



Azionamenti e sistemi per l'automazione e la domotica

Luca Sani

Dipartimento di Sistemi Elettrici e Automazione

Università di Pisa

tel. 050 2217364

email luca.sani@dsea.unipi.it



Linguaggi di programmazione

Programmazione del PLC

I vari linguaggi di programmazione sono stati standardizzati con la normativa IEC 1131.

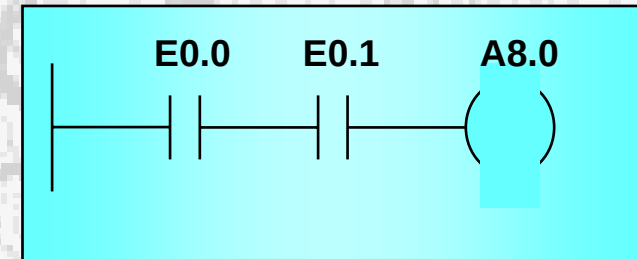
Sono definiti i seguenti linguaggi

- **LADDER**
- **LISTA ISTRUZIONI**
- **FUNCTIONAL BLOCK DIAGRAM**
- **SFC**

I primi tre sono linguaggi a basso livello, l'ultimo è un linguaggio ad alto livello

Ladder (KOP)

Si rifà agli schemi di controllo in logica cablata (si rifà al concetto di contatto NA e NC)



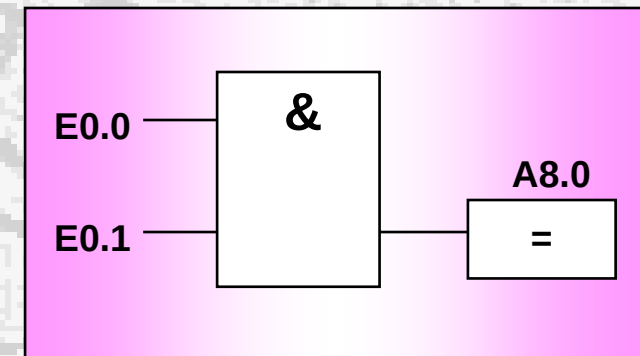
Lista istruzioni (AWL)

E' un set di istruzioni (simile all'Assembler) con cui si descrivono le operazioni booleane da realizzare affinché si verifichi una determinata azione.

U	E0.0
U	E0.1
=	A8.0

Functional Block Diagram (FUP)

Le istruzioni di controllo sono codificate da un linguaggio grafico nato originariamente per descrivere reti digitali.

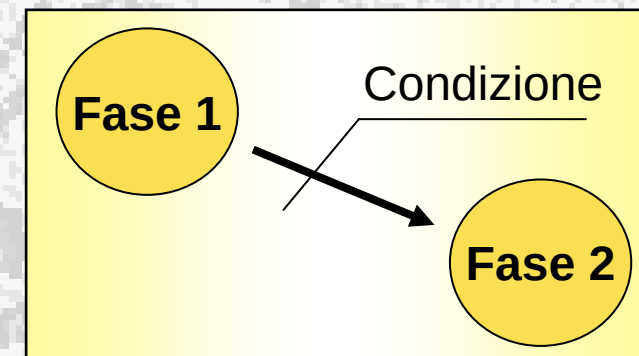


Sequential Functional Chart (GRAPH)

Si impiegano i concetti di FASE e TRANSIZIONE

Fase = Azione da compiere

Transizione = condizione da soddisfare per passare da una fase all'altra





Linguaggio Ladder (a contatti)

Schema di controlli a logica cablata

In un impianto elettrico di una macchina si distingue sempre un **circuito di potenza** e quello di **comando**.

Quello di potenza rappresenta l'alimentazione degli attuatori (in genere i motori).

Quello di comando (circuito di controllo) è sempre a bassa tensione e permette di modificare la configurazione del circuito di potenza.

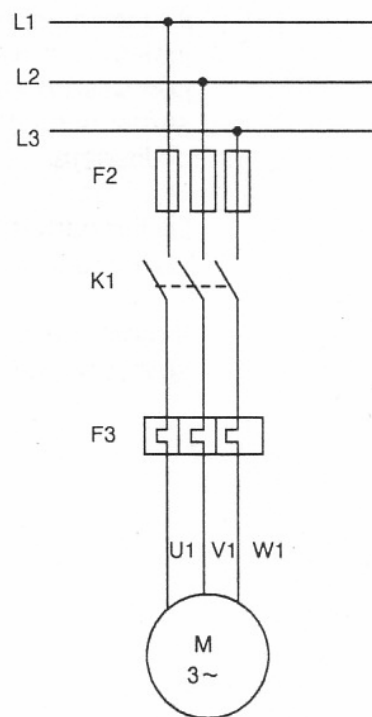
I primi sistemi di controllo si basavano su dispositivi elettromeccanici (relay), temporizzatori, contattori. La progettazione di tali dispositivi utilizzavano schemi a contatti (NA, NC). Si parlava di schemi a **logica cablata**.

I circuiti di potenza fanno uso dei contattori.

Esempio di circuito di comando in logica cablata

L'esempio più semplice di circuito di comando in logica cablata è quello relativo al comando “marcia/arresto” di un motore trifase.

Il circuito di potenza è:



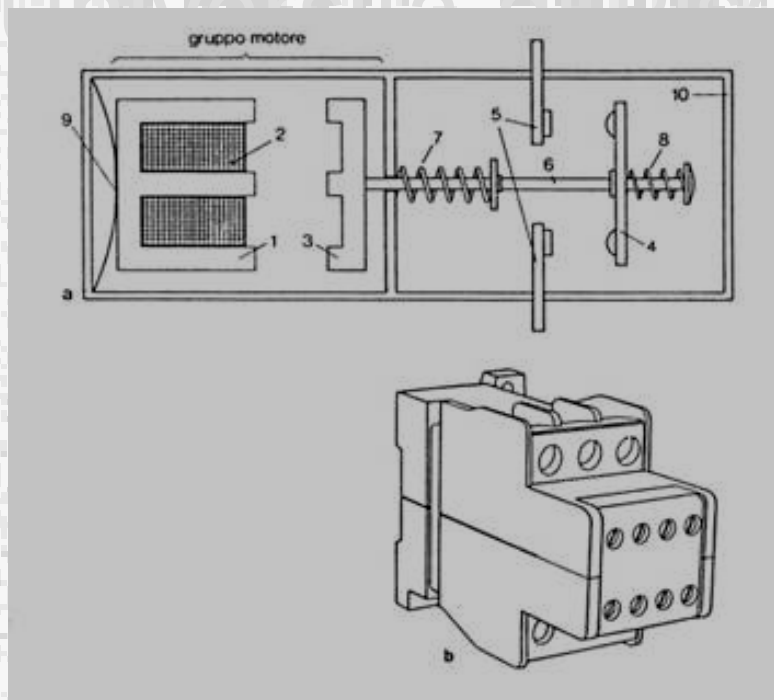
K1 è il contattore

F3 è il termico (protezione da sovraccarichi)

F2 sono i fusibili (protezione da corto circuiti)

Esempio di circuito di comando in logica cablata

Il contattore è un **dispositivo meccanico di manovra, generalmente previsto per un numero elevato di operazioni, avente una sola posizione di riposo, ad azionamento non manuale, capace di stabilire, sopportare ed interrompere correnti in condizioni di sovraccarico. La posizione di riposo corrisponde ordinariamente alla posizione di apertura dei contatti principali**“ (CEI 17-3, Fascicolo 1035)



- 1 - Nucleo fisso
- 2 - Bobina
- 3 - Nucleo mobile
- 4 - Contatti mobili
- 5 - Contatti fissi
- 6 - Cursore
- 7 - Molla di rimando (di apertura)
- 8 - Molla di pressione dei contatti principali
- 9 - Molla del sistema ammortizzante
- 10 - Struttura di sostegno e di contenimento

Esempio di circuito di comando in logica cablata

Il contattore presenta una serie di contatti ausiliari che possono essere utilizzati nel circuito di comando.

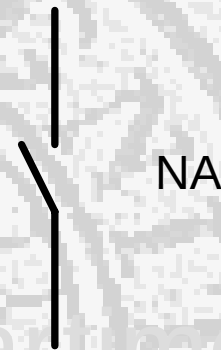
Nel circuito di comando si fa uso solo di **pulsanti** (elemento di comando con una sola posizione stabile) e non di **interruttori** (elemento di comando con due posizioni stabili) con cui si vanno ad eccitare le bobine dei contattori.

Un tale soluzione conferisce al circuito di comando una "sicurezza intrinseca", dovuta al fatto che la mancanza o un abbassamento di tensione nel circuito di alimentazione della bobina provoca automaticamente la diseccitazione della bobina stessa e quindi l'apertura del contattore interrompendone il servizio.

Inoltre, al riapparire della tensione al suo valore di normale funzionamento, impedisce la richiusura del contattore, evitando così la partenza indesiderata del motore (cosa che invece si verificherebbe se la bobina fosse comandata tramite un interruttore).

Esempio di circuito di comando in logica cablata

Nella logica cablata si parla di contatto normalmente aperto (NA) quando in condizioni di riposo questo è aperto.

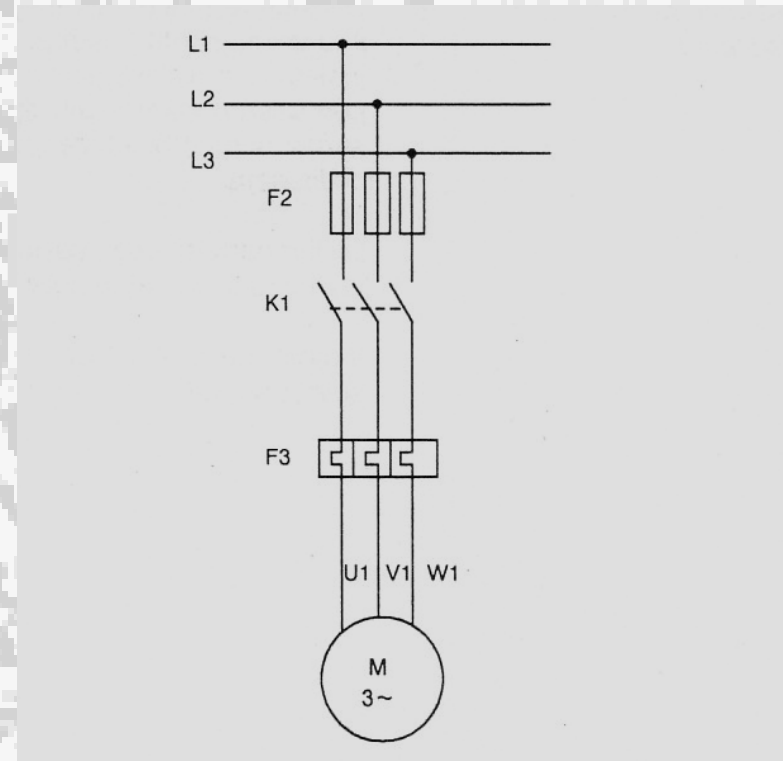
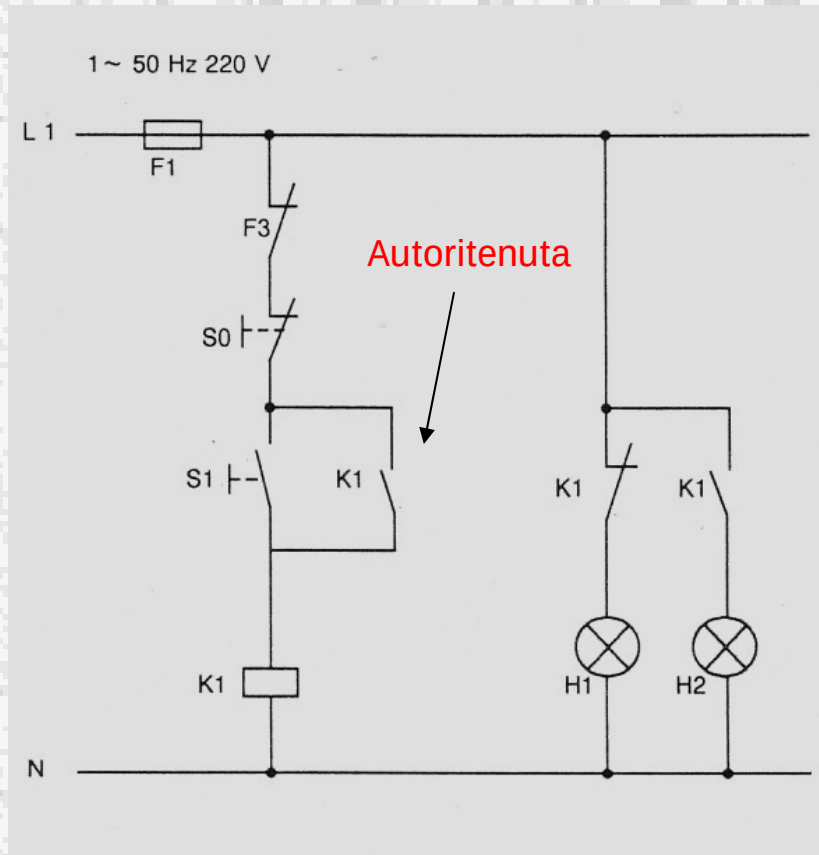


Nella logica cablata si parla di contatto normalmente chiuso (NC) quando in condizioni di riposo questo è chiuso.



Esempio di circuito di comando in logica cablata

Il circuito di comando per la marcia/arresto del motore è:



K1 contatti ausiliari del contattore

Principi del linguaggio Ladder

E' il più vecchio linguaggio di programmazione per PLC. Per permettere agli operatori un rapido passaggio all'impiego dei PLC la logica degli schemi a contatti è stata mantenuta come linguaggio di programmazione.

Si basa su simboli di provenienza "elettrica":

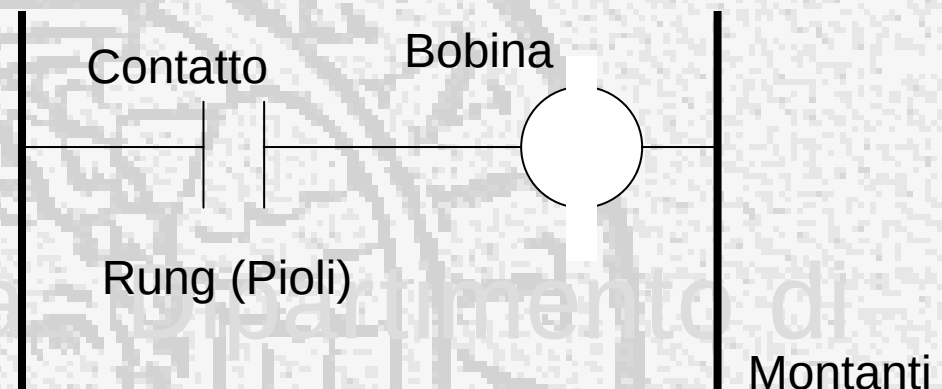
- binari di potenza (power rail),
- contatti elettrici
- avvolgimenti magnetici (coil)

Si articola in linee orizzontali dette "rung"

Ciascun "rung" può contenere contatti, bobine, blocchi funzionali.

Principi del linguaggio Ladder

Ciascun "rung" deve essere connesso necessariamente al binario di potenza sinistro (left power rail), mentre il collegamento con quello destro è opzionale



MONTANTI = sono le due linee dell'alimentazione

RUNG = rappresentano le combinazioni dello stato dei contatti che si devono verificare affinché una bobina sia eccitata/diseccitata.

Un programma è un insieme di Rung

Regole di esecuzione dei Rung

Un programma scritto in linguaggio Ladder viene eseguito valutando un rung alla volta.

Il flusso di energia può andare sempre e solo da sinistra a destra, senza possibilità di inversione.

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).

Nel caso in cui vengano disegnati rung adiacenti, l'ordine di valutazione generalmente procede dal rung posto a sinistra verso quello posto alla destra.

Elementi di base del linguaggio Ladder

<u>Power rail</u>	
<u>Linee Elettriche orizzontali</u>	
<u>Connessioni ai power rail</u>	
<u>Contatto normalmente aperto</u>	
<u>Contatto normalmente chiuso</u>	
<u>Bobine (Coil)</u>	
<u>Negated coil</u>	

Ladder Codifica S7

Ogni elemento è individuato dal seguente codice:

X A.B

X lettera che identifica il tipo di elemento
A.B indirizzo dell'elemento

Elemento	Simbolo	Esempio
INGRESSI	E	(E 0.4)
USCITE	A	(A 3.4)
VARIABILI INTERNE	M	(M 1.4)
TEMPORIZZATORI	T	(T 2.4)
CONTATORI	C	(C 0.4)

Utilizzo dei contatti

Ad ogni contatto viene associata una variabile binaria. Tale variabile viene **solamente letta** (può coincidere con l'immagine di ingresso o di un'uscita, una variabile interna, lo stato di un temporizzatore o di un contatore). Ossia un contatto è un elemento di valutazione dello stato di variabili booleane (bit) e servono per permettere o no il flusso di energia sul rung dove si trovano.

Si possono avere:

- Contatti normalmente aperti (NA)
- Contatti normalmente chiusi (NC)

Contatti NA

---| |--- (**Contatto normalmente aperto**) è chiuso se il valore della variabile associata al contatto (IN) è "1". Se il contatto è chiuso, la corrente passa attraverso il contatto, e il bit di risultato logico combinatorio (RLC) è "1". La corrente fluisce a destra **per qualunque scansione del Programma Ladder** fino a quando la variabile IN diviene 0.

Se il valore della variabile (IN) associata al contatto è "0", il contatto è aperto. Se il contatto è aperto, la corrente non passa, e il bit di risultato logico combinatorio dell'operazione (RLC) è "0".

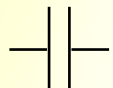
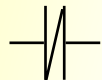
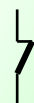


Utilizzo dei contatti

---| / |--- (**Contatto normalmente chiuso**) è chiuso se il valore del bit interrogato, che viene memorizzato nella variabile (IN) associata al contatto, è "0". Se il contatto è aperto, la corrente passa attraverso il contatto e il bit di risultato logico (RLC) è "1". La corrente fluisce a destra **per qualunque scansione del Diagramma Ladder** fino a quando la variabile IN diviene 1. Se il valore della variabile IN è "1", il contatto è aperto, la corrente non passa, e il bit di risultato logico combinatorio dell'operazione (RLC) è "0".



Contatti Normalmente Aperti (NA) e Normalmente Chiusi (NC)

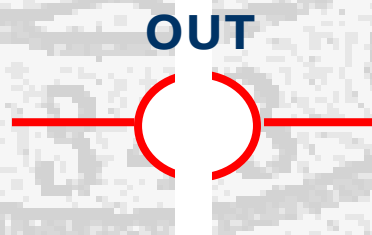
Processo			Interpretazione nel programma utente				
Il sensore è un...	Il sensore viene ...	C'è tensione all'ingresso?	Stato del segnale all'ingresso	Interrogazione per Stato del segnale "1"		Interrogazione per Stato del segnale "0"	
				Simbolo / Istruzione	Risultato dell'interr.	Simbolo / Istruzione	Risultato dell'interr.
Contatto NA 	attivato 	Si	1	"Contatto NA" 	"Si" 1	"Contatto NC" 	"No" 0
	non attivato 	No	0		"No" 0		"Si" 1
Contatto NC 	attivato 	No	0		"No" 0		"Si" 1
	non attivato 	Si	1		"Si" 1		"No" 0

Utilizzo delle bobine (coil)

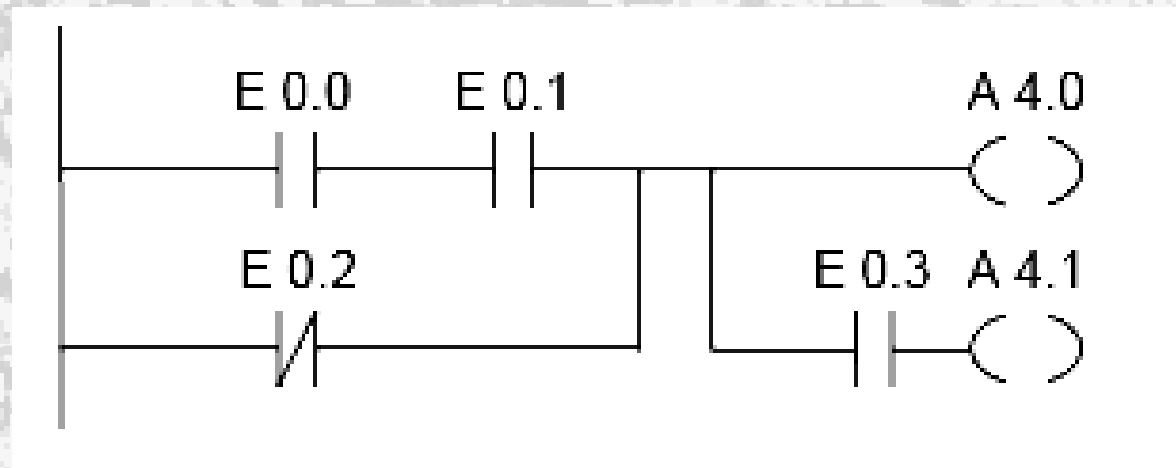
Ad ogni bobina viene associata una variabile binaria. La variabile viene **scritta** (può coincidere con l'immagine di un'uscita digitale, oppure una variabile interna, un temporizzatore o un contatore).

Bobina: la variabile OUT associata al **Coil** è posta a 1 se vi è una corrente che fluisce da sinistra. La variabile rimane a 1 **per qualunque scansione del Programma Ladder** fino a quando la corrente cessa di fluire da sinistra. Se invece la corrente non può circolare attraverso il circuito per raggiungere la bobina la variabile OUT viene impostato a "0".

Una bobina di uscita può essere posizionata solo all'estremità destra di un percorso di corrente di uno schema a contatti.



Utilizzo delle bobine Esempio



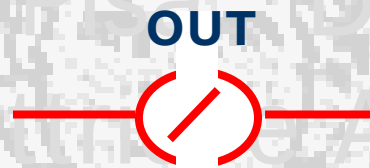
L'uscita A 4.0 è "1" se:
(lo stato all'ingresso E 0.0 e E 0.1 è "1") o lo stato all'ingresso E 0.2 è "0".

L'uscita A 4.1 è "1" se:
(lo stato all'ingresso E 0.0 e E 0.1 è "1" o lo stato all'ingresso E 0.2 è "0") e
lo stato all'ingresso E 0.3 è "1".

(e = operazione AND; o = operazione OR)

Utilizzo dei coil

Negated Coil: la variabile OUT associata al **Negated Coil** è posta a 0 se vi è una corrente che fluisce da sinistra. La variabile rimane a 0 per qualunque scansione del Programma Ladder fino a quando la corrente cessa di fluire da sinistra.



Altri contatti del linguaggio Ladder (1/2)

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



La corrente fluisce da sinistra a destra del Positive Transition-Sensing Contact, se la variabile IN passa da 0 a 1. La corrente fluisce a destra solo per una scansione del Programma Ladder (quella relativa alla transizione).

Scansione	Valore di IN quando viene valutato il rung	Corrente alla Destra
1	OFF	OFF
2	ON	ON
3	ON	OFF
4	ON	OFF
5	OFF	OFF

Altri contatti del linguaggio Ladder (2/2)

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



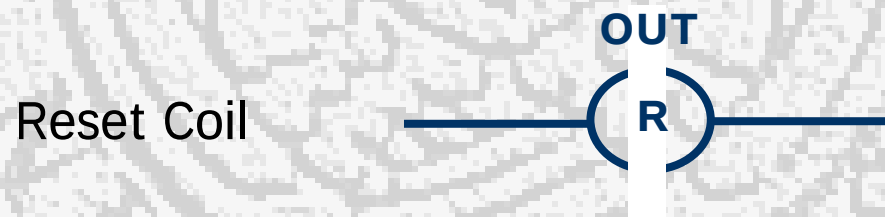
La corrente fluisce da sinistra a destra del Negative Transition-Sensing Contact, se la variabile IN passa da 1 a 0. La corrente fluisce a destra solo per una scansione del Diagramma Ladder (quella relativa alla transizione).

Scansione	Valore di IN quando viene valutato il rung	Corrente alla Destra
1	ON	OFF
2	OFF	ON
3	OFF	OFF
4	OFF	OFF
5	ON	OFF

Altri coil del linguaggio Ladder

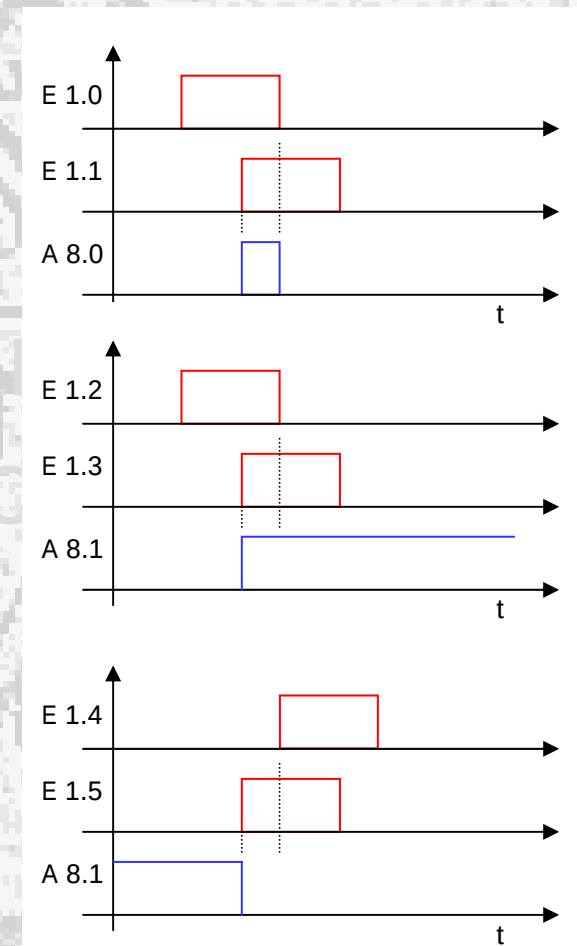
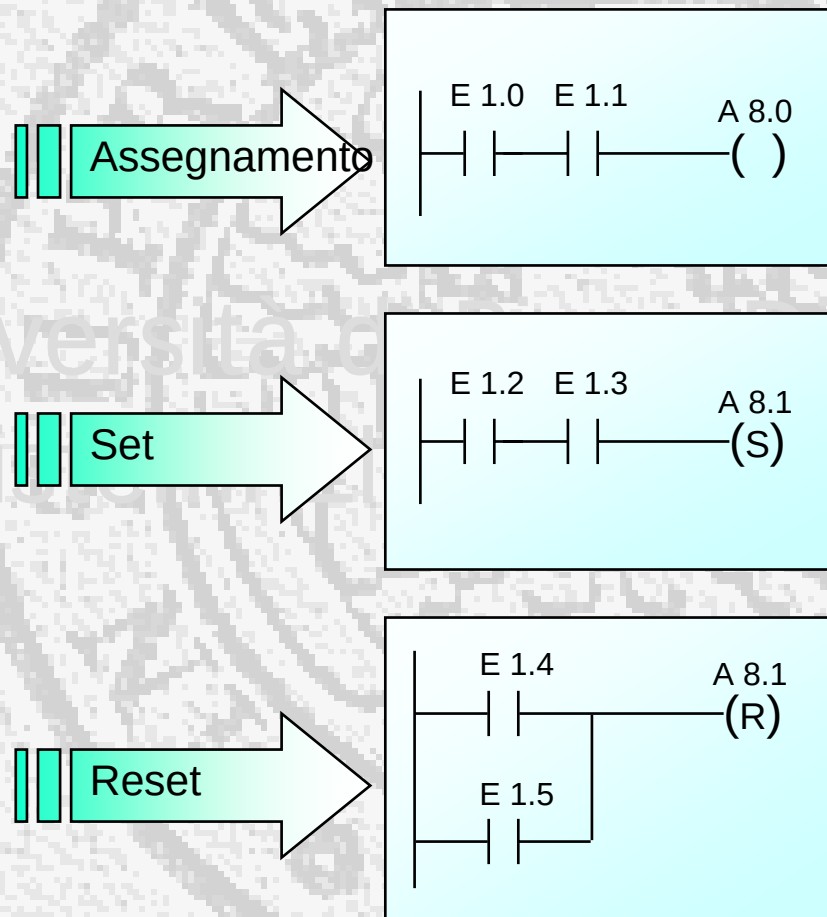


La variabile OUT associata al coil e' posta a 1 se vi e' una corrente che fluisce da sinistra. La variabile rimane a 1 per qualunque scansione del Diagramma Ladder fino a quando viene utilizzato un coil RESET.

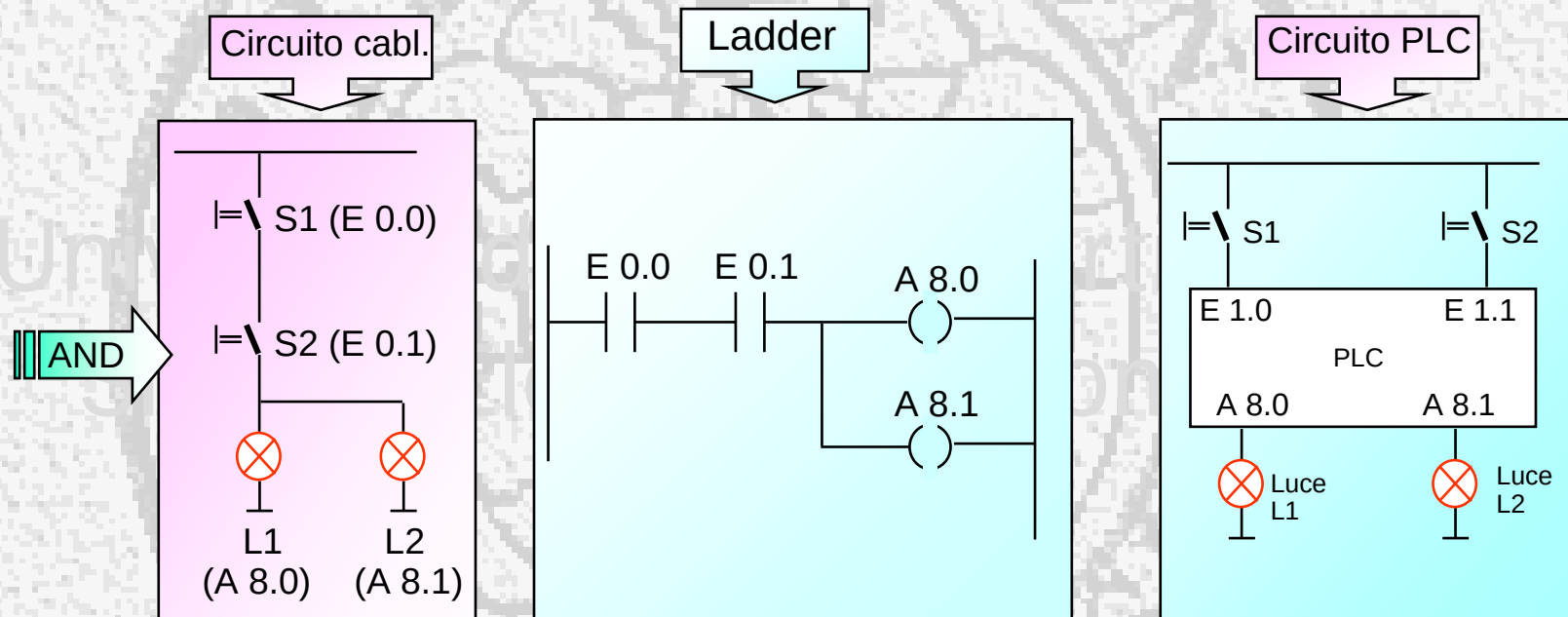


La variabile OUT associata al coil e' posta a 0 se vi e' una corrente che fluisce da sinistra. La variabile rimane a 0 per qualunque scansione del Diagramma Ladder fino a quando viene utilizzato un coil SET.

Assegnamento, Set, Reset



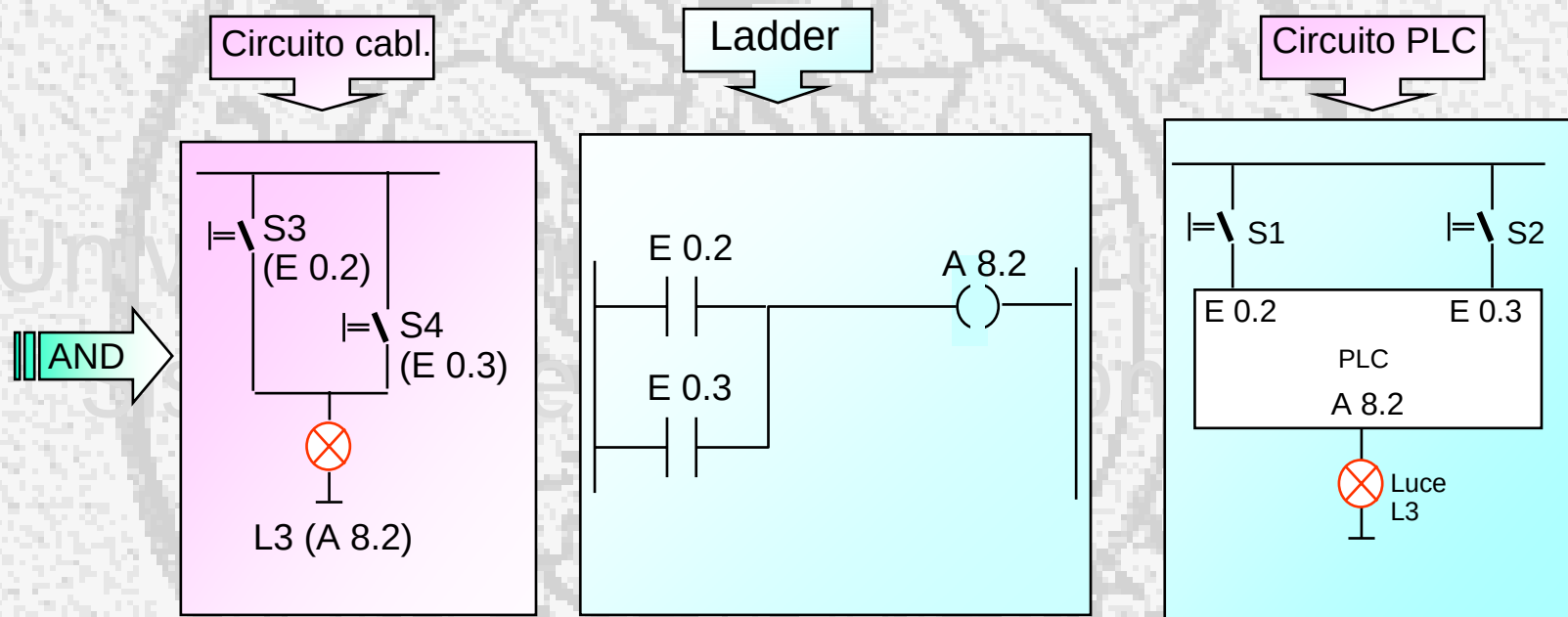
Operazione logica binaria: AND



$$(E 0.0 * E 0.1) = A 8.0$$

$$(E 0.0 * E 0.1) = A 8.1$$

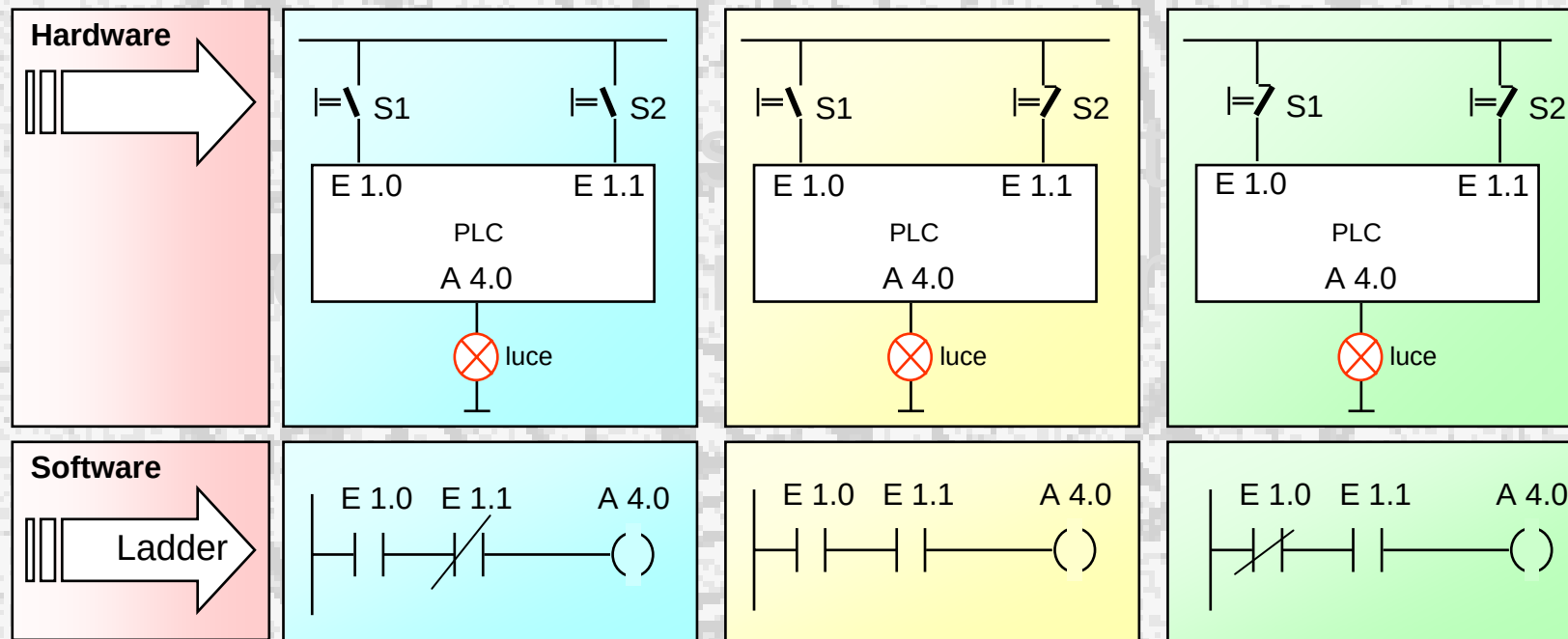
Operazione logica binaria: OR



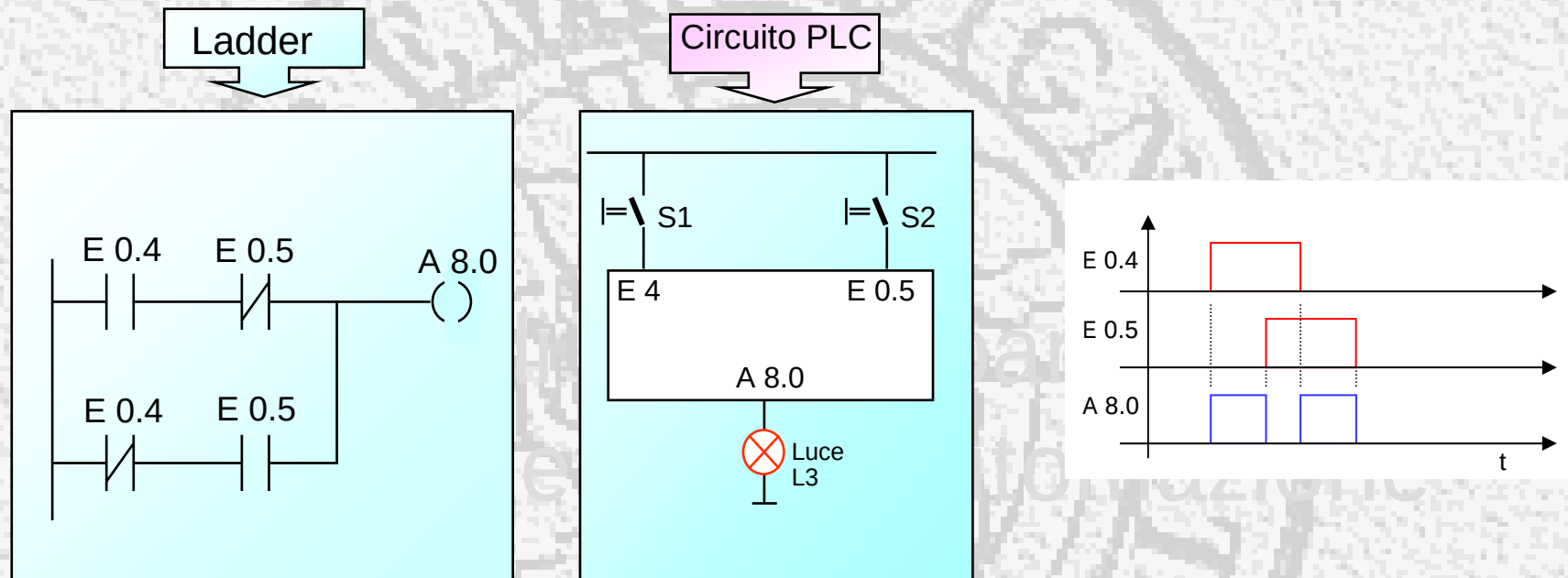
$$(E 0.0 + E 0.1) = A 8.2$$

Esercizio

In tutti e tre i casi la luce deve accendersi solo quando S1 è attivato e S2 non è attivato!



Operazioni logiche binarie: OR Esclusivo (XOR)

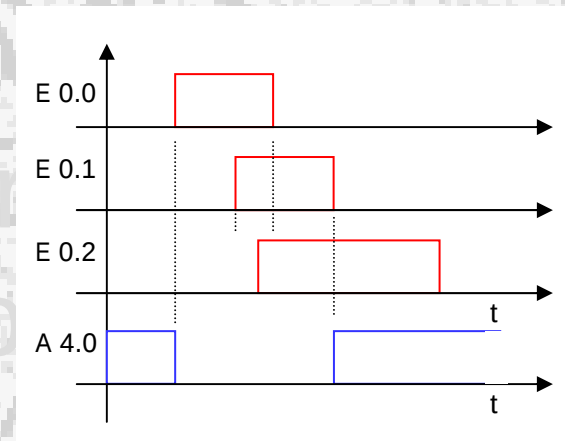
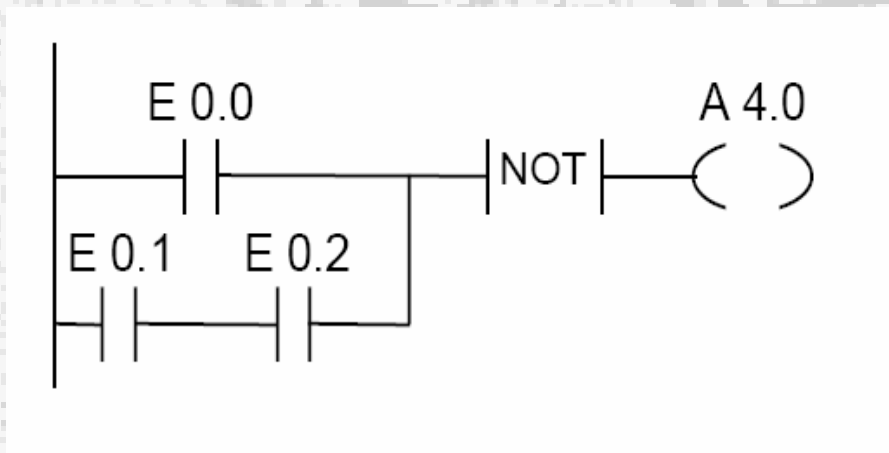


$$\overline{(E\ 0.4 + E\ 0.5)} = A\ 8.0$$

$$\overline{(E\ 0.4 + E\ 0.5)} = A\ 8.0$$

Operazioni logiche binarie: NOT

Inverte il risultato della combinazione



$$\overline{(E 0.0)} = A 4.0$$

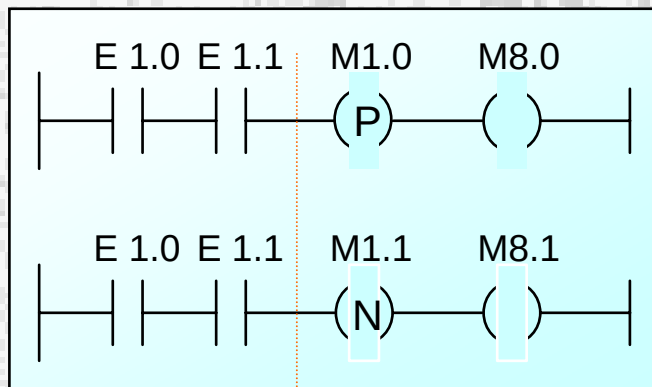
$$\overline{(E 0.1 * E 0.2)} = A 4.0$$

L'uscita A 4.0 è "0" se:

lo stato all'ingresso E 0.0 è "1" o lo stato di E 0.1. e E 0.2 è "1".

(e = operazione AND; o = operazione OR)

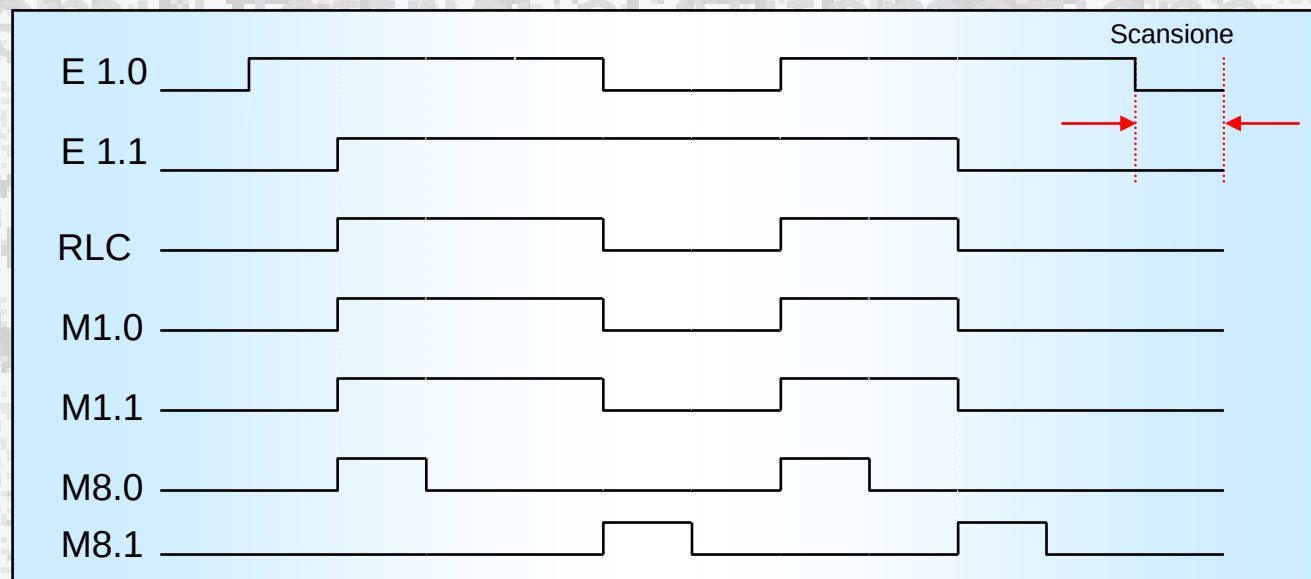
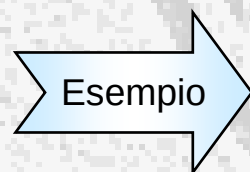
Rilevazione del fronte



M 8.0 va a 1 quando si ha un fronte di salita del RLC e vi rimane per una scansione.

M 8.1 va a 1 quando si ha un fronte di discesa del RLC e vi rimane per una scansione.

RLC = **R**isultato **L**ogico **C**ombinatorio



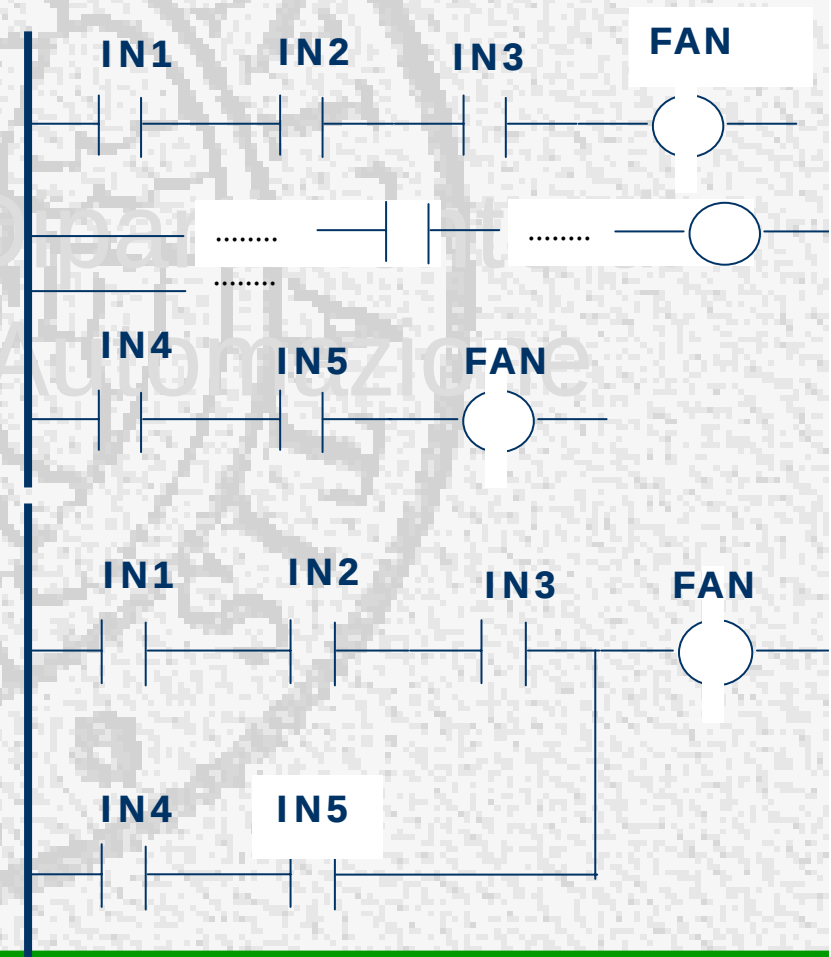
Effetti collaterali delle regole di esecuzione dei Rung

L'ordine di valutazione comporta la necessità di riunificare i rung che operano delle modifiche sulle stesse variabili o sulle stesse uscite.

Esempio:

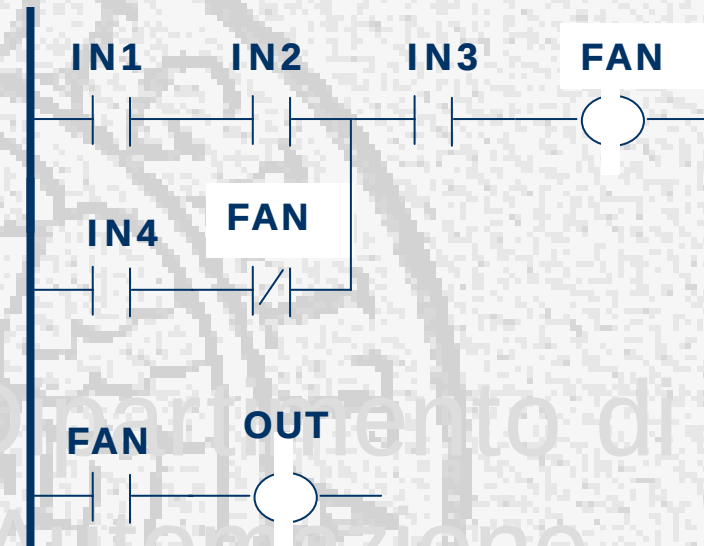
L'effetto di IN1, IN2, e IN3 sulla variabile FAN è nullo. La variabile può essere modificata solo da IN4 e IN5, a causa della posizione del rung che li contiene

Riunificando i rung, la variabile FAN verrà aggiornata solo dopo aver valutato il rung composto dagli ingressi IN1, IN2, IN3, IN4 e IN5



Feedback Paths

E' possibile che un rung presenti dei contatti e dei coil ai quali vengono associate le stesse variabili



Il valore della variabile associata al contatto FAN è quello valutato nell'ultima valutazione (rung precedente).

NOTA: L'uso dei Feedback può essere pericoloso, in quanto può portare ad una instabilità delle uscite del PLC.

Ad esempio, l'uscita FAN diviene instabile se inizialmente $FAN=0$ e nel tempo gli ingressi IN1, IN2, IN3 e IN4 si mantengono costanti e pari a $IN1=0$, $IN2=0$, $IN3=1$, $IN4=1$

Tecnica di programmazione Ladder

- Rappresentare la soluzione del problema con una macchina a stati.
- Ogni stato viene rappresentato da una variabile locale.
- Per ogni stato vengono identificate le azioni da eseguire.
- Vengono identificate gli eventi che producono il passaggio di stato. Ciascun evento dovrà corrispondere ad una variabile.

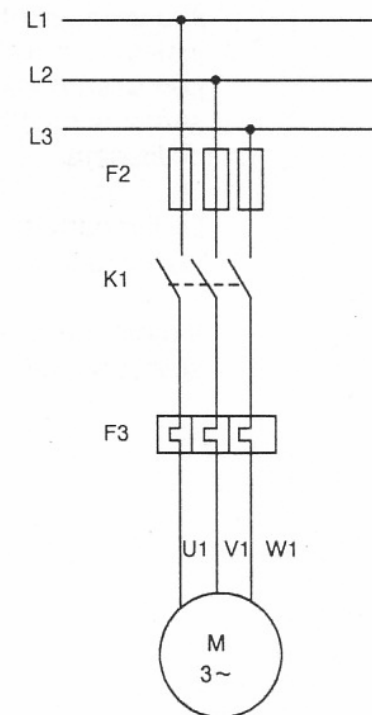
Esempio 1. Avviamento/Arresto di un motore

Descrizione del problema

Un motore asincrono viene inserito premendo il pulsante S1 (pulsante di marcia) e disinserito con il pulsante S0 (pulsante di arresto).

Le lampade di segnalazione H1 e H2 visualizzano lo stato di attività del motore.

Il motore è protetto con un interruttore termico di sovraccorrente e con fusibili per il corto circuito.



Esempio 1. Avviamento/Arresto di un motore

Si definiscono gli ingressi e si associano ai corrispondenti ingressi del PLC.

Simbolo	Variabile PLC	Descrizione
S0	E 0.1	OFF Pulsante N.A.
S1	E 0.2	ON Pulsante N.A.

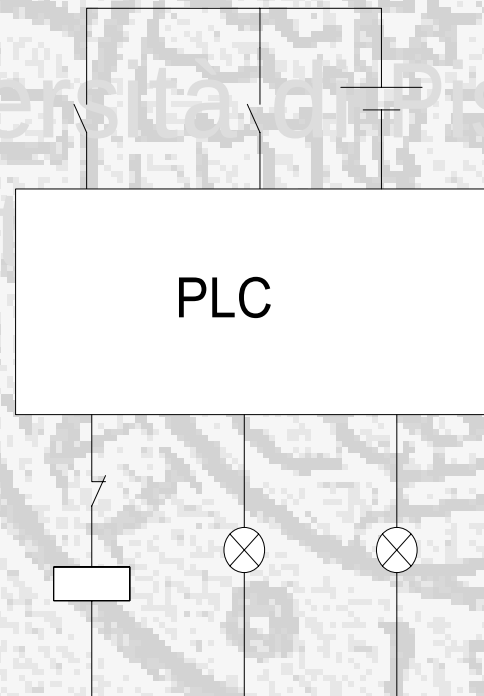
Esempio 1. Avviamento/Arresto di un motore

Si definiscono le uscite e si associano alle corrispondenti uscite del PLC.

Simbolo	Variabile PLC	Descrizione
K1	A 2.0	Bobina contattore motore
H1	A 2.1	Segnalazione motore fermo
H2	A 2.2	Segnalazione motore in marcia

Esempio 1. Avviamento/Arresto di un motore

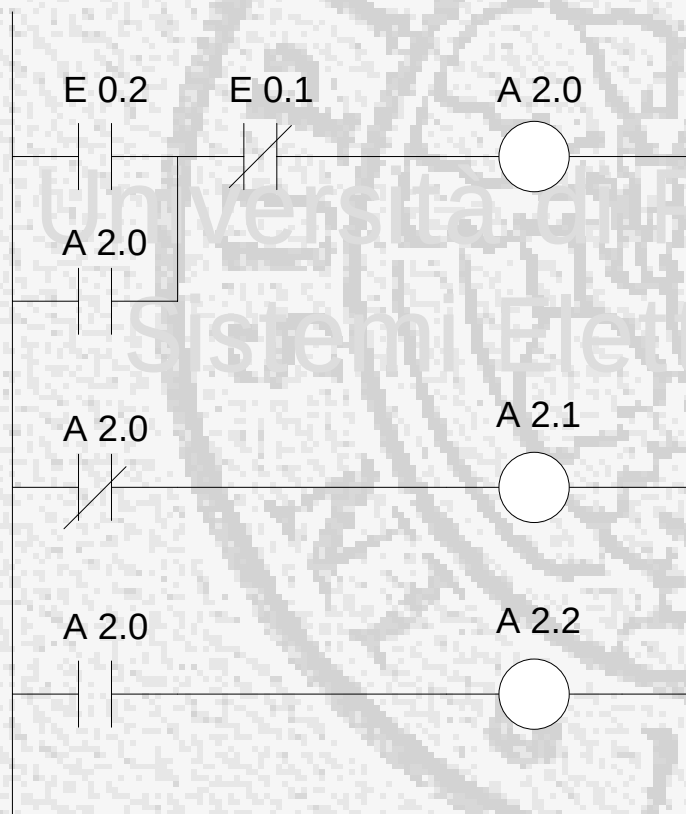
Lo schema di collegamento al PLC è il seguente:



Per motivi di sicurezza il contatto normalmente chiuso dell'interruttore di sovracorrente F3 deve essere collegato a monte della bobina del contattore K1. Se si attiva l'interruttore di sovracorrente, la bobina del contattore K1 viene disinserita direttamente via hardware senza l'intervento del PLC. L'interruttore termico di sovracorrente deve essere dotato di un dispositivo per il ripristino manuale per evitare un successivo inserimento del contattore K1 dovuto al suo ripristino automatico.

Esempio 1. Avviamento/Arresto di un motore

Il programma ladder è il seguente:



La bobina A 2.0 viene eccitata se è chiuso E 0.2 e aperto E 0.1. Deve essere prevista l'autoritenuta.

La bobina A 2.1 viene eccitata se è diseccitata la A 2.0.

La bobina A 2.2 viene eccitata se è eccitata la A 2.0.

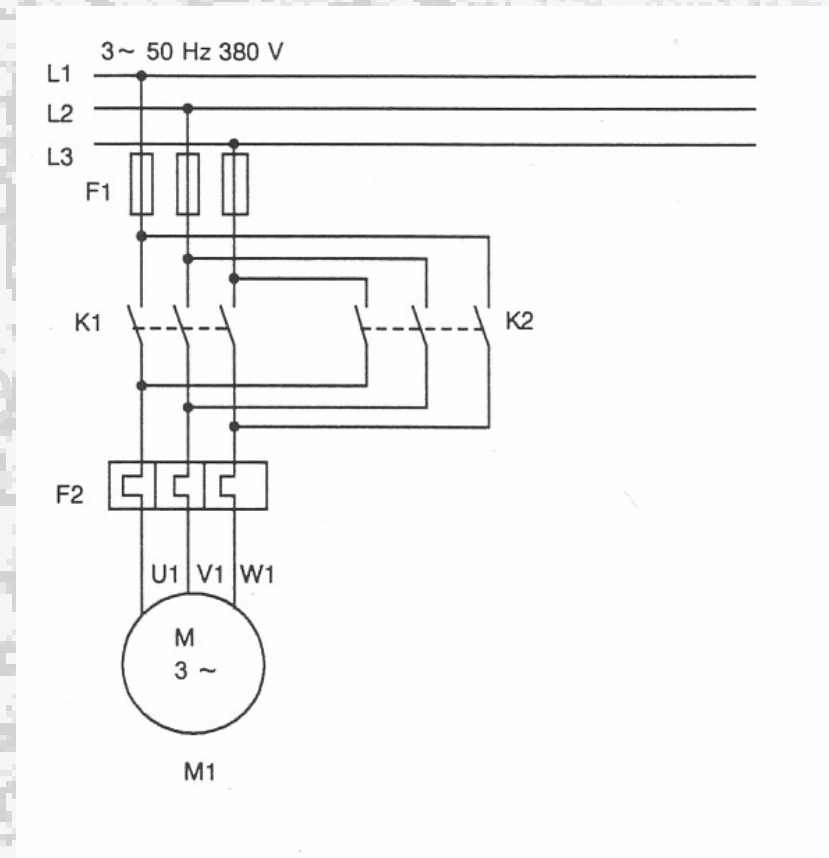
Esempio 2. Circuito invertitore per motore trifase

Descrizione del problema

Un motore asincrono viene inserito premendo il pulsante S1 per il funzionamento “marcia avanti”, S2 per “marcia indietro”, e disinserito con il pulsante S0.

Le lampade di segnalazione H1, H2 e H3 visualizzano lo stato di attività del motore.

Il motore è protetto con un interruttore termico di sovraccorrente e con fusibili per il corto circuito.



Esempio 2. Circuito invertitore per motore trifase

Si definiscono gli ingressi e si associano ai corrispondenti ingressi del PLC.

Simbolo	Variabile PLC	Descrizione
S0	E 0.1	OFF Pulsante N.A.
S1	E 0.2	Pulsante Marcia avanti N.A.
S2	E 0.3	Pulsante Marcia indietro N.A.

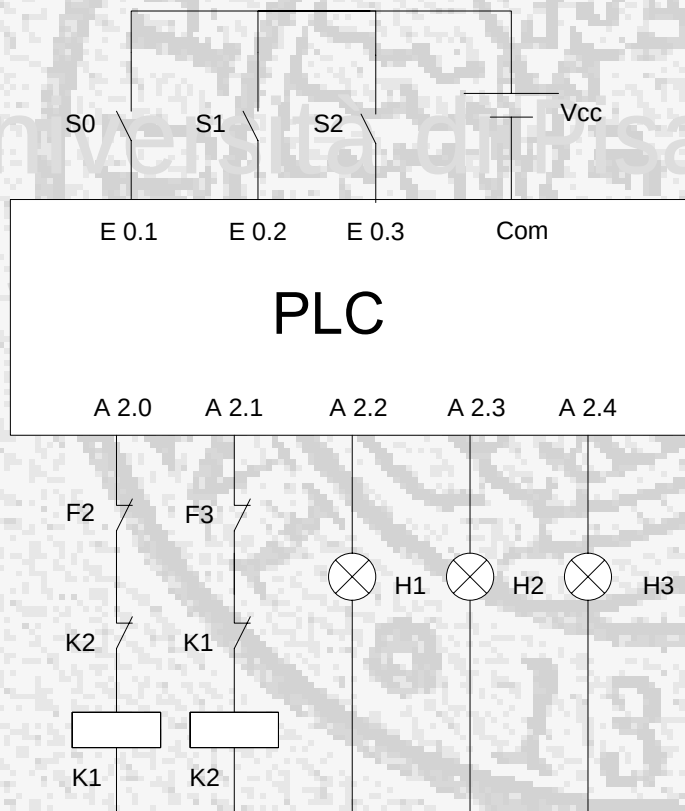
Esempio 2. Circuito invertitore per motore trifase

Si definiscono le uscite e si associano alle corrispondenti uscite del PLC.

Simbolo	Variabile PLC	Descrizione
K1	A 2.0	Bobina contattore funz. 'marcia avanti'
K2	A 2.1	Bobina contattore funz. 'marcia indietro'
H1	A 2.2	Segnalazione motore fermo
H2	A 2.3	Segnalazione motore in 'marcia avanti'
H3	A 2.4	Segnalazione motore in 'marcia indietro'

Esempio 2. Circuito invertitore per motore trifase

Lo schema di collegamento al PLC è il seguente:

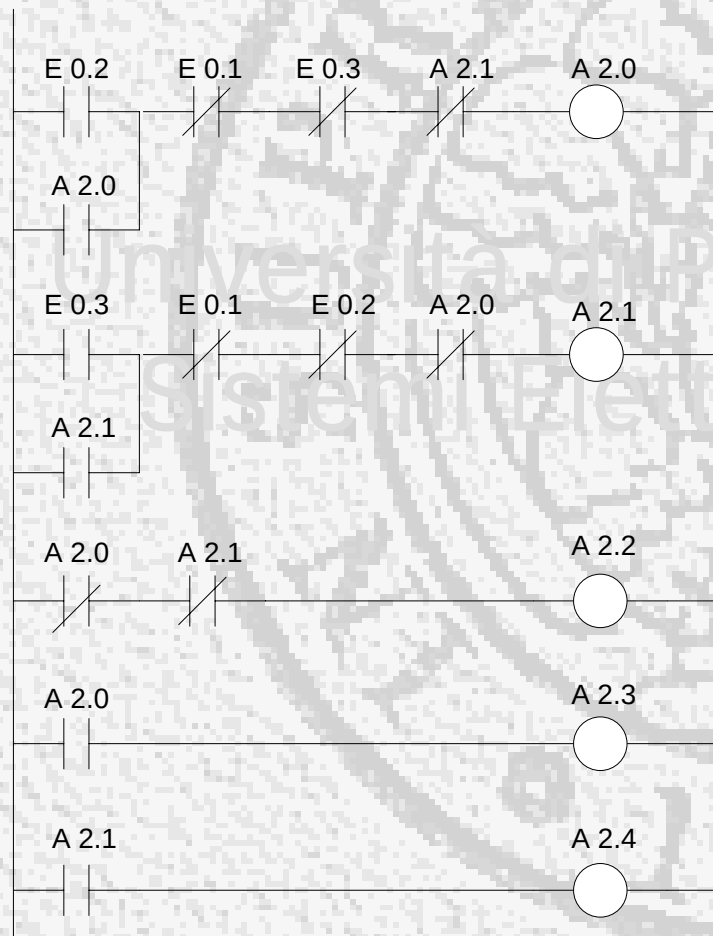


I contatti dell'interruttore termico devono essere collegati sulle linee di alimentazione dei due contattori.

Inoltre un contattore non deve poter essere inserito fintanto che l'altro non è completamente disinserito. La concomitanza di entrambi i comandi di eccitazione potrebbe verificarsi a causa delle velocissime reazioni di comando elettronico rispetto alle reazioni delle parti meccaniche in movimento.

Esempio 2. Circuito invertitore per motore trifase

Il programma ladder è il seguente:



La bobina A 2.0 (Marcia avanti) viene eccitata se è chiuso E 0.2, è aperto E 0.1, se non è premuto E 0.3 e se il motore non è già in marcia indietro. Deve essere prevista l'autoritenuta.

La bobina A 2.1 (Marcia indietro) viene eccitata se è chiuso E 0.3, è aperto E 0.1, se non è premuto E 0.2 e se il motore non è già in marcia avanti. Deve essere prevista l'autoritenuta.

La bobina A 2.2 se sono diseccitate la A 2.0. e la A 2.1.

La bobina A 2.3 viene eccitata se è eccitata la A 2.0.

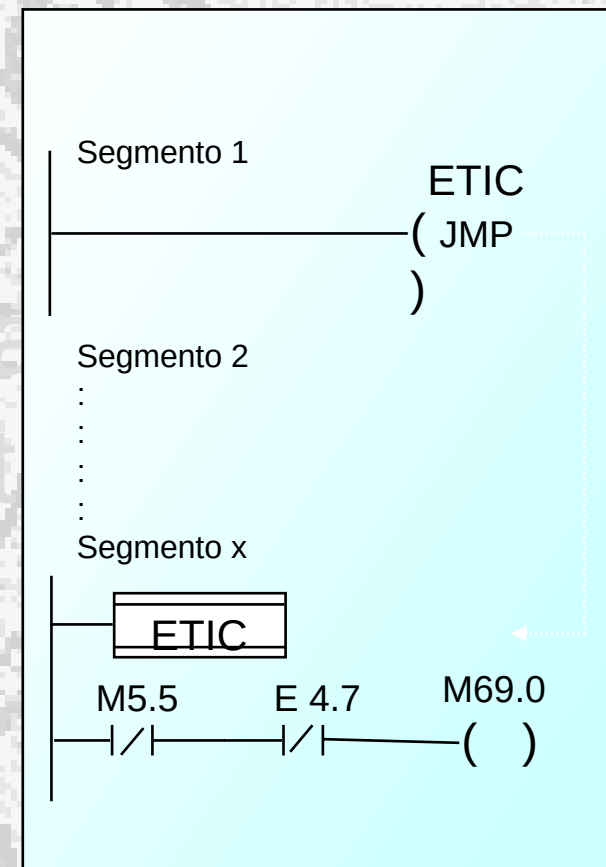
La bobina A 2.4 viene eccitata se è eccitata la A 2.1.

Salto Incondizionato (Indipendente dall'RLC)

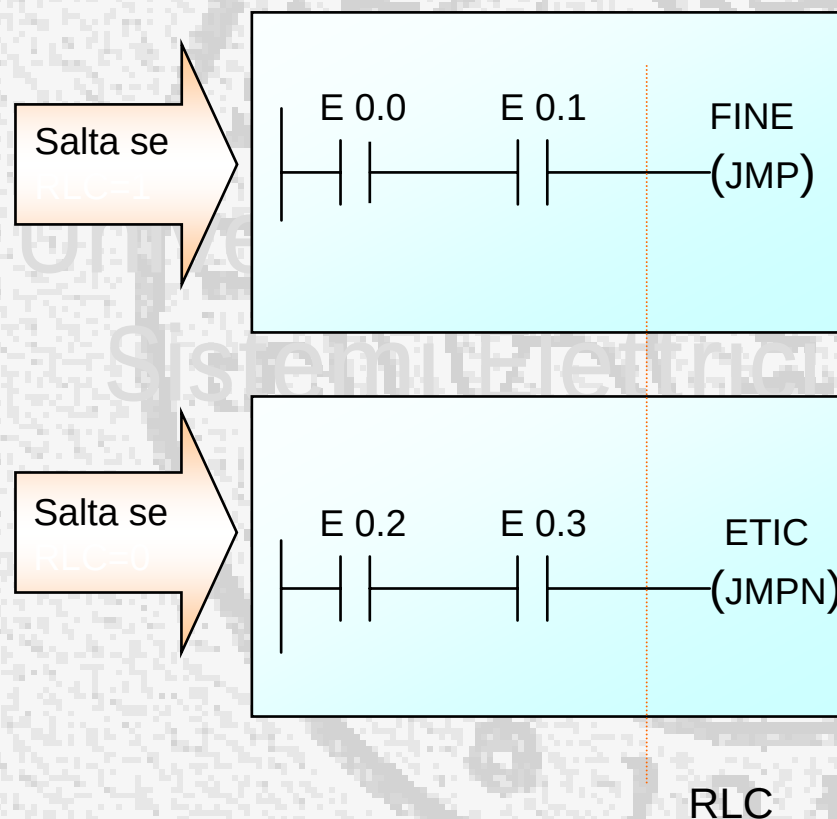
Funziona come un salto assoluto se nessun ulteriore elemento è interposto tra il percorso di corrente a sinistra e l'operazione (vedi esempio).

Per ogni ---(JMP) deve essere presente anche una destinazione (etichetta di salto).

Le operazioni tra l'operazione di salto e l'etichetta non vengono eseguite.



Salto condizionato (Dipendente dall'RLC)



Se $RLC = 1$ il programma salta al Rung con etichetta 'FINE'. Le operazioni tra l'operazione di salto e l'etichetta non vengono eseguite.

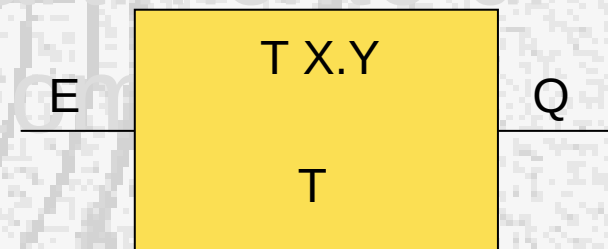
Se $RLC = 0$ il programma salta al Rung con etichetta 'ETIC'. Le operazioni tra l'operazione di salto e l'etichetta non vengono eseguite.

Temporizzatori (Timer)

E' un elemento con un'uscita binaria Q (1 bit) il cui valore cambia con un certo tempo di ritardo T (specificato in una variabile interna) rispetto al fronte di salita di una opportuna variabile di ingresso (E). Un temporizzatore è identificato da un nome composto dalla lettera T e da una coppia di numeri.

Si possono avere

- Temp. ad impulsob (SI)
- Temp. a ritardo all'inserzione (SE)
- Temp. con ritardo all'inserzione con memoria (SS)
- Temp. con ritardo alla disinserzione (SA)



Nei rung successivi ci possono essere dei contatti (NC o NA) associati alla variabile di uscita del contatore e sono identificati con il nome del timer stesso.

Temporizzatori (Timer)

Il valore del tempo di ritardo viene specificato con la seguente sintassi:

S5T#aH_bM_cS_dMS

- laddove H = ore, M = minuti, S = secondi, MS = millisecondi;
a, b, c, d vengono definiti dall'utente

- La base di tempo viene selezionata automaticamente e il valore viene arrotondato al numero immediatamente inferiore rispetto ad essa.

Il valore di tempo massimo che si può immettere è 9.990 secondi, o 2H_46M_30S.

Esempi:

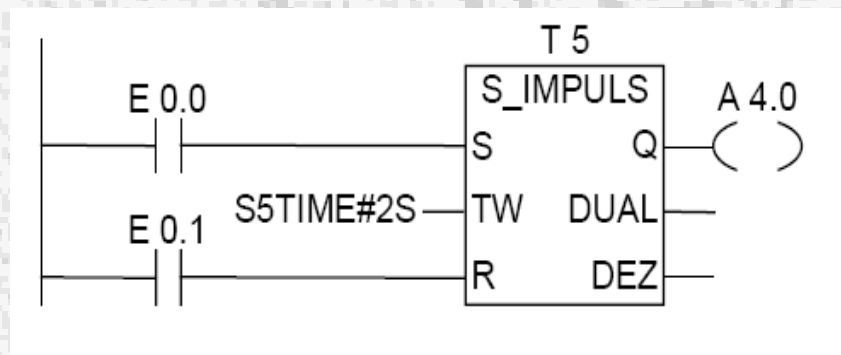
S5TIME#4S --> 4 secondi

s5t#2h_15m --> 2 ore e 15 minuti

S5T#1H_12M_18S --> 1 ora, 12 minuti e 18 secondi

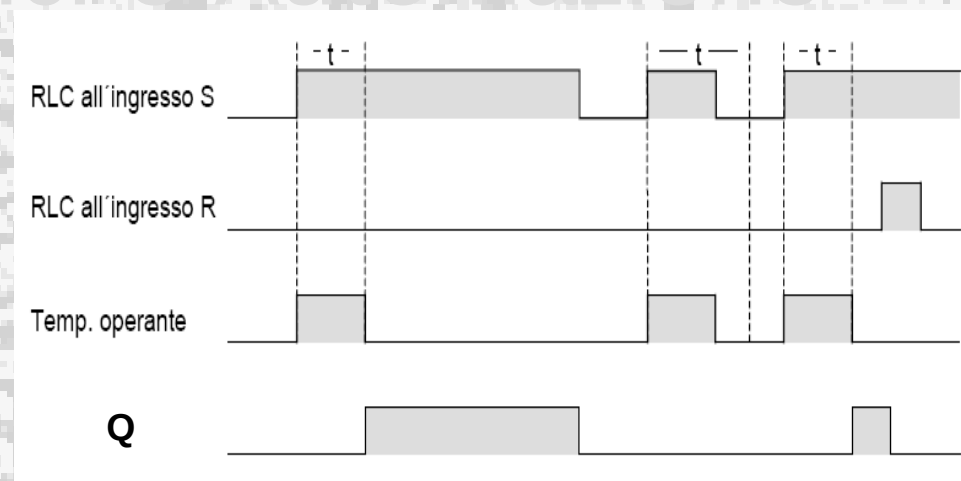
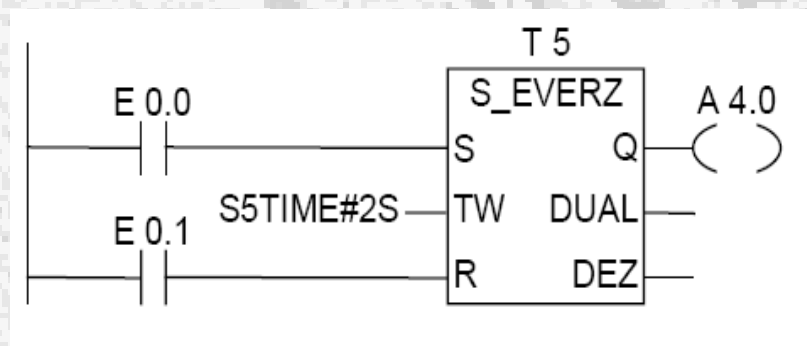
S_IMPULS Parametrizza e avvia bobina temporizzatore come impulso (SI)

Se lo stato di segnale dell'ingresso E 0.0 cambia da "0" a "1" (ovvero se c'è un fronte di salita nel RLC), il temporizzatore T5 viene avviato. Il temporizzatore continua ad essere attivo con il tempo specificato di due secondi (2 s) finché l'ingresso E 0.0 = 1. Se lo stato di segnale di E 0.0 cambia da "1" a "0" prima che il tempo sia trascorso, il temporizzatore viene arrestato. Se lo stato di segnale di E 0.0 cambia da "1" a "0" prima che il tempo sia trascorso, il temporizzatore viene resettato. Se lo stato di segnale di E 0.1 cambia da "0" a "1" mentre il temporizzatore è attivo, il temporizzatore viene resettato. L'uscita A 4.0 è "1" finché il temporizzatore è attivo, e "0" se il tempo è trascorso od è stato resettato.



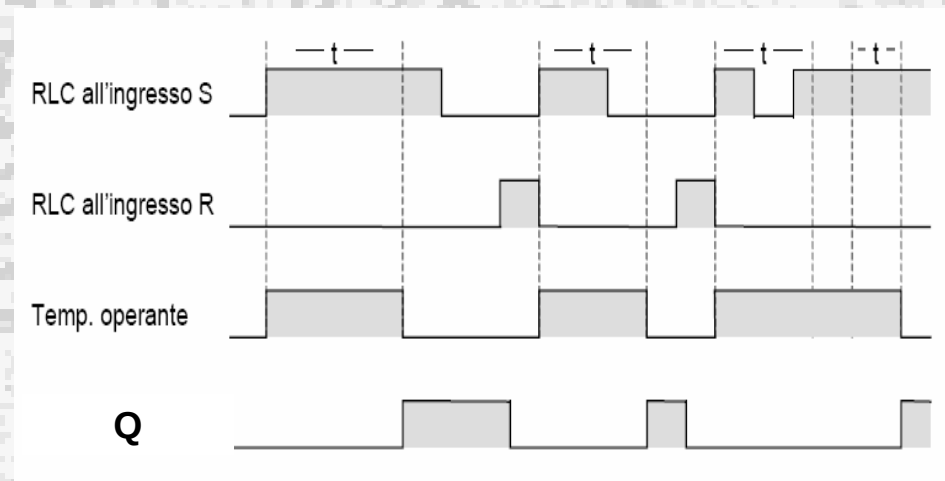
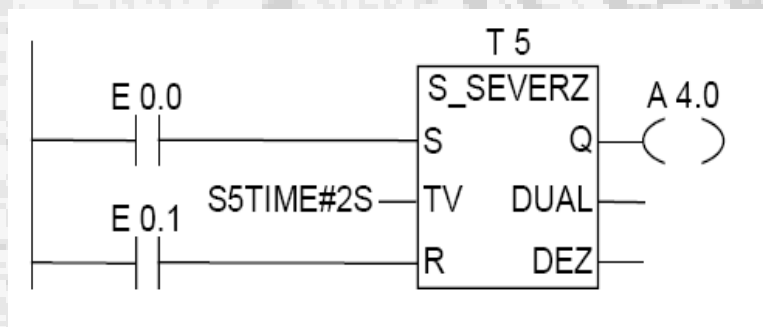
S_EVERZ Parametrizza e avvia bobina temporizzatore come ritardo all'inserzione (SE)

Se lo stato di segnale dell'ingresso E 0.0 cambia da "0" a "1" (ovvero se c'è un fronte disalita nel RLC), il temporizzatore T5 viene avviato. Se il tempo specificato (2 s in questo caso) è trascorso e lo stato di segnale all'ingresso E 0.0 è sempre "1", l'uscita A 4.0 è "1". Se lo stato di segnale di E 0.0 cambia da "1" a "0", il temporizzatore viene arrestato e l'uscita A 4.0 è "0". (Se lo stato di segnale dell'ingresso E 0.1 cambia da "0" a "1", il temporizzatore viene resettato, a prescindere dal fatto se il temporizzatore sia o no attivo).



S_SEVERZ Parametrizza e avvia bobina temporizzatore come ritardo all'inserzione con memoria (SS)

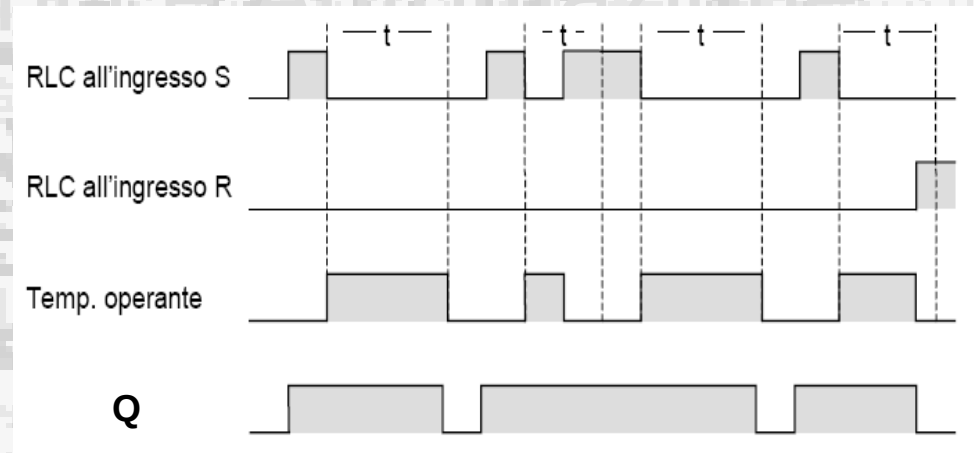
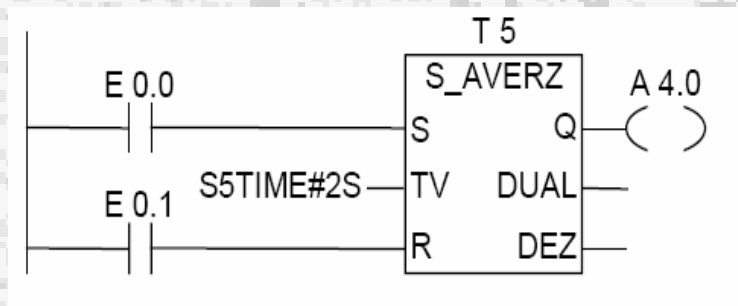
Se lo stato di segnale dell'ingresso E 0.0 cambia da "0" a "1" (ovvero se c'è un fronte di salita nel RLC), il temporizzatore T5 viene avviato. Il temporizzatore continua ad essere attivo senza considerazione del cambiamento dello stato di segnale dell'ingresso E 0.0 da "1" a "0". Se lo stato di segnale di E 0.0 cambia da "0" a "1" prima che il tempo sia trascorso, il temporizzatore viene avviato nuovamente. L'uscita A 4.0 è "1" se il tempo è trascorso. (Se lo stato di segnale dell'ingresso E 0.1 cambia da "0" a "1", il temporizzatore viene resettato, a prescindere dal RLC in S).



S_AVERZ Parametrizza e avvia bobina temporizzatore come ritardo alla disinserzione (SA)

Se lo stato di segnale dell'ingresso E 0.0 cambia da "1" a "0", il temporizzatore viene avviato.

A 4.0 è "1" se lo stato di segnale di E 0.0 è "1", oppure se il temporizzatore è operante.
(Se lo stato di segnale dell'ingresso E 0.1 cambia da "0" a "1", mentre il tempo è attivo, il temporizzatore viene resettato).

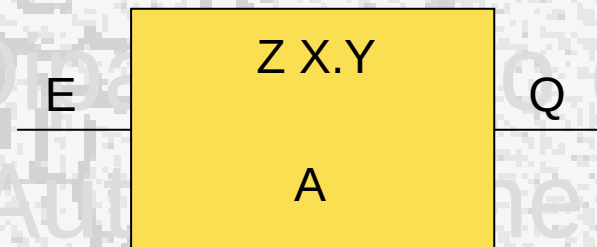


Contatore

E' un blocco funzionale che segnala il raggiungimento di un valore di conteggio.
E' un elemento con un'uscita binaria Q (1 bit) il cui valore dipende dal numero di volte (valore specificato) che una opportuna variabile di ingresso (E) passa dallo stato zero allo stato 1. Un contatore è identificato da un nome composto dalla lettera Z e da una coppia di numeri.

Si possono avere

- Contatori in avanti
- Contatori all'indietro
- Contatori in avanti e all'indietro

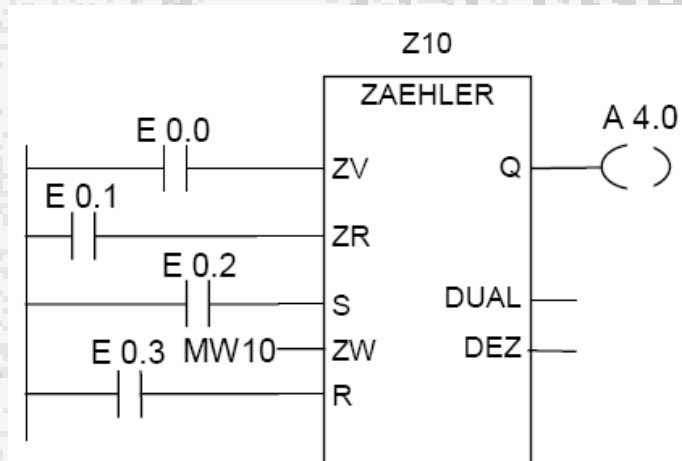


Ad ogni fronte di salita di E la variabile interna (A) del contatore viene incrementata o decrementata di un'unità a seconda del tipo di contatore. Il range di tale variabile è compresa tra 0 e 999.

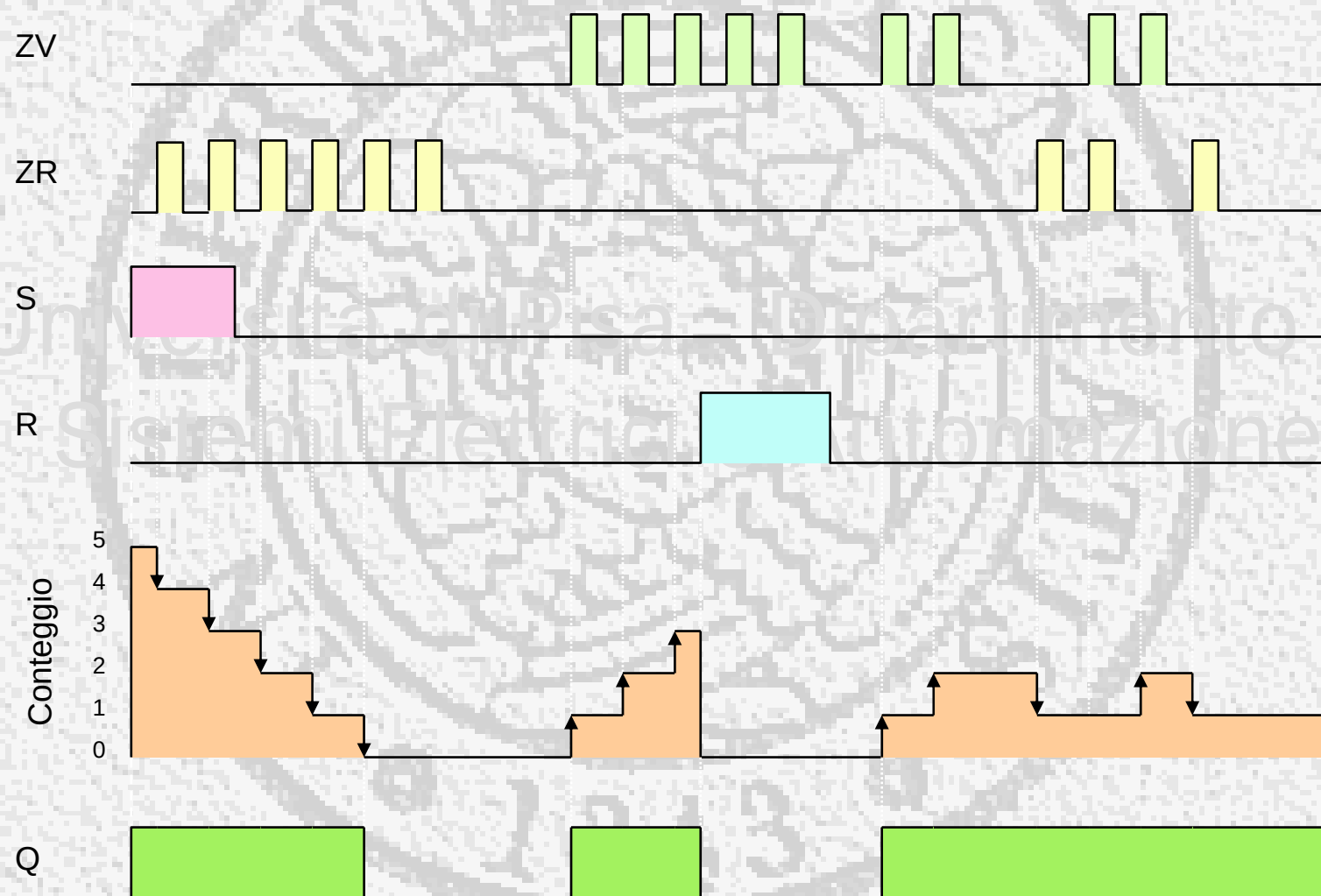
Nei rung successivi ci possono essere dei contatti (NC o NA) associati alla variabile di uscita del contatore e sono identificati con il nome del contatore.

Contatore Parametrizza e conta in avanti/all'indietro

Con un cambiamento dello stato di segnale da "0" a "1" all'ingresso E 0.2, il contatore viene preimpostato con il valore di MW10. Se lo stato di segnale di E 0.0 cambia da "0" a "1", il valore del contatore Z10 viene incrementato di "1", eccetto quando il valore di Z10 è uguale a "999". Se E 0.1 cambia da "0" a "1", Z10 viene decrementato di "1", eccetto quando il valore di Z10 è uguale a zero. A 4.0 è "1", se Z10 è diverso da zero.

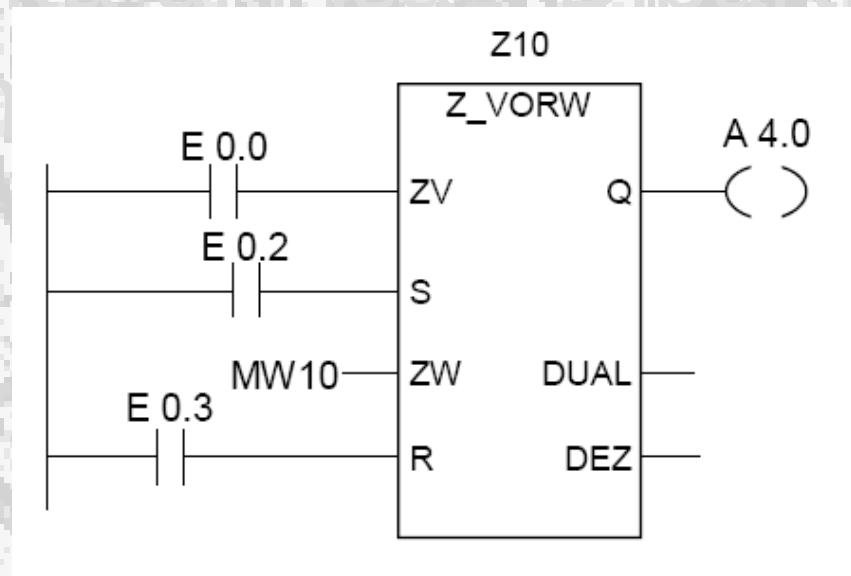


Contatori: Diagrammi Funzionali



Contatore Parametrizza e conta in avanti

Con un cambiamento dello stato di segnale da "0" a "1" all'ingresso E 0.2, il contatore viene preimpostato con il valore di MW10. Se lo stato di segnale di E 0.0 cambia da "0" a "1", il valore del contatore Z10 viene incrementato di "1", eccetto quando il valore di Z10 è uguale a "999". A 4.0 è "1", se Z10 è diverso da zero.



Contatore Parametrizza e conta all'indietro

Un cambiamento dello stato di segnale da "0" a "1" all'ingresso E 0.2 setta il contatore con il valore di MW10. Se lo stato di segnale di E 0.0 cambia da "0" a "1", il valore del contatore Z10 viene decrementato di "1", eccetto quando il valore di Z10 è uguale a "0". A 4.0 è "1", se Z10 è diverso da zero.

