

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 396 800**

51 Int. Cl.:

**G06F 9/30** (2006.01)

**G06F 9/35** (2006.01)

**G06F 9/318** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **26.11.2008 E 08857235 (9)**

97 Fecha y número de publicación de la concesión europea: **10.10.2012 EP 2215544**

54 Título: **Microprocesador o microcontrolador potenciado**

30 Prioridad:

**30.11.2007 US 991570 P**

**27.06.2008 US 147647**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

**27.02.2013**

73 Titular/es:

**MICROCHIP TECHNOLOGY INCORPORATED  
(100.0%)  
2355 WEST CHANDLER BOULEVARD  
CHANDLER, AZ 85224-6199, US**

72 Inventor/es:

**JULICHER, JOSEPH;  
SMIT, ZACHARIAS MARTHINUS;  
STEEDMAN, SEAN;  
DELPOR, VIVIEN;  
ZDENEK, JERROLD S.;  
ELLISON, RYAN SCOTT y  
SCHROEDER, ERIC**

74 Agente/Representante:

**CARPINTERO LÓPEZ, Mario**

ES 2 396 800 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

**DESCRIPCIÓN**

Microprocesador o microcontrolador potenciado

El campo técnico de la presente solicitud se refiere a microprocesadores o a microcontroladores.

5 Los microcontroladores constituyen, en general, un sistema en un chip y comprenden un microprocesador y una pluralidad de componentes periféricos. Existe una amplia variedad de tales microcontroladores que tienen una arquitectura de 8 bits, de 16 bits y de 32 bits. Los microcontroladores existentes, tal como los microcontroladores de 8 bits fabricados por Microchip Technology Inc., proporcionan una arquitectura flexible. Tales microcontroladores comprenden una arquitectura Harvard en la que las memorias de programa y de datos están separadas. Los microcontroladores del presente tipo comprenden además un sistema de almacenamiento en bancos específico que  
10 permite el acceso a la memoria de datos. En general, la memoria de datos se divide en una pluralidad de bancos y un registro de selección de banco define cual de los bancos está seleccionado y es accesible actualmente. Para acceder a otros bancos, tiene que volver a programarse el registro de selección de banco. Incluso a pesar de que un esquema de almacenamiento en bancos, por lo tanto, sólo permite el acceso a un banco de memoria definido, estos controladores incluyen unas instrucciones que fuerzan una conmutación a un banco predefinido. Esto proporciona un rendimiento mejorado y potente a pesar de las limitaciones de acceso generales.

15 No obstante, aún existen cuellos de botella en el acceso a la memoria. Por lo tanto, existe necesidad de una arquitectura de microcontrolador mejorada.

20 Por lo tanto, es un objetivo de la invención mejorar el acceso a la memoria, potenciando de este modo la funcionalidad de un microprocesador o microcontrolador. El presente objetivo puede lograrse mediante un microprocesador de n bits y un procedimiento tal como se define en las reivindicaciones independientes. Mejoras adicionales se caracterizan en las reivindicaciones dependientes.

25 De acuerdo con una realización, un dispositivo de microprocesador de n bits puede comprender una unidad central de procesamiento de n bits (CPU); una pluralidad de registros de funciones especiales y de registros de propósito general con los que se establece una correlación en memoria con una pluralidad de bancos, con por lo menos dos registros de dirección de memoria indirecta de 16 bits a los que puede acceder dicha CPU a través de todos los bancos; una unidad de acceso a banco para acoplar dicha CPU con uno de dicha pluralidad de bancos; una memoria de datos acoplada con la CPU; y una memoria de programa acoplada con la CPU, en el que dichos registros de dirección indirecta pueden accionarse para acceder a dicha memoria de datos o memoria de programa y en el que un bit en cada uno de dichos registros de dirección de memoria indirecta indica un acceso a dicha memoria de datos o a dicha memoria de programa.

30 De acuerdo con una realización adicional, n = 8. De acuerdo con una realización adicional, el microprocesador de n bits puede comprender además por lo menos un registro virtual, en el que la escritura de datos en dicho registro virtual da lugar a un acceso indirecto usando uno de dichos registros de dirección de memoria indirecta a dicha memoria de datos o a dicha memoria de programa. De acuerdo con una realización adicional, se establece una correlación del por lo menos un registro virtual con cada banco de memoria. De acuerdo con una realización adicional, el microprocesador de n bits puede comprender además un registro virtual, en el que la lectura de datos a partir de dicho registro virtual da lugar a un acceso indirecto usando uno de dichos registros de dirección de memoria indirecta a dicha memoria de datos o a dicha memoria de programa. De acuerdo con una realización adicional, puede establecerse una correlación en memoria del por lo menos un registro virtual con cada banco de memoria. De acuerdo con una realización adicional, el registro virtual no puede implementarse físicamente en dicho dispositivo de microprocesador. De acuerdo con una realización adicional, la memoria de datos puede tener una anchura de 8 bits y dicha memoria de programa puede tener una anchura de 14 bits. De acuerdo con una realización adicional, un bit adicional de una dirección de memoria de datos puede definir un intervalo de direcciones de memoria ampliado que se usa en un modo de depuración. De acuerdo con una realización adicional, el bit en dicho registro de dirección indirecta puede ser el bit más significativo.

35 De acuerdo con otra realización, un procedimiento para accionar un dispositivo de microprocesador de n bits puede comprender las etapas de: proporcionar una unidad central de procesamiento de n bits (CPU); proporcionar una pluralidad de registros de funciones especiales y de registros de propósito general con los que se establece una correlación en memoria con una pluralidad de bancos; proporcionar por lo menos dos registros de dirección de memoria indirecta de 16 bits a los que puede acceder dicha CPU a través de todos los bancos; proporcionar una unidad de acceso a banco para acoplar dicha CPU con uno de dicha pluralidad de bancos; proporcionar una memoria de datos acoplada con la CPU; proporcionar una memoria de programa acoplada con la CPU; y realizar un direccionamiento indirecto usando un registro de dirección indirecta para acceder a dicha memoria de datos o memoria de programa, en el que un bit en cada uno de dichos registros de dirección de memoria indirecta indica un acceso a dicha memoria de datos o a dicha memoria de programa, respectivamente.

45 De acuerdo con una realización adicional, n = 8. De acuerdo con una realización adicional, el procedimiento puede comprender además la etapa de generar un acceso de escritura a memoria indirecta mediante la escritura de datos en por lo menos un registro virtual, en el que el acceso a memoria indirecto usa uno de dichos registros de dirección

de memoria indirecta. De acuerdo con una realización adicional, puede establecerse una correlación en memoria del registro virtual con todos los bancos de memoria. De acuerdo con una realización adicional, el procedimiento puede comprender además la etapa de generar un acceso de lectura a memoria indirecta mediante la lectura de datos a partir de por lo menos un registro virtual, en el que el acceso de lectura a memoria indirecta usa uno de dichos registros de dirección de memoria indirecta. De acuerdo con una realización adicional, puede establecerse una correlación en memoria del registro virtual con todos los bancos de memoria. De acuerdo con una realización adicional, el registro virtual no puede implementarse físicamente en dicho dispositivo de microprocesador.

De acuerdo con una realización adicional, la memoria de datos puede tener una anchura de 8 bits y dicha memoria de programa puede tener una anchura de 14 bits. De acuerdo con una realización adicional, un bit adicional de una dirección de memoria de datos puede definir un intervalo de direcciones de memoria ampliado que se usa en un modo de depuración. De acuerdo con una realización adicional, el bit en dicho registro de dirección indirecta puede ser el bit más significativo.

Otras ventajas técnicas de la presente divulgación serán fácilmente evidentes para un experto en la técnica a partir de las siguientes figuras, descripciones y reivindicaciones. Varias realizaciones de la presente solicitud pueden obtener sólo un subconjunto de las ventajas que se exponen. Ninguna ventaja es crítica para las realizaciones.

Una comprensión más completa de la presente divulgación y de las ventajas de la misma puede adquirirse consultando la siguiente descripción tomada junto con los dibujos adjuntos, en los que números de referencia similares indican características similares, y en los que:

La figura 1 muestra un diagrama de bloques de una arquitectura de microcontrolador de acuerdo con una realización.

La figura 2 muestra una realización de un contador de programa y su acoplamiento a un bus interno.

Las figuras 3–6 muestran diferentes operaciones sobre el contador de programa dependiendo de la instrucción que se está ejecutando.

La figura 7 muestra una realización de una lógica de control y pila.

La figura 8 muestra una realización de una pila y correlación de memoria de programa.

La figura 9 muestra una realización de un registro de selección de banco.

La figura 10 muestra una realización de la estructura de un banco de memoria en la memoria de datos.

La figura 11 muestra un esquema de acceso para un acceso a memoria indirecto.

La figura 12 es una tabla que muestra el resumen de registro de funciones especiales de acuerdo con una realización.

La figura 13A y B son una tabla que muestra el conjunto de instrucciones de un microprocesador o microcontrolador de acuerdo con una realización.

A pesar de que las realizaciones de la presente divulgación se han mostrado, descrito, y se definen por referencia a las realizaciones a modo de ejemplo de la divulgación, tales referencias no implican una limitación para la divulgación, y no ha de inferirse limitación alguna de este tipo. La materia objeto que se da a conocer es capaz de considerables modificación, alteración, y equivalentes en forma y función, como se les ocurrirá a los expertos en la técnica pertinente y que tienen el beneficio de la presente divulgación. Las realizaciones que se muestran y se describen de la presente divulgación son sólo ejemplos, y no son exhaustivas del alcance de la divulgación.

La figura 1 muestra, entre otros, diferentes unidades funcionales de una unidad central de procesamiento de un microprocesador o microcontrolador relevante para acceder a la memoria de datos y de programa en una arquitectura de tipo Harvard de acuerdo con una realización. Un registro de instrucciones 110 puede almacenar una instrucción extraída que puede decodificarse mediante una lógica de control 130. Una pluralidad de registros de direccionamiento indirecto 150 puede proporcionarse, almacenando cada uno una dirección. Incluso a pesar de que se muestran por separado en la figura 1, estos y otros registros de funciones especiales son parte de un bloque de registro de funciones especiales 185. Para realizar un acceso de direccionamiento indirecto, se realiza una lectura de, o una escritura en, un registro de funciones especiales dedicado en el bloque de registro de funciones especiales 185. En el caso de que se ejecute un acceso de direccionamiento indirecto de este tipo, la lógica de control 130 controla un multiplexor 140 que recibe en el lado de entrada una dirección a partir del registro de dirección indirecta 150. De acuerdo con una realización, los registros de dirección indirecta tienen un bit predefinido, tal como el bit más significativo, el cual se decodifica mediante la lógica de control 130 para determinar si se realizará un acceso indirecto a la memoria de programa 120 o a la memoria de datos 160.

En una arquitectura de 8 bits, la memoria de datos tiene una anchura de 8 bits. De acuerdo con la arquitectura Harvard, la memoria de programa está separada de la memoria de datos y puede tener un tamaño diferente. Por ejemplo, en una realización, la memoria de programa puede tener una anchura de 14, 15 o 16 bits. Por lo tanto, pueden decodificarse unas instrucciones que incluyan, por ejemplo, una dirección directa de 8 bits. Alguna instrucción puede contener incluso más bits de dirección, que pueden combinarse con otro registro para realizar, por ejemplo, un direccionamiento relativo.

La memoria de datos 160 puede estar dividida en una pluralidad de bancos que pueden decodificarse mediante 7 u 8 bits. Cada banco puede considerarse como un archivo de registro. De acuerdo con una realización, la memoria de

datos contiene un total de 16 bancos, teniendo cada uno 256 ubicaciones de memoria direccionables lo que da como resultado una memoria total de 4.096 bytes. De acuerdo con otra realización, pueden proporcionarse 16 bancos con 128 ubicaciones de memoria direccionables. Son posibles otras configuraciones, de acuerdo con varias realizaciones. La unidad central de procesamiento puede incluir además una unidad lógica aritmética 170 (ALU) acoplada con un registro de trabajo 180. El registro de trabajo 180 es uno de una pluralidad de registros de funciones especiales del bloque 185. Tal como se ha indicado anteriormente, la memoria de datos 160 puede estar dividida en una pluralidad de bancos 1...n. Con la excepción de ciertos modos de direccionamiento, las instrucciones sólo tienen acceso, en general, al banco de memoria seleccionado. Por lo tanto, se establece una correlación en memoria de ciertos registros de funciones especiales importantes a partir del bloque de registro de funciones especiales 185 con todos los bancos de memoria 1...n tal como se indica mediante la flecha entre la memoria de datos 160 y el bloque de registro de funciones especiales 185. De acuerdo con varias realizaciones, puede establecerse una correlación en memoria de los registros de funciones especiales que no se encuentran en correlación con todos los bancos de memoria con un único banco de memoria en las diferentes ubicaciones de memoria respectivas. De acuerdo con varias realizaciones, no puede establecerse una correlación en memoria de algunos registros de funciones especiales con la memoria en absoluto.

Incluso a pesar de que la correlación de ciertos registros de funciones especiales con todos los bancos de memoria limita la RAM libre accesible en cada banco de memoria, muchas secuencias de conmutación de banco por lo demás necesarias pueden evitarse para acceder al registro de funciones especiales importantes. Por ejemplo, un microcontrolador puede tener 12 registros de funciones especiales, tal como el registro de trabajo 180 en correlación con todos los bancos de memoria. Por lo tanto, de acuerdo con una realización, cada banco de memoria puede tener 244 o 116 ubicaciones de memoria libres dependiendo del tamaño de cada banco de memoria. Son posibles otras configuraciones, de acuerdo con varias realizaciones.

De acuerdo con una realización, el microcontrolador potenciado puede contener un registro de trabajo 180 (WREG) y una ALU 170 de 8 bits. La ALU 170 puede ser una unidad aritmética de propósito general. La misma realiza funciones aritméticas y Booleanas entre los datos en el registro de trabajo 180 y cualquier archivo de registro. La ALU 170 puede tener una anchura de 8 bits y puede ser capaz de operaciones de adición, sustracción, desplazamiento y lógicas. A menos que se mencione de otro modo, las operaciones aritméticas pueden ser de complemento a dos en cuanto a su naturaleza. El WREG 180 puede ser un registro de trabajo de 8 bits que se usa para las operaciones de ALU. El registro WREG 180 es direccionable y puede ser leerse o escribirse directamente en el mismo. De acuerdo con una realización, el WREG 180 es uno de los 12 SFR comunes con los que se establece una correlación con todos los 16 bancos – su dirección puede ser, por ejemplo, 09h. La Unidad Aritmética y Lógica 170 puede ser capaz de llevar a cabo operaciones aritméticas o lógicas sobre dos operandos o sobre un único operando. Todas las instrucciones de único operando funcionan o bien sobre el registro WREG 180 o bien sobre el archivo de registro dado. Para las instrucciones de dos operandos, uno de los operandos es el registro WREG 180 y el otro es o bien un archivo de registro o bien una constante inmediata de 8 bits. Dependiendo de la instrucción ejecutada, la ALU 170 puede afectar a los valores de los bits de Acarreo (C), de Acarreo de Dígito (DC) o Cero (Z) en el registro de STATUS 190. Los bits C y DC funcionan como un bit de acarreo negativo y de acarreo negativo de dígito de orden superior, respectivamente, en una sustracción. Los ejemplos que afectan a estos bits de estatus son instrucciones tales como SUBLW y SUBWF tal como se explicará con más detalle a continuación.

El registro de STATUS 190, tal como se muestra en la figura 1 puede contener: estatus Aritmético de la ALU tal como indicador de acarreo negativo/ acarreo, de acarreo negativo/ acarreo de dígito y de cero; RP<1:0>, que representa las dos selecciones de banco más bajas; bits para el direccionamiento directo; IRP, que representa el bit de registro de selección de banco para el direccionamiento indirecto; bit de tiempo de espera y bit de apagado.

De acuerdo con una realización, el microcontrolador potenciado comprende un Registro de Selección de Banco (BSR), con el fin de mantener la compatibilidad con versiones anteriores, también los bits RP<1:0> pueden mantenerse. Puede existir una creación de reflejo bidireccional completa entre RP<1:0> y BSR<1:0>: el cambio de un registro, por lo tanto, cambia de forma automática el otro. Por lo tanto, RP<1: 0> y BSR<1:0> apuntan hacia la misma memoria física.

Incluso a pesar de que el microcontrolador potenciado contiene múltiples registros de dirección indirecta completos 150 (registros de selección de archivo (FSR)), el bit IRP puede mantenerse también para la compatibilidad con versiones anteriores – este es un reflejo bidireccional directo de FSROH<0>. El cambio de FSROH<0>, por lo tanto, cambia de forma automática el IRP, y viceversa. IRP y FSRH0h<0> apuntan hacia la misma memoria física.

De acuerdo con una realización, el registro de STATUS 190 es común a través de todos los bancos, y puede encontrarse en 03h. Un guardado de contexto rápido en caso de interrupción, tal como se explicará con más detalle a continuación, se implementa para la totalidad de los bits de STATUS, excepto TO y PD. El registro de STATUS puede ser el destino para cualquier instrucción, como cualquier otro registro. Si el registro de STATUS es el destino para una instrucción que afecte a los bits Z, DC o C, entonces estos bits se establecen o se borran de acuerdo con la lógica de dispositivo. Además, los bits TO y PD no permiten la escritura. Por lo tanto, el resultado de una instrucción con el registro de Estatus 190 como destino puede ser diferente del previsto. Por ejemplo, la instrucción CLRF STATUS borrará la totalidad de los bits excepto TO y PD, y entonces el bit Z se establecerá de forma automática mediante la lógica de dispositivo. Esto deja el registro de STATUS 190 como '000u u100' (en el que u =

sin cambios). Los bits C y DC funcionan como un bit de Acarreo Negativo y de Acarreo Negativo de Dígito de orden superior, respectivamente, en una sustracción.

El registro de Estatus 190 puede contener, por lo tanto, los siguientes bits:

- 5           IRP: bit de Selección de Banco de Registro Indirecto (Para la compatibilidad con versiones anteriores. Refleja FSR0H<0>) 1 = Banco 2, 3 (100h–1FFh); 0 = Banco 0, 1 (00h–FFh)
- RP<1:0>: bits de Selección de Banco de Registro (Para la compatibilidad con versiones anteriores. Refleja BSR<1:0>) 00 = Banco 0 (00h–7Fh); 01 = Banco 1 (80h–FFh); 10 = Banco 2 (100h–17Fh); 11 = Banco 3 (180h–1FFh)
- 10          TO: Bit de tiempo de espera (No se guarda de forma automática en caso de interrupción) 1 = Después del encendido, instrucción de CLRWDT o instrucción de SLEEP; 0 = se produjo un tiempo de espera de temporizador guardián;
- PD: Bit de apagado (No se guarda de forma automática en caso de interrupción) 1 = Después del encendido o mediante la instrucción de CLRWDT; 0 = Mediante la ejecución de la instrucción de SLEEP
- 15          Z: bit Cero: 1 = El resultado de una operación aritmética o lógica es cero; 0 = El resultado de una operación aritmética o lógica no es cero.
- DC: bit de Acarreo Negativo/ Acarreo de Dígito (instrucciones ADDWF, ADDLW, SUBLW, SUBWF): 1 = Un acarreo de orden superior a partir del 4º bit de orden bajo del resultado producido; 0 = Sin acarreo de orden superior a partir del 4º bit de orden bajo del resultado.
- 20          C: bit de Acarreo Negativo/ Acarreo (instrucciones ADDWF, ADDLW, SUBLW, SUBWF): 1 = Un acarreo de orden superior a partir del bit Más Significativo del resultado producido; 0 = Sin acarreo de orden superior a partir del bit Más Significativo del resultado producido.

Para el Acarreo Negativo, la polaridad puede invertirse. Una sustracción se ejecuta añadiendo el complemento a dos del segundo operando. Para las instrucciones de giro (RRF, RLF), este bit se carga con el bit de orden o bien bajo o bien alto del registro fuente.

- 25          El núcleo de microcontrolador potenciado puede direccionar directa o indirectamente sus archivos (bancos) de registro o la memoria de datos. De acuerdo con una realización, se establece una correlación de todos los registros de funciones especiales, incluyendo el Contador de Programa (PC) y el Registro de Trabajo (WREG), en la memoria de datos. El núcleo de microcontrolador potenciado tiene un conjunto de instrucciones ortogonal (simétrico) que hace posible llevar a cabo cualquier operación sobre cualquier registro usando cualquier modo de direccionamiento.
- 30          Esta naturaleza simétrica y carencia de ‘situaciones óptimas especiales’ hacen la programación con el núcleo de microcontrolador potenciado simple si bien eficiente. Además, la curva de aprendizaje para un usuario se reduce de forma significativa.

- 35          De acuerdo con una realización, los siguientes bloques pueden realizarse en el núcleo: contador de Programa; decodificación de Instrucción; ALU; Bloqueo/ Bloqueo de IR de memoria de sólo lectura (ROM); lógica de generación de direcciones y FSR; Registro de Trabajo; WREG; bits de Estatus en STATUS; control de vectorización de Interrupciones; con guardado de contexto rápido; el registro de selección de banco BSR.

- 40          Los siguientes bloques no pueden considerarse parte de la CPU: La pila; Circuitería de generación de restablecimiento (temporizador guardián (WDT), Restablecimiento de encendido (POR), Restablecimiento de apagamiento parcial (BOR), etc.); Memoria de Programa; Periféricos; RAM; Control de Direcciones de RAM; control y generador del reloj Q; Marcado y Habilitación de Interrupción; registros de control de interrupción; bits de Configuración; la palabra ID de Dispositivo; ubicaciones de ID; controladores de reloj.

- 45          Tal como se ha indicado anteriormente, de acuerdo con una realización, los siguientes registros pueden ser comunes a través de todos los 16 bancos: INDF0 (que se usa para el direccionamiento indirecto y no puede ser un registro físico); INDF1 (que se usa para el direccionamiento indirecto y no puede ser un registro físico); PCL (LSB de PC); STATUS; FSR0L (Dirección de memoria indirecta 0, LSB); FSR0H (Dirección de memoria indirecta 0, MSB); FSR1L (Dirección de memoria indirecta 1, LSB); FSR1H (Dirección de memoria indirecta 1, MSB); WREG, el registro de trabajo; BSR (Registro de selección de banco); PCLATH (memoria intermedia de escritura para el contador de programa PC<14:8>); registro de control de interrupción INTCON.

- 50          De acuerdo con una realización, el microprocesador o microcontrolador puede comprender además una unidad de control de interrupción 105 que puede estar acoplada con ciertos registros de funciones especiales 115. Tras la aparición de una interrupción o de cualquier otra conmutación de contexto inducida, tal como intercepciones de software, los contenidos de los siguientes registros seleccionados a partir de los registros de funciones especiales en correlación con todos los bancos de memoria se guardan de forma automática. STATUS (excepto para TO y PD); BSR, PCLATH, WREG, FSR0 (Tanto FSR0H como FSR0L), y FSR1 (Tanto FSR1H como FSR1L). Los registros se

devuelven de forma automática a sus valores previos a la interrupción tras un retorno con respecto a una interrupción.

5 Con el presente fin, tal como se muestra en la figura 1, el núcleo de microprocesador o microcontrolador puede tener por lo menos un conjunto adicional de registros 115 que almacenarán el contenido de los registros que se han mencionado anteriormente. No obstante, de acuerdo con otras realizaciones, una pila, memoria de datos adicional, etc. puede proporcionarse para almacenar el contenido de estos registros tras una conmutación de contexto.

10 Tal como se muestra en la figura 2, el contador de programa PC consiste en dos registros de 8 bits concatenados PCL 230 y PCH 240. Este direcciona las palabras en la memoria de programa, y de acuerdo con una realización, tiene, por ejemplo, una anchura de 15 bits. Por lo tanto, el mismo puede usarse para direccionar hasta 32 kW. Se establece una correlación de PCL 230, el byte bajo del PC, en la memoria de datos 160. Por lo tanto, este es común a través de todos los bancos y se encuentra, por ejemplo, en la dirección 02h. El PCL 230 es legible y permite la escritura al igual que cualquier otro registro. En cualquier instante en el que PCL 230 se modifique mediante una instrucción, ya sea una escritura en el PCL 230 o una lectura-modificación-escritura, la CPU puede necesitar ejecutar un NOP forzado con el fin de recuperar la instrucción en la nueva dirección. El PCH 240 es el byte alto del PC y no es directamente direccionable. Debido a que no puede establecerse una correlación de PCH en la memoria de programa o de datos, el registro PCLATH 220 (bloqueo alto de PC) se usa como unos bloqueos de retención para los bits altos del PC (PCH<6:0>). De acuerdo con una realización, se establece una correlación de PCLATH 220 en la memoria de datos. Por lo tanto, este es común a través de todos los bancos, y puede encontrarse en la dirección 0Ah. El PCLATH 220 se actualiza sólo mediante una carga explícita del registro. Por lo tanto, de acuerdo con una realización, ninguna otra instrucción actualiza el mismo. El microcontrolador potenciado puede estar diseñado de tal modo que la actualización del PCLATH 220 no actualiza de forma automática el PCH 240. El PCH 240 se actualiza sólo con el contenido de PCLATH 220 en las instrucciones GOTO, CALL y CALLW. El PC se incrementa en 1 después de cada extracción de instrucción en el borde de subida de Q1 a menos que: Se modifique por una instrucción de GOTO, CALL, RETURN, RETLW, RETFIE, o de Ramificación; Se modifique por una respuesta de interrupción; Debido al destino escribir en el PCL mediante una instrucción.

Los "saltos" son equivalentes a un ciclo de NOP forzado en la dirección saltada. Tal como se muestra en la figura 2, las operaciones del PC 230, 240 y el PCLATH 220 para diferentes instrucciones son tal como sigue:

30 Instrucciones de lectura en el PCL 230: PCL 230 → bus de datos 200 → ALU 170 o destino.  
 Instrucciones de escritura en el PCL 230: datos de 8 bits → bus de datos 200 → PCL 230.  
 Instrucciones de Lectura-Modificación-Escritura en el PCL 230: Cualquier instrucción que realice una operación de lectura-escritura-modificación en el PCL 230.  
 Lectura: PCL 230 → bus de datos 200 → ALU 170; Escritura: resultado de 8 bits → bus de datos → PCL 230.  
 Instrucción RETURN: TOS → PC <14:0> 230, 240.

35 La figura 3 muestra además una instrucción extraída 210 y su efecto sobre el contador de programa 230, 240. La figura 3 se aplica a las siguientes instrucciones:

40 Instrucción GOTO: Se proporciona una dirección de destino en la instrucción de 2 palabras (código de operación (*opcode*)) 210. Opcode<7:0> → PCL<7:0> 230; Opcode<10:8> → PCH<2:0> 240; PCLATH<6:3> 220 → PCH<6:3> 240.  
 Instrucción CALL: Una dirección de destino de 11 bits se proporciona en la instrucción 210. Insertar la dirección actual (PC + 1) en la pila. Opcode <7:0> → PCL <7:0> 230; Opcode <10:8> → PCH <2:0> 240; PCLATH <6:3> 220 → PCH <6:3> 240.

La figura 4 muestra la configuración para las siguientes instrucciones:

Instrucción BRA: Un desplazamiento con signo de 9 bits se proporciona en la instrucción 210. Opcode <8:0> + PC (+ 1) → PC.

45 La figura 5 muestra la configuración para las siguientes instrucciones:

Instrucción BRW: Un desplazamiento sin signo de 8 bits que se proporciona en el WREG 180. PC (+ 1) + WREG → PC.

La figura 6 muestra la configuración para las siguientes instrucciones:

50 Instrucción CALLW: Una dirección de destino de 8 bits se proporciona en el WREG 180. Insertar la dirección actual (PC + 1) en la pila. WREG <7:0> → PCL <7:0>; PCLATH <6:0> → PCH <6:0>.

55 Para habilitar un área de memoria de prueba, el contador de programa 230, 240 puede tener un 16º bit oculto, PC<15>. Cuando el PC<15> se establece, las instrucciones se extraerán de un área prueba. En el modo de usuario, PC<15> es siempre 0, y no puede modificarse en software. En el modo de prueba, PC<15> tiene un valor por defecto de 0, pero puede establecerse mediante una instrucción de "cargar configuración". El mismo puede borrarse mediante una instrucción de "Restablecer PC". El acceso a memoria de prueba puede comportarse con un

depurador en circuito (ICD) tal como sigue:

5           –ICD\_v1: PC<15> se fuerza a valor alto para extraer el vector ICD, pero vuelve a 0 después de que se ejecute la primera instrucción (GOTO ICD\_ROUTINE). Desde entonces hasta salir de la ejecución de depuración, PC<14:11> se fuerza a '1111b', pero PCLATH no se ve afectado. ICD\_v2: Si la ejecución de depuración de ICD es en la memoria de usuario, entonces esta sigue a ICD\_V1. Si la ejecución de depuración es en la memoria de prueba, entonces PC<15> se fuerza a valor alto para extraer el vector ICD, y vuelve a cero con icd\_return.

10           Los dispositivos de microcontrolador potenciado de acuerdo con varias realizaciones pueden soportar una memoria de pila de una anchura de 15 bits y una profundidad de 16 bits tal como se muestra en la figura 7. Durante el funcionamiento normal, se supone que la pila 310 tiene una profundidad de 16 palabras. En el modo normal, no se proporciona acceso de usuario directo alguno a la pila 310. La pila no es parte del espacio de memoria ni de programa o ni de datos. El PC (Contador de Programa) completo se inserta en la parte superior de la pila cuando se ejecuta una instrucción CALL o CALLW, o se confirma una solicitud de interrupción. El valor de PC se extrae de la pila con una instrucción RETURN, RETLW, o RETFIE. El PCLATH no se ve afectado por ninguna de las instrucciones de tipo CALL o RETURN.

15           La pila 310 funciona como una RAM de 16 palabras por 15 bits y un puntero a la pila de 4 bits. Incrementar/ decrementar el puntero más allá de 1111b/0000b dará lugar a que el mismo experimente un reinicio cíclico a 0000b/1111b. Durante una instrucción de tipo CALL, el puntero a la pila se incrementa y los contenidos del PC se cargan en la pila 310. Durante una instrucción de tipo RETURN los contenidos a los que se apunta se transfieren al PC y entonces el puntero a la pila se decreta. El puntero a la pila se inicializa a '1111b' después de todos los Restablecimientos. La primera llamada dará lugar a que el puntero experimente un reinicio a 0000b.

20           Un suceso de desbordamiento/ subdesbordamiento de pila tiene lugar cuando el puntero a la pila se incrementa/ decreta más allá de 1111b/0000b Y se realiza una transferencia de datos a/ desde la pila en la nueva dirección. Un suceso de desbordamiento no debe tener lugar en la primera CALL. En la 16ª llamada consecutiva, el puntero a la pila tendrá un valor de 1111b. La 17ª CALL consecutiva dará lugar a que el puntero a la pila experimente un reinicio cíclico a 0000b, y a que el PC se cargue en esta posición. Lo anterior constituye un suceso de desbordamiento. Cuando se han extraído elementos de la pila bastantes veces para alcanzar 0000b, las extracciones adicionales darán lugar a que los contenidos en 0000b se transfieran al PC, y a que el puntero a la pila se decreta. El puntero a la pila experimentará un reinicio a 1111b, lo anterior NO constituye un suceso de subdesbordamiento. Sólo con la siguiente extracción consecutiva, cuando se intente una transferencia de datos a partir de 1111b, tendrá lugar un suceso de subdesbordamiento. En el modo de usuario, un desbordamiento o subdesbordamiento de la pila establecerá el bit apropiado (STOF o STUF) en el registro de PC, y dará lugar a un restablecimiento de software. – –

30           Cuando se está en el modo de ICD, la pila puede tener dos segmentos. La pila de modo de usuario normal tiene aún una profundidad de 16 palabras, pero puede proporcionarse también una pila de ICD separada. En el modo de ICD será posible acceder a la parte de arriba de la pila y al puntero a la pila. En el modo de ICD, la parte de arriba de la pila (TOS) es legible y permite la escritura. Dos ubicaciones de registro, TOSH y TOSL, direccionarán la ubicación de RAM de la pila a la que apunta el puntero a la pila (STKPTR). Después de una instrucción de tipo CALL, el software puede leer el valor insertado mediante la lectura de los registros TOSH y TOSL. Estos valores pueden colocarse sobre una pila de software definida por el usuario. La escritura en los registros TOS dará lugar a que el PC se actualice con el nuevo valor con la siguiente instrucción de tipo RETURN.

35           El registro STKPTR contiene el valor del puntero a la pila. Tal como se ha indicado anteriormente, STKPTR es accesible sólo en el modo de ICD. En el modo de ICD, puede accederse al puntero a la pila a través del registro STKPTR. El usuario puede leer y escribir los valores del puntero a la pila. El puntero a la pila se inicializa a '1111b' después de todos los Restablecimientos. En el modo de ICD, un restablecimiento de software automático no se aplicará tras un suceso de desbordamiento/ subdesbordamiento. No obstante, aún se establecerá el bit de desbordamiento (STOF) o de subdesbordamiento (STUF) de la pila en el registro de PC, lo que permite la verificación por software de un estado de pila. Debido a que un restablecimiento no se ejecutará tras un subdesbordamiento/ desbordamiento, el puntero a la pila (STKPTR) no se restablecerá. Tras un desbordamiento/ subdesbordamiento, el puntero a la pila experimentará un reinicio cíclico y entonces reanudará el funcionamiento normal. El usuario tiene que borrar los bits de desbordamiento (STOF) o de subdesbordamiento (STUF) de forma explícita – y estos se establecerán de nuevo de forma automática tras desbordamientos/ subdesbordamientos posteriores. Los sucesos de Desbordamiento/ Subdesbordamiento de pila sólo pueden estar producidos por una instrucción que modifique el puntero a la pila usando el hardware del puntero a la pila.

40           Esto incluye: CALL, TRAP; RETURN, RETFIE, RETLW; Cualquier intercepción de ICD y interrupción. Las instrucciones que modifiquen el puntero a la pila a través de la ALU (sólo en modo de ICD) no darán lugar al estado de desbordamiento/ subdesbordamiento de pila, y por lo tanto no establecerán los bits de (STOF) o de (STUF). Son ejemplos (sólo en modo de ICD): INCF STKPTR; STKPTR DECF; ADDWF STKPTR.

45           Cuando un dispositivo se restablece, el PC se carga con el vector de Restablecimiento (Oh). El puntero a la pila se inicializa a '1111b', y la Parte de Arriba del registro de la Pila (TOS) es '0000h'. Una segunda inserción incrementa el puntero a la pila y entonces carga el PC actual en el nivel de la pila. En la 16ª llamada consecutiva, el puntero a la

pila tendrá un valor de 1111b. La 17ª CALL consecutiva dará lugar a que el puntero a la pila experimente un reinicio cíclico a 0000b, y a que el PC se cargue en esta posición. Lo anterior constituye un suceso de desbordamiento. Una extracción RETURN copiará los contenidos de la pila a los que se apunta en el PC y entonces decrementará el puntero a la pila. Cuando se han extraído elementos de la pila bastantes veces para alcanzar 0000b, las extracciones adicionales darán lugar a que los contenidos en 0000b se transfieran al PC, y a que el puntero a la pila se decremente. El puntero a la pila experimentará un reinicio a 1111t, lo anterior NO constituye un suceso de subdesbordamiento. Sólo con la siguiente extracción consecutiva, cuando se intente una transferencia de datos a partir de 1111b, tendrá lugar un suceso de subdesbordamiento. En el modo de depuración (ICD), una instrucción especial dará lugar a que el STKPTR se incremente. El PC no se carga en TOS. Corresponde al usuario garantizar que el TOS se carga con los datos apropiados antes de que se ejecute la instrucción de INCF STKPTR (esta instrucción es, por lo tanto, equivalente a una instrucción PUSH. La instrucción de STKPTR DECF decrementará el puntero a la pila, el PC no se carga con el valor de TOS.

Tal como se muestra en la figura 8, el contador de programa de 15 bits 400 es capaz de direccionar un espacio de memoria de programa de 32k x 14 bits. El espacio de memoria de programa 430 contiene principalmente instrucciones para su ejecución; no obstante, pueden almacenarse y accederse a tablas de datos asimismo (usando una infraestructura de lectura/ escritura de memoria de programa heredada, así como la nueva funcionalidad asociada con el direccionamiento indirecto, véase la sección de FSR para más detalle). Existe también un 16º bit de PC oculto durante el funcionamiento normal, y cuando se establece, es posible direccionar otros 32k x 14 de memoria reservada para los bits de configuración, la ID de Dispositivo y la memoria de prueba 440. Este bit sólo se establece en el modo de prueba o en el modo de programación. Tal como se muestra en la figura 8, el vector de Restablecimiento se encuentra en '0000h' y el vector de interrupción se encuentra en '0004h'.

De acuerdo con una realización, el microcontrolador potenciado puede direccionar una memoria de datos 160 y una RAM de propósito general de hasta 2.048 bytes. El bus de dirección de memoria de datos tiene una anchura de 11 bits. La memoria de datos está dividida en 16 bancos de 128 bytes cada uno, que contiene los Registros de Propósito General (GPR) y Registros de funciones especiales (SFR). El banco se selecciona mediante el registro de selección de banco (BSR<3:0>) tal como se muestra en la figura 9. El registro BSR puede acceder a más de 16 bancos; no obstante, lo anterior puede permitirse sólo en el modo de ICD para permitir un espacio de datos más grande para los registros de ICD. (En el modo de ICD, puede accederse a 32 bancos). Se establece también una correlación de los bits BSR0 y BSR1 en el registro de STATUS como RP0 y RP1, respectivamente. Lo anterior es para permitir la compatibilidad con versiones anteriores de los dispositivos existentes. Una lectura de, o una escritura en, BSR0/1 afectará a RP0/1 y viceversa. La figura 10 muestra una correlación en memoria de datos a modo de ejemplo.

Cuando el núcleo está funcionando con un módulo de ICD o en otra realización, el número total de bancos puede ampliarse a 32 para 4.096 direcciones totales. Lo anterior es para permitir que se establezca una correlación de los registros de ICD en el espacio de datos y que no se use correlación en memoria del usuario alguna. Lo anterior sólo se habilitará cuando el módulo de ICD esté habilitado. El usuario no tendrá acceso a banco alguno más grande que el Banco 15. Son posibles otras configuraciones, de acuerdo con varias realizaciones.

La figura 11 muestra una realización de un esquema de direccionamiento indirecto que es un modo de direccionamiento de la memoria de datos en el que la dirección de memoria de datos en la instrucción se determina mediante otro registro. Lo anterior puede ser útil para pilas o tablas de datos en la memoria de datos/ de programa. El valor del registro 960 del registro de selección de archivo (FSR) se usa como la dirección de memoria de datos. El microcontrolador potenciado tiene dos pares de registros de 16 bits 960 para el direccionamiento indirecto. Estos así denominados pares de registro de selección de archivo son: FSR0H y FSR0L; y FSR1H y FSR1L. Los FSR 960 son registros de 16 bits y por lo tanto permiten el direccionamiento de hasta 65.536 ubicaciones. Un único bit 965 en estos registros de selección de archivo 960 puede usarse para determinar si se realizará un acceso a la memoria de datos 160 o a la memoria de programa 120. Con el presente fin, la unidad lógica 930 retransmite el contenido del registro de selección de archivo 960 o bien a la dirección de memoria de programa 910 o bien a la dirección de RAM 920 para acceder a las memorias respectivas.

El direccionamiento no indirecto usa el contenido respectivo "Archivo" de una instrucción extraída 950 a través del multiplexor 940. Los accesos a banco se realizan entonces a través de la dirección de RAM 920 en la que la dirección de banco se proporciona mediante el registro de acceso a banco respectivo para formar una dirección completa 920. Tal como se muestra en la figura 11, los registros de selección de archivo 960 se proporcionan para el direccionamiento indirecto. Para el direccionamiento indirecto, la salida del multiplexor 940 proporciona la dirección del "Archivo" a través del contenido del registro de selección de archivo respectivo 960. Si se selecciona el direccionamiento indirecto, el bit 15 que se indica mediante el número 965 de cada registro de selección de archivo 960 se usa para determinar si se realizará un acceso a la memoria de programa 120 a través de la dirección de memoria de programa 910 o a la memoria de datos 160 a través de la dirección de datos 920. El direccionamiento indirecto se inicia a través de una señal de selección respectiva (que no se muestra) que controla el multiplexor 940.

Cuando FSRn<15> es igual a 0, se direcciona la memoria de datos 160. Cuando FSRn<15> es 1, se direcciona la memoria de programa 120. Se establece una correlación de la memoria de datos 160 en la primera mitad de este espacio de direcciones (0000h a 7FFFh). Si FSRn<15> es 0, FSRn<10:0> apunta hacia las ubicaciones de memoria



de datos 000h a 7FFh. En el presente caso, se ignora FSRn<11:14>. Se establece una correlación de la memoria de programa 120 en la mitad superior de este espacio de direcciones (8000h a FFFFh). Si FSRn<15> es 1, entonces FSRn<14: 0> apunta hacia las direcciones de memoria de programa 0000h a 7FFFh. En resumen: Las direcciones 8000h y más altas apuntan a la memoria de programa. Las direcciones por debajo de 8000h apuntan a la memoria de datos, usándose sólo los 11 bits más bajos para direccionar la decodificación.

Además, existen unos registros virtuales INDF0 y INDF1 (Véase la figura 10), que no están implementados físicamente. La lectura de, o la escritura en, estos registros activa el direccionamiento indirecto, siendo el valor en el registro FSR correspondiente la dirección de los datos. Si el propio INDF0 (o INDF1) del archivo se lee indirectamente a través de un FSR, todos los '0' se leen (el bit Cero está establecido). De forma similar, si se escribe indirectamente en INDF0 (o INDF1), la operación será equivalente a un NOP, y los bits de STATUS no se ven afectados. Las escrituras indirectas en el espacio de memoria de programa (FSRn<15> = 1) no tendrán efecto, y conducirán a que se ejecute el equivalente de un NOP. Las lecturas indirectas a partir del espacio de memoria de programa (FSRn<15> = 1) darán lugar a que los primeros 8 bits de la ubicación de memoria de programa se transfieran al registro de destino, el EEDATH/PMDATH se actualizará con los 6 bits superiores. De acuerdo con varias realizaciones, puede implementarse otro mecanismo para inducir el direccionamiento indirecto.

La figura 12 muestra un resumen de todos los registros de funciones especiales de acuerdo con una realización, tal como se usa en cualquier banco n de la memoria de datos.

Las figuras 13A y B muestran el conjunto de instrucciones en el que se explican con detalle a continuación las instrucciones potenciadas usando la arquitectura tal como se ha explicado anteriormente.

DDFSR Añadir Literal a FSRn: Sintaxis: ADDFSR n, k  
Operandos:  $-32 \leq k \leq 31$

$$n \in [0,1]$$

Operación:  $FSR(n) + k \rightarrow FSR(n)$

Estatus Afectados: Ninguno

Codificación: 11 0001 0nkk kkkk

Descripción: El literal de 6 bits con signo 'k' se añade a los contenidos del FSR especificado por 'n'. FSRn está limitado al intervalo 0000h –FFFFh.

Incrementar/ decrementar el mismo más allá de estos límites dará lugar a que el mismo experimente un reinicio cíclico. Obsérvese que las direcciones 8000h y más grandes apuntan a la memoria de programa. Las direcciones por debajo de 8000h apuntan a la memoria de datos, usándose sólo los 11 bits más bajos para direccionar la decodificación.

Palabras: 1

Ciclos: 1

Actividad de Ciclo Q:

Q1	Q2	Q3	Q4
Decodificar	Leer FSRn'	Procesar los Datos	Escribir en el FSLN
Ejemplo:	ADDFSR 1, 23h		
	Antes de la Instrucción FSR1 = 0FFh		
	Después de la Instrucción FSR1 = 0122h		

DDWFC Añadir WREG y bit de CARRY (acarreo) a f: Sintaxis: ADDWFC f {,d}

Operandos:  $0 \leq f \leq 127$

$$d \in [0,1]$$

Operación:  $(WREG) + (f) + (C) \rightarrow \text{dest}$

Estatus Afectados: C, DC, Z

Codificación: 11 1101 dfff ffff

Descripción: Añadir WREG, el indicador de CARRY y la ubicación de memoria de datos 'f'. Si 'd' es '0', el resultado se coloca en el WREG. Si 'd' es '1', el resultado se coloca en la ubicación de memoria de datos 'f'.

Ciclos: 1

Actividad de Ciclo Q:

Q1	Q2	Q3	Q4
Decodificar	Leer reg 'f'	Procesar los Datos	Escribir en el destino
Ejemplo:	ADDWFC REG, 0		
	Antes de la Instrucción	bit de CARRY = 1	
		REG = 02h	
		WREG = 4Dh	
	Después de la Instrucción	bit de CARRY = 0	
		REG = 02h	
		WREG = 50h	

Desplazamiento Aritmético a la Derecha de SRF: Sintaxis: ASRF f {,d}

Operandos:  $0 \leq f \leq 127$

(continuación)

5	<u>Desplazamiento Aritmético a la Derecha de SRF</u> : Sintaxis: ASRF f {,d}			
		$d \in [0,1]$		
	Operación:	$(f<7>) \rightarrow \text{dest}<7>$ $(f<7:1>) \rightarrow \text{dest}<6:0>$ , $(f<0>) \rightarrow C$ ,		
	Estatus Afectados:	C, Z		
10	Codificación:	11 0111 dfff ffff		
	Descripción: Los contenidos del registro 'f' se desplazan un bit a la derecha a través del indicador de CARRY. El MSb permanece sin cambios. Si 'd' es '0', el resultado se coloca en el WREG. Si 'd' es '1', el resultado se vuelve a almacenar en el registro 'f'.			
	Palabras: 1			
15	Ciclos: 1			
	Actividad de Ciclo Q:			
	Q1	Q2	Q3	Q4
	Decodificar	Leer reg 'f'	Procesar los Datos	Escribir en el destino
	Ejemplo:	ASRF REG, 0		
20	Antes de la Instrucción		REG = 1110 0110 C = 1	
	Después de la Instrucción		REG = 1110 0110 WREG = 1111 0011 C = 0	
25	<u>Una Ramificación Relativa</u> : Sintaxis: BRA k			
	Operandos: $-256 \leq k \leq 255$			
	Operación: $(PC) + k \rightarrow PC$			
	Estatus Afectados: Ninguno			
	Codificación: 11 001k kkkk kkkk			
30	Descripción: Añadir el número del complemento a 2 'k' al PC. Debido a que el PC se habrá incrementado para extraer la siguiente instrucción, la nueva dirección será $PC + 1 + k$ . Esta instrucción es una instrucción en dos ciclos.			
	Palabras: 12			
	Actividad de Ciclo Q:			
35	Q1	Q2	Q3	Q4
	Decodificar	Leer literal 'k'	Procesar los Datos	Escribir en PC
	Sin operación	Sin operación	Sin operación	Sin operación
	Ejemplo: 0023h BRA 005h			
	Antes de la Instrucción		PC = dirección 0023h)	
40	Después de la Instrucción		PC = dirección 0029h)	
	<u>Ramificación Relativa a RW con WREG</u> : Sintaxis: BRW			
	Operandos: Ninguno			
	Operación: $(PC) + (WREG) \rightarrow PC$			
	Estatus Afectados: Ninguno			
45	Codificación: 00 0000 0000 1011			
	Descripción: Añadir los contenidos de WREG (sin signo) al PC. Debido a que el PC se habrá incrementado para extraer la siguiente instrucción, la nueva dirección será $PC + 1 + (WREG)$ . Esta instrucción es una instrucción en dos ciclos. Los contenidos de WREG se tratan como un número sin signo.			
	Palabras: 1			
50	Ciclos: 2			
	Actividad de Ciclo Q:			
	Q1	Q2	Q3	Q4
	Decodificar	Leer reg WREG	Procesar los Datos	Escribir en PC
	Sin operación	Sin operación	Sin operación	Sin operación
55	Ejemplo: 0024h BRW		Antes de la Instrucción PC = dirección 0024h) WREG = 85h	
			Después de la Instrucción PC = dirección 00AAh)	
	<u>ALLW Llamada de Subrutina Con WREG</u> : Sintaxis: CALLW			
60	Operandos: Ninguno			
	Operación: $(PC) + 1 \rightarrow TOS$ , $(WREG) \rightarrow PC<7:0>$ , $(PCLATH<6:0>) \rightarrow PC<14:8>$			
	Estatus Afectados: Ninguno			
65	Codificación: 00 0000 0000 1010			

(continuación)

ALLW Llamada de Subrutina Con WREG: Sintaxis: CALLW

5 Descripción: llamada de Subrutina con WREG. En primer lugar, la dirección de retorno (PC + 1) se inserta en la parte superior de la pila de retorno. Entonces, los contenidos de WREG se cargan en PC<7:0>, y los contenidos de PCLATH en PC<14:8>. CALLW es una instrucción en dos ciclos.

Palabras: 1

Ciclos: 2

Actividad de Ciclo Q:

10	Q1	Q2	Q3	Q4
	Decodificar	Leer el registro WREG	Procesar los Datos	Escribir en PC
	Sin operación	Sin operación	Sin operación	Sin operación
	Ejemplo: 0025h CALLW			

15	Antes de la Instrucción	PC = dirección (0025h) WREG = 1Ah PCLATH = 10h
20	Después de la Instrucción	TOS = dirección (0026h) PC = 101Ah WREG = 1Ah PCLATH = 10h

Desplazamiento Lógico a la Izquierda de SLF: Sintaxis: LSLF f {,d}

Operandos:  $0 \leq f \leq 127$  d  
Operación: (f<7>) → C  
(f<6:0>) → dest<7:1>  
0 → dest<0>

25 Estatus Afectados: C, Z  
Codificación: 11 0101 dfff ffff  
30 Descripción: Los contenidos del registro 'f' se desplazan un bit a la izquierda a través del indicador de CARRY. Un '0' se desplaza al LSb. Si 'd' es '0', el resultado se coloca en el WREG. Si 'd' es '1', el resultado se vuelve a almacenar en el registro 'f'.

Palabras: 1

Ciclos: 1

Actividad de Ciclo Q:

35	Q1	Q2	Q3	Q4
	Decodificar	Leer reg 'f'	Procesar los Datos	Escribir en el destino
	Ejemplo: LSLF REG, 0			

40	Antes de la Instrucción	REG = 1110 0110 C = 0
	Después de la Instrucción	REG = 1110 0110 WREG = 1100 1100 C = 1

Desplazamiento Lógico a la Derecha de SRF: Sintaxis: LSRF f {,d}

Operandos:  $0 \leq f \leq 127$   
 $d \in [0,1]$   
Operación: 0 → dest<7>  
(f<7:1>) → dest<6:0>,  
(f<0>) → C,

50 Estatus Afectados: C, Z  
Codificación: 11 0110 dfff ffff  
Descripción: Los contenidos del registro 'f' se desplazan un bit a la derecha a través del indicador de CARRY. Un '0' se desplaza al MSb. Si 'd' es '0', el resultado se coloca en el WREG. Si 'd' es '1', el resultado se vuelve a almacenar en el registro 'f'.

Palabras: 1

Ciclos: 1

Actividad de Ciclo Q:

55	Q1	Q2	Q3	Q4
	Decodificar	Leer el registro 'f'	Procesar los Datos	Escribir en el destino
	Ejemplo: LSRF REG, 0			

60	Antes de la Instrucción	REG = 1110 0110 C = 1
	Después de la Instrucción	REG = 1110 0110 WREG = 0111 0011 C = 0

OVIW Mover INDFn a WREG, con pre-/ post-incremento/ decremento

65 Sintaxis: MOVIW ++/--FSRn++/--

$n \in [0,1]$

(continuación)

OVIW Mover INDFn a WREG, con pre-/ post-incremento/ decremento

5 Operandos:  
 Operación: INDFn → WREG  
 Aplicar una operación de pre-/ post-incremento/ decremento a FSRn.  
 Estatus Afectados: Z (Sólo si el destino es WREG)  
 Codificación: 00 0000 0001 0nmm  
 10 MM Operación  
 00 ++FSLN  
 01 --FSRn  
 10 FSRn++  
 11 FSRn--  
 15 Descripción: Esta instrucción se usa para mover datos entre uno de los registros indirectos (INDFn) y WREG. Antes/ después de este movimiento, el puntero (FSRn) se actualiza pre-/ post-incrementando/ decrementando el mismo. FSRn está limitado al intervalo 0000h – FFFFh. Incrementar/ decrementar el mismo más allá de estos límites dará lugar a que el mismo experimente un reinicio cíclico. Obsérvese que las direcciones más grandes que 7FFFh apuntan a la memoria de programa. Las direcciones por debajo de 8000h apuntan a la memoria de datos, usándose sólo los 11 bits más bajos para direccionar la decodificación. La operación de incremento/ decremento sobre FSRn NO afectará a bit alguno de STATUS. Esta instrucción sólo puede afectar al indicador Z si se mueve un valor de 00h a WREG.  
 Palabras: 1  
 Ciclos: 1

25 Actividad de Ciclo Q:  
 Q1 Decodificar  
 Q2 Actualizar el puntero si se Lee el registro Fuente.  
 Q3 Procesar los Datos  
 Q4 Escribir en el registro de Destino. Actualizar el puntero si fuera necesario.

Ejemplo: MOVIW FSR0++  
 Antes de la Instrucción FSR0 = 22h  
 WREG = FFh  
 (22h) = 12h  
 (23h) = 33h  
 Después de la Instrucción FSR0 = 23h  
 WREG = 12h  
 (22h) = 12h  
 (23h) = 33h

Ejemplo: MOVIW ++FSR0  
 Antes de la Instrucción FSR0 = 22h  
 WREG = FFh  
 (22h) = 12h  
 (23h) = 33h  
 Después de la Instrucción FSR0 = 23h  
 WREG = 33h  
 (22h) = 12h  
 (23h) = 33h

OVIW Mover INDFn a WREG. Direccionamiento Indirecto Indizado

Sintaxis: MOVIW [k]FSRn  
 Operandos:  $-32 \leq k \leq 31$   
 $n \in [0,1]$   
 Operación: (FSRn + k) → WREG  
 50 Mover datos entre la ubicación a la que apunta FSRn + k y WREG.  
 Estatus Afectados: Z (Sólo si el destino es WREG)  
 Codificación: 11 1111 0nkk kkkk  
 Descripción: Esta instrucción se usa para mover datos entre una ubicación a la que apunta FSRn + k y WREG. FSRn no se actualiza con k. El intervalo direccionable está limitado a 0000h – FFFFh. Indizar más allá de estos límites dará lugar a un reinicio cíclico de la dirección. Obsérvese que las direcciones 8000h y más altas apuntan a la memoria de programa. Las direcciones por debajo de 8000h apuntan a la memoria de datos, usándose sólo los 11 bits más bajos para direccionar la decodificación. Direccionar el cálculo (la adición de k a FSRn) NO afectará a bit alguno de STATUS. Esta instrucción sólo puede afectar al indicador Z si se mueve un valor de 00h a WREG.  
 Palabras: 1  
 Ciclos: 1

(continuación)

OVIW Mover INDFn a WREG. Direccionamiento Indirecto Indizado

Actividad de Ciclo Q:

5	Q1 Decodificar	Q2 Direccionar el cálculo. Leer el registro Fuente.	Q3 Procesar los Datos	Q4 Escribir en el registro de Destino.
---	-------------------	---	--------------------------	---

Ejemplo: MOVIW [5]FSRO

Antes de la Instrucción FSR0 = 22h  
WREG = FFh  
(22h) = 12h  
(27h) = 39h

10	Después de la Instrucción	FSR0 = 22h WREG = 39h (22h) = 12h (27h) = 39h
----	---------------------------	--

Ejemplo: MOVIW [-5]FSRO

Antes de la Instrucción FSR0 = 22h  
WREG = FFh  
(1Dh) = 17h  
(22h) = 12h

20	Después de la Instrucción	FSR0 = 22h WREG = 17h (1Dh) = 17h (22h) = 12h
----	---------------------------	--

OVLB Mover literal a BSR: Sintaxis: MOVLB k

Operandos:  $0 \leq k \leq 15$

Operación:  $k \rightarrow \text{BSR}$

Estatus Afectados: Ninguno

Codificación: 00 0000 0010 kkkk

Descripción: El literal de cuatro bits 'k' se carga en el Registro de Selección de Banco (BSR).

Palabras: 1

Ciclos: 1

Actividad de Ciclo Q:

30	Q1 Decodificar	Q2 Leer literal 'k'	Q3 Procesar los Datos	Q4 Escribir literal 'k' en BSR
----	-------------------	------------------------	--------------------------	-----------------------------------

OVL P Mover literal a PCLATH: Sintaxis: MOVLP k

Operandos:  $0 \leq k \leq 127$

Operación:  $k \rightarrow \text{PCLATH}$

Estatus Afectados: Ninguno

Codificación: 11 0001 1kkk kkkk

Descripción: El literal de siete bits 'k' se carga en el registro de PCLATH.

Palabras: 1

Ciclos: 1

Actividad de Ciclo Q:

40	Q1 Decodificar	Q2 Leer literal 'k'	Q3 Procesar los Datos	Q4 Escribir literal 'k' en PCLATH
----	-------------------	------------------------	--------------------------	--------------------------------------

OVWI Movimiento Indirecto Indizado: Sintaxis: MOVWI [k]FSRn

Operandos:  $-32 \leq k \leq 31$

$$n \in [0,1]$$

Operación:  $\text{WREG} \rightarrow (\text{FSRn} + k)$

Mover datos entre el WREG y la ubicación a la que apunta  $\text{FSRn} + k$ .

Estatus Afectados: Z (Sólo si el destino es WREG)

Codificación: 11 1111 1nkk kkkk

Descripción: Esta instrucción se usa para mover datos entre el WREG y la ubicación a la que apunta  $\text{FSRn} + k$ .  $\text{FSRn}$  no se actualiza con  $k$ . El intervalo direccionable está limitado a  $0000\text{h} - \text{FFFFh}$ . Indizar más allá de estos límites dará lugar a un reinicio cíclico de la dirección. Obsérvese que las direcciones  $8000\text{h}$  y más altas apuntan a la memoria de programa. Las direcciones por debajo de  $8000\text{h}$  apuntan a la memoria de datos, usándose sólo los 11 bits más bajos para direccionar la decodificación. Direccionar el cálculo (la adición de  $k$  a  $\text{FSRn}$ ) NO afectará a bit alguno de STATUS.

Palabras: 1

Ciclos: 1

(continuación)

OVWI Movimiento Indirecto Indizado: Sintaxis: MOVWI [k]FSRn

Actividad de Ciclo Q:

5	Q1 Decodificar	Q2 Direccionar el cálculo. Leer el registro Fuente.	Q3 Procesar los Datos	Q4 Escribir en el registro de Destino.
---	-------------------	---	--------------------------	---

Ejemplo: MOVWI [3]FSRO

10	Antes de la Instrucción	FSR0 = 22h WREG = FFh (22h) = 12h (25h) = 56h
	Después de la Instrucción	FSR0 = 22h WREG = FFh (22h) = 12h (25h) = FFh

Ejemplo: MOVWI [-1]FSRO

15	Antes de la Instrucción	FSR0 = 22h WREG = FFh (21h) = 56h (22h) = 12h
20	Después de la Instrucción	FSR0 = 22h WREG = FFh (21h) = FFh (22h) = 12h

Actividad de Ciclo Q:

25	Q1 Decodificar	Q2 Leer el registro 'f'	Q3 Procesar los Datos	Q4 Escribir en el Destino
----	-------------------	----------------------------	--------------------------	------------------------------

OVWI Mover WREG a INDFn, con pre-/ post-incremento/ decremento

Sintaxis: MOVWI ++/--FSRn++/--

Operandos:

$$n \in [0,1]$$

Operación: WREG → INDFn  
Aplicar una operación de pre-/ post-incremento/ decremento a FSR usado.  
Estatus Afectados: Z (Sólo si el destino es WREG)  
Codificación: 00 0000 0001 1nmm

35	MM	Operaciones
	00	++FSLN
	01	--FSRn
	10	FSRn++
	11	FSRn--

Descripción: Esta instrucción se usa para mover datos entre el WREG y uno de los registros indirectos (INDFn). Antes/ después de este movimiento, el puntero (FSRn) se actualiza pre-/ post-incrementando/ decrementando el mismo. FSRn está limitado al intervalo 0000h – FFFFh. Incrementar/ decrementar el mismo más allá de estos límites dará lugar a que el mismo experimente un reinicio cíclico. Obsérvese que las direcciones 8000h y más altas apuntan a la memoria de programa. Las direcciones por debajo de 8000h apuntan a la memoria de datos, usándose sólo los 11 bits más bajos para direccionar la decodificación. La operación de incremento/ decremento sobre FSRn NO afectará a bit alguno de STATUS.

Palabras: 1

Ciclos: 1

Actividad de Ciclo Q:

50	Q1 Decodificar	Q2 Actualizar el puntero si fuera necesario.	Q3 Procesar los Datos	Q4 Escribir en INDFn. Actualizar el puntero si fuera necesario.
----	-------------------	---	--------------------------	---

Ejemplo: MOVWI FSR0--

55	Antes de la Instrucción	FSR0 = 22h WREG = FFh (21h) = 56h (22h) = 12h
	Después de la Instrucción	FSR0 = 21 h WREG = FFh (21h) = 56h (22h) = FFh



**REIVINDICACIONES**

1. Un dispositivo de microprocesador de n bits que comprende:

una unidad central de procesamiento de n bits (CPU);  
 una pluralidad de registros (185) de funciones especiales y de registros de propósito general con los que se establece una correlación en memoria con una pluralidad de bancos, en el que los registros (185) de funciones especiales comprenden por lo menos dos registros (150; 960) de dirección de memoria indirecta de 16 bits a los que puede acceder dicha CPU a través de todos los bancos;  
 una unidad de acceso a banco para acoplar dicha CPU con uno de dicha pluralidad de bancos;  
 una memoria (160) de datos acoplada con la CPU; y  
 una memoria (120) de programa acoplada con la CPU, en el que dichos registros (150; 960) de dirección de memoria indirecta pueden accionarse para acceder a dicha memoria (160) de datos o memoria (120) de programa y en el que un bit (965) en cada uno de dichos registros (150; 960) de dirección de memoria indirecta determina un acceso a dicha memoria (160) de datos o a dicha memoria (120) de programa.

2. El microprocesador de n bits de acuerdo con la reivindicación 1, en el que n = 8.

3. El microprocesador de n bits de acuerdo con la reivindicación 1 o 2, que además comprende por lo menos un registro (INDF0; INDF1) de funciones especiales que no está implementado físicamente, en el que la escritura de datos en dicho registro (INDF0; INDF1) de funciones especiales da lugar a un acceso indirecto usando uno de dichos registros (150; 960) de dirección de memoria indirecta a dicha memoria (160) de datos o a dicha memoria (120) de programa.

4. El microprocesador de n bits de acuerdo con una de las reivindicaciones anteriores, que además comprende un registro (INDF0; INDF1) de funciones especiales que no está implementado físicamente, en el que la lectura de datos a partir de dicho registro (INDF0; INDF1) de funciones especiales da lugar a un acceso indirecto usando uno de dichos registros (150; 960) de dirección de memoria indirecta a dicha memoria (160) de datos o a dicha memoria (120) de programa.

5. El microprocesador de n bits de acuerdo con la reivindicación 3 o 4, en el que se establece una correlación del por lo menos un registro (INDF0; INDF1) de funciones especiales con cada banco de memoria.

6. El microprocesador de n bits de acuerdo con una de las reivindicaciones anteriores 3–5, en el que un acceso a memoria indirecto a dicha memoria de programa usando uno de dichos registros (150; 960) de dirección de memoria indirecta lee los n bits más bajos de una ubicación de memoria de programa.

7. El microprocesador de n bits de acuerdo con una de las reivindicaciones anteriores, en el que dicha memoria (160) de datos tiene una anchura de 8 bits y dicha memoria (120) de programa tiene una anchura de 14 bits.

8. El microprocesador de n bits o el procedimiento de acuerdo con la reivindicación 7, en el que un bit adicional de una dirección de memoria de datos define un intervalo de direcciones de memoria ampliado que se usa en un modo de depuración.

9. El microprocesador de n bits o el procedimiento de acuerdo con la reivindicación 8, en el que dicho bit adicional en dicho registro (150; 960) de dirección indirecta es el bit más significativo.

10. Un procedimiento para accionar un dispositivo de microprocesador de n bits que comprende

una unidad central de procesamiento de n bits (CPU);  
 una pluralidad de registros (185) de funciones especiales y de registros de propósito general con los que se establece una correlación en memoria con una pluralidad de bancos, en el que los registros de funciones especiales comprenden por lo menos dos registros (150; 960) de dirección de memoria indirecta de 16 bits a los que puede acceder dicha CPU a través de todos los bancos;  
 una unidad de acceso a banco para acoplar dicha CPU con uno de dicha pluralidad de bancos;  
 una memoria (160) de datos acoplada con la CPU;  
 una memoria (120) de programa acoplada con la CPU; en el que el procedimiento comprende:

realizar un direccionamiento indirecto usando un registro (150; 960) de dirección indirecta para acceder a dicha memoria (160) de datos o memoria (120) de programa, en el que un bit (965) en cada uno de dichos registros (150; 960) de dirección de memoria indirecta indica un acceso a dicha memoria (160) de datos o a dicha memoria (120) de programa, respectivamente.

11. El procedimiento de acuerdo con la reivindicación 10, en el que n = 8.

12. El procedimiento de acuerdo con la reivindicación 10 u 11, que además comprende la etapa de generar un acceso de escritura a memoria indirecta mediante la escritura de datos en por lo menos un registro (INDF0; INDF1) de funciones especiales que no está implementado físicamente, en el que el acceso a memoria indirecto usa uno de



dichos registros (150; 960) de dirección de memoria indirecta.

5 13. El procedimiento de acuerdo con una de las reivindicaciones anteriores 10–12, que además comprende la etapa de generar un acceso de lectura a memoria indirecta mediante la lectura de datos a partir de por lo menos un registro (INDF0; INDF1) de funciones especiales que no está implementado físicamente, en el que el acceso de lectura a memoria indirecta usa uno de dichos registros (150; 960) de dirección de memoria indirecta.

14. El procedimiento de acuerdo con la reivindicación 12 o 13, en el que se establece una correlación en memoria de dicho registro (INDF0; INDF1) de funciones especiales con todos los bancos de memoria.

10 15. El procedimiento de acuerdo con una de las reivindicaciones anteriores 12–14, que además comprende la lectura de los n bits más bajos de una ubicación de memoria de programa cuando se realiza un acceso a memoria indirecto a dicha memoria de programa usando uno de dichos registros (150; 960) de dirección de memoria indirecta.

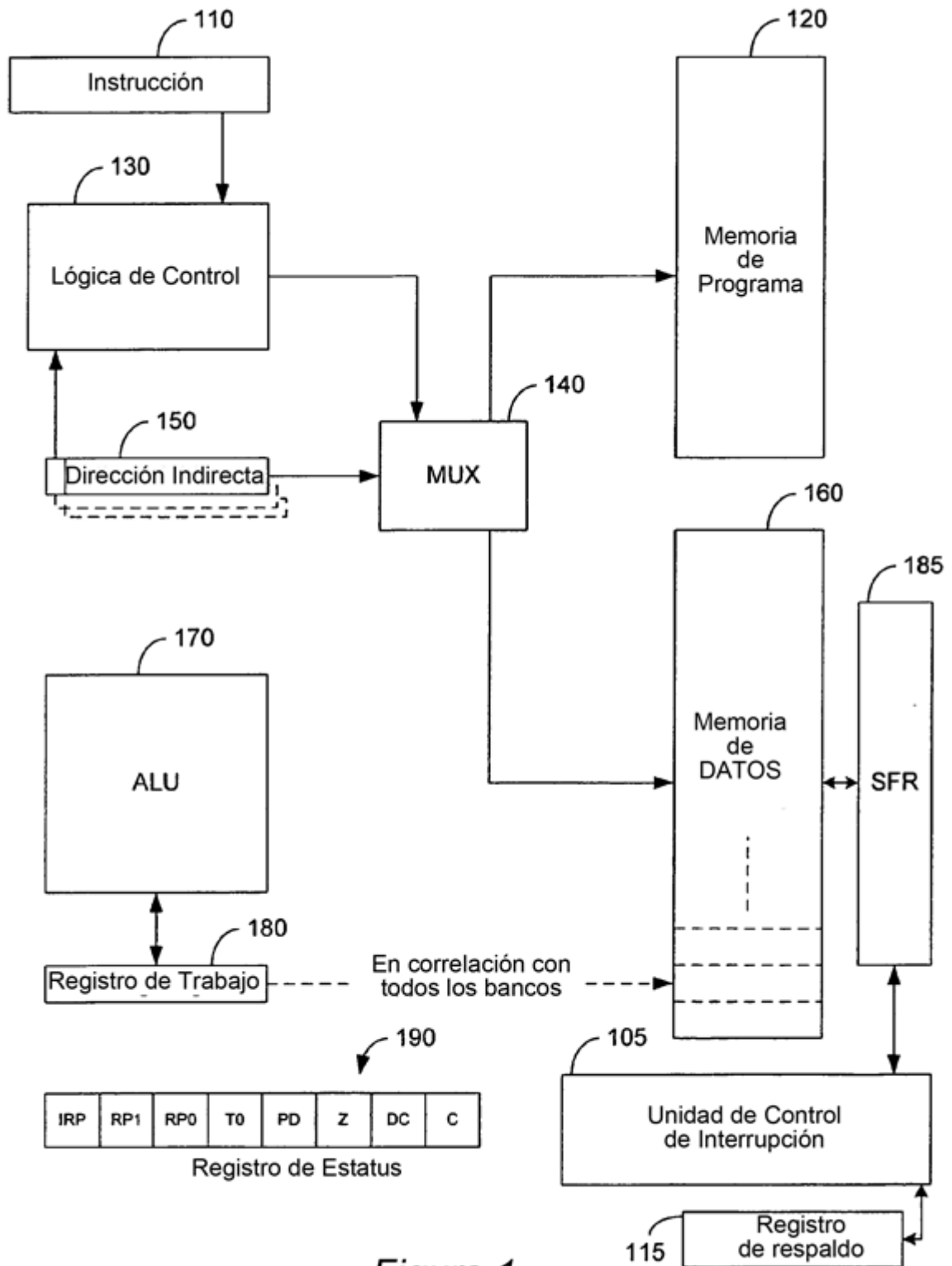


Figura 1

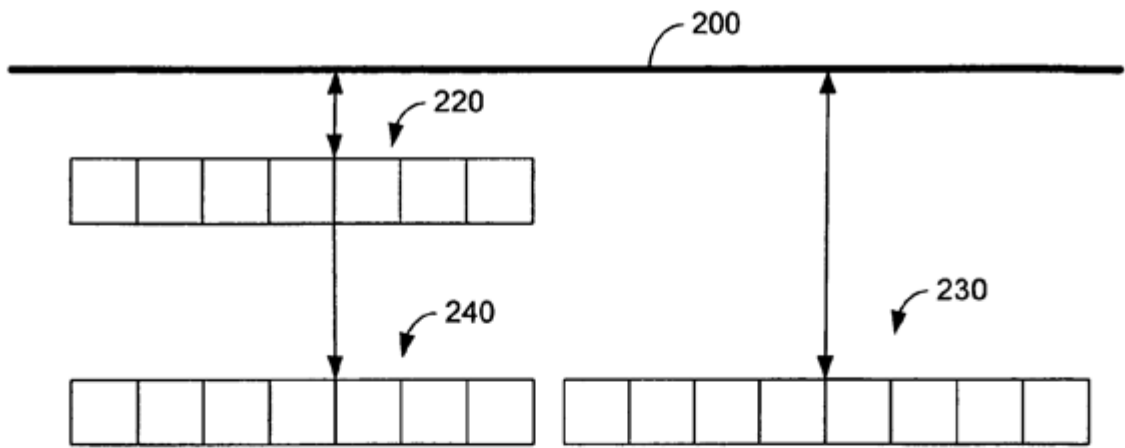


Figura 2

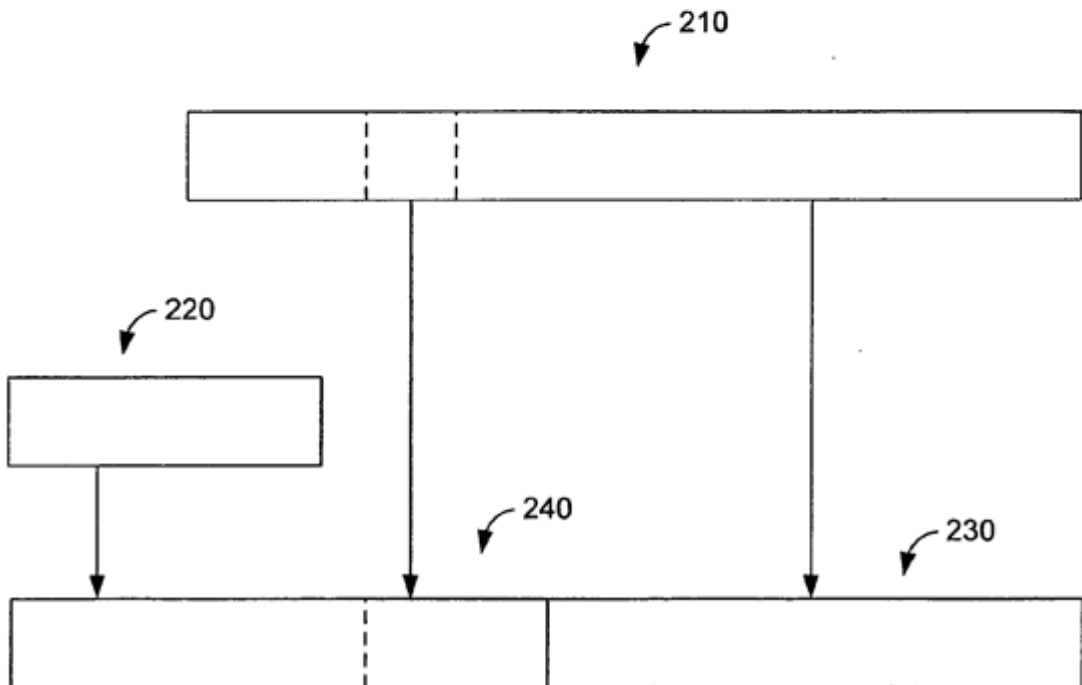


Figura 3

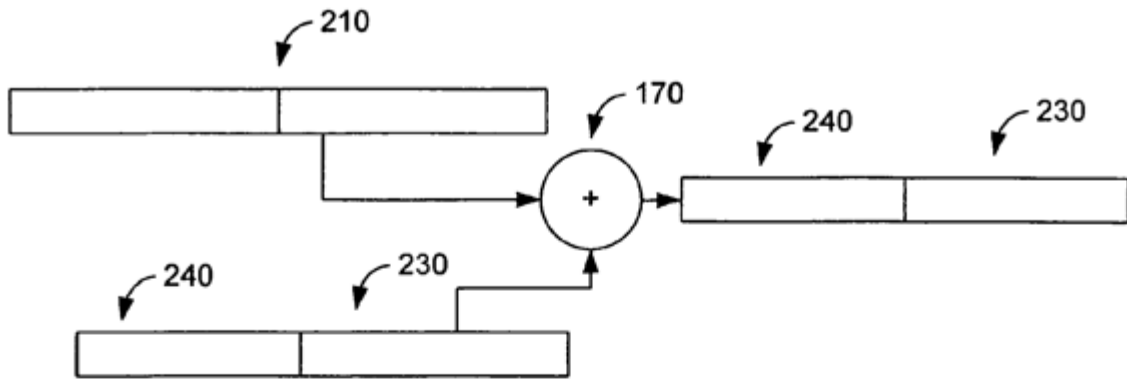


Figura 4

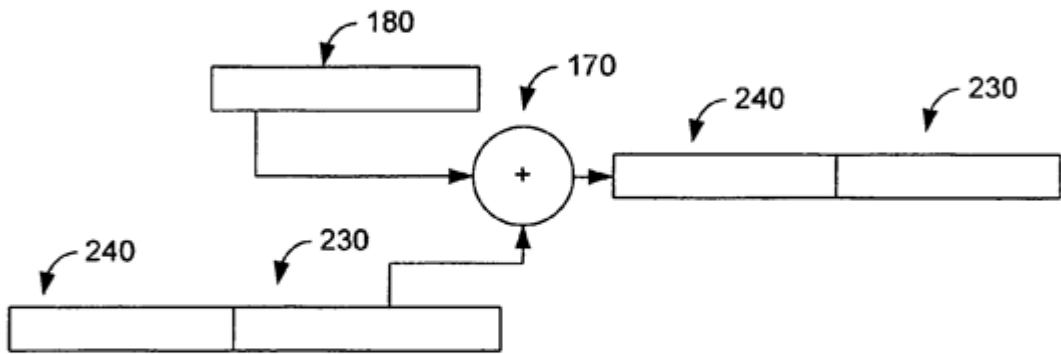


Figura 5

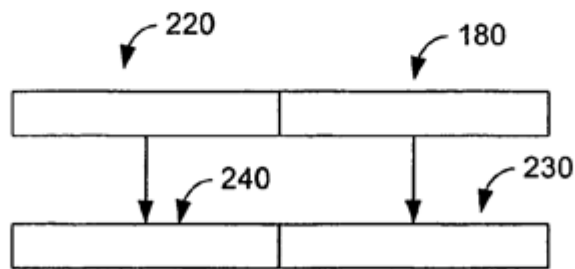


Figura 6

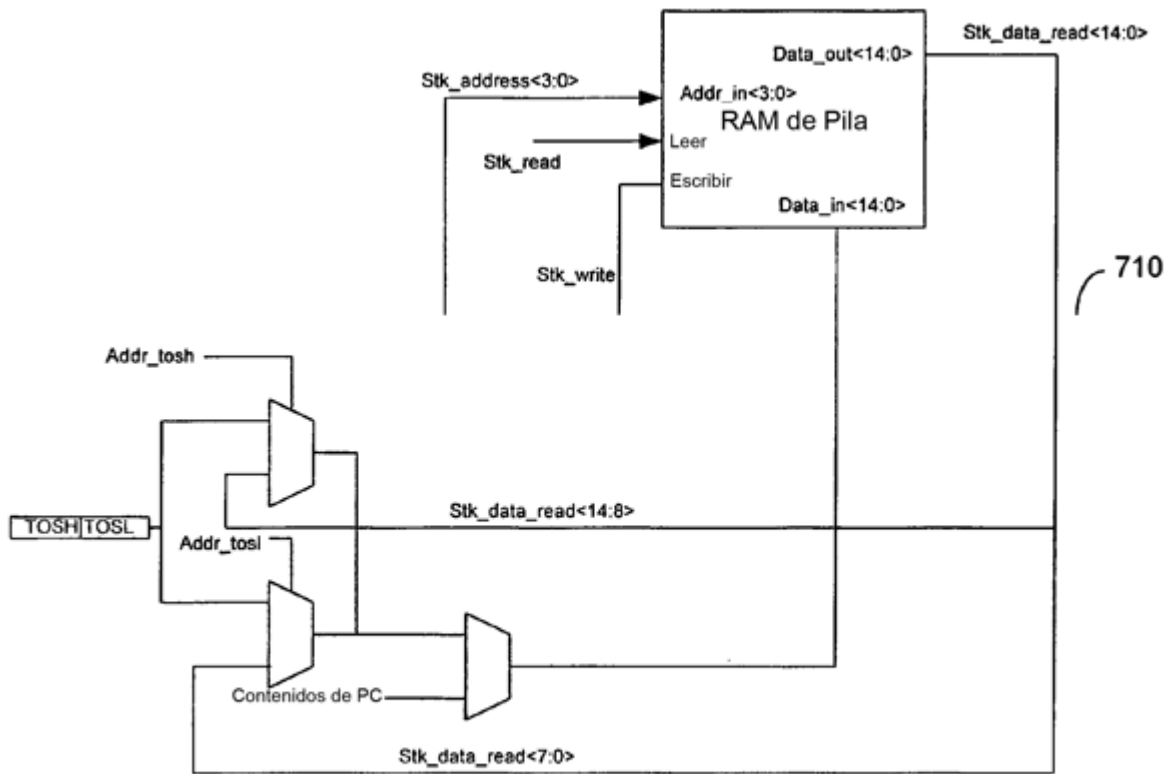


Figura 7

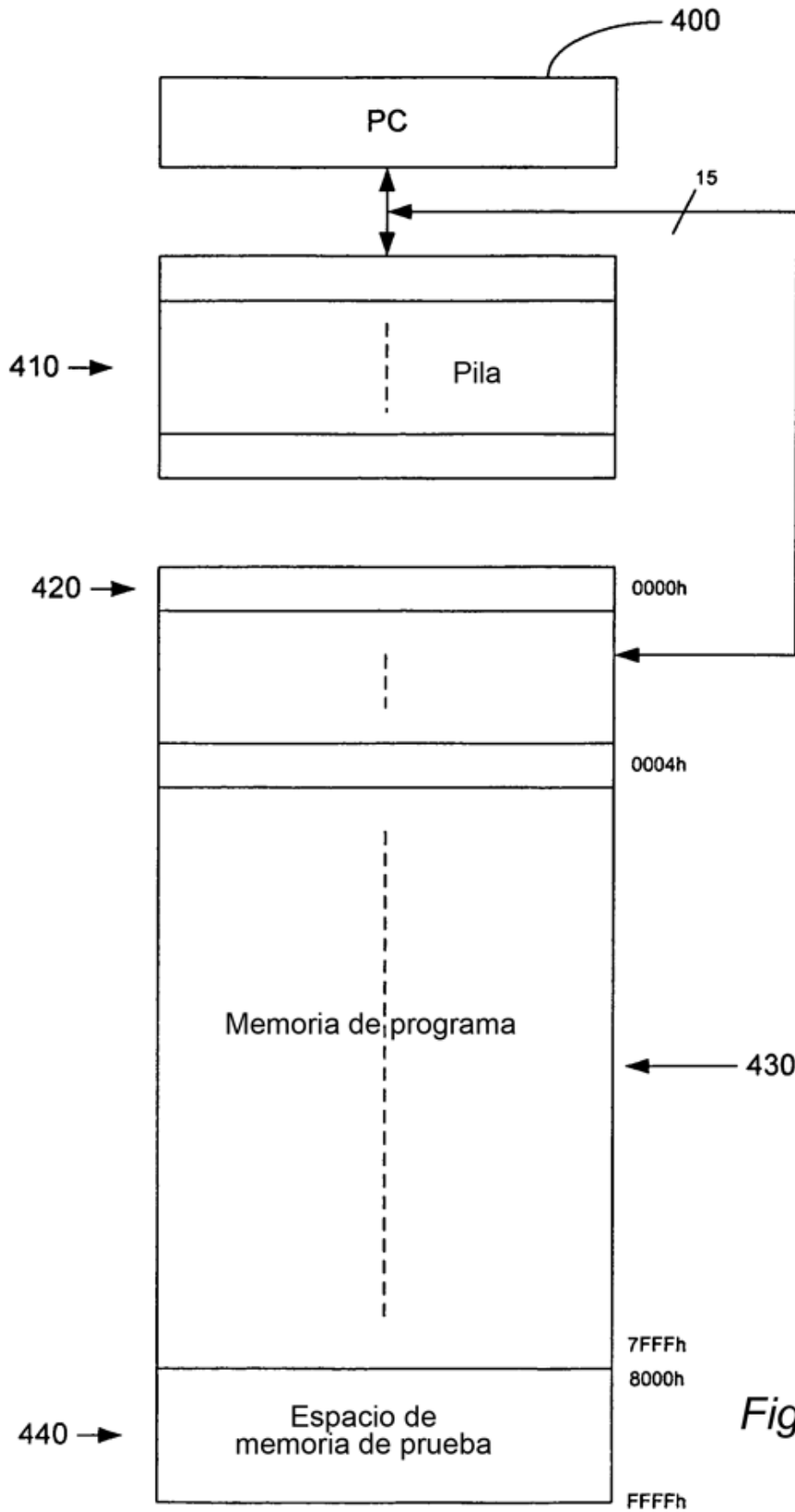
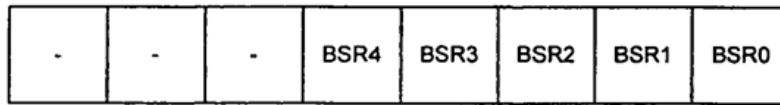
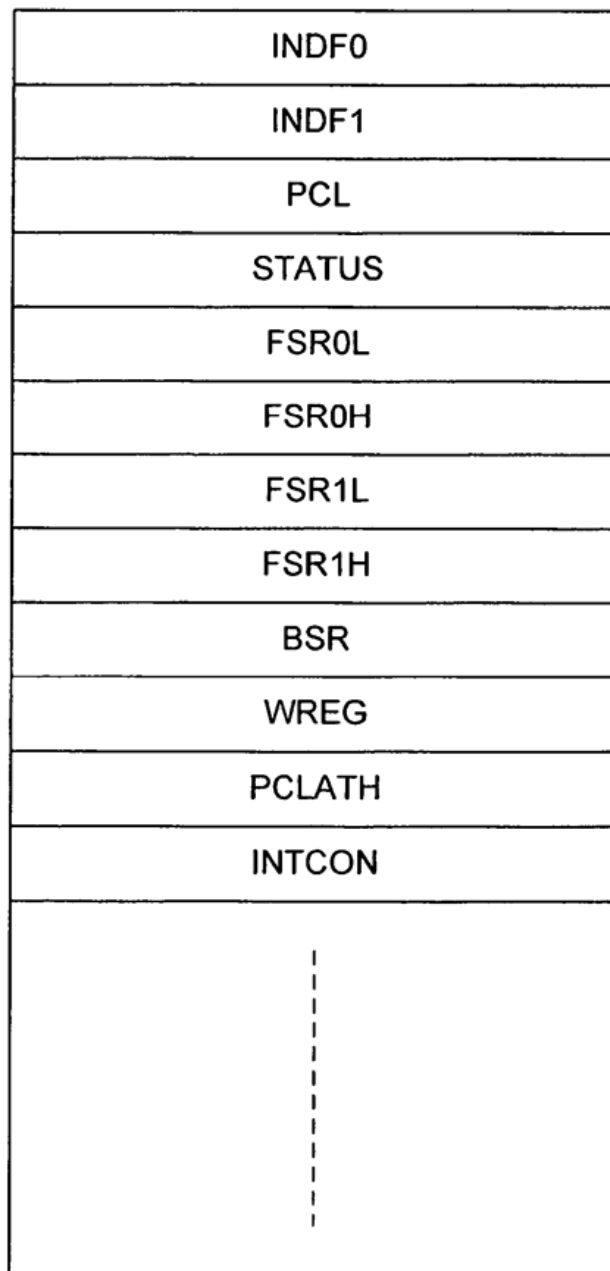


Figura 8



Registro de Selección de Banco

*Figura 9*



Banco n

*Figura 10*

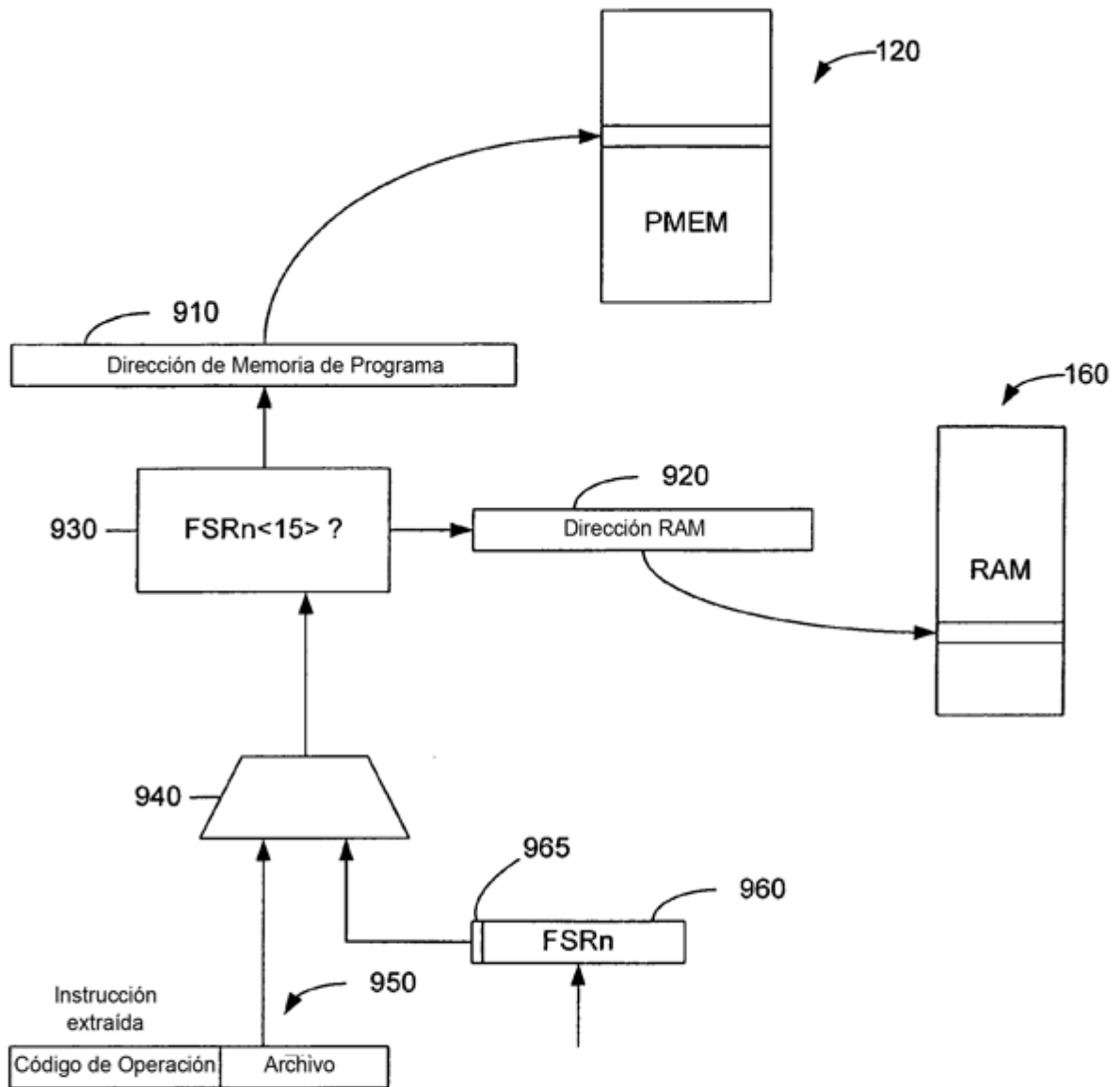


Figura 11



Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en: POR, BOR	Valor en todos los otros restablecimientos	
Banco n												
000h	INDF0	Direccionar esta ubicación usa contenidos de FSR0H/ FSR0L para direccionar memoria de datos (no memoria física)								xxxx xxxx	xxxx xxxx	
001h	INDF1	Direccionar esta ubicación usa contenidos de FSR1H/ FSR1L para direccionar memoria de datos (no memoria física)								xxxx xxxx	xxxx xxxx	
002h	PCL	Byte menos significativo del Contador de Programa (PC)								0000 0000	0000 0000	
003h	STATUS	IRP	RP1	RP0	T0	PD	Z	DC	C	0001 1xxx	0001 1xxx	
004h	FSR0L	Dirección de Memoria de Datos Indirecta 0, LSB								xxxx xxxx	xxxx xxxx	
005h	FSR0H	Dirección de Memoria de Datos Indirecta 0, MSB								xxxx xxxx	xxxx xxxx	
006h	FSR1L	Dirección de Memoria de Datos Indirecta 1, LSB								xxxx xxxx	xxxx xxxx	
007h	FSR1H	Dirección de Memoria de Datos Indirecta 1, MSB								xxxx xxxx	xxxx xxxx	
008h	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
009h	WREG	WREG7	WREG6	WREG5	WREG4	WREG3	WREG2	WREG1	WREG0	xxxx xxxx	xxxx xxxx	
00Ah	PCLATH	—	Memoria Intermedia de Escritura para los 7 bits superiores del Contador de Programa								-000 0000	-000 0000
00Bh	INTCON	GIE	PEIE	TOIE	INTE	RABIE	TOIF	INTF	RABIF	0000 0000	0000 0000	
Leyenda: x = desconocido, u = sin cambios, q = el valor depende del estado, - = sin implementar, léase como '0', r= reservado Las ubicaciones sombreadas están sin implementar, leer como '0'												

Figura 12

Operandos mnemónicos	Descripción	Ciclos	Opcode de 14 bits				Estatus Afectados	Observa- ciones	
			MSb		LSb				
<b>OPERACIONES DE REGISTRO EN ARCHIVO ORIENTADAS A BYTES</b>									
ADDWF	f, d	Añadir WREG y f	1	00	0111	dfff	ffff	C, DC, Z	1, 2
ADDWFC	f, d	Añadir con Acarreo WREG y f	1	11	1101	dfff	ffff	C, DC, Z	1, 2
ANDWF	f, d	Y de WREG con f	1	00	0101	dfff	ffff	Z	1, 2
ASRF	f, d	Desplazamiento a la Derecha Arimético	1	11	0111	dfff	ffff	C, Z	1, 2
LSLF	f, d	Desplazamiento a la Izquierda Lógico	1	11	0101	dfff	ffff	C, Z	1, 2
LSRF	f, d	Desplazamiento a la Derecha Lógico	1	11	0110	dfff	ffff	C, Z	1, 2
CLRF	f	Borrar f	1	00	0001	1fff	ffff	Z	2
CLRWF	-	Borrar WREG	1	00	0001	0000	00xx	Z	
COMF	f, d	Complementar f	1	00	1001	dfff	ffff	Z	1, 2
DECF	f, d	Decrementar f	1	00	0011	dfff	ffff	Z	1, 2
INCF	f, d	Incrementar f	1	00	1010	dfff	ffff	Z	1, 2
IORWF	f, d	OR inclusivo de WREG con f	1	00	0100	dfff	ffff	Z	1, 2
MOVF	f, d	Mover f	1	00	1000	dfff	ffff	Z	1, 2
MOVWF	f	Mover WREG a f	1	00	0000	1fff	ffff		
RLF	f, d	Girar a la Izquierda f a través del Acarreo	1	00	1101	dfff	ffff	C	1, 2
RRF	f, d	Girar a la Derecha f a través del Acarreo	1	00	1100	dfff	ffff	C	1, 2
SUBWF	f, d	Restar WREG de f	1	00	0010	dfff	ffff	C, DC, Z	1, 2
SUBWFB	f, d	Restar con Acarreo Negativo WREG de f	1	11	1011	dfff	ffff	C, DC, Z	1, 2
SWAPF	f, d	Intercambiar Cuartetos en f	1	00	1110	dfff	ffff		1, 2
XORWF	f, d	OR Exclusivo de WREG con f	1	00	0110	dfff	ffff	Z	1, 2
<b>OPERACIONES DE SALTO ORIENTADAS A BYTES</b>									
DECFSZ	f, d	Decrementar f, Saltar si es 0	1(2)	00	1011	dfff	ffff		1, 2, 3
INCFSZ	f, d	Incrementar f, Saltar si es 0	1(2)	00	1111	dfff	ffff		1, 2, 3
<b>OPERACIONES DE REGISTRO EN ARCHIVO ORIENTADAS A BITS</b>									
BCF	f, b	Borrar Bit de f	1	01	10bb	bfff	ffff		1, 2
BSF	f, b	Establecer Bit de f	1	01	11bb	bfff	ffff		1, 2
<b>OPERACIONES DE SALTO ORIENTADAS A BITS</b>									
BTFSZ	f, b	Probar Bit de f, Saltar si Borrado	1 (2)	01	10bb	bfff	ffff		3
BTFSZ	f, b	Probar Bit de f, Saltar si Establecido	1 (2)	01	11bb	bfff	ffff		3
<b>OPERACIONES CON LITERALES</b>									
ADDLW	k	Añadir literal y WREG	1	11	1110	kkkk	kkkk	C, DC, Z	
ANDLW	k	Y de literal con WREG	1	11	1001	kkkk	kkkk	Z	
IORLW	k	OR inclusivo de literal con WREG	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	Mover literal a BSR	1	00	0000	001k	kkkk		
MOVLP	k	Mover literal a PCLATH	1	11	0001	1kkk	kkkk		
MOVLW	k	Mover literal a WREG	1	11	0000	kkkk	kkkk		
SUBLW	k	Sustraer WREG de literal	1	11	1100	kkkk	kkkk	C, DC, Z	
XORLW	k	OR exclusivo de literal con WREG	1	11	1010	kkkk	kkkk	Z	

Observaciones 1: Cuando un registro de E/S se modifica como una función de sí mismo (por ejemplo, MOVF GPIO, 1) el valor usado será el valor presente en las propias patillas. Por ejemplo, si el bloqueo de datos '1' para una patilla configurada como entrada y se pasa a valor bajo por un dispositivo externo, los datos volverán a escribirse con un '0'.  
 2: Si esta instrucción se ejecuta en el registro TMR0 (y donde sea aplicable, d = 1) el contador predivisor se borrará si se asigna al módulo Timer0.  
 3: Si el Contador de Programa (PC) se modifica, o si una prueba condicional es verdadera, la instrucción requiere dos ciclos. El segundo ciclo se ejecuta como una NOP.

Figura 13A

Operandos mnemónicos	Descripción	Ciclos	Opcode de 14 bits				Estatus Afectados	Observa- ciones
			MSb		LSb			
<b>OPERACIONES DE CONTROL</b>								
<b>BRA</b>	<b>k</b>	Ramificación Relativa	2	11	001k	kkkk	kkkk	
<b>BRW</b>	-	Ramificación Relativa con WREG	2	00	0000	0110	0000	
<b>CALL</b>	<b>k</b>	Llamar Subrutina	2	10	0kkk	kkkk	kkkk	
<b>CALLW</b>	-	Llamar Subrutina con WREG	2	00	0000	0000	1010	
<b>GOTO</b>	<b>k</b>	Ir a dirección	2	10	1kkk	kkkk	kkkk	
<b>RETFIE</b>	<b>k</b>	Volver de Interrupción	2	00	0000	0000	1001	
<b>RETLW</b>	<b>k</b>	Volver con literal en WREG	2	11	0100	kkkk	kkkk	
<b>RETURN</b>	-	Volver de Subrutina	2	00	0000	0000	1000	
<b>OPERACIONES INHERENTES</b>								
<b>CLRWDT</b>	-	Borrar Temporizador Guardián	1	00	0000	0110	0100	$\overline{TO}$ , $\overline{PD}$
<b>NOP</b>	-	Sin Operación	1	00	0000	0000	0000	
<b>RESET</b>	-	Restablecimiento de Dispositivo de Software	1	00	0000	0110	0001	
<b>SLEEP</b>	-	Pasar a modo de espera	1	00	0000	0110	0011	$\overline{TO}$ , $\overline{PD}$
<b>OPTIMIZADO PARA COMPILADOR-C</b>								
<b>ADDFSR</b>	<b>n, k</b>	Añadir Literal a FSRn	1	11	0001	0nkk	kkkk	
<b>MOVIW</b>		Mover INDFn a WREG, con pre-/ post-inc/ dec	1	00	0000	0001	0nmm	Z
		Mover INDFn a WREG, indirecto indizado	1	11	1111	0nkk	kkkk	Z
<b>MOVWI</b>		Mover WREG a INDFn, con pre-/ post-inc/ dec	1	00	0000	0001	1nmm	
		Mover WREG a INDFn, indirecto indizado	1	11	1111	1nkk	kkkk	

Observaciones 1: Cuando un registro de E/S se modifica como una función de sí mismo (por ejemplo, MOVF GPIO, 1) el valor usado será el valor presente en las propias patillas. Por ejemplo, si el bloqueo de datos '1' para una patilla configurada como entrada y se pasa a valor bajo por un dispositivo externo, los datos volverán a escribirse con un '0'.

2: Si esta instrucción se ejecuta en el registro TMR0 (y donde sea aplicable, d = 1) el contador predivisor se borrará si se asigna al módulo Timer0.

3: Si el Contador de Programa (PC) se modifica, o si una prueba condicional es verdadera, la instrucción requiere dos ciclos. El segundo ciclo se ejecuta como una NOP.

Figura 13B