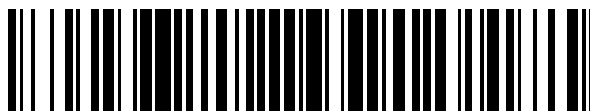


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 653 951**

51 Int. Cl.:

G06F 15/80 (2006.01)

G06F 9/30 (2006.01)

G06F 9/35 (2006.01)

G06F 13/16 (2006.01)

G06F 12/06 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **28.11.2012** **PCT/SE2012/051320**

87 Fecha y número de publicación internacional: **27.06.2013** **WO13095257**

96 Fecha de presentación y número de la solicitud europea: **28.11.2012** **E 12816532 (1)**

97 Fecha y número de publicación de la concesión europea: **18.10.2017** **EP 2751705**

54 Título: **Procesador digital de señales y método para direccionar una memoria en un procesador digital de señales**

30 Prioridad:

20.12.2011 SE 1151230

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

09.02.2018

73 Titular/es:

MEDIATEK SWEDEN AB (100.0%)

Teknikringen 10

583 30 Linköping, SE

72 Inventor/es:

NILSSON, ANDERS;

TELL, ERIC y

ALFREDSSON, ERIK

74 Agente/Representante:

IZQUIERDO BLANCO, María Alicia

ES 2 653 951 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).

Procesador digital de señales y método para direccionar una memoria en un procesador digital de señales**Descripción****5 Campo Técnico**

La presente invención se refiere a un procesador digital de señales de acuerdo con el preámbulo de la reivindicación 1. Dicho procesador es particularmente adecuado para sistemas OFDM.

10 Antecedentes y Técnica Relacionada

Muchos dispositivos de comunicación móviles usan un transceptor de radio que incluye uno o más procesadores digitales de señales (DSP).

15 Para un mayor rendimiento y fiabilidad muchos terminales móviles usan actualmente un tipo de DSP conocido como procesador de banda base (BBP), para manejar muchas de estas funciones de procesamiento de señales asociadas con el procesamiento de la señal de radio recibida y preparar señales para su transmisión.

20 Muchas de las funciones realizadas frecuentemente en tales procesadores se realizan en grandes números de muestras de datos. Por lo tanto, un tipo de procesador conocido como procesador Instrucción Única Múltiples Datos (SIMD) es útil ya que permite que la misma instrucción se realice para un vector completo de datos en lugar de un entero en un momento. Esta clase de procesador es capaz de procesar instrucciones de vectores, lo que significa que una única instrucción realiza la misma función para un número limitado de unidades de datos. Los datos se agrupan en bytes o palabras y se comprimen en un vector para ser operados.

25 Como un desarrollo adicional de la arquitectura SIMD, se ha desarrollado la arquitectura Instrucción Única flujo de Tareas Múltiples (SIMT). Tradicionalmente en la arquitectura SIMT se proporcionan una o dos unidades de ejecución de vectores que usan rutas de datos SIMD en asociación con una unidad de ejecución de números enteros, que puede ser parte de un procesador de núcleo. La Solicitud de Patente Internacional WO 2007/018467 divulga un DSP de acuerdo con la arquitectura SIMT, que tiene un núcleo del procesador que incluye una unidad de ejecución de números enteros y una memoria de programa, y dos unidades de ejecución de vectores que están conectadas a, pero no integradas en el núcleo. Las unidades de ejecución de vectores pueden ser Unidades Lógicas Aritméticas Complejas (CALU) o Unidades de Acumulación Múltiple Complejas (CMAC). Los datos a ser procesador en las unidades de ejecución de vectores se proporcionan de unidades de memoria de datos conectados con las unidades de ejecución de vectores a través de una red en chip.

35 Las unidades de memoria comprenden unidades de generación de direcciones que están dispuestas para controlar la orden de lectura o escritura en cualquier momento dado. Por flexibilidad aumentada, la unidad de generación de direcciones puede habilitar diferentes modos de lectura, o patrones, como lectura de cada enésima dirección en la memoria. Estos modos tienen que proporcionar un patrón regular, que limita las posibles maneras en que los datos pueden leerse o escribirse. Además, los modos disponibles se preseleccionan para una unidad de generación de direcciones particular, y no pueden cambiarse.

45 El artículo Nilsson, A y Tell, E: "An 11mm², 70mW fully programmable baseband processor for mobile WiMAX and DVB-T/H in 0.12 μ m CMOS", describe un DSP tipo SIMT y expone brevemente que "a medida que los bancos de memoria pueden aceptar direccionamiento externo de la red, pueden usarse memorias de números enteros así como aceleradores para proporcionar secuencias de direcciones para el direccionamiento de vectores irregulares. Esto también proporciona la capacidad de hacer direccionamiento de vectores indirecto". Este artículo no aborda cualquiera de los problemas implicados en implementar realmente tal solución, y por lo tanto tampoco proporciona una solución viable.

Sumario de la Invención

55 Es un objetivo de la presente invención permitir un direccionamiento más flexible de las memorias de datos de un procesador en una arquitectura SIMT.

60 Este objetivo se logra de acuerdo con la presente invención por un procesador digital de señales que comprende al menos una unidad funcional, que puede ser una unidad de ejecución de vectores o un acelerador, y al menos una primera unidad de memoria dispuesta para proporcionar datos a ser operados por la unidad funcional, una tercera unidad y una red en chip que conecta la unidad funcional, la primera unidad de memoria y la tercera unidad. El procesador digital de señales está caracterizado porque la tercera unidad está dispuesta para proporcionar datos de direccionamiento en la forma de un vector de dirección para ser usado para direccionar la primera unidad de memoria, dicha tercera unidad siendo conectable a la primera unidad de memoria de tal manera que los datos de direccionamiento pueden usarse para controlar la lectura de y/o la escritura en la primera unidad de memoria y porque el procesador comprende además una unidad de interfaz de dirección de memoria dispuesta

entre la primera unidad de memoria y la red en chip para compensar la latencia entre la primera unidad de memoria y la tercera unidad de memoria sirviendo como un almacenamiento intermedio de los datos de direccionamiento recuperados de la tercera unidad, en donde la unidad de interfaz de dirección de memoria está dispuesta para, cuando se configura la primera unidad de memoria para recibir los datos de direccionamiento de la tercera unidad, comenzar la lectura de los datos de direccionamiento de la tercera unidad antes de que la primera unidad de memoria solicite los datos de direccionamiento.

La invención también se refiere a un método de direccionamiento de una memoria en un procesador digital de señales que comprende al menos una unidad funcional y al menos una primera unidad de memoria dispuesta para proporcionar datos para ser operados por la unidad funcional, y una red en chip que conecta la unidad funcional y la primera unidad de memoria, y una tercera unidad dispuesta para proporcionar datos de direccionamiento para la primera unidad de memoria en la forma de un vector de dirección, dicho método comprendiendo los pasos de:

- configurar la primera unidad de memoria para recibir datos de direccionamiento de la tercera unidad, proporcionar datos de direccionamiento de la tercera unidad a la primera unidad de memoria,
- leer datos de, o escribir datos en la primera unidad de memoria de acuerdo con los datos de direccionamiento,
- compensar la latencia entre la primera y la tercera unidad por media de una unidad de interfaz de dirección de memoria (256) dispuesta entre la primera unidad de memoria y la red en chip para compensar la latencia entre la primera unidad de memoria y la tercera unidad sirviendo como un almacenamiento intermedio entre la primera y la tercera unidad, para los datos de direccionamiento recuperados de la tercera unidad, y
- cuando la primera unidad de memoria está configurada para recibir los datos de direccionamiento de la tercera unidad, comenzar a leer, por la unidad de interfaz de dirección de memoria los datos de direccionamiento de la tercera unidad, antes de que la primera unidad de memoria solicite los datos de direccionamiento.

Por lo tanto, de acuerdo con la invención, el direccionamiento puede lograrse rápida y eficientemente en cualquier orden a lo largo de la unidad de memoria. Los datos pueden direccionarse en una memoria en cualquier orden, tan eficientemente como una secuencia ordenada de datos ya que el direccionamiento puede manejarse en paralelo con el procesamiento realizado por la unidad funcional. La primera unidad de memoria puede disponerse para recibir datos de direccionamiento de la tercera unidad a través de un bus dedicado o a través de una red en chip.

La unidad de interfaz de dirección de memoria proporciona una solución a los problemas provocados por la latencia entre la primera y la tercera unidad. En particular, una señal de lectura de la primera unidad tardará algunos ciclos de reloj en alcanzar la tercera unidad. Cuando comienza a leer operaciones habrá un retardo de varios ciclos de reloj antes de que el primer artículo de los datos alcance realmente la unidad de ejecución. Almacenando las primeras muestras de direcciones en la unidad de interfaz de direcciones de memoria, antes de que la primera unidad los solicite, puede reducirse el retardo al inicio.

La tercera unidad puede ser cualquier unidad en el procesador, por ejemplo:

- una unidad de memoria, referida como la segunda unidad de memoria, preferiblemente una unidad de memoria de números enteros,
- una unidad de ejecución escalar,
- una unidad de ejecución de vectores, o
- una unidad de aceleradora

Pueden usarse diferentes unidades para el direccionamiento de diferentes bancos de memoria.

La primera unidad de memoria puede ser una memoria compleja o una memoria de números enteros.

La unidad de interfaz de dirección de memoria comprende preferiblemente:

- Medios de memoria de latencia dispuestos para almacenar un número representativo de la latencia entre la primera y la tercera unidad,
- Medios de memoria de muestra dispuestos para almacenar el número de muestras, es decir, el número de artículos de la dirección a ser transferidos de la tercera unidad a la primera unidad,
- Un registro para enviar los artículos de la dirección de la tercera unidad a la primera unidad.

Por consiguiente, la unidad de interfaz de dirección de memoria está preferiblemente dispuesta para realizar las siguientes funciones:

- leer al menos un primer artículo de la dirección de la tercera unidad cuando la primera unidad conecta con la tercera unidad para obtener información de la dirección, sin esperar a una señal de lectura

- cuando se recibe una señal de lectura de la primera unidad, enviar el primer artículo de la dirección a la primera unidad
- leer los siguientes artículos de la dirección de la tercera unidad posteriormente hasta que todos los artículos de la dirección se han leído y enviar los siguientes artículos de la dirección a la primera unidad posteriormente cuando se reciban las señales de lectura de la primera unidad.

La unidad de interfaz de dirección de memoria realiza un seguimiento del número de artículos de la dirección a leer de la tercera unidad cambiando el contador de muestras cada vez que se lee un artículo de la dirección. Esto es ventajoso porque la unidad de interfaz de dirección de memoria continuará recibiendo solicitudes de lectura de la primera unidad después de que haya leído la última muestra de la dirección de la tercera unidad. De esta manera la unidad de interfaz de dirección de memoria conoce cuando detener la recuperación de artículos de la dirección de la tercera unidad. Cuando la recuperación se detiene, sólo se envían las últimas muestras de la dirección de la unidad de interfaz de dirección de memoria a la tercera unidad.

Es a menudo importante que todos los cálculos se realicen con la misma sincronización, independientemente de donde vienen los datos. Como la latencia puede variar dependiendo de que unidades están implicadas, una ventaja de la invención es que la sincronización puede controlarse evitando diferencias de retardo provocadas por números diferentes de pasos de segmentación entre diferentes unidades.

En una realización particularmente ventajosa, la segunda unidad de memoria comprende una unidad de generación de direcciones dispuesta para controlar la lectura de datos de la dirección de la segunda unidad de memoria de acuerdo con un patrón predefinido. En esta realización, el método puede comprender el paso de controlar lecturas de las direcciones de la segunda unidad de memoria por medio de una unidad de generación de direcciones dispuesta para controlar la lectura de datos de la dirección de la segunda unidad de memoria de acuerdo con un patrón predefinido. Esto permite el manejo de subconjuntos de las direcciones mantenidos en la segunda unidad de memoria, que es particularmente ventajoso para aplicaciones de TV digital basadas en OFDM.

El procesador de acuerdo con la invención es particularmente muy adecuado para estándares de telecomunicaciones basados en OFDM, por ejemplo, de acuerdo con LTE y los estándares LTE avanzado.

Breve Descripción de los Dibujos

A continuación se describirá la invención con más detalle, a modo de ejemplo, y con referencia a los dibujos añadidos.

La Fig. 1 ilustra un ejemplo de la arquitectura SIMT.

La Fig. 2 ilustra la función de indicadores de dirección.

La Fig. 3 ilustra una primera realización de la invención en un ejemplo simplificado de la arquitectura SIMT.

La Fig. 4 ilustra una segunda realización de la invención en un ejemplo simplificado de la arquitectura SIMT.

La Fig. 5 ilustra una unidad de interfaz de dirección de memoria de acuerdo con una realización de la invención.

La Fig. 6 ilustra una tercera realización de la invención.

Descripción Detallada de Realizaciones

La Figura 1 ilustra un ejemplo de un procesador de banda base 200 del estado de la técnica de acuerdo con la arquitectura SIMT. El procesador 200 incluye un núcleo del controlador 201 y una primera 203 y una segunda 205 unidades de ejecución de vectores que se tratarán con más detalle a continuación. Una unidad FEC 206 está conectada a la red en chip. En una implementación concreta, por supuesto, la unidad FEC puede comprender varias unidades diferentes.

Una unidad de interfaz del host 207 proporcionar conexión a un procesador del host no mostrado en la Fig. 1 de una manera bien conocida en la técnica. Una unidad de terminal de entrada digital 209 proporciona conexión a una unidad de terminal de entrada de una manera bien conocida en la técnica.

Como es común en la técnica, el núcleo del controlador 201 comprende una memoria de programa 211 así como lógica de problemas de instrucción y funciones para soporte multi-contexto. Para cada contexto de ejecución, o hilo, soportado esto incluye contador de programa, puntero de pila y archivo de registro (no mostrados explícitamente en la figura 1). Típicamente, se soportan 2-3 hilos. El núcleo del controlador 201 también comprende una unidad de ejecución de números enteros 212 de un tipo conocido en la técnica.

La primera unidad de ejecución de vectores 203 en este ejemplo es una unidad de ejecución de vectores CMAC, y la segunda unidad de ejecución de vectores es una unidad de ejecución de vectores CALU. Cada unidad de ejecución de vectores 203, 205 comprende un controlador de vectores 213, una unidad de carga/almacenamiento de vectores 215 y un número de rutas de datos 217. El controlador de vectores de cada unidad de ejecución de

vectores está conectado a la memoria del programa 211 del núcleo del controlador 201 a través de lógica de problemas, para recibir señales de problemas relacionadas con instrucciones de la memoria del programa.

La función de las rutas de datos 217, 227 y las unidades de carga/almacenamiento de vectores 215, 225 es bien conocida en la técnica y no se tratará con detalle en este documento.

Podría haber un número arbitrario de unidades de ejecución de vectores, incluyendo sólo unidades CMAC, sólo unidades CALU o un número adecuado de cada tipo. Puede haber también otros tipos de unidad de ejecución de vectores distintos a CMAC y CALU. Como se ha explicado anteriormente, una unidad de ejecución de vectores es un procesador que es capaz de procesar instrucciones de vectores, lo que significa que una única instrucción realiza la misma función para una variedad de unidades de datos. Los datos pueden ser complejos o reales, y están agrupados en bytes o palabras y comprimidos en un vector a ser operado por una unidad de ejecución de vectores. En este documento, las unidades CALU y CMAC se usan como ejemplos, pero cabe señalar que las unidades de ejecución de vectores pueden usarse para realizar cualquier función adecuada en vectores de datos.

Como se conoce en la técnica, se usan típicamente un número de aceleradores 242, ya que permiten la implementación eficiente de ciertas funciones de banda base como codificación de canales e intercalación. Tales aceleradores son bien conocidos en la técnica y no se tratarán con detalle en la presente. Los aceleradores pueden ser configurables para ser reutilizados por muchos estándares diferentes.

Una red en chip 244 conecta el núcleo del controlador 201, la unidad de terminal de entrada digital 209, la unidad de interfaz de host 207, las unidades de ejecución de vectores 203, 205, los bancos de memoria 230, 232, el banco de número de enteros 238 y los aceleradores 242. Las unidades de ejecución de vectores, las unidades de ejecución escalares, las unidades de ejecución de números enteros y los aceleradores son referidos colectivamente en este documento como unidades funcionales. Una unidad de ejecución escalar es solo capaz de procesar una muestra a la vez, pero esta muestra puede ser un valor real o complejo.

Para permitir varias operaciones de vectores concurrentes, el procesador tiene preferiblemente un sistema de memoria distribuida donde la memoria se divide en varios bancos de memoria, representados en la Figura 1 por el banco de Memoria 0 230 al banco de Memoria N 231. Cada banco de memoria 230, 231 tiene su propia memoria 232, 233 y unidad de generación de direcciones AGU 234, 235 respectivamente. Las memorias 232, 233 son típicamente, pero no necesariamente, memorias complejas. Esta disposición en conjunción con la red en chip mejora la eficiencia de potencia del sistema de memoria y el rendimiento del procesador ya que pueden realizarse múltiples cálculos de direcciones en paralelo. El PBBP de la Fig. 2 incluye también preferiblemente bancos de memorias de números enteros 238, incluyendo cada uno una memoria 239 y una unidad de generación de direcciones 240.

Cada memoria tiene indicadores de direcciones que indican la posición en la memoria que debería leerse o escribirse después..

Por ejemplo, las instrucciones

out r0, CDM0_ADDR
out r1, CDM1_ADDR

establecen las posiciones en la memoria de datos complejos 0 y la memoria de datos complejos 1, respectivamente, que deberían leerse o escribirse.

Cada unidad de generación de direcciones 234, 235 realiza un cálculo de direcciones para controlar el orden en el que los datos se van a leer de o escribir en la memoria correspondiente 232, 233. Para flexibilidad aumentada, la unidad de generación de direcciones también puede disponerse para habilitar dos o más modos diferentes. Varios de tales modos son conocidos en la técnica. La lógica de generación de direcciones puede por ejemplo realizar: direccionamiento lineal, invertido de bits, de módulo y 2D con diferentes incrementos, incluyendo incrementos negativos. Por ejemplo, el generador de direcciones puede estar dispuesto para leer cada artículo de datos K-ésima de acuerdo con la función de direccionamiento, K siendo un número entero. Alternativamente, el generador de direcciones podría estar dispuesto para direccionar la memoria hacia atrás. Por lo tanto, como ejemplo, si la dirección de comienzo es 0, el generador de direcciones puede estar dispuesto para leer de la memoria correspondiente de acuerdo con tres modos diferentes:

- Leer cada dirección consecutivamente, es decir, 0, 1, 2, 3, 4, 5, 6, 7
- Leer cada dirección K-ésima, es decir, si K=2; 0, 2, 4, 6

Si la dirección de comienzo es 10 y K=2

- Leer hacia atrás, es decir 10, 8, 6, 4, 2, 0

Para, por ejemplo, hacer que una unidad de ejecución de vectores multiplique artículos de datos de dos memorias diferentes, la instrucción podría parecer como sigue:

cmac.n CDM0, CDM1

n siendo la longitud del vector a ser operado. Esto podría realizarse en un vector de n artículos de datos de cada una de las memorias CDM0 y CDM1, comenzado con el artículos de datos indicado por el indicador de la memoria respectiva.

Siempre y cuando los datos de una memoria particular se vayan a leer consecutivamente, o en un orden soportado por su unidad de generación de direcciones, no hay problemas. Pero el generador de direcciones no soporta una situación donde el vector a ser operado en la unidad de ejecución de vectores se mantenga en la memoria en un orden irregular. Por ejemplo, multiplicar juntas las secuencias

CDM0[1, 3, 7, 5, 11]

y

CDM1[0,1,2,3,4]

sería una operación complicada ya que no hay un patrón regular en la secuencia a ser leída desde CDM0. De acuerdo con el estado de la técnica, por lo tanto la dirección tendría que configurarse en el AGU manualmente, antes de cada lectura de datos de la memoria. Esto provocaría un retardo significativo en la lectura de datos que reduciría el rendimiento general del procesador.

La Figura 2 ilustra la función del indicador de direcciones en una primera y una segunda unidad de memoria. Cada unidad de memoria comprende un número de artículos de datos, incluyendo una secuencia de datos que se proporciona como datos de entrada a una unidad de ejecución de vectores.

La primera unidad de memoria 230 es CDM0 y la secuencia de datos relevante se muestra como un bloque 230a en esta unidad de memoria. Un indicador de direcciones 230b indica el comienzo de este bloque, para indicar donde debería comenzar la lectura de datos.

La segunda unidad de memoria 231 es CDM1 que tiene también una secuencia de datos a ser usada como datos de entrada por una unidad funcional, como una unidad de ejecución de vectores. Esta secuencia de datos se muestra como un bloque 231a, con un indicador de direcciones que indica 231b al comienzo de este bloque. Como puede verse, la localización de la secuencia de datos 231a en la segunda memoria 213 puede ser diferente de la localización de la secuencia de datos 230a en la primera memoria 230.

A medida que procede la lectura, el indicador se moverá para indicar la siguiente dirección a ser leída en cualquier momento dado. Tradicionalmente, la información del indicador se toma de la unidad de generación de direcciones 234, 235, de la memoria correspondiente 230, 231.

La Figura 3 es un dibujo simplificado que muestra sólo las partes de la arquitectura SIMT que son particularmente relevantes para la presente invención. Las partes mostradas, usando los mismos números de referencia que en la Figura 1, son: una de las unidades de ejecución de vectores, en este ejemplo, la unidad de ejecución de vectores CALU 205, el primer 230 y el segundo 231 bancos de memoria, y la red en chip 244 que conecta estas tres unidades juntas. Como antes, cada uno de los bancos de memoria 230 y 231 comprende una memoria 232, 233, y una unidad de generación de direcciones 234, 235. Hay también un tercer banco de memoria 250 que comprende una memoria 252 y una unidad de generación de direcciones 254. El tercer banco de memoria está también conectado a las otras unidades a través de la red 244.

El tercer banco de memoria 250 es típicamente un banco de memoria de números enteros, lo que le hace adecuado para mantener información de direcciones en la forma de un vector de datos que puede ser referido como un vector de datos. A este banco de memoria se hace referencia algunas veces en este documento como la memoria de direcciones. Por otra parte, puede ser exactamente del mismo tipo que el primer y el segundo bancos de memoria 230, 231, que son referidos como memorias de datos. De acuerdo con la invención la memoria 252 del tercer banco de memoria 250 mantiene un vector de direcciones para ser usado para direccionar el segundo banco de memoria 231. Como se muestra simbólicamente por la flecha que conecta el segundo y el tercer bancos de memoria la conexión pasa por alto la unidad de generación de direcciones 235 del segundo banco de memoria 231 para direccionar la memoria 233 directamente.

Como los datos en la memoria 252 del tercer banco de memoria 250 pueden cambiar bastante fácilmente, esto proporciona una manera muy flexible de direccionar el segundo banco de memoria 231.

Leer los datos de dirección de una unidad de memoria separada introduce adicionalmente latencia en el sistema. Cuando la unidad de ejecución está lista para comenzar a recibir datos del segundo banco de memoria, enviará una señal de lectura al segundo banco de memoria, que es el banco que proporciona los datos para los cálculos realizados por la unidad de ejecución. El segundo banco de memoria enviará luego una señal de lectura al banco de memoria de direcciones. Sólo entonces puede el segundo banco de memoria enviar los artículos de datos a la unidad de ejecución. Por lo tanto habrá una latencia que provocará un retardo en el comienzo de la ejecución de vectores.

Para superar este retardo, en esta realización una unidad de interfaz de dirección de memoria 256 está dispuesta entre el banco de direcciones 250 y el segundo banco de memoria 231. La unidad de interfaz de dirección de memoria 256 sirve como una unidad de interfaz de dirección de memoria para el almacenamiento inmediato de los artículos de las direcciones recuperados del tercer banco de memoria 250. El diseño y funciones de la unidad de interfaz de dirección de memoria 256 se tratará con más detalle en relación con la Figura 5.

Como un complemento, la unidad de generación de direcciones del tercer banco de memoria 250 también puede usarse para configurar un modo de lectura, como se ha tratado anteriormente en relación con la Figura 1, por ejemplo para leer cualquier otro elemento de datos del tercer banco de memoria. Esto significa que en algunas situaciones los mismos contenidos del tercer banco de memoria pueden usarse para aplicaciones diferentes. Por ejemplo, podría conseguirse una función de repetición.

Como se entenderá, podría usarse una disposición similar también para direccionar el primer banco de memoria 230, o la unidad de generación de direcciones 234 del primer banco de memoria podría usarse de una manera convencional. Además, podrían proporcionarse cualquier número de bancos de memoria y unidades funcionales de cualquier tipo. Podría haber un número de bancos de memoria que podrían usarse como memorias de direcciones. Un banco de memoria y un banco de memoria podrían cambiar fácilmente el orden en que se sus entradas se leen o escriben conectando con la memoria de direcciones apropiada, ya que todas las unidades están interconectadas a través de la red 244.

La Figura 4 ilustra otra realización de la invención. Las partes mostradas, usando los mismos números de referencia que en la Figura 1, son: la unidad de ejecución de vectores CMAC 203, la unidad de ejecución de vectores CALU 205, el primer 230 y el segundo 231 bancos de memoria, y la red 244 que conectar estas tres unidades entre sí. Como antes, cada uno de los bancos de memoria 230 y 231 comprende una memoria 232, 233, y una unidad de generación de direcciones 234, 235. También se muestra un tercer banco de memoria 250, que comprende una memoria 252 y una unidad de generación de direcciones 254. El tercer banco de memoria también está conectado a las otras unidades a través de la red 244 y puede usarse como una memoria de direcciones como se ha tratado en relación a la Figura 3. En la realización mostrada en la Figura 4, el segundo banco de memoria 235 se direcciona desde la unidad de ejecución de vectores CMAC 203. Esto significa que las direcciones a ser leídas desde el segundo banco de memoria 231 están siendo calculadas en la unidad de ejecución de vectores CMAC 203. Por supuesto, esto es solamente un ejemplo. Ya que el experto se dará cuenta que el direccionamiento podría realizarse desde una unidad de ejecución de vectores CALU o desde cualquier otro tipo de unidad funcional como una unidad de ejecución de vectores, otra unidad de ejecución o acelerador.

Como se entenderá, las realizaciones mostradas en las Figuras 4 y 5 podrían combinarse de manera que algunos bancos de memoria se direccionasen por medio de sus unidades de generación de direcciones internas 231, 232 otros se direccionarían desde bancos de memoria separados 250, y aún otros desde unidades funcionales 203, 205.

Usando otra unidad de ejecución de vectores para calcular las direcciones desde la que leer en una memoria particular, el direccionamiento de memoria puede lograrse de una manera muy flexible. Esto es particularmente útil en aplicaciones como

- Extracción de pilotos y separación de usuarios en sistemas OFDM
- Procesamiento de ramas Rake en sistemas CDMA

Un método de acuerdo con la invención para habilitar el direccionamiento de una unidad de memoria desde otra unidad de la red, por ejemplo, la memoria de datos de números enteros IDM:

- 1) Configurar la primera unidad de memoria, por ejemplo CDM0 para usar IDM como una fuente de direcciones.
- 2) Configurar la segunda unidad de memoria, por ejemplo, CDM1 para usar direccionamiento lineal por medio de su unidad de generación de direcciones.
- 3) Procesar los datos en la unidad de ejecución de vectores, por ejemplo cálculos, en los datos proporcionados desde la primera y la segunda unidad de memoria en el orden que se proporcionan.
- 4) Para cada artículo de datos leído de la unidad de memoria por la unidad de ejecución de vectores, la memoria se programa para recuperar una nueva dirección de la red, es decir, o de la tercera unidad de

memoria, en la realización de la Fig. 3, o de la segunda unidad de ejecución de vectores, en la realización de la Fig. 4. La nueva dirección indicará la posición en la primera unidad de memoria desde la que los datos se van a leer después.

Alternativamente, para escribir los resultados del procesamiento realizado por una unidad de ejecución de vectores en una memoria de datos:

1) Configurar la primera unidad de memoria, por ejemplo, CDM0 para usar IDM como una fuente de direcciones.

2) Configurar la segunda unidad de memoria, por ejemplo, CDM1 para usar direccionamiento lineal por medio de su unidad de generación de direcciones.

3) Procesar los datos en la unidad de ejecución de vectores, por ejemplo cálculos, y escribir el resultado en una memoria de datos.

4) Para cada artículo de datos escrito en la unidad de memoria de datos por la unidad de ejecución de vectores, la memoria se programa para recuperar una nueva dirección de la red, es decir, o de la tercera unidad de memoria, en la realización de la Fig. 3, o de la segunda unidad de ejecución de vectores, en la realización de la Fig. 4. La nueva dirección indicará la posición en la memoria de datos en la que los datos deben escribirse después.

En los métodos de ejemplo anteriores, por supuesto los datos de direccionamiento podrían obtenerse de una unidad de ejecución de vectores o de otra unidad en el procesador, en lugar de la memoria de direcciones.

En ambos ejemplos mostrados en las Figuras 3 y 4, la información de direcciones a ser proporcionada desde la memoria de direcciones, o desde la unidad de ejecución de vectores apropiada, respectivamente, debe sincronizarse de tal manera que la siguiente dirección a ser leída de, o escrita en, alcance el segundo banco de memoria un ciclo de reloj antes de que realmente deba leerse de o escribirse en. Es decir: debería hacer una lectura por delante de los elementos de datos debido a la segmentación. La cantidad de datos que se lee antes puede controlarse por señales de control que se propagan desde el bloque de memoria que está direccionado a través de la red a la fuente de direcciones. Alternativamente, puede controlarse por un valor fijado programado en la unidad de interfaz de direcciones de memoria. La lectura anterior puede implementarse también por fuentes de direcciones que introducen una cantidad predefinida de datos de direcciones sobre la red, donde el número de etapas de la segmentación se codifica en el hardware.

Para superar los problemas provocados por la latencia entre la unidad de ejecución de vectores 203 que proporciona los datos de direcciones y el banco de memoria 231 que va a usar los datos de direcciones, una unidad de interfaz de dirección de memoria 256 está dispuesta entre la unidad de ejecución de vectores 203 y el segundo banco de memoria 231. La unidad de interfaz de dirección de memoria 256 es similar a la unidad de interfaz de dirección de memoria 256 de la Figura 3 y sirve como una unidad de interfaz de dirección de memoria para el almacenamiento intermedia de los artículos de la dirección recuperados del tercer banco de memoria 250.

La Figura 5 muestra una unidad de interfaz de dirección de memoria 256 de acuerdo con una realización preferida de la invención. La interfaz de memoria tiene una memoria 258 y una unidad de control 260. La unidad de control 260 comprende dos memorias:

- una memoria de latencia 262 que mantiene el número de pasos de segmentación requeridos para leer de la tercera unidad a la primera memoria.
- un contador de muestras 264 dispuesto para hacer seguimiento del número de muestras de direcciones a ser leídas de la tercera unidad 203, 250.

La memoria de latencia 262 está típicamente, pero no necesariamente, pre-programada. La memoria de recuento de muestras está dispuesta para ser configurada para cada operación como sea necesario.

Como se ha tratado anteriormente la tercera unidad es la que proporciona información de las direcciones. Esta puede ser una unidad de memoria 250 como se muestra en la Figura 3 o una unidad de ejecución de vectores 203 como se muestra en la Figura 4.

Cuando el banco de memoria de datos (no mostrado en la Figura 5) está configurado para recibir datos de direccionamiento desde la tercera unidad, que proporciona direcciones, la unidad de interfaz de dirección de memoria 256 lee los primeros artículos de los datos de la dirección de la tercera unidad a su memoria 258. De esta manera, cuando la unidad de ejecución que va a recibir los datos del banco de memoria de datos envía una señal de lectura para señalar que ya está listo para recibir el primer artículos de la dirección, el primer artículo de la dirección ya está almacenado en la unidad de interfaz de dirección de memoria y puede ser enviado a la segunda unidad sin retardo. Sin la unidad de interfaz de dirección de memoria, el procedimiento sería

- la unidad de ejecución envía una señal de lectura a la unidad de memoria

- la unidad de memoria envía una señal de lectura a la unidad que va a proporcionar la dirección
 - la unidad que va a proporcionar la dirección responde enviando la primera dirección.
 - la unidad de memoria, tras recibir la primera dirección, envía el artículo de datos a la unidad de ejecución.
- Por lo tanto, tomará varios ciclos de reloj antes de que la unidad de ejecución pueda empezar a trabajar. Una vez comenzada, sin embargo, las direcciones podrían ser entregadas al ritmo apropiado.

Para configurar el sistema se realizan los pasos siguientes:

1. El núcleo ordena el direccionamiento externo de la memoria de datos enviando una señal a la interfaz de memoria para llenar la cola a la memoria de datos o escribiendo sus registros de control.
2. La unidad de interfaz de dirección de memoria realiza un número suficiente de operaciones de lectura desde la unidad que proporciona los datos de direcciones para tener en su memoria un número de artículos de la dirección correspondientes a la latencia de la red. Esto significa que el número de artículos de la dirección debería corresponderse con el número de pasos de segmentación que se tienen que realizar para recuperar la dirección.

Durante la ejecución, la unidad de interfaz de dirección de memoria continúa enviando artículos de datos desde su registro a la unidad de memoria de datos y recuperando nuevos artículos de datos de direcciones consecutivamente.

La unidad funcional que recibe los artículos de datos continuará enviando señales de lectura a la unidad de memoria hasta que haya recibido el número apropiado de artículos de datos, y la unidad de memoria enviará a su vez señales de lectura a la unidad de interfaz de dirección de memoria. Como algunos artículos de datos ya fueron leídos de la unidad que proporciona las direcciones antes de que la unidad funcional comenzase a enviar señales de lectura, esto significa que algunas señales de lectura se enviarán después de que todos los artículos de la dirección se hayan leído de la unidad que proporciona direcciones.

Por lo tanto, la latencia significa que la unidad de interfaz de dirección de memoria continuará leyendo artículos de la dirección de la tercera unidad después de tener que parar. Más precisamente leería tantos artículos de la dirección como el número que almacenó en su registro cuando estaba siendo configurada en el paso 2 anterior. Para evitar que esto suceda, el contador de muestras realiza un seguimiento del número de muestras de direcciones que se han recuperado. Cuando el número deseado de muestras de direcciones se ha leído de la tercera unidad en la unidad de interfaz de dirección de memoria, la unidad de interfaz de dirección de memoria para de recuperar nuevas muestras de direcciones aunque continuará recibiendo señales de lectura desde la memoria de datos. En cambio, para los últimos artículos de la dirección, la unidad de interfaz de dirección de memoria vaciará su memoria para proporcionar estos artículos a la memoria de datos.

En una realización preferida, la información de direcciones está sincronizada de tal manera que la siguiente dirección a ser leída de, o escrita en, llega al segundo banco de memoria un ciclo de reloj antes de lo que realmente debería ser leída o escrita.

La Figura 6 muestra una realización ventajosa de un procesador de acuerdo con la invención, en el que un número de unidades de memoria pueden compartir un número más pequeño de unidades de interfaz de dirección de memoria. Se usan los mismos números de referencia que anteriormente para las mismas unidades como se muestra en las Figuras anteriores. Como puede verse el procesador de acuerdo con esta realización tiene las mismas unidades que los procesadores mostrados en las Figuras 3 y 4, todos conectados a través de la red en chip 244. Adicionalmente, la realización de la Figura 6 tiene una barra cruzada de direcciones 270 dispuesta para proporcionar información de las direcciones a la unidad de memoria deseada 230, 231. En la Figura 6, se muestran dos unidades de interfaz de dirección de memoria 256, ambas de las cuales están conectadas con la barra cruzada de direcciones 270. La barra cruzada de direcciones funciona entonces como un concentrador, seleccionando para cada unidad de memoria 230, 231 de cuál de las unidades de interfaz de dirección de memoria 256 debería recibir los datos de direccionamiento.

Las realizaciones de la invención son particularmente útiles en aplicaciones en las que se usan patrones de direcciones complejos, que no pueden ser pre-programadas en el momento del diseño o no son factibles de ser almacenados predefinidos en la memoria del sistema. Tales patrones podrían basarse en parámetros de tiempo de ejecución y deben computarse dinámicamente.

Por ejemplo, el estándar de telecomunicaciones basado en OFDM conocido como LTE (Evolución a Largo Plazo) usa asignación dinámica de frecuencias a los usuarios. Es necesario seleccionar las frecuencias asignadas a un usuario. En otras situaciones se desea seleccionar todos los tonos de pilotos, lo que puede hacerse en base a una tabla. Tradicionalmente, este se logra mirando en una tabla para obtener la información de la dirección de los tonos de pilotos, luego se carga el artículo de datos deseado de la memoria, seleccionando aleatoriamente los datos en las frecuencias a colocar los puntos de datos relevantes adyacentes entre sí y luego almacenar los puntos de datos de vuelta en la memoria.

Este tipo de patrón de direcciones no puede programarse en una unidad de generación de direcciones tradicional. Esto significa que, por ejemplo, para realizar un FFT de los tonos de pilotos, el indicador de dirección tendrá que configurarse varias veces para cada operación, significando que solo la administración de los datos será tan complicada que disminuirá el rendimiento (utilización) del procesador DSP significativamente.

Si, en su lugar, la unidad de memoria se programa para recuperar una nueva dirección de la red en chip, como se ha tratado anteriormente, cada punto de datos a ser usado puede direccionarse directamente, reduciendo la capacidad necesaria para la administración de datos, y aumentando de esta manera la utilización y rendimiento del DSP.

Otra aplicación en la que la presente invención es particularmente útil es en aplicaciones de TV digital. Cada símbolo de OFDM en DVB-T2 consiste de hasta 32768 sub-portadores, resultando en un conjunto de 32768 puntos de datos. Esta señal comprende tonos de pilotos a ser usados como datos de referencia, que se distribuyen de manera desigual a lo largo del espectro de frecuencia. Dicho patrón de direcciones no puede ser manejados fácilmente en la unidad de generación de direcciones tradicionales. De acuerdo con la invención, las direcciones podrían simplemente ser almacenadas en la tercera unidad de memoria y escogidas de allí por la unidad de memoria que proporciona los datos a la unidad de ejecución de vectores.

Para TV digital es también posible usar sólo un subconjunto de los 32768 (32k) puntos. El subconjunto puede ser de 16k, 8k, 4k, 2k o 1k puntos, es decir, medio, cuarto, etc., hasta 1/32 de los puntos. De acuerdo con la invención, sería sólo necesario almacenar una tabla de direcciones, ya que podría seleccionarse un subconjunto de esta tabla configurando las direcciones en consecuencia en la memoria de direcciones.

La invención es también útil cuando se van a recibir datos de direccionamiento de unidades que tienen sincronización impredecible. Ejemplos de tales unidades son co-procesadores programables o unidades de corrección de errores como turbo descodificadores. Típicamente los co-procesadores programables pueden entregar un flujo de direcciones con un rendimiento medio que coincide con el requerido, pero los datos se entregan en ráfagas pequeñas. De la misma manera, los bloques de corrección de errores trabajarán iterativamente en un conjunto de datos hasta que sea correcto, y es imposible predecir exactamente cuántos ciclos tardará. Por lo tanto la producción de tales unidades serán ráfagas impredecibles de datos. Una unidad de interfaz de dirección de memoria de acuerdo con la invención puede usarse para igualar las ráfagas de datos. Si el registro 258 mostrado en la Figura 5 se reemplaza con una cola FIFO la unidad de interfaz de dirección de memoria puede almacenar el número de datos contenidos en una ráfaga y enviarlos consecutivamente a la siguiente unidad.

Reivindicaciones

1. Un procesador digital de señales (200) que comprende al menos una unidad funcional, que puede ser una unidad de ejecución de vectores (203, 205), o un acelerador, y al menos una primera unidad de memoria (230, 231) dispuesta para proporcionar datos para ser operados por la unidad funcional, una tercera unidad (250) y una red en chip (244) que conecta la unidad funcional, la primera unidad de memoria y la tercera unidad, dicho procesador digital de señales estando **caracterizado porque** la tercera unidad (250) está dispuesta para proporcionar datos de direccionamiento en la forma de un vector de direcciones para ser usado para direccionar la primera unidad de memoria (230, 231), dicha tercera unidad siendo conectable a la primera unidad de memoria (230, 231) de tal manera que los datos de direccionamiento pueden usarse para controlar la lectura de y/o la escritura en la primera unidad de memoria (230, 231) y porque el procesador comprende además una unidad de interfaz de dirección de memoria (256) dispuesta entre la primera unidad de memoria y la red en chip para compensa la latencia entre la primera unidad de memoria y la tercera unidad sirviendo como un almacenamiento intermedio de los datos de direccionamiento recuperados de la tercera unidad, en donde la unidad de interfaz de dirección de memoria está dispuesta para, cuando la primera unidad de memoria está configurada para recibir los datos de direccionamiento de la tercera unidad, comenzar la lectura de los datos de direccionamiento de la tercera unidad antes de que la primera unidad de memoria solicite los datos de direccionamiento.
2. Un procesador de acuerdo con la reivindicación 1, en donde la tercera unidad es una unidad de memoria de direcciones, preferiblemente una memoria de números enteros, que mantiene los datos de direcciones para direccionar la primera unidad de memoria.
3. Un procesador de acuerdo con la reivindicación 1, en donde la tercera unidad es una segunda unidad funcional.
4. Un procesador de acuerdo con la reivindicación 1 ó 2, en donde la primera unidad de memoria (230, 231) es una memoria compleja.
5. Un procesador de acuerdo con cualquiera de las reivindicaciones anteriores 2 ó 4 cuando depende de la reivindicación 2, en donde la segunda unidad de memoria comprende una unidad de generación de direcciones dispuesta para controlar la lectura de los datos de direcciones de la segunda unidad de memoria de acuerdo con un patrón predefinido.
6. Un procesador de acuerdo con cualquiera de las reivindicaciones anteriores, que comprende además una barra cruzada de direcciones (270) que interconecta al menos una unidad de interfaz de dirección de memoria (256) y al menos una primera y una segunda unidad de memoria (230, 231), para permitir que los datos de direcciones se proporcionen a una seleccionada de la primera y la segunda unidad de memoria a través de la unidad de interfaz de dirección de memoria (256).
7. Un procesador de acuerdo con cualquiera de las reivindicaciones anteriores, adaptado para telecomunicaciones, por ejemplo, de acuerdo con LTE y/o estándar LTE avanzado.
8. Un procesador de acuerdo con cualquiera de las reivindicaciones 1-7, adaptado para señales de televisión digital.
9. Un método de direccionamiento de una memoria en un procesador digital de señales (200) que comprende al menos una unidad funcional, que puede ser una unidad de ejecución de vectores (203, 205) o un acelerador, y al menos una primera unidad de memoria (230, 231) dispuesta para proporcionar datos para ser operados por la unidad de ejecución de vectores, y una tercera unidad (250) dispuesta para proporcionar datos de direccionamiento en la forma de un vector de direcciones, y una red en chip (244) que conecta la unidad de ejecución de vectores y la primera y la tercera unidades de memoria, dicho método comprendiendo los pasos de:
 - configurar la primera unidad de memoria para recibir datos de direccionamiento desde la tercera unidad,
 - proporcionar datos de direccionamiento desde la tercera unidad a la primera unidad de memoria,
 - leer los datos de, o escribir los datos en la primera y la tercera unidad por medio de una unidad de interfaz de dirección de memoria (256) dispuesta entre la primera unidad de memoria y la red en chip para compensar la latencia entre la primera unidad de memoria y la tercera unidad sirviendo como un almacenamiento intermedio entre la primera y la tercera unidad, para los datos de direccionamiento recuperados de la tercera unidad, y
 - cuando la primera unidad está configurada para recibir los datos de direccionamiento desde la tercera unidad, comenzar a leer, por la unidad de interfaz de dirección de memoria los datos de direccionamiento de la tercera unidad antes de que la primera unidad solicite los datos de direccionamiento.
10. Un método de acuerdo con la reivindicación 9, en donde la tercera unidad es una unidad de memoria de direcciones, preferiblemente una unidad de memoria de números enteros, que mantiene datos de direcciones para direccionar la primera unidad de memoria.

11. Un método de acuerdo con la reivindicación 9, en donde la tercera unidad es una unidad funcional.

12. Un método de acuerdo con cualquiera de las reivindicaciones 9-11, en donde la primera unidad de memoria (230, 231) es una memoria compleja.

13. Un método de acuerdo con cualquiera de las reivindicaciones 10 ó 12 cuando depende de la reivindicación 10, que comprende el paso de controlar la lectura de direcciones de la unidad de memoria de direcciones por medio de una unidad de generación de direcciones dispuesta para controlar la lectura de los datos de direcciones de la segunda unidad de memoria de acuerdo con un patrón predefinido.

14. Un método de acuerdo con cualquiera de las reivindicaciones 9-13, que comprende el paso de sincronizar la información de direcciones de tal manera que la siguiente dirección a ser leída, o escrita, alcanza la primera unidad de memoria un ciclo de reloj antes de que realmente deba leerse o escribirse.

15. Un método de acuerdo con la reivindicación 14, en donde la cantidad de datos que se lee por adelantado se controla por fuentes de direcciones que introducen una cantidad predefinida de datos de direcciones sobre la red, el número de etapas de segmentación estando codificadas en el hardware.

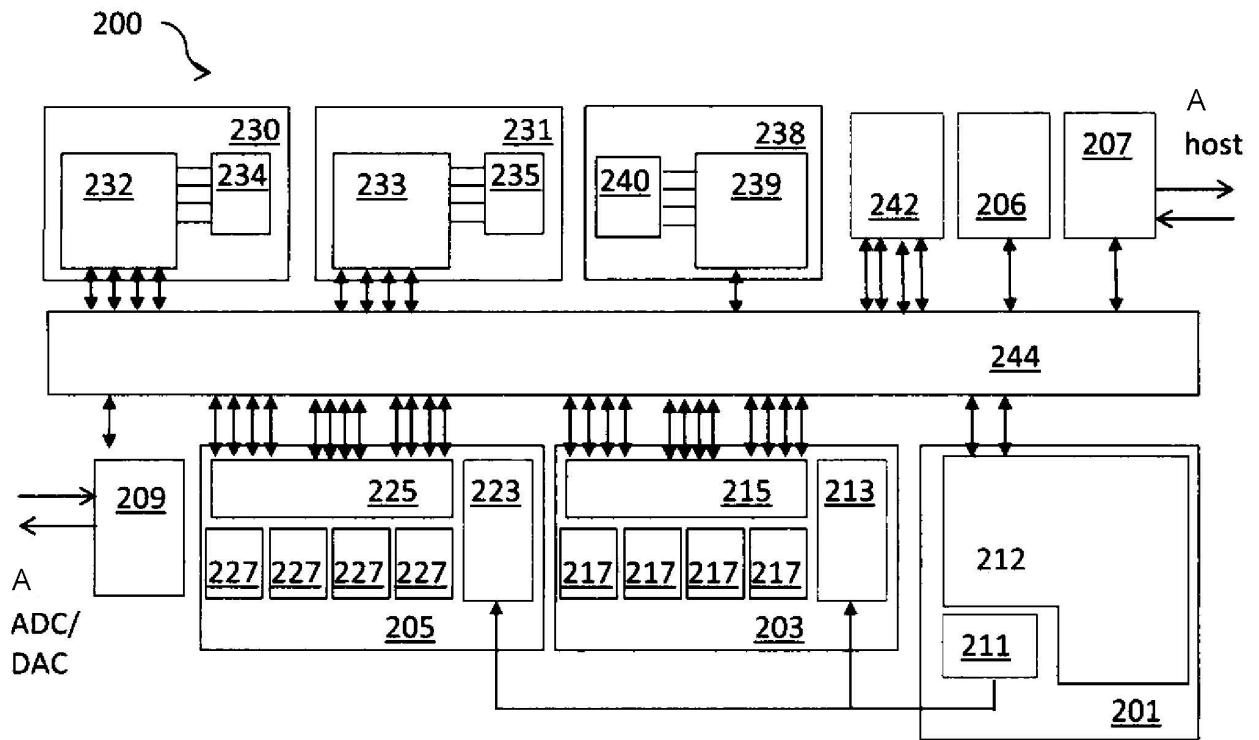


Fig. 1

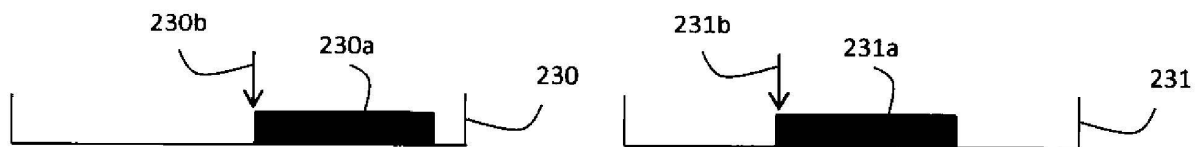
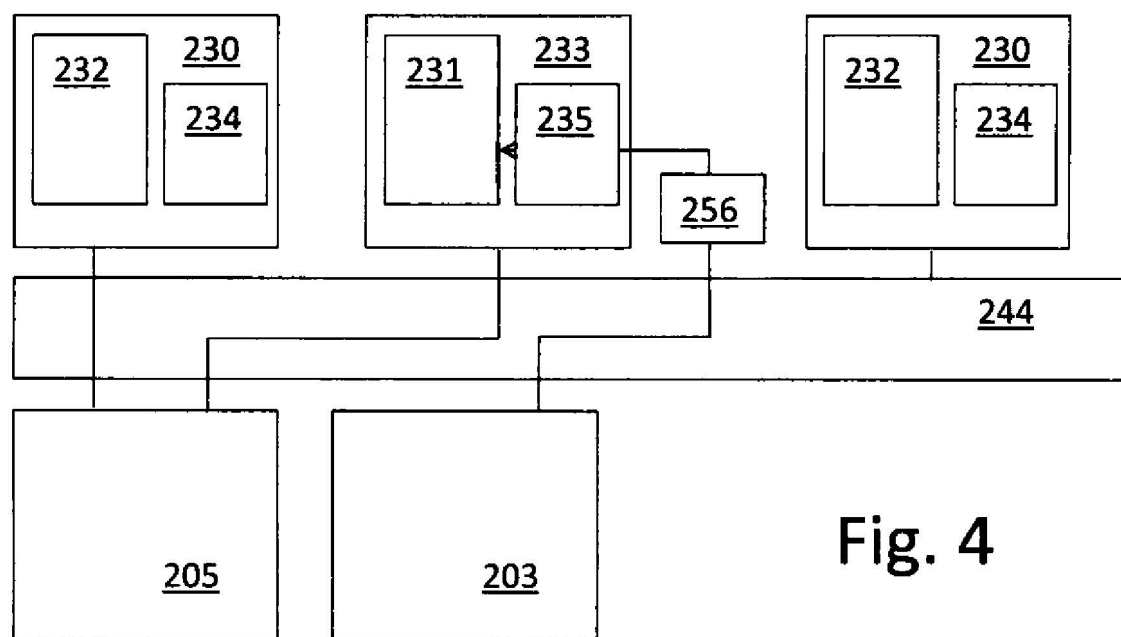
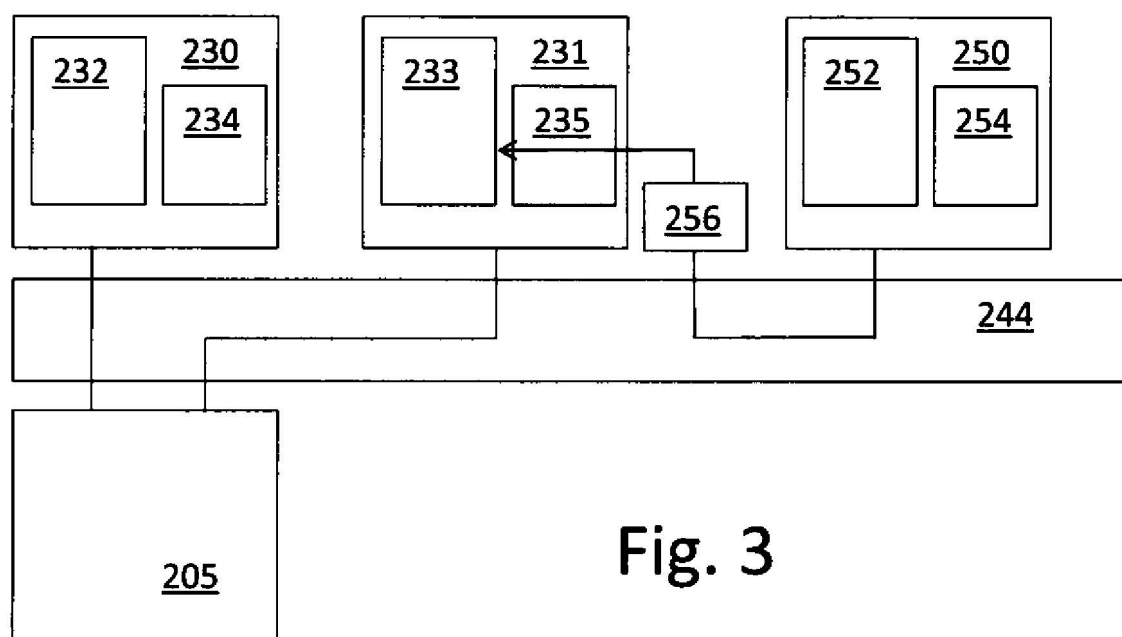


Fig. 2



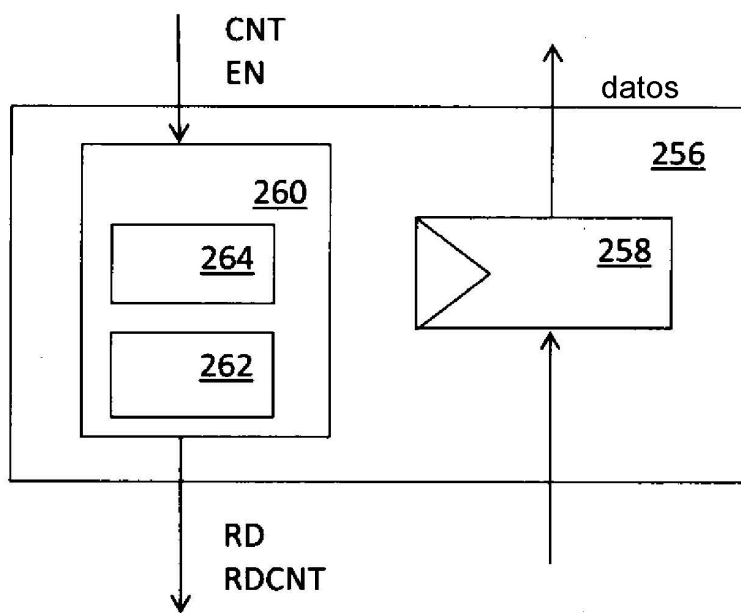


Fig. 5

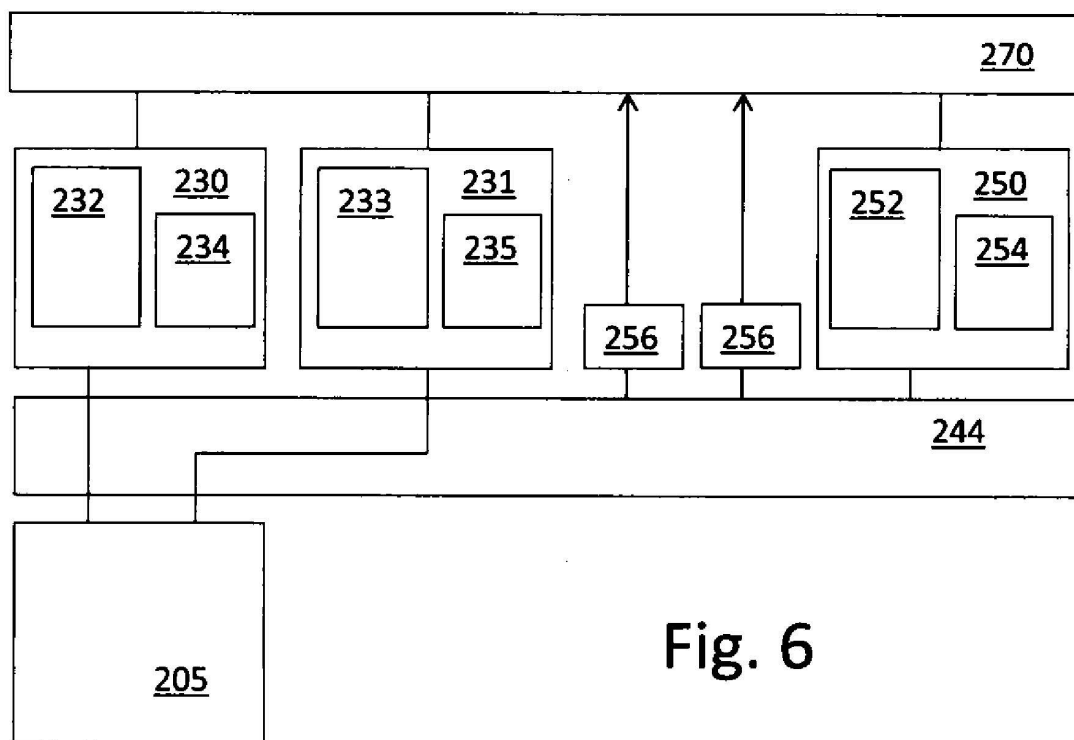


Fig. 6