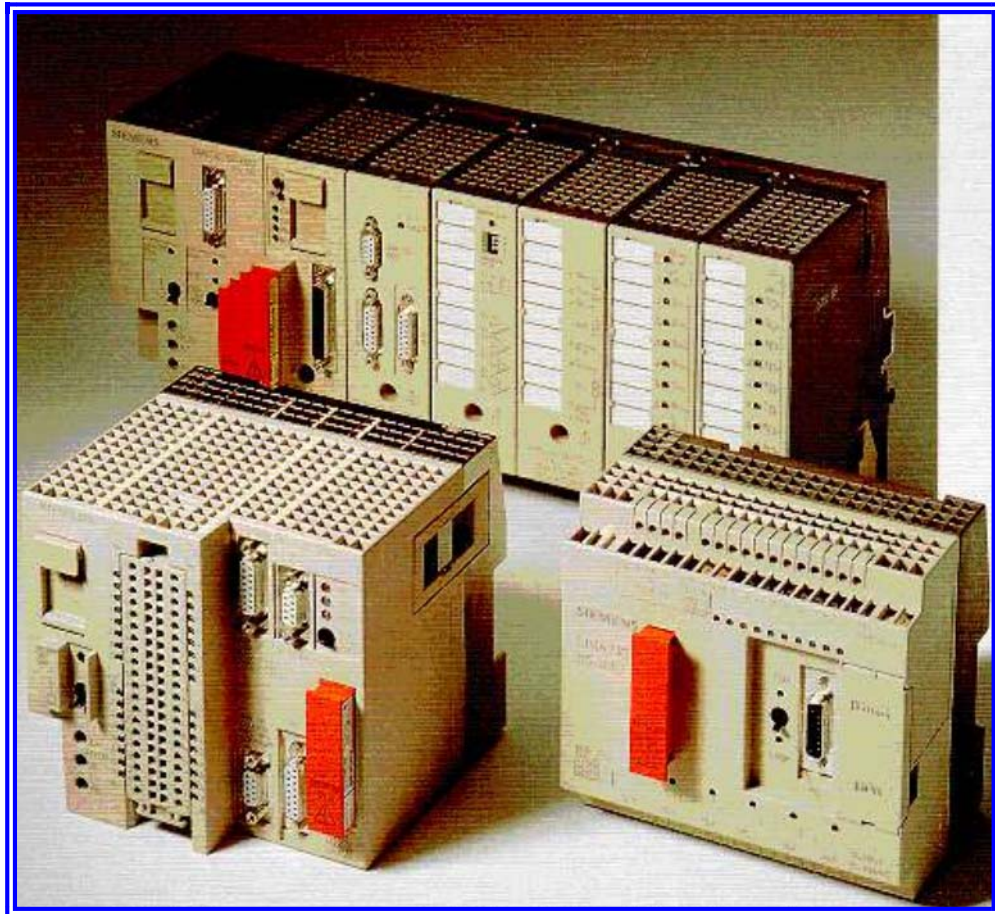


# Programmable Logic Controller



*- Controllore a Logica Programmabile -*

(con particolare riferimento ai PLC Siemens S5-100U, S5-95U e S5-90U)

**Dispensa a cura del Prof. Silvio Bani**

## INTRODUZIONE AI CONTROLLORI LOGICI PROGRAMMABILI (PLC)

### *Introduzione*

Il controllore a logica programmabile (PLC : Programmable Logic Controller) è una apparecchiatura elettronica costituzionalmente molto simile a un calcolatore, ma realizzata con delle particolarità specifiche, soprattutto per quanto riguarda il linguaggio di programmazione e l'interfacciamento con i dispositivi esterni.

Questa nuova categoria di apparecchiature, apparsa per la prima volta sul mercato, attorno agli anni '70, è stata sempre più raffinata ed è oggi in continua e rapida evoluzione. Avvalendosi delle più moderne tecnologie elettroniche, i controllori a logica programmabile hanno raggiunto elevata affidabilità nelle applicazioni industriali, dove esistono condizioni ambientali molto severe causate da disturbi elettrici, alte temperature, shock meccanici e sostanze inquinanti.

Il controllore a logica programmabile sta assumendo il ruolo di componente fondamentale per la realizzazione della fabbrica automatica. Non esiste infatti settore di produzione, trasformazione o controllo nel quale non si trovino esempi di applicazioni del controllore a logica programmabile.

I controllori delle ultime generazioni hanno perso le caratteristiche originali di semplici sequenziatori per assumere sempre più quelle di veri calcolatori di processo, dotati di possibilità di comunicazione con altri computers e di sofisticate interfacce con l'utente.

### *Logica cablata e logica programmata*

Un sistema, progettato per automatizzare una macchina o un impianto, è costituito da tre parti essenziali:

- sezione di ingresso;
- sezione di elaborazione;
- sezione di uscita.

Le sezioni di ingresso e di uscita svolgono prevalentemente il compito di adattare i segnali elettrici che provengono dall'impianto e quelli che vanno a comandare gli apparati azionatori.

Il controllo delle condizioni che consentono la sequenza di attivazione a organi che costituiscono la macchina è eseguito dall'automatismo, identificabile con la sezione di elaborazione. È possibile realizzare un automatismo con componenti elettromeccanici (relè, temporizzatori) oppure utilizzando circuiti statici, cioè circuiti elettronici a piccola e media scala di integrazione che realizzano le funzioni logiche fondamentali (AND, OR, NOT), le funzioni di conteggio, addizione e memorizzazione. In base alla tecnologia adottata si parla di :

- sistema con logica cablata (di tipo elettromeccanico);
- sistema con logica cablata statica (di tipo elettronico).

Le reti di questo tipo sono reti combinatorie cioè reti in cui il valore che assume l'uscita (o le uscite) in un certo istante è funzione soltanto del valore che in quell'istante assumono gli ingressi.

Negli ultimi anni c'è stata una tendenza a sostituire i sistemi con logica a relè con i sistemi a logica statica. I principali motivi che hanno spinto verso questa innovazione sono i seguenti:

- i sistemi produttivi richiedono prestazioni sempre più sofisticate e ciò imporrebbe un aumento del volume e della complessità della sezione di elaborazione a causa del maggiore ingombro dei circuiti elettromeccanici rispetto a quelli statici;
- i relè hanno una durata limitata nel tempo se il numero di commutazioni è elevato, mentre i componenti statici, non avendo organi meccanici in movimento, sono praticamente esenti da usura;
- il tempo di commutazione dei relè è elevato rispetto a quello dei circuiti statici;
- l'impiego dei circuiti a relè è molto dispendioso e poco pratico quando bisogna realizzare funzioni complesse come il conteggio, la temporizzazione e la memorizzazione.

Per contro i circuiti a relè presentano alcuni vantaggi:

- affidabilità molto buona quando non è richiesto un elevato numero di commutazioni;
- alimentazione che non richiede particolari requisiti;
- ottima immunità ai disturbi;

- manutenzione e ricerca guasti che non necessitano di strumentazione particolare e costosa, né di personale particolarmente specializzato.

Tuttavia i sistemi a logica cablata, sia statici che elettromeccanici, hanno il grosso svantaggio di offrire scarsa flessibilità a eventuali modifiche della funzione logica.

Si tenga presente che molte volte è necessario apportare modifiche a causa di varianti od estensioni del progetto, per errori e lacune evidenziate in sede di primo impiego della macchina o dell'impianto. Ogni variazione al progetto comporta una modifica del cablaggio. Nel caso di circuiti statici integrati, le cui connessioni sono realizzate con circuiti stampati, una modifica al cablaggio non è facile da realizzare.

Gli svantaggi derivanti da un sistema rigido come quello a logica cablata sono poco sentiti nelle produzioni di serie, dove non si richiedono modifiche notevoli né in fase di messa in servizio né in fase operativa. Oggi però lo sviluppo di un progetto pesa molto rispetto a tutti gli altri fattori di costo, per cui la scarsa flessibilità dei sistemi cablati si fa sentire anche quando le varianti al progetto iniziale sono di piccola entità.

Per ovviare all'inconveniente della scarsa flessibilità si sono adottate soluzioni che impiegano apparecchiature molto simili a un elaboratore.

Un elaboratore è un'apparecchiatura che esegue una serie di comandi (o istruzioni) in sequenza, registrati precedentemente nella memoria elettronica dell'elaboratore stesso. L'insieme delle istruzioni, che svolgono una funzione richiesta, costituisce il programma applicativo. Risulta quindi evidente che, disponendo di un'apparecchiatura in grado di realizzare le principali funzioni logiche richieste per la conduzione di una macchina o di un processo secondo un determinato programma memorizzato, è possibile modificare un intero ciclo produttivo anche in fase operativa, semplicemente cambiando il programma.

L'innovazione di questi sistemi rispetto ai sistemi classici di automazione (sia a relè che statici) è rappresentata dalla programmabilità. L'impiego dei PLC ha contribuito al diffondersi della dizione "logica programmata", contrapposta a quella di "logica cablata".

## *Costruttori, criteri di scelta e impieghi*

Oggi sul mercato esistono diversi modelli di controllori a logica programmabile e tutte le maggiori case costruttrici sono presenti con un loro prodotto. Questi si differenziano a seconda delle necessità d'utilizzo: si va dai più semplici controllori programmabili (capaci di gestire solo ingressi e uscite digitali) a sistemi in grado di controllare processi molto complessi.

Fra le maggiori case costruttrici possiamo ricordare Siemens, Allen-Bradley, Telemecanique, Omron, C.G.E. e Hitachi. Oltre a questi "colossi" vi sono altre case, più piccole, comunque competitive presenti sul mercato come ad esempio Festo e Lovato.

La scelta fra un modello e l'altro è legata a molti fattori: il prezzo, la reperibilità sul mercato, l'affidabilità del prodotto e la conoscenza specifica del prodotto che possiede chi deve sviluppare il software. Per conoscenza specifica s'intende la conoscenza del softwareista relativamente al prodotto che usa, alla padronanza del linguaggio di programmazione (diverso per ogni casa costruttrice), dell'hardware che ha a disposizione e come meglio riesce a conciliare i prodotti con lo scopo prefissato che deve raggiungere. Risulta praticamente impossibile conoscere in modo approfondito i linguaggi di programmazione e la gamma di prodotti di ogni singola casa costruttrice per operare di conseguenza una scelta del prodotto da impiegare. La scelta di utilizzare prodotti Siemens, nel nostro caso, è determinata dal fatto che la casa detiene circa il 40% del mercato relativo dei PLC. L'esperienza insegna che una volta effettuata la scelta sulla casa costruttrice i tecnici si specializzano sempre di più sui prodotti della stessa e difficilmente si potrà pensare ad una scelta diversa.

I primi PLC nacquero per la gestione di soli ingressi e uscite digitali, con una "lista di istruzioni" (linguaggio di programmazione) composto da sole operazioni logiche. Pur con questi limiti, vi era il vantaggio di avere all'interno del PLC un notevole numero di temporizzatori, di contatori e memorie. Ci si svincolava poi dal numero di contatti fisici che possono essere messi a disposizione da un qualsiasi relè. Visto il successo sul mercato, le case costruttrici pensarono di ampliare le possibilità di utilizzo dei PLC per poterli utilizzare in controlli di processo sempre più complessi. Vennero così realizzate schede per la gestione di ingressi e uscite analogiche, per il controllo di temperatura, per il controllo di

dosaggio, per il posizionamento degli assi elettrici, ecc. che possono essere inserite direttamente nel rack del PLC dal quale prelevare l'alimentazione ad avere accesso diretto al bus di comunicazione con la CPU.

Negli ultimi anni, per quanto riguarda il posizionamento degli assi elettrici, si è notevolmente ampliata la gamma delle schede a disposizione, al punto tale da rendere sempre più sottile il confine esistente tra un PLC e un controllo numerico (CNC).

Questo sviluppo repentino dei PLC ha fatto sorgere un altro problema: la comunicazione uomo-macchina. Per questo motivo sono nati i "pannelli operatore", cioè dei dispositivi che, allacciati direttamente al bus del PLC, rendono possibile la modifica dei parametri di processo. È possibile, addirittura, collegare il PLC con un personal computer il quale assume il ruolo di "supervisore" nel controllo di processo, relegando al PLC il compito di esecutore.

Sul mercato, ogni casa costruttrice, dispone di una propria gamma di prodotti diversificati per potenzialità e costi, indirizzata a soddisfare le diverse esigenze. PLC è un termine generico che individua macchine che possono essere utilizzate sia per controlli di processo semplici che complessi, in base al modello di PLC utilizzato. Ogni casa dispone infatti di diversi modelli che differiscono per possibilità di ampliamento, velocità di scansione del programma, possibilità di utilizzo di schede speciali. Queste differenze fanno variare il costo di un PLC da poche centinaia di migliaia di lire fino a svariati milioni.

I PLC sono utilizzati nei più svariati settori sia civili che industriali. Ad esempio Siemens produce modelli di PLC per impieghi civili dedicati alla gestione degli ascensori ed alla gestione degli impianti di riscaldamento. Nell'ambito industriale vengono gestite linee di produzione, palettizzazione, immagazzinaggio ecc.

## *Confronto tra PLC e MicroProcessori*

Il controllore a logica programmabile è un apparecchiatura in grado di eseguire sequenzialmente un insieme ordinato di istruzioni scritte nella propria memoria. Le istruzioni hanno il compito di acquisire lo stato dei segnali digitali o analogici provenienti dalla macchina, elaborare questi segnali secondo preordinate funzioni logiche e matematiche e, in base ai risultati dell'elaborazione, determinare lo stato delle uscite che comandano gli attuatori della macchina.

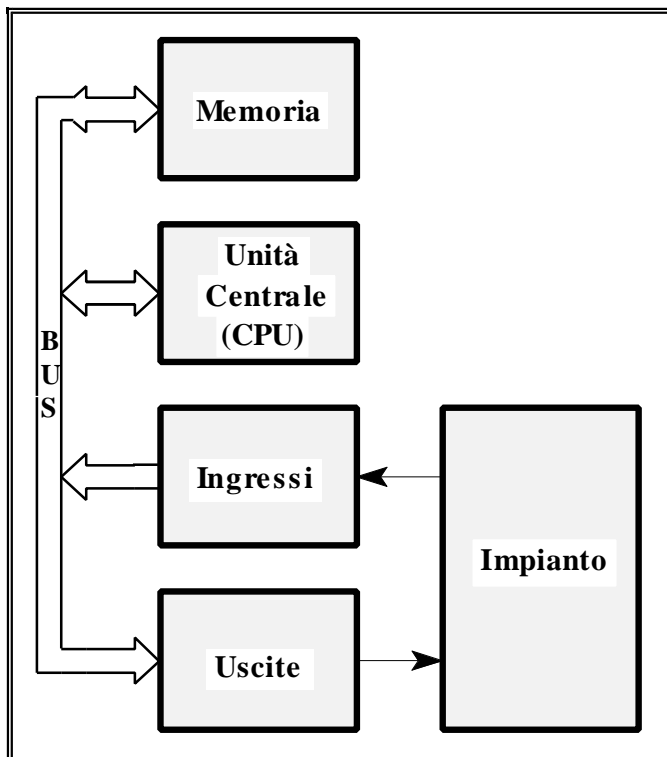
Queste modalità operative sono analoghe per i PLC e per i microprocessori quindi tutti i dispositivi citati possono avere impieghi equivalenti. La fondamentale differenza sta nel linguaggio di programmazione. Il PLC permette di programmare con un linguaggio evoluto, non in assembler come per i microprocessori, in quanto all'interno della CPU dei PLC vi è un compilatore da linguaggio di programmazione ad assembler. Si può dire perciò che la programmazione non avviene in micro istruzioni (tipiche del linguaggio assembler), ma tramite macro-istruzioni. Questo fattore penalizza, per certi versi, le prestazioni di un PLC rispetto a quelle di un microprocessore in particolare la velocità di esecuzione del programma applicativo (nei PLC questo problema può essere risolto utilizzando più processori che lavorano contemporaneamente, ed eseguono un continuo scambio di dati).

Anche i costi, a parità di prestazioni, sembrerebbero privilegiare un sistema a microprocessore ma la difficoltà di programmazione in linguaggio assembler fa sì che i PLC siano maggiormente impiegati nell'automazione sia civile che industriale.

## *Architettura di un PLC*

Un PLC è costituito fondamentalmente da cinque parti (figura 1):

- sezione **CPU** (Central Processing Unit), è il cervello del sistema, prende tutte le decisioni logiche;
- sezione di **MEMORIA**, è il supporto fisico dove sono registrate le istruzioni che costituiscono il programma applicativo e i dati necessari per funzioni ausiliarie;
- sezione di **INGRESSO**, riceve i segnali elettrici (digitali o analogici) provenienti dall'impianto (finecorsa, pressostati, pulsanti, selettori, ecc.);
- sezione di **USCITA**, riceve i segnali prodotti dall'elaborazione e li adatta per comandare con potenze adeguate i vari organi attuatori (motori, elettrovalvole, teleruttori, frizioni, visualizzatori, lampade, ecc.);
- **BUS**, consente la comunicazione tra le varie sezioni.



**Figura 1** - L'hardware di un PLC è costituito dalla sezione Unità Centrale, dalla sezione di Memoria e dalle sezioni di Ingresso/Uscita. I dati e gli indirizzi scambiati tra i blocchi transitano su un BUS di comunicazione.

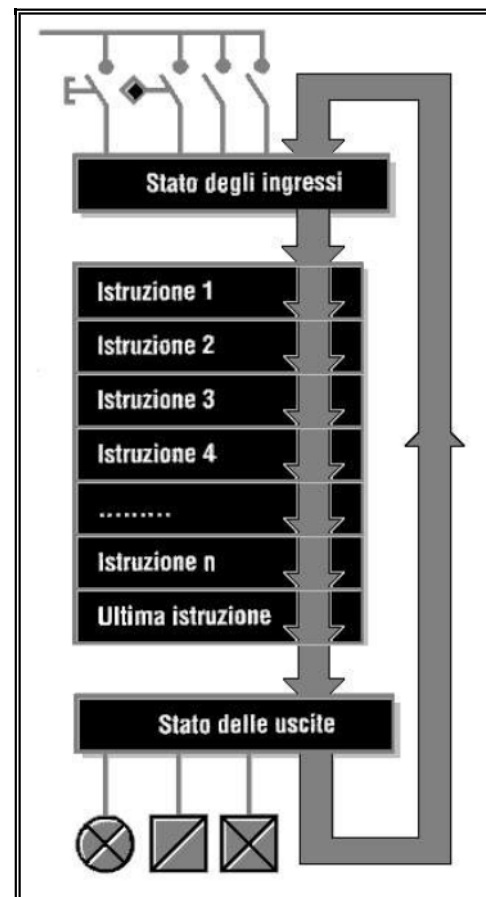
### Unità centrale di processo (CPU)

La CPU, Central Processing Unit o unità centrale costituisce l'intelligenza del PLC. Essa legge lo stato dei segnali di ingresso provenienti dalla macchina da controllare ed esegue, in sequenza, le istruzioni registrate in memoria. In base ai risultati dell'elaborazione provvede ad aggiornare lo stato delle uscite del sistema (figura 2).

Le prime CPU erano costruite cablando molti componenti discreti; oggi sono state sostituite dal microprocessore, che ha notevolmente migliorato le prestazioni di velocità e potenza di elaborazione e ha drasticamente ridotto l'ingombro.

Lo scambio di informazioni tra il microprocessore e le sezioni I/O e di memoria avviene mediante la trasmissione, su una struttura chiamata BUS, di una parola alla volta, costituita da un certo numero di bit: per esempio 4,8,16,32; da qui la dizione di microprocessori a 4,8,16,32 bit.

**Figura 2** - La CPU legge lo stato dei segnali d'ingresso provenienti dalla macchina da controllare, esegue in sequenza le istruzioni registrate in memoria e, in base ai risultati logici dell'elaborazione provvede ad aggiornare lo stato delle uscite del sistema.



## Memoria

Normalmente si intende per memoria del PLC solamente la memoria che contiene il programma applicativo. In realtà in un PLC la memoria è suddivisa in diverse zone, ciascuna adibita a compiti differenti. Anche se le soluzioni adottate dai costruttori sono molto diverse fra loro, si può in generale considerare la memoria suddivisa nelle seguenti parti (figura 3):

- memoria per il programma utente;
- memoria per le funzioni interne;
- memoria per lo stato degli ingressi e delle uscite;
- memoria per il software di sistema (sistema operativo).

La quantità di memoria disponibile per contenere il programma utente e i dati varia secondo il modello e la casa costruttrice del PLC.

L'estensione in memoria è misurata in K words, dove K è l'abbreviazione di kilo; in questo ambiente, anziché 1000 vale  $1024 (2^{10})$ . Una Word in genere può essere di 8 o 16 bit. Se un costruttore indica una memoria massima di 8K significa  $8 \times 1024$  word, la quantità di memoria coincide con il numero di istruzioni inseribili in memoria.



**Figura 3** - La memoria del PLC è suddivisa in diverse zone, ciascuna adibita a compiti differenti:

- memoria per il programma utente;
- memoria per le funzioni interne;
- memoria per lo stato degli ingressi e delle uscite;
- memoria per il software di sistema .

### Memoria per il programma utente

La memoria per il programma utente contiene le istruzioni che costituiscono il programma. Esse sono registrate in memoria una dopo l'altra, opportunamente codificate in una sequenza di bit. La CPU legge ed esegue ciascuna delle istruzioni seguendo un ciclo di scansione che parte dalla prima istruzione fino all'ultima, per poi ricominciare da capo (figura 4).

Questa memoria è di tipo RAM, di solito realizzata con tecnologia CMOS. Essendo una memoria volatile è necessario salvaguardare il contenuto nel caso di mancanza improvvisa di tensione; per questo motivo è prevista una alimentazione supplementare, con una piccola batteria tampone al litio.

**Figura 4** - Le istruzioni che costituiscono il programma sono registrate in memoria una dopo l'altra, in modo sequenziale. Ogni istruzione è codificata in binario. Nello esempio a ogni istruzione logica corrisponde una parola binaria di 16 bit. I 4 bit più significativi rappresentano il codice operativo (STR=1010, ANDNOT=1011, ecc.). I rimanenti bit rappresentano l'indirizzo dell'operando o l'operando stesso.

Locazioni di memoria	CODICE MNEMONICO	Locazioni di memoria	CODICE BINARIO
0	STR 001	0	1010000000000001
1	AND NOT 002	1	1011000000000010
2	OUT 040	2	1100000000101000
3	STR NOT 040	3	1110000000000010
...	AND 002	...	1101000000000100
99	OUT 041	99	1100000000101001
100		100	
101		101	
...		...	
...		...	
4094		4094	
4095		4095	

## **Memoria per le funzioni interne**

La memoria per le funzioni interne è ancora una memoria indirizzabile dall'utente ma che contiene i dati necessari all'elaborazione logica e alle funzioni ausiliarie e i risultati intermedi prodotti da operazioni aritmetiche e booleane (logiche).

Per esempio, nel normale funzionamento di un PLC può essere necessario non trasferire direttamente in uscita il risultato di una funzione logica, ma immagazzinare tale valore in un bit di memoria per poterlo utilizzare come variabile di ingresso in altre equazioni logiche.

Lo stato di questo risultato intermedio è detto bit interno di immagazzinamento, flag, merker, bobina interna, coil.

I risultati intermedi di espressioni aritmetiche sono costituiti da parole interne, cioè da una serie di bit organizzati a gruppi di 4, 8, 16, 32.

I PLC dispongono di funzioni standard come i temporizzatori, i contatori e i registri a scorrimento (shift register) che sono di grande aiuto nella programmazione di sistemi di controllo. I valori numerici relativi a queste funzioni, per esempio il numero che esprime il tempo in una funzione logica di ritardo, oppure il numero che rappresenta il valore massimo da conteggiare, devono essere memorizzati in una parte della memoria per le funzioni interne. Anche questa memoria è di tipo RAM CMOS, con batteria tampone.

## **Memoria per lo stato degli ingressi e delle uscite**

Una zona di memoria del PLC è riservata agli ingressi e alle uscite. Questa zona, ha una allocazione di tipo statico, cioè i dati I/O sono registrati sempre negli stessi indirizzi; questo a differenza dei programmi utente e dei dati per funzioni interne che sono allocati dinamicamente e cioè nelle zone di memoria che di volta in volta sono libere. In questi indirizzi viene registrata all'inizio di ogni ciclo di scansione un'immagine dello stato degli ingressi; in base al loro valore e alla funzione logica voluta dal programma utente è aggiornata la zona che contiene l'immagine delle uscite.

Il valore di uno stato di ingresso o di uscita è molte volte di tipo on-off. Per leggere con facilità lo stato di un ingresso o per comandare un'uscita i PLC sono progettati con un hardware opportuno, che consente di indirizzare singolarmente ogni bit della memoria immagine di I/O.

## **Memoria per il software di sistema**

I controllori a logica programmabile (PLC) sono dotati anche di una memoria non volatile di tipo ROM o EPROM, nella quale è memorizzato il sistema operativo, cioè una serie di programmi definiti dal costruttore indispensabili per gestire il corretto funzionamento di tutto l'hardware.

Senza sistema operativo un controllore a logica programmabile, così come qualsiasi calcolatore, non sarebbe altro che un insieme di circuiti elettronici senza vita.

Le funzioni principali del sistema operativo sono:

- diagnostica del sistema, intesa come segnalazione di guasti, presenza dell'alimentazione, ecc.;
- controllo della scansione e della durata del programma utente;
- caricamento dei programmi applicativi da dispositivi di programmazione esterni al controllore;
- pilotaggio di eventuali periferiche collegate al controllore programmabile attraverso interfaccia di comunicazione.

## ***Sezioni di ingresso e uscita: input/output (I/O)***

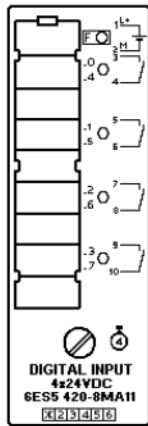
Il PLC è un apparecchiatura fortemente orientata all'interfacciamento con dispositivi industriali. Le sue sezioni di ingresso e uscita devono essere compatibili con una vasta gamma di segnali elettrici che vanno dai 5V DC ai 220V AC, dai segnali digitali a quelli analogici.

La disponibilità di diverse soluzioni di I/O è uno degli elementi più importanti per valutare le prestazioni di un modello o di una marca di PLC; inoltre è un parametro che determina fortemente il costo finale della configurazione.

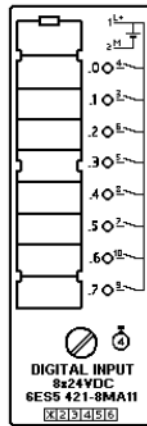
I moduli di I/O, dovendo rispondere alle esigenze più diversificate, sono disponibili oramai per tutti i modelli di PLC in una vasta gamma, con caratteristiche più o meno standard, inseribili in uno qualsiasi degli slots per l'I/O presenti nel rack del PLC.

Per capire meglio le caratteristiche tecniche delle unità I/O si devono esaminare i cataloghi dei costruttori in cui sono presentate alcune unità di ingresso e di uscita.

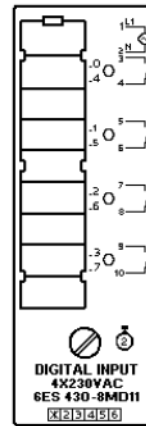
### Blocchi ingressi digitali



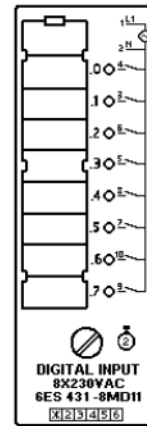
6ES5 420-8MA11  
430-8MB11



6ES5 421-8MA11



6ES5 430-8MC11  
430-8MD11



6ES5 431-8MC11  
431-8MD11

Dati tecnici				
Ingressi digitali	6ES5 430-8MC11	6ES5 430-8MD11	6ES5 431-8MC11	6ES5 4318MD11
<b>Numero degli ingressi</b>	<b>4</b>	<b>4</b>	<b>8</b>	<b>8</b>
Separazione di potenziale - per gruppi di	si (optoisolatori) 4	si (optoisolatori) 4	si (optoisolatori) 4	si (optoisolatori) 4
<b>Tensione d'ingresso</b>				
- valore nominale	AC 115 V	AC 230 V	AC 115 V	AC 230 V
- frequenza	47÷63Hz	47÷63Hz	47÷63Hz	47÷63Hz
Tensione d'ingresso				
- per segnale "0"	0÷40V	0÷70V	0÷40V	0÷95V
- per segnale "1"	85÷135V	170÷264V	85÷135V	195÷253V
Corrente d'ingresso				
- per segnale "1"	tip. 10mA (a 115Vac) 6mA (a 115Vdc)	15mA (a 230Vac) 2,5mA (a 230Vdc)	12mA (a 115Vac) 2,5mA (a 115Vdc)	16mA (a 230Vac) 1,8mA (a 230Vdc)
Tempo di ritardo				
- per "0" -> "1"	tip. 10ms	10ms	10ms	5ms
- per "1" -> "0"	tip. 20ms	20ms	20ms	15ms
Lunghezza dei conduttori non schermati	max 100m	100m	100m	100m
Corrente di riposo per collegamento con interruttori BERO a due fili	max 5mA	5mA	4mA	5mA
Segnalazione di guasto "manca generatore di tensione"	no	no	no	no
Tensione d'isolamento (attacchi esterni verso guida, attacchi interni, altri blocchi)				
- secondo VDE 0160	AC 250V	AC 250V	AC 250V	AC 250V
- valore di prova	AC 1500V	AC 1500V	AC 1500V	AC 1500V
Assorbimento di corrente				
- interno (a 9V, da CPU o interfaccia IM) max	16mA	16mA	32mA	32mA
Potenza dissipata	2,8W	2,5W	2,5W	3,6W
Peso	0,21Kg	0,21Kg	0,26Kg	0,26Kg



## Blocchi uscite digitali

Dati tecnici				
Uscite digitali	6ES5 440-8MA11	6ES5 440-8MA21	6ES5 441-8MA11	6ES5 450-8MB11
<b>Numero delle uscite</b>	<b>4</b>	<b>4</b>	<b>8</b>	<b>4</b>
Separazione di potenziale - a gruppi di	no -	no -	no -	si (optoisolatori) 4
<b>Tensione di Alimentazione Up, Us</b> (per il carico)				
- valore nominale	DC 24 V	DC 24 V	DC 24 V	DC 24/60 V
- ondulazione Uss	3,6 V	3,6 V	3,6 V	3,6 V
- campo ammissibile (inclusa ondulazione) valore per t<0,5 s	20÷30V 35V	20÷30V 35V	20÷30 V 35V	20÷75V 87V
<b>Corrente di uscita per segnale "1"</b>				
- valore nominale	0,5A	2A	0,5A	0,5A
- campo ammissibile (1)	5mA÷0,5A	5mA÷2A	5mA÷0,5A	5mA÷0,5A
- funzionamento parallelo	max 0,4A	1,6A	0,4A	non consentito
Carico di lampade	max 5W	10W	5W	5W
Protezione da cortocircuito	elettronica	elettronica	no	elettronica
Limitazione dell'extratensione induttiva di apertura (interna)	a -15V	-15V	-15V	-30V
Frequenza di commutazione per				
- carico ohmmico	max 100Hz	100Hz	100Hz	100Hz
- carico induttivo	max 2Hz	0,5Hz	2Hz	0,5Hz
Caricabilità complessiva (2)	100%	50%	100%	50%
Corrente residua con segnale "0"	max 0,5mA	0,5mA	0,5mA	0,5mA
Livello di segnale alle uscite				
- con segnale "0"	max +3V	+3V	+4,8V	+3V
- con segnale "1"	min Up -1,2V	Up -1,5V	Up -1,2V	Up -2,5V
Lunghezza dei conduttori non schermati	max 100m	100m	100m	100m
Segnalazione di guasto "Cortocircuito, manca Up/Us"	Led rossi	Led rossi	no	Led rossi
Tensione d'isolamento (attacchi esterni verso guida, attacchi interni, altri blocchi)				
- secondo VDE 0160	-	-	-	-
- valore di prova	-	-	-	AC 500V
Assorbimento di corrente				
- interno ( dal'unità centrale; 9V)	max 15mA	15mA	14mA	15mA
- esterno (a 24V, senza carico)	max 25mA	25mA	15mA	30mA
Potenza dissipata	tip. 3W	4,8W	3,5W	5W
Peso	ca. 0,2Kg	0,2Kg	0,22Kg	0,2Kg

(1) Un ingresso di un blocco ingressi digitali è consentito come carico minimo.

(2) Riferita alla somma di tutte le correnti nominali.

## Unità di INPUT

Il data sheet di pagina 7 descrive quattro tipi di unità di ingresso digitali. Una unità di ingresso digitale converte i segnali binari (di ON - OFF) del processo da controllare al livello dei segnali interni del controllore, che tipicamente sono segnali elettrici TTL (0÷5V). Lo stato dei segnali è visualizzato con LED verdi o rossi presenti sulla parte frontale della scheda.

Per ogni unità di ingresso sono specificati i dati tecnici. Esaminiamo quelli più significativi.

**Numero di ingressi :** è il numero di ingressi collegabili a ciascuna unità. Per i moduli descritti a pagina 7 due unità sono a quattro ingressi e due a otto. Ci sono anche PLC con unità a 16 ingressi e altre unità, denominate ad alta densità, che hanno 32 punti d'ingresso.

Notiamo che le quattro unità illustrate sono caratterizzate da ingressi con separazione a optoisolatori. Questo significa che esiste un disaccoppiamento elettrico tra i circuiti esterni e quelli interni, allo scopo di proteggere i circuiti elettrici del PLC da eventuali malfunzionamenti distruttivi legati al processo come sbalzi di tensione o sovraccarichi di corrente.

**tensione d'ingresso :** è il valore nominale di tensione che può essere applicato a ciascun ingresso. Se il valore di tensione è inferiore a una determinata soglia, quel segnale è interpretato come 0 logico; se il valore di tensione è superiore, il segnale è considerato 1 logico. Per ciascuna unità sono specificati gli intervalli di tensione in cui il segnale vale 1 oppure 0 logico.

**tempo di ritardo :** un segnale di ingresso viene acquisito correttamente se il suo livello logico si mantiene per un tempo superiore al tempo di ritardo. È importante valutare questo parametro quando bisogna acquisire segnali che potrebbero variare velocemente, per non rischiare di perdere delle informazioni. Esistono comunque allo scopo unità speciali di conteggio veloce.

## Unità di OUTPUT

Una unità di uscita digitale converte i segnali, interni al PLC e prodotti dall'elaborazione logica, al livello dei segnali esterni del processo comandato. Anche le unità di uscita visualizzano lo stato dei segnali con LED verdi o rossi, situati sulla parte frontale della scheda.

Il data sheet di pagina 8 descrive le caratteristiche di quattro unità di uscita digitali; commentiamo i dati tecnici più significativi.

**Numero delle uscite:** è il numero di punti di uscita disponibili sulla scheda, ai quali sono collegati gli organi attuatori. Sono disponibili per quasi tutti i modelli unità a 4, 8, 16, 32 punti di uscita.

**Tensione di alimentazione:** è il valore nominale di tensione fornito dal singolo punto di uscita.

**Corrente di uscita:** è il valore nominale di corrente che un singolo punto di uscita può fornire. Esistono schede da 0.1A, 0.5A, 1A, 2A. È importante, nella scelta degli utilizzatori, che tale limite di corrente non venga superato. A questo scopo, tra i dati tecnici, sono dichiarate anche le potenze ammissibili con carichi resistivi e carichi induttivi.

**Frequenza di commutazione:** stabilisce la massima velocità di commutazione con cui è possibile comandare un attuatore.

I PLC, dovendo coprire una vastissima gamma di campi applicativi, da semplici sistemi logici per piccole macchine operatrici a complessi sistemi sequenziali per l'automazione di grandi impianti, sono stati diversificati secondo la capacità, intesa come il massimo numero di punti di ingresso e uscita collegabili a un'unica apparecchiatura. In base a questo parametro i PLC sono normalmente suddivisi in tre fasce:

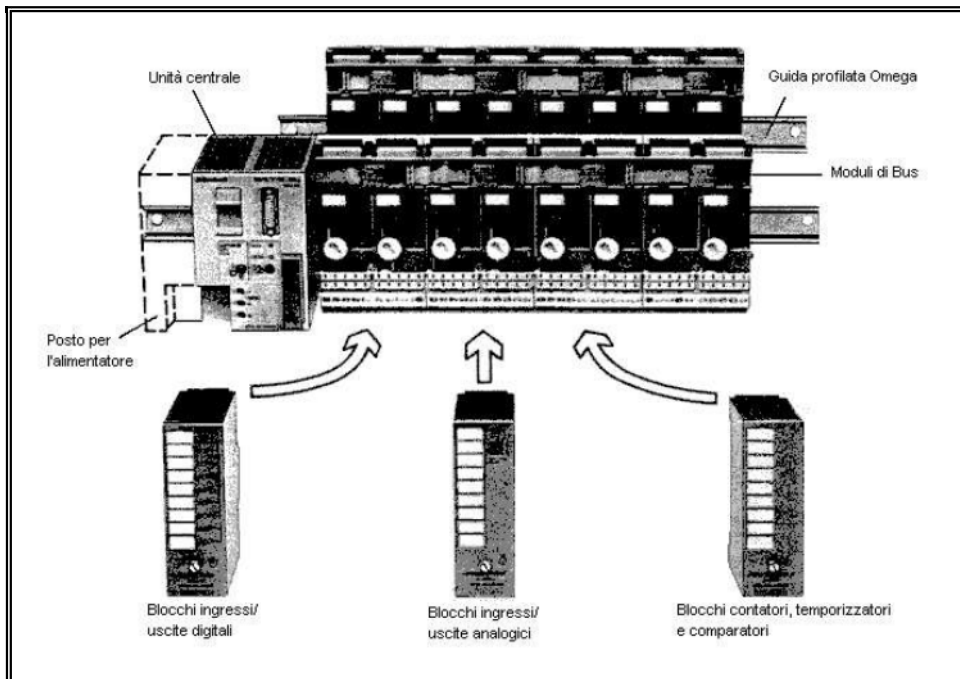
- piccoli: da 0 a 128 I/O;
- medi da 129 a 512 I/O;
- grandi da 513 a 1024 I/O e oltre.

Bisogna stare attenti però a non considerare il numero di I/O come l'unico fattore di importanza per un PLC: ci sono infatti anche altri parametri come la capacità di calcolo, la facilità di programmazione, la quantità di memoria, ecc.

## Caratteristiche meccaniche

Dal punto di vista costruttivo il PLC si presenta come un rack robusto e protetto. Lo chassis può alloggiare un numero di schede (moduli) variabile da modello a modello.

Ogni scheda occupa normalmente uno slot e il tipo di scheda determina la configurazione hardware del PLC. La figura 5 dimostra la flessibilità di configurazione hardware del PLC.



**Figura 5** - Ogni scheda occupa normalmente uno slot. È possibile scegliere tra vari tipi di schede, in modo da realizzare la configurazione più opportuna per ogni applicazione.

Il rack contiene di solito una scheda per l'alimentazione, una scheda CPU e più schede di I/O. Il numero di schede di I/O è espandibile fino a completare il numero di slots disponibili nel rack e in alcuni modelli si possono aggiungere ulteriori racks.

Il numero di racks collegabili all'unità centrale del PLC è limitato e varia da modello a modello. I collegamenti tra più racks dello stesso sistema sono effettuati mediante appositi cavi, mentre all'interno di ciascun rack il collegamento tra le varie schede di I/O e la scheda CPU avviene tramite un BUS posto sulla parte posteriore del rack.

I racks devono essere installati in armadi con fessure di aerazione grado di protezione IP2 in locali puliti e asciutti) o in cabine asciutte (armadi tipo IP54), oppure su telaio in ambienti operativi puliti e asciutti.

## Software dei controllori logici programmabili

### Introduzione

il programma secondo il quale il PLC funziona è contenuto nella memoria utente, le istruzioni sono registrate una dopo l'altra.

Il funzionamento del PLC è ciclico, cioè la CPU per elaborare il programma legge ed esegue in sequenza le istruzioni, partendo dalla prima. Dopo l'elaborazione dell'ultima istruzione l'esecuzione ricomincia da capo.

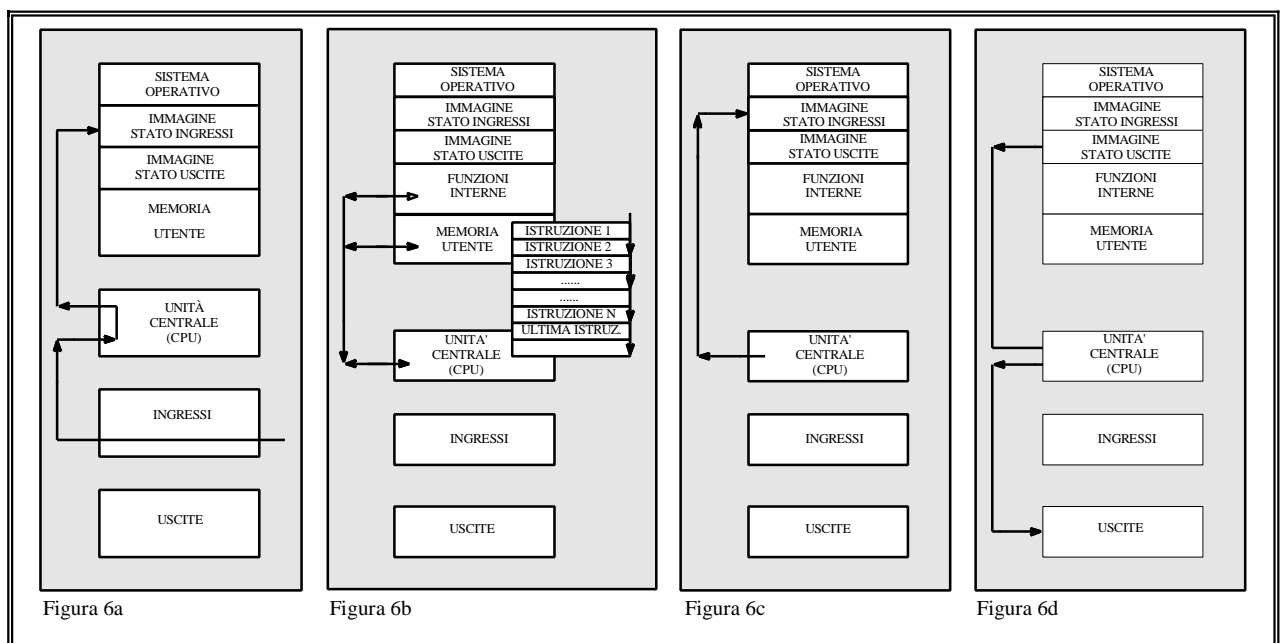
All'inizio di ogni ciclo di scansione la CPU interroga i segnali d'ingresso e li registra nella memoria immagine dello stato degli ingressi, in modo da creare una "fotografia" della situazione del processo. Durante l'elaborazione del programma tutte le interrogazioni sul valore degli ingressi fanno

riferimento a questa memoria immagine. In questo modo lo stato dei segnali di ingresso si conserva inalterato per un intero ciclo di elaborazione, cosicché i cambiamenti di segnale che si verificano durante il ciclo non possono creare disturbi.

I risultati prodotti dall'elaborazione durante il ciclo sono registrati temporaneamente nella memoria immagine dello stato delle uscite. Solo al completamento del ciclo di programma il contenuto della memoria immagine è trasferito alle corrispondenti unità di uscita; in seguito viene attivato un nuovo ciclo di elaborazione. Il tempo necessario per scandire tutte le istruzioni del programma viene chiamato tempo di ciclo. È un parametro importante per individuare la potenza di un PLC ed è in stretta dipendenza dalla velocità di lavoro della CPU e dal tipo di istruzioni. Per poter paragonare i tempi di ciclo di modelli diversi di PLC, ci si riferisce a una lunghezza standard di programma di 1024 (cioè  $2^{10}$ ) locazioni di memoria.

L'ordine di grandezza dei tempi di ciclo per  $2^{10}$  istruzioni va da qualche millisecondo alle decine di millisecondi. Si comprende che se il tempo di ciclo è molto breve è possibile cogliere in tempo reale tutte le variazioni dei segnali del processo senza perdere informazioni.

La figura 6 illustra il principio di funzionamento suddiviso nelle fasi principali.



**Figura 6a** - Prima di iniziare la scansione del programma applicativo, la CPU interroga i segnali d'ingresso e li registra nella memoria immagine dello stato degli ingressi. Durante l'elaborazione del programma tutte le interrogazioni sul valore degli ingressi fanno riferimento a questa memoria immagine.

In questo modo lo stato dei segnali d'ingresso si conserva inalterato per un intero ciclo di elaborazione, cosicché i cambiamenti di segnale che si verificano durante il ciclo non possono creare disturbi.

**Figura 6b** - Il funzionamento del PLC è ciclico, cioè la CPU per elaborare il pro-

gramma legge ed esegue in sequenza le istruzioni, partendo dalla prima fino all'ultima istruzione del programma.

**Figura 6c** - I risultati prodotti dall'elaborazione durante il ciclo sono registrati temporaneamente nella memoria immagine dello stato delle uscite.

**Figura 6d** - Solo al completamento del ciclo di programma il contenuto della memoria immagine è trasferito alle corrispondenti unità; in seguito viene attivato un nuovo ciclo di elaborazione. Il tempo necessario per scandire tutte le istruzioni del programma viene chiamato tempo di ciclo.

## *Software dei PLC - linguaggi*

Il software è l'elemento determinante per dare all'hardware del PLC quella flessibilità che i sistemi di controllo a logica cablata non possiedono. Infatti è sufficiente cambiare solamente il programma affinché una certa configurazione hardware del PLC sia adatta a svolgere un compito di controllo diverso.

Il software che le aziende producono per i loro PLC si avvale quasi esclusivamente di linguaggi di programmazione specifici per le applicazioni di automazione, caratterizzando definitivamente il PLC come un calcolatore per il controllo industriale. Il motivo per cui sono stati creati tali linguaggi sta nel fatto che linguaggi ad alto livello tipo Fortran, Basic, Pascal, pur facilitando la stesura dei programmi, non sono adatti alle applicazioni di controllo, in quanto non esiste un rapporto immediato fra le istruzioni e l'azione nel processo.

I linguaggi di programmazione dei PLC sono orientati ai problemi di automazione (sono linguaggi di tipo POL, cioè Problem Oriented Languages) e utilizzano soprattutto istruzioni di logica combinatoria (AND, OR, NOT); di caricamento e trasferimento di segnali di processo, che operano sullo stato dei singoli segnali (elaborazione di bit); e funzioni standard di conteggio, temporizzazione e memorizzazione. Nei modelli di classe medio-grande sono disponibili istruzioni che manipolano i dati organizzati in byte e in word (elaborazione a byte e a word); tipicamente si tratta di istruzioni di caricamento e di confronto.

Quasi tutti i costruttori hanno diversificato i linguaggi per PLC proponendo tre rappresentazioni diverse, ciascuna applicabile in funzione delle specificità del problema e della particolare esperienza professionale dell'utente. Le tendenze di mercato sono, in ogni caso, prevalentemente due.

- Linguaggio KOP, noto anche come schema a contatti o rete ladder, che ha una certa somiglianza con gli schemi elettrici a relè. E' il primogenito tra i linguaggi utilizzati dato che, inizialmente, il PLC andava a sostituire i normali quadri a logica cablata che utilizzavano relè.

- Linguaggio AWL o lista di istruzioni, che utilizza abbreviazioni mnemoniche delle descrizioni delle funzioni. Il programma assomiglia ad un listato Assembly.

- Il terzo tipo di linguaggio utilizza una rappresentazione delle istruzioni nella quale i compiti dell'automazione sono espressi in forma grafica a blocchi con i simboli conformi alle norme DIN 40700 e DIN 40719. Questi linguaggi sono chiamati blocchi funzionali o FUP.

## Lista istruzioni (AWL)

La lista di istruzioni o AWL è il linguaggio più utilizzato grazie alle maggiori potenzialità di programmazione disponibili. Si approfondirà quindi questo tipo di software.

L'AWL (lista di istruzioni) fa uso di simboli, cioè di abbreviazioni mnemoniche, delle funzioni. I simboli sono spesso le iniziali del nome della funzione, in modo che sia agevole ricordare il significato.

Le case costruttrici non hanno unificato i simboli usati, per cui il programmatore è costretto a utilizzare abbreviazioni in tedesco, in inglese, in francese o in italiano (nel caso Siemens si utilizzano abbreviazioni dal tedesco).

Un programma in linguaggio AWL è composto da una sequenza di istruzioni e assomiglia ad un listato assembly. La figura 7 fornisce un esempio di come è strutturato un programma in AWL.

Un'istruzione è l'elemento più piccolo del programma ed è costituita da:

- una **operazione** o codice operativo, che individua cosa bisogna fare. Per esempio U = UND in tedesco individua una operazione logica AND (collegamento serie).
- un **operando** che indica il segnale sul quale bisogna operare. Per esempio E 2.3 (E = Eingang in tedesco) indica il segnale d'ingresso 2.3.

Un operando è a sua volta costituito da:

- una **sigla dell'operando (o codice dell'operando)**.

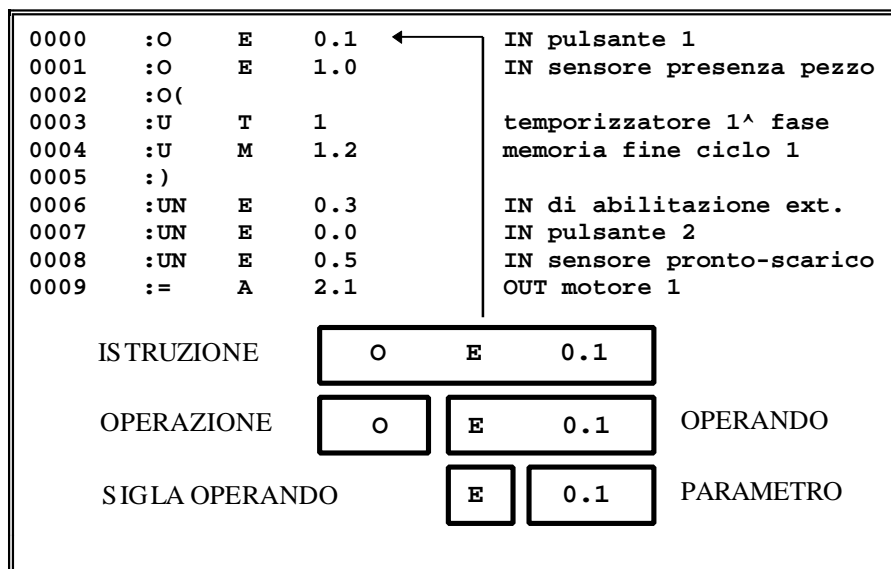
per esempio:

E = eingang = ingresso

A = ausgang = uscita

- un **parametro**.

Per esempio, può indicare il numero dell'ingresso o dell'uscita interessato dall'operazione.



**Figura 7** - Un programma in linguaggio AWL è composto da una sequenza di istruzioni. Un'istruzione è l'elemento più piccolo del programma ed è costituita da un'operazione (o codice operativo) e da un operando. L'operando può essere costituito, nel caso più generale, da una sigla e da un parametro. Il commento a destra delle istruzioni risulta molto importante perché facilita la lettura del listato.

## *Fasi di esecuzione di un programma*

L'esecuzione di un programma avviene in tre fasi diverse:

**Prima fase:** l'unità di governo legge gli ingressi, forma il PAE (immagine di processo degli ingressi);

**Seconda fase:** utilizzando il valore degli ingressi riportati nel PAE, l'unità di governo esegue il programma, e nel contempo forma il PAA (immagine di processo delle uscite);

**Terza fase:** L'unità di governo aggiorna le uscite leggendo i valori riportati dall'esecuzione del programma nel PAA.

All'accensione vengono azzerati tutti i contatori, i temporizzatori ed i merker non ritentivi. Dopo che il programma è stato caricato e lanciato, il PLC esegue in modo ripetitivo tutte le istruzioni, dalla prima all'ultima e poi riprende dalla prima e ripete ciclicamente il programma.

Durante l'esecuzione del programma, la CPU comunica con i blocchi esterni solo due volte, La prima comunicazione avviene all'inizio del programma, per acquisire i dati di input dai moduli di ingresso, con questi valori si crea l'immagine dello stato degli ingressi PAE, che verrà utilizzata durante tutta l'esecuzione del programma. Quindi se un'ingresso varia durante l'esecuzione del programma, dopo che il PAE è stato caricato, non produce nessuna influenza sul risultato di quel ciclo di programma.

I risultati che vengono ottenuti durante l'esecuzione del programma, non influenzano immediatamente le uscite del PLC, ma vengono memorizzati formando il PAA, immagine delle uscite, dopodiché avviene la seconda comunicazione, al termine dell'esecuzione del programma, per aggiornare i blocchi di output con i risultati delle elaborazioni contenuti nel PAA. Dopo l'aggiornamento delle uscite si riprende dalla prima fase.

## *Linguaggio di programmazione AWL STEP 5 - SIEMENS*

Il linguaggio di programmazione step 5 distingue tre tipi di operazioni:

- operazioni fondamentali;
- operazioni integrative o complementari;
- operazioni di sistema.

<i>LINGUAGGIO DI PROGRAMMAZIONE STEP 5</i>			
	<b>Operazioni fondamentali</b>	<b>Operazioni complementari</b>	<b>Operazioni di sistema</b>
<b>Campo d'impiego</b>	in tutti i blocchi	solo nei blocchi Funzionali (FB)	solo nei blocchi Funzionali (FB)
<b>Tipo di rappresentazione</b>	AWL, KOP, FUP	AWL	AWL
<b>Note</b>			per utenti con buone conoscenze del sistema

## Repertorio degli operandi

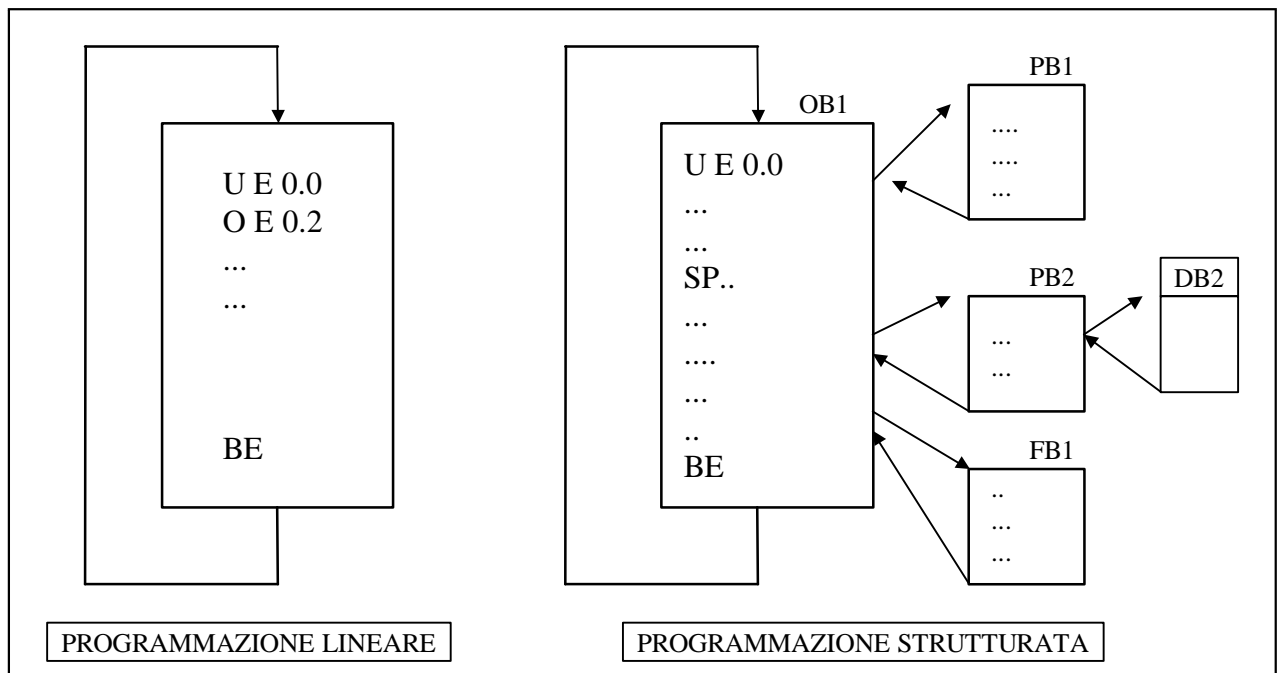
Il linguaggio di programmazione step 5 riconosce i seguenti operandi:

<i>OPERANDI</i>	<i>significato</i>	<i>breve descrizione</i>
<b>E</b>	Ingressi	Interfacce da processo a controllore
<b>A</b>	Uscite	Interfacce da controllore a processo
<b>M</b>	Merker	Memoria per risultati binari intermedi
<b>D</b>	Dati	Memoria per risultati digitali intermedi
<b>T</b>	Temporizzatori	Memoria per attuare temporizzazioni
<b>Z</b>	Contatori	Memoria per attuare conteggi
<b>P, Q</b>	Periferie	Interfacce da processo a controllore
<b>K</b>	Costanti	Valori numerici definiti
<b>OB, PB, SB, FB, DB.</b>	Blocchi	Ausili per strutturare il programma

## *Organizzazione del programma - ISTRUZIONI DI SALTO*

Il programma può essere organizzato in due forme diverse: quella lineare e quella strutturata. La programmazione lineare utilizza un solo blocco all'interno del quale è contenuto tutto il programma, è comodo utilizzare questo metodo quando il programma è breve e semplice.

La programmazione strutturata suddivide il programma in più parti, con la possibilità di saltare da una parte all'altra in diversi modi. Tale metodo di programmazione è il più utilizzato in quanto permette una larga flessibilità, una facile rilettura e quindi immediata individuazione dei guasti. Permette, inoltre, la separazione di varie funzioni all'interno del programma stesso e, potendo decidere quando richiamare tali funzioni si riduce il tempo necessario per l'esecuzione del programma.





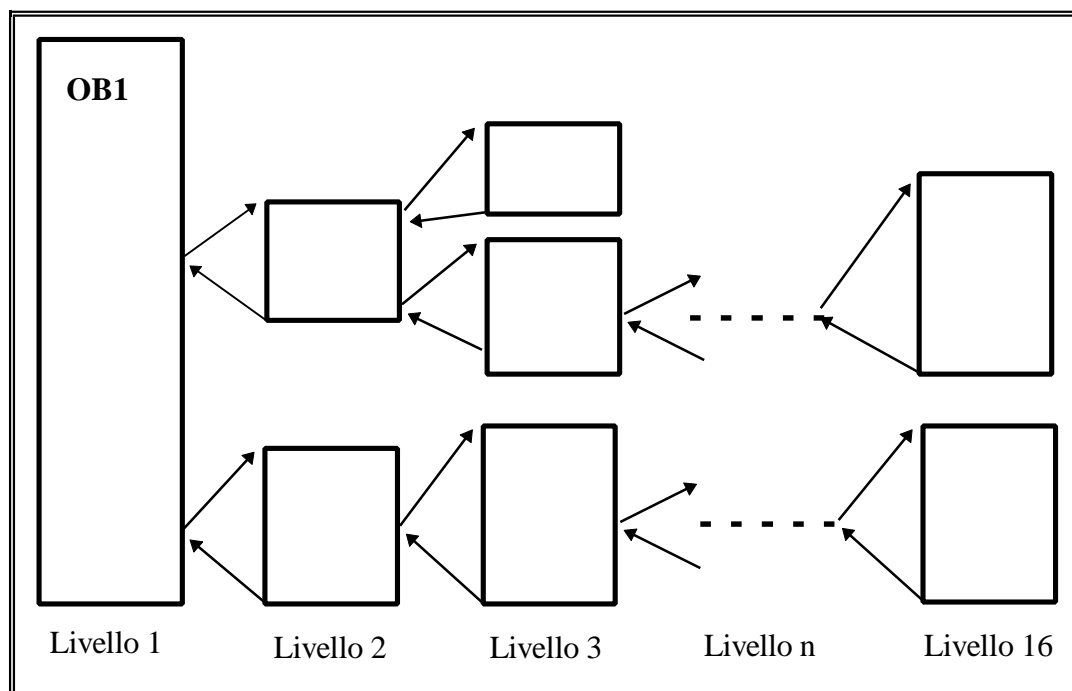
### **Blocchi disponibili nei controllori S5-100U e S5-95U**

Si analizza in modo particolare il linguaggio Step 5 riferito ai controllori S5-100U e S5-95U perché presenti in laboratorio (in ogni caso la programmazione nei vari modelli di PLC S5 Siemens differisce solo per alcuni particolari e alcune istruzioni).

Per risolvere compiti complessi si suddivide opportunamente il programma in varie parti distinte (Blocchi). Nel linguaggio di programmazione step 5 esistono cinque tipi di blocchi a loro volta suddivisibili in segmenti:

- **blocchi organizzativi (OB)**, che servono per gestire il programma applicativo;
- **blocchi di programma (PB)** e
- **blocchi di passo (SB)**, sono due tipi di blocchi analoghi utili per strutturare il programma e per la programmazione di comandi sequenziali (i blocchi SB sono disponibili solo nell'AG S5-95U);
- **blocchi funzionali (FB)**, sono blocchi di programma speciali. Parti di programma frequentemente ricorrenti o particolarmente complessi (p.es. funzioni di segnalazione o di calcolo) vengono programmate entro blocchi funzionali; essi sono parametrizzabili (solo nell'AG S5-95U) e possiedono un repertorio di istruzioni più esteso (p.es. operazioni di salto all'interno dello stesso segmento);
- **blocchi dati (DB)**, dove l'utente memorizza i dati necessari per l'elaborazione del programma applicativo; sono dati, per esempio, i valori reali, i valori limite oppure i testi.

Con il richiamo di un blocco si può abbandonare un blocco in corso di esecuzione per saltare ad un blocco diverso (figura 8). I blocchi di programma, funzionali e di passo possono essere incatolati fino ad un massimo di 16 livelli.



**Figura 8** - Profondità d'incatolamento

Blocchi organizzativi	(OB..)
Blocchi di programma	(PB..)
Blocchi di passo	(SB.. Solo nel PLC S5-95U)
Blocchi funzionali	(FB..)
Blocchi dati	(DB..)

Il blocco OB1 è il blocco principale; in esso viene scritto l'elenco organizzativo del programma stesso, formato dai richiami alle varie sottoprocedure scritte negli altri blocchi.

<b>BLOCCHI A DISPOSIZIONE</b>		
	Nel PLC S5-100U	Nel PLC S5-95U
OB	OB1, 21, 22, 34. Lunghezza max. 4Kbyte	OB1, 3, 13, 21, 22, 31, 34, 251. Lunghezza max. 8Kbyte
PB	PB0÷PB63 Lunghezza max. 4Kbyte	PB0÷PB255 Lunghezza max. 8Kbyte
SB	---- ----	SB0÷SB255 Lunghezza max. 8Kbyte
FB	FB0÷FB63 Lunghezza max. 4Kbyte	FB0÷FB255 (FB240÷FB255 riservati) Lunghezza max. 8Kbyte
DB	DB2÷DB63 Lunghezza max. 256 parole dati	DB2÷DB255 Lunghezza max. 256 parole dati

Ogni blocco può essere suddiviso in segmenti utili per semplificare la lettura del programma.

Per saltare dal blocco organizzativo agli altri, oppure tra i vari altri blocchi, si possono utilizzare diverse istruzioni di salto, che possono essere:

<b>Operazione</b>	<b>Operandi</b>	<b>Descrizione della funzione</b>
SPA	PB,FB,SB*,OB*	Richiamo incondizionato di un blocco
SPB	PB,FB,SB*,OB*	Richiamo condizionato di un blocco (solo se RLC=1)
A	DB	Apri il blocco dati
BE		Fine blocco
BEA		Salto di ritorno incondizionato da un blocco
BEB		Salto di ritorno condizionato da un blocco (solo se RLC=1)

\* SB e OB possono essere richiamati soltanto nei Plc S5-95U

## Uso del P.C. con Software STEP<sup>C</sup> 5 AG 90 Siemens

Istruzioni:

a) digitare A:\>S5; invio (o C:\S5>S5 se il software è su disco fisso)

b) alla schermata

I: ITALIANO
D: TEDESCO

digitare I

c) alla schermata

SCELTA PACCHETTO				SIMATIC S5 / KOMI			
KOP, FUP, AWL .....V 3.0							
F1	F2	F3	F4	F5	F6	F7	F8
PACCHETTO	PROG.SERV	INFO	VERSIONE	INTERF.	DRIVE	NUOVA SEL	RITORNO

- digitare F1

d) alla schermata

PREDISPOSIZIONE				SIMATIC S5 / KOMI			
RAPPRESENT. : KOP		FILE-PROGRAMMA: ST.S5D					
SIMBOLICO : NO		FILE-SIMBOLICO:					
COMMENTI : SI		FILE-PIEDINAT.:					
PIEDINATURA : NO		FILE-STAMPANTE:					
SOMMA CONTR. : NO		FILE-VIA :					
FUNZIONAMENTO: OFF							
NOME-VIA :							
F1	F2	F3	F4	F5	F6	F7	F8
		SCELTA			CONFERMA	INFO	

- digitare il nome del file accanto al campo FILE-PROGRAMMA. Il cursore deve essere posizionato a lato del campo (massimo 6 caratteri; l'estensione ST.S5D non è modificabile)

- scegliere il tipo di rappresentazione (KOP, FUP, AWL) con il tasto F3 oppure premere la freccia verso destra del tastierino numerico

- scegliere la modalità di funzionamento    ON = con PLC  
    OFF = senza PLC

- confermare con il tasto F6

Prof. Bani S.

**N.B.** Per muovere (verso l'alto o il basso) il cursore si utilizzano le frecce del tastierino numerico; per passare da una colonna all'altra, usare il tasto TAB oppure SHIFT + Frecce.

e) Alla schermata

SCELTA FUNZIONI				SIMATIC S5 / KOMI			
RAPPRESENT. : AWL				FILE-PROGRAMMA: A:PROVA1ST.S5D			
SIMBOLICO : NO				FILE-SIMBOLICO:			
COMMENTI : SI				FILE-PIEDINAT.:			
PIEDINATURA : NO				FILE-STAMPANTE:			
SOMMA CONTR. : NO				FILE-VIA :			
FUNZIONAMENTO: ON							
NOME-VIA :							
F1	F2	F3	F4	F5	F6	F7	F8
INTRODUZ.	EMISSIONE	TEST	FUNZ-AG	INFO-AG	PREDISP.	AUSIL	RITORNO

attivare:

- tasto F1 per scrivere un nuovo programma
- tasto F2 per trasferire un programma da File/PLC/Ram
- tasto F3 per testare il programma
- tasto F8 per tornare alla schermata precedente

f) alla schermata

INTRODUZIONE				SIMATIC S5 / KOMI			
F1			F4				F8
BLOCCO			MASCH.				RITORNO

premere F1

Quando compaiono i campi: INTROD.DISP: \_\_ BLOCCO: \_\_

nella prima voce inserire la sigla del dispositivo dove verrà scritto il programma, nella seconda voce inserire il nome del blocco programma (di solito OB1).

I dispositivi disponibili sono:

AG = PLC

FD = floppy disk

PG = dispositivo di programmazione.

Confermare con il tasto INS (zero) del tastierino numerico.



## ***TRASFERIRE UN PROGRAMMA (salvare su disco)***

Per memorizzare (su supporto magnetico) un programma scritto nella RAM del PLC o per caricare (da dischetto) un programma copiandolo in RAM del PLC è necessario impiegare un comando che permette di trasferire il programma da un dispositivo all'altro.

Dal menù **S C E L T A F U N Z I O N I**

- a) premere F7 = AUSIL
- b) compare il sottomenù **F U N Z I O N I A U S I L I A R I E**
- c) premere F1 = TRASF.
- d) nei relativi campi indicare il dispositivo sorgente, il blocco da trasferire e la destinazione.  
I dispositivi disponibili sono:  
AG = PLC  
FD = floppy disk (o Hard Disk, come indicato nel menù scelta funzioni inizialmente)  
PG = dispositivo di programmazione.

**N.B.** per trasferire tutti i blocchi contemporaneamente inserire B nel campo BLOCCO: .

## ***Elenco dei BLOCCHI, SEGMENTI, HELP in linea***

Per visualizzare l'elenco dei blocchi utilizzati.

Dal menù **S C E L T A F U N Z I O N I**

- a) premere F7 = AUSIL
- b) compare il sottomenù **F U N Z I O N I A U S I L I A R I E**
- c) premere F3 = INDICE
- d) nel primo campo indicare il dispositivo sorgente (AG o FD) e B al campo BLOCCO:, se nel campo STA: si mette 3 l'elenco viene inviato alla stampante.

Tutti i blocchi, per una programmazione più ordinata, sono suddivisibili in **segmenti** (Max 255 per blocco). Per inserire un nuovo segmento posizionarsi nel punto desiderato (in fase di immissione o emissione) e premere il tasto Fine segmento (Shift + Backspace).

A volte, premendo il tasto **Help** (Backspace), è possibile ottenere alcune spiegazioni sui comandi che compaiono sullo schermo.

## ***Scelta della SERIALE per il collegamento al PC***

Alla schermata **S C E L T A P A C C H E T T O**

- a) premere F5 = INTERF.
- b) selezionare interfaccia TTY/AS511
- c) selezionare (tramite tasto F3) STANDARD = COM1 oppure SPECIAL(1) = COM2

## ***COMPRESSIONE della MEMORIA***

Modificando più volte uno stesso programma in AG, può capitare che compaia il messaggio di memoria insufficiente (perché il PLC non sovrascrive le modifiche, ma utilizza sempre nuove allocazioni di memoria). E' quindi necessario eseguire il comando di compressione della memoria:

Dal Menù **S C E L T A F U N Z I O N I**

- a) premere F4 = FUNZ.AG
- b) premere F3 = COMPRESS.

=====

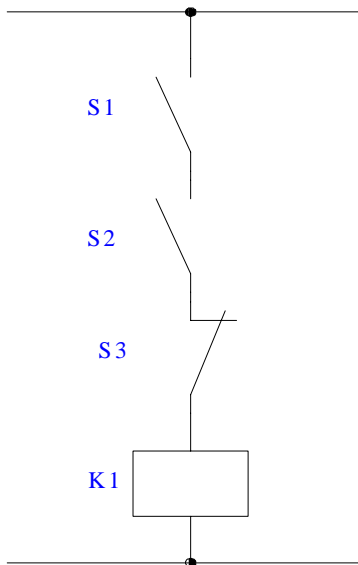
## ISTRUZIONI del PLC S5-100U Siemens

### FUNZIONI COMBINATORIE

Esempi di programmazione di funzioni logiche combinatorie, confrontando le rappresentazioni KOP, FUP, AWL.

1) AND 3 INGRESSI (di cui uno negato)

Schema funzionale:



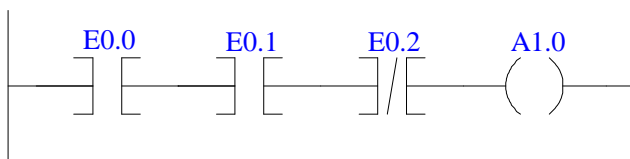
Assegnazioni:

S1 = E0.0  
S2 = E0.1  
S3 = E0.2  
K1 = A1.0

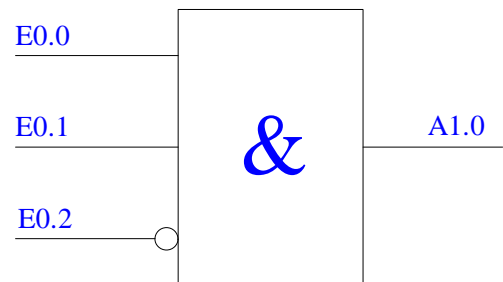
Programma AWL

```
U E 0.0  
U E 0.1  
UNE 0.2  
= A 1.0
```

Programma KOP

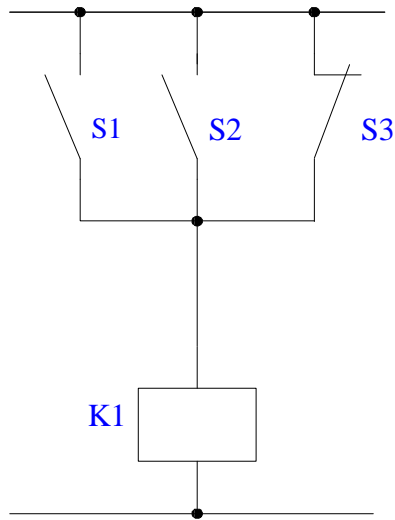


Programma FUP



2) OR 3 INGRESSI (di cui uno negato)

Schema funzionale

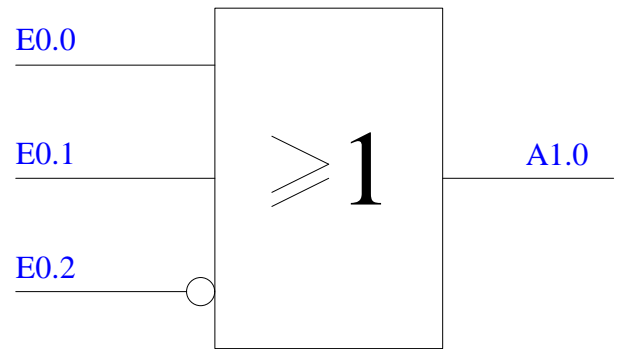
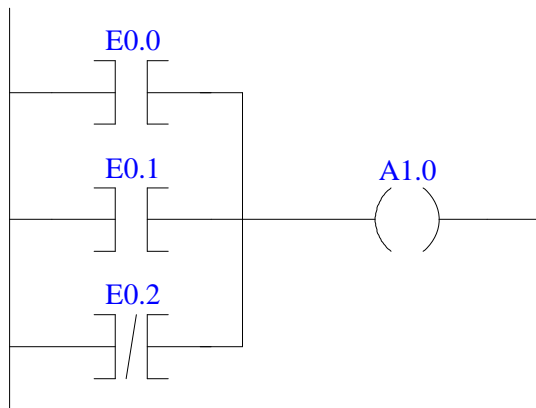


Assegnazioni:

- S1 = E0.0
- S2 = E0.1
- S3 = E0.2
- K1 = A1.0

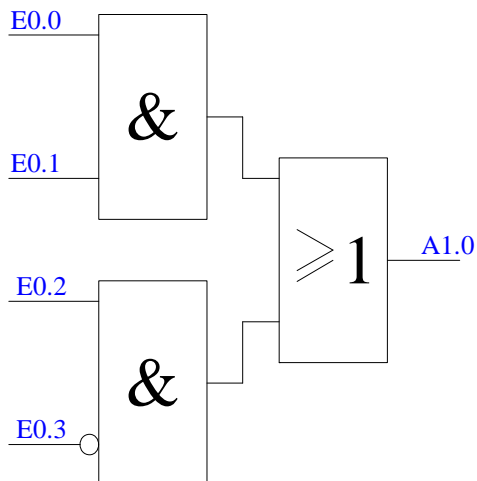
Programma AWL

- O E 0.0
- O E 0.1
- ON E 0.2
- = A 1.0

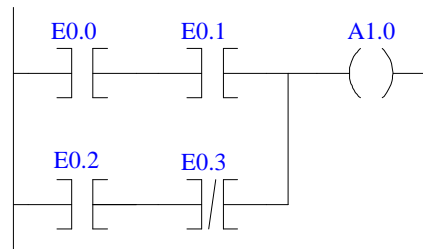


3) SOMMA DI PRODOTTI

Programma FUP



Programma KOP



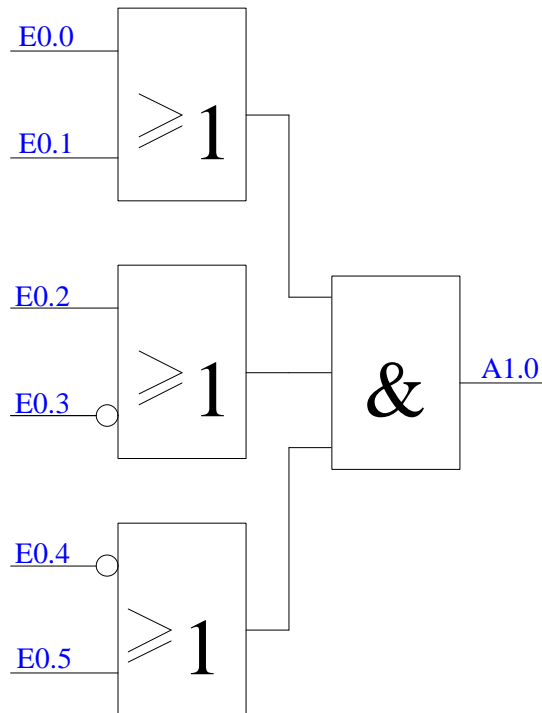
Programma AWL

- U E 0.0
- U E 0.1
- O
- U E 0.2
- UNE 0.3
- = A 1.0

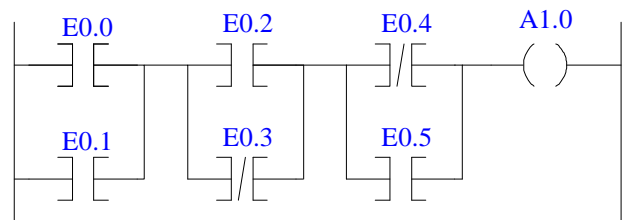


4) PRODOTTO DI SOMME

Programma FUP



Programma KOP



Programma AWL

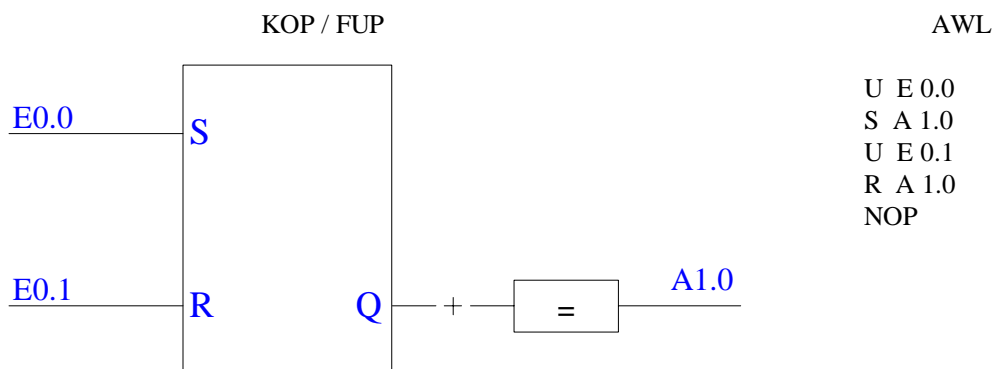
```

U(
O E 0.0
O E 0.1
)
U(
O E 0.2
ON E 0.3
)
U(
ON E 0.4
O E 0.5
)
= A 1.0
    
```

## FUNZIONI SET - RESET

Le istruzioni SET e RESET permettono, rispettivamente di settare (portare a livello alto) o resettare (portare a livello basso) un uscita tramite un impulso in ingresso.

Programmando in KOP o FUP si utilizza un blocco funzionale, programmando in AWL vengono usate le istruzioni (S) e (R).



Il PLC esegue le istruzioni in sequenza quindi, nell'esempio sopra riportato prevale il Reset (nel caso si attivino entrambe gli ingressi), scambiando le istruzioni prevale il Set.

Per poter tradurre il programma scritto in AWL in linguaggio KOP o FUP si deve inserire l'istruzione NOP (No Operation).

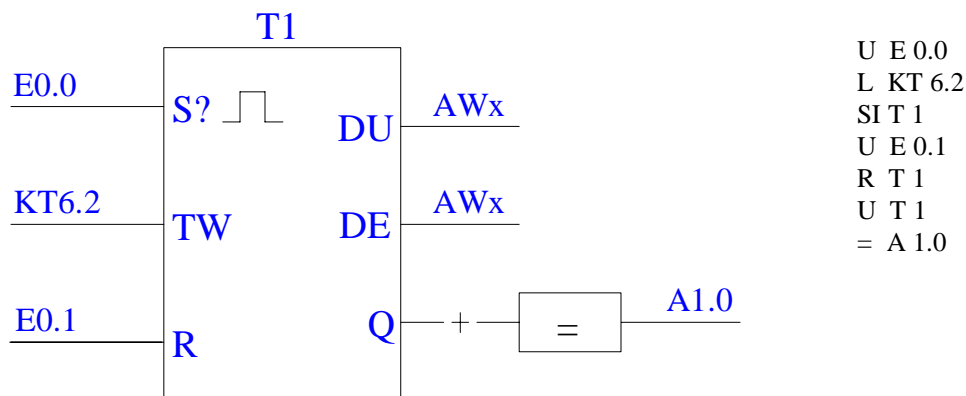
## TEMPORIZZATORI (Timers)

Il temporizzatore è l'equivalente elettrico del monostabile.

Il PLC presenta diversi tipi di temporizzatori in base al legame esistente tra attivazione dell'ingresso e la risposta (legata anche al trascorrere del tempo) dell'uscita. Le funzioni di temporizzazione sono cinque: SI, SV, SE, SS ed SA.

Ogni modello di PLC mette a disposizione un certo numero di temporizzatori, tipicamente 16, 32, 64, 128, ecc. Il PLC S5-100U con CPU 102 contiene 32 temporizzatori (TOT31), con possibilità di espansione attraverso appositi moduli.

Il temporizzatore, schematizzato come in figura, presenta tre ingressi e tre uscite.



**S** = ingresso di START; se attivato (**l'attivazione avviene sul fronte** di salita o discesa del segnale di start in base al tipo di temporizzazione) viene caricato il valore dell'intervallo di tempo impostato all'ingresso TW. L'intervallo di tempo decresce fino ad azzerarsi.

**TW** = ingresso in cui viene caricata la COSTANTE DI TEMPO che determina la durata della funzione di temporizzazione.

**R** = ingresso di RESET; se attivato forza l'uscita a zero.

**DU** = uscita digitale in codice binario; fornisce il valore istantaneo del tempo.

**DE** = uscita digitale in codice BCD; fornisce il valore istantaneo del tempo.

**Q** = USCITA binaria; può assumere valore 0 o 1.

### DEFINIZIONE DELLA COSTANTE DI TEMPO KT

Nella programmazione KOP e FUP la costante di tempo che determina il funzionamento del temporizzatore viene digitata vicino all'ingresso TW. In AWL si utilizza l'istruzione LKT xxx.y; con xxx si intende un numero compreso tra 0 e 999 mentre y rappresenta un moltiplicatore secondo la tabella:

Ad esempio KT005.3 indica una costante di tempo pari a  $5 \times 10 = 50s$ . Il massimo valore impostabile è KT999.3 cioè  $9990s = 2h 46' 30''$ .

In AWL l'istruzione necessaria per caricare un tempo di 45s è LKT45.2 oppure LKT450.1. La differenza si traduce nella modalità di scansione del tempo nella fase di temporizzazione. Infatti il tempo 45.2 verrà decrementato ogni secondo mentre il tempo 450.1 ogni decimo di secondo con un miglioramento della precisione.

Valore di y	tempo in secondi
0	x 0,01 s
1	x 0,1 s
2	x 1 s
3	x 10 s

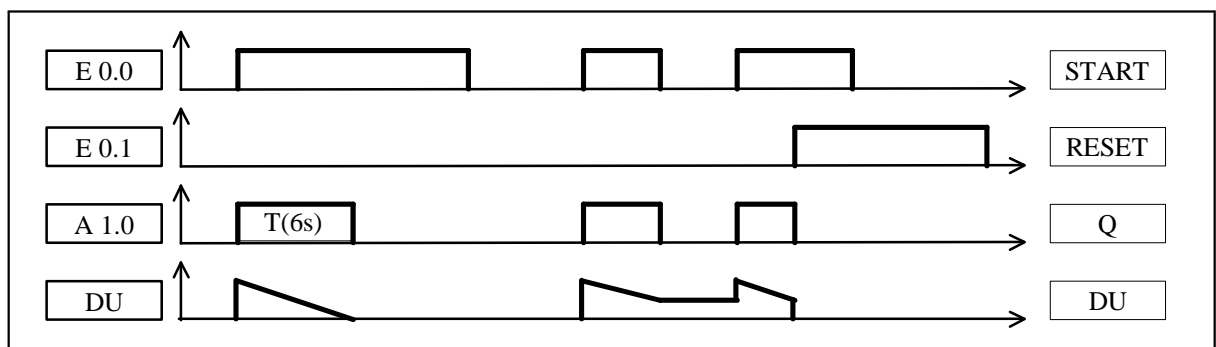
## FUNZIONI DI TEMPORIZZAZIONE

**Funzione SI** = impulso (condizionato dallo stato dell'ingresso).

Quando l'ingresso di start è attivo viene caricata la costante di tempo (KT) prefissata che decresce fino ad annullarsi.

L'uscita Q resta alta per l'intervallo di tempo KT. Se durante la temporizzazione viene disattivato l'ingresso di start, l'uscita si porta istantaneamente a zero.

Listato AWL	Commento
U E 0.0	ingresso esterno collegato allo start
L KT 6.2	costante di tempo di 6s
SI T 1	funzione impulso
U E 0.1	ingresso esterno che attiva il reset
R T 1	reset di T1
U T 1	interrogazione sullo stato di Q
= A 1.0	uscita esterna collegata a Q



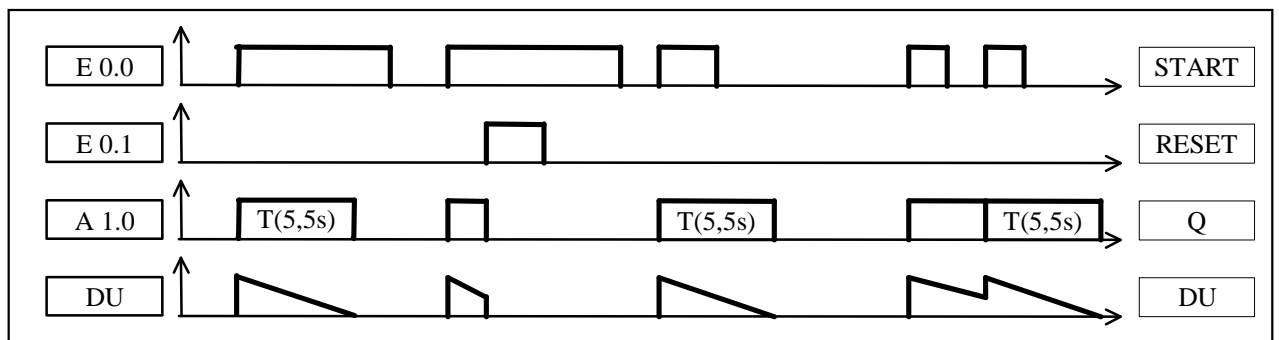
**Funzione SV** = impulso prolungato (non condizionato dallo stato dell'ingresso di Start).

Quando l'ingresso di start è attivo (fronte di salita) viene caricata la costante di tempo (KT) prefissata che decresce fino ad annullarsi e l'uscita Q resta alta per l'intervallo di tempo KT.

Se durante la temporizzazione si ha un fronte di discesa sull'ingresso di start, l'uscita non cambia.

Nel caso in cui l'ingresso subisca un nuovo fronte di salita la costante di tempo viene ricaricata.

Listato AWL	Commento
U E 0.2	ingresso esterno collegato allo start
L KT 55.1	costante di tempo di 5,5s
SV T 2	funzione impulso prolungato
U E 0.3	ingresso esterno che attiva il reset
R T 2	reset di T2
U T 2	interrogazione sullo stato di Q
= A 1.0	uscita esterna collegata a Q



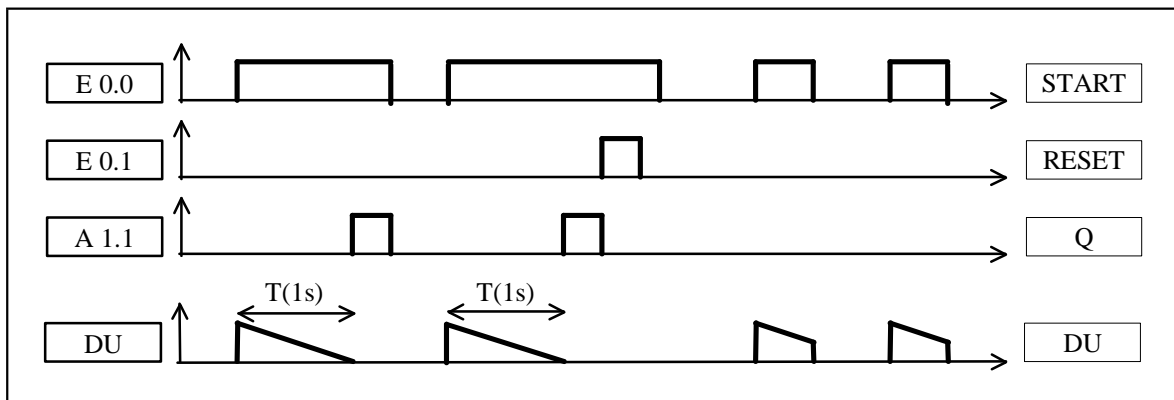
**Funzione SE** = ritardo all'inserzione (condizionato dallo stato dell'ingresso di start).

Quando l'ingresso di start è attivo viene caricata la costante di tempo (KT) prefissata che decresce fino ad annullarsi.

L'uscita Q va alta al termine l'intervallo di tempo KT.

L'uscita Q va bassa se si disattiva l'ingresso di start o se si attiva l'ingresso di reset. Se l'ingresso di start viene disattivato prima che scada il tempo impostato si annulla la temporizzazione.

Listato AWL	Commento
U E 0.0	ingresso esterno collegato allo start
L KT 1.2	costante di tempo di 1s
SE T 3	funzione ritardo all'eccitazione
U E 0.1	ingresso esterno che attiva il reset
R T 3	reset di T3
U T 3	interrogazione sullo stato di Q
= A 1.0	uscita esterna collegata a Q



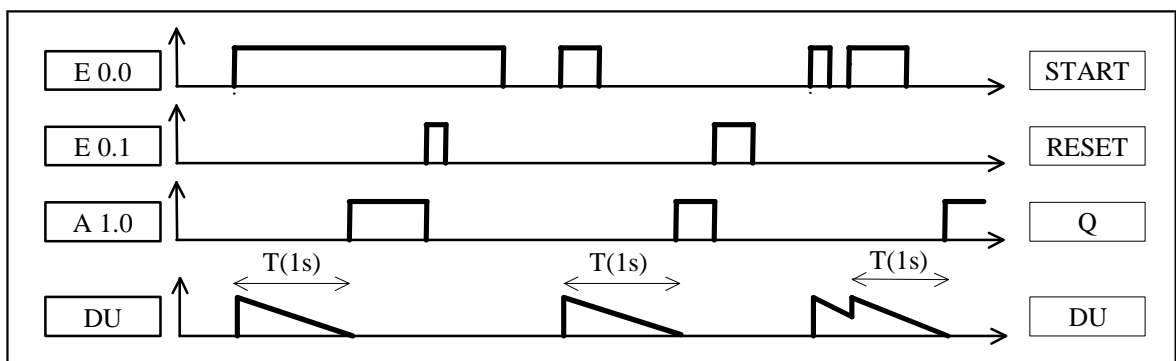
**Funzione SS** = ritardo all'inserzione con memoria (non condizionato dallo stato dell'ingresso di start).

Quando l'ingresso di start è attivo (fronte di salita) viene caricata la costante di tempo (KT) prefissata che decresce fino ad annullarsi.

L'uscita Q va alta al termine l'intervallo di tempo KT, anche se nel frattempo si disattiva l'ingresso di start.

L'uscita Q va bassa nel momento in cui si attiva l'ingresso di reset. Un ulteriore fronte di salita sull'ingresso di start (durante la temporizzazione) fa ripartire il tempo impostato.

Listato AWL	Commento
U E 0.0	ingresso esterno collegato allo start
L KT 10.1	costante di tempo di 1s
SS T 4	funzione ritardo all'eccitazione con memoria
U E 0.1	ingresso esterno che attiva il reset
R T 4	reset di T4
U T 4	interrogazione sullo stato di Q
= A 1.0	uscita esterna collegata a Q



**Funzione SA** = ritardo alla disinserzione (non condizionato dallo stato dell'ingresso di Start).

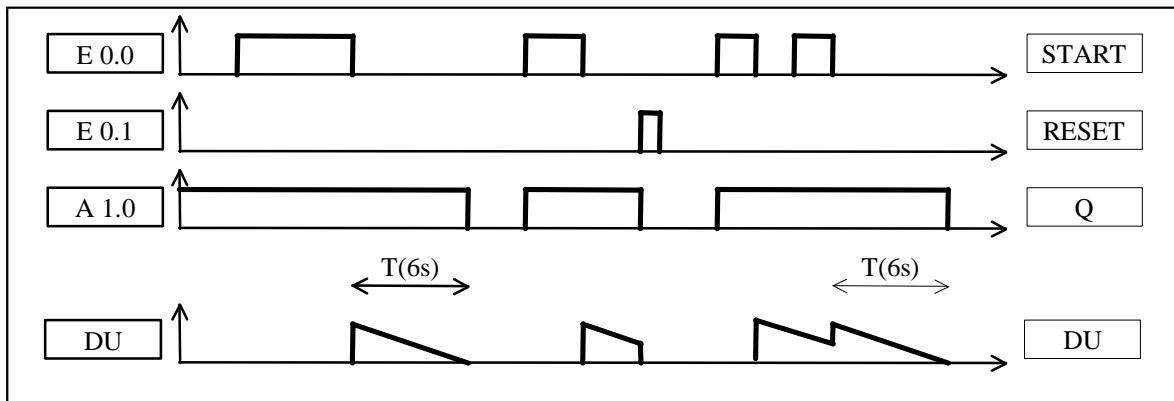
L'utilizzo di questa funzione presuppone che l'uscita Q sia a livello alto (se l'uscita fosse bassa verrebbe comunque portata alta all'attivazione dell'ingresso di start).

Quando sull'ingresso di start si ha un fronte di discesa viene caricata la costante di tempo (KT) prefissata che decresce fino ad annullarsi.

L'uscita Q va bassa al termine l'intervallo di tempo KT anche se nel frattempo si disattiva l'ingresso di start.

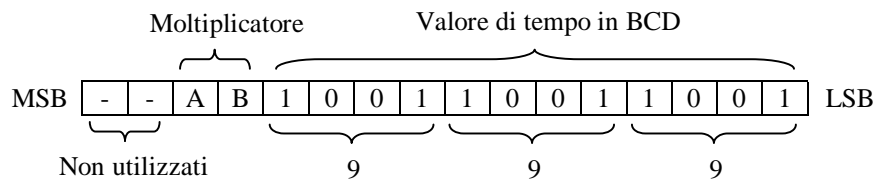
Un ulteriore fronte di salita sull'ingresso di start (durante la temporizzazione) fa ripartire il tempo impostato.

Listato AWL	Commento
U E 0.0	ingresso esterno collegato allo start
L KT 60.1	costante di tempo di 6s
SA T 5	funzione ritardo alla diseccitazione
U E 0.1	ingresso esterno che attiva il reset
R T 5	reset di T5
U T 5	interrogazione sullo stato di Q
= A 1.0	uscita esterna collegata a Q



## LETTURA DELLA COSTANTE DI TEMPO DA UNA WORD

La costante di tempo dei temporizzatori può anche essere caricata da una Word (per esempio da una Merker Word) rispettando il seguente formato:



I valori del moltiplicatore possono essere:

A	B	moltiplicatore
0	0	x 0,01 s
0	1	x 0,1 s
1	0	x 1 s
1	1	x 10 s

Esempio:

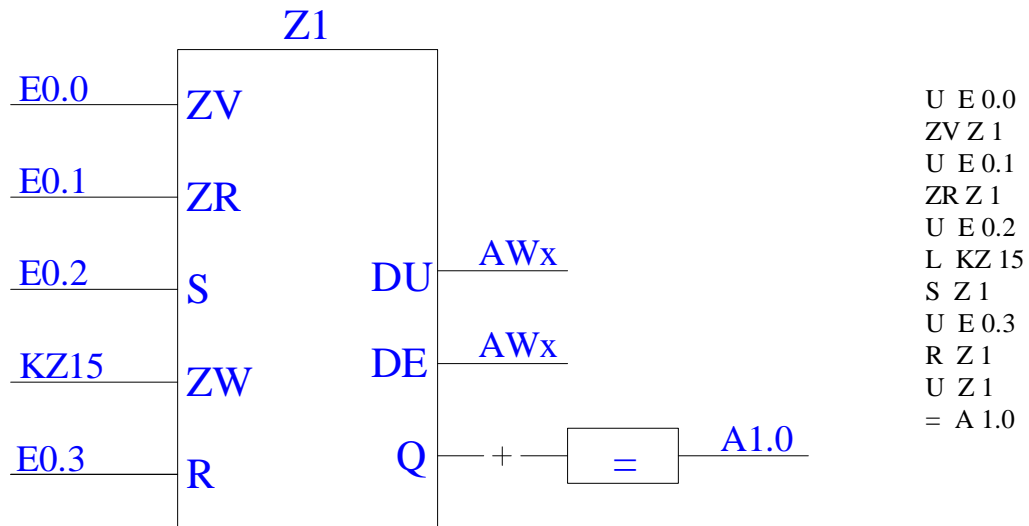
UE 0.0  
L MW 10  
SV T1

## CONTATORI (Counters)

Nei controlli industriali occorrono spesso contatori per effettuare dei conteggi (per esempio il conteggio del numero di bottiglie, di macchine, di pezzi lavorati, ecc.).

Il PLC S5-100U Con CPU 102 contiene 32 contatori (Z0BZ31). Essi sono caratterizzati dal numero massimo di eventi conteggiabili e dalla possibilità di contare avanti e indietro.

Il contatore è schematizzato come in figura:



**ZV** = conteggio avanti: se all'ingresso si applica un fronte di salita il contatore si incrementa di una unità.

**ZR** = conteggio indietro: se all'ingresso si applica un fronte di salita il contatore si decrementa di una unità.

**ZW** = (Preset): ingresso dove viene caricato un numero utile per settare il contatore. In AWL il caricamento della costante ZW avviene con l'istruzione LKZ xxx dove xxx è un numero compreso tra 0 e 999. Nella programmazione KOP/FUP il numero va riportato in prossimità del relativo ingresso.

**S** = ingresso di SET; se attivato il contatore si predispone a contare a partire dal numero impostato attraverso l'ingresso ZW.

**R** = ingresso di RESET; se attivato forza l'uscita a zero.

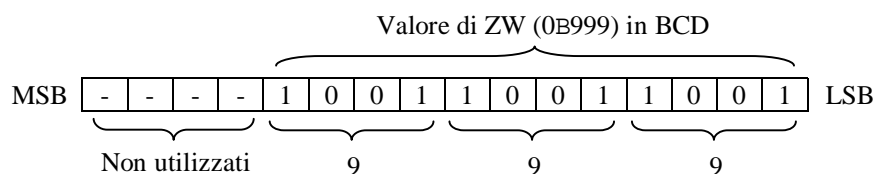
**DU** = uscita digitale in codice binario; fornisce il valore istantaneo del conteggio.

**DE** = uscita digitale in codice BCD; fornisce il valore istantaneo del conteggio.

**Q** = USCITA binaria; assume valore zero soltanto quando il contenuto del contatore è zero, altrimenti è alta. L'uscita può essere interrogata con le funzioni logiche UZ (And), OZ (Or), ONZ (Or negato), UNZ (And negato).

### LETTURA DELLA COSTANTE DI CONTEGGIO DA UNA WORD

Il valore della costante di conteggio (ZW) può anche essere caricata da una Word (per esempio da una Merker Word) rispettando il seguente formato:



Esempio:

UE 0.0  
LMW 10  
S Z1

## MERKER

Il Merker è un elemento di memoria che serve a memorizzare in modo provvisorio il risultato di una operazione. Il dato memorizzato può essere ad un bit (M), a 8 bit (byte, MB), a 16 bit (word, MW).

Sigla Operando	Descrizione della funzione	Campo
M	Merker 1 bit	0.0 B 127.7
MB	Merker 8 bit, byte	0 BBBBB
MW	Merker 16 bit, word	0 BBBBB

Esempio:

U E 0.0 se è attivato l'ingresso E0.0

S M 0.0 viene settato il merker 0.0 che memorizza l'attivazione di E0.0 anche quando esso va basso

U E 0.1 se è attivato l'ingresso E0.1

R M 0.0 viene resettato il merker, lo stato del merker indica quale dei due ingressi è stato attivato per ultimo.

L'utilizzo delle celle di memoria dei Merker avviene secondo la tabella sottoriportata. Occorre prestare molta attenzione nell'impiego delle Word: ad esempio utilizzare la MW0 e la MW1 comporta la sovrapposizione di un byte.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MW0 { Byte - 0	M0.7	M0.6	M0.5	M0.4	M0.3	M0.2	M0.1	M0.0
MW1 { Byte - 1	M1.7	M1.6	M1.5	M1.4	M1.3	M1.2	M1.1	M1.0
MW2 {	M2.7	M2.6	M2.5	M2.4	M2.3	M2.2	M2.1	M2.0
	M3.7	M3.6	M3.5	M3.4	M3.3	M3.2	M3.1	M3.0
Byte - 4	M4.7	M4.6	M4.5	M4.4	M4.3	M4.2	M4.1	M4.0
Byte - X	Mx.7							

Esempio: peso e identificazione dei bit utilizzati dalla MW 0.

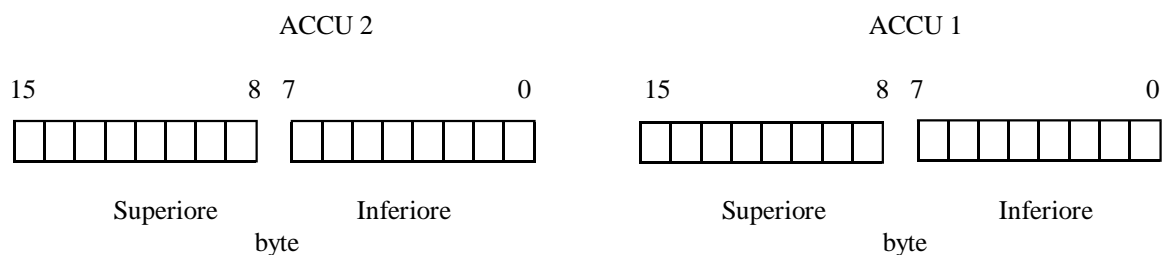
MSB	M0.7	M0.6	M0.5	M0.4	M0.3	M0.2	M0.1	M0.0	M1.7	M1.6	M1.5	M1.4	M1.3	M1.2	M1.1	M1.0	LSB
-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-----

## REGISTRI ACCUMULATORI

Il PLC utilizza due registri accumulatori, ACCU 1 e ACCU 2, per poter eseguire le operazioni sui byte (8 bit) e sulle parole (Word, 16 bit).

Le operazioni consentite sono di tipo logico digitale (AND, OR, EXOR), aritmetico e di confronto.

Ogni registro ha una capienza di 16 bit.



## CARICAMENTO e TRASFERIMENTO

Le istruzioni di caricamento hanno sempre per destinazione ACCU 1. Il contenuto di ACCU1 viene trasferito come parola (16 bit) in ACCU 2.

**Caricamento:** gli accumulatori possono essere caricati con delle costanti. Queste costanti possono essere introdotte in diversi formati:

Operatore	Formato	Tipo della costante	Campo
L	KB	Num. in virgola fissa di 8 bit senza segno	0 B 255
L	KF	Num. in virgola fissa di 16 bit con segno	-32768 BBBBBBB
L	KM	Stringa di 16 bit	sequenza 16 bit qualsiasi
L	KH	Stringa esadecimale (4 cifre)	0000 <sub>H</sub> BFFFF <sub>H</sub>
L	KY	Due numeri in virgola fissa di 8 bit senza segno	0 B255, 0 B 255
L	KC	due caratteri alfanumerici	Codice ASCII

Il caricamento di una costante avviene sempre, indipendentemente dallo stato del risultato logico combinatorio (RLC). Un numero in virgola fissa viene caricato come numero binario. I valori negativi dei numeri in virgola fissa (numeri con segno meno) vengono caricati nella forma del complemento a 2.

Esempio:

L KF xxx = il numero xxx viene caricato in ACCU 1, il contenuto di ACCU 1 va in ACCU 2 mentre il contenuto di ACCU 2 va perso.

Con l'operazione "caricamento" (L) si possono caricare in ACCU 1 le informazioni contenute nei campi degli operandi E,A,M,D,T, e Z. I valori vengono interpretati come numeri di 16 bit in virgola fissa. Il contenuto dell'accumulatore visualizzato sul monitor, tramite la funzione test (F3), è in esadecimale.

Esempi:

L Tx = il contenuto del temporizzatore Tx viene caricato in ACCU 1 come numero binario a 16 bit.

L Zx = il contenuto del contatore Zx viene caricato in ACCU 1 come numero binario a 16 bit.

Con l'operazione "caricamento codificato" (LC) si possono caricare in ACCU 1, codificati in BCD, i valori di tempo o di conteggio esistenti in forma binaria. Il valore attuale pertanto viene caricato come numero decimale codificato binario. Il contenuto dell'accumulatore visualizzato sul monitor, tramite la funzione test, è in decimale.

Esempi:

LC Tx = il contenuto in codice BCD del temporizzatore Tx viene caricato in ACCU 1 e il contenuto di ACCU 1 va in ACCU 2.

LC Zx = il contenuto in codice BCD del contatore Zx viene caricato in ACCU 1.

Con l'operazione "trasferimento" (T) si possono copiare informazioni dall'ACCU 1 nei campi degli operandi E,A,M, e D.

## COMBINAZIONI LOGICHE DIGITALI

I valori di due operandi possono essere caricati negli accumulatori. Gli operandi vengono interpretati ed elaborati come numeri binari. Il risultato viene caricato in ACCU 1 mentre ACCU 2 rimane invariato.

Operazione	Descrizione della funzione
UW	Combinazione AND bit a bit di ACCU 1 e ACCU 2
OW	Combinazione OR bit a bit di ACCU 1 e ACCU 2
XOW	Combinazione OR esclusivo bit a bit di ACCU 1 e ACCU 2



## OPERAZIONI ARITMETICHE

Mediante operazioni aritmetiche i dati contenuti nei campi degli operandi E,A,M, e D vengono combinati tra loro. A questo scopo gli operandi devono venire caricati negli accumulatori. Il risultato si troverà in ACCU 1 mentre ACCU 2 rimane invariato. I dati vengono interpretati come numeri fissa di 16 bit in virgola fissa con segno. In caso di overflow si ha il settaggio dell'indicatore OV.

Operazione	Descrizione della funzione
+F	Addizione dei contenuti di ACCU 1 ed ACCU 2 (ACCU 1 + ACCU 2)
-F	Sottrazione del contenuto di ACCU 1 da quello di ACCU 2 (ACCU 2 - ACCU 1)

Esempio:

Si vuole sapere quanto manca al contatore Z1 per raggiungere il valore indicato (45)

L KF 45            carica il numero 45 in ACCU 1  
 L Z 1              carica in ACCU 1 il contenuto di Z1, il contenuto di ACCU 1 va in ACCU 2  
 -F                 sottrazione ACCU 2 - ACCU 1, il risultato va in ACCU 1.

## OPERAZIONI DI CONFRONTO

Sono operazioni che permettono di confrontare i contenuti di due operandi. I valori devono prima essere caricati in ACCU 1 e ACCU 2. Il risultato agisce su RLC (risultato logico combinatorio). I dati vengono interpretati come numeri di 16 bit in virgola fissa con segno.

Operazione	Descrizione della funzione
!= F	Il contenuto di ACCU 2 è uguale a quello di ACCU 1?
> < F	Il contenuto di ACCU 2 è diverso da quello di ACCU 1?
> F	Il contenuto di ACCU 2 è maggiore a quello di ACCU 1?
> = F	Il contenuto di ACCU 2 è maggiore/uguale a quello di ACCU 1?
< F	Il contenuto di ACCU 2 è minore a quello di ACCU 1?
< = F	Il contenuto di ACCU 2 è minore/uguale a quello di ACCU 1?

I contenuti degli accumulatori non vengono modificati.

Se la funzione di confronto è soddisfatta, il risultato logico (RLC) è "1".

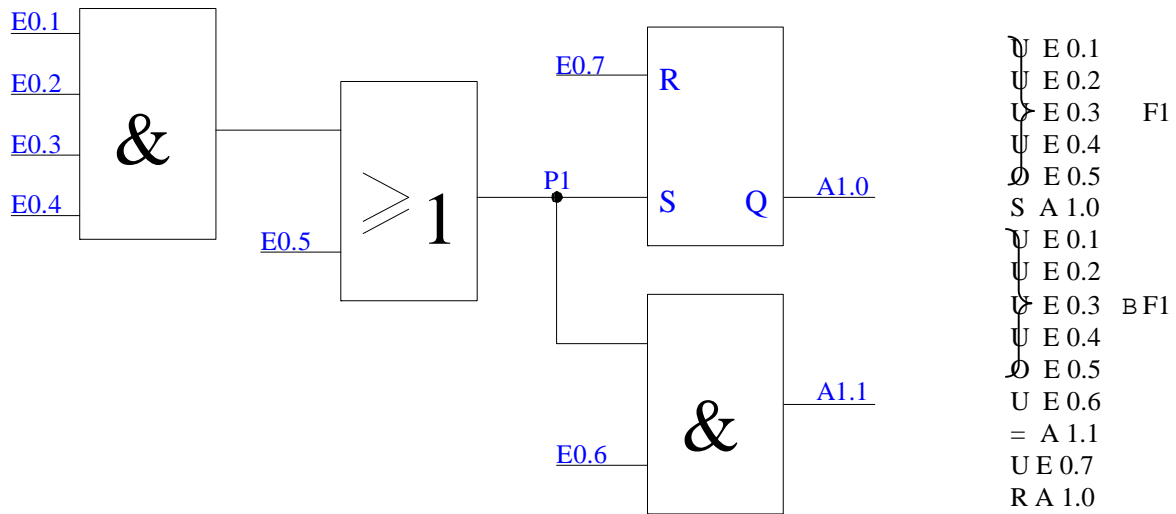
Esempio:

Si vuole confrontare il contenuto del contatore Z1 col numero 24 e nel caso di uguaglianza attivare l'uscita A1.0

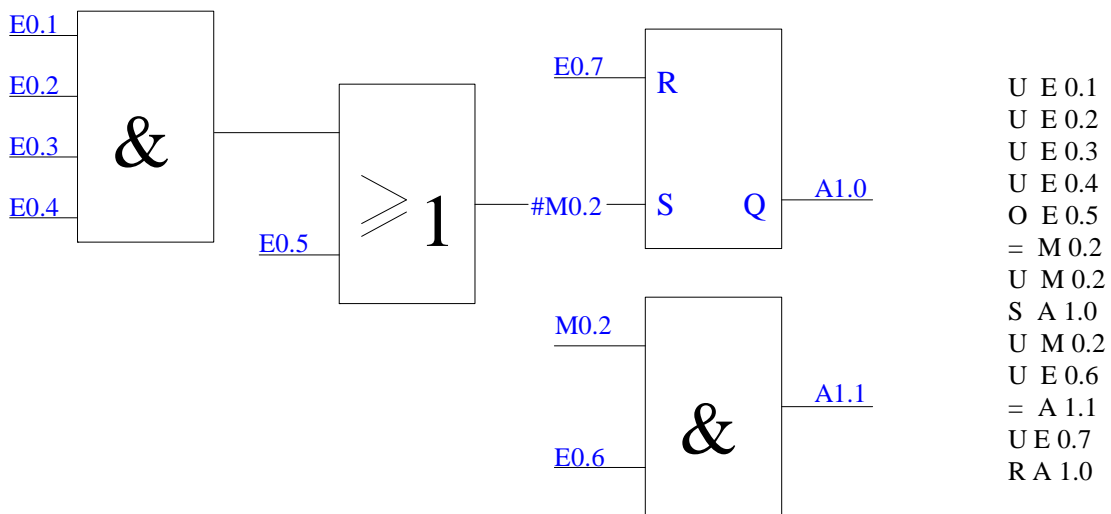
L KF 24            carica in ACCU 1 il numero 24  
 L Z1                carica in ACCU 1 il contenuto di Z1, ACCU1 va in ACCU2  
 != F                Confronta se ACCU 1 = ACCU 2  
 = A 1.0            Se il contatore contiene 24 l'uscita A1.0 va alta.

## FUNZIONE CONNETTORE

Un'applicazione delle memorie (merker) è la "funzione connettore" che consente di evitare ripetizioni di parti uguali di programma.



Per evitare la ripetizione della funzione F1, dell'esempio, si inserisce nel punto P1 una connessione come indicato di seguito.



La connessione #M0.2 realizza la memorizzazione del risultato della funzione F1 che può essere utilizzata in qualsiasi punto del programma.

## **ESERCIZI**

- 1) Utilizzando le funzioni logiche combinatorie:
  - realizzare porte NAND (funzione OR con ingressi negati), NOR, EXOR e Flip-Flop;
  - segnalazione luminosa;
  - comando dei ventilatori di un garage sotterraneo;
  - visualizzatore a sette segmenti.
  
- 2) Utilizzando le istruzioni Set - Reset e merker:
  - Marcia/Arresto di un motore tramite due pulsanti (Start - Stop);
  - rilevare il fronte di un segnale (fronti di salita e discesa);
  - simulazione del funzionamento di un relè passo passo;
  - comando ad impulsi di un impianto di riscaldamento.
  
- 3) Utilizzando le funzioni di temporizzazione:
  - generatore di impulsi con due temporizzatori;
  - controllo di una frequenza esterna;
  - insegna luminosa;
  - semaforo pedonale.
  
- 4) Utilizzando le funzioni di conteggio, caricamento e confronto:
  - lavaggio chimico;
  - orologio software;
  - gestione di un magazzino intermedio.

N.B. Gli Esercizi con le soluzioni sono riportate nel fascicolo “Esercizi in linguaggio AWL” (File:PLC-ESER.DOC).

## **BIBLIOGRAFIA:**

Daniele Fuselli - Tecnologia Disegno e Progettazione vol. 2 - Ed. Zanichelli

Manuale di sistema del controllore programmabile S5-90U/S5-95U - Siemens

SIMATIC S5, Pacchetto STEP<sup>B</sup> 5 AG 90 - Siemens

Esempi applicativi con SIMATIC S5 - Siemens

Corso Base S15 - Siemens

## WSO O EVMS \$hikpčevksq irxm

<b>INTRODUZIONE AI CONTROLLORI LOGICI PROGRAMMABILI (PLC)</b> .....	<b>1</b>
INTRODUZIONE.....	1
LOGICA CABLATA E LOGICA PROGRAMMATA .....	1
COSTRUTTORI, CRITERI DI SCELTA E IMPIEGHI.....	2
CONFRONTO TRA PLC E MICROPROCESSORI .....	3
ARCHITETTURA DI UN PLC .....	3
UNITÀ CENTRALE DI PROCESSO (CPU).....	4
MEMORIA .....	5
<i>Memoria per il programma utente</i> .....	5
<i>Memoria per le funzioni interne</i> .....	6
<i>Memoria per lo stato degli ingressi e delle uscite</i> .....	6
<i>Memoria per il software di sistema</i> .....	6
SEZIONI DI INGRESSO E USCITA: INPUT/OUTPUT (I/O).....	6
Unità di INPUT.....	9
Unità di OUTPUT .....	9
CARATTERISTICHE MECCANICHE.....	10
<b>SOFTWARE DEI CONTROLLORI LOGICI PROGRAMMABILI</b> .....	<b>10</b>
INTRODUZIONE.....	10
SOFTWARE DEI PLC - LINGUAGGI .....	12
<b>LISTA ISTRUZIONI (AWL)</b> .....	<b>13</b>
FASI DI ESECUZIONE DI UN PROGRAMMA .....	14
LINGUAGGIO DI PROGRAMMAZIONE AWL STEP 5 - SIEMENS .....	14
<i>Repertorio degli operandi</i> .....	15
ORGANIZZAZIONE DEL PROGRAMMA - ISTRUZIONI DI SALTO .....	15
<i>Blocchi disponibili nei controllori S5-100U e S5-95U</i> .....	16
<b>USO DEL P.C. CON SOFTWARE STEP<sup>c</sup> 5 AG 90 SIEMENS</b> .....	<b>18</b>
CORREGGERE UN PROGRAMMA .....	20
RICHIAMARE E STAMPARE UN PROGRAMMA .....	20
TRASFERIRE UN PROGRAMMA (SALVARE SU DISCO) .....	21
ELENCO DEI BLOCCHI, SEGMENTI, HELP IN LINEA.....	21
SCELTA DELLA SERIALE PER IL COLLEGAMENTO AL PC.....	21
COMPRESSIONE DELLA MEMORIA .....	21
<b>ISTRUZIONI DEL PLC S5-100U SIEMENS</b> .....	<b>22</b>
FUNZIONI COMBINATORIE.....	22
FUNZIONI SET - RESET .....	24
TEMPORIZZATORI (TIMERS).....	25
DEFINIZIONE DELLA COSTANTE DI TEMPO KT.....	25
FUNZIONI DI TEMPORIZZAZIONE .....	26
LETTURA DELLA COSTANTE DI TEMPO DA UNA WORD .....	28
CONTATORI (COUNTERS).....	29
LETTURA DELLA COSTANTE DI CONTEGGIO DA UNA WORD .....	29
MERKER.....	30
REGISTRI ACCUMULATORI .....	30
CARICAMENTO e TRASFERIMENTO.....	31
COMBINAZIONI LOGICHE DIGITALI .....	31
OPERAZIONI ARITMETICHE.....	32
OPERAZIONI DI CONFRONTO.....	32
FUNZIONE CONNETTORE .....	33
ESERCIZI.....	34
BIBLIOGRAFIA:.....	34