

## Il temporizzatore TMR0

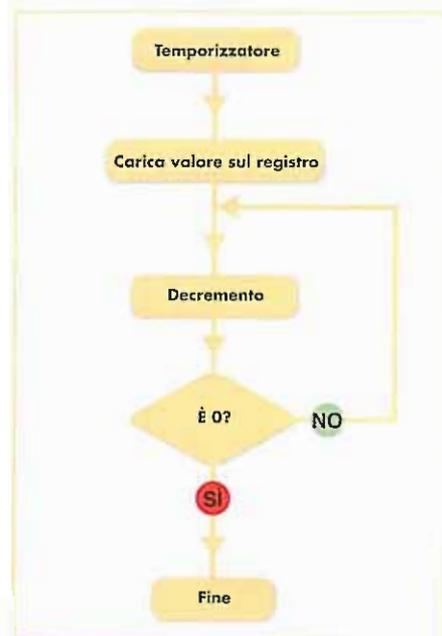
I temporizzatori sono contatori che si caricano con un determinato valore all'inizio del conteggio, e questo valore sarà aumentato, o diminuito, a ogni impulso di clock. Quando vanno in overflow, ossia superano il valore massimo che possono contare, oppure vanno a zero, fanno automaticamente passare il flag corrispondente a 1, per avvisare della fine del conteggio del tempo.

Sono molto utili perché evitano al processore di doversi dedicare, all'interno del programma principale, a conteggi di tempo. Sono temporizzatori specializzati e quando terminano attivano sempre il flag. Se è necessario possono anche provocare un interrupt al processore. Nella figura è riportata la struttura generale di un temporizzatore, e l'organigramma di funzionamento quando è impiegato per misurare il tempo. Il

TMR0 è un contatore a 8 bit, ascendente, che quando raggiunge il valore massimo (che è FF Hex) passa a 00 Hex con il successivo impulso, e alza il flag ponendo TMR0IF = 1, che è anche chiamato TOIF. Gli impulsi di clock per l'incremento del TMR0 possono anche arrivare dall'esterno, tramite il pin RA4/TOCKI, oppure tramite l'oscillatore principale a una frequenza di  $F_{osc}/4$ .

### Calcolo del tempo

Il tempo che impiega il TMR0 dal momento in cui è caricato con il valore sino a quando va in overflow dipende dal valore del registro TMR0, che occupa l'indirizzo 1 del banco 0 della RAM. Inoltre dipende anche dalla possibile attivazione del Predivisore di Frequenza che può essere



**Organigramma del funzionamento di un temporizzatore per la misurazione del tempo.**

applicato a questo temporizzatore.

Questo Predivisore divide per un determinato range gli impulsi da applicare al temporizzatore. Il Predivisore può essere destinato al TMR0 oppure al Watchdog. La formula da utilizzare per calcolare il tempo è la seguente:

$$\text{Tempo} = 4 \times T_{osc} \times (256 - N) \times \text{Range Predivisore}$$

dove N è il valore in decimale che è stato precedentemente caricato nel TMR0. Se ad esempio si carica il valore 156 in decimale e il processore funziona a 4 MHz, ricevendo gli impulsi dell'oscillatore interno, il tempo che impiega il temporizzatore ad andare in overflow sarà:



**Struttura generale di un temporizzatore.**



**Il TMR0 può lavorare con impulsi di clock interni o esterni e quando va in overflow imposta il Flag TOIF = 1.**

# I favolosi PIC16F87X

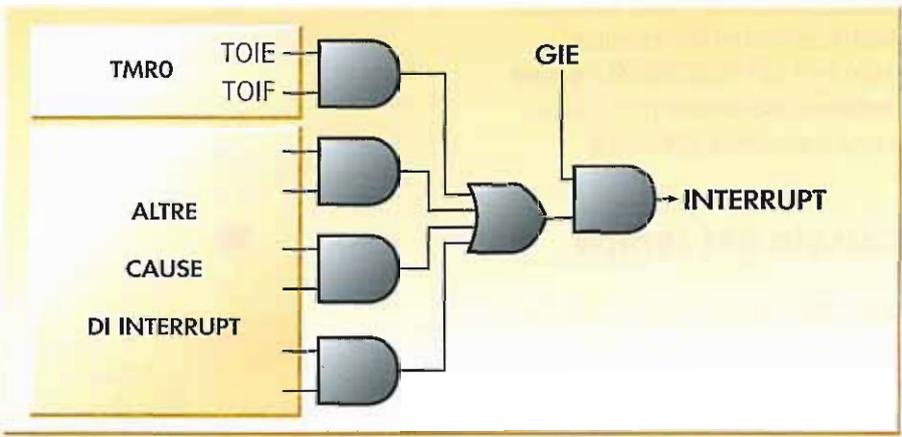


$$\text{Tempo} = 4 \times 250 \text{ ns} \times (256 - 156) \times 16 = 1,6 \text{ ms}$$

## Struttura interna

Nello schema della figura sono riportati gli elementi che circondano e complementano il TMR0. Per gestire lunghi intervalli di tempo bisogna aumentare la durata fra gli impulsi di clock, questo si può fare utilizzando il Predivisore di Frequenza programmabile che divide la frequenza degli impulsi per un dato valore.

Sia il TMR0 che il Watchdog possono utilizzare il divisore di frequenza che è gestito mediante un bit del registro OPTION.



## Il registro OPTION

La maggior parte dei bit di questo registro regolano il comportamento del TMR0, solo i due più significativi hanno funzioni assegnate ad altre risorse, come si può vedere nella figura.

Se il TMR0 va in overflow il TOIF passa a 1. Per fare in modo che si generi un interrupt è necessario che GIE = TOIE = 1.

OPTION							
RBPU#	INTDG	TOCS	TOSE	PSA	PS2	PS1	PS0
PS2:PS0 Valore con cui funziona il divisore di frequenza							
PS2	PS1	PS0	Divisione del TMR0	Divisione del WDT			
0	0	0	1:2	1:1			
0	0	1	1:4	1:2			
0	1	0	1:8	1:4			
0	1	1	1:16	1:8			
1	0	0	1:32	1:16			
1	0	1	1:64	1:32			
1	1	0	1:128	1:64			
1	1	1	1:256	1:128			

**PSA:** Assegnazione del divisore di frequenza  
 1= Il divisore di frequenza è assegnato al WDT  
 0= Il divisore di frequenza è assegnato al TMR0

**TOSE:** Tipo di fronte in TOCK1  
 1= Incremento del TMR0 con fronte di discesa  
 0= Incremento del TMR0 con fronte di salita

**TOCS:** Tipo di clock per il TMR0  
 1= Impulsi inseriti tramite TOCK1 (contatore)  
 0= Impulsi di clock interno Fosc/4 (temporizzatore)

**INTDG:** Fronte attivo interrupt esterno  
 1= Fronte di salita  
 0= Fronte di discesa

**RBPU#:** Resistenze pull up porta B  
 1= Disattivata  
 0= Attivata

Distribuzione dei bit del registro OPTION e assegnazione di funzioni.

## L'interrupt del TMR0

Quando va in overflow il TMR0, viene posto a 1 unicamente il flag TOIF. Per fare in modo che il processore si accorga di questo avvenimento ci sono due alternative: la prima consiste nel far testare dal programma lo stato del bit TOIF fino a che trovi il valore 1.

Con questo metodo la CPU non presta immediata attenzione al TMR0, ma lo fa secondo la sequenza del programma. Per avvisare immediatamente la CPU si utilizza l'interrupt. Per generare l'interrupt quando TOIF = 1 bisogna impostare a 1 i vari bit di abilitazione, quello generale e quello particolare del temporizzatore, cioè, GIE = TOIE = 1.