

RIPASSIAMO LE BASI DELL'ELETTROTECNICA

L'articolo offre un quadro completo ed aggiornato delle leggi che regolano l'elettrotecnica, leggi che anche un tecnico pratico, quale può essere un installatore, non può esimersi dal conoscere per svolgere razionalmente il proprio lavoro.

Parte diciasette: Elettronica Digitale

A cura della Redazione

17.1 - Generalità

L'elettronica digitale si occupa del trattamento di segnali elettrici per effettuare, tramite essi, operazioni di carattere logico matematico. I segnali elettrici utilizzati nei sistemi digitali possono assumere solo due livelli di tensione: alto, o livello 1 e basso, o livello 0.

Esaminando il comportamento dei transistori si è visto che essi possono lavorare in due modi diversi: linearmente o in commutazione. Nel primo caso vengono utilizzati tipicamente come amplificatori, nel secondo vengono invece impiegati come interruttori elettronici potendo così simulare le condizioni di interruttore aperto o chiuso.

Questi due diversi modi di trattare i segnali e quindi le informazioni che essi contengono, consentono di suddividere l'elettronica in due parti:

- *elettronica analogica*, che comprende tutti i sistemi che elaborano segnali variabili con continuità in modo che essi presentino un andamento analogo a quello dell'informazione;
- *elettronica numerica o digitale* (dall'inglese digit = cifra), che comprende i sistemi nei quali i segnali possono assumere due soli livelli di tensione: alto (o livello 1) e basso (o livello 0) ⁽¹⁾ per cui l'informazione può essere trasmessa o elaborata mediante segnali costituiti da una serie di impulsi (Figura 17.1) la cui ampiezza dipende dal valore attribuito al livello di tensione 1 ed a quello di tensione 0.

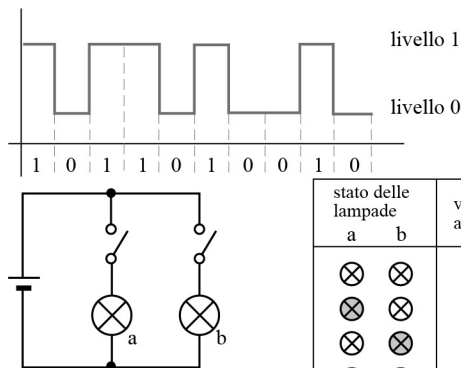


Figura 17.1 - Rappresentazione grafica di un segnale digitale.

stato delle lampade		valore numerico associato
a	b	
⊗	⊗	0
⊗	⊙	1
⊙	⊙	2
⊙	⊗	3

Figura 17.2 - Esempio di rappresentazione di valori numerici.

Nell'elettronica digitale l'unità di informazione elementare viene indicata con il termine *bit*, contrazione delle parole inglesi *binary digit* (cifra binaria) e può assumere quindi i due stati indicati convenzionalmente con i numeri 0 e 1.

Per trasmettere informazioni più complesse è necessario evidentemente ricorrere a più stadi binari. Si consideri, infatti, il circuito rappresentato in figura 17.2 costituito da due lampade *a* e *b* comandate indipendentemente; se si associa un valore numerico rispettivamente a ciascuno dei seguenti stati: lampade spente, lampada *a* accesa, lampada *b* accesa, entrambe le lampade accese, è possibile rappresentare i numeri da 0 a 3.

Aumentando a quattro il numero delle lampade le combinazioni possibili risultano $2^4 = 16$, per cui sono rappresentabili i valori da 0 a 15 e così via.

Gli esempi riportati non devono indurre a ritenere che i circuiti digitali siano utilizzati esclusivamente nei sistemi di calcolo, in realtà con questa tecnica possono essere trasmesse ed elaborate informazioni costituite da grandezze variabili in modo numerico, da caratteri alfanumerici (lettere dell'alfabeto, cifre, segni di punteggiatura e speciali) ed anche da grandezze di tipo analogico.

17.2 - Caratteristiche della tecnica digitale

Con la tecnica digitale più che il valore assoluto assunto dai segnali elettrici, è importante che la differenza tra i livelli alto (1) e basso (0) sia sufficientemente elevata.

Con la tecnica digitale è possibile memorizzare dati e conservare le informazioni per lungo tempo.

Ciò che caratterizza principalmente i sistemi digitali è che, ai fini di un loro corretto funzionamento, non risulta rilevante l'esatto valore di tensione assunto dal segnale, ma è solo necessario che la differenza tra il valore relativo al livello 1 e quello corrispondente al livello 0 sia sufficientemente elevata per poter distinguere senza dubbio i due stati.

Di conseguenza nelle apparecchiature elettroniche possono essere utilizzati circuiti di commutazione che risultano molto più semplici dei circuiti analogici e per i quali possono essere impiegati componenti con prestazioni che non devono soddisfare tolleranze particolarmente ristrette. Malgrado ciò i sistemi digitali risultano più precisi dei corrispondenti analogici in quanto le informazioni sono costituite in pratica da numeri che possono essere elaborati in modo molto accurato.

Uno degli aspetti più interessanti della tecnica digitale è rappresentato dalla possibilità di memorizzare i dati e conservare quindi le informazioni per lungo tempo.

Ciò può essere dedotto facilmente osservando nuovamente la figura 17.2 se i relativi interruttori risultano in una data posizione, le lampade indicano un valore numerico che permane sino a che gli interruttori stessi non vengono nuovamente manovrati.

Un ulteriore vantaggio dei sistemi digitali deriva dal fatto che essi si basano, come si vedrà nei prossimi paragrafi, su un numero limitato di unità strutturali di base (porte logiche OR, AND, NOT, flip-flop ecc.) le quali, risultando perfettamente compatibili tra loro, possono essere connesse senza necessità di ulteriori componenti.

I pregi dei sistemi digitali si sono ulteriormente accentuati con l'avvento dei circuiti integrati che hanno consentito una varietà di applicazioni di gran lunga superiore a quella dei circuiti analogici.

17.3 - Sistemi di numerazione

L'insieme delle convenzioni utilizzate per rappresentare i numeri si definisce sistema di numerazione.

I sistemi di numerazione possono essere di vario tipo, i più utilizzati sono i sistemi decimale, binario e ottale.

Un sistema di rappresentazione dei dati che si avvalga di elementi o segnali elettrici deve basarsi su un criterio di codifica capace di esprimere numeri e lettere secondo i due stati di funzionamento 0 e 1. Su questo principio operano i sistemi di numerazione, cioè quelli che permettono la rappresentazione di soli numeri e i sistemi di codifica per la rappresentazione di numeri e lettere.

Sistema decimale

Il sistema di numerazione decimale esprime qualsiasi numero mediante dieci cifre (0, 1, 2, ..., 9) ciascuna delle quale rappresenta una potenza detta base dieci in funzione della posizione che occupa nell'allineamento.

Il sistema di numerazione decimale esprime qualsiasi numero mediante dieci simboli (0, 1, 2, ..., 9) detti cifre.

La notazione decimale è di tipo posizionale, ossia associa a ciascuna cifra del numero da rappresentare una potenza della base 10 che dipende strettamente dalla sua posizione nell'allineamento. Così, ad esempio, il numero 361 equivale a:

$$3 \cdot 10^2 + 6 \cdot 10^1 + 1 \cdot 10^0$$

Con la struttura del sistema decimale ad ogni cifra viene assegnato automaticamente un dato peso chiamato unità, decina, centinaia ecc.; nei numeri frazionari ad ogni cifra a destra della virgola viene assegnata una potenza negativa il cui esponente è ancora funzione della posizione occupata da ciascuna cifra.

Sistema binario

Il sistema di numerazione binario esprime qualsiasi numero mediante due cifre 0 e 1, ordinate secondo potenze decrescenti di 2.

Il sistema di numerazione binario esprime i numeri mediante due simboli 0 e 1 chiamati bit, ordinati secondo potenze decrescenti di 2.

I simboli 0 e 1 hanno lo stesso significato che assumono nel sistema decimale, è però diverso il peso attribuito alla posizione assunta da ciascuna cifra dato che la base, come detto, è due. Ad esempio, il numero 1 101 assume, nel sistema binario, il seguente significato:

$$1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

ed equivale, nel sistema decimale, a 13.

È evidente la differenza di lunghezza del messaggio rispetto a quella del sistema decimale (13, in decimale, sarà: $1 \cdot 10^1 + 3 \cdot 10^0$), ma nel sistema binario i coefficienti hanno solo due contro i dieci del primo.

Sistema ottale

Il sistema di numerazione ottale esprime qualsiasi numero mediante otto cifre (0, 1, 2, ..., 7) ordinate secondo potenze decrescenti di 8.

Il sistema di numerazione ottale è a base 8 e quindi utilizza 8 cifre: 0, 1, 2, 3, 4, 5, 6 e 7.

Scrivendo, ad esempio, 164 nel sistema a base 8, s'intende:

$$1 \cdot 8^2 + 6 \cdot 8^1 + 4 \cdot 8^0$$

pari a 116 nella notazione decimale.

L'impiego del sistema ottale risulta utile in quanto chi utilizza gli elaboratori numerici trova più comodo ricordare e trascrivere una breve sequenza di cifre a base 8 che non un lungo gruppo di cifre binarie. Ad esempio, una sequenza di 12 cifre binarie viene considerata come una successione di 4 blocchi di 3 cifre binarie cui corrisponde un numero di quattro cifre in base 8:

binario	111	101	011	110
ottale	7	5	3	6

Nella tabella 17.1 sono posti a confronto i valori dei primi 16 numeri del sistema decimale espressi in binario e ottale.

Tabella 17.1 - Raffronto tra valori numerici espressi nei sistemi decimale, binario e ottale.

Decimale	Binario	Ottale	
0	0	0	000
1	1	1	001
2	10	2	010
3	11	3	011
4	100	4	100
5	101	5	101
6	110	6	110
7	111	7	111
8	1000	10	001 000
9	1001	11	001 001
10	1010	12	001 010

17.4 - Codificazione

Si definisce sistema di codifica l'insieme delle convenzioni utilizzate per rappresentare numeri e lettere.

La codificazione è l'operazione mediante la quale si fa corrispondere a numeri e a lettere una serie di bit.

La trasformazione di un valore espresso in forma decimale in uno equivalente espresso in forma digitale viene attuata tramite una conversione detta codifica, con la quale si fa corrispondere al numero decimale una serie di bit.

Questo insieme ordinato di bit viene definito parola e il numero di bit costituisce la lunghezza della parola stessa.

Uno dei sistemi di codifica fra i più usati e i più semplici è costituito dal codice NBCD (Natural Binary-Code-Decimal) mediante il quale si fa corrispondere a ciascuna delle nove cifre del sistema decimale una combinazione di 4 bit.

Nessuno impedisce di stabilire altre corrispondenze tra gruppi di 4 bit e cifre decimali. Ciò che in pratica si cerca di fare è di limitare (l'introduzione di errori casuali): più probabili quando il passaggio da una cifra alla successiva comporta il cambiamento di più di un bit della quartina. Ad esempio, nel codice NBCD il passaggio, da 7 a 8 comporta il cambiamento di ben quattro bits. Un codice che non presenta questo inconveniente è il GRAY come si può rilevare dalla tabella 17.2.

Codici alfanumerici

Un codice è detto di tipo alfanumerico quando è in grado di rappresentare, oltre ai numeri, anche lettere, segni di punteggiatura, comandi vari, ecc.

Qualora si debba rappresentare in codice una informazione alfanumerica, cioè contenente cifre, lettere dell'alfabeto o simboli, è molto utilizzato il codice ASCII (American Standard Code for Information Interchange) che utilizza 8 bit divisi in un gruppo da quattro e uno da tre più un bit per il controllo di parità, per verificare cioè resistenza di eventuali errori di codifica nei sistemi che operano con codici binari.

Tabella 17.2 – Rappresentazione delle prime 11 cifre decimali nei vari codici.

Valore decimale	Codice NBCD 8-4-2-1	Codice Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	0001	1111

17.5 - Circuiti logici

Le porte logiche (o circuiti logici) sono circuiti elettronici di commutazione utilizzati l'elaborazione di segnali digitali. In particolare forniscono un segnale in uscita in forma binaria che dipende dallo stato (0 o 1) dei segnali presenti all'ingresso.

Le tabelle della verità rappresentano, in forma tabulare, l'insieme delle combinazioni tra segnali d'ingresso e quello di uscita delle porte logiche.

L'elaborazione dei segnali digitali viene realizzata mediante circuiti di commutazione, detti circuiti logici o porte logiche (logic gates), che forniscono in uscita un segnale binario che dipende dallo stato (0 o 1) dei segnali presenti nello stesso istante ai loro ingressi.

Le porte logiche costituiscono i circuiti base per la realizzazione di tutte le apparecchiature digitali. Dal loro impiego combinato hanno origine, infatti, i circuiti che svolgono le funzioni più complesse.

I circuiti logici sono muniti di uno o più ingressi e di una uscita. Gli ingressi sono generalmente contraddistinti con lettere, mentre l'uscita può essere indicata con una lettera o come risultato di una operazione logica. In questo secondo caso risulta rappresentata da una relazione matematica corrispondente alla funzione svolta dal circuito.

Le varie combinazioni dei segnali d'ingresso ed il corrispondente segnale di uscita possono essere espressi in forma tabulare nelle cosiddette tabelle della verità, mediante le quali risulta particolarmente facile rappresentare in modo conciso l'effettivo comportamento del corrispondente circuito digitale.

I circuiti logici possono avere logica positiva o negativa a seconda che allo stato logico 1 del segnale corrisponda una tensione di livello maggiore (più positivo) rispetto ai livelli dello stato 0 o minore (più negativo) (Figura 17.3).

Nella trattazione che segue si farà riferimento solo a circuiti logici funzionanti con logica positiva ed inoltre gli esempi circuitali riportati riguarderanno, ai fini didattici, schemi di principio.

I circuiti logici sono realizzati mediante circuiti integrati, tuttavia per evidenziare loro funzione si ricorrerà nel seguito a rappresentazioni di tipo elettromeccanico.

Il circuito *OR* (espressione inglese che corrisponde nella nostra lingua alla congiunzione *o*) è un circuito logico per il quale il segnale di uscita è a livello 1 quando uno o più ingressi sono a livello 1.

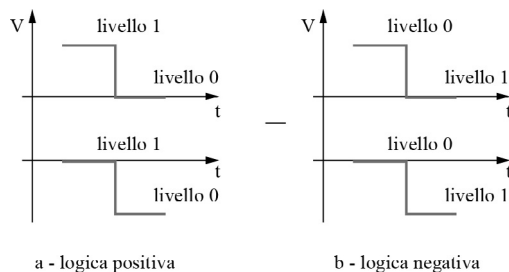


Figura 17.3 - Rappresentazione convenzionale delle logiche positiva e negativa.

La funzione circuitalre *OR* corrisponde all'azione svolta da due o più contatti in parallelo fra loro: è sufficiente che uno solo di questi sia chiuso (ingresso a livello logico 1) perché la lampada risulti

accesa (uscita a livello 1).

Nella tabella 17.3 è schematizzata la funzione *OR* realizzata mediante contatti meccanici, ma tale funzione può essere realizzata con un circuito a diodi o transistori.

Il circuito *AND* (espressione corrispondente, nella lingua italiana, alla congiunzione e) è un circuito logico la cui uscita è a livello 1 solo se tutti gli ingressi si trovano contemporaneamente a livello 1.

Tale funzione corrisponde all'azione svolta da due o più contatti in serie: l'accensione della lampada avviene solo quando tutti i contatti sono chiusi (Tabella 17.3).

Il circuito *NOT* (negazione) è un circuito che fornisce in uscita un segnale di inverso a quello presente all'ingresso (se il segnale d'ingresso è a livello 0 il segnale di uscita risulta a livello 1 e viceversa). Per tale ragione questo circuito è talvolta indicato con il termine invertitore (o inverter).

Il circuito elettromeccanico che realizza questo tipo di funzione logica è rappresentato da un contatto normalmente aperto il quale in chiusura eccita la bobina di un relè che comanda l'apertura di un contatto normalmente chiuso (Tabella 17.3).

Il circuito *NOR* (contrazione delle sigle *NOT* e *OR*) può essere considerato come la combinazione delle funzioni *OR* e *NOT*, in quanto può essere ottenuta invertendo mediante un circuito *NOT* l'uscita del circuito *OR*. In tal modo è sufficiente che un solo ingresso sia a livello 1 perché l'uscita vada a 0 e viceversa allorché tutti gli ingressi sono a livello 0 e l'uscita è a livello 1.

Nella versione elettromeccanica la funzione *NOR* può essere ottenuta con il circuito indicato nella tabella 17.3: solo quando entrambi i contatti sono aperti (livello 0) la lampada risulta accesa (livello 1).

Il circuito *NAND* (contrazione di *NOT* e *AND*) corrisponde alla combinazione di una porta *AND* ed una *NOT* e quindi in pratica ad una porta *AND* la cui uscita però risulta invertita. L'uscita risulta quindi a livello 0 solo quando tutti gli ingressi sono a livello 1; viceversa, in tutti gli altri casi, è a livello 1 (Tabella 17.3).

L'*OR* esclusivo (exclusive *OR*, abbreviato *EX-OR*) è un circuito che fornisce in uscita un segnale a livello 1 quando uno solo dei suoi ingressi è a livello 1.

Il funzionamento del circuito elettromeccanico corrispondente (Tabella 17.3) è facile da comprendere: solo quando il contatto *A* è nella posizione 1 e quello *B* è nella posizione 0 (o viceversa) la bobina del relè è eccitata e quindi chiude il contatto *U* determinando l'accensione della lampada (ossia $U=1$).

17.6 - Codificatori (encoder)

I codificatori (encoder) sono circuiti combinatori capaci di convertire un certo numero di variabili d'ingresso in uno speciale codice di uscita. Vengono realizzati con porte logiche opportunamente associate.

Si è già accennato al fatto che i segnali digitali devono essere codificati in modo che ad ogni combinazione di bit corrisponda in modo univoco un'informazione elementare. Da ciò deriva la necessità di utilizzare circuiti in grado di convertire l'informazione, espressa in forma alfanumerica, in un codice binario.

Tabella 17.3 – Funzioni logiche elementari.

Funzione	Simbolo circuitale	Circuito equivalente	Tabella della verità															
OR			<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>U</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	U	0	0	0	1	0	1	0	1	1	1	1	1
			A	B	U													
0	0	0																
1	0	1																
0	1	1																
1	1	1																
AND			<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>U</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	U	0	0	0	1	0	0	0	1	0	1	1	1
			A	B	U													
0	0	0																
1	0	0																
0	1	0																
1	1	1																
NOT			<table border="1"> <thead> <tr> <th>A</th> <th>U</th> </tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	A	U	0	1	1	0									
			A	U														
0	1																	
1	0																	
NOR			<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>U</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	U	0	0	1	1	0	0	0	1	0	1	1	0
			A	B	U													
0	0	1																
1	0	0																
0	1	0																
1	1	0																
NAND			<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>U</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	U	0	0	1	1	0	1	0	1	1	1	1	0
			A	B	U													
0	0	1																
1	0	1																
0	1	1																
1	1	0																
EX-OR			<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>U</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	U	0	0	0	1	0	1	0	1	1	1	1	0
			A	B	U													
0	0	0																
1	0	1																
0	1	1																
1	1	0																

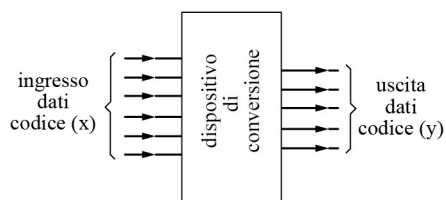


Figura 17.4 - Schematizzazione di un dispositivo per la conversione di codici.

I circuiti che svolgono la funzione di conversione sopra indicata possono essere schematicamente rappresentati, come indicato in figura 17.4, mediante blocchi con un numero di ingressi ed uscite che dipende dal tipo di segnali in ingresso e in uscita.

Il circuito che consente di convertire i dati espressi, ad esempio in forma numerica, in segnali digitali codificati viene detto codificatore (encoder). Questo circuito presenta tanti ingressi quanti sono gli stati che la variabile da codificare può assumere e tante uscite quanti sono i bit necessari per esprimere univocamente ciascuno stato della variabile in ingresso.

Si supponga, a titolo di esempio, di voler codificare i valori corrispondenti ai soli stati numerici di una calcolatrice. Gli ingressi dovranno necessariamente essere $N=10$ a ciascuno dei quali corrisponde un numero da 0 a 9, mentre il numero n delle uscite dipenderà dal numero di bit minimo necessario per esprimere ciascuna delle 10 cifre (ossia $n = 4$).

Premendo un tasto, il circuito codificatore deve determinare il livello logico di ciascuna uscita in modo tale che la combinazione degli stati, secondo il codice previsto, individui univocamente il valore corrispondente alla cifra decimale relativa al tasto premuto. Se, ad esempio, si utilizza il codice NBCD, la combinazione degli stati agli ingressi e quelli all'uscita deve essere quella indicata nella tabella di figura 17.5a.

Dalla tabella si possono rilevare gli stati che le uscite possono assumere in relazione ai vari livelli che tutti gli ingressi possono presentare contemporaneamente.

Un circuito logico in grado di realizzare quanto sopra indicato può essere ottenuto con l'impiego delle porte logiche OR assumendo, di conseguenza, la configurazione rappresentata in figura 17.5b.

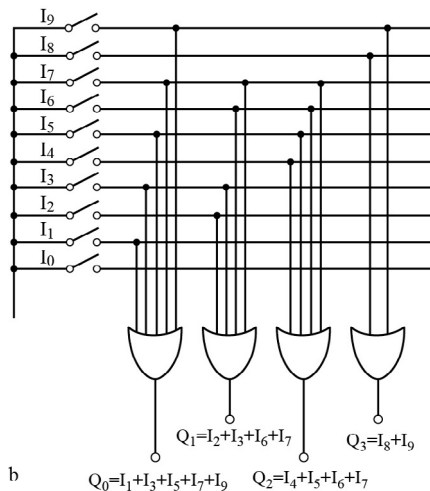
Ingressi										Uscite			
I_9	I_8	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

a

Figura 17.5 – Codificatore a 10 ingressi e uscita a quattro bit con porte OR:

a - tabella della verità;

b - schema logico.



b

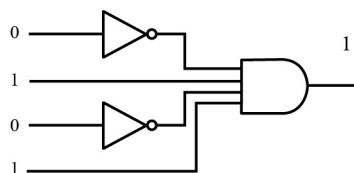


Figura 17.6 - Schematizzazione di un circuito per decodifica di una parola binaria.

17.7 - Decodificatori (decoder)

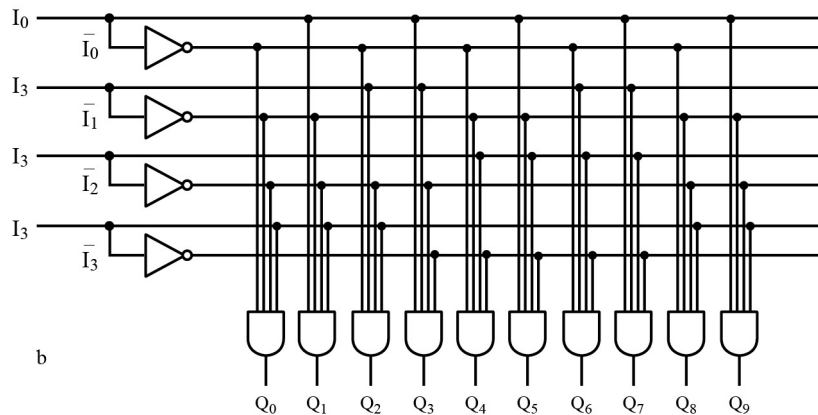
I decodificatori (decoder) sono circuiti combinatori capaci di convertire il codice dei segnali di ingresso in un codice diverso per i segnali di uscita.

I decodificatori vengono realizzati con porte logiche opportunamente associate fra loro.

Un decodificatore (decoder) è costituito, in genere, da un circuito avente tanti ingressi quanti sono i bit della parola in ingresso e da tante uscite quanto sono i possibili stati che la variabile in uscita può assumere. Si consideri uno dei più comuni decodificatori, ossia quello che converte i dati espressi in codice NBCD in valori decimali; in questo caso il numero degli ingressi risulta $n = 4$ mentre le uscite sono 10 corrispondenti alle cifre $0 \div 9$.

Ingressi				Uscite									
I_3	I_2	I_1	I_0	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9
x	x	x	x	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
0	1	1	1	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	0	0	0	0

a

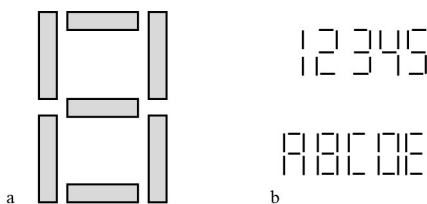


b

Figura 17.7 - Decodificatore NBCD/decimale realizzato con porte NOT e AND:

a - tabella della verità;

b - schema logico.



a

b

Figura 17.8 - Visualizzatore a 7 segmenti:

a - esempi di numeri;

b - esempi di lettere.

La decodifica può essere realizzata mediante un numero di porte AND corrispondente al numero delle uscite: quando ad una data porta perviene la relativa combinazione di bit lo stato dell'uscita passa da 0 a 1.

Si supponga, ad esempio, di voler decodificare la parola 0101 corrispondente al valore decimale

5: è evidente che se ciascun bit viene portato all'ingresso del circuito indicato in figura 17.6, l'uscita della porta assume lo stato 1 mentre per tutte le altre combinazioni di bit l'uscita risulterà sempre a livello logico 0.

Collegando opportunamente 10 porte si può quindi realizzare un decodificatore NBCD/decimale. Lo schema delle connessione può essere desunto dalla tabella della verità (Figura 17.7a) dalla quale si può rilevare come ogni uscita sia funzione dello stato degli ingressi.

Lo stato complementare degli ingressi è ottenuto direttamente all'interno del circuito utilizzando le porte NOT (Figura 17.7b).

Visualizzatori

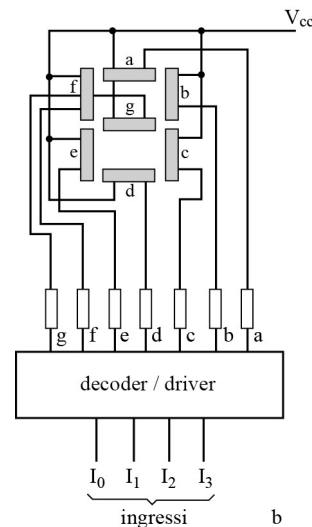
I visualizzatori sono decodificatori che rendono visibili, in uscita, lettere dell'alfabeto, numeri ed altri simboli.

Nelle apparecchiature elettroniche si fa uso principalmente, per rendere visibili dati e informazioni, di indicatori optoelettronici del tipo a 7 segmenti. Questo indicatore è costituito da 7 trattini che possono essere illuminati tutti contemporaneamente o

Ingressi				Uscite							Cifra visualizzata
I ₃	I ₂	I ₁	I ₀	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	0	1	1	9

a

Figura 17.9 - Visualizzatore a LED:
a - tabella delta verità;
b - schema eli principio.



b

solo alcuni per volta in modo da formare le varie cifre o lettere (Figura 17.8).

In relazione alla tecnica di illuminazione dei trattini si possono distinguere visualizzatori a LED (light Emitting Diode) oppure visualizzatori a cristalli liquidi.

Nella figura 17.9 è illustrato un decoder che comanda un display a LED e la relativa tabella della verità dalla quale si può rilevare come, per ciascuna combinazione dei segnali di ingresso, vengano portate a livello 1 le uscite corrispondenti ai LED da pilotare affinché risulti visualizzata la cifra corrispondente.

I visualizzatori a cristalli liquidi si basano sull'impiego di speciali cristalli posti tra due elettrodi. In assenza di tensione fra i due elettrodi i cristalli sono trasparenti mentre alimentando i due elettrodi con tensione alternata essi diventano scuri e quindi visibili. Poiché il visualizzatore è composto, come il tipo a LED, da 7 trattini si riesce in tal modo a rappresentare qualsiasi numero o lettera.

17.8 - Selettori (Multiplexer)

*I selettori sono blocchi logici a più ingressi e ad una sola via di uscita.
I selettori vengono realizzati con porte logiche opportunamente associate tra loro.*

I multiplexer sono circuiti che presentano un certo numero di ingressi dati (data inputs), alcuni ingressi di selezione (select inputs), detti anche ingressi di indirizzo e una uscita. Essi consentono di selezionare uno alla volta i data inputs e quindi di trasferire il dato sulla linea di uscita.

Ipotizzando una versione elettromeccanica, i multiplexer possono essere assimilati ad una serie di relè: a seconda del relè alimentato viene connessa una linea di ingresso con quella di uscita (Figura 17.10)

La scelta della linea di ingresso viene attuata, nei circuiti digitali, inviando agli ingressi di selezione un segnale in codice binario: ciascuna combinazione di bit del segnale attiva un relè che connette di conseguenza un determinato ingresso con l'uscita la quale assume il relativo livello logico.

Poiché la selezione dell'ingresso dati deve avvenire mediante parole composte da

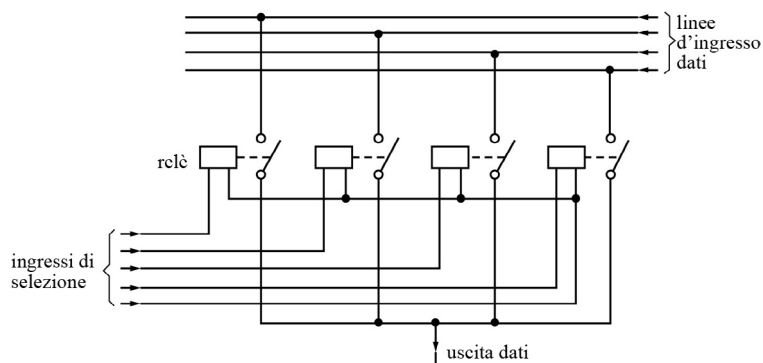


Figura 17.10 - Schematizzazione elettromeccanica di un multiplexer.

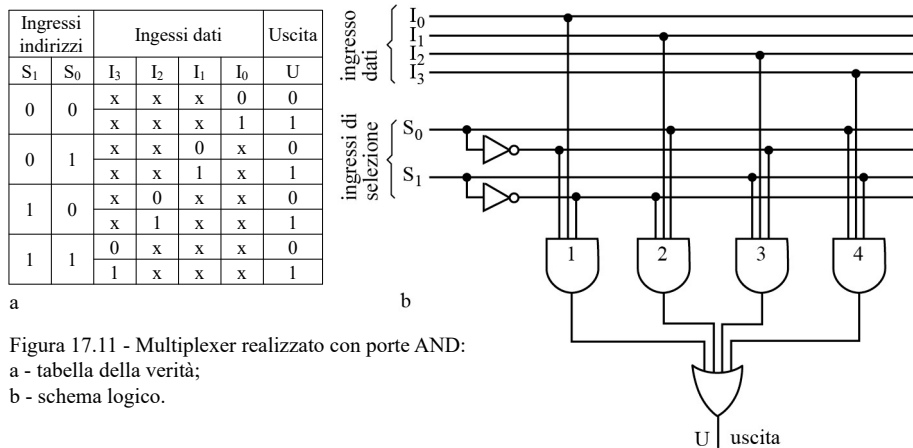


Figura 17.11 - Multiplexer realizzato con porte AND:
a - tabella della verità;
b - schema logico.

un numero di bit tale da consentire di ottenere le combinazioni sufficienti ad individuare univocamente tutti gli ingressi dei dati, se questi ultimi sono N numero di ingressi di selezione n dovrà essere tale da soddisfare la relazione $2^n = N$.

Nella figura 17.11 è indicato un circuito multiplexer a 4 ingressi dati e la relativa tabella della verità. Da quest'ultima si rileva che con segnali agli ingressi di indirizzo a livello $S_0 = 0$ e $S_1 = 0$ risulta abilitata la porta AND 1 per cui il dato presente sull'ingresso I_0 viene trasferito all'uscita sia esso 0 o 1. Analogamente con segnale di selezione di stato $S_0 = 1$ e $S_1 = 0$ viene abilitata la porta AND 2 e così via.

Si osservi che abilitando una porta alla volta solo lo stato dell'ingresso abilitato determina il livello logico dell'uscita, mentre gli stati degli altri ingressi sono irrilevanti.

17.9 - Distributori (Demultiplexer)

I distributori sono blocchi logici a un ingresso e a più uscite. Vengono realizzati con porte logiche opportunamente associate tra loro.

Il demultiplexer svolge la funzione opposta a quella del multiplexer. Esso, infatti, consente di selezionare una tra tutte le sue uscite e di trasferirvi i dati presenti all'unico ingresso. Anche in questo dispositivo la selezione viene effettuata mediante una parola di n bit applicata agli ingressi di indirizzo.

In relazione al numero N delle uscite sono necessari n ingressi d'indirizzo in modo che sia verificata la relazione $2^n = N$.

Nella figura 17.12 è rappresentato un circuito demultiplexer a 8 uscite e quindi con 3 ingressi di indirizzo. L'abilitazione al funzionamento del circuito è determinata dalla presenza del segnale stesso all'ingresso dati.

Osservando la tabella della verità riportata nella figura si può facilmente comprendere il funzionamento del demultiplexer.

La presenza dello stato logico 1 sull'ingresso dati abilita tutte le porte AND, ma solo quelle relative alla combinazione dei livelli logici presenti sugli ingressi di indi

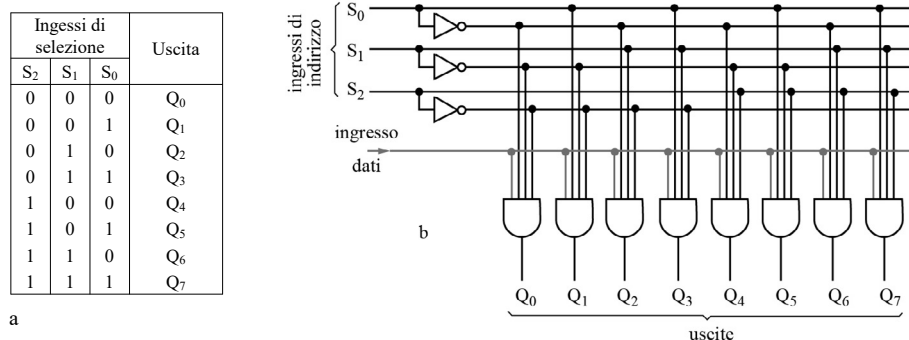


Figura 17.12 – Demultiplexer realizzato con porte AND:

a – tabella della verità;

b – schema logico.

rizzo consentono di trasferire il dato all'uscita. La sequenza dei bit dell'ingresso dati è quindi riprodotta sull'uscita prescelta.

17.10 - Circuiti logici di memoria

I circuiti logici di memoria sono dispositivi capaci di fornire in uscita un segnale che dipende non solo dallo stato dei segnali di ingresso, ma anche dalla sequenza con la quale tali stati di manifestano. Questi circuiti sono caratterizzati da due stati: stato di set, quando l'uscita è a livello 1 e stato di reset, quando l'uscita è a livello 0. Sono circuiti logici di memoria i circuiti elettronici denominati flip-flop (multivibratori).

La maggior parte dei circuiti digitali richiede, per il funzionamento, di poter conservare le informazioni in forma binaria per un certo tempo.

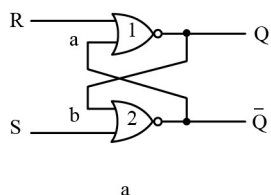
La funzione di memoria è svolta da un dispositivo in grado di mantenere un bit, ricevuto all'ingresso al tempo t_1 , per un intervallo Δt e rendere quindi disponibile l'informazione al tempo $t_2 = t_1 + \Delta t$.

In particolare lo stato delle uscite di questo elemento non dipende, ad un dato istante, solamente da quello dei vari ingressi, ma anche dallo stato dei segnali ricevuti nell'intervallo di tempo precedente e quindi dalla sequenza degli stati dei segnali di ingresso. Per tale motivo i dispositivi di memoria vengono anche denominati sequenziali.

Un circuito di memoria può essere realizzato molto semplicemente collegando due porte NOR come indicato in figura 17.3a. Gli ingressi, che vengono in genere definiti con terminologia inglese, set (simbolo S) e reset (simbolo R) determinano, in relazione al loro stato, i livelli logici delle uscite Q e \bar{Q} . Queste risultano complementari tra di loro nel senso che se $Q = 1$ necessariamente è $\bar{Q} = 0$ e viceversa.

Questi dispositivi di memoria vengono detti flip-flop o anche multivibratori bistabili.

In genere vengono definiti: stato di set, la condizione per cui $Q = 1$ e stato di reset quella per cui $\bar{Q} = 0$; lo stato di reset viene spesso indicato anche con il termine inglese clear (stato di azzeramento).



b

Stato delle uscite all'istante t		Stato assunto dagli ingressi all'istante t		Stato assunto dalle uscite all'istante $t+1$	
Q_t	\bar{Q}_t	S_t	R_t	Q_{t+1}	\bar{Q}_{t+1}
0	1	0	0	0 (= Q_t)	1 (= \bar{Q}_t)
1	0	0	0	1 (= Q_t)	0 (= \bar{Q}_t)
0	1	1	0	1	0
1	0	1	0	1 (= Q_t)	0 (= \bar{Q}_t)
0	1	0	1	0 (= Q_t)	1 (= \bar{Q}_t)
1	0	0	1	0	1
0	1	1	1	stato non possibile	
1	0	1	1	stato non possibile	

Figura 17.13 - Circuito logico flip-flop realizzato con due porte NOR (a) e relativa tabella della verità (b).

Il funzionamento del flip-flop sopracitato, che viene in genere indicato con la sigla R - S , può essere sintetizzato nella tabella della verità rappresentata in figura 17.13b.

Per una migliore comprensione del funzionamento del dispositivo, nella tabella sono stati indicati gli stati delle uscite nell'istante t , gli stati che gli ingressi possono assumere sempre nell'istante n , nonché lo stato delle uscite determinato, all'istante $t + 1$, dai livelli logici presenti agli ingressi.

17.11 - Registri

I registri sono sistemi di memoria costituiti da più circuiti flip-flop opportunamente collegati tra loro e capaci di acquisire informazioni in forma binaria e a carattere non permanente.

Un circuito flip-flop è in grado di immagazzinare un solo bit per cui, per memorizzare una parola formata da n bit, sono necessari n dispositivi. L'insieme di più flip-flop viene denominato con il termine di registro.

I registri sono suddivisi su due categorie:

- a scorrimento (shift-register);
- paralleli.

I registri a scorrimento si possono rappresentare con uno schema a blocchi del tipo indicato in figura 17.14 in cui ogni elemento è previsto per la memorizzazione di un bit.

I dati da memorizzare vengono inviati in forma seriale pervenendo quindi all'ingresso del registro uno alla volta, generalmente in sincronismo con un segnale di cadenzamento (o di clock) costituito da una sequenza di impulsi di periodo costante: ad ogni impulso di clock viene memorizzato un bit.

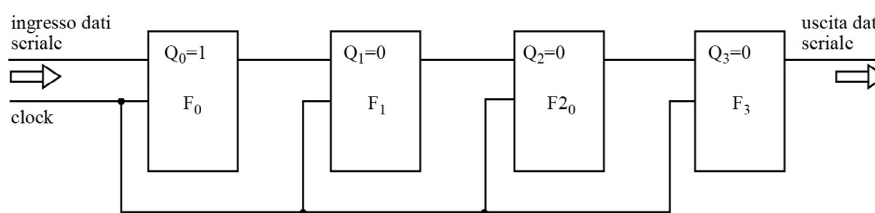


Figura 17.14 - Schematizzazione di un registro a scorrimento.

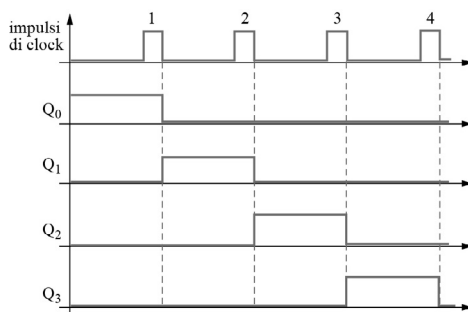


Figura 17.15 - Esempio di funzionamento di un registro a scorrimento a quattro bit con commutazione sul fronte del segnale di clock.

Sempre con riferimento alla figura 17.14 si osservi che i collegamenti tra i vari flip-flop (ad eccezione del primo) sono tali per cui il livello logico presente all'ingresso di ciascuno di essi è determinato dallo stato del bistabile precedente. In questo modo l'uscita Q di ogni elemento assume, subito dopo l'impulso di clock, il livello logico presente all'ingresso prima dell'applicazione dell'impulso di clock stesso. Di conseguenza il dato viene fatto scorrere dall'ingresso all'uscita passando da un flip-flop a quello successivo ad ogni transizione del segnale di clock.

Questo tipo di funzionamento è illustrato in figura 17.15 nella quale si è ipotizzato che inizialmente tutti i bistabili siano in stato di reset tranne il primo che è in stato di set, ma con ingresso a livello 0. Se i dati vengono presentati all'ingresso del registro in successione temporale con la

stessa frequenza del segnale di clock, essi entrano bit dopo bit nel registro e scorrono all'interno del registro stesso verso l'uscita memorizzati di volta in volta nei vari flip-flop. Interrompendo il segnale di clock dopo un numero di impulsi pari al numero di flip-flop che formano il registro, i dati risultano registrati e permangono in memoria per il tempo desiderato. I registri parallelo si possono invece rappresentare con lo schema a blocchi di figura 17.16; anche in questo caso ogni elemento è previsto per la memorizzazione di un bit. Nella configurazione parallelo l'ingresso dei dati avviene attraverso tante linee quanti sono i bit della parola da memorizzare ed il registro presenta quindi altrettanti ingressi ai quali i dati pervengono contemporaneamente.

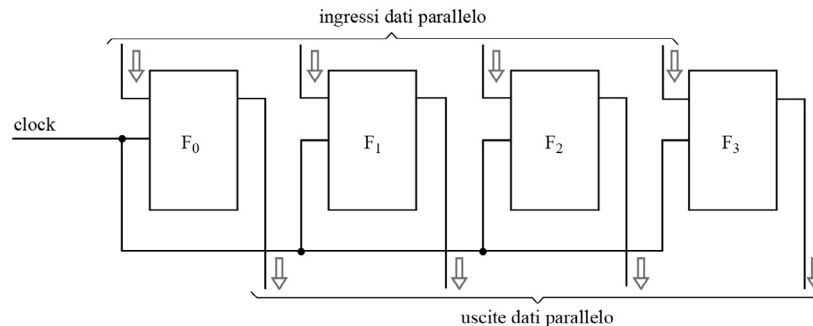


Figura 17.16 - Schematizzazione di un registro parallelo.

I registri, siano essi del tipo a scorrimento o parallelo, sono sempre provvisti comando di Clear e di Preset: con il primo si forzano tutte le uscite del registro allo stato 0, mentre con Preset si portano tutte le uscite allo stato 1. Analogamente a quanto visto per gli ingressi, in uscita dai registri i dati possono essere resi disponibili in forma seriale o parallela.

17.12 - Contatori

I contatori sono circuiti elettronici capaci di acquisire e mantenere un'informazione numerica presentata in ingresso sotto forma di impulsi successivi.

Un contatore è composto da celle di memoria e da porte logiche opportunamente collegate tra loro; l'insieme è in grado di fornire in uscita un segnale che indica il numero degli impulsi applicati all'ingresso del contatore. I contatori si distinguono in relazione alla base della numerazione corrispondente al codice dei segnali presenti all'ingresso: quelli principali sono a base 2 e da essi si possono derivare quelli funzionanti con altri codici. Un contatore binario deve presentare un ingresso e un numero di uscite che dipende dalla lunghezza della parola (numero di bit) la quale, a sua volta, definisce il numero massimo che il dispositivo può contare. Per cui se, ad esempio, il contatore è costituito da 4 flip-flop (e quindi ha 4 uscite) il valore più elevato che può conteggiare è espresso dal numero binario 1111 che corrisponde al numero 15 in forma decimale.

Poiché il numero delle possibili combinazioni M_0 (definito modulo di contatore) che gli stati logici sulle uscite possono determinare risulta $M_0 = 2^n$, il valore massimo di impulsi che il dispositivo può conteggiare è $M_0 - 1$ dato che una delle combinazioni degli stati forniti dalle uscite deve essere assunta come stato iniziale. Dopo l'applicazione di un numero $M_0 - 1$ di impulsi il contatore raggiunge lo stato finale e un ulteriore impulso riporta le uscite allo stato iniziale.

I contatori binari possono essere costituiti collocando opportunamente il necessario numero di

flip-flop, in genere però vengono realizzati in forma integrata in quanto si ottengono, in tal modo, dispositivi di piccolo ingombro ed elevate prestazioni. I contatori binari in commercio sono numerosi e possono essere raggruppati in base al modo di commutazione, al codice di conteggio e al modo di conteggio. In figura 17.17a è raffigurato un contatore binario realizzato con una serie di flip-flop di cui l'uscita di ciascuno è collegata all'ingresso di clock del successivo.

Con questo tipo di circuito la commutazione dei flip-flop avviene quando il segnale passa dal livello 1 a quello 0. Supponendo che inizialmente i flip-flop siano nello stato di reset, il primo impulso fa commutare F_0 la cui uscita passa quindi a $Q_0 = 1$, mentre gli altri flip-flop restano a 0 in quanto ai rispettivi ingressi non vi è stata una transizione del segnale dal livello alto a quello basso. Le uscite del contatore risultano pertanto allo stato indicato nella seconda riga della tabella riportata in figura 17.17b ed il dispositivo ha così contato il primo impulso.

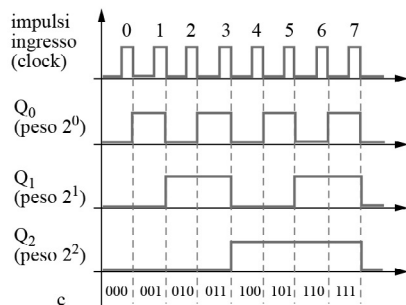
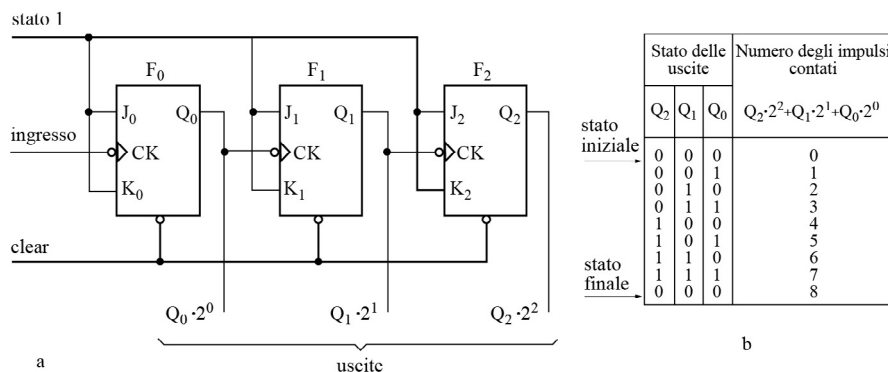


Figura 17.17 Contatore asincrono binario:
a - schema del circuito;
b - stato delle uscite;
c - forma d'onda dei segnali in uscita.

Poiché ora l'ingresso CK del flip-flop F_0 si trova a livello logico 1, al secondo impulso (che riporta F_0 a 0) F_1 commuta a livello 1, mentre F_2 permane a 0. In questa situazione lo stato delle uscite è quello indicato dalla terza riga della tabella e il contatore ha contato 2 impulsi (infatti lo stato 010 corrisponde al valore decimale 2). Gli impulsi successivi determinano tutti gli stati delle uscite indicati nella tabella ai quali corrispondono le forme d'onda riportate in figura 17.17c.

17.13 - Memorie a semiconduttore

Le memorie sono dispositivi capaci di immagazzinare, conservare in modo temporaneo o permanente e restituire, quanto richiesto, informazioni in forma binaria. Possono essere ad accesso casuale o ad accesso sequenziale. Sul piano strutturale si possono avere memorie a matrice, a scorrimento e di massa.

Nei sistemi digitali le memorie costituiscono il supporto su cui sono poste tutte le informazioni necessarie allo svolgimento di una determinata funzione, ossia tutti i bit che costituiscono i dati da elaborare e le istruzioni per il loro trattamento. Le memorie possono essere ad accesso casuale, quando la lettura o la scrittura (2) di un dato avviene in tempi uguali per tutte le locazioni di memoria disponibili o ad accesso sequenziale, quando le suddette operazioni avvengono con tempi che dipendono dalla posizione dell'informazione nella memoria (per leggere o scrivere un dato è necessario scorrere tutti quelli che lo precedono). Dal punto di vista strutturale si possono invece distinguere tre tipi di memorie: singole (o registri), a matrice e di massa.

Le memorie del primo tipo sono in grado di registrare un solo bit o il gruppo di bit che costituisce una parola. In questo secondo caso i bit possono essere letti o scritti sequenzialmente uno alla volta o contemporaneamente. Le memorie a matrice possono memorizzare un numero elevato di bit, ma questi possono essere letti o scritti uno alla volta o al più a gruppi di 4,8 o 16 bit (cioè una parola alla volta). L'accesso alla memoria è di tipo casuale.

Le memorie di massa (dischi, nastri magnetici ecc.), infine, consentono di immagazzinare enormi quantità di bit (dell'ordine di milioni), ma lettura e scrittura avvengono in modo sequenziale.

Dal punto di vista della permanenza dei dati le memorie si distinguono in volatili e non volatili. Le prime mantengono il dato solo se restano alimentate e lo perdono quando l'alimentazione cessa anche per un tempo molto breve; le seconde invece lo mantengono indefinitamente. Le memorie a registro sono, per loro natura, memorie volatili; le memorie a matrice possono essere sia volatili che non volatili, mentre le memorie di massa sono sempre non volatili. Elemento caratteristico di ogni memoria è la capacità, cioè la quantità di bit che può essere immagazzinata. Se n è il numero di bit per parola e m è il numero di parole che si possono immagazzinare, la capacità di memoria C sarà data da:

$$C = n \cdot m$$

Le memorie possono avere capacità variabili tra qualche decina di migliaia a diverse decine di milioni di bit. Nel seguito si ricordano alcune delle soluzioni più diffuse di memorie a matrice realizzate mediante circuiti integrati.

Memorie RAM

Le memorie *RAM* (Random Access Memory) sono costituite da un numero di celle elementari pari al numero di bit che si intende memorizzati. La disposizione di queste celle è a matrice con n righe ed m colonne (Figura 17.18).

La struttura di una memoria *RAM* comprende:

- i decodificatori di indirizzo che in funzione dell'istruzione ricevuta (parola binaria), ossia del bit inviato a ciascun ingresso, abilitano (portano a livello 1) una delle linee di selezione riga (X) e una linea di selezione colonna (Y). La locazione di memoria, che si trova connessa con le linee di selezione colonna e riga contemporaneamente a livello 1, è abilitata e riceve il bit del dato o trasferisce all'uscita il bit precedentemente memorizzato;
- il controllo, che ha il compito di sovraintendere alle operazioni da eseguire nelle locazioni. Ad esempio, se l'ingresso W/R è a livello 1, il bit presente sull'ingresso dati DI viene memorizzato, viceversa se è a livello 0 il bit della locazione selezionata viene letto, ossia viene trasferito all'uscita DO (senza venir cancellato dalla locazione stessa) attraverso il buffer (amplificatore di corrente) mediante le linee D e \bar{D} .

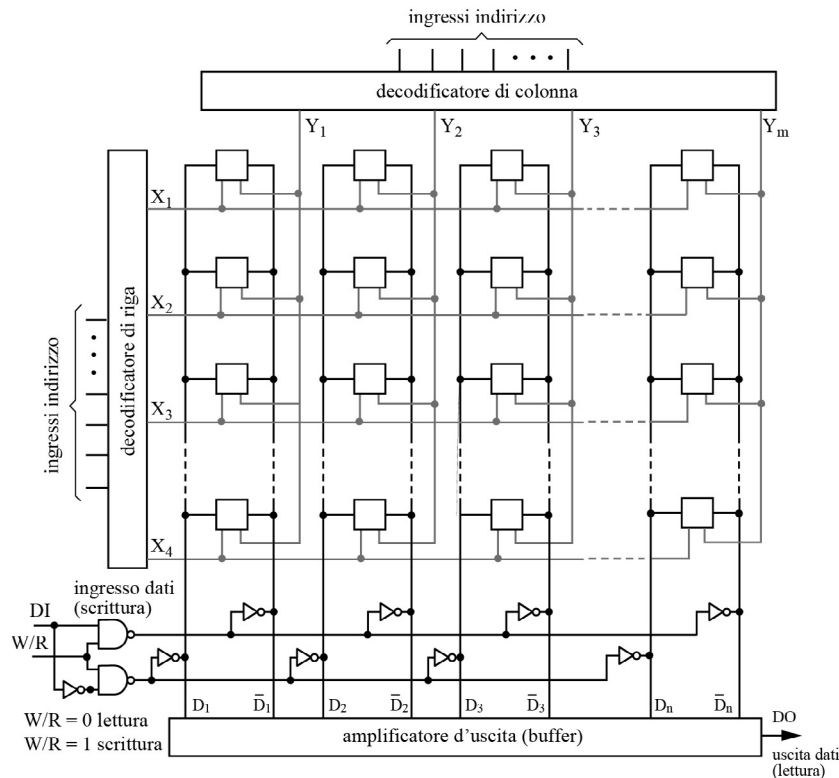


Figura 17.18 - Struttura di una generica memoria RAM. Le linee di selezione della riga e della colonna sono evidenziate in colore.

Si osservi come gli indirizzi vengano inviati in modo parallelo, ossia una parola di vari bit contemporaneamente agli ingressi, mentre i dati vengano letti o memorizzati serialmente e cioè un bit alla volta. Si noti, inoltre, come l'impiego delle memorie implichi la necessità di utilizzare due segnali digitali di cui uno relativo ai dati e l'altro (indirizzo) destinato ad individuare la locazione di memoria. Le memorie *RAM* sono dette volatili in quanto mantengono i dati solo se alimentate; in mancanza di alimentazione, anche per un solo istante, i dati vengono persi.

Memorie ROM

Le memorie *ROM* (Read Only Memory) sono simili alle *RAM*, ma vengono utilizzate per immagazzinare in permanenza dati o programmi che si prevede di non dovere più modificare. Si tratta quindi di memorie non volatili per cui il loro contenuto permane anche in caso di mancanza di alimentazione. Le informazioni contenute da una memoria *ROM* sono immesse all'atto della costruzione della medesima e non possono essere più cambiate.

Memorie PROM, EPROM e EAROM

Le memorie *PROM* (Programmable Read Only Memory) sono memorie di sola lettura come le *ROM*, non sono volatili e possono essere scritte, tramite apparecchi speciali, anche dall'utente. Le memorie *EPROM* (Erasable Programmable Read Only Memory) sono analoghe alle *ROM*, ma presentano il vantaggio di poter essere scritte e cancellate più volte. Per tali operazioni è necessario staccare la memoria dall'apparecchiatura e utilizzare appositi dispositivi. Le memorie *EAROM* (Electrically Alterable Read Only Memory) rappresentano, infine, un ulteriore miglioramento rispetto alle precedenti in quanto possono essere scritte o cancellate durante il servizio anche solo parzialmente.

17.14 - Sistemi a logica programmata

Un sistema a logica programmata è costituito da una o più unità logiche capaci di svolgere determinate funzioni in base ad istruzioni inserite nella memoria dell'unità stessa.

L'insieme delle apparecchiature che costituisce il sistema è detto hardware, mentre i programmi da utilizzare per lo svolgimento delle funzioni previste è detto software.

Nella progettazione e nella costruzione delle apparecchiature elettroniche più complesse, si può operare secondo due criteri:

- *con logica cablata*, quando si ricorre ad un insieme di circuiti digitali assemblati in modo da costituire una rete logica che esegue sempre ed esclusivamente la stessa funzione. Qualsiasi modifica al ciclo operativo può essere ottenuta solo variando il cablaggio della rete stessa;
- *con logica programmata*, quando si ricorre ad una unità logica in grado di realizzare funzioni diverse fra loro sulla base di un insieme di istruzioni (programma) inserito nella memoria dell'unità stessa. I legami logici del ciclo operativo non sono quindi dovuti al cablaggio, ma vengono stabiliti dal programma.

In un sistema a logica programmata si distinguono due parti fondamentali: l'hardware, che comprende le apparecchiature che compongono il sistema e il software, cioè l'insieme dei programmi operativi di cui deve disporre il sistema per svolgere

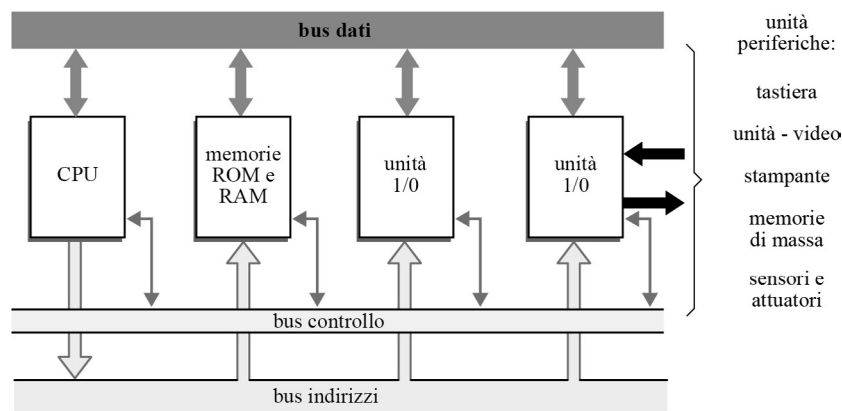


Figura 17.19 - Struttura tipica di un generico sistema a logica programmata.

le funzioni previste. Nella figura 17.19 è riportata la struttura tipica di un sistema a logica programmata; si possono distinguere:

- l'unità di processo o CPU (Central Process Unit);
- l'unità di memoria;
- le unità di ingresso e uscita dei dati (I-input, O-output) che possono essere più di una;
- i collettori (bus) per il trasferimento dei segnali.

La CPU, che può essere considerata l'unità fondamentale del sistema, svolge le seguenti funzioni: interpreta le istruzioni dei programmi⁽³⁾ predisposti per il funzionamento del sistema, distribuisce i comandi ai vari componenti ed esegue le operazioni matematiche o logiche previste dai program-

mi suddetti. Le memorie hanno il compito di immagazzinare in forma binaria sia le istruzioni dei programmi predisposti per il funzionamento del sistema che i dati da utilizzare per le elaborazioni matematiche. Le unità di ingresso-uscita (*I/O*) permettono alle unità esterne al sistema di comunicare con il medesimo sia per l'immissione dei programmi o dei dati necessari al suo funzionamento che per inviare o ricevere comandi.

I collettori (*bus*) sono i collegamenti di cui si avvale la *CPU* per colloquiare con le varie unità funzionali del sistema. Di norma si distinguono:

- collettore dati (*data bus*), che viene utilizzato per gli scambi dei dati tra *CPU*, memorie e unità di ingresso-uscita;
- collettore indirizzi (*addresses bus*), che viene utilizzato per la trasmissione delle parole in binario costituenti l'istruzione, che identifica la locazione di memoria in cui leggere o immagazzinare dati, o l'unità di ingresso-uscita;
- collettore di controllo (*control bus*), che viene utilizzato dalla *CPU* per lo scambio dei segnali di controllo del funzionamento e della disponibilità delle varie parti.

La quantità di conduttori che costituiscono il collettore definisce il numero di bit che la *CPU* può trattare o trasferire contemporaneamente sulle linee del collettore (un bit su ciascuna linea).

È evidente che la velocità di acquisizione e di elaborazione dei dati aumenta con le dimensioni della parola, per contro il sistema risulta più complesso e di conseguenza più costoso.