

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 606 128**

51 Int. Cl.:

G06F 13/16 (2006.01)

G06F 13/362 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **20.07.2010 PCT/US2010/042525**

87 Fecha y número de publicación internacional: **27.01.2011 WO11011364**

96 Fecha de presentación y número de la solicitud europea: **20.07.2010 E 10740057 (4)**

97 Fecha y número de publicación de la concesión europea: **14.09.2016 EP 2457169**

54 Título: **Arbitrador de espacio de datos**

30 Prioridad:

21.07.2009 US 227147 P

18.06.2010 US 818325

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

22.03.2017

73 Titular/es:

MICROCHIP TECHNOLOGY INCORPORATED

(100.0%)

2355 West Chandler Boulevard

Chandler, Arizona 85224-6199, US

72 Inventor/es:

CATHERWOOD, MICHAEL, I. y

DESAI, ASHISH

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 606 128 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

Arbitrador de espacio de datos

La presente divulgación se refiere a procesadores digitales, y, más particularmente, a sistemas de control de prioridad de arbitradores de datos en un procesador digital.

- 5 En sistemas embebidos que usan procesadores digitales, por ejemplo, microcontroladores, que tienen una pluralidad de dispositivos periféricos y una unidad de procesamiento central (CPU), se comparten ciertas unidades funcionales entre la CPU y al menos algunos de los periféricos tales como un controlador de acceso directo a memoria (DMA) o un controlador de depurador en circuito (ICD). Por lo tanto, se usa un arbitrador de espacio de datos para determinar derechos de acceso a la memoria o registros de funciones especiales que se comparten. En un arbitrador de espacio de datos para un sistema de múltiples maestros, la CPU es típicamente el bus maestro de la prioridad más alta. Un arbitrador de espacio de datos típico consiste en un codificador de prioridad y un conjunto de multiplexores de bus de datos. Un arbitrador de espacio de datos programable convencional implementa la capacidad de programación en el codificador de prioridad pero usa prioridades predeterminadas en las que la CPU en general tiene la prioridad más alta. Por lo tanto, cada vez que la CPU está accediendo a la memoria todos los otros periféricos normalmente se bloquean. Sin embargo, en ocasiones en sistemas embebidos, justo de manera temporal, puede ser más crítico responder a un evento externo, mediante, por ejemplo, un periférico para entonces asegurar alto rendimiento de CPU. El documento US 6.272.580 desvela un aparato y procedimiento para elevar dinámicamente un bus maestro de nivel inferior a un bus maestro de nivel superior en un sistema de arbitración de múltiples niveles.
- 10
- 15
- 20 Sin embargo, existe una necesidad de un arbitrador de espacio de datos más flexible mejorado. Este y otros objetos pueden conseguirse mediante el procesador de datos y procedimiento de acuerdo con las reivindicaciones independientes. Mejoras adicionales están caracterizadas en las reivindicaciones dependientes.

De acuerdo con una realización, un procesador digital puede comprender un bus maestro por defecto que tiene una prioridad más alta en un modo por defecto; una pluralidad de buses maestros secundarios que tienen prioridades asociadas, en el que la pluralidad de buses maestros secundarios tienen una relación de prioridad predeterminada entre sí; y un arbitrador de espacio de datos, en el que el arbitrador de espacio de datos es programable en un modo no por defecto para elevar una prioridad de cualquiera de los buses maestros secundarios para tener una prioridad más alta que la prioridad del bus maestro por defecto mientras mantiene la relación de prioridad predeterminada a únicamente aquellos buses maestros secundarios para los que el nivel de prioridad también se ha elevado por encima de la prioridad del bus maestro por defecto.

25

30

De acuerdo con una realización adicional, el arbitrador de espacio de datos puede conceder acceso a memoria de datos y la memoria de datos puede comprender al menos uno de memoria de acceso aleatorio, memoria de puerto dual, registros de funciones especiales. De acuerdo con una realización adicional, el procesador digital puede comprender adicionalmente un registro de prioridad de bus maestro que tiene bits de contraorden de prioridad, al menos uno de los buses maestros secundarios está asignado a uno de los bits de contraorden de prioridad, en el que el registro de prioridad de bus maestro especifica programablemente cuáles de la pluralidad de buses maestros secundarios tienen prioridades más altas o más bajas que la prioridad del bus maestro por defecto. De acuerdo con una realización adicional, establecer un bit de contraorden de prioridad puede provocar que se eleve un nivel de prioridad de un bus maestro secundario asociado por encima del nivel de prioridad del bus maestro por defecto. De acuerdo con una realización adicional, el procesador digital puede comprender adicionalmente un decodificador de dirección acoplado entre el arbitrador de espacio de datos y la memoria de datos. De acuerdo con una realización adicional, el procesador digital puede comprender adicionalmente un decodificador de dirección aproximada acoplado entre el arbitrador de espacio de datos y el bus maestro por defecto o buses maestros secundarios. De acuerdo con una realización adicional, el procesador digital puede ser un procesador de señales digitales y la memoria de acceso aleatorio puede dividirse en una memoria X y una memoria Y. De acuerdo con una realización adicional, el bus maestro por defecto puede ser una unidad de procesamiento central (CPU).

35

40

45

De acuerdo con otra realización, un procedimiento para arbitrar acceso de espacio de datos en un procesador digital, puede comprender: asignar prioridades a un bus maestro por defecto y a una pluralidad de buses maestros secundarios en el que el bus maestro por defecto tiene la prioridad más alta en un modo por defecto y en el que la pluralidad de buses maestros secundarios tienen una relación de prioridad predeterminada entre sí; y programar un arbitrador de espacio de datos en un modo no por defecto para elevar una prioridad de cualquiera de los buses maestros secundarios para tener una prioridad más alta que la prioridad del bus maestro por defecto mientras mantiene la relación de prioridad predeterminada a únicamente aquellos buses maestros secundarios para los que el nivel de prioridad también se ha elevado por encima de la prioridad del bus maestro por defecto.

50

De acuerdo con una realización adicional del procedimiento, el procedimiento puede comprender adicionalmente: solicitar acceso a un espacio de datos mediante un bus maestro secundario mientras el bus maestro por defecto tiene acceso al espacio de datos y si la prioridad del bus maestro secundario es más alta que la prioridad del bus maestro por defecto, conceder acceso al espacio de datos bloqueando de otro modo el bus maestro secundario. De acuerdo con una realización adicional del procedimiento, el espacio de datos puede ser memoria de datos que

55

- comprende al menos uno de memoria de acceso aleatorio, memoria de puerto dual, registros de funciones especiales. De acuerdo con una realización adicional del procedimiento, la etapa de programar un arbitrador de espacio de datos para elevar una prioridad de los buses maestros secundarios puede comprender la etapa de establecer un bit en un registro de prioridad de bus maestro. De acuerdo con una realización adicional del procedimiento, al menos uno del bus maestro secundario puede tener un bit asociado en el registro de prioridad de bus maestro. De acuerdo con una realización adicional del procedimiento, el arbitrador de espacio de datos puede comprender una unidad arbitradora separada para accesos de lectura y escritura. De acuerdo con una realización adicional del procedimiento, el arbitrador de espacio de datos puede comprender una unidad arbitradora separada para accesos de memoria y registro de funciones especiales.
- 5
- 10 Un entendimiento más completo de la presente divulgación y las ventajas de la misma puede obtenerse haciendo referencia a la siguiente descripción tomada junto con los dibujos adjuntos en los que:
- La Figura 1 muestra un diagrama de bloques de arbitradores de espacio de datos, multiplexores de bus y decodificadores de dirección en un sistema microcontrolador;
- 15 La Figura 2 muestra una tabla de prioridad de arbitrador de bus para varios buses maestros de acuerdo con una realización;
- La Figura 3 muestra un diagrama de bloques de lectura de arbitrador de acuerdo con una realización;
- La Figura 4 muestra un diagrama de bloques de escritura de arbitrador de acuerdo con una realización;
- La Figura 5 muestra un diagrama de bloques de interconexión de arbitrador de acuerdo con una realización;
- 20 La Figura 6 y 7 muestran una tabla de verdad de ejemplo de arbitrador de RAM para cuatro buses maestros de acuerdo con una realización;
- La Figura 8 muestra ejemplos de temporización de arbitradores de RAM teniendo la CPU la prioridad más alta;
- La Figura 9 muestra ejemplos de temporización de arbitradores de RAM no teniendo la CPU la prioridad más alta;
- 25 La Figura 10 muestra un diagrama de bloques de interconexión de arbitrador SFR de acuerdo con una realización; y
- La Figura 11 muestra ejemplos de temporización de arbitradores de SFR teniendo la CPU la prioridad más alta;
- Aunque las realizaciones de esta divulgación se han representado, descrito y definido por referencia a realizaciones de ejemplo de la divulgación, tales referencias no implican una limitación en la divulgación, y no ha de inferirse tal limitación. La materia objeto desvelada es considerable de modificación, alteración y equivalentes en forma y función, como se les ocurrirá a los expertos en la materia en la técnica pertinente y que tienen el beneficio de esta divulgación. Las realizaciones representadas y descritas de esta divulgación son únicamente ejemplos.
- 30
- De acuerdo con diversas realizaciones, un arbitrador de espacio de datos está diseñado para permitir que se eleve la prioridad de cualquier bus maestro con relación a la prioridad del bus maestro que tiene la prioridad más alta en modo por defecto (en lo sucesivo "maestro por defecto"), aumentando de esta manera la flexibilidad de respuesta en tiempo real en un sistema de procesador digital. Esto puede conseguirse de una manera fácil manteniendo la relación de prioridad predeterminada entre el bus maestro elevado y únicamente aquellos buses maestros para los que el nivel de prioridad también se ha elevado por encima de la prioridad del maestro por defecto. Por lo tanto, no es necesario reasignación complicada de niveles de prioridad de cada bus maestro. De acuerdo con las enseñanzas de esta divulgación, cada bus maestro (excepto el maestro por defecto) puede optar a contraordenar esta prioridad.
- 35
- 40 Si el maestro por defecto es la CPU, cualquier prioridad de bus maestro (no de CPU) puede definirse, por ejemplo, a través de bits de contraorden de prioridad programables definidos por usuario, para que sea más alta o más baja que la de la CPU. Sin embargo, la prioridad relativa de todos los buses maestros en el grupo de prioridades más altas que las del maestro por defecto está diseñada para que no cambie. De manera similar, la prioridad relativa de todos buses maestros en el grupo de prioridades más bajas que las del maestro por defecto no cambia. Las solicitudes de bus maestro entrantes simplemente se reordenan, por ejemplo, usando multiplexores, basándose en el estado de los bits de contraorden de prioridad. Se envían a continuación a un codificador de prioridad fija sencillo y se priorizan como es habitual. Las señales de concesión de bus resultantes se vuelven a mapear a continuación de vuelta a su orden original, por ejemplo, usando de nuevo el estado de los bits de contraorden de prioridad, y al bus maestro ganador se le concede acceso de bus. Todos los otros buses maestros se bloquean.
- 45
- 50 La Figura 1 muestra un diagrama de bloques principal de un procesador 100 de señales digitales con tres buses maestros, una CPU 110, una unidad 105 de depurador en circuito (ICD) y un controlador 115 de acceso directo a memoria (DMA). Sin embargo, pueden aplicarse otras configuraciones con un procesador de fin general y diferentes buses maestros como se apreciará por un experto en la materia. Un decodificador 120 de dirección aproximada está acoplado con estos buses maestros para decidir si una solicitud alcanza a la memoria de acceso aleatorio (RAM) o a

los registros de funciones especiales (SFR). Este decodificador de dirección aproximada genera una pluralidad de señales de salida que se acoplan con respectivos arbitradores 130 y 140 de lectura especializados, y arbitradores 135 y 145 de escritura especializados. Los arbitradores 130, 135 de RAM están acoplados con un decodificador de dirección de nivel de panel para acceder a dos memorias 170 y 175 de acceso aleatorio estáticas (SRAM) separadas. Además, se proporciona una SRAM 180 de puerto dual (DP) que está acoplada, por una parte, con la salida del decodificador 160 de dirección y, por otra, con un decodificador 190 de dirección aproximada especial que puede ser parte de o estar asociado con el decodificador 120 de dirección aproximada. Las salidas de todos los dispositivos 170, 175 y 180 de memoria están acopladas a través de diversos multiplexores 125, 150, 155 y 185 con los diferentes buses maestros 110, 105, 115. Una interfaz 165 de dispositivo periférico puede acoplarse directamente a través del multiplexor 185 con la unidad 105 de ICD. Sin embargo, como se ha mencionado anteriormente puede usarse otra configuración de memoria de acuerdo con diversas realizaciones, por ejemplo, puede usarse una única memoria con un procesador de fin general.

También, en esta realización específica de un procesador de señales digitales o microcontrolador, se proporcionan arbitradores separados para RAM y SFR. Además, en cada arbitrador se proporcionan arbitradores separados o accesos (130; 140) de lectura y accesos (135; 145) de escritura. Sin embargo, pueden usarse otros diseños y el número de arbitradores real puede variar. Por ejemplo, puede usarse un único arbitrador para proporcionar la misma funcionalidad.

De acuerdo con diversas realizaciones, los arbitradores son configurables mediante los diferentes buses maestros. Esto puede hacerse "al vuelo", en otras palabras, un programa de usuario puede modificar la configuración de acuerdo con respectivas necesidades del sistema en el que está embebido el dispositivo. La Figura 2 muestra una tabla con una posible implementación en la que un procesador digital está asociado con, por ejemplo, cinco buses maestros tales como una CPU que es el bus maestro de la prioridad más alta en un modo por defecto, y una pluralidad de buses maestros secundarios tales como una unidad de Ethernet, una unidad de USB, un controlador de DMA, y una unidad de ICD. Este esquema sencillo se amplía fácilmente para añadir más buses maestros o usar menos, y no se basa en un codificador de prioridad programable más grande, potencialmente más lento. De acuerdo con una realización, el sistema puede no ser una solución completamente programable debido a que la prioridad relativa de bus maestro en cada grupo puede permanecer fija. Esto puede conseguirse, por ejemplo, estableciendo bits respectivos en el registro de funciones especiales MSTRPR 195 que está acoplado con los arbitradores 130, 135, 140 y 145 como se muestra en la Figura 1.

El registro 195 MSTRPR puede configurarse como sigue:

Mitad superior:

U	U	U	U	U	U	U	U
0	0	0	0	0	0	0	0
bit 15				bit 8			

Mitad inferior:

U	U	L/E	U	L/E	U	L/E	U
0	0	DMAC (M3)	0	USB (M2)	0	ETH (M1)	0
bit 7			bit 0				

en el que los bits 15-6 no se usan y los bits 6, 4, 2 y 0 están reservados. El bit 5 de DMAC (M3) se usa para modificar la prioridad de bus maestro de DMAC con relación a la prioridad de la CPU. Cuando se establece a 1: la prioridad de bus maestro de DMAC se eleva por encima de la de la CPU mientras mantiene la relación de prioridad predeterminada a únicamente aquellos buses maestros secundarios para los que el nivel de prioridad también se ha elevado por encima de la prioridad de la CPU. Cuando se establece a 0: no hay cambio a la prioridad de bus maestro de DMAC. La prioridad del bus maestro de DMAC permanece por debajo de la de la CPU y está en la relación predeterminada a todos los otros buses maestros secundarios que no se han elevado por encima de la de la CPU. El Bit 3 de USB (M2) se usa para modificar la prioridad de bus maestro de USB con relación a la prioridad de la CPU. Cuando se establece a 1: la prioridad de bus maestro de USB se eleva por encima de la de la CPU mientras mantiene la relación de prioridad predeterminada a únicamente aquellos buses maestros secundarios para los que el nivel de prioridad también se ha elevado por encima de la prioridad de la CPU. Cuando se establece a 0: no hay cambio a la prioridad de bus maestro de USB. La prioridad del bus maestro USB permanece por debajo de la de la CPU y está en la relación predeterminada a todos los otros buses maestros secundarios que no se han elevado por encima de la CPU. El bit 1 de ETH (M2) se usa para modificar la prioridad de bus maestro de Ethernet con relación a la prioridad de la CPU. Cuando se establece a 1: la prioridad de bus maestro de Ethernet se eleva por encima de la de la CPU mientras mantiene la relación de prioridad predeterminada a únicamente aquellos buses maestros secundarios para los que el nivel de prioridad también se ha elevado por encima de la prioridad de la CPU. Cuando

se establece a 0: no hay cambio a la prioridad de bus maestro de Ethernet. La prioridad del bus maestro de Ethernet permanece por debajo de la de la CPU y está en la relación predeterminada a todos los otros buses maestros secundarios que no se han elevado por encima de la CPU.

5 De acuerdo con una realización, las prioridades por defecto en reseteo se establecen de manera que la CPU tiene la prioridad más alta y el ICD la prioridad más baja. En la realización anteriormente mostrada, la prioridad del bus maestro de ICD no se asigna a un bit en el registro 195 MSTRPR. Por lo tanto, su prioridad permanece en el nivel más bajo. Sin embargo, en otras realizaciones, la prioridad de este dispositivo puede elevarse también y puede asignarse a un bit en el registro 195. De acuerdo con una realización, todos los buses maestros de prioridad elevada mantienen la misma relación de prioridad con relación entre sí. De acuerdo con una realización, todos los maestros
10 cuya prioridad permanece por debajo de la de la CPU mantienen la misma relación de prioridad con relación entre sí. Por lo tanto, los buses maestros secundarios pueden dividirse en dos grupos. El primer grupo contiene los buses maestros secundarios con una prioridad elevada y el segundo grupo contiene los buses maestros secundarios cuya prioridad no se ha elevado.

15 Como se muestra en la Figura 2, únicamente se usan ciertos bits y dan como resultado ocho configuraciones diferentes teniendo cada una un diferente orden de prioridad de los respectivos buses maestros. Este esquema relativamente sencillo permite una implementación fácil en la que se mantiene la decodificación a un mínimo. Sin embargo, en otras realizaciones, pueden aplicarse otros esquemas de decodificación que permiten una mayor asignación de prioridades.

20 Los arbitadores 130, 135, 140, 145 de espacio de datos priorizan solicitudes concurrentes desde los buses maestros (iniciadores) 110, 105, 115 para recursos de bus de espacio de datos (objetivos) 170, 175 y 180 usando una prioridad predeterminada. La matriz de bus conecta el bus maestro ganador con el recurso solicitado. Los buses maestros perdedores no se les concede acceso al recurso solicitado y pueden bloquearse hasta que la arbitración determina que el recurso está disponible para usarse por ellos (un azar estructural). Los arbitadores que controlan acceso al espacio de datos excluyendo el espacio de SFR se denominan como los arbitadores 130 y 135
25 de RAM de escritura y lectura. Los arbitadores que controlan el acceso al espacio de SFR se denominan como los arbitadores 140 y 145 de SFR de lectura y escritura. Cada arbitador 130, 135, 140, 145 está parametrizado para soportar 'n' buses maestros donde $n = \text{NUM-BUS-MASTERS}$, que permite que la misma celda hoja se use para todas las instancias. Los buses maestros de la CPU 110, DMAC 115 e ICD 105 pueden estar presentes en muchas realizaciones, por lo tanto en general $\text{NUM-BUS-MASTERS} \geq 3$ para los arbitadores de RAM. De acuerdo con una
30 realización específica, únicamente la CPU 110 y el ICD 105 pueden acceder al espacio de dirección de SFR, por lo tanto $\text{NUM-BUS-MASTERS} = 2$ (siempre) para los arbitadores 140 y 145 de SFR.

Los recursos de espacio de datos (DS) (que no incluye el espacio de direcciones de SFR) y objetivos de arbitador pueden ser como sigue de acuerdo con una realización. Cada recurso requiere un arbitador.

1. lectura de DS RAM (SRAM + DPSRAM frontal)

35 2. escritura de DS RAM (SRAM + DPSRAM frontal)

De acuerdo con una realización específica como se muestra en la Figura 1, el controlador 115 de DMA accede a DPSRAM 180 mediante el puerto trasero, por lo que puede observarse como un caso especial. Los recursos de espacio de direcciones de SFR y objetivos de arbitador pueden ser como sigue de acuerdo con una realización. Cada recurso requiere un arbitador.

40 1. Escritura de SFR

2. Lectura de SFR

45 De acuerdo con una realización, el bus maestro de la prioridad más baja es $M[n-1]$ donde $n = \text{NUM-BUS-MASTERS}$. Tanto para el arbitador de RAM como el de SFR, esto puede ser siempre la macro del ICD. Las prioridades de bus maestro restantes son ligeramente programables por el usuario mediante el registro de control de prioridad de bus maestro, MSTRPR como se ha explicado anteriormente.

50 Una lectura de bus maestro 110, 105, 115 será una lectura de datos desde la RAM 170, 175 (para los arbitadores de RAM) o de espacio de SFR (para los arbitadores de SFR). Una escritura de bus maestro será una escritura de datos a la RAM 170, 175 (para los arbitadores de RAM) o de espacio de SFR (para los arbitadores de SFR). El control entre los buses maestros 110, 105, 115 y los arbitadores 130, 135, 140, 145 puede realizarse usando una solicitud de bus/toma de contacto de concesión. Todos los buses maestros 110, 105, 115 pueden solicitar un recurso de bus 170, 175, 180 declarando sus señales de solicitud de bus de lectura o escritura. Estarán únicamente permitidos a usar el recurso solicitado cuando el arbitador 130, 135, 140, 145 asociado haya concedido acceso declarando la correspondiente señal de concesión de bus.

55 La interfaz de arbitador de bus maestro de CPU puede ser la misma que todos los otros maestros. Sin embargo, puesto que puede completar tanto una lectura como una escritura en un ciclo de bus, en el que todos los otros buses maestros pueden, o bien leer, o bien escribir para cada transacción de bus, la temporización de la interfaz es

ligeramente diferente.

Adicionalmente, en un modo de baja potencia, tal como por ejemplo, un modo EN REPOSO, la CPU 110 puede estar operando a una velocidad de reloj significativamente inferior que la de los dispositivos periféricos. Para evitar que la CPU 110 bloquee solicitudes de recursos posteriores de los periféricos, en el que un ciclo de solicitud de CPU podría abarcar muchos ciclos periféricos, las solicitudes de la CPU 110 no están basadas en el reloj Q sino que están sincronizadas a los relojes P antes de la presentación a los arbitradores.

Puede haber 3 decodificadores de dirección implementados de acuerdo con una realización como se muestra en la Figura 1. El primer decodificador 120 de direcciones (decodificador de espacio aproximado de SFR) determina si las direcciones de lectura y escritura de la CPU 110 y el ICD 105 tienen como objetivo los arbitradores 130, 135, 140, 145 de SFR o RAM. El segundo decodificador 190 de direcciones (decodificador de espacio aproximado de DPSRAM) determina si la dirección de DMA (lectura o escritura) tiene como objetivo o no el DPSRAM 180 (trasero). El tercer decodificador 160 de direcciones (decodificador de panel WY) toma las direcciones resultantes de los arbitradores de lectura y escritura de RAM y los dirige a cualquiera del espacio de direcciones X o Y.

Un arbitrador de bus puede ser un bloque de lógica combinatoria que prioriza continuamente todas las solicitudes de bus entrantes, genera una señal de concesión ganadora, y dirige el bus de direcciones maestro ganador al recurso arbitrado. Puede consistir en un codificador de prioridad y un conjunto de multiplexores. El codificador de prioridad examina todas las solicitudes de bus entrantes y las prioriza basándose en una codificación de prioridad programable como se ha explicado anteriormente. El resultado de esta codificación se pasa a los multiplexores de direcciones y datos, y a un bus de concesión codificada de tipo "one-hot" que concede las transacciones de bus al bus maestro ganador. Los buses de direcciones y de datos correspondientes se seleccionan y el bus maestro ganador obtiene acceso al recurso solicitado. Todos los otros buses maestros solicitantes se retienen (bloquean) hasta el momento que se haya completado la transacción del bus maestro ganador.

Los arbitradores 135, 145 de escritura de datos también dirigen el bus de datos maestro ganador al recurso arbitrado. La temporización de las señales de solicitud entrantes dicta la temporización de la señal de concesión resultante, el bus de direcciones de recursos y (según sea apropiado) el bus de datos de recursos. La Figura 3 muestra un diagrama de temporización respectivo de transacciones de ejemplo para los arbitradores 130, 135 de RAM teniendo la CPU 110 la prioridad más alta. La Figura 4 muestra un diagrama de temporización similar no teniendo la CPU 110 la prioridad más alta. La salida del multiplexor de direcciones se retiene en un biestable P2 para mantener la dirección ganadora válida más allá de Q3 (la dirección de CPU puede cambiar después de Q3).

Un diagrama de temporización de transacciones de ejemplo para los arbitradores de SFR se muestra en la Figura 5 en el que esa CPU siempre tiene la prioridad más alta de acuerdo con una realización. Las Figuras 6 y 7 muestran señales de control recibidas y generadas mediante el arbitrador en accesos de lectura y escritura, respectivamente.

El controlador 115 de DMA, ICD 105 y todos los buses maestros de periféricos declaran solicitudes de bus ($xxx_bmx.mst_rd_req = 1$ o $xxx_bmx.mst_wr_req = 1$, donde 'xxx' representa la abreviatura de bus maestro) en P3 si se requiere el acceso (lectura o escritura). El análisis a continuación supone que el dispositivo no está en modo de baja potencia, donde las frecuencias de reloj Q y de reloj P pueden diferir. Por ejemplo, en un modo EN REPOSO de baja potencia, la CPU sincroniza las solicitudes de arbitrador de bus a los relojes de P para mantener el rendimiento de arbitrador para cualquier bus maestro de periféricos de prioridad más baja. De acuerdo con una realización, para solicitudes de lectura de CPU, la CPU 110 debe pre-decodificar combinatoriamente la instrucción cargada en ROMLATCH en Q3 para determinar si se requiere una solicitud de lectura. Una solicitud de lectura de CPU ($xcpu_bmx.mst_rd_req = 1$) se declarará por lo tanto algún tiempo después de Q3. La arbitración para tanto solicitudes de lectura como de escritura se completa combinatoriamente y la concesión de bus ganadora se declara a continuación ($xcpu_bmx.mst_rd_gnt = 1$). Un biestable de tipo "latch" de CPU Q1 (biestable si la temporización de concesión hará la configuración de Q1) retiene el resultado de la concesión, declarando $cpu_pfu_stall = 1$ y bloqueando la CPU si $xcpu_bmx.mst_rd_gnt = 0$.

De acuerdo con una realización, para la solicitud de escritura de CPU, la CPU 110 declarará la solicitud de escritura ($xcpu_bmx.mst_wr_req = 1$) en Q3 del ciclo de instrucción activo. Para los buses maestros de periféricos, las concesiones de arbitrador se examinan antes de P3 y si se declaran, provocarán que se niegue la señal de solicitud de bus maestro asociada en P3. Si la concesión de bus de un bus maestro de periféricos solicitante no se declara, la solicitud de bus maestro de periféricos permanecerá declarada hasta el momento que halle que se ha de declarar la concesión. El bus maestro de periféricos comprobará el estado de la señal de concesión antes de P3 del siguiente ciclo.

De acuerdo con una realización, para lecturas de CPU, se examina la concesión de arbitrador de lectura durante Q1 y si $xcpu_bmx.mst_rd_gnt = 1$, la instrucción en marcha se permite continuar la ejecución. La solicitud, $xcpu_bmx.mst_rd_req$, se negará en el siguiente Q3. Si la CPU 110 está solicitando un acceso de lectura y la concesión de bus maestro no se declara ($xcpu_bmx.mst_rd_gnt = 0$), la CPU 110 bloqueará la PFU ($cpu_pfu_stall = 0$) y la instrucción de CPU en marcha (inhibiendo todas las actualizaciones de registro). La solicitud permanecerá declarada hasta el momento que se conceda a la CPU acceso a recurso para completar el ciclo bloqueado.

- De acuerdo con una realización, para escrituras de CPU, se examina la concesión de arbitrador de escritura anterior a Q3. Si `xcpu_bmx.mst_wr_gnt = 1`, la instrucción en marcha completa la ejecución y no se bloquea la siguiente instrucción. Si la concesión de bus maestro no se declara (`xcpu_bmx.mst_wr_gnt = 0`), la instrucción en marcha reintentará (completa la ejecución), almacenando en memoria intermedia la escritura de datos. Las siguientes instrucciones se bloquean hasta el momento que se halle la concesión a declarar. La solicitud de escritura permanece declarada mientras la CPU 110 está bloqueada. En ambos casos, la CPU 110 comprobará el estado de la señal de concesión anterior a Q3 del siguiente ciclo. La CPU 110 retirará (completará) la instrucción en marcha pero almacenará en memoria intermedia la escritura de datos para finalización en algún tiempo más tarde (es decir después de que se declara la concesión). Para permitir que finalice cada instrucción después de que se inicie, la CPU puede ser el único bus maestro que pueda almacenar en memoria intermedia una escritura de datos. Todas las otras solicitudes de bus maestro se bloquean para tanto accesos de lectura como de escritura. Una lectura de bus maestro ganadora se completará en Q3 del ciclo después de la solicitud. Una escritura de bus maestro ganadora tendrá lugar en la Q1 que sigue a la solicitud de Q3.
- De acuerdo con una realización, las lecturas y escrituras de bus maestro pueden finalizar a una velocidad de una por ciclo de bus para la CPU 110. La CPU 110 emite la solicitud (para lectura o escritura) en el mismo ciclo que la finalizará (si se concede acceso al recurso solicitado).
- El caudal de datos para buses maestros de periféricos (es decir todos los otros buses maestros excepto la CPU) es asimétrico para lecturas y escrituras. Las solicitudes de lectura del bus maestro de periféricos que tendrán lugar en un ciclo finalizarán la transacción (si se concede acceso al recurso) en el siguiente ciclo. Sin embargo, las solicitudes de escritura de bus maestro de periféricos finalizarán la transacción (si se concede acceso al recurso) en el mismo ciclo que la solicitud.
- De acuerdo con una realización, los arbitradores de bus pueden no estar en tuberías, por lo que un bus maestro de periféricos debe completar una transacción solicitada antes de solicitar otro acceso de bus. El caudal de datos de bus maestro de periféricos máximo es por lo tanto una transacción cada 2 ciclos para lecturas (y una transacción cada ciclo para escrituras).
- Cada uno de los buses maestros de periféricos puede realizar cualquiera de una lectura o una escritura (pero no ambas) durante una transacción arbitrada que han ganado. Sin embargo, de acuerdo con una realización, las solicitudes de lectura y escritura pueden arbitrarse por separado. En consecuencia, una solicitud de lectura y una de escritura desde dos buses maestros puede servirse en un ciclo. La CPU 110 puede completar una lectura y/o una escritura en un ciclo de bus. Uno o ambos de estos accesos pueden arbitrarse.
- El registro de control de prioridad de bus maestro `MSTRPR<15:0>` del espacio de datos extendido (EDS), que puede localizarse en la CPU 110 puede usarse para modificar la prioridad de cada uno de los buses maestros de periféricos con relación a la de la CPU 110.
- De acuerdo con una realización, puede asignarse a cada bus maestro de periféricos un bit en el registro `MSTRPR`. Las asignaciones de bits para buses maestros específicos se fijan pero 'M' números asociados con cada bus maestro variarán entre variables, dependiendo de qué buses maestros estén presentes. Pueden añadirse buses maestros al registro `MSTRPR` (en localizaciones fijas) a medida que se desarrollan. De acuerdo con una realización, las entradas de bus maestro de arbitrador son siempre secuenciales (es decir M0, M1, M2, M3 etc.) en todas las variables.
- Cuando el bit asignado está establecido, el bus maestro de periféricos correspondiente se eleva para que esté por encima de el de la CPU 110. La prioridad de la CPU permanecerá por encima de la de los buses maestros de periféricos cuya prioridad no se ha elevado. De acuerdo con una realización, la prioridad relativa de los buses maestros de periféricos elevada por encima de la prioridad de la CPU permanecerá igual como estaba antes de que se elevara por encima de la de la CPU 110. De manera similar, la prioridad relativa de los buses maestros de periféricos no elevados por encima de la prioridad de la CPU permanecerá igual como estaba antes de que se elevara por encima de la de la CPU 110. El efecto del registro `MSTRPR` es como se muestra en la tabla mostrada en la Figura 2 donde se están arbitrando 5 buses maestros.
- De acuerdo con una realización, no todos los dispositivos pueden soportar Ethernet y/o periféricos USB, y la prioridad relativa asignada de estos buses maestros puede diferir de la mostrada en el ejemplo.
- El parámetro `NUM-BUS-MASTERS` define el número de buses maestros soportados por cada uno de los arbitradores de RAM de DS/EDS. De acuerdo con una realización, puede requerirse un arbitrador de lectura y uno de escritura separados. Se muestra una tabla de verdad de arbitrador de RAM en la tabla representada en las Figuras 9 y 10 para `NUM-BUS_MASTERS=4`, y se aplica a tanto el arbitrador de lectura como al de escritura. Arbitra acceso de bus maestro a todo el espacio de DS/EDS excepto el espacio de direcciones de SFR.
- De acuerdo con una realización, los buses maestros de la CPU e ICD siempre están presentes. Los restantes buses maestros presentes en la tabla mostrada en la Figura 9 y 10 son el USB y DMAC que se asignan a los bits de registro de prioridad de bus maestro `MSTRPR<3>` y `MSTRPR<5>` respectivamente. Todos los otros bits `MSTRPR` se reservan o no se usan y por lo tanto son bits 'no preocupantes'.

Un diagrama de bloques para los arbitradores de RAM y bloque de decodificación de direcciones con respectivas interconexiones de señal se muestra en la Figura 8. De acuerdo con una realización, la CPU 110 siempre se asigna a la entrada M0 del arbitrador 130 y 135. Todos los otros maestros se asignan a otras entradas de arbitrador (M1, M2 etc. sin huecos) en orden descendente de prioridad como se define mediante la especificación del dispositivo. En consecuencia, puede asignarse el mismo bus maestro a diferentes puertos de entrada de arbitrador (Mx) en diferentes variables.

Todos los buses maestros deben acceder a RAM de DS/EDS mediante el arbitrador 130, 135 de RAM (la DPSRAM trasera para el DMAC se considera una excepción de acuerdo con una realización). De acuerdo con una realización, el bus maestro de ICC puede únicamente obtener acceso al DS/EDS cuando ningún otro maestro está solicitando acceso. La prioridad de bus maestro de ICD nunca puede elevarse por encima de la de cualquier otro maestro.

De acuerdo con una realización, se supone que todos los buses maestros de periféricos transferirán únicamente siempre datos a o desde direcciones en la SRAM de DS o DPSRAM (cualquier otra dirección sería indicativa de operación errónea). Se permitirán por lo tanto las solicitudes de bus para cualquier dirección de bus maestro. La decodificación de bloque de SRAM de DS y DPSRAM reside después del multiplexor de dirección de arbitrador, por lo que las direcciones fuera de este rango darán como resultado arbitración pero no tendrán efecto sobre la memoria de DS. Sin embargo, el acceso a DS/EDS no implementado dará como resultado una trampa de dirección ilegal de CPU 110 (si el bus maestro solicitante era la CPU 110) o una trampa de software genérica (si el bus maestro solicitante era cualquier bus maestro de periféricos excepto el ICD 105). Si el bus maestro de ICD 105 intenta acceder a DS/EDS no implementado, no tendrá lugar la trampa. La concesión se emite como es habitual pero una lectura devolverá todo 0, y una escritura no tendrá efecto. La CPU 110 debe enviar una señal al controlador de interrupciones para indicar que el bus maestro realizó la solicitud.

De acuerdo con una realización, en un procesador de señales digitales, las solicitudes de lectura de RAM X e Y de la CPU (`cpu_xram_rd` y `cpu_yram_rd`) pueden unirse en O lógica para crear una única señal de solicitud de lectura de RAM de CPU RAM (`xcpu_bmx.mst_rd_req`).

El decodificador 120 de dirección aproximada de SFR dirige la dirección de CPU a cualquiera del arbitrador 130, 135 de RAM o al arbitrador 140, 145 de bus de SFR basándose en un límite de dirección predeterminado para el espacio de SFR definido por el parámetro `SFR_BYTE_SIZE`. Las solicitudes de bus asociadas con una dirección de CPU que es menor que el límite de SFR, se encaminarán al arbitrador 140, 145 de SFR. Pero las solicitudes asociadas con una dirección de CPU que es mayor o igual que el límite de SFR, se encaminará al arbitrador 130, 135 de RAM.

La salida de los arbitradores 130, 135, 140, 145 se alimenta a los decodificadores 160 de dirección de RAM donde la dirección ganadora se encamina al espacio de direcciones de DS/EDS correcto (X o Y) basándose en los parámetros de límite de dirección X e Y `XRAM_START_ADDR`, `XRAM_ADDR_WIDTH` e `YRAM_ADDR_WIDTH`. El decodificador 160 de direcciones X/Y contiene biestables Q2 para muestrear y retener las señales de control de RAM que seleccionan el origen 170, 175 de datos de RAM. De acuerdo con una realización, la RAM 170, 175 X e Y pueden localizarse siempre para que sean contiguas en el DS. Por ejemplo, la RAM 175 Y se coloca inmediatamente después del final de la RAM 170 X como se muestra en la Figura 1.

Como se ha mencionado anteriormente, un procesador de señales digitales o microcontrolador puede no incluir DPSRAM. Los espacios de direcciones de SRAM y (frontal) DPSRAM pueden combinarse en un único espacio de direcciones de RAM X de DS/EDS para todos los buses maestros excepto para el DMAC 115. Puesto que de acuerdo con una realización, el DMAC 115 tiene exclusivamente acceso al bus trasero de DPSRAM, la arbitración para este bus maestro únicamente se requiere para direcciones que caen fuera del rango de direcciones DPSRAM (es decir, para SRAM únicamente). Se permite que otros buses maestros accedan al bus de frontal de DPSRAM mientras que el DMAC 115 está accediendo al bus trasero. DPSRAM 180 puede incluir lógica para detectar y señalar escrituras concurrentes inadvertidas a la misma dirección de DPSRAM desde ambos puertos.

DPSRAM 180 puede localizarse en la misma dirección para ambos accesos frontal y trasero. El decodificador 190 de dirección aproximada de DPSRAM examina tanto solicitudes de datos de lectura como de escritura desde la DMAC 115 y determina si tienen o no por objetivo el espacio de direcciones DPSRAM. Si se detecta un acceso de DPSRAM, la solicitud de lectura o escritura correspondiente se encamina directamente a la DPSRAM 180 y no a los arbitradores 130, 135, 140, 145 de RAM. El decodificador 190 de dirección aproximada de DPSRAM también debe emitir inmediatamente una concesión de bus a la DMAC 115 para permitir el acceso al puerto DPSRAM trasero para finalizar. De hecho, el arbitrador puede verse como que ignora esta solicitud y puede conceder acceso a otro bus maestro a DS/EDS, incluyendo acceso a la DPSRAM 180 mediante el bus frontal.

De acuerdo con una realización, el arbitrador 140, 145 de SFR puede soportar únicamente 2 buses maestros, CPU 110 y ICD 105, por lo que `NUM-BUS-MASTER = 2` (fijo). La CPU 110 siempre ganará cualquier arbitración, independientemente del estado de `cpu_arb_master_priority<7:0>` puesto que el ICD 105 siempre está a la prioridad más baja (y no puede bloquear el ciclo).

La salida resultante de los arbitradores 140, 145 de SFR será la señal de dirección, datos y lectura (o escritura) del bus maestro ganador. Para cumplir temporización para ciertos periféricos, la dirección de lectura ganadora se

retiene en un biestable P2. La decodificación de dirección de periféricos se completa en cada periférico.

5 De acuerdo con una realización, para cumplir objetivos de velocidad de dispositivo, todas las concesiones de bus de SFR en respuesta a solicitudes de lectura de SFR (con la excepción de direcciones de SFR de registro de CPU mapeadas en memoria como se explicará en más detalle a continuación) se ponen en tuberías para retardarlas en un ciclo. Los accesos de lectura de SFR darán como resultado un bloqueo de CPU y son por lo tanto 2 operaciones de ciclo.

10 Los registros de CPU mapeados en memoria pueden residir en el espacio de direcciones de SFR. Sin embargo, estos registros residen físicamente en la CPU 110. Todas las solicitudes de escritura de SFR se conceden inmediatamente (sometidas a arbitración satisfactoria), permitiendo que se completen las escrituras de SFR en 1 ciclo.

Un diagrama de bloques para los arbitradores de SFR y bloque de decodificación de direcciones se muestra en la Figura 11. Un diagrama de temporización de transacciones de ejemplo para los arbitradores de SFR se muestra en la Figura 5 (la CPU tiene siempre la prioridad más alta).

15 De acuerdo con una realización, el ICD 105 puede leer o escribir cualquier SFR pero las escrituras a SFR de la CPU se inhiben a través de firmware de ICD. Adicionalmente, si se activa cualquier protección de código, todos los accesos de SFR de ICD se inhiben mediante la macro del ICD de acuerdo con una realización, una trampa de CPU que desactiva las escrituras de CPU al espacio de SFR (`cpu_disable_memwrite = 1`) no deberían interferir con ninguna solicitud de escritura de SFR de ICD coincidente.

20 No están activos relojes de sistema en un modo de EN SUSPENSIÓN de baja potencia, por lo que todos los arbitradores 130, 135, 140, 145 están inactivos. En un modo EN ESPERA, los relojes Q de la CPU se desactivan mientras los relojes P periféricos permanecen activos. En consecuencia, de acuerdo con una realización, la CPU 110 no puede hacer ninguna solicitud de bus de arbitrador mientras está en el modo EN ESPERA. Puesto que todos los buses maestros de periféricos (incluyendo la macro de ICD) que operan usando relojes P, los arbitradores 130, 135, 140, 145 de bus de RAM y de SFR continúan operando de manera normal para estos buses maestros en modo EN ESPERA.

25 De acuerdo con una realización, en otro modo EN REPOSO de baja potencia, los relojes Q de la CPU corren a una fracción binaria (programable) de los relojes P periféricos. Como se ha mencionado anteriormente, todos los buses maestros de periféricos corren usando los relojes P. En consecuencia, la CPU 110 debe proponer todas las solicitudes de bus de arbitrador hasta el último ciclo (reloj P) de cada ciclo de reloj Q. El fallo al hacer esto daría como resultado que la CPU 110 monopolizara el arbitrador de bus, bloqueando todos los buses maestros de prioridad inferior durante la duración de cada ciclo de CPU que requiere acceso de bus.

30 Aunque las realizaciones de esta divulgación se han representado, descrito y se definen por referencia a realizaciones de ejemplo de la divulgación, tales referencias no implican una limitación sobre la divulgación, y no se ha de inferir tal limitación. La materia objeto desvelada es considerable de modificación, alteración y equivalentes en forma y función, como se le ocurrirá al experto en la materia en la técnica pertinente y que tiene el beneficio de esta divulgación. Las realizaciones representadas y descritas de esta divulgación son ejemplos únicamente.

REIVINDICACIONES

1. Un procesador digital, que comprende:

un bus maestro (110; M0) por defecto que tiene una prioridad más alta en un modo por defecto;
 una pluralidad de buses maestros (105, 115; M1, M2, M3, M4) secundarios que tienen prioridades asociadas, en
 5 el que la pluralidad de buses maestros (105, 115; M1, M2, M3, M4) secundarios tienen una relación de prioridad
 predeterminada entre sí;
 estando el procesador digital **caracterizado por** un arbitrador (130, 135, 140, 145) de espacio de datos, en el
 que el arbitrador (130, 135, 140, 145) de espacio de datos es programable en un modo no por defecto para
 10 elevar una prioridad de cualquiera de dichos buses maestros (105, 115; M1, M2, M3, M4) secundarios para que
 tenga una prioridad más alta que la prioridad del bus maestro (110; M0) por defecto mientras mantiene la
 relación de prioridad predeterminada a únicamente aquellos buses maestros (105, 115; M1, M2, M3, M4)
 secundarios para los que el nivel de prioridad también se ha elevado por encima de la prioridad del bus maestro
 (110; M0) por defecto.

2. El procesador digital de acuerdo con la reivindicación 1, en el que el arbitrador (130, 135, 140, 145) de espacio de
 15 datos concede acceso a la memoria (170, 175, 180) de datos y en el que la memoria (170, 175, 180) de datos
 comprende al menos uno de entre memoria de acceso aleatorio, memoria de puerto dual y registros de funciones
 especiales.

3. El procesador digital de acuerdo con la reivindicación 1 o 2, que comprende adicionalmente un registro (195) de
 20 prioridad de bus maestro que tiene bits de contraorden de prioridad, al menos uno de dichos buses maestros (105,
 115; M1, M2, M3, M4) secundarios está asignado a uno de dichos bits de contraorden de prioridad, en el que el
 registro (195) de prioridad de bus maestro especifica programablemente cuáles de la pluralidad de buses maestros
 (105, 115; M1, M2, M3, M4) secundarios tienen prioridades más altas o más bajas que la prioridad del bus maestro
 (110, M0) por defecto.

4. El procesador digital de acuerdo con la reivindicación 3, en el que establecer un bit de contraorden de prioridad
 25 provoca que se eleve un nivel de prioridad de un bus maestro (105, 115; M1, M2, M3, M4) secundario asociado por
 encima del nivel de prioridad del bus maestro (110; M0) por defecto.

5. El procesador digital de acuerdo con una de las reivindicaciones anteriores 2-4, que comprende adicionalmente
 un decodificador (160) de direcciones acoplado entre el arbitrador (130, 135, 140, 145) de espacio de datos y la
 memoria (170, 175, 180) de datos.

6. El procesador digital de acuerdo con una de las reivindicaciones anteriores, que comprende adicionalmente un
 30 decodificador (120) de dirección aproximada acoplado entre el arbitrador (130, 135, 140, 145) de espacio de datos y
 el bus maestro (110; M0) por defecto o los buses maestros (105, 115; M1, M2, M3, M4) secundarios.

7. El procesador digital de acuerdo con una de las reivindicaciones anteriores 2-6, en el que el procesador digital es
 35 un procesador de señales digitales y la memoria de acceso aleatorio se divide en una memoria X (170) y una
 memoria Y (175).

8. El procesador digital de acuerdo con una de las reivindicaciones anteriores, en el que el bus maestro por defecto
 es una unidad de procesamiento central (CPU).

9. Un procedimiento para arbitrar acceso de espacio de datos en un procesador digital, que comprende:

asignar prioridades a un bus maestro (110; M0) por defecto y a una pluralidad de buses maestros (105, 115; M1,
 40 M2, M3, M4) secundarios, en el que el bus maestro (110; M0) por defecto tiene la prioridad más alta en un modo
 por defecto y en el que la pluralidad de buses maestros (105, 115; M1, M2, M3, M4) secundarios tienen una
 relación de prioridad predeterminada entre sí; estando el procedimiento **caracterizado por** la etapa de
 programar un arbitrador (130, 135, 140, 145) de espacio de datos en un modo no por defecto para elevar una
 45 prioridad de cualquiera de dichos buses maestros (105, 115; M1, M2, M3, M4) secundarios para que tenga una
 prioridad más alta que la prioridad del bus maestro (110; M0) por defecto mientras mantiene la relación de
 prioridad predeterminada a únicamente aquellos buses maestros (105, 115; M1, M2, M3, M4) secundarios para
 los que el nivel de prioridad también se ha elevado por encima de la prioridad del bus maestro (110; M0) por
 defecto.

10. El procedimiento de acuerdo con la reivindicación 9, que comprende adicionalmente: solicitar acceso a un
 50 espacio de datos mediante un bus maestro (105, 115; M1, M2, M3, M4) secundario mientras dicho bus maestro
 (110; M0) por defecto tiene acceso a dicho espacio de datos y si la prioridad del bus maestro (105, 115; M1, M2, M3,
 M4) secundario es más alta que la prioridad del bus maestro (110; M0) por defecto, conceder acceso al espacio de
 datos bloqueando de otra manera el bus maestro (105, 115; M1, M2, M3, M4) secundario.

11. El procedimiento de acuerdo con la reivindicación 9 o 10, en el que el espacio de datos es memoria (170, 175,
 55 180) de datos que comprende al menos uno de memoria de acceso aleatorio, memoria de puerto dual y registros de

funciones especiales.

5 12. El procedimiento de acuerdo con una de las reivindicaciones anteriores 9-11, en el que dicha etapa de programar un arbitrador (130, 135, 140, 145) de espacio de datos para elevar una prioridad de dichos buses maestros (105, 115; M1, M2, M3, M4) secundarios comprende la etapa de establecer un bit en un registro (195) de prioridad de bus maestro.

10 13. El procedimiento de acuerdo con la reivindicación 12, en el que los buses maestros (105, 115; M1, M2, M3, M4) secundarios están cada uno asignados a un bit en dicho registro (195) de prioridad de bus maestro y el registro (195) de prioridad de bus maestro permite que se establezcan múltiples bits para elevar el nivel de prioridad de múltiples de dichos buses maestros (105, 115; M1, M2, M3, M4) secundarios por encima del nivel de dicho bus maestro (110; M0) por defecto.

14. El procedimiento de acuerdo con la reivindicación 12 o 13, en el que al menos uno de dichos buses maestros (105, 115; M1, M2, M3, M4) secundarios tiene un bit asociado en dicho registro (195) de prioridad de bus maestro.

15 15. El procedimiento o procesador digital de acuerdo con una de las reivindicaciones anteriores, en el que el arbitrador (130, 135, 140, 145) de espacio de datos comprende una unidad arbitadora separada para accesos de lectura y escritura y/o el arbitrador (130, 135, 140, 145) de espacio de datos comprende una unidad arbitadora separada para accesos de memoria y de registro de funciones especiales.

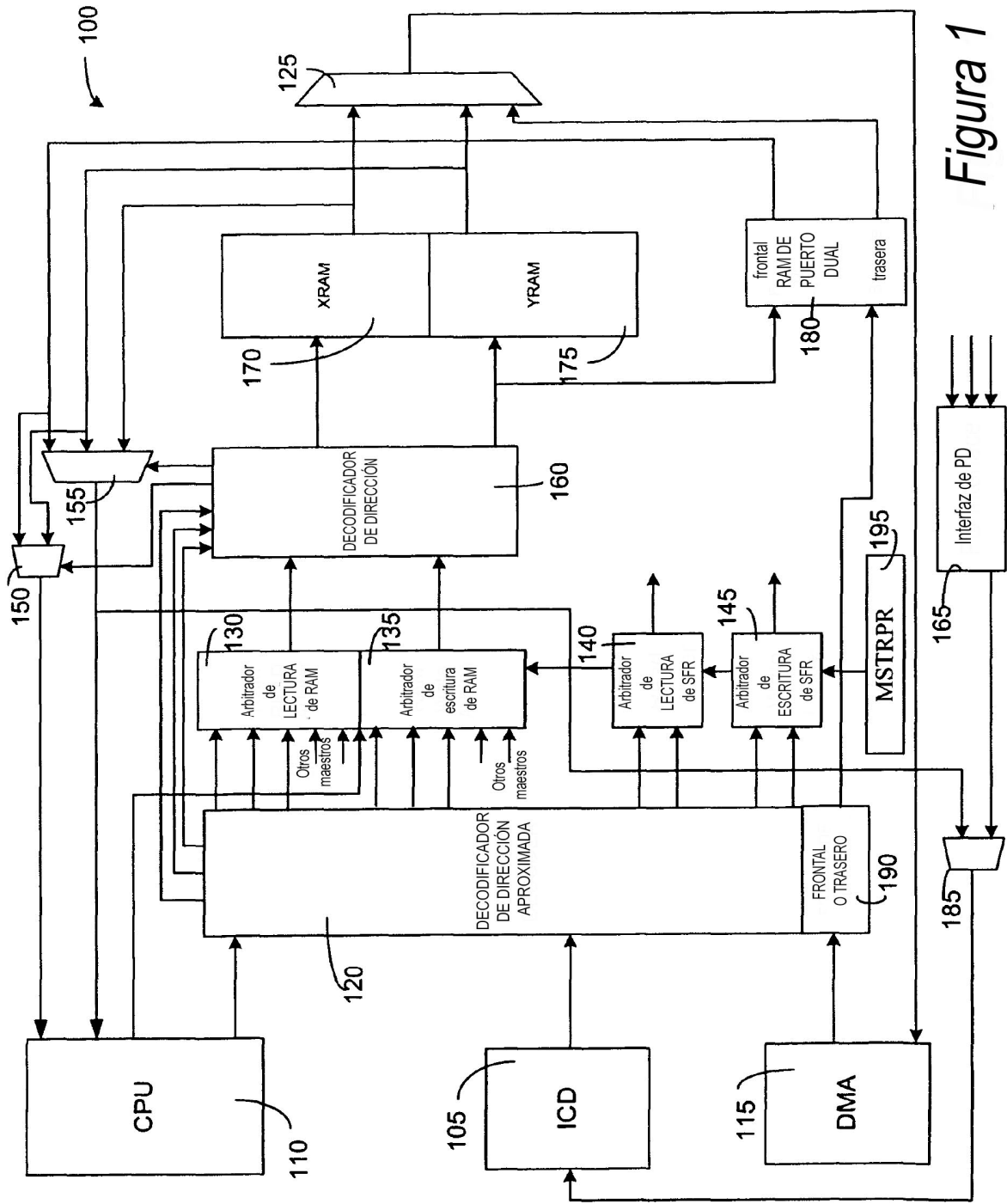


Figura 1

MSTRPR<15:0>	0x0000	0x0002	0x0008	0x0020	0x000A	0x0022	0x0028	0x002A
Prioridad de bus maestro en arbitrador	Por defecto	Ethernet ▲	USB ▲	DMA ▲	Ethernet+ USB ▲	Ethernet+ DMA ▲	USB+DMA ▲	Ethernet+ USB+DMA ▲
1 (más alta)	CPU (M0)	Ethernet	USB	DMA	Ethernet	Ethernet	USB	Ethernet
2	Ethernet (M1)	CPU	CPU	CPU	USB	DMA	DMA	USB
3	USB (M2)	USB	Ethernet	Ethernet	CPU	CPU	CPU	DMA
4	DMA (M3)	DMA	DMA	USB	DMA	USB	Ethernet	CPU
5 (más baja)	ICD (M4)	ICD	ICD	ICD	ICD	ICD	ICD	ICD

Figura 2

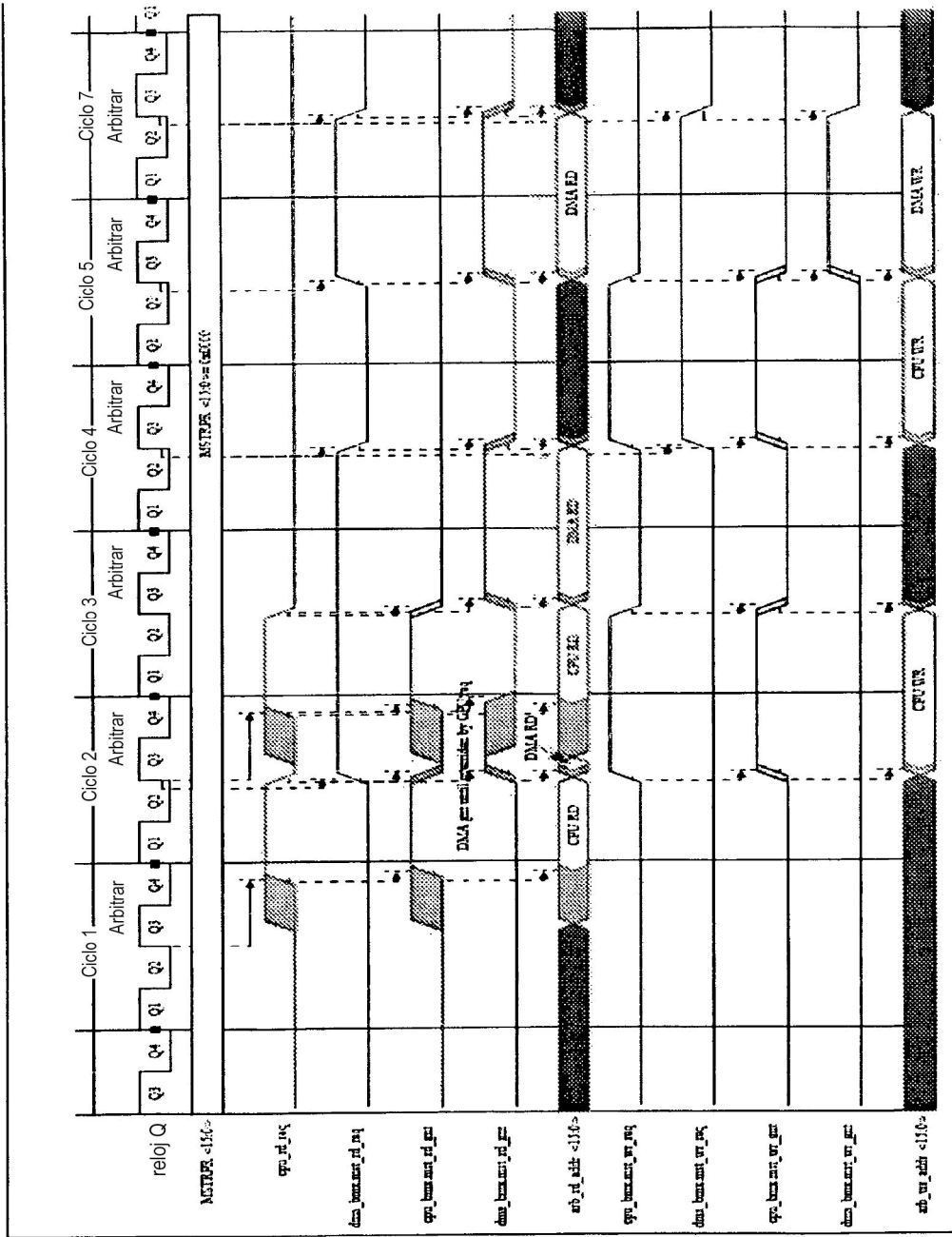


Figura 3

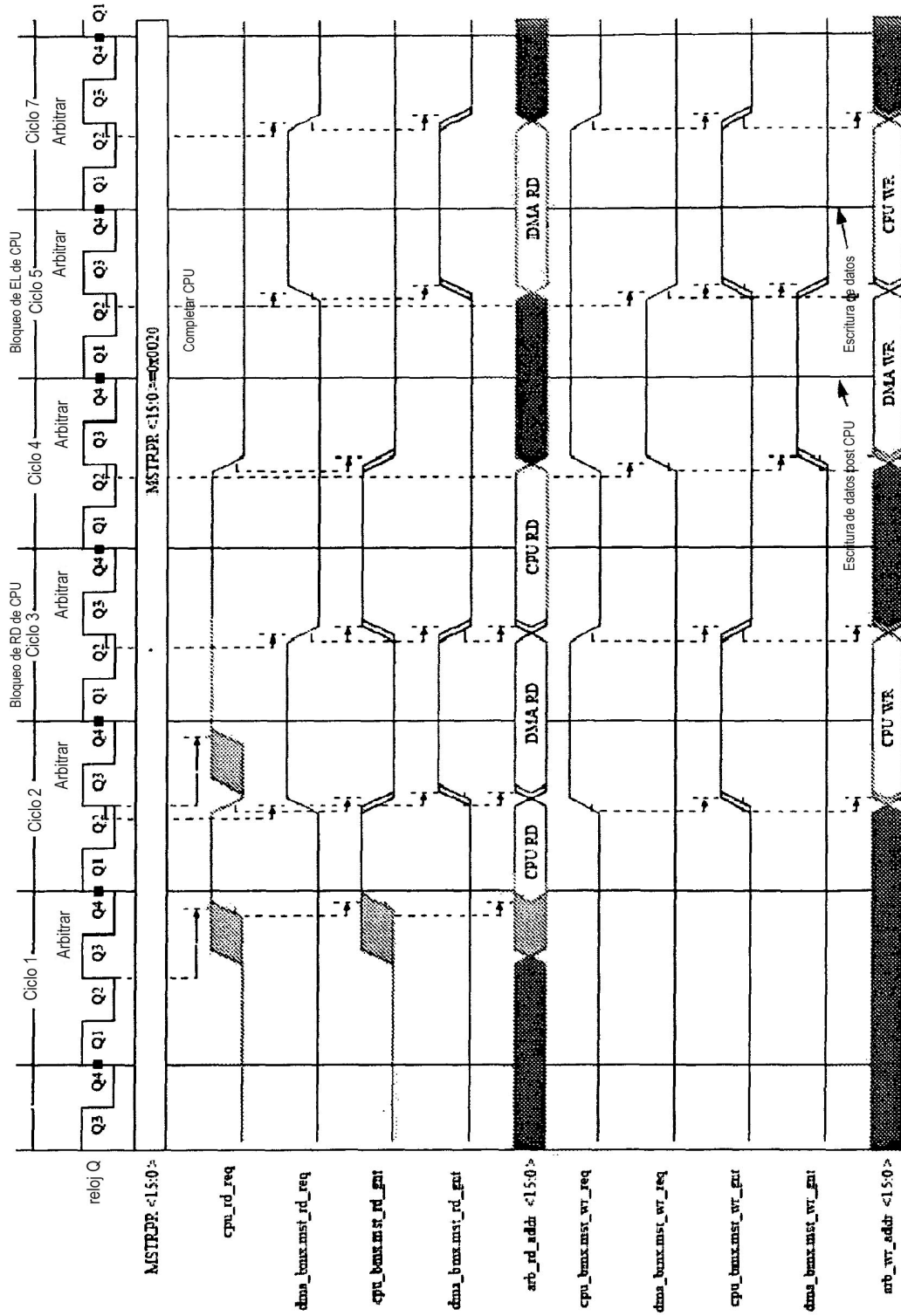


Figura 4

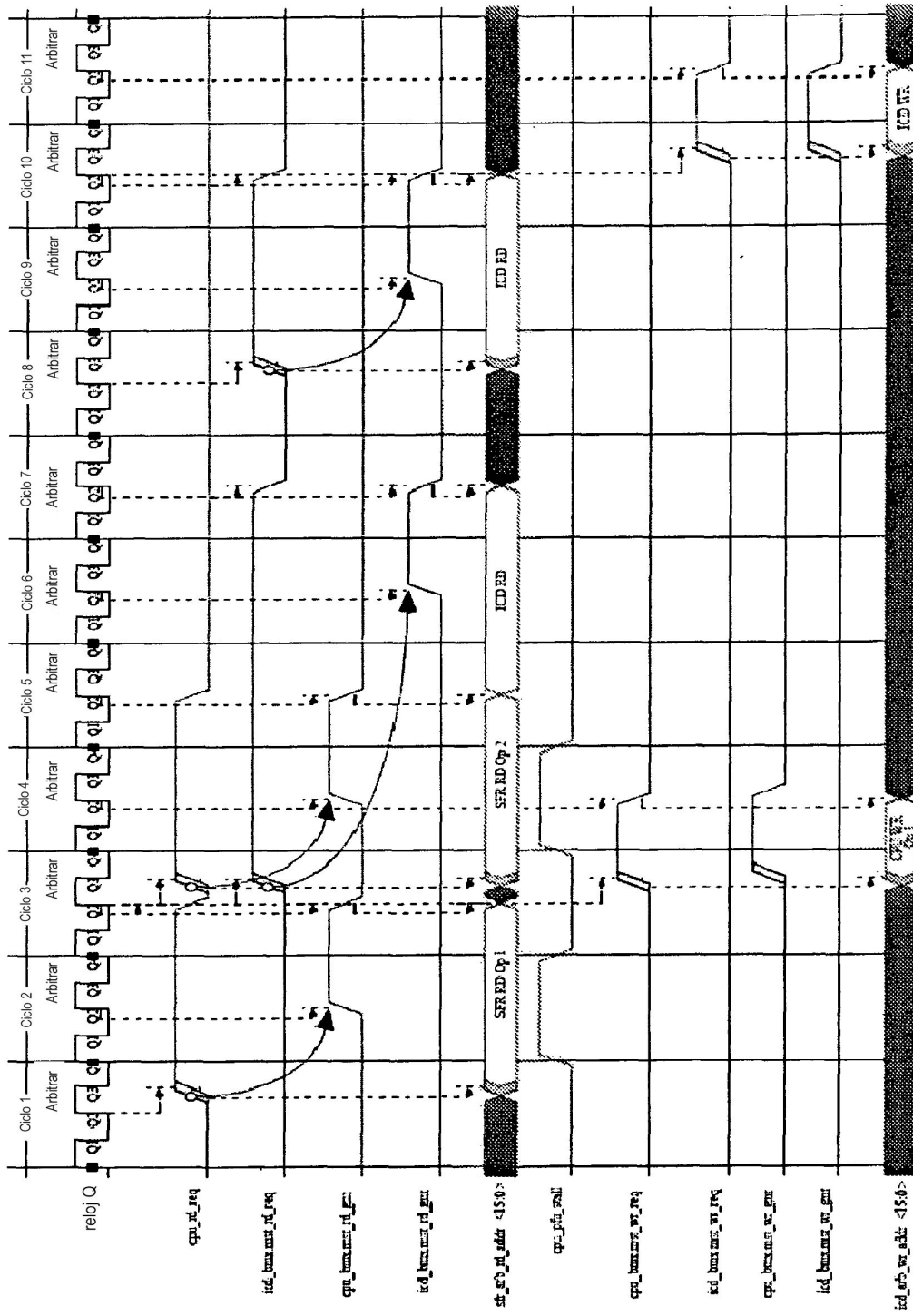


Figura 5

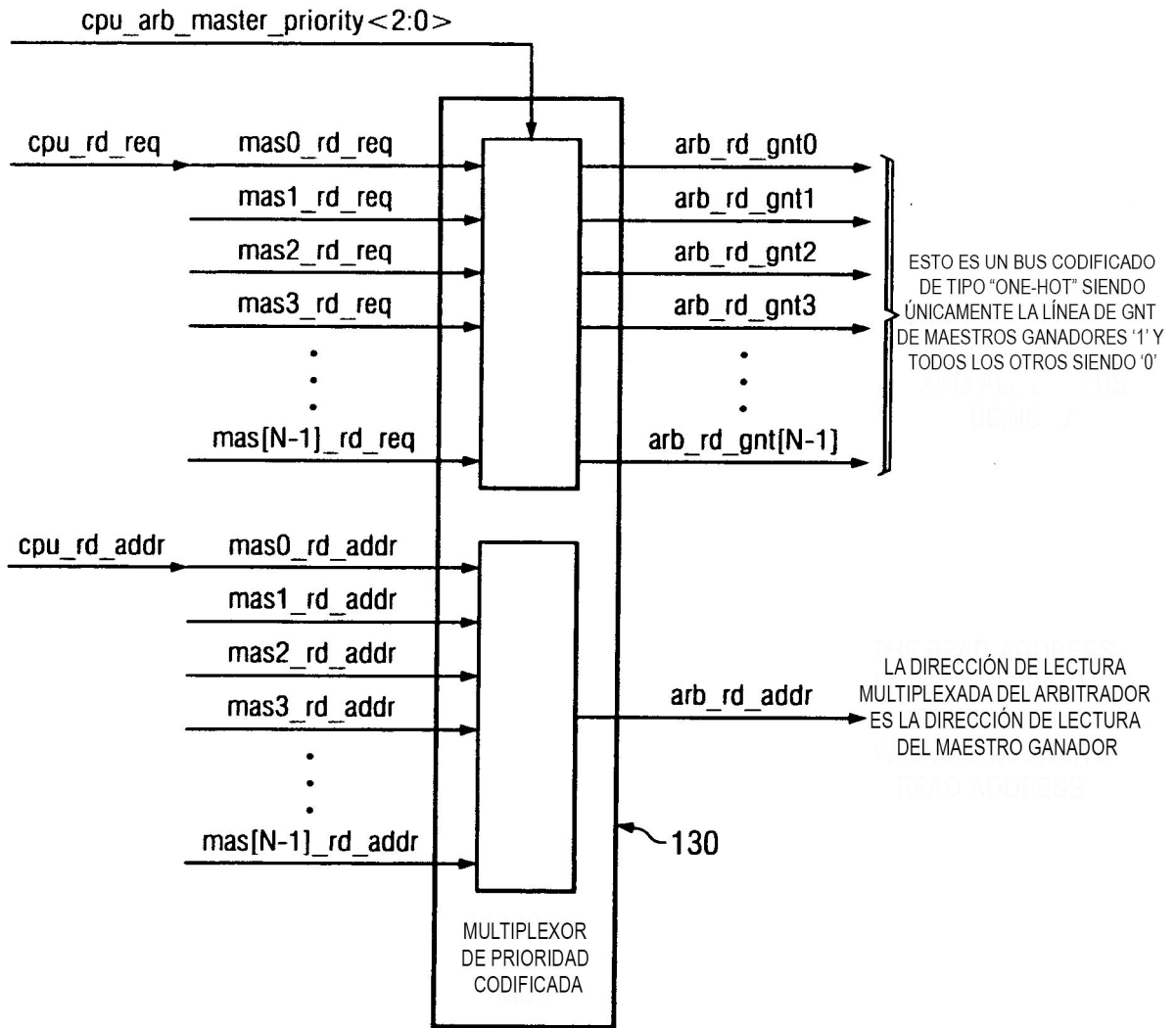


Figura 6

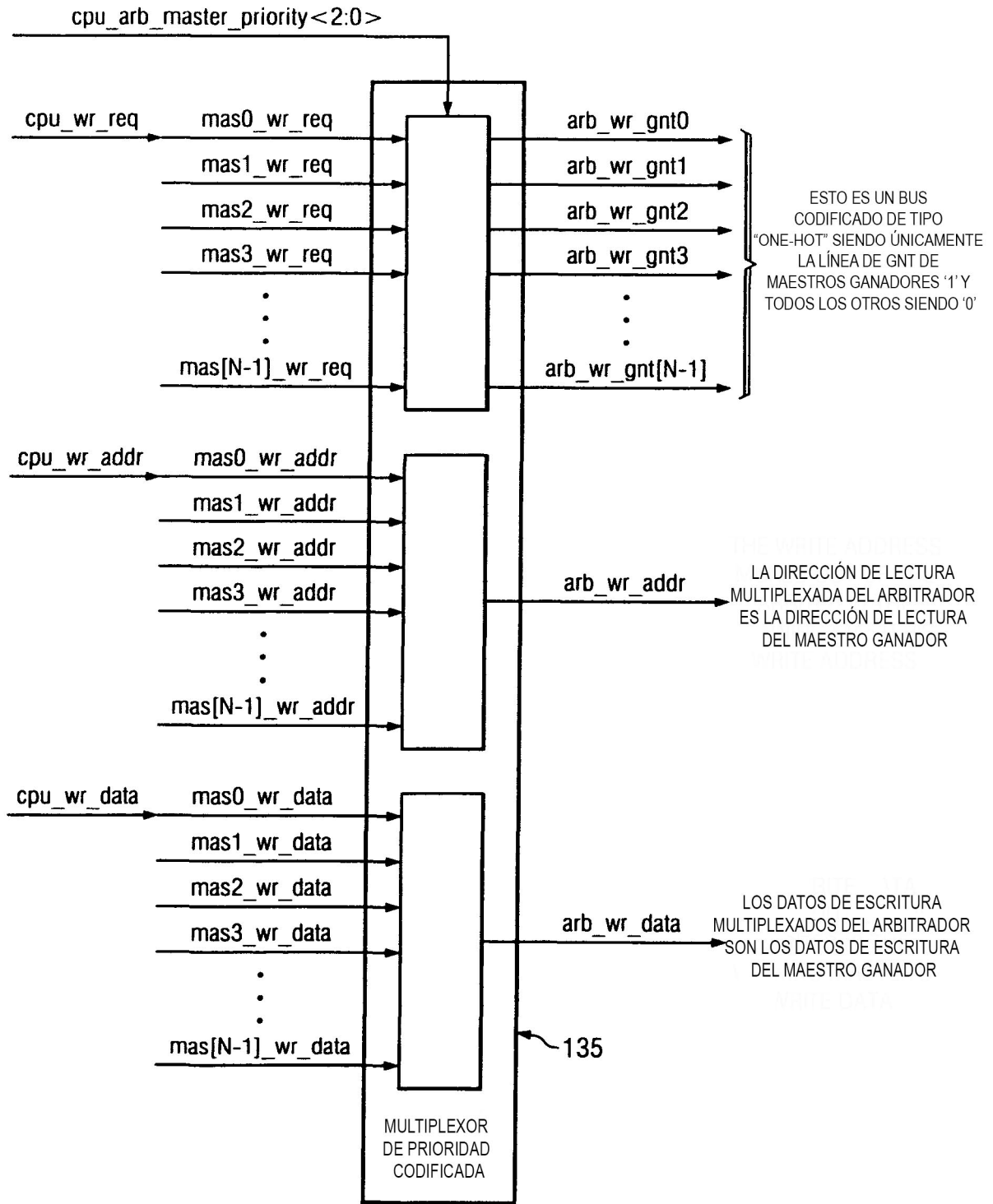


Figura 7

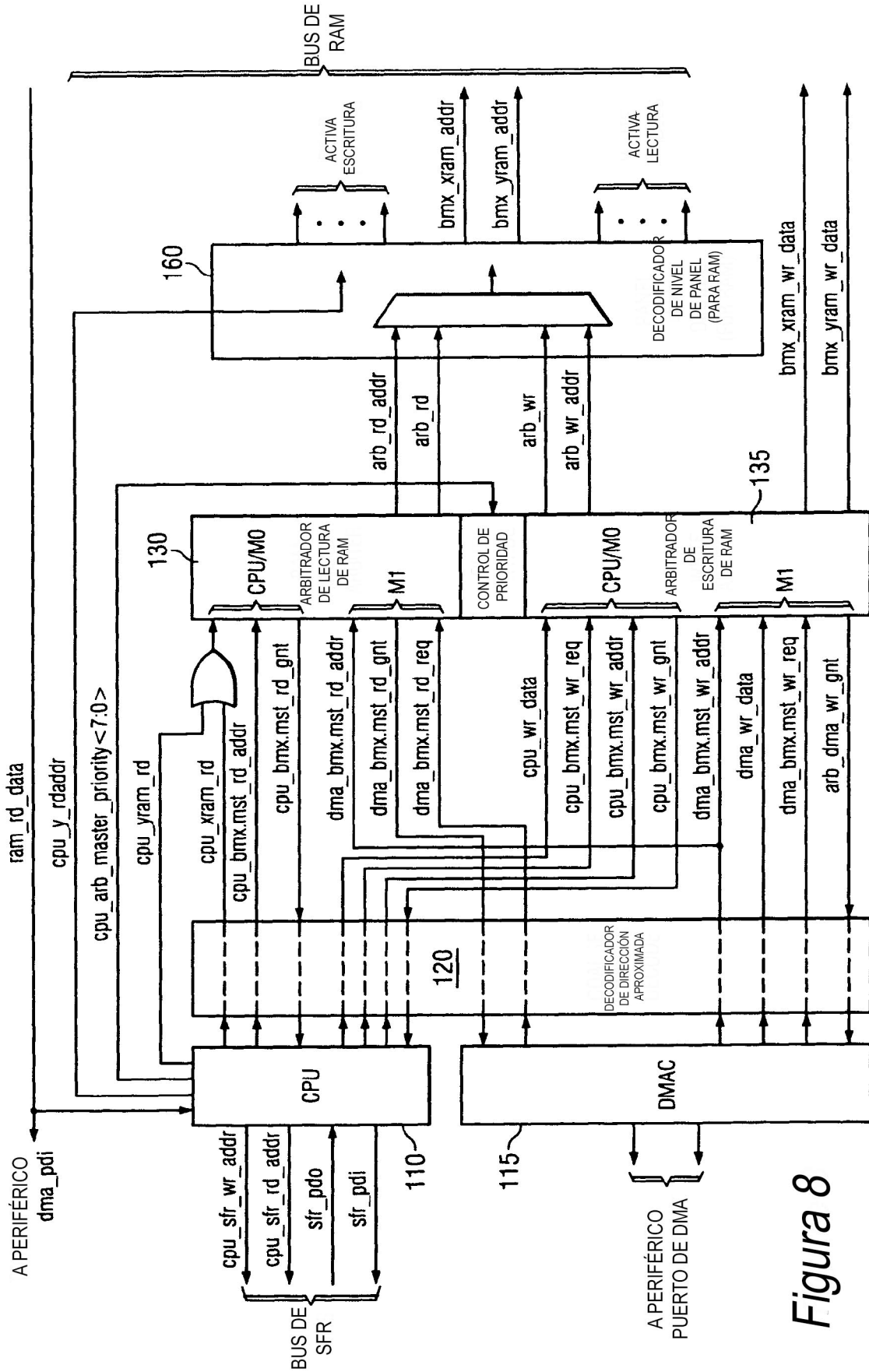


Figura 8

MSTRPR<7>P<7> cpu_arp_master_priority<7>P<7> (see text)	Solicitud ¹				Concesión				Bus seleccionado (siguiente ciclo)
	CPU (M0)	USB (M1)	DMAC (M2)	ICD (M3)	CPU (M0)	USB (M1)	DMAC (M2)	ICD (M3)	
0x00	0	0	0	0	0	0	0	0	CPU ²
0x00	0	0	0	1	0	0	0	1	ICD
0x00	0	0	1	0	0	0	0	1	DMAC
0x00	0	0	1	1	0	0	0	0	DMAC
0x00	0	1	0	0	0	1	0	0	USB
0x00	0	1	0	1	0	0	0	0	USB
0x00	0	1	1	0	0	1	0	0	USB
0x00	0	1	1	1	0	0	0	0	USB
0x00	1	0	0	0	1	0	0	0	CPU
0x00	1	0	0	1	1	0	0	0	CPU
0x00	1	0	1	0	1	0	0	0	CPU
0x00	1	0	1	1	1	0	0	0	CPU
0x00	1	1	0	0	1	0	0	0	CPU
0x00	1	1	1	0	1	0	0	0	CPU
0x00	1	1	1	1	1	0	0	0	CPU
0x08	0	0	0	0	0	0	0	0	CPU ²
0x08	0	0	0	1	0	0	0	1	ICD
0x08	0	0	1	0	0	0	0	1	DMAC
0x08	0	0	1	1	0	0	0	0	DMAC
0x08	0	1	0	0	0	1	0	0	USB
0x08	0	1	0	1	0	0	0	0	USB
0x08	0	1	1	0	0	0	0	0	USB
0x08	0	1	1	1	0	0	0	0	USB
0x08	1	0	0	0	1	0	0	0	CPU
0x08	1	0	1	0	1	0	0	0	CPU
0x08	1	0	1	1	1	0	0	0	CPU
0x08	1	1	0	0	1	0	0	0	USB
0x08	1	1	0	1	0	0	0	0	USB
0x08	1	1	1	0	1	0	0	0	USB
0x08	1	1	1	1	1	0	0	0	USB

Proridad más alta de la CPU (por defecto)

Elevar proridad de USB

Figura 9

METRPR<7>D- cpu_nb_mastw_priority<7>D- (véase texto)	Solicitud ¹			Concesión			Bus seleccionado (siguiente ciclo)		
	CPU (M0)	USB (M1)	DMAC (M2)	ICD (M3)	CPU (M0)	USB (M1)		DMAC (M2)	ICD (M3)
0x20	0	0	0	0	0	0	0	0	CPU ²
0x20	0	0	0	1	0	0	0	1	ICD
0x20	0	0	1	0	0	0	1	0	DMAC
0x20	0	0	1	1	0	0	1	0	DMAC
0x20	0	1	0	0	0	0	1	0	USB
0x20	0	1	0	1	0	0	0	0	USB
0x20	0	1	1	0	0	0	1	0	DMAC
0x20	0	1	1	1	0	0	1	0	DMAC
0x20	1	0	0	0	1	0	0	0	CPU
0x20	1	0	0	1	1	0	0	0	CPU
0x20	1	0	1	0	0	1	0	0	DMAC
0x20	1	0	1	1	0	0	1	0	DMAC
0x20	1	1	0	0	1	0	0	0	CPU
0x20	1	1	0	0	1	0	0	0	CPU
0x20	1	1	1	0	0	0	0	0	DMAC
0x20	1	1	1	1	0	0	1	0	DMAC
0x28	0	0	0	0	0	0	0	0	CPU ²
0x28	0	0	0	1	0	0	0	1	ICD
0x28	0	0	1	0	0	0	1	0	DMAC
0x28	0	1	0	0	0	0	1	0	DMAC
0x28	0	1	0	0	0	0	1	0	USB
0x28	0	1	0	1	0	0	0	0	USB
0x28	0	1	1	0	0	0	0	0	USB
0x28	0	1	1	1	0	0	0	0	USB
0x28	1	0	0	0	1	0	0	0	CPU
0x28	1	0	0	1	0	0	0	0	CPU
0x28	1	0	1	0	0	0	1	0	DMAC
0x28	1	0	1	0	0	0	1	0	DMAC
0x28	1	1	0	0	0	0	1	0	USB
0x28	1	1	0	1	0	0	0	0	USB
0x28	1	1	1	0	0	0	0	0	USB
0x28	1	1	1	1	0	0	0	0	USB

Elevar prioridad de DMAC

Elevar prioridad de USB + DMAC

Figura 10

Nota: CPU obtendrá una concesión si gana la arbitración de SFR o de RAM
 Nota: ICD obtendrá una concesión si gana la arbitración de SFR o de RAM

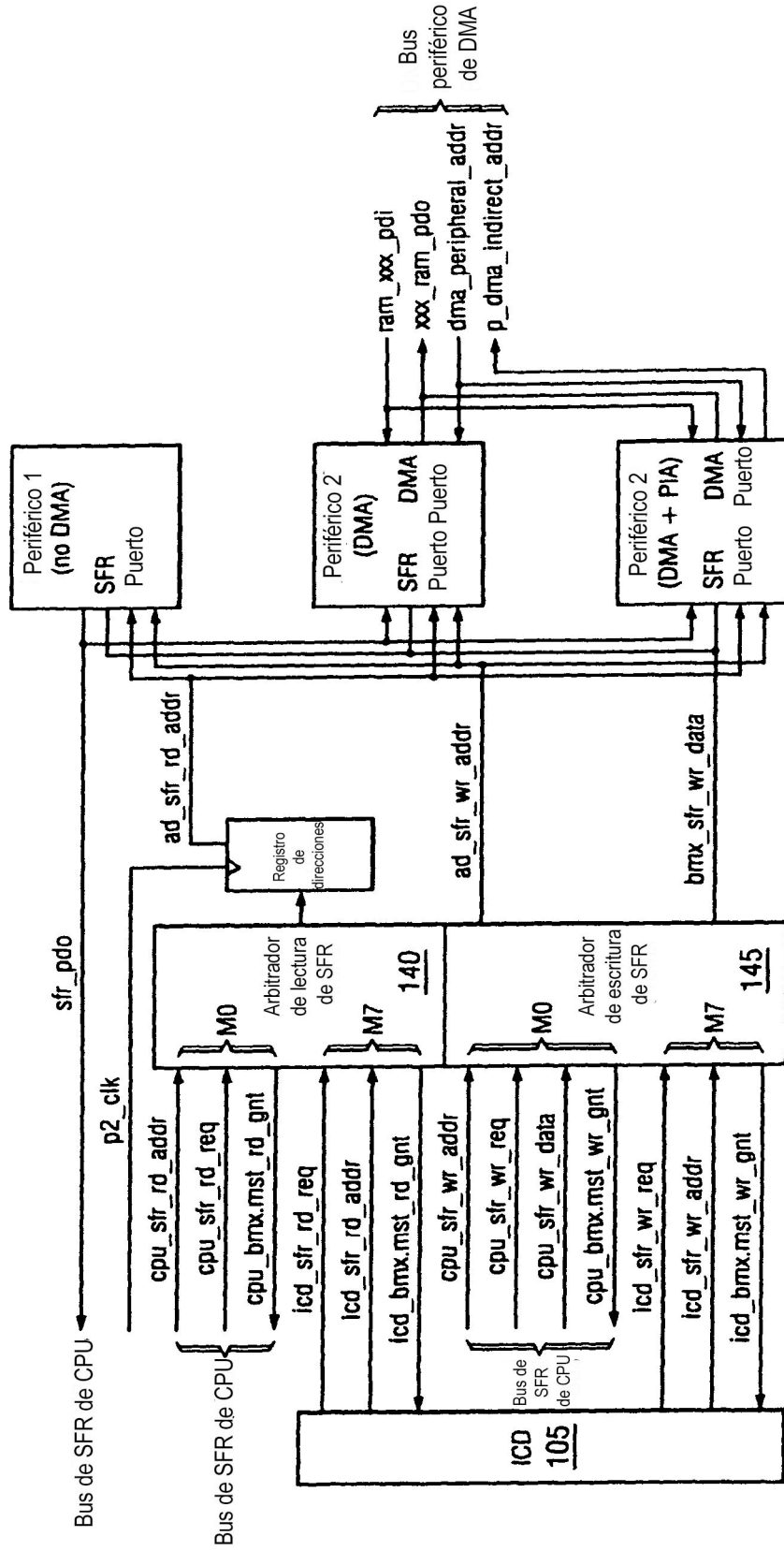


Figura 11