



Protocolli di Comunicazione

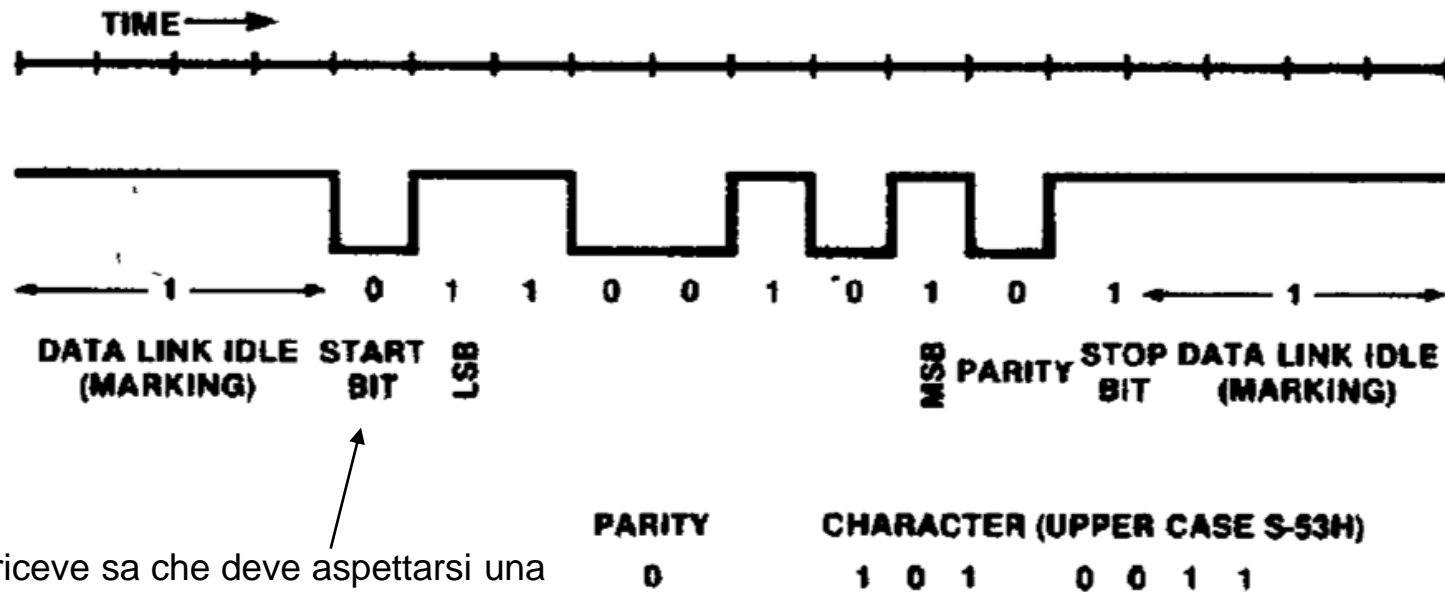
Elettronica dei Sistemi Digitali L-A
Università di Bologna, Cesena

Aldo Romani

A.a. 2005-2006

UART

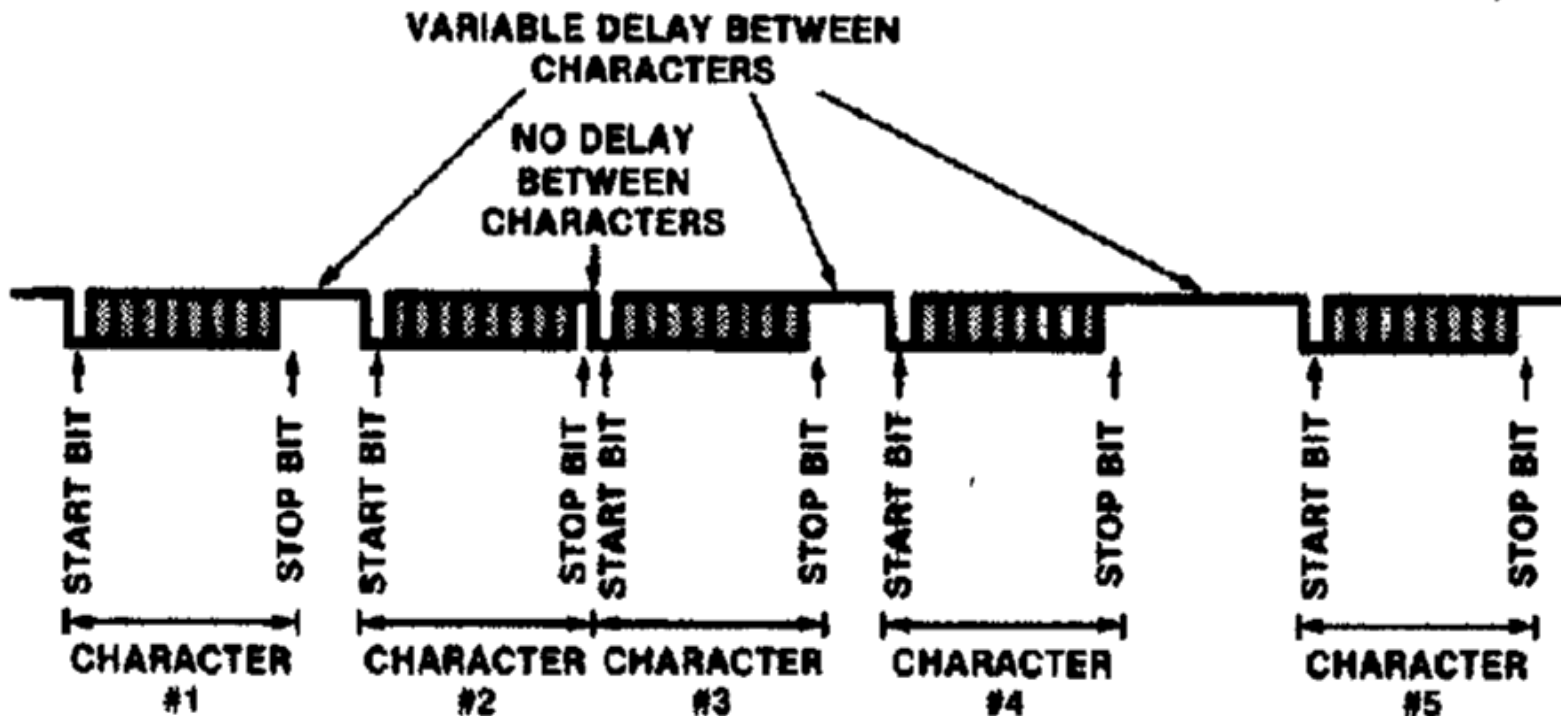
- UART Universal Asynchronous Receiver/Transmitter
- Clock implicit e trasmissione di “trame” di bit (R e T hanno il proprio oscillatore, e c'è accordo preventivo su velocità e struttura della trama).
- Linea MARK →
START BIT [1bit] → WORD [da 5 a 8 bit] (→ PARITY BIT [PARI/DISPARI])
→ STOP BITS [1-2 bits]



Qui chi riceve sa che deve aspettarsi una sequenza di bit in arrivo, dopo lo START

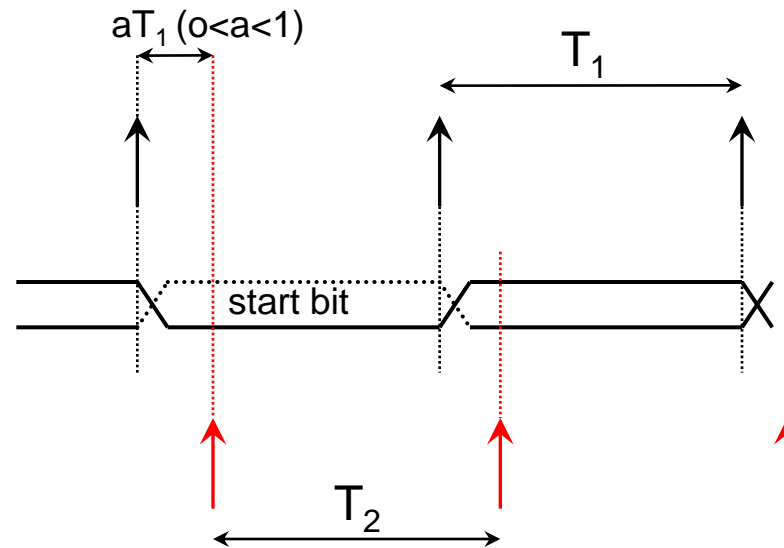
UART

- Trasmissione di trame.
- Clock implicito → 2 oscillatori, possibile disallineamento → dimensione trama limitata



UART

- supponiamo che R e T abbiano oscillatori con periodo T_1 e T_2
- supponiamo che R campioni i segnali trasmessi da T sui fronti di T_2 e che inizialmente siano sfasati di aT_1 rispetto a T_1
- se $T_2 < T_1$ ad ogni bit lo sfasamento tra i due fronti cala di un $\Delta T = T_1 - T_2$
T può trasmettere max N bit:
dove $N \Delta T < aT_1$
- se $T_2 > T_1$ ad ogni bit lo sfasamento tra i due fronti aumenta di un $\Delta T = T_2 - T_1$
T può trasmettere max N bit:
dove $N \Delta T < (1-a)T_1$



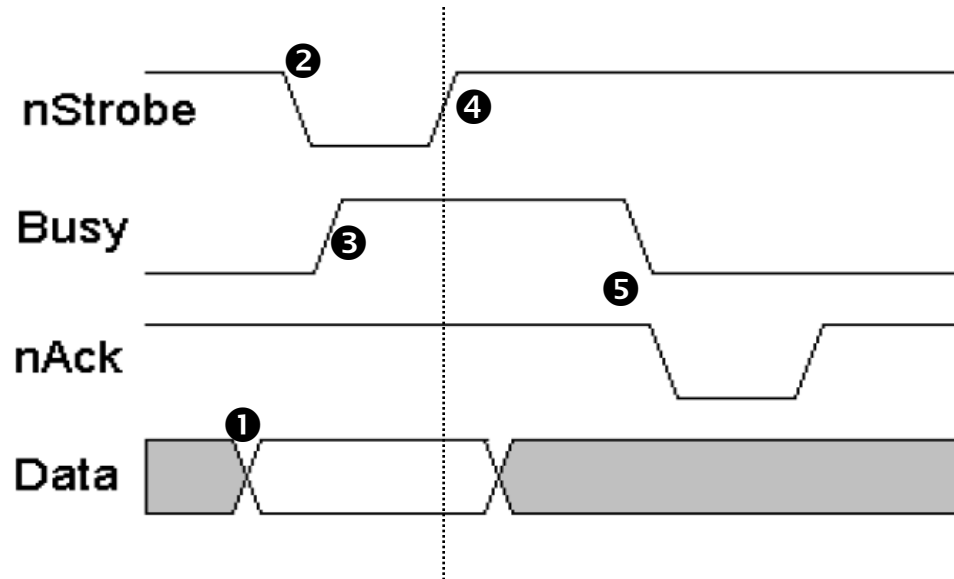
La condizione migliore è quella in cui poniamo $a = \frac{1}{2}$.

In tal caso:

$$N \Delta T < \frac{T_1}{2}$$

(es. a 9600bps, con trame da 10 bit, la differenza max tra i 2 periodi di clock deve essere 5.2us)

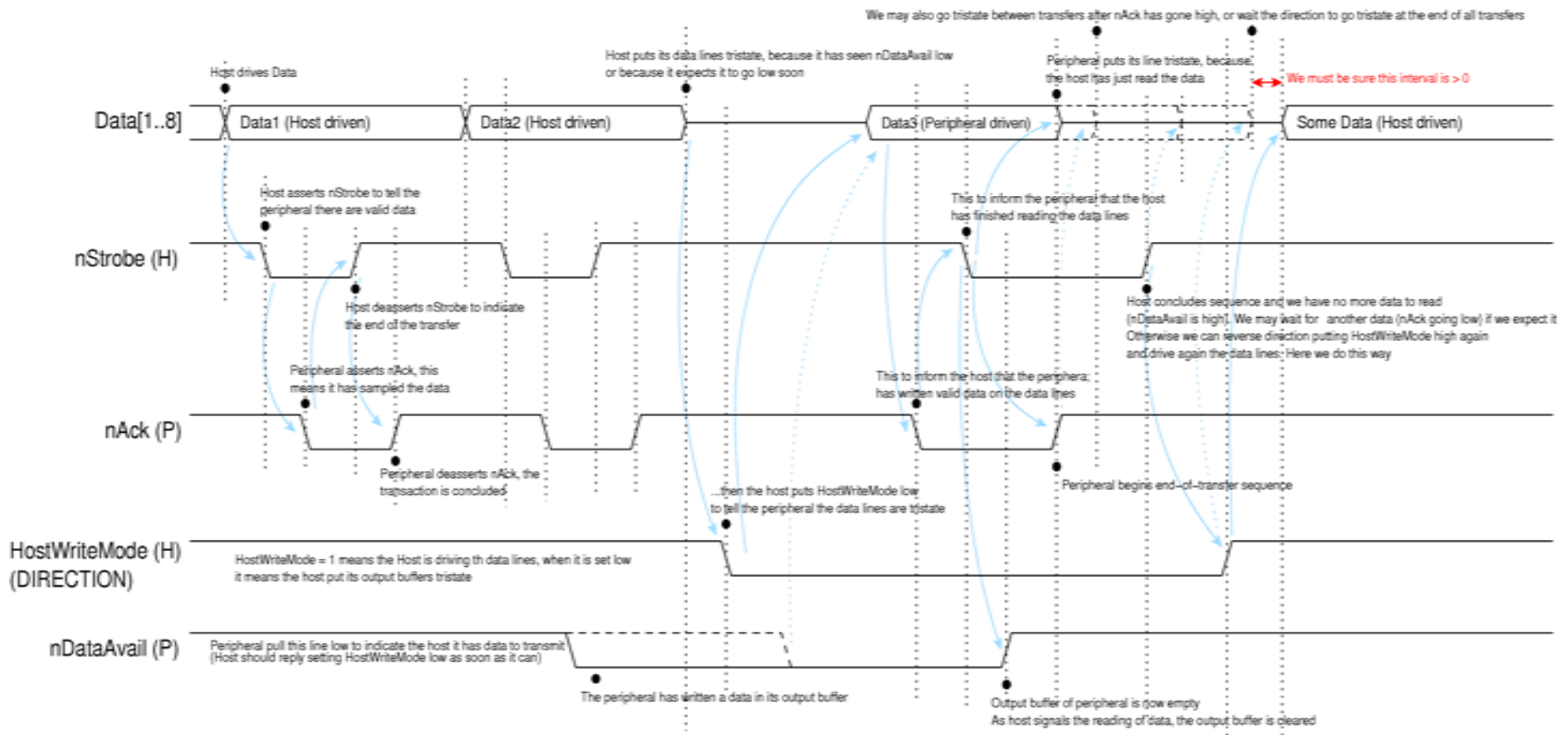
Handshake Centronics



1. Il master scrive il dato,
2. Il master segnala che questo è disponibile con un impulso su nSTROBE di durata pari ad almeno 1us
3. La periferica risponde alzando la linea BUSY
4. Il fronte di salita di nSTROBE causa l'acquisizione del dato da parte della periferica
5. La periferica, non appena termina di processare il dato in ingresso, disattiva BUSY e dà un impulso negativo sulla linea nACK.

Handshake a 4 fasi per Porta Parallela

- A differenza delle comunicazioni UART la porta parallela prevede dei segnali di sincronizzazione.
- Una proposta di protocollo di “Hand-shake”:



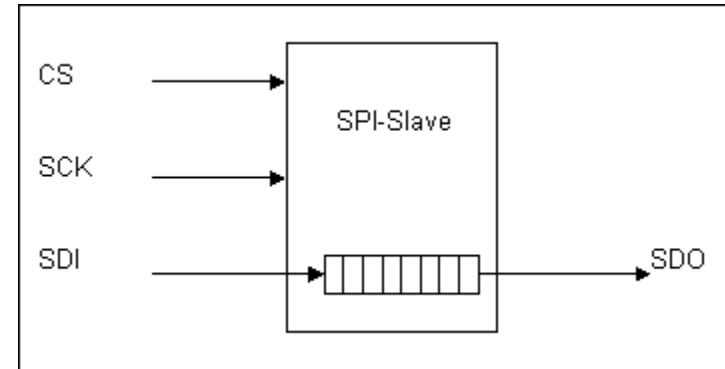
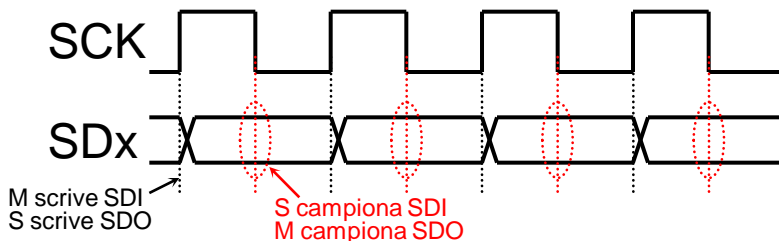
Protocollo SPI

- SPI = Serial Peripheral Interface (Motorola)
 - Seriale
 - Sincrono
 - Master/Slave (1 Master)
 - “Scambio” di dati tra dispositivi
- Protocollo “semplice”.
- Diffuso in MCU (microcontrollori) e componenti (es. convertitori, memorie, etc etc)

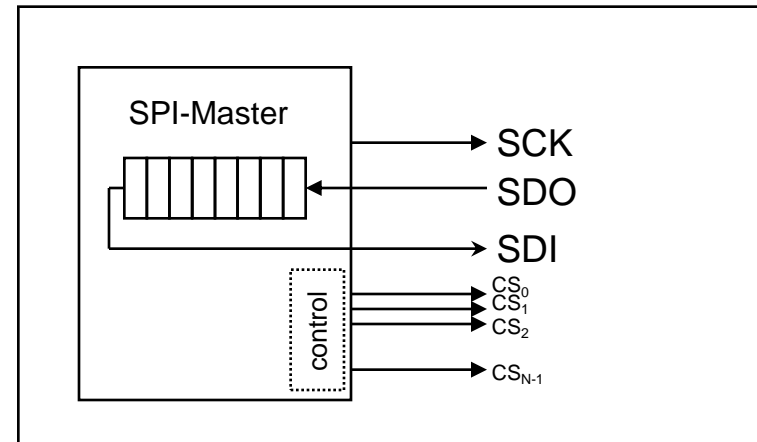
Protocollo SPI

- Il protocollo prevede lo “scambio” di dati tra il Master e uno Slave
- Ogni device ha uno shift register contenente i dati. Il trasferimento prevede lo scambio del contenuto.
 - In ogni trasferimento avvengono gli scambi $M \rightarrow S$ e $S \rightarrow M$
- Il Master indirizza lo slave, e gestisce il trasferimento con il segnale SCK
 - Spesso \rightarrow Dati in uscita vengono scritti in corrispondenza del fronte di salita [discesa] di SCK
 - I dati vengono campionati sul fronte opposto

Spesso \rightarrow
programmabile!!



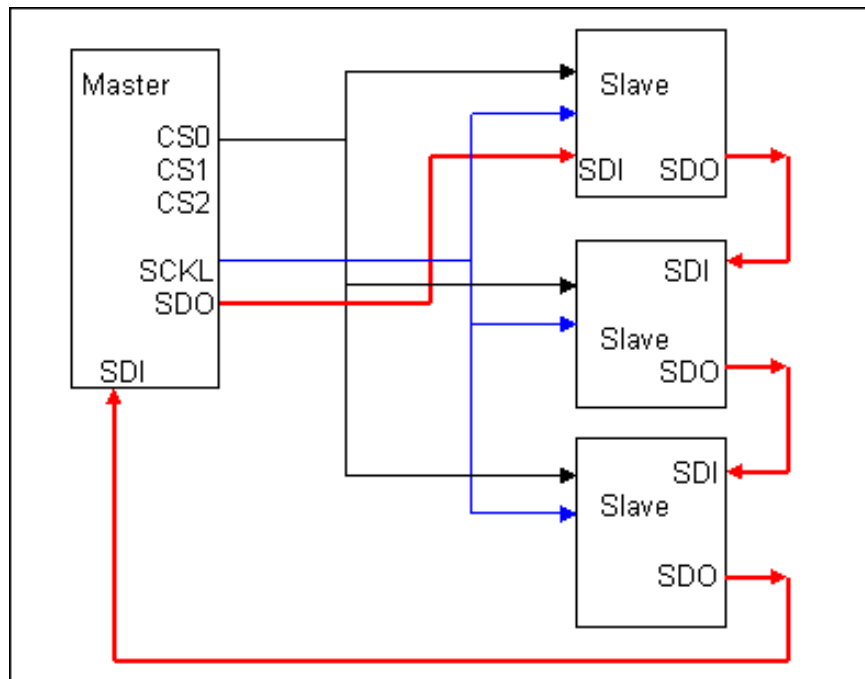
Struttura di uno slave



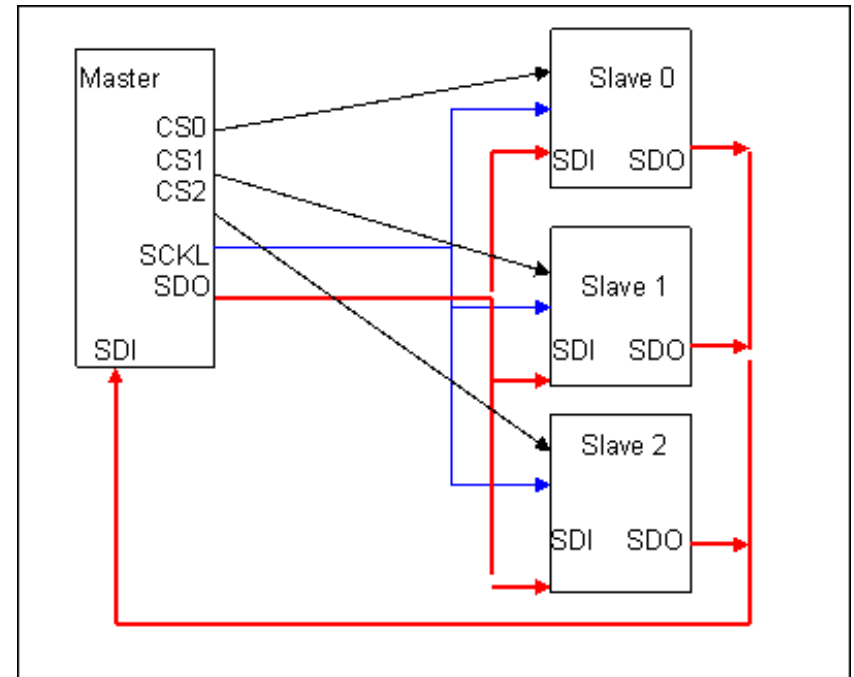
Struttura di un master

Protocollo SPI

- Principali tipi di collegamento tra Master e Slaves



Slaves in cascata



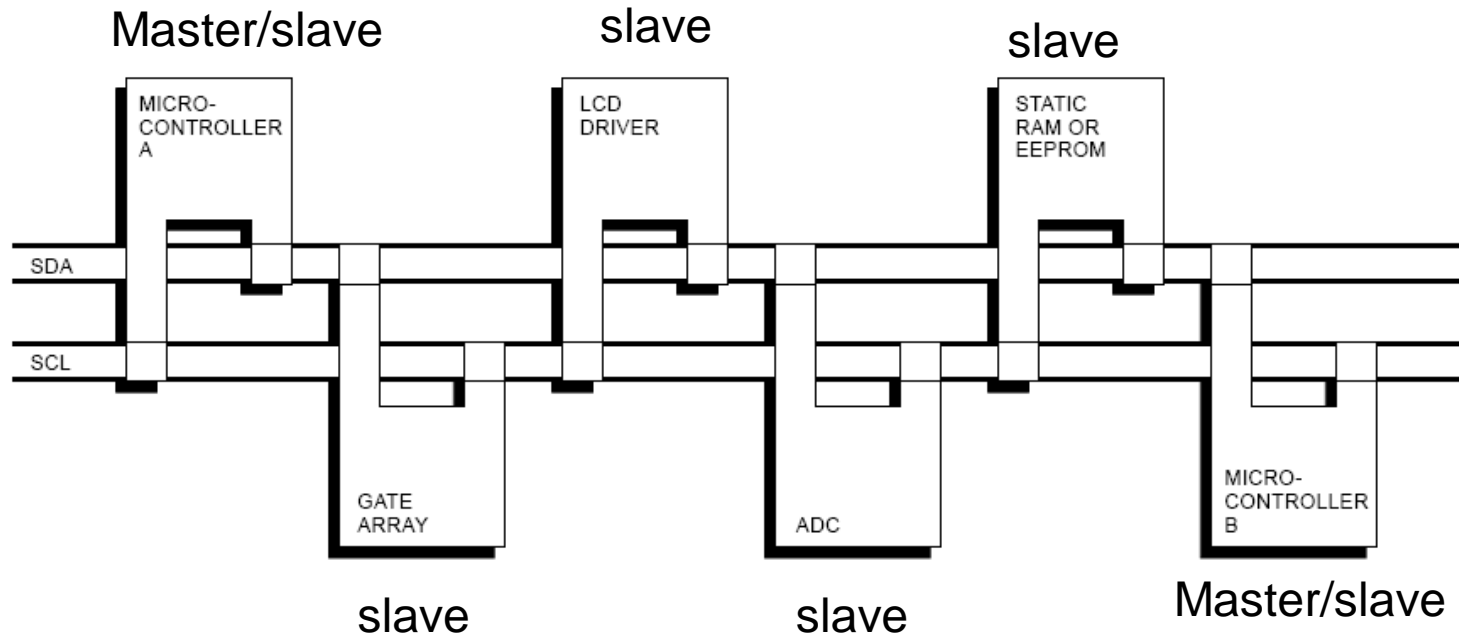
Slaves indipendenti

Protocollo I²C

- Proposto da Philips
- 2 Linee + GND
- Master/Slave (Multi-master) → $C_L < 400\text{pF}$
- Indirizzabilità dei singoli device sul bus
- Trasferimenti dati fino a 100kbps
(400kbps)

Protocollo I²C

- Esempio di configurazione

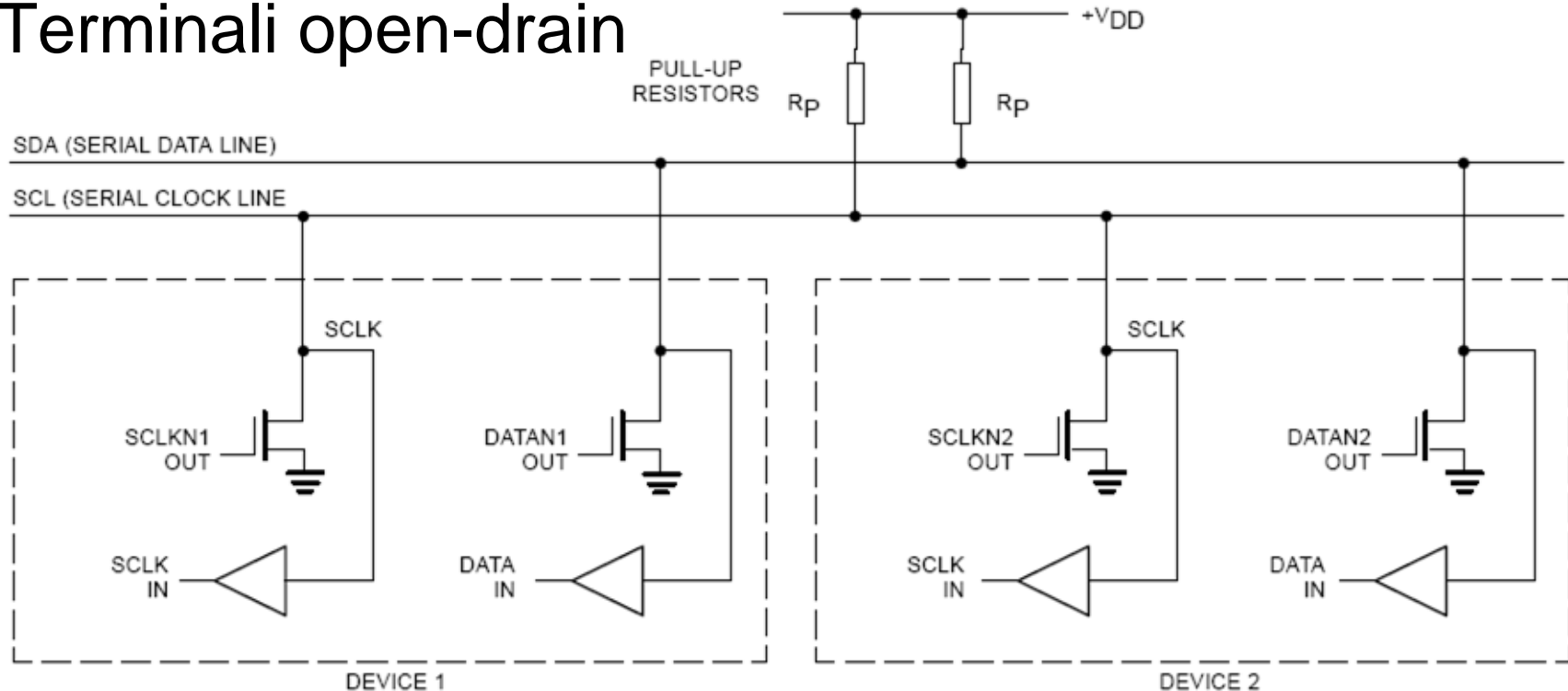


Protocollo I²C

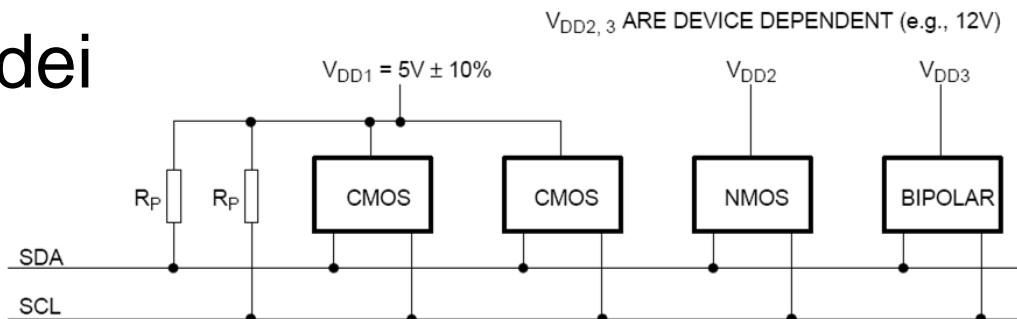
- Gestione del protocollo da parte del master
- Il master A deve trasmettere allo slave B
 - A indirizza B
 - A invia i dati a B
 - A termina il trasferimento
- Il master A deve ricevere dallo slave B
 - A indirizza B
 - A riceve i dati da B
 - A termina il trasferimento
- Sistema Multi-Master → necessità di arbitraggio

Protocollo I²C

- Struttura del bus e dei dispositivi
- Terminali open-drain

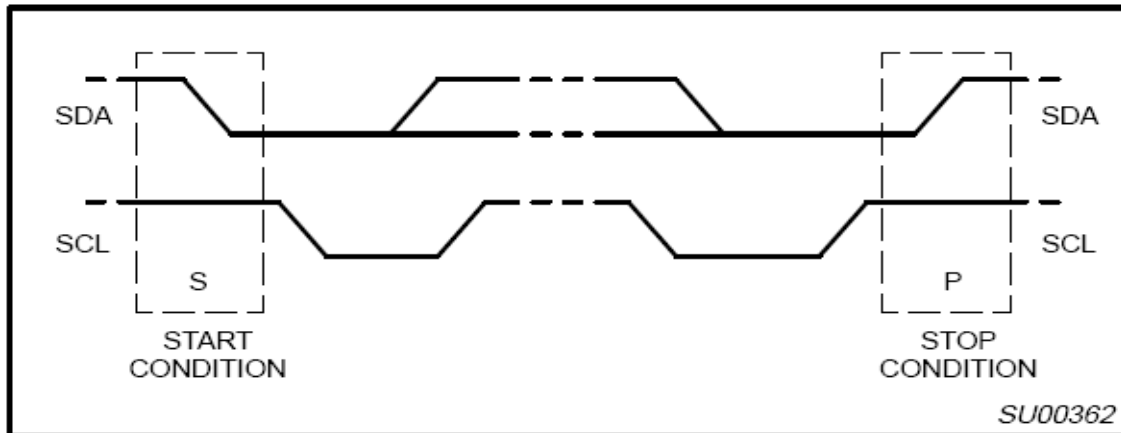


- Power supplies dei dispositivi indipendenti



Protocollo I²C

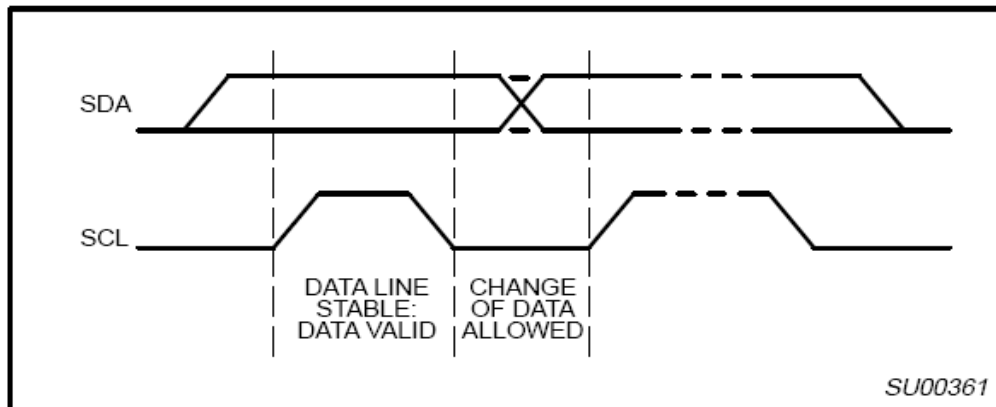
- Fasi del protocollo



START: transizione H→L di SDA mentre SCL è H

STOP: transizione L→H di SDA mentre SCL è H

SEMPRE GENERATE DAL MASTER!

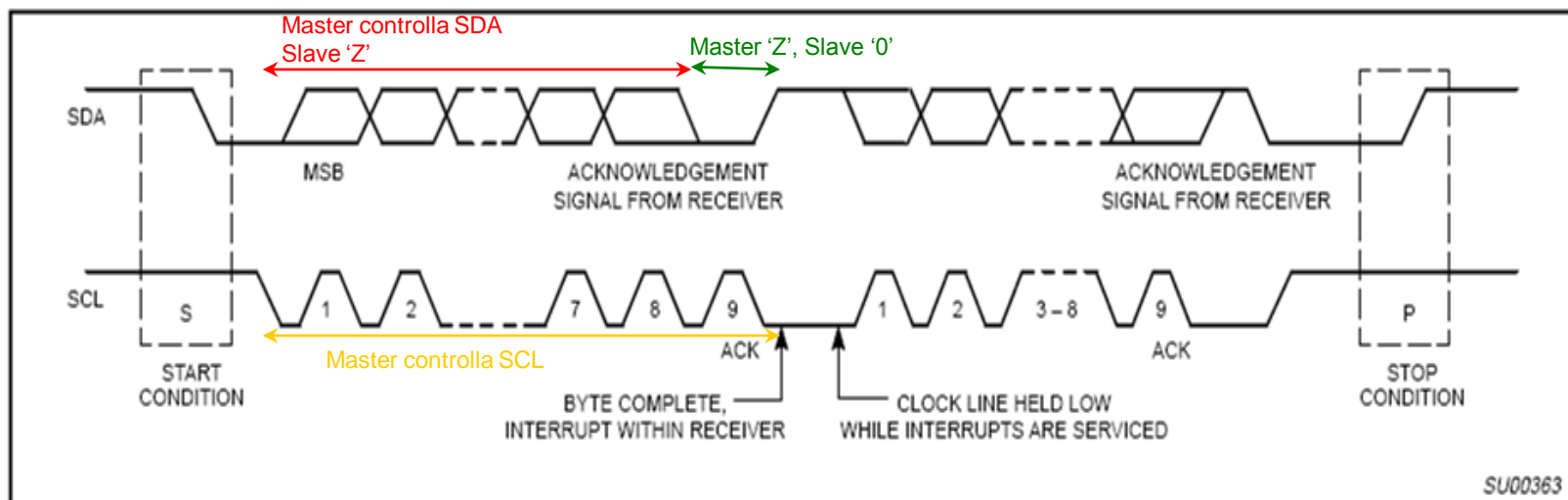


Trasferimento di un bit.

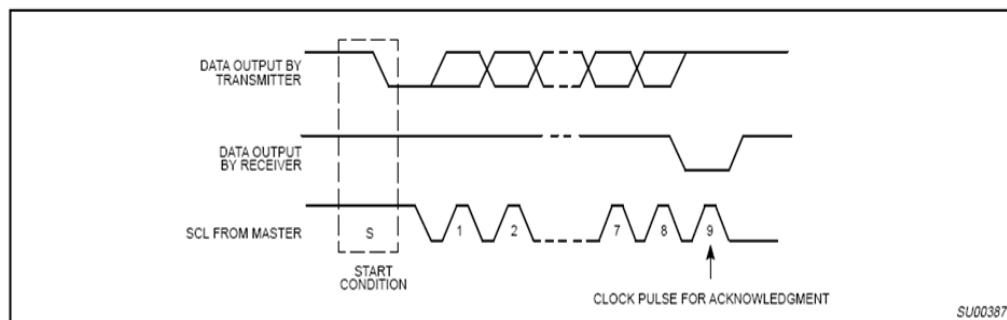
Il dato deve essere stabile mentre avviene l'impulso di SCL, sempre pilotato dal master

Protocollo I²C

- Trasferimento dati elementare
 - 8 bit + bit di ACKNOWLEDGE



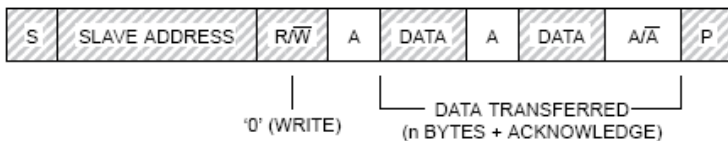
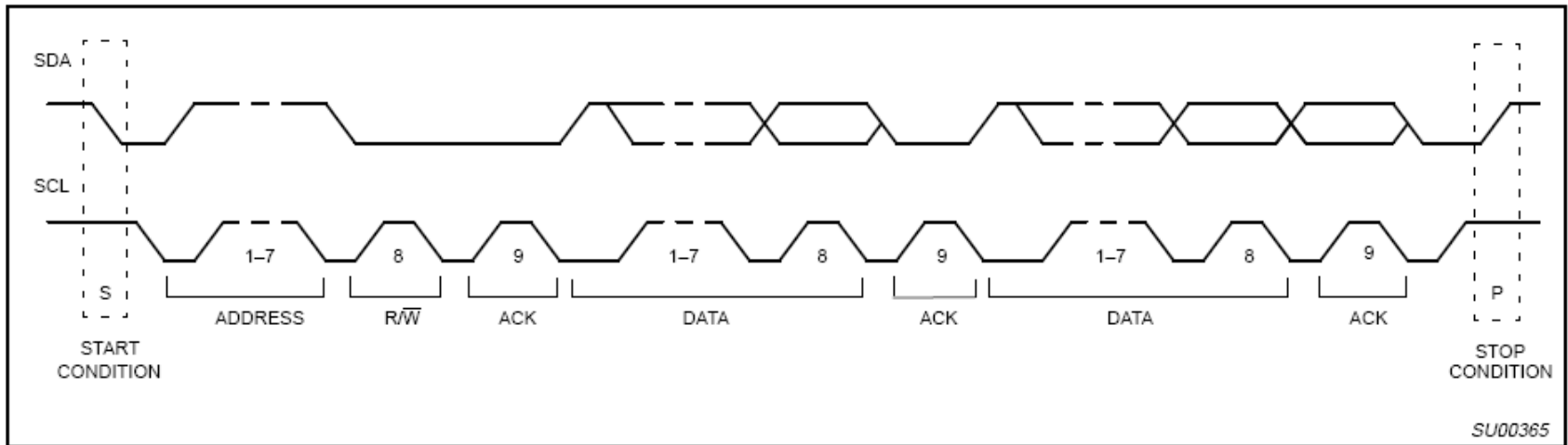
Andamento delle linee del bus



Trasferimento dal
punto di vista del
ricevitore

Protocollo I²C

- Trasferimento dati completo

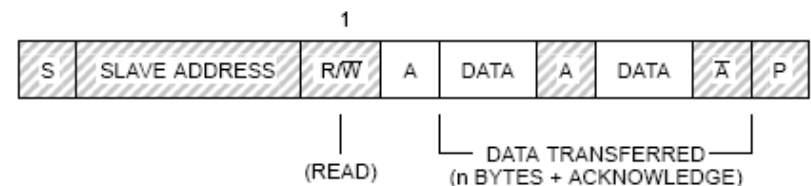


FROM MASTER TO SLAVE

FROM SLAVE TO MASTER

A = ACKNOWLEDGE (SDA LOW)
 Ā = NOT ACKNOWLEDGE (SDA HIGH)
 S = START CONDITION
 P = STOP CONDITION

Trasferimento M→S



Trasferimento S→M

Protocollo I²C

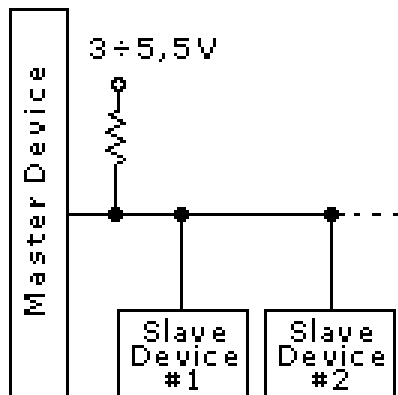
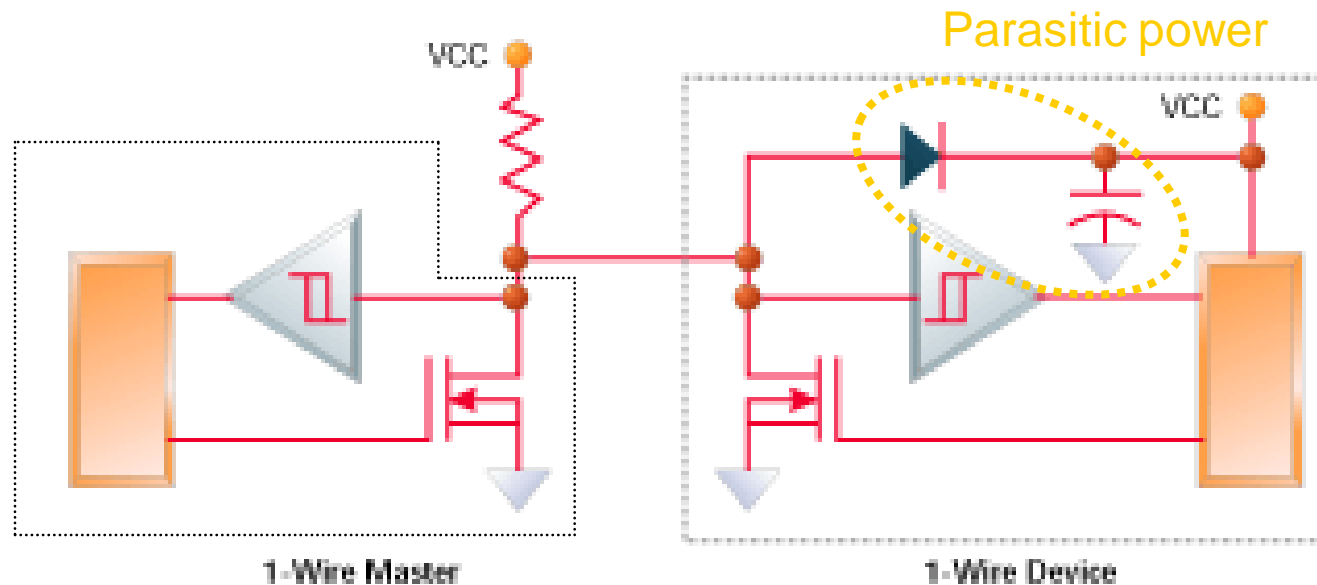
- Arbitraggio
 - Cosa succede se più master cominciano un ciclo contemporaneamente?
 - Ogni master M_i deve controllare istante per istante se il valore della linea è = a quello che vuole imporre. Diversamente c'è un altro Master M_{ATT} attivo
 - In tal caso M_i deve tornare subito in slave mode (M_{ATT} potrebbe volerlo indirizzare come slave!)

Protocollo 1-wire

- Protocollo Master/Slave (Dallas/Maxim)
- 1 Master
- Temporizzazioni implicite
- 1 linea + GND → quindi POWER e segnale sulla stessa linea!!
- Basso transfer rate
- Richiede logica interna per calcolo delle temporizzazioni
- Periferiche: EEPROM, memorie, I-buttons, sensori

Protocollo 1-wire

- Struttura dei dispositivi

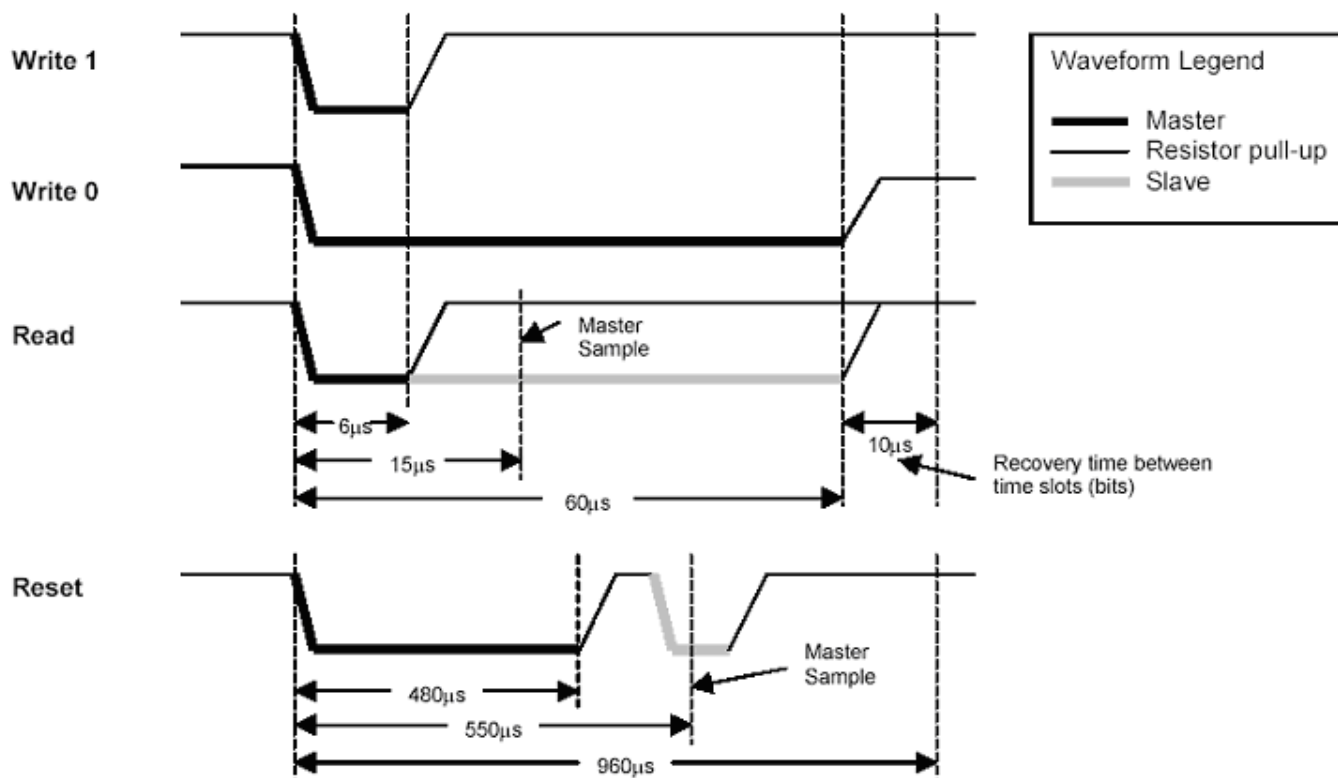


- Architettura multi-punto con master
- In ogni dispositivo è memorizzato un numero di serie personale di 64 bit (indirizzo univoco del dispositivo) e diviso in 3 parti.
 - 1 byte, contiene il codice identificativo del tipo di dispositivo
 - 6 byte contenenti un numero personale.
 - 1 byte di CRC (Cyclic Redundancy Check)

1-WIRE OPERATIONS Table 1

Operation	Description	Implementation
Write 1 bit	Send a '1' bit to the 1-Wire slaves (Write 1 time slot)	Drive bus low, delay 6 μ s Release bus, delay 64 μ s
Write 0 bit	send a '0' bit to the 1-Wire slaves (Write 0 time slot)	Drive bus low, delay 60 μ s Release bus, delay 10 μ s
Read bit	Read a bit from the 1-Wire slaves (Read time slot)	Drive bus low, delay 6 μ s Release bus, delay 9 μ s Sample bus to read bit from slave Delay 55 μ s
Reset	Reset the 1-Wire bus slave devices and ready them for a command	Drive bus low, delay 480 μ s Release bus, delay 70 μ s Sample bus, 0 = device(s) present, 1 = no device present Delay 410 μ s

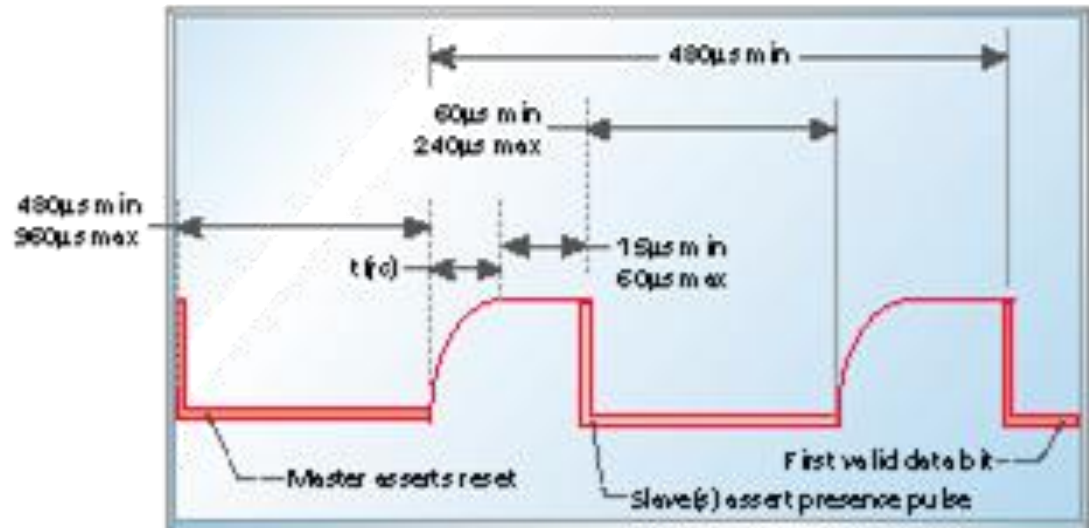
1-WIRE WAVEFORMS Figure 1



Protocollo 1-wire

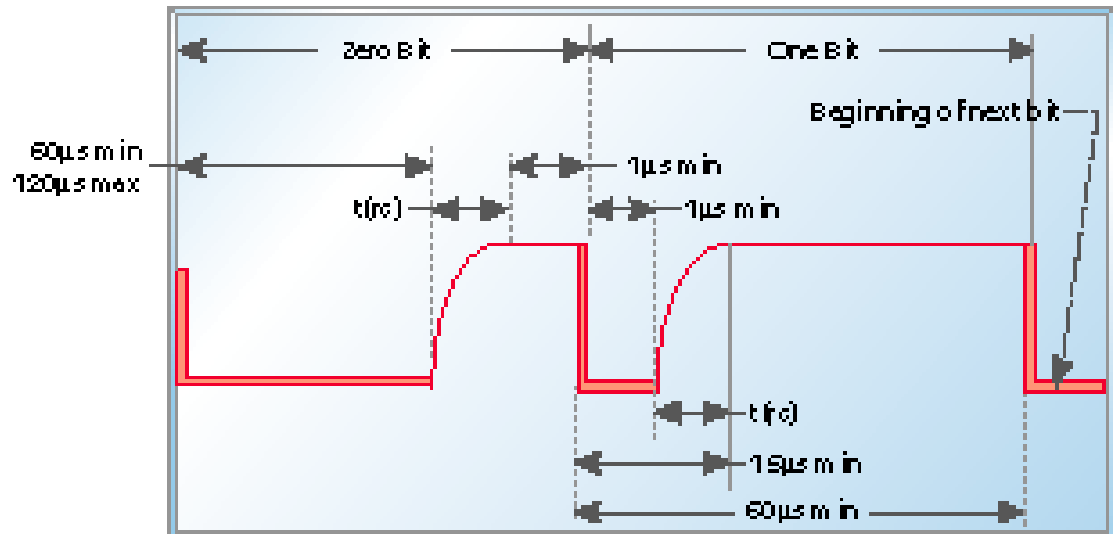
- Ogni trasferimento comincia con la fase di RESET

- Impulso basso del master per almeno 480 μ s
- Slave risponde con un impulso (se presente)



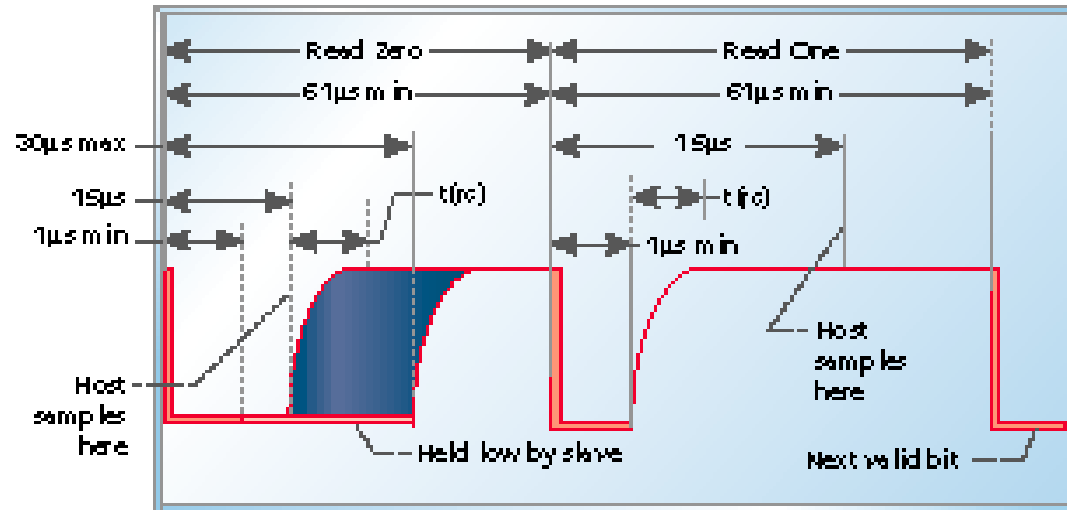
- Scrittura di un '1' o di uno '0' logico

In seguito il master può indirizzare un singolo slave presente sulla rete utilizzando i "ROM commands" previsti dal protocollo di comunicazione (comandi "ad alto livello" costruiti usando queste operazioni elementare)



Protocollo 1-wire

- Lettura di 1 bit



- Ogni dispositivo (EEPROM, memorie, sensori, etc. etc.) ha un ID univoco, necessario in presenza di più devices.

