

# Sistemi di controllo industriali per applicazioni meccatroniche

# Il sistema binario

## SISTEMA DECIMALE

Un qualunque numero del sistema di numerazione decimale si può scrivere nel modo seguente:

$$N = K_n x^n + K_{n-1} x^{n-1} + \dots + K_0 x^0 + K_{-1} x^{-1} + K_{-2} x^{-2} + \dots + K_{-m} x^{-m} \quad (1)$$

dove la base  $x$  vale 10, mentre i coefficienti  $K_i$  possono assumere i 10 simboli ammessi da questo sistema, vale a dire: 0, 1, 2 ..., 8, 9. Il valore dell'esponente  $n$  è dato dal numero di cifre a sinistra della virgola meno uno. Per esempio, il numero 987,6 si può scrivere:

$$987,6_{(10)} = 9 \times 10^2 + 8 \times 10^1 + 7 \times 10^0 + 6 \times 10^{-1}$$

Alle potenze in base 10 si dà il nome di pesi.

## SISTEMA BINARIO

Il sistema di numerazione decimale non si adatta ai moderni elaboratori elettronici in quanto l'esecuzione delle operazioni risulterebbe molto laboriosa.

Viene invece adoperato il sistema di numerazione binario che si avvale di due soli segni, 0 e 1, i quali sono usualmente denominati bit (Binary digit). Le cifre dei numeri binari vengono posizionate dai relativi pesi, che naturalmente sono potenze in base due. Ad esempio il numero binario 11010 (che si legge: uno uno zero uno zero) si può esprimere nella maniera seguente:

$$11010_{(2)} = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

Si è già accennato al fatto che negli elaboratori elettronici viene adoperato il sistema di numerazione binario. Infatti esso presenta i seguenti vantaggi principali:

- è rappresentabile con dispositivi che possono assumere uno fra due soli "stati": contatti aperti/chiusi, lampadine accese/spente, transistori in conduzione/saturazione ecc.

- consente una drastica riduzione di circuiti: basti pensare ad es. che in codice decimale occorrono  $2 \times 10 = 20$  lampadine per rappresentare i numeri 0 + 99 mentre in BP con 7 lampadine si possono rappresentare i numeri da 0 a 127 (infatti le combinazioni di stati possibili sono  $2^7 = 128$  e quindi il numero massimo rappresentabile è:  $2^7 - 1 = 127$ ).

# Algebra di Boole

## LOGICA DELLE PROPOSIZIONI

Una frase logica o di senso compiuto del comune linguaggio può essere considerata composta di affermazioni o proposizioni semplici, vere o false, variamente congiunte tra loro. Quindi queste proposizioni semplici possono essere considerate grandezze binarie (vero = 1, falso = 0) e come tali trattate algebricamente.

Esaminiamo la frase: "Se la macchina è disponibile ed è funzionante, si può lavorare"; essa può essere rappresentata nel modo seguente:

F = rappresenta l'intera frase che si realizza nel "si può lavorare"

A = la macchina è disponibile

B = la macchina è funzionante

· = simbolo della congiunzione "e" (AND) che lega le due proposizioni A e B.

Evidentemente le proposizioni A e B possono essere vere (1) o false (0), però la relazione scritta sopra esprime il fatto che la frase completa è vera solo quando A e B sono vere congiuntamente.

Se con B abbiamo rappresentato l'affermazione: "la macchina è funzionante", chiamiamo complementare l'affermazione contraria: "la macchina non è funzionante" e la indichiamo con  $\bar{B}$  (B negato o NOT B). La proposizione complementata è, a sua volta, vera o falsa, quindi

per  $B = 1$        $\bar{B} = 0$

per  $B = 0$        $\bar{B} = 1$

Esaminiamo ora la frase: "Se aziono l'interruttore diretto oppure quello generale, la macchina si disalimenta". Essa può essere così rappresentata:

F = rappresenta l'intera frase che si realizza nel "la macchina si disalimenta"

A = aziono l'interruttore diretto

B = aziono l'interruttore generale

+ = simbolo della congiunzione "oppure" (OR) che lega le due proposizioni A e B

$F = A + B$

# Algebra di Boole (segue)

dove le proposizioni A e B possono essere vere (1) o false (0), ma la relazione scritta sopra significa che la frase completa è vera quando è vera A oppure è vera B oppure entrambe sono vere. Esaminiamo infine la frase più complessa: "la macchina si muove (Av) quando è presente il segnale di posizionamento (Gp) e il consenso che non è bloccata (C1) oppure quando è presente il segnale di funzionamento manuale (Man) e il consenso di direzione voluta (D).

Per quanto detto in precedenza si può rappresentare questo discorso con la seguente relazione:

$$Av = Gp \cdot \overline{C1} + Man \cdot D$$

## OPERAZIONI BOOLEANE FONDAMENTALI

### AND o prodotto logico

Si ha quando la congiunzione "e" (·) lega più variabili in modo che la funzione sia vera (1) soltanto se tutte le variabili sono vere (1).

Rappresentiamo questa operazione mediante una lampada F che è alimentata attraverso due interruttori A e B secondo lo schema di Fig 3.1. La lampada si accende solo se i due interruttori sono chiusi congiuntamente; in altre parole la lampada F è accesa se si chiudono gli interruttori A e (AND) B. Quando detto si può scrivere:

$$A \cdot B = F$$

dove F è la funzione "accensione" che si vuole determinare ed A e B sono le variabili o stato degli interruttori A e B. Se poniamo:

$$\text{interruttore chiuso} = 1 \quad \text{lampada accesa} = 1$$

$$\text{interruttore aperto} = 0 \quad \text{lampada spenta} = 0$$

# Algebra di Boole (segue)

dalla relazione  $AB=F$  si possono dedurre i seguenti postulati:

$$1 \cdot 1 = 1$$

che corrisponde al fatto che se si chiudono entrambi gli interruttori ( $A = B = 1$ ) la lampada si accende ( $F = 1$ ). Inoltre

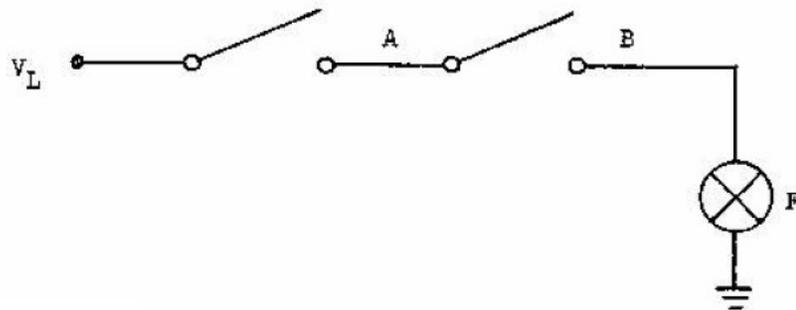
$$1 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

cioè: se si chiude un solo interruttore ( $A = 1$  e  $B = 0$  o viceversa) la lampada si spegne ( $F = 0$ ). Infine

$$0 \cdot 0 = 0$$

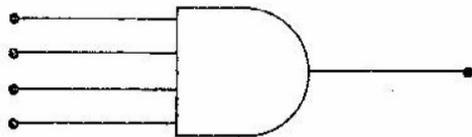
se si aprono entrambi gli interruttori ( $A$  e  $B = 0$ ) la lampada si spegne ( $F = 0$ ).



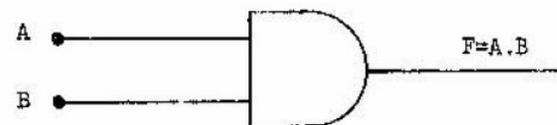
Schema di un circuito AND

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

Tabella della verità di un circuito AND



Simbolo grafico del blocco logico AND

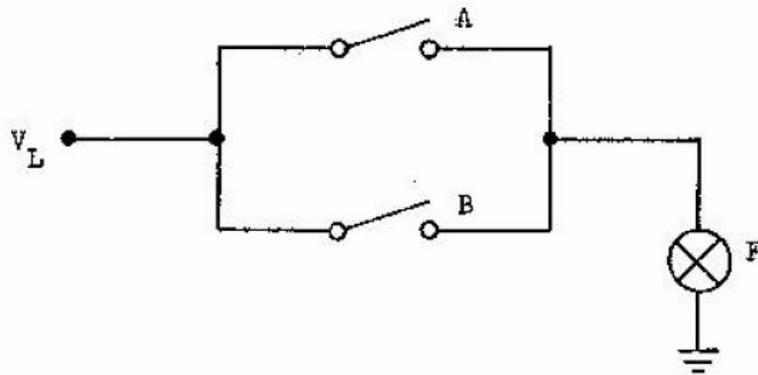


Simbolo grafico dell'espressione booleana  $F = A.B$

# Algebra di Boole (segue)

## OR o somma logica

Si ha quando la congiunzione "o" (+) lega più variabili in modo che la funzione sia vera (1) quando almeno una o entrambe le variabili sono vere (1).



Schema di un circuito OR

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Tabella della verità di un circuito OR

# I microprocessori per l'automazione

- i sistemi di automazione richiedono funzioni di controllo che generalmente sono svolte da sistemi a microprocessore.
- la prevalenza del microprocessore rispetto ad altre forme di controllo è dovuta ad una gamma di fattori:
  - *controllo a programma memorizzato*
  - *elaborazione digitale*
  - *velocità di operazione*
  - *flessibilità di progettazione*
  - *integrazione*
  - *costi*

# Controllo a programma memorizzato

- un *microprocessore* è un elaboratore in grado di eseguire istruzioni da un *programma* in memoria
- il *programma* è una sequenza logica di operazioni, ed è:
  - *deterministico*
  - *ripetibile*
- è l'ideale per esecuzioni di funzioni di controllo e di elaborazioni del segnale basate su algoritmi matematici e logici

# Elaborazione digitale

- nel sistema a microprocessore le informazioni sono rappresentate in forma digitale, da numeri binari
- una informazione digitale:
  - non è danneggiata da rumori analogici
  - ha una risoluzione variabile a seconda delle esigenze delle applicazioni

# Velocità di operazione

- il continuo progresso tecnologico nel campo informatico accresce continuamente la velocità di elaborazione dei microprocessori
- un microprocessore (tipico) esegue le istruzioni in ordine *sequenziale*
- tuttavia, l'estrema velocità fa sì che nel tempo di risposta dell'osservatore (il sistema controllato) sequenze di istruzioni siano eseguite virtualmente simultaneamente

# Flessibilità di progettazione

- un sistema a microprocessore offre al progettista una estrema flessibilità: lo stesso hardware può essere preparato a svolgere un numero elevato di funzioni diverse
- ciò è possibile, nell'ambito dei limiti fisici che caratterizzano l'hardware (risorse di memoria, di input e di output) cambiando il programma esecutivo (software)

# Integrazione

- per un microprocessore, gli ingombri e la alimentazione necessaria per il funzionamento sono molto ridotti
- ciò è dovuto ai progressi nella miniaturizzazione dei componenti elettronici

# Costi

- i costi dei microprocessori continuano a diminuire
- con la diminuzione dei costi il numero di applicazioni attuabili cresce

# Sistemi a microprocessore interno in tempo reale (*real-time*)

I requisiti di una tipica applicazione meccatronica vengono svolti da una classe di sistemi a microprocessore definiti *sistemi a microprocessore in tempo reale*:

- *in tempo reale*: il microprocessore controlla un processo che si svolge nella realtà, nel dominio del tempo.

L'altra grande categoria di sistemi a microprocessore è costituita dai *sistemi a microprocessore per elaborazione dati*, come i normali PC, che non sono correlati direttamente con un processo fisico.

# Confronto tra le due categorie di sistemi a microprocessore

## Applicazioni interne in tempo reale

## Applicazioni di elaborazione dati

### Memoria di sistema

Piccole quantità di memoria non volatile di sola lettura per la memorizzazione del programma e di memoria lettura - scrittura per la memorizzazione dei dati

Grandi quantità di memoria lettura-scrittura, solo una piccola quantità di memoria di programma non volatile usata durante l'inizializzazione del sistema

### Input/ Output

Soprattutto interfacce digitali e analogiche per sensori e attuatori

Soprattutto I/O digitali (tastiera, schermo, hard disk, stampanti, altre periferiche, rete)

### Processore

Vasta gamma di prestazioni per applicazioni differenti; necessaria risposta veloce agli eventi esterni

Necessarie prestazioni elevate per migliorare la produttività dell'utente; il tempo di risposta generalmente non è critico

# Input / Output

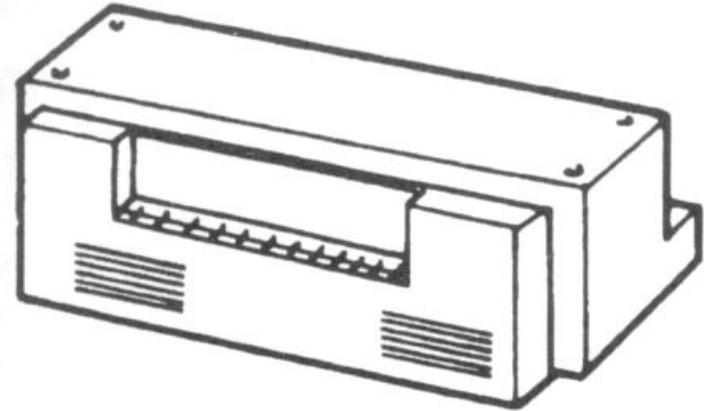
- per un sistema a microprocessore per elaborazione dati i dispositivi di I/O sono fondamentalmente dedicati al colloquio tra uomo e macchina:
  - video
  - tastiera
  - mouse
- per un sistema interno gli I/O sono principalmente interfacce macchina-macchina (*verso sensori e attuatori*); l'interfaccia uomo-macchina è meno importante

# Velocità del processore

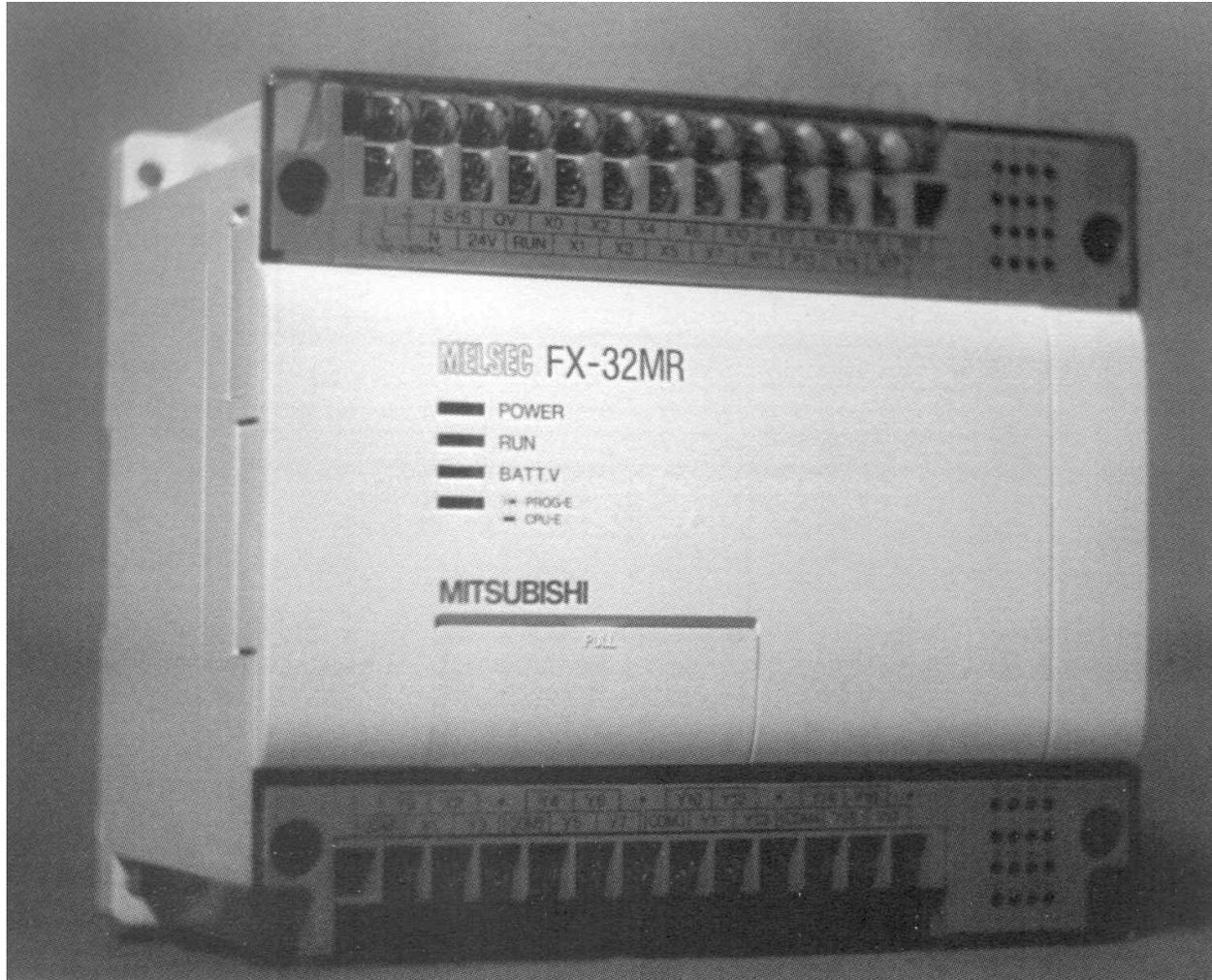
- una delle più importanti caratteristiche del processore è la velocità con cui esegue le istruzioni
- per un sistema di elaborazione dati il *tempo di risposta* non è critico, anche se l'attesa può rallentare la produttività dell'operatore o semplicemente infastidirlo
- per un sistema in tempo reale un tempo di risposta troppo alto può avere conseguenze catastrofiche (si pensi al sistema di controllo di un aeroplano)

# Prima tipologia di sistema a microprocessore interno: *PLC stand-alone*

- *PLC: programmable logic controller*
- sistemi piccoli non modulari
- 8-100 I/O digitali (2 stati)
- semplice programmazione (generalmente *ladder*)
- generalmente sostituiscono semplici circuiti elettronici, introducendo flessibilità e riprogrammabilità

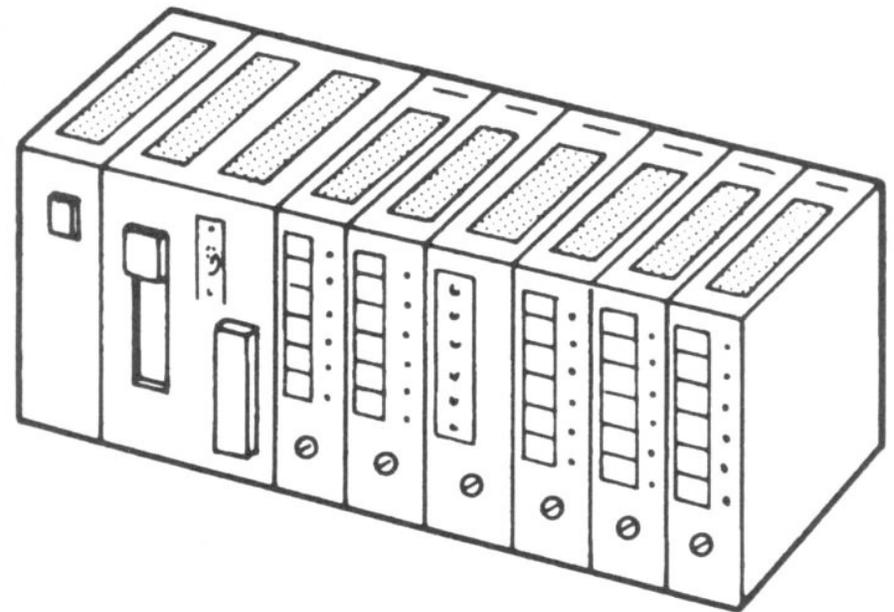


# Esempio di PLC stand-alone



# Seconda tipologia di sistema a microprocessore interno: *PLC modulare*

- sistema di taglia media
- espandibilità modulare
- 32-500 I/O digitali e analogici
- funzioni di programmazione più avanzate

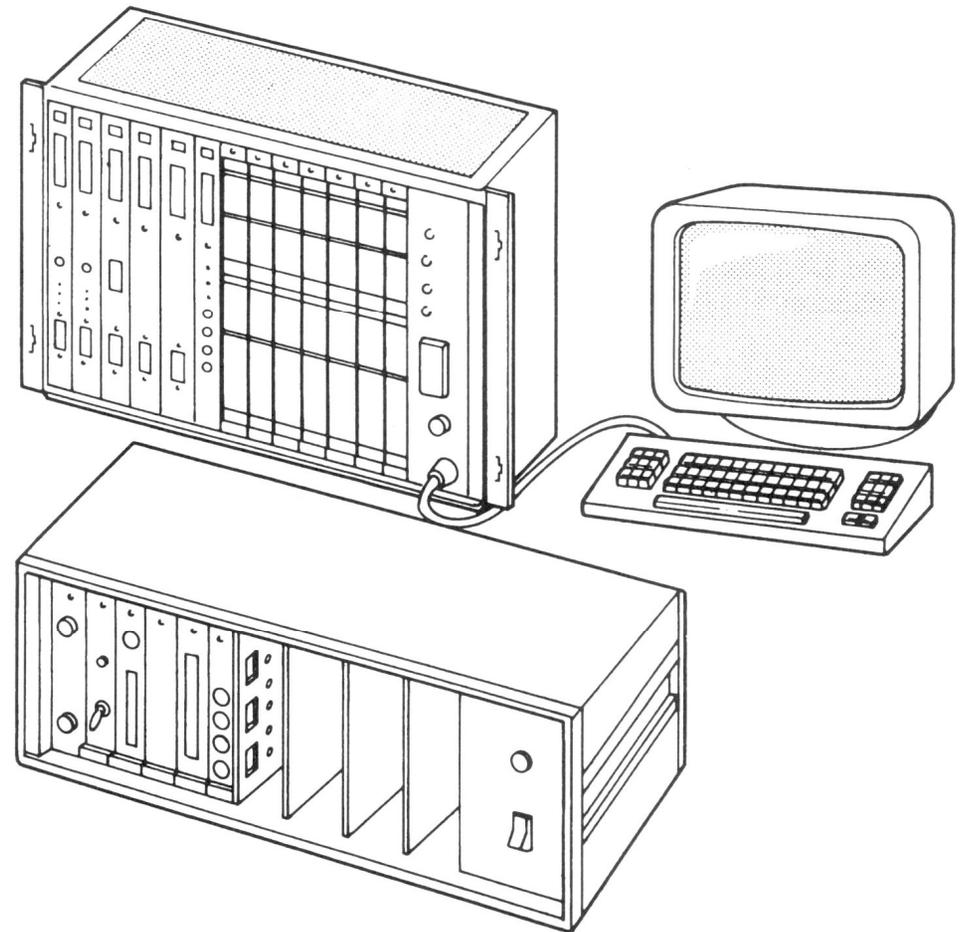


# Esempio di PLC modulare



# Terza tipologia di sistema a microprocessore interno: *sistema basato su microprocessore evoluto*

- sistema di grossa taglia
- ad architettura chiusa o modulare
- anche controllo assi, ovvero regolazione continua del moto di attuatori (es: robot, macchina utensile)
- programmazione evoluta, con linguaggio compilato ad alto livello (Pascal, C, linguaggi proprietari derivati)



# Esempio di PLC per controllo assi

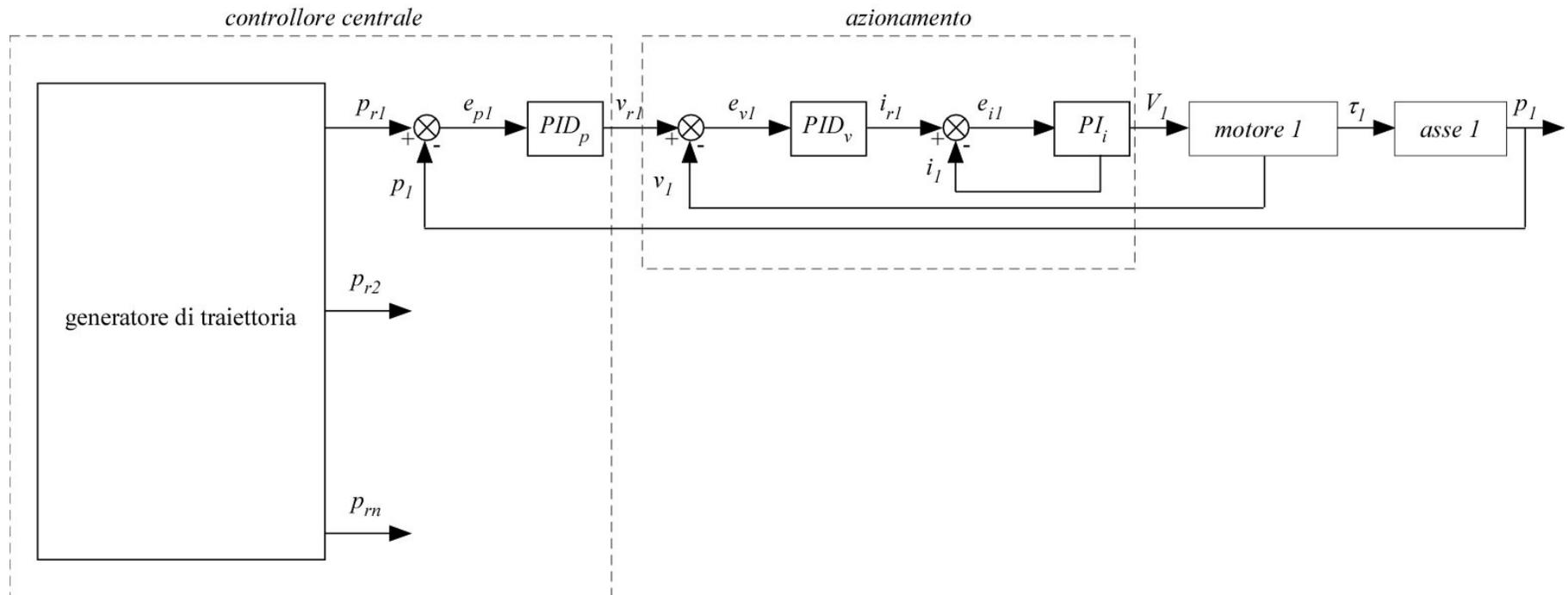


**Nota:** i PLC che effettuano controllo assi sono basati, come gli altri, su un microprocessore a logica digitale che lavora non in *tempo continuo* ma in *tempo discreto*; ovvero il microprocessore legge gli input, effettua i calcoli ed aggiorna lo stato degli output ad intervalli fissi che corrispondono al suo *tempo di campionamento*. Tuttavia, se il tempo di campionamento del controllore è sufficientemente piccolo rispetto alla dinamica del sistema, si può considerare l'intero sistema in ciclo chiuso, composto da una parte «fisica» in tempo continuo e da una «digitale» in tempo discreto, come se fosse interamente in tempo continuo.

*Nella quasi totalità delle applicazioni meccatroniche l'approccio ingegneristico consiste nel selezionare un controllore con tempo di campionamento molto piccolo rispetto alla dinamica del sistema controllato, dopodiché si studia il sistema in ciclo chiuso come se fosse in tempo continuo.*

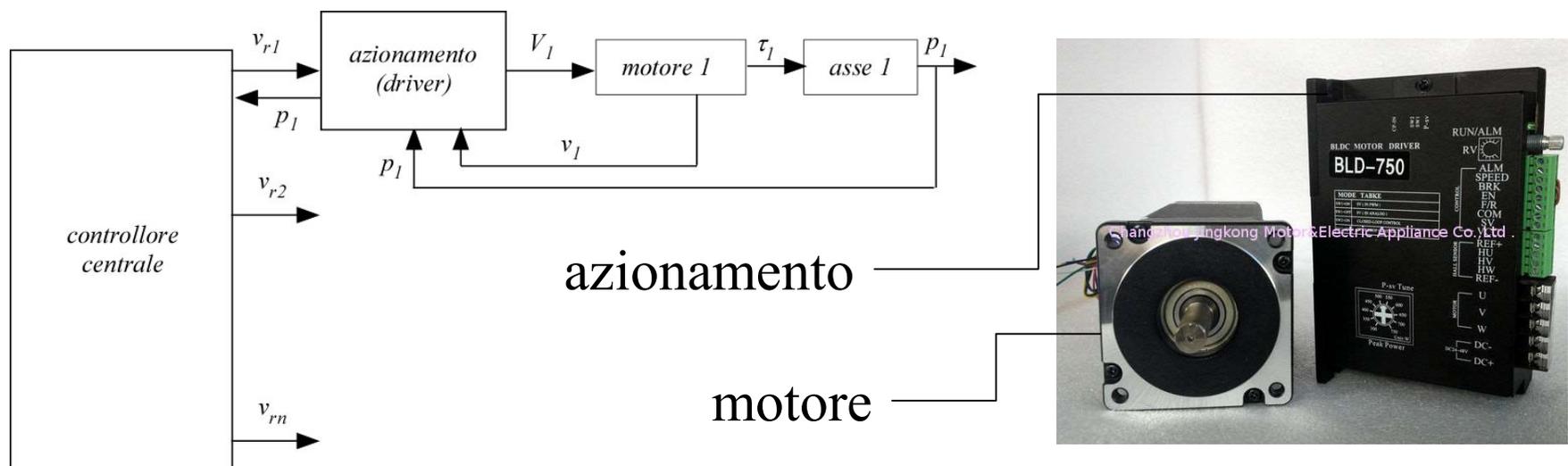
# Schema tipico controllo assi

- lo schema tipico per eseguire il controllo assi è quello di figura
- ci sono tre anelli di controllo uno interno all'altro: posizione, velocità, corrente
- il più esterno è quello di posizione: per ogni asse c'è un riferimento temporale di posizione (lineare o angolare)  $p_{ri}$  per l'asse  $i$ -esimo, calcolato dal generatore di traiettoria interno al *controllore centrale* (il PLC); ovviamente gli assi sono coordinati temporalmente
- sulla base dell'errore di posizione  $e_{pi}$ , tramite il controllo PID di posizione, si origina un riferimento di velocità  $v_{ri}$  che viene inviato, in digitale o in analogico (solitamente  $0 \div 10V$ , oppure  $-10 \div +10 V$ ), all'*azionamento* (*driver*)
- l'azionamento è il dispositivo, collegato alla rete elettrica, che alimenta il motore; a differenza del controllore centrale, che ha solo *elettronica di controllo*, ha anche *elettronica di potenza*



# Schema tipico controllo assi

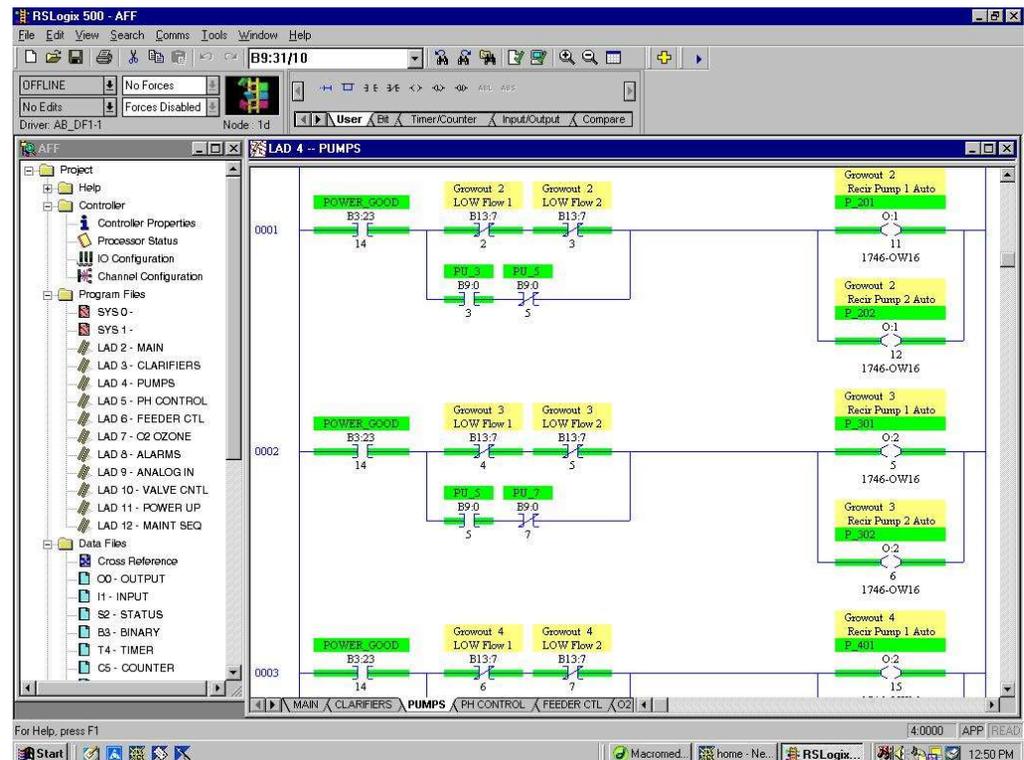
- la velocità del motore  $v_i$  viene misurata o ottenuta per derivazione dalla  $p_i$ ; sulla base dell'errore di velocità  $e_{v_i}$  tramite il controllo PID di velocità si calcola il riferimento di corrente  $i_{r_i}$
- l'anello di corrente (solitamente PI, la componente derivativa non serve) genera la tensione  $V_i$  applicata al motore (i dettagli dipendono da tipo di motore, ci possono essere più fasi) da cui nasce la coppia erogata  $\tau_i$  che muove l'asse
- l'anello di posizione è chiuso dal controllore centrale; gli anelli di velocità e corrente dall'azionamento
- la misura della posizione dell'asse  $p_i$  di solito è effettuata tramite encoder digitali, collegati all'azionamento, il quale oltre ad usarla, ad esempio per gestire fine corsa o calcolare la velocità del motore se non è presente un sensore di velocità, ritrasmette il segnale anche al controllore centrale (*ripetizione encoder*)
- a volte non è presente l'anello intermedio di velocità, e l'anello di posizione origina direttamente un riferimento di corrente (che per ragioni elettrotecniche è proporzionale alla coppia erogata)
- se uno o più assi fossero controllati in coppia anziché in posizione (raro) il controllore centrale genererebbe un riferimento di corrente/coppia anziché di posizione



# Linguaggio *ladder* per la programmazione dei PLC

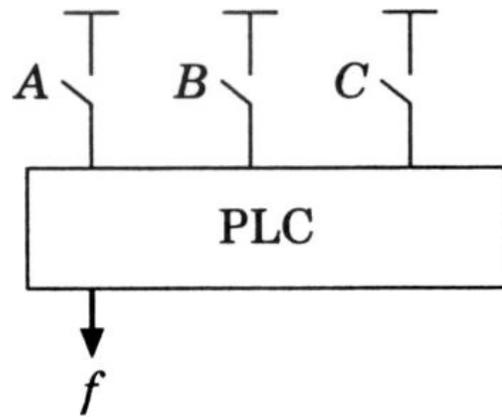
# Linguaggio ladder

- il linguaggio ladder (scala a pioli) è un metodo molto diffuso per la programmazione dei PLC di piccola taglia
- è un linguaggio di tipo grafico, basato sull'analogia con un circuito elettrico
- analizzeremo la logica del linguaggio, senza entrare nel merito del metodo pratico di inserimento del programma, che varia da PLC a PLC
- generalmente si usa un normale PC sul quale è installato un software per lo sviluppo del programma ladder; quindi si compila l'eseguibile e lo si trasferisce sul PLC via USB o seriale

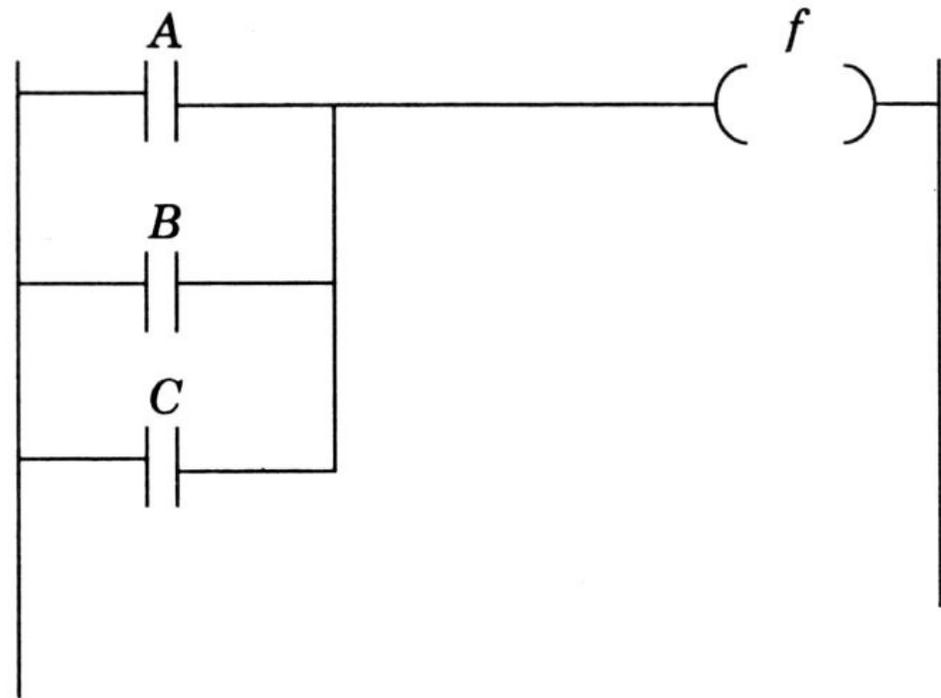


# Funzione OR (interruttori in parallelo)

$$f = A+B+C$$



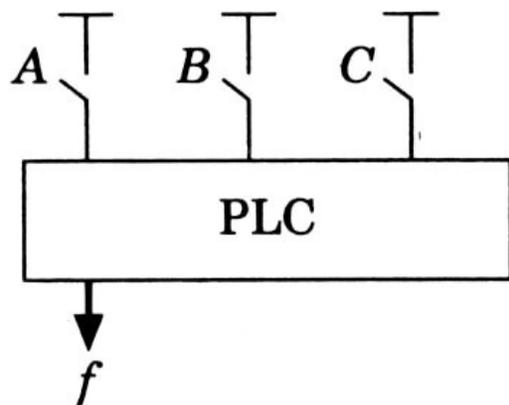
layout del PLC



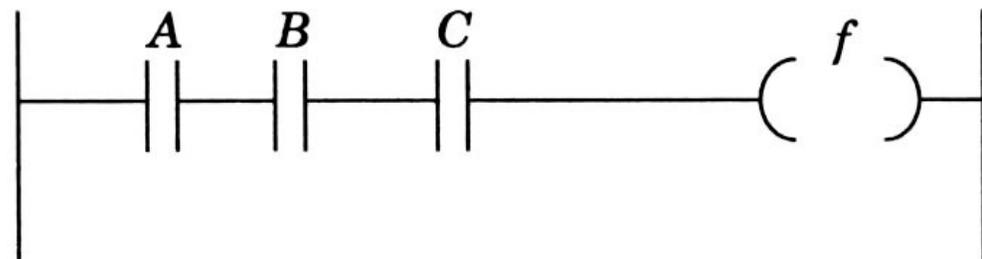
programma

# Funzione AND (interruttori in serie)

$f = ABC$

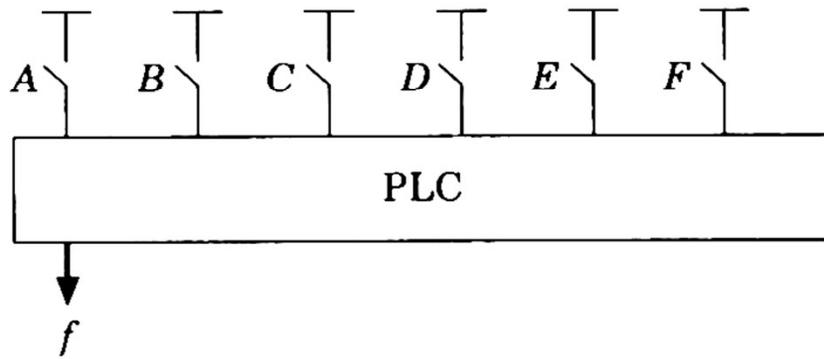


layout del PLC  
(si noti che è uguale al precedente)

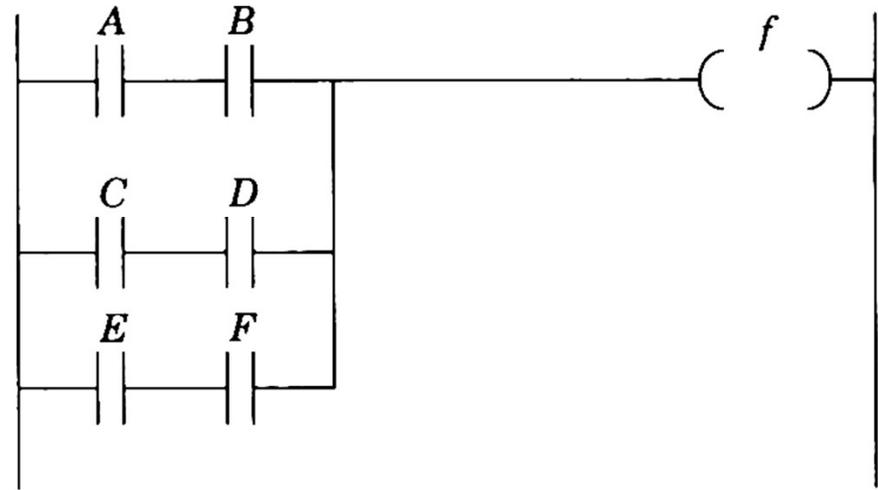


programma

# Funzione AND e OR composte

$$f=AB+CD+EF$$


layout del PLC

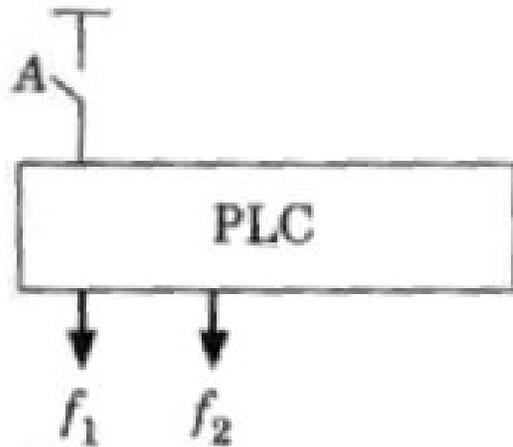


programma

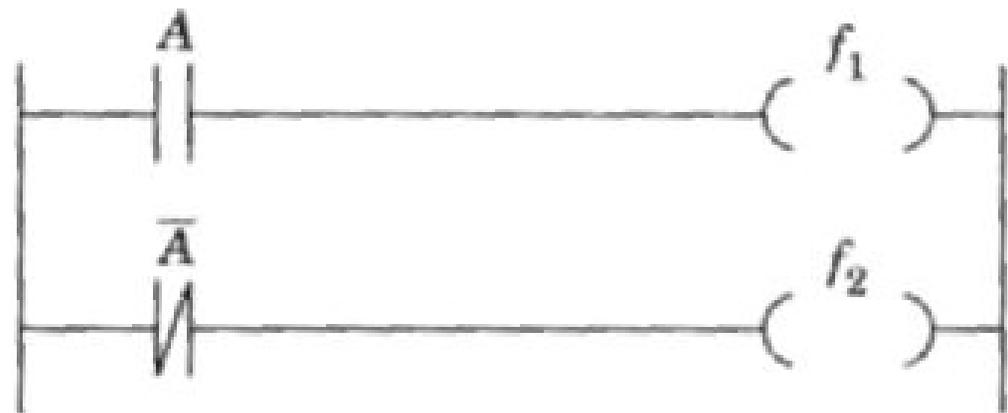
# Negazione

$$f_1 = A$$

$$f_2 = \text{not}(A)$$

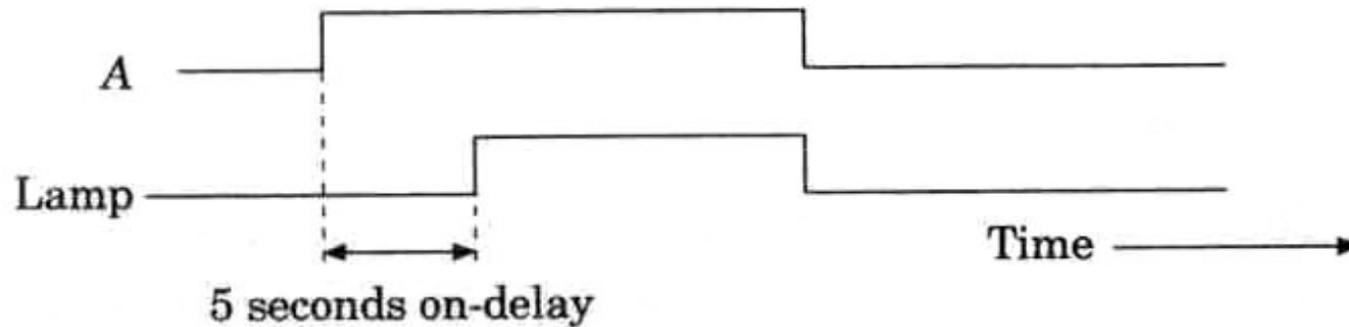
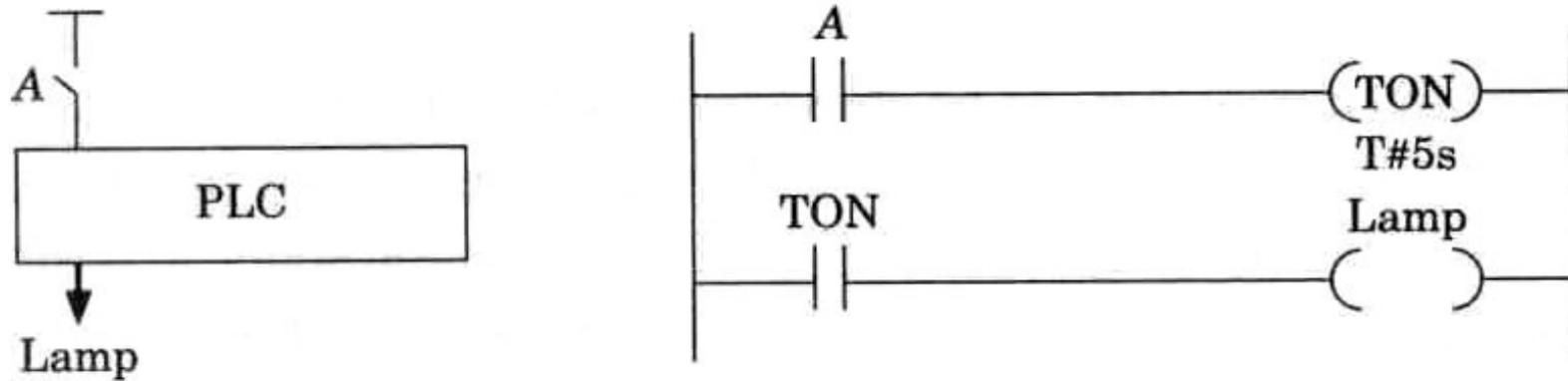


layout del PLC

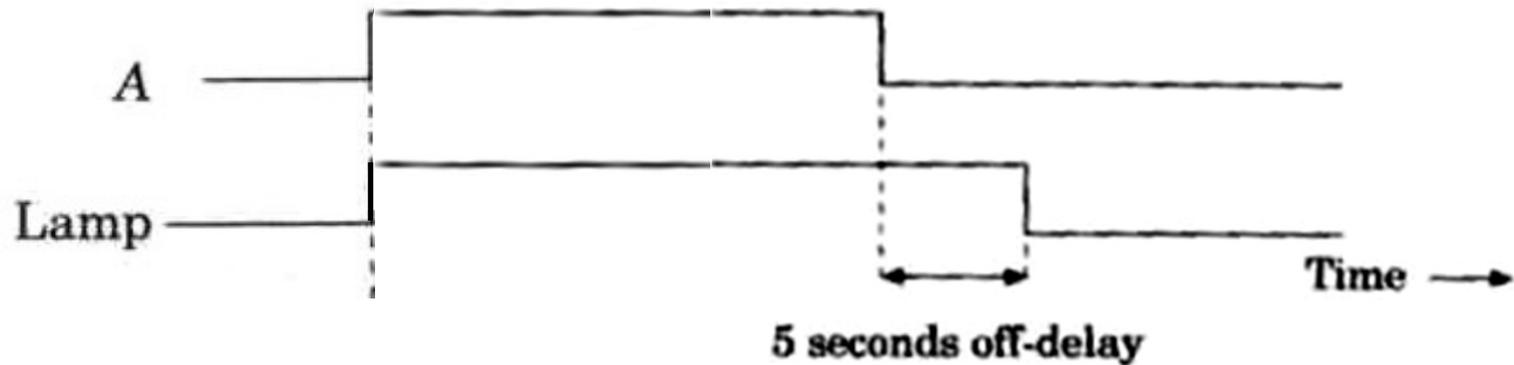
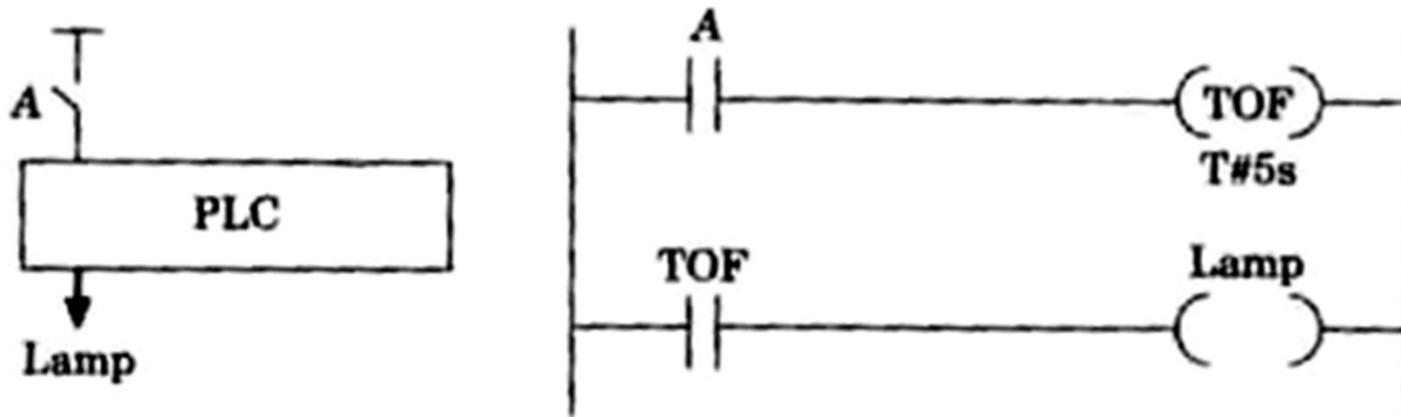


programma

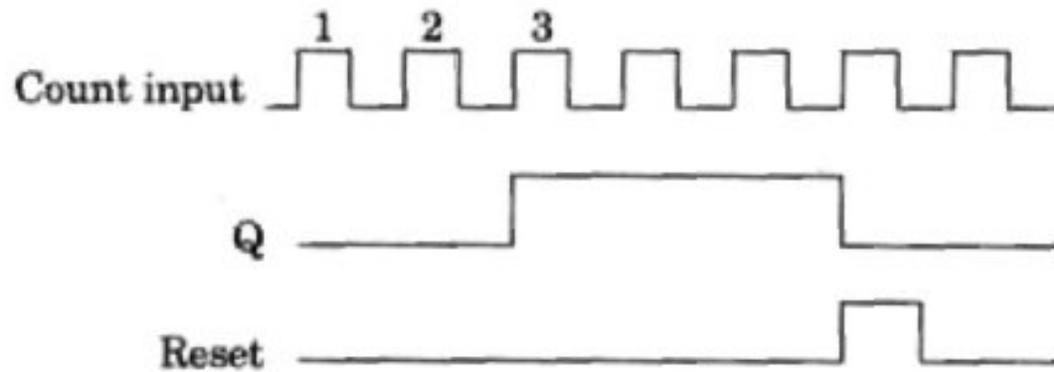
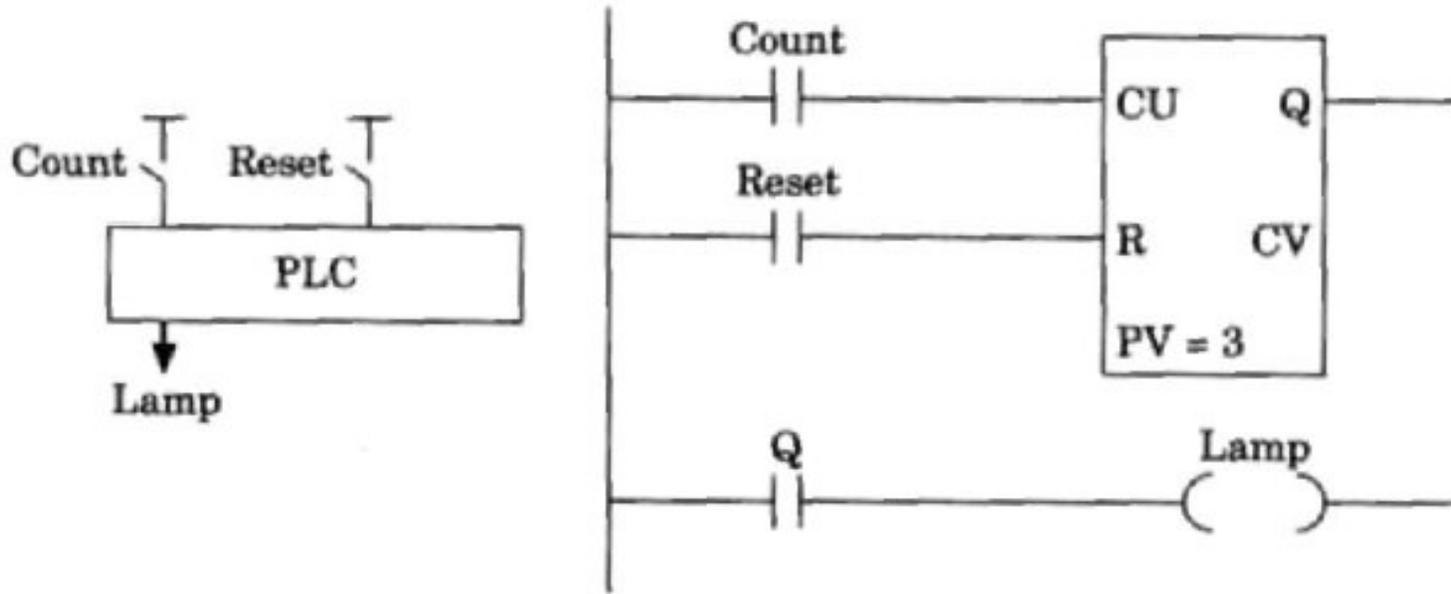
# Temporizzatore (timer) (on-delay, TON)



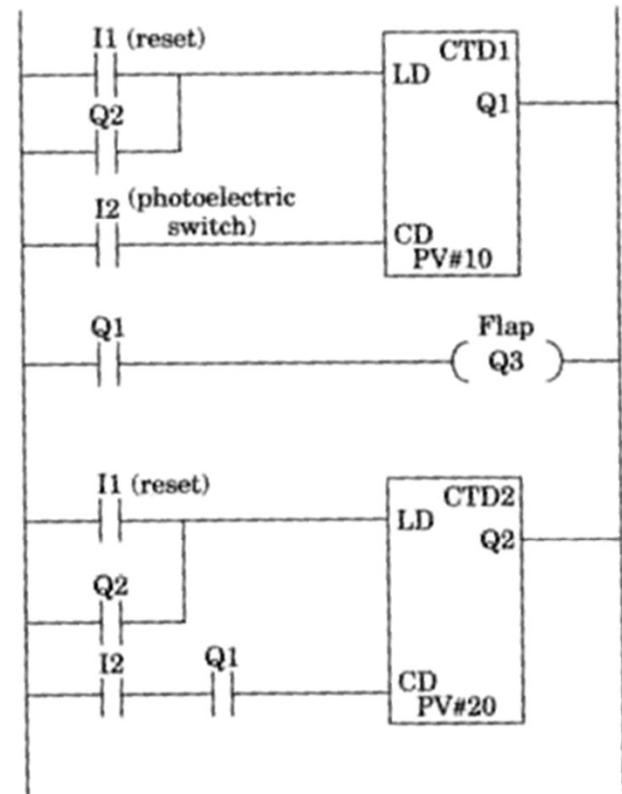
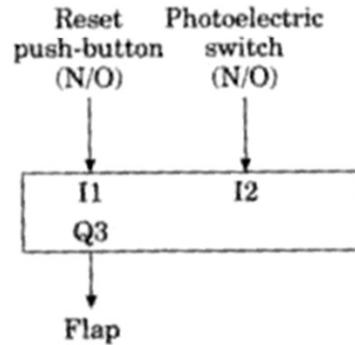
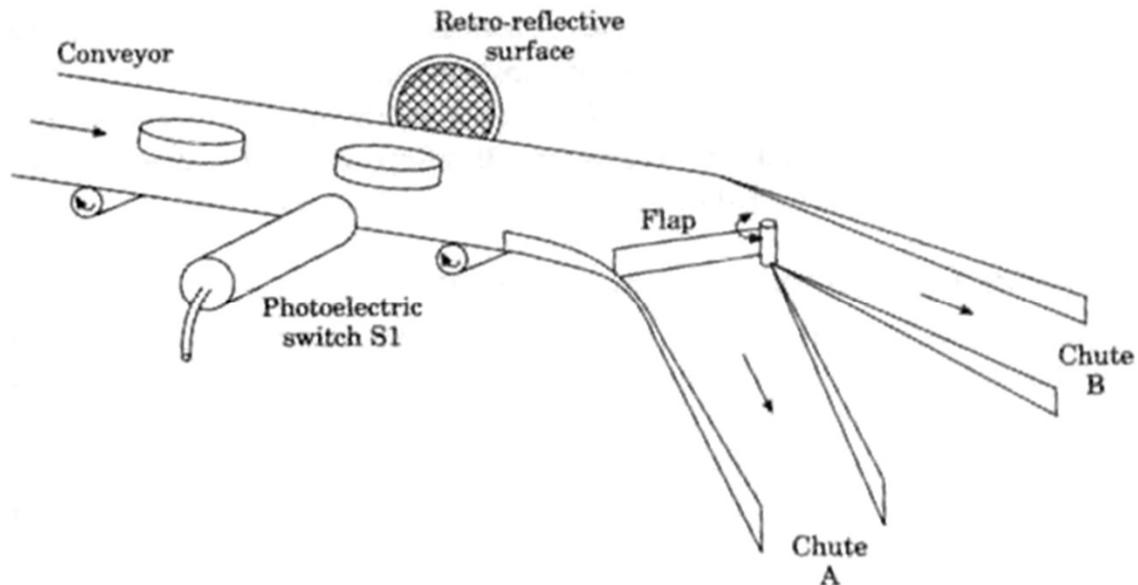
# Temporizzatore (timer) (off-delay, TOF)



# Contatore



# Esempio applicativo: imballaggio



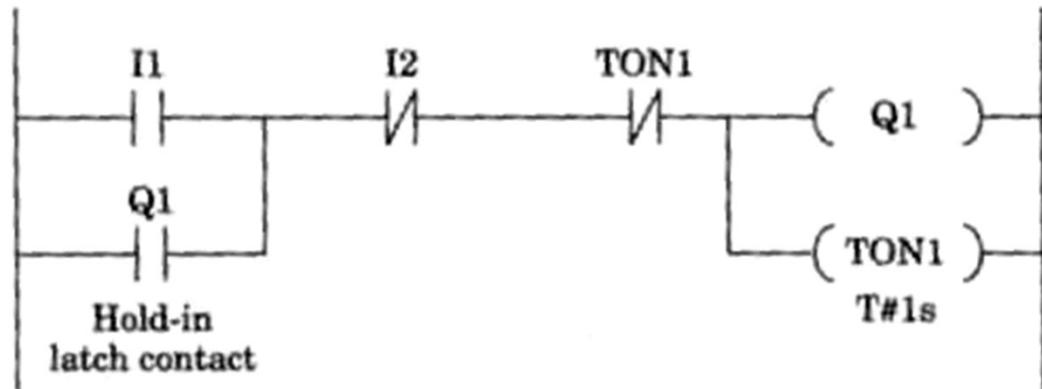
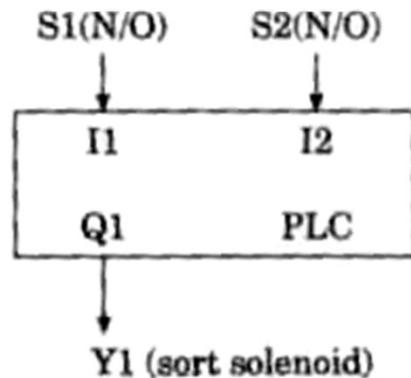
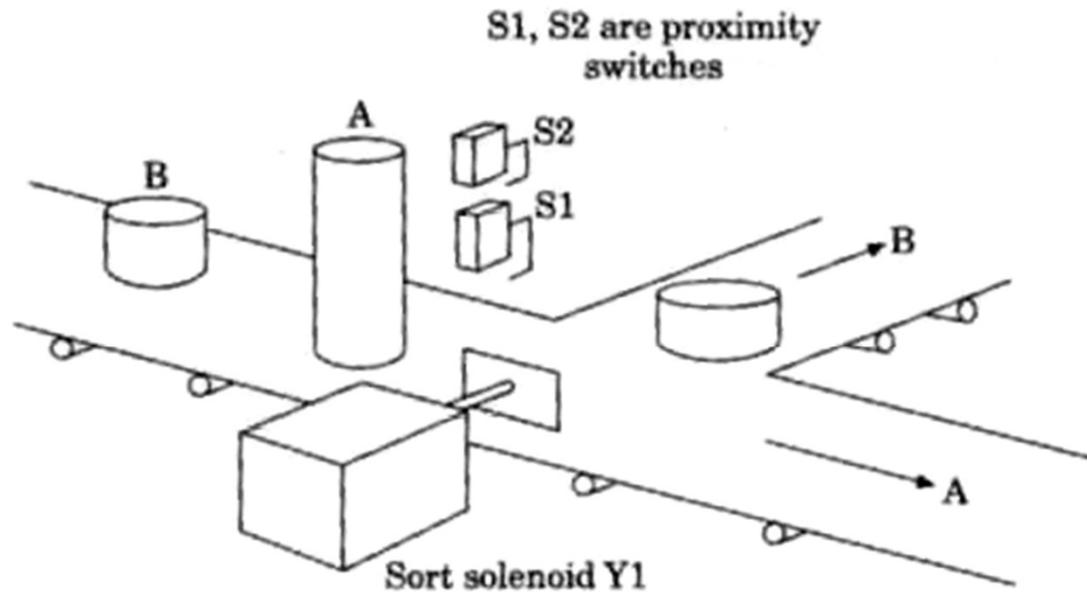
# Esempio applicativo: imballaggio

- obiettivo: 10 pezzi devono essere inviati allo scivolo A e 20 pezzi allo scivolo B; quindi il ciclo deve ricominciare
- un deviatore azionato da solenoide smista i pezzi allo scivolo A o allo scivolo B
- un solenoide è un elettromagnete che sposta un cursore se attivato
- se il solenoide è attivato, i pezzi vengono deviati allo scivolo B
- se il solenoide è disattivato, per l'azione di una molla di ritorno il deviatore si sposta e i pezzi vengono deviati allo scivolo A

# Esempio applicativo: imballaggio

- l'azionamento del pulsante di reset (ingresso I1 del PLC) azzerava i contatori e disattiva Q1, e di conseguenza l'uscita Q3 del PLC, collegata al solenoide
- pertanto, i pezzi vanno allo scivolo A, e vengono contati da I2 sul contatore CTD1
- quando il primo contatore ha contato 10 pezzi, Q1 viene attivato, e di conseguenza i pezzi iniziano ad andare allo scivolo B.
- da questo momento il contatore CTD2 inizia a contare (prima il suo ingresso CD riceveva un segnale nullo, perché aveva in ingresso I2 in serie a Q1)
- quando CTD2 ha contato 20 pezzi, il suo output Q2 azzerava entrambi i contatori, Q1 viene disattivato, anche Q2, e di conseguenza i pezzi ricominciano ad andare allo scivolo A
- il ciclo ricomincia e continua fino allo spegnimento della macchina o fino ad un reset

# Esempio applicativo: selezione e separazione di componenti



# Esempio applicativo: selezione e separazione di componenti

- obiettivo: smistare su un nastro trasportatore due categorie di oggetti in base all'altezza
- il componente di tipo B deve essere deviato verso un secondo nastro da uno spintore azionato da un solenoide, con molla di ritorno
- il componente di tipo A deve proseguire sullo stesso nastro
- due sensori di prossimità S1 e S2 sono posizionati subito prima dello spintore
- al passaggio di A davanti allo spintore i due sensori S1 e S2 si attivano; lo spintore non viene attivato

# Esempio applicativo: selezione e separazione di componenti

- al passaggio di B davanti allo spintore si attiva solo S1; lo spintore viene attivato e B viene deviato; il timer inizia a funzionare
- si noti che nel breve intervallo di tempo in cui il pezzo passa da davanti ai sensori e arriva davanti allo spintore, lo spintore stesso sta accelerando; il sistema deve essere studiato adeguatamente per coordinare la spinta laterale con l'arrivo del pezzo, eventualmente introducendo un ritardo nell'azionamento dello spintore (che, per semplicità, non è presente nel programma ladder di figura)
- dopo 1 secondo, TON1 è attivo, e ciò disattiva Q1 (lo spintore ritorna indietro) e lo stesso TON1
- se non ci fosse il circuito "latch" (la connessione in parallelo di I1 e Q1 a sinistra) lo spintore si sarebbe fermato appena il pezzo B si fosse allontanato dal sensore S1