



SAPIENZA
UNIVERSITÀ DI ROMA

Controllo logico sequenziale

Automazione

28/10/2015

Vincenzo Suraci



SAPIENZA
UNIVERSITÀ DI ROMA

Corso di Laurea: INGEGNERIA
Insegnamento: AUTOMAZIONE
Docente: DR. VINCENZO SURACI

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

STRUTTURA DEL NUCLEO TEMATICO

- INTRODUZIONE
- REALIZZAZIONE SU EMBEDDED SYSTEM
- REALIZZAZIONE SU PLC
- REALIZZAZIONE SU SOFT PLC



SAPIENZA
UNIVERSITÀ DI ROMA

Corso di Laurea: INGEGNERIA
Insegnamento: AUTOMAZIONE
Docente: DR. VINCENZO SURACI

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

INTRODUZIONE



CONTROLLO LOGICO-SEQUENZIALE

Abbiamo visto che un **sistema di controllo real time** deve agire a livello di campo, di coordinamento e di conduzione per eseguire tutti i **task necessari al corretto funzionamento** del sistema complesso.





CONTROLLO LOGICO-SEQUENZIALE

In **Automazione** i sistemi di controllo **HARD REAL-TIME CENTRALIZZATI** di **PICCOLA TAGLIA** possono essere realizzati tramite **EMBEDDED CONTROLLER** che eseguono opportuni programmi software sviluppati per il controllo logico sequenziale.





CONTROLLO LOGICO-SEQUENZIALE

In Automazione i sistemi di controllo **HARD/SOFT REAL-TIME CENTRALIZZATI** possono essere realizzati tramite **PLC** o **Soft PLC** che eseguono programmi software per il controllo logico sequenziale.





CONTROLLO LOGICO-SEQUENZIALE

In Automazione i sistemi di controllo **HARD/SOFT REAL-TIME DISTRIBUITI** operanti a livello di conduzione sono realizzati tramite **SCADA** (Supervisory Control and Data Acquisition).





CONTROLLO LOGICO-SEQUENZIALE

In Automazione i sistemi di controllo **HARD/SOFT REAL-TIME DISTRIBUITI** operanti a livello di campo, coordinamento e conduzione, possono anche essere realizzati tramite **DCS** (Distributed Control System).





SAPIENZA
UNIVERSITÀ DI ROMA

Corso di Laurea: INGEGNERIA
Insegnamento: AUTOMAZIONE
Docente: DR. VINCENZO SURACI

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

REALIZZAZIONE SU EMBEDDED SYSTEM



Sistemi di Controllo Embedded

DEFINIZIONE

Un qualsiasi sistema realizzato tramite **una singola scheda elettronica** oppure tramite **un singolo circuito integrato** viene chiamato **EMBEDDED SYSTEM**.

DEFINIZIONE

Un **sistema di controllo** realizzato tramite un **embedded system** viene chiamato **EMBEDDED CONTROLLER**.

OSSERVAZIONE

Un **sistema di controllo embedded** contiene al suo interno **tutto il necessario** sia per **connettere** il controllore al sistema da controllare, sia per **eseguire** gli algoritmi di controllo definiti dall'utente.

Un **sistema embedded** viene progettato o scelto in maniera tale che la configurazione **hardware** e **software** sia **ad-hoc** rispetto al problema di **Automazione** da risolvere. Ciò richiede la **conoscenza a priori dei compiti da eseguire**.



Sistemi di Controllo Embedded

VANTAGGI

Noto a priori il problema di automazione il sistema embedded permette di **ridurre l'hardware**, lo **spazio**, i **consumi energetici**, i **tempi di realizzazione** ed il **costo** necessari a realizzare il sistema di controllo.

SVANTAGGI

Di contro, un sistema embedded è caratterizzato da **scarsa flessibilità**, **bassissima estendibilità** e **difficile intercambiabilità**.

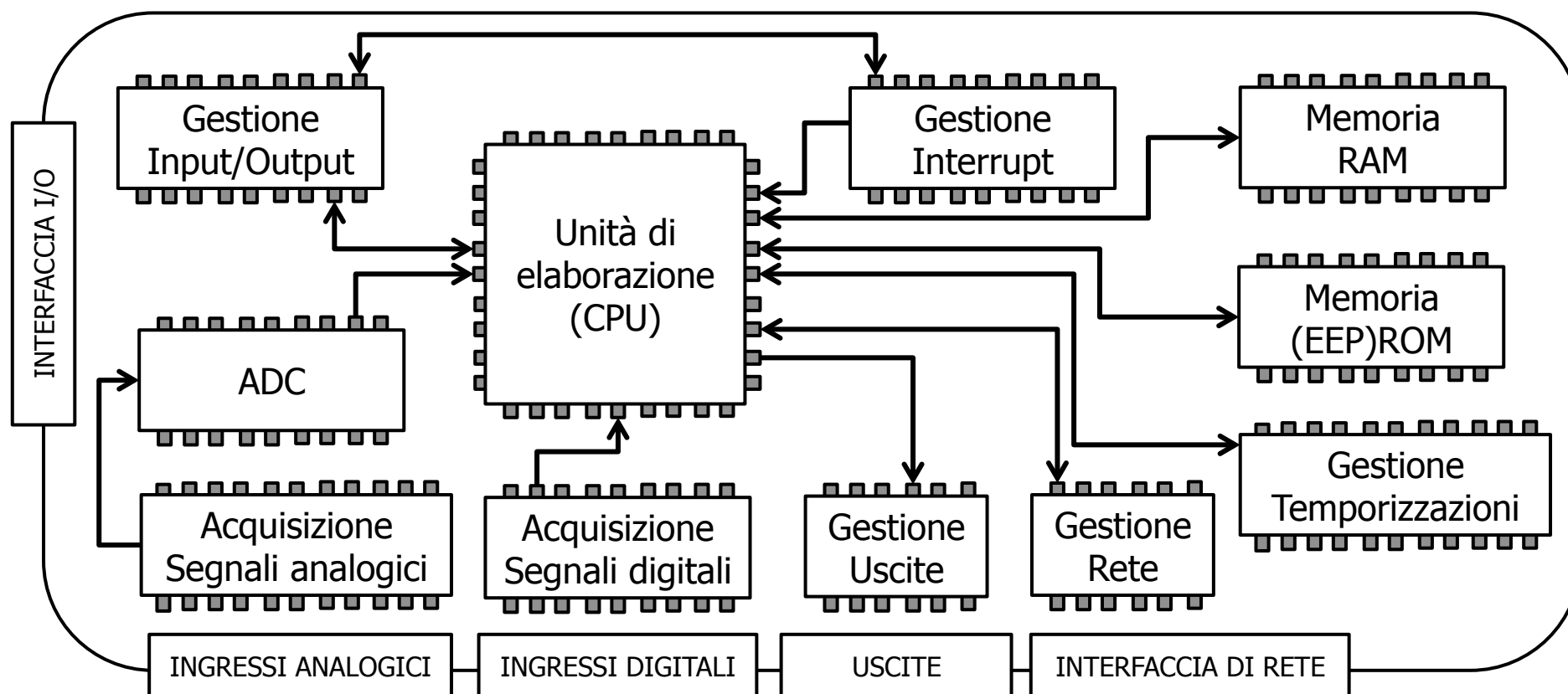
OSSERVAZIONE

L'uso dei sistemi embedded nell'ambito della **Automazione** si è diffuso moltissimo soprattutto a **livello di campo**, dove i **task** del sistema di controllo sono ripetuti e **noti a priori**.



Architettura Hardware dei Controllori Embedded

Un controllore embedded è realizzato su un'unica scheda, come in figura:

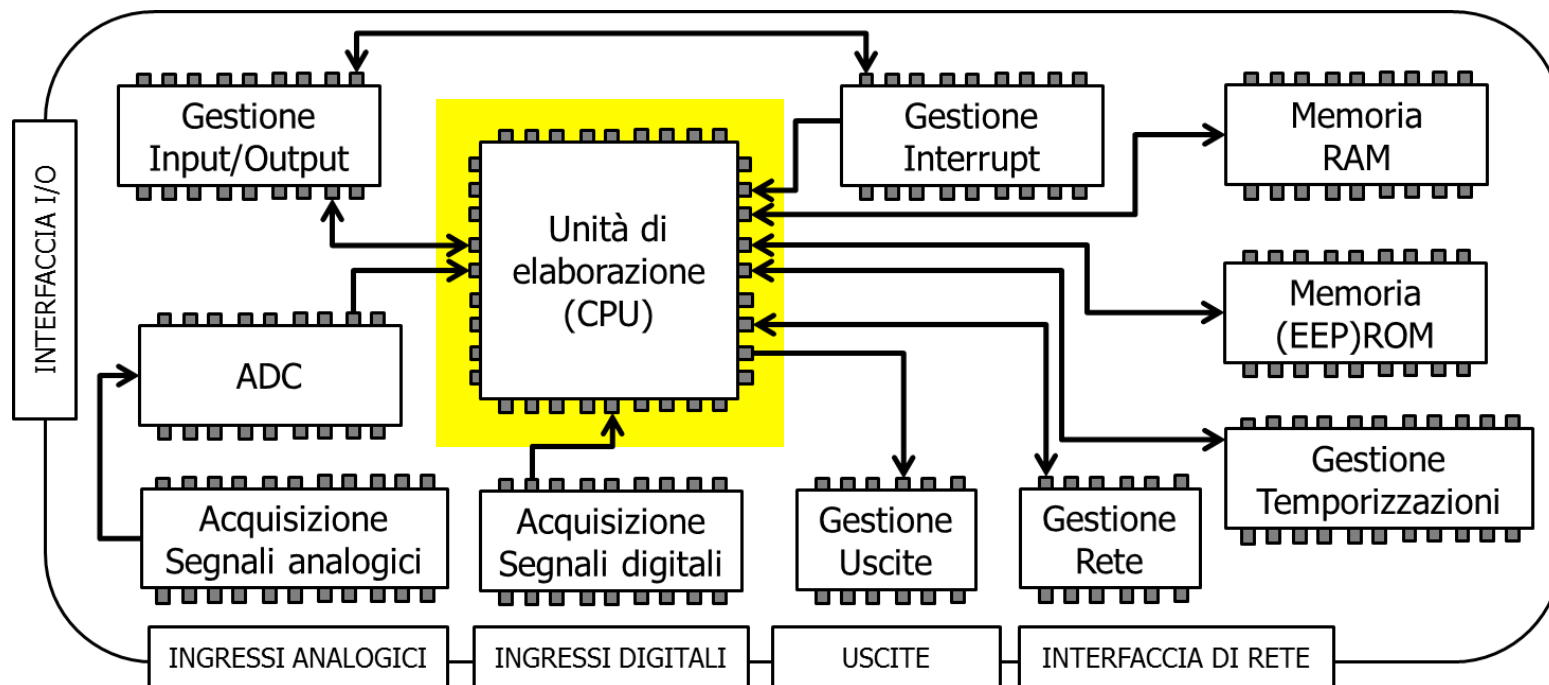




Architettura Hardware dei Controllori Embedded

CENTRAL PROCESSING UNIT (CPU)

È il **processore** che si occupa di **eseguire** sia il **Basic I/O System (BIOS)**, ovvero il **sistema operativo**, che il **programma realizzato dall'utente** che implementa l'**algoritmo di controllo**.

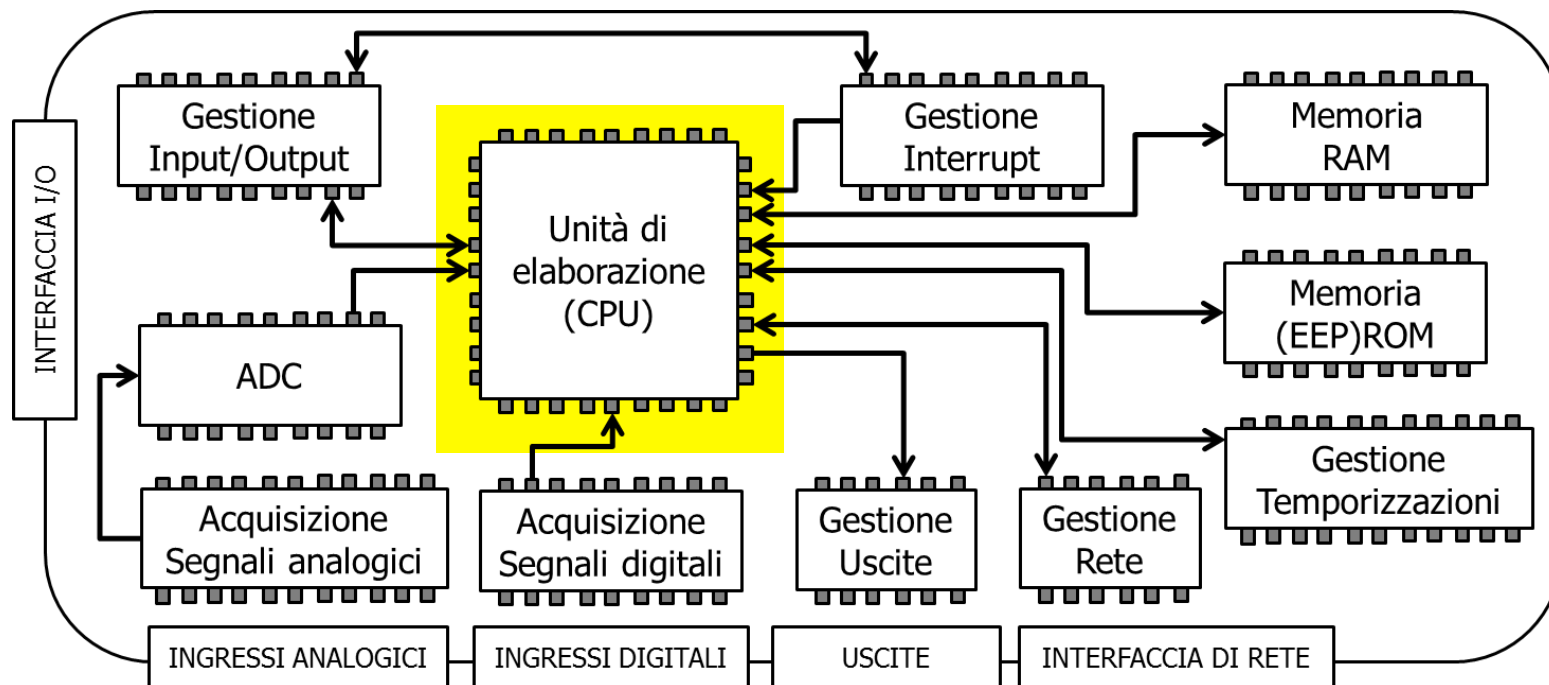




Architettura Hardware dei Controllori Embedded

CENTRAL PROCESSING UNIT (CPU)

La **CPU** può essere un **microprocessore general purpose** (*numeri interi*), oppure un **digital signal processor** (*numeri interi e reali*), o un **field programmable gate array** (*funzioni logiche*)

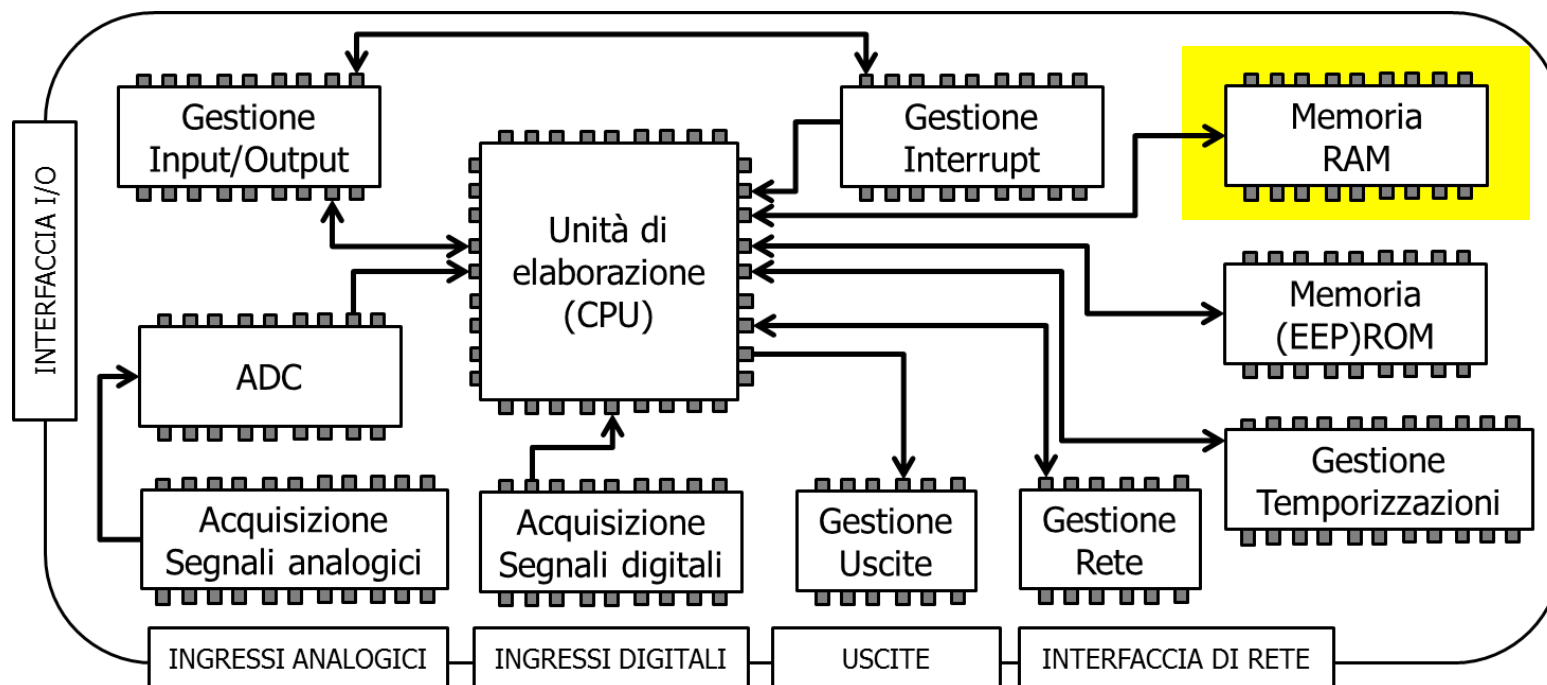




Architettura Hardware dei Controllori Embedded

MEMORIA RAM

È la **memoria volatile** necessaria per mantenere lo **stato del programma**, ovvero tutti i **dati temporanei** necessari durante l'elaborazione degli **algoritmi di controllo** logico sequenziale.

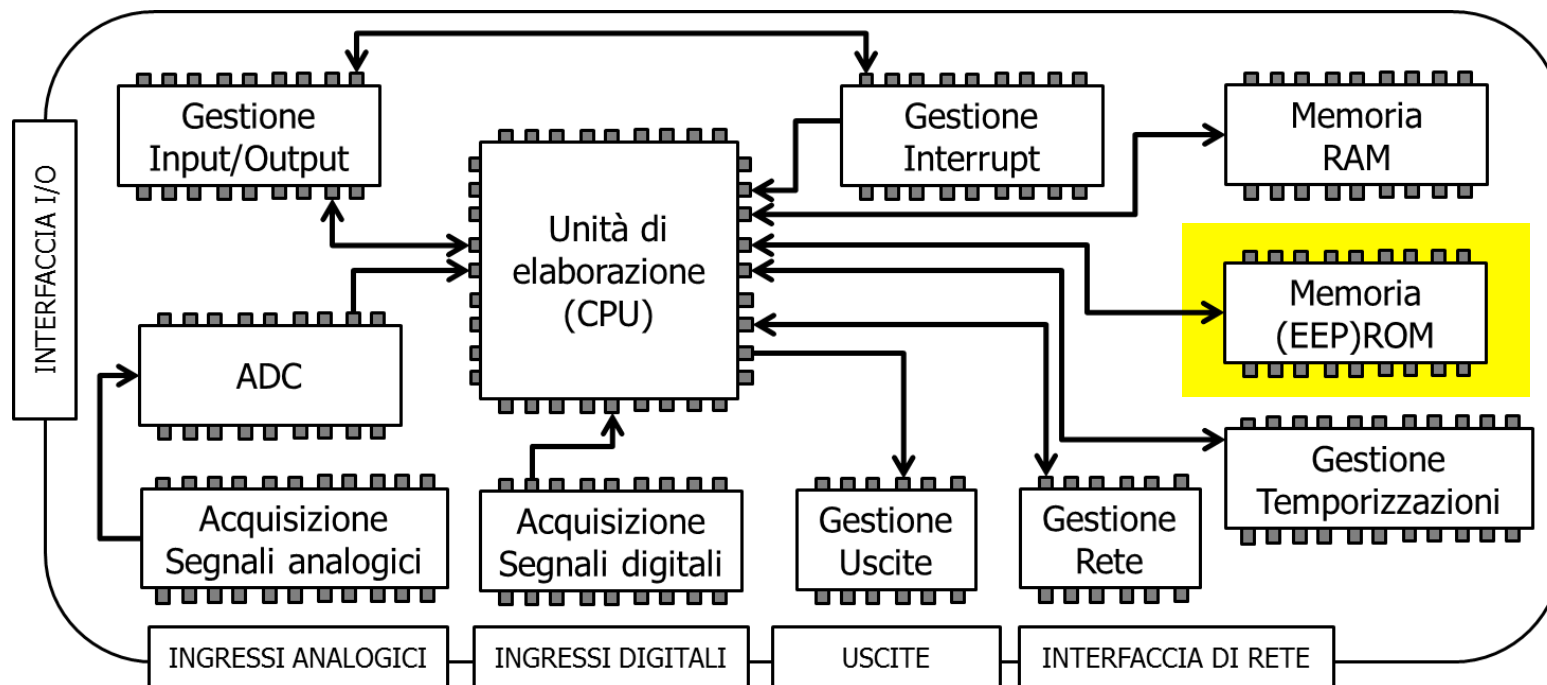




Architettura Hardware dei Controllori Embedded

MEMORIA (EEP)ROM

È la **memoria non volatile** necessaria per **memorizzare permanentemente** il **sistema operativo (ROM)** e il **programma utente ([EEP]ROM)**.

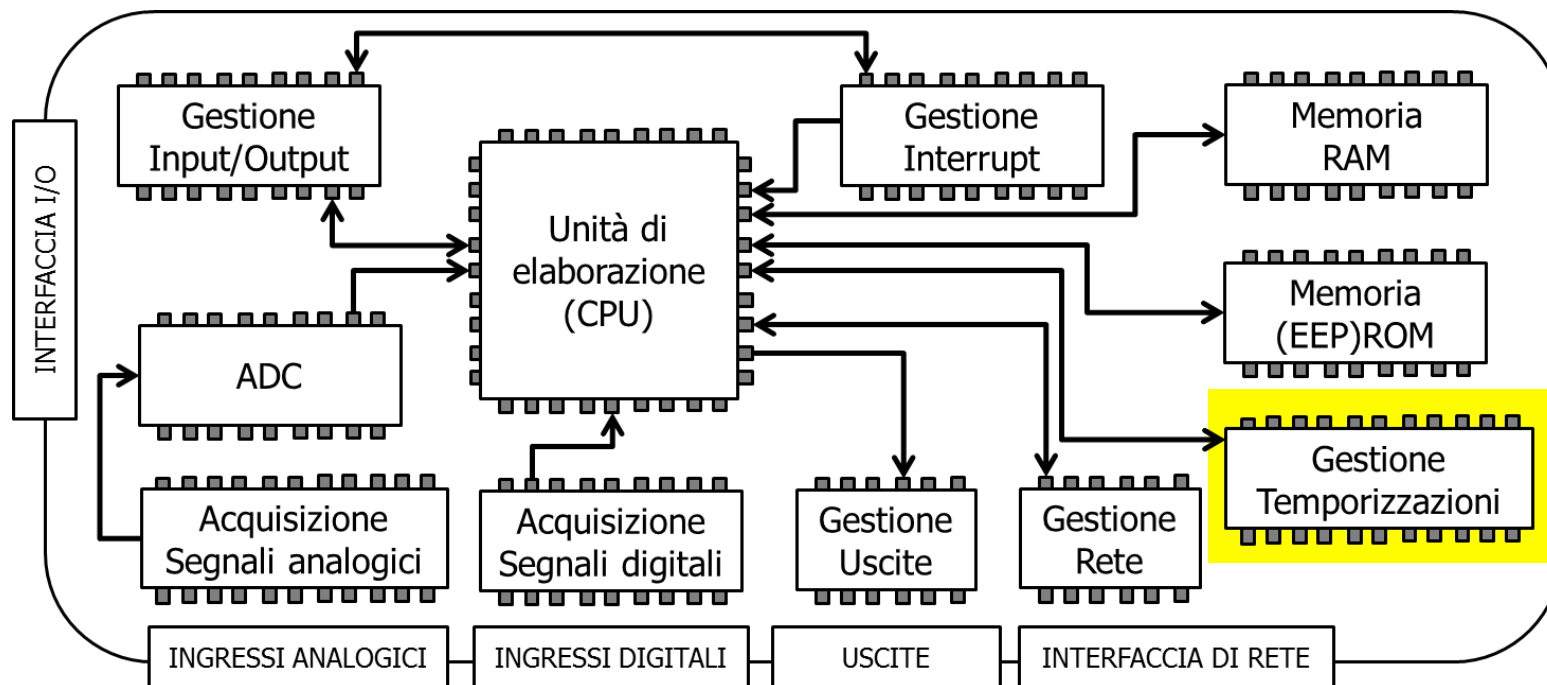




Architettura Hardware dei Controllori Embedded

GESTIONE TEMPORIZZAZIONI

Gestione dei **timer** utili per la **sincronizzazione** e la **temporizzazione** delle attività della CPU.

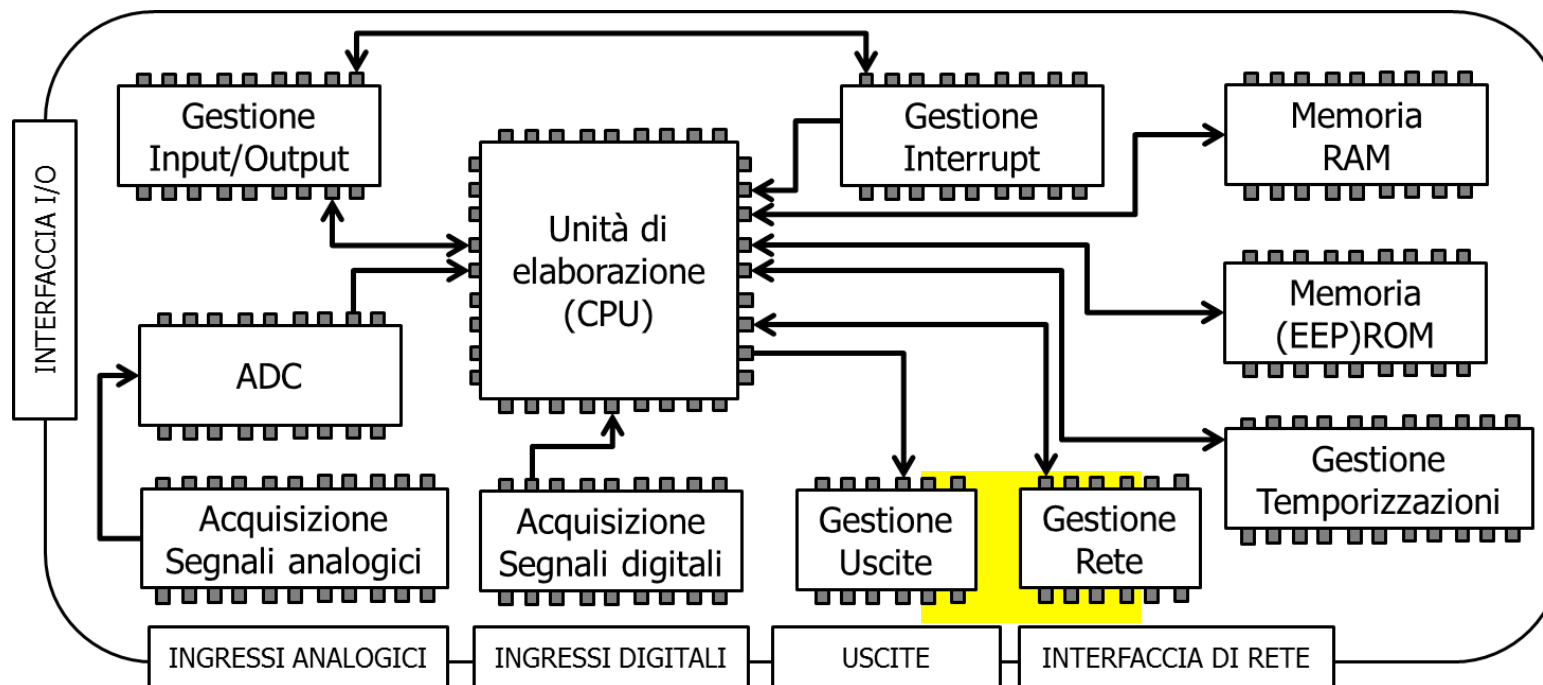




Architettura Hardware dei Controllori Embedded

GESTIONE RETE

Contiene i **protocolli di base** (di livello **fisico** e di **accesso al mezzo** trasmissivo) per **gestire la comunicazione con altri dispositivi di controllo** (embedded e non).

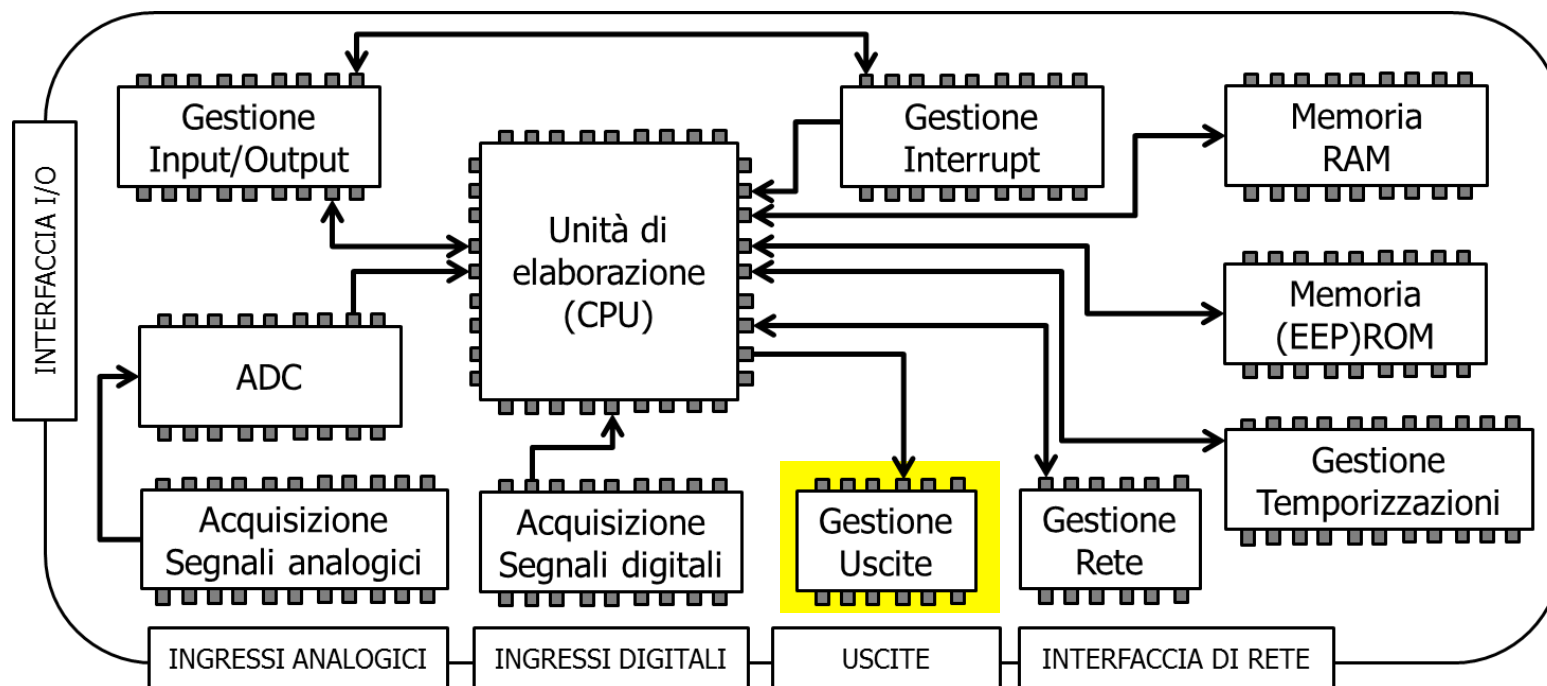




Architettura Hardware dei Controllori Embedded

GESTIONE USCITE

Contiene una serie di circuiti per la **generazione di segnali analogici** (DAC) e **digitali**.

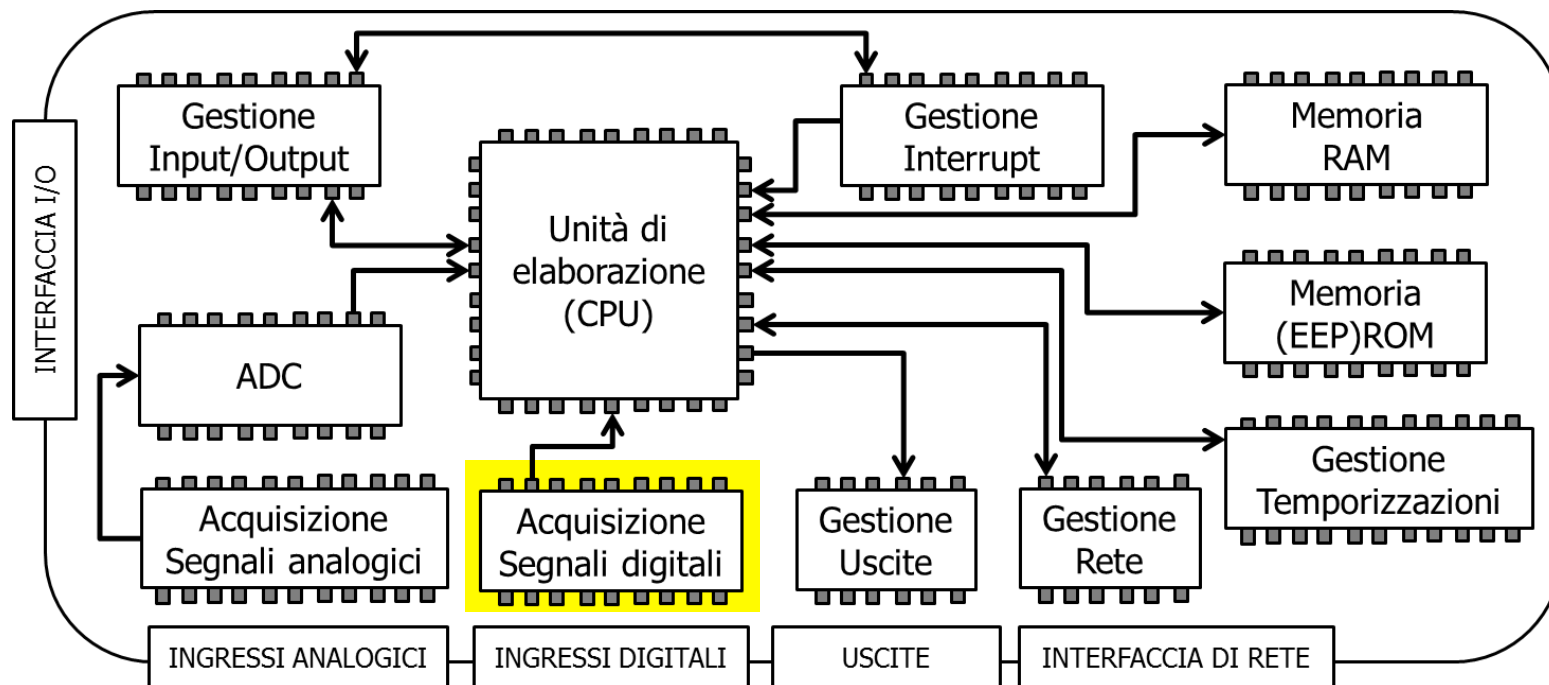




Architettura Hardware dei Controllori Embedded

ACQUISIZIONE SEGNALI DIGITALI

Circuiti per l'accoppiamento con i segnali digitali in ingresso (ad es. TTL, OPTO)

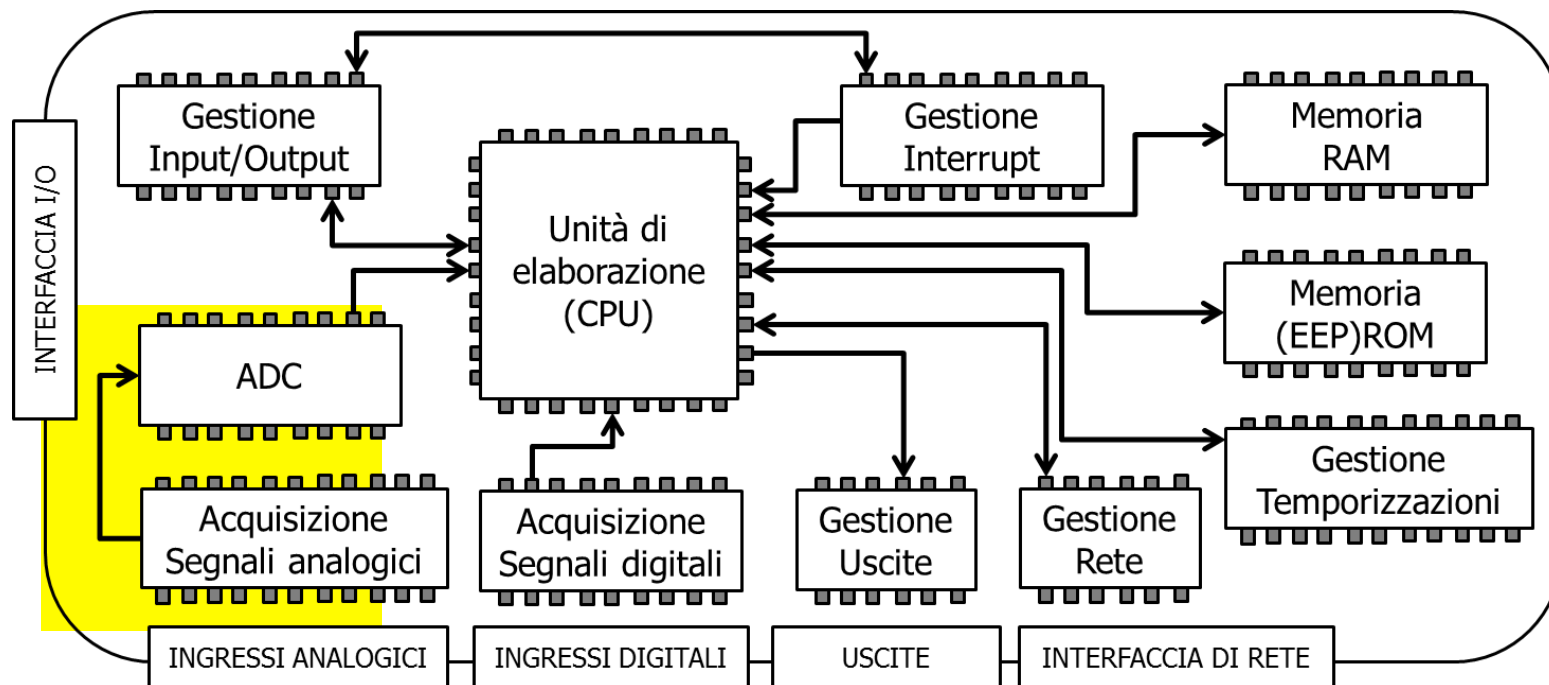




Architettura Hardware dei Controllori Embedded

ACQUISIZIONE SEGNALI ANALOGICI + ADC

Circuiti per l'**accoppiamento con i segnali analogici in ingresso**, l'eventuale **multiplexer** e il **convertitore A/D (ADC)** per il **campionamento** e la **quantizzazione**.

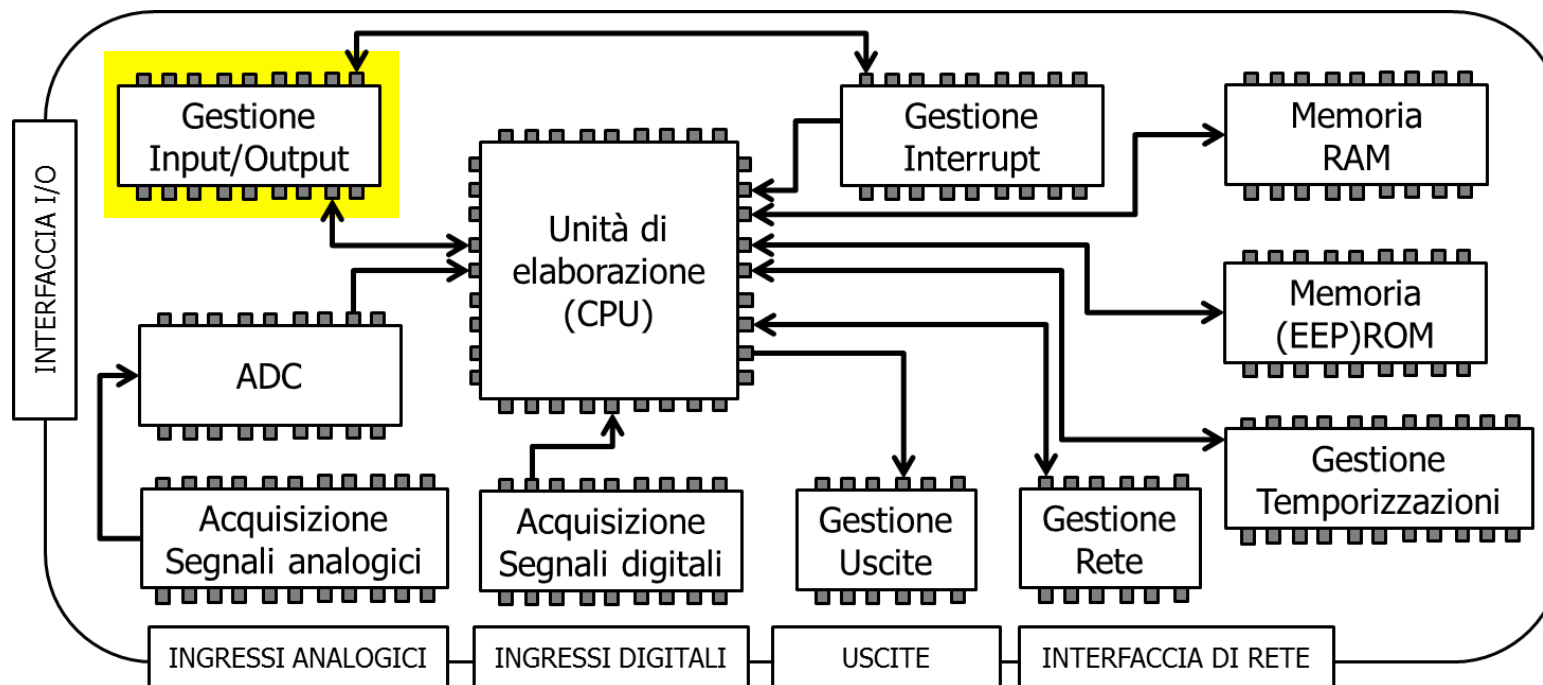




Architettura Hardware dei Controllori Embedded

GESTIONE I/O

Circuiti per l'accoppiamento con bus dedicati o schede di espansione.

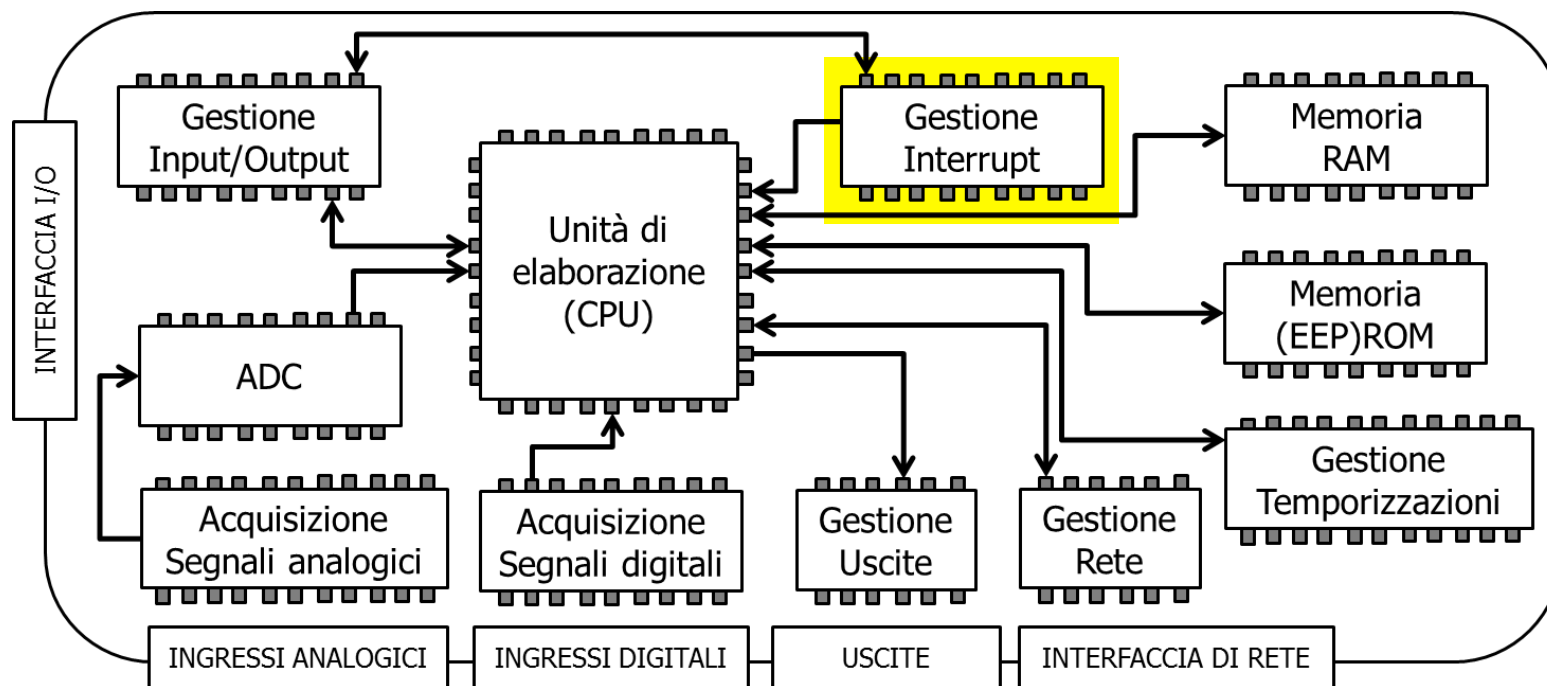




Architettura Hardware dei Controllori Embedded

GESTIONE INTERRUPT

Circuiti per la **rilevazione degli eventi**.





Microcontrollori

DEFINIZIONE

Un **embedded controller** realizzato tramite **un singolo circuito integrato** viene chiamato **microcontrollore**.

OSSERVAZIONE

Ha senso prendere in considerazione l'utilizzo di microcontrollori esclusivamente in quei sistemi di **Automazione** che richiedano:

- notevole **riduzione degli ingombri** (cellulari, elettrodomestici, servizi di rete, centraline elettroniche, wearable devices, etc.)
- **numero limitato di segnali di ingresso / uscita** (digitali o analogici)
- **basso consumo energetico** (alimentazione a batteria)
- **HMI minimale** (ad es. tastiera, display LCD, LED) o nulla
- **interoperabilità e integrazione** con altri dispositivi **limitata o nulla**

Scenari applicativi appetibili per i μ **controllori** sono: **Building Automation, Smart Grid, Smart Cities, Wireless Sensor and Actuator Networks, Mobilità, Health.**



SAPIENZA
UNIVERSITÀ DI ROMA

Corso di Laurea: INGEGNERIA
Insegnamento: AUTOMAZIONE
Docente: DR. VINCENZO SURACI

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

REALIZZAZIONE SU PLC



Controllori a BUS

Quando si deve realizzare un sistema di controllo logico sequenziale caratterizzato da:

- **complessità** di calcolo degli algoritmi molto elevata
- **elevato numero** di ingressi e/o uscite analogiche/digitali
- **HMI** complesse ed articolate
- **interoperabilità** con altri sistemi di controllo ed informativi

si preferisce sostituire i **controllori embedded** con i **CONTROLLORI A BUS**.

DEFINIZIONE

Un **BUS** è un insieme di **linee di trasporto di energia e di informazione** (in generale di **tipo elettrico**) che permettono la **comunicazione tra più dispositivi**.

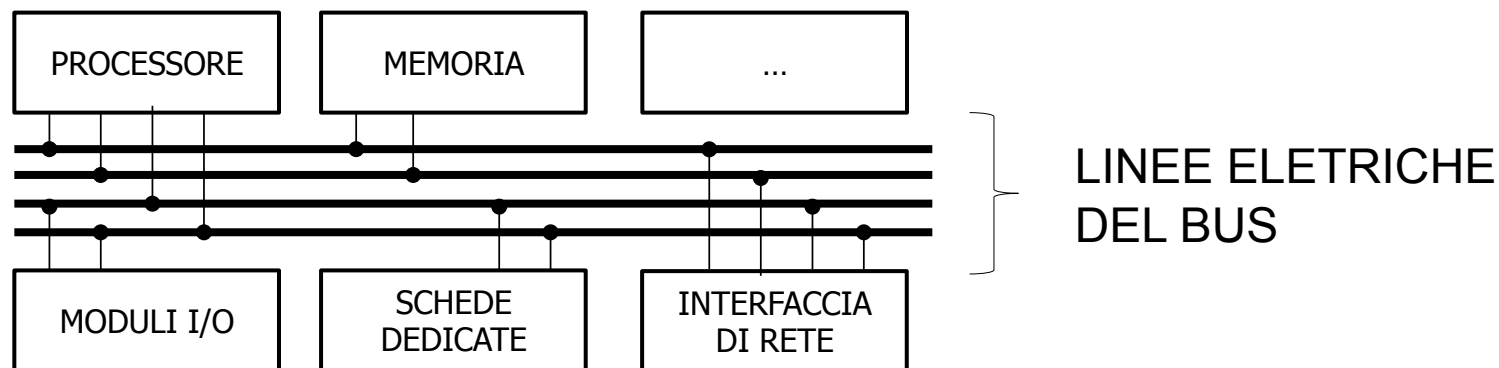
OSSERVAZIONE

Per **identificare un BUS** è pertanto necessario fissare il **numero di linee**, definire le **funzionalità offerte** da ciascuna linea, i **protocolli di comunicazione** usati dai dispositivi interconnessi e le **interfacce meccaniche**.



Architettura a BUS di un sistema di controllo

In una **architettura a bus**, ad un modulo principale ospitante il **processore**, vengono **connessi tutti gli altri moduli** necessari a comporre il controllore (memoria, moduli I/O, periferiche di HMI, interfacce di rete, schede dedicate, etc.).



Le **linee elettriche** del BUS vengono raggruppate e differenziate in base alle **funzioni**:

- **linee di INDIRIZZO**
- **linee DATI**
- **linee di ALIMENTAZIONE**
- **linee di CONTROLLO**



Caratteristiche dei BUS

- **massimo numero di unità di elaborazione** – un bus multi-processore è dotato di un protocollo di comunicazione in grado di gestire l'accesso concorrente a più CPU
- **velocità massima di trasmissione dati**
- **massimo numero di moduli interconnettibili**
- **dimensione massima della memoria indirizzabile**
- **sincrono** – basato su clock (time-driven)
- **asincrono** – basato su interrupt requests IRQ (event-driven)



Esempi di BUS general purpose

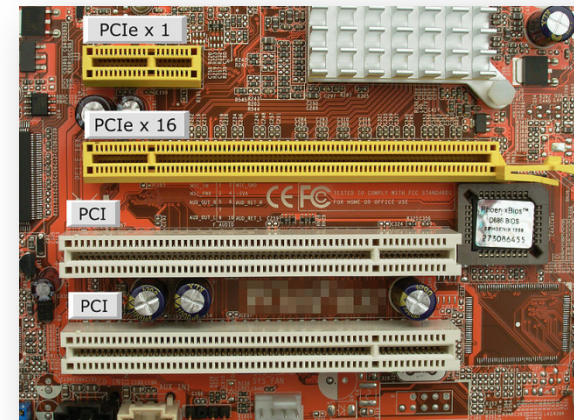
ISA (Industry Standard Architecture)

- Ampiezza BUS: 8/16 bit
- Frequenza operativa: 4.77/6/8 Mhz
- Velocità massima: 4/5 Mbit/sec



PCI (Peripheral Component Interconnect)

- Ampiezza BUS: 32 bit
- Frequenza operativa: 33.33 MHz
- Velocità massima: 1064 Mbit/s
- Memoria Indirizzabile: 32 bit (4 Gigabytes)



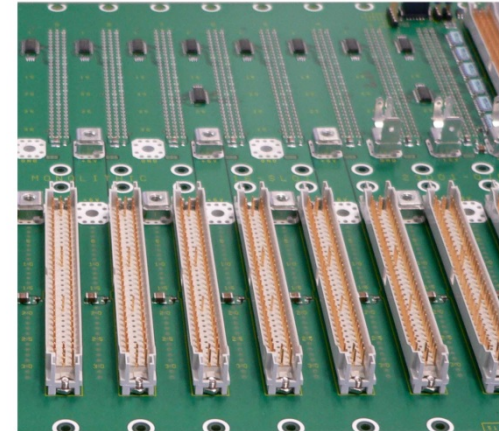


Esempi di BUS industriali

VME/VME64x (VERSABUS Module Eurocard)

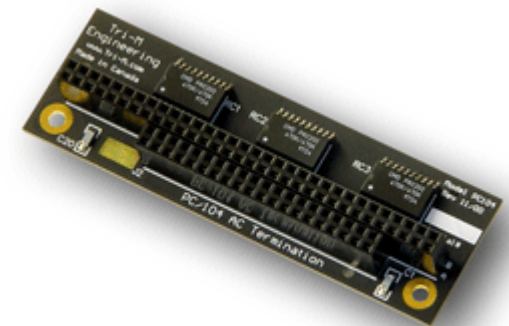
- Ampiezza BUS: 32 bit
- Velocità massima: 40 Mbit/s
- Memoria Indirizzabile: 64 bit

VME / VME64x Backplanes



PC104 & PC104+

- Ampiezza BUS: 8-16-32 bit
- Velocità massima: 40 Mbit/s





Pro e contro della architettura a BUS

VANTAGGI

- **flessibilità** della **progettazione** del sistema di controllo in termini di **componenti** e **funzionalità**
- **semplicità** della **programmazione** della logica di controllo spesso supportata da **tool**, **librerie** e ampie **community** di sviluppatori

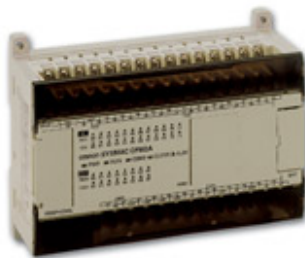
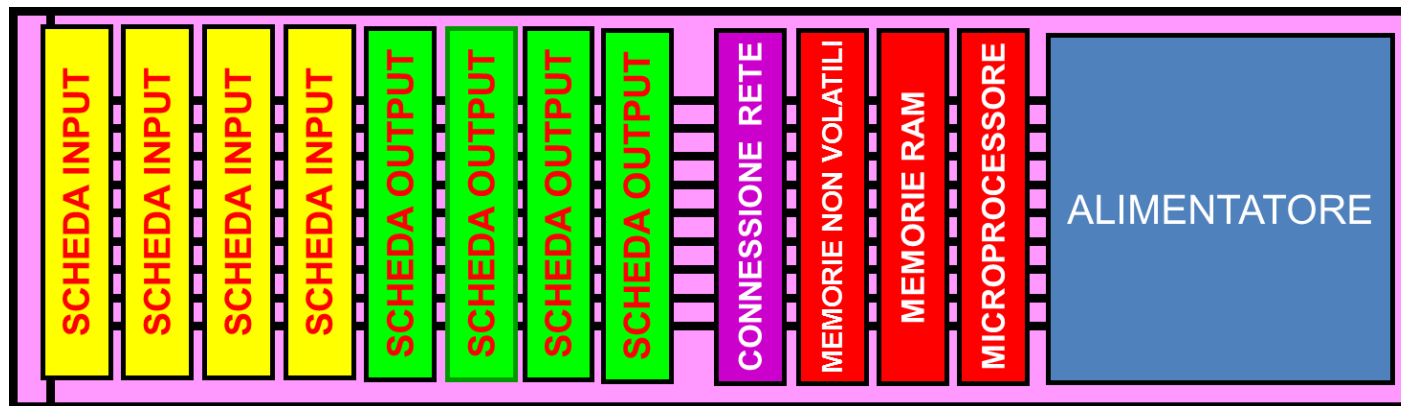
SVANTAGGI

- **complessità** della **progettazione** del sistema di controllo, in particolare nella **scelta** di ogni singolo **componente**
- **complessità** del **sistema operativo** (comunque trasparente all'utente finale)



Programmable Logic Controller (PLC)

I **controllori a logica programmabile** sono controllori basati su **architettura a bus** e hanno riscosso ampio successo in ambito della **automazione industriale**.



μPLC



PLC MEDI

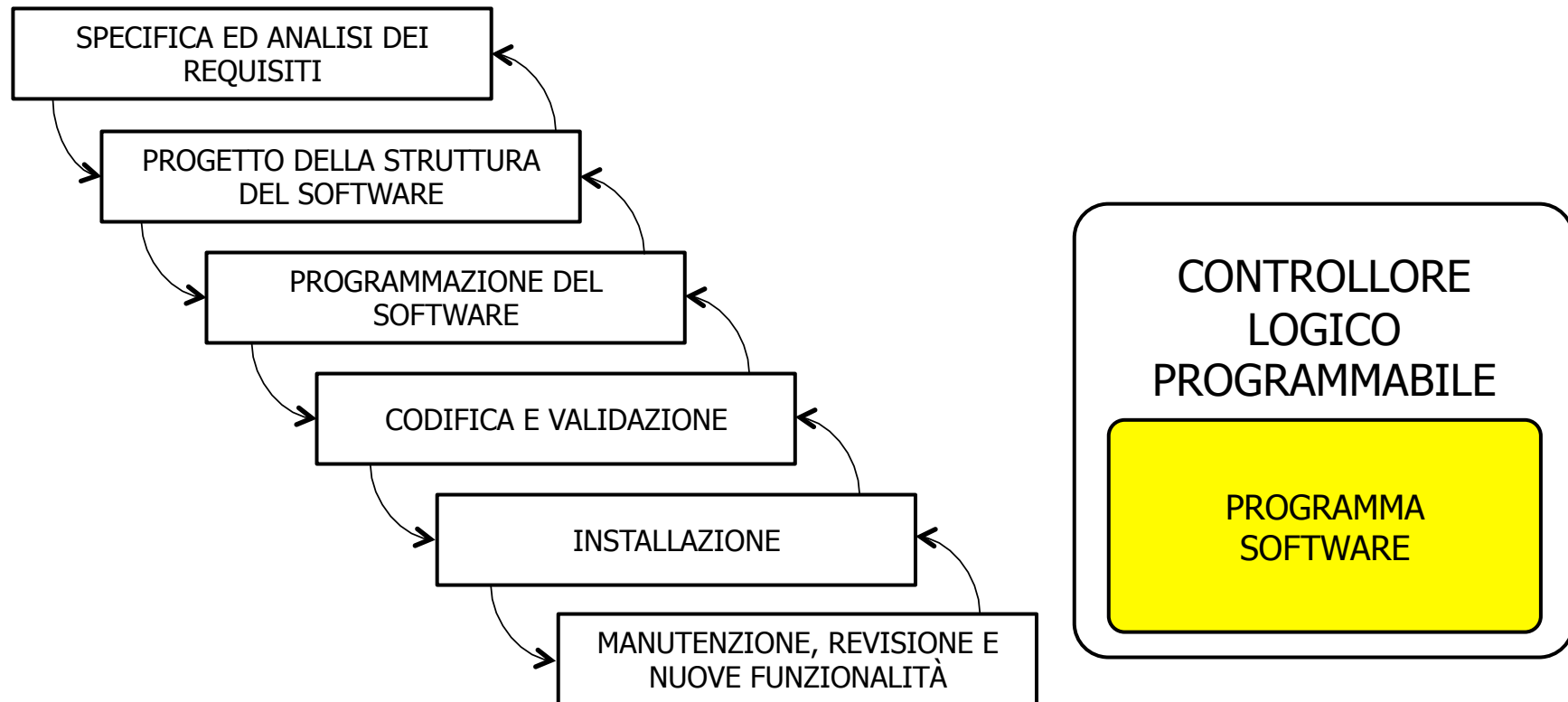


PLC GRANDI



CONTROLLO LOGICO-SEQUENZIALE nei PLC

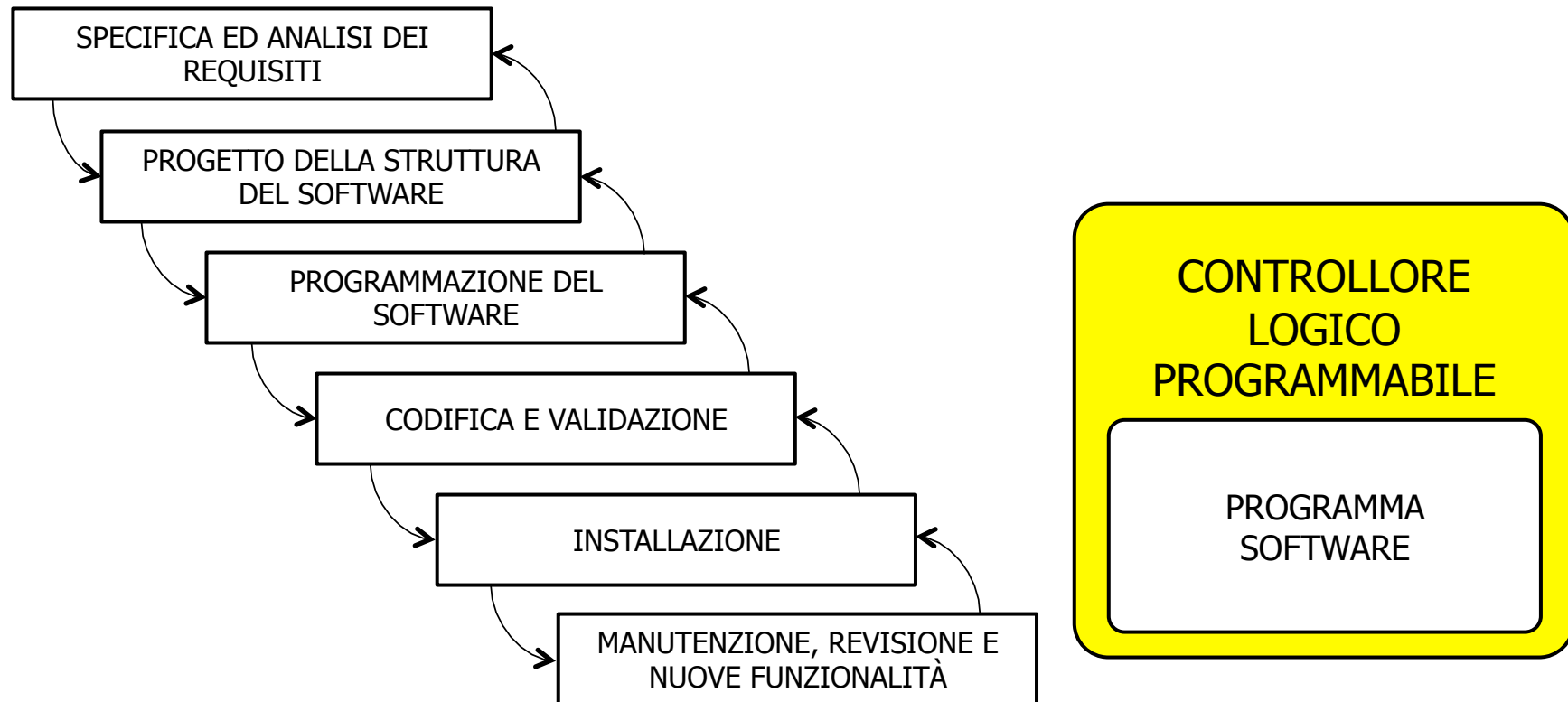
Il **software** finalizzato al **controllo logico sequenziale** segue un **ciclo di vita** ben consolidato.





CONTROLLO LOGICO-SEQUENZIALE nei PLC

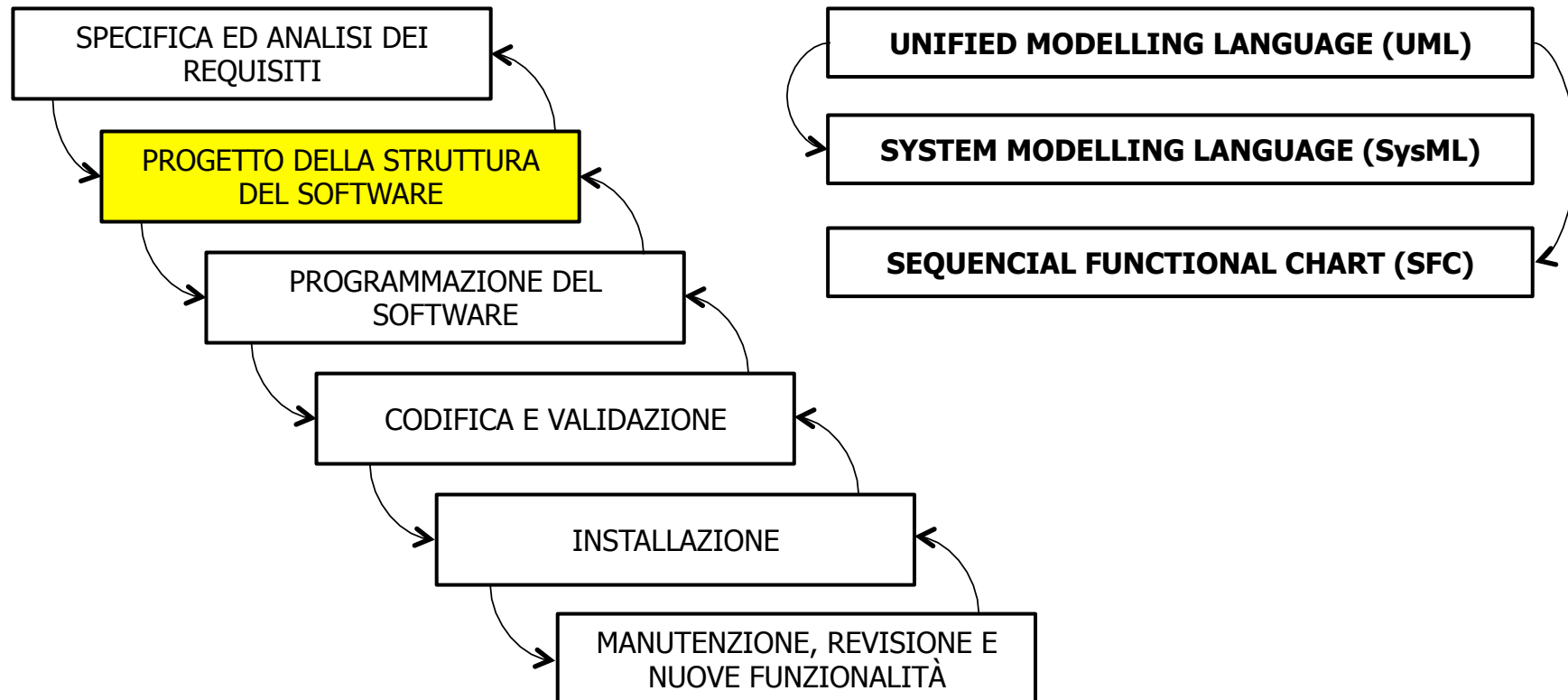
Ogni produttore di **PLC** ha sviluppato **tool proprietari** a supporto del ciclo di vita del software. Per superare **ambiguità** sono stati definiti specifici **linguaggi** per la **progettazione** e la **programmazione** del **software**.





CONTROLLO LOGICO-SEQUENZIALE nei PLC

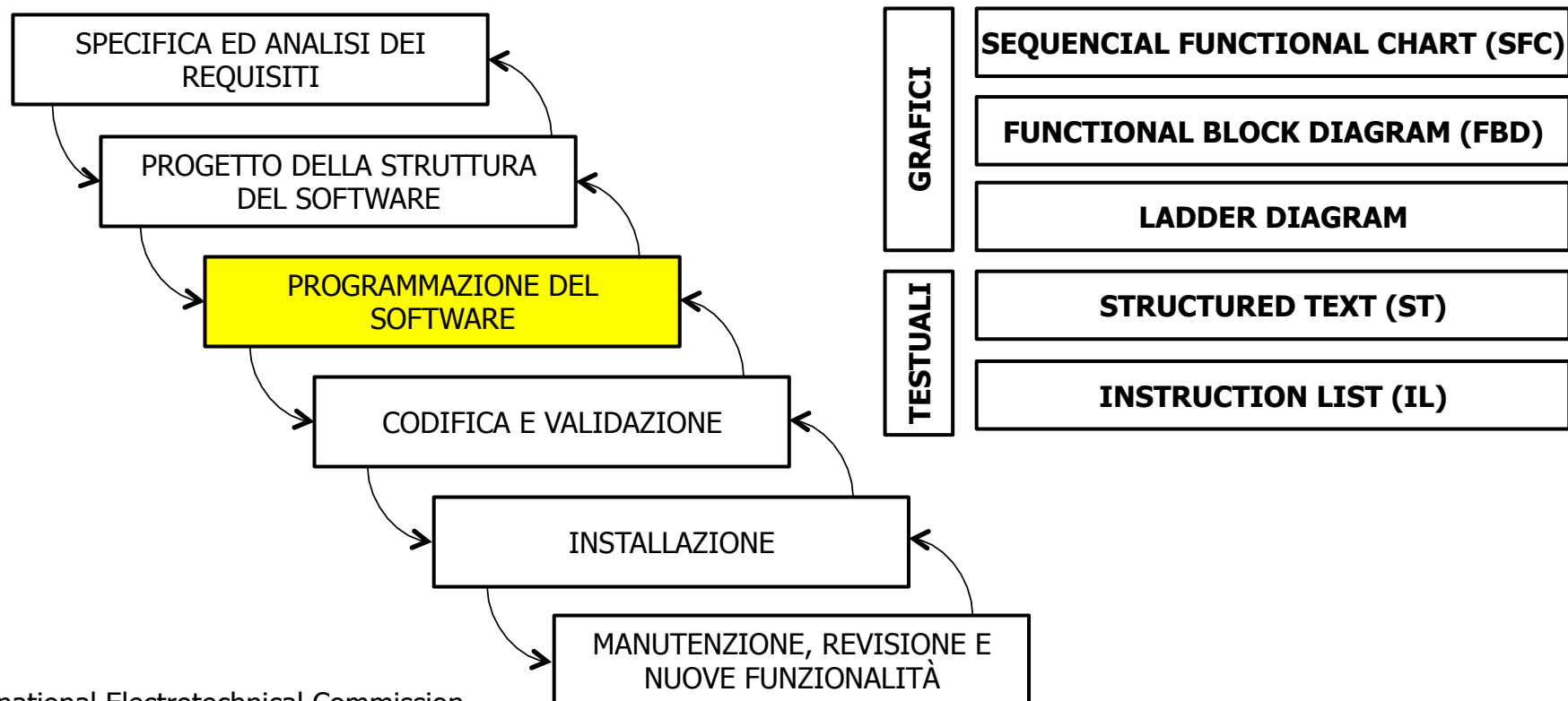
La **progettazione** del **software** finalizzato al **controllo logico sequenziale** ha mutuato dall'ingegneria del software alcuni **linguaggi formali**.





CONTROLLO LOGICO-SEQUENZIALE nei PLC

La **programmazione** del **software** finalizzato al **controllo logico sequenziale** ha visto consolidarsi l'uso di **linguaggi** definiti nella norma **IEC* 61131-3**.



* International Electrotechnical Commission



SAPIENZA
UNIVERSITÀ DI ROMA

Corso di Laurea: INGEGNERIA
Insegnamento: AUTOMAZIONE
Docente: DR. VINCENZO SURACI

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

REALIZZAZIONE SU SOFT PLC



PC Industriali

OSSERVAZIONE

L'evoluzione del settore informatico e la **diffusione capillare** dell'uso dei **personal computer** per uso generico ha comportato:

- **abbattimento dei costi**
- ampia **disponibilità di hardware**
- ampia **disponibilità di software** (a volte **open source**)
- **molteplicità dei vendor**
- **funzionalità avanzate** (HMI, SDK, OS, networking)
- **know-how diffuso e consolidato**

IDEA

Perché non **adattare i PC** ad essere utilizzati in un **ambiente industriale** (condizioni ambientali estreme, forti disturbi elettromagnetici, vibrazioni, particolato, etc.) realizzando pertanto il concetto di **PC INDUSTRIALI**?



Soft PLC

OSSERVAZIONI

- I **PLC** sono ormai **diffusissimi** nel settore della automazione ed in particolare nel settore dell'**automazione industriale**
- I produttori di PLC aggiornano continuamente i cataloghi, rendendo **disponibili moduli nuovi e più avanzati** che sono **retro-compatibili**
- **Nessun imprenditore rivoluziona il proprio sistema di automazione** cambiando i buoni e vecchi PLC con PC Industriali **senza una motivazione economica o strategica** ben precisa

IDEA

- Perché non adattare i **PC industriali** ad **emulare via software i PLC**, ovvero perché non adottare una soluzione basata su **SOFT PLC**?



Pro e Contro dei Soft PLC

VANTAGGI

- I **Soft PLC** ereditano tutti i vantaggi dei **PLC** e dell'uso dei **PC**

SVANTAGGI

- I Soft PLC devono essere **INSTALLATI** su **PC INDUSTRIALI**
- I Soft PLC richiedono l'uso di particolari schede in grado di moltiplicare il numero di **ingressi** ed **uscite** utilizzando i BUS tipici dei PC (PCI, ISA)
- I Soft PLC richiedono una **modifica al SO** per renderlo **REAL TIME** (le versioni desktop di Linux, Windows XP, Vista, 7 non sono nativamente real time)



BIBLIOGRAFIA

Sezione 1.3



TITOLO

**Sistemi di automazione industriale
Architetture e controllo**

AUTORI

Claudio Bonivento
Luca Gentili
Andrea Paoli

EDITORE

McGraw-Hill