POLITECNICO DI MILANO

V Facoltà di Ingegneria

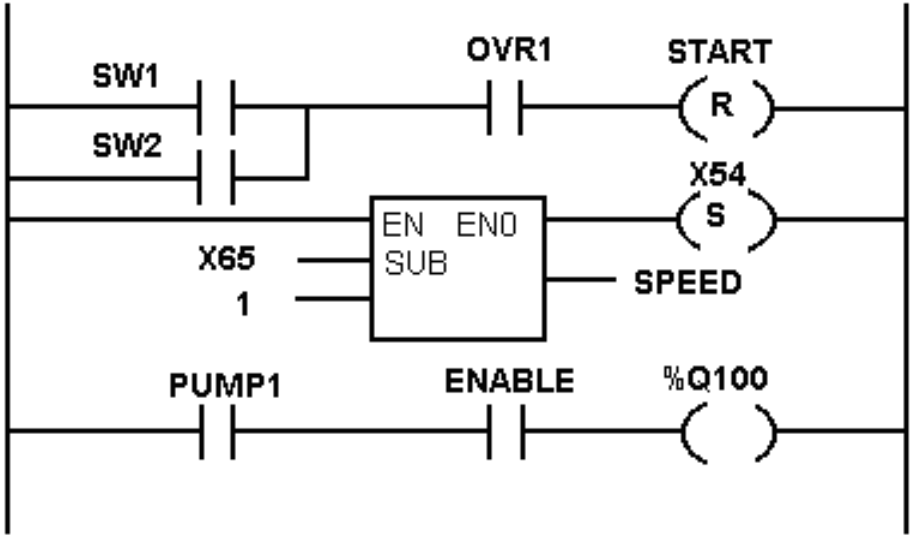
POLITECNICO DI MILANO
DEI

LD

Ladder Diagram

Sistemi ad Eventi Discreti

Sommario



- Introduzione
- Elementi Base
- Elementi Dinamici
- Temporizzazione e Contatori
- Controllo Programma e Blocchi Funzioni

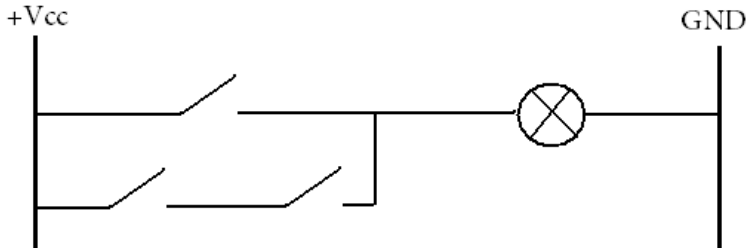
✓ Ladder Diagram (*LD*):

- Diagramma a scala per via della disposizione grafica dei simboli che richiama la forma di una scala a pioli

- Sinonimi italiani:
  - *"Linguaggio a contatti"*
  - *"Diagramma a relè"*

✓ Ladder Diagram (*LD*):

- Linguaggio grafico
- Si basa sulla trasposizione di una rete elettrica molto semplice in logica di programmazione
  - *Contatti*
  - *Bobine*
- Richiamano le vecchie logiche “pre-PLC” realizzate tramite reti elettriche cablate → *Facilitare il passaggio al nuovo*

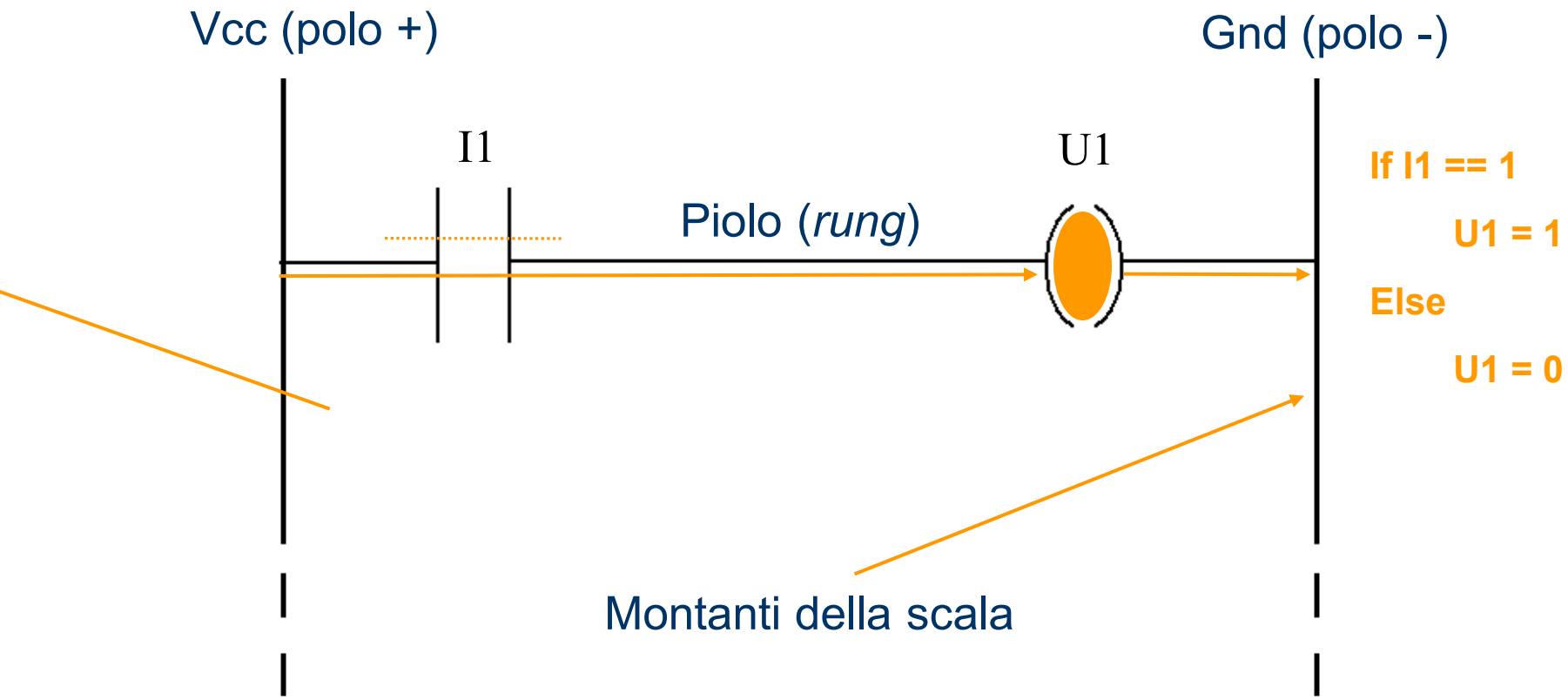


- Esistono convenzioni sia sulla disposizione grafica che sulla nomenclatura degli oggetti

Introduzione – Convenzioni e Aree Dati

- ✓ **Area degli Ingressi**
  - Ix:y (Il bit y della word x)
- ✓ **Area delle Uscite**
  - Ux:y (Il bit y della word x)
- ✓ **Area dei Temporizzatori (timer)**
  - $T_1 \ldots T_n$
- ✓ **Area dei Contatori**
  - $C_1 \ldots C_m$
- ✓ **Area PID e Area Utente**

NB: convenzioni variano da costruttore a costruttore,
nel corso useremo nomi simbolici semplificati!

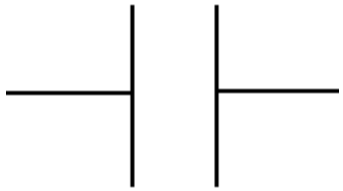


Significato: SE il contatto I1 è CHIUSO, ALLORA la corrente può andare dal polo + al polo -, attivando così la bobina U1

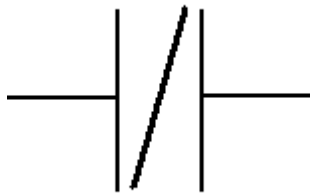
Elementi Base - Ingressi

- ✓ **Gli ingressi sono inseriti sempre sulla sinistra del piolo**

- ✓ **Contatto Normalmente Aperto**
  - Se il bit associato vale **1**, il contatto si **chiude**
  - Se il bit associato vale **0**, il contatto si **apre**

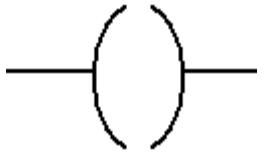


- ✓ **Contatto Normalmente Chiuso**
  - Se il bit associato vale **1**, il contatto si **apre**
  - Se il bit associato vale **0**, il contatto si **chiude**

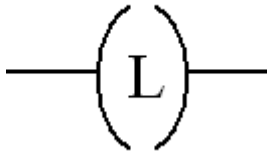


Elementi Base - Uscite

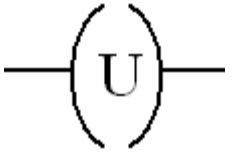
- ✓ **Le uscite sono inserite sempre sulla destra del piolo**
- ✓ **Bobina**
  - **Si attiva quando passa corrente**. Quindi, il bit ad essa associata sale al valore logico 1 (ON) se le condizioni logiche alla sua sinistra sono verificate



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

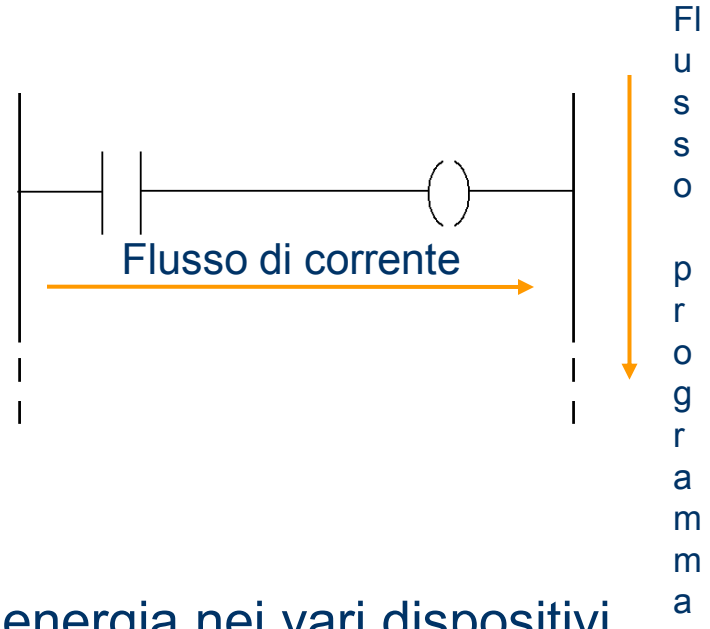


- ✓ **Unlatch Bobina**
  - **Riporta allo stato logico 0** (OFF) un'uscita (simile al SET di un flip-flop)



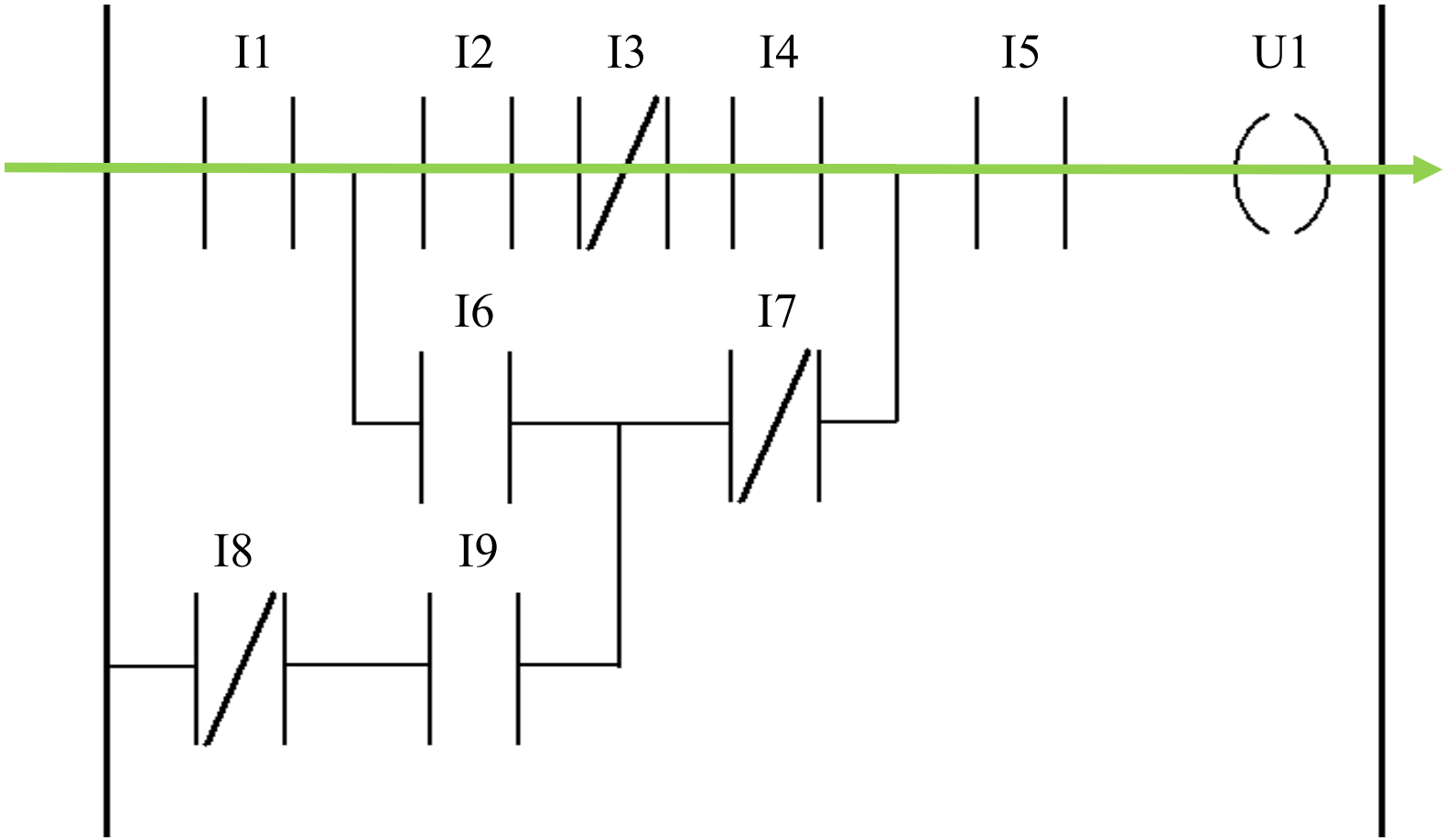
Elementi Base - Programma

- **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.
  - I pioli vengono scanditi:
    - **Dall'alto verso il basso**
    - **Da sinistra a destra**



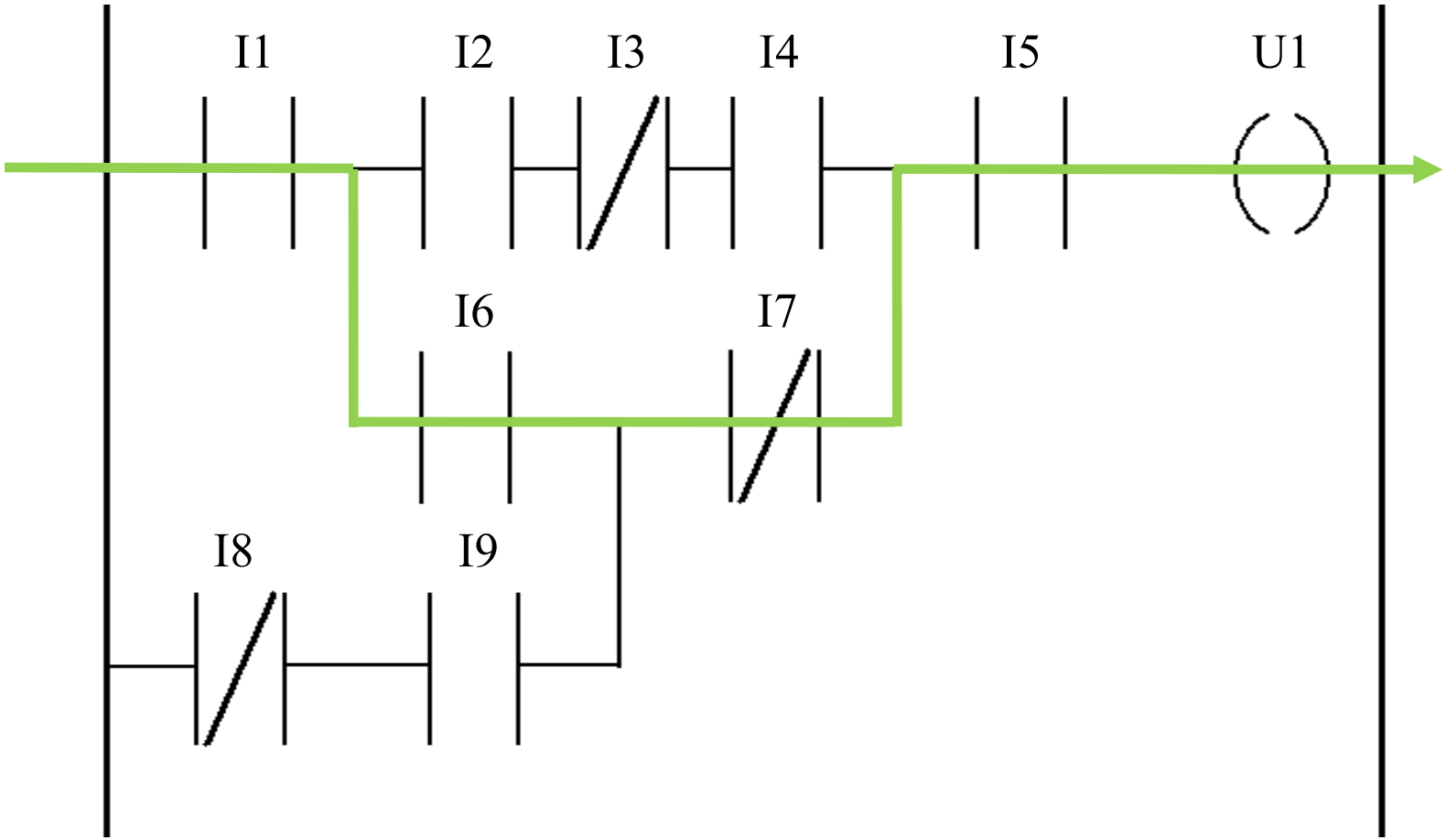
  - Quindi, nel singolo piolo, il flusso di energia nei vari dispositivi va solo da sinistra a destra, senza possibilità che questo si inverta.

Elementi Base – Regole di scansione dei pioli



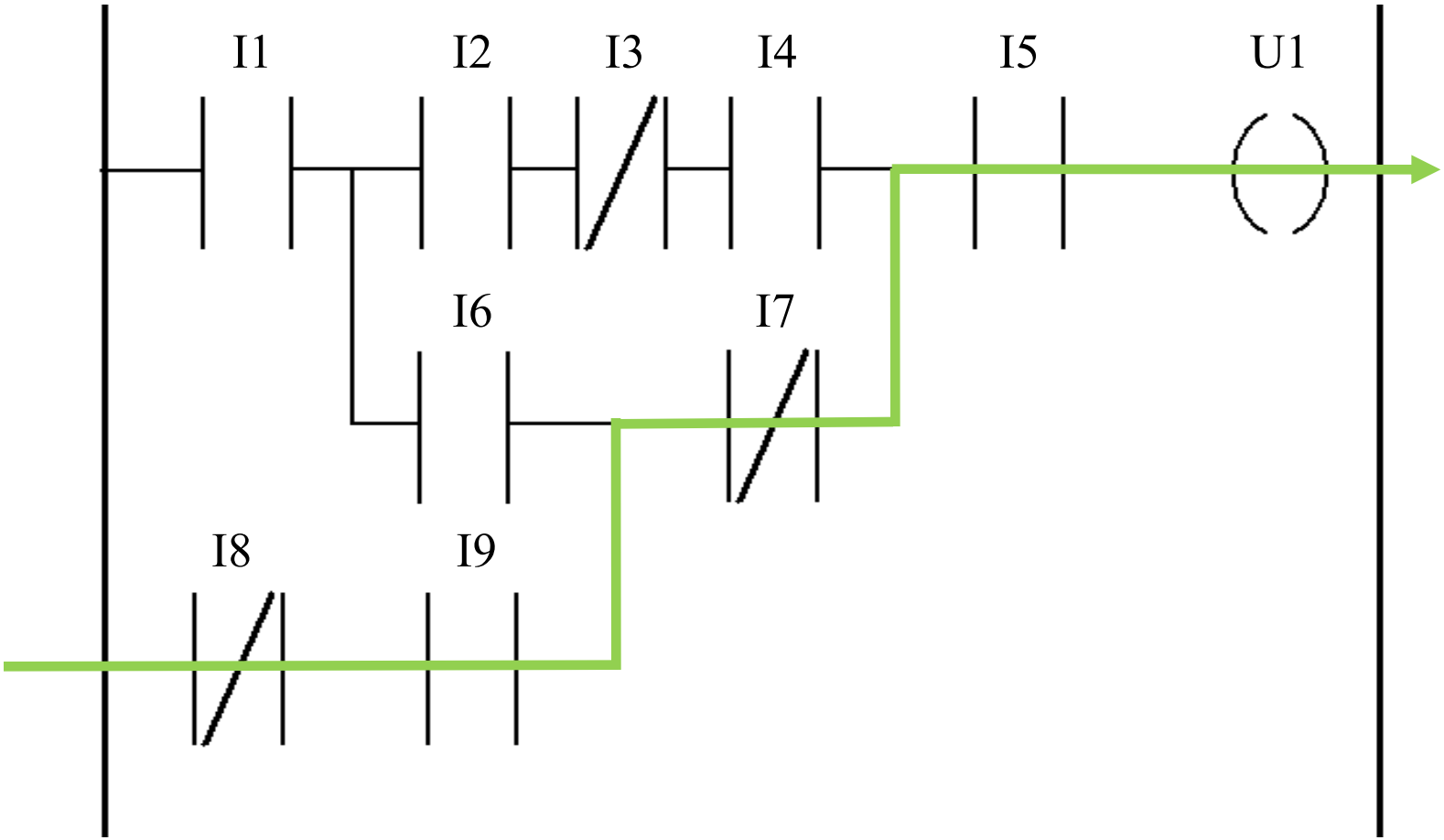
- ✓ **Flusso di energia solo da sinistra a destra senza possibilità di inversione**

Elementi Base – Regole di scansione dei pioli



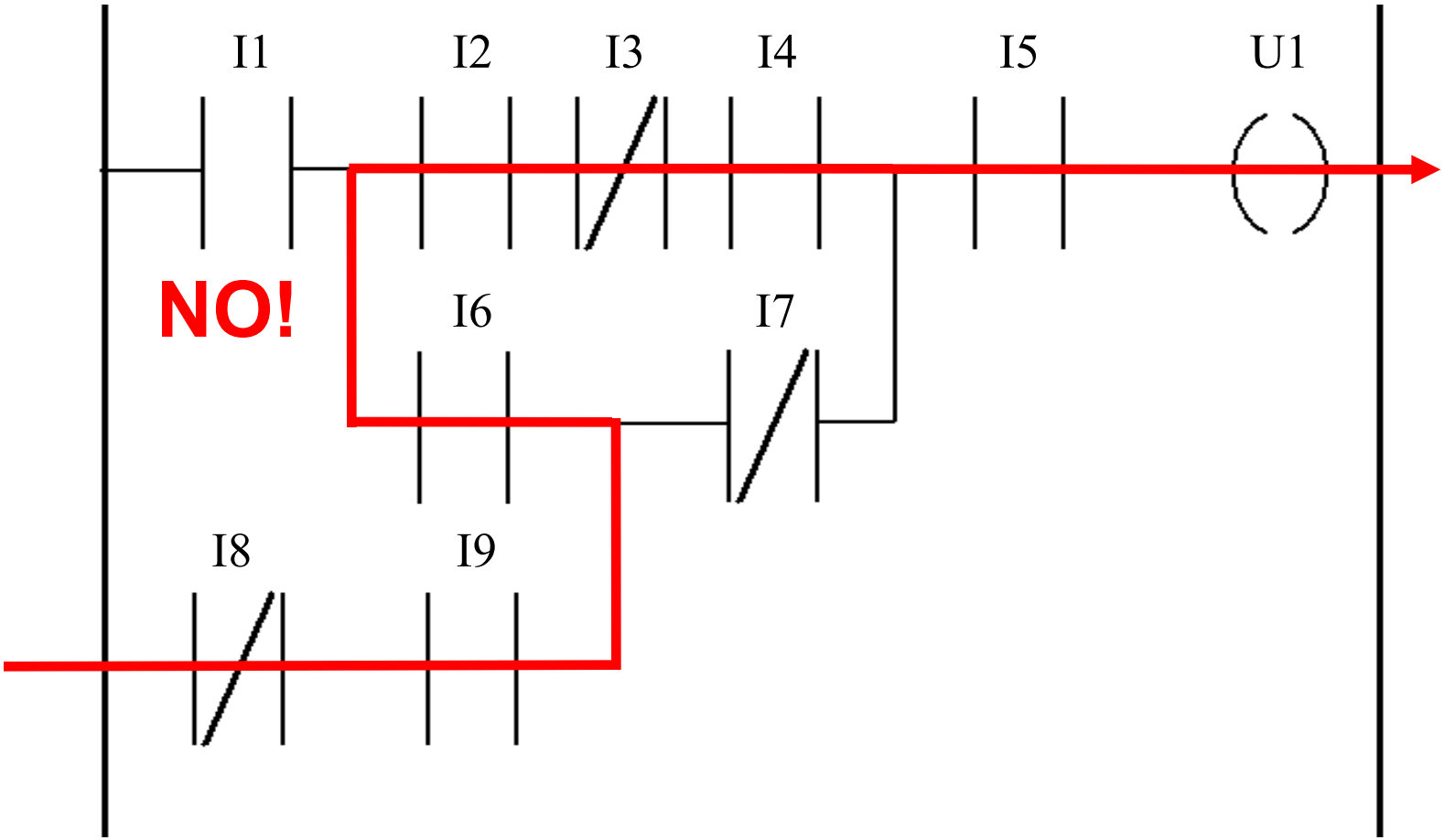
- ✓ **Flusso di energia solo da sinistra a destra senza possibilità di inversione**

Elementi Base – Regole di scansione dei pioli



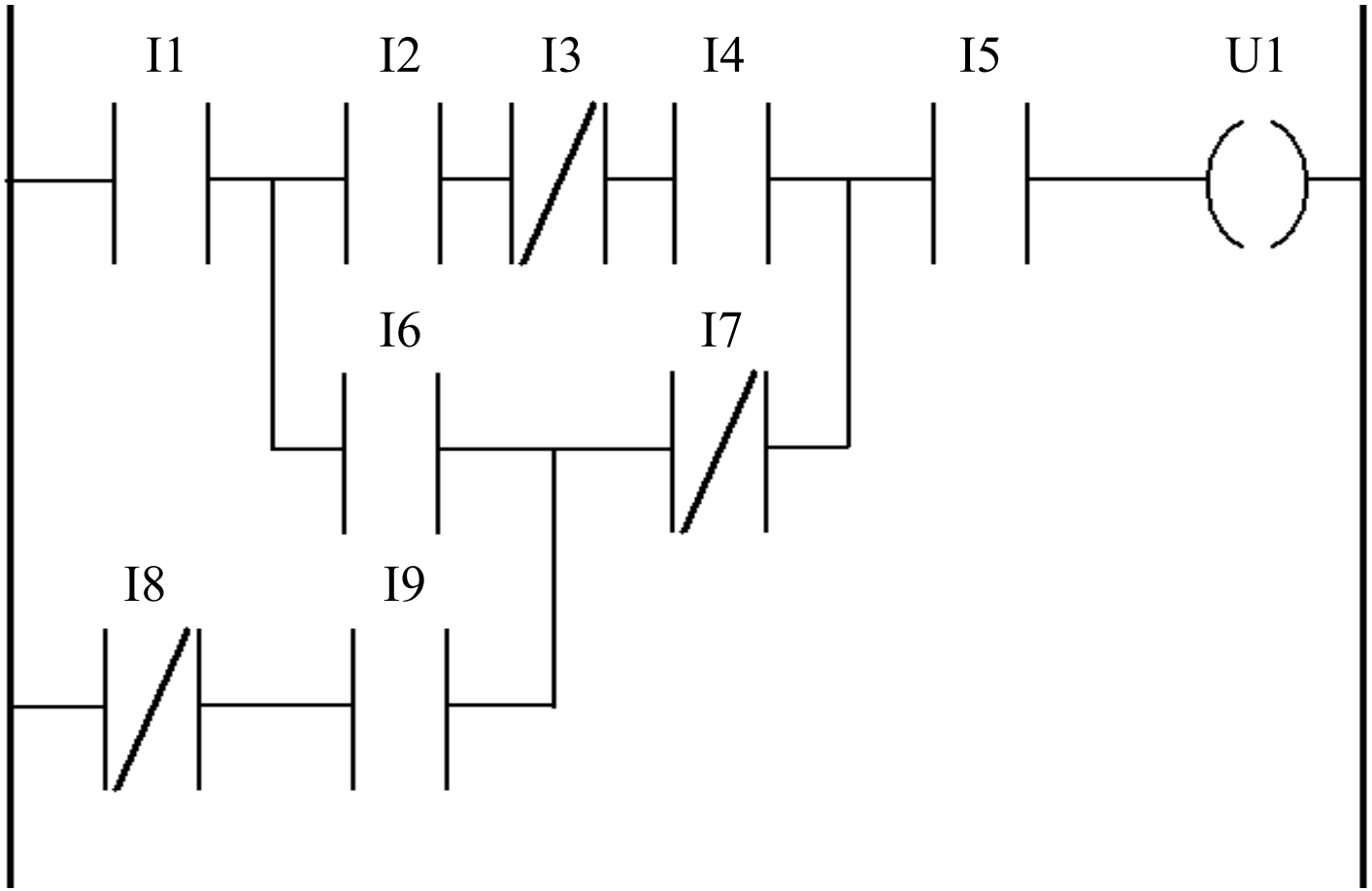
- ✓ **Flusso di energia solo da sinistra a destra senza possibilità di inversione**

Elementi Base – Regole di scansione dei pioli



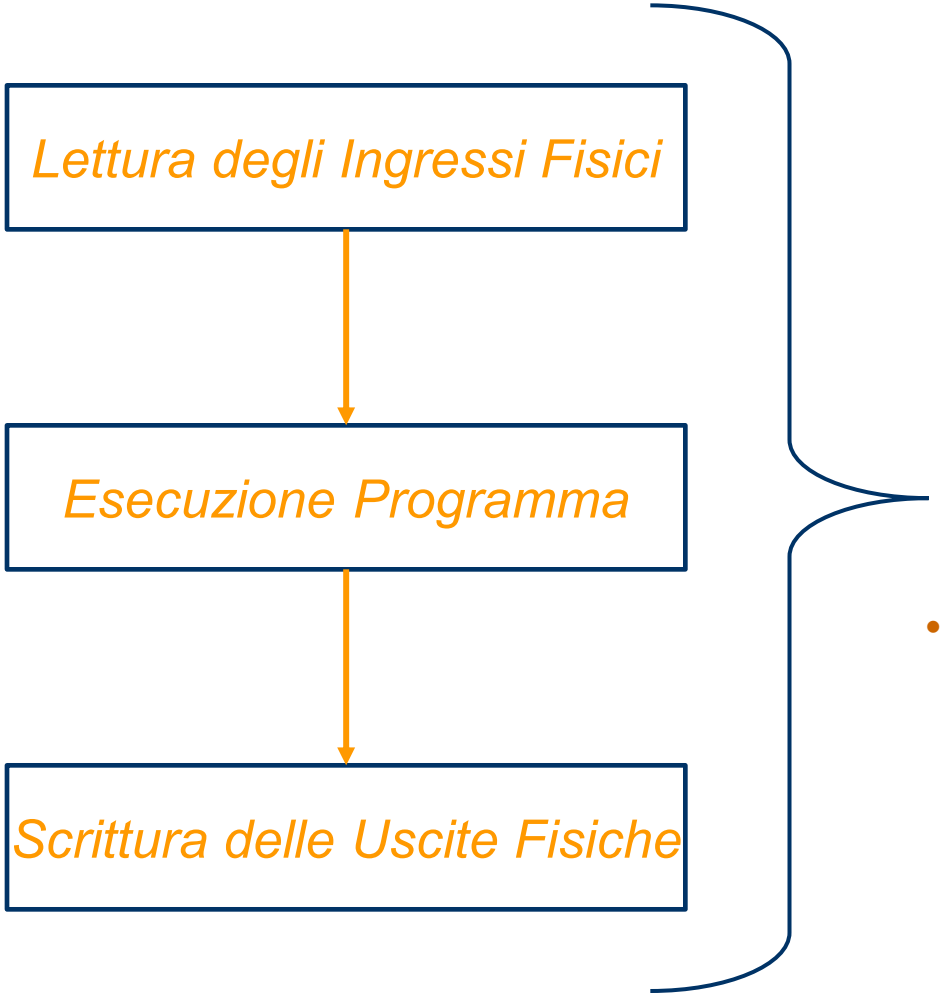
- ✓ **Flusso di energia solo da sinistra a destra senza possibilità di inversione**

Elementi Base – 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)))**]

Esecuzione – Ciclo a Coppia Massiva



CICLO A COPPIA MASSIVA

Ad ogni ciclo vengono:

- Letti gli ingressi ed aggiornate le variabili del programma
- Eseguita la sequenza di pioli
- Scritte le uscite in accordo con i valori delle variabili del programma

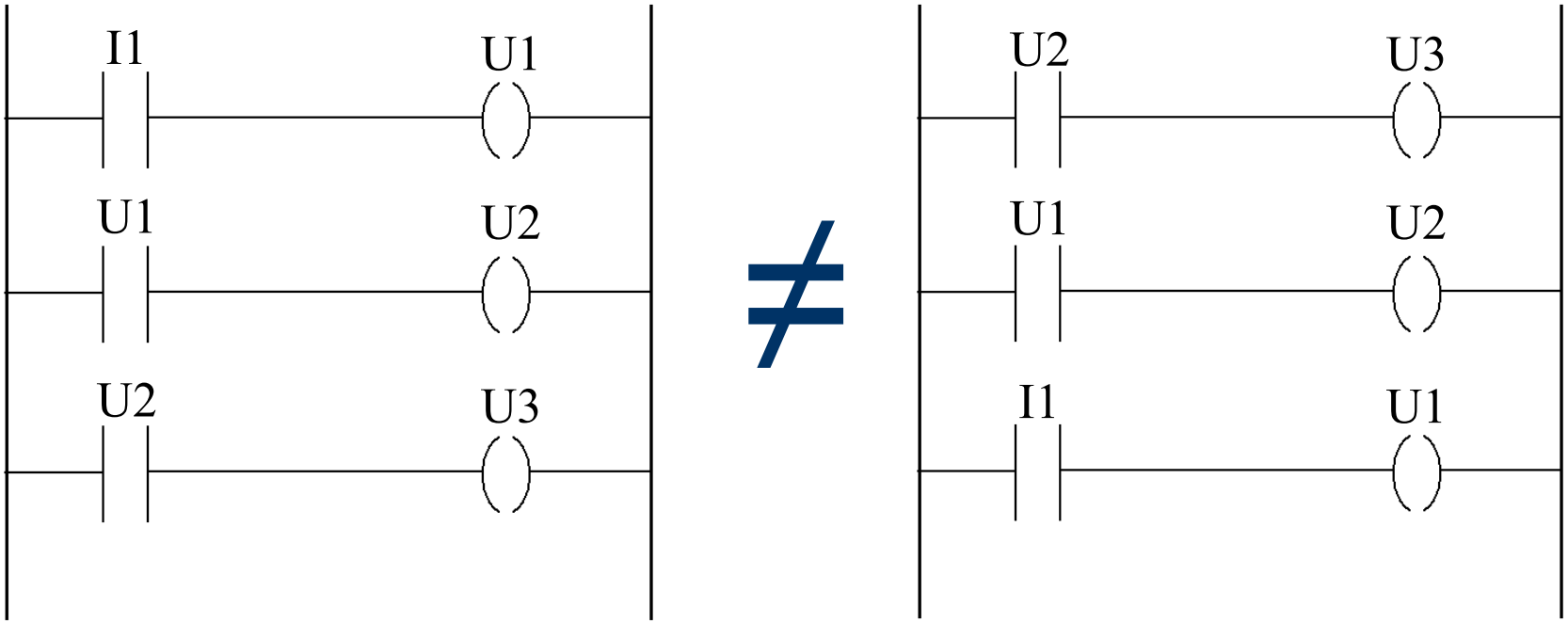
Esecuzione – Sincronizzazione con I/O

- 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.

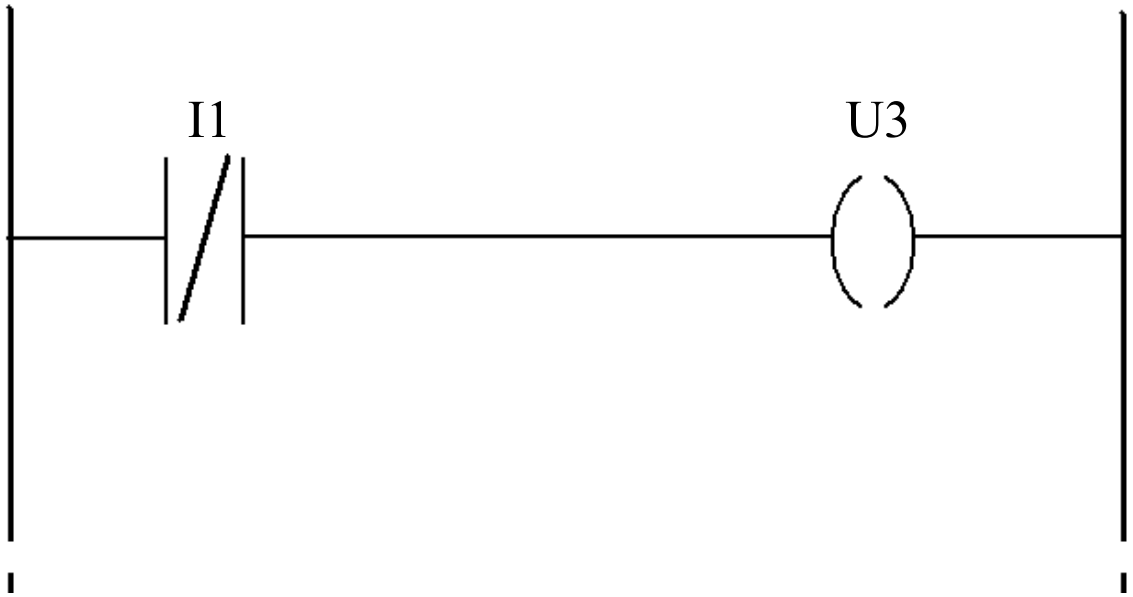
- Quindi con questo LD ad ogni ciclo scrivo 0 o 1 in U1! Inoltre, il suo valore permane fino alla prossima esecuzione (al ciclo successivo) della stessa istruzione.

Esecuzione – Sincronizzazione con I/O

- Il valore delle variabili lette in ingresso rimane costante per tutto ciclo del programma quindi i seguenti due programmi sono diversi:

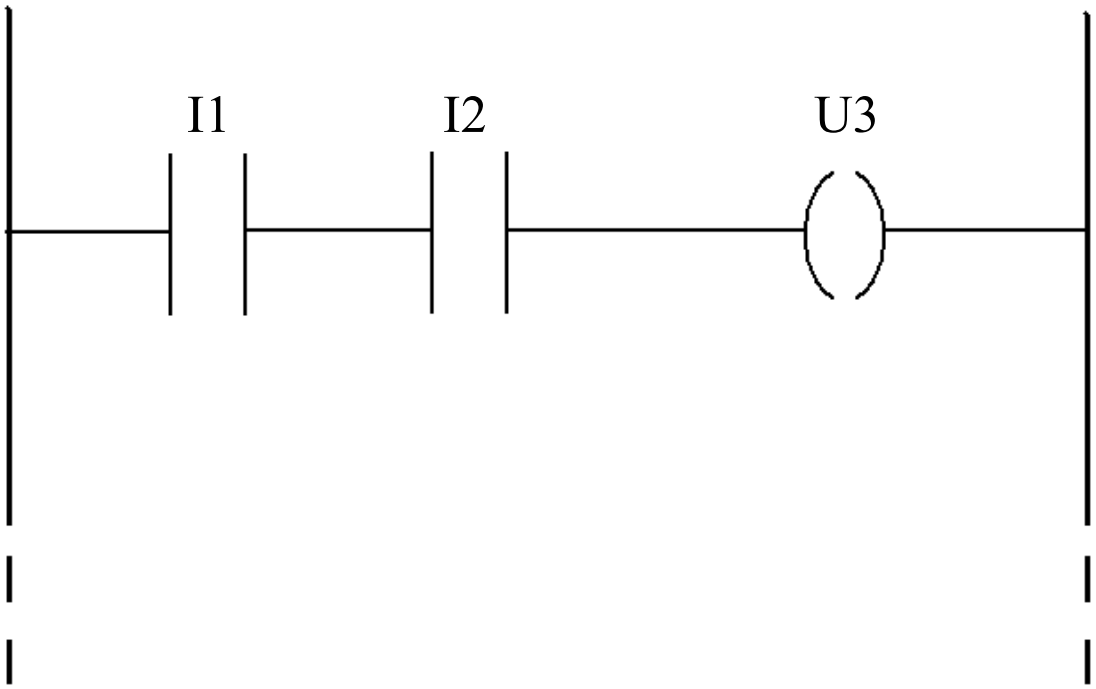


Esempi – Funzione logica NOT



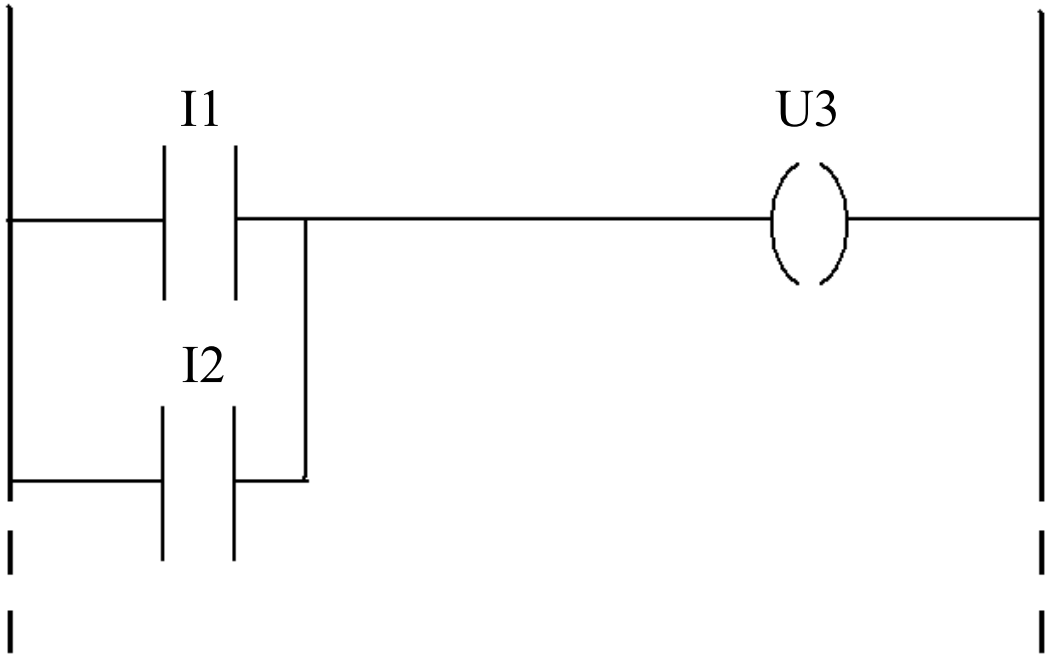
U3 = NOT (I1)

Esempi – Funzione logica AND



U3 = I1 AND I2

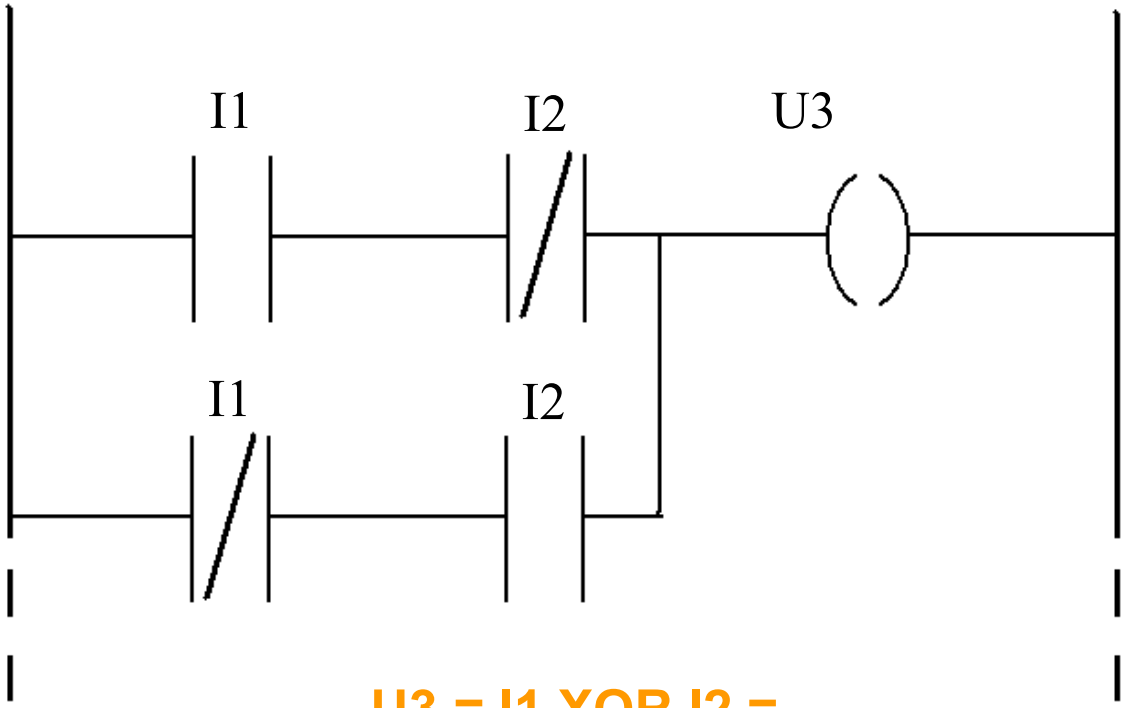
Esempi – Funzione logica OR



U3 = I1 OR I2

Esempi – Funzione logica XOR

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

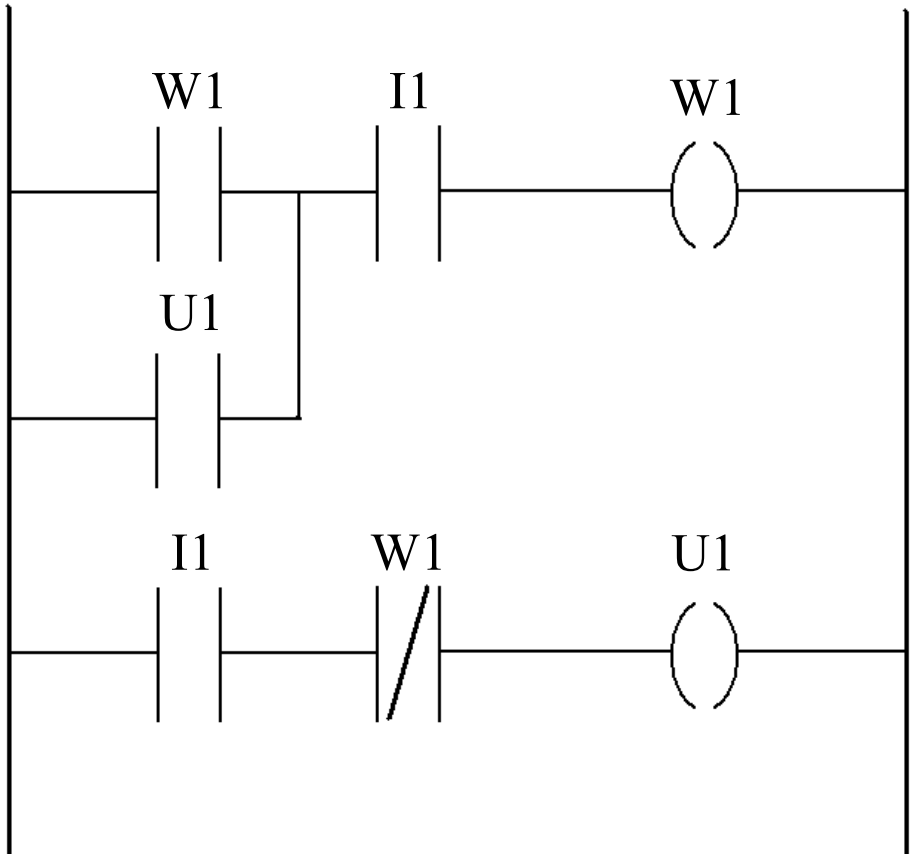


U3 = I1 XOR I2 =
[I1 AND NOT(I2)] OR [NOT(I1) AND I2]

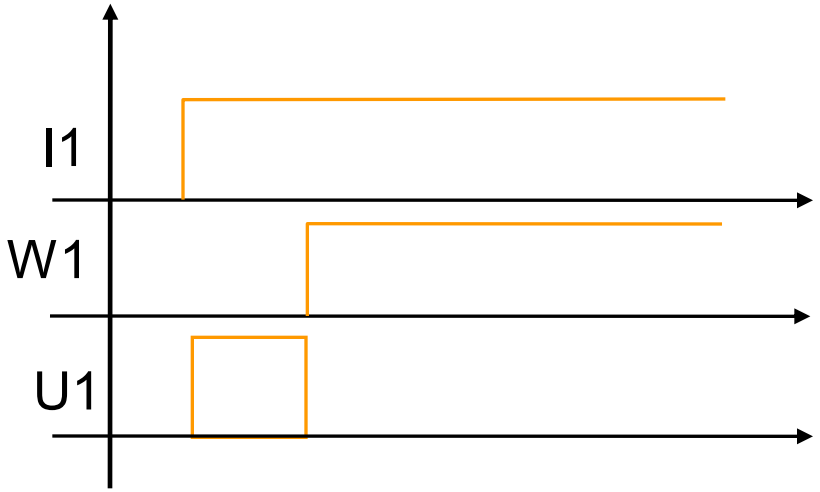
✓ Elementi Dinamici

- Le istruzioni di base del LD sono “logiche”:
  uscita = 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.

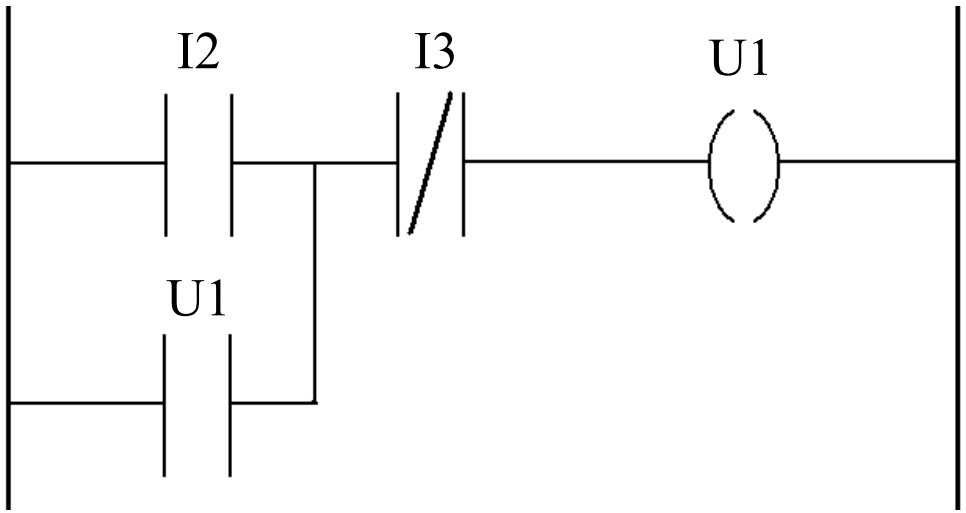
Elementi di memoria



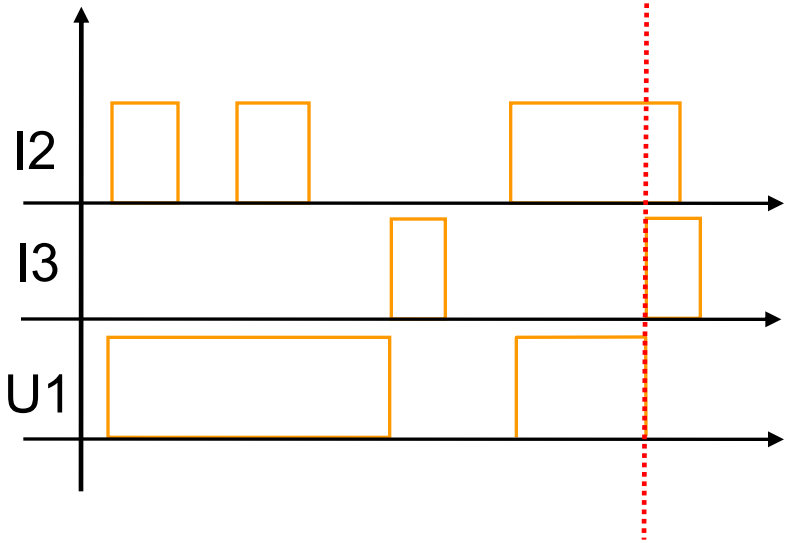
- ✓ *Monostabile*
  (genera un impulso di durata un tempo di ciclo dopo un fronte di salita su I1)



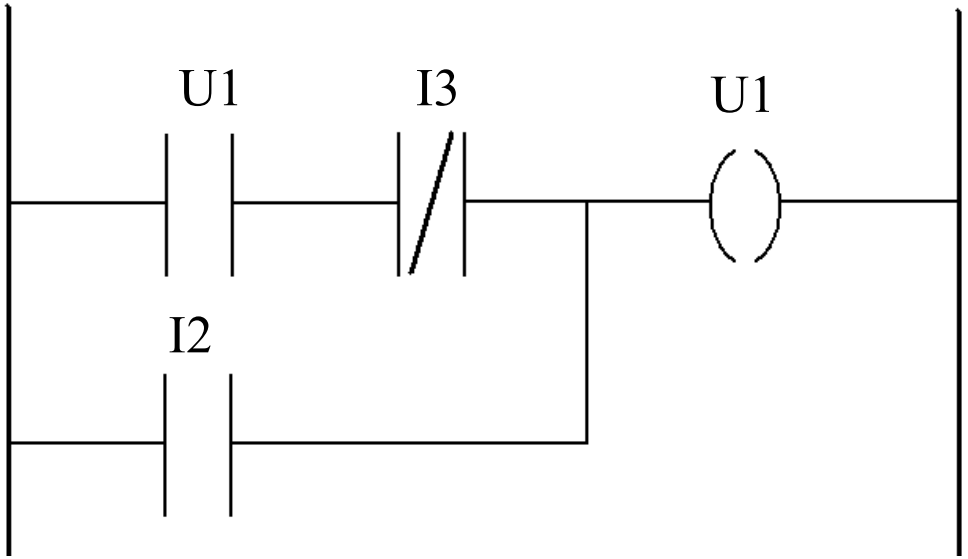
Elementi di memoria



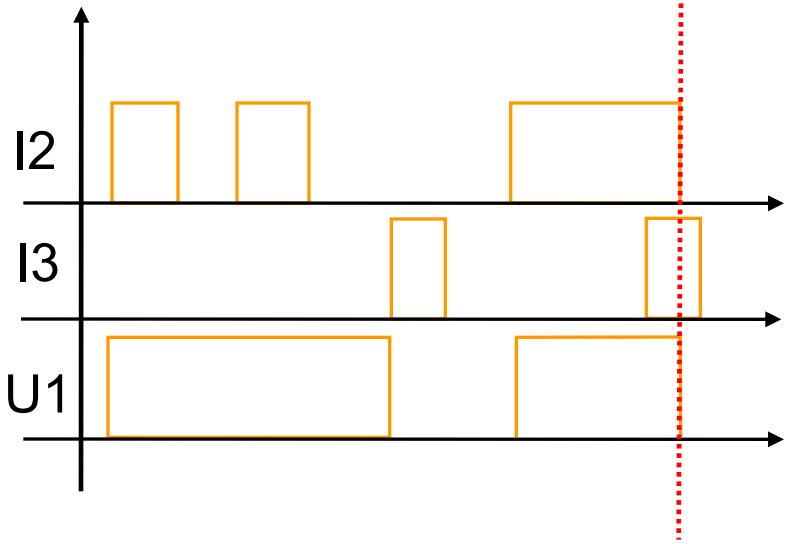
- ✓ *Flip-Flop a reset vincente*
  I2 set
  I3 reset



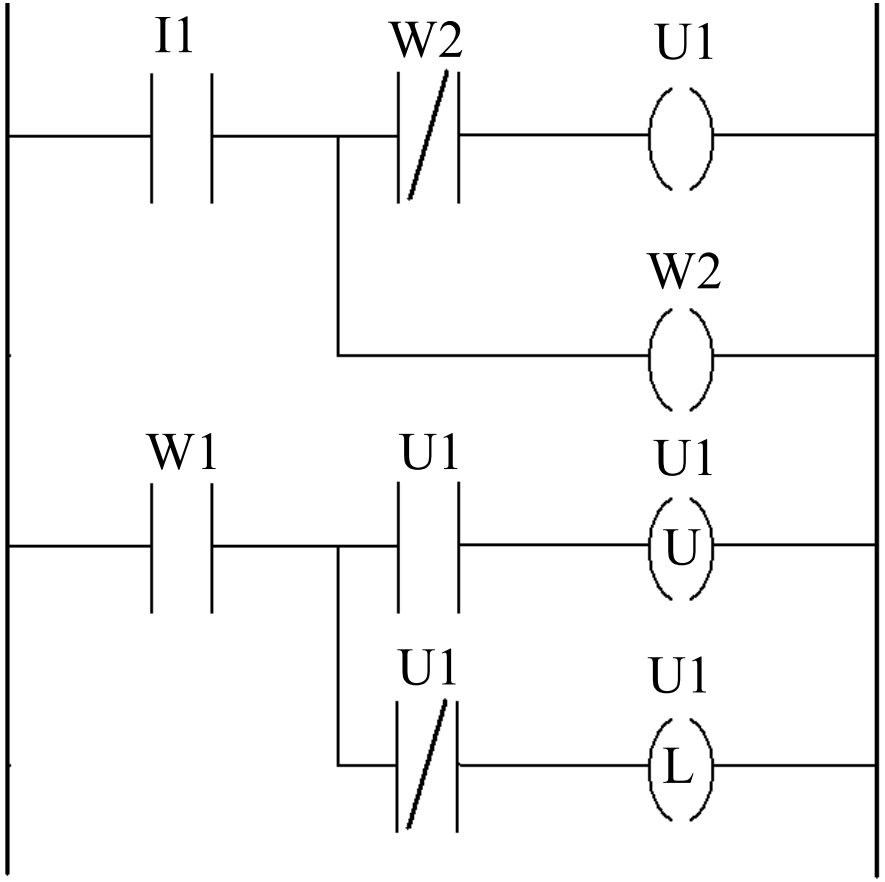
Elementi di memoria



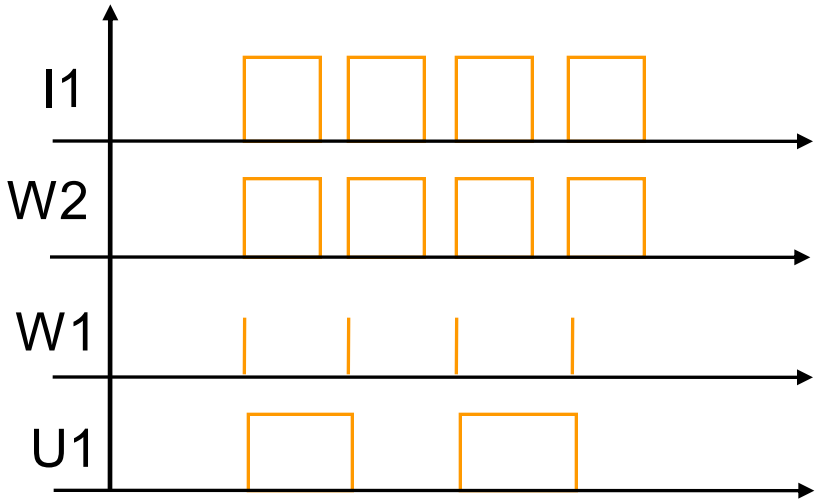
- ✓ *Flip-Flop a set vincente*
  I2 set
  I3 reset



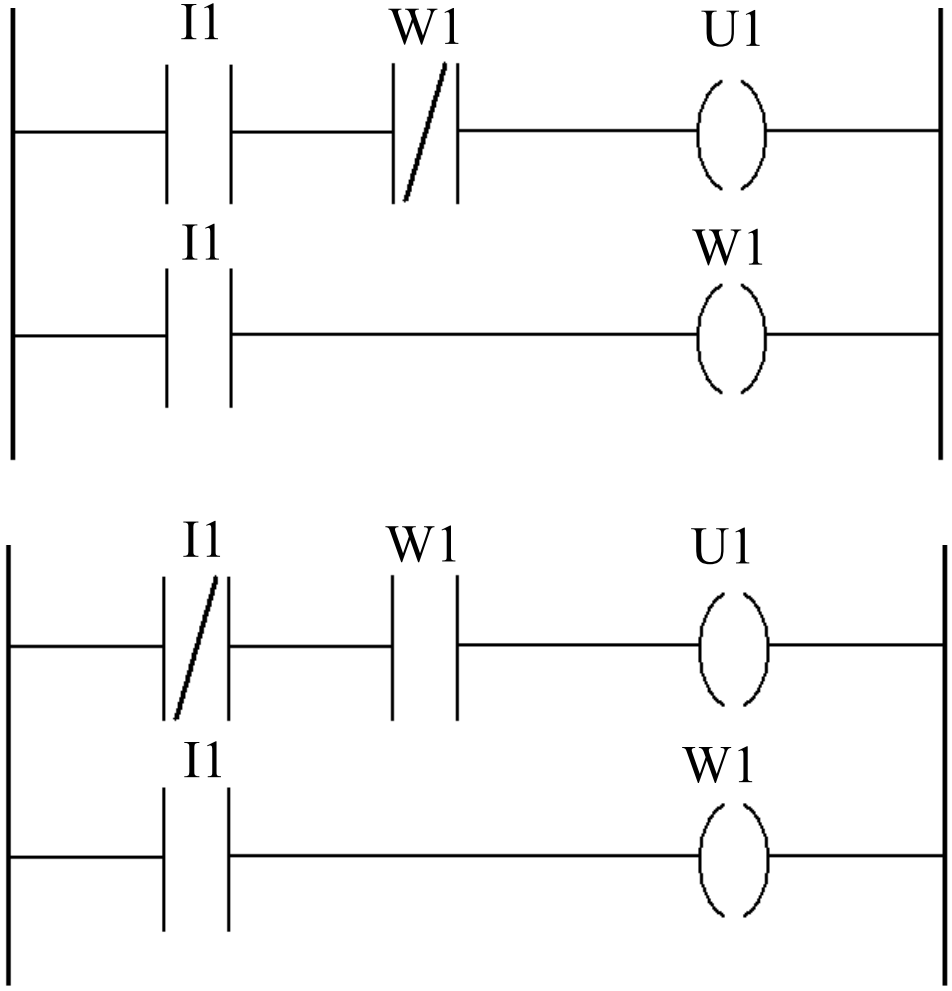
Elementi di memoria



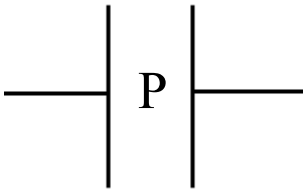
- *Flip-flop di tipo D (un impulso su I1 fa commutare l'uscita)*



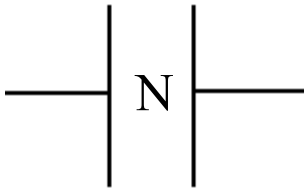
Riconoscimento di fronti: varianti



- *Alcuni ambienti CAD di programmazione di PLC in LD prevedono la presenza di istruzioni avanzate per il riconoscimento dei fronti:*
  - *Positive edge*



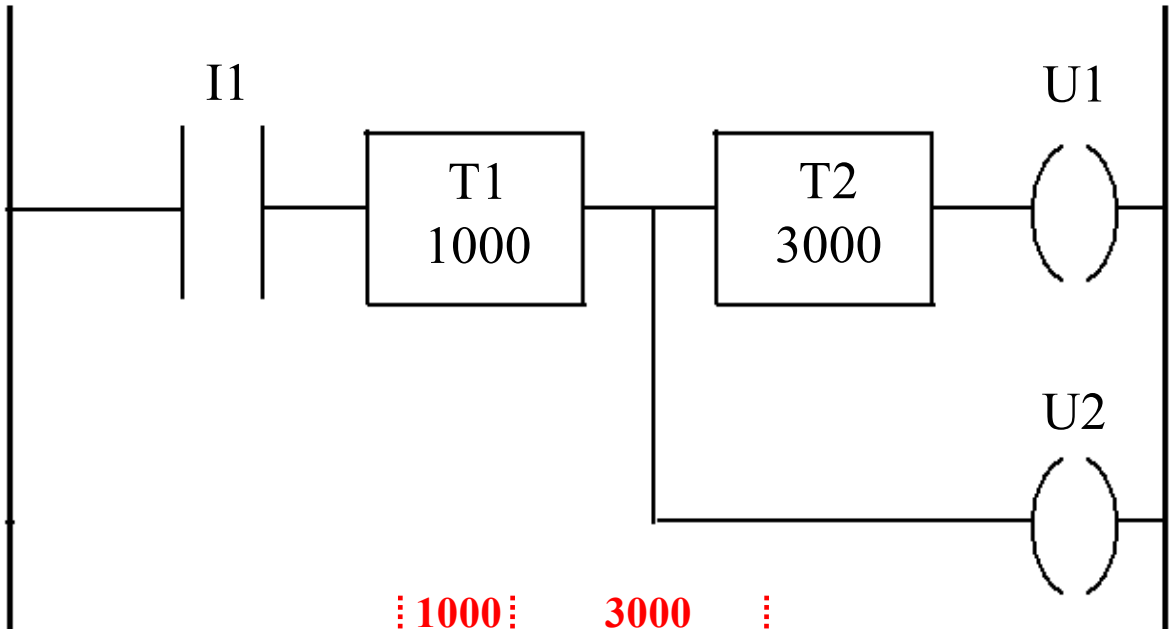
  - *Negative edge*

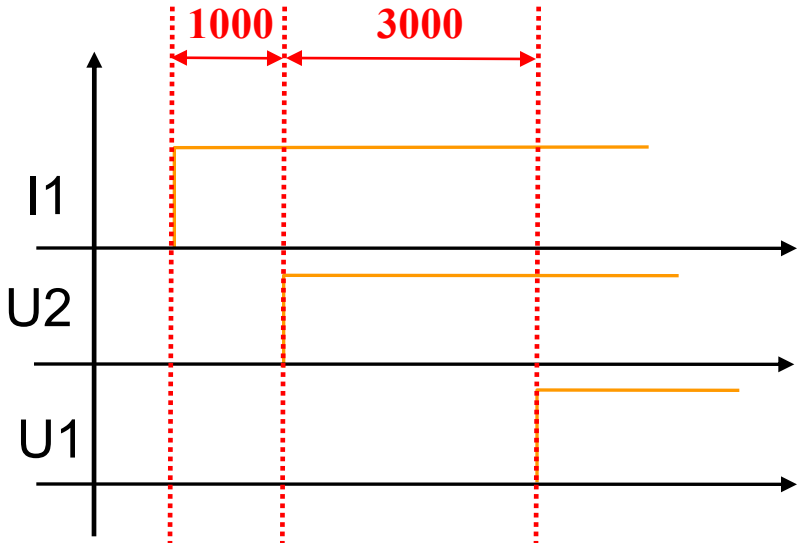


Istruzioni di Temporizzazione

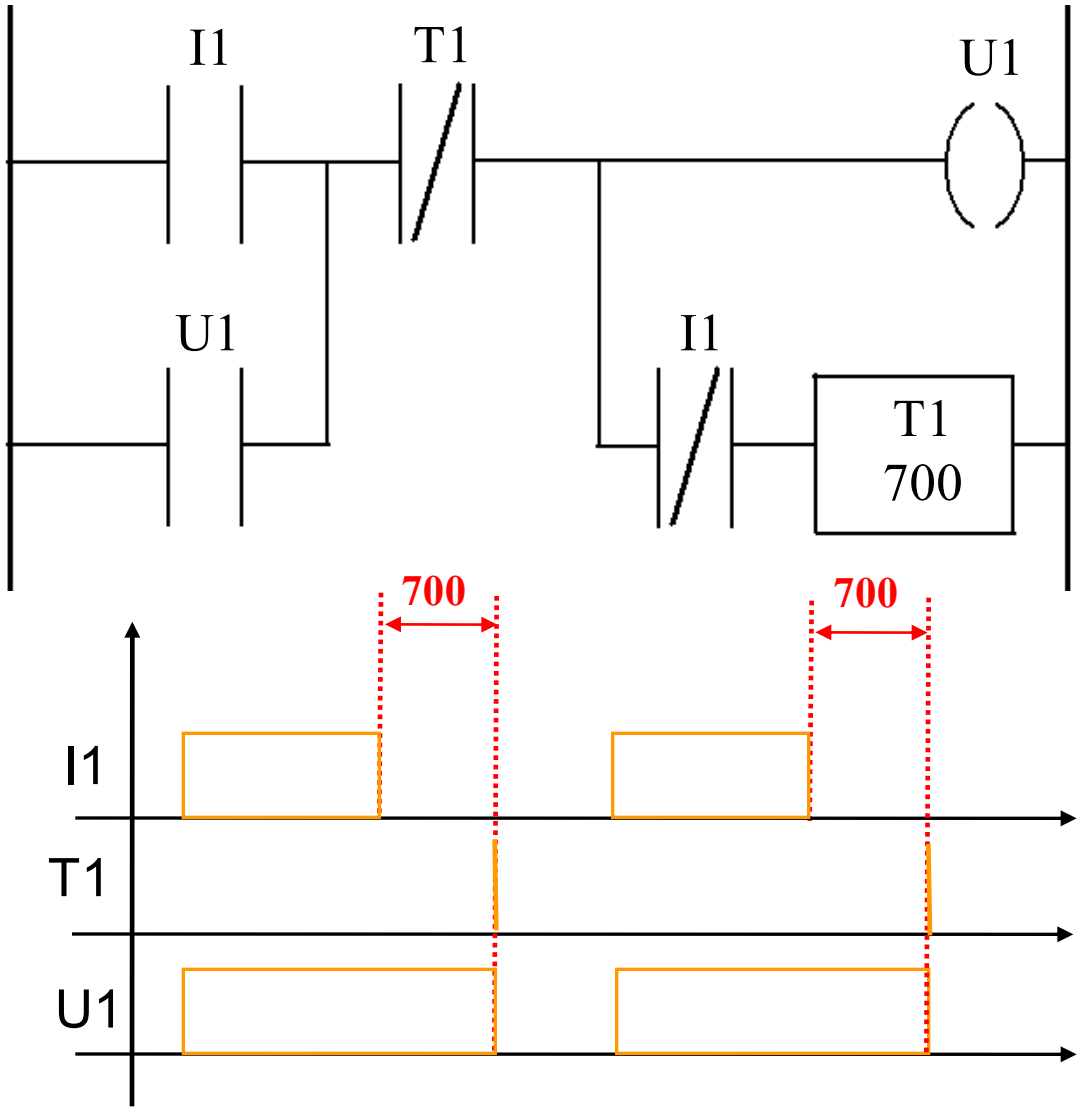
- **Temporizzatore → Tx**
  - Se il piolo che lo contiene consente il fluire della corrente, conta il trascorrere del tempo fino ad un valore preimpostato.
  - Quando tale valore è raggiunto Tx diventa vero.
  - In Tx.acc è possibile leggere il tempo trascorso.
  - Se il piolo torna falso prima del completamento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

Istruzioni di Temporizzazione

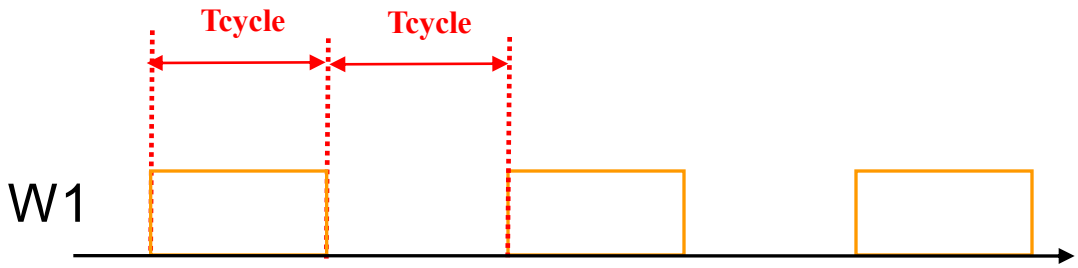




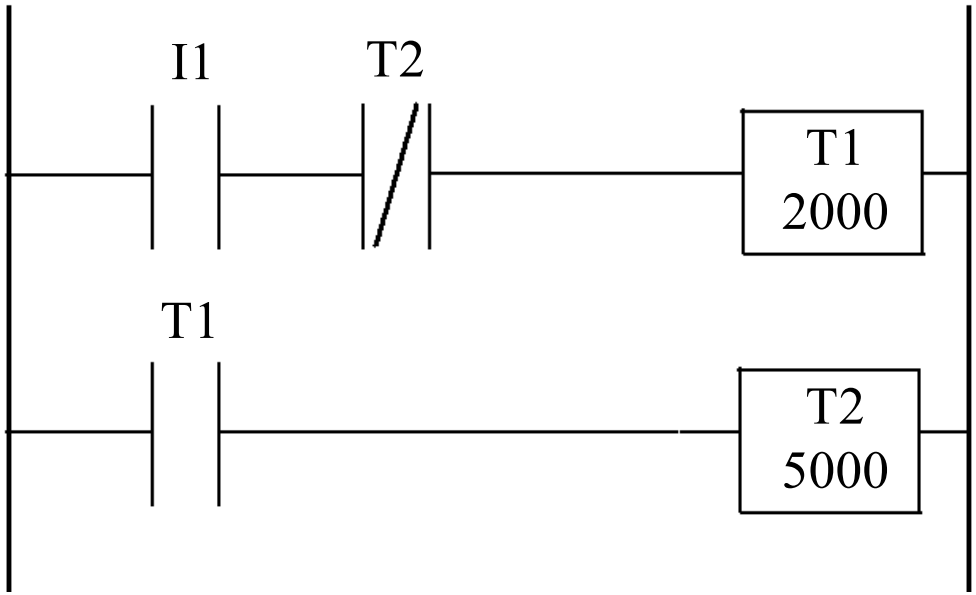
Ritardo di spegnimento

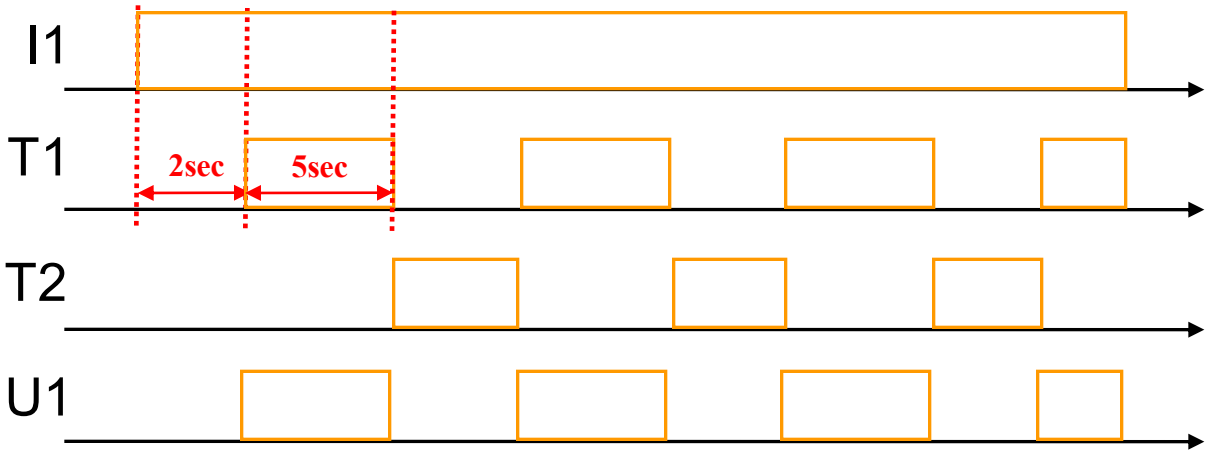


Oscillatore ad Onda Quadra

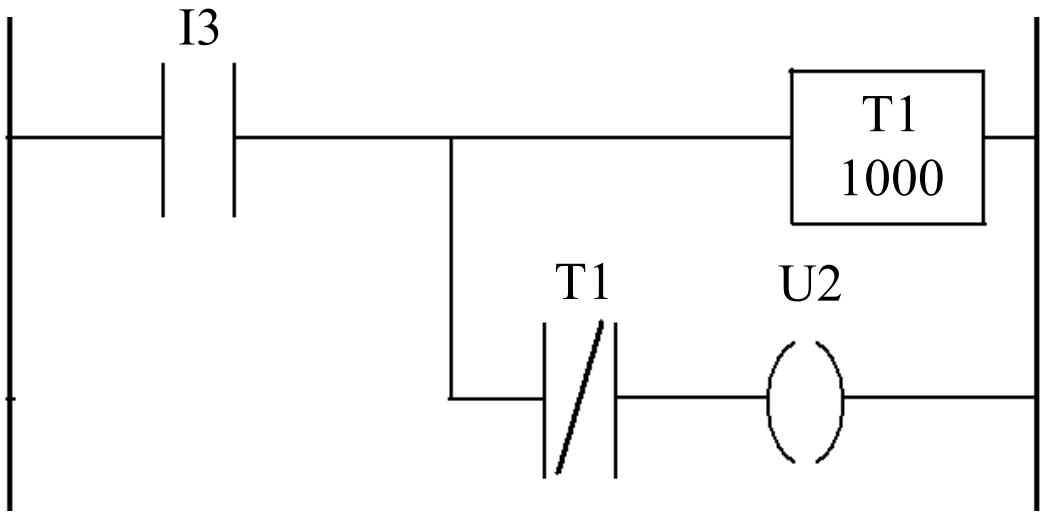


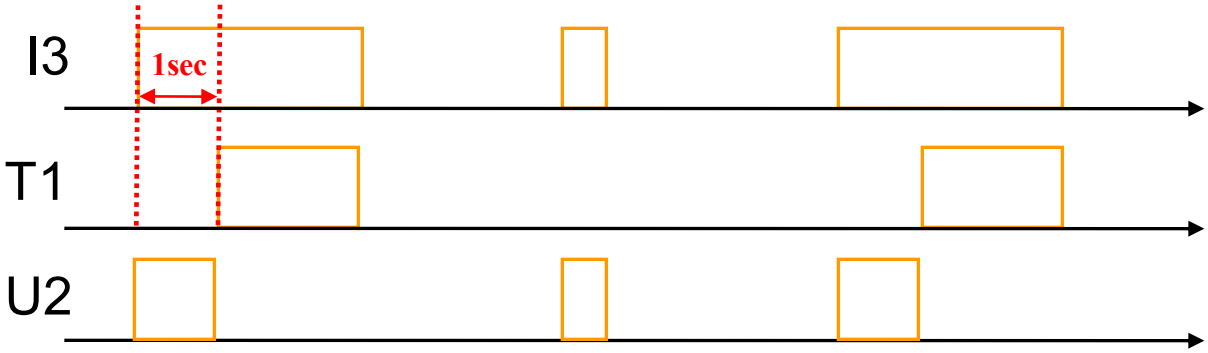
Oscillatore ad Onda Quadra



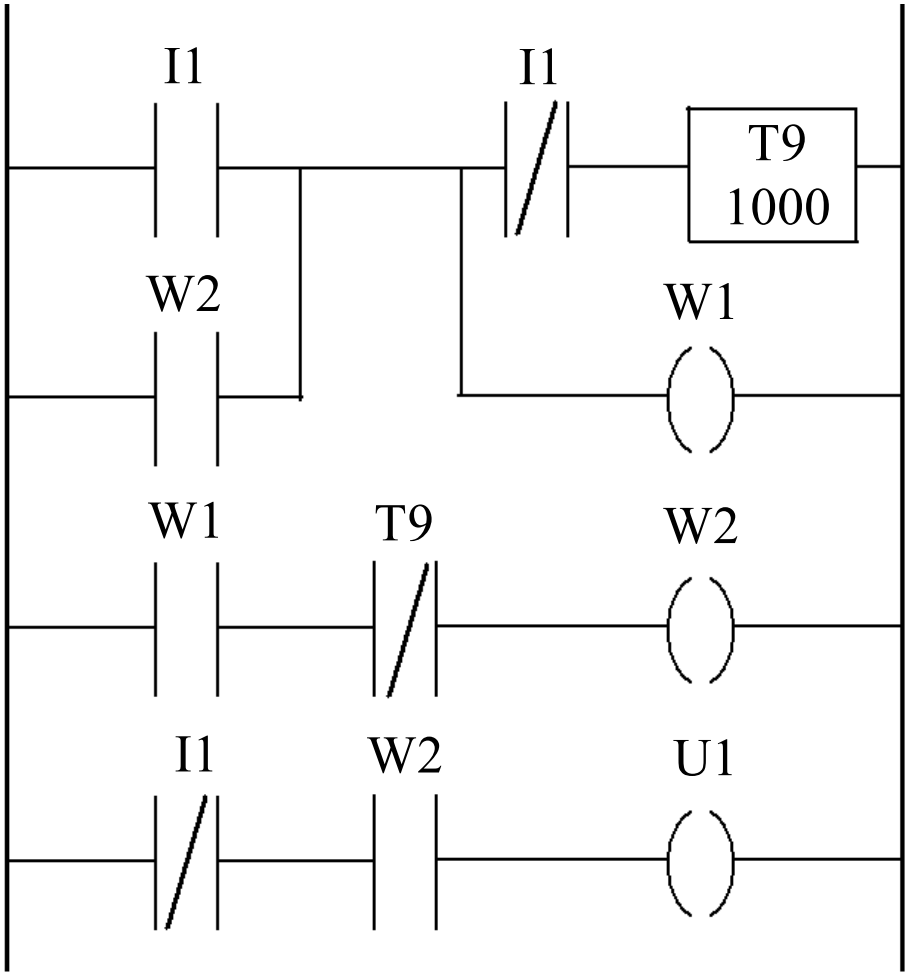


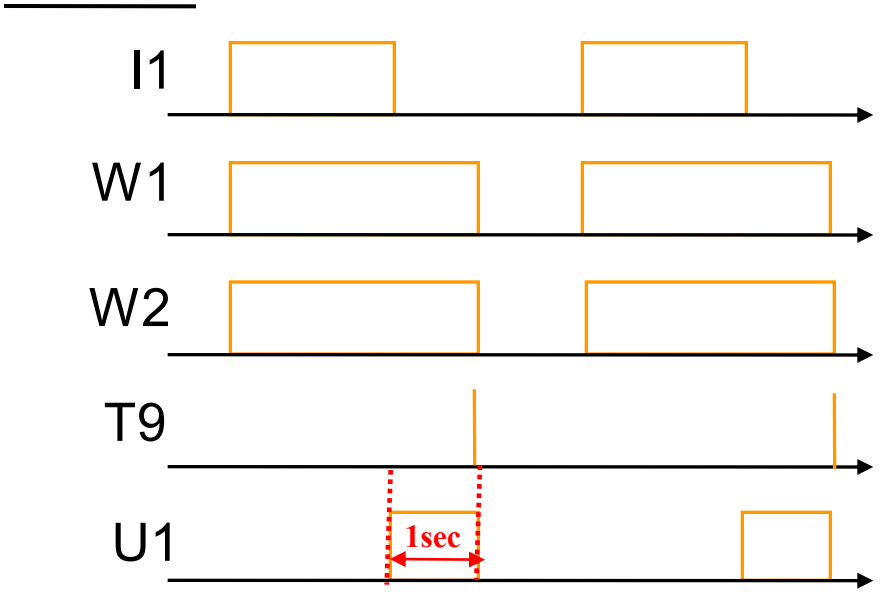
Impulso all'Accensione



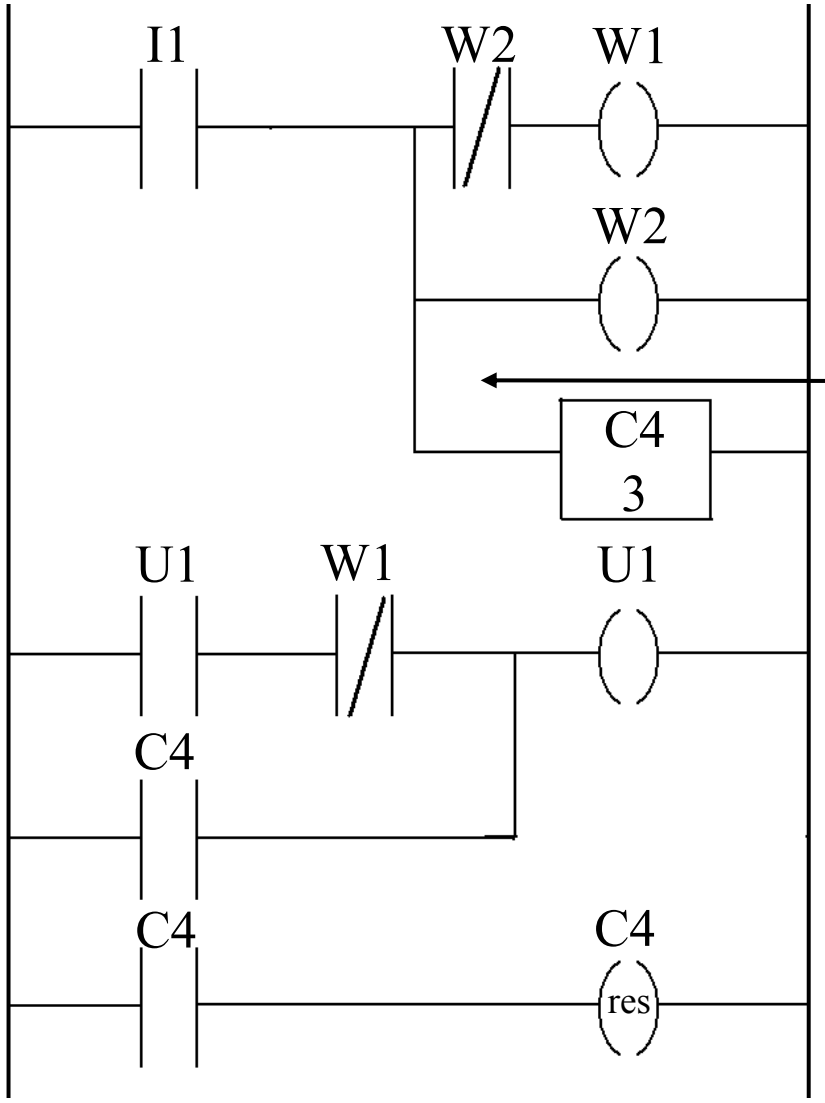


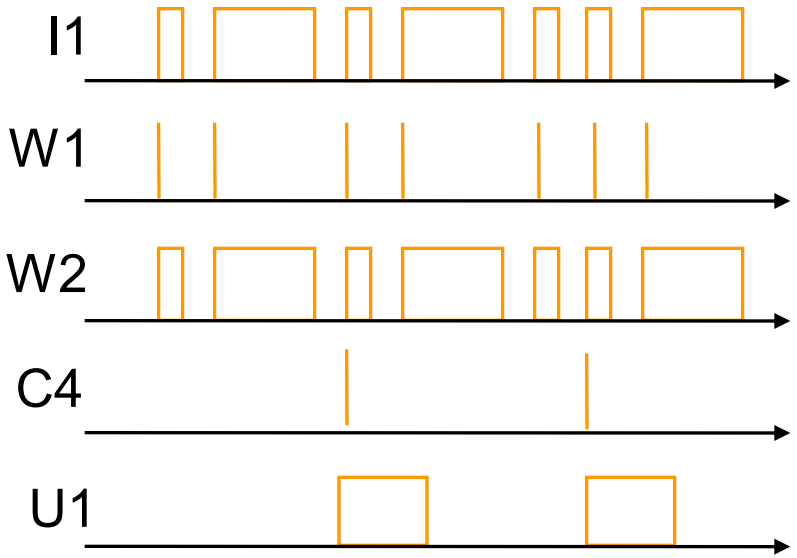
Impulso allo Spegnimento





Conteggio ad Eventi



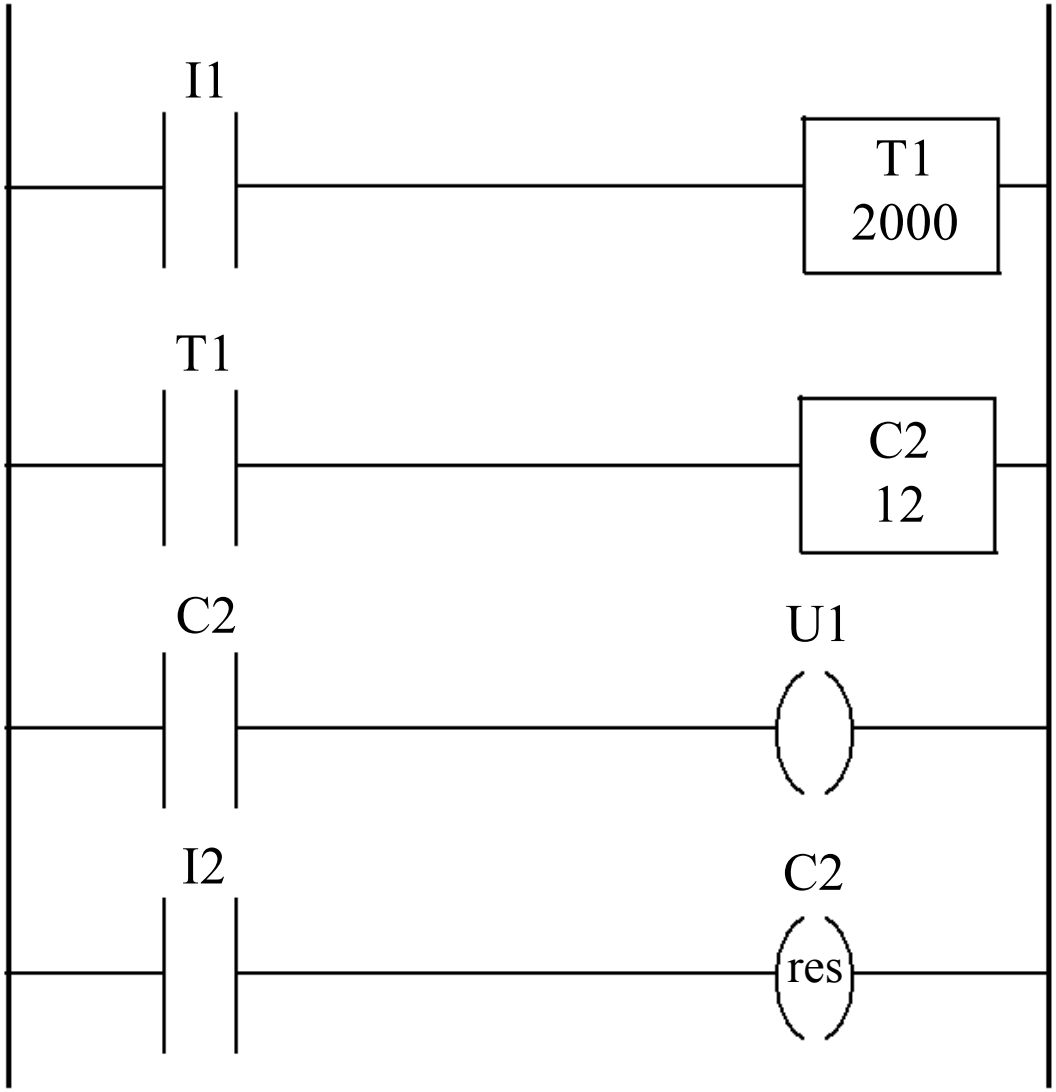


Istruzioni di Conteggio

- **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

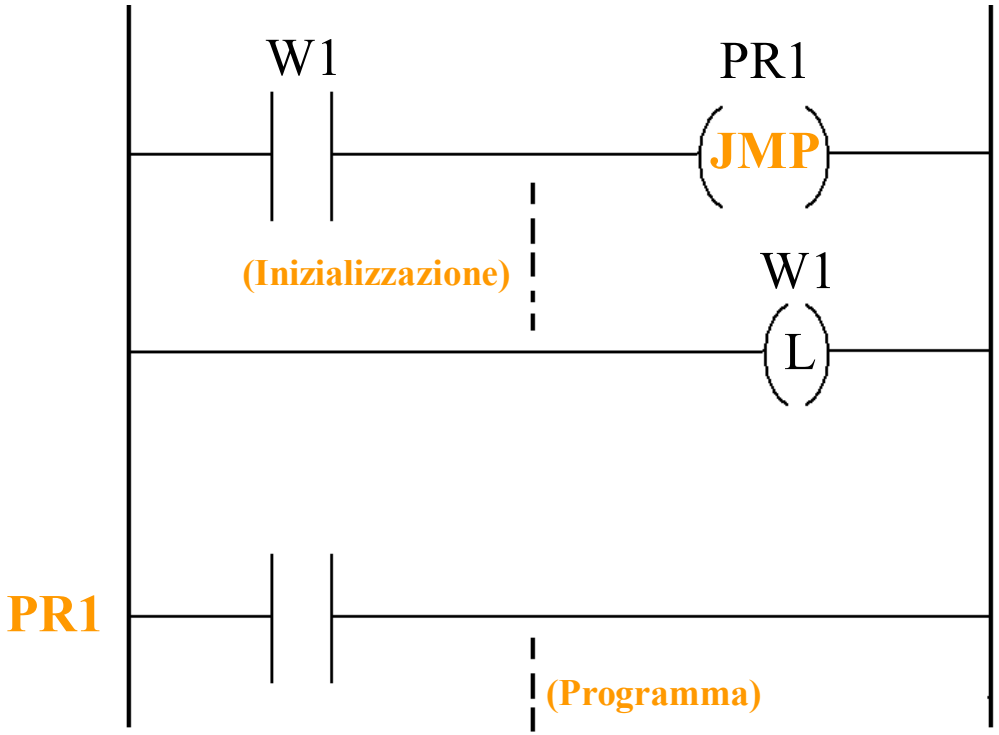
- **Reset contatore**
  - Riporta a zero il contatore Cx → RES

Istruzioni di Conteggio



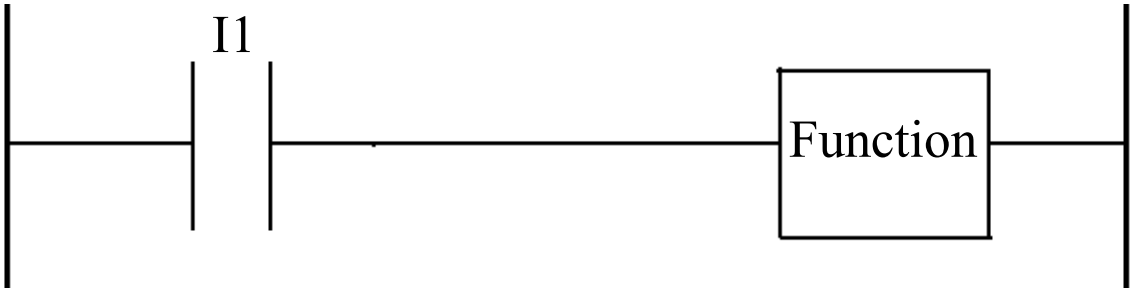
Controllo del Programma

- **Etichetta --|LBL|--**
  - Si usa per effetturare dei “salti” di programma
- **Salto --(JMP)--**
  - Se il piolo è abilitato il programma passa al piolo con l’etichetta indicata



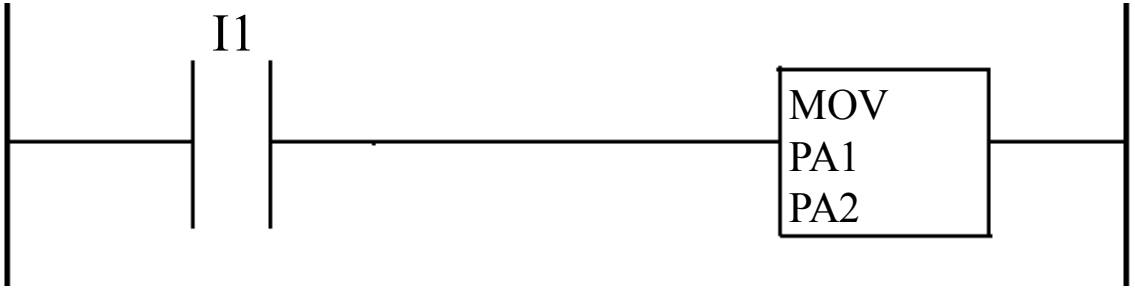
Funzioni e blocchi funzione

- **Il LD è di più di una pura rete elettrica.**
  - Funzioni avanzate e più espressive che si ispirano alle istruzioni di un comune linguaggio di programmazione di alto livello, come il C, il Pascal, ecc.
  - Le “istruzioni complesse” sono inglobate graficamente in LD in blocchi che vanno collegati con un piolo ed il loro uso è immediato



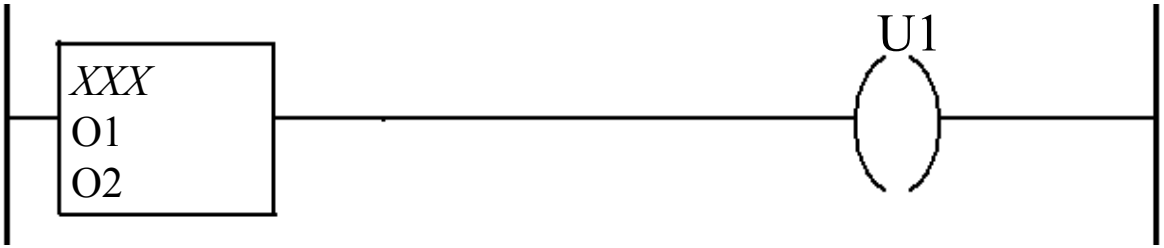
Trasferimento di memoria

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



Operazioni Aritmetico/Logiche

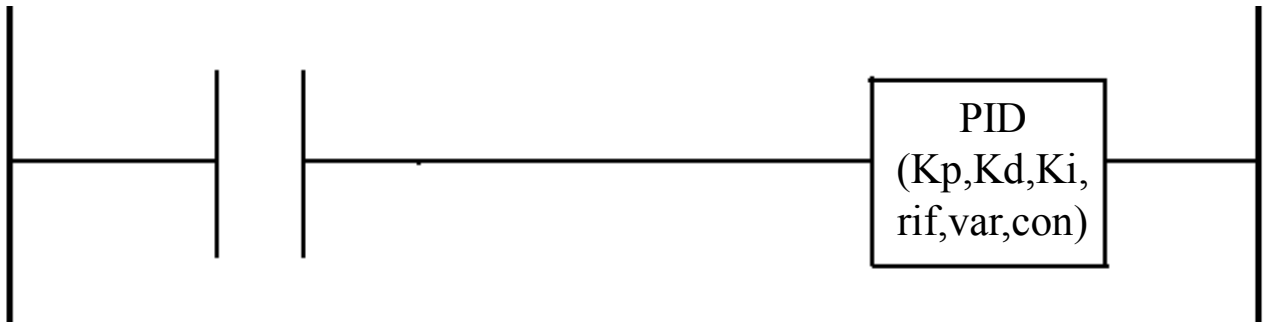
- Le istruzioni di comparazione fanno parte delle condizioni di attivazione dei pioli (EQU, NEQ, GEQ, LEQ, GRT, LES)



Funzioni e blocchi funzione

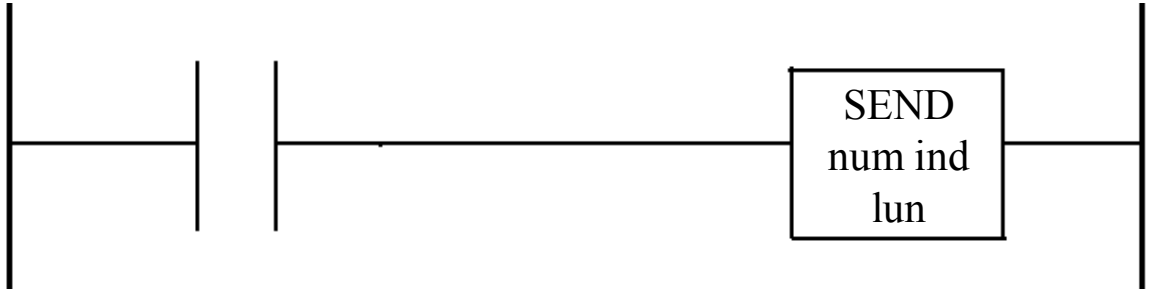
✓ Regolatore Proporzionale Integrale Derivativo

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



Funzioni e blocchi funzione

- **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