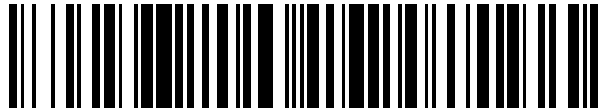


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 562 153**

21 Número de solicitud: 201531184

51 Int. Cl.:

**G06F 17/50** (2006.01)

12

SOLICITUD DE PATENTE

A1

22 Fecha de presentación:

**10.08.2015**

43 Fecha de publicación de la solicitud:

**02.03.2016**

71 Solicitantes:

**SIGNADYNE SPAIN, S.L. (100.0%)**  
**Parc Mediterrani de la Tecnologia, B1 Av. Canal**  
**Olimpic s/n**  
**08860 Castelldefels (Barcelona) ES**

72 Inventor/es:

**OLIVERIO, Néstor Hugo y**  
**ALMENDROS PARRA, Marc**

74 Agente/Representante:

**CARPINTERO LÓPEZ, Mario**

54 Título: **Sistema y método de configuración de hardware de instrumentos programables de control, test y medida**

57 Resumen:

Sistema y método de configuración de hardware de instrumentos programables de control, test y medida. Sistema y método de configuración de hardware de un instrumento programable (10) de control, test y medida que comprende una FPGA (30) integrada con dos secciones, una primera sección estática (31) y una sección dinámica (33), conectadas por una interfaz lógica (32). La sección estática (31) comprende código implementado y prefijado de fábrica (34), previamente optimizado, mientras que la sección dinámica (33) comprende código programable (35) por el propio usuario. Se simplifica así notablemente la configuración del instrumento programable (10) por parte del usuario, además de reducirse los tiempos de compilación y optimizarse la operación y el rendimiento de la FPGA (30).

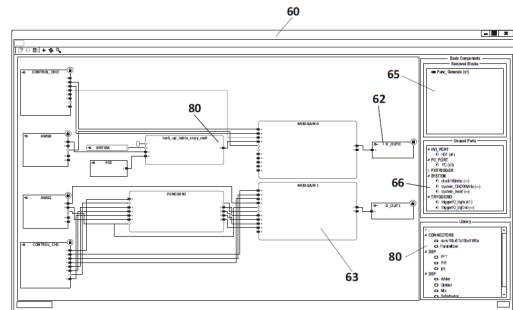


FIG. 3

## DESCRIPCIÓN

Sistema y método de configuración de hardware de instrumentos programables de control, test y medida.

5

### OBJETO DE LA INVENCION

La presente invención pertenece al campo técnico de la electrónica y, en concreto, se aplica al área industrial de los instrumentos programables para ejecutar funciones de control, test y/o medida.

10

### ANTECEDENTES DE LA INVENCION

Durante los últimos años, las exigencias impuestas sobre los instrumentos programables de control, test y/o medida se han visto incrementadas notablemente. Han crecido significativamente los requerimientos en término de número de entradas y salidas, velocidad de adquisición y capacidad de procesado. Adicionalmente, al requerimiento de procesar una gran cantidad de datos en tiempo real, se une la necesidad de adaptar dicho procesado a diversas aplicaciones particulares con una misma plataforma hardware y muchas veces en un tiempo muy reducido.

15

20

La tecnología FPGA ('arreglos de compuertas programables en campo', del inglés 'field programmable gate array') proporciona una alternativa muy prometedora para afrontar estos retos, dado su carácter reconfigurable y su gran capacidad de procesado. No obstante, para implementar las configuraciones necesarias para la correcta operación de los instrumentos de control, test y medida, es necesario un conocimiento avanzado de hardware lógico, así como de las herramientas de diseño altamente especializadas para FPGA, llegando a tener que definir interconexiones físicas y tiempos de propagación de señales internas. Esto hace que para implementar sistemas basados en FPGA se requiera expertos en la materia y además los tiempos de implementación y puesta en marcha de estos sistemas sean muy largos incluso para usuarios expertos. Los sistemas de control, test y medida requieren las prestaciones que ofrecen las FPGA pero necesitan que su uso y puesta en marcha sea rápido y lo más simple posible sin que el usuario tenga que ser un experto en programación de hardware.

25

30

35

Algunas soluciones comerciales proporcionan un hardware que permite realizar aplicaciones de control, test y medida que incluyen una FPGA y le proporcionan al usuario un paquete con un proyecto marco –framework, en inglés- para la herramienta de software provista por el fabricante de la FPGA incluida en el instrumento. Este proyecto marco incluye los componentes necesarios para interactuar con el hardware fijo del equipo y el usuario puede modificarlo para agregar las funcionalidades que desee. El problema de estos sistemas es que requiere de usuarios expertos en FPGA que por lo general no es el caso de los ingenieros de control, test y medida, y además los tiempos de desarrollo y puesta en marcha son muy largos lo que también limita mucho su aplicación a control, test y medida.

10

Con el fin de simplificar el proceso, existen algunas herramientas gráficas que permiten implementar código de FPGA de manera más sencilla, pero al no estar diseñadas para ningún instrumento específico, siguen requiriendo interacciones complejas para su implementación sobre el hardware de cada instrumento particular, siendo típicamente necesario recurrir a las herramientas de software provistas por el fabricante de la FPGA incluida en el instrumento y utilizar algún proyecto marco como en el caso anterior. Es decir, estas herramientas gráficas no contemplan de forma nativa la implementación sobre instrumentos comerciales, lo cual es una tarea muy compleja y sufre de inconvenientes similares a los descritos para la solución anterior.

20

Otras soluciones proporcionan instrumentos modulares que incluyen una FPGA y un software específico para programar estos instrumentos, que consiste en un entorno de programación gráfico que adapta un lenguaje de programación de software basado en bucles, condicionales, etc, a programación hardware FPGA. La herramienta gráfica de alto nivel permite simplificar el proceso de programación del hardware FPGA para el usuario, pero a costa de una reducción de rendimiento debido a una menor posibilidad de optimización comparada con una programación realizada con herramientas avanzadas. Además el mecanismo de programación de software adaptado a hardware presenta serios inconvenientes que según el uso que haga el usuario suele resultar en un uso muy deficiente de la FPGA.

30

En todos los casos anteriormente mencionados, durante el proceso de personalización del hardware FPGA del instrumento por parte del usuario se implementa por completo toda la lógica de la FPGA, y esto ocurre cada vez que el usuario cambia alguna funcionalidad por mínima que sea. Esto sumado a la limitada capacidad de optimización de un usuario no

35

experto, compromete seriamente el rendimiento y el funcionamiento tanto de las funciones incluidas en las librerías del fabricante como de las funciones adicionales desarrolladas por el usuario. También por este motivo, los tiempos de compilación y puesta en marcha del instrumento pueden resultar prohibitivamente largos.

5

En definitiva, sigue existiendo la necesidad en el estado de la técnica de un método y sistema de configuración de hardware de instrumentos programables de control, test y medida, que permita configurar dinámicamente dichos instrumentos de manera sencilla y optimizada, sin necesidad de tener conocimientos específicos de FPGA y reduciendo los tiempos de espera asociados a cambios en dicha configuración.

10

## DESCRIPCIÓN DE LA INVENCIÓN

La presente invención sirve para solucionar el problema mencionado anteriormente, resolviendo los inconvenientes que presentan las soluciones comentadas en el estado de la técnica, mediante una reconfiguración parcial de la FPGA que separa: una lógica FPGA dinámica programable por el usuario y una lógica FPGA estática prefijada implementada de fábrica. Ambas lógicas son reconfigurables, pero la lógica estática se desarrolla en fábrica por expertos, mientras que la dinámica se implementa con herramientas simplificadas para el usuario final, quien no necesita conocimientos de programación de FPGA.

15

20

En el contexto de la invención, se definen y usan los siguientes conceptos:

- Código de usuario: Es el código fuente "de alto nivel" que implementa las funcionalidades FPGA de usuario para personalizar la operación del instrumento programable de control, test y/o medida
- Lógica FPGA: Es el resultado del proceso de implementación del código FPGA en la FPGA y comprende las compuertas lógicas y todos los componentes internos que componen la FPGA, su posicionamiento e interconexión dentro de la FPGA.
  - Lógica FPGA Dinámica: Es el resultado de la implementación del código de usuario.
  - Lógica FPGA Estática: Es lógica FPGA prefijada, desarrollada e implementada de fábrica la cual no puede ser modificada por el usuario.

25

30

- Sección estática / dinámica: es una zona de la FPGA destinada a contener la lógica FPGA estática / dinámica.

En un primer aspecto de la invención se presenta un sistema de configuración de hardware de un instrumento programable de control, test y medida, que comprende una FPGA integrada en dicho instrumento, dotándole de capacidades de programación o reconfiguración dinámica. La FPGA comprende dos secciones:

- Una sección estática que incluye al menos una lógica FPGA estática prefijada implementada de fábrica. Dicha lógica FPGA estática puede comprender funcionalidades esenciales del instrumento cuya complejidad excede los conocimientos de un usuario medio, así como funcionalidades adicionales cuya optimización a bajo nivel con herramientas avanzadas resulta crítica para su correcto funcionamiento en la FPGA.
- Una sección dinámica que incluye al menos una lógica FPGA dinámica programable por parte del usuario. Puesto que el usuario utiliza las funcionalidades provistas por la lógica FPGA de la sección estática y no modifica la sección estática, se simplifica enormemente la implementación de la lógica FPGA dinámica con las nuevas funcionalidades de usuario. Además al no modificarse en lo absoluto la lógica FPGA estática en el proceso de implementación de la lógica FPGA dinámica se mantiene la operación optimizada de las funcionalidades de fábrica independientemente de las personalizaciones introducidas por el usuario. Asimismo, la reprogramación de la sección dinámica no afecta a la sección estática de la FPGA, por lo que la lógica FPGA estática puede continuar su operación durante la reprogramación o reconfiguración de la lógica FPGA dinámica.

La lógica FPGA estática prefijada determina a su vez una interfaz lógica fija que permite interconectar cualquier lógica FPGA dinámica con la lógica FPGA estática, lo que permite al usuario hacer uso de las funcionalidades de fábrica e interconectarlas con la lógica FPGA dinámica desarrollada por él mismo para personalizar el instrumento.

Preferentemente, la sección estática puede configurarse con múltiples lógicas FPGA estáticas prefijadas implementadas de fábrica, seleccionables por el usuario. También preferentemente, la sección dinámica puede configurarse con múltiples lógicas FPGA dinámicas intercambiables durante la operación de la lógica FPGA estática y del resto del

instrumento. Asimismo tanto la sección estática como la dinámica pueden subdividirse en subsecciones estáticas y dinámicas respectivamente, de forma que cualquiera de las subsecciones puede reconfigurarse mientras las otras siguen funcionando. Las lógicas FPGA pueden almacenarse en el ordenador del usuario o en una memoria volátil o no volátil del mismo instrumento. El intercambio entre distintas lógicas FPGA puede realizarse por indicación del usuario, o incluso de manera automatizada al cumplirse condiciones descritas por el propio código programable. Además las lógicas FPGA pueden programarse en memoria no volátil del módulo para la configuración del módulo automáticamente en el arranque.

10

Preferentemente, el sistema comprende una interfaz de usuario para la programación de la sección dinámica, es decir, para el desarrollo del código de usuario que contiene las instrucciones o funciones para personalizar el instrumento de control, test y medida. A través de la interfaz de usuario, el usuario introduce instrucciones o funciones de alto nivel que son preferentemente convertidas a lógica FPGA para la sección dinámica en un servidor de implementación. De acuerdo con opciones preferentes, el servidor puede estar integrado en un mismo software junto con la interfaz de usuario, o bien estar integrado en un software separado que puede correr localmente o de forma remota, proporcionando así servicios de implementación en la nube.

20

Preferentemente, el sistema comprende uno o más perfiles de hardware estáticos para cada instrumento. Los perfiles de hardware estáticos consisten en un perfil estático para la interfaz de usuario y la correspondiente lógica FPGA estática e interfaz lógica. Preferentemente el perfil estático de la interfaz de usuario consiste en una descripción simplificada de la lógica FPGA estática asociada, formado por bloques que describen las funcionalidades de la lógica FPGA estática e incluyen los puertos de la interfaz lógica entre la lógica FPGA estática y dinámica. La interfaz de usuario permite seleccionar entre los perfiles estáticos disponibles para cada instrumento sobre los cuales el usuario realiza la programación de las nuevas funcionalidades. Los diferentes perfiles de hardware incluyen funcionalidades en la sección estática implementadas en fábrica orientadas a aplicaciones específicas de control, test y medida del instrumento y que por su complejidad requieren una implementación y optimización realizada por expertos con herramientas avanzadas.

25

Preferentemente, la interfaz de usuario comprende además de buses estándar, buses adaptados específicamente a las características del hardware y de los datos generados por

35

5 cada instrumento. Los buses son grupos de señales y datos que permiten comunicar acciones o datos complejos entre funciones sin necesidad de conocer ni manipular expresamente e individualmente los datos y señales que componen el bus. Asimismo, el sistema comprende preferentemente al menos una librería con modelos predefinidos que el usuario puede utilizar para la programación de la sección dinámica. Estas librerías al igual que los perfiles estáticos incluyen de forma nativa interfaces maestro y esclavo para los buses específicos, procesando y manipulando las señales y datos complejos del instrumento de forma transparente para el usuario, de forma de que éste puede programar nuevas funcionalidades de forma simple, sin tener que implementar técnicas complejas de procesado y gestión de señales diferentes según cada instrumento.

15 Preferentemente, la interfaz de usuario proporciona funcionalidades adicionales al usuario, tales como validación de código y/o cálculo de latencias, información que preferentemente es suministrada al usuario durante el desarrollo del código. De acuerdo con opciones preferentes de implementación, el interfaz de usuario puede ser un interfaz gráfico, un interfaz de texto, o un interfaz que combina ambas modalidades.

20 En un segundo aspecto de la invención se presenta un método de configuración de hardware de un instrumento programable de control, test y medida, comprendiendo dicho instrumento una FPGA integrada con una sección dinámica y una sección estática conectadas a través de una interfaz lógica. El método comprende los siguientes pasos:

- Ejecutar en la sección dinámica una lógica FPGA dinámica desarrollada por un usuario. La configuración de la sección dinámica de acuerdo con dicha lógica es independiente de la configuración de la sección estática de la FPGA, por lo que puede realizarse mientras la lógica FPGA estática sigue operativa. Además la sección dinámica puede a su vez dividirse en subsecciones de forma que una subsección puede reconfigurarse mientras las otras siguen operativas. Preferentemente, la lógica FPGA dinámica se genera en un servidor de implementación a partir del código de usuario de alto nivel desarrollado por el usuario a través de una interfaz de usuario. El servidor puede ejecutarse local o remotamente de acuerdo con opciones preferentes del método. Asimismo, la interfaz puede ser gráfica, textual, o combinar ambas opciones, así como incorporar librerías de componentes predefinidos para la sección dinámica.

35

- Preferentemente, el método puede comprender seleccionar entre una pluralidad de lógicas FPGA estáticas prefijadas implementadas de fábrica para reconfigurar la sección estática. Nótese que cambiar entre lógicas FPGA estáticas, denominados perfiles de hardware del instrumento, implica la reconfiguración de la sección estática, pero al estar previamente implementadas en lógica FPGA de fábrica, se minimiza el tiempo en el que el instrumento permanece inoperativo, al mismo tiempo que se aumenta la versatilidad del mismo sin comprometer su funcionamiento o rendimiento. La sección estática puede dividirse en subsecciones de forma que una subsección estática puede reconfigurarse mientras las otras permanecen operativas.
- También preferentemente, el método puede comprender la configuración en la sección dinámica de múltiples lógicas FPGA dinámicas desarrolladas por el usuario, siendo dichas lógicas intercambiables por instrucción explícita del usuario o por el propio comportamiento automatizado del instrumento. Estas lógicas pueden almacenarse en el ordenador del usuario, en una memoria volátil del instrumento o en una memoria no volátil del mismo instrumento para su configuración automática en el arranque.
- Preferentemente, el método comprende la utilización de buses estándar y buses específicos a cada instrumento para la interconexión de las funciones que utiliza el usuario, de forma que interconexiones complejas se realicen sin necesidad de conocer los detalles internos de las señales y datos en el bus, es decir como una única señal. Las funciones de las librerías y las interfaces con la lógica FPGA estática incluyen interfaces maestro y esclavo para estos buses de forma que procesan y manipulan estos buses de datos complejos de forma transparente para el usuario.
- Preferentemente, el método comprende comprobar en tiempo real durante la programación del código de usuario la validez de las conexiones, permitiendo solo aquellas conexiones válidas y sugiriendo al programador acciones para realizar las conexiones correctamente.
- Preferentemente, el método comprende también calcular de manera automatizada latencias asociadas al código de usuario y preferentemente mostrar esta información durante el desarrollo del mismo para acelerar el proceso y simplificar un factor muy



importante en la programación de hardware FPGA, sin necesidad de realizar simulaciones o implementar la lógica FPGA y correrla en hardware, procesos que suelen requerir bastante tiempo sobretodo el de implementación.

- 5           – Preferentemente, el método comprende validar localmente el código programable desarrollado antes de iniciar la fase de implementación de la lógica FPGA dinámica la cual suele requerir un tiempo importante. Es decir, el propio software de diseño permite validar el código desarrollado por el usuario para verificar parámetros generales de funcionamiento, interconexiones entre funciones de usuario y con la
- 10           lógica FPGA estática, recursos utilizados y otros parámetros que permiten detectar problemas en una fase temprana antes de iniciar la implementación en la FPGA.

El método y el sistema de la invención logran independizar la etapa de generación del hardware FPGA de usuario o lógica FPGA dinámica, de la lógica FPGA estática prefijada

15           implementada de fábrica que proporciona las funcionalidades esenciales –core functions, en inglés- del instrumento y sobre las cuales el usuario desarrolla la aplicación. Así se implementan y optimizan por separado la lógica FPGA estática prefijada de fábrica y la lógica FPGA dinámica de usuario, ya que la lógica de fábrica estática no es modificada durante la personalización del módulo por parte del usuario. Esto permite reducir

20           significativamente los tiempos y la dificultad de implementación de la lógica FPGA dinámica desarrollada por el usuario, la cual como consecuencia puede desarrollarse con herramientas simplificadas utilizables por usuarios sin experiencia en programación de FPGA. Mientras a su vez se garantiza que en el proceso de personalización del instrumento las funcionalidades de fábrica no se verán afectadas durante la implementación de la lógica

25           FPGA dinámica. La lógica FPGA de fábrica proporciona las funcionalidades esenciales del módulo, así como funcionalidades adicionales complejas importantes para ciertas aplicaciones del instrumento, las cuales son desarrolladas y optimizadas en fábrica con técnicas complejas y herramientas avanzadas de bajo nivel que permiten hacer un uso muy eficiente de los recursos de la FPGA y lograr el máximo rendimiento, sin lo cual muchas de

30           estas funcionalidades muy demandantes no pueden implementarse. Adicionalmente, la separación estático-dinámico permite mantener la lógica FPGA estática completamente protegida y oculta del usuario, salvaguardando el know-how de los desarrolladores. Finalmente, el uso de reconfiguración parcial de la FPGA para la sección dinámica o estática permite que el usuario re programe parte de la FPGA mientras otras partes siguen

funcionando en la aplicación, e incluso que una parte del sistema por hardware dispere la reconfiguración de ella misma u otra parte.

### **BREVE DESCRIPCIÓN DE LAS FIGURAS**

5

A continuación se pasa a describir de manera muy breve una serie de dibujos que ayudan a comprender mejor la invención y que se relacionan expresamente con una realización de dicha invención que se presenta como un ejemplo no limitativo de ésta.

10 FIGURA 1.-. Muestra un diagrama de bloques de la arquitectura de un instrumento programable de control, test y medida según una posible realización de la invención.

FIGURA 2.-. Presenta un diagrama de bloques de un sistema de configuración de instrumentos programables, así como de la información intercambiada entre dichos bloques,  
15 según una posible realización de la invención.

FIGURA 3.-. Ejemplifica una posible realización de la interfaz de usuario de la invención.

### **REALIZACIÓN PREFERENTE DE LA INVENCION**

20

En este texto, el término "comprende" y sus derivaciones (como "comprendiendo", etc.) no deben entenderse en un sentido excluyente, es decir, estos términos no deben interpretarse como excluyentes de la posibilidad de que lo que se describe y define pueda incluir más elementos, etapas, etc.

25

Dado que el proceso de implementación de una FPGA consiste en la generación de la información necesaria para determinar las interconexiones y el posicionamiento de los componentes lógicos internos de dicha FPGA, en el presente texto, el término 'lógica FPGA' debe entenderse como un conjunto de información que define dichas interconexiones y  
30 posicionamientos. En particular, y como se detalla a lo largo de la descripción, una 'lógica FPGA estática' se refiere a un conjunto de interconexiones y posicionamientos de componentes internos de la FPGA prefijado implementado de fábrica, mientras que una 'lógica FPGA dinámica' se refiere a un conjunto de interconexiones y posicionamientos de componentes internos de la FPGA resultantes de implementar el código de usuario.

35

La Figura 1 ejemplifica un instrumento programable (10) de control, test y medida sobre el que se aplican implementaciones particulares del método y sistema de la invención. En una implementación preferente, el instrumento programable (10) es un instrumento modular de una plataforma PXI express (extensiones PCI para instrumentación, del inglés 'PCI eXtensions for Instrumentation') o PCI express (Interconexión de Componentes Periféricos Express, del inglés 'Peripheral Component Interconnect Express'), si bien pueden existir implementaciones en cualquier otra plataforma para instrumentos de control, test y medida compatibles con la tecnología FPGA.

El instrumento programable (10) comprende un hardware fijo (20) gestionado por una FPGA (30). La FPGA (30) comprende uno o más puertos de entrada y/o salida (50), así como un bus de comunicación (40), que puede venir acompañado además de puertos para cualquier otra señal requerida por la FPGA (30), como por ejemplo señales de reloj, etc. Nótese que el método y sistema de configuración de hardware del instrumento programable (10) actúan únicamente sobre la FPGA (30), siendo por lo tanto independientes del resto de componentes del instrumento programable (10).

La FPGA comprende una sección estática (31) y una sección dinámica (33), conectadas a través de una interfaz lógica (32). La sección estática (31) comprende lógica FPGA estática (34) implementada y optimizada de fábrica, mientras que la sección dinámica (33) comprende lógica FPGA dinámica (35) programable por el usuario. La lógica FPGA estática (34) comprende las funcionalidades esenciales -core functions, en inglés- del instrumento y aquéllas cuya implementación a bajo nivel y optimización avanzada resultan críticas para el funcionamiento del instrumento programable (10).

La lógica FPGA estática (34) implementa todas las funciones necesarias para controlar el hardware fijo (20), como por ejemplo: comunicación PCIe (Interconexión de Componentes Periféricos Express, del inglés 'Peripheral Component Interconnect Express'), DMA (memoria de acceso directo, del inglés 'Direct Memory Access'), memoria RAM (memoria de acceso aleatorio, del inglés 'Random Memory Access) y no volátil, conversores ADC (Convertor de señales analógicas a digitales, del inglés 'Analogue to Digital converter) y DAC (Convertor de señales digitales a analógicas, del inglés Digital to Analogue converter), etc.

La lógica FPGA estática (34) también implementa funciones importantes para el funcionamiento específico del instrumento. Por ejemplo en un instrumento de generación

analógica de señales, la lógica FPGA estática (34) puede comprender funcionalidades de generación de señales periódicas, generación de señales arbitrarias, moduladores, filtros, etc.

- 5 La lógica FPGA estática (34) implementa las funcionalidades que suelen tener requerimientos de performance muy estrictos y que requieren para su implementación el uso de herramientas avanzadas complejas, como por ejemplo interfaces de memoria DDR3 con tasas de transferencia de decenas de GigaBytes por segundo o interfaces PCI Express de varios GigaBytes por segundo. Este tipo de funcionalidades requiere de la utilización de  
10 frecuencias de trabajo y tamaños de datos muy elevados y de lógica muy compleja que requiere de un análisis experto a bajo nivel para lograr su correcto funcionamiento.

La figura 2 presenta con mayor detalle los elementos del sistema involucrados en la programación de la sección dinámica (33). El usuario introduce las instrucciones y funciones  
15 de alto nivel que forman el código de usuario (61) a través de una interfaz de usuario (60). La interfaz de usuario (60) comprende un perfil estático (83) sobre el cual el usuario implementa el código de usuario (61). Preferentemente el usuario selecciona de una librería de perfiles estáticos (82a) el que desea utilizar según las opciones de cada instrumento específico. La librería de perfiles estáticos (82a) forma parte de la librería de perfiles de  
20 hardware (82) que además incluye la librería de lógica FPGA estática (82b), que incluye una lógica FPGA estática (34) y la interfaz lógica (32) correspondientes a cada perfil estático (83). La librería de perfiles de Hardware (82) incluye uno o más perfiles de hardware para cada instrumento. Cada perfil de hardware está compuesto por el perfil estático (83) y su correspondiente lógica FPGA estática (34) que a su vez define una interfaz lógica (32). El  
25 perfil estático (83) consiste en una descripción simplificada de la lógica FPGA estática, que captura sus funcionalidades e incorpora la interfaz lógica (32) para que el usuario implemente el código de usuario (61) y lo conecte a las funcionalidades provistas por lógica FPGA estática (34). Dicho código de usuario (61) es transmitido a un servidor de implementación (70), que lo combina con unos modelos (81) de una o más librerías (80)  
30 para generar la lógica dinámica (35) que se ejecuta en la sección dinámica (33). El servidor utiliza información de la lógica FPGA estática (34) y de la correspondiente interfaz lógica (32) para la implementación de la lógica FPGA dinámica (35) compatible con la lógica FPGA estática (34) según el perfil de hardware seleccionado. Los modelos (81) de las librerías (80) facilitan al usuario la programación de la sección dinámica (33), pero al contrario que la  
35 lógica FPGA estática, son personalizables. La lógica FPGA estática (34) seleccionada con el

perfil de hardware se configura en la sección estática (31) de la FPGA. La lógica FPGA estática y dinámica se configuran en la FPGA de manera independiente, de forma que pueden programarse múltiples lógicas FPGA dinámicas para una misma lógica FPGA estática mientras la lógica FPGA estática y el resto del instrumento siguen funcionando, 5 posibilitando el cambio muy rápido de lógicas FPGA dinámicas desarrolladas por el usuario.

El servidor de implementación (70) puede estar integrado en el mismo software de la interfaz de usuario (60), o ser tanto un servidor local corriendo en el mismo ordenador que la interfaz de usuario, como un servidor remoto, conectado al software de la interfaz de usuario (60) 10 mediante medios de conexión alámbricos o inalámbricos. En particular, el servidor remoto proporciona servicios de implementación en la nube.

La figura 3 ejemplifica la interfaz de usuario (60) para el desarrollo del código de usuario (61) para la programación de la sección dinámica (33). En este caso, se trata de una interfaz 15 gráfica basada en bloques interconectados. En la pantalla principal se aprecian tres tipos de bloques:

- Bloques estáticos (62), correspondientes a funcionalidades de la lógica FPGA estática (34). Los bloques estáticos (62) son parte del perfil estático (83), el usuario 20 por lo tanto no modifica dichos bloques estáticos (62), sino que utiliza sus puertos lógicos para conectarlos con el resto de bloques de su diseño. Los puertos lógicos de los bloques estáticos (62) conforman la interfaz lógica (32) de la FPGA. Nótese que los bloques estáticos (62) no son necesariamente una representación fidedigna de la implementación del código de fábrica, es decir, pueden suponer una vista 25 simplificada que no refleja la complejidad de la implementación real de las funcionalidades estáticas representadas.
- Bloques dinámicos (63), cuyo contenido viene predefinido, por ejemplo a partir de los modelos (81) de la librería (80), que el usuario utiliza para realizar las 30 funcionalidades de la lógica dinámica (35), y por tanto pueden ser seleccionados, modificados y/o eliminados por parte del usuario
- Bloques de usuario (64), programados en su totalidad por el usuario. Los bloques de usuario (64) pueden ser implementados utilizando la interfaz de usuario (60) o ser 35 importados desde cualquier otra herramienta de programación de hardware FPGA.

Para facilitar las labores de programación del usuario, la interfaz de usuario (60) puede comprender un código dinámico por defecto que implementa las funcionalidades de fábrica que caracterizan al instrumento y que hacen que este sea completamente funcional y que sirve al usuario como base para el desarrollo de su código. Además la interfaz de usuario (60) puede comprender otros elementos, como por ejemplo una lista de bloques dinámicos incluidos por defecto de fábrica eliminados (65), o de puertos de conexión disponibles (66), así como cualquier otro elemento que simplifique el uso del software y el desarrollo del código de usuario. El interfaz de usuario (60) puede comprender medios de selección que permiten al usuario seleccionar de entre múltiples perfiles estáticos (83) disponibles para el instrumento específico, así como configurar diferentes lógicas dinámicas (35) y estáticas (34) en el instrumento. Diferentes perfiles estáticos (83) y su correspondiente lógica FPGA estática (34) para un mismo instrumento incluyen distintos bloques de alto rendimiento importantes para un grupo de aplicaciones, que requieren de una implementación con conocimientos muy avanzados y que de incluirlos como librerías para la zona dinámica dificultan su implementación y comprometen el correcto funcionamiento del instrumento programable (10).

La interfaz de usuario comprende buses estándar y también buses específicos para cada instrumento, que agrupan señales y datos complejos en una única conexión para simplificar y acelerar el desarrollo del código por parte del usuario. Entre los buses estándar se incluyen conexiones AXI, acceso a memoria, etc, mientras que los buses específicos dependen de los datos que genera y procesa cada instrumento. Así por ejemplo módulos de adquisición o generación de señales de varias centenas de megahercios o gigahercios requieren la utilización de técnicas de paralización de datos para poder ser manipulados dentro de la FPGA, estas técnicas complican extremadamente la programación de los instrumentos. Mediante el uso de buses específicos el usuario puede manipular los datos sin conocer esta complejidad ya que tanto los bloques de fábrica como los de librería incluyen interfaces específicas para estos buses y manipulan estos datos complejos de forma nativa y transparente para el usuario como una única señal.

La interfaz de usuario (60) puede comprender funcionalidades adicionales como validación de código o cálculo de latencias. Con la información de latencia de cada bloque, el software permite visualizar rápidamente durante la edición los retrasos en las diferentes conexiones del diagrama, así como ajustar dichos retrasos de forma muy sencilla en diagramas con múltiples bloques.

La interfaz de usuario (60) puede comprender funciones de simulación del código de usuario que incorporen modelos de los bloques estáticos y dinámicos y que permitan realizar simulaciones de comportamiento y visualizar las señales en los diferentes puntos del diagrama. Además la interfaz de usuario (60) puede permitir generar un instrumento virtual que el sistema reconoce como un instrumento real, de forma que el software de usuario puede interactuar con el instrumento virtual como si fuese uno real con la lógica FPGA de usuario ya implementada y configurada en el instrumento. De esta forma se puede simular el funcionamiento del instrumento dentro de la aplicación de control, test y medida generando los estímulos reales que recibirá el instrumento real pero con mayor control y posibilidad de visualizar toda la información de simulación de cualquier punto del diagrama sin necesidad de implementar la lógica de usuario y la lógica adicional de depuración en el instrumento real, lo cual suele requerir del tiempo que requiere la implementación de la lógica FPGA y además las capacidades de depuración en hardware suelen tener más limitaciones que las simulaciones.

15

La interfaz de usuario (60) puede comprender funcionalidades de depuración que permite agregar puntos para depuración o validación (debugging) por hardware, para los cuales el software genera la lógica de hardware necesaria automáticamente. Una vez generados dichos puntos de depuración, el software recibe del instrumento (10) información de depuración a través de alguno de los buses de comunicaciones disponible en el instrumento (por ejemplo PCI Express), y la muestra a través de la interfaz de usuario (60). La sección estática (31) puede comprender perfiles estáticos específicos con funcionalidades específicas para realizar dicha depuración.

25

Una vez el usuario ha finalizado su diseño a través de la interfaz de usuario (60), el software genera la lógica FPGA dinámica (35) correspondiente únicamente a la sección dinámica (33) y a la interfaz lógica (32) que define las interconexiones entre la sección dinámica (33) y la lógica FPGA estática (34). El software primero procesa y valida el código de usuario (61) de forma local y luego lo envía al servidor de implementación (70) para la generación de la lógica dinámica (35) y el correspondiente fichero de reconfiguración parcial de la sección dinámica (33). Antes de proceder a la implementación en el servidor de implementación (70), el software valida localmente la correcta realización del código de usuario, su interconexión con la lógica FPGA estática y verifica que los recursos utilizados están disponibles en la sección dinámica (33). El servidor de implementación (70) une el código de usuario ya procesado localmente con los posibles ficheros de conexiones correspondientes

35

a los bloques de las librerías (80), generando un fichero de configuración parcial para la sección dinámica (33) compatible con la lógica FPGA estática (34), que refleja el posicionamiento e interconexión de la lógica FPGA dinámica (35). En el proceso de implementación solo se genera la lógica FPGA dinámica (35), por lo que el tiempo y dificultad de implementación son muy inferiores al de regenerar la lógica FPGA (30) completa y además no se modifica en lo absoluto la lógica FPGA estática (34) de la sección estática (31) por lo que se mantiene el rendimiento y las optimizaciones de todas las funcionalidades de fábrica. Con el fichero de configuración parcial, el usuario puede reprogramar la sección dinámica (33) de la FPGA (30) en cualquier momento mientras que la lógica estática y el resto del módulo siguen funcionando.

La gestión de la carga de la nueva configuración parcial de la FPGA (30), es decir, la configuración de la lógica dinámica (35) del usuario en la sección dinámica (33) se puede realizar desde la propia interfaz de usuario (60) o, por medio de las librerías suministradas, desde cualquier otra aplicación software. De esta forma, se puede disponer de diferentes configuraciones dinámicas previamente generadas y cargarlas dinámicamente a medida que el software de la aplicación de control, test y medida requiere diferentes procesados por hardware. La información de reconfiguración parcial puede programarse en la FPGA (30) directamente, en una memoria volátil del instrumento o en una memoria no volátil incluida en el instrumento para su carga automática en el arranque. Además, se pueden guardar más de una reconfiguración en memoria volátil o no volátil, permitiendo modificar dinámicamente la selección o configuración del hardware (20) del instrumento programable (10).

La carga de las diferentes lógicas FPGA estáticas correspondiente a los diferentes perfiles estáticos (83) puede realizarse de manera similar a la configuración de la lógica FPGA dinámica, desde la interfaz de usuario (60) o por medio de librerías suministradas. Además tanto la sección dinámica (33) como la sección estática (31) pueden dividirse en subsecciones de forma que las subsecciones pueden reprogramarse mientras las otras siguen funcionando.

A la vista de esta descripción y figuras, el experto en la materia podrá entender que la invención ha sido descrita según algunas realizaciones preferentes de la misma, pero que múltiples variaciones pueden ser introducidas en dichas realizaciones preferentes, sin salir del objeto de la invención tal y como ha sido reivindicada.



## REIVINDICACIONES

1. Un sistema de configuración de hardware de un instrumento programable (10) de control, test y medida que comprende una FPGA (30) integrada, **caracterizado por que** la FPGA (30) comprende:
- 5 (30) comprende:
- una sección estática (31) que comprende al menos una lógica FPGA estática (34) prefijada;
  - una sección dinámica (33) que comprende al menos una lógica FPGA dinámica (35) programable por un usuario;
  - 10 -una interfaz lógica (32) que conecta la sección estática (31) y la sección dinámica (33).
2. El sistema de configuración, de acuerdo con la reivindicación 1, **caracterizado por que** la sección estática (31) comprende una pluralidad de lógicas FPGA estáticas (34) prefijadas
- 15 seleccionables por el usuario y reconfigurables durante la operación del instrumento programable (10).
3. El sistema de configuración, de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado por que** la sección dinámica (33) comprende una pluralidad de lógicas
- 20 dinámicas (35) programables por el usuario y reconfigurables durante la operación del instrumento programable (10) y mientras la sección estática (31) sigue funcionando.
4. El sistema de configuración, de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado por que** la sección estática (31) y la sección dinámica (33) comprenden una
- 25 pluralidad de subsecciones y al menos una de las subsecciones es reconfigurable individualmente por selección del usuario mientras las restantes subsecciones siguen funcionando.
5. El sistema de configuración, de acuerdo con cualquiera de las reivindicaciones anteriores,
- 30 **caracterizado por que** comprende al menos un perfil de hardware para cada instrumento (10), el perfil de hardware comprendiendo la lógica FPGA estática (34) y la interfaz lógica (32) implementadas y optimizadas en fábrica para adaptar el instrumento (10) a requerimientos específicos de distintas aplicaciones de control, test y medida.

6. El sistema de configuración, de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado por que** comprende almacenar una o más lógicas FPGA estáticas (34) y lógicas FPGA dinámicas (35) en una memoria del instrumento (10), que se selecciona entre memoria volátil y no volátil.

5

7. El sistema de configuración, de acuerdo con la reivindicación 6, **caracterizado por que** en caso de memoria no volátil, las lógicas FPGA estáticas (34) y lógicas FPGA dinámicas (35) almacenadas son utilizadas para reconfigurar el instrumento (10) automáticamente en el arranque.

10

8. El sistema de configuración, de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado por que** comprende una interfaz de usuario (60) a través de la que el usuario introduce instrucciones y funciones de alto nivel que forman un código de usuario (61) para programar la sección dinámica (33) y conectarla con funciones de la lógica FPGA estática (34) a través de la interfaz lógica (32).

15

9. El sistema de configuración, de acuerdo con la reivindicación 8, **caracterizado por que** la interfaz de usuario (60) comprende al menos un perfil estático (83) para cada instrumento (10), asociado cada perfil estático (83) a la lógica FPGA estática (34) y la interfaz lógica (32), y donde el perfil estático (83) describe las funcionalidades de la lógica FPGA estática (34) asociada, en forma de modelos estáticos (62) que incluyen unos puertos de la interfaz lógica (32) que el usuario utiliza para interactuar en el código de usuario (61) con la lógica FPGA estática (34).

20

10. El sistema de configuración, de acuerdo con cualquiera de las reivindicaciones 8-9, **caracterizado por que** la interfaz de usuario (60) valida localmente el código de usuario (61) antes de ser enviado al servidor de implementación (70) para ser implementado en lógica FPGA dinámica (35) para la sección dinámica (33).

25

11. El sistema de configuración, de acuerdo con cualquiera de las reivindicaciones 8-10, **caracterizado por que** la interfaz de usuario (60) calcula unas latencias asociadas a los modelos utilizados en el código de usuario (61) y visualiza esta información durante el desarrollo y antes de ser implementado en una lógica FPGA dinámica (35) para la sección dinámica (33).

30

35

12. El sistema de configuración, de acuerdo con cualquiera de las reivindicaciones 8-11, **caracterizado por que** la interfaz de usuario (60) comprende funciones específicas de depuración del código de usuario (61) y lee información de depuración a través de unos puertos de comunicación del instrumento (10) y visualiza la información en tiempo real en la misma interfaz de usuario (60).
13. El sistema de configuración, de acuerdo con cualquiera de las reivindicaciones 8-12, **caracterizado por que** la interfaz de usuario (60) comprende funcionalidades de simulación para implementar un instrumento virtual que el sistema reconoce como real.
14. El sistema de configuración, de acuerdo con cualquiera de las reivindicaciones 8-13, **caracterizado por que** la interfaz de usuario (60) es una interfaz gráfica.
15. El sistema de configuración, de acuerdo con cualquiera de las reivindicaciones 8-13, **caracterizado por que** la interfaz de usuario (60) es una interfaz de texto.
16. El sistema de configuración, de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado por que** comprende buses de datos específicos adaptados a las características particulares de cada instrumento programable (10) y que agrupan los datos y señales complejas generadas por el instrumento y las representa como una única señal.
17. El sistema de configuración, de acuerdo con cualquiera de las reivindicaciones anteriores, **caracterizado por que** comprende al menos una librería (80) con una pluralidad de modelos (81) utilizables por el usuario para desarrollar el código de usuario (61) y que incluyen puertos de interfaz para los buses de datos específicos a cada instrumento que manipulan y procesan los múltiples datos y señales de los buses específicos como una única señal.
18. El sistema de configuración, de acuerdo con cualquiera de las reivindicaciones anteriores **caracterizado por que** comprende un servidor de implementación (70) que convierte las instrucciones y funciones de alto nivel que forman el código de usuario (61) en lógica FPGA dinámica (35) compatible con la lógica FPGA estática (34) y la interfaz lógica (32), y el servidor de implementación (70) genera información de configuración parcial para programar la sección dinámica (33).

19. El sistema de configuración, de acuerdo con la reivindicación 18, **caracterizado por que** el servidor de implementación (70) está integrado en un software de edición que comprende el interfaz de usuario (60).
- 5 20. El sistema de configuración, de acuerdo con la reivindicación 18, **caracterizado por que** el servidor de implementación (70) está integrado en un software remoto.
21. El sistema de configuración, de acuerdo con la reivindicación 20, **caracterizado por que** el software remoto proporciona servicios de implementación en la nube.
- 10
22. Método de configuración de un instrumento programable (10) de control, test y medida que comprende una FPGA (30) integrada, **caracterizado por que** comprende ejecutar una lógica FPGA dinámica (35) programada por un usuario en una sección dinámica (33) de la FPGA (30), estando dicha sección dinámica (33) conectada a través de una interfaz lógica (32) a una sección estática (31) que comprende al menos una lógica FPGA estática (34) implementada y prefijada.
- 15
23. El método de configuración, de acuerdo con la reivindicación 22, **caracterizado por que** comprende recibir una selección del usuario entre una pluralidad de perfiles de hardware que comprenden lógicas FPGA estáticas implementadas y prefijadas configurables en la sección estática (31) e intercambiables durante la operación del instrumento (10) programable para adaptarlo a diferentes requerimientos de aplicaciones específicas de control, test y medida.
- 20
24. El método de configuración, de acuerdo con cualquiera de las reivindicaciones 22-23, **caracterizado por que** comprende configurar en la sección dinámica (32) una pluralidad de lógicas FPGA dinámicas (35) intercambiables durante la operación del instrumento programable (10) y mientras la lógica FPGA estática (34) sigue funcionando.
- 25
25. El método de configuración, de acuerdo con cualquiera de las reivindicaciones 22-24, **caracterizado por que** comprende recibir instrucciones y funciones de alto nivel que forman un código de usuario (61) para programar la sección dinámica (33) a través de una interfaz de usuario (60).
- 30
26. El método de configuración, de acuerdo con la reivindicación 25, **caracterizado por que** comprende convertir las instrucciones y funciones de alto nivel del código de usuario (61) en
- 35

lógica FPGA dinámica (35) para la sección dinámica (33) compatible con la lógica FPGA estática (34) en un servidor de implementación (70).

5 27. El método de configuración, de acuerdo con la reivindicación 26, **caracterizado por que** el paso de convertir las instrucciones y funciones de alto nivel del código de usuario (61) en lógica FPGA dinámica (35) se ejecuta en un servidor (70) integrado en un software de edición que comprende el interfaz de usuario (60).

10 28. El método de configuración, de acuerdo con la reivindicación 26, **caracterizado por que** el paso de convertir las instrucciones y funciones de alto nivel del código de usuario (61) en lógica FPGA dinámica (35) se ejecuta en un servidor (70) remoto.

15 29. El método de configuración, de acuerdo con cualquiera de las reivindicaciones 22-28, **caracterizado por que** comprende validar localmente el código de usuario (61) antes de ser implementado en lógica FPGA dinámica (35) para la sección dinámica (33).

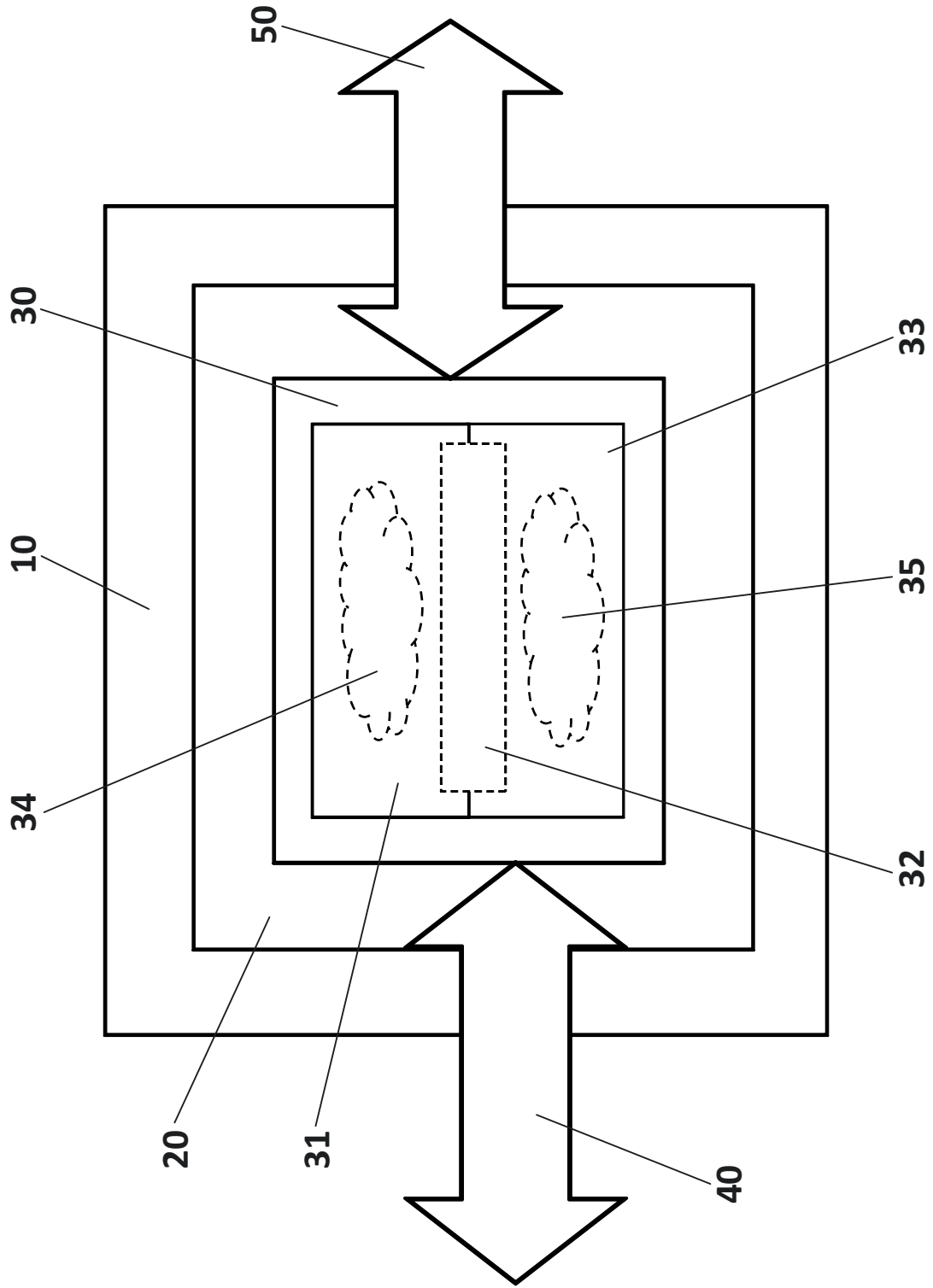
20 30. El método de configuración, de acuerdo con cualquiera de las reivindicaciones 22-29, **caracterizado por que** comprende calcular unas latencias asociadas al código de usuario (61) antes de ser implementado en la lógica FPGA dinámica (35) para la sección dinámica (33).

25 31. El método de configuración, de acuerdo con cualquiera de las reivindicaciones 22-30, **caracterizado por que** comprende recibir las instrucciones y funciones de alto nivel (61) a través de una interfaz gráfica.

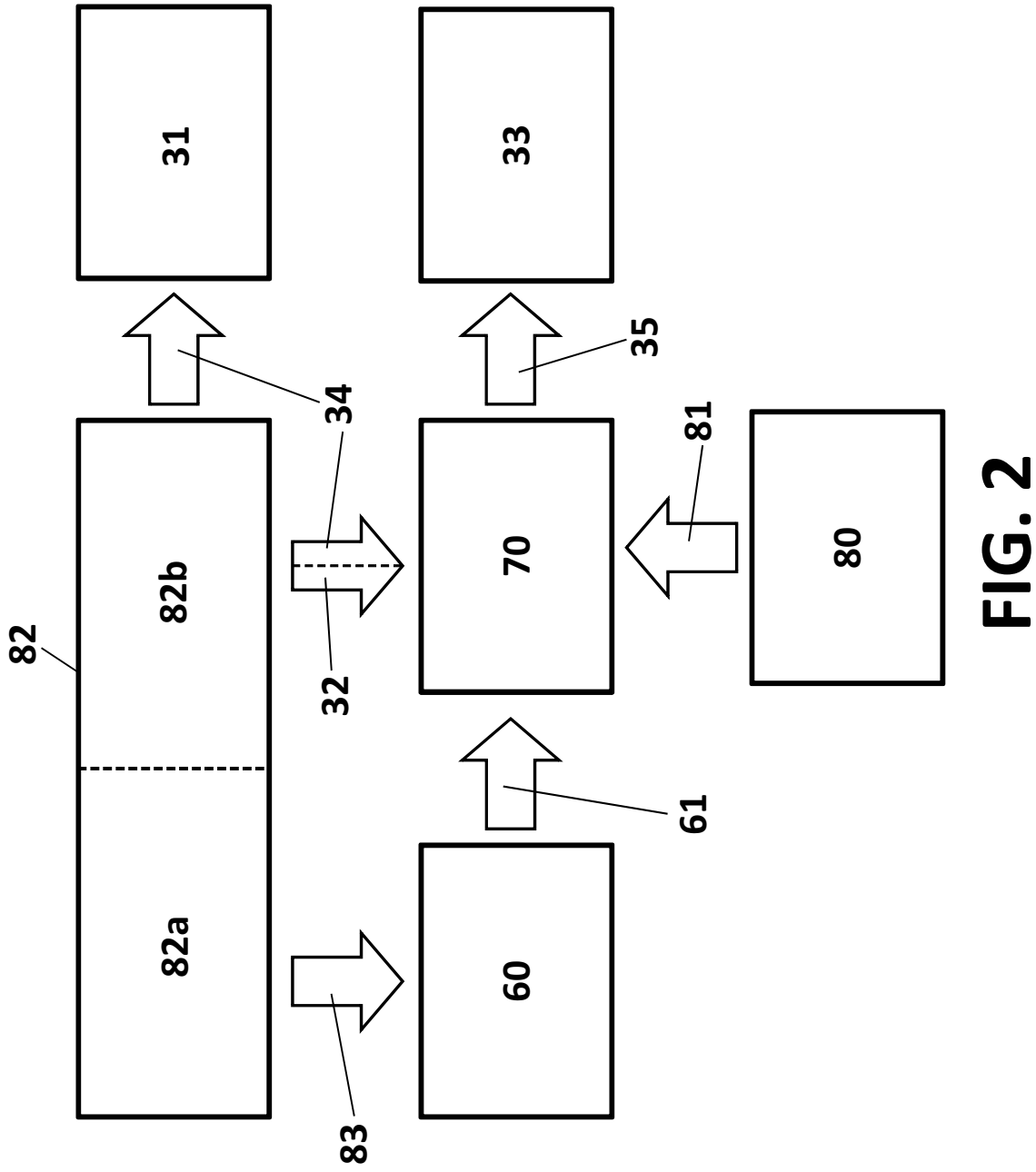
32. El método de configuración, de acuerdo con cualquiera de las reivindicaciones 22-31, **caracterizado por que** comprende recibir las instrucciones y funciones de alto nivel del código de usuario (61) a través de una interfaz de texto.

30 33. El método de configuración, de acuerdo con cualquiera de las reivindicaciones 22-32, **caracterizado por que** comprende utilizar buses de datos específicos para cada instrumento de manera que los datos y señales complejas contenidos en el bus se manipulan e interconectan entre funciones como una única señal.

34. El método de configuración, de acuerdo con cualquiera de las reivindicaciones 22-33, **caracterizado por que** comprende generar la lógica FPGA dinámica (35) usando al menos una librería (80) con una pluralidad de modelos (81), que incluyen interfaces para unos buses de cada instrumento (10) y que procesan las señales y datos de los buses como una
- 5 única señal.



**FIG. 1**





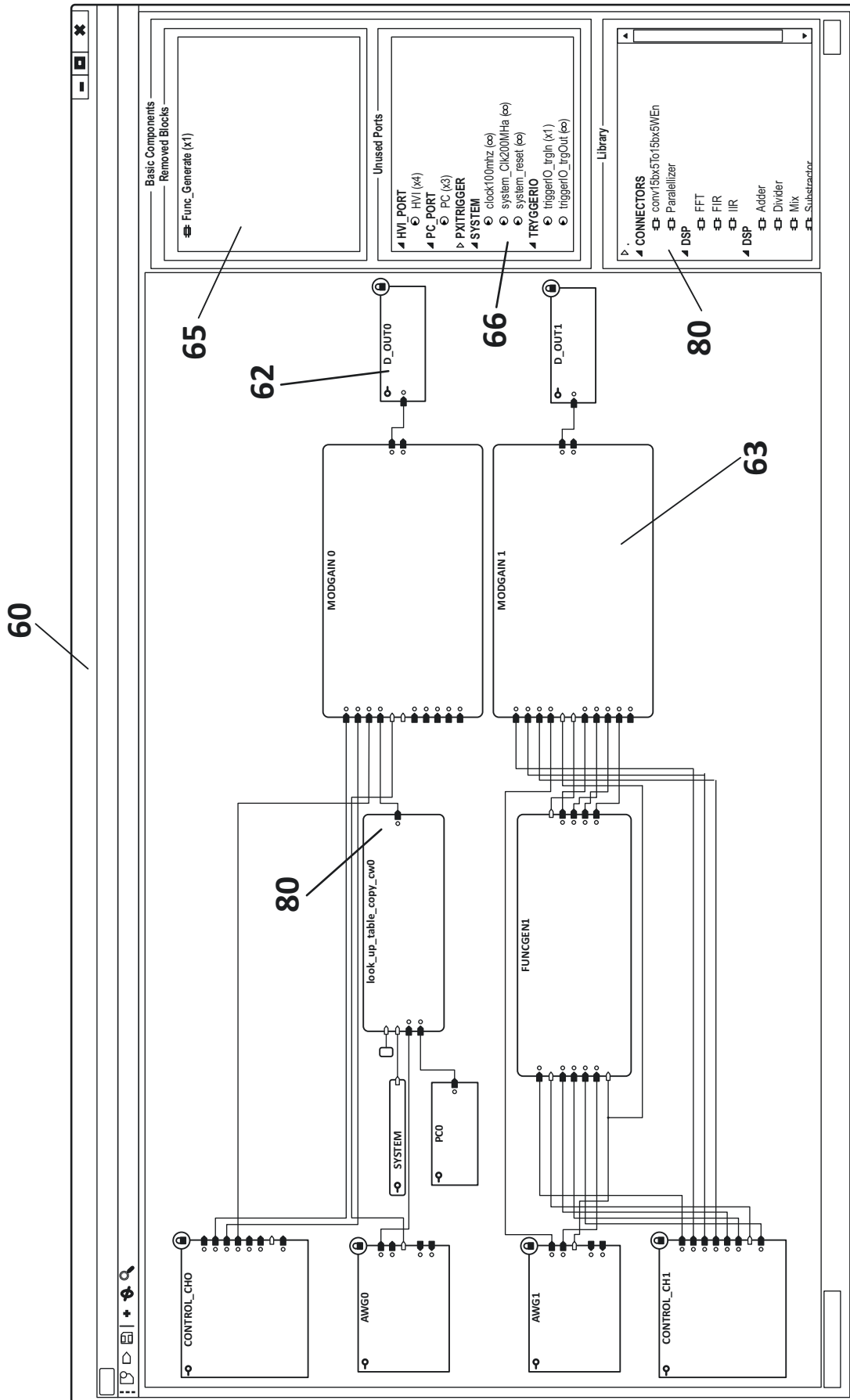


FIG. 3



OFICINA ESPAÑOLA  
DE PATENTES Y MARCAS

ESPAÑA

②① N.º solicitud: 201531184

②② Fecha de presentación de la solicitud: 10.08.2015

③② Fecha de prioridad:

INFORME SOBRE EL ESTADO DE LA TECNICA

⑤① Int. Cl.: **G06F17/50** (2006.01)

DOCUMENTOS RELEVANTES

Categoría	⑤⑥ Documentos citados	Reivindicaciones afectadas
X	US 8997033 B1 (HEW YIN CHONG) 31.03.2015, descripción: columna 2, línea 50 – columna 6, línea 3; figura 4.	1-34
A	WO 2009033630 A1 (UNIV FRIEDRICH ALEXANDER ER et al.) 19.03.2009, todo el documento.	1-34
A	US 7640526 B1 (BLODGET BRANDON J et al.) 29.12.2009, todo el documento.	1-34
A	US 7902866 B1 (PATTERSON CAMERON et al.) 08.03.2011, todo el documento.	1-34
A	US 2003038842 A1 (PECK JOSEPH E et al.) 27.02.2003, todo el documento.	1-34

Categoría de los documentos citados

X: de particular relevancia

Y: de particular relevancia combinado con otro/s de la misma categoría

A: refleja el estado de la técnica

O: referido a divulgación no escrita

P: publicado entre la fecha de prioridad y la de presentación de la solicitud

E: documento anterior, pero publicado después de la fecha de presentación de la solicitud

**El presente informe ha sido realizado**

para todas las reivindicaciones

para las reivindicaciones nº:

Fecha de realización del informe  
23.02.2016

Examinador  
M. Muñoz Sánchez

Página  
1/4

Documentación mínima buscada (sistema de clasificación seguido de los símbolos de clasificación)

G06F

Bases de datos electrónicas consultadas durante la búsqueda (nombre de la base de datos y, si es posible, términos de búsqueda utilizados)

INVENES, EPODOC, WPI

Fecha de Realización de la Opinión Escrita: 23.02.2016

**Declaración**

<b>Novedad (Art. 6.1 LP 11/1986)</b>	Reivindicaciones 5-7, 10-21, 23, 26-34	<b>SI</b>
	Reivindicaciones 1-4, 8-9, 22, 24-25	<b>NO</b>
<b>Actividad inventiva (Art. 8.1 LP11/1986)</b>	Reivindicaciones	<b>SI</b>
	Reivindicaciones 5-7, 10-21, 23, 26-34	<b>NO</b>

Se considera que la solicitud cumple con el requisito de aplicación industrial. Este requisito fue evaluado durante la fase de examen formal y técnico de la solicitud (Artículo 31.2 Ley 11/1986).

**Base de la Opinión.-**

La presente opinión se ha realizado sobre la base de la solicitud de patente tal y como se publica.

**1. Documentos considerados.-**

A continuación se relacionan los documentos pertenecientes al estado de la técnica tomados en consideración para la realización de esta opinión.

Documento	Número Publicación o Identificación	Fecha Publicación
D01	US 8997033 B1 (HEW YIN CHONG)	31.03.2015
D02	WO 2009033630 A1 (UNIV FRIEDRICH ALEXANDER ER et al.)	19.03.2009
D03	US 7640526 B1 (BLODGET BRANDON J et al.)	29.12.2009
D04	US 7902866 B1 (PATTERSON CAMERON et al.)	08.03.2011
D05	US 2003038842 A1 (PECK JOSEPH E et al.)	27.02.2003

**2. Declaración motivada según los artículos 29.6 y 29.7 del Reglamento de ejecución de la Ley 11/1986, de 20 de marzo, de Patentes sobre la novedad y la actividad inventiva; citas y explicaciones en apoyo de esta declaración**

Se considera D01 el documento más próximo del estado de la técnica al objeto de la solicitud.

**Reivindicaciones independientes**

Reivindicación 1: El documento D01, divulga un método de reconfiguración de regiones de circuitos integrados (FPGA, por ejemplo) haciéndose en su caso dicha reconfiguración en tiempo de ejecución. Las regiones que se reconfiguran se llaman dinámicas en oposición a las estáticas que no se reconfigurarán, entre dichas secciones o regiones existe implícita una interfaz lógica. Por otra parte se prevé el uso de un asistente de diseño electrónico (EDA) que facilita una interfaz de usuario para hacer la reconfiguración y con otras funcionalidades como la fusión de varios archivos de configuración (columna 2 línea 50 □ columna 6 línea 3, figura 4). Teniendo en cuenta que el método divulga los elementos del sistema reivindicado así como su funcionalidad e interacción el documento D01 afecta a la novedad de la reivindicación 1 según el art. 6.1 de la ley 11/86 de patentes.

Reivindicación 22: En consonancia con el análisis de la reivindicación 1 el documento D01 afecta a la novedad de la reivindicación 22 según el art. 6.1 de la ley 11/86 de patentes.

**Reivindicaciones dependientes**

Reivindicación 2-4: el contenido de estas reivindicaciones se recoge en el documento D01.

Reivindicaciones 8-9: el contenido de estas reivindicaciones se recoge en el documento D01.

Así el documento D01 afecta a la novedad de las reivindicaciones 2-4, 8-9 según el art. 6.1 de la ley 11/86 de patentes.

Reivindicaciones 5-7: la implementación y optimización en cumplimiento de unos requisitos o su ubicación en memorias volátiles o no volátiles se consideran opciones que el experto en la materia podría incluir en su caso en el método del documento D01 por ser comúnmente conocidas, y en consecuencia evidentes para él.

Reivindicaciones 10-21: las características relativas de implementación a través de servidor, el procesamiento de una señal única y las funcionalidades depuración, simulación, utilización de bibliotecas (librerías) se consideran opciones habituales del campo técnico de la solicitud y por tanto evidentes para el experto en la materia. Ilustrativamente, se citan los documentos D02, D03, D04 y D05.

Por tanto, el documento D01 afecta a la actividad inventiva de las reivindicaciones 5-7, 10-21 según el art. 8.1 de la ley 11/86 de patentes.

Reivindicaciones 23, 26-34: Dado el paralelismo entre el contenido de estas reivindicaciones y las reivindicaciones 5-7, 10-21 se concluye que el documento D01 afecta a la actividad inventiva de las reivindicaciones 23 y 26-34 según el art. 8.1 de la ley 11/86 de patentes.

Reivindicaciones 24-25: el contenido de estas el contenido de estas reivindicaciones se recoge en el documento D01. Así el documento D01 afecta a la novedad de las reivindicaciones 24-25 según el art. 6.1 de la ley 11/86 de patentes.