

09-05-07



Introduzione ai microcontrollori PIC

Daniele Beninato: beninato@dei.unipd.it
Nicola Trivellin: trive1@tin.it

Che cosa è un microcontrollore?

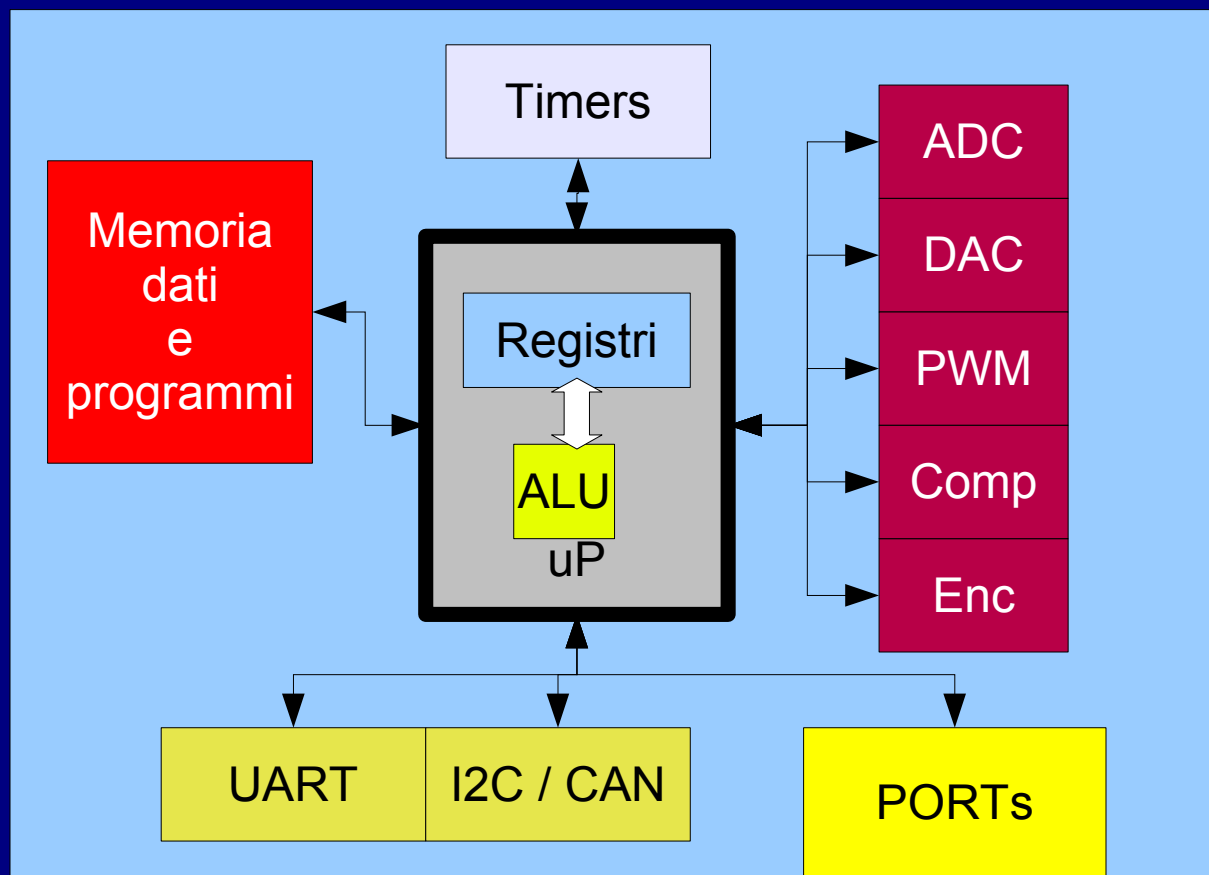
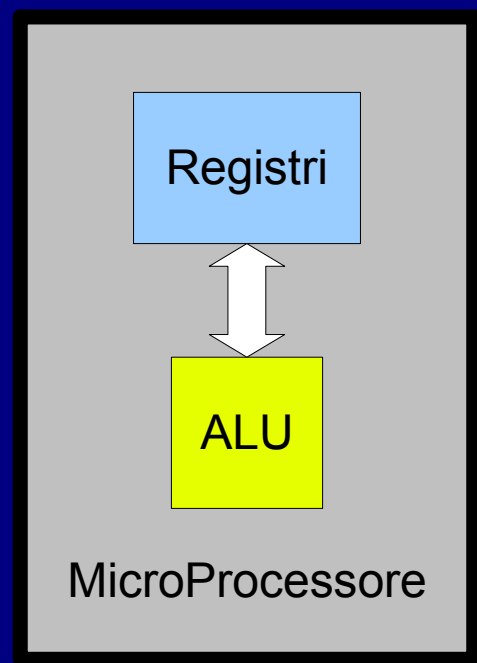
Un microcontrollore è un sistema a microprocessore completo, integrato in un solo chip, progettato per ottenere la massima autosufficienza funzionale ed ottimizzare il rapporto prezzo-prestazioni per una specifica applicazione, a differenza, ad esempio, dei microprocessori impiegati nei personal computer, adatti per un uso più generale.

I microcontroller sono la forma più diffusa e più invisibile di computer. Comprendono la CPU, un certo quantitativo di memoria RAM e memoria ROM (può essere PROM, EPROM, EEPROM o FlashROM) e una serie di interfacce di I/O (input/output) standard, fra cui molto spesso bus (I²C, SPI, CAN, LIN). Le periferiche integrate sono la vera forza di questi dispositivi: si possono avere convertitori ADC e convertitori DAC multicanale, timer/counters, USART, numerose porte esterne bidirezionali bufferizzate, comparatori, PWM.

Sono contenuti nella quasi totalità di apparecchi ed elettrodomestici.

La loro capacità di calcolo è molto limitata e di solito eseguono lo stesso programma (firmware) per tutta la durata del loro funzionamento.

Un microcontrollore è un sistema a microprocessore completo, integrato in un solo chip, progettato per ottenere la massima autosufficienza funzionale ed ottimizzare il rapporto prezzo-prestazioni per una specifica applicazione, a differenza, ad esempio, dei microprocessori impiegati nei personal computer, adatti per un uso più generale.



Che cosa è un microcontrollore PIC?

PIC è una famiglia di circuiti integrati a semiconduttore con funzioni di microcontrollore. Microchip technology non usa il termine PIC come un acronimo; il suo nome aziendale è: "PICmicro". Anche se generalmente sta per "Peripheral Interface Controller", originariamente la "General Instrument" usava l'acronimo per "Programmable Intelligent Computer". Originariamente sviluppato nel 1975 è in grado di svolgere un set di istruzioni ridotto (RISC).

Il PIC16f84 è un microcontrollore ad 8 bit e ha a disposizione una trentina di istruzioni. Tutti i registri sono di 8 bit e la memoria è organizzata in byte.

L'esecuzione delle istruzioni è scandita da un segnale di clock, cioè un'onda quadra a frequenza costante.

Che cosa sono le periferiche?

▶ **Porte**: costituiscono le periferiche di base per le comunicazioni da e verso l'esterno. Possono essere configurate come ingressi o uscite digitali.

▶ **Memoria**: Permette di conservare dati anche in assenza di alimentazione e contiene il codice programma, è di tipo FLASH!

▶ **ADC**: ad alcune porte del PIC è possibile applicare un ingresso analogico e quantizzarlo con una risoluzione tipica di 10bit.

▶ **Timer**: sono presenti sempre e vengono utilizzati spesso per generare ritardi finiti o per la misurazione d'intervalli di tempo (ad esempio eventi esterni).

▶ **PWM**: permette, utilizzando un Timer di generare un segnale di comando PWM.

▶ **Comunicazioni**: consente l'interfacciamento attraverso protocolli standard (I2C, UART, CAN, ...) con altri dispositivi.

Che cosa sono le periferiche?

Per poter utilizzare le periferiche sono presenti dei registri che servono a configurarle. Si tratta di locazioni di memoria di 1byte in cui ad ogni bit è associato un significato specifico.

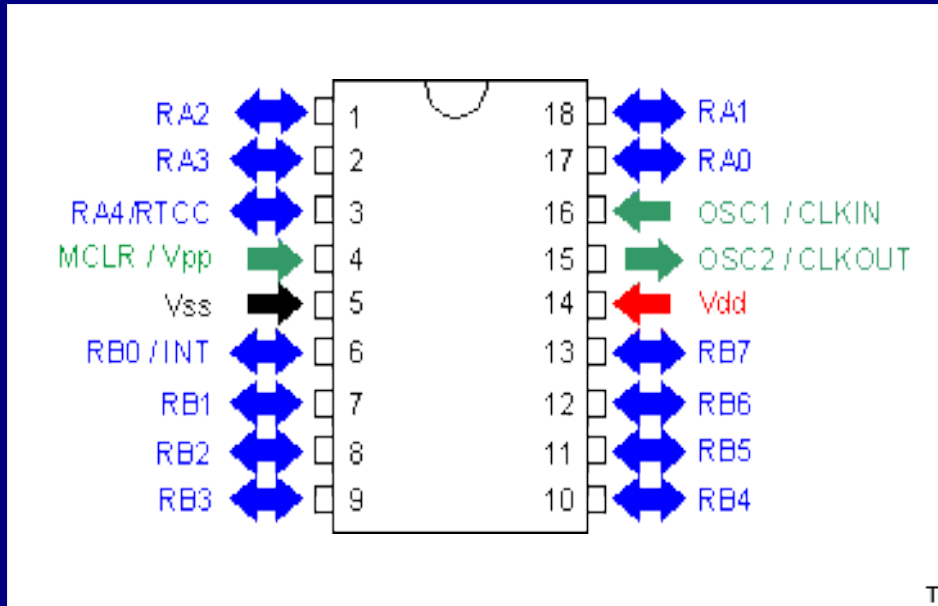
Ogni registro ha un nome riservato e l'elenco dei registri con la loro funzione è presente nei data-sheet.

Ad esempio:

Ogni pin di una porta può essere configurato come ingresso o uscita attraverso il registro TRISx.

Mentre il valore presente o inviato alla porta è scritto nel registro PORTx

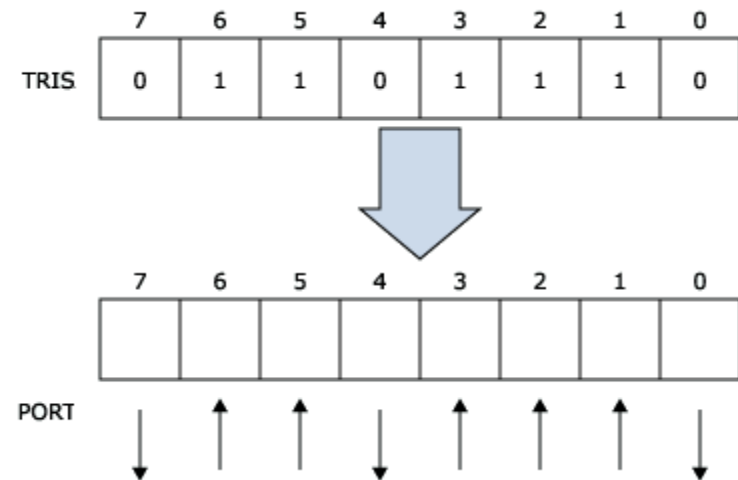
Periferiche sul pic 16F84: 1 PORTA e MEZZA (anzi 5/8)



Vogliamo configurare i pin della porta B come 5 ingressi e 3 uscite, scriveremo quindi nel registro TRISB il valore 01101110.

Se voglio che il pin 7 della porta B sia un'uscita il bit 7 del registro TRISB dovrà essere a 0 .

Nel registro PORTB potremmo andare a leggere i valori dei pin 6, 5, 3, 2, 1 e a scrivere quelli dei pin 7, 4 e 0.



Periferiche sul pic 16F84: il Timer

Il Timer è un particolare registro chiamato TMR0 il cui contenuto viene incrementato con cadenza regolare e programmabile direttamente dall'hardware del PIC.

Il registro TMR0 può essere impostato ad un valore qualsiasi, compreso tra 0 e 255, dopodichè sarà incrementato automaticamente dal controllore con una frequenza basata sul clock del microcontrollore.

Al raggiungimento di 255 (1111 1111) il timer riparte da zero.

Attraverso il registro OPTION è possibile impostare il numero di cicli di clock necessari all'incremento di una unità del timer.

Questo divisore viene definito PRESCALER.

Registro accumulatore: è un registro temporaneo chiamato W, indispensabile per eseguire le istruzioni RISC.

```
movlw    0
movwf    TRISB

movlw    10
movwf    TMR0

ciclo
movfTMR0,W
movwf    PORTB
goto    ciclo
```

Per far eseguire delle operazioni al microprocessore, è necessario “parlargli” in linguaggio macchina (stringhe di bit).

Ogni istruzione corrisponde ad una stringa di bit, poiché sarebbe impensabile per noi imparare il codice binario delle istruzioni, si possono utilizzare delle etichette mnemoniche che un compilatore trasformerà in stringhe di bit.

Consideriamo le prime due istruzioni:

```
movlw 0
movwf TRISB
```

```
movlw 10
movwf TMR0
```

ciclo

```
movf TMR0, W
movwf PORTB
goto ciclo
```

movlw 0

MOVE Literal to W

Questa istruzione assegna all'accumulatore W il valore costante (literal) contenuto nel "campo operandi", in questo caso zero.

0 -> W

movwf TRISB

MOVE W to File Register

Questa istruzione assegna al registro contenuto nel campo operandi il valore memorizzato nell'accumulatore W.

In questo caso il contenuto di W viene copiato nel registro TRISB, impostando in questo modo tutti i pin della porta B come uscite.

0 -> W e poi W -> TRISB = 0 -> TRISB

```
movlw    0
movwf    TRISB
```

```
movlw    10
movwf    TMR0
```

```
ciclo
```

```
movf    TMR0,W
movwf    PORTB
goto    ciclo
```

Infine l'ultima istruzione:

goto **ciclo**

salta alla sezione di codice etichettata come “ciclo” e ricomincia ad eseguire le operazioni da quel punto.

Nel nostro caso verranno eseguite all'infinito le ultime 3 istruzioni e quindi i pin di uscita della porta B, fungeranno da contatore binario.

```

movlw    0
movwf    TRISB

movlw    10
movwf    TMR0

ciclo
movf TMR0,W
movwf PORTB
goto ciclo

```

La terza e la quarta istruzione svolgono la stessa identica funzione delle istruzioni precedenti con il risultato di assegnare al registro TMR0 il valore iniziale 10.

10 -> W -> TMR0

“ciclo” è una semplice etichetta ed ha lo scopo di marcare (come un segnalibro) una determinata posizione all'interno del codice.

movf TMR0,W

Copia il contenuto del primo registro inserito nel campo operandi nel secondo registro, nel nostro caso sposta il valore attuale del timer nell'accumulatore W.

movwf PORTB

come già visto copia il contenuto di W nel registro PORTB. Questo ha l'effetto immediato di fornire nei pin della porta B 0 o 5 volt a seconda dello stato logico del bit corrispondente.

Che cosa vogliamo fare?

Introduzione al linguaggio Assembly del PIC

Utilizzo del PIC 16F84A tramite esempi pratici

Se rimane tempo PIC 16F876

Riferimenti per materiale

www.microchip.com

www.tanzilli.com

Che cosa (non) serve sapere?

Architettura degli elaboratori

Fondamenti di informatica 2

Microprocessori & DSP

Quando ci possiamo trovare?