

1 IL SEGNALE

1.1 Il segnale analogico

Tutti i dispositivi per comunicare si scambiano dei segnali. Attraverso l'aria i tablet e gli smartphone, attraverso i cavi i computer desktop. I segnali portano anche le trasmissioni televisive e quelle radiofoniche, terrestri e satellitari. Con i segnali viaggiano tutti i dati necessari a trasferire suoni, immagini, video, file, pagine web, ecc. che sono rappresentati sotto forma di lunghe sequenze di bit.

Esattamente come avviene all'interno di un computer, anche per le trasmissioni a distanza occorre quindi trasferire da un punto all'altro dei bit.

L'informazione può essere trasmessa a distanza variando opportunamente una qualche caratteristica del fenomeno fisico utilizzato per la trasmissione. Associando a questa variazione uno dei due valori del bit e alla mancanza di variazione (o a una diversa variazione) l'altro valore del bit, si realizza la trasmissione dei dati.

Tale variazione si propaga, con una certa velocità, lungo il mezzo di trasmissione e dopo un certo tempo arriva all'altra estremità del mezzo, dove può essere rilevata. Per esempio, se il mezzo è un cavo elettrico, si può variare la tensione applicata a un'estremità. Tale variazione di tensione verrà successivamente rilevata all'altra estremità.

Qualunque sia il mezzo trasmissivo e il fenomeno fisico utilizzato, i segnali trasmessi possono essere di due tipi: **analogici** e **digitali**.

I **segnali analogici** possono assumere un qualsiasi valore all'interno di un determinato intervallo senza soluzione di continuità. In particolare un segnale analogico **periodico**, cioè che assume valori che si ripetono ciclicamente in modo regolare nel tempo, è particolarmente adatto a trasportare i bit.

I segnali periodici più utilizzati sono quelli **sinusoidali**, che consentono di comporre e descrivere qualsiasi altro segnale periodico (teorema di Fourier).

Ogni segnale sinusoidale fa riferimento a una grandezza che varia nel tempo e che viene scelta per descrivere il segnale, per esempio una differenza di potenziale, un'intensità di corrente o un'intensità luminosa.

Per descrivere un segnale sinusoidale periodico si utilizzano tre parametri:

- **ampiezza**: la distanza tra il valore medio e quello massimo della grandezza scelta;
- **frequenza**: il numero di volte in cui si ripete il segnale in un secondo (viene misurata in hertz);
- **fase**: intervallo di tempo, espresso in gradi, tra l'inizio di un segnale sinusoidale e un tempo prefissato preso come riferimento. Ne deriva che lo **sfasamento** tra due segnali è l'intervallo di tempo, sempre espresso in gradi, che intercorre tra due segnali sinusoidali con la stessa frequenza.

Nella **FIGURA 1** è mostrato un grafico con un segnale analogico periodico (l'asse orizzontale indica il tempo, quello verticale l'ampiezza dell'onda sinusoidale), in cui l'onda disegnata ha un'ampiezza di 5 volt a 0,25 secondi, di 0 volt a 0,5 secondi e di -5 volt a 0,75 secondi (come grandezza è stata usata la tensione da +5 V a -5 V).

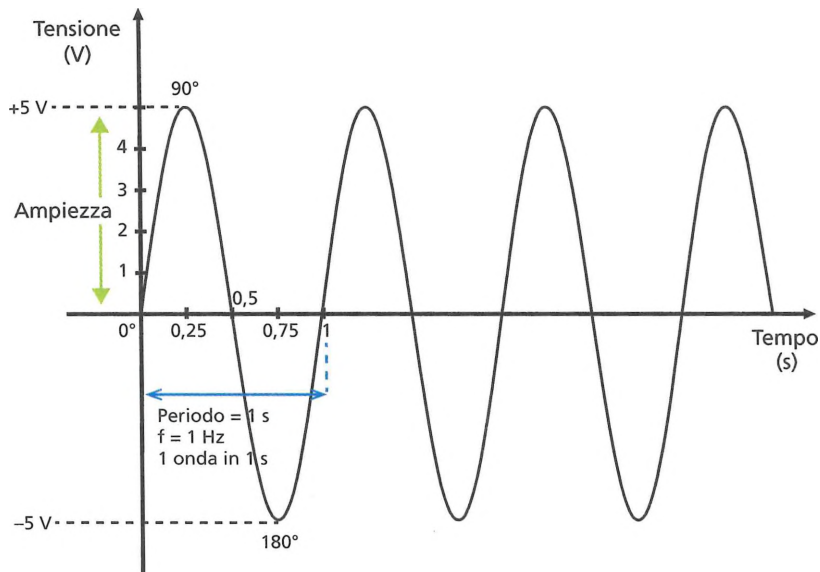


FIGURA 1 Segnale analogico sinusoidale periodico

La frequenza dell'onda è di 1 Hz, ossia di un ciclo ogni secondo. Infatti il ciclo inizia dal punto 0, prosegue fino a raggiungere l'ampiezza massima (+5), scende poi fino al punto minimo (-5) e quindi ritorna al punto iniziale; il tutto avviene nel tempo di un secondo.

Nella **FIGURA 2** sono mostrate due diverse forme d'onda che rappresentano due segnali analogici e hanno lo stesso periodo, la stessa ampiezza, la stessa frequenza, ma fase diversa. Infatti un ciclo completo equivale a 360° , dove 0° rappresenta il punto di partenza, 90° il picco positivo, 180° il punto in cui interseca l'asse orizzontale, 270° il picco negativo e 360° la conclusione del ciclo. I due segnali analogici rappresentati in figura hanno una differenza di fase di 90° . Il calcolo dello sfasamento tra due segnali con la stessa frequenza si ottiene tenendo conto che alla durata di un periodo, indicato con T, corrispondono 360° , quindi si misura l'intervallo I e il periodo T sull'asse del tempo e successivamente si imposta la proporzione:

$$I : T = \theta : 360^\circ$$

dove θ indica lo sfasamento.

Risulta quindi che $\theta = (360^\circ \cdot I)/T$ (espresso in gradi).

Nell'esempio si ha quindi: $\theta = (360^\circ \cdot 0,25)/1 = 90^\circ$.

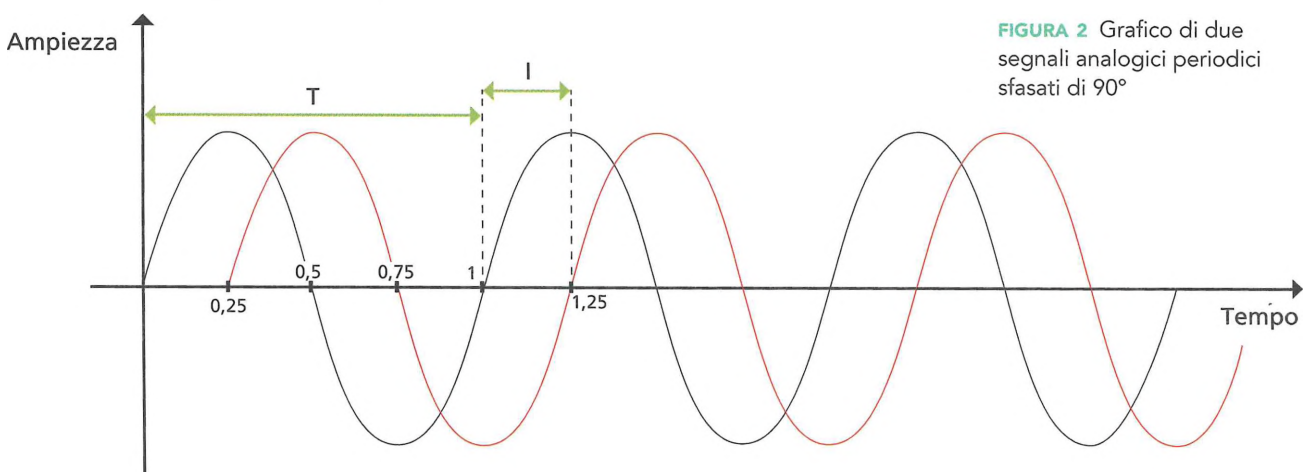


FIGURA 2 Grafico di due segnali analogici periodici sfasati di 90°

1.2 Il segnale digitale

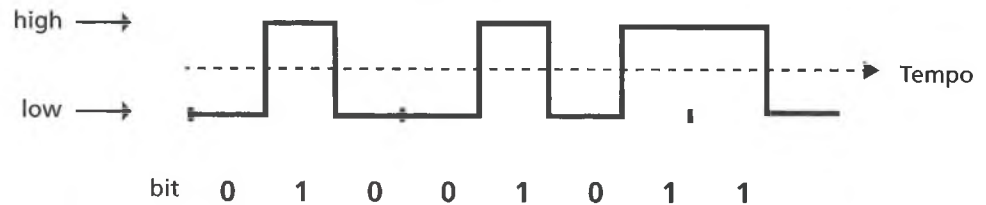
I **segnali digitali** hanno due caratteristiche che li distinguono dai segnali analogici:

- possono assumere solo un numero limitato di valori discreti (due nel caso di segnali binari);
- la transizione da un valore all'altro avviene in modo quasi istantaneo.

Per queste caratteristiche un segnale digitale è rappresentato con un'onda rettangolare, che nel caso di segnali digitali binari è costituita da due valori: uno alto (*high*) che rappresenta un bit 1 e uno basso (*low*) che rappresenta un bit 0.

Nella **FIGURA 3** possiamo vedere un segnale digitale binario costituito da 8 bit (un byte).

FIGURA 3 Segnale digitale binario

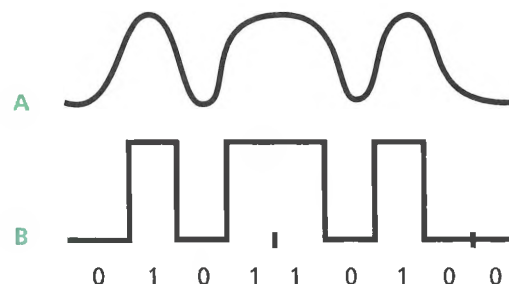


I segnali digitali, e in particolare quelli binari, godono di alcuni importanti pregi nei confronti di quelli analogici.

■ MAGGIOR ROBUSTEZZA RISPETTO AL RUMORE

Rumore e disturbi, per quanto possano essere minimi, sono sempre presenti, sia nel trasmettitore, sia nel ricevitore, ma sono soprattutto presenti nel mezzo di comunicazione, sia esso un cavo o l'aria. I segnali digitali per loro natura sono meno soggetti ai disturbi rispetto ai segnali analogici. I segnali analogici sono costituiti da funzioni continue che possono assumere infiniti valori: il rumore inevitabilmente si sovrappone al segnale trasmesso e lo modifica rendendo difficile risalire al segnale originario. I segnali digitali, invece, presentano solamente un numero finito di valori separati da un "salto" tra uno e l'altro. Tale salto è determinato dal superamento di una soglia. Se il rumore non ha ampiezza (e potenza) tale da determinare il superamento della soglia e quindi il salto tra due valori, allora il rumore non riesce ad alterare il segnale. Il destinatario, ricevendo il valore non ammissibile, provvederà a convertirlo nel valore accettabile più vicino, annullando l'effetto del disturbo (**autorigenerazione**). Naturalmente se il rumore fosse tale da far superare la soglia, il destinatario sarebbe indotto in errore e il segnale verrebbe convertito in un valore sbagliato in assenza di tecniche di correzione. Una delle ragioni che fanno preferire l'utilizzo dei segnali digitali rispetto a quelli analogici è proprio l'autorigenerazione: i segnali digitali contengono errori che, al di sotto di una certa soglia, si autocorreggono. In un segnale digitale binario solo parzialmente modificato da un disturbo (**FIGURA 4A**), proprio la limitazione ai soli valori "0" e "1" rende i bit ancora riconoscibili (**FIGURA 4B**).

FIGURA 4 Segnale digitale binario autorigenerato



■ MAGGIOR INTEGRAZIONE DEI SISTEMI DI TRASMISSIONE

La conversione in digitale di tutte le trasmissioni (audio, video, dati, testo, ecc.) rende ogni invio simile a tutti gli altri: un flusso di bit con la sola aggiunta di intestazioni (bit di *header*) che specificano il tipo di informazioni trasmesse in ogni blocco. Qualunque apparato dotato di connessione di rete è in grado di inviare e ricevere qualsiasi tipo di trasmissione digitale. Anche il mezzo fisico (cavo, fibra, aria) e la tecnologia (elettrica, ottica, wireless) diventano irrilevanti. I flussi di bit, raggruppati in byte e poi in pacchetti (*packet*) aventi regole precise, uniformano le fasi di invio, trasferimento e ricezione.

■ MAGGIOR ADATTAMENTO A ESSERE ESEGUITI E MEMORIZZATI

Il linguaggio dei segnali digitali è lo stesso dei microprocessori: un linguaggio binario. Con le opportune interfacce, le sequenze di bit trasmesse possono facilmente essere lette ed eventualmente eseguite o memorizzate.

I dispositivi di memoria sono in grado di conservare grosse quantità di dati con l'utilizzo di tecniche digitali. La presenza di due soli stati fisici nei supporti di memoria, associati rispettivamente a 0 e 1, rende la tecnologia dei circuiti integrati in grado di leggere e scrivere sempre più velocemente.

■ MAGGIORE ADATTABILITÀ A TECNICHE DI ELABORAZIONE DEL SEGNALE

Mentre l'elaborazione dei segnali analogici è generalmente limitata alle operazioni di amplificazione, di modulazione e di filtraggio, l'elaborazione dei segnali digitali può consentire operazioni complesse:

- **rivelazione e correzione degli errori:** nei sistemi digitali si possono realizzare circuiti e algoritmi per la rivelazione e la correzione degli errori in trasmissione, come vedremo nella Lezione 4 di questa unità;
- **crittografia:** con un sistema digitale l'informazione è codificata ed è quindi possibile adottare forme di crittografia per rendere incomprensibili, tranne che al destinatario, le informazioni trasmesse;
- **incapsulamento:** protezione dei dati in trasmissione con l'aggiunta di ulteriori header;
- **privacy:** modifica degli indirizzi privati dei mittenti e dei destinatari;
- **compressione:** riduzione della quantità di dati digitali da trasmettere, comprimendo opportunamente il segnale ed evitando di ripetere l'invio delle informazioni che si ripetono uguali (ad esempio le immagini in sequenza hanno spesso sfondi che restano a lungo invariati).

FISSA LE CONOSCENZE

- Descrivi le differenze tra segnale analogico e segnale digitale (massimo 5 righe).
- Quali sono i tre parametri che descrivono un segnale analogico sinusoidale?
- Che cosa si intende per sfasamento tra due segnali analogici?
- Quali valori assume un segnale digitale binario?
- Che cosa si intende con autorigenerazione di un segnale digitale?

2 LE MODULAZIONI DIGITALI

2.1 Modulare e demodulare

La trasmissione dati si basa sulla trasmissione digitale, però ci possono essere casi in cui il tipo di connessione utilizzata è in grado di trasmettere solo segnali analogici (per esempio le linee telefoniche sono state progettate per trasportare segnali analogici). Se ci si connette a una rete geografica, come Internet, tramite una linea telefonica, i dati che escono dal computer (digitali) devono essere convertiti in segnali analogici prima di essere inviati sulla linea telefonica. In caso di ricezione dati avviene il processo inverso: i segnali analogici che giungono dalla linea devono essere convertiti nella forma originaria digitale. Questo processo è chiamato **modulazione/demodulazione** ed è realizzato tramite un apparato denominato **modem**.

Nella **modulazione**, un'onda, chiamata **segnale portante** (carrier signal), è combinata con un altro segnale, chiamato **segnale modulante** (modulating signal), per produrre un unico segnale che trasporta l'informazione da un sistema a un altro. Quando il segnale modulante è combinato al segnale portante, esso modifica un parametro del segnale portante, per esempio la frequenza o l'ampiezza o la fase. Il risultato è un nuovo segnale che viene inviato sul mezzo trasmissivo e quando arriva a destinazione l'apparato ricevente separa il segnale modulante da quello portante (**demodulazione**), ricostruendo in tal modo il segnale originario.

Le modulazioni si basano sulla trasformazione della sequenza di bit in un segnale digitale modulante che, opportunamente combinato con una portante analogica sinusoidale, origina il segnale analogico da trasmettere.

Le modulazioni digitali fondamentali sono:

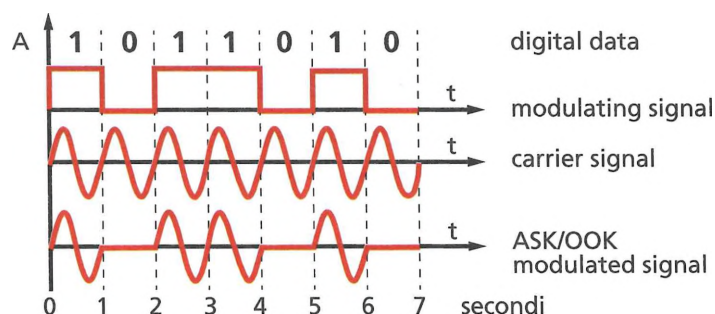
- **ASK** (Amplitude Shift Keying): modulazioni a cambiamento di ampiezza;
- **FSK** (Frequency Shift Keying): modulazioni a cambiamento di frequenza;
- **PSK** (Phase Shift Keying) e **DPSK** (Differential PSK): modulazioni a cambiamento di fase;
- **QAM** (Quadrature Amplitude Modulation): modulazioni di più bit alla volta.

2.2 ASK (Amplitude Shift Keying)

Nella ASK il segnale modulante decide tra due diverse ampiezze di una portante sinusoidale, che per semplicità di rappresentazione grafica supporremo a frequenza 1 Hz (un solo segnale in un periodo).

La più usata, mostrata in **FIGURA 5**, è la **OOK** (On-Off Keying), che associa all'1 la presenza della portante (con ampiezza invariata) e allo 0 l'assenza della portante (ampiezza zero).

FIGURA 5 Modulazione OOK di tipo ASK



È possibile realizzare modem che lavorano su due diversi valori di ampiezza, sempre da associare all'1 e allo 0.

La ASK è stata la prima modulazione a essere usata (telescriventi e ponti radio) ed è la più semplice da realizzare. È però caduta in disuso per l'estrema sensibilità al rumore.

2.3 FSK (Frequency Shift Keying)

Nella FSK il segnale modulante decide tra due portanti sinusoidali a frequenze diverse (FIGURA 6). Alla frequenza di una portante è associato l'1, alla frequenza dell'altra lo 0. Nella figura si sono usate la frequenza 1 Hz per i bit a 1 e la frequenza 2 Hz (due cicli in un periodo) per i bit a 0. La FSK è stata utilizzata dai primi modem per Internet (voce e dati) e per le trasmissioni tra cellulari GSM.

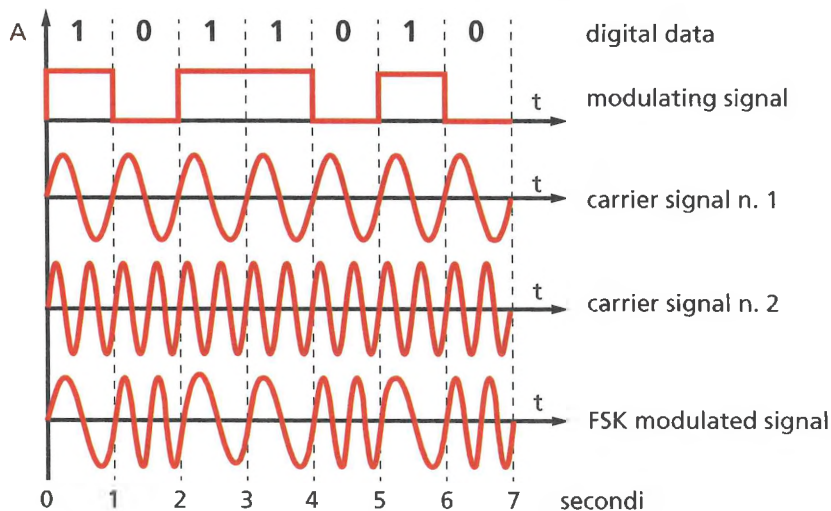


FIGURA 6 Modulazione di tipo FSK

2.4 PSK (Phase Shift Keying)

Nella PSK il segnale modulante decide tra due diverse fasi di una portante sinusoidale (FIGURA 7).

A una fase della portante è associato l'1, all'altra fase lo 0.

La PSK utilizza le fasi 0° (nessuno sfasamento della portante) e 180° (sfasamento di 180° della portante) rispettivamente per l'1 e lo 0 ed è la più attuale tra le modulazioni fondamentali.

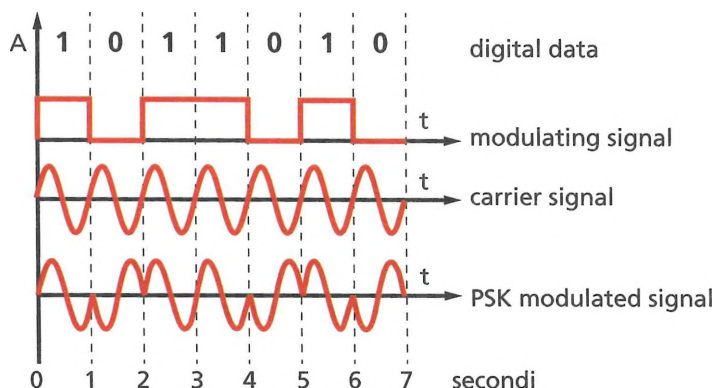
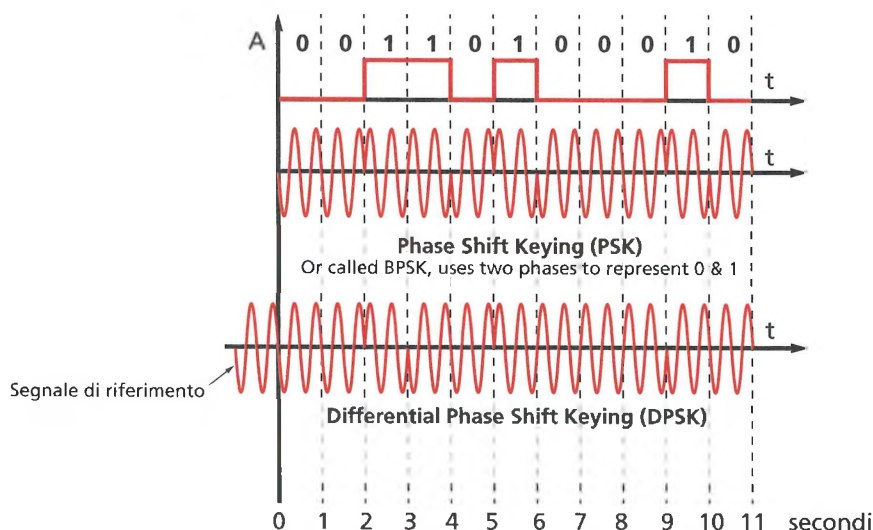


FIGURA 7 Modulazione di tipo PSK

Esiste anche una variante della PSK, chiamata **DPSK** (Differential PSK), che, invece di associare un'onda ben precisa e sempre la stessa a ogni bit, segue una regola basata sullo **sfasamento rispetto al bit precedente**: al bit 0 non corrisponde alcun cambio di fase rispetto al bit precedente, al bit 1 corrisponde un cambio di fase di 180° rispetto al bit precedente. Serve quindi un **segnale di riferimento** prima del primo bit da trasmettere, poi ogni bit fa da riferimento per il bit successivo. Mittente e destinatario devono accordarsi sul segnale di riferimento da usare. Quindi con la DPSK, a differenza di tutte le altre tecniche viste, uno stesso bit può essere rappresentato a volte da un'onda e a volte da un'onda opposta: dipende dal bit che lo precede.

La **FIGURA 8** mostra il confronto tra le due modulazioni. In questa figura ogni bit è rappresentato da un segnale con una frequenza di 2 Hz..

FIGURA 8 Modulazione di tipo PSK e DPSK a confronto



Le tre modulazioni digitali fondamentali appena viste hanno una caratteristica comune: modulano un bit per volta, cioè associano un'onda o una regola (di sfasamento) a un singolo bit.

In altre parole, ogni periodo del segnale sinusoidale trasmesso porta un bit di informazione.

È possibile fare in modo che un'onda porti più di un bit d'informazione attraverso le modulazioni QAM.

2.5 QAM (Quadrature Amplitude Modulation)

Le prime modulazioni di tipo QAM trasportavano 2 bit per volta utilizzando 4 fasi e creando quindi un'associazione del tipo PSK, come per esempio:

- 00 $\rightarrow 0^\circ$
- 01 $\rightarrow 90^\circ$
- 10 $\rightarrow 180^\circ$
- 11 $\rightarrow 270^\circ$

nota come 4-QAM dibit (**FIGURA 9**). Il modem realizza uno **sfasamento** tra un segnale e l'altro pari ai gradi indicati. Il procedimento è dunque simile alla DPSK, ma utilizzando 4 fasi anziché 2 è possibile trasmettere una coppia di bit anziché un bit solo. Quindi, anche in questo caso, una coppia di bit non è sempre espressa dalla stessa onda e serve un segnale di riferimento per la prima coppia di bit.

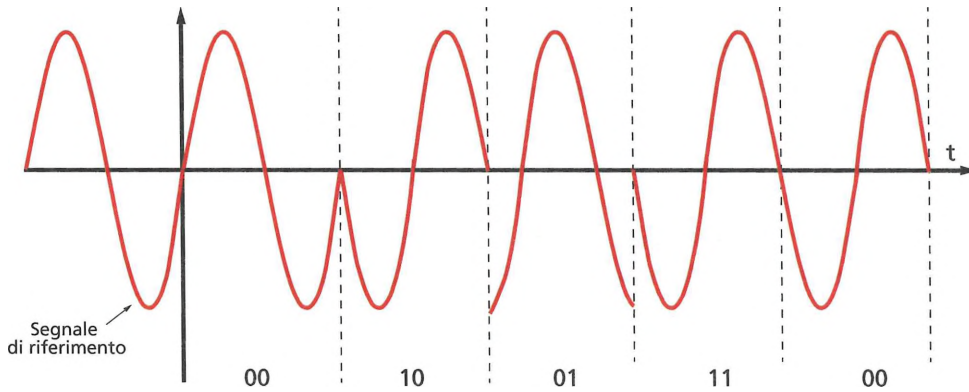


FIGURA 9 Modulazione di tipo 4-QAM dibit

Con 8 fasi (usando per esempio i valori multipli di 45°) si possono trasportare 3 bit per modulazione (8-QAM tribit), con 16 fasi 4 bit (16-QAM quadribit) e così via.

Vale la regola:

$$\text{numero fasi} = 2^{(\text{numero bit})} \quad \text{oppure} \quad \text{numero bit} = \log_2(\text{numero fasi})$$

Poiché c'è la necessità di trasmettere ancora più bit per volta senza correre rischi di errore in demodulazione dovuti a sfasamenti troppo ravvicinati, si è pensato di aggiungere contemporaneamente una variazione di ampiezza (PSK + ASK).

Le modulazioni QAM sono le uniche che modulano più di un bit per volta, associandoli a più valori di una stessa grandezza (di solito la fase) oppure agendo su due grandezze (di solito ampiezza e fase), mentre ogni altra modulazione modifica una sola delle grandezze che caratterizzano un segnale analogico (o ampiezza o frequenza o fase) e lascia inalterate le altre due.

Un esempio è dato dalla 16-QAM quadribit, realizzata con 12 fasi e 4 ampiezze ($\pm A$ e $\pm 3A$), che dà origine alla *costellazione* mostrata in FIGURA 10.

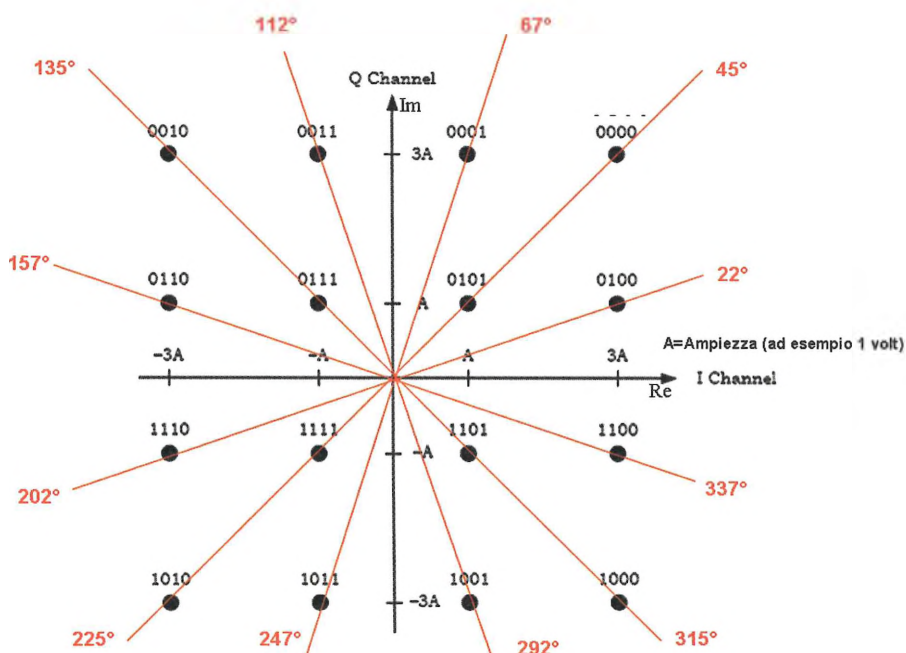


FIGURA 10 Costellazione della modulazione di tipo 16-QAM quadribit

I valori di ampiezza sono rappresentati in componente reale e immaginaria rispettivamente sugli assi I Channel e Q Channel.

Per esempio, la stringa di dati digitale 1001 è associata a un'onda sfasata di 292° e con ampiezza A in componente reale e $-3A$ in componente immaginaria.

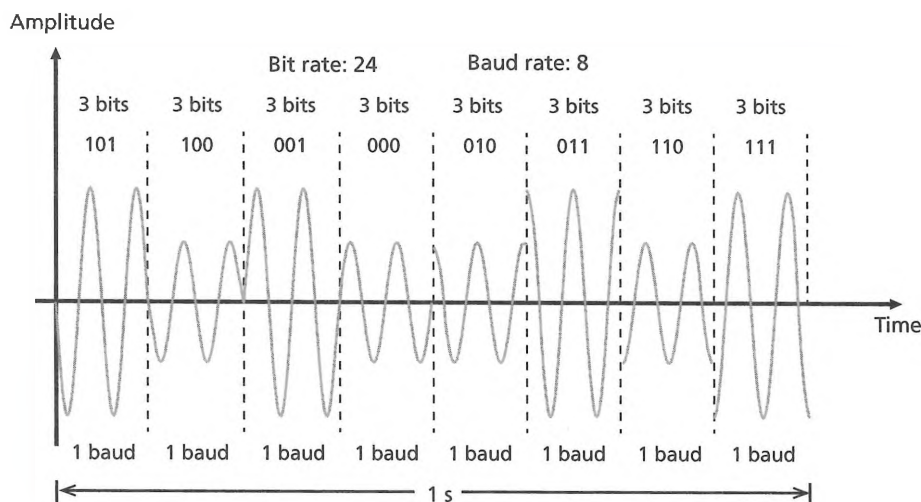
Le modulazioni QAM, che consentono la trasmissione di più bit con un'unica modulazione, introducono il concetto di #baud rate.

Nella FIGURA 11 vediamo un esempio di 8-QAM tribit (4 fasi e due ampiezze), che in un secondo trasmette 24 bit (bit rate = 24 bps) con un baud rate di 8 baud/s.

#techwords

Il **baud rate** è il numero di segnali diversi trasmessi in un secondo.

FIGURA 11 Bit rate e baud rate



IN ENGLISH PLEASE

Bit rate is the number of bits per second. Baud rate is the number of signal units per second. Baud rate is less than or equal to the bit rate.

Example 1

An analog signal carries 4 bits in each signal unit. If 1000 signal units are sent per second, find the baud rate and the bit rate.

Solution

Baud rate = 1000 baud/s

Bit rate = $1000 \times 4 = 4000$ bps

Example 2

The bit rate of signal is 3000. If each signal unit carries 6 bits, what is the baud rate?

Solution

Boud rate = $3000/6 = 500$ baud/s

FISSA LE CONOSCENZE

- Che cosa si intende con modulazione di un segnale?
- Descrivi, anche aiutandoti con un grafico, come funziona la FSK (*Frequency Shift Keying*).
- Descrivi, anche aiutandoti con un grafico, come funziona la DPSK (*Differential Phase Shift Keying*).
- Che differenza c'è tra bit rate e baud rate?

3 IL CANALE DI COMUNICAZIONE

3.1 Il multiplexing del canale

Quando un segnale deve essere trasmesso, viene inviato attraverso un **#canale**, cioè un mezzo fisico di trasmissione.

I canali possono essere logici o fisici: sono **logici** quando si realizzano più percorsi distinti utilizzando lo stesso mezzo fisico (per esempio lo stesso cavo di rame), sono **fisici** quando si realizzano più percorsi distinti utilizzando mezzi fisici diversi.

La tecnica che consente di separare un mezzo fisico in più canali logici viene detta **multiplexing** e permette di far viaggiare più segnali simultaneamente su uno stesso mezzo fisico.

Esistono diverse tecniche per effettuare il multiplexing dei segnali: TDM (Time Division Multiplexing), FDM (Frequency Division Multiplexing), WDM (Wavelength Division Multiplexing).

#techwords

Un **canale** (channel) è un percorso definito che mette in comunicazione due nodi.

■ TDM (TIME DIVISION MULTIPLEXING)

Questa tecnica prevede di suddividere ogni frame in intervalli (slot) e assegnare ogni slot a un dispositivo di input (FIGURA 12). In questo modo ogni frame inviato contiene una parte di dati di ciascun mittente. Il risultato è quello di portare avanti più trasmissioni simultaneamente, distribuendo il canale fisico, anziché concederlo a turno a un solo mittente per volta.

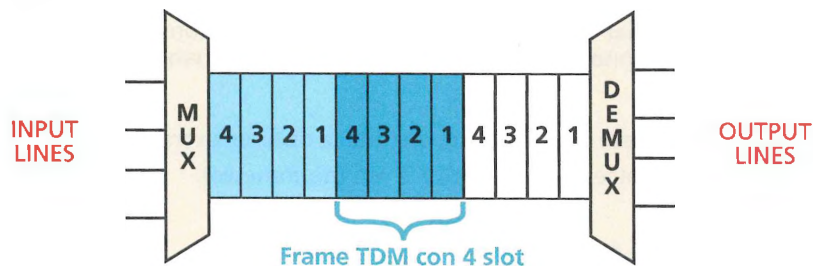


FIGURA 12 Tecnica TDM

Il dispositivo di multiplexing (MUX) è dotato di un buffer, in cui prepara il frame prima di inviarlo. Qualora non tutte le linee di input avessero dati da trasmettere, il MUX può comunque trasmettere il frame con bit riempitivi (padding) o riempirlo, assegnando più slot a uno stesso mittente. Il dispositivo di demultiplexing (DEMUX) mantiene su ciascuna linea di output lo slot relativo al destinatario corrispondente.

■ FDM (FREQUENCY DIVISION MULTIPLEXING)

Questa tecnica prevede invece di suddividere il canale in **sottocanali**, uno per ciascun mittente (FIGURA 13).

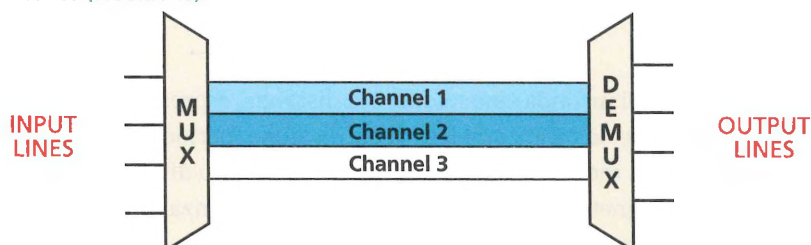
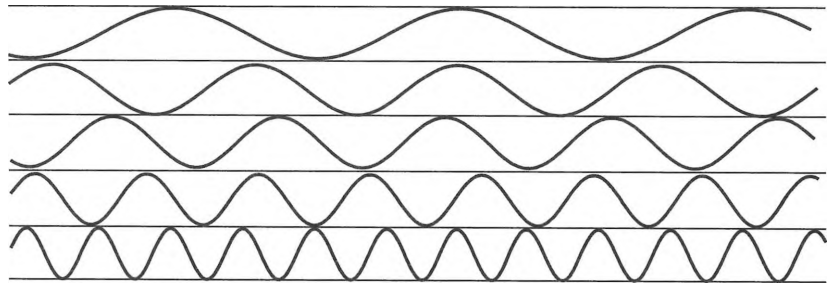


FIGURA 13 Tecnica FDM

In pratica, ogni sottocanale lavora a frequenze diverse dagli altri (FIGURA 14), evitando interferenze reciproche.

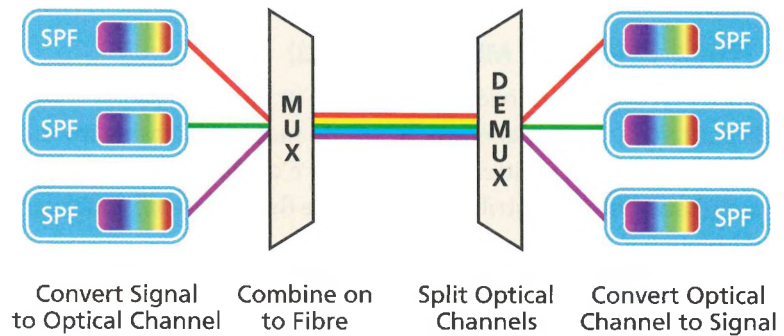
FIGURA 14 Sottocanali a frequenze diverse



■ WDM (WAVELENGTH DIVISION MULTIPLEXING)

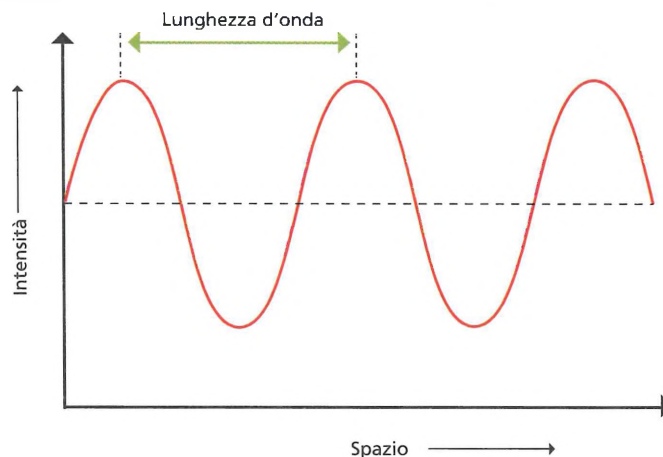
È una tecnica implementata nelle attuali reti in fibra ottica, che permette di usare in simultanea su una singola fibra tra gli 80 e i 160 canali logici, trasmessi su diverse lunghezze d'onda (FIGURA 15).

FIGURA 15 Tecnica WDM



Il multiplexing può essere di tipo denso, quando la differenza tra le lunghezze d'onda (FIGURA 16) di canali adiacenti è inferiore a 1 nm (nanometro).

FIGURA 16 Lunghezza d'onda



La lunghezza d'onda di un'onda sinusoidale è la distanza, espressa in nanometri, percorsa dall'onda durante un ciclo completo di oscillazione (periodo). Il nanometro (nm) è un sottomultiplo del metro ed equivale a un miliardesimo di metro: $1 \text{ nm} = 10^{-9} \text{ m}$. In un'onda elettromagnetica tra lunghezza d'onda e frequenza esiste un rapporto di proporzionalità inversa.

IN ENGLISH PLEASE

The **WDM** unit consists of a Signal Processing Facility (**SPF**), a multiplexer (**MUX**) which is responsible for joining the signals at the transmitting end, and a demultiplexer (**DEMUX**) which is responsible for splitting the signals at the receiving end.

CWDM and DWDM

There are two types of WDM: **Coarse** and **Dense** Wavelength Division Multiplexing (CWDM and DWDM).

CWDM uses a wide spectrum and accommodates eight channels. This wide spacing of channels allows for the use of moderately priced optics, but limits capacity. CWDM is typically used for lower-cost, lower-capacity, shorter-distance applications where cost is the paramount decision criteria.

DWDM systems pack 16 or more channels into a narrow spectrum window very near the 1550 nm local attenuation minimum. Decreasing channel spacing requires the use of more precise and costly optics, but allows for significantly more scalability. Typical DWDM systems provide 1-44 channels of capacity, with some new systems, offering up to 80-160 channels. DWDM is typically used where high capacity is needed over a limited fiber resource or where it is cost prohibitive to deploy more fiber.

The Cisco® enhanced wavelength-division multiplexing (EWDM) product line allows users to scale the speed and capacity of the services offered in a coarse wavelength-division multiplexing (CWDM) network by offering the ability to insert up to 8 dense wavelength-division multiplexing (DWDM) wavelengths to the existing 8-wavelength CWDM channel plan.

Product Overview

The Cisco EWDM product line provides the ability to overlay up to 8 DWDM wavelengths with the 8 CWDM channels (1470, 1490, 1510, 1530, 1550, 1570, 1590, and 1610 nm). The principle is very simple, yet it is a unique approach in that the 8 DWDM wavelengths are inserted in between CWDM channels. EWDM allows 5 DWDM channels to be multiplexed between the 1530-nm and 1550-nm CWDM wavelengths and 3 DWDM channels between the 1550-nm and 1570-nm CWDM wavelengths. A total of 8 CWDM plus 8 DWDM wavelengths can be supported on the same fiber infrastructure (see FIGURES 17, 18, 19).

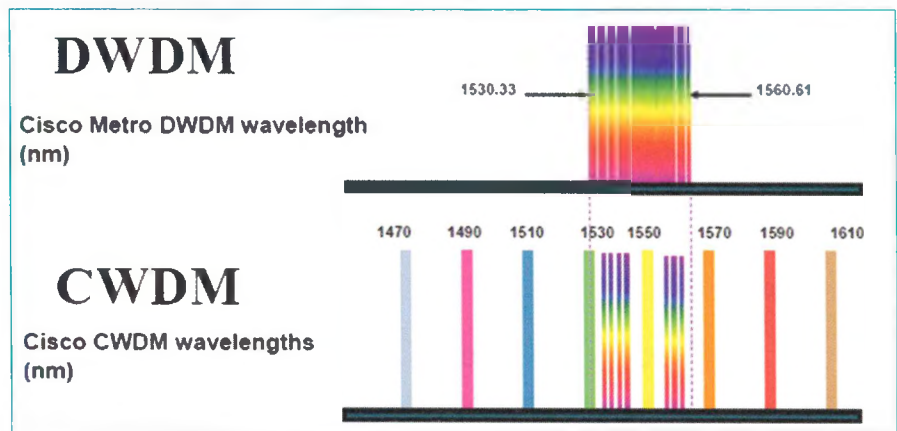


FIGURA 17 Cisco EWDM Concept

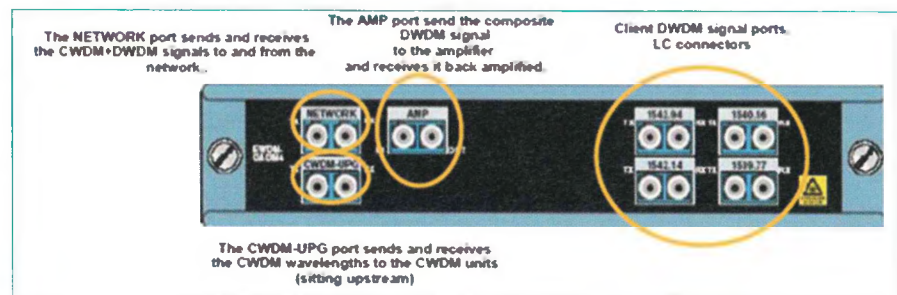


FIGURA 18 Cisco EWDM Passive Unit Front Panel Layout

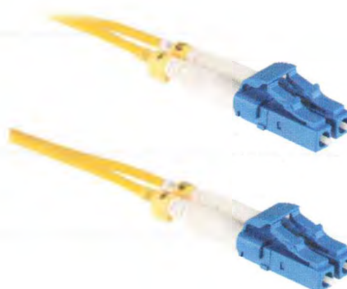


FIGURA 19 LC optical connector

3.2 La codifica di linea

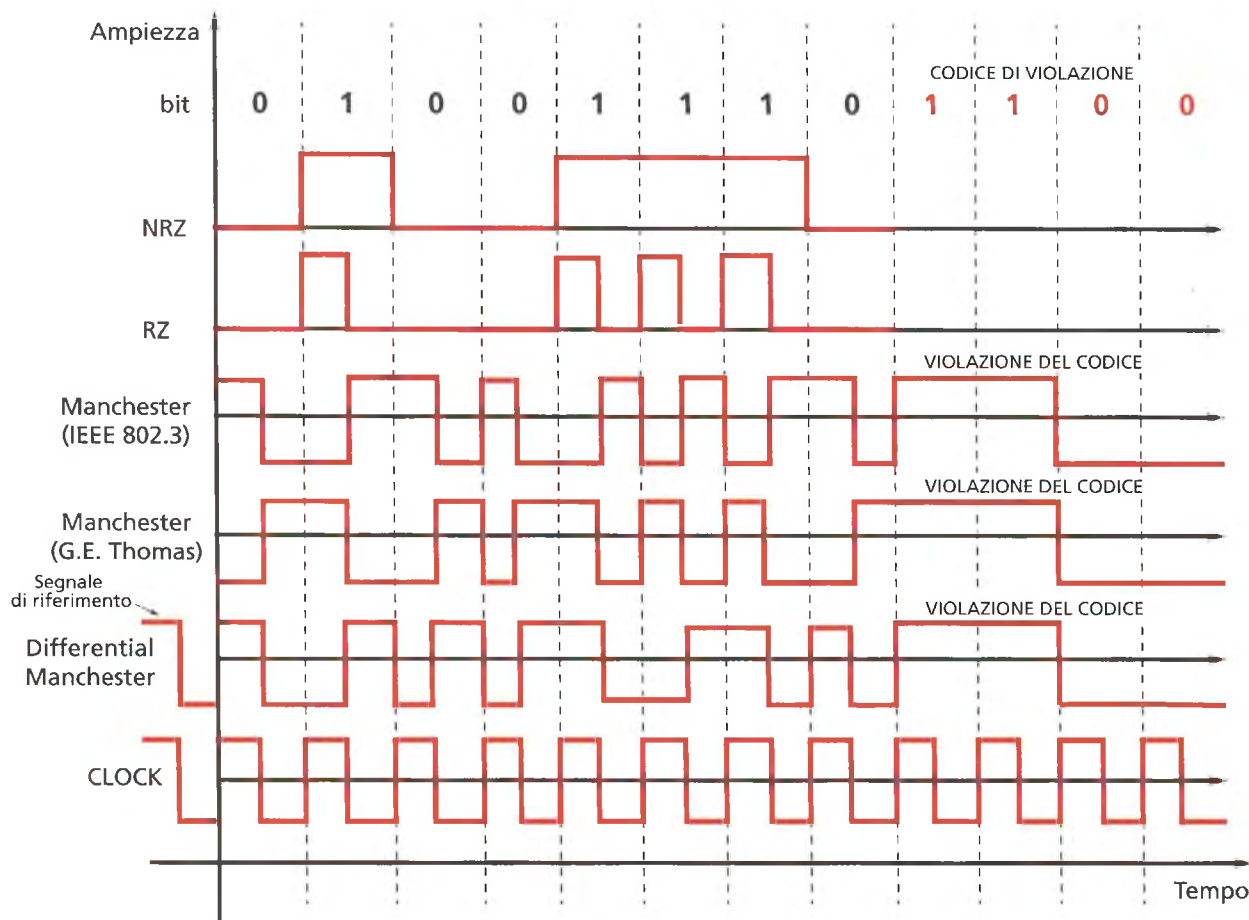
I dati da trasmettere sul canale devono essere prima trasformati in un segnale elettrico con la **codifica di linea** (in ricezione si utilizza la **decodifica di linea**) che serve ad adattare il segnale fisico digitale al particolare mezzo trasmissivo utilizzato. Inoltre, la codifica di linea deve permettere di mantenere il **sincronismo** tra trasmettitore e ricevitore.

Il segnale di sincronismo è il segnale di clock che sincronizza le schede di rete delle macchine, quindi una caratteristica importante della codifica di linea è quella di mettere insieme i dati con il segnale di sincronismo, cosa che permette al ricevitore di effettuare correttamente la decodifica del segnale ricevuto.

La trasmissione dei dati in forma digitale prevede che ai valori 1 e 0 dei bit da inviare si associno determinati valori del fenomeno fisico che è stato scelto per la trasmissione (per esempio la differenza di potenziale). La semplice scelta di associare due differenti valori fisici (uno per lo 0 e uno per l'1) non è quella ottimale, in quanto comporta problemi di sincronismo tra trasmettitore e ricevitore.

FIGURA 20 Le differenti codifiche di linea per una stessa sequenza di bit

Le tre principali tecniche di codifica sono: **NRZ** (Not Return to Zero), molto semplice, usata nei computer e nelle centrali numeriche, **RZ** (Return to Zero), usata nelle centraline telefoniche, **Manchester** e le sue varianti, usate soprattutto nelle reti dati locali. La **FIGURA 20** riassume tutte le tecniche di seguito descritte.



■ NRZ (NOT RETURN TO ZERO)

È la codifica più semplice e associa un valore alto al bit 1 e un valore basso al bit 0. Si tratta di un metodo che non richiede circuiti complicati perché i dati che entrano come 1 o 0 vengono passati direttamente all'uscita senza modifiche. Il problema principale di questa codifica è la difficoltà a mantenere il sincronismo a fronte di una lunga sequenza di bit uguali (tutti 1 o tutti 0) che porta il segnale ad avere lo stesso valore per un lungo intervallo di tempo: un minimo disallineamento nel clock del ricevitore comporterà un'interpretazione errata della sequenza di bit.

■ RZ (RETURN TO ZERO)

È simile a NRZ, con la differenza di portare il segnale a zero a ogni semiperiodo. Il bit 1 è quindi rappresentato da un valore alto per metà del periodo di clock e poi da un valore basso per la restante metà. Questa codifica risolve il problema di lunghe sequenze di bit 1 (valore alto) ma non di quelle di bit 0 (valore basso).

■ MANCHESTER

In questa codifica, definita all'Università di Manchester da cui prende il nome, il segnale di clock del trasmettitore e il segnale dei dati vengono combinati per garantire una transizione per ogni bit. Esistono due opposte convenzioni, entrambe con numerosi sostenitori, su come rappresentare il bit 1 e il bit 0:

- la prima è specificata nello standard Ethernet (IEEE 802.3) da cui il nome Manchester 802.3 e afferma: il bit 1 è rappresentato da una transizione basso-alto al semiperiodo e il bit 0 da una transizione alto-basso al semiperiodo;
- la seconda è quella proposta da G.E. Thomas, che specifica l'opposto: il bit 1 è rappresentato con una transizione alto-basso al semiperiodo, il bit 0, viceversa, è rappresentato con una transizione basso-alto al semiperiodo.

Con questa codifica si elimina il problema delle lunghe sequenze di bit con uguale valore: infatti il sincronismo tra trasmettitore e ricevitore è mantenuto grazie alle continue transizioni.

Per contro, la sua efficienza è molto inferiore rispetto alle precedenti in quanto per ogni bit da trasmettere vengono trasferiti due valori e quindi il consumo di banda è doppio. Un importante vantaggio della codifica di Manchester, tuttavia, è che permette di essere violata: il trasmettitore può emettere una sequenza di bit 1 o 0 senza effettuare la transizione, così da fornire un'informazione che il ricevitore può facilmente codificare come fine del messaggio. Un tipico esempio di **codice di violazione** è la sequenza **1100**, trasmessa senza i cambi di fronte alto-basso o basso-alto.

L'ambiguità su quale rappresentazione utilizzare è superata dalla codifica detta **Differential Manchester Coding**. In questa codifica la transizione usata per codificare il dato è all'inizio del periodo invece che nel semiperiodo. Quindi una transizione all'inizio di un bit rappresenta uno 0, mentre l'assenza della transizione all'inizio rappresenta un 1. Rimane comunque, come nella normale codifica di Manchester, la transizione nel semiperiodo. Occorrerà però un segnale di riferimento iniziale per codificare il primo bit in trasmissione, come nelle modulazioni DPSK e QAM.

#prendinota

Violare il codice di Manchester vuol dire non fare la transizione a metà periodo.

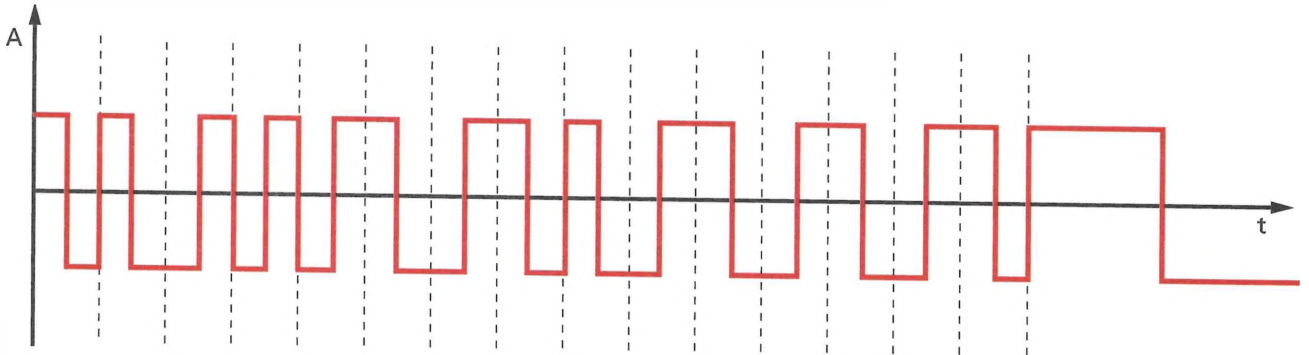
#prendinota

Il codice di violazione usato dalle codifiche di Manchester (e non dalle altre codifiche) indica la fine della trasmissione.

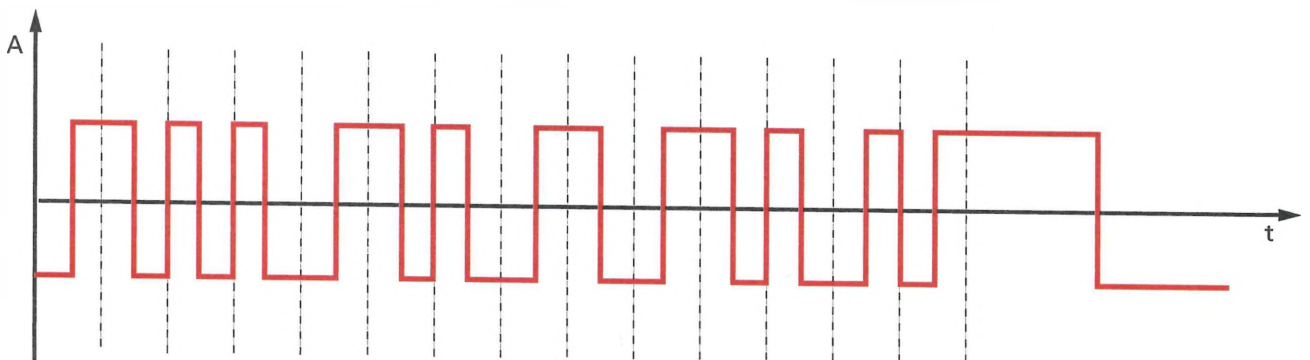
esercizio**→ PROBLEMA**

Codificare secondo le regole della codifica di Manchester (nella versione di G.E. Thomas) la seguente sequenza di bit:

1100010110101011100

→ SVOLGIMENTO**esercizio****→ PROBLEMA**

Data la seguente successione di bit codificati secondo la codifica di Manchester (nella versione di G.E. Thomas), determinare il flusso di bit ricevuto.

**→ SVOLGIMENTO**

La sequenza di bit ricevuti è:

011101101011001100

3.3 Caratteristiche di una trasmissione dati

Come si è visto, una trasmissione dati consiste nell'invio di segnali da un nodo trasmettitore a un nodo ricevitore.

Gli scenari che vengono a crearsi sono molteplici, in quanto ogni trasmissione, e quindi ogni rete, si caratterizza in base a vari parametri e modalità di trasmissione. Nel seguito sono presentati i principali.

■ SIMPLEX E DUPLEX

La trasmissione dati, sia essa analogica o digitale, è caratterizzata dalla direzione in cui viaggiano i segnali sui mezzi trasmissivi:

- **trasmissione simplex:** i segnali possono viaggiare in una sola direzione. Un esempio è l'uso del megafono per parlare a molte persone: la voce viaggia in una sola direzione;
- **trasmissione half-duplex:** i segnali possono viaggiare in entrambe le direzioni in un mezzo trasmissivo, ma in una sola direzione alla volta. Un esempio è il walkie-talkie;
- **trasmissione full-duplex:** i segnali possono viaggiare in entrambe le direzioni contemporaneamente; spesso queste trasmissioni sono dette bidirezionali o, semplicemente, duplex. Un esempio è la trasmissione telefonica: chiamato e chiamante possono ascoltare e parlare in simultanea usando la stessa linea telefonica.

La trasmissione full-duplex è tipica delle reti dati e spesso si associa all'utilizzo di più canali sullo stesso mezzo fisico.

Per esempio, se si utilizzano due distinti fili, uno per trasmettere e uno per ricevere, ciascun filo consente una trasmissione half-duplex, e mettendo insieme questi due fili in un cavo, si ottiene un mezzo fisico che consente una trasmissione full-duplex.

L'impiego di mezzi trasmissivi full-duplex aumenta la velocità con cui i dati sono trasportati nella rete ed è una necessità, per esempio, per fornire un servizio di telefonia su Internet. Molti apparati di rete come i modem e le schede di rete consentono di specificare se si usa una connessione half-duplex o full-duplex.

■ POINT-TO-POINT E POINT-TO-MULTIPOINT

Un'altra importante caratteristica delle comunicazioni è il numero di nodi trasmettitori e ricevitori coinvolti in una stessa trasmissione. Si distinguono due casi:

- **point-to-point:** è un tipo di trasmissione che coinvolge solo due nodi, uno che trasmette e uno che riceve; questo scenario viene anche indicato come trasmissione di tipo **unicast**;
- **point-to-multipoint:** questa trasmissione coinvolge un trasmettitore e molti ricevitori e a sua volta si distingue in:
 - **broadcast:** la trasmissione avviene tra un trasmettitore e molti ricevitori sconosciuti senza preoccuparsi se il segnale trasmesso potrà essere usato dalla stazione ricevente (un esempio tipico è una stazione televisiva che trasmette un programma a migliaia di antenne riceventi; il trasmettitore non potrà sapere esattamente con chi ha comunicato, ossia chi ha ricevuto il segnale e visto il programma televisivo); questo tipo di trasmissione può essere usato sia nelle reti via cavo (wired) che senza fili (wireless) in quanto è molto semplice da realizzare e veloce;
 - **multicast:** in questo tipo di trasmissione un nodo invia i segnali a un insieme ben definito di ricevitori (per esempio un amministratore di rete decide quali workstation della rete locale possono ricevere un certo video).

IN ENGLISH PLEASE

Originally, all broadcasting was composed of analog signals using analog transmission techniques and more recently broadcasters have switched to digital signals using digital transmission.

■ THROUGHPUT E BANDWIDTH

Un elemento molto importante nella trasmissione dati è la velocità di trasmissione, misurata in **bit per secondo (bps)**, che indica il numero di bit trasmessi in un secondo (TABELLA 1).

TABELLA 1 Unità di misura della velocità di trasmissione di una linea

Unità di misura della velocità di trasmissione	Simbolo	Equivalenza
bit per secondo	bps	Unità di misura
Kilobit per secondo	Kbps	1 Kbps = 10^3 bps = 1.000 bps
Megabit per secondo	Mbps	1 Mbps = 10^6 bps = 1.000.000 bps
Gigabit per secondo	Gbps	1 Gbps = 10^9 bps = 1.000.000.000 bps
Terabit per secondo	Tbps	1 Tbps = 10^{12} bps = 1.000.000.000.000 bps

#prendinota

Bandwidth è la velocità teorica raggiungibile mentre **throughput** è la velocità effettivamente misurata che risulta sempre essere minore della bandwidth per vari motivi: tipo di dati che devono essere trasmessi, numero di utenti della rete, topologia della rete, dispositivi di rete, disturbi presenti nel mezzo trasmissivo (il cosiddetto rumore).

La velocità di trasmissione viene espressa attraverso due concetti:

- **throughput**: è la quantità di dati (reale) che sono transitati in un canale in un certo periodo di tempo. Viene espresso come numero di bit trasmessi in un secondo. Il throughput è un'informazione utile per capire se c'è traffico in rete;
- **bandwidth** (larghezza di banda): è la quantità di dati massima (teorica) che può fluire in un canale in un dato periodo di tempo. Viene espresso come la quantità di bit trasmissibili in un secondo. La bandwidth è come il numero di corsie in autostrada: più ce ne sono e più auto possono viaggiare.

Le velocità di bandwidth e throughput variano a seconda del mezzo fisico utilizzato per la trasmissione, dato che il segnale viaggia all'interno del mezzo trasmissivo con una certa frequenza e la velocità di trasmissione è strettamente legata alla frequenza del segnale. Anche la distanza che il segnale riesce a coprire dipende dal mezzo fisico utilizzato per la trasmissione. Nell'Unità 6 affronteremo i diversi mezzi fisici di trasmissione e le loro performance in termini di velocità trasmissiva (bandwidth e throughput) e distanze coperte.

FISSA LE CONOSCENZE

- Che cosa si intende per multiplexing del canale?
- Come funziona la tecnica FDM (*Frequency Division Multiplexing*)?
- Che differenza c'è tra le codifiche Manchester e Differential Manchester?
- Spiega la caratteristica duplex della trasmissione dati (massimo 5 righe).
- Spiega la caratteristica point-to-point e point-to-multipoint della trasmissione dati (massimo 5 righe).
- Che differenza c'è tra bandwidth e throughput?

4 IL CONTROLLO DEGLI ERRORI IN TRASMISSIONE

4.1 Codici rilevatori e correttori

Il segnale inviato attraverso un canale può essere soggetto a rumore elettrico, interferenze e distorsioni che alterano il messaggio e lo rendono incomprensibile al ricevente o, peggio ancora, con un contenuto informativo differente da quello inviato dal mittente. Per fare in modo che il destinatario del messaggio sia in grado di riconoscere quando i dati ricevuti non corrispondono a quelli originali, li possa scartare richiedendone la ritrasmissione oppure possa correggerli, è necessario che il mittente aggiunga appositi codici ai bit da trasmettere, che verranno interpretati dal destinatario.

Il controllo dell'errore si basa su **codici di ridondanza** che aggiungono dei bit all'informazione da trasmettere. L'aggiunta di questi bit ridondanti consente al ricevente di verificare la correttezza dell'intera trasmissione.

Tali codici si suddividono in:

- **codici rilevatori** (error detection): in grado solo di rilevare la presenza o meno di errori nella sequenza di bit ricevuti dal destinatario, ma non la loro posizione; in questo caso il ricevente può chiedere la ritrasmissione del messaggio o segnalare l'errore all'applicazione;
- **codici correttori** (error correction): in grado di rilevare una o più posizioni errate e quindi di correggerle per semplice inversione del bit (un 1 diventa 0 e viceversa) senza che l'applicazione se ne accorga.

Quanti devono essere questi bit ridondanti (aggiunti per il controllo dell'errore) affinché il destinatario si accorga dell'eventuale errore?

Dati m bit di dati e r bit ridondanti, si ottiene un blocco complessivo di n bit ($n = m + r$), detto **codeword**, che corrisponde alla sequenza di bit trasmessa sul canale.

I codici di rilevamento/correzione degli errori si basano sul seguente principio:

i bit di ridondanza si calcolano in modo tale che non vengano utilizzate tutte le possibili codeword.

Quindi delle 2^n possibili codeword, 2^m saranno valide (cioè codeword che si possono trasmettere) e le altre indicheranno la presenza di errori. Meno sono le codeword valide, rispetto all'insieme di tutte le possibili codeword, più è possibile riconoscere e correggere gli errori.

esempio

Prendiamo una sequenza di $n = 2$ bit: questa può assumere $2^n = 2^2 = 4$ differenti configurazioni: 00, 01, 10, 11. Se tutte e 4 queste configurazioni sono usate come dati validi, un errore trasformerà una parola valida in un'altra altrettanto valida, rendendo così impossibile rilevare l'errore.

esempio

Aggiungendo alla sequenza originale un bit di ridondanza si ottengono $2^3 = 8$ configurazioni (codeword) di cui solo 4 valide. Nella **TABELLA 2** sono elencate, per ogni codeword valida, le codeword errate risultanti da un errore singolo.

TABELLA 2 Codeword valide ed errate con due bit di dati e un bit di ridondanza

Stati validi	001	010	100	111
stati di errore	000	000	000	110
stati di errore	011	011	110	011
stati di errore	101	110	101	101

Con l'aggiunta di un bit e la scelta delle codeword ammesse gli stati di errore non possono essere interpretati come stati validi; risulta così semplice la rilevazione dell'errore (singolo). Inoltre ogni codeword errata differisce dalla corrispondente valida per un solo bit, mentre due configurazioni valide differiscono tra loro per 2 bit (per esempio 010 e 111 differiscono nel primo e nell'ultimo bit).

Date due codeword valide si definisce **distanza di Hamming** tra esse il numero di bit di cui differiscono a parità di posizione. Dato un codice a n bit, si definisce **distanza di Hamming del codice** la distanza di Hamming **minima** tra tutte le codeword del codice stesso.

Ciò significa che, se due codeword hanno distanza di Hamming pari a d , saranno necessari d errori di singoli bit per trasformare una codeword valida in un'altra valida, ingannando il ricevente.

La proprietà di un codice di rilevare/correggere gli errori dipende dalla sua distanza di Hamming. Infatti, vale quanto segue:

- **per rilevare k errori** è necessario un codice la cui distanza sia $d = k + 1$ perché è impossibile che k errori di singoli bit trasformino una codeword valida in un'altra codeword valida: ne servono $k + 1$;
- **per correggere k errori** è necessario un codice con una distanza $d = 2k + 1$, perché in esso le codeword valide sono così distanti che, anche se si verificassero k alterazioni di bit, la codeword originale risulterebbe più vicina a quella alterata che a qualunque altra, per cui sarebbe univocamente determinabile.

Nel primo esempio la distanza di Hamming è 1:

$$d = k + 1; \text{ con } d = 1 \text{ si ha } k = 0$$

non è possibile rilevare alcun errore.

Nel secondo esempio la distanza di Hamming è 2:

$$d = k + 1; \text{ con } d = 2 \text{ si ha } k = 1$$

è possibile rilevare un errore singolo, ma non correggerlo, infatti:

$$d = 2k + 1; \text{ con } d = 2 \text{ si ha } k = 0$$

Se vogliamo che un codice sia in grado di rilevare errori (singoli e doppi) e correggere automaticamente errori singoli, occorre che la distanza di Hamming minima sia 3:

$$\begin{aligned} d &= k + 1; \text{ con } d = 3 \text{ si ha } k = 2 \rightarrow \text{rileva fino a errori doppi} \\ d &= 2k + 1; \text{ con } d = 3 \text{ si ha } k = 1 \rightarrow \text{corregge errori singoli} \end{aligned}$$

Poter correggere un errore automaticamente significa evitare la ritrasmissione della stringa: il ricevente individua l'errore e lo corregge da sé.

Naturalmente questo ha un costo in termini di un maggior numero di bit da trasmettere e, quindi, una maggiore occupazione del canale trasmissivo.

Possiamo sintetizzare con le seguenti implicazioni:

distanza di Hamming alta → tanti bit ridondanti → codeword più lunghe → correzione automatica errori

Vediamo le tre principali tecniche di correzione dell'errore in trasmissione mediante codici rilevatori/correttori:

- codici di parità (solo rilevatore errori singoli);
- codici di ridondanza ciclica (solo rilevatore);
- codici di Hamming (rilevatore e correttore).

#prendinota

Per poter correggere automaticamente un errore singolo occorre sapere la posizione del bit sbagliato all'interno della stringa. I codici correttori riescono a fare questo.

4.2 Codici di parità

I codici di parità sono quelli in cui la distanza di Hamming è 2 e sono quindi in grado di rilevare la presenza di un **errore singolo** (senza poterlo correggere perché non ne determinano la posizione) o, più in generale, rilevano l'occorrenza di un numero **dispari** di errori. Alla sequenza di bit da trasmettere si aggiunge un bit di controllo in modo che il numero totale (bit di dati più il bit di controllo) di "1" sia pari (*parità pari*) oppure dispari (*parità dispari*) a seconda del protocollo scelto.

esempio

Sia data la sequenza di bit: 01100010101111

Il numero di bit "1" è 8, quindi pari, allora il bit di parità sarà 0 per parità pari e 1 per parità dispari:

parità pari: 011000101011110

parità dispari: 011000101011111

Il ricevitore provvederà a ricalcolare il bit di parità sulla sequenza di bit ricevuta, escluso il bit di parità aggiunto, e confronterà il bit di parità ottenuto con quello ricevuto: se sono diversi la trasmissione non è avvenuta correttamente, se invece sono uguali è probabile che la sequenza ricevuta sia quella originale; tuttavia, poiché questa tecnica non rileva gli errori doppi, non se ne ha la garanzia assoluta.

Un altro problema che i codici di parità presentano è che non sono in grado di riconoscere quando l'errore è sul bit di parità: questo comporta il rilevamento di un errore quando, invece, la sequenza dei bit dati è stata ricevuta correttamente.

Per questi problemi, l'utilizzo dei codici di parità è limitato ai trasferimenti all'interno del computer (tramite la motherboard). Per esempio i bus SCSI e PCI usano la parità per trovare errori di trasmissione, inoltre molte memorie cache includono tale sistema di controllo e correzione. Dato che nelle cache i dati sono solo una copia di quelli nella RAM, se vi si trova un errore, può essere ricaricata.

4.3 Codici di ridondanza ciclica

Un altro metodo per la rilevazione degli errori è quello dei codici di ridondanza ciclica (**CRC**, Cyclic Redundancy Code). È il codice più utilizzato in rete per la rilevazione degli errori in trasmissione.

Un codice ciclico prevede l'aggiunta di un numero limitato di bit di controllo (generalmente 16 o 32) in coda alla sequenza di bit da trasmettere.

Calcolo CRC da parte del trasmettitore:

1. gli **m** bit della sequenza da trasmettere vengono considerati come coefficienti di un polinomio di grado $m-1$, che chiameremo **M(x)**, e valgono 0 o 1 a seconda del valore del bit corrispondente;
un esempio con $m = 8$ è: $10110011 \rightarrow M(x) = x^7 + x^5 + x^4 + x + 1$
2. viene preso un secondo polinomio, chiamato **polinomio generatore G(x)**, di grado **r** stabilito a livello di standard internazionale e, quindi, noto a priori sia al trasmettitore che al ricevitore, le cui caratteristiche sono:
 - è sempre di grado inferiore al polinomio **M(x)** da trasmettere ($r < m - 1$);
 - ha sempre il coefficiente del termine x^0 uguale a 1;
3. alla sequenza **m** di bit che costituisce **M(x)** si aggiunge in coda (cioè nelle posizioni meno significative) un numero di bit 0 pari al grado **r** del polinomio **G(x)**, ottenendo un blocco di **n = m + r** bit, dove **m** è il numero dei bit dati da trasmettere e **r** è il numero di bit ridondanti di controllo, ottenendo così un nuovo polinomio **D(x)**;
4. il polinomio **D(x)** viene diviso, in modulo 2, per **G(x)** effettuando l'operazione logica XOR (eXclusive OR) (TABELLA 3) tra il resto parziale e la stringa del divisore, cioè **G(x)**;
5. i bit di **resto** della divisione, polinomio **R(x)**, sono i bit di ridondanza (**CRC**) che andranno a sostituire gli **r** bit uguali a 0 che erano stati aggiunti in coda al polinomio **M(x)**, ottenendo così il blocco di bit da trasmettere formato dai bit di dato del messaggio originale più i bit del CRC calcolati tramite la divisione.

TABELLA 3 eXclusive OR

XOR
$0 \oplus 0 = 0$
$0 \oplus 1 = 1$
$1 \oplus 0 = 1$
$1 \oplus 1 = 0$

#prendinota

I polinomi generatori standard più usati sono:

CRC-CCITT

$$G(x) = x^{16} + x^{12} + x^5 + 1$$

CRC-IBM

$$G(x) = x^{16} + x^{15} + x^2 + 1$$

CRC-IEEE 802.3 ETHERNET

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

CRC-ISO

$$G(x) = x^{64} + x^4 + x^3 + x + 1$$

Calcolo CRC da parte del ricevitore: per rilevare la presenza di un errore il ricevitore divide il polinomio, ricavato dalla sequenza di bit ricevuti, per lo stesso **G(x)**, e verifica che il resto sia nullo. Se non lo è, il ricevitore deduce che si è verificato un errore durante la trasmissione.

Le lunghezze polinomiali più comunemente usate sono:

- 9 bit (CRC-8)
- 17 bit (CRC-16)
- 65 bit (CRC-64)
- 13 bit (CRC-12)
- 33 bit (CRC-32)

esempio

Supponiamo di dover trasmettere la sequenza di bit dati: 101010101101 e di usare come polinomio generatore **G(x) = x³ + x + 1**.

$M(x) = x^{11} + x^9 + x^7 + x^5 + x^3 + x^2 + 1$ è il polinomio corrispondente alla sequenza di bit dati

G(x) = x³ + x + 1 è un polinomio di grado 3 che corrisponde alla stringa di bit: 1011

Dovendo procedere con la divisione, si aggiunge in coda alla sequenza di bit dati un numero di 0 pari al grado del polinomio generatore, quindi il dividendo diventerà la stringa: 101010101101000 corrispondente a **D(x)**, mentre il divisore sarà la stringa

1011, corrispondente a $G(x)$. La divisione in modulo 2 serve a calcolare il **resto** della divisione:

$$\begin{array}{r}
 101010101101000 : 1011 \\
 \underline{1011} \\
 0001101 \\
 \underline{1011} \\
 01100 \\
 \underline{1011} \\
 01111 \\
 \underline{1011} \\
 01001 \\
 \underline{1011} \\
 001001 \\
 \underline{1011} \\
 001000 \\
 \underline{1011} \\
 00110 \\
 \Rightarrow CRC = 110
 \end{array}$$

La sequenza di bit che viene trasmessa sul canale diventa: 101010101101110.

esercizio

→ PROBLEMA

Data la seguente sequenza di bit da trasmettere: 10100101101011, calcolare il CRC sapendo che il polinomio generatore è: $G(X) = x^4 + x + 1$

→ SVOLGIMENTO

$$\begin{array}{r}
 101001011010110000 : 10011 \\
 \underline{10011} \\
 0011110 \\
 \underline{10011} \\
 011011 \\
 \underline{10011} \\
 010001 \\
 \underline{10011} \\
 00010010 \\
 \underline{10011} \\
 000011100 \\
 \underline{10011} \\
 011110 \\
 \underline{10011} \\
 011010 \\
 \underline{10011} \\
 01001 \\
 \Rightarrow CRC = 1001
 \end{array}$$

La sequenza di bit che sarà trasmessa sul canale è: 101001011010111001

4.4 Codici di Hamming

I codici di Hamming sono codici aventi distanza di Hamming (d) minima uguale a 3. Permettono dunque la correzione automatica degli errori singoli (k = 1):

$$d = 2k + 1; \text{ con } d = 3 \text{ si ha } k = 1 \text{ corregge errori singoli}$$

La tecnica utilizzata per l'individuazione della posizione del bit errato si basa sull'aggiunta, in **posizioni ben precise**, di alcuni bit ridondanti di controllo, chiamati anche **bit di parità o di check**.

Supponendo, come al solito, che una codeword lunga n bit sia costituita da una parte dati (lunga m bit) e da una parte ridondante di controllo (lunga r bit)

$$n = m + r$$

allora la tecnica di Hamming impone la seguente regola:

$$2^r - 1 \geq m + r$$

Se prendiamo il caso estremo della disequazione, cioè quando vale l'**uguale**, abbiamo:

r = 3 → m = 4 → n = 7 (per trasmettere 4 bit di dati servono 3 bit di check: codeword lunga 7 bit);

r = 4 → m = 11 → n = 15 (per trasmettere 11 bit di dati servono 4 bit di check: codeword lunga 15 bit);

r = 5 → m = 26 → n = 31 (per trasmettere 26 bit di dati servono 5 bit di check: codeword lunga 31 bit);

e così via.

Come è stato detto, in questa tecnica è fondamentale la **posizione** in cui vengono collocati i bit di check.

Vale la regola secondo cui i bit di check si aggiungono nelle posizioni corrispondenti alle **potenze del due, partendo da sinistra**, che chiameremo:

$$b_1 \quad b_2 \quad b_4 \quad b_8 \quad b_{16} \quad \dots$$

Ogni bit di check in più (quindi una potenza del 2 in più) permette di correggere errori singoli fino al bit di check successivo (quindi la prossima potenza del 2). Dunque, con **b₁** e **b₂** controllo fino a **b₃**; aggiungendo **b₄** controllo fino a **b₇**; aggiungendo **b₈** controllo fino a **b₁₅**;

Codeword			
Posizione	b ₁	b ₂	b ₃

Codeword							
Posizione	b ₁	b ₂	b ₃	b ₄	b ₅	b ₆	b ₇

Codeword															
Posizione	b ₁	b ₂	b ₃	b ₄	b ₅	b ₆	b ₇	b ₈	b ₉	b ₁₀	b ₁₁	b ₁₂	b ₁₃	b ₁₄	b ₁₅

Se si aggiungesse il bit di check b_{16} , si potrebbe controllare l'errore singolo fino al bit di dati b_{31} .

Si dovrà quindi, prima di tutto, inserire i bit di dati, lasciando libere le posizioni riservate ai bit di check.

Supponiamo di voler trasmettere i seguenti 11 bit di dati:

<div>1 1 1 0 0 1 1 1 0 1 0</div> <div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>															
Codeword			1		1	1	0		0	1	1	1	0	1	0
Posizione	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}

Servono dunque 4 bit di controllo (b_1 , b_2 , b_4 , b_8) per un totale di codeword di 15 bit.

Per il calcolo dei 4 bit di check ci si avvale della seguente **TABELLA 4**:

Bit:	Controlla le posizioni:
bit 1	3, 5, 7, 9, 11, 13, 15
bit 2	3, 6, 7, 10, 11, 14, 15
bit 4	5, 6, 7, 12, 13, 14, 15
bit 8	9, 10, 11, 12, 13, 14, 15

TABELLA 4 Schema per il calcolo dei bit di check

La prima riga della tabella dice che il bit di check b_1 contribuisce a controllare i bit trasmessi nelle posizioni 3, 5, 7, 9, 11, 13, 15. La seconda riga della tabella dice che il bit di check b_2 contribuisce a controllare i bit trasmessi nelle posizioni 3, 6, 7, 10, 11, 14, 15. E così via.

Ad esempio per il bit di informazione in posizione 6 servono i bit di check b_2 e b_4 poiché il 6 compare nelle righe di b_2 e b_4 .

Perché la Tabella 4 impone l'utilizzo di certi bit di check e non di altri, per ogni bit di informazione da controllare?

Dalla tabella si vede che ogni bit di check controlla una serie di bit di cui è indicata la posizione.

Un certo bit di check controlla proprio quei bit specificati e non altri per una **questione di pesi**.

Prendiamo la solita posizione 13 che in binario vale 1101 e consideriamo che è data dalla somma dei pesi:

$$1 + 4 + 8 = 13$$

Andranno dunque controllati i pesi nelle posizioni b_1 , b_4 , b_8 per avere informazioni sulla posizione 13.

La Tabella 4 infatti inserisce il 13 nel calcolo di b_1 , b_4 , b_8 e non di b_2 .

Poiché non possono esistere due numeri diversi con stessi identici pesi, i bit di peso discordanti individuati dallo XOR individuano **univocamente** la posizione errata.

La parità può essere scelta sia pari che dispari. Supponiamo **pari** e procediamo al calcolo grazie alla Tabella 4:

bit 1 $\rightarrow 1, 1, 0, 0, 1, 0, 0 \rightarrow b_1 = 1$

bit 2 $\rightarrow 1, 1, 0, 1, 1, 1, 0 \rightarrow b_2 = 1$

bit 4 $\rightarrow 1, 1, 0, 1, 0, 1, 0 \rightarrow b_4 = 0$

bit 8 $\rightarrow 0, 1, 1, 1, 0, 1, 0 \rightarrow b_8 = 0$

Inseriamo i bit di controllo calcolati, ottenendo la codeword finale da trasmettere:

Codeword	1	1	1	0	1	1	0	0	0	1	1	1	0	1	0
Posizione	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}

Il destinatario ripete l'intero identico procedimento, cioè ricalcola in maniera autonoma i bit di check, utilizzando la stessa Tabella 4 e la stessa parità pari.

Ottenuti i nuovi bit di check, li confronta con quelli ricevuti nella codeword attraverso l'operazione logica XOR.

Se il risultato dello XOR tra i bit di check è costituito da soli 0, allora la stringa ricevuta non presenta errori; se invece il risultato dello XOR non contiene solo zeri, allora c'è un errore singolo correggibile. La posizione del bit errato è data dal valore risultante dallo XOR convertito in decimale.

Supponiamo, per esempio, la presenza di un **errore di trasmissione in posizione 13** e quindi di aver ricevuto:

Codeword	1	1	1	0	1	1	0	0	0	1	1	1	1	1	0
Posizione	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}

Il ricalcolo dei bit di check produrrebbe:

bit 1 $\rightarrow 1, 1, 0, 0, 1, 1, 0 \rightarrow b_1 = 0$

bit 2 $\rightarrow 1, 1, 0, 1, 1, 1, 0 \rightarrow b_2 = 1$

bit 4 $\rightarrow 1, 1, 0, 1, 1, 1, 0 \rightarrow b_4 = 1$

bit 8 $\rightarrow 0, 1, 1, 1, 1, 1, 0 \rightarrow b_8 = 1$

e quindi lo XOR in ordine di peso:

b_8	b_4	b_2	b_1	
1	1	1	0	bit di check ricalcolati dal destinatario
0	0	1	1	bit di check ricevuti nella codeword dal destinatario
<hr/>				
1	1	0	1	

Il numero binario ottenuto (**1101**), convertito in decimale, corrisponde proprio alla posizione (**13**) in cui c'è l'errore.

Il destinatario può quindi correggere automaticamente il bit errato, senza chiedere la ritrasmissione della codeword.