

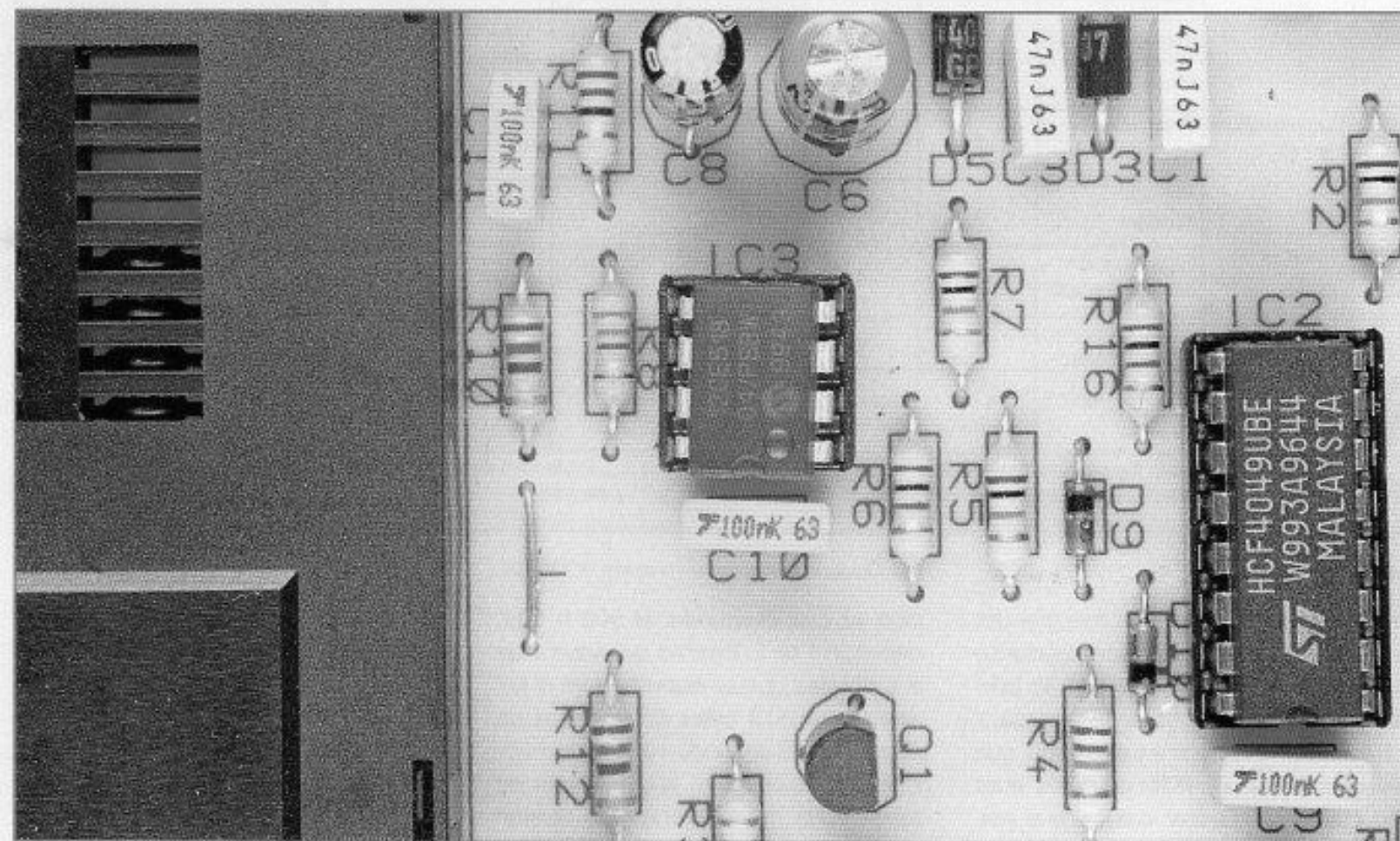
MICROCONTROLADORES PIC12CXXX

La familia 12Cxxx está compuesta por ocho modelos

Los microcontroladores que vamos a estudiar componen una familia de bajo coste, pero con altas prestaciones. Sus encapsulados, de ocho patillas, les confiere una presencia un poco dudosa, pues no parece que componentes tan diminutos dispongan de tantas funciones como tienen: memoria EEPROM para datos, cuatro canales de convertidores Analógico/Digital, temporizado-

res, etc. Evidentemente, no todos los modelos que componen la familia disponen de tales funciones; de ahí que esté compuesta por ocho dispositivos diferentes de los cuales el que reúne todas estas características, por lo tanto el más completo, es el PIC16CE674. Todos ellos son compatibles con todos los elementos que les son comunes, es decir, repertorio de 33 instrucciones, modos de configuración del oscilador principal del

sistema, etc. Las pequeñas diferencias de cada modelo, que en algunas ocasiones son insignificantes –por ejemplo, la gama 67X tiene dos instrucciones más que la gama inferior– en otras, tales diferencias se ven acentuadas si el diseño a realizar requiere elevada velocidad de proceso, pues mientras la gama 5XX tiene una frecuencia máxima de operación de 4 MHz., la más alta llega hasta los 10 MHz.



Los PIC12CXXX, aunque son diminutos en tamaño, proporcionan grandes prestaciones.

FAMILIA PIC12CXXX

		PIC12C508 (A)	PIC12C509 (A)	PIC12CE518	PIC12CE519	PIC12C671	PIC12C672	PIC12CE673	PIC12CE674
Reloj	Máxima frecuencia de operación (MHz)	4	4	4	4	10	10	10	10
Memoria	Memoria EEPROM de Programa	512x12	1024x12	512x12	1024x12	1024x14	2048x14	1024x14	2048x14
	Memoria RAM de Datos	25	41	25	41	128	128	128	128
Periféricos	Memoria EEPROM de Datos (bytes)	---	---	16	16	---	---	16	16
	Módulos Temporizadores	TMR0	TMR0	TMR0	TMR0	TMR0	TMR0	TMR0	TMR0
	Canales de conversión A/D (8 bits)	---	---	---	---	4	4	4	4
Características	Despertar de SLEEP por cambios en pines	Si	Si	Si	Si	Si	Si	Si	Si
	Fuentes de Interrupción	---	---	---	---	4	4	4	4
	Pines I/O	5	5	5	5	5	5	5	5
	Pines de entrada	1	1	1	1	1	1	1	1
	Pull-ups internas	Si	Si	Si	Si	Si	Si	Si	Si
	Programación serie en circuito	Si	Si	Si	Si	Si	Si	Si	Si
	Número de Instrucciones	33	33	33	33	35	35	35	35
	Encapsulado	8-pin DIP, JW, SOIC	8-pin DIP, JW, SOIC	8-pin DIP, JW, SOIC	8-pin DIP, JW, SOIC	8-pin DIP, JW, SOIC	8-pin DIP, JW, SOIC	8-pin DIP, JW	8-pin DIP, JW

Todos tienen Reset interno de conexión de alimentación Watchdog, Protección de Código.
 Todos son programables modo serie con GP0 para datos y GP1 para reloj.

 Circuitos que componen la familia PIC12CXXX.

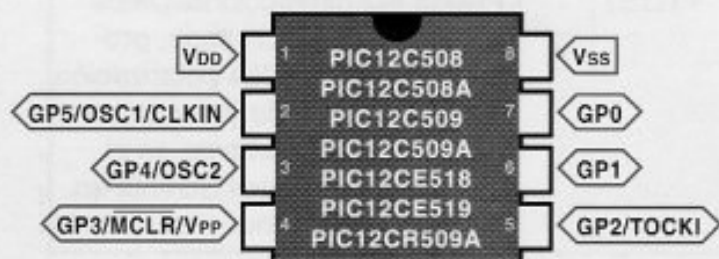
Así pues, si el diseño no requiere grandes velocidades de trabajo ni son necesarios convertidores Analógico/Digital, se pueden emplear los más bajos de la gama. Y si, además, tampoco es necesario que tenga memoria EEPROM de datos, pues no es preciso guardar datos y que permanezcan almacenados, aún cuando se quite la alimentación al circuito, la elec-

ción ya está efectuada: el 508 o 509, dependerá de lo extenso que vaya a ser el programa que se elabore, pues el primero tiene 512 bytes de memoria de programa y el segundo 1024 bytes. Como ya se comentó páginas atrás, en todos estos modelos hay una versión que tiene una ventana para poder borrar la memoria de programa con rayos UVA.

Por el momento nos vamos a centrar en la gama de los modelos 5XX y después veremos las funciones aplicables a los de la gama más alta.

ARQUITECTURA

La configuración de estos microcontroladores es similar a los de las restantes gamas. Su organización se basa en la



Circuito	Memoria			
	EPROM Programa	ROM Programa	RAM Datos	EEPROM Datos
PIC12C508	512x12	-	25	-
PIC12C508A	512x12	-	25	-
PIC12C509	1024x12	-	41	-
PIC12C509A	1024x12	-	41	-
PIC12CE518	512x12	-	25	16
PIC12CE519	1024x12	-	41	16
PIC12CR509A	-	1024x12	41	-

Disposición de las patillas y características más importantes de los microcontroladores 12C5XX.

arquitectura Harvard; con ella, la CPU accede de forma independiente y simultánea a la memoria de datos y a la de instrucciones. El aislamiento que este sistema proporciona permite que cada bus tenga el tamaño más adecuado. Así, el bus de datos tiene una longitud de 8 bits (un byte), mientras que el de las instrucciones es de 12 bits.

En la ejecución de las instrucciones se aplica la técnica de segmentación (Pipe-Line); de este modo el procesador puede ejecutar la instrucción y buscar la siguiente al mismo tiempo, es lo que se denomina ciclo de instrucción y equivale

a cuatro ciclos de reloj. Todas las instrucciones, excepto las de salto, se ejecutan en un sólo ciclo, por lo que el tiempo que le lleva al procesador en realizarlo es de 1 microsegundo (tomando como frecuencia de reloj 4 MHz.).

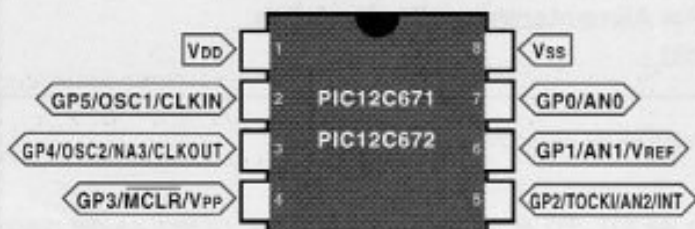
Todas las instrucciones pueden manejar cualquier elemento de la arquitectura como fuente o como destino, por lo que las instrucciones son ortogonales.

La ALU (Unidad Lógica Aritmética) realiza sus operaciones aritmético-lógicas con dos operandos, uno que se recibe desde el registro W, que hace las funciones de Acumulador de los microprocesadores

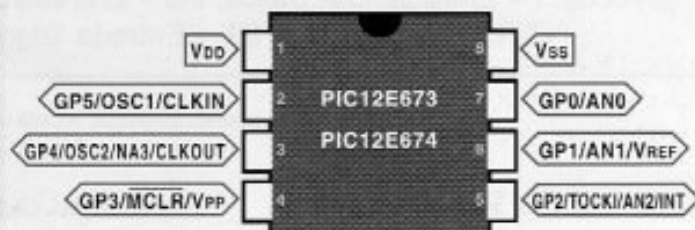
convencionales, y otro que puede provenir de cualquier registro. El mencionado registro W, trabaja con ocho bits y no es un registro direccionable, es decir, no ocupa una dirección en el mapa de memoria.

Dependiendo de la instrucción que se ejecute, la ALU puede alterar el contenido del registro de Estado. Este registro tiene ocho bits y cada uno de ellos informa del estado de la "bandera" afectada por la operación que se acaba de realizar. Así por ejemplo, después de una operación de suma, resta o desplazamiento, el bit 0 (Carry), nos indicará si ha habido o no desbordamiento.

Circuito	EPROM	RAM
PIC12C671	1024x14	128x8
PIC12C672	2048x14	128x8



Circuito	Memoria		
	Programa	Datos RAM	Datos EEPROM
PIC12CE673	1024x14	128x8	16x8
PIC12CE674	2048x14	128x8	16x8



Disposición de las patillas y características más importantes de los microcontroladores 12C6XX.

DESCRIPCIÓN DE PATILLAS DEL PIC12CE5XXX

NOMBRE	DIP PIN #	SOIC PIN #	TIPO I/O/P	TIPO BUFFER	DESCRIPCIÓN
GP0	7	7	I/O	TTL/ST	Puerto I/O bidireccional/Datos para programación serie, programable para débil polarización Pull-up de entrada y despertar de SLEEP por cambios en el pin. Entrada Trigger Schmitt en programación serie.
GP1	6	6	I/O	TTL/ST	Puerto I/O bidireccional/Reloj para programación serie, programable para débil polarización Pull-up de entrada y despertar de SLEEP por cambios en el pin. Entrada Trigger Schmitt en programación serie.
GP2/T0CK1	5	5	I/O	ST	Puerto I/O Bidireccional. Puede configurarse como T0CK1.
GP3/MCLR/VPP	4	4	I	TTL/ST	Puerto de entrada/Master clear (Reset)/Entrada de Voltaje de programación. Programable para débil polarización Pull-ups de entrada y despertar de SLEEP por cambios en el pin. Entrada Trigger Schmitt cuando es configurada como MCLR.
GP4/OSC2	3	3	I/O	TTL	Puerto I/O Bidirección/Salida de oscilador Conexión para oscilador de cristal o resonador en modo XT y LP.
GP5/OSC1/CLKIN	2	2	I/O	TTL/ST	Puerto I/O bidireccional/Entrada de oscilador o cristal externo. GPIO, ST en modo oscilador RC externo.
VDD	1	1	P	---	Alimentación positiva.
VSS	8	8	P	---	Masa o común para lógica e I/O.

Leyenda: I = Entrada, O = Salida, I/O = Entrada/Salida, P = Alimentación, --- = No usado, TTL = Entrada TTL, ST = Entrada Trigger Schmitt.

 Funciones asignadas a cada una de las patillas de los microcontroladores.

CICLO DE INSTRUCCIÓN

La señal de reloj es dividida internamente por cuatro, obteniendo cuatro señales de reloj Q1, Q2, Q3 y Q4. Internamente el Contador de Programa (PC) se ve incre-

mentado en cada tránsito de Q1. En el ciclo de ejecución, la instrucción es almacenada en el Registro de Instrucciones (IR) durante el ciclo Q1 y durante Q2, Q3 y Q4 tiene lugar la decodificación y ejecu-

ción de la misma. La lectura del dato del operando tiene lugar en el ciclo Q2 y es escrito en el destino durante el ciclo Q4. El ciclo de instrucción se puede ver en el cronograma que muestra la ilustración,

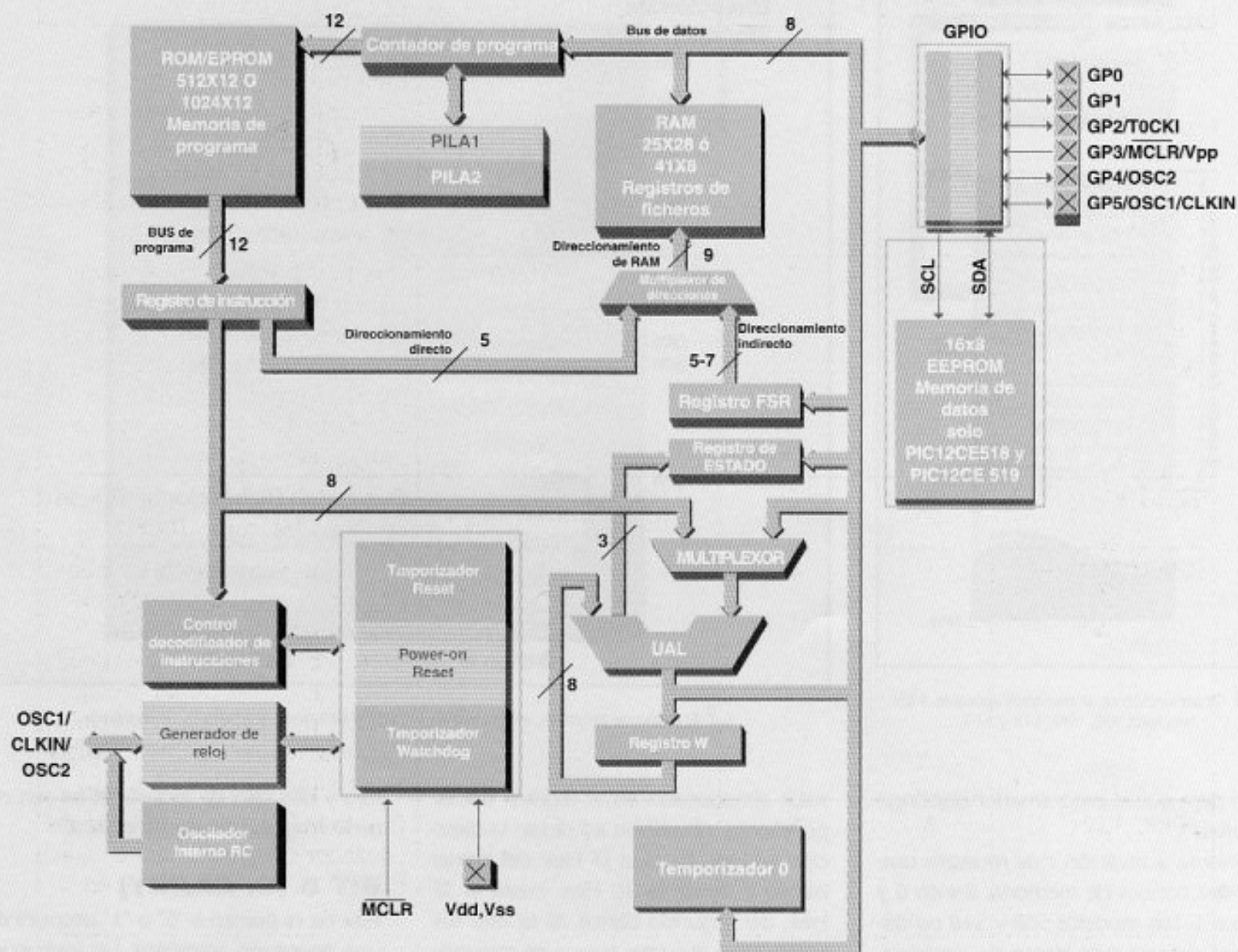


Diagrama de bloques de los PIC 12CX5XX.

al cual se le han añadido cuatro instrucciones como ejemplo: cargar W con 03, llevar este valor al puerto GPIO, llamar a una subrutina y poner a 1 el bit Bit 1 del puerto GPIO.

ORGANIZACIÓN DE LA MEMORIA

Como ya se ha mencionado, hay dos memorias diferentes, por un lado está la memoria de programa y por otro la de datos. Las ilustraciones correspondientes nos muestran cómo están organizados

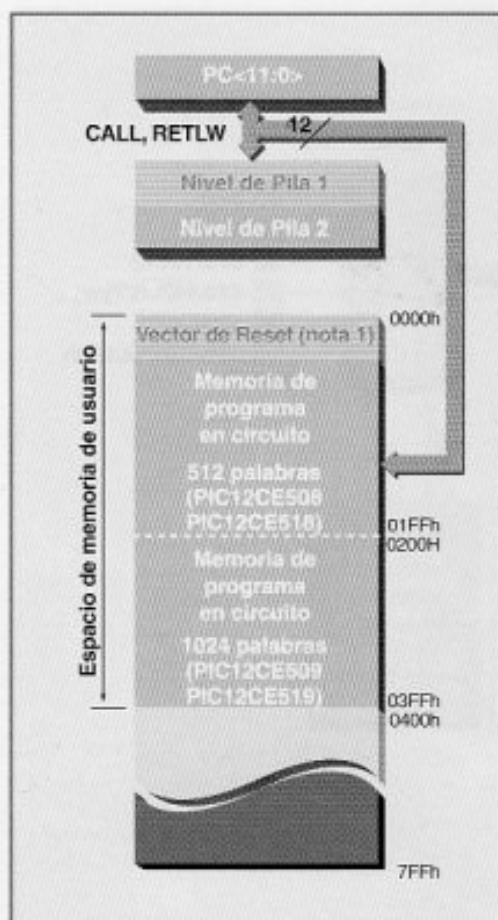
los bancos de dichas memorias. En el primero de ellos se puede comprobar cómo éste es común para los modelos 508, 509 518 y 519. La diferencia que hay entre los 508-518 y 509-519, es que los primeros sólo tienen 512 bytes de memoria, por lo que la dirección más alta se sitúa en 01FF Hex. y los segundos disponen de 1024 bytes y llegan hasta 03FF Hex.

La memoria de datos está compuesta por Registros o bytes de memoria RAM, de modo que cada dirección de la

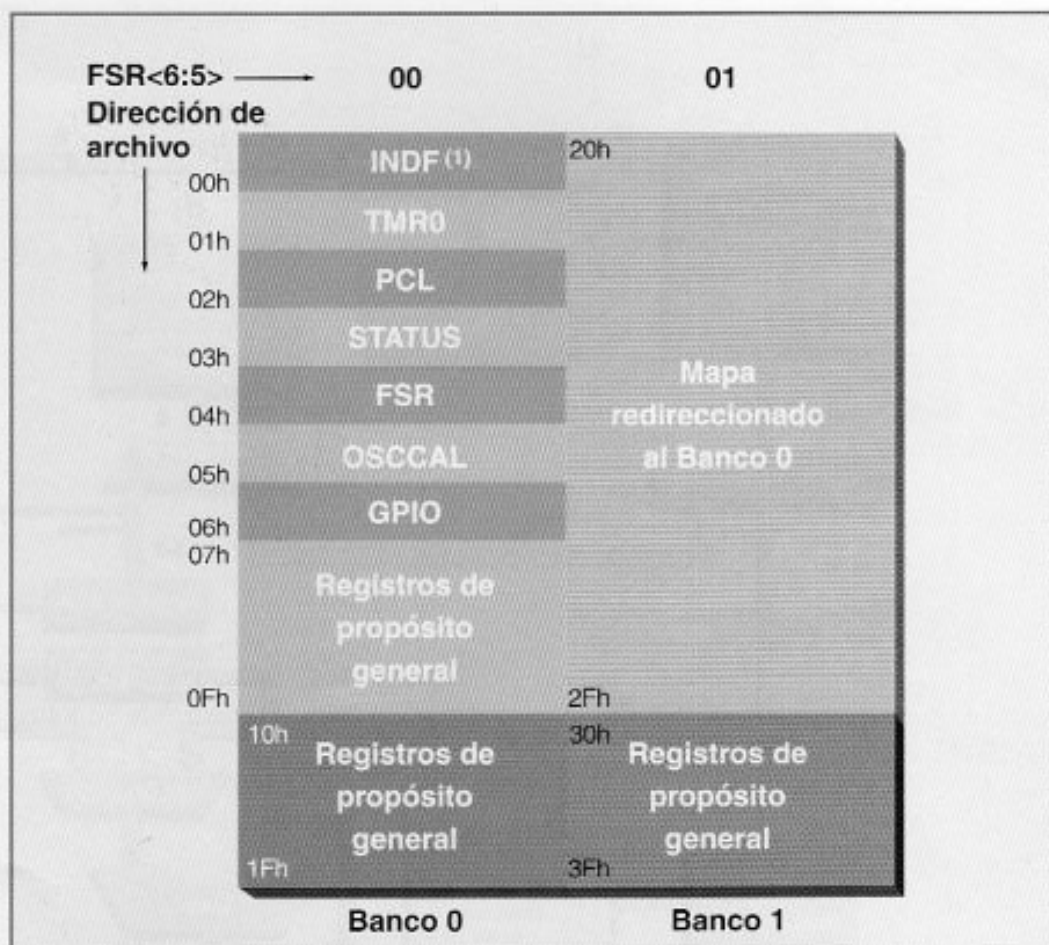
memoria puede ser especificada por el nombre del Registro; éstos están divididos en dos grupos: Registros especiales y Registros de propósito general.

Los Registros de funciones especiales ocupan las siete primeras direcciones de memoria RAM, en el orden representado en la ilustración del mapa de registros; cada uno de estos registros será detalladamente analizado más adelante.

Desde la dirección 07 Hex. en adelante son Registros de propósito general, es decir son direcciones libres de memoria



Organización de la memoria aplicable a los modelos 508, 509, 518 y 519.



El mapa de Registros, en el caso de los 509 y 519 tiene dos bancos de memoria.

RAM para que el programador disponga de ellas.

La misma ilustración nos muestra que hay dos bancos de memoria: Banco 0 y Banco 1. Los modelos 508 y 518 no disponen del segundo banco de memoria, por lo que la máxima dirección se sitúa en 1F Hex. (Banco 0). También podemos observar cómo hay una división en la dirección 0F Hex. en ambos bancos; su explicación es sencilla, pero primero se ha de entender que el hecho de que haya dos bancos en los modelos 509 y 519, supone que, en todo momento, se debe especificar en el programa el banco con el que se desea trabajar. Por consiguiente, si estamos trabajando con el Banco 1, no serán accesibles los registros del Banco 0. Pero esto no es del todo cierto, de ahí que haya una línea de división en 0F Hex. Hasta esta dirección las direcciones de los registros son "transparentes" es decir, desde cualquier banco de memoria se puede obtener el

valor almacenado en el registro correspondiente, no siendo así desde la dirección 10 Hex. hasta la 1F Hex. del primer banco y desde la 30 Hex. hasta la 3F Hex. del segundo banco. Si se está trabajando en el primer banco de memoria y se llama a un registro que ocupa una dirección superior a 30 Hex., obligatoriamente se ha de cambiar de banco de trabajo para poder acceder a él. Apparentemente puede parecer una labor complicada trabajar con los bancos de memoria, pero no es así, ya veremos algunos ejemplos que nos aclararán cualquier duda al respecto.

REGISTRO DE ESTADO (DIRECCIÓN 03 HEX.)

Aunque el Registro de Estado no ocupa la primera dirección de la memoria (ocupa la 03 Hex.), comenzaremos por este Registro ya que será uno de los primeros en utilizar. Como todos los registros, tiene ocho bits, cinco de ellos son de lectura/escri-

tura y dos sólo de lectura; el bit seis no tiene ninguna misión específica.

BIT 0 (C: CARRY)

Este bit es puesto a "0" o "1" después de una operación aritmética. Las instrucciones ADDWF (Suma del registro W y otro registro), SUBWF (Resta a un registro el contenido de W), RRF y RLF (Desplaza un bit a la derecha o izquierda, el contenido de un registro respectivamente), alteran el contenido de este bit. Por lo tanto, es puesto a uno cuando en una suma se produce un acarreo, en la resta no lo hay, cuando se desplaza a la derecha (RRF) y el bit LSB es un uno; y, por último, cuando hay un desplazamiento a la izquierda (RLF) y el bit MSB es un uno.

Una simple lectura de este bit nos informará de lo ocurrido en el proceso de la operación; por ejemplo, deseamos sumar al registro W, que tiene almacenado el valor 02 Hex. el contenido del registro

XX que tiene 0EHex. y guardar el resultado en W, tendremos:

```
ADDWF XX,W
BTFSS STATUS, C
GOTO NuevaDirección
...
```

Continúa el programa
La primera instrucción suma el contenido de los dos registros y guarda el resultado en el registro W que hace las veces de Acumulador. La segunda instrucción saltará una línea de programa si el bit 0 del Registro de Estado es un uno (Bit "C") –en este caso 02 + 0E Hex. es igual a 01, con un bit de arrastre o acarreo, por lo que el bit 0 del Registro de Estado será un uno– por lo tanto, el programa continuará ejecutándose después de la línea GOTO NuevaDirección y no se ejecutará esta instrucción de salto. Por el contrario, si la suma de ambos valores hubiera sido inferior o igual a FF Hex. el valor de "C" habría sido "0" y sería ejecutada la instrucción GOTO.

Esto mismo ocurre en las restas y desplazamientos, aunque en este último caso hay que tener en cuenta lo siguiente: al desplazar a la izquierda un bit el contenido de un registro, por ejemplo, el acarreo "C" se coloca en el bit menos significativo del registro, y "C" tomará el valor del bit más significativo de este último. Siguiendo con el ejemplo anterior, tendremos que el bit del acarreo quedó con el valor "1" y el registro W tomó el valor "01 Hex.", si hacemos un desplazamiento en W a la izquierda guardándolo en W, tendremos que después de:

```
RLF W,W
```

El nuevo valor de W será "03 Hex." y el acarreo "0", pues primero éste se introduce por la izquierda, se desplaza un bit y "C" toma el valor de MSB:

```
"C"<-00000001 <- "C=1"
```

los nuevos valores serán:

```
"C=0" y W = "00000011"
```

Si el desplazamiento hubiera sido a la derecha RRF W,W, el resultado sería:

```
"C=1->"10000000 ->"C=1"
```

donde "C" ha tomado el valor "1" del bit LSB.

Si no nos interesa que el "1" del bit de acarreo no interfiera en el resultado del desplazamiento, bastará con borrarlo antes de proceder a desplazar el contenido del registro. Para borrarlo insertaremos la instrucción:

```
BCF STATUS,C; borra "C"
RLF W,W; desplaza W
```

En este caso el resultado será muy diferente al que se obtuvo en el primer ejemplo:

```
"C"<-00000001 <- "C=0"
C="0" y W= "00000010"
```

SUMARIO DE REGISTROS DE FUNCIONES ESPECIALES (SFR) DEL PIC12CX5XX

Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset de conexión de alimentación	Otros reset
N/A	TRIS	—	—	GP5	GP4	GP3	GP2	GP1	GP0	- -11 1111	- -11 1111
N/A	OPTION	Bits de control Timer0, Timer0/WDT prescaler wake-up, y weak pull-ups								1111 1111	1111 1111
00h	INDF	Contenido de la dirección FSR de memoria (no es un registro físico)								xxxx xxxx	uuuu uuuu
01h	TMR0	Contador de 8 bits en tiempo real								xxxx xxxx	uuuu uuuu
02h	PCL	8 bits bajos del PC								1111 1111	1111 1111
03h	STATUS	GPWUF	—	PA0	TO	PD	Z	DC	C	0001 1xxx	q00q quuu
04h	FSR (12CE518)	Puntero dirección indirecta de memoria								111x xxxx	111u uuuu
04h	FSR (12CE519)	Puntero dirección indirecta de memoria								110x xxxx	11uu uuuu
05h	OSCCAL	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	—	—	1000 00- -	uuuu uu- -
06h	GPIO	SCL	SDA	GP5	GP4	GP3	GP2	GP1	GP0	11xx xxxx	11uu uuuu

REGISTRO DE ESTADO DEL PIC 12CE5XX

R/W-o	R/W-o	R/W-o	R-1	R-1	R/W-x	R/W-x	R/W-x
GPWUF	PA0	TO	PD	Z	DC	C	
Bit7	6	5	4	3	2	1	bit0

R= Sólo lectura
W= Sólo escritura
-n= Valor al reiniciar

- bit 7: **GPWUF**: GPIO bit reset
1 = Reset debido al despertar de SLEEP o cambios en los pines
0 = Después de alimentación u otros Reset
- bit 6: Sin implementar
- bit 5: **PA0**: Bit de selección de página
1 = Página 1 (200h - 3FFh) - PIC12CE519
0 = Página 0 (000h - 1FFh) - PIC12CE518 y PIC12CE519
Cada página es de 512 bytes.
- bit 4: **TO**: Bit Time-out
1 = Después de alimentación, instrucción CLRWDWT o SLEEP
0 = Cuando ha transcurrido el tiempo de WDT
- bit 3: **PD**: Bit power-down
1 = Después de poner alimentación o instrucción CLRWDWT
0 = Por la ejecución de la instrucción SLEEP
- bit 2: **Z**: Bit cero
1 = Cuando el resultado de una operación aritmética o lógica es cero
0 = Cuando el resultado de una operación aritmética o lógica no es cero
- bit 1: **DC**: Bit carry/borrow (para las instrucciones ADDWF y SUBWF)
ADDWF
1 = Se produce cuando hay acarreo en el cuarto bit
0 = Se produce cuando no hay acarreo en el cuarto bit
SUBWF
1 = Se produce cuando no hay acarreo en el cuarto bit
0 = Se produce cuando hay acarreo en el cuarto bit
- bit 0: **C**: Bit carry/borrow (acarreo para las instrucciones ADDWF, SUBWF, RRF y RLF)
ADDWF **SUBWF** **RRF o RLF**
1 = Se produce acarreo 1 = No se produce arrastre Carga bit con LSB o MSB respectivamente
0 = No se produce acarreo 0 = Se produce arrastre

BIT 2 (Z: ZERO)

Otro bit del Registro de Estado, muy similar al descrito en los párrafos anteriores, es el bit Z. Este bit toma el valor "1" cuando el resultado de una operación aritmética o lógica es cero; después de efectuar una resta (SUBWF) o cualquier operación lógica (ANDWF, ANDLW, XORWF, etc.), se puede comprobar si el resultado es cero leyendo el bit Z:

BTFSS STATUS,Z
GOTO NuevaDirección

Si la operación anterior a la instrucción de salto BTFSS fuera cero, el flujo de programa continuará después de la instrucción GOTO ya que el bit Z habrá tomado el valor "1".

En el próximo capítulo veremos estos mismos ejemplos con el simulador de MPLAB, es imprescindible dominar las instrucciones de salto por comprobación de los bits de Acarreo y Zero, cualquier programa que se elabore estará lleno de comprobaciones de estos bits del registro de Estado.

El Registro de estado ocupa la dirección 03 Hex. del mapa de memoria.

REGISTRO DE OPCIONES DEL PIC 12CE5XX

W-1	W-1	W-1	W-1	W-1	W-1	W-1	W-1
GPWU	GPWU	TOCS	TOSE	PSA	PS2	PS1	PS0
Bit7	6	5	4	3	2	1	bit0

W = Sólo escritura
U = Sin implementar
-n = Valor al reiniciar

- bit 7: **GPWU**: Habilita despertar con los pines (GP0, GP1, GP3)
1 = Deshabilita
0 = Habilita
- bit 6: **GPPU**: Habilita debil polarización pull-ups (GP0, GP1, GP3)
1 = Deshabilita
0 = Habilita
- bit 5: **TOCS**: Selecciona entrada de reloj para Timer0
1 = Entrada pin T0CK1
0 = Entrada interna Fosc/4
- bit 4: **TOSE**: Selecciona transición para Timer0
1 = Se incrementa en la transición "1" a "0" del pin T0CK1
0 = Se incrementa en la transición "0" a "1" del pin T0CK1
- bit 3: **PSA**: Bit de asignación para el prescaler
1 = El prescaler se asigna al WDT
0 = El prescaler se asigna al Timer0
- bit 2-0: **PS2:PS0**: Bit de los valores de división
- | Valor del bit | Divisor Timer 0 | Divisor WDT |
|---------------|-----------------|-------------|
| 000 | 1:2 | 1:1 |
| 001 | 1:4 | 1:2 |
| 010 | 1:8 | 1:4 |
| 011 | 1:16 | 1:8 |
| 100 | 1:32 | 1:16 |
| 101 | 1:64 | 1:32 |
| 110 | 1:128 | 1:64 |
| 111 | 1:256 | 1:128 |

El Registro de Opciones es de escritura solamente.