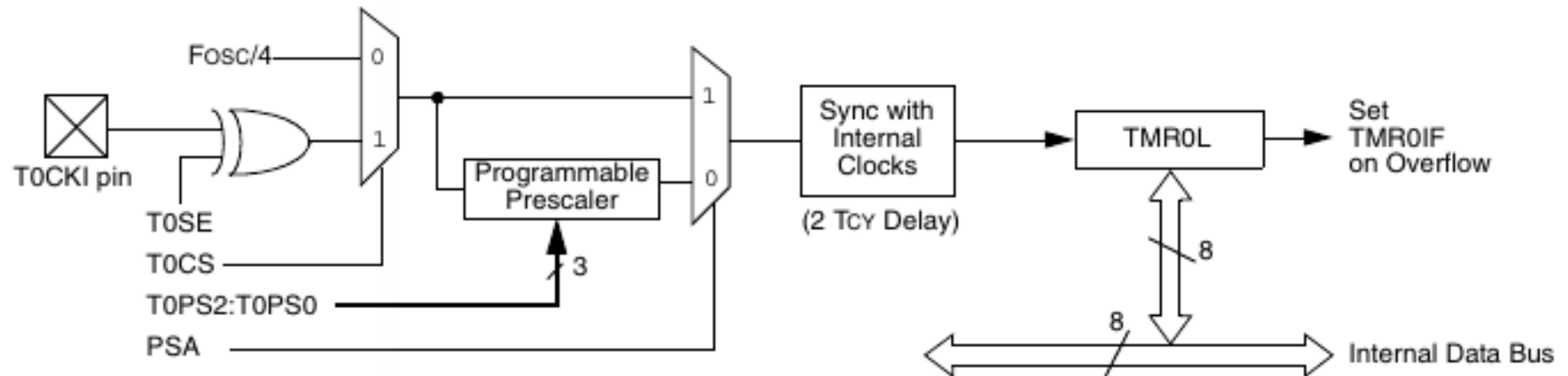


# **MICROCONTROLADORES**

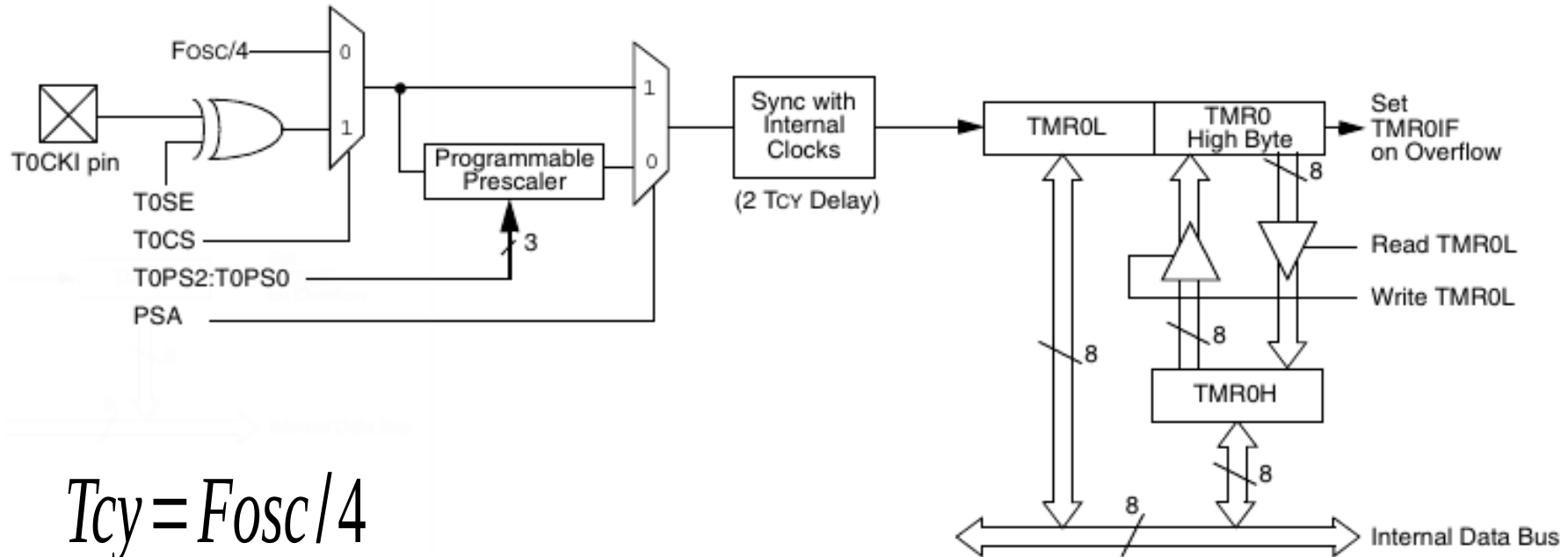
# TIMER 0 - 8 bits.



$$T_m = 2 * T_{cy} + (T_{cy} * Prescaler) * (255 - TMR0L)$$

$$T_{cy} = F_{osc} / 4$$

# TIMER 0 - 16 bits



$$T_{cy} = F_{osc} / 4$$

$$T_m = 2 * T_{cy} + (T_{cy} * Prescaler) * (65535 - TMR0)$$

LA ESPECIFICACIÓN DE TMR0H Y TMR0L DEBE REALIZARSE EN UN ORDEN ESPECIFICO, PRIMERO LA PARTE ALTA Y DESPUÉS LA PARTE BAJA.

# TIMER 0

- Para iniciar el funcionamiento del Timer 0, se debe colar en uno el bit TMR0ON y en cero para detenerlo.
- Después de que se active la bandera de overflow del Timer 0, se debe volver a cargar los registros TMR0L y TMR0H para volver a iniciar la contemporización

# TIMER 0 - LIBRERIA.

**Timer\_0.h : Definición de las cabecera de las funciones y las variables globales requeridas por las funciones del archivo Timer\_0.c.**

**Timer\_0.c : Código fuente de todas las funciones usadas para configurar y testear el estado del tiempo establecido para el Timer.**

# Timer 0 - timer\_0.h

**//Variables globales**

**int periodo; //Periodo requerido para el timer**

**unsigned char tl; //Guarda el calculo para TMR0L**

**unsigned char th; ////Guarda el calculo para TMR0H**

**void set\_timer(int time); //configura timer**

**void set\_time(); //establece TMR0L y TMR0H**

**void init\_timer(unsigned char state); //Enciende Timer**

**Unsigned char flag\_time(); //Lee la bandera de overflow**

# Timer 0 - Timer\_0.c

**void set\_timer(int time):**

**Calcula la configuración del prescaler y de TMR0L-TMR0H.**

**Se usa un rango de tiempo para determinar el prescaler y los TMR0L-TMR0H se obtienen de la diferencia de la base de tiempo mínima obtenida con el prescaler configurado y el tiempo**

| T0PS   | 000   | 001    | 010    | 011    | 100    | 101    | 110    | 111    |
|--------|-------|--------|--------|--------|--------|--------|--------|--------|
| BASE   | 1 us  | 2 us   | 4 us   | 8 us   | 16 us  | 32 us  | 64 us  | 128 us |
| LIMITE | 65 ms | 131 ms | 262 ms | 524 ms | 1,04 s | 2,09 s | 4,19 s | 8,38 s |

# TIMER 0 - Timer\_0.c

```
TMR0L-TMR0H=Limite- time;
```

## **void set\_time():**

```
TMR0L=tl
```

```
TMR0H=th
```

## **void init\_timer(unsigned char state):**

```
TMR0ON= state
```

## **Unsigned char flag\_time():**

```
If (TMR0IF=1){
```

```
    set_time()
```

```
    TMR0IF=0
```

```
    int_timer(1)
```

```
    Return 1
```