



IN.SI.

CONTROLLORI PROGRAMMABILI

- Corso PLC
- Livello base - Introduzione

© 2010-2017 Gilberto Padovani - Tutti i diritti riservati - E' espressamente vietata qualsiasi duplicazione del presente documento.

Tutti i diritti sono riservati a norma di legge. Nessuna parte di questo documento può essere riprodotta senza l'autorizzazione dell'autore, Gilberto Padovani.

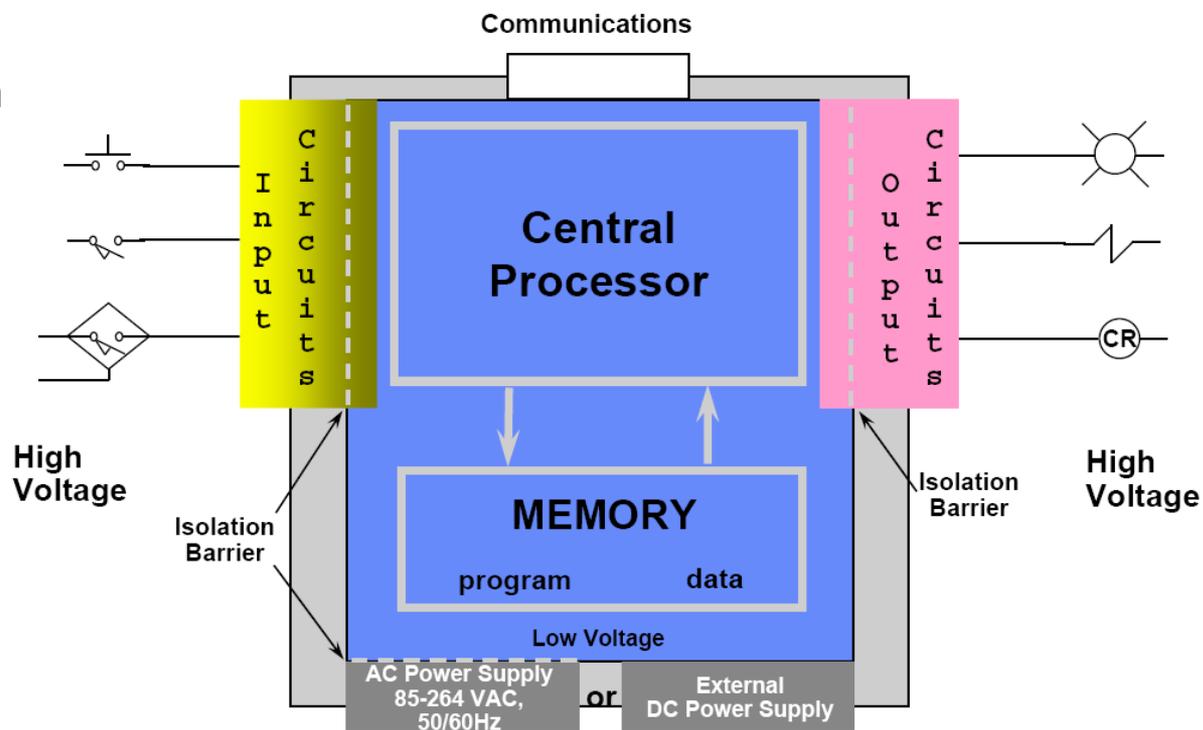
E' espressamente vietato trasmettere ad altri il seguente documento, né in formato cartaceo, né elettronico, né per denaro né a titolo gratuito.

Tutti i marchi, i brevetti registrati, i software o porzioni di essi descritti in questo documento, sono dei legittimi proprietari.



STRUTTURA DEL PLC

- Alimentatore
- CPU
- Memoria di Sistema
- Memoria Utente
- Schede di I/O
- Periferiche





Alimentatore

- Deve essere ben dimensionato, per fornire i **corretti livelli di tensione** al microprocessore e agli altri circuiti elettronici presenti all'interno della cpu.
- Per alcuni modelli all'esterno del PLC e' presente un commutatore o una morsettiera, per la selezione della tensione di rete:
 - 110 vac
 - 120 vac
 - 220 vac
 - 240 vac
 - 24 vdc



CPU

- E' la parte fondamentale del PLC, avvalendosi di un microprocessore, organizza e coordina tutte le attività di comando.
- Interagisce con le memorie (di sistema e utente) con le schede di i/o e con le periferiche.
- La potenza della cpu viene espressa attraverso il set delle istruzioni e la velocità di elaborazione.



Memoria di sistema

- Costituisce il sistema operativo del PLC, contiene il set delle istruzioni.
- La memoria di sistema viene scritta su memoria rom, al fine di evitare cancellazioni involontarie.
- L'utente non può accedere alla memoria di sistema.



Memoria utente o memoria programma

- E' la memoria dove il programmatore scrive le istruzioni che il PLC deve eseguire.
- Esistono diverse possibilità:
 - RAM (per sviluppo e collaudo)
 - EPROM (per programma definitivo)
 - EEPROM o FLASH RAM(sia per sviluppo che per versione definitiva)
- La capacità della memoria programma e' il **n. massimo di istruzioni** che essa può contenere es.: 32KB, 64KB, 128KB, 256KB, 512KB... 2MB, 4MB...



Batteria

- Mantiene il contenuto della memoria dati anche in caso di cadute di alimentazione. Viene detta "**batteria tampone**". Tipica dei plc di vecchia generazione...
- La durata tipica delle batterie tampone era di circa 5 anni (dipendente dall'uso, dall'umidità, dalla temperatura).
- Quando la batteria e' in fase di scarica viene attivato un led, posto sul frontale del PLC.
- Se la memoria che contiene il programma utente e' di tipo Ram, e la batteria si scarica del tutto, vengono persi tutti i dati. ***La batteria si sostituisce a plc acceso!***



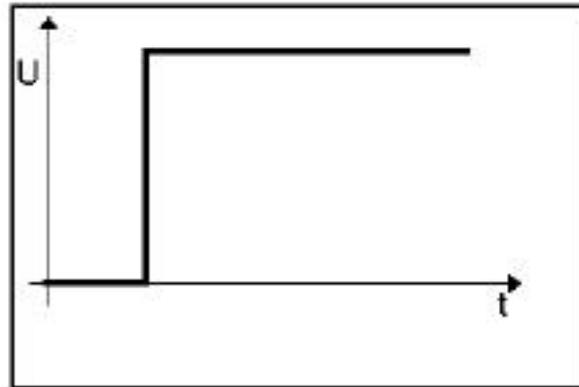
Schede di I/O

- Consentono il collegamento tra il PLC e gli i/o presenti in campo:
 - Schede di Ingresso digitali
 - Schede di Uscita digitali (a relè, transistor, triac)
 - Schede di Ingresso analogiche
 - Schede di Uscita analogiche
 - Schede di Linguaggio Basic
 - Schede di Contatori veloci, Controllo assi, Pid
 - Schede per la comunicazione seriale



Ingressi digitali

- Indicati con il valore logico 0/1
 - Livello logico 0 = non c'è passaggio di corrente elettrica.
 - Livello logico 1 = c'è passaggio di corrente elettrica.

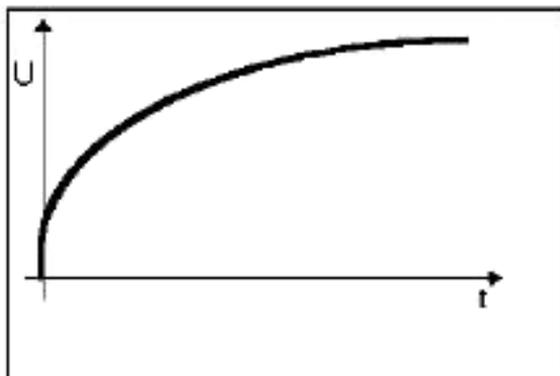


Andamento di un segnale digitale

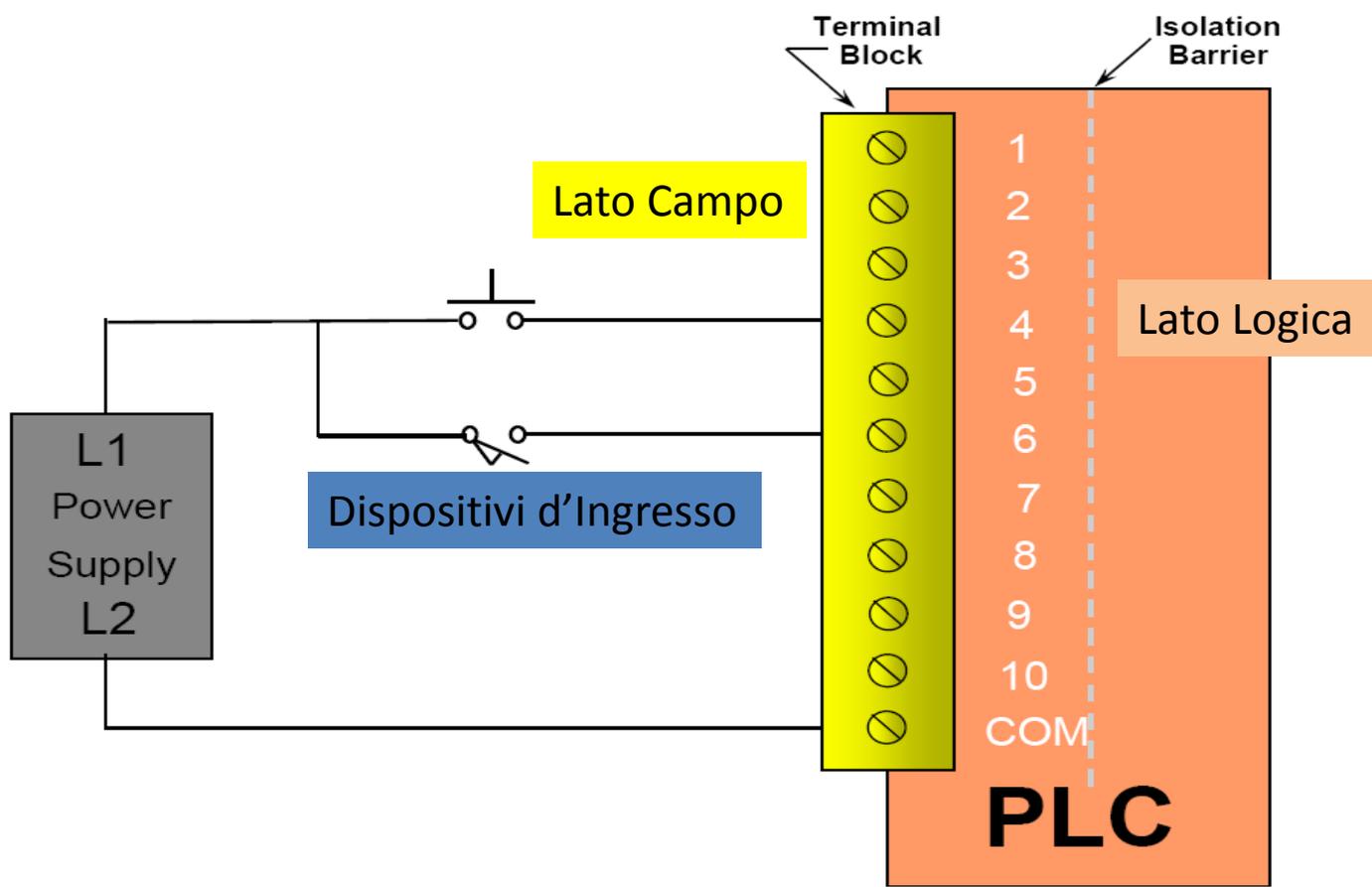


Ingressi analogici

- **In tensione:** All'ingresso viene applicata una tensione variabile da 0 a 10VDC (1-5V) secondo il valore letto da un trasduttore (pressione, temperatura, velocità, etc).
- **In corrente:** All'ingresso viene applicata una corrente variabile da 4 a 20mA (0-20) secondo il valore letto da un trasduttore (pressione, temperatura, velocità, etc).



Andamento di un segnale analogico



Tipi di ingressi

Pulsanti – Selettori – Fine corsa – Sensori – Contatti avv. Motori
Contatti relè – Misuratori di livello, temperatura, umidità etc.

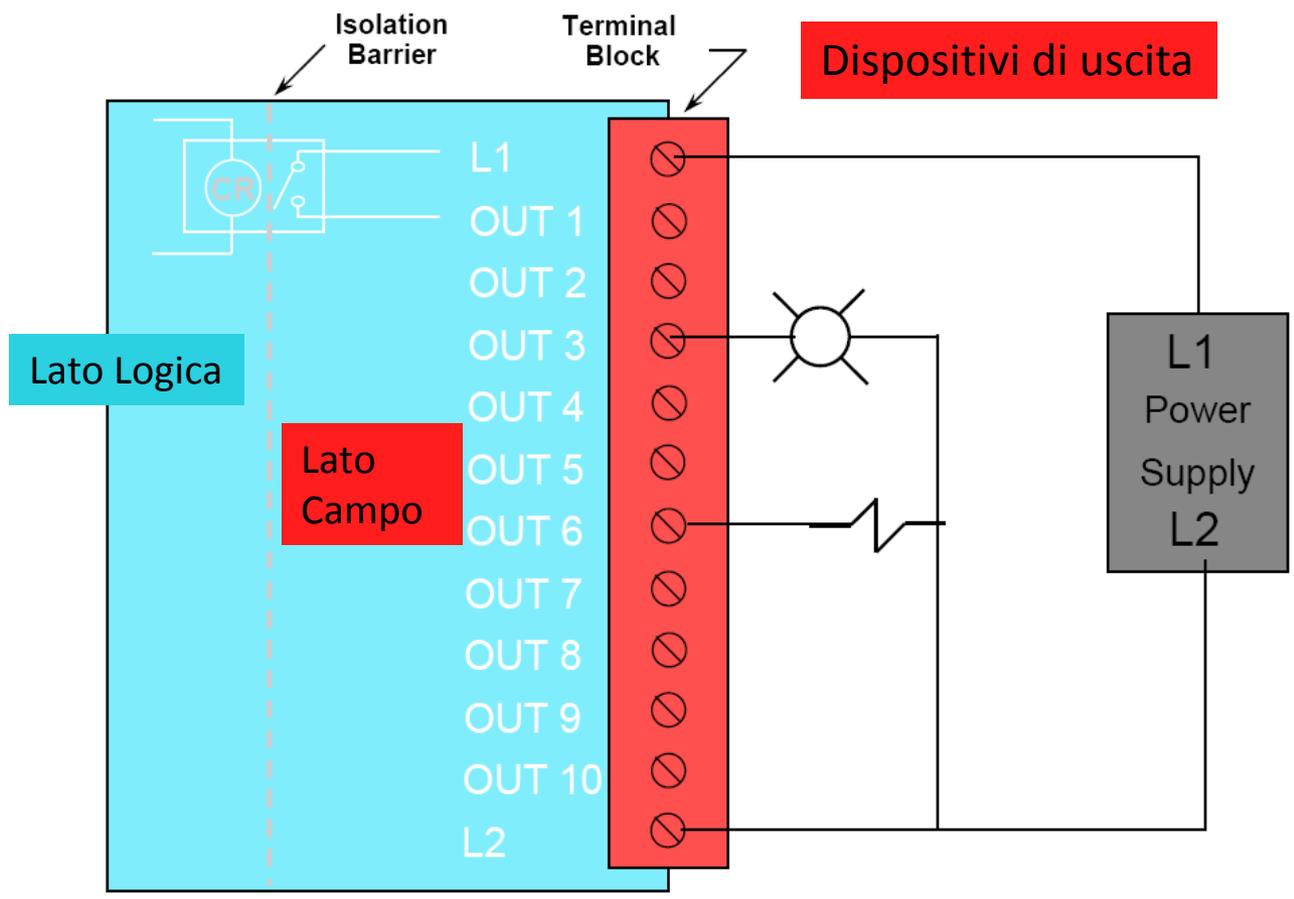


Uscite digitali

- Indicati con il valore logico 0/1
 - Livello logico 0 = non c'è passaggio di corrente elettrica.
 - Livello logico 1 = c'è passaggio di corrente elettrica.

Uscite analogiche

- **In tensione:** All'uscita viene applicata una tensione variabile da **0 a 10VDC** (1-5V) per comandare un attuatore regolabile (esempio: velocità inverter).
- **In corrente:** All'uscita viene applicata una corrente variabile da **4 a 20mA** (0-20) per comandare un attuatore regolabile (esempio: velocità inverter).



Tipi di uscite

Valvole – Elettrovalvole – Relè di controllo – Lampade – Ventilatori – Comandi Motori
Pilotaggio di attuatori tramite uscite analogiche.





Periferiche

- Consentono un ampliamento delle prestazioni del PLC ed una facilitazione del dialogo con l'operatore:
 - Consolle di programmazione
 - Display o Terminali Operatore
 - Interfaccia per personal computer
 - Interfaccia per stampante
 - Dispositivi di memorizzazione di massa

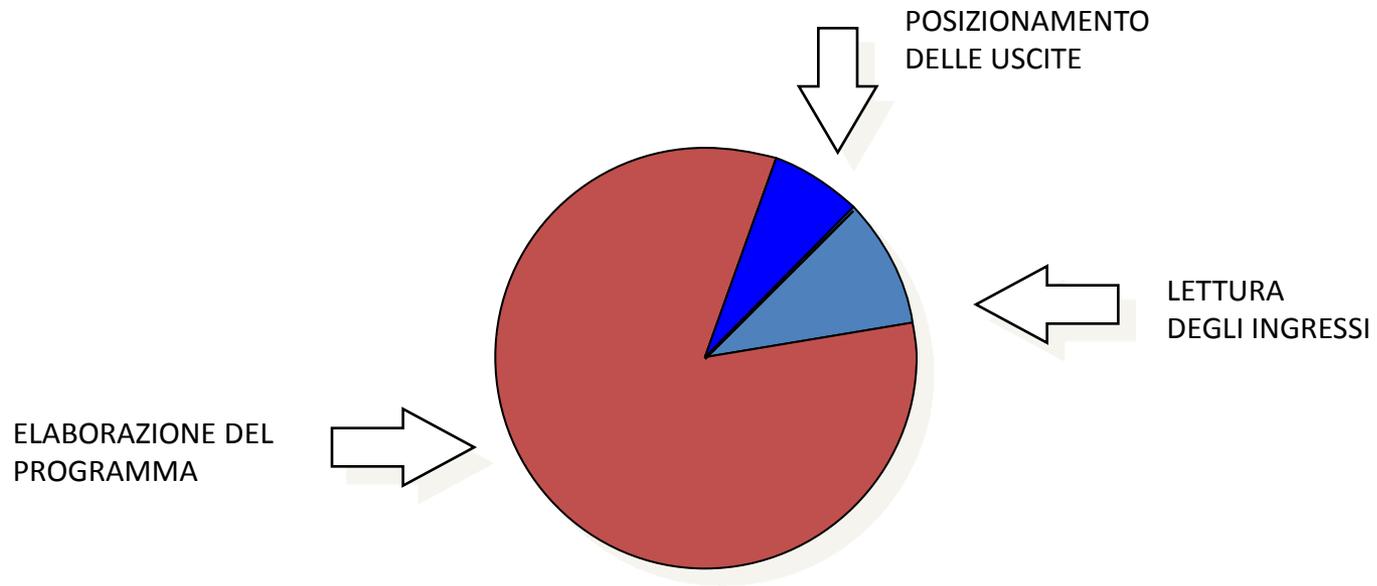


- Il **PLC** fondamentale ha due modalita' operative:
 - Esecuzione
 - Program
- **Esecuzione**
 - **RUN**: non è possibile scrivere sulle aree interne
 - **Monitor**: è possibile scrivere sulle aree interne
- **Program**
 - Il PLC non esegue la scansione del programma
 - E' possibile modificare con periferiche esterne il programma utente



CICLO DI SCANSIONE

- E' il tempo che impiega il PLC per acquisire gli ingressi, elaborare il programma, posizionare le uscite.



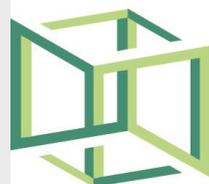


- **Micro Plc:** Fino a 32 punti di Ingressi/Uscite, esclusivamente di tipo digitali.
- **Plc di piccola taglia:** Da 32 fino a 256 Ingressi/Uscite sia digitali che analogici. Gestione reti, Moduli speciali.
- **Plc di media taglia:** Da 256 fino a 2048 Ingressi/Uscite sia digitali che analogici. Gestione reti, Moduli speciali.
- **Plc di grande taglia:** Oltre i 2048 Ingressi/Uscite. Top di gamma sia per quanto concerne la scelta dell'hardware, delle periferiche e del software.



- Allen Bradley
- Siemens
- Omron
- Telemecanique
- Ge Fanuc
- Toshiba
- Mitsubishi
- Matsushita
- Hitachi
- Klockner – Moeller
- Modicon
- ABB
-





IN.SI.

CONTROLLORI PROGRAMMABILI

- Corso PLC
- Sistemi di Numerazione

© 2010-2017 Gilberto Padovani - Tutti i diritti riservati - E' espressamente vietata qualsiasi duplicazione del presente documento.

Tutti i diritti sono riservati a norma di legge. Nessuna parte di questo documento può essere riprodotta senza l'autorizzazione dell'autore, Gilberto Padovani.

E' espressamente vietato trasmettere ad altri il seguente documento, né in formato cartaceo, né elettronico, né per denaro né a titolo gratuito.

Tutti i marchi , i brevetti registrati, i software o porzioni di essi descritti in questo documento, sono dei legittimi proprietari.



Costituisce la base del sistema di calcolo di qualsiasi elaboratore.

- Viene definita **Binaria** ogni rappresentazione dell'informazione che faccia uso di due soli simboli.

Tipicamente i due simboli adottati sono lo "zero e l'uno", oppure "vero/falso", oppure "**ON/OFF**".

E' facilmente applicabile agli schemi di collegamento a relè.

- Si utilizza per esprimere correlazioni logiche legate al funzionamento del sistema da controllare.



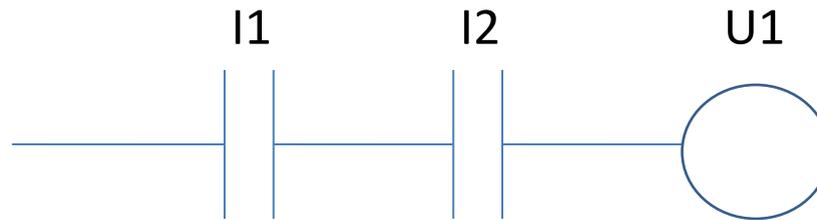


- **AND:** L'uscita è ON solo se tutti gli ingressi sono ON (circuito serie).
- OR:** L'uscita è ON quando almeno uno degli ingressi è ON. (circuito parallelo).
- NAND:** L'uscita è OFF solo se tutti gli ingressi sono ON.
- **NOR:** L'uscita è OFF solo quando almeno uno degli ingressi è ON.
- **XOR:** L'uscita è ON quando i due ingressi sono diversi tra loro.
- **NOT:** L'uscita "nega" la condizione dell'ingresso.





- **AND:** L'uscita è ON solo se tutti gli ingressi sono ON (circuito serie).

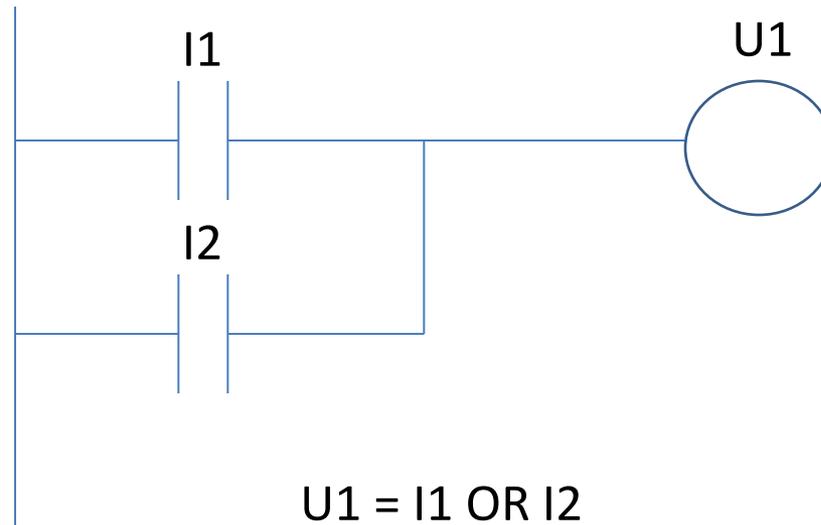


$$U1 = I1 \text{ AND } I2$$





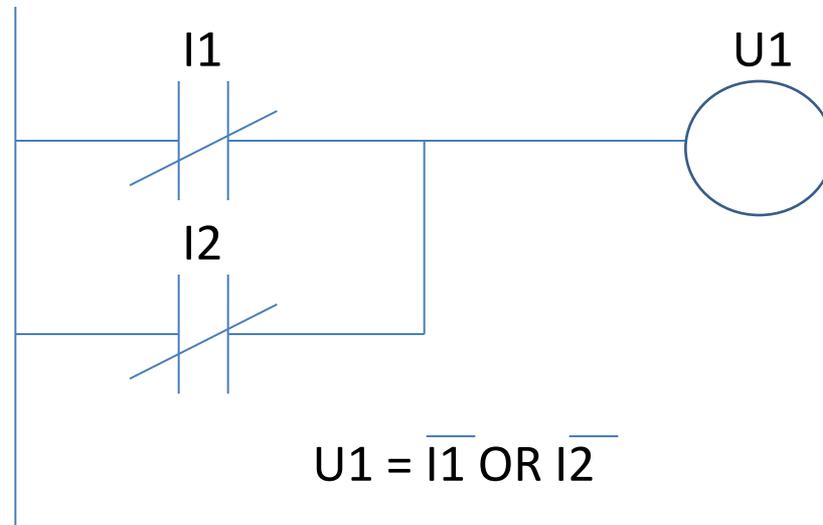
- **OR:** L'uscita è ON quando almeno uno degli ingressi è ON (circuito parallelo).





OPERATORE NAND

- **NAND:** L'uscita è OFF solo se tutti gli ingressi sono ON.





- **Sistema Decimale: Base 10 Cifre da 0÷9**

$$10^3 \quad 10^2 \quad 10^1 \quad 10^0$$

$$4 \quad 3 \quad 2 \quad 1 \quad 4321 = 4 \times 10^3 + 3 \times 10^2 + 2 \times 10^1 + 1 \times 10^0$$

$$4321 = 4 \times 1000 + 3 \times 100 + 2 \times 10 + 1 \times 1$$

Sistema Binario: Base 2 Cifre da 0÷1

$$2^3 \quad 2^2 \quad 2^1 \quad 2^0$$

$$1 \quad 1 \quad 0 \quad 1 \quad 1101 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$1101 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 \Rightarrow 13$$

Sistema Esadecimale: Base 16 Cifre da 0÷9 e da A÷F

$$16^3 \quad 16^2 \quad 16^1 \quad 16^0$$

$$5 \quad 7 \quad 2 \quad 3 \quad 5723 = 5 \times 16^3 + 7 \times 16^2 + 2 \times 16^1 + 3 \times 16^0$$

$$5723 = 5 \times 4096 + 7 \times 256 + 2 \times 16 + 3 \times 1$$





- **Sistema BCD: Cifre da 0÷9**
 - E' un tipo di codifica che viene usata nelle variabili numeriche dove bisogna effettuare operazioni di:
 - **Conteggio**
 - **Calcolo matematico**
 - **Comparazioni**

E' formata da cifre che possono assumere valori da **0** a **9**.





TABELLA DI CODIFICA

DEC	HEX	BINARIO	BCD
0000	0000	0000 0000 0000 0000	0000 0000 0000 0000
0001	0001	0000 0000 0000 0001	0000 0000 0000 0001
0002	0002	" " " 0010	" " " 0010
0003	0003	" " " 0011	" " " 0011
0004	0004	" " " 0100	" " " 0100
0005	0005	" " " 0101	" " " 0101
0006	0006	" " " 0110	" " " 0110
0007	0007	" " " 0111	" " " 0111
0008	0008	" " " 1000	" " " 1000
0009	0009	" " " 1001	" " " 1001
0010	000A	" " " 1010	" " 0001 0000
0011	000B	" " " 1011	" " 0001 0001
0012	000C	" " " 1100	" " 0001 0010
0013	000D	" " " 1101	" " 0001 0011
0014	000E	" " " 1110	" " 0001 0100
0015	000F	" " " 1111	" " 0001 0101
0016	0010	" " 0001 0000	" " 0001 0110





- **BIT:** Unità elementare di informazione, può assumere solo due stati: **ON/OFF**.

- BYTE:** Indica una sequenza composta da 8 BIT.

- WORD:** In generale indica una sequenza di Bit di lunghezza uguale a quella delle parole di memoria gestite da un calcolatore.
Indica una sequenza composta da 16 BIT.

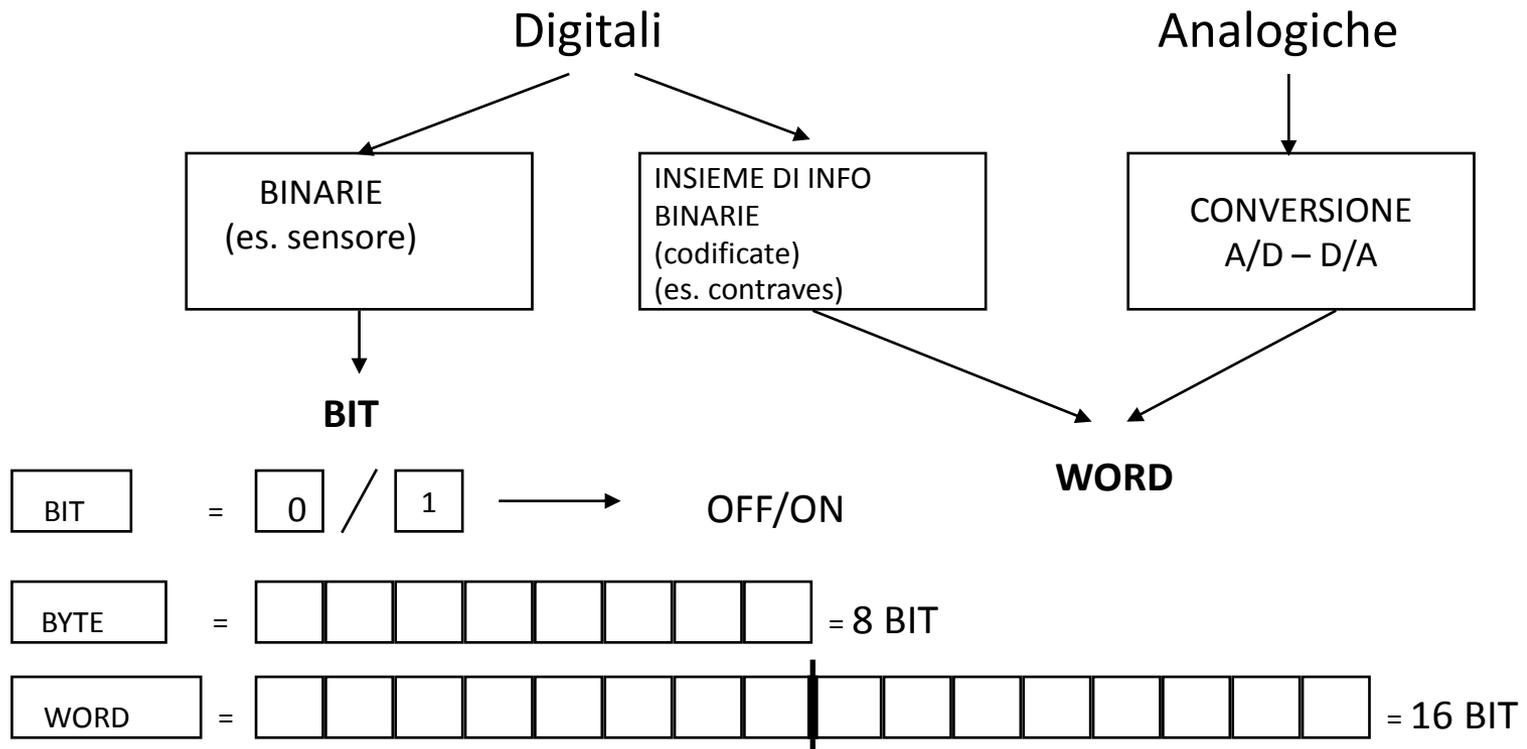
- **DWORD:** Doppia word. Composta da una sequenza di 32 BIT.





BIT... WORD CON IL PLC

- Le informazioni che il plc è in grado di gestire sono:





IN.SI.

CONTROLLORI PROGRAMMABILI

- Corso PLC
- Linguaggi di Programmazione

© 2010-2017 Gilberto Padovani - Tutti i diritti riservati - E' espressamente vietata qualsiasi duplicazione del presente documento.

Tutti i diritti sono riservati a norma di legge. Nessuna parte di questo documento può essere riprodotta senza l'autorizzazione dell'autore, Gilberto Padovani.

E' espressamente vietato trasmettere ad altri il seguente documento, né in formato cartaceo, né elettronico, né per denaro né a titolo gratuito.

Tutti i marchi, i brevetti registrati, i software o porzioni di essi descritti in questo documento, sono dei legittimi proprietari.



Quale ambiente di programmazione?

- Uno dei problemi più sentiti nell'ambito della programmazione dei plc, è l'assenza di una piattaforma comune.
- Non esiste un ambiente dove poter programmare tutti i plc presenti nel mercato. (sarebbe la fortuna dei programmatori)
- Ogni costruttore plc è legato al proprio ambiente di sviluppo, che propone a pagamento insieme all'Hardware.



Normativa 61131-3

- Un tentativo di armonizzare l'attività di programmazione plc è stata la creazione di uno standard che prende il nome di **61131-3**, creato da **PlcOpen**.
- Lo standard definisce 5 tipi di linguaggi:
 - Ladder
 - Instruction List
 - Sequenzial Function Chart
 - Function Block Diagram
 - Structured Text



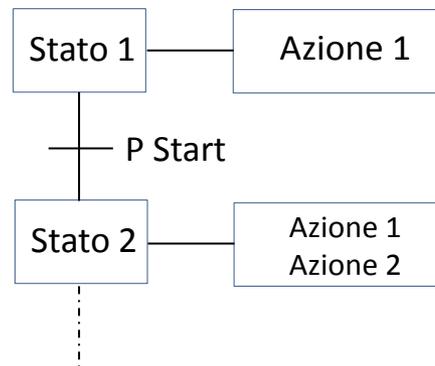
IL - Instruction List (AWL x Simatic)

- E' un linguaggio di basso livello, con una struttura molto simile a quella dell'**Assembler**.
- Rappresenta il linguaggio di base per un plc a norma 61131, non in modo implicito, ma per il fatto che quasi tutti i gli altri linguaggi vengono tradotti in **IL** prima di essere trasferiti nel Plc.
 - **U** E0.0
 - **U** E0.1
 - **=** A4.0



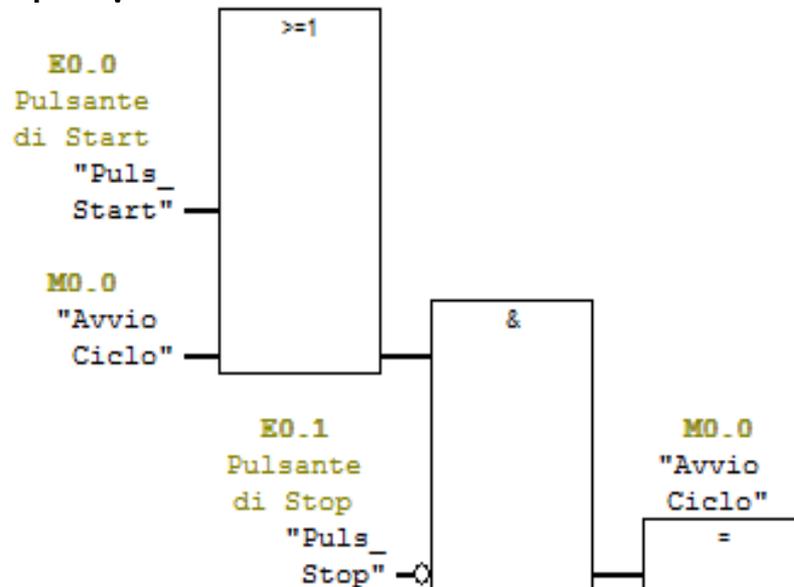
SFC - Sequenzial Function Chart (S7Graph per Simatic S7)

- Linguaggio orientato al sequencing. Permette lo sviluppo di programmi **orientati alla sequenzialità**.
- Si basa sul concetto di stati e transizioni. Dentro gli **Stati** avvengono le azioni, le **Transizioni** sono invece le condizioni che permettono il passaggio da uno stato ad un altro.
- Sia gli stati che le transizioni **sono univoci** all'interno della sequenza, fornendo una sorta di modello matematico funzionale e sicuro.



FBD - Function Block Diagram (FUP x Simatic S7)

- E' un linguaggio di alto livello di tipo grafico, viene rappresentato con la simbologia tipica dei **circuiti elettronici**.
- Per questo viene apprezzato e utilizzato soprattutto dai tecnici di estrazione elettronica.
- Non permette di sfruttare tutte le potenzialità del plc, non è adatto per la gestione di programmi con





ST - Structured Text (SCL x Simatic S7)

- Il linguaggio ST è un linguaggio di alto livello, simile a quello del Pascal, quindi viene più frequentemente utilizzato dai tecnici di **estrazione informatica**.
- E' particolarmente adatto per applicazioni dove occorre effettuare **calcoli aritmetici** complessi.

- **IF** Var1 = 1 **THEN**
 Giri = 0;
 Contr_Temp = 0
 Init = 1;
- **ELSE**
 Init = 0;
- **ENDIF**



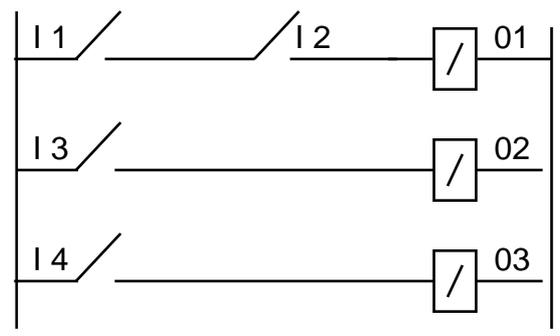
Il Ladder (FUP x Simatic S7)

- **Il Ladder** (diagramma a relè) è il più vecchio linguaggio di programmazione per i plc. Fu creato inizialmente per rendere più accessibile la programmazione ai tecnici di estrazione elettrica.
- Si basa infatti sui simboli di provenienza elettrica:
 - Binari di potenza, contatti elettrici e avvolgimenti magnetici.
- E' in assoluto il più utilizzato (e il più veloce da imparare) dei linguaggi di programmazione dedicati ai **plc**. Molto potente nelle fasi di monitoraggio e test dei software realizzati.

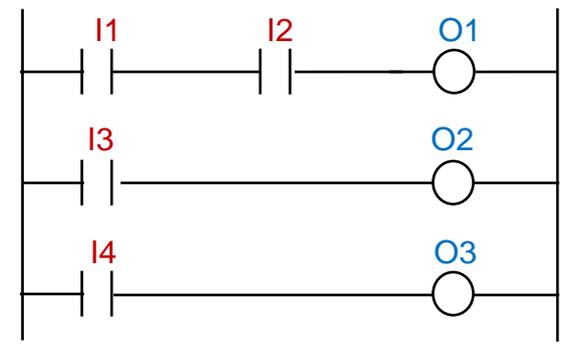


LADDER vs SCHEMA ELETTRICO

SCHEMA ELETTRICO



LADDER





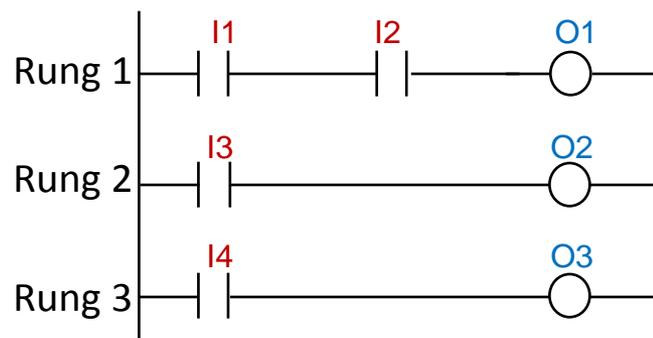
LADDER - ELEMENTI

- **Contatto NA** 
- **Contatto NC** 
- **Bobina (Coil)** 
- **Bobina Negata** 
- **Set Bobina** 
- **Reset Bobina** 



LADDER - REGOLE DI ESECUZIONE

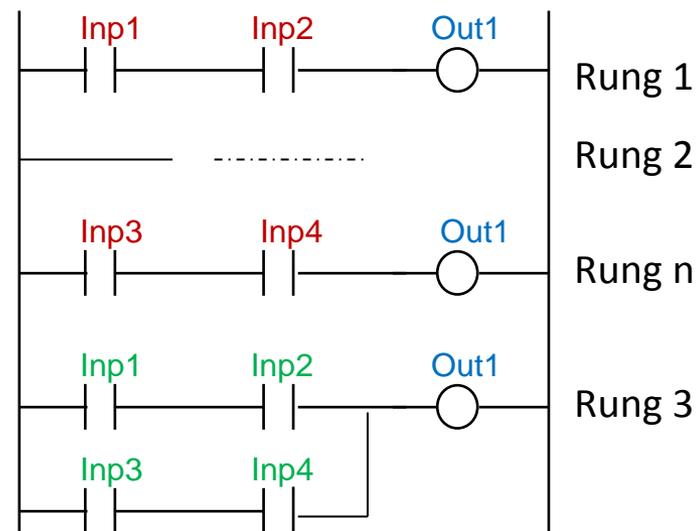
- Un programma scritto in linguaggio **ladder** viene eseguito in modo sequenziale.
- In un programma di 50 rami (rung), esso viene eseguito dal primo ramo al cinquantesimo, per poi ricominciare nuovamente dal primo (dopo aver aggiornato le uscite e aver letto gli ingressi).





Duplicazione delle Uscite/Variabili

- Uno degli errori più frequenti, durante la stesura del programma è la **duplicazione delle uscite**. (Si intendono uscite le assegnazioni che vengono fatte nella parte finale destra del circuito).
- In questo caso nel primo rung, l'effetto di **Inp1** e **Inp2** è nullo sull'uscita **Out1**. Chi influisce sulla condizione di **Out1**, sono **Inp3** e **Inp4** nel Rung2. Ecco allora sul **Rung3**, come deve essere riscritto il programma, tenendo conto di tutte le condizioni.





IN.SI.

CONTROLLORI PROGRAMMABILI

- Corso PLC Siemens S7-1200
- Panoramica Plc Simatic

© 2010-2017 Gilberto Padovani - Tutti i diritti riservati - E' espressamente vietata qualsiasi duplicazione del presente documento.

Tutti i diritti sono riservati a norma di legge. Nessuna parte di questo documento può essere riprodotta senza l'autorizzazione dell'autore, Gilberto Padovani.

E' espressamente vietato trasmettere ad altri il seguente documento, né in formato cartaceo, né elettronico, né per denaro né a titolo gratuito.

Tutti i marchi, i brevetti registrati, i software o porzioni di essi descritti in questo documento, sono dei legittimi proprietari.

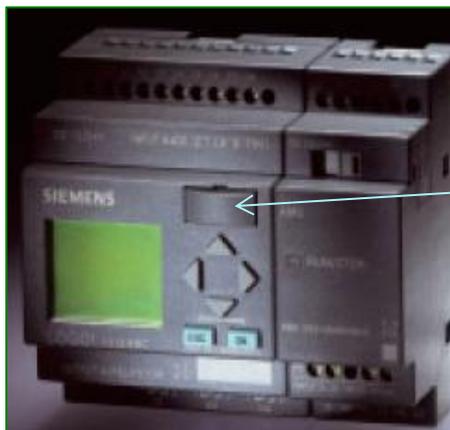


Siemens è un produttore Tedesco, è il primo per volume di vendite in Europa

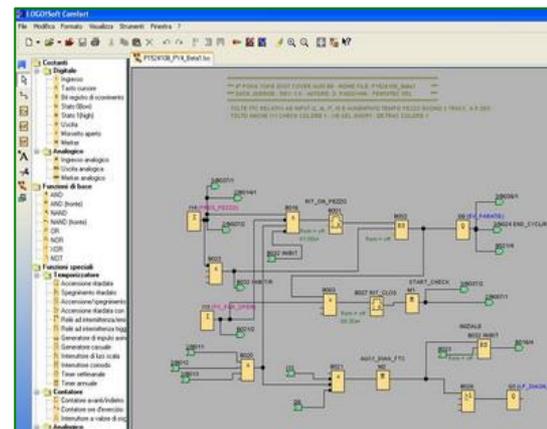
- Propone **plc** di tutte le taglie (famiglia Simatic), dai micro plc ai grandi plc, sistemi di Reti proprietarie, Terminali operatore, Sensoristica, Pulsaneria. Negli ultimi 10 anni, ha saputo rinnovare sia l'hardware che il software, fornendo componenti sempre più compatti e veloci.
- Per il software esistono cinque versioni:
 - **Logo Comfort** per plc logo
 - **MicroWin** per i plc S7-200
 - **Step7** per i plc s7-300 e 400
 - **Tia Portal** per i plc S7300-400 e S71200-1500
 - **S5** per i vecchi plc S5



LOGO



Porta di programmazione



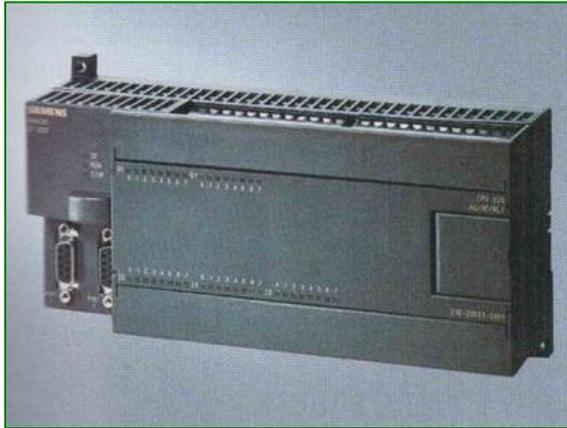
- A partire da 8 Ingressi e 4 Uscite **Logo** è il microPlc di Siemens, utilizzato per:
 - Illuminazione scale, esterna
 - Tende parasole
 - Automazione cancelli
 - Impianti di Aerazione
 - Irrigazione Giardino

- Si può programmare con i tasti a bordo Cpu oppure per programmi più complessi, si può utilizzare il software **LogoComfort**

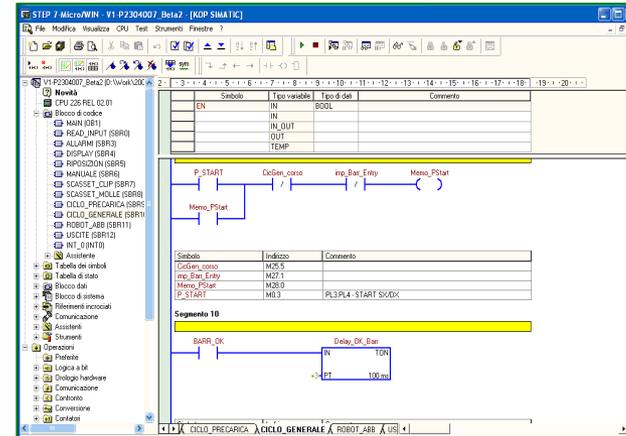


PLC DI FASCIA BASSA

S7-200



- Il Plc **compatto**, per la massima capacità di automazione e con il minimo onere economico
- Caratterizzato da facilità di montaggio, di programmazione e di utilizzo
- Con alto grado di integrazione, compatto e potente

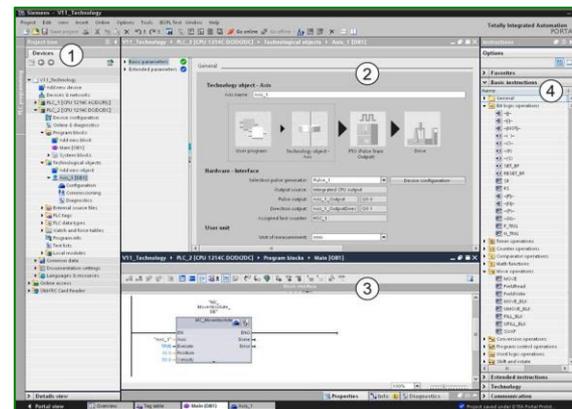


- La programmazione avviene per mezzo del software **MicroWin**
- Supporta la programmazione di compiti d'automazione difficili, con help online
- Ampio repertorio di funzioni
- Con 3 editor standard **AWL, KOP, FUP**; è possibile commutare tra gli editor in qualsiasi momento



PLC DI FASCIA BASSA / MEDIA

S7-1200

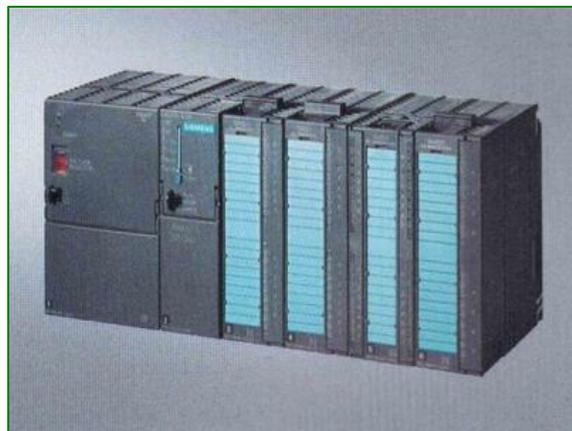


- La **nuova proposta di Siemens** per i plc di fascia bassa. Unisce la semplicità dell' S7-200 ad una potenza simile a quella del 300
- Modularità, elevata performance, facilità di utilizzo
- Nasce con il concetto di **integrazione** per compiti di Automazione semplici, ma altamente precisi
- La programmazione avviene all'interno del nuovo ambiente **Tia Portal**. Sia nella versione Basic che Professional
- Integrazione completa con i **Basic Panel**, con cui condivide lo stesso Database
- A partire dalla versione V11 sp2, contiene i linguaggi di programmazione: **KOP, FUP, SCL**

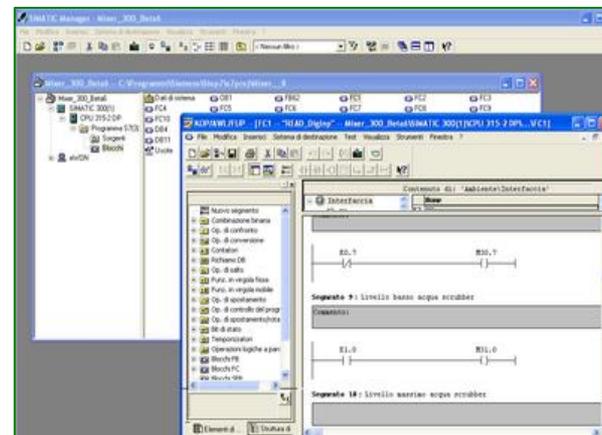


PLC DI FASCIA MEDIA

S7-300



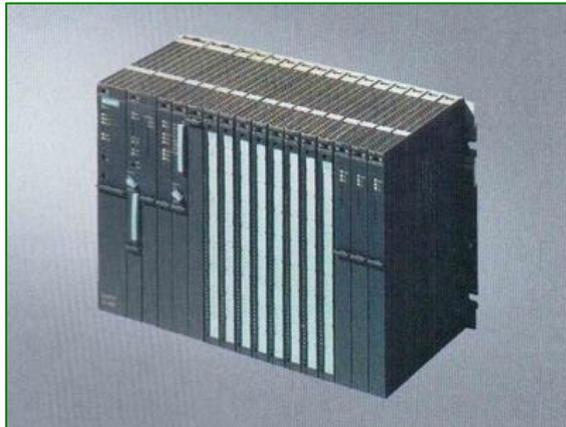
- Il Plc modulare **per tutti i compiti di automazione**
- Gamma di unità modulari completa, per l'adattamento ottimale al grado di automazione richiesta
- Ampliabile senza problemi, in funzione della crescita dei compiti
- Potente, grazie alle molteplici funzioni integrate



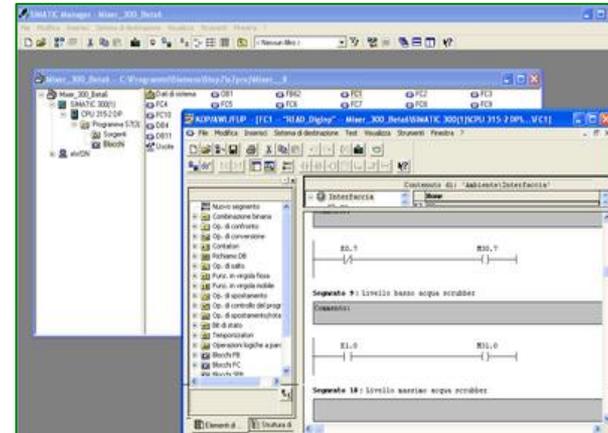
- La programmazione avviene per mezzo del software **Step7**
- Programmazione in linguaggi evoluti
- Massimo repertorio di funzioni
- Con 3 editor standard **AWL, KOP, FUP** + 2 editor speciali: **S7Graph** e **S7-SCL**



S7-400



- La soluzione ideale per compiti ed esigenze molto spinte
- Con gamma completa di unità e numerose Cpu a potenza diversificata, per l'adattamento ottimale al compito di automazione
- Impiegabile in modo flessibile con la facile realizzabilità di strutture decentrate



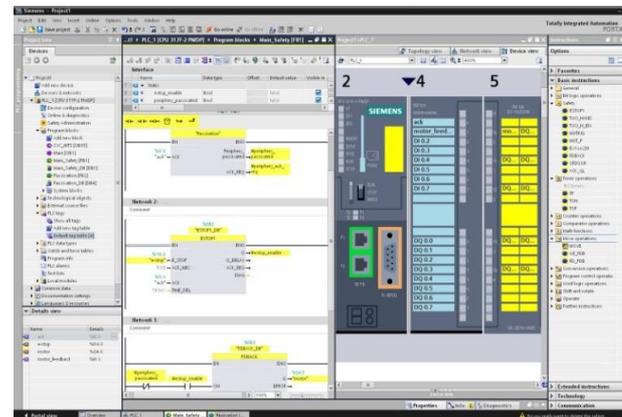
- La programmazione avviene per mezzo del software **Step7**
- Programmazione in linguaggi evoluti
- Massimo repertorio di funzioni
- Con 3 editor standard **AWL, KOP, FUP** + 2 editor speciali: **S7-Graph** e **S7-SCL**



S7-1500



- Il nuovo controllore dalla potenza davvero ragguardevole. Unisce la potenza all'efficienza.
- Dotato di **display a bordo**, per la diagnostica di primo livello
- Adatto a qualsiasi compito di automazione, anche in zone a rischio, con CPU di tipo **FailSafe**



- La programmazione avviene all'interno dell'ambiente **Tia Portal**, solo nella versione Professional
- Integrazione completa con i **Basic Panel**, con cui condivide lo stesso Database
- Permette l'integrazione anche con i controllo Assi della serie **Simotion**

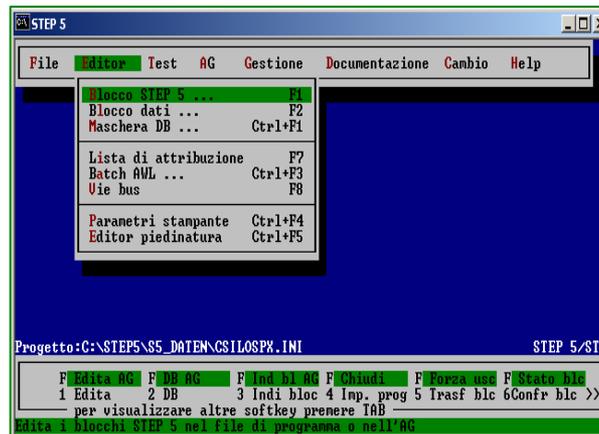


II "PLC" SIEMENS

S5-115U



- Uno dei plc più **affidabili** della storia, ormai fuori produzione, vanta il più alto numero di installazioni in Europa. Impianti in funzione da oltre 25 anni.
- L'esecuzione costruttiva **robusta** consentiva l'applicazione anche in condizioni di servizio ed ambientali molto severe.

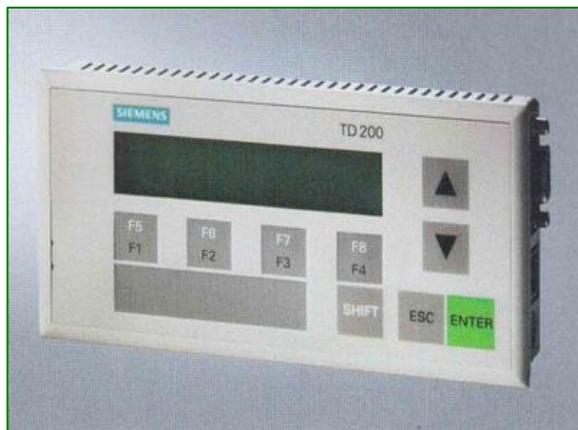


- La programmazione avveniva per mezzo del software **Step5** in ambiente DOS.
- Con 3 editor standard **AWL**, **KOP**, **FUP**.
- Di uso non facile, permetteva comunque una buona programmazione.



TERMINALI OPERATORE

TD200



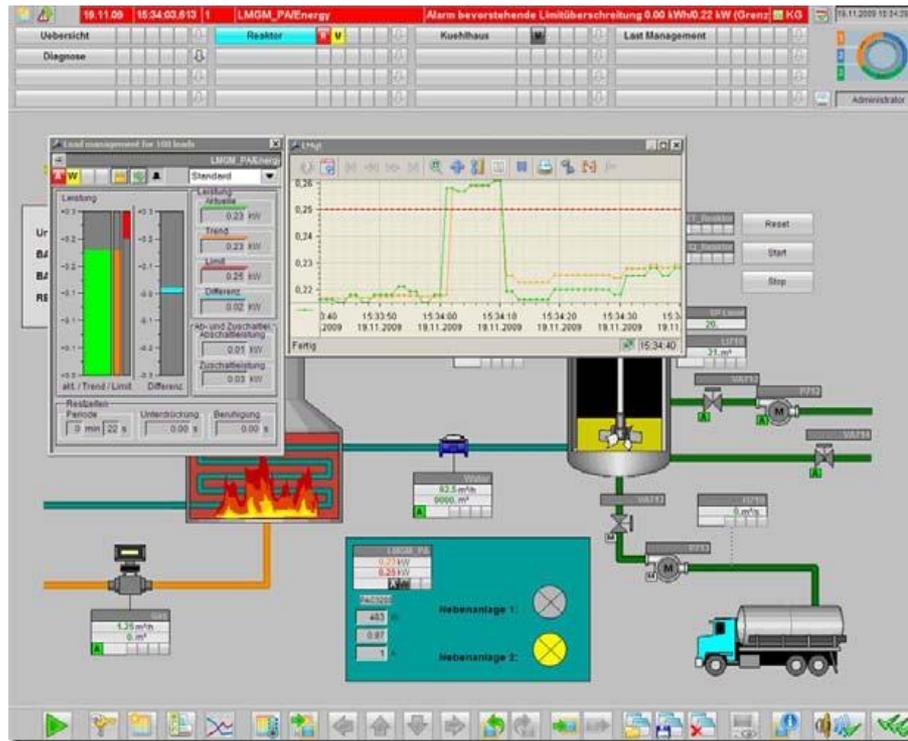
- Terminale **alfanumerico** a 2 righe. Viene utilizzato in coppia con il plc S7-200
- La programmazione avviene **all'interno** dell'ambiente MicroWin, (programma per S7-200) in modo semplice e veloce

TP177



- Terminale grafico, di tipo **Touch**. E' uno dei terminali entry level della Siemens, per il servizio e la supervisione di macchine e impianti di piccole dimensioni
- La programmazione si effettua in Tia Portal tramite l'ambiente integrato **WinCC Flexible**

Programma di Supervisione



- Con l'ambiente WinCC installato su un computer è possibile rappresentare graficamente l'impianto, gestirne i comandi, visualizzarne gli allarmi etc.



IN.SI.

CONTROLLORI PROGRAMMABILI

- Caratteristiche Plc S7-1200

© 2010-2017 Gilberto Padovani - Tutti i diritti riservati - E' espressamente vietata qualsiasi duplicazione del presente documento.

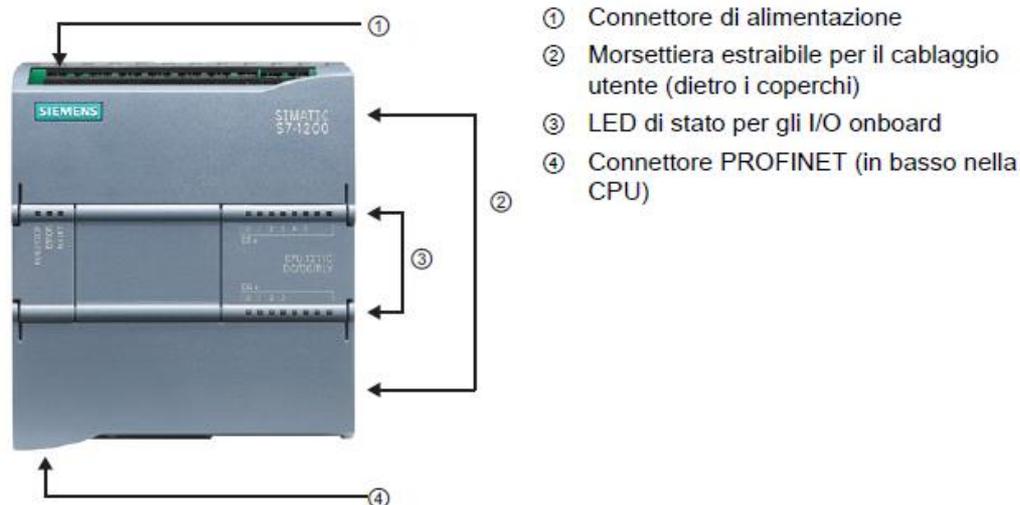
Tutti i diritti sono riservati a norma di legge. Nessuna parte di questo documento può essere riprodotta senza l'autorizzazione dell'autore, Gilberto Padovani.

E' espressamente vietato trasmettere ad altri il seguente documento, né in formato cartaceo, né elettronico, né per denaro né a titolo gratuito.

Tutti i marchi, i brevetti registrati, i software o porzioni di essi descritti in questo documento, sono dei legittimi proprietari.



Dettagli del Plc S7-1200 - la cpu entry level 1211C



- ① Connettore di alimentazione
- ② Morsettiera estraibile per il cablaggio utente (dietro i coperchi)
- ③ LED di stato per gli I/O onboard
- ④ Connettore PROFINET (in basso nella CPU)

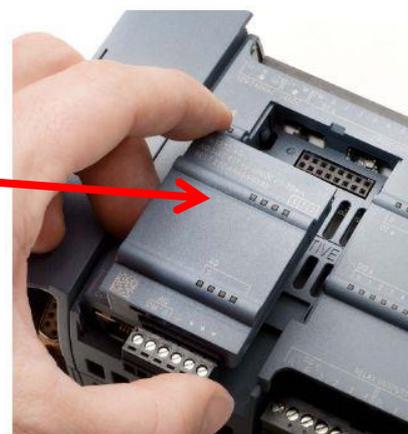
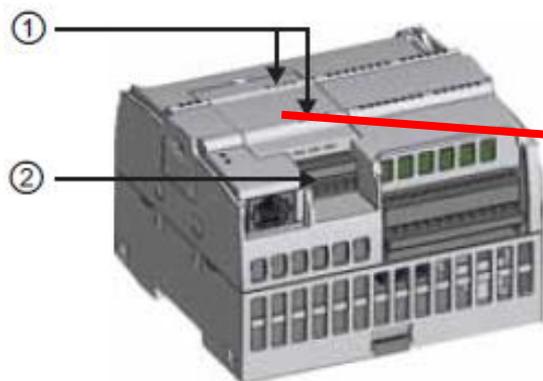
- Il **Plc S7-1200** è la nuova proposta di Siemens per i plc di fascia bassa. Lentamente ma inesorabilmente, sta togliendo quote di mercato al plc S7-200, suo predecessore!
- I modelli di **CPU** del S7-1200 attualmente sono 3:
 - Cpu **1211C**, Cpu **1212C**, Cpu **1214**



Signal board

Le signal board (SB) consentono di aggiungere ingressi e uscite alla CPU. Le SB possono disporre di I/O digitali o analogici e vengono installate sul lato anteriore della CPU.

- SB con 4 I/O digitali (2 ingressi DC e 2 uscite DC)
- SB con 1 uscita analogica

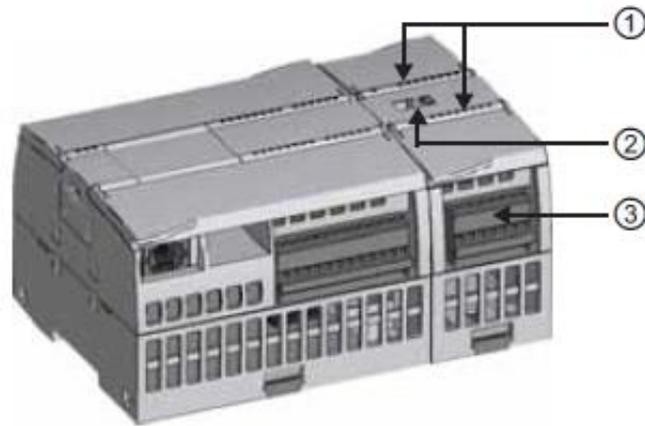


- ① LED di stato dell'SB
- ② Morsetti estraibile per il cablaggio utente



Moduli di I/O

I moduli di I/O consentono di ampliare la funzionalità della CPU e vanno collegati a destra della CPU.



- ① LED di stato degli I/O del modulo
- ② Connettore di bus
- ③ Morsetteria estraibile per il cablaggio utente





Le caratteristiche delle CPU - 1

<i>Caratteristiche CPU</i>	CPU 1211C	CPU 1212C	CPU 1214C
3 CPU	<i>DC/DC/DC, AC/DC/RLY, DC/DC/RLY</i>		
Memoria di lavoro intergrata	<i>25 KB</i>	<i>25 KB</i>	<i>50 KB</i>
Memoria di caricamento integrata	<i>1 MB</i>	<i>1 MB</i>	<i>2 MB</i>
Memoria ritentiva integrata	<i>2 KB</i>	<i>2 KB</i>	<i>2 KB</i>
Memory Card	<i>SIMATIC Memory Card (opzionale)</i>		
I/O digitali integrati	<i>6 Ingressi / 4 Uscite</i>	<i>8 Ingressi / 6 Uscite</i>	<i>14 Ingressi / 10 Uscite</i>
Ingressi analogici integrati	<i>2 Ingressi</i>		
Immagine di processo	<i>1024 Bytes per Ingressi / 1024 Bytes per Uscite</i>		
Signal Board	<i>1 max.</i>		
Moduli di segnale	<i>nessuno</i>	<i>2 max.</i>	<i>8 max.</i>
Max. I/O – Digitali	<i>14</i>	<i>82</i>	<i>284</i>
Max. I/O – Analogici	<i>3</i>	<i>19</i>	<i>67</i>



Le caratteristiche delle CPU - 2

<i>Caratteristiche CPU</i>	<i>CPU 1211C</i>	<i>CPU 1212C</i>	<i>CPU 1214C</i>
Contatori veloci	<i>3 totali</i>	<i>4 totali</i>	<i>6 totali</i>
– Fase singola	<i>3 @ 100 kHz</i>	<i>3 @ 100 kHz and 1 @ 30 kHz</i>	<i>3 @ 100 kHz and 3 @ 30 kHz</i>
– Fase in quadratura	<i>3 @ 80 kHz</i>	<i>3 @ 80 kHz and 1 @ 30 kHz</i>	<i>3 @ 80 kHz and 3 @ 30 kHz</i>
Uscite a impulsi	<i>2 @ 100 kHz (DC Uscite) / 2 @ 1 Hz (RLY Uscite)</i>		
Pulse Catch Ingressi	<i>6</i>	<i>8</i>	<i>14</i>
Interrupt ciclici temporizzati	<i>4 total with 1 ms resolution</i>		
Interrupt su fronte	<i>6 salita & 6 discesa</i>	<i>8 salita & 8 discesa</i>	<i>12 salita & 12 discesa</i>
Accuratezza orologio	<i>± 60 secondi / mese</i>		
Ritenzione orologio	<i>10 giorni tip. / 6 giorni min. a 40°C attraverso un condensatore ad alta capacità</i>		



Le caratteristiche delle CPU - 3

<i>CPU Performance</i>	CPU 1211C	CPU 1212C	CPU 1214C
Operazione booleana	<i>0.1 μs / istruzione</i>		
Movimento di word	<i>12 μs / istruzione</i>		
Matematica reale	<i>18 μs / istruzione</i>		
<i>CPU Comunicazione / Connettività</i>	CPU 1211C	CPU 1212C	CPU 1214C
Numero di porte	<i>1</i>		
Tipo	<i>Interfaccia RJ45</i>		
Velocità dei dati	<i>10/100 Mb/s</i>		
Moduli di comunicazione	<i>3 max.</i>		



Interfaccia Industrial Ethernet/PROFINET integrata



- Il SIMATIC S7-1200 dispone di un'interfaccia **Industrial Ethernet/PROFINET** integrata, che provvede alla comunicazione oltre che con lo STEP 7 Basic integrato, anche con la rete Profinet IO.
- Supporta inoltre, la comunicazione con i SIMATIC **HMI Basic Panel** per la visualizzazione, con altri controllori per la comunicazione CPU-CPU e con apparecchiature di terzi, per ulteriori possibilità di integrazione.
- L'interfaccia di comunicazione consiste in una **connessione RJ45** immune ai disturbi con funzionalità **Auto-Cross-over**, capace di supportare una molteplicità di collegamenti Ethernet, con una velocità di trasmissione dati fino a 10/100 Mbit/s.



Funzioni Integrate: Ingressi e Uscite HighSpeed



- Il 1200 dispone di serie, di funzioni integrate molto interessanti come gli **Ingressi HighSpeed** da utilizzare per funzioni di Conteggio e Misura.
- La **Cpu 1212C** dispone fino a **4 contatori veloci**, tre a 100 Khz e uno a 30Khz. Questo consente il monitoraggio preciso di trasduttori incrementali, conteggi di frequenza e veloce rilevamento di eventi di processo.
- **Le Uscite highspeed per regolazione di velocità, posizione o punto di lavoro.** Due uscite highspeed sono state integrate nella Cpu 1212C. Esse possono funzionare come uscite PTO (Pulse Train Output) o come uscite PWM (Pulse Width Modulation).



- **Area degli I/O digitali on board:**
8 ingressi (I0.0) e 6 uscite (Q0.0)
- **Area Max degli I/O**
82 punti per gli ingressi (I0.0) e uscite digitali (Q0.0)
19 punti per gli ingressi (IW64) e per le uscite analogiche (QW96)
- **Area di lavoro Merker:**
4096 Byte (32.768 bit) di Merker M (esempio M0.0)
di cui 1.796 Byte ritentivi.



- **Area di Memoria L:**

Quando viene richiamato un blocco di codice la CPU assegna la memoria **(L)** temporanea o locale, che potrà essere utilizzata durante l'esecuzione del blocco.

Al termine dell'esecuzione la CPU riassegna la memoria locale, per l'esecuzione di altri blocchi di codice.

- **Area Dati:** OB, FC, FB, DB

Blocchi **Organizzativi**, Funzioni, Blocchi **Funzionali** e Blocchi **Dati**. Il limite è la memoria Utente.



I Bit dell' area I/O vengono allocati ai moduli di Ingressi/Uscita fisici del plc

- Sono i bit o canali, a cui vengono associati i terminali esterni (**I/O Fisici**).
- Sono **indirizzabili** come Bit, come Byte, come Word o come doppia Word.
- Non sono ritentivi. In caso di mancanza di tensione vanno ad **OFF**.
- **101 punti** totali a disposizione, tra ingressi e uscite per la cpu 1212C.



Vengono utilizzati come bit di lavoro nei programmi

- Sono i bit o canali interni, che **non possono essere utilizzati** per gli I/O fisici del plc.
- Possono essere **indirizzati** come Bit, come Byte, come Word o come doppia Word
- **Non tutti sono ritentivi (solo 1.796 Byte)**. Per i restanti, in caso di mancanza di tensione, vanno ad OFF.
- **32.768 Bit** a disposizione (4.096 Byte) - esempio **M0.0**
 - La quantità di **Ritenzione**, viene impostata da software, attraverso un tool specifico.



I Temporizzatori

- Le istruzioni di temporizzazione vengono utilizzate per creare **ritardi programmati**.
- Sono indirizzabili solo come canali. La nomenclatura non è più preimpostata (T), ma liberamente definibile a livello simbolico.
- Per salvare i propri dati ciascun temporizzatore utilizza una struttura memorizzata in un **blocco dati**, che viene assegnato quando si inserisce l'istruzione nell'editor.
- **Il loro limite non è più fisso** come invece per i plc S7-200/300. Il limite dei Timer è dato sostanzialmente dalla quantità di memoria utente a disposizione!



I Contatori

- Le istruzioni di conteggio **consentono di contare sia gli eventi** interni del programma che quelli esterni del processo.
- Sono indirizzabili solo come canali. La nomenclatura non è più preimpostata (Z), ma liberamente definibile a livello simbolico.
- Per salvare i propri dati ciascun contatore utilizza una struttura memorizzata in un **blocco dati**, che viene assegnato quando si inserisce l'istruzione nell'editor.
- **Il loro limite non è più fisso** come invece per i plc S7-200/300. Il limite dei Contatori è dato sostanzialmente dalla quantità di memoria utente a disposizione!



Aree e Blocchi Dati

- **Blocchi Dati(DB)**

Non contengono istruzioni Step7 e servono alla registrazione dei **Dati Utente**. La loro dimensione è variabile e liberamente definibile.

- **Funzioni (FC)**

Le funzioni fanno parte dei **Blocchi** programmati dall'Utente, contengono istruzioni Step 7 e possono utilizzare i Blocchi Dati.

- **Blocchi Funzionali (FB)**

Anche gli **FB** fanno parte dei **Blocchi** programmati dall'Utente, con la particolarità di avere un Blocco Dati correlato come memoria (**DB di istanza**).



Blocchi Organizzativi - OB

- I blocchi organizzativi (**OB**) rappresentano l'interfaccia tra il sistema operativo e il programma utente.
- Vengono **richiamati** dal sistema operativo e comandano l'elaborazione ciclica del programma su interrupt, il comportamento di avvio del sistema di automazione e la gestione degli errori.
- Gli OB determinano la **sequenza** (eventi di avvio) in cui verranno elaborate le singole parti del programma.
- All'interno degli OB, oltre a istruzioni S7, si possono **richiamare** i blocchi funzioni (FC), i blocchi funzionali (FB) e altri OB specifici.



IN.SI.

CONTROLLORI PROGRAMMABILI

- Corso PLC Siemens S7-1200
- Indirizzamenti e Linguaggi

© 2010-2017 Gilberto Padovani - Tutti i diritti riservati - E' espressamente vietata qualsiasi duplicazione del presente documento.

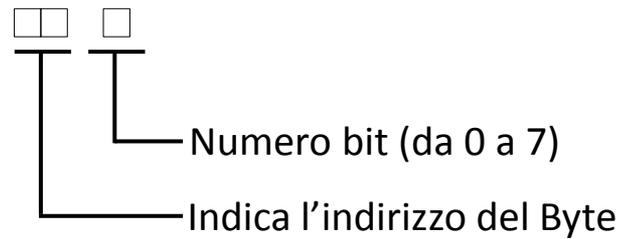
Tutti i diritti sono riservati a norma di legge. Nessuna parte di questo documento può essere riprodotta senza l'autorizzazione dell'autore, Gilberto Padovani.

E' espressamente vietato trasmettere ad altri il seguente documento, né in formato cartaceo, né elettronico, né per denaro né a titolo gratuito.

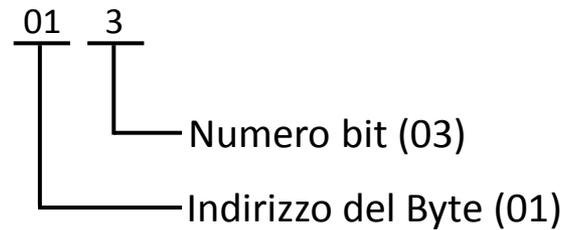
Tutti i marchi, i brevetti registrati, i software o porzioni di essi descritti in questo documento, sono dei legittimi proprietari.



Indirizzi a bit



- **Esempio** l'indirizzo del bit 3 nel Byte 01 si presenta come illustrato di seguito.



Byte	7	6	5	4	3	2	1	0
00								
01								
02								

Bit: 3



Indirizzi a word



- Nell' indirizzamento a word, così come in quello a bit, occorre inserire un **prefisso** che ne identifichi l'appartenenza.
- **I**: Ingresso Bit - **IB**: Ingresso Byte - **IW**: Ingresso Word
ID: Ingresso doppia Word
- **Q**: Uscita Bit - **QB**: Uscita Byte - **QW**: Uscita Word
QD: Uscita doppia Word
- **M**: Merker Bit - **MB**: Merker Byte - **MW**: Merker Word
MD: Merker doppia Word



Area Operandi DB

Blocco dati	Blocco dati, aperto con "AUF DB":	DB	I blocchi dati memorizzano informazioni per il programma. Essi sono definibili in modo tale che tutti i blocchi di codice possano accedervi (DB globali), oppure possono essere assegnati a un determinato FB o SFB (DB di istanza).
	Bit di dati	DBX	
	Byte di dati	DBB	
	Parola di dati	DBW	
	Doppia parola di dati	DBD	
	Blocco dati, aperto con "AUF DI":	DI	
	Bit di dati	DIX	
	Byte di dati	DIB	
	Parola di dati	DIW	
	Doppia parola di dati	DID	

IPI - Immagine di Processo degli Ingressi

- la CPU **campiona gli ingressi** (fisici) della periferia immediatamente prima dell'esecuzione dell'OB di ciclo in ogni ciclo di scansione, **e scrive i valori rilevati**, nell'immagine di processo degli ingressi.
- L'accesso all'immagine di processo degli ingressi può essere effettuato a bit, byte, parola e doppia parola. È consentito l'accesso sia in scrittura che in lettura, ma generalmente gli ingressi dell'immagine di processo vengono solo letti.
- Aggiungendo il prefisso **":P"** all'indirizzo, si fa in modo che gli ingressi digitali e analogici della CPU, **vengano letti immediatamente**.



- La differenza tra un accesso mediante $ly.x:P$, invece che $ly.x$ consiste nel fatto che **i dati provengono direttamente dall'ingresso** a cui si accede, invece che dall'immagine di processo degli ingressi.
- L'accesso $ly.x:P$ è considerato una "**lettura diretta**" perché i dati vengono prelevati direttamente dall'origine, invece che dalla copia dell'ultima immagine di processo degli ingressi, aggiornata.
- Poiché gli ingressi fisici ricevono i loro valori direttamente dall'apparecchiatura in campo a cui sono collegati, **non è consentito scrivervi**. Ciò significa che gli accessi $ly.x:P$ sono di sola lettura, diversamente dagli accessi $ly.x$ che possono essere di lettura o di scrittura.

IPU - Immagine di Processo delle Uscite

- La CPU **copia nelle uscite fisiche** i valori memorizzati nell'immagine di processo delle uscite. L'accesso all'immagine di processo delle uscite può essere effettuato a bit, byte, parola e doppia parola. È consentito l'accesso sia in lettura che in scrittura.
- Aggiungendo **":P"** all'indirizzo, si fa in modo che le uscite fisiche digitali e analogiche della CPU, **vengano scritte immediatamente**.
- La differenza tra un accesso mediante **Qy.x:P** invece di **Qy.x**, consiste nel fatto che i dati vengono scritti direttamente nell'uscita a cui si accede, oltre che nell'immagine di processo delle uscite (*la scrittura viene effettuata in entrambi i punti*).



- L'accesso $Qy.x:P$ viene definito "**scrittura diretta**" perché i dati vengono inviati direttamente all'uscita di destinazione, che non deve quindi attendere il successivo aggiornamento dell'immagine di processo delle uscite.
- Poiché le uscite fisiche comandano direttamente le apparecchiature in campo a cui sono collegate, non è consentito leggerle. Ciò significa che gli accessi $Qy.x:P$ **sono di sola scrittura**, diversamente dagli accessi $Qy.x$ che possono essere sia di lettura che di scrittura.



Tabella confronto numerazioni

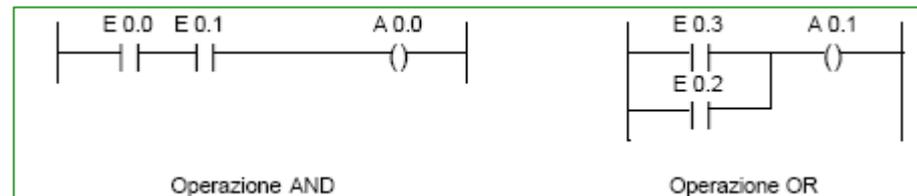
Decimale	Esadecimale	Binario	BCD	
0	0	0000		0000
1	1	0001		0001
2	2	0010		0010
3	3	0011		0011
4	4	0100		0100
5	5	0101		0101
6	6	0110		0110
7	7	0111		0111
8	8	1000		1000
9	9	1001		1001
10	A	1010	0001	0000
11	B	1011	0001	0001
12	C	1100	0001	0010
13	D	1101	0001	0011
14	E	1110	0001	0100
15	F	1111	0001	0101
16	10	10000	0001	0110





Quali linguaggi ?

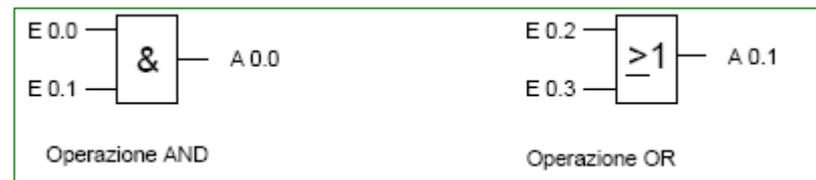
- Il Plc S7-1200 consente la programmazione con tre tipi di Linguaggi: il **KOP**, il **FUP** e l' **SCL**. **Non è stato inserito l'Awl, perché non corrispondeva alla normativa 61131-3** (poi nel 1500 è stato reinserito).
- **KOP** - Schema a Contatti (LADDER).
Di estrazione elettrotecnica, rappresenta graficamente il sistema da controllare per mezzo di simboli a norma DIN 19 239.





Linguaggio FUP

- **FUP** - Schema Logico.
Di estrazione elettronica. il sistema da controllare viene rappresentato graficamente mediante simboli a norma DIN 40700 e 19239.
- Ogni funzione è rappresentata da un simbolo che ne indica la funzione.
- A sinistra del simbolo compaiono gli ingressi e a destra le uscite.





Linguaggio SCL

- **SCL** - E' il linguaggio strutturato di Siemens (**ST per la norma 61131-3**), molto simile al Pascal, è un linguaggio di alto livello indicato per svolgere compiti di automazione complessi.
- **Adatto a tecnici di estrazione informatica**, trova applicazione nella gestione dei dati, nell'ottimizzazione dei processi, nella gestione delle ricette, nello svolgimento di compiti matematici e statistici

```

(*****)
(* RESET DEI BIT *)
(* Reset dei bit di controllo *)
(*****)

IF Sistema.Reset
  THEN
    Bing.Inizializzazione := FALSE;
    Bing.Condizioni := FALSE;
    Bing.Autoreset := FALSE;
    Bing.StopFineCiclo := FALSE;
    Bing.ModoServizio := FALSE;
    Bing.MainAttivo := FALSE;
  END_IF;

```



IN.SI.

CONTROLLORI PROGRAMMABILI

- Corso PLC Siemens S7-1200
- Set Istruzioni Base 1

© 2010-2017 Gilberto Padovani - Tutti i diritti riservati - E' espressamente vietata qualsiasi duplicazione del presente documento.

Tutti i diritti sono riservati a norma di legge. Nessuna parte di questo documento può essere riprodotta senza l'autorizzazione dell'autore, Gilberto Padovani.

E' espressamente vietato trasmettere ad altri il seguente documento, né in formato cartaceo, né elettronico, né per denaro né a titolo gratuito.

Tutti i marchi, i brevetti registrati, i software o porzioni di essi descritti in questo documento, sono dei legittimi proprietari.

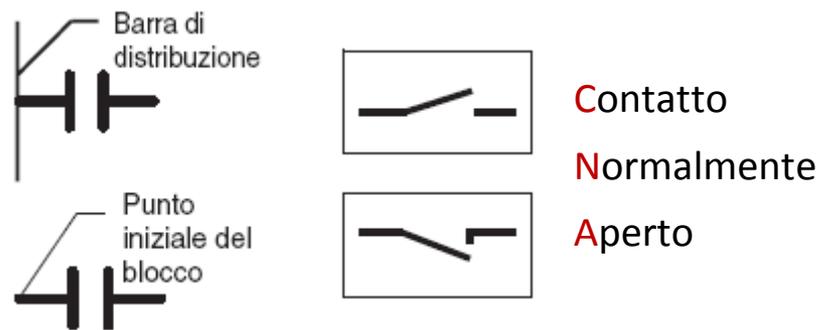


In generale

- La **CPU 1212C** dispone di oltre 400 istruzioni, con le quali è possibile realizzare comandi e controlli di qualsiasi processo.
- Le istruzioni sono suddivise in categorie, a seconda del tipo di operazione da eseguire.
- Nel corso base affronteremo le istruzioni di **“ingresso sequenza”**, **“uscita sequenza”**, **“controllo sequenza”**, **“temporizzazione e conteggio”**, **“confronto”**, **“spostamento dati”**, **“scorrimento dati”**, **“incremento e decremento”**, **“matematiche”**, **“conversione”**, **“logiche”**.



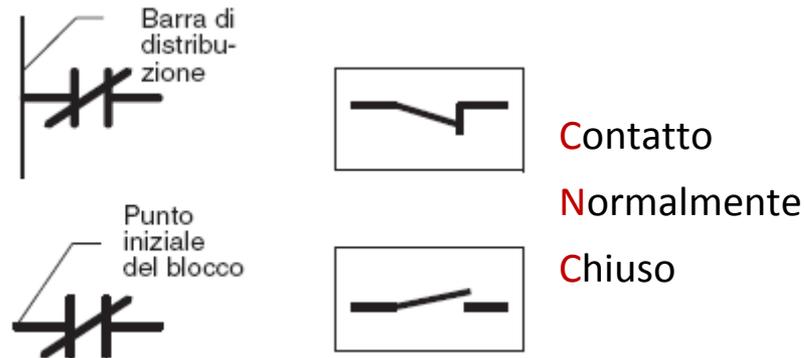
Load



- Indica un inizio logico e crea una condizione di esecuzione ON/OFF **basata sullo stato del bit** che viene assegnato all'operando.
- In **KOP (ladder)** le operazioni a contatto diretto, normalmente aperto, viene rappresentato tramite contatto grafico, come riportato nella figura in alto.



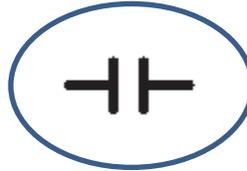
Load Not



- Indica un inizio logico e crea una condizione di esecuzione ON/OFF, **basata sull' inversione** dello stato ON/OFF, del bit che viene assegnato all' operando.
- In **KOP (ladder)** le operazioni a contatto diretto, normalmente chiuso, viene rappresentato tramite contatto grafico, come riportato nella figura in alto.

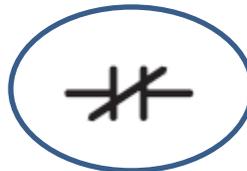


AND



- Esegue un **AND** logico dello stato del bit operando che viene specificato, e della condizione di esecuzione corrente.

AND Not



- Inverte lo stato del bit operando che viene specificato, ed esegue un **AND** logico, con la condizione di esecuzione corrente.



OR



- Esegue un **OR** logico dello stato On/Off del bit operando specificato, e della condizione di esecuzione corrente.

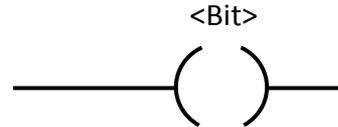
OR Not



- Inverte lo stato del bit operando ed esegue un **OR** logico, con la condizione di esecuzione corrente.

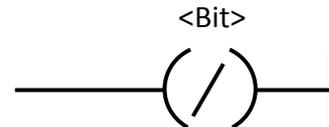


Assegnazione



- **Invia in uscita il risultato** dell'elaborazione logica, al bit specificato. (Usato sia per variabili fisiche che interne).

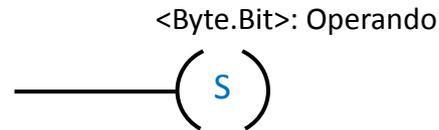
Nega Assegnazione



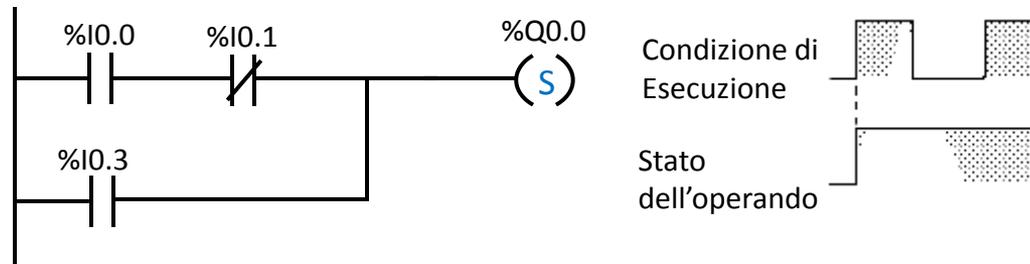
- **Inverte il risultato** dell'elaborazione logica, e lo invia in uscita al bit specificato. (Usato sia per variabili fisiche che interne).



S - Imposta Uscita

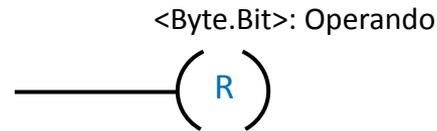


- L'istruzione **S** (*Imposta Uscita*) imposta su **ON** l'operando (definito come *Byte.Bit*), quando la condizione di esecuzione è ON.

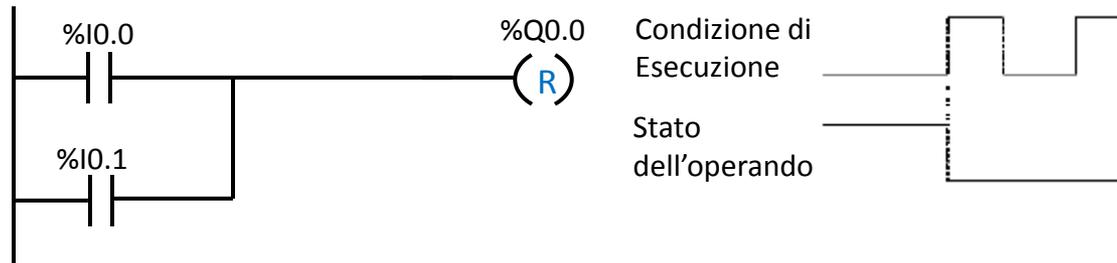




R - Resetta Uscita

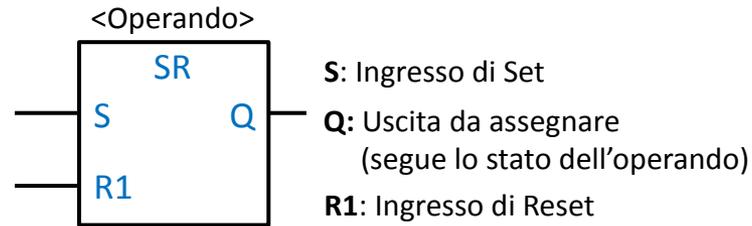


- L'istruzione **R** (*Resetta Uscita*) imposta su **OFF** l'operando (definito come *Byte.Bit*), quando la condizione di esecuzione è ON.

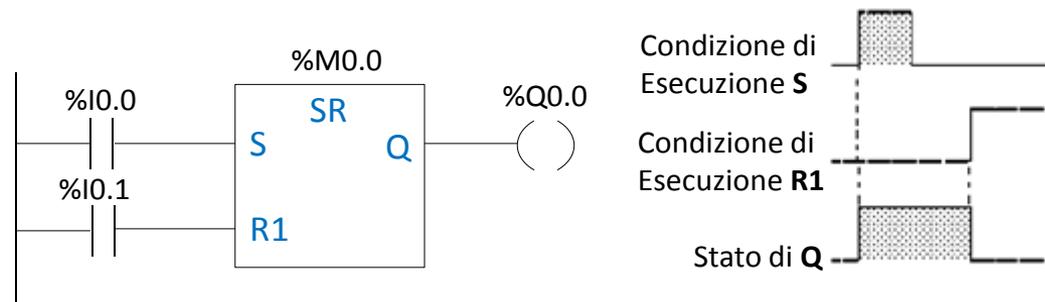




SR - Flip Flop

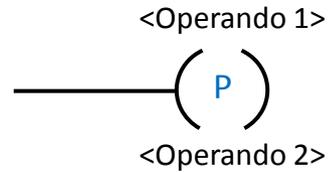


- L'istruzione **SR** (*Setta Resetta flip flop*) o blocco funzionale bistabile, simula la funzionalità di un relè a ritenuta.

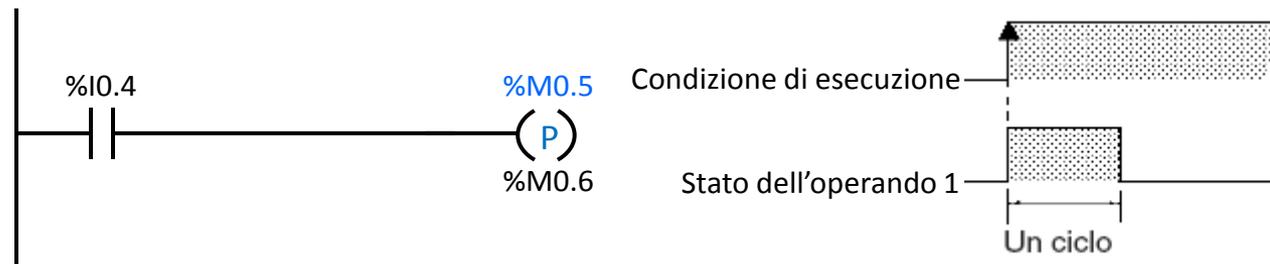




P - Positive Pulse

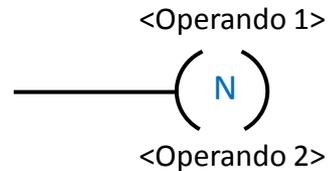


- L'istruzione **P** genera sul **fronte di salita** del segnale d'ingresso un impulso sull'*Operando 1*, avente la durata di 1 ciclo di scansione. Sull' *Operando 2* viene memorizzato lo stato del ciclo precedente.

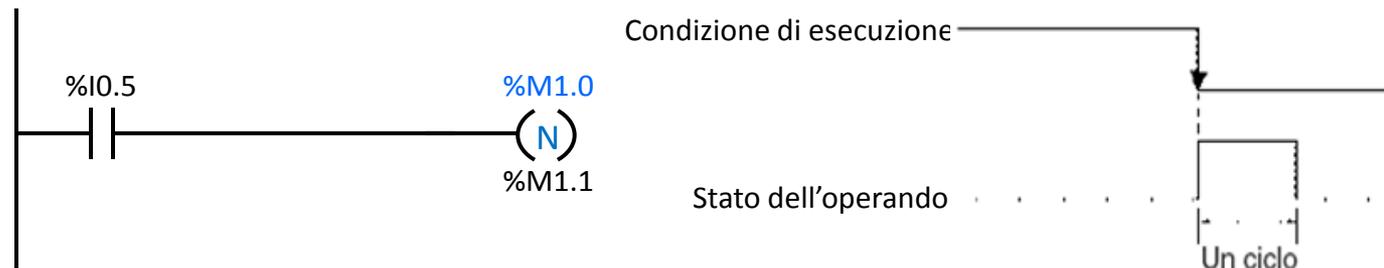




N - Negative Pulse

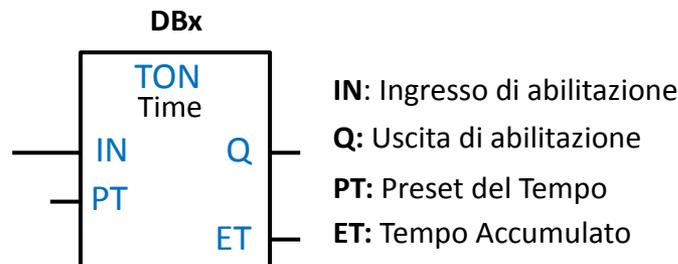


- L'istruzione **N** genera sul **fronte di discesa** del segnale d'ingresso un impulso sull'*Operando 1*, avente la durata di 1 ciclo di scansione. Sull' *Operando 2* viene memorizzato lo stato del ciclo precedente.





TON

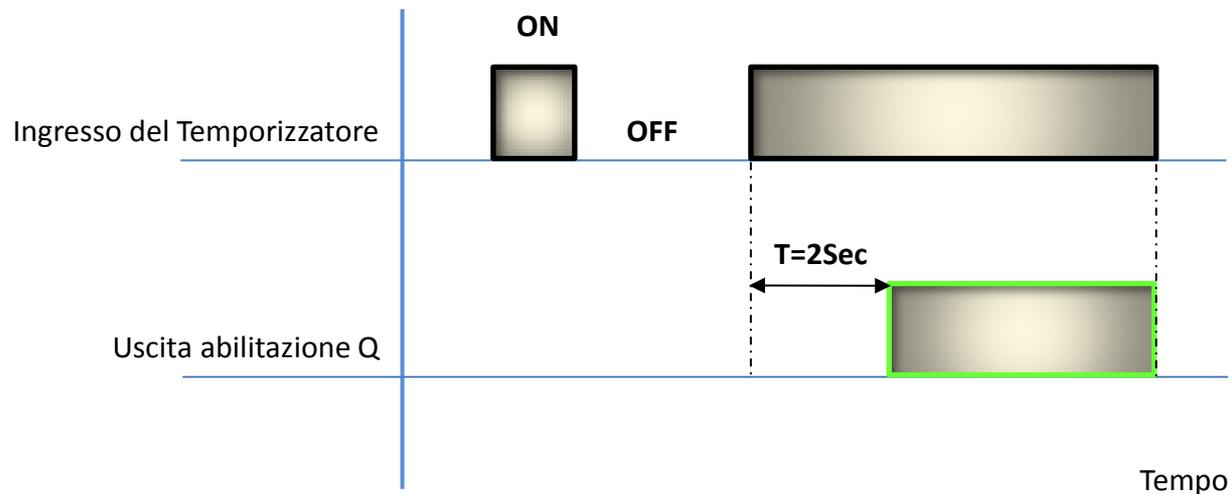


- L'istruzione **TON** (*ritardo all'inserzione*) genera un **ritardo all'eccitazione**, rispetto al segnale d' ingresso. La durata del ritardo viene impostato nel parametro **PT** "Preset".
- Il temporizzatore **IEC** è una struttura del tipo di dati di sistema che può essere dichiarata tramite un blocco dati (**DB**), oltre che come variabile locale del tipo TON, nella sezione "Input, "InOut"".
- La **base tempi** viene indicata in modo esplicito, per esempio **T#5S** significa: 5 secondi.



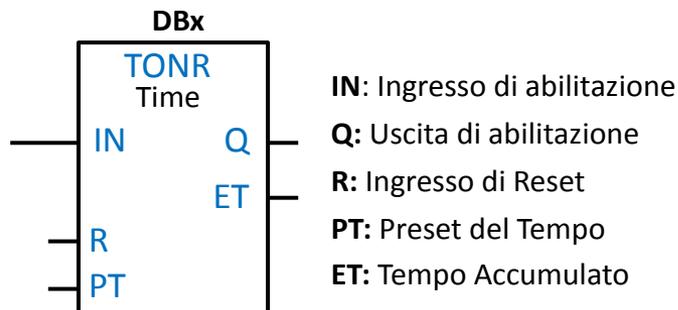
TON

- Quando l'ingresso del temporizzatore va ad ON, il valore corrente del Timer **inizia a decrementare**.
- Affinchè il Timer diventi attivo (Uscita abilitazione **Q** a ON) il segnale di ingresso *deve durare almeno quanto il valore del preset*. (esempio PT=2S significa 2 secondi)





TONR

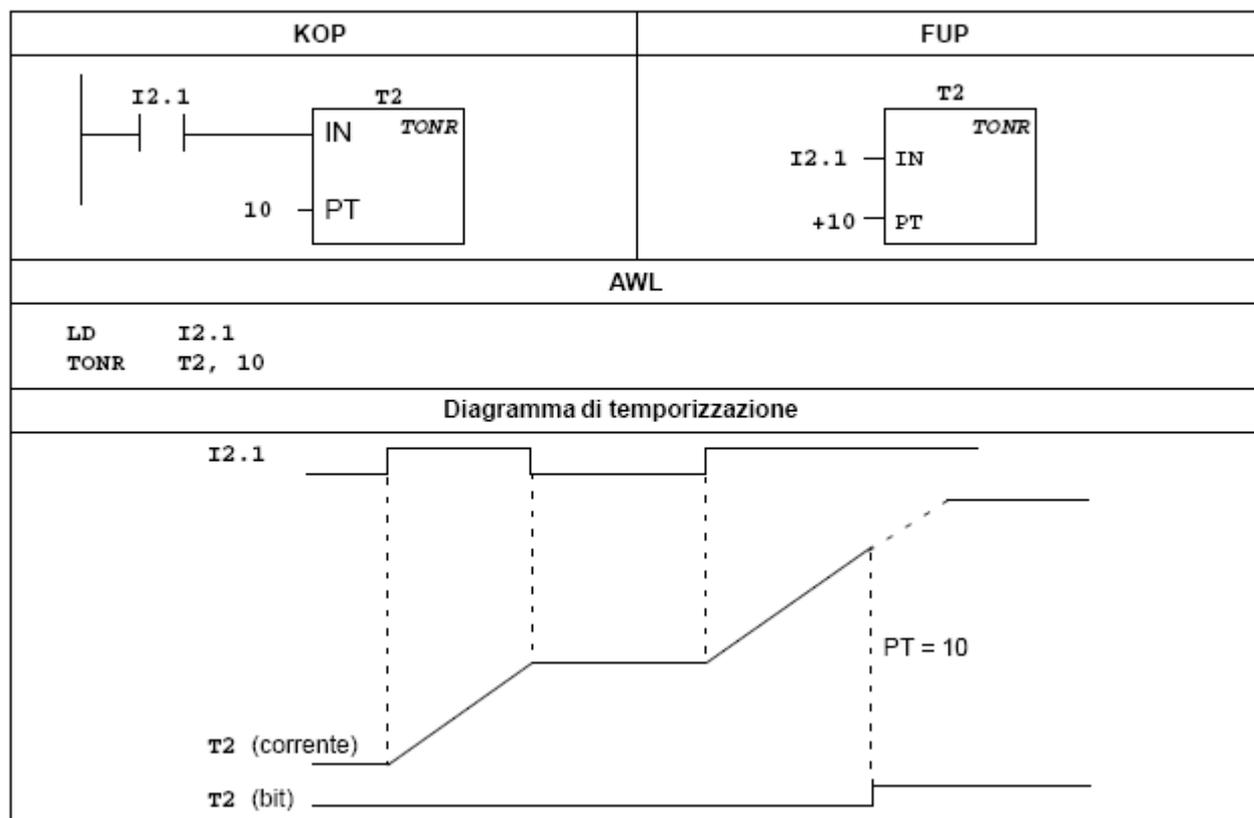


- L'istruzione **TONR** (*Ritardo all'inserzione con memoria*) genera un **ritardo all'eccitazione** rispetto al segnale d'ingresso, e memorizza il tempo trascorso. La durata del ritardo viene impostato nel parametro **PT** "Preset".
- Anche il TONR è un temporizzatore **IEC**, e ha le stesse caratteristiche del TON. Viene associato ad un blocco dati (**DB**).
- La **base tempi** viene indicata in modo esplicito, per esempio **T#5S** significa: 5 secondi.



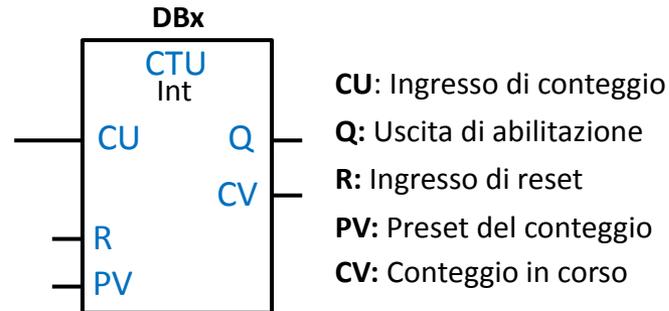
TONR

Esempio di temporizzatore di ritardo all'inserzione con memoria





CTU



- L'istruzione **CTU** (*Conta in avanti*) incrementa di "1" il valore del contatore specificato, sul fronte del segnale di ingresso, se il valore del contatore è minore di "32.767".
- **32.767** è il valore massimo impostabile nell'operando **PV**. Quando il contatore, partendo dal valore corrente arriva al valore del *Preset Value*, il conteggio termina, e l'uscita **Q** viene posta a "ON".



Esempio conteggio in Avanti

Instance name		Instance name		Name	Declaration	Data type	Description
LAD	CTU	FBD	CTU				
	Data type		Data type				
—	CU	Q	—	CU	INPUT	BOOL	Count up input
—	R	CV	—	R	INPUT	BOOL	Reset input
—	PV	Q	—	PV	INPUT	Data type *)	Defined counted value
			—	Q	OUTPUT	BOOL	Counter status
			—	CV	OUTPUT	Data type *)	Actual count value

*) USINT, UINT, UDINT, SINT, INT, DINT

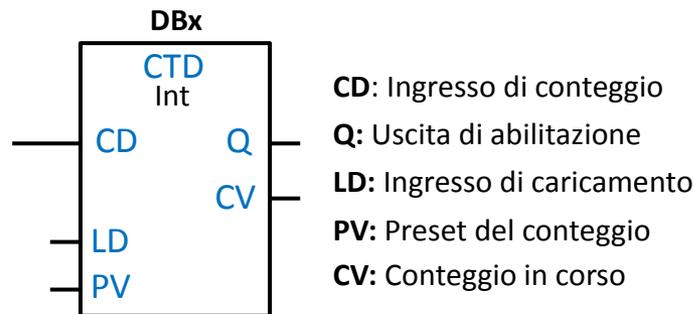
Function

The timing diagram illustrates the following sequence of events:

- CU (Count up input):** A series of 10 pulses.
- R (Reset input):** A single pulse occurring when the counter value is 6.
- PV (Defined count value):** A staircase signal that increases from 0 to 6 and then decreases back to 0.
- CV (Actual counted value):** A staircase signal that increases from 0 to 6 as long as CU is active. When R is active, CV resets to 0. When PV is 6, CV reaches 6. When PV decreases, CV follows it down to 0.
- Q (Counter status):** A pulse that occurs when CV reaches PV (at value 6).



CTD



- L'istruzione **CTD** (*Conta indietro*) decrementa di "1" il valore del contatore specificato, sul fronte di salita del segnale di ingresso, se il valore del contatore è maggiore di "0".
- **32.767** è il valore massimo impostabile nell'operando **PV**. Quando il contatore, partendo dal valore corrente arriva a "0", il conteggio termina, e l'uscita **Q** viene posta a "ON".



Esempio conteggio all'Indietro

Instance name		Instance name		Name	Declaration	Data type	Description
LAD	CTD	FBD	CTD	CD	INPUT	BOOL	Count down input
	Data type		Data type	LOAD	INPUT	BOOL	Load input
— CD	Q	— CD		PV	INPUT	Data type *)	Defined count value
— LOAD	CV	— LOAD	CV	Q	OUTPUT	BOOL	Counter status
— PV		— PV	Q	CV	OUTPUT	Data type *)	Actual count value

*) USINT, UINT, UDINT, SINT, INT, DINT

Function

Down counter input CD

Load input LOAD

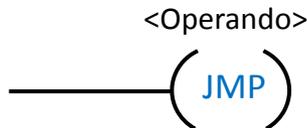
Defined count value PV

Actual counted value CV

Counter status Q



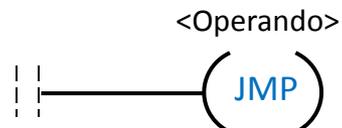
JMP(salto assoluto)



- Salto in un altro segmento se **RLO = 1** (assoluto).
“RLO” = Risultato Logico Combinatorio” ---> Condizione di esecuzione!
- **Esegue un salto** se nessun ulteriore elemento è interposto tra il percorso di corrente a sinistra e l'operazione.
- Per ogni - - - (JMP) deve essere presente anche una destinazione (**etichetta di salto**). Le operazioni tra l'operazione di salto e l'etichetta non vengono eseguite.



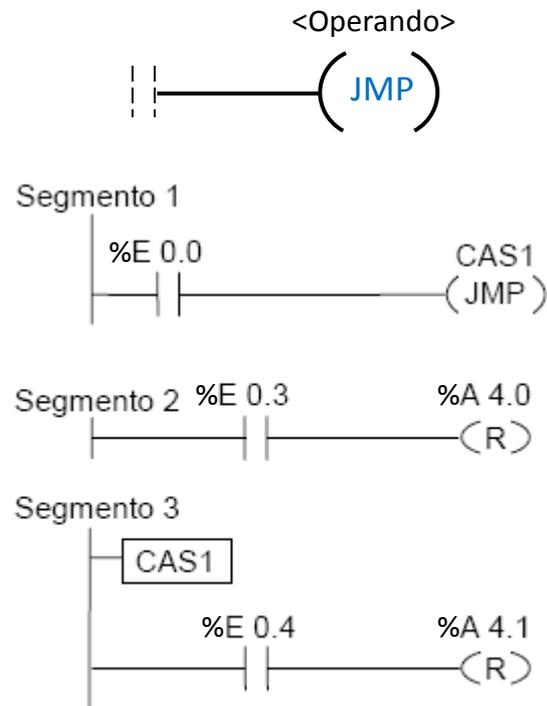
JMP(salto condizionato)



- Salto in un altro segmento se **RLO = 1** (condizionato).
“RLO” = Risultato Logico Combinatorio” ---> Condizione di esecuzione!
- **Esegue un salto** se gli elementi interposti tra il percorso di corrente a sinistra e l'operazione, da come risultato **RLO = 1**.
- Per ogni - - - (**JMP**) deve essere presente anche una destinazione (**etichetta di salto**). Le operazioni tra l'operazione di salto e l'etichetta non vengono eseguite.



JMP(salto condizionato)



- Se E 0.0 = 1, viene eseguito il salto all'etichetta **CAS1**. A causa del salto, l'operazione per resettare l'uscita A 4.0 **non viene eseguita**, anche se E 0.3 = 1.



LABEL

LABEL

- L'istruzione **LABEL** Identifica la **destinazione** di un'operazione di salto.
- L'etichetta di salto puo essere composta da **lettere dell'alfabeto, da numeri, o dal carattere underscore “_”**. La prima posizione deve essere identificata o con una lettera o dal carattere underscore.
- Quindi per ogni ---(**JMP**) o ---(**JMPN**) deve essere presente anche una etichetta di salto (**LABEL**).



Richiamo Blocco



- Non esiste un'istruzione per il richiamo dei Blocchi di programma FC, FB, OB. Per utilizzarli all'interno del segmento voluto, **è sufficiente trascinarli utilizzando il mouse.**
- Il **richiamo** al blocco può essere sia *condizionato* che *incondizionato*.
- Terminata l'esecuzione del blocco richiamato, il programma riprende l'elaborazione, dall'**istruzione subito successiva** al richiamo.



IN.SI.

CONTROLLORI PROGRAMMABILI

- Corso PLC Siemens S7-1200
- Merker di Clock e di Sistema

© 2010-2017 Gilberto Padovani - Tutti i diritti riservati - E' espressamente vietata qualsiasi duplicazione del presente documento.

Tutti i diritti sono riservati a norma di legge. Nessuna parte di questo documento può essere riprodotta senza l'autorizzazione dell'autore, Gilberto Padovani.

E' espressamente vietato trasmettere ad altri il seguente documento, né in formato cartaceo, né elettronico, né per denaro né a titolo gratuito.

Tutti i marchi, i brevetti registrati, i software o porzioni di essi descritti in questo documento, sono dei legittimi proprietari.



I Merker di Sistema

- Il **merker Byte di sistema** contiene dei merker con valori definiti. Consente alcune funzioni specialistiche e possono essere utilizzati nel programma utente.
- Il merker byte di sistema **non è fisso**. Puo' essere liberamente scelto durante la parametrizzazione dei dati della CPU.

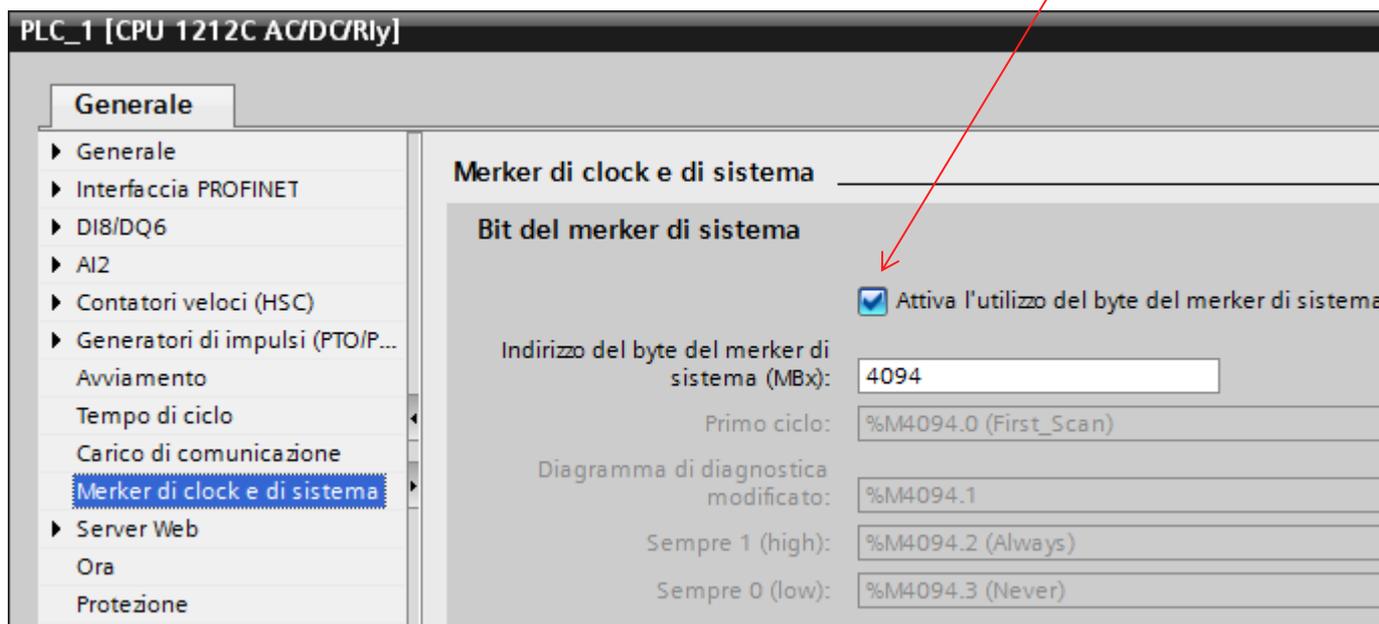
La seguente tabella illustra il significato dei merker di sistema:

Bit del byte del merker di sistema	7	6	5	4	3	2	1	0
Significato	Riservato (=0)	Riservato (=0)	Riservato (=0)	Riservato (=0)	=0	=1	=1 se cambia lo stato della diagnostica	=1 nel primo ciclo dopo l'avvio, altrimenti 0



MERKER DI SISTEMA/2

- Nelle proprietà della Cpu scegliere la voce “**Merker di clock e di sistema**”. Per attivare la funzione basta scegliere il byte da associare e spuntare la casella relativa. Premere OK prima di uscire.





I Merker di Clock

- Durante la scrittura di un programma a volte nasce l'esigenza di dover far lampeggiare una lampada ad una frequenza fissa. Come si fa?
- Il 1200 mette a disposizione un **merker Byte di clock** dove sono presenti 8 merker, che hanno la capacità di fornire l'uscita a varie frequenze (fisse).
- Il merker byte di clock **non è fisso**. Può essere liberamente scelto e impostato, durante la parametrizzazione dei dati della CPU.



MERKER DI CLOCK/1

Nelle proprietà della CPU scegliere "Merker di clock e di sistema".

Scegliere l'indirizzo del Byte e attivare l'utilizzo.

PLC_1 [CPU 1212C AC/DC/Rly]

Generale

- ▶ Generale
- ▶ Interfaccia PROFINET
- ▶ DI8/DQ6
- ▶ AI2
- ▶ Contatori veloci (HSC)
- ▶ Generatori di impulsi (PTO/...
- Avviamento
- Tempo di ciclo
- Carico di comunicazione
- Merker di clock e di sistema**
- ▶ Server Web
- Ora
- Protezione
- Risorse di collegamento
- Panoramica indirizzi

Sempre 0 (low): %M4094.3 (Never)

Bit del merker di clock

Attiva l'utilizzo del byte del merker di clock

Indirizzo del byte del merker di clock (MBx): 4095

Frequenze di clock

Clock 10 Hz	%M4095.0 (Clock_01_S)
Clock 5 Hz	%M4095.1 (Clock_02_S)
Clock 2,5 Hz	%M4095.2 (Clock_04_S)
Clock 2 Hz	%M4095.3 (Clock_05_S)
Clock 1,25 Hz	%M4095.4 (Clock_08_S)
Clock 1 Hz	%M4095.5 (Clock_1_S)
Clock 0,625 Hz	%M4095.6 (Clock_1.6_S)
Clock 0,5 Hz	%M4095.7 (Clock_2_S)



IN.SI.

CONTROLLORI PROGRAMMABILI

- Corso PLC Siemens S7-1200
- Set Istruzioni Base 2

© 2010-2017 Gilberto Padovani - Tutti i diritti riservati - E' espressamente vietata qualsiasi duplicazione del presente documento.

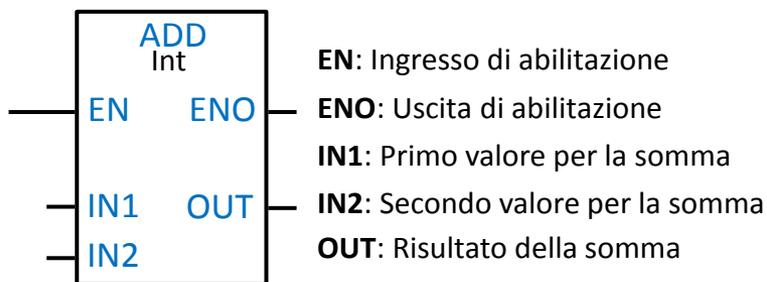
Tutti i diritti sono riservati a norma di legge. Nessuna parte di questo documento può essere riprodotta senza l'autorizzazione dell'autore, Gilberto Padovani.

E' espressamente vietato trasmettere ad altri il seguente documento, né in formato cartaceo, né elettronico, né per denaro né a titolo gratuito.

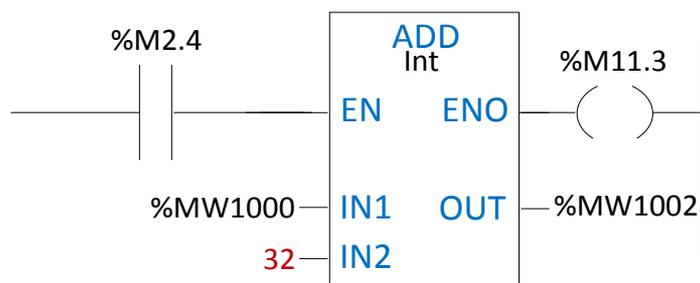
Tutti i marchi, i brevetti registrati, i software o porzioni di essi descritti in questo documento, sono dei legittimi proprietari.



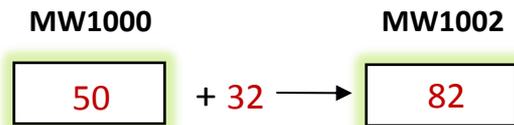
ADD_Int



- L'istruzione **ADD_Int** **somma** numeri interi a 16 bit. L'operazione somma il contenuto di **IN1** con **IN2** e pone il risultato in **OUT**.

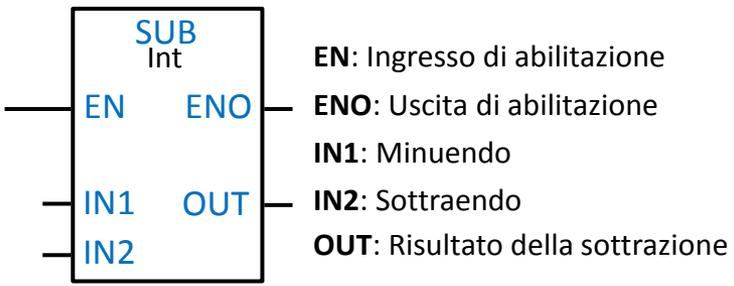


Quando il bit **M2.4** passa da OFF ad ON, il contenuto di **MW1000** viene **sommato** a **32**. Il risultato viene memorizzato in **MW1002**.

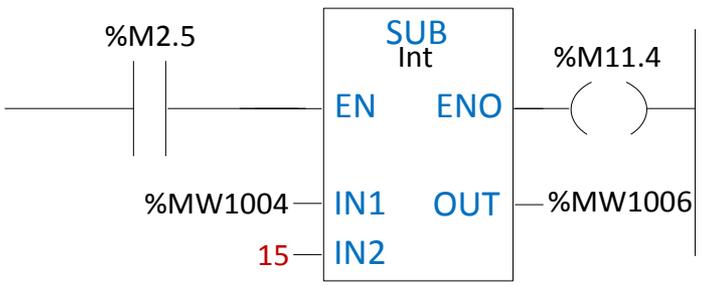




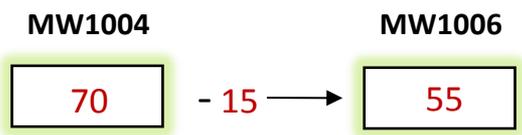
SUB_Int



- L'istruzione **SUB_Int** esegue una **sottrazione** tra numeri interi a 16 bit. L'operazione sottrae a **IN1** il contenuto di **IN2** e pone il risultato in **OUT**.

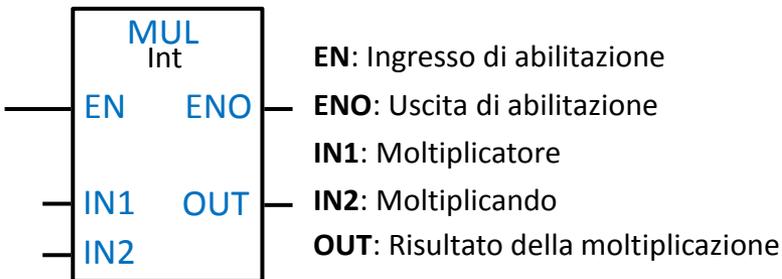


Quando il bit **M2.5** passa da OFF ad ON, il contenuto di **MW1004** viene **sottratto** a **15**. Il risultato viene memorizzato in **MW1006**.

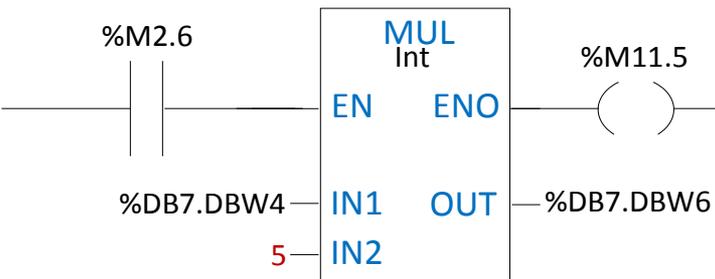




MUL_Int



- L'istruzione **MUL_Int** esegue una **moltiplicazione** tra numeri interi a 16 bit. L'operazione moltiplica **IN1** con **IN2**, il risultato viene posto in **OUT**.

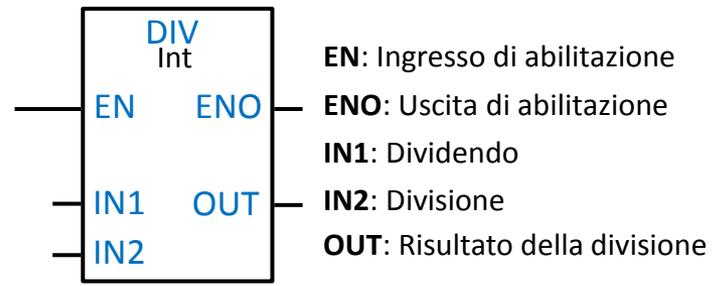


Quando il bit **M2.6** passa da OFF ad ON, il contenuto di **DB7.DBW4** viene **moltiplicato** con **5**. Il risultato viene memorizzato in **DB7.DBW6**

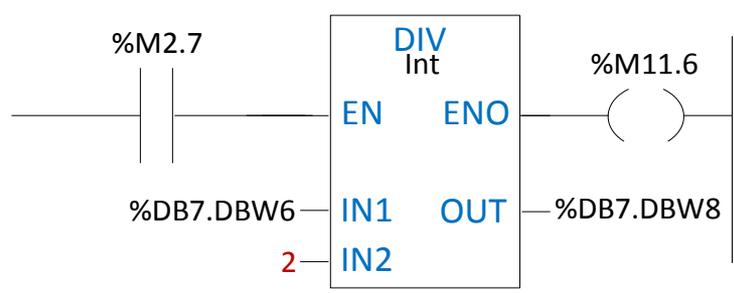




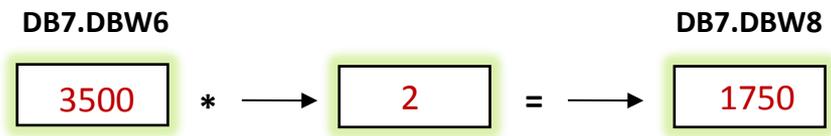
DIV_Int



- L'istruzione **DIV_Int** esegue una **divisione** tra numeri interi a 16 bit. L'operazione divide **IN1** con **IN2**, il risultato viene posto in **OUT**.

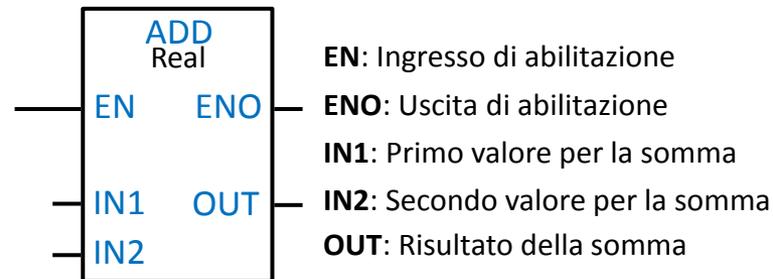


Quando il bit **M2.7** passa da OFF ad ON, il contenuto di **DB7.DBW6** viene **diviso per 2**. Il risultato viene memorizzato in **DB7.DBW8**

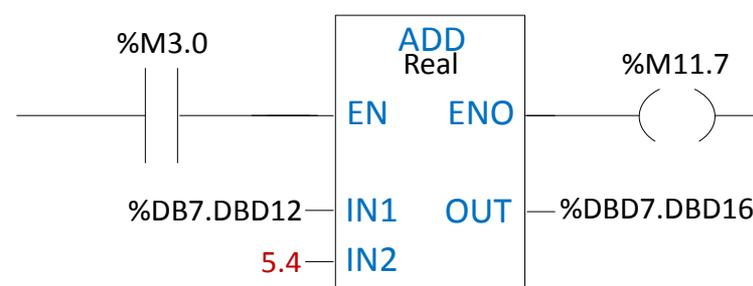




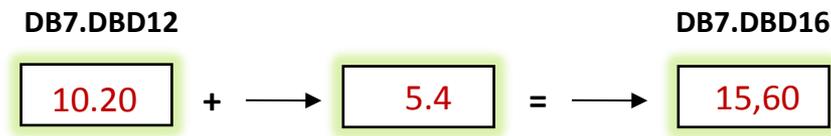
ADD_Real



- L'istruzione **ADD_Real** somma numeri in virgola mobile (*numeri reali*). L'operazione somma il contenuto di **IN1** con **IN2** e pone il risultato in **OUT**.



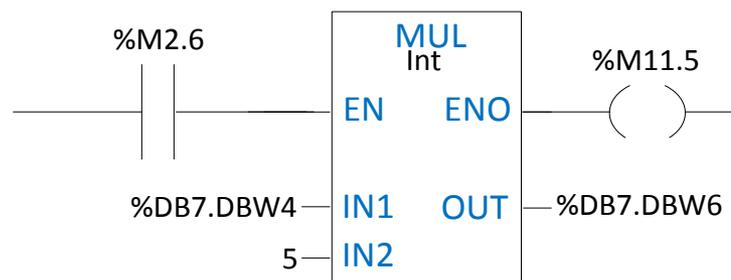
Quando il bit **M3.0** passa da OFF ad ON, il contenuto di **DB7.DBD12** viene **sommato** a **5.4**. Il risultato viene memorizzato in **DB7.DBD16**





NOTE SULLE ISTRUZIONI MATEMATICHE

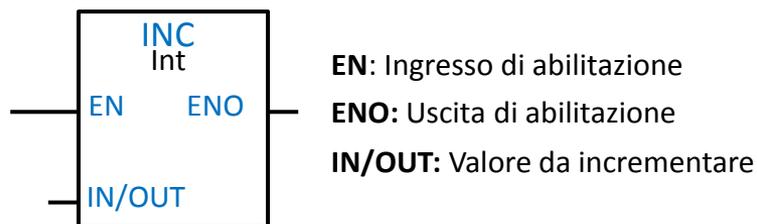
- L'esecuzione delle istruzioni matematiche possono **introdurre degli errori** nel caso in cui i risultati, determinano dei valori fuori range.



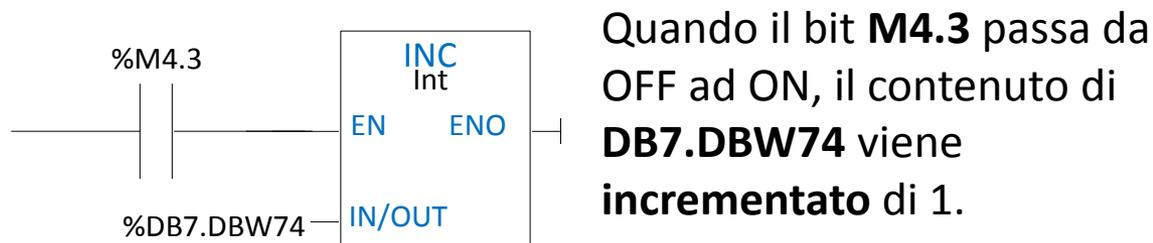
- In questo caso se il risultato della moltiplicazione non rientra nel campo permesso per un numero a **16 bit** il marker **M11.5** viene posto a **OFF**.



INC

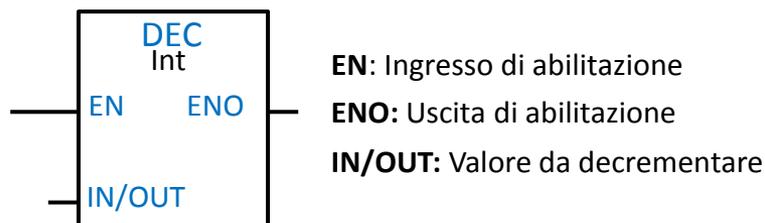


- L'istruzione **INC** consente di **incrementare** il valore dell'operando nel parametro IN/OUT, al successivo valore superiore e di leggere il risultato.

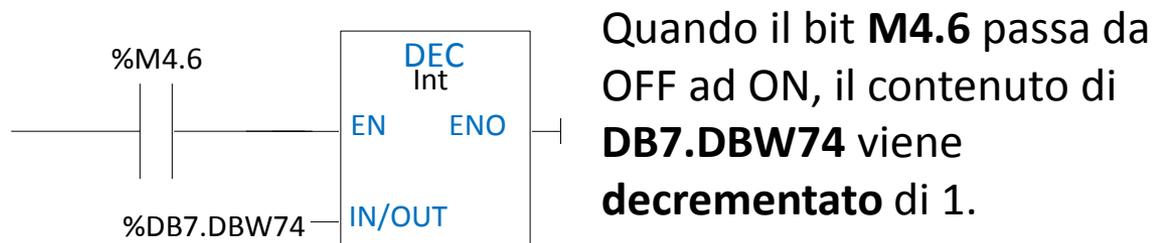




DEC

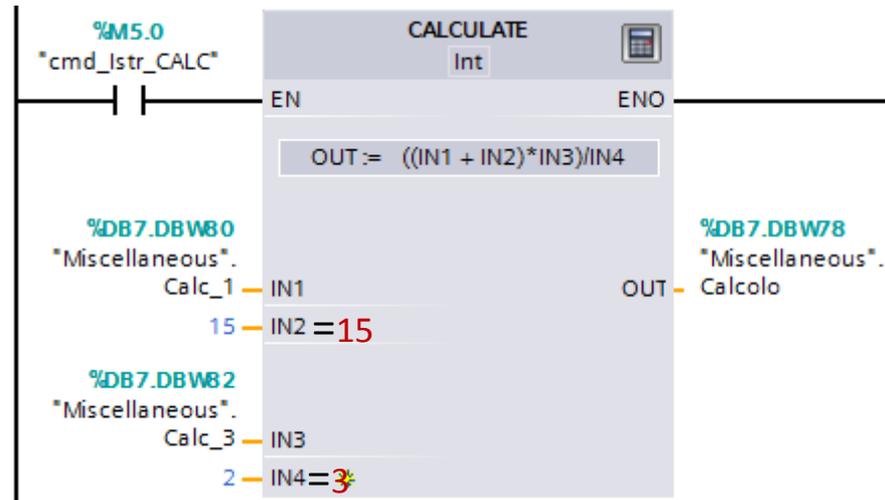


- L'istruzione **DEC** consente di **decrementare** il valore dell'operando nel parametro IN/OUT, al successivo valore inferiore e di leggere il risultato.





CALCULATE



- L'istruzione **CALCULATE** consente di definire ed eseguire un'espressione per il calcolo di operazioni matematiche o combinazioni logiche complesse in funzione del tipo di dati scelto.

$$\begin{array}{ccccccc}
 \text{DB7.DBW80} & & & \text{DB7.DBW82} & & & \text{DB7.DBW78} \\
 \boxed{15} & + & \boxed{15} & * & \boxed{3} & / & \boxed{2} = \boxed{45} \\
 & & & & \longrightarrow & &
 \end{array}$$



- Le istruzioni di confronto **vengono eseguite tra due valori** (costanti e/o contenuti dei canali specificati) e creano una condizione di esecuzione **ON** quando la condizione di confronto è vera.
- Per il confronto si possono utilizzare numeri Interi a **8 Bit**, numeri Interi a **16** e a **32 Bit**, numeri in **Virgola mobile** (*numeri reali*).
- Oltre ai numeri è possibile effettuare le comparazioni anche sulle **stringhe**, sui **singoli caratteri**, sulle variabili di tipo **Time** e su quelle **Date and Time**.





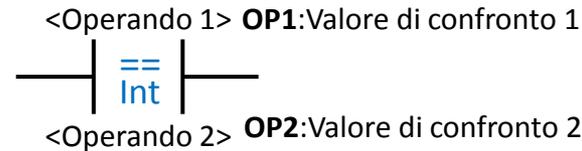
- I tipi fondamentali di confronto sono:
 - == Uguale
 - <> Diverso
 - > Maggiore di
 - < Minore di
 - >= Maggiore o uguale
 - <= Minore o uguale
- Il formato dei dati che possono essere inseriti in una comparazione:

Stringhe di Bit, numeri Interi, numeri in Virgola mobile, caratteri, Time, Date, Tod, Dtl.

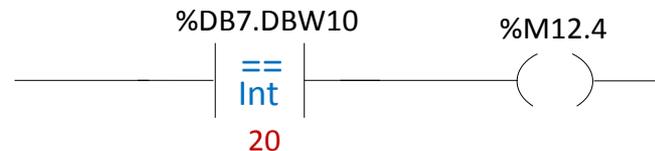




CMP ==



- Compara se il contenuto di **OP1** e **OP2** (costanti o valori dei canali specificati) sono **uguali**.

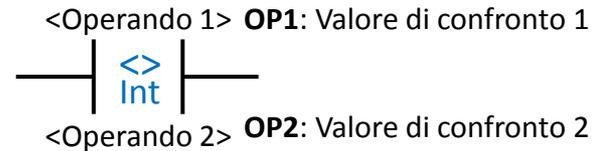


- Se il contenuto di DB7.DBW10 è **uguale** a 20, la condizione di esecuzione è vera e il merker M12.4 diventa ON.

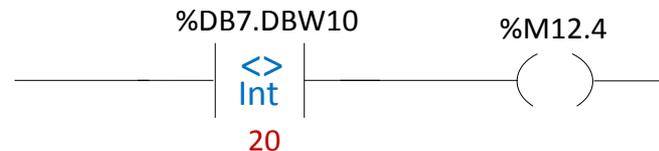




CMP <>



- Compara se il contenuto di **OP1** e **OP2** (costanti o valori dei canali specificati) sono **diversi**.

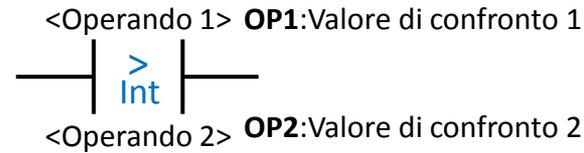


- Se il contenuto di DB7.DBW10 è **diverso** da **20**, la condizione di esecuzione è vera e il merker M12.4 diventa ON.

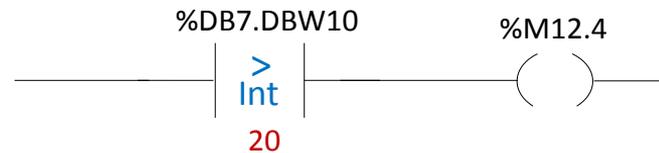




CMP >



- Compara se il contenuto di **OP1** è **maggiore** di **OP2** (costante o valore di un canale specificato).

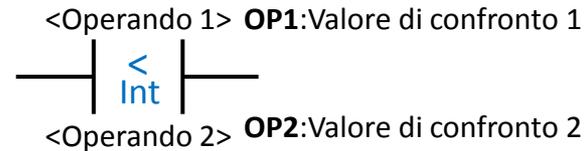


- Se il contenuto di DB7.DBW10 è **maggiore** di **20**, la condizione di esecuzione è vera e il merker M12.4 diventa ON.

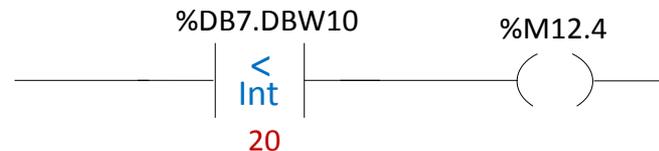




CMP <



- Compara se il contenuto di **OP1** è **minore** di **OP2** (costante o valore di un canale specificato).

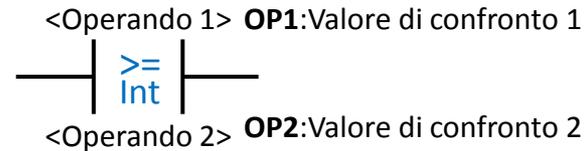


- Se il contenuto di DB7.DBW10 è **minore** di **20**, la condizione di esecuzione è vera e il merker M12.4 diventa ON.

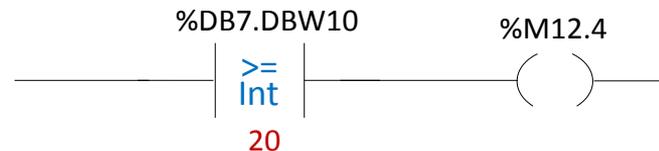




CMP >=



- Compara se il contenuto di **OP1** è **maggiore** o **uguale** a **OP2** (costante o valore di un canale specificato).

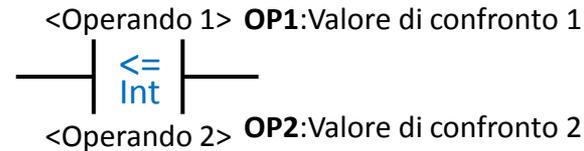


- Se il contenuto di DB7.DBW10 è **maggiore** o **uguale** a **20**, la condizione di esecuzione è vera e il merker M12.4 diventa ON.

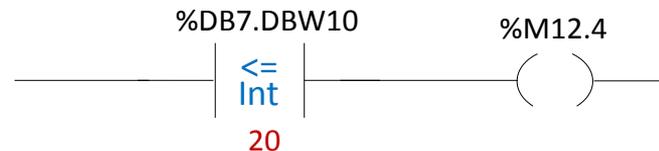




CMP <=



- Compara se il contenuto di **OP1** è **minore** o **uguale** a **OP2** (costante o valore di un canale specificato).

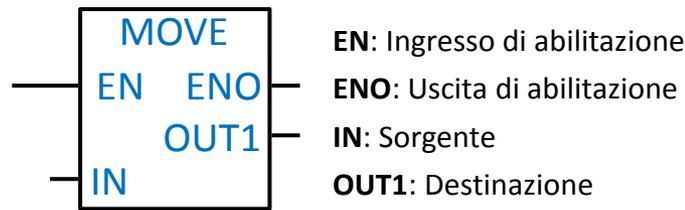


- Se il contenuto di DB7.DBW10 è **minore** o **uguale** a **20**, la condizione di esecuzione è vera e il merker M12.4 diventa ON.

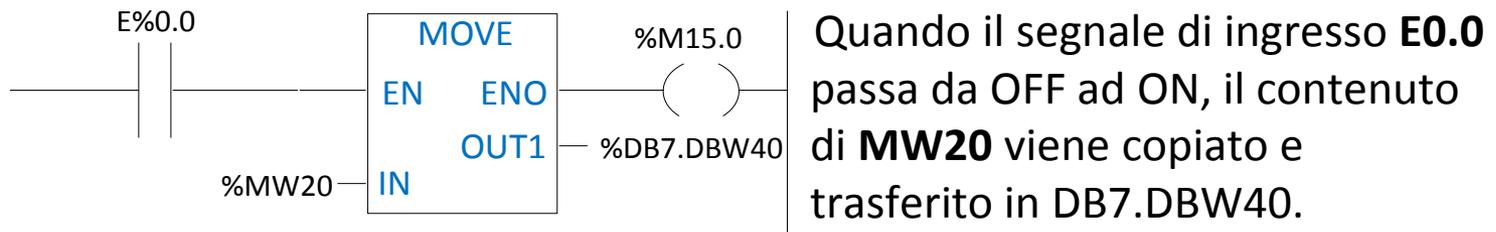




MOVE

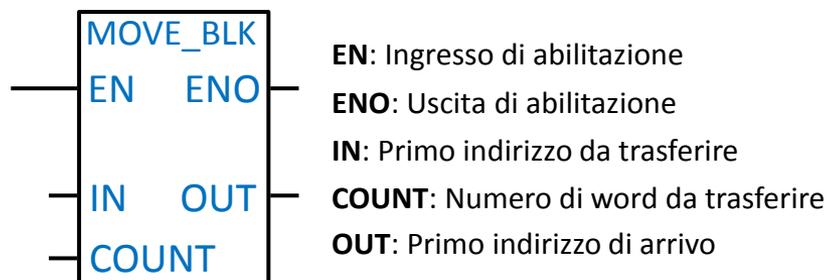


- L'istruzione **MOVE** (*Copia Valore*) consente di trasferire il contenuto dell'operando dell'ingresso **IN**, copiandolo nell'operando dell'uscita **Out1**.

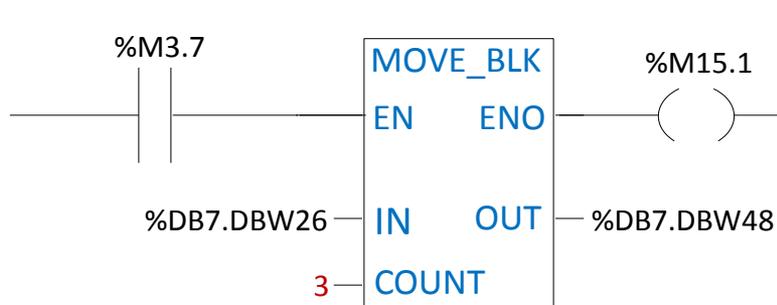




MOVE_BLK



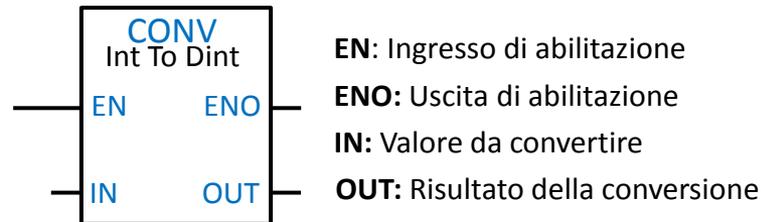
- L'istruzione **MOVE_BLK** (*Muovi blocchi di Word*) trasferisce a partire dall'indirizzo "IN" il numero di word specificato dall'operando "COUNT" copiandole a partire dall'indirizzo specificato in "OUT".



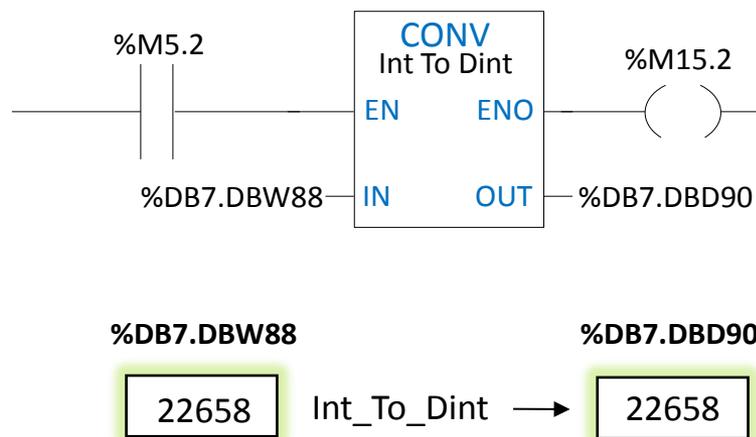
Quando il segnale di ingresso **M3.7** passa da OFF ad ON, il contenuto da **DB7.DBW26** a **DB7.DBW30** viene trasferito in **DB7.DBW48, .DBW50, .DBW52**



Int_To_Dint



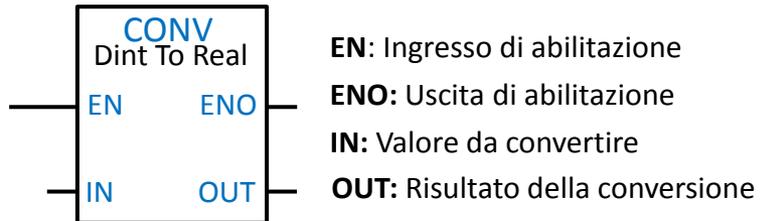
- L'istruzione **Int_To_Dint** (Intero-Doppio Intero) converte un numero **Intero** a **16** bit, in un numero **Intero** a **32** Bit.



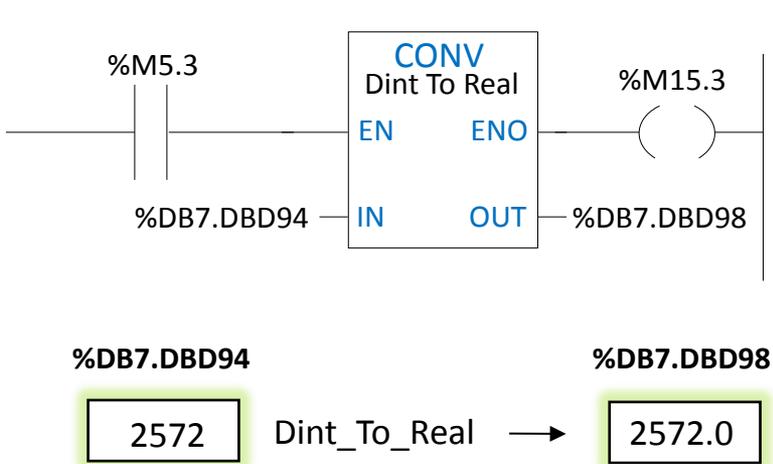
Quando il merker **M5.2** passa da OFF ad ON, il contenuto di **DB7.DBW88** viene letto come un numero intero a 16 bit e convertito in un numero intero a 32 bit su **DB7.DBD90**



Dint_To_Real



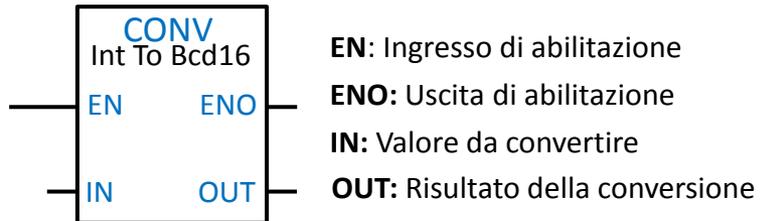
- L'istruzione **Dint_To_Real** (Doppio Intero-Reale) converte un numero **Intero** a 32 bit, in un numero a **Virgola mobile**.



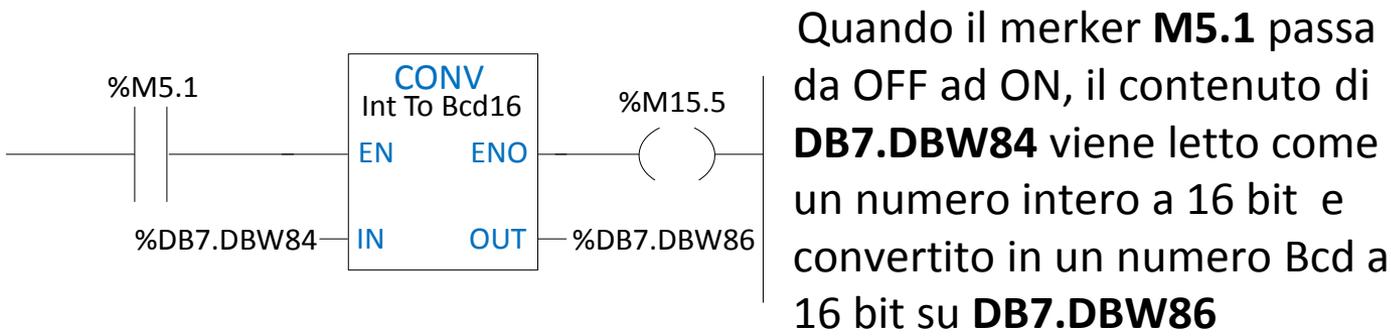
Quando il merker **M5.3** passa da OFF ad ON, il contenuto di **DB7.DBD94** viene letto come un numero intero a 32 bit e convertito in un numero a virgola mobile su **DB7.DBD98**



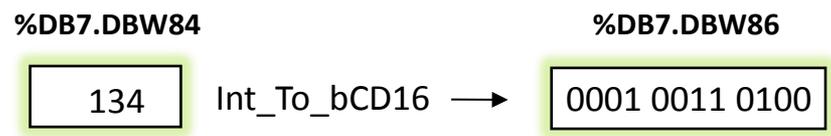
Int_To_Bcd16



- L'istruzione **Int_To_Bcd16** (Intero - Bcd 16 Bit) converte un numero **Intero** a 16 bit, in un numero **Bcd** anch'esso a 16 Bit.

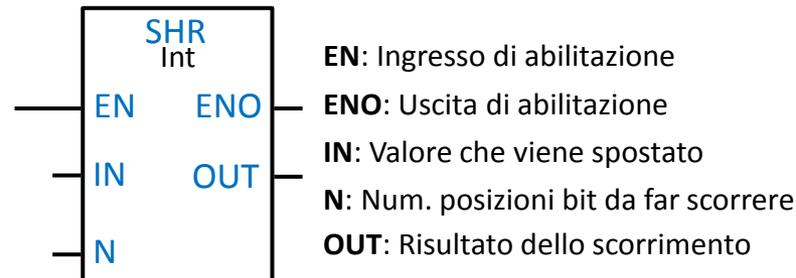


Quando il merker **M5.1** passa da OFF ad ON, il contenuto di **DB7.DBW84** viene letto come un numero intero a 16 bit e convertito in un numero Bcd a 16 bit su **DB7.DBW86**





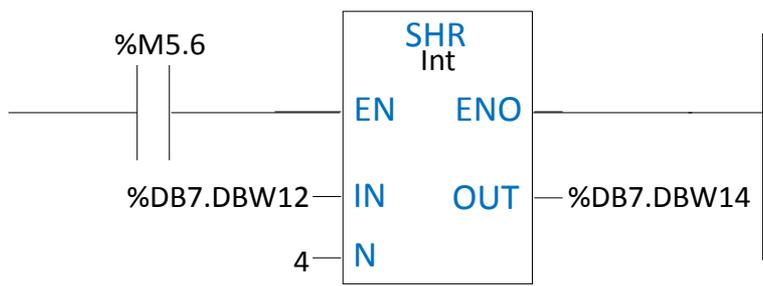
SHR_Int



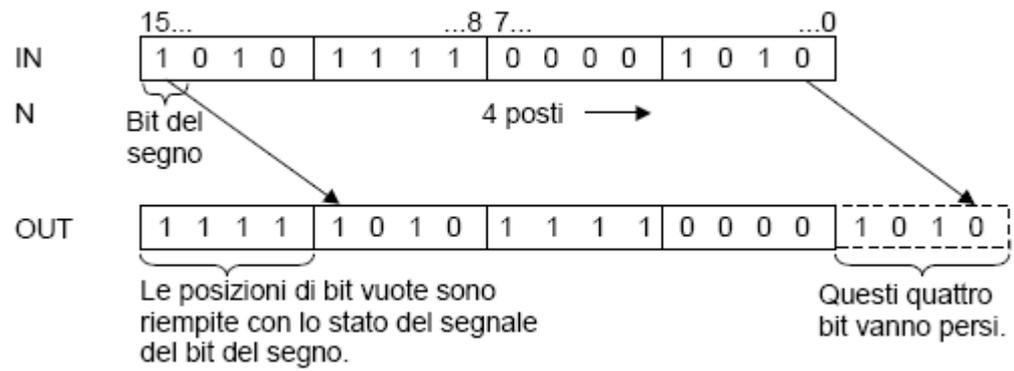
- L'istruzione **SHR_Int** (*Shift Register Right*) consente di spostare verso destra, il contenuto dell'operando nell'ingresso **IN**, bit per bit. Il numero di posizioni con cui deve essere spostato il valore, viene indicato nel parametro **N**.
- Il risultato dello scorrimento sarà presente nel parametro di uscita **Out**.



SHR_Int

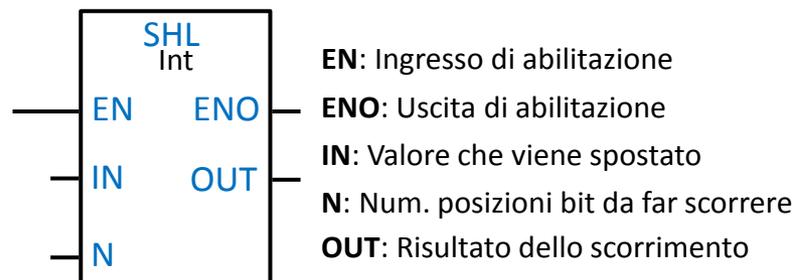


L'istruzione **SHR_Int** viene attivata quando **M5.6 = ON**. **DB7.DBW12** viene caricata e fatta **scorrere a destra** nella misura del **numero di bit** specificato nell'operando **N**. Il risultato viene trasferito su **DB7.DBW14**.





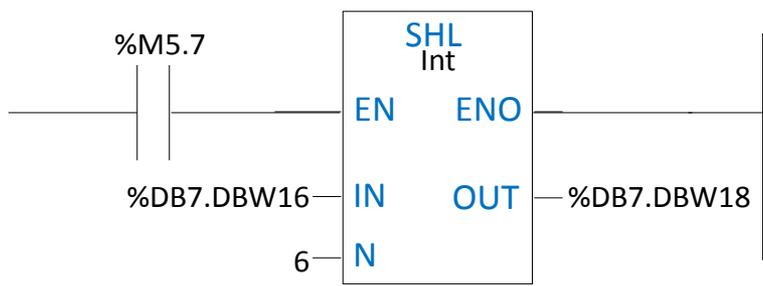
SHL_Int



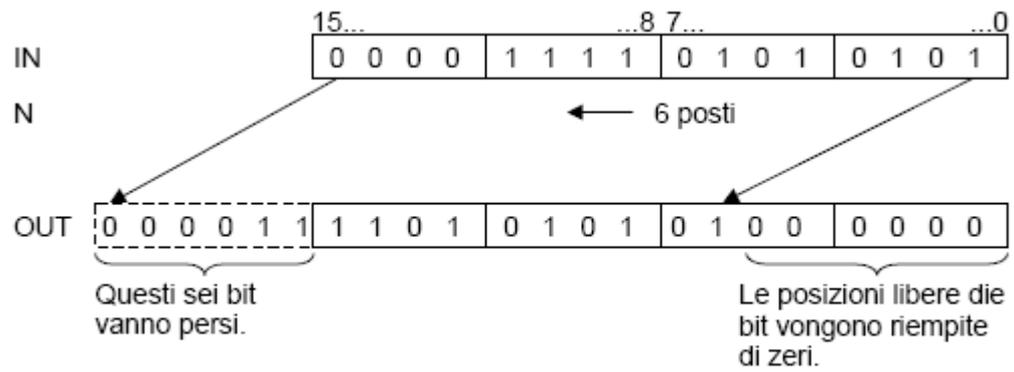
- L'istruzione **SHL_Int** (*Shift Register Left*) consente di spostare verso sinistra, il contenuto dell'operando nell'ingresso **IN**, bit per bit. Il numero di posizioni con cui deve essere spostato il valore, viene indicato nel parametro **N**.
- Il risultato dello scorrimento sarà presente nel parametro di uscita **Out**.



SHL_Int

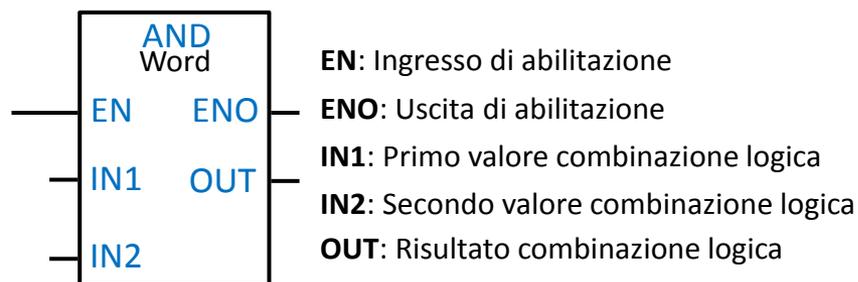


L'istruzione **SHL_Int** viene attivata quando **M5.7** = ON. DB7.DBW16 viene caricata e fatta **scorrere a sinistra** nella misura del **numero di bit** specificato nell'operando **N**. Il risultato viene trasferito su DB7.DBW18.





AND_Word



- L'istruzione **AND_Word** esegue un **AND** logico tra i Bit corrispondenti nei valori dei canali e/o costanti specificati in **In1** e **In2**. Il risultato viene posto in **Out**.

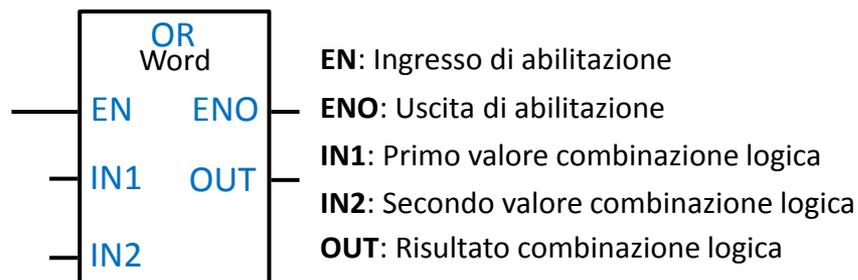
IN1	IN2	OUT
1	1	1
1	0	0
0	1	0
0	0	0

Esempio:

```
IN1  01010101 01010101
IN2  00000000 00001111
OUT  00000000 00000101
```



OR_Word



- L'istruzione **OR_Word** esegue un **OR** logico tra i Bit corrispondenti nei valori dei canali e/o costanti specificati in **In1** e **In2**. Il risultato viene posto in **Out**.

IN1	IN2	OUT
1	1	1
1	0	1
0	1	1
0	0	0

Esempio:

```
IN1  01010101 01010101
IN2  00000000 00001111
OUT  01010101 01011111
```