

REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

Oficio N° 3023 de 2 de marzo de 2020

Ref. Expediente N° NC2018/0008835

Señor(a)
ICAT LLC

TÍTULO DE LA SOLICITUD: “PROCESADOR CON NÚCLEO CANALIZADO ALGORITMICO RECONFIGURABLE Y COMPILADOR CANALIZADO DE CORRELACION ALGORITMICA”

NÚMERO DE SOLICITUD INTERNACIONAL: PCT/US2017/015143

FECHA DE PRESENTACIÓN INTERNACIONAL: 26 de enero de 2017.

PRIORIDAD: 26/01/2016, US (ESTADOS UNIDOS DE AMÉRICA), 62/287,265

Como resultado del examen de patentabilidad realizado a la solicitud indicada en la referencia, atendiendo a lo consagrado en el artículo 45 de la Decisión 486 de 2000, se le comunica que esta Dirección ha encontrado que la materia objeto de solicitud no cumple con algunos requisitos establecidos en la Decisión, con fundamento en los siguientes aspectos:

1. Objeto de la invención

De acuerdo con lo establecido en el numeral 1.2.2.2 del Capítulo Primero del Título X de la Circular Única para efectos del examen de patentabilidad, cuando la solicitud contenga más de diez (10) reivindicaciones y no se haya pagado la tasa complementaria, sólo se estudiarán las cubiertas por la tasa pagada al momento de realizar el examen de patentabilidad consagrado en el artículo 45 de la Decisión 486. Por lo tanto, se estudiarán las reivindicaciones 1 a 23 del radicado N° NC2018/0008835 del 24 de agosto de 2018.

El problema planteado en esta solicitud consiste en la necesidad de eliminar casi toda la sobrecarga generada por los sistemas de procesamiento convencionales, mientras que se mantiene la flexibilidad de procesar una amplia gama de algoritmos y el uso de un lenguaje de programación de nivel superior estándar, como “C”, para el desarrollo de software para el sistema de procesamiento.

Para resolver este problema técnico, la solicitud en estudio proporciona un procesador paralelo canalizado en un chip que comprende una unidad de procesamiento y un arreglo de puertas reconfigurables, programables por campo, programadas por un compilador canalizado de correlación algorítmica, que puede ser un precompilador, de modo que el compilador canalizado de correlación algorítmica precompila el código fuente diseñado para funcionar en un procesador estándar sin procesamiento paralelo para el procesamiento por parte de la unidad de procesamiento, y la unidad de procesamiento y el compilador canalizado de correlación algorítmica (denominado AMPC o ASML) configuran las puertas programables de campo para operar como procesadores paralelos canalizados. Por ejemplo, el procesador se puede referir como un núcleo canalizado algorítmico reutilizable (RAPC). Los procesadores paralelos están configurados para completar tareas sin más sobrecarga desde la unidad de procesamiento, como la sobrecarga para controlar una unidad de procesamiento aritmético.



Oficio N° 3023 de 2 de marzo de 2020

Ref. Expediente N° NC2018/0008835

2. De la solicitud de patente

2.1. Reivindicaciones (Art. 30 Decisión 486)

- El sistema de las reivindicaciones 11 a 13 está definido como: “(...) *que comprende los pasos de (...)*” (reiv. 11), “(...) *que comprende además la etapa de (...)*” (reiv. 12) y “(...) *en el que la etapa de (...)*” (reiv. 13). Las reivindicaciones deben caracterizar un producto por su forma o constitución y las partes que lo componen, y el procedimiento por la secuencia de etapas técnicas que inciden sobre una materia que se encaminan a la obtención de una cosa o resultado. Se sugiere reescribir estas reivindicaciones mostrando las partes estructurales que conforman el sistema.
- El capítulo reivindicatorio no es claro, ya que presenta dos reivindicaciones enumeradas igualmente (reivs. 5). Se sugiere reenumerar el capítulo en orden secuencial.
- El término: “*un compilador canalizado de correlación algorítmica*” (reivs. 1, 2, 14, 16 y 17), no es conocido en el campo técnico. Se sugiere al solicitante definir dicho término.

2.2. Descripción (Art. 28 Decisión 486 literales (a) – (f))

- El capítulo descriptivo presenta errores de ortografía y/o redacción: “(...) *proporciona un hardware configurable de nuevo configurable para (...)*” (pág. 39, párr. 0070, línea 16), “(...) *AMPC genera el firmware de configuración de configuración utilizado para configurar los algoritmos de procesamiento (...)*” (pág. 45, párr. 0079, líneas 21 a 23). Se sugiere realizar las modificaciones necesarias.

3. Determinación del Estado de la Técnica

La fecha para determinar el estado de la técnica es el 26 de enero de 2016 que corresponde a la fecha de presentación de la solicitud de prioridad.

Consultadas las fuentes de información con que se cuenta, se encontraron los siguientes documentos que anticipan el objeto de esta solicitud:

Ítem	Documento	Título	Fecha de Publicación
D1	US 2003/0066057	“System, method and article of manufacture for collaborative hardware design”	03 de abril de 2003
D2	US 2014/0040552	“Multi-core compute cache coherency with a release consistency memory ordering model”	06 de febrero de 2014

El documento D1¹ divulga un sistema, método y artículo de fabricación para proporcionar uno o más módulos que se ajustan a una especificación de diseño de hardware. Se recibe una especificación de usuario para al menos una parte de un diseño de hardware. Se identifican uno o más módulos que se ajustan a la especificación. Los módulos se recuperan y se envían al usuario. El módulo se puede usar para configurar un dispositivo de hardware configurable (resumen).

¹ <https://patents.google.com/patent/US20030066057A1/en>

Oficio N° 3023 de 2 de marzo de 2020

Ref. Expediente N° NC2018/0008835

El documento D2² divulga un método que incluye almacenar, con un primer procesador programable, datos variables compartidos en líneas de caché de un primer caché del primer procesador. El método incluye además ejecutar, con el primer procesador programable, una operación de almacenamiento con liberación, ejecutar, con un segundo procesador programable, una operación de carga con adquisición, y cargar, con el segundo procesador programable, el valor del valor compartido de datos variables de un caché del segundo procesador programable (resumen).

4. Examen de Patentabilidad (Art. 14 Decisión 486)

4.1. Novedad (Art 16 Decisión 486)

La reivindicación 1 consiste en: “Un núcleo canalizado algorítmico reutilizable, que comprende: (...)” (Radicado N° NC2018/0008835 del 24 de agosto de 2018).

Tabla 1. Comparación de las características técnicas esenciales, con las del estado de la técnica

Características esenciales	D1
Un núcleo canalizado algorítmico reutilizable, que comprende:	El MMT según una realización ejemplar de la presente invención logra una reconfigurabilidad flexible mediante el uso de dos FPGA Virtex Xilinx XCV1000 de un millón de compuertas independientes. Uno de los FPGA permanece estáticamente configurado con la funcionalidad de red cuando el dispositivo está encendido. El otro FPGA se reconfigura con los datos proporcionados por el maestro. Los dos FPGA se comunican directamente a través de un bus de 36 bits con 4 bits reservados para el protocolo de enlace y dos canales unidireccionales de 16 bits (pág. 13, párr. 0206).
una unidad de procesamiento;	Una configuración de hardware típica de una estación de trabajo de acuerdo con una realización preferida tiene una unidad central de procesamiento (110), tal como un microprocesador, y varias otras unidades interconectadas a través de un bus de sistema (112) (pág. 2, párr. 0047).
un arreglo de puertas programables de campo reconfigurables, en donde las puertas programables de campo son programadas por un compilador canalizado de correlación algorítmica, de modo que el compilador canalizado de correlación algorítmica precompila el código fuente diseñado para funcionar en un procesador estándar sin procesamiento paralelo para procesamiento por la unidad de procesamiento;	El MMT según una realización ejemplar de la presente invención logra una reconfigurabilidad flexible mediante el uso de dos FPGA Virtex Xilinx XCV1000 de un millón de compuertas independientes. Uno de los FPGA permanece estáticamente configurado con la funcionalidad de red cuando el dispositivo está encendido. El otro FPGA se reconfigura con los datos proporcionados por el maestro. Los dos FPGA se comunican directamente a través de un bus de 36 bits con 4 bits reservados para el protocolo de enlace y dos canales unidireccionales de 16 bits (pág. 13, párr. 0206). Una realización preferida se escribe usando Handel-C. Handel-C es un lenguaje de programación que permite a un ingeniero de software o hardware apuntar directamente a las FPGAs (Field Programmable Gate Arrays) de una manera similar a las herramientas de desarrollo de compilador cruzado de microprocesador clásico, sin recurrir a un lenguaje de descripción de hardware. De este modo, el diseñador puede darse cuenta directamente de la capacidad informática en tiempo real del FPGA (pág. 4, párr. 0071). Handel-C está diseñado para permitir la compilación de programas en hardware síncrono. Su objetivo es compilar

² <https://patents.google.com/patent/US20140040552A1/en>

Oficio N° 3023 de 2 de marzo de 2020

Ref. Expediente N° NC2018/0008835

	algoritmos de alto nivel directamente en hardware de nivel de puerta (pág. 4, párr. 0072). Handel-C incluye construcciones paralelas que proporcionan los medios para que el programador aproveche este beneficio en sus aplicaciones. El compilador compila y optimiza el código fuente de Handel-C en un archivo adecuado para la simulación o una lista de red que se puede colocar y enrutar en un FPGA real (pág. 4, párr. 0075). En la Tabla 1 se exponen varias macros Handel-C que pueden generarse para su uso en diversas implementaciones de la presente invención. El documento "Manual de referencia del lenguaje Handel-C: versión 3", incorporado por referencia anterior, proporciona más información sobre la generación de macros en Handel-C (pág. 18, párr. 0318). Fpgacomms.h Commun- Fpgacomms () Implementa dos canales de hardware unidireccionales de 16 bits para comunicarse entre los dos FPGA (pág. 19, Tabla 1).
la unidad de procesamiento y el compilador canalizado de correlación algorítmica configuran las puertas programables de campo para que funcionen como procesadores canalizados paralelos.	El procesamiento se ha dividido entre el hardware y el procesador de acuerdo con dos criterios. En primer lugar, se escribe algo de código para el procesador porque requiere mucho control pero no necesita ejecutarse particularmente rápido (ahorrando así espacio en el FPGA), pero también se ha particionado parte del código en el procesador para que, con pequeños cambios en el programa código, el decodificador se puede cambiar para que pueda manejar transmisiones de audio MPEG2 y, por lo tanto, se puede usar en conducción con un decodificador de video para reproducir películas completas (pág. 17, párr. 0286). La biblioteca parallel_port.h contiene rutinas para acceder al puerto paralelo. Esto implementa un controlador de puerto paralelo como un proceso independiente, modelado estrechamente en la interfaz de puerto paralelo que se encuentra en una PC IBM. El controlador permite el acceso simultáneo a los puertos de control, estado y datos (como se define en una PC IBM) de la interfaz paralela. Se accede a estos puertos leyendo y escribiendo en los canales del proceso del controlador. Las lecturas y escrituras en estos canales se encapsulan en otros procedimientos macro para proporcionar una API intuitiva (pág. 31, párr. 0636).

De acuerdo con la tabla anterior, las características esenciales del núcleo canalizado algorítmico reutilizable de la reivindicación 1 habrían sido divulgadas previamente en el documento D1, por lo tanto, la materia reivindicada no es nueva.

Asimismo, el documento D1 divulga las características de las reivindicaciones dependientes así:

- Reivindicación 2: Dichas bibliotecas se pueden generar utilizando macros precompiladoras. En general, se accede a una biblioteca que incluye una pluralidad de funciones. Se prueba una constante del precompilador para que se pueda seleccionar una o más de las funciones de la biblioteca en función de la prueba (pág. 9, párr. 0144).
- Reivindicación 3: El MMT según una realización ejemplar de la presente invención

Oficio N° 3023 de 2 de marzo de 2020

Ref. Expediente N° NC2018/0008835

logra una reconfigurabilidad flexible mediante el uso de dos FPGA Virtex Xilinx XCV1000 de un millón de compuertas independientes. Uno de los FPGA permanece estáticamente configurado con la funcionalidad de red cuando el dispositivo está encendido. El otro FPGA se reconfigura con los datos proporcionados por el maestro. Los dos FPGA se comunican directamente a través de un bus de 36 bits con 4 bits reservados para el protocolo de enlace y dos canales unidireccionales de 16 bits (pág. 13, párr. 0206). Una realización preferida se escribe usando Handel-C. Handel-C es un lenguaje de programación que permite a un ingeniero de software o hardware apuntar directamente a las FPGAs (Field Programmable Gate Arrays) de una manera similar a las herramientas de desarrollo de compilador cruzado de microprocesador clásico, sin recurrir a un lenguaje de descripción de hardware. De este modo, el diseñador puede darse cuenta directamente de la capacidad informática en tiempo real del FPGA (pág. 4, párr. 0071). Handel-C está diseñado para permitir la compilación de programas en hardware síncrono. Su objetivo es compilar algoritmos de alto nivel directamente en hardware de nivel de puerta (pág. 4, párr. 0072). Handel-C incluye construcciones paralelas que proporcionan los medios para que el programador aproveche este beneficio en sus aplicaciones. El compilador compila y optimiza el código fuente de Handel-C en un archivo adecuado para la simulación o una lista de red que se puede colocar y enrutar en un FPGA real (pág. 4, párr. 0075). Dichas bibliotecas se pueden generar utilizando macros precompiladoras. En general, se accede a una biblioteca que incluye una pluralidad de funciones. Se prueba una constante del precompilador para que se pueda seleccionar una o más de las funciones de la biblioteca en función de la prueba. En un aspecto, la constante del precompilador puede incluir una pluralidad de versiones. Como opción, la versión puede seleccionarse utilizando una macro de precompilador. En otro aspecto, la constante del precompilador se prueba para determinar el estado de un aparato en el que se ejecutan las funciones. En tal aspecto, el estado del aparato puede basarse en un tamaño de bit actual. En otro aspecto, la biblioteca puede estar escrita en Handel-C (pág. 9, párr. 0144).

- Reivindicación 4: Otra realización de la presente invención puede escribirse al menos en parte usando JAVA, C y el lenguaje C ++ y utilizar metodología de programación orientada a objetos (pág. 5, párr. 0078).
- Reivindicación 5: La plataforma ilustrativa desarrollada para este propósito se llama Terminal Multimedia (MMT). No presenta un programa almacenado dedicado ni una Unidad Central de Procesamiento (CPU). En cambio, los programas se implementan en matrices de puertas programables de campo (FPGA) que se utilizan tanto para controlar periféricos como para procesar datos con el fin de crear una flexibilidad similar a la CPU utilizando solo una lógica reconfigurable y una metodología de diseño de software (pág. 13, párr. 0204). El tiempo dedicado a cargar las instrucciones de configuración y verificar que las instrucciones se hayan cargado correctamente puede ser significativo, particularmente cuando dicha carga se lleva a cabo en el campo (pág. 3, párr. 0059). Por muchas razones, a menudo es deseable tener capacidades de reprogramación en el sistema para que la reconfiguración de FPGA se pueda llevar a cabo en el campo (pág. 3, párr. 0060). Los dispositivos FPGA que tienen memorias de configuración del tipo reprogramable son, al menos en teoría, "programables en el sistema" (ISP). Esto significa que no existe la posibilidad de cambiar las instrucciones de configuración dentro del dispositivo FPGA mientras el dispositivo FPGA está "en el sistema" porque la memoria de configuración es inherentemente reprogramable (pág. 3, párr. 0061).
- Reivindicación 6: El procesamiento se ha dividido entre el hardware y el procesador



Oficio N° 3023 de 2 de marzo de 2020

Ref. Expediente N° NC2018/0008835

de acuerdo con dos criterios. En primer lugar, se escribe algo de código para el procesador porque requiere mucho control pero no necesita ejecutarse particularmente rápido (ahorrando así espacio en el FPGA), pero también se ha particionado parte del código en el procesador para que, con pequeños cambios en el programa código, el decodificador se puede cambiar para que pueda manejar transmisiones de audio MPEG2 y, por lo tanto, se puede usar en conducción con un decodificador de video para reproducir películas completas (pág. 17, párr. 0286).

- Reivindicación 7: El FGPA contiene preferiblemente la lógica necesaria para lo siguiente: (pág. 35, párr. 0753) bus de comando (pág. 35, párr. 0767). Cada FPGA tiene acceso a dos bancos de SRAM compartida, nuevamente organizados como 256K palabras x 32 bits. Estos tienen un tiempo de acceso de 16 ns. Se utiliza una serie de interruptores rápidos para cambiar estas RAM entre los FPGA y estos son controlados por el CPLD que actúa como árbitro (pág. 24, párr. 0443). Dispositivo lógico programable complejo (CPLD) (pág. 14, párr. 0216).
- Reivindicación 9: Se proporciona una red de interconexión para transportar tráfico de señal dentro del dispositivo FPGA entre varios CLB y/o entre varios IOB y/o entre varios IOB y CLB. Al menos parte de la red de interconexión es típicamente configurable para permitir el enrutamiento de señales definido de manera programable entre varios CLB y/o IOB de acuerdo con las instrucciones de enrutamiento definidas por el usuario almacenadas en los medios de memoria que definen la configuración (pág. 3, párr. 0055). El tiempo dedicado a cargar las instrucciones de configuración y verificar que las instrucciones se hayan cargado correctamente puede ser significativo, particularmente cuando dicha carga se lleva a cabo en el campo (pág. 3, párr. 0059). Por muchas razones, a menudo es deseable tener capacidades de reprogramación en el sistema para que la reconfiguración de FPGA se pueda llevar a cabo en el campo (pág. 3, párr. 0060). Los dispositivos FPGA que tienen memorias de configuración del tipo reprogramable son, al menos en teoría, "programables en el sistema" (ISP). Esto significa que no existe la posibilidad de cambiar las instrucciones de configuración dentro del dispositivo FPGA mientras el dispositivo FPGA está "en el sistema" porque la memoria de configuración es inherentemente reprogramable (pág. 3, párr. 0061). Cada FPGA tiene acceso a dos bancos de SRAM compartida, nuevamente organizados como 256K palabras x 32 bits. Estos tienen un tiempo de acceso de 16 ns. Se utiliza una serie de interruptores rápidos para cambiar estas RAM entre los FPGA y estos son controlados por el CPLD que actúa como árbitro (pág. 24, párr. 0443). Dispositivo lógico programable complejo (CPLD) (pág. 14, párr. 0216). El procesamiento se ha dividido entre el hardware y el procesador de acuerdo con dos criterios. En primer lugar, se escribe algo de código para el procesador porque requiere mucho control pero no necesita ejecutarse particularmente rápido (ahorrando así espacio en el FPGA), pero también se ha particionado parte del código en el procesador para que, con pequeños cambios en el programa código, el decodificador se puede cambiar para que pueda manejar transmisiones de audio MPEG2 y, por lo tanto, se puede usar en conducción con un decodificador de video para reproducir películas completas (pág. 17, párr. 0286).
- Reivindicación 10: Se emula un procesador en lógica reconfigurable. El procesador interpreta los datos en el flujo de bits y descuantifica los datos decodificados en el software. El procesador también se puede utilizar para controlar el hardware reconfigurable (pág. 16, párr. 0254). El procesador tiene el control general del hardware y decide qué parámetros pasar a los bancos de filtros y al decodificador Huffman. Para pasar datos hacia y desde los otros procesos, al hardware se le ha asignado al espacio de direcciones por encima de la memoria física (1 Meg). La lógica

Oficio N° 3023 de 2 de marzo de 2020

Ref. Expediente N° NC2018/0008835

de control del hardware incluye 16 registros de 32 bits, que se pueden usar para suministrar parámetros al hardware o leer datos del hardware (por ejemplo, las tablas de Huffman para usar para una trama particular se pasan al hardware a través de algunos de estos registros de estado y el número total de bits leídos durante la decodificación devueltos en otro registro para que el procesador los lea). La lógica de control para el hardware también se ha mapeado en varias direcciones. Por lo tanto, para iniciar el proceso de decodificación de Huffman, el procesador escribe en la dirección apropiada y luego se detiene hasta que se completa la decodificación (pág. 17, párr. 0284).

Las reivindicaciones dependientes 2 a 7, 9 y 10 tampoco cumplen con el requisito de novedad.

4.2. Nivel Inventivo (Art. 18 Decisión 486)

La reivindicación 8 consiste en: *“El núcleo de la reivindicación 7, que comprende además (...)”* (Radicado N° NC2018/0008835 del 24 de agosto de 2018).

El documento D1 se considera el estado de la técnica más cercano a la invención definida en la reivindicación 8 porque divulga un MMT según una realización ejemplar de la invención, el cual logra una reconfigurabilidad flexible mediante el uso de dos FPGA Virtex Xilinx XCV1000 de un millón de compuertas independientes. Uno de los FPGA permanece estáticamente configurado con la funcionalidad de red cuando el dispositivo está encendido. El otro FPGA se reconfigura con los datos proporcionados por el maestro. Los dos FPGA se comunican directamente a través de un bus de 36 bits con 4 bits reservados para el protocolo de enlace y dos canales unidireccionales de 16 bits (pág. 13, párr. 0206).

La diferencia entre la invención definida en la reivindicación 8 y el MMT divulgado de D1 consiste en que la invención incluye un procesador de señal digital, en el que el procesador de señal digital comprende un procesador matemático reconfigurable para realizar cálculos matemáticos.

El efecto que se logra al incluir un procesador de señal digital, en el que el procesador de señal digital comprende un procesador matemático reconfigurable para realizar cálculos matemáticos es optimizar los núcleos para aplicaciones que requieran operaciones numéricas a muy alta velocidad.

Por lo tanto, el problema técnico objetivo que pretende resolver esta invención se puede formular así: ¿Cómo modificar el MMT conocido en D1 con el fin de optimizar los núcleos para aplicaciones que requieran operaciones numéricas a muy alta velocidad?

Sin embargo, el incluir un procesador de señal digital, en el que el procesador de señal digital comprende un procesador matemático reconfigurable para realizar cálculos matemáticos para optimizar los núcleos para aplicaciones que requieran operaciones numéricas a muy alta velocidad, ya está divulgado en el documento D2 que se refiere a que aún otros procesadores, como los procesadores de señal digital (DSP) pueden ser adecuados para aplicaciones específicas, como el procesamiento de señales digitales, que pueden incluir la realización de operaciones matemáticas específicas. Dichas operaciones pueden incluir transformadas rápidas de Fourier (FFT) y transformaciones discretas de coseno (DCT) (pág. 2, párr. 0024).



Oficio N° 3023 de 2 de marzo de 2020

Ref. Expediente N° NC2018/0008835

En consecuencia, la persona normalmente versada en la materia, estaría motivada a incluir un procesador de señal digital, en el que el procesador de señal digital comprende un procesador matemático reconfigurable para realizar cálculos matemáticos del documento D2 en el MMT de D1, para así llegar al objeto de la reivindicación 8. Por lo cual se considera obvia.

La reivindicación dependiente 8 no menciona alguna característica que, junto con las características de la reivindicación independiente, aporte nivel inventivo al núcleo de la reivindicación 1, por lo cual se considera que no tiene nivel inventivo.

La reivindicación 11 consiste en: “Un sistema que comprende una pluralidad de núcleos según la reivindicación 1 (...)” (Radicado N° NC2018/0008835 del 24 de agosto de 2018).

Tabla 2. Comparación de las características técnicas esenciales, con las del estado de la técnica:

Características esenciales	D1	D2
Un sistema que comprende una pluralidad de núcleos,	Un sistema, método y artículo de fabricación para proporcionar uno o más módulos que se ajustan a una especificación de diseño de hardware. Se recibe una especificación de usuario para al menos una parte de un diseño de hardware. Se identifican uno o más módulos que se ajustan a la especificación. Los módulos se recuperan y se envían al usuario. El módulo se puede usar para configurar un dispositivo de hardware configurable (resumen).	Una unidad central de procesamiento (CPU), una unidad de procesamiento de gráficos (GPU), un procesador de señal digital (DSP) u otro tipo de núcleo de procesamiento pueden ejecutar instrucciones al adquirir o escribir en una variable compartida entre todos los procesadores de un sistema informático multinúcleo (pág. 1, párr. 0004).
procesar todas las operaciones matemáticas usando procesadores de señal digital de uno o más de la pluralidad de núcleos;	No se menciona.	Una unidad de procesamiento central (CPU), una unidad de procesamiento de gráficos (GPU), un procesador de señal digital (DSP) u otro tipo de núcleo de procesamiento pueden ejecutar instrucciones al adquirir o escribir en una variable compartida entre todos los procesadores de un sistema de computación central. Cuando un núcleo del procesador (denominado "procesador") intenta acceder a datos variables compartidos, el núcleo del procesador puede ejecutar operaciones de sincronización, como operaciones de carga con adquisición y/o almacenamiento con liberación para garantizar que cualquier operación aplicada a datos variables compartidos ocurre de manera ordenada y coherente (pág. 1, párr. 0004). Aún otros procesadores, como los procesadores de señal digital (DSP) pueden ser adecuados para aplicaciones específicas, como el procesamiento de señales digitales, que pueden incluir la realización de operaciones matemáticas específicas. Dichas operaciones pueden incluir transformadas rápidas

Oficio N° 3023 de 2 de marzo de 2020

Ref. Expediente N° NC2018/0008835

		de Fourier (FFT) y transformaciones discretas de coseno (DCT) (pág. 2, párr. 0024).
procesar todas las funciones lógicas usando uno o más de los procesadores lógicos de uno o más de la pluralidad de núcleos.	Se emula un procesador en lógica reconfigurable. El procesador interpreta los datos en el flujo de bits y descuantifica los datos decodificados en el software. El procesador también se puede utilizar para controlar el hardware reconfigurable (pág. 16, párr. 0254). Estos se dividen entre implementaciones de hardware puro y algo de software en un núcleo de procesador RISC incorporado ligero, preferiblemente implementado en Handel-C (pág. 16, párr. 0258).	Recientemente ha habido un movimiento hacia la denominada informática heterogénea en la que múltiples procesadores o núcleos de procesador ("núcleos") están interconectados en un sistema. Sin embargo, existen muchos desafíos asociados con los sistemas informáticos heterogéneos. Ciertos núcleos pueden ser adecuados para tareas particulares. Como ejemplo, las CPU pueden ser más adecuadas para ejecutar programas con cantidades significativas de lógica condicional (por ejemplo, ramas y saltos) (pág. 2, párr. 0024).

El documento D1 se considera el estado de la técnica más cercano a la invención definida en la reivindicación 11 porque divulga un MMT según una realización ejemplar de la invención, el cual logra una reconfigurabilidad flexible mediante el uso de dos FPGA Virtex Xilinx XCV1000 de un millón de compuertas independientes. Uno de los FPGA permanece estáticamente configurado con la funcionalidad de red cuando el dispositivo está encendido. El otro FPGA se reconfigura con los datos proporcionados por el maestro. Los dos FPGA se comunican directamente a través de un bus de 36 bits con 4 bits reservados para el protocolo de enlace y dos canales unidireccionales de 16 bits (pág. 13, párr. 0206).

La diferencia entre la invención definida en la reivindicación 11 y el MMT divulgado de D1 consiste en que la invención incluye procesar todas las operaciones matemáticas usando procesadores de señal digital de uno o más de la pluralidad de núcleos.

El efecto que se logra al incluir procesar todas las operaciones matemáticas usando procesadores de señal digital de uno o más de la pluralidad de núcleos es optimizar los núcleos para aplicaciones que requieran operaciones numéricas a muy alta velocidad.

Por lo tanto, el problema técnico objetivo que pretende resolver esta invención se puede formular así: ¿Cómo modificar el MMT conocido en D1 con el fin de optimizar los núcleos para aplicaciones que requieran operaciones numéricas a muy alta velocidad?

Sin embargo, el incluir procesar todas las operaciones matemáticas usando procesadores de señal digital de uno o más de la pluralidad de núcleos para optimizar los núcleos para aplicaciones que requieran operaciones numéricas a muy alta velocidad, ya está divulgado en el documento D2 que se refiere a que una unidad de procesamiento central (CPU), una unidad de procesamiento de gráficos (GPU), un procesador de señal digital (DSP) u otro tipo de núcleo de procesamiento pueden ejecutar instrucciones al adquirir o escribir en una variable compartida entre todos los procesadores de un sistema de computación central. Cuando un núcleo del procesador (denominado "procesador") intenta acceder a datos variables compartidos, el núcleo del procesador puede ejecutar operaciones de sincronización, como operaciones de carga con adquisición y/o almacenamiento con liberación para garantizar que cualquier operación aplicada a datos variables compartidos ocurre de manera ordenada y coherente (pág. 1, párr. 0004). Aún otros procesadores, como los procesadores de señal digital (DSP) pueden ser adecuados para aplicaciones específicas, como el procesamiento de señales digitales, que pueden

Oficio N° 3023 de 2 de marzo de 2020

Ref. Expediente N° NC2018/0008835

incluir la realización de operaciones matemáticas específicas. Dichas operaciones pueden incluir transformadas rápidas de Fourier (FFT) y transformaciones discretas de coseno (DCT) (pág. 2, párr. 0024).

En consecuencia, la persona normalmente versada en la materia, estaría motivada a incluir procesar todas las operaciones matemáticas usando procesadores de señal digital de uno o más de la pluralidad de núcleos del documento D2 en el MMT de D1, para así llegar al objeto de la reivindicación 11. Por lo cual se considera obvia.

Asimismo, el documento D1 divulga las características de las reivindicaciones dependientes así:

- Reivindicación 12: Una unidad de procesamiento central (CPU), una unidad de procesamiento de gráficos (GPU), un procesador de señal digital (DSP) u otro tipo de núcleo de procesamiento pueden ejecutar instrucciones al adquirir o escribir en una variable compartida entre todos los procesadores de un sistema de computación central. Cuando un núcleo del procesador (denominado "procesador") intenta acceder a datos variables compartidos, el núcleo del procesador puede ejecutar operaciones de sincronización, como operaciones de carga con adquisición y/o almacenamiento con liberación para garantizar que cualquier operación aplicada a datos variables compartidos ocurre de manera ordenada y coherente (pág. 1, párr. 0004). Al menos parte de la red de interconexión es típicamente configurable para permitir el enrutamiento de señales definido de manera programable entre varios CLB y/o IOB de acuerdo con las instrucciones de enrutamiento definidas por el usuario almacenadas en los medios de memoria que definen la configuración (pág. 3, párr. 0055).
- Reivindicación 13: Para cumplir con los desafíos del sistema descrito anteriormente, el diseño MMT se puede hacer usando Handel-C. Está basado en ANSI-C y lo aprende rápidamente cualquier persona que haya realizado el desarrollo de software C. Se han implementado extensiones para admitir paralelismo, variables de ancho arbitrario y otras características familiares en el diseño de hardware, pero se enfoca mucho en las metodologías de diseño de software (pág. 4, párr. 0210).
- Reivindicación 14: Dichas bibliotecas se pueden generar utilizando macros precompiladoras. En general, se accede a una biblioteca que incluye una pluralidad de funciones. Se prueba una constante del precompilador para que se pueda seleccionar una o más de las funciones de la biblioteca en función de la prueba. En un aspecto, la constante del precompilador puede incluir una pluralidad de versiones. Como opción, la versión puede seleccionarse utilizando una macro de precompilador. En otro aspecto, la constante del precompilador se prueba para determinar el estado de un aparato en el que se ejecutan las funciones (pág. 9, párr. 0144). Los FPGA pueden leer y escribir en la memoria directamente, mientras que el procesador ARM tiene acceso a ella a través del CPLD. Las macros para leer y escribir comandos simples en la máquina de estado interna de Flash RAM se proporcionan en la biblioteca de macros klib.h (como recuperar la información de identificación y estado de la RAM), pero el desarrollador debe mejorarlas para implementar procedimientos más complejos como la programación de bloques y el bloqueo. Las macros proporcionadas están destinadas a ilustrar el mecanismo básico para acceder a Flash RAM (pág. 23, párr. 0430).
- Reivindicación 15: Por lo general, cada uno de los muchos CLB de un FPGA tiene al menos una tabla de búsqueda (LUT) configurable por el usuario para definir cualquier



Oficio N° 3023 de 2 de marzo de 2020

Ref. Expediente N° NC2018/0008835

tabla de verdad deseada, en la medida permitida por el espacio de direcciones de la LUT. Cada CLB puede tener otros recursos tales como recursos de preprocesamiento de señal de entrada LUT y recursos de posprocesamiento de señal de salida LUT (pág. 3, párr. 0054). El número de bits lógicos que se utilizan para definir las instrucciones de configuración de un dispositivo FPGA dado tiende a ser bastante grande (por ejemplo, 1 Megabits o más) y generalmente crece con el tamaño y la complejidad del FPGA objetivo. El tiempo dedicado a cargar las instrucciones de configuración y verificar que las instrucciones se hayan cargado correctamente puede ser significativo, particularmente cuando dicha carga se lleva a cabo en el campo (pág. 3, párr. 0059).

- Reivindicaciones 16 a 18: El MMT según una realización ejemplar de la presente invención logra una reconfigurabilidad flexible mediante el uso de dos FPGA Virtex Xilinx XCV1000 de un millón de compuertas independientes. Uno de los FPGA permanece estáticamente configurado con la funcionalidad de red cuando el dispositivo está encendido. El otro FPGA se reconfigura con los datos proporcionados por el maestro. Los dos FPGA se comunican directamente a través de un bus de 36 bits con 4 bits reservados para el protocolo de enlace y dos canales unidireccionales de 16 bits (pág. 13, párr. 0206). Una realización preferida se escribe usando Handel-C. Handel-C es un lenguaje de programación que permite a un ingeniero de software o hardware apuntar directamente a las FPGAs (Field Programmable Gate Arrays) de una manera similar a las herramientas de desarrollo de compilador cruzado de microprocesador clásico, sin recurrir a un lenguaje de descripción de hardware. De este modo, el diseñador puede darse cuenta directamente de la capacidad informática en tiempo real del FPGA (pág. 4, párr. 0071). Handel-C está diseñado para permitir la compilación de programas en hardware síncrono. Su objetivo es compilar algoritmos de alto nivel directamente en hardware de nivel de puerta (pág. 4, párr. 0072). Handel-C incluye construcciones paralelas que proporcionan los medios para que el programador aproveche este beneficio en sus aplicaciones. El compilador compila y optimiza el código fuente de Handel-C en un archivo adecuado para la simulación o una lista de red que se puede colocar y enrutar en un FPGA real (pág. 4, párr. 0075). En la Tabla 1 se exponen varias macros Handel-C que pueden generarse para su uso en diversas implementaciones de la presente invención. El documento "Manual de referencia del lenguaje Handel-C: versión 3", incorporado por referencia anterior, proporciona más información sobre la generación de macros en Handel-C (pág. 18, párr. 0318). Fpgacomms.h Commun- Fpgacomms () Implementa dos canales de hardware unidireccionales de 16 bits para comunicarse entre los dos FPGA (pág. 19, Tabla 1).
- Reivindicaciones 19 y 20: Cada FPGA tiene acceso a dos bancos de SRAM compartida, nuevamente organizados como 256K palabras x 32 bits. Estos tienen un tiempo de acceso de 16 ns. Se utiliza una serie de interruptores rápidos para cambiar estas RAM entre los FPGA y estos son controlados por el CPLD que actúa como árbitro (pág. 24, párr. 0443). El procesamiento se ha dividido entre el hardware y el procesador de acuerdo con dos criterios. En primer lugar, se escribe algo de código para el procesador porque requiere mucho control pero no necesita ejecutarse particularmente rápido (ahorrando así espacio en el FPGA), pero también se ha particionado parte del código en el procesador para que, con pequeños cambios en el programa código, el decodificador se puede cambiar para que pueda manejar transmisiones de audio MPEG2 y, por lo tanto, se puede usar en conducción con un decodificador de video para reproducir películas completas (pág. 17, párr. 0286).

Oficio N° 3023 de 2 de marzo de 2020

Ref. Expediente N° NC2018/0008835

De igual manera, el documento D2 divulga las características de las reivindicaciones dependientes así:

- Reivindicación 19: Aún otros procesadores, como los procesadores de señal digital (DSP) pueden ser adecuados para aplicaciones específicas, como el procesamiento de señales digitales, que pueden incluir la realización de operaciones matemáticas específicas. Dichas operaciones pueden incluir transformadas rápidas de Fourier (FFT) y transformaciones discretas de coseno (DCT) (pág. 2, párr. 0024). Como ejemplo, las CPU pueden ser más adecuadas para ejecutar programas con cantidades significativas de lógica condicional (por ejemplo, ramas y saltos) (pág. 2, párr. 0024).
- Reivindicación 21: La placa puede incluir dos FPGA Xilinx Virtex™ 2000e (952), (954), un procesador Intel StrongARM SA1110 (956), una gran cantidad de memoria (958), (960) y varios puertos de E/S (962) (pág. 19, párr. 0319).

Las reivindicaciones dependientes 12 a 21 no mencionan alguna característica que, junto con las características de la reivindicación independiente, aporte nivel inventivo al sistema de la reivindicación 11, por lo cual se considera que no tienen nivel inventivo.

5. Conclusión con respecto a novedad y/o nivel inventivo

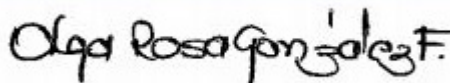
Los requisitos de Patentabilidad de la presente solicitud están afectados así:

Ítem	Documento	Reivindicaciones afectadas	Requisito que afecta
D1	US 2003/0066057	1 a 7, 9 y 10	Novedad
		8 y 11 a 20	Nivel Inventivo
D2	US 2014/0040552	8, 11, 19 y 21	Nivel inventivo

En cumplimiento del artículo 45 de la Decisión 486 de 2000, se le indica que debe dar respuesta dentro del término de sesenta (60) días contados a partir de la fecha de notificación de la presente comunicación. En caso de no dar respuesta al presente requerimiento, o si a pesar de la respuesta subsistieran los impedimentos para la concesión, se denegará la patente.

Si como respuesta a este requerimiento, el solicitante modifica: i) el capítulo reivindicatorio, ii) la descripción de la invención a proteger, iii) o presenta un nuevo capítulo reivindicatorio deberá observar las instrucciones contenidas en el numeral 1.2.2.1.1., del Capítulo Primero del Título X de la Circular Única y la Dirección de Nuevas Creaciones seguirá el procedimiento descrito en el numeral 1.2.2.5.1.

Notifíquese el presente oficio conforme a lo dispuesto en el numeral 6.2 del Capítulo Sexto del Título I de la Circular Única, informándoles que contra el mismo no procede recurso alguno en actuación administrativa.



OLGA ROSA GONZÁLEZ FONSECA
DIRECTORA DE NUEVAS CREACIONES (E)

Oficio N° 3023 de 2 de marzo de 2020

Ref. Expediente N° NC2018/0008835

**INFORME DE BÚSQUEDA
EXAMEN DE PATENTABILIDAD**

Solicitud No.		Examen de patentabilidad							
NC2018/0008835		1°	X	2°		3°		Revoca	
Solicitud Internacional No.		Fecha solicitud internacional: (DD/MM/AÑO)							
PCT/US2017/015143		26 de enero de 2017							
Fecha de determinación estado de la técnica (DD/MM/AÑO)				Prioridad		Presentación			
26/01/2016				X					
Solicitante									
ICAT LLC									
CRITERIOS DE BÚSQUEDA									
Reivindicaciones objeto de búsqueda				1 a 23					
Palabras clave utilizadas				FPGA, DSP, multi-core, process, software					
Clasificaciones utilizadas para la búsqueda				CIP:	G06F 7/57, G06F 9/45, G06F 19/00				
DOCUMENTOS CONSIDERADOS RELEVANTES (Documentos patente, Literatura no patente (LNP): Artículos, catálogos, etc.)									
Informes de búsqueda / Bases de datos consultadas	N° de solicitud de patente / N° de patente / Autor LNP ⁽¹⁾	Fecha de Publicación /Presentación (DD/MM/AAAA)	Reivindicaciones /Secuencias afectadas	Citas relevantes del estado de la técnica			Categoría del documento citado ⁽²⁾		
SIPI	CO15116548	31/08/2015	-	-			A		
ISA	US2003066057	03/04/2003	1 a 20	Todo el documento			X, Y		
	US7818725	19/10/2010	-	-			A		
	US2009320008	24/12/2009	-	-			A		
	US2005010743	13/01/2005	-	-			A		
EPO	Weinhardt ³	04/1999	-	-			A		
USPTO	US2001032327	18/10/2001	-	-			A		
	US2012109849	03/05/2012	-	-			A		
	US2014040552	06/02/2014	8 y 11 a 21	Todo el documento			Y		
	US2008218202	11/09/2008	-	-			A		
	US2006081971	20/04/2006	-	-			A		
Googlepatent	US5815715 KR100998586 US20060005173	29/09/1998 07/12/2010 05/01/2006	-	-			A		
(1) Cita completa literatura no patente (LNP)		³ Weinhardt, M., & Luk, W. (1999, April). Pipeline vectorization for reconfigurable systems. In Seventh Annual IEEE Symposium on Field-Programmable Custom Computing Machines (Cat. No. PR00375) (pp. 52-62). IEEE.							
(2) Categorías de documentos citados									
“A” Documento que define el estado general de la técnica no considerado como particularmente relevante. “E” Solicitud de patente o patente anterior pero publicada en la fecha de presentación internacional o en fecha posterior. “L” Documento citado para llamar la atención sobre por ejemplo, dudas acerca de una reivindicación de prioridad o para determinar la fecha de publicación de otra cita, siempre van acompañados de una explicación, por la cual se citan. “O” Documento del tipo actas de conferencia, este tipo de documentos siempre estará acompañado por otra letra que indique la pertinencia del documento, por ejemplo, O,X, O,Y o bien O,A. “P” Documento publicado antes de la fecha de presentación internacional pero con posterioridad a la fecha de prioridad reivindicada. “T” Documento publicado en fecha posterior a la de presentación o la de prioridad de la solicitud internacional y no entra en conflicto con dicha solicitud, pero puede ser útil para la mejor comprensión del principio o teoría en que se basa la invención, o para demostrar que el razonamiento o los hechos en los que se basa dicha invención son incorrectos. “X” Documento particularmente relevante; la invención reivindicada no puede considerarse nueva o que implique una actividad inventiva por referencia al documento aisladamente considerado. “Y” Documento particularmente relevante; la invención reivindicada no puede considerarse que implique una actividad inventiva cuando el documento se asocia a otro u otros documentos de la misma naturaleza, cuya combinación resulta evidente para un experto en la materia.									

Oficio N° 3023 de 2 de marzo de 2020

Ref. Expediente N° NC2018/0008835

Fecha del reporte de búsqueda: 22/02/2020
Examinador: Angélica María Ramírez Cano