

# *CLOCK GENERATION MODULE (CGM)*

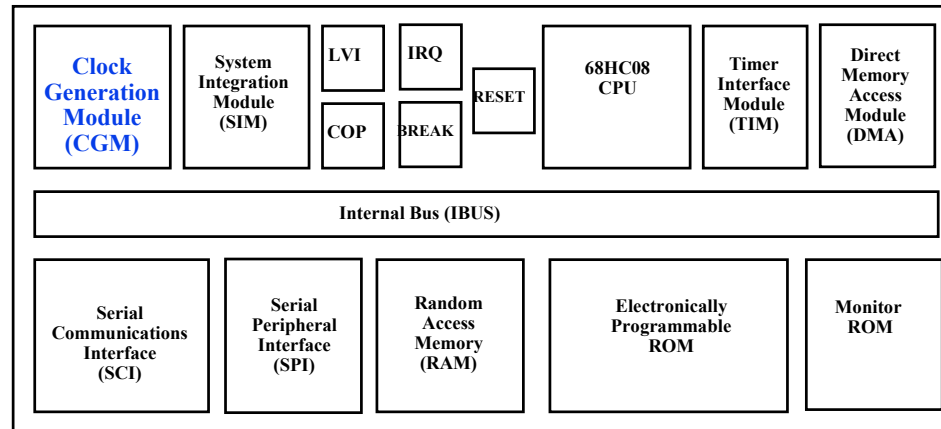
*“Módulo de Generación  
de Clock”*

*Curso de Microcontroladores  
Familia HC908 Flash de Motorola*

*Parte II*

ING. DANIEL DI LELLA DDFAE For Motorola Products

## *Clock Generation Module*



Genera la frecuencia de Clock base para el MPU

Dos fuentes de clock seleccionables (en el GP32, ver otros derivados):

- **Oscilador a Cristal**
- **Phase Lock Loop (PLL)** (Oscilador por enganche de fase)
  - Usa un oscilador a cristal de baja frecuencia para producir frecuencias de clock más altas.

La salida es usada por el “**S**ystem **I**ntegration **M**odule” **SIM** .

Todos los sub sistemas usan el “system clock” para “timing”

- Excepto el “**S**erial **C**ommunications **I**nterface **M**odule” **SCI**
  - Usa la frecuencia de oscilador del cristal **CGMXCLK** o la del Bus .

*Curso de Microcontroladores*

*Familia HC908 Flash de Motorola*

*Parte II*

ING. DANIEL DI LELLA DDFAE For Motorola Products



## *Oscilador a Cristal*

Un cristal externo es conectado entre **OSC1** y **OSC2**

**CGMXCLK** es la salida del oscilador de clock

- Es igual a la frecuencia externa del cristal
- **4 X** de la frecuencia interna del Bus (si es seleccionada)

CUANDO SE USA EL VCO PARA GENERAR **SYSCLK**, EL CRISTAL ES:

- “Bufferiado” para producir **CGMRCLK**
  - Referencia de frecuencia para el circuito del PLL

El “SIM module” puede habilitar/deshabilitar el oscilador via

la linea **SIMOSCEN**

- Detiene el oscilador a cristal y el módulo del PLL

## *Módulo PLL*

El PLL consiste de:

- Oscilador Controlado por Tensión de ancho de banda programable (VCO)
- Módulo divisor de frecuencia del VCO programable
- Detector de fase
- Filtro de Lazo
- Detector de Enclavamiento
- Opera en Modo “Adquisición” o Modo “Tracking”

**CGMVCLK** es la salida del “PLL module”

- Es el **exacto multiplo entero** de **CGMXCLK**
- **4 X** de la frecuencia interna del Bus (si es seleccionada)

## *VCO Programable*

Genera la señal **CGMVCLK** basado en:

- **Divisor de Frecuencia**
  - Divide frecuencia produciendo “clocks” más rápidos
  - Los bits de selección de multiplicación controlan división de Frecuencia
- **Detector de Fase**
  - Compara CGMXCLK y la salida del divisor de Frecuencia
  - Detecta errores de fase
  - Genera pulso de corrección
- **Filtro de Lazo**
  - Convierte el pulso de corrección del detector de fase en tensión de corrección para el VCO.

**Ancho de Banda programable mejora la inmunidad al ruido**

- Los Bits de rango del VCO determinan los limites del Ancho de Banda.

## Secuencia de programación del PLL

### Ejemplo de Uso

Meta : **8MHz** de frecuencia de Bus desde un cristal de **32,768 Khz** (baja frecuencia).

1. Seleccionar la frecuencia de Bus deseada,  $F_{BUSDES}$ .  $F_{BUSDES} = 8 \text{ MHz}$

2. Calcular la frecuencia de VCO deseada ( 4 veces la FBusDes ),  $F_{VCLKDES}$   
 $F_{VCLKDES} = 4 \times F_{BUSDES}$   $F_{VCLKDES} = 4 \times F_{BUSDES} = 4 \times 8 \text{ MHz} = 32 \text{ MHz}$

3. Elegir una frecuencia práctica de referencia del PLL ,  $F_{RCLK}$ .  $F_{RCLK} = 32,768\text{Khz}$

tipicamente usamos  $F_{rclk} = 32,768\text{Khz}$  y  $R = 1$  ( reference clock divider ) y  $P = 0$  (ver manual de datos)

4. Seleccionar un multiplicador de la frecuencia del VCO ,  $N$

$$N = F_{VCLKDES} / F_{RCLK} \text{ ( el entero positivo más cercano )}$$

$$N = F_{VCLKDES} / F_{RCLK} = 32 \text{ MHz} / 32,768\text{kHz} = 977 \text{ ( 3D1 Hexa )}$$

5. Calcular y verificar la adecuación de las frecuencias del VCO y el Bus ,  $F_{VCLK}$  y  $F_{BUS}$ .

$$F_{VCLK} = N \times F_{RCLK}$$

$$F_{VCLK} = N \times F_{RCLK} = 977 \times 32,768 \text{ Khz} = 32,014336 \text{ MHz}$$

$$F_{BUS} = (F_{VCLK}) / 4$$

$$F_{bus} = 8,003584 \text{ Mhz}$$

6. Seleccionar un multiplicador de rango( pot. De 2 ) del VCO , E.

De acuerdo a tabla ( ver manual ) para  $F_{vclk} = 32 \text{ MHz}$   $E = 2$

## *Secuencia de programación del PLL Cont.*

7. Seleccionar un multiplicador de rango lineal del VCO, **L**.  $L = \text{Redondeo} ( F_{vclk} / 4 \times F_{NOM} ) = 32,014336\text{MHz} / 4 \times 38,4\text{KHz} = 208,4266 = \mathbf{208 ( D0 hexa )} = L$

donde  $F_{NOM}$  = frecuencia nominal VCO = 38,4 KHz @ 5.0 Volts

8. Calcular y verificar la frecuencia de centro del rango del VCO,  $F_{VRS}$ .

$$F_{VRS} = L \times 2^E \times F_{NOM} \quad F_{VRS} = L \times 4 \times F_{NOM} = 208 \times 4 \times 38,4 \text{ KHz} = \mathbf{31,9488 \text{ MHz}}$$

$$[ F_{VRS} - F_{vclk} ] < 0 = F_{NOM} \times 2^E / 2 \quad \mathbf{\text{cumple !!!}}$$

### **NOTA:**

Si se exceden los valores máximos de la frecuencia del Bus y del VCO puede causar un **“crash”** (no funcionamiento) en el MCU !!! . El circuito de PLL posee poco margen de **“overclocking”** (excederse del límite de frecuencia de clock), ya que su principal limitación pasa por la frecuencia máxima que puede generar el VCO dentro del PLL. No ocurre lo mismo cuando se utiliza el oscilador a cristal externo (OSC1 / OSC2) ya que puede soportar bien el **“overclocking”** , pero se debe ser conciente que se está fuera de la curva estadística de garantía de buen funcionamiento.

## *Programando el PLL*

Programar los registros del PLL según:

- A) En el bit “PRE” del PLL Control Register ( PCTL ), programar el equivalente binario de “P”.
- B) En el bit “VPR” del PLL Control Register ( PCTL ), programar el equivalente binario de “E”.
- C) En el “PLL multiplier select register Low y High” ( PMSL ) y ( PMSH ), programar el equivalente binario de “N”.
- D) En el PLL range select register ( PMRS ), programar el equivalente binario de “L”.
- E) En el PLL reference divider select register ( PMDS ), programar el equivalente binario de “R”.

## *REGISTROS DEL PLL*

**PCTL**

<b>READ:</b>	PLLIE	PLLF	PLLON	BCS	PRE1	PRE0	VPR1	VPR0
<b>WRITE:</b>								
<b>RESET:</b>	0	0	1	0	0	0	0	0

**PMSH**

<b>READ:</b>	0	0	0	0	MUL11	MUL10	MUL9	MUL8
<b>WRITE:</b>								
<b>RESET:</b>	0	0	0	0	0	0	0	0

**PMSL**

<b>READ:</b>	MUL7	MUL6	MUL5	MUL4	MUL3	MUL2	MUL1	MUL0
<b>WRITE:</b>								
<b>RESET:</b>	0	1	0	0	0	0	0	0

## *REGISTROS DEL PLL Cont.*

PMRS

<b>READ:</b>	VRS7	VRS6	VRS5	VRS4	VRS3	VRS2	VRS1	VRS0
<b>WRITE:</b>								
<b>RESET:</b>	0	1	0	0	0	0	0	0

PMDS

<b>READ:</b>	0	0	0	0	RDS3	RDS2	RDS1	RDS0
<b>WRITE:</b>								
<b>RESET:</b>	0	0	0	0	0	0	0	1

Estos valores son los que deberían cargarse en los correspondientes registros del PLL para obtener una Fbus = 8.0Mhz, según el ejemplo explicado anteriormente.

*(Ver rutina de ejemplo al final del cap.)*

Tabla resumen:

- Fbus = 8.0Mhz
- FRCLK = 32.768 KHz
- R = 1
- N = 3D1
- P = 0
- E = 2
- L = D0

## *Información Adicional - Adquisición vs Tracking -*

### **Modo Adquisición**

- Permite grandes correcciones de frecuencia en el “PLL module”.
- La frecuencia no está fija ( **not locked** ).

### **Modo Tracking**

- La frecuencia del VCO está cerca de la frecuencia programada, se requieren solo pequeñas correcciones.
- La frecuencia está fija ( **locked** ).

Al iniciar el funcionamiento del módulo de PLL, el mismo comienza en el “modo Adquisición”, ya que la frecuencia del lazo no está estabilizada y el sistema tiende a alcanzar la frecuencia final por medio de grandes correcciones en frecuencia.

Una vez alcanzada la frecuencia final, y dentro de una “banda de error” típica de estos sistemas realimentados digitalmente, el PLL pasa a funcionar en “modo tracking” pues las compensaciones en frecuencia son mínimas y a los efectos prácticos, puede considerarse la frecuencia de salida como “FIJA”.

## *Información Adicional*

### *- Auto vs Manual -*

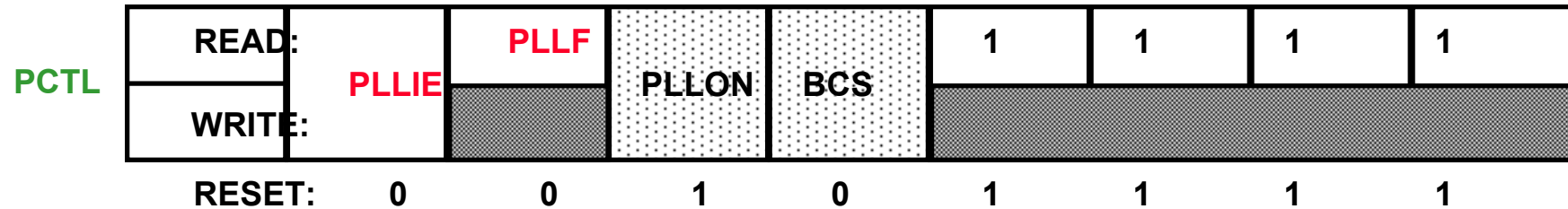


#### **PLL Bandwidth Control Register (PBWC)**

- **AUTO - Automatic Bandwidth Control (Acquisition vs Tracking)**  
 1 = Automatic bandwidth control  
 0 = Manual bandwidth control
- **LOCK - Lock Indicator ( se auto setea )**  
 1 = frecuencia del VCO correcta o fijada (locked)  
 0 = frecuencia del VCO incorrecta o no fija (unlocked)
- **ACQ – Acquisition** 1 = Tracking mode  
 0 = Acquisition mode

Quando se coloca al PLL en modo “AUTO”, el mismo selecciona en forma automática el modo de seguimiento de la frecuencia (adquisition vs tracking), de esta forma, ante eventuales “desenganches” del PLL el sistema tenderá a autocorregirse e indicará esta situación por medio de “flags” especiales y una interrupción específica, si esta fuera habilitada.

## *Información Adicional - Interrupciones -*



### PLL Control Register(PCTL)

- **PLL Interrupt Enable (PLLIE)**
  - Habilita la interrupción al CPU cuando el bit “LOCK” cambia.
  - Deshabilitada cuando está en modo manual !!!
- **PLL Interrupt Flag (PLLIF)**
  - Pasa a “1” cuando el PLL entra o sale del estado “locked”.

## *Información Adicional - Modos Low Power -*

### Low Power Modes

- **WAIT**
  - No afecta al CGMC
- **STOP**
  - Deshabilita al CGMC poniendo la línea **SIMOSCEN** en Low (bit **OSCSTOPENB** en el CONFIG2 register)
  - Deshabilita el VCO
  - Limpia el BCS bit
- **PLL Desengage Mode (alternativa low power)**
  - Limpia los bits BCS y PLLON
    - Desengancha al PLL sin apagarlo
    - Queda preparado para cuando vuelve de un WAIT.

## RUTINA DE EJEMPLO

```
=====
;::::::::: CODIGO DE PRUEBA DEL PLL          ::::::
=====
; RESET:
; 1) Setea frecuencia de BUS= 8MHz mediante el PLL con cristal de
;    32.768KHz.
=====
RESET:
;1)
    rsp                ;SP=$00FF
;setea PLL para generar 8 MHZ de frec.BUS desde cristal de 32K768
;R=1    N=3D1    P=0    E=2    L=D0
;
clr     PCTL          ;PLL:OFF
bset   1,PCTL        ;P= 0  E= 2
mov    #3,PMSH       ;N HI= 3
mov    #1,PMDS       ;R= 1 (default)
mov    #$D1,PMSL     ;N LOW= D1
mov    #$D0,PMRS     ;L= D0
bset   PCTL_B_PLLON,PCTL ;PLL:ON
bset   PBWC_B_AUTO,PBWC ;enganche automatico
wait_pll:                ;esperar que LOCK sea = 1
brclr  PBWC_B_LOCK,PBWC,wait_pll;
bset   PCTL_B_BCS,PCTL  ;cambia a frecuencia del PLL
```

**FIN CAPITULO 7 !!!**