

RELOJ DEL SISTEMA: OSCILADORES Y PLL (section 7)

El sistema de reloj de los PIC24H dispone de las siguientes opciones:

- Dos relojes internos y dos entradas para conectar relojes externos.
- Un PLL (*Phase-Locked Loop*) que permite modificar la frecuencia de operación de la CPU.
- Cambio de un reloj a otro en tiempo real.
- Detección de fallo del reloj.

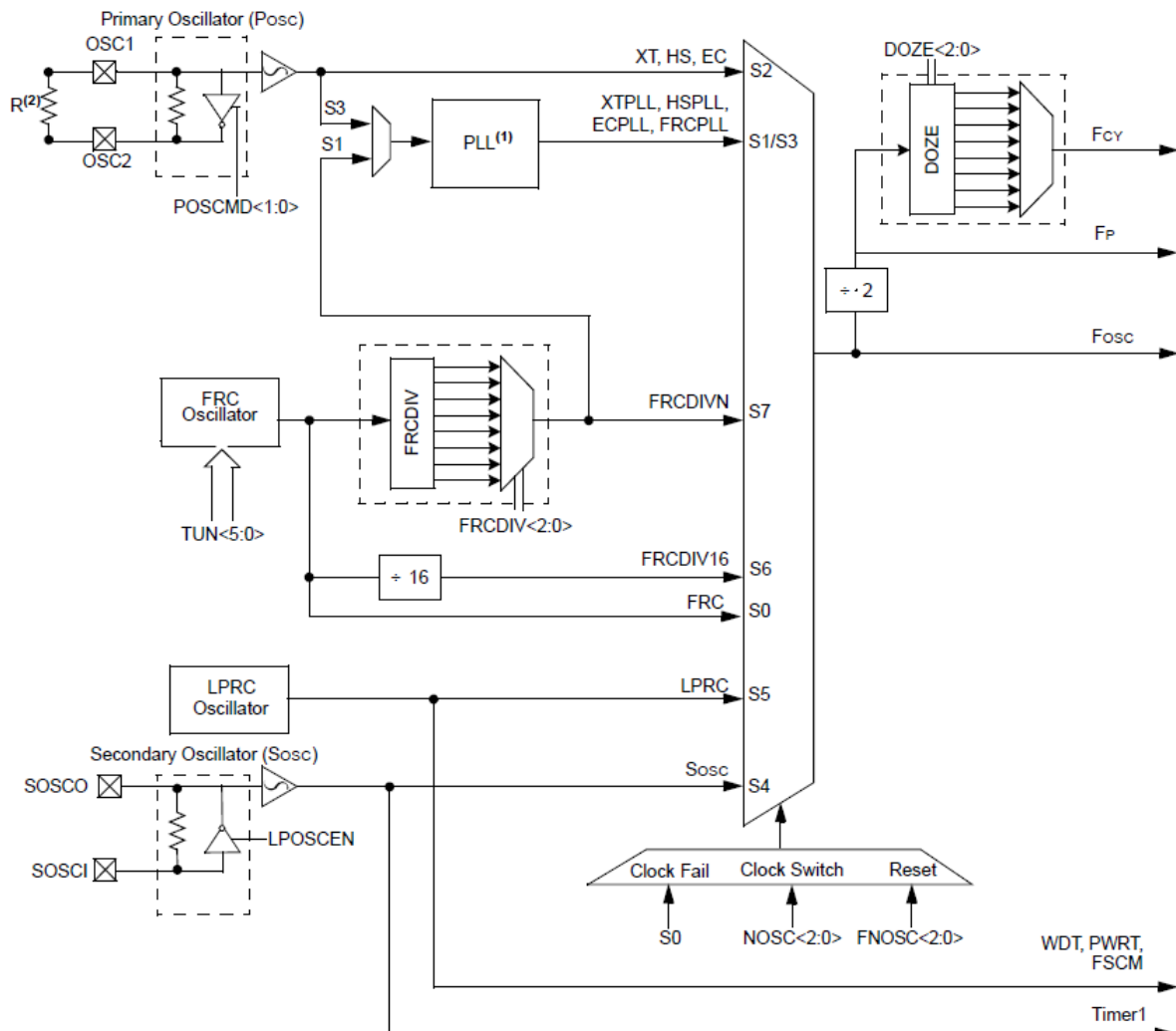
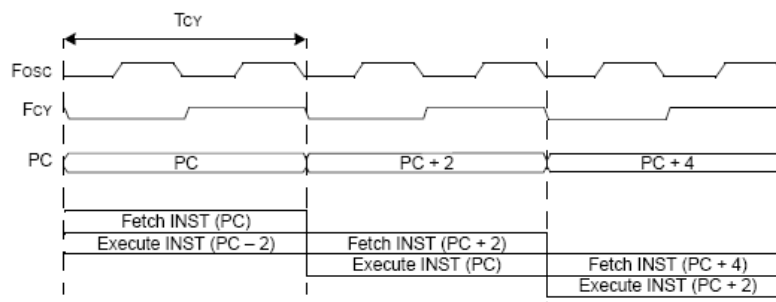


DIAGRAMA del SISTEMA del OSCILADOR en el PIC24H



Tiempos de ciclo de instrucción / Reloj

El microcontrolador PIC24 dispone de los siguientes registros para controlar el funcionamiento de los osciladores:

- **2 registros de configuración**, mapeados en el espacio de memoria de programa: **FOSCSEL** y **FOSC**.
- **4 registros propios (SFR)**: **OSCCON**, **CLKDIV**, **OSCTUN** y **PLLFBD**

FOSCSEL (Oscillator Source Selection): Registro de Selección del Oscilador Principal

U	U	U	U	U	U	U	U	R/P	U	U	U	U	R/P	R/P	R/P	
-	-	-	-	-	-	-	-	IESO	-	-	-	-	FNOSC<2:0>			
Bit15								Bit8	Bit7	bit0						

FOSC (Oscillator Configuration): Registro de Configuración del Oscilador

U	U	U	U	U	U	U	U	R/P	R/P	R/P	U	U	R/P	R/P	R/P
-	-	-	-	-	-	-	-	FCKSM<1:0>	IOL1WAY	-	-	OSCIOFNC	POSCMD<1:0>		
Bit15								Bit8	Bit7	bit0					

R = Readable bit P = Programmable bit U = Unused bits, program to Logic '1'

IESO: Internal External Start-up Option bit

1 = Start-up device with FRC, then automatically switch to the user-selected oscillator source when ready

0 = Start-up device with user-selected oscillator source

FNOSC<2:0>: Initial Oscillator Source Selection bits

111 = Internal Fast RC (FRC) oscillator with postscaler

110 = Reserved

101 = LPRC oscillator

100 = Secondary (LP) oscillator

011 = Primary (XT, HS, EC) oscillator with PLL

010 = Primary (XT, HS, EC) oscillator

001 = Internal Fast RC (FRC) oscillator with PLL

000 = FRC oscillator

FCKSM<1:0>: Clock Switching Mode bits

1x = Clock switching is disabled, Fail-Safe Clock Monitor is disabled

01 = Clock switching is enabled, Fail-Safe Clock Monitor is disabled

00 = Clock switching is enabled, Fail-Safe Clock Monitor is enabled

OSCIOFNC: OSC2 Pin Function bit (except in XT and HS modes)

1 = OSC2 is clock output

0 = OSC2 is general purpose digital I/O pin

POSCMD<1:0>: Primary Oscillator Mode Select bits

11 = Primary oscillator disabled

10 = HS Crystal Oscillator mode

01 = XT Crystal Oscillator mode

00 = EC (External Clock) mode

Oscilador	Modo	Fuente	FNOSC<2:0>	POSCMD<1:0>
S0	Fast RC Oscillator (FRC)	Internal	000	xx (11)
S1	Fast RC Oscillator with PLL (FRCPLL)	Internal	001	xx (11)
S2	Primary Oscillator (EC)	Primary	010	00
S2	Primary Oscillator (XT)	Primary	010	01
S2	Primary Oscillator (HS)	Primary	010	10
S3	Primary Oscillator with PLL (ECPLL)	Primary	011	00
S3	Primary Oscillator with PLL (XTPLL)	Primary	011	01
S3	Primary Oscillator with PLL (HSPLL)	Primary	011	10
S4	Secondary Oscillator (SOSC)	Secondary	100	xx (11)
S5	Low Power RC Oscillator	Internal	101	xx (11)
S6	Fast RC Oscillator with /16 divider (FRCDIV16)	Internal	110	xx (11)
S7	Fast RC Oscillator with /N divider (FRCDIVN)	Internal	111	xx (11)

Selección de los diferentes Tipos de Osciladores

La activación de algunos de estos *bits* se puede hacer a partir de las definiciones hechas en el fichero: p24HJ256GP610A.h

```

** FNOSC_FRC          Fast RC oscillator
** FNOSC_FRCPLL       Fast RC oscillator w/ divide and PLL
** FNOSC_PRI          Primary oscillator (XT, HS, EC)
** FNOSC_PRIPLL       Primary oscillator (XT, HS, EC) w/ PLL
** FNOSC_SOSC         Secondary oscillator
** FNOSC_LPRC         Low power RC oscillator
** FNOSC_FRCDIV16     Fast RC oscillator w/ divide by 16
** FNOSC_LPRCDIVN     Low power Fast RC oscillator w/divide by N

** Two-speed Oscillator Startup :
** IESO_OFF           Disabled
** IESO_ON            Enabled

** Clock switching and clock monitor:
** FCKSM_CSECME       Both enabled
** FCKSM_CSECMD       Only clock switching enabled
** FCKSM_CSDCMD       Both disabled

** OSCIOFNC_ON        Digital I/O
** OSCIOFNC_OFF       OSC2 is clock O/P

** POSCMD_EC          External clock
** POSCMD_XT          XT oscillator
** POSCMD_HS          HS oscillator
** POSCMD_NONE        Primary disabled

```

Registro OSCCON (*Oscillator Control*): Registro de estado y control de cambio de oscilador

U-0	R-y	R-y	R-y	U-0	R/W-y	R/W-y	R/W-y	R/SO-0	R/W-0	R-0	U-0	R/C-0	U-0	R/W-0	R/W-0
-	COSC<2:0>			-	NOSC<2:0>			CLKLOCK	LOCK	LOCK	-	CF	-	LPOSCEN	OSWEN
Bit15					Bit8			Bit7							bit0

COSC<2:0>: Indica el tipo de oscilador actual (read-only)

- 111 = Oscilador Interno RC Rápido con postscaler (Fast RC Oscillator with Divide-by-N: FRCDIVN)
- 110 = Oscilador Interno RC Rápido con postscaler =16 (Fast RC Oscillator with Divide-by-16: FRCDIV16)
- 101 = Low-Power RC (LPRC) Oscillator
- 100 = Oscilador Secundario (SOSC) (LP)
- 011 = Oscilador Primario con PLL (XTPLL, HSPLL, ECPLL)
- 010 = Oscilador Primario (Posc) (XT, HS, EC)
- 001 = Oscilador Interno Rápido con PLL (Fast RC Oscillator with PLL: FRCPLL)
- 000 = Oscilador Interno Rápido (Fast RC Oscillator: FRC)

NOSC<2:0>: Selecciona el nuevo tipo de oscilador

- 111 = Oscilador Interno RC Rápido con postscaler (Fast RC Oscillator with Divide-by-N: FRCDIVN)
- 110 = Oscilador Interno RC Rápido con postscaler =16 (Fast RC Oscillator with Divide-by-16: FRCDIV16)
- 101 = Low-Power RC (LPRC) Oscillator
- 100 = Oscilador Secundario (SOSC) (LP)
- 011 = Oscilador Primario con PLL (XTPLL, HSPLL, ECPLL)
- 010 = Oscilador Primario (Posc) (XT, HS, EC)
- 001 = Oscilador Interno Rápido con PLL (Fast RC Oscillator with PLL: FRCPLL)
- 000 = Oscilador Interno Rápido (Fast RC Oscillator: FRC)

CLKLOCK: Habilita bloqueo de reloj

Si FCKSM<1:0> (FOSC<7:6>) = 01, entonces:

- 1 = Cambio de reloj deshabilitado, sistema de fuente de reloj bloqueada
- 0 = Cambio de reloj habilitado, sistema de fuente de reloj permitida por cambio de reloj

LOCK: Bit de estado de bloqueo del PLL (read-only)

- 1 = PLL bloqueado
- 0 = PLL no bloqueado

CF: Clock Fail Detect bit (read or cleared by application)

- 1 = FSCM ha detectado fallo de reloj / 0 = FSCM no ha detectado fallo

LPOSCEN: Habilita Oscilador secundario (SOSC)

- 1 = Habilitado / 0 = Deshabilitado

OSWEN: Habilita cambio de reloj

- 1 = Petición de cambio de reloj al especificado por NOSC<2:0>
- 0 = Cambio de oscilador completada

Registro CLKDIV (Clock Divisor): Control del modo Doze y de los prescaler y postscaler

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROI	DOZE<2:0>	DOZEN		FRCDIV<2:0>		PLLPOST<1:0>		-			PLLPRE<4:0>				
Bit15				Bit8	Bit7									bit0	

ROI: Recupera la acción de las interrupciones

1 = Las interrupciones borran bit DOZEN y la relación del reloj de CPU y periféricos es 1:1

0 = Las interrupciones no tienen efecto en el bit DOZEN

DOZE<2:0>: Selecciona la reducción de reloj FCY

111 = FCY / 128

110 = FCY / 64

101 = FCY / 32

100 = FCY / 16

011 = FCY / 8 (por defecto)

010 = FCY / 4

001 = FCY / 2

000 = FCY / 1

DOZEN: Habilita el Modo DOZE (Este bit se borra cuando se active el bit ROI y ocurre una interrupción)

1 = DOZE<2:0> especifica la proporción entre el reloj de los periféricos y el de la CPU

0 = Reloj de CPU y periféricos es el mismo (1:1)

FRCDIV<2:0>: Configura el Postscaler del Oscilador Interno FRC

111 = FRC / 256

110 = FRC / 64

101 = FRC / 32

100 = FRC / 16

011 = FRC / 8

010 = FRC / 4

001 = FRC / 2

000 = FRC / 1 (por defecto)

PLLPOST<1:0>: Selecciona la Salida VCO del Divisor ('N2', PLL postscaler)

00 = Output / 2

01 = Output / 4 (por defecto)

10 = Reserved

11 = Output / 8

PLLPRE<4:0>: PLL Phase Detector Input (N1)

11111 = Input / 33

...

00001 = Input / 3

00000 = Input / 2 (por defecto)

Registro PLLFBD (PLL Feedback Divisor): Selección del valor M del PLL

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	-	-	-	-	-									
Bit15							Bit8	Bit7							bit0

PLLDIV<8:0>: PLL Feedback Divisor ('M', PLL multiplier)

111111111 \Rightarrow M= 513 (511+2)

...

000110000 \Rightarrow M= 50 (48+2 por defecto)

...

000000010 \Rightarrow M= 4 (2+2)

000000001 \Rightarrow M= 3 (1+2)

000000000 \Rightarrow M= 2 (0+2)

Registro OSCTUN (FRC Oscillator Tuning): Ajuste de la frecuencia del oscilador interno

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	-	-	-	-	-	-	-	-	TUN<5:0>							
Bit15								Bit8		Bit7		bit0					

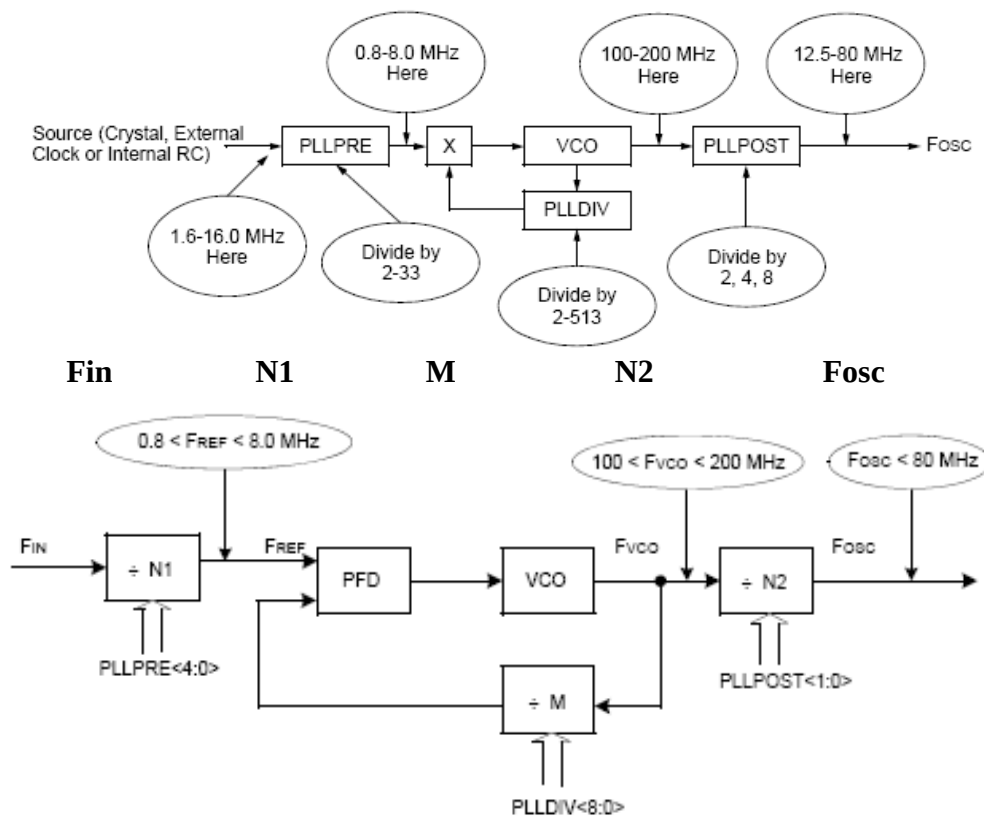
R = Readable bit W=Writable bit C=Clearable bit S=settable bit U = Unused bits, program to Logic '1'

TUN<5:0>: Ajusta frecuencia Oscilador FRC

011111 = Center frequency + 11.625% (8.23 MHz)
 011110 = Center frequency + 11.25% (8.20 MHz)
 ...
 000001 = Center frequency + 0.375% (7.40 MHz)
000000 = Center frequency (7.37 MHz nominal)
 111111 = Center frequency - 0.375% (7.345 MHz)
 ...
 100001 = Center frequency - 11.625% (6.52 MHz)
 100000 = Center frequency - 12% (6.49 MHz)

PHASE-LOCKED LOOP (PLL)

El oscilador principal y el FRC interno pueden usar el PLL para aumentar la frecuencia de la CPU. El PLL del micro se puede configurar para conseguir distintos comportamientos:



Fin: frecuencia entrada, **Fosc:** frecuencia salida, **PFD:** Phase Frequency Detector, **VCO:** Voltage Controlled Oscillator

Figura 1. Esquema del PLL del PIC24H

$$F_{OSC} = F_{IN} \times \left(\frac{M}{N1 \times N2} \right) = F_{IN} \times \left(\frac{(PLLDIV + 2)}{(PLLPRE + 2) \times 2(PLLPOST + 1)} \right)$$

$$N1 = PLLPRE + 2$$

$$N2 = 2 \times (PLLPOST + 1)$$

$$M = PLLDIV + 2$$

Por ejemplo, para conseguir un velocidad de ejecución de 40 MIPS, hay que trabajar con una frecuencia de oscilación, F_{osc} , de 80 MHz. Si la frecuencia de entrada F_{in} es 8 MHz, podría obtenerse ese valor con los siguientes valores para M, N1 y N2: $80 = 8 * 40 / (2 * 2)$.

Valores asociados al PLL en el encendido (POR = *Power On Reset*)

Register	Bit	Field	Value
CLKDIV<4:0>	PLLPRE<4:0>	00	N1 = 2
CLKDIV<7:6>	PLLPOST<1:0>	01	N2 = 4
PLLFBD<8:0>	PLLDIV<8:0>	000110000	M = 50

Con lo que obtenemos que: $F_{ref} = F_{in} (1/N1) = 0.5F_{in}$
 $F_{vco} = F_{in} (M/N1) = F_{in} (50/2) = 25F_{in}$
 $F_{osc} = F_{in} (M/(N1*N2)) = F_{in} (50/(2*4)) = 6,25F_{in}$

Dado que debe cumplirse que $100 \text{ MHz} < F_{vco} < 200 \text{ MHz}$, se concluye que para trabajar con esos valores, la frecuencia de entrada al módulo PLL (F_{in}) debe cumplir $4 \text{ MHz} < F_{in} < 8 \text{ MHz}$.

Al trabajar con la Explorer 16, que tiene un oscilador de 8 MHz, podemos trabajar sin PLL a esa frecuencia, o podemos utilizar el PLL para conseguir una frecuencia entre 8 y 80 MHz.

Para configurar el reloj a 8 MHz sin PLL, incluiremos en el programa principal las siguientes líneas:

```
# pragma config FNOSC=PRIPLL    //Primary oscillator (XT, HS, EC) w/PLL
# pragma config FCKSM=CSDCMD    //Clock switching and clock monitor disabled
# pragma config OSCIOFNC=OFF    //OSC2 is clock
# pragma config POSCMD=XT       //XT oscillator
```

Para configurar el reloj en modo seguro a una frecuencia hasta los 80 MHz, mediante el oscilador de 8 MHz de la Explorer 16, seguiremos los siguientes pasos.

```
# pragma config FNOSC=PRIPLL    //Primary oscillator (XT, HS, EC) w/PLL
# pragma config FCKSM=CSECMD
// Clock switching enabled clock monitor disabled
# pragma config OSCIOFNC=OFF    //OSC2 is clock
# pragma config POSCMD=XT       //XT oscillator
// Oscilador principal (Posc) en modo XT
// Configurar PLL prescaler, PLL postsacler y divisor PLL del siguiente modo
// Frec=Fosc*M(N1*N2). En nuestro caso Fosc=8
// Dependiendo de los valores de M, N2 y N1 obtendremos una frecuencia u otra
// Hay que tener en cuenta las restricciones de la figura 1 (esquema del PLL)
// con respecto a los valores de las frecuencias intermedias: Fref y Fvco
#define PLL_M xx
#define PLL_N2 xx
#define PLL_N1 xx

PLLFBD = PLL_M;                // M =
CLKDIVbits.PLLPOST = PLL_N2;   // N2 =
CLKDIVbits.PLLPRE = PLL_N1;    // N1 =

// Tras esas inicializaciones hay que cambiar el reloj y hacer que sea el
// oscilador primario con el PLL (NOSC = 0b011)
```

```

__builtin_write_OSCCONH(0x03);
__builtin_write_OSCCONL(0x01);

    // Las dos líneas anteriores son equivalentes a: OSCCON= 0x0301;
while (OSCCONbits.COSC! = 0b011); // Esperar que finalice el cambio de reloj
while(OSCCONbits.LOCK! = 1) ;      // Esperar hasta que el PLL esté listo

// A partir de este momento la frecuencia de reloj del micro será la que
// hayamos seleccionado en funcion de los valores M, N1 y N2

```

Cuando la frecuencia de entrada no está en el rango de 4 a 8 MHz, es posible utilizar el PLL haciendo lo siguiente:

1. Alimentar el dispositivo con el oscilador interno FRC o POSC, sin PLL
2. Modificar los valores de PLLDIV, PLLPRE y PLLPOST teniendo en cuenta la frecuencia de entrada, para conseguir que se cumplan las siguientes restricciones:
 - F_{REF} esté entre 0,8 y 8,0 MHz
 - F_{VCO} esté entre 100 y 200 MHz
3. Modificar el reloj por software a uno de los siguientes modos del PLL:
 - Modo XT, el oscilador POSC puede estar entre 3-10 MHz
 - Modo HS, el oscilador POSC puede estar entre 10-40 MHz
 - Modo EC, el oscilador POSC puede estar entre 0,8-64 MHz