(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 805 334**

(51) Int. CI.:

***G06F 21/57*** (2013.01)
***G01R 31/3177*** (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **24.04.2012 PCT/US2012/034735**

(87) Fecha y número de publicación internacional: **15.11.2012 WO12154398**

(96) Fecha de presentación y número de la solicitud europea: **24.04.2012 E 12782825 (9)**

(97) Fecha y número de publicación de la concesión europea: **15.04.2020 EP 2707735**

(54) Título: **Sistemas y procedimientos de implementación de validación de contenido de circuitos basados en microordenadores**

(30) Prioridad:

**10.05.2011 US 201161484587 P**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
**11.02.2021**

(73) Titular/es:

**ELECTRONIC WARFARE ASSOCIATES, INC. (100.0%)**
**13873 Park Center Road, Suite 500**
**Herndon, VA 20171, US**

(72) Inventor/es:

**LA FEVER, GEORGE, BERNARD y**
**FLAUM, ISER, B.**

(74) Agente/Representante:

**GONZÁLEZ PECES, Gustavo Adolfo**

## DESCRIPCIÓN

Sistemas y procedimientos de implementación de validación de contenido de circuitos basados en microordenadores

Esta solicitud se basa en y se deriva del beneficio de la fecha de presentación de la Solicitud de Patente Provisional de los Estados Unidos No. 81/484,587, presentada el 10 de mayo de 2011.

**Campo**

La presente divulgación se dirige al campo de la verificación de la operación de circuito y, más particularmente, a sistemas y procedimientos de validación remota de circuitería que incluyen firmware de circuitos basados en microordenadores.

El documento US2010180169 describe un sistema y un procedimiento para realizar escaneos de límites de forma remota sobre una placa de circuitos, dispositivo y/o sistema a través de una red.

Lyle Pittroff describe JTAG: “Tutorial: The Role of JTAG in system debug & test throughout the embedded system development lifecycle”, 30 de abril de 2010, recuperado de Internet: http://boundaryscan.blogspot.de/2010/04/tutorial-role-of-jtag-in-system-debug.html. El artículo de Wikipedia describe “Joint Test Action Group - Wikipedia, the free encyclopedia”, 18 de diciembre de 2010 (2010-12-18), recuperado de Internet: http://en.wikipedia.org/w/index.php?title=Joint_Test_Action_Group&oldid=403093499

El documento US5535331 describe las operaciones de un dispositivo de procesamiento de datos que se rastrean al detectar una dirección de salto en la secuencia del contador del programa y empujar la dirección de salto a una pila de rastreo.

**Breve descripción de las figuras**

La invención se define por las características de las reivindicaciones independientes. Las realizaciones preferidas se proporcionan en las reivindicaciones dependientes.

Se hace referencia a los dibujos adjuntos, en los que los elementos que tienen las mismas designaciones de referencia representan elementos similares en todas partes y en los que:

La Figura 1 ilustra un sistema básico que muestra varios componentes en un sistema de conexión de instalación de validación ejemplar de acuerdo con una implementación.

Las Figuras 2A y 2B son diagramas de bloques de un sistema de escaneo de circuitos y un instrumento de escaneo, respectivamente, de acuerdo con ciertas implementaciones,

La Figura 3 es un diagrama de bloques que ilustra varios elementos ejemplares y conexiones de red, que incluyen puertos de depuración objetivo y conexiones de servidor de cliente, de acuerdo con ciertas implementaciones.

La Figura 4 es un diagrama de un sistema e instrumento de escaneo remoto diferente, que muestra una de varias permutaciones de interfaz, de acuerdo con una determinada implementación.

La Figura 5A es un diagrama de bloques de elementos informáticos distintos locales ejemplares que muestran una de varias interfaces de puerto estándar para un instrumento de escaneo, de acuerdo con ciertas implementaciones.

La Figura 5B es un diagrama de bloques de un instrumento de escaneo independiente local ejemplar, que muestra elementos funcionales dentro para realizar el escaneo, de acuerdo con ciertas implementaciones.

La Figura 5C es un diagrama de bloques de un instrumento de escaneo de factor de forma de unidad flash local, de acuerdo con una implementación.

Las Figuras 6A y 6B son diagramas que ilustran un procesamiento de escaneo ejemplar al nivel del servidor, nivel del instrumento y nivel objetivo junto con interacciones de nivel, de acuerdo con ciertas implementaciones.

La Figura 7 es una ilustración esquemática que muestra características de extracción y temporización de firmware ejemplares, de acuerdo con ciertas implementaciones.

**Descripción de realizaciones de la invención**

En lo siguiente, el término “microordenador”, “microcontrolador”, “procesador” y/o “CPU” se pueden emplear indistintamente. Adicionalmente, la frase “puerto de diagnóstico” o “puerto de depuración” se puede utilizar indistintamente para referirse a interfaces de microordenador que están dedicadas a exponer el acceso de control de

comprobabilidad especial a operaciones internas, distintas de las interfaces y puertos de aplicación. Dichos puertos normalmente emplearán protocolos de señalización JTAG y señales para interacción externa. La frase “flash” o “memoria flash” es intercambiable con cualquier dispositivo electrónico cuyos patrones de contenido binario ordenados se pueden cargar eléctricamente para su almacenamiento, y luego se pueden recuperar incluso después de haber sido desconectados.

Esta divulgación se refiere a la validación de código almacenado en circuitos y/o los propios circuitos. Los circuitos que se van a validar pueden comprender placas y/o un conjunto de una o más placas, dispositivos y/o sistemas. Los sistemas y procedimientos en la presente memoria pueden facilitar la confirmación segura de que los patrones binarios contenidos en dichos circuitos no se modifican de sus valores previstos, por ejemplo, como se instala en el momento de producción o actualización. Las innovaciones actuales también pueden incluir características de validación de valores de firma reducidos derivados de los patrones binarios, en lugar de o además de patrones binarios completos (originales). Como tal, los sistemas y procedimientos en el presente documento pueden validar el contenido de la memoria sobre un objetivo en ejecución mediante diversas técnicas, que incluyen la comparación de datos de firmware extraídos del objetivo como un patrón completo de valores contra una referencia conocida y/o comparar valores de firma reducidos derivados de los patrones binarios contra referencia(s) de firma conocidas. Adicionalmente, los sistemas y procedimientos en la presente memoria pueden comparar uno o ambos tipos de datos en el servidor remoto (después de cargar datos o firma) y/o pueden realizar comparaciones en el controlador de escaneo local, después de descargar la referencia desde una ubicación remota tal como un servidor. Adicionalmente, de acuerdo con ciertas implementaciones, no se necesitan adaptaciones especiales en los circuitos probados o en cualquier contenido del programa para admitir algunas realizaciones de las características de validación en la presente memoria, más allá de la provisión de una conexión de puerto de depuración.

Por medio de algunas implementaciones, por ejemplo, un servidor web remoto puede asegurar automáticamente que no se hayan introducido corrupciones o desviaciones en el código, ya sea con culpa o sin culpa. De acuerdo con varios aspectos en la presente memoria, se pueden detectar variaciones de dicho código, que podrían dar lugar a un comportamiento incorrecto, desviado o no autorizado de los circuitos que lo contienen. Más aún, las innovaciones en la presente memoria incluyen procedimientos para realizar de forma remota la validación periódica de dicho contenido programado como una tarea de fondo, lo que permite la detección e indicación de cualquier alteración inesperada del código mientras los circuitos continúan realizando sus funciones principales.

Muchas de las realizaciones utilizadas para la siguiente ilustración se refieren a la validación remota del código de firmware almacenado en un almacenamiento no volátil (por ejemplo, memoria Flash, etc.) de los circuitos que se están probando. Se presume que dichos elementos de almacenamiento no han cambiado desde que se cargaron o actualizaron correctamente, y se compararon fácilmente con su contenido de diseño previsto. Sin embargo, también se proporcionan realizaciones relacionadas con la validación del almacenamiento volátil (por ejemplo, RAM, etc.), de acuerdo con aspectos de la presente invención. Con respecto al almacenamiento volátil, por ejemplo, la validación se puede lograr dadas las condiciones estables, tales como el contenido conocido en ciertos intervalos de tiempo y/o direcciones. Dichas innovaciones se pueden utilizar, entre otras cosas, cuando la CPU objetivo ha transferido el ejecutable desde FLASH hasta RAM y se pueden establecer dichos sectores como bien definidos e inmutables por un tiempo durante el ciclo de validación.

En algunas realizaciones, por ejemplo, el componente de microordenador local transfiere sectores ejecutables desde la memoria flash u otras fuentes hasta la RAM, que luego se puede convertir en el depósito de búsqueda para la ejecución. En la presente memoria, por ejemplo, la información que incluye instrucciones ejecutables almacenadas en almacenamiento volátil o RAM también se puede validar cuando los sectores se caracterizan por tener estados establecidos y permanecer completamente en los estados establecidos por la duración de la secuencia de extracción completa (cubriendo una o más etapas incrementales) Dicho contenido general de memoria sería reconocible en comparación con los patrones esperados.

Adicionalmente, en los sistemas y procedimientos que permiten la validación de códigos colocados en almacenamiento volátil, dichos códigos se pueden haber originado a partir de almacenamiento no volátil, o se pueden haber construido con el tiempo a partir de varias otras fuentes, tales como elementos descargados externamente, parámetros generados en tiempo de ejecución, etc. Con la condición de que dichos patrones y ubicaciones de almacenamiento se conozcan de forma remota y sean lo suficientemente estables por la duración de un ciclo de validación completo, se pueden procesar de manera equivalente y autenticar de acuerdo con las innovaciones en la presente memoria.

Las realizaciones en la presente memoria se aplican a circuitos que incluyen uno o más microordenadores operados mediante la ejecución de código de firmware almacenado. Ciertas realizaciones pueden implicar además microordenadores que tienen un puerto de prueba de depuración, por ejemplo, un EJTAG o COP (puerto de operación de ordenador) expuesto con una conexión eléctricamente accesible. En la presente memoria, estos puertos pueden admitir la interacción con equipos externos, mientras el programa de aplicación del circuito continúa ejecutándose. En algunas implementaciones, por ejemplo, dichos puertos de depuración proporcionan visibilidad al instrumento conectado externamente del código de firmware almacenado de una manera benigna. Mediante el uso de dichos puertos, las realizaciones en la presente memoria pueden permitir la visibilidad y/o realizar pruebas sin interrumpir sustancialmente las operaciones de tiempo de ejecución en curso, excepto por la detención momentánea

de la ejecución del programa de aplicación. Los puertos de depuración de este tipo están frecuentemente disponibles en muchos dispositivos de microordenadores comerciales. Algunas realizaciones también pueden utilizar el acceso adicional del instrumento a una conexión de red, tales como Internet, para permitir el intercambio de información con un sitio o servidor remoto. De esta manera, por ejemplo, los instrumentos de escaneo en la presente memoria se pueden configurar para validar, a través de la comunicación con el sitio o servidor remoto, la información de prueba, es decir, para confirmar la configuración u operación del sistema deseada del objetivo.

En el contexto de las realizaciones en la presente memoria, el firmware se puede definir como un tipo de software, que incluye parámetros de datos relacionados, que es legible y ejecutable como instrucciones de microordenador almacenadas en uno o más de los dispositivos de almacenamiento no volátiles de un circuito. Dichos dispositivos pueden ser programables, cargados con este firmware y capaces de retenerlo incluso si el circuito queda desconectado (es decir, almacenamiento no volátil). Estos dispositivos generalmente se cargan con la información de firmware en cuestión durante la fabricación o en un momento de actualización posterior por medios eléctricos. Son susceptibles de ser reemplazados físicamente por otros dispositivos de construcción similar que contienen contenido alterado como un medio para cambiar la programación del circuito. También son susceptibles de tener un código alternativo no autorizado cargado eléctricamente en lugar de los patrones previstos. Normalmente, el contenido de este firmware determina fundamentalmente las características de comportamiento del circuito general cuando se manipula mediante su microordenador.

En otras realizaciones, el firmware tal como se utiliza en la presente memoria también se puede aplicar al código que se mantiene durante un tiempo en almacenamiento volátil, tal como RAM, que se origina en almacenamiento no volátil u otras fuentes, cuyo patrón de contenido permanece suficientemente estable para ser conocido, por ejemplo, a un sitio remoto seguro.

La Figura 1 ilustra un sistema básico que muestra los componentes de un sistema de conexión de instalación de validación ejemplar, de acuerdo con ciertas implementaciones. El sistema ilustrado en la Figura 1 comprende objetivos 110 tales como placas de circuito dentro de los componentes de una pila, un instrumento 120 de escaneo y un enlace o conexión 122 a una red. Todos estos elementos del sistema pueden estar acoplados a una red a través de una plataforma de validación que contiene el instrumento 120 de escaneo, que a su vez se conecta a los circuitos objetivo. El sistema establecido en la Figura 1 es ejemplar, que muestra solo un sistema ilustrativo con el que se pueden asociar aspectos de las presentes invenciones. Se incluyen otras variaciones dentro del alcance de las innovaciones en la presente memoria. Con referencia a la Figura 1, los objetivos 110 que se van a validar pueden comprender una o más placas de circuito conectadas a una carcasa 115 de bastidor de chasis. En la presente memoria, por ejemplo, los circuitos clave contenidos dentro de las placas de circuito, cuyo firmware debe ser validado, normalmente incluyen y son controlados por un microordenador 130 que tiene un puerto de depuración JTAG y ejecuta programas de firmware almacenados leídos desde dispositivos de memoria Flash.

De acuerdo con el sistema de la Figura 1, un instrumento de escaneo se puede acoplar a una serie de circuitos objetivo de depuración JTAG encadenados. Por ejemplo, dichos sistemas pueden incluir cableado para encadenar los puertos JTAG de múltiples circuitos juntos, lo que permite la conexión a un puerto de acceso de prueba (TAP) de controlador de escaneo JTAG estándar de la industria. El instrumento 120 de escaneo mostrado en la presente memoria se ilustra en el contexto de un componente de la plataforma, aunque puede estar dentro o asociado con otros componentes, tener un factor de forma diferente, etc., y en esta divulgación se denomina “controlador de escaneo”. “instrumento de escaneo” o simplemente “instrumento” en todo momento. Dicho instrumento puede gestionar las secuencias de señal dirigidas hacia y recibidas desde los circuitos objetivo a través de los puertos de depuración encadenados. Las señales, en la presente memoria, en este ejemplo, pueden ser señales que cumplen con el protocolo JTAG y transmiten comandos y respuestas de depuración del microordenador para operar esta característica simultáneamente con la ejecución de la aplicación en tiempo de ejecución.

En la realización ilustrada, el instrumento 120 de escaneo también se conecta a través de un Ethernet 140 (o un accesorio similar) a Internet, ya sea directamente, y/o se puede conectar a través de un ordenador host de cliente local intermedio. Como tal, el instrumento se puede configurar como visible para un servidor 150 web remoto, que puede dirigir el instrumento 120 de escaneo para utilizar las funciones de comando de depuración del microordenador y recibir información recopilada de los circuitos objetivo en cuestión.

Como se explica con más detalle a continuación, el instrumento de validación o escaneo 120 puede detener periódicamente cada microordenador para extraer incrementalmente fragmentos de memoria Flash (y/o RAM estable) en direcciones de interés. Luego, el instrumento de escaneo puede permitir que los microordenadores continúen haciendo funcionar la ejecución de la aplicación durante tiempos globales o intercalados, a través de características e innovaciones que tienen un impacto mínimo en la operación. Estos constituirán ciclos mínimos de extracción de memoria de detención/ejecución aplicados al objetivo. Luego, se procesa algorítmicamente un total o un subtotal del contenido del firmware recopilado para calcular una representación de firma única (como un código de resumen hash), también denominado vector del código muestreado. La firma resultante se puede, por ejemplo, transmitir a través de Internet al servidor 150 remoto, que luego recibe la firma calculada para compararla con el valor de firma de referencia conocido/deseado para el circuito o sistema dado.

El servidor 150 remoto también puede supervisar el instrumento de escaneo local para programar y lanzar cada actividad de extracción de firmware. Adicionalmente, en algunas implementaciones, solo el código de firma del contenido del firmware, en lugar del contenido del firmware en sí, se transmite a través de Internet, maximizando la seguridad y minimizando el tiempo de comunicación. Sin embargo, la transmisión de todo el conjunto de valores de firmware extraído también sigue siendo una alternativa viable. Como defensa de seguridad adicional, el algoritmo de generación de firmas puede incluir características de codificación, tales como números aleatorios, marcas de tiempo, etc., que se proporcionan desde el servidor remoto y se agregan a los valores Flash extraídos durante la generación de resumen hash. El uso de dichas funciones de codificación, como números aleatorios o marcas de tiempo, también se puede utilizar para alterar los resultados en cada tiempo de ciclo, determinable por el servidor remoto, y prevenir o confundir los intentos de analizar patrones Flash objetivo por parte de oyentes no autorizados. Dichas características también evitarían la sustitución de la firma inmutable como un engaño (por ejemplo, por fuentes de redes de intermediarios clandestinas), cuando las firmas del circuito real pueden haber cambiado debido a corrupción.

Ciertas realizaciones pueden emplear características de puerto de depuración del microcontrolador, por ejemplo, EJTAG o COP (puerto de operación de ordenador) que evitan requisitos para adaptaciones especiales en los circuitos o firmware de los productos que se van a validar. Dicho puerto provisto comúnmente es un punto de acceso de prueba incorporado distinto de todas las demás interfaces de aplicación del microcontrolador utilizadas para e/s convencionales con dispositivos periféricos (por ejemplo, memoria). Este puerto de depuración dedicado (normalmente utilizando el protocolo JTAG) permite la intervención externa del procesamiento normal para admitir las operaciones de prueba. Incluye suspender brevemente la búsqueda y ejecución del código de instrucción normal. Adicionalmente, permite que el microcontrolador suspendido dominante transmita el contenido de memoria desde el almacenamiento del circuito hasta el instrumento externo. Adicionalmente, se puede ordenar al microordenador que reanude la ejecución del código después del punto de suspensión.

Las características de depuración adicionales también se pueden proporcionar a través de los sistemas y procedimientos incluidos en la presente memoria, que incluyen comandos para escribir el almacenamiento de memoria desde el instrumento externo, pasar individualmente la búsqueda/ejecución de una instrucción, saltar a una ramificación de código fuera de secuencia, y realizar otras acciones de la CPU. Estas características pueden, por ejemplo, evitar la necesidad de construir dichos productos con elementos especiales de hardware o software de validación de seguridad ya que el instrumento externo puede acceder a fragmentos de contenido de memoria entre ejecuciones de instrucciones normales por parte del microcontrolador a través de su puerto de depuración. Este procedimiento esencialmente roba ciclos de breves intervalos de tiempo del procesamiento normal de la CPU para obtener visibilidad del contenido de la memoria. De acuerdo con lo anterior, los circuitos objetivo disponibles se pueden acomodar y/o asegurar de acuerdo con estas realizaciones. Adicionalmente, ciertas implementaciones se pueden configurar para evitar la inclusión de cualquier procesamiento incorporado en el firmware del circuito en el resultado de la firma, para que no se instalen corrupciones subrepticias para imitar el resultado adecuado y/u ocultar las desviaciones que están presentes. De hecho, en algunas implementaciones, la firma de validación se puede crear como una función exclusiva solo de los patrones de firmware almacenados, por ejemplo, como se observa externamente sin ayuda del código validado en sí mismo. Sin embargo, incluso estas características no necesariamente impiden que el instrumento de escaneo descargue el código de asistencia temporal en la RAM y le ordene al microordenador objetivo que ejecute dichas rutinas auxiliares para acelerar las operaciones de recopilación y firma. Sin embargo, algunas de estas innovaciones pueden requerir el conocimiento de dicho espacio RAM disponible para no perturbar el procesamiento de la aplicación en curso.

Como se mencionó anteriormente, un instrumento que opera uno o más de los puertos de depuración del circuito se puede conectar operativamente a la red, ya sea directamente, tal como a través de un puerto Ethernet, o indirectamente, tal como a través de la asistencia de un componente de procesamiento local interviniente u ordenador del cliente. En el último caso, el ordenador local se puede conectar operativamente a la red, bajo el control del servidor remoto, y a su vez expondría un puerto capaz de comunicarse con el instrumento. En la presente memoria, el instrumento de escaneo y/o el componente informático pueden incluir software controlador para permitir el acceso entre la red y el instrumento conectado, y admitir interacciones de depuración con los circuitos que se validan. Como tal, un instrumento de escaneo conectado directamente a la red o asistido por un ordenador servidor de comunicaciones separado se puede implementar de acuerdo con diversas realizaciones en la presente memoria.

Las realizaciones divulgadas en la presente memoria emplean la conexión de equipos de prueba que se conectan y operan las características de depuración de los microordenadores de los circuitos, permitiendo la extracción incremental de datos de firmware almacenados como una operación de fondo. El instrumento de escaneo externo puede operar el microcontrolador objetivo a través de su puerto de depuración mediante comandos secuenciales para detener la ejecución normal después de la instrucción actual, transmitir al instrumento ciertos registros de contexto para su preservación, transmitirle una o más palabras de la memoria (no volátil, por ejemplo, flash o volátil, por ejemplo, RAM) para la extracción, devolver los valores guardados del instrumento a ciertos registros que restauran el contexto y continuar la ejecución normal en la instrucción que sigue a la detención. Mediante repeticiones del ciclo anterior, que permiten intervalos de tiempo de ejecución intervinientes significativos de procesamiento normal, el instrumento de escaneo puede extraer secciones de contenido de memoria para su posterior compactación/cifrado de firma y validación de patrones con una interrupción mínima. Como se describe, no se requiere hardware especial (además del puerto de depuración) ni rutinas de código incorporadas en esta

operación. El impacto principal en el sistema objetivo, que se tendrá en cuenta en el diseño de temporización juicioso de este procedimiento, consiste en las múltiples suspensiones breves de la ejecución del código de la aplicación. Estos pueden disminuir algo la capacidad de rendimiento objetivo (debido al ciclo de robo del procesador) y posponer la respuesta de interrupción a eventos aleatorios (mientras el microordenador está suspendido). El primer impacto se relaciona con la cantidad de margen de procesamiento disponible en un sistema objetivo dado, que puede experimentar cierta degradación. El segundo se refiere a la capacidad de respuesta objetivo a sucesos externos indeterminados, que señalan la redirección del procesamiento de interrupción a las rutinas de manejador a medida que se demandan servicios especiales.

Mantener los ciclos de extracción de memoria mínimamente breves y máximamente escasos son objetivos de diseño logrados en la incorporación de los sistemas y procedimientos de la presente memoria. Estos ciclos de extracción se pueden realizar mientras el microordenador está ejecutando el código de la aplicación, aunque el microordenador puede detenerse momentáneamente y luego reiniciarse entre las instrucciones. La información extraída luego se puede transportar a una ubicación remota o servidor para compararla con una firma de referencia correcta conocida, para validar su integridad inalterada. La información extraída, o una representación de firma única computada de la misma, se pueden comunicar a un sitio seguro para su comparación. En la presente memoria, por ejemplo, los patrones de código se pueden procesar en una firma de resumen hash compacta SHA-2 (o similar) para el enlace ascendente a un servidor seguro fuera del sitio a través de una red como Internet. Los aspectos de las innovaciones en la presente memoria también pueden incluir una funcionalidad que proporciona una variación de firma para cada instancia de código transmitido. Adicionalmente, los sistemas pueden incluir otras características que permiten validar el cumplimiento de los patrones de código de firmware según se autorizan en la entrega o actualización, tal como estar configurados para realizar pruebas y/o comparaciones de forma automática y/o periódica.

Como se estableció anteriormente, un componente o servidor informático remoto puede proporcionar una comprobación segura de firmas computadas. De acuerdo con diversas realizaciones en la presente memoria, dicho componente o servidor puede computar una firma de referencia original correcta del código fuente de fábrica. En la presente memoria, por ejemplo, un fabricante puede proporcionar características detalladas de la CPU, mapas de memoria y comportamiento de temporización del objetivo. En algunas implementaciones, esta información permite que el instrumento se configure para extraer solo sectores críticos de firmware que no alteran y, posiblemente, RAM estable. Dicha información también puede informar a los desarrolladores cómo configurar instrumentos de escaneo para un ajuste óptimo de las interrupciones del ciclo de detención/ejecución de la CPU objetivo. De hecho, las divulgaciones cooperativas de los desarrolladores de circuitos pueden ser necesarias para ciertos circuitos y sistemas.

Como se indicó anteriormente, varias realizaciones de las presentes innovaciones no requieren adaptaciones de objetivos especiales, aparte de un puerto de depuración de microordenador. Sin embargo, la conexión de la instrumentación del equipo de prueba, en la presente memoria, no pretende excluir siempre los casos en los que el montaje de componentes o funcionalidad de dicha instrumentación se incorpora al circuito que se está probando. De manera similar, dicha instrumentación se podría incorporar en un ordenador distinto como un aumento conectado o alojado dentro de su recinto, y todas esas permutaciones se incluyen dentro del contexto de las innovaciones en la presente memoria.

Como un ejemplo, varios microordenadores comunes admiten un puerto de depuración de protocolo JTAG, a través del cual se pueden facilitar aspectos de las innovaciones. Dichas capacidades de depuración se deben distinguir de las características de escaneo de límites heredadas de JTAG, en general, que se centran en la prueba de conexión de pines del dispositivo IC y, por lo tanto, permiten una serie de procedimientos de prueba de circuito estándar.

La Figura 2A es un diagrama de bloques de un sistema 100 ilustrativo que involucra un instrumento 120 de escaneo de acuerdo con ciertas implementaciones. En la presente memoria, un objetivo 110 se conecta a un instrumento 120 de escaneo a través de una interfaz o conector 115 para realizar pruebas de la circuitería relevante y/o extraer contenido de memoria. Si bien la siguiente descripción generalmente se refiere a un objetivo 110, las innovaciones en la presente memoria también se aplican a circuitos, placas, dispositivos, sistemas, otros componentes, etc. Junto con un componente 130 informático, que puede ser un procesador, bandeja para servidor, ordenador personal, etc., el instrumento 120 de escaneo ejemplar, en la presente memoria, se ilustra con o dentro de los componentes 140 locales colocados en la proximidad inmediata de los circuitos de prueba. Estos componentes 140 locales pueden ser parte de un sistema de procesamiento local unitario, que puede albergar ambos elementos, o su funcionalidad se puede distribuir en dos o más subcomponentes, como se describe en otra parte. Los componentes 140 locales se pueden comunicar con un servidor 150 remoto a través de una red, y pueden incluir opcionalmente un subcomponente 132 de pantalla/GUI/navegador, tal como en el componente 130 informático. El servidor 150 remoto puede contener o acceder a una biblioteca de información específica de placa y escaneos 152 de prueba, y también puede incluir los componentes/funcionalidad de la interfaz 156 de usuario, además de otro software, tal como el software 154 que alberga instrumentos de escaneo. Adicionalmente, el componente 130 informático también puede incluir uno o más componentes 134 de prueba, componentes 136 relacionados con controladores y puertos/conectividad, así como otros componentes 139 de escaneo de prueba. Adicionalmente, el hardware 138 del instrumento de escaneo central puede residir en el componente 130 informático, o puede ser parte del firmware 122 instalado en el instrumento 120 de escaneo.

Adicional o alternativamente a las interconexiones en banda divulgadas en otra parte en la presente memoria, la interfaz o conector 115 puede incluir un conector JTAG asociado con una cantidad creciente de placas de circuito y/o dispositivos y/o sistemas. Por ejemplo, dicho conector JTAG puede admitir una interfaz de prueba al mundo exterior para la placa, pasar señales que conducen hacia circuitos integrados (IC) accesibles a través del puerto JTAG para proporcionar un desplazamiento secuencial en bits, controlar la alimentación y señales de reloj a todos los circuitos participantes/IC, y extraer bits secuenciales que se recogen a lo largo de la cadena. Adicionalmente, el conector JTAG puede, en algunas implementaciones, ser operado indirectamente por un componente informático host. En la presente memoria, el software host puede determinar o ayudar a determinar el código particular para acceder y verificar y/o probar el patrón que se va a utilizar, y también puede identificar un significado particular para estos patrones (tipo de escaneo), según se modifica por las señales de control, aplicadas a los patrones en cuestión.

El software también puede recoger el flujo de vuelta al host para su análisis, en el que puede interpretar los datos de acuerdo con el tipo de código y/o ciclo de escaneo obtenido. Sin embargo, en algunas implementaciones, el software puede no ser consciente de la temporización exacta y la secuencia de protocolo de las señales, ya que se escanean dentro/fuera de la cadena a nivel eléctrico. En estas implementaciones, los protocolos de escaneo específicos de JTAG pueden ser manejados por el hardware de instrumento de escaneo intermedio, bajo la dirección del host. Estas implementaciones también pueden incluir hardware o firmware dedicado que permite un procesamiento de desplazamiento de escaneo más rápido (agilizando el número potencialmente grande de permutaciones para ejecutar y una larga longitud de cadena) de lo que está disponible a través del procesamiento existente a través de puertos estándar, que no pueden permitir dichas tasas tan altas. Dichos instrumentos de escaneo también pueden facilitar otros requisitos eléctricos peculiares de la placa bajo prueba, tales como niveles de voltaje especiales o tasas de transición de borde de señal, que el ordenador no puede acomodar fácilmente.

De acuerdo con estas características, los instrumentos 120 de escaneo pueden incluir o cargarse con subcomponentes de hardware, firmware y/o software configurados para hacer uno o más de los siguientes: (1) interactuar con el host (u otro componente de comparación) a través de uno de varios protocolos de enlace estándar, dependiendo de la plataforma y/o modelo del instrumento; (2) interactuar con el objetivo de acuerdo con el protocolo JTAG a altas velocidades; (3) aceptar comandos desde el host que inician escaneos específicos, tipos de secuencias de escaneo y/o extracciones de memoria a través de la cadena JTAG objetivo; (4) aceptar patrones de bits desde el host para desplazarse hacia la cadena de escaneo objetivo; (5) recopilar patrones de bits de retorno del objetivo y transmitirlos de regreso al host; (6) crear el patrón de señal de flujo de bits en serie adecuado hacia el objetivo; (7) crear las secuencias de señal de control y temporización adecuadas para operar el desplazamiento de JTAG de acuerdo con esa memoria descriptiva; (8) proporcionar un acondicionamiento de señal eléctrica peculiar según lo requiera la placa conectada (tal como el nivel de voltaje o las tasas de aumento/disminución de la señal); y/o (9) proporcionar cualquier configuración adicional y funciones de estado para facilitar la administración del instrumento 129. Los instrumentos de escaneo y los sistemas y procedimientos relacionados en la presente memoria también pueden incluir características y funcionalidades establecidas en la solicitud de los Estados Unidos No. 12/641,627, presentada el 18 de diciembre, 2009, publicada como US2010/0180169A1.

La Figura 2B ilustra una estructura generalizada de un instrumento 120 de escaneo ejemplar que se puede configurar con dichas características y en el que se puede realizar un procesamiento de escaneo apropiado. Si bien los bloques básicos dentro de dichos instrumentos de escaneo son, en general, conocidos, la instalación y operación de ciertas características en la presente memoria permiten un control innovador de los instrumentos de escaneo y/o manejo de la funcionalidad del instrumento de escaneo de forma remota. Por ejemplo, los instrumentos 120 de escaneo de acuerdo con los aspectos de las innovaciones en la presente memoria se pueden configurar para que no tengan implicación directa con el firmware del objetivo. Adicionalmente, los instrumentos 120 de escaneo en la presente memoria se pueden configurar para ejecutarse continuamente mientras el objetivo se está ejecutando, para obtener el control de sus operaciones, que incluyen la detención breve de la CPU para ejecutar secuencias de escaneo con el fin de obtener segmentos de código incorporado, seguido en ejecución de la CPU que permiten la continuación de su ejecución de aplicación. Como tal, los instrumentos 120 de escaneo generalmente están configurados de manera diferente a los componentes de escaneo de límites heredados, tales como no estar programados para deshabilitar funcionalmente un objetivo para que sus pines de dispositivo puedan ser accionados y detectados de manera arbitraria (prueba de interconexión). Dichas configuraciones enfocan el escaneo solo sobre las características de diagnóstico conectadas al procesador (por ejemplo, en relación con los intercambios de control de comando de depuración de entrada/salida del microcontrolador escaneado), contra las configuraciones dirigidas al escaneo de control de entrada/salida de pin típico (por ejemplo, en relación con sí o no los pines del dispositivo no encadenados se interconectan o no en la placa mediante trazas de circuito).

En algunas implementaciones en la presente memoria, el instrumento 120 puede estar completamente incorporado en el entorno de la placa, con el sistema operativo relevante y las aplicaciones integradas completamente en el mismo. Más aún, dichos instrumentos de escaneo se pueden configurar como completamente independientes, de modo que no requieren recursos incorporados en otro lugar para ayudar en el procesamiento de validación. Adicionalmente, la lógica 240 de control se puede configurar para generar códigos hash dentro del instrumento 120 de escaneo externo después de extraer el código de firmware objetivo, en el que solo se transmite un patrón de firma a un componente de validación de seguridad remota para realizar la comparación de verificación. En cuanto a la operación, la lógica 240 de control recibe información y transmite comunicaciones a un servidor o componente

informático a través de la interfaz 230, que puede ser una interfaz estándar en ciertas implementaciones, y procesa las comunicaciones con respecto al objetivo a través de una interfaz JTAG.

La Figura 3 es un diagrama de bloques que representa elementos y conexiones de red ejemplares, de acuerdo con ciertas implementaciones. Con referencia a la Figura 3, se ilustran varias permutaciones de elementos, instrumentos de escaneo, tipos de puertos de depuración objetivo y conexiones de red diferentes (directas frente a locales). La Figura 3 ilustra un servidor 150 que puede contener o acceder a una biblioteca 340 de valores de referencia de firma de validación, una red 320 por la cual se transmiten 315 las diversas firmas, y una o más instalaciones 310A, 310B, 310C de prueba. Dentro de cada instalación de prueba, los instrumentos 120 de escaneo se conectan a placas objetivo para la visibilidad y/o control de depuración de microordenadores. Por ejemplo, los instrumentos 120 se pueden conectar directamente a los puertos de depuración JTAG sobre las placas, ya que muchos circuitos contienen dichos puertos de depuración JTAG con acceso de zócalo empleado para emulación de desarrollo, inicialización o prueba. Adicionalmente, los objetivos en la cadena con placas JTAG de componentes sin procesador o componentes 350C también se pueden acomodar, a través de instrumentos 120 configurados con protocolos que se centran en serie en una placa/dispositivo a la vez. Una instalación 310A de prueba ejemplar mostrada en la Figura 3 comprende un cliente 370 local, así como uno o más tipos de instrumentos 120A, 120B, 120C, 120D de escaneo, que están acoplados a uno o más circuitos para ser validados 330A, 330B. La realización ilustrada muestra múltiples configuraciones, que incluyen los instrumentos 120A, 120B acoplados directamente a la red 320, así como los instrumentos 120C, 120D acoplados a la red a través de un ordenador 370 de cliente local.

La realización de la Figura 3 muestra un instrumento 120A conectado a una cadena de circuitos 350A, 350B, 350C objetivo (por ejemplo, conectado bajo el protocolo JTAG, etc.), así como un instrumento 120B conectado a un circuito 350D objetivo no encadenado (por ejemplo, utilizando un protocolo patentado, tal como BDM, SPI, MDIO o RS-232, etc.). La Figura 3 también muestra los instrumentos 120C, 120D de escaneo configurados para una conexión o comunicación que no sea de Ethernet a una red o componente remoto, indirectamente. En algunas implementaciones, estos instrumentos se pueden configurar como instrumentos 120C JTAG para acoplarse a objetivos a través de puertos JTAG en las placas, o como instrumentos 120D no JTAG para acoplarse a objetivos a través de puertos o medios de diagnóstico no JTAG. Volviendo a los instrumentos de escaneo Ethernet 120A, 120B, la Figura 3 ilustra dos tipos de instrumentos, así como varias conexiones y puertos 360A, 360B, 360C ejemplares. Un primer tipo de instrumento Ethernet en la Figura 3 es un instrumento 120A de escaneo JTAG acoplado a una cadena 350A, 350B, 350C de circuitos JTAG. Como se muestra a modo de ejemplo y no como limitación en la Figura 3, el primer circuito 350A y el segundo circuito 350B de prueba pueden estar acoplados al instrumento 120A de escaneo a través de los puertos 360A, 360B de diagnóstico JTAG de tipo procesador. Adicionalmente, se pueden incluir circuitos adicionales, como el tercer circuito 350C en la cadena JTAG que contiene solo componentes que no son del procesador y que no participan en este procedimiento de validación de firmware. La Figura 3 también ilustra, como un segundo tipo de instrumento Ethernet, un instrumento 120B de escaneo no JTAG, que se muestra acoplado a un cuarto circuito 350D a través de un puerto 360D de diagnóstico no JTAG.

En resumen, la Figura 3 muestra la variedad de circuitos incluidos en las instalaciones de circuito de la realización ejemplar del sistema de conexión de validación (variaciones de circuito objetivo-JTAG/no JTAG, encadenado/desencadenado, depuración de CPU/dispositivos sin CPU: variaciones de instrumentos de controlador de escaneo: JTAG/no JTAG, accesorio encadenado/no encadenado, Ethernet-directo-conectado-en red/local-cliente-en red). También muestra la topología general del sistema de un servidor remoto vinculado a la red central con bibliotecas de validación adjuntas, capaz de interactuar con una distribución de instalaciones dispersas que necesitan validación de firmware.

Se pueden realizar varios otros sistemas e instrumentos de escaneo remoto, más allá de los establecidos anteriormente, de acuerdo con las innovaciones en la presente memoria. Por ejemplo, la Figura 4 es un diagrama de un sistema e instrumento de escaneo remoto diferente, de acuerdo con una determinada implementación. Este muestra una configuración ilustrativa de una instalación de escaneo ejemplar ubicada en la ubicación de prueba objetivo local. La Figura 4 muestra un instrumento 120 de escaneo acoplado con un dispositivo 480 móvil para comprometer su capacidad de transportar datos en cualquier dirección con un elemento remoto tal como un servidor. La implementación de la Figura 4 mejora en gran medida la flexibilidad de la disposición de escaneo al permitir una conectividad alternativa con un servidor central en el que no está disponible el acceso a la red por cable. En tal caso, se puede cargar una aplicación en dicho dispositivo móvil para operar el controlador de escaneo adjunto, a través de conexiones comúnmente disponibles tales como USB, como su cliente local. Adicionalmente, en las diversas implementaciones divulgadas en la presente memoria, un instrumento de escaneo independiente también puede tener un factor de forma para permitir el montaje directamente en la placa objetivo, lo que ahorra espacio y proporciona comodidad y funcionalidad adicional.

La Figura 5A es un diagrama de bloques de una configuración de cliente informático local ejemplar que muestra una de varias interfaces de puerto estándar para un instrumento de escaneo, que incluye algunos elementos funcionales para realizar pruebas/escaneo/extracción, de acuerdo con ciertas implementaciones. Con referencia a la Figura 5A, se muestra un diagrama de bloques generalizado de los registros 710 vistos por el software del componente informático local, que ilustra el control/visibilidad del instrumento de escaneo y para pasar vectores hacia y desde el mismo. Dichos vectores pueden incluir direcciones de memoria de las que se extraerán datos y el contenido de la

memoria en sí mismo. Nuevamente, este es simplemente un diagrama de bloques ilustrativo, ya que la configuración de varios componentes locales puede variar, como se describe en otra parte en la presente memoria.

Adicionalmente, cuando se transportan los comandos/estado/vectores hacia o desde el instrumento 120 de escaneo local, el componente 370 informático del cliente local también puede codificar/decodificar estos elementos de datos en los formatos de transferencia apropiados para el protocolo del puerto. Por ejemplo, la información de registro de comando destinada para el instrumento 120 de escaneo se puede encapsular en paquetes TCP/IP para versiones vinculadas a Ethernet. Los elementos locales también deben proporcionar y/o procesar información adicional para dirigir el mensaje hacia/desde el destino/fuente dentro del instrumento de escaneo. De manera similar, los paquetes USB preparados para la transmisión contienen información y formato similares para ese puerto de acuerdo con sus estructuras. Los componentes del cliente local también se pueden descargar con software para manejar todas las interacciones de toma de contacto sobre el bus de acuerdo con el protocolo en curso. Como tal, además de crear el contenido del registro del instrumento de escaneo, el software descargado en el componente 370 informático del cliente local puede crear o descifrar construcciones relacionadas con enlaces e interactuar mediante las reglas de protocolo apropiadas a través del puerto. En algunas implementaciones, las rutinas de controladores autocontenidas realizan este trabajo; que se dedican a realizar dichas comunicaciones, diseñadas para operar con adaptadores de puerto instalados y disponibles para descargar desde el servidor 150 remoto en el caso de que la interfaz del instrumento del controlador de escaneo, 230, se ajuste con protocolos de red comunes, tales como Ethernet, el componente del cliente local se puede omitir con el instrumento conectado directamente a dicha red que controla el servidor remoto.

Cuando se procesan vectores en cualquier dirección, los registros 710 de instrumento de escaneo que almacenan vectores pueden ser muy largos o numerosos, es decir, para admitir cadenas objetivo arbitrariamente largas. En estas situaciones, el instrumento 120 de escaneo puede implementarlos como una serie completa de registros en forma de una memoria FIFO (primero en entrar, primero en salir) u otros procedimientos de almacenamiento de memoria masiva capaces de almacenar grandes cantidades de datos ordenados. Dicha FIFO o búfer generalmente se requiere para los vectores de escaneo devueltos objetivo, también, reteniéndolos antes del envío al componente 370 informático del cliente local y/o al servidor 150 remoto. Para acomodar casos en los que la longitud total del vector excede incluso dichos recursos, se pueden proporcionar mecanismos para que el instrumento de escaneo se acelere debido a las limitaciones de almacenamiento. En una implementación, por ejemplo, el instrumento de escaneo para momentáneamente el desplazamiento del objetivo si necesita nuevos bits de vector TDI y la FIFO está vacía, o la FIFO de almacenamiento TDO está llena esperando el enlace ascendente a la PC, sin dejar espacio para recolectar más del objetivo.

La Figura 5B es un diagrama de bloques de otro instrumento de escaneo local ejemplar del tipo independiente, que incluye recursos adicionales que permiten características tales como la operación sin la participación de otros elementos informáticos del cliente local y/o guardar los resultados de la prueba para su posterior carga. Con referencia a la Figura 5B, se muestra un instrumento de escaneo independiente, que puede incluir planes de prueba precargados y/o recursos adicionales para permitir el escaneo sin estar conectado a un componente informático o elemento de plataforma local o remota. Por ejemplo, puede contener un procesador 570 integrado, así como una memoria 580 (incorporada o extraíble), configurada para admitir que se cargue previamente con información del plan de prueba, para permitir operaciones de escaneo sobre o extracción de memoria del objetivo, y para almacenar resultados o contenido de memoria para un enlace ascendente posterior a un ordenador host. El contenido de la memoria puede descargarse a través del puerto externo del instrumento de escaneo (por ejemplo, USB) y/o cargarse utilizando elementos de memoria extraíbles enchufables o mediante otras técnicas similares. La carga a través del puerto externo se puede realizar para recuperar resultados de prueba almacenados o contenido de memoria almacenado en elementos de memoria. Esto admite una implementación altamente portátil, en la que el acceso al circuito objetivo puede ser difícil o remoto al evitar la necesidad de conectar un ordenador durante el escaneo de prueba. Dicho procesador incorporado también puede incluir firmware que le permite controlar un dispositivo portátil o móvil conectado, tal como un teléfono inteligente (véase, por ejemplo, la Figura 4B) para operar sus recursos de la misma manera que una consola de E/S de usuario. Esto podría permitir un mejor control y visibilidad de las operaciones del instrumento de escaneo mediante la explotación de dispositivos móviles, teléfonos, teléfonos inteligentes, etc., disponibles habitualmente llevados normalmente por personal relevante.

La Figura 5C es un diagrama de bloques de un instrumento de escaneo de factor de forma de unidad flash local, que ilustra elementos funcionales para realizar pruebas/escaneo/extracción de acuerdo con ciertas implementaciones. Esta versión puede incluir un conector 585 USB para acoplar directamente a un zócalo de USB, lo que evita la necesidad de un cable entre sí mismo y este procesador host. También puede contener memoria 590 (incorporada o extraíble) para permitir la carga o descarga hacia/desde un componente informático local o plataforma de planes de prueba almacenados y/o software de aplicación de prueba. En el caso de carga, los archivos del plan de prueba y/o parte o la totalidad del software de prueba y/o los datos se pueden cargar previamente en la fábrica o en el depósito antes de enchufar en la ubicación de prueba. Por el contrario, dicha memoria también se puede descargar o recargar con nuevos archivos de plan de prueba o software de prueba desde los componentes locales a los que se conecta, que incluye aquellos recibidos desde un servidor conectado de forma remota a través de una red. Dicha carga puede ocurrir en un momento separado de la prueba como una etapa de preparación. Esto admite una gran portabilidad y agilidad para el instrumento de escaneo, y también permite el transporte portátil fácil de los planes de prueba a un sitio de prueba local en el que el acceso a la red puede no estar disponible. Adicionalmente, si está

precargado con el software de la aplicación de prueba, que se puede cargar en cualquier componente local enchufado, tal como una PC, puede aprovisionar cualquier componente local, incluso si nunca se inicializa con el software de prueba, para, por ejemplo, ser inmediatamente capaz de escaneo de dispositivos objetivo. Esta modalidad puede servir para transportar software, información de prueba y datos objetivo en ambas direcciones hacia/desde una PC.

Las Figuras 6A y 6B son diagramas que ilustran un proceso de escaneo de depuración JTAG ejemplar en los diversos niveles en el sistema de conexión, que incluye el procesamiento de aplicaciones en el servidor remoto, el instrumento de escaneo y el circuito objetivo, de acuerdo con una o más implementaciones divulgadas. Esta Figura representa un proceso del servidor 602 remoto (Figura 6A) que lanza una instalación 614 dada (por ejemplo, la instancia J) (Figura 6B) para realizar una extracción de validación de firmware utilizando su proceso de instrumento 614 de control de escaneo, a través de la red. Este proceso de escaneo múltiple se puede aplicar a uno o más circuitos (por ejemplo, número de circuito CIR a LAST_CIR) en la instalación, cada uno de cuyos firmwares se va a validar. A su vez, cada circuito puede tener uno o más bloques no contiguos de contenido de firmware (ya sea no volátil... por ejemplo Flash o volátil... por ejemplo RAM) para recopilar e incluir en dicha validación. Dichos bloques (cada uno de un tamaño de bloque dado), cada uno que comienza en una dirección de memoria particular, se combinarán para producir una firma para el enlace ascendente, para su circuito particular. La extracción se producirá en incrementos de al menos una hasta palabras de recuento por fragmentos para recopilar desde el ciclo de detención/ejecución del objetivo por objetivo mínimo. Para cada interacción con el objetivo (envío de comandos, depósito de datos o extracción de datos), el instrumento realizará uno o más escaneos JTAG de un número de desplazamientos de bits.

Un proceso de validación ejemplar, como se muestra en la presente memoria, puede comenzar desde un estado 602 inicial, no escaneado (Figura 6A) como presente en el componente remoto o servidor. Cuando se activa 604 un inicio para el proceso de instalación/validación, el servidor remoto emite una instrucción 608 de validación de lanzamiento a través de una red 610 o enlace de comunicación entre el servidor y los otros elementos del sistema. Luego se transmite 612 el comando de lanzamiento al instrumento de escaneo, que puede estar en un estado 614 inicial del proceso de validación. Cuando se recibe 616 el comando de lanzamiento, el instrumento de escaneo identifica y selecciona el circuito 620 en el que se realizará el primer escaneo/análisis. A continuación, el instrumento de escaneo determina las direcciones que se deben leer 622 y validar, que incluyen la identificación o carga de listas de bloques de direcciones objetivo y la información de nodo de CPU necesaria para realizar esos escaneos 624 particulares. Cada bloque se compone de una dirección de inicio y un número contiguo de palabras que se van a extraer, que abarcan un recuento de tamaño de bloque relacionado. Las extracciones de bloque se segmentan en fragmentos de una o más palabras que requieren una cierta duración de segmento de tiempo cada una, dependiendo de las tasas de escaneo y las características objetivo. Cada dicho segmento de tiempo de fragmento se intercala con períodos significativos de ejecución normal de código de aplicación de CPU, con una relación seleccionada para minimizar el impacto en el rendimiento. Se repiten varias extracciones de palabras de recuento de fragmentos dispersos hasta que se hayan extraído todos los bloques. En la presente memoria, y a lo largo de las siguientes secuencias de prueba, el instrumento de escaneo se puede configurar para transmitir los vectores 625 de escaneo JTAG a los circuitos que se van a probar.

Volviendo al proceso representativo mostrado en la Figura 6B, para cada fragmento de extracción que se va a realizar, el instrumento escanea un comando 626 de detención para parar la CPU objetivo después de su ejecución de instrucción actual. En el ejemplo ilustrado, este comando se escanea en el puerto 628 de depuración (JTAG) de una CPU determinada mientras ejecuta su código de aplicación 630. En la presente memoria, por ejemplo, la CPU puede haber estado ejecutando ciertas instrucciones 632 de aplicación a medida que se transmiten varios comandos 633 de depuración, tal como cuando el comando de detención se escanea en 634. Al recibirlo, la CPU para la ejecución de las instrucciones 636 de aplicación después del uno en curso como se representa en la Figura 7. Si un comando de detención no es 634, se obtiene en 638 la siguiente instrucción de aplicación.

Con referencia nuevamente a la Figura 6B, el instrumento de escaneo obtiene entonces una o más palabras 640 del contexto de la CPU durante el cese momentáneo del procesamiento de la aplicación, normalmente contenido en los registros del dispositivo. En la presente memoria, se puede proseguir de acuerdo con varias ramificaciones para adquisición de código dependiendo de si se debe acceder o extraer los datos de la memoria 642 no volátil o volátil. Se accede a las palabras de datos ejecutables y se guardan, por ejemplo, desde ubicaciones 646 de RAM relevantes. Si se debe acceder a nodos no volátiles, entonces se accede a las palabras de datos y se guardan, por ejemplo, desde la memoria 644 flash. Después de obtener los datos de la memoria volátil o no volátil, se restaura 648 el contexto de la CPU y se escanea el comando de ejecución para que el procesamiento de la CPU de la(s) aplicación(es) pueda volver a un estado 650 de ejecución. Al igual que con otros comandos de depuración que se transmiten 655, los circuitos objetivo recibirán este comando 654 de ejecución, en respuesta a lo cual la CPU del circuito objetivo vuelve a la operación normal, por ejemplo, buscando la siguiente instrucción 638 en el bucle habitual de procesamiento de las instrucciones de la aplicación.

Una vez que se restablece la operación de CPU normal, el instrumento de escaneo confirma sí o no las últimas direcciones escaneadas fueron las direcciones finales que se van a leer para los circuitos que se están analizando 618. Si no es así (si hay direcciones adicionales para escanear), el instrumento de escaneo se configura para escanear por sí mismo las siguientes direcciones para los circuitos en cuestión en la secuencia 660 de prueba. En la

presente memoria, una vez que las siguientes direcciones en la secuencia de prueba se cargan/configuran para escanear, el instrumento de escaneo vuelve a la etapa de dimensionamiento 624 de bloque en el proceso de comenzar la parada de la CPU y la rutina de recopilación de datos para obtener de ese modo los datos para la siguiente secuencia de prueba de dirección. Este bucle de actualización de las direcciones que se van a escanear 660 y realizar la rutina de parada y recopilación de las direcciones y bloques actualizados se realiza hasta que se recopilan todos los datos deseados para las direcciones identificadas. Una vez que se recopilan los datos finales para cada subcomponente de direcciones escaneadas o las direcciones totales deseadas, el instrumento de escaneo puede calcular información, tal como una firma, con respecto a todos los datos recopilados que se van a validar 662 para la transmisión 664 a través de la red.

Una vez que se escanean todas las direcciones para un circuito particular, el instrumento de escaneo confirma sí o no el último circuito escaneado fue o no el circuito final para ser leído 668 para el presente procedimiento de prueba. Si no (si hay circuitos adicionales para escanear), el instrumento de escaneo inicia entonces un proceso de escaneo del siguiente circuito 670, en el procedimiento de prueba que se está realizando. En la presente memoria, una vez que se verifica el siguiente circuito para escanear, el instrumento de escaneo vuelve a la etapa de confirmar las direcciones que se escanearán 622, antes de comenzar el proceso de bucle general de realizar la parada de la CPU y la rutina de recopilación de datos para cada dirección y/o bloque de los mismos, para obtener así los datos necesarios para completar el procedimiento de prueba. Si, en 668, se confirma que el circuito escaneado es el último circuito que se va a escanear para el procedimiento de prueba instantánea, el instrumento de escaneo vuelve a un estado 614 de espera para esperar la recepción del siguiente comando 612 de lanzamiento para comenzar un procedimiento de prueba posterior.

Una vez que se reciben 672 una o más firmas (Figura 6A), se puede realizar 674 una operación de comparación de firma para determinar si o no las firmas recibidas verifican que los circuitos bajo escaneo poseen las configuraciones deseadas. Si bien este proceso de verificación se ilustra en el contexto del servidor en las Figuras 6A y 6B, dichos procesos de verificación también se pueden realizar en el instrumento de escaneo como se establece en otra parte en la presente memoria.

Si las firmas verifican con éxito que los circuitos poseen las configuraciones deseadas, el componente o servidor informático remoto puede registrar una etiqueta y/o generar indicaciones de que los circuitos o la instalación que se verifica tienen la configuración 676 correcta/deseada. Por el contrario, si las firmas verifican una configuración u operación incorrecta, el componente o servidor informático remoto puede registrar una etiqueta o generar indicaciones de que los circuitos o la instalación bajo verificación no han podido demostrar la configuración 678 correcta/deseada. Finalmente, el servidor o componente informático remoto confirma sí o no la instalación bajo prueba fue la instalación final para escanear 680. De lo contrario, el componente o servidor remoto determina la próxima instalación para probar 682 y continúa con la etapa de verificar si o no se ha producido un estado/evento de inicio, en el que se inician 604 los comandos para lanzar la validación de la próxima instalación. Si, en 680, se confirma que la instalación que se acaba de validar fue la última instalación que se procesó, el componente remoto o el servidor regresa al estado 602 inicial inactivo para esperar el próximo tiempo de validación programado.

Como se indicó anteriormente, un instrumento adjunto puede operar las características del puerto de depuración del procesador interno (CPU) mediante la transmisión de secuencias de control de comportamiento utilizando los patrones de señal desplazados, de acuerdo con los protocolos de JTAG IEEE-1149.1, IEEE-1149.4, IEEE-1149.5, IEEE-1149.6, IEEE-1149.7 o similares. Adicionalmente, dada la arquitectura de encadenamiento tipo margarita de este protocolo, múltiples puertos, de un circuito dado y/o varios de dichos circuitos en un sistema, se pueden conectar en cadena junto con unas pocas señales que proporcionan acceso a cada uno de ellos en serie. Por medio de este enfoque, un instrumento conectado puede inyectar comandos y parámetros de la CPU, mientras extrae el estado y los datos, de manera independiente y concurrente con los otros portales de entrada/salida de la aplicación de cada microordenador. En la presente memoria, algunas acciones de puerto de depuración ejemplares que se pueden emplear para controlar el procesador interno (CPU) para este propósito, incluyen: (1) detener la ejecución del tiempo de ejecución del programa de aplicación entre instrucciones; (2) extraer el contexto de la CPU (tales como los valores de registro); (3) leer las ubicaciones de firmware direccionadas nuevamente al instrumento de prueba/escaneo; (4) restaurar el contexto de la CPU (como valores de registro); y (5) reanudar la ejecución del tiempo de ejecución de la aplicación siguiendo la instrucción detenida.

Como tal, un instrumento de escaneo externo puede ordenar a cada microordenador que detenga la ejecución de la aplicación entre instrucciones, lea y guarde el contexto actual del microordenador, lea el contenido de ciertas ubicaciones visibles (específicamente una o más palabras de almacenamiento del programa) transmitidas *ala* instrumento externo para recopilación y continúe ejecutando el firmware de aplicación desde el punto de detención con un contexto restaurado. Mediante dichas técnicas, un instrumento de escaneo externo puede acceder y adquirir almacenamiento de firmware de forma incremental mientras permite que la ejecución de la aplicación se ejecute la mayor parte del tiempo sin obstáculos. Un impacto en el circuito validado es el cese de la ejecución del procesador entre dos instrucciones sucesivas, durante un breve segmento de tiempo de intervalo de depuración.

La Figura 7 es una ilustración esquemática que muestra características de temporización y extracción de firmware ejemplares de acuerdo con una implementación divulgada. Con referencia a la Figura 7, se muestra un diagrama que ilustra aspectos de la extracción de temporización y firmware mapeados contra el procesamiento de la

aplicación de la CPU (a medida que avanza en el tiempo en la dirección X). La Figura 7 representa un proceso 710 de CPU que muestra estados de procesamiento de CPU objetivo antes y durante los procesos de parada y lectura. Como se ilustra, el proceso 710 de CPU representa los períodos de procesamiento de CPU, que incluyen los estados de operaciones 712 de ejecución de aplicaciones de CPU estándar, así como los estados cuando la CPU se detiene momentáneamente 714 al recibir un comando de detención. Este último puede comenzar con intervalos de segmento de tiempo de detención/ejecución escasamente inyectados, que proporcionan espacios de operación (entre períodos de ejecución de aplicación) de corta duración. Dichos comandos inducen períodos de latencia/muerte objetivo dispersos que permiten la adquisición de incrementos de datos deseados sin una interrupción sustancial de la operación general del programa/sistema. Adicionalmente, un segmento 716 de tiempo, en la presente memoria, se puede ver como demarcar el inicio de un “período de extracción”, que se refiere a un ciclo que comienza con el inicio de un primer estado 714A de parada de CPU, que continúa a través de un período de procesamiento 712 de aplicación de CPU, y que finaliza al comienzo del siguiente estado 714B de detención de CPU.

Cada estado 714 de detención de CPU constituye un segmento 722 del tiempo de latencia de muerte de la CPU durante el cual las características de escaneo de prueba establecidas en la parte inferior de la Figura 7 ocurren, de acuerdo con una implementación divulgada. Como se muestra en la Figura 7, cada segmento 722 de tiempo ejemplar puede comprender una pluralidad de escaneos 724, 726 que ocurren entre una primera instrucción de aplicación N 720 y la instrucción de aplicación posterior N+1 728. En una implementación, la pluralidad de escaneos puede estar compuesta de escaneos 724 de información de control, dirigida en tareas de información de control como detener o iniciar la CPU y guardar o restaurar los contextos de los diversos registros, y escaneos 726 de carga útil, dirigidos a extraer una cantidad de fragmentos de bytes de contenido del firmware (flash o RAM). Cada período de escaneo también puede incluir una comunicación asociada de escaneos correspondientes de información 740 entre el instrumento 742 de escaneo y el puerto de depuración JTAG objetivo, mediante el desplazamiento de múltiples bits de escaneo JTAG. En la implementación mostrada en la Figura 7, un segmento 722 de tiempo ejemplar puede incluir: uno o más primeros escaneos 724A de información de control compuestos de comandos y datos para detener la CPU y extraer contextos de los registros; uno o más escaneos 726 de carga útil compuestos de comandos y datos para extraer una cantidad de fragmentos de bytes de contenido de firmware; y uno o más escaneos 724B de información de control adicionales compuestos de comandos y datos para restaurar contextos a los registros y reiniciar el estado de ejecución de la CPU.

Los intervalos de detención/ejecución de CPU consistentes con las innovaciones en la presente memoria pueden, en algunas circunstancias, inducir períodos muertos/de latencia dispersos en la circuitería o CPU objetivo. Un impacto de dichos puntos muertos o retardos es que el comportamiento preciso en tiempo real se puede alterar ligeramente. El impacto de los períodos de muerte/latencia dispersos depende en gran medida de la arquitectura y la funcionalidad de las aplicaciones objetivo y objetivo. Se ve un impacto mínimo, por ejemplo, en aplicaciones compuestas de numerosas tareas concurrentes que comparten la CPU en momentos aleatorios. Considerando que, el mayor impacto puede ser en aplicaciones que son sensibles a la capacidad de respuesta de procesamiento en tiempo real. En la presente memoria, sin embargo, se puede utilizar un conocimiento detallado de los objetivos para superar o acomodar estos problemas. Por ejemplo, los ciclos de trabajo y los períodos de detención/ejecución de los algoritmos se pueden ajustar para minimizar la latencia y el impacto en el rendimiento de acuerdo con los límites del hardware. Tenga en cuenta que el intervalo muerto/latencia puede degradar la respuesta de entrada de rutina del controlador de interrupción. Esto se debe a eventos externos en tiempo real o señales de temporizador que afectan la reacción del proceso de una manera algo retardada durante dichos estados detenidos. Adicionalmente, las implementaciones se pueden configurar para incluir el aplazamiento del proceso resultante en el peor de los casos para evitar efectos inaceptables. De acuerdo con algunas realizaciones de innovaciones en la presente memoria, los períodos de latencia de detención/ejecución ejemplar que utilizan parámetros seleccionados pueden ser de aproximadamente 1 milisegundo. En la presente memoria, los ‘parámetros seleccionados’ se refieren a la reducción de la latencia del tiempo muerto al maximizar la tasa de reloj JTAG SCL (de acuerdo con la capacidad del objetivo y del instrumento de escaneo), minimizar el almacenamiento de contexto/restaurar la información y minimizar el recuento de extracción de fragmentos. También implica la optimización de la relación de detención/ejecución de extracción con respecto al tiempo de período general para reducir el impacto en el rendimiento, mientras se evita que el tiempo de ciclo general de la firma se vuelva poco práctico. En una implementación de ejemplo práctica, los datos escaneados se pueden adquirir a una tasa/cantidad de aproximadamente 128 bytes de fragmentos por ciclo. De acuerdo con dichas realizaciones, se pueden emplear de manera útil períodos entre extracciones de aproximadamente 30 milisegundos por fragmento de detención/ejecución.

Dicho período de detención, aunque breve, puede introducir problemas que se van a resolver dada la posible alteración del flujo en tiempo real. Dichas consideraciones pueden incluir efectos de procesos aleatorios fuera del procesador, que incluyen los eventos del temporizador, que transcurren durante los intervalos detenidos. Estos pueden continuar para generar interrupciones, cuya respuesta se difiere debido al estado detenido, que retrasan la entrada a las rutinas de manejador. Para resolver dichos problemas, las innovaciones en la presente memoria se pueden basar en un conocimiento íntimo de los circuitos y los algoritmos asociados para minimizar las repercusiones en la aplicación de tiempo de ejecución, según lo controlado por el ciclo de trabajo de detención/ejecución. Por ejemplo, minimizar los tiempos de información de control la maximización de la velocidad de escaneo (reloj SCL más rápido) y minimizar el tamaño del fragmento, al tiempo que aumenta el período de extracción para reducir el impacto

en el rendimiento (a expensas del tiempo general de creación de firma). Adicionalmente, se pueden emplear algoritmos particulares del ciclo de trabajo de detención/ejecución para parar y acceder a los datos en función del microordenador y el hardware, las características y/o la funcionalidad general del sistema.

En otras realizaciones, de acuerdo con los problemas de rendimiento objetivo anteriores, el microordenador objetivo se puede mantener detenido solo durante breves intervalos durante períodos ampliamente dispersos para minimizar el impacto de ejecución en primer plano. En la presente memoria, el algoritmo de ciclo de trabajo del tiempo de detención/ejecución para el período de extracción se intercambiaría contra la tasa de rendimiento del ciclo general de recopilación de firmware. Por ejemplo, un algoritmo de extracción de objetivos menos intrusivo y disruptivo puede dar como resultado una producción de firmas y un tiempo de carga más largos, para ser intercambiados. De acuerdo con lo anterior, la ingeniería y la fabricación de instrumentos de escaneo y/o algoritmos asociados que implementan dichas innovaciones pueden incluir un proceso de optimización que tenga en cuenta los efectos adversos del tiempo de intervalo detenido relativo versus el tiempo mínimo para completar la validación.

Extensión para validación de hardware de circuito general

Las innovaciones actuales son compatibles bajo muchas circunstancias con las pruebas de hardware heredadas según lo soportado por JTAG. Como tal, los sistemas y procedimientos en la presente memoria pueden incluir dichas validaciones de circuito extendido. Por ejemplo, se pueden escanear los circuitos vinculados en la cadena que no contiene puertos de depuración del microordenador, aunque incluyen el acceso JTAG del hardware de escaneo de límites. Por lo tanto, un circuito que expone un puerto de depuración JTAG de microordenador puede incluir otros componentes de la cadena que no sean de microordenador. En general, una cadena JTAG podría unir placas de circuitos, componentes y sistemas que tengan una combinación de elementos, que incluyen puertos de depuración de microordenadores y dispositivos digitales en general.

El protocolo JTAG puede enfocar todos los elementos de la cadena en cualquier punto, ya sea un puerto de depuración de microordenador u otro componente con características JTAG heredadas, en secuencia. En algunas implementaciones, la inclusión de dichos elementos escalables y sin depuración también permite la verificación automática de la integridad del hardware, es decir, más allá de la aplicación principal en la presente memoria de validación de firmware.

Esta última actividad extendida (prueba de escaneo de límites heredada) requiere normalmente que los dispositivos se pongan en un estado de prueba no funcional dedicado que permita la configuración y detección de patrones de señales conectadas por pin arbitrarias. Esto abre todo el mundo de las capacidades de JTAG, tal como comprobar las aperturas/cortos de señal, los dispositivos de trabajo/instalados, los dispositivos programables de carga/subida (EEPROM, CPLD, PLD, memorias FLASH, etc.), sin la participación de ninguna ejecución de microordenador sobre el objetivo. Dichas capacidades mejoradas se incluyen en ciertos aspectos de las innovaciones actuales, por ejemplo, en las que se puede deshabilitar la funcionalidad de la aplicación objetivo. Las pruebas generales adicionales de JTAG, más allá del esquema de validación de firmware en cuestión, también pueden incluir procedimientos de puerto de depuración de la CPU mediante los cuales el procesador objetivo se opera en un modo de emulación. Esto incluye que el controlador de escaneo externo, bajo la dirección del host, ordene las operaciones de la CPU para probar ampliamente la placa de circuito en cuestión, que incluye la actividad de e/s con todos los recursos visibles, ejecutando código a toda velocidad (más rápido que el escaneo de límites JTAG heredado), ejecutando las rutinas de prueba temporalmente descargadas (con o sin asistencia del firmware incorporado) y la programación o carga de dispositivos programables en el espacio de CPU direccionable.

Realizaciones adicionales

Realizaciones adicionales pueden implicar sistemas en los que los diferentes protocolos no JTAG (por ejemplo, BDM, SPI, MDIO o RS-232) son admitidos con el puerto de depuración de microordenador de un circuito dado. En la presente memoria, los sistemas variantes de protocolo específicos del dispositivo compatibles con las innovaciones actuales pueden utilizar un instrumento con una interfaz operada de acuerdo con sus requisitos. Dicha variación de los protocolos de comando en serie se puede acomodar para realizar la extracción de firmware anterior para el proceso de validación, utilizando patrones de desplazamiento alterados adecuadamente (en oposición a JTAG). Se pueden operar para realizar la extracción del segmento de tiempo de la CPU en secuencias similares. En la presente memoria, los instrumentos se pueden conectar por separado y simultáneamente a la red junto con otros tipos de instrumentos (tales como el tipo JTAG anterior) para que el servidor remoto los opere a su vez cuando debe validar este circuito conectado. Por ejemplo, dichos instrumentos se pueden operar y controlar a través de una red al distinguir cada uno de estos instrumentos de acuerdo con su dirección de red única (tal como la dirección IP de Internet). Por lo tanto, se puede implementar un conjunto de instrumentos de escaneo de tipo variable para acomodar los tipos de puertos de depuración alternativos necesarios para validar el conjunto completo de circuitos en su totalidad, cada uno con su instrumento conectado de manera apropiada. Aquellos puertos con protocolos que admiten una conexión en cadena (similar a JTAG anterior) se podrían conectar para compartir un instrumento dado, mientras que otros tendrían su propio instrumento dedicado conectado.

Funciones de seguridad agregadas

Además de validar el firmware de los circuitos integrados para asegurar su integridad ininterrumpida, las presentes innovaciones también admiten la detección de manipulaciones sospechosas de circuitos en el sitio de instalación. En la presente memoria, se puede detectar el manejo no autorizado o manipulación de dichos circuitos, por ejemplo, como parte de un posible esfuerzo de corrupción de firmware. Por ejemplo, dichos sistemas y procedimientos pueden involucrar características que responden a condiciones inesperadas, tales como desconectar, quitar o deshabilitar dicho circuito de su ajuste de configuración operable normal, etc., y también pueden incluir aspectos de señalización o comunicación relacionados con intentos potenciales de alteración. En algunas implementaciones, estas características pueden operar entre los ciclos de validación, monitoreando efectivamente el estado en operación de cada circuito para admitir un sistema de alerta. Al utilizar dichas características en operación, los sistemas y procedimientos en la presente memoria permiten esquemas de detección que permiten confirmar que la circuitería deseada se ejecute continuamente o se 'mantenga viva' dentro de los circuitos de interés instalados, de acuerdo con los intervalos de tiempo de ejecución programados.

Algunos procesos ejemplares que se pueden implementar simultáneamente con las operaciones de aplicación en ejecución incluyen realizar escaneos especiales de circuitos instalados a través de su cadena JTAG, por ejemplo, para confirmar su presencia y estado operativo. En el caso de la cadena JTAG, por ejemplo, el servidor remoto puede ordenar un escaneo de infraestructura estándar de protocolo de la cadena conectada para realizar una detección rápida y conveniente de todos esos circuitos conectados y operables. Este monitoreo no causa interrupciones en la aplicación en curso ya que los escaneos de infraestructura no interactúan con los pines de la aplicación del dispositivo. Para los protocolos que no son JTAG, se pueden implementar interacciones no disruptivas similares con los puertos de depuración del microordenador para lograr resultados similares.

Validación solo de sectores de firmware de interés

El firmware incorporado que se va a validar puede consistir en todo el recurso de almacenamiento de memoria sobre el circuito, o algún subconjunto del mismo. En la presente memoria, por ejemplo, el servidor puede identificar en el instrumento las regiones de dirección de interés que se validarán y las que se excluirán de dicho escaneo. Como tal, se pueden identificar y evitar sectores no programados, que no se preocupan o que alteran el tiempo de ejecución de dichos medios de almacenamiento, lo que permite centrarse solo en áreas de almacenamiento de datos con patrones relevantes estáticos. De acuerdo con lo anterior, el proceso/ciclos de validación se puede configurar para cubrir una secuencia de bloques de direcciones dentro de dichos dispositivos de almacenamiento de acuerdo con las regiones persistentes. A pesar de las discontinuidades, en la presente memoria, la generación de firmas o hash todavía proporciona una confirmación significativa de dichos bloques concatenados en una sola serie de valores, ya que los resultados son repetibles. En algunas implementaciones, se pueden emplear algoritmos de código hash basados solo en una serie dada de valores ordenados fijos, ya que estos algoritmos siempre resultan en un resultado dado independientemente de las agrupaciones de direcciones.

En la presente descripción, los términos componente, módulo, secuencia y unidad funcional, se pueden referir a cualquier tipo de proceso lógico o funcional o bloques que se pueden implementar de varias maneras. Por ejemplo, las funciones de varios bloques se pueden combinar entre sí en cualquier otro número de módulos. Cada módulo se puede implementar como un programa de software almacenado en una memoria tangible (por ejemplo, memoria de acceso aleatorio, memoria de solo lectura, memoria de CD-ROM, unidad de disco duro) para ser leída por una unidad central de procesamiento para implementar las funciones de las innovaciones en la presente memoria. O bien, los módulos pueden comprender instrucciones de programación transmitidas a un ordenador de propósito general o al hardware de procesamiento de gráficos a través de una onda portadora de transmisión. También, los módulos se pueden implementar como circuitos lógicos de hardware que implementan las funciones abarcadas por las innovaciones en la presente memoria. Finalmente, los módulos se pueden implementar utilizando instrucciones de propósito especial (instrucciones SIMD), matrices lógicas programables en el campo o cualquier combinación de los mismos que proporcione el nivel deseado de rendimiento y coste.

Como se divulga en la presente memoria, las realizaciones y características de la invención se pueden implementar a través de hardware, software y/o firmware de ordenador. Por ejemplo, los sistemas y procedimientos divulgados en la presente memoria se pueden realizar de diversas formas, que incluyen, por ejemplo, un procesador de datos, tal como un ordenador que también incluye una base de datos, circuitería electrónica digital, firmware, software o en combinaciones de los mismos. Adicionalmente, aunque algunas de las implementaciones divulgadas describen componentes tales como software, sistemas y procedimientos consistentes con las innovaciones en la presente memoria incluidas, se pueden implementar con cualquier combinación de hardware, software y/o firmware. Más aún, las características mencionadas anteriormente y otros aspectos y principios de las innovaciones en la presente memoria se pueden implementar en diversos entornos. Dichos entornos y aplicaciones relacionadas se pueden construir especialmente para realizar los diversos procesos y operaciones de acuerdo con la invención o pueden incluir un ordenador de uso general o una plataforma de computación activada selectivamente o reconfigurada por código para proporcionar la funcionalidad necesaria. Los procesos divulgados en la presente memoria no están relacionados inherentemente con ningún ordenador, red, arquitectura, entorno u otro aparato en particular, y se pueden implementar mediante una combinación adecuada de hardware, software y/o firmware. Por ejemplo, se pueden utilizar varias máquinas de propósito general con programas escritos de acuerdo con las enseñanzas de la invención, o puede ser más conveniente construir un aparato o sistema especializado para realizar los procedimientos y técnicas requeridos.

Los aspectos del procedimiento y sistema descritos en la presente memoria se pueden implementar como una funcionalidad programada en cualquiera de una variedad de circuitería, que incluye dispositivos lógicos programables (“PLD”), tales como matrices de compuerta programables en campo (“FPGA”), lógica de matriz programable (“PAL”), dispositivos lógicos y de memoria programables eléctricamente y dispositivos móviles estándar, así como circuitos integrados específicos de la aplicación. Algunas otras posibilidades para implementar aspectos incluyen: dispositivos de memoria, microcontroladores con memoria (tal como EEPROM), microprocesadores integrados, firmware, software, etc. Adicionalmente, se pueden incorporar los aspectos en microprocesadores que tienen emulación de circuito basada en software, lógica discreta (secuencial y combinatoria), dispositivos personalizados, lógica difusa (neural), dispositivos cuánticos e híbridos de cualquiera de los tipos de dispositivos anteriores. Las tecnologías de dispositivos subyacentes se pueden proporcionar en una variedad de tipos de componentes, por ejemplo, tecnologías de transistores de efecto de campo de semiconductores de óxido de metal (“MOSFET”) como semiconductores complementarios de óxido de metal (“CMOS”), tecnologías bipolares como lógica acoplada a emisor (“ECL”), tecnologías de polímeros (por ejemplo, estructuras de polímero conjugado con silicio y polímero-metal conjugado con metal), mezcladas análogas y digitales, y así sucesivamente.

También se debe tener en cuenta que las diversas funciones divulgadas en la presente memoria se pueden describir utilizando cualquier número de combinaciones de hardware, firmware y/o como datos y/o instrucciones incorporados en varios medios legibles por máquina o legibles por ordenador, en términos de su comportamiento, transferencia de registro, componente lógico y/u otras características. Los medios legibles por ordenador en los que se pueden incorporar dichos datos y/o instrucciones formateadas incluyen, pero no se limitan a, medios de almacenamiento no volátiles en diversas formas (por ejemplo, medios de almacenamiento ópticos, magnéticos o semiconductores) y ondas portadoras que se pueden utilizar para transferir dichos datos formateados y/o instrucciones a través de medios de señalización inalámbricos, ópticos o cableados o cualquier combinación de los mismos. Ejemplos de transferencias de dichos datos y/o instrucciones formateados por ondas portadoras incluyen, pero no se limitan a, transferencias (cargas, descargas, correo electrónico, etc.) a través de Internet u otras redes informáticas por medio de uno o más protocolos de transferencia de datos (por ejemplo, HTTP, FTP, SMTP, etc.).

A menos que el contexto requiera mucho lo contrario, a lo largo de la descripción y las reivindicaciones, las palabras “comprender”, “que comprende” y similares se deben interpretar en un sentido inclusivo en lugar de un sentido exclusivo o exhaustivo; es decir, en el sentido de “incluir, pero no limitarse a”. Las palabras que utilizan el número singular o plural también incluyen el número plural o singular respectivamente. Adicionalmente, las palabras “en la presente memoria”, “más adelante”, “anteriormente”, “a continuación” y palabras de apariencia similar se refieren a esta aplicación como un todo y no a porciones particulares de esta aplicación. Cuando la palabra “o” se utiliza en referencia a una lista de dos o más elementos, esa palabra cubre todas las siguientes interpretaciones de la palabra: cualquiera de los elementos de la lista, todos los elementos de la lista y cualquier combinación de los elementos de la lista.

Otras realizaciones de la invención serán evidentes para aquellos expertos en la técnica a partir de la consideración de la memoria descriptiva y la práctica de la invención divulgada en la presente memoria. Se pretende que la memoria descriptiva y los ejemplos se consideren solo a modo de ejemplo, que se indican por la divulgación anterior en combinación con los siguientes párrafos que describen el alcance de una o más realizaciones de la siguiente invención.

Los sistemas y procedimientos divulgados en la presente memoria se pueden implementar como un producto de programa informático, es decir, un programa informático materialmente incorporado en un soporte de información, por ejemplo, en un medio o elemento de almacenamiento legible por máquina o en una señal propagada, para ejecución por, o para controlar la operación de, aparato de procesamiento de datos, por ejemplo, un procesador programable, un ordenador o varios ordenadores. Un programa de ordenador se puede escribir en cualquier forma de lenguaje de programación, que incluye los lenguajes compilados o interpretados, y se puede implementar en cualquier forma, incluso como un programa independiente o como un módulo, componente, subrutina u otra unidad adecuada para su uso en un entorno informático. Un programa de ordenador se puede implementar para ejecutarse en un ordenador o en varios ordenadores en un sitio o distribuirse en múltiples sitios e interconectarse mediante una red de comunicación.

Se debe entender que la descripción anterior pretende ilustrar y no limitar el alcance de la invención, que se define por el alcance de las reivindicaciones adjuntas. Otras realizaciones están dentro del alcance de las siguientes reivindicaciones.

**REIVINDICACIONES**

1. Un procedimiento de validación remota de contenido de memoria sobre uno o más circuitos (350A, 350B, 350C, 350D) objetivo que ejecutan al menos una aplicación, que comprende:

asociar al menos un instrumento (120A, 120C) de escaneo JTAG con uno o más puertos (360A, 360C) de diagnóstico JTAG de un sistema de microordenador local que incluye al menos uno de los circuitos (350A, 350C) objetivo;

asociar al menos un instrumento (120B, 120D) de escaneo no JTAG con uno o más puertos (360D) de diagnóstico no JTAG del sistema de microordenador local que incluye al menos uno de los circuitos (350D) objetivo;

realizar un proceso de escaneo distinto de un proceso de escaneo de límites, a través de al menos un instrumento (120A, 120C) de escaneo JTAG y al menos un instrumento (120B, 120D) de escaneo no JTAG, a través del acceso a al menos un elemento de almacenamiento asociado con al menos uno de los circuitos (350A, 350B, 350C, 350D) objetivo a través de uno o más puertos (360A, 360C) de diagnóstico JTAG y uno o más puertos (360D) de diagnóstico no JTAG para obtener datos del sistema durante la operación de la aplicación o aplicaciones, respectivamente, operar sobre uno o más circuitos (350A, 350B, 350C, 350D) objetivo, el proceso de escaneo que incluye:

parar la aplicación o aplicaciones en ejecución, respectivamente, operar sobre uno o más circuitos (350A, 350B, 50C, 350D) objetivo; y

leer el contenido relacionado a la aplicación o aplicaciones en ejecución, respectivamente, desde al menos un elemento de almacenamiento en el proceso de escaneo distinto de un proceso de escaneo de límites para obtener datos de prueba mientras que se para la aplicación o aplicaciones, respectivamente;

reanudar la operación de la aplicación o aplicaciones en ejecución desde el estado en el que se pararon la aplicación o aplicaciones en ejecución, respectivamente;

transmitir información de descripción de los datos de prueba a un sitio remoto; y

comparar, en el sitio remoto, la información con información conocida para confirmar la configuración y/u operación del sistema deseada.

2. El procedimiento de la reivindicación 1, en el que al menos un elemento de almacenamiento comprende un dispositivo de almacenamiento volátil, y en el que los datos de prueba comprenden contenido relacionado con información que incluye instrucciones ejecutables almacenadas en el dispositivo de almacenamiento volátil.

3. El procedimiento de la reivindicación 1 en el que el proceso de escaneo incluye adicionalmente acumular información de que describe los datos de prueba como una pluralidad de fragmentos extraídos durante las paradas momentáneas.

4. El procedimiento de la reivindicación 1 en el que la parada de aplicación o aplicaciones en ejecución, respectivamente, incluye:

ordenar uno o más circuitos (350A, 350B, 350C, 350D) objetivo para detener la ejecución de la aplicación o aplicaciones en ejecución, respectivamente, entre instrucciones;

leer y guardar el contexto o contextos del circuito (350A, 350B, 350C, 350D) objetivo actual, respectivamente, de uno o más circuitos (350A, 350B, 350C, 350D) objetivo;

recolectar los valores de los datos almacenados en ubicaciones deseadas dentro de al menos un elemento de almacenamiento;

restaurar los contextos del circuito (350A, 350B, 350C, 350D) objetivo actual de uno o más circuitos (350A, 350B, 350C, 350D) objetivo; y

comenzar la ejecución de la aplicación o aplicaciones en ejecución, respectivamente.

5. El procedimiento de la reivindicación 1 en el que la información escaneada desde uno o más circuitos (350A, 350B, 350C, 350D) objetivo se valida localmente para confirmar la configuración u operación del sistema deseada, al compararla contra un patrón de referencia descargado de forma remota.

6. El procedimiento de la reivindicación 1 en el que un ordenador (370) de cliente local incluye un puerto dedicado conectado a al menos un instrumento (120C) de escaneo JTAG o al menos un instrumento (120D) de escaneo no

JTAG, el ordenador (370) de cliente local se conecta operativamente a una red (320) para alcanzar visibilidad de comunicaciones en los instrumentos (120C, 120D) de escaneo sobre la red (320).

7. El procedimiento de la reivindicación 1 en el que uno o más circuitos (350A, 350B, 350C, 350D) objetivo que se van a validar comprenden:

uno o más circuitos (350A, 350B, 350C, 350D) que se van a probar;

un microordenador configurado localmente con los circuitos (350A, 350B, 350C, 350D); y

patrones de código y firmware no confiable en uno o más circuitos (350A, 350B, 350C, 350D) y legibles por el microordenador;

en el que al menos un instrumento (120A, 120C) de escaneo JTAG y al menos un instrumento (120B, 120D) de escaneo no JTAG se conectan a un puerto (360A, 360B, 360C, 360D) de diagnóstico del microordenador para habilitar el acceso al firmware mínimamente disruptivo mientras se ejecuta.

8. El procedimiento de la reivindicación 1 en el que al menos un instrumento (120A) de escaneo JTAG y al menos un instrumento (120B) de escaneo no JTAG se conectan por separado a un entorno de red.

9. El procedimiento de la reivindicación 1 en el que al menos un instrumento (120A, 120C) de escaneo JTAG y al menos un instrumento (120B, 120D) de escaneo no JTAG se configuran para:

operar un puerto de diagnóstico de depuración en al menos uno de los circuitos (350A, 350B, 350C, 350D) objetivo;

leer el contenido relacionado con la aplicación o aplicaciones en ejecución, respectivamente, al extraer, a través del puerto (360A, 360B, 360B, 360D) de diagnóstico, fragmentos de memoria almacenados dentro de la circuitería del circuito (350A, 350B, 350C, 350D) objetivo;

ordenar el microordenador, cuando se paran la aplicación o aplicaciones en ejecución, respectivamente, para devolver los fragmentos de memoria; y

ordenar al microordenador continuar la ejecución de la aplicación o aplicaciones en ejecución, respectivamente;

en el que los fragmentos de memoria se extraen y recolectan sin alterar sustancialmente la secuenciación normal del microordenador, excepto durante intervalos de tiempo detenidos cuando se paran la aplicación o aplicaciones en ejecución, respectivamente.

10. El procedimiento de la reivindicación 9 en el que los fragmentos de memoria extraídos y recolectados comprenden código de firmware, y al menos uno de los instrumentos (120A, 120C) de escaneo JTAG o al menos uno de los instrumento (120B, 120D) de escaneo no JTAG se configura para procesar/computar un código de representación de firma derivado de dicho código de firmware extraído y recolectado de acuerdo con un algoritmo predefinido, y transportar dicha firma al servidor (150) remoto a través de la red (320).

11. El procedimiento de la reivindicación 1 comprende adicionalmente el manejo de la extracción/recolección mediante un servidor (150) remoto, que incluye:

comenzar una operación para extraer datos de prueba que incluyen firmware; y

transmitir la información asociada con el firmware, que incluye una firma representativa del firmware extraído, entre al menos uno de los instrumentos (120A, 120C) de escaneo JTAG o al menos uno de los instrumentos (120B, 120D) de escaneo no JTAG y el servidor.

12. El procedimiento de la reivindicación 1 comprende adicionalmente el manejo de la extracción/recolección por un servidor en el sitio remoto, que incluye:

comenzar una operación para extraer datos de prueba que incluyen firmware; y

realizar una validación local de dichos datos contra su configuración u operación deseada.

13. El procedimiento de la reivindicación 1 comprende adicionalmente el manejo de al menos uno de los instrumentos (120A, 120C) de escaneo JTAG por un servidor (150) remoto que incluye detectar un estado energizado y/o estado de circuitos (350A, 350C) conectados.

14. El procedimiento de la reivindicación 1 comprende adicionalmente:

modificar los datos de prueba a través de la inclusión de un valor aleatorio o patrón de marca de tiempo proporcionado por un servidor (150) remoto;

calcular un código de representación de firma derivado de los datos de prueba modificados; y

utilizar un algoritmo predefinido para producir variación verificable de resultados que se obtuvieron para cada instancia del código de representación de firma.

15. El procedimiento de la reivindicación 1 en el que al menos las porciones del proceso de escaneo se proporcionan mediante componentes electrónicos montados con/ubicados sobre un circuito (350A, 350B, 350C, 350D) que se va a validar, dentro de un montaje en el que se ubica el circuito (350A, 350B, 350C, 350D).

16. Un sistema de escaneo configurado para validar el contenido de memoria sobre uno o más circuitos (350A, 350B, 350C, 350D) objetivo que ejecutan al menos una aplicación, que comprende:

una pluralidad de interfaces (315) configuradas para conexión con uno o más puertos (360A, 360B, 360C) de diagnóstico JTAG y uno o más puertos (360D) de diagnóstico no JTAG de un sistema de microordenador local que incluye uno o más circuitos (350A, 350B, 350C, 350D) objetivo;

uno o más componentes (120A, 120C) JTAG y uno o más componentes (120B, 120D) no JTAG configurados para realizar un proceso de escaneo distinto de un proceso de escaneo de límites de al menos un elemento de almacenamiento asociado con uno o más circuitos (350A, 350B, 350C, 350D) objetivo a través de la pluralidad de interfaces (315) para obtener datos del sistema durante la operación de una aplicación o aplicaciones, respectivamente, operar sobre uno o más circuitos (350A, 350B, 350C, 350D) objetivo, el proceso de escaneo incluye:

parar la aplicación o aplicaciones en ejecución, respectivamente, operar sobre uno o más circuitos (350A, 350B, 350C, 350D) objetivo; y

leer el contenido relacionado a la aplicación o aplicaciones en ejecución, respectivamente, de al menos un elemento de almacenamiento (en el proceso de escaneo distinto de un proceso de escaneo de límites para obtener datos de prueba mientras que se paran la aplicación o aplicaciones, respectivamente;

reanudar la operación de la aplicación o aplicaciones en ejecución desde el estado en el que se pararon la aplicación o aplicaciones en ejecución;

transmitir información de descripción de los datos de prueba a un sitio remoto;

en el que la información transmitida se compara a la información conocida para confirmar la configuración u operación del sistema deseada de los circuitos (350A, 350B, 350C, 350D) objetivo.

17. El sistema de escaneo de la reivindicación 16 en el que uno o más componentes (120A, 120C) JTAG y uno o más componentes (120B, 120D) no JTAG se configuran para validar, a través de la comunicación con el sitio remoto, la información para confirmar la configuración u operación del sistema deseada de uno o más circuitos (350A, 350B, 350C, 350D) objetivo.

18. El sistema de escaneo de la reivindicación 16 comprende adicionalmente:

un servidor (150) remoto acoplado a través de una red (320) a uno o más componentes (120A) JTAG y uno o más componentes (120B) no JTAG y configurados para validar la información para confirmar la configuración u operación del sistema deseada de uno o más circuitos (350A, 350B, 350C, 350D) objetivo.

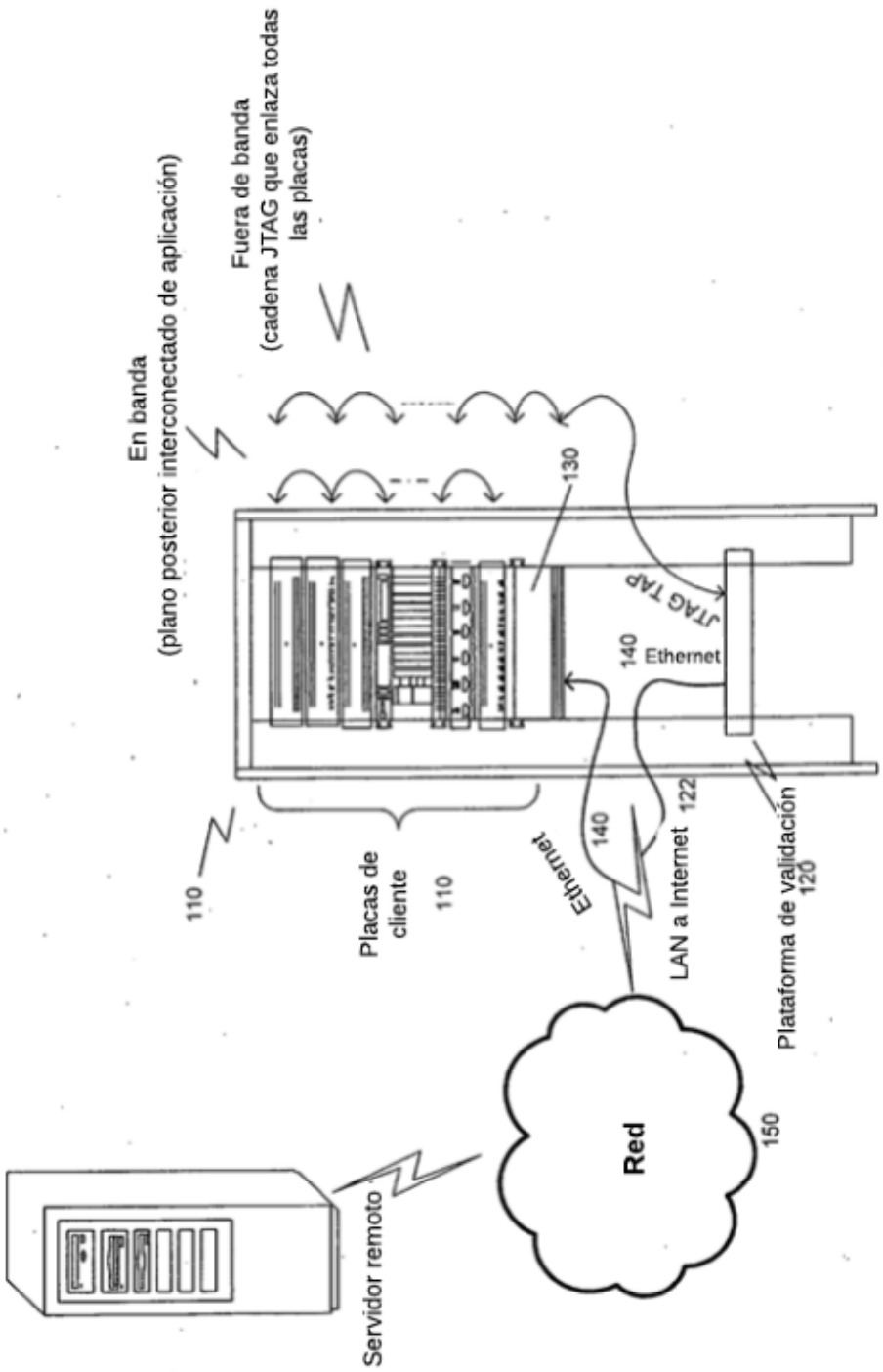
En banda
(plano posterior interconectado de aplicación)
Fuera de banda
(cadena JTAG que enlaza todas las placas)
130
JTAG TAP
140
Ethernet
110
Placas de cliente
110
Ethernet
140
122
LAN a Internet
Plataforma de validación
120
Red
150
Servidor remoto

Figura 1

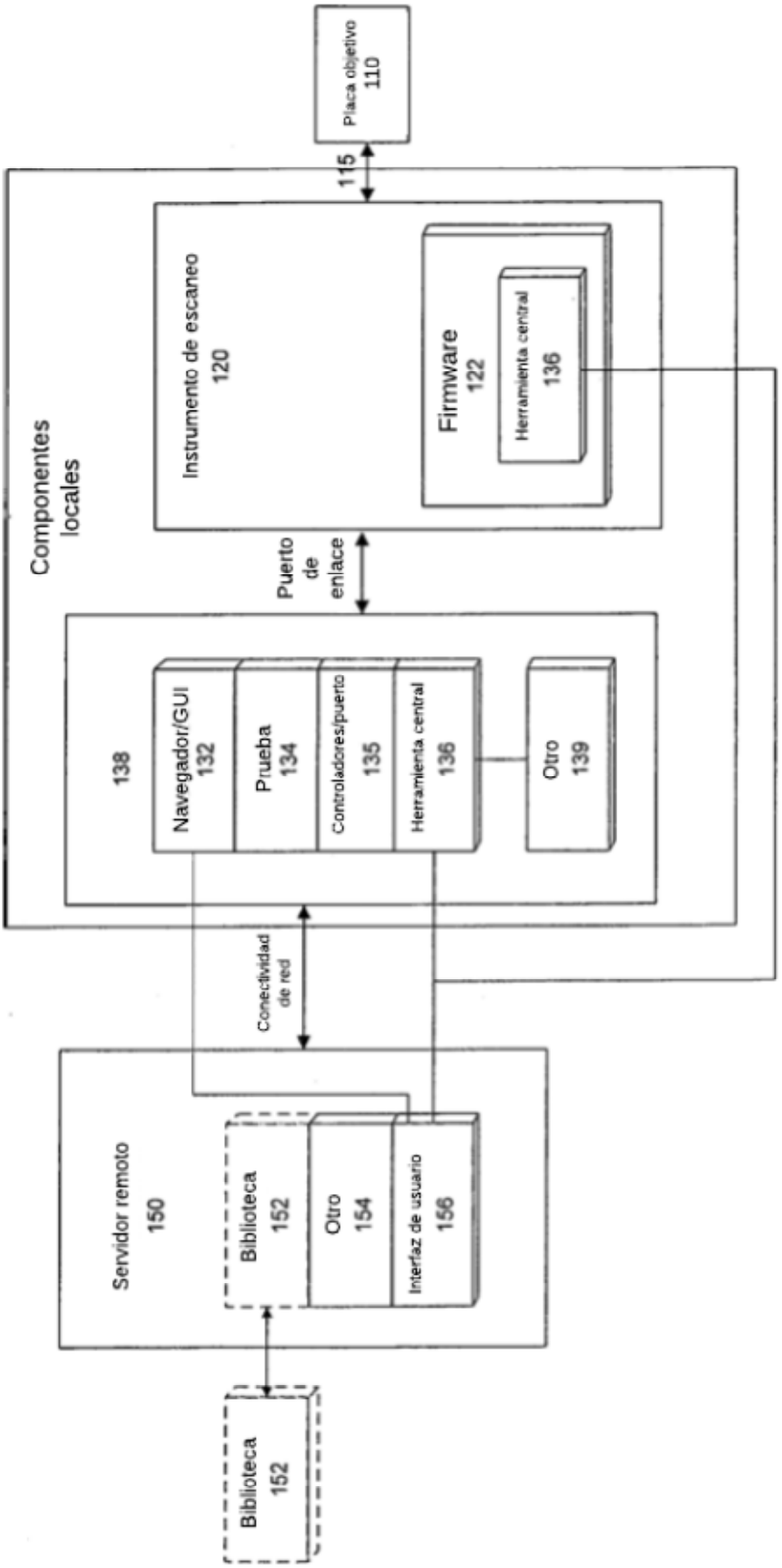
Placa objetivo
110
115
Componentes locales
Instrumento de escaneo
120
Firmware
122
Herramienta central
136
Puerto de enlace
138
Navegador/GUI
132
Prueba
134
Controladores/puerto
135
Herramienta central
136
Otro
139
Conectividad de red
Servidor remoto
150
Biblioteca
152
Otro
154
Interfaz de usuario
156
Biblioteca
152

Figura 2A

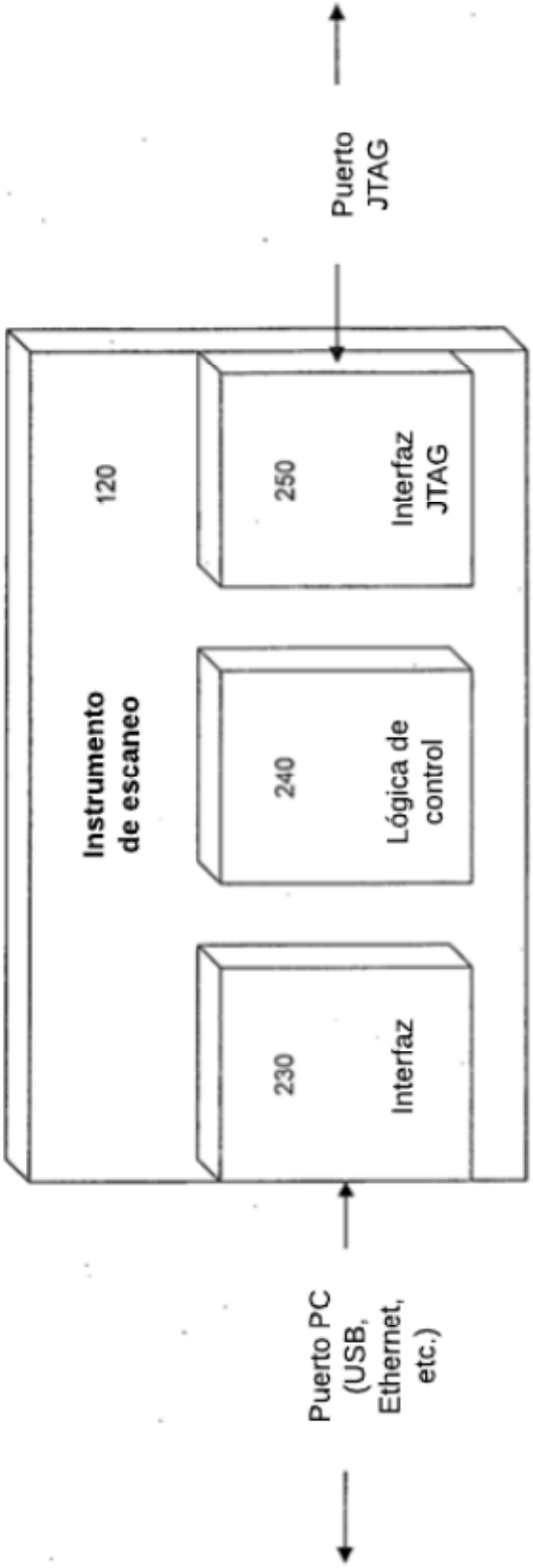
Puerto JTAG
120
250
Interfaz JTAG
Instrumento de escaneo
240
Lógica de control
230
Interfaz
Puerto PC (USB, Ethernet, etc.)

Figura 2B

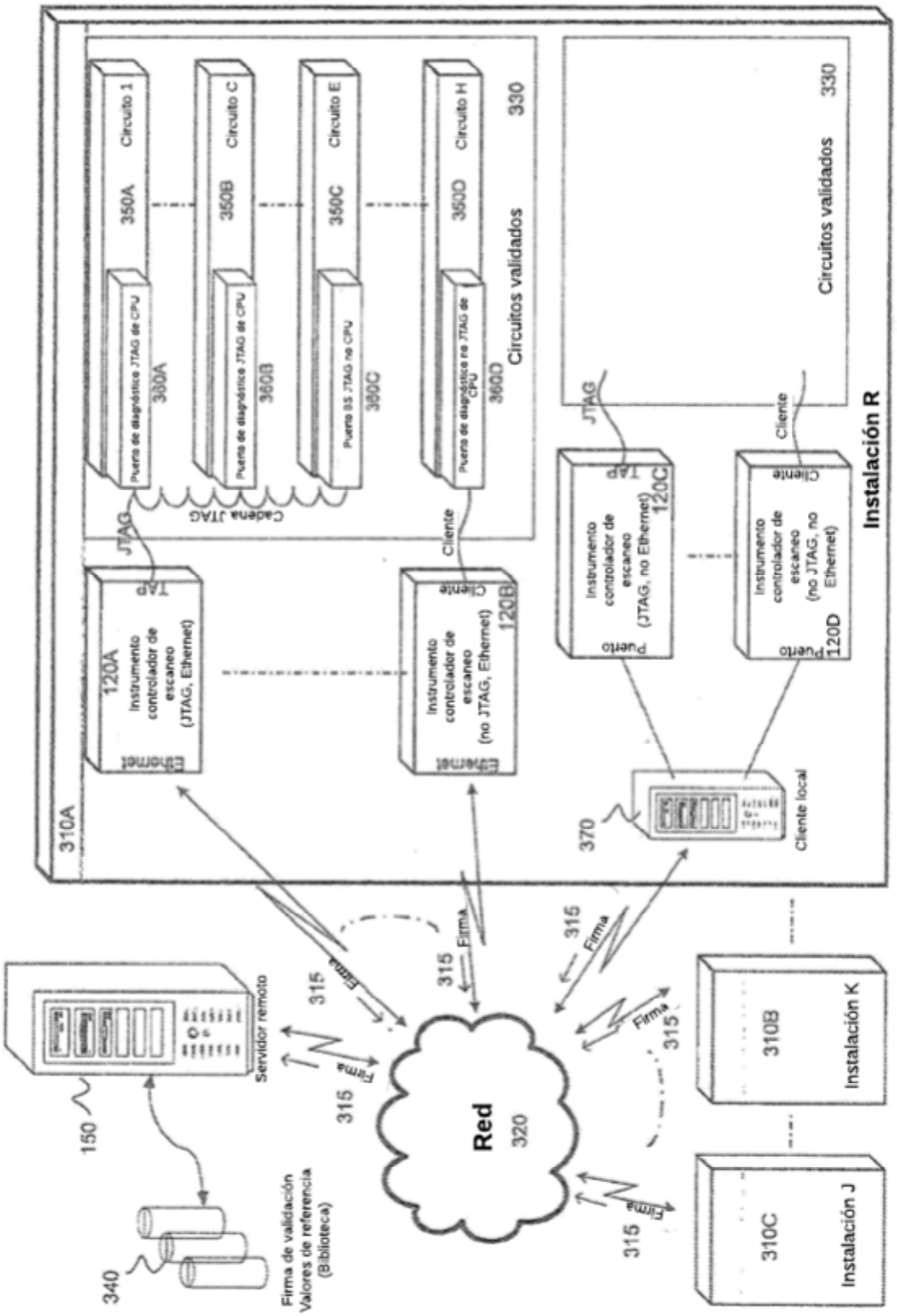


FIG. 3

Varias permutaciones de sistema de conexión de elementos

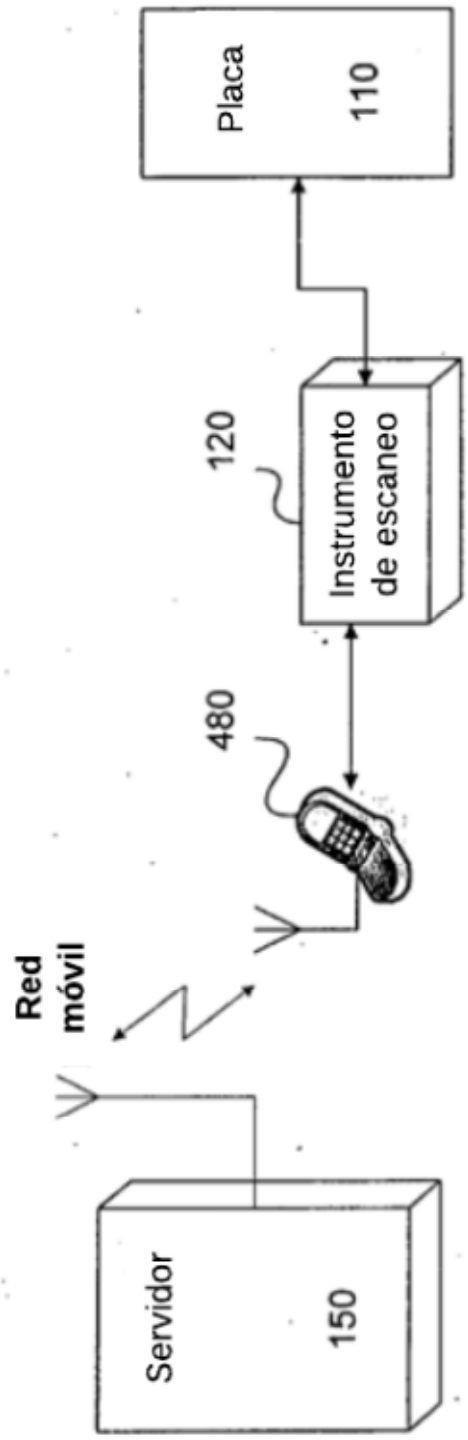
Placa
110
120
Instrumento de escaneo
480
Red móvil
Servidor
150

Figura 4

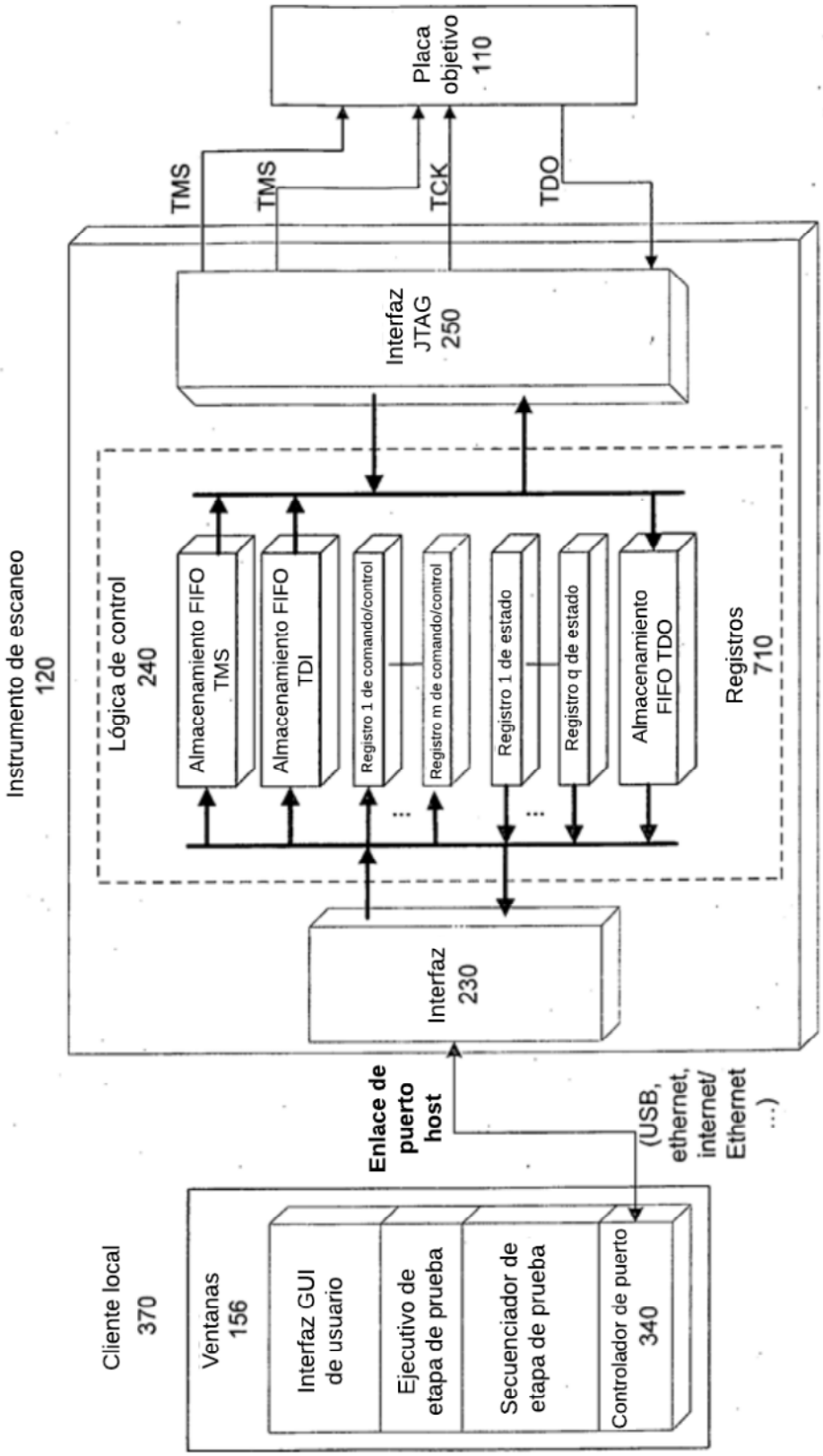
Placa objetivo 110
TMS
TMS
TCK
TDO
Instrumento de escaneo 120
Interfaz JTAG 250
Lógica de control 240
Almacenamiento FIFO TMS
Almacenamiento FIFO TDI
Registro 1 de comando/control
Registro m de comando/control
Registro 1 de estado
Registro q de estado
Almacenamiento FIFO TDO
Registros 710
Interfaz 230
Enlace de puerto host
(USB, ethernet, internet/ Ethernet ...)
Cliente local 370
Ventanas 156
Interfaz GUI de usuario
Ejecutivo de etapa de prueba
Secuenciador de etapa de prueba
Controlador de puerto 340

Figura 5A

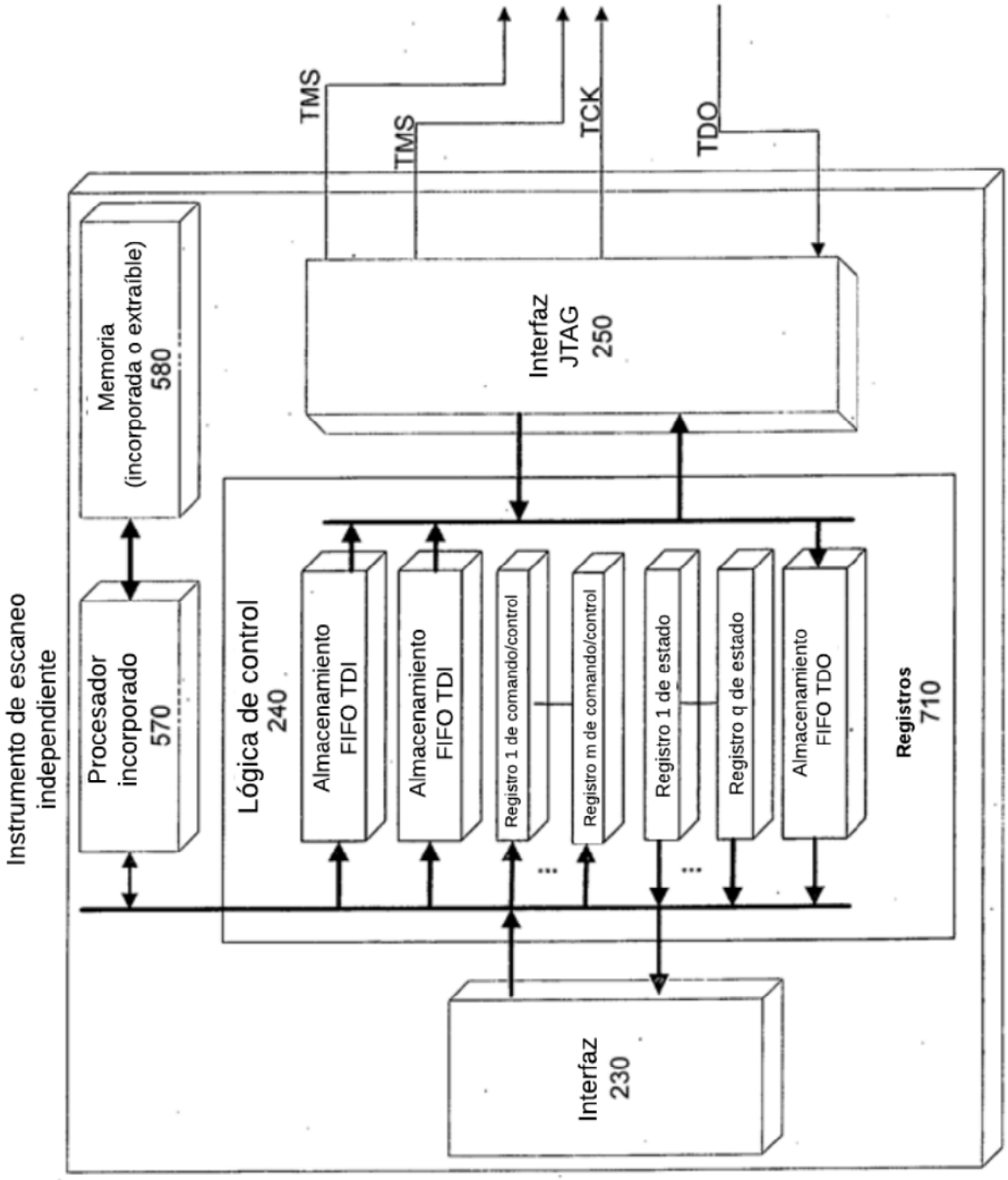
Instrumento de escaneo independiente
TMS
TMS
TCK
TDO
Memoria (incorporada o extraíble) 580
Procesador incorporado 570
Interfaz JTAG 250
Lógica de control 240
Almacenamiento FIFO TDI
Almacenamiento FIFO TDI
Registro 1 de comando/control
Registro m de comando/control
Registro 1 de estado
Registro q de estado
Almacenamiento FIFO TDO
Registros 710
Interfaz 230

**Figura 5B**

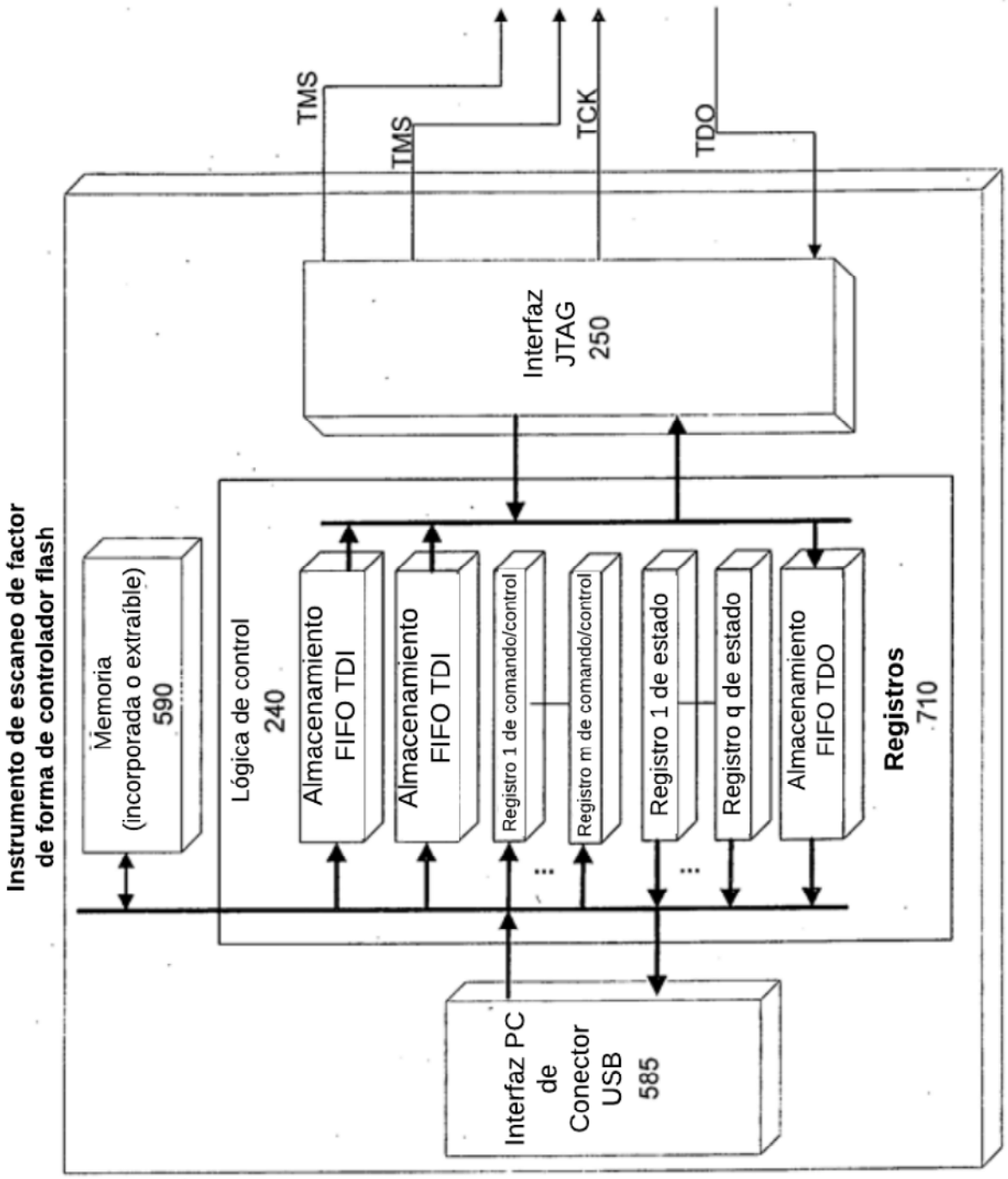
Instrumento de escaneo de factor de forma de controlador flash
TMS
TMS
TCK
TDO
Interfaz JTAG 250
Memoria (incorporada o extraíble) 590
Lógica de control 240
Almacenamiento FIFO TDI
Almacenamiento FIFO TDI
Registro 1 de comando/control
Registro m de comando/control
Registro 1 de estado
Registro q de estado
Almacenamiento FIFO TDO
Registros 710
Interfaz PC de Conector USB 585

Figura 5C

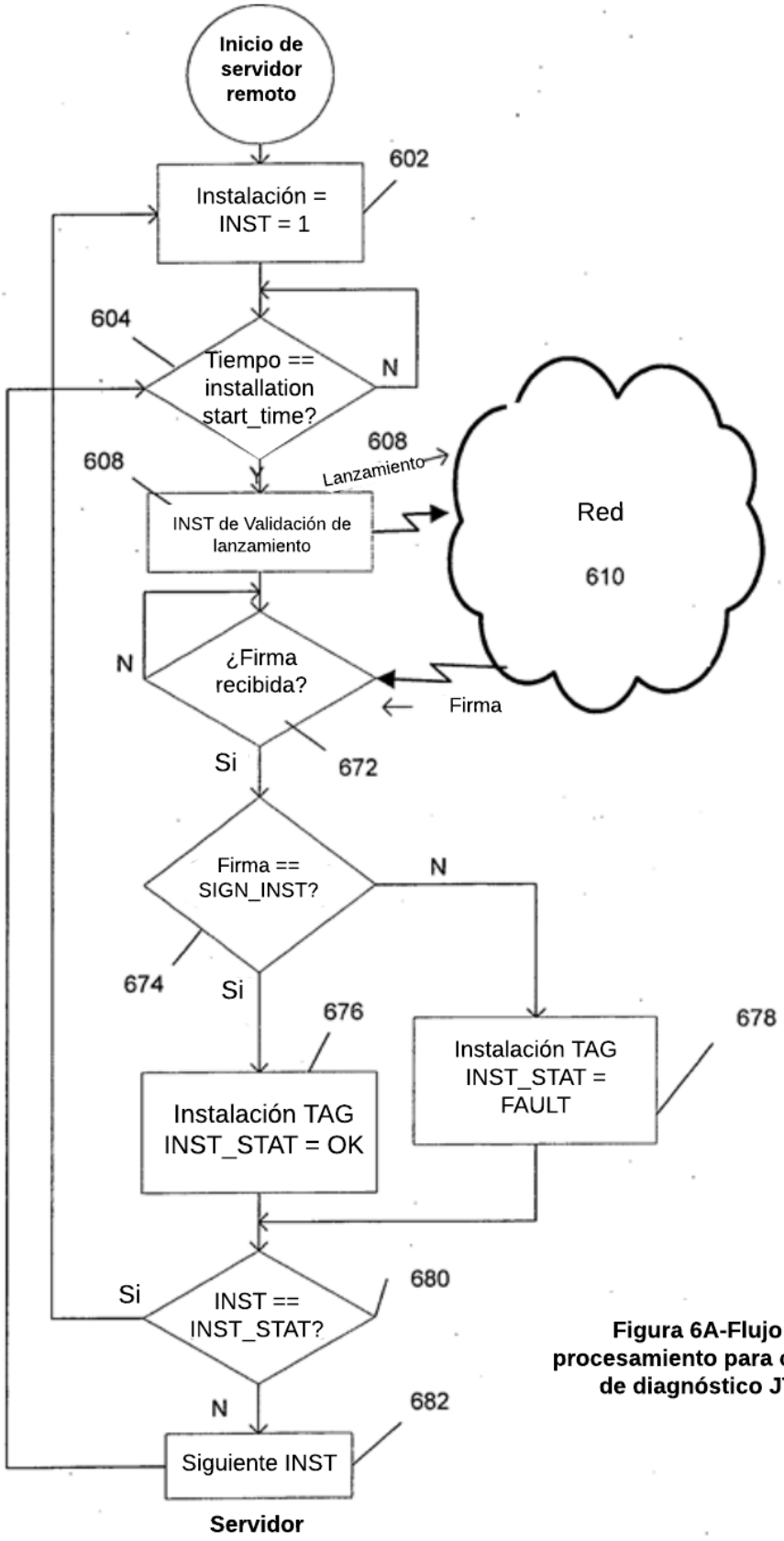


**Figura 6A-Flujo de procesamiento para circuitos de diagnóstico JTAG**

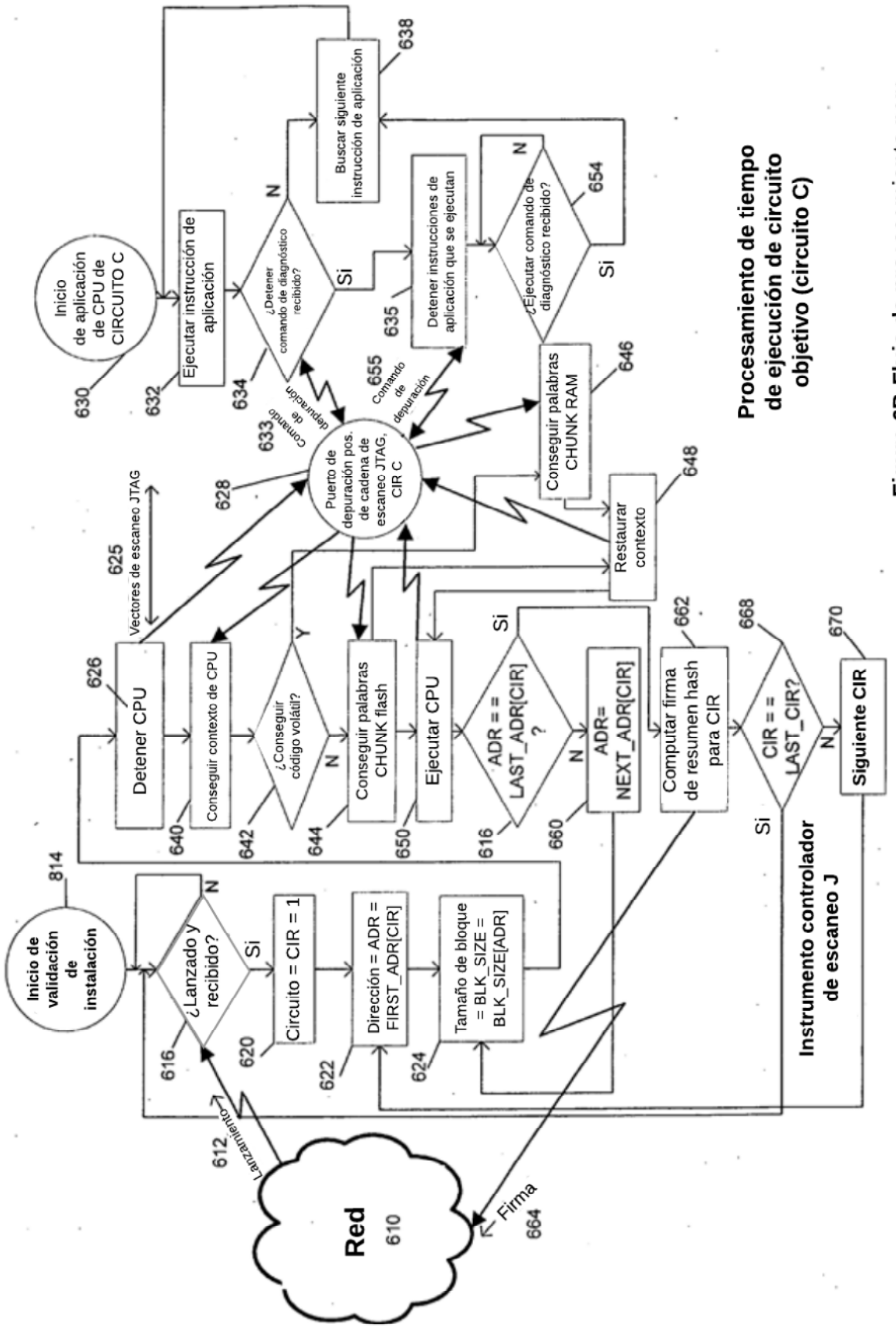


**Figura 6B-Flujo de procesamiento para circuitos de diagnóstico JTAG**

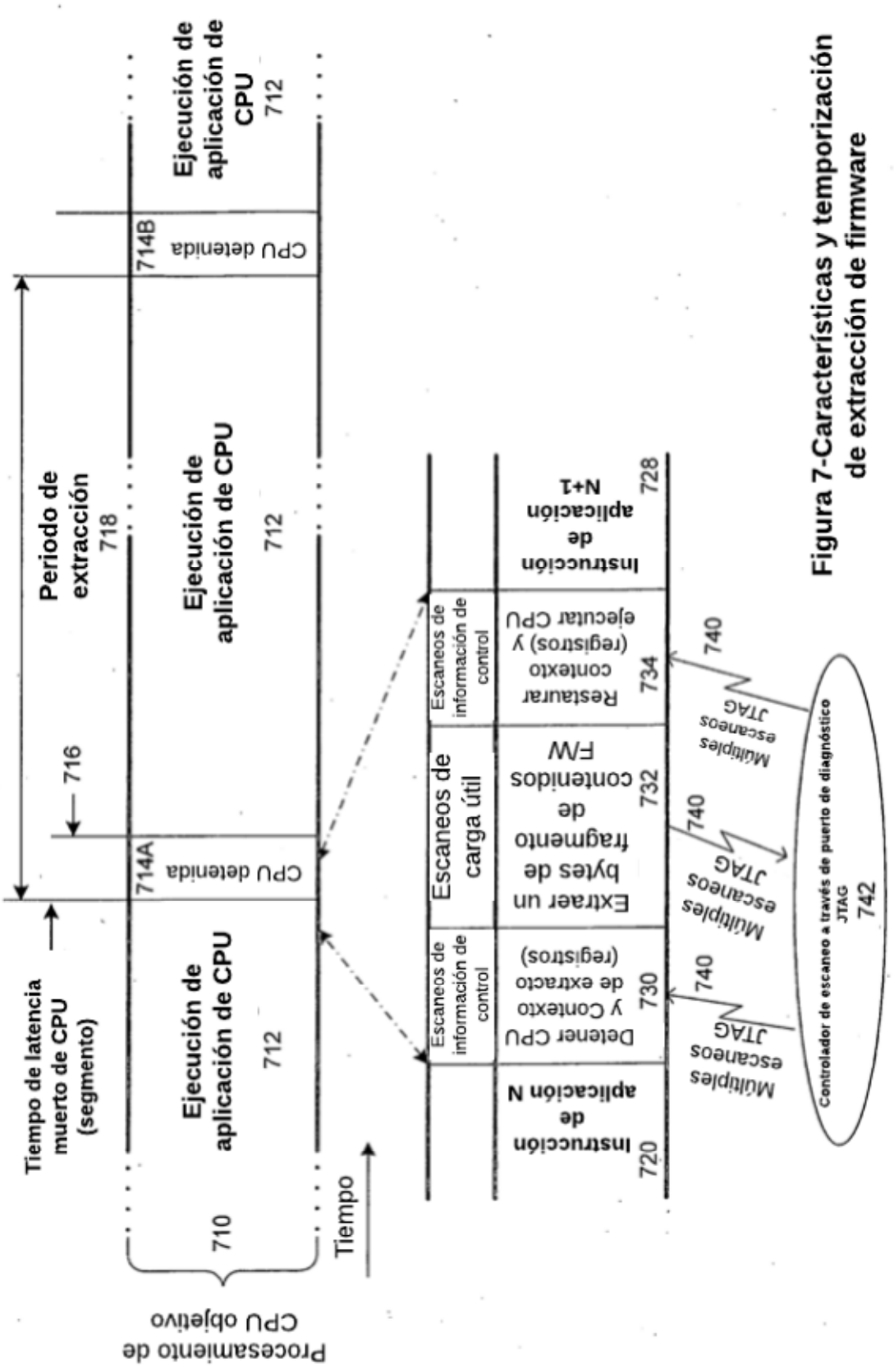


Figura 7-Características y temporización de extracción de firmware