

# TEMA 3: MICROCONTROLADOR PIC18F4550

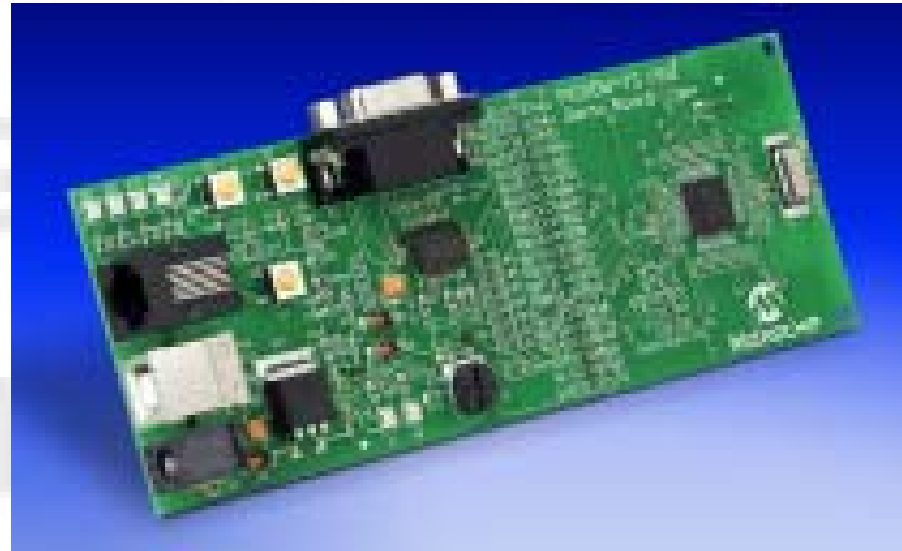
---

## INDICE:

- Descripción Funcional.
- Unidades Funcionales.
- Aplicaciones prácticas.

## BIBLIOGRAFIA:

- PIC18F2455\_2550\_4455\_4550 Data Sheet ([www.microchip.com](http://www.microchip.com))
- PIC Microcontroller: An Introduction to Software & Hardware Interfacing - Huang - Ed.: Thompson



## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / INTRODUCCION

---

#### **MICROCONTROLADORES PIC's**

- ⇒ Los PIC's de Microchip son los número 1 en ventas de microcontroladores
- ⇒ Ventajas de los microcontroladores PIC de Microchip
  - Amplia gama: gran variedad de familias que permiten adaptar el uC a las necesidades de cada aplicación
  - Herramientas de desarrollo comunes
  - Gran variedad de unidades funcionales embebidas (temporizadores, USART, I2C, SPI, unidades de comparacion/captura/PWM, Convertidores A/D, USB, receptores/transmisores de RF, Ethernet, etc...)
  - Precios competitivos
  - Buen soporte (datasheet, libros, notas de aplicación, seminarios, mucha información disponible en internet)

## **MICROCONTROLADORES PIC's**

### **⇒ Familias de microcontroladores PIC**

- **PIC10:** microcontroladores de 8 bits, de bajo coste, de 6 pines y bajas prestaciones
- **PIC12:** microcontroladores de 8 bits, de bajo coste, de 8 pines y bajas prestaciones.
- **PIC16:** microcontroladores de 8 bits, con gran variedad de número de pines y prestaciones medias.
- **PIC18:** microcontroladores de 8 bits, con gran variedad de número de pines y prestaciones medias/altas.
- **PIC24:** microcontroladores de 16 bits
- **dsPIC's**

## **FAMILIA PIC18**

### **⇒ Características fundamentales:**

- **Arquitectura RISC avanzada Harvard: 16 bit con 8 bit de datos.**
- **77 instrucciones**
- **Desde 18 a 80 pines**
- **Hasta 64K bytes de programa (hasta 2 Mbytes en ROMless)**
- **Multiplicador Hardware 8x8**
- **Hasta 3968 bytes de RAM y 1KBytes de EEPROM**
- **Frecuencia máxima de reloj 40Mhz. Hasta 10 MIPS.**
- **Pila de 32 niveles.**
- **Múltiples fuentes de interrupción**
- **Periféricos de comunicación avanzados (CAN y USB)**

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / CARACTERISTICAS FUNDAMENTALES

#### MICROCONTROLADORES PIC18F2455, PIC18F2550, PIC18F4455 y PIC18F4550

CARACTERISTICAS	PIC18F2455	PIC18F2450	PIC18F4455	PIC18F4450
Frecuencia de Operación	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz
Memoria de Programa (bytes)	24.576	32.768	24.576	32.768
Memoria RAM de Datos (bytes)	2.048	2.048	2.048	2.048
Memoria EEPROM Datos (bytes)	256	256	256	256
Interrupciones	19	19	20	20
Líneas de E/S	24	24	35	35
Temporizadores	4	4	4	4
Módulos de Comparación/Captura/PWM (CCP)	2	2	1	1
Módulos de Comparación/Captura/PWM mejorado (ECCP)	0	0	1	1
Canales de Comunicación Serie	MSSP,EUSART	MSSP,EUSART	MSSP,EUSART	MSSP,EUSART
Canal USB	1	1	1	1
Puerto Paralelo de Transmisión de Datos (SPP)	0	0	1	1
Canales de Conversión A/D de 10 bits	10 Canales	10 Canales	13 Canales	13 Canales
Comparadores analógicos	2	2	2	2
Juego de instrucciones	75 (83 ext.)	75 (83 ext.)	75 (83 ext.)	75 (83 ext.)
Encapsulados	PDIP 28 pines SOIC 28 pines	PDIP 28 pines SOIC 28 pines	PDIP 40 pines QFN 40 pines TQFP 40 pines	PDIP 40 pines QFN 40 pines TQFP 40 pines



## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / CARACTERISTICAS FUNDAMENTALES

#### PATILLAJE DEL MICROCONTROLADOR PIC18F4550

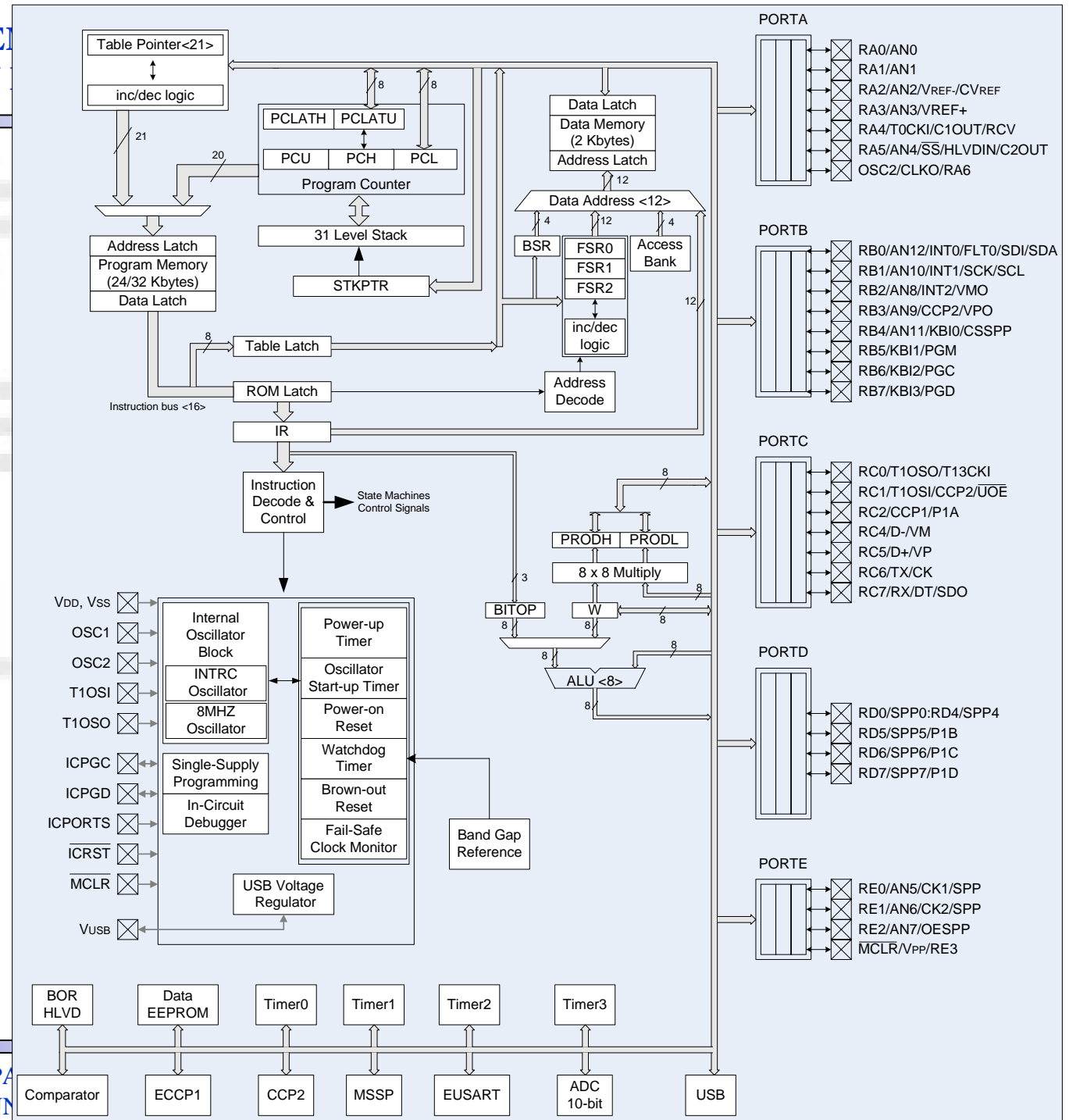
MCLR/VPP/RE3	←	□	1		40	□	→	RB7/KBI3/PGD
RA0/AN0	←	□	2		39	□	→	RB6/KBI2/PGC
RA1/AN1	←	□	3		38	□	→	RB5/KBI1/PGM
RA2/AN2/VREF-/CVREF	←	□	4		37	□	→	RB4/AN11/KBI0/CSSPP
RA3/AN3/VREF+	←	□	5		36	□	→	RB3/AN9/CCP2/VPO
RA4/T0CKI/C1OUT/RCV	←	□	6		35	□	→	RB2/AN8/INT2/VMO
RA5/AN4/SS/HLVDIN/C2OUT	←	□	7		34	□	→	RB1/AN10/INT1/SCK/SCL
RE0/AN5/CK1SPP	←	□	8		33	□	→	RB0/AN12/INT0/FLT0/SDI/SDA
RE1/AN6/CK2SPP	←	□	9		32	□	→	VDD
RE2/AN7/OESPP	←	□	10		31	□	→	VSS
VDD	→	□	11		30	□	→	RD7/SPP7/P1D
VSS	→	□	12		29	□	→	RD6/SPP6/P1C
OSC1/CLKI	→	□	13		28	□	→	RD5/SPP5/P1B
OSC2/CLKO/RA6	←	□	14		27	□	→	RD4/SPP4
RC0/T1OSO/T13CKI	←	□	15		26	□	→	RC7/RX/DT/SDO
RC1/T1OSI/CCP2 <sup>(1)</sup> /UOE	←	□	16		25	□	→	RC6/TX/CK
RC2/CCP1/P1A	←	□	17		24	□	→	RC5/D+/VP
VUSB	←	□	18		23	□	→	RC4/D-/VM
RD0/SPP0	←	□	19		22	□	→	RD3/SPP3
RD1/SPP1	←	□	20		21	□	→	RD2/SPP2

**PIC18F4455**  
**PIC18F4550**



# TE DESCRIPCION

## DIAGRAMA DE BLOQUES



## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

---

#### **ORGANIZACIÓN DE MEMORIA:**

⇒ El uC PIC18F4550 dispone de las siguientes memorias:

- **Memoria de programa:** memoria flash interna de 32.768 bytes
  - Almacena instrucciones y constantes/datos
  - Puede ser escrita/leída mediante un programador externo o durante la ejecución programa mediante unos punteros.
- **Memoria RAM de datos:** memoria SRAM interna de 2048 bytes en la que están incluidos los registros de función especial.
  - Almacena datos de forma temporal durante la ejecución del programa
  - Puede ser escrita/leída en tiempo de ejecución mediante diversas instrucciones
- **Memoria EEPROM de datos:** memoria no volátil de 256 bytes.
  - Almacena datos que se deben conservar aun en ausencia de tensión de alimentación
  - Puede ser escrita/leída en tiempo de ejecución a través de registros
- **Pila:** bloque de 31 palabras de 21 bits
  - Almacena la dirección de la instrucción que debe ser ejecutada después de una interrupción o subrutina
- **Memoria de configuración:** memoria en la que se incluyen los bits de configuración (12 bytes de memoria flash) y los registros de identificación (2 bytes de memoria de solo lectura).



## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

---

#### **MEMORIA DE CONFIGURACION:**

⇒ Se trata de un bloque de memoria situado a partir de la posición 30000H de memoria de programa (más allá de la zona de memoria de programa de usuario).

En esta memoria de configuración se incluyen:

- **Bits de configuración:** contenidos en 12 bytes de memoria flash permiten la configuración de algunas opciones del uC como:
  - Opciones del oscilador
  - Opciones de reset
  - Opciones del watchdog
  - Opciones de la circuiteria de depuración y programación
  - Opciones de protección contra escritura de memoria de programa y memoria EEPROM de datos

Estos bits se configuran generalmente durante la programación del uC, aunque también pueden ser leídos y modificados durante la ejecución del programa.

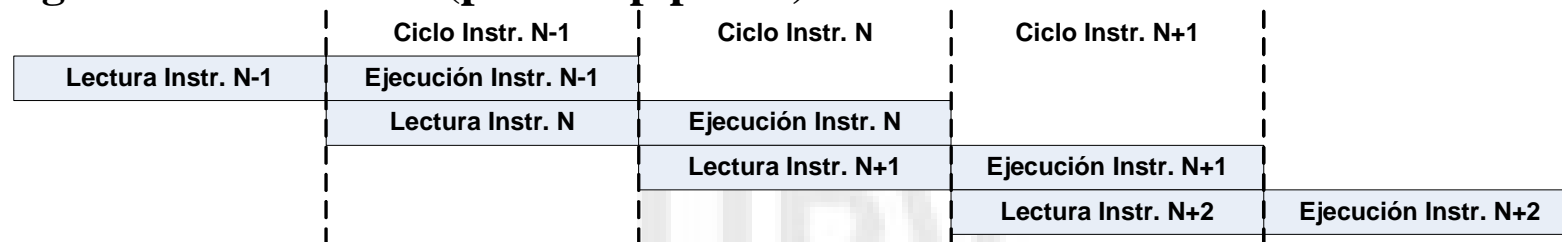
- **Registros de identificación:** se trata de dos registros situados en las direcciones 3FFFEH y 3FFFFFFH que contienen información del modelo y revisión del dispositivo. Son registros de solo lectura y no pueden ser modificados por el usuario.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

#### ARQUITECTURA HARDVARD:

- ⇒ El uC PIC18F4550 dispone buses diferentes para el acceso a memoria de programa y memoria de datos (arquitectura Harvard):
- Bus de la memoria de programa:
    - 21 líneas de dirección
    - 16/8 líneas de datos (16 líneas para instrucciones/8 líneas para datos)
  - Bus de la memoria de datos:
    - 12 líneas de dirección
    - 8 líneas de datos
- ⇒ Esto permite acceder simultáneamente a la memoria de programa y a la memoria de datos. Es decir se puede ejecutar una instrucción (lo que por lo general requiere acceso a memoria de datos) mientras se lee de la memoria de programa la siguiente instrucción (proceso pipeline).



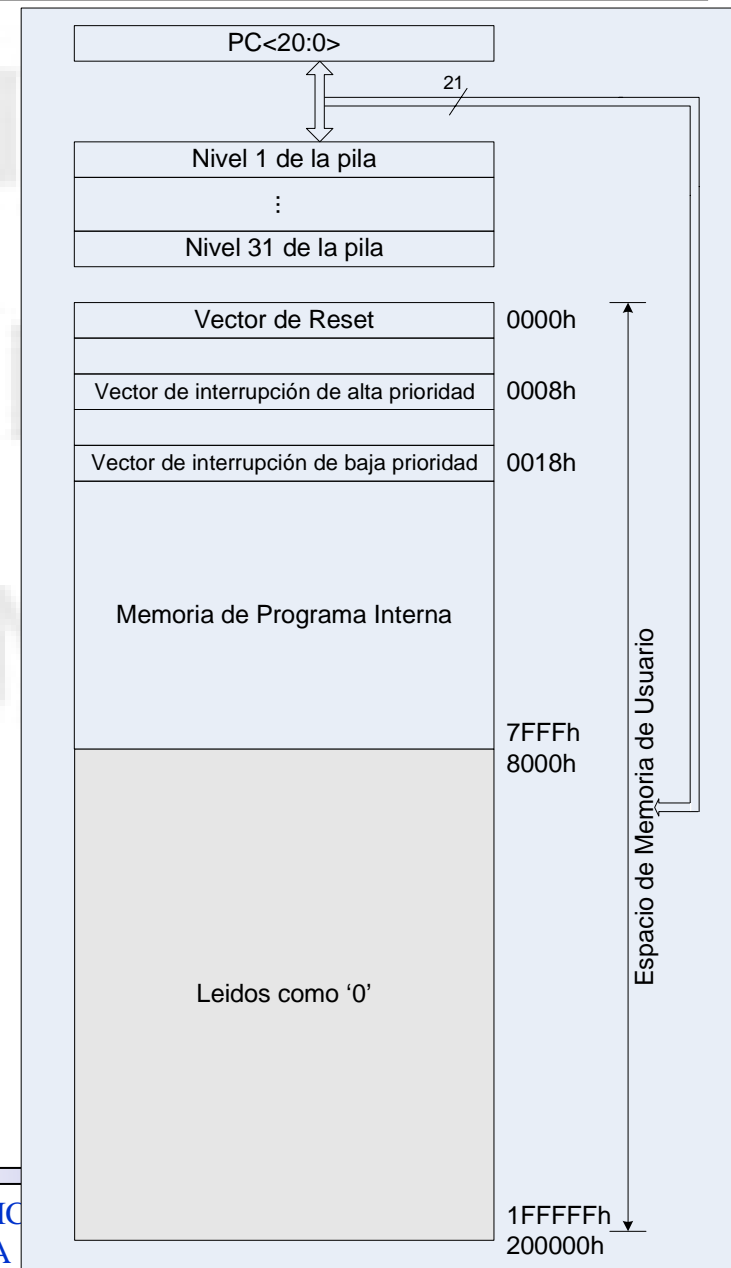
- ⇒ Por tanto la ejecución completa de **1 instrucción** (lectura instrucción + ejecución) se hace en un **1 ciclo de instrucción** ( $4 T_{OSC}$ ). EXCEPCION: las instrucciones que modifican el contenido del PC requieren 2 ciclos de instrucción.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

##### MEMORIA DE PROGRAMA:

- ⇒ El uC PIC18F4550 dispone una memoria de programa de 32.768 bytes de memoria de programa (0000H-7FFFH). Las instrucciones ocupan 2 bytes (excepto CALL, MOVFF, GOTO y LSFR que ocupan 4). Por lo tanto la memoria de programa pueden almacenar hasta 16.384 instrucciones.
- ⇒ La operación de lectura en posición de memoria por encima de 7FFFH da '0' como resultado (equivalente a la instrucción NOP).
- ⇒ Direcciones especiales de la memoria de programa
  - Vectorización del **Reset** es **0000H**
  - Vectorización de las **interrupciones de alta prioridad** es la **0008H**.
  - Vectorización de las **interrupciones de baja prioridad** es la **0018H**.



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

#### ALMACENAMIENTO DE INSTRUCCIONES EN MEMORIA DE PROGRAMA:

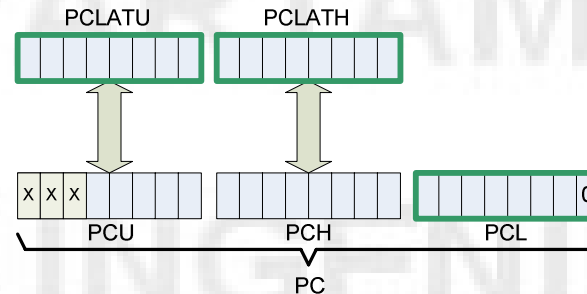
Memoria de Programa		
MOVLW 55H	55H	0020H
	0FH	0021H
CPFSEQ 20H	20H	0022H
	63H	0023H
GOTO 0110H	88H	0024H
	EFH	0025H
	00H	0026H
INCF 20H	F0H	0027H
	20H	0028H
	2BH	0029H

- ⇒ Primero se almacena la parte baja de la instrucción y luego la parte alta (para las instrucciones de 4 bytes primero los bytes menos significativos y luego los más significativos).
- ⇒ Las instrucciones siempre empiezan en direcciones pares.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

#### CONTADOR DE PROGRAMA (PC):



Puntero de 21 bits que indica la dirección en memoria de programa de la instrucción que se debe ejecutar. Está compuesto por 3 bytes:

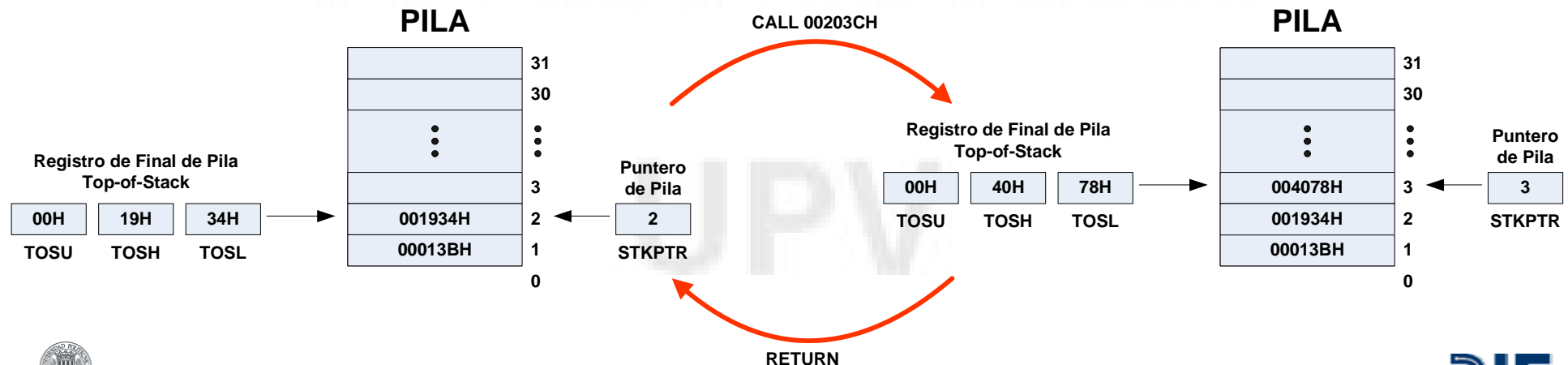
- **PCU:** parte superior del PC, registro no directamente accesible; las operaciones de lectura/escritura sobre este registro se hacen a través del registro PCLATU
- **PCH:** parte alta del PC, registro no directamente accesible; las operaciones de lectura/escritura sobre este registro se hacen a través del registro PCLATH
- **PCL:** parte baja del PC, registro directamente accesible. Una operación de lectura sobre PCL provoca que los valores de PCU y PCH pasen a PCLATU y PCLATH respectivamente. Y una operación de escritura sobre PCL provoca que los valores de PCLATU y PCLATH pasen a PCU y PCH respectivamente. El PCL siempre tiene el bit menos significativo a '0', debido a que las instrucciones siempre empiezan en direcciones pares.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

##### LA PILA DE DIRECCION:

- ⇒ La Pila es un bloque de memoria RAM independiente de 31 palabras de 21 bits que sirve para almacenar temporalmente el valor del PC cuando se produce una llamada a subrutina o una interrupción.
- ⇒ El puntero de pila (contenido en el registro STKPTR) es un contador de 5 bits que indica la posición actual del final de pila. El contenido del final de pila es accesible mediante los registros TOSU, TOSH, TOSL.
- ⇒ Cuando se procesa una interrupción o se ejecutan las instrucciones las instrucciones CALL o RCALL (el PC está apuntando a la siguiente instrucción) se incrementa el STKPTR y se almacena en el final de pila el valor del PC.
- ⇒ Cuando se ejecutan las instrucciones RETURN, RETLW o RETFIE se copia el valor almacenado en la cima de pila en el PC y se decrementa el STKPTR.



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

##### REGISTRO STKPTR

	L/R-0	L/R-0	-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
STKPTR	STKFUL	STKUNF	-	SP4	SP3	SP2	SP1	SP0

- **STKFUL**: Flag de llenado de la pila (en modo escritura únicamente puede ser puesto a '0'):
  - \* STKFUL='0': No se ha producido el llenado o desbordamiento de la pila
  - \* STKFUL='1': Se ha producido el llenado o desbordamiento de la pila
- **STKUNF**: Flag de vaciado de la pila (en modo escritura únicamente puede ser puesto a '0'):
  - \* STKUNF='0': No se ha producido el desbordamiento por vaciado de la pila
  - \* STKUNF='1': Se ha producido el desbordamiento por vaciado de la pila
- **SP4..SP0**: Puntero de pila. Estos 5 bits indican la posición del final de la pila (valor de 0 a 31)

## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

---

#### **LA PILA DE DIRECCIONES:**

⇒ **Llenado de la Pila:** si la pila llega al máximo de su capacidad (31 elementos apilados):

- Si el bit de configuración STRVEN está a ‘0’: el bit STKFUL del registro STKPTR se pone a ‘1’ y si se producen nuevos apilamientos no afectarán a la pila.
- Si el bit de configuración STRVEN está a ‘1’: el bit STKFUL del registro STKPTR se pone a ‘1’ y se producirá un reset del uC.

⇒ **Vaciado de la Pila:** si la pila está vacía y se intenta desapilar de nuevo:

- Si el bit de configuración STRVEN está a ‘0’: el bit STKUNF del registro STKPTR se pone a ‘1’, el PC se pondrá a 0000H y Puntero de pila permanecerá a 0.
- Si el bit de configuración STRVEN está a ‘1’: el bit STKUNF del registro STKPTR se pone a ‘1’ y se producirá un reset del uC.

#### **PILA RAPIDA DE REGISTRO:**

Se trata de una pila de **un solo nivel** en la que se apilan los valores del registro de estado, del W y del registro BSR cada vez que se produce una interrupción. Estos valores pueden ser recuperados si al salir de la interrupción se utiliza la instrucción “**RETFIE, FAST**”. Si están habilitadas las interrupciones de baja y alta prioridad, esta pila no debe ser utilizada en interrupciones de baja prioridad. Si no hay interrupciones habilitadas esta pila puede ser utilizada en llamadas a subrutinas (“**CALL <eti>, FAST**” y “**RETURN, FAST**”).



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

#### LECTURA DE DATOS EN MEMORIA DE PROGRAMA (FLASH):

- ⇒ La memoria de programa puede ser leída, borrada y escrita durante la ejecución del programa. La operación que se utiliza normalmente en tiempo de ejecución es la de lectura de tablas o datos almacenados en memoria de programa.
- ⇒ Existen dos formas de leer tablas de memoria de programa:

- Mediante la instrucción RETLW:

	MOVWF <DESPL.>,W	; Se almacena en valor constante de desplazamiento a W
	CALL TABLA	; Se llama a la función TABLA (después de la llamada el valor
	.	; leído de la tabla quedará almacenado en W para poder ser
	.	; utilizado)
	.	
ORG	<INICIO TABLA>	; Dirección inicial de la tabla
TABLA	ADDWF PCL	; Se suma al PCL actual el valor del desplazamiento
	RETLW <DATO0>	; Se sale de la subrutina almacenado en W DATO0
	RETLW <DATO1>	; Se sale de la subrutina almacenado en W DATO1
	.	
	.	
	.	
	RETLW <DATON>	; Se sale de la subrutina almacenado en W DATON

<DESPL.> indica la posición del elemento que se quiere leer. **Su valor debe ser el doble del valor de la posición que queremos leer.**

<INICIO TABLA> dirección de inicio de la tabla (debe ser un valor par)

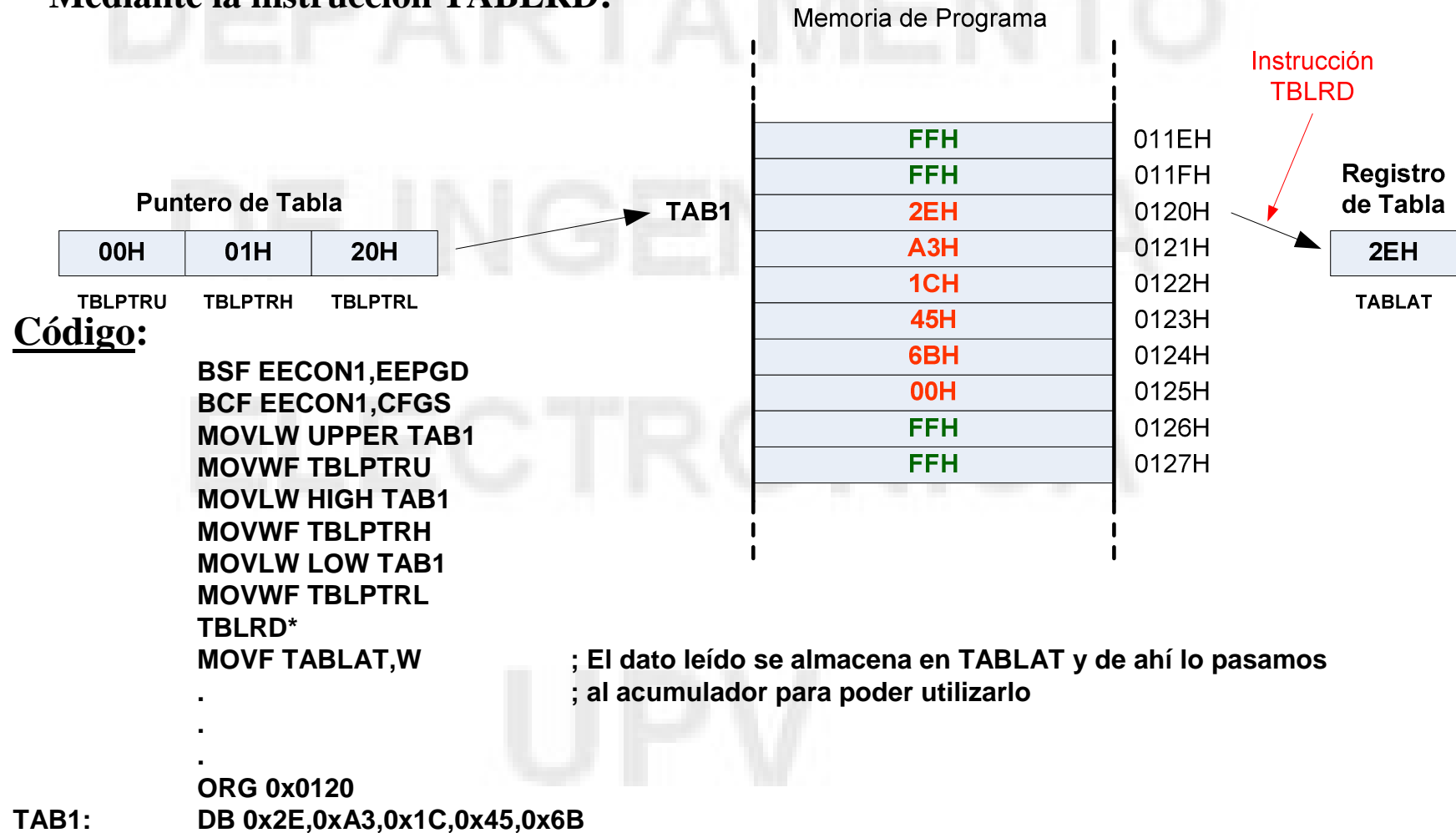
**INCONVENIENTE: Cada byte de la tabla ocupa dos bytes en memoria.**

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

## LECTURA DE DATOS EN MEMORIA DE PROGRAMA:

- Mediante la instrucción TBLRD:



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

#### **LECTURA DE DATOS EN MEMORIA DE PROGRAMA:**

- Mediante la instrucción TBLRD:

##### Proceso:

**1º** Poner a '0' el bit CFGS del registro EECON1 (CFGS='0' acceso a la memoria EEPROM o a la memoria flash de programa / CFGS='1' acceso a la memoria de configuración)

**2º** Poner a '1' el bit EEPGD del registro EECON1 (EEPGD='0' acceso a la memoria EEPROM / EEPGD='1' acceso a la memoria flash de programa).

**3º** Inicializar el puntero de tabla (registros TBLPTRU, TBLPTRH, TBLPTRL)

**4º** Leer el dato apuntado por el puntero de tabla mediante la instrucción TBLRD. El valor leído queda almacenado en el registro TABLAT. La instrucción TBLRD tiene 4 formatos:

<b>TBLRD*</b>	Lee el dato
<b>TBLRD*+</b>	Lee el dato e incrementa el puntero
<b>TBLRD*-</b>	Lee el dato y decrementa el puntero
<b>TBLRD+*</b>	Incrementa el puntero y lee el dato

Si se quiere leer una posición de memoria específica dentro de la tabla, habrá que sumarle al puntero de tabla el índice que nos lleve a dicha posición.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

##### REGISTRO EECON1

	L/R-x	L/R-x	-0	L/E-0	L/E-x	L/E-0	L/S-0	L/S-0
EECON1	EEPGD	CFGS	-	FREE	WREER	WREN	WR	RD

- **EEPGD**: Bit de selección de acceso a memoria Flash/EEPROM:
  - \* EEGD='0': Acceso a memoria de datos EEPROM
  - \* EEGD='1': Acceso a memoria Flash de programa
- **CFGS**: Bit de selección de acceso a memoria (Flash programa-EEPROM datos)/Configuración:
  - \* CFGS='0': Acceso a memoria de Flash de programa o a memoria de datos EEPROM
  - \* CFGS='1': Acceso a los registros de configuración de la memoria Flash
- **FREE**: Bit de habilitación del borrado de una fila en memoria Flash:
  - \* FREE='0': Activada únicamente la opción de lectura
  - \* FREE='1': Borrado de la fila de la memoria Flash de programa apuntada por TBLPTR en el siguiente comando de escritura (el bit se pondrá a '0' cuando la operación de borrado se haya completado)
- **WRERR**: Bit de error de escritura en memoria Flash/EEPROM:
  - \* WRERR='0': La operación de escritura en la memoria Flash/EEPROM se ha llevado a cabo correctamente
  - \* WRERR='1': Se ha producido un error en la operación de escritura en la memoria Flash/ EEPROM
- **WREN**: Bit de habilitación de la operación de escritura en memoria Flash/EEPROM:
  - \* WREN='0': Operación de escritura en la memoria Flash/EEPROM deshabilitada
  - \* WREN='1': Operación de escritura en la memoria Flash/EEPROM habilitada

## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

---

#### REGISTRO EECON1 (cont.)

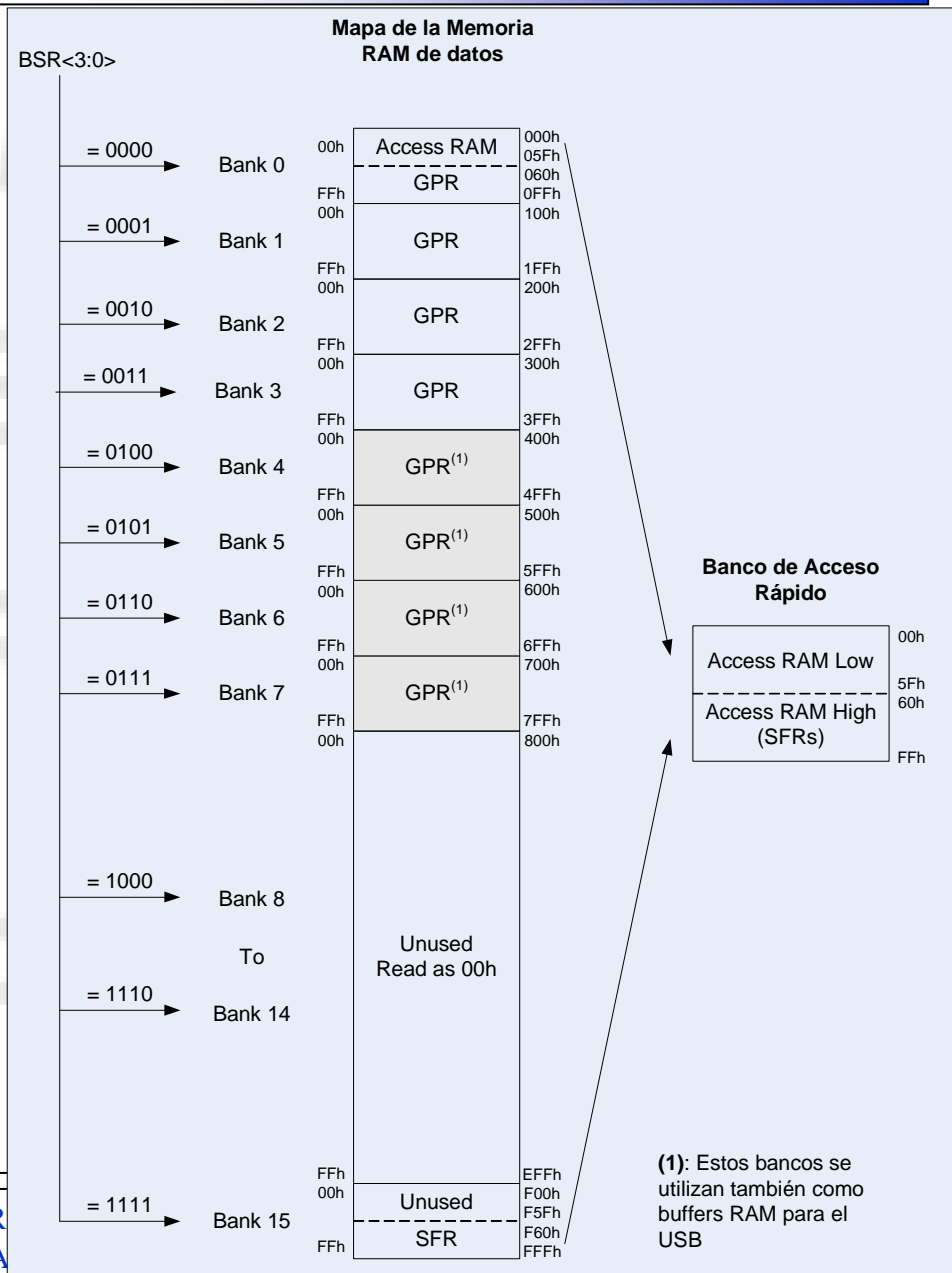
- **WR:** Bit de control de escritura en memoria Flash/EEPROM:
  - \* WR='0': La operación de escritura en la memoria Flash/EEPROM se ha completado
  - \* WR='1': Inicio de una operación de borrado/escritura en memoria EEPROM o de la operación de borrado y/o escritura en memoria Flash (cuando la operación termina el bit se pone automáticamente a '0'; por programa solo puede ponerse a '1')
- **RD:** Bit de control de lectura en memoria EEPROM:
  - \* RD='0': La operación de lectura en la memoria EEPROM se ha completado
  - \* RD='1': Inicio de una operación de lectura en memoria EEPROM (cuando la operación termina el bit se pone automáticamente a '0'; por programa solo puede ponerse a '1')

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

##### MEMORIA RAM DE DATOS:

- ⇒ El uC PIC18F4550 dispone una memoria RAM de datos 2.048 bytes (8 bancos de 256 bytes). Además dispone de 160 bytes dedicados a los registros de función especial (SFR's) situados en la parte alta del banco 15.
- ⇒ Para acceder a un byte de la memoria RAM de datos primero debe seleccionarse el banco al que pertenece el byte mediante el registro de selección de banco (BSR) y a continuación direccionar el byte dentro del banco. Además existe una modalidad de acceso rápido a las 96 posiciones de la parte baja del banco 0 y a los 160 bytes de SFR's (banco de acceso rápido).
- ⇒ Los bancos 4, 5, 6 y 7 se utilizan también para el USB.



## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

---

#### **REGISTROS DE FUNCION ESPECIAL:**

- ⇒ La memoria RAM de datos se compone de registros de propósito general (GPR's) y de registros de función especial (SFR's). Los SFR's son los registros mediante los cuales se pueden monitorizar/controlar el funcionamiento de la CPU y de las unidades funcionales del uC.
- ⇒ Se distinguen dos conjuntos de SFR's:
  - SFR's asociados con el núcleo del uC
    - CPU: WREG, STATUS, BSR, etc...
    - Interrupciones: INTCON, PIE1, PIR1, IPR1, etc...
    - Reset: RCON
  - SFR's asociados con las unidades funcionales:
    - Timers: T0CON, TMR1H, TMR1L, T1CON, etc...
    - Convertidor A/D: ADRESH, ADRESL, ADCON0, ADCON1, etc...
    - EUSART: TXREG, TXSTA, RCSTA, etc...
    - CCP: CCPR1H, CCPR1L, CCP1CON, etc...
    - MSSP: SSPSTAT, SSPDATA, SSPCFG, etc...
    - Puertos de E/S: TRISA, PORTA, TRISB, PORTB, etc...



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

##### REGISTRO STATUS

	-0	-0	-0	L/E-x	L/E-x	L/E-x	L/E-x	L/E-x
STATUS	-	-	-	N	OV	Z	DC	C

- **N**: Bit de de valor negativo. Bit utilizado para operaciones con signo (complemento a 2):
  - \* N='0': Resultado de la última operación positivo (MSB a '0')
  - \* N='1': Resultado de la última operación negativo (MSB a '1')
- **OV**: Bit de desbordamiento. Bit utilizado para operaciones con signo (complemento a 2). Indica si se ha producido desbordamiento del 7º bit (bit 6), es decir si se ha producido algún cambio en el bit 7 del resultado:
  - \* OV='0': No se ha producido desbordamiento en la operación aritmética
  - \* OV='1': Se ha producido desbordamiento en la operación aritmética
- **Z**: Bit de cero:
  - \* Z='0': El resultado de la operación aritmética o lógica ha sido diferente de 0
  - \* Z='1': El resultado de la operación aritmética o lógica ha sido 0
- **DC**: Bit de acarreo de dígito (para las instrucciones ADDWF, ADDLW, SUBLW y SUBWF):
  - \* DC='0': No se ha producido acarreo del 4º bit
  - \* DC='1': Se ha producido acarreo del 4º bit
- **C**: Bit de acarreo (para las instrucciones ADDWF, ADDLW, SUBLW y SUBWF):
  - \* DC='0': No se ha producido acarreo del 8º bit
  - \* DC='1': Se ha producido acarreo del 8º bit



## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

#### ACCESO A LA MEMORIA RAM DE DATOS:

- ⇒ La instrucción **MOVFF op1,op2** permite acceder directamente a cualquier posición de la memoria RAM de datos ya incorpora los 12 bits de la dirección de los dos operandos (es una instrucción de 2 words de 16 bits).
- ⇒ El resto de instrucciones que permiten acceder a la memoria RAM de datos incorporan un modificador “a” que establece el modo de acceso:

- Si a= 1: se accede a la totalidad de memoria mediante el BSR. Mediante los 4 bits menos significativos del BSR se selecciona el banco y mediante el operando de la instrucción se indica el byte del banco seleccionado que se quiere acceder. La instrucción **MOVLB** permite escribir directamente en el BSR el valor del banco a seleccionar (los bits BSR[7..4] no son considerados y se leen siempre como ‘0’).

**MOVLW .33** ; Se carga el valor 33 en el acumulador

**MOVLB .1** ; Se selecciona el banco de registros 1

**MOVWF VAR1,1** ; Se pasa el contenido del acumulador a VAR1 declarada en la posición 100H

- Si a = 0: se ignora el BSR y se accede al banco de acceso rápido compuesto por los primeros 96 bytes del banco 0 y los 160 bytes de los SFR's. Este método permite acceder a dichos bytes con una sola instrucción sin necesidad de seleccionar previamente el banco.

**MOVLW .33** ; Se carga el valor 33 en el acumulador

**MOVWF VAR2,0** ; Se pasa el contenido del acumulador a VAR2 declara en la posición 000H

## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

---

#### **MODOS DE DIRECCIONAMIENTO:**

⇒ El modo de direccionamiento es la forma en la que se obtienen el o los datos que van a ser utilizados en la instrucción. Existen 4 modos de direccionamiento:

**inherente, literal, directo e indirecto.**

- **Modo de direccionamiento inherente:** en este modo o bien la instrucción no tiene operando o bien el operando viene especificado en el propio código de operación de la instrucción.

RESET ; Realiza un reset por software (los operandos son todos los registros afectados por el reset)

DAW ; Ajuste decimal del acumulador (el operando es el acumulador). Formato BCD

NOP ; No realiza ninguna operación (no hay operando)

- **Modo de direccionamiento literal:** en este modo el valor del operando viene indicado de forma explícita en la instrucción.

GOTO 0x100 ; Salto incondicional a la dirección 100H (el operando es 0x100)

MOVLW .23 ; Cargar en el acumulador el valor 23 (el operando es el .23)



## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

#### MODOS DE DIRECCIONAMIENTO A LA MEMORIA RAM DE DATOS:

- Modo de direccionamiento directo: en este modo la dirección en la que se encuentra el valor del operando viene indicada de forma explícita en la instrucción.

El operando puede ser un byte o un bit:

– Operando de tipo byte:

- Mediante la instrucción **MOVFF org,dest:**

**MOVFF 0x011,0x120 ;**

10H	12H	10H	12H
11H	22H	11H	22H
12H	AFH	12H	AFH
⋮	⋮	⋮	⋮
120H	3BH	120H	22H
121H	65H	121H	65H

- Mediante la combinación del BSR y el valor de 8 bits indicado en la instrucción

**MOVLW 0xB4**  
**MOVLB .1**  
**MOVWF 0x21,1**

120H	3EH	120H	3EH
121H	15H	121H	B4H
122H	FFH	122H	FFH
⋮	⋮	⋮	⋮
FE8H	87H	FE8H	87H
WREG FE9H	B4H	FE9H	B4H

- Mediante el banco de acceso rápido

**MOVLW 0x74**  
**MOVWF 0x06,0 ; También es válido MOVWF 0x60,A**

05H	C7H	05H	C7H
06H	8DH	06H	7AH
07H	14H	07H	14H
⋮	⋮	⋮	⋮
FE8H	3BH	FE8H	22H
WREG FE9H	7AH	FE9H	65H



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

## MODOS DE DIRECCIONAMIENTO A LA MEMORIA RAM DE DATOS:

- Modo de direccionamiento directo (cont.)
  - Operando de tipo bit: en este caso en la instrucción se especifica el registro en el que se encuentra el bit y luego la posición del bit dentro del registro.

**MOVLB .2**  
**BSF 0x31,6**

	b7	b6	b5	b4	b3	b2	b1	b0
230H	0	1	1	0	1	0	1	1
231H	0	0	0	0	0	0	1	0
232H	1	1	0	1	1	1	0	1

→

	b7	b6	b5	b4	b3	b2	b1	b0
230H	0	1	1	0	1	0	1	1
231H	0	1	0	0	0	0	1	0
232H	1	1	0	1	1	1	0	1

Para facilitar las tareas de programación, en el lenguaje ensamblador los valores numéricos que corresponden a datos literales o a direcciones de memoria se representan mediante etiquetas.

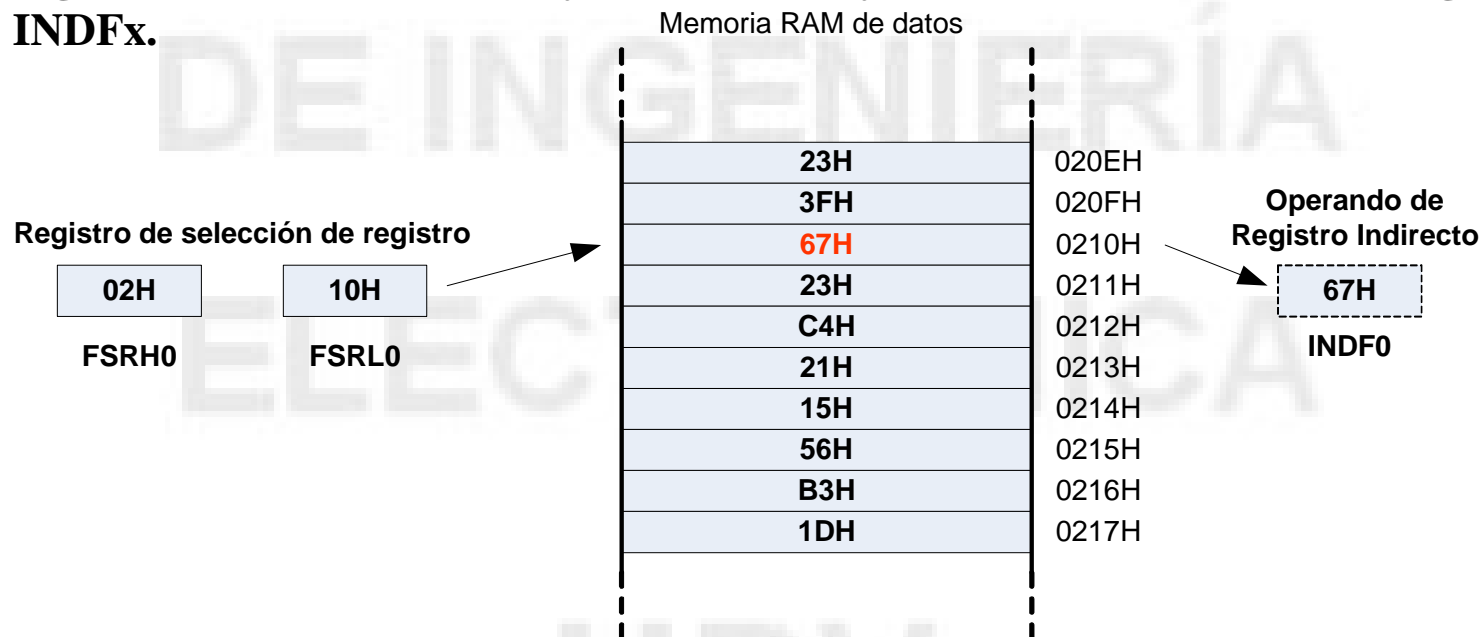
```
VAR1 EQU 0x010
CTE1 EQU .24
.
.
.
MOVLW CTE1
MOVWF VAR1,0
```

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

#### **MODOS DE DIRECCIONAMIENTO A LA MEMORIA RAM DE DATOS:**

- **Modo de direccionamiento indirecto:** en este modo la dirección de memoria en la que se encuentra el dato viene especificado en uno de los registros FSR0, FSR1 y FSR2. Para acceder al valor se debe escribir la dirección del dato (12 bits) en el registro FSRx (FSRxH[3..0] y FSRxL[3..0]) y se lee/escribe el dato en el registro INDFx.



Los registros INDFx son registros virtuales, aunque tienen una dirección asignada en la zona de SFR's, físicamente se corresponden con la dirección de memoria apuntada por el correspondiente FSRx.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

#### **MODOS DE DIRECCIONAMIENTO A LA MEMORIA RAM DE DATOS:**

- **Modo de direccionamiento indirecto (cont.):** Además de los INDFx existen otros registros virtuales que permiten acceder el dato apuntado por los FSRx, permitiendo operación adicionales:
  - **POSTDEC0, POSTDEC1, POSTDEC2:** al acceder a un registro POSTDECx se accede a la posición de memoria apuntada por el FSRx correspondiente y a continuación se decrementa el valor de dicho FSRx.
  - **POSTINC0, POSTINC1, POSTINC2:** al acceder a un registro POSTINCx se accede a la posición de memoria apuntada por el FSRx correspondiente y a continuación se incrementa el valor de dicho FSRx.
  - **PREINC0, PREINC1, PREINC2:** al acceder a un registro PREINCx se incrementa el valor del FSRx correspondiente y, a continuación, se accede a la posición de memoria apuntada por el nuevo valor del FSRx.
  - **PLUSW0, PLUSW1, PLUSW2:** al acceder a un registro PLUSWx se accede a la dirección de memoria RAM formada por la suma del valor del FSRx y del acumulador WREG (se considera en valor de WREG con signo [-127;128]). En este caso el valor del FSRx no se modifica.

La lectura de los registros virtuales (INDFx, POSTDECx, POSTINCx, PREINCx) mediante direccionamiento indirecto da por resultado 0x00. La escritura de los registros virtuales mediante direccionamiento indirecto da por resultado un NOP.

No se deben modificar los valores de los FSRx's mediante direccionamiento indirecto; se debe acceder a estos registros siempre mediante direccionamiento directo.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / INSTRUCCIONES MAS UTILIZADAS

##### Instrucciones Ensamblador más usuales:

- **MOVLW .25** ; 25→W
- **MOVWF 0x50, A** ; W → pos mem RAM [0x50]
- **MOVF 0x50,W, A** ; pos mem RAM [0x50] → W
- **MOVFF 0x70, 0x60** ; [0x70] → [0x60]
- **LFSR FSR0, 0x100** ; Cargar valor 0x100 → FSR0
- **ADDWF 0x50,W, A** ; ([0x50] + W) → W
- **ADDLW 5** ; incrementa W en 5, (W+5) → W
  
- **MOVLW .25** ; 25→W
- **ADDWFC sum, F,A** ; sumar: ( sum + W + Carry) → sum
  
- **MOVLW 0x30** ; 0x30 → W
- **SUBWF 0x50, F, A** ; Restar: ([0x50] - 0x30) → [0x50]
  
- **MOVLW n** ; n→W
- **CPFSGT cnt, A** ; compara “cnt” con “n” y salta si: cnt > n  
; Si se cumple la condición NO ejecuta sig. instrucción
- **Goto menor** ; Si cnt ≤ n salta a la etiqueta menor
- **Goto mayor** ; Si cnt > n salta a la etiqueta mayor

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

---

#### EJERCICIO:

Escribir en código de un programa que realice un retardo de 0.4 ms para el uC PIC18F4550 ( $F_{osc} = 10\text{MHz}$ ). Teniendo en cuenta que cada ciclo-instrucción tarda en ejecutarse 400 ns. El bucle tarda en ejecutarse 8us (20 ciclos de instrucción).

; Declaración de librerías

#INCLUDE <P18F4550.INC>

; Declaración de la librería de SFR

; Declaración de variables

Contador EQU 00 ; Contador del bucle

; Vectorización del Reset

ORG 0x0000

GOTO Inicio

; Salto al inicio del programa principal

; Vectorización de interrupciones

; No aplicado en este ejercicio





### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

#### EJERCICIO (cont.):

; Programa Principal

Inicio: MOVLW .50

MOVWF Contador

Buc\_r: NOP ; tiempo de ejecución un ciclo instrucción (  $t = 4 / F_{osc}$  )

NOP

NOP

; NOP numero 3

NOP

NOP

NOP

NOP

NOP

NOP

NOP

NOP

NOP

NOP

NOP

NOP

NOP

NOP

; NOP numero 17

DCFSZ Contador, A

; Decrementar contador y salta sino es cero,

; Si NO se cumple la condición → 1 ciclo instrucción, si se cumple 2 ó 3 ciclos.

BRA Buc\_r

; Salto relativo ( -1024/+1023), 1 ciclo instrucción. También se podría haber

; utilizado "GOTO Buc\_r" salto absoluto (Dir de  $\pm 20$ bits) 2 ciclo instrucción

Buc\_ppal:

GOTO Buc\_ppal

END



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / TEMPORIZADOR 0

---

#### EJERCICIO:

**Escribir en código de un programa para el uC PIC18F4550 que copie una tabla de 10 datos almacenados en memoria de programa en un bufer de memoria de datos.**

**; Declaración de librerías**

**#INCLUDE <P18F4550.INC>**

**; Declaración de la librería de SFR**

**; Declaración de variables**

**Ind\_Buf EQU 0x00 ; Contador de número de datos copiados**

**Buf\_Dat EQU 0x20 ; Bufer de recepción/transmisión**

**; Vectorización del Reset**

**ORG 0x0000**

**GOTO Main**

**; Salto al inicio del programa principal**

**; Vectorización de interrupciones**

**; No aplicado en este ejercicio**



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / TEMPORIZADOR 0

#### EJERCICIO (cont.):

; Programa Principal

Main:

```
BSF EECON1,EEPGD ; Se configura el sistema para acceso a la memoria Flash
BCF EECON1,CFGSR ; Se configura el acceso a la memoria Flash de programa
MOVLW UPPER TABLA ; Se inicializa
MOVWF TBLPTRU ; TBLPTRU al inicio de la tabla de memoria Flash
MOVLW HIGH TABLA ; Se inicializa
MOVWF TBLPTRH ; TBLPTRH al inicio de la tabla de memoria Flash
MOVLW LOW TABLA ; Se inicializa
MOVWF TBLPTRL ; TBLPTRL al inicio de la tabla de memoria Flash
CLRF Ind_Buf ; Se pone a 0 el contador de datos
LFSR FSR0,Buf_Dat ; Se Inicializa FSR0 al inicio del búfer de memoria de datos
```

Bucle\_copia:

```
TBLRD*+ ; Se lee el dato de memoria de programa y se incrementa el puntero
MOVFF TABLAT,POSTINC0 ; Se almacena el valor leído en el búfer de memoria de datos
INCF Ind_Buf ; Se incrementa el contador de datos copiados
MOVLW .10 ; Se comprueba si el contador de datos copiados
CPFSEQ Ind_Buf ; ha llegado a 10
GOTO Bucle_copia ; Si no es así repetimos la operación
```

Bucle\_principal:

```
; Bucle principal
GOTO Bucle_principal
```

; Declaración de la tabla de memoria de programa

ORG 0x0120

TABLA DB 0x2E,0xA3,0x1C,0x45,0x6B,0x44,0x67,0xF4,0x3D,0x1A

END



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

---

##### **MEMORIA EEPROM DE DATOS:**

- ⇒ El uC PIC18F4550 dispone una memoria EEPROM de datos de 256 bytes. Al ser una memoria no volátil los datos almacenados en ella se mantienen aún en ausencia de tensión de alimentación.
- ⇒ El acceso a esta memoria se realiza mediante los SFR's: EECON1, EECON2, EEDATA, EEADR.
- ⇒ Esta memoria permite hasta 1.000.000 de ciclos de borrado/escritura.
- ⇒ Se puede leer/escribir de forma individual en cada una de las 256 posiciones de memoria.
- ⇒ Cuando se realiza una operación de escritura la circuiteria interna del uC se encarga de borrar previamente la posición en la que se desea escribir. La duración de un ciclo completo de borrado/escritura de un byte en la memoria EEPROM suele ser de unos 4ms.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

##### REGISTRO EECON1

	L/R-x	L/R-x	-0	L/E-0	L/E-x	L/E-0	L/S-0	L/S-0
EECON1	EEPGD	CFGS	-	FREE	WREER	WREN	WR	RD

- **EEPGD**: Bit de selección de acceso a memoria Flash/EEPROM:
  - \* EEGD='0': Acceso a memoria de datos EEPROM
  - \* EEGD='1': Acceso a memoria Flash de programa
- **CFGS**: Bit de selección de acceso a memoria (Flash programa-EEPROM datos)/Configuración:
  - \* CFGS='0': Acceso a memoria de Flash de programa o a memoria de datos EEPROM
  - \* CFGS='1': Acceso a los registros de configuración de la memoria Flash
- **WREER**: Bit de error de escritura en memoria Flash/EEPROM:
  - \* WREER='0': La operación de escritura en la memoria Flash/EEPROM se ha llevado a cabo correctamente
  - \* WREER='1': Se ha producido un error en la operación de escritura en la memoria Flash/ EEPROM
- **WREN**: Bit de habilitación de la operación de escritura en memoria Flash/EEPROM:
  - \* WREN='0': Operación de escritura en la memoria Flash/EEPROM deshabilitada
  - \* WREN='1': Operación de escritura en la memoria Flash/EEPROM habilitada
- **WR**: Bit de control de escritura en memoria Flash/EEPROM:
  - \* WR='0': La operación de escritura en la memoria Flash/EEPROM se ha completado
  - \* WR='1': Inicio de una operación de borrado/escritura en memoria EEPROM o de las operación de borrado y/o escritura en memoria Flash (cuando la operación termina el bit se pone automáticamente a '0'; por programa solo puede ponerse a '1')



## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

---

#### REGISTRO EECON1 (cont.)

- **RD:** Bit de control de lectura en memoria EEPROM:
  - \* RD='0': La operación de lectura en la memoria EEPROM se ha completado
  - \* RD='1': Inicio de una operación de lectura en memoria EEPROM (cuando la operación termina el bit se pone automáticamente a '0'; por programa solo puede ponerse a '1')

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

## OPERACIÓN DE LECTURA EN LA MEMORIA EEPROM DE DATOS:

⇒ El proceso de lectura en la memoria EEPROM es el siguiente:

**1º** Poner a '0' el bit CFGS del registro EECON1 (CFGS='0' acceso a la memoria EEPROM o a la memoria flash de programa / CFGS='1' acceso a la memoria de configuración).

**2º** Poner a '0' el bit EEPGD del registro EECON1 (EEPGD='0' acceso a la memoria EEPROM / EEPGD='1' acceso a la memoria flash de programa).

**3º** Escribir la dirección del dato que se quiere leer en el registro EEADR.

**4º** Poner a '1' el bit RD del registro EECON1. Esto iniciará el proceso de lectura. En el siguiente ciclo de instrucción el dato estará disponible en el registro EEDATA. El bit RD se pone por hardware a '0' cuando concluye el proceso de lectura.

BCF EECON1, CFGS	; Se activa el acceso a memoria EEROM/Flash
BCF EECON1, EEPGD	; Se activa el acceso a memoria EEPROM
MOVLW .2	; Se selecciona la dirección
MOVWF EEADR	; de memoria que se quiere leer
BSF EECON1, RD	; Orden de lectura
MOVF EEDATA, W	; El dato leído se carga en el acumulador

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

#### **OPERACIÓN DE ESCRITURA EN LA MEMORIA EEPROM DE DATOS:**

⇒ El proceso de escritura en la memoria EEPROM es el siguiente:

- 1º** Poner a '0' el bit CFGS del registro EECON1 (CFGS='0' acceso a la memoria EEPROM o a la memoria flash de programa / CFGS='1' acceso a la memoria de configuración).
  - 2º** Poner a '0' el bit EEPGD del registro EECON1 (EEPGD='0' acceso a la memoria EEPROM / EEPGD='1' acceso a la memoria flash de programa).
  - 3º** Escribir la dirección en la que se desea escribir en el registro EEADR.
  - 4º** Escribir el dato que se desea grabar en memoria en el registro EEDATA.
  - 5º** Poner a '1' el bit WREN del registro EECON1 para habilitar la escritura.
  - 6º** Deshabilitar las interrupciones (bit GIE a '0'). Este paso es necesario para que la secuencia de los pasos 7º y 8º no se vea interrumpida.
  - 7º** Escribir sucesivamente los valores 55H y AAH en el registro virtual EECON2.
  - 8º** Poner a '1' el bit WR del registro EECON1. Esto iniciará el proceso de escritura que suele tardar unos 4ms.
  - 9º** Habilitar las interrupciones (bit GIE a '1').
- Cuando concluye el proceso de escritura, el bit WR se pone a '0' por hardware y el flag de interrupción EEIF se pone a '1'. El usuario debe detectar el final del proceso de escritura mediante interrupción o mediante polling y poner a '0' el bit WREN para deshabilitar la escritura en la EEPROM.



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

## OPERACIÓN DE ESCRITURA EN LA MEMORIA EEPROM DE DATOS:

⇒ El siguiente código permite la escritura de una dato en la EEPROM:

```
BCF EECON1, CFGS ; Se activa el acceso a memoria EEROM/Flash
BCF EECON1, EEPGD ; Se activa el acceso a memoria EEPROM
MOVLW .8 ; Se selecciona la dirección 8 de la memoria EEPROM
MOVWF EEADR ; de memoria en la que se quiere escribir
MOVLW 0x4F ; Se escribe en EEDATA
MOVWF EEDATA ; el valor que se desea escribir (dato:0x4F)
BSF EECON1, WREN ; Se habilita la escritura
BCF INTCON, GIE ; Se deshabilitan las interrupciones para evitar interrumpir la secuencia
MOVLW 55h ; Se escribe 55H
MOVWF EECON2 ; en el registro virtual EECON2
MOVLW 0AAh ; Se escribe AAH
MOVWF EECON2 ; en el registro virtual EECON2
BSF EECON1, WR ; Se inicia el proceso de escritura
BSF INTCON, GIE ; Una vez acabada la secuencia se vuelven a habilitar las interrupciones
.
.
.
; Una vez se detecta el final del proceso de escritura mediante el flag EEIF
BCF EECON1, WREN ; Se deshabilita la escritura en la EEPROM
```

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / ORGANIZACIÓN DE LA MEMORIA

---

#### **OPERACIÓN DE ESCRITURA EN LA MEMORIA EEPROM DE DATOS:**

- ⇒ **Verificación de los datos:** es interesante verificar que la operación de escritura se ha llevado a cabo correctamente mediante la lectura del dato escrito.
- ⇒ **Bit CPD de protección de la EEPROM:** el bit CPD del registro de configuración CONFIG5H permite la protección de la memoria EEPROM de operaciones externas de lectura/escritura. Este bit **no afecta** a las operaciones de lectura/escritura **internas**.
- ⇒ **Escritura indeseada en memoria EEPROM:** un problema bastante común en las memorias EEPROM es la corrupción de los datos almacenados debidos a operaciones de escritura indeseadas. Para evitar este tipo de situaciones durante la inicialización del uC (power-up, reset) la operación de escritura queda deshabilitada.

## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

#### **SISTEMA DE INTERRUPCIONES:**

⇒ El uC PIC18F4550 dispone de 21 fuentes de interrupciones. Se distinguen dos grupos de interrupciones:

- **Grupo general de interrupciones:**

Interrupción del Temporizador 0
Interrupción por cambio en PORTB
Interrupción externa 0
Interrupción externa 1
Interrupción externa 2

- **Grupo de interrupciones de periféricos**

Interrupción del SPP	Interrupción de fallo del oscilador
Interrupción del A/D	Interrupción del comparador
Interrupción de recepción de la EUSART	Interrupción del USB
Interrupción de transmisión de la EUSART	Interrupción de escritura en Flash/EEPROM
Interrupción del MSSP	Interrupción de colisión de bus (MSSP)
Interrupción del CCP1	Interrupción de detección de anomalías en $V_{DD}$
Interrupción del Temporizador 2	Interrupción del Temporizador 3
Interrupción del Temporizador 1	Interrupción del CCP2

## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

---

#### **SISTEMA DE INTERRUPCIONES:**

⇒ Se dispone de dos niveles de prioridad:

- Nivel alto vectorizado en la dirección 0008H
- Nivel bajo, vectorizado en la dirección 0018H

Todas las interrupciones pueden ser programadas con cualquiera de las dos prioridades, salvo la interrupción externa 0 (que siempre tiene alta prioridad).

⇒ Todas las interrupciones disponen de 3 bits de configuración (excepto la interrupción externa 0 que tiene dos):

- Bit de habilitación de interrupción: permite habilitar a nivel individual la interrupción.
- Flag de interrupción: se pone a '1' cuando se produce la condición de interrupción independientemente de si la interrupción está habilitada o no. Este flag debe ponerse '0' por software cuando se procesa la interrupción.
- Bit de prioridad de interrupción: establece si la interrupción es de alta o de baja prioridad (este bit no está disponible para la interrupción externa 0).

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

##### REGISTRO RCON

	L/E-0	L/E-1	-0	L/E-1	L-1	L-1	L/E-0	L/E-0
RCON	IPEN	SBOREN	-	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$

- **IPEN**: Bit de habilitación de prioridades en las interrupciones
  - \* IPEN='0': Sistema de prioridad en las interrupciones deshabilitado
  - \* IPEN='1': Sistema de prioridad en las interrupciones habilitado

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

##### REGISTRO INTCON

	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF

- **GIE/GIEH**: Bit de habilitación global de las interrupciones. Su función depende del valor de IPEN:

IPEN='0':

- \* GIE='0': Interrupciones deshabilitadas a nivel global
- \* GIE='1': Interrupciones habilitadas a nivel global

IPEN='1':

- \* GIEH='0': Interrupciones de alta prioridad deshabilitadas a nivel global
- \* GIEH='1': Interrupciones de alta prioridad habilitadas a nivel global

- **PEIE/GIEL**: Bit de habilitación global de las interrupciones de periféricos. Su función depende del valor de IPEN:

IPEN='0':

- \* PEIE='0': Interrupciones de periféricos deshabilitadas a nivel global
- \* PEIE='1': Interrupciones de periféricos habilitadas a nivel global

IPEN='1':

- \* GIEL='0': Interrupciones de baja prioridad deshabilitadas a nivel global
- \* GIEL='1': Interrupciones de baja prioridad habilitadas a nivel global

## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

#### REGISTRO INTCON (cont.)

- **TMR0IE**: Bit de habilitación de la interrupción de desbordamiento del Temporizador 0
  - \* TMR0IE='0': Interrupción de desbordamiento del Temporizador 0 deshabilitada
  - \* TMR0IE='1': Interrupción de desbordamiento del Temporizador 0 habilitada
- **INT0IE**: Bit de habilitación de la interrupción externa 0
  - \* INT0IE='0': Interrupción externa 0 deshabilitada
  - \* INT0IE='1': Interrupción externa 0 habilitada
- **RBIE**: Bit de habilitación de la interrupción por cambio en el Puerto B
  - \* RBIE='0': Interrupción por cambio en el Puerto B deshabilitada
  - \* RBIE='1': Interrupción por cambio en el Puerto B habilitada
- **TMR0IF**: Flag de la interrupción de desbordamiento del Temporizador 0
  - \* TMR0IF='0': No se ha producido desbordamiento del Temporizador 0
  - \* TMR0IF='1': Se ha producido desbordamiento del Temporizador 0
- **INT0IF**: Flag de la interrupción externa 0
  - \* INT0IF='0': No se ha producido un flanco en la línea RB0/INT0
  - \* INT0IF='1': Se ha producido un flanco en la línea RB0/INT0
- **RBIF**: Flag de la interrupción por cambio en el Puerto B
  - \* RBIF='0': No se ha producido ningún cambio en ninguna de las líneas RB7..RB4
  - \* RBIF='1': Se ha producido ningún cambio en ninguna de las líneas RB7..RB4

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

##### REGISTRO INTCON2

	L/E-1	L/E-1	L/E-1	L/E-1	-0	L/E-1	-0	L/E-1
INTCON2	<b>RBP</b> <b>U</b>	<b>INTEDG0</b>	<b>INTEDG1</b>	<b>INTEDG2</b>	-	<b>TMR0IP</b>	-	<b>RBIP</b>

- **RBP****U**: Bit de habilitación de las resistencias de pull-up del Puerto B:
  - \* **RBP****U**=‘0’: Las resistencias de pull-up del Puerto B están activadas
  - \* **RBP****U**=‘1’: Las resistencias de pull-up del Puerto B están desactivadas
- **INTEDG0**: Bit de selección de flanco de la interrupción externa 0:
  - \* **INTEDG0**=‘0’: La interrupción externa 0 se dispara por flanco de bajada
  - \* **INTEDG0**=‘1’: La interrupción externa 0 se dispara por flanco de subida
- **INTEDG1**: Bit de selección de flanco de la interrupción externa 1:
  - \* **INTEDG1**=‘0’: La interrupción externa 1 se dispara por flanco de bajada
  - \* **INTEDG1**=‘1’: La interrupción externa 1 se dispara por flanco de subida
- **INTEDG2**: Bit de selección de flanco de la interrupción externa 2:
  - \* **INTEDG2**=‘0’: La interrupción externa 2 se dispara por flanco de bajada
  - \* **INTEDG2**=‘1’: La interrupción externa 2 se dispara por flanco de subida
- **TMR0IP**: Bit de selección de prioridad de la interrupción del Temporizador 0
  - \* **TMR0IP**=‘0’: Prioridad baja para la interrupción del Temporizador 0
  - \* **TMR0IP**=‘1’: Prioridad alta para la interrupción del Temporizador 0
- **RBIP**: Bit de selección de prioridad de la interrupción por cambio en el Puerto B
  - \* **RBIP**=‘0’: Prioridad baja para la interrupción por cambio en el Puerto B
  - \* **RBIP**=‘1’: Prioridad alta para la interrupción por cambio en el Puerto B



## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

#### REGISTRO INTCON3

	L/E-1	L/E-1	-0	L/E-0	L/E-0	-0	L/E-0	L/E-0
INTCON3	INT2IP	INT1IP	-	INT2IE	INT1IE	-	INT2IF	INT1IF

- **INT2IP**: Bit de selección de prioridad de la interrupción externa 2
  - \* INT2IP='0': Prioridad baja para la interrupción externa 2
  - \* INT2IP='1': Prioridad alta para la interrupción externa 2
- **INT1IP**: Bit de selección de prioridad de la interrupción externa 1
  - \* INT1IP='0': Prioridad baja para la interrupción externa 1
  - \* INT1IP='1': Prioridad alta para la interrupción externa 1
- **INT2IE**: Bit de habilitación de la interrupción externa 2
  - \* INT2IE='0': Interrupción externa 2 deshabilitada
  - \* INT2IE='1': Interrupción externa 2 habilitada
- **INT1IE**: Bit de habilitación de la interrupción externa 1
  - \* INT1IE='0': Interrupción externa 1 deshabilitada
  - \* INT1IE='1': Interrupción externa 1 habilitada
- **INT2IF**: Flag de la interrupción externa 2
  - \* INT2IF='0': No se ha producido flanco en la línea RB2/INT2
  - \* INT2IF='1': Se ha producido flanco en la línea RB2/INT2
- **INT1IF**: Flag de la interrupción externa 1
  - \* INT1IF='0': No se ha producido flanco en la línea RB1/INT1
  - \* INT1IF='1': Se ha producido flanco en la línea RB1/INT1



## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPTONES

#### REGISTRO PIR1

	L/E-0	L/E-0	L-0	L-0	L/E-0	L/E-0	L/E-0	L/E-0
PIR1	SPPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF

- **SPPIF**: Flag de la interrupción de lectura/escritura en el canal de transmisión de datos en paralelo (SPP)
  - \* SPPIF='0': No se ha producido operación de lectura/escritura en el SPP
  - \* SPPIF='1': Se ha producido operación de lectura/escritura en el SPP
- **ADIF**: Flag de la interrupción del convertidor A/D
  - \* ADIF='0': No se ha completado la conversión A/D
  - \* ADIF='1': Se ha completado una conversión
- **RCIF**: Flag de la interrupción de recepción del EUSART
  - \* RCIF='0': El búfer de recepción del EUSART (RCREG) está vacío
  - \* RCIF='1': El búfer de recepción del EUSART (RCREG) está lleno
- **TXIF**: Flag de la interrupción de transmisión del EUSART
  - \* TXIF='0': El búfer de recepción del EUSART (TXREG) está lleno
  - \* TXIF='1': El búfer de recepción del EUSART (RCREG) está vacío
- **SSPIF**: Flag de la interrupción del puerto serie síncrono (SSP)
  - \* SSPIF='0': A la espera de una recepción/transmisión
  - \* SSPIF='1': Se ha producido una transmisión/recepción

## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

#### REGISTRO PIR1 (cont.)

- **CCP1IF**: Flag de la interrupción del módulo 1 de Comparación/Captura/PWM (CCP1)

##### Modo Captura:

- \* CCP1IF='0': No se ha capturado el valor del Temporizador 1
- \* CCP1IF='1': Se ha capturado el valor del Temporizador 1

##### Modo Comparación:

- \* CCP1IF='0': No se ha producido igualdad con el valor del Temporizador 1
- \* CCP1IF='1': Se ha producido igualdad con el valor del Temporizador 1

- **TMR2IF**: Flag de la interrupción de igualdad entre TMR2 y PR2

- \* TMR2IF='0': No se ha producido igualdad con el valor del Temporizador 2 y el valor del registro PR2
- \* TMR2IF='1': Se ha producido igualdad con el valor del Temporizador 2 y el valor del registro PR2

- **TMR1IF**: Flag de la interrupción de desbordamiento del Temporizador 1

- \* TMR1IF='0': No se ha producido desbordamiento del Temporizador 1
- \* TMR1IF='1': Se ha producido desbordamiento del Temporizador 1

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

##### REGISTRO PIR2

	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
PIR2	OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF

- **OSCFIF**: Flag de la interrupción de fallo en el oscilador
  - \* OSCFIF='0': El oscilador funciona correctamente
  - \* OSCFIF='1': Se ha producido un fallo en el oscilador. La señal de reloj ha sido conmutada a INTOSC
- **CMIF**: Flag de la interrupción del comparador
  - \* CMIF='0': La entrada del comparador no ha cambiado
  - \* CMIF='1': La entrada del comparador ha cambiado
- **USBIF**: Flag de la interrupción del USB
  - \* USBIF='0': El módulo USB no ha generado una interrupción
  - \* USBIF='1': El módulo USB ha generado una interrupción
- **EEIF**: Flag de la interrupción de escritura en las memorias EEPROM/ Flash
  - \* EEIF='0': La operación de escritura en memoria EEPROM/Flash no se ha iniciado o no ha concluido
  - \* EEIF='1': La operación de escritura en memoria EEPROM/Flash ha concluido
- **BCLIF**: Flag de la interrupción de colisión de bus en el SSP
  - \* BCLIF='0': No se ha producido colisión de bus en el SSP
  - \* BCLIF='1': Se ha producido colisión de bus en el SSP

## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

#### REGISTRO PIR2 (cont.)

- **HLVIF**: Flag de la interrupción de detección de tensión alta/baja
  - \* HLVIF='0': No se ha detectado ninguna condición de tensión alta o baja
  - \* HLVIF='1': Se ha detectado ninguna condición de tensión alta o baja
- **TMR3IF**: Flag de la interrupción de desbordamiento del Temporizador 3
  - \* TMR3IF='0': No se ha producido desbordamiento del Temporizador 3
  - \* TMR3IF='1': Se ha producido desbordamiento del Temporizador 3
- **CCP2IF**: Flag de la interrupción del módulo 2 de Comparación/Captura/PWM (CCP2)

#### Modo Captura:

- \* CCP2IF='0': No se ha capturado el valor del Temporizador 1 ni del Temporizador 3
- \* CCP2IF='1': Se ha capturado el valor del Temporizador 1 o del Temporizador 3

#### Modo Comparación:

- \* CCP2IF='0': No se ha producido igualdad con el valor del Temporizador 1 ni con el Temporizador 3
- \* CCP2IF='1': Se ha producido igualdad con el valor del Temporizador 1 o con el Temporizador 3

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

##### REGISTRO PIE1

	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
PIE1	SPPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE

- **SPPIE**: Bit de habilitación de la interrupción de lectura/escritura en el canal de transmisión de datos en paralelo (SPP)
  - \* SPPIE='0': Interrupción de lectura/escritura en el SPP no habilitada
  - \* SPPIE='1': Interrupción de lectura/escritura en el SPP habilitada
- **ADIE**: Bit de habilitación de la interrupción del convertidor A/D
  - \* ADIE='0': Interrupción del convertidor A/D no habilitada
  - \* ADIE='1': Interrupción del convertidor A/D habilitada
- **RCIE**: Bit de habilitación de la interrupción de recepción del EUSART
  - \* RCIE='0': Interrupción de recepción del EUSART no habilitada
  - \* RCIE='1': Interrupción de recepción del EUSART habilitada
- **TXIE**: Bit de habilitación de la interrupción de transmisión del EUSART
  - \* TXIE='0': Interrupción de transmisión del EUSART no habilitada
  - \* TXIE='1': Interrupción de transmisión del EUSART habilitada
- **SSPIE**: Bit de habilitación de la interrupción del puerto serie síncrono (SSP)
  - \* SSPIE='0': Interrupción del puerto serie síncrono (SSP) no habilitada
  - \* SSPIE='1': Interrupción del puerto serie síncrono (SSP) habilitada

## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

---

#### REGISTRO PIE1 (cont.)

- **CCP1IE**: Bit de habilitación de la interrupción del módulo 1 de Comparación/Captura/PWM (CCP1)
  - \* CCP1IE='0': Interrupción del CCP1 no habilitada
  - \* CCP1IE='1': Interrupción del CCP1 habilitada
- **TMR2IE**: Bit de habilitación de la interrupción de igualdad entre TMR2 y PR2
  - \* TMR2IE='0': Interrupción de igualdad entre TMR2 y PR2 no habilitada
  - \* TMR2IE='1': Interrupción de igualdad entre TMR2 y PR2 habilitada
- **TMR1IE**: Bit de habilitación de la interrupción de desbordamiento del Temporizador 1
  - \* TMR1IE='0': Interrupción de desbordamiento del Temporizador 1 no habilitada
  - \* TMR1IE='1': Interrupción de desbordamiento del Temporizador 1 habilitada

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

##### REGISTRO PIE2

	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
PIE2	OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE

- **OSCFIE**: Bit de habilitación de la interrupción de fallo en el oscilador
  - \* OSCFIE='0': Interrupción de fallo en el oscilador no habilitada
  - \* OSCFIE='1': Interrupción de fallo en el oscilador habilitada
- **CMIE**: Bit de habilitación de la interrupción del comparador
  - \* CMIE='0': Interrupción del comparador no habilitada
  - \* CMIE='1': Interrupción del comparador habilitada
- **USBIE**: Bit de habilitación de la interrupción del USB
  - \* USBIE='0': Interrupción del USB no habilitada
  - \* USBIE='1': Interrupción del USB habilitada
- **EEIE**: Bit de habilitación de la interrupción de escritura en las memorias EEPROM/ Flash
  - \* EEIE='0': Interrupción de escritura en las memorias EEPROM/ Flash no habilitada
  - \* EEIE='1': Interrupción de escritura en las memorias EEPROM/ Flash habilitada
- **BCLIE**: Bit de habilitación de la interrupción de colisión de bus en el SSP
  - \* BCLIE='0': Interrupción de colisión de bus en el SSP no habilitada
  - \* BCLIE='1': Interrupción de colisión de bus en el SSP habilitada



## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

---

#### REGISTRO PIE2 (cont.)

- **HLVIE**: Bit de habilitación de la interrupción de detección de tensión alta/baja
  - \* HLVIE='0': Interrupción de detección de tensión alta/baja no habilitada
  - \* HLVIE='1': Interrupción de detección de tensión alta/baja habilitada
- **TMR3IE**: Bit de habilitación de la interrupción de desbordamiento del Temporizador 3
  - \* TMR3IE='0': Interrupción de desbordamiento del Temporizador 3 no habilitada
  - \* TMR3IE='1': Interrupción de desbordamiento del Temporizador 3 habilitada
- **CCP2IE**: Bit de habilitación de la interrupción del módulo 2 de Comparación/Captura/PWM (CCP2)
  - \* CCP2IE='0': Interrupción del CCP2 no habilitada
  - \* CCP2IE='1': Interrupción del CCP2 habilitada

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

##### REGISTRO IPR1

	L/E-1	L/E-1	L/E-1	L/E-1	L/E-1	L/E-1	L/E-1	L/E-1
IPR1	SPPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP

- **SPPIP**: Bit de selección de prioridad de la interrupción de lectura/escritura en el canal de transmisión de datos en paralelo (SPP)
  - \* SPPIE='0': Prioridad baja para la interrupción de lectura/escritura en el SPP
  - \* SPPIE='1': Prioridad alta para la interrupción de lectura/escritura en el SPP
- **ADIP**: Bit de selección de prioridad de la interrupción del convertidor A/D
  - \* ADIE='0': Prioridad baja para la interrupción del convertidor A/D
  - \* ADIE='1': Prioridad alta para la interrupción del convertidor A/D
- **RCIP**: Bit de selección de prioridad de la interrupción de recepción del EUSART
  - \* RCIE='0': Prioridad baja para la interrupción de recepción del EUSART
  - \* RCIE='1': Prioridad alta para la interrupción de recepción del EUSART
- **TXIP**: Bit de selección de prioridad de la interrupción de transmisión del EUSART
  - \* TXIE='0': Prioridad baja para la interrupción de transmisión del EUSART
  - \* TXIE='1': Prioridad alta para la interrupción de transmisión del EUSART
- **SSPIP**: Bit de selección de prioridad de la interrupción del puerto serie síncrono (SSP)
  - \* SSPIE='0': Prioridad baja para la interrupción del puerto serie síncrono (SSP)
  - \* SSPIE='1': Prioridad alta para la interrupción del puerto serie síncrono (SSP)

## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

---

#### REGISTRO IPR1 (cont.)

- **CCP1IP**: Bit de selección de prioridad de la interrupción del módulo 1 de Comparación/Captura/PWM (CCP1)
  - \* CCP1IE='0': Prioridad baja para la interrupción del CCP1
  - \* CCP1IE='1': Prioridad alta para la interrupción del CCP1
- **TMR2IP**: Bit de selección de prioridad de la interrupción de igualdad entre TMR2 y PR2
  - \* TMR2IE='0': Prioridad baja para la interrupción de igualdad entre TMR2 y PR2
  - \* TMR2IE='1': Prioridad alta para la interrupción de igualdad entre TMR2 y PR2
- **TMR1IP**: Bit de selección de prioridad de la interrupción de desbordamiento del Temporizador 1
  - \* TMR1IE='0': Prioridad baja para la interrupción de desbordamiento del Temporizador 1
  - \* TMR1IE='1': Prioridad alta para la interrupción de desbordamiento del Temporizador 1

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

#### REGISTRO IPR2

	L/E-1	L/E-1	L/E-1	L/E-1	L/E-1	L/E-1	L/E-1	L/E-1
IPR2	OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP

- **OSCFIP**: Bit de selección de prioridad de la interrupción de fallo en el oscilador
  - \* OSCFIE='0': Prioridad baja para la interrupción de fallo en el oscilador
  - \* OSCFIE='1': Prioridad alta para interrupción de fallo en el oscilador
- **CMIP**: Bit de selección de prioridad de la interrupción del comparador
  - \* CMIE='0': Prioridad baja para la interrupción del comparador
  - \* CMIE='1': Prioridad alta para la interrupción del comparador
- **USBIP**: Bit de selección de prioridad de la interrupción del USB
  - \* USBIE='0': Prioridad baja para la interrupción del USB
  - \* USBIE='1': Prioridad alta para la interrupción del USB
- **EEIP**: Bit de selección de prioridad de la interrupción de escritura en las memorias EEPROM/ Flash
  - \* EEIE='0': Prioridad baja para la interrupción de escritura en las memorias EEPROM/ Flash
  - \* EEIE='1': Prioridad alta para la interrupción de escritura en las memorias EEPROM/ Flash
- **BCLIP**: Bit de selección de prioridad de la interrupción de colisión de bus en el SSP
  - \* BCLIE='0': Prioridad baja para la interrupción de colisión de bus en el SSP
  - \* BCLIE='1': Prioridad alta para la interrupción de colisión de bus en el SSP

## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

---

#### REGISTRO PIE2 (cont.)

- **HLVIP**: Bit de selección de prioridad de la interrupción de detección de tensión alta/baja
  - \* HLVE=‘0’: Prioridad baja para la interrupción de detección de tensión alta/baja
  - \* HLVE=‘1’: Prioridad alta para la interrupción de detección de tensión alta/baja
- **TMR3IP**: Bit de selección de prioridad de la interrupción de desbordamiento del Temporizador 3
  - \* TMR3IE=‘0’: Prioridad baja para la interrupción de desbordamiento del Temporizador 3
  - \* TMR3IE=‘1’: Prioridad alta para la interrupción de desbordamiento del Temporizador 3
- **CCP2IP**: Bit de selección de prioridad de la interrupción del módulo 2 de Comparación/Captura/PWM (CCP2)
  - \* CCP2IE=‘0’: Prioridad baja para la interrupción del CCP2
  - \* CCP2IE=‘1’: Prioridad alta para la interrupción del CCP2

## TEMA 3: MICROCONTROLADOR PIC18F4550

### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

#### PRIORIDAD DE LAS INTERRUPCIONES:

⇒ El bit IPEN del registro RCON permite activar el sistema de prioridades:

- IPEN a '0': todas las interrupciones tienen la misma prioridad y se vectorizan en la dirección 0008H. Para habilitar globalmente las interrupciones del grupo general de interrupciones debe ponerse a '1' el GIE del registro INTCON. Para habilitar globalmente las interrupciones del bloque de interrupciones de periféricos deben ponerse a '1' los bits GIE y PEIE del registro INTCON. En este caso, ninguna interrupción puede interrumpir el tratamiento de otra interrupción
- IPEN a '1': las interrupciones pueden configurarse con dos niveles de prioridad en función del bit de prioridad de interrupción (excepto la interrupción externa 0 que siempre tendrá nivel de prioridad alto):
  - Bit de prioridad de interrupción a '0' => **Prioridad baja**. Las interrupciones de prioridad baja se vectorizan en la dirección 0018H. Para habilitar globalmente las interrupciones de prioridad baja hay que poner a '1' el bit GIEL del registro INTCON.
  - Bit de prioridad de interrupción a '1' => **Prioridad alta**. Las interrupciones de prioridad alta se vectorizan en la dirección 0008H. Para habilitar globalmente las interrupciones de prioridad alta hay que poner a '1' el bit GIEH del registro INTCON.

Una interrupción de alta prioridad puede interrumpir el tratamiento de una interrupción de baja prioridad. Sin embargo las interrupciones de baja prioridad no pueden interrumpir a las interrupciones de alta prioridad. Por otra parte las interrupciones de baja prioridad no pueden interrumpirse entre ellas. Lo mismo ocurre con las interrupciones de alta prioridad.

**NOTA:** Además de ser habilitadas globalmente, las interrupciones deben ser habilitadas a nivel particular mediante su bit de habilitación de interrupción.

## **TEMA 3: MICROCONTROLADOR PIC18F4550**

### **DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES**

#### **PASOS A SEGUIR PARA TRABAJAR CON INTERRUPCIONES:**

- 1. Configurar el dispositivo asociado a la interrupción (Temporizadores, EUSART, convertidor A/D, etc..).**
- 2. Habilitar a nivel global las interrupciones del grupo correspondiente mediante los bits GIE/GIEH y PEIE/GIEL del registro INTCON**
- 3. Habilitar a nivel individual la interrupción mediante el bit de habilitación.**
- 4. Establecer la prioridad de la interrupción mediante el bit de prioridad (solo si previamente se ha activado el sistema de prioridad de interrupciones, IPEN='1')**
- 5. En la dirección de vectorización correspondiente (0008H o 0018H, según el caso) añadir el código de tratamiento de la interrupción que debe incluir:**
  - **Identificación de la interrupción:** dado que varias interrupciones pueden vectorizarse en la misma dirección, se deben comprobar los flags de las interrupciones habilitadas para saber cual de ellas ha provocado el salto a la dirección de vectorización.
  - **Borrado del flag de interrupción:** el flag de interrupción debe ser borrado por software una vez la interrupción ha sido identificada. De esta forma se evita que cuando finalice la rutina de tratamiento se vuelva a procesar la misma interrupción.
  - **Código de procesado de la interrupción:** por último hay que escribir el código de tratamiento de la interrupción que dependerá de cada aplicación.

**El código de tratamiento de la interrupción siempre debe terminar con la instrucción RETFIE.**



## **TEMA 3: MICROCONTROLADOR PIC18F4550**

### **DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES**

#### **PROCESO DE TRATAMIENTO DE UNA INTERRUPCION:**

- 1. Se produce la condición de disparo de la interrupción (desbordamiento del temporizador, recepción de un dato en la EUSART, etc...).**
- 2. Si la interrupción está habilitada, el uC almacena en la pila la dirección de la instrucción que debía haberse ejecutado a continuación y escribe en PC la dirección de vectorización de la interrupción. Además el uC pone el bit de habilitación global de interrupciones (GIE, GIEH o GIEL según corresponda) a '0' para que ninguna interrupción del mismo nivel que la se está tratando puede interrumpir el proceso.**
- 3. Se ejecuta el código de tratamiento de la interrupción que deberá hallarse en la dirección de la vectorización (se comprueba que interrupción ha generado el salto a la dirección de vectorización, se borra el flag de interrupción correspondiente y se ejecuta el código asociado a dicha interrupción.**
- 4. La última instrucción del código de tratamiento de la interrupción es RETFIE. Cuando se ejecuta esta instrucción, el uC vuelve a poner a '1' el bit de habilitación global de interrupciones (GIE, GIEH o GIEL) y escribe en el PC la dirección que había almacenado en la pila de forma que se continúe con la ejecución del programa a partir del punto en el que había sido interrumpido.**



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

#### EJEMPLO DE TRAMIENTO DE UNA INTERRUPCION:

El siguiente código ilustra como gestionar la interrupción mediante el Temporizador 0. Se considera que no están activadas las prioridades (IPEN='0')

```
ORG 0x0000      ; Vectorización del reset
GOTO PPAL       ; Cuando se produce el reset se salta la principio del programa principal

ORG 0x0008      ; Vectorización de las interrupciones
IT BTFSS INTCON,TMR0IF ; Se comprueba el flag de interrupción del Temporizador 0
GOTO FIN_IT     ; Si no está a '1' se salta al final
BCF INTCON,TMR0IF ; Si se ha producido interrupción del Temporizador 0 se pone a '0' el flag
                  ; y a continuación se ejecuta el código de procesado de la interrupción
                  ; CODIGO CORRESPONDIENTE
                  ; AL PROCESADO DE LA INTERRUPCION
FIN_IT
RETIE           ; Retorno de la interrupción

PPAL:
MOVLW 0xC0      ; Se configura el temporizador (modo temporizador de 8 bits) y
MOVWF T0CON     ; se pone en marcha
BSF INTCON,GIE  ; Se habilitan globalmente las interrupciones
BSF INTCON,TMR0IE ; Se habilita individualmente la interrupción del Temporizador 0

BUCLE
GOTO BUCLE
```

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### DESCRIPCION FUNCIONAL / SISTEMA DE INTERRUPCIONES

#### CONSERVACIÓN DEL VALOR DE LOS REGISTROS EN INTERRUPCIONES:

Las interrupciones se disparan durante la ejecución de código del programa principal o de otra interrupción. Esto hace que durante la ejecución de la rutina de tratamiento de la interrupción se pueda modificar el valor de registros que está siendo utilizados por otras partes del código. Para evitar que estas modificaciones alteren el correcto funcionamiento del sistema conviene almacenar los valores de estos registros al inicio de la interrupción para recuperarlos al final

⇒ **Mediante la pila rápida de registro (solo WREG, STATUS y BSR)**

```
ORG 0x0008 ; Cuando se vectoriza la interrupción WREG, BSR y STATUS se almacenan
.          ; CODIGO DE LA INTERRUPCION (los registros pueden ser modificados sin que eso
.          ; afecte al resto del programa)
.          ;
.          ;
RETFIE FAST ; Al acabar se recuperan los valores de WREG, STATUS y BSR de la pila rápida de registro
```

⇒ **Mediante variables auxiliares**

```
ORG 0x0008 ; Vectorizacion de las interrupciones
MOVFF STATUS,STATUS_TEMP ; Se almacenan en variables auxiliares los valores de los
MOVFF WREG,WREG_TEMP ; registros cuya modificación durante la ejecución del código
MOVFF BSR,BSR_TEMP ; de la interrupción pueda afectar al resto del código
. ; CODIGO DE LA INTERRUPCION (los registros pueden ser
. ; modificados si que eso afecte al resto del programa)
. ;
MOVFF BSR_TEMP,BSR ; Antes de acabar la interrupción se recuperan los valores
MOVFF WREG_TEMP,WREG ; de los registros almacenados temporalmente al inicio de la
MOVFF STATUS_TEMP,STATUS ; misma
RETFIE ; Retorno de la interrupción
```

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / INTRODUCCIÓN

**El uC PIC18F4550 dispone de una serie de Unidades Funcionales que le permiten:**

- **Realizar tareas específicas especializadas (conversion A/D, transmisión/recepción de datos, generación de señales digitales con temporizaciones programables, etc...)**
- **Optimizar el rendimiento del uC, ya que estas unidades trabajan en paralelo a la CPU permitiendo que ésta se centre en otras tareas como procesamiento de datos, cálculos, movimiento de datos, etc...**

**Las Unidades Funcionales más importantes del uC PIC18F4550 son:**

Puertos de E/S	Unidad de Comparación/Captura/PWM mejorada (ECCP)
Temporizador 0	Canal de comunicación serie EUSART
Temporizador 1	Canal de comunicación serie MSSP
Temporizador 2	Canal de comunicación serie USB
Temporizador 3	Módulo analógico de comparación
Convertidor A/D	Canal de transmisión de datos en paralelo (SPP)
Unidad de Comparación/Captura/PWM (CCP)	

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / PUERTOS DE E/S

##### PUERTOS DE E/S:

**El uC PIC18F4550 dispone 5 puertos de E/S que incluyen un total de 35 líneas digitales de E/S:**

PUERTO	LINEAS DE ENTRADA/SALIDA
PORTA	7 LINEAS DE ENTRADA/SALIDA
PORTB	8 LINEAS DE ENTRADA/SALIDA
PORTC	6 LINEAS DE ENTRADA/SALIDA+ 2 LINEAS DE ENTRADA
PORTD	8 LINEAS DE ENTRADA/SALIDA
PORTE	3 LINEAS DE ENTRADA/SALIDA + 1 LINEAS DE ENTRADA

**Todas las líneas digitales de E/S disponen de al menos una función alternativa asociada a alguna circuiteria específica del uC. Cuando una línea trabaja en el modo alternativo no puede ser utilizada como línea digital de E/S estándar.**

### TEMA 3: MICROCONTROLADOR PIC18F4550

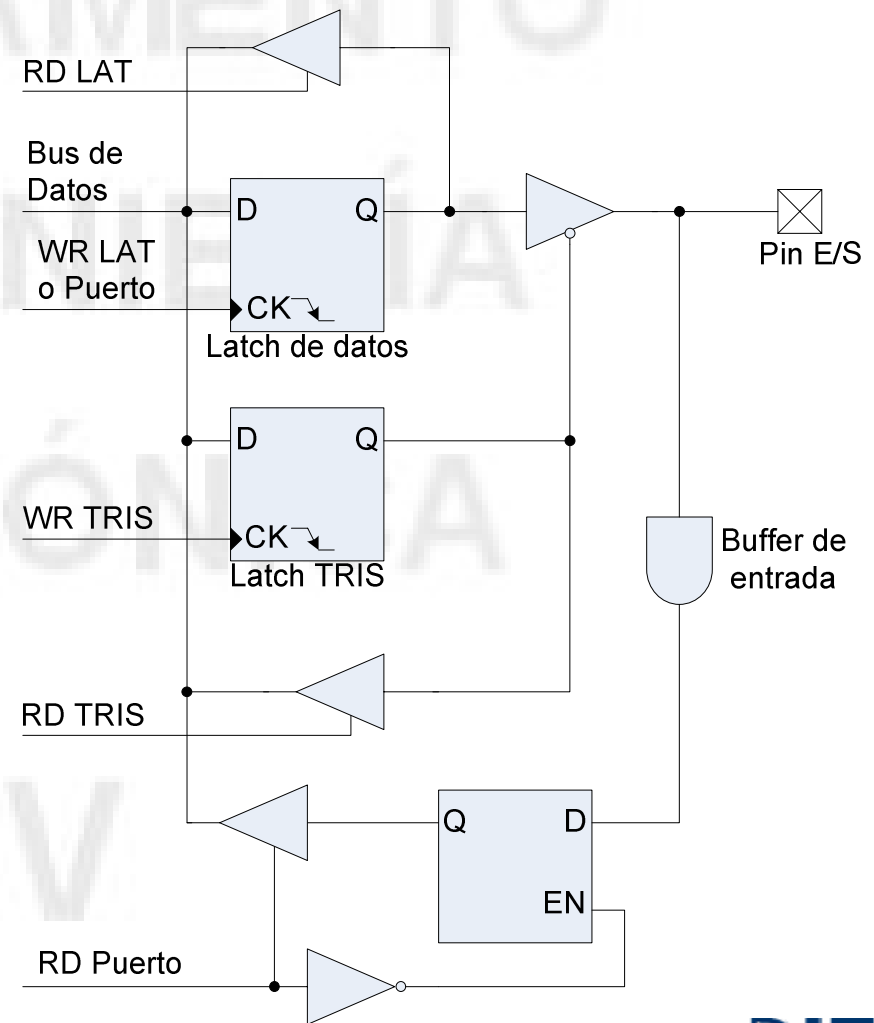
#### UNIDADES FUNCIONALES / PUERTOS DE E/S

##### REGISTROS DE UN PUERTO DE E/S:

Cada puerto de E/S tiene asociado 3 registros:

- Registro **TRIS**: mediante este registro se configuran cada una de las líneas de E/S del puerto como **ENTRADA** (bit correspondiente a '1') o como **SALIDA** (bit correspondiente a '0').
- Registro **PORT**: mediante este registro se puede leer el nivel de pin de E/S y se puede establecer el valor del latch de salida.
- Registro **LAT**: mediante este registro se puede leer o establecer el valor del latch de salida

##### DIAGRAMA DE BLOQUES GENÉRICO DE UN PIN DE E/S:



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / PUERTOS DE E/S

#### PUERTO A:

Dispone de 7 líneas de E/S. Las funciones alternativas son:

- **RA0:** entrada analógica (AN0)/ entrada de comparación (C1IN-)
- **RA1:** entrada analógica (AN1)/ entrada de comparación (C2IN-)
- **RA2:** entrada analógica (AN2)/ entrada de comparación (C2IN+)
- **RA3:** entrada analógica (AN3)/ entrada de comparación (C1IN+)
- **RA4:** entrada de reloj del Temporizador 0 (T0CKI)/salida de comparación (C1OUT)
- **RA5:** entrada analógica (AN4)/ salida de comparación (C2OUT)/HLVDIN  
entrada de detección de tensión alta/baja
- **RA6:** entrada del oscilador principal (OSC2)/salida de señal de reloj (CLK0)

En el reset las líneas RA0, RA1, RA2, RA3 y RA5 se configuran como líneas de entrada analógicas. Para poder utilizarlas como líneas digitales de E/S hay que desactivar la función analógica:

<b>MOVLW 0FH</b>	<b>; Se desactiva la función de entrada analógica</b>
<b>MOVWF ADCON1</b>	<b>;</b>
<b>MOVLW 07H</b>	<b>; para las líneas RA0, RA1, RA2, RA3 y RA4</b>
<b>MOVWF CMCON</b>	<b>; Configura los comparadores para entrada digital.</b>
<b>MOVLW 3CH</b>	<b>;</b>
<b>MOVWF TRISA</b>	<b>; para las líneas RA0, RA1, RA2 y RA3</b>
	<b>; Se configuran RA5, RA4, RA3 y RA2 como entradas</b>
	<b>; y RA1 y RA0 como salidas</b>

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / PUERTOS DE E/S

##### REGISTRO ADCON1

	-0	-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
ADCON1	-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0

- **VCFG1**: Bit de configuración de la tensión de referencia  $V_{REF-}$ :
  - \* VCFG1='0':  $V_{REF-}$  se conecta a  $V_{SS}$
  - \* VCFG1='1':  $V_{REF-}$  se conecta a la línea física RA2
- **VCFG0**: Bit de configuración de la tensión de referencia  $V_{REF+}$ :
  - \* VCFG0='0':  $V_{REF+}$  se conecta a  $V_{DD}$
  - \* VCFG0='1':  $V_{REF+}$  se conecta a la línea física RA2
- **PCFG3..PCFG0**: Bits configuración de los puertos de conversión A/D. Mediante estos bits se establecen que líneas físicas (RA5..RA0, RB4..RB0, RE1 y RE0) van a trabajar como entradas del convertidor A/D (Ver Tabla de configuración de líneas de conversión A/D).

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

**Tabla de configuración de líneas de conversión A/D:**

PCFG3..PCFG0	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
0000	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / PUERTOS DE E/S

#### REGISTRO CMCOM

	R-0	R-0	L/E-0	L/E-0	L/E-0	L/E-1	L/E-1	L/E-1
CMCOM	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0

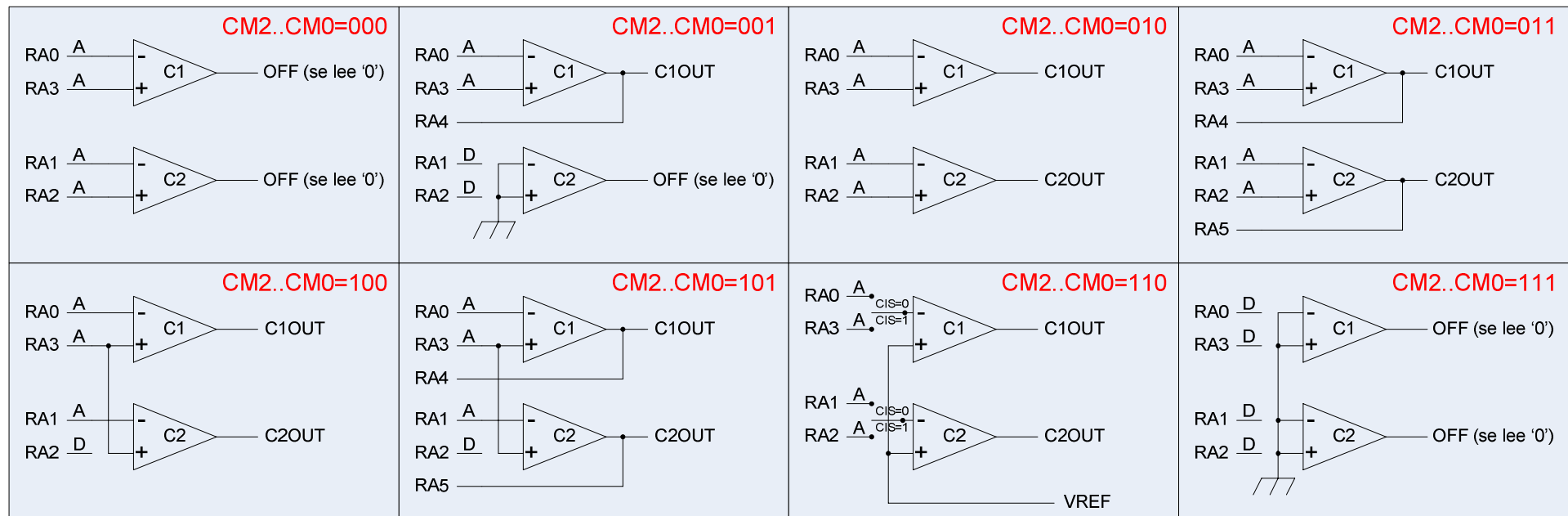
- **C2OUT**: Bit de salida del comparador 2:
  - Cuando C2INV='0'
    - Si  $C2\_Vin+ > C2\_Vin \Rightarrow C2OUT='1'$
    - Si  $C2\_Vin+ < C2\_Vin \Rightarrow C2OUT='0'$
  - Cuando C2INV='1'
    - Si  $C2\_Vin+ < C2\_Vin \Rightarrow C2OUT='1'$
    - Si  $C2\_Vin+ > C2\_Vin \Rightarrow C2OUT='0'$
- **C1OUT**: Bit de salida del comparador 1:
  - Cuando C1INV='0'
    - Si  $C1\_Vin+ > C1\_Vin \Rightarrow C1OUT='1'$
    - Si  $C1\_Vin+ < C1\_Vin \Rightarrow C1OUT='0'$
  - Cuando C1INV='1'
    - Si  $C1\_Vin+ < C1\_Vin \Rightarrow C1OUT='1'$
    - Si  $C1\_Vin+ > C1\_Vin \Rightarrow C1OUT='0'$
- **C2INV**: Bit de inversión del comparador 2
  - \* C2INV='0': Comparador C2 NO invertido
  - \* C2INV='1': Comparador C2 invertido

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / PUERTOS DE E/S

#### REGISTRO CMCOM

- **C2INV**: Bit de inversión del comparador 2
  - \* C2INV='0': Comparador C2 NO invertido
  - \* C2INV='1': Comparador C2 invertido
- **C1S**: Bit de conmutación de la entrada del comparador (CM2..CM0=110).
  - \* CIS='0': Pin C1\_Vin+ conectado a pin RA3/ Pin C2\_Vin+ conectado a pin RA2
  - \* CIS='1': Pin C1\_Vin+ conectado a pin RA0/ Pin C2\_Vin+ conectado a pin RA1
- **CM2..CM0**: Bits de selección del modo de comparación



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / PUERTOS DE E/S

##### **PUERTO B:**

**Dispone de 8 líneas de E/S. Las funciones alternativas son:**

- **RB0:** entrada analógica (AN12)/ interrupción externa 0 (INT0)/entrada de fallo del ECCP (FLT0)/entrada de datos del SPI (SDI)/línea de datos del I<sup>2</sup>C (SDA)
- **RB1:** entrada analógica (AN10)/ interrupción externa 1 (INT1)/línea de reloj del SPI (SDI)/línea de reloj del I<sup>2</sup>C (SDA)
- **RB2:** entrada analógica (AN8)/ interrupción externa 2 (INT2)/salida de datos del USB (VCMO)
- **RB3:** entrada analógica (AN9)/ línea de E/S del CCP2 (CCP2)/salida de datos del USB (VPO)
- **RB4:** entrada analógica (AN11)/ interrupción por cambio en pin (KBI0)/ salida de CS del SSP (CSSP)
- **RB5:** interrupción por cambio en pin (KBI1)/ línea de programación (PGM)
- **RB6:** interrupción por cambio en pin (KBI2)/ línea de programación (PGC)
- **RB7:** interrupción por cambio en pin (KBI3)/ línea de programación (PGD)

**Resistencias de pull.up:** Todas las líneas del puerto B disponen de resistencias de pull-up internas que pueden ser activadas poniendo el bit RBPU del registro INTCON2 a '0' (RPBU='1' después de un reset). Si una línea del puerto B se configura como salida la resistencia de pull-up correspondiente se desactiva automáticamente.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / PUERTOS DE E/S

##### PUERTO B (cont.):

Por defecto, en el reset las líneas RB4..RB0 están programadas como entradas analógicas. Existen dos formas de configurar RB4..RB0 como líneas de E/S digitales:

- Poniendo a '0' el bit PBADEN del registro de configuración CONFIG3H=> en el reset RB4..RB0 se configuran como líneas de E/S digitales
- Si PBADEN='1' (valor por defecto) se pueden configurar RB4..RB0 como líneas del E/S digitales desactivando la función analógica:

MOVLW 0x07	; Se desactiva la función de entrada analógica
MOVWF ADCON1	; para las líneas RB0, RB1, RB2, RB3 y RB4
MOVLW 0xF0	; Se configuran RB7, RB6, RB5 y RB4 como entradas
MOVWF TRISB	; y RB3, RB2, RB1 y RB0 como salidas

### **INTERRUPCIONES DEL PUERTO B:**

- **Interrupción externa 0:**
  - Flag de interrupción: bit INT0IF (INTCON)
  - Bit de habilitación: bit INT0IE (INTCON)
  - Selección de flanco: bit INTEDG0 (INTCON2): ‘0’->fl. bajada/’1’-> fl. subida
  - Prioridad: la interrupción 0 siempre tiene prioridad alta

Si se produce un flanco en la línea RB0 (flanco de subida o de baja dependiendo del valor del bit INTEDG0) se pone el flag INT0IF a ‘1’. Si el bit de habilitación INT0IE está a ‘1’ y las interrupciones están habilitadas a nivel global se genera una interrupción y el uC pasa a ejecutar el código situado a partir de la posición 0008H.

- **Interrupción externa 1:**
  - Flag de interrupción: bit INTCON3.**INT1IF**
  - Bit de habilitación: bit INTCON3.**INT1IE**
  - Selección de flanco: bit INTCON2.**INTEDG1**: ‘0’->fl. bajada/’1’-> fl. subida
  - Prioridad: bit INT1IP (INTCON3): ‘0’->pr. baja/’1’->pr. alta

Si se produce un flanco en la línea RB1 (flanco de subida o de baja dependiendo del valor del bit INTEDG1) se pone el flag INT1IF a ‘1’. Si el bit de habilitación INT1IE está a ‘1’ y las interrupciones están habilitadas a nivel global se genera una interrupción y el uC pasa a ejecutar el código situado a partir de la posición 0008H o 0018H (según el nivel de prioridad establecido).

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / PUERTOS DE E/S

#### INTERRUPCIONES DEL PUERTO B (cont.):

- Interrupción externa 2:
  - Flag de interrupción: bit INTCON3.**INT2IF**
  - Bit de habilitación: bit INTCON3.**INT2IE**
  - Selección de flanco: bit INTCON2.**INTEDG2**: ‘0’->fl. bajada/’1’-> fl. subida
  - Prioridad: bit INTCON3.**INT2IP**: ‘0’->pr. baja/’1’->pr. alta

Si se produce un flanco en la línea RB2 (flanco de subida o de baja dependiendo del valor del bit INTEDG2) se pone el flag INT2IF a ‘1’. Si el bit de habilitación INT2IE está a ‘1’ y las interrupciones están habilitadas a nivel global se genera una interrupción y el uC pasa a ejecutar el código situado a partir de la posición 0008H o 0018H (según el nivel de prioridad establecido).



**INTERRUPCIONES DEL PUERTO B (cont.):**

- **Interrupción por cambio en pines RB7..RB4:**
  - **Flag de interrupción: bit RBIF (INTCON)**
  - **Bit de habilitación: bit RBIE (INTCON)**
  - **Prioridad: bit RBIP (INTCON2): ‘0’->pr. baja/‘1’->pr. alta**

**Solo puede asociarse a las líneas RB7..RB4 que hayan sido configuradas como entrada. Cada vez que se produce un cambio en estas líneas se pone a ‘1’ el flag RBIF. Si el bit de habilitación RBIE está a ‘1’ y las interrupciones están habilitadas a nivel global se genera una interrupción y el uC pasa a ejecutar el código situado a partir de la posición 0008H ó 0018H (según el nivel de prioridad establecido).**

**Si se utiliza esta interrupción no es aconsejable hacer “**polling**” del puerto B. Antes de salirse de la rutina de tratamiento de la interrupción hay que hacer lo siguiente:**

- **Acceder al puerto B con cualquier instrucción excepto MOVFF, para deshacer la desigualdad entre valor antiguo y nuevo valor**
- **Poner a ‘0’ el bit RBIF**

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / PUERTOS DE E/S

---

##### PUERTO C:

Dispone de 5 líneas de E/S (RC0, RC1, RC2, RC6 y RC7) y 2 líneas de solo entrada (RC4 y RC5). Las funciones alternativas son:

- **RC0:** salida del oscilador del Temp. 1 (T1OSO)/ entrada de contador de los Temp. 1 y 3 (T13CKI)
- **RC1:** entrada del oscilador del Temp. 1 (T1OSI)/ línea de E/S del CCP2 (CCP2)/ salida OE del transceiver del USB (UOE)
- **RC2:** línea de E/S del CCP1 (CCP1)/ salida PWM del ECCP1 (P1A)
- **RC4:** línea menos del bus USB (D-) / línea de entrada del USB (VM)
- **RC5:** línea más del bus USB (D-) / línea de entrada del USB (VP)
- **RC6:** salida de transmisión del EUSART (TX)/ línea de reloj del EUSART (CK)
- **RC7:** entrada de recepción del EUSART (RX)/ línea de datos síncrona del EUSART (DT)/ salida de datos del SPI (SDO)

En el reset todas las líneas del puerto C quedan configuradas como entradas digitales.



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / PUERTOS DE E/S

---

##### **PUERTO D:**

**Dispone de 8 líneas de E/S. Las funciones alternativas son:**

- **RD0: línea de datos del SPP (SPP0)**
- **RD1: línea de datos del SPP (SPP1)**
- **RD2: línea de datos del SPP (SPP2)**
- **RD3: línea de datos del SPP (SPP3)**
- **RD4: línea de datos del SPP (SPP4)**
- **RD5: línea de datos del SPP (SPP5) / salida PWM del ECCP1 (P1B)**
- **RD6: línea de datos del SPP (SPP6) / salida PWM del ECCP1 (P1C)**
- **RD7: línea de datos del SPP (SPP7) / salida PWM del ECCP1 (P1D)**

**Resistencias de pull.up:** Todas las líneas del puerto D disponen de resistencias de pull-up internas que pueden ser activadas poniendo el bit RDPUR del registro PORTE a '1' (RDPUR='0' después de un reset). Si una línea del puerto D se configura como salida la resistencia de pull-up correspondiente se desactiva automáticamente.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / PUERTOS DE E/S

##### PUERTO E:

Dispone de 3 líneas de E/S (RE0, RE1 y RE2) y 1 línea de solo entrada (RE3). Las funciones alternativas son:

- RE0: entrada analógica (AN5)/ salida de reloj 1 del SPP (CK1SPP)
- RE1: entrada analógica (AN6)/ salida de reloj 2 del SPP (CK2SPP)
- RE2: entrada analógica (AN7)/ salida de habilitación del SPP (OESPP)
- RE3: Línea de reset externo (MCLR) / línea de programación (VPP)

En el reset todas las líneas RE2..RE0 se configuran como entradas analógicas. Para poder utilizarlas como líneas digitales de E/S hay que desactivar la función analógica:

MOVLW 0FH	; Se desactiva la función de entrada analógica
MOVWF ADCON1	; para las líneas RE0, RE1 y RE2
MOVLW 06H	; Se configuran RE2 y RE1 como entradas
MOVWF TRISE	; y RE0 como salida

La línea RE3 por defecto tiene la función de Reset del uC. Si se desea desactivar la función de Reset y utilizar RE3 como línea de entrada digital hay que poner a '0' el bit MCLRE del registro de configuración CONFIG3H.

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 0

---

#### **TEMPORIZADOR 0:**

##### **Características fundamentales:**

- **Configurable como temporizador/contador de 8 bits/16 bits**
- **Pre-escalar de 8 bits programable**
- **Interrupción por desbordamiento**



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 0

#### REGISTRO T0CON



- **TMR0ON**: Bit de puesta en marcha del Temporizador 0
- **T08BIT**: Bit de configuración del modo 8-bit/16-bit:
  - \* T08BIT='0': Modo 16-bit
  - \* T08BIT='1': Modo 8-bit
- **T0CS**: Bit de configuración del modo contador/temporizador
  - \* T0CS='0': Modo temporizador (TMR0 se incrementa en cada ciclo de instrucción  $F_{osc}/4$ )
  - \* T0CS='1': Modo contador (TMR0 se incrementa en cada transición de la línea RA4/T0CKI)
- **T0SE**: Bit de selección de flanco en modo contador
  - \* T0SE='0': TMR0 se incrementa en los flancos de subida de la línea RA4/T0CKI
  - \* T0SE='1': TMR0 se incrementa en los flancos de bajada de la línea RA4/T0CKI
- **PSA**: Bit de activación del pre-escalar
  - \* PSA='0': Pre-escalar activado
  - \* PSA='1': Pre-escalar no activado
- **T0PS2..T0PS0**: Bits de selección del pre-escalar del Temporizador 0:

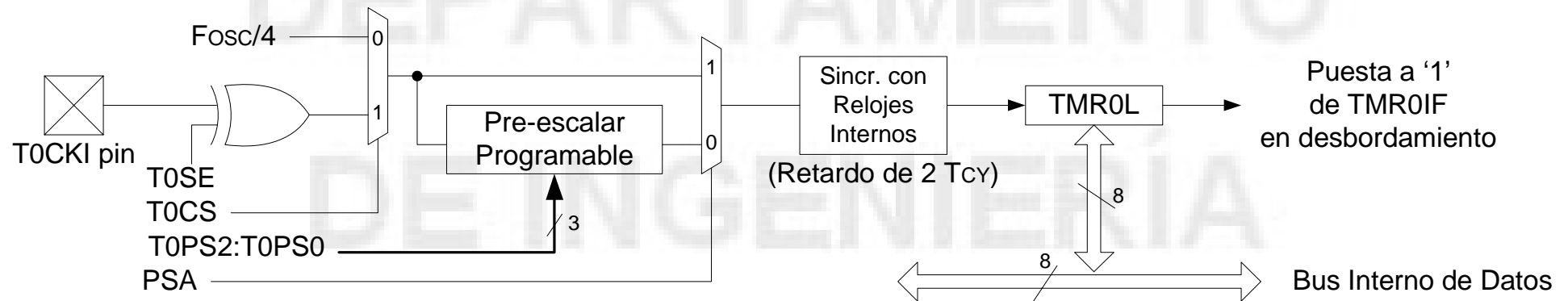
T0PS2	T0PS1	T0PS0	Valor del pre-escalar
0	0	0	1:2
0	0	1	1:4
0	1	0	1:8
0	1	1	1:16
1	0	0	1:32
1	0	1	1:64
1	1	0	1:128
1	1	1	1:256



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / TEMPORIZADOR 0

##### Temporizador 0 en modo de 8 bits:



Se trabaja sobre **TMR0L** que es la parte baja del registro del temporizador 0 **TMR0**

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 0

#### Modo temporizador de 8 bits:

- **T08BIT (T0CON)='1'/T0CS (T0CON)='0'**
- **Dos opciones:**
  - Sin pre-escalar PSA (T0CON)='0': el registro TMR0L se incrementa con cada ciclo de instrucción ( $F_{OSC}/4$ ).
  - Con pre-escalar PSA (T0CON)='1': el contador de pre-escalar se incrementa en cada ciclo de instrucción ( $F_{OSC}/4$ ). Cuando el pre-escalar alcanza el valor programado en los bits PS2..PS0 (T0CON) produce un pulso que incrementa el registro TMR0L.
- Cuando se produce desbordamiento de TMR0L se pone a '1' el flag TMR0IF

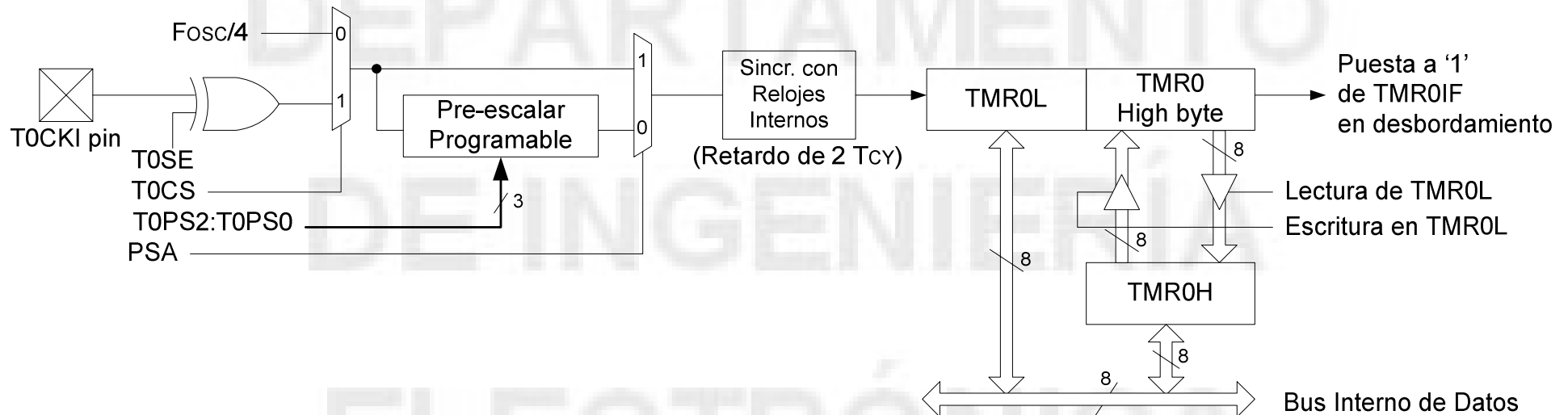
#### Modo contador de 8 bits:

- **T08BIT (T0CON)='1'/T0CS (T0CON)='1'**
- **Dos opciones:**
  - Sin pre-escalar PSA (T0CON)='0': el registro TMR0L se incrementa con cada flanco de subida (T0SE='0') o de bajada (T0SE='0') de la línea T0CKI (RA4).
  - Con pre-escalar PSA (T0CON)='1': el contador de pre-escalar se incrementa con cada flanco de subida (T0SE='0') o de bajada (T0SE='0') de la línea T0CKI (RA4). Cuando el pre-escalar alcanza el valor programado en los bits PS2..PS0 (T0CON) se produce un pulso que incrementa el registro TMR0L.
- Cuando se produce desbordamiento de TMR0L se pone a '1' el flag TMR0IF

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 0

#### Temporizador 0 en modo de 16 bits:



En este caso se trabaja sobre el registro TMR0 completo.

La parte alta de TMR0 no es accesible directamente. Se accede a ella a través de TMR0H que funciona como un búfer:

- Cuando se lee el valor de TMR0L, el valor de la parte alta de TMR0 pasa al registro TMR0H. Por tanto para leer el valor de TMR0 primero leemos TMR0L y luego leemos TMR0H.
- Cuando se escribe en TMR0L, el valor del registro TMR0H pasa a la parte alta de TMR0. Por tanto para escribir en TMR0 primero escribimos el valor de la parte alta en TMR0H y luego escribimos el valor de la parte baja en TMR0L.

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 0

#### Modo temporizador de 16 bits:

- **T08BIT (T0CON)='1'/T0CS (T0CON)='0'**
- **Dos opciones:**
  - Sin pre-escalar PSA (T0CON)='0': el registro TMR0 (16 bits) se incrementa con cada ciclo de instrucción ( $F_{OSC}/4$ ).
  - Con pre-escalar PSA (T0CON)='1': el contador de pre-escalar se incrementa en cada ciclo de instrucción ( $F_{OSC}/4$ ). Cuando el pre-escalar alcanza el valor programado en los bits PS2..PS0 (T0CON) produce un pulso que incrementa el registro TMR0.
- **Cuando se produce desbordamiento de TMR0 se pone a '1' el flag TMR0IF**

#### Modo contador de 16 bits:

- **T08BIT (T0CON)='1'/T0CS (T0CON)='1'**
- **Dos opciones:**
  - Sin pre-escalar PSA (T0CON)='0': el registro TMR0 se incrementa con cada flanco de subida (T0SE='0') o de bajada (T0SE='0') de la línea T0CKI (RA4).
  - Con pre-escalar PSA (T0CON)='1': el contador de pre-escalar se incrementa con cada flanco de subida (T0SE='0') o de bajada (T0SE='0') de la línea T0CKI (RA4). Cuando el pre-escalar alcanza el valor programado en los bits PS2..PS0 (T0CON) se produce un pulso que incrementa el registro TMR0.
- **Cuando se produce desbordamiento de TMR0 se pone a '1' el flag TMR0IF**



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 0

---

#### Interrupción del Temporizador 0:

- Flag de interrupción: bit TMR0IF (INTCON)
- Bit de habilitación: bit TMR0IE (INTCON)
- Prioridad: bit TMR0IP (INTCON2): '0' -> pr. baja / '1' -> pr. alta

Si se produce el desbordamiento del Temporizador 0 se pone a '1' el flag TMR0IF. Si el bit de habilitación TMR0IE está a '1' y las interrupciones están habilitadas a nivel global se genera una interrupción y el uC pasa a ejecutar el código situado a partir de la posición 0008H o 0018H (según el nivel de prioridad establecido).

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / TEMPORIZADOR 0

#### EJERCICIO:

Escribir en código de un programa para el uC PIC18F4550 que genere una señal cuadrada por el pin RD0 de 1kHz de frecuencia mediante la interrupción del Temporizador 0 ( $F_{OSC}=10\text{MHz}$ ).

; Bloque de declaraciones y vectorizaciones

#INCLUDE <p18F4520.inc> ; Declaración de la librería de Reg. de Func. Especial

ORG 0x00 ; Vectorización del Reset

GOTO Inicio

ORG 0x08 ; Vectorización de interrupciones

GOTO IntT0\_ISR ; de alta prioridad

; Programa Principal

Inicio:

MOVLW 0xFE ; Se configura RD0 como salida

MOVWF TRISD ;

BSF PORTD,0 ; Se pone RD0 a '1' inicialmente

MOVLW 0xFB ; Valor inicial de TMR0 para una temporización de 500us ( $T_{TEMP0}/2$ ):

MOVWF TMR0H ;  $TMR0=65536-(T_{SEÑAL}/(T_{TEMP0}*2))=65536-(10\cdot 10^6/(4\cdot 2\cdot 10^3))$

MOVLW 0x1E ;  $TMR0=65536-1250=64286=0xFB1E$

MOVWF TMR0L ;

MOVLW b'10001000' ; Se configura el Temp. en modo temporizador de 16 bits y con el

MOVWF T0CON ; pre-escalar desactivado. Se pone en marcha el Temp. 0

BSF INTCON,TMR0IE ; Se habilita la interrupción del Temporizador 0

BSF INTCON,GIE ; Se habilitan a nivel global las interrupciones

Bucle:

GOTO Bucle ; Bucle infinito



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / TEMPORIZADOR 0

#### EJERCICIO (cont):

; Interrupción del temporizador 0

IntT0\_ISR:

```
BTFSS INTCON,TMR0IF ; Se comprueba si la interrupción se debe al desbordamiento del Temp. 0
GOTO Fin_intT0      ; Si no es así se sale de la interrupción
BCF INTCON,TMR0IF   ; Se pone a '0' el flag de interrupción del Temp. 0
MOVLW 0xFB          ; Valor recarga de TMR0 para la siguiente temporización de 500us
MOVWF TMR0H          ;  $TMR0 = 65536 - (T_{SEÑAL} / (T_{TEMPO} * 2)) = 65536 - (10 \cdot 106 / (4 \cdot 2 \cdot 103))$ 
MOVLW 0x1E          ;  $TMR0 = 65536 - 1250 = 64286 = 0xFB1E$ 
MOVWF TMR0L          ;
BTG PORTD,0         ; Se complementa la línea RD0 para generar la señal cuadrada
```

Fin\_intT0:

RETfie

END

**NOTA:** La temporización no será de 500us exactamente ya que no se ha tenido en cuenta el tiempo de procesado de la interrupción ni el tiempo de ejecución de las instrucciones que se ejecutan antes de recargar el TMR0. Para obtener una temporización más exacta habría que descontar estos tiempo de los 500us.

**¿Cuánto tiempo tarda en ejecutarse el código de la rutina IntT0\_ISR?**

**Repetir este ejercicio en código C**

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 1

---

#### TEMPORIZADOR 1:

##### Características fundamentales:

- Configurable como temporizador/contador de 16 bits
- Dispone de un oscilador propio que puede funcionar como:
  - Señal de reloj del temporizador 1
  - Señal de reloj del uC en modos de bajo consumo
- Pre-escalar de 3 bits programable
- Interrupción por desbordamiento

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / TEMPORIZADOR 1



- **RD16:** Bit selección de modo de lectura/escritura de TMR1
  - \* RD16='0': Lectura/escritura de TMR1 en dos operaciones independientes de 8-bit
  - \* RD16='1': Lectura/escritura de TMR1 en modo de 16-bit
- **T1RUN:** Bit de monitorización de la fuente del reloj principal del sistema (solo lectura):
  - \* T1RUN='0': Reloj principal no derivado del oscilador del Temporizador 1
  - \* T1RUN='1': Reloj principal derivado del oscilador del Temporizador 1
- **T1CKPS1..T1CKPS0:** Bits de selección del pre-escalar del Temporizador 1

T1CKPS1	T1CKPS0	Valor del pre-escalar
0	0	1:1
0	1	1:2
1	0	1:4
1	1	1:8

- **T1OSCEN:** Bit de habilitación del oscilador del Temporizador 1
  - \* T1OSCEN='0': Oscilador del Temporizador 1 desactivado
  - \* T1OSCEN='1': Oscilador del Temporizador 1 activado
- **T1SYNC:** Bit de sincronización de la señal de reloj (solo aplica cuando TMR1CS='1')
  - \* T1SYNC='0': Se sincroniza la entrada de reloj externa RC0/T1OSO/T13CKI
  - \* T1SYNC='1': No se sincroniza la entrada de reloj externa RC0/T1OSO/T13CKI

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 1

---

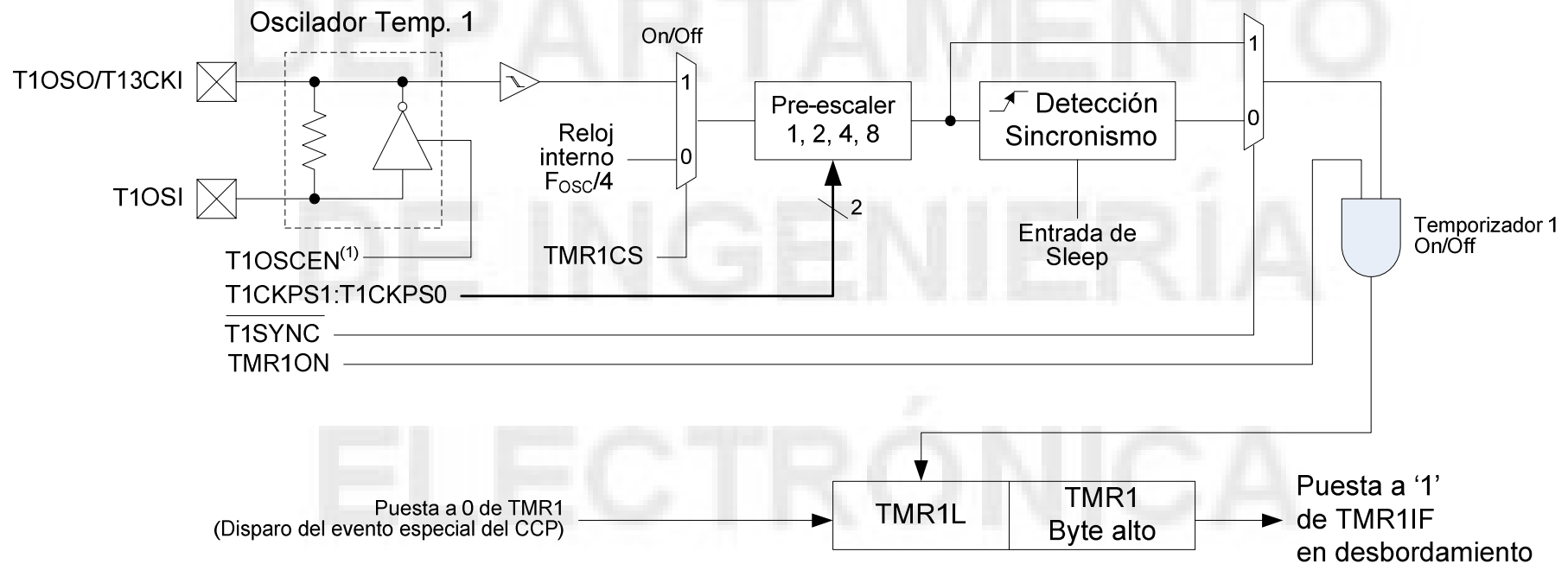
#### REGISTRO T1CON (cont)

- **TMR1CS**: Bits de selección de la fuente de incremento del Temporizador 1
  - \* TMR1CS='0': Pre-escalar del Temp. 1 se incrementa con cada ciclo de instrucción  $F_{osc}/4$
  - \* TMR1CS='1': Pre-escalar del Temp. 1 se incrementa con cada flanco de subida de la línea RC0/T1OSO/T13CKI
- **TMR1ON**: Bit de puesta en marcha del Temporizador 1

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / TEMPORIZADOR 1

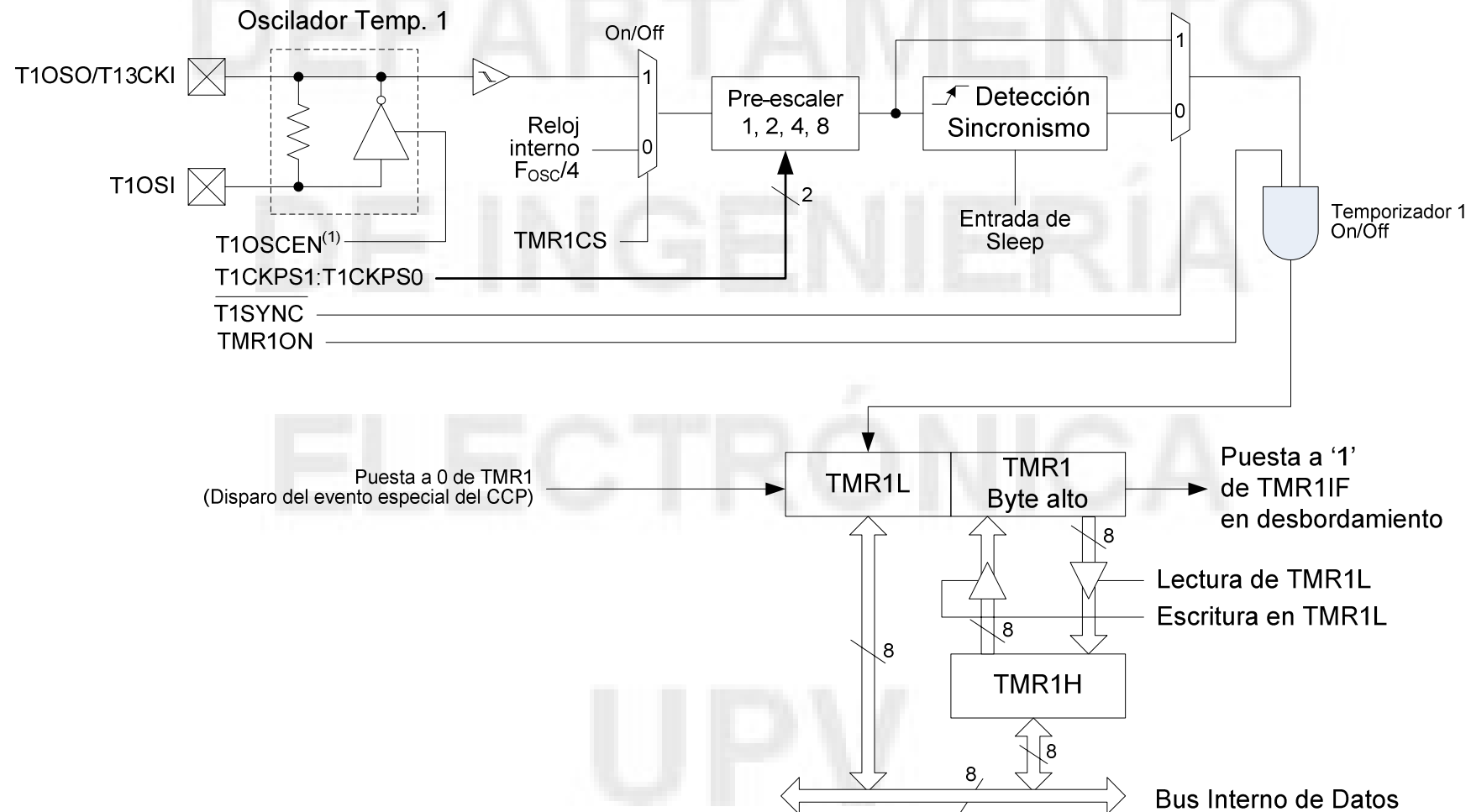
##### Diagrama de bloques (TMR1L y TMR1H registros de L/E independiente):



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / TEMPORIZADOR 1

##### Diagrama de bloques (modo lectura/escritura 16 bits):





## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 1

---

#### Acceso al registro TMR1:

Existen dos modos de acceder al registro TMR1:

- RD16 (T1CON)='0': se accede a TMR1L y TMR1H como dos registros independientes.
- RD16 (T1CON)='1': la parte alta de TMR1 no es accesible directamente. Se accede a ella a través de TMR1H que funciona como un búfer:
  - Cuando se lee el valor de TMR1L, el valor de la parte alta de TMR1 pasa al registro TMR1H. Para leer el valor de TMR1 primero leemos TMR1L y luego leemos TMR1H.
  - Cuando se escribe en TMR1L, el valor de TMR1H pasa a la parte alta de TMR1. Para escribir en TMR1 primero escribimos la parte alta en TMR1H y luego escribimos la parte baja en TMR1L.

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 1

#### Modos de funcionamiento:

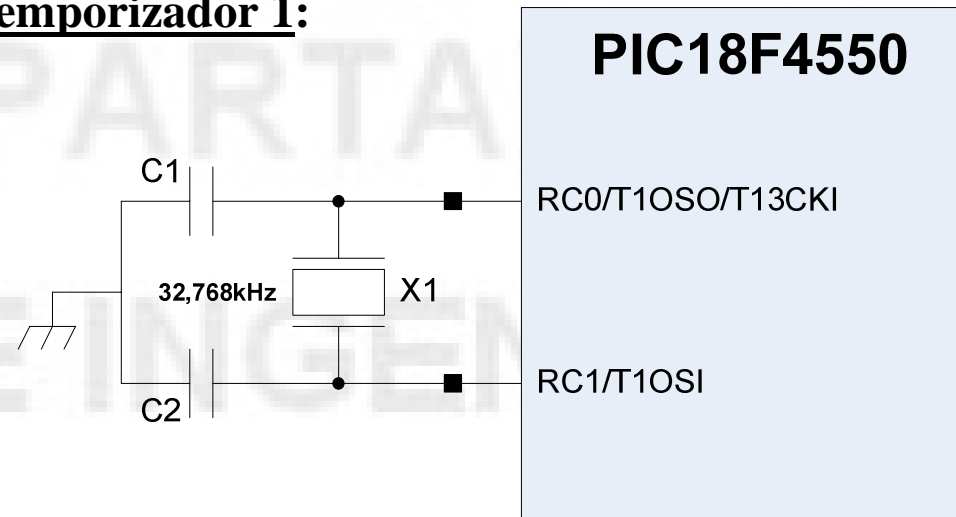
- Temporizador de 16 bits (oscilador principal) TMR1CS='0': el pre-escalar del Temporizador 1 se incrementa en cada ciclo de instrucción ( $F_{OSC}/4$ ) mientras el bit TMR1ON esté a '1'.
- Temporizador de 16 bits (oscilador propio) TMR1CS='1'/T1OSCEN='1': El pre-escalar del Temporizador 1 se incrementa en cada pulso de la señal de reloj del oscilador del Temporizador 1 ( $F_{OSCT1}$ ). La señal de reloj del oscilador del Temporizador 1 puede sincronizarse con el oscilador principal poniendo el bit T1SYNC a '0'.
- Contador síncrono TMR1CS='1'/TMR1OSCEN='0'/T1SYNC='0': el pre-escalar del Temporizador 1 se incrementa de forma síncrona con cada flanco de subida de la línea T13CKI (RC0).
- Contador asíncrono TMR1CS='1'/TMR1OSCEN='0'/T1SYNC='1': el pre-escalar del Temporizador 1 se incrementa de forma asíncrona con cada flanco de subida de la línea T13CKI (RC0).

Para los 4 modos de funcionamiento existe la posibilidad de configurar el pre-escalar mediante los bits T1CKPS1 y T1CKPS0 con 4 posibles ratios (1:1, 1:2, 1:4 y 1:8).

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / TEMPORIZADOR 1

##### Oscilador propio del Temporizador 1:



- El Temporizador 1 dispone de un oscilador propio que permite:
  - Generar la señal de reloj del Temporizador 1
  - Generar la señal de reloj principal en algunos modos de bajo consumo
- Está diseñado para trabajar a una frecuencia de 32kHz.
- Dispone de un modo de bajo consumo, aunque presenta el inconveniente de que cuando el oscilador trabaja en este modo se vuelve más sensible a las interferencias.
- A partir de este oscilador y utilizando la interrupción del Temporizador 1 se puede implementar un reloj en tiempo real de forma muy sencilla.

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 1

---

#### Interrupción del Temporizador 1:

- Flag de interrupción: bit TMR1IF (PIR1)
- Bit de habilitación: bit TMR1IE (PIE1)
- Prioridad: bit TMR1IP (IPR1): '0' -> pr. baja / '1' -> pr. alta

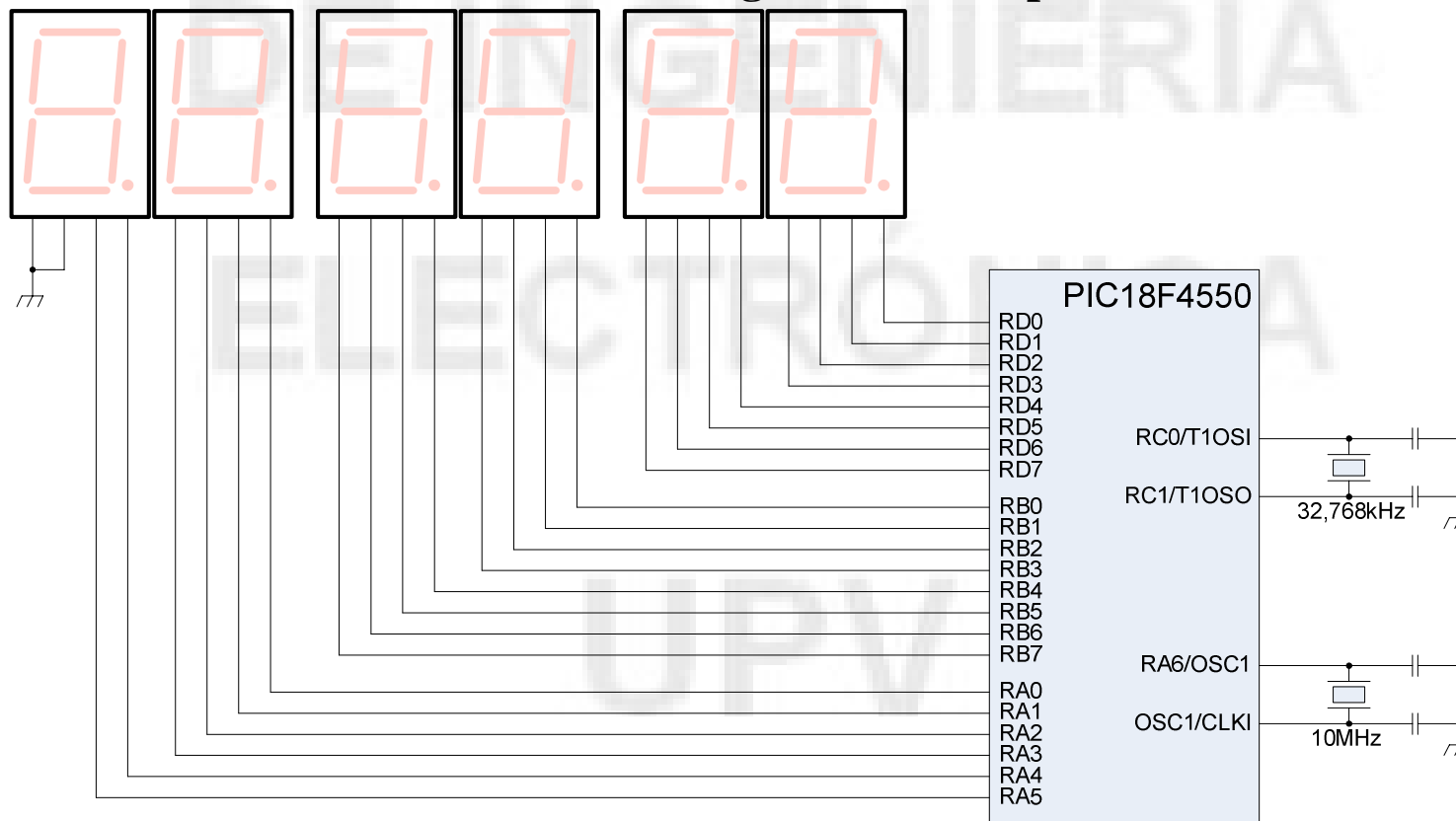
Si se produce el desbordamiento del Temporizador 1 se pone a '1' el flag TMR1IF. Si el bit de habilitación TMR1IE está a '1' y las interrupciones de periféricos están habilitadas a nivel global se genera una interrupción y el uC pasa a ejecutar el código situado a partir de la posición 0008H o 0018H (según el nivel de prioridad establecido).

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / TEMPORIZADOR 1

#### EJERCICIO:

Escribir en código de un programa para el uC PIC18F4550 que genere un reloj en tiempo real (RTC) a partir de la interrupción del Temporizador 1 trabajando con un oscilador propio de  $F_{OSC1}=32\text{kHz}$ . Los valores del RTC se mostrarán en display de 7 segmentos (con decodificadores BCD-7 segmentos incorporados).



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 1

#### EJERCICIO (cont.):

; Bloque de declaraciones y vectorizaciones

#INCLUDE <p18F4520.inc> ; Declaración de la librería de Reg. de Func. Especial

; Declaración de variables

var\_hor EQU 0x00 ; Variable de horas

var\_min EQU 0x01 ; Variable de minutos

var\_sec EQU 0x02 ; Variable de segundos

var\_BCD EQU 0x03 ; Variable que almacena en valor en BCD después de llamar a Bin\_BCD

var\_int EQU 0x04 ; Variable intermedia para la subrutina Bin\_BCD

var\_flg EQU 0x05 ; Variable de flags

; Declaración de bits

cambio EQU .0 ; Flag que indica si ha habido cambio de segundos

; Vectortizaciones

ORG 0x00 ; Vectorización del Reset

GOTO Inicio

ORG 0x08 ; Vectorización de interrupciones de alta prioridad

GOTO Int\_T1



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 1

#### EJERCICIO (cont.):

; PROGRAMA PRINCIPAL  
; Bloque de inicializaciones  
Inicio:

MOVLW 0x0F	; Se desactiva la función de entrada analógica
MOVWF ADCON1	; para todas las líneas
MOVLW 07H	; Se desactiva la función de entrada de comparador
MOVWF CMCON	; para las líneas RA0, RA1, RA2 y RA3
MOVLW 0xC0	; Se configuran RA5..RA0
MOVWF TRISA	; como salidas (horas)
CLRF PORTA	; Se ponen las líneas RA5..RA0 a '0'
CLRF TRISB	; Se configuran RB7..RB0 como salidas (minutos)
CLRF PORTB	; Se ponen las líneas RB7..RA0 a '0'
CLRF TRISD	; Se configuran RD7..RD0 como salidas (segundos)
CLRF PORTD	; Se ponen las líneas RD7..RD0 a '0'
MOVLW 80h	; Valor de inicialización de TMR1 para una temporización de 1 seg.:
MOVWF TMR1H	; $TMR1=65536-1/TOSC1=65536-32768=32768=0x1000$
CLRF TMR1L	;
MOVLW b'00001111'	; Se configura el Temporizador para trabajar con el oscilador propio,
MOVWF T1CON	; en modo asíncrono, con pre-escalar 1:1. Se pone en marcha el Temp. 1
CLRF var_sec	; Se pone a 0 la cuenta de segundos
CLRF var_min	; Se pone a 0 la cuenta de minutos
CLRF var_hor	; Se pone a 0 la cuenta de horas
BCF var_flg,cambio	; Se pone a '0' el flag de cambio de segundos
BSF PIE1, TMR1IE	; Se habilita la interrupción del Temporizador 1
BSF INTCON,PEIE	; Se habilitan a nivel global las interrupciones de periféricos
BSF INTCON,GIE	; Se habilitan a nivel global las interrupciones

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 1

#### EJERCICIO (cont.):

; Bucle principal

Bucle:

```
BTFSS var_flg,cambio ; Se comprueba si se ha producido algún cambio de segundos
GOTO Sigue_Bucle     ; Si no es así se continua con el bucle principal
BCF var_flg,cambio   ; Si se ha producido cambio de segundo ponemos a '0' el flag de cambio
MOVF var_sec,W        ; Convertimos los segundos
CALL Bin_BCD          ;                a BCD
MOVFF var_BCD,PORTD   ; Actualizamos los segundos en el display
MOVF var_min,W        ; Convertimos los minutos
CALL Bin_BCD          ;                a BCD
MOVFF var_BCD,PORTB   ; Actualizamos los minutos en el display
MOVF var_hor,W        ; Convertimos las horas
CALL Bin_BCD          ;                a BCD
MOVFF var_BCD,PORTA   ; Actualizamos las horas en el display
```

Sigue\_Bucle:

GOTO Bucle





### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / TEMPORIZADOR 1

#### EJERCICIO (cont.):

; Rutina de tratamiento de interrupción

Int\_T1:

BTFSS PIR1,TMR1IF	; Se comprueba si la interrupción ha sido por desbordamiento del Temp. 1
GOTO Fin_IT1	; Si no es así se sale de la interrupción
BCF PIR1,TMR1IF	; Se pone a '0' el flag de interrupción del Temp. 1
BSF TMR1H, 7	; Valor de recarga de TMR1: 32768=0x1000
BSF var_flg,cambio	; Se pone a '1' el flag de cambio de segundos
INCF var_sec, F	; Se incrementan los segundos
MOVLW d'59'	; Se comprueba si han
CPFSGT var_sec	; pasado 60 segundos
GOTO Fin_IT1	; Si no es así salimos de la rutina de interrupción
CLRF var_sec	; Si han pasado 60 segundos se ponen a 0 los segundos
INCF var_min,F	; Se incrementan los minutos
MOVLW d'59'	; Se comprueba si han
CPFSGT var_min	; pasado 60 minutos
GOTO Fin_IT1	; Si no es así salimos de la rutina de interrupción
CLRF var_min	; Si han pasado 60 minutos se ponen a 0 los minutos
INCF var_hor, F	; Se incrementan las horas
MOVLW d'23'	; Se comprueba si han
CPFSGT var_hor	; pasado 24 horas
GOTO Fin_IT1	; Si no es así salimos de la rutina de interrupción
CLRF var_hor	; Si han pasado 24 horas ponemos a '0' las horas

Fin\_IT1:

RETfie

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 1

#### EJERCICIO (cont.):

; Subrutina de conversión de binario a BCD

Bin\_BCD:

CLRF var\_BCD ; Ponemos var\_BCD a 0

Rep\_BCD

MOVWF var\_int ; Le restamos 10

MOVLW .10 ; al valor

SUBWF var\_int,W ; del acumulador

BTFSC STATUS,N ; Comprobamos si el resultado es negativo

GOTO BCD1 ; Si el resultado es negativo pasado al procesado final

INCF var\_BCD,F ; Si el resultado no es negativo, incrementamos var\_BCD

GOTO Rep\_BCD ; y volvemos a repetir el proceso

BCD1:

ADDLW .10 ; Si la resta anterior ha dado negativo deshacemos la resta anterior ( $W=W+10$ )

SWAPF var\_BCD ; Colocamos la parte alta de var\_BCD en su lugar

IORWF var\_BCD,F ; Se hace un OR entre el acumulador

RETURN

END



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

---

#### **CONVERTIDOR ANALÓGICO-DIGITAL:**

##### **Características fundamentales:**

- 10 bits de resolución
- 13 canales multiplexados
- Señal de reloj de conversión configurable
- Tiempo de adquisición programable (0 a  $20T_{AD}$ )
- Posibilidad de establecer el rango de tensiones de conversión mediante tensiones de referencia externas

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

##### REGISTRO ADCON0

	-0	-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
ADCON0	-	-	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON

- **CHS3..CHS0**: Bits selección del canal de conversión A/D (13 canales)
- **GO/DONE**: Bit de inicio y de monitorización del estado de la conversión A/D:
  - \* GO/DONE='0': Proceso de conversión parado
  - \* GO/DONE='1': Proceso de conversión en marcha
- **ADON**: Bit de habilitación del convertidor A/D
  - \* ADON='0': Convertidor A/D desactivado
  - \* ADON='1': Convertidor A/D activado

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

##### REGISTRO ADCON1

	-0	-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
ADCON1	-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0

- **VCFG1**: Bit de configuración de la tensión de referencia  $V_{REF-}$ :
  - \* VCFG1='0':  $V_{REF-}$  se conecta a  $V_{SS}$
  - \* VCFG1='1':  $V_{REF-}$  se conecta a la línea física RA2
- **VCFG0**: Bit de configuración de la tensión de referencia  $V_{REF+}$ :
  - \* VCFG0='0':  $V_{REF+}$  se conecta a  $V_{DD}$
  - \* VCFG0='1':  $V_{REF+}$  se conecta a la línea física RA3
- **PCFG3..PCFG0**: Bits configuración de los puertos de conversión A/D. Mediante estos bits se establecen las líneas físicas (RA5..RA0, RB4..RB0, RE1 y RE0) que van a trabajar como entradas del convertidor A/D.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

##### REGISTRO ADCON2

	L/E-0	-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
ADCON2	ADFM	-	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0

- **ADFM**: Bit de configuración del tipo de almacenamiento del resultado de la conversión en los registros ADRESH y ADRESL:
  - \* ADFM='0': El resultado de la conversión se almacena con justificación a izquierdas
  - \* ADFM='1': El resultado de la conversión se almacena con justificación a derechas
- **ACQT2..ACQT0** : Bits de configuración del tiempo de adquisición
- **ADCS2..ADCS0**: Bits selección de la señal de reloj del convertidor A/D



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

#### Selección del canal de conversión:

Para que uno de los 13 canales pueda ser seleccionado, previamente debe haber sido configurado como entrada analógica mediante los bits PCFG3..PCFG0 del registro ADCON1 (A: analógico / D: digital).

PCFG3..PCFG0	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
0000	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

##### Selección del canal de conversión (cont.):

Una vez configurado como línea de entrada analógica, un canal puede ser seleccionado mediante los bits CHS3..CHS0 del registro ADCON0.

CHS3	CHS2	CHS1	CHS0	CANAL SELECCIONADO
0	0	0	0	CANAL AN0 (RA0)
0	0	0	1	CANAL AN1 (RA1)
0	0	1	0	CANAL AN2 (RA2)
0	0	1	1	CANAL AN3 (RA3)
0	1	0	0	CANAL AN4 (RA5)
0	1	0	1	CANAL AN5 (RE0)
0	1	1	0	CANAL AN6 (RE1)
0	1	1	1	CANAL AN7 (RE2)
1	0	0	0	CANAL AN8 (RB2)
1	0	0	1	CANAL AN9 (RB3)
1	0	1	0	CANAL AN10 (RB1)
1	0	1	1	CANAL AN11 (RB4)
1	1	0	0	CANAL AN12 (RB0)
1	1	0	1	No implementado
1	1	1	0	No implementado
1	1	1	1	No implementado

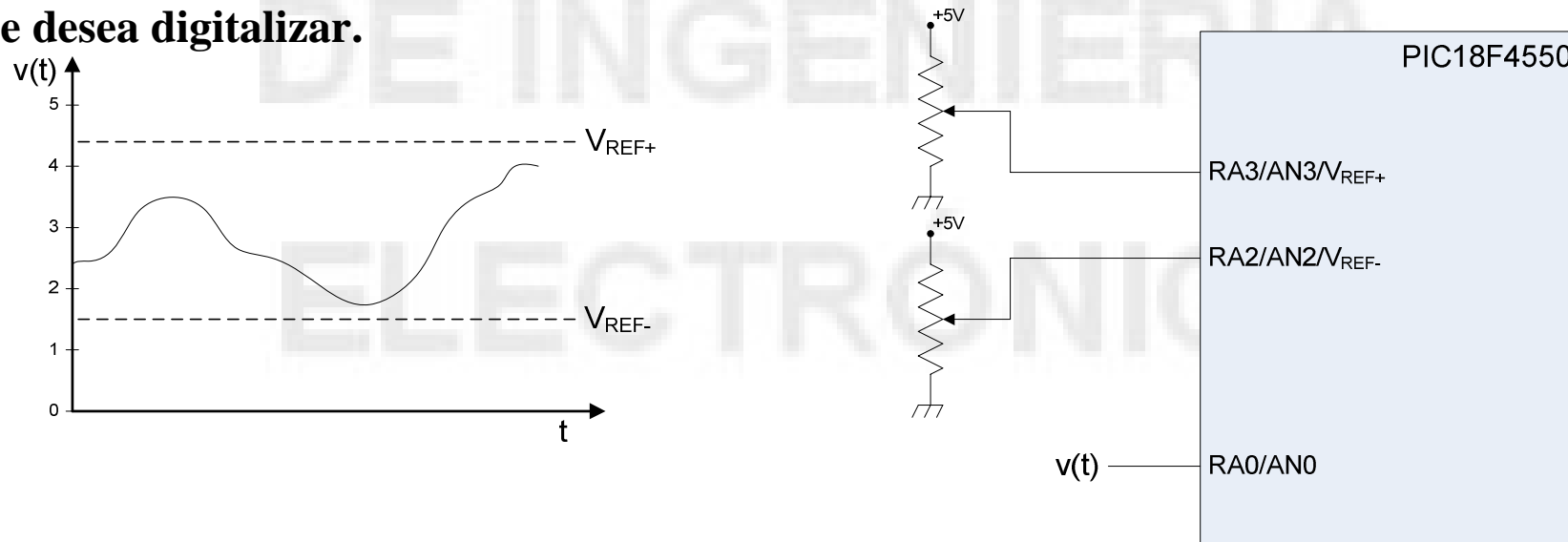


### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

##### Rango tensiones de conversión:

Por defecto el rango de tensiones de conversión del convertidor A/D del PIC18F4550 es de 0V a 5V. Sin embargo, en ocasiones puede resultar interesante modificar este rango para aumentar la resolución de la conversión acercando las tensiones de referencia máxima y mínima  $V_{REF+}$  y  $V_{REF-}$  a los límites de variación de la señal que se desea digitalizar.



Esto se puede conseguir configurando las líneas RA2/AN2/V<sub>REF-</sub> y RA3/AN3/V<sub>REF+</sub> como tensiones de referencia del convertidor A/D (poniendo a '1' los bits VCFG1 y VCFG0 del registro ADCON1). De esta forma el rango de tensiones de conversión vendrá determinado por las tensiones que se conecten en dichas líneas.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

##### Señal de reloj de conversión:

Se define  $T_{AD}$  como el tiempo de conversión de 1 bit. Una operación completa de conversión requiere un total de 11  $T_{AD}$  para 10 bits.

La señal de reloj que genera las temporizaciones  $T_{AD}$  puede ser establecida mediante los bits ADCS2..ADCS0 del registro ADCON2. Existen dos fuentes para dicha señal de reloj:

- El oscilador principal
- Una red RC interna que incorpora el propio convertidor A/D. Esta red puede utilizarse cuando se deseen realizar conversiones en modos de bajo consumo. Esta red RC permite que se puedan llevar a cabo conversiones con el oscilador principal desactivado.

ADCS2	ADCS1	ADCS0	SEÑAL DE RELOJ DE CONVERSION
0	0	0	$F_{osc}/2$
0	0	1	$F_{osc}/8$
0	1	0	$F_{osc}/32$
0	1	1	$F_{RC}$ (oscilador RC interno)
1	0	0	$F_{osc}/4$
1	0	1	$F_{osc}/16$
1	1	0	$F_{osc}/64$
1	1	1	$F_{RC}$ (oscilador RC interno)

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

---

#### Señal de reloj de conversión (cont.):

El valor de  $T_{AD}$  debe ser lo menor posible pero siempre superior al  $T_{AD}$  mínimo indicado en las hojas de datos de PIC18F4550: 0,7us.

#### **Ejemplo:**

Se elige el oscilador principal como fuente de reloj y  $F_{OSC}=10\text{MHz}$ :

- Si configuramos  $ADCS2..ADCS0=000 \Rightarrow T_{AD}=2*(1/10\cdot106)=0,2\mu\text{s} \Rightarrow$  demasiado pequeño.
- Si configuramos  $ADCS2..ADCS0=100 \Rightarrow T_{AD}=4*(1/10\cdot106)=0,4\mu\text{s} \Rightarrow$  demasiado pequeño
- Si configuramos  $ADCS2..ADCS0=001 \Rightarrow T_{AD}=8*(1/10\cdot106)=0,7\mu\text{s} \Rightarrow$  OK

Cuando se elige el oscilador RC interno el  $T_{AD}$  viene prefijado internamente y siempre cumple con la condición del valor mínimo.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

---

##### Establecimiento del tiempo de adquisición (S&H):

La circuitería interna del convertidor A/D incorpora un condensador de muestreo. Antes de realizar una conversión debemos asegurarnos de que dicho condensador ha sido totalmente cargado a la tensión del canal seleccionado.

Cuando realizamos un cambio en la selección de canal debemos esperar un tiempo que dependerá de la impedancia de entrada del convertidor A/D y de la impedancia de salida del circuito sobre el que se está haciendo la conversión.

Existen dos opciones para generar este retardo antes de comenzar la conversión:

- **Por programa:** se implementa un retardo software entre la selección del nuevo canal y el inicio de la conversión.
- **Estableciendo un tiempo de adquisición automático:** se programa un tiempo de adquisición que se establecerá de forma automática entre la orden de inicio de conversión y el muestreo de la señal para iniciar la conversión. Dicho tiempo puede ser programado mediante los bits ACQT2..ACQT0 del registro ADCON2. Este tiempo puede tener unos valores que oscilan entre  $2 \cdot T_{AD}$  y  $20 \cdot T_{AD}$ .

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

##### Establecimiento del tiempo de adquisición (cont.):

ACQT2	ACQT1	ACQT0	TIEMPO DE ADQUISICION
0	0	0	$0 \cdot T_{AD}$
0	0	1	$2 \cdot T_{AD}$
0	1	0	$4 \cdot T_{AD}$
0	1	1	$6 \cdot T_{AD}$
1	0	0	$8 \cdot T_{AD}$
1	0	1	$12 \cdot T_{AD}$
1	1	0	$16 \cdot T_{AD}$
1	1	1	$20 \cdot T_{AD}$

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

##### Almacenamiento del resultado de la conversión A/D:

Una vez acabada la conversión A/D el resultado de la misma (un valor de 10 bits) queda almacenado en los registros ADRESH y ADRESL.

Existen dos posibles formas de almacenar el resultado en estos registros dependiendo del valor del bit ADFM del registro ADCON2:

ADFM='0' RESULTADO JUSTIFICADO A IZQUIERDAS	ADFM='1' RESULTADO JUSTIFICADO A DERECHAS																						
<div>ADRESH</div> <table><tr><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td></tr></table> <div>MSB<span>LSB</span></div> <div>ADRESL</div> <table><tr><td>b1</td><td>b0</td><td></td></tr></table> <div>MSB<span>LSB</span></div>	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		<div>ADRESH</div> <table><tr><td></td><td>b9</td><td>b8</td></tr></table> <div>MSB<span>LSB</span></div> <div>ADRESL</div> <table><tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr></table> <div>MSB<span>LSB</span></div>		b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
b9	b8	b7	b6	b5	b4	b3	b2																
b1	b0																						
	b9	b8																					
b7	b6	b5	b4	b3	b2	b1	b0																

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

---

#### Proceso completo de conversión A/D (sin utilizar interrupción):

##### **1º PASO: Configuración del convertidor A/D:**

- Configuración como canales A/D de las líneas que vayan a ser utilizadas (bits PCFG3..PCFG0 del registro ADCON1)
- Configuración de las tensiones de referencia  $V_{REF+}$  y  $V_{REF-}$  (bits VCFG0 y VCFG1 del registro ADCON1)
- Configuración del reloj de conversión TAD (bits ADCS2..ADCS0 del registro ADCON2)
- Configuración del tiempo de adquisición (bits ACQT2..ACQT0 del registro ADCON2)
- Configuración del modo de almacenamiento de la conversión (bit ADFM del registro ADCON2)
- Activación del conversor (bit ADON del registro ADCON0)

##### **2º PASO: Selección del canal (bits CHS3..CHS0 del registro ADCON0)**

##### **3º PASO: Retardo de espera del tiempo de adquisición (solo en caso de no hacer uso del tiempo de adquisición automático)**

##### **4º PASO: Inicio de la conversión poniendo a '1' el bit GO/DONE del registro ADCON0**

##### **5º PASO: Bucle de espera del final de conversión (comprobación del bit GO/DONE hasta que se ponga a '0')**

##### **6º PASO: Lectura del resultado de la conversión de los registros ADRESH y ADRESL**

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

---

#### Interrupción del Convertidor analógico-digital:

- Flag de interrupción: bit ADIF (PIR1)
- Bit de habilitación: bit ADIE (PIE1)
- Prioridad: bit ADIP (IPR1): '0' -> pr. baja / '1' -> pr. alta

Cuando finaliza una conversión A/D se pone a '1' el flag ADIF. Si el bit de habilitación ADIE está a '1' y las interrupciones de periféricos están habilitadas a nivel global se genera una interrupción y el uC pasa a ejecutar el código situado a partir de la posición 0008H o 0018H (según el nivel de prioridad establecido).



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

---

#### Proceso completo de conversión A/D (utilizando interrupción):

##### **1º PASO: Configuración del convertidor A/D:**

- Configuración como canales A/D de las líneas que vayan a ser utilizadas (bits PCFG3..PCFG0 del registro ADCON1)
- Configuración de las tensiones de referencia  $V_{REF+}$  y  $V_{REF-}$  (bits VCFG0 y VCFG1 del registro ADCON1)
- Configuración del reloj de conversión TAD (bits ADCS2..ADCS0 del registro ADCON2)
- Configuración del tiempo de adquisición (bits ACQT2..ACQT0 del registro ADCON2)
- Configuración del modo de almacenamiento de la conversión (bit ADFM del registro ADCON2)
- Activación del conversor (bit ADON del registro ADCON0)

##### **2º PASO: Habilitación de la interrupción del convertidor A/D:**

- Habilitación de la interrupción del convertidor A/D (bit ADIE del registro PIE1)
- Habilitación de las interrupciones de periféricos y de las interrupciones a nivel general (bits PEIE y GIE del registro INTCON)

##### **3º PASO: Selección del canal (bits CHS3..CHS0 del registro ADCON0)**

##### **4º PASO: Retardo de espera del tiempo de adquisición (solo en caso de no hacer uso del tiempo de adquisición automático)**

##### **4º PASO: Inicio de la conversión poniendo a '1' el bit GO/DONE del registro ADCON0**

##### **5º PASO: El uC sigue ejecutando código hasta que se produzca la interrupción**

##### **6º PASO: Cuando se produce la interrupción: puesta a '0' del bit ADIF del registro PIR1 y lectura del resultado de la conversión de los registros ADRESH y ADRESL**

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

---

#### EJERCICIO:

Escribir en código de un programa para el uC PIC18F4550 que muestree una señal analógica con una cadencia de 256 muestras por segundo. Los datos de las muestras se almacenarán entre las posiciones 200H y 2FFH de la memoria RAM interna (las 256 muestras correspondientes a un determinado segundo sobrescribirán los datos correspondientes al segundo anterior). La aplicación deberá calcular la media de los datos muestreados y almacenar dicho dato en la variable Var\_Med.

NOTA:  $F_{OSC}=4\text{MHz}$

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

#### EJERCICIO (cont.):

; Bloque de declaraciones y vectorizaciones

#INCLUDE <p18F4550.inc> ; Declaración de la librería de Reg. de Func. Especial

; Declaración de variables

Ind\_Buf EQU 0x00 ; Índice del búfer de comunicación

Total\_L EQU 0x01 ; Parte baja de la suma total

Total\_H EQU 0x02 ; Parte alta de la suma total

Val\_Med EQU 0x03 ; Valor medio de las muestras

Bufer EQU 0x200 ; Bufer de almacenamiento de datos

; Vectortizaciones

ORG 0x0000 ; Vectorización del Reset

GOTO Main

ORG 0x0008 ; Vectorización de las interrupciones de alta prioridad

GOTO HighInt



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

#### EJERCICIO (cont.):

; PROGRAMA PRINCIPAL

; Bloque de inicializaciones

Main:

; Inicialización del canal de conversión A/D

MOVLW 0x01	; Se activa el conversor A/D
MOVWF ADCON0	; y se selecciona el canal AN0
MOVLW 0x0E	; Se habilita RA0 como entrada analógica
MOVWF ADCON1	; y el resto como entradas digitales
MOVLW 0x25	; Se configura el formato de resultado justificado a izquierdas
MOVWF ADCON2	; el tpo de adquisición a 8*TAD y TAD = 16*TOSC

; Inicialización del Temporizador 0

MOVLW b'10001000'	; Se configura el Temporizador 0 en modo temporizador de 16 bits
MOVWF T0CON	; con el prescalar deshabilitado
MOVLW 0xF0	; Se inicializa TMR0 para una temporización de 1/256=3906
MOVWF TMR0H	; TMR0=65536-3906E-06*(Fosc/4)=3906=F0BE
MOVLW 0xBE	;
MOVWF TMR0L	;

; Habilitación de interrupciones

BCF RCON,IPEN	; Se deshabilitan la prioridad en las interrupciones
BSF INTCON,TMR0IE	;
BSF INTCON,GIE	; Se habilitan a nivel global las interrupciones

; Inicialización de variables

CLRF Ind_Buf	; se in inicializa el índice del búfer a 0
LFSR FSR0,Bufer	; Inicializamos FSR0 al inicio del búfer

; Bucle Principal

Bucle:

GOTO Bucle



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

#### EJERCICIO (cont.):

; Rutina de tratamiento de interrupciones de alta prioridad

HighInt:

```
BTFSS INTCON,TMR0IF ; Se comprueba si la interrupción ha sido por desbordamiento del Temp. 0
GOTO Fin_IntH        ; Si no es así se va la final de la rutina de interrupción
BCF INTCON,TMR0IF    ; Se pone a '0' el flag de int. del Temp. 0
MOVLW 0xF0           ; Se inicializa TMR0 para una temporización de 1/256=3906
MOVWF TMR0H          ; TMR0=3906E-06*(Fosc/4)=3906=F0BE
MOVLW 0xBE           ;
MOVWF TMR0L          ;
BSF ADCON0,GO_DONE ; Se inicia la conversión A/D
```

Espera:

```
BTFSC ADCON0,GO_DONE; Se comprueba si la conversión ha llegado a su fin
GOTO Espera        ; Si no ha llegado a su fin, se espera
MOVFF ADRESH,POSTINC0; Se almacena el valor leído en el bufer
INCF Ind_Buf       ; Se incrementa el índice del bufer
MOVLW .0           ; Se comprueba si el índice del bufer
CPFSEQ Ind_Buf     ; ha llegado a 0 (se han almacenado 256 muestras)
GOTO Fin_IntH      ; Si no es así, se va al final
CLRF Ind_Buf       ; Si se han almacenado las 256 muestras se para al cálculo del valor medio
LFSR FSR0,Bufer    ; Se inicializan el índice del bufer y FSR0 al inicio del bufer
CLRF Total_L       ; Se inicializan la parte alta
CLRF Total_H       ; y baja del valor medio
```

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CONVERTIDOR ANALÓGICO-DIGITAL

#### EJERCICIO (cont.):

Calcula:

```
MOVF POSTINC0,W ; Se pasa al acumulador el valor apuntado por el FSR
ADDWF Total_L,F ; Se suma el valor leído a la parte baja
BTFSC STATUS,C ;
INCF Total_H,F ;
INCF Ind_Buf ; Se incrementa el índice del bufer
MOVLW .0 ; Se compara W con el índice
CPFSEQ Ind_Buf ; del bufer (para ver si se han leído 256 datos)
GOTO Calcula ; Si no es así, se repite el bucle
LFSR FSR0,Bufer ; Si se ha sumado los 256 datos del bufer se inic. FSR0 al inicio del bufer
MOVFF Total_H,Val_Med ; Se pasa la parte alta del total (total dividido por 256) a Val_Med
```

Fin\_IntH:

```
RETFIE FAST
END
```



**CANAL DE COMUNICACIÓN SERIE EUSART:**

**Características fundamentales:**

- **Modos de trabajo:**
  - **Modo asíncrono de 8 bits**
  - **Modo asíncrono de 9 bits**
  - **Modo síncrono Maestro**
  - **Modo síncrono Esclavo**
- **Auto-activación por detección de dato recibido**
- **Detección automática de velocidad de comunicación (baudrate)**
- **Transmisión y detección de carácter de BREAK (bus LIN)**

**En este tema solo se tratará el modo asíncrono básico (8 y 9 bits).**

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

##### REGISTRO TXSTA

	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L-1	L/E-0
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D

- **CSRC**: Bit selección de la fuente de señal de reloj para el modo síncrono:
  - \* CSRC='0': Modo Esclavo (señal de reloj externa)
  - \* CSRC='1': Modo Maestro (señal de reloj generada internamente)
- **TX9**: Bit de habilitación del modo de 9 bits en transmisión:
  - \* TX9='0': Se habilita el modo de 8 bits en transmisión
  - \* TX9='1': Se habilita el modo de 9 bits en transmisión
- **TXEN**: Bit de habilitación de la transmisión:
  - \* TXEN='0': Transmisión deshabilitada
  - \* TXEN='1': Transmisión habilitada
- **SYNC**: Bit de selección del modo de trabajo del EUSART
  - \* SYNC='0': Selección del modo asíncrono
  - \* SYNC='1': Selección del modo síncrono
- **SENDB**: Bit de envío del carácter de BREAK en modo asíncrono
  - \* SENDB='0': Envío del carácter de transmisión BREAK completado
  - \* SENDB='1': Se enviará un carácter BREAK en la próxima transmisión (se pone a '0' por hardware cuando finaliza el envío)



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

---

#### REGISTRO TXSTA (cont)

- **BRGH**: Bits de selección del modo de alta velocidad de comunicación (baudrate) en modo asíncrono:
  - \* BRGH='0': Se selecciona baja velocidad de comunicación
  - \* BRGH='1': Se selecciona alta velocidad de comunicación
- **TRMT**: Bit de status del registro de desplazamiento de transmisión:
  - \* TMRT='0': Registro de desplazamiento de transmisión ocupado
  - \* TMRT='1': Registro de desplazamiento de transmisión vacío
- **TX9D**: Valor del 9º Bit en transmisión (puede utilizarse como bit de paridad o para distinguir entre dirección o dato en los buses maestro-esclavo)



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

##### REGISTRO RCSTA

	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L-0	L-0	L-x
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D

- **SPEN**: Bit de activación del puerto serie:
  - \* SPEN='0': Puerto serie desactivado
  - \* SPEN='1': Puerto serie activado (se configuran las líneas RC6/TX/CK y RC7/RX/DT como líneas del puerto serie)
- **RX9**: Bit de habilitación del modo de 9 bits en recepción:
  - \* RX9='0': Se habilita el modo de 8 bits en recepción
  - \* RX9='1': Se habilita el modo de 9 bits en recepción
- **SREN**: Bit de habilitación de la recepción simple en modo síncrono Maestro:
  - \* SREN='0': Se deshabilita la recepción simple
  - \* SREN='1': Se habilita la recepción simple
- **CREN**: Bit de habilitación de la recepción:
  - \* CREN='0': Recepción deshabilitada
  - \* CREN='1': Recepción habilitada
- **ADDEN**: Bit de habilitación de la detección de dirección en el modo asíncrono de 9 bits:
  - \* ADDEN='0': Detección de dirección deshabilitada (todos los bytes recibidos en el registro de desplazamiento de recepción son procesados independientemente del valor del 9º bit recibido)
  - \* ADDEN='1': Detección de dirección habilitada (solo se procesa un byte recibido en el registro de desplazamiento de recepción si el 9º bit está a '1')

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

---

#### REGISTRO RCSTA (cont)

- **FERR**: Bit de error de formato (solo lectura):
  - \* FERR='0': No se ha producido error de formato
  - \* FERR='1': Se ha producido error de formato
- **OERR**: Bit de error de sobrescritura (solo lectura):
  - \* FERR='0': No se ha producido error de sobrescritura
  - \* FERR='1': Se ha producido error de sobrescritura
- **RX9D**: Valor del 9º bit recibido



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

##### REGISTRO BAUDCON

	L/E-0	L-1	-0	L/E-0	L/E-0	-0	L/E-0	L/E-0
BAUDCON	ADBOVF	RCIDL	-	SCKP	BRG16	-	WUE	ABDEN

- **ADBOVF**: Bit de desbordamiento de auto-detección de la velocidad de comunicación:
  - \* ADBOVF='0': No se ha producido desbordamiento durante la auto-detección
  - \* ADBOVF='1': Se ha producido desbordamiento durante la auto-detección (debe ponerse a '0' por software)
- **RCIDL**: Bit de status de la operación de recepción:
  - \* RCIDL='0': Hay una operación de recepción en marcha
  - \* RCIDL='1': No hay ninguna operación de recepción en marcha
- **SCKP**: Bit de selección de polaridad de la señal de reloj en modo síncrono:
  - \* SCKP='0': El dato está disponible en el nivel bajo de la señal de reloj (después del flanco de bajada)
  - \* SCKP='1': El dato está disponible en el nivel alto de la señal de reloj (después del flanco de subida)
- **BRG16**: Bit de habilitación del generador de velocidad de comunicación de 16 bits:
  - \* BRG16='0': Generador de velocidad de comunicación de 8 bits (solo SPBRG)
  - \* BRG16='1': Generador de velocidad de comunicación de 16 bits (SPBRG y SPBRGH)
- **WUE**: Bit de habilitación del modo de auto-activación en modo asíncrono:
  - \* WUE='0': Modo de auto-activación deshabilitado
  - \* WUE='1': Modo de auto-activación habilitado (cuando se detecta un flanco de bajada en la línea RX se pone a '1' el flag RCIF)

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

---

#### REGISTRO BAUDCON (cont)

- **ABDEN**: Bit de habilitación del modo de auto-detección de velocidad de comunicación:
  - \* ABDEN='0': Modo de auto-detección deshabilitado
  - \* ABDEN='1': Modo de auto-detección habilitado en la siguiente recepción

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

---

##### Configuración de las líneas TX y RX para el modo asíncrono:

Las líneas RC6/TX y RC7/RX deben configurarse adecuadamente para que puedan funcionar como las líneas de transmisión y recepción respectivamente:

- Poner a '1' el bit SPEN (RCSTA)
- Poner a '1' el bit 7 del registro TRISC (línea RC7/RX configurada como entrada)
- Poner a '0' el bit 6 del registro TRISC (línea RC6/TX configurada como salida)

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

##### Generación de la velocidad de comunicación para el modo asíncrono:

El uC 18F4550 dispone de un generador de velocidad de comunicación. Se trata de un temporizador cuya frecuencia se controla mediante un registro. El registro puede ser de 8 (SPBRG) o 16 bits ([SPBRGH;SPBRG]). El generador de velocidad de comunicación tiene dos velocidades de funcionamiento que pueden seleccionarse mediante el bit BGRH (TXSTA).

En la siguiente tabla se establece como calcular la velocidad de comunicación para el modo asíncrono en función de los valores de los bits BRGH y BRG16 y de los registros SPBRGH y SPBRG:

BRG16	BRGH	CALCULO DE LA VELOCIDAD DE COMUNICACIÓN
0	0	$\frac{F_{osc}}{64*(SPBRG+1)}$
0	1	$\frac{F_{osc}}{16*(SPBRG+1)}$
1	0	$\frac{F_{osc}}{16*([SPBRGH:SPBRG]+1)}$
1	1	$\frac{F_{osc}}{4*([SPBRGH:SPBRG]+1)}$

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

#### Generación de la velocidad de comunicación para el modo asíncrono (cont.):

**Ejemplo:** Calcular el valor de SPBRGH y SPBRG para la siguiente configuración:  
**Velocidad de comunicación 9600 baudios,  $F_{OSC}=40\text{MHz}$ , BRGH='0' y BRG16='1'.**

$$9600 = \frac{F_{OSC}}{16*([SPBRGH:SPBRG]+1)} \Rightarrow [SPBRGH:SPBRG] = \frac{40*10^6}{(16*9600)} - 1 = 259,4166$$

**Por tanto SPBRGH=0x01 y SPBRG=0x03. De esta forma el valor real de la velocidad de comunicación sería:**

$$\text{Velocidad de comunicación} = \frac{F_{OSC}}{16*([SPBRGH:SPBRG]+1)} = \frac{40*10^6}{16*(259+1)} = 9615,38$$

**La desviación entre el valor deseado (9600) y el valor obtenido se debe al redondeo que se realiza sobre [SPBRGH;SPBRG]. Para minimizar esta desviación es conveniente elegir adecuadamente el valor de BRGH y BRG16 en función de  $F_{OSC}$ . La siguiente tabla muestra distintas posibilidades de cálculo para 9600 baudios.**

BRG16	BRGH	$F_{OSC}$						
		40 MHz	20 MHz	10 MHz	8 MHz	4 MHz	2 MHz	1 MHz
0	0	9615	9766	9766	9615	8929	-	-
0	1	9766	9615	9615	9615	9615	9615	-
1	0	9615	9615	9615	9615	9615	9615	-
1	1	9606	9596	9615	9615	9615	9615	9615

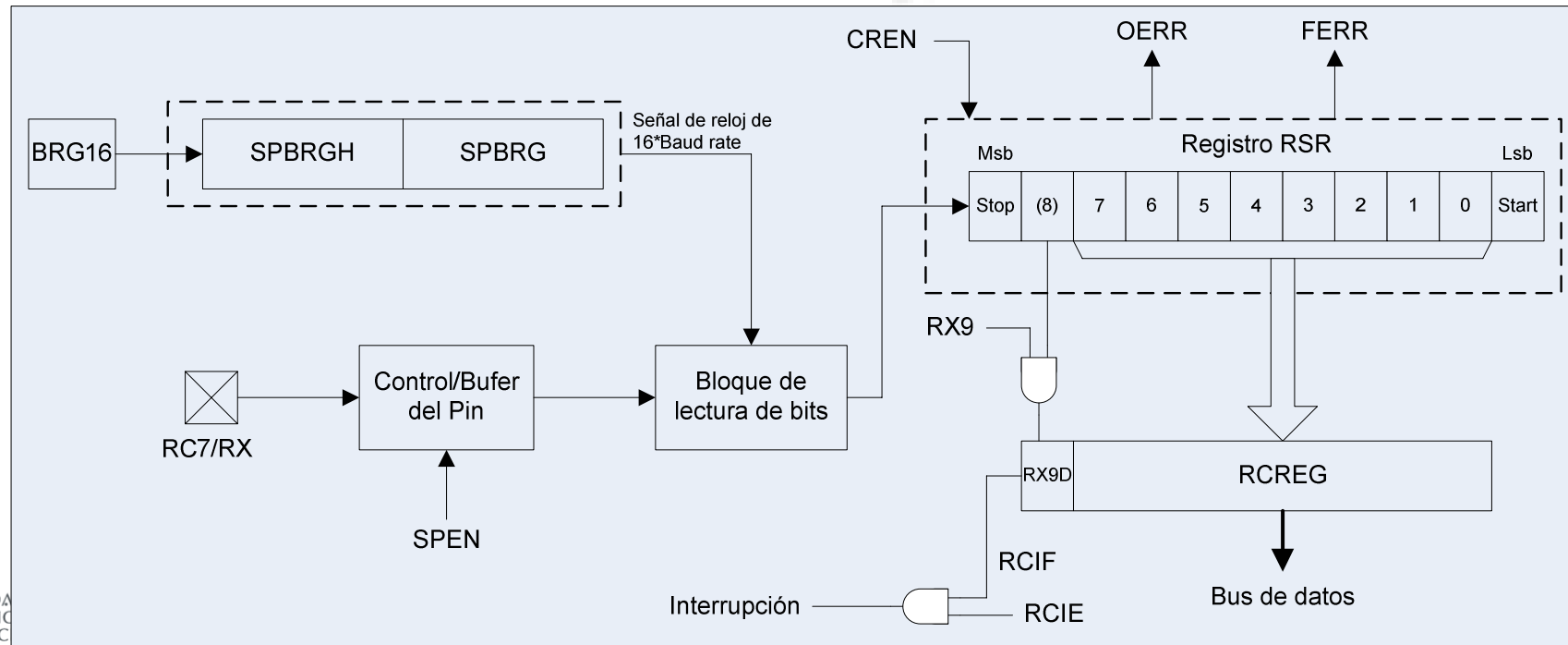


### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

##### Recepción en el modo asíncrono de la USART:

El bloque de recepción de la USART incorpora un registro de desplazamiento serie (RSR). Los datos entran en serie por el pin RC7/RX y son muestreados por el bloque de lectura de bits (que trabaja a una frecuencia de 16 veces el baudrate). Si el bit de habilitación de recepción CREN está a '1', el bloque de lectura de bits identifica los bits recibidos (Start Bit, Bits de datos, 9º bit y Stop bit) y los va pasando en serie al RSR. Una vez se han recibido todos los bits correspondientes a un byte el valor recibido se pasa en paralelo de RSR al registro RCREG. De esta forma el registro RSR queda listo para recibir un nuevo dato.



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

---

##### Recepción en el modo asíncrono de la USART (cont.):

Si durante el proceso de recepción se produce algún error de formato (el valor del Stop bit el '0' en lugar de '1') o de sobrescritura (se recibe un nuevo byte antes de que el registro RSR se haya volcado en el RCREG) se pondrá a '1' el bit correspondiente (FERR/OERR).

Cada vez que se recibe un dato el flag RCIF se pone a '1'. Si el bit de habilitación de la interrupción de recepción del EUSART está a '1' se generará una interrupción. El bit **RCIF es de solo lectura**, se pone a '1' cuando llega un nuevo dato a RCREG y se pone a '0' automáticamente cuando se lee dicho registro.

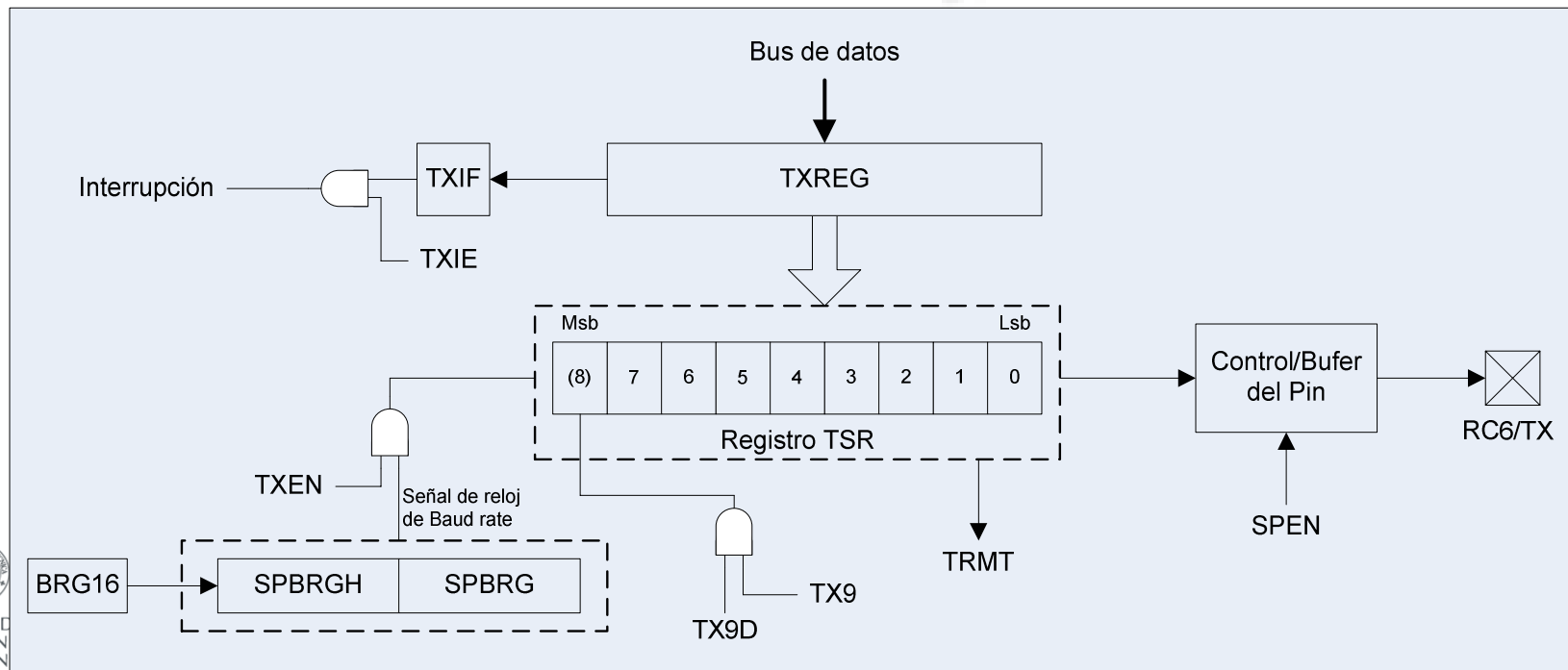
Si el modo de 9 bits ha sido activado (bit RX9 a '1') el valor del 9º bit recibido quedará almacenado en el bit RX9D.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

##### Transmisión en el modo asíncrono de la USART:

El bloque de recepción de la USART incorpora un registro de desplazamiento serie (TSR) que se encarga de enviar en serie por el pin RC6/TX los bits del dato a transmitir. La lógica que el registro del TSR se encarga de enviar además de los 8 bits de datos el Start bit y el Stop bit. Si el bit TX9 está a '1' también se enviará como 9º bit el valor contenido en TX9D. La cadencia de esta transmisión viene determinada por la señal de reloj producida por el generador de velocidad de comunicación. Para que dicha señal de reloj llegue a TSR el bit TXEN debe estar '1' (habilitación de la transmisión).



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

---

##### Transmisión en el modo asíncrono de la USART (cont.):

El bit TRMT se pone a '1' cuando el registro TRMT está vacío (este bit es de solo lectura). El bit TRMT no está asociado a ningún mecanismo de interrupción, sirve únicamente para saber cuando un dato ha sido completamente transmitido.

El flag TXIF se pone a '1' cuando el registro TXREG está vacío. El bit **TXIF es de solo lectura**, se pone automáticamente a '0' cuando se vuelve a escribir un valor en TXREG y se mantiene a '0' hasta que dicho valor pase de TXREG a TSR). Si el bit TXIF está a '1' y el bit de habilitación de la interrupción de transmisión del EUSART está a '1' se generará una interrupción. Por lo tanto, **únicamente se debe habilitar la interrupción de transmisión del EUSART después de escribir en TXREG el primer dato que queremos transmitir y debemos deshabilitarla cuando se haya enviado el último dato**. De lo contrario se estarían generando continuamente interrupciones de transmisión del EUSART.

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

---

#### Proceso de configuración del canal EUSART en modo asíncrono:

##### **1º PASO: Configuración de las líneas RX y TX:**

- Poner a '1' el bit SPEN (RCSTA)
- Poner a '1' el bit 7 del registro TRISC (línea RC7/RX configurada como entrada)
- Poner a '0' el bit 6 del registro TRISC (línea RC6/TX configurada como salida)

##### **2º PASO: Establecer el modo asíncrono (bit SYNC a '0')**

##### **3º PASO: Establecer la velocidad de comunicación configurando adecuadamente los valores de los bits BRGH y BRG16 y de los bytes SPBRGH y SPBRG**

##### **4º PASO: Habilitar la recepción y/o la transmisión mediante los bits RCEN y TXEN respectivamente**

##### **5º PASO: Si se desea utilizar el 9º bit activar su uso en recepción (bit RX9 a '1') y/o en transmisión (bit TX9)**

##### **6º PASO: Si se desea trabajar con interrupciones se deben poner los bits GIE y PEIE como corresponda según los niveles de prioridad establecidos. Además:**

- Para habilitar la interrupción de recepción se debe poner a '1' el bit RCIE
- Para habilitar la interrupción de transmisión se debe poner a '1' el TXIE después de haber enviado el primer byte que se desea transmitir y ponerlo a '0' cuando se haya enviado el último byte.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

---

##### Proceso de recepción de datos por el canal EUSART:

La recepción de datos puede hacerse por dos métodos:

- Por “polling”: en este caso el programa debe monitorizar el estado del bit RCIF. Cuando este bit se ponga a ‘1’, el dato recibido podrá ser leído del registro RCREG. Si se están recibiendo un conjunto de datos, el código debe asegurar que un determinado dato será leído del registro RCREG antes de que llegue el siguiente. De lo contrario se producirá un error de sobrescritura (OERR=‘1’).
- Mediante la interrupción: cada vez que se recibe un dato, el flag RCIF se pone a ‘1’. Si se ha habilitado la interrupción de recepción del canal EUSART (bit RCIE a ‘1’) se producirá una interrupción y el uC saltará a la dirección de vectorización que corresponda según se hayan configurado las prioridades. En dicha dirección se debe escribir el código que permita leer del registro RCREG el dato recibido.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

---

##### Proceso de transmisión de datos por el canal EUSART:

Para enviar un dato por el canal EUSART basta con escribir el valor que se desea enviar en el registro TXREG.

Sin embargo si se quiere enviar varios datos consecutivos debe tenerse cuidado de no escribir un nuevo valor en TXREG antes de que el valor anterior haya pasado al registro TSR. Para ello es necesario saber cuando el registro TXREG queda libre. Esto puede hacerse de dos formas:

- Por “polling”: en este caso el programa debe monitorizar el estado del bit TXIF. Si TXIF está a ‘1’ significa que TXREG está vacío y por tanto podemos escribir el siguiente dato.
- Mediante la interrupción: si se habilita la interrupción de transmisión del canal EUSART se producirá un interrupción cada vez que el registro TXREG esté vacío. La interrupción de transmisión del canal serie nos permite por tanto detectar cuando TXREG queda libre para poder enviar un nuevo dato. Sin embargo si habilitamos dicha interrupción cuando no estamos transmitiendo datos se generarán interrupciones de forma continuada mientras TXREG esté vacío (lo que bloquearía el funcionamiento del uC). Por tanto solo se debe habilitar esta interrupción después de haber enviado el primer dato y debe deshabilitarse cuando se haya transmitido el último dato.



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

---

##### Interrupción de recepción del canal EUSART:

- Flag de interrupción: bit RCIF (PIR1)
- Bit de habilitación: bit RCIE (PIE1)
- Prioridad: bit RCIP (IPR1): '0' -> pr. baja / '1' -> pr. alta

Si se recibe un dato por el canal EUSART se pone a '1' el flag RCIF. Si el bit de habilitación RCIE está a '1' y las interrupciones están habilitadas a nivel global se genera una interrupción y el uC pasa a ejecutar el código situado a partir de la posición 0008H o 0018H (según el nivel de prioridad establecido).

##### Interrupción de transmisión del canal EUSART:

- Flag de interrupción: bit TXIF (PIR1)
- Bit de habilitación: bit TXIE (PIE1)
- Prioridad: bit TXIP (IPR1): '0' -> pr. baja / '1' -> pr. alta

Siempre que el registro TXREG esté vacío se pone a '1' el flag TXIF. Si el bit de habilitación TXIE está a '1' y las interrupciones están habilitadas a nivel global se genera una interrupción y el uC pasa a ejecutar el código situado a partir de la posición 0008H o 0018H (según el nivel de prioridad establecido). Para evitar que se produzcan continuamente interrupciones de transmisión del canal EUSART solo debe habilitarse dicha interrupción después de haber escrito en TXREG el primer dato que quiere enviar, y debe deshabilitarse cuando se haya transmitido el último dato.



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

---

##### Utilización del 9º bit en buses Maestro-Esclavo:

El canal EUSART dispone de un modo de funcionamiento (modo de reconocimiento automático de dirección) que facilita la operación del uC PIC18F4550 dentro de un bus maestro-esclavo.

Si el canal EUSART ha sido configurado en modo de recepción de 9 bits (RX9='1') y si el bit ADDEN se pone a '1' solo se procesará la recepción de aquellos datos cuyo noveno bit sea '1'.

Para utilizar este modo en un bus maestro-esclavo debe procederse de la siguiente forma:

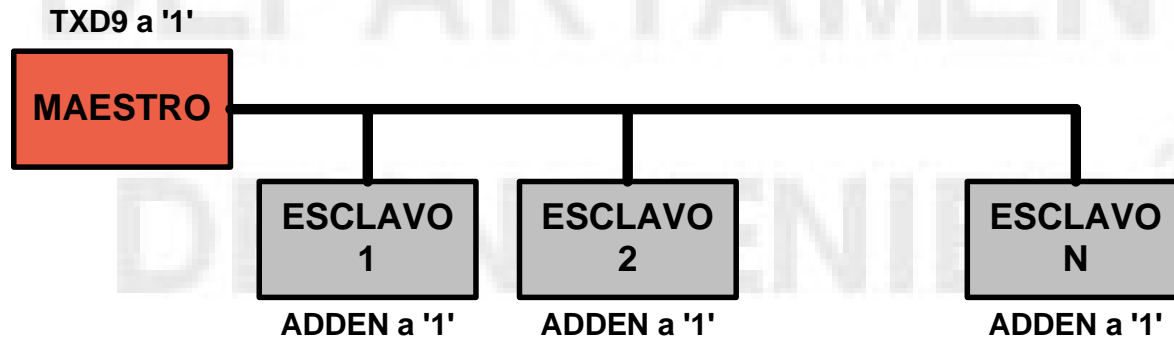
- Cuando el maestro quiere enviar una dirección debe poner el 9 bit a '1' (TX9D='1') y todos los esclavos deben habilitar el modo de reconocimiento automático de dirección (ADDEN='1'). De esta forma todos los esclavos reciben la dirección enviada por el maestro
- Una vez enviada la dirección debe empezar el intercambio de datos entre el maestro y el esclavo direccionado. Para ello, el esclavo que haya sido direccionado por el maestro debe salirse del modo de reconocimiento automático de dirección (ADDEN='0') y el maestro debe poner el 9º bit a '0' (TX9D='0'). A partir de ese momento los datos enviados por el maestro solo serán procesados por el esclavo direccionado, ya que el resto de esclavos se mantiene en el modo de reconocimiento automático de dirección.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

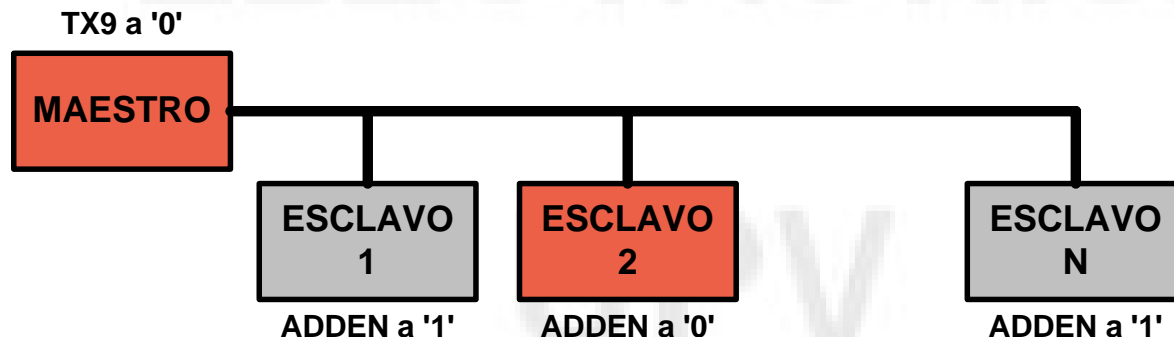
##### Utilización del 9º bit en buses Maestro-Esclavo (cont.):

##### 1º: ENVIO DE LA DIRECCION



El bit RCIF de todos los esclavos se pone a '1' cuando reciben bytes del maestro  
=> Todos los esclavos reciben la información enviada por el maestro

##### 2º: INTERCAMBIO DE DATOS



Solo el bit RCIF del esclavo 2 se pone a '1' cuando recibe bytes del maestro  
=> Solo el esclavo 2 recibe la información enviada por el maestro

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

---

##### EJERCICIO:

**Escribir en código de un programa para el uC PIC18F4550 que reciba datos por el canal EUSART con la siguiente configuración: modo asíncrono, 9600 baudios, 8 bits de datos, sin bit de paridad (9º bit). El uC debe recibir los datos y almacenarlos en un búfer situado en la memoria RAM de datos; cuando se hayan recibido 10 bytes el uC debe enviarlos a través del canal EUSART.**

**Se considera que el uC dispone de una  $F_{OSC}=20\text{MHz}$**



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

#### EJERCICIO (cont.):

#### SOLUCION EN ENSAMBLADOR

```
#include <P18F4550.INC>                ;processor specific variable definitions

;*****
; Declaración de variables
Ind_Buf EQU 0x00      ; Índice del bufer de comunicación
Buf_Com  EQU 0x080    ; Bufer de recepción/transmisión

;*****
; Vectorización del Reset
      ORG      0x0000
      goto     Main      ; Salto al inicio del programa principal

;*****
; Vectorización de las interrupciones de alta prioridad
      ORG      0x0008
      goto     HighInt   ; Salto a la rutina de tratamiento de las interr. de alta prioridad
```

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

#### EJERCICIO (cont.):

##### SOLUCION EN ENSAMBLADOR

; Programa Principal

Main:

; Inicialización de los puertos

bcf TRISC,6 ; Se configura RC6 para poder ser utilizado por la EUSART

bsf TRISC,7 ; Se configura RC7 para poder ser utilizado por la EUSART

; Inicialización de la USART

movlw B'00100100' ; Se inicializa la transmisión con 8 bits, en modo asíncrono,

movwf TXSTA ; sin envío del BREAK y con velocidad de comunicación alta (BRGH='1')

movlw B'10010000' ; Se inicializa la recepción con 8 bits y se configuran RC6

movwf RCSTA ; y RC7 como pines TX y RX

movlw B'00000000' ; Se inicializa el establecimiento de vel. de comun. con 8 bits (BRG='0')

movwf BAUDCON ; y se desactiva la autodetección de velocidad de comunicación

movlw .129 ; Se establece una velocidad de comunicación de 9600 baudios

movwf SPBRG ; Vel. Com.=Fosc/(16\*(SPBRG+1))=20000000/(16\*(129+1))=9615

; Habilitación de interrupciones

bcf RCON,IPEN ; Se deshabilitan la prioridad en las interrupciones

bsf INTCON,GIE ; Se habilitan a nivel global las interrupciones

bsf INTCON,PEIE ; Se habilitan a nivel global las interrupciones de periféricos

bsf PIE1,RCIE ; Se habilita la interrupción de recepción del canal serie

; Inicialización de variables

movlw .0 ; Se inicializa

movwf Ind\_Buf ; el índice del búfer a 0

lfsr FSR0,Buf\_Com ; Inicializamos FSR0 al inicio del búfer comunicación

Bucle:

goto Bucle



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

#### EJERCICIO (cont.):

#### SOLUCION EN ENSAMBLADOR

; Rutina de tratamiento de interrupciones de alta prioridad

HighInt:

```
        btfss PIR1,RCIF      ; Se comprueba si la interrupción ha sido por recepción USART
        goto Chk_IntTX      ; Si no es así se comprueba si ha sido por transmisión
; ***** Tratamiento de la interrupción de recepción del canal serie
        bcf PIR1,RCIF      ; Se pone a '0' el flag de int. por recepción
        btg PORTD,0        ; Se complementa el LED de PRUEBA
        movf Ind_Buf,W      ; Se pasa el valor del índice al acumulador
        movff RCREG,PLUSW0  ; Se almacena el valor leído en el búfer de com.
        incf Ind_Buf        ; Se incrementa el índice del búfer
        movlw .10          ; Se comprueba si el índice
        cpfseq Ind_Buf      ; del búfer ha llegado a 10 (se han recibido 10 datos)
        goto Fin_IntH      ; Si no es así, se va al final
        clrf Ind_Buf        ; Se inicializa el índice del búfer de com.
        movf Ind_Buf,W      ; Se pasa el valor del índice al acumulador
        movff PLUSW0,TXREG  ; Se escribe el dato del búfer en TXREG (se envía dato)
        incf Ind_Buf        ; Se incrementa el índice del búfer
        bsf PIE1,TXIE       ; Se habilita la interrupción por transmisión USART
        goto Fin_IntH
```

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

---

#### EJERCICIO (cont.):

#### SOLUCION EN ENSAMBLADOR

; \*\*\*\*\* Tratamiento de la interrupción de transmisión del canal serie

Chk\_IntTX:

btfss PIR1,TXIF	; Se comprueba si la interrupción ha sido por transmisión USART
goto Fin_IntH	; Si no es así se va al final
movf Ind_Buf,W	; Se pasa el valor del índice al acumulador
movff PLUSW0,TXREG	; Se escribe el dato del búfer en TXREG (se envía el dato)
incf Ind_Buf	; Se incrementa el índice del búfer
movlw .10	; Se comprueba si el índice
cpfseq Ind_Buf	; del búfer ha llegado a 10 (se han enviado 10 datos)
goto Fin_IntH	; Si no se han enviado 10 datos salimos
clrf Ind_Buf	; Se se han enviado los 10 datos se pone a 0 el índice
bcf PIE1,TXIE	; y se deshabilita la interrupción de transmisión USART

Fin\_IntH:

retfie            FAST

END

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

---

#### EJERCICIO (cont.):

##### SOLUCION EN C

```
#include <p18F4550.h>
#include <usart.h>
```

```
// Declaración de la rutina de tratamiento de las interrupciones de alta prioridad
void High_Int_Handler (void);
```

```
// Declaración de variables globales
unsigned char Buf_Com[10]; // Bufer de comunicación
unsigned char i; // Indice del bufer de comunicación
```

```
// Vectorización de las interrupciones de alta prioridad
#pragma code High_Interrupt = 0x8
void High_Int (void)
{
    _asm goto High_Int_Handler _endasm
}
#pragma code
```



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

#### EJERCICIO (cont.):

#### SOLUCION EN C

// Rutina de tratamiento de las interrupciones de alta prioridad

#pragma interrupt High\_Int\_Handler

void High\_Int\_Handler (void)

{

    if (PIR1bits.RCIF)           // Se comprueba si la interrupción ha sido por recepción

    {

        Buf\_Com[i] = RCREG;   // Se almacena el dato leído en la posición correspondiente  
                                  // del búfer (acceso directo al registro)

        //Buf\_Com[i] = ReadUSART(); // Se almacena el dato leído en la posición correspondiente  
                                  // del búfer (utilizando la función de la librería)

        i++;                   // Se incrementa el índice del búfer

        if (i==10)           // Se comprueba si ha llegado 10 datos

        {

            i=0;               // Se pone a 0 el índice para iniciar la transmisión

            TXREG=Buf\_Com[i];   // Se transmite el 1º dato (acceso directo al registro)

            //WriteUSART(Buf\_Com[i]); // Se transmite el 1º dato (utilizando la función de la lib.)

            i++;               // Se incrementa el índice del búfer

            PIE1bits.TXIE=1;   // Se habilita la interrupción por fin de transmisión

        }

    }



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

#### EJERCICIO (cont.):

#### SOLUCION EN C

```
else if (PIR1bits.TXIF) // Se comprueba si la interrupción ha sido por fin de transmisión
{
    if (i==10) // Se comprueba si se han transmitido los 10 datos
    {
        PIE1bits.TXIE=0; // Si se han transmitido los 10 datos
                        // se deshabilita la interr. por fin de transm.
        i=0; // Se pone a 0 el índice del búfer para la siguiente recepción
    }
    else
    {
        TXREG=Buf_Com[i]; // Si no se han transmitido los 10 datos se transmite el siguiente
        //WriteUSART(Buf_Com[i]); // Si no se han transmitido los 10 datos
                                // se transmite el siguiente (utilizando la función de la lib.)
        i++; // Se incrementa el puntero del búfer
    }
}
}
```

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / CANAL DE COMUNICACIÓN SERIE EUSART

#### EJERCICIO (cont.):

#### SOLUCION EN C

void main (void) // Programa Principal

```
{
    TRISCbits.RC6=0; // Se configura la línea RC6/TX como salida
    TRISCbits.RC7=1; // Se configura la línea RC7/RX como entrada
    TXSTA=0x24;      // Se inicializa la transmisión con 8 bits ,en modo asíncrono,
                    // sin envío del BREAK y con velocidad de comunicación alta (BRGH='1')
    RCSTA=0x90;      // Se inicializa la recepción con 8 bits y se configuran RC6 y RC7 como pines TX y RX
    BAUDCON=0x00;    // Se inicializa la vel. de comun. con 8 bits (BRG='0')
                    //y se desactiva la auto detección de velocidad de comunicación
    SPBRG=129;       // Vel. Com.=Fosc/(16*(SPBREG+1))=20000000/(16*(129+1))=9615
    // Se configura la USART en modo 8 bits, sin paridad, 1 Stop bit, 9600 baud
    // e interr. de recepción habilitada (utilizando la función de la librería)
    //OpenUSART (USART_TX_INT_OFF & USART_RX_INT_ON & USART_ASYNCH_MODE &
    //           USART_EIGHT_BIT & USART_CONT_RX & USART_BRGH_HIGH, 129);
    RCONbits.IPEN = 0; // Se desactiva el sist. de prioridades de interrup. (todas las interr. misma prioridad)
    INTCONbits.GIE = 1; // Se activan las interrupciones a nivel global
    INTCONbits.PEIE = 1; // Se activan las interrupciones de periféricos a nivel global
    PIE1bits.RCIE=1;    // Se habilita la interrupción de recepción del canal serie
    i=0;                // Se inicializa en índice del búfer
    while (1);          // Bucle sin fin
}
```

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 2

---

#### TEMPORIZADOR 2:

##### Características fundamentales:

- Temporizador de 8 bits (registro TMR2)
- Registro de periodo PR2
- Pre-escalar de 2 bits programable (1:1, 1:4, 1:16)
- Post-escalar de 4 bits (1:1...1:16 )
- Interrupción por igualdad entre TMR2 y PR2
- Se puede utilizar junto con los módulos CCP y ECCP
- Se puede utilizar como señal de reloj del módulo MSSP en modo SPI

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / TEMPORIZADOR 2

#### REGISTRO T2CON

	-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
T2CON	-	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0

- **T2OUTPS3..T2OUTPS0** : Bits de selección del post-escalar del Temporizador 2:

T2OUTPS3..T2OUTPS0	Post-escalar	T2OUTPS3..T2OUTPS0	Post-escalar
0000	1:1	1000	1:9
0001	1:2	1001	1:10
0010	1:3	1010	1:11
0011	1:4	1011	1:12
0100	1:5	1100	1:13
0101	1:6	1101	1:14
0110	1:7	1110	1:15
0111	1:8	1111	1:16

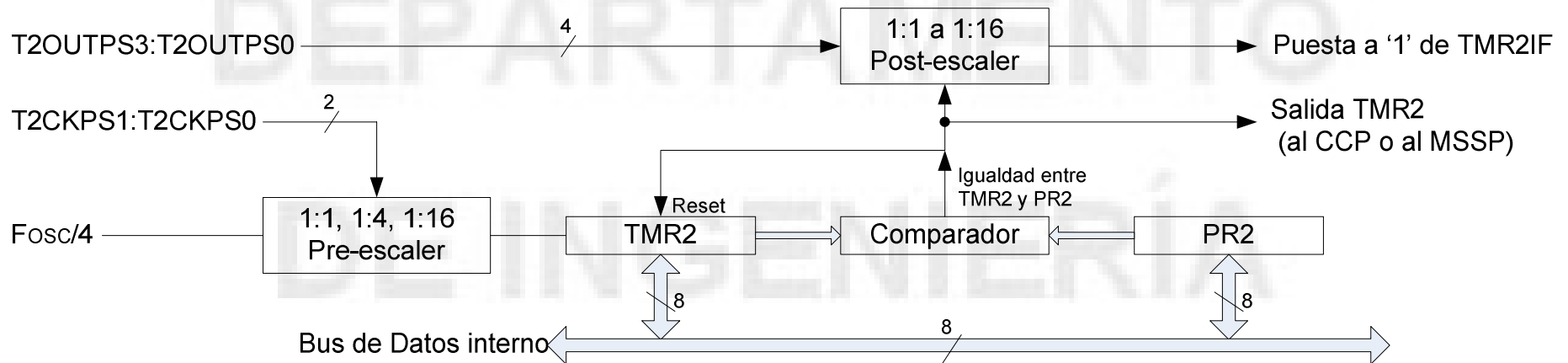
- **TMR2ON**: Bit de puesta en marcha del Temporizador 2
- **T2CKPS1..T2CKPS0**: Bits de selección del pre-escalar del Temporizador 2:

T2CKPS1	T2CKPS0	Valor del pre-escalar
0	0	1:1
0	1	1:4
1	0	1:16
1	1	

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 2

#### Temporizador 2:



**Los registros TMR2 y PR2 son de lectura escritura.**

**Los contadores del pre-escalar y el post-escalar no son accesibles por el usuario. Se ponen a 0 automáticamente cuando:**

- Se escribe en TMR2
- Se escribe en T2CON
- Se produce un reset

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 2

---

#### Funcionamiento del Temporizador 2:

El registro TMR2 se incrementa con cada pulso de reloj de una señal que se obtiene haciendo pasar la señal de frecuencia  $F_{OSC}/4$  por un pre-escalar. Los bits T2CKPS1 y T2CKPS0 permiten seleccionar si la frecuencia de la señal que incrementa TMR2 es de  $F_{OSC}/4$ ,  $F_{OSC}/16$  o  $F_{OSC}/64$ .

Cuando el valor de TMR2 se iguala con el valor del registro de periodo PR2:

- Se reinicia el valor de TMR2
- Se activa la señal de salida del Temporizador 2

La señal de salida del Temporizador 2 puede utilizarse en el módulo CCP para generar señales PWM o en el módulo MSSP como señal de reloj del modo SPI. La señal de salida del Temporizador está conectada a un contador post-escalar que se incrementa cada vez que se produce la igualdad entre TMR2 y PR2. Cuando el contador post-escalar desborda se pone a '1' el flag de interrupción del Temporizador 2 (T2IF). El número de pulsos de la señal de salida del Temporizador 2 necesarios para provocar el desbordamiento del post-escalar puede configurarse de 1 a 16 en función de los T2OUTPS3..T2OUTPS0.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / TEMPORIZADOR 2

#### EJERCICIO:

Escribir en código de un programa para el uC PIC18F4550 que genere por el pin RD0 una señal PWM de 1kHz de frecuencia y de un ciclo de trabajo del 75% mediante la interrupción del Temporizador 2 ( $F_{OSC}=4\text{MHz}$ ).

; Bloque de declaraciones y vectorizaciones

```
#include <p18F4520.inc>
TON EQU .75 ; Valor del nivel alto para un ciclo de trabajo de 75%

ORG 0x00 ; Vectorización del Reset
GOTO Inicio
ORG 0x08 ; Vectorización de interrupciones
GOTO Int2_ISR ; de alta prioridad

; Programa Principal
ORG 0x20
Inicio: MOVLW 0xFE ; Se configura RD0 como salida
        MOVWF TRISD ;
        MOVLW TON ; Se carga en PR2 el valor
        MOVWF PR2 ; correspondiente al nivel alto de la señal (TON=75)
        BSF PORTD,0 ; Se pone RD0 a '1' (se empieza con nivel alto)
        MOVLW b'01001100' ; Se configura el Temp. con un pre-escalar 1:1 y
        MOVWF T2CON ; un post-escalar de 1:10. Se pone en marcha el Temp. 2
        BSF PIE1,TMR2IE ; Se habilita la interr. del Temp.2 a nivel individual
        BSF INTCON,GIE ; Se habilitan las interrupciones a nivel global
        BSF INTCON,PEIE ; Se habilitan las interr. de periferico a nivel global
Bucle: GOTO Bucle ; Bucle sin fin
```





## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 2

#### EJERCICIO:

; Interrupción del Temporizador 2

Int2\_ISR:

BTFSS PIR1,TMR2IF	; Se comprueba si la interr. es por el Temp. 2
GOTO Fin_int2	; Si no es así se va al final
BCF PIR1,TMR2IF	; Se borra el flag de interr. del Temp. 2
BTFSC PORTD,0	; Se comprueba el estado actual
GOTO E_BAJO	; Si el estado actual es estado alto se pasa al estado bajo
MOVLW TON	; Si el estado actual es estado bajo se carga la temporización
MOVWF PR2	; del estado alto en PR2
BSF PORTD,0	; Se pone la RD0 a '1' para iniciar el estado alto
GOTO Fin_int2	; Se va al final

E\_BAJO

MOVLW TON	; Se calcula la temporización de estado bajo:
SUBLW .100	; Temp. estado bajo = 100-TON
MOVWF PR2	; Se carga el resultado en PR2
BCF PORTD,0	; Se pone RD0 a '0' para iniciar el estado bajo

Fin\_int2:

RETIE  
END



### **TEMPORIZADOR 3:**

#### **Características fundamentales:**

- **Configurable como temporizador/contador de 16 bits**
- **Dispone de varias opciones de señal de reloj en el modo temporizador:**
  - **Oscilador principal con o sin pre-escalar**
  - **Oscilador del temporizador 1 con o sin pre-escalar**
- **Pre-escalar de 3 bits programable**
- **Interrupción por desbordamiento**

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 3

#### REGISTRO T3CON

	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON

- **RD16:** Bit selección de modo de lectura/escritura de TMR3
  - \* RD16='0': Lectura/escritura de TMR3 en dos operaciones independientes de 8-bit
  - \* RD16='1': Lectura/escritura de TMR3 en modo de 16-bit
- **T3CCP2.. T3CCP1:** Bits de selección de la fuente de reloj para los módulos CCP y ECCP:

T3CCP2	T3CCP1	Modo
0	0	Temp. 1 fuente de reloj para ECCP y CCP
0	1	Temp. 1 fuente de reloj para ECCP/ Temp. 3 fuente de reloj para CCP
1	0	Temp. 3 fuente de reloj para ECCP y CCP
1	1	

- **T3CKPS1..T3CKPS0:** Bits de selección del pre-escalar del Temporizador 3

T3CKPS1	T3CKPS0	Valor del pre-escalar
0	0	1:1
0	1	1:4
1	0	1:8
1	1	1:16

- **T3SYNC:** Bit de sincronización de la señal de reloj (solo aplica cuando TMR3CS='1')
  - \* T3SYNC='0': Se sincroniza la entrada de reloj externa RC0/T1OSO/T13CKI
  - \* T3SYNC='1': No se sincroniza la entrada de reloj externa RC0/T1OSO/T13CKI

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 3

---

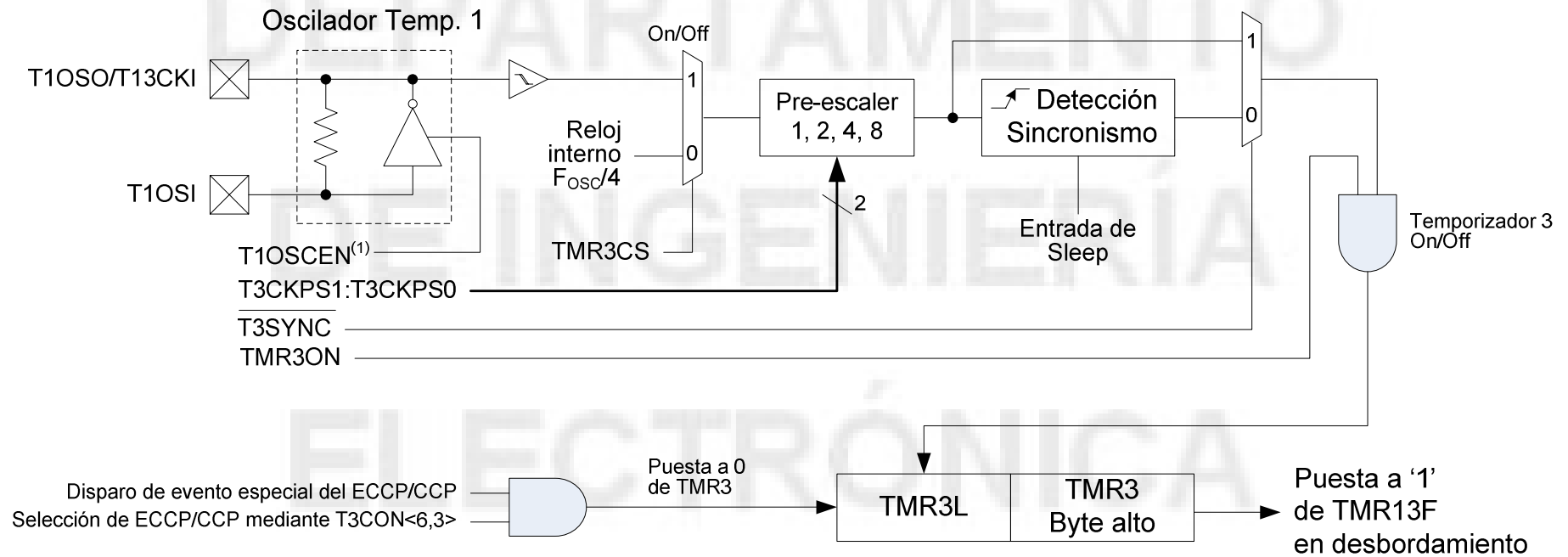
#### REGISTRO T3CON (cont)

- **TMR3CS**: Bits de selección de la fuente de incremento del Temporizador 3
  - \* TMR3CS='0': Pre-escalar del Temp. 3 se incrementa con cada ciclo de instrucción  $F_{osc}/4$
  - \* TMR3CS='1': Pre-escalar del Temp. 3 se incrementa con cada flanco de subida de la línea RC0/T1OSO/T13CKI
- **TMR3ON**: Bit de puesta en marcha del Temporizador 3

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 3

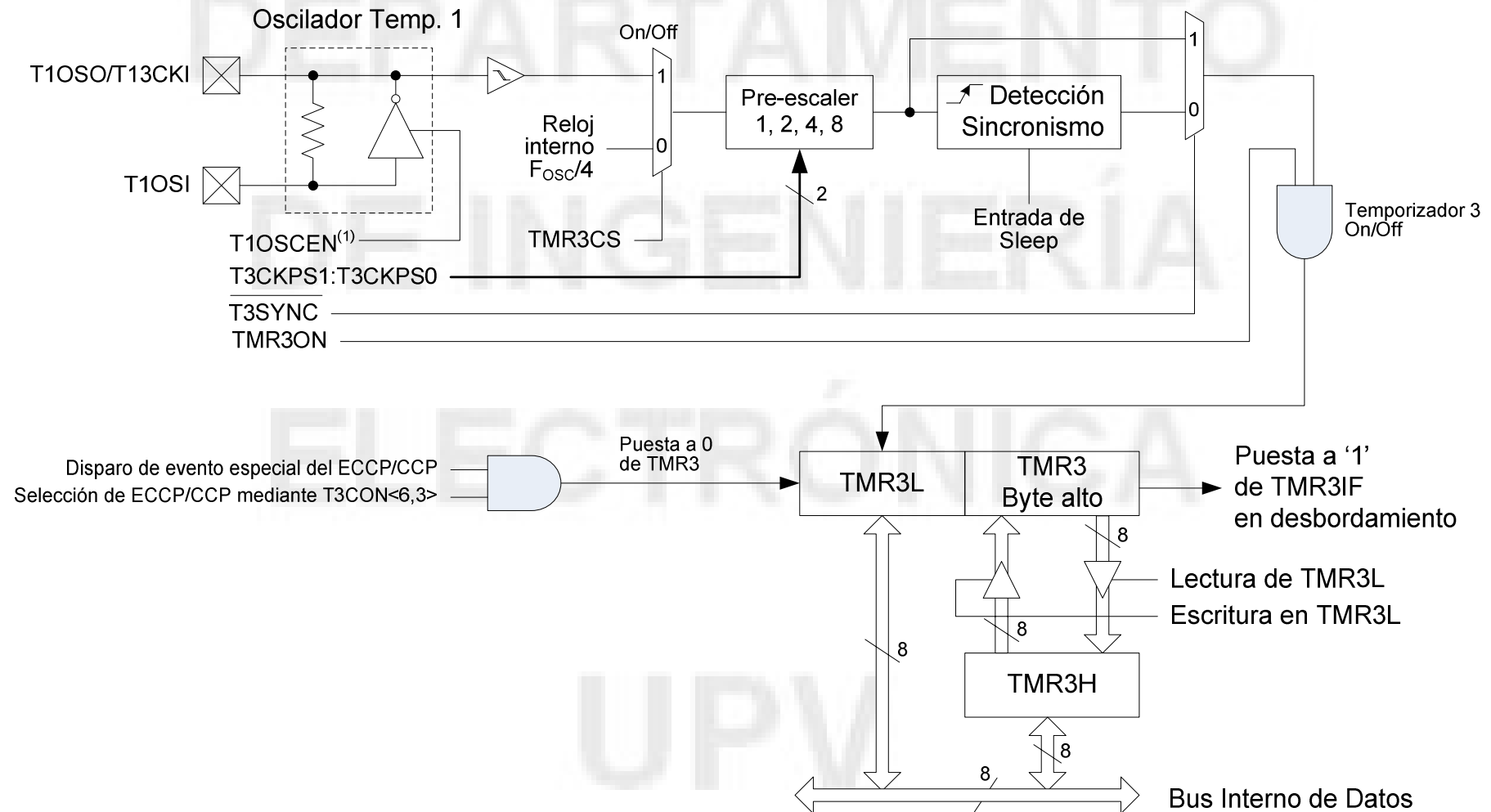
**Diagrama de bloques (TMR3L y TMR3H registros de L/E independiente):**



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 3

#### Diagrama de bloques (modo lectura/escritura 16 bits):



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 3

---

#### Acceso al registro TMR3:

Existen dos modos de acceder al registro TMR1:

- RD16 (T3CON)='0': se accede a TMR3L y TMR3H como dos registros independientes.
- RD16 (T3CON)='1': la parte alta de TMR3 no es accesible directamente. Se accede a ella a través de TMR3H que funciona como un búfer:
  - Cuando se lee el valor de TMR3L, el valor de la parte alta de TMR3 pasa al registro TMR3H. Para leer el valor de TMR3 primero leemos TMR3L y luego leemos TMR3H.
  - Cuando se escribe en TMR3L, el valor de TMR3H pasa a la parte alta de TMR3. Para escribir en TMR3 primero escribimos la parte alta en TMR3H y luego escribimos la parte baja en TMR3L.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / TEMPORIZADOR 3

##### Modos de funcionamiento:

- Temporizador de 16 bits (oscilador principal) TMR3CS='0': el pre-escalar del Temporizador 3 se incrementa en cada ciclo de instrucción ( $F_{OSC}/4$ ) mientras el bit TMR3ON esté a '1'.
- Temporizador de 16 bits (oscilador Temp. 1) TMR3CS='1' / T1OSCEN='1': el pre-escalar del Temporizador 3 se incrementa en cada pulso de la señal de reloj del oscilador del Temporizador 1 ( $F_{OSCT1}$ ). La señal de reloj del oscilador del Temporizador 1 puede sincronizarse con el oscilador principal poniendo el bit T3SYNC a '0'.
- Contador síncrono TMR1CS='1' / TMR1OSCEN='0' / T1SYNC='0': TMR1 se incrementa de forma síncrona con cada flanco de subida de la línea T13CKI (RC0).
- Contador asíncrono TMR1CS='1' / TMR1OSCEN='0' / T1SYNC='1': TMR1 se incrementa de forma asíncrona con cada flanco de subida de la línea T13CKI (RC0).

Para los 4 modos de funcionamiento existe la posibilidad de configurar el pre-escalar mediante los bits T3CKPS1 y T3CKPS0 con 4 posibles ratios (1:1, 1:2, 1:4 y 1:8).



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / TEMPORIZADOR 3

---

#### Interrupción del Temporizador 3:

- Flag de interrupción: bit TMR3IF (PIR2)
- Bit de habilitación: bit TMR3IE (PIE2)
- Prioridad: bit TMR3IP (IPR2): '0' -> pr. baja / '1' -> pr. alta

Si se produce el desbordamiento del Temporizador 3 se pone a '1' el flag TMR3IF. Si el bit de habilitación TMR3IE está a '1' y las interrupciones de periféricos están habilitadas a nivel global se genera una interrupción y el uC pasa a ejecutar el código situado a partir de la posición 0008H o 0018H (según el nivel de prioridad establecido).

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM (CCP)

##### MODULO DE COMPARACION/CAPTURA/PWM:

- **Dispone de tres modos de funcionamiento:**
  - **Modo de Captura:** se utiliza para medir eventos externos como la duración de pulsos digitales.
  - **Modo de Comparación:** se utiliza para generar señales digitales con temporizaciones programables. Este tipo de señales son muy útiles para el control de etapas de potencia (convertidores DC/DC, DC/AC, AC/DC, AC/DC).
  - **Modo PWM:** se utiliza para generar señales de modulación de ancho de pulso (PWM).
- **Elementos asociados:**
  - **Temporizador asociado:** Temporizador 1 o Temporizador 3 en función de los bits T3CCP2..T3CCP0 (registro T3CON)
  - **Registros de comparación:** CCP2RH y CCPR2L
  - **Línea de E/S:** RC1/T1OSI/CCP2 o RB3/AN9/CCP2 en función del valor del bit de configuración CCP2MX (registro de configuración CONFIG3H):
    - CCP2MX='0': CCP asociado al pin RB3/AN9/CCP2
    - CCP2MX='1': CCP asociado al pin RC1/T1OSI/CCP2 (configuración por defecto)
  - **Interrupción asociada a los modos de Captura y Comparación (flag de interrupción CCP2IF)**

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM (CCP)

##### REGISTRO CCP2CON

	-0	-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
CCP2CON	-	-	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0

- **DC2B1..DC2B0**: Bits menos significativos (bit 0 y bit 1) del ciclo de trabajo en la señal PWM (solo en modo PWM)
- **CCP2M3.. CCP2M0**: Bits de selección del modo del módulo CCP:

CCP2M3	CCP2M2	CCP2M1	CCP2M0	Modo
0	0	0	0	Módulo CCP deshabilitado
0	0	0	1	Reservado
0	0	1	0	Modo Comparación, coincidencia => complemento de la RC1 o RB3 y CCP2IF a '1'
0	0	1	1	Reservado
0	1	0	0	Modo Captura, cada flanco de bajada
0	1	0	1	Modo Captura, cada flanco de subida
0	1	1	0	Modo Captura, cada 4 flancos de subida
0	1	1	1	Modo Captura, cada 16 flancos de subida
1	0	0	0	Modo Comparación, RC1 (o RB3) inicialmente a '0', coincidencia => puesta a '1' y CCP2IF a '1'
1	0	0	1	Modo Comparación, RC1 (o RB3) inicialmente a '1', coincidencia => puesta a '0' y CCP2IF a '1'
1	0	1	0	Modo Comparación, coincidencia => CCP2IF a '1' (no hay cambio en RC1 (o RB3))
1	0	1	1	Modo Comparación, coincidencia => reset del Temporizador, inicio la conversión A/D y CCP2IF a '1'
1	1	0	0	Modo PWM
1	1	0	1	
1	1	1	0	
1	1	1	1	

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM (CCP)

---

#### Modo de Captura:

En el modo de captura el valor del Temporizador 1 o del Temporizador 3 pasa al par de registros CCPR2H;CCPR2L cuando se produce alguno de los siguientes eventos en el pin del CCP (RB3 o RC1):

- En cada flanco de bajada (CCP2M3..CCP2M0="0100")
- En cada flanco de subida (CCP2M3..CCP2M0="0101")
- Cada 4 flancos de subida (CCP2M3..CCP2M0="0110")
- Cada 16 flancos de subida (CCP2M3..CCP2M0="0111")

La selección del Temporizador que se captura se realiza mediante los bits T3CCP2 y T3CCP1 del registro T3CON:

- Si T3CCP2='0 y T3CCP1='0' el temporizador capturado será el Temporizador 1
- En cualquier otro caso el temporizador capturado será el Temporizador 3

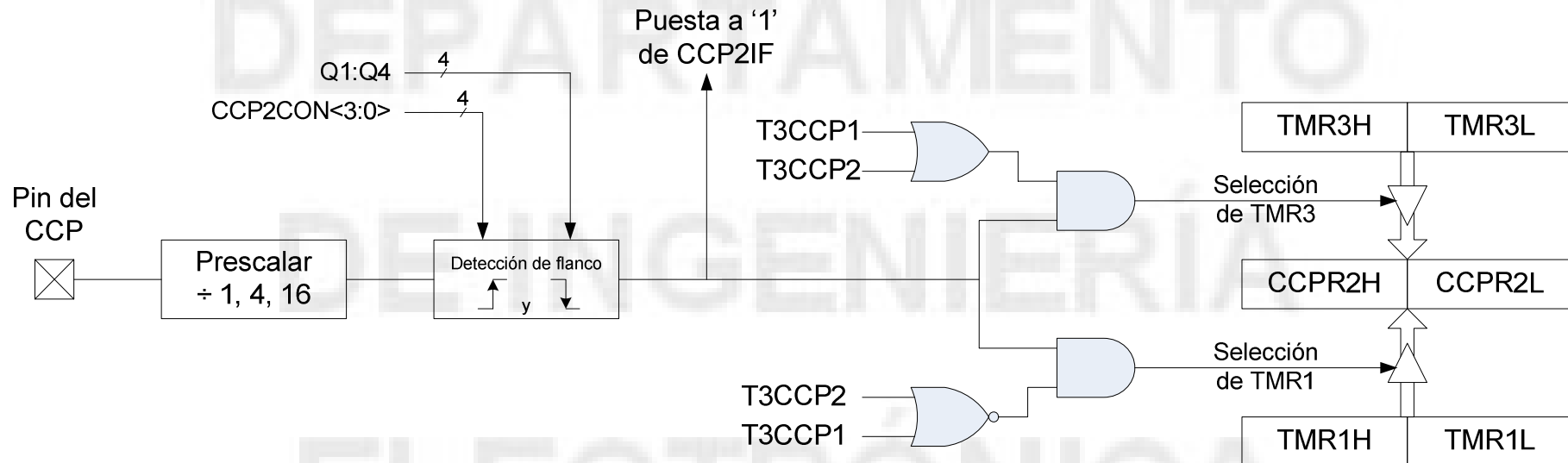
El temporizador que se utilice debe configurarse en modo Temporizador o en modo contador síncrono. Si se configura en modo contador asíncrono el modo captura no funcionará.

Cuando la condición de captura se cumple el flag de interrupción CPP2IF se pone a '1'.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM (CCP)

##### Diagrama de bloques del modo de Captura:



**Cuando se realiza un cambio de modo de captura el valor del prescalador no se inicializa, además el bit CCP2IF puede ponerse a '1' de forma no controlada. Para evitar esta falsa interrupción y resetear el prescalador, antes de cambiar de modo de captura, debe desactivarse el modo captura (CCP2CON=0x00) y a continuación establecer el nuevo modo de captura.**

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM (CCP)

---

#### Modo de Comparación:

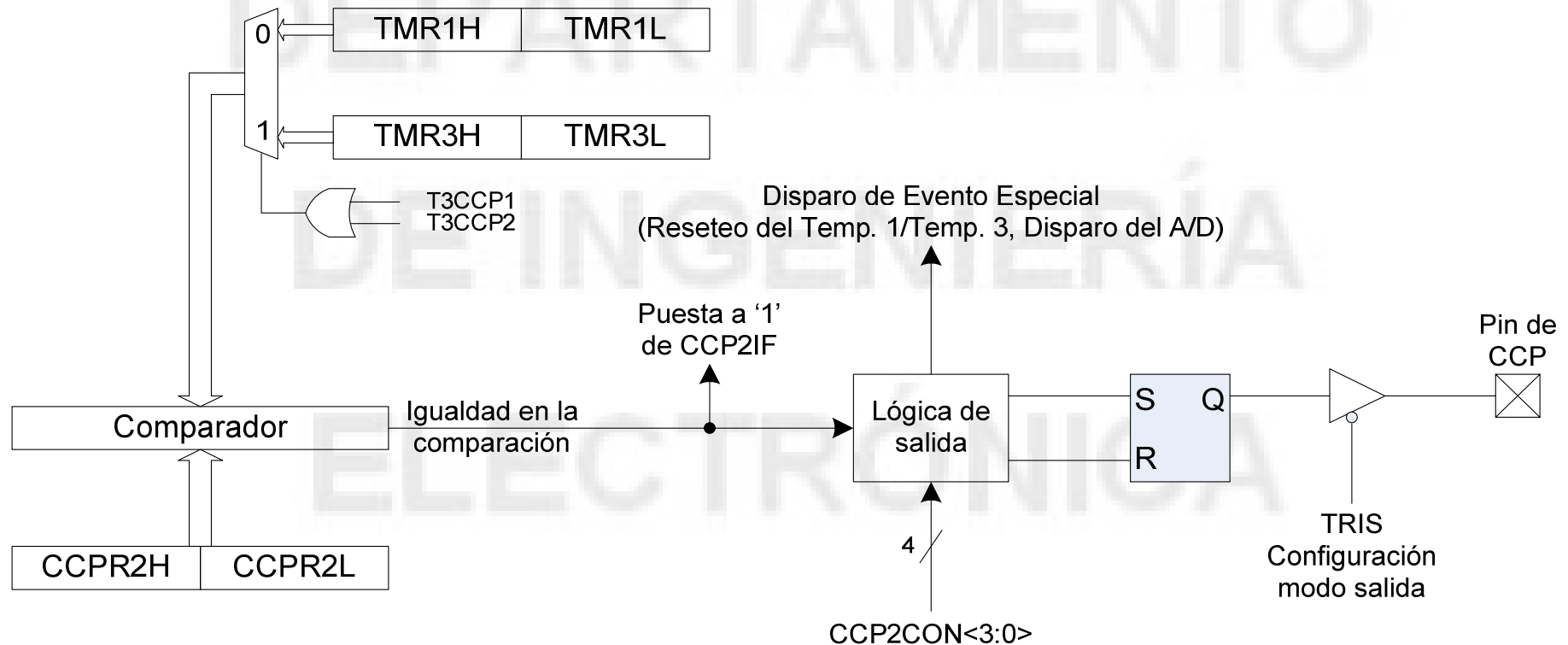
En el modo de comparación el par de registros [CCPR2H;CCPR2L] se comparan continuamente con el Temporizador asociado (Temporizador 1 o 3). Existen varios modos de comparación en función del valor de los bits: CCP2M3..CCP2M0:

- CCP2M3..CCP2M0="0010": El pin de salida del CCP2 se inicializa a '0'. Cada vez que se produce la igualdad entre [CCPR2H;CCPR2L] y el Temporizador asociado **se complementa** el pin de salida y se pone a '1' el flag CCP2IF.
- CCP2M3..CCP2M0="1000": El pin de salida del CCP2 se inicializa a '0'. Cuando se produce la igualdad entre [CCPR2H;CCPR2L] y el Temporizador asociado el pin de salida se pone a '1' y se pone a '1' el flag CCP2IF.
- CCP2M3..CCP2M0="1001": El pin de salida del CCP2 se inicializa a '1'. Cuando se produce la igualdad entre [CCPR2H;CCPR2L] y el Temporizador asociado el pin de salida se pone a '0' y se pone a '1' el flag CCP2IF.
- CCP2M3..CCP2M0="1010": El pin de salida del CCP2 toma el valor del latch correspondiente. Cuando se produce la igualdad entre [CCPR2H;CCPR2L] y el Temporizador asociado se pone a '1' el flag CCP2IF.
- CCP2M3..CCP2M0="1011": El pin de salida del CCP2 toma el valor del latch correspondiente. Cuando se produce la igualdad entre [CCPR2H;CCPR2L] y el Temporizador asociado se inicia una conversión A/D (en caso de que el conversor A/D esté adecuadamente configurado), se inicializa a 0 el Temporizador asociado y se pone a '1' el flag CCP2IF.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM (CCP)

##### Diagrama de bloques del modo de Comparación:



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM (CCP)

##### EJERCICIO:

Escribir en código de un programa para el uC PIC18F4550 que genere por el pin RC1 una señal PWM utilizando el modo de comparación del CCP. El ciclo de trabajo de la señal expresado en tanto por ciento (%) debe establecerse mediante una constante: DUTY. ( $F_{OSC}=20\text{MHz}$ ).

```
#include <p18F4550.h> // Declaración de librerías
const int DUTY=50;    // Valor del duty de la señal generada (en %)
int Ton;              // Variable correspondiente a la duracion del estado alto

void R_Int_Alta(void); // Declaración rutina de tratam. de interrupciones de alta prioridad

// Vectorización de interrupciones de alta prioridad
#pragma code Vector_Int_Alta=0x08
void Int_Alta (void)
{
    _asm GOTO R_Int_Alta _endasm // Se salta a la rutina de tratamiento de
    // interrupciones de alta prioridad
}
#pragma code
```



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM (CCP)

#### EJERCICIO:

```
// Rutina de tratamiento de interrupciones de alta prioridad
#pragma interrupt R_Int_Alta
void R_Int_Alta (void)
{
    if (PIR2bits.CCP2IF)          // Se comprueba si se ha producido interrupción del CCP
    {
        PIR2bits.CCP2IF=0;       // Si se ha producido interrupción del CCP
        // Se borra el flag de interrupción del CCP
        if (CCPR2==0)            // Se comprueba el valor anterior de CCPR2
        {
            // Si CCPR2=0:
            CCPR2=65536 - Ton;    // Se carga la temporización del nivel bajo:
            // desde 0 hasta 65536-Ton
        }
        else
        {
            // Si CCPR2<>0:
            CCPR2= 0;            // Se carga la temporización de nivel alto:
            // desde 65536-Ton hasta el overflow (es decir 0)
        }
    }
}
```

**¿Cuál sería la frecuencia de la señal generada?**

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM (CCP)

#### EJERCICIO:

// Programa Principal

void main (void)

{

TRISC=0xFD; // Se inicializa RC1 como salida

T3CON=0x41; // Se configura el temporizador 3 como temporizador,  
// sin prescalar, TMR3 fuente de CCP2 y se pone en marcha

Ton=65536/100; // Se calcula el valor correspondiente a la duración del

Ton=Ton\*DUTY; // nivel alto: Ton= (65536\*DUTY)/100

CCPR2=65536-Ton; // Se carga en CCPR2 la duración del nivel bajo

CCP2CON=0x0A; // Se configura en CCP2 en modo de comparación con  
// complemento de RC1 en cada coincidencia entre  
// CCPR2 y TMR3 (RC1 empieza con nivel bajo)

INTCONbits.PEIE=1; // Se habilitan a nivel global las interrupciones de periféricos

INTCONbits.GIE=1; // Se habilitan a nivel global todas las interrupciones

PIE2bits.CCP2IE=1; // Se habilita la interrupción del CCP2

while (1)

{

}

}



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM (CCP)

---

#### Modo PWM:

**En el modo PWM permite generar una señal PWM de ciclo de trabajo y frecuencia programables.**

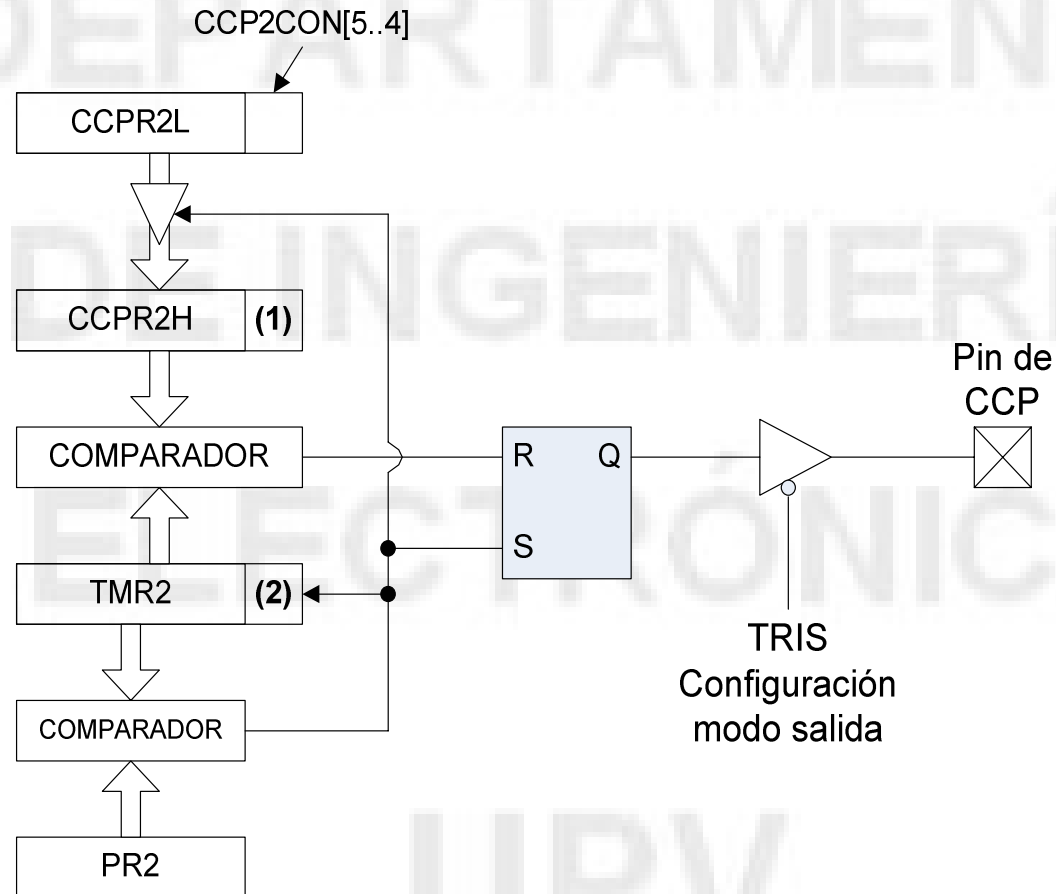
**El funcionamiento en este modo es el siguiente:**

- **Un registro de 10 bits (compuesto por CCPR2H y por un registro interno de 2 bits) se compara constantemente con un contador de 10 bits (compuesto por el registro TMR2 del Temporizador 2 y un prescalar de 2 bits).**
- **Cuando el valor del contador se hace igual al del registro se pone a '0' el pin de salida del CCP.**
- **En paralelo el registro PR2 se compara constantemente con el registro TMR2 Temporizador 2. Cuando el valor de TMR2 alcanza PR2:**
  - **Se pone a '1' el pin de salida del CCP**
  - **Se inicializa a 0 el valor del contador de 10 bits (registro TMR2 y el prescalar de 2 bits)**
  - **El valor del registro de 10 bits (registro CCPR2H y registro interno de 2 bits) se recargan con el valor del registro CPPR2L y los bits CCP2CON[5..4].**

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM (CCP)

##### Diagrama de bloques del modo PWM:



- (1) Registro interno de 2 bits que permite ampliar el registro CCPR2H a 10 bits  
(2) Prescalar de 2 bits que permite obtener una base de tiempos de 10 bits

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM (CCP)

##### Modo PWM:

Para generar la señal PWM se deben seguir los siguientes pasos:

- Configurar el pin del CCP (RC1 o RB3) como salida
- Configurar el Temporizador 2 para que trabaje como temporizador. Puede utilizarse el prescalar del Temporizador 2, pero no puede utilizarse el post-escalar.
- Configurar el módulo CCP en modo PWM.
- Establecer el periodo de la señal PWM mediante el valor del registro PR2. El periodo de la señal PWM será:

$$\text{PERIODO } (T_{\text{PWM}}) = \frac{(PR2 + 1) \cdot 4 \cdot (\text{Prescalar del Temporizador 2})}{F_{\text{OSC}}}$$

- Establecer el ciclo de trabajo de la señal mediante el registro CCP2L y los bits CCP2CON[5..4]. El ciclo de trabajo de la señal (valor entre 0 y 1) será:

$$\text{CICLO DE TRABAJO} = \frac{([CCPR2L; CCP2CON[5..4]]) \cdot (\text{Prescalar del Temporizador 2})}{F_{\text{OSC}} * T_{\text{PWM}}}$$

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM (CCP)

#### EJERCICIO:

Escribir en código de un programa para el uC PIC18F4550 que genere por el pin RC1 una señal PWM de 2kHz de frecuencia utilizando el modo PWM del CCP. El ciclo de trabajo de la señal expresado en tanto por ciento (%) debe establecerse mediante una constante: DUTY ( $F_{OSC}=20\text{MHz}$ ).

```
#include <p18F4550.h> // Declaración de librerías
const int DUTY=75;           // Valor del duty de la señal generada (en %)
unsigned short long CTon;     // Variable correspondiente a la duración del estado alto
void main (void)
{
    TRISC=0xFD;               // Se inicializa RC1 como salida
    T2CON=0x07;               // Se configura el Temp. con el prescalar a 1:16
                                // y el post-escalar a 1:1 y se pone en marcha
    CCP2CON=0x0F;             // Se configura en CCP en modo de PWM
    PR2=155;                  // Se fija el periodo  $T_{pwm}=0,5\text{ms}$  ( $F_{pwm}=2\text{kHz}$ )
                                //  $T_{pwm} = ((PR2+1)*4*Prescaler)/F_{osc}$ 
                                // =>  $PR2 = [(F_{osc}*T_{pwm})/(Prescaler*4)]-1 = [(20E6*0,5E-3)/(16*4)] = 155$ 
    CTon=DUTY;                // Se calcula el valor de cuenta para la duracion del nivel alto
    CTon=625*CTon;             //  $DUTY(\%) = (CTon*Prescaler*100)/(F_{osc}*T_{pwm})$ 
    CTon=CTon/100;             // =>  $CTon = (DUTY*F_{osc}*T_{pwm})/(Prescaler*100) = (DUTY*625)/100$ 
    CCP2L=CTon/4;              // Se carga en CCP2L la duración del nivel alto (8 bits mas significativos)
    CCP2CONbits.DC2B0=CTon&0x0001; // Se cargan los 2 bits menos significativos de la
    CCP2CONbits.DC2B1=CTon&0x0002; // duración del nivel alto en DC2B0 y DC2B1
    while (1);
}
```

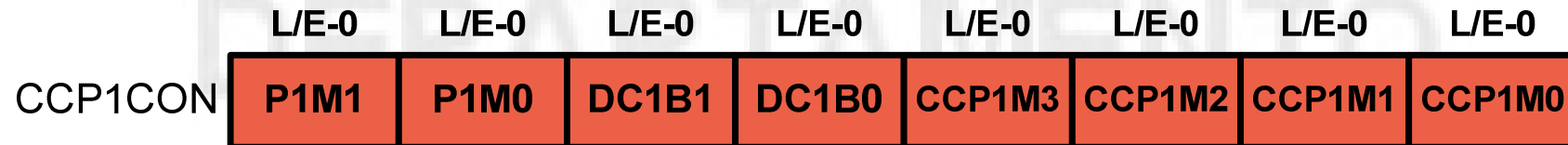
#### MODULO DE COMPARACION/CAPTURA/PWM MEJORADO:

- **Dispone de cuatro modos de funcionamiento:**
  - **Modo de Captura:** se utiliza para medir eventos externos como la duración de pulsos digitales.
  - **Modo de Comparación:** se utiliza para generar señales digitales con temporizaciones programables. Este tipo de señales son muy útiles para el control de etapas de potencia (convertidores DC/DC, DC/AC, AC/DC, AC/DC).
  - **Modo PWM:** se utiliza para generar señales de modulación de ancho de pulso (PWM).
  - **Modo PWM mejorado:** se utiliza para generar señales PWM complementarias para el control de semipuentes de transistores.
- **Elementos asociados:**
  - **Temporizador asociado:** Temporizador 1 o Temporizador 3 en función de los bits T3CCP2..T3CCP0 (registro T3CON)
  - **Registros de comparación:** CCP1RH y CCPR1L
  - **Línea de E/S:** RC2/CCP1/P1A, RD5/SPP5/P1B, RD6/SPP6/P1C y RD7/SPP7/P1D
  - **Interrupción asociada a los modos de Captura y Comparación (flag de interrupción CCP1IF)**

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### REGISTRO CPP1CON



– **P1M1..P1M0:** Bits de configuración del modo PWM:

- \* Si CCP1M3..CCP1M2="00", "01" o "10": Los bits P1M1 y P1M0 no tienen efecto sobre el funcionamiento de la ECCP
- \* Si CCP1M3..CCP1M2="11": Los bits P1M1 y P1M0 establecen la configuración de las líneas de salida PWM:

P1M1	P1M0	Modo
0	0	Modo PWM estándar: RC2/P1A salida PWM; RD5/P1B, RD6/P1C y RD7/P1D líneas de E/S
0	1	Modo PWM puente completo con salida directa: RD7/P1D modulada; RC2/P1A activa; RD6/P1C y RD7/P1D inactivas
1	0	Modo PWM semi-rama: RC2/P1A y RD5/P1B moduladas con tiempo muerto; RD6/P1C y RD7/P1D líneas de E/S
1	1	Modo PWM puente completo con salida inversa: RD5/P1B modulada; RD6/P1C activa; RD6/P1C y RD7/P1D inactivas

– **DC1B1..DC1B0:** Bits menos significativos (bit 0 y bit 1) del ciclo de trabajo en la señal PWM (solo en modo PWM y en modo PWM mejorado)



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

#### REGISTRO CPP1CON

– **CCP1M3.. CCP1M0: Bits de selección del modo del módulo ECCP:**

CCP1M3	CCP1M2	CCP1M1	CCP1M0	Modo
0	0	0	0	Módulo ECCP deshabilitado
0	0	0	1	Reservado
0	0	1	0	Modo Comparación, coincidencia => complemento de la RC2 y CCP1IF a '1'
0	0	1	1	Reservado
0	1	0	0	Modo Captura, cada flanco de bajada
0	1	0	1	Modo Captura, cada flanco de subida
0	1	1	0	Modo Captura, cada 4 flancos de subida
0	1	1	1	Modo Captura, cada 16 flancos de subida
1	0	0	0	Modo Comparación, RC2 inicialmente a '0', coincidencia => puesta a '1' y CCP1IF a '1'
1	0	0	1	Modo Comparación, RC2 inicialmente a '1', coincidencia => puesta a '0' y CCP1IF a '1'
1	0	1	0	Modo Comparación, coincidencia => CCP1IF a '1' (no hay cambio en RC2)
1	0	1	1	Modo Comparación, coincidencia => reset del Temporizador asociado y CCP1IF a '1'
1	1	0	0	Modo PWM: RC2/P1A, RD6/P1C activos a nivel alto; RD5/P1B, RD7/P1D activos a nivel alto
1	1	0	1	Modo PWM: RC2/P1A, RD6/P1C activos a nivel alto; RD5/P1B, RD7/P1D activos a nivel bajo
1	1	1	0	Modo PWM: RC2/P1A, RD6/P1C activos a nivel bajo; RD5/P1B, RD7/P1D activos a nivel alto
1	1	1	1	Modo PWM: RC2/P1A, RD6/P1C activos a nivel bajo; RD5/P1B, RD7/P1D activos a nivel bajo

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### REGISTRO ECPP1DEL

	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
ECCP1DEL	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0

- **PRSEN**: Bit de habilitación del reinicio automático del PWM (modo PWM mejorado):
  - \* Si PRSEN='0': Después de un Auto-Shutdown se debe poner a '0' el bit ECCPASE por software para reiniciar el PWM
  - \* Si PRSEN='1': Después de un Auto-Shutdown, cuando la condición que ha generado el Auto-Shutdown desaparece el bit ECCPASE se pone a '1' y el PWM se reinicia de forma automática.
- **PDC6..PDC0**: Bits de configuración del tiempo muerto en el modo PWM mejorado para semipunte (P1M1..P1M0="10"). El valor del tiempo muerto es:  
$$\text{Tiempo muerto} = [\text{PDC6}..\text{PDC0}] * 4 * T_{\text{OSC}}$$



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

#### REGISTRO ECPP1AS

	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0	L/E-0
ECCP1AS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1 PSSBD0

- **ECCPASE:** Bit de status del Auto-Shutdown (modo PWM mejorado):
  - \* Si ECCPASE='0': Las salidas PWM del ECCP están operando normalmente
  - \* Si ECCPASE='1': Se ha producido una condición de Shut-Down (desactivación automática de los pines de salida) y las salidas PWM del ECCP están en estado de shutdown
- **ECCPAS2, ECCPAS0:** Bits de selección de la fuente de Auto-Shutdown:

ECPPAS2	ECPPAS1	ECPPAS0	Fuente del Auto-Shutdown
0	0	0	Auto-Shutdown desactivado
0	0	1	Comparador analógico 1
0	1	0	Comparador analógico 2
0	1	1	Comparador 1 o Comparador 2
1	0	0	Pin RB0/FLT0
1	0	1	Pin RB0/FLT0 o Comparador 1
1	1	0	Pin RB0/FLT0 o Comparador 2
1	1	1	Pin RB0/FLT0 o Comparador 1 o Comparador 2

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

#### REGISTRO ECPP1AS

- **PSSAC1.. PSSAC0:** Bits de configuración del los pines RC2/P1A y RD6/P1C en modo shutdown:

PSSAC1	PSSAC0	Auto-Shutdown para los pines RC2/P1A y RD6/P1C
0	0	Se ponen RC2/P1A y RD6/P1C a '0'
0	1	Se ponen RC2/P1A y RD6/P1C a '1'
1	0	Se ponen RC2/P1A y RD6/P1C en modo de alta impedancia (tri-estado)
1	1	

- **PSSBD1.. PSSBD0:** Bits de configuración del los pines RD5/P1B y RD7/P1D en modo shutdown:

PSSBD1	PSSBD0	Auto-Shutdown para los pines RR5/P1B y RD7/P1D
0	0	Se ponen RD5/P1B y RD7/P1D a '0'
0	1	Se ponen RD5/P1B y RD7/P1D a '1'
1	0	Se ponen RD5/P1B y RD7/P1D en modo de alta impedancia (tri-estado)
1	1	

##### Modo de Captura:

En el modo de captura el valor del Temporizador 1 o del Temporizador 3 pasa al par de registros CCPR1H;CCPR1L cuando se produce alguno de los siguientes eventos en el pin del ECCP (RC2/CCP1):

- En cada flanco de bajada (CCP1M3..CCP1M0="0100")
- En cada flanco de subida (CCP1M3..CCP1M0="0101")
- Cada 4 flancos de subida (CCP1M3..CCP1M0="0110")
- Cada 16 flancos de subida (CCP1M3..CCP1M0="0111")

La selección del Temporizador que se captura se realiza mediante los bits T3CCP2 y T3CCP1 del registro T3CON:

- Si T3CCP2='0' el temporizador capturado será el Temporizador 1
- Si T3CCP2='1' el temporizador capturado será el Temporizador 3

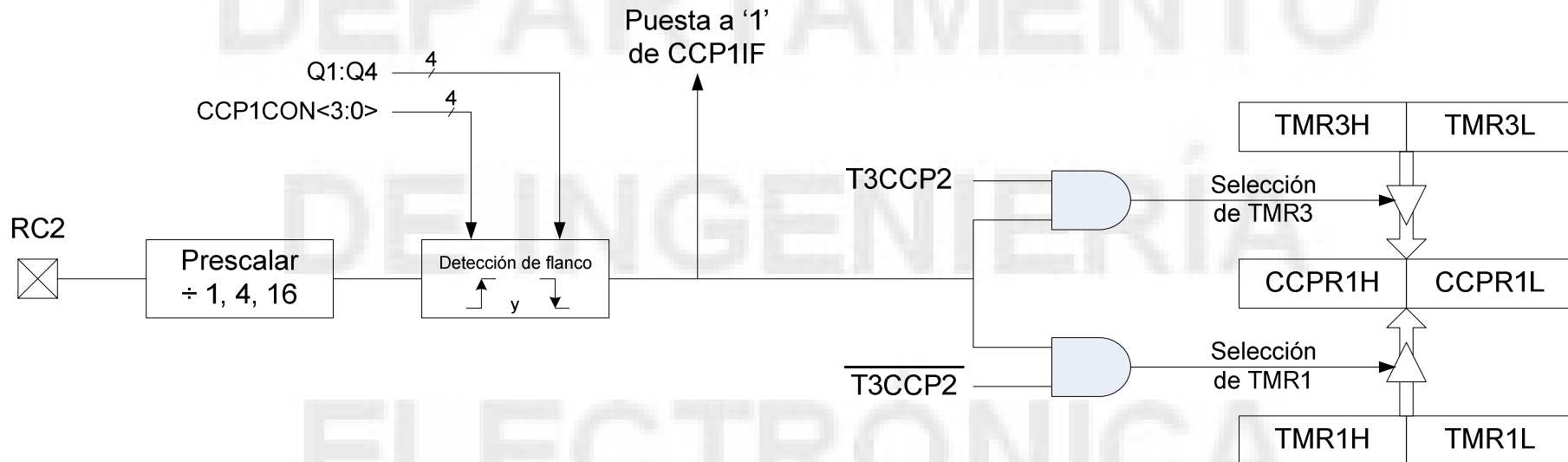
El temporizador que se utilice debe configurarse en modo Temporizador o en modo contador síncrono. Si se configura en modo contador asíncrono el modo captura no funcionará.

Cuando la condición de captura se cumple el flag de interrupción CPP1IF se pone a '1'.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### Diagrama de bloques del modo de Captura:



**Cuando se realiza un cambio de modo de captura el valor del prescalador no se inicializa, además el bit CCP1IF puede ponerse a '1' de forma no controlada. Para evitar esta falsa interrupción y resetear el prescalador, antes de cambiar de modo de captura, debe desactivarse el modo captura (CCP1CON=0x00) y a continuación establecer el nuevo modo de captura.**

##### Modo de Comparación:

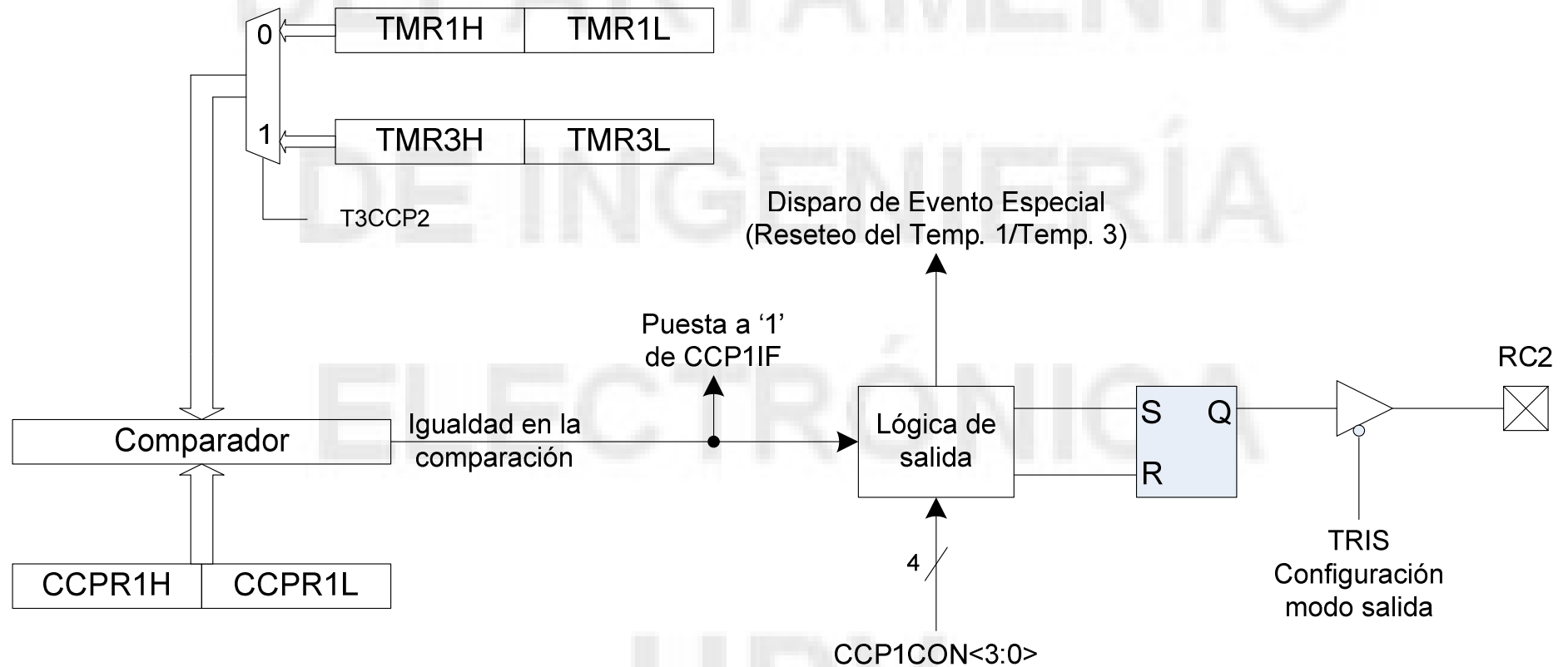
En el modo de comparación el par de registros [CCPR1H;CCPR1L] se comparan continuamente con el Temporizador asociado (Temporizador 1 o 3). Existen varios modos de comparación en función del valor de los bits: CCP1M3..CCP1M0:

- CCP1M3..CCP1M0="0010": El pin de salida RC2/CCP1 se inicializa a '0'. Cada vez que se produce la igualdad entre [CCPR1H;CCPR1L] y el Temporizador asociado se complementa el pin de salida y se pone a '1' el flag CCP1IF.
- CCP1M3..CCP1M0="1000": El pin de salida RC2/CCP1 se inicializa a '0'. Cuando se produce la igualdad entre [CCPR1H;CCPR1L] y el Temporizador asociado el pin de salida se pone a '1' y se pone a '1' el flag CCP1IF.
- CCP1M3..CCP1M0="1001": El pin de salida del RC2/CCP1 se inicializa a '1'. Cuando se produce la igualdad entre [CCPR1H;CCPR1L] y el Temporizador asociado el pin de salida se pone a '0' y se pone a '1' el flag CCP1IF.
- CCP1M3..CCP1M0="1010": El pin de salida del RC2/CCP1 toma el valor del latch correspondiente. Cuando se produce la igualdad entre [CCPR1H;CCPR1L] y el Temporizador asociado se pone a '1' el flag CCP1IF.
- CCP1M3..CCP1M0="1011": El pin de salida del RC2/CCP1 toma el valor del latch correspondiente. Cuando se produce la igualdad entre [CCPR1H;CCPR1L] y el Temporizador asociado **se inicializa a 0 el Temporizador asociado** y se pone a '1' el flag CCP1IF.

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### Diagrama de bloques del modo de Comparación:





#### EJERCICIO:

Escribir en código de un programa para el uC PIC18F4550 que genere por el pin RC2 una señal PWM utilizando el modo de comparación del ECCP. El ciclo de trabajo de la señal expresado en tanto por ciento (%) debe establecerse mediante una constante: DUTY. ( $F_{OSC}=20\text{MHz}$ ).

```
#include <p18F4550.h> // Declaración de librerías
const int DUTY=50;    // Valor del duty de la señal generada (en %)
int Ton;              // Variable correspondiente a la duracion del estado alto

void R_Int_Alta(void); // Declaración rutina de tratam. de interrupciones de alta prioridad

// Vectorización de interrupciones de alta prioridad
#pragma code Vector_Int_Alta=0x08
void Int_Alta (void)
{
    _asm GOTO R_Int_Alta _endasm // Se salta a la rutina de tratamiento de
                                // interrupciones de alta prioridad
}
#pragma code
```

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### EJERCICIO:

```
// Rutina de tratamiento de interrupciones de alta prioridad
#pragma interrupt R_Int_Alta
void R_Int_Alta (void)
{
    if (PIR1bits.CCP1IF)           // Se comprueba si se ha producido interrupción del ECCP
    {                               // Si se ha producido interrupción del ECCP:
        PIR1bits.CCP1IF=0;        // Se borra el flag de interrupción del ECCP
        if (CCPR1==0)             // Se comprueba el valor anterior de CCPR1
        {                         // Si CCPR1=0:
            CCPR1=65536 - Ton;     // Se carga la temporización del nivel bajo:
            // desde 0 hasta 65536-Ton
        }
        else
        {                         // Si CCPR1<>0:
            CCPR1= 0;              // Se carga la temporización de nivel alto:
            // desde 65536-Ton hasta el overflow (es decir 0)
        }
    }
}
```



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### EJERCICIO:

// Programa Principal

void main (void)

{

TRISC=0xFB;       // Se inicializa RC2 como salida  
T3CON=0x41;       // Se configura el temporizador 3 como temporizador,  
                  // sin prescalar, TMR3 fuente de ECCP y se pone en marcha

Ton=65536/100;     // Se calcula el valor correspondiente a la duración del  
Ton=Ton\*DUTY;     // nivel alto: Ton= (65536\*DUTY)/100

CCPR1=65536-Ton;   // Se carga en CCPR1 la duración del nivel bajo  
CCP1CON=0x0A;     // Se configura en ECCP en modo de comparación con  
                  // complemento de RC2 en cada coincidencia entre  
                  // CCPR2 y TMR3 (RC1 empieza con nivel bajo)

INTCONbits.PEIE=1; // Se habilitan a nivel global las interrupciones de periféricos

INTCONbits.GIE=1; // Se habilitan a nivel global todas las interrupciones

PIE1bits.CCP1IE=1; // Se habilita la interrupción del ECCP

while (1)

{  
}

}



##### Modo PWM estándar:

En el modo PWM estándar del módulo ECCP permite generar una señal PWM de ciclo de trabajo y frecuencia programables.

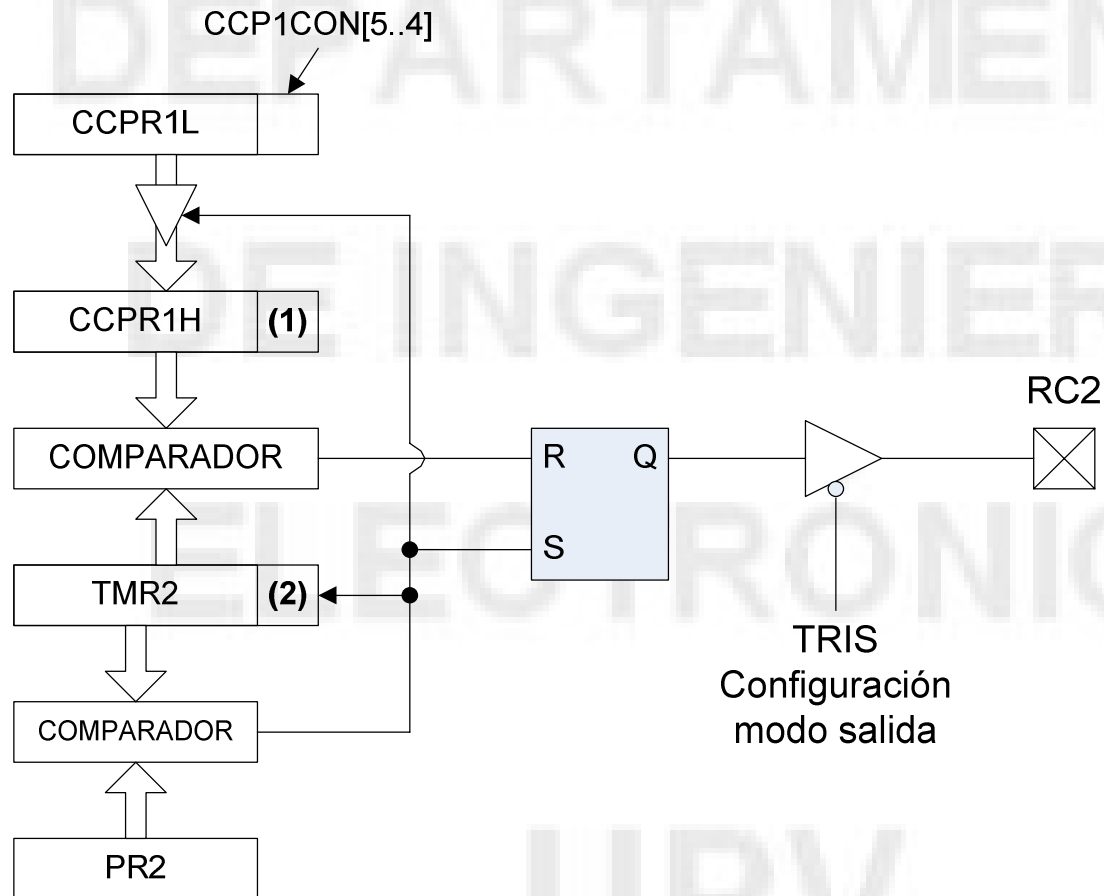
El funcionamiento en este modo es el siguiente:

- Un registro de 10 bits (compuesto por CCPR1H y por un registro interno de 2 bits) se compara constantemente con un contador de 10 bits (compuesto por el registro TMR2 del Temporizador 2 y un prescalar de 2 bits).
- Cuando el valor del contador se hace igual al del registro se pone a '0' el pin de salida RC2/CCP1.
- En paralelo el registro PR2 se compara constantemente con el registro TMR2 del Temporizador 2. Cuando el valor de TMR2 alcanza PR2:
  - Se pone a '1' el pin de salida RC2/CCP1
  - Se inicializa a 0 el valor del contador de 10 bits (registro TMR2 y el prescalar de 2 bits)
  - El valor del registro de 10 bits (registro CCPR1H y registro interno de 2 bits) se recarga con el valor del registro CPPR1L y los bits CCP1CON[5..4].

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### Diagrama de bloques del modo PWM estándar:



(1) Registro interno de 2 bits que permite ampliar el registro CCPR1H a 10 bits

(2) Prescalar de 2 bits que permite obtener una base de tiempos de 10 bits

##### Modo PWM estándar:

Para generar la señal PWM se deben seguir los siguientes pasos:

- Configurar el pin RC2/CCP1 como salida
- Configurar el Temporizador 2 para que trabaje como temporizador. Puede utilizarse el prescalar del Temporizador 2, pero no puede utilizarse el post-escalar.
- Configurar el módulo ECCP en modo PWM estándar:
  - CCP1M3..CCP1M0: “1100”
  - P1M1..P1M0= “00”
- Establecer el periodo de la señal PWM mediante el valor del registro PR2. El periodo de la señal PWM será:

$$\text{PERIODO } (T_{\text{PWM}}) = \frac{(PR2 + 1) \cdot 4 \cdot (\text{Prescalar del Temporizador 2})}{F_{\text{OSC}}}$$

- Establecer el ciclo de trabajo de la señal mediante el registro CPPR1L y los bits CCP1CON[5..4]. El ciclo de trabajo de la señal será:

$$\text{CICLO DE TRABAJO} = \frac{([CCPR1L; CCP1CON[5..4]]) \cdot (\text{Prescalar del Temporizador 2})}{F_{\text{OSC}} * T_{\text{PWM}}}$$

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### EJERCICIO:

Escribir en código de un programa para el uC PIC18F4550 que genere por el pin RC2 una señal PWM de 2kHz de frecuencia utilizando el modo PWM estándar del ECCP. El ciclo de trabajo de la señal expresado en tanto por ciento (%) debe establecerse mediante una constante: DUTY ( $F_{osc}=20\text{MHz}$ ).

```
#include <p18F4550.h> // Declaración de librerías
const int DUTY=75;           // Valor del duty de la señal generada (en %)
unsigned short long CTon;    // Variable correspondiente a la duración del estado alto
void main (void)
{
    TRISC=0xFB;           // Se inicializa RC2 como salida
    T2CON=0x07;           // Se configura el Temp. con el prescalar a 1:16
                           // y el post-escalar a 1:1 y se pone en marcha
    CCP1CON=0x0C;         // Se configura en ECCP en modo de PWM estándar
    PR2=155;              // Se fija el periodo  $T_{pwm}=0,5\text{ms}$  ( $F_{pwm}=2\text{kHz}$ )
                           //  $T_{pwm} = ((PR2+1)*4*Prescaler)/F_{osc}$ 
                           //  $\Rightarrow PR2 = [(F_{osc}*T_{pwm})/(Prescaler*4)]-1 = [(20E6*0,5E-3)/(16*4)] = 155$ 
    CTon=DUTY;            // Se calcula el valor de cuenta para la duracion del nivel alto
    CTon=625*CTon;        //  $DUTY(\%) = (CTon*Prescaler*100)/(F_{osc}*T_{pwm})$ 
    CTon=CTon/100;        //  $\Rightarrow CTon = (DUTY*F_{osc}*T_{pwm})/(Prescaler*100) = (DUTY*625)/100$ 
    CCPR1L=CTon/4;        // Se carga en CCPR2 la duración del nivel alto (8 bits mas significativos)
    CCP1CONbits.DC1B0=CTon&0x0001; // Se cargan los 2 bits menos significativos de la
    CCP1CONbits.DC1B1=CTon&0x0002; // duración del nivel alto en DC1B0 y DC1B1
    while (1);
}
```



##### Modo PWM mejorado:

El modo PWM mejorado del módulo ECCP permite generar señales PWM por los pines RC2/P1A, RD5/P1B, RD6/P1C y RD7/P1D con una gran flexibilidad. Algunas de las posibilidades de este modo son:

- Generación de 1 salida PWM (pin RC2/P1A): **modo PWM estándar**
- Generación de 2 salidas PWM complementarias (pines RC2/P1A y RD5/P1B): **modo semipunte**
- Generación de 4 salidas (pines RC2/P1A, RD5/P1B, RD6/P1C y RD7/P1D) con posibilidad de realizar modulación PWM en dos de ellas : **modo puente completo**
- Flexibilidad en la configuración de los niveles activos de las señales PWM
- Programación de tiempos muertos entre las transiciones de las señales de salida
- Programación de condiciones de Auto-Shutdown y de reinicio automático una vez desaparece la condición de Shut-down

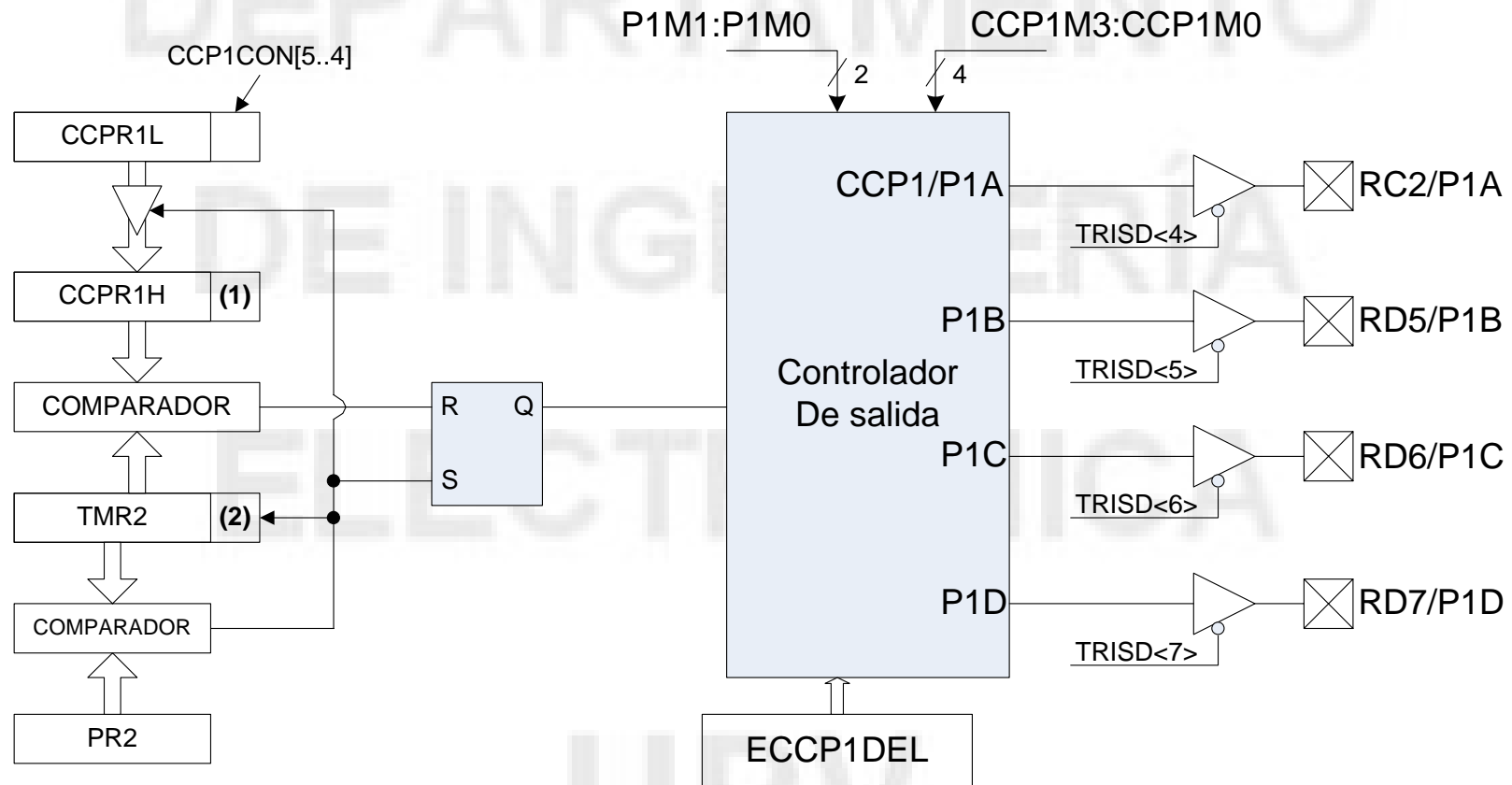
El modo PWM mejorado se puede utilizar para la generación de las señales de disparo de la etapa de potencia de convertidores DC/DC (FAC's, control unidireccional y bidireccional de motores DC) y AC/AC (inversores, control de motores AC).



# TEMA 3: MICROCONTROLADOR PIC18F4550

## UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

### Diagrama de bloques del modo PWM mejorado:



- (1) Registro interno de 2 bits que permite ampliar el registro CCPR1H a 10 bits  
(2) Prescaler de 2 bits que permite obtener una base de tiempos de 10 bits

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### Configuración de las líneas de salida en el modo PWM mejorado:

El modo PWM mejorado puede controlar hasta 4 líneas de salida en función del modo de trabajo (registro CCP1CON):

Modo ECCP	Configuración CCP1CON	RC2/P1A	RD5/P1B	RD6/P1C	RD7/P1D
PWM Estándar	00xx11xx	Pin ECCP	RD5	RD6	RD7
PWM Semi-puente	10xx11xx	P1A	P1B	RD6	RD7
PWM Puente completo	x1xx11xx	P1A	P1B	P1C	P1D

Cuando alguna de estas líneas trabaja asociada al módulo ECCP, no está disponible en ninguna de sus otras funciones (líneas de E/S u otras funciones secundarias). Para que las líneas RC2/P1A, RD5/P1B, RD6/P1C y RD7/P1D puedan trabajar en modo PWM mejorado es necesario configurarlas como líneas de salida (registros TRISC y TRISD).

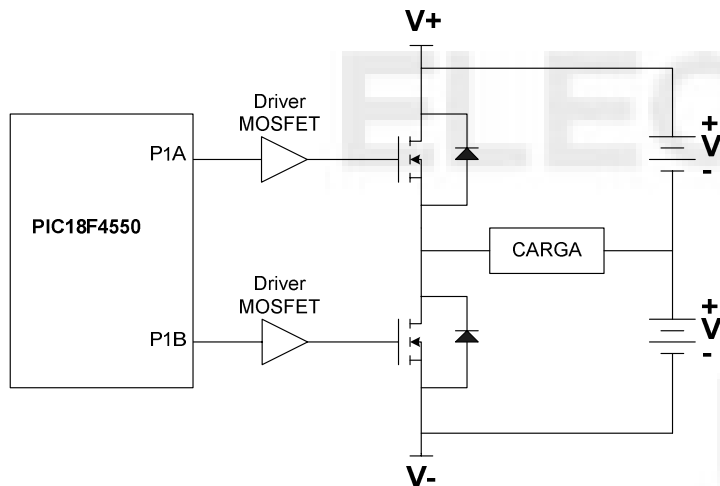
### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

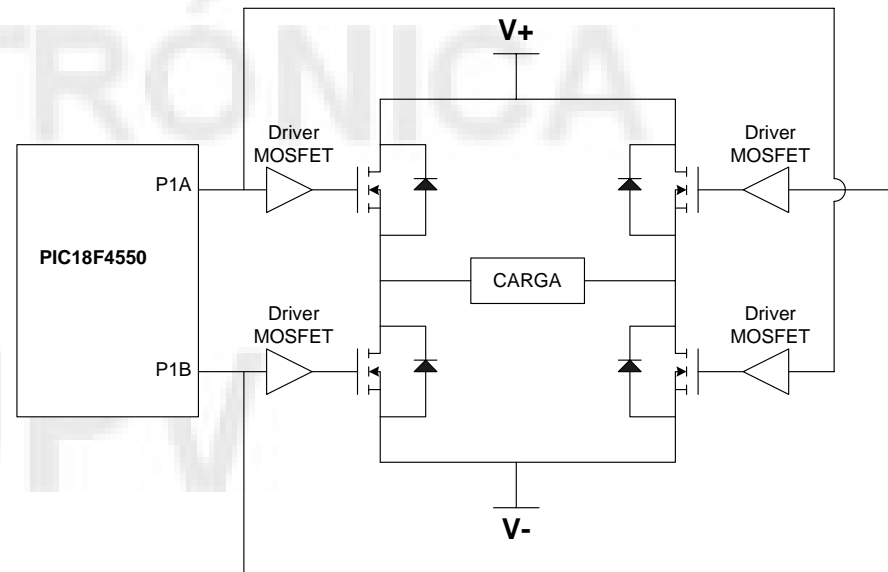
##### Modo PWM en Semi-puente:

En el modo Semi-puente el ECCP genera dos señales PWM complementarias por los pines RC2/P1A y RD5/P1B. Las señales PWM pueden configurarse para que sean activas por nivel alto o por nivel bajo. Se puede programar un tiempo muerto entre las transiciones de las señales con el fin de evitar cortocircuitos en la alimentación de la etapa de potencia. Este modo se utiliza para controlar etapas de potencia en semi-puente o una etapa en puente completo donde los 4 transistores se modulan mediante señales PWM.

CIRCUITO EN SEMI-PUENTE ESTÁNDAR



SALIDA EN SEMI-PUENTE CONTROLANDO UN PUENTE COMPLETO



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

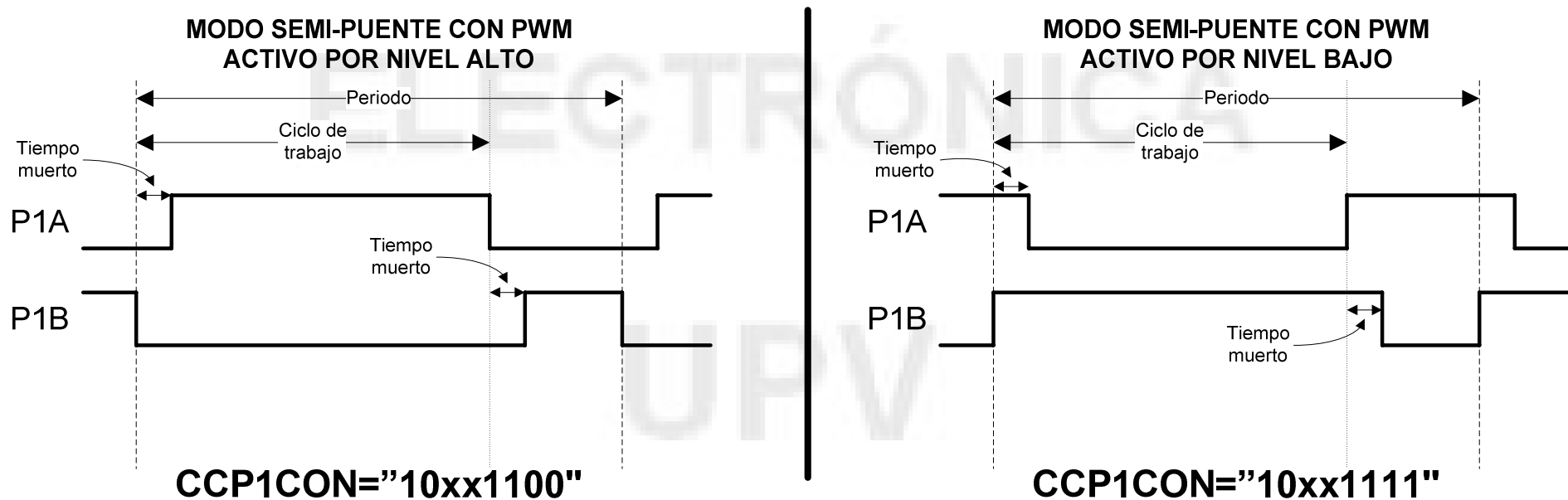
##### Modo PWM en Semi-puente: Señales generadas

Las señales PWM generadas en este modo son complementarias y se puede elegir si se quiere que sean activas por nivel alto o activas por nivel bajo (bits CCP1M3..CCP1M0).

También se puede introducir un tiempo muerto entre las transiciones de P1A y P1B.

El valor del tiempo muerto depende de los bits PCD6..PCD0:

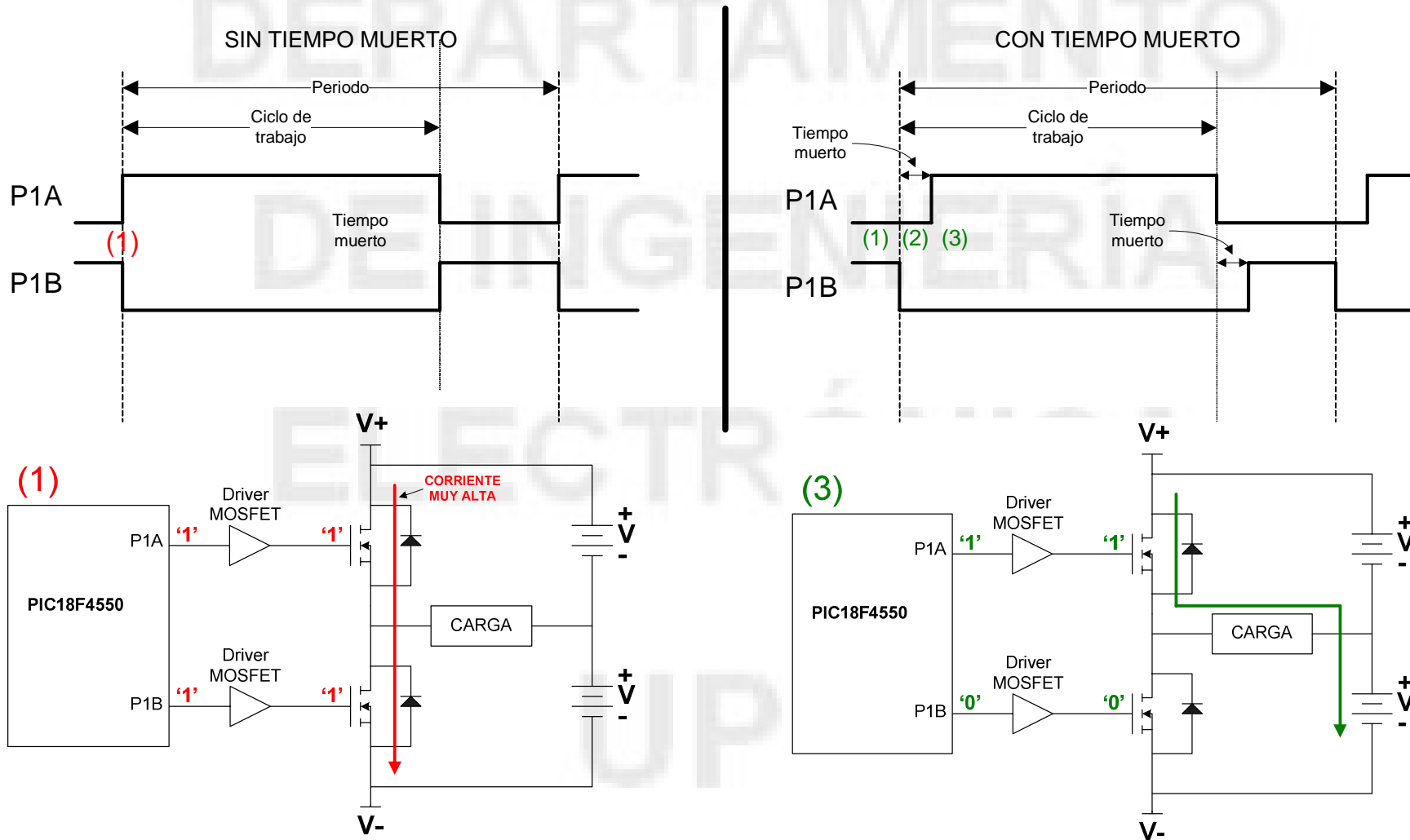
$$\text{Tiempo muerto} = [\text{PDC6}..\text{PDC0}] * 4 * \text{TOSC}$$



## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

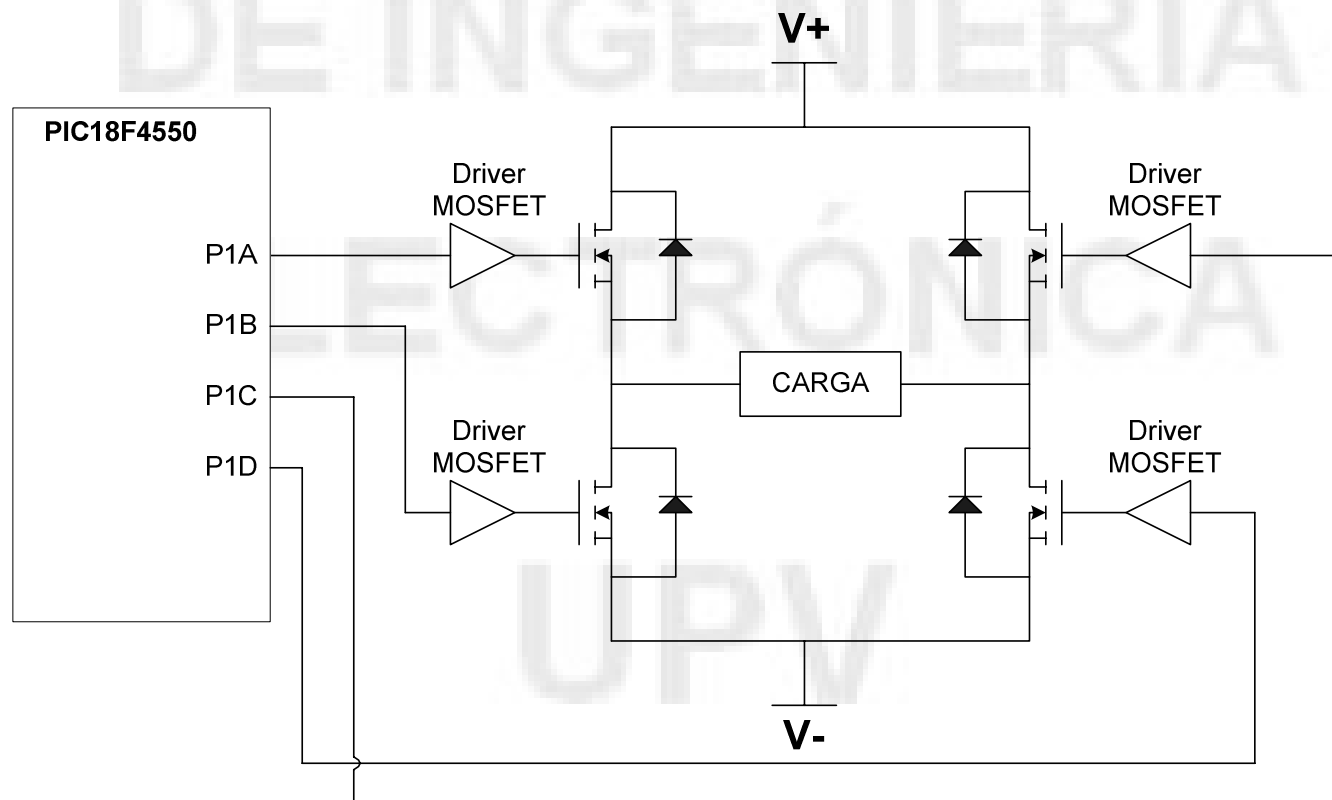
#### Modo PWM en Semi-puente: Tiempo muerto



##### Modo PWM en Puente completo:

En el modo Puente completo el ECCP genera las 4 señales de disparo de una etapa de potencia en puente completo por los pines RC2/P1A, RD5/P1B, RD6/P1C y RD7/P1D.

##### CIRCUITO EN PUENTE COMPLETO



### TEMA 3: MICROCONTROLADOR PIC18F4550

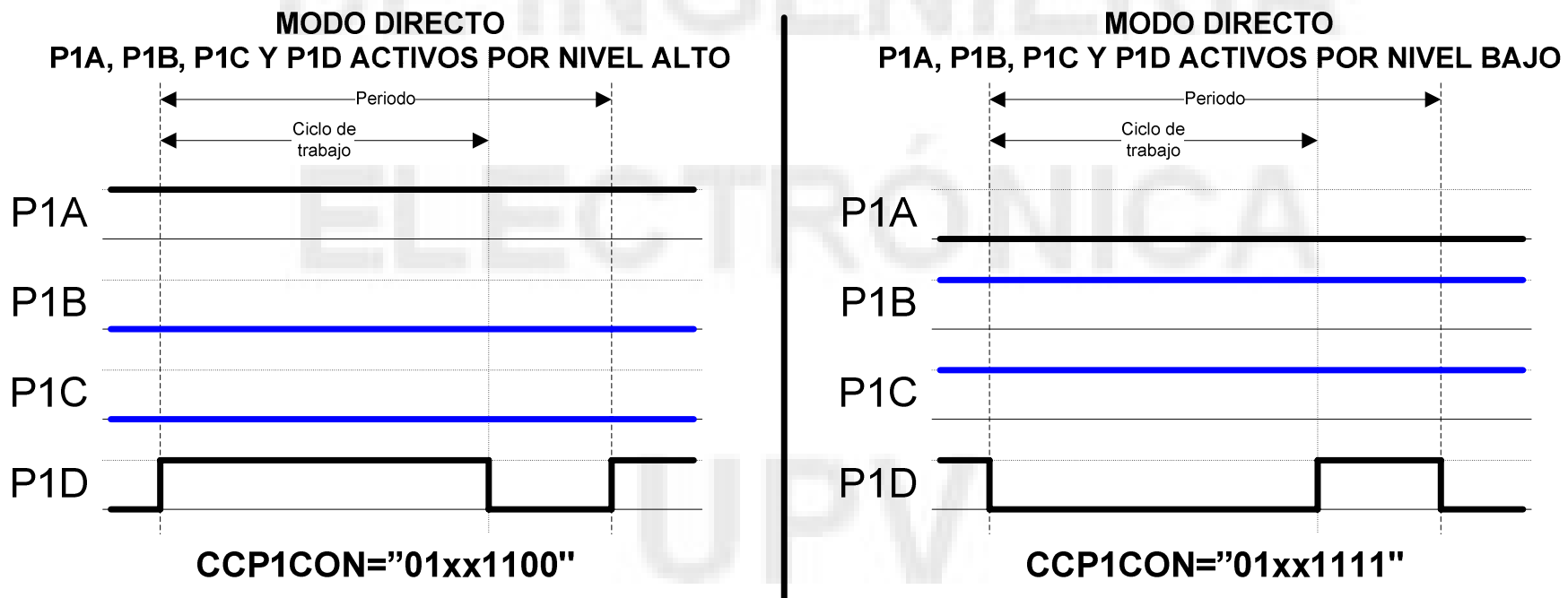
#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### Modo PWM en Puente completo: Señales generadas

Las señales generadas pueden ser en modo directo o en modo inverso:

- Modo directo (P1M1..P1M0="01"):

En este modo las señales pueden ser activas por nivel alto o activas por nivel bajo en función de los bits CCP1M3..CCP1M0.



**NOTA:** Las señales **no activas** aparecen en **azul**

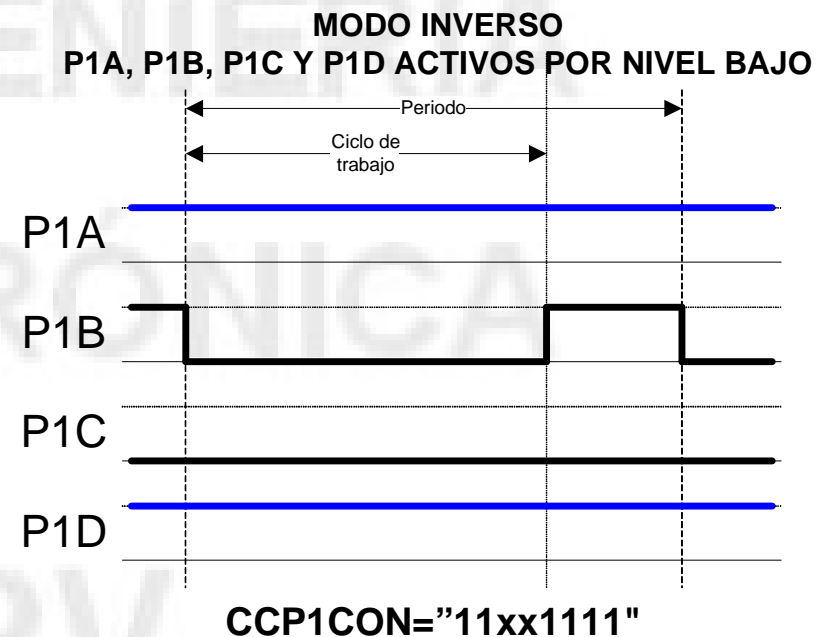
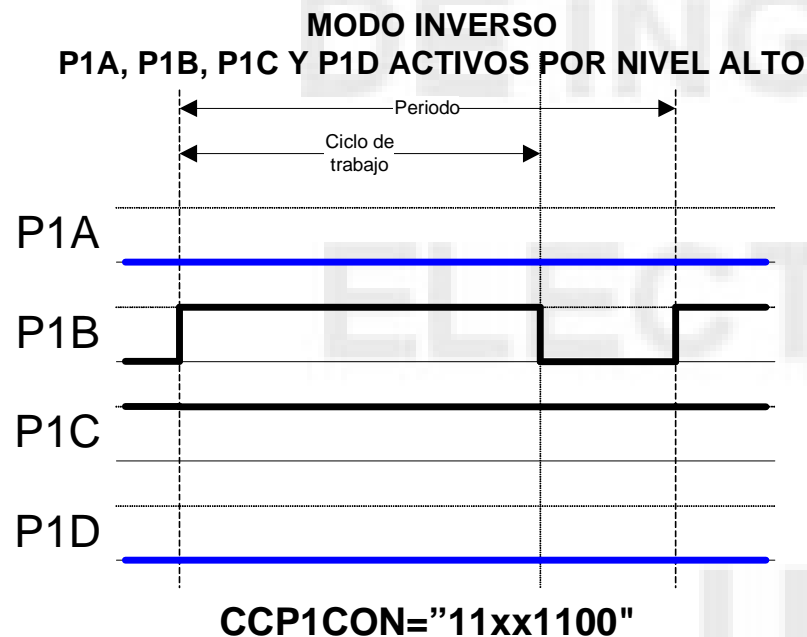
### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### Modo PWM en Puente completo: Señales generadas

- Modo inverso ( $P1M1..P1M0="11"$ ):

En este modo las señales pueden ser activas por nivel alto o activas por nivel bajo en función de los bits  $CCP1M3..CCP1M0$ .



**NOTA:** Las señales **no activas** aparecen en **azul**



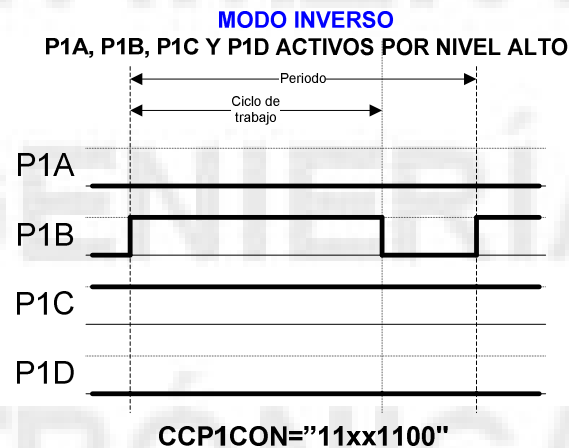
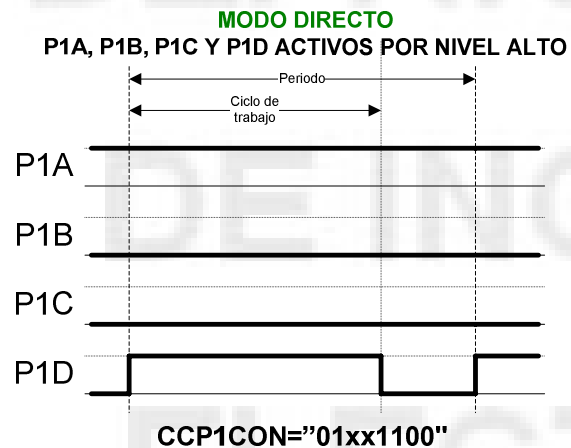
### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

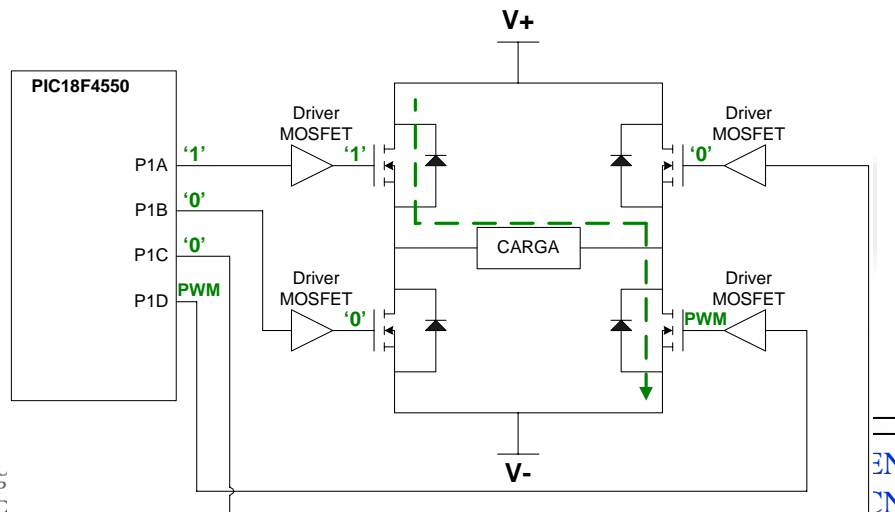
##### Modo PWM en Puente completo: Cambio de sentido

Para cambiar el sentido en el Puente completo basta con cambiar el valor de los bits

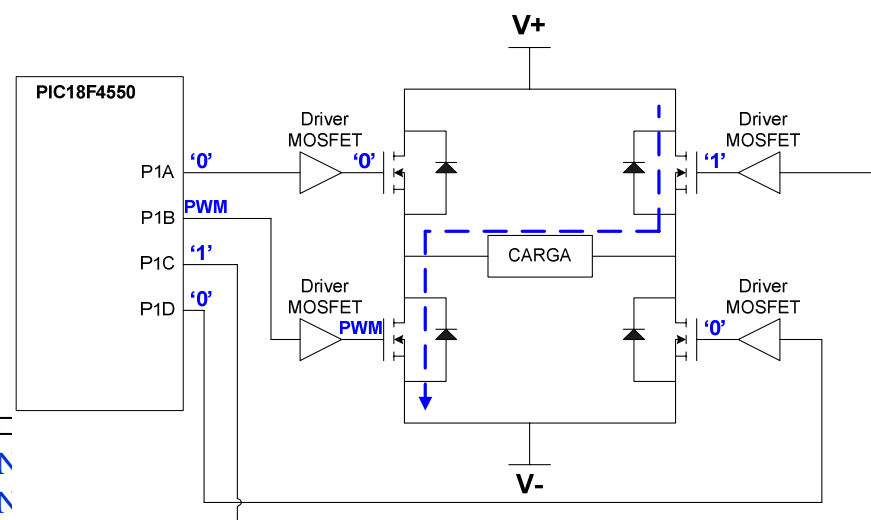
P1M1..P1M0: Sentido directo: **P1M1..P1M0="01"** -> Sentido inverso: **P1M1..P1M0="11"**



**CIRCUITO EN PUENTE COMPLETO  
SENTIDO DIRECTO**



**CIRCUITO EN PUENTE COMPLETO  
SENTIDO INVERSO**



### TEMA 3: MICROCONTROLADOR PIC18F4550

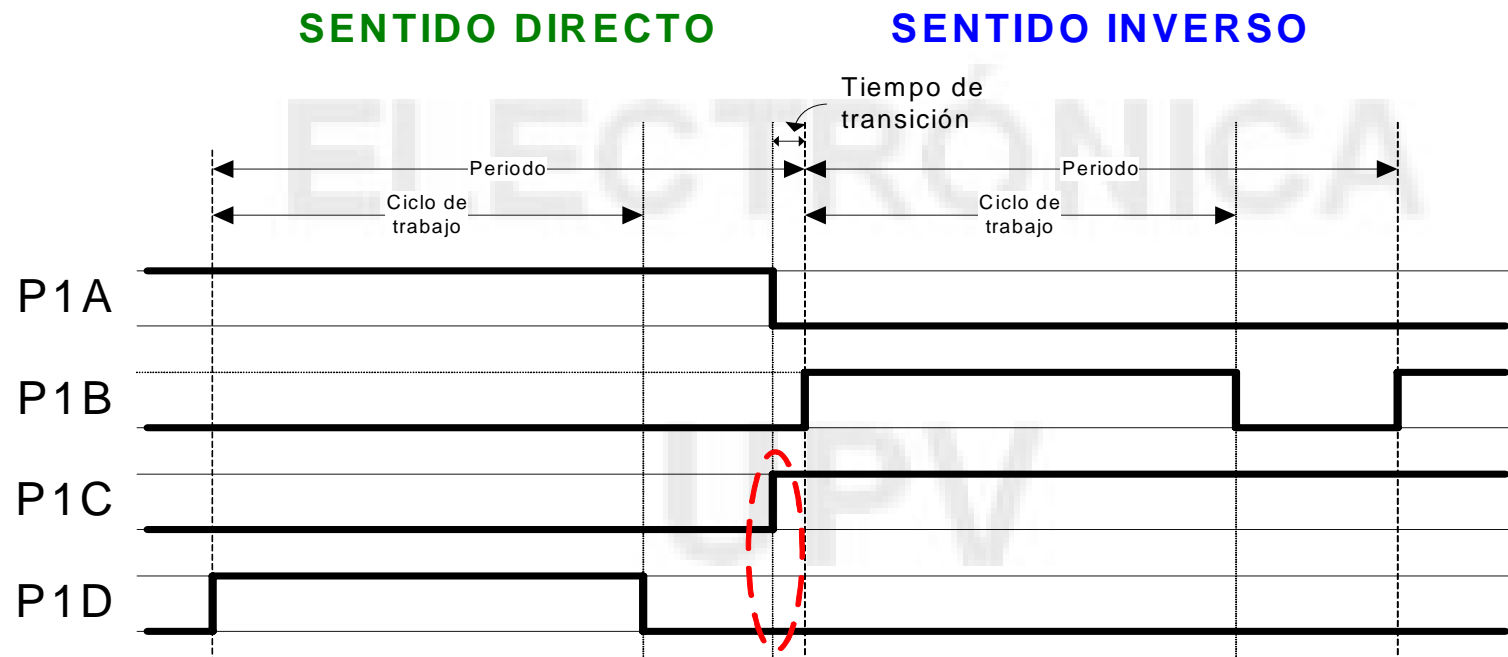
#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### Modo PWM en Puente completo: Transición en el cambio de sentido

Cuando se realiza un cambio de sentido (cambio en P1M1), el ECCP establece el nuevo sentido en el siguiente ciclo PWM. Justo antes de finalizar el último ciclo previo al cambio de sentido, se desactivan las señales moduladas (P1B o P1D) y las salidas no moduladas (P1A o P1C) se ponen en la configuración correspondiente al nuevo sentido. Esto ocurre un intervalo de tiempo antes del inicio del siguiente ciclo PWM:

Intervalo de tiempo de transición =  $4 * T_{OSC} * (\text{Prescaler Temp. 2})$

Gracias a este tiempo de transición se evitan posibles cortocircuitos en las semi-ramas del puente durante la transición .



##### Modo PWM en Puente completo: Limitaciones

A pesar del intervalo de transición, en determinadas condiciones de funcionamiento se puede producir un cortocircuito en algunas de las semi-ramas del puente completo.

Si el **cambio de sentido** se produce **cuando el ciclo de trabajo** del PWM está **cerca del 100%** y los transistores utilizados en el semipuerto tienen un **tiempo de conmutación a OFF superior al tiempo de conmutación a ON** se producirá un **cortocircuito** en las semiramas del puente.

Para evitar este problema caben varias soluciones:

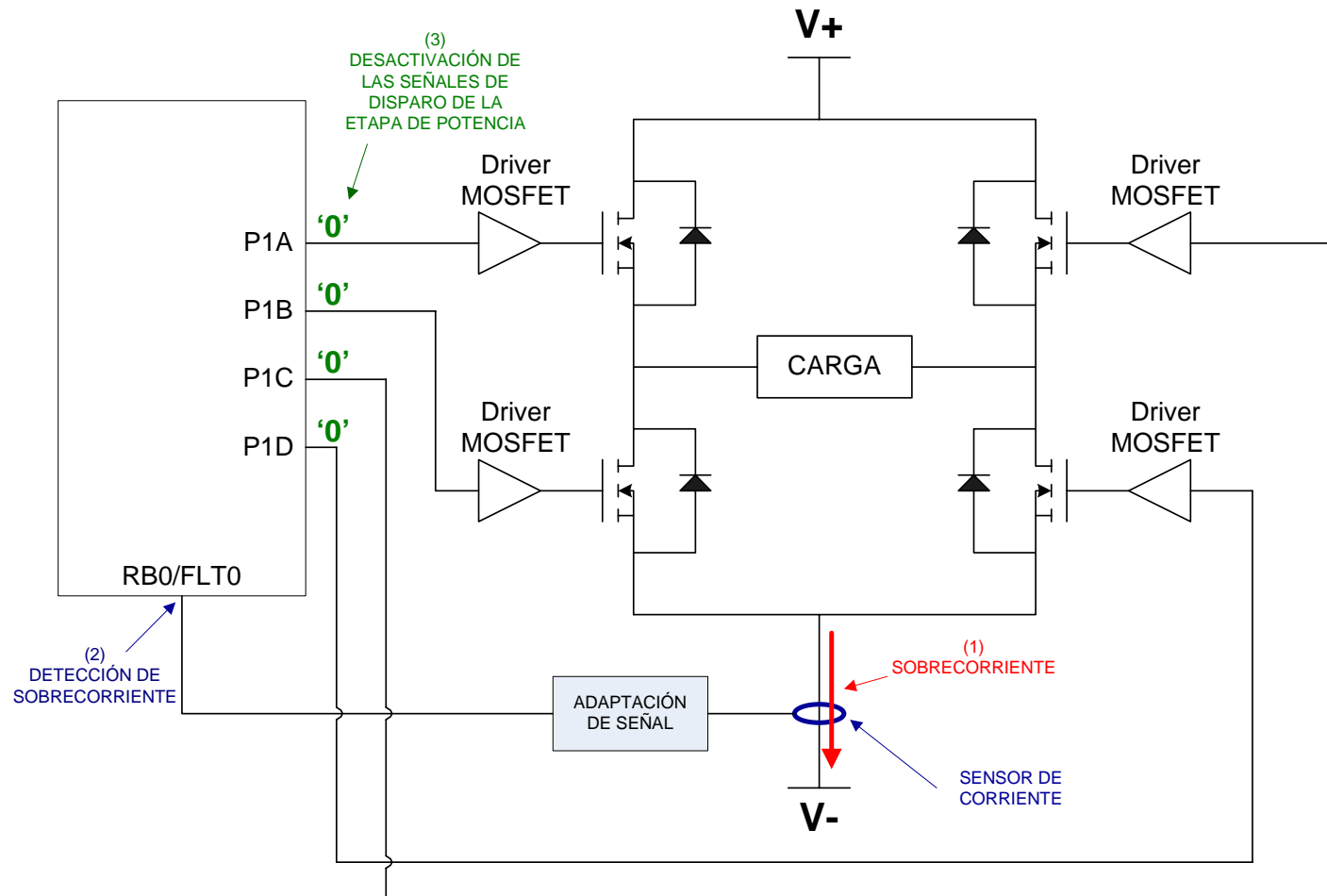
- Reducir el ciclo de trabajo ligeramente antes de realizar el cambio de sentido
- Utilizar transistores o circuitos de disparo de los transistores que permitan que la conmutación a OFF sea más rápida que la conmutación a ON

# TEMA 3: MICROCONTROLADOR PIC18F4550

## UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

### Desactivación automática del ECCP

#### DETECCIÓN DE SOBRECORRIENTE DESACTIVACIÓN DE LAS SEÑALES DE DISPARO DE LA ETAPA DE POTENCIA



##### Desactivación automática del ECCP

Se puede programar el módulo ECCP para que desactive automáticamente sus salidas (RC2/CCP1/P1A, RD5/P1B, RD6/P1C y RD7/P1D) en los siguientes supuestos:

- Se detecta un nivel bajo en la línea RB0/FLT0 (ECCPAS2..ECCPAS0=“100”)
- Se activa la salida del Comparador 1 (ECCPAS2..ECCPAS0=“001”)
- Se activa la salida del Comparador 2 (ECCPAS2..ECCPAS0=“010”)
- Cualquier combinación de los 3 supuestos anteriores:
  - ECCPAS2..ECCPAS0=“111”: Nivel bajo en RB0/FLT0 o activación salida Comparador 1 o activación salida Comparador 2
  - ECCPAS2..ECCPAS0=“110”: Nivel bajo en RB0/FLT0 o activación salida Comparador 2
  - ECCPAS2..ECCPAS0=“101”: Nivel bajo en RB0/FLT0 o activación salida Comparador 1
  - ECCPAS2..ECCPAS0=“111”: Activación salida Comparador 1 o activación salida Comparador 2

El nivel que toman las líneas RC2/CCP1/P1A, RD5/P1B, RD6/P1C y RD7/P1D cuando se produce la desactivación automática puede ser establecido mediante los bits PSSAC1..PSSAC0 y PSSBD1..PSSBD0 del registro ECCP1AS:

PSSAC1	PSSAC0	Desactivación automática De RC2/P1A y RD6/P1C
0	0	RC2/P1A y RD6/P1C a '0'
0	1	RC2/P1A y RD6/P1C a '1'
1	0	RC2/P1A y RD6/P1C en modo de alta impedancia
1	1	tri-estado)

PSSBD1	PSSBD0	Desactivación automática de RD5/P1B y RD7/P1D
0	0	RD5/P1B y RD7/P1D a '0'
0	1	RD5/P1B y RD7/P1D a '1'
1	0	RD5/P1B y RD7/P1D en modo de alta impedancia
1	1	(tri-estado)

##### Desactivación automática del ECCP (cont.)

Cuando se produce una condición de desactivación automática:

- Las salidas RC2/CCP1/P1A, RD5/P1B, RD6/P1C y RD7/P1D se ponen a los niveles programados en los bits PSSAC1..PSSAC0 y PSSBD1..PSSBD0
- El bit ECCPASE del registro ECCP1AS se pone a ‘1’

El bit ECCPASE puede ponerse a ‘1’ por software para provocar una desactivación manual. También puede ponerse el bit ECCPASE a ‘0’ por software cuando la condición de desactivación ha desaparecido para volver a activar las líneas RC2/CCP1/P1A, RD5/P1B, RD6/P1C y RD7/P1D. Sin embargo **ECCPASE no puede ponerse a ‘0’ mientras se mantenga la condición de desactivación.**

Se puede configurar el ECCP para que pueda auto-recuperarse del modo de desactivación automática cuando la condición de desactivación desaparece. Para ello hay que poner el bit PRSEN del registro ECCP1DEL a ‘1’. En este caso, cuando la condición de desactivación desaparezca el ECCP volverá a generar las señales por las líneas RC2/CCP1/P1A, RD5/P1B, RD6/P1C y RD7/P1D en el siguiente periodo PWM.

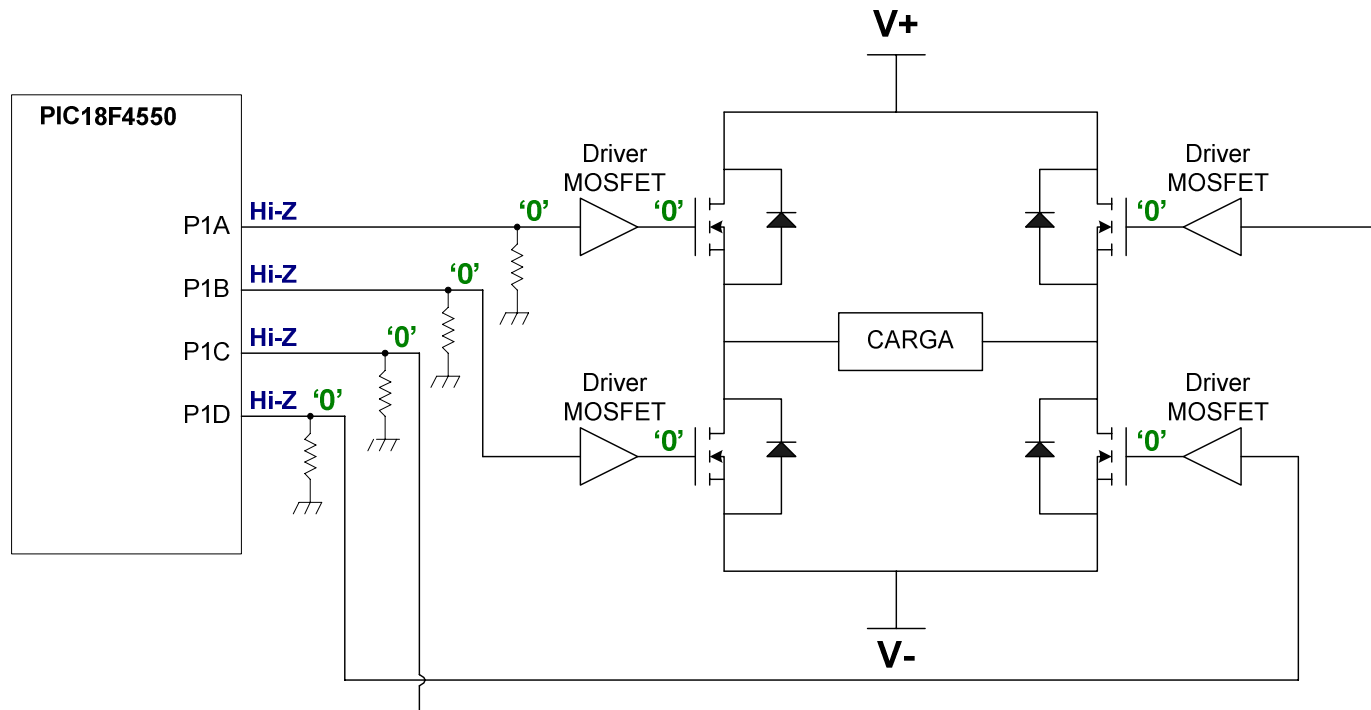
### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### Inicialización del sistema cuando se controlan etapas de potencia mediante el ECCP:

Cuando se inicia el sistema (después de un reset o al dar tensión de alimentación) las líneas RC2/CCP1/P1A, RD5/P1B, RD6/P1C y RD7/P1D quedan establecidas como entradas (alta impedancia) hasta que son configuradas apropiadamente por el programa. Para evitar corto-circuitos en la etapa de potencia en ese instante es conveniente conectar a dichas líneas resistencia de pull-up o pull-down (según convenga) de forma que se garantice que los transistores quedarán desactivados hasta que se inicialice completamente el sistema

##### SITUACIÓN AL INICIALIZAR EL SISTEMA



##### Procedimiento de inicialización del módulo ECCP en modo PWM mejorado:

1. Configurar las líneas que se vayan a utilizar (RC2/CCP1/P1A, RD5/P1B, RD6/P1C y/o RD7/P1D) **como entrada**.
2. Cargar el valor correspondiente al periodo de la señal en el registro PR2
3. Si se va a utilizar la desactivación automática:
  - Deshabilitar el modo de desactivación automática (ECCPAS2..ECCPAS0="000")
  - Configurar el periférico que funcionará como fuente de la desactivación automática (RB0/FLT0, Comparador 1 y/o Comparador 2 )
  - Espera a que **no se cumpla** la condición de autodesactivación
4. Configurar el módulo ECCP en el modo PWM deseado
  - Configurar la modalidad de salida (PWM simple, Semipuerto o Puente Completo) y la dirección (directo o inverso) mediante los bits PM1..PM0
  - Configurar la polaridad de las líneas utilizadas mediante los bits CPP1M3..CPP1M0
5. Establecer el ciclo de trabajo de la señal PWM mediante el registro CPP1RL y los bits CPP1CON[5..4]
6. Si se va a trabajar en modo de Semipuerto, configurar la duración del tiempo muerto mediante los bits PDC6..PDC0



##### Procedimiento de inicialización del módulo ECCP en modo PWM mejorado (cont.):

7. Si se va a utilizar la desactivación automática:
  - Seleccionar la fuente de desactivación mediante los bits ECCPAS2..ECCPAS0
  - Seleccionar los niveles de desactivación de las salidas RC2/CCP1/P1A, RD5/P1B, RD6/P1C y/o RD7/P1D mediante los bits PSSAC1..PSSAC0 y PSSBD1..PSSBD0
  - Poner a '1' el bit ECCPASE
8. Si se va a utilizar la autorecuperación de la desactivación automática, poner a '1' el bit PRSEN
9. Configurar y arrancar el Temporizador 2
  - Establecer el valor del prescalar
  - Activar el Temporizador 2
10. Habilitar las líneas de salida después de que haya transcurrido un periodo de la señal PWM
  - Esperar a que se produzca el desbordamiento del Temporizador 2 (TMR2IF='1')
  - Configurar las líneas que se vayan a utilizar (RC2/CCP1/P1A, RD5/P1B, RD6/P1C y/o RD7/P1D) **como salida**
  - Poner a '0' el bit ECCPASE

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### Interacciones entre el módulo ECCP y el módulo CCP:

**Dado que los módulos ECCP y CCP comparten los mismos Temporizadores asociados, pueden darse determinadas interacciones entre ellos en función del modo de trabajo:**

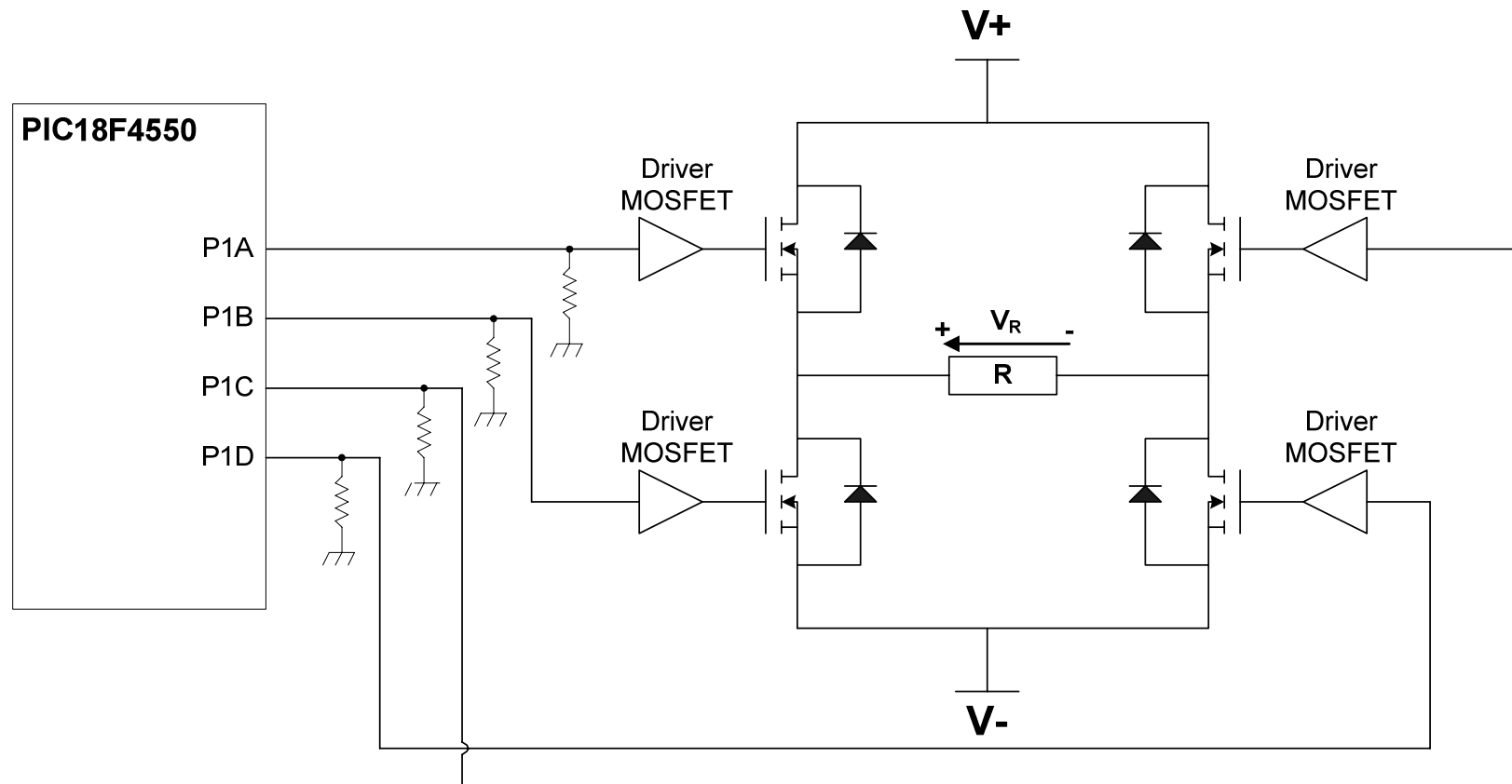
Modo CCP	Modo ECCP	Interacción
Captura	Captura	Cada módulo puede utilizar TMR1 o TMR3 como base de tiempos. La base de tiempos puede ser diferente para cada módulo
Captura	Comparación	El ECCP puede resetear el TMR1 o el TMR3 y por tanto puede afectar el funcionamiento del CCP si ambos trabajan con la misma base de tiempos
Comparación	Captura	El CCP puede resetear el TMR1 o el TMR3 y por tanto puede afectar el funcionamiento del ECCP si ambos trabajan con la misma base de tiempos
Comparación	Comparación	Ambos módulos pueden resetear el TMR1 o el TMR3 y por tanto se pueden producir conflictos si ambos trabajan con la misma base de tiempos
Captura	PWM (Mejorado y estandar)	Ninguna
Captura	PWM (Mejorado y estandar)	Ninguna
PWM (Mejorado y estandar)	Captura	Ninguna
PWM (Mejorado y estandar)	Comparación	Ninguna
PWM (Mejorado y estandar)	PWM (Mejorado y estandar)	Las señales PWM de ambos módulos tendrán la misma frecuencia y la actualización del duty se hará al mismo tiempo

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### EJERCICIO:

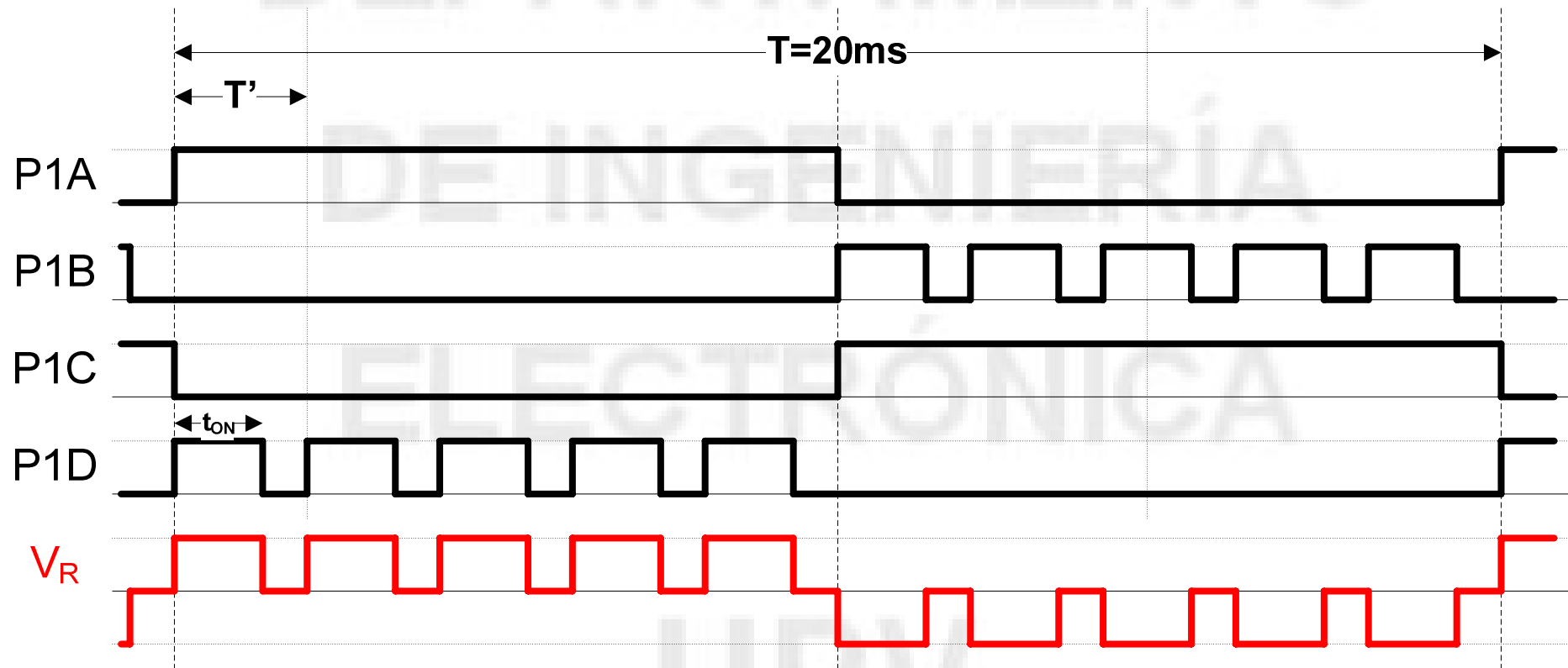
Escribir en código de un programa para el uC PIC18F4550 que genere las señales de disparo de la etapa de potencia de un inversor con carga resistiva utilizando el modo PWM mejorado del ECCP. El valor eficaz de la tensión en la carga se establece mediante una constante: DUTY ( $F_{osc}=4\text{MHz}$ ).



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### EJERCICIO:



#### EJERCICIO:

Dado que el periodo de  $V_R$  es  $T=20\text{ms}$  el periodo de los pulsos que deben generarse en P1B y P1D será  $T'=2\text{ms}$ . Los pulsos se generarán con el módulo ECCP trabajando en el modo PWM mejorado en puente completo con las líneas P1A, P1B, P1C y P1D activas a nivel alto ( $\text{CCP1M3}..\text{CCP1M0}="1100"$ ). En el semiperiodo positivo de  $V_R$  el módulo ECCP trabajará en sentido directo ( $\text{P1M1}..\text{P1M0}="01"$ ), mientras que en el semiperiodo negativo el ECCP trabajará en sentido inverso ( $\text{P1M1}..\text{P1M0}="11"$ ). El cambio de sentido debe realizarse cada 5 pulsos. Para ello se configura el postescalar del Temporizador 2 en 1:5. De esta forma cada vez que hayan generado 5 pulsos se producirá una interrupción del Temporizador 2. Dicha interrupción se utilizará para cambiar el sentido (directo o inverso). Para cambiar de sentido basta con complementar el bit P1M1.

## TEMA 3: MICROCONTROLADOR PIC18F4550

### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

---

#### EJERCICIO:

```
#include <p18F4550.h> // Declaración de librerías
```

```
// Bits de configuración
```

```
#pragma config WDT=OFF
```

```
// WatchDog desactivado
```

```
#pragma config OSC=HS
```

```
// Oscilador de alta velocidad
```

```
const int DUTY=90;
```

```
// Valor del duty de la señal generada (en %)
```

```
unsigned short long CTon;
```

```
// Variable correspondiente a la duracion del estado alto
```

```
void R_Int_Alta(void); // Declaración rutina de tratam. de interrupciones de alta prioridad
```

```
// Vectorización de interrupciones de alta prioridad
```

```
#pragma code Vector_Int_Alta=0x08
```

```
void Int_Alta (void)
```

```
{
```

```
    _asm GOTO R_Int_Alta _endasm // Se salta a la rutina de tratamiento de
```

```
}
```

```
// interrupciones de alta
```

```
prioridad
```

```
#pragma code
```

### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### EJERCICIO:

```
// Rutina de tratamiento de interrupciones de alta prioridad
#pragma interrupt R_Int_Alta
void R_Int_Alta (void)
{
    if (PIR1bits.TMR2IF) // Se comprueba si se ha producido interrupción del Temporizador 2
    {
        // Si se ha producido interrupción del Temporizador 2:
        PIR1bits.TMR2IF=0; // Se borra el flag de interrupción del Temporizador 2
        if (CCP1CONbits.P1M1==0) // Se comprueba el valor anterior del bit P1M1
        {
            // Si P1M1='0':
            CCP1CONbits.P1M1=1; // Se pone P1M1 a '1' (cambiamos a sentido inverso)
        }
        else
        {
            // Si P1M1='1'
            CCP1CONbits.P1M1=0; // Se pone P1M1 a '0' (cambiamos a sentido directo)
        }
    }
}
```



### TEMA 3: MICROCONTROLADOR PIC18F4550

#### UNIDADES FUNCIONALES / MÓDULO DE COMPARACIÓN/CAPTURA/PWM MEJORADO (ECCP)

##### EJERCICIO:

void main (void)

{

```
    TRISC=0xFB;      // Se inicializa RC2/P1A como salida
    TRISD=0x1F;      // Se inicializan RD5/P1B, RD6/P1C y RD7/P1D como salidas
    T2CON=0x27;      // Se configura el Temp. con el prescalar a 1:16
                    // y el post-escalar a 1:5 y se pone en marcha
    CCP1CON=0x4C;    // Se configura el ECCP en modo de PWM mejorado en puente
                    // completo con P1A, P1B, P1C y P1D activas por nivel alto
                    // y en sentido directo
    PR2=125;         // Se fija el periodo Tpwm=2ms
                    //      Tpwm=((PR2+1)*4*Prescaler)/Fosc
                    // => PR2 = [(Fosc*Tpwm)/(Prescaler*4)]-1
                    //      = [(4E6*2E-3)/(16*4)]= 125
    CTon=DUTY;       // Se calcula el valor de cuenta para la duración del nivel alto
    CTon=CTon*500;    //      DUTY(%)=(CTon*Prescaler*100)/(Fosc*Tpwm)
    CTon=CTon/100;    // => CTon = (DUTY*Fosc*Tpwm)/(Prescaler*100)=(DUTY*500)/100

    CCPR1L=CTon/4;    // Se carga en CCPR1 la duración del nivel alto (8 bits mas significativos)
    CCP1CONbits.DC1B0=CTon&0x0001; // Se cargan los 2 bits menos significativos de la
    CCP1CONbits.DC1B1=CTon&0x0002; // duración del nivel alto en DC1B0 y DC1B1
    INTCONbits.PEIE=1; // Se habilitan a nivel global las interrupciones de periféricos
    INTCONbits.GIE=1; // Se habilitan a nivel global todas las interrupciones
    PIE1bits.TMR2IE=1; // Se habilita la interrupción del Temporizador 2
```

while (1);

}

