

### 3.1 MÓDULO DE GENERACIÓN DE RELOJ – CGMC GENERACIÓN DE FRECUENCIA DE BUS CON EL MÓDULO CGMC (SOLO GP32)

Preparado por: Rangel Alvarado  
Estudiante Graduando de Lic. en Ing. Electromecánica  
Universidad Tecnológica de Panamá  
Panamá, Panamá  
“e-mail”: [issaiass@cwpanama.net](mailto:issaiass@cwpanama.net)  
“web site”: <http://www.geocities.com/issaiass/>

#### ÍNDICE

3.1.1	Introducción	321
3.1.2	Lazos Enganchados de Fase - PLL	322
3.1.3	Registros de Estado y Control del Módulo de Reloj	322
3.1.4	Diagrama de Flujo	325
3.1.5	Código	327
3.1.6	Simulación	333
3.1.7	Conclusión	334
3.1.8	Referencias	334
3.1.9	Problemas Propuestos	334

#### 3.1.1 Introducción

---

La velocidad del bus es limitada por el tamaño del cristal externo u oscilador que se fije a las terminales del microcontrolador. Anteriormente se informó que la velocidad de bus del microcontrolador es la cuarta parte del cristal externo; aunque parezca poco creíble, nuestra pastilla puede correr a velocidades de bus diferentes independientemente de la velocidad del cristal que se sitúe en las terminales de oscilación, todo esto gracias a un módulo especial que posee un “PLL”<sup>1</sup> y permite generar frecuencias múltiplos del cristal.

En principio, se explica a grandes rasgos que función tiene un PLL en general y en detalle:

- Cálculo de la frecuencia de bus: En la configuración del PLL para GP32 se analizan los parámetros que deben calcularse para su buen desempeño.
- Configuración del Módulo de Generación de Reloj: Inicializar el módulo para generar una frecuencia máxima de bus, dependiendo del cristal de entrada.
- Código: Muestra el listado del programa de configuración del PLL específico para un cristal de 4.9152 MHz; la función del código es conmutar un LED con un retardo de “software” a 50 Hz configurada una frecuencia de bus máxima.
- Simulación: Corroborar de manera gráfica, cómo varían los registros del módulo de reloj y deducir que acción retoma el microcontrolador, configurado estos registros; finalmente, se utiliza el PLL para generar una frecuencia de bus y un retardo de “software” para generar una frecuencia de cincuenta hertz (50 Hz).

---

<sup>1</sup> Lazos Enganchados de Fase o “Phase Locked Loop”

### 3.1.2 Lazos Enganchados de Fase - PLL

Un PLL es un circuito que permite que una señal de referencia externa controla la frecuencia y la fase de un oscilador, el mismo, detecta la señal de entrada, la filtra y compara con la señal de oscilación ajustando así la diferencia de fase (señal desplazada).

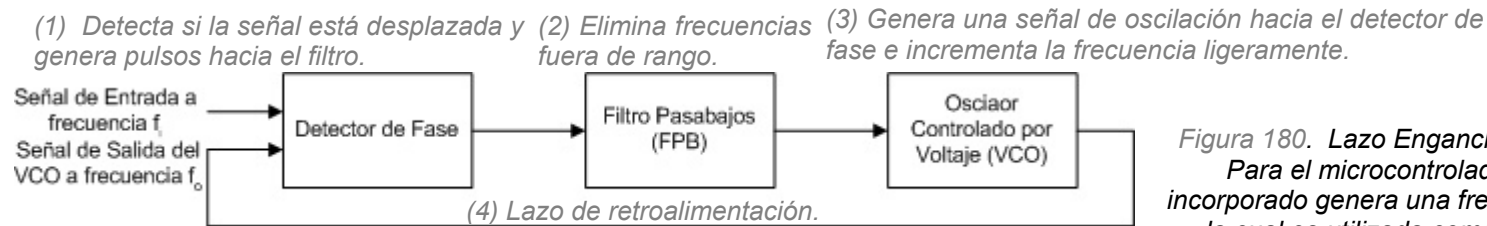


Figura 180. Lazo Enganchado de Fase o PLL. Para el microcontrolador GP32, el PLL incorporado genera una frecuencia de oscilación la cual es utilizada como reloj del sistema.

### 3.1.3 Registros de Estado y Control del Módulo de Reloj

#### 3.1.3.1 Registros de Control del PLL

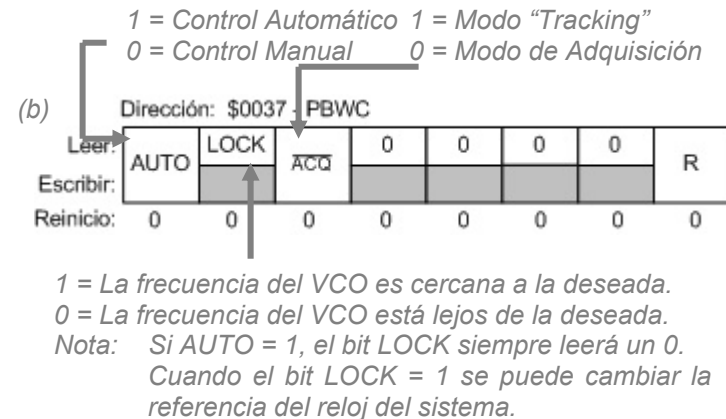
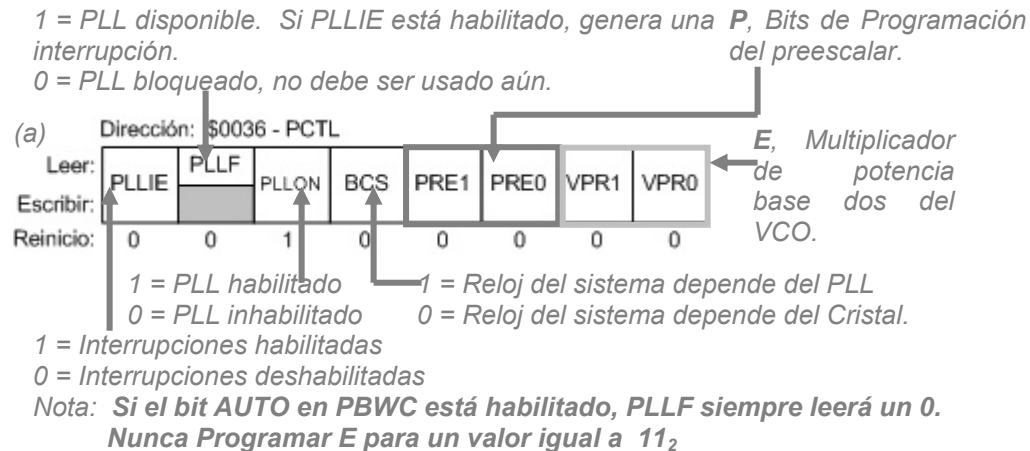


Figura 181. Registros de Estado y Control del Módulo de Reloj. (a) Registro de Control del PLL. Controla el estado del PLL, interrupciones, activación y variables P, E. (b) Registro de Control del Ancho de Banda del PLL. Configura el PLL para modo automático, define cuando esta listo y su modo de operación entre adquisición o "tracking".

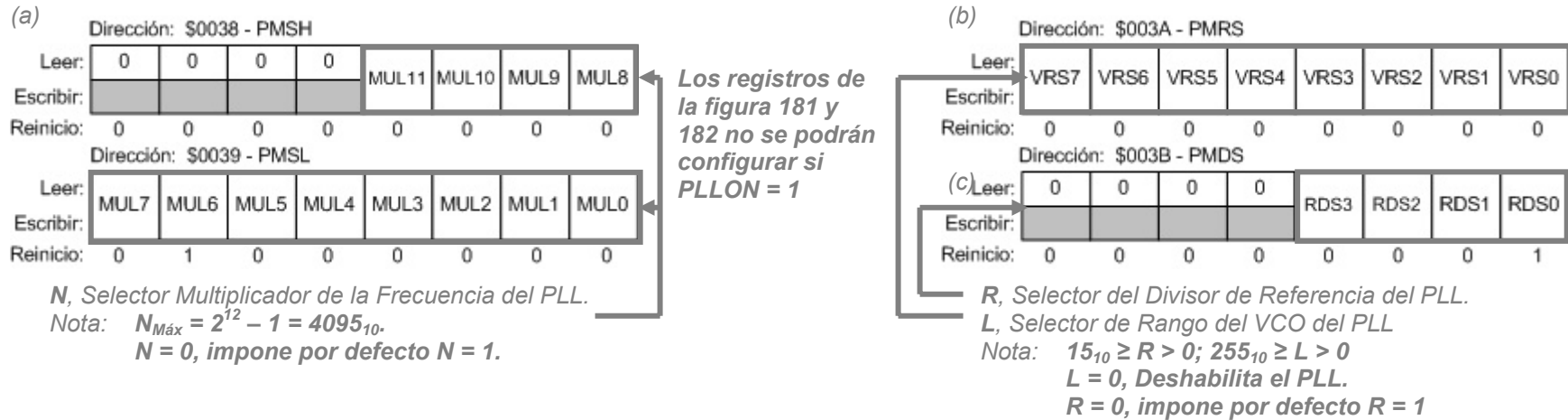


Figura 182. Registros de Estado y Control del Módulo de Reloj (Continuación). (a) Registros Selectores de Multiplicador de la Frecuencia del PLL. Controlan el Módulo del divisor de la retroalimentación del VCO. (b) Registro Selector de Rango del VCO del PLL. Controla el rango lineal en el cual opera el detector de fase. (c) Selector del Divisor de Referencia del PLL. Igualmente, este registro controla el rango lineal.

### 3.1.3.2 Pasos para la Programación del PLL del GP32

Tabla 69. Parámetros de Configuración del PLL

Valor actual de $N$	$P$
$0 < N \leq N_{Máx}$	0
$N_{Máx} < N \leq N_{Máx} \times 2$	1
$N_{Máx} \times 2 < N \leq N_{Máx} \times 4$	2
$N_{Máx} \times 4 < N \leq N_{Máx} \times 8$	3

(a)  $P$  = Registro PCTL\_PRE[1:0]

Rango de Frecuencias [MHz]	$E$
$0 < f_{VCLK} < 9.8304$	0
$9.8304 \leq f_{VCLK} < 19.6608$	1
$19.6608 \leq f_{VCLK} < 39.3216$	2

(b)  $E$  = Registro PCTL\_VPR[1:0]

1. Escoger una frecuencia de bus deseada menor o igual a ocho Megahertz (8 MHz),  $f_{BUSDES}$ .
2. Calcular la frecuencia del VCO,

$$f_{VCLKDES} = 4 \times f_{BUSDES}$$

Ecuación 8. Frecuencia Deseada de Oscilación

3. Apuntar el Cristal de referencia

$$f_{RCLK} = f_{XTAL}$$

Ecuación 9. Cristal de Referencia PLL

4. Si,  $30 \text{ kHz} \leq f_{RCLK} \leq 100 \text{ kHz}$ , escoger  $R = 1$ , de lo contrario

$$R = \text{round} \left[ R_{Máx} \times \left\{ \left( \frac{f_{VCLK}}{f_{RCLK}} \right) - \text{int} \left( \frac{f_{VCLK}}{f_{RCLK}} \right) \right\} \right]$$

Ecuación 10. Selector Divisor de Referencia del PLL

5. Calcular N para extraer P de la tabla 61(a).

$$N = \text{round}\left(\frac{R \times f_{VCLKDES}}{f_{RCLK}}\right)$$

Ecuación 11. Multiplicador de Referencia

6. Calcular la frecuencia del VCO y del bus:

$$f_{VCLK} = 2^P \times N \times f_{RCLK} / R$$

Ecuación 12. Frecuencia del VCO

$$f_{BUS} = f_{VCLK} / 4$$

Ecuación 13. Frecuencia de Bus

7. Seleccionar un valor de E de la tabla 69(b).

8. Calcular:

$$L = \text{round}\left(\frac{f_{VCLK}}{2^E f_{NOM}}\right)$$

Ecuación 14. Selector de Rango del VCO

En donde  $f_{NOM} = 38.4$  kHz.

9. Recalcular:

$$f_{VRS} = (L \times 2^E) \cdot f_{NOM}$$

Ecuación 15. Verificación del VCO

10. Verificar:

$$|f_{VRS} - f_{VCLK}| \leq f_{NOM} \times 2^{E-1}$$

Ecuación 16. Verificación de Parámetros del PLL

### 3.1.3.3 Ejemplo de Programación y Configuración del PLL

Sea un Cristal Externo de 4.9152 MHz y la frecuencia de bus de 8 MHz, configure el PLL.

- |                             |                              |                             |                     |
|-----------------------------|------------------------------|-----------------------------|---------------------|
| 1. $f_{BUSDES} = 8$ MHz.    | 4. $R = 8$ .                 | 7. $E = 2$ .                | 10. $0 \leq 76.8$ k |
| 2. $f_{VCLKDES} = 32$ MHz.  | 5. $P = 0$ .                 | 8. $L = D0_{16}$            |                     |
| 3. $f_{RCLK} = 4.9152$ MHz. | 6. $f_{VCLK} = 31.9488$ MHz. | 9. $f_{VRS} = 31.9488$ MHz. |                     |

Tabla 70. Parámetros de Configuración para Frecuencias de Bus

$f_{BUS}$ [MHz]	P	E	N	L	R
2.0	0	0	F	D5	9
2.5	0	1	2	80	1
4.0	0	1	D	D0	4
4.9152	0	2	4	80	1
7.3728	0	2	6	C0	1
8	0	2	34	D0	8

(a)  $f_{XTAL} = 4.9152$  MHz

$f_{BUS}$ [MHz]	P	E	N	L	R
2.0	0	0	F	D5	9
2.5	0	1	1	80	1
4.0	0	1	F	D5	9
4.9152	0	2	2	80	1
7.3728	0	2	3	C0	1
8	0	2	D	D0	4

(b)  $f_{XTAL} = 9.8304$  MHz

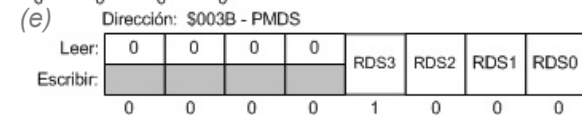
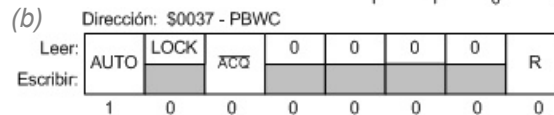
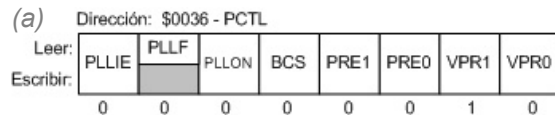
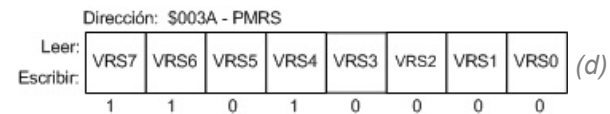
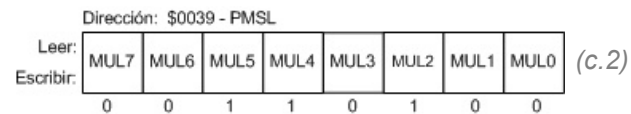
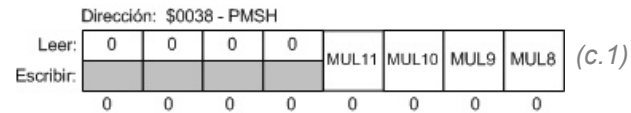


Figura 183. Configuración del PLL a frecuencia máxima de Bus con Cristal de 4.9152 MHz. (a) PCTL. Inhabilitar el PLL y luego configurar P y E. (b) PBWC. Habilitar el modo automático. (c) PSM[H:L]. (c.1) y (c.2); configurar N. (d) PMRS. Configurar L. (e) PMDS. Configurar R y luego, habilitar el PLL en el registro PCTL, PLLON = 1.

### 3.1.2.4 Vector de Interrupción del Módulo de Generación de Reloj

Tabla 71. Vector de Interrupción del Módulo PLL

Bandera	Máscara	Dirección	Vector (Dirección)
PLLIF	AUTO, PLLIE	FFF2	Vector del PLL (Alto)
		FFF3	Vector del PLL (Bajo)

Si el bit AUTO está en 1 y las interrupciones están habilitadas (PLLIE = 1) el CPU generará interrupciones si el bit LOCK cambia de estado. Además, cuando el bit AUTO, esté en 0, las interrupciones estarán deshabilitadas y el bit LOCK leerá siempre 0.

### 3.1.4 Diagrama de Flujo

El programa configura el microcontrolador GP32, para que se genere una frecuencia máxima de bus con un cristal externo de 4.9152 MHz y utilizar un retardo de “software” para conmutar el LED PTA6 a una frecuencia de cincuenta hertz (50 Hz).

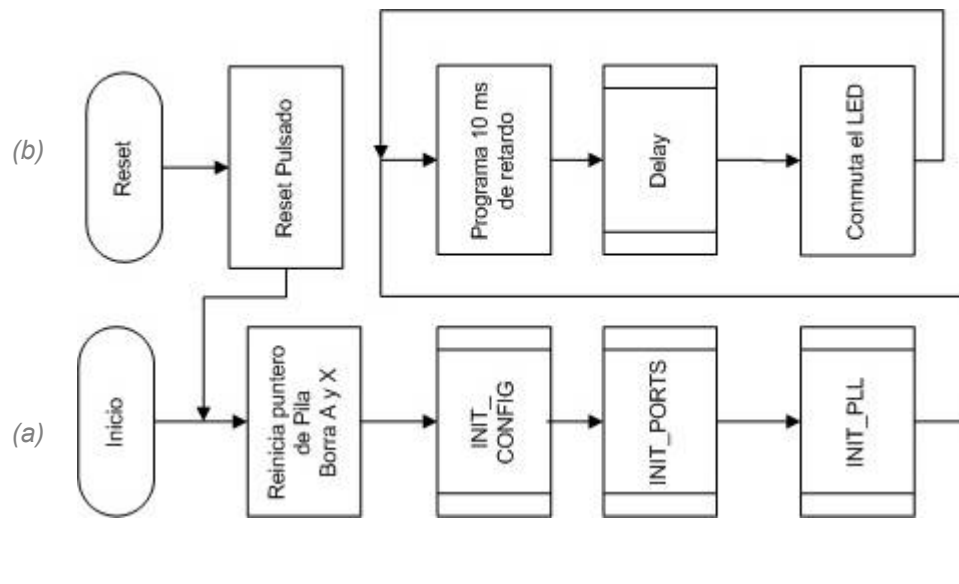
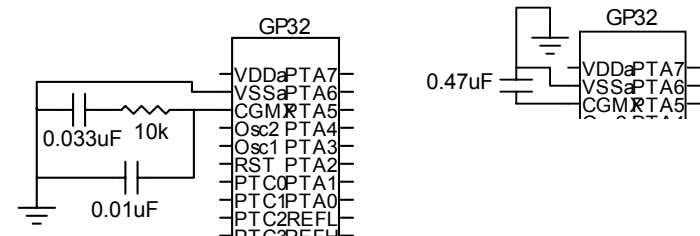


Figura 184. NT1001 – CGM. (a) Programa principal. Configura el Módulo CGMC para generar una frecuencia máxima de bus basada en un cristal de 4.9152 MHz y conmuta un LED a 50 Hz. (b) Reinicio del Sistema. Al presionar “reset”, reinicia el sistema.

Nota: Para que el PLL funcione, se necesita realizar la siguiente conexión adicional de componentes pasivos, elija cualquiera de las dos configuraciones.



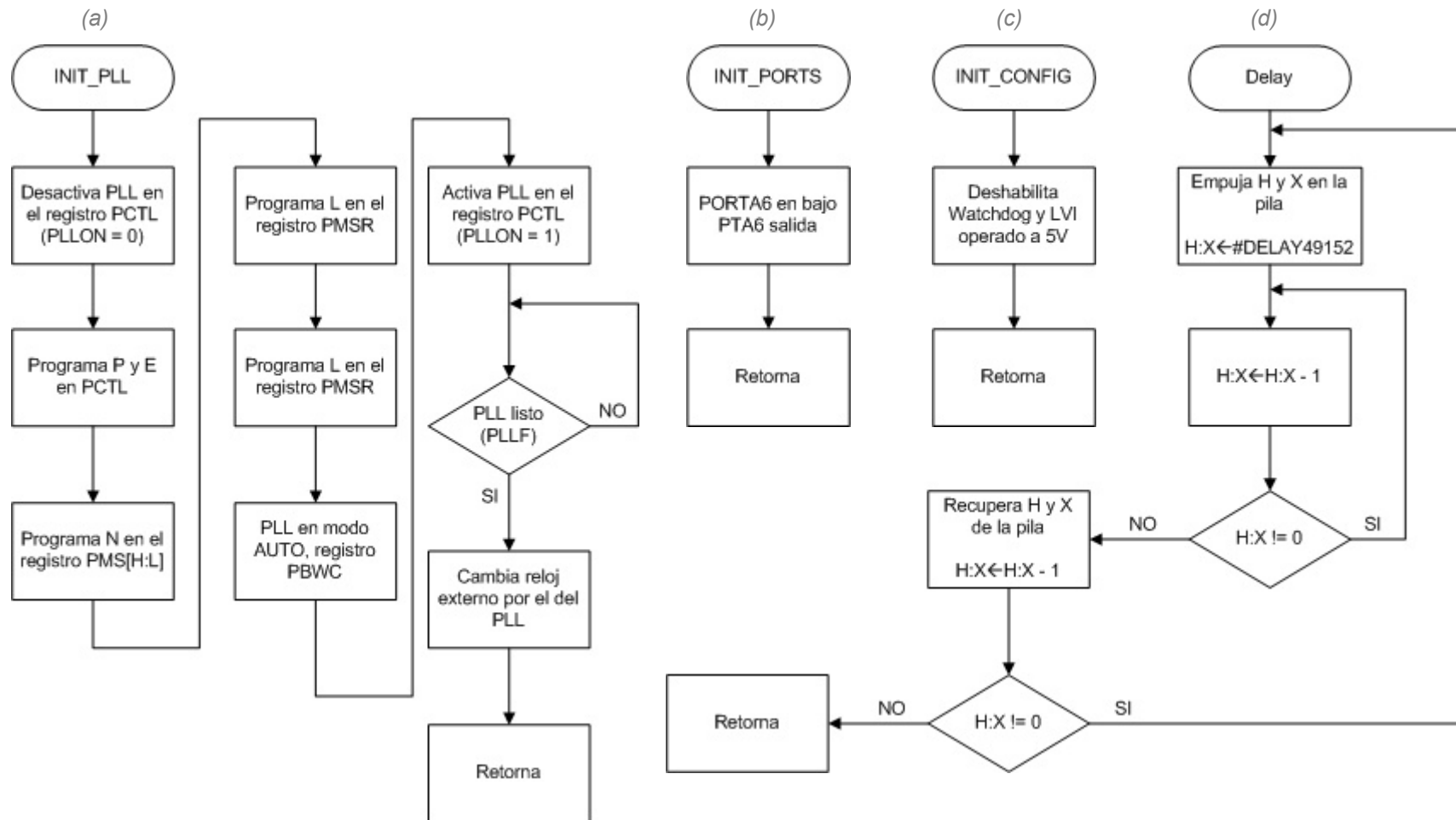


Figura 185. NT1001 – CGMC (continuación, subrutinas). (a) INIT\_PLL. Configura el módulo de generación de reloj para que en base a un cristal de 4.9152 MHz genere una frecuencia de bus máxima. (b) INIT\_PORTS. Inicializa el puerto A6 en bajo y luego lo ubica como salida para el LED. (b) INIT\_ADC. Configura para conversión en el puerto A5. (c) INIT\_CONFIG. Inicializa el registro de configuración 1 para inhabilitar el “watchdog” y hacer que el LVI trabaje a 5V. (d) Delay. Subrutina de retardo programable de un milisegundo (1 ms) de base de tiempo.



```

; Pasos para programar el PLL.
;
;       1 - Desactivar el PLL, PCTL_PLLON = 0
;
;       2 - ¿Necesito Interrupciones?, Programar P & E, PCTL_BCS = 0
;
;       3 - Programar N, PMSH[3:0] + PMSL
;
;
;       4 - Programar L, PMRS = L
;
;       5 - Programar R, PMDS = R
;
;       6 - Seleccionar modo automático (recomendado), PBWC_AUTO = 1
;
;       7 - Activar el PLL, PCTL_PLLON = 1
;
;       8 - Esperar la interrupción o que la bandera PLLF = 1
;
;       9 - Cambiar a la referencia interna del VCO, PCTL_BCS = 1
;
=====
$SET   ICS08                               ; ICS08 = 1, Vamos a simular en la pastilla
;                                           ; la velocidad de simulación es menor en la
;                                           ; PC.
;
; $SETNOT ICS08                            ; ICS08 = 0, Vamos a programar la pastilla
;                                           ; la aplicación debe correr en tiempo real
;
=====
;
;       Definiciones del Usuario
;
=====
DELAY32    equ $03E4                      ; Retardo para 8 MHz de Bus
PTA6       equ 6T                          ; Puerto A, Bit 6
DDRA6      equ 6T                          ; Registro de Direccionamiento A, Bit 6
LVI5OR3    equ %00001000                  ; CONFIG1, Bit de Operación del LVI, LVI @
;                                           ; 5V
COPD       equ %00000001                  ; CONFIG1, Bit de Inhabilitación de Watchdog,
;                                           ; BIT 0 ON
PLLIE      equ %10000000                  ; PCTL, Bit de Habilitación de interrupciones,
;                                           ; Bit 7 ON
PLLF       equ 6T                          ; PCTL, Bandera de Interrupciones
;                                           ; pendientes, Bit 6
PLLON      equ 5T                          ; PCTL, Bandera de activación del PLL, por
;                                           ; defecto en alto
BCS        equ 4T                          ; PCTL, Seleccionador del Reloj Base, Bit 4
PRE1       equ 0T                          ; PCTL, Divisor de la Frecuencia a la potencia
;                                           ; de Dos, Bit 3 OFF
PRE0       equ 0T                          ; PCTL, Divisor de la Frecuencia a la potencia
;                                           ; de dos, Bit 2 OFF
P          equ PRE1|PRE0                  ; PCTL_PRE[1:0] = P, Divisor de la frecuencia
;                                           ; a la potencia de dos
VPR1       equ %00000000                  ; PCTL, Multiplicador a la potencia de dos del
;                                           ; VCO, Bit 1 ON
VPR0       equ %00000010                  ; PCTL, Multiplicador a la potencia de dos del
;                                           ; VCO, Bit 2 OFF
E          equ VPR1|VPR0                  ; PCTL_VPR[1:0] = E, Multiplicador a la
;                                           ; potencia de dos del VCO
AUTO       equ 7T                          ; PBWC, Habilitador del Modo Automático de
;                                           ; Control del Ancho de Banda, Bit 7 ON
LOCK       equ 6T                          ; PBWC, Bandera de Aviso del PLL
;                                           ; Inicializado (Cuando AUTO = 1 y LOCK = 1)
;                                           ; el PLL corre a la frecuencia programada
MUL11_8    equ $0                          ; PMSH, Multiplicador de la frecuencia del

```

NT1001

Rev. 1 del 07.08.05



```

; VCO, Bits OFF
MUL7_0    equ $34          ; PMSL, Multiplicador de la frecuencia del
; VCO, valor de 7
N         equ MUL11_8|MUL7_0 ; PMS[H:L] = N, Multiplicador de la frecuencia
; del VCO

VRS7_0    equ $D0          ; PMRS, Selector de Rango del VCO
L         equ VRS7_0       ; PMRS = L, Selector de Rango del VCO
RDS3_0    equ $08          ; PMDS, Selector de Divisor de Referencia,
; por defecto, siempre 1
R         equ RDS3_0       ; PMDS = R, Selector de Divisor de
; Referencia

```

```

;=====
;
;           Mapa de Memoria del Microcontrolador
;=====
;=====
;=====
;
;           Registro de E/S
;=====

```

```

PORTA     equ $0000        ; Dirección, Puerto 0
DDRA      equ $0004        ; Registro de Direccionamiento, Puerto 0

```

```

;=====
;
;           Módulo Generador de Reloj
;=====
;=====

```

```

PCTL      equ $0036        ; Registro de Control del PLL, también
; parámetros P & E
PBWC      equ $0037        ; Registro de Control del Ancho de Banda del
; PLL
PMSH      equ $0038        ; Registro Selector del Multiplicador N (Alto)
PMSL      equ $0039        ; Registro Selector del Multiplicador N (Bajo)
PMRS      equ $003A        ; Registro Selector del Rango Lineal L
PMDS      equ $003B        ; Registro Selector del Divisor de Referencia R

```

```

;=====
;
;           Registro de Configuraciones
;=====
;=====

```

```

CONFIG1   equ $001F        ; Vector de configuración 1

```

```

;=====
;
;           Memoria FLASH
;=====
;=====

```

```

FLASH_START equ $8000      ; Puntero - Mem.FLASH

```

```

;=====
;
;           Vectores de Usuario
;=====
;=====

```

```

PLLH      equ $FFF8        ; Vector de Módulo de Lazo de Fase Cerrada
RESET_VEC equ $FFFE        ; Puntero del RESET

```

```

;=====
; OBJETIVO   : Inicio de Codif. del Ensam-
;              blador en Memoria FLASH.
;=====
                org FLASH_START                ; Inicio Mem. FLASH

;=====
; OBJETIVO   : Inicializa el PLL para operar
;              a la máxima frecuencia de bus
;              (8 MHz).
;              Prende y apaga el LED cada
;              10 ms, lo que causa una fre-
;              cuencia de 50 Hz
;=====
START
    rsp                    ; Inic.Stack = $00ff
    clra                   ; Borra A
    clrx                   ; Borra X
    jsr INIT_CONFIG       ; Inicializa el módulo de Configuración
    jsr INIT_PORTS        ; Inicializa Puertos
    jsr INIT_PLL          ; Inicializa Módulo Fase de Lazo Cerrado
ESPERA
    ldhx #10T             ; A retardar 10 ms
$IFNOT ICS08              ; Solo retarda si se está en modo de
                          ; programación
    jsr Delay             ; Retarda
$ENDIF
    com PORTA             ; Conmuta el puerto (50 Hz)
    bra ESPERA            ; Salta al modo de bajo consumo

;=====
; INIT_CONFIG : Inicializa registro de confi-
;              guración.
; OBJETIVO   : Habilita pin Reset, LVI a 5 V
;              y Watchdog inhabilitado.
; ENTRADA   : Ninguna
; SALIDA    : Ninguna
; REGISTROS :
; AFECTADOS : CONFIG1
;=====
INIT_CONFIG
    mov #{LVI5OR3|COPD},CONFIG1 ; LVI a 5V, Watchdog inhabilitado
    rts                          ; retorna

```

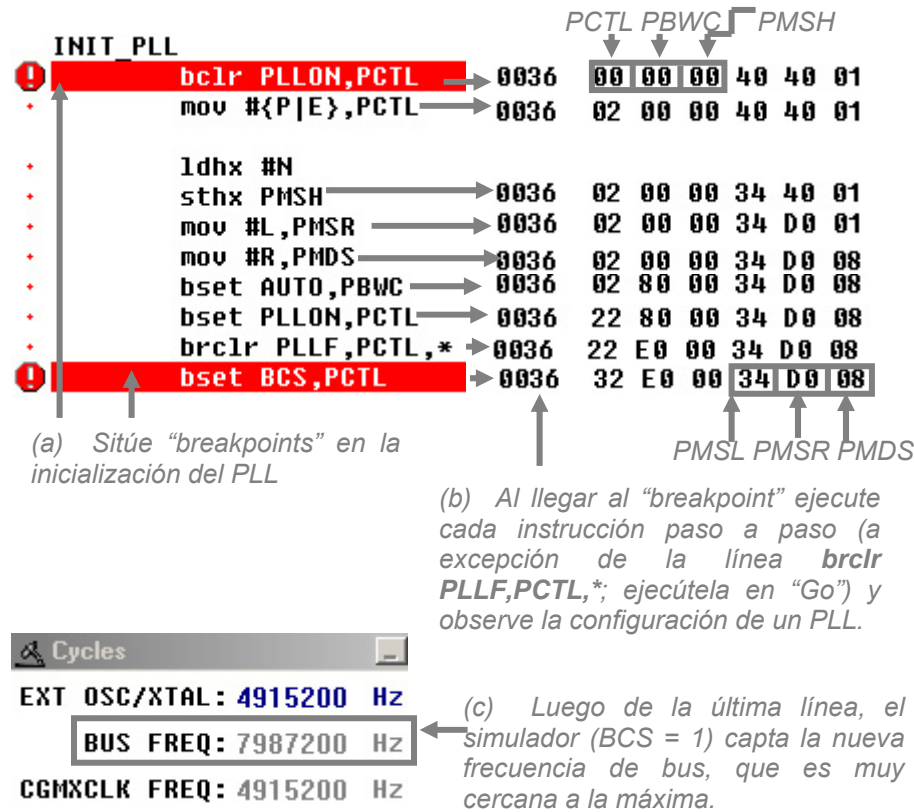
```

;=====
; INIT_PORTS : Inicializa variables y re-
;             gistros.
; OBJETIVO   : Inicializa los registros de
;             direccionamiento.
;             PTA6 = LOW
;             PTA6 = HIGH
; ENTRADA    : Ninguna
; SALIDA     : Ninguna
; REGISTROS  :
; AFECTADOS  : PORTA
;=====
INIT_PORTS
    bclr PTA6,PORTA           ; Puerto A 6 en bajo
    bset DDRA6,DDRA         ; Puerto A 6 es salida
    rts                     ; retorna
;=====
; INIT_PLL   : Inicializa el módulo de PLL.
; OBJETIVO   : Módulo PLL operando a la fre-
;             cuencia de bus MÁXIMA.
; ENTRADA    : Ninguna
; SALIDA     : Ninguna
; REGISTROS  :
; AFECTADOS  : PCTL, PMS[H:L], PMSR, PMDS,
;             PBWC
;=====
INIT_PLL
    bclr PLLON,PCTL         ; Inhabilita PLL
    mov #{P|E},PCTL        ; Selecciona reloj del sistema CGMOOUT =
                            ; CGMXCLK/2,
                            ; Programa P = 0T, E = 2T
    ldhx #N                 ; Registro de Seleccionador del Multiplicador,
                            ; N = 52T
    sthx PMSH               ; Almacena en el registro
    mov #L,PMRS             ; Selector de Rango, L = 208T
    mov #R,PMDS             ; Selector de Divisor de referencia, R = 8T
    bset AUTO,PBWC         ; PLL decide entre los modos de adquisición y
                            ; tracking
    bset PLLON,PCTL        ; Enciende el PLL
    brclr PLLF,PCTL,*      ; Espera a que se configure el PLL
    bset BCS,PCTL          ; Cambia el PLL a la referencia interna del
                            ; VCO
                            ; CGMOOUT = CGMVCLK/2
    rts                     ; Retorno

```



### 3.1.6 Simulación



- (a) Inicie WinIDE.
- (b) Cargue el archivo NT1001 – CMGC – 23 07 04.asm.
- (c) Compile.
- (d) Entre al simulador.
- (e) Añada "breakpoints" según la figura 186(a).
- (f) Corra su programa y en la sección de llamado a la rutina de inicialización del módulo reloj (INIT\_PLL) observe la ejecución de cada instrucción, ver figura 186(b).
- (g) Observe la ventana de ciclos, figura 186(c) y note que la frecuencia del bus ha cambiado a aproximadamente 8 MHz.
- (h) Ejecute el programa en modo multipaso y observe el resultado de la conmutación del LED.
- (i) Si desea "quemar" su pastilla, revisar la NT0009, Sección 1.9.5.

Nota: Recuerde mover su compilación condicional si desea que su programa corra en tiempo real:

```
; $SET ICS08
$SETNOT ICS08
```

Figura 186. Simulación del Programa Principal. (a) "Breakpoint". Sitúe "breakpoints" para visualizar la configuración del módulo de generación de reloj. (b) Registros del Módulo de Generación de Reloj. Ejecute cada instrucción y cuando haya llegado a la subrutina de inicialización, deduzca la variación que causa en el registro. (c) Frecuencia de Bus Límite. Luego de la última línea, el microcontrolador cambia al PLL como reloj de referencia, el cual actúa en el límite del bus.

### 3.1.7 Conclusión

---

Independientemente del cristal que se posea, para el microcontrolador GP32, se pueden generar frecuencias de buses mayores a la actual. Como el bus del sistema varía, afecta directamente a los módulos como el temporizador, ADC, SCI, que empiezan a calcular en base a esta referencia.

Para efecto práctico, se puede realizar un programa que calcule los parámetros de configuración del PLL; por otro lado, esta nota abarca la configuración a máxima frecuencia de bus, demostrando el poder del módulo de generación de reloj.

### 3.1.8 Referencias

---

#### 3.1.8.1 Data Técnica del MC68HC908GP32, MC68HC08GP32

(a) [http://www.freescale.com/files/microcontrollers/doc/data\\_sheet/MC68HC908GP32.pdf](http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC908GP32.pdf)

*Pág. 108 – Diagrama de Bloques del Módulo de Generación de Reloj, figura 7-1*

*Pág. 109 – Lazos Enganchados de Fase (PLL), Sección 7.4.2*

*Pág. 110 – Descripción funcional de la figura 7-1*

*Págs. 113 – 117 – Programación del PLL.*

*Págs. 119 y 135 – Circuito de Conexión Externa del PLL.*

*Pág. 122 – 130 – Información General de Registros de Control.*

*Págs. 131 y – Interrupción del PLL.*

#### 3.1.8.2 Manual de Referencia del CPU

(a) [http://www.freescale.com/files/microcontrollers/doc/ref\\_manual/CPU08RM.pdf](http://www.freescale.com/files/microcontrollers/doc/ref_manual/CPU08RM.pdf)

#### 3.1.8.3 Información General de PLL

(a) <http://users.telenet.be/educyclopedia/electronics/pll.htm>

(b) <http://www.profesores.frc.utn.edu.ar/electronica/electronicaaplicadaiii/descargas.htm>

#### 3.1.8.4 Otro Código Fuente de PLL en ensamblador

(c) <http://www.robotroom.com/PLL.html>

#### 3.1.8.5 Página “web” de Microcontroladores de esta nota

(a) <http://www.geocities.com/issaiass/index.htm>

### 3.1.9 Problemas Propuestos

---

3.1.9.1 Calcule los parámetros de configuración del PLL para GP32, para un Cristal de 9.8304 MHz y frecuencia de bus de 8 MHz.

3.1.9.2 Calcule los parámetros del PLL para un Cristal de 4.9152 MHz y frecuencia de bus de 2.5 MHz, y hacer conmutar un LED a 60 Hz utilizando el temporizador.