

Automazione e territorio



ISTITUTO D'ISTRUZIONE SUPERIORE "M. Bartolo"- PACHINO

LICEO Classico -LICEO Scientifico

LICEO delle Scienze Applicate - Liceo delle Scienze Umane

I.T.I.S. (Elettronica e Elettrotecnica - Meccanica, Meccatronica e Energia - Trasporti e Logistica)

I.T.I.S. SERALE (Elettronica e Elettrotecnica - Meccanica, Meccatronica e Energia)

Sede centrale: Viale A. Moro sn - 96018 PACHINO (SR)

Uffici: Tel. 0931-593596 Fax 0931-597915

Sede staccata: Via Fiume -96018 PACHINO (SR) - Tel. e Fax 0931846359

Codice Fiscale: 83002910897 - **Email:** sriso1400g@istruzione.it – sriso1400g@pec.istruzione.it

www.istitutobartolo.it - www.primopachino.it - www.sriso1400g.scuolanet.info

Lezione 2– La programmazione dei PLC

Linguaggi di programmazione

- I tre tipi di linguaggi più usati per programmare il PLC sono:

AWL

- AWL o Lista di Istruzioni: E' un linguaggio scritto simile all'Assembler (linguaggio di programmazione dei microprocessori)

KOP o LADDER

- KOP o Ladder o LD o a Contatti: E' un linguaggio grafico di facile utilizzo pensato per quelli che provengono dal mondo elettrotecnico.

FUP

- FUP o FB o Blocchi Funzionali: E' un linguaggio grafico pensato per quelli che provengono dal mondo elettronico.

LINGUAGGI DI PROGRAMMAZIONE

LINGUAGGI DI PROGRAMMAZIONE

```
graph LR; A[LINGUAGGI DI PROGRAMMAZIONE] --> B[GRAFICI (Simbolici)]; A --> C[LETTERALI (Alfanumerici)]; B --> D[A CONTATTI (Ladder, KOP)]; B --> E[A BLOCCHI FUNZIONALI (Porte logiche, FUP)]; B --> F[A BLOCCHI SEQUENZIALI (Grafcet)]; C --> G[A LISTA ISTRUZIONI (Boolean, AWL)]; C --> H[EVOLUTI (Basic, ...)]
```

The diagram is a hierarchical flowchart illustrating the classification of programming languages. It starts with a central green box on the left labeled 'LINGUAGGI DI PROGRAMMAZIONE'. Two arrows branch out from this box to 'GRAFICI (Simbolici)' (orange box) and 'LETTERALI (Alfanumerici)' (blue box). From 'GRAFICI', three arrows branch out to 'A CONTATTI (Ladder, KOP)', 'A BLOCCHI FUNZIONALI (Porte logiche, FUP)', and 'A BLOCCHI SEQUENZIALI (Grafcet)', all in orange boxes. From 'LETTERALI', two arrows branch out to 'A LISTA ISTRUZIONI (Boolean, AWL)' and 'EVOLUTI (Basic, ...)', both in blue boxes. The entire diagram is set against a background of a spiral-bound notebook.

GRAFICI
(Simbolici)

A CONTATTI
(Ladder, KOP)

A BLOCCHI FUNZIONALI
(Porte logiche, FUP)

A BLOCCHI SEQUENZIALI
(Grafcet)

LETTERALI
(Alfanumerici)

A LISTA ISTRUZIONI
(Boolean, AWL)

EVOLUTI
(Basic, ...)

AWL, KOP, FUP

- Le sigle AWL, KOP e FUP vengono dal tedesco:
 - AWL AnWeisungsListe: Lista di Istruzioni
 - KOP KOntaktPlan: Squema a contatti
 - FUP FUnktionsPlan: Squema a Funzioni (Blocchi funzionali)

Lista di Istruzioni

AWL

- Scrivere un programma in lista di istruzioni permette la massima flessibilità per ottenere il massimo dal PLC, ma non è il più intuitivo. Esempio di programma in lista di istruzioni:

Indirizzo Istruzione Operando

00000

LD E0.0

(Carica lo stato dell'ingresso E0.0)

AND E0.1

(Fa l'and logica con l'ingresso E0.1)

00001

OUT A0.0

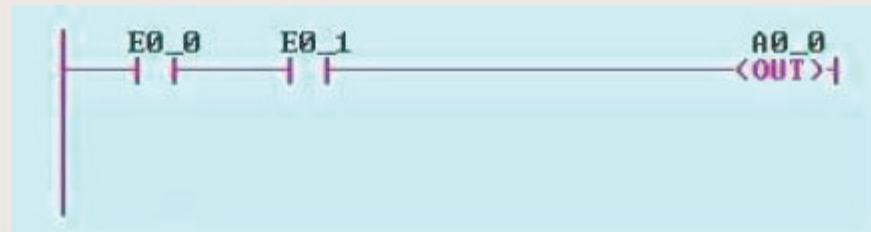
(Il risultato attiva l'uscita A0.0)

00002

A Contatti

LADDER

- Scrivere un programma in linguaggio a contatti è molto semplice ed intuitivo. E' stato pensato per quelli che provengono dal mondo dell'elettrotecnica. Lo stesso esempio precedente diventerà:

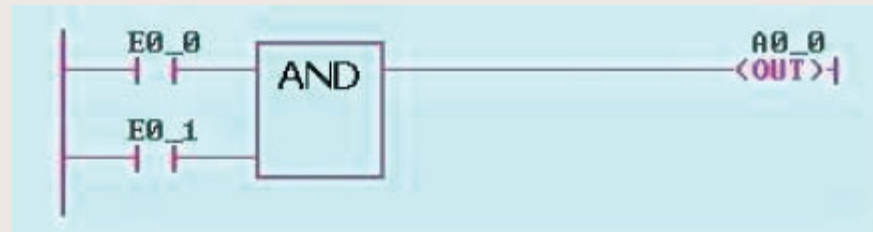


- ❖ Si articola in linee verticali dette "rung"

A Blocchi Funzionali

FUP

- Il linguaggio a Blocchi Funzionali è anche semplice ed intuitivo ma pensato per quelli che provengono dal mondo dell'elettronica. Lo stesso esempio precedente sarà:



Altri linguaggi

- Un altro tipo di linguaggio più evoluto e pensato per quelli che provengono dal mondo della programmazione pura è il Testo Strutturato. In pratica è molto simile al linguaggio C o altri linguaggi di alto livello.
- Anche se fondamentalmente questi linguaggi sono uguali per tutte le marche di PLC, nella pratica ogni marca usa uno proprio. Nel tentativo di stabilire uno standard unico si sono stabilite delle normative al riguardo per facilitare la comprensione dei programmi tra programmatori di diverse marche di PLC. Così sono nati i linguaggi Grafcet, SFC, ecc.

Circuito di autoritenuta

Sfrutta il funzionamento di un **relè** per far sì che un utilizzatore (LED) sia acceso dall'operatore tramite un pulsante di start e arrestato tramite un pulsante di stop.

Premendo il pulsante di start la bobina del relè si eccita e chiude il contatto K1, facendo in modo che la bobina si autoalimenti fino alla pressione del pulsante di stop, che apre il circuito e toglie l'alimentazione alla bobina.

Il contatto K1 bypassa il pulsante di start, che da questo momento viene tagliato fuori dal circuito (nel senso che se viene premuto o meno, non è rilevante per il circuito).

Lo scopo principale dei circuiti in autoritenuta è di evitare la ripartenza automatica (rieccitazione della bobina) a seguito dell'interruzione dell'alimentazione.

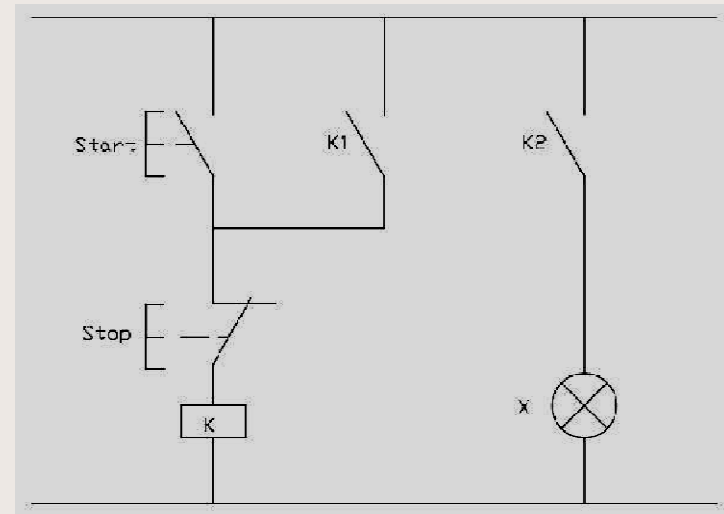
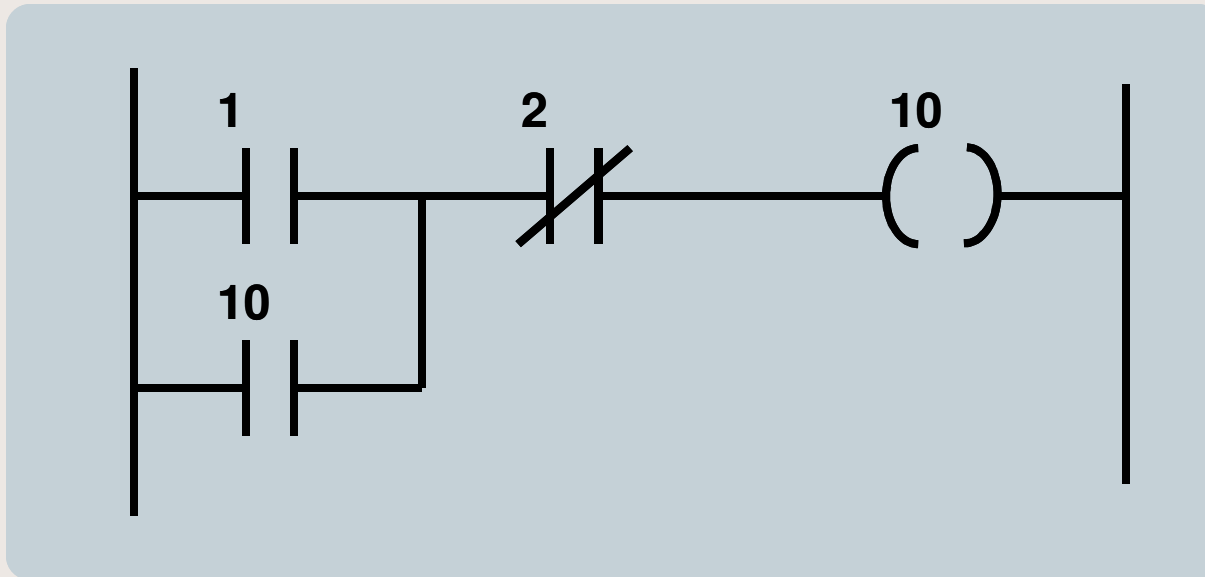


Diagramma Ladder (Grafico)



1 = Pulsante Start/Marcia NA

2 = Pulsante Stop/Arresto NC

10 = Contattore NA (K)

Lista istruzioni (Letterali)

Indirizzo	Istruzione	Operando
000	STR	1
001	OR	10
002	AND NOT	2
003	OUT	10

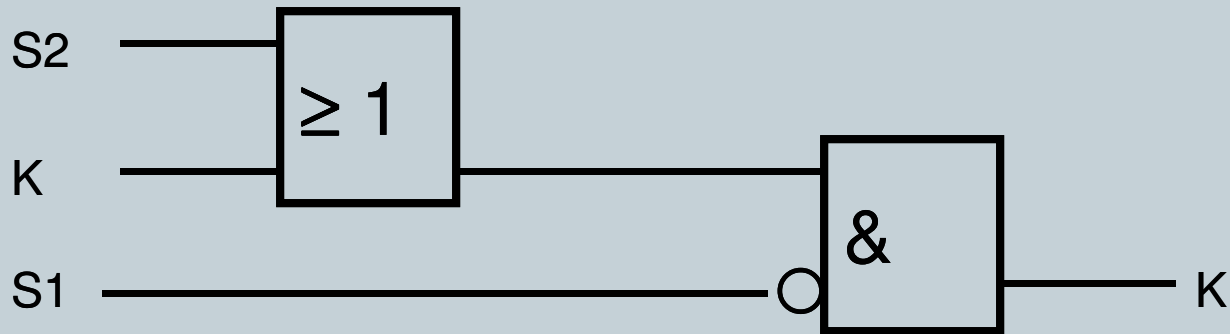
1 = Pulsante start (S2)

2 = Pulsante stop (S1)

10 = Contattore (K)

Porte logiche

$$(S2 + K) \cdot \bar{S1} = K$$

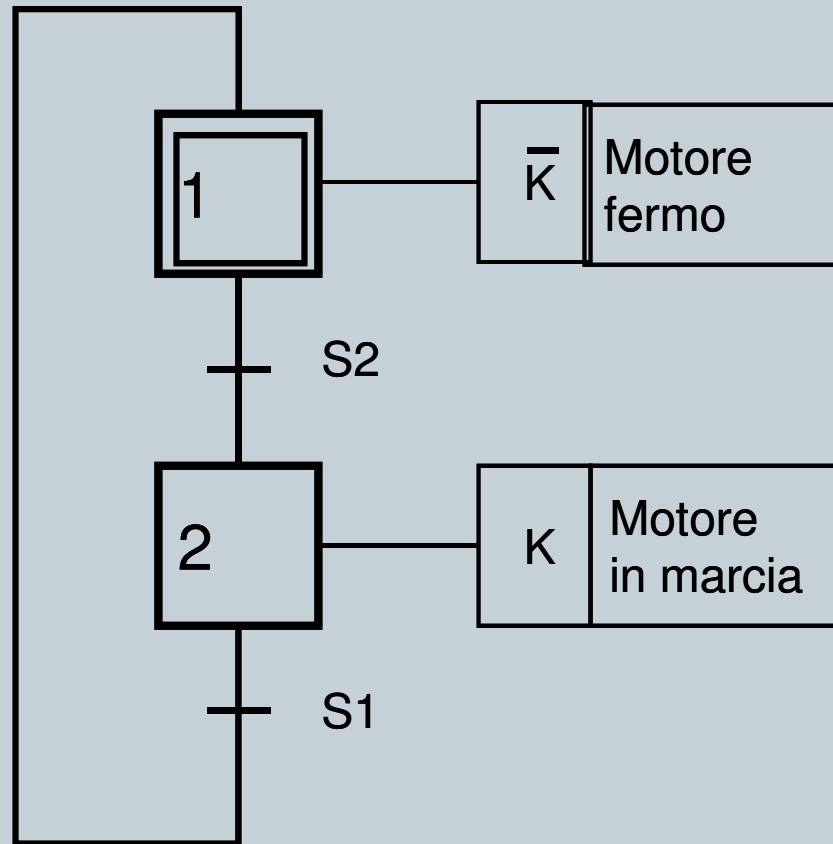


S2 = Pulsante start

S1 = Pulsante stop

K = Contattore

Grafcet



S2 = Pulsante start

S1 = Pulsante stop

K = Contattore

Il Linguaggio Ladder

❖ **POWER RAIL** (BARRA DI ALIMENTAZIONE A SX)

❖ **MASSA** (BARRA DESTRA)

❖ **LINEE ELETTRICHE ORIZZONTALI**

❖ **CONNESSIONI AI POWER RAIL**

Barra di alimentazione

Massa

❖ **CONTATTO NORMALMENTE APERTO**

(QUANDO IL CONTATTO È CHIUSO ON IN=1)

❖ **CONTATTO NORMALMENTE CHIUSO**

(QUANDO IL CONTATTO È CHIUSO OFF IN=0)

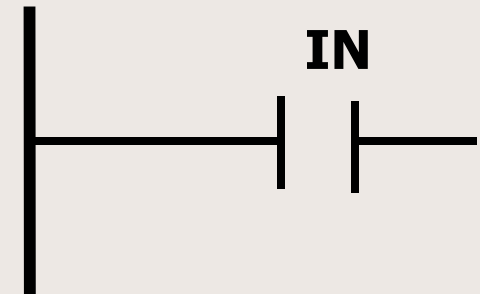
❖ **COIL/BOBINA** (USCITA A 1)

❖ **NEGATED COIL** (USCITA A 0)

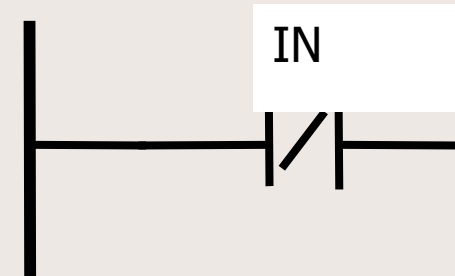
Utilizzo dei Contatti

AD OGNI CONTATTO VIENE ASSOCIATA UNA VARIABILE BINARIA. TALE VARIABILE VIENE SOLAMENTE LETTA (PUÒ COINCIDERE CON UN INGRESSO).

❖ **Contatto Normalmente Aperto:** la corrente fluisce da sinistra a destra se la variabile IN è 1. Viene usato questo segno se occorre attivare un'uscita quando il contatto è ON (chiuso, tensione presente, livello 1)



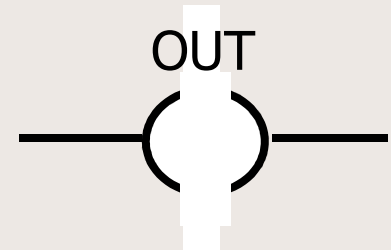
❖ **Contatto Normalmente Chiuso:** la corrente fluisce da sinistra a destra se la variabile IN è 0. Viene usato se occorre attivare un'uscita quando il contatto è OFF (aperto, tensione assente, livello 0)



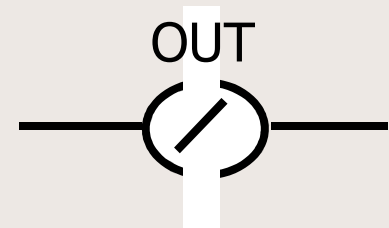
Utilizzo dei Coil

AD OGNI COIL VIENE ASSOCIATA UNA VARIABILE BINARIA. LA VARIABILE VIENE SCRITTA (PUÒ COINCIDERE CON UNA USCITA FISICA)

- ❖ **Coil:** la variabile OUT associata al Coil è posta a 1 se vi è una corrente che fluisce da sinistra.

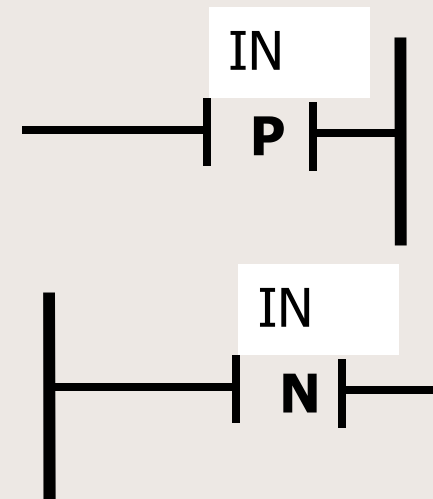


- ❖ **Negated Coil:** la variabile OUT è posta a 0 se vi è una corrente che fluisce da sinistra.



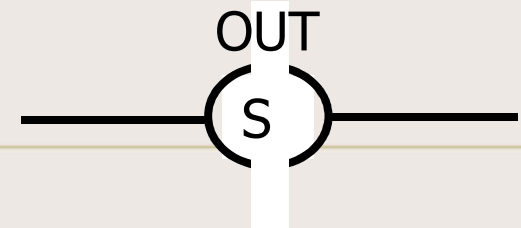
Altri Contatti del Linguaggio Ladder

- ❖ Contatto sensibile alla transizione 0-1
(Positive Transition-Sensing Contact)
- ❖ Contatto sensibile alla transizione 1-0
(Negative Transition-Sensing Contact)



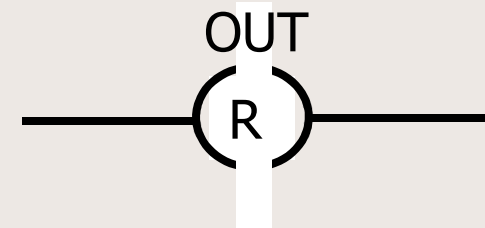
Altri Coil del Linguaggio Ladder

❖ SET COIL



La variabile OUT associata al **coil** e' posta a 1 se vi e' una corrente che fluisce da sinistra. La variabile rimane a 1 fino a quando viene utilizzato un coil RESET.

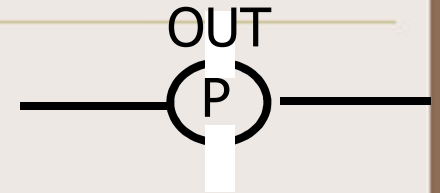
❖ RESET COIL



La variabile OUT associata al **coil** e' posta a 0 se vi e' una corrente che fluisce da sinistra. La variabile rimane a 0 r fino a quando viene utilizzato un coil SET.

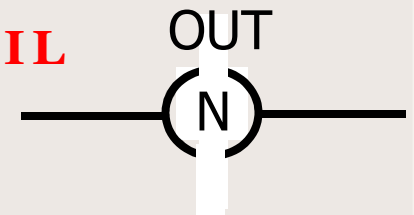
Altri Coil del Linguaggio Ladder

❖ POSITIVE TRANSITION-SENSING COIL



La variabile OUT associata al **Positive Transition-Sensing Coil** è posta a 1 se la corrente che fluisce da sinistra passa da un valore FALSE ad un valore TRUE.

❖ NEGATIVE TRANSITION-SENSING COIL



La variabile OUT associata al **Negative Transition-Sensing Coil** è posta a 1 se la corrente che fluisce da sinistra passa da un valore TRUE ad un valore FALSE.

Esempio di schemi ladder

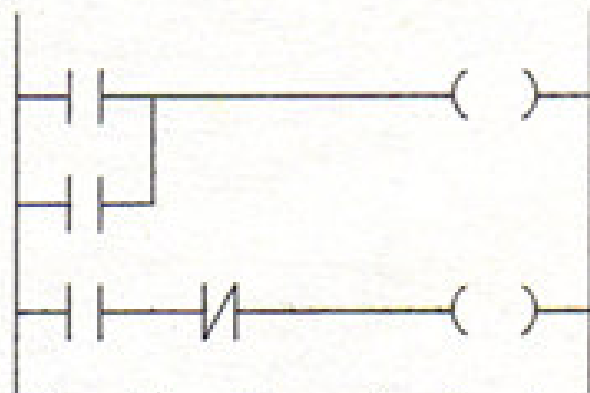


Fig. 4.4 - Esempio di schema ladder semplificato (abituamente usato).

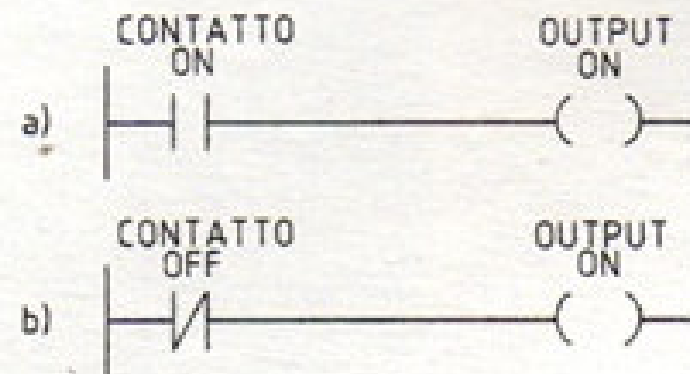


Fig. 4.5 - Uso dei contatti ON e OFF per l'attivazione delle uscite:

- a) l'uscita è attiva quando il contatto è in stato ON (chiuso);
- b) l'uscita è attiva quando il contatto è in stato OFF (aperto).

Tecniche di Programmazione (conversione schema **Funzionale** in schema **Ladder**)

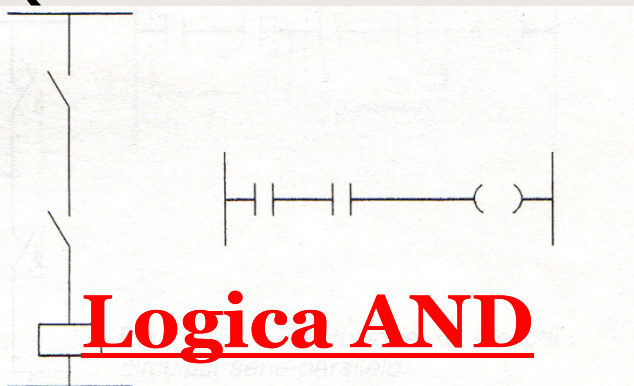


Fig. 4.8 - Schema funzionale con contatti in serie, convertito in schema ladder.

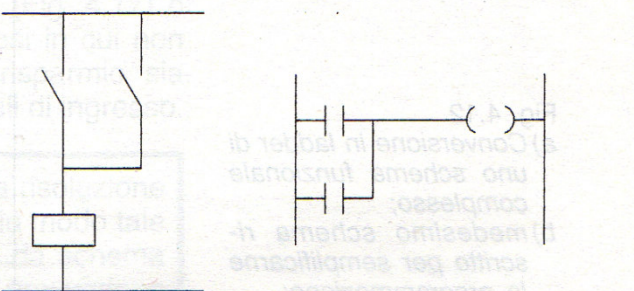


Fig. 4.9 - Schema funzionale con contatti in parallelo, convertito in schema ladder.

Logica OR

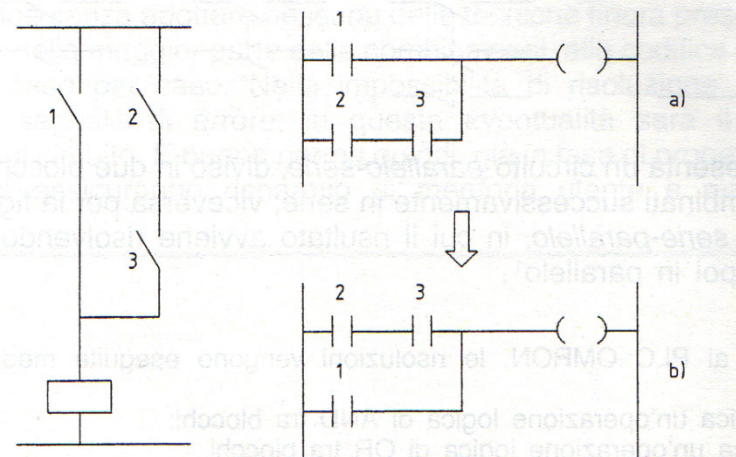
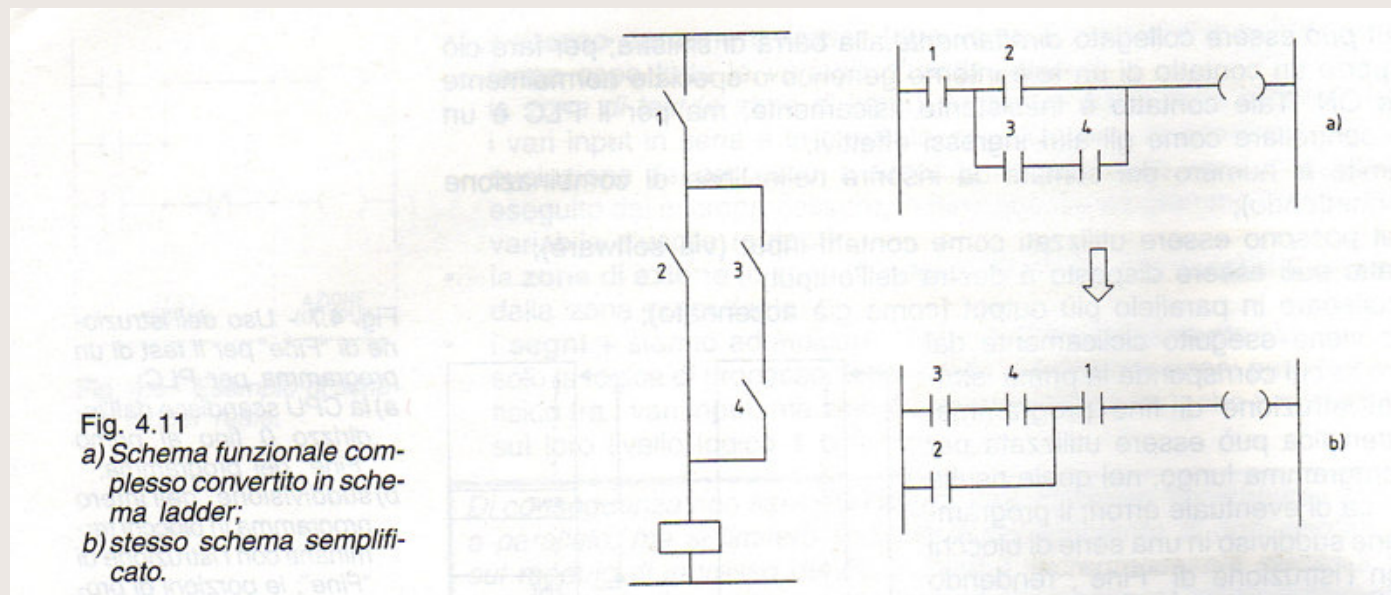


Fig. 4.10

a) Schema funzionale con contatti in serie e parallelo, convertito in schema ladder;

b) stesso schema semplificato.

Tecniche di Programmazione (conversione schema **Funzionale** in schema **Ladder**)



Tecniche di Programmazione

(conversione schema **Funzionale** in schema **Ladder**)

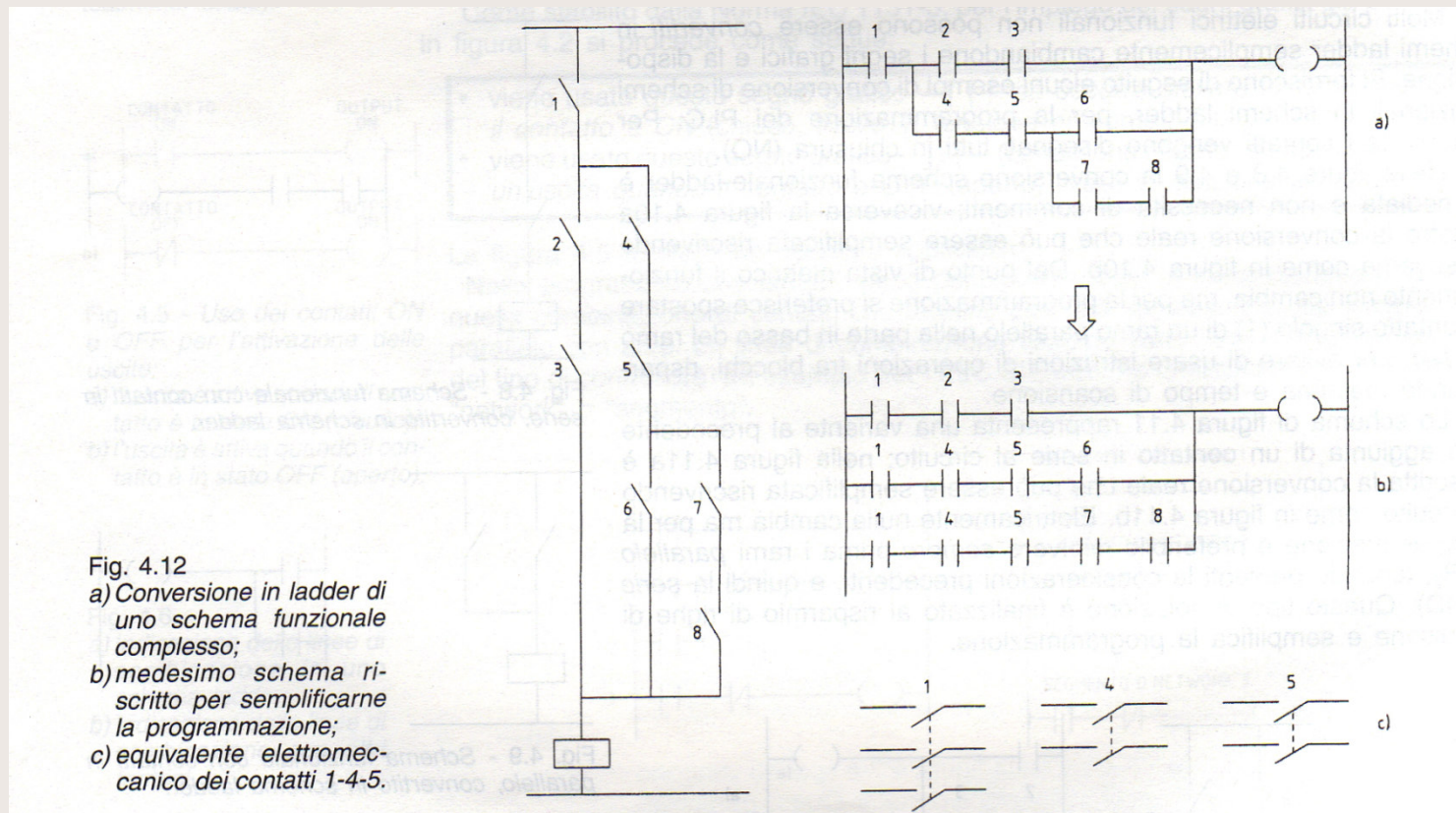


Fig. 4.12

- a) Conversione in ladder di uno schema funzionale complesso;
- b) medesimo schema riscritto per semplificarne la programmazione;
- c) equivalente elettromeccanico dei contatti 1-4-5.

Tecniche di Programmazione (conversione schema **Funzionale** in schema **Ladder**)

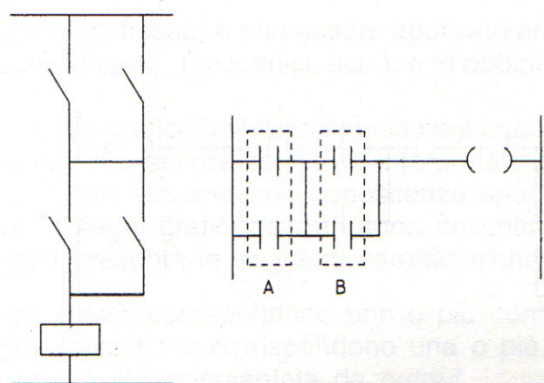


Fig. 4.13 - Schema funzionale con contatti parallelo-serie, convertito in schema ladder.

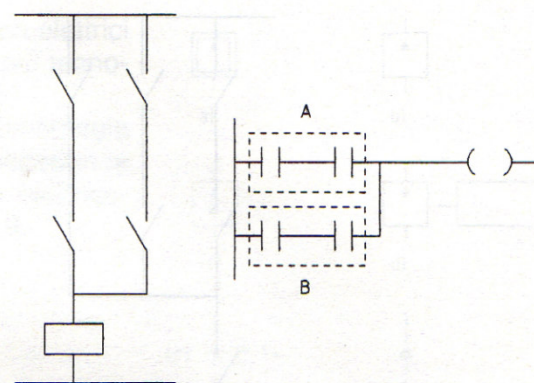


Fig. 4.14 - Schema funzionale con contatti serie-parallelo convertito in schema ladder.

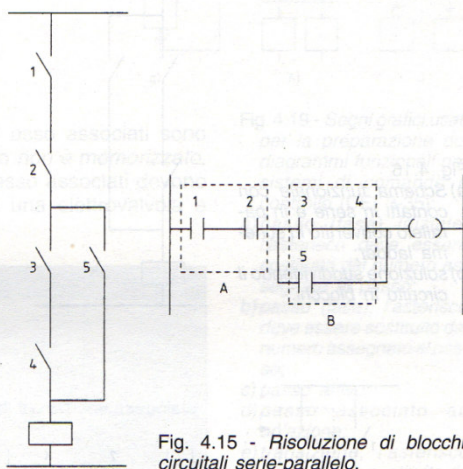
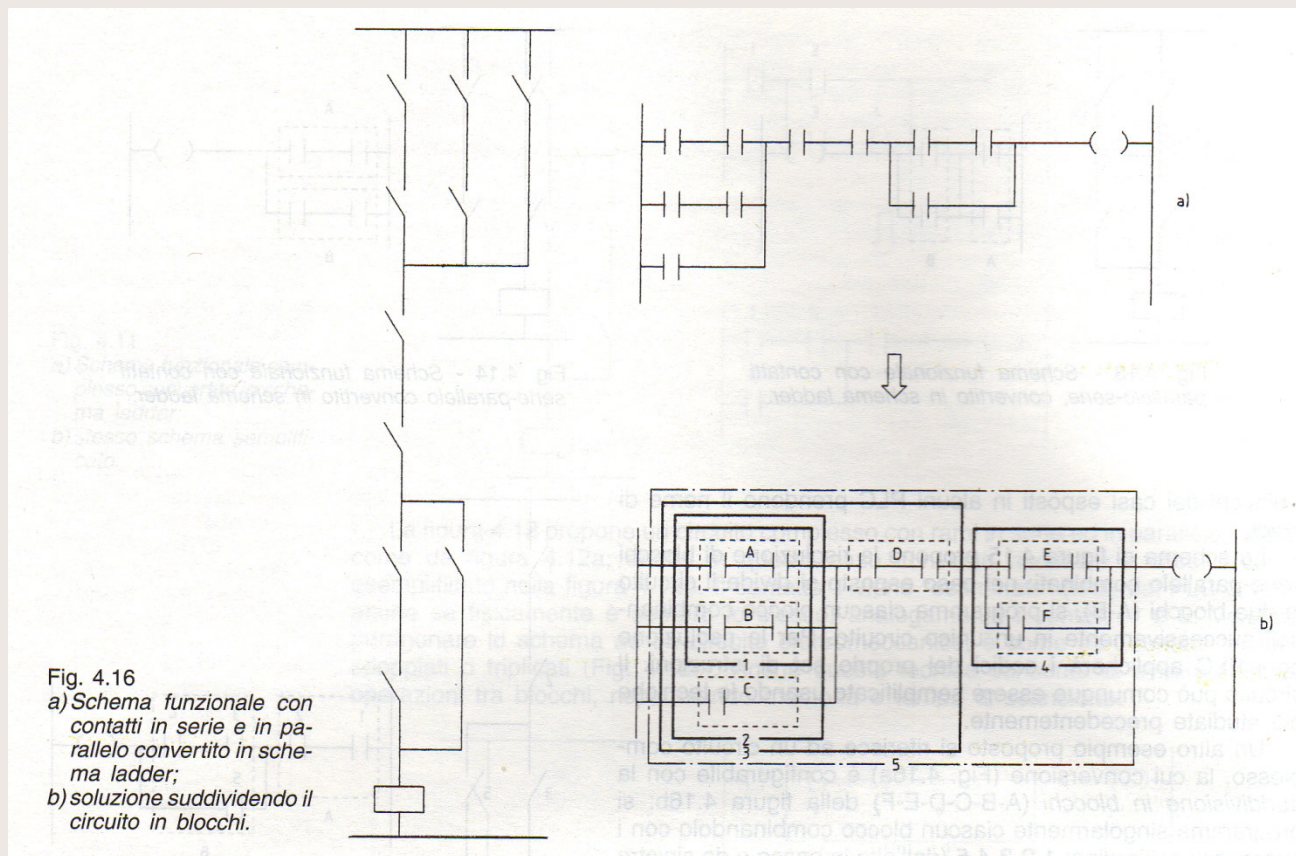


Fig. 4.15 - Risoluzione di blocchi circuitali serie-parallelo.

Tecniche di Programmazione (conversione schema **Funzionale** in schema **Ladder**)



Esempio: Circuito di autoritenuta relè

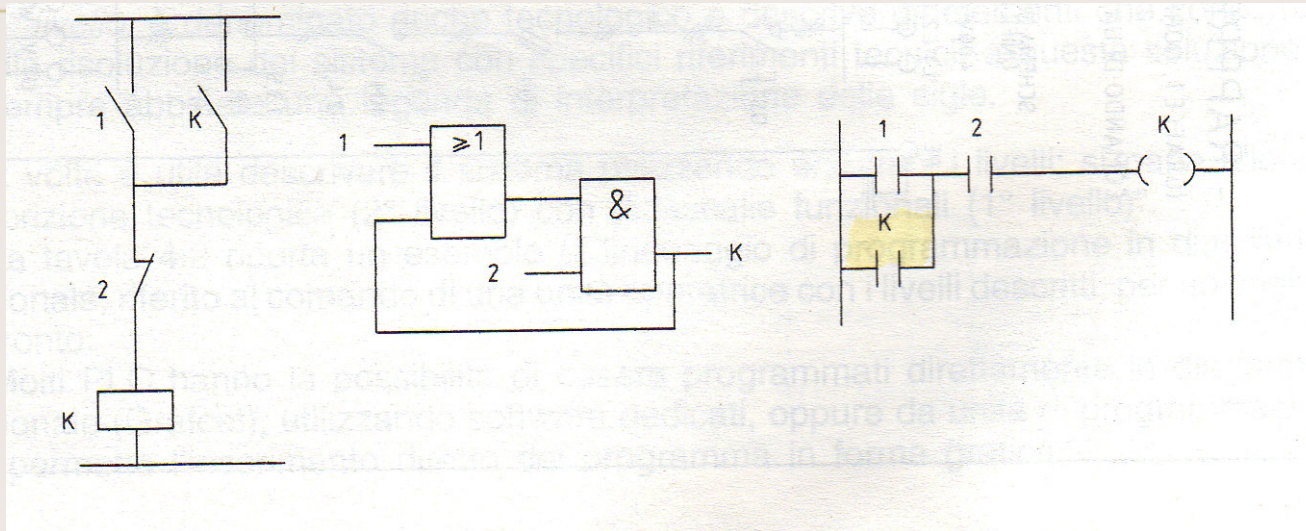
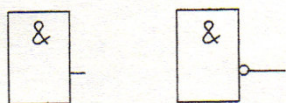


Fig. 4.26 - Circuito di autoritenuta a contatti risolto con linguaggio a blocchi logici e ladder.

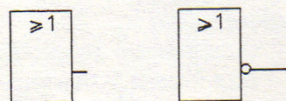
FUP

Tecniche di Programmazione Linguaggio a Blocchi Logici

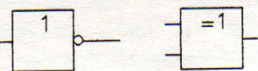
AND NAND



OR NOR



NOT XOR



Bistabile RS

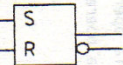


Fig. 4.24 - Esempio di segni grafici per linguaggio a blocchi logici (da Norma CEI 3-26).

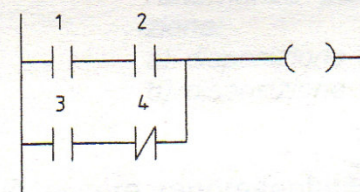
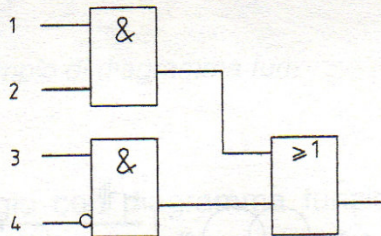


Fig. 4.25 - Confronto tra linguaggio ladder e a blocchi logici.

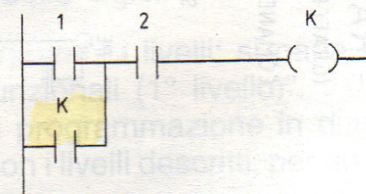
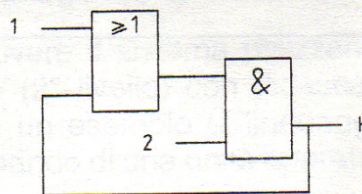
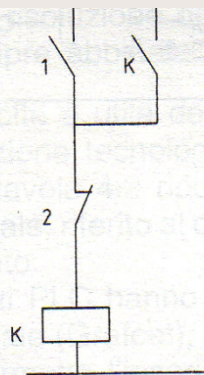
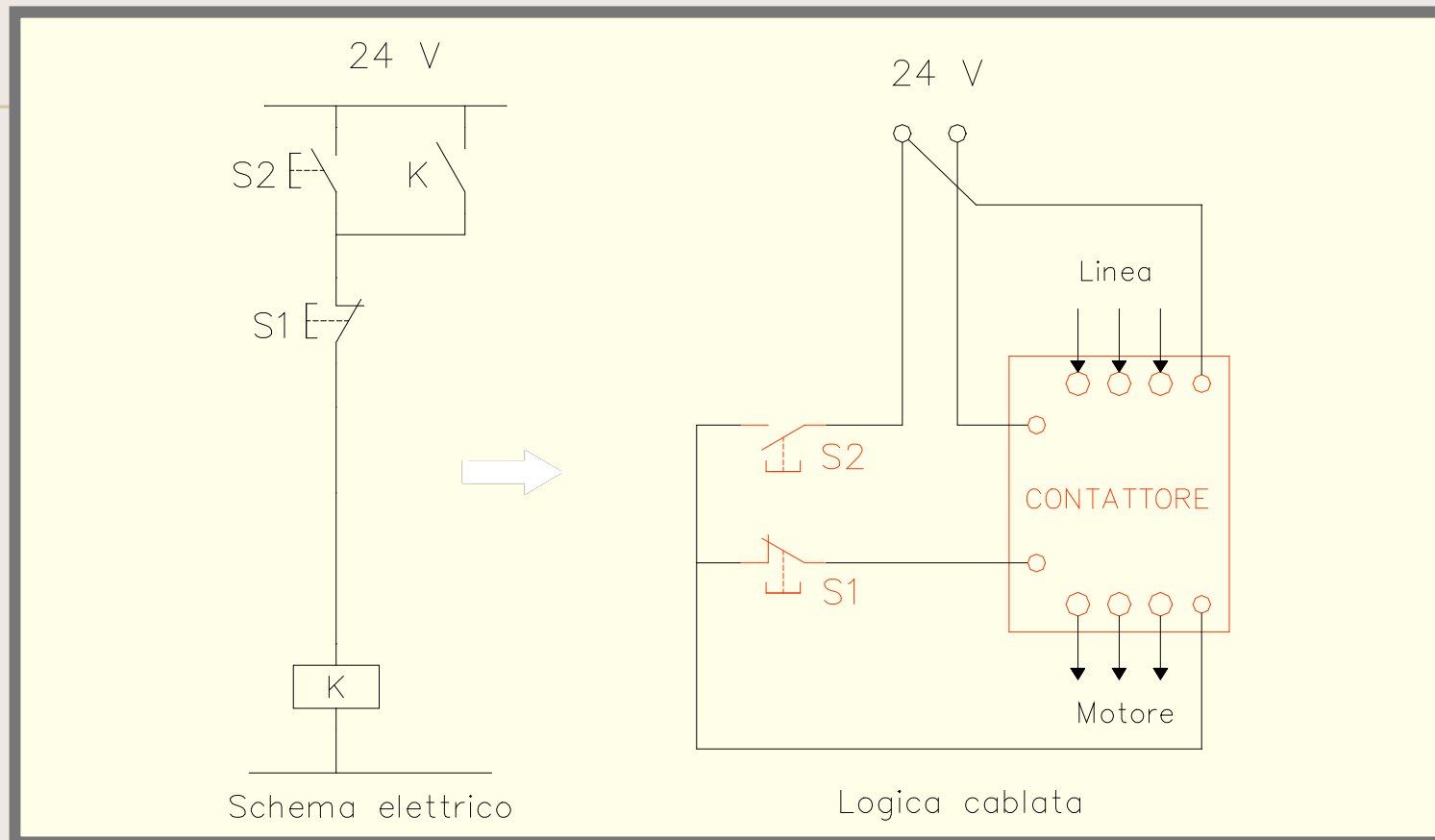


Fig. 4.26 - Circuito di auto-ritenuta a contatti risolto con linguaggio a blocchi logici e ladder.

ESEMPIO DI LOGICA CABLATA

(marcia - arresto di un motore)



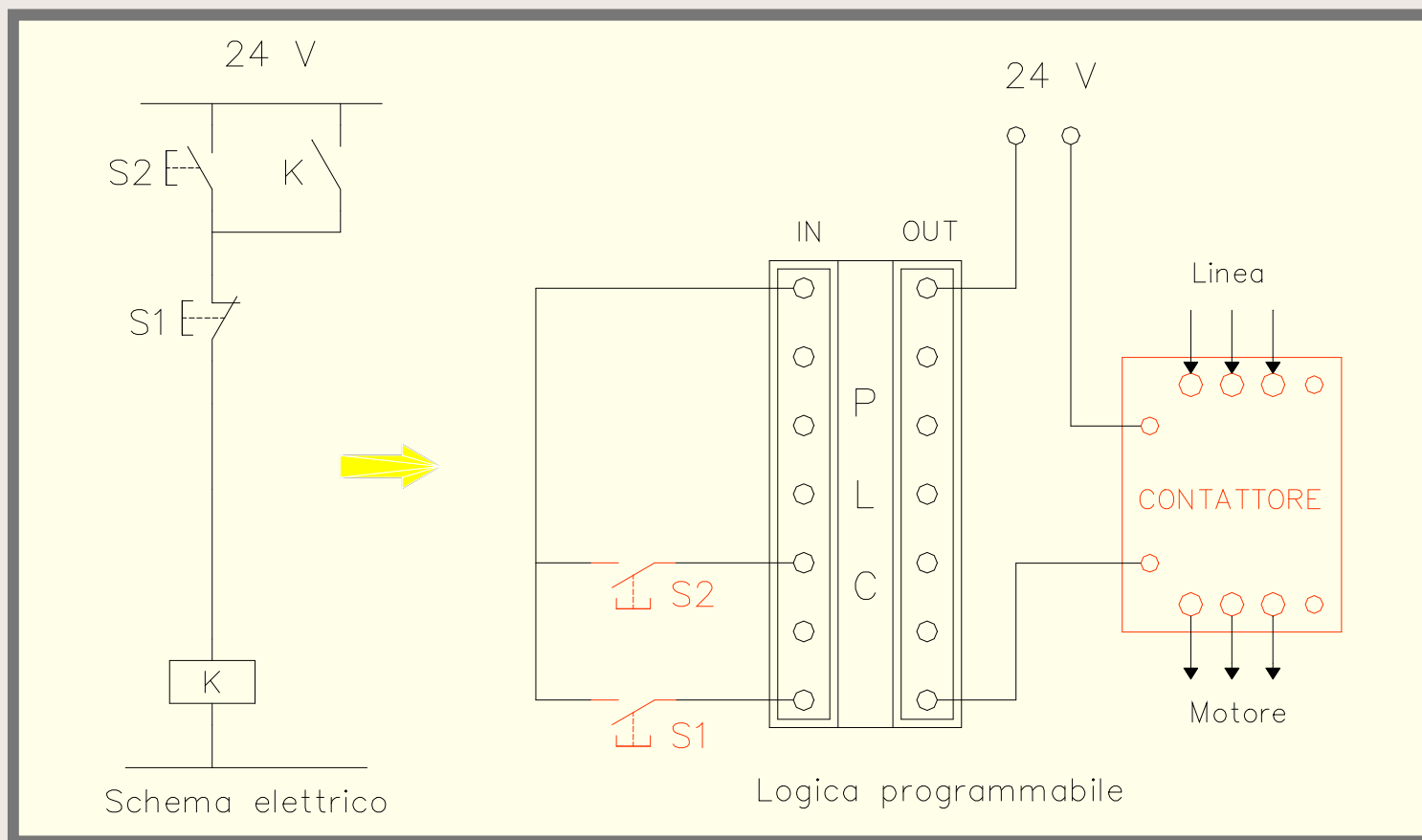
1 = Pulsante marcia (S2) 2 = Pulsante arresto (S1)

10 = Contattore (K)

Il contattore conosciuto pure come teleruttore, è un componente elettrico che viene impiegato largamente nell'avviamento dei motori elettrici.

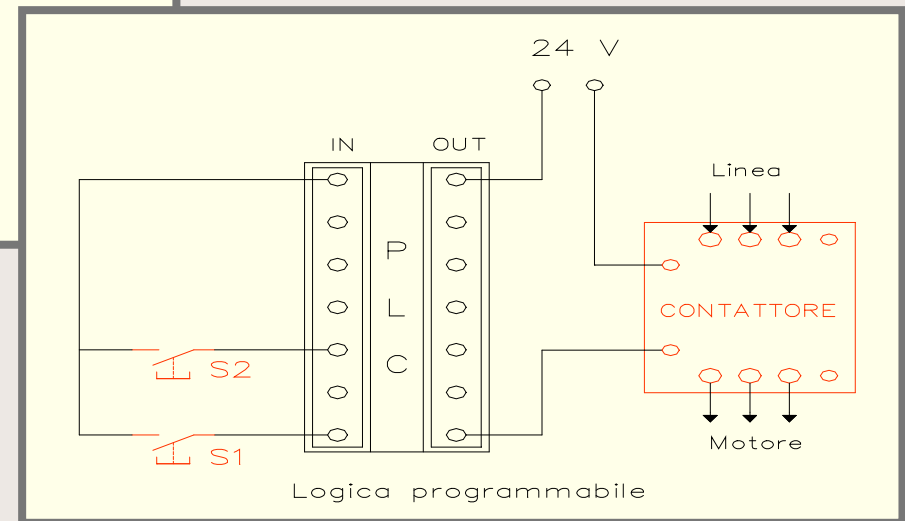
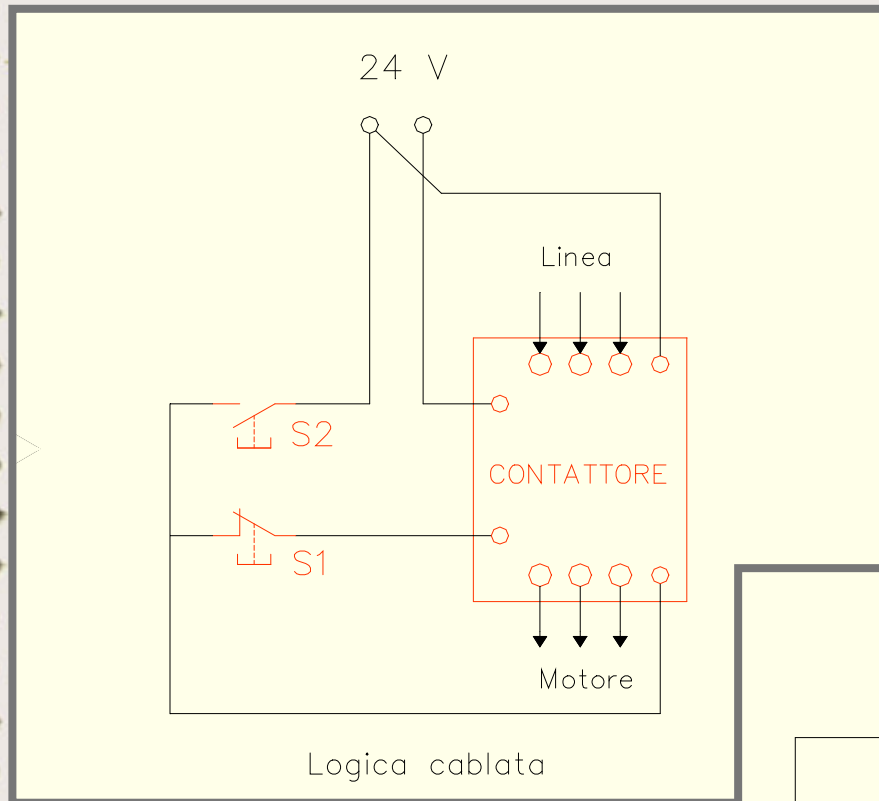
ESEMPIO DI LOGICA PROGRAMMABILE

(marcia - arresto di un motore)



DA LOGICA CABLATA A LOGICA PROGRAMMABILE

(marcia - arresto di un motore)

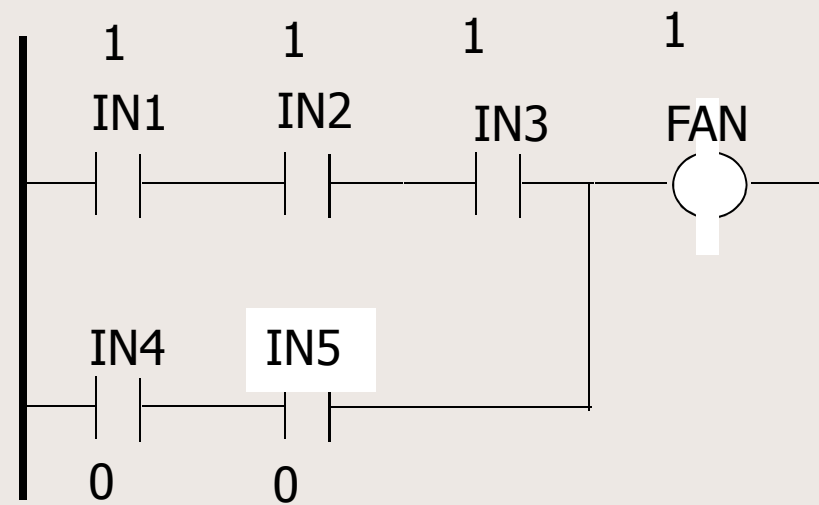
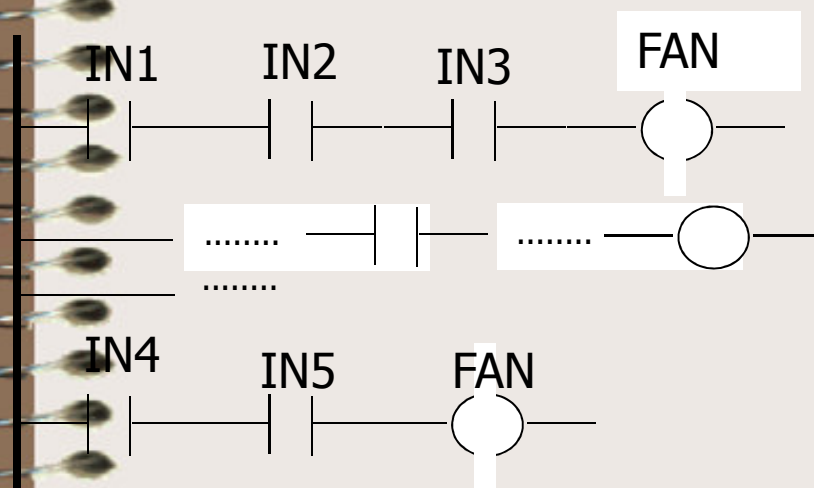


Regole di Esecuzione dei Rung

- ❖ UN PROGRAMMA SCRITTO IN LINGUAGGIO LADDER VIENE ESEGUITO VALUTANDO UN RUNG ALLA VOLTA
- ❖ L'ORDINE DI VALUTAZIONE DEI RUNG È QUELLO CHE PROCEDE DAL PRIMO RUNG IN ALTO VERSO L'ULTIMO RUNG IN BASSO
- ❖ QUANDO L'ULTIMO RUNG VIENE VALUTATO, SI INIZIA NUOVAMENTE A VALUTARE IL PRIMO RUNG (DOPO AVER AGGIORNATO LE USCITE E LETTI GLI INGRESSI)

Effetti Collaterali delle Regole di Esecuzione dei Rung

- ❖ RIUNIFICANDO I RUNG, L'USCITA REALE COLLEGATA ALLA VARIABILE FAN VERRÀ AGGIORNATA SOLO DOPO AVER VALUTATO IL RUNG COMPOSTO DAGLI INGRESSI IN1, IN2, IN3, IN4 E IN5



Function Block di Uso Comune

❖ Bistabili

➤ SR, RS

❖ Bitwise Boolean

➤ AND, OR, NOT, XOR

❖ Comparison

➤ EQ, LE, LT, GE, GT, NE

❖ Counters

➤ CTD (down), CTU (up)

❖ Altri (disponibili su alcuni PLC)

➤ PID