(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 830 433**

(51) Int. CI.:

G06F 21/12 (2013.01)
G06F 21/57 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **12.10.2016 PCT/EP2016/074391**

(87) Fecha y número de publicación internacional: **08.06.2017 WO17092917**

(96) Fecha de presentación y número de la solicitud europea: **12.10.2016 E 16782036 (4)**

(97) Fecha y número de publicación de la concesión europea: **26.08.2020 EP 3347848**

(54) Título: **Módulo programable en memoria y procedimiento para la transmisión protegida de datos a un módulo programable en memoria**

(30) Prioridad:

04.12.2015 DE 102015224300

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
03.06.2021

(73) Titular/es:

SIEMENS MOBILITY GMBH (100.0%)
Otto-Hahn-Ring 6
81739 München, DE

(72) Inventor/es:

MERLI, DOMINIK y
SCHNEIDER, DANIEL

(74) Agente/Representante:

LOZANO GANDIA, José

DESCRIPCIÓN

Módulo programable en memoria y procedimiento para la transmisión protegida de datos a un módulo programable en memoria

La invención se refiere a un módulo programable en memoria con ancla de confianza que puede cargarse, así como a un procedimiento para la transmisión protegida de datos desde una fuente de datos externa a al menos un componente de un módulo programable en memoria.

Se denominan sistemas embebidos a computadoras u ordenadores que están integrados en un contexto técnico y que por ejemplo ejecutan funciones de vigilancia o de mando en un aparato. Los sistemas embebidos reúnen usualmente microprocesadores, módulos programables en memoria, también denominados FPGA, memorias RAM, memorias flash y diversas periferias en una plataforma. La protección de estos sistemas juega un papel cada vez más importante en la garantía de un funcionamiento correcto. Una de las exigencias más importantes para ello es la integridad del sistema embebido, es decir, la protección frente a una manipulación no autorizada.

Para lograr la integridad de un tal sistema embebido, se conoce la carga segura del sistema, conocida también como “secure boot” (arranque seguro) o “trusted boot” (arranque fiable) o “verified boot” (arranque verificado). Al respecto se activa el proceso de carga de un ancla de confianza, también llamada root-of-trust (raíz de confianza). Esta ancla de confianza es por ejemplo una unidad de hardware que comprueba en cada caso la integridad de la etapa siguiente del Boot-Software que ha de cargarse mediante la comprobación de un valor hash, por ejemplo de un valor hash criptográfico, un Message Authentication Code (código de autentificación de mensaje) o de una firma digital relativa al contenido del código a cargar y a los datos necesarios. Sólo cuando la comprobación tiene éxito se arranca la siguiente etapa. Ésta comprueba a su vez la siguiente etapa, con lo que queda garantizada una llamada “Chain-of-Trust”, cadena de confianza.

Un tal mecanismo de Secure-Boot, que protege la integridad de un software que ha de ejecutarse en una Soft-CPU (unidad central de procesamiento de software) en un FPGA, se describe por ejemplo en un artículo con el título “Introducing the new libero SoC v11.6 Software Toolset” (Introducción al nuevo conjunto de herramientas de software Libero SoC v11.6”, publicado e Internet bajo la URL http://www.microsemi.com/products/fpga-soc/security/secure-boot. Este mecanismo puede utilizarse directamente y genera un buen nivel de protección.

Un punto esencial de esta solución es que el código de arranque inicial y la clave criptográfica eventualmente necesaria, tienen que estar implementados protegidos frente a manipulación. Esto sólo ha sido posible hasta ahora cuando el microprocesador o bien un módulo programable en memoria que ha de ejecutarse sobre el que está realizado el microprocesador a ejecutar, ya ha sido dotado por el fabricante de un ancla de confianza integrada en el hardware, por ejemplo de una memoria de sólo lectura para código de carga, también denominada Boot ROM. Pero éste a menudo no es el caso o bien es preciso comprar una variante costosa.

Por el documento DE 10 2013 213 314 A1 se conoce además un procedimiento para archivar al menos un valor de medida de integridad que puede calcularse en una primera zona de memoria de una memoria de un componente, determinándose el valor de medida de integridad que puede calcularse, de los que al menos hay uno, como valor de comprobación de un módulo de software del componente. Así se consigue un elemento de memoria que simplifica considerablemente la aplicación práctica de la llamada tecnología TPM (Trusted Platform Module Technologie, tecnología de módulo de plataforma de confianza), con la que puede proporcionarse una Light-Weight-TPM (TPM ligera).

El documento US 2015/0113258 A1 da a conocer además un procesador digno de confianza que se arranca con ayuda de un Pre-Bootloader (cargador de prearranque) seguro integrado en el mismo y que comprueba si un Bootloader externo es válido, arrancándose el procesador en caso de validez con ese Bootloader. En el marco del Pre-Bootloader se utiliza entonces una llamada Pre-Boot ROM, que por ejemplo está prevista para memorizar informaciones de autentificación y claves criptográficas.

Por lo tanto el objetivo de la presente invención es lograr un ancla de confianza sencilla y económica en un sistema embebido, en particular interactuando con una Soft-CPU, en un módulo programable en memoria.

El objetivo se logra mediante las medidas descritas en las reivindicaciones independientes. En las reivindicaciones secundarias se muestran ventajosos perfeccionamientos de la invención.

El módulo programable en memoria correspondiente a la invención incluye un sistema de prueba y al menos otro componente, pudiendo generarse el sistema de prueba mediante un flujo de bits de carga sobre el módulo y estando constituido el sistema de prueba para leer e introducir al menos un flujo de bits

de datos para la ejecución en el componente del módulo, de los que al menos hay uno, desde una fuente de datos externa, comprobar el flujo de bits de datos y, si el resultado de la comprobación es positivo, emitir una señal de mando que activa el componente con los datos del flujo de bits de datos.

El sistema de prueba es así un ancla de confianza, que se carga con el flujo de bits de carga en un módulo programable en memoria. El componente, de los que al menos hay uno, es por ejemplo una Soft-CPU, es decir, un microprocesador que está integrado en el diseño FPGA. El componente está entonces desactivado tras la carga del sistema de prueba y sólo se activa tras una comprobación positiva del flujo de bits de datos mediante la unidad de prueba. De esta manera se logra una protección de integridad para software, por ejemplo para un Boot Loader o un sistema operativo memorizados en una fuente de datos externa. Esta protección de integridad sólo puede romperse mediante un costosísimo flujo de bits de carga Reverse Engineering (de ingeniería inversa) del FPGA. Así aumenta claramente la seguridad para un Secure Boot sobre módulos programables en memoria económicos sin apoyo Segure Boot dedicado. Otra ventaja consiste en que el sistema embebido completo sólo tiene que ser adaptado mínimamente, no siendo necesario ningún componente de hardware adicional y el sistema de prueba puede implementarse como módulo sobre el FPGA, ahorrando muchos recursos.

En un perfeccionamiento ventajoso incluye el sistema de prueba una función criptográfica y está diseñado para comprobar los datos del flujo de datos con ayuda de un parámetro de validación y de la función criptográfica.

Así puede comprobarse en el sistema de prueba el flujo de bits de datos previsto para el componente de manera flexible antes de activarlo en el componente, para verificar que coincide con el flujo de bits de datos esperado.

En una forma de realización ventajosa la función criptográfica es una función hash y el parámetro de validación un valor hash del flujo de bits de datos creado con la función hash.

Una función hash es aquí una función poco compleja que se puede implementar con facilidad y ahorrando recursos.

En una forma de realización alternativa la función criptográfica es una función hash criptográfica y el parámetro de validación una clave secreta.

En este caso por ejemplo, la función hash criptográfica genera un código de autentificación de mensajes criptográfico HMAC u otro parámetro criptográfico utilizando la clave secreta mediante el flujo de bits de datos y se compara con un HMAC original memorizado en la fuente de datos y/o contenido en el flujo de bits de datos. Esto tiene la ventaja de que no está implementado ningún valor hash específico en el módulo programable en memoria o directamente en el sistema de prueba y en particular este valor hash específico no tiene que transmitirse al módulo en un flujo de bits de carga. Por tanto, se pueden comprobar así distintos flujos de bits de datos con un solo parámetro de validación.

En una realización ventajosa, la función criptográfica es una función de validación de firma y el parámetro de validación es una clave pública.

Esto tiene la ventaja de que no se tiene que transmitir ninguna clave secreta en el flujo de bits de carga al módulo o al sistema de prueba y por lo tanto tampoco se puede leer ni manipular.

En una forma de realización ventajosa, el parámetro de validación se puede implementar con el flujo de bits de carga en el módulo.

Esto permite introducir de forma flexible el parámetro de validación en la síntesis del módulo programable en memoria. Es decir, ya durante la configuración del módulo se crea el parámetro de validación para el software que se cargará a continuación mediante el flujo de bits de carga. Por lo tanto, al desactivar automáticamente el componente después de la configuración inicial y de una activación explícita del componente después de una prueba con éxito, es prácticamente imposible una carga o arranque no comprobada del componente.

En una forma de realización ventajosa, el módulo programable en memoria incluye una unidad de seguridad adicional, que está conectada con el componente a programar y el sistema de prueba y que puede ser activada mediante la señal de mando del sistema de prueba.

Con ello pueden detectarse datos manipulados, como por ejemplo software malicioso, y se puede evitar una manipulación del ancla de confianza.

En una forma de realización ventajosa, el sistema de prueba está diseñado para generar una señal de mando para desactivar el componente o para desactivar al menos una función parcial del componente, si

el resultado de la comprobación es negativo o para activar el componente con un segundo flujo de bits de datos, que hace imposible o sólo limitadamente posible una utilización del componente.

Así es posible una desactivación activa de un componente ya activo.

En otra forma de realización ventajosa, el módulo programable en memoria incluye un interruptor controlable, que conecta el componente o el ancla de confianza con un reloj mediante la señal de mando del sistema de prueba y por lo tanto activa el componente o el ancla de confianza o lo desconecta, desactivándolo.

Un interruptor controlable es fácil de configurar y proporciona una posibilidad sencilla de activar y/o desactivar el componente.

El procedimiento correspondiente a la invención para la transmisión protegida de datos desde una fuente de datos externa a al menos un componente de un módulo programable en memoria, incluye las etapas de generación de un sistema de prueba en el módulo mediante un flujo de bits de carga, lectura de al menos un flujo de bits de datos para su ejecución en el componente, de los que al menos hay uno, desde una fuente de datos externa en el sistema de prueba, comprobación del flujo de bits de datos en el sistema de prueba y emisión de una señal de mando si el resultado de la prueba es positivo, para activar el componente con los datos del flujo de bits de datos.

Si el componente es por ejemplo una Soft-CPU sobre un FPGA, entonces puede desplazarse mediante el procedimiento el ancla de confianza al flujo de bits de carga del FPGA. De esta manera es posible una protección de la integridad del software contenido en la fuente de datos, como Boot Loader o también la configuración de un módulo FPGA, que sólo puede romperse tomando medidas costosas.

En una forma de realización ventajosa se comprueban los datos del flujo de datos de bits con ayuda de una función criptográfica y de un parámetro de validación en el sistema de prueba.

Se dispone así para elegir de diversos procedimientos, también conocidos, que pueden elegirse en función de las necesidades de protección de los datos a cargar o también en función de la potencia del módulo programable.

En una forma de realización ventajosa, se implementa el parámetro de validación mediante el flujo de bits de carga en el módulo.

Esto hace posible cargar en un componente, individualmente cada vez que se activa el módulo programable en memoria, distintos datos, por ejemplo diferentes características de un Boot Loader o sistema operativo y comprobar estos datos individualmente con antelación.

En una forma de realización ventajosa, se genera en el sistema de prueba a partir de los datos del flujo de bits de datos, mediante una función hash como función criptográfica, un valor de prueba y se comprueba respecto al parámetro de validación.

En otra forma de realización ventajosa, se utiliza como función criptográfica una función hash criptográfica y como parámetro de validación una clave secreta y se transmite en el flujo de bits de datos un valor de comparación generado con la misma clave secreta al sistema de prueba.

Esto tiene la ventaja de que pueden transmitirse y comprobarse diversos datos de un flujo de bits de datos sin tener que adaptar el parámetro de validación, aquí en particular la clave secreta. Con ello no es necesario ningún nuevo flujo de carga para configurar un parámetro de validación modificado.

En una forma de realización ventajosa, la función criptográfica es una función de validación de claves y el parámetro de validación es una clave pública. El flujo de bits de datos se transmite a la unidad de prueba firmado con una clave secreta correspondiente a la clave pública.

Una función de validación de claves para un procedimiento de codificación asimétrico, por ejemplo una infraestructura Private Key (de clave privada), ofrece una protección de la integridad especialmente elevada y tiene la ventaja, al igual que el procedimiento simétrico antes descrito con clave secreta, de no tener que introducir el parámetro de validación de nuevo en el módulo cuando varía el flujo de bits de datos. Además, en el citado procedimiento asimétrico sólo tiene que introducirse la clave pública, pero ninguna clave secreta, mediante el flujo de bits de datos en el módulo.

En una forma de realización ventajosa, el flujo de bits de datos contiene datos de un programa de arranque, también denominado Boot Loader, de un sistema operativo o de una configuración de módulo y la fuente de datos externa es en particular una unidad de memoria exterior al módulo programable en memoria o un módulo de memoria externo.

Además se reivindica un producto de programa informático que contiene partes de código de programa que son adecuadas para ejecutar las etapas del procedimiento reivindicado.

En los dibujos se muestran a modo de ejemplo ejemplos de realización del módulo programable en memoria correspondiente a la invención, así como del procedimiento reivindicado y se describirán más en detalle en base a la siguiente descripción. Se muestra en:

figura 1 un primer ejemplo de realización de un módulo programable en memoria correspondiente a la invención en representación de bloques,
figura 2 un segundo ejemplo de realización de un módulo programable en memoria correspondiente a la invención con unidad de seguridad adicional en representación de bloques y
figura 3 un ejemplo de realización del procedimiento correspondiente a la invención como diagrama secuencial.

Todas en las características descritas y/o señaladas pueden combinarse entre sí ventajosamente dentro del alcance de la invención. La invención no queda limitada a los ejemplos de realización descritos.

La figura 1 muestra un módulo programable en memoria, denominado a continuación también abreviadamente FPGA, que se configura mediante un flujo de bits de carga 9. En este proceso, también denominado síntesis, se configuran mediante el flujo de bits de carga 9 estructuras, también denominadas módulos o componentes, en el módulo programable en memoria, que realizan los más diversos circuitos. Estos circuitos se extienden desde sencillos contadores de sincronismo hasta circuitos muy complejos, como microprocesadores. En el módulo 10 representado se configuran un reloj 12, una interfaz de entrada 14, un interruptor controlable 15, así como un sistema de prueba 13. Un componente 11 está configurado como microprocesador. Allí está conectado el reloj 12 a través del interruptor controlable 15 con el componente. La interfaz de entrada 14 tiene una conexión con la fuente de datos externa 20, así como con el componente 11. Se prefiere entonces que sólo esté activa una de ambas conexiones. La interfaz de entrada 14 está conectada además con el sistema de prueba 13.

El sistema de prueba 13 presenta una conexión a través de la interfaz de entrada 14 con la fuente de datos 20 y una conexión con el interruptor controlable 15. La unidad de prueba 13 está además configurada para emitir una señal de mando 17 al interruptor controlable 15. El sistema de prueba 13 incluye además una función criptográfica 19, así como un parámetro de validación 18.

En el ejemplo representado no está realizado este procesador como circuito directamente mediante estructuras del módulo, sino como las llamadas Soft- o Softcore-CPU, que están integradas en el diseño FPGA. Una soft-CPU es igual a una CPU externa y proporciona una estructura estandarizada. Este componente 11 y a continuación el propio módulo, debe arrancarse entonces mediante un proceso de carga seguro, también denominado “secure boot” (arranque seguro), “trusted boot” (arranque fiable) o “verified boot” (arranque verificado). Para garantizar la integridad de un sistema, se activa el proceso de carga desde una unidad segura, usualmente denominada ancla de confianza. Para ello se carga en la primera etapa un Boot Loader y a continuación un sistema operativo y otro software en el FPGA o bien a través del FPGA en otros módulos de un sistema embebido. Los datos, por ejemplo el Boot Loader, el sistema operativo o también otros datos, están memorizados entonces en una fuente de datos externa 20 y desde allí, mediante un flujo de bits de datos 16, en el módulo 10. La fuente de datos externa 20 puede ser por ejemplo un módulo de memoria flash o una memoria de sólo lectura, también denominada ROM, que está implementada sobre otro chip de memoria o en general como medio de memoria externo, como por ejemplo un lápiz USB, una tarjeta SD, un Compact Disc o similar.

Después de configurar el módulo 10 mediante el flujo de bits de carga 9 se activa automáticamente el sistema de prueba 13. La conexión entre el reloj 12 y componentes 11 está seccionada mediante el interruptor controlable 15, con lo que el componente no está activo. Mediante la interfaz de entrada 14 se transmiten datos en un flujo de bits de datos 16 al sistema de prueba 18. El sistema de prueba incluye además una función criptográfica 19, así como un parámetro de validación 18, que por ejemplo puede ser un valor hash o una clave secreta o una clave pública, que se utiliza junto con la función criptográfica para comprobar el flujo de datos 16 a cargar.

El interruptor controlable 15, que está intercalado entre el reloj y el componente 11, es una posibilidad sencilla de activar y/o desactivar de forma dedicada el componente 11. La invención descrita puede utilizarse también con otros equipos técnicos para activar y/o desactivar el componente 11. Éstos no se describen explícitamente, pero un especialista en el sector de módulos programables en memoria los conoce.

En la figura 2 se representa otra variante de realización 30 de un módulo programable en memoria. Además de las estructuras 11, 12, 13, 14, 15, la función 18 y el parámetro de validación 19 representados en la figura 1, se implementa en el módulo programable en memoria 30 una unidad de seguridad 31 adicional, que constituye un ancla de confianza dedicada. Esta unidad de seguridad 31 está conectada por un lado con el componente 11, es decir, por ejemplo una Soft-CPU y por otro lado con el interruptor

controlable 15. El interruptor controlable 15 está conectado a su vez con el reloj 12 y es controlado por la unidad de prueba 13.

En base a la figura 3 se describirá ahora el procedimiento para la transmisión protegida de datos desde una fuente de datos externa 20 a al menos un componente 11 de un módulo programable en memoria 10 ó 30.

En una primera etapa del procedimiento 40 se genera mediante un flujo de bits de carga 9 un sistema de prueba 13 sobre el módulo 10, 30. Esto puede realizarse por ejemplo en el marco de un proceso de síntesis junto con la configuración de los otros módulos y/o componentes 11, 12, 13, 14, 15, función 18 y parámetro de validación 19 descritos. El interruptor programable 15 está desactivado, con lo que el componente 11, por ejemplo una Soft-CPU, no se pone en funcionamiento. En los módulos 10, 30 representados en las figuras 1 y 2 está seccionada la conexión entre reloj 12 y componente 11. La interfaz de entrada 14 está conectada con la unidad de prueba 13. Opcionalmente puede estar conectada la interfaz de entrada 15 a la vez también con el componente 11. No obstante, los datos a leer no se procesan en el componente 11, ya que el componente 11 no está conectado con el reloj 12 y está por lo tanto inactivo.

A continuación se lee en la etapa 41 del procedimiento al menos un flujo de bits de datos para la ejecución en el componente 11, de los que al menos hay uno, del módulo 10, desde la fuente de datos externa 20 al sistema de prueba 13. Se carga por lo tanto por ejemplo el Boot Loader Code para un componente constituido como Soft-CPU a partir de una zona de memoria previamente definida de la fuente de datos 20. En el sistema de prueba 13 se procesa entonces el flujo de bits de datos 16, aquí el Boot Loader Code, con la función criptográfica 19. Por ejemplo está implementada como función criptográfica una función hash, por ejemplo SHA256 19 o SHA3, que a partir del flujo de bits de datos 16 determina un valor hash. En la etapa 42 del procedimiento se compara ese valor hash con el parámetro de validación 19 y así se comprueba el flujo de bits de datos. El parámetro de validación 19 es aquí el valor hash, formado mediante el flujo de bits de datos del Boot Loader Code a cargar con la misma función hash y que por ejemplo se configuró con el flujo de bits de carga 9 en el módulo 10, 30. El parámetro de validación 19 está implementado por ejemplo en el sistema de prueba 13 o la unidad de prueba y puede acceder a un parámetro de validación implementado fuera de la unidad de prueba 13 en el módulo 10, 30.

Si el resultado de la comprobación es positivo, se cierra el interruptor controlable 15, se conecta el reloj 12 con el componente 11 y con ello se arranca el componente 11. Si solamente estaba conectada la interfaz de entrada con el sistema de prueba 13, se libera entonces la conexión entre la interfaz de entrada 14 y el componente 11, con lo que el flujo de bits de datos se carga ahora en el componente 11. Una carga del flujo de bits de datos 16 sólo es posible tras una comprobación de integridad con éxito del flujo de bits de datos; véase la etapa 43. Si el resultado de la comprobación es negativo, permanece abierto el interruptor controlable 15 e impide así el arranque de software manipulado procedente de la fuente de datos externa, por ejemplo una memoria flash.

Si se ejecuta el procedimiento sobre el módulo programable en memoria 30 representado en la figura 2, entonces se activa la unidad de seguridad 31 adicional sólo tras validar el flujo de bits de datos 16. Así se evita que se integre software malicioso procedente de la fuente de datos externa 20, que haría mal uso de la unidad de seguridad 31, por ejemplo un ancla de seguridad, o la atacaría. La secuencia del procedimiento descrita para el módulo 10 sigue siendo válida.

Para comprobar el flujo de bits de datos 16 puede utilizarse en el sistema de prueba 13 por ejemplo como función criptográfica una función hash criptográfica, como por ejemplo la HMAC-SHA256. Como parámetro de validación se proporciona en este caso una clave secreta al módulo. La función criptográfica 19 es aquí por ejemplo un algoritmo HMAC-SHA256, que con ayuda de la clave secreta y aplicado al flujo de bits de datos 16, determina un llamado Message Authentication Code (código de autentificación de mensajes) codificado. El mismo Message Authentication Code codificado se había calculado previamente con la misma clave secreta mediante los datos del flujo de bits de datos 16 y se había memorizado en la fuente de datos externa 20. El código calculado por el sistema de prueba 13 se compara ahora con el código previamente calculado. Si existe coincidencia y por lo tanto el resultado positivo de la comprobación es positivo, se activa el componente 11 y se libera la interfaz de entrada 14 para transmitir el flujo de bits de datos 16 al componente 11. El código de autentificación de mensajes calculado en la fuente de datos externa 20 puede cargarse también con el flujo de bits de datos 16 en el sistema de prueba 13 y comprobarse allí con el parámetro de validación 18 o bien el código determinado. Esto tiene la ventaja de que pueden comprobarse cualesquiera y en particular distintos flujos de bits de datos mediante el sistema de prueba sin tener que configurar de nuevo el módulo 10, 30 mediante el flujo de bits de carga. Por otro lado, debe aportarse la clave secreta mediante el flujo de bits de datos al módulo. El flujo de bits de datos se comprueba así mediante un procedimiento criptográfico simétrico.

Como función criptográfica 19 puede utilizarse también una función de validación de firma. Entonces está memorizada como parámetro de validación 18 la clave pública de un procedimiento criptográfico

asimétrico o par de claves. El flujo de bits de datos 16 se transmite entonces firmado, incluyendo por lo tanto una firma digital, que se había determinado con la clave secreta del procedimiento criptográfico asimétrico mediante el flujo de bits de datos 16. La función de validación de la firma en el sistema de prueba 13 descifra con la clave pública la firma digital del flujo de bits de datos 16 y la compara con un valor hash formado en la unidad de prueba mediante el flujo de bits de datos 16. Si coinciden ambos valores, el resultado de la comprobación es positivo y se activa la unidad de seguridad 31 o bien el componente 11. Un tal procedimiento asimétrico tiene respecto a un procedimiento simétrico la ventaja de que en el flujo de bits de carga no se memoriza ninguna clave secreta, no teniendo por lo tanto que cargarse. Solamente se implementa la clave pública, que no es ningún secreto y que se conoce públicamente, con el flujo de bits de datos 9 en el módulo 10, 30.

Si el resultado de la comprobación es negativo, puede también desactivarse solamente una función parcial del componente 11 o bien activarse uno o varios componentes con datos falsos, con lo que una utilización de estos componentes es imposible o sólo posible con limitaciones. Además no sólo pueden comprobarse datos que se aportan para ejecutarlos en un componente 11, sino que pueden comprobarse flujos de bits de datos que se cargan en diversos componentes distintos.

En vez de los datos del sistema Boat Loader o del sistema operativo que están archivados en una memoria, pueden comprobarse también datos de la configuración de otros módulos, por ejemplo de un chip de interfaz para WLAN, chips para Ethernet o también para telefonía móvil. Igualmente puede utilizarse el procedimiento para comprobar una configuración de hardware y la presencia de determinados módulos o marcadores de salto, los llamados Jumpers o saltadores.

Todas las características descritas y o dibujadas pueden combinarse entre sí ventajosamente dentro del alcance de la invención. La invención no queda limitada a los ejemplos de realización descritos.

REIVINDICACIONES

1. Módulo programable en memoria con ancla de confianza que puede cargarse, que incluye un sistema de prueba (13) y al menos otro componente (11),
en el que el sistema de prueba (13) puede generarse mediante un flujo de bits de carga (9) sobre el módulo (10) y el componente (11), de los que al menos hay uno, está desactivado automáticamente tras la carga, estando constituido el sistema de prueba (13) para leer al menos un flujo de bits de datos (16) para ejecutarlo en el componente (11), de los que al menos hay uno, del módulo (10) desde una fuente de datos externa (20), comprobar el flujo de bits de datos (16) y, si el resultado de la comprobación es positivo, emitir una señal de mando (17) que activa el componente (11).

2. Módulo programable en memoria según la reivindicación 1,
en el que el sistema de prueba (13) incluye una función criptográfica (19) y está diseñado para comprobar los datos del flujo de datos con ayuda de un parámetro de validación (18) y de la función criptográfica (19).

3. Módulo programable en memoria según la reivindicación 2,
en el que la función criptográfica (19) es una función hash y el parámetro de validación (18) es un valor hash del flujo de bits de datos (16) creado con la función hash.

4. Módulo programable en memoria según la reivindicación 2,
en el que la función criptográfica (19) es una función hash criptográfica y el parámetro de validación (18) es una clave secreta.

5. Módulo programable en memoria según la reivindicación 2,
en el que la función criptográfica (19) es una función de validación de firma y el parámetro de validación (18) es una clave pública.

6. Módulo programable en memoria según la reivindicación 2,
en el que el parámetro de validación se puede implementar con el flujo de bits de carga (9) en el módulo (10).

7. Módulo programable en memoria según una de las reivindicaciones 1 a 6,
que incluye una unidad de seguridad (31) adicional, que está conectada con el componente a programar y el sistema de prueba (13) y que puede ser activada mediante la señal de mando (17) del sistema de prueba (13).

8. Módulo programable en memoria según una de las reivindicaciones 1 a 7,
en el que el sistema de prueba (13) está diseñado para generar una señal de mando (17) para desactivar el componente (11, 31) o al menos una función parcial del componente (11, 31), si el resultado de la comprobación es negativo o para activar el componente (11, 31) con un segundo flujo de bits de datos, que hace imposible o sólo limitadamente posible una utilización del componente (11, 31).

9. Módulo programable en memoria según una de las reivindicaciones 1 a 8,
que incluye un interruptor controlable (15), que conecta el componente (11) o el ancla de confianza con un reloj (12) mediante la señal de mando (17) del sistema de prueba (13) y por lo tanto activa el componente o el ancla de confianza o lo desconecta, desactivándolo.

10. Procedimiento para la transmisión protegida de datos desde una fuente de datos externa a al menos un componente (11) de un módulo programable en memoria, con las etapas:
- Generación (41) de un sistema de prueba (13) en el módulo (10) mediante un flujo de bits de carga (9) y desactivación del componente (11), de los que al menos hay uno,
- lectura (42) de al menos un flujo de bits de datos (16) para su ejecución en el componente (11), de los que al menos hay uno, del módulo (10) desde una fuente de datos externa (20) en el sistema de prueba (13),
- comprobación (43) del flujo de bits de datos (16) en el sistema de prueba y
- emisión (44) de una señal de mando (17) si el resultado de la comprobación es positivo, para activar el componente (11) con los datos del flujo de bits de datos (16).

11. Procedimiento según la reivindicación 10,
en el que se comprueban los datos del flujo de datos de bits con ayuda de una función criptográfica y de un parámetro de validación en el sistema de prueba (13).

12. Procedimiento según la reivindicación 11,
en el que se implementa el valor de validación (18) mediante el flujo de bits de carga (9) en el módulo.

13. Procedimiento según una de las reivindicaciones10 a 12,
en el que en el sistema de prueba (13) se genera a partir de los datos del flujo de bits de datos (16) mediante una función hash como función criptográfica (19), un valor de prueba y se comprueba respecto al parámetro de validación (18).

14. Procedimiento según una de las reivindicaciones10 a 12,
en el que la función criptográfica (19) es una función hash criptográfica y el parámetro de validación (18) es una clave secreta y se transmite en el flujo de bits de datos (16) un valor de comparación generado con la misma clave secreta al sistema de prueba (13).

15. Procedimiento según una de las reivindicaciones10 a 12,
en el que la función criptográfica (19) es una función de validación de claves y el parámetro de validación (18) es una clave pública y el flujo de bits de datos se transmite a la unidad de prueba (13) firmado con una clave secreta correspondiente a la clave pública.

16. Procedimiento según una de las reivindicaciones10 ó 15,
en el que el flujo de bits de datos (16) contiene datos de un programa de arranque, de un sistema operativo o de una configuración de módulo y la fuente de datos externa es una unidad de memoria (20) exterior al módulo o un módulo de memoria externo.

17. Producto de programa informático que contiene partes de código de programa que son adecuadas para ejecutar las etapas del procedimiento según una de las reivindicaciones 10 a 16.

FIG 1

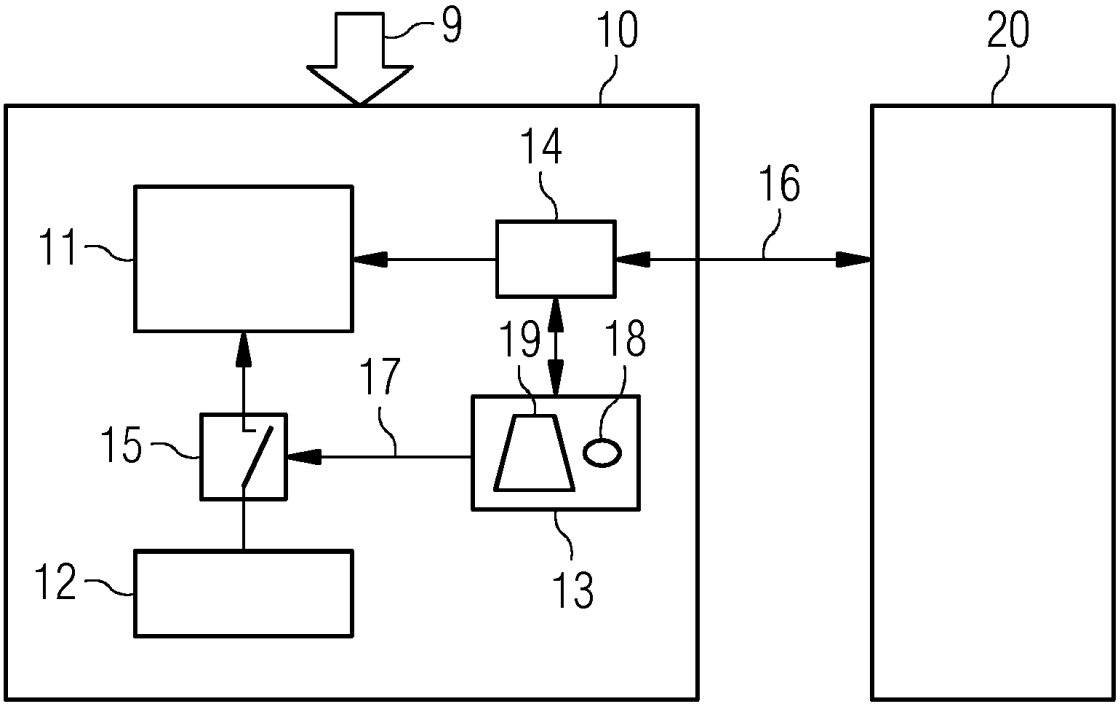

FIG 2

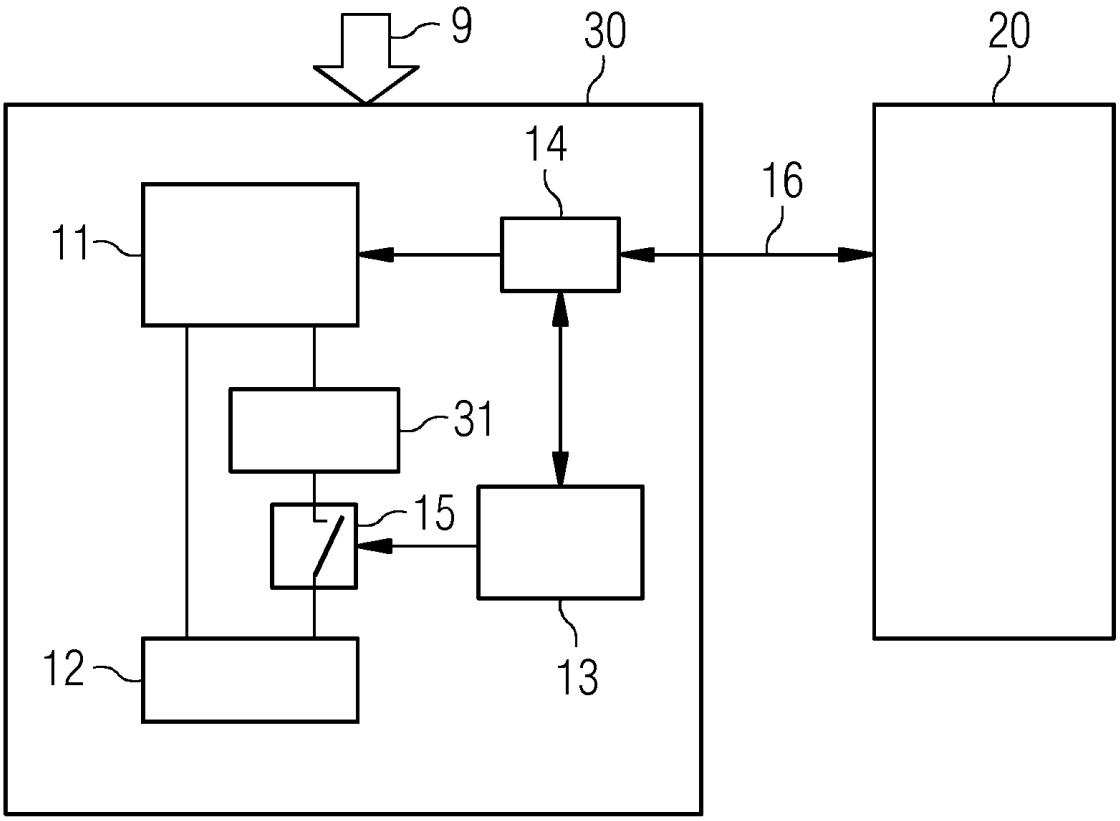

FIG 3