PORTE LOGICHE

Premessa

Le principali parti elettroniche dei computer sono costituite da circuiti digitali che, come è noto, elaborano segnali logici basati sullo 0 e sull'1.

I mattoni fondamentali dei circuiti logici sono, appunto, le porte logiche che sono in grado di soddisfare un'algebra particolare, detta ***algebra binaria,*** sviluppata dallo scienziato George Boole (1815-1864).

In questi appunti si introdurranno le porte logiche fondamentali in grado di svolgere le operazioni di somma, prodotto e negazione indicate, rispettivamente, con OR (somma logica), AND (prodotto logico) e NOT (negazione o complementazione).

Si descriveranno, inoltre, le porte logiche derivate NOR (OR seguito da un NOT), NAND (AND seguito da un NOT), XOR (noto come OR esclusivo o circuito di anticoincidenza), XNOR (noto come NOR esclusivo o circuito di coincidenza).

Ogni porta logica ha una o più variabili di ingresso ed una sola variabile di uscita. Le variabili di ingresso e di uscita sono di tipo digitale per cui è possibile inserire in una tabella tutte le possibili combinazioni che si possono verificare tra le variabili di ingresso. L'uscita assume il valore 0 oppure il valore 1 in corrispondenza di ciascuna combinazione delle variabile di ingresso in funzione della definizione assegnata.

Porte logiche fondamentali

Somma logica OR

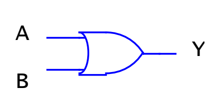
Si effettua su due o più variabili, l'uscita assume lo stato logico 1 se almeno una variabile di ingresso è allo stato logico 1.

Nel caso di due variabili di ingresso A e B, detta Y la variabile di uscita, si scrive:

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

e si legge A or B.

Nella seguente figura si mostra la tabella della verità con le quattro possibili combinazioni tra A e B ed il simbolo logico relativo ad una porta OR a due ingressi. Nella colonna Y si sono posti i valori assunti dall'uscita Y che soddisfa la definizione della porta OR.



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

Prodotto logico AND

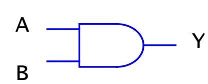
Si effettua su due o più variabili, l'uscita assume lo stato logico 1 solo se tutte variabile di ingresso sono allo stato logico 1.

Nel caso di due variabili di ingresso A e B, detta Y la variabile di uscita, si scrive la funzione logica:

$$\mathbf{Y= A \cdot B}$$

e si legge A and B.

Nella seguente figura si mostra la tabella della verità con le quattro possibili combinazioni tra A e B ed il simbolo logico relativo ad una porta AND a due ingressi. Nella colonna Y si sono posti i valori assunti dall'uscita Y che soddisfa la definizione della porta AND.



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

Negazione

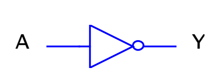
Si effettua su una sola variabile. L'uscita assume il valore logico opposto a quello applicato in ingresso.

Detta A la variabile di ingresso la negazione si scrive:

$$\mathbf{Y = \overline{A}}$$

e si legge ***A negato*** oppure ***A complementato.***

Nella seguente figura si mostra la tabella della verità con le due possibili combinazioni di A ed il simbolo logico relativo ad una porta NOT. Nella colonna Y si sono posti i valori assunti dall'uscita Y che soddisfa la definizione della porta NOT.



A	Y
0	1
1	0

Porte logiche derivate

Le porte esposte di seguito sono composte da due o piu' porte fondamentali, pero' per la loro importanza vengono rappresentate con un simbolo proprio.

Somma logica negata NOR

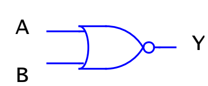
Si effettua su due o più variabili, l'uscita assume lo stato logico 0 se almeno una variabile di ingresso è allo stato logico 1. In tutti gli altri casi Y=1. Corrisponde ad una OR con in cascata una NOT

Per due variabili di ingresso A e B la funzione logica e':

$$\mathbf{Y= \overline{A+B}}$$

e si legge A nor B.

Nella seguente figura si mostra la tabella della verità con le quattro possibili combinazioni tra A e B ed il simbolo logico relativo ad una porta NOR a due ingressi. Nella colonna Y si sono posti i valori assunti dall'uscita Y che soddisfa la definizione della porta NOR.



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

Prodotto logico negato NAND

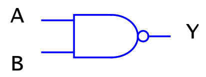
Si effettua su due o più variabili, l'uscita assume lo stato logico 0 se tutte le variabili di ingresso sono allo stato logico 1. In tutti gli altri casi Y=1. Corrisponde ad una AND con in cascata una NOT

La funzione logica si scrive:

$$\mathbf{Y = \overline{A \cdot B}}$$

e si legge A nand B.

Nella seguente figura si mostra la tabella della verità con le quattro possibili combinazioni tra A e B ed il simbolo logico relativo ad una porta NAND a due ingressi. Nella colonna Y si sono posti i valori assunti dall'uscita Y che soddisfa la definizione della porta NAND.

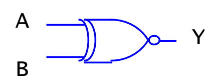


A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

OR esclusivo XOR

A differenza delle precedenti porte logiche, l'XOR opera solo su due ingressi. L'uscita vale [redacted]

Nella seguente figura si mostra la tabella della verità con le quattro possibili combinazioni tra A e B ed il simbolo logico relativo ad una porta XNOR. Nella colonna Y si sono posti i valori assunti dall'uscita Y che soddisfa la definizione della porta XNOR.



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

Quando si studieranno i modi per ricavare la funzione logica di un circuito digitale a partire dalla sua tabella di verita', si verifichera' che la funzione XOR corrisponde alla funzione:

$$\mathbf{Y = A\,B + \overline{A\,B}}$$

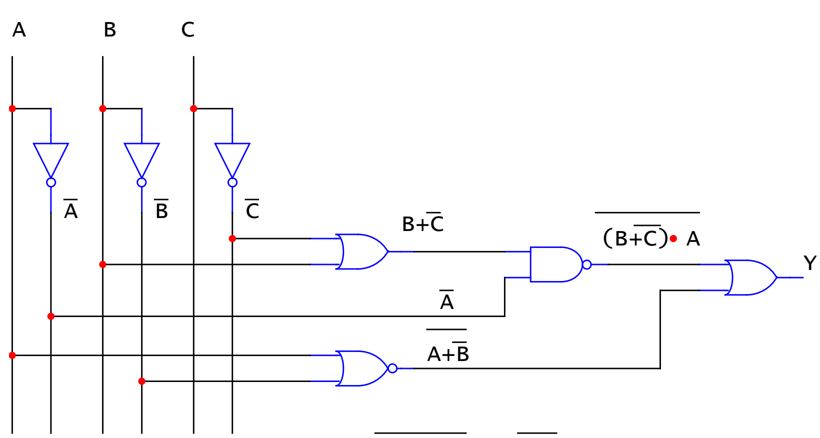
Circuiti digitali

Collegando opportunamente le porte logiche prima definite si potranno realizzare tutti i possibili circuiti digitali. Questi possono essere di due tipi:

- **Circuiti combinatori:** le uscite del circuito dipendono esclusivamente dalle entrate (ogni singola porta prima considerata e' un semplice circuito combinatorio);

- **Circuiti sequenziali:** le uscite del circuito dipendono sia dalle entrate che dallo stato interno del circuito, sono ad esempio circuiti sequenziali i latch, i flip flop, i registri, le memorie, ecc. che vedremo in seguito.

Un circuito digitale sara' rappresentato dal suo schema, dalla sua funzione logica e dalla sua tabella di verita', ognuno di queste tre definira' completamente il circuito nel senso che c'e' una corrispondenza fra schema, circuito e tabella di verita'.
Consideriamo ad es. il seguente schema composto con le porte logiche che conosciamo:



$$Y = \overline{(B+\overline{C})\bullet A} + \overline{A+\overline{B}}$$

analizzando il circuito andando da sinistra a destra possiamo facilmente ricavare la funzione logica come indicato.

Algebra di Boole

La funzione logica di un circuito, come abbiamo visto, e' espressa utilizzando i vari gli operatori logici fondamentali (somma, differenza, negazione) ed e' sottoposta alle regole di un algebra, diversa da quella che conosciamo, detta algebra di Boole.

Vi sono diverse proprietà dell'algebra Booleana che risultano utili nel manipolare le equazioni logiche al fine di semplificarla e quindi al fine di semplificare il circuito corrispondente, in particolare:

- **Proprietà di identità:**

  $A + 0 = A$

  $A \cdot 1 = A$

- **Proprietà di assorbimento:**

  $A + 1 = 1$

  $A \cdot 0 = 0$

- **Proprietà dell'inverso:**

  $A + \overline{A} = 1$

  $A \cdot \overline{A} = 0$

- **Proprietà della doppia negazione:**

  $\overline{\overline{A}} = A$

- **Proprietà dell' idempotenza:**

  $A + A = A$

  $A \cdot A = A$

- **Proprietà commutativa:**

  $A + B = B + A$

  $A \cdot B = B \cdot A$

- **Proprietà associativa:**

  $A + (B + C) = (A + B) + C$

  $A \cdot (B \cdot C) = (A \cdot B) \cdot C$

- **Proprietà distributiva:**

  $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$

  $A + (B \cdot C) = (A + B) \cdot (A + C)$

- **Teorema dell' assorbimento**

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

- **teoremi di De Morgan**:

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

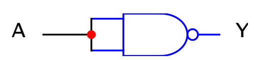
  $A + B = \overline{\overline{A} \cdot \overline{B}}$

Porte logiche universali NAND e NOR

È possibile verificare che con le sole porte NOR o con le sole porte NAND è possibile realizzare qualsiasi circuito digitale in quanto tutte le porte fondamentali (NOT, OR ed AND) possono essere realizzate con queste.

Di seguito e' riportato come si può realizzare la NOT, l'OR e l'AND con porte NAND tenendo conto delle proprieta' e dei teoremi dell' algebra di Boole.

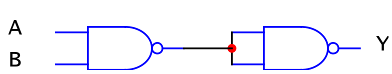
Collegando tra loro gli ingressi A e B della porta NAND come in Fig. 1a si ottiene la NOT:



$$\mathbf{Y = \overline{A\ A} = \overline{A}}$$

Fig.1a – NOT realizzata con porta NAND.

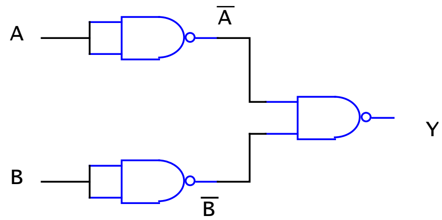
Per ottenere l'AND è sufficiente far seguire la porta NAND da una NOT realizzata come in Fig. 1b.



$$\mathbf{Y = \overline{\overline{A\ B}} = A\ B}$$

Fig. 1b – AND realizzata con porte NAND

Per ottenere l'OR, infine è sufficiente far precedere ciascun ingresso della NAND da una NOT come in Fig.1c. Infatti, per il noto teorema di **De Morgan**, la somma logica è il complemento del prodotto dei complementi.

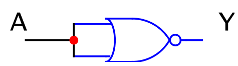


$$Y = \overline{\bar{A}\ \bar{B}} = A + B$$

Fig.1c - OR realizzata con porte NAND.

Di seguito invece e' riportato come si può realizzare la NOT, l'OR e l'AND con porte NOR.

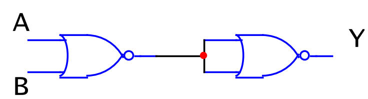
Collegando tra loro gli ingressi A e B della porta NOR come in Fig. 2a si ottiene la NOT:



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

Fig. 2a – NOT realizzata con porta NOR

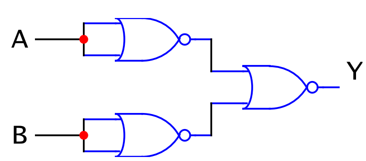
Per ottenere l' OR è sufficiente far seguire la porta NOR da un NOT realizzato come in Fig. 2b.



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

Fig. 2b – OR realizzata con porte NOR

Per ottenere l' AND, infine è sufficiente far precedere ciascun ingresso della NOR da una NOT come in Fig.2c. Infatti, per il noto teorema di **De Morgan**, il prodotto logico è il complemento della somma dei complementi.



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

Fig. 2c – AND realizzata con porte NOR

Progettazione di un circuito digitale combinatorio – 1° forma canonica.

Si vuole realizzare un circuito digitale combinatorio con tre ingressi A B C ed una uscita Y.

L'uscita deve valere 1 se almeno due ingressi sono uguali ad 1. Se, invece, vale 1 un solo ingresso o nessuno di essi l'uscita Y deve valere 0 (problema della maggioranza).

Risoluzione

Compiliamo la ***tabella della verità,*** in cui in ciascuna riga applichiamo le possibili combinazioni binarie tra gli ingressi A B C. Tali combinazioni sono: 2^3=8.

In corrispondenza di ciascuna di esse sarà possibile, grazie alla formulazione del problema, assegnare all'uscita Y il valore 0 o il valore 1.

Dalla tabella osserviamo che l'uscita Y assume il valore 1 in corrispondenza delle combinazioni ABC pari a 011, 101, 110 e 111.

Nell'ultima combinazioni tutti gli ingressi sono a 1 per cui, a maggior ragione, risulta Y=1.

Per le altre combinazioni l'uscita Y vale 0 come richiesto dal problema. Possiamo quindi ricavare la seguente tabella di verita':

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

Come si realizza il circuito? Sarà, innanzitutto, necessario ricavare la funzione logica dalla tabella della verità.

Il procedimento è il seguente: si considerano le combinazioni delle variabili di ingresso che rendono l'uscita Y uguale a 1. La combinazione di ingresso da considerare conterrà il prodotto tra A B C e ciascuna variabile sarà considerata in ***forma naturale*** o in ***forma negata*** a seconda se assume il valore 0 o 1. La prima combinazione che rende l'uscita uguale a 1 è: ABC=011, cioè A=0, B=1 e C=1. Allora A dovrà essere considerata in forma negata (complementata), mentre B e C in forma naturale (cioè senza negazione). I quattro termini che rendono la variabile di uscita Y uguale a 1 devono, poi, essere sommati logicamente tra di loro.

In definitiva si ottiene:

$$\mathbf{Y = \overline{A}\,B\,C + A\,\overline{B}\,C + A\,B\,\overline{C} + A\,B\,C}$$

l' espressione cosi' ottenuta rappresenta la funzione logica nella sua **1° FORMA CANONICA** ed ogni termine sommato si chiama **MINTERMINE**

Questa espressione si può semplificare algebricamente applicando le regole **dell'algebra di *Boole*** che solo parzialmente coincidono con quelle dell'algebra ordinaria.

In particolare, per la semplificazione che si intende effettuare, si applica la regola che aggiungendo quante volte si vuole un termine già esistente in una espressione, il valore dell'espressione non cambia. L'altra regola è che la somma logica tra una variabile e la stessa variabile negata vale 1.

Nel nostro caso aggiungiamo, nella espressione della funzione Y, il termine ABC per altre due volte. In tal caso il primo termine ha BC in comune con ABC, il secondo ha AC in comune con l'altro ABC aggiunto ed infine il terzo termine ha AB in comune con l'ultimo ABC aggiunto.

$$\mathbf{Y = \bar{A}\,B\,C + A\,\bar{B}\,C + A\,B\,\bar{C} + A\,B\,C + A\,B\,C + A\,B\,C}$$

Attraverso la messa in evidenza si ottiene:

$$\mathbf{Y = B\,C\,(A + \bar{A}) + A\,C\,(B + \bar{B}) + A\,B\,(C + \bar{C})}$$

considerando che $\mathbf{A + \bar{A} = 1}$ e così pure $\mathbf{B + \bar{B} = 1}$ e $\mathbf{C + \bar{C} = 1}$, l'espressione si riduce a:

$$\mathbf{Y = BC + AC + AB}$$

L'espressione così minimizzata è una somma di prodotti. Il circuito logico da realizzare sarà costituito da tre porte AND a due ingressi e da una porta OR finale a tre ingressi. In fig.3 si mostra il circuito risolutivo.

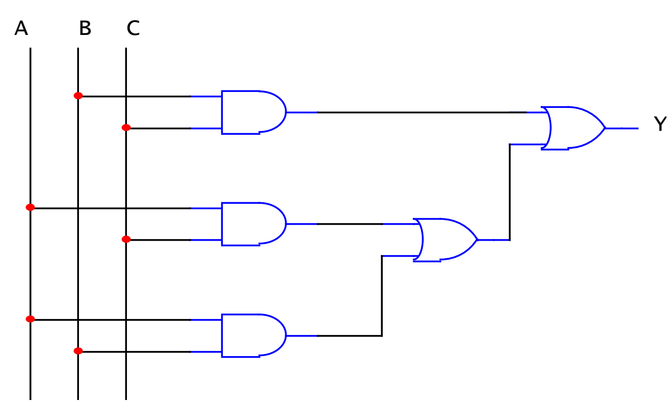


Fig.3 - Circuito risolutivo.

Per verificarne il funzionamento è sufficiente attribuire alle variabili di ingresso A B C ciascuna delle otto combinazioni mostrate nella precedente tabella ed eseguendo la somma logica tra i tre termini prodotto BC, AC e AB si puo' verificare che l' uscita corrisponde a quella della tabella.