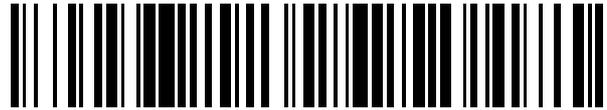


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 478 274**

51 Int. Cl.:

G11C 5/00 (2006.01)

G11C 7/22 (2006.01)

G11C 8/12 (2006.01)

G06F 13/16 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **11.02.2009 E 09001914 (2)**

97 Fecha y número de publicación de la concesión europea: **09.04.2014 EP 2251872**

54 Título: **Difusión general selectiva de datos en dispositivos conectados en serie**

30 Prioridad:

04.02.2008 US 25916 P

20.10.2008 US 254315

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

21.07.2014

73 Titular/es:

**MOSAID TECHNOLOGIES INCORPORATED
(100.0%)**

**11 Hines Road, Suite 203
Ottawa, ON K2K 2X1, CA**

72 Inventor/es:

PYEON, HONG BEOM

74 Agente/Representante:

CURELL AGUILÁ, Mireia

ES 2 478 274 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

Difusión general selectiva de datos en dispositivos conectados en serie.

5 **Campo de la invención**

La presente invención se refiere en general a un sistema de dispositivos de memoria conectados en serie en una topología en anillo con un controlador de memorias. Más particularmente, la presente invención se refiere a la selección de dispositivos de memoria en un sistema de dispositivos de memoria conectados en serie en una topología en anillo con un controlador de memorias.

Antecedentes de la invención

La memoria flash es un tipo de memoria no volátil usado comúnmente y ampliamente extendido como medios de almacenamiento de gran capacidad para electrónica de consumo, tal como, por ejemplo, cámaras digitales y reproductores digitales portátiles de música. La densidad de un componente de memoria flash disponible en la actualidad, que consta de 2 dados (*die*s) apilados, puede ser de hasta 32 Gbits (4 GB), lo cual resulta adecuado para su uso en unidades de almacenamiento flash USB populares, puesto que el tamaño de un componente flash es típicamente pequeño.

La aparición de cámaras digitales de 8 megapíxeles, dispositivos digitales portátiles de entretenimiento con capacidades de música y vídeo, y unidades de disco duro de estado sólido ha estimulado la demanda de capacidades ultra-altas para almacenar las grandes cantidades de datos a las que no puede llegar el dispositivo simple de memoria flash. Por lo tanto, múltiples dispositivos de memoria flash se combinan entre sí en un sistema para incrementar de manera efectiva la capacidad de almacenamiento disponible. Por ejemplo, para tales aplicaciones se pueden requerir densidades de almacenamiento flash de 20 GB o mayores.

La figura 1 es un diagrama de bloques de un sistema 10 de la técnica anterior en el que se ha integrado un sistema de anfitrión 12. El sistema 10 de la técnica anterior incluye un controlador de memorias 14 en comunicación con el sistema de anfitrión 12, y múltiples dispositivos de memoria no volátil 16. El sistema de anfitrión 12 incluye un dispositivo de procesamiento, tal como un microcontrolador, microprocesador, o un sistema de ordenador. El sistema 10 de la técnica anterior de la figura 1 está organizado de manera que incluye un canal 18, estando conectados en paralelo los dispositivos de memoria 16 al canal 18. Los expertos en la materia apreciarán que el sistema 10 de la técnica anterior puede tener, conectados al mismo, un número de dispositivos de memoria mayor o menor que cuatro.

El canal 18 incluye un conjunto de buses comunes, los cuales incluyen líneas de datos y de control que están conectadas a la totalidad de sus dispositivos de memoria correspondientes. Cada dispositivo de memoria se habilita o deshabilita con señales respectivas de selección (habilitación) de chips CE1#, CE2#, CE3# y CE4#, proporcionadas por el controlador de memorias 14. El “#” indica que la señal es una señal de nivel lógico bajo activo. A lo sumo, cada vez se selecciona una de las señales de selección de chip para habilitar uno de los dispositivos de memoria 16 durante una operación. El controlador de memorias 14 es responsable de emitir órdenes y datos, por medio del canal 18, a un dispositivo de memoria seleccionado, como respuesta al funcionamiento del sistema de anfitrión 12. La salida de datos de lectura de los dispositivos de memoria se transfiere, por medio del canal 18, de vuelta al controlador de memorias 14 y al sistema de anfitrión 12. El funcionamiento del sistema 10 de la técnica anterior puede ser asíncrono o síncrono. La figura 1 ilustra un ejemplo de un sistema síncrono que usa un reloj (CK), el cual se proporciona en paralelo a cada dispositivo de memoria 16 para sincronizar la transferencia de datos sobre el canal 18. En general, se dice que el sistema 10 de la técnica anterior incluye un bus *multi-drop*, en el cual dispositivos de memoria 16 están conectados en paralelo con respecto al canal 18.

Una alternativa a los buses *multi-drop* son buses en los cuales los dispositivos de memoria están conectados en serie, tal como se muestra en el documento US 2004/148 482”.

Existen problemas específicos que pueden tener un impacto negativo en el rendimiento del sistema. La estructura del sistema 10 de la técnica anterior plantea limitaciones físicas de rendimiento. Existe un número elevado de señales paralelas que se extienden por el sistema, y la integridad de señal correspondiente a las señales que transportan se puede ver deteriorada por la interferencia cruzada, el desfase de la señal, el efecto de carga de entrada y salida, y el ruido de conmutación simultánea (SSN). El consumo de energía de entrada/salida en un sistema de este tipo se convierte en un problema en la medida en la que cada pista de señal entre el controlador flash y los dispositivos de memoria flash se carga y descarga frecuentemente para la señalización. Con frecuencias cada vez mayores del reloj del sistema, el consumo de energía aumentará.

Existe también un límite práctico sobre el número de dispositivos de memoria que se pueden conectar en paralelo al canal puesto que la capacidad de accionamiento de un único dispositivo de memoria es pequeña con respecto a la carga de las largas pistas de señal. Además, a medida que aumente el número de dispositivos de memoria, se requieren más señales de habilitación de chip (CE#), y puede que sea necesario encaminar CK a los dispositivos de

memoria adicionales, siendo todas ellas más largas puesto que se encaminan a los dispositivos de memoria. Los problemas de rendimiento del reloj debidos a una distribución extensiva del mismo son bien conocidos en la técnica, y se han convertido en una cuestión problemática en sistemas de gran tamaño de la técnica anterior con muchos dispositivos de memoria 16. Por lo tanto, para que un sistema de memoria de la técnica anterior incluya un gran número de dispositivos de memoria, o bien los dispositivos de memoria se esparcen por múltiples canales o bien el accionamiento por frecuencia del sistema de memoria debería limitarse; cualquiera de las dos opciones implica compromisos. Un controlador que tenga múltiples canales y señales adicionales de habilitación de chip hace que aumente el coste del sistema. Alternativamente, el sistema resulta limitado a un número pequeño de dispositivos de memoria.

Por lo tanto, el sistema de memoria 10 de la figura 1 no resultaría adecuado como unidad de disco de estado sólido, puesto que su densidad será limitada y su rendimiento impondrá un cuello de botella en el rendimiento del sistema de anfitrión 12. Adicionalmente, puede haber aplicaciones en las que la integridad de los datos tenga una importancia primordial. Una de las técnicas para garantizar la integridad de los datos es duplicarlos en una ubicación diferente, de manera que los datos alterados se puedan reconstruir o sustituir. En el sistema de memoria de la figura 1 por ejemplo, esto se realizaría escribiendo los mismos datos en dos dispositivos de memoria 16 diferentes. No obstante, debido a su limitación de densidad se utilizaría demasiada memoria para almacenar los datos duplicados, puesto que la densidad global del sistema de memoria se reduce efectivamente a la mitad. Por lo tanto, el sistema de memoria puede alcanzar rápidamente su capacidad de almacenamiento máxima.

Por lo tanto, es deseable proporcionar un sistema de memoria de alta densidad y alto rendimiento que pueda habilitar selectivamente múltiples dispositivos de memoria para ejecutar la misma orden, con una tara de tiempo mínima.

Sumario de la invención

Es un objetivo de la presente invención proporcionar un circuito y un método para permitir que un subconjunto de una pluralidad de dispositivos ejecute la misma orden que se proporciona en un paquete de datos.

En un primer aspecto, se proporciona un método para seleccionar un subconjunto de dispositivos de una pluralidad de dispositivos conectados en serie a un controlador de memorias con el fin de ejecutar una orden. El método incluye codificar números de identificación (ID) de cada uno del subconjunto de dispositivos para proporcionar información de código; proporcionar un paquete de orden que incluye un código de operación y la información de código a cada uno de la pluralidad de dispositivos; decodificar la información de código recibida por cada uno de la pluralidad de dispositivos, y ejecutar el código de operación si un número de ID almacenado en el dispositivo está codificado en la información de código recibida. En una forma de realización, el controlador de memorias proporciona 2^x de los números de ID, donde x es un entero mayor que cero.

En otra forma de realización, cada uno de los números de ID tiene una longitud de N bits, y la información de código incluye un número de ID de Destino de N bits de longitud y un código de máscara de N bits de longitud, donde N es un valor entero mayor que 0. En la forma de realización descrita en este momento, la etapa de codificación incluye seleccionar un número de ID del subconjunto de dispositivos como número de ID de Destino; comparar los números de ID de cada uno del subconjunto de dispositivos con el número de ID de Destino para generar números de ID codificados correspondientes; y combinar lógicamente un bit en la misma posición de bit de cada uno de los números de ID codificados para generar un bit que tiene la posición de bit correspondiente en el código de máscara. Puede haber M números de ID correspondientes al subconjunto de dispositivos, donde M es un valor entero mayor que 0. Además, la etapa de comparación incluye dar salida a los resultados de comparación en una matriz de $M \times N$, y transponer la matriz de $M \times N$ a una matriz de $N \times M$, donde el bit se genera ejecutando una función lógica sobre los resultados de la comparación en una de las N filas de la matriz de $N \times M$. Un primer bit del código de máscara generado a partir de la primera fila de la matriz de $N \times M$ representa un bit más significativo del código de máscara, y un último bit del código de máscara generado a partir de la última fila de la matriz de $N \times M$ representa un bit menos significativo del código de máscara. En una variante de la presente forma de realización, la etapa de comparación comprende una etapa de ejecutar una función XNOR entre los números de ID de cada uno del subconjunto de dispositivos con el número de ID de Destino. Además, la etapa de combinación lógica puede comprender una etapa de ejecutar una función NAND entre bits en la misma posición de bit de cada uno de los números de ID codificados.

En un aspecto de la presente forma de realización, la etapa de decodificación comprende una etapa de procesar el número de ID almacenado en el dispositivo con el número de ID de Destino y el código de máscara para proporcionar una indicación de que todas las posiciones de bit del número de ID almacenado en el dispositivo están codificadas en el código de máscara. La etapa de procesado incluye comparar bits de cada posición de bit del número de ID almacenado en el dispositivo con los bits correspondientes de cada posición de bit correspondiente del número de ID de Destino para generar un número de ID de destino codificado; adicionar el código de máscara al número de ID de destino codificado para proporcionar un número de ID de destino, codificado, modificado; y multiplicar los bits del número de ID de destino, codificado, modificado entre sí con el fin de proporcionar la indicación de que todas las posiciones de bit del número de ID almacenado en el dispositivo están codificadas en la máscara. La etapa de comparar los bits de cada posición de bit del número de ID almacenado en el dispositivo

puede comprender una etapa de ejecución de una función lógica XNOR; la etapa de adicionar el código de máscara al número de ID de destino, codificado, puede comprender una etapa de ejecutar una función lógica OR; y la etapa de multiplicar los bits del número de ID de destino, codificado, modificado puede comprender una etapa de ejecutar una función lógica AND.

5 En otra forma de realización del presente aspecto, la etapa de proporcionar el paquete de orden comprende etapas en las que cada uno de la pluralidad de dispositivos recibe el paquete de orden desde uno de entre un dispositivo previo y el controlador de memorias, y traslada el paquete de orden a uno de entre un dispositivo subsiguiente y el controlador de memorias, donde el dispositivo previo y el dispositivo subsiguiente incluyen dispositivos de memoria.
 10 Otra etapa incluye inhibir el traslado del paquete de orden a uno de entre el dispositivo subsiguiente y el controlador de memorias cuando el último dispositivo del subconjunto de dispositivos decodifica la información de código. La etapa de inhibición comprende combinar lógicamente el número de ID de Destino, el código de máscara y el número de ID almacenado en el dispositivo para proporcionar una señal de último dispositivo; y fijar una memoria intermedia de salida a un valor nulo como respuesta a la señal de último dispositivo. La etapa de combinación lógica puede
 15 incluir adicionar el número de ID de Destino al código de máscara para obtener una dirección de último dispositivo designado; y comparar la dirección de último dispositivo designado con el número de ID almacenado en el dispositivo, donde la señal de último dispositivo se genera cuando la dirección de último dispositivo designado coincide con el número de ID almacenado en el dispositivo. Además, la etapa de fijación comprende una etapa de acoplar una entrada de la memoria intermedia de salida a un nivel lógico estático como respuesta a la señal de
 20 último dispositivo.

En un segundo aspecto, se proporciona un dispositivo que tiene un número de identificación de dispositivo (ID) para recibir un paquete de orden. El dispositivo incluye un trayecto de flujo pasante de datos, un intérprete de órdenes y un procesador de ID. El trayecto de flujo pasante de datos recibe el paquete de orden en un puerto de entrada y
 25 traslada el paquete de orden a un puerto de salida. El paquete de orden incluye un código de operación e información de código correspondiente a un subconjunto de números de ID de dispositivo seleccionados para ejecutar el código de operación. El intérprete de órdenes se habilita selectivamente para decodificar el código de operación. El procesador de ID recibe la información de código y decodifica la información de código para habilitar el intérprete de órdenes si el número de ID de dispositivo está codificado en la información de código. En una forma de
 30 realización del presente aspecto, el trayecto de flujo pasante de datos incluye una memoria intermedia de entrada acoplada al puerto de entrada para almacenar temporalmente el paquete de orden, y una memoria intermedia de salida acoplada a la memoria intermedia de entrada para recibir el paquete de orden y para conducir el paquete de orden al puerto de salida. La información de código puede incluir un código de máscara y un número de ID de Destino. El procesador de ID puede incluir un decodificador de coincidencias de ID para combinar lógicamente el
 35 código de máscara, el número de ID de Destino y el ID de dispositivo con el fin de proporcionar una señal de coincidencia cuando el número de ID de dispositivo esté codificado en el código de máscara. El número de ID de Destino es uno del subconjunto de números de ID de dispositivo.

En otra forma de realización del presente aspecto, el decodificador de coincidencias de ID incluye un comparador,
 40 un sumador y un multiplicador. El comparador compara cada bit del número de ID de dispositivo con un bit correspondiente del número de ID de Destino para proporcionar un número de ID de Destino, codificado. El sumador suma cada bit del código de máscara a un bit correspondiente del número de ID de Destino, codificado, para proporcionar un número de ID de destino, codificado, modificado. El multiplicador proporciona la señal de coincidencia cuando todos los bits del número de ID de destino, codificado, modificado, se encuentran en un nivel
 45 lógico activo. En la presente forma de realización, el comparador incluye puertas lógicas XNOR, donde cada puerta lógica XNOR compara un bit del número de ID de dispositivo con un bit correspondiente del número de ID de Destino. El sumador puede incluir puertas lógicas OR, donde cada puerta lógica OR suma un bit del código de máscara a un bit correspondiente del número de ID de Destino, codificado. El multiplicador puede incluir un circuito lógico AND para recibir todos los bits del número de ID de destino, codificado, modificado.
 50

Todavía en otra forma de realización del presente aspecto, el procesador de ID incluye un circuito de inhibición de salida para inhibir la conducción del paquete de orden al puerto de salida. El circuito de inhibición de salida puede
 55 incluir un sumador y un decodificador de último ID. El sumador suma el código de máscara al número de ID de Destino para proporcionar un número de último ID de dispositivo del subconjunto de números de ID de dispositivo. El decodificador de último ID compara el número de ID de dispositivo con el número de último ID de dispositivo, y proporciona una señal de control de último ID cuando el número de ID de dispositivo coincide con el número de último ID de dispositivo. La memoria intermedia de salida se deshabilita como respuesta a la señal de control de último ID, y el trayecto de flujo pasante de datos incluye un selector para trasladar selectivamente uno de entre el
 60 paquete de orden y un valor nulo a la memoria intermedia de salida. El selector traslada el valor nulo como respuesta a la señal de control de último ID, donde el valor nulo se corresponde con un bit almacenado en un registro de datos de salida acoplado al selector.

En un tercer aspecto, se proporciona un sistema de memoria que incluye un controlador de memorias y una
 65 pluralidad de dispositivos de memoria conectados en serie en una topología en anillo con el controlador de memorias. El controlador de memorias emite un paquete de orden que incluye un código de operación e información de código correspondiente a un subconjunto de números de identificación (ID) de dispositivo seleccionados para

ejecutar el código de operación. Cada uno de la pluralidad de dispositivos de memoria tiene un número exclusivo de ID de dispositivo y recibe el paquete de orden y ejecuta el código de operación si el número exclusivo de ID de dispositivo está codificado en la información de código. En una forma de realización del presente aspecto, cada uno de la pluralidad de dispositivos de memoria incluye un trayecto de flujo pasante de datos, un intérprete de órdenes y un procesador de ID. El trayecto de flujo pasante de datos recibe el paquete de orden en un puerto de entrada y traslada el paquete de orden a un puerto de salida. El intérprete de órdenes se habilita selectivamente para decodificar el código de operación. El procesador de ID recibe la información de código y decodifica la información de código para habilitar el intérprete de órdenes cuando el número exclusivo de ID de dispositivo está codificado en la información de código. La información de código comprende un código de máscara y un número de ID de Destino. El procesador de ID incluye un decodificador de coincidencias de ID para combinar lógicamente el código de máscara, el número de ID de Destino y el número de ID de dispositivo con el fin de proporcionar una señal de coincidencia cuando el número exclusivo de ID de dispositivo está codificado en el código de máscara. El procesador de ID comprende un circuito de inhibición de salida para inhibir la conducción del paquete de orden al puerto de salida cuando el número exclusivo de ID de dispositivo coincide con un número de último ID de dispositivo del subconjunto de números de ID de dispositivo.

Otros aspectos y características de la presente invención resultarán evidentes para los expertos en la materia a partir de la siguiente descripción de formas de realización específicas de la invención, en combinación con las figuras adjuntas.

Breve descripción de los dibujos

A continuación se describirán las formas de realización de la presente invención, únicamente a título de ejemplo, y haciendo referencia a las figuras adjuntas, en las cuales:

la figura 1 es un diagrama de bloques de un sistema de la técnica anterior en el que se ha integrado un sistema de anfitrión;

la figura 2A es un diagrama de bloques de un sistema que recibe una señal de reloj en paralelo;

la figura 2B es un diagrama de bloques del mismo sistema de la figura 2A que recibe una señal de reloj síncrono de origen;

la figura 3 ilustra la configuración de bytes de dispositivos conectados en serie;

la figura 4 ilustra un sistema de acuerdo con una forma de realización de la presente invención;

la figura 5 es un diagrama de flujo que ilustra cómo el sistema de memoria selecciona más de un dispositivo de memoria usando un único paquete de orden;

la figura 6 es un diagrama de bloques de un sistema de memoria de acuerdo con una forma de realización de la presente invención;

la figura 7A es un diagrama de bloques que ilustra los bloques de circuito responsables de generar la información de código usada para seleccionar múltiples dispositivos de memoria en el sistema de memoria de la figura 6;

la figura 7B muestra posibles campos de orden ejemplificativos de un paquete de orden;

la figura 8 ilustra un ejemplo de un identificador de dispositivo de acuerdo con una forma de realización de la presente invención;

la figura 9 es un esquema de circuito que ilustra un ejemplo de un codificador de máscaras de acuerdo con una forma de realización de la presente invención;

la figura 10A es una tabla que ilustra una operación de ejemplo del codificador de máscaras de la figura 9;

la figura 10B es una tabla que ilustra otra operación de ejemplo del codificador de máscaras de la figura 9;

la figura 11 es una matriz que ilustra una operación de OR lógica de direcciones seleccionadas, con el código de máscara;

la figura 12 es un diagrama de bloques que ilustra una forma de realización de una interfaz de entrada/salida de datos de un dispositivo de memoria diseñado para una interconexión en serie con otro dispositivo de memoria o un controlador de memorias;

la figura 13 es un esquema de circuito que ilustra, a título de ejemplo, un decodificador de coincidencias de ID y

un decodificador de últimos ID;

la figura 14 es un diagrama de flujo que muestra el funcionamiento del sistema de memoria de las presentes formas de realización como respuesta a un paquete de orden que direcciona selectivamente un subconjunto de los dispositivos de memoria; y,

la figura 15 ilustra un sistema de memoria de ejemplo que tiene seis dispositivos de memoria conectados en serie.

10 Descripción detallada

Las formas de realización de la invención proporcionan un método y un sistema para la difusión general selectiva de órdenes a un subconjunto de una pluralidad de dispositivos en un sistema, donde una orden puede incluir cualquier combinación de un código de operación, dirección de dispositivo, datos de escritura y una dirección de una matriz de memoria. El sistema de memoria incluye dispositivos de memoria y un controlador de memorias conectados en serie en una disposición de topología en anillo. Las formas de realización de la invención están destinadas a seleccionar múltiples dispositivos con el fin de ejecutar una orden sin hacer que todos los dispositivos del sistema de memoria ejecuten la misma orden.

Un sistema que resuelve muchos problemas de rendimiento del sistema 10 de la técnica anterior de la figura 1 es un sistema de dispositivos de memoria conectados en serie, conectados con el controlador de memoria en una configuración de topología en anillo. Las figuras 2A y 2B son diagramas de bloques que ilustran la naturaleza conceptual de un sistema de acuerdo con las formas de realización descritas en la presente. La figura 2A es un diagrama de bloques de un sistema que recibe una señal de reloj en paralelo mientras que la figura 2B es un diagrama de bloques del mismo sistema de la figura 2A que recibe una señal de reloj síncrono de origen. La señal de reloj puede ser o bien una señal de reloj con un terminal a tierra o un par diferencial de relojes.

En la figura 2A, el sistema 20 incluye un controlador de memorias 22 que presenta por lo menos un puerto de salida Ssalida y un puerto de entrada Sentrada, y los dispositivos de memoria 24, 26, 28 y 30 que están conectados en serie. Aunque no se muestra en la figura 2A, cada dispositivo de memoria tiene un puerto de entrada Sentrada y un puerto de salida Ssalida. Los puertos de entrada y de salida constan de uno o más pines o conexiones físicas que interconectan el dispositivo de memoria al sistema del cual forma parte el mismo. En una forma de realización, los dispositivos de memoria pueden ser dispositivos de memoria flash. Alternativamente, los dispositivos de memoria pueden ser DRAM, SRAM o cualquier otro tipo de dispositivo de memoria siempre que disponga de una interfaz de entrada/salida compatible con una estructura de órdenes específica, para ejecutar órdenes o para trasladar órdenes y datos al siguiente dispositivo de memoria. El presente ejemplo de la figura 2A incluye cuatro dispositivos de memoria, aunque unas formas de realización alternativas pueden incluir un único dispositivo de memoria, o cualquier número adecuado de dispositivos de memoria. Por consiguiente, si el dispositivo de memoria 24 es el primer dispositivo del sistema 20 en la medida en la que está conectado a Ssalida, entonces el dispositivo de memoria 30 es el dispositivo N-ésimo o último ya que está conectado a Sentrada, donde N es un número entero mayor que cero. Los dispositivos de memoria 26 a 28 son entonces dispositivos de memoria conectados en serie, intermedios, entre el primer y el último dispositivos de memoria.

Cada dispositivo de memoria puede adoptar un número de identificación (ID), o dirección de dispositivo (DA), diferenciado, al producirse la inicialización de puesta en marcha del sistema, de manera que los mismos son direccionables de forma individual. La solicitud de patente US nº de serie 11/622.828 titulada "APPARATUS AND METHOD FOR PRODUCING IDS FOR INTERCONNECTED DEVICES OF MIXED TYPE", la solicitud de patente US nº de serie 11/750.649 titulada "APPARATUS AND METHOD FOR ESTABLISHING DEVICE IDENTIFIERS FOR SERIALY INTERCONNECTED DEVICES", la solicitud de patente US nº de serie 11/692.452 titulada "APPARATUS AND METHOD FOR PRODUCING DEVICE IDENTIFIERS FOR SERIALY INTERCONNECTED DEVICES OF MIXED TYPE", la solicitud de patente US nº de serie 11/692.446 titulada "APPARATUS AND METHOD FOR PRODUCING IDENTIFIERS REGARDLESS OF MIXED DEVICE TYPE IN A SERIAL INTERCONNECTION", la solicitud de patente US nº de serie 11/692.326 titulada "APPARATUS AND METHOD FOR IDENTIFYING DEVICE TYPE OF SERIALY INTERCONNECTED DEVICES", la solicitud de patente US nº de serie 11/771.023 titulada "ADDRESS ASSIGNMENT AND TYPE RECOGNITION OF SERIALY INTERCONNECTED MEMORY DEVICES OF MIXED TYPE" y la solicitud de patente US nº de serie 11/771.241 titulada "SYSTEM AND METHOD OF OPERATING MEMORY DEVICES OF MIXED TYPE", de propiedad conjunta, describen métodos para generar y asignar direcciones de dispositivo para dispositivos de memoria de un sistema conectados en serie, incorporándose el contenido de dichos documentos a título de referencia en su totalidad.

Los dispositivos de memoria 24 a 30 se consideran como dispositivos conectados en serie ya que la entrada de datos de un dispositivo de memoria está conectada a la salida de datos de un dispositivo de memoria previo, formando así una organización del sistema de conexión en serie, con la excepción del primer y el último dispositivos de memoria de la cadena. El canal del controlador de memorias 22 incluye información de datos, de direcciones, y de control proporcionada por pines independientes, o por los mismos pines, conectados a líneas conductoras. La forma de realización de la figura 2A incluye un canal, donde el canal incluye puertos de Ssalida y Sentrada

correspondiente. No obstante, el controlador de memorias 22 puede incluir cualquier número adecuado de canales para dar acomodo a cadenas de dispositivos de memoria independientes. En el ejemplo de la figura 2A, el controlador de memorias 22 proporciona una señal de reloj CK, la cual está conectada en paralelo a todos los dispositivos de memoria.

En el funcionamiento general, el controlador de memorias 22 emite una orden a través de su puerto de Salida, que incluye un código de operación (*op code*), una dirección de dispositivo, información de direcciones opcional para leer o programar, y datos para programar. La orden se puede emitir como un paquete de orden de flujo continuo de bits en serie, donde el paquete se puede subdividir lógicamente en segmentos de un tamaño predeterminado. Cada segmento puede tener un tamaño de, por ejemplo, un byte. Un flujo continuo de bits es una secuencia o serie de bits proporcionados en el transcurso del tiempo. La orden es recibida por el primer dispositivo de memoria 24, el cual compara la dirección de dispositivo con su dirección asignada. Si las direcciones coinciden, entonces el dispositivo de memoria 24 ejecuta la orden. La orden se hace pasar a través de su propio puerto de salida Salida al siguiente dispositivo de memoria 26, donde se repite el mismo procedimiento. Finalmente, el dispositivo de memoria que tiene la dirección de dispositivo coincidente, al que se hace referencia como dispositivo de memoria seleccionado, llevará a cabo la operación especificada por la orden. Si la orden es una orden de lectura de datos, el dispositivo de memoria seleccionado dará salida a los datos leídos a través de su puerto de salida Salida (no representado), los cuales se hacen pasar en serie a través de dispositivos de memoria intermedios hasta que alcanzan el puerto Entrada del controlador de memorias 22. Puesto que las órdenes y datos se proporcionan en un flujo continuo de bits en serie, el reloj es usado por cada dispositivo de memoria para, mediante impulsos de reloj, introducir/dar salida a los bits en serie y para sincronizar operaciones de dispositivos de memoria internas. Este reloj es usado por todos los dispositivos de memoria del sistema 20.

Puesto que la frecuencia de reloj usada en el sistema según la figura 2A es relativamente baja, se pueden usar niveles de señalización CMOS de excursión completa, al aire, para proporcionar una comunicación robusta de datos. A esto se le hace referencia también como señalización LVTTTL, tal como sabrán bien aquellos expertos en la materia.

Se puede obtener una mejora adicional del rendimiento con respecto al sistema 20 de la figura 2A, por medio del sistema de la figura 2B. El sistema 40 de la figura 2B es similar al sistema 20 de la figura 2A, excepto que la señal de reloj CK se proporciona en serie a cada dispositivo de memoria desde un controlador de memorias 42 alternativo que proporciona la señal de reloj síncrono de origen CK. Cada dispositivo de memoria 44, 46, 48 y 50 puede recibir el reloj síncrono de origen en su puerto de entrada de reloj y reenviarlo, por medio de su puerto de salida de reloj, al siguiente dispositivo en el sistema. En algunos ejemplos del sistema 40, la señal de reloj CK se traslada desde un dispositivo de memoria a otro por medio de líneas de señal cortas. Por lo tanto, no hay presente ninguno de los problemas de rendimiento del reloj relacionados con el esquema de distribución del reloj en paralelo, y CK puede trabajar a altas frecuencias. Por consiguiente, el sistema 40 puede funcionar con una velocidad mayor que el sistema 20 de la figura 2A. Por ejemplo, se puede usar una señalización de lógica de transceptores de alta velocidad (HSTL) para proporcionar una comunicación de datos de alto rendimiento. En el formato de señalización HSTL, cada dispositivo de memoria puede recibir un voltaje de referencia que se usa para determinar un estado lógico de las señales de datos entrantes. Otro formato de señalización similar es el formato de señalización SSTL. Por consiguiente, los circuitos de entrada de datos y de reloj en los dispositivos de memoria de los sistemas 20 y 40 están estructurados de forma diferente entre sí. Los formatos de señalización tanto HSTL como SSTL deberían ser bien conocidos por los expertos en la materia.

Tal como se ha mencionado previamente, los dispositivos de memoria de las figuras 2A y 2B tienen un número de ID (Identificación) o DA (Dirección de Dispositivo) exclusivo de manera que el controlador de memorias puede seleccionar uno de ellos cuando se envía la orden, para que solamente el dispositivo de memoria seleccionado ejecute la orden. Por consiguiente, este esquema de selección de dispositivos de memoria permite la selección de solamente un dispositivo al mismo tiempo cada vez que se emite una orden. La orden contiene un único número de ID o DA, lo cual hace que la selección de múltiples dispositivos resulte imposible con este esquema de selección de dispositivos de memoria. Con fines ilustrativos, las operaciones normales en los sistemas de memoria de las figuras 2A y 2B conllevan la emisión de órdenes secuencialmente cuando dos operaciones diferentes van a ser ejecutadas por dos dispositivos de memoria diferentes. Cada vez que se emite una orden nueva, el paquete de datos puede incluir un nuevo número de ID, y se cancela por defecto la selección de un dispositivo seleccionado previamente. No obstante, si dos dispositivos de memoria diferentes van a ejecutar una orden idéntica, se emite el mismo paquete de datos dos veces, pero con diferentes números de ID de dispositivo. Así, el tiempo de funcionamiento total del sistema de memoria se incrementa ya que se emite la misma orden a más dispositivos de memoria.

Una solución parcial para este problema es la técnica de difusión general, mediante la cual todos los dispositivos de memoria conectados en serie se seleccionan al mismo tiempo y se ejecuta una orden emitida en todos los dispositivos. Tal como se describe en la solicitud de patente US nº de serie 12/033.577, de propiedad conjunta, presentada el 19 de febrero de 2008, la difusión general es una técnica para simplificar la distribución de órdenes comunes a todos los dispositivos de memoria en un sistema. Sin la difusión general, se emiten órdenes independientes a cada dispositivo de memoria por dirección de dispositivo. En la técnica de difusión general, una dirección de difusión general sustituye la dirección (DA) o número de ID del dispositivo en el paquete de datos. Cada

dispositivo de memoria tiene un registro para almacenar la dirección de difusión general que se fija previamente durante la fabricación. Por tanto, cuando se recibe el paquete de datos, cada dispositivo de memoria compara la dirección que está en el paquete de datos tanto con su dirección de dispositivo exclusiva como con su dirección de difusión general almacenada. El dispositivo de memoria ejecuta la orden cuando su dirección almacenada de difusión general coincide con la dirección de difusión general del paquete de datos. No obstante, este método no proporciona una selección múltiple de dispositivos de manera parcial, es decir, dos dispositivos de memoria de un sistema de memoria que tenga un total de cuatro dispositivos de memoria. La selección simultánea de un subconjunto de los dispositivos de memoria en el sistema de memoria se puede proporcionar para obtener características especiales para aplicaciones específicas.

Por ejemplo, el procesado de gráficos puede requerir que datos gráficos se escriban de manera repetida en múltiples dispositivos de memoria. Para aplicaciones con redundancia de datos, se escriben datos sensibles en múltiples dispositivos de memoria con el fin de hacer una copia de seguridad de los datos de escritura, protegiendo así los datos en caso de fallo que se produzca en uno de los dispositivos de memoria. Según las presentes formas de realización, se proporciona un sistema de memoria que permite la selección múltiple de dispositivos con un paquete de datos con el fin de reducir el tiempo de procesado total con respecto a la selección individual de un dispositivo de memoria. Mediante la selección de múltiples dispositivos de memoria con un paquete de datos, se minimiza la energía necesaria para procesar una orden en más de un dispositivo.

Los datos enviados a dispositivos conectados en serie se encapsulan en un paquete de datos, al que se hace referencia también como paquete de orden. Cada paquete incluye por lo menos un encabezamiento y un campo de datos. El campo de datos incluye órdenes a ejecutar, y datos a almacenar en los dispositivos, etcétera. El encabezamiento incluye información de control tal como las direcciones del dispositivo designado. En la figura 3 se ilustra un ejemplo típico de la configuración de bytes de dispositivos conectados en serie.

El paquete de orden 300 presenta la estructura ilustrada en la figura 3, e incluye tres campos, dos de los cuales son opcionales en función de la orden específica que emita el controlador de memorias. El primer campo, que es un campo obligatorio, es el campo de orden 302. El primer campo opcional es un campo de dirección 304, y el segundo campo opcional es un campo de datos 306.

El campo de orden 302 incluye dos subcampos, siendo el primero un campo de dirección de dispositivo (DA) 308 y siendo el segundo un campo de código de operación 310. El campo de dirección de dispositivo 308 se puede considerar como un encabezamiento del paquete de orden 300, y puede tener una longitud de cualquier número adecuado de bits, y se usa para direccionar cada dispositivo de memoria en el sistema. Por ejemplo, un campo de dirección de dispositivo 308 de 1 byte (8 bites) de longitud es suficiente para direccionar hasta 156 (2^8) dispositivos de memoria, aunque el campo de dirección de dispositivo 308 puede tener cualquier longitud que se desee. Se puede reservar una dirección para direccionar todos los dispositivos de memoria simultáneamente, es decir, para difundir de manera general una operación. En una forma de realización alternativa, el campo de dirección de dispositivo 308 puede incluir un campo de tipo de dispositivo para indicar el tipo de dispositivo de memoria al que va dirigido el campo de código de operación 310. Por ejemplo, el campo de tipo de dispositivo puede designar una DRAM, una SRAM o memorias flash. El campo de código de operación 310 puede tener una longitud de cualquier número adecuado de bits con el fin de representar las órdenes para cualquier número adecuado de dispositivos de memoria, y puede incluir una dirección de banco. Por ejemplo, el conjunto de órdenes de memoria flash tendrá órdenes diferentes a las de un conjunto de órdenes de DRAM teniendo en cuenta las diferencias inherentes, con lo que el campo de código de operación se asigna a funciones que den acomodo a todas las órdenes posibles de los dos conjuntos de órdenes en caso de que el sistema incluya ambos tipos de dispositivos de memoria.

No obstante, una alternativa preferida para tratar con tipos diferentes de memorias es hacer que el espacio de códigos de operación esté dividido en dos colecciones de órdenes: aquellas órdenes relacionadas con operaciones en anillo y aquellas relacionadas con operaciones de matrices de memoria. El primer conjunto puede ser uniforme para todos los tipos de memoria, aunque el segundo grupo no es necesario que lo sea. La semántica de los códigos de operación en este grupo sería específica del tipo de memoria. Por ejemplo, un código de operación particular que significase programar cuando se emitiese a una memoria flash podría significar actualizar cuando se presentase a una DRAM. Puesto que el controlador de memorias debe saber la clase de memoria que se encuentra en cada dirección de dispositivo asignada, no importa si estas funciones comparten el mismo código de operación. El campo de dirección 304 se usa para proporcionar una de entre una dirección de fila (Dir Fila), una dirección de columna (Dir Col) o una dirección completa de una matriz de memoria, en función del tipo de operación especificado por el código de operación. El campo de datos 306 incluirá cualquier número adecuado de bits de datos a escribir o programar en el dispositivo de memoria. Por lo tanto, el tamaño de los paquetes de órdenes 300 variará ya que puede que no se requieran datos de escritura para una operación particular y puede que no se requieran ni direcciones ni datos de escritura para una operación particular.

La figura 4 ilustra un sistema 400 de acuerdo con una forma de realización de la presente invención. El sistema 400 incluye un controlador de memorias 402 y una pluralidad de dispositivos conectados en serie 404, 406, 408 y 410. Tal como se ha descrito anteriormente, el controlador de memorias 402 emite una orden a través de su puerto Ssalida, que incluye un código de operación (*op code*), una dirección de dispositivo, información de direcciones

opcional para leer o programar y datos para la programación. En general, el controlador de memorias 402 calcula información de código usada para seleccionar solamente un subconjunto de los dispositivos de memoria en el sistema 400. Esta información de código se incluye en el paquete de orden que se emite para todos los dispositivos de memoria. A continuación, cada dispositivo de memoria decodifica la información de código con el fin de

5 determinar si ha sido seleccionado para responder a los otros campos del paquete de orden recibido. Por ejemplo, según la presente forma de realización, es posible seleccionar simultáneamente los dispositivos 404 y 408 para ejecutar una cierta orden, y excluir los dispositivos 406 y 410 y los dispositivos restantes (no representados) para que no la ejecuten.

10 La figura 5 es un diagrama de flujo que ilustra cómo el sistema de memoria de la presente forma de realización selecciona más de un dispositivo de memoria usando un único paquete de orden. En la etapa 502, el controlador de memorias determina un número de dispositivos de memoria para ejecutar una cierta orden. En la etapa 504, el controlador de memorias genera información de código basándose en las direcciones de ID de los dispositivos de memoria seleccionados. En la etapa 506, el controlador de memorias emite un paquete de orden que incluye la

15 orden a ejecutar y la información de código que identifica los dispositivos seleccionados. En la etapa 508, cada dispositivo de memoria recibe el paquete de orden, y decodifica la información de código para determinar si ha sido seleccionado para responder a los otros campos del paquete de orden recibido.

20 La figura 6 es un diagrama de bloques de un sistema de memorias según una forma de realización de la presente invención. Mientras que la figura 4 es un diagrama de bloques genérico que ilustra el concepto general de la selección de múltiples dispositivos, esta forma de realización muestra detalles de las señales que se propagan entre los dispositivos de memoria y el controlador de memorias. El sistema de memorias 600 incluye un controlador de memorias 602 y cuatro dispositivos de memoria conectados en serie 604, 606, 608 y 610 conectados entre sí en una configuración de topología en anillo. El controlador de memorias 602 emite paquetes de órdenes con la información

25 de código, y cada uno de los dispositivos de memoria puede decodificar la información de código y determinar si debería responder al paquete de orden. El controlador de memorias 602 proporciona señales de control en paralelo a los dispositivos de memoria. Estas incluyen la señal de habilitación chip CE# y la señal de reinicialización RST#. En un uso ejemplificativo de CE#, los dispositivos se habilitan cuando CE# se encuentra en el nivel lógico bajo. En dispositivos de la técnica anterior, una vez que un dispositivo de memoria flash inicia una operación de programación o borrado, CE# se puede poner en estado inactivo, o conducir a un nivel lógico alto. No obstante, en la presente forma de realización, la puesta en estado inactivo de CE# tiene el efecto de deshabilitar la comunicación desde Sentrada a Ssalida del dispositivo de memoria conectado en serie, deshabilitado. Puesto que los dispositivos de memoria están conectados en un anillo, la deshabilitación de cualquiera de los dispositivos interrumpe la comunicación por el anillo y el controlador de memorias se vuelve incapaz de comunicarse con la totalidad de los

30 dispositivos de memoria en el sistema de memoria. Como consecuencia, CE# es una señal común para todos los dispositivos de memoria conectados en serie, y se usa para situar la memoria completa en un estado de baja potencia. En un uso ejemplificativo de RST#, el dispositivo de memoria se fija en un modo de reinicialización cuando RST# se encuentra en el nivel lógico bajo. En el modo de reinicialización, se permite que la potencia se establezca y el dispositivo se prepara para su funcionamiento inicializando todas las máquinas de estados finitos y reinicializando todos los registros de configuración y estado a sus estados por defecto. El controlador de memorias 602 incluye puertos de salida de reloj CKO# y CKO para proporcionar señales de reloj complementarias CK y CK#, y puertos de entrada de reloj CKI# y CKI para recibir las señales de reloj complementarias desde el último dispositivo de memoria del sistema. Cada dispositivo de memoria puede incluir un sintetizador de reloj, tal como un DLL o un PLL para generar fases de los relojes recibidos. Ciertas fases se usan para centrar los flancos del reloj dentro de la ventana

35 válida de datos de entrada internamente con el fin de garantizar un funcionamiento fiable. Cada dispositivo de memoria tiene puertos de salida de reloj CKO# y CKO para trasladar las señales de reloj complementarias a los puertos de entrada de reloj del siguiente dispositivo de memoria, y puertos de entrada de reloj CKI y CKI# para recibir las señales de reloj complementarias o bien del controlador de memorias 602 o bien de un dispositivo de memoria previo. El último dispositivo de memoria 610 proporciona las señales de reloj de vuelta al controlador de memorias 602.

40 El canal del controlador de memorias 602 incluye un puerto de salida de datos Ssalida, un puerto de entrada de datos Sentrada, una entrada estroboscópica de órdenes CSI, una salida estroboscópica de órdenes CSO (eco de CSI), una entrada estroboscópica de datos DSI, y una salida estroboscópica de datos DSO (eco de DSI). El puerto de salida Ssalida y el puerto de entrada Sentrada pueden tener una anchura de un bit, o una anchura de n bits donde n es un entero positivo, en función de las características del controlador de memorias. Por ejemplo, si n es 1 entonces se recibe un byte de datos después de ocho flancos de retención de datos del reloj. Un flanco de reloj de retención de datos puede ser un flanco de reloj de subida, por ejemplo, en una operación de velocidad simple de datos (SDR), o flancos tanto de subida como de bajada del reloj, por ejemplo, en una operación de velocidad doble de datos (DDR). Si n es 2, entonces se recibe un byte de datos después de cuatro flancos de retención del reloj. Si n es 4, entonces se recibe un byte de datos después de dos flancos de retención del reloj. El dispositivo de memoria se puede configurar estáticamente o se puede configurar dinámicamente para cualquier anchura de Ssalida y Sentrada. Por tanto, en una configuración en la que n es mayor que 1, el controlador de memorias proporciona datos en flujos continuos de bits paralelos. CSI se usa para controlar o habilitar los datos de órdenes de retención que

45 aparecen en el puerto de entrada Sentrada, y tiene una duración de impulso para delimitar el tiempo en el que una orden está presente en el puerto de entrada de datos Sentrada. Más específicamente, los datos de órdenes tendrán

una duración medida por un número de ciclos de reloj, y la duración de impulso de la señal de CSI tendrá una duración correspondiente. DSI se usa para habilitar la memoria intermedia del puerto de salida Ssalida de un dispositivo de memoria seleccionado con el fin de dar salida a datos de lectura, y tiene una duración de impulso para delimitar datos de lectura proporcionados desde su puerto de salida de datos Ssalida, de manera que el controlador de memorias sabe cuándo retener datos a su vuelta de su último dispositivo de memoria.

Puesto que la forma de realización descrita actualmente, de la figura 6, está destinada a un funcionamiento de alta velocidad, se usa un formato de señalización de alta velocidad, tal como, por ejemplo, el formato de señalización HSTL. Por consiguiente, se proporciona un voltaje de referencia VREF a cada dispositivo de memoria, el cual es usado por cada dispositivo de memoria para determinar el nivel lógico de las señales recibidas en los puertos de entrada Sentrada, CSI y DSI. El voltaje de referencia VREF puede ser generado por otro circuito en la placa de circuito impreso, por ejemplo, y se fija a un nivel de voltaje predeterminado sobre la base del punto central de excursión del voltaje de la señal de HSTL.

Durante el uso de la forma de realización de la figura 6, cada dispositivo de memoria está posicionado en una placa de circuito impreso, de tal modo que la distancia y la longitud de las pistas de señal entre los pines del puerto de salida Ssalida de un dispositivo y los pines del puerto de entrada Sentrada del dispositivo siguiente en el anillo se minimizan. Alternativamente, los cuatro dispositivos de memoria se pueden reunir en un módulo de sistema encapsulado (*system in package*) (SIP) el cual minimiza adicionalmente las longitudes de las pistas de señales. El controlador de memorias 602 y los dispositivos de memoria conectados en serie 604 a 610 se conectan para formar una topología de anillo, lo cual significa que el último dispositivo de memoria 610 proporciona sus salidas de vuelta al controlador de memorias 602. Como tal, los expertos en la materia apreciarán que la distancia entre el dispositivo de memoria 610 y el controlador de memorias 602 se reduce al mínimo fácilmente.

En la forma de realización ejemplificativa del sistema de la figura 6, se muestran cuatro dispositivos de memoria conectados en serie en una topología en anillo con el controlador de memorias con un reloj síncrono de origen, aunque se puede incluir cualquier número adecuado de dispositivos de memoria y el reloj se puede proporcionar en paralelo a todos los dispositivos de memoria del sistema.

En la forma de realización de la figura 6, el controlador de memorias 602 es responsable de recibir una orden de anfitrión para ejecutar la misma operación en por lo menos dos dispositivos de memoria diferentes. Por consiguiente, el controlador de memorias 602 incluye circuitería para generar la información de código apropiada, y cada uno de los dispositivos de memoria 604, 606, 608 y 610 incluye circuitería para decodificar la información de código del paquete de orden emitido por el controlador de memorias. La figura 7A es un diagrama de bloques que ilustra los bloques de circuito responsables de generar la información de código usada para seleccionar múltiples dispositivos de memoria en el sistema de memoria de la figura 6.

La figura 7A ilustra un ejemplo del controlador de memorias 602 según una forma de realización de la invención. En el ejemplo mostrado en este momento, para simplificar el esquema no se muestran otros bloques de circuito responsables de proporcionar y recibir las otras señales mostradas en la figura 6. El controlador de memorias 602 del presente ejemplo incluye un identificador de dispositivos 700, un codificador de máscara 800, y un encapsulador de datos 900. El identificador de dispositivos 700 recibe la orden enviada por el anfitrión, a la que se hace referencia como orden de anfitrión, la cual puede incluir información que indica que múltiples dispositivos de memoria van a ejecutar la misma operación. En una situación de este tipo, el identificador de dispositivos 700 identifica los dispositivos necesarios para ejecutar la operación, lo cual podría depender de otros algoritmos preestablecidos para controlar, por ejemplo, la nivelación del desgaste. Los expertos en la materia apreciarán que la selección de dispositivos de memoria específicos se puede efectuar basándose en cualesquiera criterios preestablecidos. Una vez identificados, el identificador de dispositivos 700 designa estos dispositivos y proporciona su número de ID o direcciones preasignados al codificador de máscaras 800. Los números de ID pueden ser, por ejemplo, valores binarios, con lo que cada dispositivo de memoria tendrá un número de ID binario exclusivo.

El codificador de máscaras 800 codifica las direcciones o número de ID seleccionados del dispositivo de memoria en información de código, y proporciona esta información de código al encapsulador de datos 900. Tal como se describirá a continuación, el codificador de máscaras 800 lleva a cabo operaciones lógicas sobre todos los números de ID seleccionados con el fin de generar la información de código. El encapsulador de datos 900 genera el paquete de orden, que, según se ha descrito previamente, puede incluir un código de operación, una dirección de dispositivo, información de direcciones opcional para leer o programar en una matriz de memoria del dispositivo de memoria, y datos de escritura para programar. A continuación, el paquete de orden se envía a los dispositivos conectados en serie comenzando con el primer dispositivo de memoria 604 que está conectado directamente al puerto de salida Ssalida del controlador de memorias 602. La figura 7B presenta ejemplos de dónde se almacena la información de código en el paquete de orden.

La figura 7B muestra posibles campos de orden 302 de ejemplo de un paquete de orden 300, donde tal como se ha descrito previamente, el campo de orden 302 incluye un subcampo de dirección de dispositivo 308 y un subcampo de Código de Operación 310. En el presente ejemplo, cada subcampo de dirección de dispositivo 308 incluye una porción de código de máscara 314 y una porción de ID de destino 316, con una anchura conjunta de 1 byte.

Evidentemente, la anchura del subcampo de dirección de dispositivo 308 puede ser mayor que 1 byte para dar acomodo a un número elevado de direcciones. Cabe destacar que el código de máscara 1111 selecciona todos los dispositivos de memoria para que actúen sobre el paquete de orden. El uso de un código de máscara 1111 funciona como una dirección de difusión general, ya que todos los dispositivos de memoria responderían al paquete de orden. Por lo tanto, esta es una alternativa viable al anterior esquema de difusión general en el que, en cada dispositivo de memoria se almacena una dirección reservada, y a continuación la misma se compara con la dirección almacenada en el subcampo de DA 308. Cuando se selecciona solamente un dispositivo, todos los bits del código de máscara se fijan a cero, y la dirección del dispositivo seleccionado se envía al campo de ID de Destino del encabezamiento del paquete. Por tanto, esta es una alternativa viable a sistemas en los que, en el encabezamiento de la orden se incluye solamente una dirección, que designa únicamente un dispositivo de memoria para ejecutar una cierta orden. Para la totalidad del resto de casos, el número de bits que se fija a "1" y la posición de estos bits dependerán del número de dispositivos seleccionados y de las direcciones de estos dispositivos, según se ha descrito anteriormente.

La figura 8 ilustra un ejemplo de un identificador de dispositivos 700 de acuerdo con una forma de realización de la presente invención. El identificador de dispositivos 700 incluye una tabla 702 o registros 702 para almacenar números de ID binarios resultantes de cualquier algoritmo de generación de números de ID, tal como los descritos en las previamente mencionadas solicitudes de patente de propiedad conjunta. Cada número de ID almacenada en la tabla 702 es la dirección de uno de los dispositivos de memoria conectados en serie con el controlador de memorias 602 en el sistema 600. En el presente ejemplo, se supone que el sistema de memoria incluye 16 dispositivos de memoria, con lo que se dispone de 16 números de ID exclusivos que se almacenan en la tabla 702. Si se usa la función de difusión general, entonces una entrada adicional en la tabla 702 almacena la dirección de difusión general BA preseleccionada. Los expertos en la materia entenderán que el identificador de dispositivos 700 incluye registros suficientes para almacenar números de ID de un número máximo predeterminado de dispositivos de memoria. El identificador de dispositivos 700 incluye además un procesador de direcciones 704 que es responsable de decodificar la orden de anfitrión y a continuación de seleccionar uno o más de los dispositivos conectados en serie para ejecutar la operación indicada dentro de la orden de anfitrión. El procesador de direcciones 704 puede incluir uno o más algoritmos preestablecidos para determinar qué dispositivos de memoria se van a seleccionar, si la orden de anfitrión designa una operación en la que se van a seleccionar múltiples dispositivos de memoria. Por ejemplo, un algoritmo puede seleccionar dispositivos de memoria basándose en el espacio disponible para almacenar datos de escritura. En función de la orden de anfitrión, el procesador de dispositivos 704 proporciona los números de ID o direcciones del(de los) dispositivo(s) de memoria seleccionado(s) al codificador de máscaras 800.

El procesador de direcciones 704 distingue entre tres posibles escenarios según determine la orden de anfitrión. En un primer escenario, solamente se designa un dispositivo de memoria. En un segundo escenario, se seleccionan todos los dispositivos de memoria, y se proporciona la dirección de difusión general BA. En el primer y segundo escenarios, el codificador de máscaras 800 se salta, y o bien la dirección de difusión general o bien el número de ID único se traslada directamente al encapsulador de datos 900 por medio de la línea de señal ID_DIRECT. En un tercer escenario, se seleccionan dos o más dispositivos de memoria de entre todos los dispositivos de memoria, y sus números de ID correspondientes se proporcionan al codificador de máscaras 800.

La figura 9 es un esquema de circuito que ilustra, únicamente a título de ejemplo, el codificador de máscaras 800 de la figura 7A de acuerdo con una forma de realización de la presente invención. La figura 9 muestra un codificador diseñado para codificar hasta cuatro números de ID, aunque un experto en la materia apreciará fácilmente que el codificador de máscaras mostrado se puede diseñar y escalar para codificar un código de máscara para dos, ocho, o un número cualquiera de números ID sin apartarse del alcance de la invención. En el presente ejemplo, cada número de ID se representa como una dirección de cuatro bits.

A continuación se expone una visión general del funcionamiento del codificador de máscaras 800 de la presente forma de realización. Cuando el subconjunto seleccionado de direcciones proporcionadas por el identificador de dispositivos 700 se recibe en el codificador de máscaras 800, el codificador de máscaras 800 selecciona uno cualquiera de los números de ID seleccionados, como dirección de ID de Destino. La designación de uno de los números de ID seleccionados para que sea el ID de Destino puede ser arbitraria o se puede basar en cualesquiera criterios predeterminados. La dirección de ID de Destino se usa para codificar un código de máscara, que, junto con la dirección de ID de Destino, se envía a los dispositivos conectados en serie, en el paquete de orden. El código de máscara y la dirección de ID de Destino se incluyen ambos en un campo de datos del paquete de orden, tal como el sub-campo de DA 308 de, por ejemplo, la figura 3, y se les hace referencia en la presente en lo sucesivo como información de código. Cada número de ID se combina lógicamente con el ID de Destino para generar un número de ID codificado que tiene el mismo número de bits que el número de ID original en un primer circuito de codificación. En un segundo circuito de codificación, final, las mismas posiciones de bit de cada número de ID codificado se combinan lógicamente entre sí para generar una posición de bit correspondiente para el código de máscara. En el presente ejemplo, puesto que las direcciones de los números de ID son cuatro bits, se genera un código de máscara de cuatro bits después del circuito de codificación final.

A continuación se describen minuciosamente detalles de la forma de realización del codificador de máscaras de la figura 9. El primer circuito de codificación se corresponde con una primera fase de codificación, que incluye los

codificadores de ID 802, 804, 806 y 808. El segundo circuito de codificación se corresponde con una segunda fase de codificación, que incluye el codificador de máscaras 812. El codificador de ID 802 se designa para la dirección 1, el codificador de ID 804 se designa para la dirección 2, el codificador de ID 806 se designa para la dirección 3, y el codificador de ID 808 se designa para la dirección 4, donde cada dirección se corresponde con un número de ID. Para simplificar el esquema, se muestran únicamente los detalles del circuito correspondientes a los codificadores de ID 802 y 804, puesto que todos los codificadores de ID incluyen los mismos circuitos y se conectan a sus direcciones respectivas y al ID de Destino de la misma manera.

El codificador de ID 802 incluye un conjunto de puertas XNOR 822, 824, 826, y 828, que presentan, cada una de ellas, una entrada para recibir un bit de una posición de bit del ID de Destino y una segunda entrada para recibir un bit de una posición de bit correspondiente de Dirección 1. Tal como se muestra en la figura 9, el bit más significativo, que es el cuarto bit, está conectado a una primera entrada de la puerta XNOR 822. El bit correspondiente del ID de Destino, que es el cuarto bit, está conectado a la segunda entrada de la puerta XNOR 822. Por consiguiente, la salida del comparador 822 es el cuarto bit codificado de la Dirección 1. Las puertas XNOR 824, 826 y 828 reciben el tercer, el segundo y el primer bits de Dirección 1 y el ID de Destino respectivamente, para codificar un tercer, segundo y primer bits codificados de Dirección 1. Las puertas XNOR de los codificadores de ID 804, 806 y 808 están configuradas de la misma manera que la descrita para el codificador de ID 802, excepto que las mismas reciben bits respectivamente de Dirección 2, Dirección 3 y Dirección 4. El uso de puertas XNOR está destinado solamente a ser un ejemplo de una lógica de codificación posible. Un experto en la materia apreciará que se pueden usar otras combinaciones de puertas lógicas para lograr los mismos resultados. Por ejemplo, se puede obtener una lógica XNOR con diferentes combinaciones de puertas lógicas conocidas en la técnica.

El segundo circuito de codificación 812 incluye cuatro puertas lógicas NAND 814, 816, 818, y 820. Cada puerta lógica NAND recibe un bit de la misma posición de bit lógica de cada dirección codificada para generar un bit para la posición de bit correspondiente para el código de máscara. Por ejemplo, la salida de la puerta XNOR 822, que es la posición del cuarto bit del número de ID codificado correspondiente a Dirección 1, está conectada a una primera entrada de la puerta NAND 814. Las entradas segunda a cuarta de la puerta NAND 814 están conectadas a las salidas de cada puerta XNOR que proporciona la posición del cuarto bit de los otros números de ID codificados. Por tanto, la salida de la puerta NAND 814 representa el cuarto bit, más significativo, del código de máscara, mientras que las salidas de las puertas NAND 816, 818 y 820 representan respectivamente el tercer, el segundo y primer bits más significativos. Aunque las puertas NAND se usan en el segundo circuito de codificación de la figura 9, los expertos en la materia apreciarán que se puede usar cualquier tipo de circuitería lógica para generar cada bit del código de máscara como respuesta a las mismas posiciones de bit de cada número de ID codificado.

También es posible que el procesador de direcciones de la figura 7A esté incorporado dentro del codificador de máscaras 800. En este caso, el codificador de máscaras puede detectar las direcciones de los dispositivos seleccionados, a partir de la orden recibida en el controlador de memorias internamente. No obstante, para simplificar, cada función se ha descrito por separado, con el fin de entender mejor la invención que se describe y reivindica en la presente.

La figura 10A es una ilustración de un funcionamiento ejemplificativo del codificador de máscaras 800 de la figura 9. Este ejemplo ilustra la generación de un código de máscara cuando, en el codificador de máscaras, se reciben las siguientes cuatro direcciones de números de ID: 0100, 0101, 0110, 0111. Se supone además que el codificador de máscaras selecciona la dirección 0100 como ID de Destino. Las direcciones de números de ID se muestran en la columna de la matriz situada más a la izquierda, que se muestra en la figura 10A, mientras que el ID de Destino se muestra en la fila superior de la columna más a la derecha de la matriz mostrada en la figura 10A. Siguiendo la lógica mostrada en la figura 9, cada codificador de ID 802, 804, 806 y 808 genera números de ID codificados que se muestran en la columna más a la derecha de la matriz. Por ejemplo, los bits de los números de ID codificados se pueden considerar como una matriz de $M \times N$, donde M es un entero diferente de cero que representa una posición de columna de la matriz y N es un entero diferente de cero que representa una posición de fila de la matriz. Cada bit en la columna se introduce en una puerta NAND, tal como las puertas NAND 814, 816, 818 y 820 de la figura 9, y la salida de cada puerta NAND representa un bit del código de máscara cuya posición en la máscara se corresponde con la posición de la columna en la matriz de la figura 10A. Por ejemplo, la salida de la puerta NAND que recibe los bits de la columna más a la derecha de bits se corresponde con el bit más a la derecha del código de máscara. Dicho de otra manera, la matriz se puede transponer desde una matriz $M \times N$ a una matriz de $N \times M$, y las entradas de cada fila se proporcionan a una puerta NAND respectiva.

De acuerdo con la presente forma de realización, el código de máscara para cualquier conjunto de direcciones seleccionadas es exclusivo de dichas direcciones, y se basa en los valores numéricos de las direcciones seleccionadas con independencia de qué dirección se seleccione como el ID de Destino. Por ejemplo, si, en el ejemplo de la figura 10B, el ID de Destino seleccionado fuera 0110 en lugar de 0100, los números de ID codificados resultantes correspondientes a Dirección 1, Dirección 2, Dirección 3 y Dirección 4 serían los siguientes: 1101, 1100, 1111, y 1110 respectivamente. No obstante, cuando estos resultados se introducen en sus puertas NAND respectivas, el código de máscara obtenido sigue siendo 0011.

Una propiedad útil del código de máscara y las direcciones seleccionadas es que la adición por OR lógica del código

de máscara con cualquiera de las direcciones seleccionadas produce la dirección seleccionada que tiene el mayor valor numérico. La figura 11 muestra una matriz con las direcciones seleccionadas apareciendo en la columna de más a la izquierda, y el código de máscara apareciendo en la fila superior de la columna más a la derecha. La aplicación de la OR lógica del código de máscara con cada dirección seleccionada produce el resultado de 0111 que se corresponde con la Dirección 4. Tal como se describirá posteriormente, esta propiedad se puede usar para identificar el último dispositivo designado para ejecutar la orden, y conmutar la salida de señal de orden Qn a un valor nulo con el fin de ahorrar la energía de la señal que transporta la orden, puesto que los dispositivos entre el último dispositivo designado y el controlador de memorias no están designados. Por ejemplo, un valor nulo podría significar una señal estática que tiene el mismo nivel lógico.

Una vez que se ha generado el código de máscara, el encapsulador de datos 900 proporciona un paquete de orden que incluye el código de operación de la orden, una dirección opcional y datos de escritura opcionales y el código de máscara. Tal como se muestra en la figura 3, el código de máscara y la dirección de ID de Destino se pueden incluir en el sub-campo de DA 308, el código de operación se puede incluir en el sub-campo de Código de Operación 310, cualquier dirección se puede incluir en el campo de Dirección 304 y cualesquiera datos de escritura opcionales se pueden incluir en el campo de Datos 306. A continuación, el encapsulador de datos 900 envía el paquete de orden a todos los dispositivos de memoria en el sistema.

Las formas de realización previamente descritas ilustran cómo se genera un código de máscara de un número fijado de bits para representar todas las direcciones correspondientes a un subconjunto de una pluralidad de dispositivos de memoria en el sistema de memoria. Para que los dispositivos de memoria designados respondan al código de operación del paquete de orden, cada dispositivo de memoria incluye un procesador de ID para decodificar el código de máscara y permitir que un intérprete de órdenes del dispositivo de memoria actúe sobre un código de actuación del paquete de orden, únicamente cuando ese dispositivo de memoria tenga un número de ID de dispositivo que sea miembro del subconjunto de direcciones seleccionadas codificadas en el código de máscara.

La figura 12 es un diagrama de bloques que ilustra una forma de realización de una interfaz de entrada/salida de datos de un dispositivo de memoria diseñado para su interconexión con otro dispositivo de memoria o un controlador de memorias con la finalidad de recibir y proporcionar datos en un flujo continuo de bits en serie, tales como cada uno de los dispositivos de memoria y el controlador de memorias mostrados en la figura 6, por ejemplo. La interfaz de entrada/salida de datos 1200 incluye un trayecto de flujo pasante de datos y un trayecto de memoria.

El trayecto de flujo pasante de datos incluye una memoria intermedia de entrada 1202, un biestable 1204, un selector de datos 1206 y una memoria intermedia de salida 1208, todos ellos conectados en serie entre el puerto de entrada Sentrada y el puerto de salida Ssalida. El trayecto de flujo pasante de datos acopla paquetes de órdenes y cualesquiera datos leídos, recibidos en su puerto de entrada Sentrada desde un dispositivo de memoria previo al puerto de salida Ssalida. Por lo tanto, se pueden proporcionar paquetes de órdenes a todos los dispositivos de memoria del sistema de memorias conectadas en serie. El biestable 1204 vuelve a alinear los bits de datos recibidos en Sentrada con respecto a un reloj, mientras que el selector 1206 puede dar salida selectivamente a datos del biestable 1204 o datos de lectura del dispositivo de memoria como respuesta a la señal LAST_ID. Por consiguiente, el selector 1206 se puede implementar como un multiplexor/demultiplexor.

El trayecto de memoria incluye un procesador de ID 1210, un analizador sintáctico de órdenes 1211, un registro de órdenes 1212, un registro de direcciones 1214, un registro de datos de entrada 1216, un registro de datos de salida 1218, un intérprete de órdenes 1220, y un bloque de circuitos centrales 1222. Puesto que la estructura de los paquetes de órdenes está predefinida, tal como se muestra por ejemplo en la figura 3, el analizador sintáctico de órdenes está configurado para cargar cada uno de los registros indicados con la información correspondiente. Una vez que se ha cargado, el registro de órdenes proporciona un código de operación al intérprete de órdenes 1220, el registro de direcciones 1214 proporciona información de dirección de fila y columna al bloque de circuitos centrales 1222, y el registro de datos de entrada proporciona datos de escritura o programa al bloque de circuitos centrales 1222. El registro de datos de salida almacena datos de lectura de la matriz de memoria del bloque de circuitos centrales 1222. El intérprete de órdenes 1220 es un componente bien conocido que se usa para decodificar el código de operación incluido en el paquete de orden, y proporciona señales de control para su uso por parte del bloque de circuitos centrales 1222 y otros circuitos del dispositivo de memoria. El bloque de circuitos centrales 1222 incluye una matriz de memoria y circuitos centrales usados para controlar la matriz de memoria, tales como decodificadores de filas, de bloques y de columnas, y, por ejemplo, circuitos de trayectos de datos. El procesador de ID 1210 incluye un registro de ID de dispositivo 1224, un registro de máscaras 1226, un registro de ID de destino 1228, un sumador 1230, un decodificador de coincidencias de ID 1232 y un decodificador de últimos ID 1234. Los registros 1226 y 1228 se cargan por medio del analizador sintáctico de órdenes 1211, y proporcionan su información almacenada al sumador 1230 y los decodificadores 1232 y 1234. El ID de dispositivo se puede prealmacenar dinámicamente en el registro 1224 durante la inicialización de puesta en marcha del sistema de memoria, o se puede prealmacenar durante la fabricación del dispositivo de memoria.

La interfaz de entrada/salida de datos 1200 recibe el paquete de orden enviado desde el controlador de memorias o un dispositivo de memoria previo por medio del puerto de entrada Sentrada, y almacena el código de máscara de la información de código en el registro de máscaras 1226, el ID de Destino de la información de código en un registro

de ID Destino 1228, y la orden de código de operación en el registro de órdenes 1212. El decodificador de coincidencias de ID 1232 recibe el ID de dispositivo, el código de máscara y el ID de Destino de los registros 1224, 1226 y 1228 respectivamente, y ejecuta operaciones lógicas sobre ellos. En general, el decodificador de coincidencias ID 1232 compara la información del código de máscara con el número de ID/dirección asignado del dispositivo de memoria. Si se observa que el número de ID asignado está codificado en el código de máscara, la señal de salida ID_MATCH del decodificador de coincidencias de ID 1232 se fija a un nivel lógico activo, tal como, por ejemplo, un 1 lógico, lo cual permite que el intérprete de órdenes 1220 interprete el código de operación almacenado en el registro de órdenes 1212. Si no, el intérprete de órdenes 1220 ignora el código de operación almacenado en el registro de órdenes 1212.

En las formas de realización descritas en este momento, el procesador de ID 1210 puede incluir solamente los registros 1224, 1226, 1228 y el decodificador de coincidencias de ID 1232. Según otra forma de realización de la presente invención, el consumo de potencia se puede reducir inhibiendo o evitando que el último dispositivo de memoria designado envíe el paquete de orden a cualesquiera dispositivos de memoria restantes en el sistema de memoria. Por lo tanto, además de los registros 1224, 1226, 1228 y el decodificador de coincidencias de ID 1232, el procesador de ID 1210 puede incluir además el sumador 1230 y el decodificador de últimos ID 1234, que forman conjuntamente un circuito de inhibición de salida. Tal como se ha descrito previamente el número de ID/dirección de cualquier dispositivo de memoria designado se puede sumar al código de máscara para generar el número de ID/dirección del último dispositivo de memoria designado. En el ejemplo mostrado en este momento, de la figura 12, el sumador 1230 produce la dirección del último dispositivo de memoria designado sumando el código de máscara almacenado en el registro de máscaras 1226 al ID de Destino almacenado en el registro de ID de destino 1228, que es la dirección de uno de los dispositivos designados. A continuación, el decodificador de último ID 1234 compara la dirección del dispositivo de memoria almacenada en el registro de ID de dispositivo 1224 con la dirección del último dispositivo designado, generada por el sumador 1230. Si no se produce ninguna coincidencia (a lo que se hace referencia también como discordancia) entre la dirección del dispositivo de memoria y la dirección generada del último dispositivo designado, la señal de salida LAST_ID se fija a un nivel lógico inactivo, tal como, por ejemplo, un 0 lógico, y el selector 1206 permite el reenvío del paquete de orden al siguiente dispositivo acoplando la salida del biestable 1204 a la memoria intermedia de salida 1208. Si no, en caso de que se produzca una coincidencia, la señal de salida LAST_ID se fija a un nivel lógico activo, tal como, por ejemplo, un 1 lógico, y el selector 1206 acopla la entrada de la memoria intermedia de salida 1208 al registro de datos de salida 1218.

El registro de datos de salida 1218 puede almacenar datos de salida de una operación de lectura previa, o se puede reinicializar. En cualquiera de las situaciones, el registro de datos de salida 1218 proporciona estáticamente un bit de datos al selector 1206, el cual puede ser una señal de 1 o 0 lógico. Por lo tanto, la Ssalida del último dispositivo de memoria designado acciona y mantiene un nivel lógico estático. Por consiguiente, todos los dispositivos de memoria subsiguientes recibirán este nivel lógico estático, y se conserva el consumo de potencia del sistema de memoria puesto que no se produce ninguna conmutación de señal adicional por parte de ninguno de los puertos de salida Ssalida de los dispositivos de memoria restantes. En la presente forma de realización, se evita que el paquete de orden sea enviado a los dispositivos de memoria restantes del sistema acoplando la memoria intermedia de salida 1208 al registro de datos de salida 1218. En formas de realización alternativas, la memoria intermedia de salida 1208 se puede deshabilitar por medio de la señal LAST_ID, o la salida del selector 1206 se puede deshabilitar o desconectar de la memoria intermedia de salida 1208, o la salida del biestable 1204 se puede deshabilitar o desconectar del selector 1206.

La figura 13 es un esquema de circuito que ilustra, a título de ejemplo, el decodificador de coincidencias de ID 1232 y el decodificador de último ID 1234. El decodificador de coincidencias de ID 1232 combina lógicamente bits del código de máscara MC[1:4], el número de ID destino T_ID[1:4] y el número de ID D_ID[1:4] del dispositivo de memoria con el fin de determinar si el presente dispositivo de memoria se ha designado o no. Más específicamente, el decodificador de coincidencias de ID 1232 determina si el número de ID del dispositivo de memoria está codificado en el código de máscara. El decodificador de último ID 1234 combina lógicamente el código de máscara MC[1:4] y el número de ID D_ID[1:4] del dispositivo de memoria con el fin de determinar si el presente dispositivo de memoria es el último dispositivo designado para el paquete de orden correspondiente. Para el presente ejemplo se supone que el código de máscara, el número de ID de destino y el número de ID de dispositivo tienen cuatro bits.

El decodificador de coincidencias de ID 1232 incluye un conjunto de puertas lógicas X-NOR 1302, 1304, 1306 y 1308, un conjunto de puertas lógicas OR 1312, 1314, 1316 y 1318, y una puerta lógica AND 1320. En conjunto, la totalidad de las cuatro puertas lógicas X-NOR forma un comparador usado para comparar cada bit del ID de Destino recibido con la posición de bit correspondiente del número de ID de dispositivo. Por ejemplo, la puerta lógica X-NOR 1302 compara el bit menos significativo del número de ID de destino T_ID[1] con el bit menos significativo del número de ID de dispositivo D_ID[1], y proporciona un resultado de comparación. A la totalidad de los cuatro resultados de comparación se le puede hacer referencia como número de ID de destino codificado. La salida de cada puerta lógica X-NOR está conectada a una entrada de una puerta lógica OR correspondiente. La otra entrada de cada puerta lógica OR recibe una posición de bit correspondiente del código de máscara. En conjunto, la totalidad de las cuatro puertas lógicas OR forma un sumador para sumar el número de ID de destino codificado con el código de máscara con el fin de proporcionar un número de ID de destino, codificado, modificado. La salida resultante de cada puerta lógica OR se proporciona a la puerta lógica AND 1320 que actúa como un multiplicador, el

cual puede conducir la señal ID_MATCH o bien a un 1 o bien a un 0 lógico. En el presente ejemplo, una salida de 1 lógico de cada uno de los sumadores 1312, 1314, 1316 y 1318 indica que el presente dispositivo de memoria está designado para actuar sobre el paquete de orden. Este estado se representa al estar situado ID_MATCH en un 1 lógico. Si por lo menos una puerta lógica OR da salida a un 0 lógico, entonces el presente dispositivo de memoria no está designado, ID_MATCH se fija a 0 lógico, y el código de operación se ignora. Cuando se fija a un 1 lógico, ID_MATCH permite que el intérprete de órdenes 1220 interprete el código de operación almacenado en el registro de órdenes 1212 con el fin de generar las señales de control apropiadas para controlar circuitos en el bloque de circuitos centrales 1222.

El sumador 1230 incluye un conjunto de puertas lógicas OR 1342, 1344, 1346 y 1348, mientras que el decodificador de último ID 1234 incluye un conjunto de puertas lógicas XNOR 1341, 1343, 1345 y 1347, y una puerta lógica AND 1350. El sumador 1230 suma cada bit del código de máscara MC[1:4] a la posición de bit correspondiente del número de ID de destino T_ID[1:4]. Por ejemplo, la puerta lógica OR 1342 recibe los bits más significativos MC[4] y T_ID[4]. Tal como se ha descrito previamente, dicha suma da como resultado la dirección del último dispositivo designado en las salidas de las puertas lógicas OR. De este modo, la salida de cada puerta lógica OR 1342, 1344, 1346 y 1348 representa un bit del número de ID del último dispositivo designado. Los bits del número de ID generado del último dispositivo designado se proporcionan a una entrada de las puertas lógicas XNOR 1341, 1343, 1345 y 1347. La otra entrada de cada puerta lógica XNOR de dispositivo del decodificador de ID 1234 está conectada al bit correspondiente del número de ID de dispositivo. En conjunto, las puertas lógicas XNOR 1341, 1343, 1345 y 1347 funcionan como un comparador para comparar el número de ID de dispositivo con el número de ID generado del último dispositivo designado. En el presente ejemplo, un 1 lógico en la salida de las puertas lógicas XNOR 1341, 1343, 1345 y 1347 indica que el presente dispositivo de memoria es el último dispositivo designado para el paquete de orden correspondiente. Si por lo menos una de las salidas es un 0 lógico, entonces el presente dispositivo de memoria no es el último dispositivo designado. La salida de las puertas lógicas XNOR 1341, 1343, 1345 y 1347 está conectada a la entrada de la puerta lógica AND 1350, la cual funciona como un multiplicador. Cuando la salida LAST_ID de la puerta lógica AND 1350 se encuentra en un 1 lógico, el selector 1206 de la figura 12 desconecta el biestable 1204 de la memoria intermedia de salida 1208, y conecta el registro de salida de datos 1218 a la memoria intermedia de salida 1208 con el fin de ahorrar energía. En caso contrario, el paquete de orden se proporciona al siguiente dispositivo.

A continuación se presenta una operación ejemplificativa del decodificador de coincidencias de ID 1232 y el decodificador de último ID 1234 mostrados en la figura 13, usando los números de ID previamente designados para codificar el código de máscara. Tal como se ha mostrado previamente, cuando los cuatro números de ID designados son 0100, 0101, 0110, 0111, el código de máscara resultante es 0011. Se supone que el paquete de orden incluye el ID de dispositivo de destino de 0100 y el código de máscara de 0011, y este paquete de orden es recibido por un dispositivo de memoria que tiene el número de ID 0101. Por consiguiente, cada dispositivo de memoria recibe el paquete de orden y analiza sintácticamente los campos de tal manera que el registro de códigos de máscara 1226 almacena el código de máscara de cuatro bits mientras que el registro de ID de destino 1228 almacena el número de ID de dispositivo de destino de cuatro bits. La tabla 1 a continuación ilustra el establecimiento de correspondencias de los bits con las posiciones de bit correspondientes del código de máscara, el número de ID de dispositivo de destino y los números de ID de dispositivos designados. La figura 14 es un diagrama de flujo que muestra el funcionamiento del sistema de memoria de las presentes formas de realización como respuesta a un paquete de orden que direcciona selectivamente un subconjunto de los dispositivos de memoria.

Tabla 1

Código de máscara	MC[4]	MC[3]	MC[2]	MC[1]
	0	0	1	1
ID de destino	T_ID[4]	T_ID[3]	T_ID[2]	T_ID[1]
	0	1	0	0
ID de dispositivo 1	D_ID[4]	D_ID[3]	D_ID[2]	D_ID[1]
	0	1	0	0
ID de dispositivo 2	D_ID[4]	D_ID[3]	D_ID[2]	D_ID[1]
	0	1	0	1
ID de dispositivo 3	D_ID[4]	D_ID[3]	D_ID[2]	D_ID[1]
	0	1	1	0
ID de dispositivo 4 [4]	D_ID[4]	D_ID[3]	D_ID[2]	D_ID[1]
	0	1	1	1

En la figura 14, se supone que el anfitrión (no representado) emite una orden, es decir, una orden de lectura o escritura de datos, desde o hacia dispositivos de memoria. La orden es recibida por el controlador de memorias al cual están conectados en serie los dispositivos, en la etapa 1400. El controlador de memorias interpreta la orden y determina que los dispositivos de memoria que tienen números de ID 0100, 0101, 0110 y 0111 van a ejecutar la orden. En la etapa 1402, los cuatro números de ID de dispositivos seleccionados (ID de Dispositivo 1, ID de Dispositivo 2, ID de Dispositivo 3 e ID de Dispositivo 4) se usan para calcular o codificar el código de máscara que

es específico de los dispositivos designados. El código de máscara y el ID de Destino se encapsulan en un paquete de orden junto con el código de operación correspondiente a la orden emitida por el anfitrión, y a continuación se envían a cada uno de los dispositivos de memoria secuencialmente en la etapa 1404. En la etapa 1406, el primer dispositivo de memoria conectado al controlador de memorias decodifica el código de máscara y el ID de destino recibidos en el paquete de orden para determinar si ha sido designado o no, por medio, por ejemplo, de su decodificador de coincidencias de ID 1232. Si este dispositivo de memoria tiene un número de ID diferente a los cuatro enumerados en la Tabla 1, entonces este primer dispositivo de memoria no está designado. Por lo tanto, en la etapa 1408, la señal ID_MATCH de su decodificador de coincidencias de ID 1232 se fijaría a un 0 lógico inactivo. A continuación, el paquete de orden se traslada al siguiente dispositivo de memoria en el sistema en la etapa 1410, y el método realiza un bucle de vuelta a la etapa 1406 la cual es ejecutada por el siguiente dispositivo de memoria.

Si el siguiente dispositivo de memoria tiene asignado el ID de Dispositivo 1, entonces es uno de los dispositivos de memoria designados. A continuación se describe la etapa de decodificación 1406 en referencia a la figura 14 y usando los valores de la Tabla 1. Cuando el ID de dispositivo es 0100, las salidas de las puertas lógicas XNOR 1302, 1304, 1306 y 1308 son 1111 respectivamente. Después de que a este resultado se le aplique una OR con el código de máscara, las salidas de las puertas lógicas OR 1312, 1314, 1316 y 1318 son respectivamente 1111. Por lo tanto, la puerta lógica AND 1320 fija ID_MATCH a un 1 lógico activo. Puesto que ID_MATCH es un 1 lógico activo, se determina, en la etapa 1408, que el presente dispositivo de memoria está designado. Por lo tanto, el código de operación almacenado en el registro de órdenes 1212 es interpretado por el intérprete de órdenes 1220 en la etapa 1412, y es ejecutado. En una forma de realización del presente método, el método finaliza en la etapa 1412 mientras el paquete de orden se traslada al siguiente dispositivo de memoria en la etapa 1410. Por consiguiente, las etapas 1406, 1408 y 1410 son ejecutadas por todos los dispositivos de memoria no designados, mientras que las etapas 1406, 1408, 1410 y 1412 son ejecutadas por dispositivos de memoria que tienen el ID de Dispositivo 2, el ID de Dispositivo 3 y el ID de Dispositivo 4. Más particularmente, la etapa 1410 es ejecutada por el último dispositivo de memoria designado que tiene el ID de Dispositivo 4. No obstante, si se encuentra cualquier dispositivo de memoria subsiguiente conectado en serie entre el último dispositivo de memoria designado y el controlador de memorias, entonces cada uno de estos dispositivos de memoria intermedios recibirá y trasladará el paquete de orden al siguiente dispositivo de memoria hasta que el paquete de orden sea recibido por el controlador de memorias.

Tal como se ha mencionado anteriormente, puesto que el último dispositivo de memoria designado ha ejecutado la orden, no existe necesidad de que los dispositivos de memoria intermedios ni siquiera reciban el paquete de orden ya que nunca fueron designados para ejecutar la orden. Además, la acción de trasladar el paquete de orden desde un dispositivo de memoria no designado a otro dispositivo de memoria no designado consume potencia de transición de señales. En el escenario del peor caso usando el ejemplo previo, los primeros cuatro dispositivos de memoria en el sistema de memoria son los dispositivos de memoria designados, dejando un número máximo de dispositivos de memoria intermedios no designados.

En una forma de realización alternativa del presente método, el consumo de potencia se puede reducir haciendo que cada dispositivo de memoria designado compruebe si es el último dispositivo designado para la misma orden o no. Continuando a partir de la etapa 1412 de la figura 14 y suponiendo que el dispositivo de memoria actual tiene el ID de Dispositivo 3, en la etapa 1414 se efectúa una determinación para comprobar si el dispositivo de memoria presente es el último dispositivo de memoria designado. Esto se efectúa por medio del decodificador de último ID 1234. Por ejemplo, las salidas de las puertas lógicas OR 1342, 1344, 1346 y 1348 del sumador 1230 son 0111, con independencia del número de ID de dispositivo actual. Este resultado es el ID del último dispositivo designado, generado, el cual se compara con el número de ID de dispositivo correspondiente al presente dispositivo de memoria, que es el ID de Dispositivo 3. Entonces, las salidas de las puertas lógicas XNOR son respectivamente 1110. Por consiguiente, la puerta lógica AND 1350 conduce LAST_ID a un 0 lógico inactivo, lo cual significa que el presente dispositivo de memoria no es el último dispositivo de memoria designado. En un escenario de este tipo, el método de la figura 14 prosigue hacia la etapa 1410 y el paquete de orden se traslada al siguiente dispositivo de memoria.

Por otro lado, si el presente dispositivo de memoria tiene el ID de Dispositivo 4, entonces la salida 0111 del sumador 1230 se compara con el ID de Dispositivo 4, dando como resultado una salida de 1111. Por consiguiente, la puerta lógica AND 1350 conduce LAST_ID a un 1 lógico activo, lo cual significa que el presente dispositivo de memoria es el último dispositivo de memoria designado. En un escenario de este tipo, el método de la figura 14 prosigue hacia la etapa 1416. En la etapa 1416, se inhibe, o se evita, que el paquete de orden sea trasladado al siguiente dispositivo de memoria en la medida en la que el puerto de salida Ssalida del dispositivo de memoria se fija a un valor nulo estático. Este valor nulo puede ser un valor estático o bien de 0 lógico o bien de 1 lógico. Tal como se muestra en la figura 12 a título de ejemplo, este valor nulo se puede obtener acoplando la memoria intermedia de salida 1208 al registro de datos de salida 1218.

Basándose en el ejemplo de circuito mostrado en la figura 13, la etapa 1414 se ejecuta en paralelo con la etapa 1408, donde se detecta si el dispositivo de memoria está designado o no por medio del decodificador de coincidencias de ID 1232. Según un ejemplo de circuito alternativo, la etapa 1408 se ejecuta en serie después de la etapa 1408, siempre que el decodificador de coincidencias de ID 1232 determine que el dispositivo de memoria es uno de los que ha sido designado. Esto se puede realizar conectado ID_MATCH a otra entrada de la puerta lógica

AND 1350. Por lo tanto, la puerta lógica AND 1350 se habilita únicamente cuando ID_MATCH se encuentra en el valor de 1 lógico.

5 De este modo, tal como se muestra por medio de las formas de realización descritas anteriormente de la invención, un subconjunto de dispositivos de memoria, a los que se hace referencia como dispositivos de memoria designados o seleccionados, puede responder a un único paquete de orden emitido por el controlador de memorias. Los ejemplos del codificador de máscaras 800, mostrados previamente, para generar la información de código integrada en el paquete de orden se pueden diseñar para recibir exactamente cuatro direcciones de ID de dispositivo designadas, o se pueden escalar para recibir el número máximo de direcciones designadas previstas. No obstante, 10 un codificador de máscaras de este tipo hará que aumente el área del dispositivo de memoria debido a la circuitería lógica adicional. Por lo tanto, si el tamaño del codificador de máscaras 800 debe minimizarse, entonces el mismo se puede limitar para recibir hasta cuatro, ocho o cualquier agrupamiento de 2^n números de ID de dispositivo. No obstante, si existen más números de ID de dispositivo designados a los que puede dar acomodo el codificador de máscaras 800, entonces se requieren por lo menos dos ciclos de ejecución, donde un ciclo incluye la codificación del código de máscara y la emisión del paquete de orden correspondiente a los dispositivos de memoria. En otras palabras, por lo menos dos grupos diferentes de dispositivos de memoria ejecutarán la misma orden, pero en respuesta a por lo menos dos paquetes de orden diferentes.

20 La figura 15 ilustra la selección de seis (6) dispositivos de memoria conectados en serie 1500 en un sistema de memoria, donde el codificador de máscaras está configurado para recibir un máximo de cuatro números de ID de dispositivo designados. Se supone que cada dispositivo de memoria tiene el procesador de ID 1210 previamente descrito. En este ejemplo, el procesador de direcciones 704 de la figura 8 proporciona al codificador de máscaras 800 los números de ID de los dispositivos seleccionados, en grupos de 2^n donde n es 2. Cuando los números de ID de los dispositivos mostrados en la figura 15 se van a presentar al codificador de máscaras 800, el procesador de direcciones 704 proporciona los números de ID de estos dispositivos en dos grupos. En primer lugar se encuentra un grupo de dos direcciones (2^1), y el segundo es un grupo de cuatro direcciones (2^2). El orden se puede invertir sin ningún impacto sobre el funcionamiento del sistema. Se observa que el procesador de direcciones 704 garantiza que los números de ID de los dispositivos seleccionados, proporcionados al codificador de máscaras 800 dentro de un agrupamiento, no incluyan números de ID complementarios. Los emparejamientos ejemplificativos de números de ID 30 complementarios incluyen 0100 y 1011, 1100 y 0011, etcétera. Por lo tanto, en un primer ciclo de ejecución, se genera un primer código de máscara usando el grupo de dos direcciones. A continuación, los dos dispositivos de memoria designados ejecutan el código de operación en el paquete de orden. En un segundo ciclo de ejecución tras el primero, los cuatro dispositivos de memoria designados ejecutan el mismo código de operación en el segundo paquete de orden.

35 En la descripción anterior, con fines explicativos, se exponen numerosos detalles con el fin de proporcionar una comprensión minuciosa de las formas de realización de la invención. No obstante, se pondrá de manifiesto a los expertos en la materia que estos detalles específicos no se requieren para llevar a la práctica la invención. En otros casos, estructuras eléctricas y circuitos bien conocidos se muestran en forma de diagrama de bloques para no complicar la invención. Por ejemplo, no se proporcionan detalles específicos en relación con si las formas de realización de la invención descritas en la presente se implementan como una rutina de software, un circuito de hardware, microprogramas, o una combinación de los mismos.

45 Las formas de realización de la invención descritas anteriormente son proporcionadas únicamente a título de ejemplo. Por ejemplo, la invención no se limita a las puertas mostradas en los dibujos. El experto en la materia apreciará que se pueden usar otros circuitos y esquemas que lleven a cabo las mismas funciones realizadas por las puertas. En el controlador de memorias y los dispositivos conectados en serie también se pueden usar otras configuraciones de codificadores y de decodificadores. Se pueden introducir alteraciones, modificaciones y variaciones en las formas de realización particulares, por parte de los expertos en la materia, sin apartarse del alcance de la invención, que resulta definido únicamente por las reivindicaciones adjuntas a la misma.

50

REIVINDICACIONES

1. Método para seleccionar un subconjunto de dispositivos de una pluralidad de dispositivos conectados en serie a un controlador de memorias con el fin de ejecutar una orden, que comprende:
- 5 codificar números de identificación (ID) de cada uno del subconjunto de dispositivos para proporcionar información de código llevando a cabo operaciones matemáticas con el fin de combinar lógicamente los números de ID del subconjunto de dispositivos para generar una máscara;
- 10 proporcionar un paquete de orden que incluye un código de operación y la información de código a cada uno de la pluralidad de dispositivos;
- decodificar la información de código recibida por cada uno de la pluralidad de dispositivos ejecutando, en cada dispositivo, una operación matemática sobre la máscara y un número de ID almacenado en el dispositivo, para
- 15 determinar si la máscara codifica el número de ID almacenado en el dispositivo; y
- ejecutar el código de operación si el número de ID almacenado en el dispositivo está codificado en la información de código recibida.
- 20 2. Método según la reivindicación 1, en el que el controlador de memorias proporciona 2^x de los números de ID, en el que x es un entero mayor que cero.
3. Método según la reivindicación 1, en el que cada uno de los números de ID presenta una longitud de N bits, y la información de código incluye un número de ID de Destino de N bits de longitud y el código de máscara de N bits de
- 25 longitud, en el que N es un valor entero mayor que 0.
4. Método según la reivindicación 3, en el que la etapa de codificación comprende:
- seleccionar un número de ID del subconjunto de dispositivos como número de ID de Destino;
- 30 comparar los números de ID de cada uno del subconjunto de dispositivos con el número de ID de Destino para generar números de ID codificados correspondientes; y
- combinar lógicamente un bit en la misma posición de bit de cada uno de los números de ID codificados para
- 35 generar un bit que presenta la posición de bit correspondiente en el código de máscara.
5. Método según la reivindicación 4, en el que existen M números de ID correspondientes al subconjunto de dispositivos, en el que M es un valor entero mayor que 0.
- 40 6. Método según la reivindicación 5, en el que la etapa de comparación comprende las etapas de:
- dar salida a los resultados de comparación en una matriz de $M \times N$; y
- transponer la matriz de $M \times N$ a una matriz de $N \times M$.
- 45 7. Método según la reivindicación 6, en el que dicho un bit se genera ejecutando una función lógica sobre los resultados de la comparación en una de las N filas de la matriz de $N \times M$.
8. Método según la reivindicación 7, en el que un primer bit del código de máscara generado a partir de la primera fila de la matriz de $N \times M$ representa un bit más significativo del código de máscara, y un último bit del código de
- 50 máscara generado a partir de la última fila de la matriz de $N \times M$ representa un bit menos significativo del código de máscara.
9. Método según la reivindicación 4, en el que la etapa de comparación comprende una etapa de ejecutar una función XNOR entre los números de ID de cada uno del subconjunto de dispositivos con el número de ID de Destino.
- 55 10. Método según la reivindicación 4, en el que la etapa de combinar lógicamente comprende una etapa de ejecutar una función NAND entre bits en la misma posición de bit de cada uno de los números de ID codificados.
11. Método según la reivindicación 4, en el que la etapa de decodificación comprende una etapa de procesar el número de ID almacenado en el dispositivo con el número de ID de Destino y el código de máscara para proporcionar una indicación de que todas las posiciones de bit del número de ID almacenado en el dispositivo están
- 60 codificadas en el código de máscara.
12. Método según la reivindicación 11, en el que la etapa de procesado comprende
- 65 comparar los bits de cada posición de bit del número de ID almacenado en el dispositivo con los bits

correspondientes de cada posición de bit correspondiente del número de ID de Destino para generar un número de ID de destino codificado;

5 adicionar el código de máscara al número de ID de destino codificado para proporcionar un número de ID de destino codificado modificado; y

10 multiplicar entre sí los bits del número de ID de destino codificado modificado, con el fin de proporcionar la indicación de que todas las posiciones de bit del número de ID almacenado en el dispositivo están codificadas en la máscara.

15 13. Método según la reivindicación 12, en el que la etapa de comparar los bits de cada posición de bit del número de ID almacenado en el dispositivo comprende una etapa de ejecución de una función lógica XNOR; la etapa de adicionar el código de máscara al número de ID de destino codificado comprende una etapa de ejecutar una función lógica OR; y la etapa de multiplicar los bits del número de ID de destino codificado modificado comprende una etapa de ejecutar una función lógica AND.

20 14. Método según la reivindicación 3, en el que la etapa de proporcionar el paquete de orden comprende unas etapas en las que cada uno de la pluralidad de dispositivos recibe el paquete de orden desde uno de entre un dispositivo previo y el controlador de memorias, y traslada el paquete de orden a uno de entre un dispositivo subsiguiente y el controlador de memorias, incluyendo el dispositivo previo y el dispositivo subsiguiente los dispositivos de memoria.

25 15. Método según la reivindicación 14, que comprende además una etapa de inhibición del traslado del paquete de orden a uno de entre el dispositivo subsiguiente y el controlador de memorias cuando el último dispositivo del subconjunto de dispositivos decodifica la información de código.

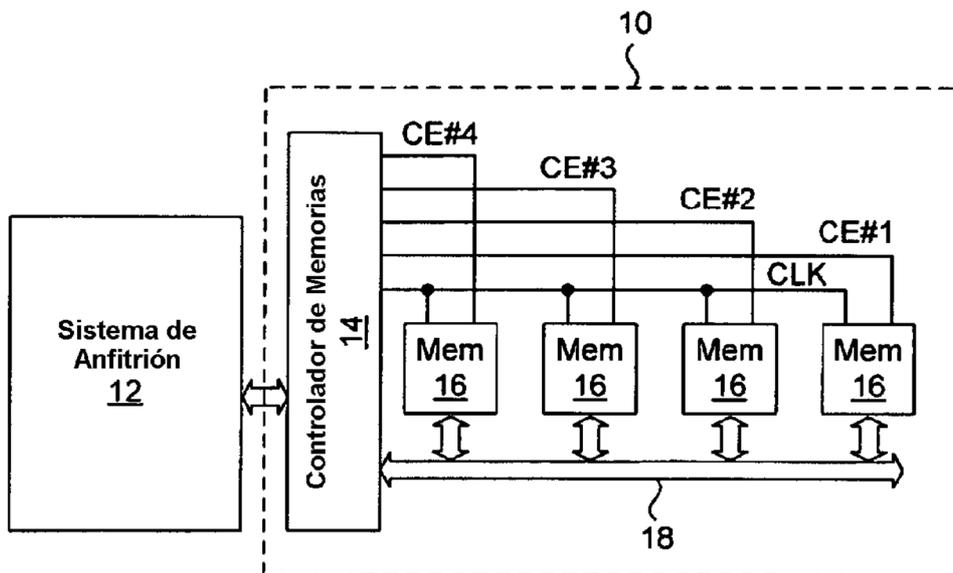


FIG. 1 (Técnica Anterior)

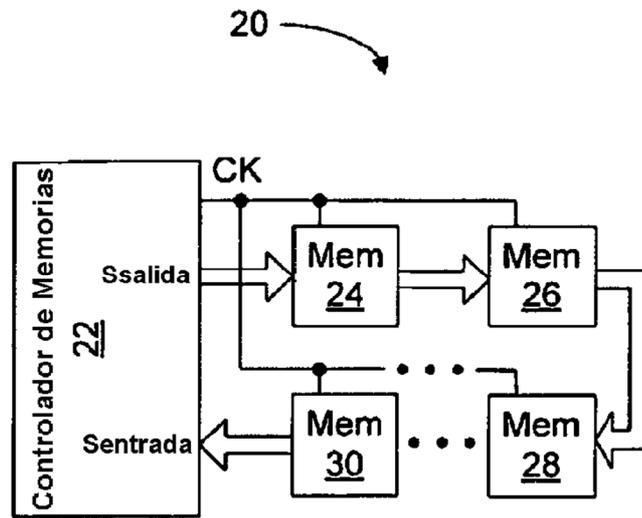


FIG. 2A

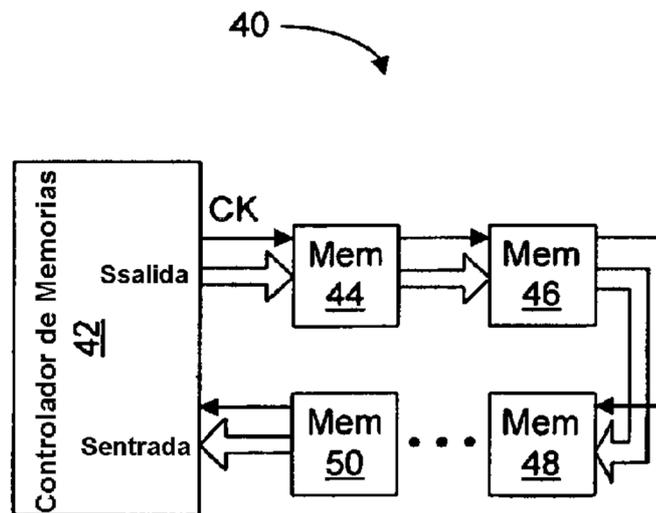


FIG. 2B

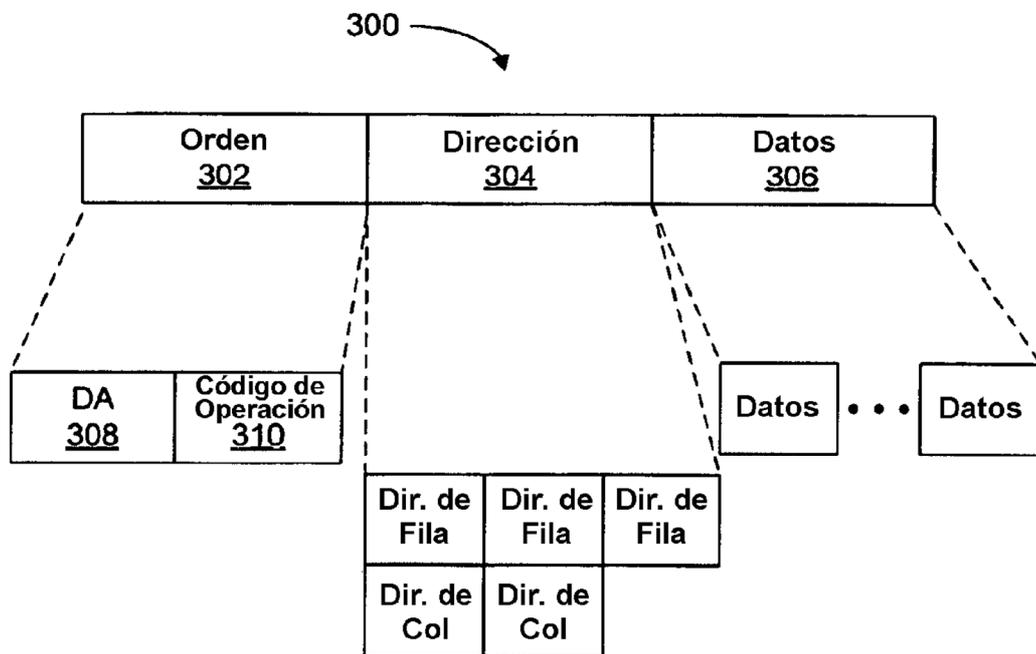


FIG. 3

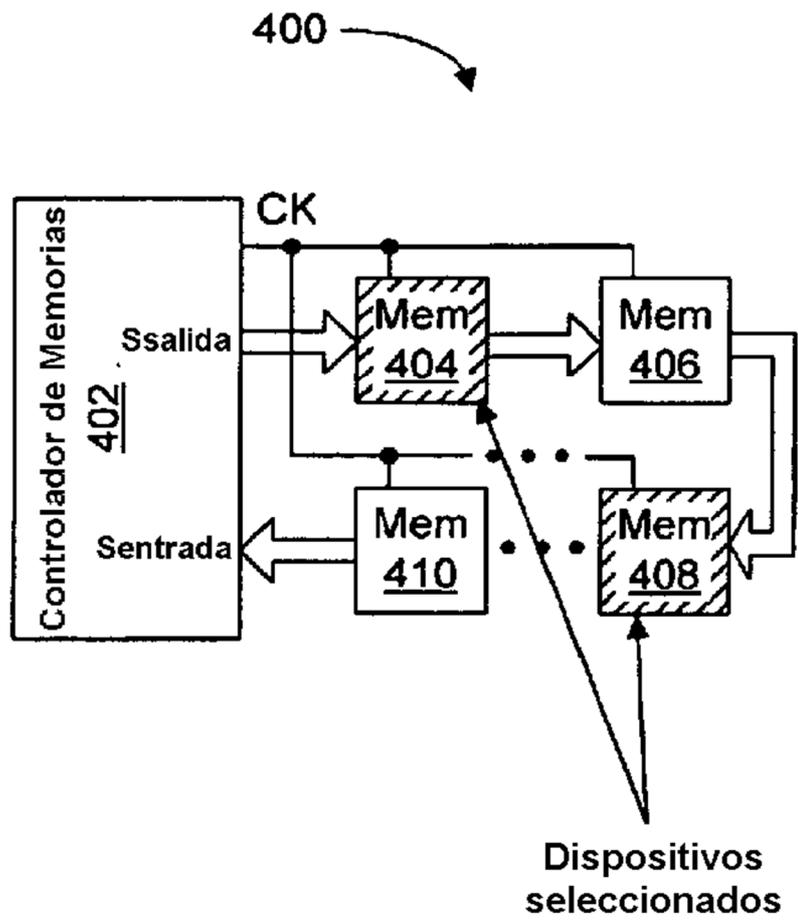


FIG. 4

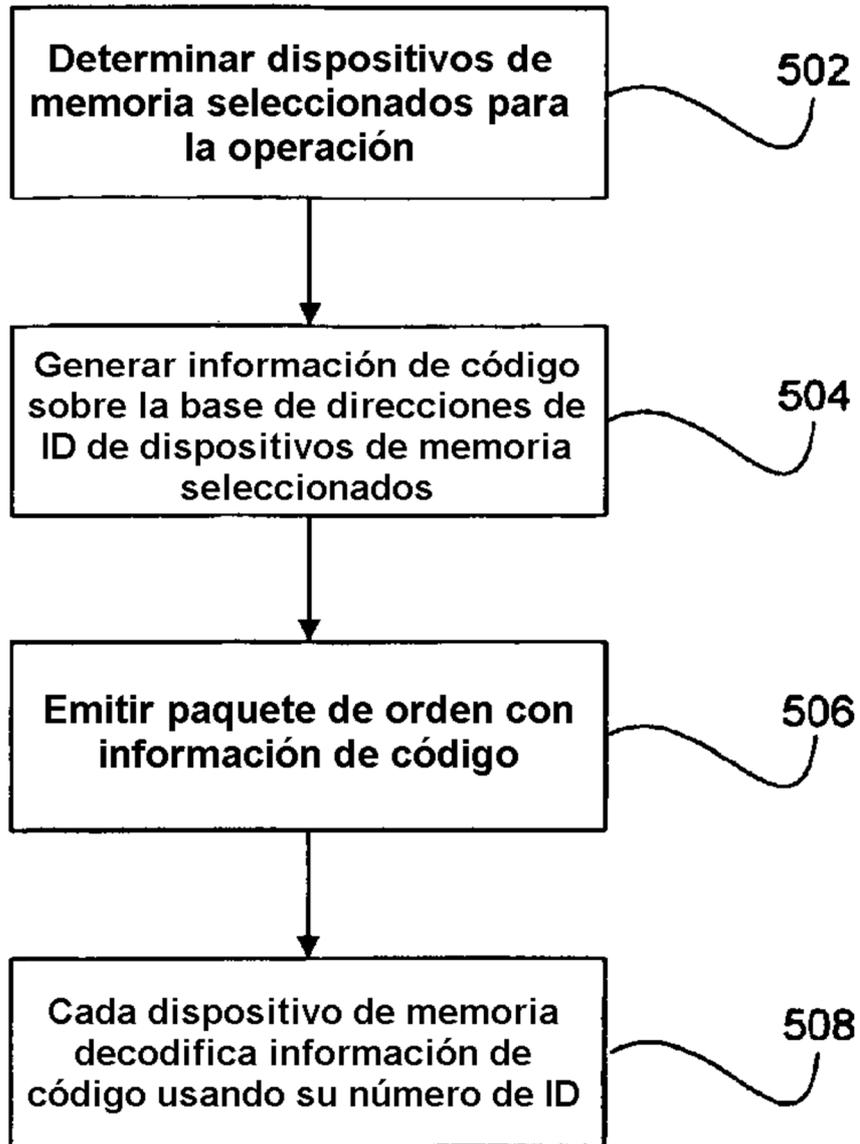


FIG. 5

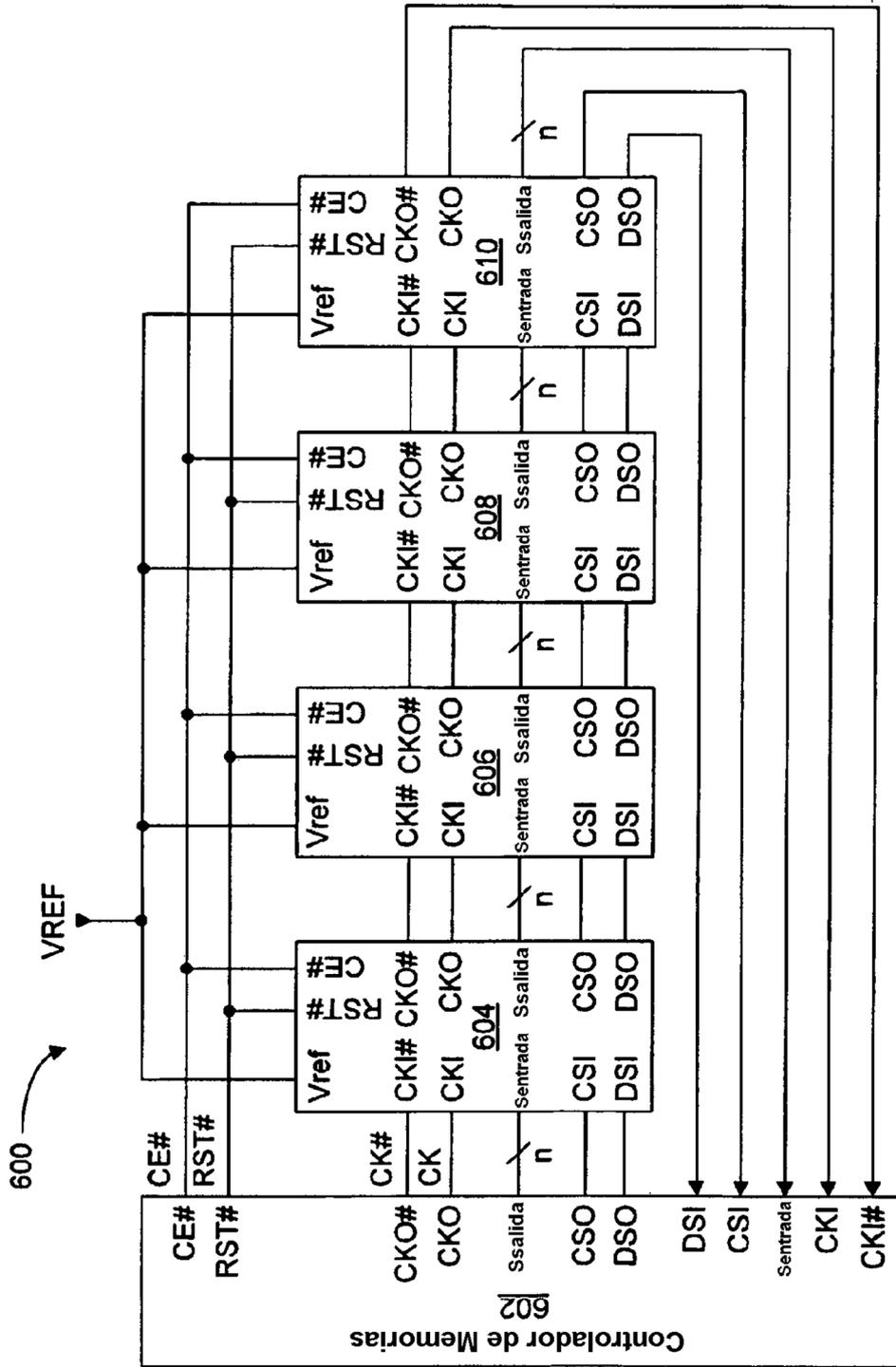


FIG. 6

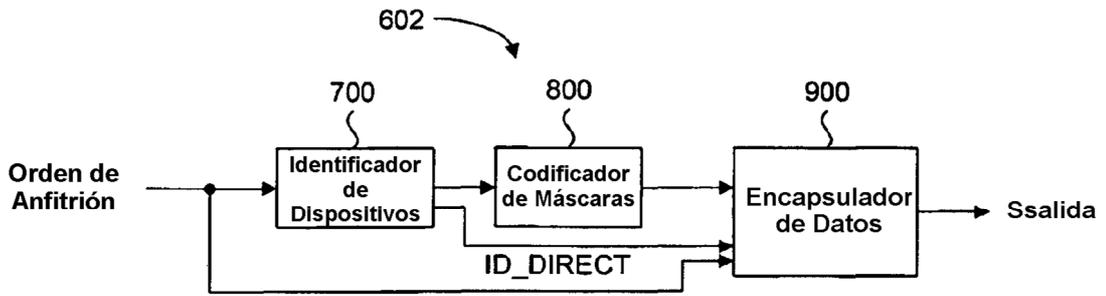


FIG. 7A

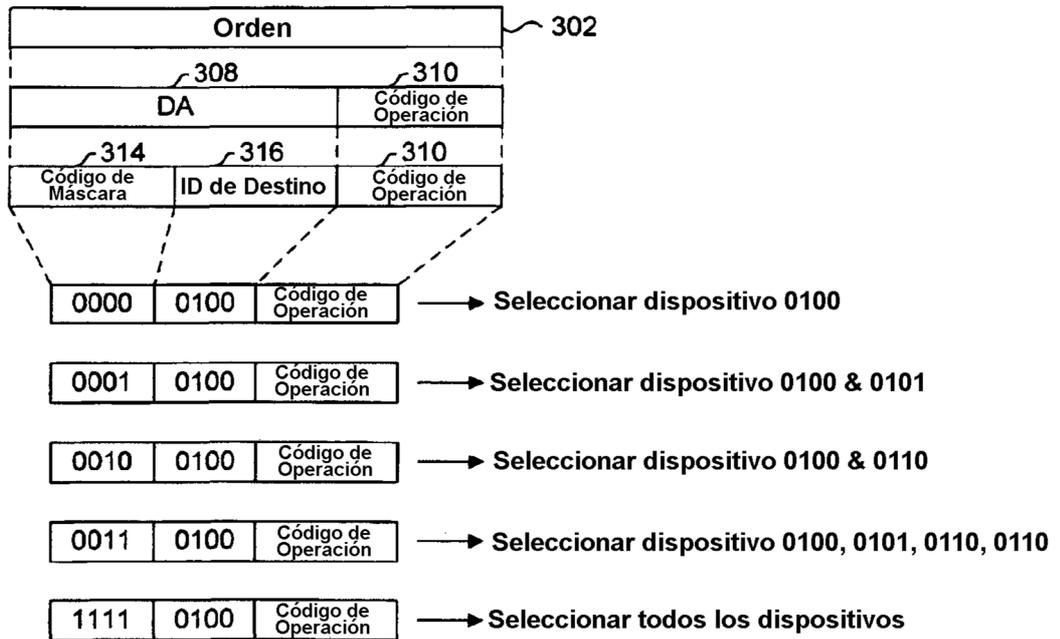


FIG. 7B

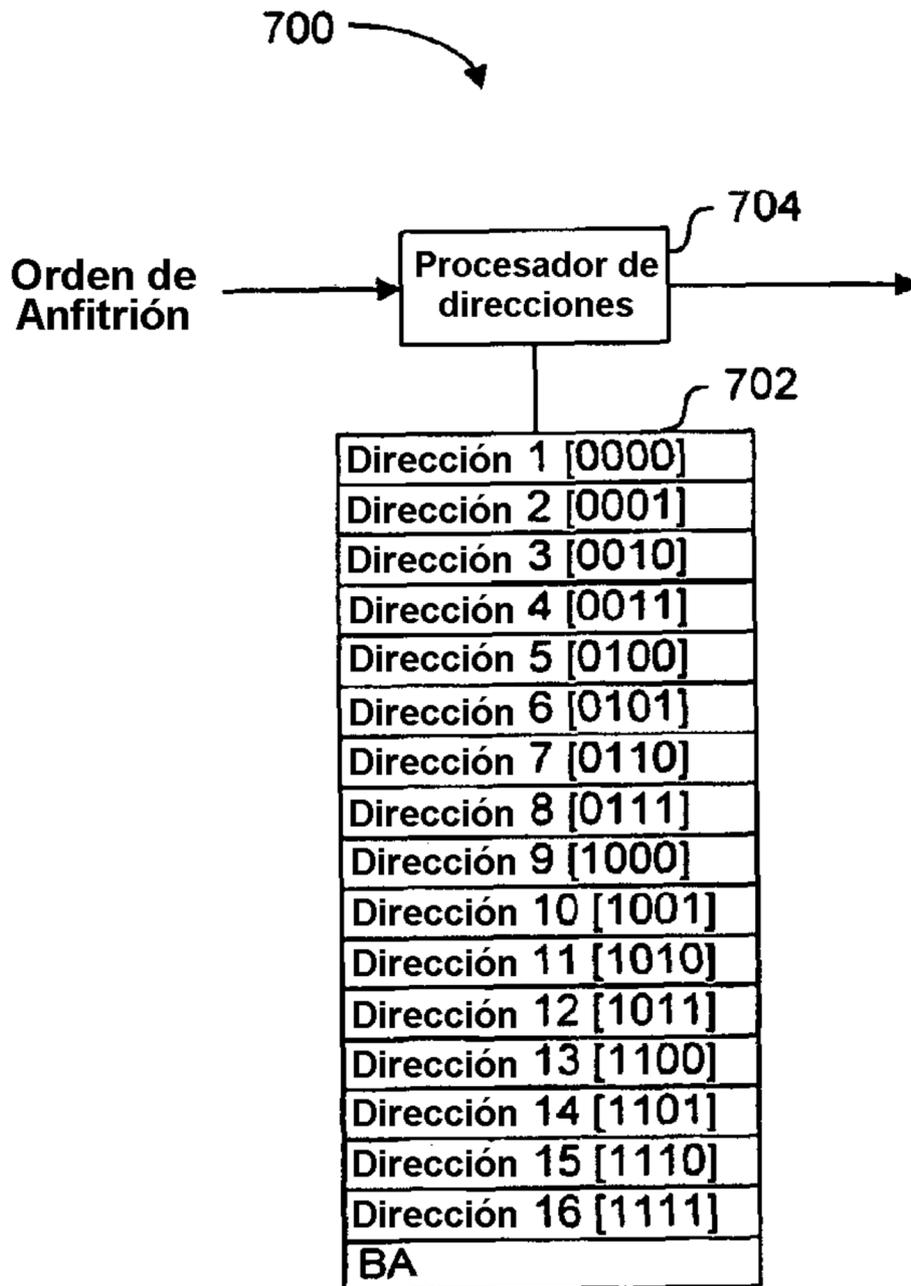


FIG. 8

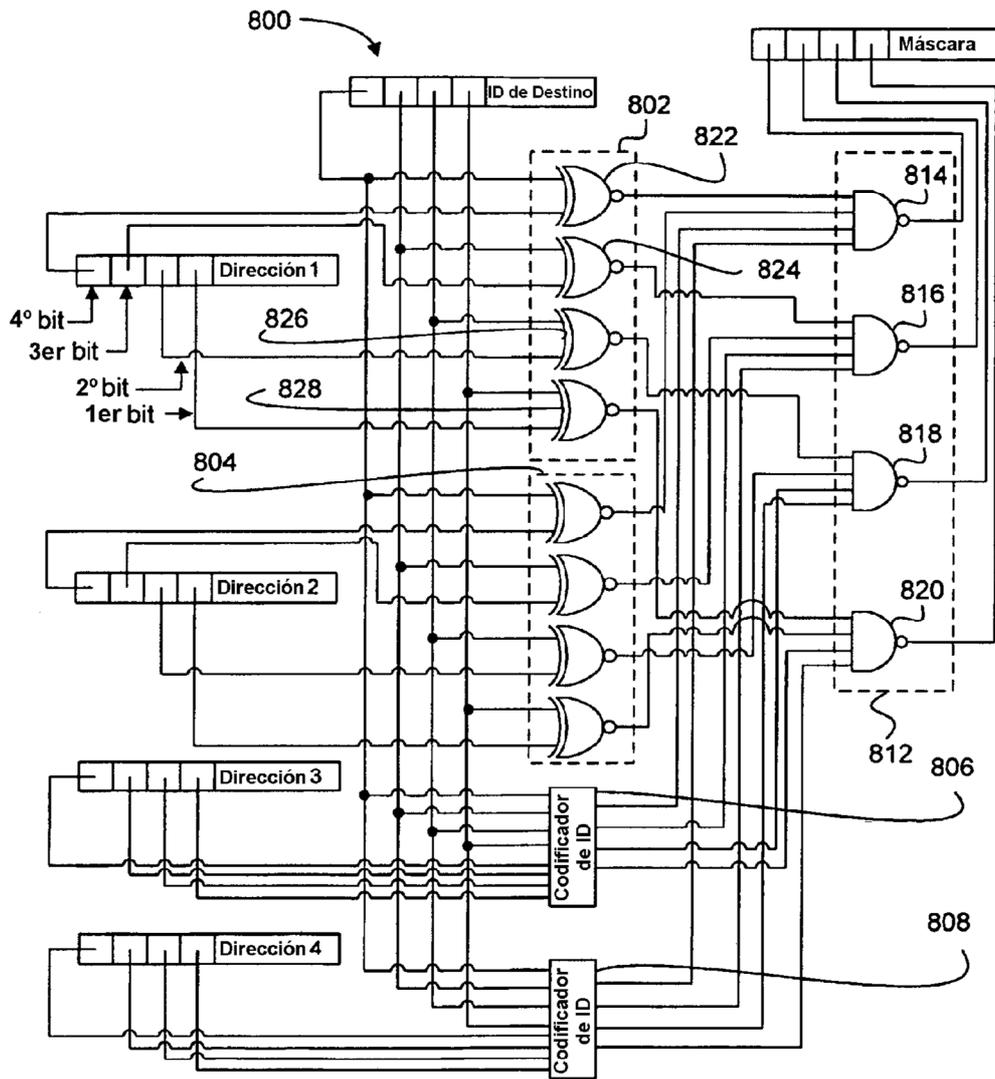


FIG. 9

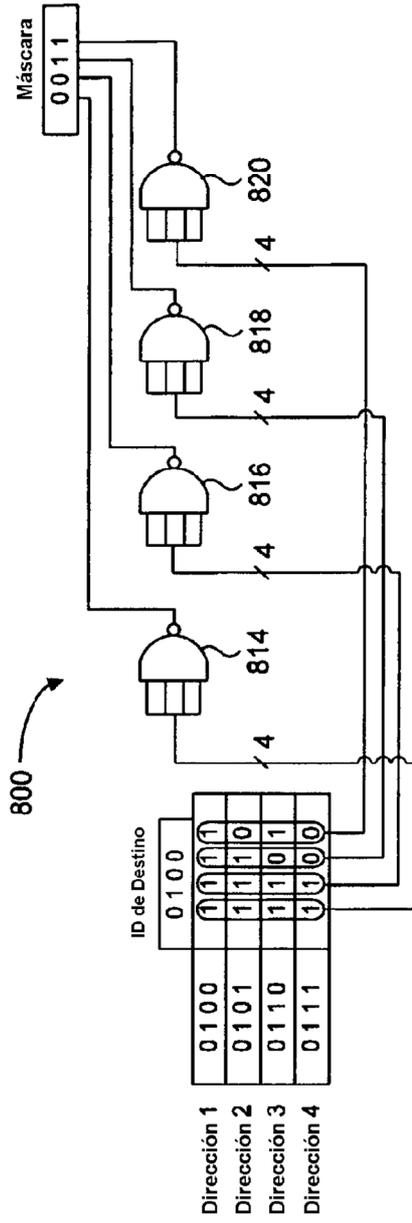


FIG. 10A

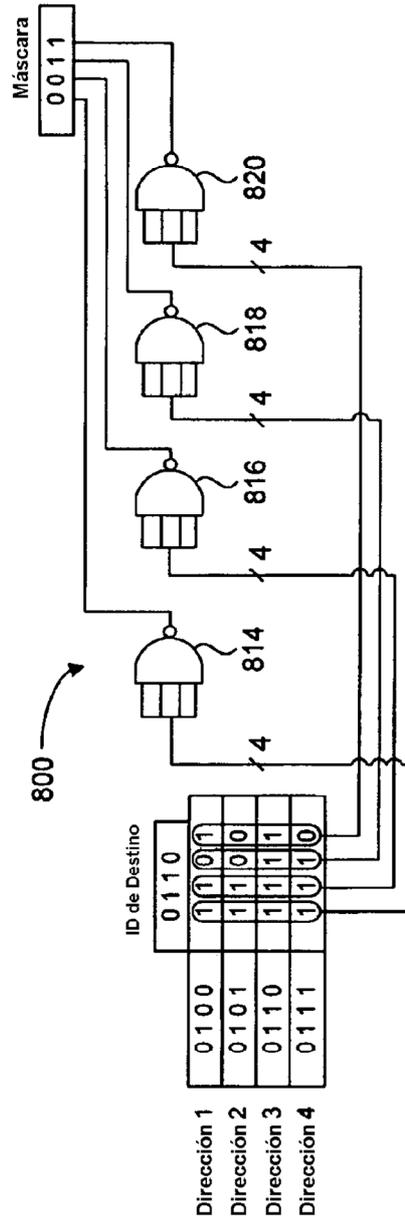


FIG. 10B

		Máscara
		0 0 1 1
Direcciones	0 1 0 0	0 1 1 1
	0 1 0 1	0 1 1 1
	0 1 1 0	0 1 1 1
	0 1 1 1	0 1 1 1

FIG. 11

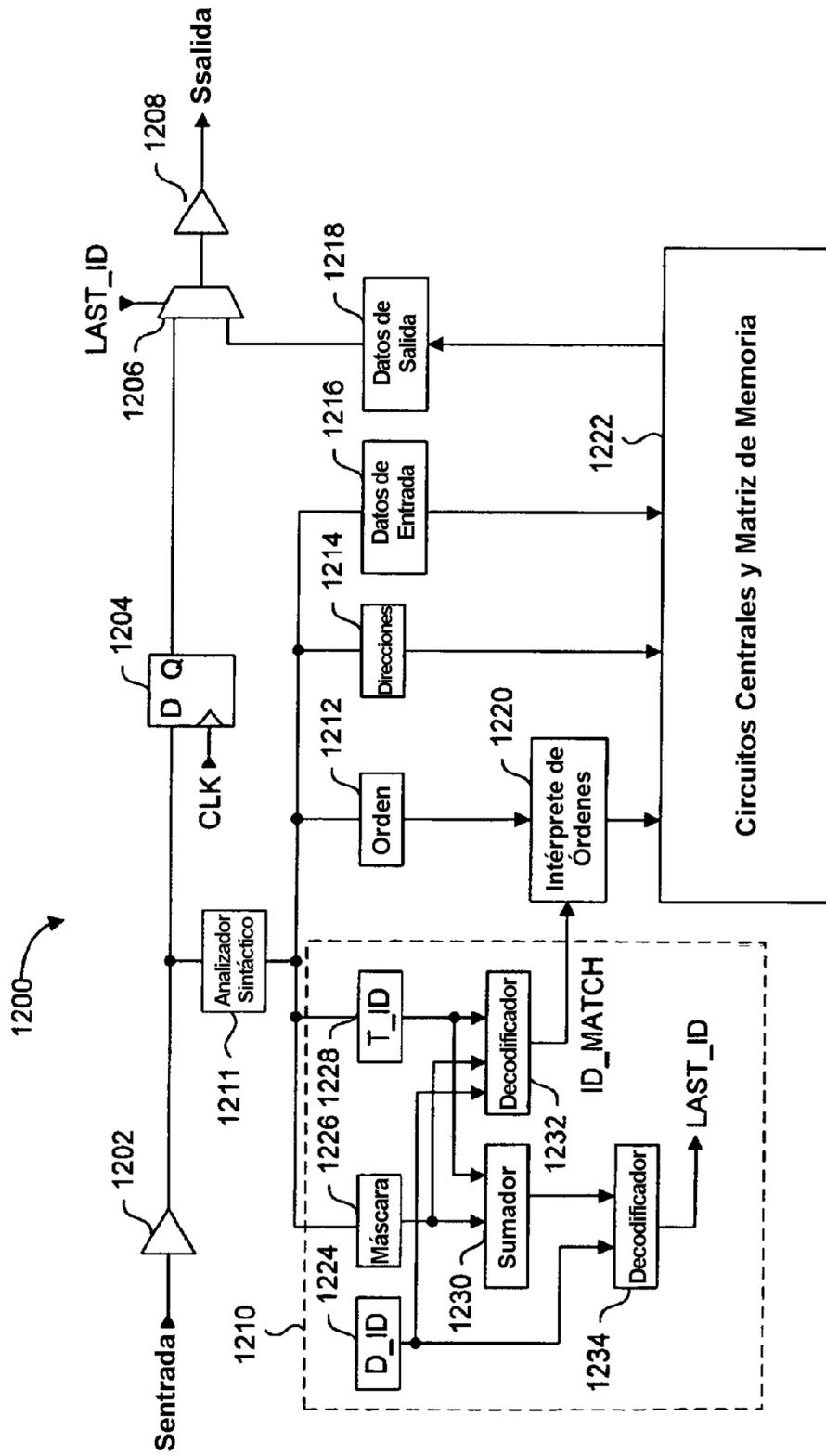


FIG. 12

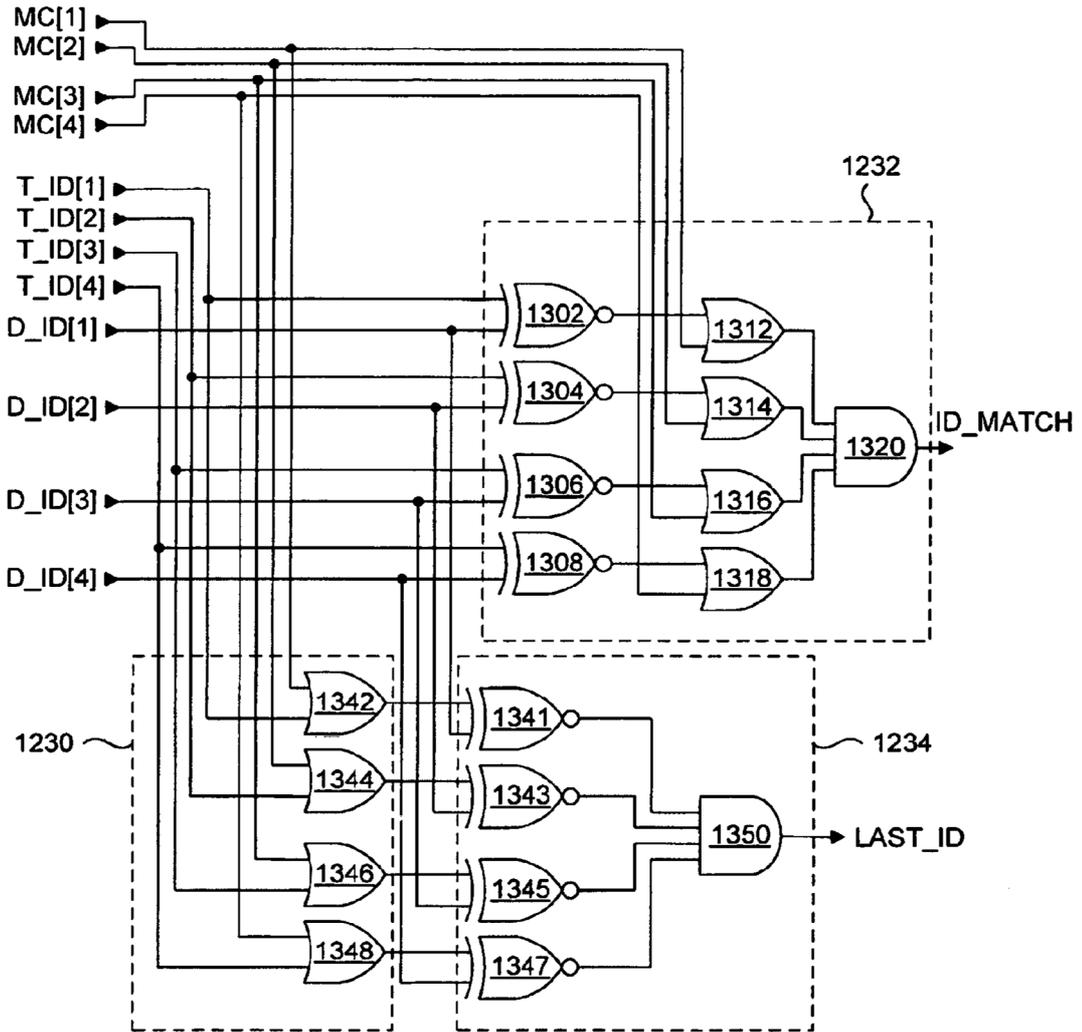


FIG. 13

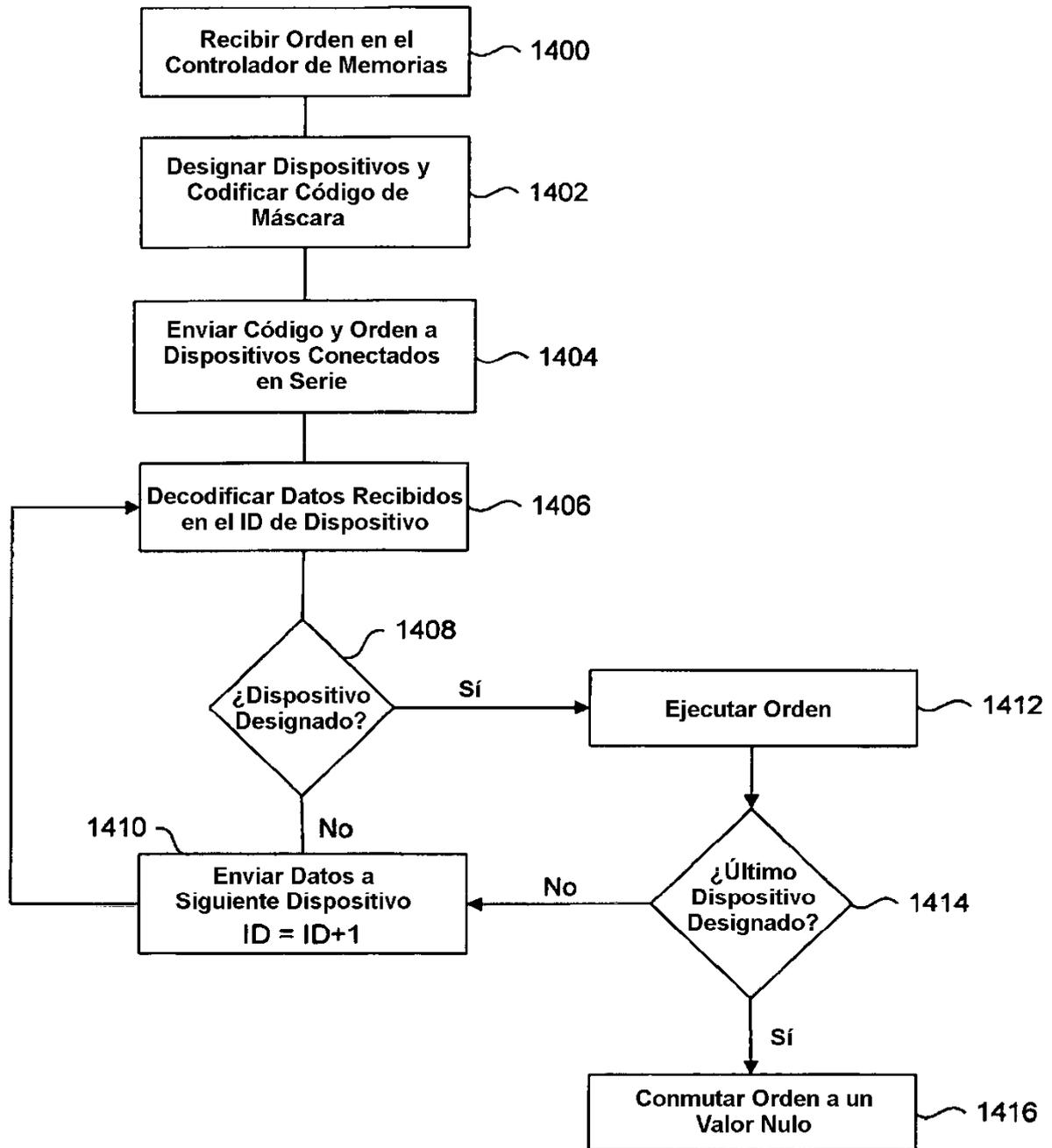


FIG. 14

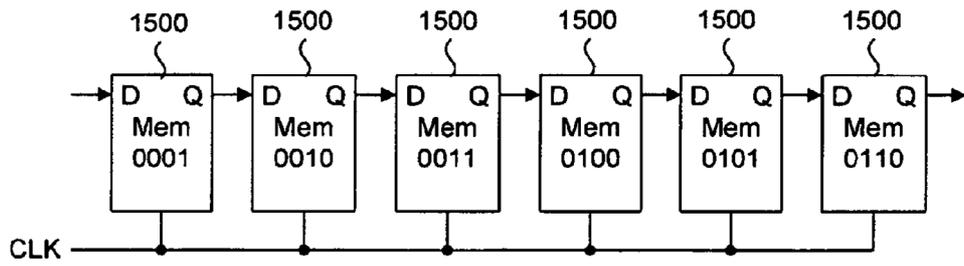


FIG. 15