



POLITECNICO DI MILANO

V Facoltà di Ingegneria



 POLITECNICO DI MILANO  
DEI

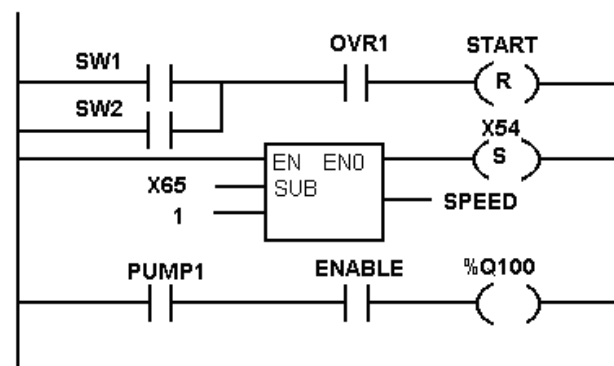
# P L C – IEC61131

## *Programmable Logic Controller*

Sistemi ad eventi Discreti



- **Introduzione**
- **Hardware di base**
- **Architettura OS e Funzionale**
- **La normativa IEC61131**
- **Conclusioni**





- ✓ **E' una apparecchiatura elettronica programmabile per il controllo di macchine e processi industriali.**
  
- ✓ **La struttura hardware a bus interno ha dato origine all'attuale PC**
  
- ✓ **Sostituto della logica cablata e dei quadri di controllo a relé**
  - Riduzione tempi e costi







- ✓ **Cresce la complessità degli impianti e in parallelo devono crescere:**

- Affidabilità, Modificabilità e Flessibilità



***Soluzione***

- ✓ **Si è quindi passati a dispositivi a logica programmata:**

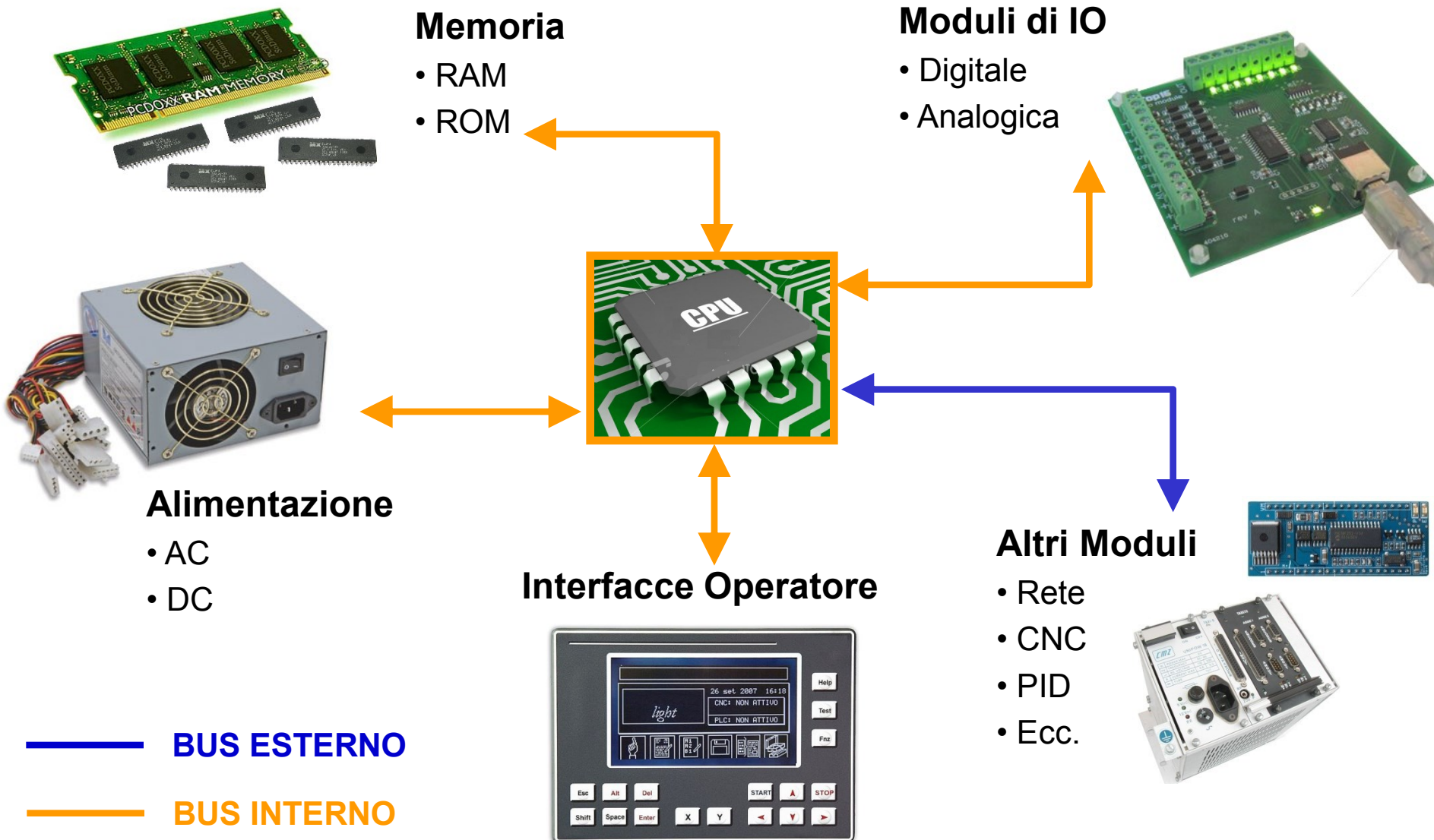
- **PLS** (Programmable Logic Sequencers)
- **PLC** (Programmable Logic controllers)
  - 1969 negli U.S.A. nell'industria automobilistica



***Vantaggi***

- ✓ **Istruzioni invece che componenti elettrici:**

- Adattabilità, semplicità e rapidità di progettazione
- Facile diagnostica
- Più adatta al controllo di sistemi complessi

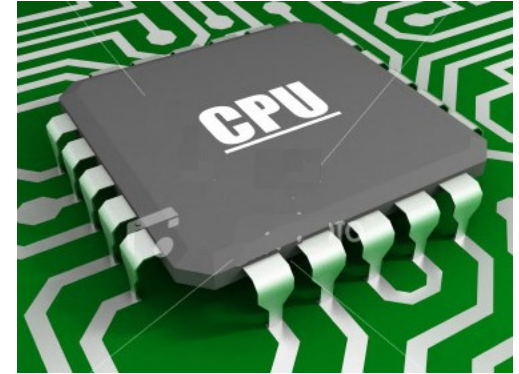






## ✓ **Struttura dedicata a:**

- Elaborazione dei dati
- Elaborazione delle istruzioni del programma di controllo
- Gestione delle operazioni logiche interne al PLC.



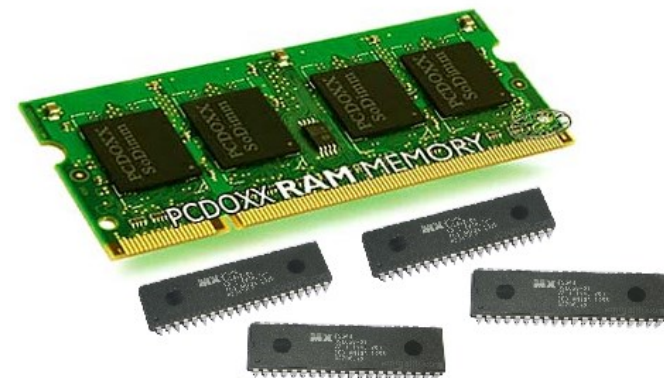
## ✓ **Esempio di tempi di elaborazione:**

- Operazioni su bit, parole, in virgola fissa: 0.1-5  $\mu$ s
- Operazioni in virgola mobile: 0.5-50  $\mu$ s



## ✓ Le memorie si dividono in :

- Volatili e permanenti (“latched”)
- Ad accesso casuale (RAM o attive) e di sola lettura (ROM o passive)



## ✓ Memoria Dati:

- Necessaria una memoria RAM (volatile); alcuni dati di controllo o impianto particolarmente importanti vanno salvati in memorie permanenti

## ✓ Memoria Programma:

- Preferibili memorie ROM, per limitare i rischi di perdita o corruzione del programma di controllo





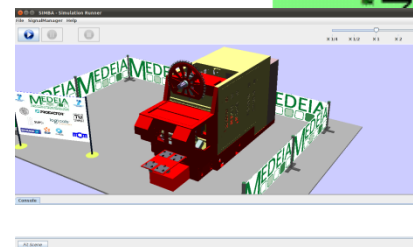
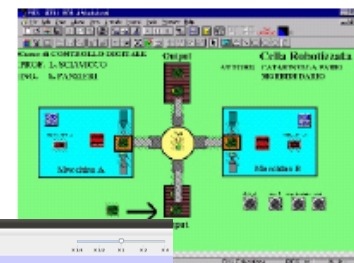
## ✓ Interfacce di programmazione

- Piccoli LCD e Tastierini
- Sistemi di sviluppo per PC



## ✓ Pannelli Operatore

- Display
- Terminali grafici
- Moduli con “Push buttons” e pulpiti
- Interfacce “all inclusive”
- HMI (Human-Machine Interface)





## ✓ Consentono la comunicazione con il processo fisico:

### ■ Ingressi/Uscite Digitali

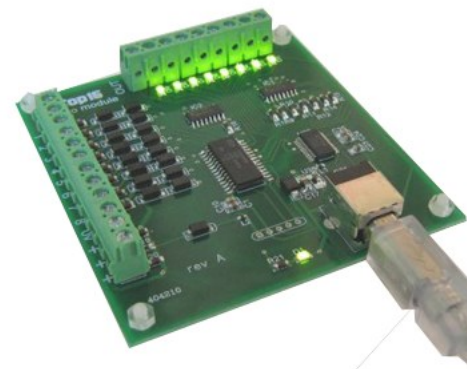
- #IO: 16-32;  $V_{in}$ : 24V DC;  $I_{in}$ : pochi mA; Is. Ottico

### ■ Ingressi Analogici

- #In: 8; Risol.: 12-14 bit; Is. ottico; ReadTime: 20 ms
- Collegabili a termocoppie, termoresistenze, ecc.

### ■ Uscite Analogiche

- #Out: 8; Risol.: 12-14 bit; Is. Ottico; WriteTime: 0.5 ms
- $V_{out}$ :  $\pm 10$  V, 0 -10 V, 1-5 V
- $I_{out}$ :  $\pm 20$  mA, 0 to 20 mA, 4 to 20 mA





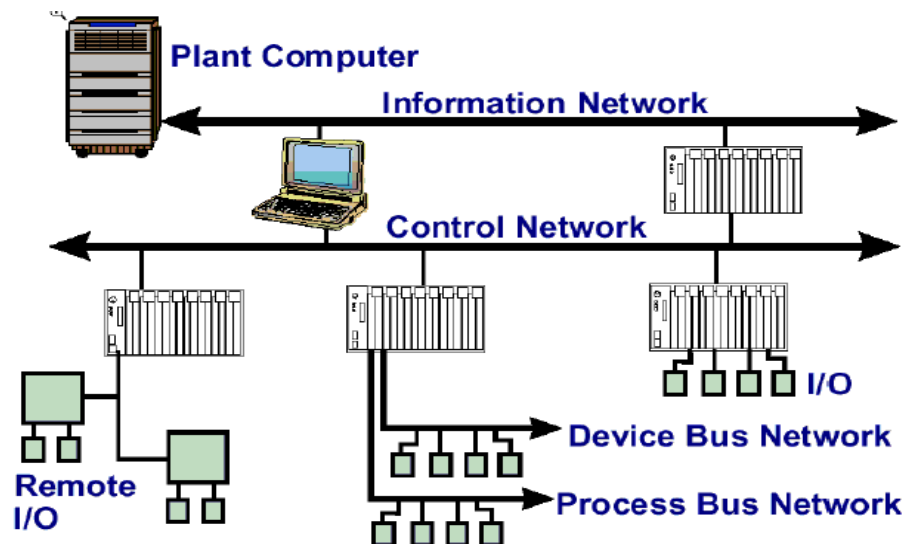
- ✓ **Schede conteggio veloce per gestire posizionamenti e controllo assi**
- ✓ **Schede PID**
  - Processore separato che realizza FdC PID
- ✓ **Moduli co-processor (gestione dati non logici: tabelle, ecc.)**
- ✓ **Moduli di Backup**
  - Duplicazione di una scheda principale:
    - Back-up freddo: la scheda guasta viene sostituita
    - Back-up caldo: la scheda sostitutiva e scheda principale in parallelo; quando questa si guasta, interviene immediatamente quella di back-up
- ✓ **Moduli di connessione in rete**
  - BUS di Campo (**Field Bus**): Profibus, CAN, Modbus RS232/TCP, ecc.





- ✓ **Bus di campo (fieldbus) è il termine fissato da IEC per indicare, in un processo automatizzato, lo standard di comunicazione tra diversi dispositivi (nodi), quali:**

- Dispositivi di campo (sensori, attuatori, ecc.)
- Dispositivi di controllo (PLC, DCS, ecc.)



- ✓ **La comunicazione tra i nodi è gestita secondo un protocollo standard o proprietario**
- ✓ **Tipi di connessioni logiche: stella o token-ring**

Modbus Master Figure





### ✓ Vantaggi del FieldBus:

- PLC e FIELDBUS sono integrabili
- Comunicazioni con il campo in forma digitale e veloce
- Abbattimento dei costi di cablaggio e installazione
- Parametrizzazione e configurazione remota dei dispositivi
- Utile per impianti distribuiti con necessità di “RealTime Control”
- Vari e semplici supporti (twisted pair, fibra ottica, radio,...)
- Non è tecnologia proprietaria: posso integrare dispositivi di produttori diversi

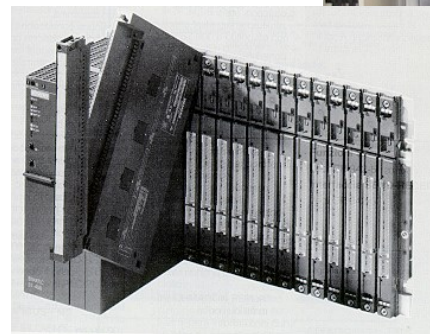
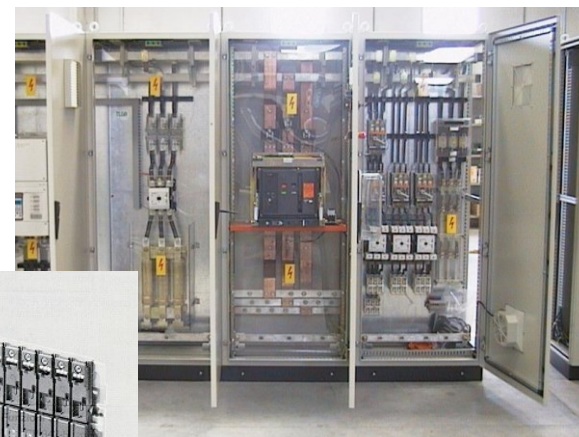
### ✓ Svantaggi del FieldBus:

- Costi dei singoli moduli
- Un bus guasto preclude la comunicazione tra molti dispositivi contemporaneamente (bus ridondato)
- Non immediato da installare



- ✓ **Il Rack contiene tutti i moduli (o schede) del sistema PLC (CPU, memorie, schede I/O, ecc.) e consente:**

- ✓ Connessione elettrica
- ✓ Connessione logica (bus)
- ✓ Connessione meccanica
- ✓ Schermatura



- ✓ **I Rack sono collegabili :**
  - ✓ Gerarchicamente con opportune schede di interfaccia e collegamento
  - ✓ In locale o decentralizzati
- ✓ **I Rack garantiscono la modularità dell'architettura PLC**





## ✓ Criteri di massima per il dimensionamento:

- Modularità
- Spazio di indirizzamento
- presenza di moduli speciali
- Velocità

## ✓ Categorie per dimensioni:

- *Compatti* (o micro):
  - Non modulari con poche decine di I/O
  - Applicazioni non industriali (building automation, distributori)
- *Piccoli*:
  - Modulari con un centinaio di I/O
- *Medi*:
  - Modulari con un migliaio di I/O
- *Grandi*:
  - Modulari con decine di migliaia di I/O





## PLC vs PC

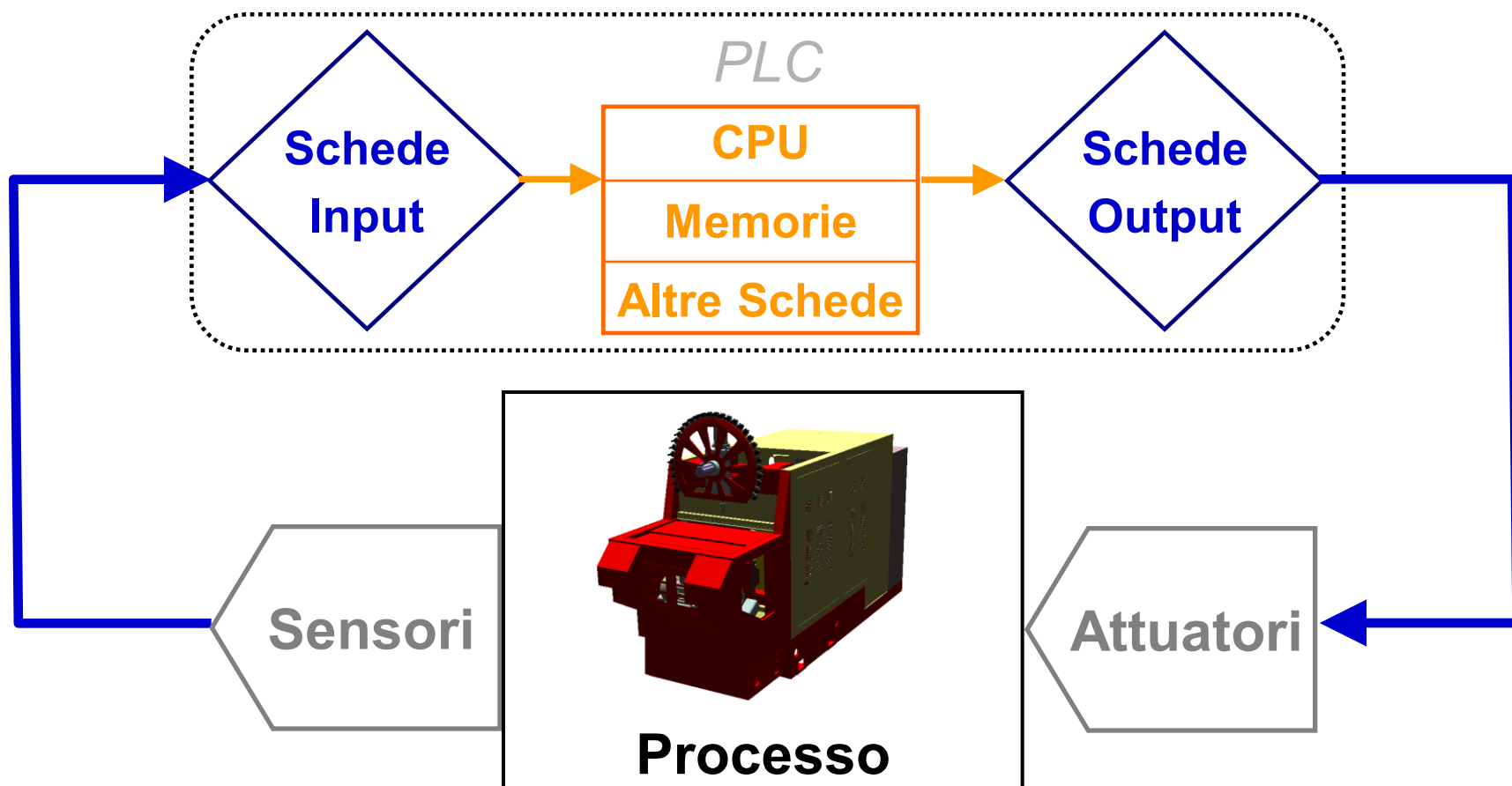
### ✓ PLC:

- Progettato per resistere in ambiente industriale
  - Vibrazioni, shock meccanici od elettrici, immunità ai disturbi (elettromagnetici, vapori, temperature, sostanza tossiche), isolamento galvanico, ecc.
- Non ha dischi mobili (HD, CD, FD)
- Dotato di “Wach-Dog” (per istruzioni, programma, ecc..)
- Sistema operativo proprietario estremamente affidabile ad elevata diagnostica sia sul SW che sull’HW
- Compatto, facilmente allocabile in un armadio elettrico
- Integrabile con altri componenti di controllo

... MA LE COSE CAMBIANO... SoftPLC...



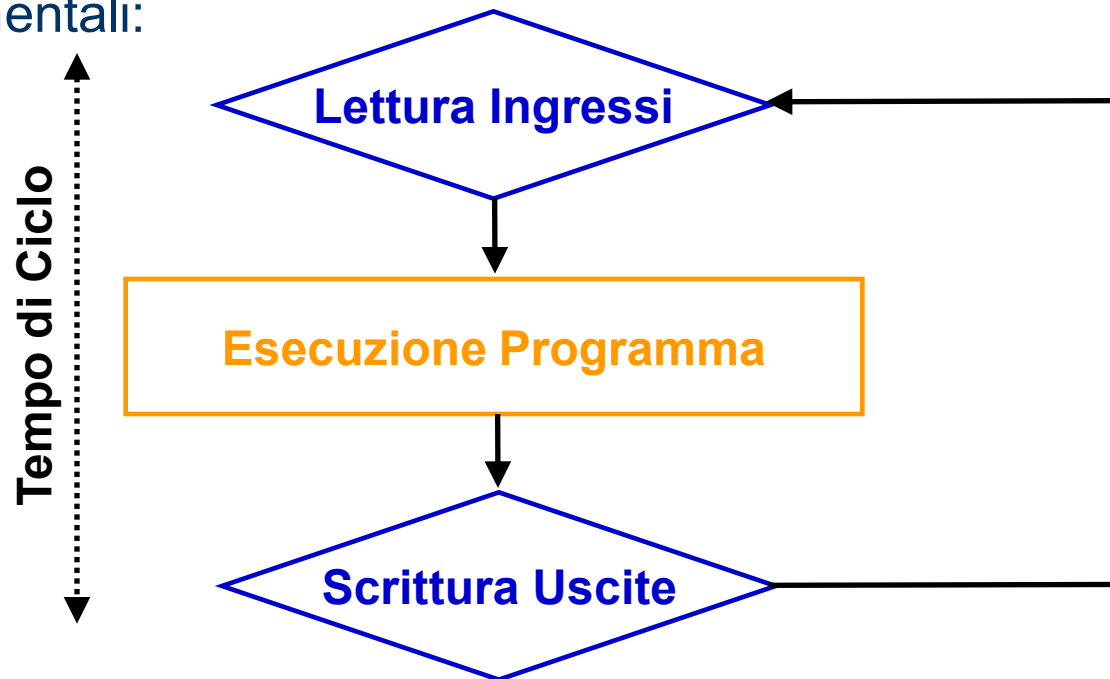
- ✓ Come altri controllori di impianti e processi industriali, il PLC si collega in “retroazione” rispetto ad essi:





## ✓ Il Ciclo del PLC

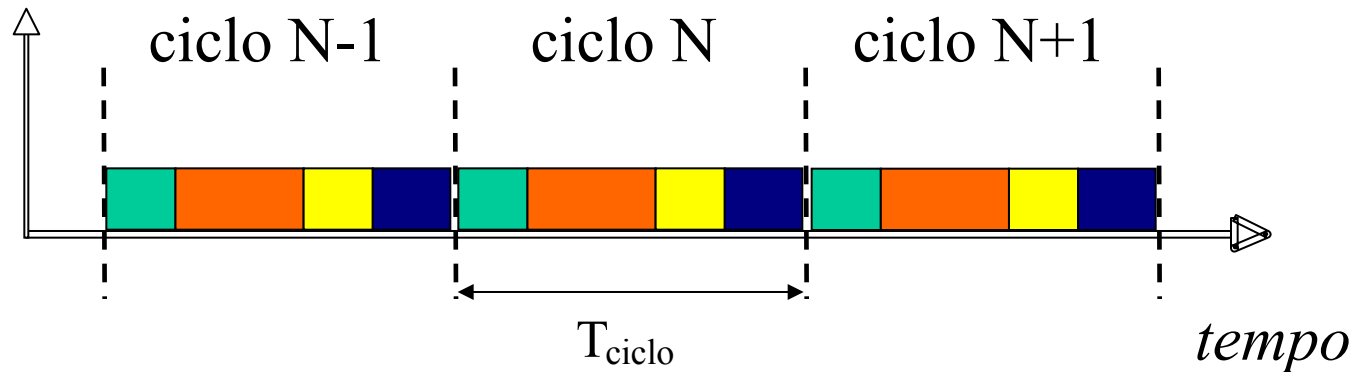
- ✓ In quanto dispositivo a segnali campionati, il PLC durante il funzionamento continua ad eseguire ciclicamente 3 fasi fondamentali:



- ✓ Nota: all'interno del tempo di ciclo vi sono anche le esecuzioni di istruzioni del sistema operativo!!



## ✓ Il Ciclo del PLC



*fase di lettura ingressi*



*fase di esecuzione del programma*



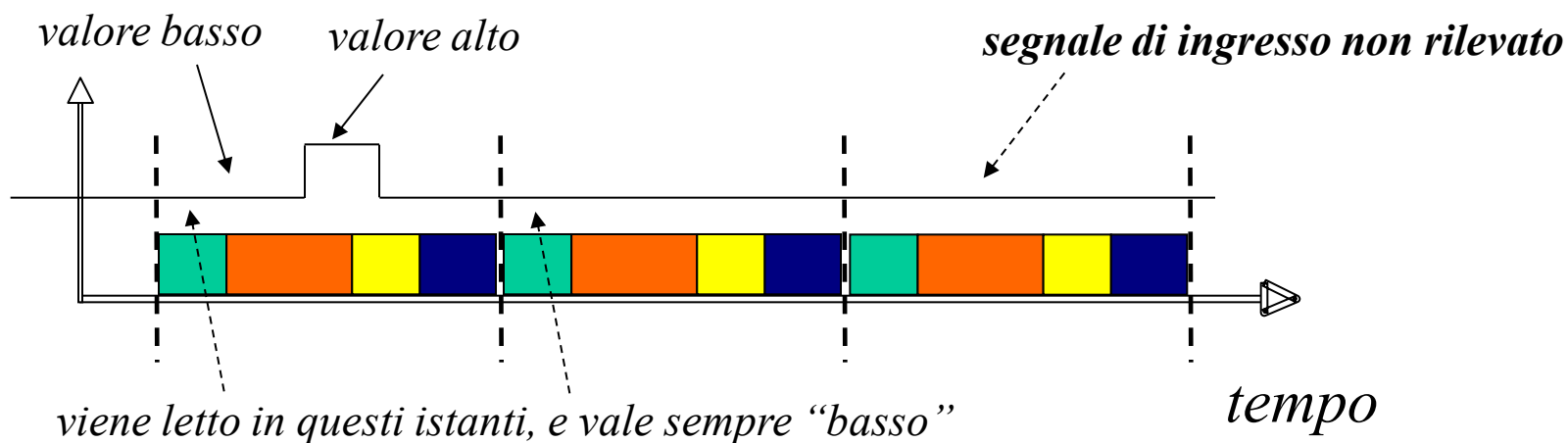
*fase di scrittura delle uscite*



*fase di esecuzione di porzioni di sistema operativo*




## ✓ Il Ciclo del PLC



 *fase di lettura ingressi*

 *fase di scrittura delle uscite*

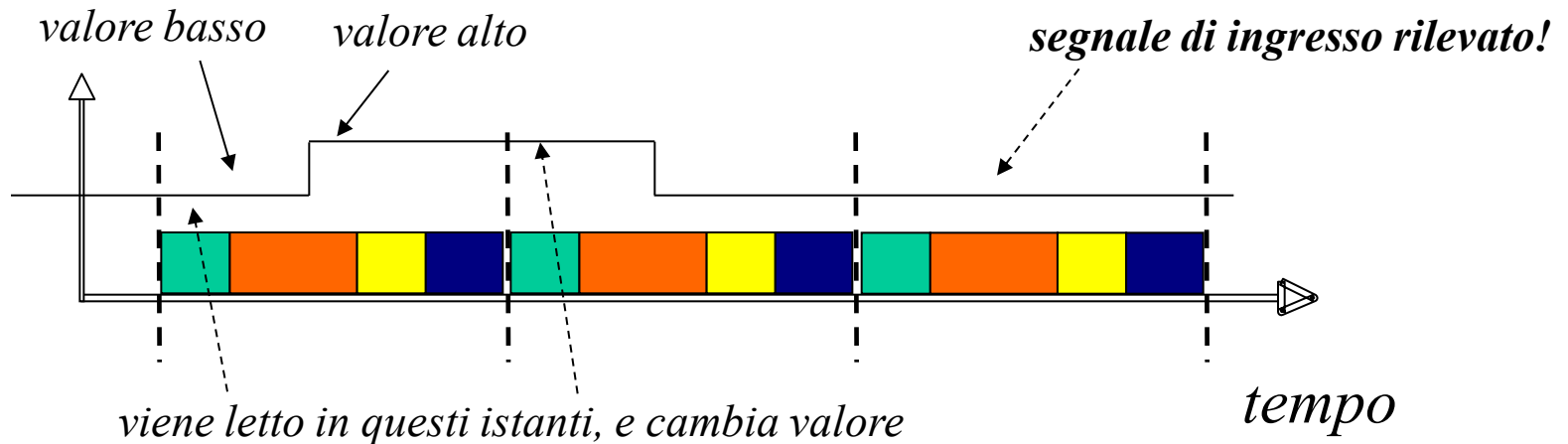
 *fase di esecuzione del programma*



 *sistema operativo*





## ✓ Il Ciclo del PLC



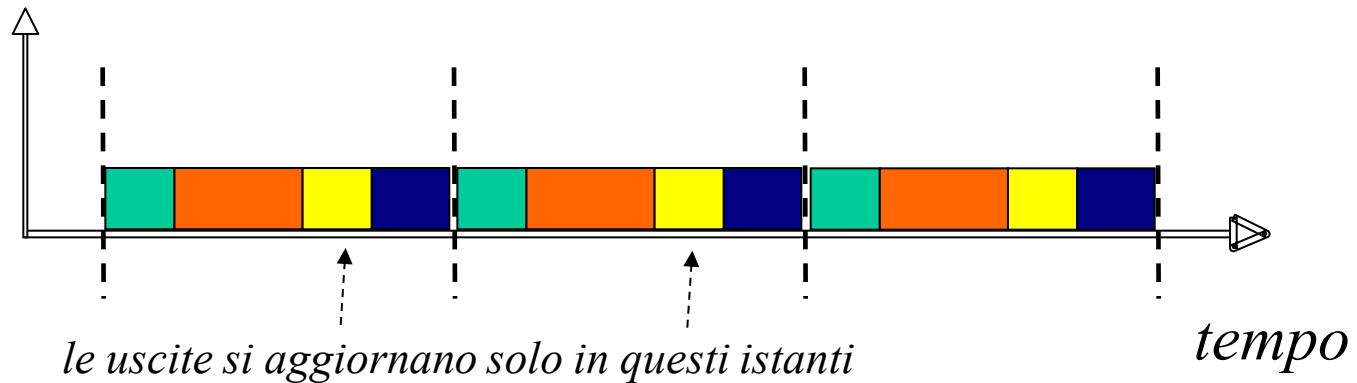
 *fase di lettura ingressi*  
 *fase di esecuzione del programma*

 *fase di scrittura delle uscite*  
 *sistema operativo*



## ✓ Il Ciclo del PLC


- ✓ Inutile aggiornare più volte un'uscita: questa si aggiorna una sola volta in Tciclo , durante la fase di scrittura!



 *fase di lettura ingressi*

 *fase di esecuzione del programma*

 *fase di scrittura delle uscite*

 *sistema operativo*



Il modo appena visto di funzionamento del PLC è detto a **Copia Massiva** (scrittura di tutte le uscite e lettura di tutti gli ingressi avvengono contemporaneamente).

✓ **Vantaggi:**

- Semplice da implementare e da capire: le fasi di lettura e scrittura sono singolarmente ininterrotte e nettamente separate
- Semplice da simulare: l'evoluzione del sistema e delle sue uscite è facilmente prevedibile

✓ **Svantaggi:**

- Scarsa reattività, se un ingresso muta subito dopo essere stato letto, la sua variazione viene recepita dopo un intero tempo di ciclo



- ✓ **Alcuni PLC ammettono anche cicli di funzionamento diversi:**
  - Lettura degli ingressi e la scrittura delle uscite anche durante la fase di esecuzione del programma. In questo modo, aumenta la frequenza di aggiornamento possibile per un ingresso o un'uscita
  
- ✓ **Attenzione a capire bene quando un'istruzione di uscita comporta**
  - Un'immediato aggiornamento della variabile fisica di uscita
  - Un'aggiornamento della variabile fisica a fine esecuzione
  
- ✓ **In alternativa, possiamo usare gli *interrupt***



## ✓ **Interrupt:**

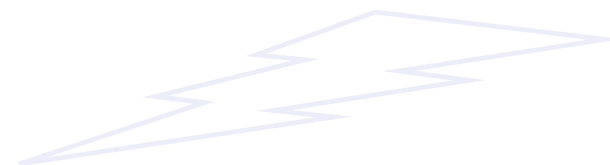
- 1) Interrompono il ciclo di programmazione
- 2) Eseguono una opportuna subroutine di gestione dell'interrupt
- 3) Restituiscono il controllo al programma

## ✓ **Interrupt Hardware**

- Generati da segnali fisici esterni (si usano per garantire una risposta veloce ad un evento)

## ✓ **Interrupt Ciclici (Software)**

- Generati periodicamente ad intervalli di tempo regolari (si usano tipicamente per campionare alcuni ingressi in tempi inferiori al tempo di ciclo)





## ✓ **Modalità di esecuzione:**

- Il PLC funziona come controllore: legge gli ingressi, esegue il programma di controllo e aggiorna le uscite

## ✓ **Modalità di validazione:**

- Il PLC esegue il programma, ma non legge gli ingressi fisici, né scrive le uscite fisiche: vengono utilizzate variabili intermedie, eventualmente connesse con il sistema di programmazione (ad esempio, un PC)

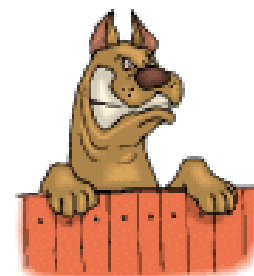
## ✓ **Modalità di programmazione:**

- Il PLC è connesso con il sistema di programmazione e accetta scritture della memoria programmi





- ✓ Sono sostanzialmente timer, che il sistema operativo associa a svariati componenti del PLC e all'esecuzione di svariate operazioni
- ✓ Ad un'operazione si associa una durata stimata massima, e quindi, se il timer arriva a contare tale intervallo di tempo prima che l'operazione sia conclusa, viene generato un errore (Watch-Dog, letteralmente “cane di guardia”)
- ✓ Esempi di operazioni con Watch-Dog:
  - ✓ Istruzioni utente
  - ✓ Istruzioni di accesso a memoria/schede di I/O / schede di rete
  - ✓ L'intero programma
  - ✓ Ecc.



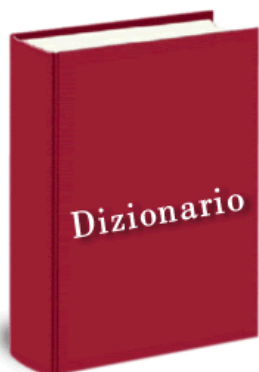


- ✓ **Normativa internazionale pubblicata la prima volta nel 1993 sull'uso dei PLC (seconda edizione 2003)**
  - Part 1: General information
  - Part 2: Equipment requirements and tests
  - Part 3: Programming languages
  - Part 4: User guidelines
  - Part 5: Messaging service specification
  - Part 6: Communications via fieldbus (Awaiting completion of fieldbus standards.)
  - Part 7: Fuzzy control programming
  - Part 8: Guidelines for the application and implementation of programming languages
- ✓ **Scopi principali:**
  - Ridurre il divario con le moderne tecniche informatiche
  - Formalizzare in modo più astratto ed essenziale i problemi di automazione e controllo
  - Proporre standard nel campo del controllo logico, per favorire la riusabilità della conoscenza e una reale concorrenza tra i costruttori



## ✓ Definizione di PLC – Programmable Logic Control

Sistema elettronico a funzionamento digitale, destinato all'uso in ambito industriale, che utilizza una memoria programmabile per l'archiviazione interna delle istruzioni orientate all'utilizzatore per l'implementazione di funzioni specifiche, come quelle logiche, di sequenziamento, di temporizzazione, di conteggio e di calcolo aritmetico, e per controllare, mediante ingressi ed uscite sia digitali che analogiche, vari tipi di macchine e processi.



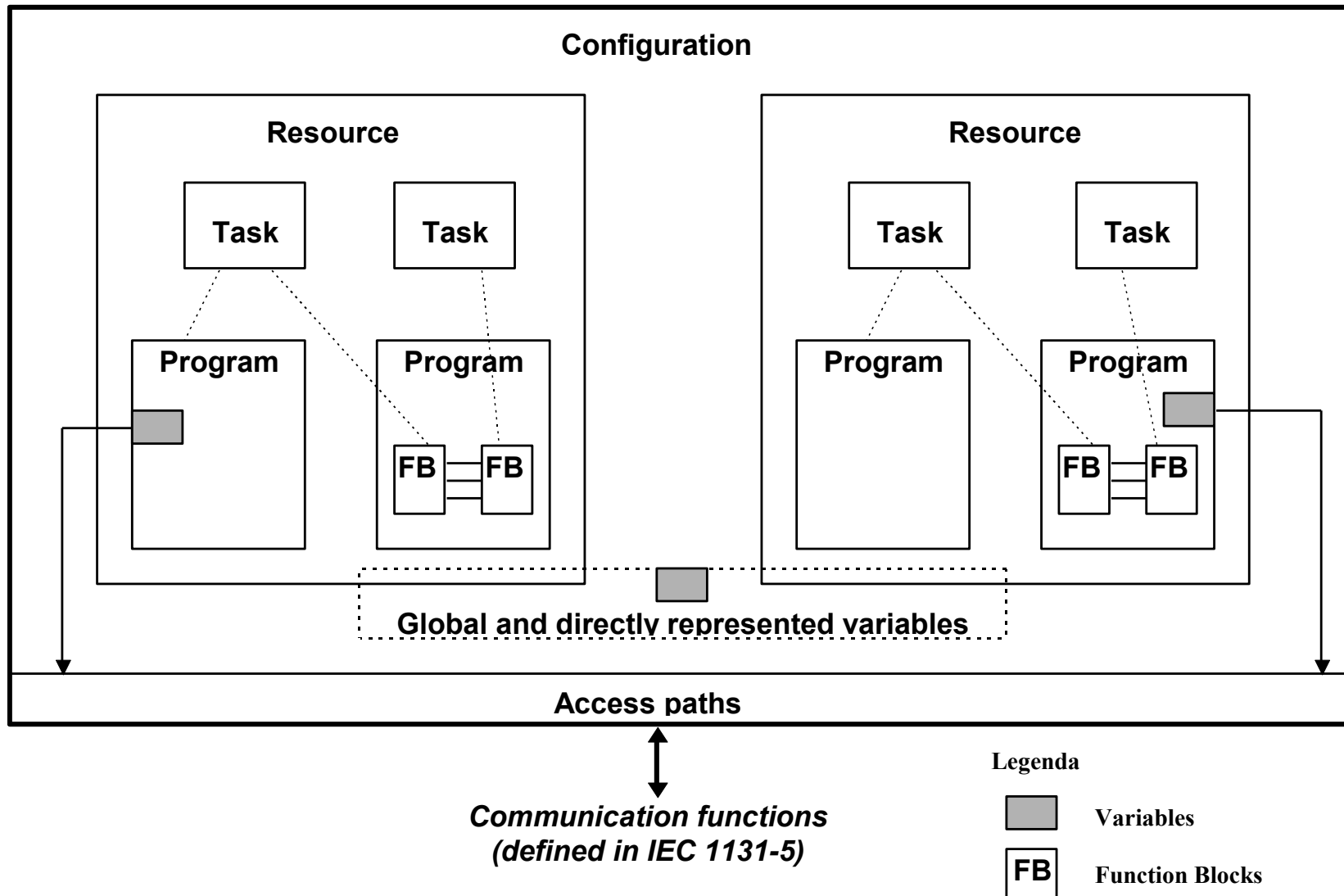


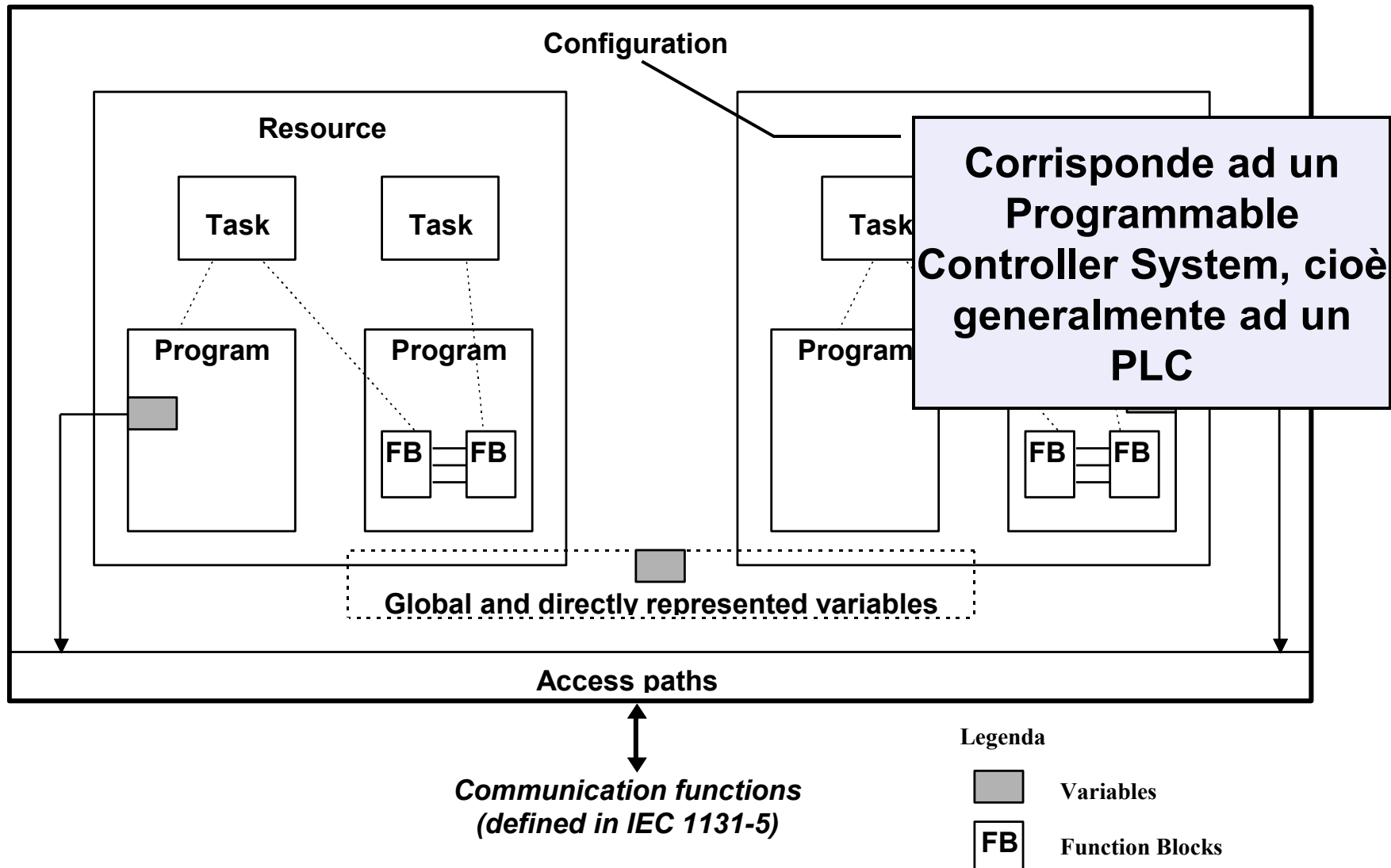
## ✓ **IEC61131-3 Programming Languages**

- ✓ Tipo di dato
- ✓ Variabili
- ✓ Configurazione
- ✓ Program Organization Units (POU)
- ✓ Configurazioni, Risorse e Tasks
- ✓ Collegamenti Esterni

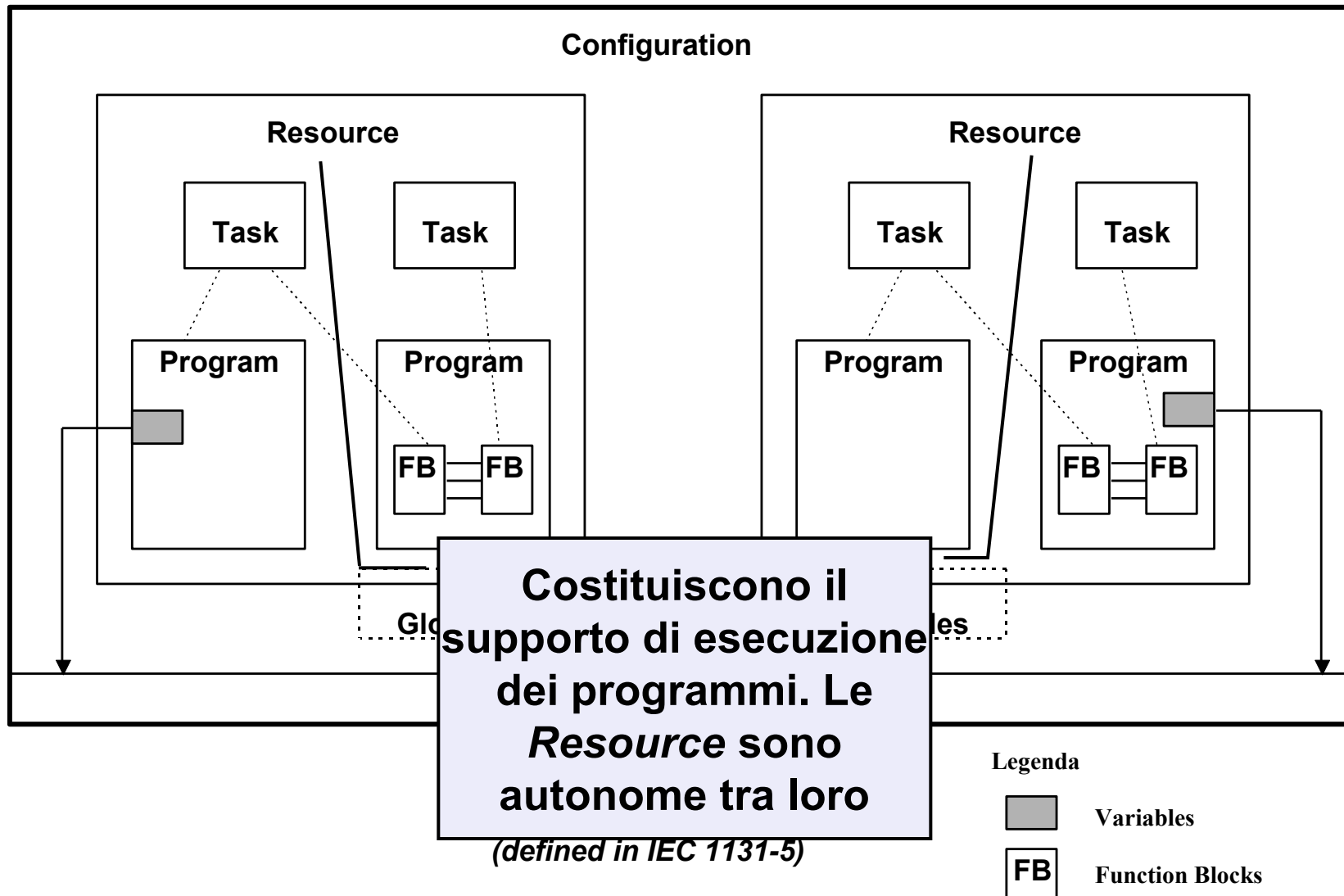
## ✓ **Linguaggi definiti**

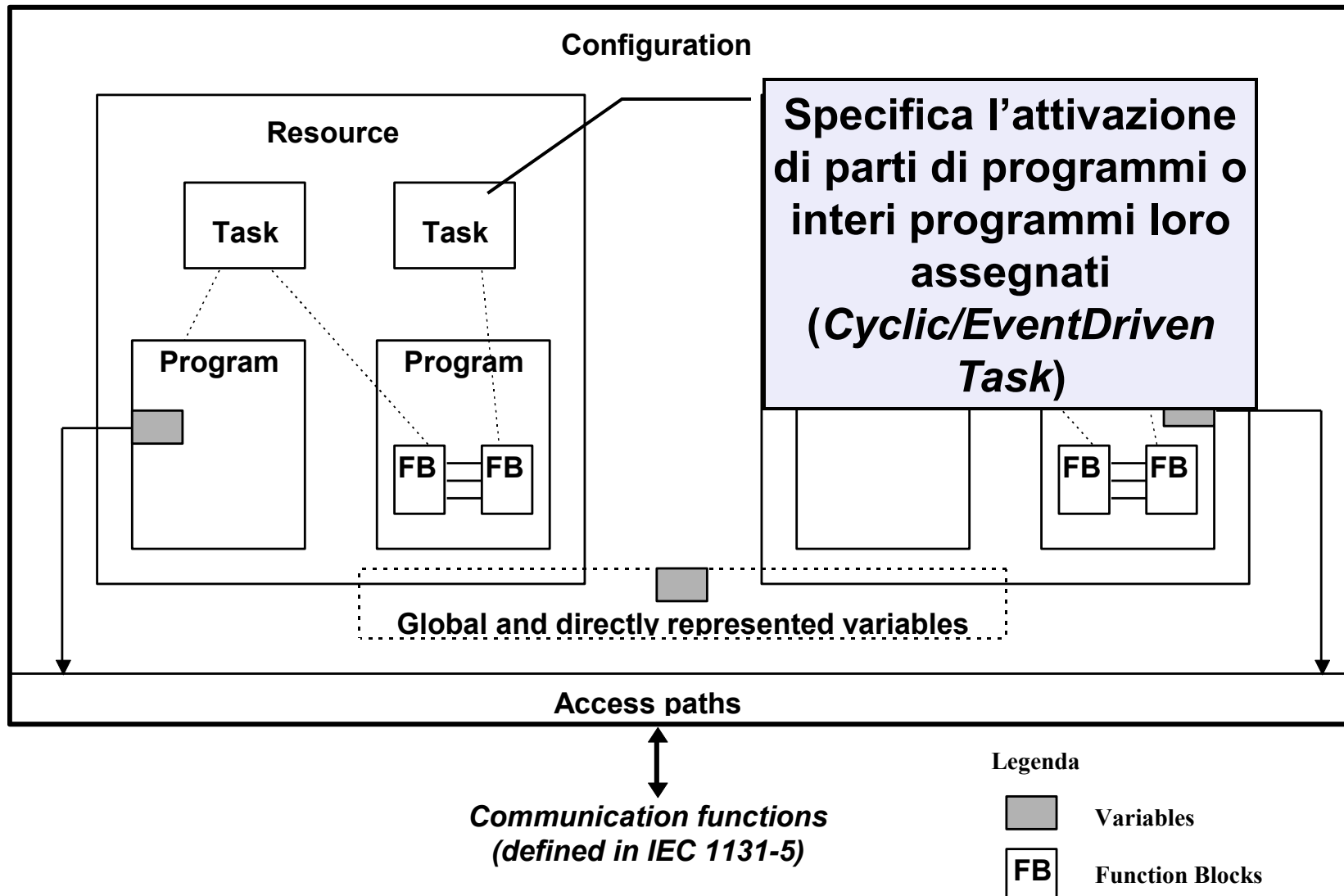
- ✓ Ladder diagram (LD), graphical
- ✓ Function block diagram (FBD), graphical
- ✓ Structured text (ST), textual
- ✓ Instruction list (IL), textual
- ✓ Sequential function chart (SFC), graphical

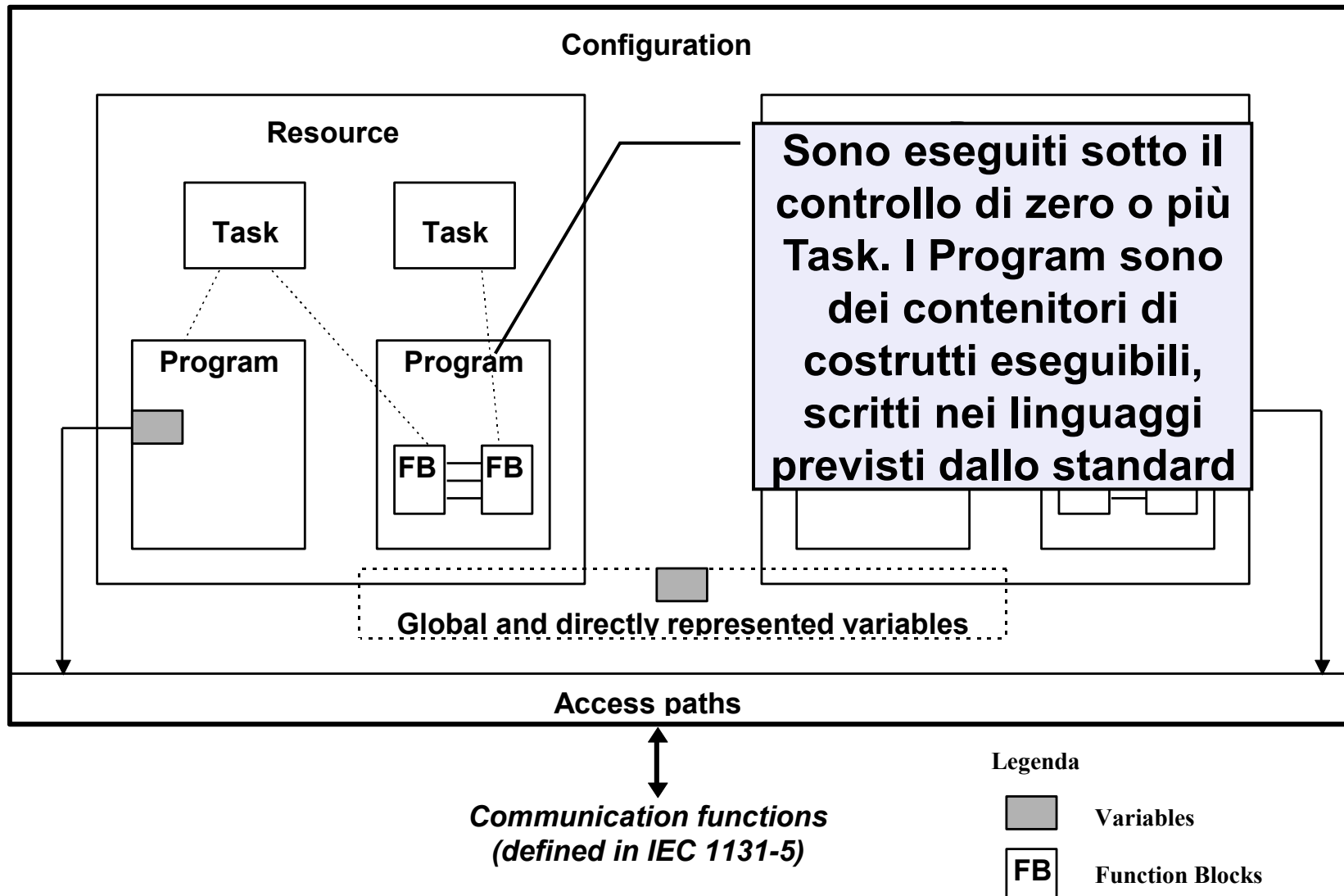














- ✓ I task specificano **l'attivazione di parti di programmi o interi programmi** loro assegnati. L'attivazione può essere periodica oppure condizionata al verificarsi di un particolare evento.

*Cyclic*



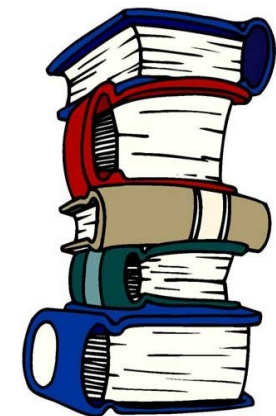
*Event Driven*



- ✓ I programmi quindi costituiscono l'apice di una struttura che il progettista può comporre gerarchicamente utilizzando i blocchi funzione (function block) e le funzioni (function).
- ✓ Blocchi Funzioni e Funzioni → Modularità!

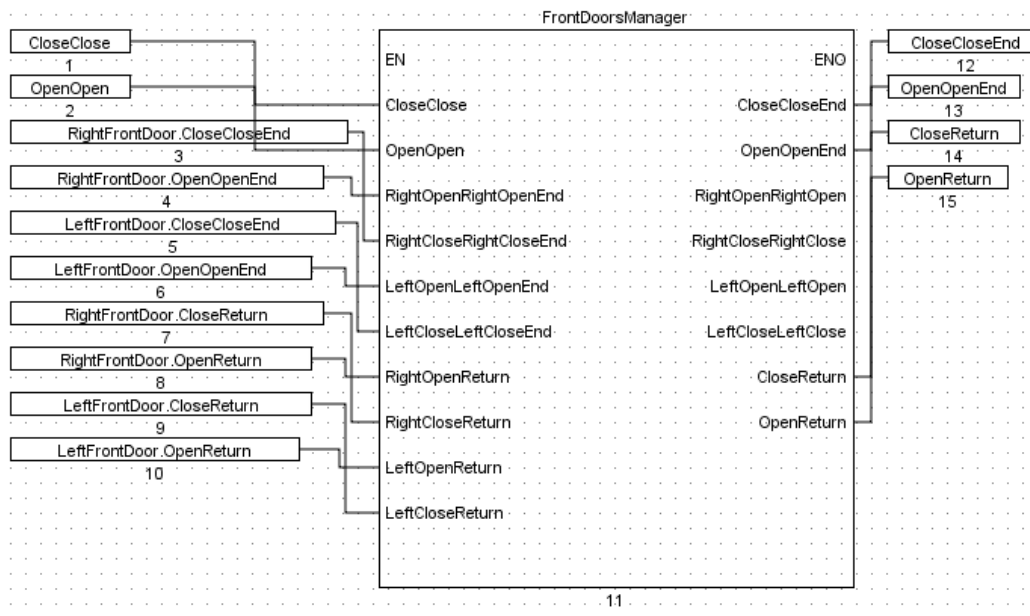


- ✓ Le funzioni sono **porzioni di costrutti eseguibili**, che restituiscono un valore dipendente dagli ingressi, senza avere variabili “interne” (di stato)
  
- ✓ Lo standard IEC 6-1131 permette la definizione e l'utilizzo di uno svariato numero di funzioni già definite in **libreria**, come ad esempio:
  - Funzioni matematiche
  - Funzioni trigonometriche
  - Funzioni logiche
  - Permette inoltre all'utente di definire le proprie funzioni e riutilizzarle liberamente in un progetto





- ✓ Lo standard permette inoltre la definizione di blocchi funzioni (a tutti gli effetti vere e proprie routine di codice, che possono essere dotate di variabili interne, o di stato)
- ✓ I blocchi funzione hanno la caratteristica di poter essere salvate in librerie e riutilizzate all'interno di vari progetti indipendentemente dal linguaggio usato





- ✓ **I moduli, che nello standard sono detti POU (Program Organisation Unit), sono:**
  - Programmi
  - Blocchi Funzione
  - Funzione
  
- ✓ **Facile (ri)utilizzo di porzioni di progetti precedentemente sviluppati o acquistati sul mercato sotto forma di librerie**



- ✓ Variabili **globali** e **locali** (dichiarate in una POU) con nomi mnemonici
- ✓ Tipi di variabili (**B** = byte, **b** = bit)
  - ✓ Bit Strings - groups of on/off values (**BOOL**(1), **BYTE**(8), **WORD**(16), **DWORD**(32), **LWORD**(64))
  - ✓ INTEGER - whole numbers (**SINT**(1B), **INT**(2B), **DINT**(4B), **LINT**(8B))
  - ✓ U - Unsigned - add a U to the type to make it unsigned integer
  - ✓ REAL - floating point IEC 559 (IEEE) (**REAL**(4B), **LREAL**(8B))
  - ✓ **TIME** - duration for timers, processes.
  - ✓ Date and Time of day (**DATE**, **TIME\_OF\_DAY**, **DATE\_AND\_TIME**)
  - ✓ **STRING** - character strings surrounded by single quotes
  - ✓ **WSTRING** - holds multi-byte strings
  - ✓ **ARRAY** - multiple values stored in the same variable
  - ✓ Derived - type derived from one of the above types
  - ✓ **STRUCT** - composite of several variables and types.
  - ✓ Generic (**ANY**)
- ✓ Proprietà “retain” per le variabili che devono essere persistenti al riavvio a “caldo”





## ✓ **Strutturazione interna di un POU**

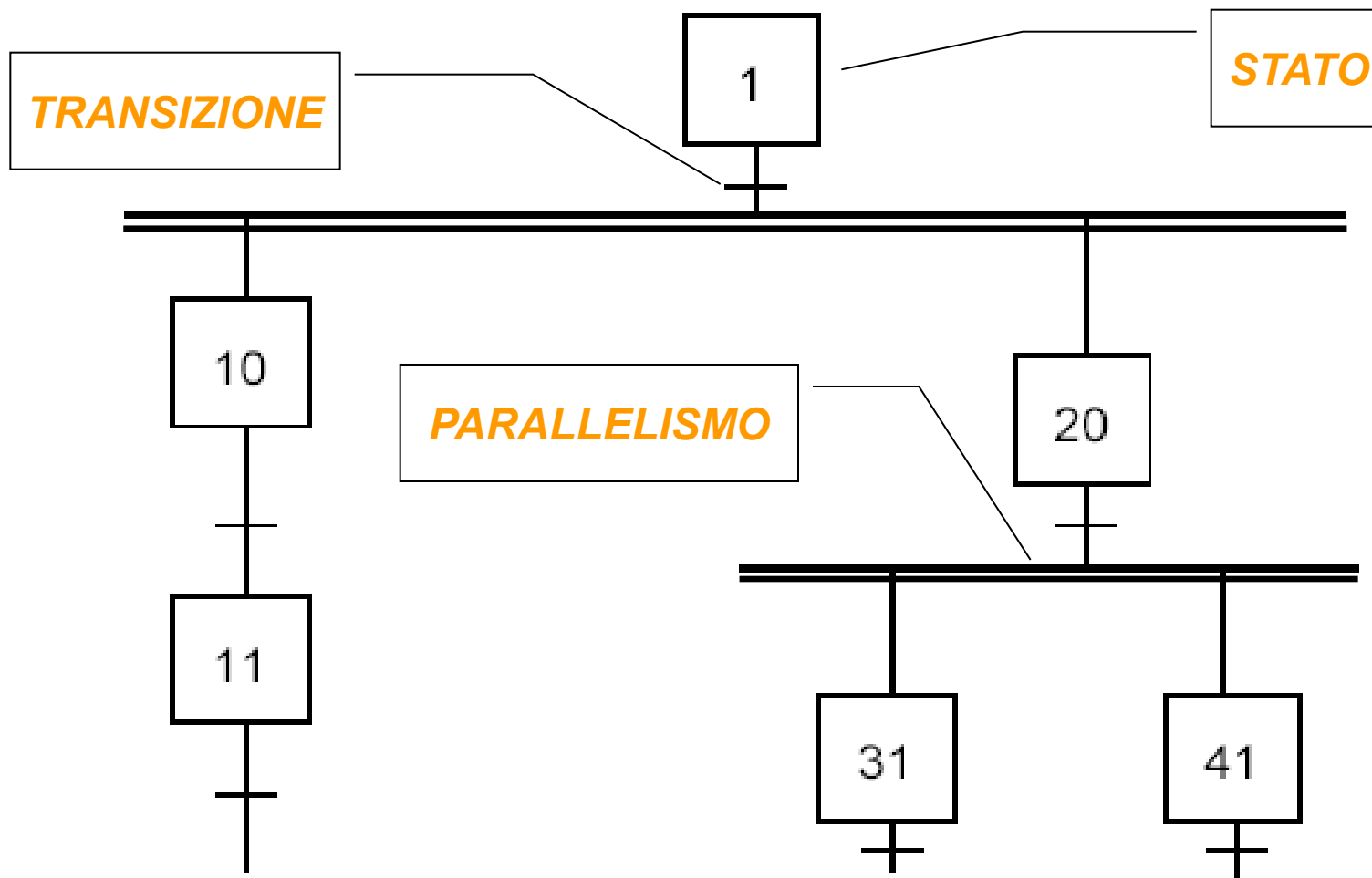
- Sequential Function Charts (SFC)
- Ladder Diagrams (LD)
- Function Block Diagrams (FBD)

**Grafici**

- Instruction List (IL)
- Structured Text (ST)

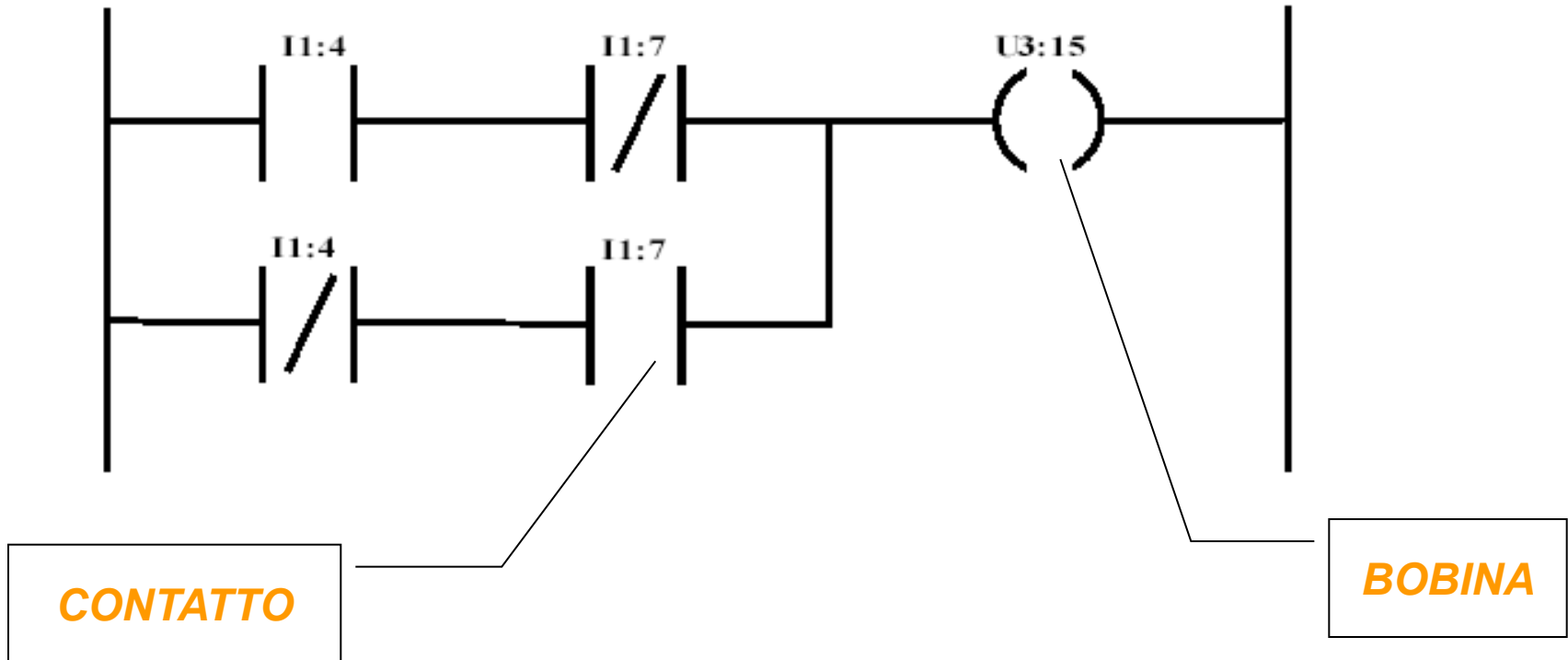
**Testuali**

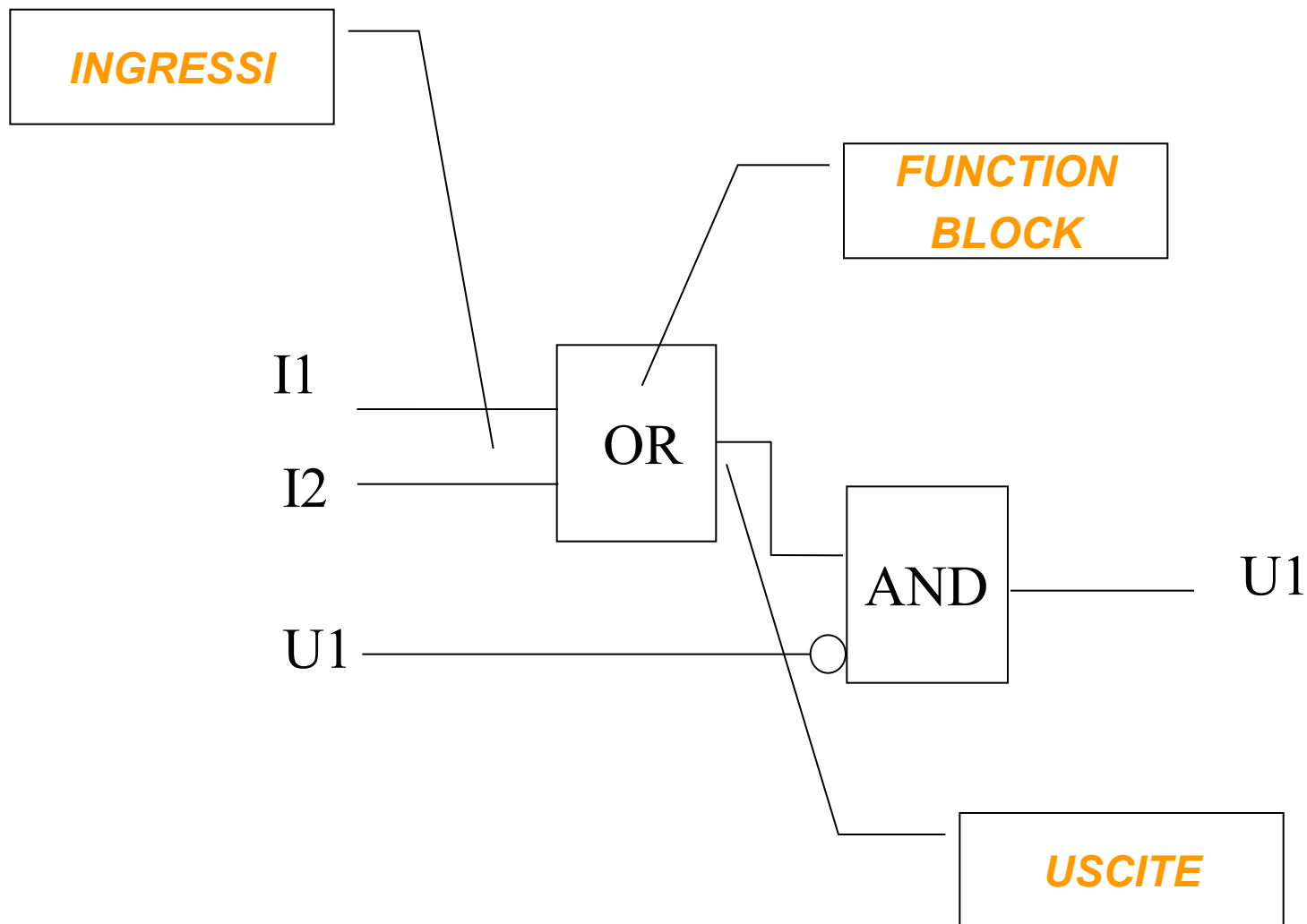
- Nota: il SFC è concepito come uno strumento di organizzazione interna di una POU, pur essendo a tutti gli effetti uno strumento formale ed eseguibile.

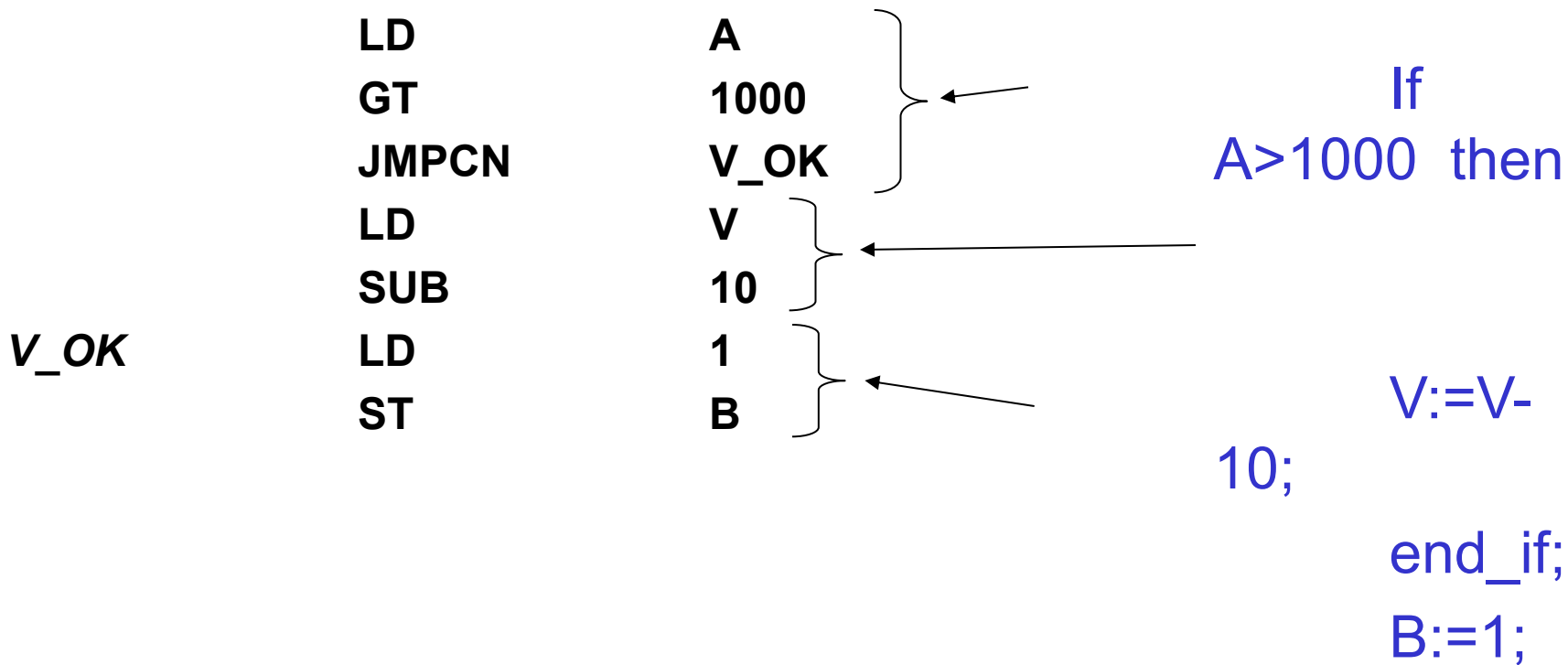




- ✓ Ogni piolo associa alle bobine una funziona booleana dei contatti
- ✓ Simile ai circuiti logici elettrici
- ✓  $U3:15 = F_{bool}(I1:4, I1:7) = (I1:4 \text{ AND } NOT(I1:7)) \text{ OR } (NOT(I1:4) \text{ AND } I1:7)$









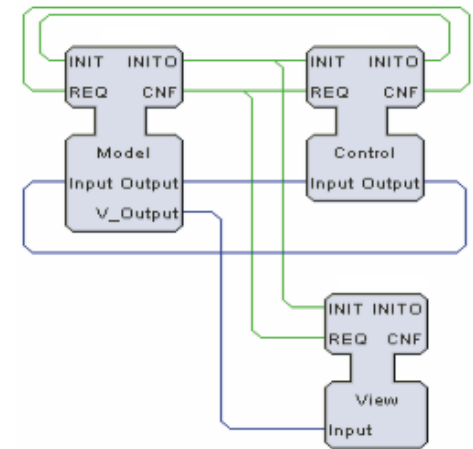
- ✓ Molto simile ai linguaggi di programmazione general purpose come il C...

```
IF A>B THEN
    D := 1;
ELSEIF A=B+2 THEN
    D:=2;
ELSE
    REPEAT
        D :=D+1;
    UNTIL (START=Off)
    END_REPEAT
END_IF;
```



## ✓ IEC-61131

- ✓ Definizione hardware dell'architettura del PLC
- ✓ Definizione di una serie di linguaggi per l'implementazione delle logiche di controllo
  - ✓ Grafici (SFC, LD, FBD)
  - ✓ Testuali (IL, ST)
  - ✓ Tipi di variabili, persistenza, ecc.
- ✓ Normativa datata
  - ✓ Prima edizione 1993
- ✓ Attualmente la più diffusa (forse l'unica come standard!)



## ✓ Evoluzioni

- ✓ IEC-61499 (Function Block NET)
- ✓ Soft PLC (es. Orchestra Control)

