(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 577 856**

(51) Int. CI.:

G06F 9/38 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **28.11.2012 E 12816375 (5)**

(97) Fecha y número de publicación de la concesión europea: **20.04.2016 EP 2751670**

(54) Título: **Procesador de señal digital**

(30) Prioridad:

21.12.2011 SE 1151233

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
19.07.2016

(73) Titular/es:

MEDIATEK SWEDEN AB (100.0%)
Teknikringen 10
583 30 Linköping, SE

(72) Inventor/es:

NILSSON, ANDERS y
TELL, ERIC

(74) Agente/Representante:

IZQUIERDO BLANCO, María Alicia

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

Procesador de señal digital

Campo técnico

[0001] La presente invención se refiere a un procesador de ensamblaje de acuerdo con el preámbulo de la reivindicación 1.

Antecedentes y técnica relacionada

[0002] Muchos de los dispositivos de comunicación móviles utilizan un transceptor de radio que incluye uno o más procesadores de señales digitales (DSP).

[0003] Para un mayor rendimiento y fiabilidad muchos terminales móviles utilizan actualmente un tipo de DSP conocido como un procesador de banda base (BBP), para el manejo de muchas de las funciones de procesamiento de señales asociadas con el procesamiento de la señal de radio recibida y la preparación de las señales para la transmisión.

[0004] Muchas de las funciones que frecuentemente se realizan con en este tipo de procesadores se llevan a cabo en un gran número de muestras de datos. Por lo tanto un tipo de procesador conocido como instrucción de Datos Múltiples de Procesador Único (SIMD) es útil debido a que permite que la misma instrucción se realice durante todo un vector de datos en lugar de en un número entero a la vez. Este tipo de procesador es capaz de procesar las instrucciones de vectores, lo que significa que una sola instrucción realiza la misma función a un número limitado de unidades de datos. Los datos se agrupan en bytes o palabras y se agrupan en un vector para ser operado.

[0005] Como un desarrollo adicional de la arquitectura SIMD, se ha desarrollado arquitectura de Tareas Múltiples de Flujo de Instrucciones Único (SIMT). Tradicionalmente en arquitectura SIMT, una o dos unidades de ejecución vectorial SIMD utilizando rutas de datos se han proporcionado en asociación con una unidad de ejecución con un número entero que puede ser parte de un procesador de núcleo.

[0006] Solicitud de Patente Internacional WO 2007/018467 divulga un DSP de acuerdo con la arquitectura SIMT, que tiene un núcleo de procesador que incluye un procesador de número entero y una memoria de programa, y dos unidades de ejecución vectoriales que están conectados al núcleo, pero no integrados en él. Las unidades de ejecución vectoriales pueden ser Unidades Lógicas Aritméticas Complejas (CALU) o Unidades de Acumulación Múltiple Complejas (CMAC). Los datos que se procesan en las unidades de ejecución vectoriales se proporcionan a partir de unidades de memoria de datos conectados a las unidades de ejecución vectoriales a través de una red de chip.

[0007] En los grandes sistemas de núcleo múltiple es difícil afectar a la partición y planificar las necesidades de recursos por delante. Para aumentar la flexibilidad, sería útil permitir que un procesador pida prestado recursos de otro procesador de señal digital. Esto puede realizarse en la técnica anterior mediante la realización de una llamada de procedimiento remoto. Esto implica la transferencia de datos a una memoria del otro procesador y solicitar la ejecución de una función por el otro procesador. Los datos resultantes debe entonces ser transferido de nuevo a una memoria del primer procesador de señal digital. Esto ocupa una cantidad considerable de la capacidad de control en el segundo procesador de señal digital, lo cual es ineficiente.

[0008] Una solución alternativa, que es común en los procesadores de señales digitales es dejar que un número de procesadores compartan una memoria que se puede acceder a todos los procesadores. La memoria puede ser una memoria de datos, una memoria de programa o una memoria de datos y programa combinados. Memorias que pueden accederse desde varios procesadores son caros y difíciles de manejar en términos de arbitraje de caché. Se vuelven impredecibles y difíciles a sincronizar.

[0009] US 5 010 477 A describe un conjunto de procesador de acuerdo con el preámbulo de la reivindicación 1, en la que en una forma de realización (figura 8), cada elemento procesador divulga su propio núcleo. De acuerdo con las enseñanzas de la figura 8, cada núcleo obtiene, decodifica y controla la ejecución de las instrucciones de su propia memoria local. Además, cada núcleo puede ejecutar una instrucción de vector-recibir, que permita un segundo núcleo en particular para el control de su unidad de memoria, ya que permite un segundo núcleo específico para escribir en el mismo.

Resumen de la invención

[0010] Por lo tanto es un objetivo de la presente invención permitir un aumento de la capacidad de un DSP. Esto se consigue según la invención mediante un procesador de señal digital tal como se define anteriormente, que tiene las características de la parte caracterizadora de la reivindicación 1.

[0011] Según la invención, dos procesadores de señales digitales se conectan por medio de un puente. Cada uno de los procesadores comprende un núcleo, y una o más otras unidades conectadas por una red de chip. Las

unidades pueden ser unidades complejas de computación, unidades de computación escalares, unidades de memoria, aceleradores y cualquier otra unidad que se puede utilizar en un DSP basado SIMT.

[0012] Por medio del puente, el primer DSP puede acceder todos a los recursos en el segundo DSP, y viceversa. Si el primer procesador para utilizar un elemento, tal como una unidad de ejecución o un acelerador, o acceder a una memoria, que se encuentra en el segundo procesador, esto se puede hacer a través del puente, cuando el primer procesador que temporalmente tome control del elemento. Lógica de cuestión de instrucciones dispuestas en el segundo procesador renuncia temporalmente el control del elemento de la lógica cuestión dispuesta en el primer procesador. Los datos se envían directamente desde la memoria de programa de la primera DSP de la unidad de ejecución de la segunda DSP que es para llevar a cabo la instrucción.

[0013] La invención permite el uso compartido de unidades ejecutivas vectoriales, aceleradores, memorias y otras unidades entre dos procesadores de señales digitales. Esto se puede utilizar para aumentar temporalmente la capacidad de uno de los procesadores. En particular, por ejemplo, un tipo de unidad que se necesita por ambas unidades, pero no se utiliza mucho, se puede organizar sólo en uno de los procesadores y de fácil acceso para ambos. Además, en lugar de almacenar los mismos datos en ambos procesadores, sólo necesitan ser almacenados en una memoria en uno de los procesadores y se puede acceder fácilmente por ambas procesadores.

Breve descripción de los dibujos

[0014]

La figura 1 muestra un DSP tipo SIMT que puede utilizarse de acuerdo con la invención
La Figura 2 muestra dos DSPs tipo SIMT interconectadas de acuerdo con una realización de la invención.
La Figura 3 muestra un núcleo a la unidad de interfaz de núcleo de acuerdo con una realización de la invención.
La Figura 4 muestra una realización de la lógica cuestión aplicada al lado remoto, visto desde el lado local.
La figura 5 muestra una realización de la lógica cuestión visto desde el lado remoto.

Descripción detallada de realizaciones

[0015] La figura 1 muestra un ejemplo de un procesador de señal digital de la técnica anterior 200 de acuerdo con la arquitectura SIMT, que pueden formar la base para la invención. El procesador 200 incluye un núcleo controlador 201 y una unidad de ejecución vectorial primera 203 y una segunda 205, conocido per se.

[0016] Una unidad de interfaz de huesped 207 proporciona la conexión a un procesador central (no mostrado en la Fig. 2), posiblemente a través de una unidad tal como un procesador de MAC. Una unidad frontal digital 209 proporciona la conexión a las unidades de transmisión y receptora de señales (no mostradas) de una manera bien conocida en la técnica.

[0017] Como es común en la técnica, el núcleo del controlador 201 comprende una memoria de programa 211, así como la lógica cuestión de instrucciones y funciones para el apoyo multi-contexto. Para cada contexto de ejecución, o de hilos, apoyado esto incluye un contador de programa, puntero de pila y el registro de archivos (no se muestra explícitamente en la figura 1). Por lo general, 2-3 hilos son compatibles.

[0018] El núcleo del controlador 201 también comprende una unidad de ejecución de números enteros 212 que comprende un archivo de registro RF, una memoria ICM de núcleo de números enteros, una unidad de multiplicador MUL y una Unidad Aritmética y Lógica/Cambio (USVA). El USVA también puede implementarse como dos unidades, la unidad aritmético-lógica y Unidad de Cambio. Estas unidades son conocidas en la técnica y no se muestran en la Figura 1.

[0019] La primera unidad de ejecución vectorial 203 en este ejemplo es una unidad de ejecución vectorial CMAC, y la unidad de ejecución segunda vectorial 205 es una unidad de ejecución de vectores CALU. Ambas unidades de ejecución vectoriales son conocidas per se. Cada unidad de ejecución vectorial 203, 205 comprende un controlador vectorial 213, 223, una unidad vectorial de carga/almacen 215, 225 y un número de rutas de datos 217, 227, respectivamente. Los controladores de vectores 213, 223 de las unidades de ejecución vectoriales están conectados a la memoria del programa 211 del núcleo del controlador 201 a través de la lógica cuestión, para recibir señales de emisión relacionadas con las instrucciones de la memoria del programa.

[0020] En la descripción anterior, la lógica cuestión decodifica la palabra de instrucción para obtener la señal de emisión y envía esta señal de cuestión de la unidad de ejecución vectorial como una señal separada. También sería posible hacer que el controlador vectorialial de la unidad de ejecución vectorial generase la señal cuestión localmente. En este caso, las señales de emisión son creadas por el controlador vectorial basado en la palabra de instrucción de la misma manera como lo sería en la lógica cuestión.

[0021] Las funciones de las rutas de datos 217, 227 y las unidades vectoriales carga/almacenamiento 215, 225 se conocen en la técnica.

[0022] Puede haber un número arbitrario de unidades de ejecución vectoriales, incluyendo sólo las unidades CMAC, sólo las unidades Calu o un número adecuado de cada tipo. También puede haber otros tipos de unidad de ejecución vectorial de CMAC y CALU. Como se explicó anteriormente, una unidad de ejecución vectorial es un procesador que es capaz de procesar las instrucciones vectoriales, lo que significa que una sola instrucción realiza la misma función a un número de unidades de datos. Los datos pueden ser complejos o reales, y se agrupan en bytes o palabras y se envasa en un vectorial que se opera por una unidad de ejecución vectorial. En este documento, las unidades Calu y CMAC se utilizan como ejemplos, pero debe tenerse en cuenta que las unidades de ejecución vectoriales se pueden utilizar para realizar cualquier función adecuada en los vectores de datos.

[0023] Como es común en la técnica, el PBBP de la fig. 2 también puede incluir uno o más bancos de memoria enteros 238, cada uno que incluye una memoria 239 y una unidad de generación de dirección 240.

[0024] Normalmente, pero no necesariamente, un número de aceleradores 242 están conectados, ya que permiten la aplicación eficaz de ciertas funciones de banda base, como la codificación de canal y el intercalado. Tales aceleradores son bien conocidos en la técnica y no se discutirán en detalle aquí. Los aceleradores pueden ser configurables para ser reutilizados por muchas normas diferentes.

[0025] Una red de chip 244 conecta el núcleo del controlador 201, la unidad frontal digital 209, la unidad de interfaz de host 207, las unidades de ejecución vectoriales 203, 205, los bancos de memoria 230, 232, el banco número entero 238 y los aceleradores 242.

[0026] Las unidades de ejecución vectoriales 203, 205 están conectadas a las otras unidades del procesador, tales como los bancos de memoria 230, 231, 242 aceleradores a través de la red 244.

[0027] En una realización, la arquitectura del conjunto de instrucciones de núcleo de procesador 201 puede incluir tres tipos de instrucciones compuestas. La primera clase de instrucciones son instrucciones RISC, que operan sobre operandos enteros. La clase RISC-instrucción incluye la mayor parte de las instrucciones específicas de control y puede ser ejecutado dentro de la unidad de ejecución de número entero 212 del núcleo del procesador 201. La siguiente clase de instrucciones son instrucciones DSP, que operan en los datos de valores complejos que tienen una parte real y una parte imaginaria. Las instrucciones DSP pueden ser ejecutadas en una o más de las unidades de ejecución vectoriales 203, 205. La tercera clase de instrucciones son las instrucciones vectoriales. Las instrucciones vectoriales pueden considerarse extensiones de las instrucciones DSP ya que funcionan con grandes conjuntos de datos y pueden utilizar los modos de direccionamiento avanzado y vectoriales de soporte. Las instrucciones vectoriales pueden operar sobre los tipos de datos complejos o reales.

[0028] La figura 2 muestra un conjunto de procesadores que comprende un primer y un segundo procesador que están interconectados de acuerdo con la invención. Cada uno de los procesadores puede ser un DSP como se discutió en relación con la figura 1. En el primer procesador de la figura 2 los mismos numerales de referencia se utilizan para los mismos tipos de unidades. En el segundo procesador los mismos números de referencia con un símbolo primo. Por lo tanto, el primer procesador comprende un núcleo 201 y dos unidades de ejecución 203, 205 y el segundo procesador comprende un núcleo 201' y dos unidades de ejecución 203', 205', etc.

[0029] Según la invención, las unidades de control del primer procesador están dispuestas para permitir que las funciones de control del segundo procesador influyan en unidades en el primer procesador, y viceversa. Las unidades de control del primer procesador también están dispuestas para influir en las unidades en el segundo procesador cuando las funciones de control del segundo procesador han permitido esto, y viceversa. Hay diferentes maneras en que las unidades en un solo procesador pueden ser modificados mediante unidades de control en el otro procesador. La Figura 2 ilustra varias de estas formas, que pueden ser im- complementadas independientemente la una de la otra.

[0030] Entre las unidades de control del primer y segundo procesador está conectado a un núcleo de unidad de interfaz central 250, para permitir que los núcleos 201, 201' de los procesadores para intercambiar mensajes entre sí. La unidad de interfaz núcleo a núcleo se discutirá con más detalle en relación con la figura 3.

[0031] Una forma de influir en las unidades de la otra procesador se ilustra por medio de una primera y una segunda unidad de puente 252, 252', que pertenece al primer y segundo procesador, respectivamente. Las unidades de puente 252, 252' están conectados entre sí, y están conectados a la red de chip 244, 244' del primer y segundo procesador, respectivamente. Por lo tanto, las dos unidades de puente juntas interconectan las redes de chip 244, 244' de los dos procesadores. A través de las unidades de puente, los dos procesadores pueden acceder a las unidades de memoria de cada uno 230, 231. Esto significa que a través de las unidades de puente una unidad de ejecución vectorial en el primer procesador puede acceder a datos de la memoria en una unidad de memoria que pertenece al segundo procesador.

[0032] Normalmente, en este modo, se denomina modo de puente, las unidades de puente 252, 252' estarán conectadas en una configuración maestro-esclavo. Preferiblemente, cada unidad de puente puede funcionar tanto

como un maestro como un esclavo. Los datos pueden ser leídos y escritos en ambas direcciones en una configuración maestro-esclavo. Por ejemplo, una unidad de ejecución 203, 205 en el primer procesador de solicitud de datos desde una memoria 230', 231' en el segundo procesador, o escribiendo datos en esta memoria, funcionará como un maestro a la unidad de puente 252 en el lado del primer procesador. La unidad de puente 252' en el lado del segundo procesador funcionará como un maestro a la unidad de memoria, que funcionará como un esclavo. La unidad de puente también puede implementarse como una unidad.

[0033] Todo lo escrito en el puerto de puente en el lado principal será remitido al lado de esclavo y la unidad esclava conectada. Los datos serán retrasados por unos ciclos en la tubería de puente antes de escribirse en la unidad esclava en la red remota.

[0034] También es posible transferir datos de una unidad memoria en el primer procesador a una unidad de memoria en el segundo procesador y viceversa a través de los puentes 252, 252'. En este modo, llamado modo de DMA, cada unidad de puente 252, 252' funcionará como un maestro en su respectivo conmutador de barras cruzadas 244, 244'.

[0035] Como se ha mencionado en relación con la figura 1, cada núcleo 201, 201' comprende una memoria de programa 211, 211', respectivamente, y una unidad de ejecución de número entero 212, 212', respectivamente. Una unidad lógica de emisión 254, 254' está dispuesta en cada núcleo, entre la memoria de programa 211, 211' y las unidades de ejecución 212, 212'. Como se indica en la Figura 2 existe una conexión remota 260 de la unidad lógica de emisión 254 del primer procesador al segundo procesador. Esta conexión a distancia 260 está dispuesta para enviar las instrucciones del primer procesador para ser ejecutado por las unidades en el segundo procesador. Esto significa que una instrucción que tuvo lugar en la memoria de programa 211 del primer procesador puede ser enviado directamente a la unidad funcional deseada en el segundo procesador, a distancia, el sin tener que ser almacenado en la memoria de programa 211' del segundo procesador. La conexión de la conexión remota 260 a las unidades en el lado alejado se discutirá con más detalle en relación con la figura 5. Para la gestión del mando a distancia son necesarias algunas funciones adicionales en relación a la unidad de control 254. Éstas no se muestran por separado en la Figura 2 pero se discutirán en relación con las figuras 4A y 4B. Normalmente, habrá una conexión similar de la unidad lógica de emisión del segundo procesador para las unidades del primer procesador. Esta conexión no se muestra en el dibujo, por razones de claridad.

[0036] La Figura 3 muestra un núcleo a la unidad de interfaz central 250 de acuerdo con una realización de la invención. La unidad de interfaz comprende una unidad de control de la interfaz 303 y una primera y una segunda cola 305, 307 para la transferencia de datos desde el primer procesador al segundo procesador y viceversa, respectivamente. De este modo los mensajes de control pueden ser escritos por un procesador de registros, tales como registros de direcciones en el otro procesador. La unidad de control de la interfaz 303 comprende uno o más registros de control y uno o más registros de datos. Los datos que se escriben en la primera cola de la base del primer procesador puede ser leído por el núcleo del segundo procesador, que es en este caso el núcleo central. Los datos que se escriben en la segunda cola desde el núcleo del segundo procesador pueden ser leídos por el núcleo del primer procesador, que es en este caso el núcleo remoto. Ambas operaciones son realizadas por el núcleo remoto que lee el registro de datos. El registro de control tiene bits de estado para indicar el estado de registros de datos, por ejemplo si están llenos o vacíos. Las colas son preferiblemente colas FIFO.

[0037] Un bus CRF 309 (registro de control del archivo de bus) se conecta tanto con el registro de control 303 como las colas 305, 307 al núcleo del procesador respectivo y para todas las unidades del procesador. Como se indica por la línea 311, el bus de CRF 309 también está conectado desde el segundo lado de la unidad de interfaz 250 para el núcleo del primer procesador. Hay una conexión similar (que no se muestra para mayor claridad) desde el primer lado de la unidad de interfaz 250 para el núcleo del segundo procesador.

[0038] Los datos que se transfieren entre los núcleos se organizan generalmente en paquetes de datos. El tamaño de un paquete puede ser arbitrariamente decidido por el usuario.

[0039] La unidad de interfaz también puede ser utilizada por el primer núcleo para generar una interrupción en el segundo núcleo y viceversa. La unidad de interfaz mantiene un contador que cuenta el número de paquetes de datos pendientes que aún no han sido leído por el núcleo remoto. El contador se incrementa cuando un paquete se escribe en el registro de datos y se disminuye cuando un paquete se lee desde el registro de datos. Cada vez que el contador va de 0 a 1 o se disminuye, pero todavía mayor que 0 se generará una interrupción. Las señales de interrupción pueden ser transferidas desde el primer núcleo a la segunda base y viceversa en una conexión 308.

[0040] La Figura 4A ilustra las funciones necesarias en el núcleo del primer procesador para el núcleo remoto de las unidades en el segundo procesador. El núcleo comprende una memoria de programa) 211 que contiene instrucciones para unidades funcionales, tanto en el primero como en el segundo DSP. Una unidad de pre-decodificación 321 está dispuesta para determinar qué unidad funcional debe recibir cada instrucción que se lee de la memoria de programa. Cuando las instrucciones se destinan a ser enviadas a la DSP remota, es decir, en este ejemplo, la segunda DSP, se envían a través de la conexión remota se muestra en la Figura 2. Esta conexión remota en realidad lleva un conjunto de señales como se discutirá en relación con la Figura 5, que representa el

manejo de las señales en el lado remoto. La palabra de instrucción se envía directamente desde la memoria de programa 211 a las unidades remotas, como se muestra por la flecha denotada instrucción remota en la figura 4A. La señal de emisión, que lleva la información acerca de qué unidad funcional debe realizar la instrucción, se envía a través de un demultiplexor) 323.

[0041] En el núcleo local, una unidad de máscara 325 recibe la señal de emisión y determina qué unidad o unidades funcionales deben recibir la instrucción. Esta unidad funcional puede ser una unidad funcional en el primer procesador, una unidad funcional en el segundo procesador, o un grupo de unidades funcionales que comprende unidades funcionales a partir de uno de los procesadores o de los dos procesadores. La función de la unidad de máscara 325 se discutirá con más detalle en lo siguiente. Cuando la unidad de máscara ha determinado, basándose en la información de control, que la unidad funcional o unidades deben realizar la instrucción, una señal de emisión se envía a estas unidades funcionales. Una señal cuestión separada se envía para cada unidad funcional, como se muestra por las flechas denotadaron señal de emisión 1 y la señal de emisión n en la Figura 4A.

[0042] De acuerdo con la realización mostrada en la Figura 4, esto se consigue de la siguiente manera. La unidad de máscara 325 comprende un registro de 326 que contiene información acerca de qué unidad funcional o unidades corresponden a la señal de emisión recibids desde el demultiplexor. Esta información se reenvía desde la unidad de máscara para el núcleo de la unidad de interfaz central. La unidad funcional real o unidades que reciben la instrucción se puede obtener en unidades de operaciones lógicas 327 que realiza una operación lógica de la señal de emisión y la información contenida en la unidad de máscara. Hay una unidad de operaciones lógicas para cada unidad funcional posible en un grupo cuestión.

[0043] La Figura 4B muestra un ejemplo de registro 326 de acuerdo con la realización anterior. La unidad de filtro comprende una máscara de identificación de las unidades de ejecución vectoriales en un grupo de unidades de ejecución vectoriales que en realidad debería recibir la instrucción. En la práctica, la máscara tiene un bit para cada unidad de ejecución vectorial, que puede ser ajustado a 0 ó 1, para indicar si la unidad de ejecución vectorial debe ser incluida en el grupo cuestión o no. Esta información es combinada en la unidad de operaciones lógicas correspondiente 327 con la información contenida en la señal de emisión para determinar qué unidades de ejecución vectoriales van a recibir la instrucción. En el ejemplo de la Figura 4B, nueve unidades de ejecución vectoriales están potencialmente incluidas en el grupo cuestión. La información almacenada en la unidad de máscara indica que la primera y la última de estas unidades de ejecución en realidad debería participar en la ejecución de la instrucción.

[0044] En lugar de utilizar una unidad de máscara como se discute en relación con las figuras 4A y 4B que sería posible tener todas las unidades funcionales codificadas en el núcleo. La unidad de máscara proporciona una mayor flexibilidad para el procesador al permitir la adición y la eliminación de unidades funcionales que pueden ser controladas por un procesador particular. Alternativamente, la selección de las unidades de ejecución vectoriales podría ser hecha por la unidad de máscara sin el uso de una señal de emisión. Esto, sin embargo, sería menos eficiente.

[0045] Fig. 5 ilustra el manejo de las instrucciones remotas en el lado receptor, es decir, en este ejemplo, en el segundo procesador. Para simplificar, una primera unidad funcional 501 y una segunda 503 se muestran en la figura. Para habilitar las unidades funcionales 501, 503 para recibir instrucciones de más de un núcleo de cada unidad funcional está equipada con un primer multiplexor 505, 506, respectivamente, que está dispuesto para recibir palabras de instrucción a partir de dos o más fuentes y alimentar la señal apropiada a la unidad funcional. Cada unidad funcional 501, 503 también está equipada con un segundo multiplexor 507, 508, respectivamente, que está dispuesto para recibir señales de emisión a partir de dos o más fuentes y alimentar la señal apropiada a la unidad de función.

[0046] Las instrucciones locales, procedentes del segunda núcleo en este caso, se reciben en una conexión marcada LOCAL. Las señales de emisión locales se reciben en las conexiones marcadas EU0 y UE1, respectivamente, muestran sólo esquemáticamente.

[0047] La comunicación con la unidad de núcleo remoto se muestra como un conjunto de cinco líneas, que corresponde a la conexión remota se muestra en la Figura 2. Como se puede ver por la más superior de las líneas en el paquete, marcada INSTR, la palabra de instrucción remota se transmite a las unidades funcionales. Como de costumbre, la unidad funcional o unidades indicadas en la señal de emisión deberá procesar la operación. Las dos líneas siguientes, marcadas colectivamente emisión se emplearán para transmitir la señal de emisión a la primera y la segunda unidad funcional, respectivamente. Las dos líneas inferiores, colectivamente marcadas READY en el conjunto se utilizan para llevar señales de preparado de la primera y la segunda unidad de la ejecución, respectivamente, de vuelta a la base del primer procesador. Una señal de preparado puede ser enviado por una unidad de ejecución para informar el núcleo que la unidad de ejecución está lista para volver a recibir otra instrucción y puede iniciar una interrupción de señal.

[0048] En el ejemplo mostrado en las figuras 4 y 5 se supone que hay dos procesadores interconectados de modo que cada unidad funcional puede ser controlado ya sea por su núcleo local o de forma remota desde el otro núcleo. Como el experto en la materia se daría cuenta, más de un procesador remoto podría estar conectado. En la figura 5

un procesador remoto adicional se ilustra por otro conjunto de conexiones, la conexión de cada unidad funcional al procesador adicional de la misma manera como se muestra para el procesador remoto en la Figura 5. Esto significa que las palabras de instrucciones y señales de emisión se transmite a las unidades funcionales a través de los multiplexores y las señales de listo se transmitiría a partir de las unidades de ejecución a las unidades de núcleo remoto de los procesadores remotos.

[0049] En lugar de enviar una palabra de instrucción y una señal de aspecto por separado, sería posible enviar una palabra de instrucción que comprende la señal de emisión y dejar que la unidad funcional que la reciba recupere la señal de emisión para determinar si una palabra de instrucción particular fue diseñada para esta unidad funcional.

[0050] También sería posible hacer frente a más de una unidad funcional a la vez, dividiendo las unidades funcionales en grupos temáticos. Solicitud co-pendiente Nº. SE 1151231-6 divulga el uso de grupos de emisión dentro de un tipo DSP SIMT. Los grupos de asuntos son grupos de unidades de ejecución que están configurados para llevar a cabo la misma instrucción vectorial de diferentes datos. Los grupos de asuntos se pueden especificar de una manera similar como el direccionamiento de las unidades funcionales que se discuten en relación con las figuras 4A y 4B. Para hacer frente a un grupo de tema, la unidad de máscara mostrada en la figura 4B debe especificar más de una unidad funcional.

[0051] En el contexto de la presente invención, sería posible dejar que los primeros grupos cuestión de acceso del procesador en la segunda DSP, de la misma manera, ya que aborda las unidades de ejecución vectoriales en la segunda DSP, y viceversa. También sería posible definir grupos de emisión que comprenden unidades de ejecución vectoriales de ambas DSP.

[0052] Tal como se comprenderá, no todas las unidades descritas son necesarias en ambos procesadores con el fin de lograr las funciones deseadas. En el caso más simple, sólo el procesador remoto tendría que tener unidades de memoria o unidades funcionales. Las unidades funcionales podrían ser sólo unidades de ejecución, sólo las unidades de aceleración, o una combinación de ambas.

Reivindicaciones

1. Un conjunto de procesadores, que comprende un primer procesador que tiene un primer núcleo (201) que comprende una memoria de programa (211), la lógica cuestión de instrucciones, y una unidad de ejecuciones de números enteros (212), comprendiendo el procesador ensamblaje además al menos una primera unidad de ejecución vectorial (203, 205) y una unidad de primera memoria (230, 231) dispuesta para almacenar los datos que hay que operar, dicho primer núcleo, primera unidad de ejecución vectorial y la primera unidad de memoria que se está interconectando por una primera red (244), y un segundo procesador que tiene un segundo núcleo (201'), en el que el primer núcleo (201) está dispuesto para permitir que el segundo núcleo controle al menos la primera unidad de ejecución vectorial (203, 205) o de la primera unidad de memoria (230, 231) en el primer procesador, **caracterizado porque** dicho programa de memoria está dispuesto para contener las instrucciones para las unidades funcionales en el primero y el segundo procesador.

2. Un conjunto de procesador de acuerdo con la reivindicación 1, que comprende además una unidad de interfaz de núcleo a núcleo (250) que tiene al menos una primera cola (303) dispuesta para almacenar los mensajes del primer núcleo (201) de tal manera que son recuperables por el segundo núcleo (201') y/o viceversa.

3. Un conjunto de procesador de acuerdo con la reivindicación 1 o 2, en el que la unidad de interfaz de núcleo a núcleo es implementada en el sentido de tener un lado local y un lado remoto.

4. Un conjunto de procesador de acuerdo con una cualquiera de las reivindicaciones anteriores, en el que al menos una de las unidades primera y segunda (203, 205, 242) es una unidad funcional y el segundo núcleo (201') comprende una segunda unidad lógica (254') que dispuesta para transmitir una instrucción a dicha unidad funcional.

5. Un conjunto de procesador de acuerdo con la reivindicación 4, en el que la segunda unidad lógica de emisión (254') está dispuesta para transformar una palabra de instrucción y una señal de emisión a dicha unidad funcional (203, 205, 242).

6. Un procesador de acuerdo con una cualquiera de las reivindicaciones anteriores, en el que el segundo procesador comprende al menos una tercera y una cuarta unidad (203', 205', 230', 231', 242'), cada uno se selecciona del grupo de unidades de ejecución vectoriales, unidades de memoria y los aceleradores, dicho segundo núcleo (201') y las unidades tercera y cuarta están interconectadas por una segunda red (244').

7. Un conjunto de procesador de acuerdo con la reivindicación 6, que comprende además una unidad de puente (252, 252') que interconecta la red de bus (244) y la segunda red (244'), dicha unidad de puente (252, 252') estando dispuesto para la transferencia de datos entre al menos una de las unidades primera y segunda (203, 205, 230, 231, 242) en el primer procesador de al menos una de la unidades tercera y cuarta (203', 205', 230', 231', 242') en el segundo procesador.

8. Un conjunto de procesador de acuerdo con una cualquiera de las reivindicaciones anteriores, en el que la primera unidad de ejecución vectorial (501) tiene un primer multiplexor dispuesto para recibir instrucciones del núcleo primero y segundo, respectivamente, y alimentar la señal apropiada a la unidad de ejecución vectorial.

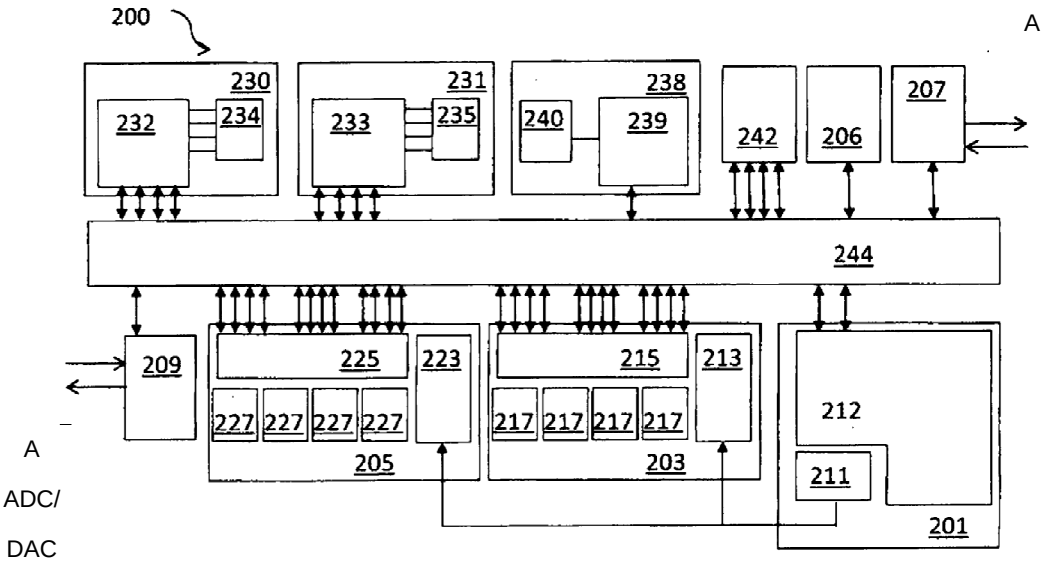
200
A acoger
230
231
238
232
234
233
235
240
239
242
206
207
244
209
225
223
215
213
227
227
227
227
217
217
217
217
212
205
203
211
201
A
ADC/
DAC

Fig. 1

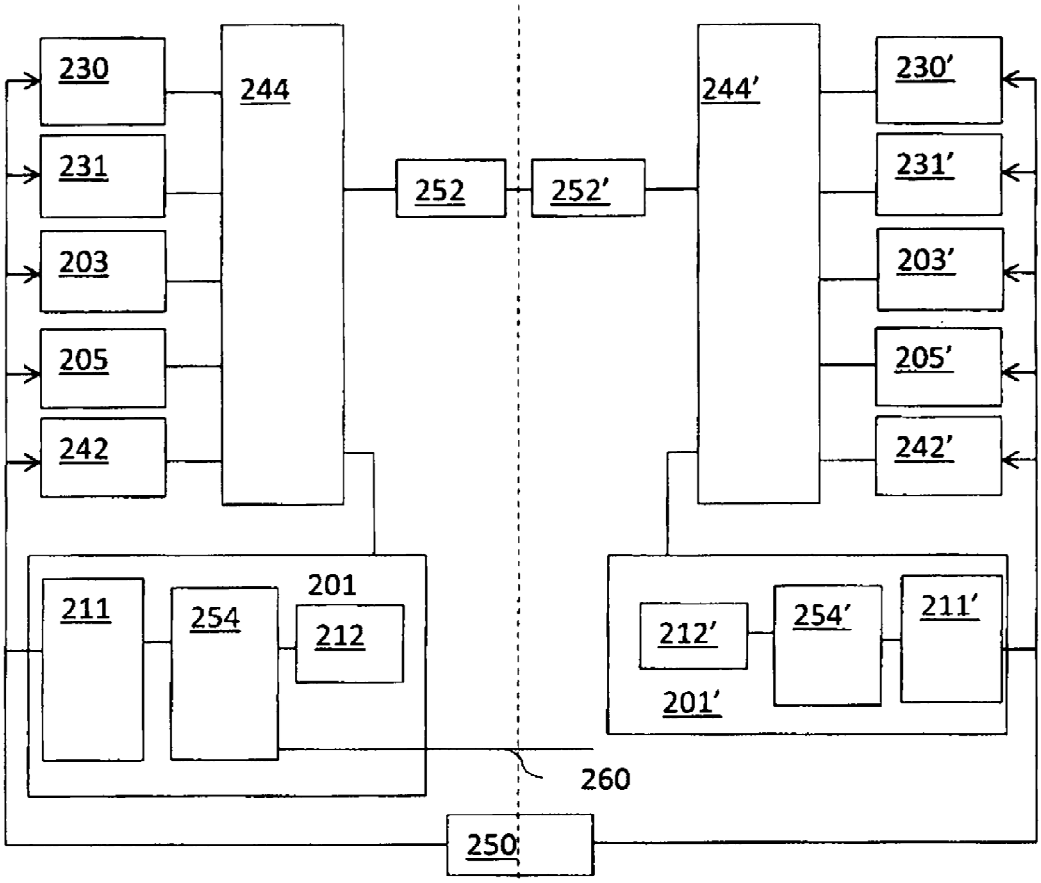
230
231
203
205
242
244
252
252'
244'
230'
231'
203'
205'
242'
201
211
254
212
212'
254'
211'
201'
260
250

Fig. 2

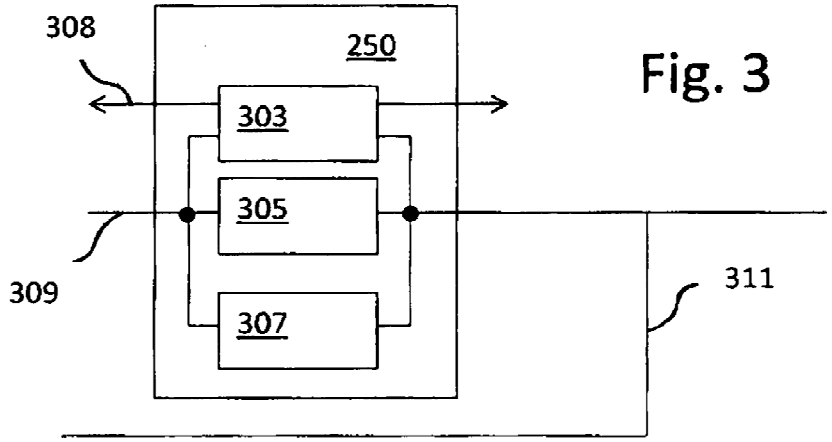
308
250
303
305
307
309
311

Fig. 3

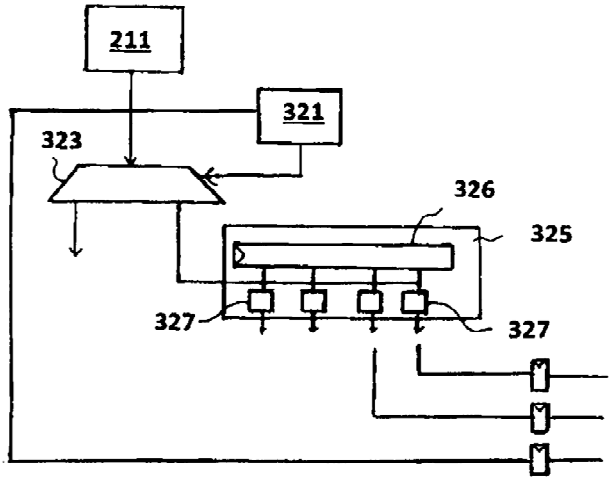

Fig. 4A

1	0	0	0	0	0	0	0	1

Fig. 4B

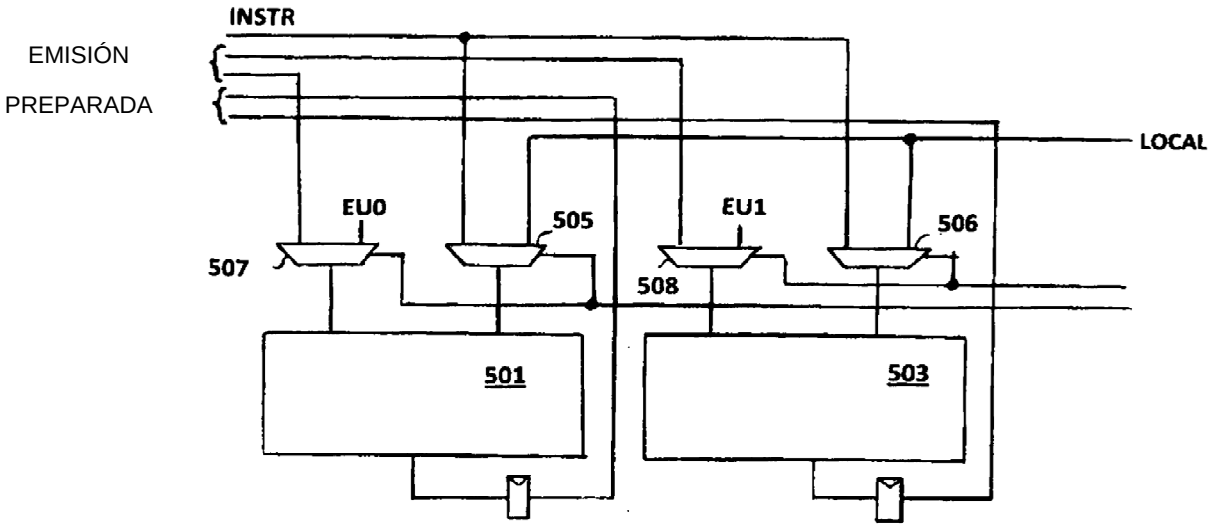

Fig. 5