

Il Linguaggio di Programmazione IEC 61131-3

Ladder

Linguaggio di Programmazione Ladder

- ❖ E' il più vecchio linguaggio di programmazione per PLC
- ❖ Si basa su simboli di provenienza "elettrica":
 - binari di potenza (power rail), contatti elettrici e avvolgimenti magnetici (coil)
- ❖ Si articola in linee verticali dette “rung”
- ❖ Ciascun “rung” può contenere contatti, coil, Function Block e Funzioni
- ❖ Ciascun “rung” deve essere connesso necessariamente al binario di potenza sinistro (left power rail), mentre il collegamento con quello destro è opzionale

Elementi di Base del Linguaggio Ladder

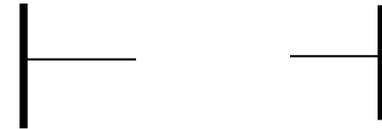
❖ Power Rail



❖ Linee Elettriche Orizzontali



❖ Connessioni ai Power Rail



❖ Contatto Normalmente Aperto



❖ Contatto Normalmente Chiuso



❖ Coil

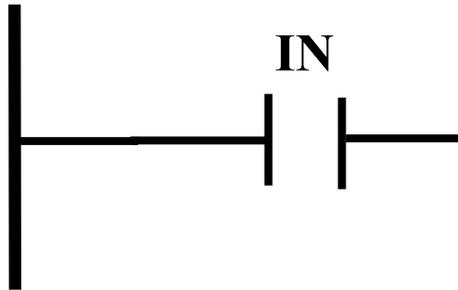


❖ Negated Coil

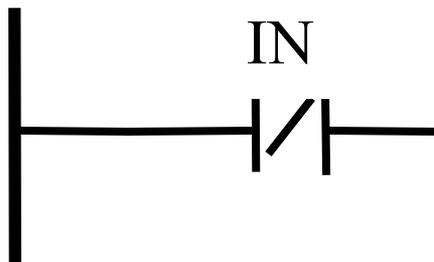


Utilizzo dei Contatti

- ❖ 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. La corrente fluisce a destra **per qualunque scansione del Programma Ladder** fino a quando la variabile IN diviene 0

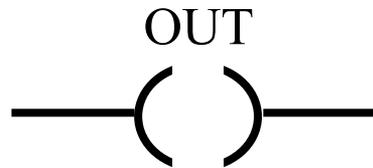


- ❖ **Contatto Normalmente Chiuso:** la corrente fluisce da sinistra a destra se la variabile IN è 0. La corrente fluisce a destra **per qualunque scansione del Diagramma Ladder** fino a quando la variabile IN diviene 1

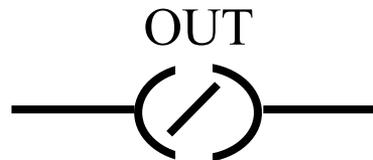


Utilizzo dei Coil

- ❖ 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. La variabile rimane a 1 **per qualunque scansione del Programma Ladder** fino a quando la corrente cessa di fluire da sinistra.

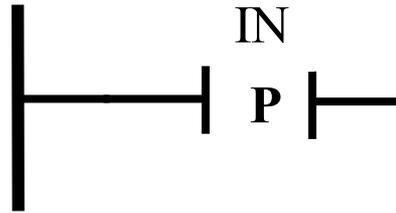


- ❖ **Negated Coil:** la variabile OUT associata al **Negated Coil** è posta a 0 se vi è una corrente che fluisce da sinistra. La variabile rimane a 0 **per qualunque scansione del Programma Ladder** fino a quando la corrente cessa di fluire da sinistra.



Altri Contatti del Linguaggio Ladder

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

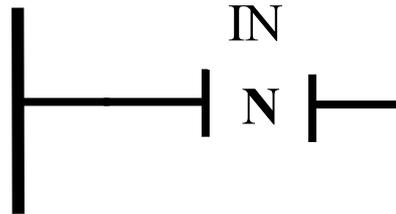


- ❖ La corrente fluisce da sinistra a destra del **Positive Transition-Sensing Contact**, se la variabile IN passa da 0 a 1. La corrente fluisce a destra **solo per una scansione del Programma Ladder** (quella relativa alla transizione).

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

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

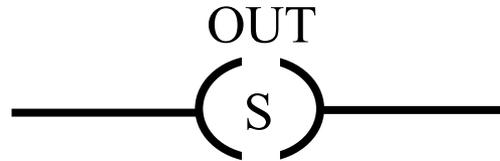


- ❖ La corrente fluisce da sinistra a destra del **Negative Transition-Sensing Contact**, se la variabile IN passa da 1 a 0. La corrente fluisce a destra **solo per una scansione del Diagramma Ladder** (quella relativa alla transizione)

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

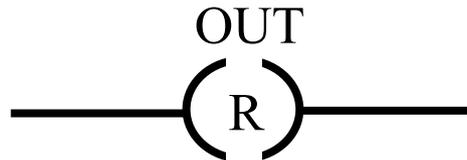
Altri Coil del Linguaggio Ladder

❖ 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 per qualunque scansione del Diagramma Ladder 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 per qualunque scansione del Diagramma Ladder fino a quando viene utilizzato un coil SET.

Altri Coil del Linguaggio Ladder

❖ Positive Transition-Sensing Coil

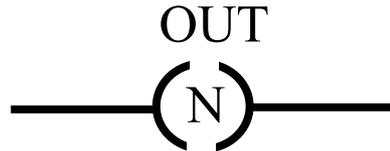


- ❖ La variabile OUT associata al **Positive Transition-Sensing Coil** è posta a 1 se la corrente che fluisce da sinistra passa da un valore FALSE ad un valore TRUE. La variabile rimane a 1 solo per una scansione del Diagramma Ladder (quella relativa alla transizione dello stato della corrente).

Scansione	Valore della corrente alla sinistra del coil quando viene valutato il rung	Valore di OUT
1	OFF	OFF
2	ON	ON
3	ON	OFF
4	OFF	OFF

Altri Coil del Linguaggio Ladder

❖ Negative Transition-Sensing Coil



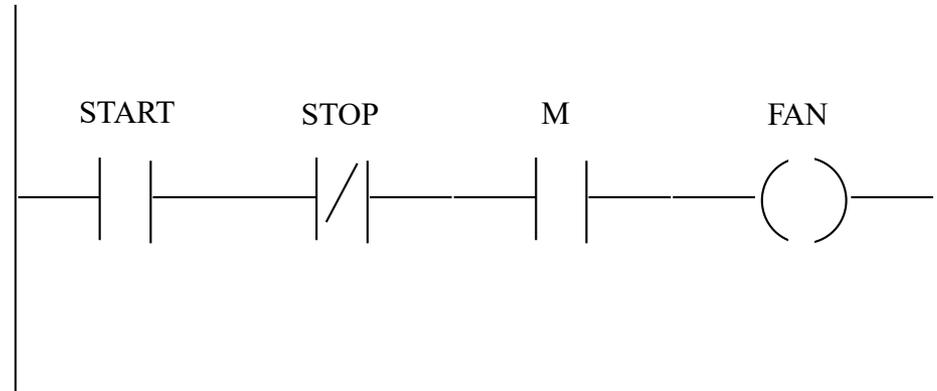
- ❖ La variabile OUT associata al **Negative Transition-Sensing Coil** è posta a 1 se la corrente che fluisce da sinistra passa da un valore TRUE ad un valore FALSE. La variabile rimane a 1 solo per una scansione del Diagramma Ladder (quella relativa alla transizione dello stato della corrente).

Scansione	Valore della corrente alla sinistra del coil quando viene valutato il rung	Valore di OUT
1	ON	OFF
2	OFF	ON
3	OFF	OFF
4	ON	OFF

Logiche di Base Realizzabili con gli Elementi del Linguaggio Ladder

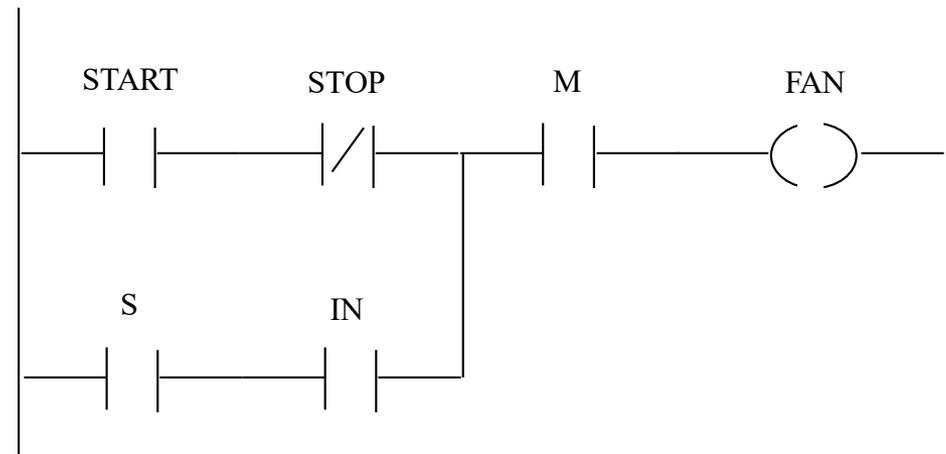
❖ Logica AND

if (START &&!STOP &&M) FAN=1



❖ Logica OR

if ((START &&!STOP ||
S && IN) && M) FAN=1



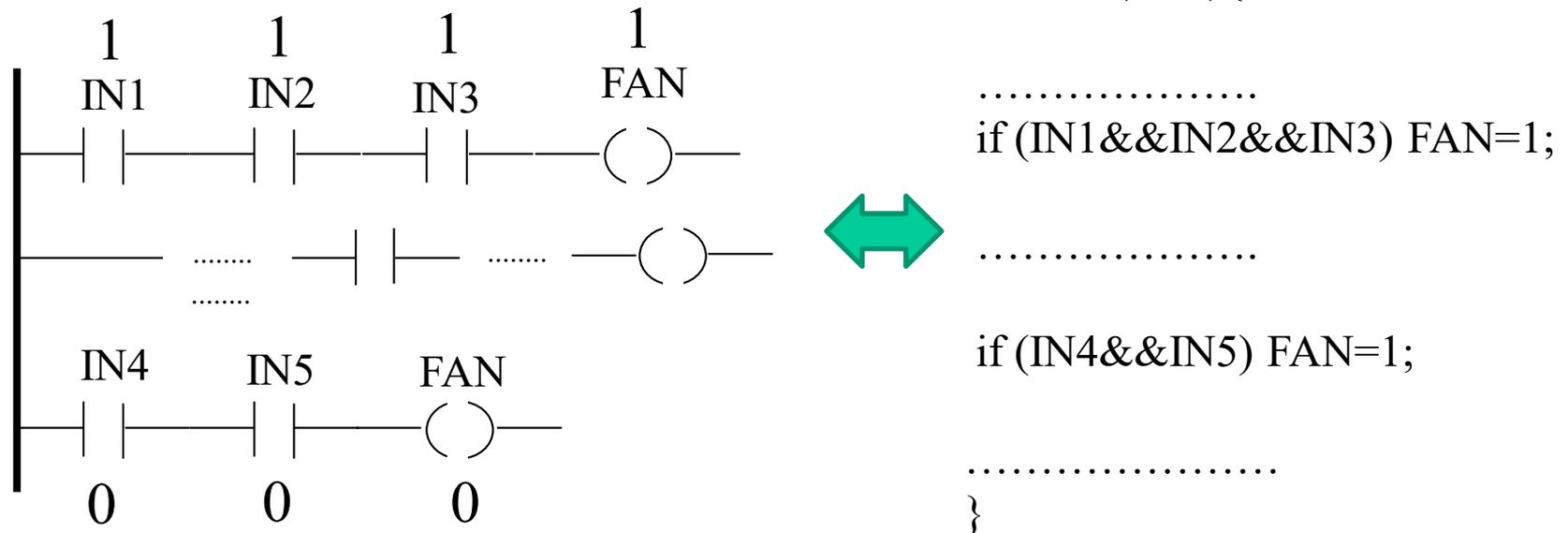
Regole di Esecuzione dei Rung

- ❖ Un programma scritto in linguaggio Ladder viene eseguito valutando un rung alla volta (da **Sinistra** a **Destra**)
- ❖ L'ordine di valutazione dei rung è quello che procede dal primo rung in alto verso l'ultimo rung in basso

Effetti Collaterali delle Regole di Esecuzione dei Rung

- ❖ L'ordine di valutazione comporta la necessità di riunificare i rung che operano delle modifiche (Write) sulle stesse **uscite** (Attenzione: vale solo per i coil "normali")

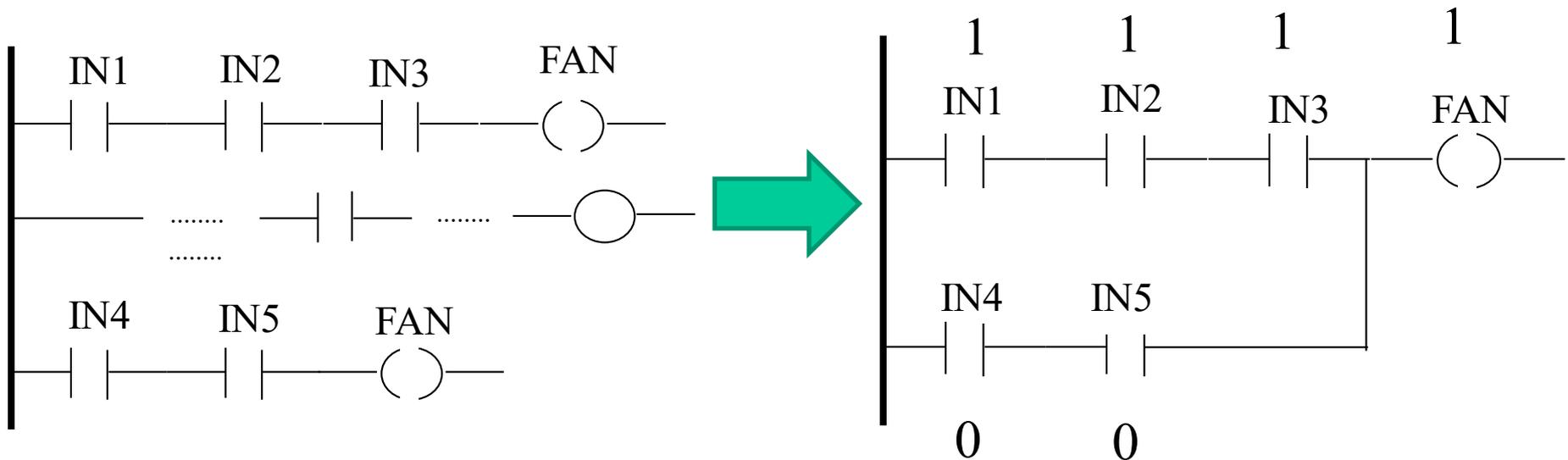
- ❖ Esempio:



L'effetto di IN1, IN2, e IN3 sull'**uscita** reale collegata alla variabile FAN è nullo. L'uscita reale può essere modificata solo da IN4 e IN5, a causa della posizione del rung che li contiene

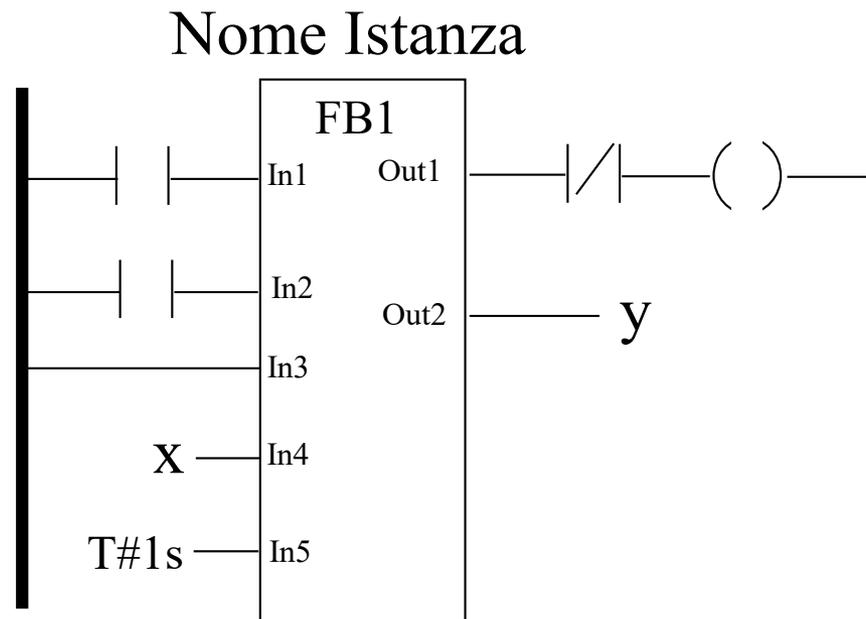
Effetti Collaterali delle Regole di Esecuzione dei Rung

- ❖ 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



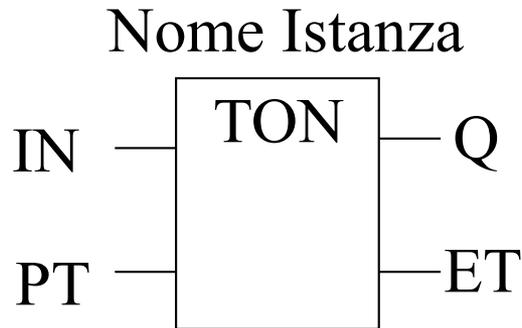
Uso di Istanze di Function Block e di Funzioni

- ❖ E' possibile connettere contatti con gli ingressi dell'istanza del FB o della funzione, purché essi siano binari
- ❖ E' possibile connettere coil con le uscite dell'istanza del FB o della funzione, purché esse siano binarie
- ❖ Nel caso in cui una Istanza di FB o una funzione richieda un ingresso binario sempre TRUE, è possibile collegare tale ingresso direttamente al power rail di sinistra
- ❖ Eventuali variabili analogiche o valori analogici (interi, reali, temporali, etc.) possono essere connessi direttamente ai corrispondenti ingressi dell'istanza del FB o della funzione

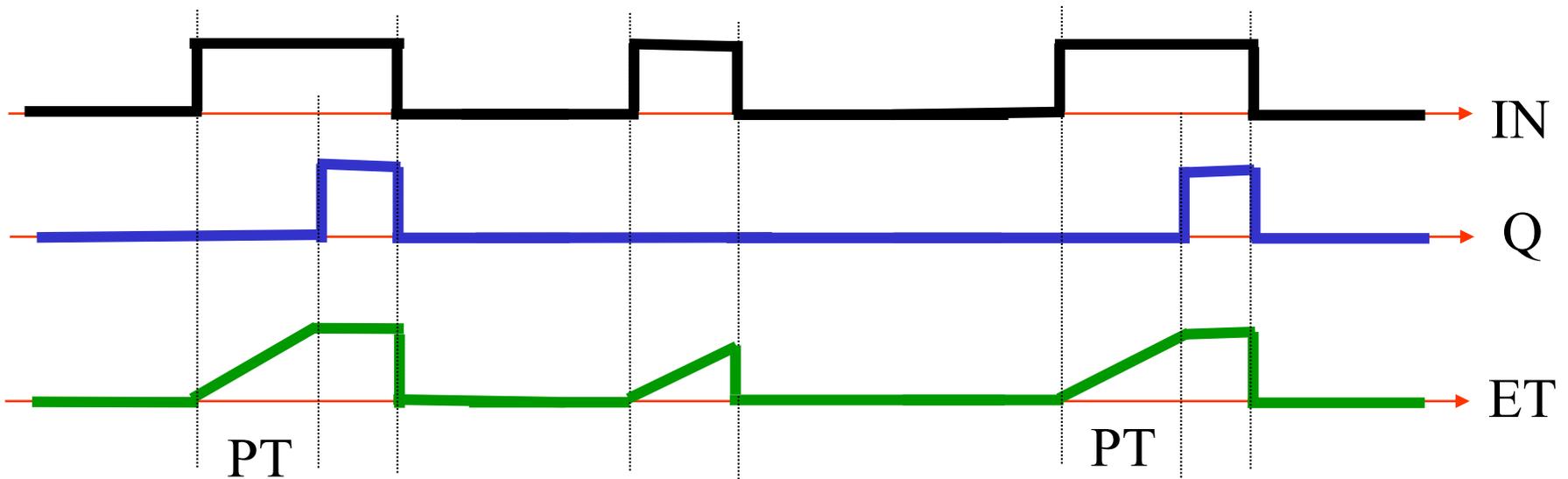


Function Block di Uso Comune

Timer TON

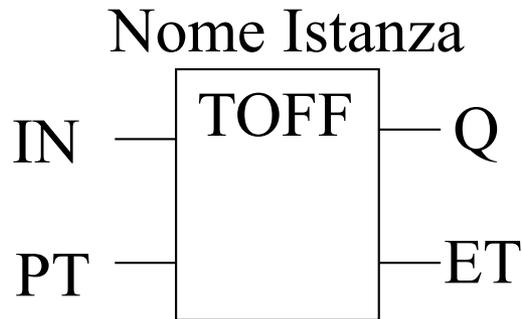


- ❖ IN: BOOL. If a rising edge is detected, the on-delay timing is started
- ❖ PT: TIME. Preset time interval for the delay
- ❖ Q: BOOL. Output
- ❖ ET: TIME. Elapsed time interval

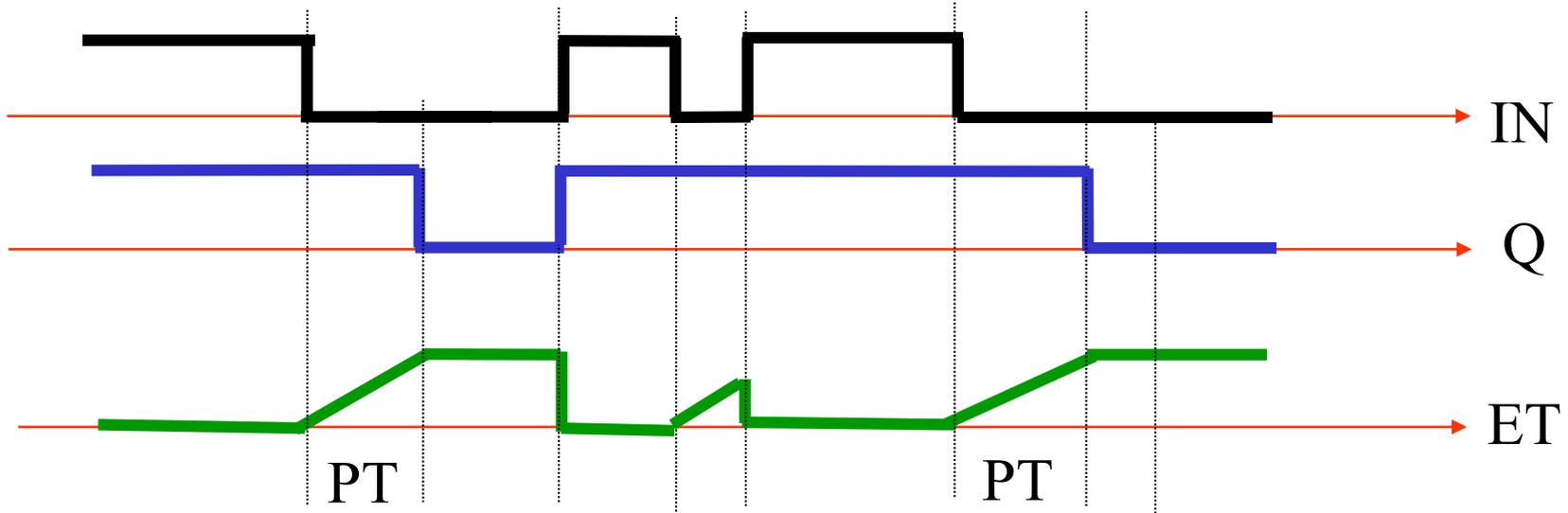


Function Block di Uso Comune

Timer TOFF

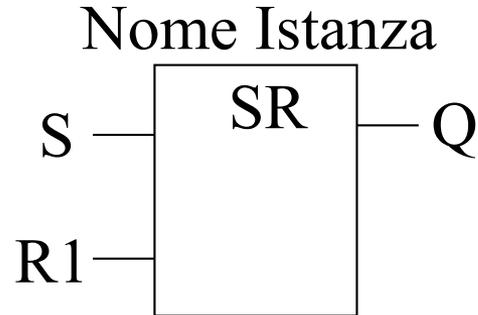


- ❖ IN: BOOL. If a falling edge is detected, the off-delay timing is started.
- ❖ PT: TIME. Preset time interval for the delay
- ❖ Q: BOOL. Output
- ❖ ET: TIME. Elapsed time interval

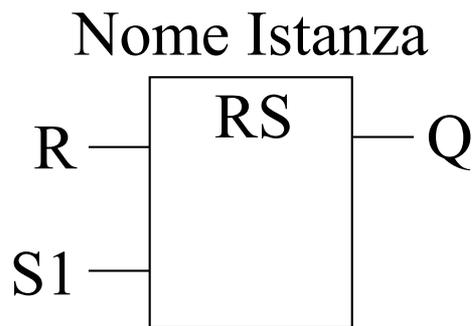


Function Block di Uso Comune

Bistabili SR e RS



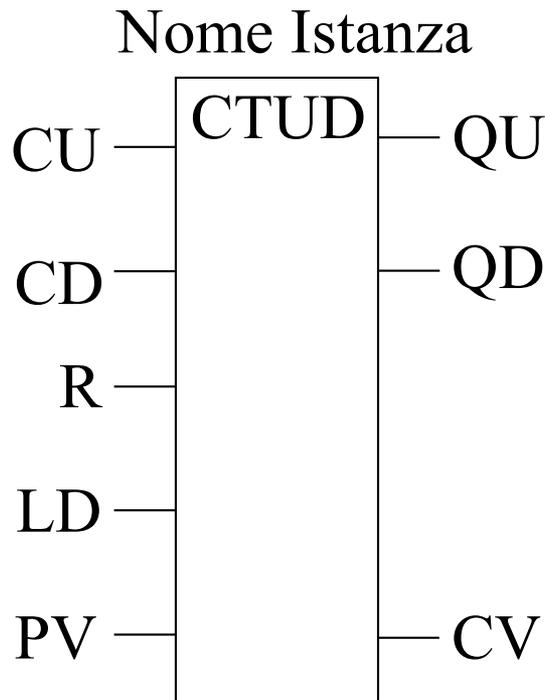
- ❖ S, R1, Q: BOOL
- ❖ Q viene messa a 1 se:
 - ✓ S ha lo stato di segnale "1".
 - ✓ R1 ha lo stato di segnale "0".
- ❖ Q viene resettata se è soddisfatta una delle seguenti condizioni:
 - ✓ S ha lo stato di segnale "0" e R1 ha lo stato di segnale "1".
 - ✓ S e R1 hanno lo stato di segnale "1".



- ❖ S, R1, Q: BOOL
- ❖ Q viene messa a 0 se:
 - ✓ R ha lo stato di segnale "1".
 - ✓ S1 ha lo stato di segnale "0".
- ❖ Q viene messa ad 1 se è soddisfatta una di queste due condizioni:
 - ✓ R ha lo stato di segnale "0" e S1 ha lo stato di segnale "1".
 - ✓ R e S1 hanno lo stato di segnale "1".

Function Block di Uso Comune

Contatore Up-Down (CTUD)



- ❖ CU: BOOL. Se si verifica un **fronte di salita**, il contatore viene incrementato.
- ❖ CD: BOOL. Se si verifica un **fronte di salita**, il contatore viene decrementato.
- ❖ R: BOOL. Se VERO il contatore viene resettato (0)
- ❖ PV: **INT**. Valore iniziale che può assumere il contatore
- ❖ LD: BOOL. Se VERO il contatore viene posto al valore iniziale PV
- ❖ QU: BOOL. E' posta a TRUE finché il contatore attuale è maggiore o uguale al valore dell'ingresso PV. In tutti gli altri casi l'uscita QU ha lo stato di segnale "0".
- ❖ QD: BOOL. E' posta a 1 finché il valore di conteggio attuale è minore o uguale a zero. In tutti gli altri casi l'uscita QD ha lo stato di segnale "0".
- ❖ CV: INT. Valore Corrente del contatore

Function Block di Uso Comune

❖ Bistabili

➤ SR, RS

❖ Bitwise Boolean

➤ AND, OR, NOT, XOR

❖ Comparison

➤ EQ, LE, LT, GE, GT, NE

❖ Counters

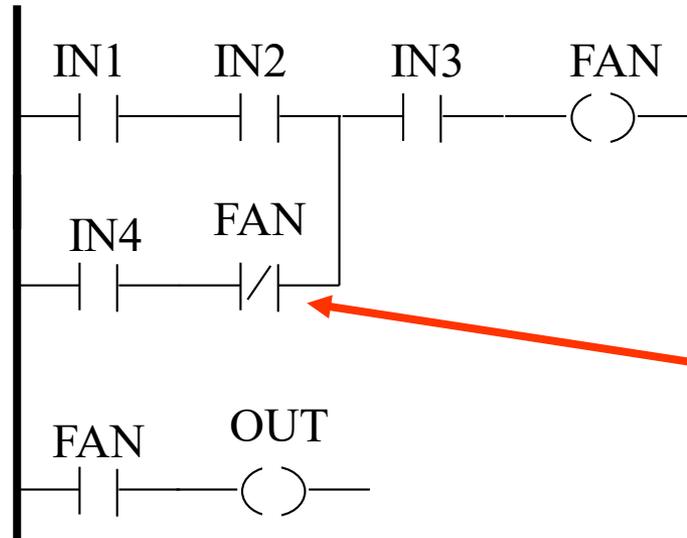
➤ CTD (down), CTU (up)

❖ Altri (disponibili su alcuni PLC)

➤ PID

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 con il Ladder

- ❖ Tecnica basata su Relazioni I/O
 - La tecnica di programmazione più istintiva e naturale che è possibile applicare al linguaggio Ladder.
- ❖ Tecnica basata sulla Macchina a Stati
 - Deve essere applicata a problemi più complessi, in cui l'attivazione delle uscite non dipende esclusivamente dagli ingressi (o da variabili interne, quali bits, contatori, timers) ma è legata al concetto di "stato".

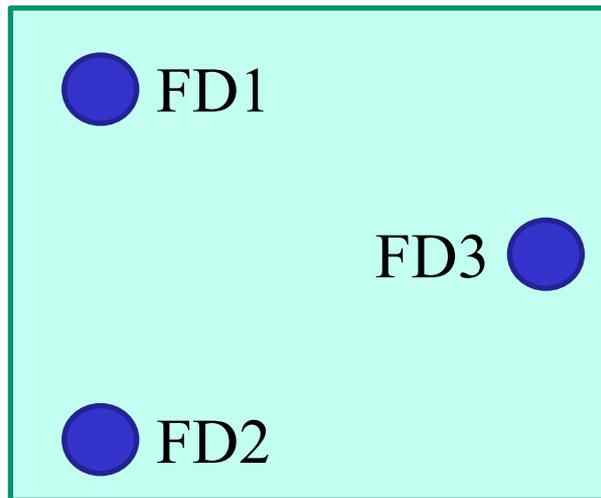
Tecnica basata su Relazioni I/O

- ❖ Consiste nell'esprimere le **relazioni causa effetto**, che legano le **uscite** da comandare agli **ingressi** o a particolari **condizioni logiche interne**
- ❖ La scrittura di un programma in Ladder deve prevedere:
 - l'esplicitazione delle **relazioni causa effetto**
 - la loro traduzione utilizzando i simboli del linguaggio Ladder.
- ❖ Nel seguito verranno mostrati degli esempi che permettono di comprendere meglio quanto detto.
- ❖ Importante: Bisogna ricordarsi di riunificare le uscite (vedi lucido relativo)

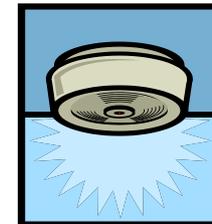
Tecnica basata su Relazioni I/O

Gestione Allarme Antincendio

- ❖ Dispositivo Antincendio con 3 Sensori: FD1, FD2, FD3
- ❖ Attivazione Manuale dell'Allarme
- ❖ Gestione Rientro Allarme (Reset): allarme continua se i sensori segnalano
- ❖ Gestione Malfunzionamenti (Voting 2 Sensori su 3)
- ❖ Indicatore dello stato dei Sensori



Indicatore
Luminoso



Allarme

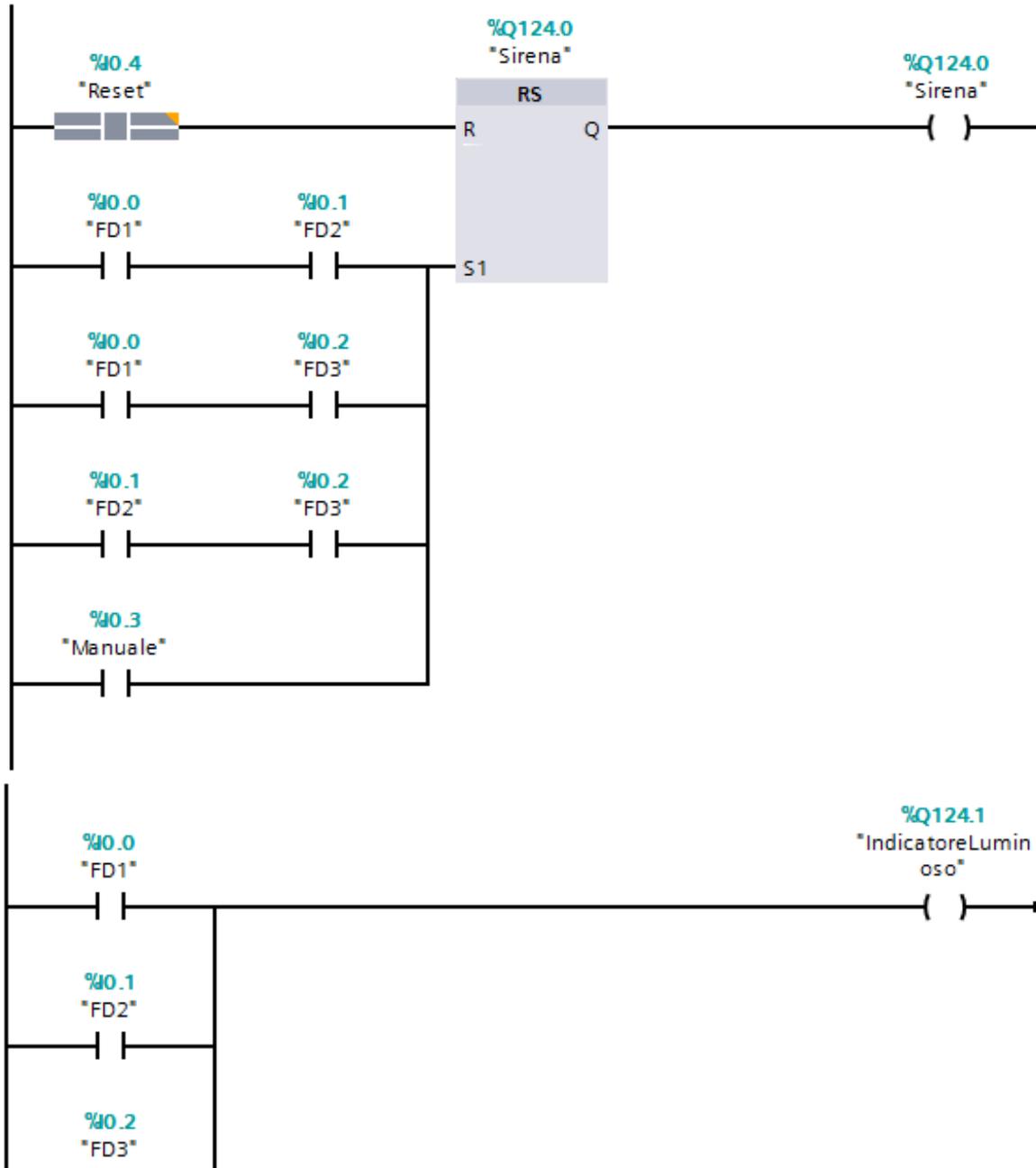


Avvio
Manuale



Reset
Allarme

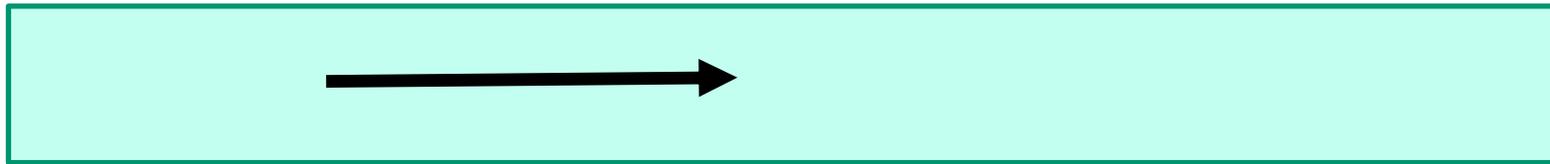
Gestione Allarme Antincendio



Tecnica basata su Relazioni I/O

Gestione Nastro Trasportatore

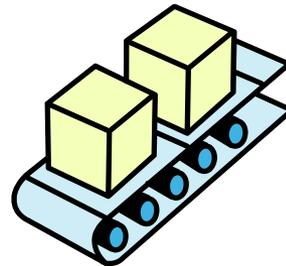
- ❖ Accensione/Spegnimento Automatico Nastro (Acceso se il numero persone/pezzi presenti > 0)



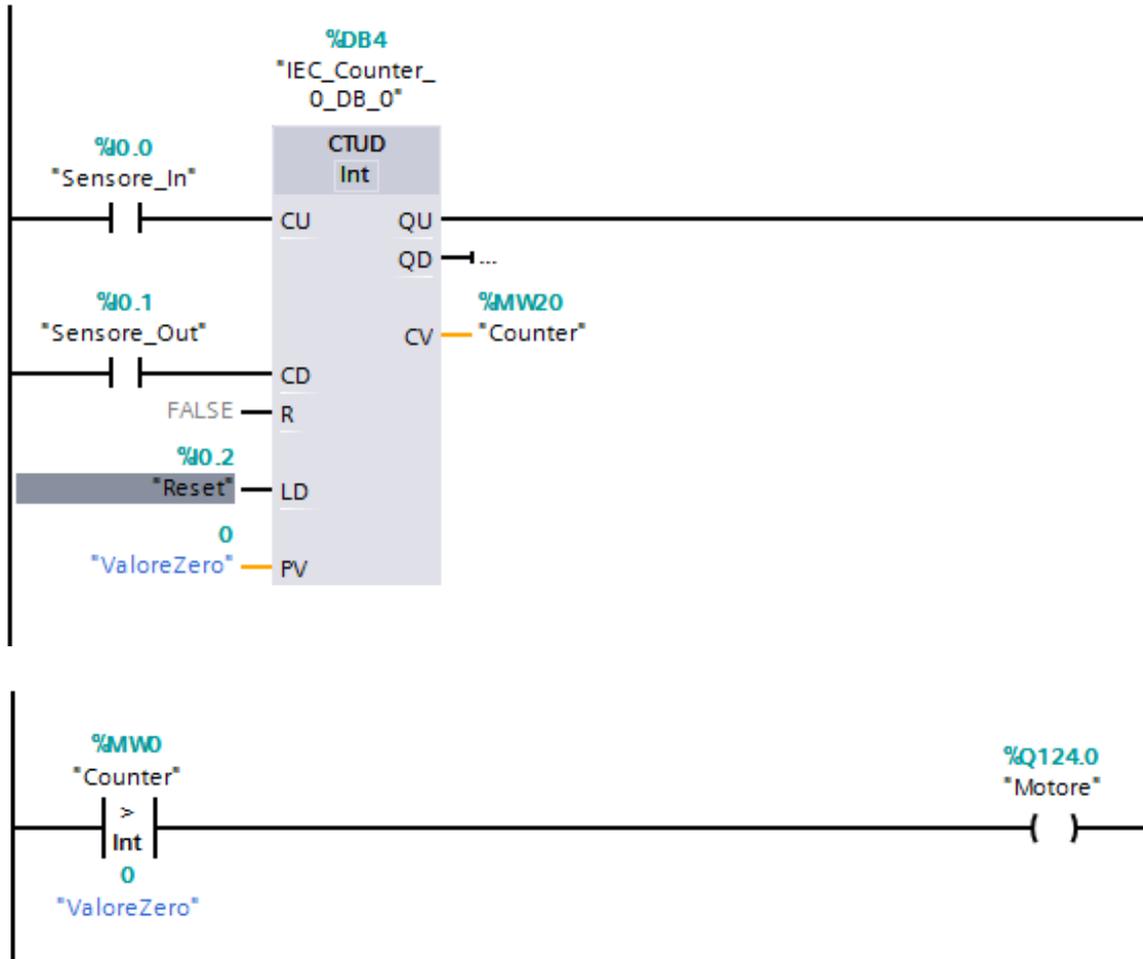
● Sensore
Ingresso

Sensore ●
Uscita

Accensione/
Spegnimento
Nastro



Gestione Nastro Trasportatore



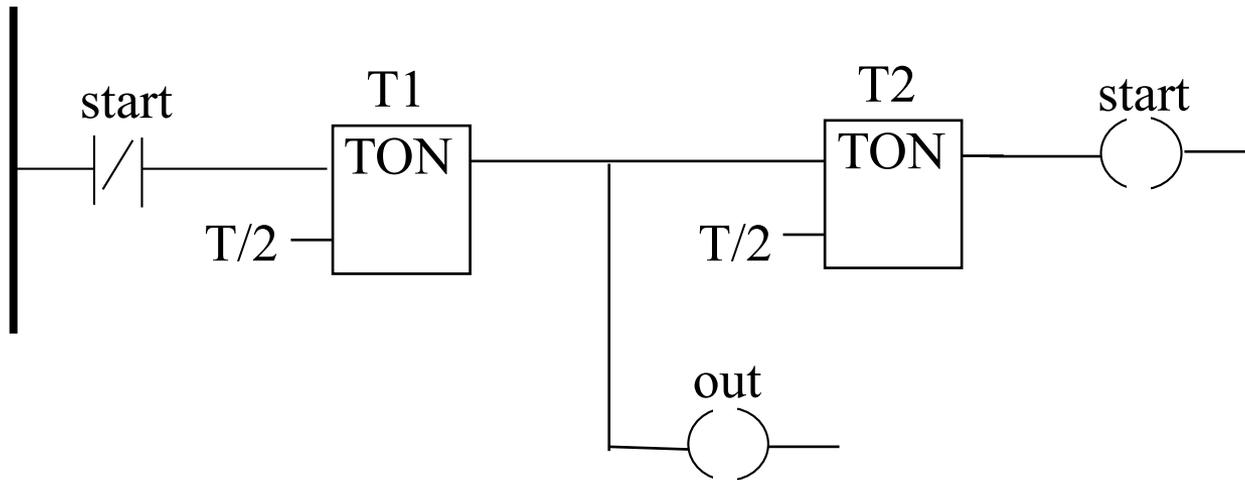
Tecnica basata su Relazioni I/O

Circuito di Clock

- ❖ Si supponga di voler realizzare un programma che permetta di fornire in uscita al PLC un segnale periodico ad onda quadra.
- ❖ Sia T il periodo del segnale.
- ❖ Sia out la variabile binaria alla quale viene associata l'uscita fisica del PLC per la quale si vuole produrre il segnale periodico.
- ❖ Si consideri nella soluzione del problema una variabile binaria interna ($start$), inizializzata a OFF (0).
 - Feedback sulla variabile $start$
- ❖ Si considerino, infine, due function block timer TON, denominati $T1$ e $T2$, ciascuno caratterizzato dal valore del PT pari a $T/2$ (semiperiodo).

Tecnica basata su Relazioni I/O

Circuito di Clock

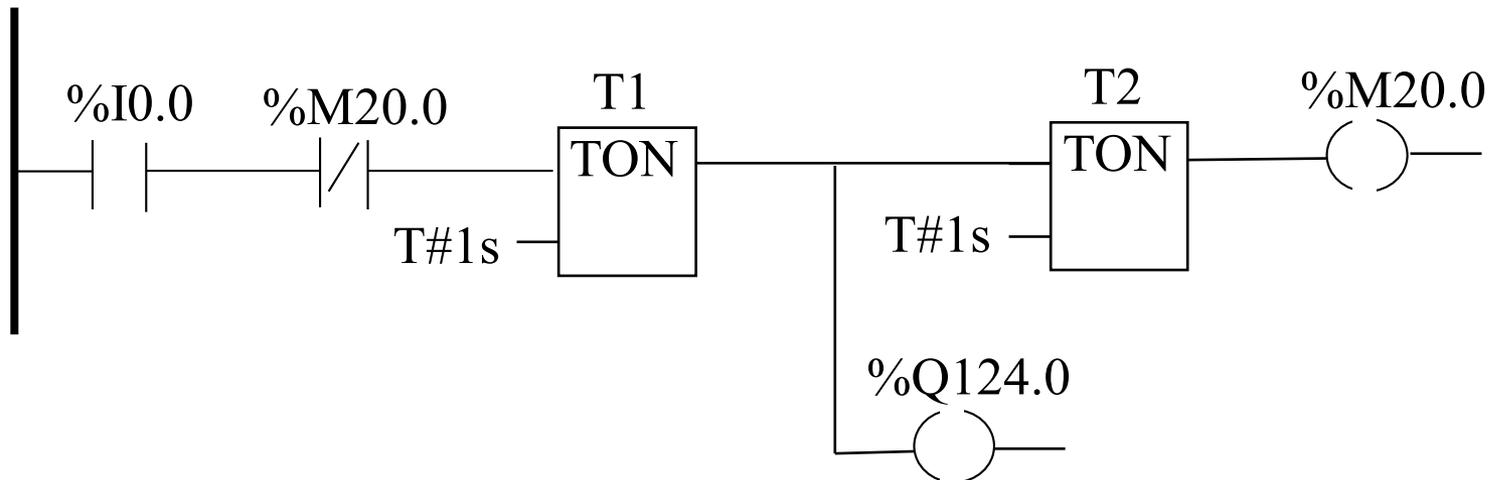


Scansione	start contatto	T1		T2		out	start coil
		Q	ET	Q	ET		
1..n	0	0	<T/2	0	0	0	0
n+1	0	1	T/2	0	0	1	0
n+2...m	0	1	T/2	0	<T/2	1	0
m+1	0	1	T/2	1	T/2	1	1
m+2	1	0	0	0	0	0	0
m+3	0	uguale alla scansione 1					
	0	stessa sequenza precedente					

Tecnica basata su Relazioni I/O

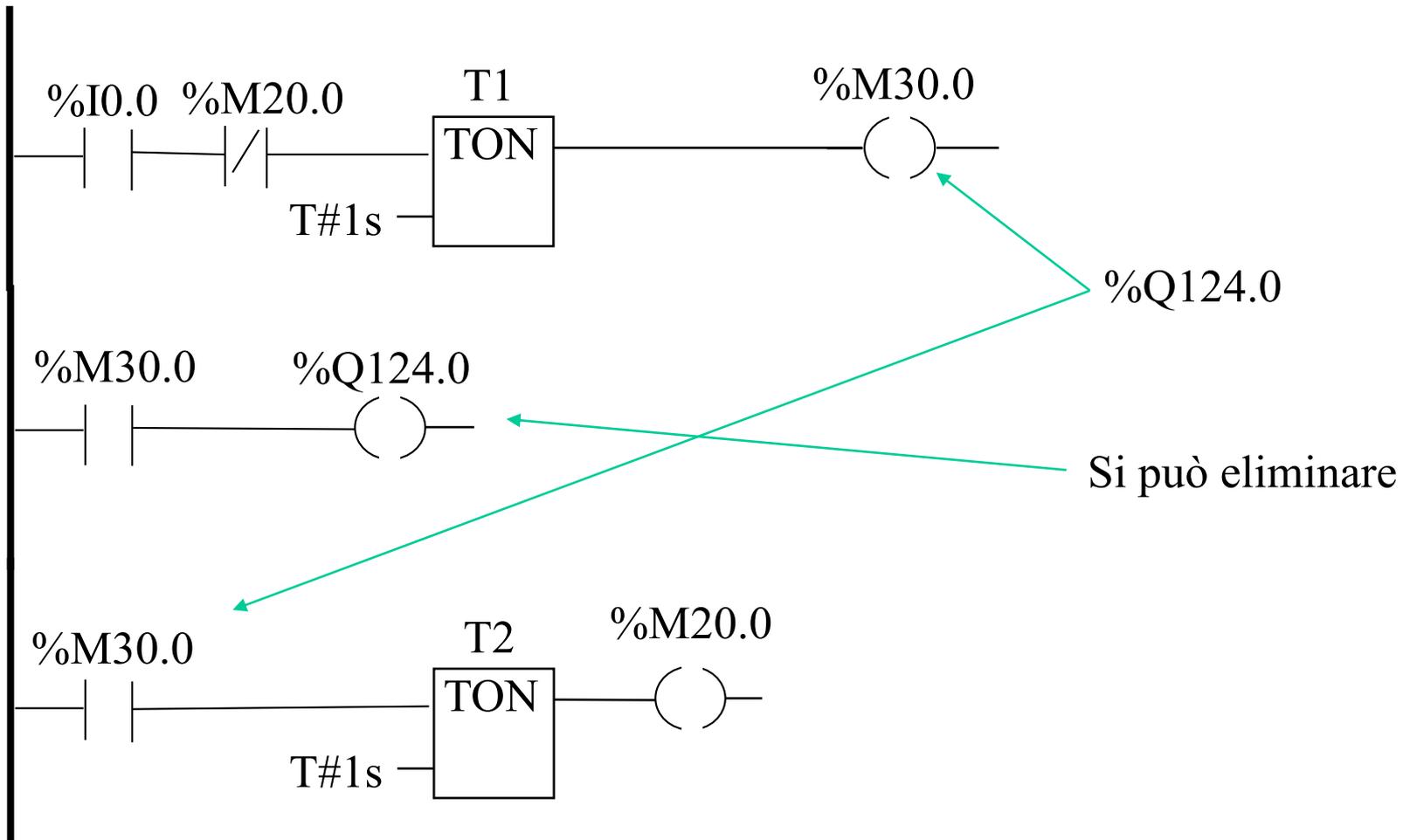
Circuito di Clock

- ❖ Si supponga di inserire un ingresso %I0.0 per gestire l'attivazione e lo stop del clock.
- ❖ Si considerino: %Q124.0 (out) e %M20.0 (start)
- ❖ Si consideri infine un semiperiodo di 1s



Tecnica basata su Relazioni I/O

Circuito di Clock



Tecnica basata sulla Macchina a Stati

- ❖ Esistono problemi in cui l'attivazione delle uscite (ad esempio i comandi agli attuatori) non dipende esclusivamente dagli ingressi (o da variabili interne, quali bits, contatori, timers) ma è legata al concetto di "stato".
- ❖ Tali problemi sono caratterizzati da soluzioni che prevedono l'evoluzione del sistema da uno stato ad un altro, a partire da uno stato iniziale per far ritorno, spesso, a tale stato.
- ❖ Per tali problemi, il comando di uno o più attuatori avviene in corrispondenza di uno stato, e può verificarsi che lo stesso attuatore venga attivato in due o più stati differenti anche in corrispondenza di ingressi diversi.
- ❖ L'evoluzione del sistema da uno stato ad un altro avviene in corrispondenza di valori assunti da particolari ingressi, oppure in base a valori di timers o di contatori, ovvero da valori di opportune espressioni logiche.

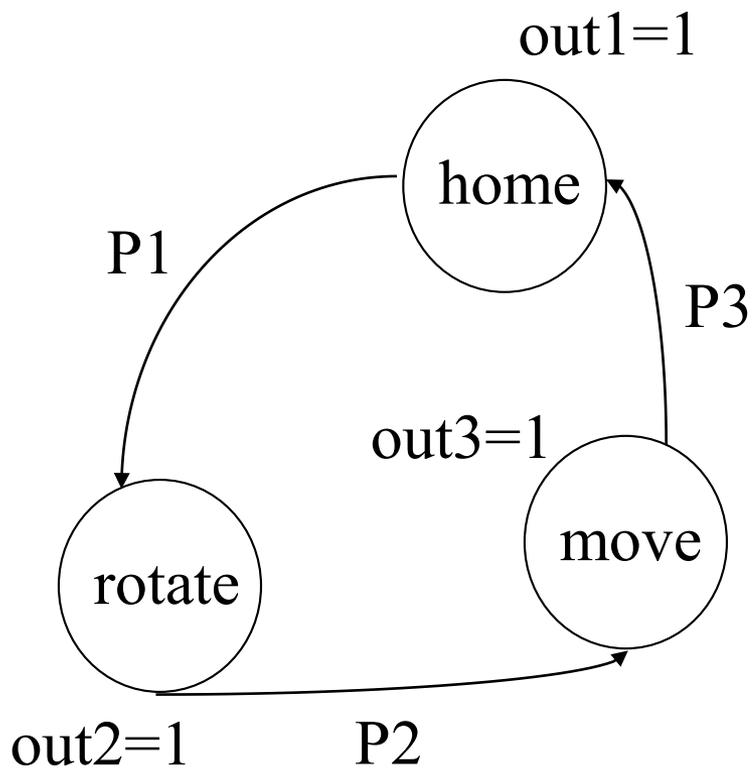
Tecnica basata sulla Macchina a Stati

- ❖ Rappresentare la soluzione del problema con una macchina a stati
- ❖ Ogni stato viene rappresentato da una variabile binaria
- ❖ Per ogni stato vengono identificate le azioni da eseguire
- ❖ Vengono identificati gli eventi che producono il passaggio di stato (transizioni)
 - Ciascuna transizione corrisponderà ad una condizione booleana

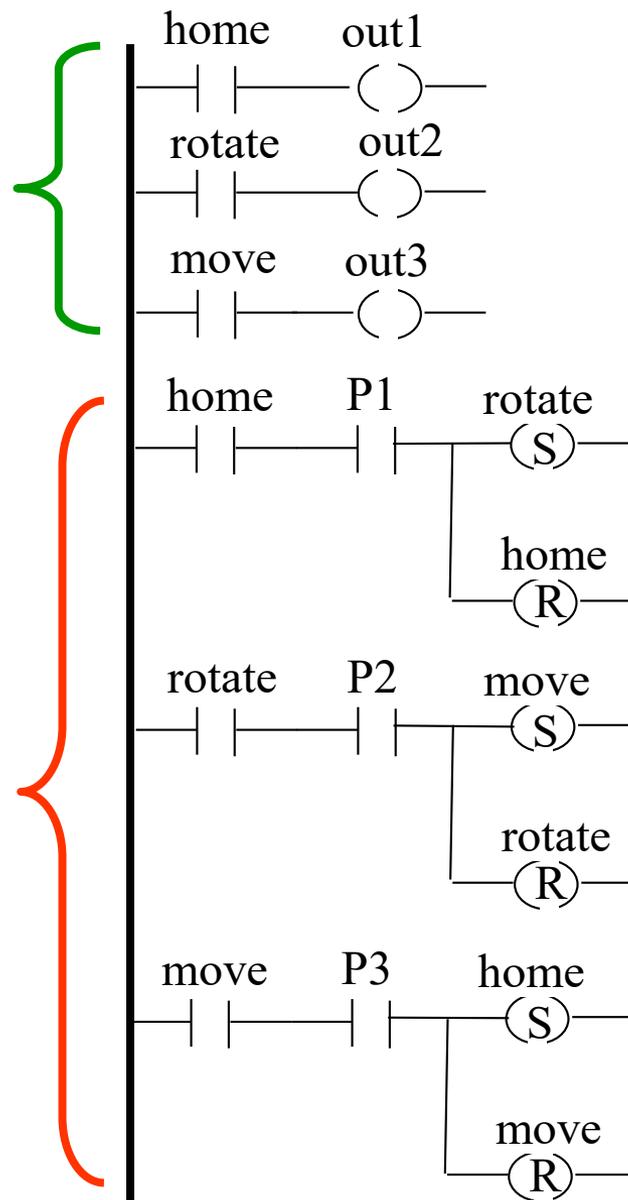
Tecnica basata sulla Macchina a Stati

- ❖ Il programma in Ladder deve codificare quattro azioni:
 - Inizializzazione.
 - ✓ Viene eseguita una sola volta, all'inizio.
 - ✓ Serve per settare la condizione iniziale, tipicamente lo stato iniziale
 - Rappresentazione delle **azioni** eseguite in ciascuno degli stati
 - Rappresentazione delle **transizioni** tra uno stato ed un altro (da uno stato a monte ad uno stato a valle)
 - Rappresentazione della **attivazione dello stato** a valle e **disattivazione dello stato precedente** (a monte) in corrispondenza di una determinata transizione

Esempio di Tecnica di Programmazione basata sulla Macchina a Stati



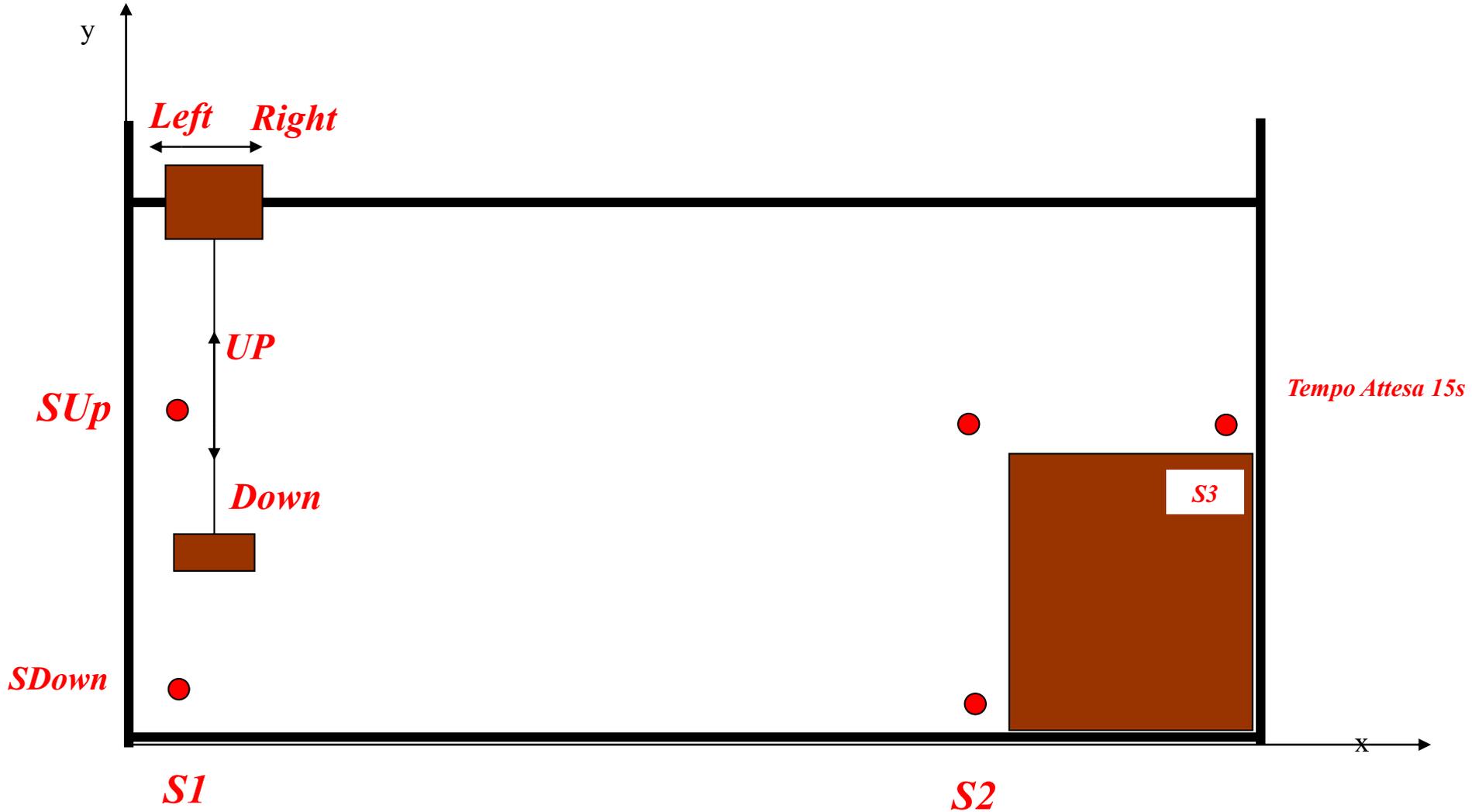
Inizializzazione: home = 1



Esempio: Carroponte

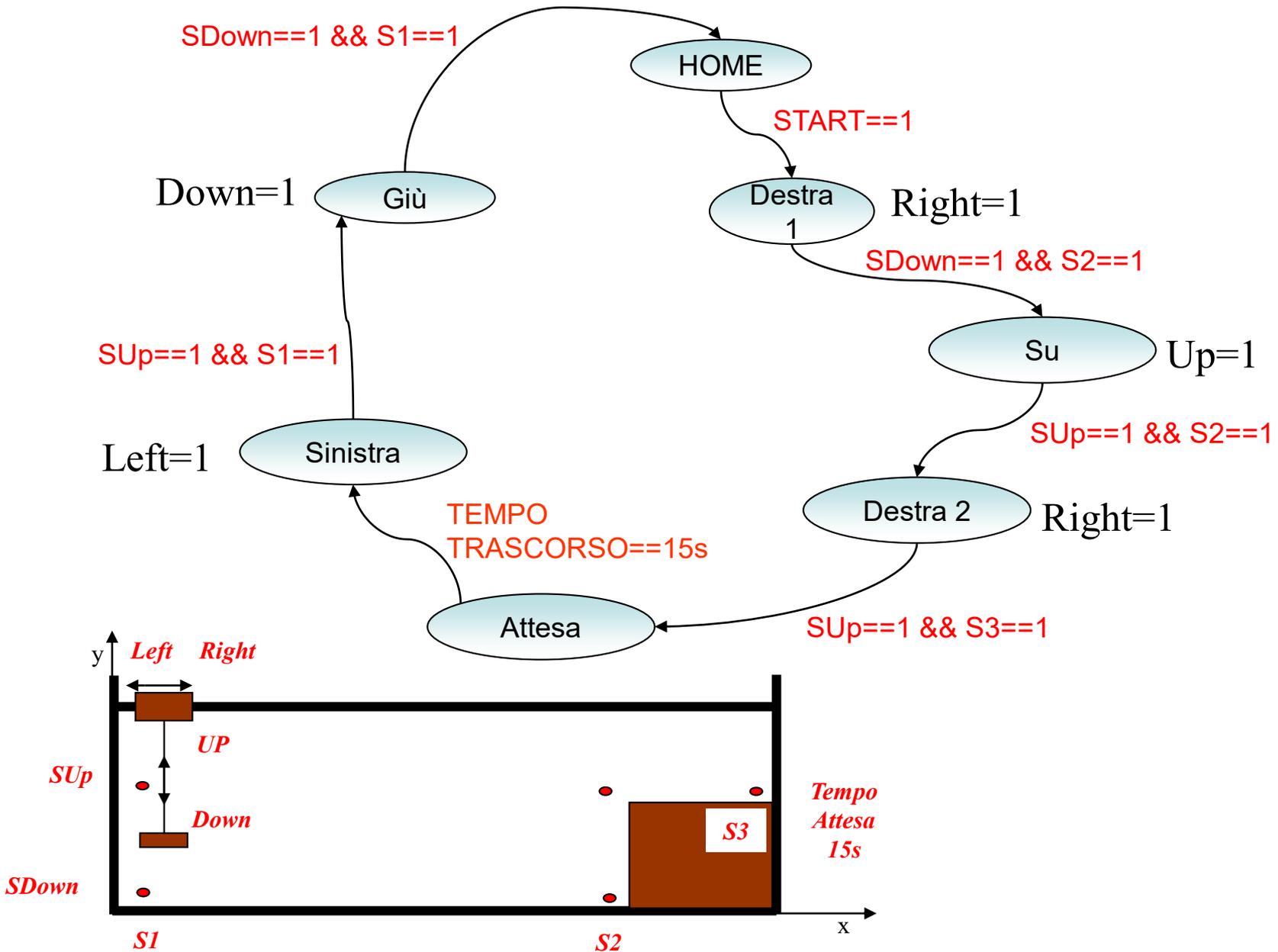


Esempio: Carroponte

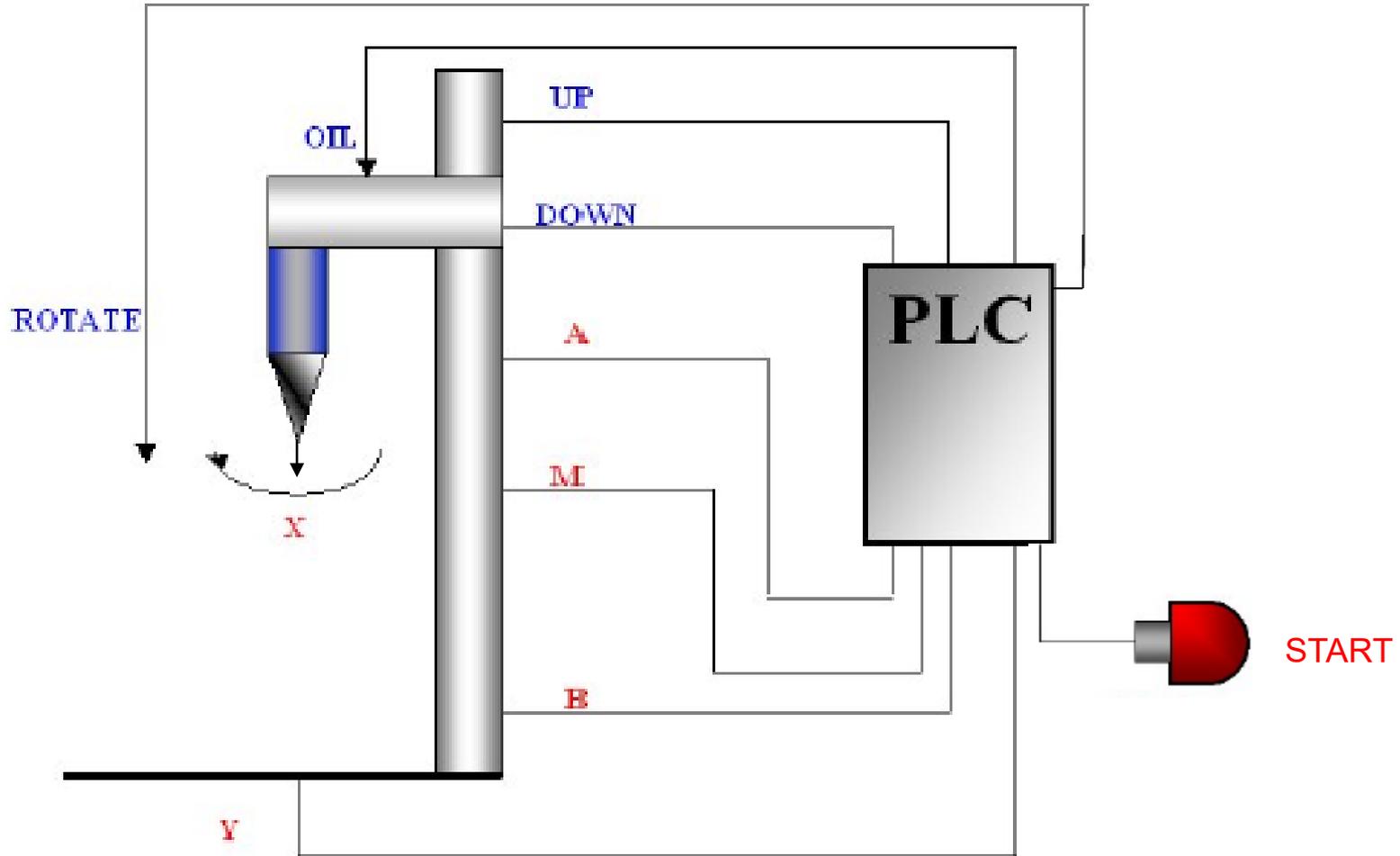


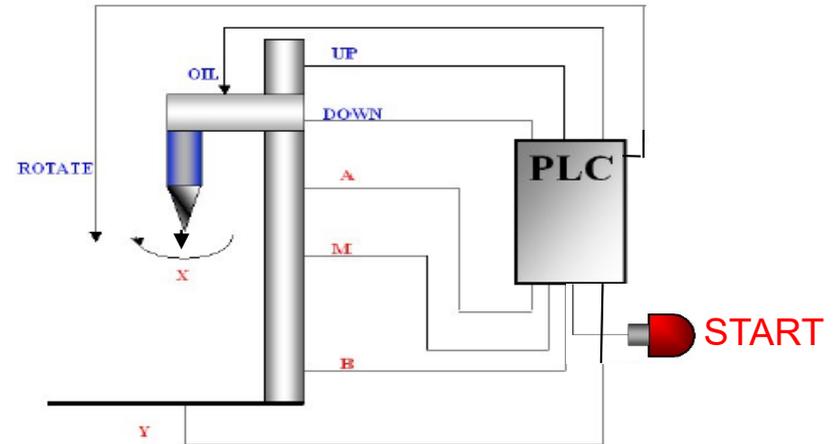
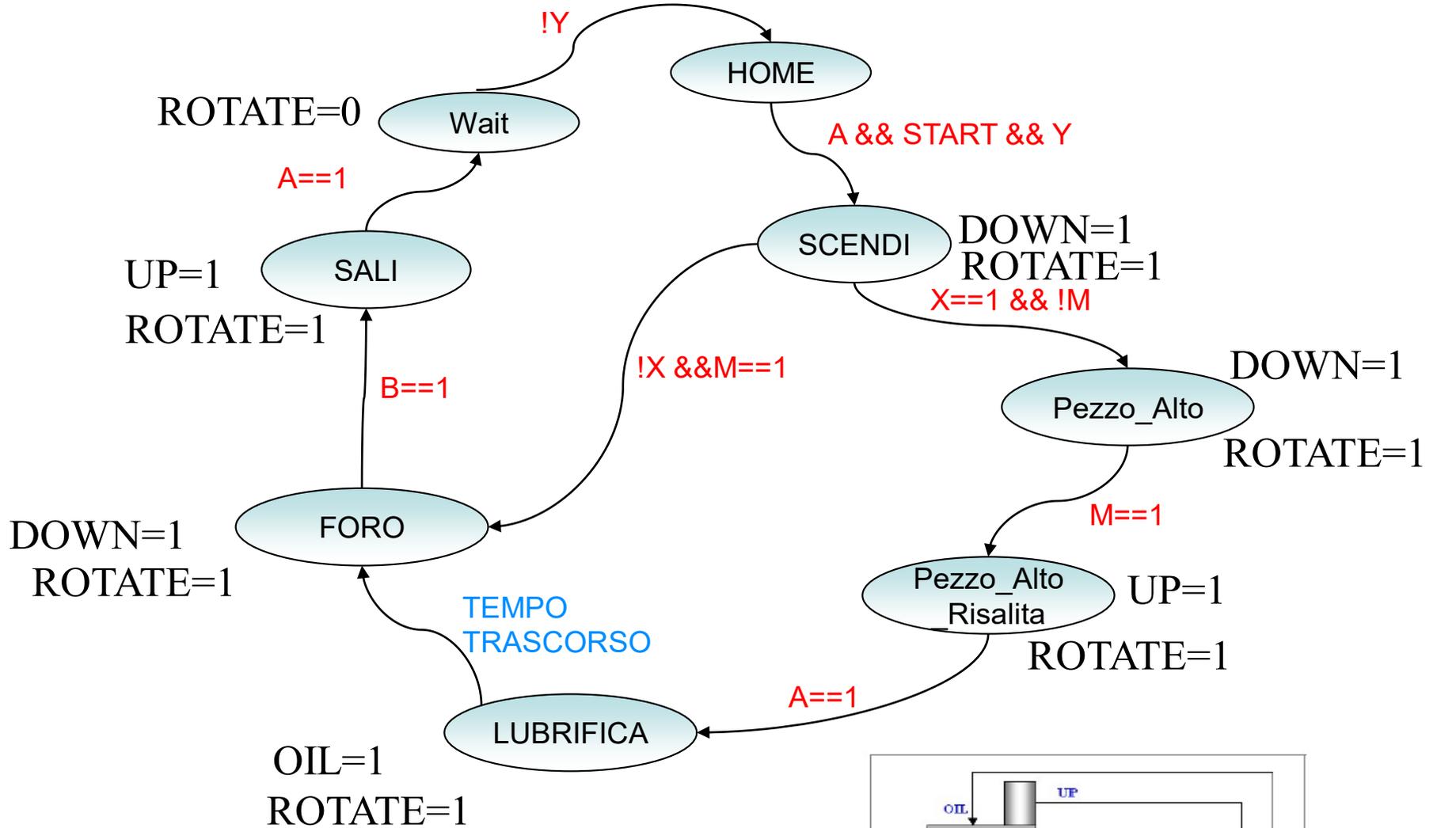
Home=(S1=1,SDown=1)

Esempio: Carroponte

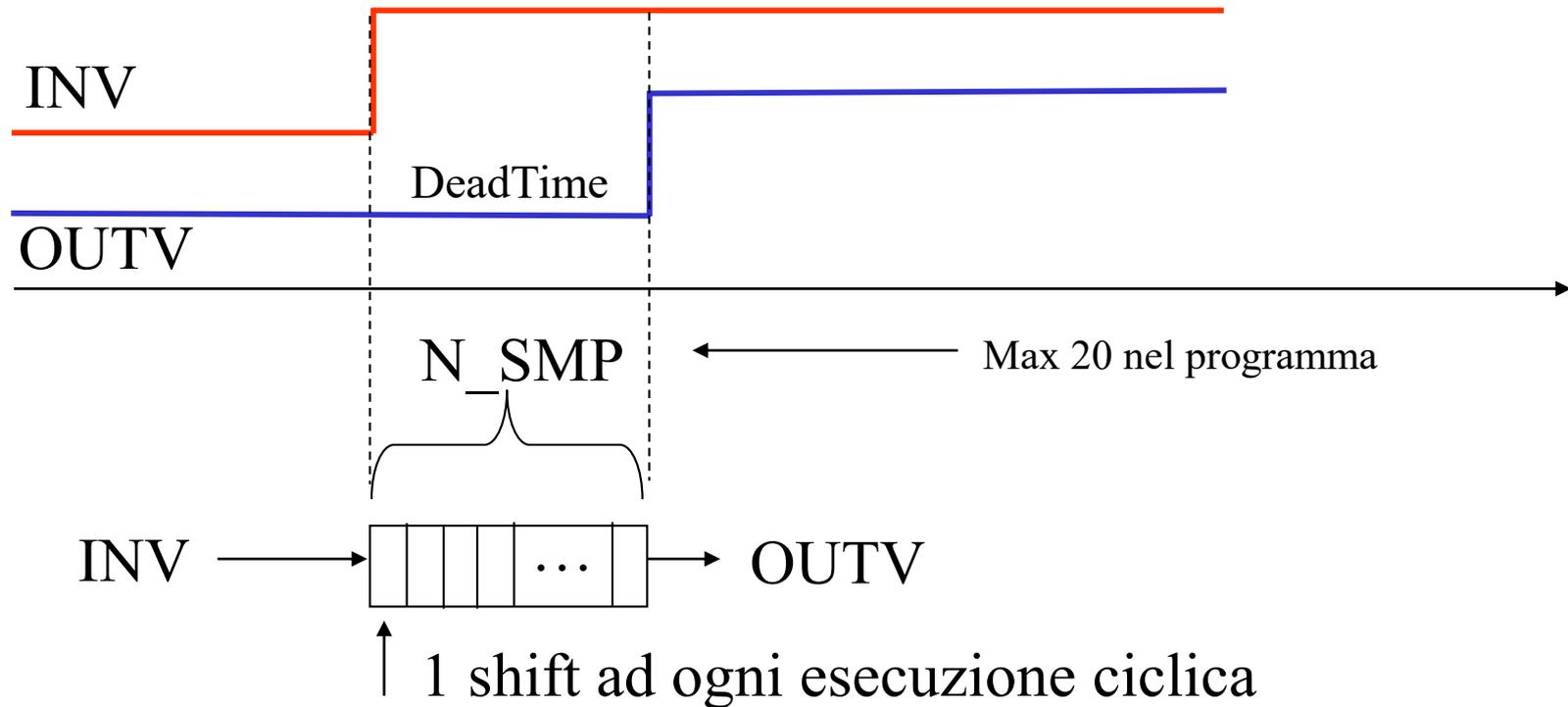


Esercizio: Trapano Automatico





Esempio OB Schedulazione Orologio



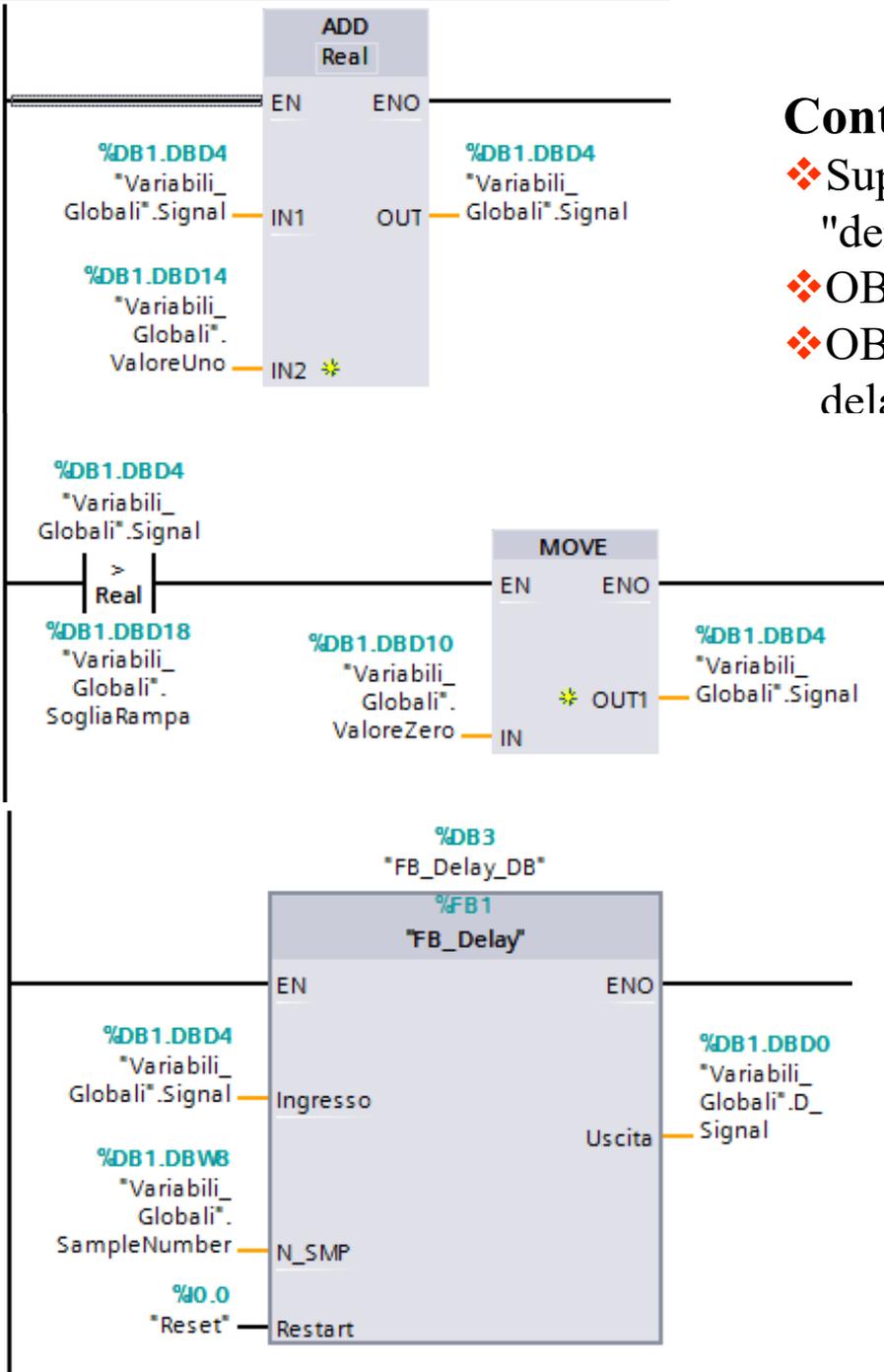
- ❖ INV = segnale reale
- ❖ OUTV = segnale INV ritardato di un intervallo temporale.
- ❖ Viene utilizzato un vettore interno di un numero massimo di elementi (memorizza una sequenza di segnali di ingresso INV)
- ❖ **Viene effettuato uno shift ad ogni esecuzione del programma**
- ❖ N_SMP rappresenta il numero di shift dopo cui il segnale in ingresso viene prodotto in uscita
- ❖ In genere **$DeadTime = N_SMP * Intervallo\ di\ schedulazione\ (OB\ Schedulazione\ Orologio)$**

Esempio OB Schedulazione Orologio

❖ Supporremo che sia possibile memorizzare Max 20 ritardi

❖ Variabili: Blocco Dati Globale

Variabili_Globali					
	Nome	Tipo di dati	Offset	Valore di avvio	A ritenzio...
Static	Static				<input type="checkbox"/>
	D_Signal	Real	0.0	0.0	<input checked="" type="checkbox"/>
	Signal	Real	4.0	0.0	<input checked="" type="checkbox"/>
	SampleNumber	Int	8.0	5	<input checked="" type="checkbox"/>
	ValoreZero	Real	10.0	0.0e+0	<input checked="" type="checkbox"/>
	ValoreUno	Real	14.0	1.0e+0	<input checked="" type="checkbox"/>
	SogliaRampa	Real	18.0	4.0e+1	<input checked="" type="checkbox"/>



Contenuto di OB Schedulazione Orologio:

- ❖ Supporremo che l'ingresso sia un segnale a "dente di sega" [0,40], con incrementi unitari
- ❖ OB realizza il segnale
- ❖ OB richiama il "**FB_Delay**" che implementa il delay

Variabili del Function Block **FB_Delay**:

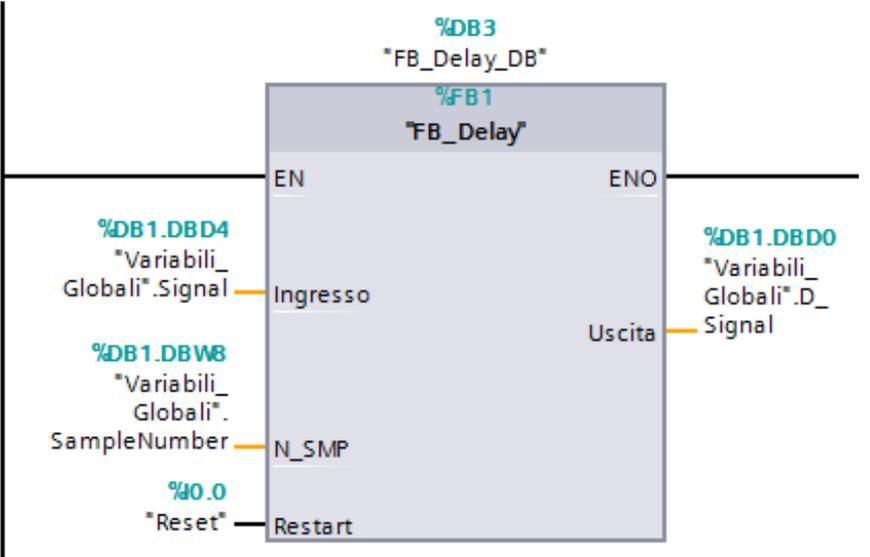
- ❖ Ingresso: tipo REAL per il segnale di ingresso
- ❖ N_SMP: tipo INT per il moltiplicatore
- ❖ Restart: tipo BOOL per il reset

Variabile di uscita:

- ❖ Uscita: tipo REAL per il segnale di uscita

Variabile statica:

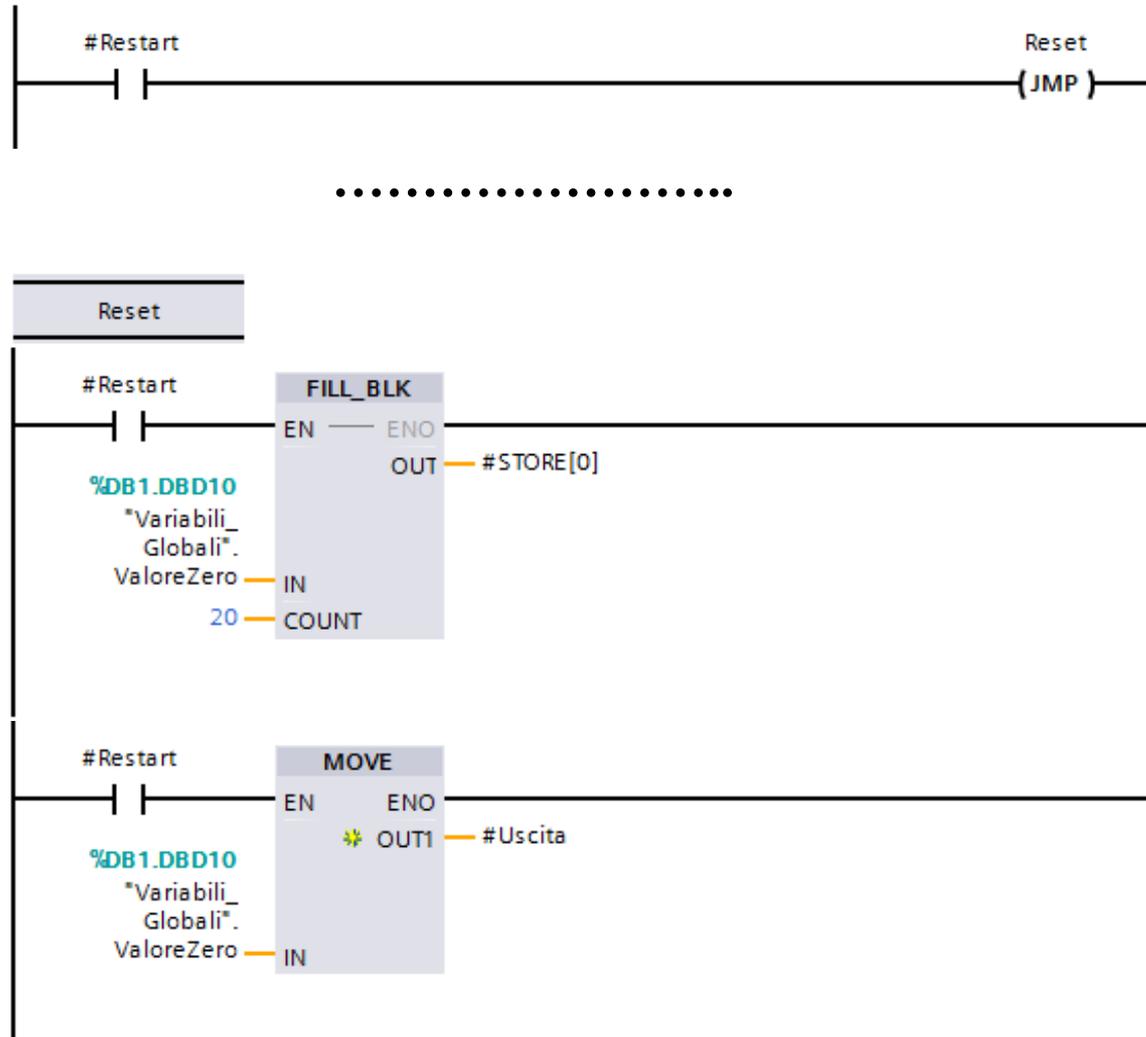
- ❖ STORE: memoria dati ARRAY di REAL
- ❖ indice_1, indice_2: tipo INT (indici vettore)
- ❖ sup: INT (indice vettore)



FB_Delay				
	Nome	Tipo di dati	Valore di default	Ritenzione
1	Input			
2	Ingresso	Real	0.0	Non a ritenzione
3	N_SMP	Int	0	Non a ritenzione
4	Restart	Bool	false	Non a ritenzione
5	Output			
6	Uscita	Real	0.0	Non a ritenzione
7	InOut			
8	<Inserisci>			
9	Static			
10	STORE	Array[0..19] of Real		A ritenzione
11	indice_1	Int	0	A ritenzione
12	indice_2	Int	0	A ritenzione
13	sup	Int	0	A ritenzione
14	Temp			
15	<Inserisci>			
16	Constant			
17	<Inserisci>			

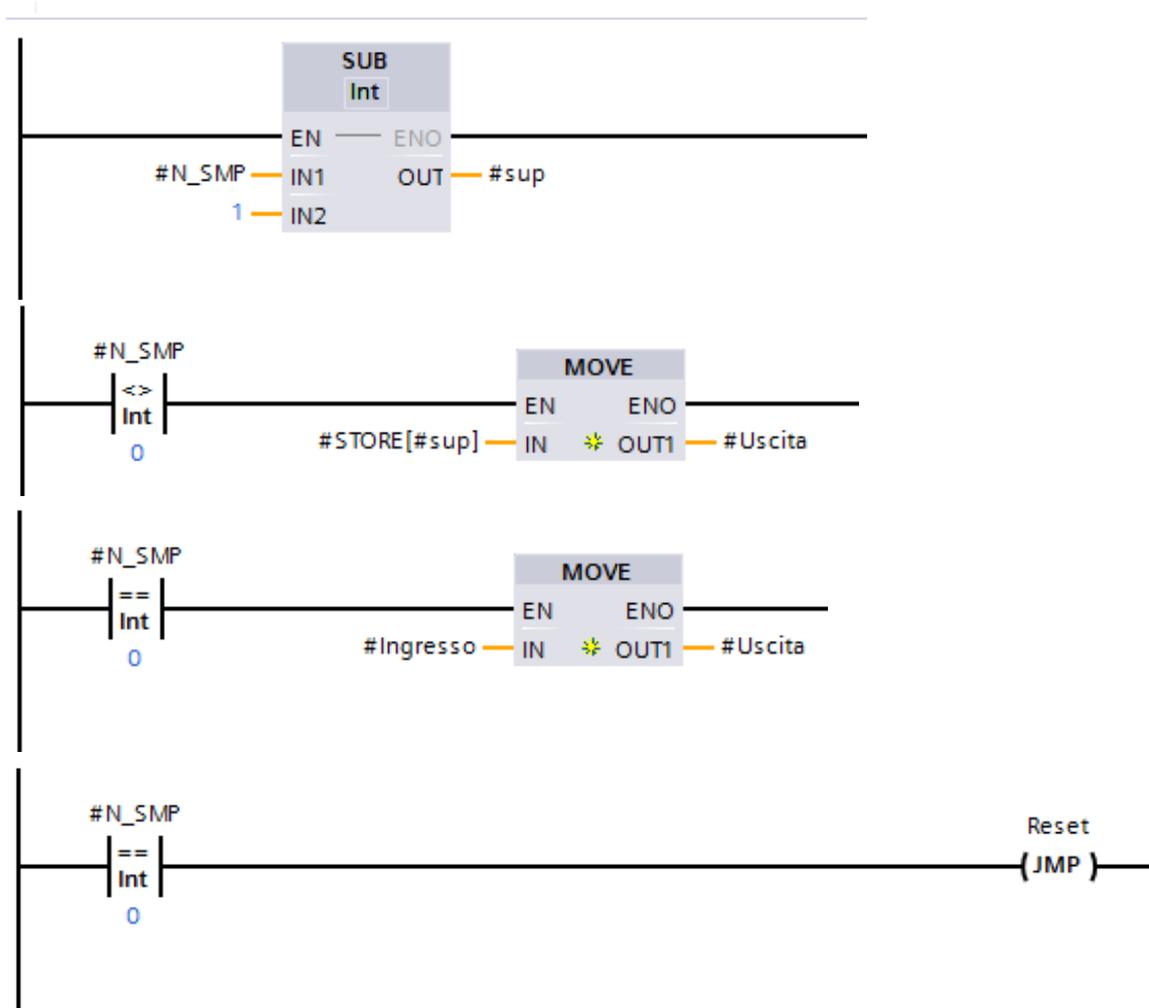
Esempio OB Schedulazione Orologio

❖ Porzione di Codice per il Reset (attivato se Restart==1):



Esempio OB Schedulazione Orologio

❖ Porzione di Codice per la "produzione" dell'uscita:



❖ Porzione di Codice per lo "shift" dell'ingresso :

