
Tecniche di Comunicazione Digitale

Sommario

- Problematiche Generali
 - Comunicazioni Parallele
 - ▶ Varianti
 - ▶ Protocolli standard
 - Comunicazioni Seriali
 - ▶ Trasmissione Isocrona, Asincrona e Sincrona
 - ▶ Segnali, codifica e modulazione
 - ▶ Gestione degli errori
 - ▶ Protocolli standard
-

Problematiche Generali

Problematiche Generali

- Informazioni \Rightarrow Sequenze di Simboli \Rightarrow Segnali
 - ▶ Codifica
 - Costo, Affidabilità, Efficienza
 - Simboli binari
 - ▶ *bit*
 - Segnali elettrici (ottici)
 - Mezzi trasmissivi
 - ▶ Cavi, etere, (fibra ottica)
 - ▶ Apparecchiature di generazione, amplificazione, conversione, ecc.
-

Problematiche Generali

- Informazioni da trasmettere
 - ▶ bit \Rightarrow caratteri \Rightarrow messaggi
- Sincronizzazione e riconoscimento dei simboli
 - ▶ Accordo tra mittente e destinatario su quali finestre temporali sono destinate alla rappresentazione dei simboli
 - ▶ Si usano segnali ausiliari per indicare gli istanti significativi per la sincronizzazione
 - Comunicazioni parallele (breve distanza)
 - Segnali aggiunti (cavi differenti)
 - Comunicazioni seriali (per distanze elevate)
 - Un solo segnale per la sincronizzazione ed il trasporto di informazione

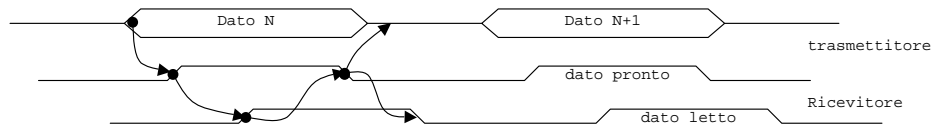
- 5 -

Comunicazioni Parallele

Comunicazioni Parallele

- Trasmissione di gruppi di bit (in genere, 8 o 16) in collegamenti a breve distanza (pochi metri)
- La sincronizzazione avviene mediante segnali ausiliari e tecniche di *handshaking*
 - Queste tecniche consentono di adeguare le velocità del trasmettitore e del ricevitore, in modo da non perdere informazioni e tenendo conto delle tolleranze sui diversi tempi di commutazione dei vari bit in parallelo

■ Full handshake

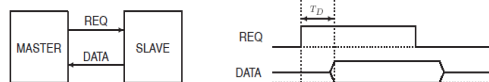


- 7 -

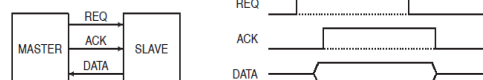
Comunicazioni Parallele

■ Varianti di handshaking

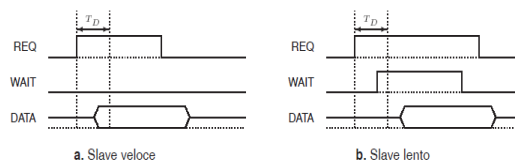
► Strobe



► Handshake



► Strobe/Handshake



- 8 -

Comunicazioni Parallele

- Protocolli paralleli standard

- ▶ Standard IEEE-488
- ▶ Centronics
- ▶ ISA/EISA
- ▶ SCSI
- ▶ PCI/ PCI X
- ▶ "Frontbus"

- 9 -

Comunicazioni Seriali

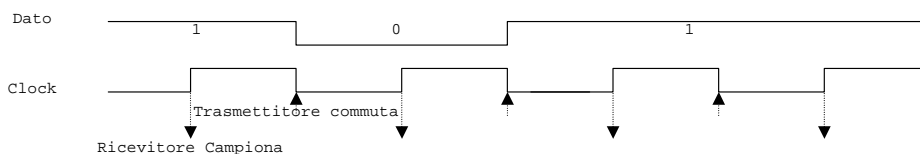
Comunicazioni Seriali

- Utilizzate per comunicazioni superiori a pochi metri tramite un singolo segnale utilizzato per trasmettere un bit alla volta
 - ▶ Tecniche di sincronizzazione
 - Consentire al ricevente il riconoscimento della finestra temporale di validità (e quindi di campionamento) del dato
 - Trasmissione Isocrona
 - » Unico segnale ausiliario di sincronizzazione (clock) comune a mittente e destinatario
 - Trasmissione Asincrona (velocità standard)
 - » Generatori di clock separati presso mittente e destinatario, di frequenze simili e sincronizzati ad ogni carattere
 - Trasmissione Sincrona (busi di campo, reti locali)
 - » Generatori di clock separati con capacità di aggancio di fase del ricevente

- 11 -

Comunicazioni Seriali

- Trasmissione Isocrona
 - ▶ La sincronizzazione di bit è basata su un unico segnale di clock, in comune tra mittente e destinatario
 - Fronti di discesa del clock
 - Il mittente imposta il livello del prossimo bit (fronti di commutazione)
 - Fronte di salita del clock
 - Il destinatario campiona sui fronti di discesa (fronti di campionamento)



- 12 -

Comunicazioni Seriali

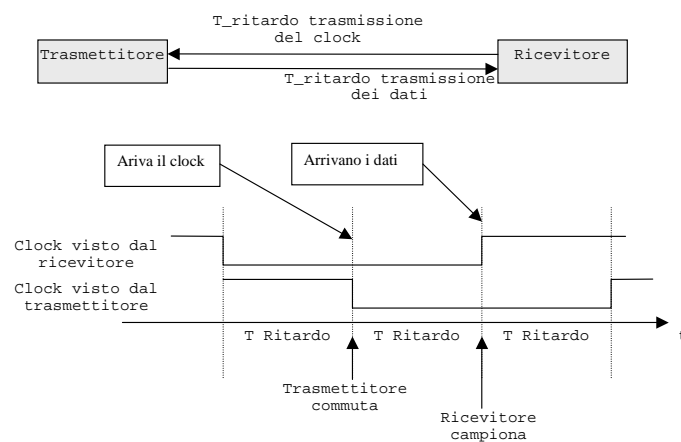
Trasmissione Isocrona

- Nelle trasmissioni isocrone, il clock è su di un conduttore a parte
 - Clock generato presso il trasmettitore
 - Clock e dati si propagano nella stessa direzione
 - » Si mantiene una corretta relazione di fase
 - Clock generato presso il ricevitore
 - Il clock ed i dati si propagano in direzioni diverse
 - » Il ritardo è pari a $2 \cdot \text{Tempo_propagazione}$
 - La distanza tra il fronte di discesa del clock ed il fronte di salita deve essere superiore a $2 \cdot \text{Tempo_propagazione}$
 - Limite alla velocità di trasmissione
- A causa della criticità dei tempi di propagazione e della linea di clock separata (possibili disturbi), la trasmissione isocrona è utilizzata per collegamenti relativamente brevi

- 13 -

Comunicazioni Seriali

Trasmissione Isocrona



- 14 -

Comunicazioni Seriali

■ Trasmissione Asincrona

- ▶ Ricevente e trasmittente hanno propri clock con frequenze simili e stabili tali da consente almeno la trasmissione di un singolo carattere
- ▶ Sincronizzazione ad ogni carattere
 - Ogni byte è preceduto da bit di start ed è seguito da almeno un bit di stop
 - Il ricevente si sincronizza sul fronte del bit di start e rimane sincronizzato per tutto il carattere
 - Il ritmo di trasmissione tra caratteri può variare
- ▶ La tecnica asincrona è molto utilizzata per brevi collegamenti punto-punto

- 15 -

Comunicazioni Seriali

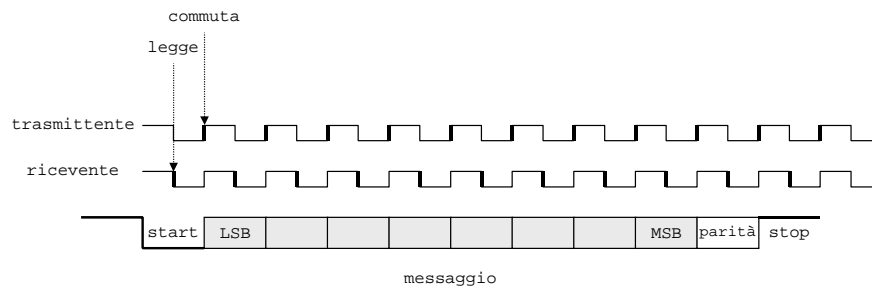
■ Trasmissione Asincrona

- ▶ Una volta sincronizzati i due clock essi si comportano come l'unico clock del caso isocrono
 - I fronti di commutazione sono forniti dal clock del trasmittente, mentre i fronti di campionamento sono forniti dal clock del ricevente e, per una corretta acquisizione, dovranno cadere all'interno della finestra temporale del bit
 - La similitudine delle frequenze dei clock deve essere tale da garantire che per tutti i bit del carattere lo sfasamento non superi l'intervallo di +/- metà della durata di un bit

- 16 -

Comunicazioni Seriali

• Trasmissione Asincrona con 8 bit più parità



- 17 -

Comunicazioni Seriali

• Trasmissione Sincrona

- ▶ Il Trasmittente ha un clock con frequenza stabile
- ▶ Il Ricevente ha un clock con aggancio di fase (Phase Lock Loop) sulle commutazioni dei dati ricevuti
 - La sincronizzazione di bit avviene dopo alcune commutazioni che consentono l'aggancio di fase
 - La sincronizzazione di Byte avviene sul riconoscimento di appositi caratteri di sincronismo (3 .. 5) anteposti ad ogni **messaggio**
 - I bit del messaggio sono trasmessi senza pause evitando lunghe sequenze senza commutazioni
 - » Per evitare che lunghe sequenze senza commutazioni facciano perdere l'aggancio di fase si aggiungono bit per interrompere sequenze senza commutazioni
- ▶ Tipica nelle trasmissioni ad alta velocità

- 18 -

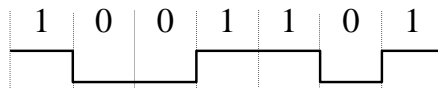
Comunicazioni Seriali

• Segnali, codifica e modulazione

- ▶ Segnali a due valori: utilizzati con cavi per trasmissioni su brevi distanze e nelle LAN

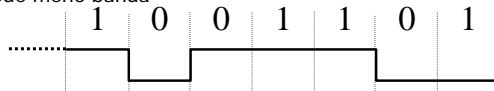
- NRZ (*Not Return to Zero*): ogni transizione indica un cambiamento di valore logico

- La più usata



- NRZI: utilizza la presenza o l'assenza di transizione per indicare un valore logico

- Meno usata ma richiede meno banda



- 19 -

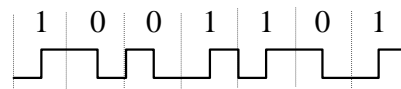
Comunicazioni Seriali

• Segnali, codifica e modulazione

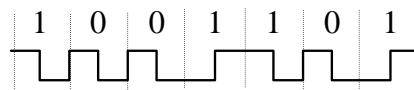
- ▶ Segnali a due valori

- Manchester bifase: uno specifico fronte indica un valore logico ben definito (salita=1; discesa=0)

- Usate nelle trasmissioni sincrone poiché favoriscono l'aggancio di fase a discapito della banda



- Manchester differenziale: un 1 (0) logico è rappresentato da un fronte all'inizio dell'intervallo mentre uno 0 (1) è indicato da un fronte nel mezzo del periodo.



- 20 -

Comunicazioni Seriali

• Gestione degli Errori

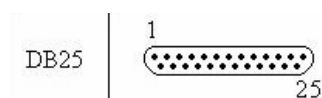
- ▶ A livello di carattere
 - Riconoscimento: bit di parità
 - Correzione: codice di hamming
- ▶ A livello di messaggio
 - Riconoscimento: caratteri aggiuntivi funzione dei caratteri trasmessi
 - Cyclic Redundancy Check
 - » Calcolato su 16 bit
 - Checksum
 - » Somma modulo 256
 - Parità longitudinale
 - Correzione: normalmente si usa la ritrasmissione

- 21 -

Comunicazioni Seriali

• Protocolli standard

- ▶ RS-232-C (1969)
 - E' stato proposto per collegare dispositivi DTE con dispositivi DCE
 - DTE: *Data Terminal Equipment* (calcolatori, stampanti, ecc.)
 - DCE = *Data Communications Equipment* (modem)
 - Segnali elettrici: 0 = ON $\Rightarrow V > +3 V$; 1 = OFF $\Rightarrow V < -3 V$
 - Distanza: < 15 m
 - Velocità: < 20.000 bit/sec
 - Sequenza: dal LSB al MSB con eventuale aggiunta della parità
 - Connettore Dte: Cannon maschio 25 contatti



- 22 -

Comunicazioni Seriali

● Protocolli standard

► RS-232-C: disposizione dei segnali principali

1	GND	massa di protezione
2	TXD	dati trasmessi
3	RXD	dati ricevuti
4	RTS	richiesta di invio
5	CTS	consenso all'invio
6	DSR	data set pronto
...		
15	TXC	clock di trasmissione
17	TXD	clock di ricezione
20	DTR	data terminal pronto

- Lo standard viene spesso adottato in forma ridotta anche per connessione diretta tra dispositivi di tipo DTE, cioè senza interposizione di modem, ma con l'uso di un cavo (detto "null modem") in cui sono incrociati i collegamenti 2-3, 15-17, ecc.

- 23 -

Comunicazioni Seriali

● Protocolli standard

► U(S)ART

- Le trasmissioni seriali, sincrone ed asincrone, sono così diffuse da suggerire la produzione dei cosiddetti U(S)ART (*Universal Synchronous Asynchronous Receiver Transmitter*) in genere studiati per un facile interfacciamento con i più diffusi microprocessori
 - *Custom Single Purpose Processor*
 - » Molto spesso tali circuiti sono integrati nei *microcontroller/SOC*
- Si tratta di circuiti integrati che svolgono le operazioni necessarie per i livelli inferiori della gerarchia ISO-OSI nelle trasmissioni seriali
 - UART NS8250 e NS16550
- Si noti che si parla di UART nel caso in cui i circuiti siano previsti per la sola, più semplice e più usata, comunicazione asincrona

- 24 -

Comunicazioni Seriali

• Protocolli standard

▶ U(S)ART

• Funzionalità tipiche

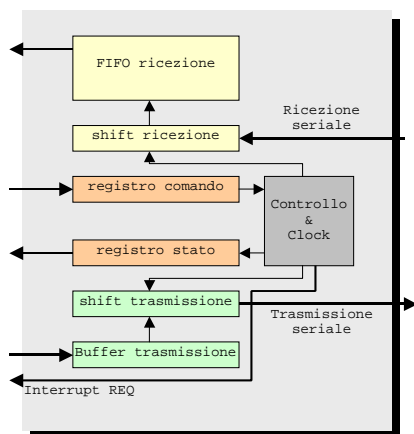
- Conversione seriale/parallelo
- Generazione/verifica della parità
- Sincronizzazione di bit o di carattere
- Bufferizzazione
- Generazione di richieste di interrupt

• Registri tipici

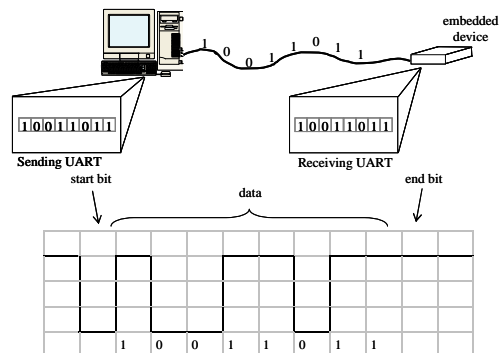
- Buffer e shift di trasmissione
- Buffer e shift di ricezione
- Registri di modo
- Registri di comando
- Registri di stato

- 25 -

Serial Transmission Using UARTs



- 26 -



Comunicazioni Seriali

● Protocolli standard

► I²C (*Inter IC*)

- Sviluppato da Philips all'inizio degli anni '80
- Pensato per l'interconnessione di più dispositivi a distanza limitata
 - Velocità di trasmissione variabile tra 100-Kbit/s con un indirizzamento a 7 bit (secondo la specifica originale) e 3.4-Mbit/s con indirizzamento a 10 bit (versioni più recenti dello standard)
- Si avvale di due linee: SDA (Serial Data Line) ed SCL (Serial Clock Line) alle quali sono connessi tutti i dispositivi interessati dalla comunicazione
 - Tali dispositivi possono essere master o slave, essendo i dispositivi master gli unici in grado di iniziare una comunicazione
 - » In generale lo standard prevede la possibilità di avere più di un dispositivo master e fornisce un meccanismo deterministico e relativamente semplice per la negoziazione del bus

- 27 -

Comunicazioni Seriali

● Protocolli standard

► I²C (*Inter IC*)

- Essendo un bus asincrono, la sincronizzazione tra master e slave avviene a ogni simbolo
- La comunicazione di un simbolo inizia con una start condition e termina con una stop condition
 - Il pacchetto trasmesso è costituito dai seguenti campi
 - » La start condition
 - » Un indirizzo di 7 o 10 bit
 - » L'operazione richiesta
 - » Un bit di acknowledge, asserito dallo slave per indicare l'avvenuta ricezione dell'indirizzo
 - » Il byte di dati
 - » Un nuovo bit di acknowledge, asserito dallo slave per indicare l'avvenuta ricezione dei dati
 - » La stop condition

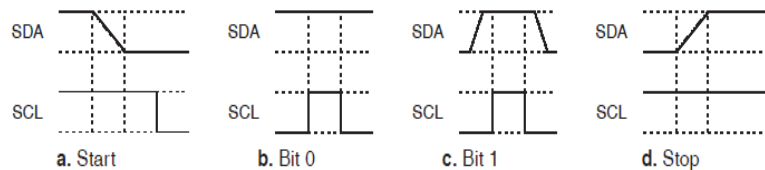
- 28 -

Comunicazioni Seriali

● Protocolli standard

► I²C (*Inter IC*)

● Condizioni di trasmissione e di inizio/fine pacchetto



● Struttura logica di un pacchetto



- 29 -

Comunicazioni Seriali

● Protocolli standard

► I²C (*Inter IC*)

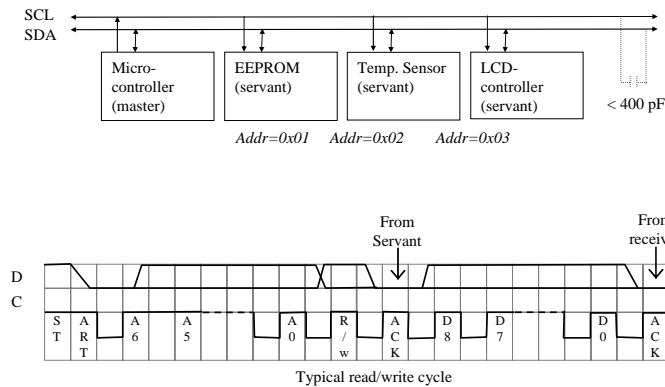
- Sia per la trasmissione dell'indirizzo sia per la trasmissione dei dati i bit sono ordinati dal più significativo al meno significativo
 - Il master ha sempre il controllo delle linee di clock e di dati, tranne che durante la trasmissione dei bit di acknowledge, per la quale è lo slave a forzare i valori su tali linee
- E' possibile connettere allo stesso bus più di un dispositivo master
 - Politica di arbitraggio che consenta a un dispositivo master di determinare se è effettivamente in grado prendere il controllo del bus
 - » Connessione *open-drain* che realizza la funzione di *wired-and*
 - Un dispositivo master che volesse prendere il controllo del bus deve verificare le due seguenti condizioni: controllare che la linea non ha valore 0 ed essere in grado di forzare un 1 sulla linea di clock

- 30 -

Comunicazioni Seriali

• Protocolli standard

► I²C (*Inter IC*)



- 31 -

Comunicazioni Seriali

• Protocolli standard

► SPI (*Serial Peripheral Interface*)

- E' basato su uno schema di trasmissione sincrona bidirezionale in cui un solo dispositivo funge da master
- Il protocollo utilizza un bus con almeno 4 linee per ogni slave
 - SCLK (*Serial Clock*)
 - MOSI (*Master Output Slave Input*)
 - MISO (*Master Input Slave Output*)
 - SS (*Slave Select*) per ogni dispositivo slave
- I segnali MISO sono connessi in *wired-or* su un'unica linea e pertanto è necessario che gli slave non attivi, ovvero quelli per cui SS è alto (il segnale slave select è asserito basso), mantengano la propria linea MISO in alta impedenza

- 32 -

Comunicazioni Seriali

■ Protocolli standard

► SPI (*Serial Peripheral Interface*)

- Per iniziare la comunicazione, il master asserisce il segnale SS per lo slave desiderato ponendolo a 0, quindi genera un segnale di clock a una frequenza uguale o minore a quella massima supportata dallo slave selezionato e lo trasmette sulla linea SCLK
 - Durante tutto il tempo in cui il clock è presente, master e slave comunicano in full-duplex sulle linee MOSI e MISO
- Uno svantaggio di questo schema consiste nel richiedere una linea di slave select per ogni slave
 - Per superare questa limitazione è possibile ricorrere a uno schema *daisy-chain*

- 33 -

Comunicazioni Seriali

■ Protocolli standard

► SPI (*Serial Peripheral Interface*)

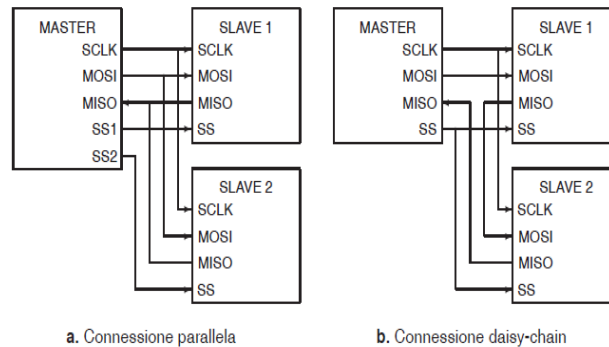
- I vantaggi più evidenti degli schemi di comunicazione basati su SPI rispetto a un bus seriale quale, per esempio, I2C consistono in un maggiore throughput dovuto sia alla comunicazione full-duplex sia all'assenza d'indirizzamento
- Per contro, non prevedendo un meccanismo di acknowledge a livello fisico, il protocollo SPI non permette di segnalare al master l'avvenuta comunicazione oppure una condizione di errore
 - Inoltre lo schema prevede un unico dispositivo master il che, spesso, è troppo limitante

- 34 -

Comunicazioni Seriali

- Protocolli standard

- ▶ SPI (*Serial Peripheral Interface*)



- 35 -

Comunicazioni Seriali

- Protocolli standard

- ▶ CAN (*Controller Area Network*)

- Il bus CAN si basa su un protocollo seriale differenziale, sviluppato originariamente negli anni '80 dalla Bosch per la connessione di dispositivi di controllo in ambienti fortemente affetti da rumore di natura elettromagnetica
 - Tale bus è particolarmente adatto per la trasmissione di messaggi di piccole dimensioni (8 byte) e utilizza un codice CRC-15 per proteggere i dati da errori di trasmissione
 - » Supporta velocità fino a 1-Mbit/s per distanze al di sotto dei 40-m, ma a frequenze più basse può coprire distanze maggiori, per esempio 500-m a 125-Kbit/s
 - Appartiene alla categoria dei cosiddetti *field-bus*

- 36 -

Comunicazioni Seriali

• Protocolli standard

► CAN (*Controller Area Network*)

- Il bus CAN presenta un meccanismo di trasmissione privo di arbitraggio esplicito grazie alla definizione del concetto di priorità
 - Un messaggio con alta priorità vince l'arbitraggio e viene trasmesso, mentre un messaggio a bassa priorità rileva che un messaggio a priorità più alta è già in trasmissione e si mette in attesa
 - » Questo meccanismo è realizzato mediante un modello basato su una logica binaria e sui concetti di bit dominante, uno 0, e bit recessivo, un 1

	dominante	recessivo
dominante	dominante	dominante
recessivo	dominante	recessivo

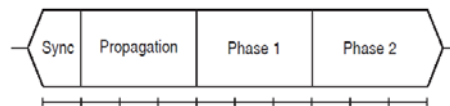
- 37 -

Comunicazioni Seriali

• Protocolli standard

► CAN (*Controller Area Network*)

- Bus asincrono che prevede l'utilizzo di una struttura per la trasmissione di ogni bit
 - Garantire allineamento del clock e degli istanti di campionamento



- Ogni bit è suddiviso in quattro segmenti
 - Il primo è il segmento di sincronizzazione di durata fissa
 - Il secondo, di propagazione, ha lo scopo di compensare i ritardi fisici
 - Infine, gli ultimi due segmenti hanno durata variabile e possono essere accorciati o allungati per mantenere i clock allineati
 - » Il dato viene sempre campionato al confine tra la fase 1 e la fase 2

- 38 -

Comunicazioni Seriali

- Protocolli standard

- ▶ CAN (*Controller Area Network*)

- Al livello superiore, il protocollo implementato mediante il bus CAN si basa sul concetto di frame
 - Un frame costituisce l'unità elementare di trasmissione e può essere di quattro tipi diversi
 - » Data Frame, Remote Frame, Error Frame e Overload Frame
 - Il tipo più comune è il data frame, in quanto utilizzato per lo scambio di dati, che costituiscono la maggior parte del traffico

SoF	Arbitration	Control	Data	CRC	Ack	EoF
-----	-------------	---------	------	-----	-----	-----