



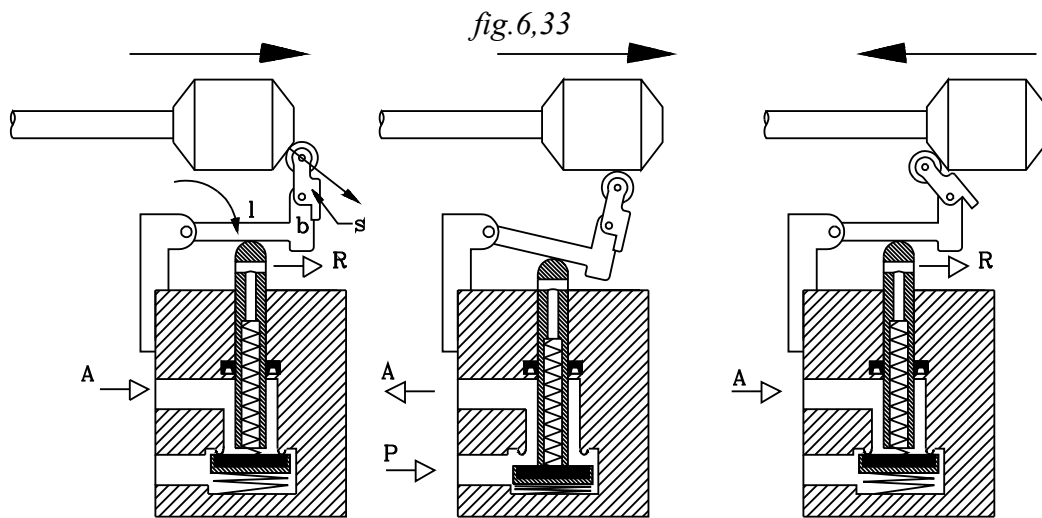
[e-mail per suggerimenti](#)

Impiego di finecorsa unidirezionali

Il finecorsa unidirezionale ha la leva di comando a rullo modificato rispetto a quello normale.

Il supporto, sul quale è montato il perno del rullo, contiene uno snodo s che può ruotare solamente in un senso e rimane rigido nel senso opposto. Nella figura lo snodo può ruotare in senso antiorario, ma non nel senso orario, in quanto in questo l'estremità di s va a battuta sul corpo b recante il perno dello snodo.

Nelle condizioni di riposo, prima che la camma passi sul rullo (fig.1), il puntalino forato è spinto dalla molla a toccare con la sua estremità sferica la leva l , che si pone in posizione parallela al corpo del finecorsa.



In questa posizione, il puntalino è distaccato dal piattello inferiore. Questo, spinto da un'altra molla posta al disotto di esso, va ad ostruire il passaggio dell'aria dall'alimentazione P all'utilizzatore A ; la quale è, invece, in comunicazione con lo scarico R , ottenuto dal foro sull'asse e all'estremità del puntalino: $A \rightarrow R$.

Quando la camma passa sul rullo, lo snodo, non potendo ruotare nel senso orario, comunica alla leva l un momento, che la fa ruotare attorno al fulcro, costringendo il puntalino ad scorrere sul suo asse, verso l'interno, fino a toccare e spostare il piattello otturatore dalla sua posizione di riposo.

Quando l'estremità del puntalino forato va ad appoggiarsi sul piattello, viene interdetto il passaggio dell'aria da A verso lo scarico R : viene otturata l'apertura del foro interno. L'abbassamento del piattello otturatore apre il passaggio dell'aria dall'alimentazione P all'utilizzazione A : $P \rightarrow A$.

La camma continuando la sua corsa, sorpassa il rullo del finecorsa. Il puntalino e la leva spinti dalle molle si riportano nella posizione di riposo, nella quale l'orifizio di utilizzazione A è in scarico: $A \rightarrow R$, mentre l'alimentazione P è interdetta.

Quando la camma torna indietro e passa sul rullo del finecorsa, lo snodo s ruota sul suo perno, senza comunicare alcun momento alla leva l . Questa durante la corsa di ritorno rimane nella posizione di riposo e il finecorsa non viene attivato: l'orifizio di utilizzazione A è in scarico: $A \rightarrow R$, mentre l'alimentazione P è interdetta.

Il finecorsa unidirezionale viene attivato solamente quando la camma tocca il rullo spostandosi nel senso nel quale lo snodo non può ruotare sul suo perno.

Il finecorsa unidirezionale va adoperato quando si vuole elaborare un segnale di breve durata: corrispondente al tempo che impiega la camma per passare sul rullo, quando si muove nel senso che determina la rotazione della leva l di comando (non ruota lo snodo).

Occorre osservare che, affinché il finecorsa unidirezionale possa assolvere alla sua funzione di generatore di segnale di breve durata, occorre posizionarlo rispetto alla camma in modo che, sicuramente, essa *possa sorpassare il rullo prima che termini la corsa attiva e alla fine di questa non rimanga a contatto con il rullo*.

Potendo ottenere con i finecorsa unidirezionali segnali di breve durata, essi possono essere adoperati per eliminare segnali bloccanti.

I segnali che con finecorsa normali durerebbero per una doppia corsa del cilindro da essi comandato, diventano di breve durata con l'impiego di finecorsa unidirezionali. Per cui, quando viene impartito il comando di pilotaggio opposto sulla valvola bistabile, sicuramente, non è presente quello della precedente commutazione, essendosi risolto nella breve durata.

Impiego di memorie

Nel caso che stiamo considerando, una stessa combinazione, dei segnali di finecorsa, si ripetono nelle fasi della sequenza ciclica degli stati dei cilindri; per cui, uno stesso segnale deve effettuare, in tempi diversi, comandi differenti.

Per eliminare la ripetizione dei segnali, si possono introdurre altri segnali, che memorizzano la sequenza, secondo la quale si sono succeduti, nel tempo, i segnali ripetuti. I segnali da introdurre vengono detti: di memoria.

Ad ogni combinazione ripetuta dei segnali di finecorsa, che si succede nel tempo, si associa un diverso segnale di memoria : esso risulta disattivato nelle altre ripetizioni.

In tal modo viene ricordata la fase nella quale si ha la ripetizione dei segnali di finecorsa.

I segnali di memoria, diversi l'uno dall'altro, si pongono in AND con i segnali ripetuti nelle fasi del ciclo. In tal modo non vi sono più segnali uguali che debbono effettuare comandi diversi; essi si contraddistinguono per almeno il segnale di memoria.

Verranno sviluppati due metodi razionali per l'introduzione dei segnali di memoria per l'eliminazione dei segnali di finecorsa ripetuti e bloccanti: *Metodo delle memorie in cascata - Metodo delle mappe di Karnaugh*.

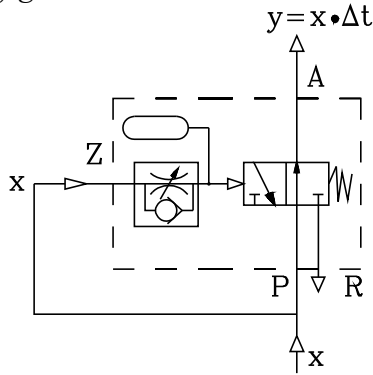
Sviluppiamo ora, con degli esempi i tre metodi accennati.

6.3.4.1 Temporizzazione dei segnali bloccanti

Come si è detto il segnale bloccante può essere eliminato trasformandolo in un impulso pneumatico che dura in un tempo Δt inferiore a quello che intercorre tra le due corse opposte del cilindro comandato.

Occorre determinare un circuito pneumatico che generi un impulso di durata Δt , quando viene azionato il finecorsa che produce il segnale bloccante. Il segnale del finecorsa bloccante, che durerebbe per un tempo t , corrispondente alla doppia corsa del cilindro comandato, viene trasformato in un impulso di durata $\Delta t < t$.

fig.6.34



Per generare l'impulso pneumatico di durata Δt si può adoperare un temporizzatore normalmente aperto.

Si invia il segnale di attivazione x , contemporaneamente sia all'alimentazione P che al pilotaggio Z del temporizzatore. In uscita da questo vi sarà aria, solamente quando viene inviato il segnale x , e per il tempo Δt occorrente a riempire il serbatoio S , in modo da raggiungere la pressione necessaria a commutare la $3/2 NA$ e interdire l'uscita A .

fig.6.35

t	x	$y = x \cdot \Delta t$
0	0	0
0	1	0
1	0	0
1	1	1

Si dice che Δt è allo stato logico 1 : $\Delta t = 1$, quando si è nell'intervallo di riempimento del serbatoio, in presenza del segnale pneumatico x , prima della commutazione della valvola $3/2$ e l'aria esce dal temporizzatore. Lo stato logico $\Delta t = 0$ si ha quando si è fuori dell'intervallo di riempimento del serbatoio (non si è nell'intervallo Δt).

Con questa posizione si può stendere la tabella della verità.

L'uscita y risulta allo stato logico 1, solamente quando si è

nell'intervallo di riempimento del serbatoio e è presente il segnale di attivazione x .

la funzione di uscita y si può esprimere nella forma:

$$y = x \cdot \Delta t$$

Ciò premesso si consideri un ciclo di lavoro di due cilindri nel quale si ha una ripetizione nelle fasi delle combinazioni dei segnali di finecorsa toccati (attivati).

IN LABORATORIO

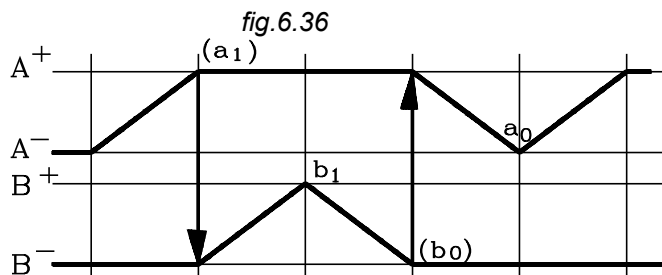
CICLO $A^+ B^+ B^- A^-$

Per brevità vengono omesse le parti riguardanti la *descrizione del problema e scelta dei componenti*. Ci si riferisca al ciclo di lavoro

Dall'analisi del diagramma corsa - passo, si rileva che il finecorsa a_1 , che comanda la fuoriuscita dello stelo del cilindro B , rimane attivato anche quando viene comandata la corsa opposta: permane per una doppia corsa del cilindro B comandato.

Così pure il finecorsa b_0 dopo aver comandato il rientro dello stelo di A permane attivato anche quando viene comandato il rientro.

Diagramma corsa - passo



I segnali di finecorsa a_1, b_0 sono bloccanti.

Tabella degli ordini

Comando	Stato comandato	Stato iniziale dei finacorsa attivati	Stato finale dei finacorsa attivati
A_1	A^+	$a_0 \cdot b_0 \rightarrow$	$a_1 \cdot b_0$
B_1	B^+	$a_1 \cdot b_0 \rightarrow$	$a_1 \cdot b_1$
B_0	B^-	$a_1 \cdot b_1 \rightarrow$	$a_1 \cdot b_0$
A_0	A^-	$a_1 \cdot b_0 \rightarrow$	$a_0 \cdot b_0$

Equazioni logiche complete e con variabili attive

Si supponga che l'unica condizione al contorno sia data dallo Start di inizio ciclo.

Si ottengono le equazioni:

Equazioni logiche complete	Equazioni logiche con variabili attive
$A^+ = a_0 \cdot b_0 \cdot Start$	$A^+ = a_0 \cdot Start$
$B^+ = a_1 \cdot b_0 \quad *$	$B^+ = a_1 \quad *$
$B^- = a_1 \cdot b_1$	$B^- = b_1$
$A^- = a_1 \cdot b_0 \quad *$	$A^- = b_0 \quad *$

Dall'esame delle equazioni logiche complete si constata che nelle fasi vi è una ripetizione delle combinazioni dei segnali di finecorsa. Inoltre dalla osservazione delle equazioni logiche complete si rileva facilmente se la variabile attiva risulta bloccante o meno.

Il finecorsa a_1 è attivo sia quando comanda la corsa B^+ ($B^+ = a_1 \cdot b_0$), che quando viene comandata, da b_1 , la corsa opposta B^- ($B^- = a_1 \cdot b_1$).

- Il finecorsa b_0 è attivo sia quando comanda la corsa A^- ($A^- = a_1 \cdot b_0$), che quando viene comandata, da a_0 , la corsa opposta A^+ ($A^+ = a_0 \cdot b_0 \cdot Start$).

Non si può togliere il segnale bloccante a_1 che comanda B^+ , ponendo in serie ad esso il segnale di finecorsa b_0 ; in quanto, si, questo non è presente nel comando B^- , ma risulta attivo insieme ad a_1 nella fase A^- per far rientrare A : il ciclo si scombina e si blocca.

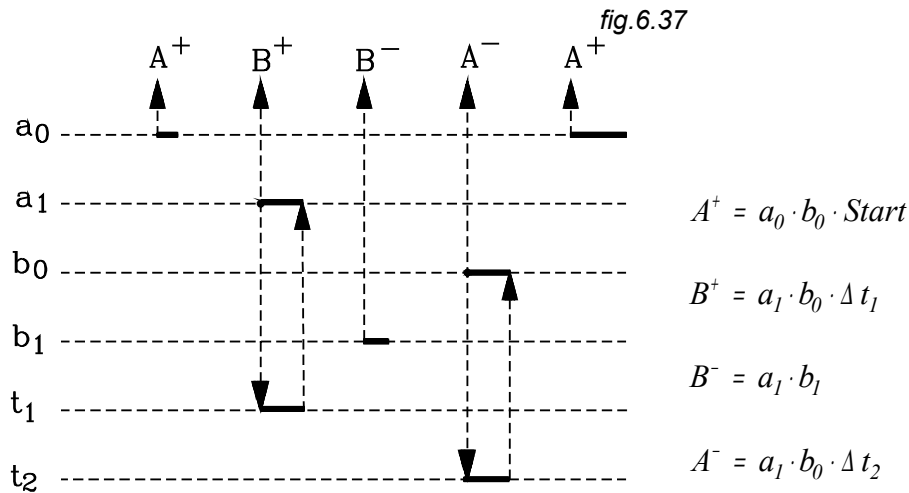
Trasformiamo i segnali bloccanti a_1, b_0 in segnali impulsivi, che durano un tempo Δt , inferiore a quello di durata della doppia corsa del cilindro comandato.

Ciò risulta molto evidente e chiaro, effettuando il diagramma di flusso a barre orizzontali.

Diagramma di flusso a barre orizzontali

Oltre alle barre orizzontali, rappresentanti le attivazioni dei segnali di finecorsa, vengono riportate quelle, indicanti le durate $\Delta t_1, \Delta t_2$ alle quali vengono ridotti, rispettivamente, i segnali bloccanti a_1, b_0 da due temporizzatori.

Per tracciare il diagramma di flusso si analizzino le equazioni logiche complete, e quelle con le variabili attive, osservando il perdurare dell'attivazione di queste nelle fasi.



Stato A^+ È comandato dal finacorsa a_0 . Esso è istantaneo: perdura nel tempo di contatto di a_0 con la camma, posta sullo stelo.

Stato B^+ È comandato dal finacorsa a_1 .

Analizzando le equazioni complete, si rileva che il segnale di finacorsa a_1 , senza l'azione del temporizzatore, perdurerebbe fino allo stato A^- , risultando attivo nella corsa di ritorno di B e quindi bloccante.

Il segnale di finacorsa a_1 viene posto in *AND* con il temporizzatore, il quale, dopo un tempo Δt_1 interdice l'uscita verso il pilotaggio B^+ della valvola bistabile, prima che venga attivato il pilotaggio B^- . In tal modo il segnale di comando di B^+ è espresso da:

$$B^+ = a_1 \cdot \Delta t_1$$

che non risulta bloccante.

Stato B^- È comandato dal finacorsa b_1 . Esso è istantaneo: perdura nel tempo di contatto di b_1 con la camma posta sullo stelo.

Stato A^- È comandato dal finacorsa b_0 .

Analizzando le equazioni complete, si rileva che il segnale di finacorsa b_0 , senza l'azione del temporizzatore, perdurerebbe fino allo stato A^+ , risultando attivo nella corsa di ritorno e quindi bloccante.

Il segnale di finacorsa b_0 viene posto in *AND* con il temporizzatore, il quale, dopo un tempo Δt_2 interdice l'uscita verso il pilotaggio A^- della valvola bistabile, prima che venga attivato il pilotaggio A^+ . In tal modo il segnale di comando di A^- è espresso da:

$$A^- = b_0 \cdot \Delta t_2$$

che non risulta bloccante.

I segnali di comando con le variabili attive e i temporizzatori si possono esprimere nella forma:

$$A^+ = a_0 \cdot Start$$

$$B^+ = a_1 \cdot \Delta t_1$$

$$B^- = b_1$$

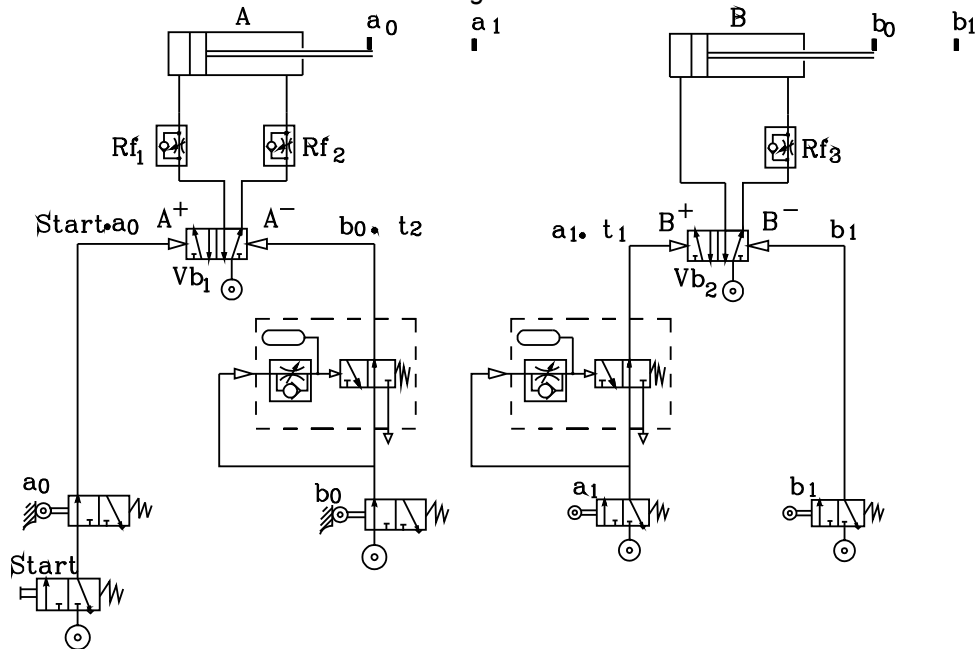
$$A^- = b_0 \cdot \Delta t_2$$

Circuito pneumatico

Dai segnali di comando si deriva facilmente il circuito pneumatico che realizza il ciclo.

$$A^+ \quad B^+ \quad B^- \quad A^-$$

fig.6.38



Stesura dello schema

Come al solito, basta trasformare le equazioni logiche degli stati comandati nei corrispondenti circuiti pneumatici.

In questo caso, va solamente rilevato che l'equazione $B^+ = a_1 \cdot \Delta t_1$ si traduce nella serie tra il finecorsa a_1 e il temporizzatore T1, la cui uscita va collegata con il pilotaggio B^+ della valvola bistabile Vb_2 .

Così l'equazione $A^- = b_0 \cdot \Delta t_2$ corrisponde alla serie tra il finecorsa b_0 e il temporizzatore T2, la cui uscita va collegata con il pilotaggio A^- della valvola bistabile Vb_1 .

Si supponga che le condizioni al contorno siano: - Il ciclo sia avviato da uno start - Le due corse del cilindro A siano lente- Il cilindro B abbia la corsa di fuoriuscita lenta e quella di rientro veloce.

Funzionamento

- Pigiato lo Start, questo prende l'aria dal finecorsa a_0 (toccato) e lo invia al pilotaggio A^+ della valvola Vb_1 . la quale, commutandosi alimenta la camera sinistra del cilindro A che compie la corsa A^+ di fuoriuscita dello stelo.
- Toccato il finecorsa a_1 , da questo esce l'aria che attraverso la valvola 3/2 NA del temporizzatore T1 viene inviata al pilotaggio B^+ della valvola Vb_2 . Questa si commuta e alimenta la camera sinistra del cilindro B che compie la corsa B^+ di fuoriuscita dello stelo. Dopo un intervallo di tempo Δt_1 , prima che venga toccato il finecorsa b_1 , il temporizzatore interrompe l'invio dell'aria al pilotaggio B^+ .
- In tal modo, quando viene azionato il finecorsa b_1 che invia l'aria al pilotaggio B^- , quello opposto B^+ non risulta contemporaneamente attivato: la valvola Vb_2 può commutarsi inviando l'aria nella camera destra del cilindro B che compie la corsa B^- di rientro stelo.
- Toccato il finecorsa b_0 , da questo esce l'aria che attraverso la valvola 3/2 NA del temporizzatore T2 viene inviata al pilotaggio A^- della valvola Vb_1 . Questa si commuta e alimenta la camera destra del

cilindro A che compie la corsa A^- di rientro dello stelo. Dopo un intervallo di tempo Δt_2 , prima che venga toccato il finecorsa a_0 , il temporizzatore interrompe l'invio dell'aria al pilotaggio A^- .

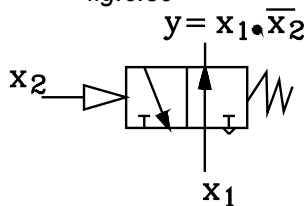
- In tal modo, quando viene azionato il finecorsa a_0 che invia l'aria al pilotaggio A^+ , quello opposto A^- non risulta contemporaneamente attivato: la valvola Vb_1 può commutarsi inviando l'aria all'ingresso dello Start per iniziare un altro ciclo.

Nel ciclo ora esaminato, per ottenere l'impulso pneumatico di durata Δt , si è adoperato un temporizzatore normalmente aperto. Ci si può domandare se si può impiegare un temporizzatore normalmente chiuso per ottenere lo stesso impulso. Ciò è possibile con una valvola che effettui la funzione di inibizione.

IN LABORATORIO

Per esercitazione si risolve il problema di ottenere un impulso pneumatico impiegando un temporizzatore normalmente chiuso. E questo anche perché è più usuale avere come dotazione di laboratorio un temporizzatore NC che non quello NA.

fig. 6.39



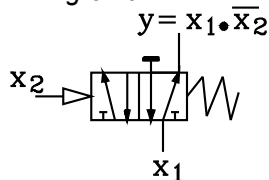
La funzione di inibizione si può ottenere con una valvola 3/2 NA monostabile con il segnale di inibizione posto sul pilotaggio e quello inibito inviato sul condotto P di alimentazione.

Il segnale x_1 va in uscita se non è presente quello di inibizione x_2 . Posto questo sul pilotaggio la valvola si commuta interdicendo l'aria in uscita.

La funzione binaria fornita dalla valvola è:

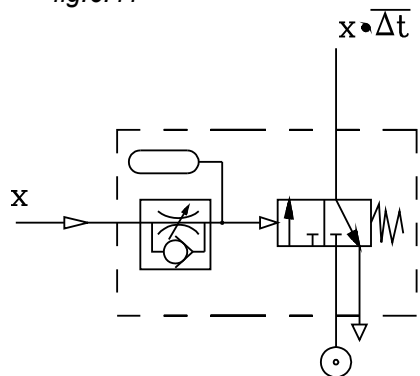
$$y = x_1 \cdot \overline{x_2}$$

fig. 6.40



In mancanza di una valvola 3/2 NA monostabile, si può ottenere la funzione di inibizione anche con una 5/2 monostabile, con l'utilizzatore A otturato: la valvola 5/2 monostabile si tramuta in una 3/2 NA monostabile.

fig. 6.41



Consideriamo ora il temporizzatore normalmente chiuso. Inviato il segnale pneumatico x al pilotaggio Z , si ha l'aria in uscita, dopo l'intervallo di tempo Δt , occorrente per riempire il serbatoio S e ottenere la sufficiente pressione per la commutazione della valvola 3/2 NC (interna al temporizzatore).

Si ha così il segnale y di uscita dal temporizzatore quando è presente il segnale x e non si è nell'intervallo Δt : quando Δt è allo stato logico 0: $\Delta t = 0$; intendendo con ciò che si è fuori dell'intervallo Δt di ritardo del temporizzatore. Si intende che:

- $\Delta t = 0$ Vuol dire che si è fuori dell'intervallo nel quale il temporizzatore interdice l'uscita dell'aria. Lo "0" al secondo membro ha il significato di stato logico "falso": non è vero che si è nell'intervallo Δt .
- $\Delta t = 1$ Vuol dire che si è nell'intervallo nel quale il temporizzatore interdice l'uscita dell'aria. La cifra "1" al secondo membro ha il significato di stato logico "vero": è vero che si è nell'intervallo Δt .

fig.6.42

t	x	y
0	0	0
0	1	1
1	0	0
1	1	0

Con queste precisazioni si può stendere la tabella della verità, dalla quale si ricava l'equazione logica di uscita dal temporizzatore:

$$y = x \cdot \overline{\Delta t}$$

Si ha in uscita aria solamente nella combinazione: $\Delta t = 0$ (al di fuori dell'intervallo di ritardo) e $x=1$ (in presenza di aria al pilotaggio).

Per ottenere l'impulso pneumatico nel tempo Δt occorre che: inviato il segnale x , questo rimanga attivo fino a che non venga interdetto da un altro segnale di inibizione.

Per ottenere ciò si adopera una valvola di inibizione *fig.6.43*. In questa si invia il segnale x nell'alimentazione P e, come segnale di inibizione da porre sul pilotaggio, si impiega quello di uscita dal temporizzatore normalmente chiuso, pilotato dallo stesso segnale x .

Infatti, dalla valvola di inibizione, il segnale x , applicato, continua ad uscire fino a che non perviene, al pilotaggio, il segnale del temporizzatore normalmente chiuso $y = x \cdot \overline{\Delta t}$, ritardato di Δt rispetto all'istante di applicazione di x .

Il segnale di uscita dalla valvola di inibizione risulterà:

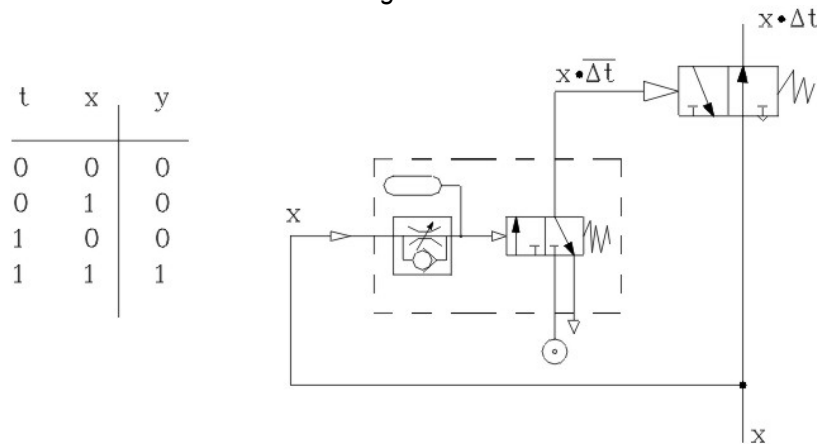
$$y = \overline{x \cdot \Delta t \cdot x}$$

Applicando il teorema di De Morgan si ottiene:

$$y = (\overline{x} + \overline{\Delta t}) \cdot x$$

$$y = x \cdot \Delta t$$

fig.6.43



Il ciclo $A^+ B^+ B^- A^-$ è stato precedentemente realizzato con un circuito, nel quale, attraverso i temporizzatori NA, sono stati trasformati in impulsi pneumatici i segnali bloccanti dei finecorsa $a_1 b_0$.

Lo stesso risultato si ottiene, impiegando per l'elaborazione di un impulso, il circuito studiato, composto dal temporizzatore NC e dalla valvola 3/2 NA monostabile che espleta la funzione di inibizione.

In mancanza delle valvole 3/2 NA monostabili possono essere impiegate le 5/2 monostabili con l'utilizzatore B otturato.

Il diagramma corsa - passo, la tabella degli ordini, il diagramma di flusso a bare orizzontali, sono argomenti già sviluppati e valgono anche per la stesura del presente circuito.

Si riassumono le equazioni logiche.

$$A^+ = a_0 \cdot \text{Start}$$

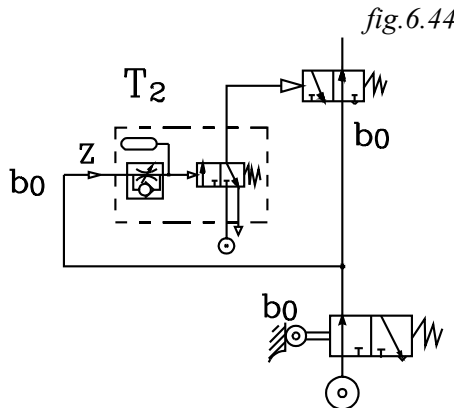
$$B^+ = a_1 \cdot \Delta t_1$$

$$B^- = b_1$$

$$A^- = b_0 \cdot \Delta t_2$$

Il circuito è quello riportato in figura fig.6.45.

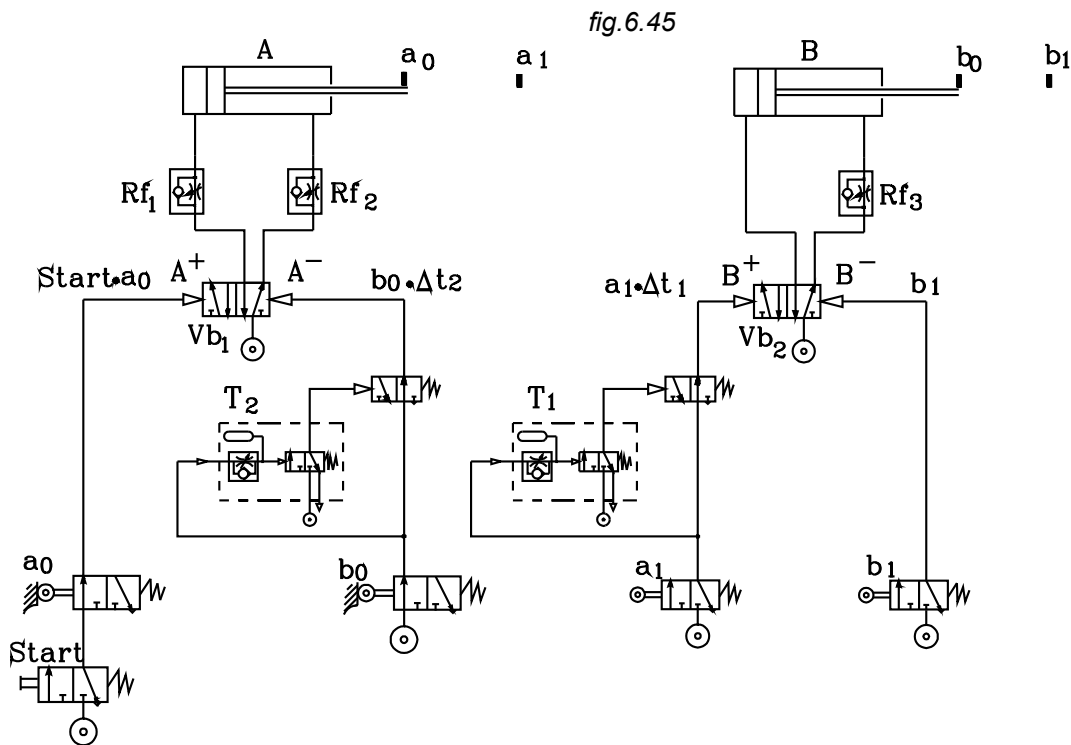
Stesura del circuito.



Riguardo alla stesura del circuito, rispetto a quello precedente, va puntualizzato in particolare il montaggio della parte riguardante la generazione dell'impulso pneumatico di durata Δt .

L'uscita del finecorsa b_0 va collegata, sia con il pilotaggio del temporizzatore NC "T2", sia con l'ingresso della valvola 3/2 NA monostabile (oppure 5/2 monostabile con l'utilizzatore B otturato). L'uscita di questa va collegata al pilotaggio A^- della valvola bistabile di potenza $Vb_{1..}$

Così pure l'uscita del finecorsa a_1 va collegata, sia con il pilotaggio del temporizzatore NC "T1", sia con l'ingresso della valvola 3/2 NA monostabile (oppure 5/2 monostabile con l'utilizzatore B otturato). L'uscita di questa va collegata al pilotaggio B^+ della valvola bistabile di potenza $Vb_{2..}$



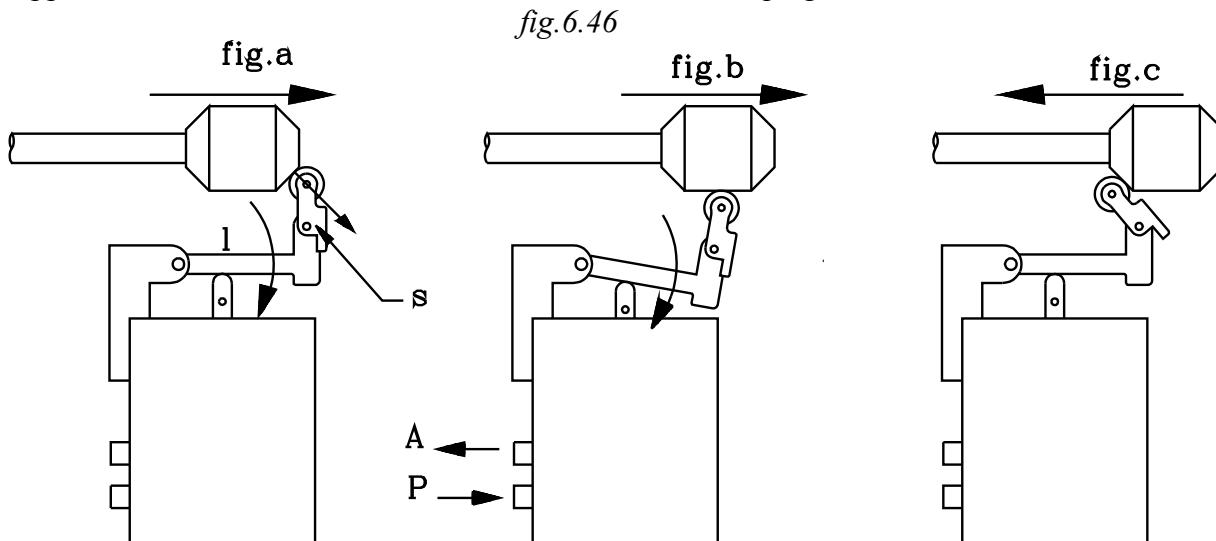
Funzionamento

- Pigiato lo Start, questo porta l'aria al finecorsa a_0 (toccato) che lo invia al pilotaggio A^+ della valvola Vb_1 , la quale, commutandosi, alimenta la camera sinistra del cilindro A che compie la corsa A^+ di fuoriuscita dello stelo.

- Toccato il finecorsa a_1 , da questo esce l'aria che va sia al pilotaggio del temporizzatore NC T_1 che all'orifizio di alimentazione della valvola 3/2 NA. Da questa esce l'aria che va al pilotaggio B^+ della valvola V_{b_2} . Questa si commuta e alimenta la camera sinistra del cilindro B che compie la corsa B^+ di fuoriuscita dello stelo. Dopo un intervallo di tempo Δt_1 , prima che venga toccato il finecorsa b_1 , il temporizzatore raggiunge la pressione tale da commutare la valvola interna 3/2 NC, da questa esce l'aria che commuta la valvola 3/2 NA, portandola in posizione di chiusura. Si interrompe così l'invio dell'aria al pilotaggio B^+ della valvola V_{b_2} .
- In tal modo, quando viene azionato il finecorsa b_1 che invia l'aria al pilotaggio B^- , quello opposto B^+ non risulta contemporaneamente attivato: la valvola V_{b_2} può commutarsi e portare l'aria nella camera destra del cilindro B che compie la corsa B^- di rientro stelo.
- Toccato il finecorsa b_0 , da questo esce l'aria che va sia al pilotaggio del temporizzatore NC "T2" che all'orifizio di alimentazione della valvola 3/2 NA. Da questa esce l'aria che va al pilotaggio A^- della valvola V_{b_1} . Questa si commuta e alimenta la camera destra del cilindro A che compie la corsa A^- di rientro dello stelo. Dopo un intervallo di tempo Δt_1 , prima che venga toccato il finecorsa a_0 , il temporizzatore raggiunge la pressione tale da commutare la valvola interna 3/2 NC, da questa esce l'aria che commuta la valvola 3/2 NA, portandola in posizione di chiusura. Si interrompe così l'invio dell'aria al pilotaggio A^- della valvola V_{b_1} .
- In tal modo, quando viene azionato il finecorsa a_0 che invia l'aria al pilotaggio A^+ , quello opposto A^- non risulta contemporaneamente attivato: la valvola V_{b_1} può commutarsi inviando l'aria all'ingresso dello Start per iniziare un altro ciclo.

6.3.4.2 Eliminazione del segnale bloccante con finecorsa unidirezionali

Un segnale bloccante può essere trasformato in uno di breve durata, inferiore al tempo della doppia corsa del cilindro da esso comandato, attraverso l'impiego di un finecorsa unidirezionale.



Questo viene attivato per il tempo che la camma, posta sullo stelo, passa sul rullo (fig.a - fig.b) e comunica una rotazione alla leva l , attraverso lo snodo s che non può ruotare in quel senso. La leva l rotando aziona il finecorsa, portando in uscita l'aria.

Quando la camma ha sorpassato il finecorsa, la leva azionata dalla molla, ritorna nella posizione di riposo, chiudendo il passaggio dell'aria verso l'uscita.

Il finecorsa si disattiva, e rimane disattivato, anche quando la camma torna a toccare il rullo nella corsa opposta a quella nella quale l'aveva attivato (fig.c).

Infatti nella corsa opposta della camma, questa, quando passa sul rullo, fa ruotare lo snodo nel senso ad esso possibile e non viene comunicata alcuna rotazione alla leva l : il finecorsa rimane disattivato.

Perché il meccanismo funzioni occorre, evidentemente, porre il finecorsa in posizione tale che, quando viene azionato dalla camma, questa poi lo sorpassi, prima che lo stelo del cilindro sia del tutto fuoriuscito o rientrato.

Occorre a tal proposito ricordare che i segnali sono di tipo *ON-OFF*, con i quali sono garantite le corse di fuoriuscita e rientro dello stelo. Essendo il segnale che stiamo trattando bloccante, esso, anche con un finecorsa normale, perdurerebbe fino al completo sfilo o rientro. Un finecorsa normale, però, verrebbe disattivato quando la camma lo sorpassa ma di nuovo attivato quando lo tocca di nuovo nella corsa di ritorno (*deve essere, invece, disattivato perché si sta comandando la corsa opposta*).

Con l'impiego di due finecorsa unidirezionali, si può risolvere, in modo molto semplice, il problema di togliere i due segnali bloccanti, riscontrati nel ciclo $A^+ B^+ B^- A^-$ già realizzato con i temporizzatori.

IN LABORATORIO

Ciclo $A^+ B^+ B^- A^-$ Impiego di finecorsa unidirezionali

Per quanto riguarda il diagramma corsa passo, la tabella degli ordini, si fa riferimento al precedente caso risolto con i temporizzatori.

Riassumiamo qui i comandi degli stati in forma completa e con le variabili attive.

Equazioni logiche complete	Equazioni logiche con variabili attive
$A^+ = a_0 \cdot b_0 \cdot Start$	$A^+ = a_0 \cdot Start$
$B^+ = a_1 \cdot b_0 \quad *$	$B^+ = a_1 \quad *$
$B^- = a_1 \cdot b_1$	$B^- = b_1$
$A^- = a_1 \cdot b_0 \quad *$	$A^- = b_0 \quad *$

I comandi con asterisco risultano bloccanti:

Il finecorsa a_1 è attivo sia quando comanda la corsa B^+ ($B^+ = a_1 \cdot b_0$), che quando viene comandata, da b_1 , la corsa opposta B^- ($B^- = a_1 \cdot b_1$).

- Il finecorsa b_0 è attivo sia quando comanda la corsa A^- ($A^- = a_1 \cdot b_0$), che quando viene comandata, da a_0 , la corsa opposta A^+ ($A^+ = a_0 \cdot b_0 \cdot Start$).

I due segnali: a_1 , b_0 , che risultano bloccanti con l'impiego di comuni finecorsa, si tramutano in altri di breve durata, adoperando finecorsa unidirezionali. Il segnale prodotto da questi dura soltanto per il tempo che la camma, posta sullo stelo, passa sul rullo, nel senso che produce la rotazione della leva l (*fig.b*).

In pratica i segnali a_1 , b_0 , vengono temporizzati in un tempo Δt che si compie prima dell'attuazione dello stato comandato: rispettivamente B^+ , A^- (completo sfilamento di dello stelo di B, completo rientro di A).

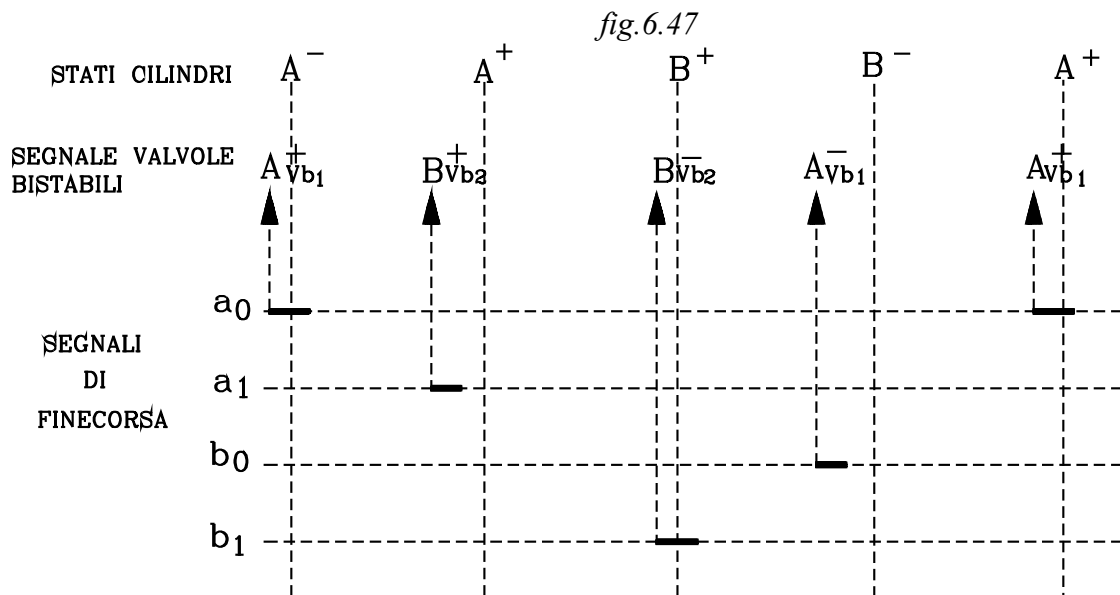
Quanto detto è chiaramente rappresentato nel diagramma di flusso a barre orizzontali di fig.6.47.

Nei diagrammi di flusso precedenti non si è fatta distinzione tra segnale di pilotaggio inviato alla valvola bistabile e lo stato del cilindro da essa comandato.

Nel presente diagramma di flusso, si riportano, in corrispondenza dei finecorsa attivati, i pilotaggi ai quali vengono inviati i segnali. Così il segnale a_0 viene inviato al pilotaggio A^+ della valvola V_{b1} .

Al disopra dei segnali di pilotaggio, in ritardo rispetto a questi, sono riportati gli stati dei cilindri nei quali si verifica l'attivazione del finecorsa e del corrispondente segnale di pilotaggio.

Si vuole ora porre in rilievo che il segnale di finecorsa, inviato al pilotaggio della valvola bistabile per il comando dello stato di una fase, avviene prima che si sia attuato lo stato della fase precedente. Così, per esempio, il segnale del finecorsa a_0 , che comanda lo stato A^+ di fuoriuscita dello stelo di A, inizia prima che lo stelo del cilindro A sia rientrato e abbia attuato lo stato precedente A^- .



Vi è un intervallo di tempo tra l'istante in cui viene toccato il finecorsa nella fase precedente a quella da esso comandata, e l'attuazione di detta fase.

I finecorsa unidirezionali elaborano un segnale di durata inferiore a detto intervallo: esso dura per il tempo che la camma passa sul rullo, fino al sorpasso, e ciò deve avvenire prima che si attui lo stato precedentemente comandato (completo sfilamento o rientro dello stelo del cilindro).

Precisamente si ha:

Finecorsa a_0 Comanda, in AND con lo Start, lo stato del cilindro A^+ .

Viene toccato prima che finisca la corsa di rientro dello stelo di A: attuazione dello stato precedente A^- .

Il finecorsa a_0 viene posizionato prima dello stato A^- , e in modo da continuare a toccare la camma, anche quando lo stelo è completamente rientrato. L'attivazione avviene nel

piccolo intervallo di tempo nel quale la camma tocca il rullo alla fine della corsa di rientro e la subitanea fuoriuscita dello stelo di A.

L'attivazione di a_0 è a cavallo dello stato A^- .

Finecorsa a_1 Comanda lo stato del cilindro B^+ .

Viene toccato prima che finisca la corsa di fuoriuscita dello stelo di A: attuazione dello stato precedente A^+ . Il finecorsa a_1 è unidirezionale; viene posizionato prima dello stato A^+ , e in modo da non toccare la camma quando lo stelo è completamente fuoriuscito (*attuazione di A^+*). L'attivazione avviene nel piccolo intervallo di tempo nel quale la camma tocca il rullo, prima che l'abbandoni alla fine della corsa A^+ .

L'attivazione di a_1 avviene nell'intervallo di tempo Δt_1 prima dello stato A^+ .

Finecorsa b_1 Comanda lo stato del cilindro B^- .

Viene toccato prima che finisca la corsa di fuoriuscita dello stelo di B: attuazione dello stato precedente B^+ . Il finecorsa b_1 viene posizionato prima dello stato B^+ . L'attivazione avviene nel piccolo intervallo di tempo nel quale la camma tocca il rullo alla fine della corsa di fuoriuscita e il subitaneo rientro dello stelo di B.

L'attivazione di b_1 è a cavallo dello stato B^+ .

Finecorsa b_0 Comanda lo stato del cilindro A^- .

Viene toccato prima che finisca la corsa di rientro dello stelo di B: attuazione dello stato precedente B^- . Il finecorsa b_0 è unidirezionale; viene posizionato prima dello stato B^- , e in modo da non toccare la camma quando lo stelo è completamente rientrato (*attuazione di B^-*). L'attivazione avviene nel piccolo intervallo di tempo nel quale la camma tocca il rullo, prima che l'abbandoni alla fine della corsa B^- .

L'attivazione di a_1 avviene nell'intervallo di tempo Δt_2 prima dello stato B^- .

Le equazioni logiche di comando sono:

$$A^+ = a_0 \cdot Start$$

$$B^+ = a_1 \rightarrow$$

$$B^- = b_1$$

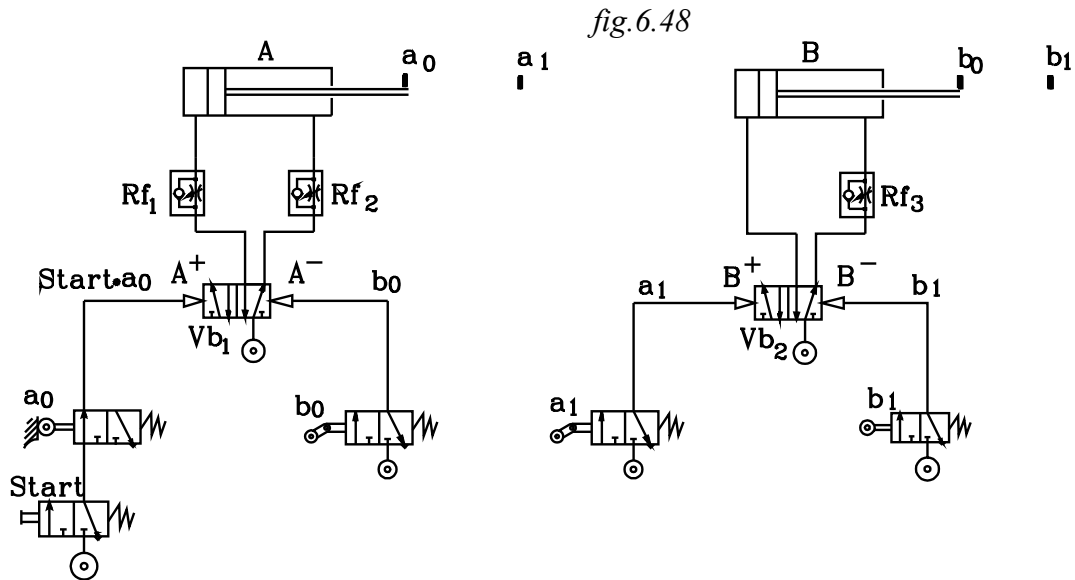
$$A^- = b_0 \rightarrow$$

Vicino ai finecorsa a_1 , b_0 si è posta una freccia per indicare che sono dei finecorsa unidirezionali.

Dalle equazioni logiche si ricava il circuito pneumatico.

Funzionamento

- Pigiato lo Start, questo porta l'aria dal finecorsa a_0 (toccato) che lo invia al pilotaggio A^+ della valvola V_{b_1} . la quale, commutandosi, alimenta la camera sinistra del cilindro A che compie la corsa A^+ di fuoriuscita dello stelo.
- Verso la fine della corsa A^+ la camma scorre su rullo del finecorsa unidirezionale a_1 generando, durante lo scorrimento, il segnale pneumatico che viene inviato al pilotaggio B^+ della valvola bistabile V_{b_2} . Questa si commuta e invia l'aria alla camera sinistra del cilindro B che effettua la corsa B^+ di fuoriuscita dello stelo. Dopo che la camma ha sorpassato il rullo di a_1 cessa il segnale pneumatico di pilotaggio B^+ .
- Occorre notare che, nella fuoriuscita dello stelo di B, viene toccato dalla camma il finecorsa b_0 , ma nel senso che provoca la rotazione dello snodo s e non della leva che comanda il finecorsa, il quale perciò non viene attivato.



- Alla fine della corsa B^+ viene toccato il finecorsa b_1 che invia l'aria al pilotaggio B^- della valvola bistabile V_{b_2} . Questa si commuta, essendo il pilotaggio B^+ disattivato, e l'aria viene inviata nella camera destra del cilindro B (quella sinistra è in scarico). Viene effettuata la corsa di ritorno B^- del rientro dello stelo.
- Verso la fine della corsa B^- la camma scorre su rullo del finecorsa unidirezionale b_0 generando, durante lo scorrimento, il segnale pneumatico che viene inviato al pilotaggio A^- della valvola bistabile V_{b_1} . Questa si commuta e invia l'aria alla camera destra del cilindro A che effettua la corsa A^- di rientro dello stelo. Dopo che la camma ha sorpassato il rullo di b_0 cessa il segnale pneumatico di pilotaggio A^- .
- Occorre notare che, nel rientro di A, viene di nuovo toccato dalla camma il finecorsa a_1 , ma nel senso che provoca la rotazione dello snodo s e non della leva che comanda il finecorsa, il quale perciò non viene attivato.
- Alla fine della corsa A^- viene toccato il finecorsa a_0 che invia l'aria al pilotaggio A^+ (A^- è disattivato). Se è pigiato lo Start inizia un altro ciclo.

6.4 ATTUAZIONE DI PARTICOLARI CONDIZIONI AL CONTORNO

Prima di procedere alla soluzione di circuiti contenenti segnali bloccanti con l'impiego di memorie, conviene soffermarsi sullo studio di circuiti adatti alla attuazione di particolari condizioni al contorno, quali: realizzazione di cicli continui - Segnali di emergenza- Condizioni di posizionamento dei cilindri in caso di emergenza.

Realizzazione di cicli automatici

Nei cicli finora studiati, per l'avvio, sono stati impiegati uno o due Start, composti da valvole 3/2 monostabili ad azionamento manuale. Queste vengono attivate solamente per il tempo nel quale si pigia il pulsante, e sono adatte, quindi, a comandare solamente un ciclo alla volta.

Si debba ora realizzare un circuito nel quale, schiacciando il pulsante S_1 si avvii un solo ciclo, schiacciando un pulsante S_2 si abbia automaticamente la ripetizione del ciclo, fino a che non si pigia un pulsante di Stop S_3 .

Si indichi con "s" il segnale di avvio del ciclo.

Con il pulsante di start S_1 deve essere emesso il segnale pneumatico "s" che duri per il tempo di azionamento manuale. Per ottenere ciò basta impiegare una valvola 3/2 NC monostabile.

$$s = S_1$$

Schiacciando il pulsante S_2 deve essere elaborato il segnale "s" che duri nel tempo, anche quando si lasci il pulsante; fino a che non si pigia il pulsante di Stop S_t . Per ottenere ciò occorre effettuare una memoria con precedenza alla cancellazione.

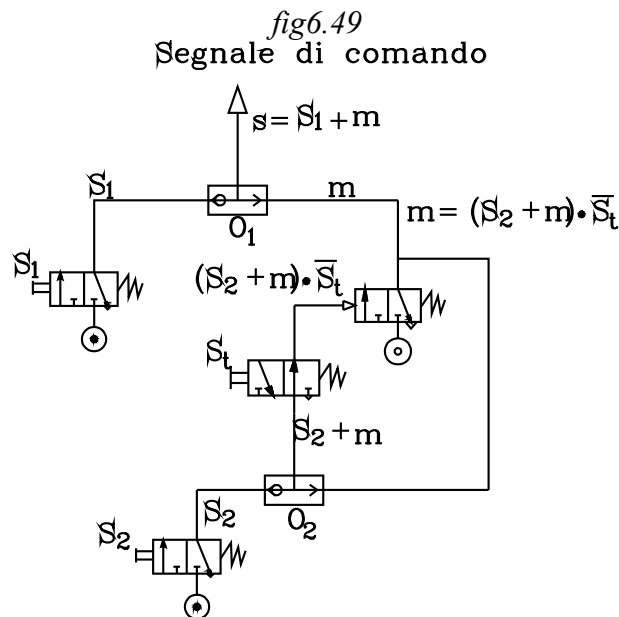
$$m = (S_2 + s) \cdot \bar{S}_t$$

Lo start S_2 si può realizzare con una valvola 3/2 NC monostabile, il segnale di resettaggio \bar{S}_t si ottiene con una valvola 3/2 NA monostabile

Il segnale "s" deve essere elaborato o nel primo oppure nel secondo caso (OR).

$$s = S_1 + m \quad \text{con} \quad m = (S_2 + s) \cdot \bar{S}_t$$

Dalla equazione logica si ricava il circuito rappresentato in figura.



Il funzionamento è evidente.

Pigiando lo start S_1 , l'aria attraverso la valvola OR "O1" viene inviata in uscita per costituire il segnale di comando.

Pigiando S_2 , l'aria va alla valvola OR "O2", passa attraverso la valvola S_t 3/2 NA monostabile e perviene al pilotaggio della 3/2 NC monostabile V. Da questa esce il segnale "s" che viene inviato sia alla valvola OR "O1" che alla "O2".

L'aria che perviene alla valvola "O1" costituisce il segnale di comando; quella che arriva alla valvola "O2" serve per automantenere l'attivazione della valvola V, pur quando si disattiva lo start S_2 . Infatti, quando il pulsante di questo viene rilasciato dopo averlo premuto, non si ha più aria all'uscita di dello start S_2 ; ma, alla valvola OR "O2", perviene aria dall'uscita della valvola V, che va al pilotaggio della stessa, passando attraverso la valvola NA S_t .

Quindi, premuto il pulsante S_2 e rilasciandolo, in uscita del circuito si ha il segnale pneumatico s che si mantiene fino a che non si pigia lo Stop S_t . Infatti, premuto il pulsante di questo, si interrompe il flusso di aria che perviene al pilotaggio della valvola V e questa, tornando nella posizione di riposo, interdice l'aria in uscita.

Per l'avvio di un ciclo si può impiegare il circuito proposto.

Così il ciclo $A^+ B^+ A^- B^-$ già realizzato con l'avvio comandato da un semplice Start si debba effettuare con la possibilità di ottenere un ciclo alla volta o una serie di cicli automatici fino alla interruzione comandata da uno Stop.

In questo caso cambia solamente il comando dello stato A^+ del cilindro A.

Indicando con S_1 lo start che comanda lo svolgimento di un ciclo alla volta, con S_2 quello per l'esecuzione di cicli automatici e con S_t lo Stop, il comando A^+ è dato da:

$$A^+ = (S_1 + m) \cdot b_0 \quad \text{dove } m = (S_2 + s) \cdot \overline{S_t}$$

Gli altri comandi rimangono invariati:

$$B^+ = a_1$$

$$A^- = b_1$$

$$B^- = a_0$$

Il circuito già precedentemente progettato viene così modificato:

fig.6.50

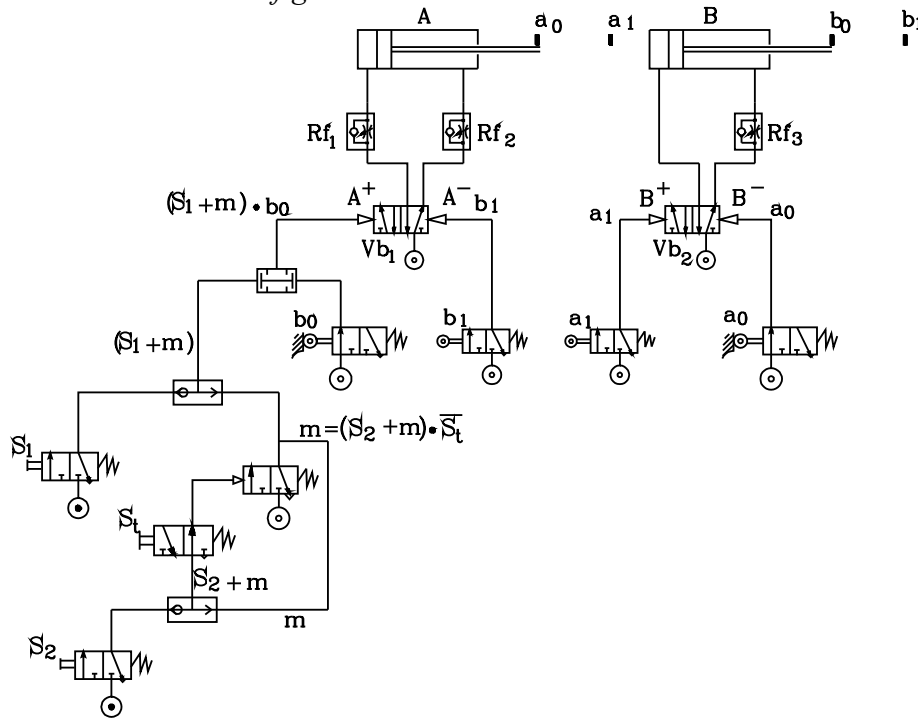
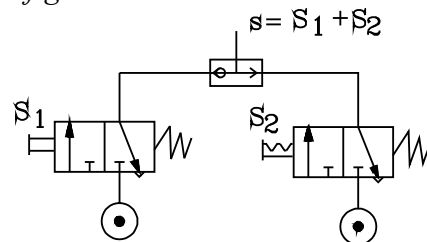


fig.6.51



Un modo più semplice di produrre un segnale continuo di comando, per ottenere la ripetizione automatica di più cicli, è quello di utilizzare uno Start, realizzato con una valvola 3/2 NC monostabile ad azionamento manuale e aggancio meccanico.

L'attivazione dello Start resta permanente per effetto dell'aggancio meccanico, questo può essere disinserito manualmente e la molla riporta la valvola in posizione di riposo, nella quale si ha l'interdizione dell'aria in uscita.

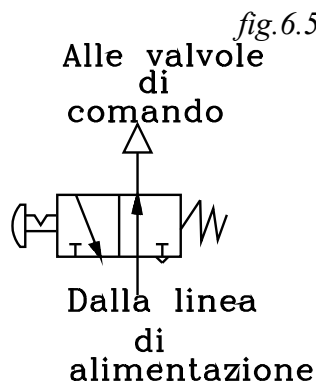
Utilizzando come avvio di un ciclo il circuito di figura, si può ottenere, pigiando e rilasciando S_1 un segnale impulsivo di comando di un solo ciclo; azionando S_2 questo si attiva e l'aggancio meccanico lo mantiene in posizione di attivazione, fino a che non viene disinserito. Durante l'inserzione di S_2 si ha la ripetizione automatica del ciclo.

6.4.1 Pulsante di emergenza

Nel funzionamento di un ciclo si possono presentare delle situazioni di emergenza che debbono essere affrontate e risolte in modi diversi, a seconda delle particolari esigenze del tipo di lavoro che si sta effettuando.

La valvola di Stop del circuito di memoria con prevalenza alla cancellazione può costituire una interruzione volontaria, ma, interrotto il segnale di avvio, il ciclo in atto continua ugualmente fino all'esecuzione dell'ultima fase.

Interruzione dell'alimentazione



Un dispositivo molto semplice di emergenza può essere costituito da un circuito che interrompe l'alimentazione delle valvole di comando.

L'interruzione può essere semplicemente effettuata con uno Stop costituito da una valvola 3/2 NA monostabile con azionamento manuale e aggancio meccanico. Usualmente l'azionamento è con pulsante a fungo di colore rosso che si aggancia meccanicamente.

Spinto il pulsante a fondo la valvola chiude il passaggio dell'aria verso l'uscita e l'aggancio, meccanico lo mantiene in posizione di chiusura. Per ripristinare il passaggio dell'aria occorre disinserire l'aggancio (*in genere con una manovra di rotazione del fungo*).

Segnale di emergenza con comando di determinate posizioni degli steli

In determinate lavorazioni occorre che nell'azionamento dell'emergenza, non solamente si deve fermare il ciclo ma gli steli dei cilindri debbano portarsi in determinate posizioni di sicurezza.

In questo caso occorre imporre particolari condizioni al contorno che, ovviamente si traducono in equazioni logiche le quali possono essere realizzate dall'appropriato circuito pneumatico.

IN LABORATORIO

Per esercizio si consideri il ciclo $A^+ B^+ B^- A^-$ realizzato con finecorsa unidirezionali nel quale si suppongano le seguenti condizioni al contorno:

- *Pigiando il pulsante S_1 si può effettuare un solo ciclo alla volta.*
- *Pigiando il pulsante S_2 si ottiene la ripetizione automatica del ciclo fino all'azionamento dello stesso o altro pulsante.*
- *Deve essere previsto un pulsante di emergenza con il quale si ferma il ciclo e gli steli dei cilindri si portino in posizione di rientro.*

Sono noti i segnali di comando degli stati dei cilindri senza le condizioni al contorno. L'imposizione di questi si traducono nella variazione delle equazioni logiche.

Rammentiamo le equazioni ottenute, ponendo come unica condizione al contorno l'avvio del ciclo con uno Start.

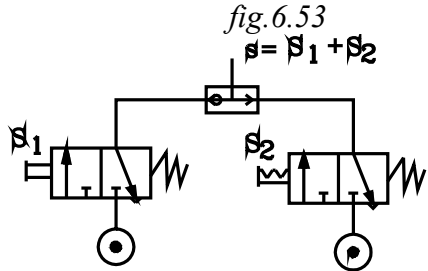
$$A^+ = a_0 \cdot \text{Start}$$

$$B^+ = a_1 \rightarrow$$

$$B^- = b_1$$

$$A^- = b_0 \rightarrow$$

Dove le frecce stanno ad indicare che i finecorsa sono di tipo unidirezionali.



Le prime due condizioni si realizzano, ponendo come segnale di avvio, quello che si ha all'uscita della valvola OR, i cui ingressi sono alimentati rispettivamente: da uno Start S_1 , costituito da una valvola 3/2 NC monostabile e da un altro Start S_2 , costituito da una valvola 3/2 NC monostabile con aggancio meccanico.

La prima equazione di comando dello stato A^+ viene così modificata:

$$A^+ = a_0 \cdot (S_1 + S_2)$$

Riguardo alla terza condizione al contorno occorre imporre:

1°- La fuoriuscita degli steli dei cilindri deve avvenire nella concomitanza delle due condizioni:

- che si verifichi l'attivazione del finecorsa che funge da variabile attiva ed attivante.
- che non sia pigiato il pulsante di emergenza.

2°- Il rientro degli steli deve avvenire quando si verifica o l'una o l'altra delle due condizioni:

- che si verifichi l'attivazione del finecorsa che funge da variabile attiva ed attivante.
- oppure che sia pigiato il pulsante di emergenza.

Da quanto esposto, il comando di fuoriuscita degli steli si ottiene con una operazione AND (*concomitanza*) tra il finecorsa che funge da variabile attiva (*di comando*) e il segnale di emergenza negato.

Indicando con "e" il segnale di emergenza si ottengono i seguenti segnali di comando di fuoriuscita degli steli, le equazioni di fuoriuscita sono:

$$A^+ = a_0 \cdot (S_1 + S_2) \cdot \bar{e}$$

$$B^+ = (a_1 \rightarrow) \cdot \bar{e}$$

Il comando di rientro degli steli si ottiene con una operazione OR. Con questa, il rientro può avvenire o con il segnale di finecorsa che funge da variabile attiva, **oppure** con il segnale di emergenza "e":

$$B^- = b_1 + e$$

$$A^- = (b_0 \rightarrow) + e$$

Dalle equazioni logiche si ricava lo schema del circuito pneumatico.

scopo didattico, nello schema si sono riportate le equazioni elaborate dai componenti pneumatici, fino ai comandi impartiti alle valvole bistabili di potenza.

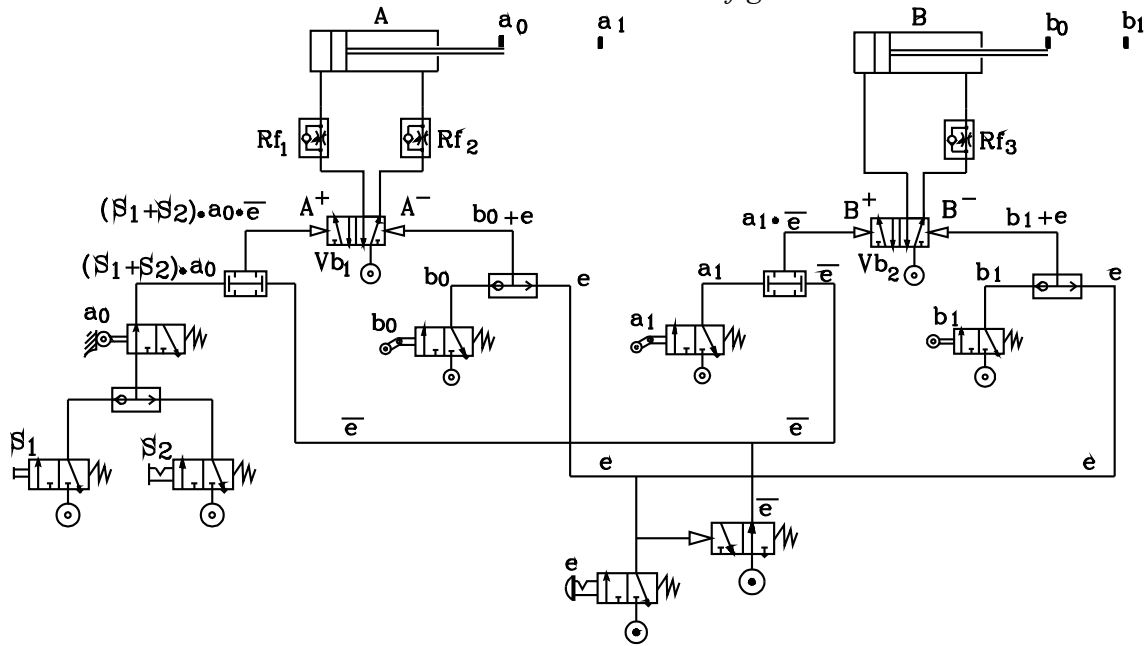
Osservazioni sulla stesura del circuito pneumatico

Conviene disegnare, dall'alto verso il basso, prima i cilindri con i regolatori di flusso e le valvole bistabili, quindi, al disotto di queste, i finecorsa che fungono da variabili attive, e posti in prossimità del pilotaggio da essi comandato.

Al disotto si disegni il circuito che elabora il segnale di emergenza in forma vera "e" ed in forma negata " \bar{e} ".

Detto circuito è costituito da una valvola 3/2 NC monostabile con comando manuale a fungo e aggancio meccanico. La sua uscita viene inviata alla linea di emergenza, contrassegnata con "e", e all'ingresso della valvola NOT, costituita da una valvola 3/2 NA monostabile.

fig.6.54



L'uscita dalla valvola NOT è collegata alla linea di emergenza negata, contrassegnata con "e". Ora, una alla volta, si traducano le equazioni logiche in circuito pneumatico.

$A^+ = a_0 \cdot (S_1 + S_2) \cdot \bar{e}$ Si disegni, al disotto di a_0 , lo start S_1 a pulsante e quello S_2 con aggancio meccanico, collegati ai due ingressi della valvola OR, ottenendo $(S_1 + S_2)$.

L'uscita della valvola OR si collega con l'ingresso del finacorsa a_0 , ottenendo $a_0 \cdot (S_1 + S_2)$.

Quest'ultimo segnale viene inviato ad un ingresso valvola AND, il cui altro ingresso viene collegato con la linea, che porta il segnale di emergenza negato "e".

Si ottiene così il segnale di comando: $A^+ = a_0 \cdot (S_1 + S_2) \cdot \bar{e}$

Questo viene inviato al pilotaggio A^+ della valvola bistabile Vb_1

$B^+ = (a_1 \rightarrow) \cdot \bar{e}$ Si collega l'uscita del finacorsa unidirezionale $(a_1 \rightarrow)$ con un ingresso della valvola AND il cui altro ingresso viene collegato alla linea che porta il segnale di emergenza negato "e". Si ottiene, così, il segnale di comando:

$$B^+ = (a_1 \rightarrow) \cdot \bar{e}$$

Questo viene inviato al pilotaggio B^+ della valvola bistabile Vb_2 .

$B^- = b_1 + e$ Si collega l'uscita del finacorsa b_1 con un ingresso della valvola OR, il cui altro ingresso viene collegato alla linea, che porta il segnale di emergenza negato "e". Si ottiene, così, il segnale di comando: $B^- = b_1 + e$

Questo viene inviato al pilotaggio B^- della valvola bistabile Vb_1 .

$A^- = (b_0 \rightarrow) + e$ Si collega l'uscita del finecorsa unidirezionale $(b_0 \rightarrow)$ con un ingresso della valvola OR il cui altro ingresso viene collegato alla linea, che porta il segnale di emergenza "e". Si ottiene, così, il segnale di comando: $A^- = (b_0 \rightarrow) + e$

Questo viene inviato al pilotaggio A^- della valvola bistabile Vb_1

Funzionamento

- Nel montaggio del circuito le valvole bistabili di potenza devono essere commutate in prima posizione, in modo che l'alimentazione P sia in comunicazione con l'orifizio che fornisce aria nella camera destra dei cilindri.
- Nelle condizioni iniziali, quando si alimenta l'impianto e il pulsante di emergenza è disattivato, la valvola NOT alimenta la linea \bar{e} e l'aria è disponibile ad uno degli ingressi delle valvole AND; inoltre le valvole di potenza bistabili alimentano le camere destre dei cilindri, ponendo gli steli in posizione rientrata.
- Azionando lo start S_1 si avvia un solo ciclo; mentre, azionando S_2 , si ha l'aggancio meccanico della valvola e si ottiene la ripetizione automatica dei cicli, fino a che non si manovra di nuovo l'azionamento della valvola, togliendo l'aggancio, disattivandola.
- Pigiato, così uno degli Start, l'aria a passa attraverso il finecorsa a_0 (toccato) e perviene ad uno degli ingressi della valvola AND, il cui altro ingresso è alimentato dalla linea \bar{e} .
- Dalla valvola AND esce l'aria che va al pilotaggio A^+ della valvola Vb_1 . la quale, commutandosi, alimenta la camera sinistra del cilindro A che compie la corsa A^+ di fuoriuscita dello stelo.
- Verso la fine della corsa A^+ la camma scorre su rullo del finecorsa unidirezionale a_1 generando, durante lo scorrimento, il segnale pneumatico che viene inviato ad uno degli ingressi della valvola AND, il cui altro ingresso è alimentato dalla linea \bar{e} .
- Dalla valvola AND esce l'aria che va al pilotaggio B^+ della valvola bistabile Vb_2 . Questa si commuta e invia l'aria alla camera sinistra del cilindro B che effettua la corsa B^+ di fuoriuscita dello stelo. Dopo che la camma ha sorpassato il rullo di a_1 cessa il segnale pneumatico di pilotaggio B^+ .
- Occorre notare che, nella fuoriuscita dello stelo di B, viene toccato dalla camma il finecorsa b_0 , ma nel senso che provoca la rotazione dello snodo s e non della leva che comanda il finecorsa, il quale non viene attivato.
- Alla fine della corsa B^+ viene toccato il finecorsa b_1 che invia l'aria ad uno degli ingressi della valvola OR. L'uscita di questa fornirà aria al pilotaggio B^- della valvola bistabile Vb_2 . Questa si commuta, essendo il pilotaggio B^+ disattivato, e l'aria viene inviata nella camera destra del cilindro B (quella sinistra è in scarico). Viene effettuata la corsa di ritorno B^- del rientro dello stelo.
- Verso la fine della corsa B^- la camma scorre su rullo del finecorsa unidirezionale b_0 generando, durante lo scorrimento, il segnale pneumatico che viene inviato ad uno degli ingressi della valvola OR e da questa al pilotaggio A^- della valvola bistabile Vb_1 . Questa si commuta e invia l'aria alla camera destra del cilindro A che effettua la corsa A^- di rientro dello stelo. Dopo che la camma ha sorpassato il rullo di b_0 cessa il segnale pneumatico di pilotaggio A^- .
- Occorre notare che, nel rientro di A, viene di nuovo toccato dalla camma il finecorsa a_1 , ma nel senso che provoca la rotazione dello snodo s e non della leva che comanda il finecorsa, il quale non viene attivato.
- Alla fine della corsa A^- viene toccato il finecorsa a_0 che invia l'aria al pilotaggio A^+ (A^- è disattivato). A questo punto se è stato pigiato per l'avvio lo Start S_1 termina il ciclo; se invece è stato azionato S_2 questo è ancora attivato dall'aggancio meccanico e inizia un altro ciclo.

Emergenza

- Pigiando il pulsante di emergenza e, in qualsiasi momento dello svolgimento del ciclo, si ottiene:
 1. l'interruzione del ciclo.

2. il rientro automatico degli steli.

- Infatti, pigiando il pulsante *e*, si commuta la valvola di emergenza *e*, da essa, esce l'aria che va sia ad alimentare la linea *e*, che al pilotaggio della valvola NOT. Questa viene commutata e si interdice l'alimentazione di aria sulla linea \bar{e} che va in scarico.
- In questa evenienza, qualunque siano le condizioni dei finacorsa, si ha che: uno degli ingressi delle valvole AND non è alimentato, mentre lo è almeno uno delle valvole OR.
- Ciò determina che, sicuramente, le valvole AND, poste sui pilotaggi di fuoriuscita stelo, sono in comunicazione con lo scarico e non forniscono così segnale di comando. Contemporaneamente le valvole OR, poste sui pilotaggi di rientro degli steli, forniscono aria, essendo uno degli ingressi in comunicazione con la linea *e*. Si ottiene così il rientro degli steli.

Temi proposti

1°

Tema

Su di una serie di pezzi deve essere effettuata una marchiatura. Ogni pezzo viene posto a mano su di un apposito portapezzi posto in posizione accessibile. Attivati due Start, il pezzo viene portato in una posizione protetta ove viene effettuata la marchiatura. Effettuata l'operazione, il portapezzo ritorna nella iniziale posizione accessibile ove viene tolto il pezzo marchiato e posto un altro da marchiare.

Specifiche:

Per sicurezza debbono essere adoperati due Start per impegnare entrambi le mani.

Le velocità di spostamento del pezzo debbono essere moderate.

Deve essere assicurata una pressione sul punzone per effettuare la marchiatura.

I due cilindri sono sistemati entro la protezione, per cui il portapezzo nella posizione in cui viene posto il pezzo da stampigliare e raccolto quello stampigliato, deve sporgere al difuori di detta protezione.

Effettuare uno schema di massima del sistema. Effettuare una appropriata relazione ove vengono trattati i seguenti punti essenziali: - Soluzione del problema proposto nel rispetto delle specifiche - Componenti utilizzati con relativa tabella - Diagramma Corsa passo - Tabella degli ordini e dei comandi - Individuazione dei segnali bloccanti - Eliminazione dei segnali bloccanti mediante finecorsa unidirezionali - Diagramma di flusso a barre orizzontali - Equazioni logiche - Circuito pneumatico -

Suggerimenti: Nelle condizioni iniziali il portapezzo è posto all'estremità dello stelo di un cilindro in posizione di fuoriuscita. In AND con il finecorsa che comanda il rientro del cilindro che comanda il punzone deve essere posto il segnale della valvola di sequenza, pilotata dalla pressione "p" che è nella camera di comando di fuoriuscita stelo...



2°

Tema

Su una serie di pezzi debbono essere effettuati in successione due punzonature comandate da due cilindri A e B. Posto un pezzo su di un apposito portapezzi e attivati due Start, viene effettuata la prima punzonatura dal cilindro A e successivamente la seconda dal cilindro B.

Specifiche:

Per sicurezza debbono essere adoperati due Start per impegnare entrambi le mani.

Deve essere previsto un dispositivo di emergenza che, impedisca la fuoriuscita degli steli e che in ogni caso li faccia rientrare.

Effettuare uno schema di massima del sistema. Effettuare una appropriata relazione ove vengono trattati i punti essenziali specificati nel tema precedente. Eliminare i segnali bloccanti mediante dei finecorsa unidirezionali.



3°

Tema

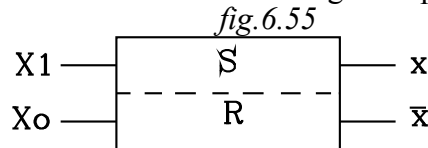
Risolvere il tema precedente (N° 2) eliminando i segnali bloccanti mediante l'uso dei temporizzatori.

6.5 TECNICA DELLE MEMORIE IN CASCATA

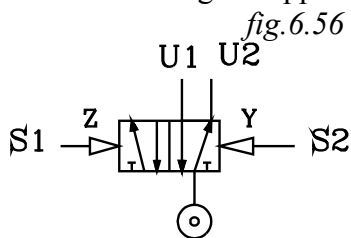
Memoria bistabile

Una memoria bistabile o *FLIP-FLOP* è un circuito o un componente che memorizza in modo stabile due condizioni di uscita.

- Un segnale di *Set* X_1 commuta la memoria, dando in uscita il segnale x .
- Un secondo segnale X_0 detto di *Reset* dà in uscita un segnale opposto \bar{x}



Una memoria pneumatica può essere costituita da una valvola bistabile 5/2, nella quale i segnali di *Set* e *Reset* vengono applicati ai pilotaggi e provocano in uscita due stati opposti.



Nelle condizioni iniziali sia alimentata la linea U_2 mentre U_1 è in scarico:

$$U_1 = 0 \quad U_2 = 1$$

Il segnale di *Set* S_1 , inviato al pilotaggio Z, commuta la valvola che inverte gli stati delle due uscite - L'aria esce da U_1 e

U_2 è in scarico:

$$U_1 = 1 \quad U_2 = 0$$

Il segnale pneumatico nella linea U_1 permane (è memorizzato) fino a che non si invia il segnale di resettaggio S_2 al pilotaggio Y

Il segnale di *Reset* S_2 , inviato al pilotaggio Y, commuta la valvola che inverte gli stati delle due uscite - L'aria esce da U_2 e U_1 è in scarico:

$$U_1 = 0 \quad U_2 = 1$$

Ovviamente il segnale pneumatico nella linea U_2 permane (è memorizzato) fino a che non si invia di nuovo il segnale di settaggio S_1 al pilotaggio Z

È evidente che in uscita si ha un segnale alla volta: quando vi è l'uno viene cancellato l'altro. Ogni segnale rimane *memorizzato* in una uscita, fino a che non viene inviato al pilotaggio della valvola un altro segnale che resettì detta uscita e setti l'altra

Si ha che:

- Al segnale di pilotaggio S_1 corrisponde il segnale pneumatico nell'uscita U_1 .

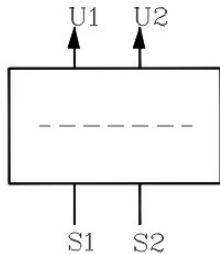
$$S_1 \xrightarrow{\text{corrisponde}} U_1$$

- Al segnale di pilotaggio S_2 corrisponde il segnale pneumatico nell'uscita U_2 .

$$S_2 \xrightarrow{\text{corrisponde}} U_2$$

- Applicando contemporaneamente i due segnali S_1, S_2 la valvola si blocca

fig.6.57



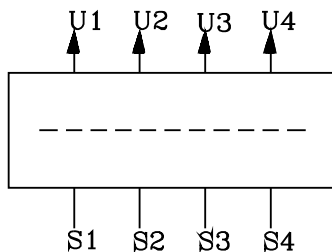
In generale, la memoria con due ingressi e due uscite viene rappresentata come in figura

Si desidera ora con più valvole bistabili 5/2 trovare un sistema capace di porre in uscita più segnali: uno alla volta, con cancellazione dei precedenti.

Le condizioni imposte sono:

- 1- Il numero dei segnali di ingresso debbono essere uguali a quelli di uscita.
- 2- Al segnale di ingresso S_i corrisponde l'unico e solo segnale di uscita U_i .
- 3- Il segnale di uscita U_i deve essere memorizzato e permanere anche quando termina il segnale di ingresso S_i che lo ha settato, e ciò fino a che non viene inviato il segnale di ingresso successivo S_{i+1} che l'annulla.
- 4- Vi deve essere in uscita, di volta in volta, un solo segnale; per cui gli altri devono essere annullati.
- 5- Un segnale non si ha nell'uscita U_i se non si è già verificata l'uscita che la precede nella sequenza U_{i-1} .
- 6- I segnali di ingresso sono efficaci solo se si presentano secondo l'ordine prestabilito.

fig.6.58



Un blocco di tale unità, in generale, viene indicato con la rappresentazione schematica di figura *fig.6.58*.

Agli n ingressi S_1, S_2, \dots corrispondono rispettivamente le n uscite U_1, U_2, \dots

Adoperando le valvole bistabili 5/2, un tale sistema si ottiene con un concatenamento detto *MEMORIE IN CASCATA*.

Una valvola bistabile può assumere due stati stabili diversi: fornisce quindi una memoria con due ingressi e due uscite.

Nelle memorie in cascata, ogni altra valvola che si aggiunge, fornisce solamente un altro ingresso e un'altra uscita.

Da quanto detto ne viene che, se n è il numero di ingressi e uscite del blocco di memorie da ottenere, per effettuarlo, occorre adoperare un numero di memorie m (*valvole bistabili*) pari al numero degli n di ingressi (uscite) meno uno:

$$m = n - 1 \quad (6.5.1)$$

Così, per ottenere un blocco di memorie con 5 ingressi e relative uscite, occorrono 4 memorie (*valvole bistabili*).

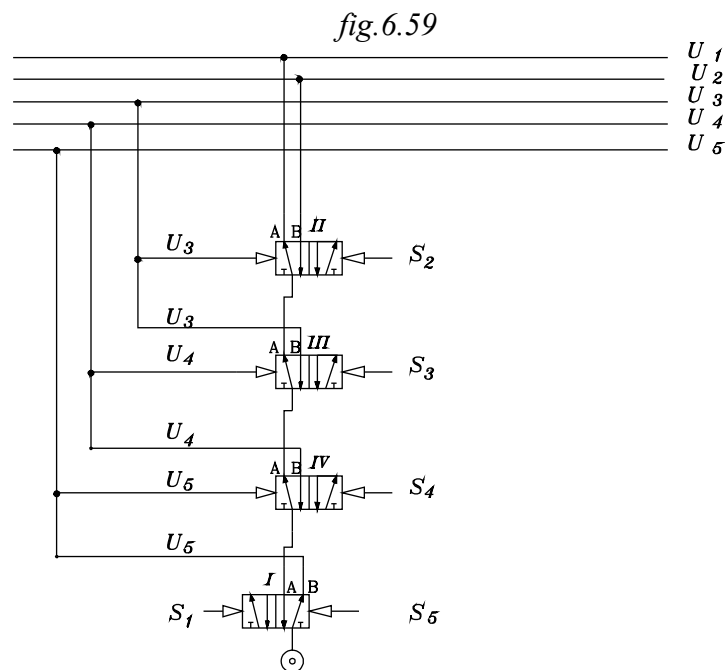
Si possono considerare due tipi di collegamenti.

Primo tipo

Per effettuare un esempio pratico si consideri un blocco di memorie con 5 ingressi e relative uscite. Occorrono $5-1=4$ memorie.

Nello schema si ponga la prima memoria "I" in basso (*vedi fig.6.59*). La seconda memoria "II" la si disegni, come ultima, in alto. Le successive: III, IV si pongano una di seguito all'altra, andando dall'alto verso il basso, fino a che l'ultima "IV" risulta adiacente alla prima "I".

Si pone la valvola I in 1° posizione; mentre tutte le altre in 2° posizione.



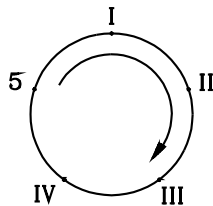
Considerando il punto 5 delle condizioni da rispettare: l'uscita U_1 si può verificare solamente se si è abilitata prima quella che la precede nel ciclo. Ciclicamente l'uscita U_1 è preceduta dall'ultima uscita U_5 .

Ne viene che, nelle condizioni iniziali, deve essere alimentata l'ultima uscita U_5 : quindi, l'uscita B della valvola bistabile, costituente la prima memoria "I", deve essere collegata con la linea U_5 .

Il collegamento procede poi nella maniera qui di seguito descritta.

- L'altra uscita A della prima valvola "I" va collegata con l'orifizio di alimentazione P della IV (che è in 2° posizione).
- Di seguito, l'uscita A della IV va all'orifizio P della III , e l'uscita A di quest'ultima va all'orifizio P della II .

fig.6.60



In ordine i segnali di SET siano : $S_1 S_2 S_3 S_4 S_5$

Si scrivano su una circonferenza, posti in senso orario, i numeri rappresentanti la sequenza di abilitazione delle 4 valvole. Il numero 5 sta per il segnale di settaggio della valvola I .

Considerato il senso orario, si vede che la valvola I segue la 4 - la 4 segue la 3 - la 3 segue la 2 .

Occorre che l'uscita di una valvola resettì la precedente e setti le relativa uscita

- L'uscita B della valvola I → deve resettare la IV
- L'uscita B della valvola IV → deve resettare la III
- L'uscita B della valvola III → deve resettare la II
- L'ultimo segnale di SET S_5 → deve resettare la valvola I

Per far ciò occorre che:

- l'uscita B dalla valvola I → va al pilotaggio Z della valvola IV e collegata alla U_5
- l'uscita B della valvola IV → va al pilotaggio Z della valvola III e collegata alla U_4
- l'uscita B della valvola III → va al pilotaggio Z della valvola II e collegata alla U_3

La valvola II ha l'uscita A collegata con la linea U_1 e l'uscita B collegata con la linea U_2 .

Nelle condizioni iniziali la valvola I è in prima posizione con l'uscita in B collegata con U_5 e A in scarico.

Tutte le altre valvole nella condizioni iniziali sono in seconda posizione: con l'uscita in A collegata con l'alimentazione P della valvola seguente, mentre B è in scarico.

I segnali di ingresso si susseguono nell'ordine $1,2,3,4,5$.

La condizione iniziale, considerando l'ordine ciclico, corrisponde all'assetto che si ottiene con l'ultimo segnale di settaggio S_5 . Ovvero:

- Nelle condizioni iniziali, la memoria I (valvola I) deve avere resettato la memoria precedente: la IV , che è l'ultima memoria nell'ordine ciclico.

- Inoltre, nelle condizioni iniziali, non può essere settata la prima uscita U_1 se non è stata precedentemente settata l'ultima U_5 .

Da queste due considerazioni risulta che:

L'uscita B della I va collegata al pilotaggio Z della IV e, contemporaneamente deve fornire il segnale U_5 sulla linea 5.

In ordine i segnali di ingresso determinano:

- S_1 Commuta la valvola I .
Viene alimentata l'uscita A della valvola I collegata all'alimentazione P della IV . Attraverso i collegamenti $A \rightarrow P$ delle successive valvole, dalla uscita A della valvola II , si ottiene il segnale pneumatico U_1 sulla linea 1 .
Viene annullato il segnale di pilotaggio Z sulla valvola IV , in quanto la B della I viene posto in scarico.
La linea 5 , attraverso l'orifizio B , viene posta in scarico.
- S_2 Commuta la valvola II .
Viene alimentata l'uscita B della valvola II che fornisce il segnale di uscita U_2 sulla linea 2 .
La linea 1 , attraverso l'orifizio A della valvola II viene posta in scarico.
- S_3 Commuta la valvola III .
Viene resettata la valvola precedente II , in quanto l'utilizzazione B della III , ora alimentata, comanda il pilotaggio Z della II .
Viene alimentata l'uscita di utilizzazione B che fornisce il segnale di uscita U_3 sulla linea 3 .
La linea 2 viene posta in scarico attraverso B della II .
- S_4 Commuta la valvola IV .
Viene resettata la valvola III , in quanto l'utilizzazione B della IV , ora alimentata, comanda il pilotaggio Z della III .
Viene alimentata l'uscita di utilizzazione B della IV che fornisce il segnale di uscita U_4 sulla linea 4 .
La linea 3 viene posta in scarico attraverso B della III .
- S_5 Commuta la valvola I .
Viene resettata la valvola IV in quanto l'utilizzazione B della valvola I , ora alimentata, comanda il pilotaggio Z della IV .
Viene alimentata l'uscita di utilizzazione B della I che fornisce il segnale di uscita U_5 sulla linea 5 .
La linea 4 viene posta in scarico attraverso la B della IV .

Nelle soluzioni di circuiti con segnali bloccanti utilizzeremo tale tipo di memorie in cascata. .

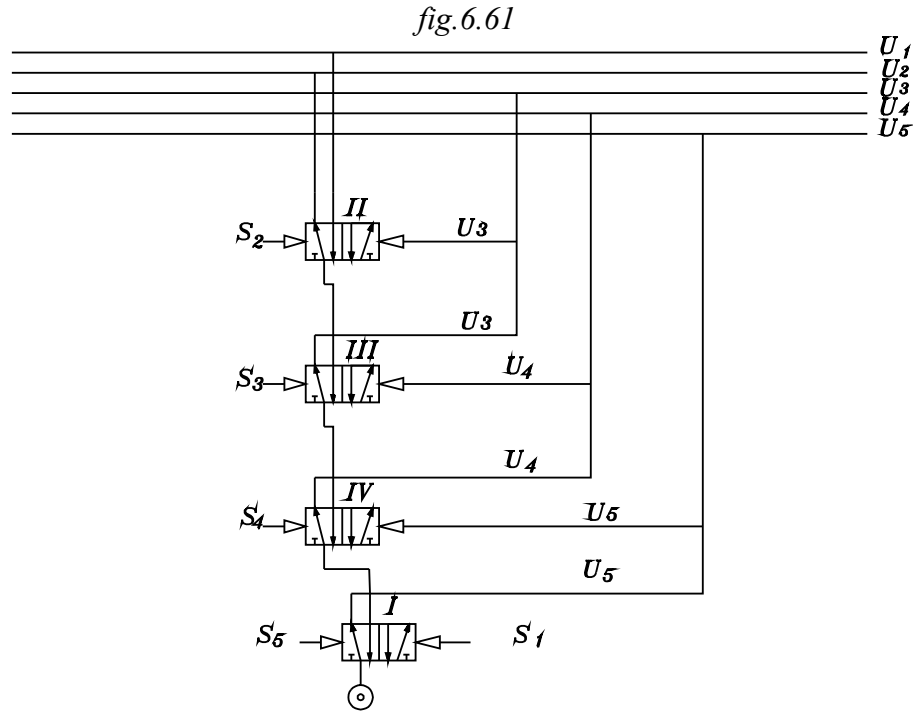
Vi è anche un altro modo di collegare le valvole per la cascata che determina lo stesso risultato, di cui si dà cenno.

Secondo tipo

La valvola I si pone in seconda posizione (*alimentato A e in scarico B*). Le altre valvole sono in prima posizione (*alimentato B e in scarico A*).

L'uscita B della valvola I è collegata all'ingresso P della IV , l'uscita B della IV è collegata all'ingresso della III e così di seguito per le altre valvole...

L'utilizzo *B* della valvola *II* fornisce il segnale di uscita *U₁* alla linea *I*. Mentre gli utilizzi *A* delle altre valvole, in ordine partendo dall'alto, forniscono i segnali di uscita *U₂*, *U₃*, *U₄*, *U₅*.



Inoltre il segnale di uscita *U₅* reseta la valvola *IV*, la *U₄* reseta la valvola *III*, la *U₃* reseta la valvola *II*.

Il funzionamento è analogo al precedente collegamento.

6.5.1 SOLUZIONE DI CICLI CON SEGNALI BLOCCANTI MEDIANTE L'USO DELLE MEMORIE IN CASCATA

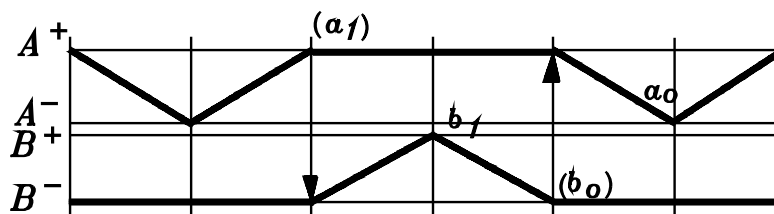
Per comprendere la soluzione del problema, si consideri il seguente ciclo contenente un *segnale bloccante*, già precedentemente risolto con altri metodi.

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

Si segua sempre, nello sviluppo degli argomenti, l'ordine adottato nei precedenti cicli, salvo poi adoperarne uno specifico per le memorie in cascata.

Diagramma corsa passo

fig.6.62



Osservando il diagramma corsa passo si nota:

- Il finecorsa a_1 rimane attivato per la doppia corsa del cilindro B comandato. Il finecorsa a_1 è un segnale bloccante.
- Il finecorsa b_0 rimane attivato per la doppia corsa del cilindro A comandato. Il finecorsa b_0 è un segnale bloccante.
- Gli altri finecorsa non permangono attivati nella doppia corsa dei cilindri comandato e quindi non sono bloccanti.

Tabella degli ordini

Si può scrivere la seguente tabella dei comandi

Comando	Stato comandato	Stato iniziale dei finecorsa	Stato raggiunto dai finecorsa
A_1	A^+	$a_0 \cdot b_0 \rightarrow$	$a_1 \cdot b_0$
B_1	B^+	$a_1 \cdot b_0 \rightarrow$	$a_1 \cdot b_1$
B_0	B^-	$a_1 \cdot b_1 \rightarrow$	$a_1 \cdot b_0$
A_0	A^-	$a_1 \cdot b_0 \rightarrow$	$a_0 \cdot b_0$

Equazioni logiche complete e con variabili attive

Si supponga che l'unica condizione al contorno sia data dallo Start di inizio ciclo.

Si ottengono le equazioni:

Equazioni logiche complete	Equazioni logiche con variabili attive
$A^+ = a_0 \cdot b_0 \cdot Start$	$A^+ = a_0 \cdot Start$
$B^+ = a_1 \cdot b_0 \quad *$	$B^+ = a_1 \quad *$
$B^- = a_1 \cdot b_1$	$B^- = b_1$
$A^- = a_1 \cdot b_0 \quad *$	$A^- = b_0 \quad *$

Rammentiamo che le condizioni dei finecorsa all'inizio del passo, poste in *AND* rappresentano, insieme alle condizioni al contorno le equazioni logiche dello stato comandato: Così, quando viene comandato A^+ sono toccati a_0 , b_0 e, per iniziare il ciclo occorre pigiare lo Start:

$$A^+ = a_0 \cdot b_0 \cdot Start$$

La variabile attiva di comando di una stato è quella nuova che compare dallo stato precedente a quello attualmente comandato.

Così, nella sequenza ciclica, lo stato A^- precede A^+ e, dal primo al secondo la variabile nuova che diviene attiva è a_0 (b_0 rimane invariata): ne viene che la variabile attiva di A^+ è a_0 , insieme allo Start di inizio ciclo:

$$A^+ = a_0 \cdot Start$$

Con asterisco sono indicati i segnali bloccanti.

L'argomento è già stato precedentemente trattato; comunque, si ribadisce qui che dalla osservazione delle equazioni logiche complete risulta evidente se la variabile attiva sia bloccante o meno.

Il finecorsa a_1 è attivo sia quando comanda la corsa B^+ ($B^+ = a_1 \cdot b_0$), che quando viene comandata, da b_1 , la corsa opposta B^- ($B^- = a_1 \cdot b_1$).

Il segnale di finecorsa a_1 è bloccante.

- Il finecorsa b_0 è attivo sia quando comanda la corsa A^- ($A^- = a_1 \cdot b_0$), che quando viene comandata, da a_0 , la corsa opposta A^+ ($A^+ = a_0 \cdot b_0 \cdot Start$).

Il segnale di finecorsa b_0 è bloccante.

Nelle equazioni di comando vi è, in due fasi diverse, la ripetizione della stessa combinazione in AND dei segnali di finecorsa.

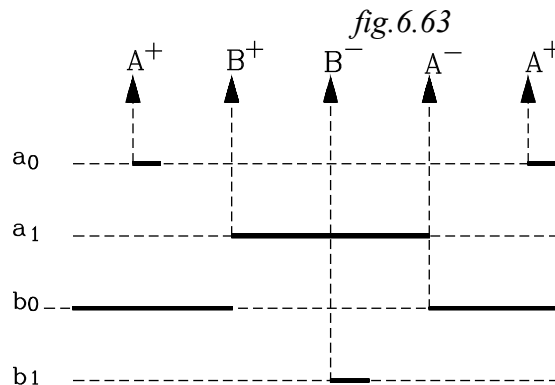
$$\begin{cases} B^+ = a_1 \cdot b_0 \\ A^- = a_1 \cdot b_0 \end{cases}$$

Diagramma di flusso a barre orizzontali

Dal diagramma di flusso si rileva:

Fincorsa a_0 Comanda lo stato A^+

Viene toccato dalla camma nella corsa di ritorno dello stelo di A e abbandonato subito dopo nella corsa di fuoriuscita.



Fincorsa a_1 Comanda lo stato B^+ .

Viene toccato dalla camma nella corsa di fuoriuscita dello stelo di A e permane attivato fino a che non viene comandato il rientro A^- .

Il finecorsa a_1 è attivo sia quando comanda corsa B^+ che quando viene comandata quella opposta B^- .

Il segnale di finecorsa a_1 è bloccante: quando nella valvola bistabile, che comanda il cilindro B, viene inviato al pilotaggio il segnale di commutazione B^- sull'altro pilotaggio è ancora attivo quello di commutazione B^+ .

Finecorsa b_1 Comanda lo stato B^- .

Viene toccato dalla camma nella corsa di fuoriuscita dello stelo di B e abbandonato subito dopo nella corsa di rientro.

Finecorsa b_0 Comanda lo stato A^- .

Viene toccato dalla camma nella corsa di rientro dello stelo di B e permane attivato fino a che non viene comandato la fuoriuscita B^+

Il finecorsa b_0 è attivo sia nel comando della corsa A^- che in quella opposta A^+ .

Il segnale di finecorsa b_0 è bloccante: quando nella valvola bistabile, che comanda il cilindro A, viene inviato al pilotaggio il segnale di commutazione A^+ sull'altro pilotaggio è ancora attivo quello di commutazione A^- .

Non si può togliere il segnale bloccante a_1 , che comanda B^+ , ponendo in serie ad esso il segnale di finecorsa b_0 ; in quanto, sì, questo non è presente nel comando B^- , ma risulta attivo insieme ad a_1 nella fase A^- per far rientrare A: il ciclo si scombina e si blocca. Lo stesso vale per il segnale bloccante b_0 .

Nel ciclo non solamente vi sono due segnali bloccanti ma anche identici in tutta la combinazione in AND dei finecorsa.

Per poter distinguere i due segnali identici, che debbono effettuare comandi diversi in fasi diverse, occorre introdurre un altro segnale U dato da una memoria. Questa, come si è detto, fornisce in uscita due segnali: $x = U_1$ e $\bar{x} = U_2$, che possono essere posti in AND con i due segnali identici e poterli così differenziare.

Così il primo segnale di memoria U_1 viene settato, memorizzato, e messo in AND con il primo segnale bloccante ($a_1 \cdot b_0$); esso, poi, deve essere resettato prima che si presenti lo stesso segnale bloccante ($a_1 \cdot b_0$), che dovrà, invece, avere in AND l'altro segnale di memoria U_2 . Questo, poi, a sua volta deve essere resettato prima che termini il ciclo.

I due segnali bloccanti che ora si verificano in tempi diversi differiscono per il diverso segnale di memoria. e risulterà:

$$B^+ = a_1 \cdot b_0 \cdot U_1$$

$$A^- = a_1 \cdot b_0 \cdot U_2$$

Prima di affrontare il metodo delle memorie in cascata, conviene riferirsi ad esempi di cicli contenenti uno o più segnali bloccanti con combinazione identica dei segnali di finecorsa in fasi diverse.

In detti cicli si vuole esaminare la necessità di introdurre segnali di memoria capaci di differenziare la stessa combinazione in AND dei finecorsa che in fasi diverse dovrebbero effettuare comandi diversi

Ad ogni fase i^{ma} in cui si presenta la stesa combinazione in AND dei finecorsa, si associa in AND un segnale di memoria U_j diverso.

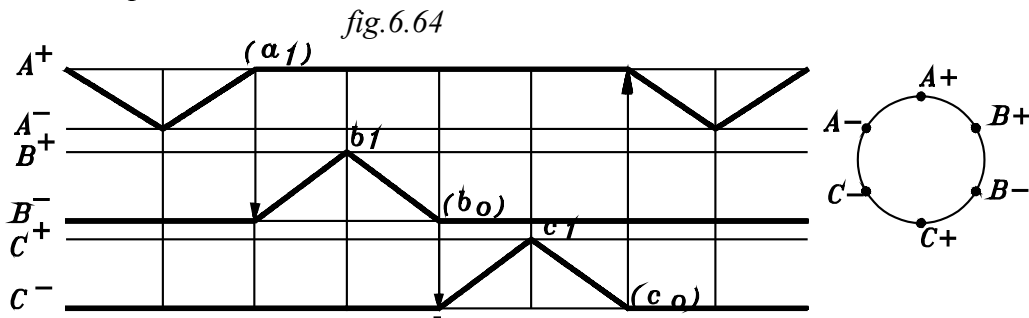
In tal modo i segnali di comando differiscono almeno per il segnale di memoria.

Consideriamo un altro ciclo con tre cilindri, ove si presentano segnali bloccanti con ripetizione in fasi diverse della stessa combinazione in *AND* dei finecorsa.

Studiamo la necessità di inserire memorie da porre in *AND* ai segnali uguali in modo da differenziarli nelle diverse fasi.

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

Diagramma corsa passo



Conviene porre il ciclo su una circonferenza la cui sequenza è ordinata in senso orario.

Come al solito il comando di uno stato viene effettuato quando si è verificato lo stato precedente. Così A^+ viene comandato dal verificarsi dello stato precedente A^- e ciò avviene quando contemporaneamente i due finecorsa a_0 , b_0 , c_0 sono entrambi toccati:

$$A^+ = a_0 \cdot b_0 \cdot c_0$$

Ecc.

Si può stendere la tabelle degli ordini e dei comandi.

Tabella degli ordini

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^-	$a_1 \cdot b_0 \cdot c_0 \rightarrow$	$a_0 \cdot b_0 \cdot c_0$

Equazioni logiche complete e con variabili attive

Equazioni logiche complete	Equazioni logiche con variabili attive
$A^+ = a_0 \cdot b_0 \cdot c_0 \cdot Start$	$A^+ = a_0 \cdot Start$
$B^+ = a_1 \cdot b_0 \cdot c_0 \quad *$	$B^+ = a_1 \quad *$
$B^- = a_1 \cdot b_1 \cdot c_0$	$B^- = b_1$
$C^+ = a_1 \cdot b_0 \cdot c_0 \quad *$	$C^+ = b_0 \quad *$
$C^- = a_1 \cdot b_0 \cdot c_1$	$C^- = c_1$
$A^- = a_1 \cdot b_0 \cdot c_0 \quad *$	$A^- = c_0 \quad *$

Nelle equazioni logiche complete compaiono tre segnali identici che in fasi diverse debbono effettuare comandi differenti.

$$\text{Combinazione identica dei finecorsa} \begin{cases} B^+ = a_1 \cdot b_0 \cdot c_0 & \text{segnale bloccante} \\ C^+ = a_1 \cdot b_0 \cdot c_0 & \text{segnale bloccante} \\ A^- = a_1 \cdot b_0 \cdot c_0 & \text{segnale bloccante} \end{cases}$$

La introduzione delle variabili attive minimizza i segnali necessari per effettuare il comando. Ciò risulta indispensabile nel progetto di circuiti cablati.

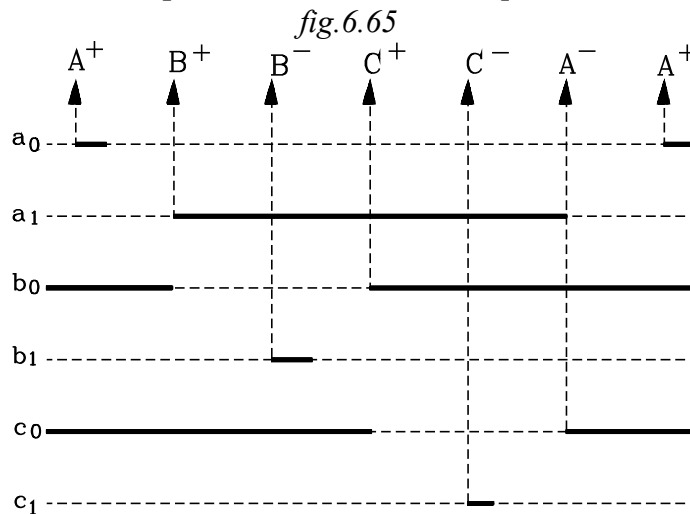
Osservando le equazioni complete si può notare che sono bloccanti le variabili attive a_1 , b_0 , c_0 . Esse permangono per la doppia corsa del cilindro comandato.

Ciò risulta molto evidente analizzando il diagramma di flusso a barre orizzontali di *fig.6.65*.

Finecorsa a₁ Comanda B^+ e permane attivato anche quando viene comandato B^- .

Finecorsa b₀ Comanda C^+ e permane attivato anche quando viene comandato C^- .

Finecorsa c₀ Comanda A^- e permane attivato anche quando viene comandato A^+ .



Come si è detto, nel caso che si verificano segnali bloccanti e identici nella combinazione in AND dei finacorsa, occorre introdurre i segnali di memoria che contraddistinguano i bloccanti che risulterebbero identici nei diversi passi del ciclo.

Una memoria fornisce due segnali diversi $x = U_1$ e $\bar{x} = U_2$ che possono contraddistinguere due segnali bloccanti; per ogni altro segnale bloccante che si presenta nel ciclo occorre introdurre una memoria

Per minimizzare il numero di memorie occorrenti, si suddivide la sequenza letterale dei comandi in gruppi, in modo che in ogni gruppo vi siano presenti solamente comandi di singole corse: non compaiono doppie corse di un cilindro

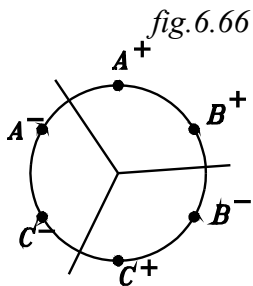
Con tale suddivisione si è certi che in ogni gruppo, non comparendo doppie corse di un cilindro, non vi siano segnali bloccanti: ogni combinazione dei segnali dei finecorsa del gruppo è diverso dall'altro. *I segnali identici compaiono in gruppi diversi.*

Ad ogni gruppo si associa un segnale di memoria diverso da porre in AND con i singoli segnali del gruppo. In tal modo i segnali bloccanti che compaiono, non nello stesso gruppo, ma in gruppi diversi, verranno contraddistinti da un differente segnale di memoria.

Nel corrente esempio si suddivide la sequenza in gruppi in modo che non comprendano sia la fuoriuscita che il rientro di un cilindro.

Si ottengono i tre gruppi:

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



Nei tre gruppi non vi compaiono: sia la corsa di andata che quella di ritorno dei cilindri.

Occorrono tre segnali di memoria U_1 U_2 U_3 da associare ciascuno ad un gruppo diverso.

$$\text{Associato} \rightarrow U_1 \begin{cases} A^+ = a_0 \cdot b_0 \cdot c_0 \\ B^+ = a_1 \cdot b_0 \cdot c_0 \end{cases} \rightarrow B^+ = a_1 \cdot b_0 \cdot c_0 \cdot U_1$$

$$\text{Associato} \rightarrow U_2 \begin{cases} B^- = a_1 \cdot b_1 \cdot c_0 \\ C^+ = a_1 \cdot b_0 \cdot c_0 \end{cases} \rightarrow C^+ = a_1 \cdot b_0 \cdot c_0 \cdot U_2$$

$$\text{Associato} \rightarrow U_3 \begin{cases} C^- = a_1 \cdot b_0 \cdot c_1 \\ A^- = a_1 \cdot b_0 \cdot c_0 \end{cases} \rightarrow A^- = a_1 \cdot b_0 \cdot c_0 \cdot U_3$$

Così, con l'introduzione dei tre segnali di memoria U_1 U_2 U_3 , i comandi degli stati B^+ C^+ A^- non sono più espressi dalla stessa combinazione in AND dei segnali di finecorsa, ma differiscono per il segnale di memoria rispettivamente associato

$$\begin{cases} B^+ = a_1 \cdot b_0 \cdot c_0 \cdot U_1 \\ C^+ = a_1 \cdot b_0 \cdot c_0 \cdot U_2 \\ A^- = a_1 \cdot b_0 \cdot c_0 \cdot U_3 \end{cases}$$

Utilizzando per i comandi gli AND tra le memorie e le sole variabili attive, quelle bloccanti diventano attivanti

$$\begin{cases} B^+ = a_1 \cdot U_1 \\ C^+ = b_0 \cdot U_2 \\ A^- = c_0 \cdot U_3 \end{cases}$$

Infatti, si consideri il comando dello stato B^+

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

Quando viene comandato il rientro B^- è ancora attivo il segnale di finacorsa a_1 (vedi diagramma di flusso), ma non lo è più il segnale di memoria U_1 (osservare i gruppi); per cui il pilotaggio B^+ della valvola bistabile non è più attivato e può essere impartito al pilotaggio B^- il comando di commutazione di rientro di B:

$$B^- = b_1$$

- * -

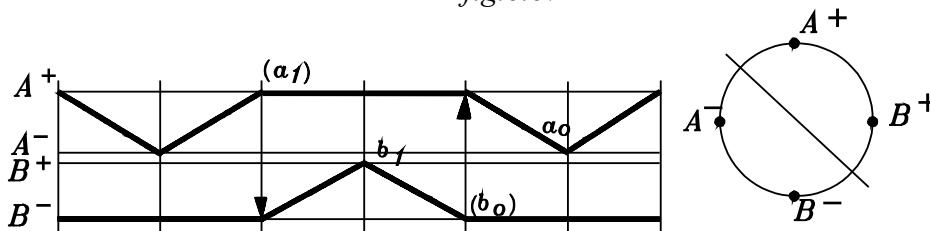
Consideriamo di nuovo il ciclo già trattato: $A^+ B^+ B^- A^-$ nel quale si sono individuati i due segnali bloccanti a_1, b_0 , con ripetizione della combinazione in AND dei segnali di finacorsa

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

Si possono suddividere le fasi del ciclo in due gruppi, che non contengano sia la corsa di andata che quella di ritorno di uno stesso cilindro (doppie corse del cilindro)

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

fig.6.67



$$\text{Associato } U_1 \rightarrow \begin{cases} A^+ = a_0 \cdot b_0 \\ B^+ = a_1 \cdot b_0 \end{cases} \rightarrow B^+ = a_1 \cdot b_0 \cdot U_1$$

$$\text{Associato } U_2 \rightarrow \begin{cases} B^- = a_1 \cdot b_1 \\ A^- = a_1 \cdot b_0 \end{cases} \rightarrow A^- = a_1 \cdot b_0 \cdot U_2$$

Il segnale $a_1 \cdot b_0$ si ripete in due fasi . In tempi diversi, una volta deve far fuoriuscire B , successivamente deve far rientrare A .

Il segnale si differenzia nei due momenti ponendo in AND con esso, prima il segnale di memoria U_1 e poi U_2 , ottenendo:

$$\begin{cases} B^+ = a_1 \cdot b_0 \cdot U_1 \\ A^- = a_1 \cdot b_0 \cdot U_2 \end{cases}$$

I comandi con le variabili attive:

$$\begin{cases} B^+ = a_1 \cdot U_1 \\ A^- = b_0 \cdot U_2 \end{cases}$$

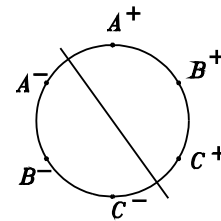
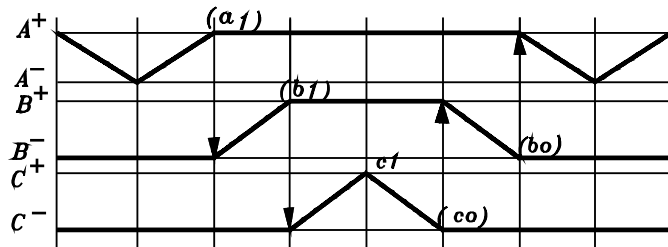
- * -

Come altro esempio consideriamo il ciclo $A^+ B^+ C^+ C^- B^- A^-$

Si può suddividere le fasi del ciclo in due gruppi, che non contengano sia la corsa di andata che quella di ritorno di uno stesso cilindro (*doppie corse del cilindro*)

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

fig.6.68



$$\text{Associato} \rightarrow U_1 \begin{cases} A^+ = a_0 \cdot b_0 \cdot c_0 \\ B^+ = a_1 \cdot b_0 \cdot c_0 \rightarrow B^+ = a_1 \cdot b_0 \cdot c_0 \cdot U_1 \\ C^+ = a_1 \cdot b_1 \cdot c_0 \rightarrow C^+ = a_1 \cdot b_1 \cdot c_0 \cdot U_1 \end{cases}$$

$$\text{Associato} \rightarrow U_2 \begin{cases} C^- = a_1 \cdot b_1 \cdot c_1 \\ B^- = a_1 \cdot b_1 \cdot c_0 \rightarrow B^- = a_1 \cdot b_1 \cdot c_0 \cdot U_2 \\ A^- = a_1 \cdot b_0 \cdot c_0 \rightarrow A^- = a_1 \cdot b_0 \cdot c_0 \cdot U_2 \end{cases}$$

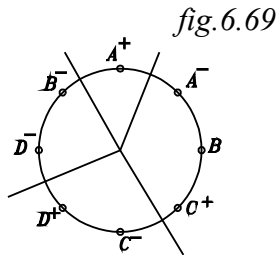
Con l'introduzione di due segnali $U_1 U_2$ vengono distinti i quattro segnali a due a due uguali. Basta l'introduzione di una memoria che fornisce le due uscite $U_1 U_2$.

6.5.1.1 Riduzione del numero di gruppi

La sequenza dei comandi va intesa come ciclica; conviene riportarla su una circonferenza. Risulta indifferente, nella suddivisione in gruppi, il punto di inizio dei raggruppamenti.

Consideriamo così la seguente sequenza:

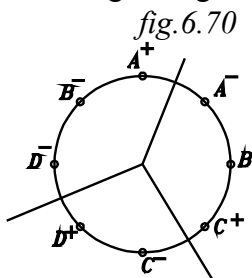
$$A^+ A^- B^+ C^+ C^- D^+ D^- B^-$$



Riportiamo la sequenza su di una circonferenza si ottiene:
Se la suddivisione in gruppi, non contenenti doppie corse dei cilindri, si inizia da A^+ , si ottengono i quattro gruppi rappresentati in figura .

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

Occorre in tal caso l'impiego di tre memorie in cascata che forniscano quattro uscite da porre in AND con i segnali uguali.



La sequenza è ciclica ; si può quindi iniziare la suddivisione in gruppi partendo da una qualsiasi fase del ciclo.

Come si può osservare dalla figura, se si inizia dalla fase A^- la suddivisione in gruppi non contenenti doppie corse, *si ottengono tre gruppi* invece di quattro.

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

Occorreranno in tal caso solamente due memorie in cascata, per fornire i tre segnali di memoria da porre in AND ai segnali uguali.

6.5.1.2 Settaggio e resettaggio del segnale di memoria in cascata

Nella suddivisione in gruppi non contenenti doppie corse, come si è detto, si associa ad ogni gruppo un segnale di memoria fornito dalle memorie in cascata .

Le memorie sono settate e resettate dai fine-corsa.

Una memoria si setta con il segnale di finecorsa della prima fase del gruppo e viene resettata, alla fine, quando viene settata la memoria del gruppo successivo.

6.5.2 METODO PRATICO PER LA PROGETTAZIONE DI CIRCUITI SEQUENZIALI CON L'UTILIZZO DELLE MEMORIE IN CASCATA

Per la comprensione riferiamoci ad un esempio pratico.

Sia da attuare il ciclo già precedentemente analizzato, per la determinazione delle memorie, occorrenti a trasformare i segnali bloccanti in attivanti :

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

Ora si vuole descrivere un metodo razionale, per poter settare e resettare le memorie occorrenti nel momento opportuno e porle in AND, quando necessita, con le variabili attive, in modo da togliere i segnali bloccanti.

Per la progettazione del circuito pneumatico e la stesura della relativa relazione si proceda nella seguente maniera:

- 1°- Data la sequenza del ciclo, effettuare il diagramma *corsa-passo*, mettendo in rilievo le condizioni al contorno.
- 2°- Effettuare la *tabella dei comandi* e degli stati ordinati
- 3°- Scrivere per esteso le *equazioni logiche*, date dalla combinazione dei finecorsa toccati, in AND con le condizioni al contorno. In corrispondenza scrivere le equazioni con le variabili attive
- 4°- Effettuare il *diagramma di flusso a barre orizzontali*, ponendo in rilievo solamente i comandi, dati dai segnali di finecorsa che costituiscono variabili attive
- 5°- Dalle equazioni logiche complete o dal diagramma di flusso si possono porre in rilievo le *variabili attive bloccanti*, ricordando che:
Un segnale bloccante è quella variabile attiva, che comanda la corsa di un cilindro e si prolunga nello stato attivo fino a comprendere una doppia corsa (ed oltre) di detto cilindro.
- 6°- Porre la sequenza letterale su una circonferenza e dividerla in *gruppi non contenenti corse opposte* di uno stesso cilindro. Iniziare la suddivisione da una lettera che rende minimo il numero di gruppi.
- 7°- Associare ad ogni gruppo un *segnale di memoria*.
- 8°- Effettuare la *tabella (matrice) dei segnali*, come verrà spiegato nell'esempio, nel punto corrispondente.
- 9°- Scrivere le *equazioni logiche* di settaggio delle memorie e di comando degli stati dei cilindri.
- 10°- Effettuare il *diagramma di flusso a barre orizzontali* completo del settaggio e resettaggio delle memorie.
- 11°- Effettuare lo *schema del circuito pneumatico*.



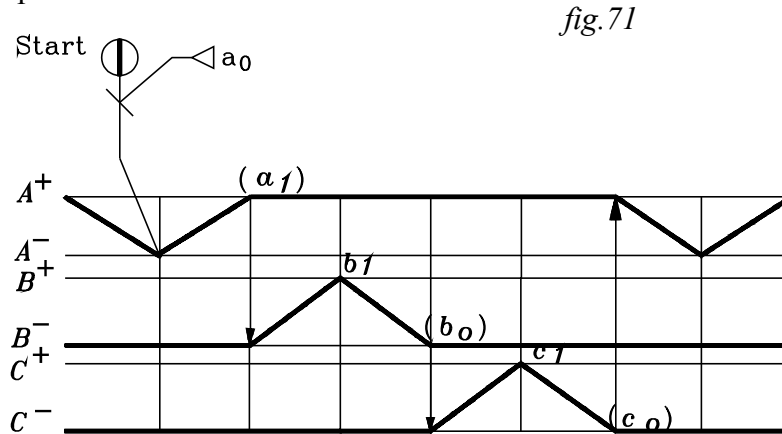
Per spiegare il metodo conviene riferirsi ad una applicazione pratica, da realizzare poi in laboratorio.

CICLO A⁺ B⁺ B⁻ C⁺ C⁻ A⁻

La realizzazione del ciclo si effettui sviluppando gli 11 punti proposti

1°- *Data la sequenza del ciclo, effettuare il diagramma corsa-passo, mettendo in rilievo le condizioni al contorno.*

Il diagramma corsa passo è:



La condizione al contorno è data dallo *Start* da porre in *AND* con la variabile attiva che comanda A^+ .

2°- *Effettuare la tabella degli ordini e degli stati comandati.*

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^-	$a_1 \cdot b_0 \cdot c_0 \rightarrow$	$a_0 \cdot b_0 \cdot c_0$

3°- *Scrivere per esteso le equazioni logiche, date dalla combinazione dei finecorsa toccati all'inizio della fase (fine della fase precedente)
In corrispondenza scrivere le equazioni con le variabili attive: le nuove che compaiono nella fase rispetto alla precedente.*

Equazioni logiche complete e con variabili attive

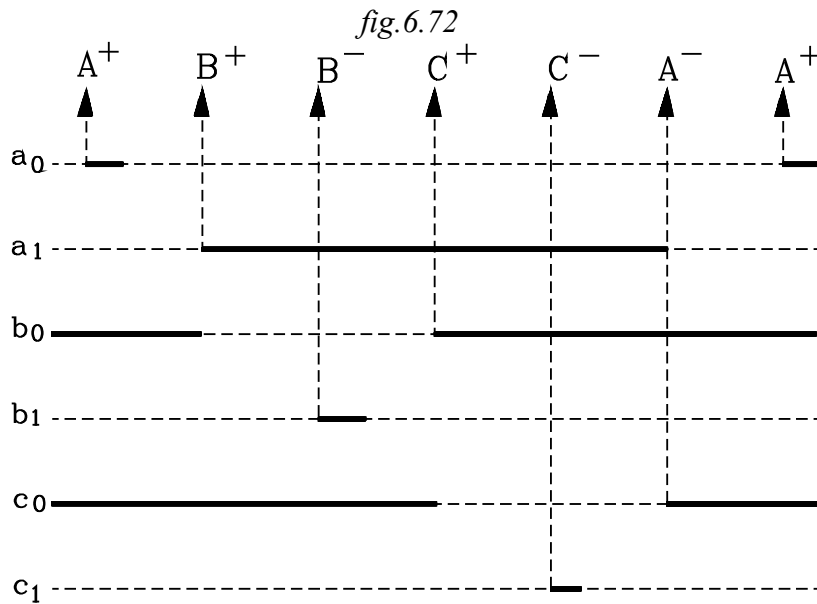
Equazioni logiche complete	Equazioni logiche con variabili attive
$A^+ = a_0 \cdot b_0 \cdot c_0$	$A^+ = a_0$
$B^+ = a_1 \cdot b_0 \cdot c_0$ *	$B^+ = a_1$ *
$B^- = a_1 \cdot b_1 \cdot c_0$	$B^- = b_1$
$C^+ = a_1 \cdot b_0 \cdot c_0$ *	$C^+ = b_0$ *
$C^- = a_1 \cdot b_0 \cdot c_1$	$C^- = c_1$
$A^- = a_1 \cdot b_0 \cdot c_0$ *	$A^- = c_0$ *

Nel ciclo vi sono ripetizioni della stessa combinazione in *AND* dei segnali di finecorsa nei tre comandi B^+ C^+ A^- .

$$\begin{cases} B^+ = a_1 \cdot b_0 \cdot c_0 \\ C^+ = a_1 \cdot b_0 \cdot c_0 \\ A^- = a_1 \cdot b_0 \cdot c_0 \end{cases}$$

4°- *Diagramma di flusso a barre orizzontali*

È stato già precedentemente descritto e sono stati rilevati i segnali bloccanti.



5°- *Dalle equazioni logiche complete o dal diagramma di flusso si individuano i segnali bloccanti: quelli che permangono attivi sia quando comandano una corsa che quella opposta.*

Finecorsa a_1 Comanda B^+ e permane attivato anche quando viene comandato B^- .

Il segnale di finecorsa a_1 è bloccante.

Finecorsa b_0 Comanda C^+ e permane attivato anche quando viene comandato C^- .

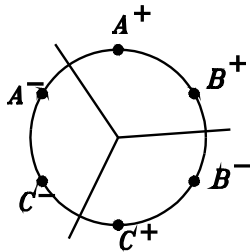
Il segnale di finecorsa b_0 è bloccante.

Finecorsa c_0 Comanda A^- e permane attivato anche quando viene comandato A^+ .

Il segnale di finecorsa c_0 è bloccante.

6°- *Porre la sequenza letterale su una circonferenza e dividerla in gruppi non contenenti corse opposte. Iniziare la suddivisione da una lettera che rende minimo il numero di gruppi.*

fig.6.73



Iniziando dalla lettera A^+ , la prima inversione si ha dopo B^+ ; la seconda inversione si ha dopo la lettera C^+ .

La suddivisione in gruppi, partendo dalla lettera A^+ è:

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

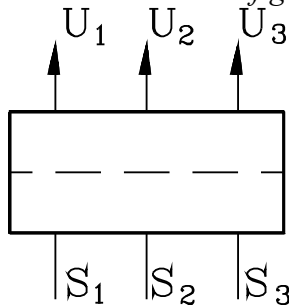
Ciascun gruppo non contiene doppie corse di uno stesso cilindro.

Corse opposte dello stesso cilindro si trovano in gruppi diversi. Così:

- A^+ si trova nel primo gruppo mentre A^- si trova nel terzo gruppo.
- B^+ è nel primo gruppo e B^- nel secondo.
- C^+ è nel secondo gruppo e C^- nel terzo.

7°- *Associare ad ogni gruppo un segnale di uscita dalle memorie in cascata*

fig.6.74



I segnali bloccanti si trovano in tre gruppi diversi. Per contraddistinguerli si associa ad ogni gruppo un segnale di memoria. Questi vengono fornite da due memorie in cascata, che forniscono tre uscite: $U_1 U_2 U_3$, rispettivamente settate da $S_1 S_2 S_3$.

Per l'impiego delle memorie in cascata si debbono applicare i seguenti criteri.

a- *Le memorie in cascata debbono fornire $n=3$ segnali di memoria pari al numero dei gruppi.*

b- *Si impiegano $m=2$ memorie costituite da valvole bistabili: il numero di memorie è pari a quello dei segnali di uscita meno uno: $m=3-1=2$.*

c- *Il numero dei segnali di uscita dalle memorie in cascata è pari al numero dei segnali di memoria da associare ai gruppi: pari cioè al numero dei gruppi stessi - $U_1 U_2 U_3$.*

d- Ad ogni segnale di uscita dalle memorie in cascata vi deve corrispondere un segnale di settaggio di ingresso. Occorrono, nel caso in esame, 3 segnali di settaggio: $S_1 S_2 S_3$.

8- Si effettua la seguente matrice dei segnali suddivisi per gruppi.

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

Sulla prima riga superiore si pongono gli stati comandati : $A^+ B^+ B^- C^+ C^- A^-$.

Sulla seconda riga si scrive, in prossimità di ogni stato comandato, la relativa variabile attiva.

Così essendo $A^+ = a_0$, si scrive, sulla seconda riga, al disotto di A^+ , la variabile attiva a_0 .
Sotto B^+ si scrive a_1 e così via...

Ora, ad ogni gruppo, come si è detto, si associa un segnale di memoria:

Uscita di memoria U_1 associata al primo gruppo
Uscita di memoria U_2 associata al secondo gruppo
Uscita di memoria U_3 associata al terzo gruppo

Il segnale di memoria di un gruppo viene settato dal primo finecorsa che si presenta in esso.

Così, nel caso in esame, i segnali di settaggio sono:

S_1 Setta l'uscita U_1 .

S_1 è dato dal primo finecorsa del primo gruppo: a_0 .

$$S_1 = a_0$$

S_2 Setta l'uscita U_2

S_2 è dato dal primo finecorsa del secondo gruppo: b_1 .

$$S_2 = b_1$$

S_3 Setta l'uscita U_3

S_3 è dato dal primo finecorsa del terzo gruppo: c_1 .

$$S_3 = c_1$$

Sulla terza riga si scrive, al disotto del primo finecorsa di un gruppo, il relativo segnale di settaggio. Al disotto di tutti gli altri finecorsa dello stesso gruppo si scrive il relativo segnale di uscita

Così:

- Sulla terza riga del primo gruppo:
Si scrive S_1 al disotto del primo finecorsa del gruppo a_0 e, al disotto di tutti gli altri finecorsa (in questo caso solamente a_1), si scrive il relativo segnale di uscita U_1 (settato da S_1).
- Sulla terza riga del secondo gruppo:
Si scrive S_2 al disotto del primo finecorsa del gruppo b_1 e, al disotto di tutti gli altri finecorsa (in questo caso solamente b_0), si scrive il relativo segnale di uscita U_2 (settato da S_2).
- Sulla terza riga del terzo gruppo:
Si scrive S_3 al disotto del primo finecorsa del gruppo c_1 e, al disotto di tutti gli altri finecorsa (in questo caso solamente c_0), si scrive il relativo segnale di uscita U_3 (settato da S_3).

9°- Scrivere le equazioni logiche di settaggio delle memorie e quella di comando degli stati dei cilindri, adoperando i segnali di uscita dalle memorie in cascata, per contraddistinguere i segnali bloccanti.

Primo gruppo

Iniziando dal primo gruppo, il segnale di settaggio S_1 è dato dal primo finecorsa a_0 . Inoltre vi dovrà essere un segnale di inizio ciclo: *Start*, che verrà posto in *AND* con a_0 .

$$S_1 = a_0 \cdot \text{Start}$$

In sequenza dovrà avvenire che:

- Il segnale di settaggio S_1 attiva l'uscita sulla linea U_1 delle memorie in cascata.
- L'uscita U_1 rimane attivata (in pressione) anche quando, nella corsa di fuoriuscita dello stelo, il finecorsa \bar{a}_0 non è più toccato.
- Quando si attiva U_1 va in scarico l'ultima linea U_5 .

Iniziato il ciclo, essendo ora U_1 la sola linea alimentata, essa può fornire il comando A_1 che porta cilindro A allo stato A^+ .

$$A^+ = U_1$$

Tutti gli altri fine corsa del *I gruppo*, scritti sulla seconda riga, (che comandano gli stati dei cilindri indicati nella prima), si pongono in *AND* con l'uscita attivata U_1 .

Nel caso in esame vi è, oltre il primo fine corsa, solamente a_1 di comando dello stato B^+ ; esso si porrà in *AND* con U_1 .

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

Ricapitolando, i segnali di comando del primo gruppo:

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

Si procede alla stessa maniera per gli altri gruppi.

Secondo gruppo

- Il segnale di settaggio S_2 è attivato dal primo fine corsa b_1 del II gruppo :

$$S_2 = b_1$$

- Il segnale di settaggio S_2 attiva l'uscita sulla linea U_2 delle memorie in cascata.
- Quando si attiva U_2 va in scarico la linea U_1 .
- Essendo ora U_2 la sola linea alimentata, essa può fornire il comando B_0 che porta cilindro B allo stato B^- .

$$B^- = U_2$$

- L'altro fine corsa b_0 del gruppo, che comanda lo stato C^+ , si pone in *AND* con U_2 .

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

Ricapitolando i segnali di comando del 2° gruppo, si ha:

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

Terzo gruppo

- Il segnale di settaggio S_3 è attivato dal primo fine corsa c_1 del III gruppo :

$$S_3 = c_1$$

- Il segnale di settaggio S_3 attiva l'uscita sulla linea U_3 delle memorie in cascata.
- Quando si attiva U_3 va in scarico la linea U_2 ..
- Essendo ora U_3 la sola linea alimentata, essa può fornire il comando C_0 che porta cilindro C allo stato C^- .

$$C^- = U_3$$

- L'altro fine corsa c_0 del gruppo, che comanda lo stato A^- , si pone in *AND* con U_3 .

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

Ricapitolando i segnali di comando del 3° gruppo, si ha:

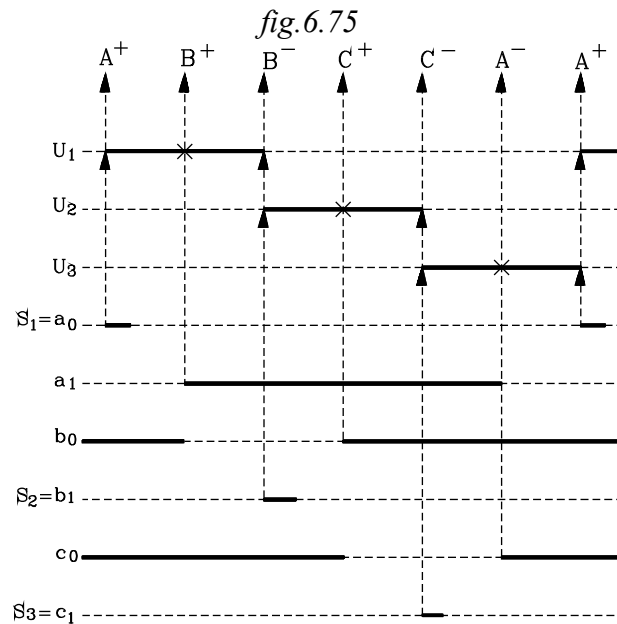
$$III \text{ gruppo} \begin{cases} S_3 = c_1 \\ C^- = U_3 \\ A^- = c_0 \cdot U_3 \end{cases}$$

Diagramma di flusso a barre orizzontali

Oltre ai finecorsa, che costituiscono le variabili attive, sono riportate le tre uscite di memoria U_1, U_2, U_3 . Inoltre le indicazioni S_1, S_2, S_3 sono poste accanto ai finecorsa che effettuano rispettivamente il settaggio di dette uscite.

Il contrassegno di una croce (X), posta sull'intersezione tra la barra orizzontale di una memoria attivata e la linea verticale, spiccata dall'inizio di attivazione di un finecorsa, sta ad indicare l'AND tra questo e detta memoria.

Con tale convenzione è di facile lettura il diagramma.



In sequenza si nota:

Finecorsa a_0 Effettua il settaggio S_1 dell'uscita U_1 , la quale rimane attivata fino a che non viene settato U_2 . Il segnale di attivazione di a_0 è di breve durata.

L'uscita U_1 dà il comando dello stato A^+ :

$$A^+ = U_1$$

Finecorsa a_1 L'intersezione tra la linea verticale tratteggiata, spiccata dall'inizio di attivazione di a_1 , e la barra orizzontale di attivazione di U_1 , segnata con una croce (X), sta ad

indicare che il finecorsa a_1 e l'uscita U_1 sono posti in *AND*. Questo dà il comando dello stato B^+ :

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

Il finecorsa a_1 permane attivato fino a quando viene impartito il comando dello stato A^- .

Finacorsa b_1 Effettua il settaggio S_2 dell'uscita U_2 , che rimane attivata fino a che non viene settato U_3 .

Si ha il resettaggio dell'uscita precedente U_1 .

Il segnale di attivazione di b_1 è di breve durata.

L'uscita U_2 dà il comando dello stato B^- :

$$B^- = U_2$$

NB- il finecorsa a_1 è ancora attivo ma non lo è U_1 per cui il pilotaggio B^+ della valvola bistabile è disattivato ed è possibile impartire all'altro pilotaggio il comando B^- .

Finacorsa b_0 La linea verticale tratteggiata, spiccata dall'inizio di attivazione di b_0 , interseca la barra di attivazione di U_2 .

L'AND tra il finecorsa b_0 e l'uscita U_2 comanda lo stato C^+ :

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

Il finecorsa b_0 permane attivato fino a quando viene impartito il comando dello stato B^+ .

Finacorsa c_1 Effettua il settaggio S_3 dell'uscita U_3 , che rimane attivata fino a che non viene settato U_1 .

Si ha il resettaggio dell'uscita precedente U_2 .

Il segnale di attivazione di c_1 è di breve durata.

L'uscita U_3 dà il comando dello stato C^- :

$$C^- = U_3$$

NB- il finecorsa b_0 è ancora attivo ma non lo è U_2 per cui il pilotaggio C^+ della valvola bistabile è disattivato ed è possibile impartire all'altro pilotaggio il comando C^- .

Finacorsa c_0 La linea verticale tratteggiata, spiccata dall'inizio di attivazione di c_0 , interseca la barra di attivazione di U_3 .

L'AND tra il finecorsa c_0 e l'uscita U_3 comanda lo stato A^- :

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

Il finecorsa c_0 permane attivato fino a quando viene impartito il comando dello stato C^+ .

Circuito pneumatico

Dalle equazioni logiche ne deriva facilmente il circuito pneumatico.

Occorre osservare che:

- I segnali di settaggio S_1 S_2 S_3 vanno inviati agli ingressi delle memorie in cascata.

- Gli altri segnali di comando degli stati dei cilindri A^+ B^+ B^- C^+ C^- A^- vanno inviati ai rispettivi pilotaggi delle valvole bistabili.

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

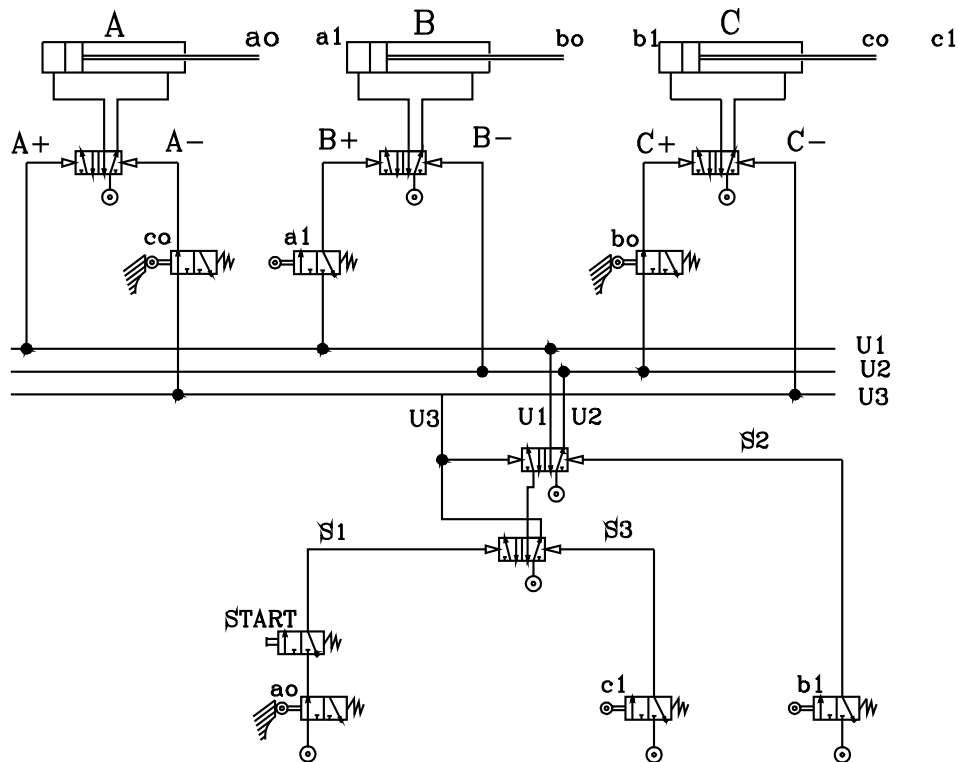
$$II \text{ gruppo} \begin{cases} S_2 = b_1 \\ B^- = U_2 \\ C^+ = b_0 \cdot U_2 \end{cases}$$

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

SCHEMA DEL CIRCUITO PNEUMATICO

Dalle funzioni logiche con le variabili attive si ottiene lo schema seguente.

fig.6.76



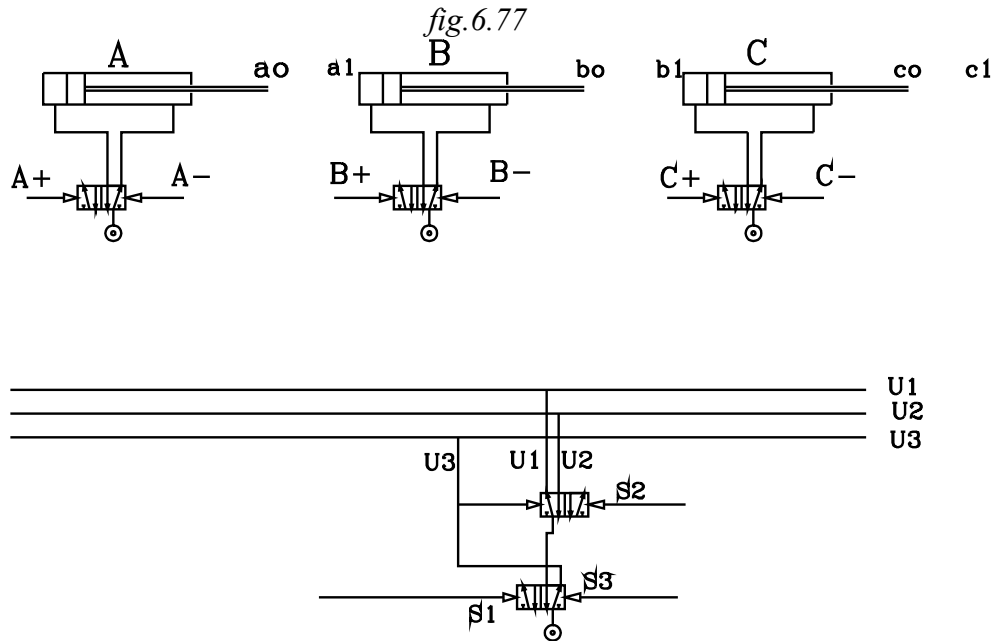
Alcune note sulla stesura dello schema

Per la stesura dello schema conviene procedere nella seguente maniera.

- 1- Si disegnano i cilindri alimentati dalle valvole bistabili 5/2. Gli steli sono nelle condizioni iniziali tutti rientrati, per cui tutte e tre le valvole sono nella posizione 1, nella quale l'aria entra nella camera destra dei cilindri rappresentati in figura.
- 2 Convien subito dopo, effettuare lo schema delle due memorie in cascata che alimentano le tre linee U_1 U_2 U_3 .

Porre attenzione che, affinché possa iniziare il ciclo, nelle condizioni iniziali, occorre che siano soddisfatte quelle finali di "fine ciclo"

Il gruppo di memorie in cascata deve essere settato in modo che l'aria alimenti l'ultima linea U_3
 A questo punto lo schema si presenta nella forma:



- 3 Si concretizzano ora le funzioni logiche scritte.
 Iniziando dal primo gruppo:

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

Equazione logica $S_1 = a_0 \cdot Start$:

Si allaccia al pilotaggio S_1 della valvola di memoria la serie tra il finecorsa a_0 e lo Start.

Nel disegnare il finecorsa a_0 occorre osservare che esso, nelle condizioni iniziali, è toccato e quindi la valvola 3/2 monostabile, che lo rappresenta, va disegnata nella 2° posizione, con l'aria che esce dal condotto di utilizzazione. Lo start va disegnato in prima posizione.

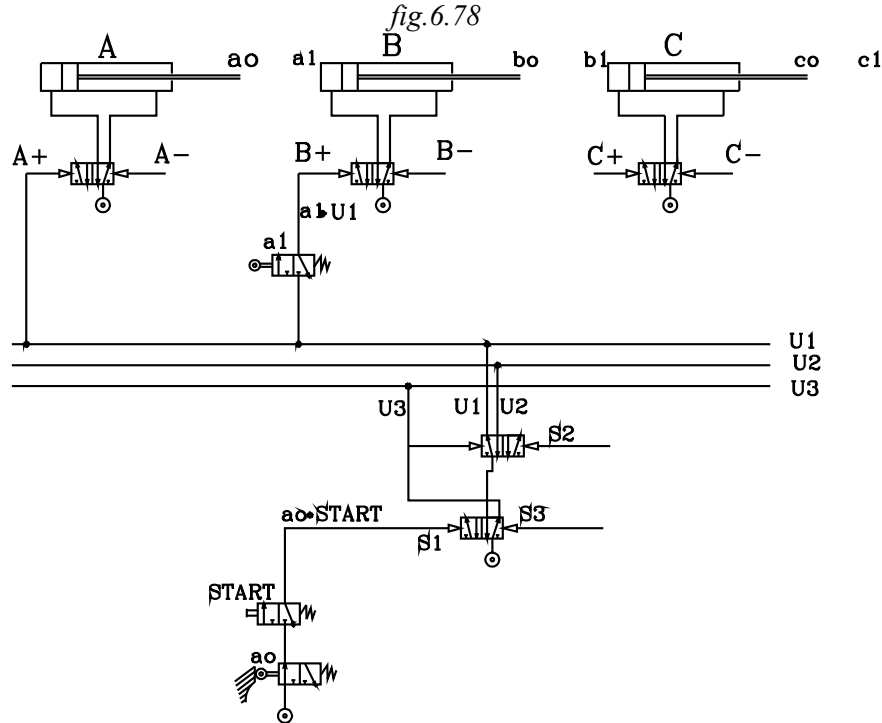
Equazione logica $A^+ = U_1$

Si collega il pilotaggio A^+ della valvola 5/2 che comanda il cilindro A con la linea U_1 .

Equazione $B^+ = a_1 \cdot U_1$

In questo caso, per disegnare lo schema, conviene partire dalla derivazione con linea U_1 quindi, da questa portarsi in ingresso del finecorsa a_1 , rappresentato dalla valvola 3/2 monostabile. Nelle condizioni iniziali il finecorsa a_1 non è toccato, quindi va disegnato in 1° posizione: l'aria non va in uscita.

Effettuando quanto detto si ottiene lo schema parziale:



Alla stessa maniera si procede per la realizzazione schematica delle funzioni logiche degli altri gruppi

Funzionamento

Nelle condizioni iniziali sono toccati i finecorsa a_0 , b_0 , c_0 . La valvola I delle memorie in cascata alimenta la linea U_3 , che porta l'aria direttamente al pilotaggio C^- e, attraverso il finecorsa c_0 toccato, ad A^- . In tal modo, alimentato l'impianto, spontaneamente, gli steli dei cilindri A e C si pongono in posizione rientrata. Occorre, invece, o manualmente o con un segnale di preimpostazione, porre la valvola bistabile di potenza che comanda il cilindro B in posizione di commutazione che alimenti la camera lato stelo.

Ora in sequenza si ha:

- Pigiato lo Start, l'aria prelevata dal finecorsa a_0 toccato, fornisce alle memorie in cascata il segnale di settaggio S_1 della linea U_1 . Infatti, S_1 commuta la valvola I che portandosi in seconda posizione alimenta l'uscita di utilizzazione A e pone in scarico la B. In tal modo l'aria, dall'uscita A della valvola I, perviene a quella dello stesso tipo della valvola II in cascata. Viene alimentata la linea U_1 mentre si porta in scarico la U_3 .
- La linea U_1 porta l'aria al pilotaggio A^+ ottenendo la corsa di fuoriuscita dello stelo del cilindro A. Il pilotaggio opposto A^- non è attivato, essendo ora U_3 in scarico.
- Toccato il finecorsa a_1 , questo commutandosi prende l'aria dalla linea U_1 e l'invia al pilotaggio B^+ ottenendo la corsa di fuoriuscita dello stelo del cilindro B. Il pilotaggio opposto B^- non è attivato essendo la linea U_2 in scarico.
- Toccato il finecorsa b_1 , questo invia l'aria al settaggio S_2 delle memorie in cascata. In tal modo si pone in scarico la linea U_1 e viene alimentata la U_2 . Infatti, l'aria che esce da b_1 commuta la valvola II e porta

l'aria sull'uscita di utilizzazione B collegata con U_2 , mentre l'uscita A, collegata con U_1 viene posta in scarico.

- La linea U_2 invia l'aria al pilotaggio B^- determinando il rientro del cilindro B.
Il pilotaggio opposto B^+ non è attivato essendo ora U_1 in scarico
- Toccato il finecorsa b_0 , questo, commutandosi, prende l'aria dalla linea U_2 e l'invia al pilotaggio C^+ ottenendo la corsa di fuoriuscita dello stelo del cilindro C.
Il pilotaggio opposto C^- non è attivato essendo la linea U_3 in scarico.
- Toccato il finecorsa c_1 , questo invia l'aria al settaggio S_3 delle memorie in cascata. In tal modo si pone in scarico la linea U_2 e viene alimentata la U_3 . Infatti, l'aria che esce da c_1 commuta la valvola I e porta l'aria sull'uscita di utilizzazione B collegata con U_3 , mentre l'uscita A viene posta in scarico.
Il collegamento U_3 alimenta la corrispondente linea e, contemporaneamente resetta la valvola precedente II, la quale si riporta nella posizione di inizio ciclo, dove sia U_1 che U_2 sono in scarico.
- La linea U_3 invia l'aria al pilotaggio C^- determinando il rientro del cilindro C.
Il pilotaggio opposto C^+ non è attivato essendo ora U_2 in scarico
- Toccato il finecorsa c_0 , questo, commutandosi, prende l'aria dalla linea U_3 e l'invia al pilotaggio A^- ottenendo la corsa di rientro dello stelo del cilindro A.
Il pilotaggio opposto A^+ non è attivato essendo la linea U_1 in scarico.

Per fissare le idee sul procedimento da seguire con il metodo delle memorie in cascata, vengono qui di seguito, brevemente, risolti altri cicli, senza dilungarsi sulla scelta dei componenti e altri punti che, ovviamente, vanno sviluppati in una relazione tecnica.

IN LABORATORIO

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

Diagramma corsa passo

fig.6.79

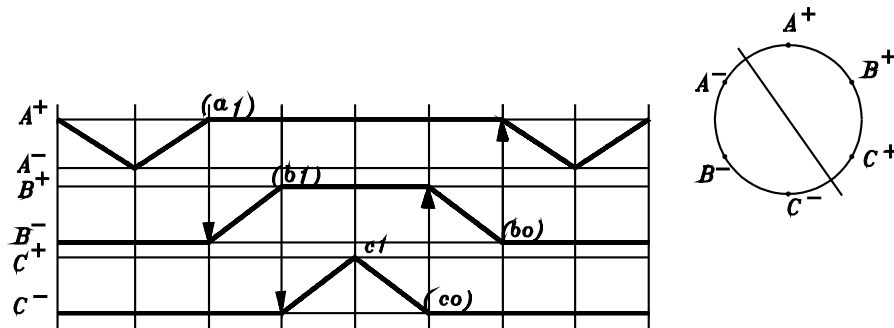


Tabella degli ordini

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

Equazioni logiche complete e con variabili attive

Equazioni logiche complete	Equazioni logiche con variabili attive
$A^+ = a_0 \cdot b_0 \cdot c_0$	$A^+ = a_0$
$B^+ = a_1 \cdot b_0 \cdot c_0$ *	$B^+ = a_1$ *
$C^+ = a_1 \cdot b_1 \cdot c_0$ *	$C^+ = b_1$ *
$C^- = a_1 \cdot b_1 \cdot c_1$	$C^- = c_1$
$B^- = a_1 \cdot b_1 \cdot c_0$ *	$B^- = c_0$ *
$A^- = a_1 \cdot b_0 \cdot c_0$ *	$A^- = b_0$ *

Nelle equazioni logiche compaiono 4 combinazioni a due a due uguali:

$$B^+ = a_1 \cdot b_0 \cdot c_0$$

$$C^+ = a_1 \cdot b_1 \cdot c_0$$

$$A^- = a_1 \cdot b_0 \cdot c_0$$

$$B^- = a_1 \cdot b_1 \cdot c_0$$

Diagramma di flusso a barre orizzontali

Dal diagramma di flusso a barre orizzontali facilmente si rilevano le variabili attive bloccanti

Finecorsa a_1 Comanda B^+ e permane attivato anche quando viene comandato B^- .

Il segnale di finecorsa a_1 è bloccante.

Finecorsa b_1 Comanda C^+ e permane attivato anche quando viene comandato C^- .

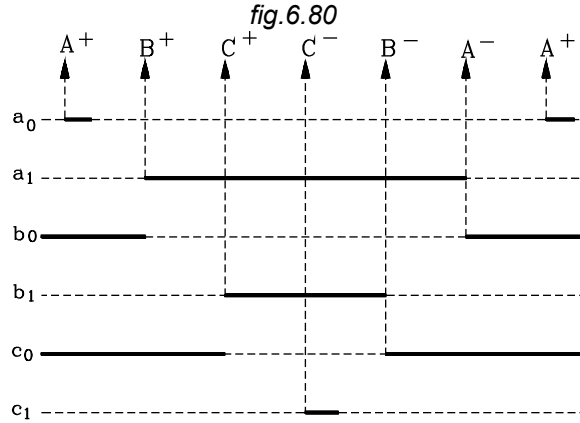
Il segnale di finecorsa b_1 è bloccante.

Finecorsa c_0 Comanda B^- e permane attivato anche quando viene comandato B^+ .

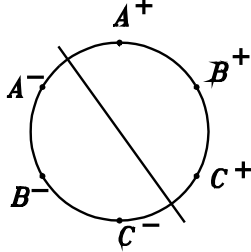
Il segnale di finecorsa c_0 è bloccante.

Finecorsa b_0 Comanda A^- e permane attivato anche quando viene comandato A^+ .

Il segnale di finecorsa b_0 è bloccante.



Divisione in gruppi
fig.6.81



Matrice dei segnali

La sequenza delle fasi viene suddivisa in due gruppi non contenenti doppie corse dei cilindri.

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

I GRUPPO			II GRUPPO		
A ⁺	B ⁺	C ⁺	C ⁻	B ⁻	A ⁻
<i>a</i> ₀	<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₁	<i>c</i> ₀	<i>b</i> ₀
<i>S</i> ₁	<i>U</i> ₁	<i>U</i> ₁	<i>S</i> ₂	<i>U</i> ₂	<i>U</i> ₂

Equazioni logiche

I GRUPPO

Settaggio memoria *U*₁

È dato dal primo finecorsa del gruppo *a*₀.

Lo Start si può porre in AND in questo comando o in *A*⁺:

$$S_1 = a_0 \cdot Start$$

Primo comando del gruppo

Il primo comando sui cilindri del gruppo può essere fornito dal segnale di memoria settato:

$$A^+ = U_1$$

Gli altri comandi del gruppo

Sono dati dall'AND tra la memoria del gruppo (*U*₁) e i finecorsa scritti nella matrice, sotto gli stati comandati:

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

$$C^+ = b_1 \cdot U_1$$

Equazioni I GRUPPO

$$\left\{ \begin{array}{l} S_1 = a_0 \cdot Start \\ A^+ = U_1 \\ B^+ = a_1 \cdot U_1 \\ C^+ = b_1 \cdot U_1 \end{array} \right. \quad \text{Lo Start si può porre anche in } A^+ \quad \left\{ \begin{array}{l} S_1 = a_0 \\ A^+ = U_1 \cdot Start \\ B^+ = a_1 \cdot U_1 \\ C^+ = b_1 \cdot U_1 \end{array} \right.$$

II GRUPPO

Si procede alla stessa maniera

Settaggio memoria U₂

È dato dal primo finecorsa del gruppo c₁.

$$S_2 = c_1$$

Primo comando del gruppo

Il primo comando sui cilindri del gruppo può essere fornito dal segnale di memoria settato

$$C^- = U_2$$

Gli altri comandi del gruppo

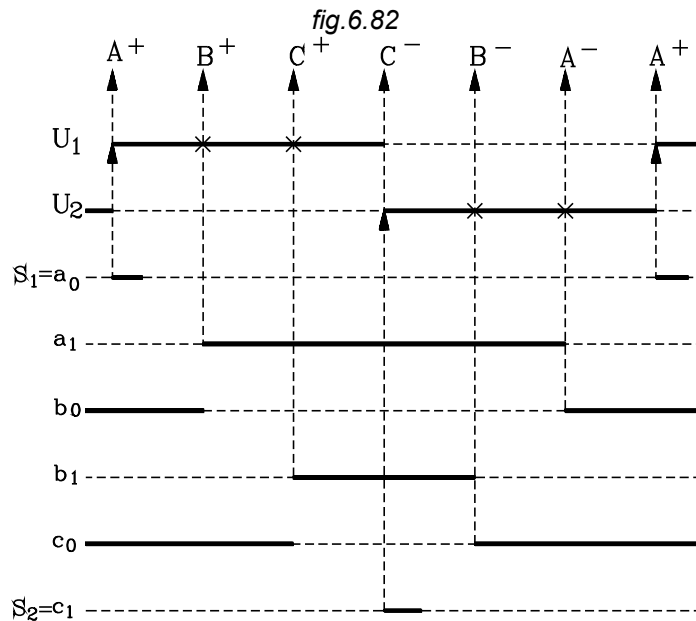
Sono dati dall'AND tra la memoria del gruppo (U₂) e i finecorsa scritti nella matrice, sotto gli stati comandati

$$B^- = c_0 \cdot U_2$$

$$A^- = b_0 \cdot U_2$$

Equazioni II GRUPPO

$$\left\{ \begin{array}{l} S_2 = c_1 \\ C^- = U_2 \\ B^- = c_0 \cdot U_2 \\ A^- = b_0 \cdot U_2 \end{array} \right.$$



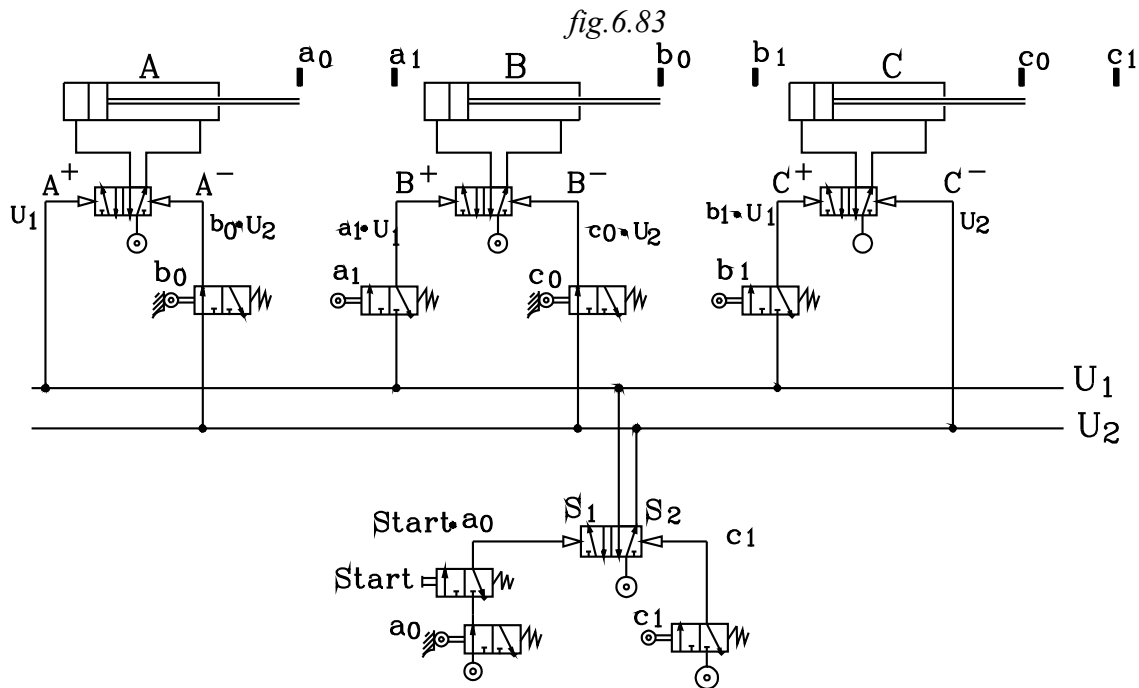
Letture del diagramma

- Il settaggio S_1 è dato dal finecorsa a_0 : $S_1 = a_0$
- S_1 setta la memoria U_1 e resetta U_2
- La memoria U_1 comanda A^+ : $A^+ = U_1$ (nel diagramma non è indicato lo Start)
- Il finecorsa a_1 in AND (segnato dalla croce X) con U_1 comanda B^+ : $B^+ = a_1 \cdot U_1$
- Il finecorsa b_1 in AND (segnato dalla croce X) con U_1 comanda C^+ : $C^+ = b_1 \cdot U_1$

- Il settaggio S_2 è dato dal finecorsa c_1 : $S_2 = c_1$
- S_2 setta la memoria U_2 e resetta U_1
- La memoria U_2 comanda C^- : $C^- = U_2$
- Il finecorsa c_0 in AND (segnato dalla croce X) con U_2 comanda B^- : $B^- = c_0 \cdot U_2$
- Il finecorsa b_0 in AND (segnato dalla croce X) con U_2 comanda A^- : $A^- = b_0 \cdot U_2$

Dalle equazioni logiche determinate si ricava facilmente il circuito, pneumatico

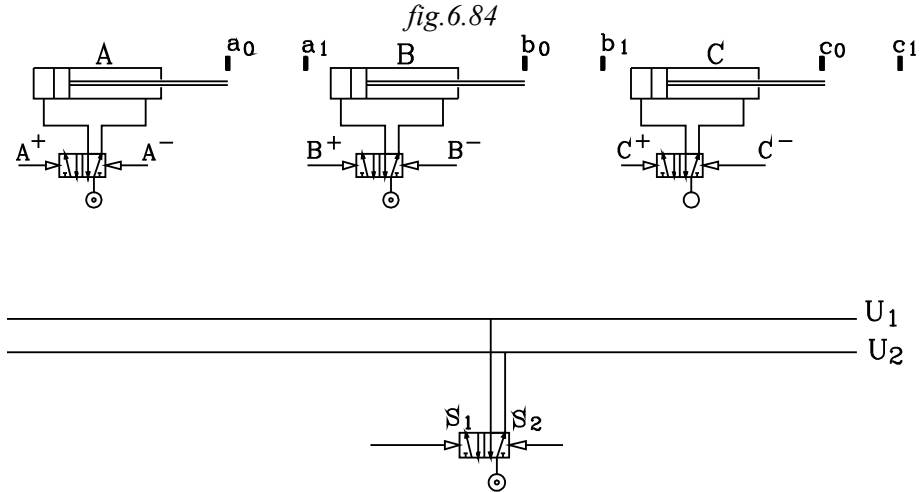
Equazioni I GRUPPO	$\begin{cases} S_1 = a_0 \cdot Start \\ A^+ = U_1 \\ B^+ = a_1 \cdot U_1 \\ C^+ = b_1 \cdot U_1 \end{cases}$	Equazioni II GRUPPO	$\begin{cases} S_2 = c_1 \\ C^- = U_2 \\ B^- = c_0 \cdot U_2 \\ A^- = b_0 \cdot U_2 \end{cases}$
--------------------	--	---------------------	--



Note sulla stesura del circuito

Si procede alla stessa maniera del circuito precedente

- Disegnare prima i tre cilindri comandati dalle valvole bistabili
- Disegnare subito l'unica memoria che alimenta le due linee U_1 U_2 posta nelle condizioni iniziali: in I posizione in modo da alimentare la linea U_2 (nelle condizioni iniziali debbono essere state attuate le finali)

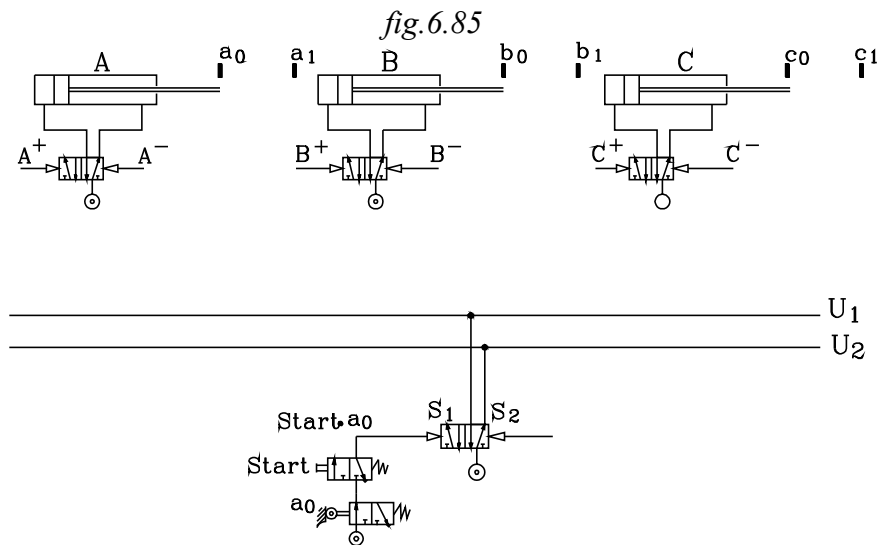


Tradurre le equazioni logiche in circuito pneumatico

Iniziando così da S_1 :

$$S_1 = a_0 \cdot Start$$

Si disegna il fincorsa a_0 in serie con lo Start e l'uscita di questo si collega con il pilotaggio S_1 delle memoria



- E così via...

Funzionamento

- Nelle condizioni iniziali è alimentata la linea U_2 . Prelevata da questa, l'aria fa rientrare gli steli, direttamente nel cilindro C e attraverso i fincorsa b_0 nel cilindro A, c_0 nel cilindro B.

- *Pigiato lo Start si commuta la memoria. Si resetta U_2 e viene alimentata la linea U_1 . Questa è collegata con il pilotaggio A^+ della valvola bistabile. Si ha la fuoriuscita dello stelo di A. Il pilotaggio A^- è disattivato essendo resettata U_2 (vedere anche il diagramma di flusso)*
- *Toccato il finecorsa a_1 , questo, commutandosi, prende l'aria dalla linea U_1 e la invia al pilotaggio B^+ delle valvola bistabile. Si ottiene la fuoriuscita dello stelo di B. Il pilotaggio B^- non è attivato in quanto U_2 è in scarico.*
- *Toccato il finecorsa b_1 , questo, commutandosi, prende l'aria dalla linea U_1 e la invia al pilotaggio C^+ della valvola bistabile. Si ottiene la fuoriuscita dello stelo di C. Il pilotaggio C^- non è attivato in quanto U_2 è in scarico.*
- *Toccato il finecorsa c_1 , questo, commutandosi, invia l'aria al pilotaggio S_2 della memoria. Si ha il resettaggio della uscita sulla linea U_1 e il settaggio della U_2 .*
- *La linea U_2 porta l'aria al pilotaggio C^- . si ottiene la corsa di rientro dello stelo di C. Il pilotaggio C^+ non è attivato in quanto, pur essendo ancora settato b_1 che ha comandato C^+ , la linea U_1 è in scarico (osservare anche il diagramma di flusso).*
- *Toccato il finecorsa c_0 , questo, commutandosi, prende l'aria dalla linea U_2 e la invia al pilotaggio B^- della valvola bistabile. Si ottiene il rientro dello stelo di B. Il pilotaggio B^+ non è attivato in quanto, pur essendo ancora settato a_1 che ha comandato B^+ , la linea U_1 è in scarico.*
- *Toccato il finecorsa b_0 questo, commutandosi, prende l'aria dalla linea U_2 e la invia al pilotaggio A^- della valvola bistabile. Si ottiene il rientro dello stelo di A. Il pilotaggio A^+ non è attivato in quanto la linea U_1 , che ha comandato A^+ , è in scarico.*

Si vuole ora realizzare un ciclo pneumatico nel quale, con una opportuna suddivisione in gruppi, si può minimizzare il numero di memorie, occorrenti a far divenire attivanti i segnali di finecorsa bloccanti.

IN LABORATORIO

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

Diagramma corsa passo

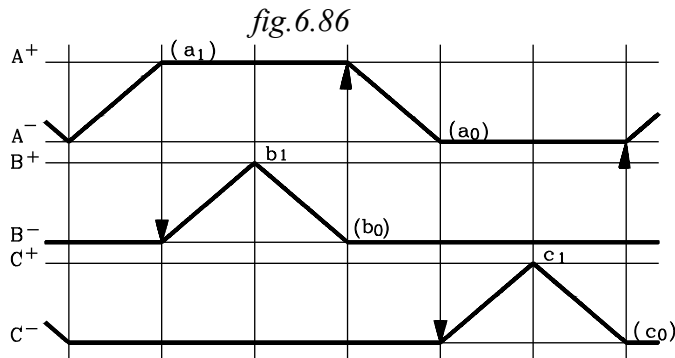


Tabella degli ordini

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

Equazioni logiche complete e con variabili attive

Equazioni logiche complete	Equazioni logiche con variabili attive
$A^+ = a_0 \cdot b_0 \cdot c_0$ *	$A^+ = c_0$ *
$B^+ = a_1 \cdot b_0 \cdot c_0$ *	$B^+ = a_1$ *
$B^- = a_1 \cdot b_1 \cdot c_0$	$B^- = b_1$
$A^- = a_1 \cdot b_0 \cdot c_0$ *	$A^- = b_0$ *
$C^+ = a_0 \cdot b_0 \cdot c_0$ *	$C^+ = a_0$ *
$C^- = a_0 \cdot b_0 \cdot c_1$	$C^- = c_1$

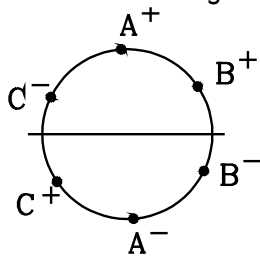
Suddivisione in gruppi non contenenti doppie corse

La sequenza ciclica, considerata a partire da A^+ , può essere suddivisa nei seguenti tre gruppi, non contenenti doppie corse dello stesso cilindro:

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

Essendo la sequenza ciclica, la suddivisione in gruppi non contenenti doppie corse dello stesso cilindro, può essere effettuata partendo da una qualsiasi fase.

fig.6.87



Conviene porre le fasi del ciclo su di una circonferenza e suddividerle nel minimo numero possibili di gruppi non contenenti le due corse opposte dello stesso cilindro

Il ciclo, partendo dalla fase C^- può essere suddiviso in *due gruppi*, nei quali non vi compaiono di uno stesso cilindro sia la corsa di fuoriuscita che di rientro: le corse opposte sono in gruppi diversi.

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

Matrice dei segnali

I GRUPPO			II GRUPPO		
C^-	A^+	B^+	B^-	A^-	C^+
c_1	c_0	a_1	b_1	b_0	a_0
S_1	U_1	U_1	S_2	U_2	U_2

Equazioni logiche

Conviene effettuare la stesura delle equazioni logiche, partendo dalla effettiva prima fase del ciclo: che è A^+ (e non C^+), in modo da avere la nozione della vera sequenza ciclica.

Lo Start è posto nel comando dello stato A^+

$$A^+ = c_0 \cdot U_1 \cdot Start$$

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

$$S_2 = b_1$$

$$B^- = U_2$$

$$A^- = b_0 \cdot U_2$$

$$C^+ = a_0 \cdot U_2$$

$$S_1 = c_1$$

$$C^- = U_1$$

Come si nota, i primi due comandi del primo gruppo ($S_1 = c_1, C^- = U_1$) vengono effettuati alla fine del ciclo.

Perché il ciclo possa iniziare con la prima fase " $A^+ = c_0 \cdot U_1 \cdot Start$ " occorre che, *nelle condizioni iniziali sia settata la linea U_1* in modo che l'aria da questa possa andare nel fincorsa c_0 e nello Start.

*La valvola bistabile di memoria, nelle condizioni iniziali, deve essere posta nella posizione di commutazione che **determina l'alimentazione della linea U_1***

Poche note sulla stesura del circuito

Come al solito conviene

- *Disegnare i tre cilindri con le valvole di potenza. Nelle condizioni iniziali gli steli sono rientrati: vanno quindi disegnati in tale posizione. Le valvole bistabili sono in prima posizione nella quale viene alimentata l'uscita di utilizzazione B*
- *Disegnare l'unica valvola di memoria con le uscite alimentanti le linee U_1, U_2 . Nelle condizioni iniziali deve essere alimentata la linea U_1 , per cui la valvola 5/2 bistabile deve essere in II posizione nella quale l'aria esce dall'utilizzatore A per alimentare detta linea.*
- *Tradurre le equazioni logiche in schema pneumatico. Così l'equazione:*

$$A^+ = c_0 \cdot U_1 \cdot Start$$

si traduce nel collegamento della linea U_1 con lo Start, questo con il fincorsa c_0 . Il segnale d'uscita da quest'ultimo viene inviato al pilotaggio A^+ . Il fincorsa c_0 va disegnato attivato, come risulta nelle condizioni iniziali

L'equazione $B^+ = a_1 \cdot U_1$ viene realizzata collegando la linea U_1 con l'alimentazione del fincorsa a_1 .

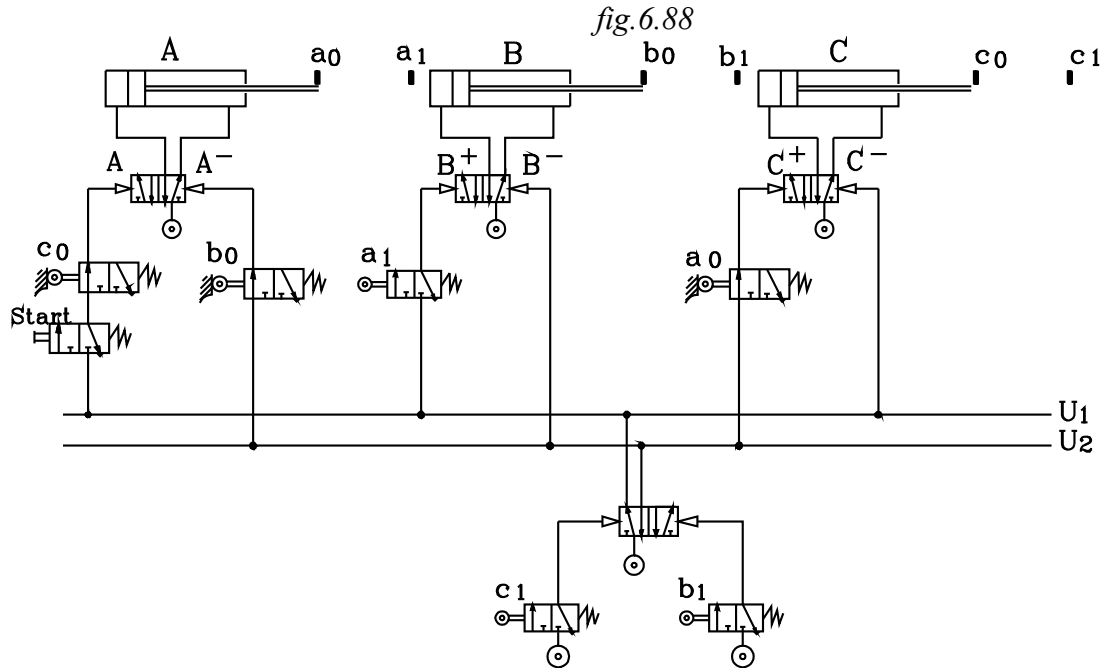
L'uscita di questo va collegato al pilotaggio B^+

E così via...

Funzionamento

- *Nelle condizioni iniziali l'aria alimenta la linea U_1 , attraverso la valvola 5/2 di memoria posta in seconda posizione. Il cilindro C si pone automaticamente in posizione di rientro, mentre occorre porre le valvole bistabili di comando dei cilindri A, B in prima posizione, in modo che i rispettivi steli siano rientrati.*

- Premuto lo Start l'aria prelevata dalla linea U_1 , viene inviata al finecorsa a_0 toccato e da questo si porta al pilotaggio A^+ . Viene così effettuata la corsa di fuoriuscita dello stelo del cilindro A.
- Toccato il finecorsa a_1 l'aria è inviata al pilotaggio B^+ : fuoriesce lo stelo di B.
- Toccato il finecorsa b_1 viene commutata la valvola bistabile di memoria. Si porta in scarico la linea U_1 e si attiva la U_2 .



- L'aria della linea U_2 si porta sul pilotaggio B^- : viene effettuata la corsa di rientro stelo di B
- Toccato il finecorsa b_0 l'aria prelevata dalla linea U_2 viene inviata al pilotaggio A^- : si ha la corsa di rientro dello stelo di A.
- Toccato il finecorsa a_0 l'aria, prelevata da U_2 si porta al pilotaggio C^- : si ha la fuoriuscita dello stelo di C.
- Toccato il finecorsa c_1 si ha la commutazione della valvola bistabile di memoria. la linea U_2 viene posta in scarico e alimentata la U_1 .
- L'aria della linea U_1 si porta al pilotaggio C^- . si ottiene la corsa di rientro stelo del cilindro C.

6.6 CIRCUITI SEQUENZIALI RISOLTI CON IL METODO DELLA MAPPA DI KARNAUGH

Con questo metodo si fa uso delle mappe di Karnaugh.

Nella mappa vengono rappresentati tutti gli stati possibili dei fine corsa, che comandano le valvole distributrici dei cilindri.

I fine corsa possono assumere solamente gli stati logici 0,1.

Si può costruire una mappa completa, data da tutte le combinazioni possibili 0,1 degli stati dei fine corsa.

Consideriamo così due cilindri A, B , ciascuno con due finecorsa: rispettivamente a_0, a_1 (cilindro A) - b_0, b_1 (cilindro B).

Si indicano con

a_0 Il segnale del finecorsa di comando a stelo rientrato del cilindro A , quando è azionato (è toccato), nella condizione logica " $a_0=1$ ".

\bar{a}_0 Il segnale del finecorsa di comando a stelo rientrato del cilindro A , quando non è azionato (non è toccato), nella condizione logica " $a_0=0$ ".

a_1 Il segnale del finecorsa di comando a stelo fuoriuscito del cilindro A , quando è azionato (è toccato), nella condizione logica " $a_1=1$ ".

\bar{a}_1 Il segnale del finecorsa di comando a stelo fuoriuscito del cilindro A , quando non è azionato (non è toccato), nella condizione logica " $a_1=0$ ".

b_0 Il segnale del finecorsa di comando a stelo rientrato del cilindro B , quando è azionato (è toccato), nella condizione logica " $b_0=1$ ".

\bar{b}_0 Il segnale del finecorsa di comando a stelo rientrato del cilindro B , quando non è azionato (non è toccato), nella condizione logica " $b_0=0$ ".

b_1 Il segnale del finecorsa di comando a stelo fuoriuscito del cilindro B , quando è azionato (è toccato), nella condizione logica " $b_1=1$ ".

\bar{b}_1 Il segnale del finecorsa di comando a stelo fuoriuscito del cilindro B , quando non è azionato (non è toccato), nella condizione logica " $b_1=0$ ".

Si costruisca mappa completa con la combinazione di tutti gli stati dei fine corsa.

Sulla orizzontale si rappresentano le combinazioni dei fine corsa di A , sulla verticale quelle dei fine corsa di B .

fig.6.6.1

		ao		ao		
		$\bar{a}_0\bar{a}_1$	\bar{a}_0a_1	a_0a_1	$a_0\bar{a}_1$	
$\bar{b}_0\bar{b}_1$				+		\bar{b}_1
\bar{b}_0b_1				+		
b_0b_1	+	+	+	+		b_1
$b_0\bar{b}_1$				+		\bar{b}_1
		\bar{a}_1	ao	\bar{a}_1		

Come si può notare, vi sono degli stati impossibili, che non si possono presentare in un normale funzionamento.

Questi sono dati dalle combinazioni che prevedono, contemporaneamente, l'azionamento del fine corsa a stelo rientrato e fuoriuscito di un stesso cilindro. Precisamente:

$$a_0a_1 \quad b_0b_1.$$

Togliendo le caselle che rappresentano detti stati, la mappa si riduce nella forma:

fig.6.6.2

	$\bar{a}_0\bar{a}_1$	a_0a_1	$a_0\bar{a}_1$
$\bar{b}_0\bar{b}_1$	+	+	+
\bar{b}_0b_1	+		
$b_0\bar{b}_1$	+		

gli stati intermedi

Vi sono ora delle caselle, segnate con (+), nelle quali è rappresentata la combinazione degli stati dei due finecorsa di uno stesso cilindro entrambi negati:

$$\bar{a}_0\bar{a}_1, \bar{b}_0\bar{b}_1$$

Ciò rappresenta la condizione che si verifica quando lo stelo si trova tra l'inizio e la fine di una corsa.

Nello studio dei circuiti logici ON/OFF che ora tratteremo non interessano le suddette condizioni logiche: vengono analizzati solamente gli stati iniziali e finali di una corsa e non

Si ha così la mappa semplificata, nella quale sono rappresentati i fine corsa azionati; nella condizione logica di comando a steli rientrati $a_0 b_0$ e nella condizione logica di comando a steli fuoriusciti $a_1 b_1$.

Notare che la combinazione \bar{a}_0a_1 sta a significare che il finecorsa a_1 è toccato e non lo è a_0 .

Per semplicità si sottintende che quando è toccato un finecorsa di un cilindro non è toccato l'altro e quindi si pone:

$$\bar{a}_0a_1 = a_1$$

Così pure, con lo stesso significato si pone:

$$\bar{b}_0b_1 = b_1$$

La mappa si presenta così nella forma:

fig.6.6.3

	a_0	a_1
b_0	1	2
b_1	3	4

Ogni casella rappresenta un ben preciso stato di azionamento dei fine corsa, corrispondenti ad una determinata posizione degli steli: all'inizio o alla fine di una corsa (*non sono rappresentati gli stati intermedi*).

Così:

- Casella 1* Azionati $a_0 b_0$. Corrisponde allo stato dei cilindri con tutti e due gli steli rientrati.
- Casella 2* Azionati $a_1 b_0$. Corrisponde allo stato dei cilindri con stelo di A fuoriuscito, stelo di B rientrato.
- Casella 3* Azionati $a_0 b_1$. Corrisponde allo stato dei cilindri con stelo di A rientrato, stelo di B fuoriuscito.
- Casella 4* Azionati $a_1 b_1$. Corrisponde allo stato dei cilindri con tutti e due gli steli fuoriusciti

6.6.1 RAPPRESENTAZIONE DEL CICLO SULLA MAPPA DI KARNAUGH

Per meglio capire il metodo, ci si riferisca dapprima ad un ciclo semplice, non contenente *segnali bloccanti*: che, quindi, non necessita dell'uso di memorie.

6.6.1.1 Ciclo quadro

Si voglia effettuare il ciclo:

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

Diagramma corsa passo:

fig.6.6.4

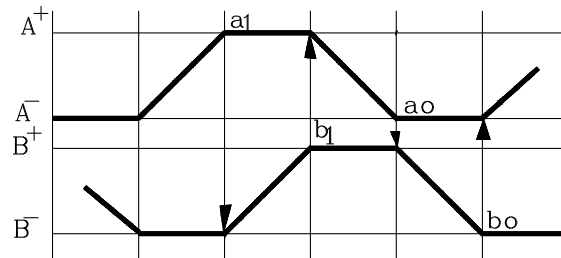


Tabella degli ordini

Stato comandato	Ordine	Variabile	Posizione raggiunta
A^+	A_1	$a_0 \cdot b_0$	$\rightarrow a_1 \cdot b_0$
B^+	B_1	$a_1 \cdot b_0$	$\rightarrow a_1 \cdot b_1$
A^-	A_0	$a_1 \cdot b_1$	$\rightarrow a_0 \cdot b_1$
B^-	B_0	$a_0 \cdot b_1$	$\rightarrow a_0 \cdot b_0$

Rappresentazione sulla mappa semplificata

Ciclo quadro

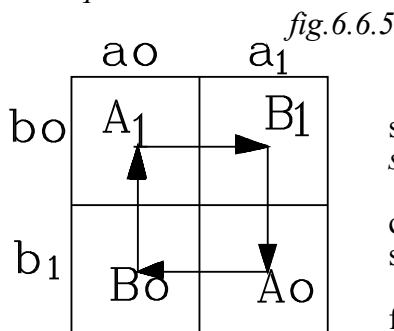


fig.6.6.5

Entro la casella $a_0 b_0$, corrispondente alla posizione dei cilindri con steli rientrati (*toccati* $a_0 b_0$), viene posto l'ordine A_1 (che comanda lo stato A^+).

Il comando A_1 , insieme allo Start, fa fuoriuscire lo stelo del cilindro A, il quale, alla fine, raggiunge il finecorsa a_1 , ottenendo lo stato $a_1 b_0$ (lo stelo di B è ancora rientrato).

Nella mappa si traccia un vettore che congiunge lo stato iniziale e finale dell'ordine impartito.

Si procede alla stessa maniera per gli altri comandi.

Si possono rilevare le seguenti osservazioni generali.

Nelle caselle della mappa sono rappresentate le combinazioni delle condizioni possibili dei finecorsa dei due cilindri nelle posizioni di fuoriuscita e rientro stelo.

L'ordine di fuoriuscita o rientro dello stelo di un cilindro viene scritto entro la casella che rappresenta la combinazione degli stati dei finecorsa che determinano il comando.

L'attuazione del comando viene rappresentato da un vettore, che congiunge la casella indicante la combinazione degli stati iniziali dei finecorsa, quando viene impartito l'ordine, con la casella indicante la combinazione degli stati finali dei finecorsa quando il comando è attuato.

- 1- Il comando A_1 è rappresentato dal vettore (*vedi tabella degli ordini*) che congiunge le caselle $a_0b_0 \rightarrow a_1b_0$
- 2- Il comando B_1 è rappresentato dal vettore che congiunge le caselle $a_1b_0 \rightarrow a_1b_1$
- 3- Il comando A_0 è rappresentato dal vettore che congiunge le caselle $a_1b_1 \rightarrow a_0b_1$
- 4- Il comando B_0 è rappresentato dal vettore che congiunge le caselle $a_0b_1 \rightarrow a_0b_0$

Equazioni logiche espresse con variabili attive

Si è data già precedentemente la definizione di variabile attiva.

Essa è quella che nello stato attuale è variata rispetto allo stato precedente.

Per scrivere l'equazione corrispondente ad un comando, occorre porre in *AND* le variabili che si sono modificate passando dallo stato precedente a quello attuale

Comando A_1

- A_1 è preceduto dal comando B_0 .
- Nel passaggio dallo stato precedente B_0 all'attuale A_1 si va dalla casella indicante la combinazione a_0b_1 a quella indicante a_0b_0 .
- Nel comando si modifica lo stato dei fine corsa di B , che passano dallo stato con b_1 azionato all'altro con b_0 azionato, mentre a_0 rimane inalterato. Inoltre per avviare il ciclo occorre attivare uno *Start*
- Le variabili attive (*le nuove rispetto alla fase precedente*) risultano $b_0 \cdot \text{Start}$:

$$A_1 = b_0 \cdot \text{Start} \quad A_1 \text{ produce lo stato } A^+ \text{ per cui:}$$

$$A^+ = b_0 \cdot \text{Start}$$

Per determinare le variabili attive occorre rispondere alle seguenti due domande:

- 1- Chi precede l'attuale comando?
- 2- Quali variabili cambiano dallo stato precedente a quello attuale?
Le nuove variabili, che compaiono nello stato attuale e non erano presenti nello stato precedente, *sono le variabili attive* e vanno poste in *AND* e formare la funzione di trasmissione del comando del passo attuale.

Comando B₁

Si risponda alle due domande:

Chi precede B₁?

B₁ è preceduto da A₁.

Quale variabile cambia dal comando precedente A₁ all'attuale B₁ ?

Si passa dalla casella a₀b₀ alla casella a₁b₀.

Si ha la variazione a₀ → a₁.

La variabile attiva è a₁.

$$B^+ = a_1$$

Comando A₀

Rispondendo alle due domande si ha che:

- A₀ è preceduto da B₁.
- Si passa dalla casella a₁b₀ alla casella a₁b₁.
- Si ha la variazione b₀ → b₁.

La variabile attiva è b₁.

$$A^- = b_1$$

Comando B₀

Rispondendo alle due domande si ha che:

- B₀ è preceduto da A₀.
- Si passa dalla casella a₁b₁ alla casella a₀b₁.
- Si ha la variazione a₁ → a₀.

La variabile attiva è a₀.

$$B^- = a_0$$

In questo ciclo non vi sono equazioni logiche che si ripetono né vi sono variabili attive bloccanti che permangono azionate in una doppia corsa del cilindro comandato.

Ciò risulta chiaro scrivendo, accanto alle equazioni con le variabili attive quelle complete, espresse con tutte le variabili, costituite dagli stati dei fine corsa.

Equazioni logiche complete		Equazioni logiche con variabili attive	
$A^+ = a_0 \cdot b_0$	*	$A^+ = b_0$	*
$B^+ = a_1 \cdot b_0$	*	$B^+ = a_1$	*
$A^- = a_1 \cdot b_1$		$A^- = b_0$	
$B^- = a_0 \cdot b_1$	*	$B^- = a_0$	*

Non occorre in tal caso l'uso delle memorie

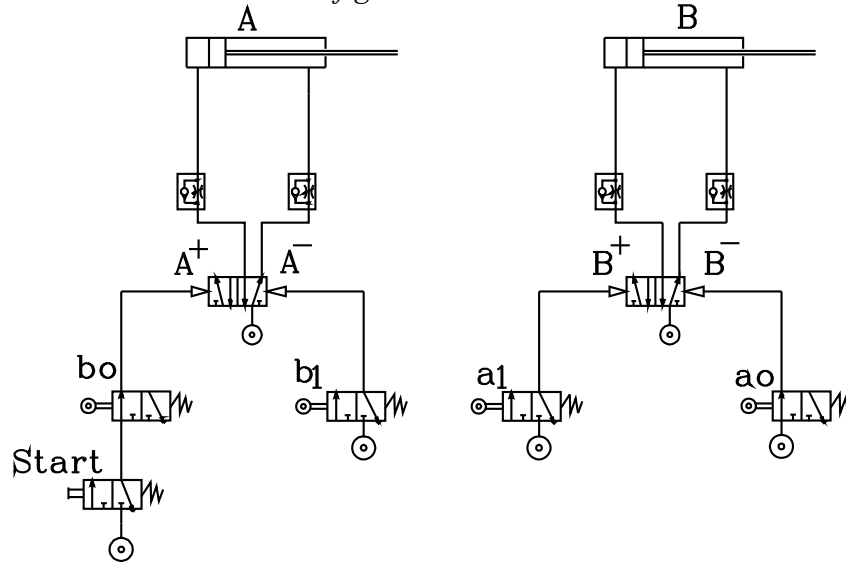
Trasformare in schema pneumatico le equazioni logiche complete comporta un aggravio circuitale con complicazione nel montaggio.

Nel cablaggio di un circuito occorre minimizzare le equazioni logiche: esse vanno espresse in funzione delle variabili attive.

Riassumendo le equazioni logiche con le variabili attive:

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

Dalle equazioni logiche, lo schema del circuito risulta:
fig.6.6.6



Funzionamento

Il funzionamento del circuito è stato già precedentemente spiegato.

6.6.2 CICLI CON SEGNALI BLOCCANTI

Per la comprensione riferiamoci ad un esempio.

6.6.2.1 Ciclo ad L

Si debba effettuare il ciclo:

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

Diagramma corsa passo
fig.6..6.7

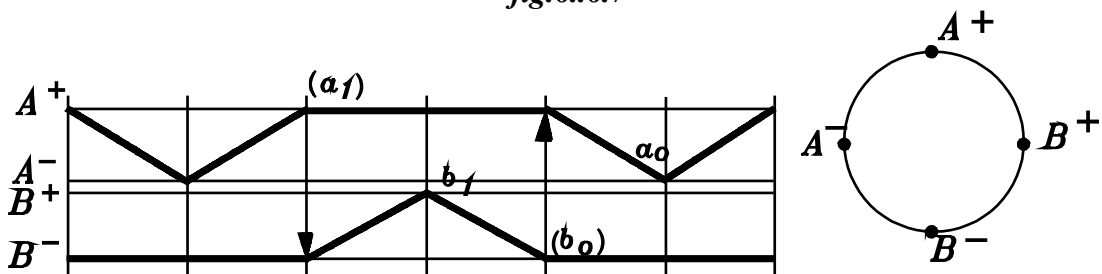


Tabella degli ordini

Stato comandato	Ordine	Variabile	Posizione raggiunta
-----------------	--------	-----------	---------------------

A^+	A_1	$a_0 \cdot b_0$	\rightarrow	$a_1 \cdot b_0$
B^+	B_1	$a_1 \cdot b_0$	\rightarrow	$a_1 \cdot b_1$
B^-	B_0	$a_1 \cdot b_1$	\rightarrow	$a_1 \cdot b_0$
A^-	A_0	$a_1 \cdot b_0$	\rightarrow	$a_0 \cdot b_0$

Le equazioni logiche rispetto alle variabili fine-corsa, scritte in forma completa sono:

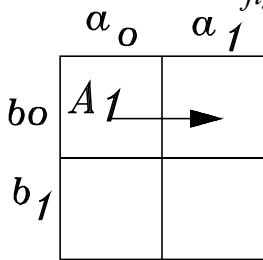
- I $A^+ = a_0 \cdot b_0$
- II $B^+ = a_1 \cdot b_0$
- III $B^- = a_1 \cdot b_1$
- IV $A^- = a_1 \cdot b_0$

Il passo II e il IV sono espressi dalla stessa equazione : vi è un segnale bloccante. Occorre l'utilizzo di una memoria per differenziare le due equazioni uguali.

6.6.2.1.1 Rappresentazione sulla mappa di Karnaugh

Ciclo ad L

fig.6.6.8



La mappa utilizzata è quella semplificata. Sul lato orizzontale riportiamo $a_0 a_1$ e su quello verticale $b_0 b_1$.

Nelle condizioni iniziali (uguali a quelle finali) i due steli sono rientrati: vengono toccati $a_0 b_0$.

Nella casella $a_0 b_0$ viene indicato il comando A_1 (che conduce allo stato A^+).

Nella mappa il *Comando* viene rappresentato da un vettore, che congiunge la casella, indicante le condizioni dei fine-corsa che determinano il comando, con la casella corrispondente allo stato dei fine-corsa, nella *posizione raggiunta* dallo stelo del cilindro all'effettuazione del comandato.

Osservando la tabella degli ordini si ha che:

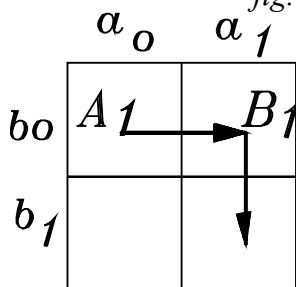
Con i due finecorsa $a_0 b_0$ toccati e azionato lo *Start*, fuoriesce lo stelo di *A*, fino a toccare il fine corsa a_1 . Gli stati finali dei finecorsa attivati sono: $a_1 b_0$.

Comando A_1

Il comando A_1 è rappresentato dal vettore che congiunge le caselle $a_0 b_0 \rightarrow a_1 b_0$

Comando B_1

fig.6.6.9



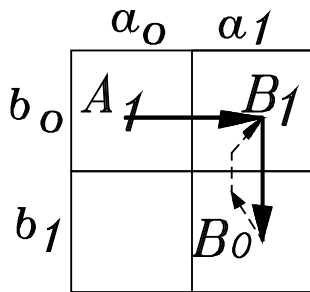
Lo stato $a_1 b_0$ dei fine corsa provoca la fuoriuscita dello stelo del cilindro *B*. Alla fine viene toccato il finecorsa b_1 .

Gli stati finali dei finecorsa attivati sono: $a_1 b_1$

Il comando B_1 è rappresentato dal vettore che congiunge le caselle $a_1 b_0 \rightarrow a_1 b_1$

Impossibilità di impartire il comando B_0

fig.6.6.10



Il seguente comando B_0 deve far rientrare lo stelo del cilindro B . Dalla tabella degli ordini, si nota che il comando B_0 è dato dalla combinazione a_1b_1 e conduce alla combinazione finale a_1b_0 dei fine corsa, la quale deve fornire il comando A_0 di rientro del cilindro A .

B_0 dovrebbe essere rappresentato dal vettore che congiunge le due caselle:

$$a_1b_1 \rightarrow a_1b_0$$

Un vettore cioè che dalla casella a_1b_1 torna indietro nella casella a_1b_0 , che rappresenta la stessa combinazione che determina il comando B_1 di fuoriuscita dello stelo di B .

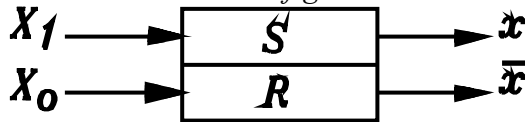
La stessa combinazione delle variabili non può fornire, in tempi diversi, comandi differenti.

La condizione a_1b_0 dei fine corsa si è già verificata nel ciclo: essa serviva a provocare la fuoriuscita dello stelo di B .

La combinazione a_1b_0 non potrà ora essere utilizzata per il rientro dello stelo di A .

Per differenziare l'equazione logica del passo seguente da quella che si è già verificata, occorre attivare, *un segnale di memoria*; proprio ora, nel passo precedente a quello in cui si verificherebbe la ripetizione della funzione logica, in modo che la combinazione che si ripete, a_1b_0 , possa essere posta in *AND* con il segnale di memoria e differenziare così la prima dalla seconda funzione.

fig.6.6.11



Indichiamo con X una memoria di tipo *R-S* (*Set-Reset*).

Il comando di *Set* X_1 (o X^+) produce l'uscita x , mentre il comando di *Reset* X_0 (o X^-) produce l'uscita \bar{x} .

I due segnali di uscita di *Set* o di *Reset* della memoria vanno ad alimentare ciascuno una linea, contrassegnate rispettivamente con x, \bar{x}

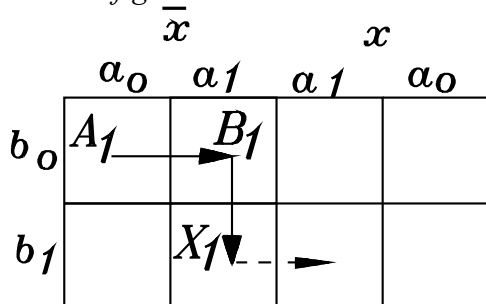
Nelle condizioni iniziali la memoria è *resettata*, per cui è alimentata la linea \bar{x}

Da quanto si è detto, occorre al terzo passo, prima che si ripeta la funzione logica, settare la memoria X col comando X_1 .

Con l'introduzione dei due segnali x, \bar{x} della memoria raddoppiano le combinazioni possibili che si possono effettuare con detti segnali e i finecorsa.

Viene così raddoppiata la mappa di Karnaugh. Ciò si ottiene ribaltando la precedente mappa, specularmente rispetto ad un asse di ribaltamento, coincidente con uno qualsiasi dei quattro lati.

fig.6.6.12



Nel ribaltamento si ha un raddoppio delle caselle della mappa. Una metà viene riferita allo stato \bar{x} della memoria l'altra metà allo stato x della stessa.

Ora le caselle aventi la stessa combinazione dei finecorsa, sono contraddistinte da un segnale di memoria differente.

Comando di settaggi memoria X_1

In tal modo, raggiunto il terzo passo, si attiva con il comando di *Set X_1* la memoria, che si commuta in uscita dallo stato \bar{x} allo stato x

Nella mappa di Karnaugh, completata con i segnali di memoria, il comando di *Set X_1* viene posto nella casella a_1b_1 ed è rappresentato dal vettore tratteggiato che congiunge le caselle con la stessa combinazione dei finecorsa a_1b_1 , ma con segnali opposti di memoria:

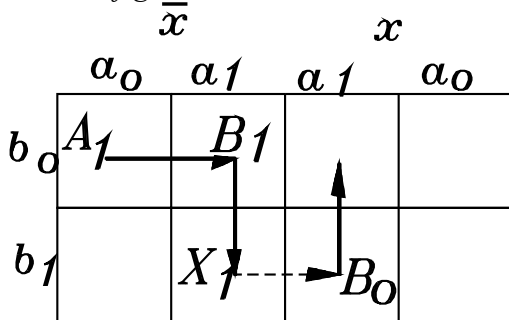
$$a_1b_1\bar{x} \rightarrow a_1b_1x$$

Il vettore è tratteggiato in quanto esso non corrisponde ad un passo del ciclo, ma solamente al settaggio della memoria: i fine corsa rimangono inalterati, ma, nella commutazione della memoria, vengono posti in AND con un segnale opposto di uscita dalla stessa.

Comando B_0

Ora è possibile dare il comando B_0 di rientro dello stelo di B , che conduce alla combinazione dei finecorsa a_1b_0 , posta però questa volta in AND con x : a_1b_0x , ed è quindi diversa dalla funzione logica precedente $a_1b_0\bar{x}$.

fig.6.6.13



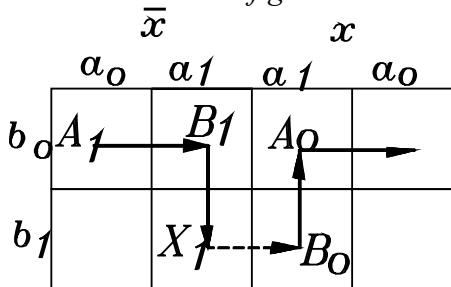
Il comando B_0 viene posto nella casella a_1b_1x ed è rappresentato dal vettore che congiunge le caselle:

$$a_1b_1x \rightarrow a_1b_0x$$

Infatti, con il comando B_0 si ha il rientro dello stelo di B , che va a toccare il fine corsa b_0 , mentre restano attivi il fine corsa a_1 e il segnale x di memoria.

Comando A_0

fig.6.6.14

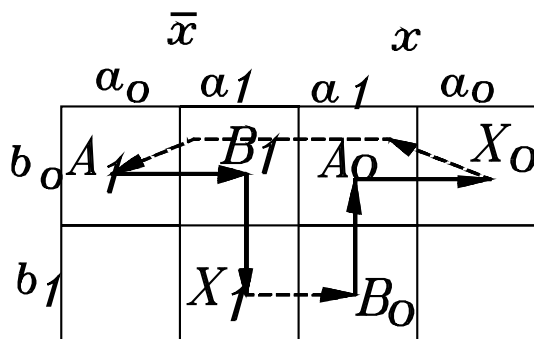


Occorre ora dare il comando A_0 di rientro dello stelo di A . Il comando A_0 è posto nella casella a_1b_0x ed è rappresentato dal vettore che congiunge le caselle:

$$a_1b_0x \rightarrow a_0b_0x$$

Infatti A_0 fa rientrare lo stelo di A che va a toccare il fine corsa a_0 , mentre restano attivi il fine corsa b_0 e il segnale x

di memoria.



Comando di resettaggio memoria X_0

fig.6.6.15

Alla fine del ciclo occorre ripristinare le condizioni iniziali.

Occorre osservare che con l'ultimo passo, sono stati riportati i due cilindri nella loro posizione iniziale con i fine corsa a_0 b_0 toccati; ma la memoria è ancora settata ed ha in uscita il segnale x . Occorre resettare la memoria, in modo di ottenere di nuovo le condizioni iniziali; ciò si ottiene con il comando di resettaggio X_0 .

Nella casella a_0b_0x si indica il comando X_0 che è rappresentato da un vettore tratteggiato che congiunge le caselle:

$$a_0b_0x \rightarrow a_0b_0\bar{x}$$

Con X_0 la memoria viene resettata e passa dallo stato x allo stato \bar{x} , mentre i fine corsa restano immutati.

6.6.2.1.1 Estrazione delle funzioni logiche

In ogni passo occorre determinare le variabili attive di comando.

Le variabili attive, come si è detto, sono quelle che si sono modificate dallo stato precedente a quello considerato.

Le variabili attive in un passo di commutazione di una valvola, non si debbono prolungare nel ciclo, fino ad essere ancora presenti, da sole, nel successivo passo di commutazione della stessa valvola in senso opposto.

Se una variabile attiva si prolunga per un doppio passo di un cilindro è un segnale bloccante

Se le variabili attive si prolungano fino alla commutazione in senso inverso di una valvola, occorre aggiungere ad esse, in AND, uno o più segnali di memoria che le diversifichi.

Per la determinazione delle funzioni logiche si procede come nell'esempio precedente, *studiando però il prolungamento delle variabili attive.*

Per l'estrazione delle funzioni logiche conviene farsi e rispondere a tre domande

- 1- Quale ordine precede il comando?
- 2- Dallo stato comandato precedente all'attuale qual'è la variabile che ha cambiato stato e quale forma ha attualmente?
Tale variabile nella forma attuale è quella attiva.
- 3- Nell'insieme delle caselle rappresentanti la variabile attiva è presente anche il comando opposto a quello da essa dato? (*fuoriuscita-rientro cilindro Set-Reset memoria*)
Se la risposta è no: la funzione è composta dalla sola variabile attiva
Se la risposta è sì: occorre porre in AND alla variabile attiva le memorie corrispondenti alla casella del comando considerato.

Stato comandato A^+

Lo stato A^+ viene comandato da A_1

Per estrarre le equazioni logiche rispondiamo alla tre domande.

I- Chi precede A_1 ?

Precede A_1 il comando X_0

II- Quale variabile cambia?

Dal precedente comando X_0 all'attuale A_1 si ha il cambiamento:

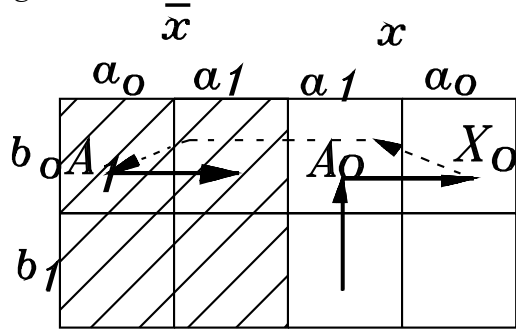
$$xa_0b_0 \rightarrow \bar{x}a_0b_0$$

Varia la variabile x , che passa da x a \bar{x}

La variabile attiva attuale è quindi \bar{x}

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

fig.6.6.16



III- Nella zona di \bar{x} compaiono i comandi opposti A_1 A_0 ?

Per rispondere alla terza domanda occorre verificare se la variabile attiva \bar{x} del comando A_1 della valvola si prolunga nel ciclo, fino ad essere presente quando si verifica il comando opposto A_0 .

Si tratteggi la zona interessante \bar{x} ; si nota che in tale zona non è presente il comando A_0 di commutazione inversa di A_1 ; per cui l'equazione logica di comando dello stato A^+ può essere espressa dalla sola variabile \bar{x} .

Inoltre per avviare il ciclo occorre un segnale di Start.
Si avrà quindi come equazione logica di A^+

$$A^+ = \bar{x} \cdot \text{Start}$$

Stato comandato B^+

Viene comandato da B_1 .

Rispondendo alle prime due domande si ha:

I - Chi precede B_1 ? Precede B_1 il comando A_1

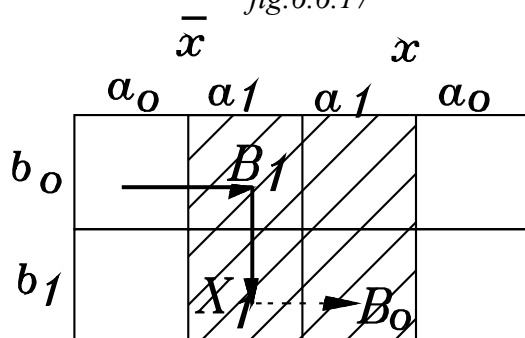
II- Quale variabile cambia? Dal precedente comando A_1 all'attuale B_1 si ha il cambiamento

$$\bar{x} \cdot a_0 \cdot b_0 \rightarrow \bar{x} \cdot a_1 \cdot b_0$$

Si ha la variazione del finecorsa dallo stato a_0 allo stato a_1

La variabile attiva è a_1

fig.6.6.17



III Nella zona di a_1 compaiono i comandi opposti B_1 B_0 ?

Per rispondere a questa domanda si tratteggi la zona interessante la variabile attiva a_1 .

Si nota che in tale zona è presente il comando B_0 di commutazione inversa di B_1 . L'attivazione del finecorsa a_1 si prolunga fino ad essere presente in due commutazioni inverse della stessa valvola.

Occorre differenziare l'equazione logica; per cui si pone in AND ad a_1 il segnale di memoria \bar{x} (si noti che il comando inverso B_0 è nella zona del segnale di memoria x).

L'equazione logica dello stato B^+ è:

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

Comando X_1 di settaggio della memoria

X_1 commuta la memoria dallo stato \bar{x} allo stato x .

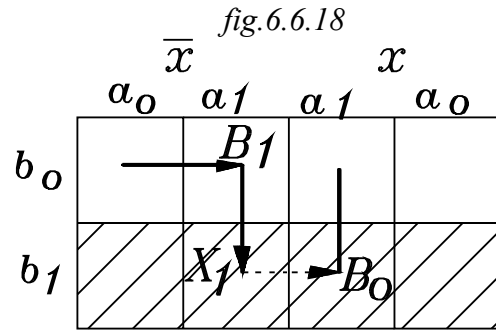
Rispondendo alle prime due domande si ha che:

- Precede X_1 il comando B_1 .
- Dal precedente al comando attuale si verifica il cambiamento delle variabili:

$$\bar{x} \cdot a_1 \cdot b_0 \rightarrow \bar{x} \cdot a_1 \cdot b_1$$

- Si ha la variazione del finecorsa dallo stato b_0 allo stato b_1

La variabile attiva è b_1 .



Per rispondere alla terza domanda si tratteggi la zona interessante b_1 ; si nota che in essa non è presente il comando X_0 di commutazione inversa di X_1 ; per cui l'equazione logica di comando di settaggio X_1 della memoria può essere espressa dalla sola variabile attiva b_1 .

memoria è data dalla sola variabile b_1 .

L'equazione logica del comando di Set X_1 della

$$X_1 = b_1$$

Stato B^-

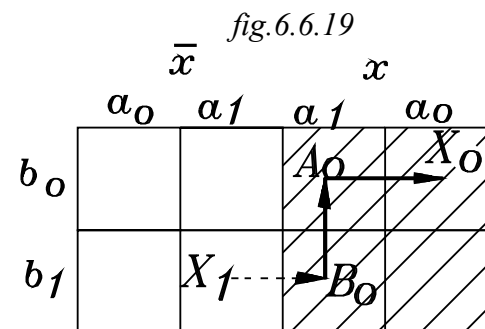
Viene comandato da B_0 .

Rispondendo alle prime due domande si ha:

- I - Chi precede B_0 ? Precede B_0 il comando X_1 .
- II- Quale variabile cambia? Dal precedente comando B_0 all'attuale X_1 si ha il cambiamento:

$$\bar{x} \cdot a_1 \cdot b_1 \rightarrow x \cdot a_1 \cdot b_1$$

La variabile attiva è x .



III- Nella zona di x compaiono i comandi opposti B_1 B_0 ?

Per rispondere alla domanda si tratteggi la zona interessante x . Si nota che in tale zona non è presente il

comando B_1 di commutazione inversa di B_0 ; per cui l'equazione logica di comando dello stato B^- può essere espressa dalla sola variabile x .

L'equazione logica dello stato B^- è:

$$B^- = x$$

Stato A^-

Viene comandato da A_0 .

Rispondendo alle prime due domande si ha:

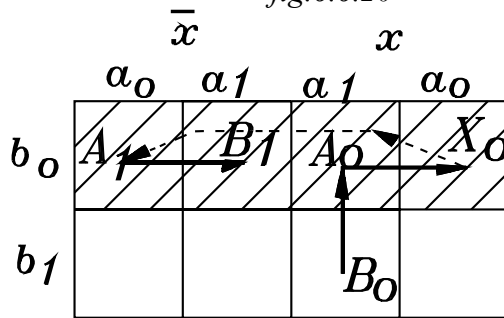
I - Chi precede A_0 ? Precede A_0 il comando B_0

II- Quale variabile cambia? Dal precedente comando A_0 all'attuale B_0 si ha il cambiamento:

$$x \cdot a_1 \cdot b_1 \rightarrow x \cdot a_1 \cdot b_0$$

La variabile attiva è b_0

fig.6.6.20



III Nella zona di b_0 compaiono i comandi opposti A_1 A_0 ?

Per rispondere alla domanda si tratteggi la zona interessante b_0 . Si nota che in tale zona è presente il comando A_1 di commutazione inversa di A_0 . L'attivazione del finecorsa b_0 si prolunga fino ad essere presente in due commutazioni inverse della stessa valvola.

Occorre differenziare l'equazione logica; per cui si pone in AND a b_0 il segnale di memoria x (si noti che il comando inverso A_1 è nella zona del segnale opposto di memoria \bar{x}).

L'equazione logica dello stato A^- è:

$$A^- = x \cdot b_0$$

Comando di Reset X_0 della memoria

X_0 commuta la memoria dallo stato x allo stato \bar{x} .

Rispondendo alle prime due domande si ha:

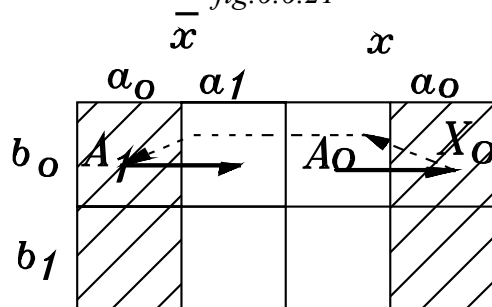
I - Chi precede X_0 ? Precede X_0 il comando A_0

II- Quale variabile cambia? Dal precedente comando A_0 all'attuale X_0 si ha il cambiamento:

$$x \cdot a_1 \cdot b_0 \rightarrow x \cdot a_0 \cdot b_0$$

La variabile attiva è a_0

fig.6.6.21



III Nella zona di a_0 compaiono i comandi opposti X_1 X_0 ?

Per rispondere alla domanda si tratteggia la zona interessante a_0 . Si nota che in tale zona non è presente il comando X_1 di commutazione inversa di X_0 ; per cui l'equazione logica di comando del resettaggio X_0 della memoria può essere espressa dalla sola variabile a_0 .

L'equazione logica del comando di resettaggio della memoria è:

$$X_0 = a_0$$

Riassumendo le equazioni logiche:

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

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

$$X_1 = b_1$$

$$B^- = x$$

$$A^- = x \cdot b_0$$

$$X_0 = a_0$$

Con le equazioni logiche si può tracciare lo schema pneumatico, contenente una sola memoria, con segnale X_1 di Set e X_0 di Reset. In uscita: X_1 Setta lo stato x e X_0 Resetta lo stato \bar{x}

La memoria ottenuta con un 5/2 bistabile alimenta le due linee con segnali di memoria x , \bar{x} .

Nelle condizioni iniziali la memoria è settata in modo da alimentare la linea \bar{x} .

Dalle equazioni logiche si ricava il circuito pneumatico di *fig. 6.6.22*

IN LABORATORIO

Per fissare le idee conviene effettuare una relazione ove viene scritto il procedimento della determinazione delle equazioni logiche mediante l'uso delle mappe di Karnaugh, e montare in laboratorio il circuito.

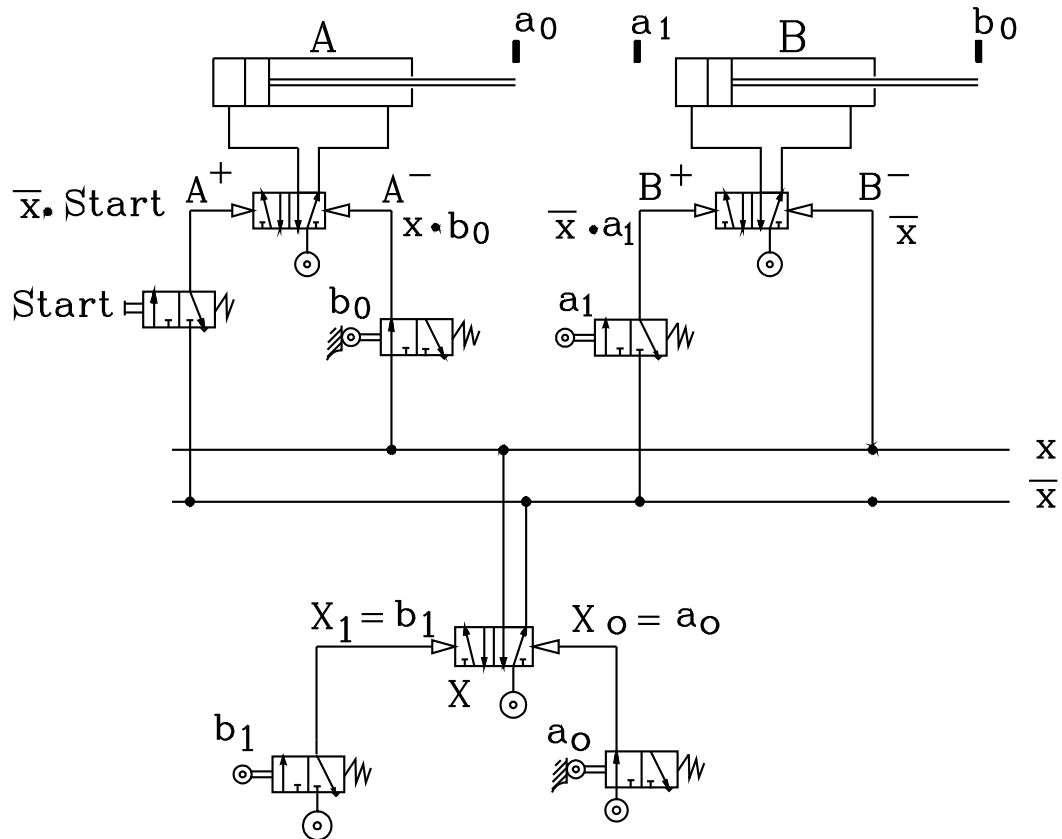
Note sulla stesura del circuito

Non sono state date le specifiche del progetto da realizzare con il ciclo proposto. L'attenzione è stata rivolta solamente alla sua realizzazione mediante l'uso della mappa di Karnaugh. L'unica condizione al contorno specificata è l'avviamento del ciclo mediante uno Start. È evidente che in un progetto reale, le specifiche fornite determinano le condizioni al contorno da aggiungere al circuito di figura.

Così non sono stati posti i regolatori di flusso ecc.

$$A^+ \quad B^+ \quad B^- \quad A^-$$

fig. 6.6.22



- Come al solito, si inizia la stesura disegnando i due cilindri con gli steli nella posizione iniziale: in questo caso entrambi rientrati.
- Si disegnano le valvole di potenza 5/2 bistabili con le indicazioni degli stati dei cilindri da esse comandati.
- Disegnare la memoria X che alimenta le due linee \bar{x} , x . Il pilotaggio X_1 , a sinistra, attiva l'uscita x , quello a destra, X_0 , l'uscita \bar{x} . Nelle condizioni iniziali deve essere alimentata la linea \bar{x}

- Si traducono le equazioni logiche nel corrispondente circuito pneumatico

$A^+ = \bar{x} \cdot Start$ Si deriva dalla linea \bar{x} un tubicino che va all'ingresso dello Start costituito da una valvola 3/2 NC monostabile. L'uscita di questa va al pilotaggio A^+

$B^+ = \bar{x} \cdot a_1$ Si deriva dalla linea \bar{x} un tubicino che va all'ingresso del finecorsa a_1 non toccato, costituito da una valvola 3/2 NC monostabile. L'uscita di questa va al pilotaggio B^+

$X_1 = b_1$ Il finecorsa b_1 alimentato viene collegato in uscita con il pilotaggio X_1 della valvola X di memoria

$B^- = x$ Si deriva dalla linea x un tubicino che va al pilotaggio B^-

$A^- = x \cdot b_0$ Si deriva dalla linea x un tubicino che va all'ingresso del finecorsa b_0 toccato (si disegna nella posizione attivata). L'uscita di questo va al pilotaggio A^-

$X_0 = a_0$ Il finecorsa a_0 , alimentato e disegnato nella posizione attivata (toccato) viene collegato in uscita con il pilotaggio X_0 delle valvola di memoria.

Funzionamento

- Nelle condizioni iniziali è alimentata la linea x . Occorre porre le due valvole di potenza in prima posizione, in modo che l'aria, in uscita dalle valvole alimenti le camere dei cilindri lato stelo.
- Pigiato lo Start, dall'uscita di questo, esce l'aria, che, prelevata dalla linea \bar{x} , va al pilotaggio A^+ . Fuoriesce lo stelo del cilindro A
- Toccato il finecorsa a_1 l'aria viene prelevata dalla linea \bar{x} e inviata al pilotaggio B^+ . Così fuoriesce lo stelo del cilindro B.
- Toccato il finecorsa b_1 , l'aria dall'alimentazione di questo passa in uscita e va al pilotaggio X_1 della memoria. La valvola di memoria si commuta e la linea \bar{x} va in scarico, mentre viene alimentata la linea x
- L'aria dalla linea x viene inviata al pilotaggio B^- , ottenendo il rientro dello stelo del cilindro B. Il pilotaggio B^+ è in scarico, essendo ancora il finecorsa a_1 ancora toccato.
- Toccato il finecorsa b_0 l'aria, prelevata dalla linea x , va al pilotaggio A^- . Si ha il rientro dello stelo del cilindro A. L'altro pilotaggio A^+ è in scarico, anche se è ancora pigiato lo Start, essendo la linea \bar{x} non attivata
- Toccato il finecorsa a_0 , l'aria dall'alimentazione di questo va al pilotaggio X_0 della valvola di memoria che commutandosi alimenta la linea \bar{x} e porta in scarico la x , ripristinando le condizioni iniziali

6.6.2.2 Ciclo con più di due cilindri e più memorie

Nel caso che il ciclo interessi più di due cilindri e in esso si verificano più segnali che si ripetono, conviene distribuire la mappa in modo che sull'orizzontale si leggano i *fine corsa* dei cilindri e sulla verticale siano riportati i segnali di memoria.

Per ogni segnale di memoria introdotto, conviene raddoppiare la mappa in senso verticale, secondo un asse di simmetria, coincidente con il lato orizzontale inferiore.

Come esempio si consideri il ciclo:

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

Diagramma corsa passo

fig.6.6.23

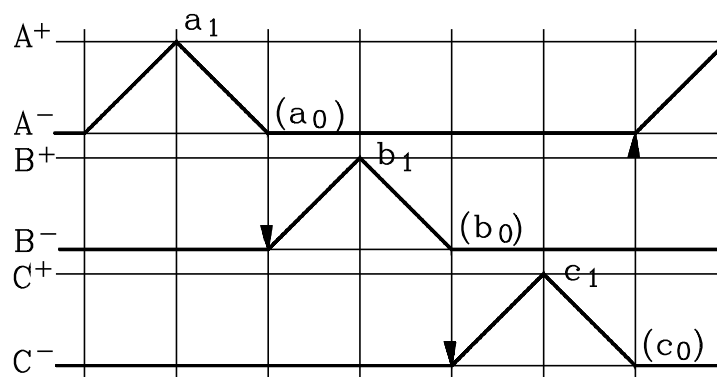


Tabella degli ordini

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	$a_0 b_0 c_0$	$\rightarrow a_0 b_1 c_0$
B^-	B_0	$a_0 b_1 c_0$	$\rightarrow a_0 b_0 c_0$
C^+	C_1	$a_0 b_0 c_0$	$\rightarrow a_0 b_0 c_1$
C^-	C_0	$a_0 b_0 c_1$	$\rightarrow a_0 b_0 c_0$

Le equazioni logiche complete scritte con le variabili *fine corsa*:

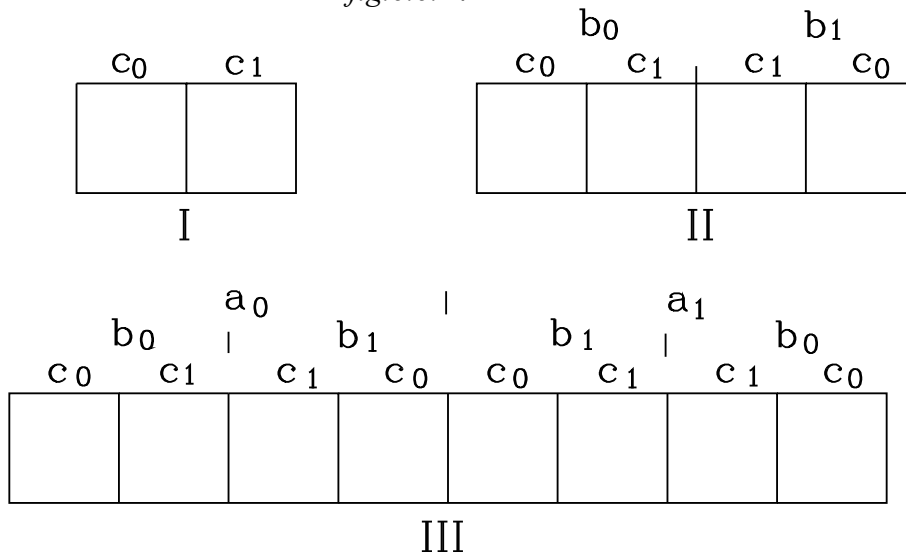
- I $A^+ = a_0 b_0 c_0$
- II $A^- = a_1 b_0 c_0$
- III $B^+ = a_0 b_0 c_0$
- IV $B^- = a_0 b_1 c_0$
- V $C^+ = a_0 b_0 c_0$
- VI $C^- = a_0 b_0 c_1$

I passi I, III, V sono espressi dalla stessa equazione: vi sono 3 segnali bloccanti. *Occorre l'impiego di 2 memorie.*
 Sono disponibili $2^2 = 4$ uscite di memoria > 3 segnali uguali.

6.6.2.1 Rappresentazione sulla mappa di Karnaugh

Si costruisca la mappa nella seguente maniera:

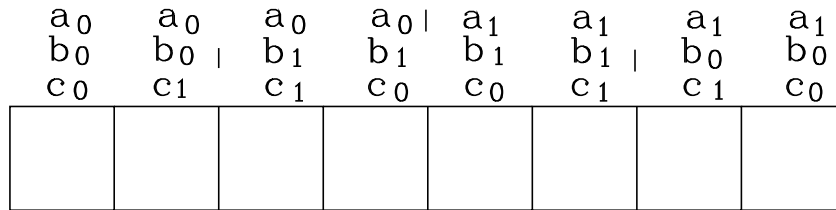
fig.6.6.24



- Partendo dalla *fig.6.6.24 I*, si disegnano le due caselle rappresentanti le attivazioni dei fine corsa di $C : c_0, c_1$.
- Introducendo i fine corsa di $B : "b_0, b_1"$, si raddoppia specularmente la mappa, secondo l'asse di simmetria, coincidente con il lato verticale destro della precedente mappa, ottenendo quella di *fig.6.6.24 II*.

- Si introducono in ultimo i fine corsa di A: " a_0, a_1 ".
Occorre raddoppiare ancora specularmente la mappa secondo l'asse di simmetria coincidente con il lato verticale destro della precedente mappa: si ottiene la mappa di *fig.6.6.24 III*.

Si scriva su ogni casella la combinazione dei fine-corsa e si ottiene la mappa di *fig.6.6.25*.



Si rappresentano ora i comandi nelle caselle, come si è fatto nel precedente esempio.

Si traccia il vettore che congiunge la casella, che rappresenta i fine corsa toccati, con quella indicante la posizione raggiunta alla fine della corsa comandata.

Quando, durante il ciclo, il vettore torna su una casella dove è stato già rappresentato un precedente comando, si raddoppia specularmente la mappa rispetto al lato orizzontale inferiore.

Nel raddoppio la prima parte della mappa si riferisce al segnale di *Reset* di memoria (*quella già esistente*), la parte speculare al segnale di *Set*.

Comando A1

Nelle condizioni iniziali i fine corsa $a_0 b_0 c_0$ sono toccati. Azionato lo *Start* fuoriesce lo stelo di *A*.

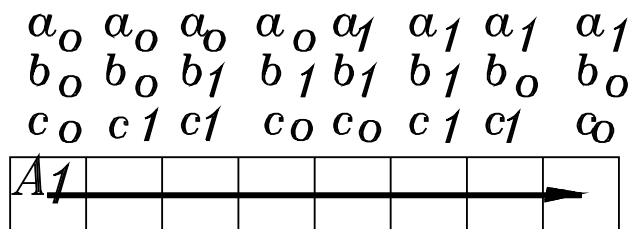
Nella casella $a_0 b_0 c_0$ viene scritto il comando A_1 . Questo viene rappresentato dal vettore che congiunge le caselle :

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

Con i tre fine corsa $a_0 b_0 c_0$ toccati e azionato lo start lo stelo di *A* fuoriesce, fino a toccare il fine corsa a_1 :

Si ha lo stato finale $a_1 b_0 c_0$.

fig.6.6.26



Settaggio memoria X1

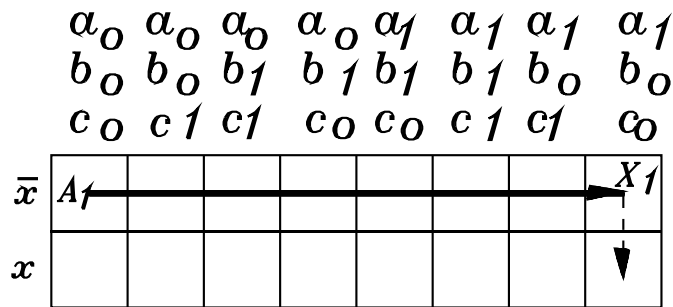
Il comando A_0 deve far tornare indietro lo stelo di *A* e, alla fine, i fine corsa assumono di nuovo la precedente configurazione $a_0 b_0 c_0$.

Il vettore dovrebbe tornare sulla precedente casella. Ciò non deve avvenire, altrimenti si verificherebbe che, una stessa combinazione dello stato dei fine-corsa, deve, in tempi diversi, produrre movimentazioni differenti dei cilindri.

Si introduce la memoria X

La mappa viene raddoppiata specularmente rispetto al lato orizzontale inferiore e le due parti, una superiore e l'altra inferiore, si riferiscono rispettivamente ai segnali di memoria \bar{x} e x .

fig.6.6.27



Il gruppo di caselle superiori si riferiscono allo stato \bar{x} di non settaggio della memoria, mentre la parte speculare inferiore si riferisce allo stato opposto x di memoria settata.

Si impartisce ora l'ordine X_1 di Set della memoria che la commuta dallo stato \bar{x} allo stato x

Si scrive X_1 nella casella $\bar{x} a_1 b_0 c_0$.

Per rappresentare il comando di settaggio della memoria, si traccia un vettore tratteggiato che congiunge le caselle:

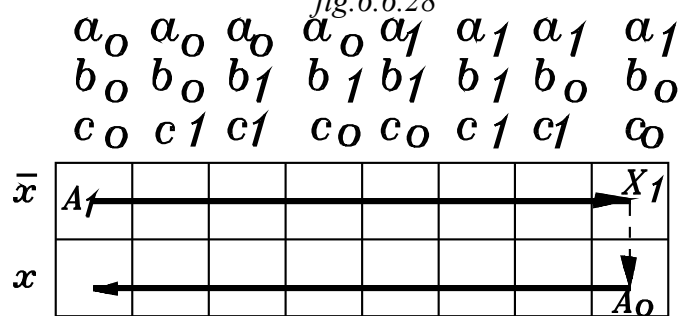
$$\bar{x} a_1 b_0 c_0 \rightarrow x a_1 b_0 c_0$$

rappresentanti lo stesso assetto dei fine corsa ma con in AND due segnali opposti di memoria.

Comando A_0 .

Ora è possibile impartire il comando A_0 , che riporta i fine corsa nelle stesse condizioni $a_0 b_0 c_0$, già presentate in una casella, ma con uno stato di memoria differente (*la memoria non era Settata, quindi lo stato era $\bar{x} a_0 b_0 c_0$*).

fig.6.6.28



Si scrive A_0 entro la casella $x a_1 b_0 c_0$.

Il comando A_0 è rappresentato da un vettore che congiunge le caselle:

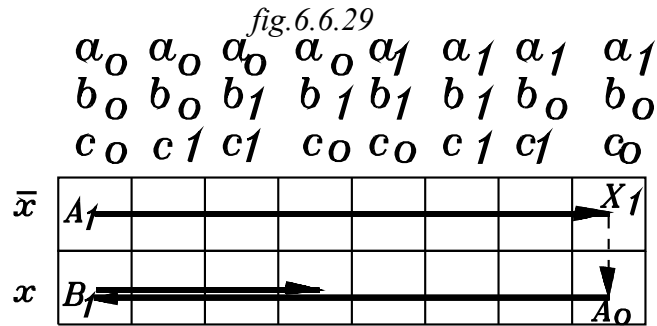
$$x a_1 b_0 c_0 \rightarrow x a_0 b_0 c_0$$

Comando B_1

Con i fine corsa $a_0 b_0 c_0$ toccati e attivato il segnale di memoria x fuoriesce lo stelo di B , fino a toccare il fine corsa b_1 .

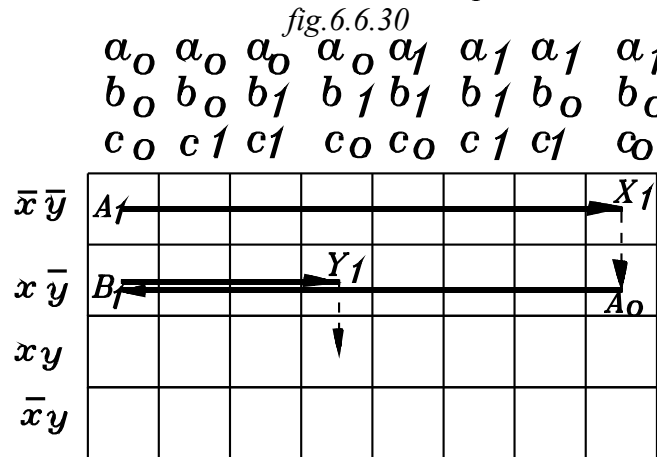
Il comando B_1 viene scritto nella casella $x a_0 b_0 c_0$ ed è rappresentato dal vettore che congiunge le caselle:

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



Settaggio della memoria Y_1

Con il comando B_0 deve rientrare lo stelo di B .
 Senza attivare un'altra memoria il vettore tornerebbe sulla precedente casella.



Si introduce un'altra memoria Y che, settata con Y_1 e resettata con Y_0 fornisce, in uscita, altri due segnali \bar{y} , y atti a differenziare le stesse funzioni logiche dei fine corsa.

La mappa viene raddoppiata specularmente rispetto al lato orizzontale inferiore e le due parti, una superiore e l'altra inferiore, si riferiscono rispettivamente ai segnali di memoria \bar{y} , y .

Occorre notare che, con l'introduzione della memoria Y , ora, nei lati verticali delle caselle vi compaiono le combinazioni dei segnali delle due memorie X e Y .

Nei comandi precedentemente considerati va aggiunto lo stato della memoria Y , che è \bar{y} (prima del segnale di Set Y_1)

Nella casella $x\bar{y}a_0b_1c_0$ si scrive il segnale di Set Y_1 , che commuta la memoria dallo stato \bar{y} allo stato y .

Il comando Y_1 è rappresentato dal vettore tratteggiato che congiunge le caselle:

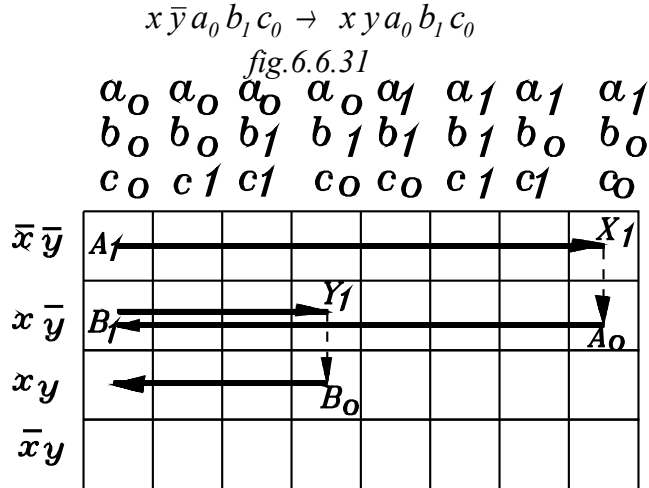
$$x\bar{y}a_0b_1c_0 \rightarrow xy a_0b_1c_0$$

nelle quali vi è indicata la stessa combinazione dei finecorsa ma con i segnali opposti di memoria \bar{y} , y .

Comando B_0

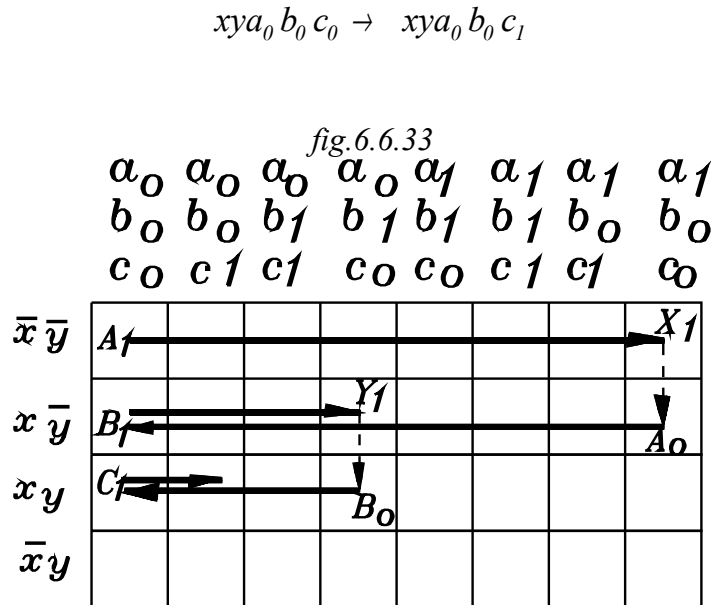
È il comando di rientro dello stelo del cilindro B .
 Attuato il comando, vengono toccati i fine corsa $a_0 b_0 c_0$, con le memorie nello stato xy .
 Si deve attuare quando si verifica la combinazione $a_0 b_1 c_0$ dei fine-corsa.
 Si scrive B_0 nella casella $xya_0 b_1 c_0$.

Il comando B_0 è rappresentato da un vettore che congiunge le caselle:



Comando C_1

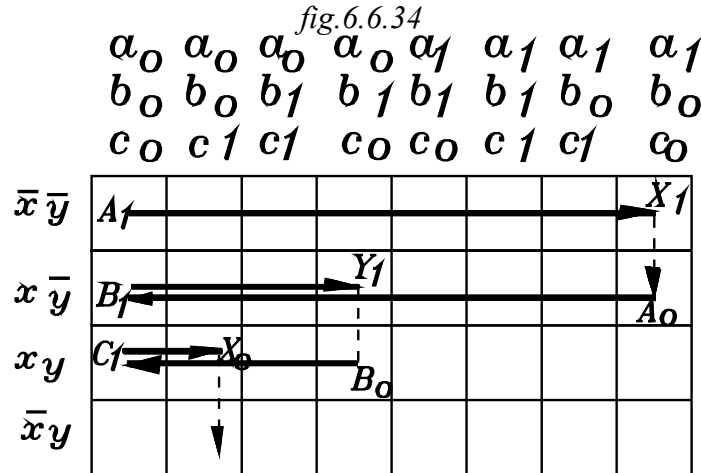
Comanda la fuoriuscita dello stelo del cilindro C .
 Si ha quando si verifica la condizione $xya_0 b_0 c_0$.
 Attuato il comando, lo stelo tocca il fine corsa c_1 , ottenendo la condizione $xya_0 b_0 c_1$.
 Il comando C_1 è rappresentato da un vettore che congiunge le caselle:



Segnale di Reset X_0

Occorre far rientrare lo stelo del cilindro C . Se non si cambiasse lo stato di una memoria, il vettore tornerebbe nella precedente casella.

Avendo introdotto due memorie, X, Y , vi sono a disposizione $2^2=4$ combinazioni possibili dei segnali \bar{x}, x, \bar{y}, y .



Due combinazioni sono state già utilizzate con il settaggio, prima di $X (X_1)$ e poi di $Y (Y_1)$. Restano altre due combinazioni a disposizione, in ordine: il Reset X_0 di X , Il Reset Y_0 di Y .

Con il primo comando di commutazione si differenzia il segnale uguale dei finecorsa a_0, b_0, c_0 , utilizzato per il rientro del cilindro C , con il secondo si riporta il ciclo nelle condizioni iniziali (nella prima casella).

L'unica casella che nel ciclo viene toccata due volte dai vettori è quella iniziale, coincidente con la finale.

Nella casella $xya_0b_0c_1$ viene scritto il segnale di Reset X_0 . Esso è rappresentato da un vettore tratteggiato che congiunge le caselle:

$$xya_0b_0c_1 \rightarrow \bar{x}ya_0b_0c_1$$

Occorre porre attenzione di resettare le memorie precedentemente settate prima di chiudere il ciclo. Ciascuna memoria può essere resettata da un finecorsa. Si deve così prevedere, negli ultimi passi, di resettare una alla volta le memorie settate.

Nel caso in studio il resettaggio della memoria x viene utilizzato per ottenere un segnale di differenziazione di segnali bloccanti. Rimane settata l'altra memoria y che potrà essere resettata nell'ultimo passo.

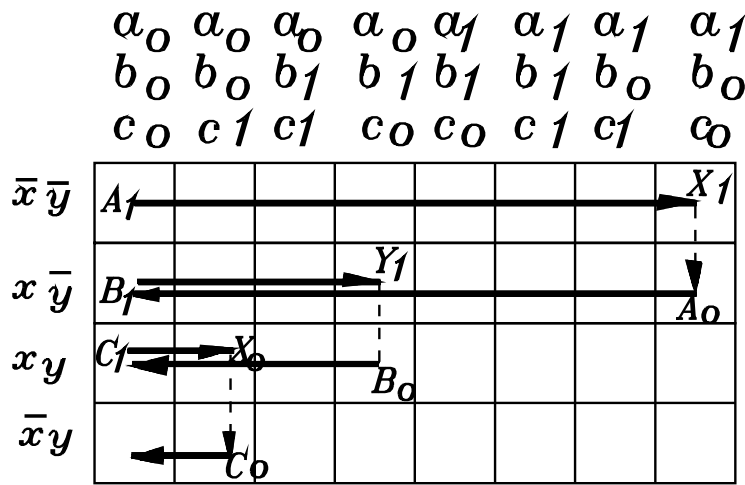
Comando C_0

Il comando C_0 fa rientrare lo stelo di C . Si deve attivare quando si verifica la combinazione dei fine-corsa: a_0, b_0, c_1 e le memorie sono nello stato $\bar{x}y$.

Il comando C_0 è rappresentato da un vettore che congiunge le caselle:

$$\bar{x}ya_0b_0c_1 \rightarrow \bar{x}ya_0b_0c_0$$

fig.6.6.35



Segnale di Reset Y_0

Prima di chiudere il ciclo occorre resettare le memorie precedentemente settate. La memoria x è stata già resettata, rimane da disattivare la memoria y .

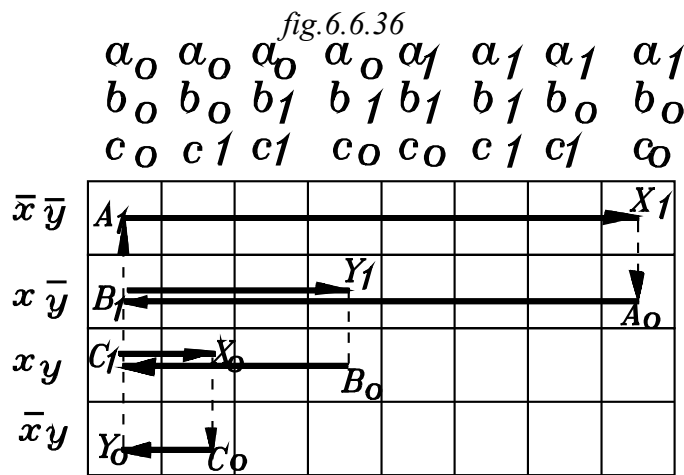
Il segnale Y_0 resetta la memoria Y e la commuta dallo stato y allo stato \bar{y} .

Rimangono inalterate le condizioni dei fine-corsa, che sono ormai nella condizione finale (*uguali a quelle iniziali*).

Si ritorna quindi nelle condizioni iniziali sia nei riguardi dei fine-corsa che dello stato delle memorie (*entrambi resettate*).

Y_0 viene scritto nella casella $x\bar{y}a_0b_0c_0$ ed è rappresentato dal vettore tratteggiato che congiunge le caselle:

$$\bar{x}y a_0 b_0 c_0 \rightarrow \bar{x}\bar{y} a_0 b_0 c_0$$



La mappa di Karnaugh si presenta così nella forma completa, rappresentante tutti i comandi e le combinazioni dei segnali che li attivano.

6.6.2.2 Estrazioni delle funzioni logiche

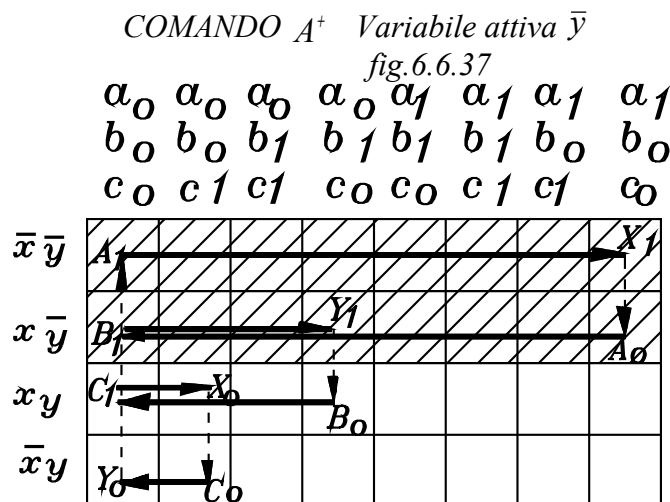
Si procede alla stessa maniera dell'esempio precedente.

- In ogni passo si individuano le variabili attive dei fine-corsa o delle memorie che determinano il comando.
- Si tratteggia la zona interessanti le variabili attive individuate.
- Si verifica se in detta zona vi è il comando del passo considerato e il suo opposto (*del tipo A_1-A_0 X_1-X_0 ...*).
- Se nella zona non si riscontrano comandi di segno opposto, allora l'equazione che caratterizza il passo è dato dalle sole variabili attive riscontrate.
- Se nella zona tratteggiata si riscontrano comandi di segno opposto, occorre porre in *AND* con le variabili attive i segnali di memoria che le differenzino.

Per semplificare l'estrazione delle funzioni logiche conviene rispondere alle solite tre domande:

- 1- Quale comando precede quello attuale?
- 2- Dallo stato comandato precedente all'attuale qual'è la variabile che ha cambiato stato e quale è la sua forma attuale?
Quest'ultima è la variabile attiva.
- 3- Nell'insieme delle caselle rappresentanti la variabile attiva è presente anche il comando opposto a quello da essa dato ? (*fuoriuscita-rientro cilindro Set-Reset memoria*).
Se la risposta è no: la funzione è composta dalla sola variabile attiva.
Se la risposta è si: occorre porre in *AND* alla variabile attiva le memorie corrispondenti alla casella del comando considerato.

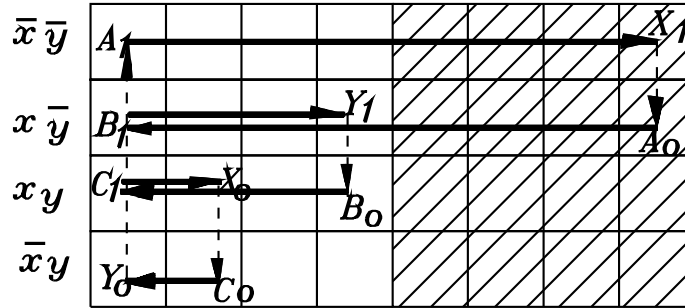
Qui di seguito, dalla *fig.6.6.37* alla *fig.6.6.47*, vengono tratteggiate le zone interessanti la variabile attiva di ogni comando.



COMANDO X_1 Variabile attiva a_1

fig.6.6.38

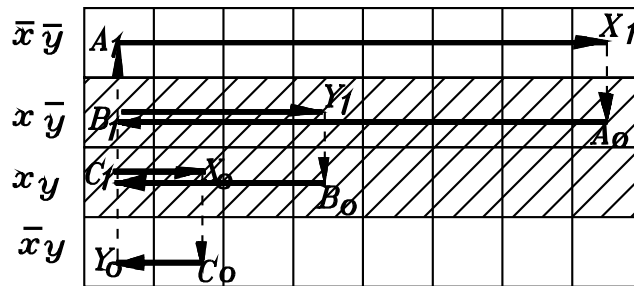
a_0 a_0 a_0 a_0 a_1 a_1 a_1 a_1
 b_0 b_0 b_1 b_1 b_1 b_1 b_0 b_0
 c_0 c_1 c_1 c_0 c_0 c_1 c_1 c_0



COMANDO A^- Variabile attiva x

fig.6.6.39

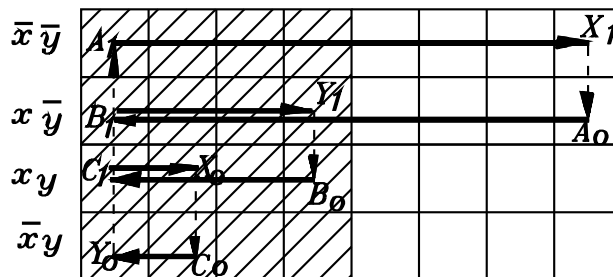
a_0 a_0 a_0 a_0 a_1 a_1 a_1 a_1
 b_0 b_0 b_1 b_1 b_1 b_1 b_0 b_0
 c_0 c_1 c_1 c_0 c_0 c_1 c_1 c_0



COMANDO B^+ Variabile attiva a_0

fig.6.6.40

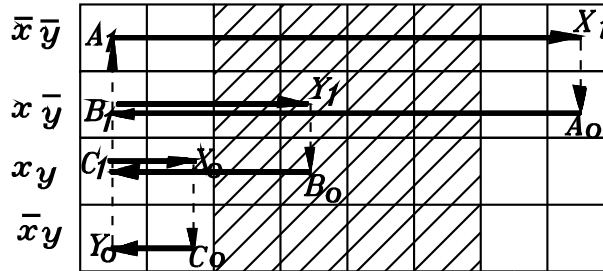
a_0 a_0 a_0 a_0 a_1 a_1 a_1 a_1
 b_0 b_0 b_1 b_1 b_1 b_1 b_0 b_0
 c_0 c_1 c_1 c_0 c_0 c_1 c_1 c_0



COMANDO Y_1 Variabile attiva b_1

fig.6.6.41

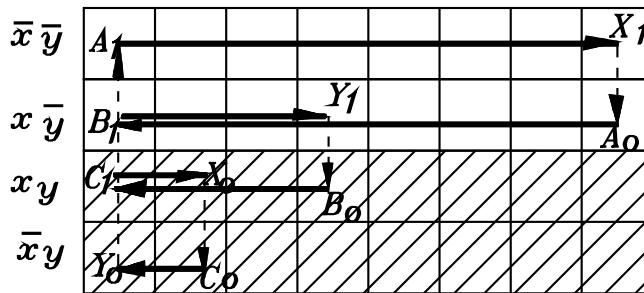
$a_0 a_0 a_0 a_0 a_1 a_1 a_1 a_1$
 $b_0 b_0 b_1 b_1 b_1 b_1 b_0 b_0$
 $c_0 c_1 c_1 c_0 c_0 c_1 c_1 c_0$



COMANDO B^- Variabile attiva y

fig.6.6.42

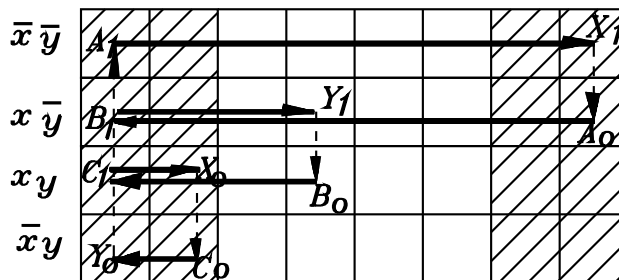
$a_0 a_0 a_0 a_0 a_1 a_1 a_1 a_1$
 $b_0 b_0 b_1 b_1 b_1 b_1 b_0 b_0$
 $c_0 c_1 c_1 c_0 c_0 c_1 c_1 c_0$



COMANDO C^+ Variabile attiva b_0

fig.6.6.43

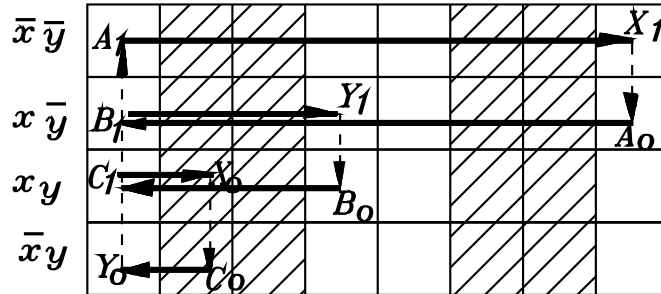
$a_0 a_0 a_0 a_0 a_1 a_1 a_1 a_1$
 $b_0 b_0 b_1 b_1 b_1 b_1 b_0 b_0$
 $c_0 c_1 c_1 c_0 c_0 c_1 c_1 c_0$



COMANDO X_0 Variabile attiva c_1

fig.6.6.44

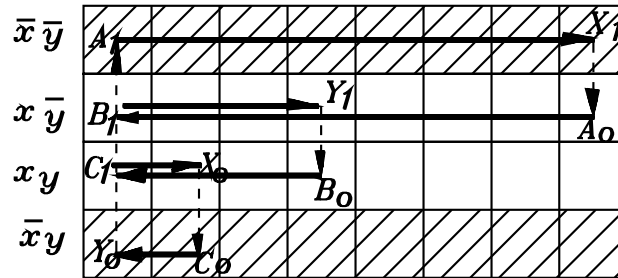
$a_0 a_0 a_0 a_0 a_1 a_1 a_1 a_1$
 $b_0 b_0 b_1 b_1 b_1 b_1 b_0 b_0$
 $c_0 c_1 c_1 c_0 c_0 c_1 c_1 c_0$



COMANDO C^- Variabile attiva \bar{x}

fig.6.6.45

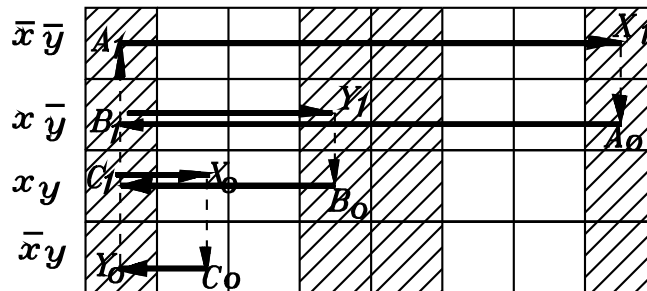
$a_0 a_0 a_0 a_0 a_1 a_1 a_1 a_1$
 $b_0 b_0 b_1 b_1 b_1 b_1 b_0 b_0$
 $c_0 c_1 c_1 c_0 c_0 c_1 c_1 c_0$



COMANDO Y_0 Variabile attiva c_0

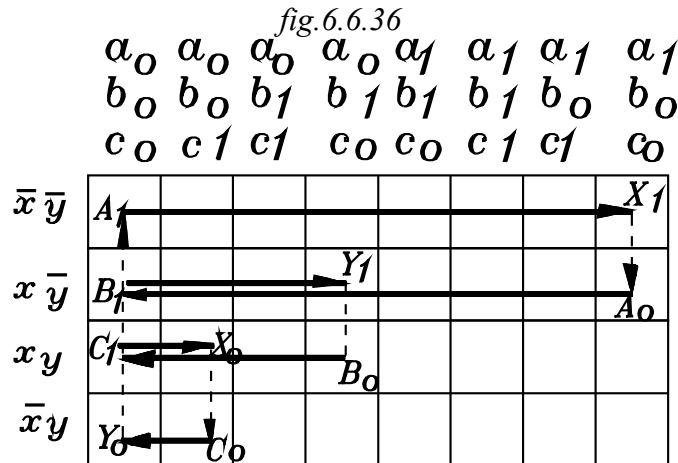
fig.6.6.46

$a_0 a_0 a_0 a_0 a_1 a_1 a_1 a_1$
 $b_0 b_0 b_1 b_1 b_1 b_1 b_0 b_0$
 $c_0 c_1 c_1 c_0 c_0 c_1 c_1 c_0$



Si seguano, in ordine, i comandi rappresentati nella mappa di Karnaugh, riportata in fig.6.6.36

Nelle figure da fig.6.6.37 a fig.6.6.46, in corrispondenza di ogni variabile attiva, sono tratteggiate tutte le caselle che la contengono.



Stato A^+ fig. 6.6.37
Viene comandato da A_1 .

- 1- Quale ordine precede il comando?
Lo precede Y_0 .
- 2- Dallo stato comandato precedente all'attuale qual'è la variabile che ha cambiato stato e quale è la sua forma attuale?
Dal precedente al comando attuale si ha il cambiamento di stato:

$$\bar{x}y a_0 b_0 c_0 \rightarrow \bar{x}\bar{y} a_0 b_0 c_0$$

La variabile che muta è \bar{y} : essa rappresenta quindi la "variabile attiva"

- 3- Nell'insieme delle caselle rappresentanti la variabile attiva \bar{y} (zona tratteggiata di fig.6.6.37) sono presenti i comandi opposti A_1, A_0 ?

Nella zona tratteggiata di fig.6.6.37 si notano i comandi opposti A_1, A_0 di fuoriuscita e rientro dello stelo del cilindro A.

Si pone in AND alla variabile attiva \bar{y} il segnale di memoria \bar{x} interessante la casella contenente il comando A_1

Inoltre per iniziare il ciclo occorre un segnale di Start che verrà posto in AND con le precedenti variabili.

L'equazione di A^+ è:

$$A^+ = \bar{x} \cdot \bar{y} \cdot Start$$

Segnale X_1 di Settaggio della memoria X (fig.6.6.38)

- X_1 commuta la memoria dallo stato \bar{x} allo stato x .
- Precede X_1 il comando A_1 .
- Dal precedente al comando attuale le variabili presentano la variazione:

$$\bar{x}\bar{y} a_0 b_0 c_0 \rightarrow \bar{x}\bar{y} a_1 b_0 c_0$$

- La variabile che muta è a_1 : essa è la variabile attiva.
- Nella zona di a_1 , non vi sono i comandi opposti $X_1 X_0$.
- L'equazione di X_1 è data dalla sola variabile attiva a_1 :

$$X_1 = a_1$$

Stato comandato A^- (fig. 6.6.39)

- Viene comandato da A_0 .
- Precede A_0 il settaggio di memoria X_1 .
- Dal precedente al comando attuale le variabili presentano la variazione:

$$\bar{x} \bar{y} a_1 b_0 c_0 \rightarrow x \bar{y} a_1 b_0 c_0$$

- La variabile che muta è x : essa è la variabile attiva
- Nella zona interessante la variabile x non vi sono i comandi opposti $A_0 A_1$.
- L'equazione logica è data dalla sola variabile attiva x .

$$A^- = x$$

Stato comandato B^+ (fig. 6.6.40)

- Viene comandato da B_1 .
- Precede B_1 il comando A_0 .
- Dal precedente al comando attuale le variabili presentano la variazione:

$$x \bar{y} a_1 b_0 c_0 \rightarrow x \bar{y} a_0 b_0 c_0$$

- La variabile che muta è a_0 : essa è la variabile attiva
 - Nella zona interessante il fine-corsa a_0 vi sono i comandi opposti: B_1 e B_0 .
 - Si pone in *AND* con a_0 la combinazione delle variabili di memoria $x \bar{y}$ corrispondente alla casella ove è scritto il comando B_1 .
- L'equazione di B^+ è:

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

Segnale di Settaggio Y_1 della memoria Y . (fig. 6.6.41)

- Y_1 commuta la memoria dallo stato \bar{y} allo stato y .
- Precede Y_1 il comando B_1 .
- Dal precedente al comando attuale le variabili presentano la variazione:

$$x \bar{y} a_0 b_0 c_0 \rightarrow x \bar{y} a_0 b_1 c_0$$

- La variabile che muta è b_1 : essa è la variabile attiva.
- Nella zona interessante il fine-corsa b_1 non vi sono i comandi opposti $Y_1 Y_0$.
- L'equazione logica di Y_1 è data dalla sola variabile attiva b_1 .

$$Y_1 = b_1$$

Stato comandato B^- (Fig. 6.6.42)

- Viene comandato da B_0 .
- Precede B_0 il comando Y_1 .
- Dal precedente al comando attuale le variabili presentano la variazione:

$$x \bar{y} a_0 b_1 c_0 \rightarrow x y a_0 b_1 c_0$$

- La variabile che muta è y
- Nella zona interessante la variabile y non vi sono i comandi opposti $B_0 B_1$

- L'equazione logica di B^- è data dalla sola variabile attiva y .

$$B^- = y$$

Stato comandato C^+ (fig. 6.6.43)

- Viene comandato da C_I .
- Precede C_I il comando B_O .
- Dal precedente al comando attuale le variabili presentano la variazione:

$$xya_0 b_1 c_0 \rightarrow xy a_0 b_0 c_0$$

- La variabile che muta è b_0 : essa è la variabile attiva
- Nella zona interessante il fine-corsa b_0 vi sono i comandi opposti: C_I e C_O .
- Si pone in *AND* con b_0 le combinazioni delle variabili di memoria xy corrispondente alla casella ove è scritto il comando C_I .
- L'equazione logica di C^+ è data dalla variabile attiva b_0 posta in *AND* con la combinazione dei segnali di memoria xy .

$$C^+ = b_0 \cdot x \cdot y$$

Segnale di Resettaggio X_0 della memoria X (Fig. 6.6.44)

- X_0 commuta la memoria dallo stato x allo stato \bar{x} .
- Precede X_0 il comando C_I .
- Dal precedente al comando attuale le variabili presentano la variazione:

$$xya_0 b_0 c_0 \rightarrow xy a_0 b_0 c_1$$

- La variabile che muta è c_1 : essa è la variabile attiva.
- Nella zona interessante il fine-corsa c_1 non vi sono i comandi opposti X_I X_O .
- L'equazione logica di X_0 è data dalla sola variabile attiva c_1 .

$$X_0 = c_1$$

Stato comandato C^- (fig. 6.6.45)

- Viene comandato da C_O .
- Precede C_O il comando X_0 .
- Dal precedente al comando attuale le variabili presentano la variazione:

$$xya_0 b_0 c_1 \rightarrow \bar{x}y a_0 b_0 c_1$$

- La variabile che muta è \bar{x} : essa è la variabile attiva.
- Nella zona interessante la variabile \bar{x} non vi sono i comandi opposti C_O C_I .
- L'equazione logica di C^- è data dalla sola variabile attiva \bar{x} .

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

Segnale di Resettaggio Y_0 della memoria Y (Fig. 6.6.46)

- Y_0 commuta la memoria dallo stato y allo stato \bar{y} .

- Precede Y_0 il comando C_0 .
- Dal precedente al comando attuale le variabili presentano la variazione:

$$\bar{x}y a_0 b_0 c_1 \rightarrow \bar{x}y a_0 b_0 c_0$$

- La variabile che muta è c_0 : essa è la variabile attiva
- Nella zona interessante il fine-corsa c_0 vi sono i comandi opposti $Y_1 Y_0$.
- Si pone in *AND* con c_0 la combinazione dei segnali di memoria: $\bar{x}y$, corrispondenti alla casella contenente il comando Y_0 .
- L'equazione logica di Y_0 è:

$$Y_0 = c_0 \cdot \bar{x} \cdot y$$

Riassumendo le equazioni logiche si ha:

$$\begin{array}{ll}
 A^+ = \bar{x} \cdot \bar{y} \cdot Start & B^- = y \\
 X_1 = a_1 & C^+ = b_0 \cdot x \cdot y \\
 A^- = x & X_0 = c_1 \\
 B^+ = a_0 \cdot x \cdot \bar{y} & C^- = \bar{x} \\
 Y_1 = b_1 & Y_0 = c_0 \cdot \bar{x} \cdot y
 \end{array} \quad (6.6.2.2)$$

Dalle equazioni logiche si ricava lo schema pneumatico di *fig.6.6.47*

IN LABORATORIO

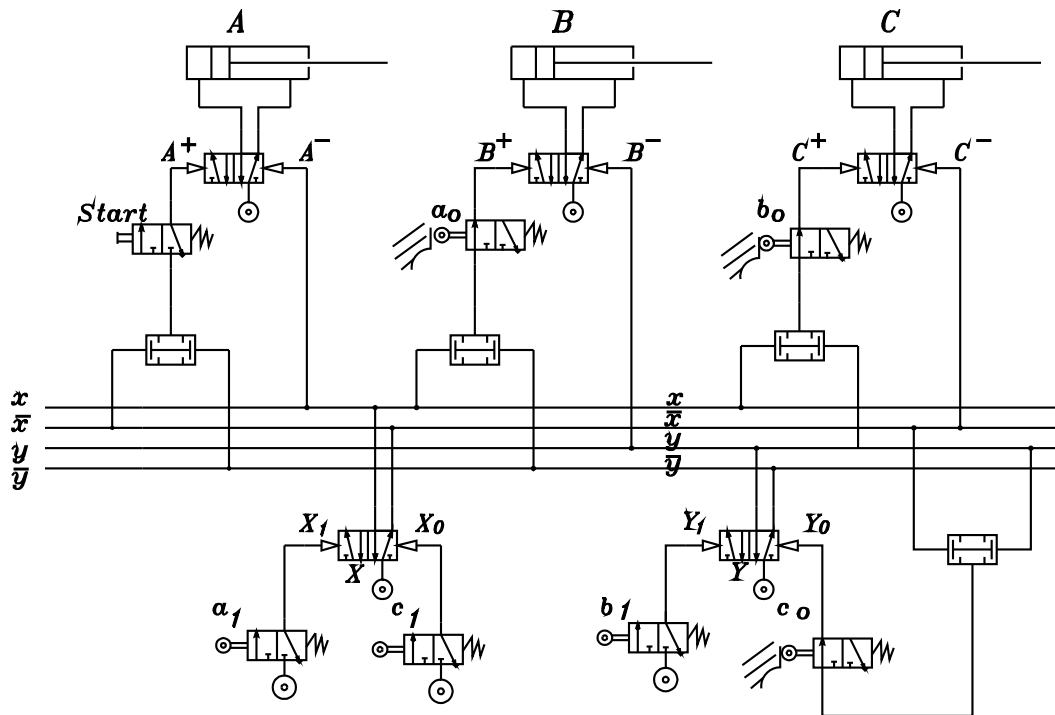
Per fissare le idee conviene effettuare una relazione ove viene scritto il procedimento della determinazione delle equazioni logiche mediante l'uso delle mappe di Karnaugh, e montare poi in laboratorio il circuito.

Note sulla stesura del circuito

- Disegnare i tre cilindri con gli steli nella posizione iniziale: tutti rientrati
- Disegnare l valvole di potenza 5/2 bistabili nella prima posizione in modo che alimentino le camere dei cilindri lato stel.o
- Disegnare le due valvole di memoria X,Y nella posizione iniziale. La valvola X alimenta la linea \bar{x} , la Y la linea \bar{y} .
- Si traducono ora le equazioni logiche in circuito pneumatico.

$$A^+ \quad A^- \quad B^+ \quad B^- \quad C^+ \quad C^-$$

fig.6.6.47



$A^+ = \bar{x} \cdot \bar{y} \cdot Start$ I due ingressi della valvola AND vengono collegati, rispettivamente con le linee \bar{x} e \bar{y} . L'aria in uscita dall'AND va all'ingresso dello Start (valvola 3/2 NC monostabile), e, da questo al pilotaggio A^+ .

$X_1 = a_1$ Si collega l'uscita del finecorsa a_1 (non toccato e alimentato) con il pilotaggio X_1 della valvola di memoria X.

$A^- = x$ L'aria, derivata dalla linea x è inviata con un tubicino al pilotaggio A^- .

$B^+ = a_0 \cdot x \cdot \bar{y}$ I due ingressi della valvola AND vengono collegati, rispettivamente con le linee x e \bar{y} . L'aria in uscita dall'AND va all'ingresso del finecorsa a_0 , e, da questo, al pilotaggio B^+ .

$Y_1 = b_1$ Si collega l'uscita del finecorsa b_1 (non toccato e alimentato) con il pilotaggio Y_1 della valvola di memoria Y.

$B^- = y$ L'aria, derivata dalla linea y è inviata con un tubicino al pilotaggio B^- .

$C^+ = b_0 \cdot x \cdot y$ I due ingressi della valvola AND vengono collegati, rispettivamente con le linee x e y . L'aria in uscita dall'AND va all'ingresso del finecorsa b_0 , e, da questo, al pilotaggio C^+ .

$X_0 = c_1$ Si collega l'uscita del finecorsa c_1 (non toccato e alimentato) con il pilotaggio X_0 della valvola di memoria X.

$C^- = \bar{x}$ L'aria, derivata dalla linea \bar{x} è inviata con un tubicino al pilotaggio C^- .

$Y_0 = c_0 \cdot \bar{x} \cdot y$ *I due ingressi della valvola AND vengono collegati, rispettivamente con le linee \bar{x} e y .
L'aria in uscita dall'AND va all'ingresso del finecorsa c_0 , e, da questo, al pilotaggio C^+ .*

Diagramma di flusso a barre orizzontali

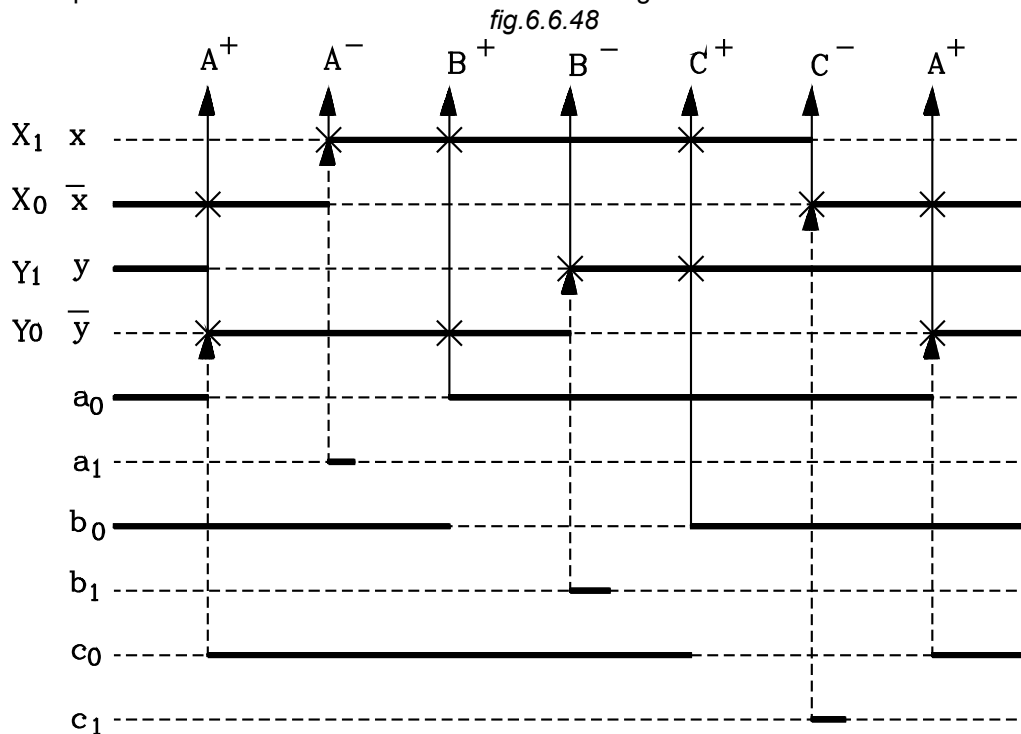
Nel diagramma vengono rappresentate oltre agli stati dei finecorsa anche quelli delle uscite delle memorie \bar{x} , x , \bar{y} , y .

I segnali di memoria che compongono l'equazione di comando sono contrassegnati con una croce (X).

Occorre rammentare che il finecorsa costituente la variabile attiva è attivato alla fine dello stato comandato: all'inizio di quello successivo. Così a_1 è attivato quando viene attuato lo stato A^+ , all'inizio delle fase di comando A^- .

Inoltre occorre ricordare che un finecorsa, attivato all'attuazione di uno stato comandato è disattivato quando viene impartito il comando opposto. Così il finecorsa a_1 attivato all'attuazione di A^+ si disattiva nel comando opposto A^- .

Fatta questa precisazione si dà una breve descrizione del diagramma di flusso a barre orizzontali.



Settaggio e resettaggio delle memorie

Conviene rappresentare prima lo stato delle memorie attivate dai finecorsa, in quanto i comandi sono composti o da un solo segnale di memoria o da una combinazione di questi con uno dei finecorsa.

Memoria x $\begin{cases} X_1 = & \text{Il finecorsa } a_1 \text{ setta l'uscita } x \text{ e porta in scarico } \bar{x}. \text{ Il finecorsa } a_1 \text{ viene attivato} \\ X_0 = & \text{all'attuazione del comando } A^+ \text{ (inizio } A^- \text{) e disattivato un istante dopo il comando} \\ & \text{dello stato opposto } A^- \text{.} \end{cases}$

Il finecorsa c_1 setta l'uscita \bar{x} e porta in scarico x . Il finecorsa c_1 viene attivato all'attuazione del comando C^+ (inizio C^-) e disattivato un istante dopo il comando dello stato opposto C^- .

Memoria x $\begin{cases} Y_1 = b_1 & \text{Il finecorsa } b_1 \text{ setta l'uscita } y \text{ e porta in scarico } \bar{y}. \text{ Il finecorsa } b_1 \text{ viene} \\ X_0 = c_0 \cdot \bar{x} & \text{attivato all'attuazione del comando } B^+ \text{ (inizio } B^- \text{) e disattivato un istante} \\ & \text{dopo il comando dello stato opposto } B^- \text{.} \end{cases}$

Il finecorsa C_0 , in AND con i segnali di memoria $\bar{x} \cdot y$ setta l'uscita \bar{y} e porta in scarico y . Il finecorsa C_0 viene attivato all'attuazione del comando C^- (inizio A^+) e disattivato quando viene comandato lo stato opposto C^- .

Rappresentazione dei comandi

A^+ Viene comandato dai due segnali di memoria $\bar{x} \cdot \bar{y}$ in AND con lo start non indicato nel diagramma.

$$A^+ = \bar{x} \cdot \bar{y} \cdot Start$$

Sono poste due croci in corrispondenza di \bar{x} e \bar{y} .

Occorre notare che quando viene impartito il comando dello stato A^+ è già stata settata da C_0 l'uscita di memoria \bar{y} , alla fine della fase precedente.

A^- È comandato dal segnale di memoria x .

$$A^- = x$$

L'uscita x è già stata settata dal finecorsa a_1 attivato alla fine della fase precedente.

B^+ È comandato dal finecorsa a_0 in AND con i segnali di memoria $x \cdot \bar{y}$.

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

Sono poste due croci in corrispondenza di x e \bar{y} .

B^- È comandato dal segnale di memoria y .

$$B^- = y$$

L'uscita y è già stata settata dal finecorsa b_1 attivato alla fine della fase precedente.

C^+ È comandato dal finecorsa b_0 in AND con i segnali di memoria $x \cdot y$.

$$C^+ = b_0 \cdot x \cdot y$$

Sono poste due croci in corrispondenza di x e y .

C^- È comandato dal segnale di memoria \bar{x} .

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

L'uscita \bar{x} è già stata settata dal finecorsa c_1 attivato alla fine della fase precedente.

Funzionamento del circuito pneumatico

- Nelle condizioni iniziali l'aria alimenta le linee \bar{x} , \bar{y} . Lo stelo del cilindro C è in posizione rientrata, spinto dall'aria derivata dalla linea \bar{x} .
- Nelle condizioni iniziali occorre porre le valvole di potenza bistabili, alimentanti i cilindri A , B nella posizione di commutazione che alimentano le camere lato stelo, in modo che questi risultino rientrati.
- Pigiato lo Start, l'aria, proveniente dalle linee \bar{x} , \bar{y} e in uscita dalla valvola AND, passa attraverso lo Start stesso ed è inviata al pilotaggio A^+ . Si ha la fuoriuscita dello stelo di A .
- Toccato il finecorsa a_1 , questo invia l'aria al pilotaggio X_1 della valvola di memoria X , la quale commutandosi alimenta la linea x portando \bar{x} in scarico.
- L'aria derivata dalla linea x va al pilotaggio A^- .

Si ottiene la corsa di rientro dello stelo di A. Il pilotaggio opposto A^+ non è attivato essendo ora la linea \bar{x} in scarico e manca aria ad un ingresso della valvola AND.

- Toccato il finecorsa a_0 , l'aria, in uscita dalla valvola AND e proveniente dalla due linee alimentate x , \bar{y} e attraverso il finecorsa a_0 stesso va al pilotaggio B^+ .
Si ha la fuoriuscita dello stelo del cilindro B. Il pilotaggio opposto non è attivato essendo la linea y in scarico.
- Toccato il finecorsa b_1 questo invia l'aria la pilotaggio Y_1 della valvola di memoria Y, la quale commutandosi alimenta la linea y portando \bar{y} in scarico.
- L'aria derivata dalla linea y va al pilotaggio B^- .
Si ottiene la corsa di rientro dello stelo di B. Il pilotaggio opposto B^+ non è attivato essendo ora la linea \bar{y} in scarico e manca aria ad un ingresso della valvola AND.
- Toccato il finecorsa b_0 , l'aria, in uscita dalla valvola AND e proveniente dalla due linee alimentate x , y e attraverso il finecorsa b_0 stesso va al pilotaggio C^+ .
Si ha la fuoriuscita dello stelo del cilindro C. Il pilotaggio opposto non è attivato essendo la linea \bar{x} in scarico.
- Toccato il finecorsa c_1 questo invia l'aria la pilotaggio X_0 della valvola di memoria X, la quale commutandosi alimenta la linea \bar{x} portando x in scarico.
- L'aria derivata dalla linea \bar{x} va al pilotaggio C^- .
Si ottiene la corsa di rientro dello stelo di C. Il pilotaggio opposto C^+ non è attivato essendo ora la linea x in scarico e manca aria ad un ingresso della valvola AND.
- Toccato il finecorsa c_0 , l'aria, proveniente dalle linee \bar{x} , y e in uscita dalla valvola AND, passa attraverso il finecorsa c_0 , ed è inviata la pilotaggio Y_0 della valvola di memoria Y commutandosi alimenta la linea \bar{y} portando y in scarico.
Si ripristinano così le condizioni iniziali del ciclo.



Avanti...

[Clic per continuare](#)


Indietro...

[Clic per precedente](#)


Indietro...

[Clic per la pagina iniziale](#)