Modulo 8

Elettronica Digitale

Contenuti:

- *Introduzione*
- *Sistemi di numerazione posizionali*
- *Sistema binario*
- *Porte logiche fondamentali*
- *Porte logiche universali*
- *Metodo della forma canonica della somma per la determinazione delle funzioni logiche di una variabile*
- *Minimizzazione delle funzioni logiche mediante le mappe di Karnaugh*
- *Circuiti combinatori*
- *Circuiti sequenziali*
- *Flip-Flop SR*

Obiettivi:

Mettere l'allievo in grado di conoscere i principi fondamentali su cui si basa l'elettronica digitale.
Mettere l'allievo in grado di capire la differenza tra hardware e software.
Mettere l'allievo in grado di comporre una semplice tabella di verità e relativo circuito logico.
Mettere l'allievo in grado di comprendere la differenza tra circuiti combinatori e circuiti sequenziali.

Introduzione

L'elettronica digitale si occupa di tutti i circuiti che costituiscono i computer o i dispositivi di controllo automatico di tipo elettronico. Per apparati quali i computer, questo ramo dell'elettronica è denominato ***hardware.***
L'informatica si occupa invece di tutti i sistemi operativi e i programmi che permettono di realizzare svariatissime funzioni attraverso il computer, questo ramo dell'elettronica è denominato ***software.***

Sistemi di numerazione posizionali

Viene definito ***sistema di numerazione l'insieme di un numero finito di simboli e l'insieme delle regole che assegnano un solo significato alla scrittura formata da un certo numero degli stessi simboli.***
Il sistema di numerazione da noi utilizzato è quello decimale. Tale sistema è costituito da dieci cifre che possono essere opportunamente combinate, esse sono:

1 2 3 4 5 6 7 8 9 0

Quando noi scriviamo 324, nel sistema decimale, intendiamo che 3 rappresenti la ***cifra più significativa*** e diciamo che è la ***cifra delle centinaia***, 4 rappresenta invece la ***cifra meno significativa*** e diciamo che è la ***cifra delle unità,*** infine la cifra 2 al centro viene detta ***cifra delle decine.*** Il numero può essere scritto come segue:

$$324 = 300 + 20 + 4 = 3* 10^2 + 2* 10^1 + 4* 10^0$$

sappiamo che 10^0 è uguale a 1 e 10^1 è uguale a 10 per cui il numero 324 è costituito dalla somma di tre termini costituiti dalla cifra moltiplicata per una potenza in base 10 con esponente crescente da destra verso sinistra. Se si ripete il procedimento per altri numeri si vede che è sempre possibile ricondurli ad una scrittura come quella esaminata. Da questo segue che ***l'esponente della base*** determina il ***peso della cifra*** per cui è moltiplicato, cioè individua il grado di *significatività* della cifra.

Il sistema di numerazione decimale è un sistema posizionale in quanto la significatività delle cifre è determinata dalla loro posizione nel numero e quindi dall'esponente della base che identifica il sistema di numerazione, cioè 10.

Da quanto esposto si possono desumere alcune regole di carattere generale valide per i sistemi di numerazione posizionali:

- La ***base della potenza*** del sistema di numerazione indica il ***numero di simboli*** da cui è costituito
- In un sistema di numerazione posizionale la cifra all'estrema destra è la ***cifra meno significativa***, mentre quella alla estrema sinistra è la ***cifra più significativa***
- In un sistema di numerazione posizionale il ***peso*** è l'***esponente*** della base del sistema

Vi sono altri sistemi di numerazione posizionali, oltre a quello decimale, che sono utilizzati in elettronica per la elaborazione dei dati:

- Sistema binario
- Sistema ottale
- Sistema esadecimale

E' sempre possibile convertire i vari sistemi di numerazione tra loro, consideriamo il sistema decimale e quello binario e vediamo come avviene la trasformazione.

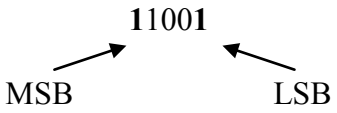
Sistema Binario

Esistono sistemi fisici di diversa natura che sono in grado di trovarsi unicamente in due stati possibili. In particolare, i circuiti elettronici degli elaboratori lavorano in due soli stati che corrispondono a valori diversi di una tensione elettrica. I dispositivi per l'elaborazione delle informazioni dalle memorie ai microprocessori e tutti gli apparati per i collegamenti con le periferiche funzionano in due soli stati ed è quindi necessario stabilire delle convenzioni per la rappresentazione delle informazioni. La situazione fisica, di uno o più dispositivi all'interno dei ***circuiti integrati***, viene codificata attraverso un ***sistema di numerazione posizionale detto sistema binario.***
Il nome binario deriva dal fatto che tale sistema ha ***due sole cifre***:

0 e 1

denominate ***bit***.

Il termine ***bit*** è l'abbreviazione di ***Binary Digit***.
Un insieme di più cifre del tipo 1110011 viene denominata ***stringa o parola***.
In particolare una parola di ***otto bit*** viene denominata ***byte***.
La ***cifra più significativa*** è indicata con la sigla ***MSB (Most Significant Bit).***
La ***cifra meno significativa*** è indicata con la sigla ***LSB (Least Significant Bit).***



Nella tecnica digitale si usano, generalmente, parole di lunghezza standard di 9, 12, 16, 32 bit.

Conversione da decimale a binario

Il metodo consiste nel dividere il numero decimale per la base del sistema in cui lo si vuole convertire: nel nostro caso la base è 2, tale operazione prosegue fino a quando il risultato della divisione è uguale a zero.
Il resto di ciascuna divisione rappresenta il numero binario corrispondente. La cifra meno significativa risulta essere il resto della prima operazione e la cifra più significativa è il resto dell'ultima divisione.
Vediamo l'esempio seguente:
consideriamo il numero 24 e trasformiamolo nell'equivalente numero binario

$$24 : 2 = 12 + \mathbf{0} \longleftarrow \text{LSB}$$

$$12 : 2 = 6 + \mathbf{0}$$

$$6 : 2 = 3 + \mathbf{0}$$

$$3 : 2 = 1 + \mathbf{1}$$

$$1 : 2 = 0 + \mathbf{1} \longleftarrow \text{MSB}$$

Il numero decimale 24, convertito in numero binario diventa 11000.

Conversione da binario a decimale

Consideriamo il numero binario 111 e lo convertiamo nel corrispondente numero decimale. Moltiplichiamo ciascuna cifra del numero per una potenza la cui base è 2 e, per esponente, un esponente crescente da destra a sinistra partendo da 0 per la prima cifra che è quella meno significativa.

$$(111)_2 = 1 * 2^2 + 1 * 2^1 + 1 * 2^0 = 4 + 2 + 1 = 7$$

Il numero 7 è il numero decimale corrispondente al numero binario 111.

Operazioni nel sistema binario

Il sistema di numerazione binario è un sistema posizionale e in esso si possono eseguire le quattro operazioni come nel sistema decimale.

Somma binaria

Come nel sistema decimale anche qui abbiamo una somma e un riporto. Sappiamo che il sistema è costituito da due cifre per cui la tabella di base delle ***somme*** e dei ***riporti*** parte dalla somma di due bit secondo lo schema seguente:

$$0 + 0 = 0 \quad \text{Riporto } 0$$

$$0 + 1 = 1 \quad \text{Riporto } 0$$

$$1 + 0 = 1 \quad \text{Riporto } 0$$

$$1 + 1 = 0 \quad \text{Riporto } 1$$

Come in tutti i sistemi posizionali il ***riporto*** viene sommato alla cifra più significativa.

<u>Esempio:</u>

Sommiamo i seguenti numeri binari

1001 + 0111 =

1001 +

0111 =

10000

Riporto
Nella somma binaria il riporto è parte del risultato, cioè il numero è 10000.

Sottrazione binaria

Il sistema utilizzato per effettuare una sottrazione binaria che viene ora descritto è denominato ***complemento a 2.*** Questo metodo è suddiviso in due fasi:

- Complemento a uno
- Complemento a due

Iniziamo a definire il complemento a 1:

si definisce complemento a 1 di un numero binario quel numero che si ottiene sostituendo ogni cifra del numero con il suo complementare.

La cifra complementare di 0 è 1 e la cifra complementare di 1 è 0.

Dopo avere ottenuto il complemento a uno si somma ad esso il bit 1 alla cifra meno significativa, cioè la LSB.
Il secondo tipo di "operazione" viene denominata ***complemento a 2***, che porta al risultato della sottrazione sommando al minuendo il risultato del complemento a 2.
E' possibile che nel risultato compaia un ***riporto*** il quale**,** in questo caso, non fa parte del risultato ma stabilisce il ***segno della operazione***. Quando questo accade, per ottenere il numero corretto bisogna effettuare un ulteriore complemento a uno e a 2 per ottenere il risultato esatto.

Quando il riporto vale 1 è di segno positivo, quando il riporto vale 0 è di segno negativo.

<u>Esempio:</u>

Vogliamo sottrarre il numero 0101 al numero 1000

Minuendo 1000

Sottraendo 0101

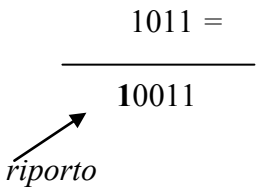
Complemento a 1 del sottraendo 1010
Effettuiamo il complemento a 2 1010 +

1 =

1011

Sommiamo ora il numero ottenuto al minuendo

1000 +



Il riporto è **1** *quindi il segno del numero è positivo.*

Moltiplicazione binaria

Le regole per la moltiplicazione binaria si possono desumere da quelle della moltiplicazione nel sistema decimale. Inoltre basta tenere presente che una moltiplicazione è un insieme di somme ed è in questo modo che gli elaboratori effettuano tale tipo di operazione.

Divisione binaria

Per la divisione il metodo è più complesso e non verrà trattato nel presente testo.

Logica positiva e negativa

Si è detto, all'inizio, che i circuiti possono assumere due soli stati per cui ad essi vanno associati i bit del sistema binario per codificare con esso, sia la situazione di ingresso e di uscita sia lo stato in un determinato punto del circuito. Anche in questo caso si devono stabilire delle regole. Identifichiamo le situazioni sopra descritte con la presenza o l'assenza di una tensione, si tratta di stabilire quale bit associare alla presenza della tensione e quale alla assenza di tensione.
I segnali utilizzati in elettronica digitale sono di tipo ***digitale***, cioè possono assumere valori discreti nel tempo, ne è un esempio la tensione disegnata nel grafico di figura 1.

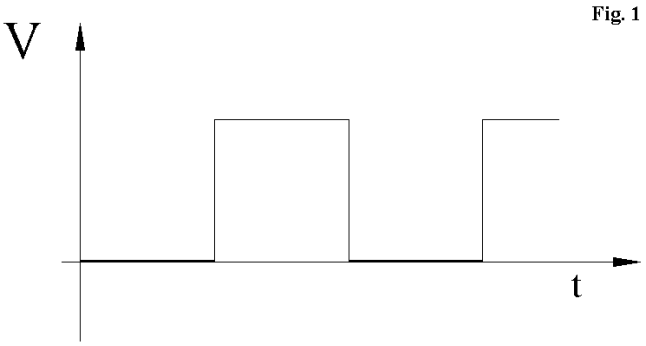
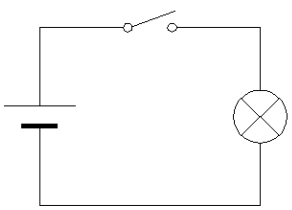


Fig. 1

Prendiamo in considerazione dei semplici dispositivi che abbiano due sole posizioni possibili quali, ad esempio, l'interruttore e la lampadina come nel circuito disegnato.



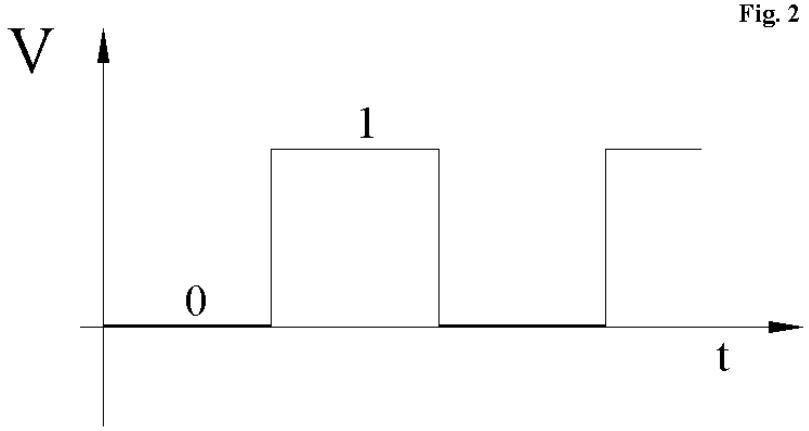
Indichiamo con ***logica positiva*** quella nella quale al bit 0 associamo:

- Tensione nulla
- Interruttore aperto
- Lampadina spenta

Mentre al bit 1 associamo:

- Tensione diversa da zero
- Interruttore chiuso
- Lampadina accesa

Fig. 2



Il grafico di figura 2 mette in evidenza come ad una tensione pari a 0 Volt si faccia corrispondere il bit 0 mentre, alla tensione diversa da zero si faccia corrispondere il bit pari a 1.

Indichiamo con ***logica negativa*** quella nella quale al bit 0 associamo:

- Tensione diversa da zero
- Interruttore chiuso
- Lampadina accesa

Mentre al bit 1 associamo:

- Tensione nulla
- Interruttore aperto
- Lampadina spenta

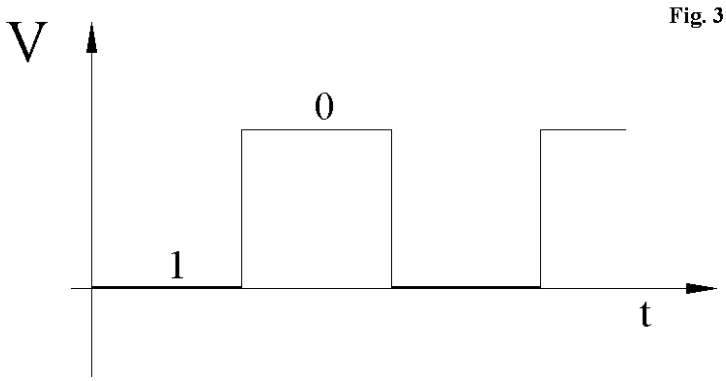


Fig. 3

Il grafico di figura 3 mette in evidenza come ad una tensione pari a 0 Volt si faccia corrispondere il bit 1 mentre, alla tensione diversa da zero si faccia corrispondere il bit pari a 0.

Noi utilizziamo la ***logica positiva*** per tutte le applicazioni.

Porte logiche fondamentali

Prima di iniziare lo studio dei circuiti di elettronica digitale introduciamo degli operatori di tipo logico.
Negli elaboratori elettronici, oltre alle operazioni aritmetiche infatti, si eseguono delle operazioni logiche. I circuiti preposti a questo tipo di operazione vengono detti ***porte logiche***. Le ***porte logiche fondamentali sono tre:***

- ***AND: esegue il prodotto logico***
- ***OR: esegue la somma logica***
- ***NOT: esegue la negazione logica***

Ciascuna di queste porte è definita mediante un ***simbolo grafico*** e una ***tabella di verità*** che andiamo ora ad analizzare. Per ciascuna porta viene inoltre disegnato il corrispettivo circuito a interruttori che può servire per comprendere come si arriva ai dati della tabella di verità. Per circuiti logici complessi la realizzazione del circuito a interruttori non è più conveniente in quanto diventa troppo complesso.

Porta logica AND

La porta logica **AND** esegue il prodotto logico secondo la seguente tabella di verità:

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Fig.4

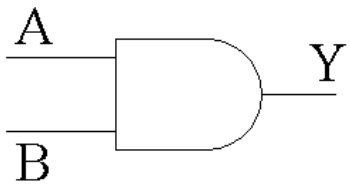
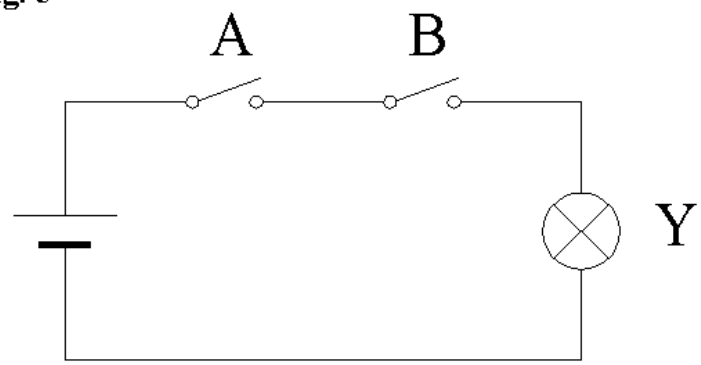


Fig. 5



In figura 4 e figura 5 sono disegnati il simbolo grafico e lo schema ad interruttori della porta. Se ci soffermiamo sul simbolo grafico in esso si individuano due morsetti di ingresso indicati con **A** e **B** e denominati ***variabili di ingresso***. In ciascun ingresso si possono presentare i bit 1 o 0 e, in base alla combinazione dei valori di ingresso, l'uscita **Y** assume un valore ben definito che può essere 1 o 0 e che viene denominato ***funzione di uscita***. Consideriamo ora il circuito a interruttori in cui le variabili di ingresso A e B della porta logica sono rappresentate dai due interruttori e la variabile di uscita è costituita dalla lampada Y. Il circuito è alimentato da una batteria e i due interruttori sono inseriti in serie fra loro e con la lampada. Come già detto lavoriamo in logica positiva per cui la lampada si accende (bit 1) quando entrambi gli interruttori sono chiusi (bit 1), negli altri casi la lampada rimane spenta (bit 0), questo è evidenziato dalla analisi della tabella di verità.
La porta logica considerata è costituita da due ingressi ed una uscita ma vi sono porte logiche che hanno tre ingressi ed in questo caso il numero delle possibili combinazioni degli ingressi cambia.

Per la determinazione del numero di combinazioni di ingresso possibili si procede come segue:

- ***Si conta il numero di ingressi della porta***
- ***Si considera il sistema di numerazione in cui ci si trova (nel nostro caso il sistema binario)***
- ***Si prende la base del sistema di numerazione e la si eleva ad una potenza pari al numero degli ingressi, cioè , nel nostro caso, $2^2 = 4$***
- ***Il numero 4 rappresenta il numero di combinazioni possibili***

Per ordinare correttamente e senza tralasciare nessuna combinazione, specialmente quando le variabili diventano 4 o 5 o più, si procede come segue:

- ***Si parte dalla prima colonna a sinistra e si scrivono , partendo dal bit 0, tanti 0 quante sono le combinazioni diviso 2 e poi i bit 1per le restanti combinazioni***
- ***Nella seconda colonna , sempre partendo con i bit uguali a 0, se ne scrivono la metà dei precedenti e così per le successive colonne***

Si nota che il numero di zeri e uno corrispondono ad una potenza che ha per base 2 (se il sistema considerato è quello binario) e per esponente il "peso" della cifra corrispondente considerando la colonna di estrema destra come quella che contiene le cifre meno significative.

Porta logica OR

Fig.6

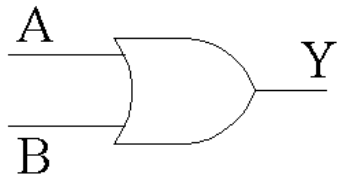
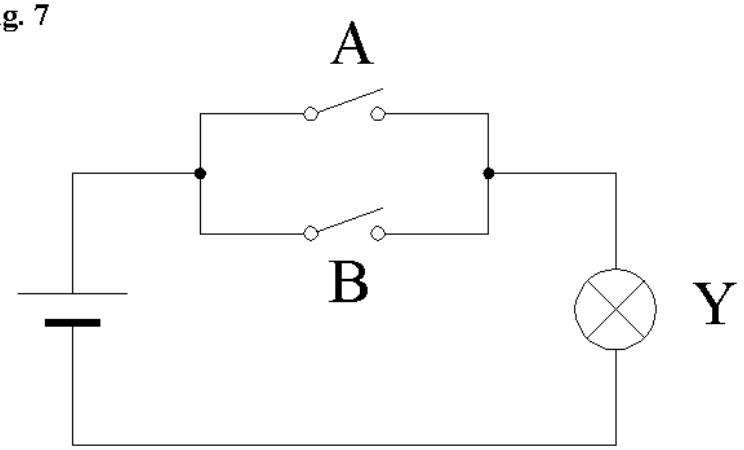


Fig. 7



La porta logica **OR** esegue la somma logica secondo la seguente tabella di verità:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

In figura 6 e 7 sono disegnati il simbolo grafico e lo schema ad interruttori. In questo caso si vede che i due interruttori sono posti tra loro in parallelo per cui è sufficiente che uno dei due sia chiuso (bit 1) perché la lampada sia accesa. Nella tabella di verità si ha che la funzione di uscita **Y** è uguale a 0 solo quando le due variabili di ingresso **A** e **B** a zero.

Porta logica NOT

La porta logica **NOT** esegue la ***negazione*** o ***complementazione*** della variabile di ingresso secondo la seguente tabella di verità:

A	Y
0	1
1	0

In figura 8 e 9 sono disegnati il simbolo grafico e lo schema ad interruttori della porta logica.

Fig.8

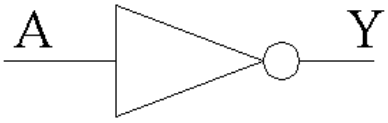
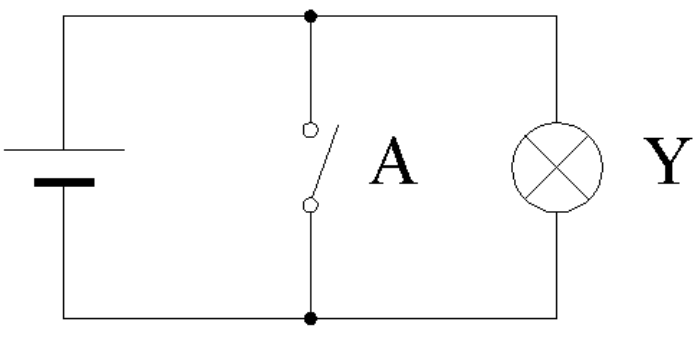


Fig. 9



In questo caso si ha ***una sola variabile di ingresso A*** e l'uscita assume il valore opposto rispetto a quello di ingresso. L'uscita $Y = \bar{A}$ viene detta ***negazione o complementare di A. La barra sopra la lettera indica sempre la presenza di una negazione.*** Nel simbolo grafico il pallino nel vertice del triangolo rappresenta la negazione. Dalla analisi del circuito con l'interruttore si può comprendere il funzionamento della porta. Quando l'interruttore è aperto (bit 0) la corrente circola nel circuito la lampada si accende (bit 1). Quando l'interruttore è chiuso (bit 1) si crea un corto circuito per cui la lampada risulta spenta (bit 0) perché non è percorsa da corrente.

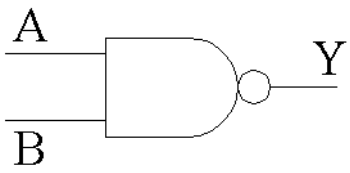
Porte Logiche Universali

Le ***porte logiche universali*** sono due e vengono chiamate così perché con esse si possono realizzare anche le porte logiche fondamentali:

- ***Porta logica NAND***
- ***Porta logica NOR***

Porta logica NAND

Fig.10



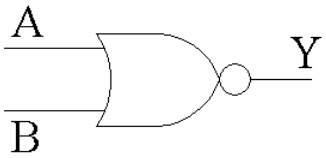
Il simbolo grafico della porta NAND è costituito da quella AND più il pallino che indica la negazione cioè la porta NOT come si vede nella figura 10. Infatti è costituita dall'insieme delle due porte e la tabella di verità è quella relativa alla ***porta AND negata***.

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

La porta NAND realizza il prodotto logico negato.

Porta logica NOR

Fig.11



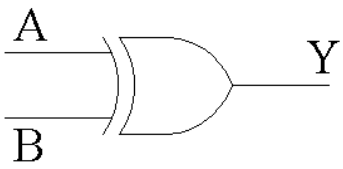
Il simbolo grafico della porta NOR è costituto da quella OR più il pallino che indica la negazione cioè la porta NOT come si vede in figura 11. Infatti è costituita dall'insieme delle due porte e la tabella di verità è quella relativa alla ***porta OR negata***.

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

La porta NOR realizza la somma logica negata.

Porta logica XOR

Fig.12



Questa porta logica non fa parte delle porte universali ma riveste una particolare importanza per le sue applicazioni tra cui il confronto di bit. La funzione logica che si realizza all'uscita è la seguente:

$$Y = A\bar{B} + \bar{A}B$$

La sua tabella di verità è la seguente:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Come si vede dalla tabella di verità la funzione di uscita è uguale a 1 solo quando le variabili di ingresso sono diverse fra loro.

Esercitazioni

- *Realizzare una porta AND con sole porte NAND*
- *Realizzare una porta AND con sole porte NOR*
- *Realizzare una porta OR con sole porte NOR*
- *Realizzare una porta OR con sole porte NAND*
- *Realizzare una porta NOT con sole porte NAND*
- *Realizzare una porta NOT con sole porte NOR*

Teoremi dell'algebra di Boole

L'algebra che esprime le situazioni di grandezze fisiche in un circuito digitale è denominata ***Algebra di Boole o Algebra delle commutazioni.***
Quando si studiano e si progettano i circuiti digitali si deve tenere presente quelli che sono i costi di realizzazione per cui i vari circuiti logici vanno ottimizzati per avere una buona resa e costi contenuti. Per realizzare questo si devono analizzare le tabelle di verità delle funzioni e scrivere le funzioni logiche minimizzate per poi passar alla realizzazione pratica della funzione.
Analizziamo alcuni teoremi fondamentali.

Teorema di identità

A + 0 = A

A * 1 = A

Teorema di annullamento

A + 1 = 1

A * 0 = 0

Esercitazione
Considera i teoremi sopra menzionati e realizza, per ciascuno, il corrispondente circuito ad interruttori disegnandolo di fianco a ciascun teorema. Confrontati poi con i compagni.

Primo teorema dell'assorbimento

A + AB = A

A (A + B) = A

Secondo teorema dell'assorbimento

A + AB = A + B

Teoremi di De Morgan

Il primo teorema di De Morgan dice che la somma negata di due o più variabili è uguale al prodotto delle singole variabili negate.

$$\overline{A + B} = \overline{A} * \overline{B}$$

Il secondo teorema di De Morgan dice che il prodotto negato di due o più variabili è uguale alla somma delle variabili ciascuna negata.

$$\overline{A * B} = \overline{A} + \overline{B}$$

Questi teoremi, come quelli precedenti,sono alcuni dei teoremi che servono per poter ***minimizzare le funzioni logiche***.

Esercitazione
Considera i teoremi dell'assorbimento e i teoremi di De Morgan e dimostra le identità utilizzando le tabelle di verità.

Metodo della forma canonica della somma per la determinazione delle funzioni logiche di una variabile

Mediante questo metodo si può ottenere la funzione di uscita se è nota la tabella di verità. In questo caso però non si ha una minimizzazione della funzione per cui, per ottimizzarla si devono applicare i teoremi ricordati sopra.

Per forma canonica della somma si intende una espressione in cui sono presenti tutte le variabili negate o non negate e i termini sono legati fra loro da una operazione di somma logica.

Esempio:
Supponiamo di avere la seguente tabella di verità dove A, B e C sono le tre variabili di ingresso e Y è la variabile di uscita.

A	B	C	Y	Mintermini
0	0	0	0	
0	0	1	0	
0	1	0	1	$\bar{A}B\bar{C}$
0	1	1	1	$\bar{A}BC$
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	1	ABC

Per determinare la funzione si considerano tutte le combinazioni in cui la variabile di uscita vale 1 e, in corrispondenza si determinano i ***mintermini che sono dati dal prodotto di tutte le variabili di ingresso negate o non negate a seconda che, nella combinazione, valgano zero oppure 1.***
La variabile di uscita Y è data da

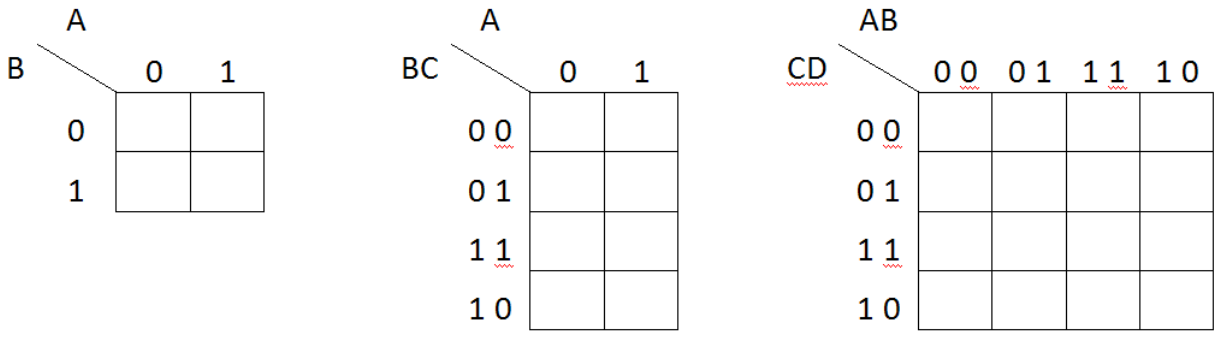
$$Y = \bar{A}B\bar{C} + \bar{A}BC + ABC$$

Esercitazione:
Costruisci liberamente una tabella di verità a tre variabili e determina la funzione di uscita utilizzando il metodo della forma canonica della somma.

Minimizzazione delle funzioni logiche mediante le Mappe di Karnaugh

Le funzioni logiche si possono scrivere, partendo dalla tabella di verità, come indicato precedentemente con le forme canoniche. Tali forme canoniche possono essere semplificate per ottenere forme circuitali più semplici. Un metodo per poter ottenere tale semplificazione è quello di utilizzare le mappe di Karnaugh. La mappa di Karnaugh è costituita da un quadrato o da un rettangolo che al suo interno può contenere altri quadrati e questo dipende dal numero delle variabili di ingresso. Ogni quadrato (chiamato anche cella) corrisponde ad una possibile combinazione delle variabili di ingresso e al suo interno contiene il valore della variabile di uscita.
La caratteristica fondamentale della struttura è dovuta al fatto che quando si passa da una cella a quella vicina (sulla stessa riga o colonna) può variare solo una variabile alla volta. Quanto detto avviene anche per le due celle agli estremi di una riga o di una colonna. Nella figura 18 sono visualizzate le mappe per due, tre e quattro variabili.

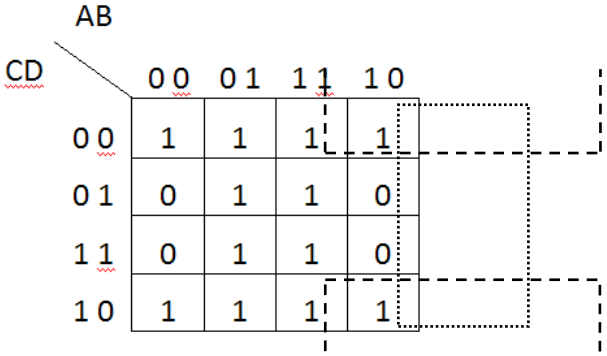
Fig. 18



Noi consideriamo mappe con quattro variabili al massimo e vediamo come si procede per la compilazione della mappa e la stesura della funzione di uscita.

Fig. 19

A	B	C	D	U
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

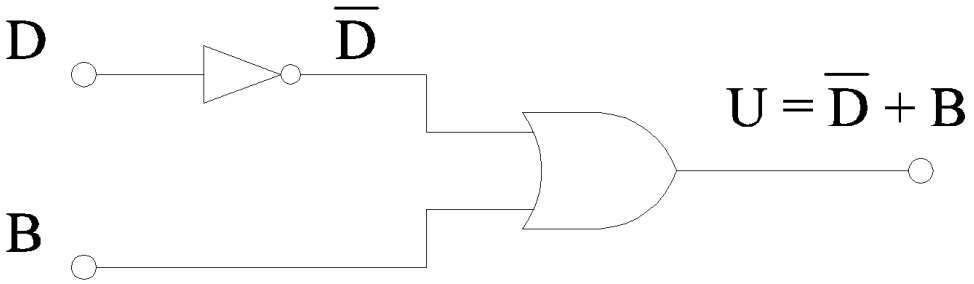


Determinazione della funzione di uscita con l'uso dei mintermini

Prendiamo in esame la funzione logica di quattro variabili definita dalla tabella di verità di figura 19. Si disegna la Mappa di Karnaugh con le quattro variabili e relative combinazioni, si inseriscono nelle celle corrispondenti i valori 1 della variabile, successivamente si raggruppano gli 1 posti in celle vicine in blocchi di uno, due, quattro, otto (secondo le potenze in base due) e in questo modo si riducono i mintermini della funzione in quanto una variabile rimane costante. Eseguita tale operazione si scrive la funzione come somma dei mintermini ottenuti.
La funzione si scrive quindi, come somma dei prodotti di ciascun insieme raggruppato. Il prodotto è dato dalle variabili che non cambiano nel raggruppamento e la variabile che vale uno viene presa diretta, mentre la variabile che vale zero viene presa negata.
Nella mappa di figura 19 si raggruppano i quattro uno delle caselle superiori con gli ultimi quattro delle caselle inferiori in cui rimane invariata la variabile D con valore 0 (zero) per cui viene scritta come $\overline{D}$ negata, inoltre si raggruppano gli otto uno centrali in cui l'unica variabile che rimane invariata è B con valore 1 (uno) quindi viene scritta in forma diretta cioè: B. La funzione di uscita sarà data dal prodotto dei mintermini:

$$U = \overline{D} + B$$

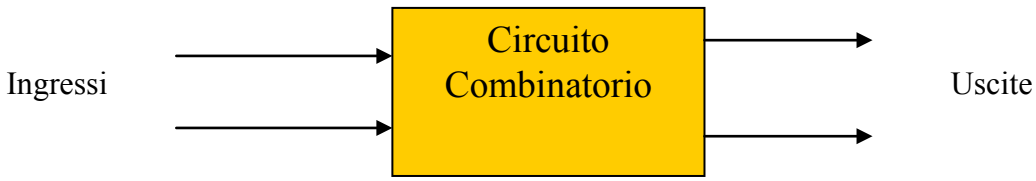
Che è l'espressione della funzione minimizzata ed ora se ne può disegnare il circuito logico.



Circuiti combinatori

Per circuito combinatorio si intende un blocco logico con uno o più variabili di ingresso e di uscita , in cui queste ultime dipendono solo dagli stati attuali degli ingressi.

Sono circuiti che non hanno capacità di memorizzare la situazione precedente.



I circuiti combinatori possono eseguire diverse funzioni tra cui:

- Esecuzioni di operazioni aritmetiche
- Esecuzioni di operazioni logiche

- Selezionare dati
- Conversione di dati

Esempio:

Consideriamo l'esecuzione di operazioni aritmetiche e vediamo come si possa eseguire la somma di due bit.
Supponiamo di voler sommare 1 a 1da cui si ottiene:

$$1 + 1 = 0 \qquad \text{Riporto } 1$$

In ingresso abbiamo due variabili costituite dai due bit e in uscita abbiamo due funzioni:

- La somma
- Il riporto

E' necessario un circuito logico costituito da due ingressi e due uscite. Costruiamo la tabella di verità per la somma aritmetica e il suo riporto, in base alla quattro combinazioni di ingresso.
Indichiamo con:

A primo bit di ingresso

B secondo bit di ingresso da sommare al primo

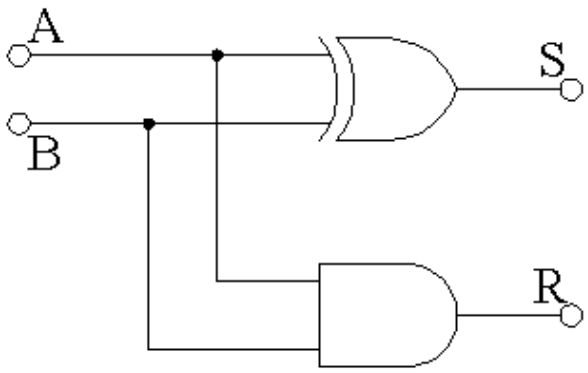
S somma dei due bit

R riporto dei due bit

A	B	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

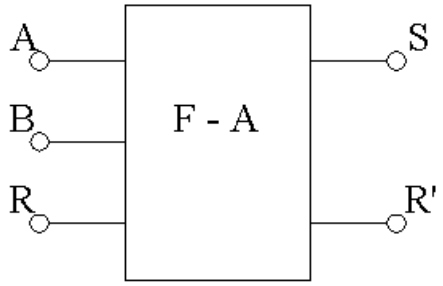
Guardiamo la colonna della funzione di uscita **S,** essa corrisponde alla tabella di verità della porta XOR, mentre la colonna della funzione di uscita **R** corrisponde alla tabella di verità della porta AND per cui costruiamo il circuito logico di figura 13.

Fig. 13



Utilizzando due porte logiche si ottiene la somma aritmetica di due bit. Questo circuito logico viene chiamato ***half – adder.***
Il circuito in grado di sommare fra loro due bit è denominato ***full – adder*** e lo schema blocchi è quello rappresentato in figura 14.

Fig. 14



<u>Esercitazione:</u>
Realizza il circuito logico del full – adder dopo avere scritto la tabella di verità. Determina la funzione di uscita utilizzando la forma canonica della somma.

Codici

Nell'ambito della elaborazione e trasmissione dei dati, in elettronica, trovano larga applicazione i ***codici*** che hanno diverse finalità tra cui:

- Trasmissione di dati o simboli
- Rilevazione di errori intervenuti durante la trasmissione

Un esempio di codice è sicuramente il sistema binario in quanto, attraverso le ***stringe binarie***, vengono inviati i dati al microprocessore per la elaborazione e successivamente restituiti alle periferiche. Dato che tale codice presuppone stringhe molto lunghe di bit se ne utilizzano altri che associano, ad ogni numero o simbolo, una ***parola di quattro bit.***
La codifica si rende necessaria per poter memorizzare, trasmettere e ricevere in tempi brevi le informazioni.
Ci sono vari tipi di codici, quali ad esempio:

- ***BCD***
- ***ASCII***
- ***Codice Gray***
- ***Codice a eccesso 3***
- ***Codice Aiken 2421***

Codice BCD

Questo codice codifica ciascuna cifra decimale mediante una parola di quattro bit, secondo la tabella seguente.

Dec.	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Nella colonna di sinistra sono indicati i numeri decimali e nella colonna di destra sono indicati i corrispettivi in codice BCD. Dato che le combinazioni possibili con quattro bit sono $2^4 = 16$, in questo caso vengono utilizzate solo le prime dieci combinazioni e le altre vengono chiamate ***combinazioni di indifferenza***. Mediante questo codice si possono scrivere i numeri decimali associando, a ciascuna cifra, la corrispondente parola di quattro bit.

Codice ASCII (American Standard Code for Information Interchange)

Questo codice è di tipo alfanumerico, a sette bit e può rappresentare 238 caratteri. Nel caso si voglia trasmettere un numero di caratteri maggiore si può utilizzare il codice ASCII a otto bit.

Circuiti sequenziali

Vengono definiti ***circuiti sequenziali*** quelli in cui il valore della variabile di uscita dipende dalla situazione degli ingressi nell'istante considerato ma anche dalla loro situazione negli istanti precedenti. Una rete sequenziale sarà considerata tale se in essa sono contenuti ***circuiti di memoria,*** cioè dispositivi in grado di mantenere l'informazione ricevuta dagli ingressi.
Questi circuiti vengono utilizzati per:

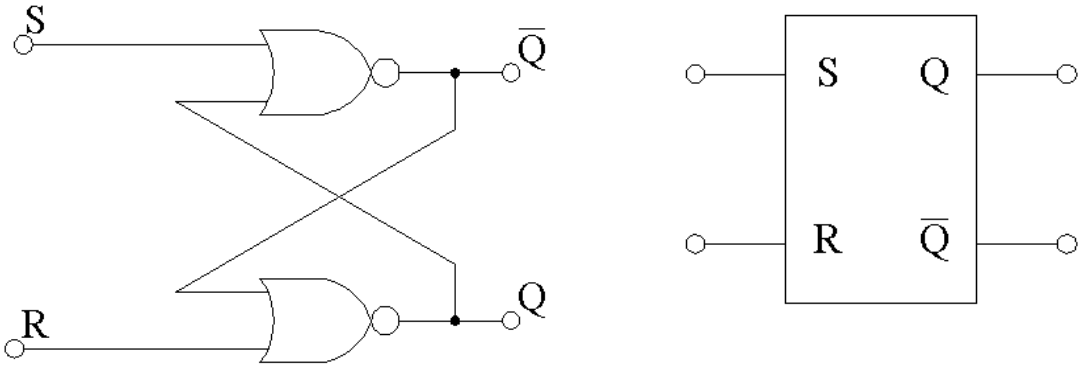
- circuiti di memoria: dispositivi in grado di trattenere le informazioni ricevute e restituirle in base ad una determinata situazione presente agli ingressi.
- contatori: dispositivi in grado di cambiare stato, secondo una determinata frequenza, in base ad una determinata situazione presente agli ingressi.

Tutti questi dispositivi sono costituiti da circuiti sequenziali semplici che vengono denominati ***flip-flop***. Noi analizziamo un semplice flip-flop, rimandando l'analisi più dettagliata ad un testo specifico.

FLIP – FLOP SR

Il più comune circuito di memoria è il ***flip-flop***. Consideriamo un flip-flop costituito da porte NOR come quello di figura 15. Esso è costituito da due porte NOR e si vede che uno degli ingressi di ciascuna porta è collegato all'uscita dell'altra, per cui il dispositivo è costituito da due ingressi e due uscite. Gli ingressi sono denominati ***SET e RESET.*** Le uscite, indicate con Q e $\bar{Q}$, assumono sempre una il valore opposto all'altra.

Fig.15 **Fig.16**



Il flip-flop di figura è denominato ***flip-flop SR*** ed ha il seguente funzionamento:

Quando entrambi gli ingressi S ed R si trovano a zero le due uscite Q e $\bar{Q}$ non cambiano di stato rispetto alla situazione precedente.Quando l'ingresso S è a 0 e R è a 1 allora l'uscita Q si porta a 0, mentre con S a 1 e R a 0 l'uscita Q si porta a 1. Se gli ingressi si trovano contemporaneamente

a 1 la condizione viene detta di indifferenza perché l'uscita può portarsi indifferentemente a 1 o a 0.
La tabella di verità è la seguente:

S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	situazione di indifferenza	

Oltre a questo tipo di flip-flop ve ne sono altri forniti di un maggiore numero di ingressi ciascuno con una sua specifica funzione, si ricordano i flip-flop JK, D, ecc. Le opportune combinazioni di più flip-flop permettono di ottenere, come già osservato precedentemente, circuiti più complessi che svolgono funzioni specifiche fino ad arrivare, ad esempio, al microprocessore il quale elabora le informazioni che gli giungono dalle varie periferiche. Uno schema semplificato di un sistema computerizzato, può essere quello di figura 17. In esso sono presenti:

- CPU: unità principale di elaborazione dei dati
- M: unità di memorie per immagazzinare i dati in entrata e in uscita
- I / O: unità periferiche di ingresso ed uscita delle informazione da elaborare

Queste unità sono collegate fra loro da diversi canali, denominati ***bus***, per la trasmissione dei dati fra le varie unità.

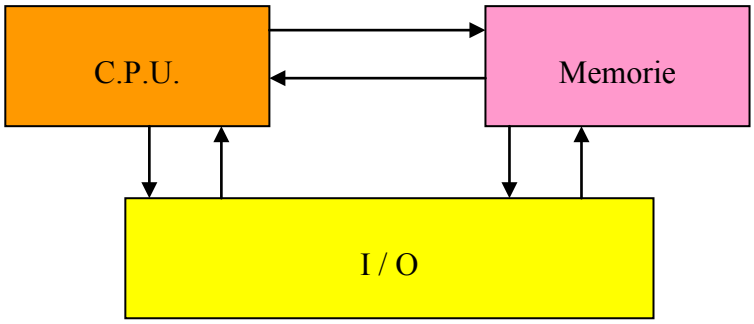


Fig. 17

Per ciascun blocco si può effettuare uno studio di progettazione approfondita che esula dalle finalità di questo modulo.