# Serial Communication Buses

# Serial Communication

- Sending data one bit at one time, sequentially
- Serial vs parallel communication
  - cable cost (or PCB space), synchronization, distance !
  - speed ?
- Improved serial communication technology allows for transfer at higher speeds and is dominating the modern digital technology:
  - RS232, RS-485, I2C, SPI, 1-Wire, USB, FireWire, Ethernet, Fibre Channel, MIDI, Serial Attached SCSI, Serial ATA, PCI Express, etc.

# RS232, EIA232

- The Electronic Industries Alliance (EIA) standard RS-232-C (1969)
  - definition of physical layer (electrical signal characteristics: voltage levels, signaling rate, timing, short-circuit behavior, cable length, etc.)
  - 25 or (more often) 9-pin connector
  - serial transmission (bit-by-bit)
  - asynchronous operation (no clock signal)
  - truly bi-directional transfer (full-duplex)
  - numerous handshake lines (seldom used)
  - many protocols use RS232 (e.g. Modbus)

# Voltage Levels

- RS-232 standard convert TTL/CMOS-level signals into bipolar voltage levels to improve noise immunity and support long cable lengths

| TTL/CMOS (UART) | → | RS232: |
|---|---|---|
| 0V = logic zero | → | +3V…+12V (SPACE) |
| +5V (+3.3V) = logic one | → | −3V…−12V (MARK) |

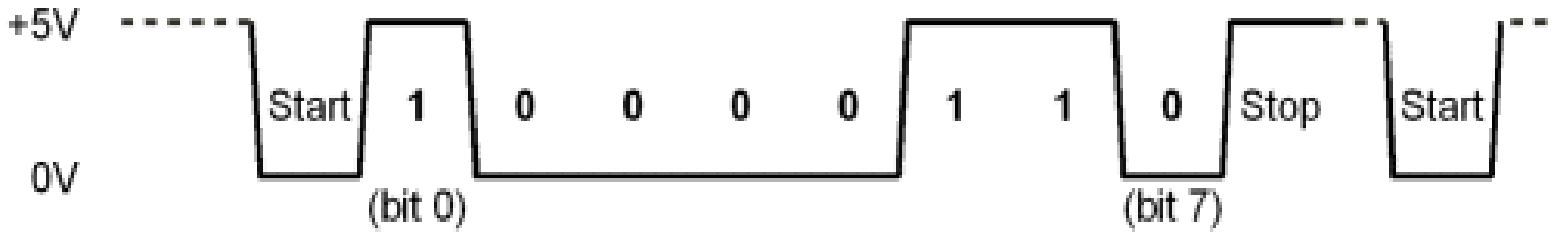
- Some equipment ignores the negative level and accepts a zero voltage level as the "MARK" state
- The "dead area" (−3V ... +3V) may vary, many receivers are sensitive to differentials of 1V

# Data frame

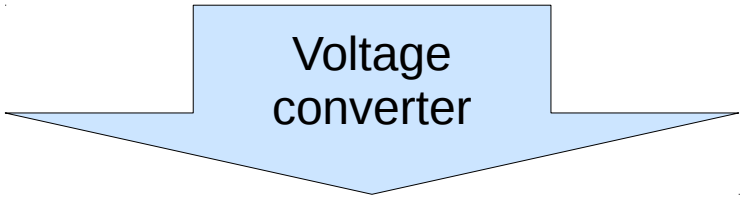
- Complete one-byte frame consists of:

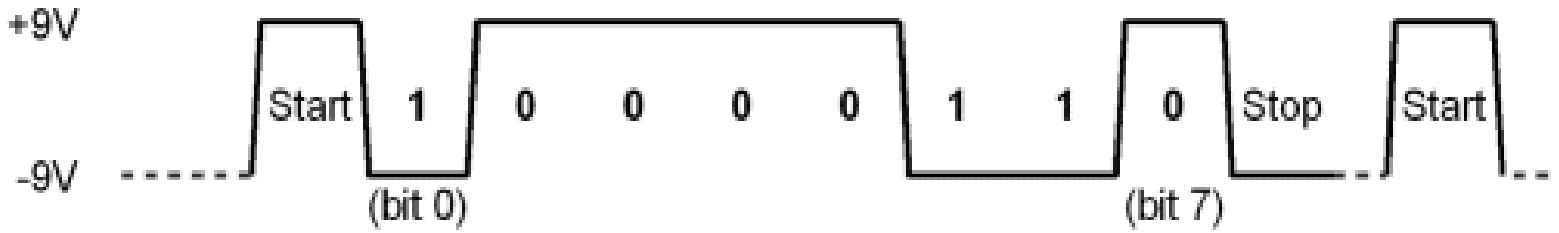
  start-bit (SPACE), data bits (7, 8), stop-bits (MARK)

  e.g. 9600-8N1 or 19200-7E1




TTL (5V) RS-232 byte transfer (ASCII 'b'=0x61 hex)




Typical RS-232 byte transfer (ASCII 'b'=0x61 hex)

# Data Rates

- Pure EIA RS-232C standard permits data rates up to 19200 bps and cable lengths up to 400 meters (but not both)
- Data rates as high as 256kbps (over less than 2 m) are possible
- Universal Asynchronous Receiver/Transmitter (UART) – component responsible for fast communication via RS-232
- Common 16550 UART incorporates a 16-byte FIFO and is mandatory for communications at speeds above 9600 bps

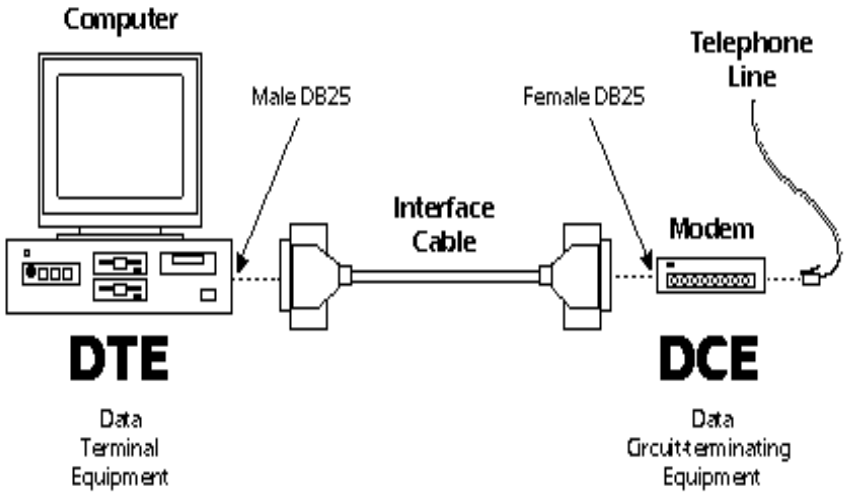
| data rate (bps) | maximum distance (meters) | (feet) |
|---|---|---|
| 19200 | 15 | 45 |
| 9600 | 25 | 76 |
| 4800 | 50 | 152 |
| 2400 | 100 | 304 |
| 1200 | 200 | 608 |
| 600 | 400 | 1216 |

# DTE and DCE

- The RS-232 defines two classes of devices:

male chauvinism?

  - data terminal equipment (DTE) – computer (male)
  - data communication equipment (DCE) – modem (female)



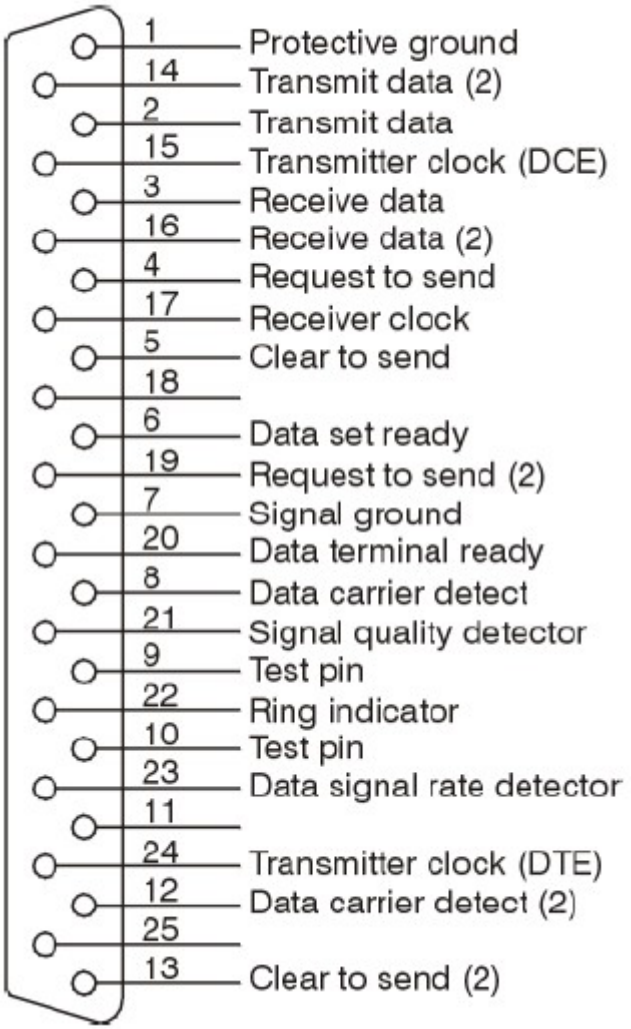

| Signal Name | | Pin Number | | Direction |
|---|---|---|---|---|
| | | 25-pin | 9-pin | |
| transmitted data | TD | 2 | 3 | DTE ▶ DCE |
| received data | RD | 3 | 2 | DTE ◀ DCE |
| request to send | RTS | 4 | 7 | DTE ▶ DCE |
| clear to send | CTS | 5 | 8 | DTE ◀ DCE |
| data terminal ready | DTR | 20 | 4 | DTE ▶ DCE |
| data set ready | DSR | 6 | 6 | DTE ◀ DCE |
| data carrier detect | DCD | 8 | 1 | DTE ◀ DCE |
| ring indicator | RI | 22 | 9 | DTE ◀ DCE |
| signal ground | GND | 7 | 5 | |

  - e.g. TD is output line on DTE, but input line on DCE
  - distinction between DTE&DCE allows to avoid confusion

# Signals

- Complete RS232 standard define a lot of signals and is fairly complex
- Many communications options exist, but only very few are used in practice

```
CTS     Clear To Send [DCE --> DTE]
DCD     Data Carrier Detected (Tone from a modem) [DCE --> DTE]
DCE     Data Communications Equipment eg. modem
DSR     Data Set Ready [DCE --> DTE]
DSRS    Data Signal Rate Selector [DCE --> DTE] (Not common)
DTE     Data Terminal Equipment eg. computer, printer
DTR     Data Terminal Ready [DTE --> DCE]
FG      Frame Ground (screen or chassis)
NC      No Connection
RCk     Receiver (external) Clock input
RI      Ring Indicator (ringing tone detected)
RTS     Request To Send [DTE --> DCE]
RxD     Received Data [DCE --> DTE]
SG      Signal Ground
SCTS    Secondary Clear To Send [DCE --> DTE]
SDCD    Secondary Data Carrier Detected (Tone)[DCE -> DTE]
SRTS    Secondary Ready To Send [DTE --> DCE]
SRxD    Secondary Received Data [DCE --> DTE]
STxD    Secondary Transmitted Data [DTE --> DTE]
TxD     Transmitted Data [DTE --> DTE]
```

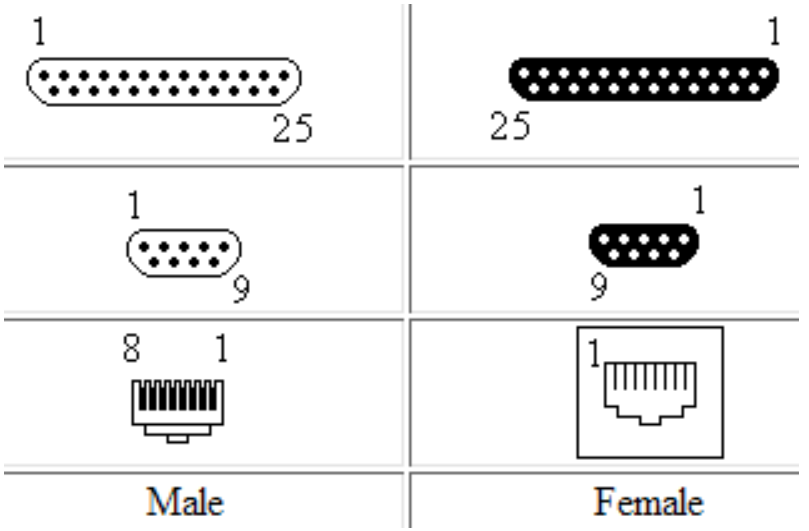


# Handshaking

- Hardware handshaking
  - RTS/CTS
    - DTE asserts the "request to send" (RTS) signal when it is ready to receive data and deasserts it when it cannot accept data; a DCE asserts "clear to send" (CTS) when it is ready to receive data
  - DTR/DTE
    - DTE asserts the "data terminal ready" (DTR) signal, and DCE asserts the "data set ready" (DSR) signal
- Software handshaking
  - XON/XOFF
    - requires that the receiver send a character (Control-S, ASCII 19) to halt data transfer and another character (Control-Q, ASCII 17) to resume transfer

# Interface

- Sub-D 25, Sub-D 9, RJ45 and others

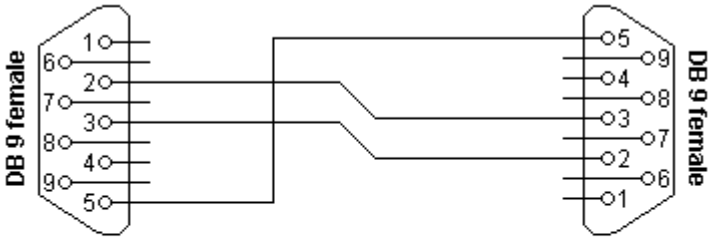



- DTE ↔ DCE – Straight connection
- DTE ↔ DTE – Null-modem (cross-over cable)
- DTE – Loopback wiring

- Various DB25 to DB9 adapters

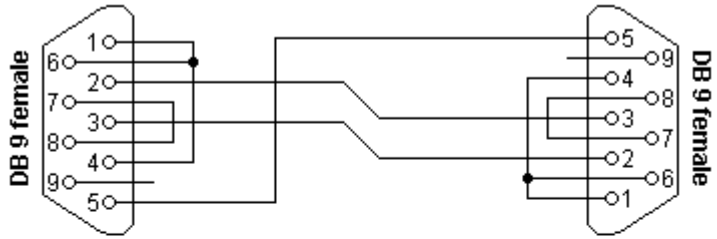
# Null-modem Wirings

### Simple null modem without handshaking



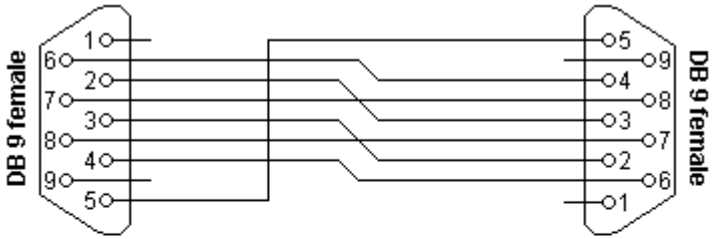

| Connector 1 | Connector 2 | Function |
|---|---|---|
| 2 | 3 | Rx ← Tx |
| 3 | 2 | Tx → Rx |
| 5 | 5 | Signal ground |

### Null modem with loop back handshaking



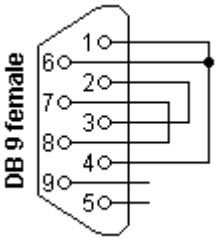

| Connector 1 | Connector 2 | Function |
|---|---|---|
| 2 | 3 | Rx ← Tx |
| 3 | 2 | Tx → Rx |
| 5 | 5 | Signal ground |
| 1 + 4 + 6 | - | DTR → CD + DSR |
| - | 1 + 4 + 6 | DTR → CD + DSR |
| 7 + 8 | - | RTS → CTS |
| - | 7 + 8 | RTS → CTS |

### Null modem with full handshaking




| Connector 1 | Connector 2 | Function |
|---|---|---|
| 2 | 3 | Rx ← Tx |
| 3 | 2 | Tx → Rx |
| 4 | 6 | DTR → DSR |
| 5 | 5 | Signal ground |
| 6 | 4 | DSR ← DTR |
| 7 | 8 | RTS → CTS |
| 8 | 7 | CTS ← RTS |

### RS232 loopback test plug for terminal emulation software




| DB9 | DB25 | Function |
|---|---|---|
| 1 + 4 + 6 | 6 + 8 + 20 | DTR → CD + DSR |
| 2 + 3 | 2 + 3 | Tx → Rx |
| 7 + 8 | 4 + 5 | RTS → CTS |

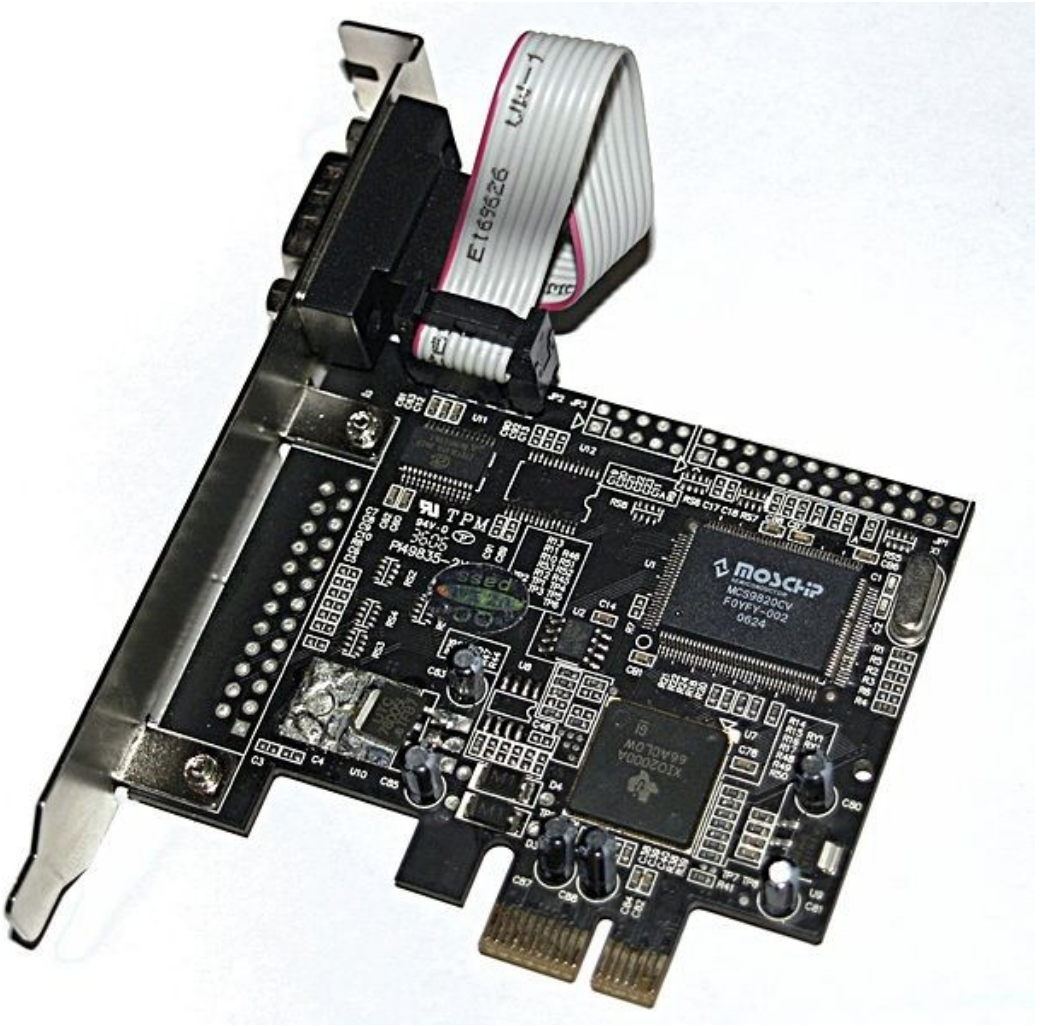
# Weaknesses

- Common ground
- Requirement for positive and negative supplies
- Low transfer rate
- Point-to-point only communication
- Asymmetrical definitions of DTE and DCE
- Many unnecessary communication options
- No method for sending power to a device
- Large recommended connector

# Strengths

- RS-232 is being superseded by USB for local communications but ...
  - USB is more complex (physical layer + protocol + driver) and has no direct analog to the terminal programs for 'raw' communication
  - Lots of communication protocols can be easily implemented over RS232 (by software or hardware)
  - Simplicity makes it good for industrial applications where speed is not critical but distance is long
  - It is available in most of microcontrollers
  - Lot od cheap RS232<->USB converters

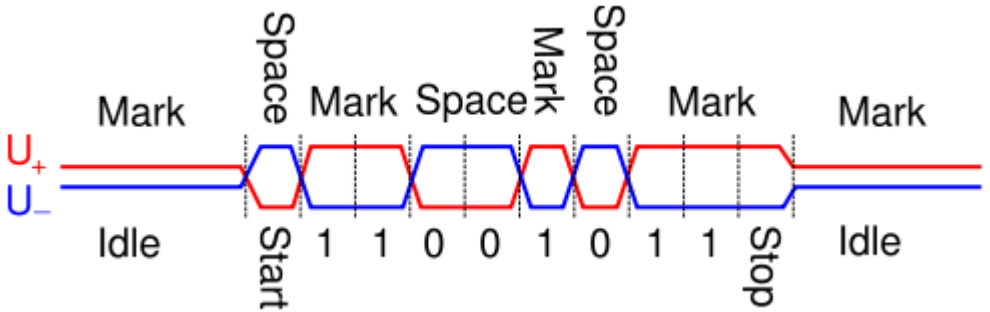
# RS232 Forever ...




PCI-Express Card with 1 RS232C Port

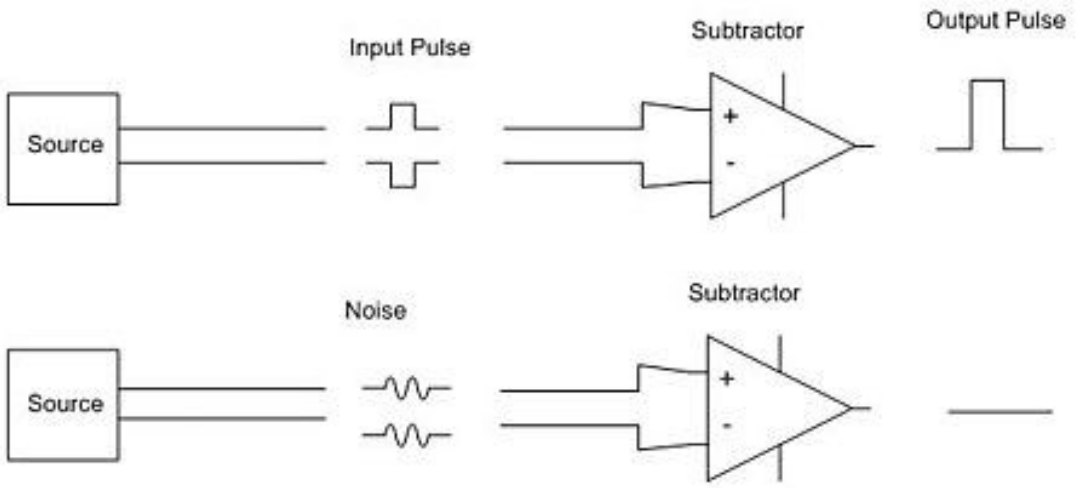
# Related Standards

- RS-485
  - a two-wire differential signaling (U+, U-)
  - **half-duplex**
  - multipoint serial connection (inexpensive local networks)
  - speeds up to 35 Mbit/s (10 m) and 100 kbit/s (1200 m)



# Differential Signaling

- Tolerance of ground offsets
- Suitability for use with low-voltage electronics
- Resistance to electromagnetic interference
- RS-485, PCI Express, USB, ...

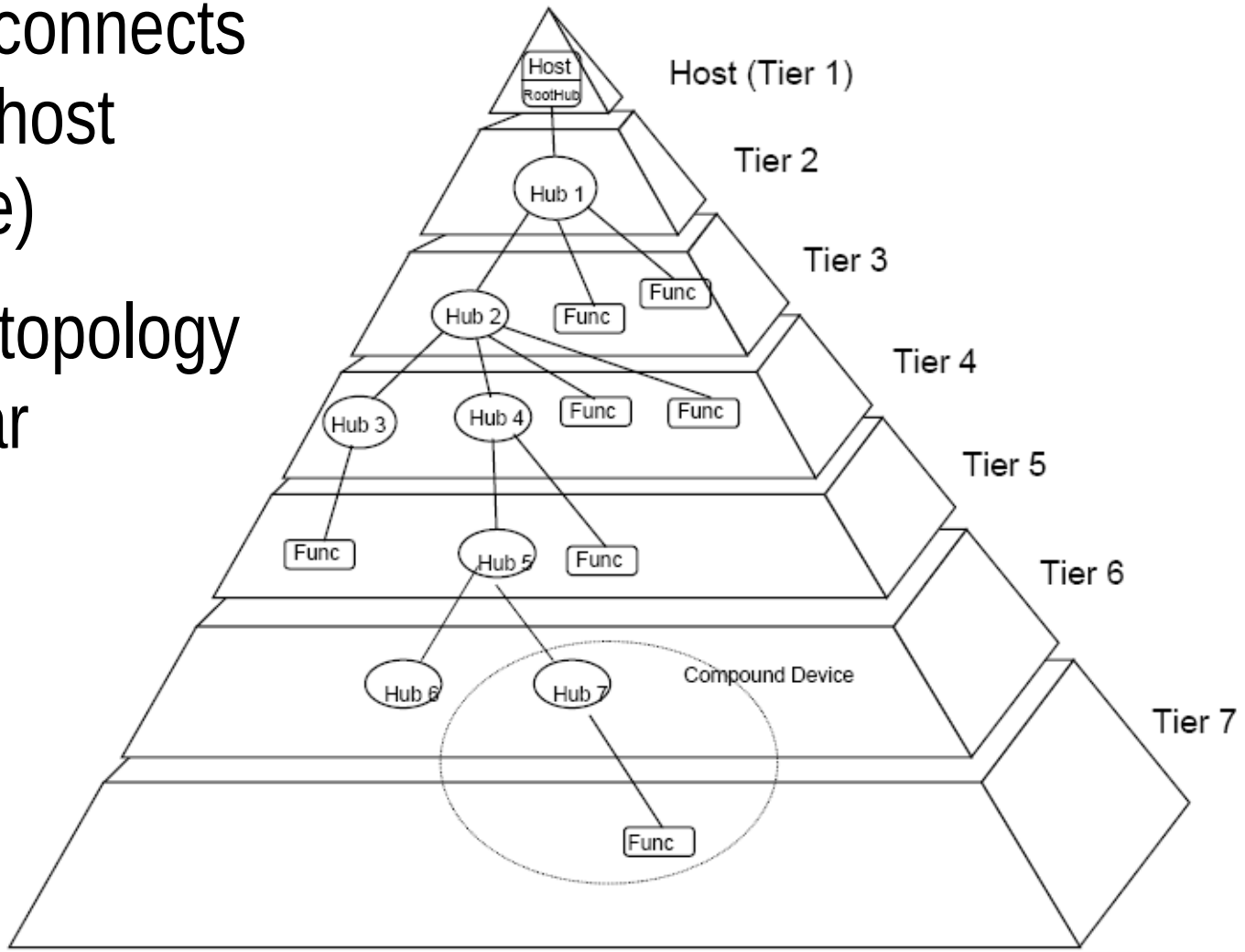


# USB

- Universal Serial Bus (USB) was intended to help retire all legacy varieties of serial and parallel ports (1995)
  - single standardized interface socket
  - plug-and-play + hot swapping
  - providing power to devices (2.5 – 4.5 – 15W)
  - allowing many standard devices to be used without requiring manufacturer specific
- No direct access to physical layer
- Complex software+hardware
  - error correction, protocols, device classes, etc.

# USB Bus Topology

- USB system: host+interconnect+devices
- USB always connects devices with host (master-slave)
- Interconnect topology is a tiered star

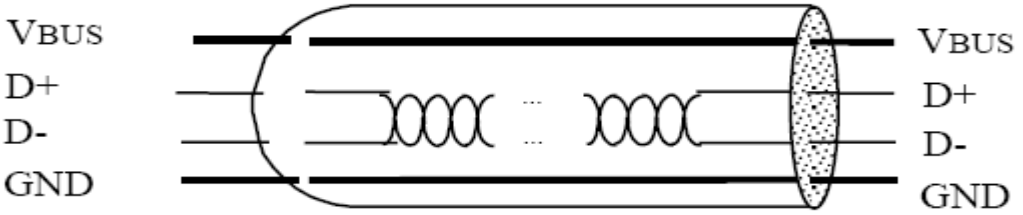


# Speed

- Several data rates:
  - USB 1.x low-speed 1.5 Mb/s
  - USB 1.x full-speed 12 Mb/s
  - USB 2.0 high-speed 480 Mb/s
  - USB 3.0 Super-Speed
    - 4.8 Gbit/s (600 MB/s)
    - Full-duplex with additional twisted pair (10 pins in total)
    - USB 3.1 SuperSpeed+ 10 Gbit/s
- The actual throughput attained with real devices is about ⅔ of the maximum theoretical bulk data transfer rate

# Electrical Interface (1.x, 2.0)

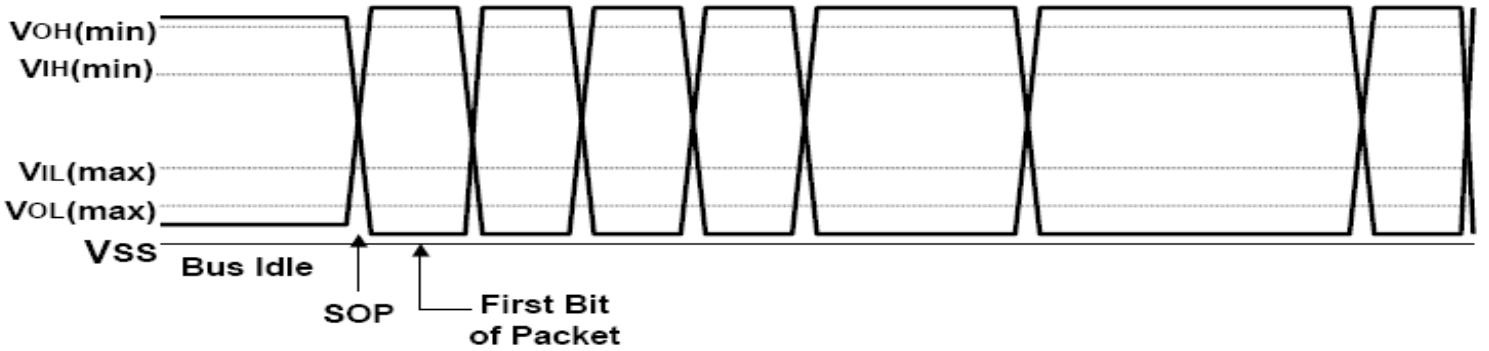
- Four-wire cable, differential signaling, half-duplex,
- Twisted pair data cable with 90Ω ±15% impedance

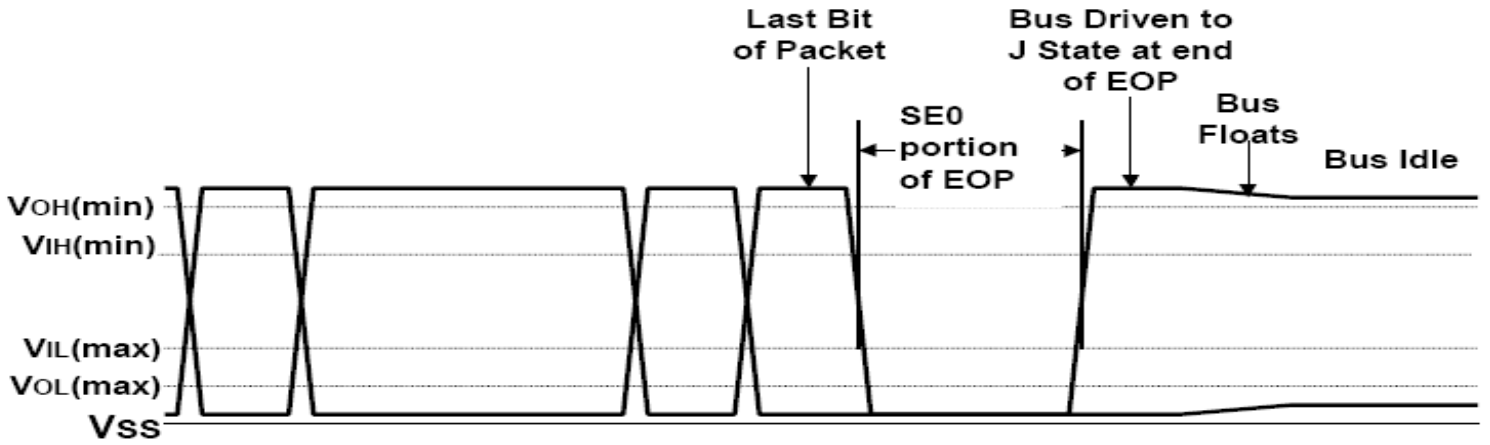



- 0.0–0.3V Low and 2.8–3.6V High (Low/Full Speed)
- ±400mV in High Speed (+ protocol to negotiate HS)
- Clock tolerance:
  - 480.00 Mbit/s ±0.05%
  - 12.000 Mbit/s ±0.25%
  - 1.50 Mbit/s ±1.5%

# Data Signaling (Low/Full)

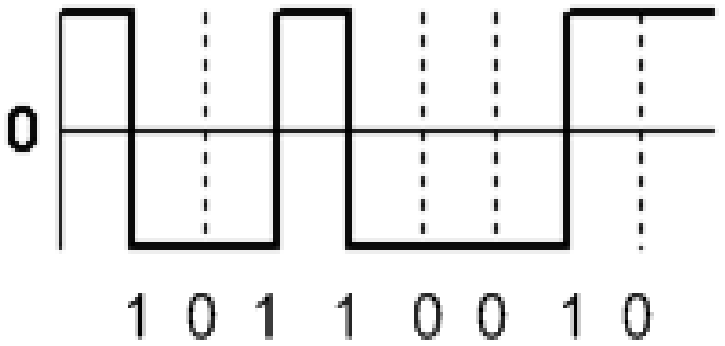
- Data transmission is organized in packets
- NRZI (non-return-to-zero inverted) encoding



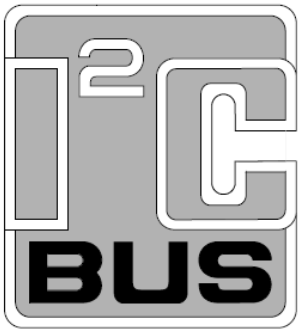
# NRZI Encoding

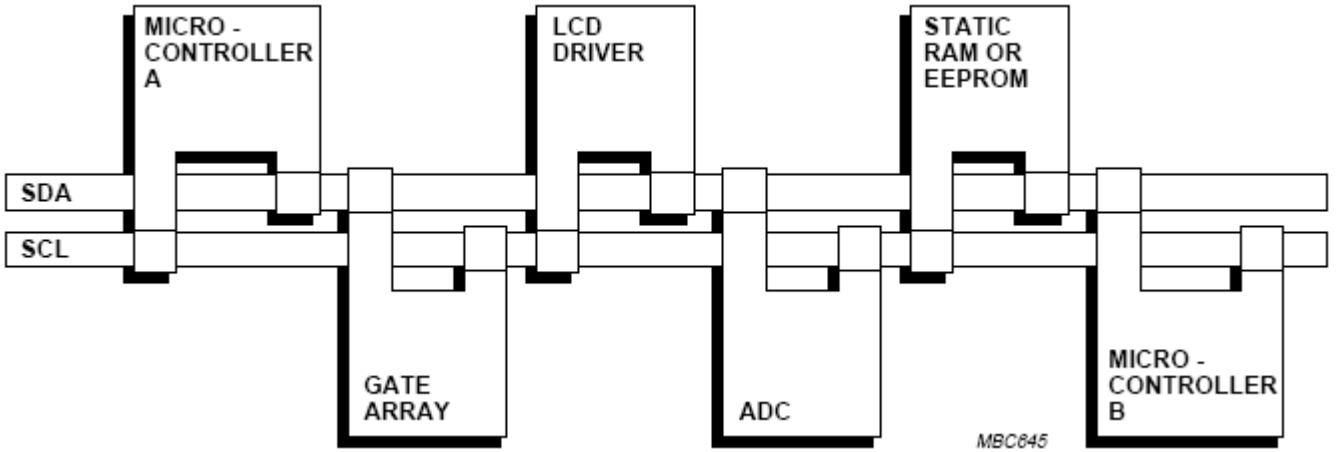
- Binary code in which "1s" and "0s" are represented with no other neutral or rest condition
- Not a self-synchronizing code, synchronization technique must be used to avoid bit slip
- "1" is represented by a transition
- "0" has no transition



# I²C - Inter-Integrated Circuit

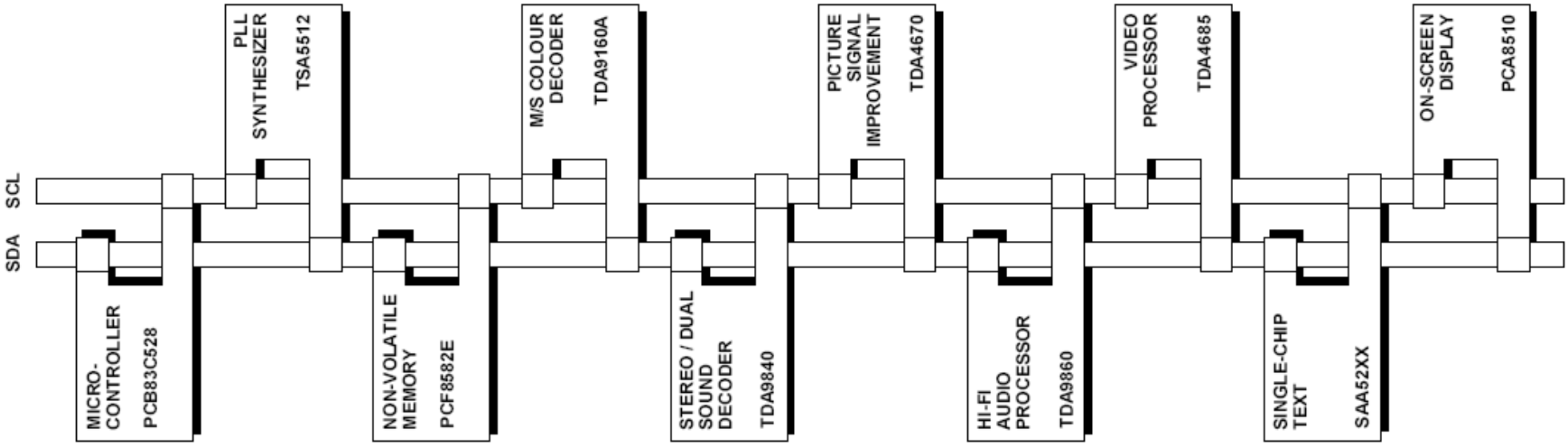
- I²C (*I-squared-C*) – Philips (NXP), early 80's (XX)
  - communication between integrated circuits
  - only two wires (+ common ground) → small PCB's
  - multi-master/multi-slave, bidirectional, 8-bit
  - no strict baud rate requirements (master sends the clock)
  - 7bit or 10bit unique device addressing
  - simple – hardware or (easy) software implementation



# Highly-integrated TV set

- Example of TV set architecture with I2C (historical)

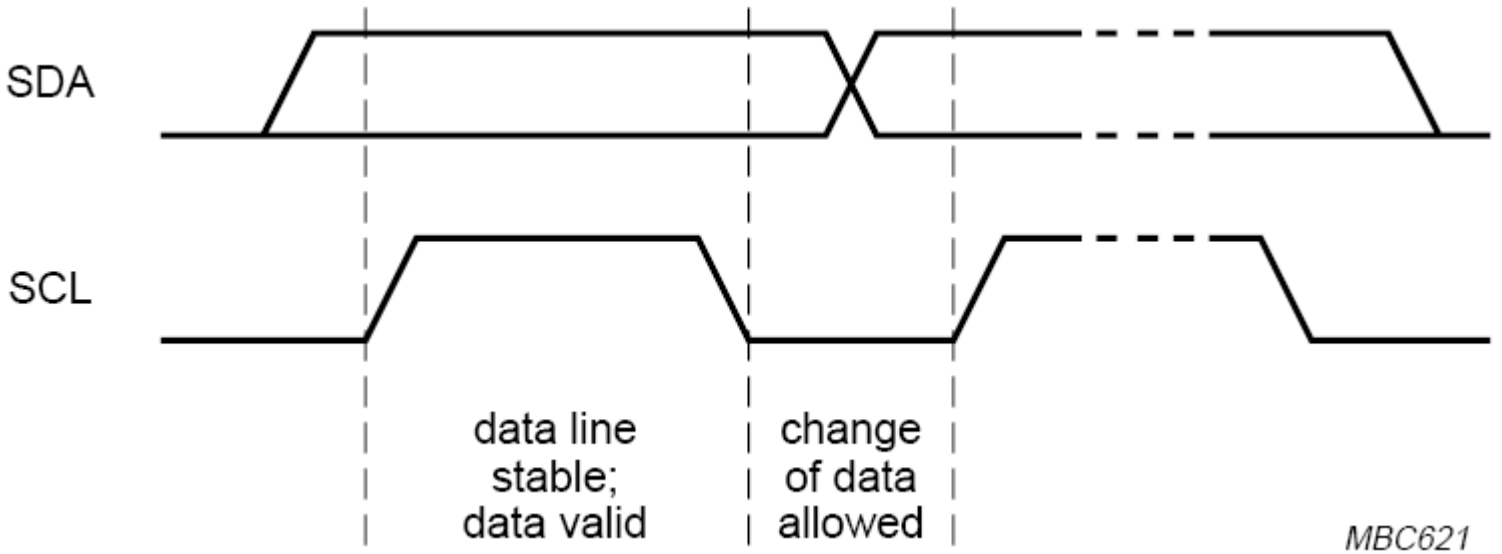


# Characteristics

- Speed:
  - 100 kbps (standard mode)
  - 400 kbps (fast mode)
  - 3.4 Mbps (high-speed mode)
- Two-wired bus (+ ground)
  - serial data line (SDA)
  - serial clock line (SCL)
- Voltage levels
  - not fixed, depends on supply level of voltage
  - HIGH → 1
  - LOW → 0

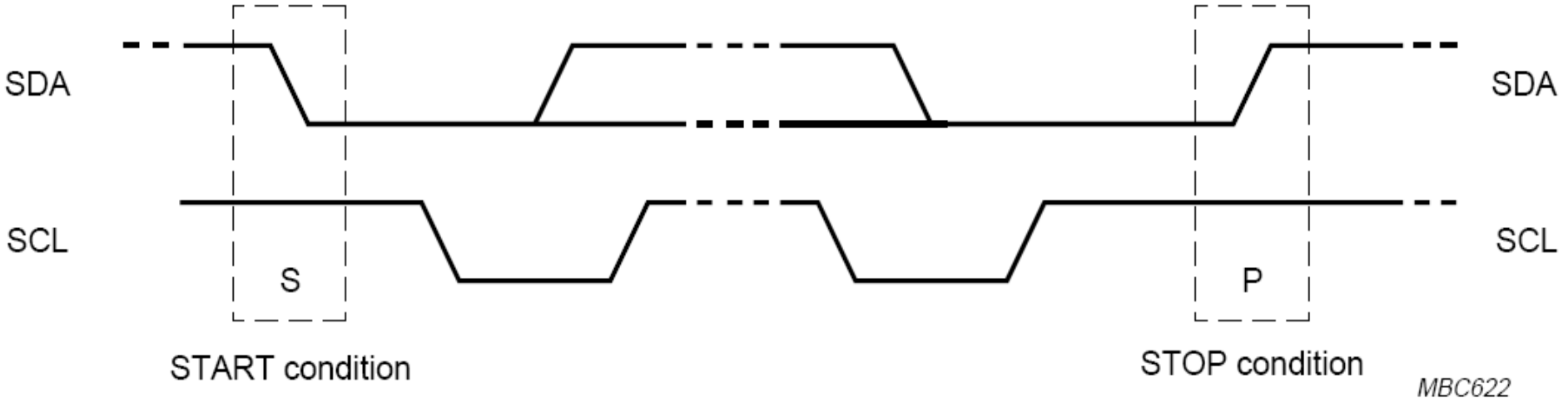
# Bit Transfer

- Bit transfer is level triggered
  - SCL = 0↑1 → SDA = valid data
  - one clock pulse per data bit
  - stable data during high clocks
  - data change during low clocks



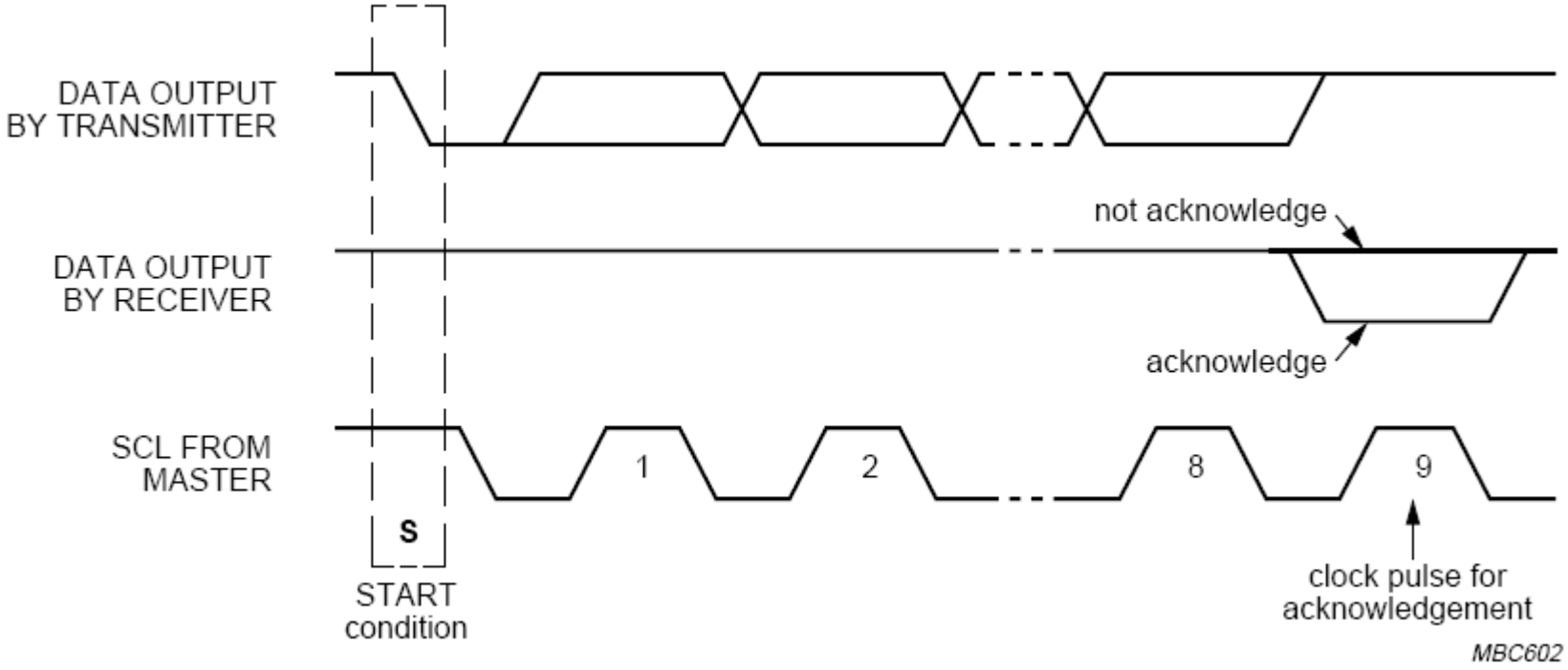
# Start and Stop

- SCL & SDA are high when idle
  - open-drain + pull-up resistor
- Start condition (S)
  - SDA 1↓0 transition when SCL = 1
- Stop condition (P)
  - SDA 0↑1 transition when SCL = 1



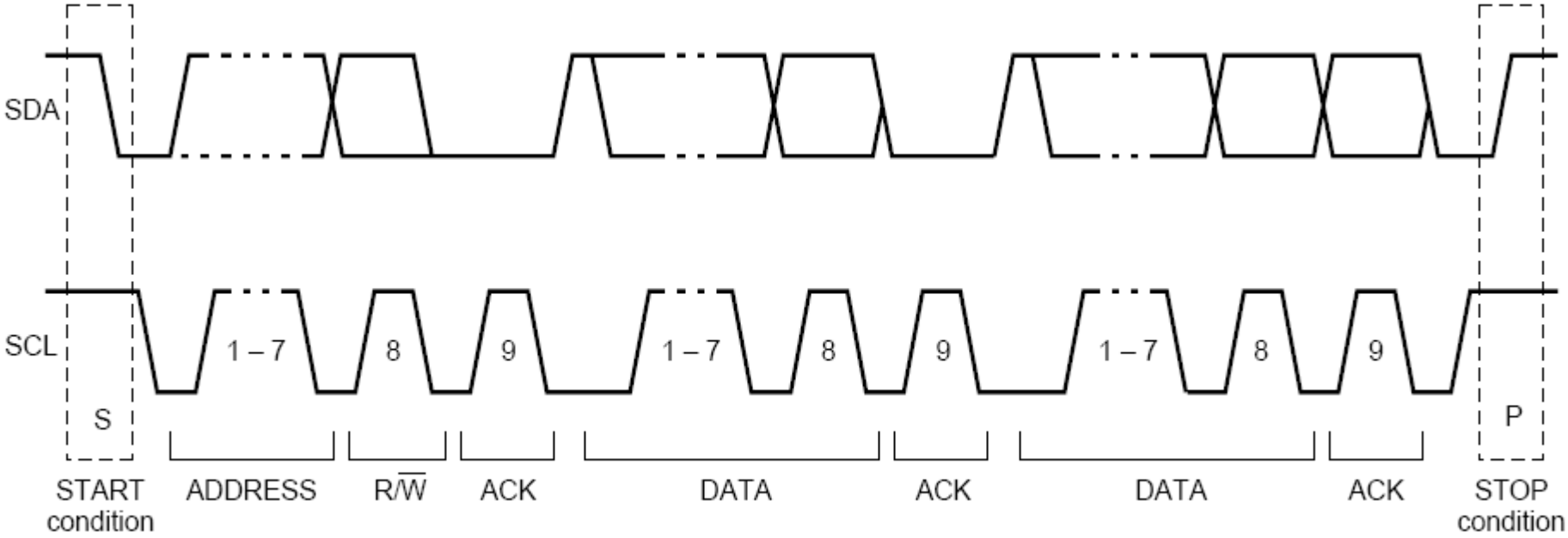
# Acknowledge

- All transfers are initiated by master
- Slave must respond with acknowledge



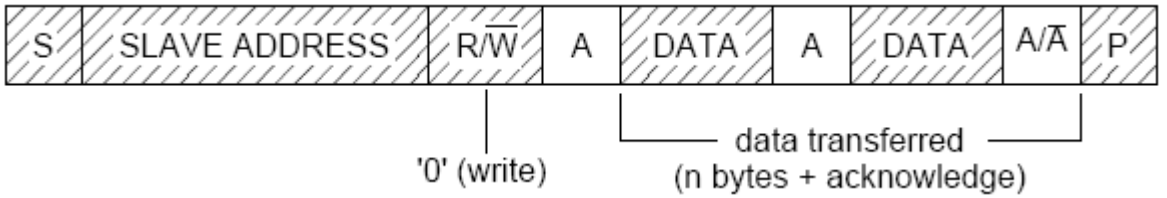
# Data Frame

- Start + Slave address + Direction Command
- Acknowledge from slave
- Data (master → slave or slave → master) + Ack.
- Stop

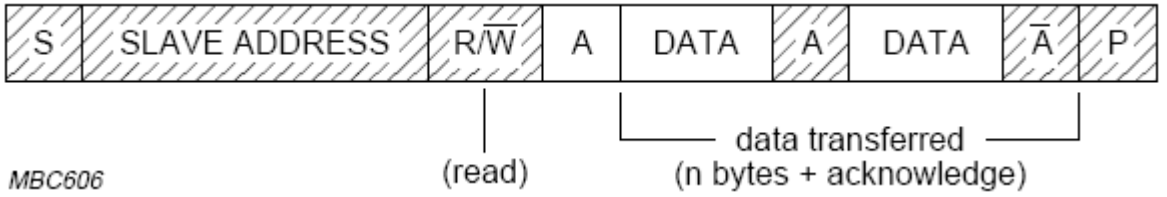


# Frame Formats

## Master transmitter



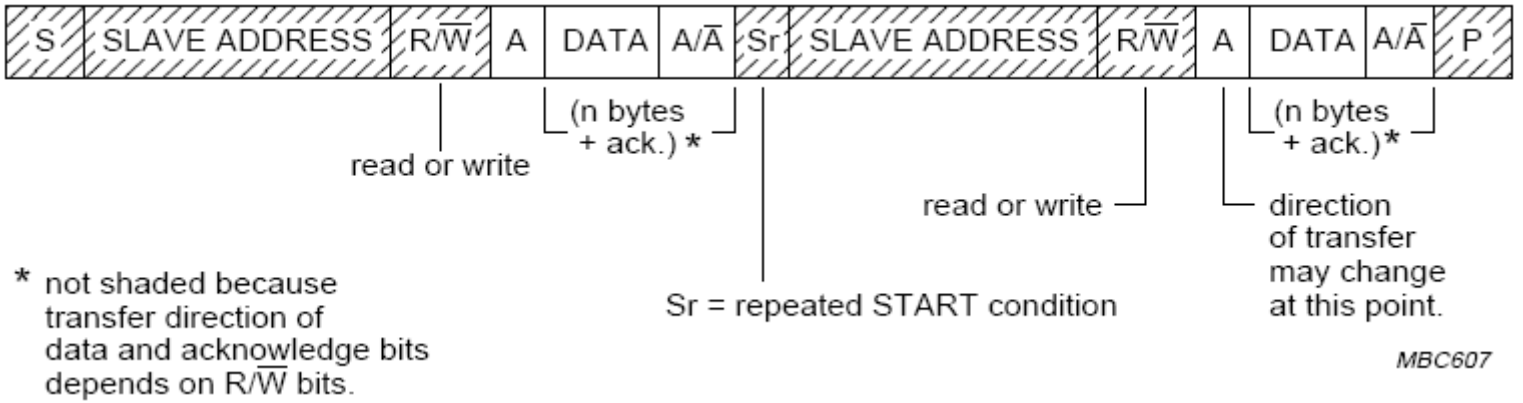

## Master receiver






A = acknowledge (SDA LOW)
Ā = not acknowledge (SDA HIGH)
S = START condition
P = STOP condition

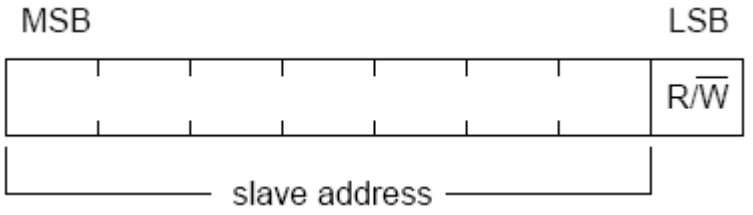
## Combined



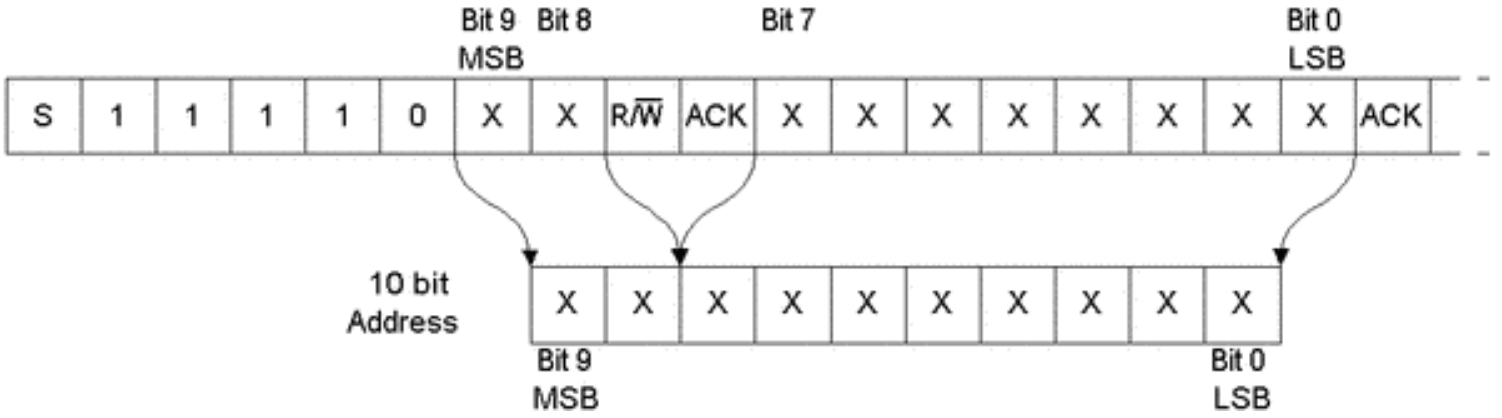

* not shaded because transfer direction of data and acknowledge bits depends on R/W bits.

# Addressing

- 7-bit addressing
  - max. 112 slaves
  - 7 addresses reserved
    - e.g. broadcast, start-byte, 10-bit extension




- 10-bit addressing
  - 11110 prefix + 10 address bit (R/W in the middle)
  - 1024 new addresses



# Closing Remarks

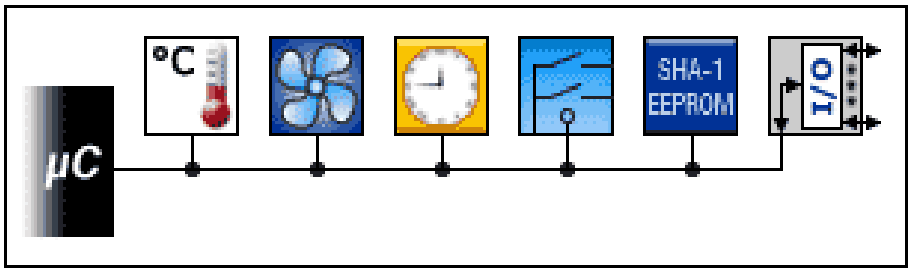
- Elegance of I²C: simplicity & effectiveness
  - shared bus, reasonable speed, two wires (pins), hotplug
  - flexible timing: no strict baud rate, clock stretching
  - arbitration & collision detection in multi-master mode
- One of most common communication standard for microcontrollers & integrated circuits
- No licensing fees for I²C implementation, but required to obtain I²C slave addresses
- Limited Address Space:
  - devices with configurable address
    - higher bits indicate model, lower bits - type

# Derivatives

- System Management Bus (SMBus)
  - low-bandwidth devices on a PC motherboard
    - laptop's battery, temperature, fan, or voltage sensors, configuration data of DDR2, ...
- VESA Display Data Channel (DDC)
  - connection between monitor graphics card
    - with VGA, DVI, HDMI connectors
- Two Wire Interface (TWI)
  - $I^2C$ implemented in chips from Atmel
- ...

# 1-Wire®

- Registered trademark of Dallas Semiconductor Corp. (now Maxim-Dallas)
  - Signaling and power using 1 wire (+ ground)
  - Low-speed ( < 1 kbps)
  - Long range (up tp 500m)
  - Low cost
  - Easy implementation




- Typically used to communicate with small inexpensive devices such as digital thermometers and weather instruments

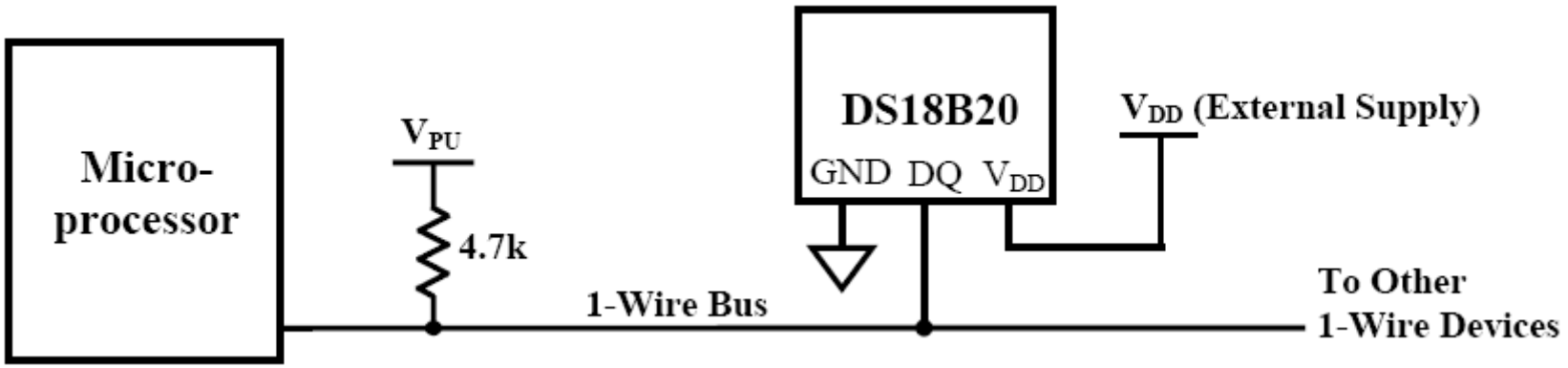
# Available Devices

- Memory: EPROM, SRAM, EEPROM, ROM, NV
- Temperature Sensors and Switches
- 1-Wire Interfaces
- A-to-D Converters
- Timekeeping and Real-Time Clocks
- Battery Protectors, Selectors, and Monitors
- ...

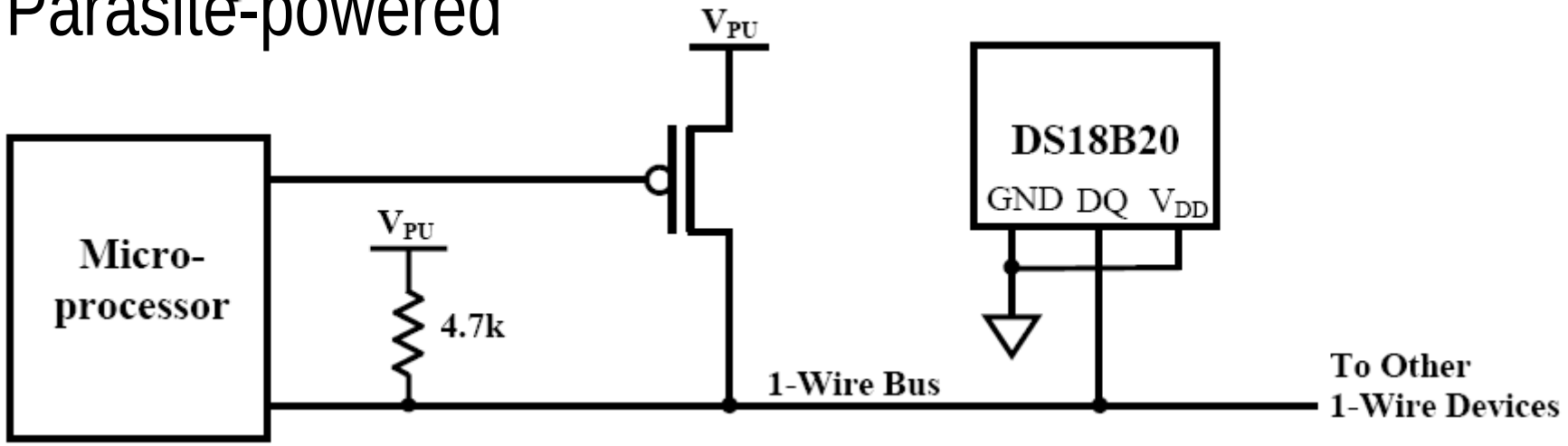
# Powering 1-Wire Devices

- With external supply

DS18B20 - Programmable Resolution 1-Wire Digital Thermometer, –55°C to +125°C, ±0.5°C accuracy and user-definable nonvolatile alarm settings




- Parasite-powered



# Device Addressing

- There is always only one bus master
- Many slave devices can share the same bus
- Each device has a unique 64-bit serial number

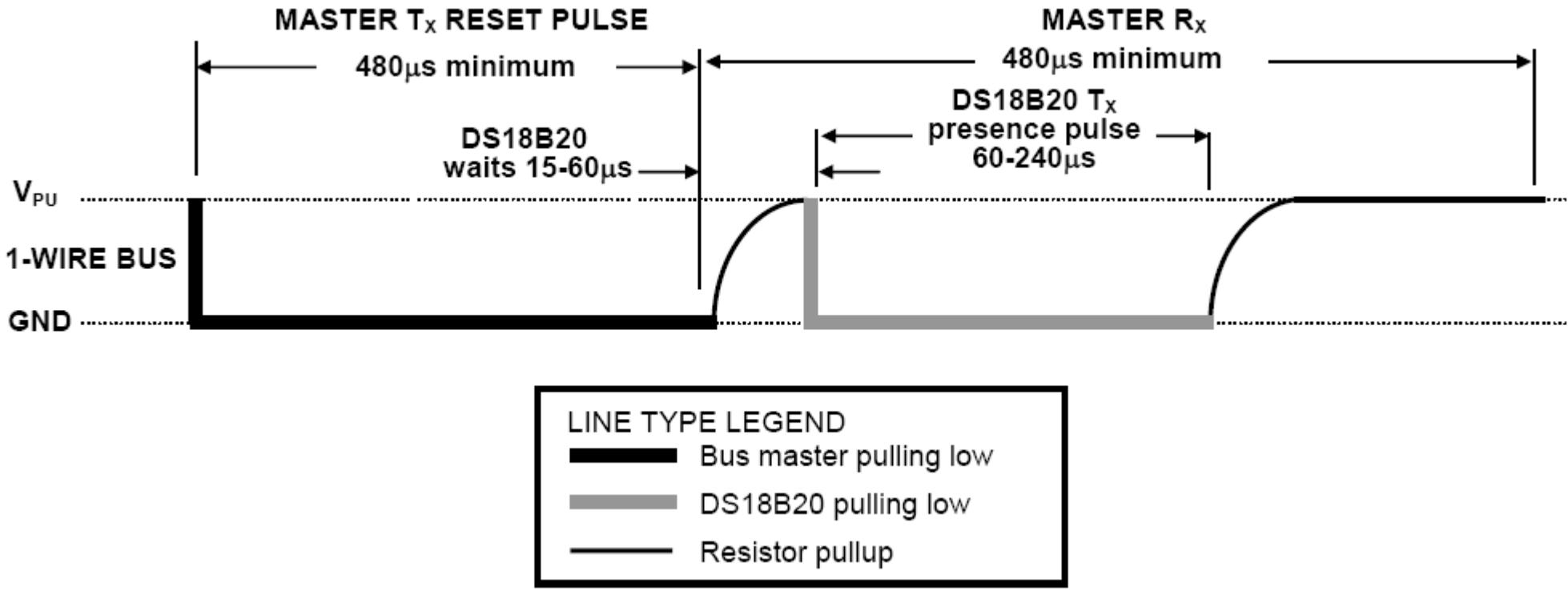

- There are broadcast commands and commands addressed to particular devices
- The bus also has an algorithm to recover the address of every device on the bus

# Transaction Sequence

- Step 1. Initialization
  - reset pulse
- Step 2. ROM Command
  (followed by any required data exchange)
  - Search, Read, Match, Skip, Alarm, ...
- Step 3. Function Command
  (followed by any required data exchange)
  - Read, Write, ReadPowerSupply, ...

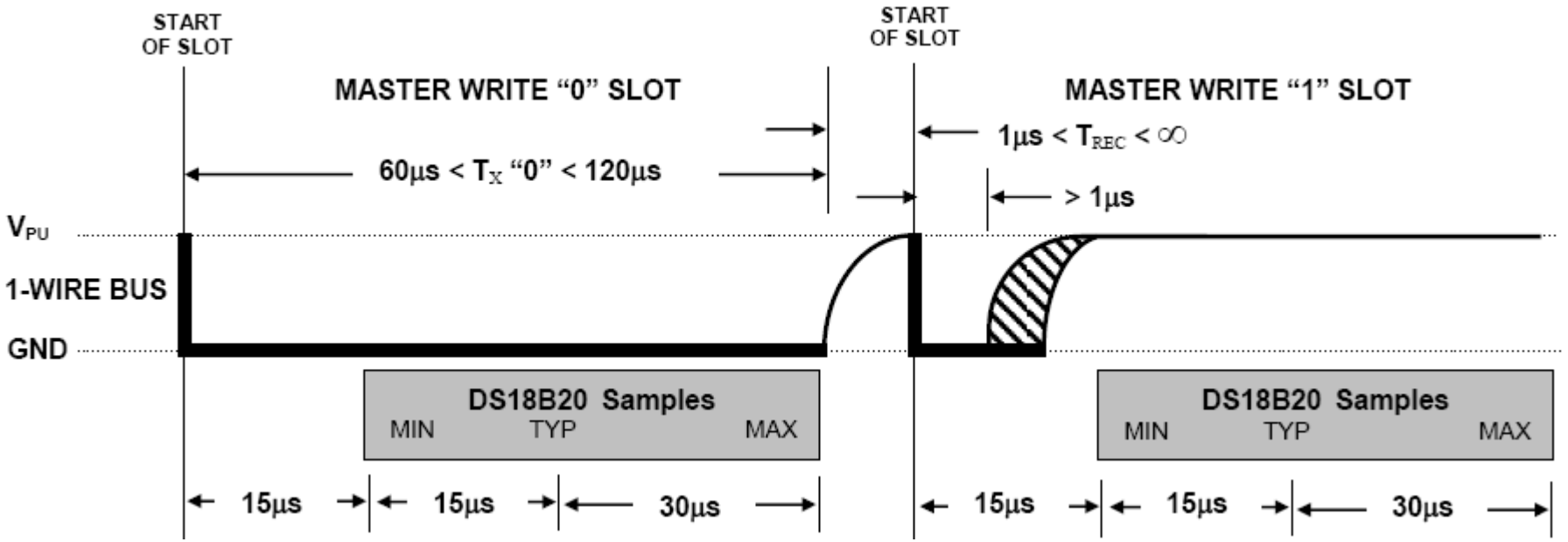
# Initialization

- Reset pulse transmitted by the bus master followed by presence pulse(s) transmitted by the slave(s)
- The presence pulse lets the bus master know that slave devices are on the bus



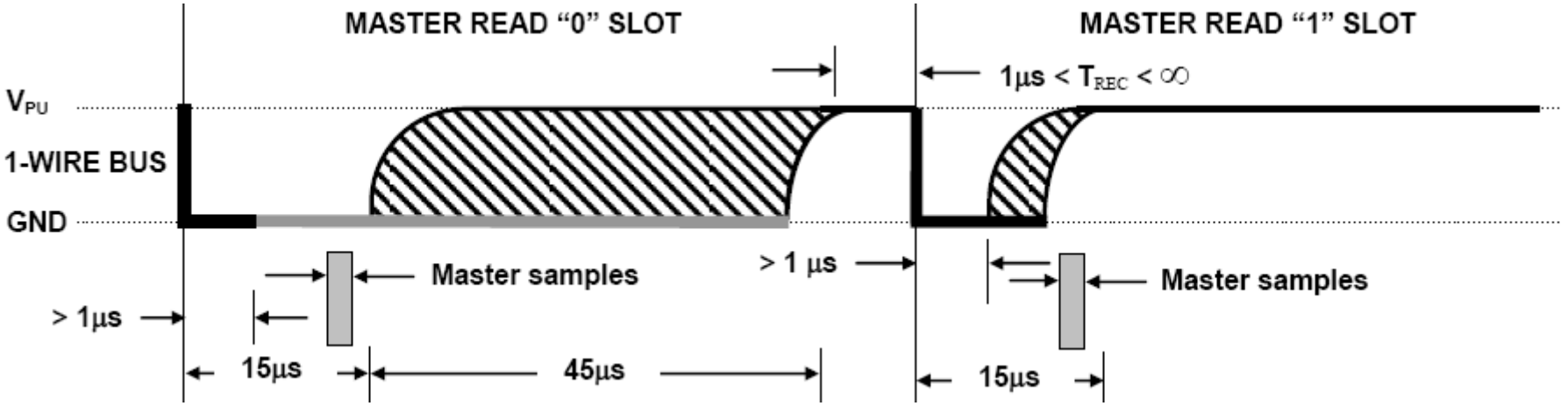
# Master Write Bit

- Write time slots are initiated by the master pulling the 1-Wire bus low
  - Write 0 time slot: after pulling the bus low, the bus master continues to hold the bus low for the time slot
  - Write 1 time slot: after pulling the bus low, the bus master releases the bus (pullup will pull the bus high)



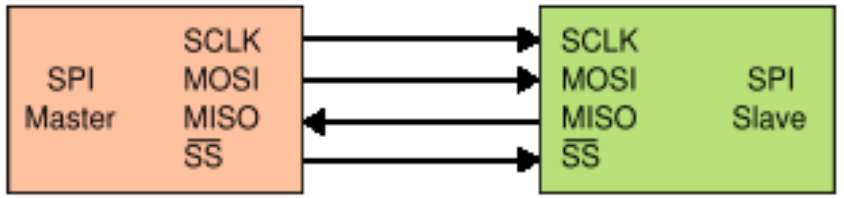
# Master Read Bit

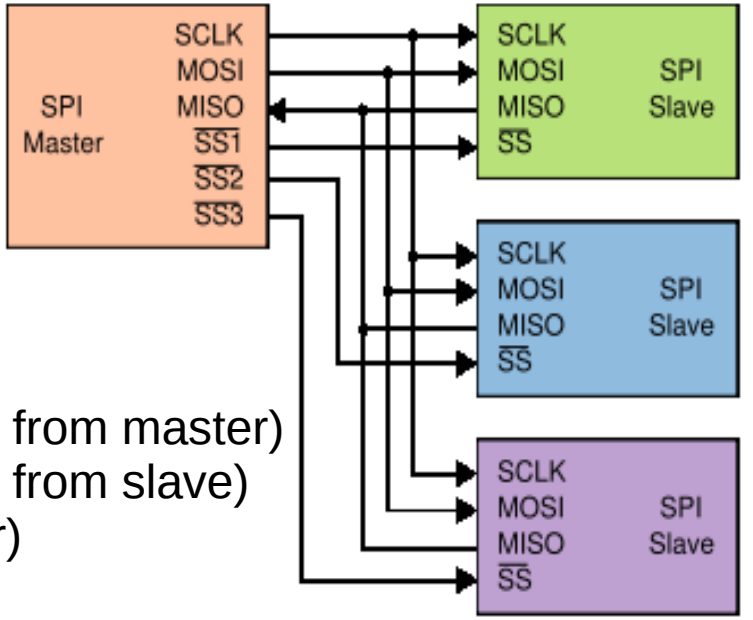
- Slaves can only transmit data to the master after the master issues read-related commands
  - After the master initiates the read time slot, the slave begins transmitting a 1 or 0 on bus
  - 1 is transmitted by leaving the bus high and 0 by pulling the bus low



# SPI

- Serial Peripheral Interface Bus (SPI) - Motorola
  - four wire" serial bus
  - synchronous serial data link (clock signal)
  - full duplex mode
  - multiple slaves are allowed with individual slave select (chip select)

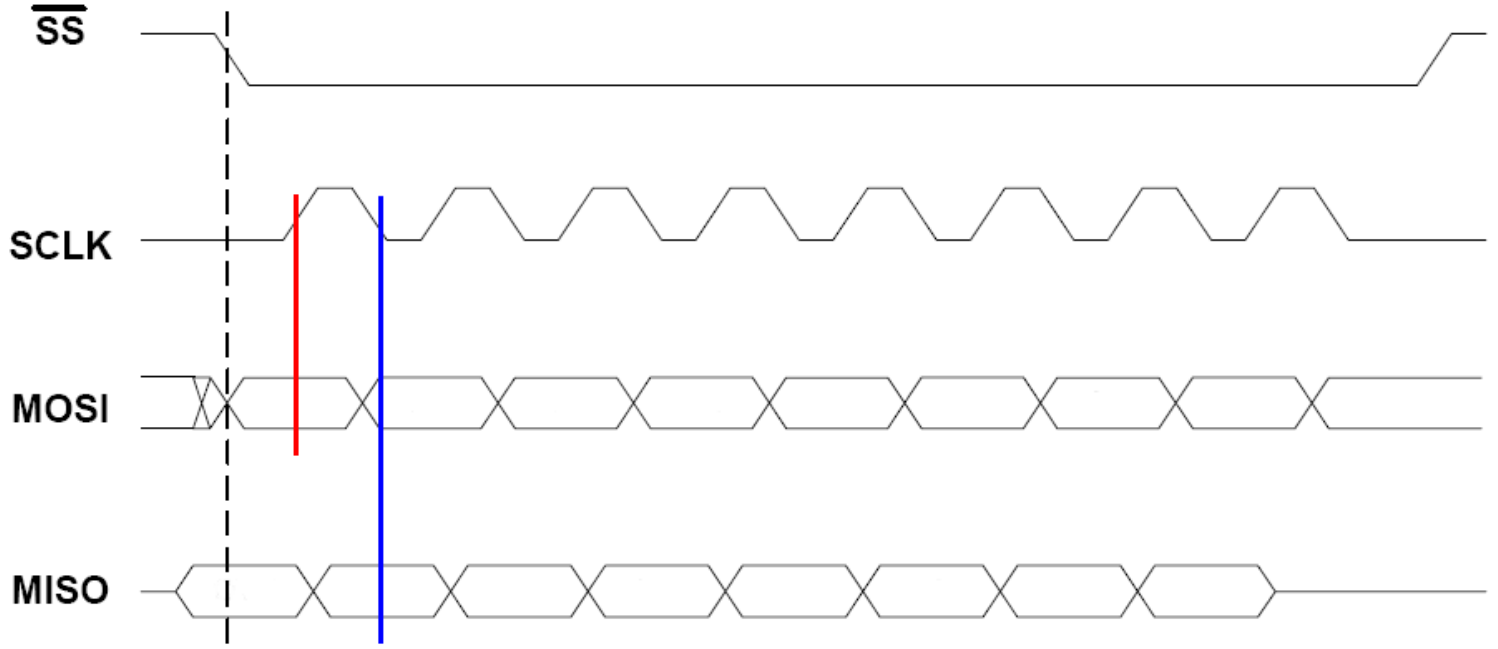



SCLK — Serial Clock (output from master)
MOSI/SIMO — Master Output, Slave Input (output from master)
MISO/SOMI — Master Input, Slave Output (output from slave)
SS — Slave Select (active low; output from master)

# Data Transmission

- During each SPI clock cycle, a full duplex data transmission occurs:
  - the master sends a bit on the MOSI line; the slave reads it from that same line (rising SCLK)
  - the slave sends a bit on the MISO line; the master reads it from that same line (falling SCLK)



# Closing Remarks

- Advantages
  - Full duplex communication
  - Very fast - higher throughput than $I^2C$
  - Arbitrary choice of message size and content
  - Extremely simple hardware interfacing
  - No arbitration or associated failure modes
  - No synchronization problems
- Disadvantages
  - more lines/pins than $I^2C$
  - No hardware flow control, no slave acknowledgment
  - limited to a single slave
  - short distances