

## Contenido

Timer .....	2
Registros de control .....	2
Para un Timer tipo A (T1CON).....	2
Para un timer Tipo B (TxCON) .....	3
Para un timer tipo C (TyCON) .....	4
Ejercicios de Parcial.....	4
Ejercicio 1.....	4
Ejercicio 2.....	5
Ejercicio 3.....	5
Oscilador .....	6
PHASE-LOCKED LOOP (PLL).....	6
N1 (Prescaler) .....	6
M (Divisor de realimentación) .....	6
N2 (Postcaler) .....	6
Fórmulas.....	7
Programas para calcular rápidamente .....	7
Registros de Control.....	7
DMA .....	8
Registros.....	8
DMAxCON: DMA Channel x Control Register .....	8
DMAxREQ: DMA Channel x IRQ Select Register .....	9
DMAxSTA: DMA Channel x DPSRAM Start Address Offset Register A .....	10
DMAxSTB: DMA Channel x DPSRAM Start Address Offset Register B .....	10
DMAxPAD: DMA Channel x Peripheral Address Register .....	10
DMAxCNT: DMA Channel x Transfer Count Register .....	10
DSADR: Most Recent DMA DPSRAM Address Register .....	11
DMACS0: DMA Controller Status Register 0.....	11
DMACS1: DMA Controller Status Register 1.....	11
Ejercicios de parcial.....	11
<b>Ejercicio 1</b> .....	11
<b>Ejercicio 2</b> .....	12
<b>Ejercicio 3</b> .....	12
<b>Ejercicio 4</b> .....	13
<b>Ejercicio 5</b> .....	13

## Timer

La serie de micros **dsPIC33F** dispone de 9 timers de 16 bits, de los cuales el Timer1 trabaja de manera individual, y el resto puede trabajar de forma individual o en parejas como Timer de 32 bits. Se clasifican en 3 categorías:

- Timer tipo A (Timer1)
- Timer tipo B (Timer2, Timer4, Timer6 y Timer8), nos referimos a él como “Timerx”
- Timer tipo C (Timer3, Timer5, Timer7 y Timer9), nos referimos a él como “Timery”

Los timers tipo B y tipo C pueden ser combinados para formar un timer de 32 bit.

### Registros de control

Cada módulo timer es un contador/temporizador de 16 bits que consiste en los siguientes registros de lectura/escritura.

- **TMRn: 16-bit Timer Count register**

Si el timer es de 16 bits, cuenta de 0 hasta  $2^{16}$ . Si en cambio implemento un timer de 32 bits, el contador, contará hasta  $2^{32}$ . En el caso de un timer de 32 bits, debo combinar un timer tipo B con uno tipo C, por ejemplo TMR2 y TMR3.

Table 11-2: 32-bit Timer Combinations

Type B Timer (lsb)	Type C Timer (msb)
Timer2	Timer3
Timer4	Timer5
Timer6	Timer7
Timer8	Timer9

TMR2= 0 limpia el Timer lsb (less significant word).

TMR3= 0 limpia el Timer msb (most significant word).

- **PRn: 16-bit Timer Period register associated with the timer**

Cada vez que se incrementa el registro **TMRn**, se compara con el registro **PRn** y en caso de igualdad se pone a 0 el **TMRn** y se señala el bit de interrupción **TnIF**. El registro **PR1** por defecto vale 0xFFFF con lo que el periodo del Timer1 será ese, pero podemos ajustarlo al valor que queramos, lo que nos permite seleccionar una frecuencia de interrupción programable muy útil.

Cada Timer tiene asociado bits para el control de interrupciones:

- Interrupt Enable Control bit (TnIE)
- Interrupt Flag Status bit (TnIF)
- Interrupt Priority Control bits (TnIP)

### Para un Timer tipo A (T1CON)

El registro se llama T1CON y se deben configurar los siguientes bits

T1CON. [Bit]=[configuracion]			
Bit		Configuracion	Observaciones
15	TON	1-Habilito el timer 0-Desabilito el timer	Normalmente deshabilito el timer al inicio y al final de la función lo habilito.
14	Unimplemented	Leído como “0”	No se toca
13	TSIDL	1-Discontinua la operación del timer. 0-Continúa la operación del timer	Solo lo configuro cuando voy a usar el timer en modo IDLE. El modo IDLE es el modo de bajo consumo de energía.
12-7	Unimplemented	Leído como “0”	No se toca
6	TGATE	Cuando CTS=1 este bit se ignora Cuando CTS=0: Si TGATE=1, enable gated timer operation. Si TGATE=0, disable gated timer operation.	Se configura cuando se va a utilizar el timer en modo GATE. En el modo GATE, el timer opera con el clock interno (TCS=0). Se usa para medir la duración de una señal externa.
5-4	TCKPS<1:0>	11=1:256 10=1:64 01=1:8 00=1:1	Preescalers. Sirven para <b>ralentizar las señales de reloj</b> . Por la entrada entra una señal de reloj de frecuencia f y por la salida se obtiene una de frecuencia menor.

			<p>Para un prescaler de N bits, las <b>fórmulas</b> que relacionan las frecuencias y periodos de entrada con los de salida son:</p> <table><tr><td></td><td>Reloj entrada</td><td>Reloj salida</td></tr><tr><td>Frecuencia</td><td><math>f_{in}</math></td><td><math>f_{out} = \frac{f_{in}}{2^N}</math></td></tr><tr><td>Periodo</td><td><math>T_{in}</math></td><td><math>T_{out} = 2^N T_{in}</math></td></tr></table>		Reloj entrada	Reloj salida	Frecuencia	$f_{in}$	$f_{out} = \frac{f_{in}}{2^N}$	Periodo	$T_{in}$	$T_{out} = 2^N T_{in}$
	Reloj entrada	Reloj salida										
Frecuencia	$f_{in}$	$f_{out} = \frac{f_{in}}{2^N}$										
Periodo	$T_{in}$	$T_{out} = 2^N T_{in}$										
3	Unimplemented	Leído como “0”	No se toca									
2	TSYNC	Cuando CTS=1 Si TSYNC=1 (SINCRONO). Si TSYNC=0 (ASINCRONO). Cuando CTS=0 se ignora	Bit de configuración de sincronización. Se configura cuando se va a utilizar el timer en modo contador Sincrono o Asincrono (Solo en el Timer 1 o tipo A). SINCRONO: el clock externo de entrada se sincroniza con el clock interno. ASINCRONO: el clock externo de entrada NO se sincroniza con el clock interno.									
1	TCS	1= clock externo desde el pin TxCK 0=clock interno (Fosc/2)	Permite seleccionar una fuente de clock externa o la frecuencia interna de instrucciones.									
0	Unimplemented	Leído como “0”	No se toca									

### Para un timer Tipo B (TxCON)

El registro se llama TxCON, (x=2,4,6,8) y se deben configurar los siguientes bits:

T1CON. [Bit]=[configuracion]												
Bit		Configuracion	Observaciones									
15	TON	Cuando T32=1 1=enciendo el timer de 32 bits 0=apago el timer de 32 bits Cuando T32=0 1=enciendo el timer de 16 bits 0=apago el timer de 16 bits	Normalmente deshabilito el timer al inicio y al final de la funcion lo habilito.									
14	Unimplemented	Leido como "0"	No se toca									
13	TSIDL	1-Discontinua la operación del timer. 0-Continua la operación del timer	Solo lo configuro cuando voy a usar el timer en modo IDLE. El modo IDLE es el modo de bajo consumo de energía.									
12-7	Unimplemented	Leido como "0"	No se toca									
6	TGATE	Cuando CTS=1 este bit se ignora Cuando CTS=0: Si TGATE=1, enable gated timer operation. Si TGATE=0, disable gated timer operation.	Se configura cuando se va a utilizar el timer en modo GATE. En el modo GATE, el timer opera con el clock interno (TCS=0). Se usa para medir la duración de una señal externa.									
5-4	TCKPS<1:0>	11= 1:256 10=1:64 01=1:8 00=1:1	Preescalers. Sirven para <b>ralentizar las señales de reloj</b> . Por la entrada entra una señal de reloj de frecuencia f y por la salida se obtiene una de frecuencia menor.  Para un prescaler de N bits, las <b>fórmulas</b> que relacionan las frecuencias y periodos de entrada con los de salida son: <div><table><tr><td></td><td>Reloj entrada</td><td>Reloj salida</td></tr><tr><td>Frecuencia</td><td><math>f_{in}</math></td><td><math>f_{out} = \frac{f_{in}}{2^N}</math></td></tr><tr><td>Periodo</td><td><math>T_{in}</math></td><td><math>T_{out} = 2^N T_{in}</math></td></tr></table></div>		Reloj entrada	Reloj salida	Frecuencia	$f_{in}$	$f_{out} = \frac{f_{in}}{2^N}$	Periodo	$T_{in}$	$T_{out} = 2^N T_{in}$
	Reloj entrada	Reloj salida										
Frecuencia	$f_{in}$	$f_{out} = \frac{f_{in}}{2^N}$										
Periodo	$T_{in}$	$T_{out} = 2^N T_{in}$										
3	T32	1= timer de 32 bits. 0=2 timers de 16 bits por separado.	1= TMRx y TMRy forman un timer de 32 bits. 0=TMRx y TMRy forman 2 timers de 16 bits por separado.									
2	unimplemented	Leido como "0"	No se toca									
1	TCS	1= clock externo desde el pin TxCK 0=clock interno (Fosc/2)	Permite seleccionar una fuente de clock externa o la frecuencia interna de instrucciones.									
0	Unimplemented	Leido como "0"	No se toca									

## Para un timer tipo C (TyCON)

El registro se llama TyCON, (y=3,5,7,9) y se deben configurar los siguientes bits.

T1CON. [Bit]=[configuracion]												
Bit		Configuracion	Observaciones									
15	TON	1=enciendo el timer de 16 bits 0=apago el timer de 16 bits	Normalmente deshabilito el timer al inicio y al final de la function lo habilito.									
14	Unimplemented	Leido como "0"	No se toca									
13	TSIDL	1-Discontinua la operación del timer. 0-Continua la operación del timer	Solo lo configuro cuando voy a usar el timer en modo IDLE. El modo IDLE es el modo de bajo consumo de energía.									
12-7	Unimplemented	Leido como "0"	No se toca									
6	TGATE	Cuando CTS=1 este bit se ignora Cuando CTS=0: Si TGATE=1, enable gated timer operation. Si TGATE=0, disable gated timer operation.	Se configura cuando se va a utilizar el timer en modo GATE. En el modo GATE, el timer opera con el clock interno (TCS=0). Se usa para medir la duración de una señal externa.									
5-4	TCKPS<1:0>	11= 1:256 10=1:64 01=1:8 00=1:1	Preescalers. Sirven para <b>ralentizar las señales de reloj</b> . Por la entrada entra una señal de reloj de frecuencia f y por la salida se obtiene una de frecuencia menor.  Para un prescaler de N bits, las <b>fórmulas</b> que relacionan las frecuencias y periodos de entrada con los de salida son: <div><table><tr><td></td><td>Reloj entrada</td><td>Reloj salida</td></tr><tr><td>Frecuencia</td><td><math>f_{in}</math></td><td><math>f_{out} = \frac{f_{in}}{2^N}</math></td></tr><tr><td>Periodo</td><td><math>T_{in}</math></td><td><math>T_{out} = 2^N T_{in}</math></td></tr></table></div>		Reloj entrada	Reloj salida	Frecuencia	$f_{in}$	$f_{out} = \frac{f_{in}}{2^N}$	Periodo	$T_{in}$	$T_{out} = 2^N T_{in}$
	Reloj entrada	Reloj salida										
Frecuencia	$f_{in}$	$f_{out} = \frac{f_{in}}{2^N}$										
Periodo	$T_{in}$	$T_{out} = 2^N T_{in}$										
3-2	unimplemented	Leido como "0"	No se toca									
1	TCS	1= clock externo desde el pin TxCK 0=clock interno (Fosc/2)	Permite seleccionar una fuente de clock externa o la frecuencia interna de instrucciones.									
0	Unimplemented	Leido como "0"	No se toca									

## Resumen modos de configuracion

Table 11-1: Timer Modes Configuration

Mode	Bit Setting		
	TCS	TGATE <sup>(2)</sup>	TSYNC <sup>(1)</sup>
Timer	0	0	x
Gated Timer	0	1	x
Synchronous Counter	1	x	1
Asynchronous Counter <sup>(3)</sup>	1	x	0

**Note 1:** TSYNC bit is available for the Type A timer only and is ignored for both of the timer modes.

**2:** TGATE bit is ignored for both of the counter modes.

**3:** Asynchronous Counter mode is supported by the Type A timer only.

## Ejercicios de Parcial

**Ejercicio 1:** Se desea generar una interrupción con el timer de 32 bits cada 3 segundos, para una frec de 60MIPS. Indicar en Hexadecimal el valor de los registros PR3 y PR2.

Fosc=60MIPS

T=3 seg (es el periodo de PR2 y PR3 concatenados)

$T \times F_{osc} = 3 \text{ seg} \times 60 \text{ MIPS} = 180 \times 10^6 = 0ABA9500$  (valor en hexa de los registros PR2 y PR3 concatenados)

Donde

PR3 es la palabra más significativa (msw) y PR2 es la palabra menos significativa (lsw), por lo que la configuración será:

PR3=0ABA

PR2=9500

**Ejercicio 2 :** Se desea generar una interrupción con el timer de 32 bits cada 2.5 seg, para una frec de 60MIPS. Indicar en hexadecimal el valor de los registros .

$F_{osc} = 60 \text{ MIPS}$

$T = 2.5 \text{ seg}$  (es el periodo de PR2 y PR3 concatenados)

$T \times F_{osc} = 2.5 \text{ seg} \times 60 \text{ MIPS} = 150 \times 10^6 = 08F0D180$  (valor en hexa de los registros PR2 y PR3 concatenados)

Donde

PR3 es la palabra más significativa (msw) y PR2 es la palabra menos significativa (lsw), por lo que la configuración será:

PR3=08F0

PR2=D180

**Ejercicio 3:** Se desea generar una interrupción con el timer de 32 bits cada 0.25 segundos, para una frecuencia de 20 MIPS. Indicar en hexadecimal el valor de los registros PR3 y PR2.

Codigo de Matlab

```
t=0.25;
```

```
f=20000000;
```

```
N=t*f;
```

```
dec2hex(N) % Dividir en dos numeros de 4 cifras
```

Resultado

PR3 PR2

004C 4B40

# Oscilador

## PHASE-LOCKED LOOP (PLL)

El oscilador primario (Posc) y el oscilador interno FRC (Fast RC) pueden utilizarse opcionalmente en un chip PLL para obtener más altas velocidades de operación.

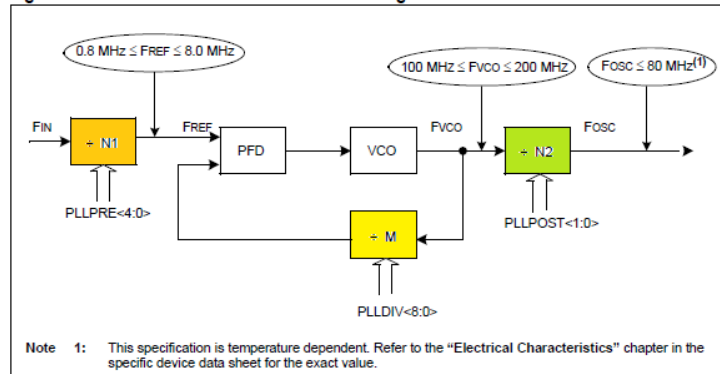
La frecuencia de entrada (Fref) del detector de frecuencia de fase (PFD) y la frecuencia de salida (Fvco) del oscilador controlado por voltaje (VCO) deben cumplir los siguientes requerimientos:

Fref de PFD debe estar entre los 0.8 y 8MHz.

Fvco de VCO debe estar en el rango de los 100 y 200MHz.

Fosc debe ser menor o igual que 80MHz.

Figure 39-8: dsPIC33F/PIC24H PLL Block Diagram



### N1 (Prescaler)

Se encarga de bajar la frecuencia de entrada para que la misma se encuentre en el rango de frec del PDF (entre 0.8 y 8MHz). Se configura mediante el registro CLKDIV, los bits implicados son PLLPRE (bits 0 a 4).

**PLLPRE<4:0>**: PLL Phase Detector Input Divider Select bits (also denoted as 'N1', PLL prescaler)

11111 = Input/33

- 
- 
- 

00001 = Input divided by 3

00000 = Input divided by 2 (default)

### M (Divisor de realimentación)

Se encarga de adecuar la frecuencia del VCO para que este entre los 100 y 200 MHz.

Se configura mediante el registro PLLFBD, los bits implicados son PLLDIV (bit 0 al 8).

**PLLDIV<8:0>**: PLL Feedback Divisor bits (also denoted as 'M', PLL multiplier)

111111111 = 513

- 
- 
- 

000110000 = 50 (default)

- 
- 
- 

000000010 = 4

000000001 = 3

000000000 = 2

### N2 (Postcaler)

Limita la frecuencia Fosc, la cual debe ser como máximo 80MHz.

Se configura mediante el registro CLKDIV, los bits implicados son PLLPOST (bits 6 a 7)

**PLLPOST<1:0>**: PLL VCO Output Divider Select bits (also denoted as 'N2', PLL postscaler)

11 = Output divided by 8

10 = Reserved

01 = Output divided by 4 (default)

00 = Output divided by 2

## Valores por defecto del PLL

Table 39-6: PLL Mode Defaults

Register	Bit Field	Value at POR	PLL Divider Ratio
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

## Fórmulas

Relación entre la frecuencia de entrada y la frecuencia del oscilador

Equation 39-3: Fosc Calculation

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

Where:

$$N1 = PLLPRE + 2$$

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

$$M = PLLDIV + 2$$

Relación entre la frecuencia de entrada y FVCO

Equation 39-4: Fvco Calculation

$$F_{VCO} = F_{IN} \times \left( \frac{M}{N1} \right) = F_{IN} \times \left( \frac{(PLLDIV + 2)}{(PLLPRE + 2)} \right)$$

Equation 39-5: FREF at a POR

$$F_{REF} = F_{IN} \left( \frac{1}{N1} \right) = 0.5(F_{IN})$$

Equation 39-6: Fvco at a POR

$$F_{VCO} = F_{IN} \left( \frac{M}{N1} \right) = F_{IN} \left( \frac{50}{2} \right) = 25(F_{IN})$$

Equation 39-7: Fosc at a POR

$$F_{OSC} = F_{IN} \left( \frac{M}{N1 \cdot N2} \right) = 6.25(F_{IN})$$

## Programas para calcular rápidamente

[https://play.google.com/store/apps/details?id=com.karlyamashita.microchippllcalculator&hl=es\\_US](https://play.google.com/store/apps/details?id=com.karlyamashita.microchippllcalculator&hl=es_US)

<http://www.chaputronia.com/calculadora-de-valores-pll-para-dspic33-y-pic24/>

## Registros de Control

No entraremos en detalle en el resto de los registros, debido a que lo importante acá (al menos para el parcial) es saber configurar el Preescaler (N1), el postcaler (N2) y el divisor de realimentación (M).

El resto de los registros están detallados a partir de la pagina 8 de la **Section 39. Oscillator (Part III)**.

### 39.4 SPECIAL FUNCTION REGISTERS

The following Special Function Registers (SFRs) provide run-time control and status of the oscillator system:

- **OSCCON: Oscillator Control Register**

OSCCON controls clock switching and provides status information that allows current clock source, PLL lock, and clock fail conditions to be monitored.

- **CLKDIV: Clock Divisor Register**

CLKDIV controls Doze mode and selects a PLL prescaler, a PLL postscaler, and an FRC postscaler.

- **PLLFBD: PLL Feedback Divisor Register**

PLLFBD selects the PLL feedback divisor.

- **OSCTUN: FRC Oscillator Tuning Register**

OSCTUN is used to tune the Internal FRC Oscillator frequency in software. It allows the Internal FRC Oscillator frequency to be adjusted over a range of ±12%.

- **ACLKCON: Auxiliary Clock Control Register**

ACLKCON controls the Auxiliary Oscillator mode and the auxiliary output clock divider.

## DMA

DMA significa direct access memory, osea acceso directo a memoria, es un controlador el cual actúa entre "memorias" haciendo transferencias o recepciones independientemente de la CPU.

¿Y para que sirve?

-Sirve para hacer transferencias/recepciones entre distintos perifericos que trae el pic (SPI, USART, CAN, ADC...) y la memoria de sistema (ya sea memoria DMA o la RAM normal del pic).

¿Que diferencia hay entre usar el metodo tradicional y usar el DMA para usar los perifericos?

-En principio si vamos a usar ese periferico para transmitir pocos bytes no merece la pena usar el DMA, pero cuando queremos enviar/recibir gran cantidad de datos es muy muy útil. El número de bytes o words (2 bytes) está limitado a 16KBytes o bien 16KWords (lo que serian 32Kbytes) por buffer, normalmente se pueden usar 2 buffers por lo cual seria 32kbytes y 32kwords respectivamente, todo esto depende de que tipo de transferencia queramos hacer (8bits o 16bits).

-¿Alguna ventaja mas?

-Si, son varias mas, entre ellas esta en que hace la transferencia automaticamente sin que la CPU actue, por lo cual podemos estar transfiriendo datos por usart mientras nuestro programa está haciendo otra cosa, incluso podemos dejar que lo haga de forma continua (una vez acaba empieza de nuevo). Asi nos evitamos tener hacer un for(), o esperando interrupciones para estar enviando/recibiendo datos, por lo cual poderemos hacer un código más complejo sin tener que recurrir a mucha velocidad ya que nos evitamos el estar ejecutando funciones para las transmisiones/recepciones.

-¿memoria DMA y memoria RAM normal ?

La diferencia de la memoria DMA y la RAM es que la DMA es de 2 puertos, esto quiere decir que el controlador DMA puede estar escribiendo/leyendo la memoria DMA mientras que la CPU tambien lo hace, sin que ninguno tenga que esperar a que el otro acabe.

-¿Que mas deberia saber?

-El DMA es un controlador, que cuenta con determinados canales, normalmente se usan 2 canales por periferico (uno para enviar y otro para recibir).

## Registros

A continuación se explican los registros para usarla, y los bits de configuración.

### DMAxCON: DMA Channel x Control Register

Este registro configura el canal DMA correspondiente habilitando / deshabilitando el canal, especificando el tamaño de la transferencia de datos, la dirección y el método de interrupción de bloque, y seleccionando el modo de direccionamiento de canal DMA, el modo operativo y el modo de escritura de datos nulos.

DMAxCONbits.(bit)=(configuracion)		(x= 0, 1, ....., 15)	
Bit		Configuracion	Observaciones
15	CHEN	1= canal activado 0= canal desactivado	Bit de activación de canal
14	SIZE	1= transfiero Byte (8 bits) 0= transfiero Word (16 bits)	Es el tamaño de lo que se va a transferir.
13	DIR	1= transferencia 0=recepcion	1=leo la RAM y escribo el periférico (transferencia) 0=leo el periférico y escribo la RAM (Recepción)
12	HALF	1=interrupción a la mitad 0=interrupción al final	Se genera una interrupcion al final de la transferencia/recepcion, o bien cuando lleve la mitad de los datos (0=al final), (1=a la mitad).
11	NULLW	1= escribo dato nulo 0=Opero de forma normal	Se utiliza para enviar datos nulos
10-6	Unimplemented	Leido como "0"	No se toca
5-4	AMODE	11= Reservado 10= Modo de redireccionamiento indirecto. 01=Registro indirecto sin post-incremento 00= Registro indirecto con post-incremento	Tipo de direccionamiento 01= con este tipo de modo haríamos que el DMA envíe los datos que apunten a determinada direccion de memoria durante x veces sin que el registro de memoria se incremente, por lo cual siempre apuntaria a la misma direccion y siempre enviaria el mismo dato.  00= Es lo contrario del anterior.
3-2	Unimplemented	Leido como "0"	No se toca
1-0	MODE	11=one shot (Ping- Pong activado)	Modo de operación del canal



		10=Continuo (Ping- Pong activado) 01= one shot (Ping- Pong desactivado) 00= Continuo (Ping- Pong desactivado)	11=envía el número de datos que queramos, y una vez los ha enviado se desactiva automáticamente, tiene el modo ping-pong activado. Se pueden enviar 2 buffers, el modo ping-pong quiere decir esto, una vez envía el buffer1 pasaria al buffer2 y una vez acabe el buffer2 se desactivaría. 10= Envía continuamente, tiene el modo ping-pong activado, lo que haria seria, transferir buffer1, luego el 2, luego el 1 de nuevo, luego 2.... y así hasta que lo desactivemos. 01= Funciona como el "11" pero no tiene el modo ping-pong activado, por lo cual envía el buffer1, y una vez se acabe se desactiva. 00= Funciona como el "10" pero sin el modo ping-pong, envía continuamente el buffer1 hasta que lo desactivemos.
--	--	---	--

## DMAxREQ: DMA Channel x IRQ Select Register

Este registro asocia el canal DMA con un periférico específico compatible con DMA.

DMAxREQ (x=0,1,.....,15)			
Bit		Configuracion	Observaciones
15	FORCE	1=Forzar transferencia 0= Automatic DMA transfer initiation request	Este bit se usa para FORZAR la transferencia sin que el periférico lo haga, muchos periféricos (SPI por ejemplo) no empiezan la transmisión hasta que reciben la orden de hacerlo, poniendo el bit FORCE a 1 haces eso.
14-7	Unimplemented	Leido como "0"	No se toca
6-0	IRQSEL	1001111 = DAC1 Left Data Output (DAC1) 1001110 = DAC1 Right Data Output (DAC1) 1000110 = TX Data Request (ECAN1) 0111100 = CODEC Transfer Done (DCI) 0101101 = PMP Master Data Transfer (PMP) 0100010 = RX Data Ready (ECAN1) 0100001 = SPI2 Transfer Done (SPI2) 0011111 = UART2 Transmitter (UART2TX) 0011110 = UART2 Receiver (UART2RX) 0001101 = ADC1 Convert Done (ADC1) 0001100 = UART1 Transmitter (UART1TX) 0001011 = UART1 Receiver (UART1RX) 0001010 = Transfer Done (SPI1) 0001000 = Timer3 (TMR3) 0000111 = Timer2 (TMR2) 0000110 = Output Compare 2 (OC2) 0000101 = Input Capture 2 (IC2) 0000010 = Output Compare 1 (OC1) 0000001 = Input Capture 1 (IC1) 0000000 = External Interrupt 0 (INT0)	Sirve para indicar la dirección del periférico con el cual va a interactuar la DMA.  Por ejemplo, en la función de configuración del DMA, para poner la dirección del DAC1 canal derecho escribo: DMA1REQ=1001110 O bien DMA1REQ=78 Que es lo mismo

## DMAxSTA: DMA Channel x DPSRAM Start Address Offset Register A

Este registro especifica el desplazamiento de la dirección de inicio principal desde la dirección base DMA DPSRAM del bloque de datos que se transferirá por el canal x DMA hacia o desde la DPSRAM. Las lecturas de este registro devuelven el valor del último desplazamiento de la dirección de transferencia DPSRAM. Las escrituras en este registro mientras el canal x está habilitado (es decir, activo) puede provocar un comportamiento impredecible y debe evitarse.

DMA1STA = __builtin_dmaoffset(LeftBufferA);		
Bits	Configuracion	Observacion
15-0		Bits de desplazamiento de la dirección de inicio DPSRAM primaria (origen o destino)

## DMAxSTB: DMA Channel x DPSRAM Start Address Offset Register B

Este registro especifica el desplazamiento de la dirección de inicio secundaria desde la dirección base DMA DPSRAM del bloque de datos que se transferirá por el canal DMA x hacia o desde la DPSRAM. Las lecturas de este registro devuelven el valor del último desplazamiento de la dirección de transferencia DPSRAM. Las escrituras en este registro mientras el canal x está habilitado (es decir, activo) puede provocar un comportamiento impredecible y debe evitarse.

DMA1STB = __builtin_dmaoffset(LeftBufferB);		
Bits	Configuracion	Observacion
15-0		Bits de desplazamiento de dirección de inicio DPSRAM secundario (origen o destino)

## DMAxPAD: DMA Channel x Peripheral Address Register

**Este registro de lectura / escritura contiene la dirección estática del registro de datos periféricos.** Las escrituras en este registro mientras el canal DMA correspondiente está habilitado (es decir, activo) puede provocar un comportamiento impredecible y debe evitarse.

DMA1PAD = (volatile unsigned int)&DAC1LDAT;		
Bits	Configuracion	Observacion
15-0		Bits de registro de dirección de periférico

## DMAxCNT: DMA Channel x Transfer Count Register

**Este registro contiene el recuento de transferencias.** DMAxCNT + 1 representa el número de solicitudes de DMA que el canal debe atender antes de que la transferencia del bloque de datos se considere completa. Es decir, un valor DMAxCNT de "0" transferirá un elemento. El valor del registro DMAxCNT es independiente del tamaño de los datos de transferencia (bit SIZE en el registro DMAxCON). Las escrituras en este registro mientras el canal DMA correspondiente está habilitado (es decir, activo) puede provocar un comportamiento impredecible y debe evitarse.

DMA1CNT = N    (N+1 es la cantidad de datos que quiero transmitir)		
Bits	Configuracion	Observacion
15-10	-	Reservados
0-9	-	Bits de registro de conteo de transferencia DMA

Además de los registros individuales del canal DMA, el controlador DMA tiene estos tres registros de estado DMA.

## DSADR: Most Recent DMA DPSRAM Address Register

Este registro de estado de solo lectura de 16 bits es común a todos los canales DMA. Captura la dirección del acceso DPSRAM más reciente (lectura o escritura). Se borra en Restablecer y, por lo tanto, contiene el valor "0x0000" si se lee antes de cualquier actividad de DMA. Se puede acceder a este registro en cualquier momento, pero está pensado principalmente como una ayuda de depuración.

## DMACS0: DMA Controller Status Register 0

Este registro de estado de solo lectura de 16 bits contiene las banderas DPSRAM y Peripheral Write Collision, XWCOLx y PWCOLx, respectivamente. Para obtener más información, consulte 38.10 "Colisiones de escritura de datos".

## DMACS1: DMA Controller Status Register 1

Este registro de estado de solo lectura de 16 bits indica qué canal de DMA estuvo activo más recientemente y proporciona el estado del modo Ping-Pong de cada canal de DMA al indicar qué registro de desplazamiento de dirección de inicio de canal DMA x DPSRAM, DMAxSTA o DMAxSTB está seleccionado.

## Ejercicios de parcial

**Ejercicio 1:** Cual es el nombre del registro (y campo de ser necesario) a modificar si se desea cambiar la dirección del periférico apuntando en el canal DMA2?

**RTA:** Para cambiar la dirección del periférico apuntado en el canal DMA2 se debe configurar el registro DMA2PAD. En él se guarda la dirección estática del periférico.

Recordar que:

**El canal DMA necesita saber en qué dirección de destino periférica leer o escribir, y cuándo realizar la misma.** Esta información se configura en el registro DMA Channel x Peripheral Address (DMAxPAD) y el registro DMA Channel x IRQ Select (DMAxREQ), respectivamente.

La Tabla 38-1 enumera los valores que se escribirán en estos registros para asociar un periférico particular con un canal DMA dado.

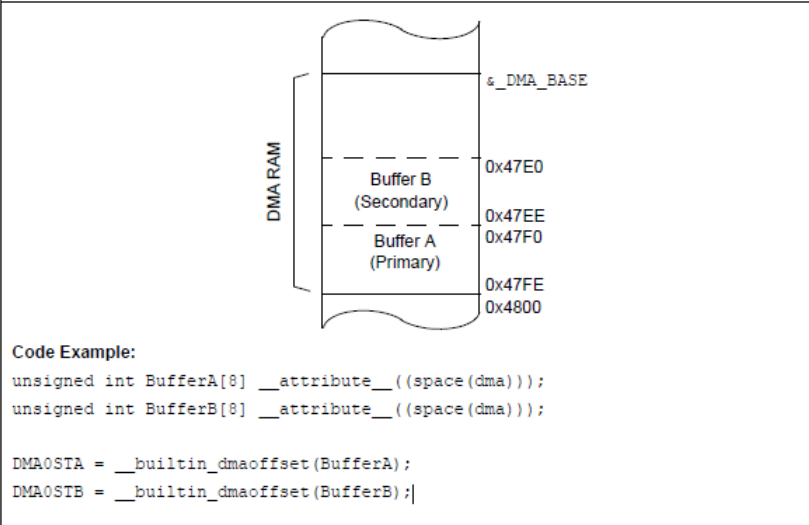
Table 38-1: DMA Channel to Peripheral Associations

Peripheral to DMA Association	DMAxREQ Register IRQSEL<6:0> Bits	DMAxPAD Register Values to Read from Peripheral	DMAxPAD Register Values to Write to Peripheral
External Interrupt 0 (INT0)	0000000	—	—
Input Capture 1 (IC1)	0000001	0x0140 (IC1BUF)	—
Input Capture 2 (IC2)	0000101	0x0144 (IC2BUF)	—
Output Compare 1 Data – OC1	0000010	—	0x0182 (OC1R)
Output Compare 1 Secondary Data (OC1)	0000010	—	0x0180 (OC1RS)
Output Compare 2 Data (OC2)	0000110	—	0x0188 (OC2R)
Output Compare 2 Secondary Data (OC2)	0000110	—	0x0186 (OC2RS)
Timer2 (TMR2)	0000111	—	—
Timer3 (TMR3)	0001000	—	—
Transfer Done (SPI1)	0001010	0x0248 (SPI1BUF)	0x0248 (SPI1BUF)
Transfer Done (SPI2)	0100001	0x0268 (SPI2BUF)	0x0268 (SPI2BUF)
UART1 Receiver (UART1RX)	0001011	0x0226 (U1RXREG)	—
UART1 Transmitter (UART1TX)	0001100	—	0x0224 (U1TXREG)
UART2 Receiver (UART2RX)	0011110	0x0236 (U2RXREG)	—
UART2 Transmitter (UART2TX)	0011111	—	0x0234 (U2TXREG)
RX Data Ready (ECAN1)	0100010	0x0440 (C1RXD)	—
TX Data Request (ECAN1)	1000110	—	0x0442 (C1TXD)
CODEC Transfer Done (DCI)	0111100	0x0290 (RXBUF0)	0x0298 (TXBUF0)
ADC1 Convert Done (ADC1)	0001101	0x0300 (ADC1BUF0)	—
PMP Master Data Transfer (PMP)	0101101	0x0608 (PMDIN1)	0x0608 (PMDIN1)
DAC1 Right Data Transfer (DAC1)	1001110	—	0x03F6 (DAC1RDAT)
DAC1 Left Data Transfer (DAC1)	1001111	—	0x03F8 (DAC1LDAT)

**Ejercicio 2:** Cual es el nombre del registro (y campo de ser necesario) a modificar si se desea cambiar la dirección de memoria RAM del buffer A en el canal DMA2?

RTA: El registro que debemos modificar es el DMA2STA.

Figure 38-6: Primary and Secondary DMA Buffer Allocation with MPLAB® IDE



Para funcionar correctamente, el DMA necesita conocer la dirección DPSRAM para leer o escribir desde el comienzo de la memoria DMA. Esta información se configura en el registro DMAxSTA y DMAxSTB.

**Ejercicio 3:** Cual es el nombre del registro (y campo de ser necesario) a modificar si se desea cambiar la cantidad de datos a transmitir en el canal DMA2?

RTA: Debo modificar el registro DMA2CNT

En el proceso de configuración de DMA, cada canal de DMA debe programarse para atender N + 1 número de solicitudes (es decir, debo programar la cantidad de datos que quiero transmitir) antes de que la transferencia del bloque de datos se considere completa.

El valor "N" se especifica programando el registro DMA Channel x Transfer Count (DMAxCNT). Es decir, un valor DMAxCNT de "0" transferirá un elemento.

Por ejemplo:

DMA2CNT = 31; (N=31)

Como N+1=32 , en la línea de código anterior estoy configurando el canal DMA2 para transmitir la cantidad de 32 datos.

El valor del registro DMAxCNT es independiente del tamaño de los datos de transferencia (byte o palabra), que se especifica en el bit Tamaño de transferencia de datos, TAMAÑO (DMAxCON).

Si me pidieran modificar el tamaño de los datos a transmitir :

DMA2CONbits.SIZE=1 (Transfiero un dato de tamaño 8 bits)

SIZE	1= transferencia Byte (8 bits) 0= transferencia Word (16 bits)	Es el tamaño de lo que se va a transferir.
------	---	--

**Ejercicio 4:** Cual es el nombre del registro (y campo de ser necesario) a modificar si se desea cambiar la dirección del periférico apuntado en el canal DMA0?

**RTA:** se debe cambiar el registro DMA0PAD.

Ver ejercicio 1 para más detalle.

**Ejercicio 5:** Cual es el registro a modificar si se desea cambiar el origen de las solicitudes (señal que dispara al comienzo de la transacción) en el canal DMA2?

**RTA:** Se debe modificar el registro DMA2REQ

**El canal DMA necesita saber en qué dirección de destino periférica leer o escribir, y cuándo realizar la misma.** Esta información se configura en el registro DMA Channel x Peripheral Address (DMAxPAD) y el registro DMA Channel x IRQ Select (DMAxREQ), respectivamente.

Por ejemplo, Supongamos que quiero leer datos del ADC, pero quiero que comience a leer cuando ingrese algún dato por la UART.

El origen de la solicitud para comenzar a leer el ADC me lo estaría dando la UART, por lo que debo configurar el DMAxREQ.

Ahora bien, también debo configurar en que dirección debo leer (en este caso el ADC), y eso lo hago mediante el registro DMAxPAD.