Cap. 13

13. CONTROLLORI LOGICI PROGRAMMABILI - PLC

13.1 CAMPO APPLICATIVO E STORIA DEI PLC

Il contesto applicativo che ha portato alla comparsa sul mercato dei Controllori Logici Programmabili è quello del controllo di macchine e impianti caratterizzati da una decisa prevalenza di segnali, sia di stato che di controllo, di tipo binario o, come si dice spesso in gergo, "ON/OFF", tipica dei processi discreti.

Le funzionalità richieste per il controllo sono soprattutto abilitazioni, interblocchi, sequenze, conteggi, fasi, ecc., l'ambiente è tipicamente quello industriale "rustico" con disturbi di varia natura (elettrici, meccanici, ambientali) e i requisiti temporali sui tempi di risposta vanno da pochi ms a centinaia di ms.

Le prime applicazioni di automazione di sistemi di questo tipo sono state realizzate mediante l'uso della "logica cablata", cioè di controllori la cui funzionalità è data dalla particolare interconnessione di componenti elementari.
Le tecnologie utilizzate, in dipendenza dai requisiti di costi, sicurezza e velocità sono di tipo pneumatico, elettromeccanico (relè e simili) ed elettronico (porte logiche a vari livelli di integrazione).

Una caratteristica saliente dei sistemi a logica cablata è data dal parallelismo fisico con cui operano tutti i componenti che realizzano quella che possiamo chiamare "funzionalità intrinseca", cioè operatività basata sulla struttura e sulle proprietà fisiche dei dispositivi.

Tra gli aspetti positivi di queste tecniche, che ne consentono la sopravvivenza anche con l'avvento rivoluzionario degli strumenti informatici, va annoverata la capacità di interazione diretta, con adeguati livelli energetici, con fenomeni elettrici e maccanici.

In particolare la logica a relè merita una, sia pur sintetica, rassegna degli aspetti caratteristici.

13.2 LOGICA A RELE'

I relè sono dispositivi elettromeccanici dotati di uno o più contatti elettrici che da una configurazione a riposo passano ad una diversa configurazione quando il relè viene "eccitato", cioè alimentato con una opportuna potenza elettrica applicata alla sua "bobina" (*coil*).

Tra il circuito di pilotaggio ed il circuito attuato esiste una totale separazione galvanica (resistenza pressochè infinita) e un'ottima capacità di sostenere differenze di potenziale (anche molte migliaia di Volt).

Il rapporto tra i livelli energetici dei segnali di pilotaggio (dell'ordine delle centinaia di mW) e quelli in uscita è di 2..4 ordini di grandezza.

I contatti approssimano molto bene il comportamento ideale di resistenza infinita (aperti) e nulla (chiusi). I contatti sono detti NA (Normalmente Aperti) se a riposo (relè diseccitato) sono aperti, altrimenti sono detti NC (Normalmente Chiusi). La maggior parte dei relè prevede contatti "a scambio" (in alcuni casi anche più di uno), che consentono di scegliere tra un contatto NC e uno NA.

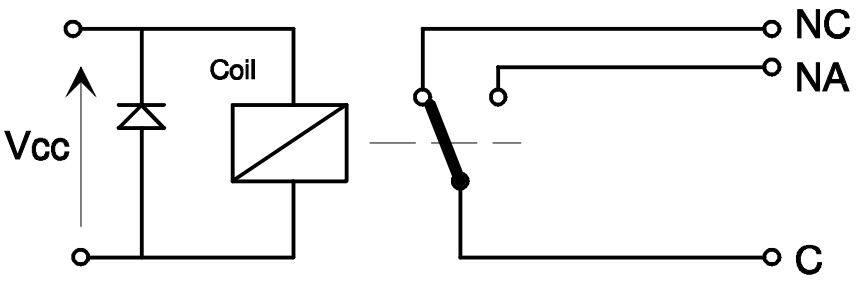

Fig. 13.1 - Schema di relè con singolo contatto di scambio tra morsetto Comune (C) e morsetti NC e NA. Il diodo protegge da sovratensioni dovute al comportamento induttivo della bobina..

Le funzioni logiche elementari sono realizzabili con opportuni collegamenti tra i contatti di tanti relè quanti sono i segnali di ingresso, applicati alle rispettive bobine, come segue.

NOT

L'uscita è ottenuta tramite il contatto NC del relè.

AND

L'uscita è ottenuta collegando in serie i contatti NA delle variabili di ingresso dritte e NC di quelle negate.

OR

L'uscita è ottenuta collegando in parallelo i contatti NA delle variabili di ingresso dritte e NC di quelle negate.

Con le note tecniche per la realizzazione di reti logiche combinatorie e, mediante retroazioni, di quelle sequenziali, è possibile realizzare mediante relè una qualsiasi funzione logica.

Si ricorda che tipici problemi dell'impiego di relè (soprattutto connessi con circuiti elettronici) sono i seguenti.

- La soppressione delle extratensioni di apertura dovute al comportamento fortemente induttivo del circuito di pilotaggio, ottenuta con diodi di spegnimento e gruppi RC.
- I rimbalzi, soprattutto in chiusura, dei contatti di attuazione dovuti alle loro caratteristiche meccaniche elastiche ed inerziali, il cui effetto viene eliminato con opportuni filtri circuitali (RC + ingresso Schmitt a isteresi) o con tecniche software.
- La lentezza, con tempi di attivazione e di rilascio generalmente compresi tra 1 e 20 ms.
- La "vita" relativamente limitata, espressa in numero di cicli.

I tipici modelli di guasto dei relè sono:
- interruzione della bobina.
- contatti "incollati" chiusi
- contatti bloccati aperti.

Il principale problema dei sistemi a logica cablata è costituito dalla loro rigidità, che comporta difficoltà di modifica e manutenzione, ma anche dai loro costi ed ingombri che, soprattutto nei sistemi appena un pò complessi, li vedono molto spesso perdenti nei confronti di sistemi a "logica programmata".

I sistemi a logica programmata, in sostanza i calcolatori, realizzano le loro funzionalità mediante l'esecuzione sequenziale di istruzioni che compongono un programma opportunamente caricato nella memoria operativa.
I processori sono in larghissima misura indipendenti dalla particolare applicazione ed usufruiscono così di economie di scala, realizzazione con tecnologie ad elevati livelli di integrazione, ecc.

I programmi dipendono invece dalle particolari applicazioni, ma risultano di progettazione, realizzazione, messa a punto e modifica estremamente facilitate (rispetto al cablaggio) grazie alla disponibilità di sempre più sofisticati strumenti di sviluppo di tipo informatico.

Si è così assistito ad un affiancamento ai sistemi a logica cablata di sistemi programmabili che progressivamente hanno finito per soppiantarli quasi completamente, soprattutto dove maggiormente è cresciuta l'esigenza di elaborazioni complesse rispetto all'interfacciamento di potenza.

Con l'avvento dei microprocessori i PLC hanno avuto un notevole impulso ma, più che fare sfoggio di avanzate caratteristiche informatiche, hanno dovuto guadagnarsi la fiducia dei tecnici impiantisti (quadristi, elettricisti, ecc.) con doti di robustezza e di facilità di impiego.

La robustezza è ottenuta con costruzioni a livelli qualitativi industriali e con particolari accorgimenti nei circuiti di interfaccia che, esposti alle ingiurie del mondo esterno, sono l'elemento più critico.

La facilità di applicazione è stata perseguita in un modo efficace, anche se agli informatici può sembrare riduttivo e quasi ridicolo, scimmiottando "reti virtuali di relè". In tal modo il personale con esperienza elettrica si è trovato più rapidamente a proprio agio nei primi semplici approcci alla nuova tecnica dei PLC.

Negli anni '80 i PLC hanno conquistato una notevole importanza nelle applicazioni di automazione e, pur manifestando tendenze a convergere verso impostazioni più moderne ed informatiche, mantengono ancora una caratterizzazione che merita un approfondimento.

Dopo questa premessa passiamo ad una sintetica rassegna di caratteristiche e tecniche di impiego di questi particolari calcolatori specializzati.

13.3 CARATTERISTICHE TIPICHE DEI PLC

13.3.1 CPU

Alcuni PLC, soprattutto di fascia media o alta, adottano come processore uno dei microprocessori "general purpose" diffusi sul mercato (ad es. 8085, 8051, 68020, ecc.), mentre altri utilizzano processori appositamente sviluppati. Questi ultimi sono dotati di un set di istruzioni macchina orientato ad eseguire con la massima efficienza operazioni logiche su singoli bit, operazioni aritmetiche in codifica BCD, ed in genere operazioni semplici su dati dei tipi BIT, NIBBLE (4 bit) e STRINGA.
Pochi registri interni e ridottissime modalità di indirizzamento su aree di memoria ridotte (1..8 K) consentono di avere istruzioni tutte della stessa lunghezza e con tempi di esecuzione brevi ed eguali.
La potenza di calcolo delle CPU utilizzate nei PLC è generalmente espressa indicando in ms il tempo necessario per eseguire 1024 (1K) istruzioni, che va dai 25 ms/K per i PLC più lenti fino ai 0.4 ms/K per i più potenti.

13.3.2 MEMORIE

La memoria operativa dei PLC è generalmente preallocata in modo rigido alle diverse funzioni previste, con caratteristiche ed estensioni anche molto variabili da modello a modello, ma generalmente secondo la struttura esemplificata nel seguito.

- ROM per il SW di sistema, invisibile all'utente ed in genere non dichiarata nella documentazione tecnica.
- EPROM o EEPROM di 1..16 Kparole per il programma applicativo.
- RAM preallocata per le variabili di sistema, per indicatori di stato (overflow del periodo di ciclo, scarsa tensione di alimentazione, ecc.), per temporizzatori SW (1, 10, 100 ms, 1, 60 s) e per contatori (0..9999).
- RAM preallocata per le immagini degli IN e degli OUT.
- RAM "scratchpad" per le variabili interne dichiarate dall'applicazione (256..8K).
- MEMORIE DI MASSA sono usate solo raramente,per motivi di ostilità ambientale.

13.3.3 SEGNALI I/O DA PROCESSO

on/off robusti e condizionati
analogici (con livelli standard)
isolamento galvanico (opto)

I PLC sono dotati di morsetti previsti per i tipici segnali di ingresso e uscita da processo.

13.3.3.1 SEGNALI DIGITALI (ON/OFF)

13.3.3.1.1 INGRESSI DIGITALI

Gli ingressi sono di vari tipi, adatti per tensioni continue o alternate nominali di 24, 48, 110 e 220 V, con optoisolamento e con filtraggio contro spurii (passa-basso fino a qualche decina di Hz) e protezioni da sovratensioni.
Alcuni ingressi sono detti "veloci" perchè dotati di filtro di ingresso a banda più larga (kHz) che lascia passare anche sequenze di commutazioni più rapide per conteggi ecc.
Gli ingressi presentano elevati margini di rumore: con 24 V nominali si rileva LOW per tensioni di 0 . . 7 V e si rileva HIGH per tensioni di 17 . . 30 V. Anche l'assorbimento in corrente è sostanzioso: da 5 a 25 mA.
In genere ad ogni bit di ingresso è associato un LED indicatore che consente una immediata visione del loro stato.

13.3.3.1.2 USCITE DIGITALI

Le uscite digitali sono varie, in dipendenza dei diversi tipi e livelli energetici dei carichi, sono anch'esse isolate galvanicamente e generalmente protette contro sovraccarichi.
Si trovano:

- uscite a relè, adatte sia per CC che per CA;
- uscite a transistor open-collector per CC;
- uscite a triac per CA.

Si hanno uscite adatte per tensioni di 24..220 V e correnti da 100 mA fino a 8 A.
In alcuni modelli di PLC è possibile "sentire" in ingresso lo stato di eventuale sovraccarico delle uscite.
Anche le uscite sono generalmente dotate di LED indicatori dello stato dei singoli bit.

13.3.3.2 SEGNALI ANALOGICI

Poichè i PLC sono prevalentemente orientati alla gestione logica di segnali digitali, le interfacce per segnali analogici in genere costituiscono moduli opzionali aggiuntivi per i PLC di media e alta fascia.

13.3.3.2.1 INGRESSI ANALOGICI

Sono previsti ingressi in tensione (5 V, 10 V) o in corrente (0..20 mA) con filtraggio, multiplexer, Sample-Hold e convertitore A/D con risoluzioni di 10..16 bit.

Spesso, per ridurre il carico di lavoro del processore principale del PLC, le interfacce di ingressi analogici sono dotate di una propria gestione autonoma, con processore dedicato, delle operazioni sequenziali di selezione canale, campionamento ed avvio della conversione A/D, che sono ripetute ciclicamente.
L'isolamento galvanico, spesso previsto, può riguardare in blocco tutti gli ingressi o anche una separazione tra un canale e l'altro.

13.3.3.2.2 USCITE ANALOGICHE

Possono essere in tensione (0 - 5V, 0 - 10V) o in corrente (0 - 20 mA) e con risoluzione di 10 . . 16 bit.

13.3.4 ROBUSTEZZA

I PLC devono poter essere installati e funzionare regolarmente anche in ambienti ostili per le sollecitazioni meccaniche (vibrazioni), termiche, e per presenza di polveri, umidità, ecc.
Sono perciò costruiti con componenti selezionati e tecniche costruttive adatte.
Sono generalmente assenti unità a nastri o dischi magnetici, troppo delicate.

13.3.5 TOLLERANZA AI GUASTI (FAULT TOLERANCE)

In genere nei PLC vengono adottati accorgimenti che riducono gli effetti di eventuali guasti. Soprattutto si adottano tecniche di recupero del comportamento corretto in seguito ad errori transitori.

Un tipico esempio è costituito dal meccanismo detto "*watch- dog*" (letteralmente "cane da guardia"), che consiste di un circuito monostabile che durante il corretto funzionamento viene "riarmato" ("retriggerato") prima che scada la durata dello stato instabile. In caso di mancato riarmo il circuito scatta e riporta, generalmente tramite un segnale RESET, il PLC in uno stato corretto da cui esso riprende l'attività. Il mancato riarmo in genere è sintomo di mancato completamento corretto di un ciclo di programma entro un *time-out* prestabilito.

13.3.6 SICUREZZA (FAIL-SAFE)

La sicurezza consiste nella garanzia (o almeno elevata probabilità) che i guasti e gli errori portino il PLC, ed in particolare le sue uscite, in uno stato di minimo rischio per persone e cose. Si tratta di un problema tanto importante quanto complicato, soprattutto perchè dipende molto dalla particolare applicazione quali stati, o anche quali transitori, sono più sicuri e quali meno.
Alcuni PLC sono progettati e realizzati con particolare attenzione a questo problema.

Le tecniche adottate per perseguire gli obbiettivi di tolleranza ai guasti e sicurezza sono generalmente le seguenti.

- Ridondanza doppia con *back-up* caldo: gli elementi critici sono duplicati e uno dei due esemplari opera effettivamente, mentre l'altro rimane di scorta (*back-up*), ma attivo (caldo), e pronto a subentrare in caso di guasto del primo.
- Ridondanza tripla con voto: gli elementi critici sono triplicati e contemporaneamente attivi. In caso di discordanza dei risultati, rilevata mediante circuiti votatori, viene assunto valido il comportamento assunto da almeno due dispositivi concordanti.

13.3.7 FACILITA' DI PROGRAMMAZIONE

Come si è accennato nell'introduzione del capitolo, i PLC hanno fatto della facilità di programmazione una loro arma vincente. Tale facilità si evidenzia però soprattutto per algoritmi piuttosto semplici (semplicità "*in the small*").

In particolare sono state adottate tecniche di programmazione orientate a semplici problemi "booleani" e con formalismi noti ai tecnici del settore, cercando di "aggirare" la loro iniziale e solo recentemente parzialmente attenuata ostilità verso l'uso dei calcolatori in queste applicazioni.

Nel seguito del capitolo presenteremo alcuni dei più diffusi linguaggi di programmazione e gli strumenti, i sistemi di sviluppo, utilizzati per la realizzazione di programmi per questo particolare ambiente.

13.3.8 VERIFICABILITA' DEL REAL-TIME

- In sede di progetto si valuta che sia $Tr < 2*Tc$
- In esecuzione viene generato un interrupt di supero Tc

Il rispetto rigoroso di requisiti *real-time* è una caratteristica necessaria nella maggior parte dei sottosistemi di automazione, e costituisce quindi uno degli obbiettivi di progetto dei PLC.

Si noti che è stata inizialmente privilegiata la garanzia del rispetto dei tempi di risposta, rispetto alla velocità: ricordiamo che lo scopo era di sostituire armadi di relè i cui tempi di commutazione sono dell'ordine delle decine di ms. Naturalmente il campo di applicabilità si estende con i moderni e veloci PLC che presentano tempi di risposta dell'ordine del millisecondo.

Per una buona prevedibilità dei tempi di risposta si è adottata la tecnica di esecuzione ciclica "*time-driven*" di una stessa sequenza di istruzioni (il programma) indipendentemente dallo stato e dagli eventi del sistema da controllare (v. cap.10).
Con questo approccio, detto Tc il periodo di ciclo e Tr il ritardo tra una variazione in ingresso al PLC e la conseguente variazione delle uscite, si ha:

$$\mathbf{Tr < 2*Tc}$$

Sono poi previsti meccanismi di riconoscimento del caso in cui il tempo di esecuzione dell'intero programma superi il tempo Tc di ciclo previsto.

In sede di messa a punto dell'applicazione il progettista potrà decidere, in caso di supero, se accettare un aumento del periodo Tc o ridurre la lunghezza del programma o, se necessario, passare ad un PLC più veloce.

In esercizio la segnalazione di supero del periodo Tc potrà portare a "saltare" la parte finale del programma, che si suppone meno critica, o al blocco in sicurezza del sistema.

13.4 STRUTTURA HW - INTERFACCE

Rispetto alle caratteristiche di estendibilità del numero degli I/O, della quantità di memoria e dalla potenza di calcolo, si classificano generalmente i modelli di PLC in tre fasce.

13.4.1 PLC DI FASCIA BASSA

Hanno una configurazione fissa e con pochi I/O (ad es. 16 IN + 16 OUT) semplici e limitati ai segnali digitali.

Il processore è costituito da un microcontrollore "single chip" corredato, oltre che di porte di I/O, di temporizzatori ed eventualmente di UART per la gestione di collegamenti seriali.

La memoria operativa è fissa e costituita da un'area RAM (1 K) per i dati e un'area su EPROM o EEPROM (2 K) per i programmi applicativi, mentre il SW di sistema, piuttosto ridotto, è stabilmente memorizzato su ROM.

13.4.2 PLC DI FASCIA MEDIA

Presentano la tipica architettura a BUS dei microcalcolatori, che consente una buona modularità.

Sono quindi previsti diversi tipi di moduli che consentono di espandere le interfacce con le funzionalità richieste dalle applicazioni.
Si hanno moduli digitali a relè, a stato solido e per conteggi veloci, moduli analogici, moduli per controllo di assi, moduli per collegamenti seriali punto a punto o in rete.

I moduli che realizzano le interfacce con funzionalità più complessa sono spesso dotati di "intelligenza" locale, cioè di piccoli processori specializzati per le funzioni richieste e che operando in parallelo col processore principale consentono l'ottenimento di tempi di risposta rapidi e prevedibili, oltre a semplificare la programmazione che può usufruire di primitive di livello più elevato direttamente implementate in "firmware" dal costruttore.

La memoria operativa è più estesa (16 K) e flessibile e comprende generalmente la possibilità di avere aree di RAM non volatile (in tampone).

13.4.3 PLC DI FASCIA ALTA

Hanno un'architettura modulare simile alla precedente, ma con un'espandibilità ancora maggiore e soprattutto con una notevole potenza di calcolo, così da costituire dei veri e propri calcolatori industriali.
Corrispondentemente è più ricco e complesso l'insieme dei linguaggi di programmazione e dei relativi strumenti.

In ogni caso i programmi dei PLC risiedono in memorie non volatili, così da essere immediatamente eseguibili all'accensione del sistema portando rapidamente i segnali di comando a valori consistenti e di sicurezza.
Le memorie dei programmi potranno essere delle normali EPROM che vengono programmate su apposite apparecchiature, parte integrante del sistema di sviluppo, e poi inserite negli appositi zoccoli (ZIF = *Zero Insertion Force*) all'interno del PLC.

E' più pratica e sempre più diffusa la soluzione basata su memorie EEPROM che possono essere cancellate e riprogrammate "in loco" dallo stesso PLC, che però in questo caso deve essere dotato di:
- circuiteria per la riprogrammazione delle EEPROM
- interfaccia seriale per ricevere i programmi (download) dal sistema di sviluppo;
- SW di sistema che gestisce il trasferimento di tali programmi.

13.5 MODELLO DI ESECUZIONE

Viene qui presentato sinteticamente il modello di esecuzione tipico dei PLC più semplici, notando che l'evoluzione verso funzionalità più ricche e complesse sta portando, soprattutto i PLC più potenti, verso un approccio più "informatico" di tipo *multitasking*.

13.5.1 DATI

- Memoria globale condivisa
- INPUT / OUTPUT - immagini in memoria

La memoria è preallocata ai diversi tipi di variabili quasi tutte "globali", che sono tipicamente le seguenti.
- Dati di sistema
- Dati immagini degli ingressi
- Variabili interne per risultati intermedi, contatori, ecc.
- Dati immagini delle uscite

I tipi di dati caratteristici dei PLC sono:
- BIT che rappresentano variabili booleane
- DIGIT che utilizzano un "nibble" di 4 bit per rappresentare cifre decimali in codifica BCD (*Binary Coded Decimal*).

I PLC di fascia superiore utilizzano anche i "normali" tipi BYTE, integer, real.

In generale i nomi delle variabili dei vari tipi non possono essere scelti in modo mnemonico dal programmatore, ma sono costituiti da una lettera che designa il tipo e da un numero che rappresenta l'indirizzo fisico.
Ad esempio delle variabili di **ingresso** potranno essere

I02 E04.6

delle variabili di **uscita**

O12 A07

e delle variabili **interne**

M02 M05.3

Nota - In inglese ingresso è **I**nput, in tedesco **E**ingang. Uscita è rispettivamente **O**utput e **A**usgang, mentre le variabili interne sono dette **M**emory o **M**erker.
Un indirizzo 04.6 sta ad indicare il bit in posizione 6 del byte di posizione 04 nell'area di indirizzamento.

13.5.2 PROGRAMMI

I programmi sono organizzati nella forma di sequenze lineari di istruzioni, generalmente senza cicli interni ma con la possibilità di esecuzione condizionata di alcune sequenze, analoga al noto costrutto "*if - then* - ". L'intero programma viene ripetuto ad ogni ciclo.

Le funzioni "di sistema" sono eseguite all'inizio di ogni ciclo, su attivazione da parte del timer di base che scandisce il periodo Tc, mentre le operazioni di ingresso e uscita possono essere eseguite con diverse modalità dipendenti dal tipo di PLC:

13.5.2.1 MODALITA' SINCRONA

All'inizio di ogni ciclo vengono acquisiti tutti gli ingressi "catturando" il loro valore ed aggiornando di conseguenza le corrispondenti immagini in memoria.
Questo approccio, simile a quello delle macchine sequenziali sincrone, garantisce stabilità e consistenza di tutte le variabili di ingresso e semplifica la scelta della corretta successione delle operazioni logiche parziali (evitando corse critiche).

Le uscite sono aggiornate a fine ciclo, riportando ai morsetti i valori assunti in quel momento dalle corrispondenti variabili di immagine in memoria.
In questo modo si evitano i "*glitch*" (alee statiche).

13.5.2.2 MODALITA' ASINCRONA

Con questa modalità ogni volta che si esegue un'operazione su valori in ingresso, questi vengono acquisiti dall'esterno, mentre i risultati destinati alle uscite vengono immediatamente emessi appena calcolati.

Questo approccio consente di ridurre i tempi di risposta, ma richiede una molto maggior attenzione alla successione delle istruzioni del programma e una conoscenza più precisa dei tempi di reazione dei fenomeni del sistema controllato. Infatti la stessa variabile di ingresso utilizzata in due diversi punti del programma può assumere diversi valori, se nel frattempo il segnale in ingresso è cambiato.

Sono previsti anche i casi "misti", tra cui è probabilmente più significativo il caso IN-Sincrono / OUT-Asincrono.

Il temporizzatore che scandisce il periodo di esecuzione in genere provoca un interrupt. In tal modo la routine di sistema che viene attivata in seguito a tale interruzione è in grado di accorgersi se non era stata ancora completata l'esecuzione del ciclo precedente. Questa evenienza, a seconda dei casi, puòsemplicemente impostare a TRUE un indicatore (*flag*) predefinito di supero del periodo oppure può provocare di forza il lancio di un nuovo ciclo di esecuzione.

Come si è accennato presentando le varie tecniche di parallelismo virtuale (v. cap. 10), l'approccio ciclico semplice, accanto ai vantaggi di semplicità e predicibilità dei tempi di risposta, presenta il difetto di uno scarso sfruttamento della potenza di calcolo del processore che è egualmente distribuita su funzioni che possono invece avere urgenza ed importanza molto diverse tra loro.

Una soluzione adottata su alcuni modelli di PLC consiste nella possibilità di accettare interruzioni da eventi esterni la cui gestione è quindi rapida ed affidata ad opportune routine di servizio.

Un'altra interessante soluzione è costituita da un'organizzazione esecutiva in cicli multipli. Si prevede cioè una separazione delle varie funzioni ed un loro raggruppamento in "classi di urgenza", ognuna delle quali costituisce un "sottoprogramma" eseguito ciclicamente.
Naturalmente i periodi di esecuzione dei vari sottoprogrammi sono diversi: ad es. 1, 20, 500 ms.
Con questo approccio si virtualizzano tanti processori, con memoria condivisa, quanti sono i cicli previsti.

13.6 LINGUAGGI DI PROGRAMMAZIONE

I linguaggi di programmazione dei PLC sono nati per consentire ai tecnici una semplice simulazione di logiche cablate, a relè o elettroniche. Si tratta di linguaggi "proprietari", cioè proposti dai singoli produttori di PLC per le proprie famiglie di prodotti, ma che hanno seguito alcuni filoni principali all'interno dei quali si presentano con caratteristiche molto simili. Ha senso quindi presentare le principali classi di linguaggi, notando che per ognuna di esse si è avuta una progressiva evoluzione con aggiunte di

nuove funzioni, introduzione di varie tecniche per la modularità, la leggibilità e la documentazione.
Recentemente (1993) è stata emanato lo standard internazionale IEC 1131-3 sui linguaggi di programmazione per i Controllori Programmabili.

E' interessante notare che, almeno nella loro impostazione di base, la maggior parte dei linguaggi di programmazione tipici dei PLC presentano un'impostazione "dichiarativa" (o descrittiva) analoga agli schemi circuitali, in cui la collocazione "topologica" delle varie funzionalità è considerata irrilevante, mentre è significativo solo il legame funzionale tra ogni risultato e gli operandi che concorrono alla sua valutazione.
Ovviamente il mappaggio di tali funzioni su un'esecuzione fisicamente sequenziale limita la libertà di collocazione topologica dei vari blocchi funzionali se si vuole mantenere l'equivalenza (e quindi la correttezza) tra le diverse "collocazioni", ed in particolare minimizzare i tempi di risposta (v. cap.10).

Per i vari linguaggi si riporta il **termine inglese** e accanto anche la **sigla tedesca**.

13.6.1 SCHEMI A CONTATTI - LADDER - KOP

Si tratta di un linguaggio **grafico**, tipico della scuola americana ma supportato praticamente da tutti i PLC, così da poter essere considerato il loro linguaggio di programmazione tipico.

I programmi scritti in questo linguaggio prendono il nome di "*ladder diagram*" per la loro tipica struttura a "scala" in cui ogni "piolo" rappresenta la funzionalità necessaria per il calcolo di un risultato.

Gli operandi sono variabili booleane rappresentate con "contatti" NA (normalmente aperti) se diritte o NC (normalmente chiusi) se negate.
Queste variabili possono essere sia valori in ingresso (I), sia valori interni (Memory) che valori di uscita (O). I risultati possono essere variabili interne (M) o uscite (O).

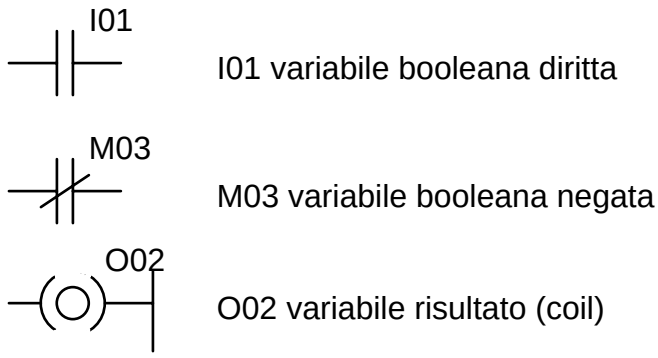

Fig. 13.2 - Tipici simboli del linguaggio LADDER.

Le operazioni logiche sono descritte dal particolare collegamento dei contatti, in serie per l'operatore AND ed in parallelo per l'operatore OR.

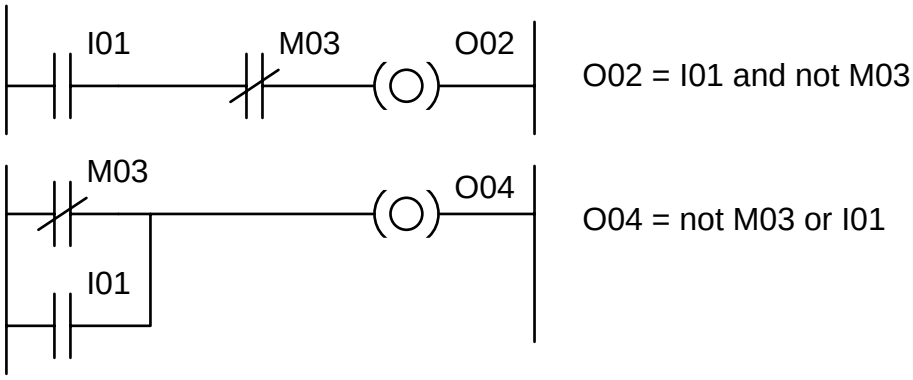

Fig. 13.3 - Esempi di semplici funzioni logiche.

Possono essere descritte varie configurazioni con il limite di un massimo numero di variabili (contatti) in senso orizzontale. In questi "circuiti" il flusso ideale di corrente è da intendersi sempre da sinistra verso destra.

L'uscita di ugni funzione **combinatoria** è attribuita ad una variabile interna o di uscita e l'assegnamento è rappresentato dal simbolo della "bobina" di un relè.
Eventuali reti **sequenziali** sono ottenibili utilizzando contatti associati alla variabile risultato (retroazioni).

ESEMPIO DI RETE SEQUENZIALE

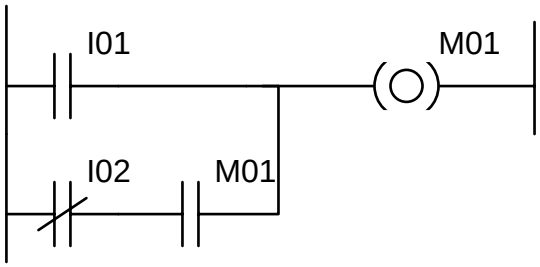

Fig. 13.4 - Rete sequenziale (flip-flop). M01 è l'uscita di un flip-flop con SET = I01 e RESET = I02.
I01 può essere interpretato come pulsante di start di un motore, attivato dall'uscita M01, mentre I02 può essere considerato il pulsante di arresto. Si noti il contatto M01 di "autoritenuta".

La leggibilità "locale" di questo linguaggio è abbastanza immediata, soprattutto per chi sia abituato a ragionare in termini di contatti, e limitata solo dai nomi non mnemonici delle variabili.

La funzionalità globale di un programma è invece di lettura a difficoltà crescente in modo esponenziale con la lunghezza del programma, soprattutto se non si abbia cura di collocare in posizioni adiacenti, quando possibile, le funzioni tra loro dipendenti o correlate.

Le estensioni a questi linguaggi proposte dai vari costruttori di PLC, sono generalmente basate sull'aggiunta di altri dispositivi, oltre alla bobina di relè, che svolgono funzioni più complesse ma considerate primitive dal programmatore applicativo. In qualche modo le estensioni portano questo linguaggio a sconfinare verso le caratteristiche dei blocchi funzionali presentati nel seguito.

13.6.2 LISTA ISTRUZIONI - BOOLEAN - AWL

Il linguaggio a lista di istruzioni è di tipo **testuale**, più vicino alle consuete notazioni dell'algebra booleana e al paradigma imperativo, ed è originario della scuola tedesca. Anche questo tipo di linguaggio è disponibile per tutti i modelli di PLC, soprattutto nella variante “verticale”.

-- VERTICALE
La variante più diffusa è molto simile ad un semplice linguaggio Assembler con istruzioni ad un operando esplicito ed uno implicito (a pila LIFO). Ogni funzione combinatoria è realizzata da una successione (verticale) di istruzioni elementari, l'ultima delle quali provoca il trasferimento del risultato nella variabile di destinazione.

I nomi delle variabili sono tipicamente costituiti da una lettera che ne indica il tipo (Input = Eingang, Output = Ausgang, Memory) seguito da un numero di posizione.

ESEMPIO O2 = not I3 and I5

```
LOAD_NOT  I3
AND           I5
STORE     O2
```

La leggibilità di questo linguaggio è molto scarsa e neppure la modularità è il suo forte: il pregio principale sta nell'estrema semplicità dell'editor e del traduttore.

Le estensioni a questo tipo di linguaggio disponibili sui vari PLC, sono tipicamente costituite da altre istruzioni, tipicamente nel linguaggio Assembler del processore utilizzato per il PLC, per operazioni di shift, aritmetiche, di salto, ecc.

-- ORIZZONTALE
Si tratta di una variante molto simile alle istruzioni di assegnamento dei linguaggi ad alto livello, in cui le operazioni sono descritte sotto forma di espressioni (booleane) in una stessa riga (orizzontale).

ESEMPIO

```
O2 := NOT I3 AND I5
```

Le estensioni a questo tipo di linguaggio portano a forme simili ai linguaggi di programmazione "classici".

13.6.3 BLOCCHI FUNZIONALI - FUNCTIONAL BLOCKS - FUP

Il linguaggio a blocchi funzionali è di tipo **grafico**. Ogni operatore, o funzione, è rappresentato da un rettangolo contrassegnato dall'indicazione del tipo di operazione, mentre le variabili sono rappresentate dai "morsetti" di ingresso e di uscita dei vari blocchi.
Le funzioni complesse sono descritte mediante opportune connessioni tra blocchi elementari.
Si tratta di una rappresentazione che, ancora una volta, richiama gli schemi circuitali, fornendo una descrizione di tipo "data flow" o "funzioni di trasferimento" che ben si prestano ad una facile lettura, alla modularizzazione e a diversi livelli di astrazione.

La disponibilità di librerie di blocchi che realizzano funzioni anche piuttosto complicate (contatori, filtri, regolatori PID, ecc.) consente una facile estensione ad elaborazioni logiche complesse e anche numeriche.

Per alcuni PLC questo linguaggio è disponibile solo come supporto di documentazione, proprio per le sue doti di leggibilità.

13.6.4 GRAFCET (Graphes Commandes Etapes Transitions)

Il GRAFCET è un linguaggio **grafico** di origine francese relativamente recente (standardizzato nel 1982 con norma UTE NFC 03-190 e poi nel 1993 con TEC 1131-3 SFC), che si propone, contrariamente ai linguaggi grafici precedenti, orientati al "flusso dei dati", di evidenziare il "flusso di controllo", cioè l'evoluzione temporale sia **sequenziale** che **parallela**, e si propone come strumento di descrizione sia di specifiche che di progetti.

Per questo linguaggio sono definiti due livelli:
1) Livello funzionale "astratto".
2) Livello tecnologico che riporta i dettagli implementativi con riferimenti precisi a sensori, attuatori, tecnologie, ecc.

Il GRAFCET si ispira alle note reti di Petri, che tanta attenzione hanno suscitato per la loro possibilità di descrivere evoluzioni temporali discrete in parallelo. Analogamente alle reti di Petri si utilizzano grafi con archi orientati che collegano nodi di due tipi diversi: Tappe (Etapes in francese) e transizioni (Transitions).

Il linguaggio GRAFCET non è strettamente legato solo al mondo dei PLC come i linguaggi presentati precedentemente in questo capitolo, ed è stato presentato più diffusamente nel capitolo 3.

13.6.5 BASIC, C, PASCAL (like)

Soprattutto per i PLC di fascia media e alta sono talvolta disponibili linguaggi molto simili ai classici linguaggi di programmazione, eventualmente con varianti o estensioni orientate alla manipolazione di bit. Il primo linguaggio ad avere una certa diffusione è stato il BASIC, per cui sono stati realizzati anche microprocessori con "interprete" BASIC incorporato, ma recentemente sta acquistando interesse anche l'onnipresente linguaggio C.

13.7 SISTEMI DI SVILUPPO

I PLC non sono in genere attrezzati per supportare direttamente lo sviluppo dei loro programmi. Pur essendo infatti spesso dotati di processori di potenza paragonabile a quelli dei PC, sono largamente carenti delle memorie di massa, stampanti e interfacce operatore necessarie per lo sviluppo, documentazione e traduzione dei programmi.

La diffusione dei PC e le loro dotazioni tipiche, ne hanno ben presto fatto il tipico strumento di appoggio per tutte le operazioni di sviluppo di applicazioni, anche se molti costruttori di PLC tendono a fornire specifici sistemi di sviluppo.

I costruttori di PLC forniscono in genere dei pacchetti di tipo CASE (Computer Aided Software Engineering) che tra l'altro contengono:

- Editor guidati da sintassi per i vari linguaggi grafici o testuali supportati.
- Traduttori in codice macchina.
- Simulatori per esecuzioni di prova a bordo del PC.
- Programmi di controllo dei Programmatori di EPROM.
- Programmi di colloquio seriale con i PLC per "*downloading*" dei programmi eseguibili e per operazioni di *debug* "remoto".

13.8 COLLEGAMENTI CON HOST E IN RETE

Generalmente i PLC sono privi di interfaccia operatore, o al più presentano alcuni indicatori di stato a LED e qualche pulsante di comando.
Nelle applicazioni più semplici l'interazione con l'esterno, oltre al processo da controllare, si limita a semplici pannelli operatore costituiti da pulsanti, commutatori e lampade (spie) che possono essere direttamente collegati ai morsetti di ingresso e di uscita.

Nelle applicazioni più complesse che richiedono elaborazioni anche numeriche dei dati acquisiti, memorizzazione storica, stampe e visualizzazioni ed interazioni con l'operatore più articolate, si ricorre generalmente a calcolatori di tipo PC eventualmente in versione "industriale", a cui il PLC viene collegato tramite una porta di comunicazione seriale.
La connessione segue generalmente lo standard EIA-RS232 o RS485, con velocità da 1200 a 9600 Baud e protocollo imposto dal PLC.

Il calcolatore, detto anche "*host*", svolge prevalentemente compiti di supervisione, mentre il controllo in tempo reale è affidato al PLC. Per le funzioni più lente o con vincoli temporali meno stringenti si ha una certa libertà di scelta progettuale nell'attribuzione di tali funzioni al PC o al PLC.

In molti casi il calcolatore *host* viene utilizzato anche come sistema di sviluppo, consentendo quindi modifiche o aggiustaggi del programma del PLC talora anche in esercizio.

13.8.1 RETI DI PLC

Recentemente ha assunto una crescente importanza la possibilità di collegare "in rete" anche numerosi PLC.
Si tratta in genere, purtroppo, di reti proprietarie con cui sono compatibili solo i PLC dello stesso costruttore, che consentono la realizzazione di sistemi distribuiti con una facile distribuzione di risorse e compiti su distanze anche di diverse centinaia di metri.

Il protocollo ciclico adottato da queste reti realizza una memoria virtualmente condivisa che consente ad ogni PLC di utilizzare come ingressi tutte, o parte, le variabili di ingresso e uscita degli altri PLC in rete.

Con una struttura a rete possono anche essere razionalizzate le interazioni dei calcolatori di supervisione con i vari PLC.

13.9 ALTRE LETTURE

David G. Johnson
PROGRAMMABLE CONTROLLERS FOR FACTORY AUTOMATION
Marcel Dekker, Inc. 1987

P. Galeano, V. Alessandroni
I CONTROLLORI PROGRAMMABILI
Ed. Jackson

A. Cavalcoli
PLC - I CONTROLLORI PROGRAMMABILI
Ed. CLUP

M. Blanchard
COMPRENDRE, MAITRISER ET APPLIQUER LE GRAFCET
Ed. Cepadues 1979

Autori vari
I PLC NEI SISTEMI DI AUTOMAZIONE
Giornata di studio - Aprile 1991
ANIPLA - p.le Morandi 2 - Milano