Digital Communication Techniques

V 2.0

AA 2020-2021

Summary

- General Issues

Parallel Communications

- Variants
- Standard Protocols

- Serial Communications
  - Isochronous, Asynchronous and Synchronous Transmission
  - Signals, encoding and modulation
  - Error handling
  - Standard protocols

General Issues

General Issues

- Information ⇒ Sequences of Symbols ⇒ Signs
  - Encoding
    - Cost, Reliability, Efficiency
- Binary symbols
  - *bit*
- Electrical (optical) signals
- Transmission media
  - Cables, ether, (optical fibre)
  - Generation, amplification, conversion equipment, etc.

General Issues

- The characteristics of the transmission media, often cascaded together, determine the type of signals most suitable for efficient transmission
  - In digital electronic circuits, bits are placed in correspondence with ranges of electrical voltage values
    - Level signal
  - In the various transmission media, the quantities used as signals can still be voltage or current, but bit values are not always associated with ranges of signal values
    - Modulation
      - -A characterising aspect of signals is their frequency band, which must be compatible with the transmission medium

General Issues

- Main characteristics of a transmission medium
  - Bandwidth expressed in Hz (100 KHz..1000 MHz)
    - Maximum transmission speed
  - Attenuation expressed in dB/Km (0.4..1.5)
    - Maximum distance that can be covered without repeaters
  - Propagation speed in Km/s (15000..150000)
    - The information transmission speed (baud = symbols/s) is different from the signal propagation speed (m/s)
  - Characteristic impedance in Ohm

(50..1000) Noise immunity

General Issues

- Information Transfer Directions
  - Simplex
    - Provides a single one-way channel allowing transfers from a sender to one or more recipients
  - Half-Duplex
    - Provides one bidirectional channel or two one-way channels with opposite orientation to each other on which, mutually exclusive, each device can either transmit or receive
      - This approach is often used even when *full-duplex* could be used, as the strict alternation of transmission and reception greatly simplifies the protocols
  - Full-Duplex
    - Requires the use of two separate channels for reception and transmission, which can thus take place simultaneously

General Issues

- Information to be transmitted
  - bit ⇒ characters ⇒ messages
- Synchronisation and symbol recognition
  - Agreement between sender and receiver on which time windows are to be used for symbol representation
  - Auxiliary signals are used to indicate significant synchronisation instants
    - Parallel communications (short distance)
      - Added signals (different cables)
    - Serial communications (for long distances)
      - One signal for synchronisation and information transport

General Issues

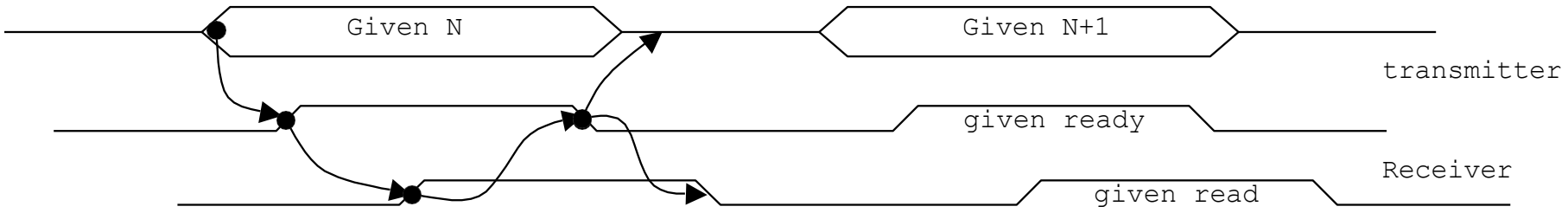
- Preventing errors
  - Reduction of disturbing phenomena
  - Protection of signals from disturbances
  - Choice of *robust* signals
- Recognition and correction of errors
  - Transmission error rate
  - Undetected error rate
  - Recognised error rate
    - With rejection of incorrect information
    - With reconstruction of information
- Management in space and time of communications
  - Avoiding conflicts of access to transmission media
    - Protocols

- Enabling information to reach its destination
  - Message routing

Parallel Communications

Parallel Communications

- Transmission of groups of bits (typically 8 or 16) in short distance links (a few metres)
- Synchronisation takes place via auxiliary signals and *handshaking* techniques
  - These techniques make it possible to adjust the speeds of the transmitter and receiver so as not to lose information and taking into account the tolerances on the different switching times of the various bits in parallel
    - Full handshake

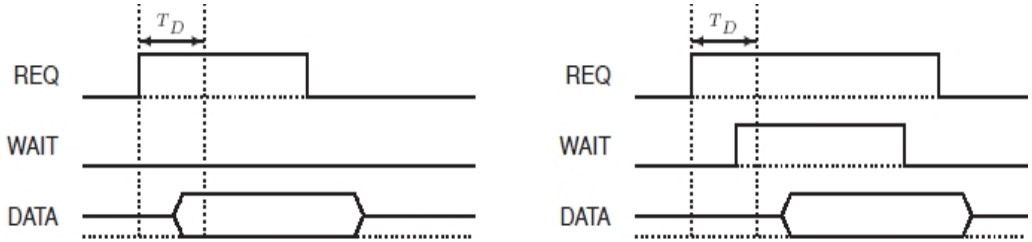


Parallel Communications

- Variants of handshaking
  - Strobe

  - Handshake

  - Strobe/Handshake



a. Slave veloce **b.** Slave lento

Parallel Communications

- Standard parallel protocols
  - IEEE-488 standard
  - Centronics
  - ISA/EISA
  - SCSI
  - PCI/ PCI X
  - Frontbus

Serial Communications

Serial Communications

- Used for communication over a few metres via a single signal used to transmit one bit at a time in time succession

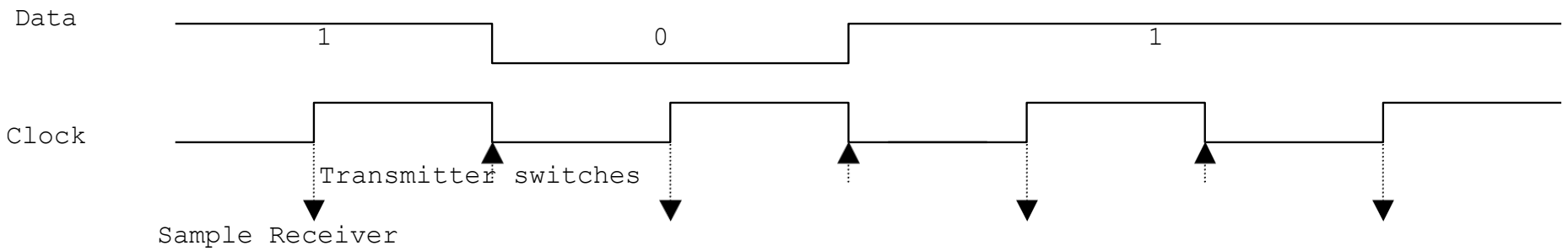
- Typical baud rates
  - Medium: microcontrollers, PLCs
    - 1200, 2400, 4800 and 9600 bits/sec
  - High speed standards: microcontrollers, modems
    - 19200, 38400, 56900,... bits/sec
  - Average speeds for special connections: field bus
    - 100 ... 1000 kbit/sec
  - High speed for LAN
    - 1 ... 1 Gbit/sec

Serial Communications

- Synchronisation techniques
  - Objective: to enable the receiver to recognise the time window of validity (and thus sampling) of the data
    - Isochronous Transmission
      - Single auxiliary synchronisation signal (clock) common to sender and receiver
    - Asynchronous transmission (standard speed)
      - Separate clock generators at sender and receiver, of similar frequencies and synchronised to each character
    - Synchronous transmission (fieldbuses, local networks)
      - Separate clock generators with receiver phase lock capability

Serial Communications

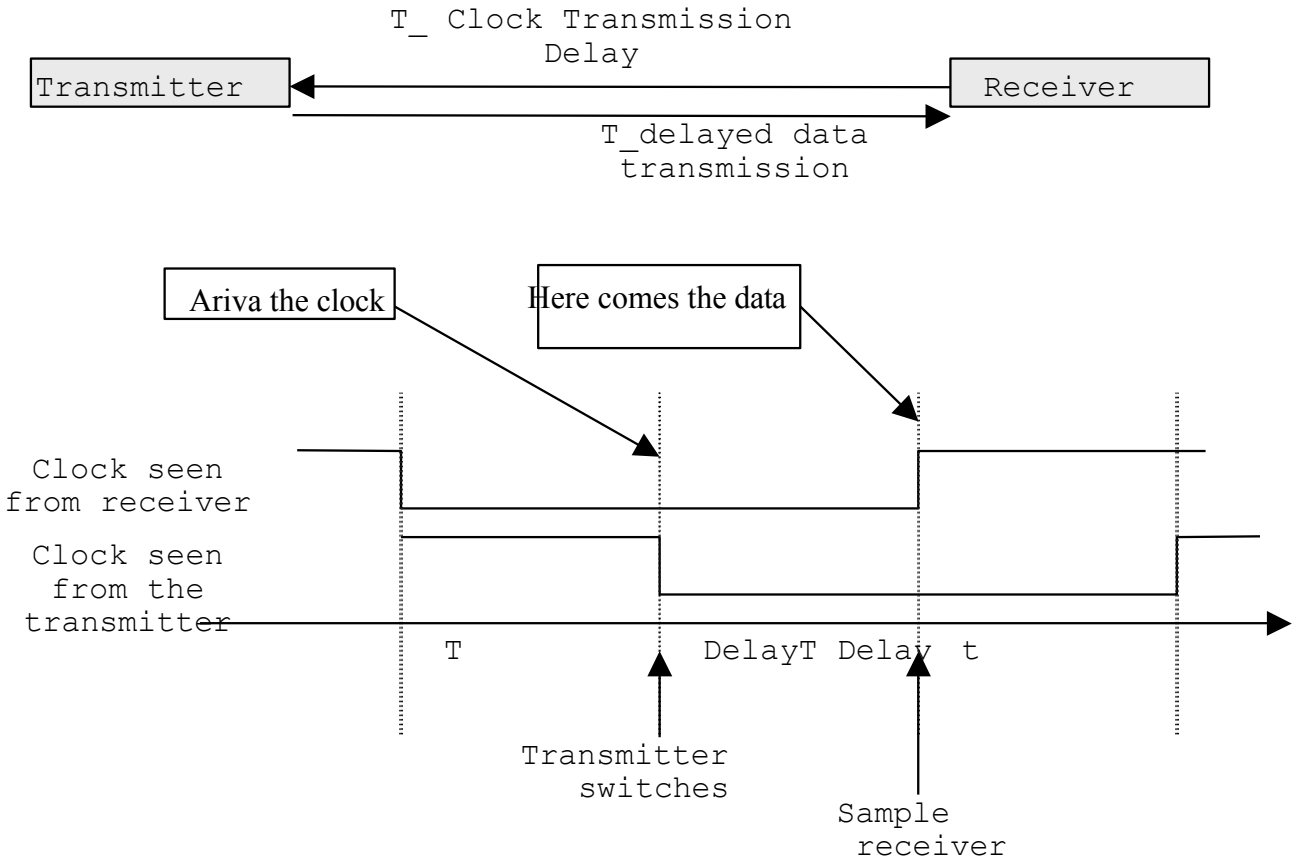
- Isochronous Transmission
  - Bit synchronisation is based on a single clock signal, shared between sender and receiver
    - Downward clock fronts
      - The sender sets the level of the next bit (switching edges)
    - Rising clock front
      - The receiver samples on the downhill fronts (sampling fronts)



Serial Communications

- Isochronous Transmission
  - In isochronous transmissions, the clock is on a separate conductor
    - Clock generated at the transmitter
      - Clock and data propagate in the same direction
        - A correct phase relationship is maintained
    - Clock generated at receiver
      - Clock and data propagate in different directions
        - The delay is equal to `2*Propagation time`
    - The distance between the falling edge of the clock and the rising edge must be greater than `2*Tempo_propagation`
      - Transmission speed limitation
  - Due to the criticality of propagation times and the separate clock line (possible disturbances), isochronous transmission is used for relatively short links

- Isochronous Transmission



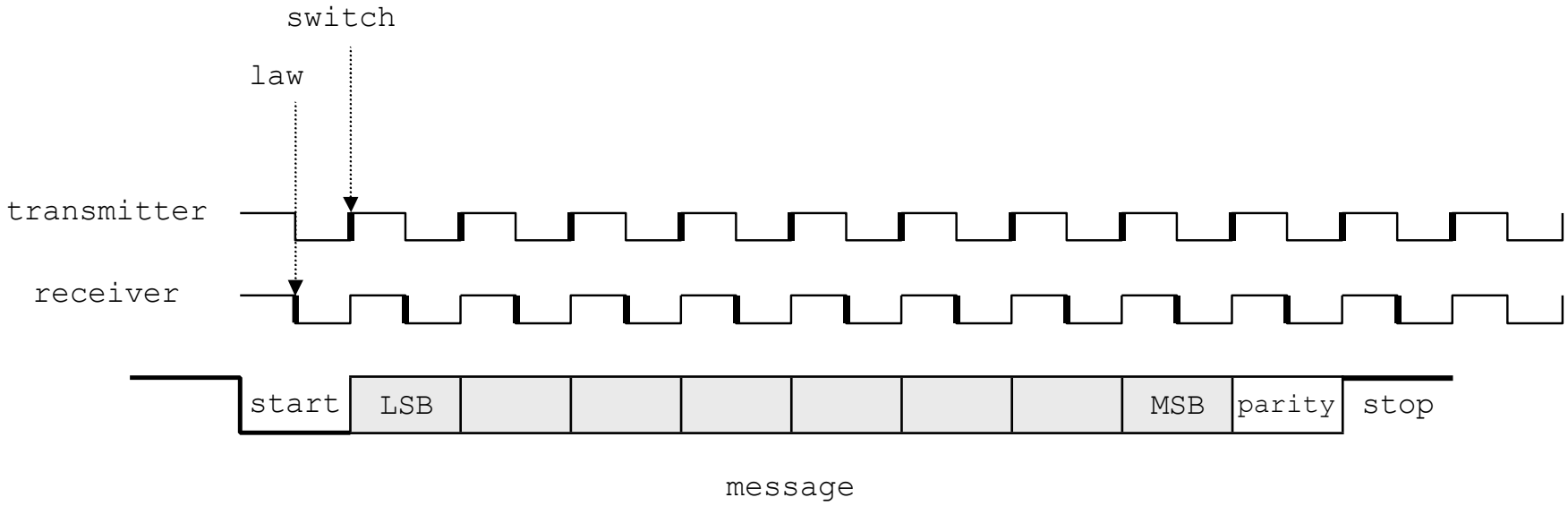
Serial Communications

- Asynchronous Transmission
  - Receiver and transmitter have their own clocks with similar and stable frequencies allowing at least the transmission of a single character
  - Synchronisation with each character
    - Each byte is preceded by `start bits` and followed by at least one `stop bit`
    - The receiver synchronises on the front of the `start bit` and remains synchronised throughout the character
      - The rate of transmission between characters may vary
  - The asynchronous technique is widely used for short point-to-point connections

- Asynchronous Transmission
  - Once the two clocks are synchronised, they behave as the only clock in the isochronous case
    - The switching edges are provided by the transmitter's clock, while the sampling edges are provided by the receiver's clock and, for proper acquisition, must fall within the time window of the bit
    - The similarity of clock frequencies must be such as to ensure that for all bits of the character, the phase shift does not exceed the interval of +/- half the duration of a bit

Serial Communications

- Asynchronous transmission with 8 bits plus parity

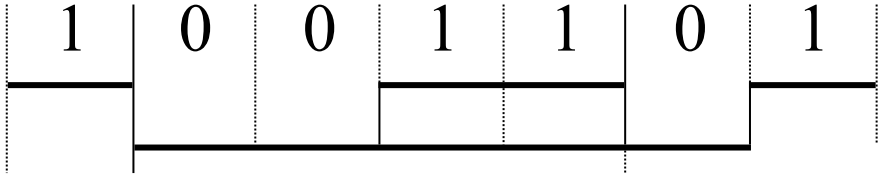


Serial Communications

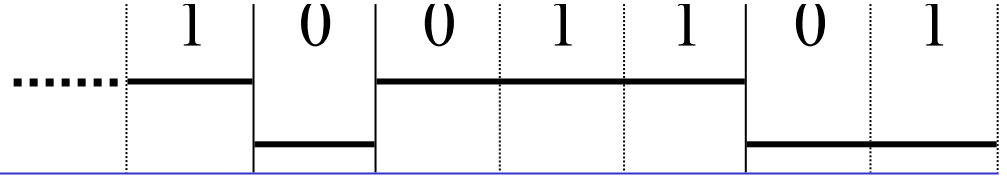
- Synchronous Transmission
  - The transmitter has a stable frequency clock
  - The receiver has a phase lock loop clock on the received data switches
    - Bit synchronisation takes place after a number of switchings that allow phase locking
    - Byte synchronisation takes place on the recognition of special synchronisation characters (3 ... 5) placed before each **message**
      - -The message bits are transmitted without pauses, avoiding long sequences without switching
        - " To prevent long sequences without switching from losing phase lock, bits are added to interrupt sequences without switching
  - Typical in high-speed transmissions

Serial Communications

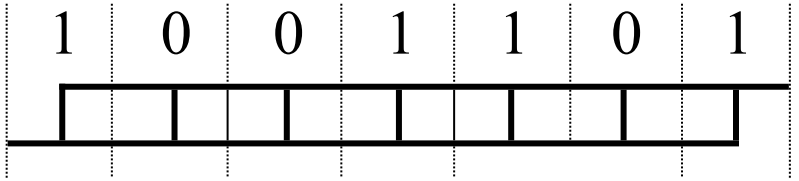
- Signals, coding and modulation
  - Two-value (or baseband) signals: used with cables for short distance transmissions and in LANs
    - NRZ (*Not Return to Zero*): each transition indicates a change in logical value
      - The most used



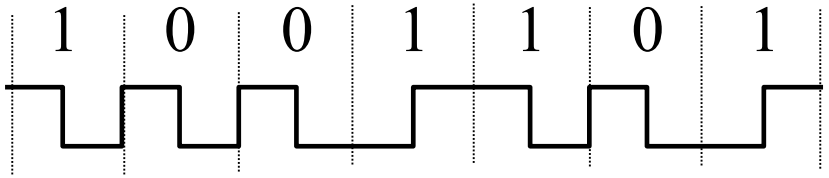
    - NRZI: uses the presence or absence of transition to indicate a logical value
      - Less used but requires less bandwidth



- Signals, coding and modulation
  - Two-value (or baseband) signals
    - Two-phase Manchester: a specific edge indicates a well-defined logical value (rise=1; fall=0)
      - Used in synchronous transmissions as they favour phase locking at the expense of bandwidth



    - Differential Manchester: a logical 1 (0) is represented by an edge at the beginning of the interval while a 0 (1) is indicated by an edge in the middle of the period.



Serial Communications

- Signals, coding and modulation
  - Sine signals
    - Amplitude modulated
    - Frequency modulated
    - Phase modulated
  - These signals are used for modem communications (modulators/demodulators) over switched telephone lines, coaxial cables or radio links
    - The modem in transmission is supplied with the baseband signal, as we have seen above, and the modem in reception reproduces its pattern fairly faithfully, so that at the digital terminals the modulation is transparent

Serial Communications

- Error Management
  - In terms of character
    - Acknowledgement: `parity bit`
    - Correction: `hamming code`
  - At message level
    - Recognition: additional characters function of transmitted characters
      - Cyclic Redundancy Check
        - Calculated on 16 bits
      - Checksum
        - Sum Form 256
      - Longitudinal parity
    - Correction: retransmission is normally used

Serial Communications

- Standard Protocols
  - RS-232-C (1969)
    - It was proposed to connect DTE devices with DCE devices
      - DTE: *Data Terminal Equipment* (computers, printers, etc.).
      - DCE = *Data Communications Equipment* (modem)

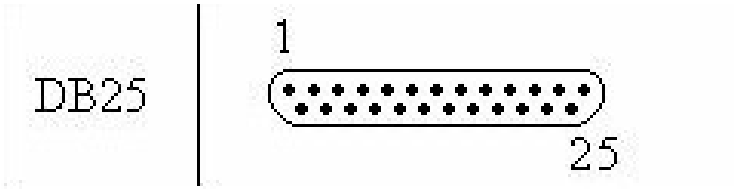
    - Electrical signals: 0 = ON ⇒ V > +3 V; 1 = OFF ⇒ V < -3

    V- Distance: < 15 m

    -Speed: <20,000 bits/sec

    -Sequence: fromLSB to MSB with possible addition of parity-

    Dte connector: Cannon male 25 contacts



Serial Communications

- Standard Protocols
  - RS-232-C: main signal layout

1	GND	protective mass
2	TXD	data transmitted
3	RXD	data received
4	RTS	submission request
5	CTS	consent to sending
6	DSR	data set ready
...		
15	TXC	transmission clock
17	TXD	reception clock
20	DTR	data terminal ready

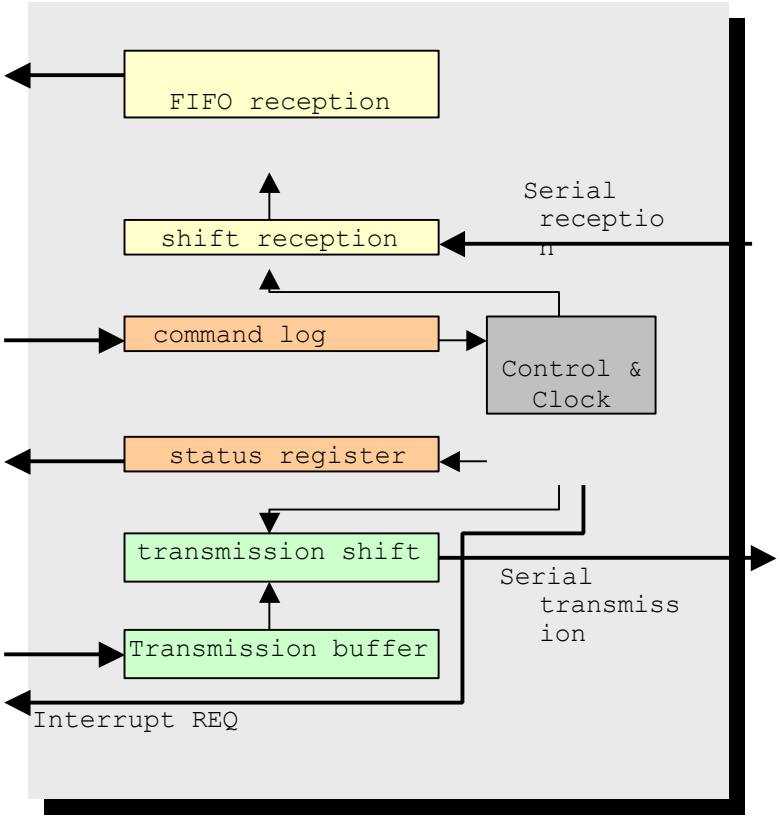
  - The standard is also often adopted in a reduced form for direct connection between DTE-type devices, i.e. without the interposition of a modem, but with the use of a cable (called a '**null modem**') in which connections 2-3, 15-17, etc. are crossed.

Serial Communications

- Standard Protocols
  - U(S)ART
    - Serial transmissions, both synchronous and asynchronous, are so widespread that they have prompted the production of integrated circuits known as U(S)ART (*Universal Synchronous Asynchronous Receiver Transmitter)*, generally designed for easy interfacing with popular microprocessors
      - Often such circuits are even integrated in microcontrollers/SOCs
    - These are integrated circuits that perform the operations required for the lower levels of the ISO-OSI hierarchy in serial transmissions
      - UART NS8250 and NS16550
    - Note that we speak of UARTs in the case where circuits are intended for the simpler and more widely used asynchronous communication only

Serial Communications

- Standard Protocols
  - U(S)ART
    - Functions
      - Serial/parallel conversion
      - Parity generation/verification
      - Bit or character synchronisation
      - Buffering
      - Generation of interrupt requests
    - Typical registers
      - Buffer and transmission shift
      - Buffer and reception shift
      - Mode registers
      - Command registers
      - State registers

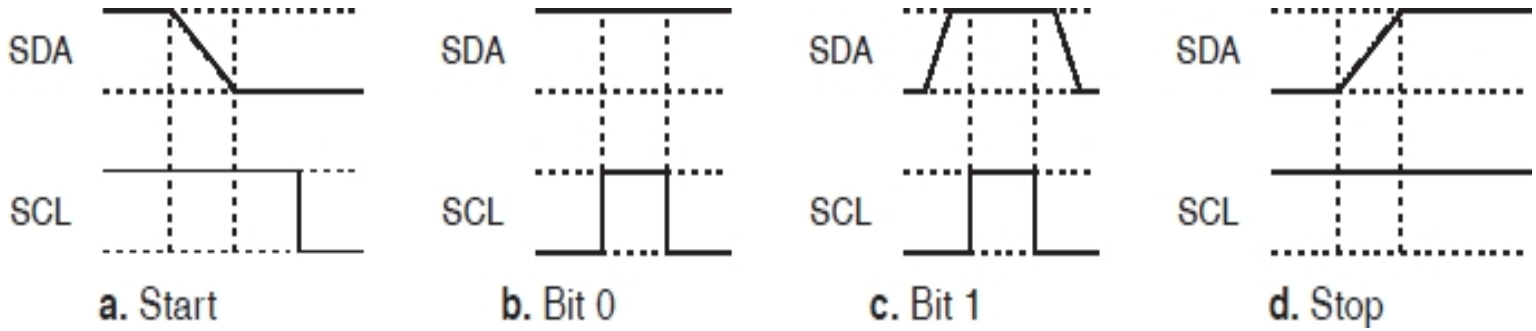


Serial Communications

- Standard Protocols
  - I2C (*Inter IC*)
    - Developed by Philips in the early 1980s
    - Designed for the interconnection of several devices over a limited distance
      - Variable transmission rates between 100~Kbit/s with 7-bit addressing (according to the original specification) and 3.4~Mbit/s with 10-bit addressing (newer versions of the standard)
    - It uses two lines: SDA (Serial Data Line) and SCL (Serial Clock Line) to which all devices involved in communication are connected
      - Such devices can be master or slave, the master devices being the only ones able to initiate communication
        " In general, the standard provides for the possibility of having more than one master device and provides a deterministic and relatively simple mechanism for bus negotiation

- Standard Protocols
  - I2C (*Inter IC*)
    - Being an asynchronous bus, synchronisation between master and slave occurs at each symbol
    - The communication of a symbol starts with a start condition and ends with a stop condition
      - The transmitted package consists of the following fields
        - The start condition
        - A 7- or 10-bit address
        - The requested operation
        - An acknowledge bit, asserted by the slave to indicate receipt of the address
        - The data byte
        - A new acknowledge bit, asserted by the slave to indicate successful data reception
        - The stop condition

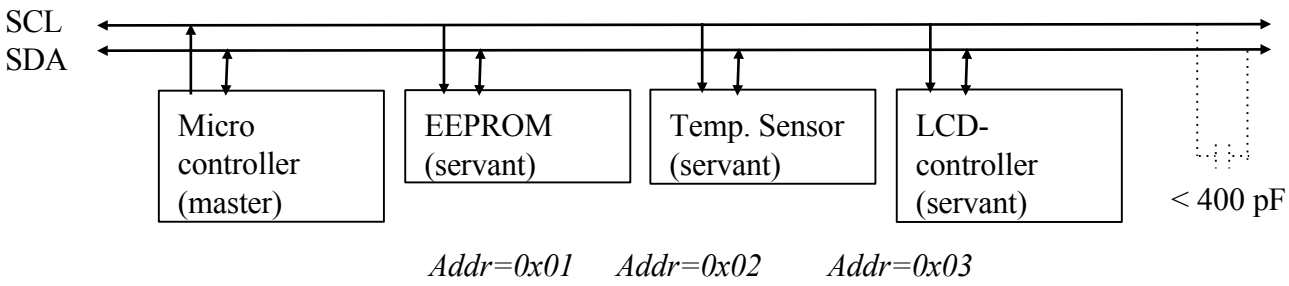
- Standard Protocols
  - I2C (*Inter IC*)
    - Transmission and start/end conditions

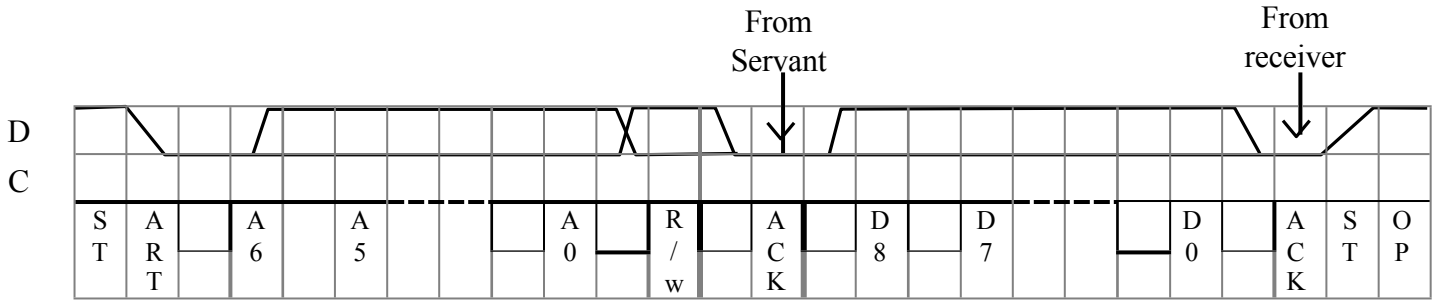


    - Logical structure of a package

- Standard Protocols
  - I2C (*Inter IC*)
    - For both address and data transmission the bits are ordered from most significant to least significant
      - The master always has control of the clock and data lines, except during the transmission of acknowledge bits, for which the slave forces the values on these lines
    - More than one master device can be connected to the same bus
      - Arbitration policy that allows a master device to determine whether it is actually able to take control of the bus
        - *Open-drain* connection realising the *wired-and* function
      - A master device wishing to take control of the bus must check the following two conditions: check that the line has no value 0 and be able to force a 1 on the clock line

- Standard Protocols
  - I2C (*Inter IC*)





Typical read/write cycle

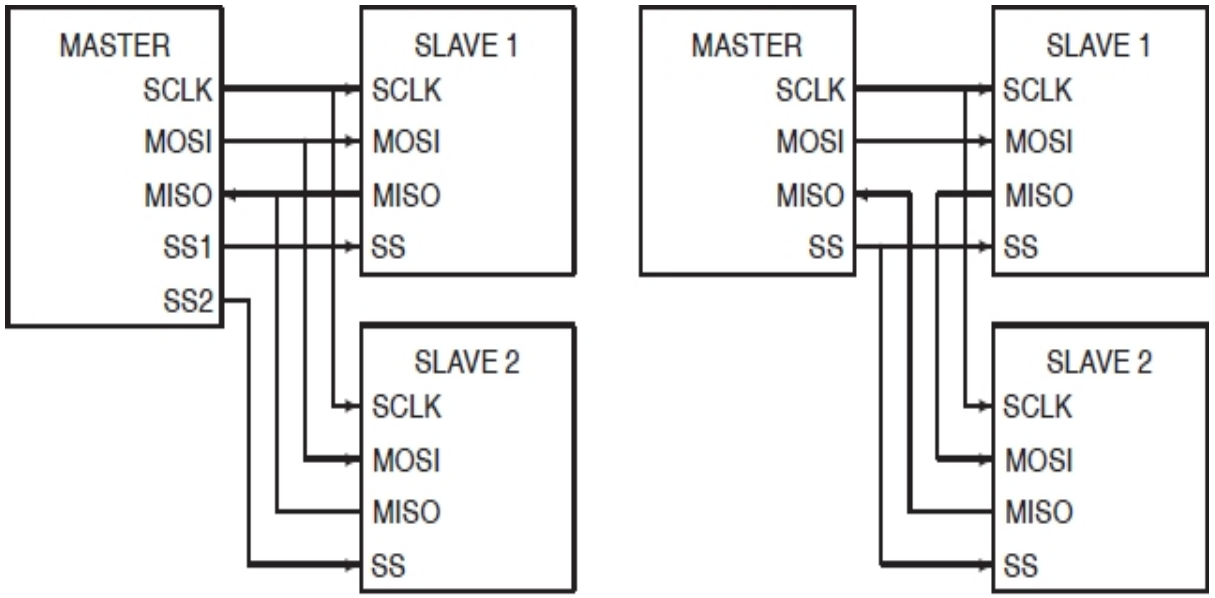
- Standard Protocols
  - SPI (*Serial Peripheral Interface*)
    - It is based on a two-way synchronous transmission scheme in which one device acts as the master
    - The protocol uses a bus with at least 4 lines for each slave
      - SCLK (*Serial Clock*)
      - MOSI (*Master Output Slave Input*)
      - MISO (*Master Input Slave Output*)
      - SS (*Slave Select*) for each slave device
    - MISO signals are *wired-or* connected on a single line and therefore it is necessary for non-active slaves, i.e. those for which SS is high (the slave select signal is claimed to be low), to keep their MISO line in high impedance

- Standard Protocols
  - SPI (*Serial Peripheral Interface*)
    - To initiate communication, the master asserts the SS signal for the desired slave by setting it to 0, then generates a clock signal at a frequency equal to or lower than the maximum frequency supported by the selected slave and transmits it on the SCLK line
      - During the entire time the clock is present, master and slave communicate full-duplex on the MOSI and MISO lines
    - A disadvantage of this scheme is that it requires a slave select line for each slave
      - A scheme can be used to overcome this limitation *daisy-chain*

- Standard Protocols
  - SPI (*Serial Peripheral Interface*)
    - The most obvious advantages of SPI-based communication schemes over a serial bus such as, for example, I2C are higher throughput due to both full-duplex communication and the absence of addressing
    - On the other hand, as there is no acknowledge mechanism at the physical level, the SPI protocol does not allow the master to be notified of successful communication or an error condition
      - In addition, the scheme provides for a single master device, which is often too limiting

Serial Communications

- Standard Protocols
  - SPI (*Serial Peripheral Interface*)



a. Connessione parallela

b. Connessione daisy-chain

SERIAL PERIPHERAL INTERFACE (SPI) -Adv

Presenter
2020-11-27 08:48:03

The main advantages of the SPI interface are:
-Full-duplex transmission
- high transmission frequencies can be achieved
- arbitrary length of data to be transmitted (not limited to 8 bits)
- maximum flexibility
- simplicity
simple and robust hardware structure
- clock produced from a single source and requiring no precision
- signals are unidirectional and lend themselves easily to being buffered or isolated

Pros of SPI bus:

- Full-duplex transmission
- High transmission frequencies can be achieved
- Arbitrary length of data to be transmitted (not limited to 8 bits)
- Maximum flexibility
- Simplicity
- Simple and robust hardware structure
- Clock produced by a single source
- Signals are unidirectional and can be buffered or isolated

SERIAL PERIPHERAL INTERFACE (SPI) - Disa

Cons of SPI BUS:

- Use of more lines
- The chip select system requires one pin for each slave
- No standardisation
- SPI is multi Slave, but not multi Master
- No confirmation of transmission
- No peripheral addressing (excluding SS signal)

Presenter
2020-11-27 08:48:04

The main advantages of the SPI interface are:
-Full-duplex transmission
- high transmission frequencies can be achieved
- arbitrary length of data to be transmitted (not limited to 8 bits)
- maximum flexibility
- simplicity
simple and robust hardware structure
- clock produced from a single source and requiring no precision
- signals are unidirectional and lend themselves easily to being buffered or isolated

SERIAL PERIPHERAL INTERFACE (S

Comparison between ISP and I2C

- Both are synchronous serial data transfer standards
- SPI has better throughput than I2C
- SPI is simpler, but can only support one master

It is recommended to use SPI:

- In case of only one Master and a limited number of Slaves
- When the maximum transmission speed is required
- When it is not possible to have a synchronous communication m odule and emulate a sofware connection

It is recommended to use I2C:

- In case you are in multi-master systems
- When you have systems with many units connected to the same bus
- Where a synchronous serial communication management module exists integrated in the microcontroller, since a complete I2C software connection is not simple.

Presenter
2020-11-27 08:48:04

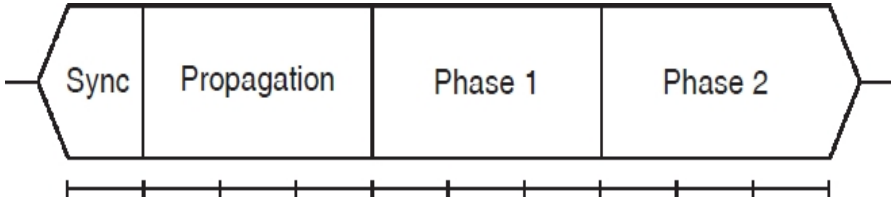
Let us compare SPI and I2C. Both are synchronous serial data transfer standards. Both are designed for communication over short distances. In general, it can be said that SPI h a s a better throughput than I2C due to full-duplex communication and the lack of addressing. On the other hand, the SPI protocol can only support one master and does not provide a different acknowledgement mechanism than the I2C protocol. In conclusion, the use of SPI is recommended: in the case where one finds oneself with only one Master and a limited number of Slaves (which, in general, is the typical situation for a microcontroller system) when maximum transmission speed is desired
when you do not have a synchronous communication module and emulate the connection via I2C software is used when:
in the case of multi-Master systems
when you have systems with many unitś connected to the same bus
where a synchronous serial communication management module is integrated in the microcontroller, since a complete emulation of I2C' via software is not easy.

- Standard Protocols
  - CAN (*Controller Area Network*)
    - The CAN bus is based on a differential serial protocol, originally developed in the 1980s by Bosch for the connection of control devices in environments heavily affected by electromagnetic noise
      - This bus is particularly suitable for the transmission of small messages (8 bytes) and uses a CRC-15 code to protect the data from transmission errors
        - " Supports speeds up to 1~Mbit/s for distances below 40~m, but at lower frequencies can cover greater distances, e.g. 500~m at 125~Kbit/s

- Standard Protocols
  - CAN (*Controller Area Network*)
    - The CAN bus features a transmission mechanism without explicit arbitrage thanks to the definition of the priority concept
      - -A high-priority message wins arbitration and is transmitted, while a low-priority message detects that a higher-priority message is already in transmission and puts itself on hold
        - " This mechanism is realised through a model based on binary logic and the concepts of dominant bit, a 0, and recessive bit, a 1

	dominante	recessivo
dominante	dominante	dominante
recessivo	dominante	recessivo

- Standard Protocols
  - CAN (*Controller Area Network*)
    - Asynchronous bus using a structure for the transmission of each bit
      - Ensure alignment of clock and sampling instants



    - Each bit is divided into four segments
      - The first is the synchronisation segment of fixed duration
      - The second, propagation, aims to compensate for physical delays
      - Finally, the last two segments have variable durations and can be shortened or lengthened to keep the clocks aligned
        - " The data is always sampled at the border between phase 1 and phase 2

- Standard Protocols
  - CAN (*Controller Area Network*)
    - At a higher level, the protocol implemented via the CAN bus is based on the frame concept
      - A frame constitutes the elementary transmission unit and can be of four different types
        "Data Frame, Remote Frame, Error Frame and Overload Frame
    - The most common type is the data frame, as it is used for the exchange of data, which constitute the bulk of traffic