

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



[e-mail per suggerimenti](#)

---

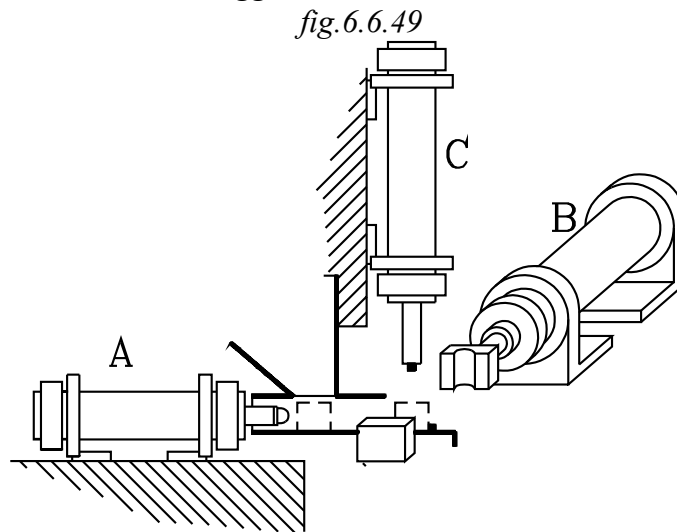
*IN LABORATORIO*

---

### 6.6.3 Tema

*Una serie di pezzi debbono subire una marchiatura. Essi vengono prelevati da un raccoglitore e spinti nel sistema di bloccaggio.*

*Bloccato il pezzo viene effettuata la marchiatura. Eseguita questa, si sblocca il pezzo che viene tolto a mano dal sistema di bloccaggio.*



#### 6.6.3.1 Specifiche del prodotto - soluzione del problema

Il pezzo depositato davanti all'estremità dello stelo del cilindro A, viene spinto nella posizione di bloccaggio.

Effettuata la corsa di ritorno dello stelo di A, viene bloccato il pezzo attraverso la corsa di fuoriuscita dello stelo di B. Bloccato il pezzo viene effettuata la doppia corsa del cilindro C per la marchiatura. Ultimata quest'ultima operazione viene sbloccato il pezzo con rientro dello stelo di B.

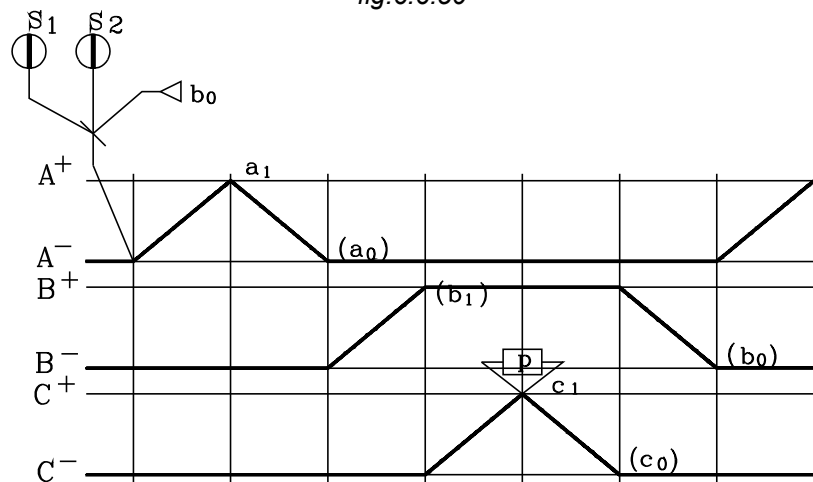
#### 6.6.3.2 Condizioni al Contorno - componenti utilizzati

- Vengono utilizzati tre cilindri a doppio effetto.
- Dovendo prelevare i pezzi a mano per sicurezza vengono adoperati due Start.
- La corsa di fuoriuscita dello stelo del cilindro A è moderata con l'impiego di un regolatore di flusso RF. Il rientro è rapido.
- La fuoriuscita dello stelo di B per il bloccaggio del pezzo deve avvenire con velocità moderata. Viene impiegato un regolatore di flusso Rf2. il rientro dello stelo nello sbloccaggio del pezzo può essere rapido.
- La fuoriuscita e il rientro dello stelo del cilindro C avvengono rapidamente.
- Viene assicurata la pressione di marcatura attraverso una valvola di soglia, in modo da sfruttare sia l'energia di velocità che una precisata pressione di soglia. Lo stelo effettua la corsa di rientro dopo che è stato toccato il finecorsa e la pressione sul pistone ha raggiunto il valore massimo.

Sigla componente	Codice	Componente
$A B C$		Cilindro a doppio effetto
$V_s$		Valvola di soglia
$Rf_2 Rf_1 Rf_3$		3 Regolatori di flusso unidirezionali
$Vb_1 Vb_2 Vb_3$		3 Valvole di potenza 5/2 bistabile
$X$		Valvola di memoria 5/2 bistabile
$a_0$		Fine corsa 3/2 a rullo e ritorno a molla
$a_1$		Fine corsa 3/2 a rullo e ritorno a molla
$b_0$		Fine corsa 3/2 a rullo e ritorno a molla
$b_1$		Fine corsa 3/2 a rullo e ritorno a molla
$c_0$		Fine corsa 3/2 a rullo e ritorno a molla
$c_1$		Fine corsa 3/2 a rullo e ritorno a molla
$S_1 S_2$		2 Start - valvola 3/2 NC monoastabile con comando manuale a tasto

### 6.6.3.3 Diagramma corsa passo

fig. 6.6.50

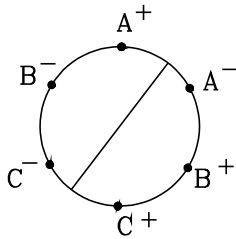


### 6.6.3.4 Tabella degli ordini

Stato comandato	ordine	variabile	posizione raggiunta
$A^+$	$A_1$	$a_0 b_0 c_0$	$\rightarrow a_1 b_0 c_0$
$A^-$	$A_0$	$a_1 b_0 c_0$	$\rightarrow a_0 b_0 c_0$
$B^+$	$B_1$	$a_0 b_0 c_0$	$\rightarrow a_0 b_1 c_0$
$C^+$	$C_1$	$a_0 b_1 c_0$	$\rightarrow a_0 b_1 c_1$
$C^-$	$C_0$	$a_0 b_1 c_1$	$\rightarrow a_0 b_1 c_0$
$B^-$	$B_0$	$a_0 b_1 c_0$	$\rightarrow a_0 b_0 c_0$

**6.6.3.5 Mappa di Karnaugh**

*fig.6.6.51*



Conviene porre la sequenza del ciclo su una circonferenza e dividere le fasi nel minimo numero di gruppi non comprendenti doppie corse dello stesso cilindro.

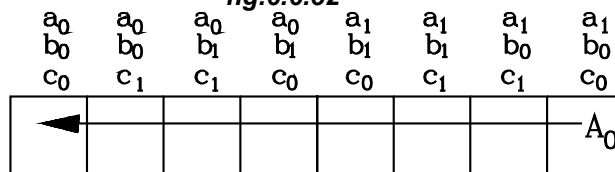
La sequenza viene divisa in due gruppi come rappresentato in figura.

$$A^- B^+ C^+ / C^- B^- A^+$$

Effettuata la mappa di karnaugh per tre cilindri, ponendo la combinazione dei finecorsa su di una stessa riga, si inizia a rappresentare i comandi da  $A_0$ .

Comando  $A_0$

*fig.6.6.52*

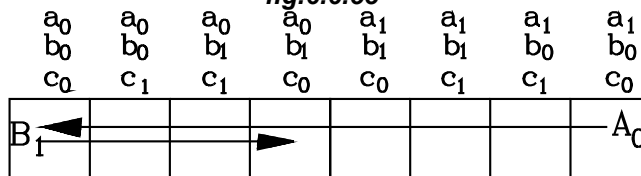


Il comando è rappresentato dal vettore congiungente le caselle:

$$a_1 b_0 c_0 \rightarrow a_0 b_0 c_0$$

Comando  $B_1$

*fig.6.6.53*

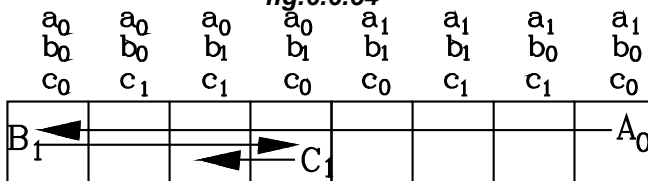


Il comando è rappresentato dal vettore congiungente le caselle:

$$a_0 b_0 c_0 \rightarrow a_0 b_1 c_0$$

Comando  $C_1$

*fig.6.6.54*

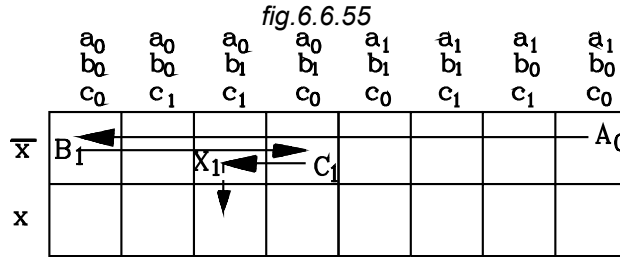


Il comando è rappresentato dal vettore congiungente le caselle:

$$a_0 b_1 c_0 \rightarrow a_0 b_1 c_1$$

Comando  $X_1$

Dopo il comando  $C_1$  dello stato  $C^+$  di fuoriuscita dello stelo del cilindro C, occorrerebbe dare il comando opposto di rientro  $C_0$  che condurrebbe però alla stessa casella. In tal modo vi sarebbero due comandi dati dalla stessa combinazione dei finecorsa. Ciò risulta impossibile.

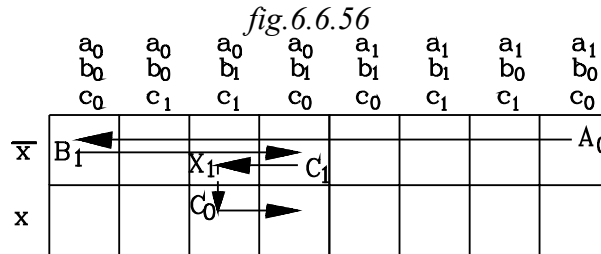


Occorre settare una memoria che dallo stato  $\bar{x}$  passa allo stato  $x$ .  
 Si deve introdurre un segnale di memoria che raddoppia la mappa di Karnaugh.  
 Si impartisce l'ordine  $X_1$  di settaggio della memoria. I finecorsa rimangono nello stesso stato mentre varia quella della memoria:  $\bar{x} \rightarrow x$ .

L'ordine di settaggio è rappresentato da un vettore tratteggiato che congiunge le caselle:

$$\bar{x}a_0b_1c_1 \rightarrow xa_0b_1c_1$$

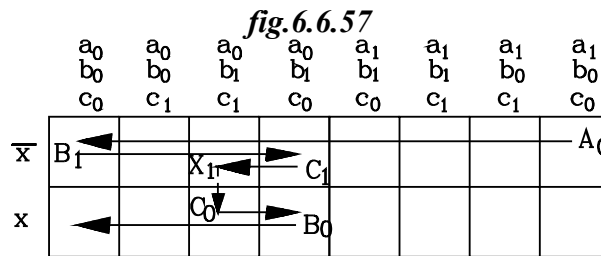
Comando  $C_0$



Il comando è rappresentato dal vettore congiungente le caselle:

$$xa_0b_1c_1 \rightarrow xa_0b_1c_0$$

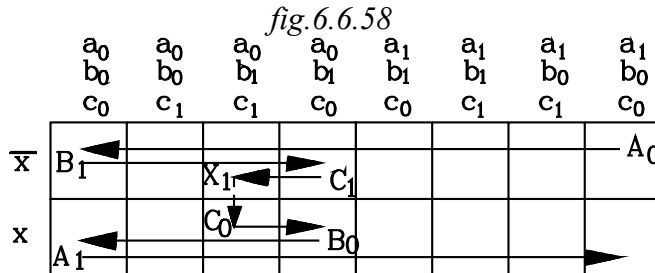
Comando  $B_0$



Il comando è rappresentato dal vettore congiungente le caselle:

$$xa_0b_1c_0 \rightarrow xa_0b_0c_0$$

Comando  $A_1$

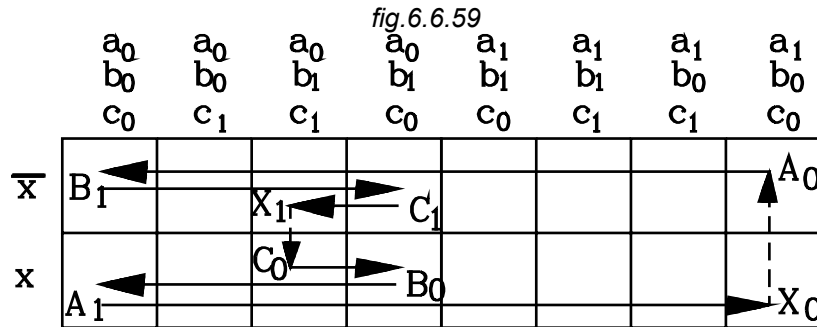


Il comando è rappresentato dal vettore congiungente le caselle:

$$xa_0b_0c_0 \rightarrow xa_1b_0c_0$$

**Ordine di resettaggio memoria  $X_0$**

Alla fine del ciclo viene resettata la memoria con l'ordine  $X_0$  che effettua il passaggio di stato  $x \rightarrow \bar{x}$ .



L'ordine di resettaggio della memoria è rappresentato da un vettore tratteggiato che congiunge le caselle:

$$xa_1b_0c_0 \rightarrow \bar{x}a_1b_0c_0$$

Si ritorna alla prima casella chiudendo il ciclo e ottenendo, così la mappa di Karnaugh completa.

**6.6.3.6 Estrazione delle equazioni logiche**

Per determinare le equazioni logiche occorre partire dalla prima fase del ciclo che effettivamente si deve effettuare.

Si deve partire di comando  $A_1$ .

Occorre sempre rispondere alle tre domande.

**Comando  $A_1$  - stato comandato  $A^+$**

- Lo precede  $B_0$ .
- Dallo stato precedente all'attuale compare la nuova variabile  $b_0$ .  
 $b_0$  è la variabile attiva.

$$A^+ = b_0$$

- Nell'area di  $b_0$  vi compaiono i comandi opposti  $A_1 A_0$ .  
Occorre porre la variabile  $x$  di memoria in AND con  $b_0$ .

Equazione logica

Sono previsti due Start da porre in AND per avviare il ciclo

$$A^+ = S_1 \cdot S_2 \cdot x \cdot b_0$$

**Ordine del resettaggio della memoria  $X_0$**

- Lo precede  $A_1$ .
- Dallo stato precedente all'attuale compare la nuova variabile  $a_1$ .  
 $a_1$  è la variabile attiva.

$$X_0 = a_1$$

- Nell'area di  $a_1$  non vi compaiono i comandi opposti  $X_1 X_0$ .  
La variabile attiva è anche attivante, sufficiente da sola a determinare l'ordine.

Equazione logica

$$X_0 = a_1$$

**Comando  $A_0$  -stato comandato  $A^-$**

- Lo precede  $X_0$ .

- Dallo stato precedente all'attuale compare la nuova variabile  $\bar{x}$ .  
 $\bar{x}$  è la variabile attiva.

$$A^- = \bar{x}$$

- Nell'area di  $\bar{x}$  non vi compaiono i comandi opposti  $A_1 A_0$ .  
La variabile attiva è anche attivante, sufficiente da sola a determinare il comando.

Equazione logica

$$A^- = \bar{x}$$

*Comando  $B_1$  - stato comandato  $B^+$*

- Lo precede  $A_0$ .
- Dallo stato precedente all'attuale compare la nuova variabile  $a_0$ .  
 $a_0$  è la variabile attiva

$$B^+ = a_0$$

- Nell'area di  $a_0$  vi compaiono i comandi opposti  $B_1 B_0$ .  
Occorre porre la variabile  $\bar{x}$  di memoria in AND con  $a_0$ .

Equazione logica

$$B^+ = \bar{x} \cdot a_0$$

*Comando  $C_1$  - stato comandato  $C^+$*

- Lo precede  $B_1$
- Dallo stato precedente all'attuale compare la nuova variabile  $b_1$ .  
 $b_1$  è la variabile attiva.

$$C^+ = b_1$$

- Nell'area di  $b_1$  vi compaiono i comandi opposti  $C_1 C_0$ .  
Occorre porre la variabile  $\bar{x}$  di memoria in AND con  $b_1$

Equazione logica

$$C^+ = \bar{x} \cdot b_1$$

*Ordine del settaggio della memoria  $X_1$*

- Lo precede  $C_1$
- Dallo stato precedente all'attuale compare la nuova variabile  $c_1$ .  
 $c_1$  è la variabile attiva.

$$X_1 = c_1$$

- Nell'area di  $c_1$  non vi compaiono i comandi opposti  $X_1 X_0$ .  
La variabile attiva è anche attivante, sufficiente da sola a determinare l'ordine.

Equazione logica

$$X_1 = c_1$$

*Comando  $C_0$  - stato comandato  $C^-$*

- Lo precede  $X_1$ .
- Dallo stato precedente all'attuale compare la nuova variabile  $x$ .  
 $x$  è la variabile attiva.

$$C^- = x$$

- Nell'area di  $x$  non vi compaiono i comandi opposti  $C_1 C_0$ .

La variabile attiva è anche attivante, sufficiente da sola a determinare il comando.

- Nelle specifiche del prodotto vi è la condizione di rientro dello stelo del cilindro solamente quando si è raggiunta la pressione massima  $p$ .  
Si impone come condizione al contorno dell'equazione logica, l'AND con la soglia  $p$  di pressione.

Equazione logica

$$C^- = x \cdot p$$

Comando  $B_1$  - stato comandato  $B^-$

- Lo precede  $C_0$
- Dallo stato precedente all'attuale compare la nuova variabile  $c_0$   
 $c_0$  è la variabile attiva

$$B^- = c_0$$

- Nell'area di  $C_0$  vi compaiono i comandi opposti  $B_1$   $B_0$   
Occorre porre la variabile  $x$  di memoria in AND con  $c_0$

Equazione logica

$$B^- = x \cdot c_0$$

Riassumendo, le equazioni logiche sono:

$$A^+ = S_1 \cdot S_2 \cdot x \cdot b_0$$

$$X_0 = a_1$$

$$A^- = \bar{x}$$

$$B^+ = x \cdot a_0$$

$$C^+ = \bar{x} \cdot b_1$$

$$X_1 = c_1$$

$$C^- = x \cdot p$$

$$B^+ = x \cdot c_0$$

$$B^- = x \cdot c_0$$

Dalle equazioni logiche si ricava il circuito pneumatico

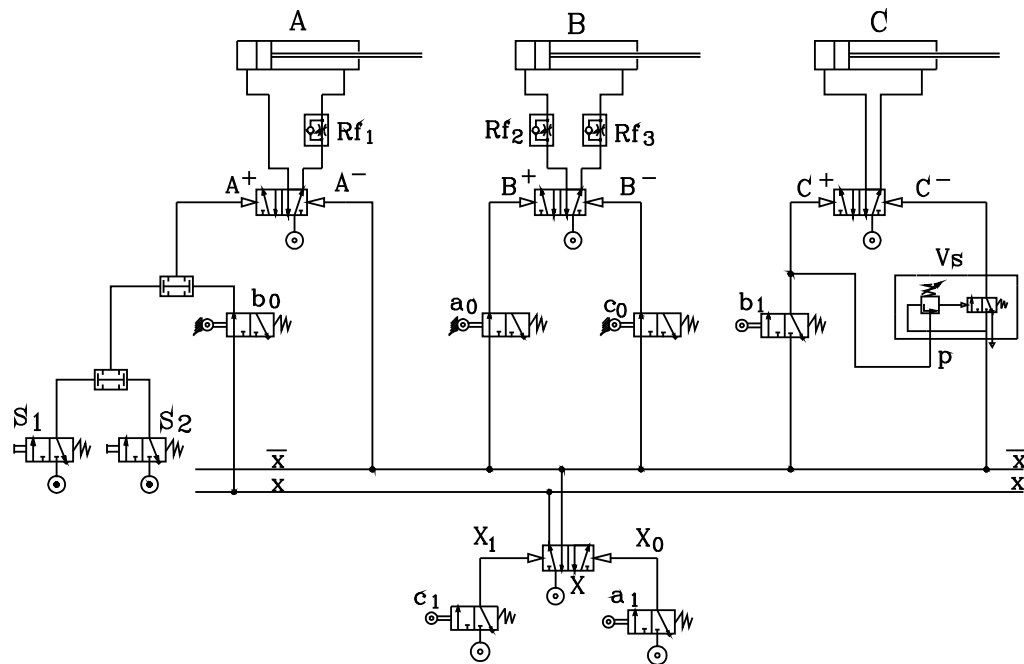
$A^+ = S_1 \cdot S_2 \cdot x \cdot b_0$  Si collegano i due start  $S_1$   $S_2$  agli ingressi della valvola AND. L'uscita di questa è collegata all'ingresso di un'altra valvola AND il cui altro ingresso è alimentato dal fincorsa  $b_0$  (toccato) che preleva l'aria dalla linea  $x$ .

$X_0 = a_1$  Il fincorsa  $a_1$  viene collegato con il pilotaggio  $X_0$  della memoria.

$A^- = \bar{x}$  Il pilotaggio  $A^-$  della valvola bistabile di potenza viene collegato con la linea  $\bar{x}$ .

$B^+ = \bar{x} \cdot a_0$  Il fincorsa  $a_0$  preleva l'aria dalla linea  $\bar{x}$  e la invia al pilotaggio  $B^+$  della valvola bistabile di potenza.

fig. 6.6.60



$C^+ = \bar{x} \cdot b_1$  Il finecorsa  $b_1$  preleva l'aria dalla linea  $\bar{x}$  e la invia al pilotaggio  $C^+$  della valvola bistabile di potenza.

$X_1 = c_1$  Il finecorsa  $c_1$  viene collegato con il pilotaggio  $X_1$  della memoria.

$C^- = x \cdot p$  L'aria viene prelevata dalla linea  $x$  ed inviata all'alimentazione  $P$  della valvola di soglia. Il pilotaggio della valvola di soglia è collegata con il condotto di mandata dell'aria nella camera del cilindro  $C$ , lato pistone. In tal modo la valvola di soglia si commuta quando la pressione in detta camere raggiunge il valore  $p$  prefissato

$B^+ = x \cdot c_0$  Il finecorsa  $c_0$  preleva l'aria dalla linea  $x$  e la invia al pilotaggio  $B^+$  della valvola bistabile di potenza.

#### Funzionamento

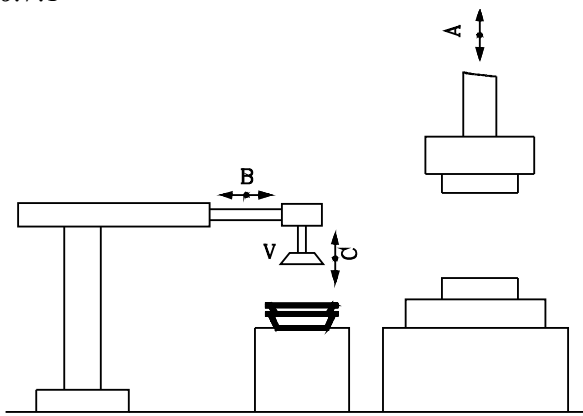
- Nelle condizioni iniziali occorre che la valvola di memoria 5/2 bistabile sia posta in seconda posizione, in modo da alimentare la linea  $x$ .
- Pigiat i due start  $S_1 S_2$ , questi alimentano la valvola AND. Da questa esce l'aria che va all'ingresso di un'altra valvola AND, il cui altro ingresso è alimentato dal finecorsa  $b_0$  toccato che preleva l'aria dalla linea  $x$ . L'aria in uscita dall'ultima valvola AND va al pilotaggio  $A^+$  determinando la corsa di fuoriuscita dello stelo del cilindro  $A$ .
- Toccato il finecorsa  $a_1$  viene commutata la valvola di memoria. La linea  $x$  va in scarico e viene alimentata la linea  $\bar{x}$ .
- L'aria prelevata con un tubicino dalla linea  $\bar{x}$  va al pilotaggio  $A^-$ , determinando la corsa di ritorno dello stelo di  $A$ .
- Toccato il finecorsa  $a_0$ , questo preleva l'aria dalla linea  $\bar{x}$  e la invia al pilotaggio  $B^+$ , determinando la corsa di fuoriuscita dello stelo di  $B$ .
- Toccato il finecorsa  $b_1$ , questo preleva l'aria dalla linea  $\bar{x}$  e la invia al pilotaggio  $C^+$ , determinando la corsa di fuoriuscita dello stelo di  $C$ .



- Toccato il finecorsa  $C_1$  viene commutata la valvola di memoria. La linea  $\bar{x}$  va in scarico e viene alimentata la linea  $x$ .
- L'aria prelevata da un tubicino dalla linea  $x$  va all'alimentazione  $P$  della valvola di soglia. Questa non si commuta fino a ch  non si raggiunge la pressione  $p$  nella camera di alimentazione del cilindro  $C$ , lato pistone.  
Raggiunta detta pressione la valvola di soglia si commuta e l'aria prelevata dalla linea  $x$  va al pilotaggio  $C^-$ , determinando la corsa di rientro dello stelo di  $C$ .
- Toccato il finecorsa  $b_0$ , questo preleva l'aria dalla linea  $x$  e la invia al pilotaggio  $B^-$ , determinando la corsa di rientro dello stelo di  $B$ .

## 6.7 Temi proposti

### 6.7.1



In una operazione di imbutitura, il pezzo da imbutire viene caricato a mano.

Pigiati i due Start viene effettuato dal cilindro A la doppia-corsa di imbutitura.

Tornato il cilindro A nella posizione superiore di riposo, il braccio B sposta la ventosa V sull'asse dell'imbutito.

Raggiunto l'asse dell'imbutito, la ventosa, comandata dal cilindro C, effettua la doppia corsa di andata (presa del pezzo imbutito) e ritorno nella posizione superiore di riposo.

Al raggiungimento di tale posizione il cilindro B si ritrae e porta la ventosa con l'imbutito preso sul mucchio, ove verr  lasciato cadere.

cilindro B si ritrae e porta la ventosa con l'imbutito preso sul mucchio, ove verr  lasciato cadere.

### Condizioni al contorno

- Deve essere assicurata la pressione per l'imbutitura nella camera di fuoriuscita stelo del cilindro A.
- La velocit  nella corsa di imbutitura deve essere controllata.
- Occorrono due Start per impegnare ambo le mani.
- Prevedere un sistema di emergenza in modo che quando   attivato non vengano esclusi gli Start e il cilindro A Si ponga in posizione di rientro.

Risolvere il circuito pneumatico attraverso le memorie in cascata - Effettuare una appropriata relazione

### 6.7.2

Risolvere il precedente problema mediante il metodo delle mappe di Karnaugh

### 6.7.3

Su ogni dischetto di una serie si deve eseguire una punzonatura.

I dischetti vengono montati a mano sull'apposito porta pezzo e vengono bloccati attraverso la corsa di un cilindro A. Un cilindro B, sul cui stelo   montato il punzone, effettua la doppia corsa di punzonatura. Eseguita questa, il cilindro A effettua la corsa di ritorno liberando

*il dischetto dal bloccaggio. Alla fine un cilindro C effettua una doppia corsa per espellere il dischetto dal portapezzo.*

*Risolvere il circuito pneumatico attraverso le memorie in cascata - Effettuare una appropriata relazione.*

6.7.4

*Risolvere il precedente problema mediante il metodo delle mappe di Karnaugh.*

6.7.5

*Effettuare i cicli:  $A^- B^+ C^+ C^- A^+ B^-$  -  $A^- B^+ C^- C^+ B^- A^+$  con il metodo delle memorie in cascata e delle mappe di Karnaugh.*

---

## 7 SEQUENZIATORI

In presenza di segnali bloccanti, come si è constatato, occorre adoperare un certo numero di memorie ausiliarie.

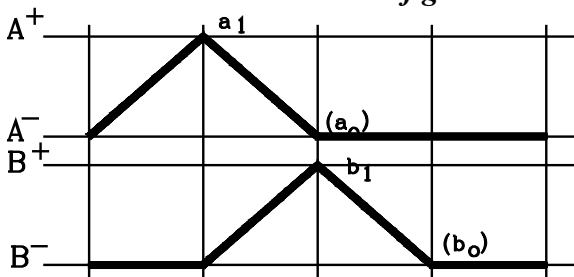
Minimizzando le memorie, occorre aumentare la combinazione dei segnali che contraddistinguono i vari passi. Per diminuire il numero di segnali per passo occorre aumentare il numero di memorie. Al limite, *adoperando una memoria ausiliaria per ogni passo, questo verrà contraddistinto dalla sola variabile attiva in AND con la memoria aggregata e le eventuali condizioni al contorno.*

Nella catena sequenziale si utilizza una memoria per ogni passo, e se  $n$  sono il numero di memorie, in ogni passo una sola memoria alla volta è nella condizione logica 1.

Occorre inoltre, in ogni passo, settare la memoria aggregata al passo stesso e resettare la memoria precedente.

Il passo successivo non deve essere effettuato fintantoché non torna il segnale, che indica l'avvenuta effettuazione del comando impartito nel passo attuale.

Consideriamo il ciclo  $A^+ A^- B^+ B^-$   
*fig. 7.1*



Adoperando un sequenziatore ad ogni passo viene associata una memoria. Nella tabella della verità, riportata qui di seguito, vengono impiegate le 4 memorie  $x, y, z, u$ , associate rispettivamente ai quattro passi. Nell'ultima colonna sono riportate le istruzioni di settaggio e resettaggio delle memorie.

Passo	$a_0$	$a_1$	$b_0$	$b_1$	$x$	$y$	$z$	$u$	Istruzione
1	1	0	1	0	1	0	0	0	set A
2	0	1	0	0	0	1	0	0	reset A
3	1	0	1	0	0	0	1	0	set B
4	1	0	0	1	0	0	0	1	reset B
5=1	1	0	1	0	1	0	0	0	set A

Con l'introduzione di una memoria per passo, l'equazione logica in questo si riduce alla variabile attiva in AND con la memoria ausiliaria ad esso associata e alle condizioni al contorno:

$$A^+ = b_0 \cdot x \cdot Start$$

$$A^- = a_l \cdot y$$

$$B^+ = a_0 \cdot z$$

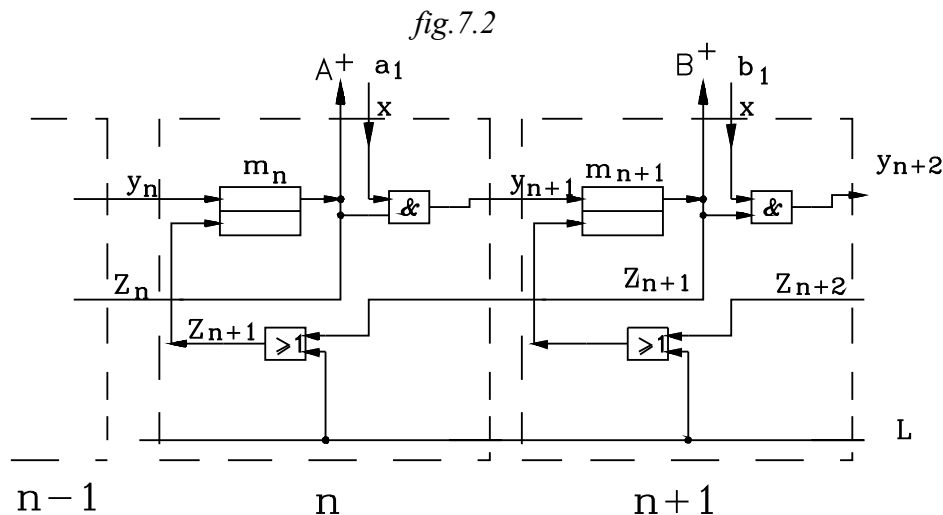
$$B^- = b_l \cdot u$$

### 7.1 Principio di funzionamento del sequenziatore

Ci si riferisca al ciclo  $A^+ B^+ B^- A^-$  già più volte studiato, contenente un segnale bloccante se non si utilizzano le memorie.

Con l'utilizzo di una memoria diversa per passo, non si presenterà mai nel ciclo un segnale bloccante, in quanto, i segnali di comando differiscono almeno per il segnale di memoria differente uno dall'altro.

Il principio di funzionamento del sequenziatore è rappresentato dallo schema logico di figura, nel quale sono rappresentati due passi successivi generici, indicati con  $n$ ,  $n+1$ . Lo schema logico del passo  $n-1$  non è rappresentato in fig.



Per comprendere il funzionamento del sequenziatore, si supponga che i due moduli successivi " $n$ ,  $n+1$ " debbano comandare la sequenza  $A^+ B^+$ .

Occorre tenere presente i seguenti punti:

- Il sequenziatore è modulare.
- Ogni modulo è utilizzato per effettuare un passo.
- In ogni passo è utilizzata una memoria, alla quale è inviato un segnale di Set  $y_n$ , proveniente dal modulo precedente  $n-1$ , e un segnale di Reset  $Z_{n+1}$ , proveniente dal modulo successivo  $n+1$ .
- Il segnale di uscita dalla memoria  $m_n$  si dirama, e viene inviato:
  - 1°- All'organo comandato: *nell'esempio, la valvola distributrice che comanda  $A^+$* .
  - 2°- All'ingresso dell'AND, per l'elaborazione del segnale di attivazione del modulo successivo  $n+1$ .

- 3°- All'ingresso della memoria del modulo precedente  $n-1$  e costituisce il segnale  $Z_n$  di resettaggio di essa.

Il segnale di comando di uscita dalla memoria  $m_n$ , come si è detto, va all'*AND*, ma non esce da esso per elaborare il segnale  $y_{n+1}$  che attiva il modulo successivo  $n+1$ , fintantoché non perviene all'*AND* stesso, dall'ingresso  $x$ , il segnale di ritorno (di retroazione) che indichi l'avvenuto comando: *nell'esempio, lo stelo tocchi il fine-corsa  $a_1$ .  $x=a_1$*

Così, nell'esempio considerato, pervenuto nel modulo  $n$  il segnale  $y_n$  di comando di  $A^+$ , in uscita dalla memoria  $m_n$  si ha un segnale che comanda  $A^+$ , resetta la memoria precedente, e si porta all'ingresso dell'*AND*.

Quando lo stelo ha effettuato la corsa di andata e ha toccato  $a_1$ , solamente allora, si ha il segnale di ritorno " $x=a_1$ ", che, in *AND* con il comando " $A^+$ ", determina il segnale d'uscita  $y_{n+1}$  dal modulo  $n$ , di comando del modulo successivo  $n+1$ : comando  $B^+$ .

Il comando  $y_{n+1}$  va all'ingresso della memoria  $m_{n+1}$ . In uscita di essa si ha il segnale che si dirama e: comanda  $B^+$ , va all'ingresso dell'*AND*, e determina il segnale  $Z_{n+1}$  di resettaggio della memoria precedente  $m_n$ .

Il segnale  $Z_{n+1}$  viene inviato all'ingresso dell'operatore logico *OR* del modulo  $n$ . L'uscita dall'*OR* va a resettare la memoria  $m_n$ . Cosicché, settata la memoria  $m_{n+1}$  del modulo  $n+1$ , ed elaborato in questo il segnale che comanda l'attuatore, viene resettata la memoria precedente  $n$ .

All'ingresso di tutti gli *OR* può essere inviato il segnale di preset "L". Questo segnale serve per resettare tutte le memorie, salvo l'ultima (*vedi oltre*), prima di avviare il ciclo.

Per effettuare un ciclo con il sequenziatore occorre utilizzare tanti moduli quanti sono i passi del ciclo. Tutti i moduli sono uguali salvo l'ultimo, differente dagli altri per la diversa posizione dell'operatore logico *OR* (*vedi oltre*).

Riassumendo brevemente i segnali che si avvicendano in un modulo  $n$  per il suo funzionamento:

- 1- Perviene all'ingresso della memoria  $m_n$  del modulo  $n$  il segnale di comando  $y_n$  dal modulo precedente ( $n-1$ )
- 2- Dalla memoria  $m_n$  esce il segnale che va a comandare il moto dello stelo del cilindro (*esempio  $A^+$* ), va all'ingresso dell'*AND* ove staziona e va ( $Z_n$ ) all'ingresso dell'*OR* del modulo precedente, la cui memoria viene resettata dal segnale di uscita dall'*OR* stesso. *In tal modo, quando si setta un modulo viene resettato il precedente.*
- 3- Il segnale che staziona nell'*AND* non determina quello di uscita da questo, fino a che non arriva, attraverso l'ingresso  $x$ , il segnale che certifica l'avvenuto comando: nell'esempio attivato il finecorsa  $a_1$
- 4- Pervenuto all'ingresso dell'*AND* il segnale che certifica l'attuazione del comando, si ha in uscita dal modulo il segnale  $y_{n+1}$  di comando del modulo successivo.

- 5- Il segnale di comando  $y_{n+1}$  setta la memoria del modulo successivo  $n+1$  e da questa esce il segnale  $Z_{n+1}$  che entra nell'OR e va a resettare la memoria del modulo  $n$ .

### Realizzazione pratica del modulo generico

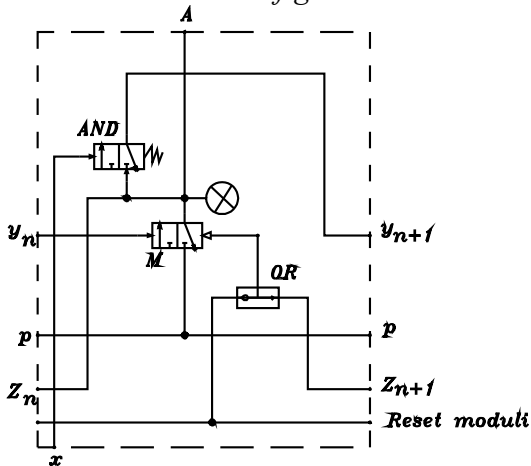
Si fa riferimento al modulo realizzato dalla ditta *FESTO*.

Il modulo viene indicato con la sigla *TAA*.

In detto modulo l'operatore logico *AND* è costituito da una valvola 3/2 monostabile con i due ingressi posti rispettivamente nel pilotaggio e nella alimentazione *P*. Si avrà aria in uscita solamente se sono presenti i due segnali.

### 7.2.1 MODULO TAA

fig.7.3



tutti i moduli generici.

La memoria è costituita da una *valvola 3/2 bistabile NC*, il cui segnale di uscita si dirama e viene inviato all'alimentazione  $P_1$  dell'*AND*, all'uscita *A* del modulo verso l'organo comandato, e, inoltre, lo stesso segnale costituisce il resettaggio  $Z_n$  del modulo precedente.

Il segnale  $x$ , che va al pilotaggio della valvola *AND*, è il segnale di ritorno dall'organo comandato, e indica l'avvenuto comando e che, quindi, si può proseguire per iniziare il passo successivo.

All'ingresso dell'*OR* del modulo  $n^{\text{mo}}$  viene inviato il segnale  $Z_{n+1}$  proveniente dal modulo successivo per il resettaggio della memoria; l'altro ingresso dell'*OR* è collegato con la linea di resettaggio "*L*" di

Così, immesso il segnale  $y_n$ , proveniente dal modulo precedente, questo commuta la memoria *M*, e l'aria, prelevata dalla linea "*p*" di alimentazione, uscendo dalla valvola *M*, si dirama andando: nell'uscita "*A*" del modulo attuale, all'ingresso della valvola *AND* e, attraverso il condotto  $Z_n$ , a resettare il modulo precedente.

Quando da  $x$  proviene il segnale di avvenuto comando, inviato da un sensore posto sull'attuatore, allora la valvola 3/2 funzionante da porta *AND* si commuta e da essa esce l'aria costituente il segnale  $y_{n+1}$  che attiverà il modulo successivo.

Attivato il modulo successivo, dalla sua memoria uscirà il segnale  $Z_{n+1}$  che perverrà all'*OR* per il resettaggio del modulo attuale  $n$ .

Una sequenza di tali moduli *TAA* permette di avere un segnale attivo alla volta, con resettaggio del precedente; inoltre, non viene attivato il modulo successivo fintantoché non arriva al modulo attuale il segnale di avvenuto comando, inviato da un sensore posto sull'attuatore.

Nelle condizioni iniziali tutti i moduli *TAA* devono essere resettati. Per ottenere ciò si invia un segnale di preset nella linea "*L*": l'aria in uscita da tutti gli *OR* portano le valvole di memoria 3/2 in posizione di interdizione dell'ingresso verso l'uscita.

### 7.7.2 Schema logico dell'ultimo modulo

L'ultimo modulo deve essere diverso dagli altri. Esso, nelle condizioni iniziali, prima che si avvii il ciclo, deve fornire in uscita il segnale  $y_{n+1}$ , da porre in *AND* con lo Start. I segnale di uscita dall'*AND* deve essere inviato al primo modulo per settare la sua memoria.

Perché abbia inizio il primo passo deve pervenire al primo modulo il segnale  $y_{n+1}$  che indichi l'avvenuta attivazione dell'ultimo.

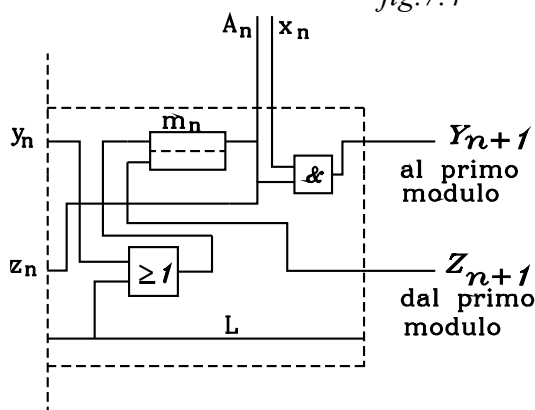
Prima di iniziare il ciclo occorre che tutti i moduli precedenti all'ultimo abbiano la memoria resettata, in modo da non fornire segnali in uscita. L'ultimo modulo, invece, nelle condizioni iniziali, deve fornire il segnale  $y_{n+1}$  in uscita da inviare al primo modulo.

Per settare la memoria dell'ultimo modulo e resettare tutte quelle degli altri moduli, si invia, prima di iniziare il ciclo, un impulso nella la linea  $L$ , alla quale sono collegati gli ingressi degli operatori OR.

Nei moduli precedenti all'ultimo, l'uscita *dell'OR* va a resettare la memoria, portandola in *I* posizione, interdicendo l'aria in uscita. Nell'ultimo modulo l'uscita *dell'OR* va a settare la memoria, portandola in *II* posizione, ponendo così l'aria in comunicazione dall'ingresso "*p*" all'uscita.

Viene riportato qui di seguito lo schema logico dell'ultimo modulo.

fig.7.4



Il segnale  $y_n$  di ingresso, proveniente dal modulo precedente, viene inviato all'ingresso *dell'OR* il quale è collegato con la linea " $L$ " di preset.

L'uscita *dall'OR* va a settare la memoria  $m_n$ .

Settata la memoria si ottiene il segnale  $A_n$  in uscita dal modulo, di comando dell'attuatore.

Detto segnale va all'ingresso *dell'AND*, ove staziona, e viene inviato ( $Z_n$ ) al modulo precedente per resettare la sua memoria.

Quando all'ingresso dell'*AND* arriva il segnale  $x_n$  di "avvenuto comando", allora, e solo allora, dal modulo viene emesso, in uscita, il segnale  $y_{n+1}$  che, in serie allo Start va a comandare il primo modulo per l'inizio di un nuovo ciclo.

Per iniziare il primo ciclo, occorre che vi sia in uscita dal modulo in esame il segnale  $y_{n+1}$ ; ma perché ciò avvenga occorre porre la memoria nello stato di Set. Per settare la memoria si invia un impulso sulla linea " $L$ " collegata all'ingresso dell'*OR* la cui uscita effettua il settaggio.

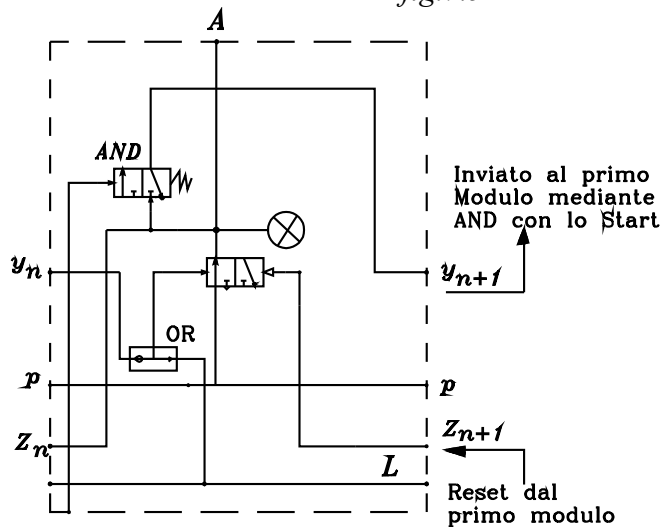
Quando verrà attivato il primo modulo esso produrrà il segnale  $z_{n+1}$  che andrà a resettare l'ultimo modulo.

#### 7.7.2.1 Realizzazione pratica dell'ultimo modulo

Ci si riferisce al modulo fornito dalla ditta *FESTO*

## MODULO TAB

fig.7.5



L'ultimo modulo, come si è detto, è differente dagli altri.

L'OR viene posto prima della memoria 3/2 e questa, nelle condizioni iniziali, viene commutata in II posizione (*memoria settata*).

Il modulo deve assolvere a funzioni diverse dagli altri.

Infatti esso, attivato per ultimo, deve resettare il modulo precedente e deve fornire il segnale di fine ciclo  $y_{n+1}$  da porsi in AND con lo Start, in modo da dare il via ad un nuovo ciclo.

Ne viene che, nelle condizioni

iniziali, prima del segnale di Start, dall'ultimo modulo deve uscire il segnale  $y_{n+1}$  che permette l'inizio del nuovo ciclo.

Prima di iniziare il ciclo deve essere settata nella II posizione la memoria dell'ultimo modulo, così, in uscita da essa vi è il segnale che va all'AND del modulo stesso.

Inoltre, all'inizio di un ciclo, perché si effettui il primo passo, occorre che siano rispettate le condizioni dell'ultimo (*non inizia il primo passo se non è stato effettuato l'ultimo*); per cui è attivato l'attuatore che fornisce il segnale  $x$  di "avvenuto comando" all'ultimo modulo TAB. Detto segnale si porta al pilotaggio della valvola AND 3/2 alla quale perviene contemporaneamente, nella alimentazione, il segnale di uscita della memoria.

Si avrà così che nelle condizioni iniziali dal modulo TAB esce il segnale  $y_{n+1}$  che va in AND con lo Start di inizio ciclo.

Una volta pigiato lo Start, iniziato il ciclo, l'ultimo modulo TAB deve essere disattivato e la memoria 3/2 deve essere posta in I posizione, in modo da annullare il segnale  $y_{n+1}$ . Per ottenere ciò,  $Z_{n+1}$  dell'ultimo modulo viene collegato con lo  $Z_n$  del primo modulo.

Riassumendo, il funzionamento del modulo TAB:

- 1- Nelle condizioni iniziali la memoria del TAB deve essere attivata e il segnale di uscita da essa si porta alla valvola AND del modulo stesso.

Per ottenere il settaggio della memoria dell'ultimo modulo e il resettaggio dei precedenti si invia un impulso di aria nel condotto L.

- 2- Contemporaneamente l'attuatore dell'ultimo passo, attraverso la  $x$  invia il segnale al pilotaggio della valvola 3/2 funzionante da AND, in tal modo da essa esce il segnale  $y_{n+1}$  che va all'ingresso di un altro AND insieme allo Start per attivare il primo modulo e iniziare il ciclo.
- 3- Dal quest'ultimo operatore AND esce il segnale  $y_n$  di attivazione del primo modulo. Questo, attivato, fornisce il segnale di settaggio del II modulo e il segnale di resettaggio  $Z_n$  che va a resettare l'ultimo modulo TAB ( $Z_n = Z_{n+1}$ ).



- 4- Così attivato il primo modulo, viene disattivato l'ultimo: la valvola di memoria 3/2 di questo viene commutata in prima posizione (disattivata) e dall'AND del modulo TAB non esce più il segnale  $y_{n+1}$ .
- 5- Quando dal penultimo modulo viene inviato al modulo TAB il segnale  $y_n$ , questo attraverso l'OR, va a commutare la valvola di memoria e la pone in posizione attivata di fine ciclo (coincidenti con le condizioni iniziali) e, quando perviene il segnale di avvenuto comando dalla x, dalla valvola 3/2 AND esce il segnale  $y_{n+1}$ .

Da quanto si è detto, nelle condizioni iniziali, prima di iniziare il primo ciclo, occorre settare in II posizione la memoria dell'ultimo modulo TAB e resettare tutte le altre e porle in I posizione. Per ottenere ciò si invia un impulso attraverso la linea "L" a tutti gli OR; in tal modo, data la loro disposizione circuitale, il segnale di uscita da essi setterà la memoria dell'ultimo modulo e resetterà quella di tutti gli altri.

Per comprendere più agevolmente quanto detto, conviene riferirsi ad un esempio pratico.

---

### IN LABORATORIO

---

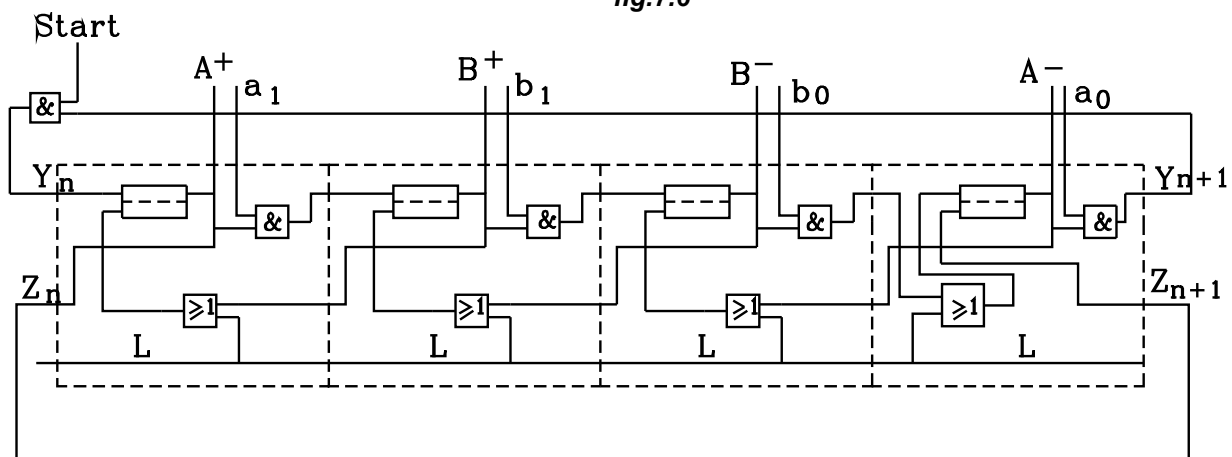
#### CICLO $A^+ B^+ B^- A^-$

Con il sequenziatore effettuiamo il ciclo semplice  $A^+ B^+ B^- A^-$  nel quale vi sarebbe un segnale bloccante se si effettuasse con i precedenti metodi di soluzione.

Con il sequenziatore ogni passo ha associata una memoria, quindi nessun segnale attivo diviene bloccante.

In fig. viene riportato lo schema logico di funzionamento, mentre nella pagina seguente è rappresentata la realizzazione con i moduli TAA e TAB.

**fig.7.6**



Nelle condizioni iniziali tutti i moduli TAA sono resettati, mentre l'ultimo modulo TAB è settato nella posizione II fig.7.6 per cui dall'uscita  $y_{n+1}$  dell'ultimo modulo esce l'aria che staziona ad uno degli ingressi dell'AND.

Inviando il segnale di Start, *dall'AND* esce il segnale  $y_n$ . Questo setta la memoria del primo modulo, che attiva l'uscita  $A^+$ , si porta all'ingresso *dell'AND*, e, attraverso un collegamento esterno  $Z_n - Z_{n+1}$ , arriva al resettaggio della memoria dell'ultimo modulo (*che nelle condizioni iniziali è settato*).

Lo stelo di  $A$  fuoriesce fino a che non tocca il fine-corsa  $a_1$ . Toccato questo, viene inviato il segnale di avvenuta fuoriuscita *stelo all'AND*, dove staziona il precedente segnale ottenuto dalla memoria attivata.

A questo punto *dall'AND* esce il segnale di comando  $y_{n+1}$  per il settaggio del modulo successivo.

Il segnale di uscita  $y_{n+1}$  dal primo modulo setta la memoria del secondo. Da essa esce il segnale che va ad attivare l'uscita  $B^+$ , si porta all'ingresso *dell'AND*, e, attraverso  $Z_n$  e  $Z_{n+1}$ , si porta *all'OR* del modulo precedente per resettarne la memoria.

Lo stelo di  $B$  fuoriesce fino a che non tocca il fine-corsa  $b_1$ . Toccato questo, viene inviato il segnale di avvenuta fuoriuscita *all'AND*, dove staziona il precedente segnale ottenuto dalla memoria attivata.

A questo punto *dall'AND* esce il segnale di comando  $y_{n+1}$  per il settaggio del modulo successivo.

Il segnale di uscita  $y_{n+1}$  dal secondo modulo setta la memoria del terzo. Da essa esce un segnale che va ad attivare l'uscita  $B^-$ , si porta all'ingresso *dell'AND*, e, attraverso  $Z_n$  e  $Z_{n+1}$ , si porta *all'OR* del modulo precedente per resettarne la memoria.

Lo stelo di  $B$  rientra fino a che non tocca il fine-corsa  $b_0$ . Toccato questo, viene inviato il segnale di avvenuto rientro *all'AND*, dove staziona il precedente segnale ottenuto dalla memoria attivata.

A questo punto *dall'AND* esce il segnale di comando  $y_{n+1}$  per il settaggio del modulo successivo.

Questo segnale entra nell'ultimo modulo  $TAB$  e va all'ingresso *dell'OR* la cui uscita va a settare la memoria (*resettata all'inizio del ciclo dallo  $Z_n$  del primo modulo*).

Dalla memoria esce un segnale che va ad attivare l'uscita  $A^-$ , si porta all'ingresso *dell'AND*, e, attraverso  $Z_n$  e  $Z_{n+1}$ , va *all'OR* del modulo precedente per resettarne la memoria.

Lo stelo di  $A$  rientra fino a che non tocca il fine-corsa  $a_0$ . Toccato questo, viene inviato il segnale di avvenuto rientro *all'AND*, dove staziona il precedente segnale ottenuto dalla memoria attivata.

A questo punto *dall'AND* dell'ultimo modulo esce il segnale di comando  $y_{n+1}$ . Questo viene posto, attraverso un collegamento esterno, in AND con lo Start.

L'uscita *dall'AND* da il segnale  $y_n$  di settaggio del primo modulo.

Ora se lo Start è attivato, essendo  $y_{n+1}=1$ , allora inizia un nuovo ciclo, altrimenti si ha la fine del ciclo.

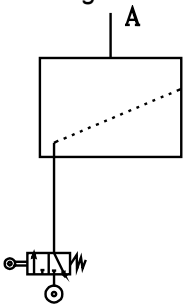
Per iniziare il primo ciclo occorre che la memoria dell'ultimo modulo  $TAB$  sia settata, mentre tutte le memorie dei moduli  $TAA$  siano resettate.

Per essere sicuri di ciò, si invia un impulso alla linea di preset "L": allora tutte le memorie dei moduli  $TAA$  vengono resettate dagli OR, mentre la memoria dell'ultimo modulo viene settata ponendosi nelle condizioni iniziali di inizio ciclo.

I passi del sequenziatore possono essere schematizzati in una forma molto semplice che prescinde dalla costituzione dei moduli e dal loro schema logico.

fig.7.7

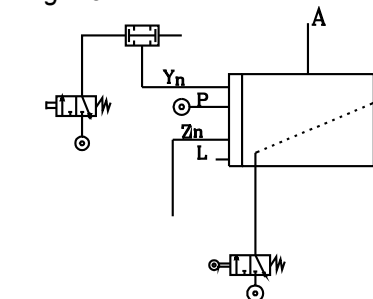
#### Passo intermedio



I passi intermedi vengono schematizzati con un rettangolo, sul cui lato superiore viene indicato il segnale di uscita "A" dal modulo, sul lato inferiore è rappresentato il condotto che trasporta il segnale  $x$ , proveniente dal fine-corsa, indicante l'avvenuta attuazione del comando. Una linea tratteggiata sta ad indicare, che arrivato nel modulo il segnale  $x$ , viene attivato il passo successivo.

#### Primo passo

fig.7.8



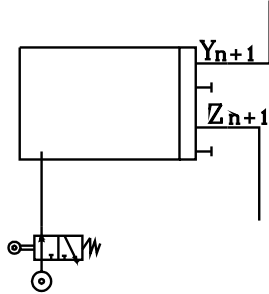
Sul lato sinistro del primo passo vengono indicati:

- 1- Il condotto che porta il segnale di ingresso  $y_n$  inviato *dall'AND*, ove arriva il segnale  $y_{n+1}$  proveniente dall'ultimo modulo e il segnale di Start.
- 2- Il condotto di alimentazione  $P$ .

- 3- Il condotto  $Z_n$  per il resettaggio dell'ultimo modulo.
- 4- La linea di preset "L".

Ultimo passo

fig.7.9



Sul lato destro dell'ultimo passo viene indicato:

- 1- Il condotto del segnale  $y_{n+1}$  di uscita dall'ultimo modulo che va, insieme al segnale di *Start*, all'ingresso dell'*AND*, la cui uscita produce il segnale  $y_n$  di comando del primo modulo.
- 2- Il condotto del segnale  $Z_{n+1}$ , proveniente dal primo modulo per il resettaggio della memoria dell'ultimo.

Il ciclo  $A^+ B^+ B^- A^-$  viene rappresentato dallo schema:

fig.7.10

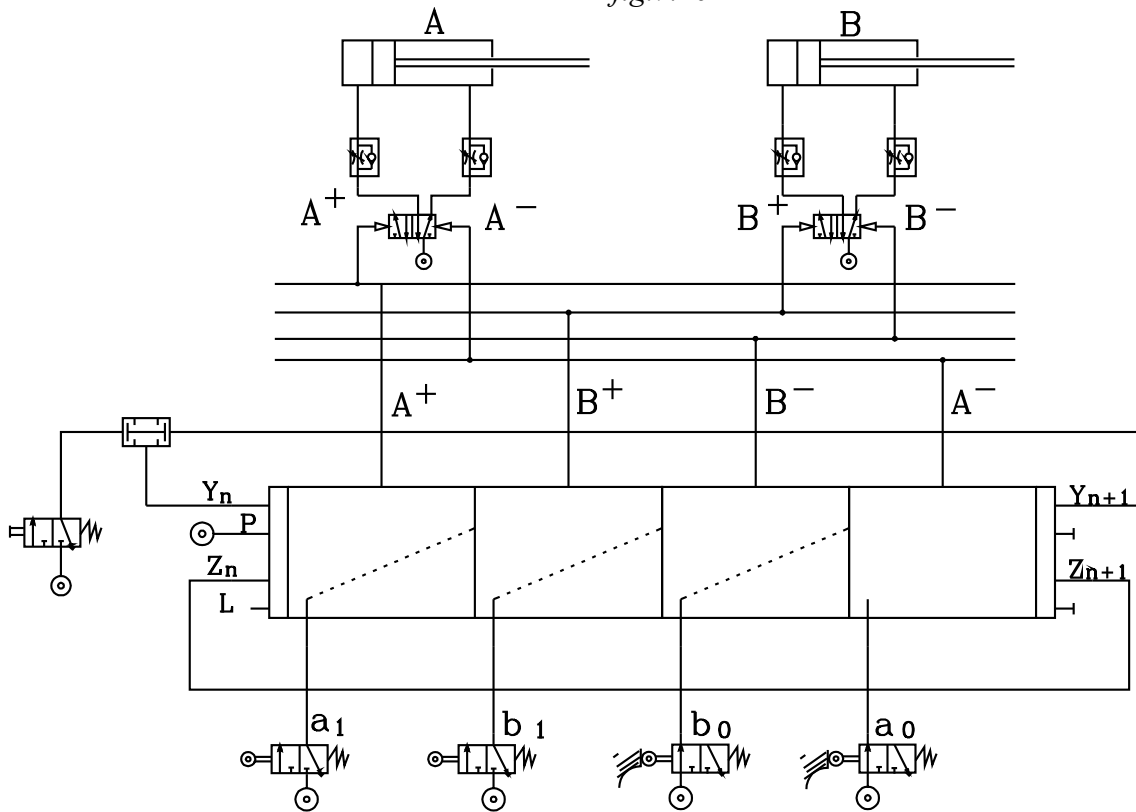
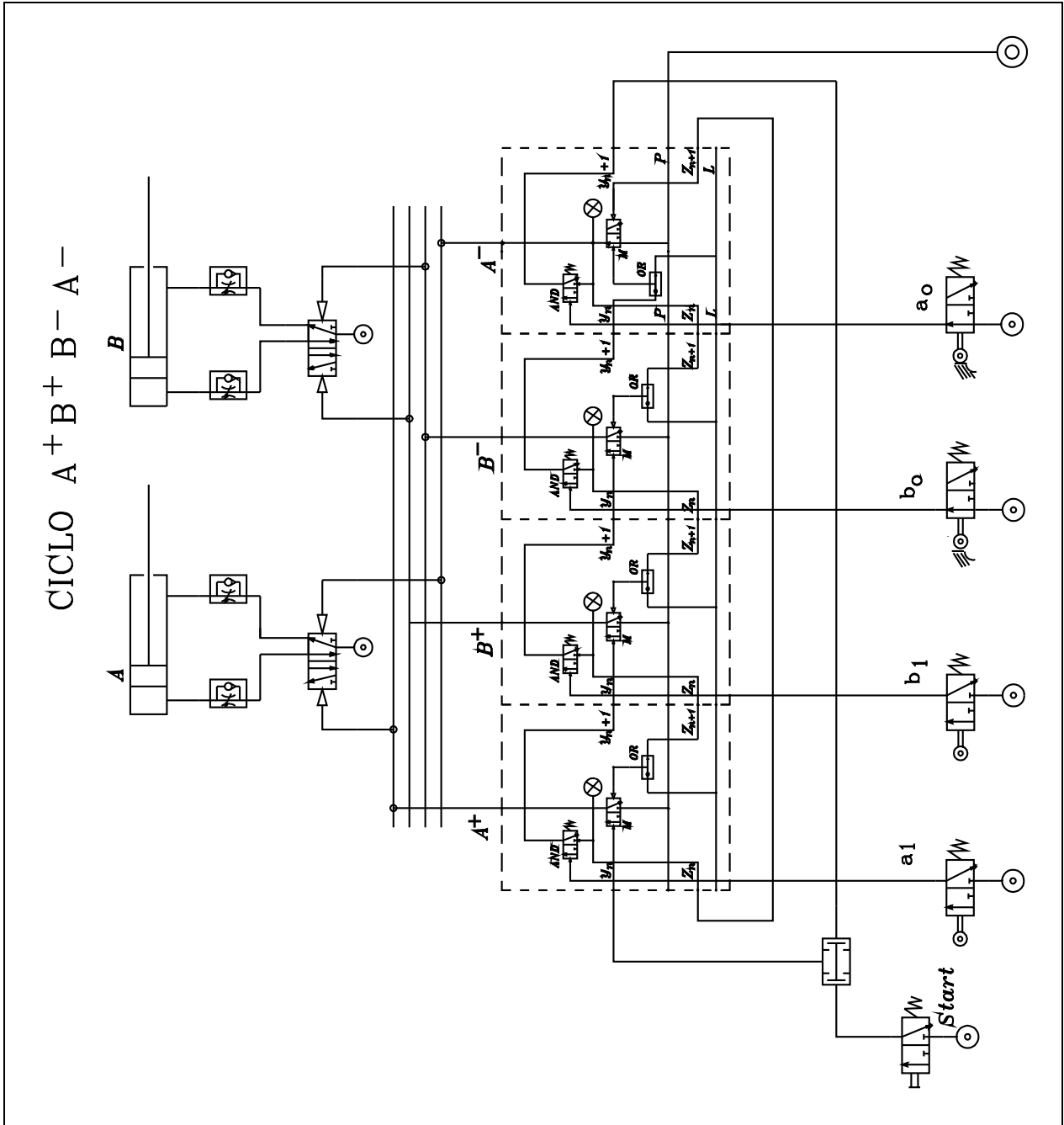


fig.7.11



### 7.3 GRAFCET

Con l'impiego dei sequenziatori torna utile rappresentare le varie fasi del ciclo con un grafico detto grafcet.

Occorre osservare che un qualsiasi ciclo è un insieme di fasi che si succedono nel tempo in un numero di stadi distinti.

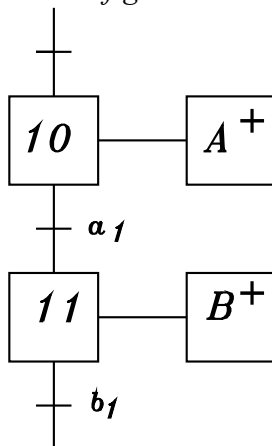
Il ciclo si può suddividere in una successione di varie fasi nelle quali viene svolta una determinata operazione (*fuoriuscita stelo*, *rotazione motore ecc.*).

Vi sono degli stadi di transizione nei quali si passa da una fase alla successiva.

Tutto ciò può essere descritto da un grafico, nel quale:

- 1- Ogni fase è rappresentata da un quadrato nel quale viene scritto il numero d'ordine, affiancato da un altro quadrato nel quale viene riportata la funzione che si deve operare nella fase stessa.
- 2- Lo stato di riposo del sistema è contraddistinto da un quadrato con doppio riquadro : *fase 0*.
- 3- I quadrati contenenti il numero d'ordine sono collegati da tratti di linea, nel senso della sequenza, che si svolge dall'alto verso il basso, indicanti gli stadi di transizione.
- 4- Tra una fase e l'altra vi è una fase di transizione nella quale la fase precedente permane fino a che *non si verifica una condizione che permetta il passaggio alla fase successiva*.
- 5- Tale condizione viene indicata con un trattino orizzontale che interseca la linea di transizione tra la fase che si sta operando e la successiva

*fig.7.12*



Al trattino viene associata la condizione logica che permette la transizione da una fase all'altra.

Così, per meglio capire, si supponga che si debba effettuare la sequenza  $A^+B^+$ .

Alla fase 10 è associata l'operazione  $A^+$  di fuoriuscita dello stelo di A.

La fase permane fino a che non si verifica la condizione logica di fine operazione: ciò avviene quando è toccato il finecorsa  $a_1$  di fuoriuscita stelo.

Quando si verifica tale condizione, e solamente allora, si ha la transizione dalla fase attuale alla successiva.

E' evidente che la transizione, e quindi il verificarsi della condizione logica ( *$a_1$  toccato*) si ha soltanto se è stata attivata la fase immediatamente precedente (*fase 10 nell'esempio*)

La transizione viene superata quando, attivata la fase precedente, la condizione logica di attuazione di detta fase diviene vera.

Il superamento di una transizione conduce alla fase successiva.

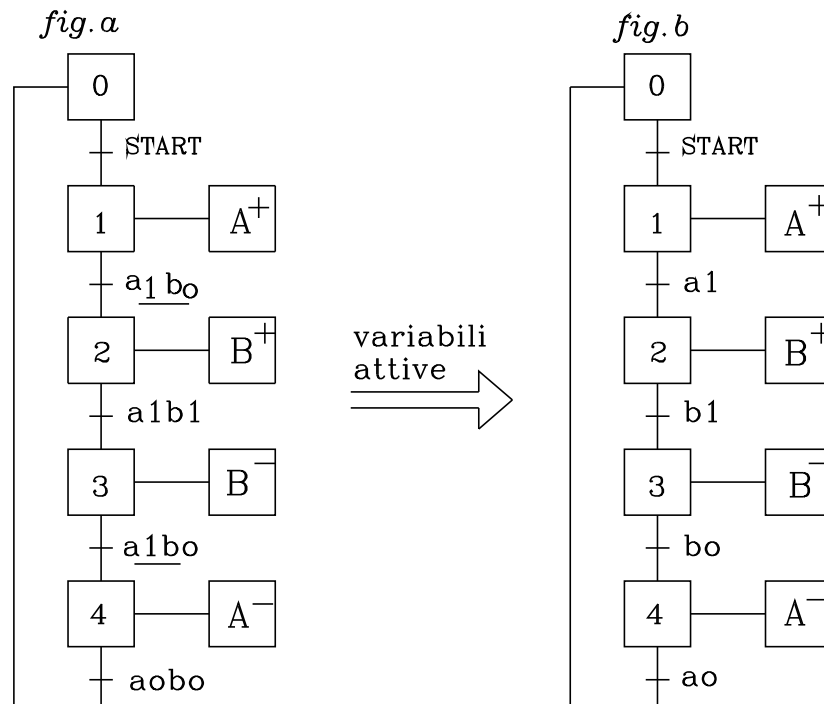
L'attivazione della fase successiva produce la disattivazione della fase precedente.

Si consideri così il ciclo  $A^+ B^+ B^- A^-$ .

Si effettui il Grafcet, ponendo nelle transizioni la combinazione logica dei fine-corsa toccati: *fig.7.12 a*.

Nella *fig.7.12 b* viene rappresentato il Grafcet ove nelle transizioni sono poste le variabili attive.

*fig.7.13*



Nel Grafcet di *fig.7.13 a* si notano, negli stati di transizione  $x_i$ , le due combinazioni uguali  $a_1 b_0$  che costituirebbero segnali bloccanti senza l'ausilio di memorie.

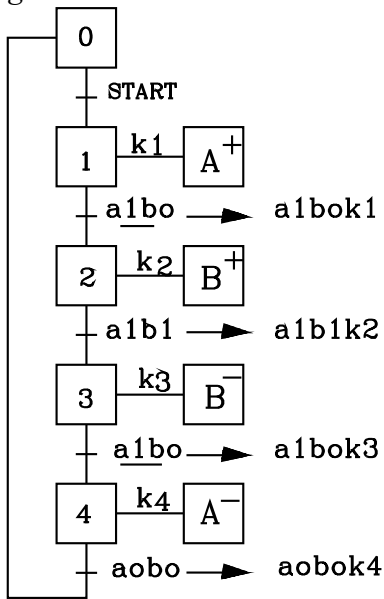
Nei sequenzatori ad ogni fase viene associata una memoria *fig.7.13c*. Alla fase  $i^{ma}$  viene associata la memoria  $k_i$  il cui settaggio dà il segnale di comando del passo stesso ( $A^+ B^+$  ecc.), che indicheremo genericamente con  $C_i$ .

Non si attiva il passo successivo fino a ch , settata la memoria  $k_i$ , non si verifica la combinazione dei fine-corsa, scritta nella transizione: segnale di ritorno indicante l'avvenuta attuazione del comando.

Il comando di attivazione della fase successiva   dato dall'AND tra il segnale di transizione e il segnale di memoria. Essendo associata ad ogni fase una memoria diversa non vi saranno nel ciclo segnali di comando uguali: essi differiranno almeno per il segnale di memoria.

Con riferimento alla *fig.7.13 c* si nota ora che, con l'introduzione dei segnali di memoria, i segnali di comando della II e IV fase non sono pi  uguali: differiscono per il diverso segnale di memoria, precisamente:

fig.7.14



$$A^+ \quad A^+ = y_1 = a_o \cdot b_o \cdot k_4 \cdot Start$$

Si ha la fuoriuscita dello stelo di A ( $A^+$ ) quando si è attuata l'ultima fase: si è settata la memoria  $k_4$  - è rientrato lo stelo di A - sono toccati  $a_o$ ,  $b_o$  - viene pigiato lo Start

$$B^+ \quad B^+ = a_1 \cdot b_o \cdot k_1$$

La fuoriuscita dello stelo di B si ha quando, settata la memoria  $k_1$ , è fuoriuscito lo stelo di A e sono toccati i fine corsa  $a_1$ ,  $b_o$

$$B^- \quad B^- = a_1 \cdot b_1 \cdot k_2$$

Il rientro dello stelo di B si ottiene quando, settata la memoria  $k_2$  della fase precedente, è fuoriuscito lo stelo del cilindro B e sono toccati i fine corsa  $a_1$ ,  $b_1$

$$A^- \quad A^- = a_1 \cdot b_o \cdot k_3$$

Il rientro dello stelo del cilindro A si ottiene quando, settata la memoria  $k_3$  della fase precedente, è rientrato lo stelo del cilindro B e sono toccati i fine corsa  $a_1$ ,  $b_o$

Adoperando il sequenziatore pneumatico, in pratica non occorre scrivere alcuna equazione logica: è sufficiente tracciare il Grafcet, ponendo nelle transizioni solamente le variabili attive.

E' evidente che si volesse un controllo totale sulle posizioni dei finecorsa, nelle transizioni occorrerebbe porre la combinazione completa dello stato di tutti i finecorsa toccati. Ciò porterebbe però ad una complicazione circuitale con un aggravio che si riflette sulla spesa dell'impianto.

Così il ciclo  $A^+ B^+ B^- A^-$  può essere rappresentato dal Grafcet di fig.7.13 b.

In tal modo ogni fase corrisponde ad un modulo.

Nel quadrato posto accanto a quello col n° di fase vi è indicato il collegamento dell'uscita del modulo con la valvola di comando dell'attuatore: così " $A^+$ " indica che l'uscita del modulo deve essere collegato con il pilotaggio della valvola bistabile che comanda la fuoriuscita del cilindro A.

Il fine-corsa riportato nel trattino di transizione dà il collegamento da effettuare tra il fine-corsa indicato e il condotto del segnale di ritorno  $x$  al modulo del sequenziatore: così " $a_1$ " sta ad indicare che il fine-corsa  $a_1$  viene collegato con il condotto  $x$  del modulo.

---

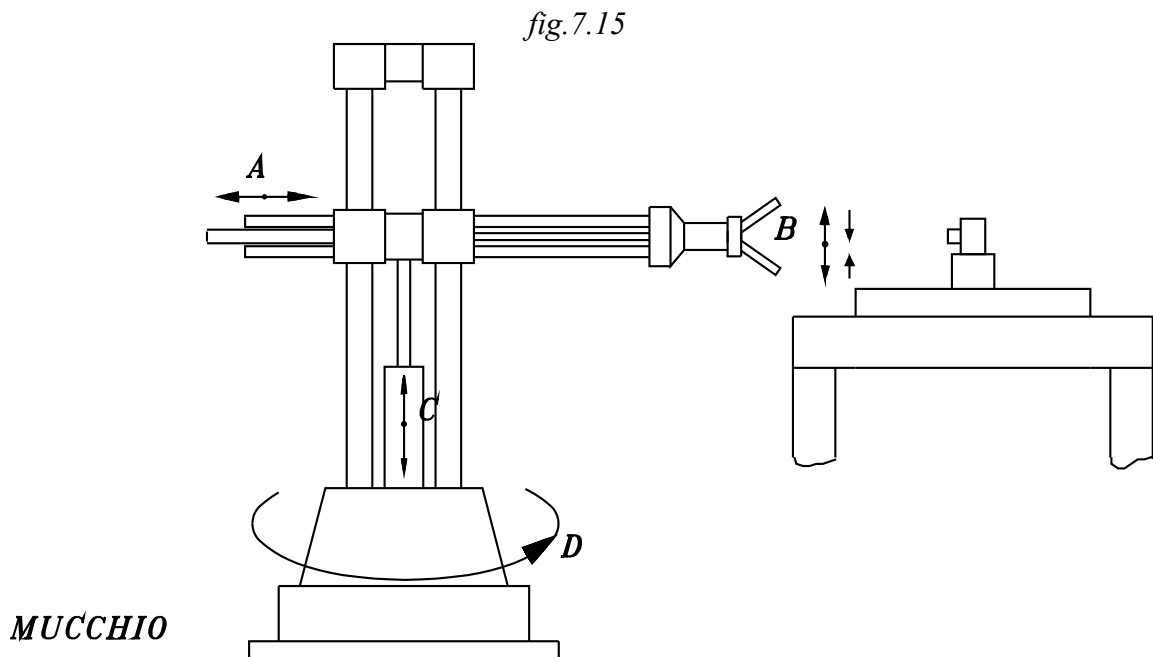
## IN LABORATORIO

---

### 7.3.1 APPLICAZIONE DEL SEQUENZIATORE PNEUMATICO

Nella figura fig.7.15 è rappresentato schematicamente un robot sequenziale.

Il robot deve prendere dei pezzi dal nastro trasportatore, trasportarli e lasciarli cadere nel mucchio.



La movimentazione avviene con le seguenti fasi:

- 1° Il braccio è al livello del pezzo.  
Quando questo, trasportato dal nastro, si trova in posizione di presa, si ferma il nastro trasportatore e viene dato il segnale  $S$  di inizio del ciclo. Il cilindro  $A$  fa fuoriuscire il braccio ( $A^+$ ), portando la pinza in posizione di presa. Un sensore di prossimità o meccanico rileva la corretta posizione del pezzo.
- 2° Quando il braccio ha raggiunto il finecorsa  $a_1$  e si ha il segnale di corretta posizione del pezzo:  $u=1$ , allora viene chiusa la pinza attraverso il cilindro  $B$  ( $B^+$ ).
- 3° Toccato il finecorsa  $b_1$  viene comandato il cilindro  $C$ , il quale fa fuoriuscire lo stelo ( $C^+$ ), alzando il pezzo dal nastro trasportatore.
- 4° Toccato il finecorsa  $c_1$ , viene comandato il cilindro  $D$ , il quale fa fuoriuscire lo stelo ( $D^+$ ). Il moto rettilineo dello stelo viene tramutato in moto rotatorio del montante del robot attraverso un meccanismo di rocchetto dentiera.
- 5° Effettuata una rotazione di  $180^\circ$  e toccato il finecorsa  $d_1$ , viene comandato il cilindro  $C$  ad abbassarsi in posizione di deposizione del pezzo nel pallet ( $C^-$ ).
- 6° Toccato il finecorsa  $c_1$  e rilevata la giusta posizione del pezzo attraverso un sensore di prossimità  $w$ , si apre la pinza con rientro del cilindro  $B$  ( $B^-$ ).
- 7° Toccato  $b_0$  si ritrae il braccio ( $A^-$ ).
- 8° Toccato  $a_0$  il robot ruota di  $180^\circ$  in senso inverso al precedente, rientra il cilindro  $D$  ( $D^-$ ) fino a toccare il finecorsa  $d_0$ .

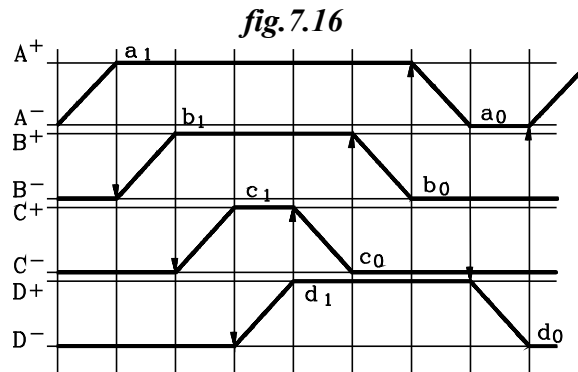
Quando un altro pezzo arriva in posizione allora viene prodotto il segnale  $S$  di inizio di un altro ciclo.

Il ciclo è:

$$A^+ B^+ C^+ D^+ C^- B^- A^- D^-$$



## Diagramma corsa passo



## Grafcet

fig. 7.17

Il grafcet è composto dalle seguenti fasi:

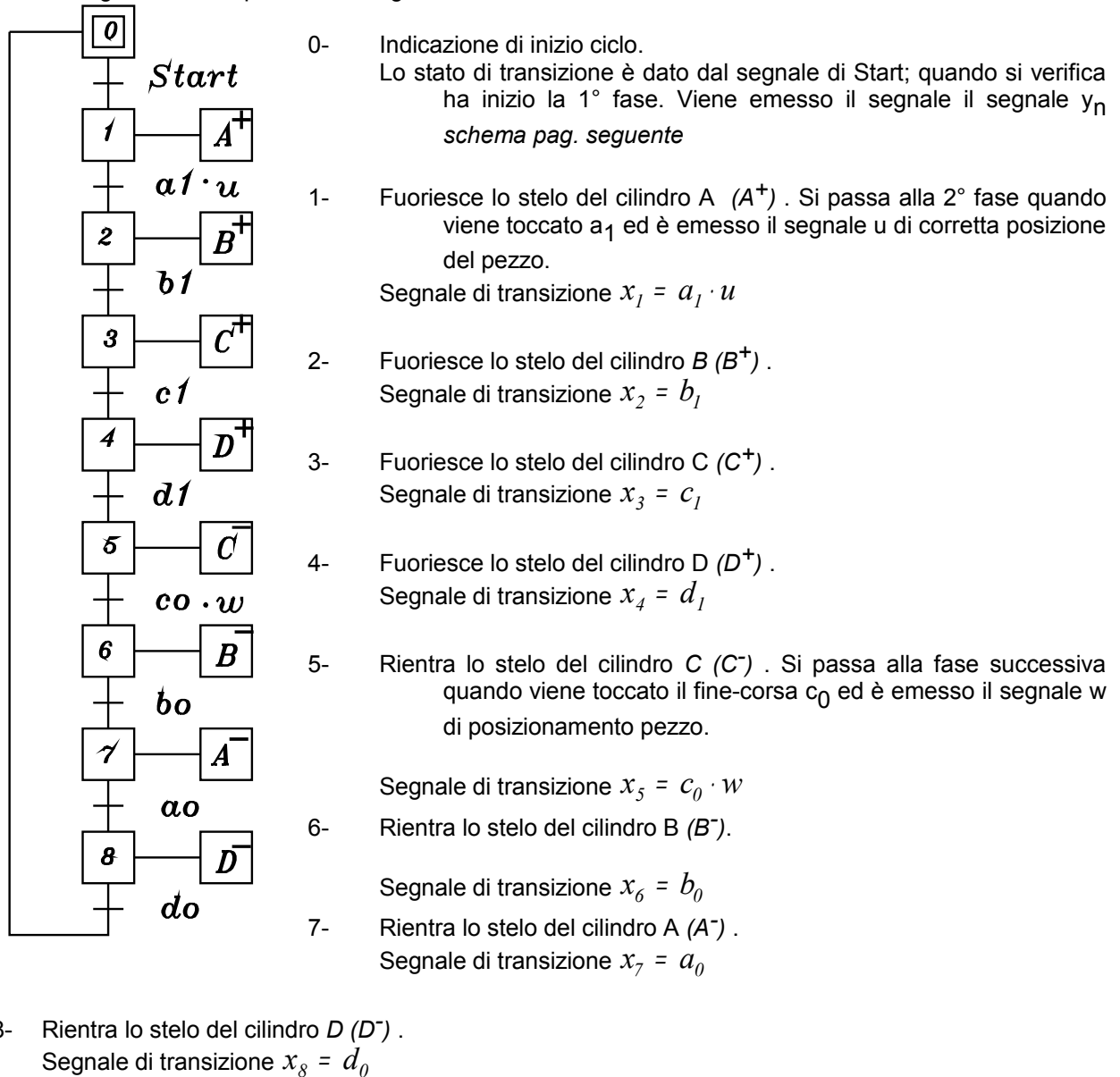
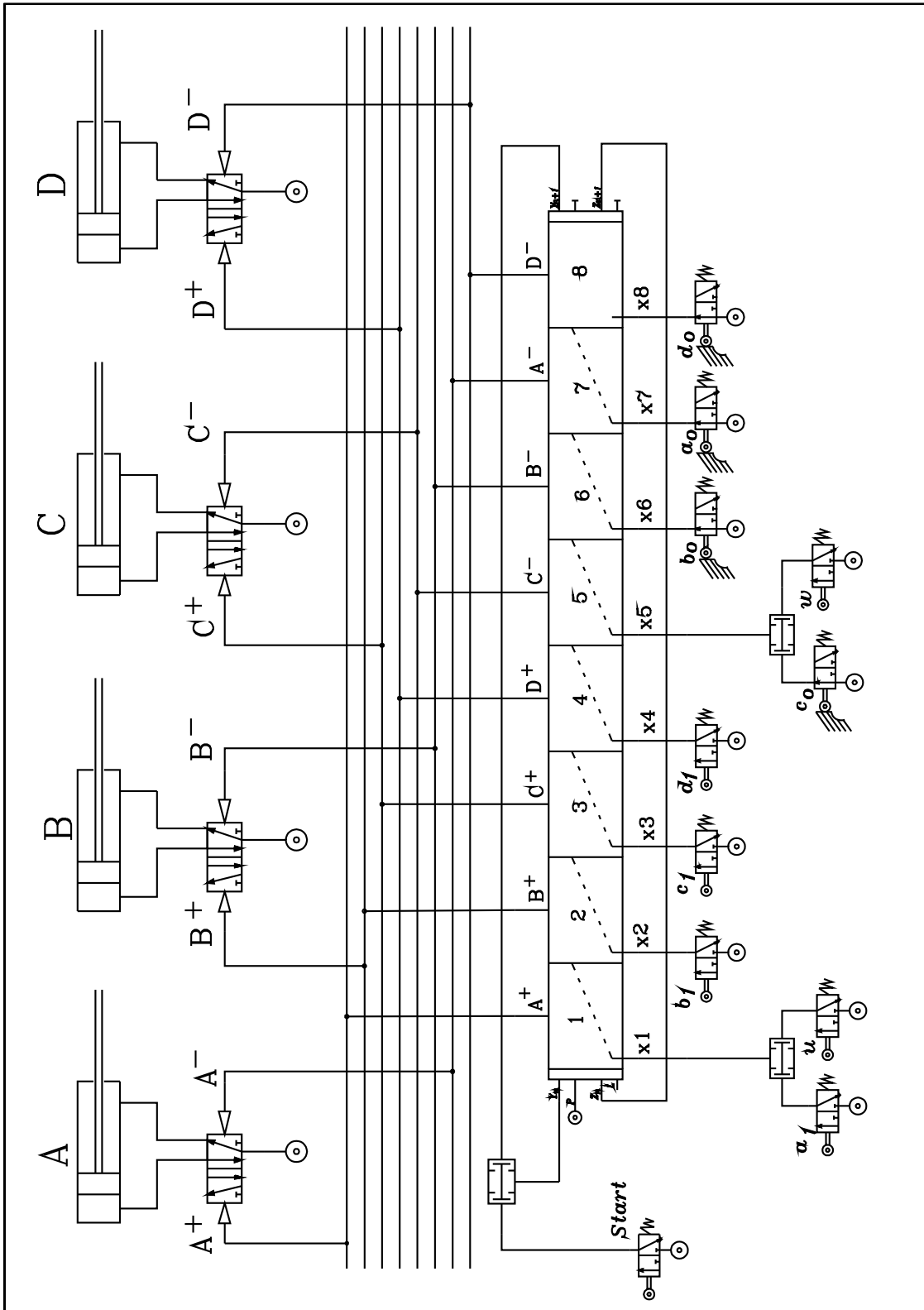


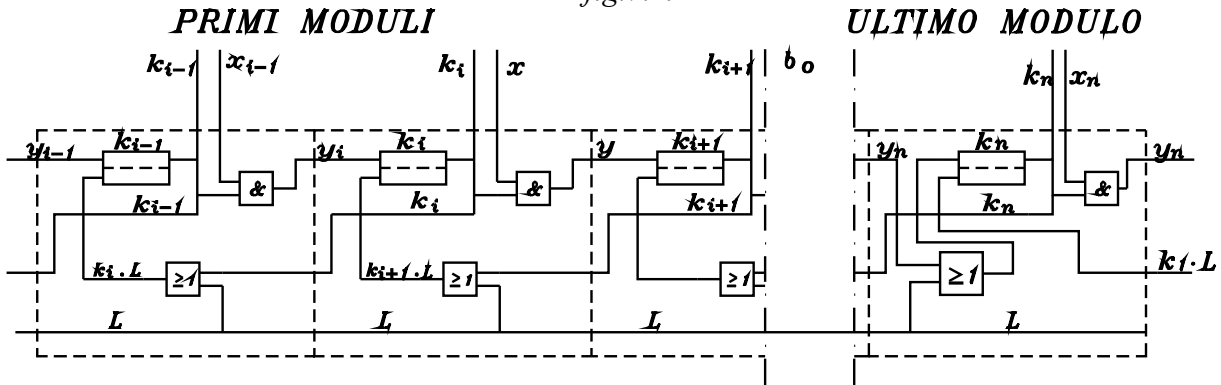
fig.7.18



### 7.3.2 Equazioni logiche del sequenziatore

Dallo schema logico del sequenziatore si possono ricavare le equazioni logiche tra ingresso e uscita dei moduli. Tali equazioni torneranno utili nella progettazione dei circuiti elettropneumatici.

fig.7.19



Si considerino tre moduli generici consecutivi :  $i-1, i, i+1$ .

Si indicano con:

$k_i$  La memoria associata al modulo  $i$ .

Sarà  $k_{i-1}$  la memoria del modulo  $i-1$  e  $k_{i+1}$  quella del modulo  $i+1$ .

$y_i$  Il segnale di ingresso del modulo  $i$ , di settaggio della sua memoria.

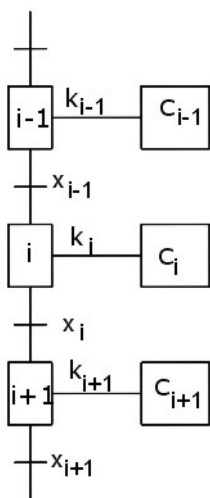
Sarà  $y_{i-1}$  il segnale di ingresso del modulo  $i-1$ , e  $y_{i+1}$  quello di ingresso del modulo  $i+1$ .

$C_i$  Il segnale di comando di uscita dal modulo  $i$  (per esempio  $A^+, B^+$  ecc..)

Sarà  $C_{i-1}$  il segnale di comando del modulo  $i-1$  e  $C_{i+1}$  quello del modulo  $i+1$ .

$x_i$  Il segnale di ritorno dall'attuatore al modulo  $i$ , indicante l'avvenuta attuazione del comando.

Sarà  $x_{i-1}$  il segnale di ritorno al modulo  $i-1$  e  $x_{i+1}$  quello di ritorno al modulo  $i+1$ .



La fase  $i$ ma viene attivata, producendo il comando  $C_i$ , quando si sono verificati: sia il settaggio della memoria  $k_{i-1}$  precedente sia la condizione logica  $x_{i-1}$  della transizione precedente, indicante l'avvenuta attuazione del comando  $C_{i-1}$ .

La fase  $i$ ma, che produce il comando  $C_i$ , viene disattivata quando verrà attivata la memoria successiva dal nuovo comando di transizione.

Un segnale di memoria  $k_{i+1}$  resetta la precedente memoria  $k_i$ .

Si è indicato con  $y_i$  il segnale che attiva la fase  $i$ ma e che produce il segnale di comando  $C_i$ . Quest'ultimo segnale deve permanere fino a che non si verifica la condizione di transizione  $x_i$  indicante l'avvenuta attuazione del comando stesso.

Occorre quindi associare alla fase  $i$ ma una memoria bistabile  $K_i$ , che settata, produce il segnale di comando  $C_i$ :

$$C_i = k_i$$

La memoria verrà resettata quando verrà settata la memoria successiva  $k_{i+1}$ .

$$\bar{k}_i = k_{i+1} \quad \text{oppure} \quad k_i = \overline{k_{i+1}}$$

Il segnale di comando della memoria  $k_i$  della fase  $i^{ma}$  è prodotto al verificarsi del segnale di comando della memoria precedente  $C_{i-1} = k_{i-1}$  e del segnale  $x_{i-1}$  di avvenuta attuazione del comando stesso:

$$k_i = k_{i-1} \cdot x_{i-1}$$

Le equazioni logiche sono:

$y_i = k_{i-1} \cdot x_{i-1}$  Attivazione della fase  $i^{ma}$ , proveniente dalla transizione della fase precedente.

$k_i = y_i$  Segnale di memoria della fase  $i^{ma}$ . Settaggio memoria.

$\bar{k}_{i-1} = k_i + L$  Resettaggio della memoria della fase precedente  $i-1$ . Il segnale  $L$  è quello che si invia all'inizio del ciclo per resettare tutte le memorie che precedono l'ultima.

$C_i = k_i$  Segnale di comando della fase  $i^{ma}$ . Segnale di uscita dal modulo.

$y_{i+1} = k_i \cdot x_i$  Segnale di transizione della fase  $i^{ma}$  in AND con il segnale di memoria e attivazione del modulo successivo  $i+1$ .

$k_{i+1} = y_{i+1}$  Segnale di settaggio della memoria successiva  $i+1$ .

$\bar{k}_i = k_{i+1} + L$  Resettaggio della memoria  $i^{ma}$ .

Seguiamo i primi passaggi:

Nel modulo precedente  $i-1$  è stata settata la memoria  $k_{i-1}=1$  che ha effettuato il comando  $C_{i-1}$ . Si ha così lo stato di transizione.

Quando il comando  $C_{i-1}$  è effettuato, allora viene emesso il segnale  $x_{i-1}=1$  di avvenuto comando. Questo, posto in AND con il segnale di memoria  $k_{i-1}$ , determina il segnale  $y_i$  di comando della fase successiva  $i$ :

$$y_i = k_{i-1} \cdot x_{i-1} = 1 \cdot 1 = 1$$

Viene settata la memoria  $k_i$ .

$$k_i = y_i = 1$$

La memoria adoperata nel sequenziatore pneumatico è costituita da una valvola bistabile 3/2. Si è indicato con  $k_i$  il segnale di Set quando l'aria va in uscita. Viene indicato con  $\bar{k}_i$  la condizione di Reset, nella posizione della valvola che interdice il passaggio dell'aria in uscita.

Il segnale  $k_i$  reseta la memoria precedente  $k_{i-1}$

$$\bar{k}_{i-1} = k_i + L = 1 + L = 1 \quad \text{quindi} \quad k_{i-1} = 0$$

Il segnale di memoria  $k_i$  va in uscita ed effettua il comando  $C_i$

$$C_i = k_i = 1$$

Si ha una nuova transizione.

Quando viene emesso il segnale di avvenuto comando  $x_i=1$ , questo posto in *AND* con il segnale di memoria  $k_i=1$  determina il segnale di comando  $y_{i+1}$  della fase  $i+1$  successiva.

$$y_{i+1} = k_i \cdot x_i = 1 \cdot 1 = 1$$

Viene settata la memoria  $k_{i+1}$

$$k_{i+1} = y_{i+1} = 1$$

Il segnale di memoria  $k_{i+1}$  resetta la memoria precedente  $k_i$

$$\bar{k}_i = k_{i+1} + L = 1 + L = 1 \quad \text{quindi} \quad k_i = 0$$

E così via...

Affinché il ciclo possa iniziare per la prima volta, *occorre garantire che nelle condizioni iniziali sia attivato l'ultimo modulo.*

Occorre cioè che sia settata l'ultima memoria  $k_n$  e resettate tutte le precedenti.

Per ottenere ciò, come si è visto nei sequenziatori pneumatici, viene aggiunto un segnale  $L$  di inizializzazione, il quale viene posto in *OR* con il segnale di resettaggio di tutti i moduli che precedono l'ultimo.

L'ultimo modulo  $n$  è diverso dagli altri.

La memoria viene settata dal segnale di transizione  $y_n=k_{n-1} \cdot x_{n-1}$  in *OR* con il segnale di inizializzazione  $L$ .

$$k_n = y_n + L$$

Come si è detto il segnale di inizializzazione  $L$  va posto in *OR* con i segnali di resettaggio delle memorie dei moduli precedenti:

$$\bar{k}_i = k_{i+1} + L \quad \text{oppure si può scrivere} \quad k_i = \overline{k_{i+1} + L}$$

In tal modo, inviando l'impulso di preset  $L=1$ , vengono resettate tutte le memorie  $k_i$  precedenti all'ultima:

$$\bar{k}_i = k_{i+1} + 1 = 1 \quad \text{quindi} \quad k_i = 0$$

mentre viene settata l'ultima

$$k_n = y_n + L = y_n + 1 = 1$$

## 8 ELETTROPNEUMATICA

Nella pneumatica finora studiata le funzioni logiche di comando sono state effettuate con circuiti nei quali fluiva l'aria. Sia il pilotaggio che l'azionamento di potenza è affidata all'azione dell'aria.

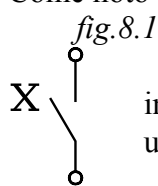
Nella elettropneumatica le funzioni di comando sono effettuate con circuiti elettrici, mentre il circuito di potenza che aziona gli attuatori è pneumatico.

Il comandi elaborati dal circuito elettrico vengono inviati alle bobine delle elettrovalvole che distribuiscono il fluido attivo ai cilindri.

Il circuito elettrico è costituito da contatti, comandati o direttamente da pulsanti o da bobine alle quali si inviano segnali di eccitazione.

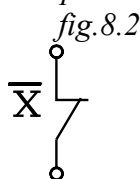
### 8.1 Simbologia

Come noto i contatti possono essere o normalmente aperti o normalmente chiusi.



Il contatto *normalmente aperto* è indicato con una stanghetta inclinata che interrompe la continuità della linea e posta in modo che la chiusura avvenga con una rotazione in senso orario.

Al contrario del circuito pneumatico, il contatto normalmente aperto "NA", in condizioni di riposo non permette il passaggio del segnale elettrico di comando (*la valvola pneumatica NA permette il passaggio dell'aria*).



Il contatto *normalmente chiuso* è rappresentato con una lineetta inclinata che si appoggia su di un'altra orizzontale, dando la continuità elettrica sulla linea. L'apertura avviene con una rotazione in senso orario.

Il contatto normalmente chiuso, in condizioni di riposo, permette il passaggio del segnale elettrico.

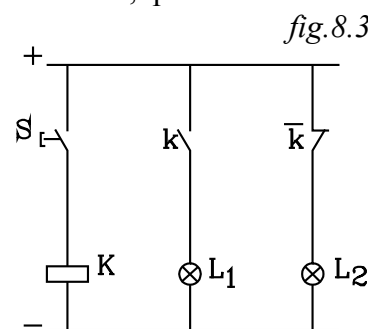
I circuiti di comando sono formati da più linee in parallelo, che si rappresentano dall'alto verso il basso, collegate tra una linea positiva in alto ed una negativa in basso. Un ramo va letto dall'alto verso il basso. Nella parte superiore, al di sotto della linea positiva (*zona di test*), vengono posti l'insieme dei *contatti, start, fine corsa, sensori di prossimità*, il cui collegamento (*in serie o parallelo*) determinano la funzione logica che comanda o una *bobina* o *l'elettrovalvola* di comando di un *attuatore*, posti nell'estremità inferiore, prima della linea negativa (*zona d'azione*).

Le bobine di un relè è rappresentata con un rettangolo contrassegnato da una lettera maiuscola. I contatti da essa comandati vengono contraddistinti dalla stessa lettera, preferibilmente in forma minuscola, senza alcun altro segno se il contatto è normalmente aperto, sormontato da un trattino se è normalmente chiuso.

Nello schema di figura la bobina  $K$  viene eccitata dallo start  $S$ . I contatti comandati dalla bobina sono indicati con la stessa lettera che la contraddistingue con carattere minuscolo "k".

Il contatto normalmente chiuso è sormontato da una lineetta (*indicante lo stato logico negato*).

Il circuito è costituito da tre rami in parallelo, posti in verticale tra le due linee, positiva in alto e negativa in basso.



Nella *zona superiore di test* sono rappresentati: lo *Start S* che eccita la bobina *K*, il contatto *k* che comanda l'accensione della lampada  $L_1$ , il contatto  $\bar{k}$  normalmente chiuso che comanda la lampada  $L_2$ . Nella *zona d'azione* sono rappresentati: la bobina *K*, le lampade  $L_1, L_2$ .

Le semplici funzioni logiche che si possono ricavare dai contatti rappresentati nella zona di test sono:

$$\begin{aligned} K &= S \\ L_1 &= k \\ L_2 &= \bar{k} \end{aligned}$$

Con lo *Start S* non inserito risulta diseccitata la bobina *K*:  $k=0$ .

In queste condizioni è accesa la lampada  $L_2$  e spenta  $L_1$ : 
$$\begin{cases} L_1 = k = 0 \\ L_2 = \bar{k} = \bar{0} = 1 \end{cases}$$

Con la *Start S* inserito risulta eccitata la bobina *K*:  $k=1$ .


In queste condizioni è accesa la lampada  $L_1$  e spenta  $L_2$ : 
$$\begin{cases} L_1 = k = 1 \\ L_2 = \bar{k} = \bar{1} = 0 \end{cases}$$


Questa prima parte dell'elettropneumatica si riferisce all'esecuzione di programmi detti a *logica cablata*. In questi, un problema pneumatico viene risolto inviando all'elettrovalvole di comando degli attuatori dei segnali, elaborati in un programma, fornito da un circuito elettrico, costituito da elementi fisici (bobine e contatti) da dover montare e collegare tra loro.


Nella seconda parte verrà studiata la soluzione di problemi di comando mediante *logica programmata*.

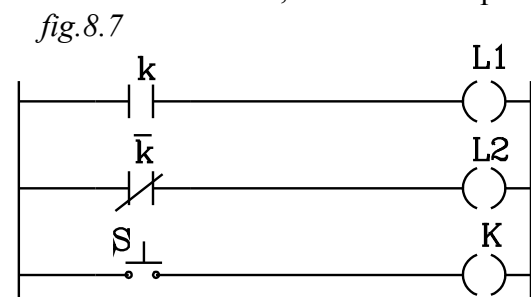
Nella logica programmata le funzioni logiche che forniscono i segnali di comando all'elettrovalvole, o agli attuatori in generale, sono elaborate via software da un programma memorizzato nella memoria di un computer dedicato alla soluzione di problemi di comando denominato *Controllore a Logica Programmata* (PLC).

La simbologia adoperata nella stesura del circuito elettrico nei *PLC* è diversa da quella che adoperiamo per la logica cablata.

*fig.8.4*  
 Il contatto normalmente aperto viene rappresentato con due stanghette parallele e contrassegnato dalla stessa lettera della bobina di comando.

*fig.8.5*  
 Il contatto normalmente chiuso viene rappresentato con due stanghette parallele sbarrate e contrassegnato con la stessa lettera della bobina sormontata da una stanghetta.

*fig.8.6*  
 Un'elettrovalvola, una bobina, una uscita in genere, viene indicata da un cerchietto, recante al disopra l'indicazione distintiva.



Nella figura *fig.8.7* è rappresentato lo stesso circuito studiato precedentemente con simbolismo diverso.

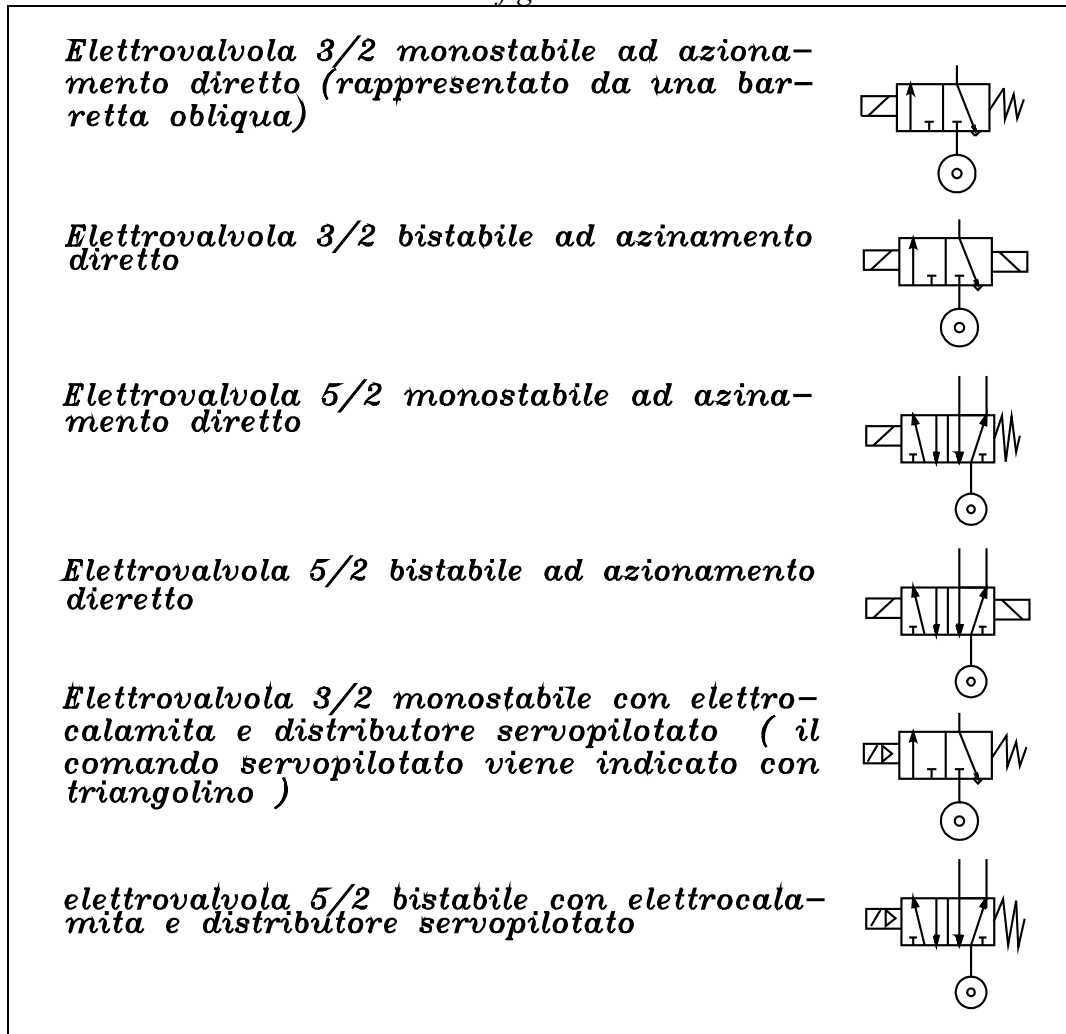
Il circuito è composto di più linee di corrente in parallelo che si svolgono da una linea positiva a sinistra ad un'altra negativa a destra.

Il circuito si compone ancora di una *zona di test* nella prima parte sinistra, composta dai contatti delle bobine, degli start ecc. che formano le equazioni logiche di comando, seguita, nell'ultima parte a destra, dalla *zona di azione*, composta dalla bobine, le elettrovalvole, le lampade ecc.

### 8.1.1 Elettrovalvole

Sono valvole di distribuzione nelle quali l'indirizzamento del flusso del fluido in uscita viene ottenuto mediante l'azione di un avvolgimento elettrico (*bobina*), il cui campo magnetico agisce su di un nucleo ferromagnetico solidale al sistema di commutazione della valvola.

fig.8.8



L'uso dell'elettrovalvole e del sistema elettropneumatico è consigliabile nel caso di lunghe distanze dal punto di comando e la posizione della valvola comandata.

Inoltre l'elettropneumatica è indispensabile nel caso di comandi con *PLC*.

L'elettrovalvole possono essere *ad azionamento diretto o servopilotate*

L'azionamento diretto si effettua solo per piccoli diametri di passaggio a 2 o 3 vie ; altrimenti si preferiscono valvole servocomandate, in modo da non dover adoperare bobine di dimensioni eccessive.

### 8.1.3 Finecorsa

Nei circuiti elettropneumatici i finecorsa debbono fornire un segnale elettrico. Essi si presentano come dei contatti che chiudono o aprono un circuito.

Possono così essere:

*Finecorsa normalmente aperto NA non toccato*



Esso si chiude quando è toccato. Si rappresenta con un contatto aperto che si chiude ruotando verso destra (*corredato dell'indicazione del tipo di finecorsa: ad esempio rullo di comando*).

*Finecorsa normalmente aperto NA toccato*

Può capitare che, nelle condizioni iniziali, il contatto normalmente aperto del finecorsa risulta chiuso perché toccato. In tal caso il finecorsa NA si rappresenta chiuso ma affiancato da una freccetta (indicante la chiusura del finecorsa NA chiuso perché toccato).

*Finecorsa normalmente chiuso NC non toccato*

Esso si apre quando è toccato. Si rappresenta con un contatto normalmente chiuso che si apre ruotando verso destra (*corredato dell'indicazione del tipo di finecorsa: ad esempio rullo di comando*).

*Finecorsa normalmente chiuso NC toccato*

Nelle condizioni iniziali il finecorsa normalmente chiuso può risultare aperto perché toccato. In tal caso il finecorsa NC si rappresenta chiuso ma affiancato da una freccetta.

fig.8.9

***FINECORSA normalmente aperto NA a rullo non toccato***



***FINECORSA normalmente aperto NC a rullo non toccato***



***FINECORSA normalmente chiuso NC aperto perche' toccato***



***FINECORSA normalmente aperto NA chiuso perche' toccato***



## COSTITUZIONE DI UN CIRCUITO COMBINATORIO ELETTROPNEUMATICO

Il circuito combinatorio elettropneumatico si compone dagli attuatori, comandati dall'elettrovalvole distributrici, alle cui bobine vengono forniti i segnali *ON/OFF* dal circuito elettrico, che determina le funzioni logiche combinatorie, attraverso l'apertura e chiusura dei contatti.

Nel circuito combinatorio non occorrono i segnali di memoria, per cui i segnali sono ottenuti dagli *Start*, dai trasduttori di prossimità come i finecorsa ecc. Questi possono comandare direttamente le bobine dell'elettrovalvole di potenza e quindi fornire la funzione logica di comando.

*Spesso occorre però suddividere il circuito logico di comando a bassa tensione da quello di potenza a più alta tensione.*

In tal caso i finecorsa non comandano direttamente le bobine delle elettrovalvole di potenza. L'apertura o chiusura dei finecorsa comandano l'eccitazione di bobine a bassa tensione i cui contatti forniscono i segnali di comando alle bobine dell'elettrovalvole.

Per una migliore comprensione ci si riferisce a semplici esempi.

---

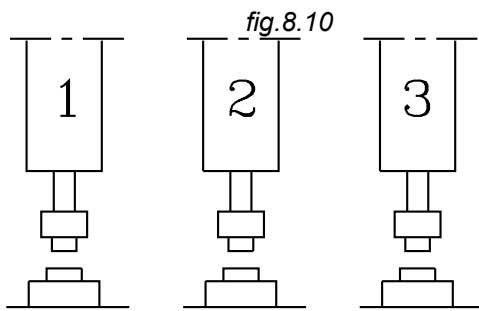
## IN LABORATORIO

---

### 8.2.1 I ESEMPIO

Su una serie di pezzi in uscita da una produzione debbono essere conati, dopo un controllo, a seconda del risultato, tre numeri di qualità diversi.

Per la conatura vengono adoperati tre cilindri, pneumatici, comandati da due pulsanti  $S_1$  e  $S_2$  che eccitano, rispettivamente, due bobine  $X_1, X_2$ .



I cilindri vengono comandati da elettrovalvole 5/2 monostabili.

Il comando avviene nella seguente maniera:

- 1- Pigiando  $S_1$  viene coniato dal cilindro 1 la qualità "1".
- 2- Pigiando  $S_2$  viene coniato dal cilindro 2 la qualità "2".
- 3- Pigiando contemporaneamente entrambi i pulsanti  $S_1 S_2$  viene coniato dal cilindro 3 la qualità "3".

Per la soluzione si effettui la tabelle della verità:

	$x_1$	$x_2$	$Y_1$	$Y_3$	$Y_5$
1	0	0	0	0	0
	0	1	0	1	0
	1	0	1	0	0
	1	1	0	0	1

Dalla tabella della verità si possono ricavare le equazioni logiche:

$$x_1 = S_1$$

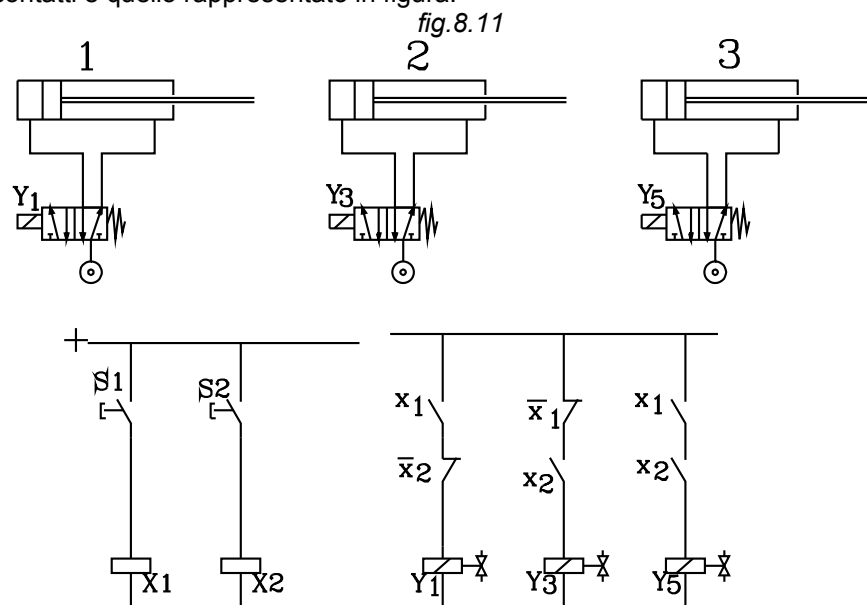
$$x_2 = S_2$$

$$Y_1 = x_1 \cdot \bar{x}_2$$

$$Y_3 = \bar{x}_1 \cdot x_2$$

$$Y_5 = x_1 \cdot x_2$$

Il circuito a contatti è quello rappresentato in figura.



### Stesura del circuito

- Lo start  $S_1$  eccita la bobina  $X_1$ ; all'eccitazione di questa si chiudono i contatti normalmente aperti  $x_1$  e si aprono quelli normalmente chiusi  $\bar{x}_1$ .
- Lo start  $S_2$  eccita la bobina  $X_2$  e si chiudono i suoi contatti normalmente aperti  $x_2$ , mentre si aprono quelli normalmente chiusi  $\bar{x}_2$ .
- Le elettrovalvole 5/2 monostabili vengono indicate con numeri dispari, mancando la seconda bobina di commutazione:

$Y_1$  Elettrovalvola che comanda il cilindro 1.

$Y_3$  Elettrovalvola che comanda il cilindro 2.

$Y_5$  Elettrovalvola che comanda il cilindro 3.

Dalle equazioni logiche si ha:

- 1- l'elettrovalvola  $Y_1$  si eccita quando è eccitata la bobina  $X_1$  e diseccitata la  $X_2$ .  
Infatti in serie alla  $Y_1$  vi è la combinazione dei contatti  $x_1 \cdot \bar{x}_2$ .
- 2- l'elettrovalvola  $Y_3$  si eccita quando è eccitata la bobina  $X_2$  e diseccitata la  $X_1$ .  
Infatti in serie alla  $Y_3$  vi è la combinazione dei contatti  $\bar{x}_1 \cdot x_2$ .
- 3- L'elettrovalvola  $Y_5$  si eccita quando sono eccitate, contemporaneamente le due bobine  $X_1$  e  $X_2$ .  
Infatti in serie alla  $Y_5$  vi è la combinazione dei contatti  $x_1 \cdot x_2$ .

### FUNZIONAMENTO

- Azionando lo start  $S_1$  si eccita la bobina  $X_1$  e si chiudono i contatti  $x_1$ . Osservando i contatti in serie all'elettrovalvola  $Y_1$ , si nota che il contatto  $x_1$  viene chiuso mentre il contatto  $x_2$  normalmente chiuso rimane tale, non essendo eccitata la bobina  $X_2$ .
- Chiudendo  $S_1$  si eccita  $Y_1$  e fuoriesce lo stelo del cilindro 1.
- Osservando le linee di corrente che alimentano le elettrovalvole  $Y_3, Y_5$  si nota che, quando si chiude solamente lo start  $S_1$  esse non vengono eccitate. Infatti nella linea che alimenta  $Y_1$  il contatto  $\bar{x}_1$  normalmente chiuso si apre e  $x_2$ , normalmente aperto, rimane tale. Pigiando  $S_1$  non si ha passaggio di corrente neppure sul ramo che alimenta  $Y_3$ : infatti il contatto  $x_1$  si chiude ma  $x_2$  rimane aperto.
- Chiudendo lo start  $S_2$  si chiudono i contatti normalmente aperti  $x_2$  e si aprono quelli normalmente chiusi  $\bar{x}_2$ . Viene eccitata l'elettrovalvola  $Y_3$ : la corrente passa sulla linea che l'alimenta, trovando  $x_2$  che si è chiuso e  $\bar{x}_1$  che rimane chiuso. Nelle condizioni di  $S_2$  chiuso non vengono eccitate le elettrovalvole  $Y_1, Y_5$ : è interrotta la corrente di alimentazione.
- Pigiando entrambi gli start  $S_1, S_2$ , si eccitano le bobine  $X_1, X_2$ , si chiudono i contatti  $x_1, x_2$ , si eccita l'elettrovalvola  $Y_5$  con fuoriuscita dello stelo del cilindro 5.

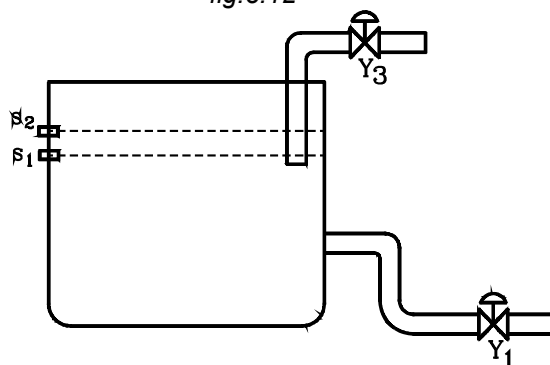
---

## IN LABORATORIO

---

### 8.2.2 II ESEMPIO

fig. 8.12



livello è compreso tra quello minimo e il massimo.

Il livello del liquido contenuto in un serbatoio deve essere compreso tra un limite minimo "min" e uno massimo "Max".

Un sensore di prossimità  $S_1$  rileva il livello minimo e un altro  $S_2$  il livello massimo.

Una lampada rossa "R" deve accendersi quando il livello è, o al di sotto del livello minimo o è al di sopra di quello massimo.

Una lampada verde deve accendersi quando il

Il serbatoio viene alimentato dall'apertura dell'elettrovalvola  $Y_3$  e scaricato dall'apertura dell'elettrovalvola  $Y_1$ .

L'elettrovalvola  $Y_1$  di scarico dovrà eccitarsi quando il livello supera il livello massimo.

L'elettrovalvola  $Y_3$  di alimentazione dovrà eccitarsi quando il livello è al disotto del livello minimo.

Le due elettrovalvole dovranno rimanere chiuse quando il livello è tra il minimo e il massimo.

Soluzione

*NB Per un'esercitazione dimostrativa del funzionamento, a laboratorio, si impieghino due elettrovalvole 5/2 monostabili comandanti due cilindri la cui corsa di fuoriuscita rappresenti il meccanismo di apertura del passaggio dell'acqua nelle tubazioni.*

Tabella della verità

$X_1$	$X_2$	R	V	$Y_1$	$Y_3$	NOTE
0	0	1	0	0	1	Al di sotto del minimo
0	1	0	-	-	-	Evento impossibile
1	0	0	1	0	0	Tra i due limiti
1	1	1	0	1	0	Al di sopra dei limiti

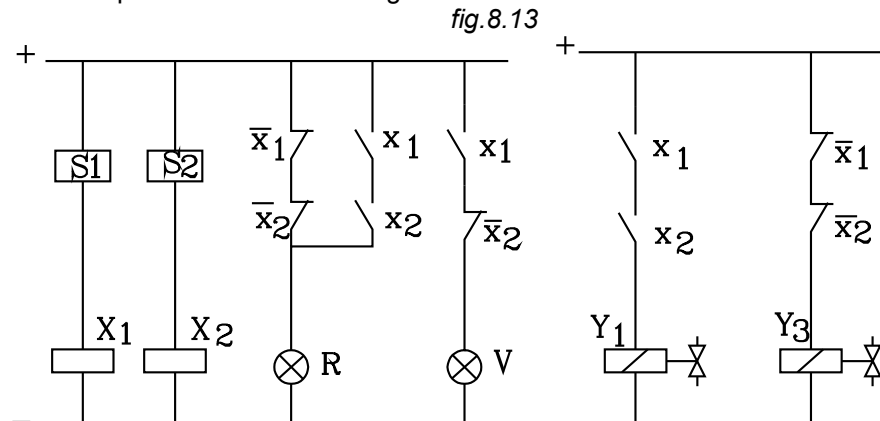
Dalla tabella della verità si possono ricavare le funzioni logiche di comando:

$$\text{Eccitazione delle bobine} \quad \begin{cases} X_1 = S_1 \\ X_2 = S_2 \end{cases}$$

$$\text{Accensione delle lampade} \quad \begin{cases} R = \bar{x}_1 \cdot \bar{x}_2 + x_1 \cdot x_2 \\ V = x_1 \cdot \bar{x}_2 \end{cases}$$

$$\text{Eccitazione elettrovalvole} \quad \begin{cases} Y_1 = x_1 \cdot x_2 \\ Y_3 = \bar{x}_1 \cdot \bar{x}_2 \end{cases}$$

Il circuito a contatti espresso dalle funzioni logiche che soddisfano alla tabella della verità è:



I due sensori sono indicati genericamente con due rettangolini contenenti la sigla identificatrice. (per esercitazione si possono adoperare sensori capacitivi induttivi o semplicemente due start).

Verifica del funzionamento

- Quando non sono attivati i due sensori  $S_1, S_2$  (livello al disotto del minimo) i contatti normalmente chiusi  $\bar{x}_1, \bar{x}_2$  portano la corrente sulla lampada rossa "R" accendendosi; inoltre la corrente attraverso la stessa combinazione raggiunge l'elettrovalvola  $Y_3$  eccitandola (si apre l'ingresso dell'acqua nel serbatoio).
- Quando è attivato il sensore  $S_1$  e disattivato  $S_2$  (livello tra i due limiti) è eccitata solamente la bobina  $X_1$ . In tali condizioni la corrente può scorrere solamente sulla linea che alimenta la lampada verde. Nelle altre linee di corrente si ha un contatto aperto e l'altro chiuso.

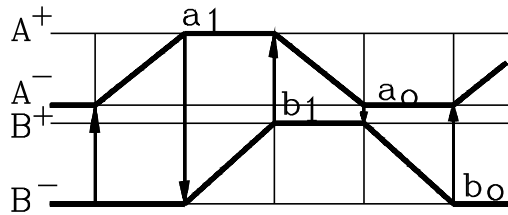
- Quando sono attivati i due sensori  $S_1, S_2$  (livello al sopra del massimo) i contatti normalmente aperti  $x_1, x_2$  portano la corrente sulla lampada rossa "R" accendendosi; inoltre la corrente attraverso la stessa combinazione raggiunge l'elettrovalvola  $Y_1$  eccitandola (si apre la fuoriuscita dell'acqua dal serbatoio).

IN LABORATORIO

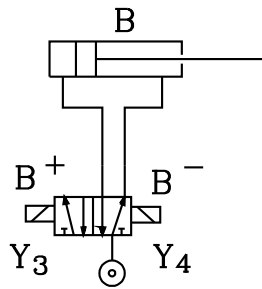
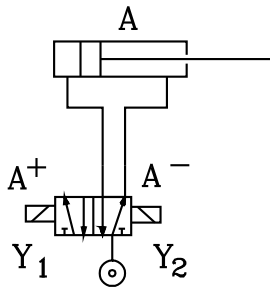
8.2.3 III ESEMPIO

Si debba effettuare il ciclo:

$A^+ B^+ A^- B^-$



Il circuito pneumatico è composto dai due cilindri collegati alle due elettrovalvole 5/2 bistabili.



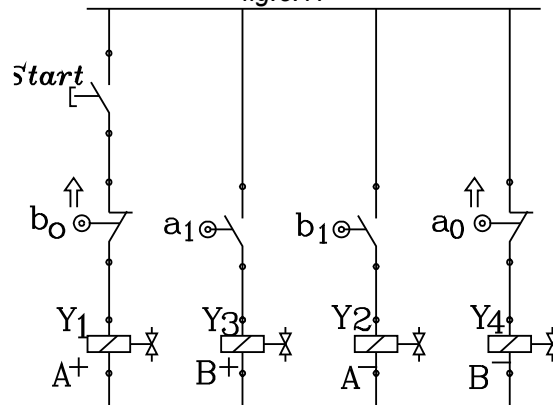
Le equazioni logiche sono:

$$\begin{cases} A^+ = b_0 \cdot Start \\ B^+ = a_1 \\ A^- = b_1 \\ B^- = a_0 \end{cases}$$

*I soluzione - comando diretto*

- Il comando  $A^+$  è dato dalla serie dello *Start* con il finecorsa  $b_0$
- Così la bobina  $Y_1$  dell'elettrovalvola che comanda  $A^+$  è posta direttamente in serie con lo *Start* e il finecorsa  $b_0$  normalmente aperto ma chiuso perché toccato (il finecorsa chiuso è affiancato da una freccia).
- La bobina  $Y_3$  che comanda  $B^+$  è posta in serie con il finecorsa  $a_1$  normalmente aperto e non toccato.
- La bobina  $Y_2$  che comanda  $A^-$  è posta in serie con il finecorsa  $b_1$  normalmente aperto e non toccato.
- La bobina  $Y_4$  che comanda  $B^-$  è posta in serie con il finecorsa  $a_0$  normalmente aperto ma chiuso perché toccato.

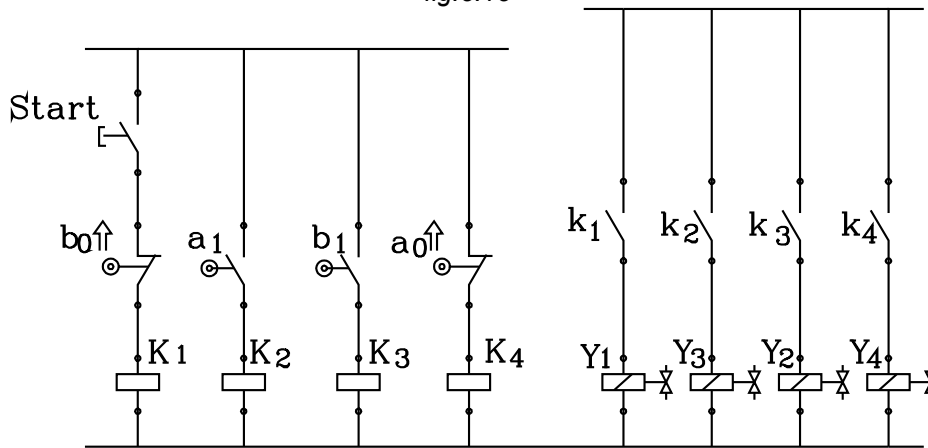
fig.8.17



*II Soluzione - Comando indiretto*

I comandi di finecorsa e lo *Start* non comandano direttamente le bobine delle elettrovalvole ma bobine ausiliarie di comando a bassa tensione i cui contatti forniscono i segnali di comando all'elettrovalvole di potenza.

fig.8.18



### 8.3 MEMORIE

Come già esposto in pneumatica, nei circuiti sequenziali occorre l'impiego di memorie i cui segnali possano essere posti in AND con quelli che, identici nella combinazione, debbono produrre due comandi diversi.

La memoria è un circuito che può avere due stadi opposti di Set e Reset. Il Set dà in uscita un segnale  $y=1$ ; mentre il Reset fornisce il segnale opposto  $y=0$ .

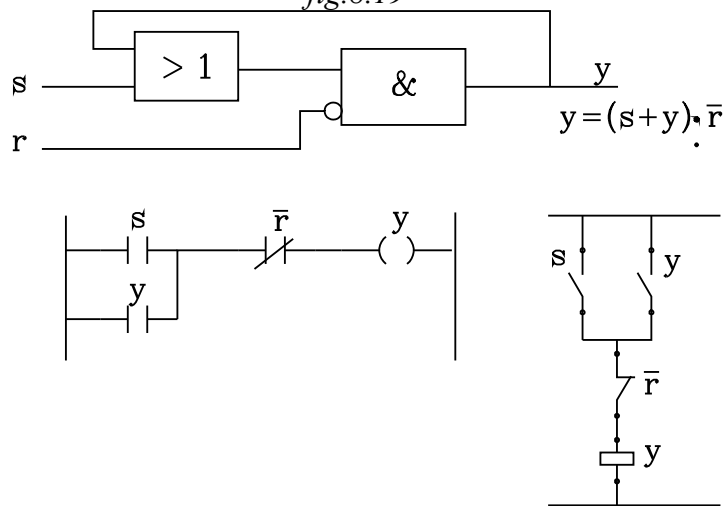
Le memorie bistabili possono essere di due tipi:

- 1°- Con precedenza alla cancellazione.
- 2°- Con precedenza alla ascritta.

#### 8.3.1 Memoria con precedenza alla cancellazione

Nella fig.8.19 viene riportato lo schema logico di memoria.

fig.8.19



Il segnale di Set  $s$  viene posto in parallelo con il segnale di uscita  $y$  e l'uscita  $(s+y)$  dal parallelo viene posta in serie con il segnale di Reset negato  $\bar{r}$ .

Si ottiene così, all'uscita il segnale:

$$y = (s + y) \cdot \bar{r}$$

Dove, come noto, l'uscita  $y$  al secondo membro si riferisce allo stato attuale, mentre quella al primo membro determina lo stato futuro.

Quando  $r=0$  e  $s=1$ , qualunque sia il valore iniziale di  $y$  si ha:

$$y = (1 + y) \cdot 0 \quad \rightarrow \quad y = 1$$

Ottenuto  $y=1$ , restando  $r=0$  e quindi  $\bar{r} = 1$ , anche se il segnale di *Set* si annulla ( $s=0$ ), si automantiene il segnale di uscita.

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

Quando  $r=1$  si ha il resettaggio della memoria. Infatti, qualunque sia lo stato logico dell'uscita  $y$ , quando  $r = 1$ , la variabile  $\bar{r} = \bar{1} = 0$  in AND con il parallelo porta allo stato logico 0 la funzione logica:

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

Nella figura sono riportati i circuiti a contatti nelle due versioni.

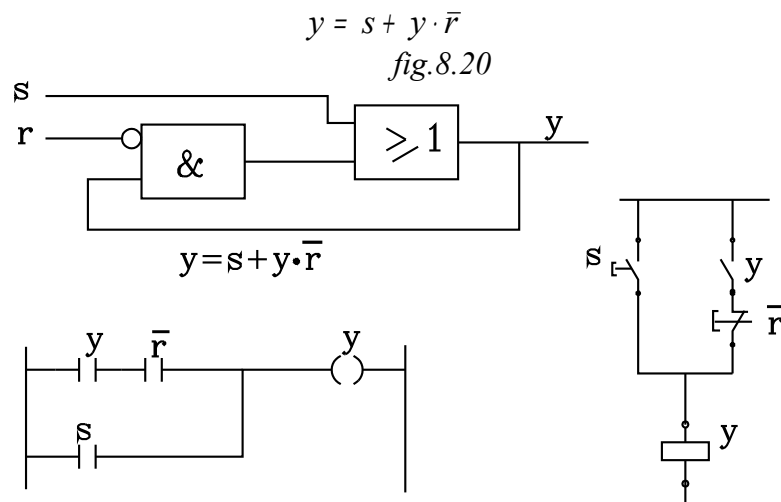
In pratica, con la chiusura di  $s$ , si eccita la bobina, si chiude il contatto  $y$ , che automantiene l'eccitazione della bobina stessa.

Quando si apre il contatto normalmente chiuso in serie al parallelo, in ogni caso, si diseccita la bobina

### 8.3.2 Memoria con precedenza alla scrittura

Per ottenere una memoria nella quale sia preminente il segnale di *Set*, occorre che questi venga posto in *OR* con l'uscita. Occorre cioè scambiare gli operatori combinatori.

Così il segnale di *Reset*  $\bar{r}$  viene posto in *AND* con il segnale di uscita  $y$  e l'uscita dall'*AND* è posta in *OR* con il segnale di *Set*  $s$ .



Risulta prevalente l'operazione *OR* e quindi, comunque sia la condizione iniziale dell'uscita e del segnale di *Reset*, quando  $s=1$  l'uscita  $y$  risulta attivata  $y=1$ .

$$y = 1 + \bar{r} \cdot y = 1$$

## 8.4 Cicli con segnali bloccanti e impiego di valvole bistabili

### 8.4.1 Ciclo ad L

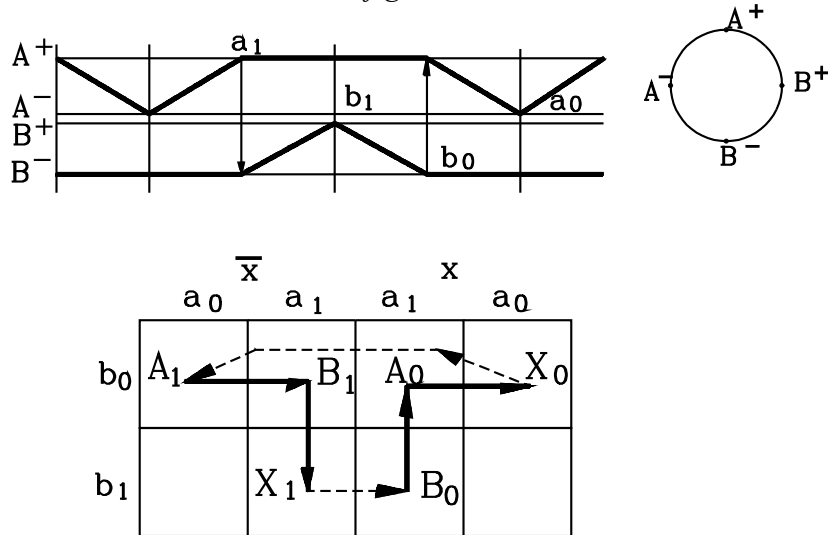
Per spiegare il metodo di soluzione conviene riferirsi ad un esempio pratico.

Consideriamo il ciclo ad L :

$$A^+ B^+ B^- A^-$$

con un segnale bloccante già studiato precedentemente e risolto con circuito pneumatico.

fig.8.21



Il ciclo risolto con la mappa di Karnaugh (punto 6.6.2.1) ha fornito le seguenti equazioni logiche:

$$\begin{aligned}
 A^+ &= \bar{x} \cdot Start \\
 B^+ &= a_1 \cdot \bar{x} \\
 X_1 &= b_1 \\
 B^- &= x \\
 A^- &= b_0 \cdot x \\
 X_0 &= a_0
 \end{aligned}
 \tag{8.4.1}$$

Si considera prima il circuito con comandi diretti.

Dalle equazioni logiche si rileva che occorre una memoria atta a fornire con segnale di Set  $X_1$  l'uscita  $x$  e con un segnale di Reset  $X_0$  l'uscita  $\bar{x}$ .

Viene adoperata una memoria con priorità alla cancellazione.

Il segnale di Set  $X_1$  viene fornito dal finecorsa  $b_1$  che verrà posto in parallelo con il contatto normalmente aperto  $x$  di automantenimento della bobina della memoria  $X$ .

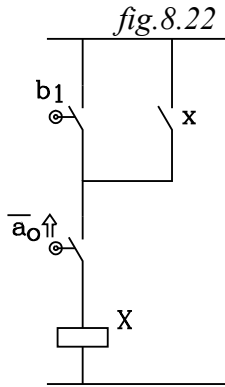
La memoria viene resettata dal segnale  $X_0$  fornito dal fine corsa  $a_0$ , posta in serie con la bobina  $X$ .

Il finecorsa  $a_0$  deve essere un finecorsa normalmente chiuso in modo che quando è toccato apra il circuito e disecciti la bobina della memoria.

$$\text{Memoria } X \begin{cases} \text{Set} & X_1 = b_1 \\ \text{Reset} & X_0 = a_0 \end{cases}$$

$$\text{Equazione logica della memoria è: } X = (b_1 + x) \cdot \bar{a}_0$$





Si tratta di una memoria con precedenza alla cancellazione, nel quale il segnale di settaggio è dato da  $b_1$  e resettaggio da  $a_0$  e si traduce nel circuito a contatti già studiato.

Occorre notare che per effettuare il resettaggio della memoria occorre impiegare il finecorsa  $\bar{a}_0$  normalmente chiuso, che si apre quando viene toccato.

Nelle condizioni iniziali il finecorsa Normalmente Chiuso  $a_0$  risulta aperto perché toccato: ciò è indicato dalla freccia sovrastante.

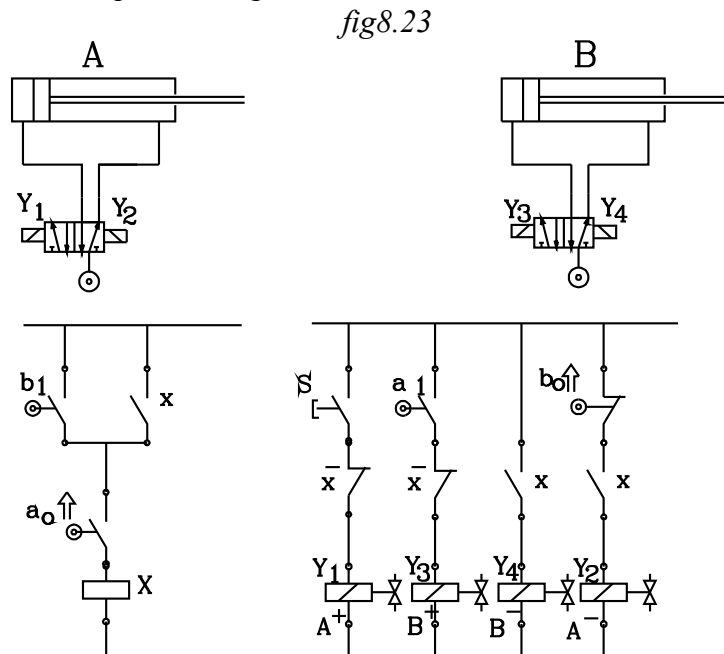
La memoria risulterà settata quando  $b_1=1$  e  $a_0=0$  e risulta:

$$X = (1 + x) \cdot \bar{0} = 1 \cdot 1 = 1$$

Quando  $a_0=1$  allora la memoria viene resettata e risulta :

$$X = (b_1 + x) \cdot \bar{1} = (b_1 + x) \cdot 0 = 0$$

Per stendere il circuito conviene partire dallo schema della memoria; quindi si riportano i circuiti corrispondenti alle equazioni logiche di comando dei cilindri.



- Comando  $A^+$  È ottenuto dalla bobina  $Y_1$  dell'elettrovalvola in serie con  $\bar{x}$  e Start  
 Comando  $B^+$  È ottenuto dalla bobina  $Y_3$  dell'elettrovalvola in serie con  $a_1$  e  $\bar{x}$   
 Comando  $B^-$  È ottenuto dalla bobina  $Y_4$  dell'elettrovalvola in serie con  $x$   
 Comando  $A^-$  È ottenuto dalla bobina  $Y_2$  dell'elettrovalvola in serie con  $b_0$  e  $x$   
 (il finecorsa  $b_0$  è normalmente aperto ma nelle condizioni iniziali chiuso perché toccato).

NB Nei circuiti di questo tipo tutti i finecorsa sono del tipo Normalmente Aperti, salvo quelli che debbono resettare la memoria che risultano Normalmente Chiusi.

Nello schema, come al solito, le bobine dell'elettrovalvole bistabili sono designate con il carattere  $Y$ , corredato da un pedice numerico dispari per le corse di fuoriuscita dei cilindri e da uno pari per le corse di rientro. Così:

$A^+$  → Elettrovalvola  $Y_1$

- $A^-$  → Elettrovalvola  $Y_2$
- $B^+$  → Elettrovalvola  $Y_3$
- $B^-$  → Elettrovalvola  $Y_4$

**Descrizione del funzionamento del circuito**

**Comando diretto**

- Pigiato lo **Start**, attraverso il contatto normalmente chiuso  $\bar{x}$  viene eccitata la bobina  $Y_1$  che fa fuoriuscire lo stelo di A ( $A^+$ ).
- Toccato il finecorsa  $a_1$ , attraverso il contatto NC  $\bar{x}$ , viene eccitata la bobina  $Y_3$  che determina la fuoriuscita dello stelo di B ( $B^+$ ).
- Toccato il finecorsa  $b_1$ , viene settata la memoria  $X$ ; per cui si chiudono i contatti  $x$  e si aprono i contatti  $\bar{x}$ . In tal modo si diseccitano le bobine  $Y_1$   $Y_3$  e si eccita la bobina  $Y_4$ . Questa comanda il rientro dello stelo di B ( $B^-$ ).
- Occorre notare che nell'istante in cui si chiudono i contatti  $x$  è toccato il finecorsa  $b_1$  e quindi  $b_0$  non è toccato. La bobina  $Y_2$  verrà eccitata quando rientra lo stelo di B e viene toccato  $b_0$ : allora viene comandato il rientro di A ( $A^-$ ).
- Rientrato lo stelo di A viene toccato il finecorsa  $a_0$  normalmente chiuso il quale si apre, diseccitando la bobina della memoria  $X$

**Comando indiretto**

Nello schema seguente viene riportato il circuito con comando indiretto.

I comandi previsti nelle funzioni logiche vengono inviati a bobine eccitate a bassa tensione. Dette bobine aprono e chiudono contatti posti in serie alle bobine delle elettrovalvole di potenza a più alta tensione.

Sono state indicate con  $K_1$   $K_2$   $K_3$   $K_4$  le bobine che comandano rispettivamente  $Y_1$   $Y_3$   $Y_4$   $Y_2$

Si ha:

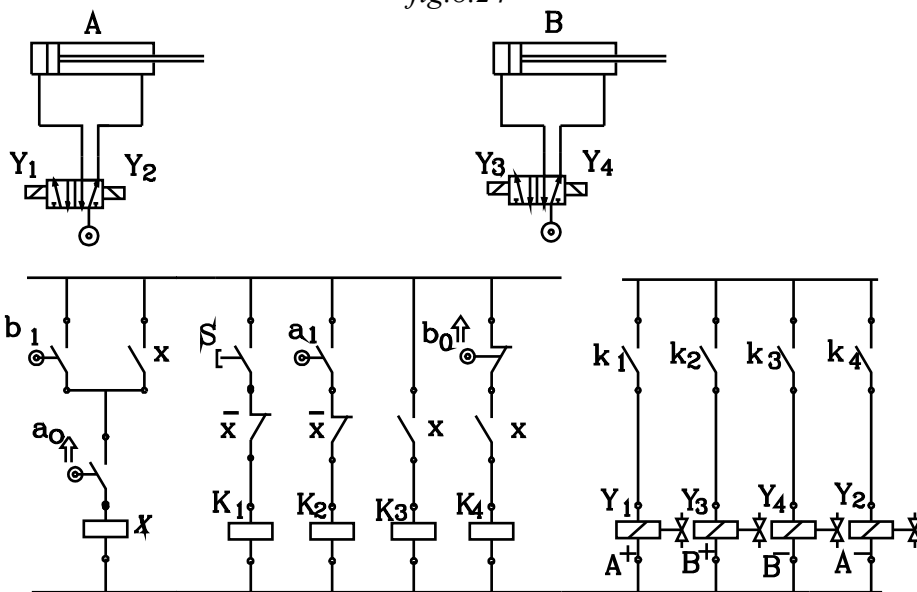
$$A^+ \rightarrow Y_1 = K_1 = \bar{x} \cdot Start$$

$$B^+ \rightarrow Y_3 = K_2 = a_1 \cdot \bar{x}$$

$$B^- \rightarrow Y_4 = K_3 = x$$

$$A^- \rightarrow Y_2 = K_4 = b_0 \cdot x$$

fig.8.24



Funzionamento del comando indiretto

- Pigiato lo Start, attraverso il contatto normalmente chiuso  $\bar{x}$  viene eccitata la bobina  $K_1$  che chiudendo il contatto  $k_1$  in serie con la bobina  $Y_1$  fa fuoriuscire lo stelo di A ( $A^+$ ). Il finecorsa normalmente chiuso  $\bar{a}_0$ , ora non toccato si chiude: si chiude.
- Toccato il finecorsa  $a_1$ , attraverso  $\bar{x}$ , viene eccitata la bobina  $K_2$  che chiude il contatto  $k_2$ ; questo è in serie con  $Y_3$  che determina la fuoriuscita dello stelo di B ( $B^+$ ).
- Toccato il finecorsa  $b_1$ , viene settata la memoria X; per cui si chiudono i contatti x e si aprono i contatti  $\bar{x}$ . In tal modo si diseccitano le bobine  $K_1$   $K_2$  (di conseguenza le elettrovalvole  $Y_1$   $Y_3$ ) e si eccita la bobina  $K_3$  che con il contatto  $k_3$  determina l'eccitazione della bobina dell'elettrovalvola  $Y_4$ . Questa comanda il rientro dello stelo di B ( $B^-$ ).
- Occorre notare che nell'istante in cui si chiudono i contatti x è toccato il finecorsa  $b_1$  e quindi  $b_0$  non è toccato.
- La bobina  $K_4$  verrà eccitata quando rientra lo stelo di B e viene toccato  $b_0$ : si chiuderà allora il contatto  $k_4$  in serie con la bobina dell'elettrovalvola  $Y_2$  che comanda il rientro di A ( $A^-$ ).
- Rientrato lo stelo di A viene toccato il finecorsa  $a_0$  normalmente chiuso il quale si apre, diseccitando la bobina della memoria X

---

IN LABORATORIO

---

### 8.4.2 Ciclo con due memorie

Si debba effettuare il seguente ciclo con 3 cilindri, 2 elettrovalvole 5/2 bistabili e 6 finecorsa:

$$A^+ B^+ B^- C^+ C^- A^-$$

fig. 8.25

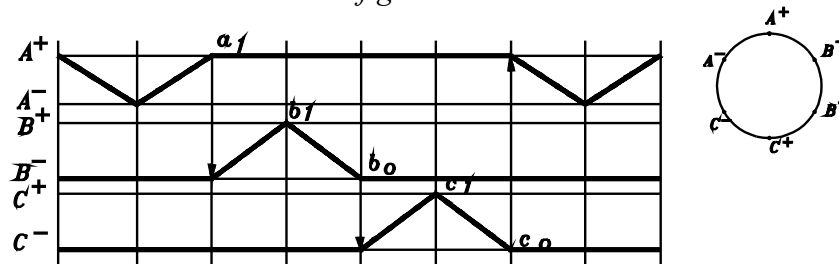


Tabella degli ordini

Stato comandato	ordine	variabile	posizione raggiunta
$A^+$	$A_1$	$a_0 b_0 c_0$	$\rightarrow a_1 b_0 c_0$
$B^+$	$B_1$	$a_1 b_0 c_0$	$\rightarrow a_1 b_1 c_0$
$B^-$	$B_0$	$a_1 b_1 c_0$	$\rightarrow a_1 b_0 c_0$
$C^+$	$C_1$	$a_1 b_0 c_0$	$\rightarrow a_1 b_0 c_1$
$C^-$	$C_0$	$a_1 b_0 c_1$	$\rightarrow a_1 b_0 c_0$
$A^-$	$A_0$	$a_1 b_0 c_0$	$\rightarrow a_0 b_0 c_0$

Per effettuare la mappa si opera nella stessa maniera descritta nel ciclo a L.

Comando  $A_1$ :

Vettore che congiunge le caselle  $a_0 b_0 c_0 \rightarrow a_1 b_0 c_0$

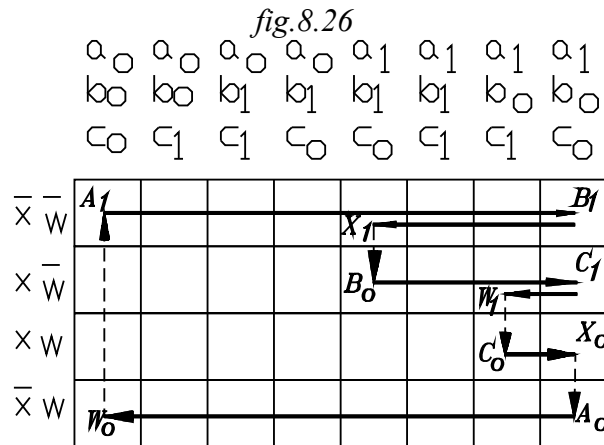
Comando  $B_1$

Vettore che congiunge le caselle  $a_1 b_0 c_0 \rightarrow a_1 b_1 c_0$

Occorre ora comandare il rientro dello stelo di B, ma ciò porta a tornare nella casella  $a_1 b_0 c_0$  già utilizzata per far fuoriuscire lo stelo di B.

Occorre introdurre una memoria X che fornisce i due segnali  $x - \bar{x}$  e raddoppiare la mappa

(Le caselle precedenti si riferiscono allo stato  $\bar{x}$  di memoria non settata)



**Segnale  $X_1$  di settaggio memoria**

Vettore tratteggiato che congiunge  $\bar{x}a_1b_1c_0 \rightarrow xa_1b_1c_0$

**Comando  $B_0$**

Vettore che congiunge le caselle  $xa_1b_1c_0 \rightarrow xa_1b_0c_0$

**Comando  $C_1$**

Vettore che congiunge le caselle  $xa_1b_0c_0 \rightarrow xa_1b_0c_1$

Con il comando seguente  $C_0$ , se non si attivasse un'altra memoria, si tornerebbe nella precedente casella.

Occorre attivare una memoria W che fornisce i due segnali  $w - \bar{w}$  e raddoppiare la mappa.

(Le precedenti caselle saranno contraddistinte dal segnale  $\bar{w}$  della memoria non settata)

**Segnale  $W_1$  di settaggio memoria**

Vettore tratteggiato che congiunge  $x\bar{w}a_1b_0c_1 \rightarrow xwa_1b_0c_1$

**Comando  $C_0$**

Vettore che congiunge le caselle  $xwa_1b_0c_1 \rightarrow xwa_1b_0c_0$

Prima di ultimare il ciclo occorre resettare le memorie X e W.

In ordine: si resettì prima la X, quindi si effettuò il comando  $A_0$  e, alla fine, si resettì la W.

**Segnale  $X_0$  di resettaggio memoria**

Vettore tratteggiato che congiunge  $xwa_1b_0c_0 \rightarrow \bar{x}wa_1b_0c_0$

**Comando  $A_0$**

Vettore che congiunge le caselle  $\bar{x}wa_1b_0c_0 \rightarrow \bar{x}wa_0b_0c_0$

**Segnale  $W_0$  di resettaggio memoria**

Vettore tratteggiato che congiunge  $\bar{x}wa_0b_0c_0 \rightarrow \bar{x}\bar{w}a_0b_0c_0$

**Estrazione delle equazioni logiche**

**Stato comandato  $A^+$**

- Il comando  $A_1$  è preceduto da  $W_0$
- Dalla precedente casella contenente il comando  $W_0$  a quella attuale contenente  $A_1$  cambia la variabile da  $w$  in  $\bar{w}$
- La variabile attiva è  $\bar{w}$

- Nella zona di  $\bar{w}$  non c'è il comando opposto  $A_0$ :

$$A^+ = \bar{w} \cdot Start$$

#### Stato comandato $B^+$

- $B_1$  è preceduto da  $A_1$ .
- Da  $A_1$  a  $B_1$  varia  $a_1$ .
- La variabile attiva è  $a_1$ .
- Nella zona di  $a_1$  vi è il comando opposto  $B_0$ .
- Occorre porre in AND con  $a_1$  i segnali di memoria  $\bar{x} \cdot \bar{w}$ :

$$B^+ = a_1 \cdot \bar{x} \cdot \bar{w}$$

#### Attivazione della memoria $X_1$

- $X_1$  è preceduto da  $B_1$ .
- Cambia la variabile da  $b_0$  in  $b_1$ .
- La variabile attiva è  $b_1$ .
- Nella zona di  $b_1$  non vi è il comando opposto di disattivazione  $X_0$ :

$$X_1 = b_1$$

#### Stato comandato $B^-$

- $B_0$  è preceduto da  $X_1$ .
- Cambia la variabile da  $\bar{x}$  in  $x$ .
- La variabile attiva è  $x$ .
- Nelle caselle che si riferiscono alla variabile  $x$  non vi è il comando opposto  $B_1$ :

$$B^- = x$$

#### Stato comandato $C^+$

- $C_1$  è preceduto da  $B_0$ .
- Cambia la variabile da  $b_1$  a  $b_0$ .
- La variabile attiva è  $b_0$ .
- Nella zona di  $b_0$  vi è il comando opposto  $C_0$ .
- Occorre porre in AND con  $b_0$  i segnali di memoria  $x \cdot \bar{w}$ :

$$C^+ = b_0 \cdot x \cdot \bar{w}$$

#### Attivazione della memoria $W_1$

- $W_1$  è preceduto da  $C_1$ .
- Cambia la variabile da  $c_0$  a  $c_1$ .
- La variabile attiva è  $c_1$ .
- Nella zona di  $c_1$  non vi è il segnale opposto di disattivazione  $W_0$ :

$$W_1 = c_1$$

#### Stato comandato $C^-$

- $C_0$  è preceduto da  $W_1$ .
- Cambia la variabile da  $\bar{w}$  a  $w$ .
- La variabile attiva è  $w$ .
- Nelle caselle che si riferiscono alla variabile  $w$  non vi è il comando opposto  $C_1$ :

$$C^- = w$$

#### Disattivazione della memoria $X_0$

- $X_0$  è preceduto da  $C_0$ .
- Cambia la variabile da  $c_1$  a  $c_0$ .
- La variabile attiva è  $c_0$ .
- Nella zona di  $c_0$  vi è il comando opposto  $C_1$ .
- Occorre porre in AND con  $c_0$  i segnali di memoria  $x \cdot w$ .

$$X_0 = c_0 \cdot x \cdot w$$

#### Stato comandato $A^-$

- $A_0$  è preceduto da  $X_0$ .
- Cambia la variabile da  $x$  a  $\bar{x}$ .
- La variabile attiva è  $\bar{x}$ .
- Nella zona di  $\bar{x}$  vi è il comando opposto  $A_1$ .
- Occorre porre in AND con  $\bar{x}$  il segnale di memoria  $w$ :

$$A^- = \bar{x} \cdot w$$

#### Disattivazione della memoria $W_0$

- $Y_0$  è preceduto da  $A_0$ .
- Cambia la variabile da  $a_1$  ad  $a_0$ .
- La variabile attiva è  $a_0$ .
- Nella zona di  $a_0$  non vi è il segnale opposto  $W_1$ :

$$W_0 = a_0$$

Le funzioni logiche risultano:

$$A^+ = \bar{w} \cdot Start$$

$$B^+ = a_1 \cdot \bar{x} \cdot \bar{w}$$

$$X_1 = b_1$$

$$B^- = x$$

$$C^+ = b_0 \cdot x \cdot \bar{w}$$

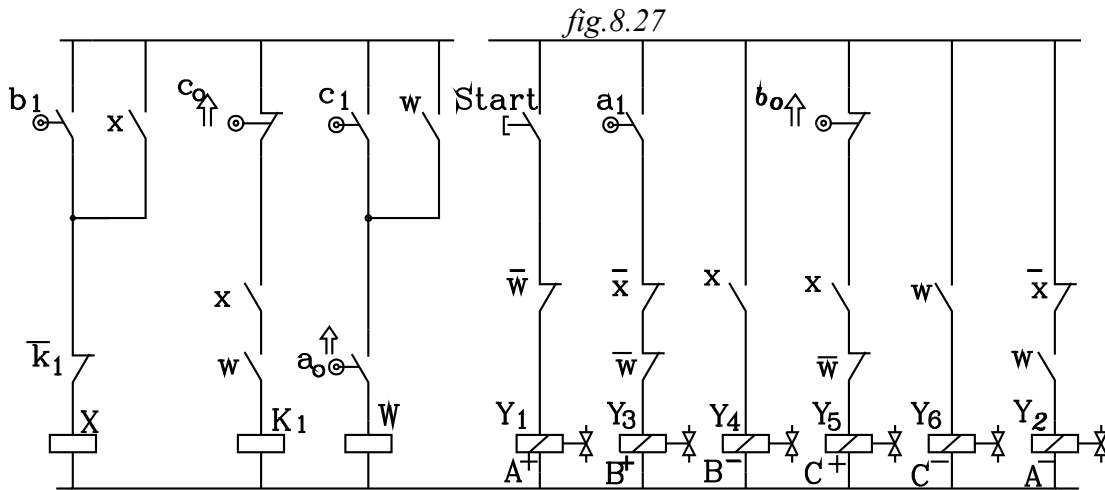
$$W_1 = c_1$$

$$C^- = w$$

$$X_0 = c_0 \cdot x \cdot w$$

$$A^- = \bar{x} \cdot y$$

$$W_0 = a_0$$



*Costruzione del circuito*

- Conviene iniziare a stendere lo schema dalle due memorie  $X, W$  con priorità alla cancellazione.

$$I \text{ memoria } X \left\{ \begin{array}{l} \text{Segnale di Set} \quad X_1 = b_1 \\ \text{Segnale di Reset} \quad X_0 = c_0 \cdot x \cdot w \end{array} \right.$$

- Equazione logica della memoria  $X$ :

$$X = (b_1 + x) \cdot \overline{c_0 \cdot x \cdot w}$$

- Il fincorsa normalmente aperto  $b_1$  viene posto in parallelo al contatto normalmente aperto  $x$  di autoritenuta della bobina della memoria  $X$

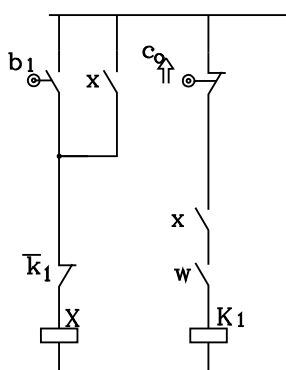


fig.8.28

- Il resettaggio della memoria  $X$  è determinato dalla combinazione dei segnali in AND tra loro e negati:

$$\overline{c_0 \cdot x \cdot w}$$

Conviene assumere una bobina ausiliaria  $K_1$  che viene eccitata dalla combinazione  $c_0 \cdot x \cdot w$ :

$$K_1 = c_0 \cdot x \cdot w$$

- Quindi si pone come resettaggio della memoria X un contatto normalmente chiuso della bobina ausiliaria  $K_1$

- Si sostituisce così al posto di  $\overline{c_0 \cdot x \cdot w}$  il contatto normalmente chiuso  $\overline{k_1}$ .

$$\overline{k_1} = \overline{c_0 \cdot x \cdot w}$$

L'equazione della memoria X diviene:

$$\begin{cases} X = (b_1 + x) \cdot \overline{k_1} \\ \text{dove} \\ K_1 = c_0 \cdot x \cdot w \end{cases}$$

- Analizzando sia il circuito che le equazioni logiche proposte si può constatare che quando si verifica la combinazione  $c_0 \cdot x \cdot w$  che deve resettare la memoria, la bobina ausiliaria  $K_1$  si eccita:

$$K_1 = 1$$

Il contatto normalmente chiuso  $\overline{k_1}$  si apre e si resetta la memoria:

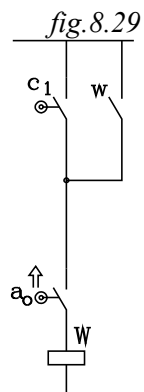
$$X = (b_1 + x) \cdot \overline{1} = (b_1 + x) \cdot 0 = 0$$

In generale quando il segnale di resettaggio di una memoria è costituito da una combinazione di segnali, allora questi si assumono come segnali di eccitazione di una bobina ausiliaria K. Un contatto normalmente chiuso di questa viene impiegato per la desiderata diseccitazione della memoria

$$\text{Il memoria } W \begin{cases} \text{Segnale di Set} & W_1 = c_1 \\ \text{Segnale di Reset} & W_0 = a_0 \end{cases}$$

- Equazione logica della memoria W

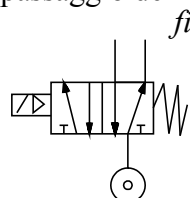
$$W = (c_1 + w) \cdot a_0$$



- Il finecorsa normalmente aperto  $c_1$  si pone in parallelo con il contatto  $w$  di autoritenuta.
- In serie con la bobina della memoria W si pone il finecorsa  $a_0$  che deve essere normalmente chiuso in modo che quando viene attivato esso apra il circuito e resetti la memoria.
- Nelle condizioni iniziali il finecorsa  $a_0$  è toccato e quindi risulta aperto.
- Per il restante circuito basta seguire le equazioni logiche già scritte.

## 8.5 CIRCUITI CON IMPIEGO DI VALVOLE MONOSTABILI

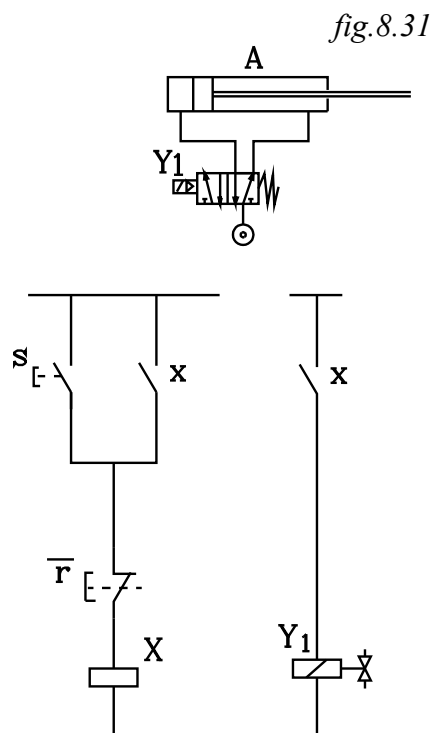
Nella valvola monostabile vi è una sola bobina che comanda il flusso dell'aria. Con il segnale di Set essa viene commutata e si ha il passaggio dell'aria in un senso. Tolto il segnale, la valvola, automaticamente, torna nella posizione di riposo, con conseguente commutazione e passaggio del fluido in senso opposto.



Occorre quindi inviare all'unica bobina della valvola un solo segnale che può assumere i due valori opposti 1,0 per attivarla o disattivarla e ottenere le due posizioni opposte di commutazione.

Spesso occorre che il segnale di attivazione della valvola debba restare memorizzato stabilmente fino a che non si invii un segnale di Reset che la disattivi commutandola. In tal caso il segnale di attivazione o disattivazione della bobina dell'elettrovalvola è ottenuto con una memoria con priorità alla cancellazione.

Si supponga di dover comandare con un'elettrovalvola monostabile un cilindro A che deve effettuare le due corse  $A^+$   $A^-$  comandate rispettivamente da un segnale di Set "s" e uno di Reset "r"



Allora un segnale di Set  $s$ , posto in OR con il contatto di autoritenuta  $x$ , eccita la memoria  $X$  che chiude il contatto  $x$  in serie con la bobina  $Y_1$  dell'elettrovalvola, la quale effettua la corsa  $A^+$ .

Quando si attiva il segnale di Reset  $\bar{r}$ , posto in serie con la bobina della memoria  $X$ , questa viene diseccitata, provocando l'apertura del contatto  $x$ , in serie con la bobina dell'elettrovalvola. Questa, non essendo più alimentata, viene commutata dalla molla: essa torna nella posizione iniziale e comanda la corsa di ritorno del cilindro  $A^+$

Il segnale di Set si ottiene con un contatto normalmente aperto  $s$ , mentre il segnale di Reset deve essere un contatto normalmente chiuso  $\bar{r}$ , in modo da aprire il circuito quando viene attivato.

La bobina della memoria, attraverso un contatto normalmente aperto da essa comandato, eccita o diseccita la bobina dell'elettrovalvola.

$$X = (s + x) \cdot \bar{r}$$

$$Y_1 = x$$

---

### IN LABORATORIO

---

#### 8.5.1 Ciclo senza segnali bloccanti

Si inizi a considerare un circuito semplice senza segnali bloccanti in cui le memorie servono solamente per l'utilizzo delle valvole monostabili.

##### CICLO QUADRO $A^+ B^+ A^- B^-$ con valvole monostabili

Per brevità si svolge solamente la parte riguardante la formulazione delle equazioni logiche di comando del ciclo, con l'estrazione di quelle occorrenti per comandare le memorie da abbinare ad una valvola monostabile

Come già noto le equazioni logiche di comando provenienti dalle elettrovalvole sono:



$$A^+ = b_0 \cdot Start$$

$$B^+ = a_1$$

$$A^- = b_1$$

$$B^- = a_0$$

Non vi sono segnali bloccanti : non occorrono quindi memorie ausiliarie da aggiungere a quelle abbinare alle valvole monostabili.

Si impiegano due elettrovalvole monostabili.

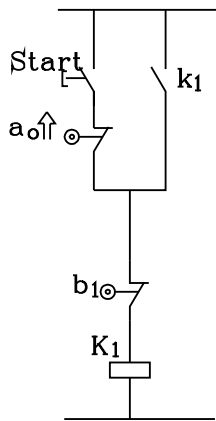
Si indica con  $Y_1$  la bobina dell'elettrovalvola che comanda il cilindro A , con  $Y_3$  quella dell'elettrovalvola che comanda il cilindro B. (Mancano, rispetto alle elettrovalvole bistabili le bobine  $Y_2$   $Y_4$  sostituite dalle molle).

Si indicano con  $K_1, K_3$  le due bobine di memoria ausiliarie che comandano rispettivamente le elettrovalvole  $Y_1 Y_3$ .

$$Y_1 = k_1$$

$$Y_3 = k_3$$

fig.8.32



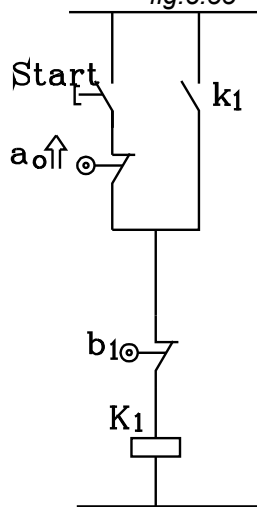
La bobina  $K_1$  di memoria comanda con il suo segnale di Set la corsa di fuoriuscita  $A^+$  del cilindro A, mentre col segnale di Reset comanda la corsa di ritorno  $A^-$

$$\text{Memoria } K_1 \left\{ \begin{array}{l} \text{Segnale di Set} \\ \text{Segnale di Reset} \end{array} \right. \quad \begin{array}{l} A^+ = b_0 \cdot Start \\ A^- = b_1 \end{array}$$

L'equazione logica della memoria  $K_1$  risulta:

$$K_1 = (b_0 \cdot Start + k_1) \cdot \bar{b}_1$$

fig.8.33



La bobina  $K_3$  di memoria comanda con il suo segnale di Set la corsa di fuoriuscita  $B^+$  del cilindro B, mentre col segnale di Reset comanda la corsa di ritorno  $B^-$

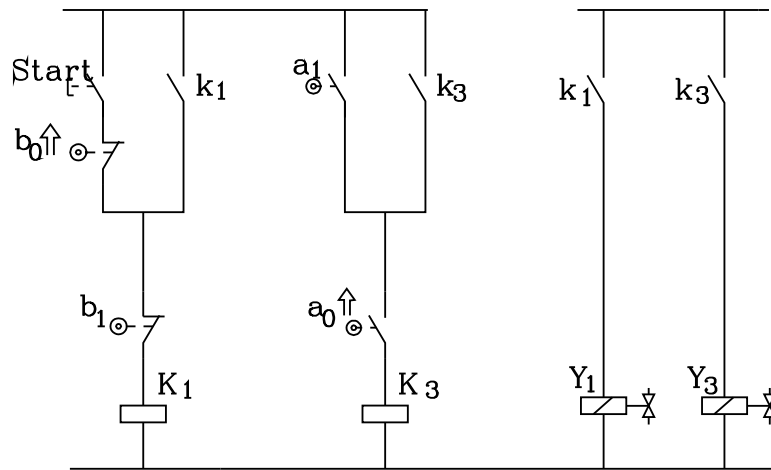
$$\text{Memoria } K_3 \left\{ \begin{array}{l} \text{Segnale di Set} \\ \text{Segnale di Reset} \end{array} \right. \quad \begin{array}{l} B^+ = a_1 \\ B^- = a_0 \end{array}$$

L'equazione logica della memoria  $K_3$  risulta:

$$K_3 = (a_1 + k_2) \cdot \bar{a}_0$$

Si ottiene il circuito di fig.8.34:

fig.8.34



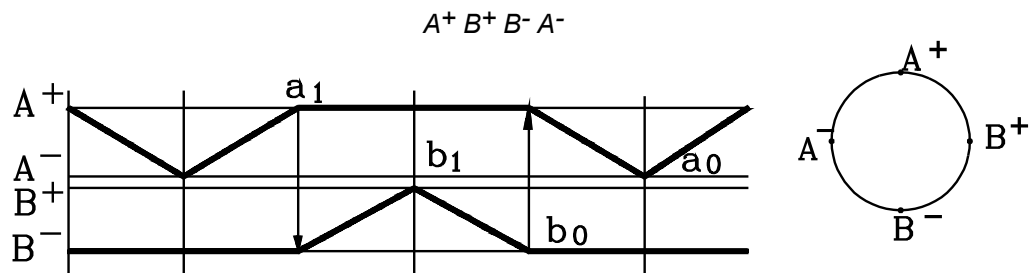

---

*IN LABORATORIO*

---

### 8.5.2 Impiego delle valvole monostabili in un ciclo con un segnale bloccante

Con l'impiego delle elettrovalvole monostabili si debba effettuare il ciclo ad  $L$  già attuato con quelle bistabili:



Le equazioni logiche sono state già determinate (punti 6.6.2.1 e 8.4.1)

$$\begin{aligned}
 A^+ &= \bar{x} \cdot Start \\
 B^+ &= a_1 \cdot \bar{x} \\
 X_1 &= b_1 \\
 B^- &= x \\
 A^- &= b_0 \cdot x \\
 X_0 &= a_0
 \end{aligned}
 \tag{8.5.2}$$

In questo caso, oltre alle due memorie di comando dei monostabili, occorre impiegare la memoria  $X$  necessaria per distinguere il segnale bloccante identico nel I e IV passo  $a_1 b_0$

$$\text{Memoria } X \begin{cases} \text{Set} & X_1 = b_1 \\ \text{Reset} & X_0 = a_0 \end{cases}$$

Equazione logica della memoria  $X$ :

$$X = (b_1 + x) \cdot \bar{a}_0$$

Si indicano con  $K_1, K_3$  le due bobine di memoria ausiliarie che comandano rispettivamente le elettrovalvole  $Y_1 Y_3$ :

$$\begin{aligned}
 Y_1 &= k_1 \\
 Y_3 &= k_3
 \end{aligned}$$

La bobina  $K_1$  di memoria comanda con il suo segnale di Set la corsa di fuoriuscita  $A^+$  del cilindro A, mentre col segnale di Reset comanda la corsa di ritorno  $A^-$

$$\text{Memoria } K_1 \left\{ \begin{array}{l} \text{Segnale di Set} \quad A^+ = \bar{x} \cdot \text{Start} \\ \text{Segnale di Reset} \quad A^- = x \cdot b_0 \end{array} \right.$$

Equazione logica della memoria:

$$K_1 = (\bar{x} \cdot \text{start} + k_1) \cdot \overline{x \cdot b_0}$$

Il segnale di Reset è dato dalla combinazione " $x \cdot b_0$ ".

Si assume una bobina ausiliaria  $H$  che viene eccitata dalla combinazione " $x \cdot b_0$ ". Un contatto  $\bar{h}$  normalmente chiuso di  $H$  effettua il reset della memoria  $X$ .

Si pone:

$$H = x \cdot b_0$$

L'equazione logica della memoria  $K_1$  risulta:

$$K_1 = (\bar{x} \cdot \text{start} + k_1) \cdot \bar{h}$$

La bobina  $K_3$  di memoria comanda con il suo segnale di Set la corsa di fuoriuscita  $B^+$  del cilindro B, mentre col segnale di Reset comanda la corsa di ritorno  $B^-$

$$\text{Memoria } K_3 \left\{ \begin{array}{l} \text{Segnale di Set} \quad B^+ = \bar{x} \cdot a_1 \\ \text{Segnale di Reset} \quad B^- = x \end{array} \right.$$

L'equazione logica della memoria  $K_3$  risulta:

$$K_3 = (\bar{x} \cdot a_1 + k_3) \cdot \bar{x}$$

Sviluppando l'espressione si ottiene:

$$K_3 = \bar{x} \cdot a_1 + k_3 \cdot \bar{x}$$

Raccogliendo  $\bar{x}$  si ottiene:

$$K_3 = (a_1 + k_3) \cdot \bar{x}$$

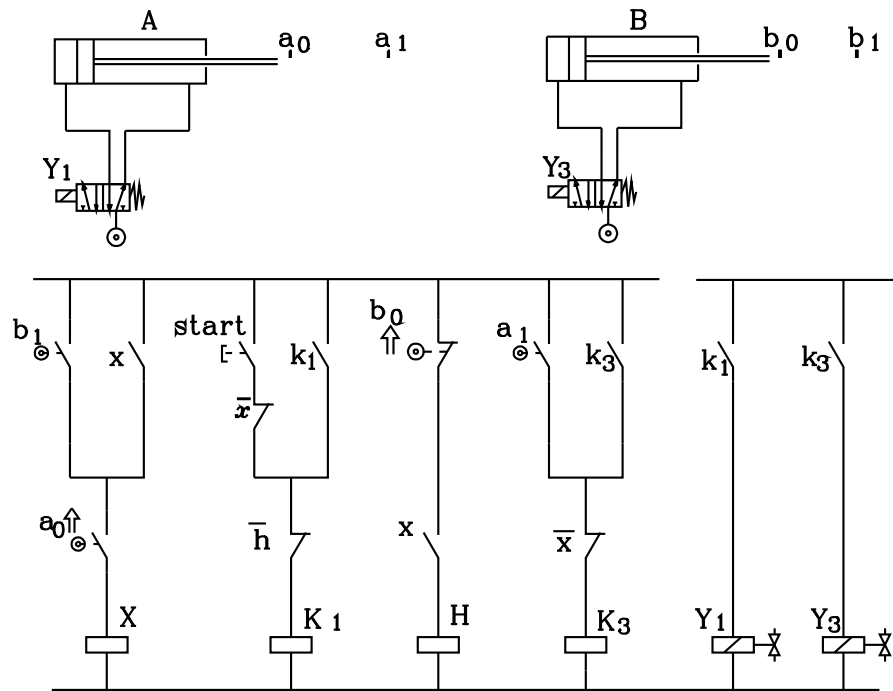
Riassumendo si ha:

$$X = (b_1 + x) \cdot \bar{a}_0 \quad K_1 = (\bar{x} \cdot \text{start} + k_1) \cdot \bar{h} \quad ; \quad H = x \cdot b_0 \quad ; \quad K_3 = (a_1 + k_3) \cdot \bar{x}$$

$$\begin{array}{l} Y_1 = k_1 \\ Y_3 = k_3 \end{array}$$

Dalle equazioni logiche si ottiene il circuito:

fig.8.36



#### Descrizione del funzionamento

- Pigiato lo Start, si eccita la bobina  $K_1$ , si chiude il relativo contatto  $k_1$  normalmente aperto e si eccita la bobina dell'elettrovalvola  $Y_1$ .
- Commutata  $Y_1$ , il cilindro A effettua la corsa di lavoro  $A^+$ . Toccato il finecorsa  $a_1$ , si eccita la bobina  $K_3$ , si chiude il relativo contatto NA  $k_3$  e si eccita la bobina dell'elettrovalvola  $Y_3$ .
- Commutata la elettrovalvola  $Y_3$ , il cilindro B effettua la corsa di lavoro  $B^+$ .
- Toccato il finecorsa  $b_1$ , si eccita la memoria X (il finecorsa  $a_0$ , normalmente chiuso, è chiuso perché non toccato, essendo lo stelo di A fuoriuscito)
- Eccitata la bobina X, si chiudono i contatti  $x$  e si aprono quelli NC  $\bar{x}$
- Chiuso il contatto  $x$  la bobina H non si eccita, in quanto il finecorsa  $b_0$ , normalmente aperto, non è toccato, essendo lo stelo di B fuoriuscito.
- Invece, aperto il contatto NC  $\bar{x}$ , si diseccita la bobina  $K_3$ , si aprono i relativi contatti  $k_3$  e si diseccita l'elettrovalvola  $Y_3$ : questa commuta il flusso dell'aria nel cilindro B che effettua la corsa di ritorno  $B^-$ .
- Toccato  $b_0$ , ora può eccitarsi la bobina H, che ha in serie  $b_0 \cdot x$ , in questo istante entrambi chiusi. La bobina H apre il relativo contatto NC  $h$ , in serie con la bobina  $K_1$  che si diseccita.
- Diseccitata la bobina  $K_1$ , si apre il relativo contatto NA  $k_1$  e si diseccita la bobina dell'elettrovalvola  $Y_1$  che commuta il flusso dell'aria nel cilindro A il quale effettua la corsa di ritorno  $A^-$ .
- Toccato  $a_0$ , questo, essendo un contatto normalmente chiuso, attivato si apre e diseccita la bobina della memoria X: si aprono i contatti NA  $x$  e si richiudono quelli NC  $\bar{x}$  ripristinando le condizioni iniziali.

---

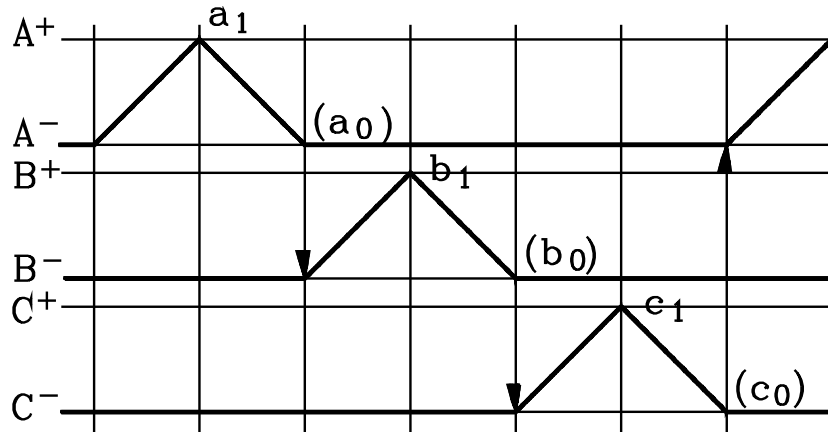
### IN LABORATORIO

---

#### 8.5.3 Impiego delle valvole monostabili in un ciclo con 2 segnali bloccanti

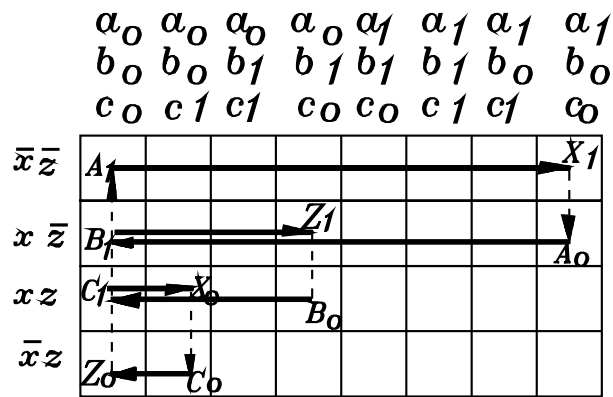
CICLO  $A^+ A^- B^+ B^- C^+ C^-$  con valvole monostabili

fig.8.37



La mappa di Karnaugh è la seguente

fig.8.38



Per chiarezza di simbolismo, si rappresenta con X la prima memoria e con Z la seconda (era stata indicata con Y). Ciò perché nelle esercitazioni vengono designate con  $Y_1, Y_2...$  le bobine delle elettrovalvole (attrezzature FESTO).

Le bobine delle elettrovalvole monostabili vengono indicate con pedici espressi con numeri dispari (manca la seconda bobina).

Le equazioni logiche sono state già precedentemente determinate (punto 6.6.2.2). Esse sono:

$$\begin{aligned}
 A^+ &= \bar{x} \cdot \bar{z} \cdot Start & X_0 &= c_1 \\
 X_1 &= a_1 & C^- &= \bar{x} \\
 A^- &= x & Z_0 &= c_0 \cdot \bar{x} \cdot z \\
 B^+ &= a_0 \cdot x \cdot \bar{z} \\
 Z_1 &= b_1 \\
 B^- &= z \\
 C^+ &= b_0 \cdot x \cdot z
 \end{aligned}$$

Memorie

Equazione della prima memoria X

$$\text{Memoria X} \begin{cases} \text{Set} & X_1 = a_1 \\ \text{Re set} & X_0 = c_1 \end{cases}$$

Equazione logica

$$X = (a_1 + x) \cdot \bar{c}_1$$

Equazione logica della seconda memoria Z

$$\text{Memoria Z} \begin{cases} \text{Set} & Z_1 = b_1 \\ \text{Reset} & Z_0 = c_0 \cdot \bar{x} \cdot z \end{cases}$$

Equazione logica

$$Z = (b_1 + z) \cdot \overline{c_0 \cdot \bar{x} \cdot z}$$

Si assume una bobina ausiliaria per il Reset di Z. Si pone:

$$H = c_0 \cdot \bar{x} \cdot z$$

L'equazione logica della memoria Z diviene:

$$Z = (b_1 + z) \cdot \bar{H}$$

#### Comando bobine dell'elettrovalvole monostabili

Le bobine dell'elettrovalvole monostabili sono indicate con  $Y_1, Y_3, Y_5$  e precisamente:

$Y_1$	Comanda il cilindro A	$\begin{cases} \text{Segnale di Set} & \rightarrow A^+ \\ \text{Segnale di Reset} & \rightarrow A^- \end{cases}$
$Y_3$	Comanda il cilindro B	$\begin{cases} \text{Segnale di Set} & \rightarrow B^+ \\ \text{Segnale di Reset} & \rightarrow B^- \end{cases}$
$Y_5$	Comanda il cilindro C	$\begin{cases} \text{Segnale di Set} & \rightarrow C^+ \\ \text{Segnale di Reset} & \rightarrow C^- \end{cases}$

Le bobine dell'elettrovalvole monostabili vengono comandate, ciascuna, da un contatto normalmente aperto, appartenente ad una memoria che fornisce con il segnale di Set il comando di fuoriuscita dello stelo e con il segnale di Reset il comando di rientro.

Memoria  $K_1$  → Comanda l'elettrovalvola  $Y_1$   $Y_1=K_1$

Memoria  $K_3$  → Comanda l'elettrovalvola  $Y_3$   $Y_3=K_3$

Memoria  $K_5$  → Comanda l'elettrovalvola  $Y_5$   $Y_5=K_5$

Equazioni logiche delle memorie associate all'elettrovalvole:

1- Comando elettrovalvola  $Y_1$

$$\text{Memoria } K_1 \begin{cases} \text{Set} & A^+ = \bar{x} \cdot \bar{z} \cdot \text{Start} \\ \text{Reset} & A^- = x \end{cases}$$

Equazione memoria  $K_1$

$$K_1 = (\bar{x} \cdot \bar{z} \cdot \text{Start} + k_1) \cdot \bar{x}$$

Svolgendo l'operazione logica si ottiene:

$$K_1 = \bar{x} \cdot \bar{z} \cdot Start + k_1 \cdot \bar{x} \quad \text{raccogliendo la } \bar{x} \text{ si ottiene:}$$

$$K_1 = (\bar{z} \cdot Start + k_1) \cdot \bar{x}$$

Comando elettrovalvola  $Y_1$  :

$$Y_1 = k_1$$

2- Comando elettrovalvola  $Y_3$

$$\text{Memoria } K_3 \left\{ \begin{array}{l} \text{Set} \quad B^+ = a_0 \cdot x \cdot \bar{z} \\ \text{Reset} \quad B^- = z \end{array} \right.$$

Equazione della memoria  $K_3$

$$K_3 = (a_0 \cdot x \cdot \bar{z} + k_3) \cdot \bar{z}$$

Svolgendo le operazioni logiche si ottiene:

$$K_3 = a_0 \cdot x \cdot \bar{z} + k_3 \cdot \bar{z} \quad \text{raccogliendo } \bar{z} \text{ si ottiene:}$$

$$K_3 = (a_0 \cdot x + k_3) \cdot \bar{z}$$

Comando elettrovalvola  $Y_3$

$$Y_3 = k_3$$

3- Comando elettrovalvola  $Y_5$

$$\text{Memoria } K_5 \left\{ \begin{array}{l} \text{Set} \quad C^+ = b_0 \cdot x \cdot z \\ \text{Reset} \quad C^- = \bar{x} \end{array} \right.$$

Equazione della memoria  $K_5$ :

$$K_5 = (b_0 \cdot x \cdot z + k_5) \cdot \bar{x} \quad \text{da cui } K_5 = (b_0 \cdot x \cdot z + k_5) \cdot x$$

Svolgendo le operazioni logiche si ha:

$$K_5 = b_0 \cdot x \cdot z + k_5 \cdot x \quad \text{raccogliendo } x \text{ si ottiene:}$$

$$K_5 = (b_0 \cdot z + k_5) \cdot x$$

Comando elettrovalvola  $Y_5$ :

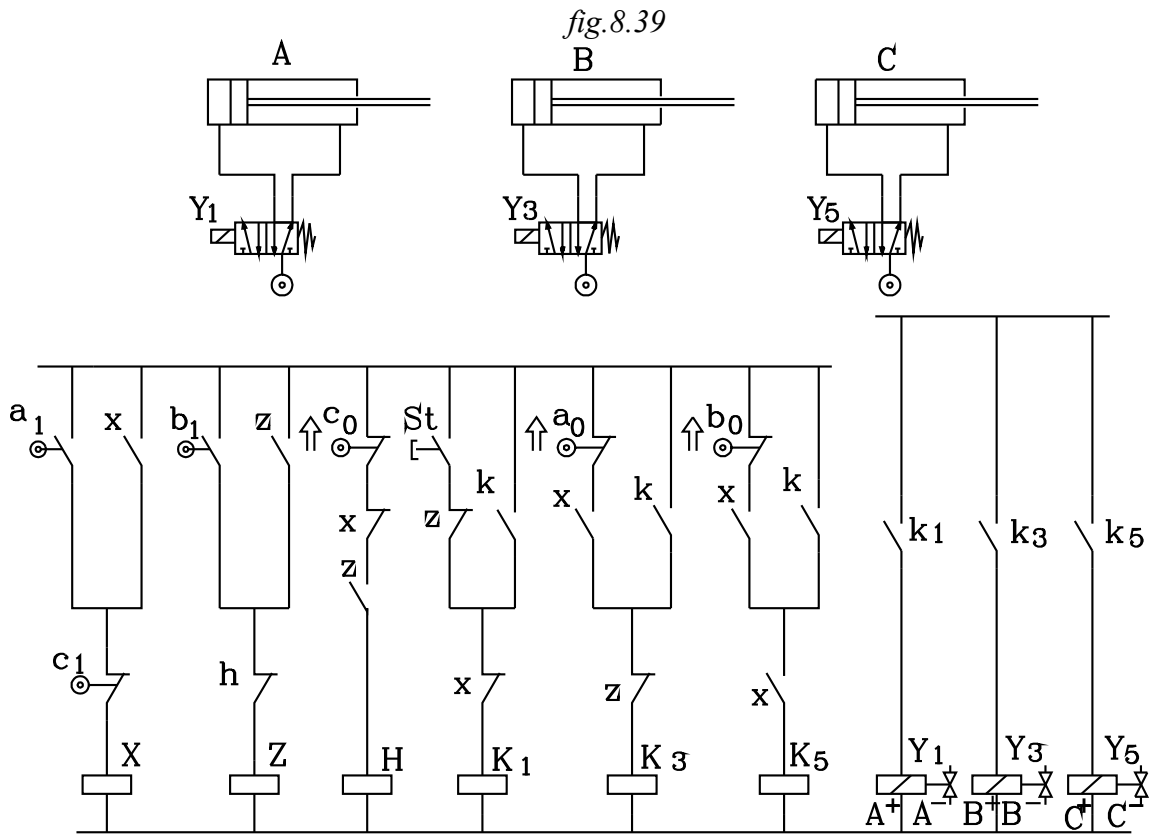
$$Y_5 = k_5$$

Riassumendo le equazioni logiche:

$$X = (a_1 + x) \cdot \bar{c}_1 \quad Z = (b_1 + z) \cdot \bar{h} \quad H = c_0 \cdot \bar{x} \cdot z \quad K_1 = (\bar{z} \cdot Start + k_1) \cdot \bar{x}$$

$$Y_1 = k_1 \quad K_3 = (a_0 \cdot x + k_3) \cdot \bar{z} \quad Y_3 = k_3 \quad K_5 = (b_0 \cdot z + k_5) \cdot x \quad Y_5 = k_5$$

Dalle equazioni logiche si ottiene il circuito di *fig. 8.*:





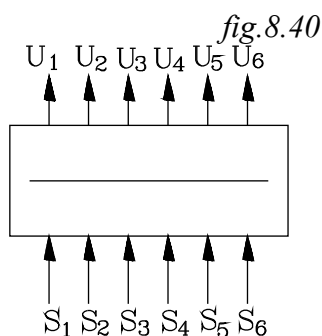
## 8.6 METODO DELLE MEMORIE IN CASCATA

Nei circuiti pneumatici sono stati studiati i sistemi di memoria denominati in cascata, nei quali è possibile avere in uscita un segnale alla volta, attivato da un segnale di *Set*.

Si vuole costruire per i circuiti elettro pneumatici un sistema di memorie che rispondano agli stessi requisiti di quelle in cascata applicate nei circuiti pneumatici.

Detto sistema di memorie deve soddisfare alle seguenti condizioni:

- 1- Il numero dei segnali di ingresso debbono essere uguali a quelli di uscita
- 2- Ad ogni segnale di ingresso vi deve corrispondere uno e un solo segnale di uscita
- 3- Il segnale di uscita deve essere memorizzato e permanere anche quando si annulla il segnale di ingresso e ciò fino a quando non si verifica il segnale di *Reset* di quella uscita
- 4- Quando viene settata una uscita tutte le altre sono resettate
- 5- I segnali di settaggio debbono succedersi in sequenza, secondo un ordine stabilito.
- 6- Nelle condizioni di riposo deve essere settata l'ultima memoria
- 7- Il settaggio di una memoria determina il resettaggio di tutte le precedenti; per cui, nell'ordine ciclico la prima memoria resetta l'ultima che nelle condizioni di riposo deve risultare settata.



Per chiarezza di esposizione si consideri un sistema di memorie che debba fornire le  $n=6$  uscite:

$$U_1, U_2, U_3, U_4, U_5, U_6$$

Siano rispettivamente:

$$S_1, S_2, S_3, S_4, S_5, S_6$$

i segnali di settaggio delle 6 uscite.

La prima memoria può dare in uscita i due segnali opposti  $0,1$  ognuna delle altre memorie è adibita a fornire in uscita un solo segnale.

Da ciò ne viene che se  $n$  è il numero dei segnali di uscita (*e di ingresso*), il numero di memorie  $m$  occorrenti per il sistema risulta:

$$m = n - 1$$

Nel caso in esame, dove il numero di uscite, preso come esempio esplicativo è  $n=6$ , per il sistema occorre utilizzare  $m=6-1=5$  memorie:  $m_1, m_2, m_3, m_4, m_5$ .

Come si è esposto nei precedenti 7 punti, ogni memoria è attivata da un proprio segnale di settaggio  $S_i$ . Essa fornisce il segnale di uscita di utilizzazione  $U_i$ , il quale, contemporaneamente, determina il resettaggio di tutte le memorie precedenti, che si sono succedute nell'ordine ciclico. Così:

$m_1$  È resettata dalle 4 uscite delle successive memorie  
 $U_2, U_3, U_4, U_5$

$m_2$  È resettata dalle 3 uscite delle successive memorie:  
 $U_3, U_4, U_5$

$m_3$  È resettata dalle 2 uscite delle successive memorie:  
 $U_4, U_5$

$m_4$  È resettata dalla uscita dell'ultima memoria:  
 $U_5$

*L'ultima memoria  $m_5$  è resettata dall'ultimo segnale di settaggio  $S_6$*

Ogni memoria settata fornisce una uscita. L'ultima uscita si ha quando sono resettate tutte le memorie

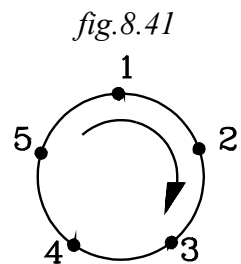
In pratica conviene procedere nella seguente maniera:

1°

Si pongono le memorie, che si succedono in ordine ciclico di settaggio, su di una circonferenza e ordinate in senso orario.

Nell'esempio esplicativo vi sono  $m=6-1=5$  memorie

Siano in ordine  $S_1, S_2, S_3, S_4, S_5, S_6$  i segnali di settaggio e  $U_1, U_2, U_3, U_4, U_5, U_6$  quelli di uscita



2°

Ogni memoria  $m_i$ , attivata dal segnale di settaggio  $S_i$ , fornisce un'uscita  $U_i$  e questo viene resettato dai segnali di uscita delle memorie successive.

Così la prima memoria  $m_1$ , attivata dal segnale di settaggio  $S_1$ , fornisce l'uscita  $U_1$ , questa viene resettata dai segnali di uscita delle 4 successive:  $U_2, U_3, U_4, U_5$  (seguire l'ordine ciclo sulla circonferenza nel senso orario)

La bobina della memoria viene contraddistinta dalla stessa indicazione del suo segnale di uscita.

Le condizioni imposte sull'uscita  $U_1$  determinano l'equazione logica:

$$U_1 = (S_1 + U_1) \cdot \overline{U_2} \cdot \overline{U_3} \cdot \overline{U_4} \cdot \overline{U_5}$$

Così si procede per le successive uscite delle memorie, salvo l'ultima memoria che viene resettata dall'ultimo segnale di settaggio  $S_6$  (rimane poi da determinare l'ultima uscita)

memoria	Settaggio	Uscita	Resettaggio memoria	Equazione logica uscita della memoria
$m_1$	$S_1$	$U_1$	$U_2, U_3, U_4, U_5$	$U_1 = (S_1 + U_1) \cdot \overline{U_2} \cdot \overline{U_3} \cdot \overline{U_4} \cdot \overline{U_5}$
$m_2$	$S_2$	$U_2$	$U_3, U_4, U_5$	$U_2 = (S_2 + U_2) \cdot \overline{U_3} \cdot \overline{U_4} \cdot \overline{U_5}$
$m_3$	$S_3$	$U_3$	$U_4, U_5$	$U_3 = (S_3 + U_3) \cdot \overline{U_4} \cdot \overline{U_5}$
$m_4$	$S_4$	$U_4$	$U_5$	$U_4 = (S_4 + U_4) \cdot \overline{U_5}$
$m_5$	$S_5$	$U_5$	$S_6$	$U_5 = (S_5 + U_5) \cdot \overline{S_6}$

L'ultima uscita dal sistema di memorie si ottiene quando risultano resettate tutte le uscite delle precedenti memorie:

$$U_6 = \overline{U_1} \cdot \overline{U_2} \cdot \overline{U_3} \cdot \overline{U_4} \cdot \overline{U_5}$$

Ricapitolando le equazioni logiche delle 6 memorie in cascata:

$$U_1 = (S_1 + U_1) \cdot \overline{U_2} \cdot \overline{U_3} \cdot \overline{U_4} \cdot \overline{U_5}$$

$$U_2 = (S_2 + U_2) \cdot \overline{U_3} \cdot \overline{U_4} \cdot \overline{U_5}$$

$$U_3 = (S_3 + U_3) \cdot \overline{U_4} \cdot \overline{U_5}$$

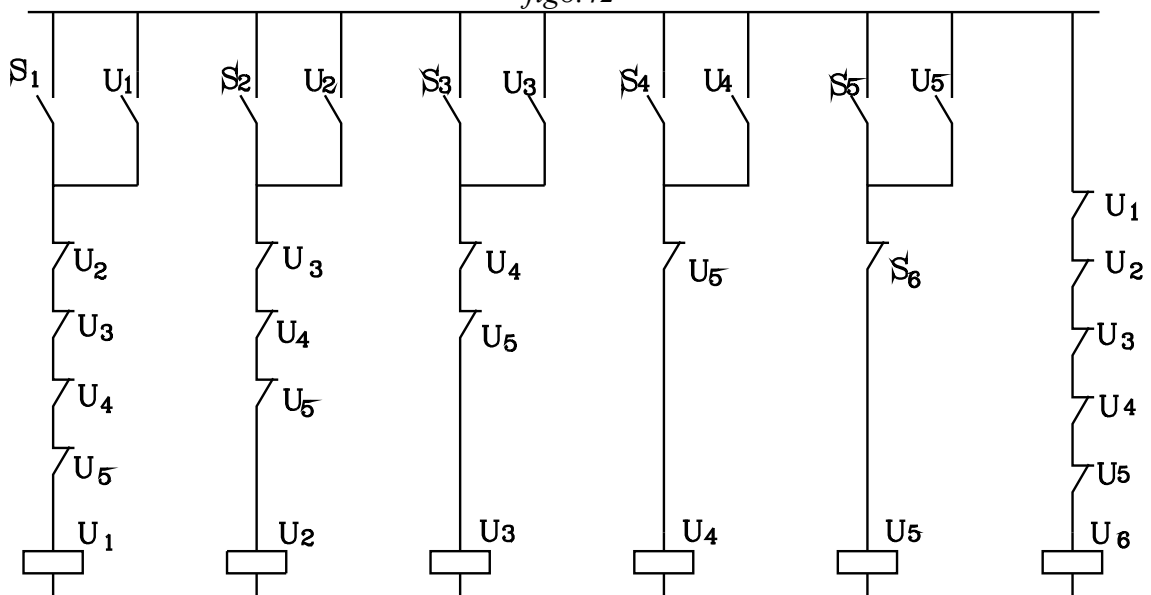
$$U_4 = (S_4 + U_4) \cdot \overline{U_5}$$

$$U_5 = (S_5 + U_5) \cdot \overline{S_6}$$

$$U_6 = \overline{U_1} \cdot \overline{U_2} \cdot \overline{U_3} \cdot \overline{U_4} \cdot \overline{U_5}$$

Dalle equazioni logiche si ottiene lo schema delle 6 memorie in cascata

fig8.42



### 2.6.1 Una memoria con due uscite

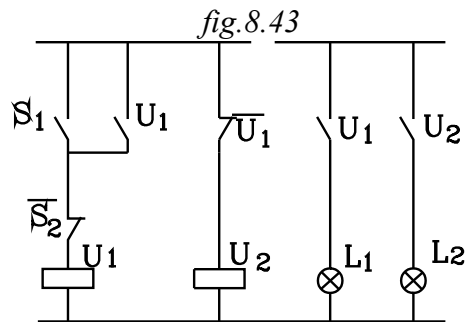
Una memoria ha due stadi opposti: di SET e RESET; per cui possono essere elaborati due segnali di uscita  $U_1$   $U_2$  diversi, comandati, rispettivamente, dai due segnali di ingresso  $S_1$ ,  $S_2$ .

Considerando le condizioni su esposte per le memorie occorre che:

- $S_1$  setti l'uscita  $U_1$
  - $S_2$  resetti  $U_1$
  - Nel resettaggio di  $U_1$  venga settata l'uscita  $U_2$
- Le equazioni logiche sono:

$$U_1 = (S_1 + U_1) \cdot \overline{S_2}$$

$$U_2 = \overline{U_1}$$



Qui accanto è riportato il circuito che realizza le equazioni logiche. Le uscite  $U_1$  e  $U_2$  sono, inviate rispettivamente alle due lampade  $L_1$ ,  $L_2$

Come si può osservare:

-Nelle condizioni iniziali è eccitata la bobina  $U_2$ , ed è quindi attivata l'uscita  $U_2$  ( e ciò, corrisponde alle condizioni che si debbono avere all'inizio di un ciclo). La lampada  $L_2$  è accesa.

Pigiando  $S_1$ , si eccita la bobina  $U_1$ : si chiude il suo contatto di autoritenuta  $U_1$  mentre si apre il contatto normalmente chiuso  $\overline{U_1}$  in serie con la bobine  $U_2$  diseccitandola. In tal modo, si spegne la lampada  $L_2$  perché si apre il contatto  $U_2$  mentre si accende  $L_1$  per effetto della chiusura di  $U_1$

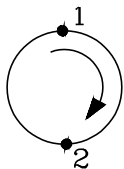
Pigiando  $S_2$  si diseccita la bobina  $U_1$ , resettando la memoria. Si apre il contatto normalmente aperto  $U_1$  e si chiude quello normalmente chiuso  $\overline{U_1}$  eccitando la bobina  $U_2$ . In tal modo si spegna la lampada  $L_1$  e si accende  $L_2$ .

Quindi in sequenza:

- Nelle condizioni iniziali è attivata l'uscita  $U_2$ .
- Attivando  $S_1$  si ha il resettaggio dell'uscita  $U_2$  e il settaggio di  $U_1$
- Attivando  $S_2$  si ha il resettaggio dell'uscita  $U_1$  e il settaggio di  $U_2$

### 2.6.2 Due memorie tre uscite

*fig.8.44*



Applicando il procedimento spiegato nel caso di 6 memorie e applicabile a qualsiasi numero di esse si ha:

memoria	Settaggio	Uscita	Resettaggio memoria	Equazione logica uscita della memoria
$m_1$	$S_1$	$U_1$	$U_2$	$U_1 = (S_1 + U_1) \cdot \overline{U_2}$
$m_2$	$S_2$	$U_2$	$S_3$	$U_2 = (S_2 + U_2) \cdot \overline{S_3}$

L'ultima uscita  $U_3$  dal sistema di memorie si ottiene quando risultano resettate tutte le uscite delle precedenti memorie:

$$U_3 = \overline{U_1} \cdot \overline{U_2}$$

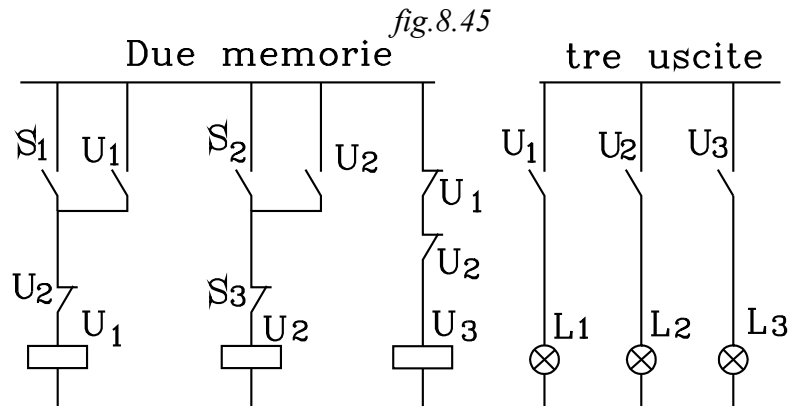
Riassunto equazioni logiche:

$$U_1 = (S_1 + U_1) \cdot \overline{U_2}$$

$$U_2 = (S_2 + U_2) \cdot \overline{S_3}$$

$$U_3 = \overline{U_1} \cdot \overline{U_2}$$

Dalle equazioni logiche si ricava il circuito del sistema di memorie

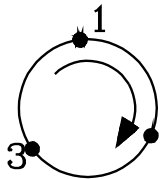


In sequenza si ha:

- Nelle condizioni iniziali è attivata l'uscita  $U_3$ .
- Attivando  $S_1$  si ha il resettaggio dell'uscita  $U_3$  e il settaggio di  $U_1$
- Attivando  $S_2$  si ha il resettaggio dell'uscita  $U_1$  e il settaggio di  $U_2$  ( $U_3$  rimane disattivata)
- Attivando  $S_3$  si ha il resettaggio dell'uscita  $U_1$   $U_2$  e il settaggio di  $U_3$

### 8.6.3 Tre memorie quattro uscite

*fig.8.46*



Applicando il procedimento effettuato nei casi precedenti si ha:

memoria	Settaggio	Uscita	Resettaggio memoria	Equazione logica uscita della memoria
$m_1$	$S_1$	$U_1$	$U_2, U_3$	$U_1 = (S_1 + U_1) \cdot \overline{U_2} \cdot \overline{U_3}$
$m_2$	$S_2$	$U_2$	$U_3$	$U_2 = (S_2 + U_2) \cdot \overline{U_3}$
$m_3$	$S_3$	$U_3$	$S_4$	$U_3 = (S_3 + U_3) \cdot S_4$

L'ultima uscita  $U_4$  dal sistema di memorie si ottiene quando risultano resettate tutte le uscite delle precedenti memorie:

$$U_4 = \overline{U_1} \cdot \overline{U_2} \cdot \overline{U_3}$$

Riassumendo le equazioni logiche

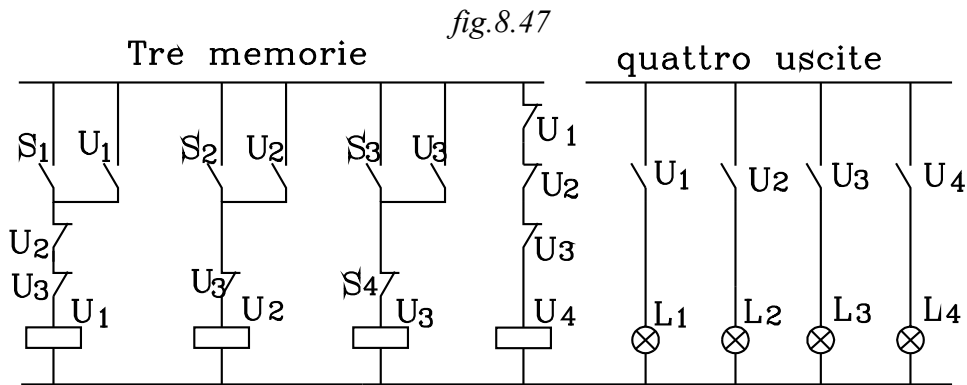
$$U_1 = (S_1 + U_1) \cdot \overline{U_2} \cdot \overline{U_3}$$

$$U_2 = (S_2 + U_2) \cdot \overline{U_3}$$

$$U_3 = (S_3 + U_3) \cdot S_4$$

$$U_4 = \overline{U_1} \cdot \overline{U_2} \cdot \overline{U_3}$$

Dalle equazioni logiche si ricava il circuito del sistema di memorie:



### 8.6.4 UTILIZZO DELLE MEMORIE IN CASCATA NEI CICLI

Per spiegare il sistema di utilizzazione delle memorie in cascata proposte per i circuiti elettropneumatici ci si riferisce al ciclo  $A^+ B^+ B^- C^+ C^- A^-$  già studiato e realizzato con circuito pneumatico.

#### IN LABORATORIO

##### 8.6.4.1 Cicli con valvole bistabili

Si dà qui per acquisito il metodo per estrarre le equazioni logiche da utilizzare nelle memorie in cascata. Si riporta qui brevemente il procedimento dell'assegnazione delle memorie ai gruppi già noto.

$CICLO A^+ B^+ B^- C^+ C^- A^-$

Diagramma corsa passo

fig.8.48

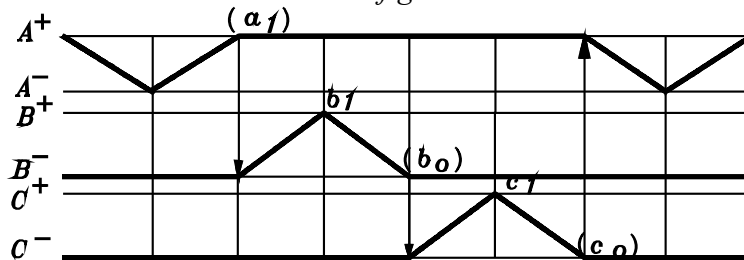


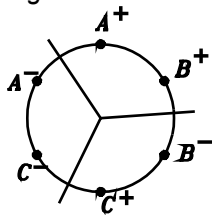
Tabella dei comandi

Comando	Stato comandato	Stato iniziale dei finecorsa	Stato raggiunto dai finecorsa
---------	-----------------	---------------------------------	----------------------------------

$A_1$	$A^+$	$a_0 \cdot b_0 \cdot c_0 \rightarrow$	$a_1 \cdot b_0 \cdot c_0$
$B_1$	$B^+$	$a_1 \cdot b_0 \cdot c_0 \rightarrow$	$a_1 \cdot b_1 \cdot c_0$
$B_0$	$B^-$	$a_1 \cdot b_1 \cdot c_0 \rightarrow$	$a_1 \cdot b_0 \cdot c_0$
$C_1$	$C^+$	$a_1 \cdot b_0 \cdot c_0 \rightarrow$	$a_1 \cdot b_0 \cdot c_1$
$C_0$	$C^-$	$a_1 \cdot b_0 \cdot c_1 \rightarrow$	$a_1 \cdot b_0 \cdot c_0$
$A_0$	$A^-$	$a_1 \cdot b_0 \cdot c_0 \rightarrow$	$a_0 \cdot b_0 \cdot c_0$

*Suddivisione in gruppi non contenenti doppie corse*

fig.8.49



Partendo da  $A^+$  si ottengono i tre gruppi:

$$A^+ B^+ / B^- C^+ / C^- A^-$$

che non comprendono doppie corse e, quindi, non contenenti segnali bloccanti.

Ad ogni gruppo si associa un segnale di uscita dalle memorie in cascata.

Occorrono quindi 3 uscite  $U_1, U_2, U_3$  e  $m=3-1=2$  memorie

*Matrice dei comandi*

I gruppo		II gruppo		III gruppo	
$A^+$	$B^+$	$B^-$	$C^+$	$C^-$	$A^-$
$a_0$	$a_1$	$b_1$	$b_0$	$c_1$	$c_0$
$S_1$	$U_1$	$S_2$	$U_2$	$S_3$	$U_3$

Eseguito il procedimento già spiegato per le memorie in cascata si estraggono le seguenti equazioni logiche:

$$I \text{ gruppo } \begin{cases} S_1 = a_0 \cdot Start \\ A^+ = U_1 \\ B^+ = a_1 \cdot U_1 \end{cases} \quad (8.6.4.1.1)$$

$$II \text{ gruppo } \begin{cases} S_2 = b_1 \\ B^- = U_2 \\ C^+ = b_0 \cdot U_2 \end{cases} \quad (8.6.4.1.2)$$

$$III \text{ gruppo } \begin{cases} S_3 = c_1 \\ C^- = U_3 \\ A^- = c_0 \cdot U_3 \end{cases} \quad (8.6.4.1.3)$$

*Esecuzione del circuito elettropneumatico con elettrovalvole bistabili*

Conviene iniziare con la stesura del circuito riguardante le due memorie in cascata.

Le equazioni logiche delle due memorie in cascata con 3 uscite sono:

$$\begin{aligned} U_1 &= (S_1 + U_1) \cdot \overline{U_2} \\ U_2 &= (S_2 + U_2) \cdot \overline{S_3} \quad (8.6.4.1.4) \\ U_3 &= \overline{U_1} \cdot \overline{U_2} \end{aligned}$$

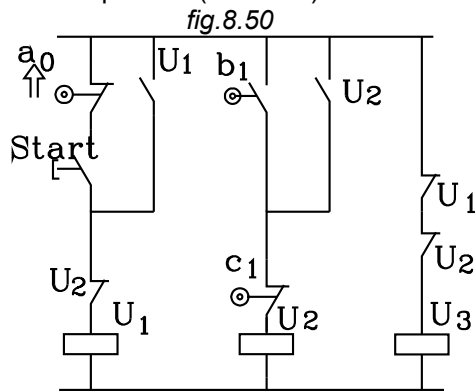
Dove  $S_1, S_2, S_3$  si ricavano dalle equazioni logiche (8.6.4.1.1) - (8.6.4.1.2) - (8.6.4.1.3).

$$\begin{aligned} S_1 &= a_0 \cdot Start \\ S_2 &= b_1 \\ S_3 &= c_1 \end{aligned}$$

Per cui le equazioni logiche delle uscite dalle due memorie in cascata risultano:

$$\begin{aligned} U_1 &= (a_0 \cdot Start + U_1) \cdot \overline{U_2} \\ U_2 &= (b_1 + U_2) \cdot \overline{c_1} \quad (8.6.4.1.5) \\ U_3 &= \overline{U_1} \cdot \overline{U_2} \end{aligned}$$

Le equazioni (8.6.4.1.5) si traducono nel seguente circuito:



Questo circuito fornisce i segnali di memoria da porre in AND ai segnali di comando, come è stato effettuato nei tre gruppi.

Occorre poi aggiungere il circuito di comando delle elettrovalvole corrispondente alle equazioni logiche ricavate dalla matrice dei comandi.

Si indicano come al solito le elettrovalvole corrispondenti ai comandi:

$A^+$  → Elettrovalvola  $Y_1$

$A^-$  → Elettrovalvola  $Y_2$

$B^+$  → Elettrovalvola  $Y_3$

$B^-$  → Elettrovalvola  $Y_4$

$C^+$  → Elettrovalvola  $Y_5$

$C^-$  → Elettrovalvola  $Y_6$

$A^+ = U_1$  →  $Y_1 = U_1$

$B^+ = a_1 \cdot U_1$  →  $Y_3 = a_1 \cdot U_1$

$B^- = U_2$  →  $Y_4 = U_2$

$C^+ = b_0 \cdot U_2$  →  $Y_5 = b_0 \cdot U_2$

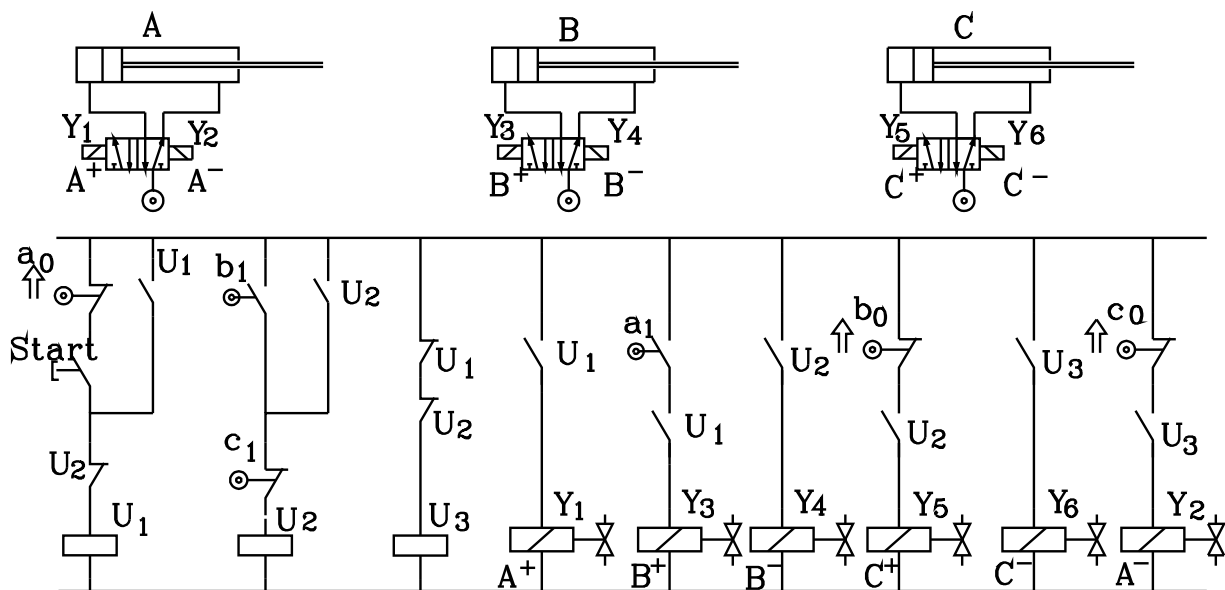
$C^- = U_3$  →  $Y_6 = U_3$

$A^- = c_0 \cdot U_3$  →  $Y_2 = c_0 \cdot U_3$

Dalle equazioni logiche si ottiene il circuito di *fig.8.51*:



fig.8.51



#### Breve descrizione del funzionamento

- Nelle condizioni iniziali sono diseccitate le bobine  $U_1, U_2$  ed è eccitata la bobina  $U_3$ .
- Pigiato lo Start si eccita la bobina  $U_1$  della prima memoria e si chiudono i relativi contatti N.A  $U_1$  e si aprono quelli N.C  $\overline{U_1}$ . La bobina  $U_3$  viene diseccitata, mentre quella della memoria  $U_1$  rimane eccitata e viene attivata la bobina  $Y_1$  dell'elettrovalvola, che, commutandosi, effettua la corsa  $A^+$  di fuoriuscita dello stelo del cilindro A.
- Toccato il finecorsa  $a_1$  si eccita l'elettrovalvola  $Y_3$  (in serie con  $a_1$  e  $\overline{U_3}$ ). Viene effettuata la corsa  $B^+$  di fuoriuscita dello stelo del cilindro B.
- Toccato il finecorsa  $b_1$  si eccita la bobina della seconda memoria  $U_2$ . Si chiudono i relativi contatti N.A  $U_2$  e si aprono quelli N.C  $\overline{U_2}$ . In tal modo si diseccita la prima memoria  $U_1$  e resta diseccitata la bobina  $U_3$ ; La chiusura dei contatti  $U_2$  determina l'eccitazione della bobina dell'elettrovalvola  $Y_4$  che, commutandosi, determina la corsa  $B^-$  di rientro dello stelo del cilindro B.
- I contatti  $\overline{U_1}$  si sono aperti e quindi sono diseccitate le bobine  $Y_1, Y_3$  dell'elettrovalvole. La chiusura di  $U_2$  non determina in questo momento l'eccitazione della bobina  $Y_5$  in quanto essendo fuoriuscito lo stelo di B (esso sta rientrando), non viene toccato il finecorsa  $b_0$ .
- Una volta che è rientrato lo stelo del cilindro B viene toccato  $b_0$  ( $U_2$  è chiuso) e si eccita la bobina  $Y_5$  dell'elettrovalvola. Questa si commuta e effettua la corsa  $C^+$  di fuoriuscita dello stelo del cilindro C.
- Toccato il finecorsa N.C  $c_1$  questo, aprendosi, interrompe la continuità elettrica verso la bobina  $U_2$  che si diseccita. Si aprono i contatti N.A  $U_2$  e si chiudono i N.C  $\overline{U_2}$ .
- Ora la memoria  $U_1$  rimane diseccitata in quanto lo stelo del cilindro A è ancora fuoriuscito e non viene toccato il finecorsa  $a_0$ ; contemporaneamente, come si è detto  $U_2$  è diseccitata, ne viene che tutti e due i contatti N.C  $\overline{U_1}, \overline{U_2}$  risultano chiusi. Viene così eccitata la bobina  $U_3$  che chiude i suoi contatti N.A  $U_3$ .
- La chiusura dei contatti  $U_3$  determina l'eccitazione della bobina  $Y_6$  dell'elettrovalvola, che commutandosi, comanda la corsa  $C^-$  di rientro dello stelo del cilindro C. In questo momento la chiusura di  $U_3$  non può determinare l'eccitazione della bobina  $Y_2$ , in quanto il finecorsa  $c_0$  non risulta toccato (lo stelo fuoriuscito sta rientrando).
- Quando viene toccato il finecorsa  $c_0$  allora, risultando chiuso  $U_3$ , si eccita la bobina  $Y_2$  dell'elettrovalvola la quale commutandosi determina la corsa  $A^-$  di rientro dello stelo del cilindro A.
- Toccato  $a_0$  si ritorna nelle condizioni iniziali.
- Se lo Start è pigiato inizia un nuovo ciclo.

### 8.6.4.2 Esecuzione del circuito elettropneumatico con elettrovalvole monostabili

Il circuito delle memorie in cascata sia lo stesso che è stato utilizzato usando le elettrovalvole bistabili.

---

#### IN LABORATORIO

---

CICLO  $A^+ B^+ B^- C^+ C^- A^-$

Come si è precedentemente spiegato per poter utilizzare le valvole monostabili occorre impiegare per ciascuna di esse una memoria, la quale comanda la corsa di lavoro con il segnale di settaggio e quella di ritorno con quello di resettaggio.

Si analizzino i segnali di Set e di Reset da inviare alle valvole monostabili per effettuare le corse di fuoriuscita e rientro degli steli dei cilindri

Memoria $K_1$	→	Comanda l'elettrovalvola $Y_1$	$Y_1=K_1$
Memoria $K_3$	→	Comanda l'elettrovalvola $Y_3$	$Y_3=K_3$
Memoria $K_5$	→	Comanda l'elettrovalvola $Y_5$	$Y_5=K_5$

Come al solito le tre elettrovalvole comandano rispettivamente:

$Y_1$	Comanda il cilindro A	{	<i>Segnale di Set</i> → $A^+$	→	$A^+$	}	<i>Segnale di Reset</i> → $A^-$
$Y_3$	Comanda il cilindro B	{	<i>Segnale di Set</i> → $B^+$	→	$B^+$	}	<i>Segnale di Reset</i> → $B^-$
$Y_5$	Comanda il cilindro C	{	<i>Segnale di Set</i> → $C^+$	→	$C^+$	}	<i>Segnale di Reset</i> → $C^-$

Per comodità riportiamo le equazioni logiche già determinate

$$\begin{aligned}
 A^+ &= U_1 && \rightarrow Y_1 = U_1 \\
 B^+ &= a_1 \cdot U_1 && \rightarrow Y_3 = a_1 \cdot U_1 \\
 B^- &= U_2 && \rightarrow Y_4 = U_2 \\
 C^+ &= b_0 \cdot U_2 && \rightarrow Y_5 = b_0 \cdot U_2 \\
 C^- &= U_3 && \rightarrow Y_6 = U_3 \\
 A^- &= c_0 \cdot U_3 && \rightarrow Y_2 = c_0 \cdot U_3
 \end{aligned}$$

Equazioni logiche delle memorie associate all'elettrovalvole:

1- Comando elettrovalvola  $Y_1$

$$\text{Memoria } K_1 \left\{ \begin{array}{l} \text{Set} \quad A^+ = U_1 \\ \text{Re set} \quad A^- = c_0 \cdot U_3 \end{array} \right.$$

Equazione memoria  $K_1$

$$K_1 = (U_1 + k_1) \cdot c_0 \cdot U_3$$

Si pone:

$$H_1 = c_0 \cdot U_3$$

L'equazione logica della memoria diviene:

$$K_1 = (U_1 + k_1) \cdot \bar{h}_1$$

Comando elettrovalvola  $Y_1$

$$Y_1 = k_1$$

2- Comando elettrovalvola  $Y_3$

$$\text{Memoria } K_3 \begin{cases} \text{Set} & B^+ = a_1 \cdot U_1 \\ \text{Reset} & B^- = U_2 \end{cases}$$

Equazione della memoria  $K_3$

$$K_3 = (a_1 \cdot U_1 + k_3) \cdot \bar{U}_2$$

Comando elettrovalvola  $Y_3$

$$Y_3 = k_3$$

2- Comando elettrovalvola  $Y_5$

$$\text{Memoria } K_5 \begin{cases} \text{Set} & C^+ = b_0 \cdot U_2 \\ \text{Reset} & C^- = U_3 \end{cases}$$

Equazione della memoria  $K_5$

$$K_5 = (b_0 \cdot U_2 + k_5) \cdot \bar{U}_3$$

Comando elettrovalvola  $Y_5$

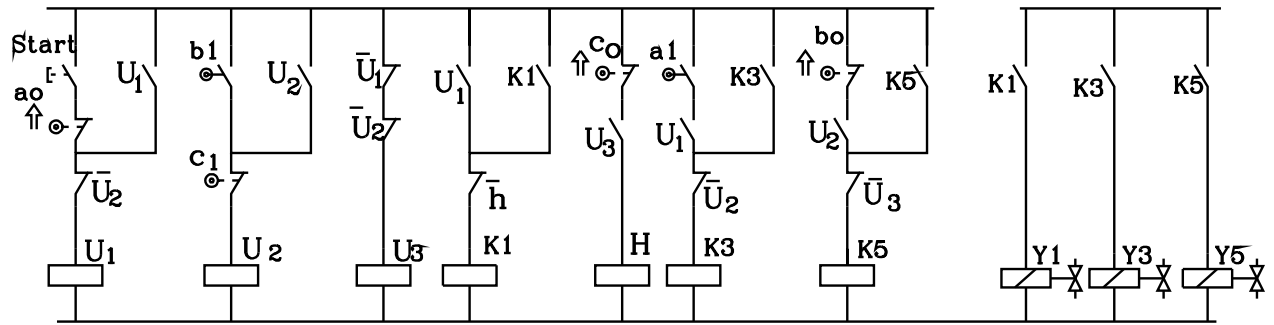
$$Y_5 = k_5$$

Riassumendo le equazioni logiche:

$$\text{Memorie in cascata} \begin{cases} U_1 = (a_0 \cdot \text{Start} + U_1) \cdot \bar{U}_2 \\ U_2 = (b_1 + U_2) \cdot \bar{c}_1 \\ U_3 = \bar{U}_1 \cdot \bar{U}_2 \end{cases} \quad K_1 = (U_1 + k_1) \cdot \bar{h}_1 \quad \boxed{Y_1 = k_1}$$

$$K_3 = (a_1 \cdot U_1 + k_3) \cdot \bar{U}_2 \quad \boxed{Y_3 = k_3} \quad K_5 = (b_0 \cdot U_2 + k_5) \cdot \bar{U}_3 \quad \boxed{Y_5 = k_5}$$

fig.8.52



## 8.7 GRAFCET NEI CIRCUITI ELETTROPNEUMATICI A LOGICA CABLATA

Si vogliono qui adattare le equazioni logiche dei sequenziatori pneumatici alla soluzione di circuiti elettropneumatici cablati.

Si rammenta le condizioni che debbono essere rispettate.

- 1- Ad ogni passo  $i^{mo}$  viene assegnata una memoria  $K_i$
- 2- La memoria  $K_i$  comanda al bobina dell'elettrovalvola  $Y_i$  che, con la sua commutazione effettua il passo; contemporaneamente essa resetta la memoria precedente  $K_{i-1}$
- 3- La memoria  $K_i$  viene attivata solamente quando si realizza la condizione di transizione precedente: *quando, cioè, è stata attivata la memoria della fase precedente  $k_{i-1}$  e torna il segnale  $x_{i-1}$  indicante l'attuazione del comando (è toccato il fine corsa o è attivato un sensore di prossimità)*
- 4- Il segnale di transizione  $H_{i-1}$  della fase  $(i-1)^{ma}$  che comanda la fase successiva  $i^{ma}$  è dato dall'AND dei due segnali  $k_{i-1}$  e  $x_{i-1}$
- 5- Nel passo  $i^{mo}$ , attivata la memoria  $K_i$ , comandata la commutazione dell'elettrovalvola  $Y_i$ , non si verifica il passo successivo  $i+1$  fino a che non termina lo stato di transizione dell'attuale passo. Ciò avviene quando viene attuato dal finecorsa il segnale di avvenuto comando  $x_j$ .
- 6- Dalla transizione della fase  $i^{ma}$  viene elaborato il segnale di comando  $H_i$  della fase successiva  $(i+1)^{ma}$ , ponendo in AND il segnale di memoria  $K_i$  con quello  $x_j$  indicante l'attuazione del comando.

fig.8.53

Si considerino così due fasi successive  $i$ ,  $i+1$ . Seguendo i 6 punti precedenti si determinano le equazioni logiche che regolano le due fasi.

Alla fase  $i$  si associa la memoria  $K_i$  che comanderà la bobina  $Y_i$  dell'elettrovalvola.

Alla fase  $i+1$  si associa la memoria  $K_{i+1}$  che comanda la bobina  $Y_{i+1}$  dell'elettrovalvola.

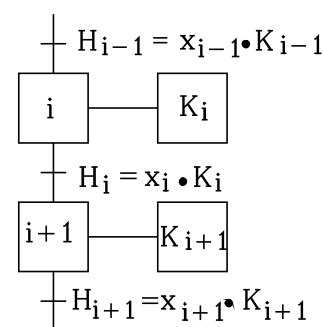
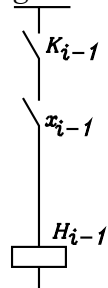


fig.8.54

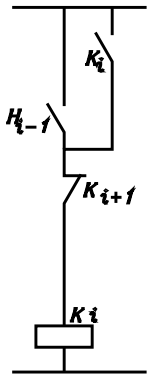


Lo stato di transizione che precede la fase  $i^{ma}$  è dato dall'attivazione della memoria precedente  $K_{i-1}$  e dal segnale  $x_{i-1}$  di attuazione del comando. Questi segnali vengono posti in AND e determinano il segnale di comando  $H_{i-1}$  della fase  $i^{ma}$ .

$$H_{i-1} = k_{i-1} \cdot x_{i-1} \quad (8.7.1)$$

La memoria  $K_i$  è attivata dal segnale di transizione  $H_{i-1}$  della fase precedente e viene resettata dal segnale di memoria successiva  $K_{i+1}$ .

fig.8.55



$$\text{Memoria } K_i \begin{cases} \text{Set } H_{i-1} \\ \text{Reset } K_{i+1} \end{cases}$$

Equazione logica della memoria  $K_i$

$$K_i = (H_{i-1} + k_i) \cdot \overline{k_{i+1}} \quad (8.7.2)$$

La memoria  $K_i$  comanda la bobina  $Y_i$  dell'elettrovalvola

$$Y_i = k_i \quad (8.7.3)$$

Attivata la memoria  $K_i$  e comandata la bobina dell'elettrovalvola  $Y_i$ , segue uno stato di transizione della fase. Non si ottiene il passo successivo  $i+1$  fino a che non viene verificata l'attuazione del comando impartito dalla memoria  $K_i$ . La verifica viene effettuata attraverso l'attivazione di un finecorsa (o di un sensore di prossimità) che elabora il segnale  $x_i$  indicante l'esecuzione del comando.

Il comando del passo successivo  $i+1$  si ottiene dal segnale di transizione  $H_i$  ottenuto dall'AND del segnale di memoria  $K_i$  con quello  $x_i$  indicante l'avvenuto comando.

$$H_i = k_i \cdot x_i \quad (8.7.4)$$

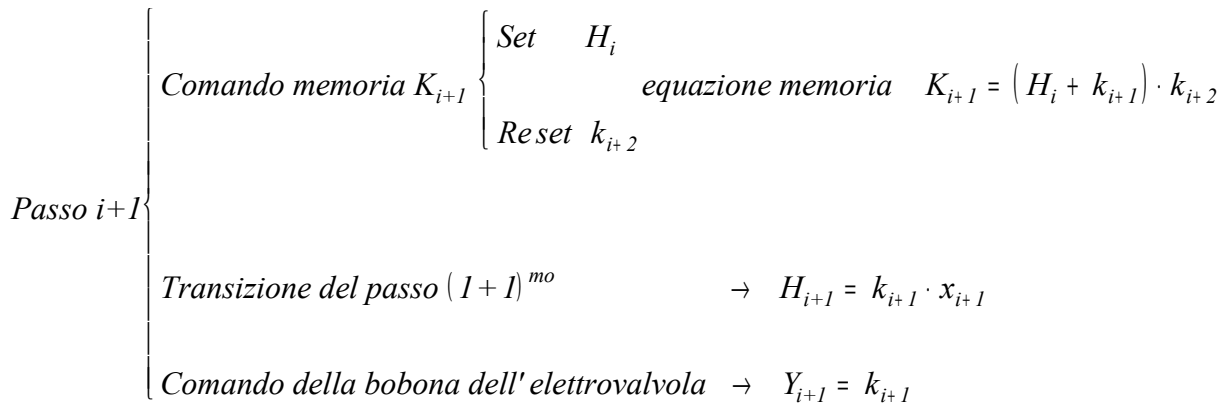
Le equazioni, logiche (8.7.1) (8.7.2) (8.7.3) si ripetono modularmente nei passi successivi . Diverso risulta invece l'equazione dell'ultimo passo.

Così considerando l'attuazione dei due passi successivi  $i$  e  $i+1$  si ha:

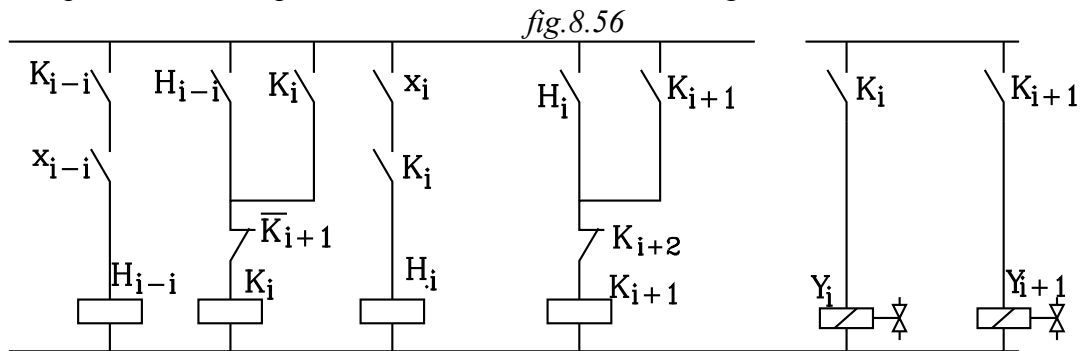
Transizione passo  $i-1$

Segnale di comando passo  $i^{mo}$        $H_{i-1} = k_{i-1} \cdot x_{i-1}$

$$\text{Passo } i \left\{ \begin{array}{l} \text{Comando memoria } K_i \begin{cases} \text{Set } H_{i-1} \\ \text{Reset } k_{i+1} \end{cases} \quad \text{equazione memoria } K_i = (H_{i-1} + k_i) \cdot k_{i+1} \\ \text{Transizione del passo } i^{mo} \quad \rightarrow H_i = k_i \cdot x_i \\ \text{Comando bobina dell'elettrovalvola} \quad \rightarrow Y_i = k_i \end{array} \right.$$



Le equazioni dei due passi successivi si traducono nel seguente circuito:



### 8.7.1 Primo ed ultimo modulo

In un ciclo continuo conviene considerare le fasi su di una circonferenza ne viene che:

La prima fase viene attivata solamente se è stata attivata l'ultima fase "n" ed è tornato il segnale di avvenuto comando  $x_n$

Il segnale di comando della memoria  $k_1$  prima fase è fornito da quello di transizione dell'ultimo passo  $n$ , dato dall'AND tra il segnale di attivazione dell'ultima memoria  $k_n$  e il segnale  $x_n$  di attuazione del comando.

$$H_n = k_n \cdot x_n \quad (8.7.1.1)$$

La prima memoria  $k_1$  sarà attivata da quello di transizione  $H_n$  in AND con lo Start e sarà disattivata dal segnale di attivazione della memoria successiva  $k_2$ .

$$K_1 = (H_n \cdot Start + k_1) \cdot \overline{k_2} \quad (8.7.1.2)$$

Il segnale di attivazione della prima memoria  $k_1$  resetta la memoria precedente, coincidente con l'ultima memoria.

$$K_n = (H_{n-1} + k_n) \cdot \overline{k_1} \quad (8.7.1.3)$$

Nelle condizioni iniziali, quando viene alimentato il circuito, si presenta lo stesso problema osservato nel sequenziatore pneumatico:

Pigiato lo Start, deve attivarsi la prima memoria  $K_1$ . Ma, perché ciò avvenga occorre che sia attivata l'ultima bobina del segnale di transizione  $H_n$ , formato dall'AND tra il segnale di memoria  $K_n$  e  $x_n$ .

La memoria  $K_n$  dell'ultimo modulo, inizialmente, non è attivata: risulta nullo il segnale  $H_n$  e, quindi, il ciclo non può partire.

Occorre fare in modo che, nelle condizioni iniziali, la memoria dell'ultimo passo sia attivata mentre tutte le precedenti siano *disattivate*.

Si risolve il problema aggiungendo una bobina ausiliaria  $Pr$  comandata da un pulsante  $Prs$ . Sull'ultima memoria si pone in *OR* ai contatti di eccitazione un contatto normalmente aperto  $Pr$  della bobina ausiliaria; mentre, della stessa, su tutte le altre precedenti memoria, si pone in serie alle loro bobine un contatto normalmente chiuso  $\overline{Pr}$ .

L'equazione dell'ultima memoria risulta:

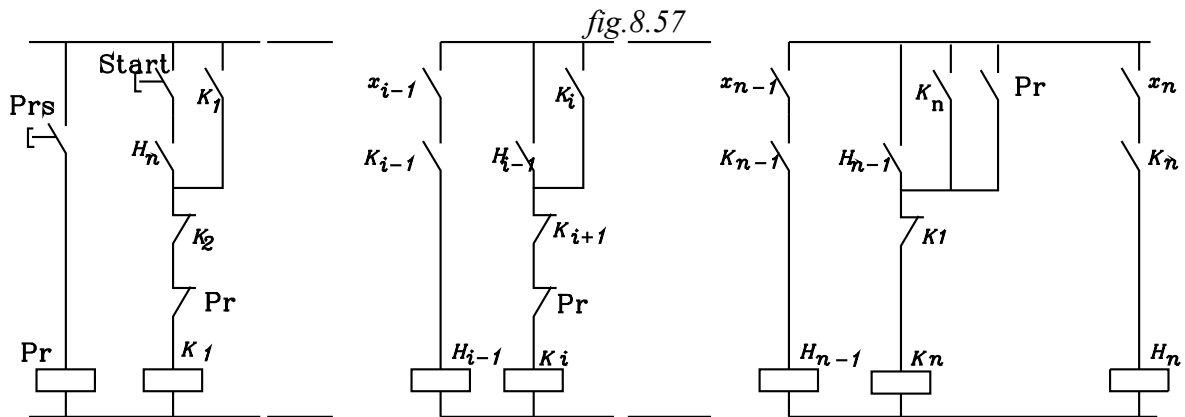
$$K_n = (H_{n-1} + k_n + Pr) \overline{k_1} \quad (8.7.1.4)$$

L'equazione generica della memoria  $i^{ma}$  è del tipo:

$$K_i = (H_{i-1} + k_i) \cdot \overline{k_{i+1}} \cdot \overline{Pr} \quad (8.7.1.5)$$

Dove:

$$Pr = Prs \quad (8.7.1.6)$$



In tal modo, prima di iniziare il ciclo, si pigia il tasto  $Prs$ . Si eccita la bobina  $Pr$  e il contatto normalmente aperto  $Pr$  si chiude eccitando l'ultima memoria  $K_n$ , mentre tutti i contatti normalmente chiusi  $\overline{Pr}$  si aprono garantendo la diseccitazione di tutte le alte memorie.

Pigiato lo  $Start$  può iniziare il ciclo. Il tasto di preset  $Prs$  deve essere aperto: esso viene chiuso in un solo istante prima di iniziare il ciclo.

Si consideri un semplice esempio che chiarisce il metodo proposto.



---

*IN LABORATORIO*

---

### 8.7.2 Esempio di circuito con segnale bloccante risolto con il metodo delle memorie in cascata.

Si debba effettuare il ciclo ad  $L$  più volte risolto con i diversi metodi.

CICLO  $A^+ B^+ B^- A^-$

Ad ogni passo è associata una memoria che comanda la bobina dell'apposita elettrovalvola, e precisamente:

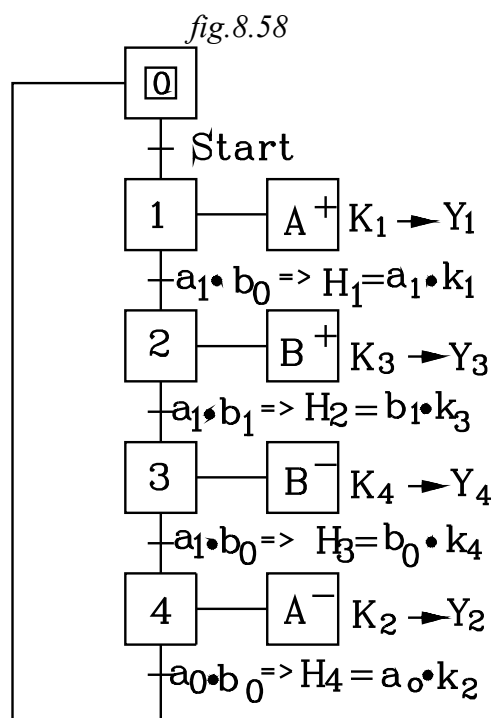
Passo	Stato comandato	Memoria	Elettrovalvola
1°	$A^+$	$K_1$	$Y_1$
2°	$B^+$	$K_3$	$Y_3$
3°	$B^-$	$K_4$	$Y_4$
4°	$A^-$	$K_2$	$Y_2$

Nel *grafcet* accanto agli stati comandati nelle fasi sono stati indicati le memorie e le bobine dell'elettrovalvole.

Nelle zone di transizione sono riportate le combinazioni complete dei fine corsa. Accanto viene scritto il segnale di comando della fase successiva, costituito *dall'AND* tra la variabile attiva e il segnale di memoria attivato nella fase.

Si indichi con  $Pr$  la bobina ausiliaria, adoperata per settare l'ultima memoria e resettare tutte le altre e con  $Prs$  il tasto che la eccita.

$$Pr = Prs$$



Seguendo i successivi passi si ha:

1° Passo

Il segnale di comando è fornito dall'ultimo stato di transizione:

$$H_1 = a_0 \cdot k_2$$

La prima memoria è settata dal segnale  $H_1$  in AND con lo Start ed è resettata dal segnale della memoria successiva  $k_3$  e dal segnale di preset della bobina ausiliaria  $Pr$

$$K_1 = (H_1 \cdot Start + k_1) \cdot \overline{k_3} \cdot \overline{Pr}$$

La memoria  $K_1$  comanda la bobina  $Y_1$  dell'elettrovalvola

$$Y_1 = k_1$$

Segue lo stato di transizione del primo passo.

Si ha il passaggio alla fase successiva solamente quando attivata la memoria  $k_1$ , comandato  $Y_1$  viene toccato il finecorsa  $a_1$ . Viene così elaborato il segnale di comando del II passo:

$$H_1 = a_1 \cdot k_1$$

### 2° Passo

La seconda memoria  $k_3$  è settata dal segnale  $H_1$  ed è resettata dal segnale della memoria successiva  $k_4$  e dal segnale di preset della bobina ausiliaria  $Pr$

$$K_3 = (H_1 + k_1) \cdot \overline{k_4} \cdot \overline{Pr}$$

La memoria  $K_3$  comanda la bobina  $Y_3$  dell'elettrovalvola

$$Y_3 = k_3$$

Segue lo stato di transizione del secondo passo.

Si ha il passaggio alla fase successiva solamente quando attivata la memoria  $k_3$ , comandato  $Y_3$  viene toccato il finecorsa  $b_1$ . Viene così elaborato il segnale di comando del III passo:

$$H_2 = b_1 \cdot k_3$$

### 3° Passo

La terza memoria  $k_4$  è settata dal segnale  $H_2$  ed è resettata dal segnale della memoria successiva  $k_2$  e dal segnale di preset della bobina ausiliaria  $Pr$

$$K_4 = (H_2 + k_3) \cdot \overline{k_2} \cdot \overline{Pr}$$

La memoria  $K_4$  comanda la bobina  $Y_4$  dell'elettrovalvola

$$Y_4 = k_4$$

Segue lo stato di transizione del terzo passo.

Si ha il passaggio alla fase successiva solamente quando attivata la memoria  $k_4$ , comandato  $Y_4$  viene toccato il finecorsa  $b_0$ . Viene così elaborato il segnale di comando del IV passo:

$$H_3 = b_0 \cdot k_4$$

### 4° Passo

La quarta memoria  $k_2$ , essendo l'ultima è diversa dalle altre.

Essa è settata dal segnale  $H_3$  e dal segnale  $Pr$  della bobina ausiliaria.

Il segnale di Reset è dato dal segnale della memoria successiva che corrisponde a quella del primo passo  $k_1$ .

$$K_2 = (H_3 + Pr + k_2) \cdot \overline{k_1}$$

La memoria  $K_2$  comanda la bobina  $Y_2$  dell'elettrovalvola

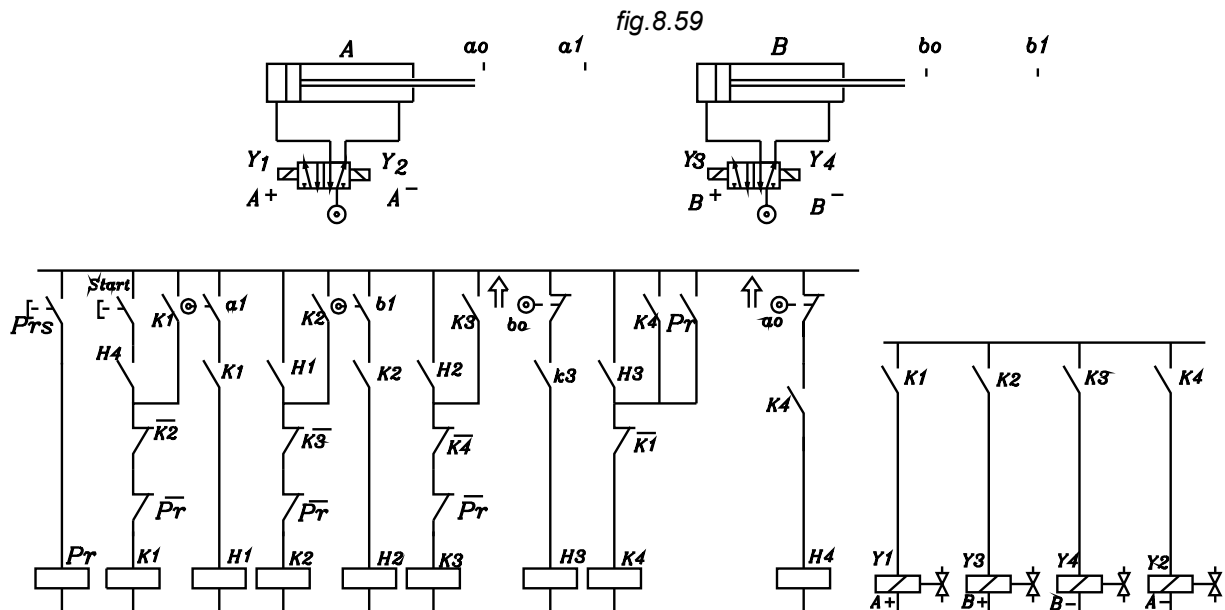
$$Y_2 = k_2$$

Segue lo stato di transizione del quarto passo.

Si ha il passaggio alla fase successiva solamente quando attivata la memoria  $k_2$ , comandato  $Y_2$ , viene toccato il finecorsa  $a_0$ . Viene così elaborato il segnale di comando del I passo:

$$H_4 = a_0 \cdot k_2$$

Si ottiene così il circuito elettropneumatico di figura:



### 7.8.3 Schema a contatti sequenziale semplificato

Nel circuito sequenziale studiato, occorre aggiungere il segnale di inizializzazione "Pr" per fare in modo che, all'inizio del ciclo, sia attivata l'ultima memoria  $K_n$ , il cui segnale, in AND con  $x_n$  di transizione fornisca il segnale  $H_n$  di Set della prima memoria  $K_1$ :

$$H_n = x_n \cdot k_n$$

$$K_1 = (H_n + k_1) \cdot k_2$$

Si può ovviare al segnale di inizializzazione  $Pr$ , facendo dipendere l'ultimo segnale di transizione  $H_n$  (di comando del primo modulo) solamente dal segnale  $x_n$  di avvenuto comando e non, anche, dal settaggio dell'ultima memoria  $K_n$ .

Ciò è possibile solamente se la combinazione dei finecorsa  $x_n$  di transizione dell'ultima passo non si presenta uguale in una delle transizioni dei passi precedenti e, quindi, risulta di per se stesso un segnale distinto da tutti gli altri.

Si può far dipendere l'ultimo segnale di transizione  $H_n$  (di comando del primo modulo) solamente dal segnale  $x_n$  di avvenuto comando solamente se questo non è un segnale bloccante.

Si può porre in tal caso come segnale di comando della prima memoria:

$$H_n = x_n$$

È da osservare che con l'eliminazione del segnale di inizializzazione semplifica il circuito ma toglie una verifica iniziale sulle memorie.

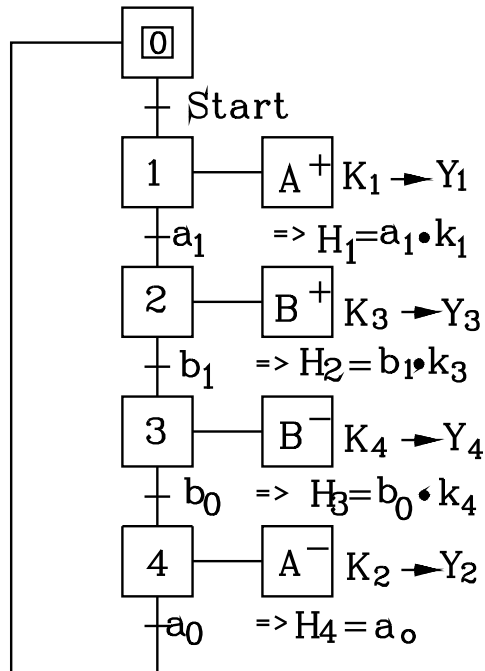
---

 IN LABORATORIO
 

---

## 8.7.3 Con la semplificazione anzi detta si effettui il ciclo:

fig.8.60

A<sup>+</sup> B<sup>+</sup> B<sup>-</sup> A<sup>-</sup>

Nel graficet, in ogni stato di transizione della fase considerata, è riportato il finecorsa che rappresenta la variabile attiva del passo e, tra parentesi, la combinazione dei finecorsa toccati (o attivati nel caso di finecorsa costituiti da sensori di prossimità).

Come si può notare, la combinazione dei segnali di finecorsa "a<sub>0</sub> · b<sub>0</sub>" dell'ultimo passo non è un segnale bloccante. Si può quindi far dipendere il segnale di comando della memoria k<sub>1</sub> del primo passo solamente dal segnale di finecorsa che rappresenta la variabile attiva a<sub>0</sub> dell'ultimo passo.

Accanto ad ogni quadrato, contenente l'operazione da eseguire, è indicato il segnale di memoria e l'elettrovalvola da esso comandata.

Nello stato di transizione viene indicato il segnale di comando della fase successiva, costituito dal finecorsa che viene toccato alla fine della transizione in AND con il

segnale di memoria della fase considerata.

Nell'ultimo stato di transizione, come si è detto, si considera solo il segnale di finecorsa a<sub>0</sub> che rappresenta la variabile attiva.

Le equazioni logiche sono:

$$H_4 = a_0 \cdot \text{Start} \quad \text{Transizione 4° passo e comando 1° memoria}$$

1° Passo

$$K_1 = (H_4 + k_1) \cdot \overline{k_3} \quad \text{Settaggio e resettaggio 1° memoria } K_1$$

$$Y_1 = k_1 \quad \text{Comando dell'elettrovalvola } Y_1 (A^+)$$

$$H_1 = a_1 \cdot k_1 \quad \text{Transizione 1° passo e comando della memoria } K_3 \text{ del secondo passo.}$$

2° Passo

$$K_3 = (H_1 + k_3) \cdot \overline{k_4} \quad \text{Settaggio e resettaggio 2° memoria } K_3$$

$$Y_3 = k_3 \quad \text{Comando dell'elettrovalvola } Y_3 (B^+)$$

$$H_2 = b_1 \cdot k_3 \quad \text{Transizione 2° passo e comando della memoria } K_4 \text{ del terzo passo.}$$

3° Passo

$$K_4 = (H_2 + k_4) \cdot \overline{k_2} \quad \text{Settaggio e resettaggio 3° memoria } K_4$$

$$Y_4 = k_4 \quad \text{Comando dell'elettrovalvola } Y_4 (B^-)$$

$$H_3 = b_0 \cdot k_4 \quad \text{Transizione 3° passo e comando della memoria } K_2 \text{ del quarto passo.}$$

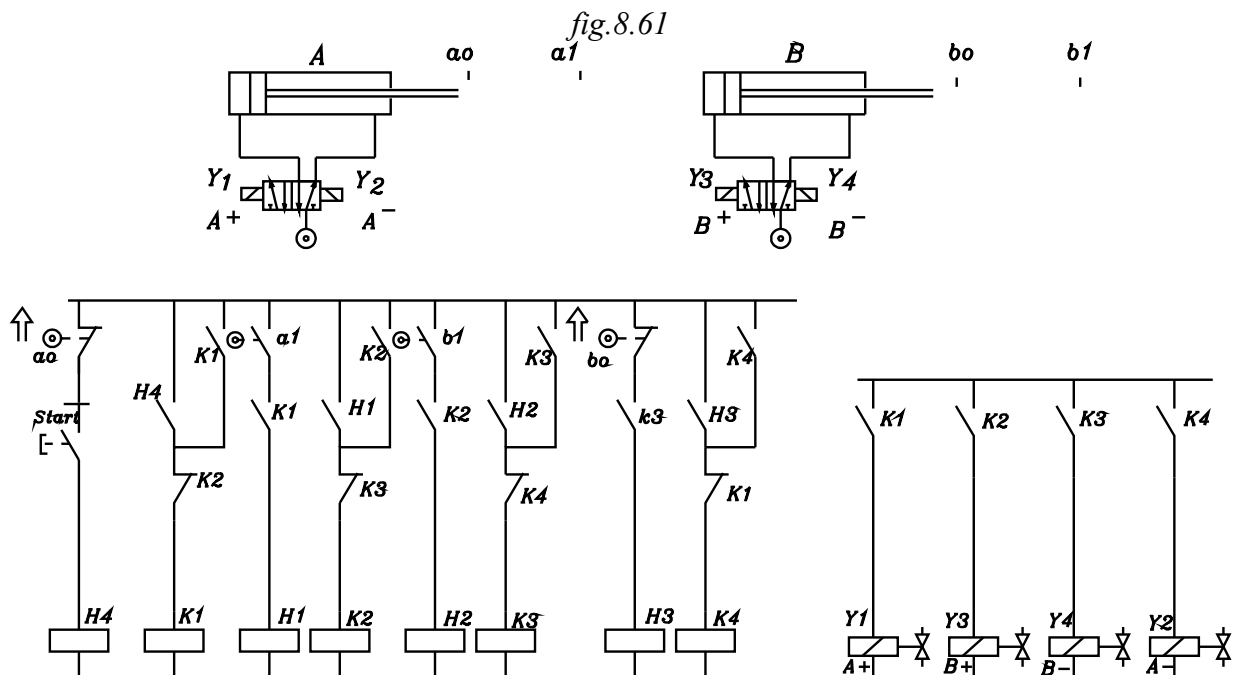
4° Passo

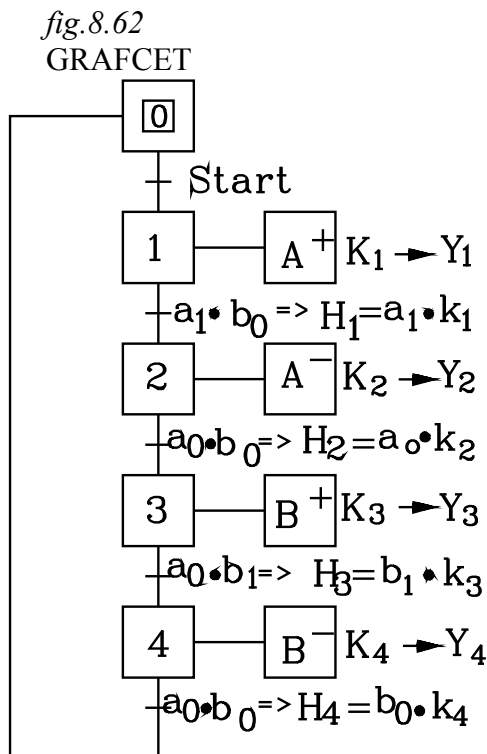
$K_2 = (H_3 + k_2) \cdot \overline{k_1}$       *Settaggio e resettaggio 4° memoria  $K_2$*

$Y_2 = k_2$       *Comando dell'elettrovalvola  $Y_2$  ( $A^-$ )*

$H_4 = a_0 \cdot Start$       *Transizione 4° passo e comando della memoria  $K_1$  del primo passo.*

Sigla componente	Componente
A	Cilindro a doppio effetto
B	Cilindro a doppio effetto
Start	Pulsante di Start
$a_0$	Finecorsa
$a_1$	Finecorsa $a_1$
$b_0$	Finecorsa $b_0$
$b_1$	Finecorsa $b_1$
K1	Bobina ausiliaria per il contatto del comando di Y1
K2	Bobina ausiliaria per il contatto del comando di Y2
K3	Bobina ausiliaria per il contatto del comando di Y3
K4	Bobina ausiliaria per il contatto del comando di Y4
Y1	Bobina dell'elettrovalvola bistabile di comando $A^+$
Y2	Bobina dell'elettrovalvola bistabile di comando $A^-$
Y3	Bobina dell'elettrovalvola bistabile di comando $B^+$
Y4	Bobina dell'elettrovalvola bistabile di comando $B^-$



CICLO  $A^+ A^- B^+ B^-$ 

Si debbano effettuare più cicli consecutivi, lasciando pigiato lo Start. Non è possibile in questo caso semplificare l'ultimo segnale di transizione in quanto la combinazione dei finecorsa  $a_0 \cdot b_0$  si è già verificata nel ciclo.

Rimanendo pigiato lo START, si verifica che due segnali di comando si attivano contemporaneamente:  $b_0 \cdot Start$  al rientro dello stelo di B che comanda un nuovo ciclo e  $a_0 \cdot Y_2$  al rientro dello stelo di A che comanda la fuoriuscita di B.

Si ha che quando rientra il cilindro B, allora fuoriesce A ( $b_0 \cdot Start$  è attivato), ma fuoriesce anche B ( $a_0 \cdot Y_2$  è attivato) e il ciclo si blocca.

Occorre differenziare il segnale di eccitazione di  $Y_1$  ponendo in AND il segnale  $Y_4$ . In tal caso, come si è detto nei sequenziatori, occorre

aggiungere il segnale di PRESET da porre in AND negato nelle linee di corrente che comandano l'elettrovalvole salvo l'ultima nella quale detto segnale viene posto in OR.

Nel caso di un circuito cablato, in elettropneumatica, occorre adoperare delle bobine ausiliarie che forniscono i segnali all'elettrovalvole dei cilindri.

Vengono adoperati gli stessi simboli del ciclo precedente.

Le equazioni logiche sono:

$$\begin{aligned}
 PR &= IN \\
 H4 &= b_0 \cdot K4 \cdot Start \\
 K1 &= (H4 + K4) \cdot \overline{K2} \cdot PR \\
 H1 &= a_1 \cdot K1 \\
 K2 &= (H1 + K2) \cdot \overline{K3} \cdot PR \\
 H2 &= a_0 \cdot K2 \\
 K3 &= (H2 + K3) \cdot \overline{K4} \cdot PR \\
 H3 &= b_1 \cdot K3 \\
 K4 &= (H3 + K4 + PR) \cdot \overline{K1}
 \end{aligned}$$

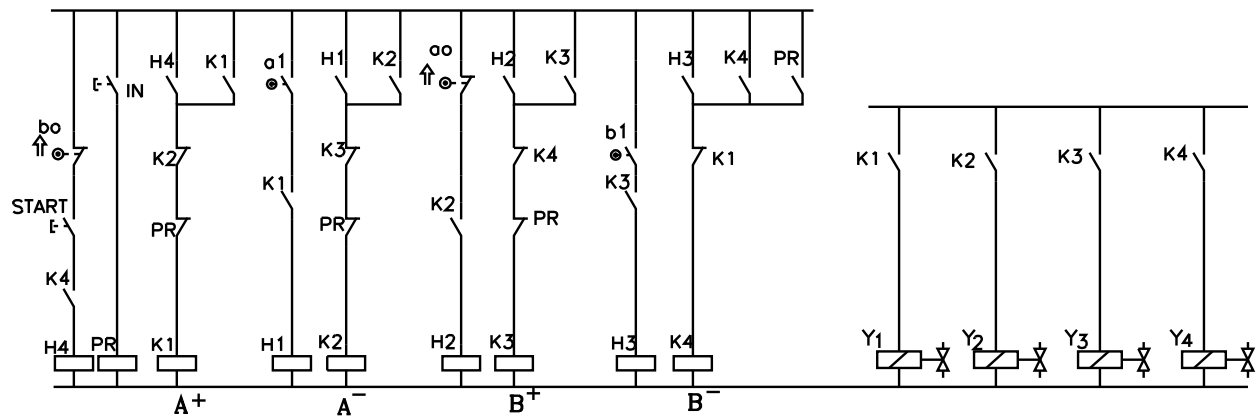
$$Y1 = K1$$

$$Y2 = K2$$

$$Y3 = K3$$

$$Y4 = K4$$

Il circuito risulta il seguente:



## 8.8 Temi proposti

### 8.1

Un foglio di lamiera deve essere profilato con deformazione a freddo. Il foglio viene posto a mano sull'apposita controstampo. Pigiati due Start il foglio viene profilato dall'apposito stampo comandato da un cilindro pneumatico. Effettuato il ciclo di profilatura un espulsore comandato da un altro cilindro espelle il foglio profilato.

Condizioni al contorno

Debbono essere impiegati due start per impegnare entrambi le mani. Deve essere previsto un pulsante Em di emergenza, attivando il quale i cilindri debbono tornare nella posizione di rientro e non risultano attivi gli Start.

Effettuare un disegno schematico del sistema - Stendere una appropriata relazione che spieghi la soluzione del problema con lo svolgimento dei seguenti punti:

- Descrizione delle soluzione del problema proposto con un disegno schematico del sistema.
- Lista dei componenti impiegati con tabella riassuntiva.
- Diagramma corsa - passo con l'indicazione delle condizioni al contorno.
- Stendere la tabella degli ordini e dei comandi.
- Effettuare il diagramma di flusso a barre orizzontali.
- Scegliere il metodo più appropriato per effettuare il ciclo di lavoro.
- Stendere le equazioni logiche.
- Stendere il ciclo elettropneumatico.
- Descrivere il funzionamento del ciclo elettropneumatico effettuato.

Risolvere il ciclo mediante il metodo delle mappe di Karough.

Suggerimento

Conviene porre il tasto di emergenza in modo da interrompere la corrente elettrica sulla linea di alimentazione ed assumere come valvole di potenza le monostabili; in modo che, quando viene interrotta l'eccitazione delle bobine, la commutazione data dalle molle portano l'aria nelle camere dei cilindri nel lato di rientro degli steli.

8.2 Risolvere il tema precedente mediante il metodo delle memorie in cascata.

8.3 Per effettuare la piegatura del foglio di cui al tema 1°, l'impianto deve essere modificato come segue.

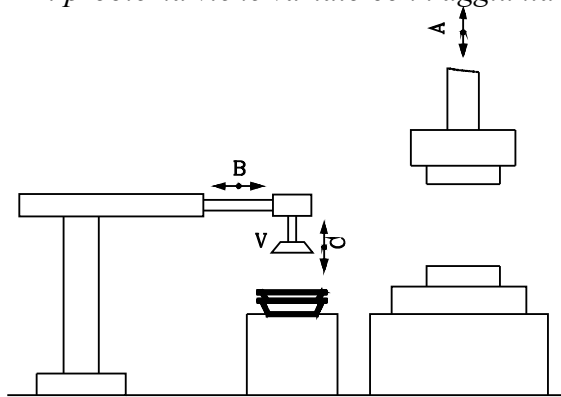
- Il controstampo su cui viene depositato il foglio di lamiera, nelle condizioni iniziali è al difuori di una zona protetta non accessibile dalla mani dell'operatore.
- Posto il foglio sul controstampo e attivati i due start, attraverso la corsa di un cilindro il controstampo si porta nella posizione di piegatura posta nella zona protetta.

- Viene quindi effettuata la profilatura del foglio attraverso uno stampo comandato da un altro cilindro pneumatico.
- Effettuata la piegatura il controstampo si porta di nuovo nella posizione iniziale ove un espulsore al disotto del foglio profilato lo espelle.

Effettuare la relazione svolgendo i diversi punti indicati nel tema 1°

Risolvere il ciclo di lavoro mediante il metodo delle memorie in cascata con valvole di potenza bistabili e monostabili.

- 8.4 Risolvere il tema precedente 3° mediante il metodo delle mappe di Kanaugh con valvole di potenza bistabili e monostabili .
- 8.5 Risolvere il tema precedente 3° mediante il metodo del sequenziatore con valvole di potenza bistabili.
- 8.6 Risolvere con il metodo delle memorie in cascata in elettropneumatica il problema proposto nel punto 6.7.1 risolto con tecnica pneumatica.  
Il problema viene variato con l'aggiunta di una specifica in più sul ciclo.



In una operazione di imbutitura, il pezzo da imbutire viene caricato a mano sull'apposito controstampo, posto in posizione esterna a quella di figura.

Pigiati i due Start **il controstampo viene condotto in posizione di imbutitura da un cilindro D non rappresentato in figura. Raggiunta tale posizione** viene effettuato dal cilindro A la doppia-corsa di imbutitura.

Tornato il cilindro A nella posizione superiore di riposo, il braccio B sposta la

ventosa V sull'asse dell'imbutito.

Raggiunto questo, la ventosa, comandata dal cilindro C, effettua la doppia corsa di andata (presa del pezzo imbutito) e ritorno nella posizione superiore di riposo.

Al raggiungimento di tale posizione Il cilindro B si ritrae e porta la ventosa con l'imbutito preso sul mucchio, ove verrà lasciato cadere.

Alla fine il cilindro D torna nella posizione iniziale.



[Clic per continuare](#)



[Clic per precedente](#)



[Clic per la pagina iniziale](#)