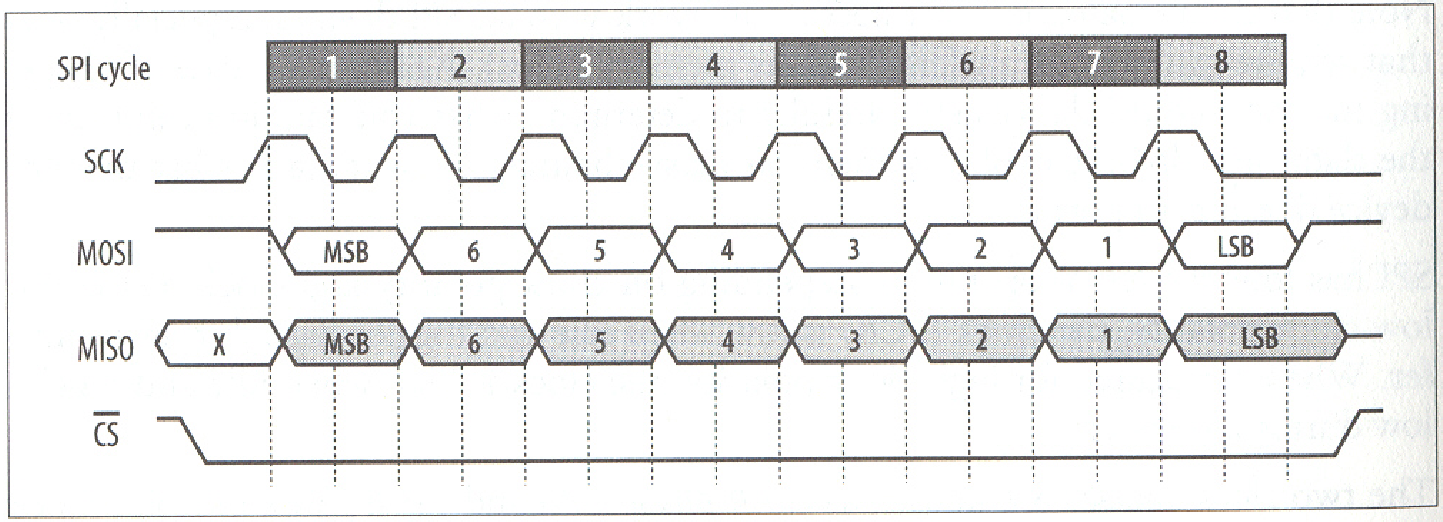


Figure 9-6. SPI timing with clock polarity low and clock phase one

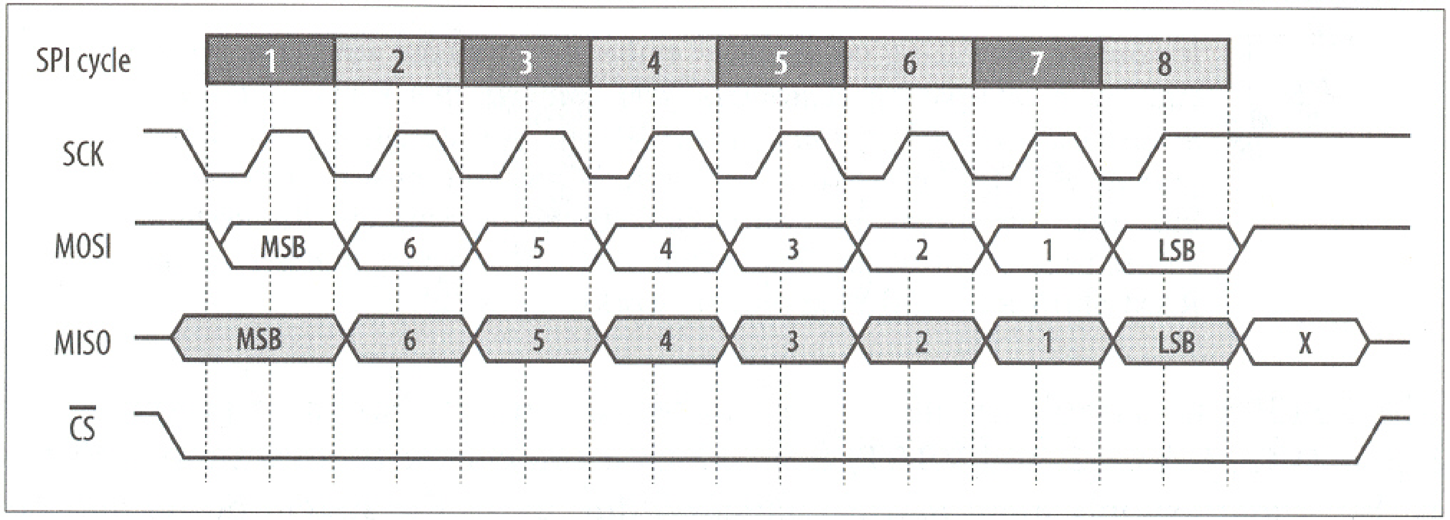
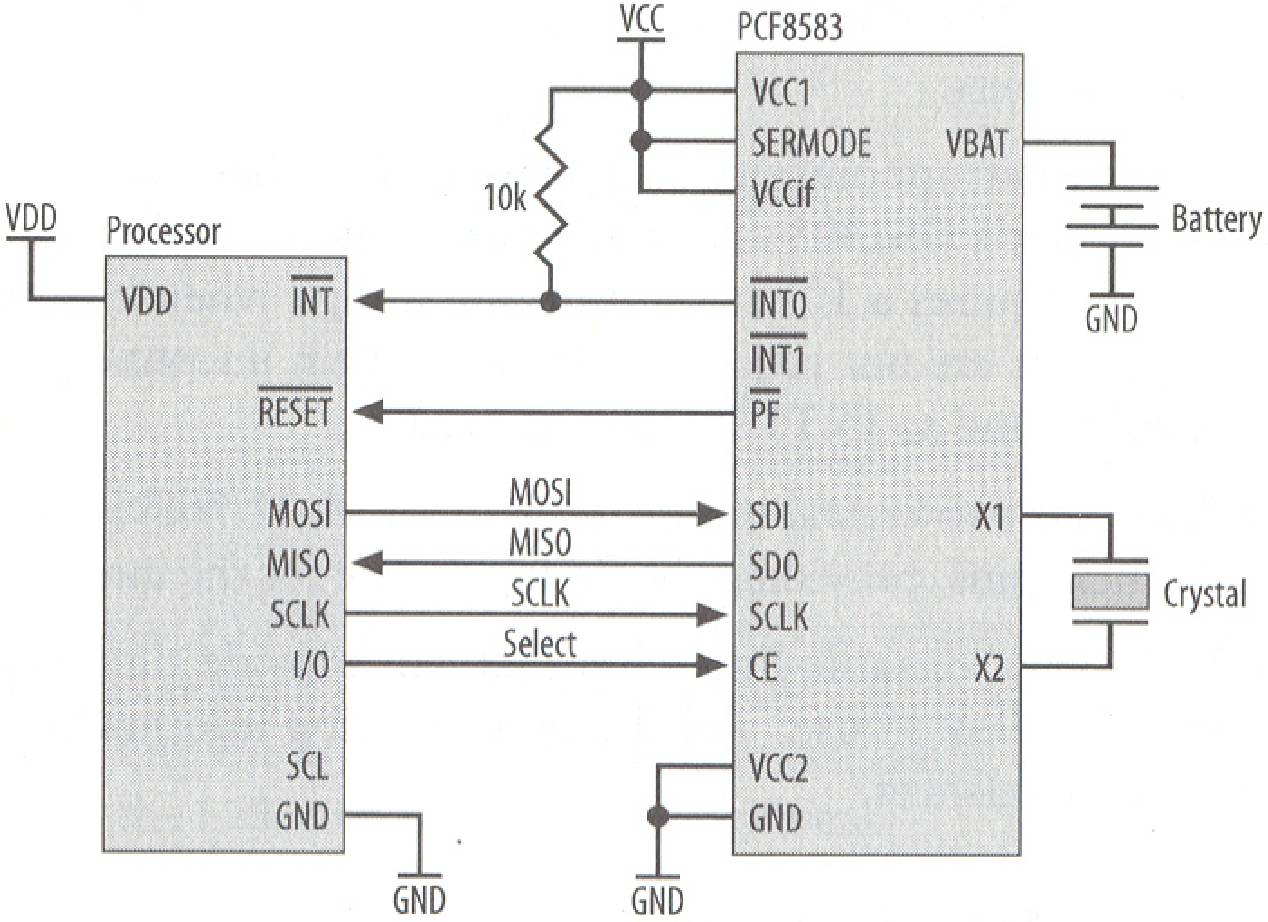


Figure 9-7. SPI timing with clock polarity high and clock phase one

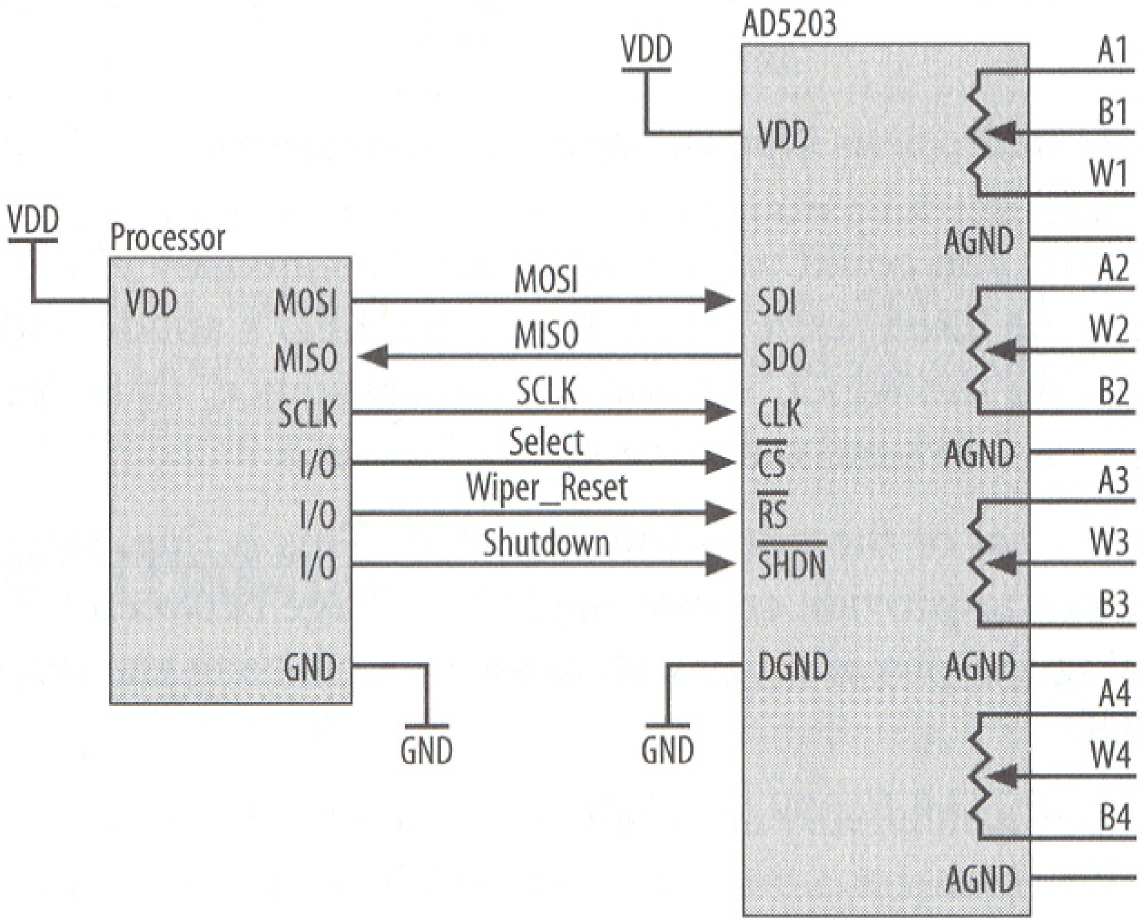
SPI (VIII)

- SPI-based Clock/Calendar



SPI (IX)

- SPI-based digital potentiometer

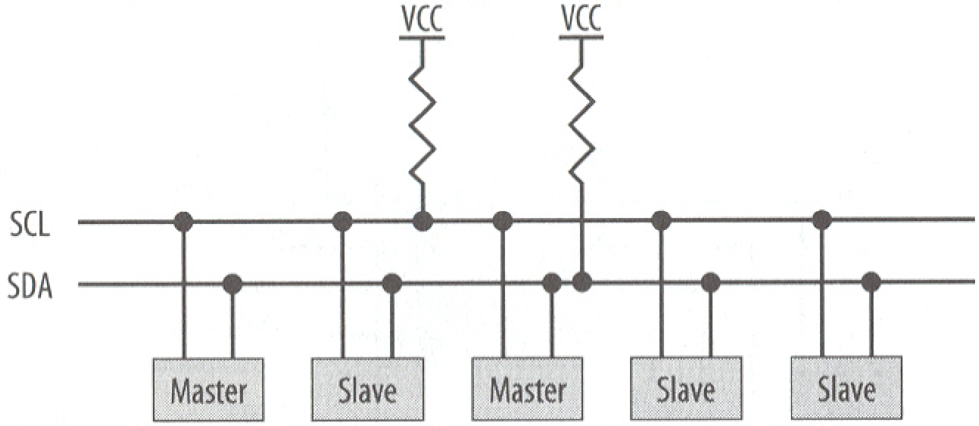


6. Inter Integrated Circuit (I^2C)

- I^2C features
  - A very cheap, yet effective, network used to interconnect peripheral devices within small-scale embedded systems.
  - Uses two wires to connect multiple devices in a multi-drop bus.
  - The bus is bidirectional, low-speed, and synchronous to a common clock.
  - Devices may be attached or detached from the I^2C bus without affecting other devices.
  - Supported by Microchip, Phillips, Intel, etc.
  - 100 kbps in standard mode, 400 kbps in fast mode.

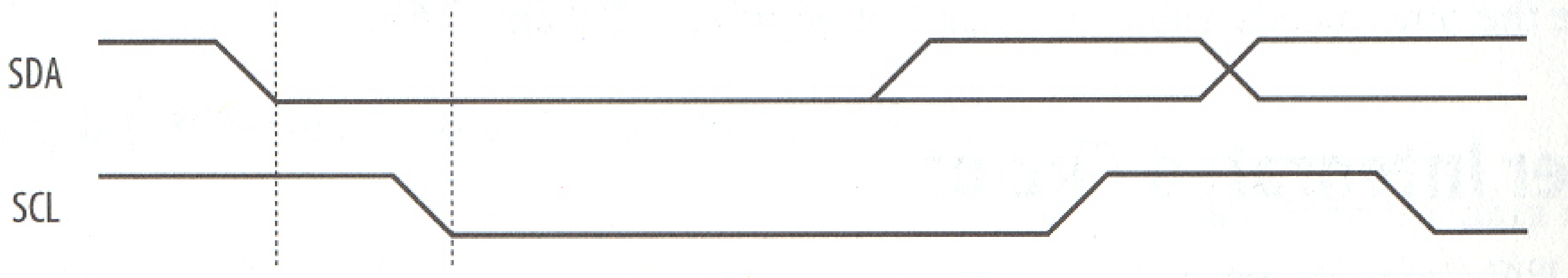
I^{2}C (II)

- Wires of I^{2}C
  - SDA (Serial DAta)
  - SCL (Serial CLock)
  - Both open drain, bidirectional
    - Connected to a positive power supply via a pull-up resisto.r
    - Remain high when not in use.
- Multi-master bus
  - Each device connected to the I2C bus has a unique address and can operate as a transmitter (a bus master), a receiver (a bus slave), or both.
  - Block diagram ->



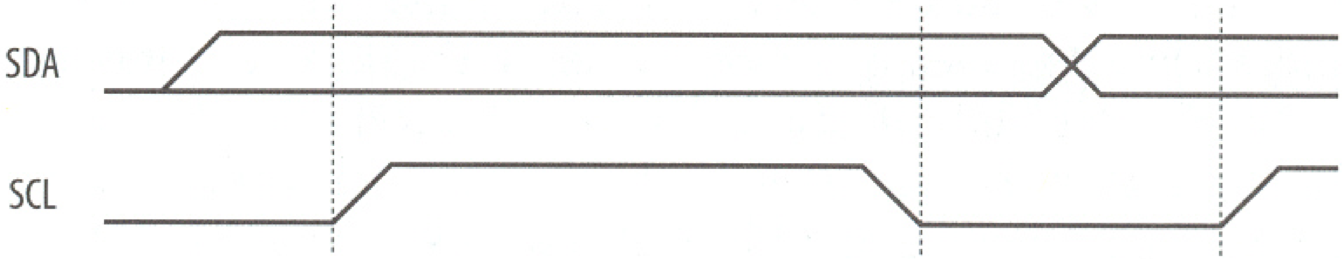
I^{2}C (III)

- I2C operation
  - I2C shares the same signal line for master transmission and slave response.
  - When idle, both SDA and SCL are high.
  - An I2C transaction begins with SDA goes low, followed by SCL.
    - A packet transmission is commencing.
  - While SCL is low, SDA transitions for the first valid data bit.
    - Start condition ->

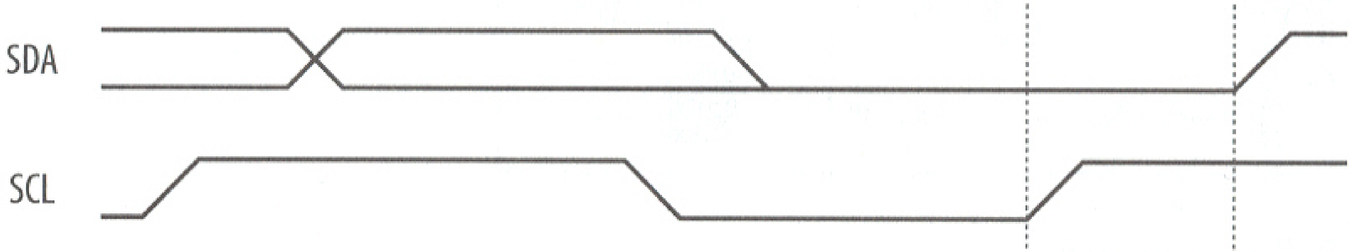


I^{2}C (IV)

- I2C operation (cont'd)
  - The bit is sampled on the rising edge of SCL and must remain valid until SCL goes low once more.
  - SDA transitions to the next bit, before SCL goes high once more ->

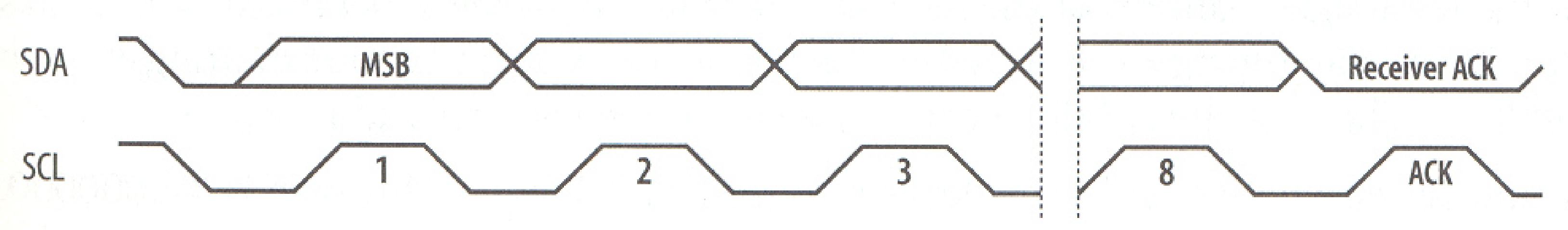


  - The transaction completes by SCL returning high followed by SDA: Stop condition ->



I^{2}C (V)

- Remarks
  - Any number of bytes may be transmitted in an I2C packet.
    - More significant bit of the packet is transmitted first.
  - If the receiver is unable to accept any more bytes, it can abort the transmission by holding SCL low.
    - This forces the transmitter to wait until SCL is released again.
- Acknowledge
  - Each byte transmitted must be acknowledged by the receiver.
    - Upon the transmission of the 8th bit, the master releases SDA.
    - The master then generates an additional clock pulse on SCL, which triggers the receiver to acknowledge the byte by pulling SDA low (Receiver ACK) ->

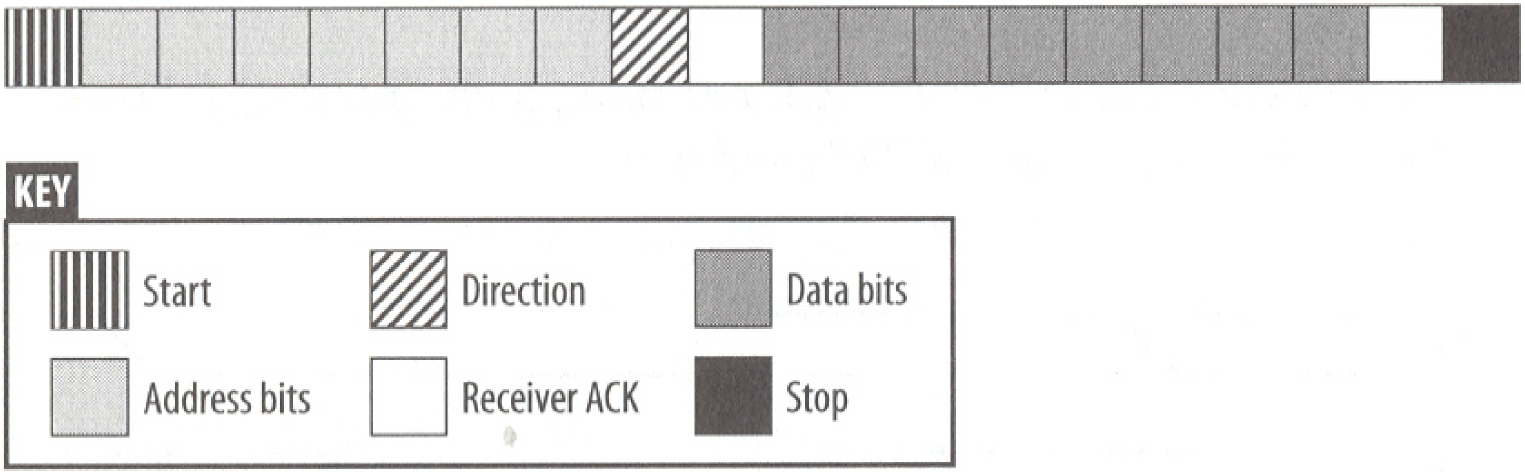


I^{2}C (VI)

- Multimaster bus
  - More than one master may attempt to start transmission at the same time.
  - A master transmitting a 0 bit will pull SDA low, but will leave the bus in its default state if the bit is to be a 1.
  - If two masters begin simultaneous transmission, a master leaving the bus in its default state for 1 1 bit, but detecting the bus pulled low by another master, will register an error condition and abort the transmission.

I^{2}C (VII)

- I2C packet ->

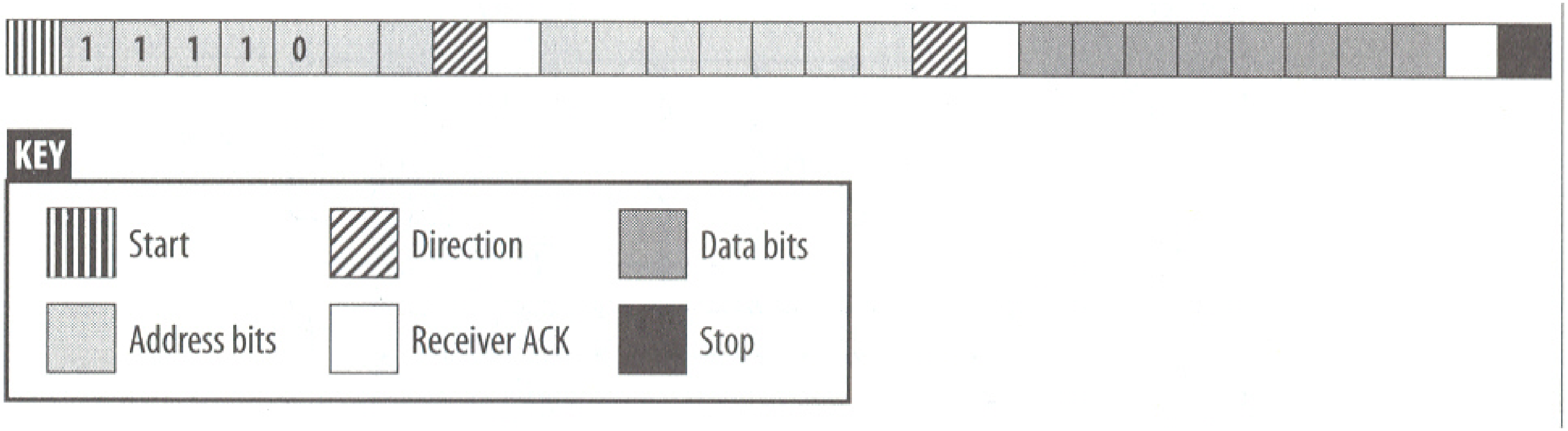


I^{2}C (VIII)

- Special addresses
  - General call address (%0000000 with dir 0)
    - Broadcasts to all I2C devices.
    - The master determines what slaves are available.
    - Second byte
      - 0x06: Reset slaves and respond with their addresses.
      - 0x04: Respond with their addresses without reset.
      - Other even data: Ignored.
      - Other odd data: A master transmits its own address to other masters.
        - Other bits contain the master's address.
  - Start byte (%0000001)
    - Indicate that a long data transfer is beginning.

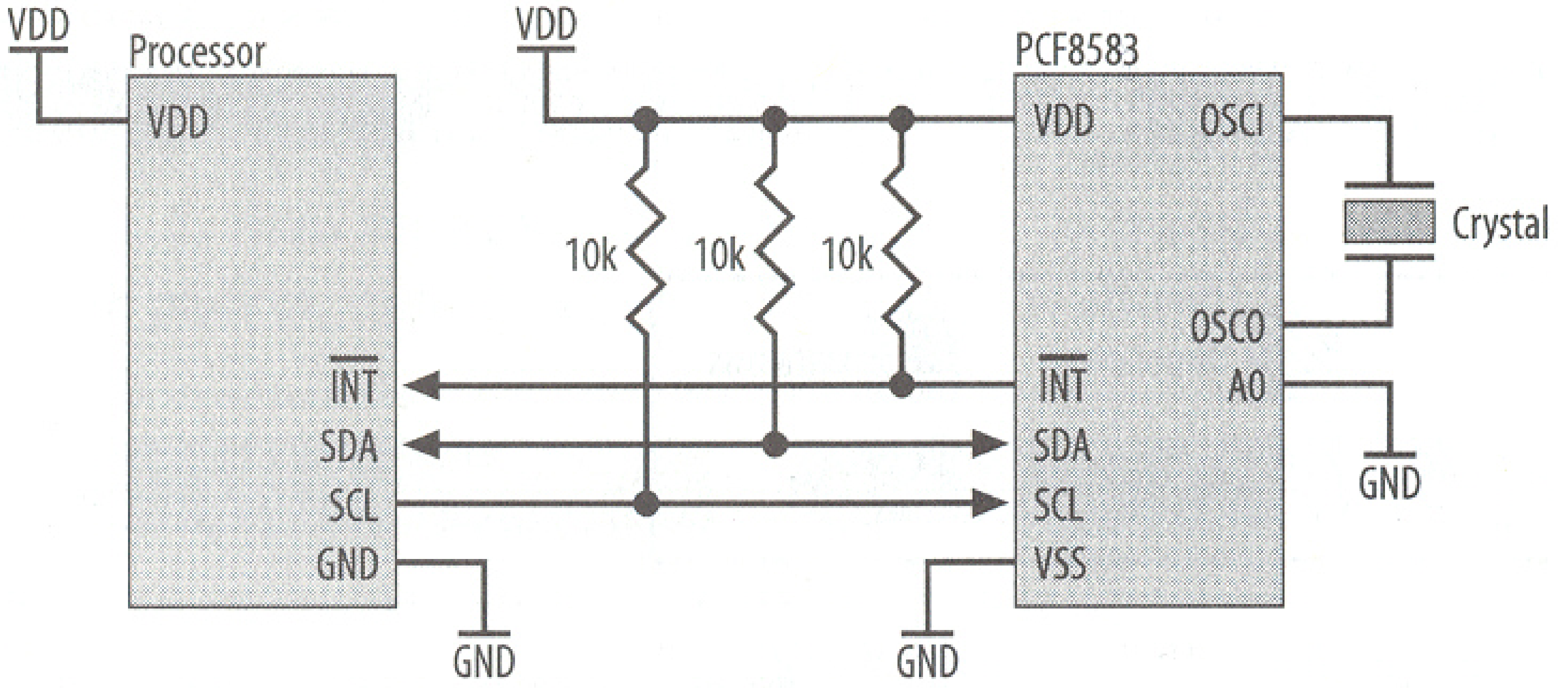
I^2C (IX)

- Extended 10-bit addressing
  - The first address byte begins with %11110xx
  - Address: 2 least-significant bits in the first byte and 8 bits of the second byte ->



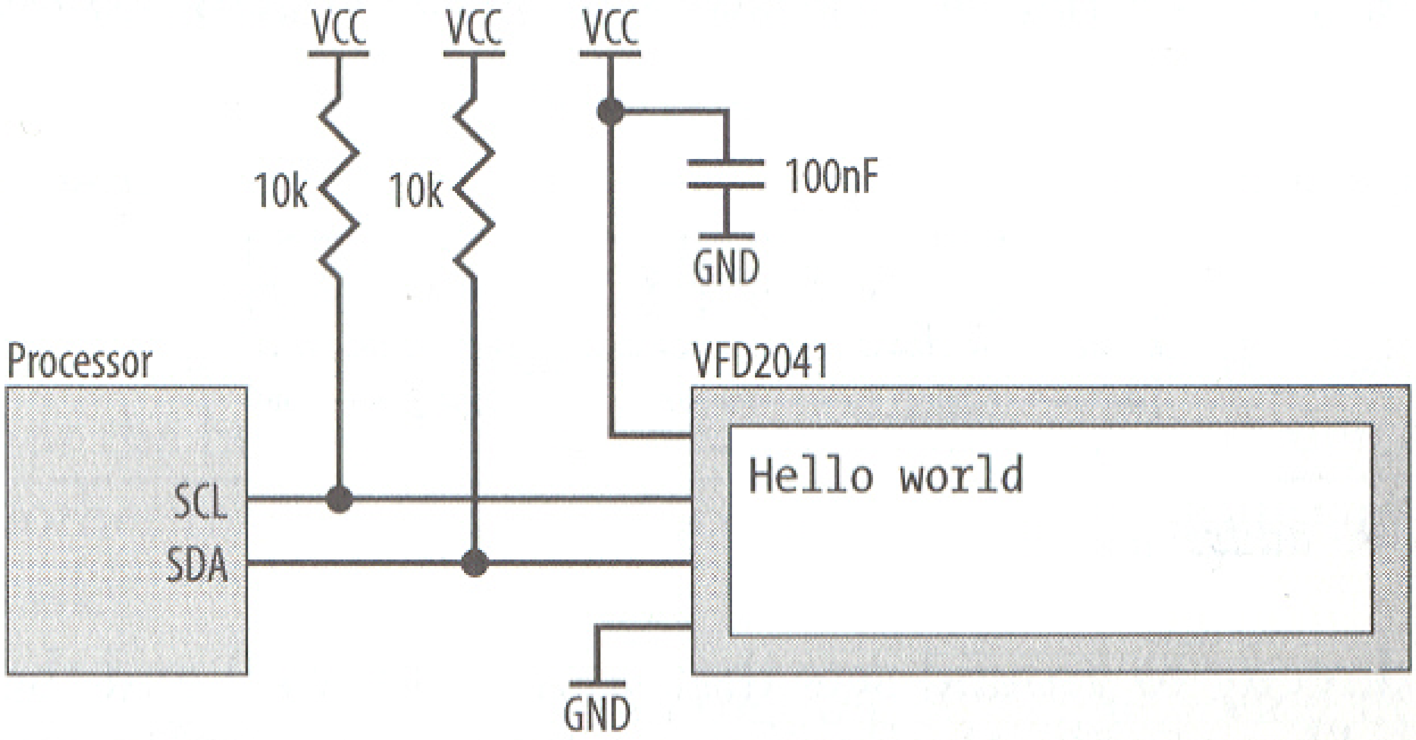
I^{2}C (X)

- Adding a real-time clock with I2C



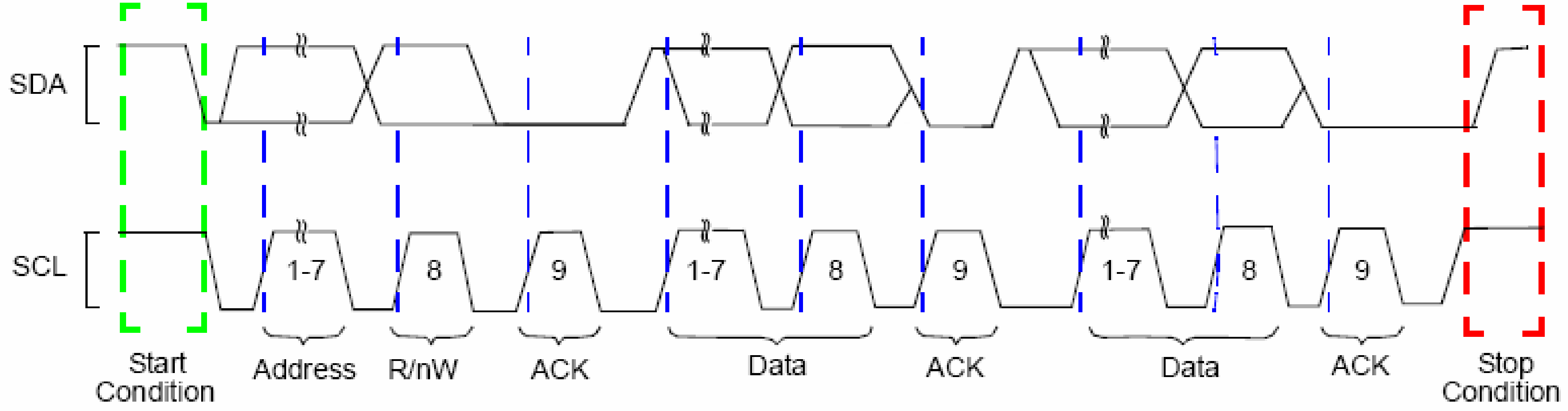
I^2C (XI)

- Adding a small display with I2C



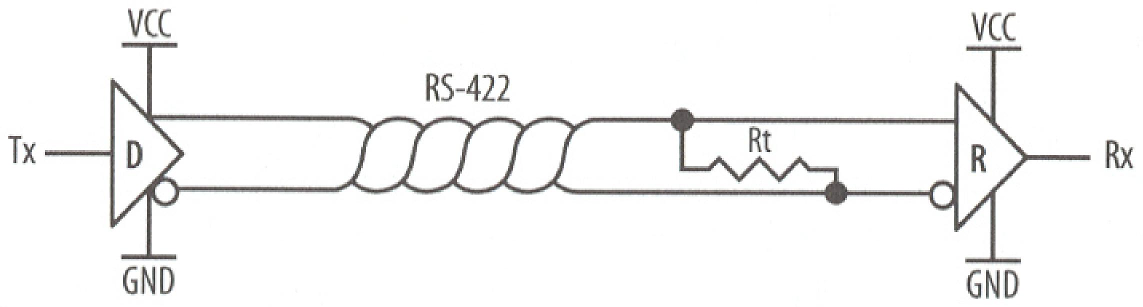
I^{2}C in Xscale (XII)

- Register Definitions
  - I2C Bus Monitor Register (IBMR)
  - I2C Data Buffer Register (IDBR)
  - I2C Control Register (ICR)
  - I2C Status Register (ISR)
  - I2C Slave Address Register (ISAR)
- Complete data transfer



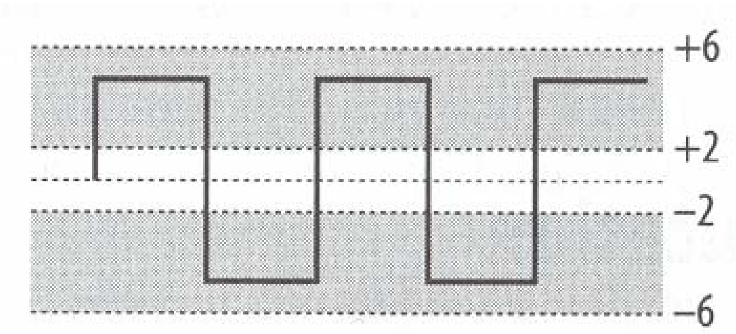
7. RS-422

- RS-422
  - Uses the difference signal
    - Balanced transmission: Not referenced to local ground.
  - Two wires: twisted pair or differential pair
    - Any noise or interference will affect both wires of the twisted pair, but the difference between them will be less effected: Common mode rejection.
  - Can carry data over longer distances and at higher rates with greater noise immunity than RS-232C.
    - Up to 1200 meters
  - RS-422 ->

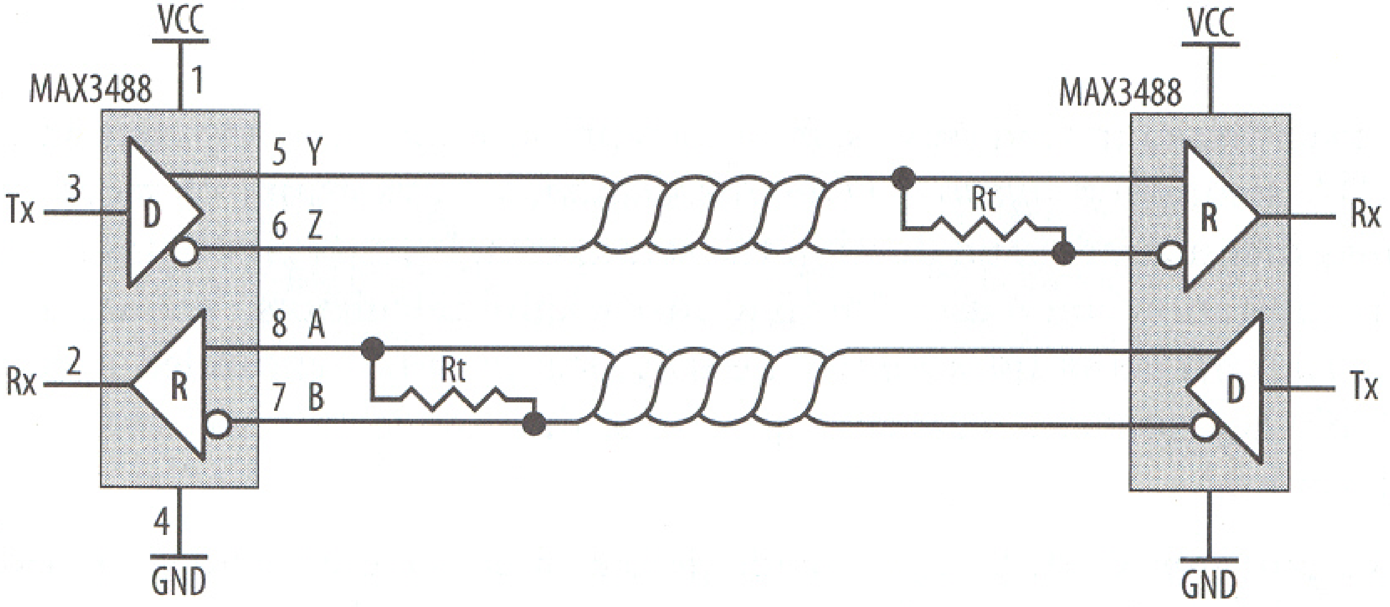


RS-422 (II)

- RS-422 voltage levels ->



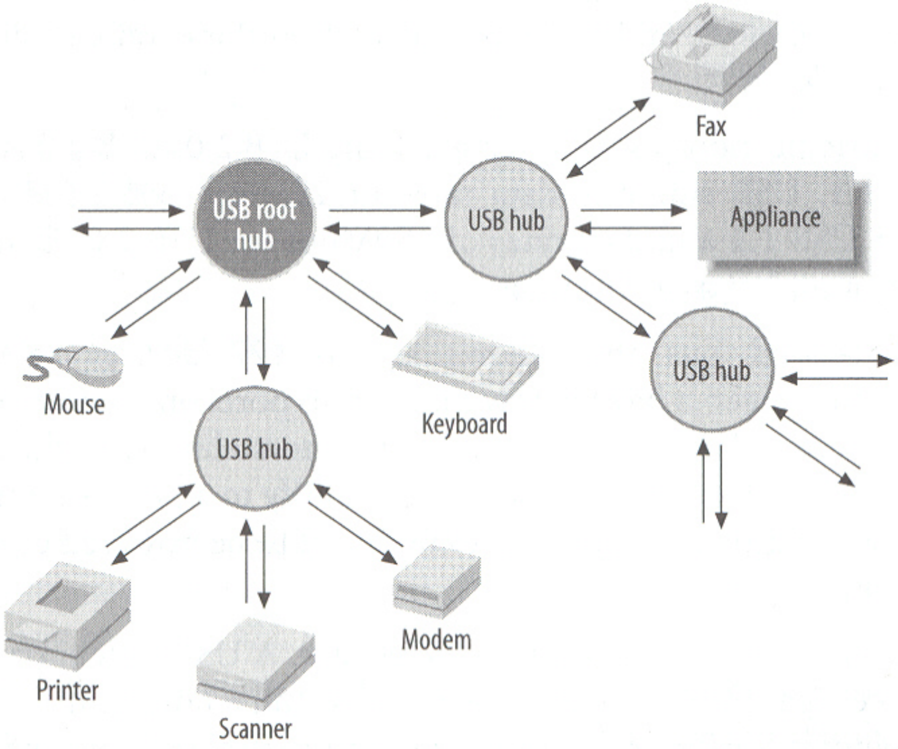
- Bidirectional RS-422 interface ->



8. Universal Serial Bus (USB)

- USB
  - Allows peripherals and computers to interconnect in a standard way with a standard protocol.
  - Possibility of plug and play for peripherals.
  - Supports the connection of printers, modems, mice, keyboard, joysticks, scanners, cameras, and much more.
  - Advantage for the user
    - No manual setup: Devices interact with the host computer's OS.
      - Extra layer of complexity to software.
    - Can provide power to peripherals through the cable.
      - No external power supply (or power cable) is required.
    - Hot swappable.
  - Specifications
    - USB 1.1: 1.5 and 12 Mbps
    - USB 2.0: 480 Mbps
    - http://www.usb.org

USB (II)



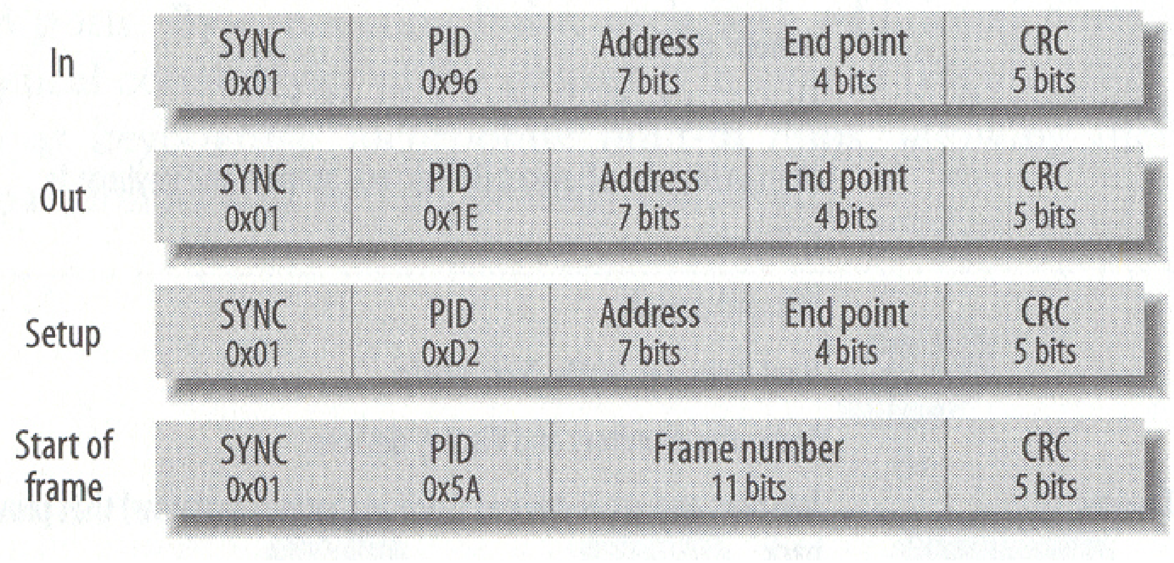
- Structures of USB
  - A high-speed bus up to 127 devices with tiered star structure ->
    - Root hub
    - USB hub
  - The host will regularly poll hubs for their status.
  - Bus enumeration
    - The detection and identification of USB devices
  - USB supports only one host computer.
    - Specifically intended for peripheral interfacing.

USB (III)

- USB packets
  - Four types of transfers
    - Control transfer: Configure the bus and devices, and return status information
    - Bulk transfer: Moves data asynchronously over USB.
    - Isochronous transfer: Used to moving time-critical transfer. Unidirectional and without CRC (Cyclic Redundancy Check)
    - Interrupt transfer: Receive data at regular intervals, ranging from 1 to 255 ms.
  - Packets
    - SYNC (synchronization): locks the receiver's clock. 0x01.
    - PID (Packet ID): Functions. 8 bits.
      - Upper 4 bit = inverse of lower 4 bit. For additional error checking.
    - Content (data, address, etc)
    - CRC (Cyclic Redundancy Check).

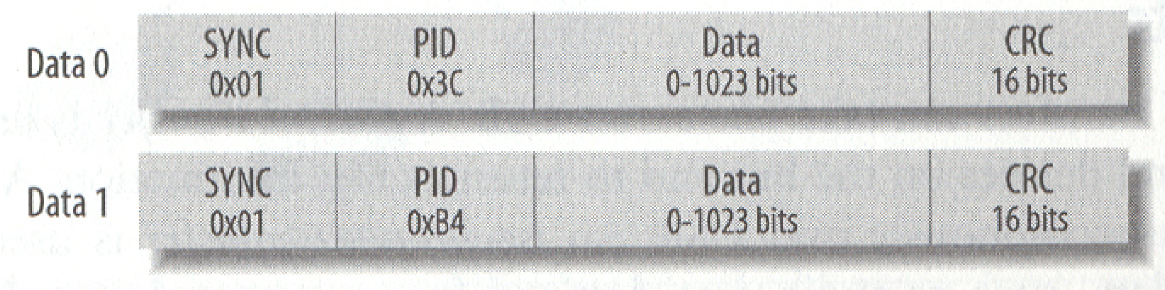
USB (IV)

- USB packet types
  - Token:



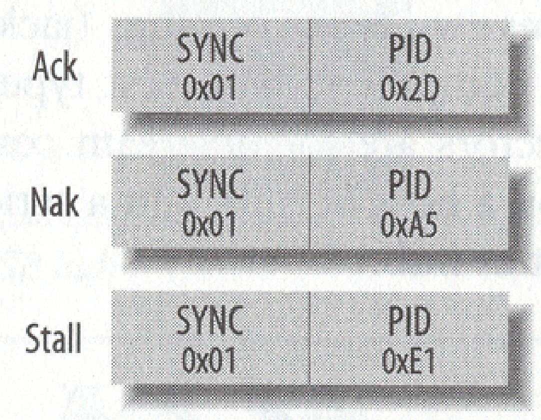
In	SYNC 0x01	PID 0x96	Address 7 bits	End point 4 bits	CRC 5 bits
Out	SYNC 0x01	PID 0x1E	Address 7 bits	End point 4 bits	CRC 5 bits
Setup	SYNC 0x01	PID 0xD2	Address 7 bits	End point 4 bits	CRC 5 bits
Start of frame	SYNC 0x01	PID 0x5A	Frame number 11 bits		CRC 5 bits

  - Data:



Data 0	SYNC 0x01	PID 0x3C	Data 0-1023 bits	CRC 16 bits
Data 1	SYNC 0x01	PID 0xB4	Data 0-1023 bits	CRC 16 bits

  - Handshaking:



Ack	SYNC 0x01	PID 0x2D
Nak	SYNC 0x01	PID 0xA5
Stall	SYNC 0x01	PID 0xE1

USB (V)

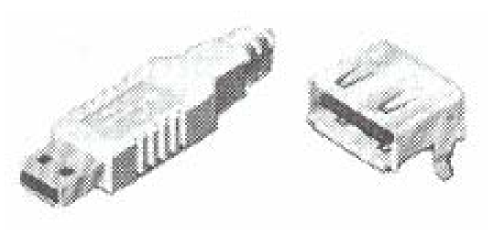
- USB physical interface
  - 4 wires, shielded

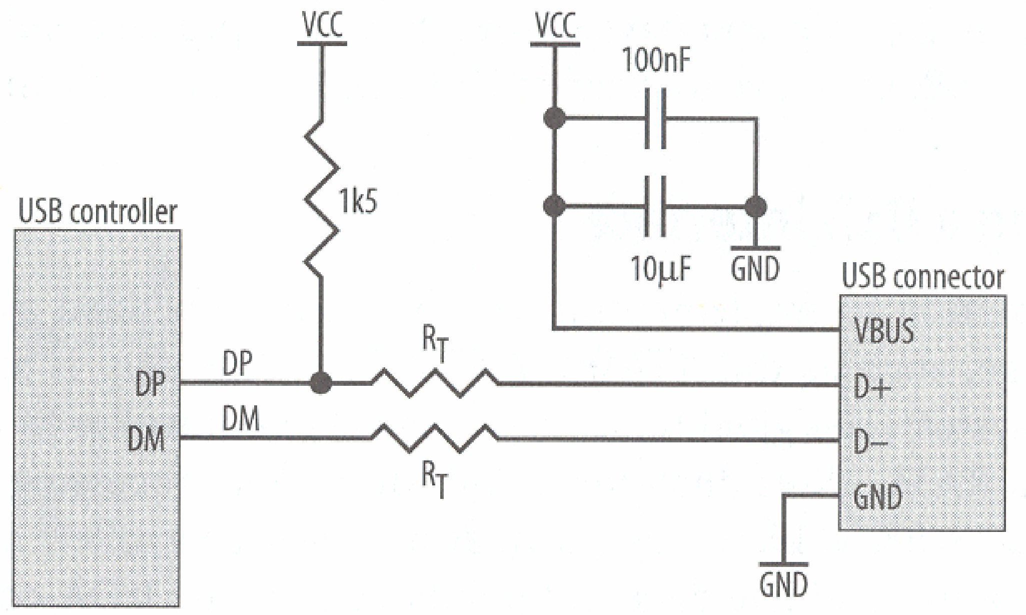
Connector pin	Signal	Purpose	Wire color
1	VBUS	USB device power (+5V)	Red
2	D+ (DP)	Differential data line	Green
3	D- (DM)	Differential data line	White
4	GND	Power and signal ground	Black

  - D+, D- twisted
  - Vbus: bus-powered, self-powered devices.

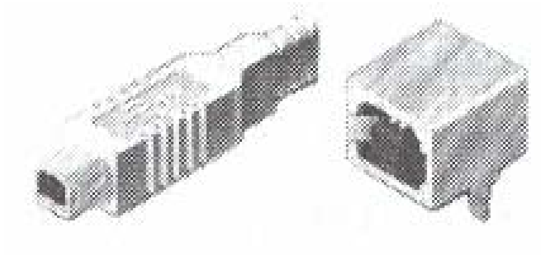
USB (VI)

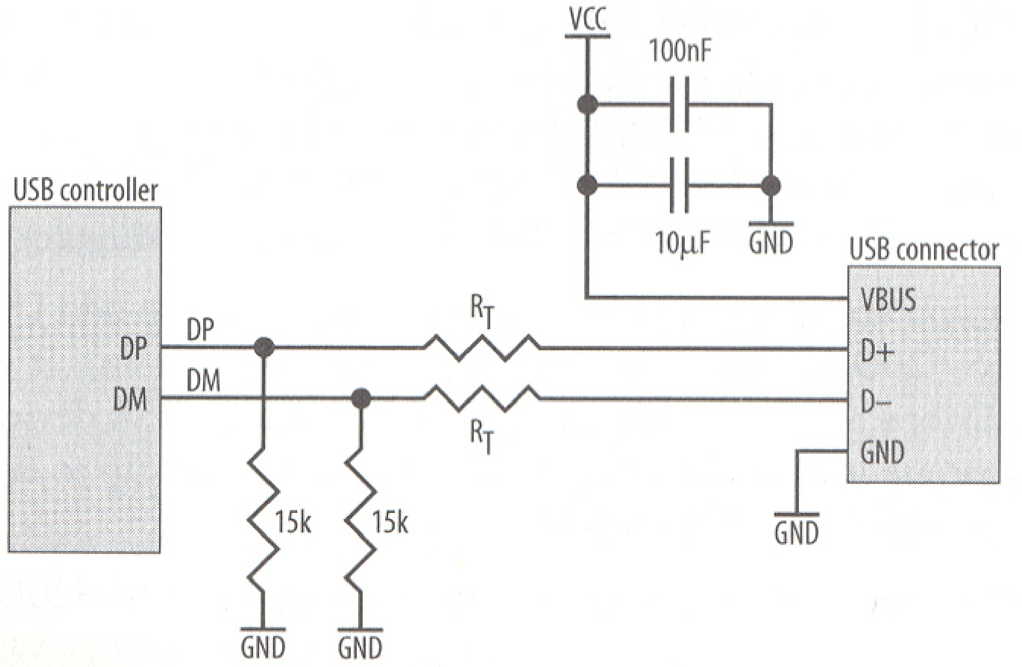
- USB Connectors
  - Upstream: Series A





  - Downstream: Series B





References

- SPI, I2C, RS422, and USB
  - John Catsoulis, “Designing Embedded Hardware”, O’Reilly, 2003.
- I2C
  - PXA255 Developer’s Manual, http://developer.intel.com