

Introduzione

“Ladder Diagram” =

- letteralmente “diagrammi a scala”, perchè la disposizione grafica dei suoi simboli ricorda una scala
- in italiano è più usato il termine “linguaggio a contatti” o “diagramma a relè”, poichè gli elementi di base del linguaggio ricordano contatti elettrici
- nel seguito, lo indicheremo con il simbolo LD

Caratteristiche principali

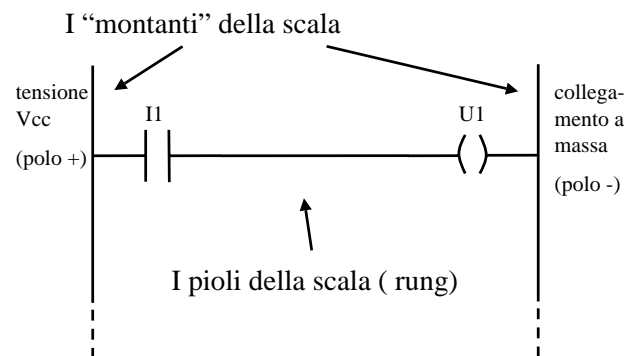
- è un linguaggio grafico
- si basa sulla trasposizione, in logica di programmazione, del funzionamento di una rete elettrica molto semplice, in cui l’obiettivo è alimentare opportuni utilizzatori elettrici (bobine) tramite interruttori (chiamati anche contatti o relè).
- il motivo è storico: le logiche di controllo prima del PLC venivano fatte direttamente con circuiti elettrici.
- ci sono convenzioni sia su come disporre gli elementi grafici sia sui nomi

Area Dati e convenzioni tipiche

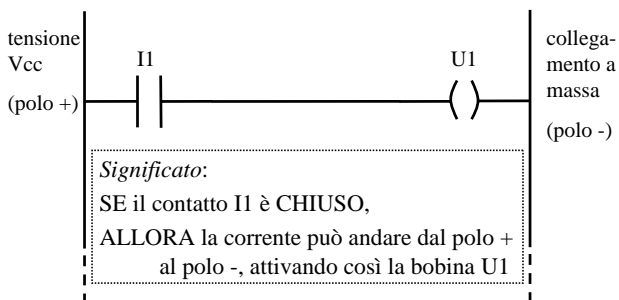
- Area degli Ingressi
 - $I_x:y$ (Il bit y della word x)
- Area delle Uscite
 - $U_x:y$
- Area dei Temporizzatori (timer)
 - $T_1..T_n$
- Area dei Contatori
 - $C_1..C_n$
- Area PID, e Area Utente

NOTA: le convenzioni cambiano da costruttore a costruttore; noi useremo nomi simbolici più semplici

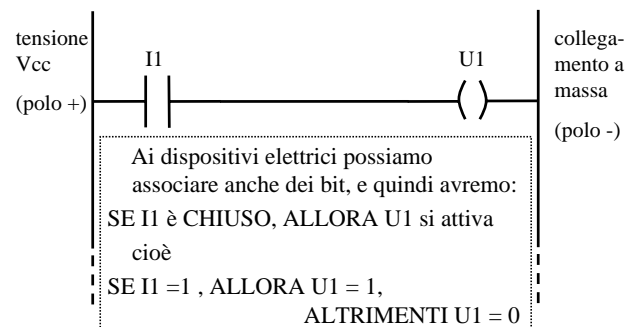
Elementi di Base



Elementi di Base



Elementi di Base

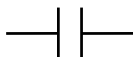


Elementi di base per gli ingressi

Convenzionalmente, gli ingressi si mettono sulla sinistra, mentre le uscite sulla destra del grafo

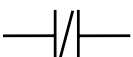
Contatto normalmente aperto

- Se il bit associato vale 1, il contatto si chiude
- Se il bit associato vale 0, il contatto si apre



Contatto normalmente chiuso

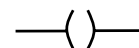
- Se il bit associato vale 1, il contatto si apre
- Se il bit associato vale 0, il contatto si chiude



Elementi di base per le uscite

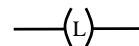
Bobina

Si attiva quando passa corrente. Quindi, il bit ad essa associata sale al valore logico 1 (ON) se le condizioni logiche alla sua sinistra sono verificate



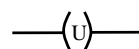
Latch bobina

Mantiene lo stato logico 1 (ON) anche quando le condizioni di attivazione vengono a mancare (simile al SET di un flip-flop)



Unlatch bobina

Riporta allo stato logico 0 (OFF) un'uscita (simile al SET di un flip-flop)



Costruiamo un programma

Disponendo uno dopo l'altro i pioli (cioè le istruzioni), siamo così in grado di costruire un programma.

Attenzione:

il LD è un linguaggio, non una rete elettrica, quindi occorre specificare come vengono interpretati i pioli. Dovremo quindi specificare:

- come vengono scanditi i pioli
- quando vengono letti gli ingressi e aggiornate le uscite

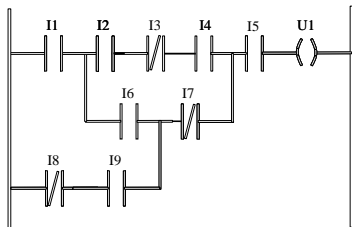
Regole di scansione dei pioli

I pioli vengono scanditi:

- dall'alto verso il basso, e
- da sinistra verso destra.

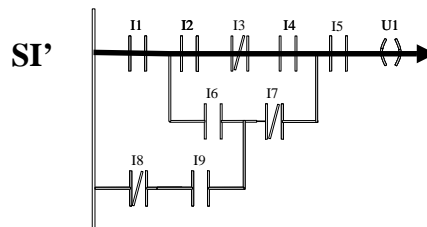
Quindi, nel singolo piolo, il flusso di energia nei vari dispositivi va solo da sinistra a destra, senza possibilità che questo si inverta.

Regole di scansione dei pioli



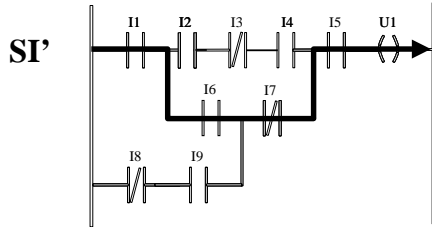
- Flusso di energia solo da sinistra a destra senza possibilità di inversione

Regole di scansione dei pioli



- Flusso di energia solo da sinistra a destra senza possibilità di inversione

Regole di scansione dei pioli



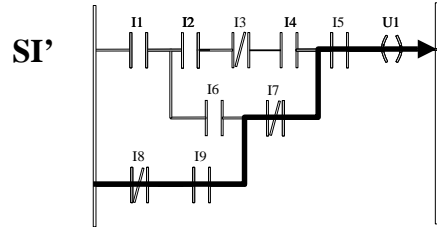
- Flusso di energia solo da sinistra a destra senza possibilità di inversione

Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

1:

Regole di scansione dei pioli



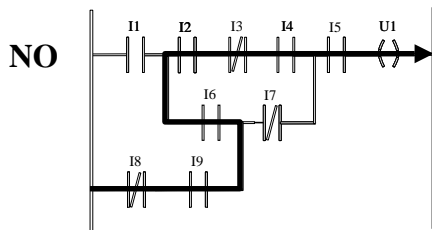
- Flusso di energia solo da sinistra a destra senza possibilità di inversione

Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

1:

Regole di scansione dei pioli



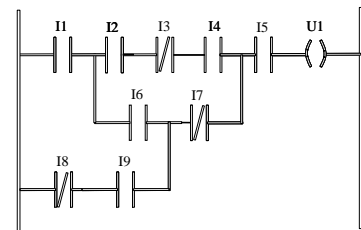
- Flusso di energia solo da sinistra a destra senza possibilità di inversione

Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

1:

Regole di scansione dei pioli



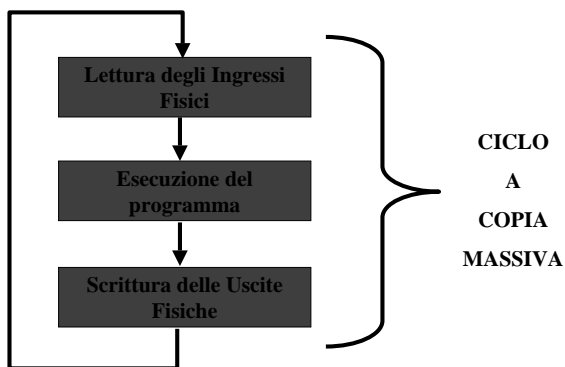
U1= I5 AND [(I4 AND NOT(I3) AND I2 AND I1)
OR (NOT(I7) AND
(I6 AND I1 OR (NOT(I8) AND I9)))]

Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

1:

Sincronizzazione con I/O



Automazione Industriale: Prof. Luca Ferrarini

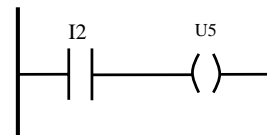
LD: NON distribuire, grazie!

1:

Sincronizzazione con I/O

Nota:

Ogni piolo viene scandito in ogni ciclo (a meno delle istruzioni di salto): pertanto le uscite associate alle bobine normali (senza latch o unlatch) vengono scritte ad ogni ciclo!



Quindi con questo LD ad ogni ciclo scrivo 0 o 1 in U5! Inoltre, il suo valore permane fino alla prossima esecuzione (al ciclo successivo) della stessa istruzione.

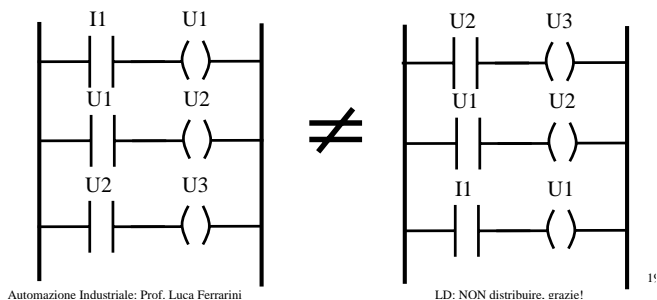
Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

1:

Sincronizzazione con I/O

Inoltre, il valore delle variabili lette in ingresso **rimane costante per tutto il ciclo** del programma quindi i seguenti due programmi sono diversi:



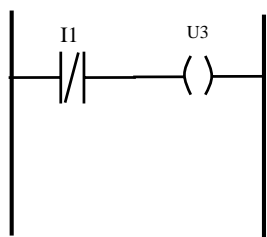
Esempi

Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

20

Programmazione di una NOT



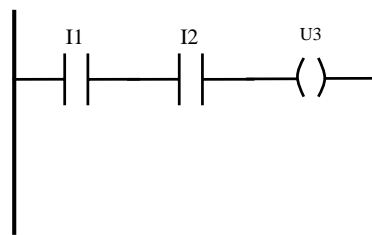
$U3 = \text{not } I1$

Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

21

Programmazione di una AND



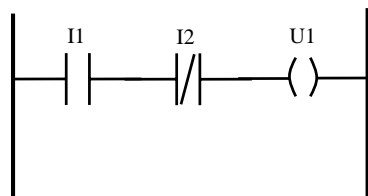
$U3 = I1 \text{ and } I2$

Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

22

Esempio di Rung



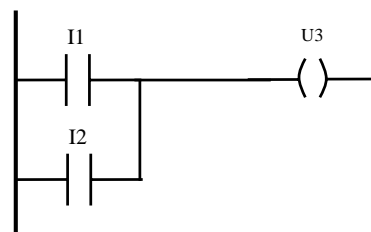
$U1 = I1 \text{ and } (\text{not}(I2))$

Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

23

Programmazione di una OR



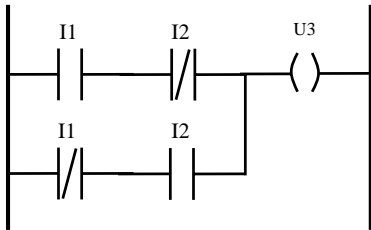
$U3 = I1 \text{ or } I2$

Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

24

Programmazione di una EX-OR



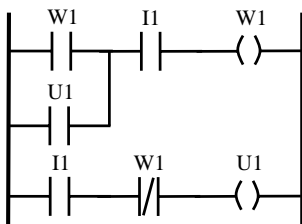
I1	I2	U3
0	0	0
0	1	1
1	0	1
1	1	0

$$U3 = I1 \text{ ex-or } I2 = [I1 \text{ and } (\text{not} I2)] \text{ or } [(\text{not} I1) \text{ and } I2]$$

Elementi dinamici in LD

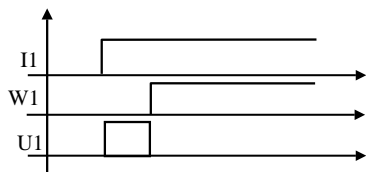
- Le istruzioni di base del LD sono “logiche”: assegnano un’uscita in base ad una pura combinazione logica delle variabili associate ai contatti.
- Tuttavia, è possibile in LD associare ad un contatto anche una variabile di uscita.
- Tale possibilità, unita all’esecuzione sequenziale e ciclica di uno schema LD, rende possibile la creazione di elementi dinamici, o con memoria, e l’identificazione di variazioni di una variabile.
- Gli esempi seguenti trascurano il tempo di ciclo.

Elementi di memoria

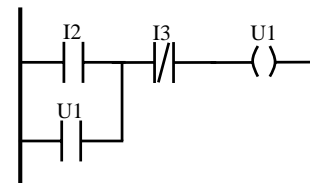


Monostabile

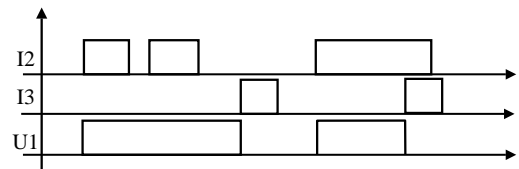
(genera un impulso di durata un tempo di ciclo dopo un fronte di salita su I1)



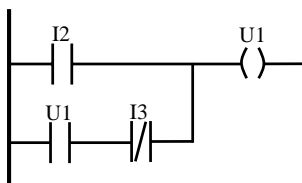
Elementi di memoria



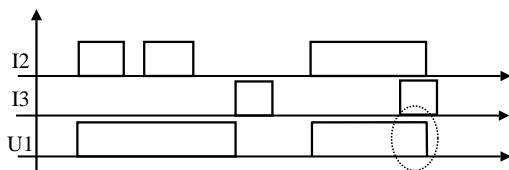
Flip-flop a reset vincente



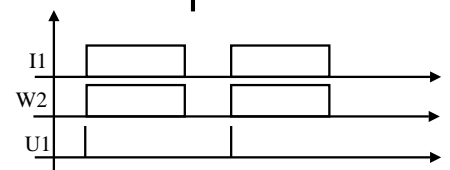
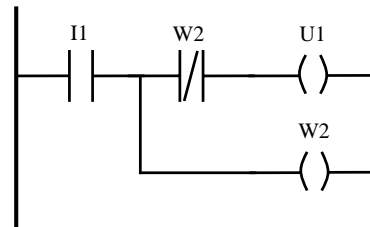
Elementi di memoria



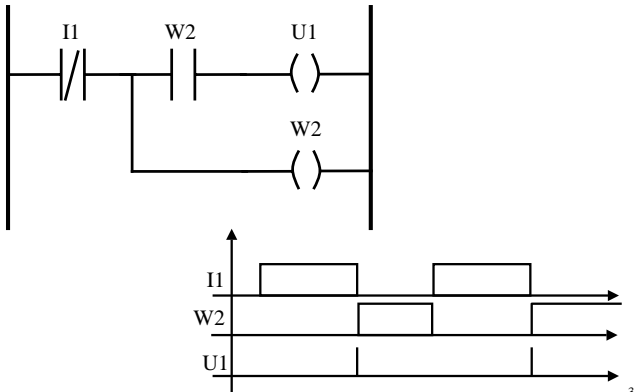
Flip-flop a set vincente



Riconoscimento di fronte di salita



Riconoscimento di fronte di discesa

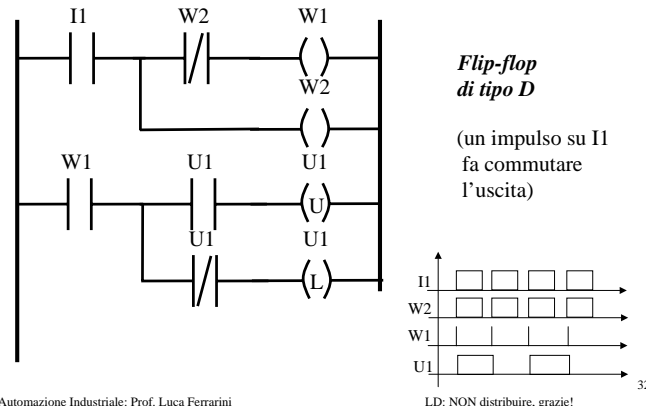


Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

3:

Elementi di memoria

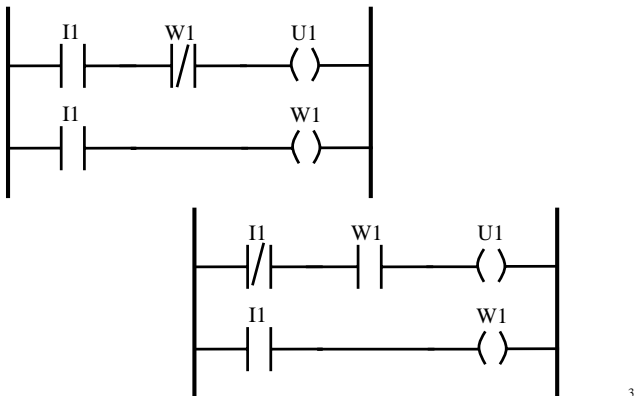


Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

3:

Riconoscimento di fronti: varianti



Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

3:

Riconoscimento di fronti: varianti

- Alcuni ambienti CAD di programmazione di PLC in LD prevedono la presenza di istruzioni avanzate.
- Ad esempio, un'attributo al contatto identifica che tale contatto si chiude sul fronte di salita (**P**, dall'inglese *Positive edge*) o sul fronte di discesa (**N**, dall'inglese *Negative edge*) della variabile associata.



Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

3:

Controllo del Programma

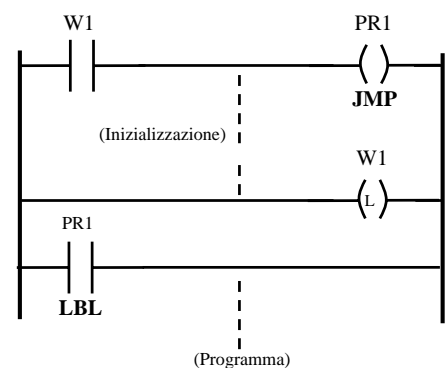
- Etichetta --|LBL|--
 - Si usa per effettuare dei "salti" di programma
- Salto --(JMP)--
 - Se il piolo è abilitato il programma passa al piolo con l'etichetta indicata
- Salto a sottoprogramma --(JSR)--
- Inizio sottoprogramma --|SBR|--
- Ritorno da sottoprogramma --(RET)--
- Master Control Relay --(MCR)-- ... --(MCR)--
 - I pioli della zona delimitata non vengono eseguiti e le bobine di uscita vengono resettate se il piolo connesso al simbolo (MCR) non è abilitato, altrimenti si eseguono tutte.

Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

3:

Esempio di utilizzo del salto

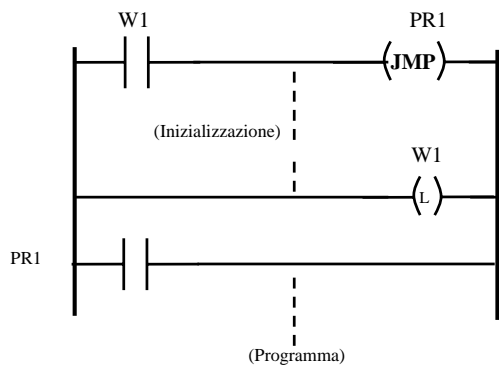


Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

3:

Esempio di utilizzo del salto

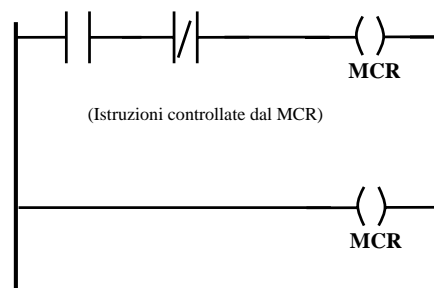


Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

3'

Esempio di utilizzo del Master Control Relay



Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

3'

Funzioni e blocchi funzione

- Il LD è infine dotato di notevoli altre estensioni, rispetto ad una pura rete elettrica.
- esse si ispirano alle istruzioni di un comune linguaggio di programmazione di alto livello, come il C, il Pascal, ecc.
- tali “istruzioni complesse” sono inglobate graficamente in LD in blocchi che vanno collegati con un piolo
- il loro uso è immediato

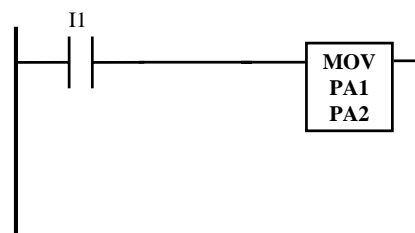
Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

3'

Istruzione MOV

- Trasferimento di memoria
 - Il contenuto di una word PA1 è trasferito in un'altra word PA2



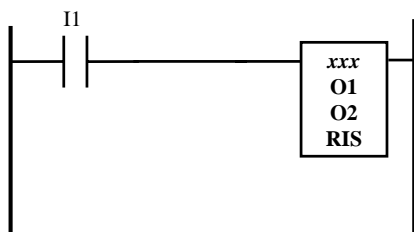
Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

4'

Operazioni Aritmetico/Logiche

- Operazioni aritmetico/logiche a due operandi (O1 e O2; RIS è il risultato)
 - ADD
 - MUL
 - SUB
 - DIV
 - AND
 - OR



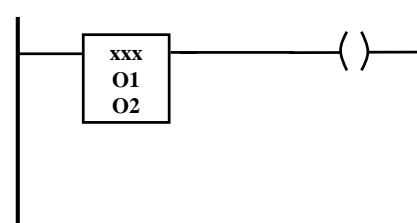
Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

4'

Operazioni Aritmetico/Logiche

- Le istruzioni di comparazione fanno parte delle condizioni di attivazione dei pioli
 - EQU
 - NEQ
 - GEQ
 - LEQ
 - GRT
 - LES

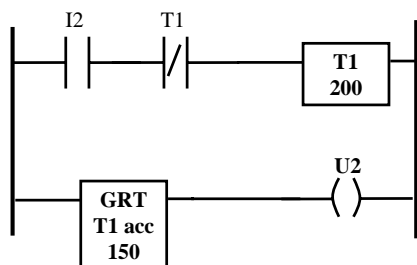


Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

4'

Esempio Oscillatore



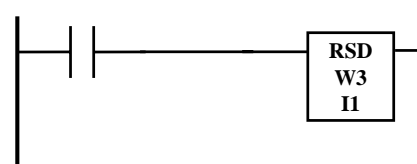
Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

4:

Registro a scorrimento a destra

- Registro di scorrimento a destra
 - Nel caso in cui il piolo sia attivato la word W viene spostata (*shift*) a destra di un bit
 - A sinistra entra il bit indicato come secondo operando



Automazione Industriale: Prof. Luca Ferrarini

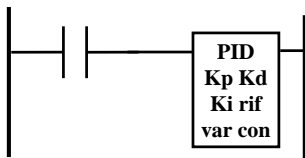
LD: NON distribuire, grazie!

4:

Istruzione PID

- Regolatore Proporzionale Integrale Derivativo

- Kp: guadagno proporzionale
- Kd : guadagno derivativo
- Ki: guadagno integrale
- rif: word del riferimento
- var: variabile da controllare
- con: valore del controllo



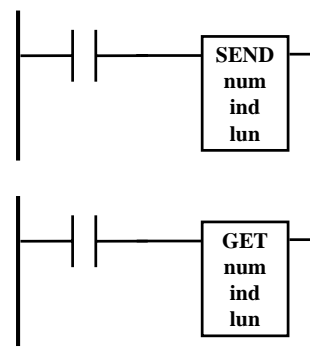
Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

4:

Istruzione di comunicazione via rete

- SEND
 - Invia un blocco di word ad un altro PLC connesso in rete
 - num: identificativo del PLC
 - ind: indirizzo di partenza del blocco da spedire
 - lun: lunghezza del blocco
- GET
 - Riceve un blocco di word da un altro PLC connesso in rete



Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

4:

Istruzioni di Temporizzazione

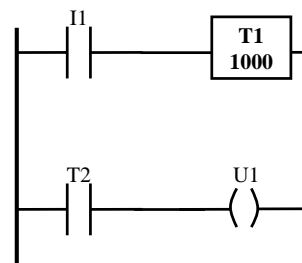
- Temporizzatore → Tx
 - Se il piolo che lo contiene consente il fluire della corrente, conta il trascorrere del tempo fino ad un valore preimpostato.
 - Quando tale valore è raggiunto Tx diventa vero.
 - In Tx.acc è possibile leggere il tempo trascorso
 - Se il piolo torna falso prima del completamento del tempo Tx si disattiva.
- Temporizzatore a ritenuta → TxR
 - Continua a contare anche se il piolo si inattiva diventa falso.
- Reset temporizzatore → RES
 - Ferma il temporizzatore e lo inizializza a 0.

Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

4:

Temporizzatori

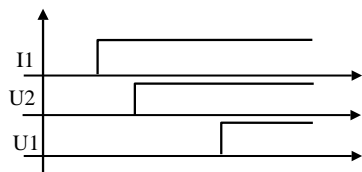
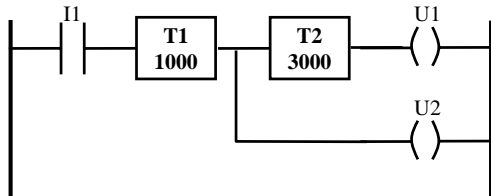


Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

4:

Temporizzatori

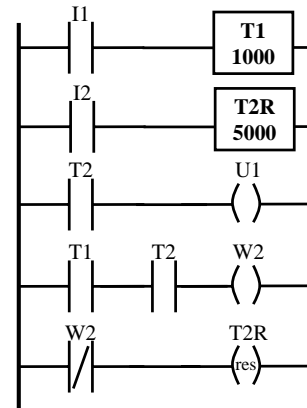


Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

4

Temporizzatori

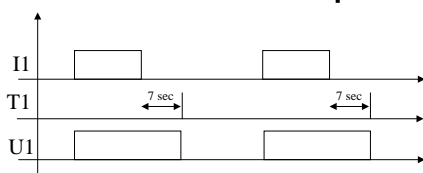
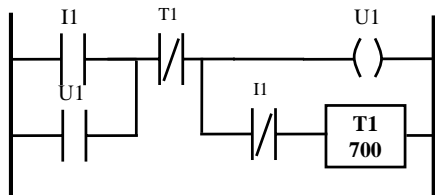


Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

5

Ritardo di Spegnimento

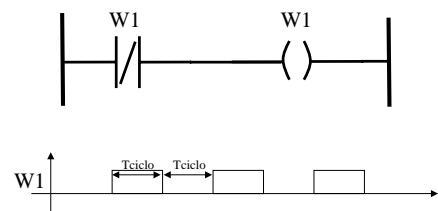


Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

5

Oscillatore ad Onda Quadra



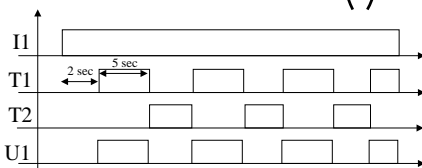
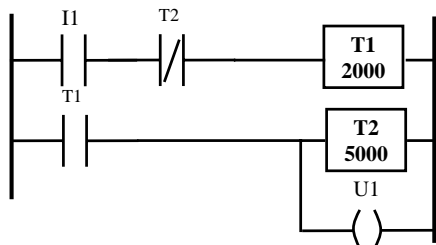
oscillatore ad onda quadra con periodo pari a 2 tempi di ciclo!

Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

5

Oscillatore ad Onda Quadra



Automazione Industriale: Prof. Luca Ferrarini

LD: NON distribuire, grazie!

5

Istruzioni di Conteggio

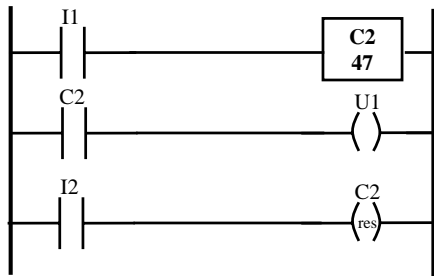
- Contatore ad incremento
 - Se il piolo di attivazione subisce una transizione falso→vero, allora il contatore Cx si incrementa di un'unità.
 - Cx.acc contiene il valore attuale del contatore
 - Cx diventa vero quando il contatore raggiunge il valore preimpostato.
- Reset contatore
 - Riporta a zero il contatore Cx → RES

Automazione Industriale: Prof. Luca Ferrarini

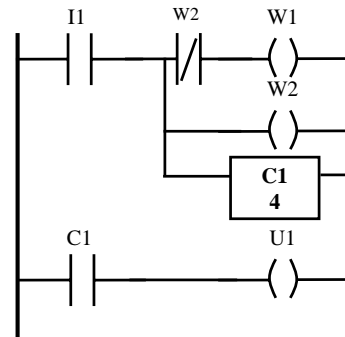
LD: NON distribuire, grazie!

5

Contatore



Conteggio di Eventi



Contatori e temporizzatori

