

Cilc per tutti gli appunti (AUTOMAZIONE – TRATTAMENTI TERMICI ACCIAIO – SCIENZA delle COSTRUZIONI...)



[e-mail per suggerimenti](#)

7 CIRCUITI SEQUENZIALI - MEMORIE

L'argomento verrà trattato più estesamente nel 2° e 3° volume. Qui si introdurranno i concetti generali, ponendo in evidenza il ruolo delle memorie senza affrontare la soluzione pratica di circuiti sequenziali, che saranno oggetto di studio del 2° volume.

Nei circuiti combinatori finora analizzati, la funzione di trasmissione y dipendeva solamente dalla combinazione dei valori delle variabili binarie di ingresso.

Ad una combinazione delle variabili di ingresso $x_1, x_2, x_3 \dots$, qualunque sia il tempo e la successione secondo cui esse si verificano, corrisponde una e una sola funzione di trasmissione:

$$y = f(x_1, x_2, x_3 \dots) \quad (7.1)$$

Nei circuiti combinatori occorre che siano verificate le seguenti ipotesi essenziali:

- La funzione di trasferta y non dipenda né dal tempo, né dalla sequenza secondo cui si verifica la combinazione dei valori delle variabili binarie di ingresso.
- Il flusso dei segnali avvenga sempre nello stesso senso: dall'ingresso verso l'uscita.
- Non vi siano rami di retroazione, per cui le uscite non vadano ad influire sulle condizioni degli ingressi.
- I componenti che costituiscono il circuito lavorino senza introdurre ritardi temporali in ingresso.

In un sistema reale vi possono essere degli elementi inerziali, non trascurabili, che comportano un ritardo del funzionamento. In questo caso il sistema è come se avesse memoria degli stati precedenti.

Per effetto del ritardo degli elementi inerziali, quando viene impartito un comando (segnale di ingresso), questo non viene eseguito istantaneamente e il sistema si trova ancora nello stato precedente che aveva nell'istante di immissione del comando stesso.

L'effetto di memoria dello stato precedente fa sì che uno stesso comando impartito al sistema in istanti diversi sortisce un effetto diverso.

Variabili di stato

Occorre così introdurre nuove variabili che descrivano e definiscano il sistema nell'istante attuale " i " e in quello successivo " $i+1$ " nel quale si attua il comando. A queste variabili si dà nome di *variabili di stato*.

Nei circuiti sequenziali si può far coincidere la variabile di stato, che definisce lo stato del sistema, con la variabile di uscita

Così l'uscita, che definisce lo stato del sistema nell'istante attuale " i ", si presenta come una variabile secondaria di ingresso accanto a quella primaria costituita dal segnale di comando. Anche da esso, infatti, dipende l'uscita del sistema.

Nello studio dei circuiti sequenziali occorre così considerare due istanti successivi: l'istante " i " attuale nel quale viene immesso il segnale di ingresso (comando), nel quale l'uscita si trova nello stato S_i e l'istante successivo " $i+1$ ", detto istante futuro nel quale il sistema ha subito una evoluzione (transizione) portando l'uscita ad un nuovo valore S_{i+1} detto stato futuro.

Per lo studio del comportamento di un circuito sequenziale, nella tabella della verità (detta anche tabella dello stato futuro), occorre considerare oltre alle variabili di ingresso, le variabili di uscita attuali nell'istante " i ", costituenti le variabili di stato attuali S_i e le stesse uscite nell'istante futuro " $i+1$ ", costituenti le variabili di stato futuro S_{i+1} .

Nello studio di un circuito sequenziale occorre considerare un *prima* e un *dopo*. La funzione di trasmissione determina l'evoluzione del sistema da uno stato al successivo: dall'istante i ad $i+1$.

In un circuito sequenziale, a parità delle combinazioni dei valori binari delle variabili di ingresso, la funzione di trasmissione y dipende dallo stato precedente S_i delle variabili di stato.

$$y = f(x_1, x_2, \dots, S_i, S_{i+1}, \dots) \quad (7.2)$$

Come si vedrà nel 2° volume, nel progetto di particolari circuiti sequenziali, si può avere la necessità che una stessa variabile di ingresso, immessa in due istanti diversi di un ciclo, debba dare una uscita diversa.

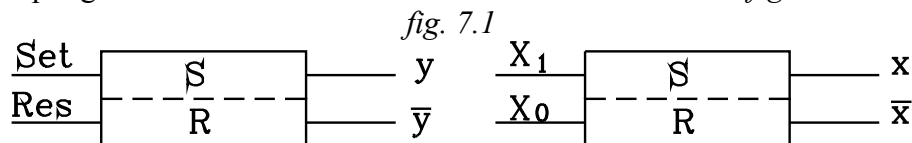
Per ottenere ciò, occorre disporre di altri due segnali (*detti di memoria*) da porre in *AND* al segnale di ingresso (*di comando*), in modo che, quando si presenta per la prima volta gli si pone in *AND* il primo segnale di memoria (*ed è disabilitato il secondo*), quando compare per la seconda volta gli si pone in *AND* il secondo segnale di memoria (*ed è disabilitato il primo*).

In tal modo lo stesso segnale di comando, in tempi diversi, ha in *AND* un differente *segnale di memoria*, ed essendo così diversificato ha la possibilità di ottenere uscite diverse.

7.1 Memorie

Il circuito di memoria è un circuito sequenziale capace di memorizzare due stati opposti in uscita, attivati, ciascuno, da un rispettivo ingresso.

Nella forma più generica una memoria viene indicata dal simbolo di *fig. 7.1*



Le uscite possono essere indicate con una stessa lettera posta in forma vera o in forma negata; ciò proprio ad indicare che sono due stati opposti ai quali si possono associare i valori binari 1,0

$$y = 1 \quad \bar{y} = 0$$

Le uscite verranno indicate a seconda dei casi con una lettera diversa. Ad esempio:

$$(y, \bar{y}) \quad (x, \bar{x}) \quad (U, \bar{U}), \dots$$

L'ingresso che determina l'uscita in forma vera (principale) viene detto di settaggio "*Set*", quello che porta in uscita la forma negata viene detto di resettaggio "*Reset*" o brevemente "*Res*".

Anche i segnali di ingresso, di *Set* e *Reset*, a seconda dei casi, verranno indicati con una lettera diversa. Ad esempio:

$$(I_1, I_2) \quad (I_1, I_0) \quad (X_1, X_0)$$

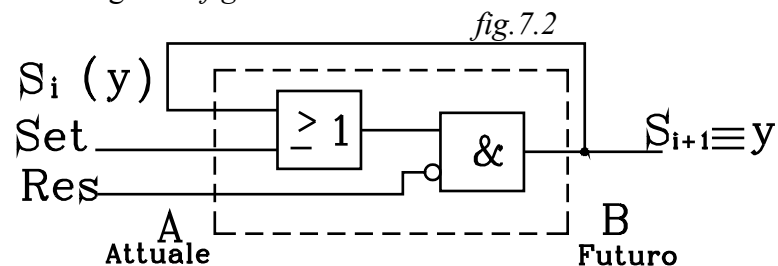
Nel testo viene preferito scegliere una lettera maiuscola per gli ingressi e minuscola per le uscite. ma ciò non è vincolante.

Le memorie possono essere ottenute o con il cablaggio dei componenti logici di base: NOT AND OR, della specifica tecnologia in uso (*pneumatica*, *elettrica ecc.*); oppure con componenti integrati.

Introduciamo ora due tipi di memorie che verranno più ampiamente studiate ed analizzate in seguito.

7.1.1 Memoria con precedenza alla cancellazione

Si consideri lo schema logico di *fig.7.2*



In questo la variabile d'uscita y torna in ingresso influenzando il valore che assumerà l'uscita stessa nell'istante futuro. In tal modo, a parità dei valori binari dati agli ingressi primari *Set*, *Res*, nell'istante attuale, a seconda del valore che ha in questo stesso istante la variabile di uscita si può ottenere, nell'istante futuro, un'uscita diversa.

Occorre pensare che le porte logiche, qualunque sia la tecnologia di realizzazione costruttiva, hanno in realtà una certa inerzia e, quindi, un ritardo di risposta dell'uscita rispetto all'ingresso.

Da quanto detto, occorre esaminare un istante iniziale (attuale) " i ", nel quale vengono considerate le immissione degli ingressi, compreso il valore che la variabile d'uscita ha in quell'istante, per poi determinare, nell'istante futuro " $i+1$ ", il valore che assumerà l'uscita, in conseguenza di quegli ingressi introdotti nell'istante precedente " i ".

Viene assunta come variabile di stato l'uscita " y ". Questa va considerata in due istanti diversi: l'attuale " i " e il futuro " $i+1$ ", costituendo così, rispettivamente, la variabile di stato attuale S_i e la futura S_{i+1} .

La variabile di stato futura S_{i+1} corrisponde al valore che la variabile d'uscita " y " assumerà, nell'istante futuro " $i+1$ " in uscita (punto B dello schema), che rappresenta *il dopo*.

La variabile di stato attuale S_i corrisponde al valore che attualmente, nell'istante " i ", possiede la variabile d'uscita " y ", che viene riportata in ingresso (punto A dello schema) rappresentante *il prima*.

La funzione binaria che si stabilisce tra l'uscita (punto B) e l'ingresso (punto A) determina il valore che avrà l'uscita " y " nell'istante futuro " $i+1$ " rispetto agli ingressi *Set*, *Res* immessi nell'istante precedente (attuale) " i " e al valore che in questo aveva la stessa uscita " y ".

La relazione binaria determina l'evoluzione del sistema tra l'istante attuale "i" e quello futuro "i+1". Per questo la funzione di trasmissione viene anche denominata *funzione di stato futuro* (Vol 3° Automi).

Dallo schema logico si ricava facilmente la funzione binaria.

$$S_{i+1} = (Set + S_i) \cdot \overline{Res} \quad (7.3)$$

Dove S_i al secondo membro rappresenta la variabile d'uscita nell'istante iniziale (attuale) "i", S_{i+1} è la variabile d'uscita "y" nell'istante futuro "i+1", *Set*, *Res* le variabili di ingresso primarie.

La relazione (7.3) può anche scrivere, per concisione, nella forma:

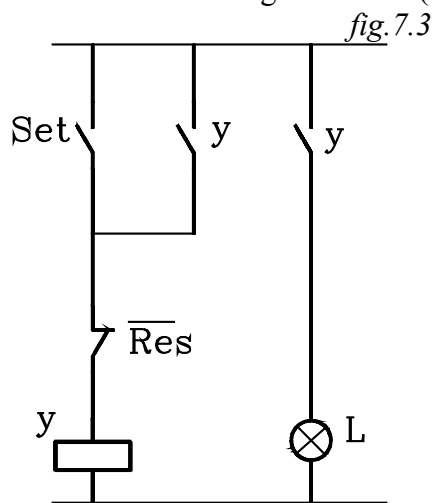
$$y = (Set + y) \cdot \overline{Res} \quad (7.4)$$

Sottintendendo che la "y" al secondo membro rappresenta il valore dell'uscita nell'istante iniziale "i" e quella al primo il valore della stessa nell'istante successivo "i+1".

Per capire meglio il funzionamento della memoria conviene tradurre lo schema logico in schema a contatti.

È da dire che, nello schema a contatti, la variabile d'uscita di una memoria del tipo considerato corrisponde all'eccitazione o diseccitazione di una bobina, il cui valore binario viene riportato in ingresso.

Dalla funzione logica binaria (7.4) si ottiene il circuito di *fig. 7.3*



Se nell'istante iniziale "i" la bobina è diseccitata: " $S_i = y = 0$ " e si chiude il contatto *Set*: " $Set = 1$ ", essendo quello \overline{Res} normalmente chiuso, la corrente passa sulla bobina. Nell'istante successivo "i+1" la bobina si eccita rendendo l'uscita "y" attiva: $S_{i+1} = y = 1$.

Dall'istante "i" al successivo "i+1" l'uscita "y" passa dal valore "0" al valore "1" (dalla diseccitazione alla eccitazione).

$$v = (1 + 0) \cdot \overline{0} = 1$$

Nell'istante "i+1" si chiude il contatto "y" comandato dalla bobina; cosicché, anche se il contatto *Set* viene aperto, attraverso quello "y" ora chiuso, la corrente continua a passare nella bobina, facendola rimanere eccitata.

$$y = (0 + 1) \cdot \overline{0} = 1$$

Attraverso il contatto "y", detto di *autoritenuta*, che riporta il segnale di uscita in ingresso, la bobina rimane eccitata, costituendo così la memorizzazione dello stato d'uscita ottenuto con il segnale *Set*, detto di *settaggio*

Nella figura il contatto "y" è posto in serie alla lampada "L" in modo che possa essere visibile l'effetto dell'eccitazione della bobina. Quando la bobina si eccita e "y" si chiude si accende la lampada.

Se ora in un istante successivo " $i+2$ " viene attivato il contatto Res ($Res=1$), questo, essendo normalmente chiuso, si apre: $Res = \bar{I} = 0$, interrompe la corrente che alimenta la bobina, diseccitandola e porta la variabile di uscita " y " a valore logico "0".

$$y = (0 + 1) \cdot \bar{I} = 0$$

Si osservi che qualunque sia la condizione logica del segnale Set di settaggio e dell'uscita " y ", quando si attiva il segnale Res di resettaggio, la bobina, nell'istante successivo, risulta diseccitata. Infatti quando si attiva il contatto Res , questo si apre interrompendo comunque l'alimentazione della bobina, in qualsiasi condizione dei contatti: di Set e " y ".

$$y = (Set + y) \cdot \bar{I} = 0$$

Immettendo contemporaneamente i due segnali di settaggio Set e resettaggio Res , risulta prevalente l'azione di quest'ultimo. La memoria si dice con prevalenza alla cancellazione o disattivazione.

$$y = (1 + y) \cdot \bar{I} = 0$$

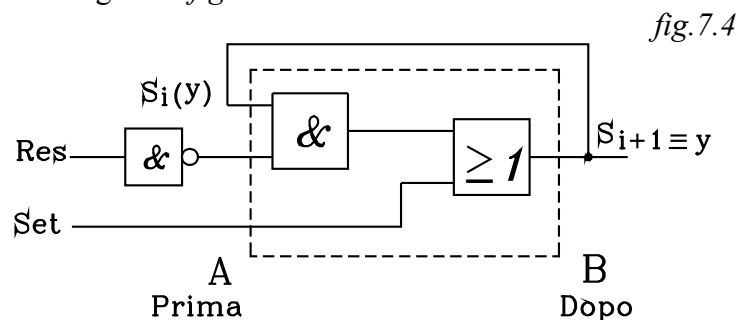
IN LABORATORIO

Per esercitazione verificare quanto detto nel precedente paragrafo:

- montare il circuito di figura fig.7.3
- Verificare che pigiato il tasto di Set si accende la lampada L , che rimane accesa anche quando si apre il tasto di Set
- Pigiando il tasto Res la lampada si spegne.
- Pigiando sia il tasto Set che Res : se la lampada è accesa si spegne se è spenta rimane in tale condizione

7.1.2 Memoria con precedenza alla scrittura (attivazione)

Si consideri lo schema logico di fig.7.4



Anche in questo caso la variabile d'uscita y torna in ingresso influenzando il valore che assumerà l'uscita stessa nell'istante futuro.

Viene assunta come variabile di stato l'uscita " y ": $S \equiv y$ che andrà considerata nei due istanti successivi " i " e " $i+1$ ".

La relazione binaria, che si stabilisce tra l'ingresso (punto A) e l'uscita (punto B), determina il valore che il segnale di uscita " y " assumerà nell'istante futuro " $i+1$ ", in funzione degli ingressi:

Set, *Res*, immessi nell'istante precedente (attuale) "*i*" e del valore che in questo aveva la stessa uscita "*y*".

Dallo schema logico si ricava facilmente la funzione binaria.

$$S_{i+1} = \overline{Res} \cdot S_i + Set$$

Dove S_i al secondo membro rappresenta la variabile d'uscita nell'istante iniziale (attuale) "*i*", mentre S_{i+1} è la variabile d'uscita "*y*" nell'istante futuro "*i+1*"; *Set*, *Res* sono le variabili di ingresso primarie.

La relazione (7.3) si può anche scrivere, per concisione, nella forma:

$$y = \overline{Res} \cdot y + Set \quad (7.5)$$

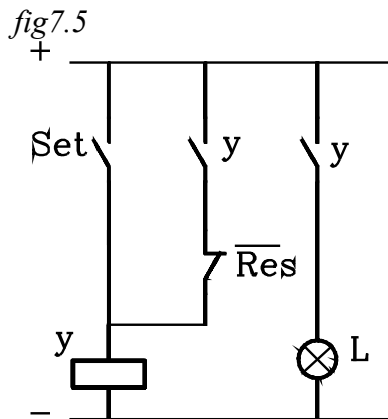
Sottintendendo che la "*y*" al secondo membro rappresenta il valore dell'uscita nell'istante iniziale "*i*" e quella al primo il valore della stessa nell'istante successivo "*i+1*".

Dalla funzione logica binaria (7.5) si ottiene il circuito di fig.7.5

Se nell'istante iniziale "*i*" la bobina è diseccitata: " $S_i = y = 0$ " e si chiude il contatto *Set*: " $Set = 1$ " la corrente passa sulla bobina qualunque sia lo stato del contatto *Res*. Nell'istante successivo "*i+1*" la bobina si eccita rendendo l'uscita "*y*" attiva: $S_{i+1} = y = 1$.

Dall'istante "*i*" al successivo "*i+1*" l'uscita "*y*" passa dal valore "0" al valore "1" (dalla diseccitazione alla eccitazione).

$$y = \overline{Res} \cdot 0 + 1 = 1$$



Nell'istante "*i+1*" si chiude il contatto "*y*" comandato dalla bobina, cosicché anche se il contatto *Set* viene aperto, attraverso quello "*y*" di ritenuta ora chiuso, la corrente continua a passare nella bobina facendola rimanere eccitata, se nello stesso istante non è attivato il segnale *Res*.

$$v = \overline{0} \cdot 1 + 0 = 1$$

La memoria viene resettata quando non è attivo il segnale di settaggio " $Set = 0$ " e attivato quello di reset " $Res = 1$ ".

$$y = \overline{1} \cdot 1 + 0 = 0$$

Se si attivano contemporaneamente, nell'istante "*i*", i due contatti *Set*, *Res*, prevale quello di settaggio: la memoria, nell'istante "*i+1*" rimane attivata

$$y = \overline{1} \cdot 1 + 1 = 1$$

Immettendo contemporaneamente i due segnali di resettaggio *Res*, e di settaggio *Set* risulta prevalente l'azione di quest'ultimo. La memoria si dice con prevalenza alla scrittura o attivazione.

Per esercitazione verificare quanto detto nel precedente paragrafo:

- montare il circuito di figura fig.7.5
- Verificare che pigiato il tasto di *Set* si accende la lampada *L*, che rimane accesa anche quando si apre il tasto di *Set*
- Pigiando il tasto *Res* la lampada si spegne se non è contemporaneamente pigiato il tasto di *Set*.
- Pigiando sia il tasto *Set* che *Res*: se la lampada comunque si accende, anche se è pigiato il tasto di *Reset*

7.2 Flip - Flop

Nell'elettronica sono realizzati circuiti di memoria che vanno sotto il nome di *flip-flop*, ottenuti o utilizzando elementi combinatori elementari (porte logiche) oppure con circuiti integrati.

Qui verranno descritti i principali tipi di *flip-flop* determinando, per ciascuno di essi, lo schema logico, la cui realizzazione potrà poi essere effettuata con tecnologie diverse.

I *flip-flop* sono circuiti sequenziali bistabili, dotati di due ingressi e due uscite. Queste possono assumere, a seconda dei valori binari assegnati agli ingressi, due stati stabili opposti: nel senso che, quando un'uscita è nello stato logico "1" l'altra è allo stato "0" e viceversa.

Le transizioni di stato possono avvenire nei determinati istanti di tempo nei quali cambiano i valori binari degli ingressi, oppure, con sincronismo, in istanti scanditi da un orologio detto *Clock*.

Quando i *flip-flop* hanno transizioni non sincronizzate si dicono *flip-flop asincroni* (oppure memorie *latch* (chiavistello); quando, invece, la scansione degli istanti di transizione da uno stato all'altro è regolata da un orologio i *flip-flop* vengono detti "*flip-flop sincroni*".

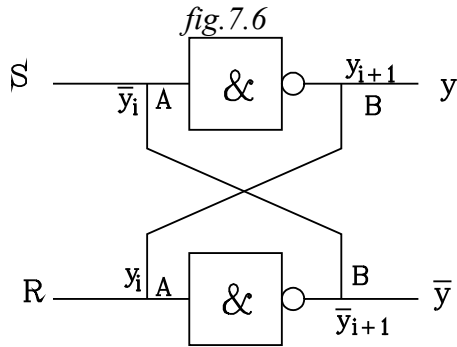
7.2.1 Flip-Flop RS asincrono

Il nome RS del *flip-flop* deriva dal fatto che in questo vi è un ingresso, detto di *Set* (attivazione), che determina l'attivazione dell'uscita principale che indicheremo con "*y*" (può essere indicata con altri simboli: usuale è anche "*q*"), mentre disattiva l'altra uscita, e un secondo ingresso, detto di *Reset* (disattivazione), che attiva l'uscita complementare che indicheremo con \bar{y} , mentre disattiva la prima.

L'indicazione dell'uscita complementare con \bar{y} è giustificata perché essa si deve trovare sempre allo stato logico opposto a quello dell'uscita principale. Quando questa è allo stato logico 1 ($y=1$) si dice anche che il *flip-flop* è *settato*; quando l'uscita principale è allo stato logico "0" ($y=0$) il *flip-flop* si dice *resettato*: in questo caso risulterà $\bar{y} = 1$.

I *flip-flop* RS possono essere di due tipi: con settaggio a livello basso o alto; ossia si possono costruire *flip-flop* che vengono attivati ($y=1$) con segnali di *Set* $S = 0$, e altri attivati con segnali di *Set* $S = 1$. Lo stesso discorso vale per il resettaggio *R*

Un *flip-flop* che viene attivato con segnale di *Set* $S = 0$ e resettato con $R=0$ può essere costituito da due porte NAND collegate come in fig.7.6



Come si può constatare dalla figura vi è anche in questo tipo di memorie un ritorno dell'uscita in ingresso: l'uscita da una porta NAND è collegata con l'ingresso dell'altra.

L'ingresso $S=0$ determina il settaggio dell'uscita y e il resettaggio di \bar{y} , Viceversa l'ingresso $R=0$ determina l'attivazione dell'uscita \bar{y} e la disattivazione di y .

Non debbono essere posti in ingresso contemporaneamente il segnale di settaggio $S=0$ e $R=0$, in quanto l'uscita, in questo caso, risulta indefinita.

Nello studio del *flip-flop* occorre considerare un istante attuale " i " ed uno futuro " $i+1$ ".

I segnali di ingresso si considerano immessi nell'istante attuale " i "; la variabile di uscita, invece, occorre considerarla in due istanti successivi: quella attuale y_i è il valore logico dell'uscita nello stesso istante attuale " i " in cui si considerano gli ingressi, quella futura y_{i+1} è il valore logico dell'uscita che si otterrà, nell'istante successivo " $i+1$ ", nella transizione verso il nuovo stato stabile (che può essere anche lo stesso dell'istante precedente)

In "B" lo stato logico dell'uscita da considerarsi è la y_{i+1} che si otterrà nello stato futuro " $i+1$ ", mentre nell'ingresso "A" è applicato lo stato logico dell'uscita y_i che si ha nell'istante attuale " i ".

Dallo schema funzionale si ricava facilmente la funzione binaria dello stato futuro dell'uscita

$$y_{i+1} = \overline{S \cdot \bar{y}_i} \quad (7.6) \quad \text{uscita } y$$

$$\bar{y}_{i+1} = \overline{R \cdot y_i} \quad (7.7) \quad \text{uscita } \bar{y}$$

fig.7.7

S	R	y_i	\bar{y}_i	y_{i+1}	\bar{y}_{i+1}
0	0	1	0	?	?
0	0	0	1	?	?
0	1	1	0	1	0
0	1	0	1	1	0
1	0	1	0	0	1
1	0	0	1	0	1
1	1	1	0	1	0
1	1	0	1	1	0

Si può così stendere la tabella della verità

In questa ogni combinazione degli stati logici binari di S e R è ripetuta a seconda dei due stati opposti che possono assumere attualmente le due uscite y_i, \bar{y}_i .

Così, per esempio, la combinazione $S=0, R=1$ viene ripetuta per $y_i=1, \bar{y}_i=0$ e per la condizione opposta $y_i=0, \bar{y}_i=1$ (3° - 4° riga)

Sostituendo nelle (7.6) e (7.7) le variabili S ed R e le variabili di uscita attuali y_i, \bar{y}_i si ottengono le variabili di uscita future.

Così per esempio, si consideri la 3° riga.

- Le variabili di ingresso sono: $S=0, R=1$.
- L'uscita attiva nell'istante attuale " i " è $y_i=1$.
- Questa si riporta all'ingresso dell'altro NAND, ove è immesso $R=1$ e determina l'uscita \bar{y}_{i+1}

$$\bar{y}_{i+1} = \overline{R \cdot y_i} = \overline{1 \cdot 1} = 0$$

- Ora all'ingresso del primo NAND si ha : $S = 0$ e $\bar{y}_i = 0$ e si ottiene in uscita:

$$y_{i+1} = \overline{S \cdot \bar{y}_i} = \overline{1 \cdot 0} = 1$$

In questo caso si constata che: $y_{i+1} = y_i$ $\bar{y}_{i+1} = \bar{y}_i$

Si ha uno stato stabile ove lo stato futuro è uguale a quello attuale

Si consideri ora la 4° riga:

- Le variabili di ingresso sono: $S = 0$ $R = 1$.
- L'uscita attiva nell'istante attuale "i" è $\bar{y}_i = 1$.
- Questa si riporta all'ingresso del primo NAND, ove è immesso $S = 0$ e determina l'uscita y_{i+1}

$$y_{i+1} = \overline{S \cdot \bar{y}_i} = \overline{0 \cdot 1} = 1$$

- Questa uscita si porta all'ingresso dell'altro NAND dove $R = 1$ e si ha nell'istante successivo:

$$\bar{y}_{i+1} = \overline{R \cdot y_i} = \overline{1 \cdot 1} = 0$$

In questo caso si ha una transizione del *flip-flop* che fa passare l'uscita y dallo stato iniziale $y_i = 0$ a quello futuro $y_{i+1} = 1$; mentre l'altra uscita passa dallo stato iniziale $\bar{y}_i = 1$ a quello futuro $\bar{y}_{i+1} = 0$

In un istante successivo "i+3" restando $S = 0$ $R = 1$, con le uscite ottenute, si ricade nel caso precedente, analizzato nella 3° riga e si ha uno stato stabile.

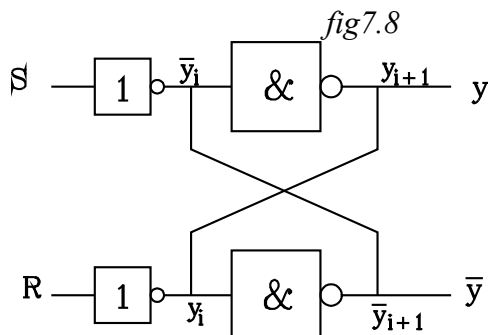
Lo stesso ragionamento vale per le righe successive della tabella della verità.

Un discorso a parte va fatto per le prime due righe della tabella della verità, nelle quali si prospetta il caso che in entrambi gli ingressi sia immesso un segnale di attivazione: $S = 0$ $R = 0$ (ricordiamo che si ha l'attivazione con basso livello "0").

In questo caso lo stato futuro non è stabile e le uscite possono variare di valore logico da un istante all'altro.

Concludendo, considerando y come uscita principale di attivazione, questo tipo di *flip-flop* viene settato ponendo in ingresso $S = 0$ ($R = 1$) e resettato ponendo $R = 0$ ($S = 1$)

Affinché il segnale di attivazione del *Set* e *Reset* avvenga con logica positiva a livello alto "1" basta porre un NOT su ciascun ingresso dei due NAND. Si ottiene lo schema logico di *fig.7.8*



In questo *flip-flop* si ottiene il settaggio dell'uscita principale y quando il segnale di *Set* è $S = 1$ (e $R = 0$). Si ha il resettaggio quando $R = 1$ (e $S = 0$)

Le funzioni logiche divengono:

$$y_{i+1} = \overline{\overline{S} \cdot \bar{y}_i}$$

$$\bar{y}_{i+1} = \overline{\overline{R} \cdot y_i}$$

fig.7.9

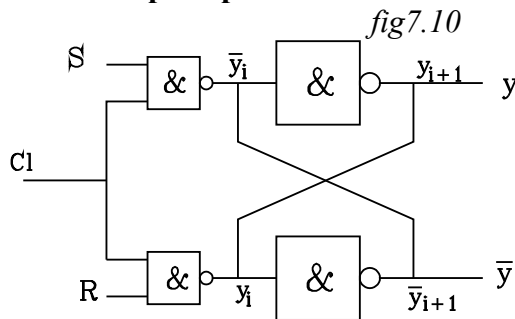
S	R	S	R	y_i	\bar{y}_i	y_{i+1}	\bar{y}_{i+1}
0	0	1	1	0	1	0	1
0	0	1	1	1	0	1	0
0	1	1	0	0	1	0	1
0	1	1	0	1	0	0	1
1	0	0	1	0	1	1	0
1	0	0	1	1	0	1	0
1	1	0	0	0	1	?	?
1	1	0	0	1	0	?	?

Ragionando come per il *flip-flop* precedente si ottiene la tabella della verità riportata in fig.7.9.

Si osservi che in ingresso dei NAND si riportano i segnali \bar{R} e \bar{S} . Questi in combinazione delle uscite attuali y_i, \bar{y}_i danno una tabella analoga alla precedente.

Dall'analisi della tabella della verità si conclude che il *flip-flop* è settato quando $S = 1$ e $R = 0$ e resettato quando $S = 0$ e $R = 1$

7.2.2 Flip-Flop RS sincrono



In questi *flip-flop* le transizioni da uno stato all'altro si ottengono solamente in presenza di un segnale di sincronismo, che si succede in istanti regolari di tempo, scanditi da un orologio (*clock*).

Il clock viene posto in AND con i segnali di Set "S" e Reset "R" in modo che questi vengano resi attivi solamente in presenza di quello.

In figura si riporta lo schema logico del *flip-flop sincrono*, con segnali di settaggio in logica di livello alto. In tal caso gli AND sono seguiti da un NOT, costituendo un NAND in ingresso

Nella tabella della verità la combinazione degli stati logici opposti degli ingressi S, R (0,1 1,0) vengono ripetuti a seconda delle due combinazioni possibili delle due uscite attuali y_i, \bar{y}_i : 0,1 - 1,0 e a seconda che il segnale di clock sia "0" od "1".

Così la combinazione 0,0 di S,R , viene ripetuta 4 volte: con la combinazione 0,1 di y_i, \bar{y}_i ripetuta due volte con segnali di clock $Cl=1$ e $Cl=0$ e con la combinazione 1,0 di y_i, \bar{y}_i ripetuta due volte con segnali di clock $Cl=1$ e $Cl=0$

S	R	y_i	\bar{y}_i	Cl	y_{i+1}	\bar{y}_{i+1}
0	0	0	1	0	0	1
0	0	0	1	1	0	1
0	0	1	0	0	1	0
0	0	1	0	1	1	0
0	1	0	1	0	0	1
0	1	0	1	1	0	1
0	1	1	0	0	1	0
0	1	1	0	1	0	1
1	0	0	1	0	0	1
1	0	0	1	1	1	0
1	0	1	0	0	1	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	0	1	1	?	?
1	1	1	0	0	1	0
1	1	1	0	1	?	?

fig.7.11

Quando il segnale di clock "Cl" è allo stato logico "0" ($Cl=0$) dai due AND di ingresso (prima del NOT) non si ha alcun segnale di uscita e quindi il *flip-flop* non ha alcuna transazione di stato e le uscite rimangono invariate: gli stati futuri coincidono con gli attuali.

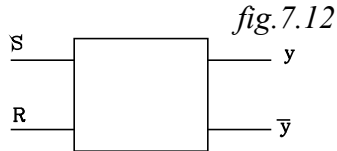
Quando il segnale di clock "Cl" è allo stato logico "1" ($Cl=1$) il *flip-flop* risponde ai segnali di Set "S" e Reset "R" come il precedente

asincrono. La differenza rispetto a questo risiede nel fatto che le transizioni si ottengono solamente quando $Cl=1$

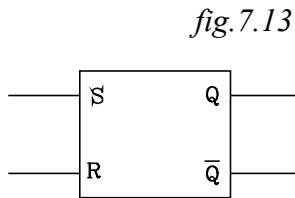
In conclusione per redigere la tabella della verità basta tener conto che per $Cl=1$ si hanno le stesse transazioni riportate nella tabella precedente, mentre, per $Cl=0$ non si hanno transazioni e lo stato futuro delle uscite è uguale a quello attuale.

7.2.3 Simboli per i Flip - Flop RS

Il *Flip - Flop* per comodità di rappresentazione vengono indicati con apposito simbolo.



Nella figura *fig.7.12* è rappresentato il FF RS con logica positiva (settaggio a livello alto).



Le uscite che abbiamo indicato, per uniformità con i capitoli precedenti, con y, \bar{y} , vengono usualmente indicate con Q, \bar{Q} .

fig.7.13

La tabella della verità si può porre in forma abbreviata, indicando solamente lo stato logico futuro dell'uscita principale: la secondaria assumerà lo stato complementare.

fig.7.14

S	R	y_{i+1}
0	0	y_i
1	0	1
0	1	0
1	1	Indesiderato

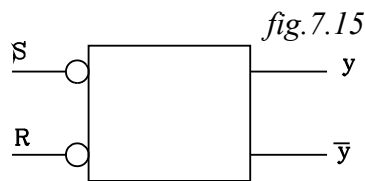
Così nella bella di figura *fig.7.14* viene riportata la tabella della verità del FF RS con logica positiva.

Quando non vi è alcun segnale sui due ingressi ($S = 0, R = 0$), allora lo stato futuro dell'uscita è uguale a quello attuale $y_{i+1} = y_i$.

Quando $S=1, R=0$ risulta $y_{i+1} = 1$

Quando $S=0, R=1$ risulta $y_{i+1} = 0$

Non è desiderato il caso che entrambi gli ingressi siano settati.



Nella figura *fig.7.15* è rappresentato un FF RS con logica negativa. Avendo indicato con lo schema di *fig.7.13* il *flip - flop* con logica positiva per indicare quella con logica negativa si pongono due negati agli ingressi. (Nello schema logico con i NAND si è fatto il contrario).

Il simbolo che indica il FF RS sincrono è uguale a quello asincrono con l'aggiunta dell'ingresso del clock. Nella figura *fig.7.16* è rappresentato un FF RS sincrono con logica positiva, nel quale l'ingresso "C" indica il segnale di *clock*

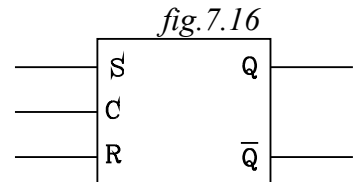


fig.7.17

S	R	C	y_{i+1}
x	x	0	y_i
1	0	1	1
0	1	1	0
1	1	1	Non desiderato

In questo caso la tabella della verità abbreviata è quella rappresentata in figura *fig.7.17*.

Quando il segnale di *clock* è allo stato logico "0", allora, qualunque siano gli stati logici degli ingressi (indicati con x) lo stato logico futuro y_{i+1} dell'uscita principale risulta uguale a quella attuale: $y_{i+1} = y_i$

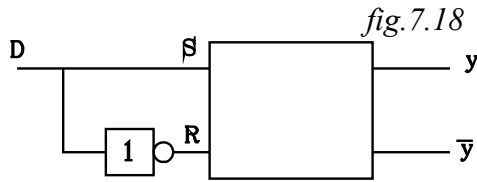
Quando il segnale di *clock* è allo stato logico "1" il FF RS sincrono si comporta come quello asincrono:

Quando $S=1 R=0$ risulta $y_{i+1} = 1$

Quando $S=0 R=1$ risulta $y_{i+1} = 0$

Non è desiderato il caso che entrambi gli ingressi siano settati

7.2.4 Flip - Flop tipo D asincrono



Si ricava dal FF RS collegando l'ingresso di settaggio *S* con quello di resettaggio *R* attraverso un NOT. Si ottiene così un FF con un solo ingresso che viene indicato con *D* coincidente con *S*.

Sia il FF SR utilizzato con logica positiva. In tal caso risulta:

$$y_{i+1} = \overline{\overline{S} \cdot \overline{y}_i}$$

$$\overline{y}_{i+1} = \overline{\overline{R} \cdot y_i}$$

Essendo :

$$S = D \quad e \quad R = \overline{D}$$

Le espressioni delle uscite nell'istante futuro diventano:

$$y_{i+1} = \overline{\overline{D} \cdot \overline{y}_i}$$

$$\overline{y}_{i+1} = \overline{D \cdot y_i}$$

Considerando quindi l'uscita principale *y* nell'istante futuro "*i+1*", si ha che:

$$\text{Per } D = 1 \quad y_{i+1} = 1$$

$$\text{Per } D = 0 \quad y_{i+1} = 0$$

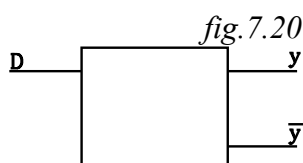
Nel FF *D* l'uscita principale *y* assume lo stesso stato logico dell'ingresso *D*, memorizzandolo, con un ritardo, corrispondente alla transizione tra l'istante attuale "*i*" e l'istante futuro "*i+1*".

Il FF *D* memorizza con un ritardo lo stato logico inviato all'unico ingresso *D*, rendendolo disponibile nell'uscita principale *y*.

fig.7.19

D	y_{i+1}
1	1
0	0

Nella figura *fig.7.19* viene riportata tabelle della verità abbreviata



Nella figura *fig.7.20* è riportato il simbolo del FF *D*. Come già detto le uscite vengono qui indicate con y, \overline{y} per uniformità di simbolismo con gli argomenti trattati precedentemente

7.2.5 Flip - Flop tipo D sincrono

In questo caso si collega l'ingresso di Set S del *flip - flop* sincrono con quello di Reset R attraverso un NOT, come rappresentato in figura *fig.7.21*.

In ingresso si ha oltre a D il segnale di *clock* C .

Questo tipo di FF presenta le stesse commutazioni di quello asincrono, quando il segnale di *clock* C è allo stato logico "1".

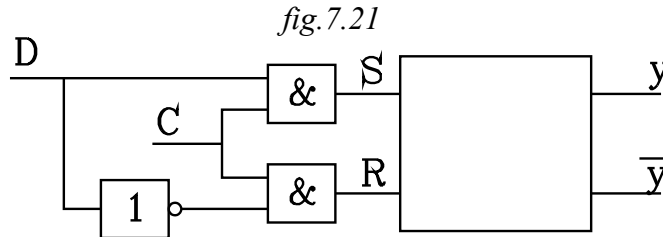


fig.7.22

D	C	y_{i+1}
x	0	y_i
0	1	0
1	1	1

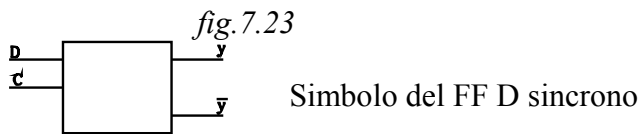
Nella figura *fig.7.22* è riportata la tabella della verità abbreviata. Quando il segnale di *clock* è allo stato logico "0", all'uscita dei due AND si ottiene lo stato logico "0": $S = 0$, $R = 0$.

In tal caso, in uscita dal *flip - flop*, nell'istante futuro " $i+1$ " si otterrà lo stesso stato logico che si ha nell'istante attuale " i " ($y_{i+1} = y_i$), e qualunque sia lo stato logico dell'ingresso D .

Nella tabella della verità l'indifferenza dello stato logico dell'ingresso D è indicato con una "x"

Quando il segnale di *clock* è allo stato logico "1", allora il *flip - flop* si comporta come l'asincrono:

$$\begin{aligned} \text{Per } D = 1 \quad & y_{i+1} = 1 \\ \text{Per } D = 0 \quad & y_{i+1} = 0 \end{aligned}$$



7.2.5 Abilitazione del DFF sul fronte di salita del segnale di clock

In questo tipo di FF D la commutazione avviene quando si ha una variazione dello stato dell'ingresso D e in corrispondenza del fronte di salita del segnale di *clock*: quando, cioè, questo si porta dallo stato logico "0" allo stato "1".

Per la commutazione del *flip - flop* occorre il verificarsi di due eventi:

1. che sia variato lo stato logico dell'ingresso D
2. che vi sia la variazione del segnale di clock dal livello basso a quello alto

Se varia lo stato logico dell'ingresso D quando quello del segnale di clock è allo stato "1", non si ottiene alcuna commutazione: occorre, perché ciò avvenga, che il segnale di clock torni allo stato "0" e si riporti allo stato "1", in modo da avere un fronte di salita che abiliti il *flip - flop*.

Nella figura *fig.7.24* è riportata la tabella della verità abbreviata. In questa si nota che:

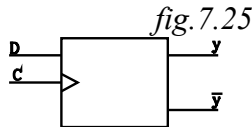
fig.7.24

D	C	y_{i+1}
x	0	y_i
x	1	y_i
x	\downarrow	y_i
1	\uparrow	1
0	\uparrow	0

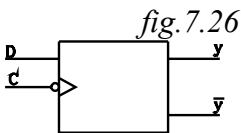
- Quando il segnale di *clock* permane stabilmente ad uno stato logico: "0" od "1", allora lo stato dell'uscita nell'istante futuro risulta uguale a quello attuale, qualunque sia lo stato dell'ingresso *D*.
In corrispondenza di $C=0$ e $C=1$ si segna nell'ingresso *D* una "x", indicante l'indifferenza dello stato.
- Quando il segnale di *clock* passa per un fronte di discesa, dallo stato logico "1" allo stato "0" rimane invariata l'uscita, qualunque sia lo stato logico dell'ingresso *D*.
- In corrispondenza del fronte di salita del segnale di *clock* il *flip - flop* si comporta come l'asincrono:

$$\begin{aligned} \text{Per } D = 1 \quad y_{i+1} &= 1 \\ \text{Per } D = 0 \quad y_{i+1} &= 0 \end{aligned}$$

Si possono avere anche *flip - flop* di tipo *D* nei quali l'abilitazione alla commutazione è ottenuta con il fronte di discesa del segnale di *clock*: quando passa dallo stato "1" allo stato "0".



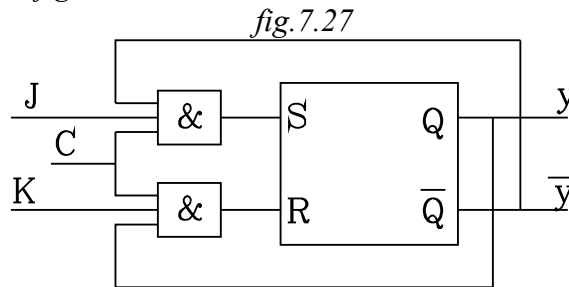
Simbolo del FF D abilitato dal fronte di salita del segnale di *clock*



Simbolo del FF D abilitato dal fronte di discesa del segnale di *clock*

7.2.6 Flip - Flop JK sincrono

Il *flip- flop* JK si ricava da uno RS sincrono collegando le uscite agli ingressi degli AND come rappresentato in figura fig.7.27.



L'uscita *y* viene collegata all'ingresso dell'AND posto sul *Reset "R"*, mentre l'uscita \bar{y} è collegata con l'ingresso dell'altro AND, posto sul *Set "S"*.

Con questo tipo di FF viene eliminata la condizione di indeterminazione che si verificava quando entrambi gli ingressi si ponevano nello stato di settaggio: stato 1 nel caso di logica positiva.

Nel FF JK, quando gli ingressi si pongono nella stessa condizione di settaggio, in corrispondenza del segnale di *clock* si ottiene l'inversione degli stati in uscita.

L'abilitazione alla commutazione può essere ottenuta o con fronte di salita o di discesa del segnale di *clock*

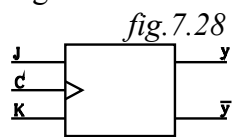
Così, nel caso di logica positiva e con abilitazione alla commutazione con fronte di salita del segnale di *clock*, quando entrambi gli ingressi sono nello stato logico 1, allora, in corrispondenza

di un fronte di salita segnale di *clock*, si avrà una inversione degli stati nelle due uscite: se $y = 0$ e $\bar{y} = 1$ si ha una commutazione con inversione degli stati e risulta: $y = 1$ e $\bar{y} = 0$

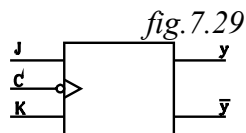
Infatti si supponga che nell'istante attuale "*i*" sia $y = 0$ e $\bar{y} = 1$, si abbiano entrambi gli ingressi nello stato logico "1" e si verifica un fronte di salita abilitante del segnale di *clock*. In queste condizioni si ottiene che:

- L'uscita $y = 0$ nell'istante "*i*" si riporta nell'ingresso dell'AND del resettaggio, conducendo così l'uscita \bar{y} ad assumere nell'istante futuro "*i+1*" lo stesso valore "0" posto in ingresso: $\bar{y}_{i+1} = 0$.
- Contemporaneamente l'uscita $\bar{y} = 1$ nell'istante "*i*" si riporta nell'ingresso dell'AND del settaggio che porta l'uscita "y" ad assumere nell'istante "*i+1*" lo stesso valore "1" posto in ingresso: $y_{i+1} = 1$

In corrispondenza di un altro fronte di salita del segnale di *clock* si avrà un'altra inversione degli stati delle uscite.



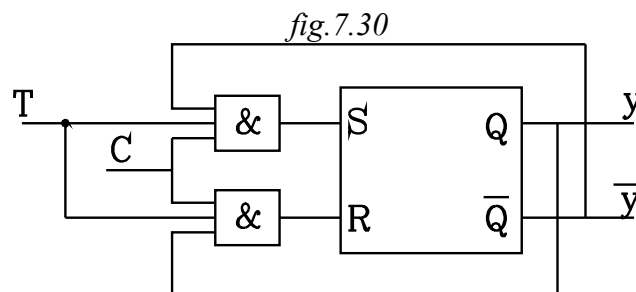
Simbolo di un JKFF con abilitazione sul fronte di salita



Simbolo di un JKFF con abilitazione sul fronte di discesa

7.2.6 Flip - Flop T sincrono

Si ottiene dal FF JK collegando l'ingresso K al J, lasciando questo come unico ingresso del FF. L'unico ingresso si indica con T.



Dal collegamento dei due ingressi ne deriva che in questo *flip-flop* si ha l'inversione degli stati delle uscite quando l'ingresso T è allo stato logico "1" e si verifica il segnale di abilitazione del *clock*, che può essere sul fronte di salita o di discesa.

Con questo FF quando T è allo stato logico "1" ad ogni segnale di abilitazione del *clock* si ha una inversione degli stati logici delle uscite.

L'uscita principale *y* passa quindi alternativamente dallo stato logico 1 allo stato 0 nei successivi segnali di abilitazione del *clock*.

7.2.6 Cenni sulla utilizzazione dei Flip - Flop

7.2.6.1 Memorie

I *flip-flop* si prestano ad essere utilizzati per la memorizzazione delle informazioni. Un FF memorizza un sol *bit*, che può assumere i due stati opposti "0,1". Per memorizzare parole composte da più bit occorre l'impiego di un certo numero di FF opportunamente collegati tra loro.

Così la memoria RAM è costituita da celle elementari che possono assumere i due stati opposti "0,1" realizzate con FF. Così pure i Registri possono essere costituiti da un insieme di FF collegati in cascata.

7.2.6.2 Contatori

Nelle applicazioni di automazione occorre spesso effettuare operazioni di conteggio.

Questo consiste nel fornire sequenzialmente una informazione diversa, espressa con una parola da "*n*" *bit*, in corrispondenza ad una successione di segnali impulsivi di comando.

Ad ogni impulso deve corrispondere una parola ad *n bit* in un opportuno codice: molto usato è il codice BCD.

Il contatore è costituito da un numero "*n*" di FF di tipo T, opportunamente collegati tra loro, costituenti un modulo.

Ogni FF può assumere due stati opposti; ne viene che con "*n*" FF si possono ottenere 2^n parole diverse. Questo numero rappresenta il massimo conteggio consentito dal modulo del contatore.

Per aumentare la capacità di conteggio si possono collegare più moduli in cascata.

Il conteggio può essere effettuato "*verso l'alto*" (*Up*) con parole che indicano numeri crescenti; oppure "*verso il basso*" (*Down*) con parole che, partendo dal valore massimo impostato, scandiscono numeri decrescenti.

Vi sono contatori, detti bidirezionali, che permettono tutte e due le modalità di conteggio: *UP/DOWN*

I contatori possono essere di tipo *sincrono* o *asincrono*.

Nei contatori sincroni il segnale di *clock* viene inviato contemporaneamente a tutti i FF abilitando le uscite che determinano la parola corrispondente al conteggio. Nei contatori asincroni l'impulso di conteggio viene inviato al primo FF e questo con ritardo lo invia al secondo, e così via, in cascata fino all'ultimo.

8 UTILIZZO DI FUNZIONI BOOLEANE NEL FOGLIO ELETTRONICO

8.1 Variabili di tipo booleano – Operazioni booleane

Come noto la variabile booleana può assumere solamente due valori opposti:

VERO - FALSO

che corrispondono ai due valori 0 1 delle variabili di funzioni binarie.

Le due variabili possono essere immesse nella celle nella forma semplice, scrivendo: VERO o FALSO. Però, per garantire la compatibilità con altri programmi di fogli di calcolo, conviene usare, per le due variabili la seguente sintassi:

VERO() **FALSO()**

8.2 Operazioni sulle variabili booleane

Le operazioni sulle variabili booleane corrispondono a quelle operate sulle binarie: AND, OR, NOT, indicate in EXCEL, rispettivamente con: E(), O(), NON().

8.3 Funzione logica E()

Corrisponde alla funzione logica **AND**, la quale restituisce il valore logico **VERO** solamente quando tutti gli argomenti su cui opera hanno valore **VERO**; restituisce **FALSO** quando almeno uno degli argomenti ha valore **FALSO**.

Sintassi

L'operazione **E()** tra le variabili logiche V1, V2, V3 si scrive:

=E(V1;V2;V3;...)

L'operazione **E** tra due valori logici restituiscono i seguenti risultati:

=E(VERO;VERO) è uguale a **VERO**

=E(VERO;FALSO) è uguale a **FALSO**

=E(FALSO;VERO) è uguale a **FALSO**

=E(FALSO;FALSO) è uguale a **FALSO**

La funzione **E()** può operare su riferimenti di celle; così se si vogliono porre in AND i valori contenuti nelle due celle B5, B6 si scriverà:

=E(B5;B6)

Se alcuni argomenti dell'operazione **E** sono riferimenti che contengono testo o sono vuoti, essi vengono ignorati; se, invece, tutti gli argomenti sono vuoti l'operazione **E** restituisce il valore di **errore: #VALORE!**

Così supponiamo che l'intervallo B3:B6 contiene: **VERO, FALSO, VERO, un testo, e due celle vuote**, risulta:

=E(B3;B6) uguale a **FALSO**

Esempio

Si abbia la funzione logica

$$y_1 = x_1 \cdot x_2 \cdot x_3 \quad (a)$$

Dove le variabili possano assumere i due valori opposti: VERO, FALSO

Con

$VERO \equiv 1$

$FALSO \equiv 0$

Siano:

$x_1 = VERO$

$x_2 = FALSO$

$x_3 = VERO$

poste le variabili:

$x_1 \rightarrow C4$

$x_2 \rightarrow D4$

$x_3 \rightarrow E4$

la funzione (a) si esprimerà nel comando EXCELL
 $=E(C4;D4;E4)$

Fig.8.1

	A	B	C	D	E	F
1						
2				y1 = x1 * x2 * x3		
3			X1	X2	X3	y1
4			VERO	FALSO	VERO	=E(C4;D4;E4)
5						

si ottiene

X1	X2	X3	y1
VERO	FALSO	VERO	FALSO

5.2.2 Funzione logica O()

Corrisponde alla funzione logica **O**, la quale restituisce il valore logico **VERO** quando almeno uno degli argomenti su cui opera ha valore **VERO**; restituisce **FALSO** quando tutti gli argomenti hanno valore **FALSO**.

Sintassi

L'operazione **O()** tra le variabili logiche V1, V2, V3 si scrive:

$=O(V1;V2;V3;...)$

L'operazione **O()** tra due valori logici restituiscono i seguenti risultati:

$=O(VERO;VERO)$ è uguale a **VERO**

$=O(VERO;FALSO)$ è uguale a **VERO**

$=O(FALSO;VERO)$ è uguale a **VERO**

$=O(FALSO;FALSO)$ è uguale a **FALSO**

La funzione **O()** può operare su riferimenti di celle; così se si vogliono porre in AND i valori contenuti nelle due celle B5, B8 si scriverà:

$=E(B5;B8)$

Se alcuni argomenti dell'operazione **O** sono riferimenti che contengono testo o sono vuoti, essi vengono ignorati; se, invece, tutti gli argomenti sono vuoti l'operazione **O** restituisce il valore di **errore**: **#VALORE!**

Così supponiamo che l'intervallo B3:B6 contiene: **VERO, FALSO, VERO, un testo, e due celle vuote**, risulta:

=O(B3;B6) uguale a **VERO**

5.2.2 Funzione logica NON()

Corrisponde alla funzione logica NOT la quale inverte il valore logico dell'argomento

Sintassi

L'operazione **NON()** della variabile logica **V** si scrive:

=NON(V)

Risulta:

NON(VERO) uguale a **FALSO**

NON(FALSO) uguale a **VERO**

Nota

Con l'introduzione della variabile **NON()** una variabile negata verrà indicata con un simbolo soprassegnato ad esempio \bar{x}

Esempio

Sia data la funzione

$$y = \bar{x}_1 \cdot x_2 + x_3 \cdot \bar{x}_2 \quad (b)$$

tradurre la funzione in comandi EXCEL e determinare il suo valore nel caso che le variabili siano:

1 x_1 =VERO x_2 =VERO x_3 =FALSO

2 X_1 =FALSO x_2 =VERO x_3 =VERO

Fig.8.2

	A	B	C	D	E	F	G	H	I
1									
2			$y = \bar{x}_1 \cdot x_2 + x_3 \cdot \bar{x}_2$						
3									
4		x_1	x_2	x_3	y				
5		VERO	VERO	FALSO	=O(E(NON(B5);C5)+E(D5;NON(C5)))				

Conviene, come al solito porre i valori in corrispondenza delle variabili
In corrispondenza della variabile y in comandi EXCEL si traduce in

$$=O(E(NON(B5);C5);E(D5;NON(C5)))$$

Globalmente la funzione è un OR tra due AND:

$$O(E();E())$$

Il primo AND è tra **NON(B5)** e **C5**:

$$E(NON(B5);C5)$$

Il secondo AND è tra **D5** e **NON(C5)**:

$$E(D5;NON(C5))$$

Per i valori di cui ai punti 1 e 2 si ottiene:

Fig.8.3

	A	B	C	D	E
1					
2			$y = \bar{x}_1 \cdot x_2 + x_3 \cdot \bar{x}_2$		
3					
4		x1	x2	x3	y
5		VERO	VERO	FALSO	FALSO
6		FALSO	VERO	VERO	VERO

Infatti

$$\bar{1} \cdot 1 + 0 \cdot \bar{1} = 0$$

$$\bar{0} \cdot 1 + 1 \cdot \bar{1} = 1$$

5.2.3 Operatori di relazione

Risultano di tipo booleano gli operatori di relazione quali l'uguaglianza, la disuguaglianza, ecc. Essi sono:

- uguale =
- minore <
- maggiore >
- diverso <>
- minore o uguale <=
- maggiore o uguale >=

Gli operatori di relazione vengono usati per verificare delle condizioni su degli operandi dello stesso tipo. Si osservi che:

=E(1<>3) restituisce **VERO** Le funzioni **E()** ed **O()** operate su di un solo argomento restituiscono la veridicità di esso. Così **=E(1<>3)** restituisce **VERO** giacché risulta vero che 1 è diverso da 3

=E(1<3;4>5) restituisce **FALSO** Il primo argomento **1<3** è **VERO**, il secondo **4>5** è **FALSO**; l'operazione **E (AND)** tra i due restituisce il valore **FALSO**

5.2.3 Funzione informativa NUM

Le funzioni informative restituiscono una informazione sull'argomento o un valore di questo di tipo diverso¹.

La funzione informativa = **NUM()** restituisce un valore dell'argomento convertito in numero.

Sintassi

La conversione del **valore** dell'argomento in numero si scrive:

=NUM(valore)

valore da convertire	=NUM(valore) restituisce
Un numero	Quel numero
Una data	Il numero seriale
VERO	1
FALFO	0

Nelle funzioni binarie e nella soluzione di circuiti combinatori sono state utilizzate variabili binarie contrassegnate di due valori opposti 0 1, queste corrispondono alle variabili booleane di Excel **FALSO()** **VERO()**. Può quindi essere utile impiegare la funzione =**NUM()** per

¹ Per un approfondimento sull'argomento consultare i manuali specializzati

convertire, nella rappresentazione di un circuito combinatorio o sequenziale, **VERO** con 1 e **FALSO** con 0.

5.2.2 Funzione logica SE()

La funzione logica =SE() pone come argomento un condizionale, che a seconda sia o no verificato restituisce due valori diversi; precisamente:

- se è verificata la condizione, la funzione logica =SE() restituisce un determinato valore specificato;
- se la condizione non è verificata la funzione logica =SE() restituisce, o un determinato valore alternativo specificato, oppure, nel caso che non lo sia, il valore **FALSO**.
- Se, dopo il condizionale, è specificato un solo valore, allora questo verrà restituito nel caso che la condizione sia vera, altrimenti sarà restituito il valore **FALSO**

Sintassi

Dopo il **SE** si pone entro parentesi, in ordine: il test condizionale, il punto e virgola (;), il valore restituito dalla condizione vera, il punto e virgola (;),(;), il valore restituito dalla condizione falsa.

=SE(test condizione; valore_vero; valore_falso)

Il **test** è costituito da una relazione booleana di confronto espressa con gli operatori “= < >ecc.”

I valori specificati dopo il test possono essere: valori numerici - il valore di una cella – un calcolo –, una operazione logica – un testo.

Il valore corrispondente ad un testo va posto tra le doppie virgolette: “**TESTO**”

Facciamo degli esempi.

- Supponiamo che nella cella A11 viene immesso la misura del livello di un recipiente che deve mantenersi al valore di riferimento di 100 litri. Nella cella A13 deve comparire il messaggio: LIVELLO INSUFFICIENTE, se il valore di A11 è inferiore di 100; se, invece, il valore di A11 è superiore o uguale a 100 deve comparire il messaggio: SUPERATO IL LIVELLO. Nella cella A11 si scrive:

=SE(A11<100;"LIVELLO INSUFFICIENTE";"SUPERATO IL LIVELLO")

- Nella cella **B7** si vogliono tramutare i caratteri 1, 0 immessi in **B6** rispettivamente nei valori: **VERO FALSO**. Nella cella **B7** si scriverà:

=SE(B6=1;VERO);FALSO()

Oppure semplicemente:

=SE(B6=1;VERO)

Esempio di operatori di relazione e della funzione logica SE

Come esempio si propone di risolvere una equazione di secondo grado

$$a \cdot x^2 + b \cdot x + c = 0$$

le cui soluzioni dipendono dal segno del discriminante (che indicheremo con “delta”)

$$\text{delta} = b^2 - 4 \cdot a \cdot c$$

delta>0 si hanno due soluzioni x_1, x_2 reali e distinte

delta=0 si hanno due soluzioni $x_1 = x_2$ reali e coincidenti

delta<0 si hanno due soluzioni complesse e coniugate del tipo

$$x_1 = \text{real} - j \cdot \text{imm} ; x_2 = \text{real} + j \cdot \text{imm}$$

dove si è indicato con “j” l’immaginario

Costruzione foglio EXCEL

1

Si scrive la tabella contenente i coefficienti a,b,c dell’equazione di secondo grado, e il discriminante “delta”.

Conviene porre subito dei valori nelle celle riferite ai coefficienti, in modo da ottenere subito dei risultati in corrispondenza delle funzioni che si vanno digitando.

Poniamo:

a = 1 1 in C6
 b = -5 -5 in D6
 c = 6 6 in E6

		F6		=		=(D6^2-4*C6*E6)	
	A	B	C	D	E	F	G
1							
2	Soluzione di un equazione di 2° grado				$a \cdot x^2 + b \cdot x + c = 0$		
3							
4			a	b	c	delta	
5							
6			1	-5	6	1	
7							
8							
9							
10							

Nella cella F6 si scrive l’espressione del discriminante **delta**

$$\text{delta} = b^2 - 4 \cdot a \cdot c$$

tradotto in istruzione EXCEL:

$$b^2 - 4 \cdot a \cdot c \longrightarrow =(D6^2-4*C6*E6)$$

Nel caso particolare dei valori assegnati ai coefficienti si ottiene **delta=1**

2

Si costruisce la tabella riguardante il caso che il discriminante sia maggiore di zero

$$\text{Delta} > 0 \longrightarrow \text{F6} > 0$$

Si hanno due soluzioni reali e distinte x1,x2

		I6		=		=SE(F6>0;(-D6+F6^(1/2))/(2*C6);"")						
	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	Soluzione di un equazione di 2° grado				$a \cdot x^2 + b \cdot x + c = 0$			Soluzioni reali e distinte				
3												
4			a	b	c	delta		x1	x2			
5												
6			1	-5	6	1		2	3			
7												
8												
9												

Nelle celle H6, H7, rispettivamente si scrivono le istruzioni che determinano le soluzioni reali e distinte di x1 e x2

Soluzioni	Cella	Espressione discorsiva per la determinazione della soluzione	Istruzione in EXCEL
x_1	H6	SE delta>0 allora $x_1 = \frac{-b - \sqrt{\text{delta}}}{2 \cdot a}$ altrimenti scrivi la stringa vuota ""	=SE(F6>0;(-D6-F6^(1/2))/(2*C6);"")
x_2	I6	SE delta>0 allora $x_1 = \frac{-b + \sqrt{\text{delta}}}{2 \cdot a}$ altrimenti scrivi la stringa vuota ""	=SE(F6>0;(-D6+F6^(1/2))/(2*C6);"")

Analizziamo la traduzione della espressione $\frac{-b - \sqrt{\text{delta}}}{2 \cdot a}$ in istruzione EXCEL

Iniziamo dalla espressione $\sqrt{\text{delta}}$

$$\sqrt{\text{delta}} \longrightarrow \text{F6}^{(1/2)}$$

$$-b - \sqrt{\text{delta}} \longrightarrow -\text{D6} - \text{F6}^{(1/2)}$$

poi l'espressione deve essere divisa per il prodotto $2 \cdot a$. Quindi:

L'istruzione $-\text{D6} - \text{F6}^{(1/2)}$ va posta tra parentesi e divisa "/" per $(2 \cdot \text{C6})$ entro parentesi.

$$\frac{-b - \sqrt{\text{delta}}}{2 \cdot a} \longrightarrow \frac{(-\text{D6} - \text{F6}^{(1/2)})}{(2 \cdot \text{C6})}$$

Questa espressione è assegnata alla cella se è verificata la condizione $\text{F6} > 0$:

$$=\text{SE}(\text{F6} > 0; (-\text{D6} - \text{F6}^{(1/2)}) / (2 \cdot \text{C6}))$$

altrimenti è assegnata alla cella una stringa vuota, espressa da : ""

$$=\text{SE}(\text{F6} > 0; (-\text{D6} - \text{F6}^{(1/2)}) / (2 \cdot \text{C6}); "")$$

Nel caso particolare dei valori assegnati ai coefficienti si ottiene $x_1=2$; $x_2=3$

2

Si costruisce la tabella riguardante il caso che il discriminante sia uguale a zero

$$\text{Delta} = 0 \longrightarrow \text{F6} = 0$$

Si hanno due soluzioni reali e coincidenti $x_1=x_2$

Prima di costruire la tabella conviene assegnare ai coefficienti della equazione, valori che determinano soluzioni reali e coincidenti, in modo da verificare subito l'esattezza delle istruzioni immesse nelle celle. Così, si pongano:

$$\begin{array}{ll} a = 1 & 1 \text{ in C6} \\ b = 6 & 6 \text{ in D6} \\ c = 9 & 9 \text{ in E6} \end{array}$$

Nelle cella K6, si scrive l'istruzione che determina il valore delle due soluzioni $x_1=x_2$ coincidenti

F	G	H	I	J	K	L
$\cdot x + c = 0$		Soluzioni reali e distinte			Soluzioni reali e coincidenti	
delta		x1	x2		x1=x2	
					-3	
					=SE(F6=0;-D6/2/C6;"")	

Riga 6 →

Soluzioni	Cella	Espressione discorsiva per la determinazione della soluzione	Istruzione in EXCEL
$x_1 = x_2$	K6	SE delta=0 allora $x_1 = x_2 = \frac{-b}{2 \cdot a}$ altrimenti scrivi la stringa vuota ""	=SE(F6=0;-D6/2/C6;"")

Nel caso particolare dei valori assegnati ai coefficienti si ottiene $x_1=x_2=-3$

3

Si costruisce la tabella riguardante il caso che il discriminante sia maggiore di zero

$$\Delta < 0 \longrightarrow F6 < 0$$

Si hanno due soluzioni complesse e coniugate.

Si vuole che esse compaiano nella forma:

$$reale - j \cdot immaginario \quad reale - j \cdot immaginario$$

Si osservi che il discriminante **delta** è negativo. Si può scrivere nella forma:

$$-\sqrt{|\delta|}$$

le due soluzioni x_1, x_2 si presentano nelle espressioni

$$x_1 = \frac{-b - \sqrt{-|\delta|}}{2 \cdot a} \qquad x_2 = \frac{-b + \sqrt{-|\delta|}}{2 \cdot a}$$

la radice quadra si può esprimere:

$$\sqrt{-|\delta|} = \sqrt{-1 \cdot |\delta|} = \sqrt{-1} \cdot \sqrt{|\delta|}$$

$$\sqrt{-1} = j$$

quindi

$$x_1 = \frac{-b - j\sqrt{|\delta|}}{2 \cdot a} \qquad x_2 = \frac{-b + j\sqrt{|\delta|}}{2 \cdot a}$$

occorre porre entro radice il valore positivo di **delta**, ed essendo esso negativo occorre cambiarlo di segno e utilizzare le seguenti espressioni

$$x_1 = \frac{-b - j\sqrt{-\text{delta}}}{2 \cdot a}$$

$$x_1 = \frac{-b + j\sqrt{-\text{delta}}}{2 \cdot a}$$

$$x_1 = \frac{-b}{2 \cdot a} - j \frac{\sqrt{-\text{delta}}}{2 \cdot a}$$

$$x_1 = \frac{-b}{2 \cdot a} + j \frac{\sqrt{-\text{delta}}}{2 \cdot a}$$

per esprimere il numero complesso si utilizzano tre celle, e, rispettivamente, la prima a sinistra per il numero reale, la seconda al centro per il simbolo dell'immaginario "j" con segno ed la terza a destra per il numero immaginario.

Per la soluzione x1 sono impegnate le celle.

- M6 parte reale
- N6 simbolo immaginario con il segno -
- O6 parte immaginaria

J	K	L	M	N	O	P	Q	R	S	T
Soluzioni reali e coincidenti			Soluzioni complesse coniugate							
x1=x2			x1			x2				
Riga 6 →			2	-j	3	2 +j 3				
=SE(F6<0;(-D6)/(2*C6);"")			=SE(F6<0;"-j";"")			=SE(F6<0;(-F6)^(1/2)/(2*C6);"")				

Per una migliore rappresentazione del numero complesso conviene: restringere la cella dell'immaginario e selezionare l'allineamento "Al centro"; scegliere un allineamento "A destra" per il numero reale e "Sinistro" per l'immaginario.

Per la soluzione x1 sono impegnate le celle.

- P6 parte reale
- Q6 simbolo immaginario con segno +
- O6 parte immaginaria

L	M	N	O	P	Q	R	S	T	U	V	
Soluzioni reali e coincidenti			Soluzioni complesse coniugate								
x1			x2								
2 -j 3			2	+j	3						
=SE(F6<0;(-D6)/(2*C6);"")			=SE(F6<0;"+"j";"")			=SE(F6<0;(-F6)^(1/2)/(2*C6);"")					

Per le due variabili si hanno le seguenti istruzioni

Soluzioni	Parti	Cella	Espressione discorsiva	Istruzione EXCEL
x_1	numero reale	M6	SE delta<0 allora $\frac{-b}{2 \cdot a}$ altrimenti stringa vuota ""	=SE(F6<0;(-D6)/(2*C6);"")
	Simbolo immaginario	N6	SE delta<0 allora Stringa "- j" altrimenti stringa vuota ""	=SE(F6<0;"-j";"")
	Numero immaginario	O6	SE delta<0 allora $\frac{\sqrt{-\text{delta}}}{2 \cdot a}$ altrimenti stringa vuota ""	=SE(F6<0 ;(-F6)^(1/2)/(2*C6) ; » »)
x_2	numero reale	P	SE delta<0 allora $\frac{-b}{2 \cdot a}$ altrimenti stringa vuota ""	=SE(F6<0;(-D6)/(2*C6);"")
	Simbolo immaginario	Q	SE delta<0 allora Stringa "+ j" altrimenti stringa vuota ""	=SE(F6<0;" +j";"")
	Numero immaginario	R	SE delta<0 allora $\frac{\sqrt{-\text{delta}}}{2 \cdot a}$ altrimenti stringa vuota ""	=SE(F6<0;(-F6)^(1/2)/(2*C6);"")

Concludendo

1

Caso di due soluzioni reali e distinte

delta>0

si ponga:

$$\begin{aligned} a &= 1 \\ b &= -5 \\ c &= 6 \end{aligned}$$

$$x^2 - 5 \cdot x + 6 = 0$$

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1																			
2	Soluzione di un'equazione di 2° grado				$a \cdot x^2 + b \cdot x + c = 0$			Soluzioni reali e distinte		Soluzioni reali e coincidenti		Soluzioni complesse coniugate							
3																			
4			a	b	c	delta		x1	x2		x1=x2			x1			x2		
5																			
6			1	-5	6	1		2	3										
7																			

si ottiene:

$$\begin{aligned} x_1 &= 2 \\ x_2 &= 3 \end{aligned}$$

2

Caso di due soluzioni reali e coincidenti

Delta=0

si ponga:

$$\begin{aligned} a &= 1 \\ b &= 6 \\ c &= 9 \end{aligned}$$

$$x^2 + 6 \cdot x + 9 = 0$$

F10		=																	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1																			
2	Soluzione di un'equazione di 2° grado				$a \cdot x^2 + b \cdot x + c = 0$			Soluzioni reali e distinte		Soluzioni reali e coincidenti		Soluzioni complesse coniugate							
3			a	b	c	delta		x1	x2		x1=x2			x1				x2	
4																			
5			1	6	9	0					-3								
6																			
7																			
8																			
9																			

si ottiene:

$$x_1 = x_2 = -3$$

3

Caso di due soluzioni complesse coniugate

Delta<0

si ponga:

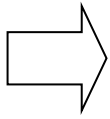
$$\begin{aligned} a &= 1 \\ b &= 6 \\ c &= 9 \end{aligned}$$

$$x^2 - 6 \cdot x + 13 = 0$$

G10		=																	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1																			
2	Soluzione di un'equazione di 2° grado				$a \cdot x^2 + b \cdot x + c = 0$			Soluzioni reali e distinte		Soluzioni reali e coincidenti		Soluzioni complesse coniugate							
3			a	b	c	delta		x1	x2		x1=x2			x1				x2	
4																			
5			1	-6	13	-16								3 - j 2				3 + j 2	
6																			
7																			
8																			
9																			

si ottiene:

$$x_1 = 3 - j2 \quad x_2 = 3 + j2$$



Esempi in laboratorio - Esercitazioni sulle funzioni logiche

L8.1 Traduzione di funzioni logiche nella tabella della verità – Mappa di Karnaugh

Siano date le due funzioni logiche:

$$\begin{cases} y = x_1 \cdot \bar{x}_2 + x_1 \cdot x_2 & (a) \\ y = \bar{x}_1 \cdot \bar{x}_2 + x_1 \cdot x_2 & (b) \end{cases}$$

Si vogliono eseguire la tabella della verità e la mappa di karnaugh

Nella figura Fig. 5.1 sono riportate le tabelle della verità e le mappe di Karnaugh riferite alle due funzioni, espresse con i valori logici **VERO FALSO**, tradotti poi nei simboli **1 0**. Analizziamo nel dettaglio la loro costruzione.

Consideriamo la prima equazione logica

(a)

Fig.8.4

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1																
2		TRADUZIONE DI UNA EQUAZIONE LOGICA NELLA TABELLA DELLA VERITA'														
3		MAPPA DI CARNAUGH														
4																
5		equazione $y = x_1 \cdot \bar{x}_2 + x_1 \cdot x_2$					$y = \bar{x}_1 \cdot \bar{x}_2 + x_1 \cdot x_2$									
6		O(E(B10;NON(C10));E(B10;C10))					O(E(NON(G10);NON(H10));E(G10;H10))									
7		X1	X2	Y			X1	X2	Y							
8		FALSO	FALSO	FALSO			FALSO	FALSO	VERO							
9		FALSO	VERO	FALSO			FALSO	VERO	FALSO							
10		VERO	FALSO	VERO			VERO	FALSO	FALSO							
11		VERO	VERO	VERO			VERO	VERO	VERO							
12																
13																
14																
15																
16		X1	X2	Y			X1	X2	Y							
17		0	0	0			0	0	1							
18		0	1	0			0	1	0							
19		1	0	1			1	0	0							
20		1	1	1			1	1	1							
21			\bar{x}_1	x_1				\bar{x}_1	x_1							
22			0	1				0	1							
23		\bar{x}_2	0	0	1		\bar{x}_2	0	1	0						
24		x_2	1	0	1		x_2	1	0	1						
25																
26																

- Sotto le etichette delle due variabili x_1 x_2 si riportino tutte le combinazioni logiche possibili tra **VERO** e **FALSO**

X1	X2
FALSO	FALSO
FALSO	VERO
VERO	FALSO
VERO	VERO

- nella cella **D10**, corrispondente al valore della funzione operata sulle due celle adiacenti **B10**, **C10**, scrivere l'equazione logica, formulata con i **riferimenti relativi** ad esse e corrispondenti alle variabili binarie x_1 x_2

$$=O(E(B10;NON(C10));E(B10;C10))$$

La formula è complessivamente un **OR**: **O()**, tra due **AND**: **E()**:

- il primo **AND** è tra il valore della cella **B10** corrispondente a x_1 e il valore **negato NON(C10)** della cella **C10** corrispondente a x_2 : **E(B10;NON(C10))**
- il secondo **AND** è tra il valore della cella **B10** e **C10**: **E(B10;C10)**
- scritta la formula e pigiato l'invio, nella cella **D10** si ottiene il valore: **FALSO**, che è il risultato dell'operazione logica effettuata sulle celle **B10** **C10**
- con il metodo dello strisciamento copiare la formula sulle altre celle al di sotto di **D10**, ove si ottengono i valori della funzione logica y per le altre combinazioni delle due variabili binarie

X1	X2	Y
FALSO	FALSO	FALSO
FALSO	VERO	FALSO
VERO	FALSO	VERO
VERO	VERO	VERO

Si scriva ora la tabella della verità con i simboli binari **1 0**. Basta tradurre con la funzione **NUM()**: **VERO** nel valore in **1** e **FALSO** in **0**. Si proceda nella seguente maniera.

- Nelle celle adiacenti **B16**, **C16**, **D16** si scrivano, rispettivamente le etichette delle variabili: x_1 , x_2 , y
 - nella cella **B17**, al disotto di x , si scriva la formula che traduce il valore logico della cella **B10** nel valore numerico
- $$=NUM(B10)$$
- pigiato l'Invio viene tradotto il valore **FALSO** della cella **B10** nel numero **0**
 - con il metodo dello strisciamento copiare la formula nelle due caselle adiacenti a destra, e poi nelle due altre righe inferiori.

Si costruisca ora la mappa di Karnaugh

- Selezionare le 4 celle che costituiranno la mappa di Karnaugh: nell'esempio il gruppo **C24:D25**;
- sulla barra dei menu selezionare l'opzione **Formato**, quindi **Celle** e con l'opzione **Bordi** evidenziare i bordi della mappa;
- porre sui lati della mappa le indicazioni delle combinazioni degli stati logici delle variabili binarie x_1 x_2 , come rappresentato in figura *Fig.5.1*;
- scrivere, entro ogni cella della mappa, il riferimento a quella della tabella della verità (espressa con 1, 0) in cui è posto il valore della funzione logica y , corrispondente alla combinazione degli stati logici delle due variabili x_1 x_2 , rappresentata dalla cella della mappa stessa. Così nella prima cella **C24** della mappa, corrispondente alla combinazione 0,0 di x_1 x_2 , si scriverà il riferimento **=D17** della cella della tabella della verità in cui è riportato il corrispondente valore della funzione y . Specificatamente:

1. nella cella **C24**, corrispondente alla combinazione 00 si scrive \rightarrow **=D17**
2. nella cella **C25**, corrispondente alla combinazione 01 si scrive \rightarrow **=D18**
3. nella cella **D24**, corrispondente alla combinazione 10 si scrive \rightarrow **=D19**
4. nella cella **D25**, corrispondente alla combinazione 11 si scrive \rightarrow **=D20**

Occorre osservare che, le due tabelle della verità e la mappa di karnaugh così costruite possono essere utilizzate per qualsiasi altra funzione binaria tra due variabili x_1 x_2 : basta sostituire la formula della nuova funzione logica nella colonna y della prima tabella della verità, espressa nei valori VERO FALSO.

Così per esercitazione si costruisca la tabella della verità e la mappa di Karnaugh della seconda funzione logica

$$y = \bar{x}_1 \cdot \bar{x}_2 + x_1 \cdot x_2 \quad (b)$$

in aggiunta a quella già costruita per la prima e sfruttando questa. Si proceda nella seguente maniera:

Fig.8.5

6				
7		O(E(B10;NON(C10));E(B10;C10))		
8				
9		X1	X2	Y
10		FALSO	FALSO	FALSO
11		FALSO	VERO	FALSO
12		VERO	FALSO	VERO
13		VERO	VERO	VERO
14				
15				
16		X1	X2	Y
17		0	0	0
18		0	1	0
19		1	0	1
20		1	1	1
21				
22			x1	x1
23			0	1
24	x2	0	0	1
25	x2	1	0	1
26				
27				

- selezionare tutto il gruppo di celle comprendenti le tabelle della verità e la mappa di Karnaugh costruite per la prima funzione
- dopo aver portato il puntatore del mouse sull'area evidenziata, pigiare il tasto destro del mouse e selezionare **Copia** – più semplicemente pigiare i tasti di scelta rapida **Ctrl+ C**
- si provoca un'interruzione dei bordi dell'area evidenziata

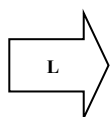
- fare clic sulla cella ove inizierà la copia del gruppo di celle da copiare: cella **F9**
- incollare con **Ctrl+ V** o con l'opzione **Incolla** del menu dinamico, selezionato con il tasto destro del mouse

Si ottiene così una copia delle tabelle e della mappa della prima funzione. Per ottenere quelle riferite alla seconda funzione, **basta sostituire la formula della nuova funzione logica nella colonna y della tabella della verità, espressa nei valori VERO FALSO.**

- Fare clic sulla cella **I10** (riferimento Fig. 5.1) ove deve essere immesso il valore della funzione logica y , corrispondente alla combinazione **FALSO, FALSO** delle due variabili binarie x_1 x_2
- Scrivere nella cella **I10** la formula della funzione logica (**b**)

$$=O(E(NON(G10);NON(H10));E(G10;H10))$$
- pigiare **Invio**
- con il metodo dello strisciamento copiare la formula sulle altre celle della funzione y : **I11, I12, I13.**

Dopo l'operazione di copia, automaticamente, si ottengono le tabelle della verità e la mappa di Karnaugh della funzione (**b**).



Esempi di circuiti combinatori

L.8.2 Con due pulsanti si debbono accendere in modo diverso tre lampadine L1, L2, L3:

1. pigiando il pulsante S1 si accendono le lampade L1 L2
2. pigiando il pulsante S2 si accendono le lampade L2 L3
3. pigiando contemporaneamente S2 S3 si accende solamente la lampada L3

Si effettui il circuito combinatorio

Per la soluzione del problema, i pulsanti S_1 , S_2 sono impiegati, rispettivamente, per l'eccitazione delle bobine X_1 , X_2 , i cui contatti x_1 , x_2 con la loro apertura o chiusura comandano l'accensione delle tre lampade.

$$x_1 = S_1 \quad x_2 = S_2$$

Le combinazioni degli stati logici dei contatti che comandano l'accensione delle lampade sono date dalla tabella della verità.

x_1	x_2	L1	L2	L3	Note
0	0	0	0	0	Nessun pulsante pigiato
0	1	0	1	1	Pigiato il pulsante S_2
1	0	1	1	0	Pigiato il pulsante S_1
1	1	0	0	1	Pigiati i pulsanti S_1 , S_2

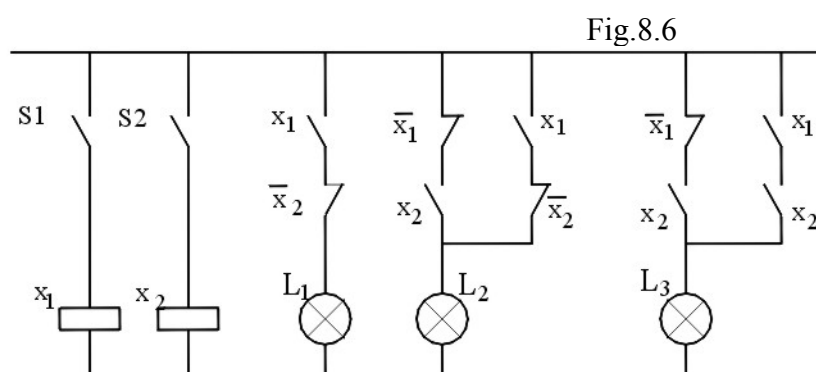
Dalla tabella della verità si estraggono le equazioni logiche d'accensione delle lampade:

$$\mathbf{L1} \text{ si accende per la combinazione di } x_1 \ x_2: \quad 1 \ 0 \quad \mathbf{L}_1 = x_1 \cdot \bar{x}_2$$

$$\mathbf{L2} \text{ si accende per le combinazioni di } x_1 \ x_2: \quad 0 \ 1 \ 1 \ 0 \quad \mathbf{L}_2 = \bar{x}_1 \cdot x_2 + x_1 \cdot \bar{x}_2$$

$$\mathbf{L3} \text{ si accende per la combinazione di } x_1 \ x_2: \quad 0 \ 1 \ 1 \ 1 \quad \mathbf{L}_3 = \bar{x}_1 \cdot x_2 + x_1 \cdot x_2$$

Dalle equazioni logiche si ricava il circuito rappresentato in figura *Excel5p3*



Si deve ora effettuare il foglio elettronico, con il quale si debbono ottenere i seguenti risultati:

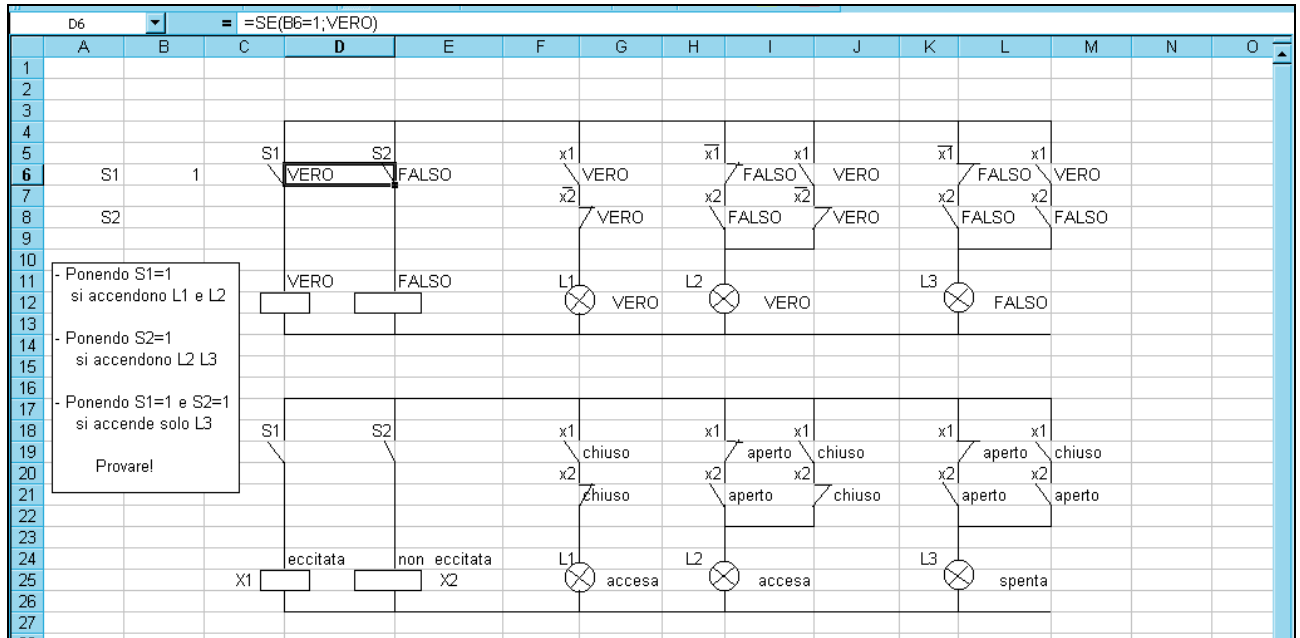
1. Ponendo il valore 1 in una cella rappresentante l'attivazione del pulsante S_1 , devono comparire i seguenti messaggi:
 - *accesa* nella cella rappresentante la lampada **L1**
 - *accesa* nella cella rappresentante la lampada **L2**
 - *spenta* nella cella rappresentante la lampada **L3**
2. Ponendo il valore 1 in una cella rappresentante l'attivazione del pulsante S_2 , devono comparire i seguenti messaggi:
 - *spenta* nella cella rappresentante la lampada **L1**
 - *accesa* nella cella rappresentante la lampada **L2**
 - *accesa* nella cella rappresentante la lampada **L3**
3. Ponendo il valore 1 nelle due celle rappresentanti l'attivazione dei pulsanti S_1 , S_2 , deve comparire i seguenti messaggi:
 - *spenta* nella cella rappresentante la lampada **L1**

- *spenta*
- *accesa*

nella cella rappresentante la lampada **L2**
 nella cella rappresentante la lampada **L3**

Si propone il foglio elettronico come riportato in figura *Fig. 5.4*

Fig.8.7



Si proceda nella seguente maniera.

- Attraverso la barra degli strumenti **Disegno**, si effettua sul foglio elettronico lo schema del circuito a contatti in corrispondenza dei quali verrà mostrato il loro stato logico **VERO – FALSO** a seconda dei valori 1 o no dati ai due pulsanti **S1 S2**; dallo stato logico dei contatti dipenderà quello delle lampade che verrà evidenziato;
- Con i comandi *copia – incolla* copiare, al disotto del circuito costruito, un altro nel quale, a seconda dei valori 1 o no dati ai due pulsanti **S1 S2**, compariranno i messaggi: “*aperto – chiuso*” in corrispondenza dei contatti; “*accesa – spenta*” in corrispondenza delle lampade; “*eccitata – non eccitata*” in corrispondenza delle bobine;
- Scrivere **S1** nella cella **A6** e **S2** nella cella **A8**, nelle cui caselle adiacenti a destra, rispettivamente **B6 B8**, verrà immesso o no il valore **1** per indicare l’attivazione del rispettivo pulsante;

Occorre ora, nel circuito superiore, nelle celle in corrispondenza delle indicazione dei *contatti*, delle *bobine* e delle *lampade*, immettere le formule che danno il loro stato logico a seconda dei valori 1 o no dati alle celle **A6 A8**, indicanti l’attivazione dei pulsanti **S1, S2**. Precisamente:

per i contatti	VERO – <i>corrisponde</i> → chiuso	FALSO – <i>corrisponde</i> → aperto
per le bobine	VERO – <i>corrisponde</i> → eccitata	VERO – <i>corrisponde</i> → non eccitata
per le lampade	VERO – <i>corrisponde</i> → accesa	VERO – <i>corrisponde</i> → spenta

- Per i contatti, occorre tramutare il valore 1 o no dato alla cella **B6** o **B8** in **VERO** o **FALSO**, attraverso la formula:

$$=SE(CELLA- PULSANTE=1;VERO)$$

- per le bobine, occorre imporre lo stesso stato logico della cella indicante il pulsante:
=CELLA- PULSANTE
- per le lampade, occorre porre nella cella corrispondente ad ognuna di esse l'equazione logica del circuito a contatti posto nella zona di teste di comando

Circuito di eccitazione della bobina X1

- Nella cella **D6** corrispondente al pulsante **S1** scrivere la formula =SE(B6=1;VERO)
- nella cella **D11** corrispondente alla bobina **X1** scrivere =D6

Circuito di eccitazione della bobina X2

- Nella cella **E6** corrispondente al pulsante **S1** scrivere la formula =SE(B8=1;VERO)
- nella cella **E11** corrispondente alla bobina **X1** scrivere =E6

Circuito di accensione della lampada L1

L'accensione della lampada **L1** è comandata dalla serie dei contatti $x_1 \cdot \bar{x}_2$. Lo stato logico di x_1 è lo stesso della bobina X1: si pone quindi nella cella **G6** lo stesso valore della cella **D11**. Lo stato logico di \bar{x}_2 è il negato della bobina X2: si pone quindi nella cella **G8** il valore negato della cella **E11**. Riassumendo:

- nella cella **G6** corrispondente al contatto x_1 si scrivere =D11
- nella cella **G8** corrispondente al contatto \bar{x}_2 si scrivere =NON(E11)

La funzione di comando della lampada **L1** è data dalla serie dei contatti $x_1 \cdot \bar{x}_2$, il cui valore logico è espresso, rispettivamente nelle due caselle **G6 G8**:

- Nella cella **G12** indicante la lampada **L1** scrivere la formula =E(G6;G8)

Circuito di accensione della lampada L2

L'accensione della lampada **L2** è comandata dalla funzione

$$L_2 = \bar{x}_1 \cdot x_2 + x_1 \cdot \bar{x}_2$$

Si tratta di effettuare il parallelo tra la serie di \bar{x}_1 con x_2 e la serie di x_1 con \bar{x}_2 . Riguardo agli stati logici dei contatti occorre notare che:

- quello di x_1 è lo stesso della bobina X1;
- quello di \bar{x}_1 è il negato della bobina X1;
- quello di x_2 è lo stesso della bobina X2;
- quello di \bar{x}_2 è il negato della bobina X2;

Ciò premesso, nelle celle corrispondenti ai contatti del parallelo si pongono le seguenti formule:

- nella cella **I6** corrispondente al contatto \bar{x}_1 si scrivere =NON(D11)
- nella cella **I8** corrispondente al contatto x_2 si scrivere = E11
- nella cella **J6** corrispondente al contatto \bar{x}_1 si scrivere = D11
- nella cella **J8** corrispondente al contatto x_2 si scrivere = NON(E11)

La funzione di comando della lampada **L2** è data dall'operazione **OR** tra l'**AND** degli stati logici delle celle **I6, I8** e l'**AND** degli stati logici delle celle **J6, J8**;

- Nella cella **I12** indicante la lampada **L2** scrivere la formula $=O(E(I6;I8);E(J6;J8))$

Circuito di accensione della lampada L3

L'accensione della lampada **L3** è comandata dalla funzione:

$$L_3 = \bar{x}_1 \cdot x_2 + x_1 \cdot x_2$$

Si procede alla stessa maniera eseguita per la lampada **L2**

L.8.3 Terzo esempio

Tema

Si debba verificare che, nel riempimento di una vasca , il livello dell'acqua sia tra un livello minimo controllato da un sensore S1 e un livello uno massimo, controllato da un sensore S2.

Vi siano due lampade una rossa L_r e una verde L_v .

- La lampada verde L_v sia accesa nel funzionamento normale, quando il livello dell'acqua sia tra il livello minimo dove è eccitato **S1** e quello massimo dove non è eccitato **S2** e spenta negli altri casi
- La lampada rossa L_r è un segnale di allarme, e si accende quando il livello è al disotto del minimo, quando non sono eccitati i due sensori **S1,S2**, o al disopra del massimo, quando sono eccitati tutti e due i sensori **S1,S2**. la lampada è spenta negli altri casi.

Associamo ai due sensori due variabili booleane X_1, X_2 che indicano: allo stato logico "1" l'eccitazione e allo stato "0" la diseccitazione, rispettivamente dei sensori S1,S2.

Tab.8.3.1

LIVELLO	SENSORE	Variabile associata
Livello massimo	S2	X_2
Livello minimo	S1	X_1

Da quanto detto si deve eseguire un circuito che esegua le seguenti accensioni delle lampade con le combinazioni delle eccitazioni dei sensori S1,S2 e corrispondenti valori delle variabili booleane X_1, X_2

Tab.8.3.2

Lampada	Eccitazione sensori S1,S2 per accensione lampada	Valori variabili X_1, X_2 per accensione lampada
Lampada verde L_v	S1 eccitato e S2 diseccitato	$X_1 = 1$ e $X_2 = 0$

Lampada rossa L_r	S1 diseccitato e S2 diseccitato Oppure S1 eccitato e S2 eccitato	$x_1 = 0$ e $x_2 = 0$ oppure $x_1 = 1$ e $x_2 = 1$
---------------------	--	--

Tabella della verità

Si può stendere la tabella della verità sia nei riguardi dell'accensione della lampada verde che per la rossa.

Tab.8.3.4 Tabella della verità

x_1	x_2	Stato Lampada verde L_v	Stato Lampada rossa L_r	Livello acqua
0	0	spenta	accesa	al di sotto del minimo
0	1	caso non possibile	caso non possibile	Non possibile
1	0	accesa	spenta	tra il minimo ed il massimo
1	1	spenta	accesa	sopra il massimo

Si noti che è impossibile che sia diseccitato il sensore del livello inferiore S1 ed eccitato quello superiore S2

Dalla tabella della verità si ricavano le funzioni logiche nelle variabili booleane x_1, x_2 , dell'accensione delle due lampade

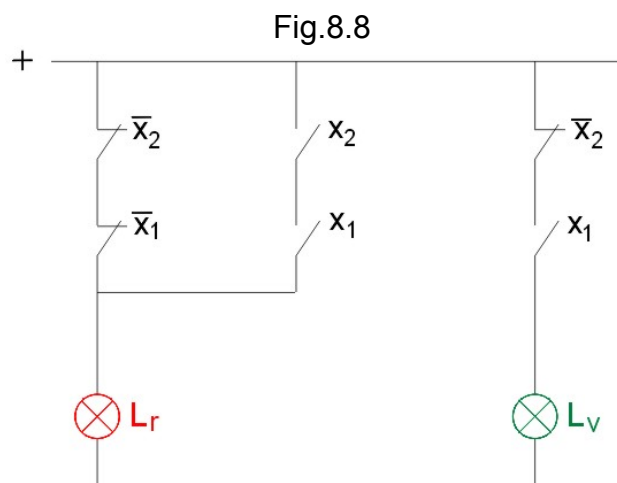
Funzione di accensione della lampada verde L_v :

$$L_v = x_1 \cdot \bar{x}_2 \quad (c)$$

Funzione di accensione della lampada rossa L_r :

$$L_r = \bar{x}_1 \cdot \bar{x}_2 + x_1 \cdot x_2 \quad (d)$$

Dalle equazioni logiche (c), (d) Si ottiene il seguente circuito a contatti



Foglio elettronico

Si voglia ora approntare il foglio elettronico, nel quale sia visualizzato il funzionamento del circuito, (nello stato dei contatti, e delle lampade), a seconda del livello dell'acqua nella vasca, segnalato dalla eccitazione o diseccitazione dei due sensori S1,S2 date, rispettivamente, dai valori booleani 1,0:

valore 0 sensore diseccitato
 valore 1 sensore eccitato

Si possono presentare i seguenti casi possibili

Tab.9.3.5

Livello acqua	Stato sensori	Valore booleano sensori	Stato dei contatti	Stato lampade
Al di sotto del minimo	S1 diseccitato S2 diseccitato	S1=0 S2=0	X_1 aperto \bar{X}_1 chiuso X_2 aperto \bar{X}_2 chiuso	L_v spenta L_r accesa
Tra minimo e massimo	S1 eccitato S2 diseccitato	S1=1 S2=0	X_1 aperto \bar{X}_1 chiuso X_2 aperto \bar{X}_2 chiuso	L_v accesa L_r spenta
Al di sopra del massimo	S1 eccitato S2 eccitato	S1=1 S2=1	X_1 aperto \bar{X}_1 chiuso X_2 aperto \bar{X}_2 chiuso	L_v spenta L_r accesa

Improntiamo ora, per passi successivi, il foglio elettronico Fig.

1

È riportata la tabella di assegnazione delle variabili booleane X_1, X_2 ai sensori S1,S2

Fig.8.9

	A	B	C
1			
2			
3	Livello	Sensore	Variabile associata
4			
5	Livello massimo	S2	X_2
6			
7	Livello minimo	S1	X_1

2

Si impronta la tabella per l'immissione dei valori booleani "0,1" dei sensori S1,S2, con il significato

0=diseccitato
 1=eccitato

Conviene immettere subito dei valori sulla colonna *Contatto sensori*

Fig.8.10

	E	F
Riga 9	S2	0
Riga 11	S1	1

F9=0, F11=1 che assegna S2=0; S1=1 corrispondente al livello dell'acqua tra il minimo ed il massimo

3

In corrispondenza degli stati 0,1 dei sensori, indicanti, rispettivamente, la loro diseccitazione o eccitazione, si hanno i seguenti valori booleani in EXCEL delle variabili associate $\bar{X}_1, X_1, \bar{X}_2, X_2$ e lo stato dei contatti che esse rappresentano

Stato sensore	Variabile	Valore booleano EXCEL	Stato del contatto
S1=0	\bar{X}_1	VERO	CHIUSO
	X_1	FALSO	APERTO
S1=1	\bar{X}_1	FALSO	APERTO
	X_1	VERO	CHIUSO
S2=0	\bar{X}_2	VERO	CHIUSO
	X_2	FALSO	APERTO
S2=1	\bar{X}_2	FALSO	APERTO
	X_2	VERO	CHIUSO

Per assegnare il valore booleano in EXCEL alla variabile x in corrispondenza del valore 0,1 del rispettivo sensore, si utilizza la funzione logica SE().

Così, dovendo assegnare alla cella H11 (Fig.8.11) il valore booleano EXCEL di \bar{X}_1 dipendente dal valore 0,1 della variabile S1 immesso nella cella F11 (Fig.8.10), si utilizza la funzione logica SE(), esprime la condizione che:

Se la cella **F11** (di S1) ha valore **1**, allora alla cella **H11** è assegnato valore **FALSO** altrimenti **VERO**

Ciò si esprime nella istruzione EXCEL:

$$=SE(F11=1;FALSO();VERO())$$

In una tabella si riportano i valori booleani “**VERO()**, **FALSO()**” in EXCEL assunti dalle variabili $\bar{X}_1, X_1, \bar{X}_2, X_2$ associate ai valori “0,1” assunti dai sensori S1, S2.

Fig.8.11

E	F	G	H	I	J	K	L	M	N	
			Valori delle variabili dipendenti dallo stato dei contatti dei sensori							
			\bar{X}_1	X_1	\bar{X}_2	X_2				
	RIGA 11									
		=SE(F11=1;FALSO();VERO())							=SE(F9=1;VERO();FALSO())	
			=SE(F11=1;VERO();FALSO())		=SE(F9=1;FALSO();VERO())					

Così nelle celle della riga 11, in corrispondenza delle variabili si scrive, rispettivamente:

Cella H11 =SE(F11=1;FALSO();VERO())

essendo una variabile negata, in corrispondenza a 1 assume il valore "FALSO". La cella F11 è quella ove si immette il valore booleano "0,1" del sensore S1

Cella I11 =SE(F11=1;VERO();FALSO())

essendo una variabile non negata, in corrispondenza a 1 assume il valore "VERO". La cella F11 è quella ove si immette il valore booleano "0,1" del sensore S1

Cella J11 =SE(F9=1;FALSO();VERO())

essendo una variabile negata, in corrispondenza a 1 assume il valore "FALSO". La cella F9 è quella ove si immette il valore booleano "0,1" del sensore S2

Cella L11 =SE(F9=1;VERO();FALSO())

essendo una variabile non negata, in corrispondenza a 1 assume il valore "VERO".

4

Ora, ottenuti dalla tabella i valori booleani VERO, FALSO in EXCEL delle variabili \bar{X}_1 , X_1 , \bar{X}_2 , X_2 dipendenti dallo stato dei sensori S1,S2 si possono scrivere le funzioni di accensioni delle lampade rossa L_r e verde verde L_v .

In una tabella Fig.8.12 si riporta, in corrispondenza delle celle riferite rispettivamente alla lampada rossa L_r e la lampada verde L_v , rispettivamente la loro funzione di accensione.

Fig.8.12

	A	B	C	D	E	F	G	H	I
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13		LAMPADA	Funzione di accensione	Funzione in EXCEL	Stato lampada				
14									
15		ROSSA	$L_r = \bar{x}_1 \cdot \bar{x}_2 + x_1 \cdot x_2$						
16									
17		VERDE	$L_v = x_1 \cdot \bar{x}_2$						
18									
19									
20									

Cellule con formule:

- Cella D11: `=O(E(H11;J11);E(I11;K11))`
- Cella E15: `=SE(D15=VERO;"ACCESA";"SPENTA")`
- Cella E17: `=E(I11;J11)`
- Cella F15: `=SE(D15=VERO;"ACCESA";"SPENTA")`
- Cella F17: `=SE(D17=VERO;"ACCESA";"SPENTA")`

Così:

Lampada rossa

Cella B15

etichetta: ROSSA

Cella C15

si indica l'espressione della funzione accensione:

$$L_r = \bar{x}_1 \cdot \bar{x}_2 + x_1 \cdot x_2^1 \quad (d)$$

Cella C16

si traduce la funzione di accensione in linguaggio EXCEL.

Tenendo conto delle corrispondenze delle variabili con le rispettive celle di appartenenza di loro valori, secondo Tab. di Fig.8.11

$\bar{x}_1 \longrightarrow$ H11

$x_1 \longrightarrow$ I11

$\bar{x}_2 \longrightarrow$ J11

$x_2 \longrightarrow$ K11

si ha

$$L_r = \bar{x}_1 \cdot \bar{x}_2 + x_1 \cdot x_2 \longrightarrow =O(E(H11;J11);E(I11;K11))$$

Cella C17

Si traduce il risultato booleano VERO o FALSO in EXCEL della funzione di accensione, rispettivamente in ACCESA, SPENTA

$$=SE(D15=VERO;"ACCESA";"SPENTA")$$

Lampada verde

Cella B17

etichetta: VERDE

Cella C17

si indica l'espressione della funzione accensione:

$$L_v = x_1 \cdot \bar{x}_2 \quad (c)$$

Cella C16

si traduce la funzione di accensione in linguaggio EXCEL.

Tenendo conto delle corrispondenze delle variabili con le rispettive celle di appartenenza di loro valori, secondo Tab. di Fig.8.11

$\bar{x}_1 \longrightarrow$ H11

$x_1 \longrightarrow$ I11

¹ L'espressione nella pagina EXCEL si può ottenere con copia- incolla da Equation Editor di Word.

$$\bar{X}_2 \longrightarrow J11$$

$$X_2 \longrightarrow K11$$

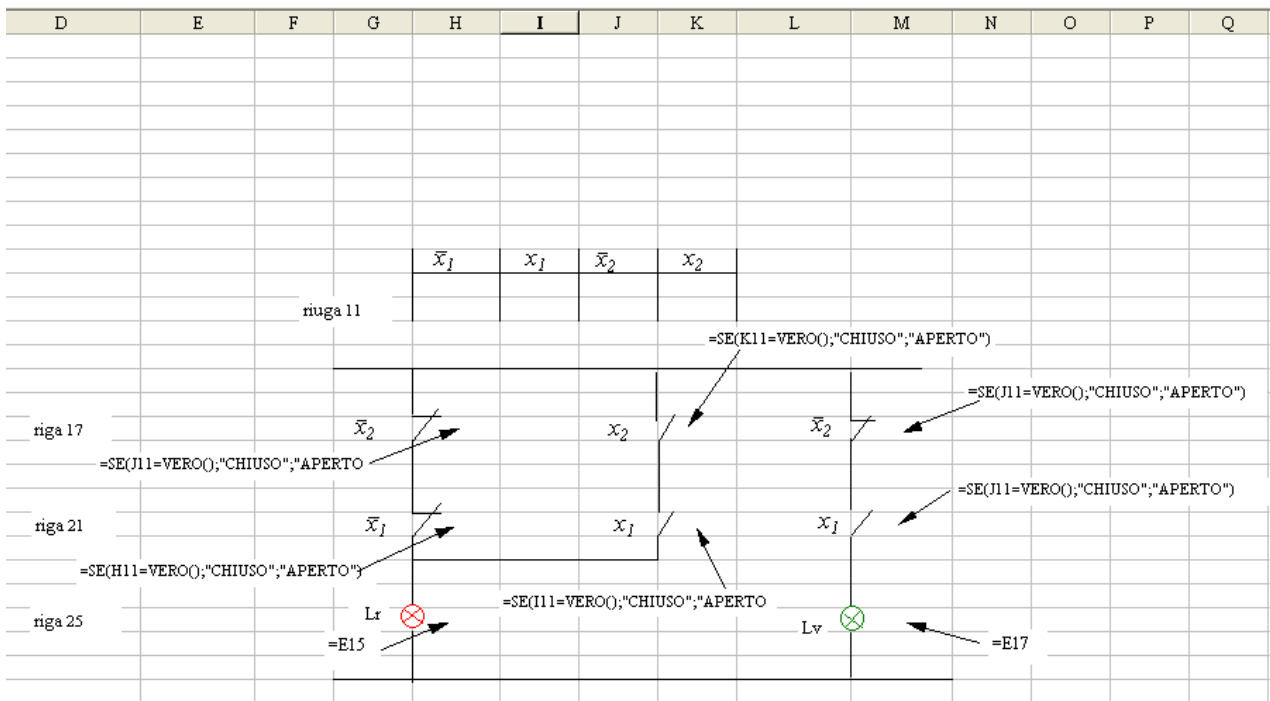
si ha

$$L_v = x_1 \cdot \bar{x}_2 \longrightarrow =E(I11;J11)$$

Cella C17 Si traduce il risultato booleano VERO o FALSO in EXCEL della funzione di accensione, rispettivamente in ACCESA, SPENTA
 $=SE(D17=VERO());"ACCESA";"SPENTA")$

5

Con gli elementi di disegno si rappresenti il circuito a contatti. Nelle celle accanto ad ogni contatto, a seconda dello stato 0,1 dei sensori dovrà comparire lo stato "APERTO" o "CHIUSO" del contatto stesso, e, accanto ad ogni lampada, lo stato "ACCESA" o "SPENTA".



Per far comparire lo stato "APERTO" o "CHIUSO" dei contatti nelle celle accanto ad essi, basta imporre con l'istruzione SE() che:

Se nella corrispondente cella del contatto della riga 11 della tab. Fig.8.11 si ha il valore **VERO** allora alla cella accanto al contatto si assegna la stringa **CHIUSO** altrimenti si assegna la stringa **APERTO**.

Così, consideriamo la cella H17 accanto al contatto \bar{X}_2 , il suo valore booleano corrispondente allo stato del sensore S2 è assegnato nella cella J11 di tab. Fig.8.11. Nella cella H17 si scriverà l'istruzione.

$$=SE(J11=VERO());"CHIUSO";"APERTO")$$

Nelle celle accanto ai contatti $\bar{X}_1, X_1, \bar{X}_2, X_2$ si scriveranno, rispettivamente le istruzioni:

- contatto \bar{X}_1 =SE(H11=VERO);"CHISO";"APERTO")
- contatto X_1 =SE(I11=VERO);"CHIUSSO";"APERTO")
- contatto \bar{X}_2 =SE(J11=VERO);"CHIUSSO";"APERTO")
- contatto X_2 =SE(K11=VERO);"CHIUSSO";"APERTO")

Le indicazioni di stringa "APERTA" o "CHIUSA" delle lampade rossa L_r e verde L_v sono state già assegnate rispettivamente nelle celle E15, E17. Così nelle celle accanto alle lampade si scriveranno le istruzioni

Lampada	Cella	Istruzione
rossa L_r	H25	=E15
verde L_v	M25	=E17

Completato il foglio verifichiamo il funzionamento.

Consideriamo il livello al disotto del minimo. Poniamo:

- S2=0 0 nella cella F9
- S1=0 0 nella cella F11

Si ottiene

The screenshot shows an Excel spreadsheet with the following content:

- Row 3:** Headers: Livello, Sensore, Variabile associata.
- Row 5:** Livello massimo, S2, X_2
- Row 7:** Livello minimo, S1, X_1
- Row 9:** Sensor S2, Contact 0, \bar{X}_1 , X_1 , \bar{X}_2 , X_2
- Row 11:** Sensor S1, Contact 0, VERO, FALSO, VERO, FALSO
- Row 13:** Headers: LAMPADA, Funzione di accensione, Funzione in EXCEL, Stato lampada
- Row 15:** ROSSA, $L_r = \bar{X}_1 \cdot \bar{X}_2 + X_1 \cdot X_2$, VERO, ACCESA
- Row 17:** VERDE, $L_v = X_1 \cdot \bar{X}_2$, FALSO, SPENTA
- Row 18-21:** Ladder logic diagram showing:
 - Branch 1: \bar{X}_2 / CHIUSO, X_1 / APERTO, \bar{X}_1 / CHIUSO
 - Branch 2: \bar{X}_2 / CHIUSO, X_1 / APERTO, X_1 / APERTO
- Row 25:** Lampada L_r d'allarme, ACCESA (with a red circle icon)
- Row 27:** Lampada L_v "Entro i livelli", SPENTA (with a green circle icon)

Compare la stringa "ACCESA" nella cella accanto alla lampada rossa L_r collegata al ramo chiuso dalla serie di contatti \bar{X}_2, \bar{X}_1 nelle cui celle adiacenti compare la stringa "CHIUSO". Mentre compare la stringa "SPENTA" nella cella accanto alla lampada verde L_v collegata al ramo aperto dalla serie di contatti \bar{X}_2, X_1 in cui risulta: il contatto

Lr



[Clic per continuare](#)

Avanti...



Indietro...

[clic per precedente](#)



[Clic per la pagina iniziale](#)

Indietro...