

Le Funzioni

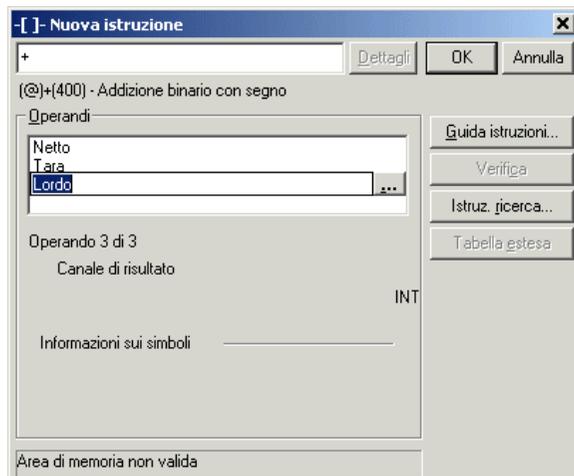
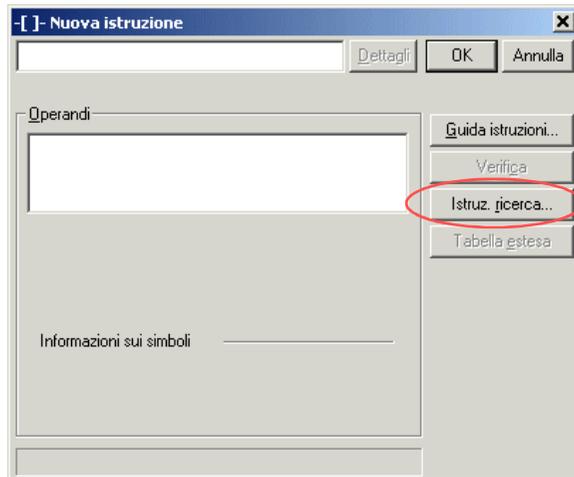
*Istruzioni su Bit, TIM, CNT e
movimento dati*



Le Funzioni

- Rappresentano quella serie di istruzioni per le quali non è previsto sulla console di programmazione il corrispondente tasto dedicato
- Ad ogni funzione è associato, in modo univoco, un codice numerico di tre cifre. E' possibile richiamare una funzione anche inserendo direttamente il codice
- Nel diagramma a relè le funzioni vengono rappresentate con un blocco funzionale in cui compare il nome dell'istruzione, insieme ai relativi parametri
- La serie CJ/CS e CP1 dispone di un set di oltre 400 istruzioni, così suddiviso:
 - 14 Istruzioni Base (schema a contatti)
 - circa 400 Istruzioni Avanzate (funzioni)

Le Funzioni



- Selezionando tra le icone di progetto l'icona  si apre la finestra di immissione di una funzione
- E' possibile richiamare direttamente la funzione, digitando il nome corrispondente, oppure è possibile selezionarla da una lista disponibile facendo clic sul tasto di ricerca
- Una volta selezionata la funzione è possibile inserirne i parametri (se richiesti) e decidere se utilizzare o meno la forma differenziale (se consentito)

Uso delle costanti numeriche

- Occorre prestare attenzione durante l'inserimento di costanti come operandi di istruzioni:
- # $\left\{ \begin{array}{l} \rightarrow \text{valori esadecimali (HEX)} \\ \rightarrow \text{valori BCD} \end{array} \right.$
- & \rightarrow valori decimali senza segno
- \pm \rightarrow valori decimali con segno

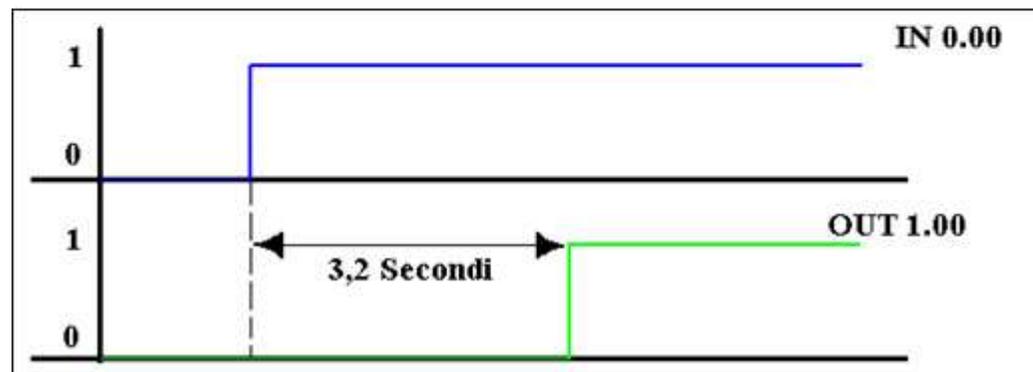
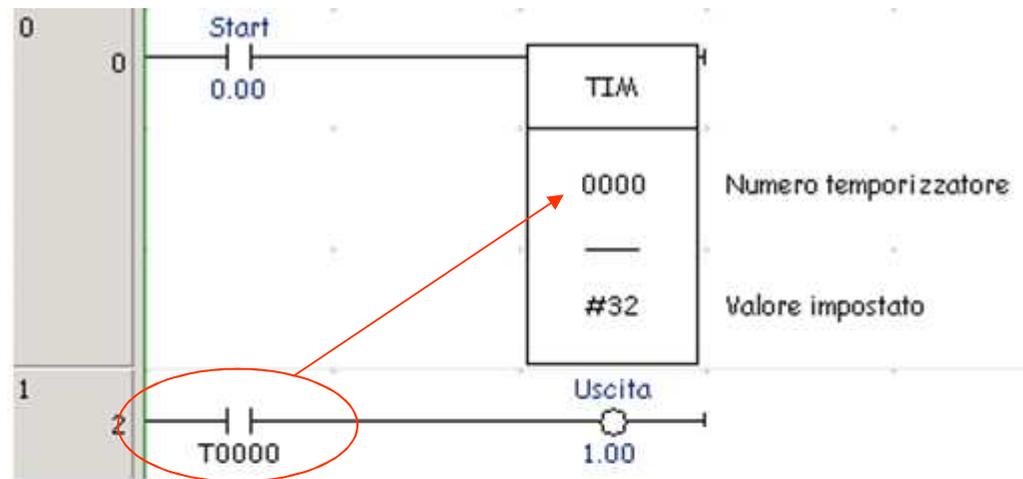
Timer

- L'istruzione TIM (Timer) viene utilizzata per generare un ritardo all'eccitazione rispetto al segnale di abilitazione 'Start'
- Il ritardo SV (Set Value) può variare da 0 a 999,9 secondi ed è programmabile in unità di 0,1 secondi
- Quando il segnale di Start va ad ON il valore attuale PV (Present Value) del timer (inizialmente pari all'SV) inizia a decrementare
- Quando il PV (non ritentivo) diventa uguale a zero, lo stato del timer va ad uno
- Le famiglie di PLC CJ/CS e CP1 dispongono di 4.096 timer (da 0 a 4.095) identificati univocamente dal proprio numero



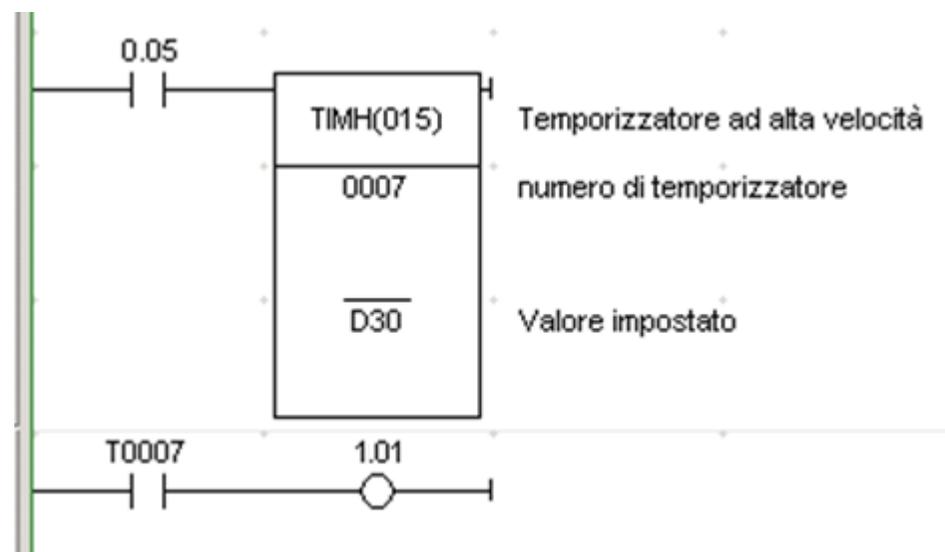
Esempio di utilizzo di un timer

- Il segnale di Start 0.00 eccita l'uscita 1.00 dopo 3,2 secondi



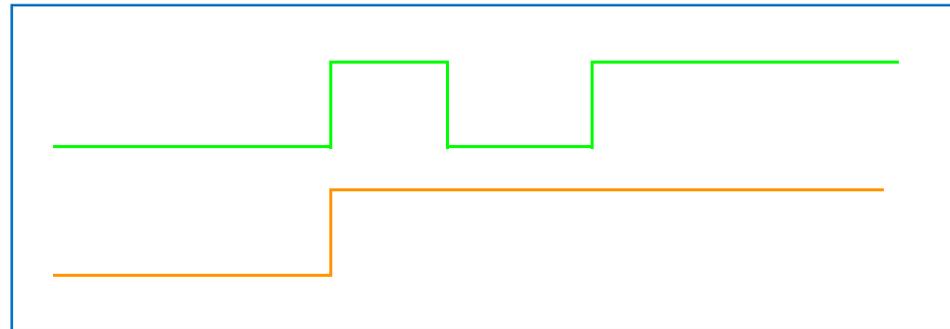
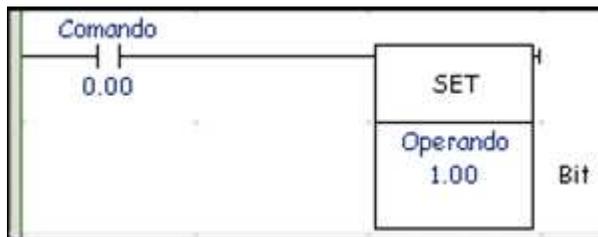
La funzione TIMH (015)

- La funzione TIMH (timer ad alta velocità) serve per generare ritardi all'eccitazione con una precisione maggiore rispetto alla funzione TIM
- Il ritardo (SV) può variare tra 0,00 e 99,99 secondi ed è programmabile in unità di centesimi di secondo

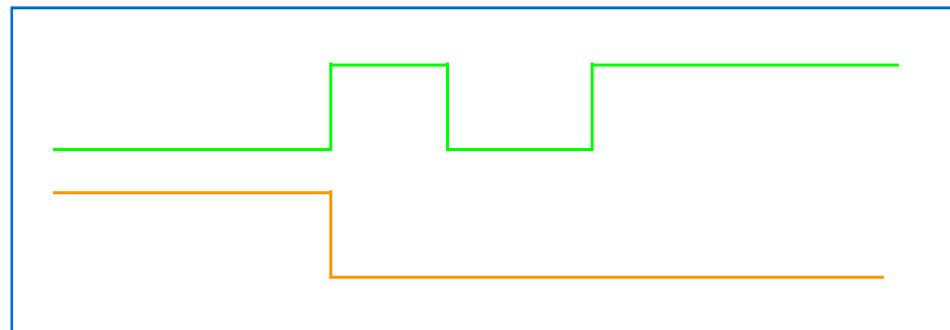
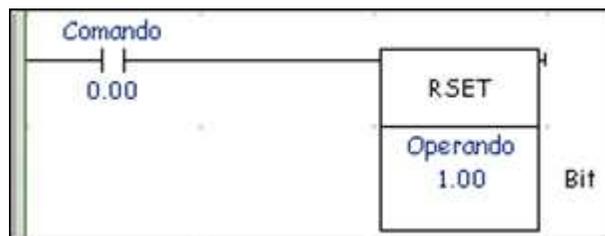


Le istruzioni di SET e RESET

- L'istruzione SET imposta ad 1 il bit indicato come operando



- L'istruzione RSET imposta ad 0 il bit indicato come operando

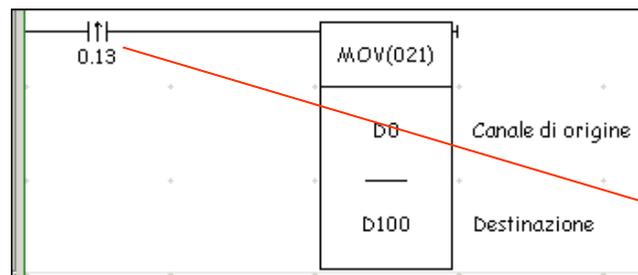


Fronte di salita e funzioni differenziate

- Con fronte di salita (discesa) di un segnale, si intende la condizione per cui, nella scansione precedente quel segnale era stato letto ad OFF (ON) dal PLC e nella scansione attuale viene letto ad ON (OFF)
- Quasi tutte le funzioni integrate nel set di istruzioni del PLC possono essere eseguite in due modalità:
 - ad ogni scansione se è presente il segnale di abilitazione della funzione
 - solo per una scansione sul fronte di salita (o di discesa) del segnale di abilitazione della funzione (funzione differenziata)

Fronte di salita e funzioni differenziate

- Originariamente nel set di istruzioni del PLC erano previste due funzioni utili per intercettare i fronti di salita (DIFU) o di discesa (DIFD) dei segnali, in modo da poter controllare con un bit l'esecuzione di tutte le istruzioni non differenziabili
- A partire dalle famiglie CS/CJ e CP1 è invece possibile differenziare i contatti direttamente durante il loro inserimento, o utilizzare delle istruzioni intermedie (UP e DOWN) consentendo di risparmiare memoria di programma

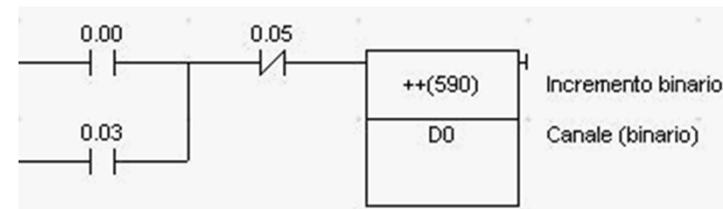


Fronte di salita e funzioni differenziate

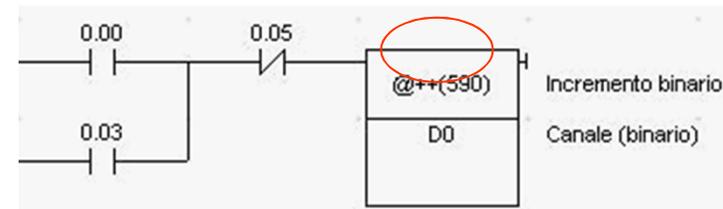
- Per differenziare una funzione è sufficiente inserire il carattere '@' prima di digitarne il nome
- Esempio:



L'istruzione '++' viene eseguita ad ogni scansione se il segnale di abilitazione è ad ON

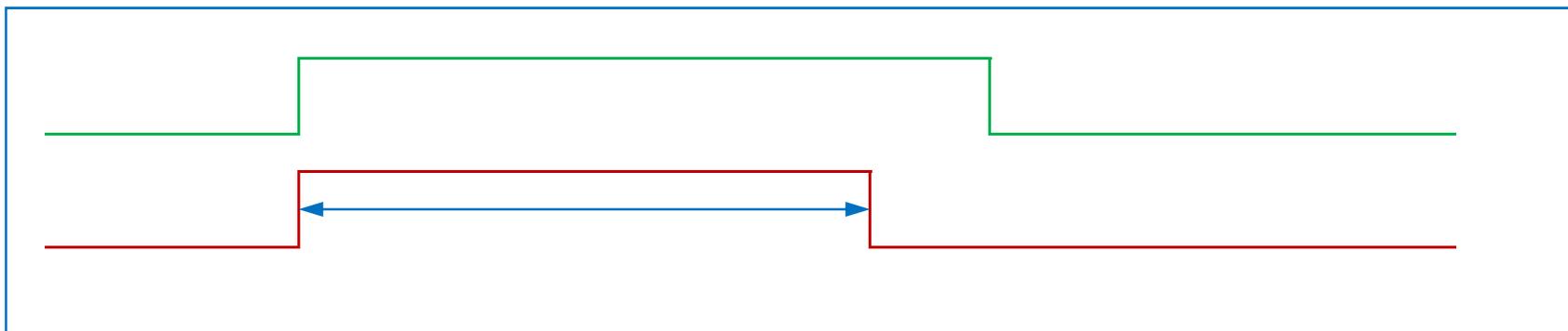
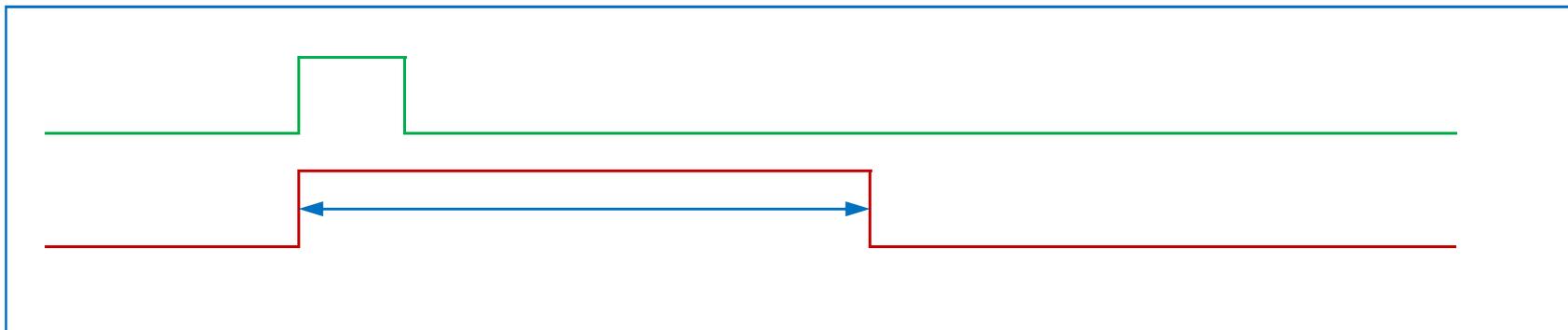


L'istruzione '++' viene eseguita per una sola scansione sul fronte di salita del segnale di abilitazione è ad ON

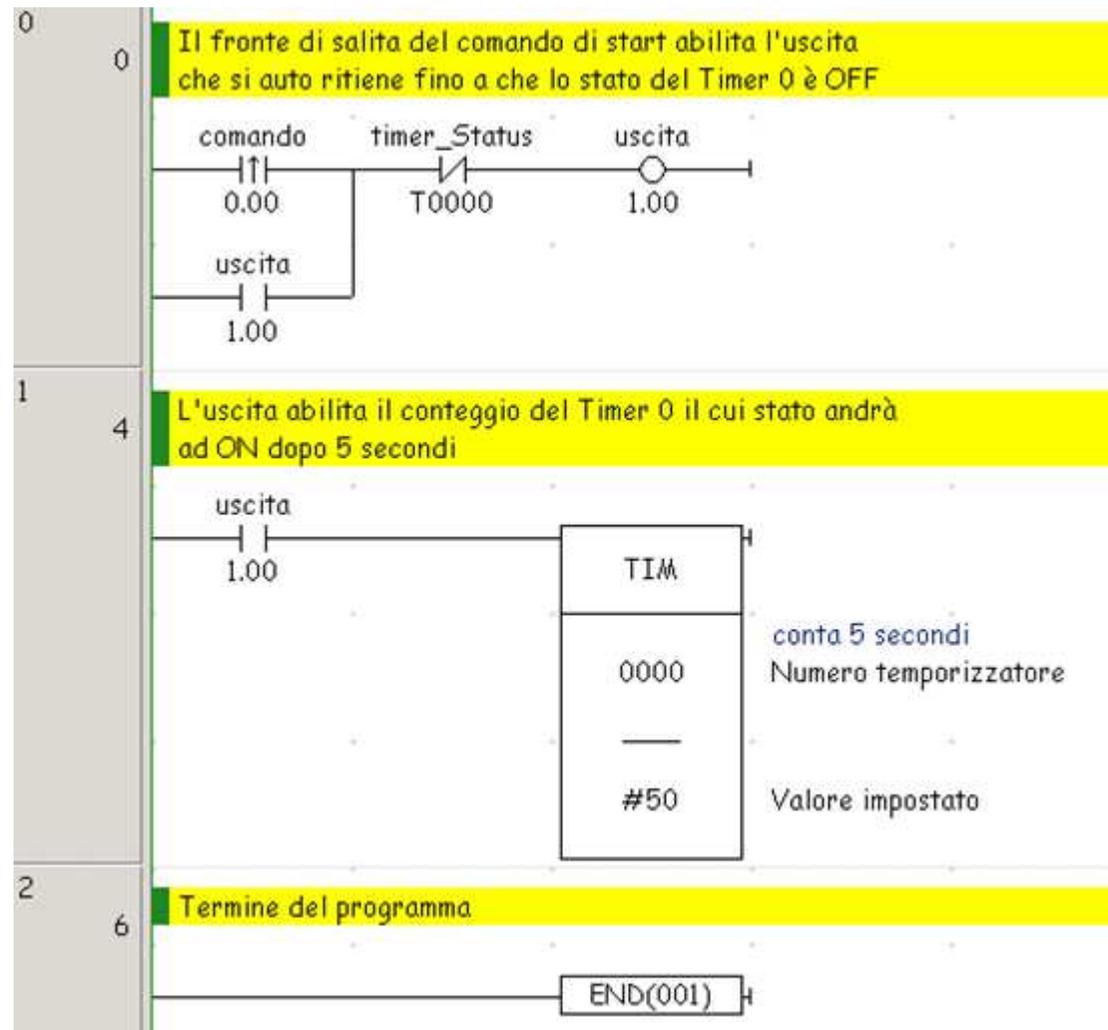


Esercizio 2: Timer monostabile

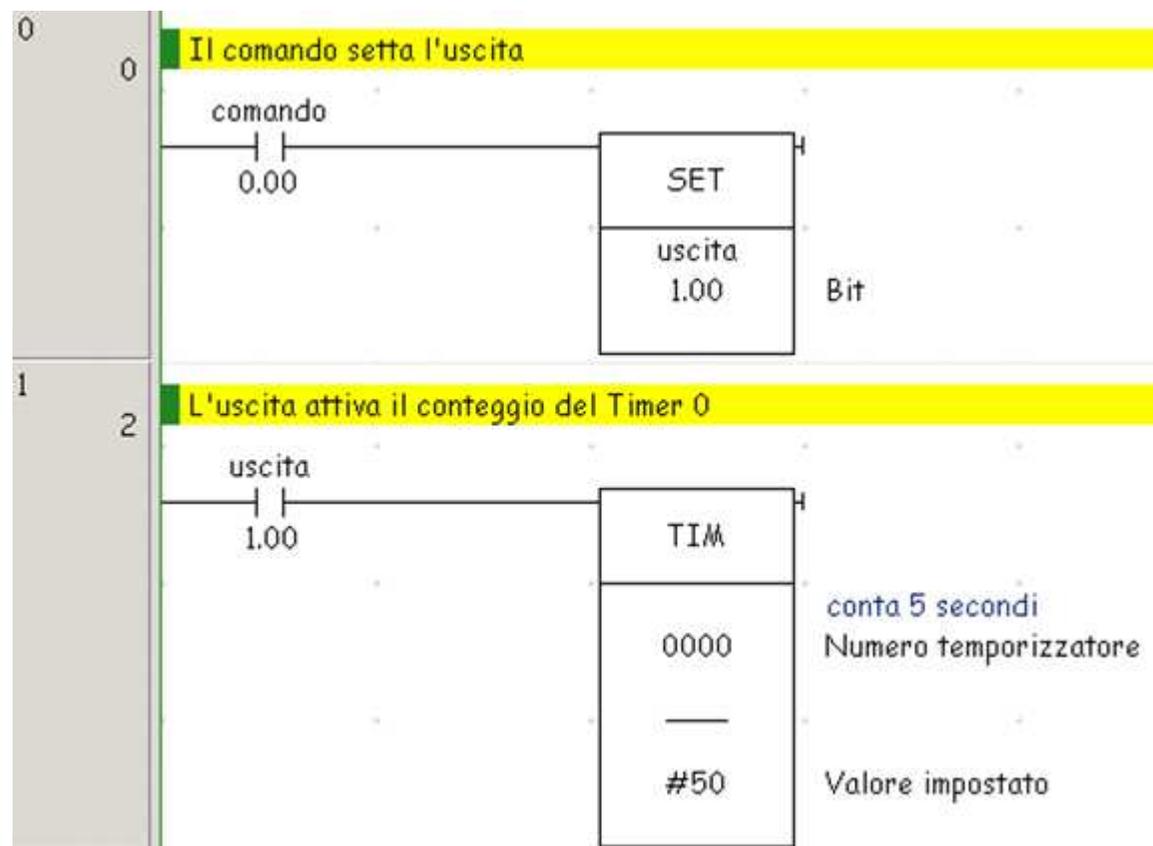
- Realizzare un timer che fissi la durata dell'attivazione di un'uscita a 5 secondi, indipendentemente dalla durata del segnale di Start



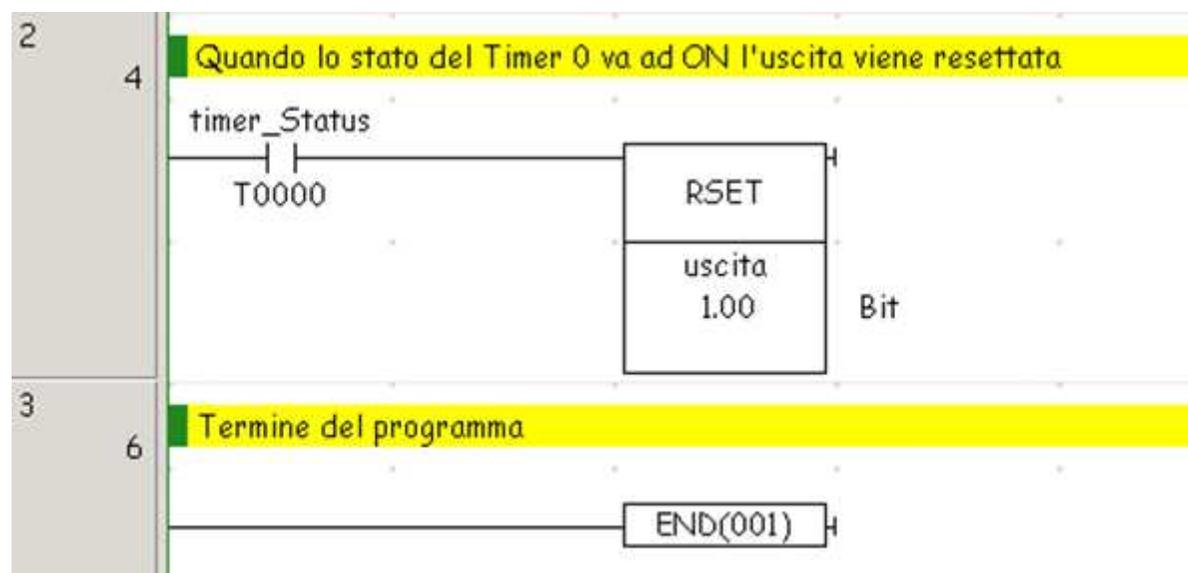
Esercizio 2/A: Soluzione con autoritenuta



Esercizio 2/B: Soluzione con SET e RESET

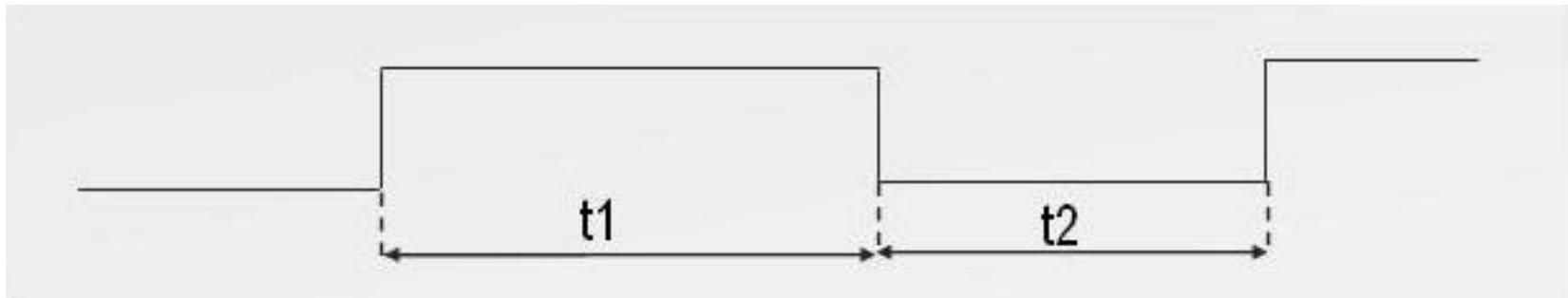


Esercizio 2/B: Soluzione con SET e RESET

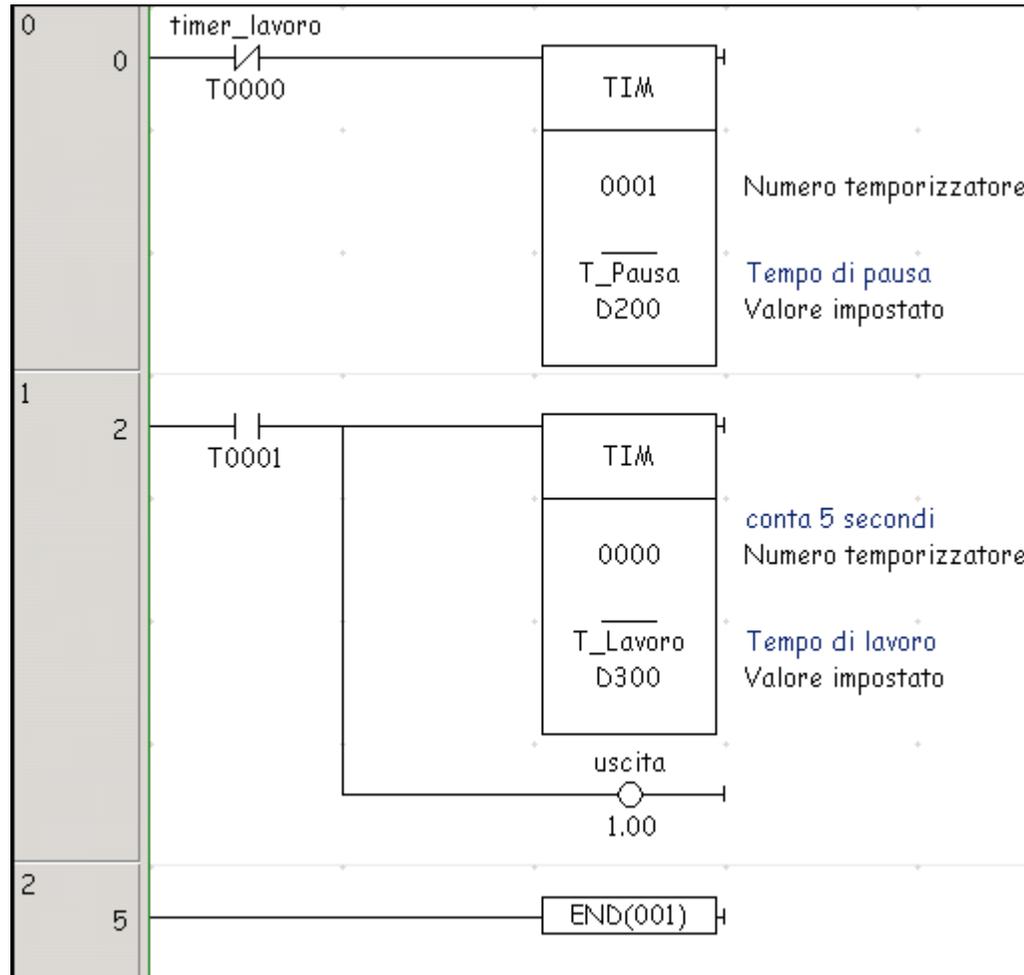


Esercizio 3: Timer Pausa/Lavoro

- Utilizzando due timer in combinazione tra loro, realizzare un'onda con Duty Cycle variabile



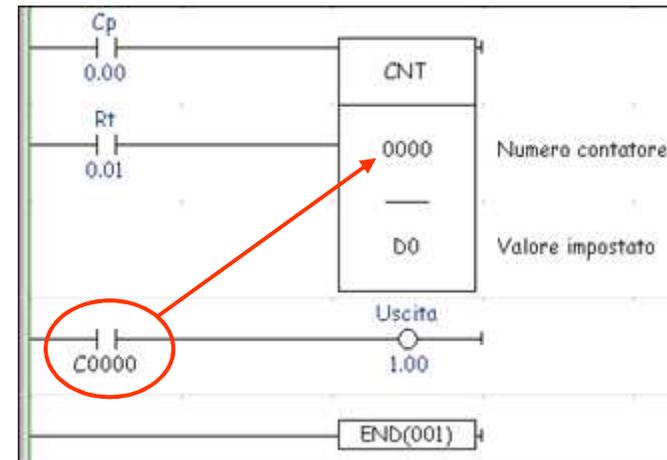
Esercizio 3: Soluzione



La funzione Contatore

- La funzione contatore (CNT) simula un contatore sottraente a predisposizione. Il valore di predisposizione (SV) può variare tra 0 e 9999

- Il contatore ha due ingressi:
Ingresso di conteggio impulsi
Ingresso di reset



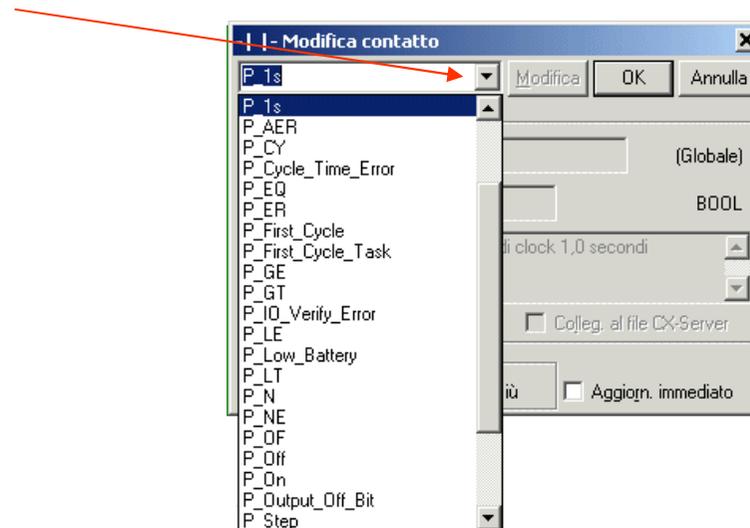
- Se Rt è OFF, ogni fronte di salita di Cp determina il decremento del PV. Quando il PV diventa uguale a zero lo stato del contatore va ad ON

La funzione Contatore

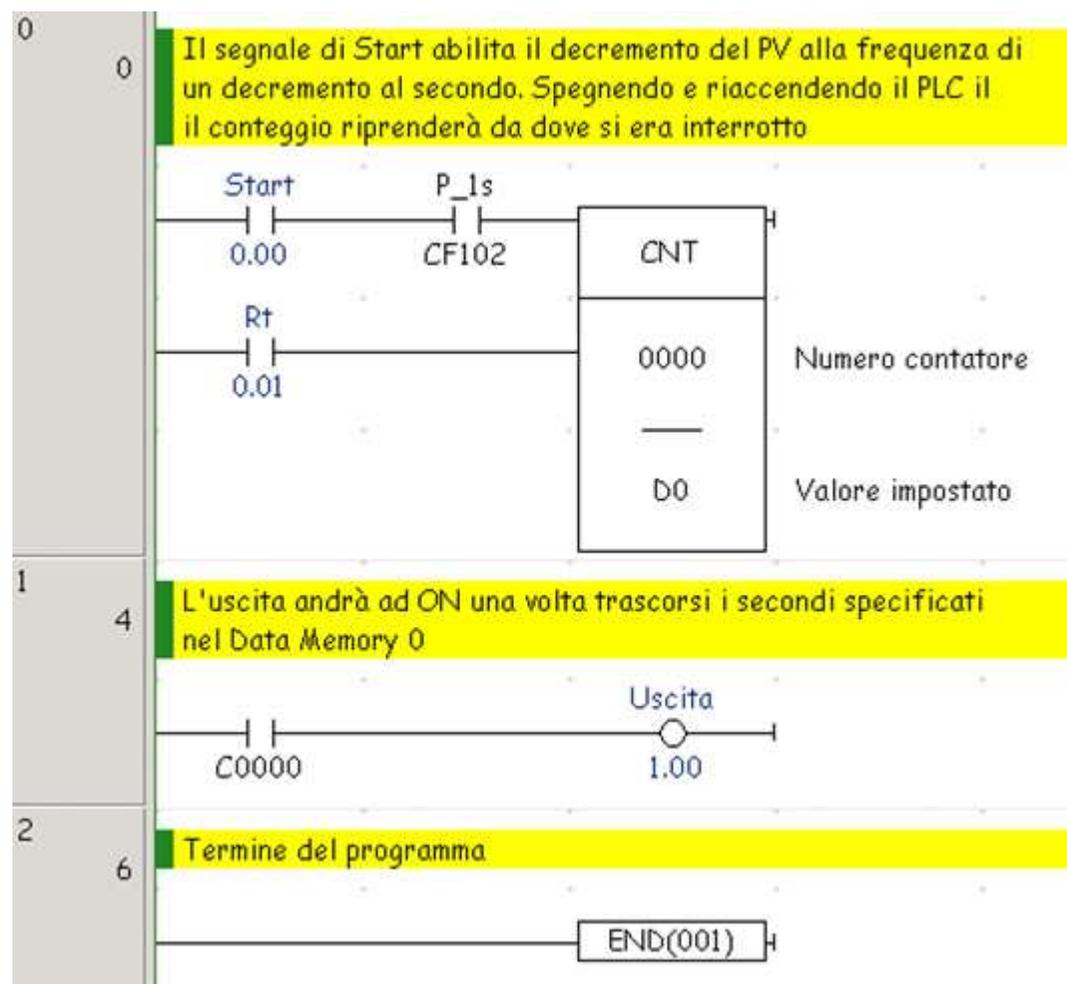
- I contatori sono ritentivi, ovvero il loro stato (bit di uscita e PV) viene mantenuto sia in caso di Power Off che di cambio della modalità operativa del PLC
- Quando il PV è uguale a 0 tutti gli eventuali impulsi di ingresso vengono ignorati
- Se Rt va ad ON il contatore si resetta (PV=SV, stato=OFF)
- Accoppiato ad un clock di sistema, un contatore può essere utilizzato per realizzare un timer ritentivo
- Accoppiando due contatori in cascata tra loro, si ottiene un conteggio pari al prodotto di SV1 e SV2

Esercizio 4: Realizzare un Timer ritentivo

- Utilizzare un contatore abbinato ad un clock di sistema di un secondo (P_1s)
- Nel PLC sono presenti molti flag di sistema (clock a varie frequenze, flag di errore, bit sempre ad ON o ad OFF, ecc)
- L'elenco completo è disponibile nella maschera di inserimento di un nuovo contatto, cliccando sulla freccia di estensione della tendina a scorrimento



Esercizio 4: Soluzione



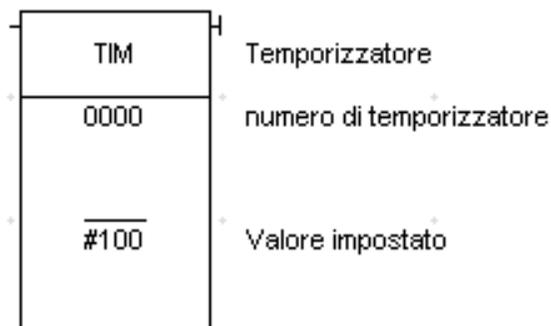
A blue vertical bar is positioned on the left side of the slide, extending from the top to the bottom. It has a slight 3D effect with a darker blue shadow on its right side.

Timer e Counter con impostazione Esadecimale

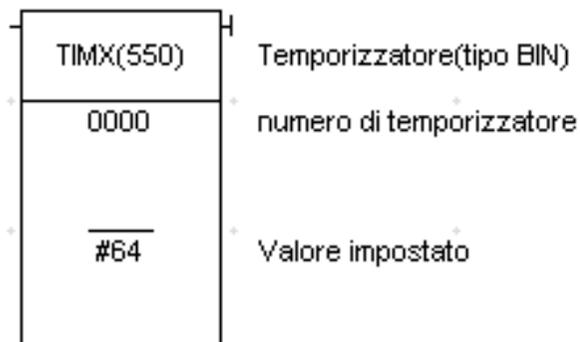
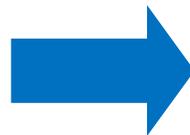
- Nei PLC serie CJ/CS e CP1 è possibile impostare timer e counter con valori binari esadecimali
- Questa possibilità estende il campo di conteggio/temporizzazione delle istruzioni timer counter abituali
- Si passa da un range di impostazione da 0 a 9999 per il BCD a un range da 0 a 65535 per la modalità binaria
- Nei PLC precedenti l'impostazione poteva avvenire solo in BCD

Timer e Counter con impostazione Esadecimale

- Le due modalità di impostazione BCD e Binaria esadecimale non possono coesistere all'interno dello stesso programma; esistono perciò delle istruzioni Timer e Counter dedicate alla modalità esadecimale
- Solo nel caso dei PLC della serie CJ2 le due modalità possono convivere



Modalità BCD 10 sec. SV=#100



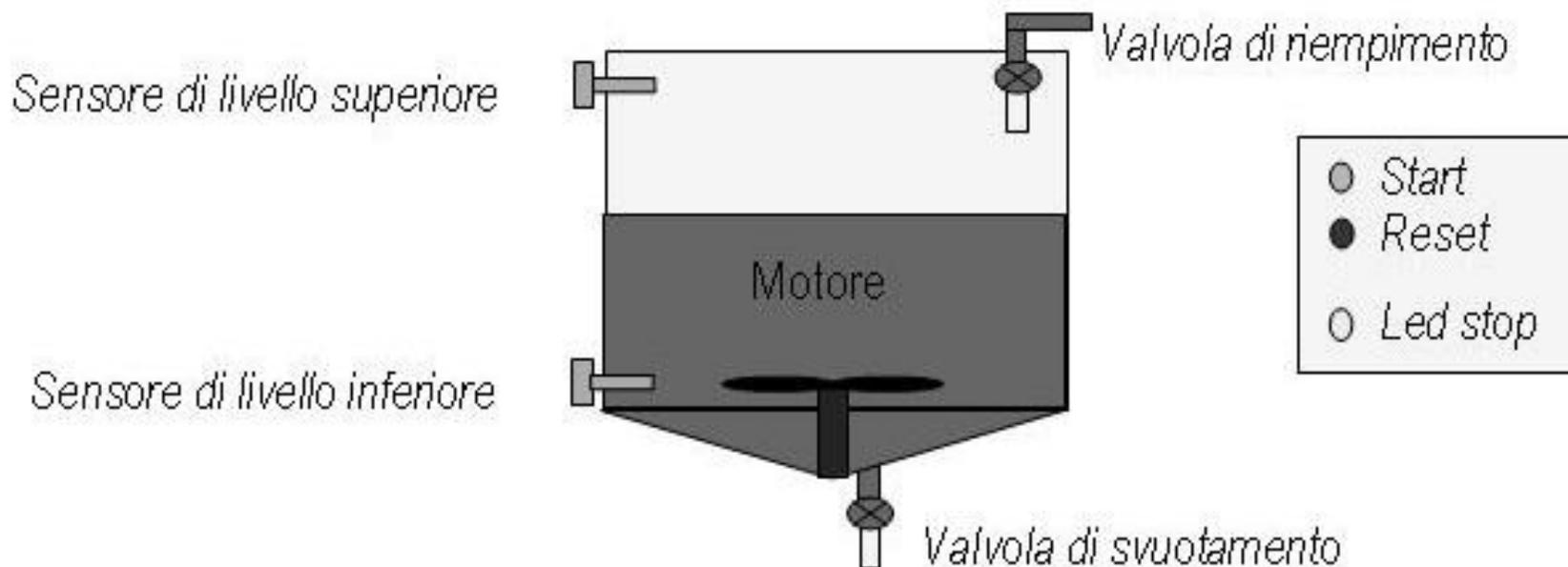
Modalità Hex 10 sec. SV=#64
Oppure SV=&100

Timer e Counter con impostazione Esadecimale

- Durante la stesura del progetto è possibile definire se la modalità di lavoro dei temporizzatori e dei contatori debba essere BCD o Esadecimale (binaria)
- CX-Programmer compilerà correttamente solo il set di istruzioni per temporizzatori e contatori relativo alla modalità specificata

Nome	Mnemonico	
	Modalità BCD	Modalità Binaria
TIMER (100 ms)	TIM	TIMX(550)
HIGH-SPEED TIMER (10 ms)	TIMH(015)	TIMHX(551)
ONE-MS TIMER (1 ms)	TMHH(540)	TMHHX(552)
ACCUMULATIVE TIMER (100 ms)	TTIM(087)	TTIMX(555)
LONG TIMER (100 ms)	TIML(542)	TIMLX(553)
MULTI-OUTPUT TIMER (100 ms)	MTIM(543)	MTIMX(554)
COUNTER	CNT	CNTX(546)
REVERSIBLE COUNTER	CNTR(012)	CNTRX(548)
RESET TIMER/ COUNTER	CNR(545)	CNRX(547)

Esercizio 5: Ciclo di riempimento - svuotamento

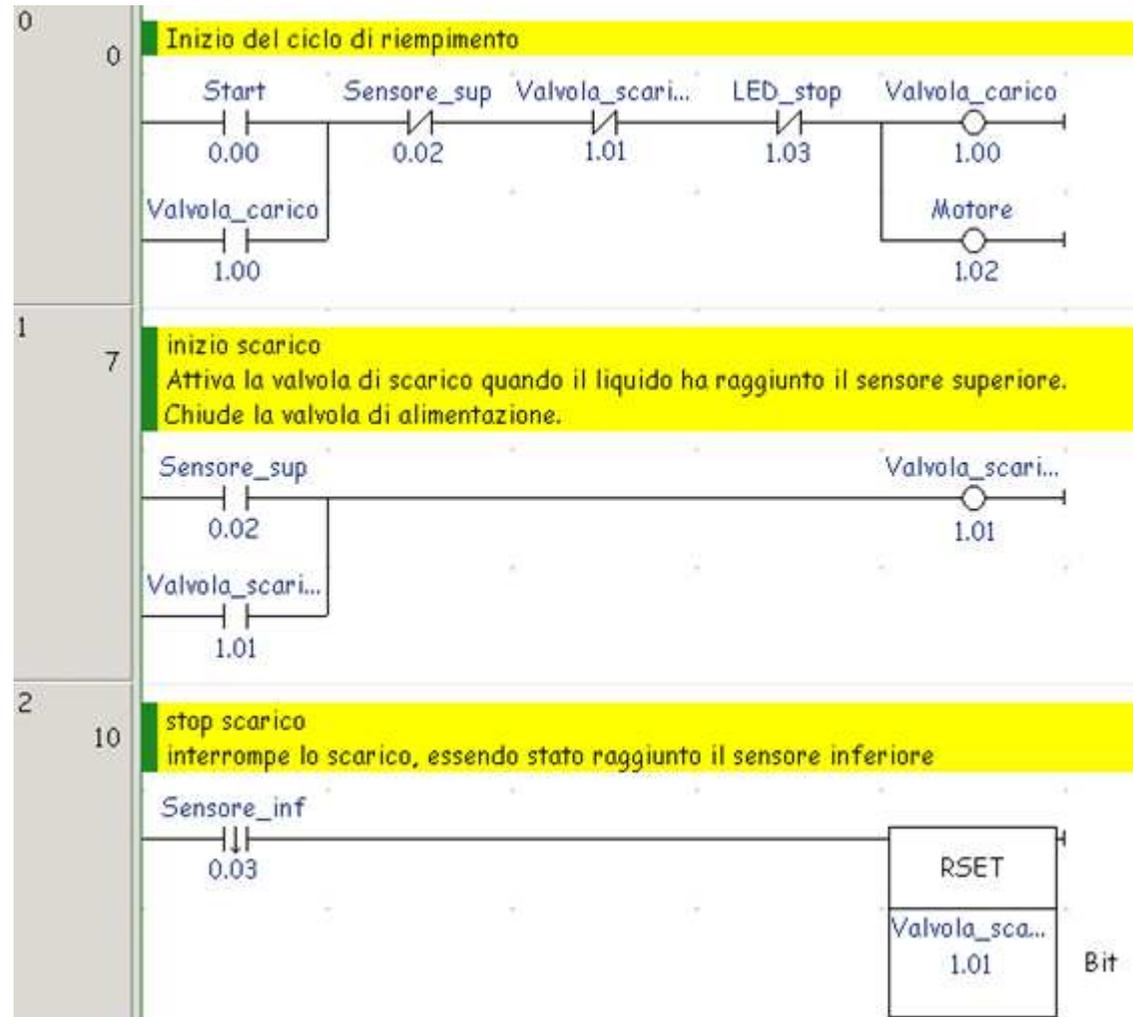


* LED_stop	BOOL	1.03	Lavoro	Indica il termine del ciclo
* Motore	BOOL	1.02	Lavoro	Motore di miscelazione
* Reset	BOOL	0.01	Lavoro	Comando di reset del ciclo
* Sensore_inf	BOOL	0.03	Lavoro	Sensore di livello inferiore
* Sensore_sup	BOOL	0.02	Lavoro	Sensore di livello superiore
* Start	BOOL	0.00	Lavoro	Pulsante di start
* Valvola_carico	BOOL	1.00	Lavoro	Valvola di alimentazione acqua
* Valvola_scarico	BOOL	1.01	Lavoro	Valvola di svuotamento del serbatoio

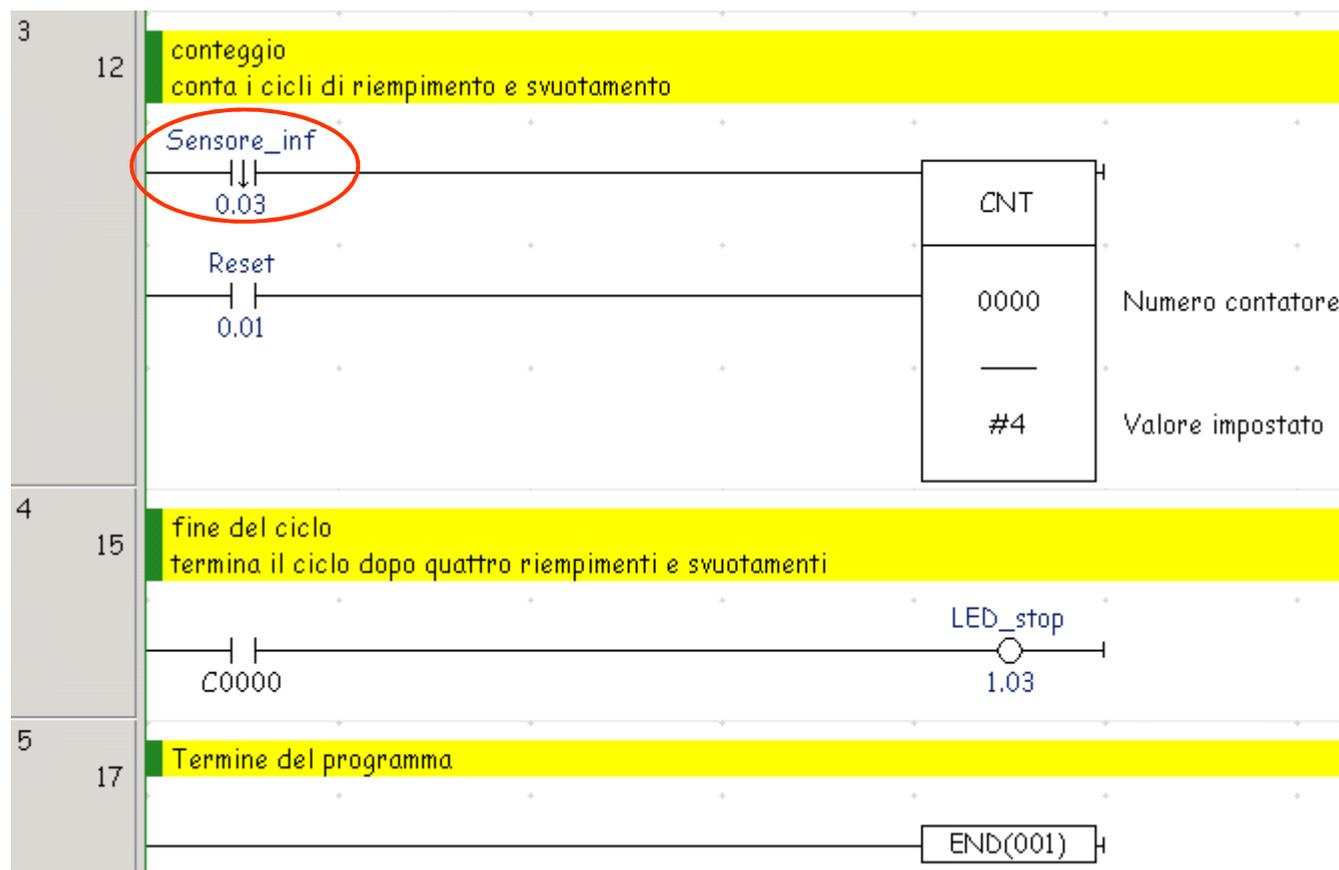
Esercizio 5: Ciclo di riempimento - svuotamento

- Alla pressione del pulsante di Start (0.00), si apre la valvola di riempimento (1.00) e si mette in moto il motore di miscelazione (1.02)
- Quando il livello raggiunge il sensore superiore (0.02), la valvola di riempimento si chiude, il motore si ferma e si apre la valvola di svuotamento (1.01)
- Quando il livello raggiunge il sensore inferiore (0.03) la valvola di svuotamento si richiude e può iniziare un altro riempimento
- Quando il ciclo è stato ripetuto quattro volte, il led di stop (1.03) si accende e un ulteriore ciclo non può essere iniziato se non dopo aver premuto il pulsante di Reset (0.01)

Esercizio 5: Soluzione

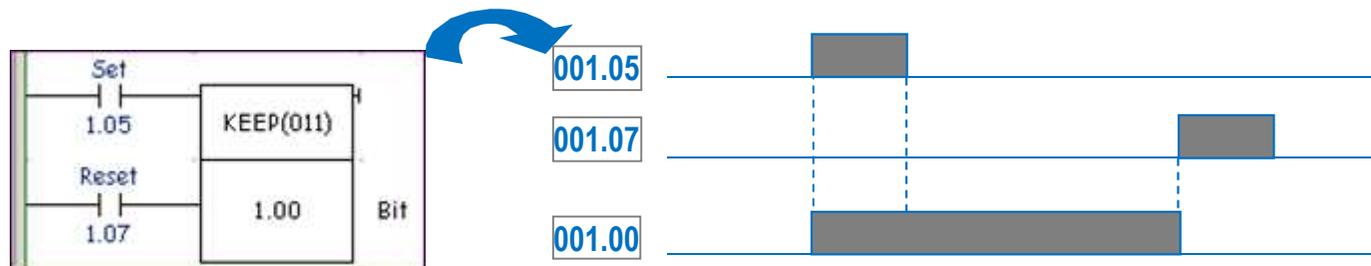


Esercizio 5: Soluzione



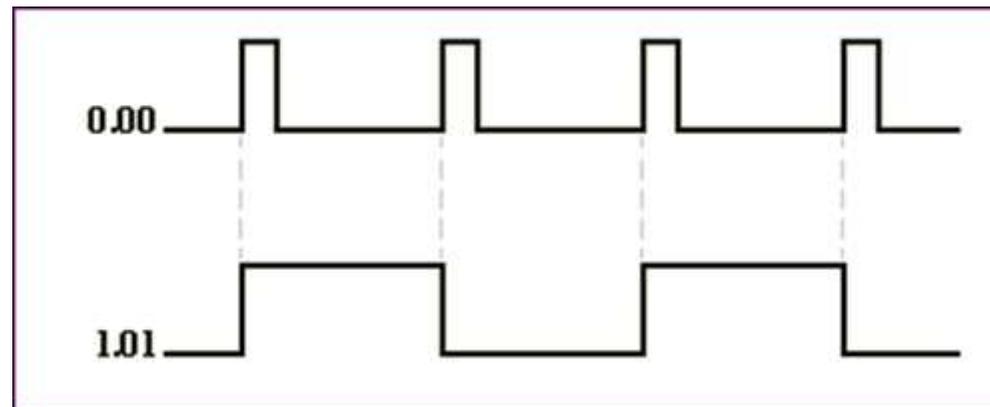
La funzione KEEP

- L'istruzione KEEP permette di realizzare un relè a ritenuta
- Si hanno due ingressi:
 - L'ingresso di Set
 - L'ingresso di Reset
- Se l'ingresso di Reset è ad OFF un impulso su S attiva in modo permanente il bit programmato come operando della funzione
- Un impulso di Reset determina la disattivazione del bit di uscita



Esercizio 6: Toggle

- Fare in modo che l'uscita 1.01 del PLC si attivi e disattivi alternativamente ad ogni fronte di salita del segnale di ingresso 0.00



Esercizio 6: Soluzione



Finestra di controllo

- Per richiamare la finestra di controllo, selezionare l'icona 
- Oppure selezionare "Visualizza" dal menu principale, e poi scegliere la voce "Controlla" del menu "Finestre"
- Oppure premere insieme i tasti "ALT" e "3"

La finestra di controllo

Nome PLC	Nome	Indirizzo	Tipo / Formato dati	Utiliz...	Valore	Valore(binario)
PLC1	NuovoProgramma1.Start	0.00	BOOL (On/Off,Contatto)		0	
PLC1	NuovoProgramma1.Stop	0.01	BOOL (On/Off,Contatto)		1	
PLC1	NuovoProgramma1.Valvola	10.01	BOOL (On/Off,Contatto)		0	
PLC1		D0	CHANNEL (Esa,Canale)		04D2 Esa	0000 0100 1101 0010
PLC1		D100	CHANNEL (Esa,Canale)		1234 Esa	0001 0010 0011 0100

Nome del PLC

Foglio attivo

Nome della variabile

Indirizzo PLC

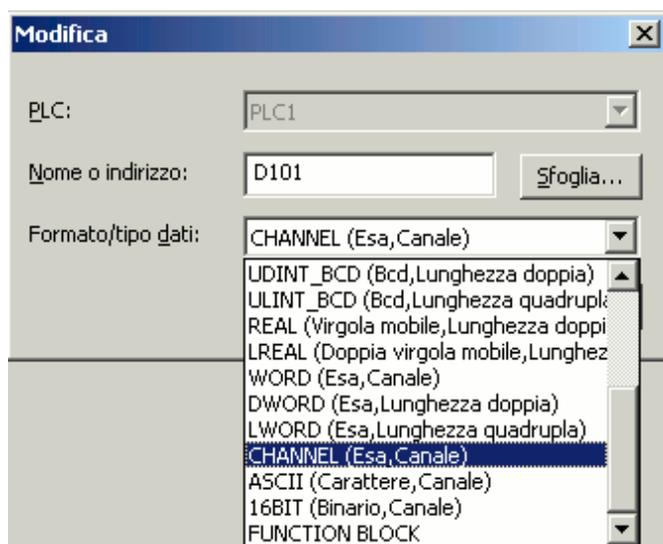
Tipo della variabile e formato di visualizzazione

Rappresentazione binaria del valore

Valore

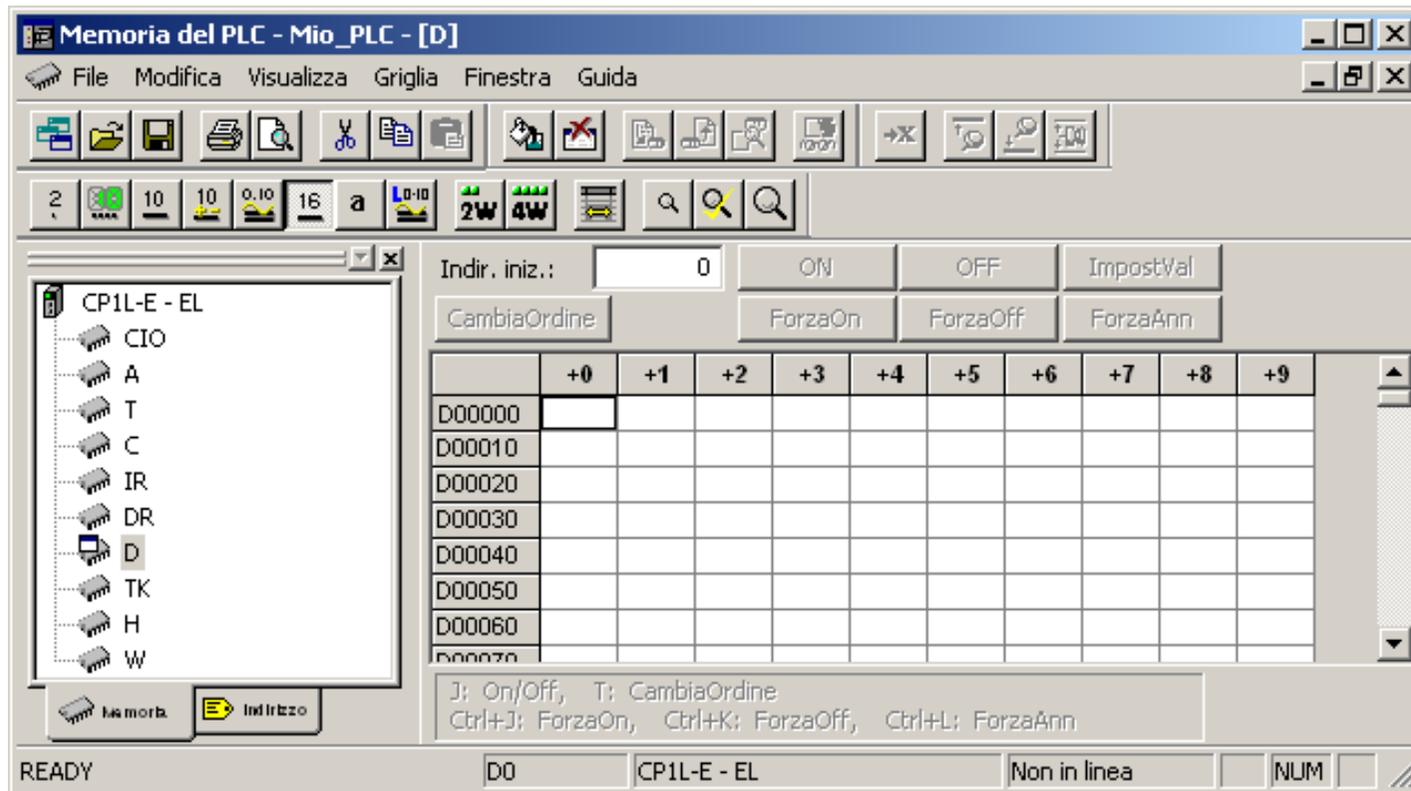
La finestra di controllo

- Per inserire un indirizzo da monitorare, premere “Ins” oppure selezionare una riga vuota e fare doppio click o premere “invio”; verrà mostrata la finestra per l’immissione di un nuovo elemento
- Per modificare un valore online, selezionare la riga desiderata e fare doppio click o premere “invio”; verrà mostrata un finestra per l’immissione del nuovo valore



Editor di memoria

- Per modificare/monitorare una maggior quantità di dati si può ricorrere all'editor di memoria



Editor di memoria

Operazioni Online

Selezione del formato

Selezione dell'area dati

Indir. iniz.:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO0000										
CIO0010										
CIO0020										
CIO0030										
CIO0040										
CIO0050										
CIO0060										
CIO0070										
CIO0080										
CIO0090										
CIO0100										
CIO0110										
CIO0120										
CIO0130										
CIO0140										
CIO0150										
CIO0160										
CIO0170										
CIO0180										

J: On/Off, T: CambiaOrdine
 Ctrl+J: ForzaOn, Ctrl+K: ForzaOff, Ctrl+L: ForzaAnn

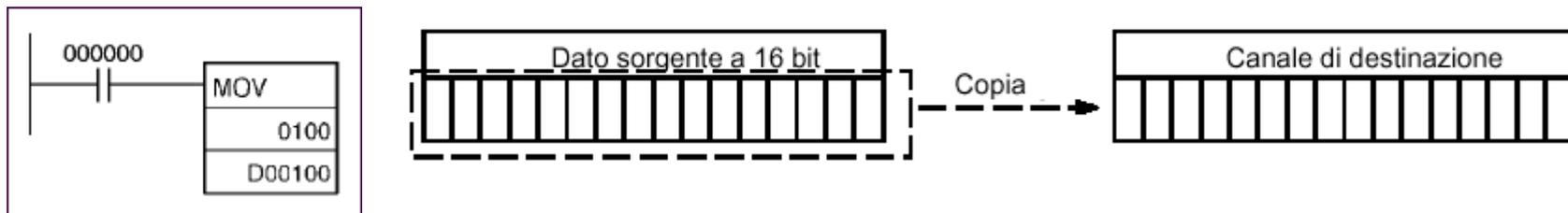
READY CIO0 CP1L-E - EL Program NUM

Selezione dell'indirizzo

Visualizzazione del contenuto dell'area dati

La funzione MOV

- L'istruzione MOV realizza la copia di un dato a 16 bit da un canale sorgente S ad un canale destinazione D
- Le aree utilizzabili per il trasferimento sono:
 - CIO, W, HR, A, D, T, C
- Se S=0 il flag di uguale P_EQ va ad ON subito dopo l'esecuzione dell'istruzione

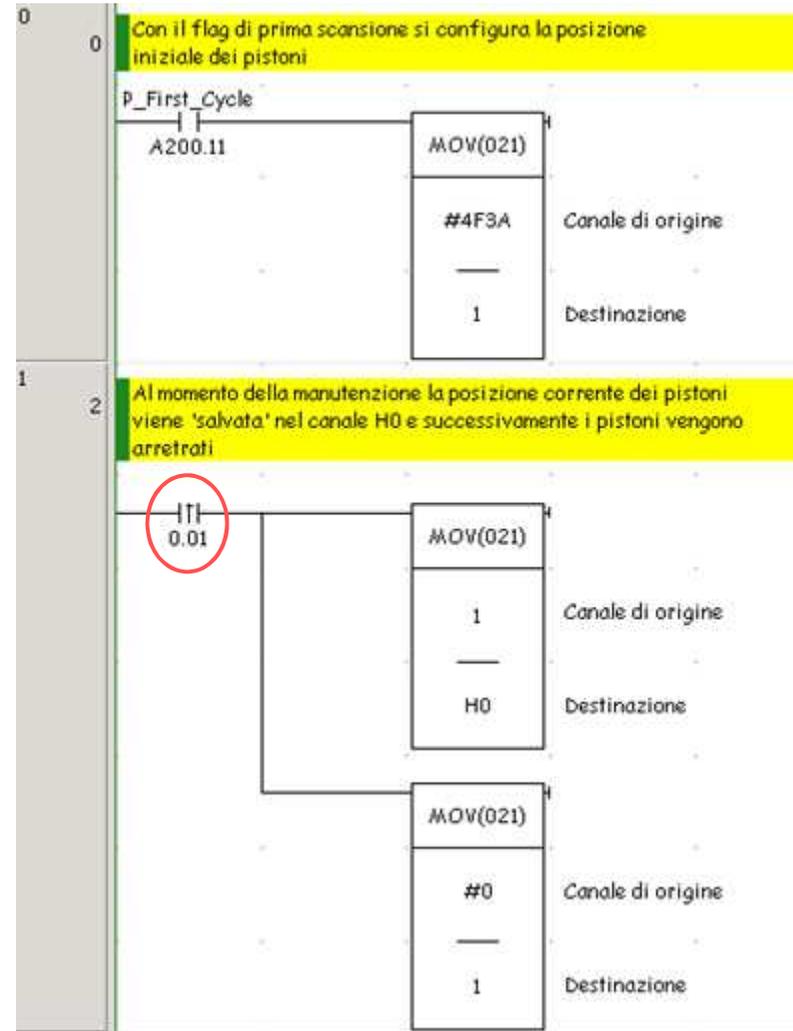


- Esiste anche la funzione MVN che effettua uno spostamento dati NEGANDO però il dato stesso

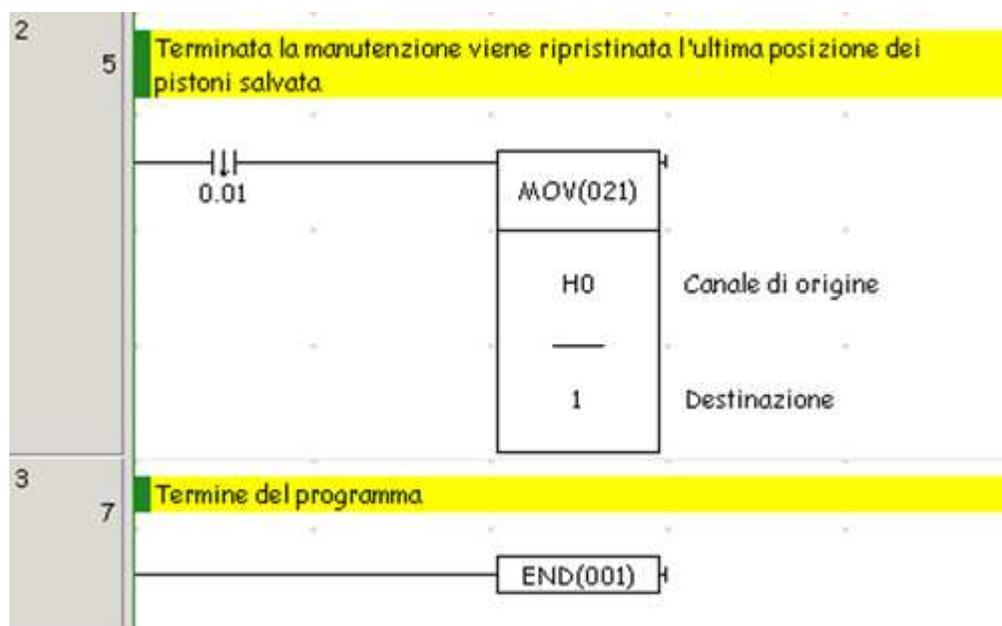
Esercizio 7: Zero macchina

- Fare in modo che all'accensione il canale 1 di uscita del PLC abbia il seguente aspetto: 0 1 0 0 1 1 1 1 0 0 1 1 1 0 1 0 sapendo che ad ogni bit di uscita corrisponde la posizione avanzata (1) o arretrata (0) di un pistone
- Per poter mantenere la macchina si deve dare la possibilità, chiudendo ad esempio il contatto 0.01 di portare tutti i pistoni in posizione arretrata
- E' necessario memorizzare la posizione dei pistoni prima della manutenzione (utilizzare il canale H0 come 'memoria della posizione') per poterla ripristinare al termine della manutenzione (contatto 0.01 aperto)

Esercizio 7: Soluzione



Esercizio 7: Soluzione

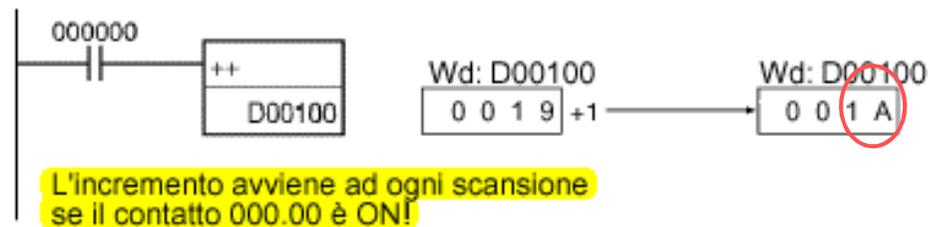


Istruzioni di Incremento e Decremento

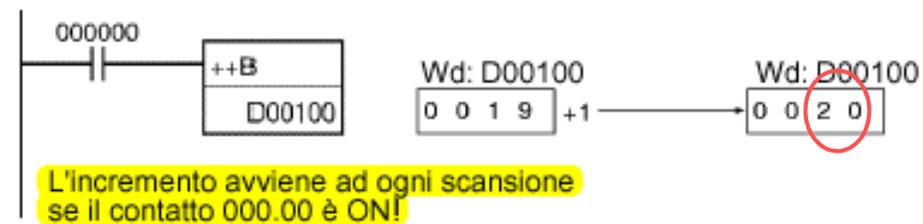
- Istruzioni di incremento e decremento in **esadecimale**:
 - ++ : Incrementa in Esadecimale il contenuto di una word
 - : Decrementa in Esadecimale il contenuto di una word
 - ++L : Incrementa in Esadecimale il contenuto di due word
 - L : Decrementa in Esadecimale il contenuto di due word
- Istruzioni di incremento e decremento in **BCD**:
 - ++B : Incrementa in BCD il contenuto di una word
 - B : Decrementa in BCD il contenuto di una word
 - ++BL : Incrementa in BCD il contenuto di due word
 - BL : Decrementa in BCD il contenuto di due word

Istruzioni di Incremento e Decremento

- Esempio Incremento Binario (++):



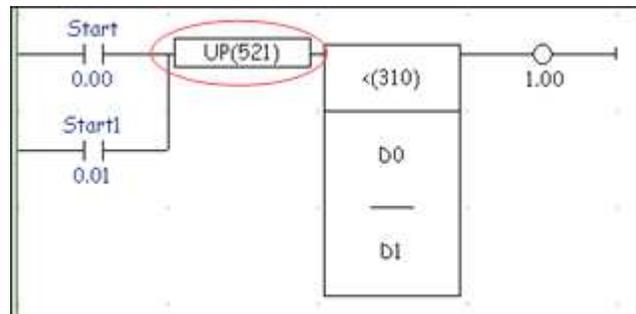
- Esempio Incremento BCD (++B):



Le istruzioni intermedie

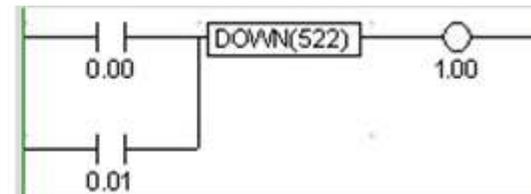
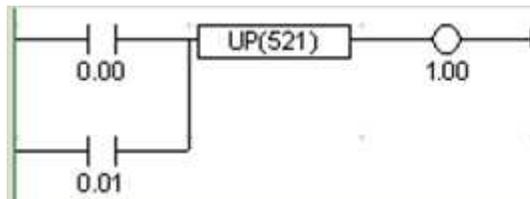
- Come già sottolineato, le nuove famiglie di PLC CJ/CS e CP1 supportano un insieme di nuove istruzioni
- Tra queste, sono state introdotte anche alcune funzioni cosiddette 'intermedie' che non si trovano al termine della Rung come tutte le altre, bensì all'interno di essa, e richiedono quindi anche un segnale di uscita
- L'utilizzo di queste istruzioni in alcuni casi semplifica notevolmente la stesura del programma

- Esempio:

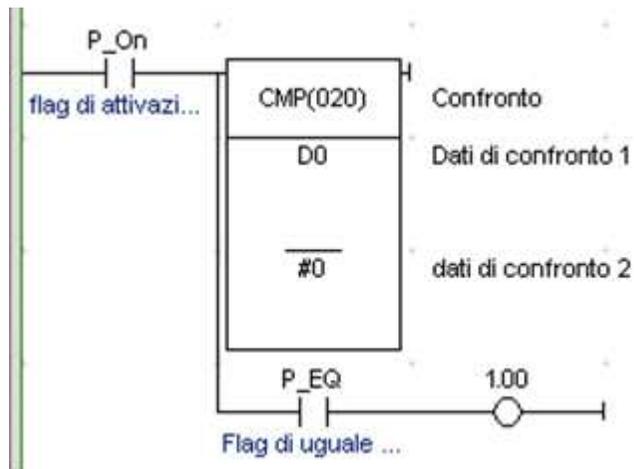


Le istruzioni intermedie

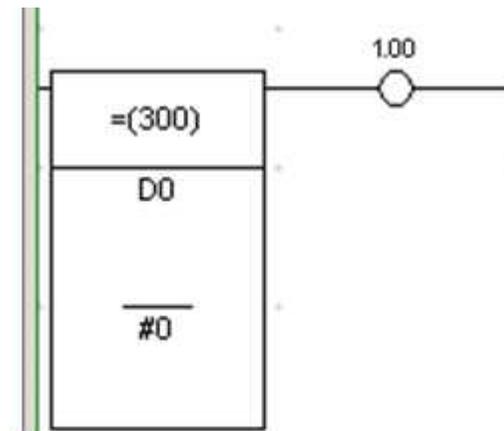
- Istruzioni di ingresso **differenziate**
 Fronte di salita (@)
 Fronte di discesa (%)
- Istruzioni di **refresh** immediato
 Refresh immediato di un ingresso o di un'uscita (!)
 Possibile combinazione con la differenziazione
- Istruzioni **UP** e **DOWN**
 Come DIFU e DIFD
 Sono istruzioni intermedie che non richiedono di un bit di appoggio



Istruzioni di comparazione intermedie



Comparazione standard



Comparazione intermedia

- Sui PLC della serie CJ/CS e CP1 è possibile utilizzare le comparazioni come istruzioni intermedie
- L'istruzione intermedia fornisce direttamente in uscita il risultato della comparazione

Istruzioni di comparazione intermedie

- Istruzioni di **comparazione**:

Sono istruzioni intermedie (=, <>, <, <=, >, >=)

- Suffisso **S** (p.es: >S): Confronto di dati binari con segno
- Suffisso **L** (p.es: >L): Confronto di dati su doppia word
- Suffisso **F** (p.es: >F): Confronto di dati in virgola mobile

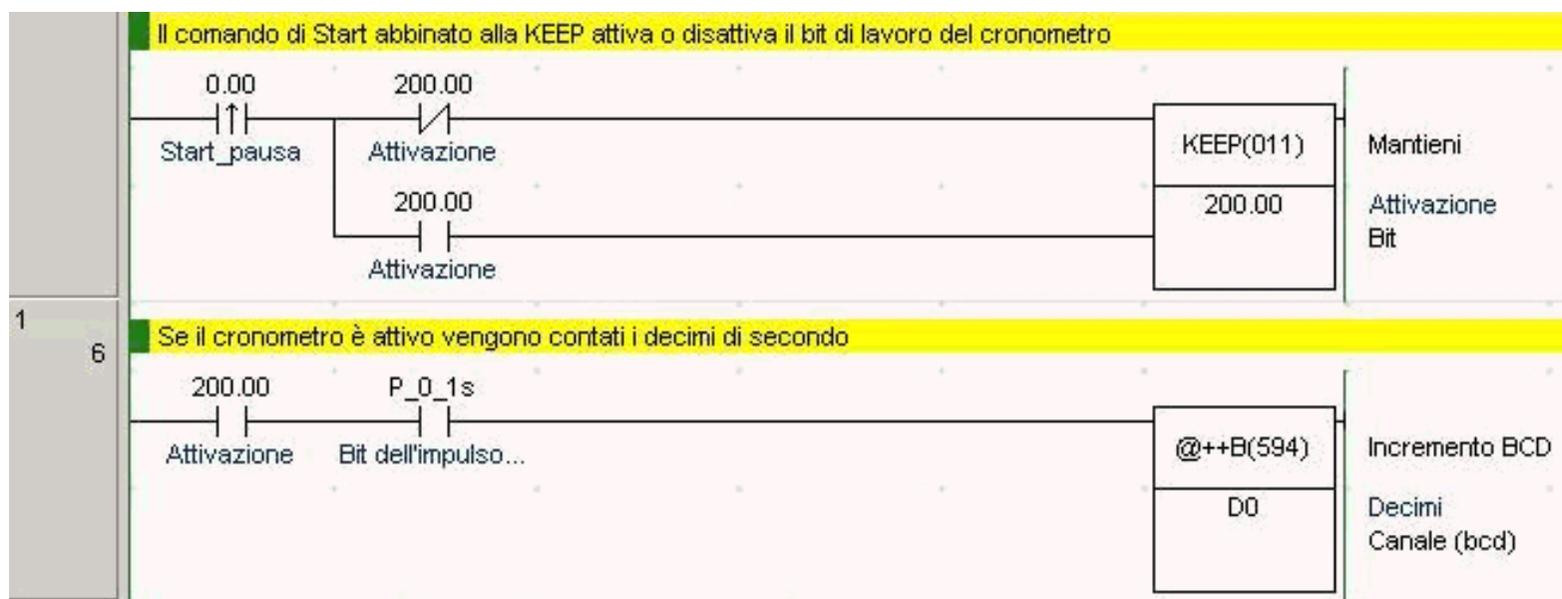
Esercizio 8: Cronometro



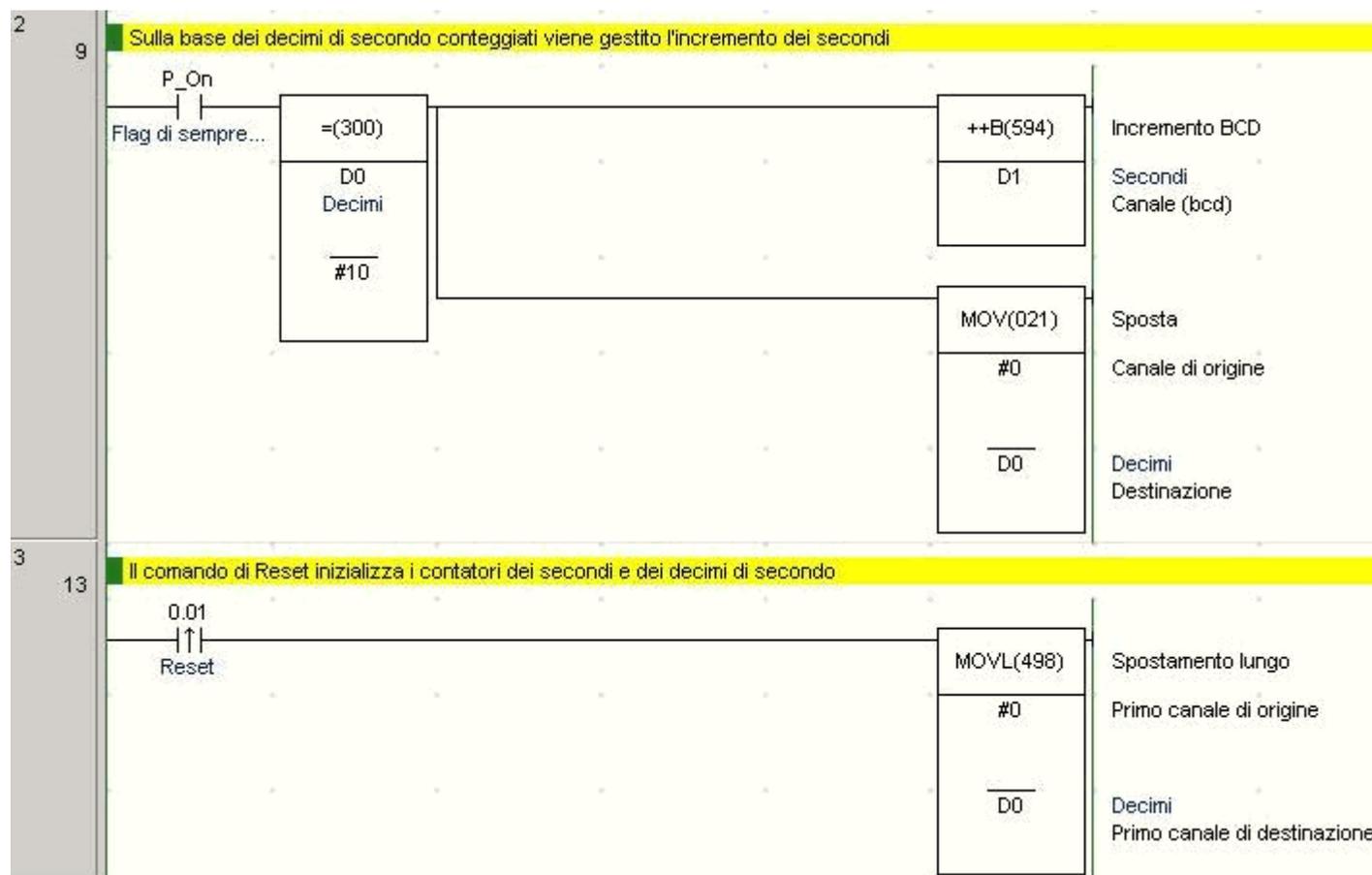
- Realizzare un cronometro che gestisca i secondi e i decimi di secondo
- Un unico pulsante dovrà comandare lo Start e lo Stop del cronometro
- Prevedere anche un pulsante di Reset

^ Attivazione	BOOL	200.00		Lavoro	cronometro attivo
▬ Decimi	CHANNEL	D0		Lavoro	decimi di secondo
^ Reset	BOOL	0.01		Lavoro	Pulsante di Reset
▬ Secondi	CHANNEL	D1		Lavoro	secondi
^ Start_pausa	BOOL	0.00		Lavoro	Pulsante di start / pausa

Esercizio 8: Soluzione



Esercizio 8: Soluzione



L'istruzione BIN

- L'istruzione BIN converte un numero decimale (BCD) di 4 cifre in un numero esadecimale (BIN) a 16 bit e trasferisce il dato convertito in un altro canale
- Se il valore convertito è 0000 allora il flag di uguale (P_EQ) viene posto ad ON
- Se il valore da convertire non è BCD, il flag di errore (P_ER) viene posto ad ON

L'istruzione BCD

- L'istruzione BCD converte un numero esadecimale (BIN) a 16 bit in un numero decimale (BCD) di 4 cifre e trasferisce il dato convertito in un altro canale
- Se il valore convertito è 0 allora si alza il flag P_EQ
- Il massimo valore convertibile è pari a 270F (HEX) che equivale al valore 9999 (BCD)
- Se il valore da convertire è maggiore di 270F il flag P_ER viene posto ad ON

Le istruzioni aritmetiche



I dati nelle istruzioni

- Ogni istruzione ha, per i propri operandi, dei Tipi di dati predefiniti
- Questo rende possibile un potente controllo sui dati in quanto, alla compilazione, il CX-Programmer effettua un confronto tra il tipo dell'operando inserito e il tipo dell'operando accettabile dall'istruzione
- Esempio: inserendo un dato definito UINT come operando di una +B(400), alla compilazione viene generato un segnale di Warning
- Sarà comunque possibile trasferire il programma sul PLC

A blue vertical bar is positioned on the left side of the slide, extending from the top to the bottom. It has a slight 3D effect with a darker blue shadow on its right side.

Istruzioni di somma e sottrazione

- Storicamente le istruzioni aritmetiche più comunemente utilizzate (ADD, SUB,.. ecc) lavoravano in formato BCD ed erano quindi soggette a tutte le limitazioni derivanti da quel tipo di codifica
- Nella nuova generazione di PLC le istruzioni aritmetiche lavorano invece di default in esadecimale

Istruzioni di somma e sottrazione

- Istruzioni di somma e sottrazione binaria:
 - + (-) Somma (sottrazione) binaria con segno
 - +L (-L) Somma (sottrazione) binaria con segno (2 word)
- Istruzioni di somma e sottrazione BCD:
 - +B (-B) Somma (sottrazione) BCD
 - +BL (-BL) Somma (sottrazione) BCD (2 word)

Istruzioni di moltiplicazione e divisione

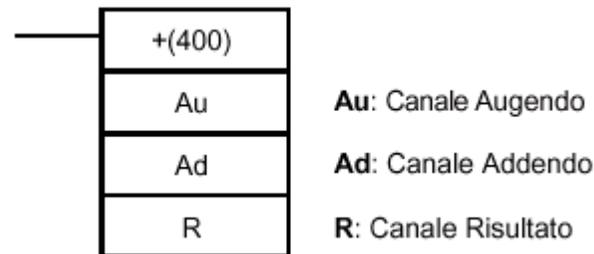
- Istruzioni di moltiplicazione e divisione binaria:
 - * (/) Moltiplicazione (divisione) binaria con segno
 - *L (/L) Moltiplicazione (divisione) binaria con segno (2 word)
 - *U (/U) Moltiplicazione (divisione) binaria senza segno
 - *UL(/UL) Moltiplicazione (divisione) binaria senza segno (2 word)
- Istruzioni di moltiplicazione e divisione BCD:
 - *B (/B) Moltiplicazione (divisione) BCD
 - *BL (/BL) Moltiplicazione (divisione) BCD (2 word)

Tabella di comparazione

Tipo di operazione	PLC serie <i>CPM1/CPM2, CQM1, C200HA</i> alpha	PLC serie <i>CJ/CS/CP1</i>
Somma (Sottrazione) BCD senza riporto	--	+B (-B)
Somma (Sottrazione) BCD con riporto	ADD (SUB)	+BC (-BC)
Somma (Sottrazione) Binaria senza riporto	--	+ (-)
Somma (Sottrazione) Binaria con riporto	ADB (SBB)	+C (-C)
Somma (Sottrazione) BCD senza riporto (2 word)	--	+BL
Somma (Sottrazione) BCD con riporto (2 word)	ADDL (SUBL)	+BCL
Somma (Sottrazione) Binaria senza riporto (2 word)	--	+L (-L)
Somma (Sottrazione) Binaria con riporto (2 word)	ADBL (SBBL)	+CL (-CL)
Moltiplicazione (Divisione) BCD	MUL (DIV)	*B (/B)
Moltiplicazione (Divisione) Binaria con segno	MBS (DBS)	* (/)
Moltiplicazione (Divisione) Binaria senza segno	MLB (DVB)	*U (/U)
Moltiplicazione (Divisione) BCD (2 word)	MULL (DIVL)	*BL (/BL)
Moltiplicazione (Divisione) Binaria con segno (2 word)	MBSL (DBSL)	*L (/L)
Moltiplicazione (Divisione) Binaria senza segno (2 word)	--	*UL (/UL)

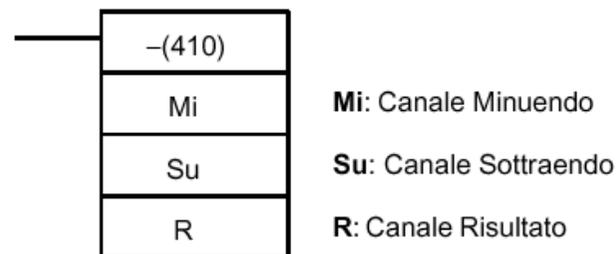
L'istruzione di somma (+)

- L'istruzione di somma + (400) esegue la somma binaria con segno tra due dati a 16 bit (canali e/o costanti)
- Gli operandi dell'istruzione sono tre:
 - **Au** e **Ad**: Addendi. Are consentite: CIO, W, H, A (da A0 ad A959), T, C, D, E, #, IR (da IR0 a IR15)
 - **R**: Risultato. Are consentite: CIO, W, H, A (da A448 ad A959), T, C, D, E, #, IR (da IR0 a IR15)



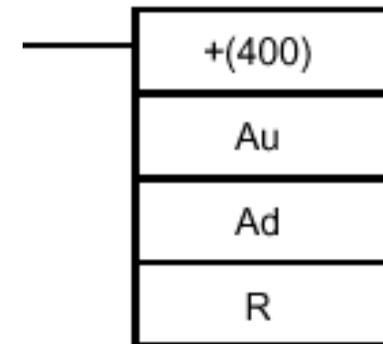
L'istruzione di sottrazione (-)

- L'istruzione di sottrazione - (410) esegue la sottrazione binaria con segno tra due dati a 16 bit (canali e/o costanti)
- Gli operandi dell'istruzione sono tre:
 - **Mi** e **Su**: Minuendo e sottraendo. Aree consentite: CIO, W, H, A (da A0 ad A959), T, C, D, E, #, IR (da IR0 a IR15)
 - **R**: Risultato. Aree consentite: CIO, W, H, A (da A448 ad A959), T, C, D, E, #, IR (da IR0 a IR15)



Uso delle costanti numeriche - esempi

SOMMA BINARIA				RISULTATO
Nome istruzione	Canale augendo	Canale addendo	Canale risultato	
+(400)	&10	&10	D0	D0 = 20 (14 hex)
+(400)	&10	#10	D1	D1 = 26 (1A hex)
+(400)	&10	+10	D2	D2 = 20 (14 hex)
+(400)	#10	#10	D3	D3 = 32 (20 hex)
+(400)	#10	+10	D4	D4 = 26 (1A hex)
+(400)	+10	+10	D5	D5 = 20 (14 hex)
+(400)	-10	+10	D6	D6 = 0 (0 hex)
+(400)	-10	-10	D7	D7 = -20 (FFEC hex)



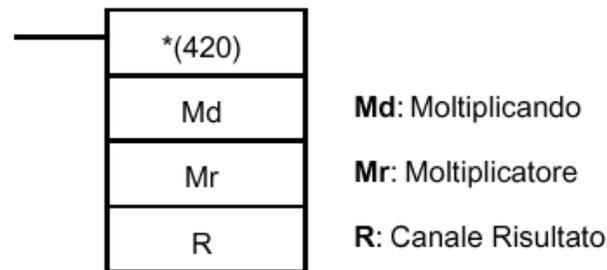
Au: Canale Augendo

Ad: Canale Addendo

R: Canale Risultato

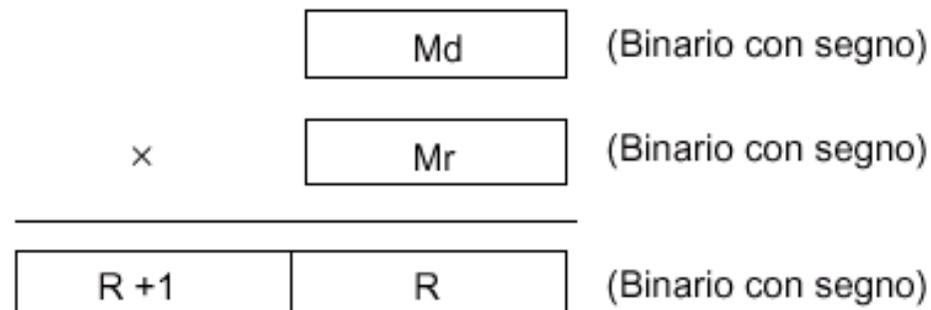
L'istruzione di moltiplicazione (*)

- L'istruzione di moltiplicazione *(420) esegue la moltiplicazione tra due dati binari con segno a 16 bit
- Gli operandi dell'istruzione sono tre:
 - **Md** e **Mr**: Moltiplicando e Moltiplicatore. Aree consentite: CIO, W, H, A (da A0 ad A959), T, C, D, E, #, IR (da IR0 a IR15)
 - **R**: Risultato. Aree consentite: CIO, W, H, A (da A448 ad A959), T, C, D, E, #, IR (da IR0 a IR15)



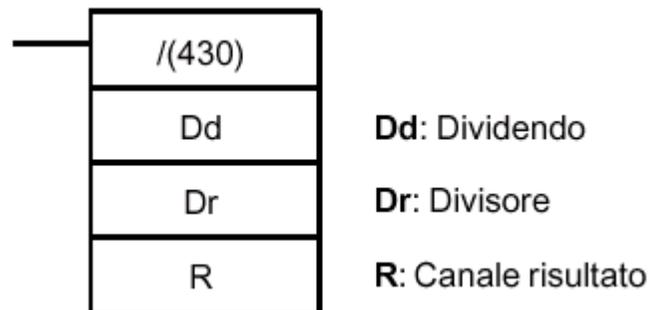
L'istruzione di moltiplicazione (*)

- Il risultato della moltiplicazione verrà posto in due canali consecutivi
- Il canale specificato nell'istruzione come terzo parametro indica il canale meno significativo in cui verrà trasferito il risultato
- Se il risultato della moltiplicazione è pari a 0 si alza il flag di uguale P_EQ



L'istruzione di divisione (/)

- L'istruzione di divisione /(420) esegue la divisione tra due dati binari con segno a 16 bit
- Gli operandi dell'istruzione sono tre:
 - **Dd** e **Dr**: Dividendo e Divisore. Aree consentite: CIO, W, H, A (da A0 ad A959), T, C, D, E, #, IR (da IR0 a IR15)
 - **R**: Risultato. Aree consentite: CIO, W, H, A (da A448 ad A959), T, C, D, E, #, IR (da IR0 a IR15)



L'istruzione di divisione (/)

- Il risultato della divisione verrà posto in due canali consecutivi e suddiviso in quoziente e resto
- Il canale successivo a quello specificato nell'istruzione come terzo parametro viene riservato per il resto
- Se il risultato della divisione è pari a 0 si alza il flag di uguale P_EQ
- La divisione per zero non viene eseguita e si alza il flag di errore P_ER

