

MICROCONTROLLORI

Introduzione

I processori per PC costituiscono meno del 20% del mercato mondiale dei microprocessori

I processori per applicazioni embedded costituiscono la restante parte del mercato

Il microcontrollore è un dispositivo che opera come un sistema embedded

Un sistema embedded è un sistema di elaborazione specializzato integrato in un dispositivo fisico in modo da controllarne le funzioni tramite un apposito programma software dedicato

Characteristics of Embedded Systems/I

- Must be **efficient**
 - Energy efficient
 - Code-size efficient (especially for systems on a chip)
 - Run-time efficient
 - Weight efficient
 - Cost efficient
- **Dedicated** towards a certain **application**

Knowledge about behavior at design time can be used to minimize resources and to maximize robustness
- Frequently **connected to physical environment** through sensors and actuators and **hybrid systems** (analog + digital parts).

Characteristics of Embedded Systems/II

- **Reactive** and **real-time**
 - Continually reacts to changes in the system's environment
 - Must compute certain results in real-time without delay
- Single-functioned
 - Executes a single program, repeatedly
- "... > 70% of the development cost for complex systems such as automotive electronics and communication systems are due to software development"

Cosa sono i microcontrollori

Il microcontrollore è un dispositivo che raggruppa un processore ed un insieme di dispositivi periferici su un unico chip:

CPU (Core)

Memoria RAM

Memoria EPROM, EEPROM, FLASH, ROM od OTP

Porte I/O

Timers e contatori

Porte di comunicazione seriale standard (UART e USART) e porte di comunicazione seriale speciali (Bus SPI, Bus I2C, USB, CAN Bus 1-Wire)

Convertitori A/D e D/A

Amplificatori operazionali e comparatori

Ha prestazioni più basse rispetto ai microprocessori general purpose

In generale non necessita di RAM ulteriore oltre quella integrata

Sono disponibili molti pin di I/O

A cosa servono

Applicazioni di controllo basate su acquisizione ed elaborazione di segnali

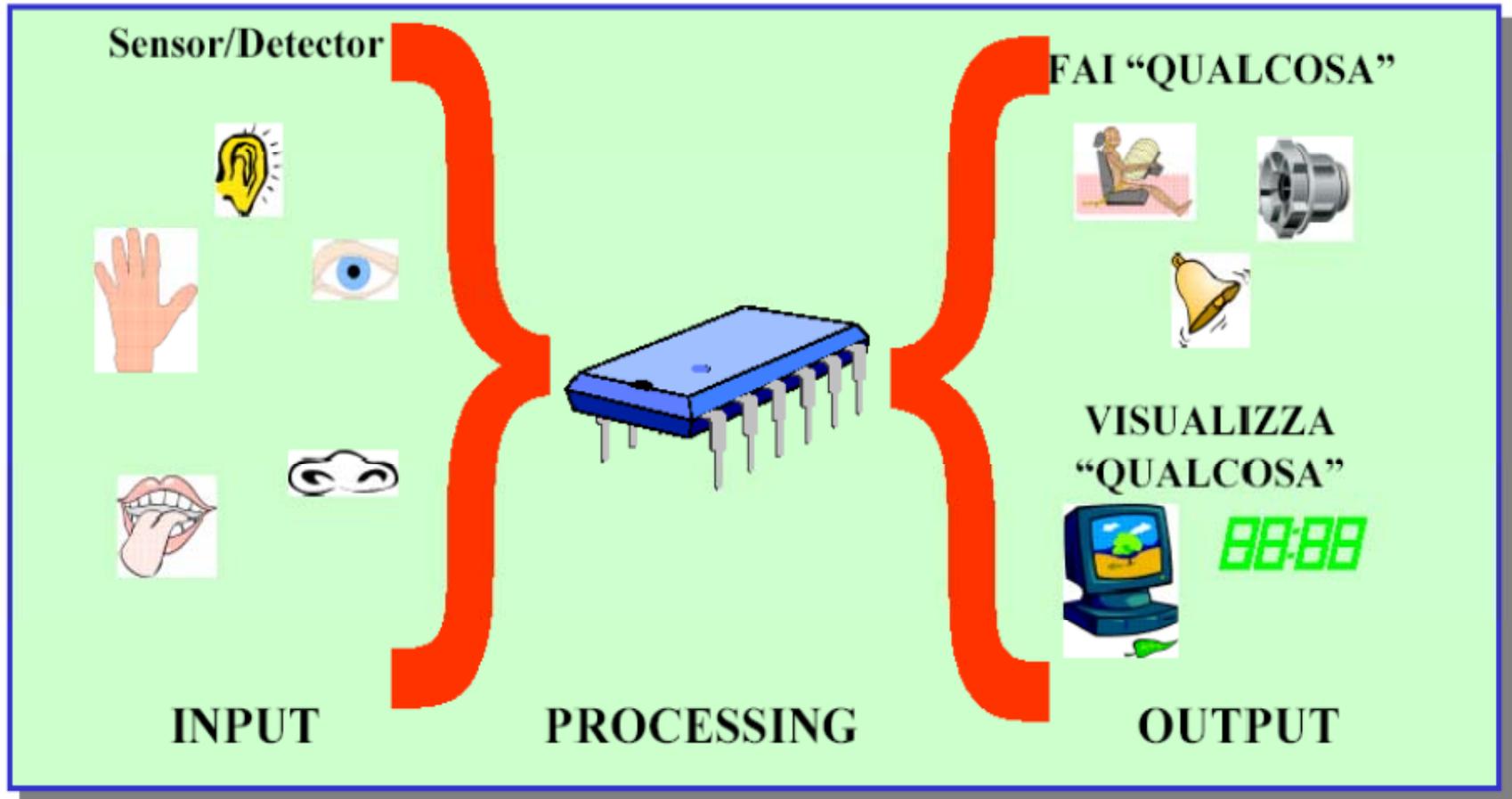
Tali applicazioni richiedevano design ad-hoc di componenti ed hardware dedicato

Oggi è possibile realizzare tali applicazioni ricorrendo ad un unico tipo di microcontrollore, cambiando solo gli aspetti software

Latest top-level BMWs contain over 100 microprocessors

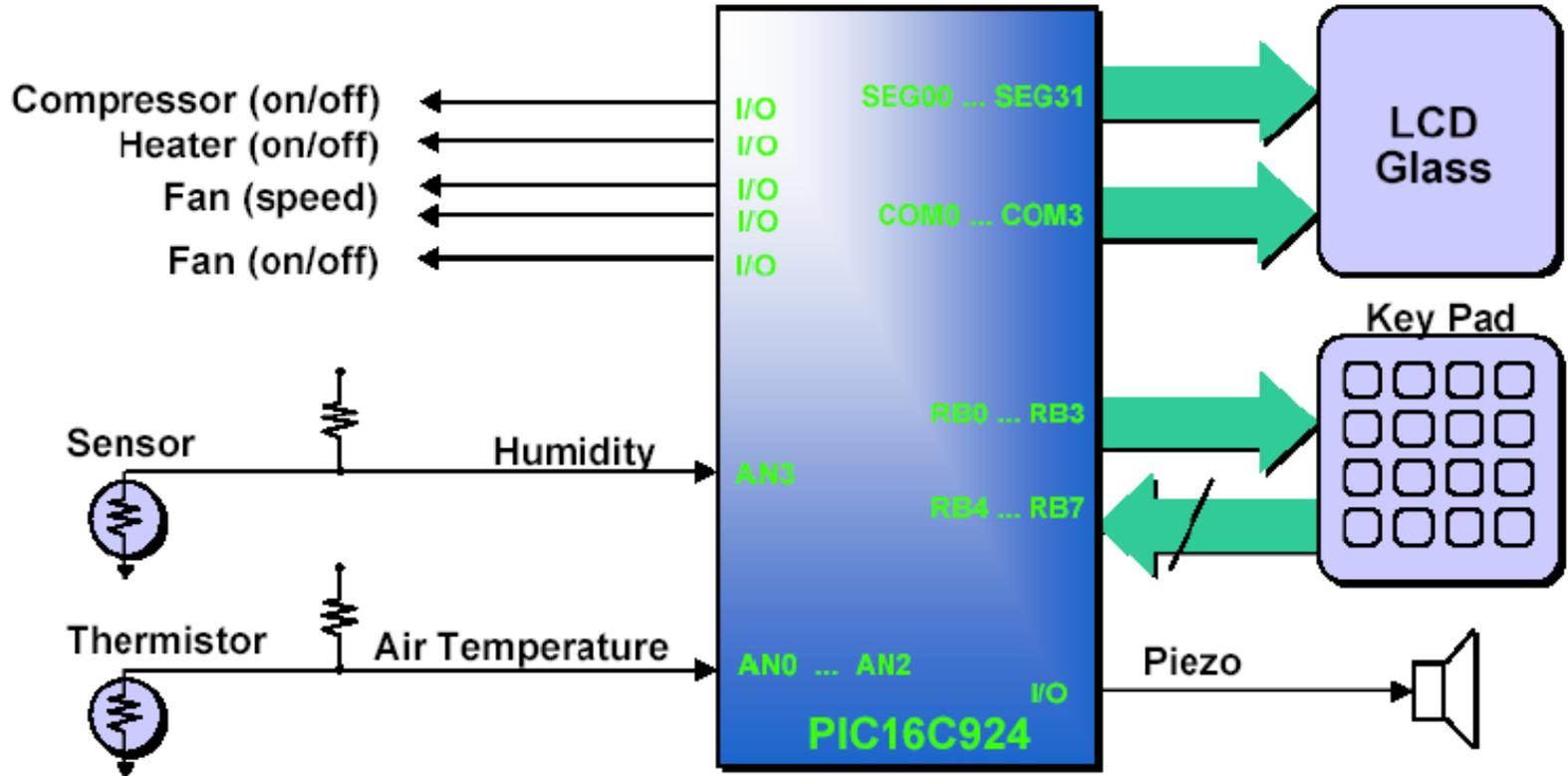
"... the New York Times has estimated that the average American comes into contact with about 60 micro-processors every day...."

Il funzionamento in generale



Esempio di applicazione

Controllo di temperatura/umidità in una carrozza di un treno



Vantaggi “single-chip”

Sono richiesti meno dispositivi “discreti” per la realizzazione di un sistema

Il sistema riesce ad avere dimensioni ridotte

Si abbassano i costi

Componenti: questi dispositivi costano qualche euro

Progettazione e produzione: componenti già integrati

Time-to-market (tempo richiesto per lo sviluppo di un sistema fino alla sua vendita: molto ridotto)

Si abbassa il consumo di potenza

Device on-chip hanno consumo minore di insiemi di device esterni

Si abbassa la sensibilità ad interferenze EM

Il sistema finale risulta più affidabile

Sono interconnessi meno componenti

Inoltre

Sistema flessibile in grado di

eseguire operazioni complesse (programmazione)

comunicare con altri dispositivi (standard di interconnessione)

Riconvertibilità dei progetti (programmazione)

Protezione contro la copiatura

La maggior parte dei single chip offre la possibilità di proteggere da lettura il programma contenuto in ROM

Risparmio energetico

Le versioni C-MOS supportano il modo di funzionamento stand-by: è possibile bloccare, via software, l'attività della CPU e quindi ottenere correnti di alimentazione molto basse.

Esempi applicativi

Prodotti per l'informazione personale

Telefoni cellulari, palmari, orologi, registratori, calcolatrici, videocamere, televisori, Hi-Fi

Componenti Laptop

Mouse, tastiere, modem, fax, schede sonore, caricatori di batterie

Applicazioni Home

Serrature per porte, sistemi di allarme, termostati, condizionatori, telecomandi, VCR, frigoriferi, lavatrici, forni a microonde, macchine per caffè

Automotive

Sistemi di bordo più (ABS, ESP, GPS, airbag, iniezione del carburante, controllo trazione, climatizzazione, computer di bordo) o meno (luci di cortesia, chiusura centralizzata, alzacristalli, autoradio, sistema di sbrinatorio e antiappannamento) critici

Settore industriale

Controllo di assi (posizione, velocità), regolatori (ON-OFF, PID, Fuzzy), robotica

Elettronica di consumo

Gadget, giocattoli

Growing economical importance of embedded systems/I

The growing market according to forecasts, e.g.:

Worldwide mobile phone sales surpassed 156.4 mln units in 2004, a **35%** increase from 2003

The worldwide portable flash player market exploded in 2003 and is expected to grow **from 12.5 mln** units in 2003 to over **50 mln units** in 2008

Global 3G subscribers will grow from an estimated **45 mln** at the end of 2004 to **85 mln in 2005**, according to Wireless World Forum.

Today's DVR (digital video recorders) users - 5% of households - will grow to **41% within five years**

Growing economical importance of embedded systems/II

The automotive sector ensures the employment of more than 4 million people in Europe. Altogether, some **8 million jobs** in total depend on the fortunes of the transport industry and related sectors

.. but embedded chips form the backbone of the electronics driven world in which we live ... they are part of almost everything that runs on electricity

79% of all high-end processors are used in embedded systems

The future is embedded, embedded is the future!

Processo produttivo

Sono generalmente realizzati con un processo HCMOS, che consente al clock di essere variato da DC fino alla massima frequenza senza controindicazioni

Ciò non è vero per i microprocessori general purpose, per i quali è richiesto un clock minimo

Ciò consente di modulare il consumo di potenza alle esigenze di prestazione dell'applicazione

La potenza dinamica è legata al clock di sistema $P = K * V_{ALIM}^2 * F_{CLK}$

Classificazione generale

Hardware:

Processore

Frequenza di clock [1] MHz a [1] GHz

Bit di parola: 4, 8, 16, 32

Set di istruzioni: repertorio e tipologia RISC e CISC

Supporto floating point o fixed point

Architettura di Von Neumann o Harvard (riguarda il tipo di collegamento degli elementi interni al processore)

Cache

Gestione interrupt

Periferiche on-chip

Numero e tipologia di periferiche disponibili e loro flessibilità

Classificazione generale

Hardware:

Connessioni esterne

Interfacce di comunicazione standard (seriali, bus di campo, ethernet, . . .)

Porte digitali generiche di ingresso / uscita

Porte analogiche

Bus di espansione per altri dispositivi (memorie, periferiche, altri controllori)

Consumo

Requisiti ambientali

Temperatura, umidità, vibrazioni, rumore EM

Compatibilità EM

Standard rispettati, varie modalità di funzionamento

Classificazione generale

Software:

Processore

Set di istruzioni

Linguaggi di programmazione disponibili (tipicamente C, Basic, Java, Assembler solo per ottimizzazioni di prestazioni e per qualche aspetto nella programmazione delle periferiche interne e di driver per dispositivi esterni)

Librerie di programmazione specifiche per l'architettura (calcoli su operandi a 64 bit con architettura a 8 bit) e per la sua gestione (es.: interrupt)

Requisiti real-time

Periferiche on-chip

Visione dello spazio di memoria e di I/O

Librerie per la programmazione delle interfacce

Connessioni esterne

Programmabilità tramite le interfacce

Consumo

Possibilità di gestire il consumo di potenza

Esecuzione di un'istruzione

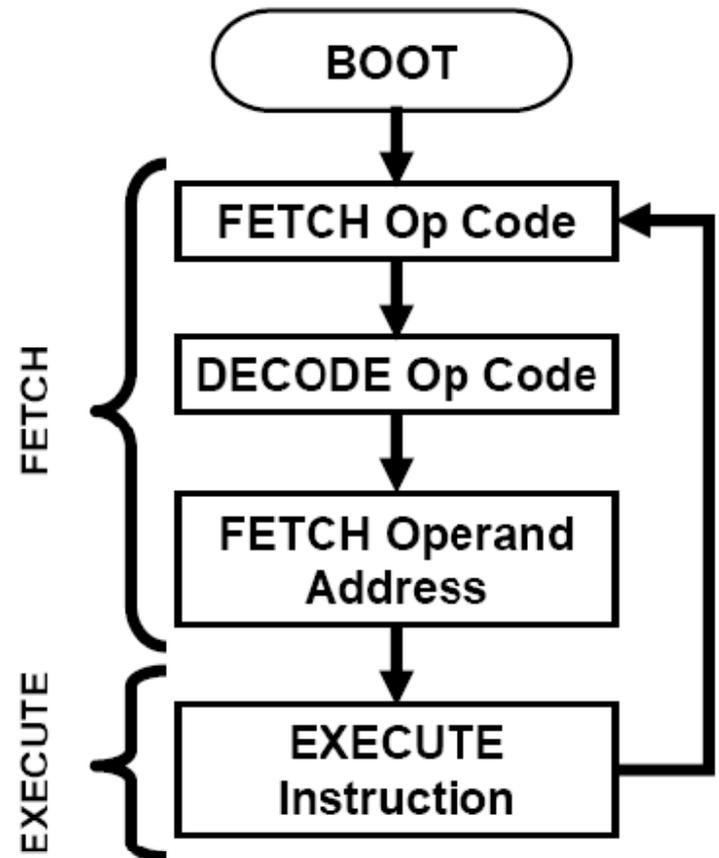
L'esecuzione di un'istruzione avviene in due fasi:

fase di fetch

La CPU carica dalla memoria il codice corrispondente all'istruzione da eseguire, lo interpreta e di conseguenza preleva dalla memoria i dati necessari all'esecuzione dell'istruzione

fase di execute

Viene eseguita l'operazione aritmetica e/o logica derivante dal codice dell'istruzione; il risultato è riscritto in un registro o in una locazione di memoria o su un dispositivo di I/O



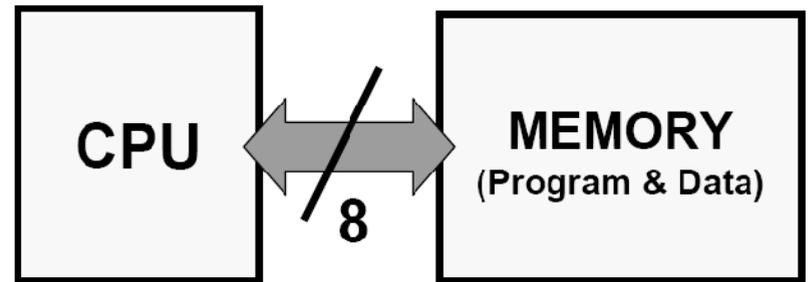
Architettura Von Neumann (Princeton)

Utilizzata di solito per processori general purpose

Prevede un bus unico tra CPU e memoria; in questo modo il progetto è molto semplificato e facilmente espandibile (es. embedded con memoria esterna)

Consente uno sfruttamento ottimo della memoria in quanto la parte di memoria "risparmiata" dal codice può essere utilizzata per i dati e viceversa

RAM (data memory) e ROM (program memory) devono avere entrambe parole della stessa lunghezza; se l'istruzione è più lunga di una parola sarà eseguita in più cicli macchina



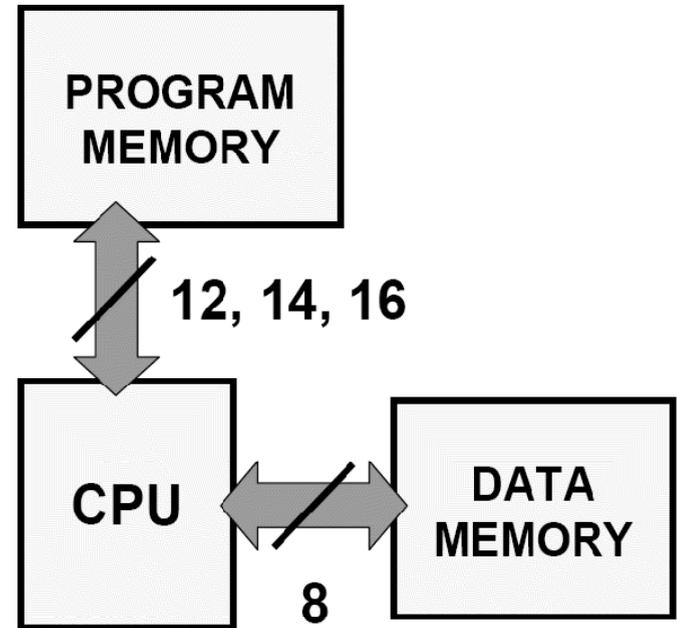
La condivisione di un bus unico fa sì che per completare un'istruzione sono necessari almeno due accessi in memoria (uno in program memory (fetch) e uno in data memory (execute)) per cui si può arrivare al limite ad eseguire UN'ISTRUZIONE OGNI DUE CICLI MACCHINA

Architettura Harvard/1

Utilizzata di solito per processori RISC

Prevede due bus separati tra CPU, program memory e data memory; il sistema è così molto complesso da progettare e da espandere (embedded)

Data memory word e program memory word possono avere lunghezza diversa; dimensionando il bus di collegamento con la program memory in modo da contenere la più grande istruzione dell'intero set, allora basterà una sola lettura per completare la ricerca del comando



La CPU può effettuare un accesso in data memory e in program memory contemporaneamente: si può arrivare ad eseguire UN'ISTRUZIONE OGNI CICLO MACCHINA

Architettura Harvard/2

Presenza di sottosezioni nel core che possono operare in parallelo con compiti specifici (es.: sezione per ricerche di codice/dati, sezione di esecuzione) → **pipeline**

PIPELINE A 2 STADI, ESECUZIONE IN SINGOLO CICLO

CLK 0

CLK 1

CLK 2

CLK 3

Fetch 1

Execute 1

Fetch 2

Execute 2

Fetch 3

Execute 3

Fetch 4

Von Neumann vs Harvard

The Von Neumann architecture has only one memory space to store both program and data memory. This means that it is necessary to fetch instructions and data from the same memory space, which limits your bandwidth because you can only transmit one piece of data or one instruction at one time.

With the Harvard architecture, there are two separate memory spaces, one for instructions and one for data. You can increase your throughput because while fetching the next instruction you can still be writing the result from the previous instruction. The other advantage is that since this is an 8-bit microcontroller, the data memory is 8-bits wide, but the program memory can be any width you choose. On existing PICmicro architectures, the program memory used is either 12-bits, 14-bits, or 16-bits wide.

CISC (Complex Instruction Set Computer)

È l'architettura della maggior parte delle CPU general purpose, di solito accompagnata ad una struttura Von Neumann

Molte istruzioni (>100) con lunghezza variabile

Generazione di codice compatto

Dimensione fisica del decoder istruzioni relativamente grande

Istruzioni "lente": più di un ciclo macchina per eseguirne una

Tempo di esecuzione deterministico

RISC (Reduced Instruction Set Computer)

È l'architettura della maggior parte dei microcontrollori a medio-alte prestazioni, di solito accompagnata ad una struttura Harvard

Poche istruzioni (<50)

Generazione di codice di elevate dimensioni

Dimensione fisica del decoder istruzioni molto ridotta

Istruzioni "veloci": un ciclo macchina per eseguire un'istruzione

Tempo di esecuzione difficilmente predicibile, a causa della presenza di architetture pipeline, superscalare e memoria cache, facilmente implementabili in processori RISC

Quando un core ad alte prestazioni è adottato per un microcontrollore, si semplifica eliminando parzialmente o totalmente tali caratteristiche

Periferiche on-chip e connessioni

Porte di I/O digitali generiche

Scopo: ingresso/uscita di segnali digitali. Es.: livelli logici, stati on-off, configurazioni binarie (interfacciamento)

Numero, massima frequenza dei segnali gestiti

Porte di I/O digitali specializzate

Es.: generazione di segnale PWM per il controllo velocità motore, segnali per pilotaggio display LCD, decodifica tastiere o acquisizione segnali da encoder.

Porte di I/O analogiche

I: (A/D) interfacciamento in ingresso con segnali analogici esterni e conversione in valori digitali interni

O: (D/A) interfacciamento analogico di uscita. Conversione di valori digitali interni in segnali analogici emessi verso l'esterno

Timer/counter

Scopo: contare lo scorrere del tempo per azioni temporizzate (timer) o il numero di eventi (counter) dovuti a transizioni logiche

Watchdog timer

Real-Time clock

Periferiche on-chip e connessioni

Porte di comunicazione

Seriali, parallele

Bus di campo, reti IP, wireless

Bus di interconnessione standard e proprietari

Scopo: connettere moduli esterni standard (decoder display a 7 segmenti), della stessa linea di prodotto o generali (es.: banco di RAM esterna)

Porte di interfacciamento speciale

Di programmazione/riprogrammazione; spesso coincidono con una porta di comunicazione

Di debug. Es.: standard JTAG

Power management

Frequency e/o voltage scaling

Modalità a basso consumo o spegnimento di tutto il chip o di parti di esso

Conseguenze

Set di istruzioni/architettura

Capacità elaborative e di indirizzamento delle strutture dati

Interessano le prestazioni complessive

Possono avere conseguenze sulla densità del codice prodotto (molto importante la quantità di memoria necessaria a contenere il programma)

Clock

A parità di architettura, condiziona prestazioni e consumo

Numero di bit

Condiziona tipicamente:

Le prestazioni su operandi di varia lunghezza e tipologia (interi, reali, complessi)

La densità del codice prodotto

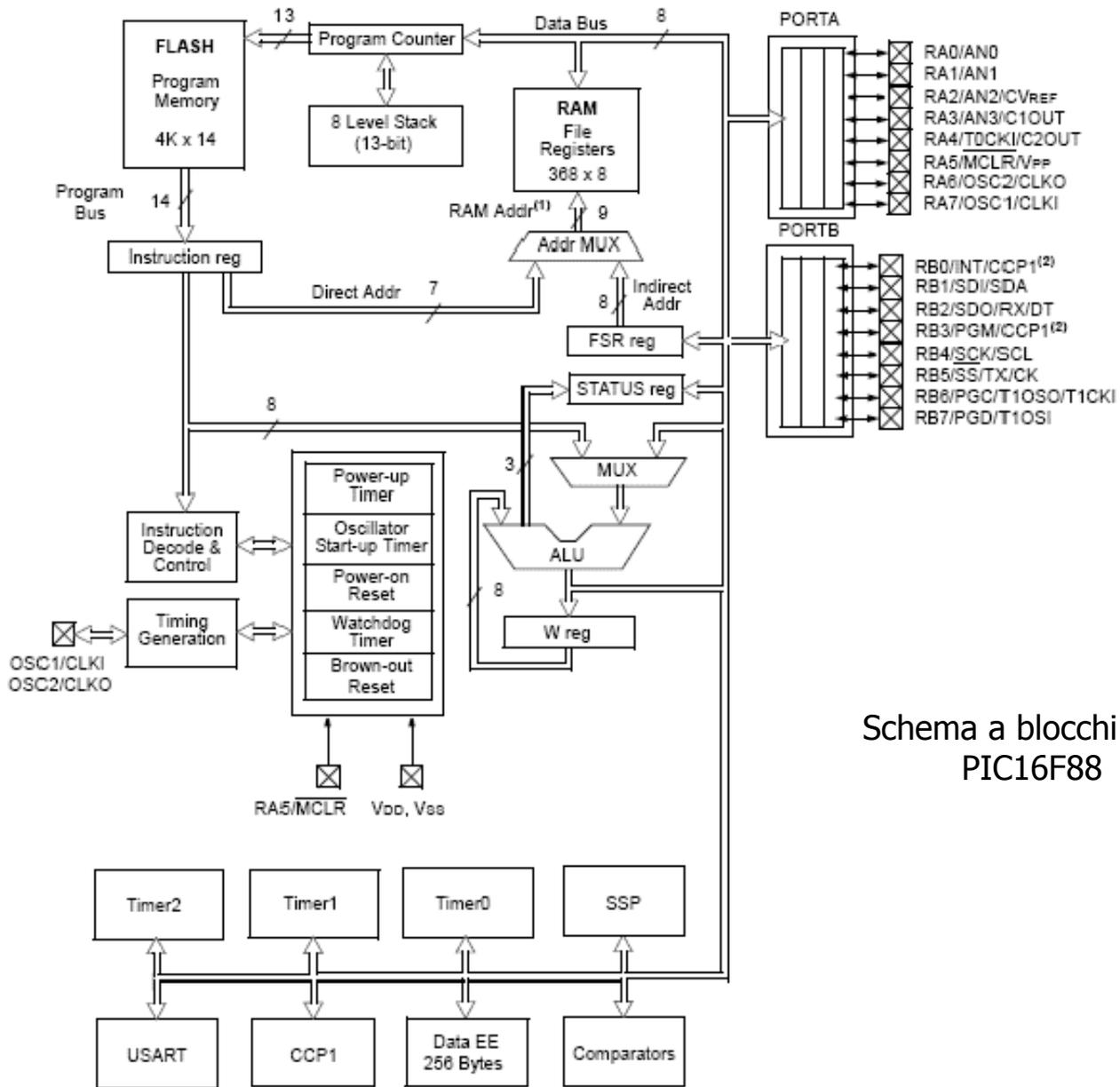
La dimensione della memoria indirizzabile

Il consumo di potenza

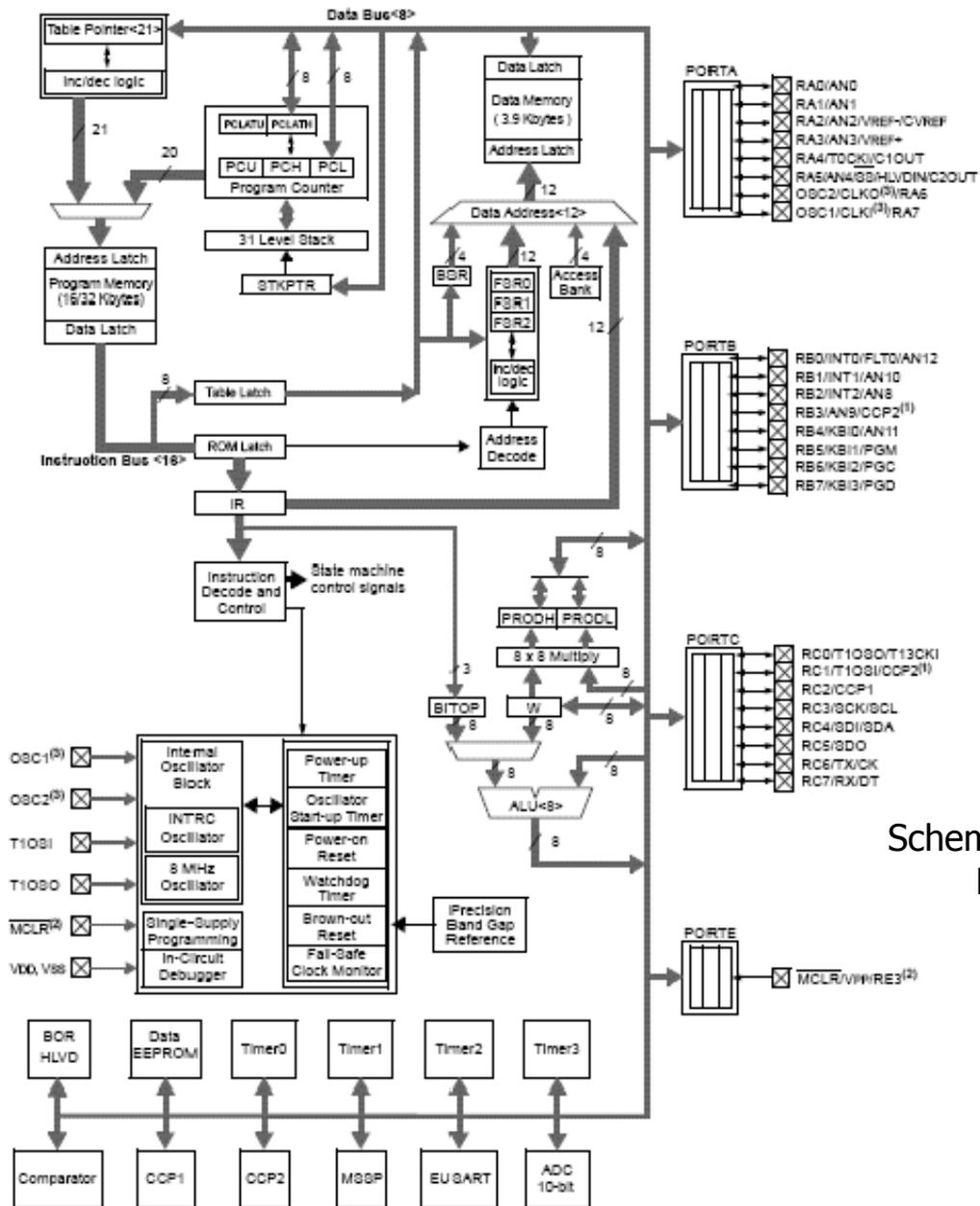
Numero e dispositivi di I/O

Condizionano l'ambiente in cui il microcontrollore può inserirsi

Ma è possibile aggiungere componenti esterni



Schema a blocchi di un PIC16F88



Schema a blocchi di un PIC18F2420

Prezzi/I

Anche i prezzi guidano nella scelta di un microcontrollore

I volumi che si intendono adottare possono comportare economie fino al 75% del prezzo unitario

Un'altra differenza deriva dal tipo di memoria programma che si vuole utilizzare:

- ROM pre-programmata

- OTP

- EPROM/EEPROM/Flash

In generale il costo di un microcontrollore è direttamente proporzionale alla capacità della memoria interna, alla velocità massima di elaborazione e dalle caratteristiche delle periferiche interne.

Tuttavia la differenza di prezzo fra microcontrollori di fascia alta e bassa si sta riducendo

Prezzi/II

Il prezzo del microcontrollore non è sempre l'unico fattore

Occorre considerare anche il costo dell'ambiente di sviluppo e le sue caratteristiche, in termini di produttività e di semplicità d'uso

Disponibilità di librerie specifiche per l'applicazione in oggetto

Un kit di sviluppo/valutazione per microcontrollori può costare fino a \$4000 per scheda

Un compilatore C, magari corredato da un IDE, può costare fino a \$1000

Inoltre RTOS (Real Time Operatig System) e librerie di software fornite da terze parti possono essere molto costose

I maggiori produttori

Atmel AVR RISC

Intel Pentium, I960, 8051

Texas Instruments TMS370

Hitachi H8/300H

Zilog Z86CXX

Siemens 80C165/6/7

Motorola 68HCXX

Microchip PIC

National Semiconductor COP800

ST ST6, ST8, ST10, ST40

Parallax Basic stamp

ARM ARM7, ARM9