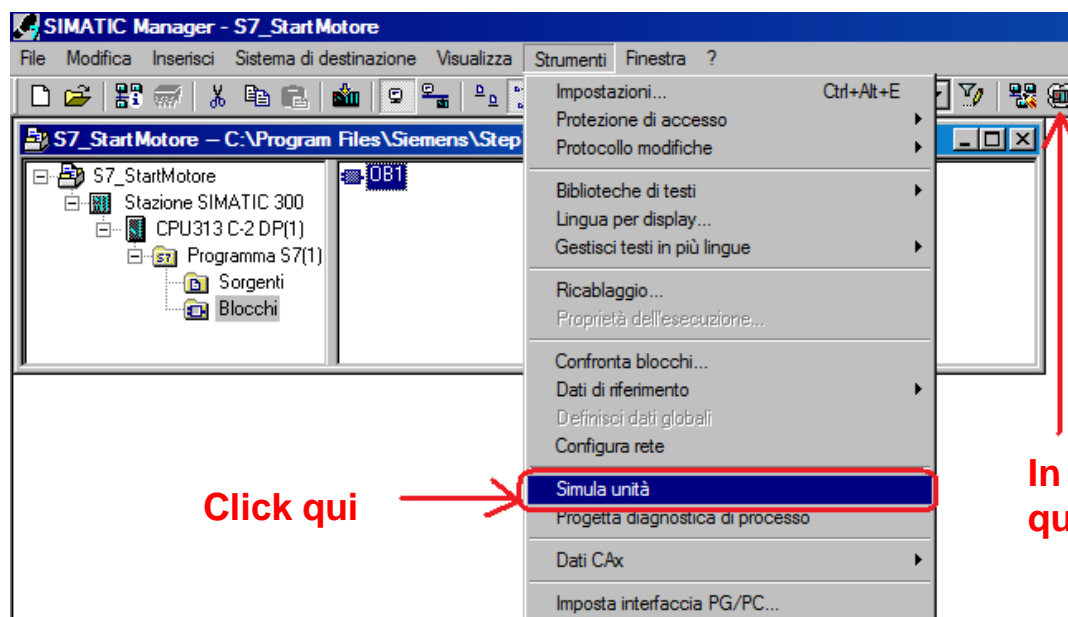


# Tecnica basata su Relazioni I/O

## Motore: Marcia/Motore/Arresto

### SIMULAZIONE:

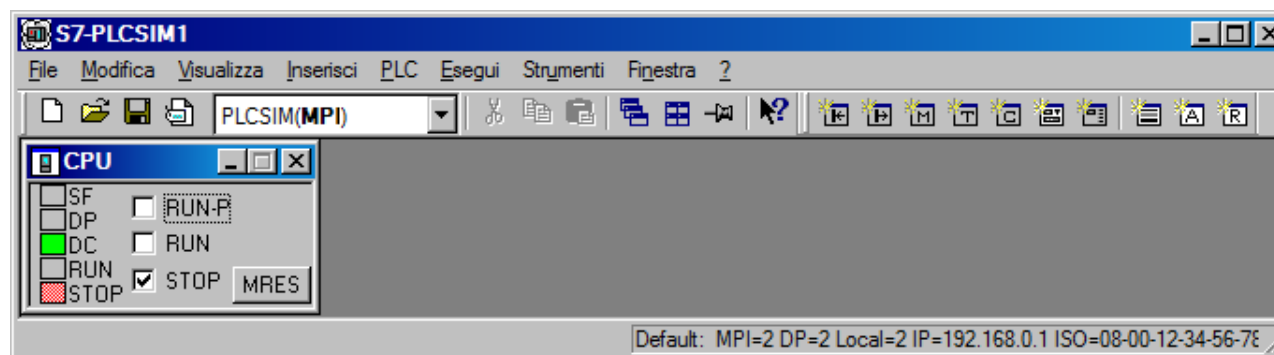
Richiamare con il menu «Strumenti» la voce «Simula unità»



Click qui

In alternativa posso usare questo bottone

Chiudere eventuali finestre (ingressi, uscite, merker,...) relativi a precedenti progetti

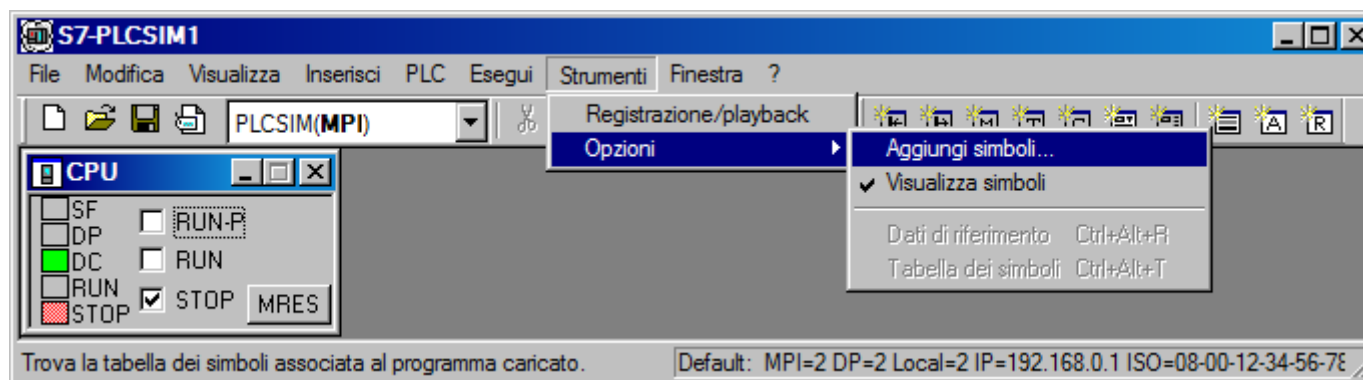


# Tecnica basata su Relazioni I/O

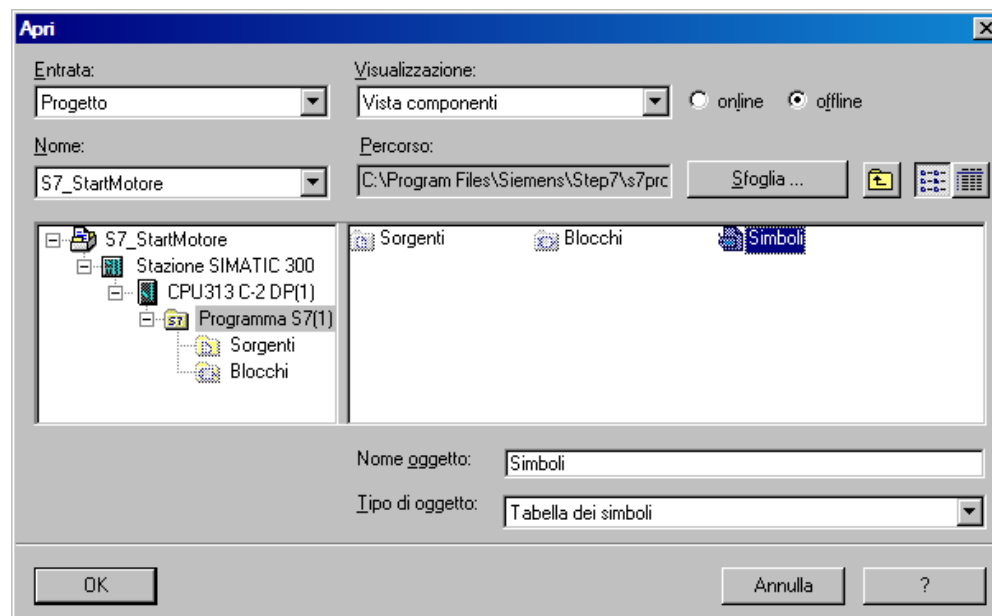
## Motore: Marcia/Motore/Arresto

### SIMULAZIONE:

Aggiungere nella finestra di simulazione la tabella dei simboli



Selezionare il nome progetto e l'icona «Simboli» da caricare e premere OK

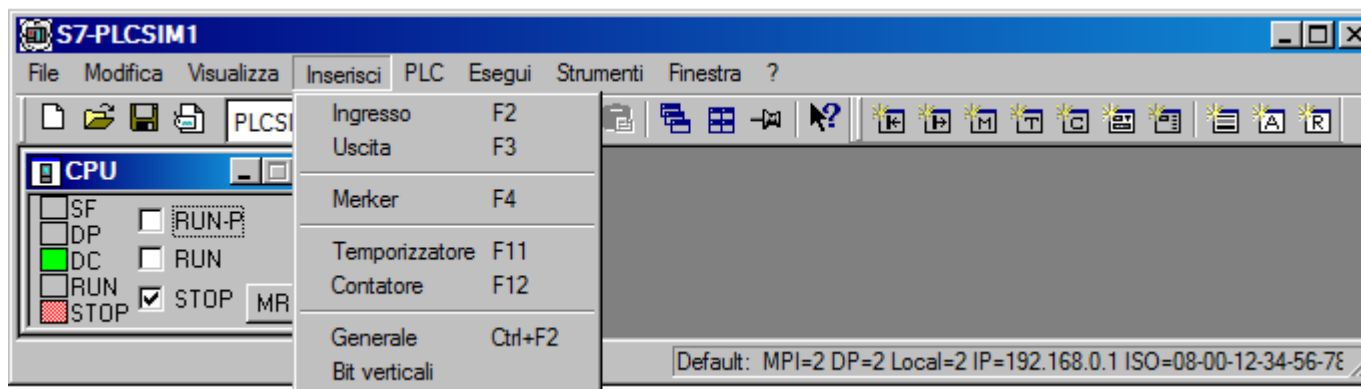


# Tecnica basata su Relazioni I/O

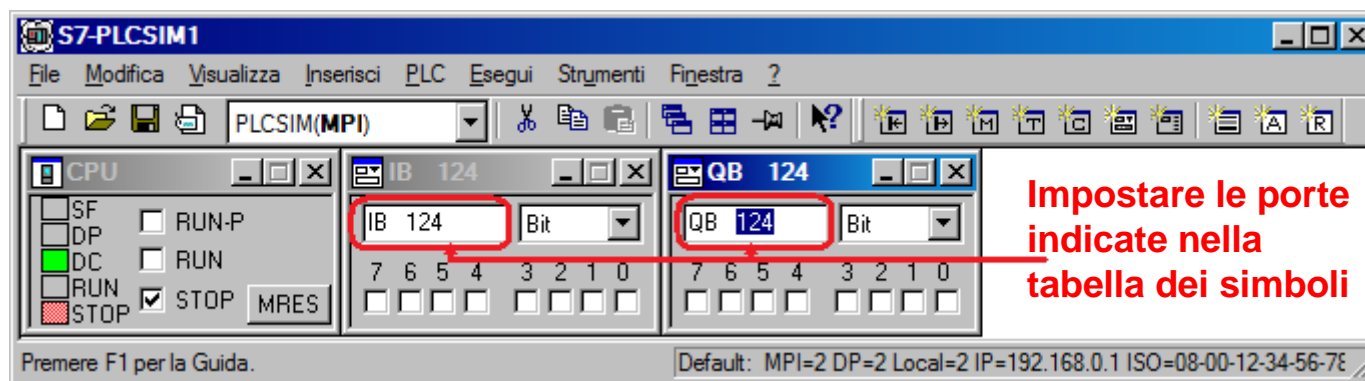
## Motore: Marcia/Motore/Arresto

### SIMULAZIONE:

Aggiungere nel simulatore gli ingressi e le uscite



Il simulatore deve contenere le seguenti porte di ingresso/uscita

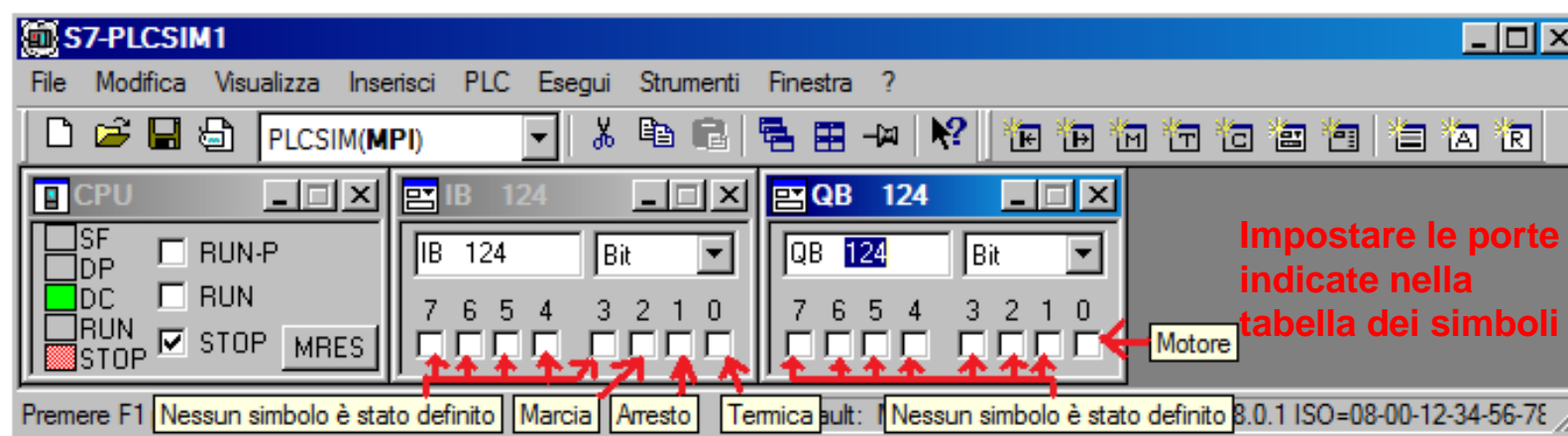


# Tecnica basata su Relazioni I/O

## Motore: Marcia/Motore/Arresto

### SIMULAZIONE:

Se tutto è ok, soffermandosi con il mouse sui bit di I/O dovrebbero apparire i tooltip indicati nella figura sottostante.



Si osservi che in corrispondenza dei bit di ingresso/uscita apparirà il tooltip a lato se non si è preventivamente caricata la tabella dei simboli

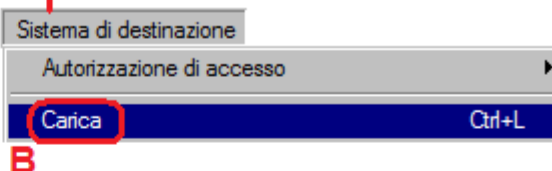
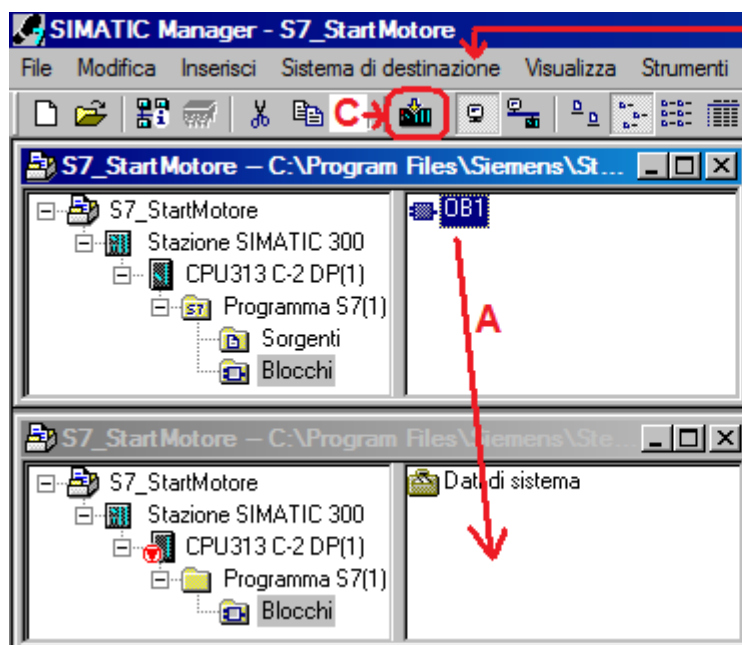
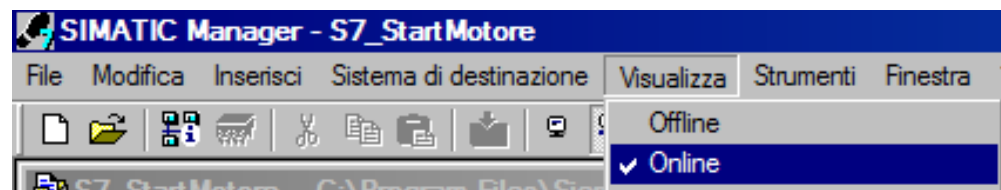
**Nessuna tabella dei simboli è stata occupata**

# Tecnica basata su Relazioni I/O

## Motore: Marcia/Motore/Arresto

### SIMULAZIONE:

Visualizzare nel SIMATIC Manager l'«online»



Carichiamo **OB1** utilizzando uno dei 3 modi:

**A** → drag & drop dall'offline all'online

**B** → Menu «Sistema di destinazione» voce «Carica»

**C** → Click sull'apposito bottone sulla barra degli strumenti

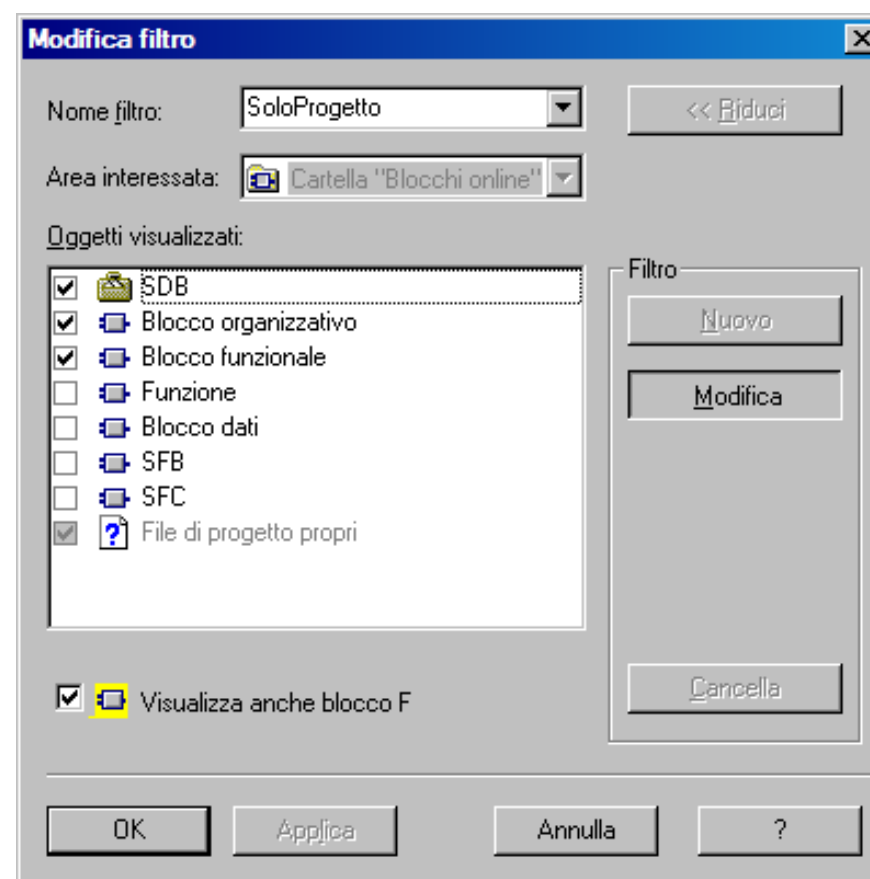
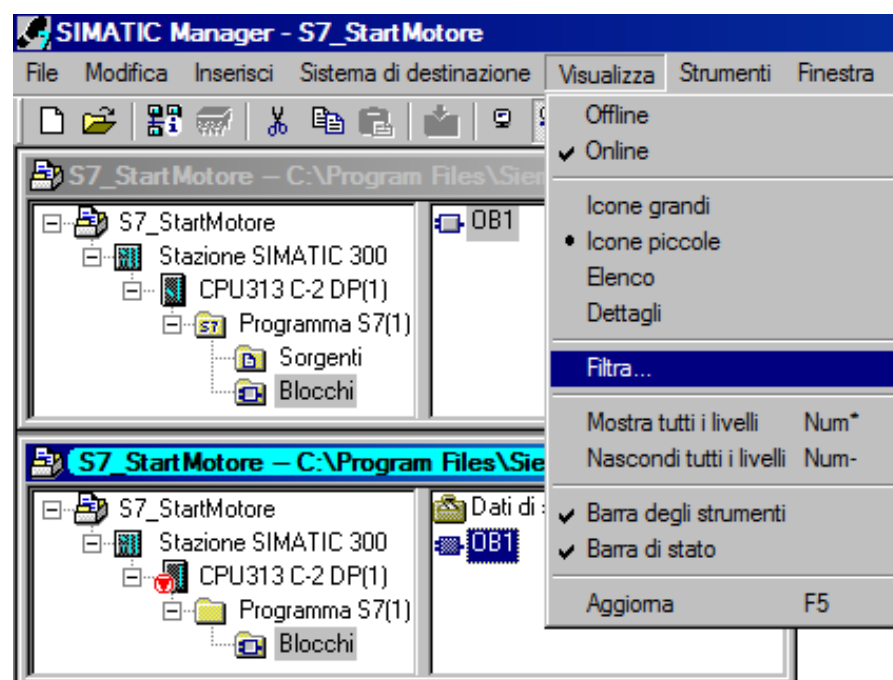


# Tecnica basata su Relazioni I/O

## Motore: Marcia/Motore/Arresto

### SIMULAZIONE:

Si osservi che l'«online» è stato filtrato



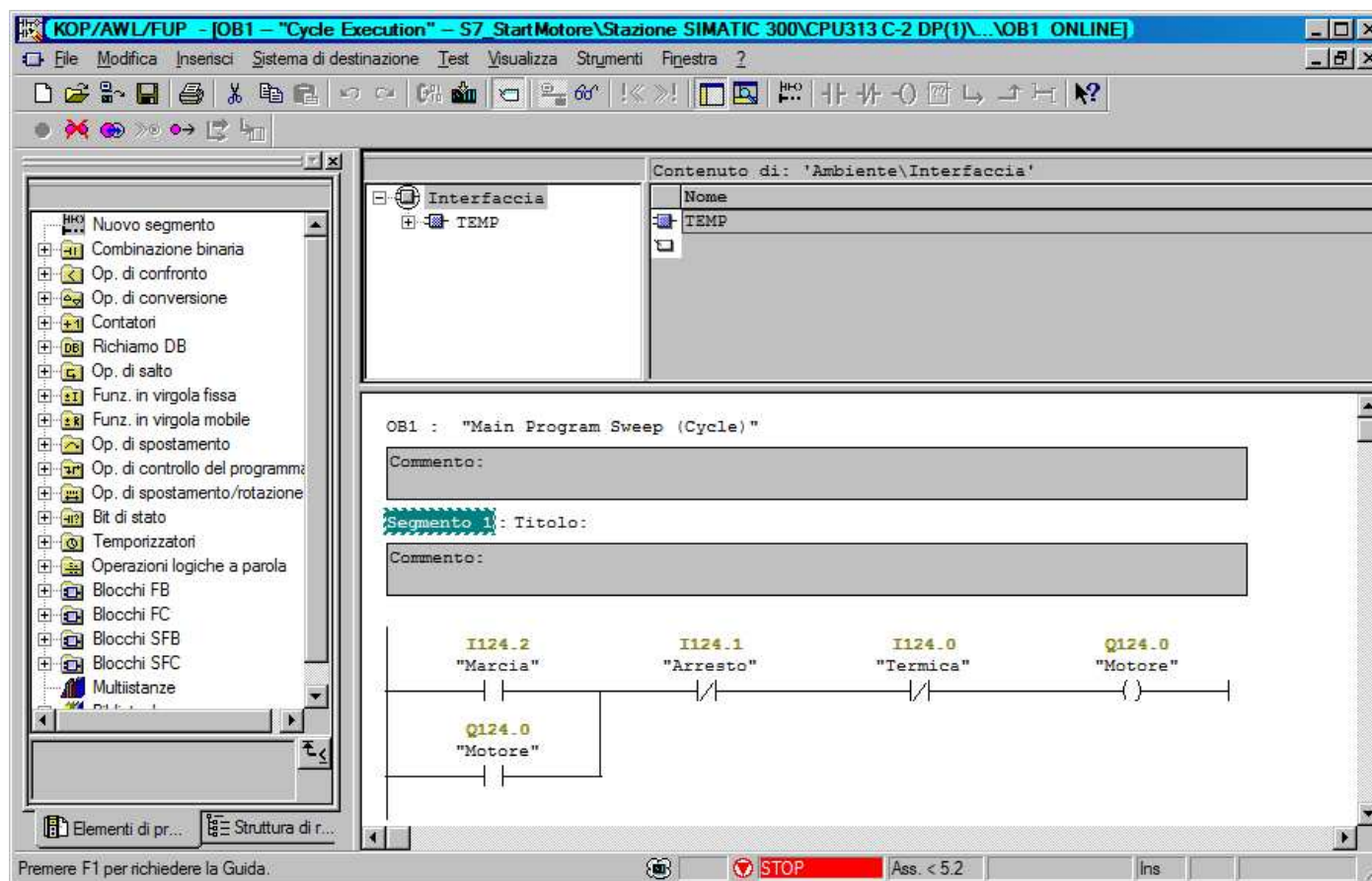
Registrando le seguenti «impostazioni di filtro»

# Tecnica basata su Relazioni I/O

## Motore: Marcia/Motore/Arresto

### SIMULAZIONE:

Facciamo doppio click sull'icona **OB1** nella finestra dell'online. Dovrebbe apparire:



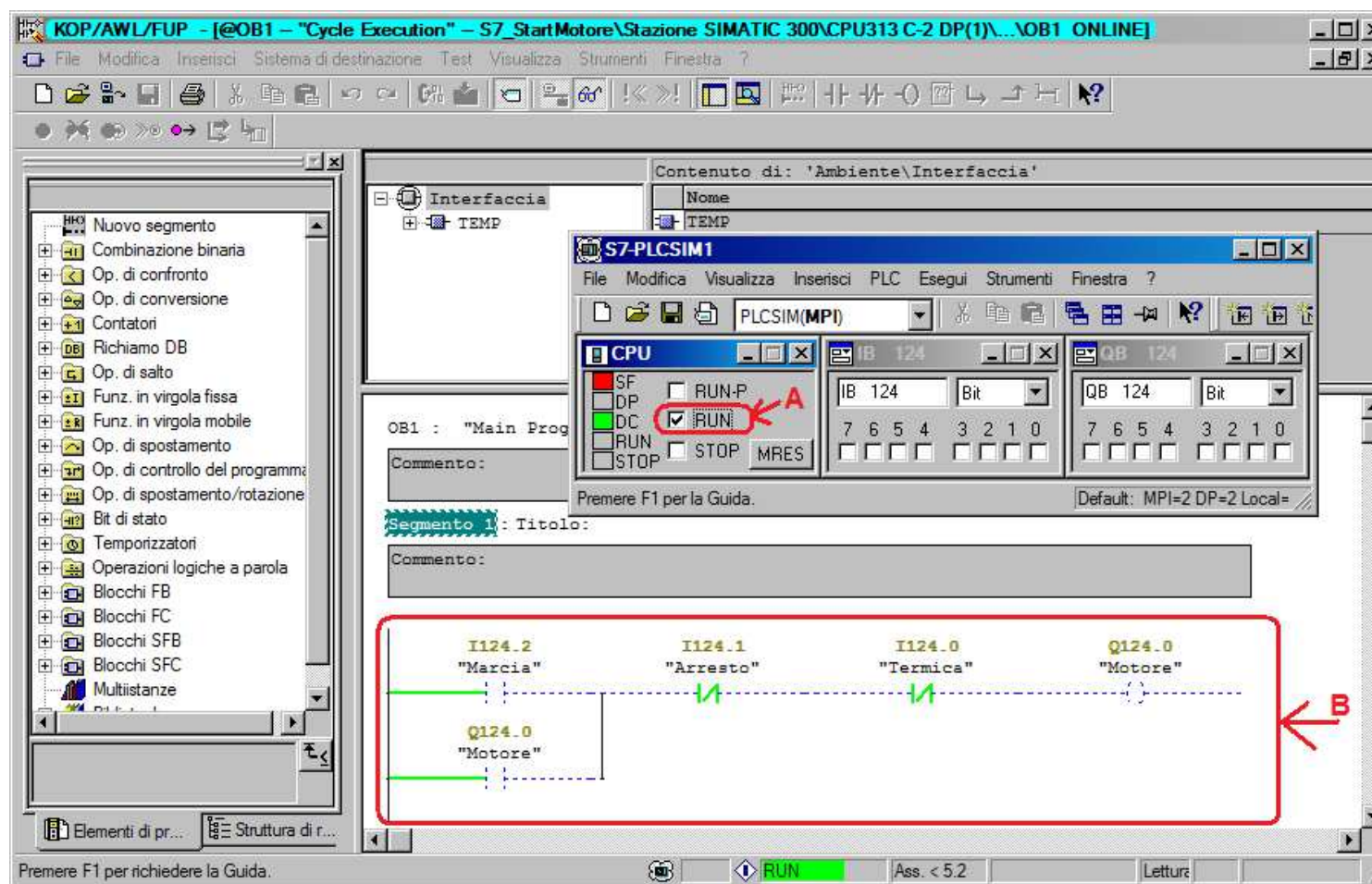
Premiamo la combinazione **Ctrl+F7** oppure selezioniamo il menu «**Test**» voce «**Controlla**» per controllare lo stato degli ingressi e delle uscite durante la simulazione



# Tecnica basata su Relazioni I/O

## Motore: Marcia/Motore/Arresto

Sovrapponiamo il simulatore con l'online e mandiamo in RUN (A) il plc.



(B) Si noti che le linee orizzontali in tensione risultano evidenziate in verde

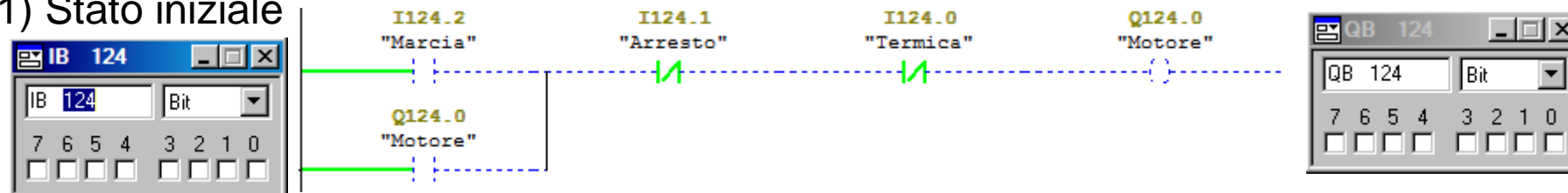


# Tecnica basata su Relazioni I/O

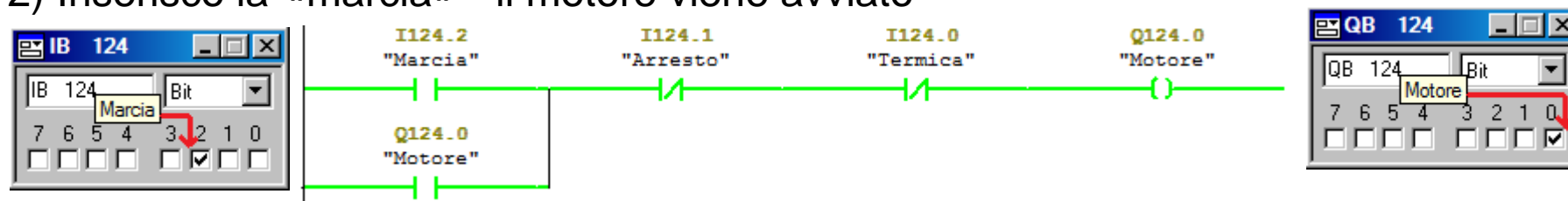
## Motore: Marcia/Motore/Arresto

### TEST:

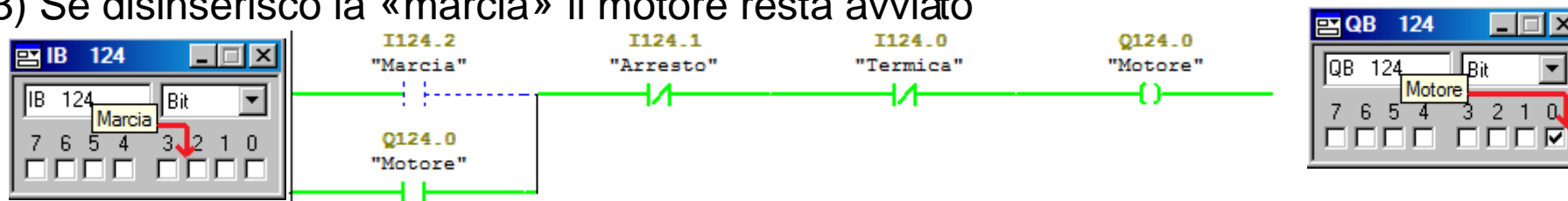
#### 1) Stato iniziale



#### 2) Inserisco la «marcia» - il motore viene avviato



#### 3) Se disinserisco la «marcia» il motore resta avviato

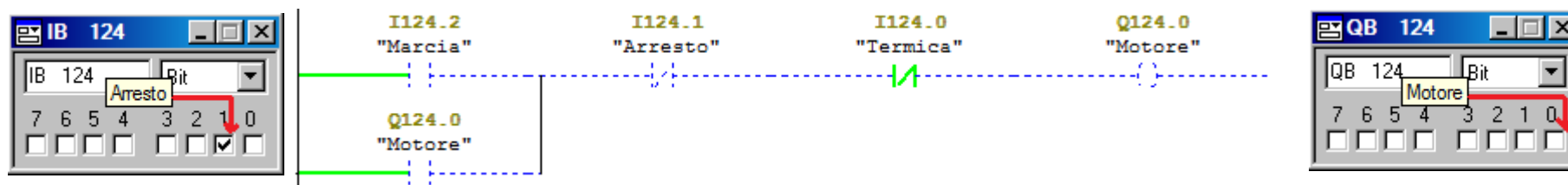


# Tecnica basata su Relazioni I/O

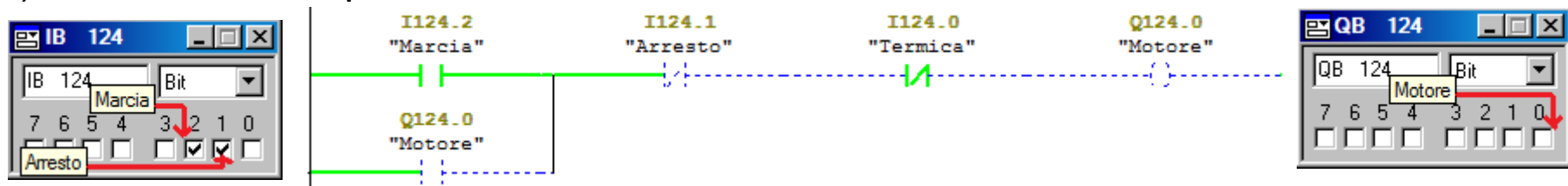
## Motore: Marcia/Motore/Arresto

TEST:

4) Se schiaccio il pulsante di arresto tutto si ferma (idem se scatta la termica)



5) Il motore resta spento anche se reinserisco la marcia



# Elementi di Base del Linguaggio Ladder

Le operazioni logiche combinatorie a bit operano con due cifre: 1 e 0. Queste due cifre costituiscono la base di un sistema numerico denominato sistema binario. Le due cifre 1 e 0 vengono denominate cifre binarie o bit. Nel mondo dei contatti e delle bobine, 1 sta a significare attivato o eccitato, e 0 sta per disattivato o diseccitato.

Le operazioni logiche combinatorie a bit interpretano gli stati di segnale di 1 e 0, e li combinano secondo la logica booleana per eseguire una varietà di funzioni. Queste combinazioni producono un risultato di 1 o 0 che è chiamato **"risultato logico combinatorio"** (RLC).

Qui di seguito sono elencate le operazioni logiche combinatorie a bit necessarie per eseguire le seguenti funzioni:

---   ---	Contatto normalmente aperto
---  /  ---	Contatto normalmente chiuso
---( )	Bobina di uscita
---( # )---	Connettore
---  <b>NOT</b>  ---	Inverti il risultato della combinazione

Le seguenti operazioni rispondono ad un RLC di 1:

---( S )	Imposta bobina
---( R )	Resetta bobina
SR	Imposta resettata flip flop
RS	Resetta imposta flip flop

Altre operazioni rispondono ad una transizione di fronte di salita o di discesa per eseguire le seguenti funzioni:

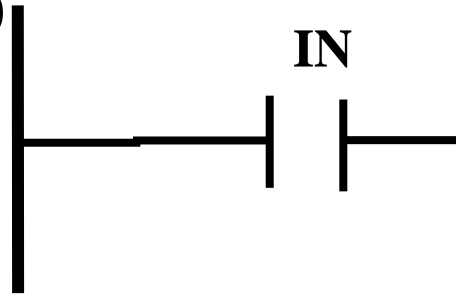
---(N)---	Rileva fronte RLC di discesa
---(P)---	Rileva fronte RLC di salita
<b>NEG</b>	Interroga rilevamento di fronte di discesa
<b>POS</b>	Interroga rilevamento di fronte di salita

 Combinazione binaria

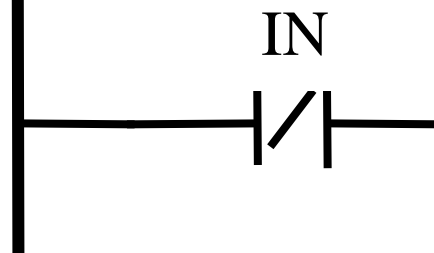
-	-   -
-	- /-
-	- NOT -
<>	-()
<>	-(#)-
<>	-(R)
<>	-(S)
[ ]	RS
[ ]	SR
<>	-(N)-
<>	-(P)-
<>	-(SAVE)
[ ]	NEG
[ ]	POS

# Utilizzo dei Contatti

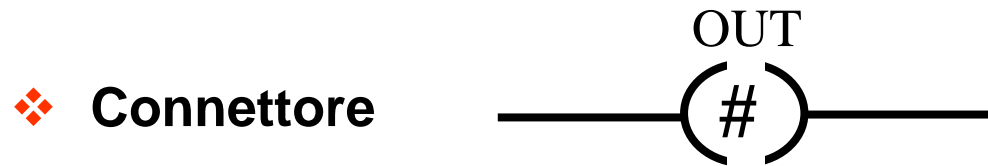
- ❖ Ad ogni contatto viene associata una variabile binaria. Tale variabile viene solamente letta (*può coincidere con un ingresso*).
- ❖ **Contatto Normalmente Aperto:** la corrente fluisce da sinistra a destra se la variabile IN è 1. La corrente fluisce a destra **per qualunque scansione del Programma 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



# Altri contatti del Linguaggio Ladder



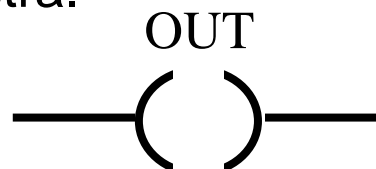
---( # )--- (Connettore) è un elemento di assegnazione intermedia che memorizza il RLC (lo stato del flusso di segnale) in un <operando> specificato. Il connettore immagazzina il risultato degli elementi precedenti. Nei collegamenti in serie con altri elementi, l'operazione ---( # )--- viene inserita come un contatto. L'elemento ---( # )--- non può essere mai collegato alla sbarra di contatto o posizionato direttamente dopo una diramazione aperta, né può essere utilizzato come termine di una diramazione.



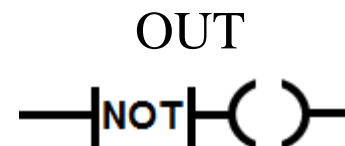
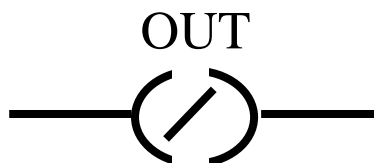
---|NOT|--- (Inverti il risultato della combinazione) inverte il bit RLC.

# Utilizzo dei Coil (bobina di uscita)

- ❖ 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.

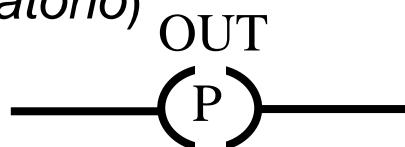


- ❖ **Negated Coil:** la variabile OUT associata al **Negated Coil** è posta a 0 se vi è una corrente che fluisce da sinistra. La variabile rimane a 0 per qualunque scansione del Programma Ladder fino a quando la corrente cessa di fluire da sinistra.



# Altri Coil del Linguaggio Ladder

- ❖ **Positive Transition-Sensing Coil** - Rileva fronte RLC di salita  
(risultato logico combinatorio)



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** - Rileva fronte RLC di discesa  
(risultato logico combinatorio)<sup>OUT</sup>

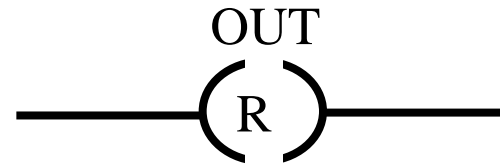


La variabile OUT associata al **Negative Transition-Sensing Coil** è posta a 1 se la corrente che fluisce da sinistra passa da un valore TRUE ad un valore FALSE. La variabile rimane a 1 solo per una scansione del Diagramma Ladder (quella relativa alla transizione dello stato della corrente).

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

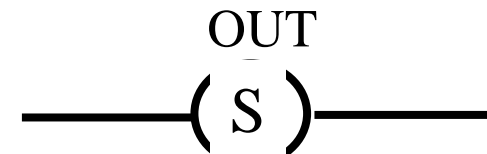
# Altri Coil del Linguaggio Ladder

## ❖ Resetta Bobina



---( R ) (Resetta uscita) viene eseguita solo se il RLC dell'operazione precedente è "1" (flusso di segnale alla bobina). Se la corrente può raggiungere la bobina (RLC è "1"), l'<operando> specificato viene impostato a "0". Se RLC è "0" (nessun flusso di segnale alla bobina), esso non ha effetto; lo stato di segnale dell'operando specificato rimane inalterato. L'<operando> può anche essere un temporizzatore (Nr. T) il cui valore di tempo viene impostato a "0", od un contatore (Nr. Z), il cui valore di conteggio viene impostato a "0".

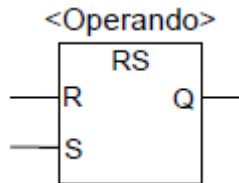
## ❖ Setta Bobina (imposta uscita)



---( S ) (Imposta uscita) viene eseguita solo se il RLC dell'operazione precedente è "1" (flusso di segnale alla bobina). Se RLC è "1", l'<operando> specificato dell'elemento viene impostato ad "1". RLC = 0 non ha alcun effetto; lo stato di segnale dell'operando specificato rimane inalterato.

# Flip-Flop RS nel Linguaggio Ladder

## ❖ Resetta Imposta flip flop



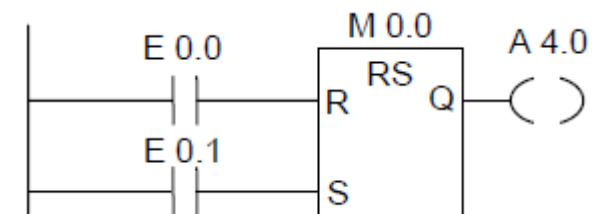
Per analizzare il funzionamento del **flip-flop RS** (Resetta imposta flip flop) vediamo questo esempio:

Se lo stato all'ingresso (R) **E 0.0** è "1" e all'ingresso (S) **E 0.1** è "0", il merker (operando) **M 0.0** viene resettato e l'uscita (Q) **A 4.0** è "0".

Invece, se lo stato di segnale all'ingresso **E 0.0** = 0 e all'ingresso **E 0.1** = 1, il merker **M 0.0** viene settato, e **A 4.0** diventa "1".

Se entrambi gli stati di segnale sono "0", non viene modificato nulla.

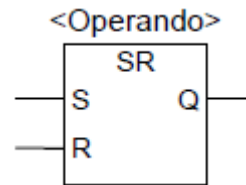
Se entrambi gli stati di segnale sono "1", l'operazione Imposta prevale per via dell'ordine di esecuzione. **M 0.0** viene settato e **A 4.0** è "1".



R	S	Q	Op.
1	0	0	R
0	1	1	S
0	0	No change	No change
1	1	1	S

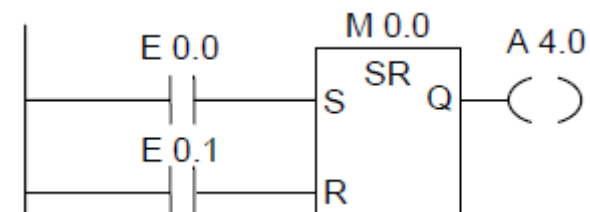
# Flip-Flop SR nel Linguaggio Ladder

## ❖ Imposta Resetta flip flop



Per analizzare il funzionamento del **flip-flop SR** (Imposta resetta flip flop) vediamo questo esempio:

Se lo stato all'ingresso (S) **E 0.0** è "1" e all'ingresso (R) **E 0.1** è "0", il merker (Operando) **M 0.0** viene resettato e l'uscita **A 4.0** è "0".



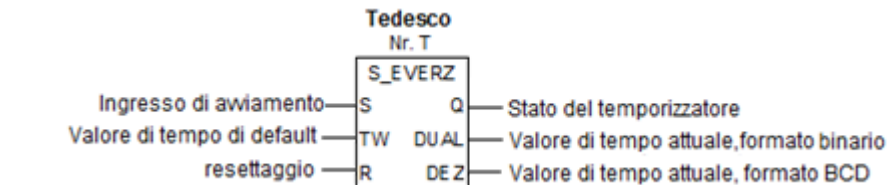
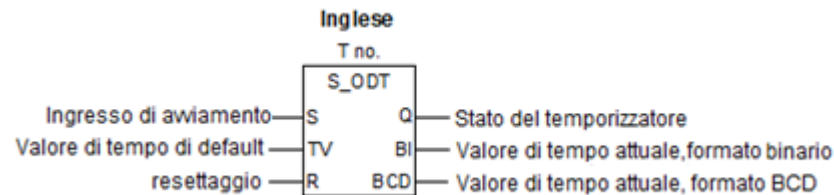
Invece, se lo stato di segnale all'ingresso **E 0.0** = 0 e all'ingresso **E 0.1** = 1, il merker **M 0.0** viene impostato, e **A 4.0** è "1".

Se entrambi gli stati di segnale sono "0", non viene modificato nulla.

Se entrambi gli stati di segnale sono "1", l'operazione Imposta prevale per via dell'ordine di esecuzione. **M 0.0** viene impostato e **A 4.0** è "1".

R	S	Q	Op.
1	0	0	R
0	1	1	S
0	0	No change	No change
1	1	0	R

# Temporizzatore S\_ODT/S\_EVERZ



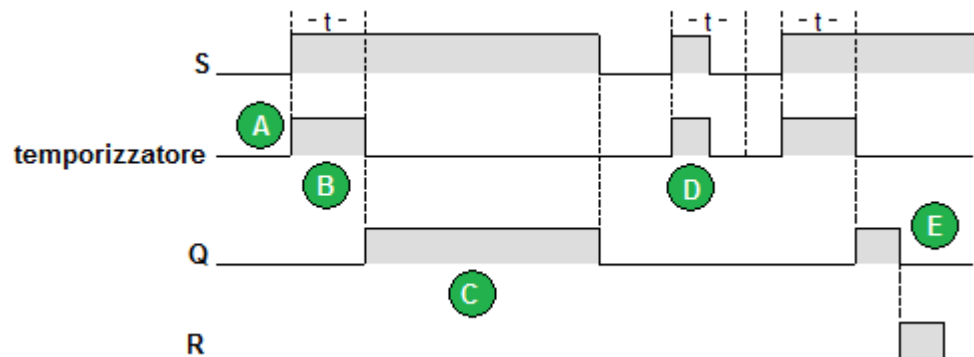
Si può precaricare un **valore di tempo** in uno dei seguenti formati:

- **W#16#wxyz** → dove w = base di tempo (ossia, l'intervallo di tempo o risoluzione) e xyz = valore in formato BCD
- **S5T#aH\_bM\_cS\_dMS** → dove 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\_EVERZ/S\_ODT

(A) Il temporizzatore viene avviato se all'ingresso di avviamento S si trova un fronte di salita (da 0 passa a 1). Per abilitare un temporizzatore è sempre necessaria una modifica di segnale **S**.

(B) Il temporizzatore continua ad operare per tutta la durata del valore di tempo indicato all'ingresso **TV/TW**, fino a quando lo stato di segnale all'ingresso **S** rimane positivo.

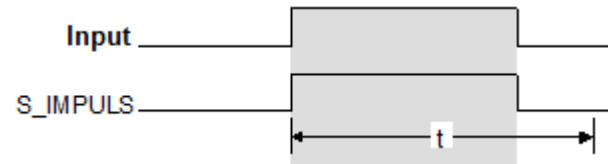
(C) Lo stato di segnale dell'uscita **Q** è "1" se il tempo **TV/TW** è trascorso senza errori e lo stato di segnale dell'ingresso **S** è rimasto nel frattempo pari a "1".

(D) Se lo stato di segnale all'ingresso **S** cambia da "1" a "0" mentre il temporizzatore è attivo, il temporizzatore viene arrestato. In questo caso, lo stato di segnale all'uscita **Q** è "0".

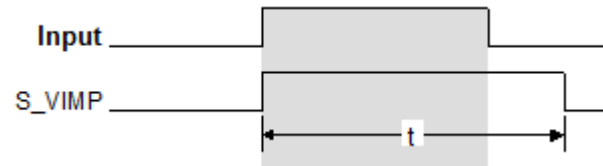
(E) Inoltre, il temporizzatore viene resettato anche quando lo stato di segnale all'ingresso di resettaggio R è "1", mentre il temporizzatore non è attivo, ed RLC è a "1" all'ingresso **S**. Lo stato di segnale all'uscita **Q** è quindi "0". Mentre il temporizzatore è attivo, esso viene resettato se vi è un cambiamento da "0" a "1" all'ingresso di resettaggio R. Anche il valore di tempo attuale BI e **BCD** vengono impostati a zero. Lo stato di segnale all'uscita **Q** è quindi "0".

Il valore di tempo attuale può essere letto alle uscite BI/DUAL e BCD/DEZ. Il valore di tempo all'uscita BI/DUAL è in formato binario, il valore all'uscita BCD/DEZ è in formato di cifra decimale in codice binario. Il **valore di tempo attuale** corrisponde al valore iniziale di TV/TW, dal quale viene sottratto il valore di tempo trascorso dal momento di avvio del temporizzatore.

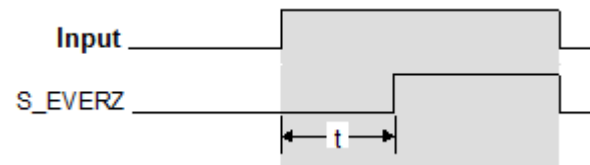
# Vari temporizzatori (Q)



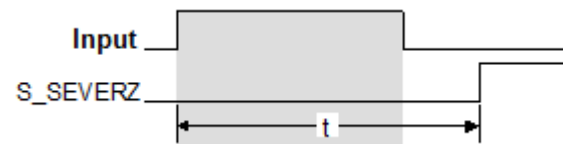
Il tempo massimo in cui il segnale di uscita resta a 1, è uguale al valore di tempo programmato  $t$ . Il segnale di uscita resta a 1 per un tempo più breve se il segnale di ingresso passa a 0.



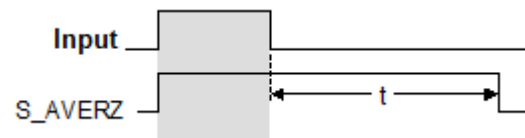
Il segnale di uscita resta a 1 per la durata programmata, indipendentemente dal tempo che il segnale di ingresso resta a 1.



**S\_EVERZ/S\_ODT:** Il segnale di uscita è 1 solo quando è trascorso il tempo programmato e il segnale di ingresso è ancora 1.

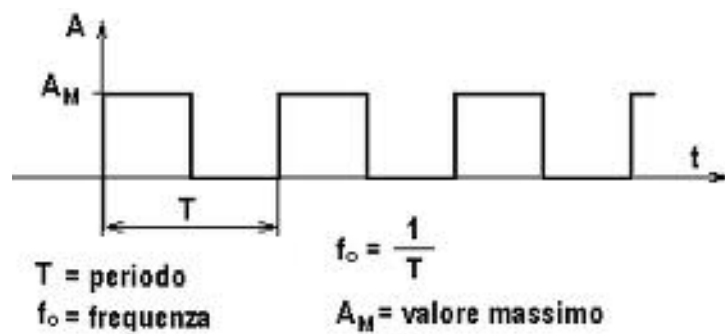
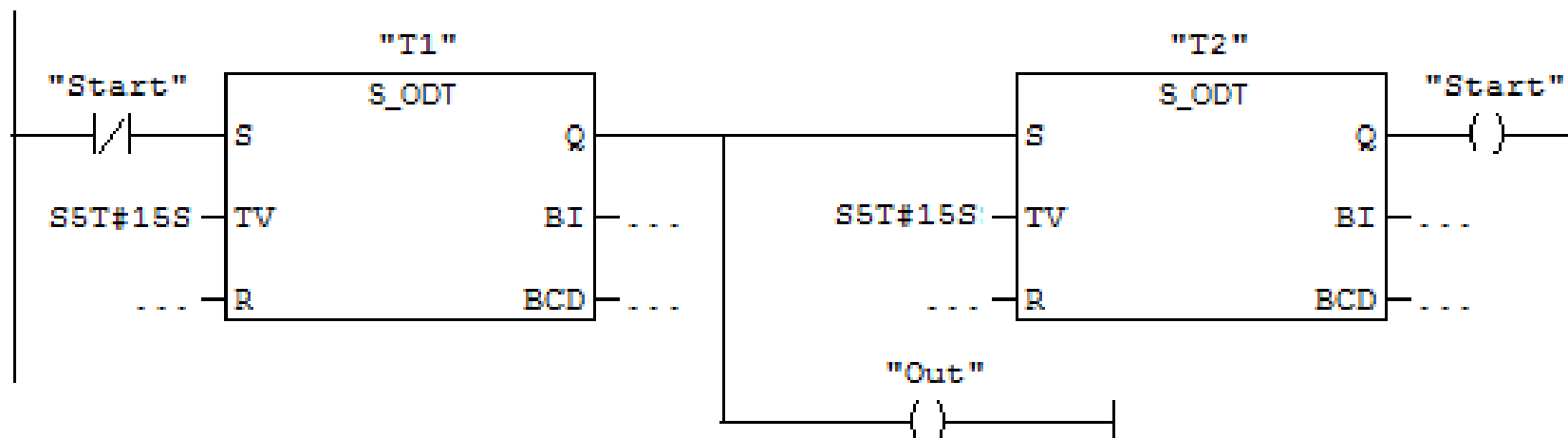


Il segnale di uscita passa da 0 a 1 solo quando è trascorso il tempo programmato, indipendentemente dal tempo in cui il segnale di ingresso resta a 1.



Il segnale di uscita è 1 quando il segnale di ingresso è 1. Il segnale di uscita resta a 1 per la durata programmata. Il tempo viene avviato quando il segnale di ingresso cambia da 1 a 0.

## ESEMPIO DUE: GENERATORE ONDA QUADRA





**FINE 5° LEZIONE**