



Automazione industriale dispense del corso

16. Linguaggio a contatti (Ladder Diagram)

Luigi Piroddi
piroddi@elet.polimi.it

Introduzione

Il *linguaggio a contatti* (o *diagramma a scala*, dall'inglese *ladder diagram*, LD) è il più diffuso linguaggio di programmazione dei PLC.

E' uno standard di fatto del mercato americano.

Caratteristiche principali del LD:

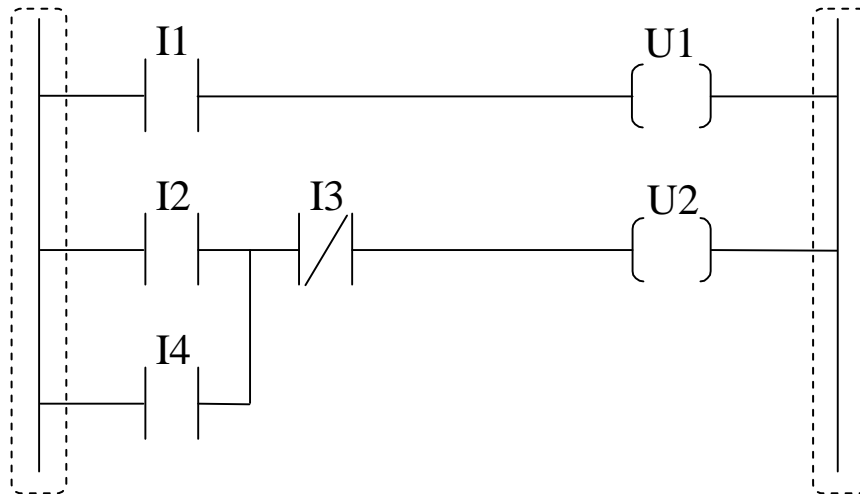
- ▶ è un linguaggio grafico
- ▶ si basa sulla trasposizione, in logica di programmazione, del funzionamento di una rete elettrica molto semplice, il cui obiettivo è di alimentare o non alimentare opportuni *utilizzatori elettrici (bobine)* tramite *interruttori* (chiamati anche *contatti* o *relé*)

La motivazione di questa trasposizione è di tipo storico:

- ▶ prima dei PLC si usavano batterie di relé elettromeccanici (v. quadri a relé)
- ▶ il linguaggio LD è nato per far accettare l'idea di “programmare” (e quindi l'uso del PLC) a chi era abituato a progettare sistemi di controllo logico con relé elettromeccanici

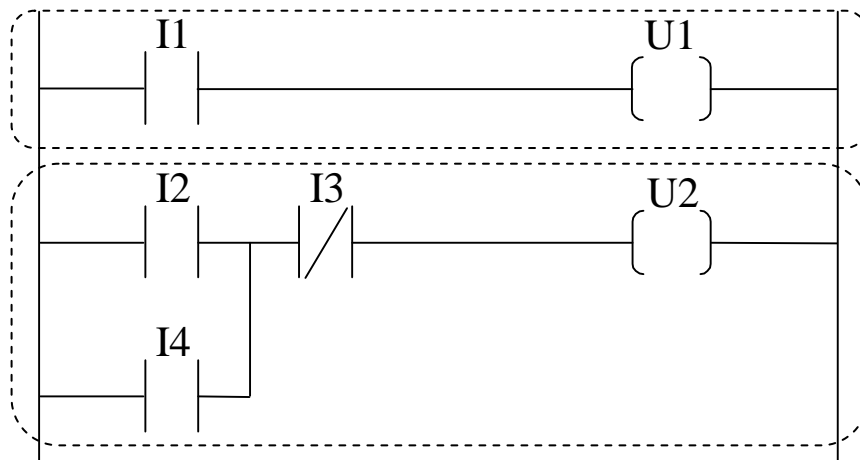
Elementi fondamentali

Gli elementi di base derivano dagli schemi di controllo a relé elettromeccanici:



► *montanti* della scala
linee verticali laterali che rappresentano l'alimentazione

- ▼ il montante di sinistra è il polo positivo collegato alla tensione V_{CC}
- ▼ il montante di destra è il polo negativo collegato a massa



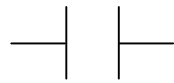
► *pioli* della scala (o *rung*)
linee orizzontali che contengono

- ▼ a sinistra *contatti* (elettrici) o *relé*
→ ingressi (o variabili interne)
- ▼ a destra *bobine*
→ uscite (o variabili interne)

Elementi di base: contatti

I *contatti* rappresentano i relé e possono essere associati agli ingressi digitali provenienti dal processo (o meglio al loro stato, rappresentato in particolari bit della memoria), oppure a condizioni interne al dispositivo.

contatto
normalmente
aperto



Se il bit associato vale 1, il contatto è chiuso e c'è continuità logica, altrimenti il contatto è aperto e non c'è continuità logica.

Può essere associato a

- bit di ingresso (Ix:y, bit y della word x)
- bit di uscita (Ux:y)
- bit associati a variabili interne (Wx:y)
- bit di stato di temporizzatori e contatori

Quasi tutti i sistemi consentono di dare ai bit anche dei nomi simbolici oltre ai nomi legati all'indirizzo fisico di memoria come quelli sopra, per migliorare la leggibilità dei programmi.

contatto
normalmente
chiuso


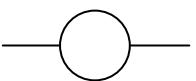
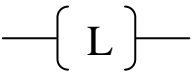





Il contatto è chiuso e assicura la continuità logica se la variabile booleana associata è falsa (bit a 0), altrimenti il contatto è aperto.

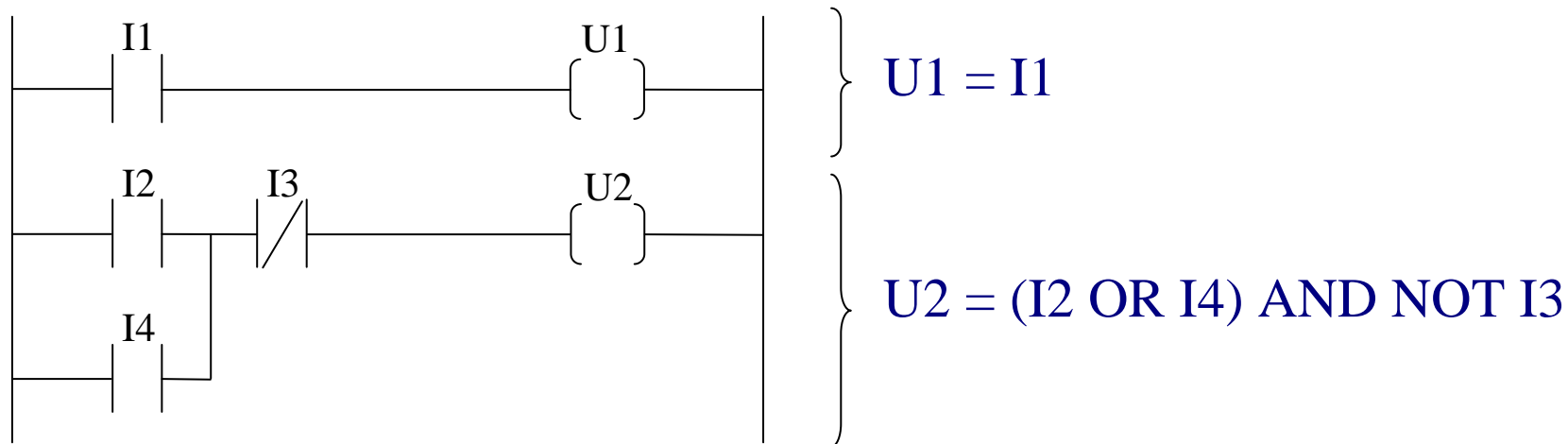
Un contatto associato ad un ingresso equivale ad un'istruzione di LOAD.

Elementi di base: bobine

Le bobine sono associate a bit di memoria, tramite i quali possono comandare uscite digitali oppure variare delle condizioni interne.

bobina	 oppure 	<p>La bobina rappresenta un generico utilizzatore elettrico, ad es. una resistenza, o una lampadina. Può essere associata a</p> <ul style="list-style-type: none"> ▪ bit di uscita (Ux:y) ▪ bit legati a variabili interne (Wx:y) <p>Va inserita sempre all'estremità destra del rung. La bobina si attiva quando passa corrente. Quindi, il bit associato sale al valore logico 1 (ON) se le condizioni logiche alla sua sinistra sono verificate, altrimenti vale 0 (OFF).</p>
bobina di tipo Latch (o Set)	 oppure 	<p>Quando si attiva, il bit associato va a 1 e mantiene tale valore finché non si attiva una bobina associata allo stesso bit.</p> <p>Il funzionamento è molto simile ad un flip-flop sollecitato con un impulso di durata finita sull'ingresso SET.</p>
bobina di tipo Unlatch (o Reset)	 oppure 	<p>Riporta allo stato logico 0 (OFF) un'uscita (v. ingresso RESET di un flip-flop).</p>

Esempio



Nel primo rung, se il contatto I1 è chiuso, la corrente può fluire dal polo positivo a quello negativo, attivando così la bobina U1.

In logica binaria, questo si traduce nell'espressione:

Se $I1 = 1$, allora $U1 = 1$, altrimenti $U1 = 0$,

dove

- ▶ alimentato = vero (1)
- ▶ non alimentato = falso (0)

Programmazione in LD

Per costruire un programma si dispongono uno dopo l'altro i pioli (le istruzioni).

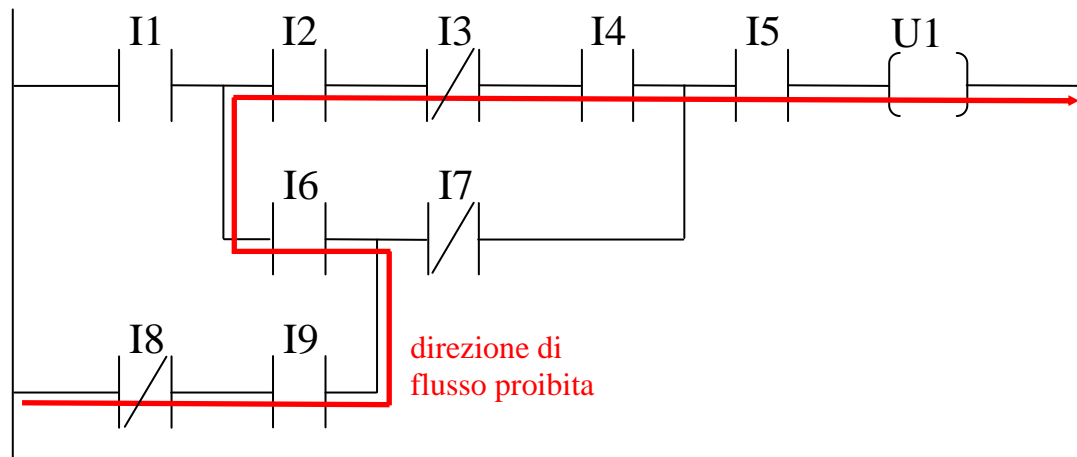
La corrispondenza di un programma LD (per definizione sequenziale) con una rete elettrica (per natura sede di fenomeni simultanei) non può essere esatta!

Occorre specificare:

- ❶ come vengono scanditi i pioli?
- ❷ quando vengono aggiornati ingressi e uscite?

Modalità di scansione dei pioli

I pioli vengono scanditi dall'alto verso il basso e da sinistra verso destra.
La “corrente” non può fluire da destra a sinistra come in una rete elettrica.



U1 si attiva se:

- ▶ $I1 = 1, I2 = 1, I3 = 0, I4 = 1, I5 = 1$
- ▶ $I1 = 1, I6 = 1, I7 = 0, I5 = 1$
- ▶ $I8 = 0, I9 = 1, I7 = 0, I5 = 1$

Se $I1 = 0$ e $I7 = 1$, la bobina non si attiva lungo il cammino I8-I9-I6-I2-I3-I4-I5 neanche se $I8 = 0, I9 = 1, I6 = 1, I2 = 1, I3 = 0, I4 = 1, I5 = 1$. Infatti, il contatto I6 sarebbe percorso in direzione contraria a quella consentita per convenzione.

Lo schema è equivalente all'istruzione logica seguente:

$$U1 = I5 \text{ AND } [(I4 \text{ AND NOT}(I3) \text{ AND } I2 \text{ AND } I1) \text{ OR } (\text{NOT}(I7) \text{ AND } ((I6 \text{ AND } I1) \text{ OR } (I9 \text{ AND NOT}(I8))))]$$

Sincronizzazione I/O

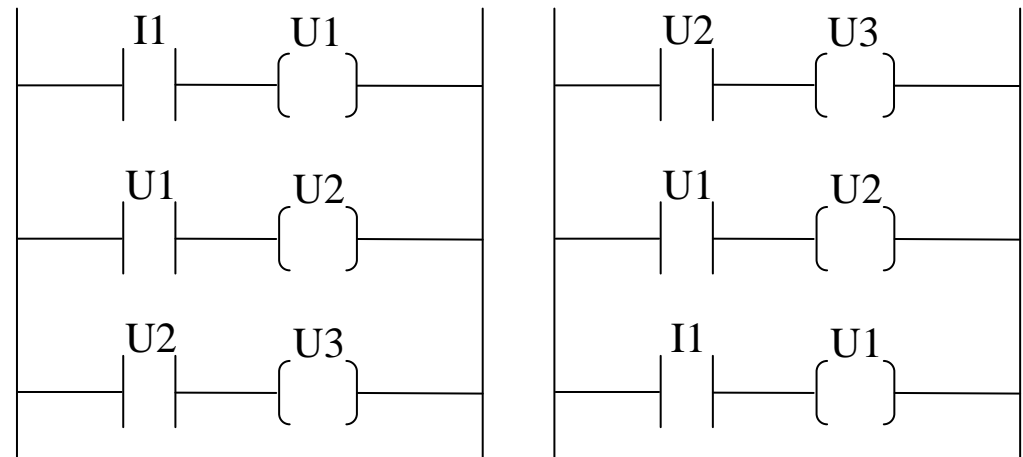
La sincronizzazione con gli ingressi e le uscite avviene secondo il ciclo di copia massiva:

- ▶ ogni piolo viene scandito in ogni ciclo di scansione (a meno di istruzioni di salto)
- ▶ le uscite associate alle bobine normali vengono scritte *ad ogni ciclo*, e il loro valore permane fino alla prossima esecuzione (ciclo successivo) della stessa istruzione
- ▶ il valore delle variabili lette in ingresso rimane costante per tutto il ciclo

I due programmi raffigurati a lato sono diversi.

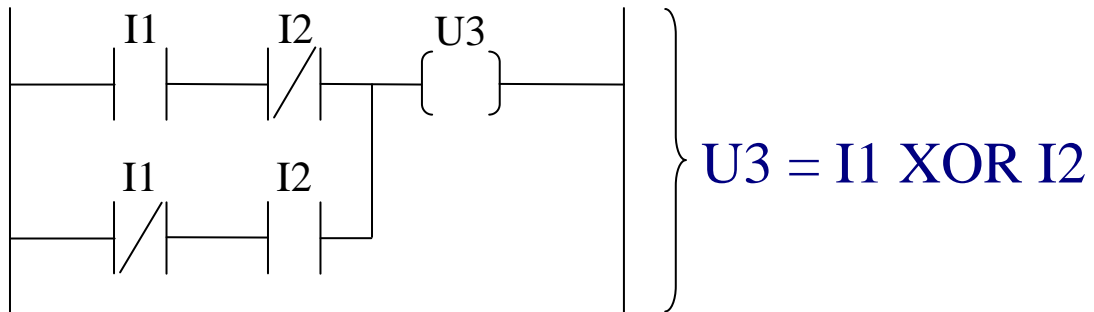
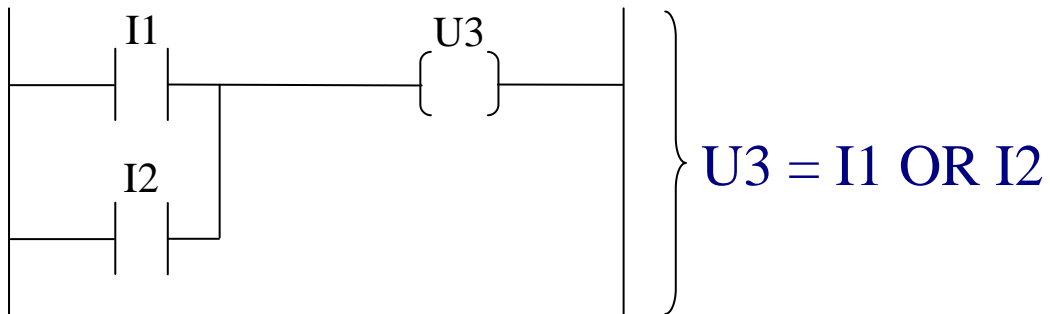
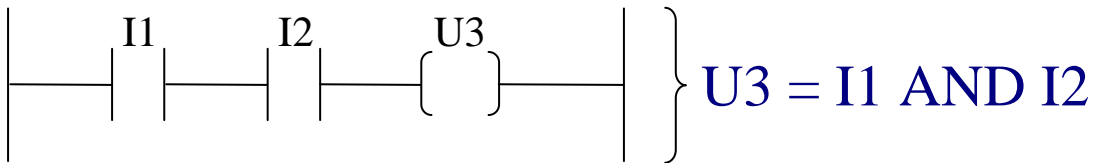
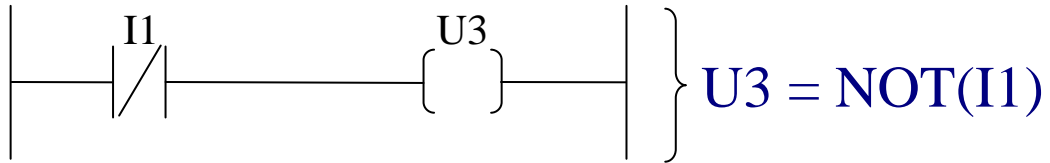
Invertendo l'ordine dei pioli, cambia il risultato dell'esecuzione delle istruzioni.

L'ordine dei rung è rilevante!



LD è un linguaggio che descrive il ciclo operativo del PLC: in un programma LD è scritta la sequenza delle cose che il PLC deve fare ad ogni ciclo.

Funzioni logiche base (statiche)



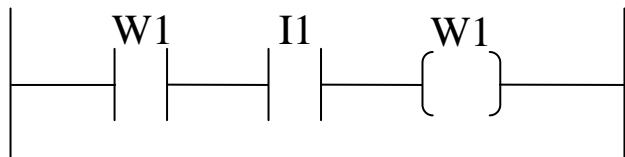
Elementi dinamici

Le istruzioni di base del LD sono “statiche”:

- ▶ assegnano il valore di un’uscita in base ad una pura combinazione logica delle variabili di ingresso associate ai contatti

E’ possibile però definire anche delle *variabili dinamiche* (variabili di stato):

- ▶ sono associate sia a contatti che a bobine

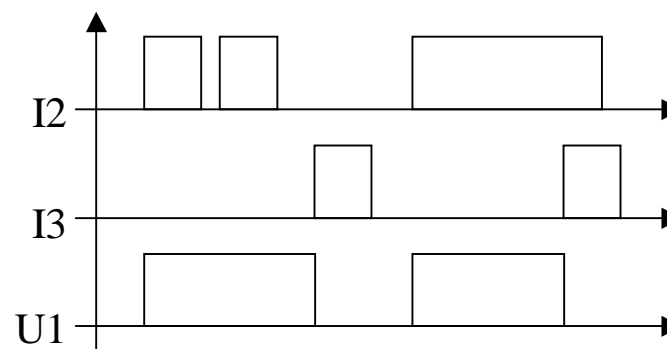
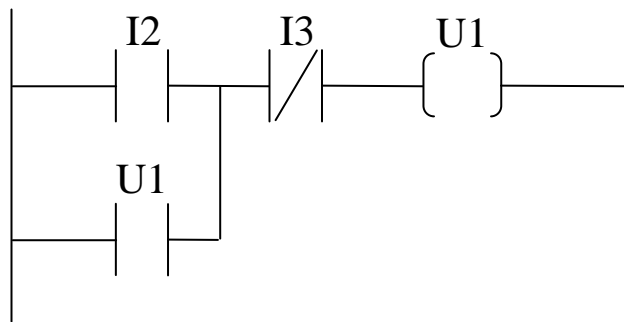
- ▶ esempio:  $W1(k) := I1(k) \text{ AND } W1(k-1)$

Esempi di elementi dinamici:

- ▶ circuito bistabile o flip-flop a reset vincente
- ▶ flip-flop a set vincente
- ▶ riconoscitore di fronte di salita/discesa
- ▶ flip-flop di tipo D

Circuito bistabile o flip-flop a reset vincente

Obiettivo: l'ingresso I2 (set) deve attivare l'uscita, e I3 (reset) disattivarla; qualora entrambi gli ingressi I2 e I3 siano attivi, l'uscita deve essere resettata a 0 (“vince” I3).

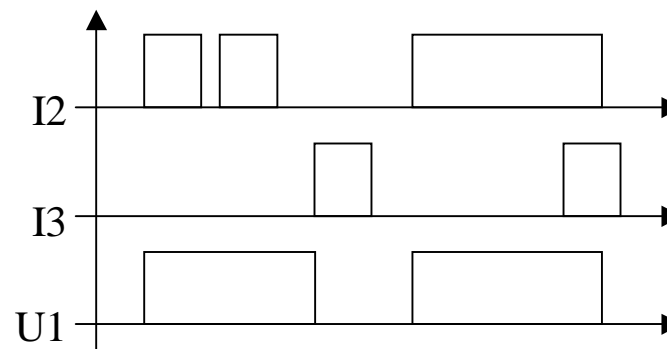
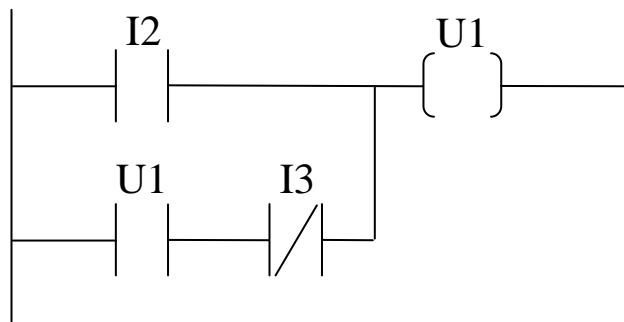


Funzionamento dinamico:

- ❶ Inizialmente tutte le variabili sono al valore 0.
- ❶ Quando sono verificate sia $I2 = 1$ che $I3 = 0$, U1 commuta a 1.
- ❷ U1 rimane a 1 finché I3 è nullo (U1 si auto-alimenta, grazie al contatto U1).
- ❸ Un impulso di durata finita su I3 pone U1 a 0.

Flip-flop a set vincente

Obiettivo: l'ingresso I2 (set) deve attivare l'uscita, e I3 (reset) disattivarla; qualora entrambi gli ingressi I2 e I3 siano attivi, l'uscita deve essere settata a 1 (“vince” I2).

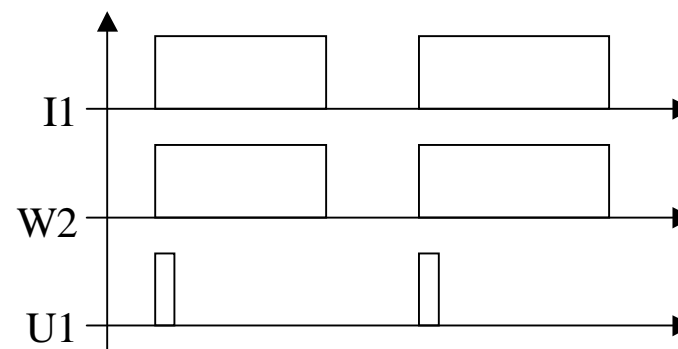
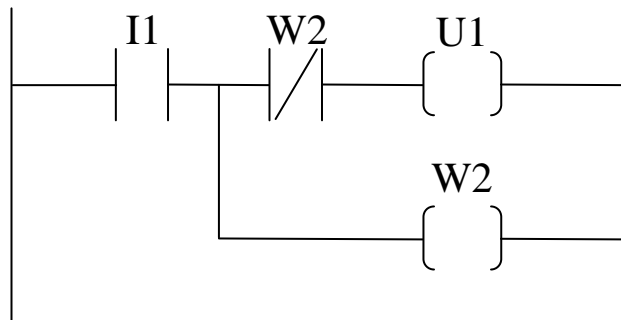


Funzionamento dinamico:

- ❶ Inizialmente tutte le variabili sono al valore 0.
- ❷ Il piolo assegna il valore 1 a U1 se $I2 = 1$.
- ❸ Finché $I3 = 0$, U1 rimane a 1 (si autoalimenta).
- ❹ U1 viene resettata se sia $I3 = 1$ che $I2 = 0$.

Riconoscitore di un fronte di salita

Obiettivo: generare un impulso della durata di un tempo di ciclo in corrispondenza di un fronte di salita su I1.



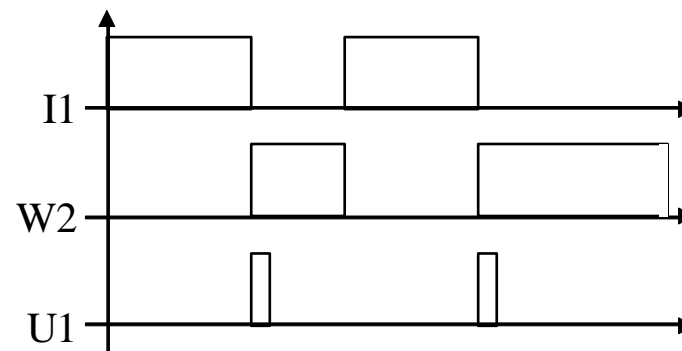
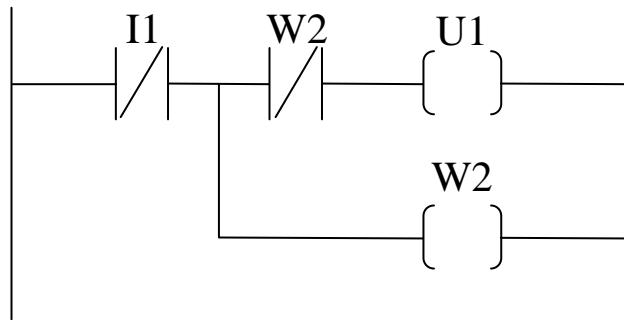
Funzionamento dinamico:

- ❶ Inizialmente tutte le variabili sono al valore 0.
- ❷ Quando si ha un fronte di salita su I1, viene alimentata la bobina U1.
Nello stesso ciclo viene alimentata anche W2.
- ❸ Al ciclo successivo, essendo $W2 = 1$, viene interrotta l'alimentazione su U1 (indipendentemente dal valore di I1).

Usando la variabile ausiliaria W2 siamo riusciti a creare un impulso di durata pari ad un tempo di ciclo tutte le volte che si verifica un fronte di salita su I1.

Riconoscitore di un fronte di discesa

Obiettivo: generare un impulso della durata di un tempo di ciclo in corrispondenza di un fronte di discesa su I1.

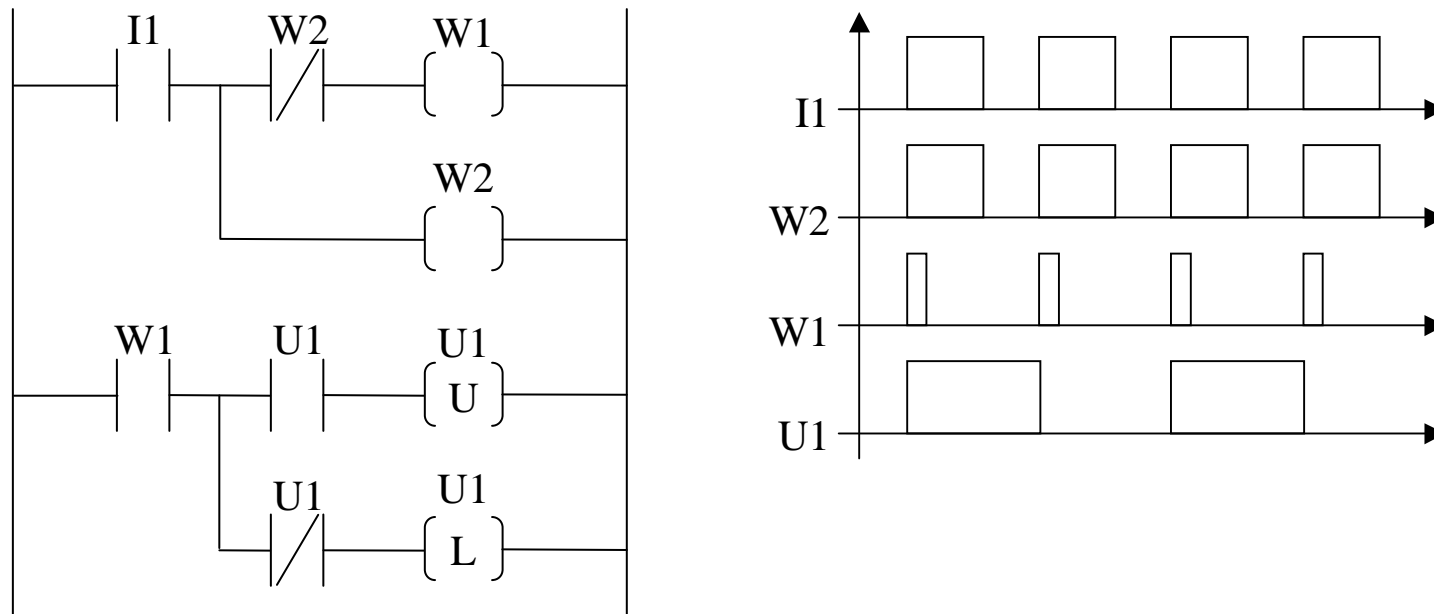


Funzionamento dinamico:

- ① Si supponga che I1 sia inizializzato a 1 (le altre variabili sono inizialmente a 0).
- ② Quando si ha un fronte di discesa su I1, viene alimentata la bobina U1. Nello stesso ciclo viene alimentata anche W2.
- ③ Al ciclo successivo, essendo $W2 = 1$, viene interrotta l'alimentazione su U1.
- ④ Quando I1 torna a 1, la bobina W2 non è più alimentata e si ripristinano le condizioni iniziali.

Flip-flop di tipo D

Obiettivo: far commutare l'uscita su ogni impulso in ingresso.

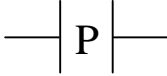

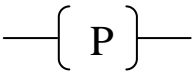
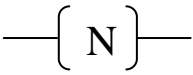


Si può sfruttare il riconoscitore di un fronte di salita appena visto (v. parte superiore del circuito) per settare la variabile interna **W1** a 1 in corrispondenza di un fronte di salita sull'ingresso **I1** (**W1** rimane a 1 per un tempo di ciclo).

Il secondo rung commuta **U1** quando **W1** = 1 (se **U1** vale 0 è attiva la bobina di tipo latch, mentre se **U1** vale 1 è attiva la bobina di tipo unlatch).

Riconoscitori di fronti: istruzioni complesse

Alcuni ambienti CAD di programmazione di PLC prevedono direttamente l'uso di istruzioni complesse di riconoscimento di fronti, per semplificare la programmazione:

contatto di tipo P (positive edge)		Il contatto si chiude sul fronte di salita della variabile booleana associata (ovvero se la variabile passa, tra due successive esecuzioni della stessa istruzione, da falso a vero).
contatto di tipo N (negative edge)		Il contatto si chiude sul fronte di discesa della variabile associata.
bobina di tipo P		La bobina assegna il valore 1 alla variabile booleana associata solo quando la sua alimentazione, tra due esecuzioni successive, passa da assente a presente.
bobina di tipo N		La bobina assegna il valore 1 alla variabile booleana associata solo quando la sua alimentazione passa da presente ad assente.