

2. - FONDAMENTI DI LOGICA

geti

2.1 - GENERALITÀ

I PLC, come tutte le apparecchiature digitali, operano secondo una logica binaria, in cui esistono due soli stati: H (High) o L (Low), 1 o 0, Sì o No, ecc. **TRUE, FALSE**

Questo concetto viene applicato alle porte logiche che possono essere utili per prendere delle decisioni.



Fig. 2.1 - a) La luce in una stanza può essere accesa soltanto se l'interruttore è chiuso e (AND) se la lampada è posta correttamente nel suo attacco - b) Il risveglio di un individuo può essere provocato dalla suoneria di una sveglia o (OR) da un rumore esterno o (OR) dalla caduta del soffitto.

Una porta logica è costituita da uno o più ingressi e da un'unica uscita.

Esiste una parte della matematica che si occupa dello studio dei sistemi binari, chiamata algebra di Boole: si applica a tutti i sistemi che lavorano su variabili che possono assumere solo due valori.

2.2 - OPERATORI FONDAMENTALI: AND, OR, NOT

Ogni operazione di un'apparecchiatura digitale, anche la più complessa, è basata su tre operatori logici fondamentali: **AND, OR, NOT**. Ogni operatore logico ha un suo simbolo normalizzato, che viene usato per fare una rappresentazione grafica di un determinato problema logico; vengono utilizzati inoltre dei segnali applicati agli ingressi chiamati A, B, C, ecc. che possono assumere, come si è detto, solo due valori: valore 1 se è presente il segnale o se è verificato quel determinato evento, valore 0 se il segnale non è presente o se quell'evento non è verificato. Su tali variabili di ingresso agisce l'operatore logico generando un'uscita chiamata Q, che potrà avere solo valore 1 oppure valore 0.

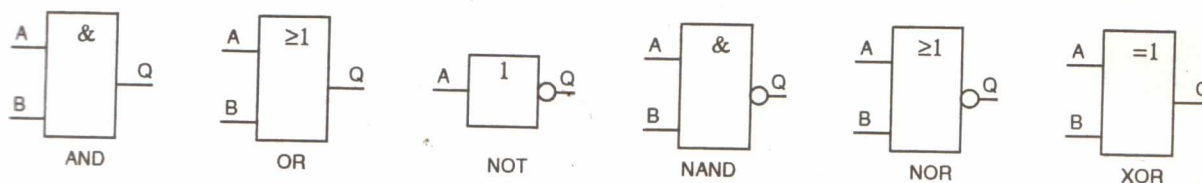
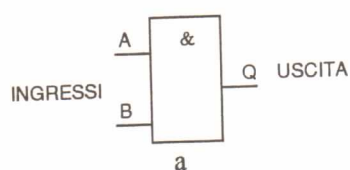


Fig. 2.2 - Segni grafici per la rappresentazione di porte logiche (norme CEI 3-26).

Nell'algebra di Boole i segnali in ingresso A, B, C, ecc. sono chiamati variabili booleane, con esse è possibile scrivere delle equazioni che rappresenteranno determinati problemi logici.

Alle variabili booleane vengono applicate degli operatori che provocano trasformazioni dei valori secondo particolari regole che sono definite nelle tabelle di verità: ogni operatore ha la sua tabella di verità.

L'operatore logico AND viene chiamato anche prodotto booleano o prodotto logico, si applica ad almeno due variabili ed è caratterizzato dalla seguente tabella di verità:



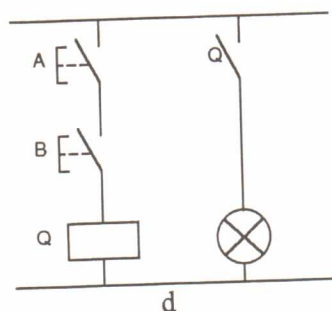
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

b

$$Q = A \text{ AND } B$$

$$Q = A * B$$

c



d

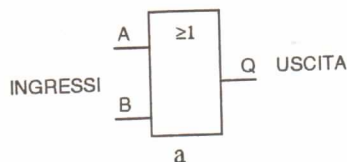
PULSANTE A	PULSANTE B	LAMPADA
APERTO	APERTO	SPENTA
APERTO	CHIUSO	SPENTA
CHIUSO	APERTO	SPENTA
CHIUSO	CHIUSO	ACCESA

e

Fig. 2.3 - Circuito logico AND - a) AND con due ingressi: segno grafico - b) Tabella della verità - c) Espressione algebrica - d) Schema equivalente a contatti - e) Tabella della verità.

Come si può notare dalla tabella, il prodotto logico Q di due o più variabili booleane vale 1 soltanto quando tutte le variabili A, B, C, ecc. valgono 1, viceversa vale 0.

L'operatore OR, chiamato anche somma booleana o somma logica, si applica ad almeno due variabili ed è caratterizzato dalla seguente tabella di verità:



a

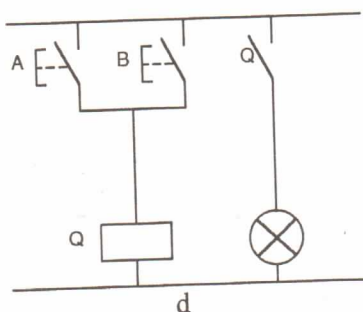
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

b

$$Q = A \text{ OR } B$$

$$Q = A + B$$

c



d

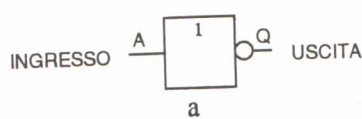
PULSANTE A	PULSANTE B	LAMPADA
APERTO	APERTO	SPENTA
APERTO	CHIUSO	ACCESA
CHIUSO	APERTO	ACCESA
CHIUSO	CHIUSO	ACCESA

e

Fig. 2.4 - Circuito logico OR - a) OR con due ingressi: segno grafico - b) Tabella della verità - c) Espressione algebrica - d) Schema equivalente a contatti - e) Tabella della verità.

Come si può notare dalla tabella, la somma logica Q di due o più variabili vale 0 solo se tutte le variabili valgono 0 mentre vale 1 se almeno una delle variabili vale 1.

L'operatore NOT si applica invece ad una sola variabile booleana, trasformando la variabile secondo la seguente tabella di verità:

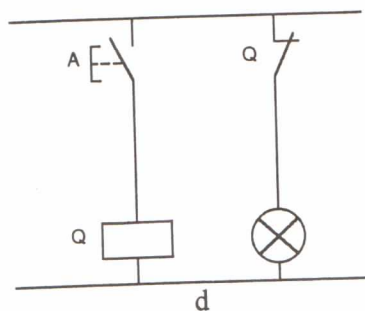


A	Q
0	1
1	0

b

$Q = \text{NOT } A$
$A = \text{NOT } Q$

c



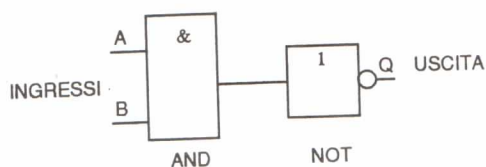
PULSANTE A	LAMPADA
APERTO	ACCESA
CHIUSO	SPENTA

e

Fig. 2.5 - Circuito logico NOT - a) Segno grafico - b) Tabella della verità - c) Espressione algebrica - d) Schema equivalente a contatti - e) Tabella della verità.

In pratica, applicando l'operatore NOT ad una variabile, tale variabile cambia valore: se valeva 1 dopo la trasformazione varrà 0 e viceversa.

Combinando opportunamente gli operatori AND, OR con l'operatore NOT si possono ottenere gli operatori NAND, NOR e XOR che sono caratterizzati dalle seguenti tabelle di verità:

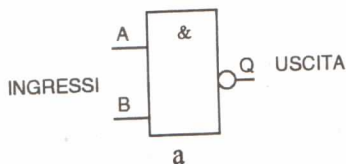


A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

b

$Q = A \text{ NAND } B$

c



PULSANTE A	PULSANTE B	LAMPADA
APERTO	APERTO	ACCESA
APERTO	CHIUSO	ACCESA
CHIUSO	APERTO	ACCESA
CHIUSO	CHIUSO	SPENTA

e

Fig. 2.6 - Circuito logico NAND - a) NAND con due ingressi segno grafico - b) Tabella della verità - c) Espressione algebrica - d) Schema equivalente a contatti - e) Tabella della verità.

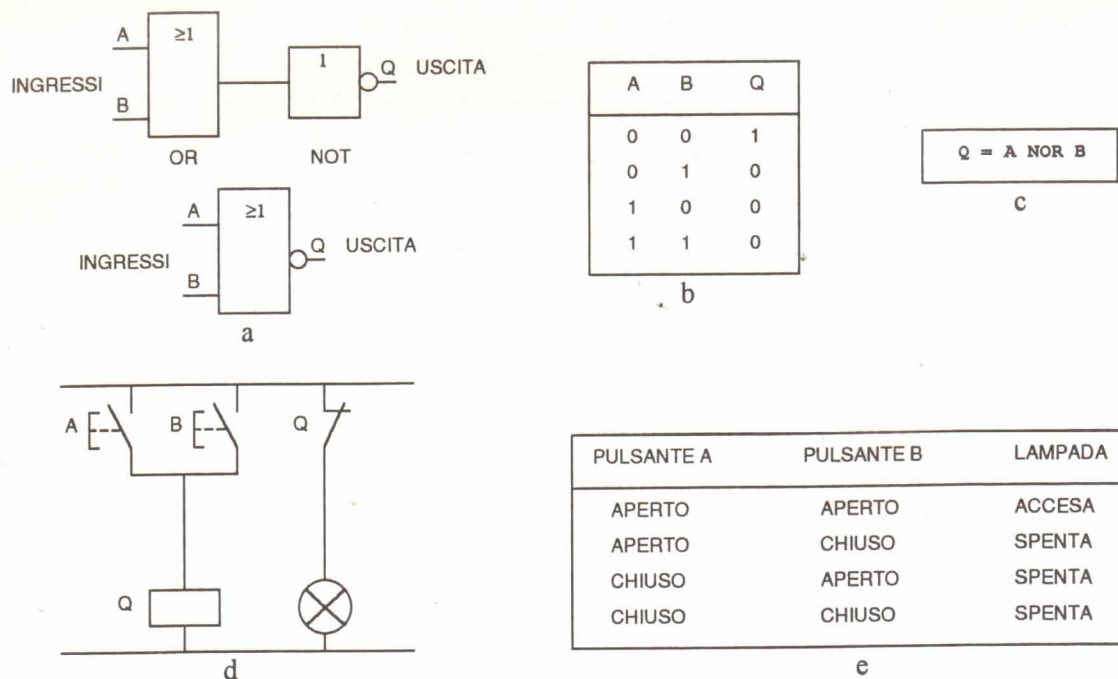


Fig. 2.7 - Circuito logico NOR - a) NOR con due ingressi segno grafico - b) Tabella della verità - c) Espressione algebrica - d) Schema equivalente a contatti - e) Tabella della verità.

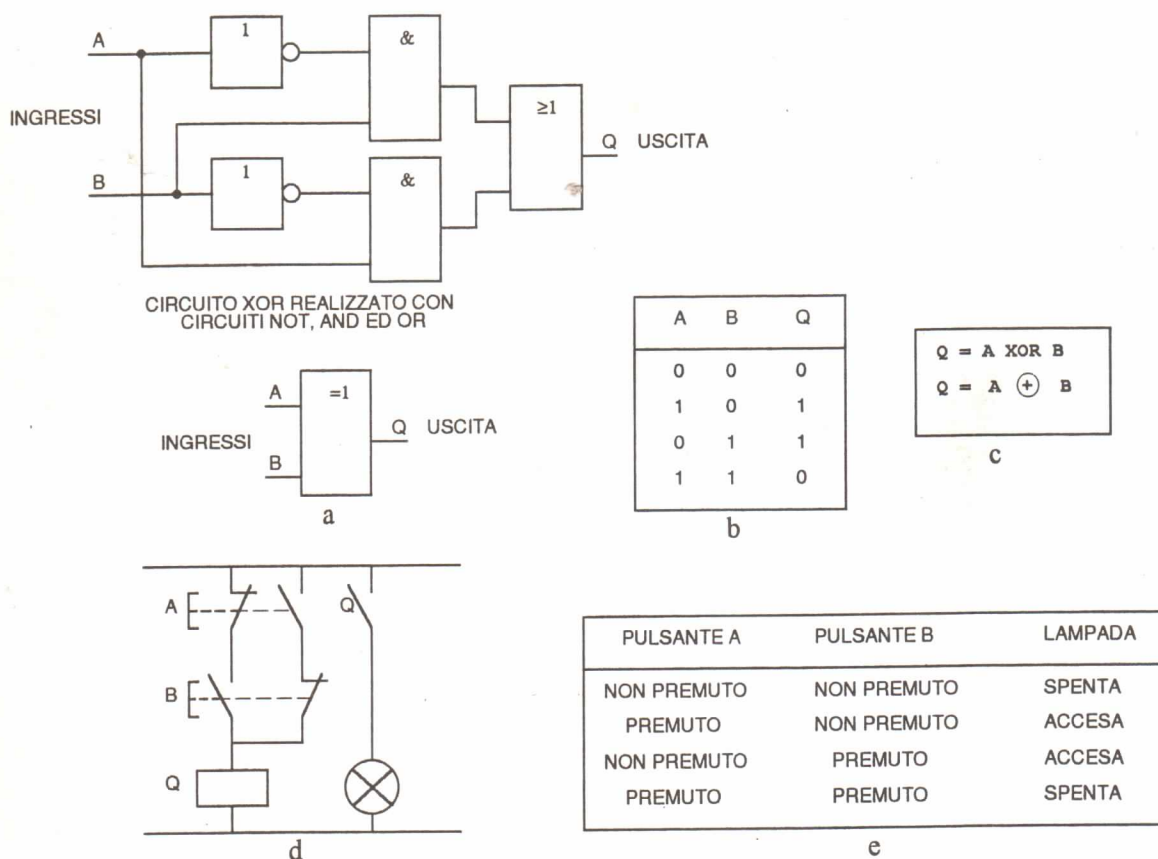


Fig. 2.8 - Circuito logico XOR - a) OR esclusivo o XOR segno grafico - b) Tabella della verità - c) Espressione algebrica - d) Schema equivalente a contatti - e) Tabella della verità.

Come si è visto, si utilizza il segno di moltiplicazione per indicare l'operazione di AND, il segno di addizione per indicare una operazione di OR, una barra posta sopra alle lettere A, B, C, ecc. Q per indicare l'operazione di NOT.

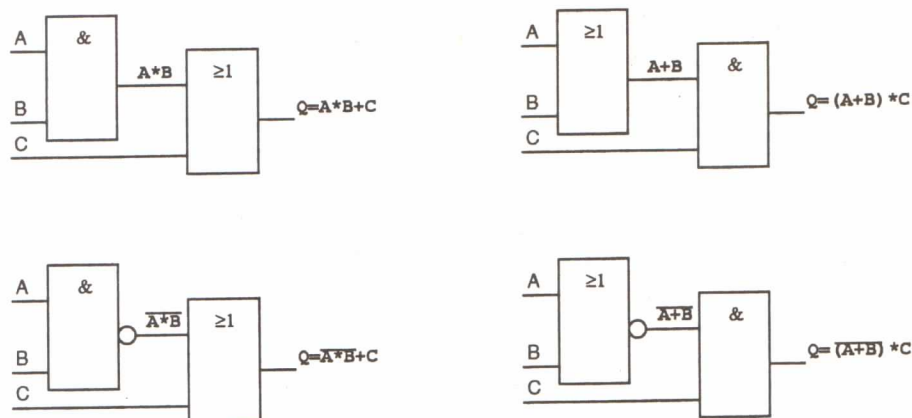


Fig. 2.9 - Esempi di combinazioni di porte logiche.

2.3 - MEMORIE

Una memoria elettronica è un circuito bistabile (flip-flop), cioè con due stati stabili, che può essere commutato da uno stato all'altro applicando dei segnali di comando.

Il tipo più semplice di memoria è costituito da due porte NOR a connessioni incrociate.

L'uscita di questa memoria viene prelevata dal terminale Q e il suo complemento logico è disponibile sul terminale $\bar{Q}$.

Il circuito funziona nel modo seguente: l'applicazione del livello logico 1 al filo S (il Set), quando il filo R è al livello logico zero, provoca il passaggio a 1 dell'uscita Q.

Riportando a zero il filo S, l'uscita della memoria viene mantenuta nella condizione in cui si trova, per mezzo del collegamento di retroazione presente tra le porte logiche NOR che costituiscono la memoria.

In questo modo viene "memorizzata" l'ultima operazione.

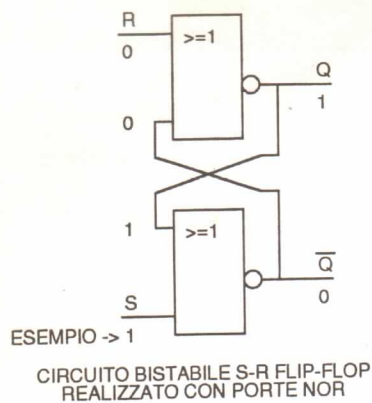
L'uscita Q può essere riportata a zero applicando il livello logico 1 al filo R (ossia al Reset).

Dopo quest'ultima operazione, l'uscita Q rimane a zero fino a che non viene applicato nuovamente un segnale alto all'ingresso S.

Per il suo funzionamento questa memoria è chiamata memoria set-reset, oppure S-R flip-flop.

All'atto della messa in tensione del circuito, la memoria richiede sempre un segnale che la piloti in posizione di partenza.

In nessun caso ai terminali S e R deve essere applicato contemporaneamente un segnale allo stato logico 1, pena l'indeterminatezza dell'uscita.



STATO INIZIALE USCITE		SEGNALE IN INGRESSO		STATO FINALE USCITE	
Q	$\bar{Q}$	S (A)	R (B)	Q	$\bar{Q}$
0	1	0	0	0	1
1	0	0	0	1	0
0	1	0	1	0	1
1	0	0	1	0	1
0	1	1	0	1	0
1	0	1	0	1	0
0	1	1	1	?	?
1	0	1	1	?	?

b)

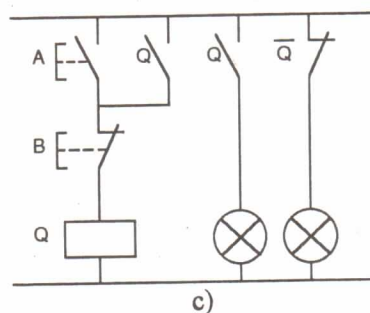
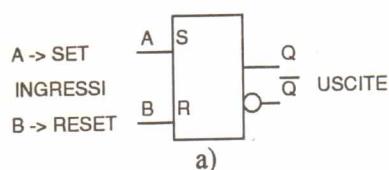


Fig. 2.10 - Circuito logico bistabile S-R flip-flop - a) Bistabile S-R flip-flop segno grafico - b) Tabella della verità - c) Schema equivalente a contatti.

2.4 - INTRODUZIONE ALLA MATEMATICA DEI CONTATTI

Come noto, i contatti di pulsanti, finecorsa, ecc. possono assumere solo due posizioni, cioè di **aperto** (non passa corrente) e **chiuso** (passa corrente); negli schemi con contatti, definirne il funzionamento significa determinare la loro combinazione affinché venga fornita corrente all'utilizzatore (es. relè).

Il contatto, come si è visto, è un organo operativo dei circuiti elettrici, capace di lasciar passare o di interrompere la corrente; può essere comandato manualmente (es. interruttore o pulsante) oppure con un comando a distanza (es. relè, contattore, ecc.).

Anche negli impianti pneumatici o meccanici è possibile trovare organi che funzionano come i contatti elettrici (es. valvole pneumatiche ad azionamento automatico o manuale, leve, ecc.); si ricorda infine che anche i transistor, i diodi, gli SCR e i TRIAC funzionanti in regime di commutazione si comportano nel medesimo modo; le regole precedentemente citate e quelle che saranno esposte di seguito valgono quindi anche per queste apparecchiature.

2.5 - REGOLE GENERALI

Un contatto si può identificare con una lettera maiuscola (es. A), che indica quale è l'operatore che lo comanda (bobina o comando manuale).

Ad un unico comando dell'operatore, tutti i contatti di uguale nome commutano contemporaneamente. I contatti a riposo possono essere normalmente aperti (NO) o normalmente chiusi (NC).

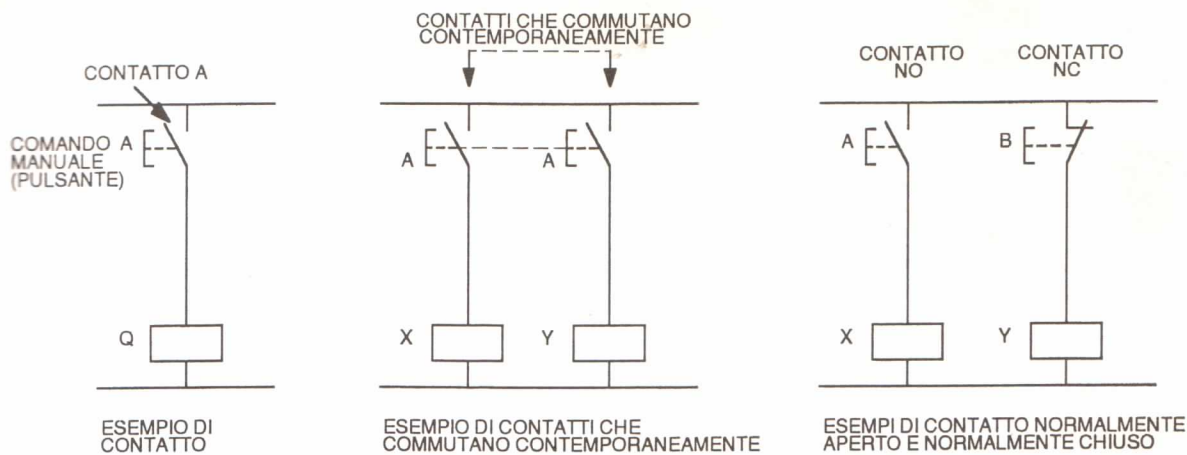


Fig. 2.11 - Esempi di contatti.

Quando si devono indicare, in un'equazione, i contatti NC, è convenzione indicarli con un trattino sopra la lettera; il contatto rappresentato nella figura che segue viene indicato nelle equazioni booleane con $\bar{A}$ (si legge A negato), nella rappresentazione grafica non si segna la negazione sul simbolo, in quanto la forma grafica del contatto indica già il suo stato di riposo.

Si dicono negati o complementari i contatti NC, perchè la funzione da loro risolta è esattamente "Topposto" del contatto NO assunto come riferimento.

Secondo la matematica dei contatti, si indica convenzionalmente con "livello logico 1" il segnale (elettrico, pneumatico, meccanico) che fa cambiare di stato un contatto e con "livello logico 0" la mancanza di segnale oppure il segnale che non fa commutare il contatto.

Per esempio il segnale logico 1 si identifica con il movimento del dito che preme il pulsante B; per il circuito della bobina (vedi figura), il segnale logico 1 è dato dal polo positivo dell'alimentazione, che gli viene trasmesso dalla chiusura del pulsante; l'eccitazione della bobina Q fa cambiare di stato i suoi contatti; il contatto NO, che a riposo trasferisce in X il livello 0, si chiude e trasmette il livello 1, mentre il contatto Q negato, che è NC e che a riposo trasmette in Y il livello 1, si apre e trasmette ora il livello 0.

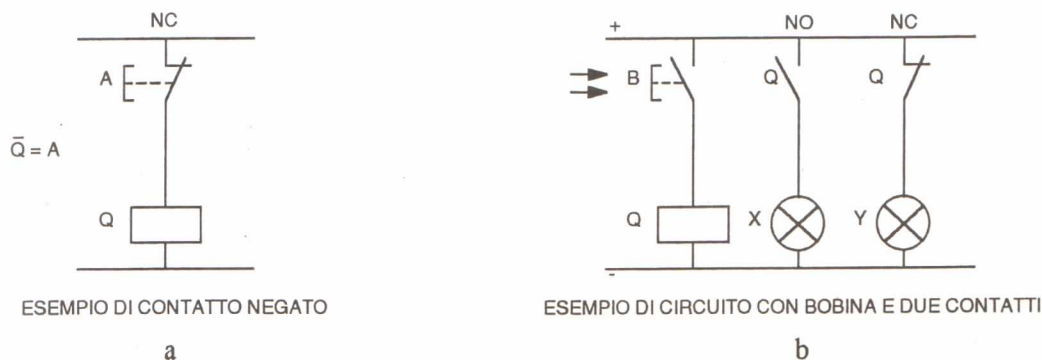


Fig. 2.12 - a) Esempio di contatto negato - b) Esempio di circuito con bobina e due contatti.

In definitiva possiamo dire:

- 1) applicare il segnale 1 ad un contatto NO vuol dire chiuderlo e far uscire il segnale 1, analogamente applicando il segnale 0 uscirà 0;
- 2) applicare il segnale 1 ad un contatto NC-vuol dire aprirlo e far uscire il segnale 0; viceversa dando ad un circuito il segnale 0, si ottiene in uscita il segnale 1: questa operazione è detta di complementazione;
- 3) i simboli 0 e 1 non sono numeri in senso aritmetico, ma indicano lo stato logico dell'elemento operatore e/o dell'uscita.

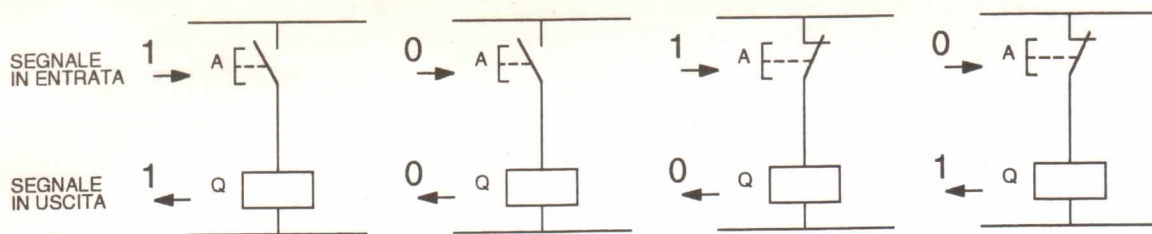


Fig. 2.13 - Esempi di segnali in entrata e in uscita in un circuito elettrico.

2.6 - PROPRIETÀ E TEOREMI DELL'ALGEBRA BOOLEANA

Come si è visto precedentemente, due o più contatti in serie eseguono un'operazione detta prodotto logico (AND), analogamente due o più contatti in parallelo eseguono una operazione detta somma logica (OR), infine l'operazione di complementazione od inversione logica (NOT) corrisponde alla trasformazione di contatti NO in contatti NC e viceversa.

Esistono inoltre nell'algebra di Boole delle proprietà che consentono di semplificare le espressioni booleane: proprietà commutativa, associativa, distributiva.

Analizzando gli schemi a contatti del tipo serie/parallelo, dopo aver determinato la loro equazione booleana, è interessante osservare che in molti casi è possibile scrivere delle espressioni e quindi degli schemi semplificati, che però sono del tutto equivalenti a quelli di partenza.

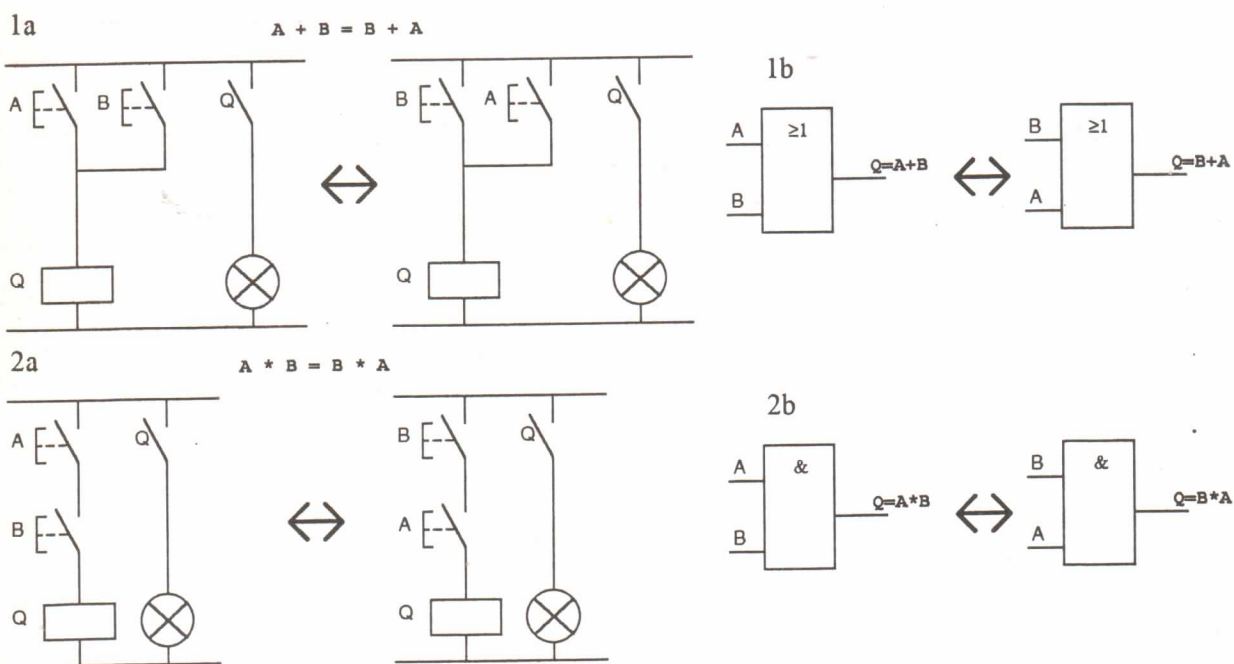


Fig. 2.14 - Proprietà commutativa: 1a-2a) schemi equivalenti a contatti - 1b-2b) schemi logici.

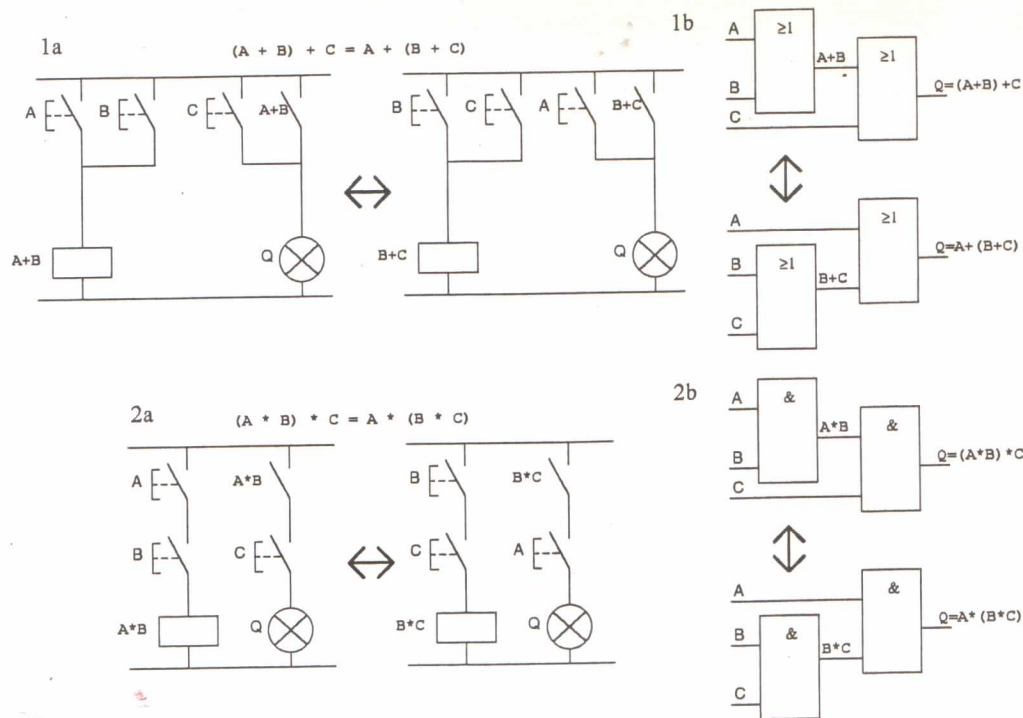


Fig. 2.15 - Proprietà associativa: 1a-2a) schemi equivalenti a contatti - 1b-2b) schemi logici.

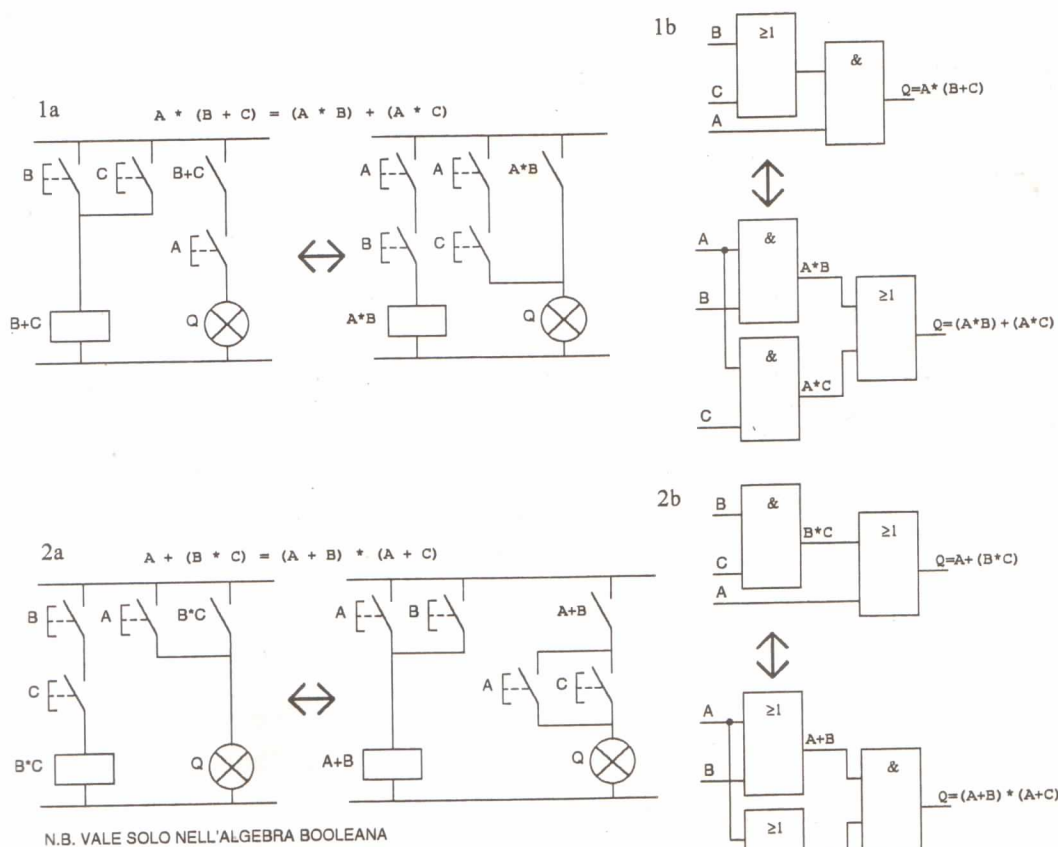


Fig. 2.16 - Proprietà distributiva: 1a-2a) schemi equivalenti a contatti - 1b-2b) schemi logici.

Oltre le proprietà viste precedentemente, valgono i seguenti teoremi.

NOME

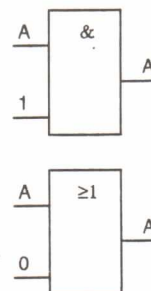
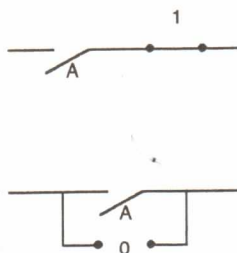
EQUAZIONE

SCHEMA ELETTRICO

SCHEMA LOGICO

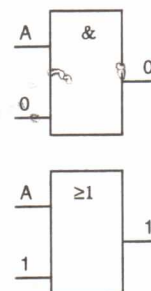
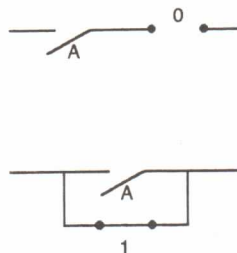
TEOREMA DI IDENTITÀ

$$\begin{cases} A \cdot 1 = A \\ A + 0 = A \end{cases}$$



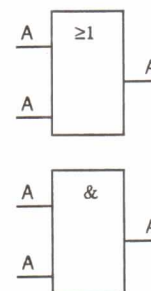
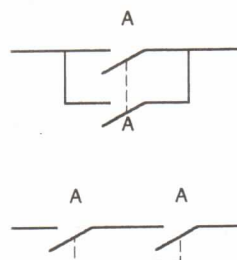
TEOREMA DI ANNULLAMENTO

$$\begin{cases} A \cdot 0 = 0 \\ A + 1 = 1 \end{cases}$$



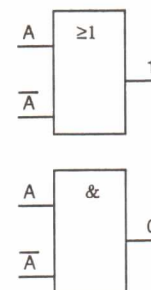
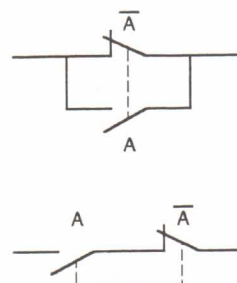
TEOREMA DI IDEMPOTENZA

$$\begin{cases} A + A = A \\ A \cdot A = A \end{cases}$$



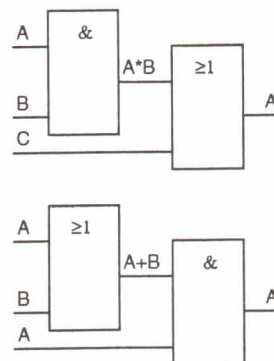
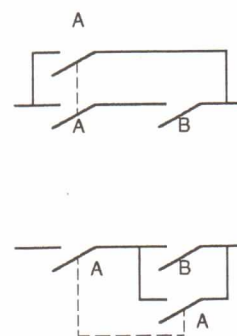
TEOREMA DEI COMPLEMENTI

$$\begin{cases} A + \bar{A} = 1 \\ A \cdot \bar{A} = 0 \end{cases}$$



TEOREMA DELL'ASSORBIMENTO

$$\begin{cases} A + A \cdot B = A \\ A \cdot (A + B) = A \end{cases}$$



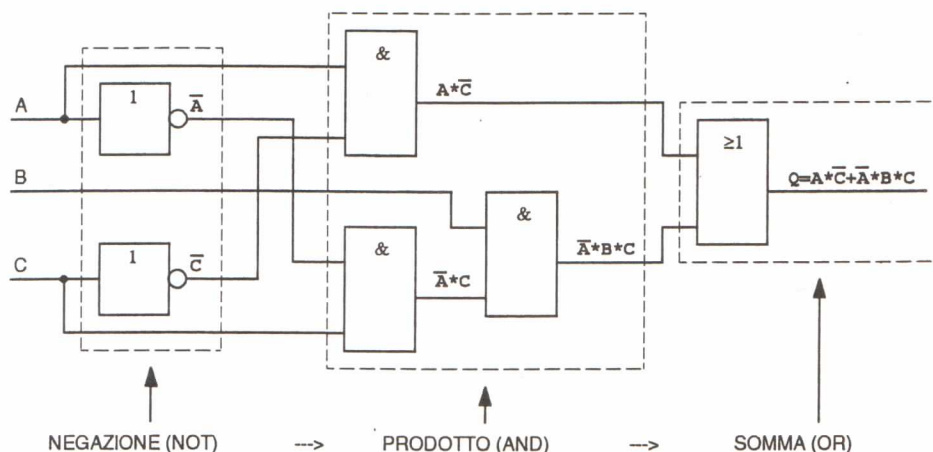
2.7 - TRASFORMAZIONE DI ESPRESSIONE BOOLEANA IN CIRCUITO LOGICO E VICEVERSA

Dovendo trasformare un'espressione booleana in circuito logico, si deve rispettare il seguente ordine:

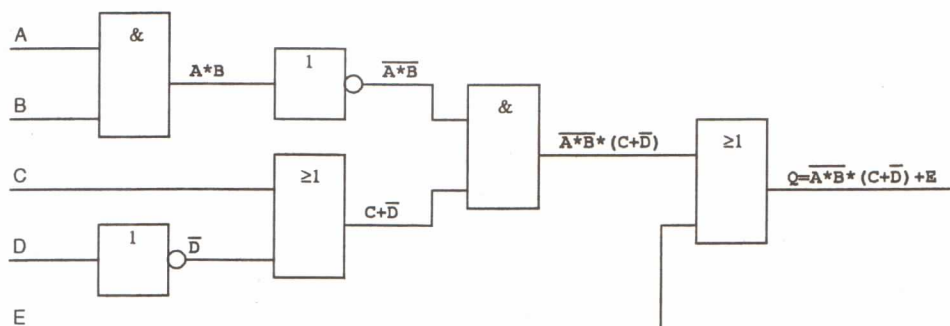
NEGAZIONE (NOT) --> PRODOTTO (AND) --> SOMMA (OR)

Se nell'espressione vi sono parentesi o trattini di complementazione estesi su più variabili, si deve risolvere prima la parte di espressione tra parentesi o sottostante il trattino, seguendo comunque l'ordine precedentemente citato.

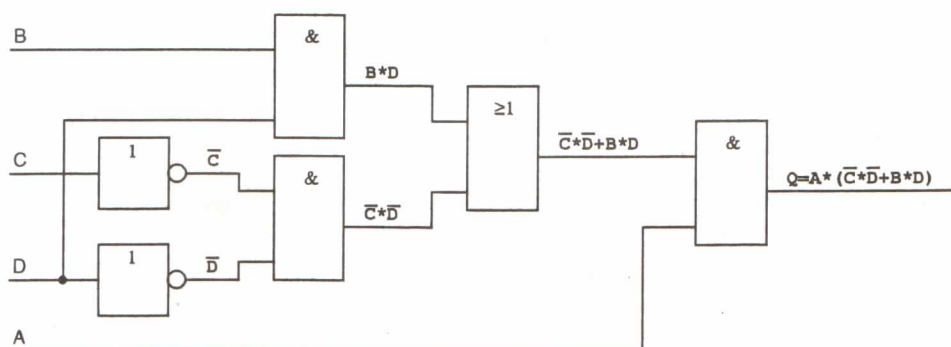
Esempio 1: $Q = A * \bar{C} + \bar{A} * B * C$



Esempio 2: $Q = \overline{A * B} * (C + \bar{D}) + E$



Esempio 3: $Q = A * (\bar{C} * \bar{D} + B * D)$



Qualora si debba trasformare una funzione booleana in un circuito logico, si deve suddividere l'espressione secondo l'ordine inverso a quello con cui vengono effettuate le operazioni.

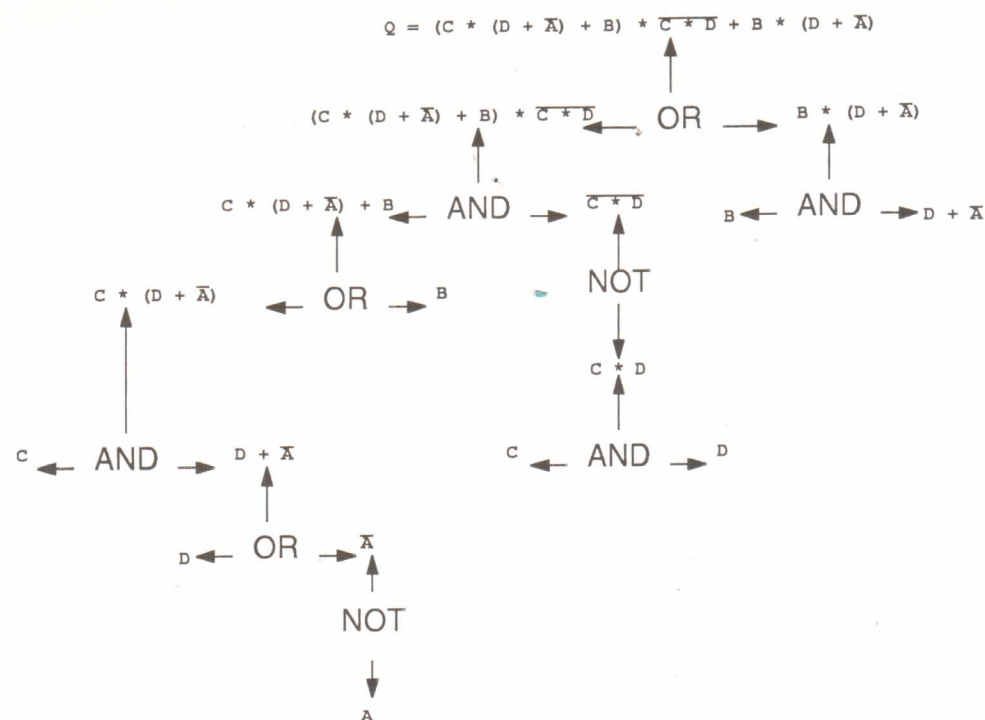


Fig. 2.17 - Scomposizione della funzione booleana.

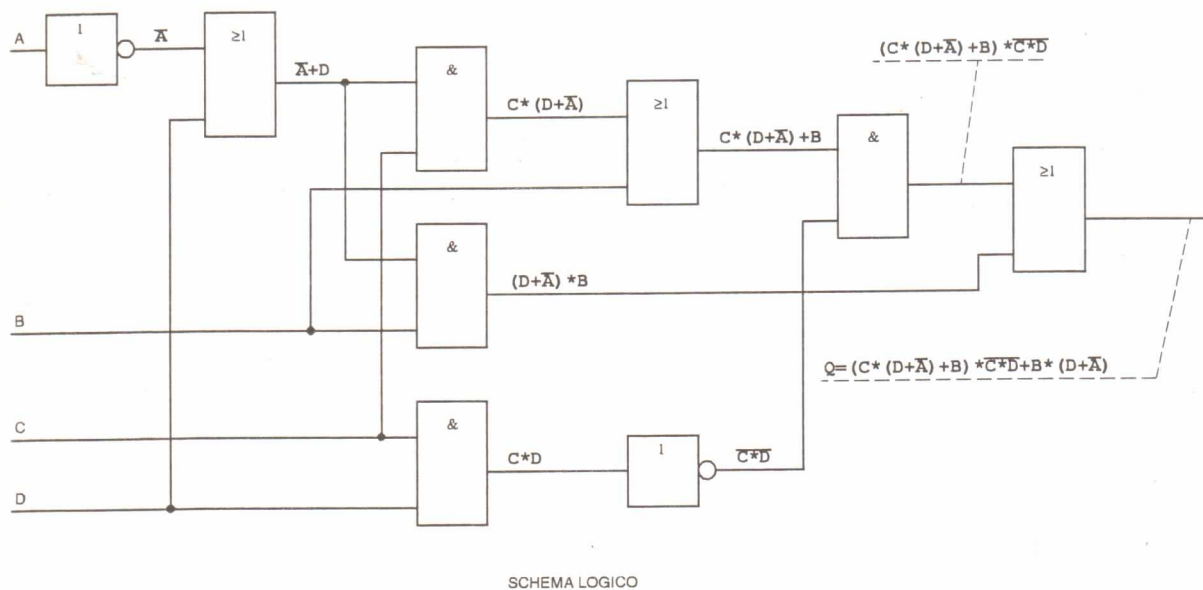


Fig. 2.18 - Schema logico relativo all'esempio trattato.

- LEGAME TRA SCHEMI A CONTATTI E EQUAZIONI LOGICHE E RELATIVA SEMPLIFICAZIONE

Alcuni schemi a contatti sono suscettibili di una semplificazione, che si può effettuare mediante l'aiuto dell'algebra binaria e delle sue proprietà.

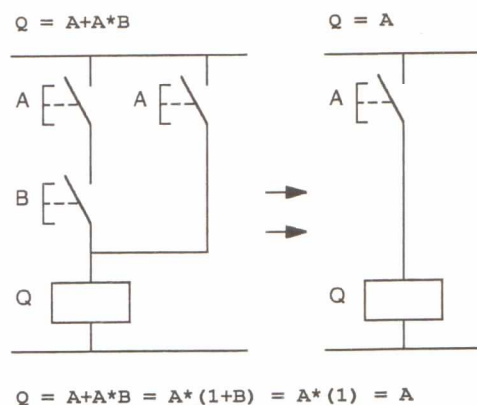


Fig. 2.19 - Esempio di semplificazione.

Nello schema della figura precedente, l'uscita Q dipende solo da A, perchè anche se B fosse chiuso, bisognerebbe che anche A lo fosse per ottenere l'uscita Q allo stato logico 1.

Volendo semplificare con l'ausilio dell'algebra di Boole, si può procedere nel seguente modo:

$$Q = A + A \cdot B = A \cdot (1 + B) = A \cdot (1) = A$$

Anche nello schema seguente, l'uscita Q dipende solo da A, infatti semplificando si ottiene:

$$Q = A \cdot (A + B) = A \cdot A + A \cdot B = A + A \cdot B = A \cdot (1 + B) = A \cdot (1) \text{ quindi } Q = A$$

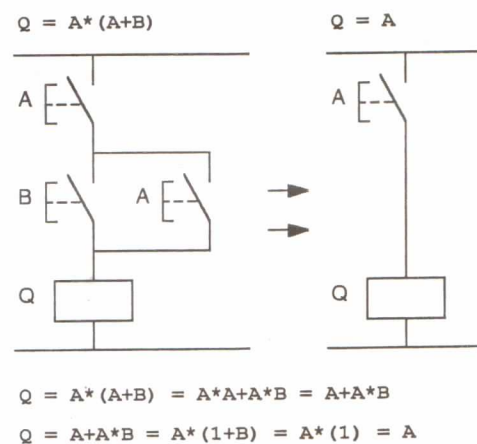
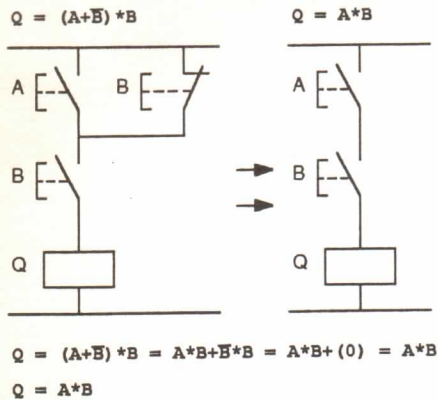


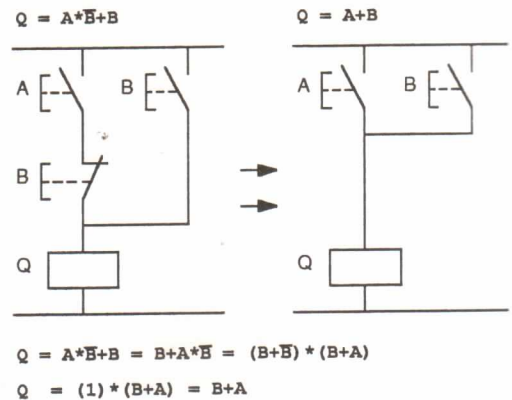
Fig. 2.20 - Esempio di semplificazione.

Esempi di semplificazione:

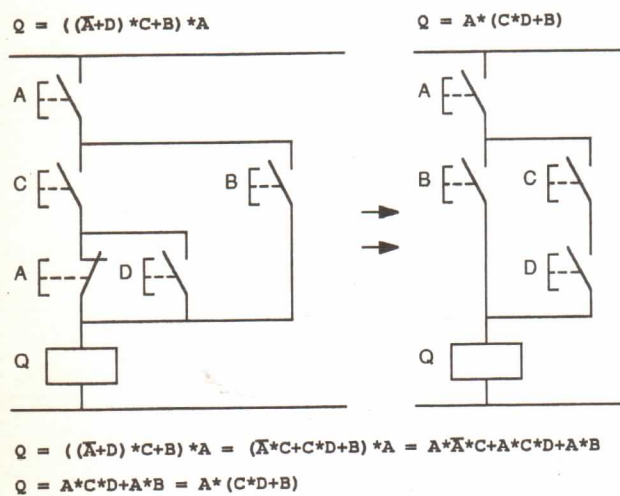
Esempio 1:



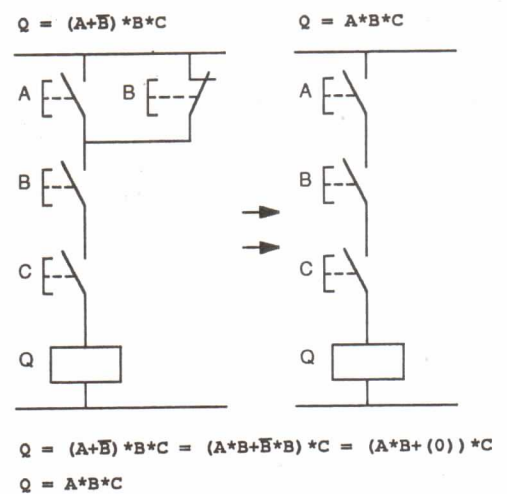
Esempio 2:



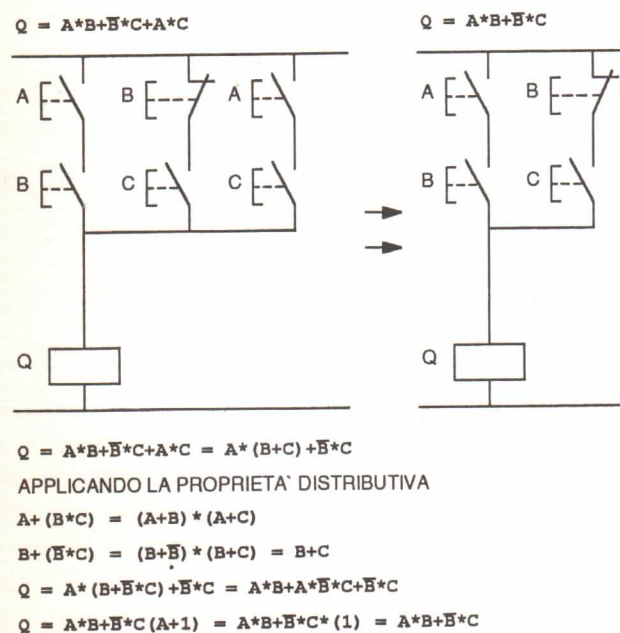
Esempio 3:



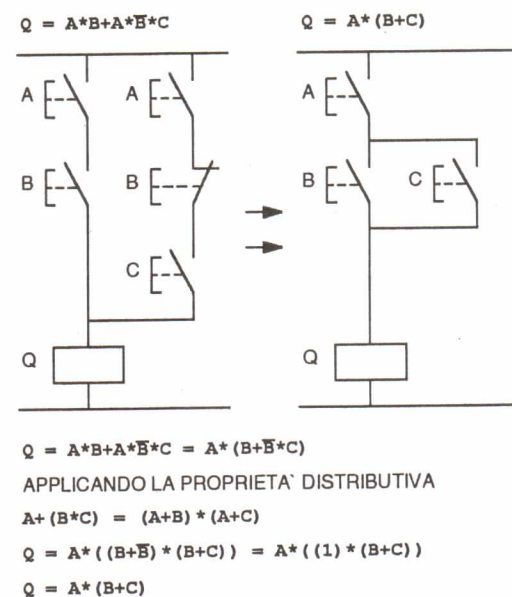
Esempio 4:



Esempio 5:



Esempio 6:



2.9 - TEOREMI DI DE MORGAN

Si può presentare, durante il progetto, il montaggio o la modifica di un impianto con una logica a contatti, la necessità di collegare dei contatti NC oppure NO, in serie o in parallelo, e di disporre solo dei loro complementari.

Può essere utile usare i teoremi di DE MORGAN per trasformare o semplificare le espressioni booleane: $\overline{A * B} = \overline{A} + \overline{B}$ e $\overline{A + B} = \overline{A} * \overline{B}$.

Analizzando le espressioni sopra indicate, si rileva che esse contengono in sostanza la trasformazione di un OR in AND e viceversa.

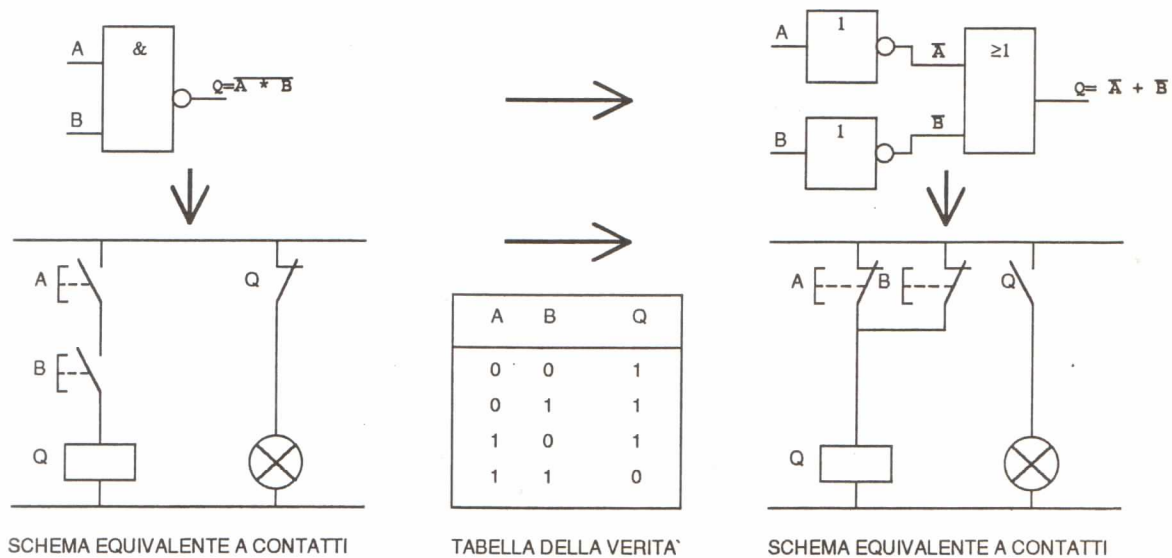


Fig. 2.21 - Teorema di DE MORGAN: $\overline{A * B} = \overline{A} + \overline{B}$.

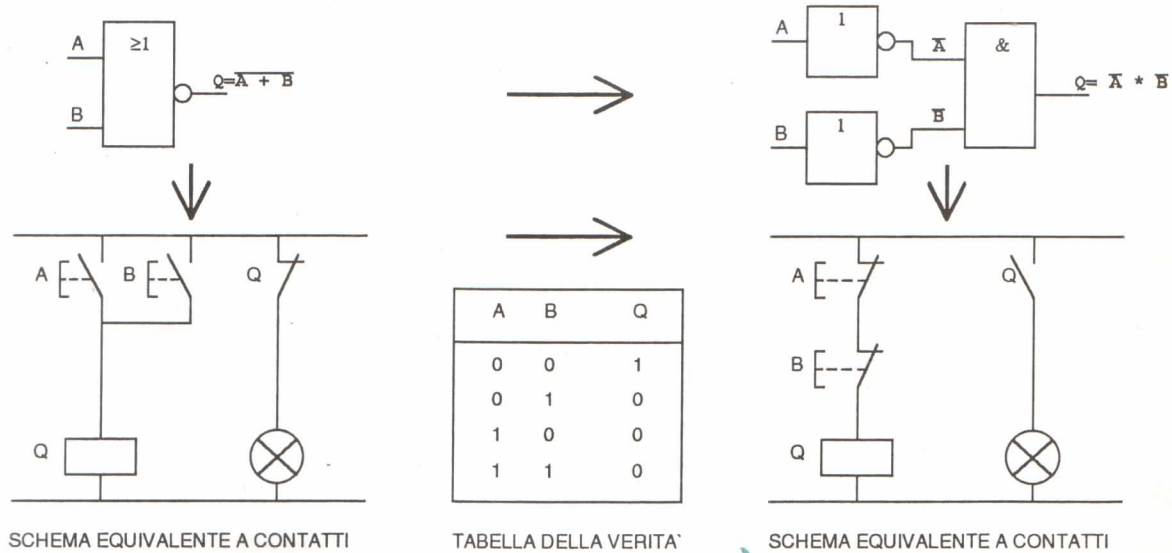


Fig. 2.22 - Teorema di DE MORGAN: $\overline{A + B} = \overline{A} * \overline{B}$.

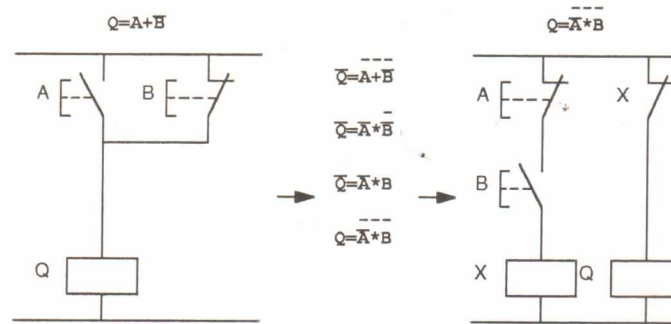
In un'espressione algebrica generica, si ottiene il suo complemento scrivendo negate le variabili che compongono la formula e cambiando il prodotto in somma e la somma in prodotto.

Ad esempio:

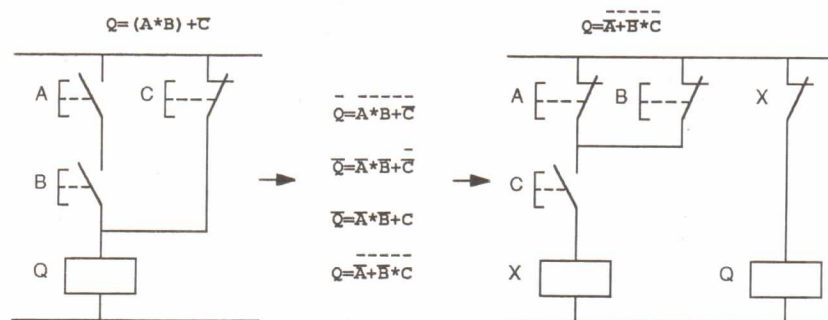
- 1) $Q = A * B$ il suo complemento, cioè $\overline{A * B}$ si ottiene con $\overline{A} + \overline{B}$
- 2) $Q = A + B$ il suo complemento, cioè $\overline{A + B}$ si ottiene con $\overline{A} * \overline{B}$

Esempi di applicazione:

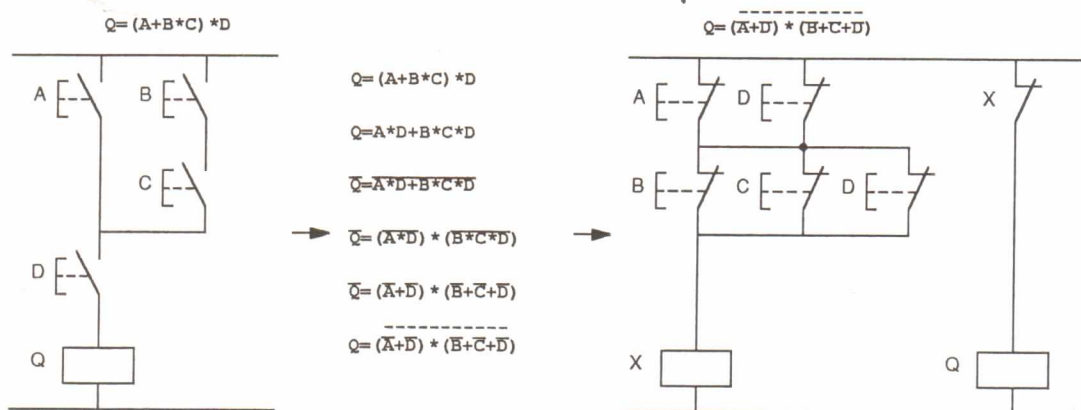
Esempio 1:



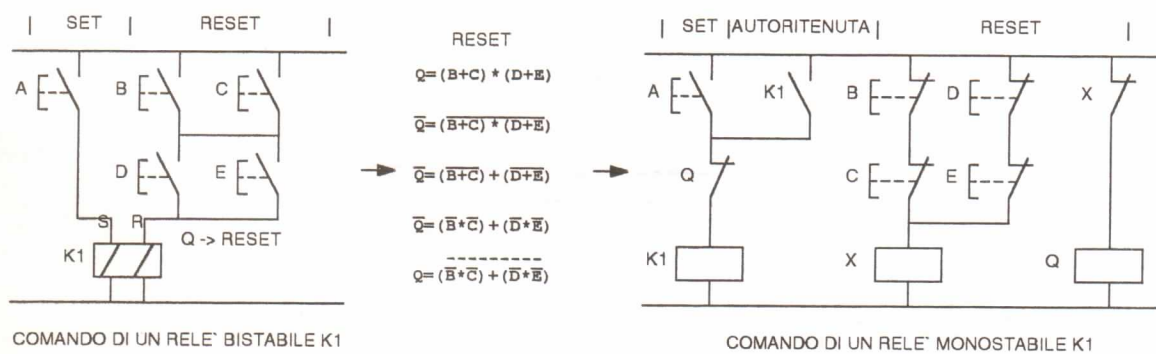
Esempio 2:



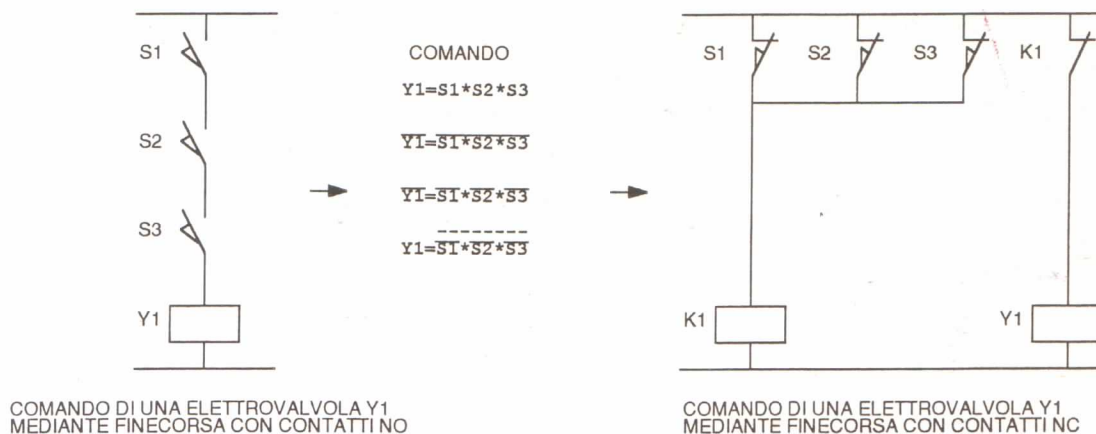
Esempio 3:



Esempio 4: trasformazione di un comando per relè bistabile in uno per il comando di un relè monostabile avente le medesime caratteristiche di funzionamento.



Esempio 5: comando di una elettrovalvola (Y1) che deve avvenire solo quando tutti i finecorsa (S1, S2, S3) sono azionati.



In pratica ciò significa che ogni circuito avente determinate caratteristiche può essere trasformato in un altro circuito con caratteristiche inverse, avente però un funzionamento equivalente al precedente: i contatti da normalmente aperti diventano normalmente chiusi e il collegamento dei contatti da serie diventa parallelo e viceversa.

Come si è visto negli esempi precedenti, una funzione booleana può essere definita mediante la sua espressione algebrica, ma può essere identificata anche dalla sua tabella di verità, la quale elenca in pratica tutte le possibili combinazioni che le variabili indipendenti (A, B, C, ecc.) possono avere ed il relativo risultato Q.

A	B	$\bar{A}$	$\bar{B}$	$\bar{A} \cdot \bar{B}$	$A + B$ OR	$\bar{A} + \bar{B}$ NOR	$A \cdot B$ AND	$\bar{A} \cdot \bar{B}$ NAND	$\bar{A} + B$	$A + \bar{B}$
0	0	1	1	0	0	1	0	1	1	1
0	1	1	0	1	1	0	0	1	1	0
1	0	0	1	0	1	0	0	1	1	0
1	1	0	0	0	1	0	1	0	0	0

Tab. 2.1 - Tabella della verità per due variabili A, B.

Come è noto, le variabili indipendenti possono avere solo due valori diversi e cioè 0 e 1, perciò le possibili combinazioni sono 2^n , dove n esprime il numero delle variabili indipendenti.

Per esempio se la funzione booleana è $Q = A \cdot B + B$, siccome $n = 2$, si avranno $2^2 = 4$ combinazioni da cui 4 ingressi che determinano una tabella composta di quattro righe che esaurisce tutte le possibili combinazioni.

Analizzando la tabella di verità è possibile semplificare una funzione booleana riuscendo ad ottenere un'altra funzione identica alla precedente, ma con un numero di operatori logici minore.

Questo in pratica si traduce in un numero minore di contatti se la funzione è realizzata con apparecchiature elettromeccaniche (relè), dispositivi elettronici se la funzione è realizzata con un circuito elettronico non programmabile (transistor o circuiti integrati), un programma di dimensioni più ridotte se la funzione è realizzata da un'apparecchiatura programmabile (PLC o computer).

A	B	C	$\bar{A}$	$\bar{B}$	$\bar{C}$	$\bar{A} \cdot B \cdot C$	$\bar{A} \cdot B \cdot \bar{C}$	$A \cdot \bar{B} \cdot C$	$A \cdot \bar{B} \cdot \bar{C}$	$B \cdot C$	$\bar{A} \cdot \bar{B} \cdot C$	$\bar{A} \cdot B \cdot \bar{C}$
0	0	0	1	1	1	0	1	0	1	0	1	1
0	1	0	1	0	1	0	1	0	1	1	0	0
1	0	0	0	1	1	0	0	1	1	0	0	0
1	1	0	0	0	1	0	1	0	1	1	0	0
1	1	1	0	0	0	0	1	0	1	1	0	0
1	0	1	0	1	0	0	1	0	1	1	0	0
0	1	1	1	0	0	1	1	0	0	1	0	0
0	0	1	1	1	0	0	1	0	1	1	0	0

Tab 2.2 - Tabella della verità per tre variabili A, B, C.

2.10 - LEGAME TRA LOGICA, CIRCUITI ELETTRONICI DIGITALI, SCHEMI ELETTRICI, PROGRAMMAZIONE DEI PLC

Come si è visto negli esempi precedenti, si può associare ad una espressione logica uno schema elettrico, il quale rappresenta un modo comodo (soprattutto per i tecnici con una preparazione elettromeccanica) di rappresentare un automatismo.

In questi schemi compaiono una serie di dispositivi come pulsanti, finecorsa, relè, lampade, ecc., i quali possono assumere come si è visto solo due stati: aperto o chiuso per i contatti, eccitato o diseccitato per relè, contattori, ecc., acceso o spento per le lampade. Si è posto che i contatti aperti abbiano stato logico 0 e i contatti chiusi stato logico 1, analogamente i relè diseccitati hanno stato logico 0 e eccitati stato logico 1, infine le lampade spente hanno stato logico 0 e accese stato logico 1.

Questi schemi sono stati trattati quindi alla stregua delle porte logiche, poichè come esse possono rappresentare una funzione booleana.

Come si vedrà meglio in seguito, è possibile realizzare l'automatismo anche con una tecnologia diversa pur raggiungendo il medesimo risultato logico. Questa tecnologia è quella dei controllori logici programmabili (PLC), nei quali l'operatore deve inserire un programma che esprime la funzione logica.

In questo modo il PLC durante il funzionamento risolverà le equazioni logiche secondo il programma e le variabili di ingresso, il cui stato logico è definito dalla posizione di pulsanti, finecorsa, ecc.

Si danno ora alcuni semplici esempi di come sia possibile realizzare, con tecnologie elettriche/elettroniche diverse, uno stesso automatismo.

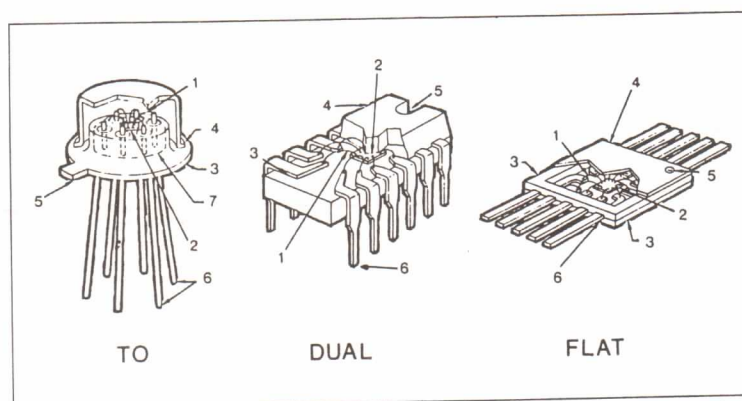


Fig. 2.23 - Soluzioni costruttive per contenitori di circuiti integrati: - 1-filo di collegamento, 2-piastrina (chip), 3-base, 4-copertura, 5-tacca di riferimento, 6-piedini (pins), 7-basetta isolata.

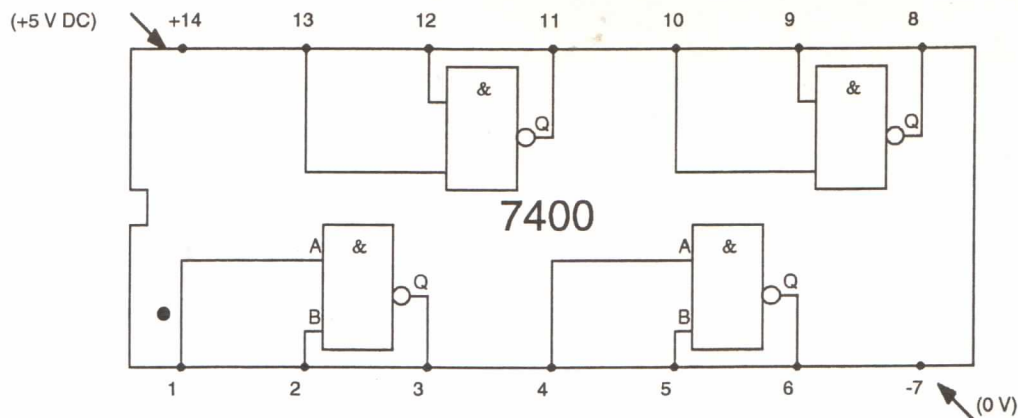


Fig. 2.24 - Circuito integrato 7400 contenente quattro porte logiche NAND a due ingressi.

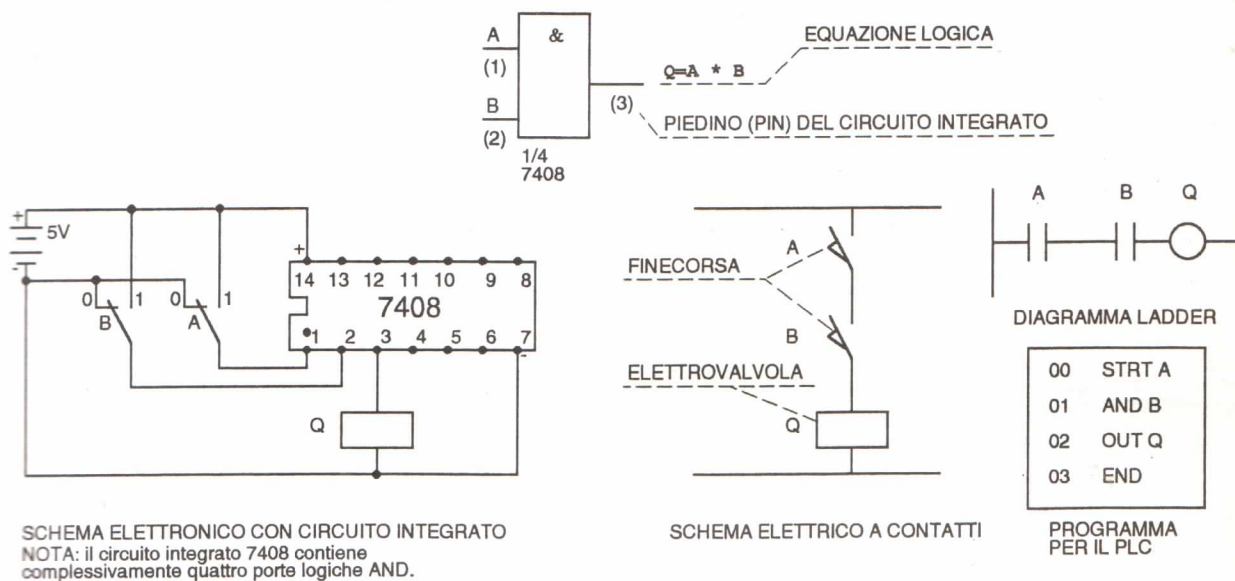


Fig. 2.25 - Controllo di una elettrovalvola mediante due finecorsa collegati in serie.

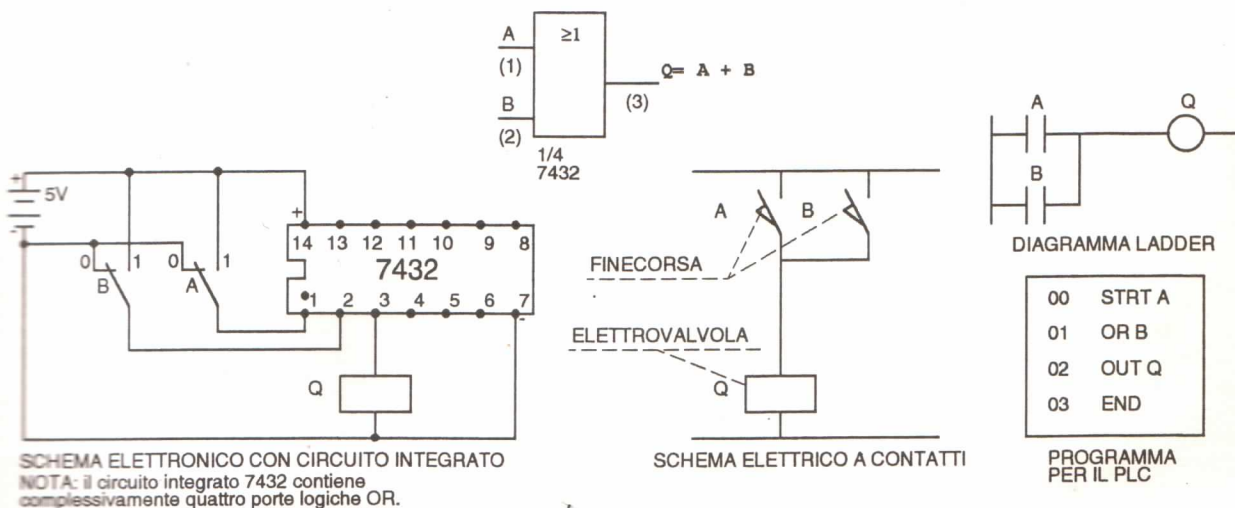
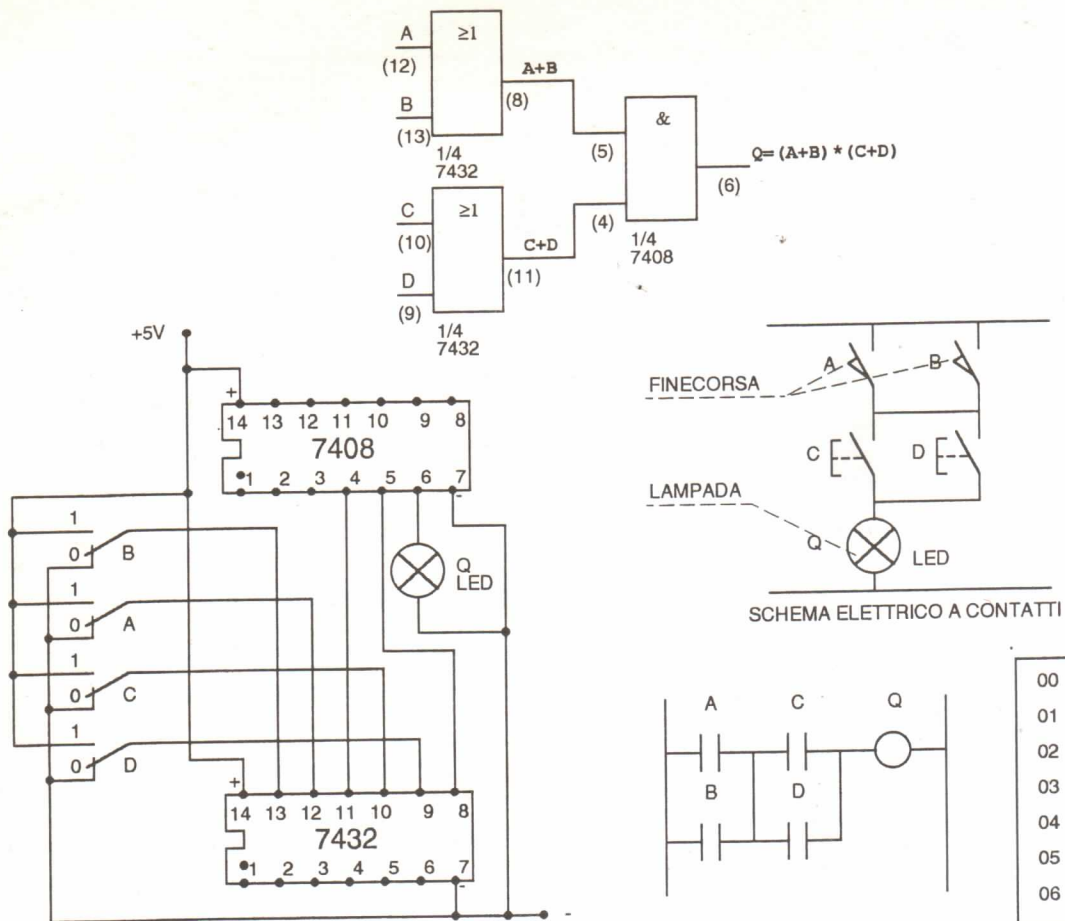


Fig. 2.26 - Controllo di una elettrovalvola mediante due finecorsa collegati in parallelo.



SCHEMA ELETTRONICO CON CIRCUITI INTEGRATI
 NOTA: il circuito integrato 7408 contiene complessivamente quattro porte logiche AND mentre il circuito integrato 7432 contiene complessivamente quattro porte logiche OR.

DIAGRAMMA LADDER

PROGRAMMA PER IL PLC

Fig. 2.27 - Controllo di una lampadina mediante due finecorsa collegati in parallelo, a loro volta in serie a due pulsanti normalmente aperti collegati in parallelo.

2.11 - DOMANDE

2.1) Quali sono gli operatori logici fondamentali?

2.2) Che cosa rappresenta la tabella della verità?

2.3) Qual è l'operatore logico che rappresenta il comportamento di contatti collegati in serie oppure collegati in parallelo?

2.4) Che cosa è una memoria?

2.5) Che cosa permettono di fare le proprietà dell'algebra di Boole?

2.6) Che cosa permettono di fare i teoremi di DE MORGAN?

2.7) Esiste un legame tra i circuiti elettronici e i circuiti elettrici?

2.12 - ESERCIZI

2.1) Rappresentare in uno schema elettrico a contatti le seguenti equazioni logiche:

$$Q = (A + B) * C * D; \quad Q = ((\bar{A} + \bar{B}) * C) + D * E$$

$$Q = A * \bar{B} * C + \bar{D} + E; \quad Q = (\bar{A} * B * \bar{C}) + (D * \bar{E} * F)$$

2.2) Rappresentare con uno schema logico le seguenti equazioni booleane:

$$Q = A * B * C + D; \quad Q = A * B + C * D; \quad Q = (A + B) * (\bar{C} + D); \quad Q = \bar{A} * (B + C * D)$$

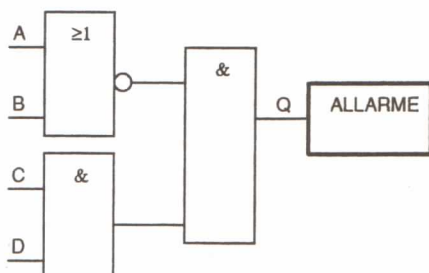
2.3) Minimizzare le seguenti equazioni booleane:

$$Q = (A + B) * (\bar{A} * \bar{B}); \quad Q = (A + B) * (\bar{A} + \bar{B}); \quad Q = (\bar{A} + \bar{B}) * (A * B * \bar{C});$$

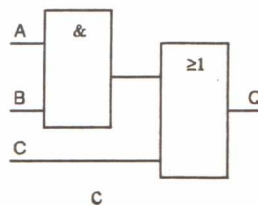
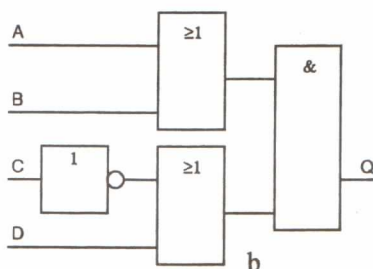
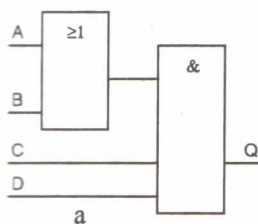
$$Q = A + \bar{B} + \bar{A} * B + \bar{C}; \quad Q = A + \bar{B} + \bar{A} * B + D * (A + B); \quad Q = (A * B + C) * (A * B + D);$$

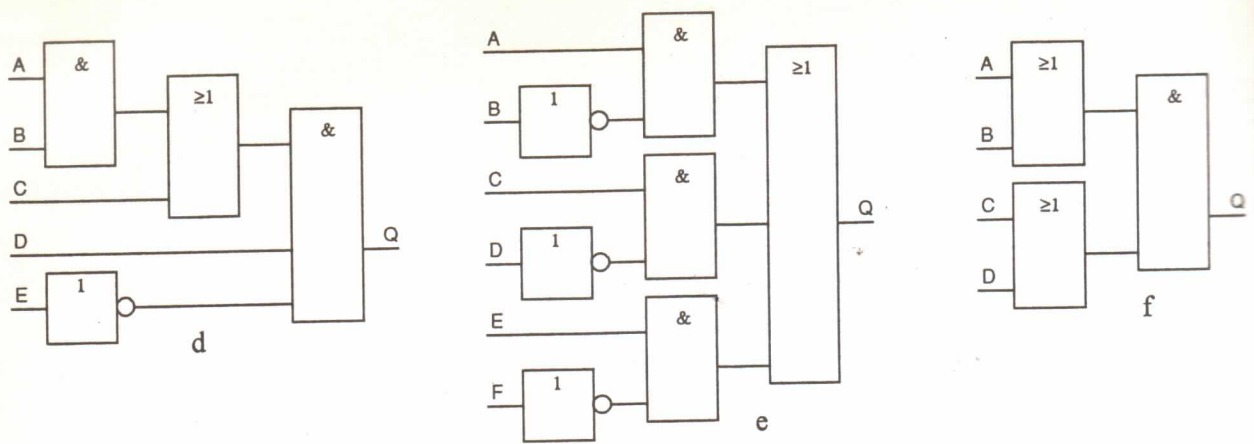
$$Q = A + \bar{A} * B + A * B; \quad Q = (A * B + C) * (A * B + D); \quad Q = B * C * \bar{D} + B * C * D$$

2.4) Il seguente schema logico è usato per attivare un allarme quando l'uscita Q è allo stato logico 1. Tracciare la tabella della verità relativa alle 16 possibili combinazioni delle 4 variabili indipendenti A, B, C, D.



2.5) Scrivere le equazioni booleane relative ai seguenti schemi logici:





2.6) Tracciare lo schema logico equivalente ai seguenti schemi elettrici a contatti:

