(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 647 941**

(21) Número de solicitud: 201600465

(51) Int. CI.:

G06F 21/50 (2013.01)

(12) PATENTE DE INVENCIÓN B1

(22) Fecha de presentación:

24.05.2016

(43) Fecha de publicación de la solicitud:

27.12.2017

Fecha de concesión:

28.08.2018

(45) Fecha de publicación de la concesión:

04.09.2018

(56) Se remite a la solicitud internacional:

PCT/ES2017/070336

(73) Titular/es:

UNIVERSIDAD DE HUELVA (100.0%)
Dr. Cantero Cuadrado nº 6
21071 Huelva (Huelva) ES

(72) Inventor/es:

JIMENEZ NAHARRO , Raúl ;
GOMEZ BRAVO , Fernando ;
GOMEZ GALAN , Juan Antonio ;
SÁNCHEZ RAYA , Manuel y
MEDINA GARCIA, Jonathan

(54) Título: **Sistema inhibidor de ataques hardware en un bus I2C, módulo esclavo y red que lo comprende**

(57) Resumen:

Un sistema inhibidor de ataques hardware para un módulo esclavo (20) en un buses I2C que incluye detector (42) para detectar el inicio de comunicación en la línea de datos SDA y para generar una señal de inicialización; un oscilador (44) para generar una señal de reloj independiente; un medidor (46) para medir automáticamente la frecuencia de la señal de sincronización SCL y para compararla con la señal de reloj del oscilador (44) y para generar una señal indicando la existencia de ataque; un dispositivo de respuesta (48) configurado para recibir la señal indicando la existencia de ataque y para regenerar la línea SCL a partir de la señal del dispositiva oscilador (44).

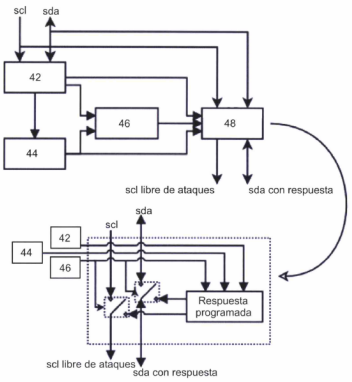

Fig. 4

ES 2 647 941 B1

DESCRIPCIÓN

Sistema inhibidor de ataques hardware en un bus I2C, módulo esclavo y red que lo comprende.

Campo técnico de la invención

La invención pertenece al campo de la robótica industrial. En particular, se relaciona con la seguridad frente a ataques a la comunicación entre dispositivos bajo el protocolo de I2C.

Antecedentes de la invención o Estado de la Técnica

Los motores o actuadores están presentes en cualquier entorno robótico, ya que van a ser los encargados del movimiento de las diferentes partes del robot (en caso de manipuladores) o del robot mismo (en caso de robots móviles). En el caso de manipuladores, los motores serán los encargados de realizar el movimiento del brazo, así como de muñeca y efector final (o parte con la que interactúa el manipulador). En cambio, en el caso de los robots móviles, la principal función de los motores consiste en dotar de movimiento a los mismos, generalmente a través de ruedas. Cualquier fallo en el proceso de comunicación con los motores provocará un error en la trayectoria del robot, y de ahí la importancia de asegurar la fiabilidad en la comunicación con los motores.

Uno de los mecanismos más usuales para la comunicación con los motores es el protocolo de comunicaciones I2C (Inter-Integrated Circuit). Este protocolo, diseñado por Philips, data en sus inicios del año 1992 en su versión 1.0. Este protocolo nació con la intención de comunicar microcontroladores y sus periféricos de una forma compacta en un mismo bus. Este bus I2C tiene las siguientes características:

- Es un bus bidireccional, por lo tanto el proceso de comunicación permite operaciones de lectura y de escritura.

- Es un bus serie, por lo tanto, la información es transmitida bit a bit.

- El bus va a conectar un elemento (generalmente un microcontrolador) que hará las funciones de maestro, y uno o varios elementos adicionales (generalmente periféricos) que harán las funciones de esclavo.

- El protocolo utilizará tres líneas diferentes la línea SCL, que será la línea encargada de la sincronización o línea de reloj (controlada por el maestro); la línea de dato SDA, que será la línea por la que se llevará a cabo el proceso de comunicación (controlada por el maestro o el esclavo según la operación realizada): la línea de tierra, para tener una referencia común.

- Los valores lógicos en la transmisión son el nivel de tierra para el '0' lógico, y alta impedancia para el '1' lógico. La alta impedancia evita la necesidad de que todos los componentes tengan el mismo nivel de polarización.

- Las líneas del bus I2C son accesibles para permitir la conexión de otros periféricos.

El proceso de comunicación requiere de una condición de espera (identificada porque las líneas SCL y SDA están ambas en alta impedancia). El proceso de comunicación comienza con una condición de inicio, identificado por una bajada en la línea SDA mientras que la línea SCL permanece en alta impedancia. A partir de ese momento, la

línea SCL varía comportándose como una señal de reloj, y la línea SDA únicamente puede cambiar mientras que la línea SCL permanece a nivel bajo, ya que mientras está a nivel alto, la línea SDA debe permanecer estable para ser monitorizada.

El proceso de comunicación de una operación de escritura es como sigue. En primer lugar, el maestro envía la dirección del esclavo con el que se quiere comunicar, y el bit de control se mantiene a '0' para identificar una escritura. El esclavo cuya dirección ha sido transmitida responderá en el siguiente ciclo con un reconocimiento. Seguidamente, el maestro envía la dirección del registro del esclavo donde quiere escribir que será contestado por el esclavo con un reconocimiento. Finalmente, el maestro envía el valor que quiere escribir, que será contestado por el esclavo con un reconocimiento. Como acto final, el maestro genera una condición de parada. Esta condición consiste en que la línea SDA pase de nivel bajo a alta impedancia mientras que la línea SCL permanece en alta impedancia. Por lo tanto, la condición de parada implica dejar a las líneas en la condición de espera.

Adicionalmente, algunas implementaciones del protocolo I2C disponen de la característica denominada "clock stretching", según la cual el periférico puede detener el proceso de comunicación durante un tiempo limitado. Mientras que el periférico detiene la comunicación (usualmente debido a que necesita más tiempo para llevar a cabo su operación), el maestro entra en una situación de espera sin modificar los valores de las dos líneas SCL y SDA del bus I2C. Cuando el periférico habilita de nuevo el proceso de comunicación, el maestro continúa por el punto en el que la comunicación se había detenido.

Este protocolo es muy vulnerable frente ataques hardware por dos motivos fundamentales. En primer lugar, las líneas del protocolo tienen una elevada accesibilidad, no solo para los periféricos conectados a ellas, sino también para cualquier otro elemento, entre los que se puede encontrar uno que perpetre un ataque at proceso de comunicación. En segundo lugar, el hecho de utilizar la alta impedancia como '1' lógico permite que el valor '0' prevalezca sobre el valor '1'. Dicha situación puede tener la consecuencia de que cuando dos elementos quieran transmitir por el bus (ya sea en la línea SCL, como bus de reloj, o en la línea SDA, como bus de datos) dos valores diferentes de forma simultanea (conocida comúnmente como colisión de información), el valor predominante sea el valor '0'. Por lo tanto, el resto de elementos conectados a dicho bus identificarían que su valor es '0', sin darse cuenta de la situación de colisión de información.

Existen muchas formas de ataques hardware sin incluir la utilización de virus informáticos. Estos ataques no necesitan grandes recursos, ni tienen porqué resultar en implementaciones complicadas. Una categoría de estos ataques es la denominada ataques por inserción de fallos.

Ataques típicos por inserción de fallos son ataques que varían la frecuencia de la señal de reloj. En el caso de disminuir la frecuencia de operación se consigue aumentar la vulnerabilidad frente a ataques por ingeniería inversa. Mientras que el aumento de la frecuencia de operación por encima de la máxima permitida provoca fallos, ya que el sistema no dispone del tiempo suficiente para finalizar su operación. Este tipo de ataques puede ser viable en cualquier sistema electrónico, como pueden ser los entornos robóticos.

No obstante hasta ahora las aplicaciones informáticas basadas en microprocesadores y/o microcontroladores han sido donde se ha centrado la consideración de los ataques hardware; mientras que, hasta la fecha, este tipo de ataques no han sido considerados en

las aplicaciones reboticas. En el campo informatice (en el cual se centra la búsqueda de defensas), el fin principal de los ataques por inserción de fallos ha sido la obtención de información privilegiada. Por lo tanto, los objetivos de estos ataques suelen ser sistemas de seguridad, tales como tarjetas inteligentes (smart cards) o mecanismos de encriptación y/o desencriptación. El objetivo de estos ataques suele ser que el sistema de seguridad tenga un fallo creando una vulnerabilidad, la cual es utilizada para obtener la información privilegiada.

Centrándose en la aplicación con un microprocesador, el principal objetivo de un ataque por inserción de fallo en la línea de reloj consiste en que el programa ejecute una determinada porción de código (por ejemplo relativa a un acceso con privilegios) que no estaba contemplada en la ejecución normal del programa. El fallo debido a la variación de la frecuencia de la señal de reloj (en este caso aumento) impide que la instrucción encargada de evitar la entrada en dicha porción de código no sea ejecutada. La defensa tradicional frente a este tipo de ataques consiste en introducir un sistema que anule la señal de reloj cuando se produce una variación en la frecuencia de la señal de reloj no permitida. De esta forma, el sistema microprocesador permanece esperando hasta que llegue un ciclo de frecuencia válida para ejecutar las instrucciones, evitando la no ejecución de ninguna instrucción. Tradicionalmente, se ha utilizado para tal fin un sensor de frecuencia basado en filtros paso banda, cuya misión es identificar zonas en las que la señal tiene una frecuencia diferente a la permitida. Las características comunes de estos sensores son las siguientes·

• La detección es realizada durante todo el tiempo de ejecución del sistema.

• La respuesta del sistema es la anulación de la señal de reloj (evitando la ejecución del fallo) durante el ataque.

Por el contrario, el efecto que causa la ejecución de estos ataques sobre una aplicación rebotica no ha sido considerado, ni por lo tanto sus defensas. Para ello, se considerará una plataforma rebotica donde el ataque será realizado sobre la comunicación entre el maestro y un esclavo en particular, conectados mediante el protocolo I2C.

Un ataque por inserción de fallos a un protocolo I2C sería tan simple como colocar un '0' lógico en la línea SCL durante un periodo determinado del proceso de comunicación. A pesar de que aparentemente se pueda interpretar que es una situación de "clock stretching", la diferencia radica en que el atacante sólo liberará la comunicación cuando esta haya finalizado; mientras que la liberación del "clock stretching" real no implica la finalización de la transmisión. Dicha acción supondría que un proceso de comunicación no fuese completado, y por lo tanto que no fuera ejecutada la orden inmersa en el mensaje. Además, la implementación de un ataque selectivo (atacar la comunicación a un esclavo determinado) es muy simple ya que el sistema atacante puede leer muy fácilmente la dirección del esclavo con el que se desea establecer la comunicación. En el caso de que se desee evitar la comunicación con dicho esclavo, después de leer la dirección, se forzará la línea SCL a '0', y por lo tanto, la comunicación no será completada y, en consecuencia, la orden no será ejecutada. Además, como ningún esclavo detecta que se desea establecer la comunicación con él, ninguno advierte que el sistema está siendo atacado.

Según lo anterior, el ataque considerado puede ser visto como un ataque por inserción de fallos variando la frecuencia de la señal de reloj, logrando que la frecuencia valga cero durante el resto del proceso de comunicación.

Teniendo en cuenta que en el campo de aplicación de la rebotica, no se ha considerado esta situación, tampoco se conoce ninguna solución frente al ataque. Se podría plantear la defensa tradicional aplicada en el campo informatice, es decir, el sensor de frecuencia basado en filtros paso-banda. No obstante, no sería de utilidad por las siguientes razones

• La detección no debe ser realizada durante todo el tiempo de ejecución sino únicamente mientras dure un proceso de comunicación. En el caso de que no se esté produciendo un proceso de comunicación, el sensor debe permanecer inactivo.

• La respuesta del sensor no puede ser la anulación de la señal de reloj, debido a que es la misma acción que la producida por el ataque.

Breve descripción de la invención

A la vista de las limitaciones identificadas en el estado de la técnica, sería deseable la inhibición de ataques hardware que evite problemas en la comunicación y la perdida de órdenes.

Es objeto de la presente un sistema para inhibir ataques hardware en un buses I2C que incluye un dispositivo detector que detecta el inicio de comunicación en la línea de datos SOA del bus I2C y que genera, tras la detección una señal de inicialización. El sistema incluye también un dispositivo oscilador que recibe la señal de inicialización del detector y que genera una señal de reloj independiente. El sistema incluye también un dispositivo medidor que recibe la señal de inicialización del detector, mide automáticamente la frecuencia de la señal de sincronización SCL y compara con la señal de reloj del oscilador para generar una señal indicando la existencia de ataque en función del resultado de la comparación. El sistema incluye además un dispositivo de respuesta que recibe la señal indicando la existencia de ataque y regenera la línea SCL a partir de la señal del dispositivo oscilador.

Opcionalmente, el dispositivo de respuesta puede detener la línea SOA mientras se detecta un ataque.

Opcionalmente, el dispositivo medidor incluye un contador en anillo.

Es otro objeto de la presente invención, un módulo esclavo para un bus I2C que comprende integrado el sistema inhibidor anterior.

Es otro objeto adicional de la presente invención, una red que incluye un bus I2C al que se conectan al menos un módulo esclavo acoplado con el dispositivo inhibidor y también un módulo maestro.

Ante la detección de dos situaciones diferentes, la presente propuesta es capaz de actuar selectivamente. Esto es:

• cuando el proceso de comunicación ha sido completado con éxito, en cuyo caso no interviene en el proceso;

• cuando no ha sido completado con éxito, debido a un ataque hardware, en cuyo caso, la respuesta será acorde a la aplicación específica (preferiblemente mediante un módulo programable).

Para defender el proceso de comunicación a través de un bus I2C en el ámbito de los dispositivos esclavos, la invención tendrá dos misiones fundamentales: en primer lugar,

debe detectar la presencia de un ataque en el proceso de comunicación monitorizando en tiempo real la frecuencia de la señal de sincronización (SCL); en segundo lugar, y en el caso de que se haya detectado un ataque, la detección llevará aparejada una respuesta (que se considere adecuada) conocida por el sistema. Por lo tanto, la presente invención presenta una solución a ataques por inserción de fallos en entornos robóticos.

A lo largo de la descripción y las reivindicaciones, la palabra "comprende" y sus variantes no pretende excluir otras características técnicas, aditivos, componentes o pasos. Los siguientes ejemplos y figuras se proporcionan a modo de ilustración, sin ser limitativos de la presente invención.

Breve descripción de las figuras

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

La FIG. 1 muestra la situación de ataque y defensa tradicional en un sistema basado en microprocesador

La FIG. 2 muestra las formas de onda relativas a una operación de escritura en un bus I2C. En la FIG. 2A se aprecia la operación normal sin ataque y en la FIG. 2B cuando ha sufrido un ataque.

FIG. 3 muestra el esquema de conexión de los diferentes módulos en un proceso de comunicación con posibilidad de ataque.

FIG. 4 Muestra el esquema en más detalle del sistema de inhibición de ataques de comunicación.

Descripción detallada de la invención

Con referencia a las figuras anteriores, se describe una realización en la que varios dispositivos esclavos están conectados a un bus I2C controlado por un dispositivo maestro.

La FIG. 1 es un ejemplo que muestra la situación de ataque y defensa tradicional en un sistema basado en microprocesador. En el caso de ataque, la frecuencia de la señal de reloj aumenta para que la instrucción N no sea ejecutada y pase a la instrucción N+1: aunque por la operación del programa no se debiera ejecutar dicha instrucción. Si hay defensa contra este tipo de ataque, el sensor de frecuencia anulará la señal de reloj cuando su frecuencia esté fuera del rango permitido manteniéndose la instrucción N a la espera de la llegada de un nuevo ciclo de la señal de reloj para ser ejecutada.

En la FIG. 2 se puede ver un ejemplo de operación de escritura sin ataque y con ataque en un bus I2C. En la FIG. 2A se muestran las formas de onda relativas a una operación de escritura normal sin ataque en la comunicación. De forma análoga, en la FIG. 2B, se ilustra una escritura que ha sufrido un ataque cuyo efecto se aprecia en las zonas señaladas. El ataque fuerza la línea SCL a nivel bajo cuando se ha identificado al esclavo que se desea atacar en el proceso de comunicación.

Debido a que el esclavo atacado no va a contestar mediante los reconocimientos, el módulo de ataque también realizará esta tarea con el fin de que el módulo maestro crea que la comunicación sigue su curso adecuado.

El mecanismo de defensa se ha ideado ventajosamente de tal forma que pueda dar solución a dos escenarios diferentes. Primero: su implementación en el mismo sustrato que el dispositivo esclavo, logrando de esta forma un dispositivo esclavo que se comunica mediante el bus I2C libre de ataques de comunicación. Segundo: su uso con dispositivos esclavos estándares de tal forma que se le añade funcionalidad sin necesidad de su modificación.

En la FIG. 3 se ilustra el esquema de conexión de los principales módulos que pueden intervenir en un proceso de comunicación. Los módulos incluidos son los siguientes:

• El módulo maestro **10** encargado de gestionar el proceso de comunicación, y de esta forma, encargado de controlar la línea de sincronización SCL.

• Un módulo esclavo defendido **20a**, es decir, acoplado al sistema inhibidor **40**.

• Un módulo esclavo sin defensa **20**, y por lo tanto, susceptible a los ataques de comunicación.

• Un módulo de ataque **30** (que añadiría el atacante) que se encargaría de gestionar dicho ataque. Dicho módulo de ataque **30** tendrá la misma filosofía de diseño que un módulo esclavo **20** convencional exceptuando que estará dispuesto a alterar los valores de la señal de sincronización SCL.

Como se puede apreciar de la FIG. 3 la inserción del módulo de ataque **30** no implica la modificación de ningún otro modulo. Basta con tener accesible las líneas del bus I2C, situación normal por la propia filosofía del protocolo.

Continuando con la FIG. 3, el módulo maestro **10** se comunica con dos esclavos: uno sin protección **20** y otro protegido **20a** mediante el sistema inhibidor **40** objeto de esta invención. También se representa un módulo atacante **30** La arquitectura del módulo esclavo defendido **20a** seria idéntica a la del módulo esclavo tradicional **20** sin defensa, con lo cual ni el protocolo de comunicación ni el hardware tradicionalmente utilizado han de sufrir modificaciones. En cambio, la utilización del sistema inhibidor **40** en cascada con el módulo esclavo **20a**, permitirá que este quede inmune a los perjuicios provocados por un ataque. En otra realización es posible integrar el sistema inhibidor **40** en el módulo esclavo, obteniendo un módulo esclavo libre de ataques **20b**.

Este sistema inhibidor **40** estará formado por cuatro elementos como se puede ver en la FIG. 4 Se explican cada uno a continuación:

• Un primer dispositivo detector **42** que detecta las condiciones de inicio y parada (para saber cuándo el sensor tiene que operar) y de inicio de ciclos (para saber cuándo tiene que iniciar una nueva medida). Este detector **42** generará las señales de inicialización para el resto de dispositivos del sistema inhibidor **40**.

• Un segundo dispositivo oscilador **44** estará dedicado a la generación de un reloj interno que controlará el funcionamiento del proceso de monitorización e inhibición, es decir, de los demás dispositivos exceptuando el detector **42**. Este reloj debe ser generado en el mismo sustrato para evitar vulnerabilidades. Esta generación interna implicará el uso de técnicas de oscilación en anillo (para disponer de una señal oscilatoria) y técnicas de división de frecuencias (para evitar un uso excesivo de hardware).

• Un tercer dispositivo medidor **46** estará destinado a medir automáticamente la frecuencia de la señal SCL tomando como base temporal la señal de reloj generada en el

oscilador **44** y la señal de inicialización generada por el detector **42**. Esta medida de frecuencia estará disponible inmediatamente en el siguiente pulso a medir o cuando se comprueba que tiene una frecuencia menor que la permitida. El medidor **46** generará una señal Indicando la existencia de un ataque según el valor de la frecuencia medida.

• Un cuarto dispositivo de respuesta **48** estará destinado a generar la respuesta del sistema inhibidor **40**. En caso de que no exista ningún ataque, la respuesta debe ser el paso de las señales SCL y SDA sin ninguna interferencia, por lo que la presencia del sistema inhibidor **40** sería transparente. En caso de que haya ataque, la respuesta implementada debe ser la que mejor se adapte a la aplicación y al uso determinado. Habitualmente, no se dejarán pasar las líneas SCL y SDA en caso de ataque, y este dispositivo de respuesta **40** autónomamente generará una respuesta que se codificará en base a un conjunto de mensajes transmitidos a través del bus I2C y que harán al esclavo **20a** inmune al ataque detectado. Hasta que no haya finalizado dicha respuesta, no se aceptará ninguna comunicación adicional (este siendo atacada o no). Por tanto, en caso de ataque, el módulo de respuesta **48** actuará como si fuera un módulo maestro **10**.

A modo de ejemplo, la respuesta considerada por defecto puede ser la parada del motor o actuador. Con este fin, este dispositivo de respuesta **48** será preferiblemente un elemento programable que implementará el envío, a través del bus I2C que lo conecta al esclavo **20a**, de los correspondientes mensajes de parada. Esto, dependiendo de la aplicación (por ejemplo un controlador de motores), puede que ser efectuado mediante un solo mensaje o mediante un conjunto de mensajes que garantice la ejecución de un perfil de velocidades que detenga de forma segura el robot. No obstante, la secuencia de mensajes será programable atendiendo a la naturaleza de la aplicación. La señal de SCL de esta transmisión empleará el oscilador **44** que es local y que garantizará una comunicación libre de ataques.

Según lo anterior, puede plantearse una primera realización en la que el sistema inhibidor **40** y el módulo esclavo **20** a proteger están incluidos en un mismo sustrato integrado obteniendo un módulo esclavo libre de ataques **20b** que ya incorpora las funciones del sistema inhibidor **40**.

Alternativamente, puede concebirse una segunda realización en la que se acoplará la el sistema inhibidor **40** a un módulo esclavo estándar **20a** en el sentido de que no tiene que gestionar la posibilidad de un ataque pero que quedará protegido entonces frente a ataques.

Implementación en un prototipo

El sistema inhibidor 40 se puede implementar mediante técnicas digitales, y por lo tanto, puede ser implementado como hardware en un circuito de aplicación específica (ASIC) o en un dispositivo programable (como FPGAs o CPLDs). Estas opciones facilitan su integración en el mismo sustrato que el módulo esclavo **20** disponiendo de esta forma de un solo bloque integrado **20b**. Así mismo puede ser utilizado de forma externa al elemento esclavo ya que no requiere de ninguna señal específica para funcionar.

En los primeros prototipos se ha empleado una opción adaptada, es decir, el sistema inhibidor **40** en un sustrato diferente al del elemento esclavo **20**. La implementación del inhibidor ha sido realizada en un dispositivo FPGA de la familia Spartan-3AN (más concretamente el modelo XC3S700AN incluido en una placa de desarrollo basada en FPGA). Mientras que el elemento esclavo utilizado ha sido el controlador de motores MD23. Debido a la adaptación, la respuesta del inhibidor ha implicado la regeneración del bus I2C, de tal forma que la respuesta pactada a una situación de ataque ha sido la

parada inmediata de los motores. No obstante, la arquitectura propuesta permite programar otro tipo de respuesta que se adaptase mejor a la aplicación específica del sistema robótico.

Con el fin de completar el entorno del proceso de comunicación, se ha implementado en la misma FPGA que el sistema inhibidor **40**, un módulo maestro **10** del bus I2C, un módulo esclavo **20a** sin defensa y un módulo de ataque **30**.

Con respecto a la implementación interna del inhibidor, se van a detallar las actuaciones seguidas en cada uno de los elementos. En el caso del dispositivo detector **42**, se ha utilizado una solución asíncrona insensible a retrasos, ya que este elemento no dispone de ninguna señal de reloj para poder operar. La secuencia de inicialización es la siguiente. En primer lugar, se para el dispositivo oscilador **44** para que la señal de reloj no afecte a las condiciones de inicialización del resto de elementos. En segundo lugar, se inicializa el dispositivo de respuesta **48** ya que necesita los datos previos del dispositivo medidor **46** en el caso de ataque (ya que tiene que regenerar la señal de sincronización). En tercer lugar, se inicializa el dispositivo medidor **46** para comenzar la medición en cero. Finalmente, se vuelve a poner en marcha el dispositivo oscilador **44** para comenzar la monitorización. En los primeros prototipos, este ciclo de inicialización tarda aproximadamente unos 15 ns.

El dispositivo oscilador **44** ha sido implementado de forma mixta. En primer lugar, se genera una frecuencia de oscilación fina utilizando un oscilador en anillo. En segundo lugar, se utiliza la señal fina como entrada a un divisor de frecuencia para generar la frecuencia final de oscilación. El divisor anterior es utilizado para reducir un excesivo uso de elementos hardware debido a generar frecuencias bajas únicamente con osciladores en anillo. En nuestro caso particular, se ha generado una frecuencia final de unos 320 ns, partiendo de una frecuencia fina inicial de unos 10 ns.

El dispositivo medidor **46** deberá realizar la medición de la forma más rápida posible que permita la utilización de dicho sistema inhibidor **40** al mayor número de situaciones posible, aunque en el caso del prototipo no fuese necesario. La cuenta de la medición ha sido implementada con la utilización de un contador en anillo. Esta solución tiene dos ventajas. En primer lugar, es la solución más rápida a la operación de cuenta. En segundo lugar, la comparación de las medidas requiere una implementación tan simple como una puerta AND de dos entradas. El tamaño de este contador en anillo debe ser tal que abarque todo el rango de la frecuencia permitida. En este prototipo se ha utilizado un límite máximo de 10 ciclos, correspondiendo a un periodo de unos 3.2 us.

El dispositivo de respuesta **48** deberá generar la respuesta del inhibidor **40**. En el caso de ataque, y como ya se ha comentado previamente, debe regenerar las líneas SCL y SDA, para que la respuesta sea la programada. La línea SDA es generada desde cero con los valores adecuados y siguiendo la sincronización del maestro **10** (para los casos en los que no había ataque). En el caso de la línea SCL, hay que regenerarla con la misma sincronización. Para ello, se utiliza la medida obtenida en el dispositivo medidor **46,** y en base a esta medida se regenerará la señal siguiendo la filosofía inversa, es decir basándose en un oscilador **44** con un contador en anillo para determinar cuando la señal SCL regenerada debe valer '0' ó '1'. Una vez que dicha señal ha sido regenerada hay que implementar el flujo normal de la comunicación I2C como si se tratara de un elemento maestro **10**.

REIVINDICACIONES

1. Sistema inhibidor de ataques hardware en un buses I2C **caracterizado** por que comprende:

- un dispositivo detector (42) configurado para detectar el inicio de comunicación en la línea de datos SDA del bus I2C y para generar, si se ha detectado, una señal de inicialización;

- un dispositivo oscilador (44) configurado para recibir la señal de inicialización del detector (42) y para generar una señal de reloj independiente;

- un dispositivo medidor (46) configurado para recibir la señal de inicialización del detector (42) y para medir automáticamente la frecuencia de la señal de sincronización SCL y para compararla con la señal de reloj del oscilador (44) y para generar una señal indicando la existencia de ataque en función del resultado de la comparación;

- un dispositivo de respuesta (48) configurado para recibir la señal indicando la existencia de ataque y para regenerar la línea SCL a partir de la señal del dispositivo oscilador (44).

2. Sistema inhibidor según la reivindicación 1, **caracterizado** por que el dispositivo de respuesta (48) está configurado para detener la línea SDA mientras se detecta un ataque.

3. Sistema inhibidor según la reivindicación 1 o 2, **caracterizado** por que el dispositivo medidor (46) comprende un contador en anillo.

4. Módulo esclavo (20b) para un bus I2C que comprende integrado un sistema inhibidor (40) según una cualquiera de las reivindicaciones 1 a 3.

5. Red que comprende un bus I2C al que se conectan al menos:

- un módulo esclavo (20a) acoplado con un dispositivo inhibidor (40) según una cualquiera de las reivindicaciones 1 a 3,

- un módulo maestro (10).

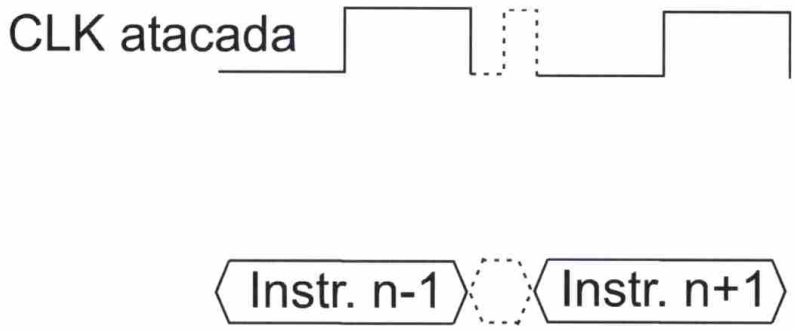
CLK atacada
Instr. n-1
Instr. n+1

Fig. 1A

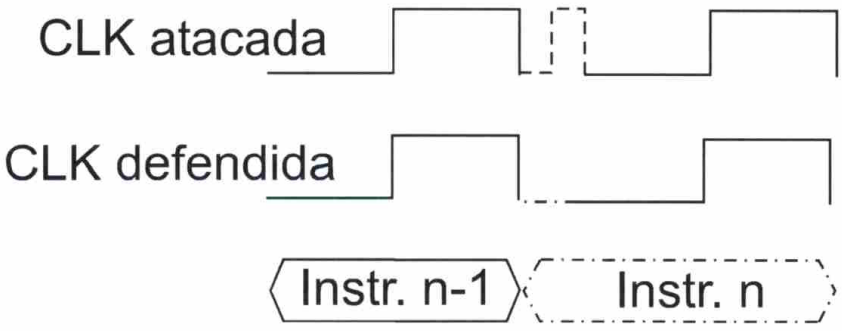
CLK atacada
CLK defendida
Instr. n-1
Instr. n

Fig. 1B

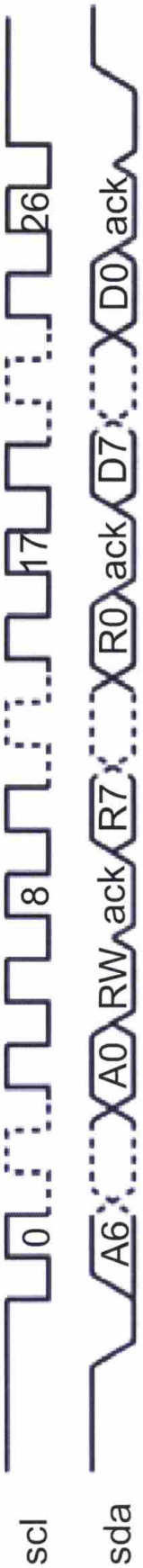
scl
sda
0
A6
A0
RW
ack
R7
R0
ack
D7
D0
ack
8
17
26

Fig. 2A

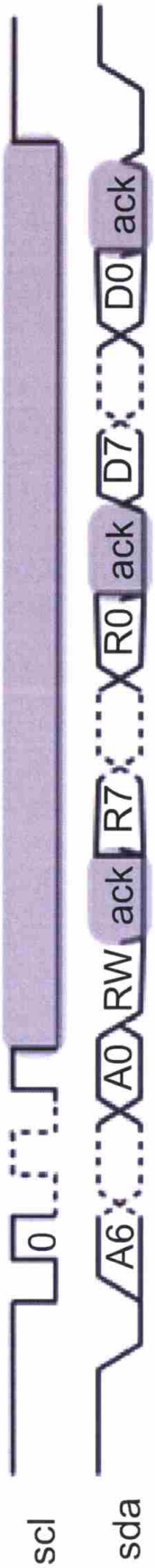
scl
sda
0
A6
A0
RW
ack
R7
R0
ack
D7
D0
ack

Fig. 2B

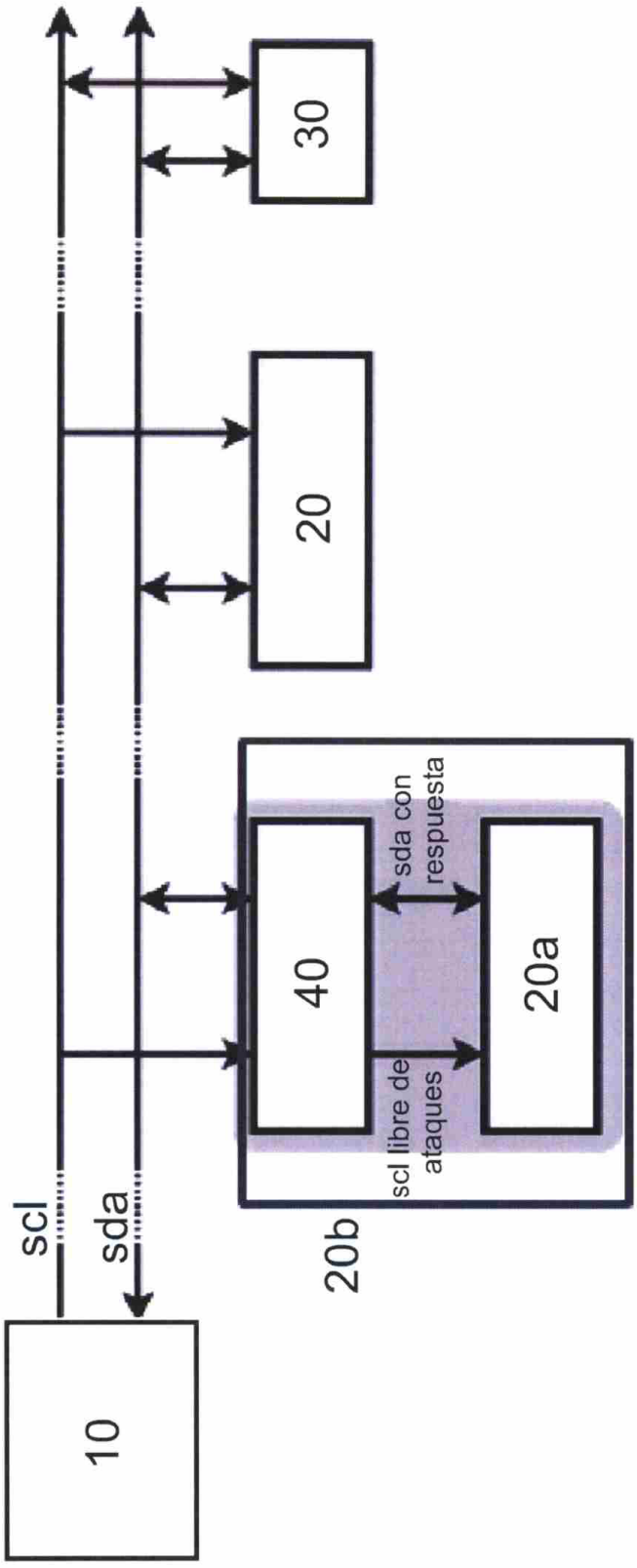
30
20
40
sda con respuesta
20a
scl libre de ataques
20b
scl
sda
10

Fig. 3

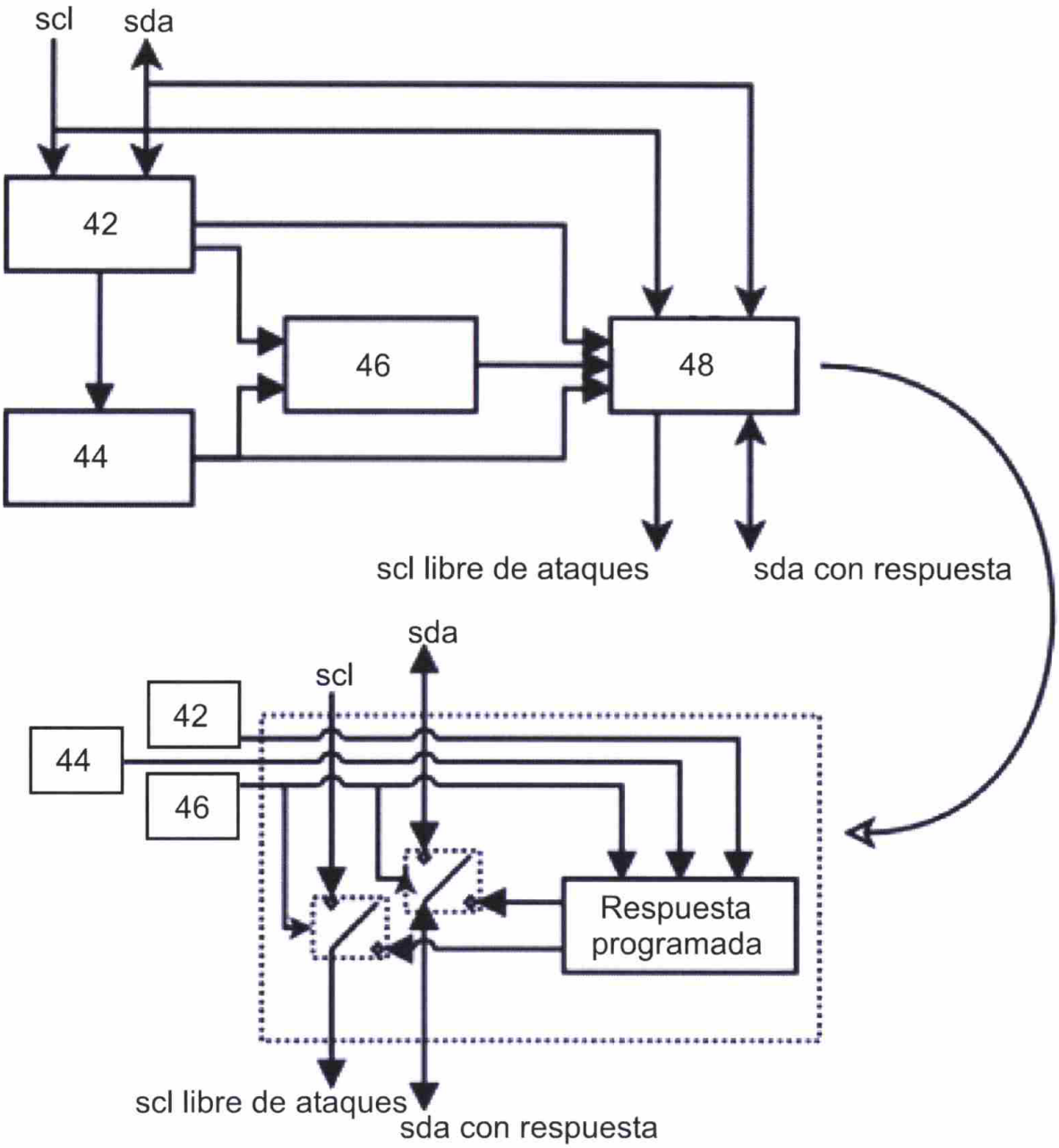
scl
sda
42
46
48
44
scl libre de ataques
sda con respuesta
sda
scl
42
44
46
Respuesta programada
scl libre de ataques
sda con respuesta

Fig. 4