



Módulo de Conversión A/D

Características generales en el PIC16F877

Ocho canales de conversión.

Cinco pines E/S de PORTA y los tres de PORTE.

Convierte la señal analógica en un número digital de 10 bits.

Tensión de referencia seleccionable por software.

Puede ser V_{DD} o la tensión aplicada en los pines RA2 y/o RA3.

Posibilidad de seguir funcionando cuando el PIC está en modo *SLEEP*.

Hay 11 registros asociados a este periférico.

Definición de pines de entrada

TRISA - PORTA - TRISE - PORTE

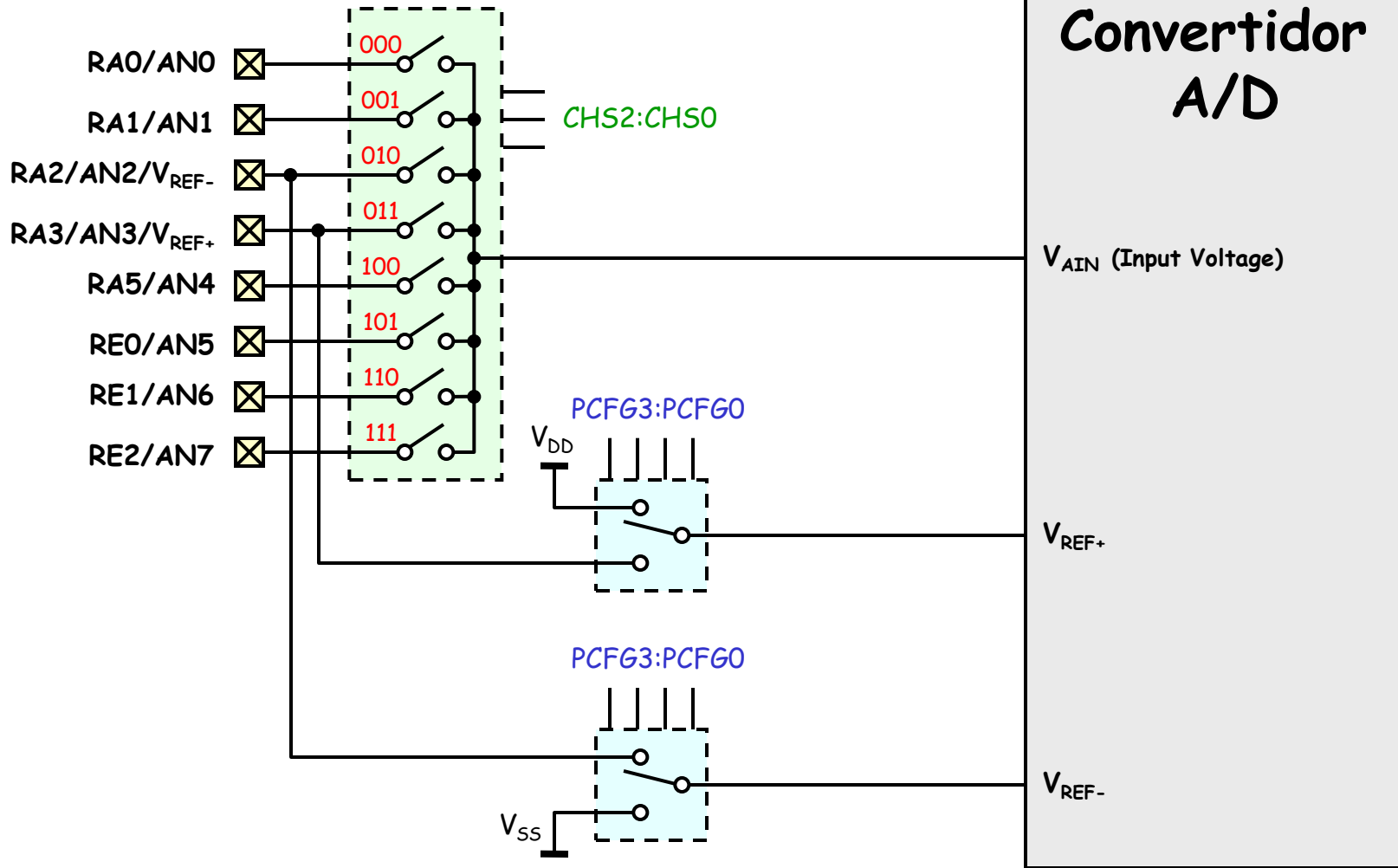
Manejo de interrupciones

INTCON - PIE1 - PIR1

Control del conversor A/D

ADCON0 - ADCON1 - ADRESH - ADRESL

Estructura interna



Registro ADCON0 (1Fh)



bit 7-6 **ADCS1:ADCS0:** Selección del reloj para la conversión A/D

00 = $f_{osc} / 2$

01 = $f_{osc} / 8$

10 = $f_{osc} / 32$

11 = f_{RC}

bit 5-3 **CHS2:CHS0:** Selección del canal de conversión

000 = Canal 0

001 = Canal 1

010 = Canal 2

011 = Canal 3

100 = Canal 4

101 = Canal 5

110 = Canal 6

111 = Canal 7

bit 2 **GO/ $\overline{\text{DONE}}$:** Estado de la conversión

Si ADON=1:

1 = Conversión en progreso

0 = Conversión finalizada

bit 0 **ADON:** Bit de encendido del convertidor A/D

1 = Módulo A/D encendido

0 = Módulo A/D apagado

Registro ADCON1 (9Fh)

ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0
------	---	---	---	-------	-------	-------	-------

bit 7 **ADFM:** Selección de formato del resultado

1 = Ajuste a la derecha

0 = Ajuste a la izquierda

bit 3-0 **PCFG3:PCFG0:** Configuración de las entradas al módulo A/D

PCFG3: PCFG0	AN7 RE2	AN6 RE1	AN5 RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0
0000	A	A	A	A	A	A	A	A
0001	A	A	A	A	V _{REF+}	A	A	A
0010	D	D	D	A	A	A	A	A
0011	D	D	D	A	V _{REF+}	A	A	A
0100	D	D	D	D	A	D	A	A
0101	D	D	D	D	V _{REF+}	D	A	A
011x	D	D	D	D	D	D	D	D
1000	A	A	A	A	V _{REF+}	V _{REF-}	A	A
1001	D	D	A	A	A	A	A	A
1010	D	D	A	A	V _{REF+}	A	A	A
1011	D	D	A	A	V _{REF+}	V _{REF-}	A	A
1100	D	D	D	A	V _{REF+}	V _{REF-}	A	A
1101	D	D	D	D	V _{REF+}	V _{REF-}	A	A
1110	D	D	D	D	D	D	D	A
1111	D	D	D	D	V _{REF+}	V _{REF-}	D	A

Registro INTCON (0Bh , 8Bh , 10Bh , 18Bh)

GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
-----	------	------	------	------	------	------	------

bit 7 **GIE:** *Habilitación global de interrupciones*

bit 6 **PEIE:** *Habilitación de interrupciones de periféricos*

Registro PIE1 (8Ch)

PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
-------	------	------	------	-------	--------	--------	--------

bit 6 **ADIE:** *Habilitación de la interrupción del convertidor A/D*

Registro PIR1 (0Ch)

PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
-------	------	------	------	-------	--------	--------	--------

bit 6 **ADIF:** *Flag de la interrupción del convertidor A/D*

1 = Conversión A/D completada.

0 = Conversión A/D aún no completada.

Pasos en una conversión A/D

1. Configurar el módulo A/D.

- Definir entradas analógicas y tensión de referencia. (ADCON1)
- Seleccionar el canal de la conversión. (ADCON0)
- Seleccionar el reloj de la conversión. (ADCON0)
- Encender el módulo A/D. (ADCON0)

2. Configurar la interrupción por conversión A/D.

- Bajar el *flag* ADIF. (PIR1)
- Habilitar la interrupción del convertidor A/D. (PIE1)
- Habilitar las interrupciones de los periféricos. (INTCON)
- Habilitar la máscara global de interrupciones. (INTCON)

3. Esperar a que transcurra el tiempo de adquisición.

- Tiempo necesario para capturar el valor analógico a convertir.
- Los valores típicos del tiempo de adquisición son del orden de 20 μ s.

4. Comenzar la conversión.

- Poner a "1" el bit GO/DONE. (ADCON0)

No activar este bit a la vez que se enciende el convertidor A/D

5. Esperar a que se complete la conversión A/D.

a) Controlando cuándo el bit GO/DONE se pone a "0".

b) Esperando a que llegue la interrupción del convertidor.

6. Leer el resultado de la conversión.

- Disponible en los registros ADRESH:ADRESL.

- Bajar el *flag* ADIF si se están usando interrupciones.

7. Llevar a cabo la siguiente conversión.

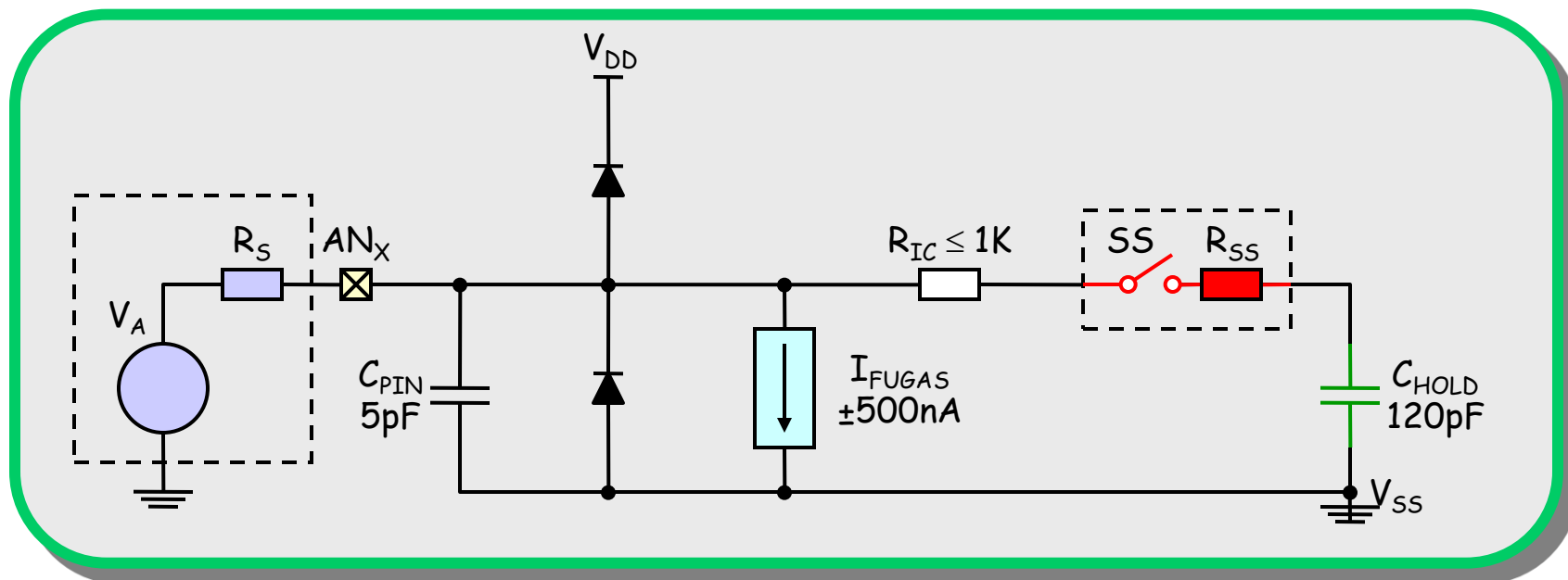
- Volver al paso 1 ó 2, según convenga.

- Espera mínima antes de empezar la siguiente adquisición: $2 \cdot T_{AD}$.

T_{AD} : Tiempo necesario para la conversión de un bit.

Tiempo de adquisición

Tiempo necesario para cargar el condensador de mantenimiento (C_{HOLD}).



Máxima impedancia recomendada para la fuente: 10K.

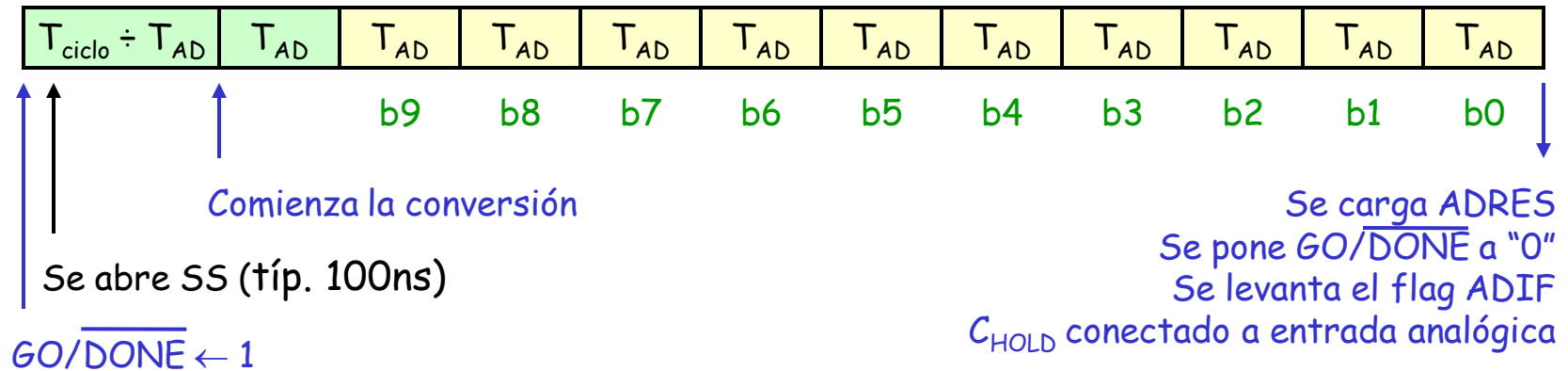
En estas condiciones, $T_{\text{ACQ}} \approx 20\mu\text{s}$.

Mientras no se complete la conversión, no empieza otra adquisición.

Esperar T_{ACQ} : a) tras una conversión; b) tras seleccionar un nuevo canal;
c) tras encender el módulo A/D.

Tiempo de conversión

La conversión de 10 bits dura $12 \cdot T_{AD}$.



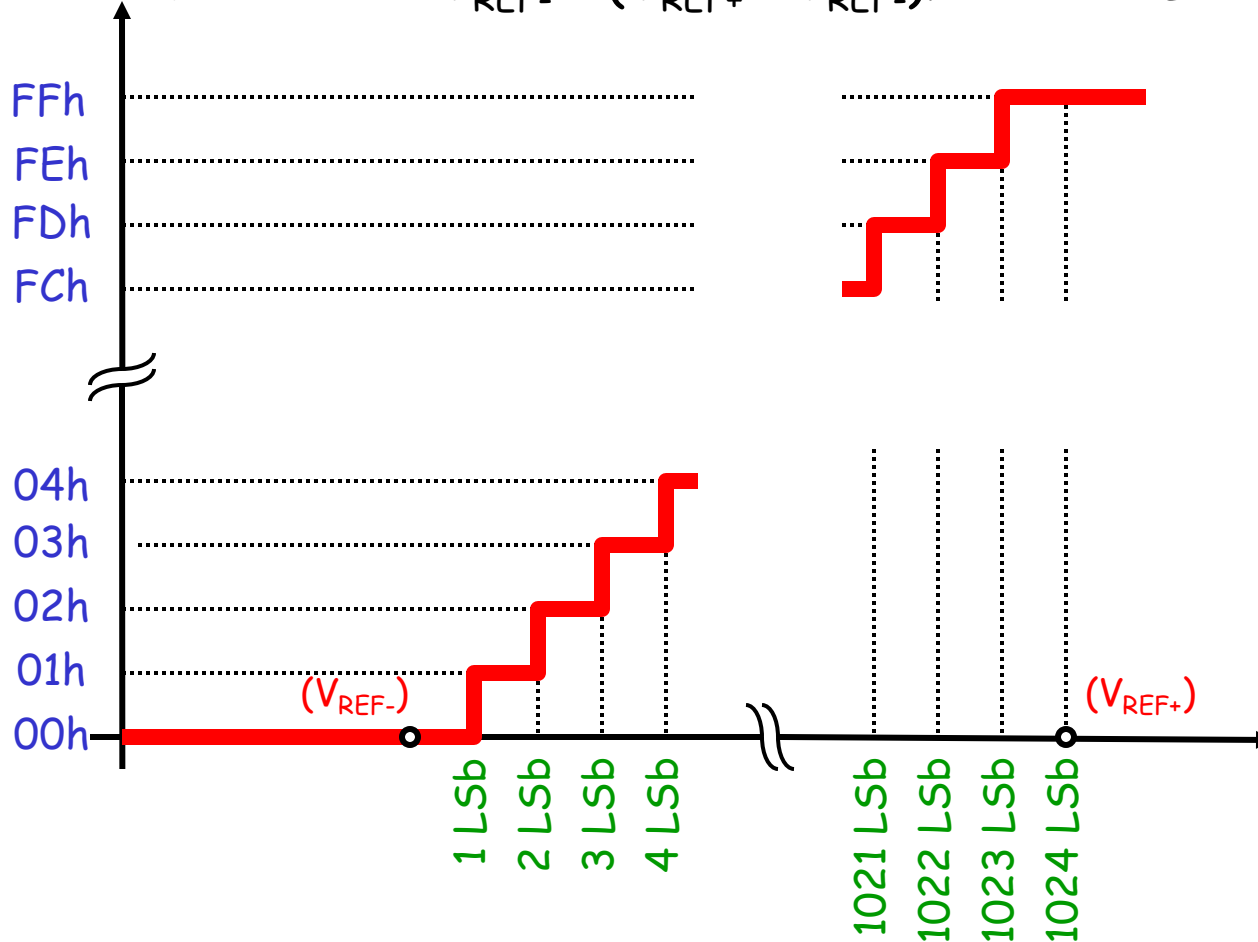
T_{AD} configurable en ADCON0 (reloj de la conversión).

$$T_{AD}=2 \cdot T_{OSC} - T_{AD}=8 \cdot T_{OSC} - T_{AD}=32 \cdot T_{OSC} - T_{AD}=2\mu s \div 6\mu s \text{ (típ. } 4\mu s)$$

Para un funcionamiento correcto se necesita un valor mínimo de $T_{AD}=1,6\mu s$.

Función de transferencia

La primera transición tiene lugar cuando la tensión analógica de entrada alcanza el valor $V_{REF-} + (V_{REF+} - V_{REF-})/1024 \equiv 1LSb$.



Conversión A/D en el compilador C de CCS

Configuración del módulo conversor A/D

setup_adc (**modo**);

modo:	ADC_OFF	(ADCON0 ← 00h)
	ADC_CLOCK_DIV_2	(ADCON0 ← 01h)
	ADC_CLOCK_DIV_8	(ADCON0 ← 41h)
	ADC_CLOCK_DIV_32	(ADCON0 ← 81h)
	ADC_CLOCK_INTERNAL	(ADCON0 ← C1h)

Definición de entradas analógicas

setup_adc_ports (**valor**);

valor:	NO_ANALOGS	(ADCON1 ← 86h)
	ALL_ANALOG	(ADCON1 ← 80h)
	ANALOG_RA3_REF	(ADCON1 ← 81h)
	A_ANALOG	(ADCON1 ← 82h)
	A_ANALOG_RA3_REF	(ADCON1 ← 83h)
	RA0_RA1_RA3_ANALOG	(ADCON1 ← 84h)
	RA0_RA1_ANALOG_RA3_REF	(ADCON1 ← 85h)

Definición de entradas analógicas (cont.)

setup_adc_ports (**valor**):

valor:	ANALOG_RA3_RA2_REF	(ADCON1 ← 88h)
	ANALOG_NOT_RE1_RE2	(ADCON1 ← 89h)
	ANALOG_NOT_RE1_RE2_REF_RA3	(ADCON1 ← 8Ah)
	ANALOG_NOT_RE1_RE2_REF_RA3_RA2	(ADCON1 ← 8Bh)
	A_ANALOG_RA3_RA2_REF	(ADCON1 ← 8Ch)
	RA0_RA1_ANALOG_RA3_RA2_REF	(ADCON1 ← 8Dh)
	RA0_ANALOG	(ADCON1 ← 8Eh)
	RA0_ANALOG_RA3_RA2_REF	(ADCON1 ← 8Fh)

Selección del canal analógico

set_adc_channel (**canal**):

canal:	0	(AN0)	4	(AN4)
	1	(AN1)	5	(AN5)
	2	(AN2)	6	(AN6)
	3	(AN3)	7	(AN7)

Lectura del resultado

valor = read_adc ();

valor: Entero de 16 bits según la directiva *#device adc=* empleada.

La influencia de dicha directiva se recoge en la siguiente tabla

#device	8 bits	10 bits	11 bits	16 bits
adc=8	00-FF	00-FF	00-FF	00-FF
adc=10	x	0-3FF	x	x
adc=11	x	x	0-7FF	x
adc=16	0-FF00	0-FFC0	0-FFE0	0-FFFF

El fichero 16f877.h incluye como primera directiva *#device PIC16F877*.

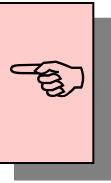
Es necesario incluir información del tipo de convertor A/D. Por ello los ficheros C que usen este módulo deberán comenzar por

```
#include "16f877.h"  
#device adc=10
```

Ejemplo

setup_adc_ports (A_ANALOG);

$W \leftarrow 10000010$
Banco 1
 $ADCON1 \leftarrow W$
Banco 0



setup_adc (ADC_CLOCK_INTERNAL);

$W \leftarrow ADCON0$
 $W \leftarrow W \& 00111000$
 $W \leftarrow W \mid 11000001$
 $ADCON0 \leftarrow W$



set_adc_channel (3);

$W \leftarrow ADCON0$
 $W \leftarrow W \& 11000111$
 $W \leftarrow W \mid 00011000$
 $ADCON0 \leftarrow W$

delay_us (20);

valor = read_adc ();

$\overline{GO/DONE} \leftarrow 1$
Espera a $\overline{GO/DONE}=0$
 $W \leftarrow ADRESH$
 $TEMP \leftarrow W$
Banco 1
 $W \leftarrow ADRESL$
Banco 0
 $REGL \leftarrow W$
 $W \leftarrow TEMP$
 $REGH \leftarrow W$

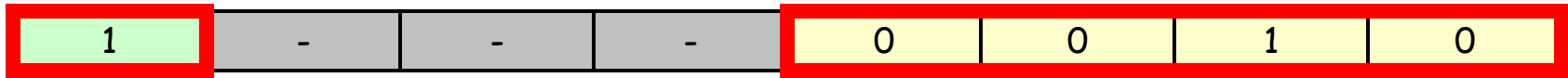
setup_adc (ADC_OFF);

$W \leftarrow ADCON0$
 $W \leftarrow W \& 00111000$
 $ADCON0 \leftarrow W$

Comentario



Registro ADCON1 (9Fh)



bit 7 **ADFM**: Selección de formato del resultado

1 = Ajuste a la derecha ←

0 = Ajuste a la izquierda

bit 3-0 **PCFG3:PCFG0**: Configuración de las entradas al módulo A/D

PCFG3: PCFG0	AN7 RE2	AN6 RE1	AN5 RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0
0000	A	A	A	A	A	A	A	A
0001	A	A	A	A	V _{REF+}	A	A	A
0010	D	D	D	A	A	A	A	A
0011	D	D	D	A	V _{REF+}	A	A	A
0100	D	D	D	D	A	D	A	A
0101	D	D	D	D	V _{REF+}	D	A	A
011x	D	D	D	D	D	D	D	D
1000	A	A	A	A	V _{REF+}	V _{REF-}	A	A
1001	D	D	A	A	A	A	A	A
1010	D	D	A	A	V _{REF+}	A	A	A
1011	D	D	A	A	V _{REF+}	V _{REF-}	A	A
1100	D	D	D	A	V _{REF+}	V _{REF-}	A	A
1101	D	D	D	D	V _{REF+}	V _{REF-}	A	A
1110	D	D	D	D	D	D	D	A
1111	D	D	D	D	V _{REF+}	V _{REF-}	D	A



Registro ADCON0 (1Fh)



bit 7-6 **ADCS1:ADCS0**: Selección del reloj para la conversión A/D

00 = $f_{osc} / 2$

01 = $f_{osc} / 8$

10 = $f_{osc} / 32$

11 = f_{RC} ←

bit 5-3 **CHS2:CHS0**: Selección del canal de conversión

000 = Canal 0

001 = Canal 1

010 = Canal 2

011 = Canal 3 ←

100 = Canal 4

101 = Canal 5

110 = Canal 6

111 = Canal 7

bit 2 **$\overline{GO/DONE}$** : Estado de la conversión

Si ADON=1:

1 = Conversión en progreso

0 = Conversión finalizada ←

bit 0 **ADON**: Bit de encendido del convertidor A/D

1 = Módulo A/D encendido ←

0 = Módulo A/D apagado



El código ensamblador recogido en este ejemplo supone que **valor** es una variable de tipo *long int* y que se ha llevado a cabo una conversión A/D de 10 bits, para lo cual el código C deberá comenzar como se indica:

```
#include "16F877.h"  
#device ADC=10           // Selecciona conversión A/D de 10 bits
```

Si no se incluye la directiva **#device ADC=10**, el compilador considera que se debe hacer una conversión A/D de 8 bits. En este caso, y considerando ahora que **valor** es una variable de tipo *int*, el código ensamblador que se genera difiere del visto.

```
setup_acd_ports (A_ANALOG):  
  W ← 00000010  
  Banco 1  
  ADCON1 ← W  
  Banco 0
```

Ajusta resultado de conversión a la izquierda

```
valor = read_adc():  
  GO/DONE ← 1  
  Espera GO/DONE=0  
  W ← ADRESH  
  REG ← W
```

El contenido de **valor** se obtiene de un solo registro: **ADRESH** (ajuste a la izda)