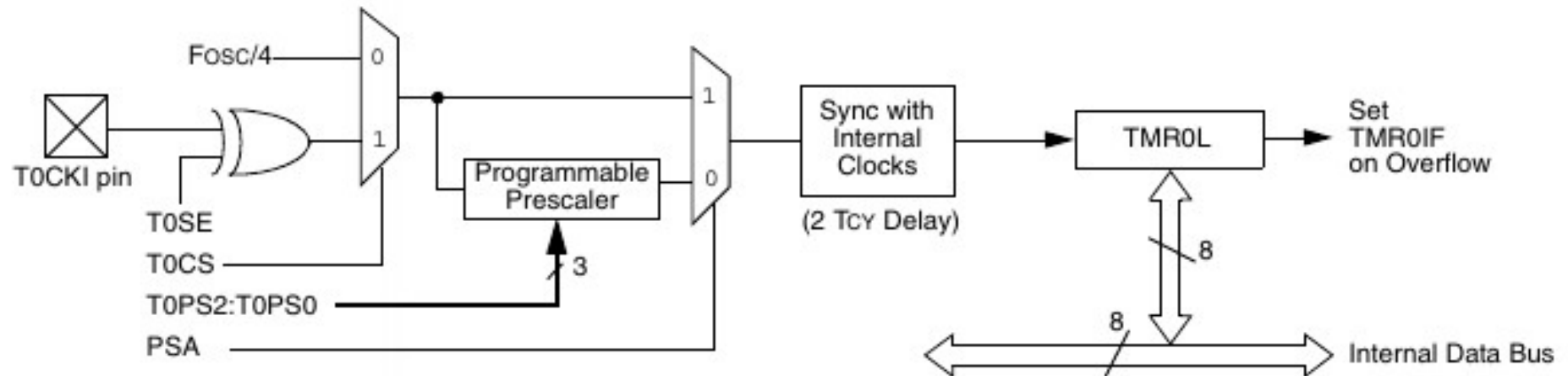


MICROCONTROLADORES

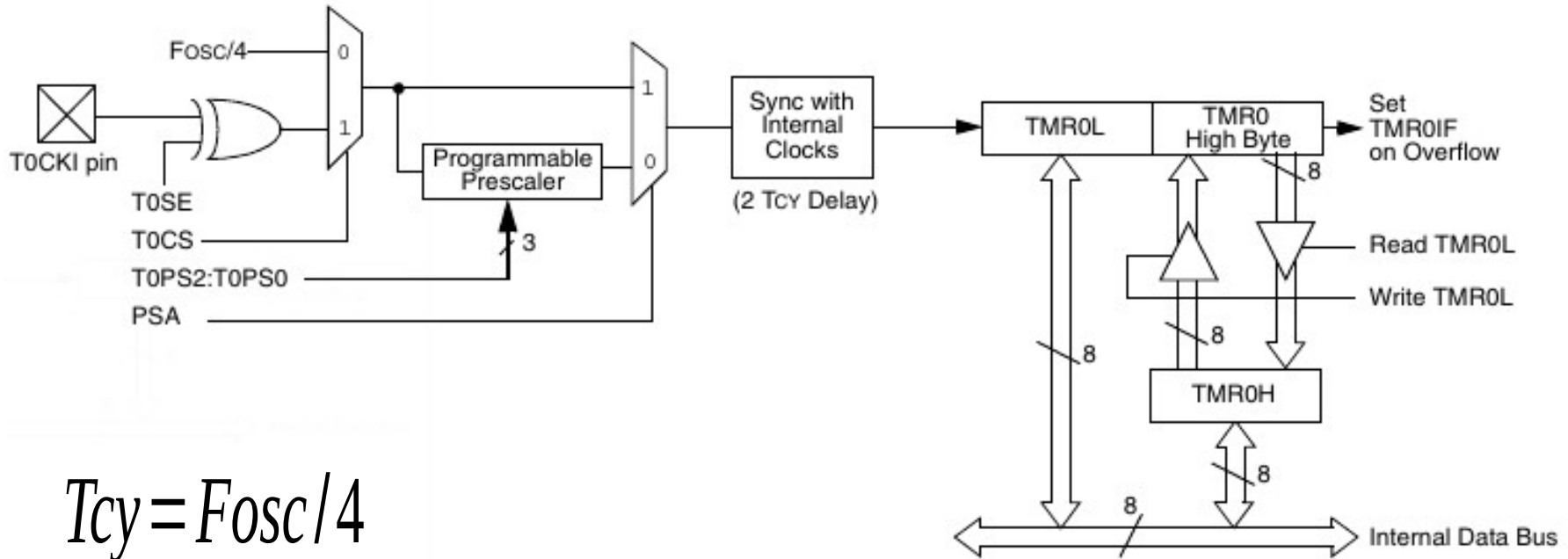
TIMER 0 – 8 bits.



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

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

TIMER 0 – 16 bits



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

$$Tm = (Tcy * Prescaler + 2 * Tcy) * (65535 - TMR0)$$

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 deseado.

T0PS	000	001	010	011	100	101	110	111
BASE	2 us	3 us	5 us	9 us	17 us	33 us	65 us	129 us
LIMITE	131 ms	196 ms	327,6 ms	589,8 ms	1,11 s	2,16 s	4,12 s	8,45 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
```