

Il codice ASCII rappresenta lo standard più utilizzato nella codifica di sorgente, anche se non è il più efficiente. Esso è stato assunto nella direttiva V.3 anche dall'europeo CCITT (*Consultative Committee International Telegraph and Telephon*) come riferimento per l'alfabeto n° 5. Si compone di caratteri alfanumerici e di controllo per il destinatario secondo la tabella 6 (il bit 0 manca perché usato come bit di parità: vedere il paragrafo 5).

I caratteri di controllo hanno il significato riportato in tabella 7.

Esiste anche una versione di codice ASCII estesa, per l'uso nei calcolatori ma che non rientra negli standard normativi, anche se molto diffusa.

TAB. 7 - Caratteri di controllo del codice ASCII

	Sigle	Definizione	Funzione
Comandi di trasmissione	ACK	Acknowledge	Conferma ricezione corretta
	CAN	Cancel	Cancellazione
	DLE	Data Link Escape	Cambio di funzione dei comandi
	EM	End of Medium	Fine del mezzo
	ETB	End of Transmission Block	Fine del blocco in trasmissione
	ENQ	Enquiry	Interrogazione
	EOT	End Of Transmission	Fine della trasmissione
	ETX	End Of Text	Fine del testo
	NAK	Negative Acknowledge	Avviso di ricezione non corretta
	SOH	Start Of Header	Inizio dell'intestazione
	STX	Start Of Text	Inizio del testo
	SYN	Synchronous Idle	Caratteri di sincronizzazione
Comandi di formato	BS	Back Space	Spostamento indietro di 1 passo
	CR	Carriage Return	Ritorno carrello
	FF	Form Feed	Salto di pagina
	LF	Line Feed	Avanzamento di una linea
	HT	Horizontal Tab	Tabulazione orizzontale
	SUB	Substitution	Sostituzione
	VT	Vertical Tabulation	Tabulazione verticale
Comandi di dispositivi	DC i	Device Control	Controllo del dispositivo i (i = 1 ÷ 4)
Separatori di informazione	FS	File Separator	Separatore di file
	GS	Group Separator	Separatore di gruppo
	RS	Record Separator	Separatore di record
	US	Unit Separator	Separatore di unità
Comandi ausiliari	BEL	Bell	Campanello
	ESC	Escape	Cambio di codice
	NUL	Null	Null
	SO	Shift Out	Elimina shift (caratteri normali)
	SI	Shift In	Inserisci shift (caratteri alternativi)

4 Codici di canale

La possibilità di trasmettere informazione su un canale è legata anche a un aspetto economico che richiede il massimo sfruttamento delle risorse del canale. Come esempio si pensi alla possibilità di collegamento di dispositivi che esiste in un PC: la porta parallela (tipicamente per la stampante) e la porta seriale (tipicamente per mouse). È evidente la semplicità fisica del cavo che collega il mouse nei confronti di un cavo per stampante. Il minore uso di conduttori di collegamento consente una semplificazione del canale in cui deve transitare l'informazione. Quando l'informazione deve percorrere lunghe distanze, per raggiungere la destinazione, la semplicità del canale diventa un aspetto irrinunciabile e governa le scelte tecniche da adottare. In altre parole se è possibile pensare a un canale parallelo per brevi distanze, nessuno potrebbe pensare di trasferire a distanza l'informazione a una stampante tramite tale canale.

I codici di canale nascono da due necessità fondamentali:

- trasformare l'informazione in forma essenzialmente seriale per un uso corretto delle possibilità del canale;
- trasferire l'informazione tra sorgente e destinatario con la massima sicurezza possibile anche in relazione al fatto che il canale possa alterare il contenuto informativo (rumore, distorsioni ecc.).

In questo paragrafo si analizzeranno i problemi legati alla trasformazione seriale dell'informazione mentre si rimanda al paragrafo successivo per il problema della sicurezza di trasmissione.

→ La trasformazione seriale dell'informazione comporta la risoluzione di *alcuni problemi legati alla sincronizzazione*, necessaria per la trasmissione dell'informazione stessa, e allo *spettro del segnale trasmesso*. Per la *sincronizzazione* del segnale informatico è *essenziale* la presenza di un opportuno segnale, il **segnale di clock**, ridondante per quanto attiene la quantità di *informazione trasmessa*, ma *fondamentale* per il suo *riconoscimento*: tale segnale indica infatti l'istante in cui deve essere assunta l'informazione trasmessa. In base a tale segnale la trasmissione può essere *classificata* come:

- **sincrona**, se il segnale di clock è presente in modo autonomo assieme al segnale informatico;
- **asincrona**, se il segnale di clock non risulta essere autonomo, ma deve essere ricavato dal segnale informatico stesso.

■ ESEMPIO 1

Un esempio di trasmissione sincrona è realizzato all'interno del calcolatore dove l'informazione viene scambiata tra le varie parti presenti in sincrono con il segnale di clock appositamente costruito. Un esempio di trasmissione asincrona è invece fornito dalla porta seriale del calcolatore dove, secondo adeguate regole di ricostruzione, viene dedotto il segnale di sincronizzazione.

- **Codice NRZ (Not Return Zero)**. Si tratta del *codice più spontaneo* nell'ambito della trasmissione dell'informazione, anche se *il meno adatto*, per le sue caratteristiche, a essere inviato a grandi distanze. Per fissare le idee si immagini di voler trasferire periodicamente un particolare byte tra una sorgente e un destinatario:

$$B = 01101010$$

8

La codifica di canale, mediante codice NRZ, viene eseguita secondo le modalità di figura 2a dove è indicato anche il segnale di clock per la sincronizzazione. *Il momento ideale per il riconoscimento dei valori binari è individuato dalla freccia (fronte di discesa del segnale di clock)*. Per il codice NRZ si ha che $t_b = T_{ck}$ ossia ogni bit mantiene il livello che gli compete per tutto il periodo di clock.

Lo *spettro del segnale* in un codice NRZ può essere individuato se si pensa al caso in cui è presente una *sequenza alternata di 1 e 0*. In tale caso il segnale risultante è un'onda quadra di *frequenza pari a*:

$$* \quad f_Q = \frac{1}{2t_b} = \frac{1}{2T_{ck}}$$

9

Lo spettro può essere subito individuato se si ricorda quanto esposto nel capitolo 3. In particolare esisteranno, oltre alla componente continua di ampiezza pari ad $A_p/2$, solo armoniche dispari le cui ampiezze sono inversamente proporzionali all'ordine dell'armonica. Saranno allora nulle le armoniche di frequenza $2f_Q$, $4f_Q$, $6f_Q$ ecc.:

$$f_z = 2nf_Q = \frac{n}{t_b} = nF_c = nf_{ck} \quad \text{con} \quad n = 1, 2, 3, \dots$$

10

Si osservi che $1/t_b$ viene spesso indicata col nome di **frequenza di cifra** (binaria) e viene indicata con F_c e *coincide con la frequenza di clock f_{ck}* .

Si consideri ora il caso in cui, nel byte inviato periodicamente, esiste un solo bit uguale a 1 mentre tutti gli altri bit sono nulli, come in figura 2b. Lo spettro può essere indivi-

Armoniche di
ampiezza nulla

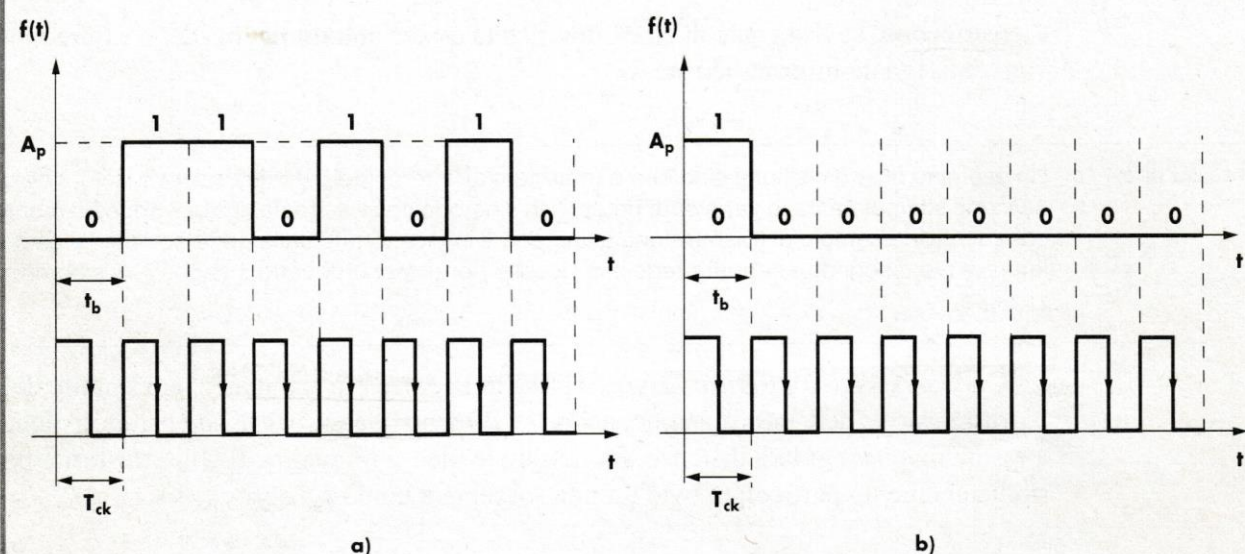
duato, al solito, ricordando quanto esposto nel capitolo 3. In particolare le frequenze di azzeramento dello spettro saranno:

$$f_z = \frac{n}{D}f = \frac{n}{t_b} \quad \text{con } n = 1, 2, 3, \dots$$

11

dove $f = 1/8t_b = f_Q/4$ e $D = 1/8$. Come si vede le frequenze di azzeramento sono ancora le stesse.

FIG. 2 Serializzazione del byte 01101010 (a) e del byte 10000000 secondo il codice NRZ (b).

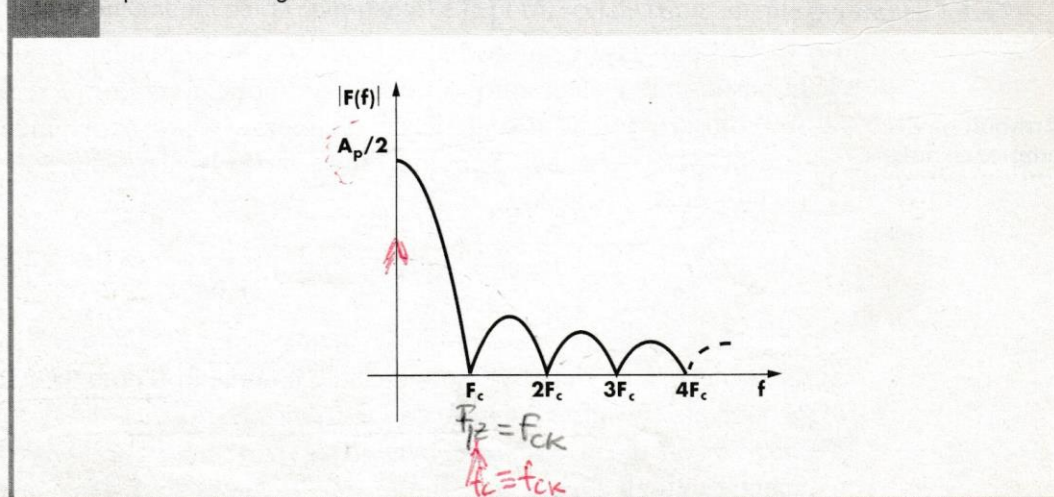


Frequenza e duty-cycle sono inversamente proporzionali

Se ora viene cambiata la disposizione dei bit nel byte viene cambiata la frequenza del segnale e in proporzione inversa il duty cycle, quindi *le frequenze di azzeramento dello spettro delle ampiezze non mutano*. La cosa è ancora più evidente se si considerano lunghe sequenze di 1 e 0 da inviare nel canale con cifre binarie equiprobabili.

Quindi lo spettro delle ampiezze di un segnale binario qualunque avrà la forma di figura 3. Si osservi che lo spettro delle ampiezze contiene una componente continua di ampiezza pari ad $A_p/2$, si estende all'infinito e non contiene la frequenza fondamentale del clock (pari alla frequenza di cifra) e neppure le sue armoniche poiché sono proprio le posizioni in cui lo spettro delle ampiezze si annulla.

FIG. 3 Spettro di un segnale seriale binario.



freq. del 1° zero = f_ck

il segn. binario modula il clock. forse il bus della porta di uscita

P1 *

C.C.

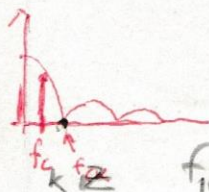
P2 *

Bandwidth

Assenza della frequenza di clock *

P3 f_{ck}

come 00K



Nel codice RZ esiste la frequenza di clock *

risultato P3 *

Non esiste P1 *

ma P4 *

2 PICCOLI!
CON UNA FAVA.

1) garantire l'estrazione del clock anche in presenza di lunghe sequenze di zeri

2) eliminare la componente continua

La presenza della componente continua è un grave problema nell'utilizzazione di questo codice al quale si cerca di ovviare introducendo non un segnale unidirezionale (variabile tra 0 e A_p) ma bidirezionale (variabile tra $-A_p$ e $+A_p$). La presenza della componente continua (che deve essere lasciata transitare per la ricostruzione corretta del segnale) altera infatti le condizioni di polarizzazione di eventuali dispositivi presenti nel ricevitore.

Meno grave è l'estensione dello spettro che può essere limitato, senza grossi problemi di riconoscimento delle cifre binarie, a una frequenza pari alla frequenza di primo azzeramento dello spettro ($n = 1$) ossia alla frequenza di cifra. La verifica può essere ottenuta col programma FOUR per esempio per il segnale di figura 2b (utilizzando la possibilità di visualizzare il segnale nel dominio del tempo con inserito un filtro passa-basso).

L'assenza dell'armonica fondamentale della frequenza di clock è forse il problema più grave e non consente la sincronizzazione della decodifica a distanza tramite estrazione dell'armonica fondamentale, salvo mediante l'uso di artifici approssimati utilizzati per sequenze brevi e limitate di bit.

In ogni caso la codifica NRZ appesantisce la ridondanza del codice di sorgente e rende meno efficiente l'uso del canale di trasmissione. Si pensi poi come possa essere problematica la sincronizzazione quando sono presenti lunghe sequenze di bit uguali a 0 oppure 1 da inviare nel canale.

• **Codice RZ (Return to Zero).** Si tratta di un codice simile al precedente ma che cerca di ovviare in parte ai problemi esposti. In questo codice la codifica del simbolo 0 rimane uguale a quella adottata per il codice NRZ, mentre per il simbolo 1 si utilizza in pratica un ciclo di clock. Quindi l'esempio di figura 2a diventa quello di figura 4.

Se lo stato alto permane per il 50% del tempo di bit t_b , si ha un codice RZ 50%, per durate inferiori si modifica di conseguenza la sigla. Se ora si pensa al caso della sequenza alternata di 1 e 0 si nota che l'onda quadra ottenuta ha una frequenza esattamente doppia rispetto al caso del codice NRZ. Quindi lo spettro delle ampiezze ha un andamento simile a quello di figura 3 ma con scala delle frequenze espansa del doppio. Così si hanno frequenze di zero a:

$$f_z = 2nf_{ck} \quad \text{con } n = 1, 2, 3, \dots$$

12

Alla frequenza di clock lo spettro delle ampiezze non si annulla più (si annulla al valore doppio di tale frequenza) e quindi, mediante un filtro selettivo è possibile estrarre la frequenza del clock necessaria alla sincronizzazione dell'informazione trasmessa.

Esiste ancora una componente continua presente nel segnale però il modulo ricavato secondo Fourier ha un valore pari alla metà di quello del codice NRZ, ossia pari a $A_p/4$. Si noti solo che l'informazione per ogni cifra deve essere assunta non in centro al tempo di bit ma a 1/4 dal suo inizio.

— COME ASK.

• **Codice bifase o Manchester.** Il codice RZ non consente l'estrazione del clock nel caso in cui nell'informazione trasmessa siano presenti lunghe sequenze di zeri. Si pensi infatti al caso in cui l'informazione sia costituita da una sequenza indefinita di zeri: le componenti spettrali sono tutte nulle e quindi non è certo possibile estrarre la frequenza di clock. Se al codice RZ si associa anche al simbolo 0 una transizione di segnale, si ottiene il codice bifase che è di solito realizzato tra una ampiezza positiva e una negativa allo scopo di eliminare anche la componente continua. Nel caso dell'esempio relativo alla figura 2a si ottiene la situazione di figura 5. La codifica è ottenuta mediante diversa disposizione del fronte di segnale al centro del tempo di bit t_b : un fronte di salita codifica il simbolo 0, mentre un fronte di discesa codifica

il simbolo 1 (ma è possibile anche il contrario). Il risultato è un segnale impulsivo di frequenza pari al clock, con *fase e duty-cycle variabili*: la fase, rispetto al clock, è di 0° oppure di 180° , mentre il duty-cycle varia tra il 25% e il 50%.

FIG. 4 Il codice RZ per il byte 01101010.

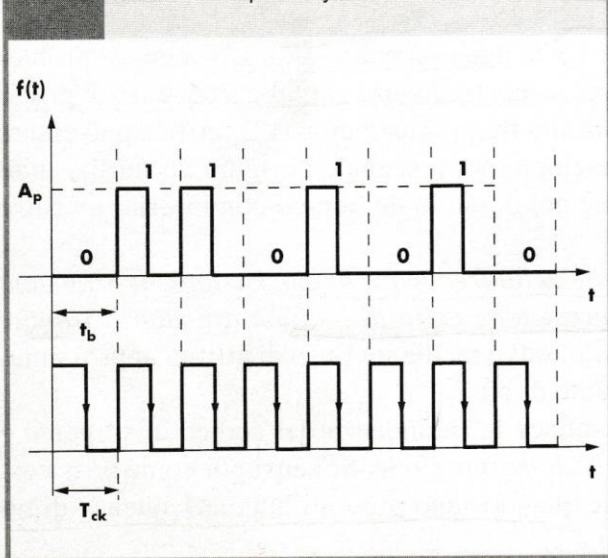
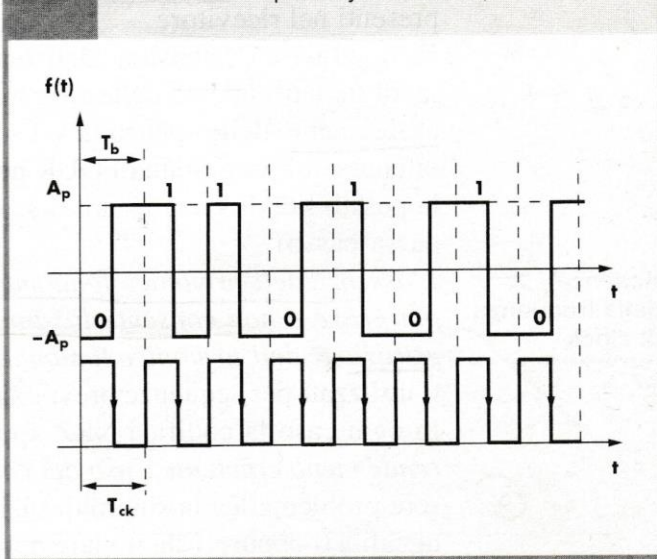


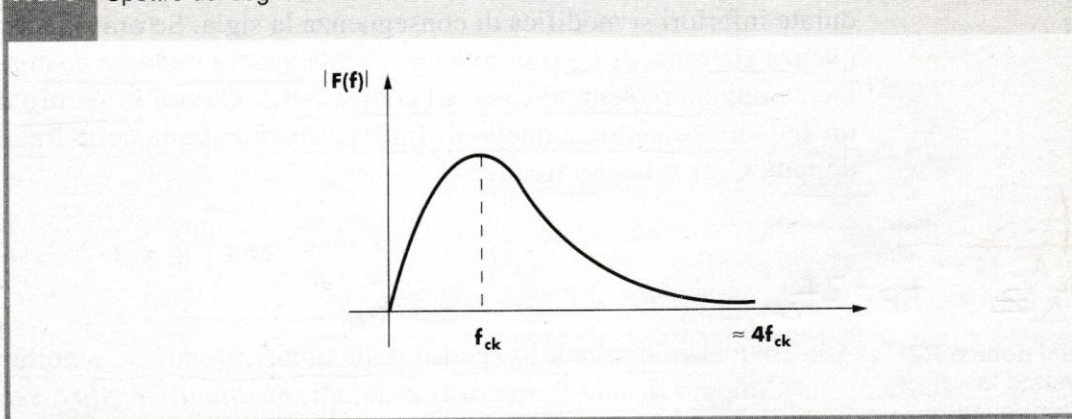
FIG. 5 Il codice bifase per il byte 01101010.



INCONVENIENTE DEL CODICE BIFASE

Di conseguenza, lo spettro del segnale contiene l'armonica fondamentale a frequenza di clock e, visto che il valore valutato su lunghe sequenze di simboli equiprobabili è nullo, la componente continua è nulla. Lo spettro ha allora una forma del tipo di quella di figura 6.

FIG. 6 Spettro del segnale con codice bifase.



La frequenza di annullamento dello spettro f_z non è identificabile (lo spettro è di ampiezza infinita) ma si può supporre che dopo $3+4 f_{ck}$ possa essere di ampiezza trascurabile.

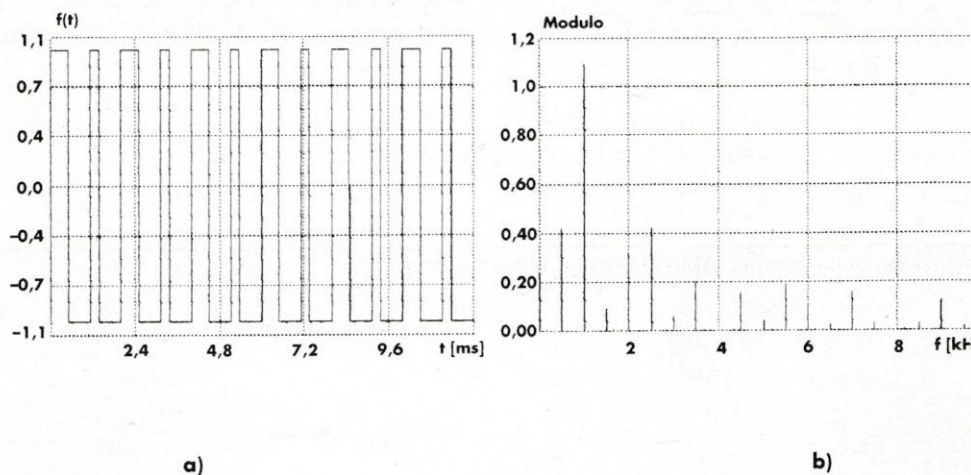
Per una verifica delle caratteristiche del codice bifase si consiglia di disegnare ciò che si ottiene nel caso di una sequenza qualsiasi di simboli binari: il risultato è un'onda quadra con duty-cycle variabile. Si provi poi, mediante il programma FOUR, a eseguire la seguente simulazione:

```
f(t)=quad(1000,t,+(0.375,*(0.125,quad(500,t,0.5))))
ti=0
tf=12e-3
```

Il codice bifase consente una facile ricostruzione del clock

Si otterrà uno spettro delle ampiezze contenente una riga molto intensa alla f_{ck} (fig. 7): l'estrazione del clock è allora un problema facilmente risolvibile. La presenza della componente continua è dovuta alla periodicità del segnale simulato con duty-cycle diverso per la parte positiva e negativa del segnale. Con un segnale casuale di lunghezza elevata a simboli equiprobabili essa si annulla completamente.

FIG. 7 Simulazione di un segnale con codice bifase (a) e relativo spettro (b).



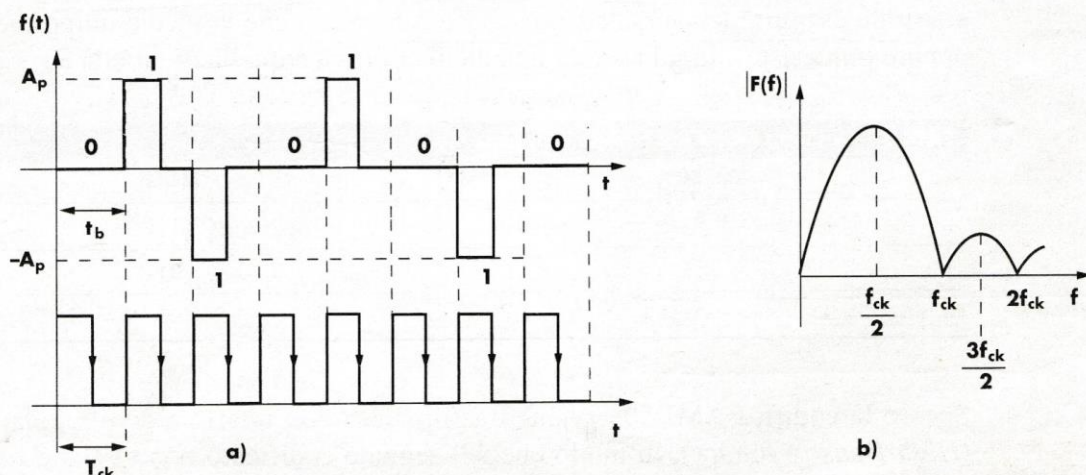
Sebbene il codice bifase abbia uno spettro di estensione maggiore rispetto al codice RZ 50% non presenta il problema della componente continua e consente una più agevole estrazione del clock. Il codice bifase è utilizzato nei casi in cui non esistono problemi di banda passante del canale come per esempio nel caso dell'informazione memorizzata su dischi magnetici. *olifetto*
è consentito di identificare l'intera sequenza di bit uguale.

Il codice AMI
deriva
dal codice RZ
.....

• **Codice AMI** (Alternate Mark Inversion). I codici NREZ e RZ presentano una elevata componente continua mentre il codice bifase è caratterizzato da uno spettro delle ampiezze troppo largo. La soluzione di tali problemi può essere ottenuta mediante il codice AMI 50% che trae origine dal codice RZ 50% mediante una inversione alternata del ciclo di clock che rappresenta il simbolo 1. Se si interpreta il simbolo 1 come presenza dell'alternativa binaria (Mark) e il simbolo 0 come la sua assenza (Space) si comprende appunto come il codice in esame effettui l'alternanza del Mark. Le dizioni Mark e Space sono legate al codice telegrafico Morse quando appunto si indicava con Mark la presenza della grandezza elettrica, mentre si parlava di Space quando la grandezza elettrica non esisteva.

Con questi presupposti il codice AMI 50%, sempre per l'esempio relativo alla figura 2a, porta alla situazione di figura 8.

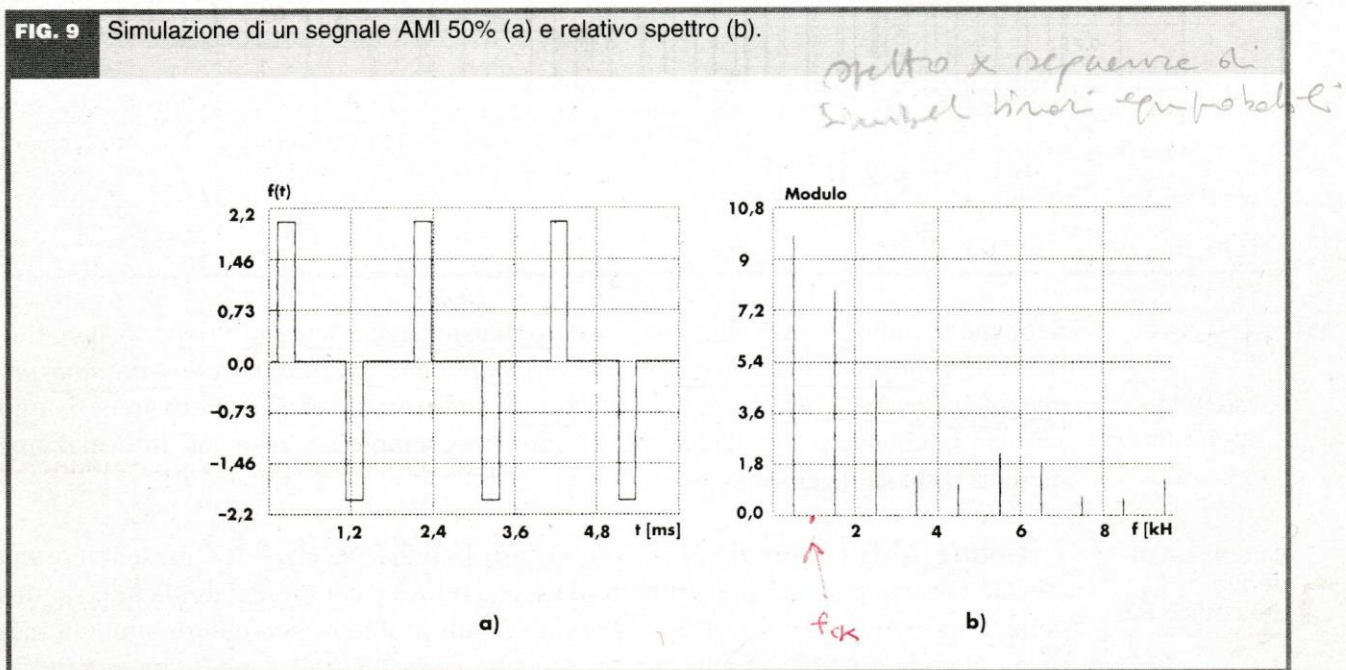
FIG. 8 Il codice AMI 50% per il byte 01101010 (a) e spettro dei segnali AMI 50% (b).



È necessario alternare tra $+A_p$ e $-A_p$ il Mark in modo che complessivamente su lunghe sequenze binarie con simboli equiprobabili la tensione continua risulti nulla. Se ora si immagina una sequenza alternata di 1 e 0, si deduce che il segnale risultante è un'onda rettangolare alternata con duty-cycle del 25% e di frequenza pari alla metà della frequenza del clock. La verifica grafica è lasciata al lettore e il risultato può essere anche controllato mediante il programma FOUR per la seguente funzione (fig. 9):

```
f(t)=*(sgn(sin(*(2,p,500,t)/(p,2))))+(1,quad(1000,t,0.25))
ti=0
tf=6e-3
```

FIG. 9 Simulazione di un segnale AMI 50% (a) e relativo spettro (b).



Il segnale contiene una riga spettrale alla frequenza pari alla metà di quella di clock, mentre alla frequenza di clock lo spettro ha ampiezza nulla. In definitiva lo spettro delle ampiezze di una codifica AMI 50%, per una sequenza estesa di simboli binari equiprobabili, assume un aspetto del tipo di figura 8b.

Non esiste la frequenza di clock ma essa può essere facilmente ottenuta mediante un raddrizzatore a onda intera. In effetti si può facilmente passare da una codifica AMI 50% a una codifica RZ 50% mediante l'operazione di raddrizzamento del segnale. Il codice AMI 50% è utilizzato nella telefonia digitale spesso in unione a una successiva codifica detta **MCMI** (*Modified Coded Mark Inversion*) necessaria per modulare gli impulsi della sorgente laser impiegata nei canali a fibra ottica. L'ulteriore codifica è necessaria nel caso di trasmissione digitale su fibra ottica poiché non è possibile eseguire l'inversione tra $+A_p$ e $-A_p$ visto che la luce è unipolare (esiste oppure non esiste). In tal caso la tabella di codifica è quella di tabella 8.

TAB. 8 - Codifica MCMI del codice AMI 50%

AMI 50%	MCMI
$+A_p$	11
0	01
$-A_p$	00

Spesso la codifica AMI 50% viene trasformata in un'ulteriore codifica, la **HDB-3** (*High Density Binary*), in modo che nel segnale codificato non esistano più di tre zeri successivi.

Modifiche
del codice AMI

RINANZIARE IL PROBLEMA
DELLA LUNGA SEQUENZA
DI BIT UGUALI